CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

APLICAÇÃO DE RECONHECIMENTO DE PADRÕES EM UM EXPERIMENTO LINGUÍSTICO

Ali Kamel Issmael Junior

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadora: Aline Gesualdi Manhães
Coorientador: José Vicente Calvano

Rio de Janeiro
Março de 2017

APLICAÇÃO DE RECONHECIMENTO DE PADRÕES
EM UM EXPERIMENTO LINGUÍSTICO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Ali Kamel Issmael Junior

Banca Examinadora:

Presidente, Professora Dra. Aline Gesualdi Manhães (CEFET/RJ) (orientador)

Dr. José Vicente Calvano (MARINHA DO BRASIL) (Coorientador)

Professora Dra. Marije Soto (UERJ)

Professor Dr. Thiago de Moura Prego (CEFET/RJ)

Rio de Janeiro
Março de 2017

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

I86 Issmael Junior, Ali Kamel
Aplicação de reconhecimento de padrões em um experimento linguistico / Ali Kamel Issmael Junior.—2017.
xvi, 90f. + apêndices : il. (algumas color.) , grafs. , tabs. ; enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca , 2017.
Bibliografia : f. 88-90
Orientadora : Aline Gesualdi Manhães
Coorientador : José Vicente Calvano

1. Potencial evocado (Eletrofisiologia). 2. Eletroencefalografia. 3. MATLAB. 4. Engenharia elétrica. I. Manhães, Aline Gesualdi (Orient.). II. Calvano, José Vicente (Coorient.). III. Título.

CDD 616.8047547

Dedicatória

Ao meu grande amigo Roberto (*in memoriam*).

Á minha mãe Maria de Fátima e à minha esposa Janayna.

Agradecimentos

À Deus, que me deu a força necessária para enfrentar mais este desafio em meio a tantas turbulências em minha vida particular.

Ao meu grande amigo, exemplo de luta e resistência, Roberto Ferreira Penedo, que sempre me apoiou e que agora está com Deus, comemorando mais esta conquista. Nunca te esquecerei meu amigo!

À minha mãe, Maria de Fátima Azevedo Penedo, à minha querida esposa, Janayna Menezes Vilela Issmael, à minha irmã Linda Soraya Issmael e meu sobrinho Joao Pedro Issmael Vieira, pelo apoio e amor incondicional por mim, e que me sustentaram em um período tão conturbado de minha vida.

À Professora Aline Gesualdi Manhães, pela orientação segura, compreensão dos problemas pelo qual passei e pela minha condução até este momento de vitória e realização.

Ao Capitão de Mar e Guerra (EN) José Vicente Calvano, pelo incentivo e apoio na coorientação, apesar das diversas tarefas e responsabilidades no nosso trabalho na Marinha do Brasil.

Aos professores do PPEEL, que puderam me passar sua experiência e conhecimento em cada disciplina que estudei, e ao pessoal da Secretaria da Pós que sempre nos ajudaram com presteza nas tarefas administrativas do curso.

Ao amigo Rivaldo da Silva Alves Junior, que desde 2004, quando servíamos no Centro de Mísseis e Armas Submarinas da Marinha (CMASM) e começamos a cursar disciplinas isoladas no Instituto Militar de Engenharia (IME), sempre me incentivou a não desistir do sonho de conseguir concluir o Mestrado e seguir minha carreira como Engenheiro, Militar e Acadêmico, lutando pelas minhas convicções.

Aos amigos e companheiros de caminhada no Mestrado, Flávio, Renata, Luís, Dionísio, Thiago, Douglas, Júlio, Wandemberg, Ricardo Campisano, Giovani, entre outros.

RESUMO

A técnica de “Event-Related Potentials” (ERP) consiste na medida de sinais biológicos cerebrais, de natureza elétrica, obtidos por meio de Eletroencefalografia (EEG), que sejam resultados diretos de estímulos a eventos sensoriais, cognitivos ou motores. Desta forma, a técnica ERP permite a análise não invasiva do funcionamento do cérebro. A partir dos resultados obtidos com o experimento de estímulos de computação linguística para palavras e sentenças, proposto por Soto (2014), do tratamento destes dados e a extração de parâmetros ERP, por meio das ferramentas EEGLAB® e ERPLAB®, baseadas no programa de simulações Matlab® (“matrix laboratory”), o resultado da pesquisa foi a obtenção de cenários de classificação supervisionados e não supervisionados, das classes propostas para o experimento mencionado e o estudo comparativo e discussão dos resultados de classificação encontrados, utlilizando a metodologia proposta por Webb (2002). Este trabalho se revelou inovador na área de Linguística, por não terem sido encontrados, ao menos até o presente momento, trabalhos similares em bases de dados de pesquisa como IEEExplorer, Web of Science, Elesevier e Spring. Foram alcançados excelentes resultados para classificação supervisionada, sendo o classificador “Random Forest” o que atingiu uma acurácia total de 100%, sendo seguido pelos classificadores “Multiclass Support Vector Machine” (MSVM) e “Naïve Bayes”, com ambos os métodos atingindo precisões totais superiores a 96%. Os resultados indicam que as abordagens não-lineares foram mais adequadas para classificar os dados da configuração da experiência ERP de Soto (2014) e que os resultados também abrem a possibilidade de se analisar sinais de indivíduos com essa metodologia ERP associada à Reconhecimento de Padrões, com possível aplicação desse tipo de análise em ferramentas diagnósticas, de avaliação de aprendizagem linguística, entre outras.

Palavras-chave: ERP. EEG. Reconhecimento de Padrões. Computação Linguística

ABSTRACT

The "Event-Related Potentials" (ERP) technique consists of the measurement of electrical biological signals obtained by electroencephalography (EEG), which are direct results of stimuli to sensory, cognitive or motor events. In this way, the ERP technique allows the non-invasive analysis of brain functioning. Based on the results obtained with Soto (2014), the treatment of these data and the extraction of ERP parameters, using the EEGLAB® and ERPLAB® tools based on the simulation program Matlab® ("matrix laboratory"). The result of the research was the obtaining of supervised and unsupervised classification scenarios for the classes proposed in the mentioned experiment and the comparative study and discussion of the classification results found, using the methodology proposed by Webb (2002). This work proved to be innovative in the area of Linguistics, since, at least until now, no similar work has been found in research databases such as IEEExplorer, Web of Science, Elesevier and Spring. Excellent results for supervised classification, with the "Random Forest" classifier reaching a total accuracy of 100%, followed by the "Multiclass Support Vector Machine" (MSVM) and "Naïve Bayes" classifiers, with both methods reaching precisions rreater than 96% were obtained. The results indicate that non-linear approaches were more adequate to classify Soto (2014) ERP configuration data and the results also open the possibility of analyzing signals from individuals with this ERP methodology associated to Pattern Recognition, with the possible application of this type of analysis in diagnostic tools, assessment of language learning, among others.

Keywords: ERP. EEG. Pattern Recognition. Linguistic Computations

SUMMARY

Figure List

Table List

Abreviations

ASR	Associative Semantic Relation
CART	Classification And Regression Tree
CNSC	Congruous Non-supportive Context
CNV	Contingent Negative Variation
CSC	Congruous Supportive-Context
ECOC	Error-correcting output codes (Matlab® function)
EEG	Electroencephalography
EPSP	Excitatory Postsynaptic Potential
ERP	Event-Related Potentials
FPR	False Positive Rate
GMM	Gaussian Mixture Model
HCA	Hierarchical Clustering Analysis
ICA	Independent Component Analysis
INSC	Incongruous Non-Supportive Context
ISC	Incongruous Supportive-Context
MEG	Magnetoencephalography
MN	Multinomial distribution
ms	Miliseconds
MVMN	Multivariate Multinomial distribution
MSVM	Multiclass Support Vector Machine
PW	Pseudo Word
RBF	Radial basis function (Gaussian)
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SN	Signal-to-Noise ratio
SOA	Stimulus Onset Asynchrony
SSR	Syntactic and Semantic Relation
SVM	Support Vector Machine
TPR	True Positive Rate
UR	Unrelated Pair

Chapter I - Introduction

Event-Related Potentials (ERP) are electrical voltages associated with a neurophysiological response induced by an external event or stimulus. ERP are obtained by means of Electroencefalography (EEG), which is a non-invasive aparatus sensible enough to measure small electrical potentials in a human scalp, as a result of the stimulation by sensory, cognitive or motor events.

This study uses the ERP experimental data of Soto (2014) that addressed underlying cognitive functions measured on target words in sentential and word priming contexts, from this ERP experiment data., is possible to treat these experimental data to investigate if there are specific parameters to discriminate the ERP signals related to each kind of stimuli. The treatment of these ERP signals involves the study of digital signal processing techniques applied with pattern recognition theory and used specific computer tools, such as EEGLAB® and ERPLAB®, based on the software Matlab®.

I.1 Motivation

As described by Gesualdi (2011), a language cognitive ERP measurement usually entails the collaboration of a multidisciplinary team of researchers, including linguists, signal processing engineers, speech therapists, psychologists, neuroscientists, among others.

This multidisciplinarity motivates the interaction and cooperation between the Graduate Program in Linguistics of the Federal University of Rio de Janeiro (UFRJ) and Graduate Program in Electrical Engineering of Federal Center of Technological Education Celso Suckow da Fonseca, CEFET/RJ. This joint work between these institutions produced the work of Soto (2014) that studied underlying cognitive function of the ERP component known as N400. Normally, as done by Soto (2014) in her work, the ERP signal interpretation used time and frequency domain analysis.

After checking on the references available in databases as IEEEXplorer, Web of Science, Elsevier, and Spring, it was not found any linguistic experiments with a study of the use of pattern recognition applied on linguistic ERP signals and also the methods and techniques from Cong et all (2015) and Kamel and Malik (2015). This indicates that this work is an innovative way of analyze this kind of data. Thus, this motivates this study, which has the goal to use the pattern recognition methodology proposed by Webb (2002) on the data from the Soto (2014) experiment.

This work is part of an effort of UFRJ, UERJ and CEFET-RJ groups to develop a methodology that will be used in other studies, involving data collection and analisys.

I.2 Objective

The goal of this work is to investigate through the behavior of the brain waves, or the EEG and ERP signals, classification scenarios considering different types of stimuli, in order to verify differences in sentences and words classes proposed by Soto (2014), by applying the pattern recognition methodology proposed by Webb (2002) on the ERP results from the Soto (2014) data experiment.

I.3 Organization

The dissertation is organized as follows: Chapter II describes the fundamentals of the ERP experiment developed by Soto (2014), the use of the software tools EEGLAB®, ERPLAB® and Matlab® to extract the ERP information used in the research and the Pattern Recognition Theor; Chapter III presents the methodology used to treat the ERP information and the creation of the Pattern Recognition Algorithm. Chapter IV presents the results found; Chapter V presents the conclusions and future developments for this work and Chapter VI presents the Bibliography used in this study.

Chapter II - Theoretical References

II.1 EEG and ERP theory

As already mentioned in the Introduction (GESUALDI, 2011), "Event-Related Potentials" (ERP) is an averaging technique that can be used emphasize specific neurophysiologic response by the measure of electrical signals from the brain obtained by means of Electroencephalography (EEG). ERP enhances signal obtained of the stimulation by sensory, cognitive or motor events. In Figure 1, the functional diagram of human brain lobes is shown, where is possible to see the associated funtions for each part of the brain.

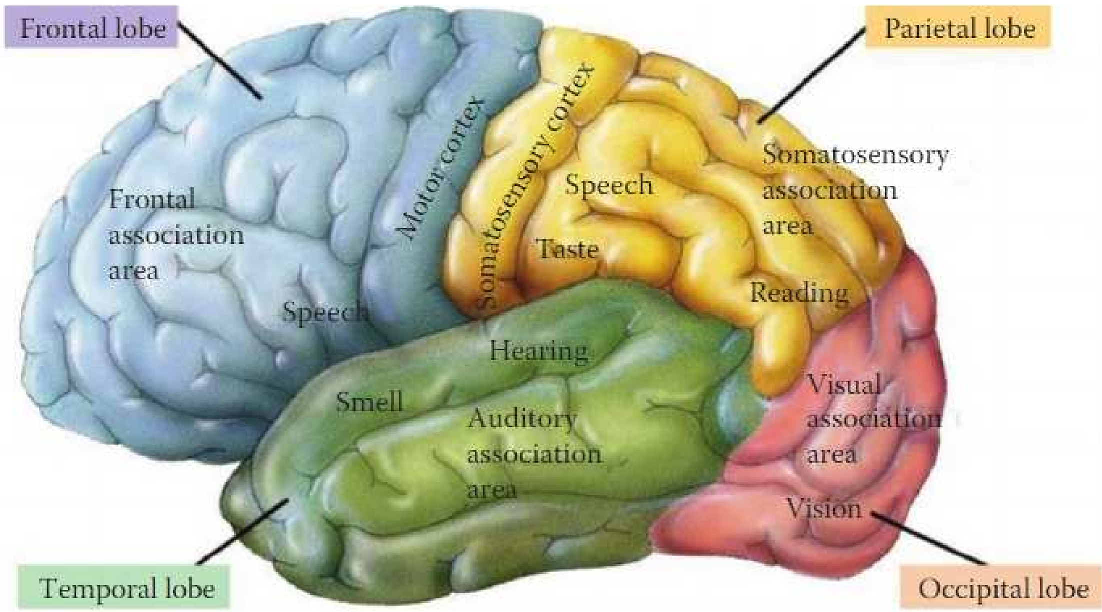


Figure 1 - Functional diagram of human brain lobe (KAMEL; MALIK, 2015)

As described by Kamel and Malik (2015), the human brain is part of the central nervous system (CNS) and consists of nerve and glia cells. Gesuladi (2011) explain that

> On the microscopic level, the brain is constituted by two cell types: neuroglia (glia, the Greek for glue) and neurons. Neuroglia are about fifty times more numerous than neurons in the brain. Neuroglial cells, or simply glial cells, are peculiarly branched cells arranged in a fine web of tissue. Traditionally, they were only regarded as the supporting structure of nervous tissue. However, neuroglia are now known to be involved in neuronal growth and migration.

In figure 2, the three types of glial cells in the brain (astrocytes, oligodendrocytes and microcytes is shown.

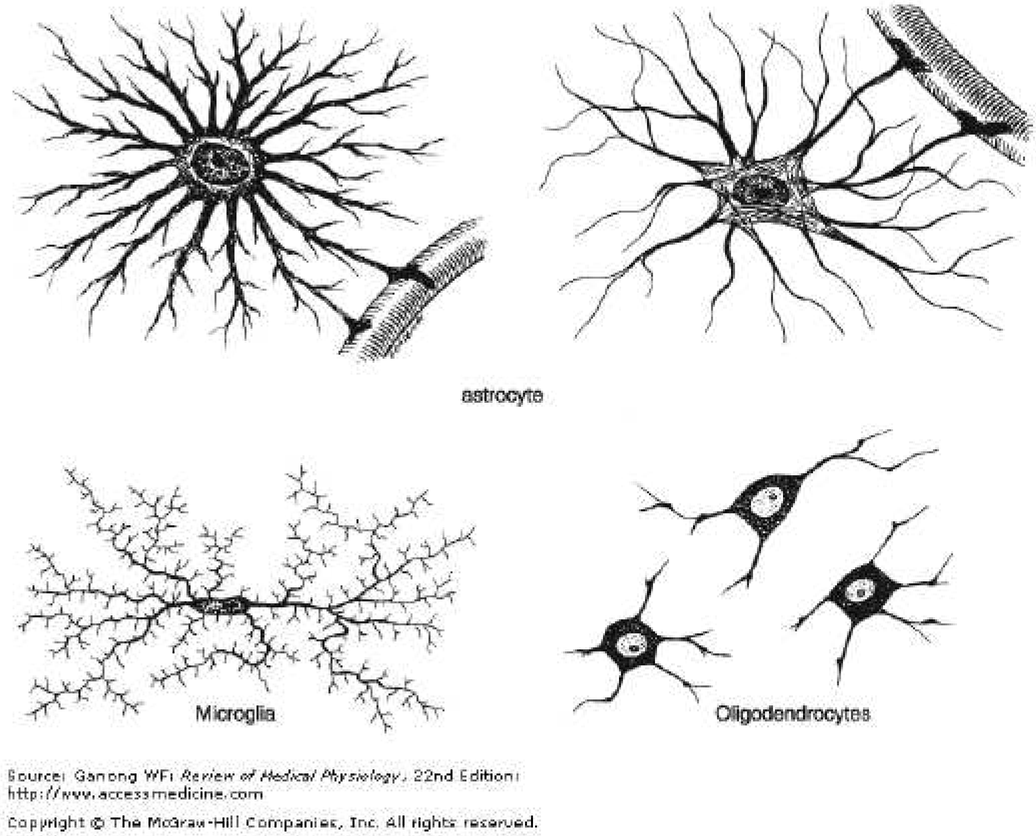


Figure 2 - The three types of glial cells: astrocyte, microglia, oligodendrocytes (GESUALDI, 2011)

As described by Gesualdi (2011), astrocytes are star-shaped cells that have as main functions filling in the space within neurons, holding neurotransmitters, that are released by neurons; provide part of the fundamental control of chemical concentrations in extracellular space; and also clean up brain waste and digest dead neurons. Oligodendrocytes (astroglia and oligodendroglia) are involved in insulation to the nerve fibers and provides transport of material to neurons and the control of the ionic environment of neurons. In addition, the microcytes (microglia) are the smallest of the glial cells, phagocytizing waste products of nerve, and play an important role in the protection of the brain from invading microorganisms (GESUALDI, 2011).

The neurons are functional units of the nervous system and an adult human brain contains, on average, 100 billion neurons and these cells process and transmit information through electrical and chemical signals (KAMEL; MALIK, 2015). Gesualdi (2011) explain that neurons operate in large sets forming neuronal circuits or neural nets and they produce electrical signals that operate as information bits, where all data reaching or leaving the body is transformed into electrical signals that are transported and processed by neurons.

As described by Gesualdi (2011),

> The cell body contains the nucleus and other cell organelles such as mitochondrion, rough and smooth endoplasmatic reticula, lysosomes, ribosomes and Golgi complex, that are responsible for cell metabolic maintenance. Departing from the cell body are dendrites and axons: cell extensions, also named processes. Dendrites are tree-like structures that collect electrical signals from other nerve cells. Again, neurons are electrically excitable and are known to be able to transmit this excitation. So, an electrical signal reaches a dendrite, goes through the cell body, which generates outgoing signal down the axon in the form of the action potential mentioned above.

Kamel and Malik (2015) indicate that neurons have a resting membrane potential of about -70 to 60 mV and depending on the information the dentrites receive form other neurons, the neuron makes a decision that is sent to the other neurons` dentrites over the axon. This explanation can be weel understood in Figure 3.

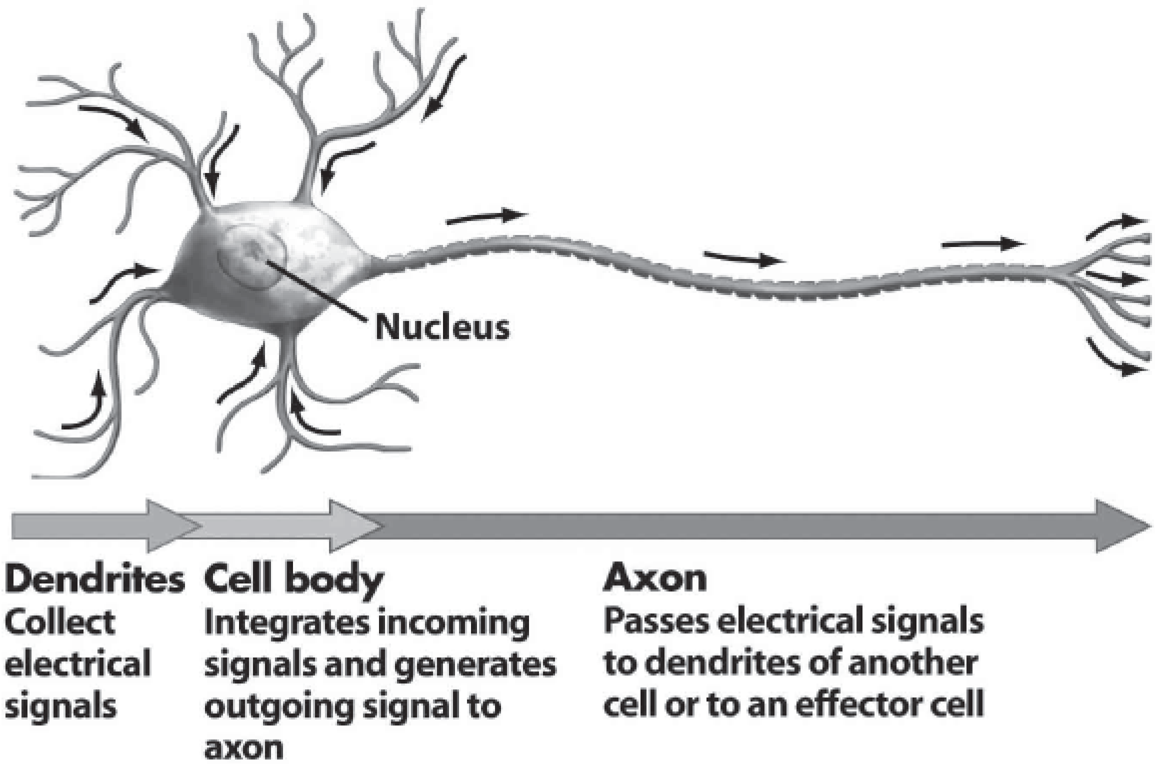


Figure 3 - The neuron functioning (GESUALDI, 2011)

Kamel and Malik (2015) explain in a briefly way that the neuron action potentials are electrical impulses created by the brain and transmiited by the neurons to the different parts of the body:

> The neuron is surrounded by the cell membrane, which controls the in and out movement of charged sodium (Na+) and potassium (K+) ions. The cell body is negatively charged on the outside and has a resting potential of −70 mV. The membrane potential becomes less negative because of the incoming electrical current from the dendrites (see A in Figure 5) [8,14]. The cell membrane completely opens up for Na+ ions if this depolarization reaches −55 mV, when the ions now enter the cell, resulting in a momentarily positive action potential (see B in Figure 5). The cell membrane also opens up again, and the K+ ions present in the cell leave (see C in Figure 5), which causes the repolarization of the membrane's potential. Owing to the loss of permeability of the cell membrane for the K+ ions, the potential temporarily falls below 70 mV (this is known as hyperpolarization; see D in Figure 5). Finally, the action potential stabilizes at the resting potential. This action potential is carried over by the axon to other neurons. Hence, the neuron becomes activated if the total electrical current from all the incoming axons exceeds a certain threshold. This results in the transfer of information (the action potential) to subsequent neurons.

The neuron action potential related with the citation can be seen in figure 4.

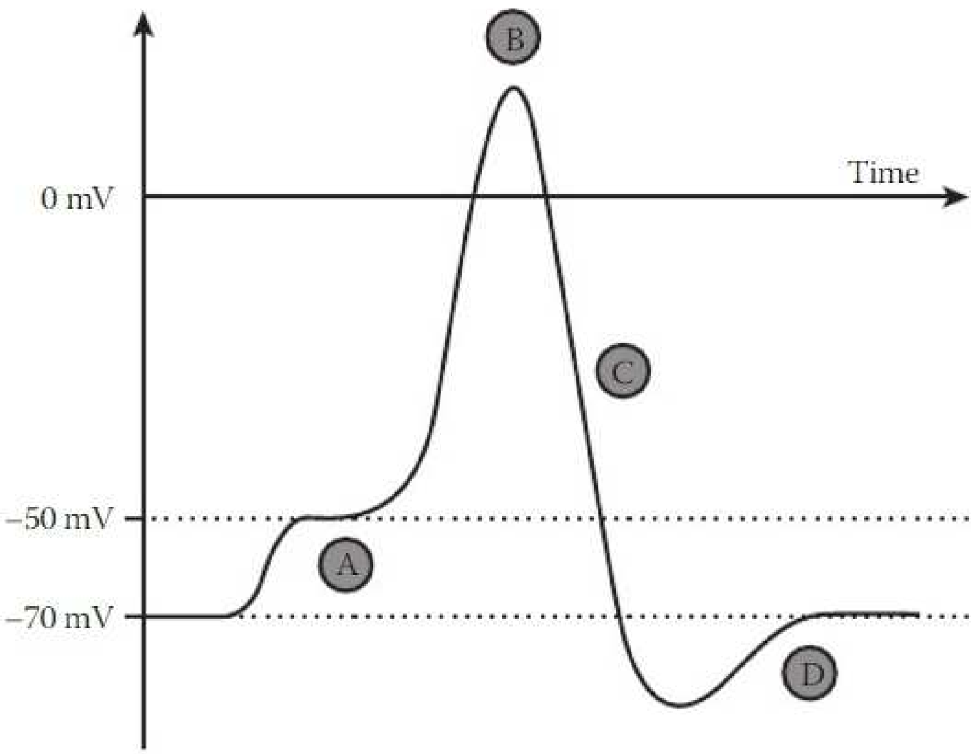


Figure 4 - Neuron action potential (KAMEL; MALIK, 2015)

Gesualdi (2011) explain that this between-neuron transmission system is named synapse, and the action potential can travel at high speed along an network of neurons. The synapses are divided in two types: electrical and chemical. The electrical synapse is related with the direct connection between two closest neurons, where there are a direct transfer of the electrical signal between them. The chemical synapse is related to a contact zone between two neurons, where the action potential to the axonal buttons triggers the release of molecules of neurotransmitters that are kept in synaptic vesicles. These molecules reach the synaptic cleft (the contact zone with the next neuron) and are absorbed by receptors, ion channels, placed at the post-synaptic membrane of the dendrites of the next cell, that open for the active synapse and close for the resting synapse. Because of this, these signals are called and considered post-synaptics. Depending on the neurotransmitter released, the action potential that will flow to the next neuron may be propagated, blocked or modified, modulating the electrical bits being transmitted along the neuron. (GESUALDI, 2011). This can be seen in Figure 5.

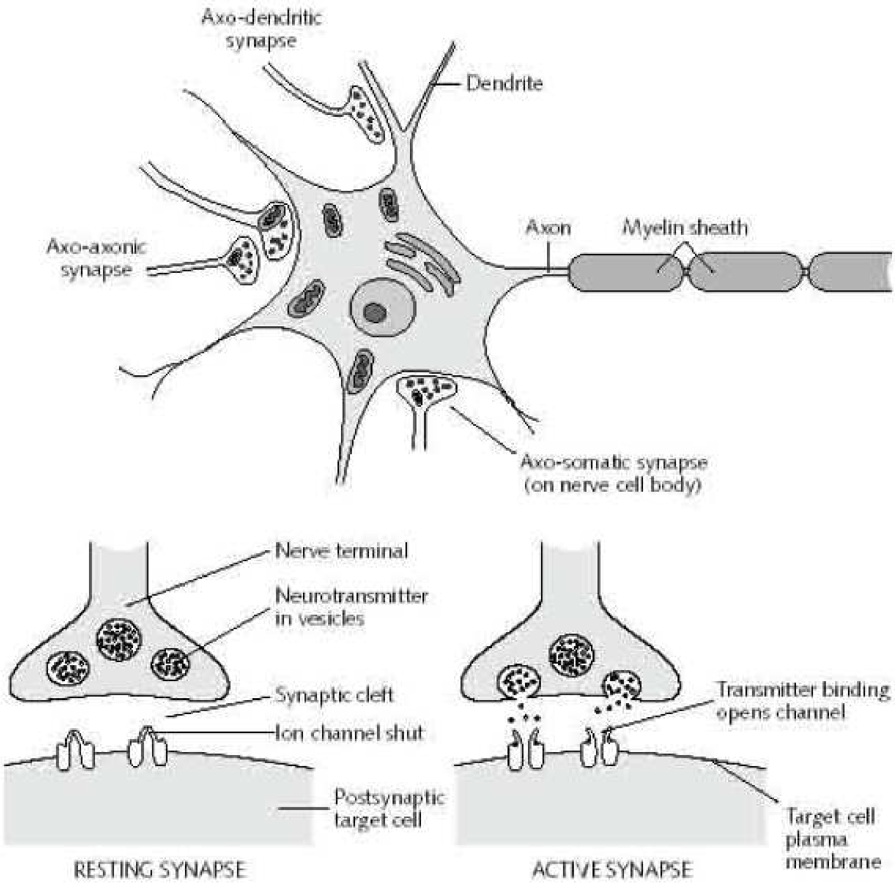


Figure 5 - The chemical synapse (GESUALDI, 2011)

According Soto (2014), potential fluctuation is called excitatory postsynaptic potential (EPSP) and is result of the action of neurotransmitters on the receptors of postsynaptic cells, manipulating the ion channels (by opening or closing). However, the EEG cannot measure the action potential of a single neuron. Kamel and Malik (2015) explain that

> The EEG activity is a reflection of the summation of the synchronous activity of a big group of neurons, probably thousands or millions, having a similar spatial orientation. A similar spatial orientation is necessary for the ions to line up and create waves that will be strong enough to pass the detection threshold. It is known that the voltage fields fall off with the square of the distance, and hence the pyramidal neurons of the cortex that are well aligned and fire together are thought to produce most EEG signals. The EEG activity from deep sources is difficult to detect as compared to the activity that happens near the skull.

Kamel and Malik (2015) indicate also that these neuron action potentials working together create neuron osclillations called brain rhytms and wave patterns. These osclillations are generated by large groups of neurons and can be characterized by the frequency, amplitude, and phase of the oscillations. Cognitive functions such as information transfer, perception, motor control, and memory are in one way or another related to neural oscillations and synchronization.

EEG recordings are normally used to investigate these neural oscillations and some neurons can generate action potentials or spikes in a rhythmic pattern with particular frequencies. Spiking patterns that are the result of bursting are considered fundamental for information coding. In many neurological disorders, the cause is excessive neural oscillation (KAMEL; MALIK, 2015).

The human brain waves normally are classified in five types by their frequency ranges. These types are known as alpha (α), theta (θ), beta (β), delta (δ), gamma (γ), and mu (μ) that are described below, according Kamel and Malik (2015):

a) Alpha waves have a frequency range of 8-13 Hz. These waves were discovered by Dr. Hans Berger in 1908. Because these waves were the first to be discovered, they are called alpha waves (first waves). Alpha waves are associated with wakefulness, closing the eye, effortless alertness, and cre-ativity. These waves normally appear in the posterior half of the head and have higher amplitude over the occipital areas;
b) Beta waves have a frequency range of 14-26 Hz. Found only in normal adults, these waves are correlated with active attention, active thinking, solv-ing critical problems, or focusing on the outside world and, therefore, are also known as sensory motor rhythm. Rhythmical beta waves are experi-enced mainly in the frontal and central regions. Beta waves are low in ampli-tude and are normally under 30 μV;

c) Gamma waves have a frequency range above 30 Hz (up to 100 Hz). These waves help to determine the binding of different populations of neurons together. They occur rarely in the human brain. They occur only during crossmodal sensory processing, that is, the process of combining different senses such as sound and sight;

d) Delta waves have a frequency range of 0.5-4.0 Hz. These waves have the highest amplitude among the other waves but have the lowest frequency. These occur frontally in adults and posteriorly in children. These waves are primarily associated with deep sleep. These delta rhythms may also be associated with subcortical lesions, deep midline lesions, or meta- bolic encephalopathy hydrocephalus; and

e) Mu wave have a frequency range of 8 to 13 Hz. These waves are mixed with other waves and sometimes partly overlap other rhythms. It shows the synchronous firing of motor neurons over the sensorimotor cortex.

The different brain rhythms and waves are shown in Figure 6.

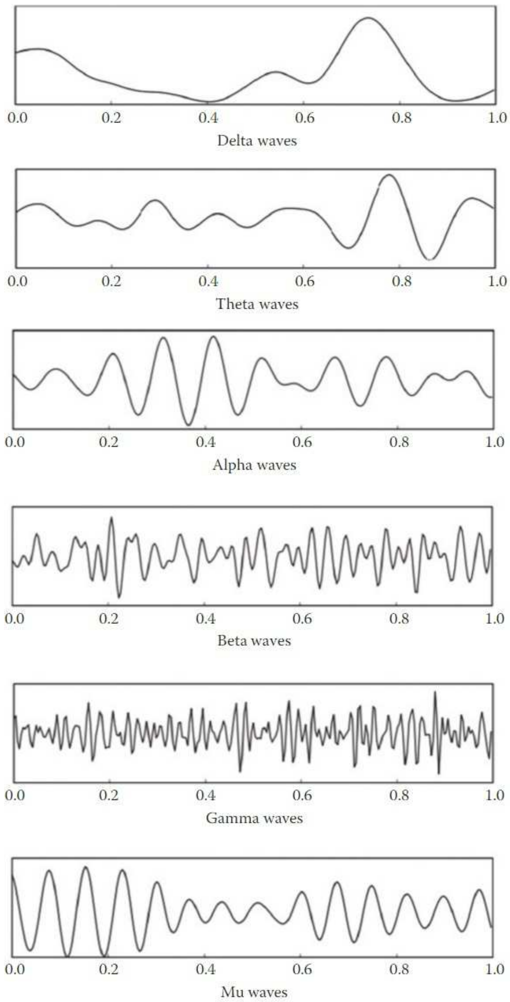


Figure 6 - Different brain rhytms and waves (KAMEL; MALIK, 2015).

In order to understand how an EEG signal is collected it is necessary to see how the wave brain signals are propagated form the brain up to the electrodes. Gesualdi (2011) indicates that bioelectricity in the cortex is propagated throughout the brain. Before it reaches the more external parts of the brain where the electrodes may sense it, it encounters propagation barriers, such as non-neuronal tissues (meninges and bone) and the cerebral spinal fluid, which absorb electricity, diminish the strength and deviate the signal. In figure 7, an cut view from the left parietal lobe is shown where it is possible to see these tissues in details.

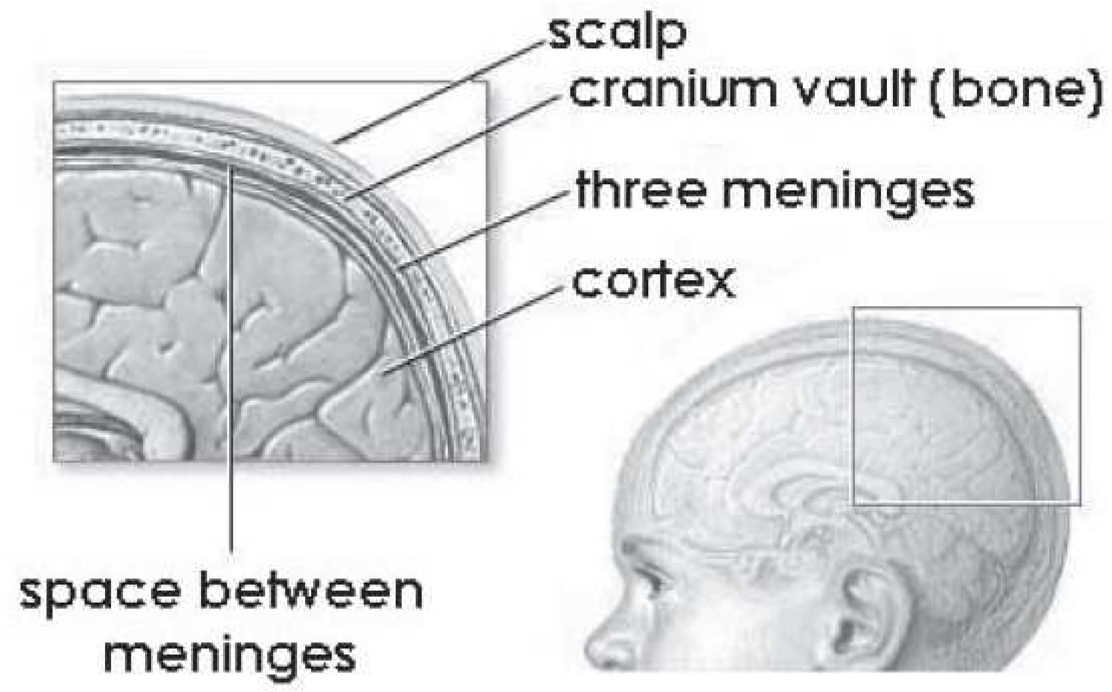


Figure 7 - Cut view of the the left parietal lobe (GESUALDI, 2011)

Thus, when the electrical signals (called spikes) reaches the cranium vault, the waves that can be sensed by the electrodes on the scalp are attenuated and have to be amplified in order to be studied. The waves sensed at the scalp have poor spatial resolution, in the order of centimeters, so it is impossible to correlate the position of the electrode on the scalp with the precise spot in the brain, where the electrical signal was originated. Thus, although there is a very homogeneous distribution of electrodes, space precision cannot be reached (GESUALDI, 2011).

The ERPs uses an EEG procedure that measures electrical activity of the brain over time using electrodes placed on the scalp. The EEG reflects thousands of simultaneously ongoing brain processes in specific points distributed in specific areas and Regions of Interest (ROI) of the scalp depending of the cognitive target of the research. These ROI are related more with the cognitive analisys process than the data collection that is related with the individual electrodes. The brain responses to a single stimulus or event of interest is not usually visible in the EEG recording of a single trial.

Therefore, to detect the brain's response to a stimulus, the experiment shall conduct several trials and make an average of the results together. Random brain activity and other noises are filtered out by this process, allowing to extract out the relevant waveform, called the ERP related to the stimulus. As mentioned by Handy (2005), the ERP technique is one of the most widely used methods in cognitive neuroscience research to study the physiological correlates of sensory, perceptual and cognitive activity associated with processing information with a low cost with relatively simple technology.

The ERP experiment simplified can be as shown in figure 8:

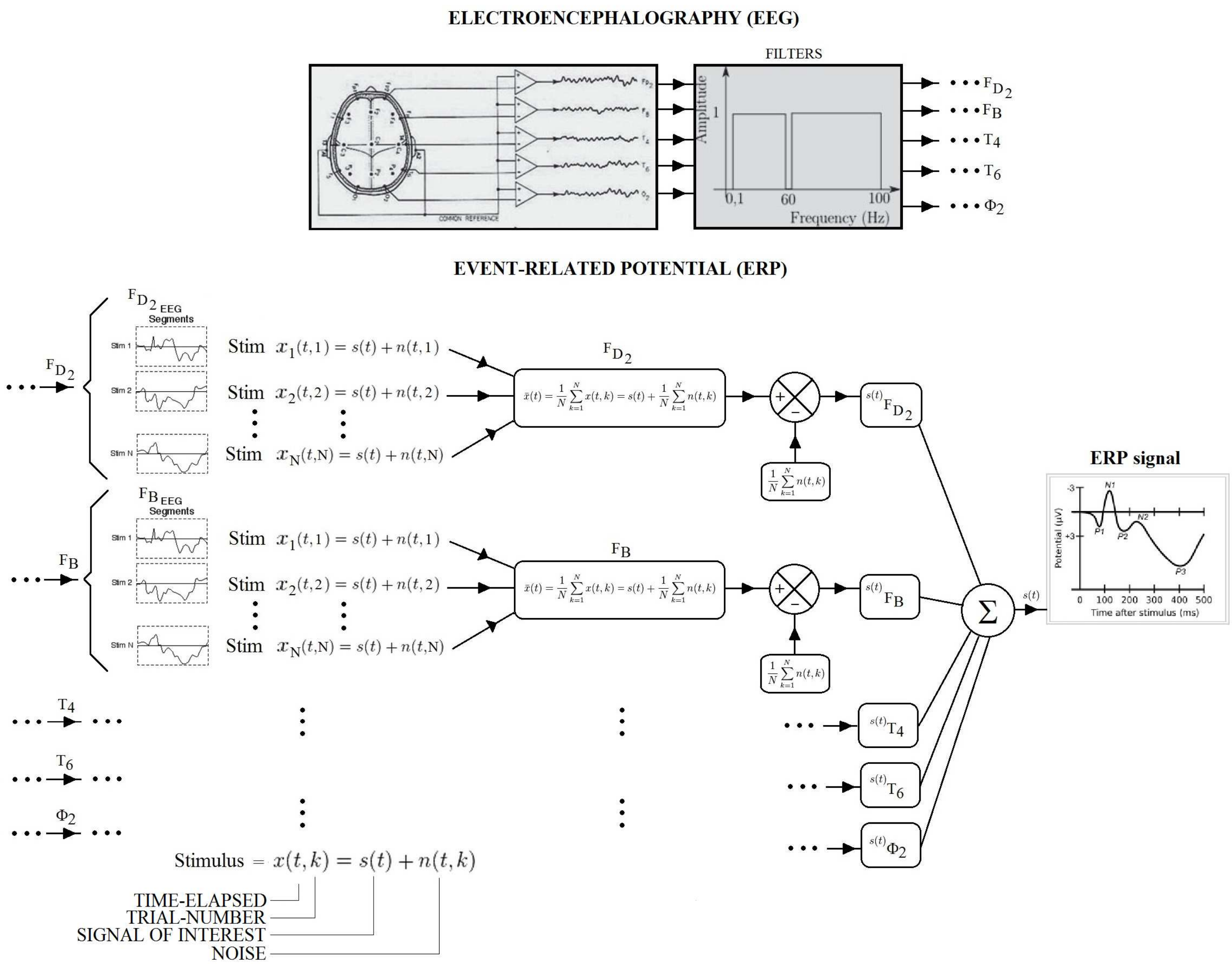


Figure 8 - Simplified schematics for ERP experiment (GESUALDI, 2011)

As can be seen in figure 8, the ERP technique consists on a signal amplification that adds up and averages specifically time-locked epochs, which are replications of a stimulus, and ideally can present a lower signal-to-noise ratio (SNR) than those of the original waveforms. The signal to noise ratio (SNR) is a quality measure to sinal processing. SNR is the ratio between the signal power and the noise power (GESUALDI, 2011).

ERP signals resulted from the experiment present a series of positive and negative voltage deflections, which are related to a set of underlying components called ERP components. The usual ERP components are referred to by a letter (N/P) indicating polarity (negative/positive), followed by a number indicating either the latency in milliseconds or the component's ordinal position in the waveform (LUCK, 2014). For example, a negative-going peak that is the first substantial peak in the waveform and often occurs about 100 milliseconds after a stimulus is presented is often called the N100 (indicating its latency is 100 ms after the stimulus and that it is negative) or N1 (indicating that it is the first peak and is negative); it is often followed by a positive peak, usually called the P200 or P2. The stated latencies for ERP components are often quite variable. For example, the P300 component may exhibit a peak anywhere between 250ms - 700ms (WIKIPEDIA, 2016). In figure 9 below, an example of an ERP waveform signal with these components can be seen.

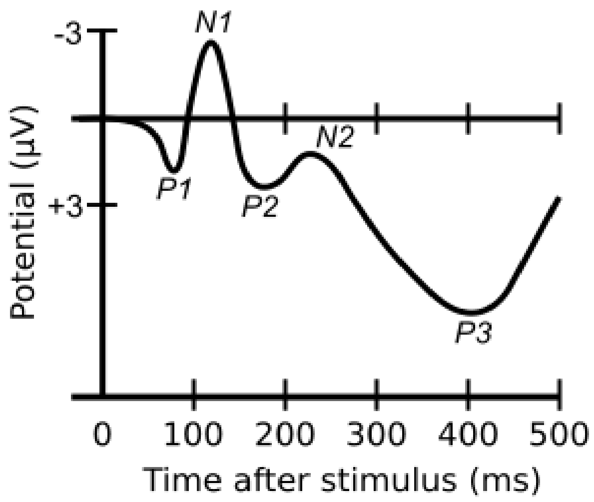


Figure 9 - A fictitious liustrative waveform graph example showing several ERP components as P1 (P100), N1(N100), P2(P200), N2 (N200) and P3 (P300) (WIKIPEDIA, 2016)

As described by Woodman (2010), an ERP component can be simply defined as one of the component waves of the more complex ERP waveform. ERP components are defined by their polarity (positive or negativegoing voltage), timing, scalp distribution, and sensitivity to task manipulations. Different ERP component nomenclatures emphasize different aspects of these defining features and to provide a jumping off point for literature reviews. In the Table 1 are exemplified several of ERP components.

Table 1 - Summary of ERP components using a variety of nomenclatures during a simple visual-manual task. This list focuses on visual components and neglects components from the auditory, language, and memory literatures. Abbreviations: CNV, Contingent Negative Variation; O- & E-waves, Orienting & Expectancy Waves; C1, component 1; N, negative; P, positive; N2pc, N2-posterior-contralateral; PCN, Posterior Contralateral Negativity; CDA, contralateral-delay activity; SPCN, Sustained Posterior Contralateral Negativity; LRP, Lateralized Readiness Potential; ERN/Ne, Error-Related Negativity/Error Negativity; Pe, Error Positivity (WOODMAN, 2010)

Nomenclature	Ordinal	Latency (peak)	Scalp Distribution	Task/Stimulus Specificity	Hypothesized Process(es) Indexed	Useful Reference
Components Preceding a Stimulus				CNV (O- & E-waves)	Anticipation, Cognitive & Motor Preparation	(Brunia, van Boxtel, & Bocker, in press)
Components Following a Stimulus	C1	P/N50-70			Sensory Processing	(Pratt, in press)
	P1	P90-100			Sensory/Perceptual Processing	(Pratt, in press)
	N1	N170-200	Posterior Versus Anterior N1	N170 for faces	Perceptual Processing, Expert Recognition, Visual Discrimination	(Hillyard, Vogel, & Luck, 1998; Rossion & Jacques, in press; Vogel & Luck, 2000)
	P2				Not Well Understood	(Crowley & Colrain, 2004)
	N2	N225-250			Object Recognition, Categorization	(Folstein & Van Petten, 2008; Pritchard et al., 1991)
	N2pc		PCN		Deployment of Covert Attention	(Luck, in press)
	P3	P300	P3a/P3b	P3a/P3b	Stimulus Evaluation Time, Categorization, Context (Working Memory) Updating, Cognitive Load	(Polich, in press)
			SPCN	CDA	Maintenance in Visual Working Memory	(Perez & Vogel, in press)
				LRP	Response Preparation	(Smulders & Miller, in press)
Components Following a Response			Medial Frontal Negativity	ERN/Ne & FBN	Error Processing, Reinforcement Learning or Response Conflict Signal	(Gehring, Liu, Orr, & Carp, in press)
				Pe	Affective or Conscious Assessment of Task Performance	(Falkenstein et al., 2000)

Soto (2014) explains that the ERP methodology was developed seriously in 1964, and 1965, where the first cognitive ERP components were discovered. The ERP components were identified as Contingent Negative Variation (CNV) which signaled the subjects' preparation for a button pressing task, and the P3, a component which reflects subjects' expectation of a certain stimulus (small when predictable, large when not) and are thus highly replicable amplitude modifications. Soto (2014) mentions:

> For instance, the P300 is a positive voltage peak, with specific temporal characteristics always approximately 300ms post stimulus. Studies tend to either investigate the underlying cognitive processes the component might reflect, dubbed ERPology by Luck (2014) or component responses are manipulated as a tool to address some other cognitive question. For example, the P300 could be used for comparing stimuli of different modalities to see if different modalities affect the response, for example, by delaying it or decreasing its amplitude, such that inferences about their processing can be made (e.g. a delayed P300 might indicate 'harder' to process). The latter approach gained credibility especially from the mid. 80s onwards as access to the equipment required for the ERP collecting became more widely available, an exciting new discoveries were reported on, such as the N400.

Concerning the data collection in Soto (2014), the fact that Excitatory Postsynaptic Potential (EPSP) generated current can be measured by an electrode at the scalp at a determined distance from its generator is due of volume conduction. This phenomenon causes the brain tissues to function as a conductor for the current until it reaches the surface of the scalp. The current of individual neurons as a result of potential difference between postsynaptic (apical) dendrites and the cell body is summed, such that the negative and positive voltage values of their summed activity can be recorded by electrodes during EEG sessions (SOTO, 2014).

The EEG signal is recorded as a continuous signal, and stimulus presentation is marked for onset, and is difficult to be detected and marked. The raw signal is usually filtered for low frequencies (e.g. high pass of 0.01Hz) and amplified. Afterthat, a computer or a separately connected trigger box marks a digital code or a pulse width on the recorded signal, allowing the marking on the continuous EEG signal of the exact onset of the stimulus, and type of stimulus shown. An illustrative example of an experiment is presented schematically in Figure 10: subjects saw many "X"s, sparsely alternated by "O"s. The fragments, called epochs, related to the event are averaged for each electrode so as to amplify the response and filter out noise coming from other neurophysiologic activity or interference of electrical equipment. These averaged responses, the ERPs, can now be compared and characterized in terms of amplitude (in μV) - the peak of the wave - and latency (in ms) - the time in which the wave peaks (SOTO, 2014).

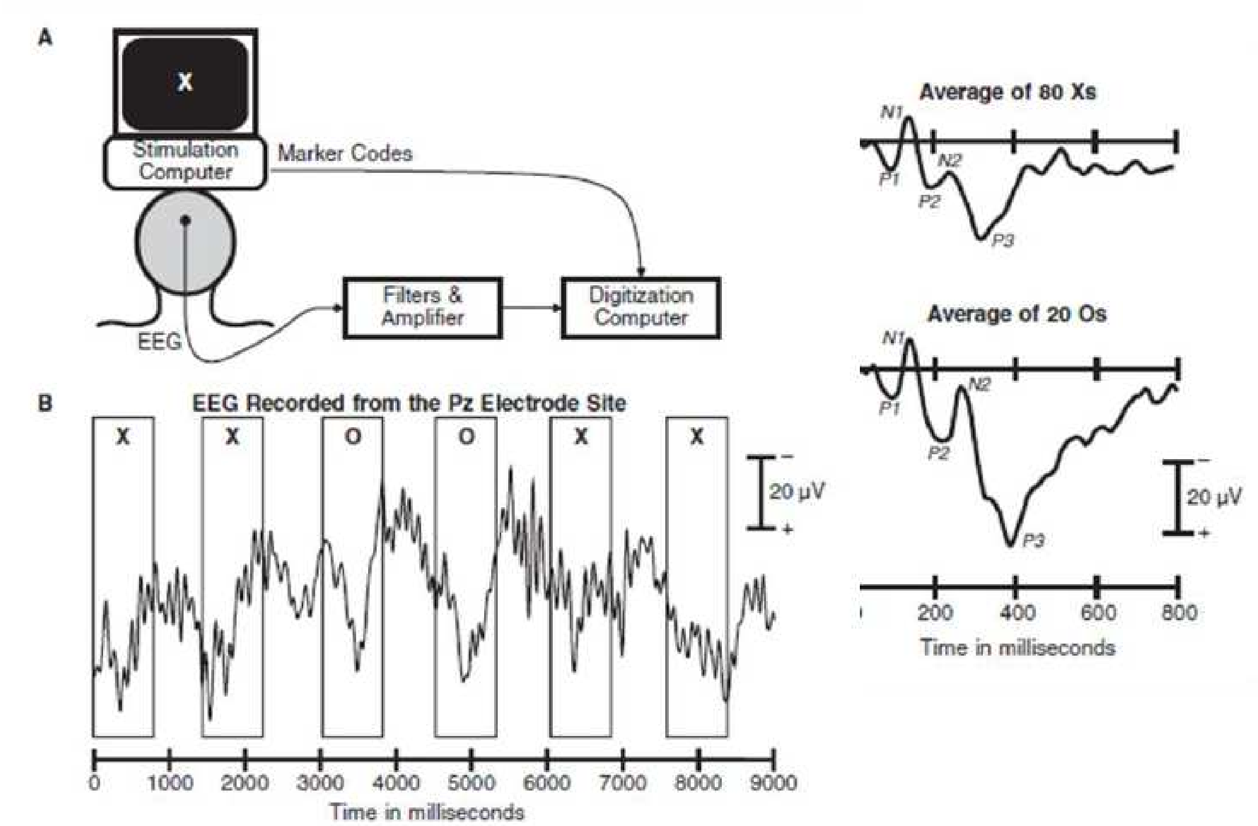


Figure 10 - Example of EEG/ERP experiment with epochs “X” and “O” (SOTO, 2014)

Concerning the target of this work (language specific properties), Soto (2014) indicates that indeed ERP methodologies have brought much evidence to show that very detailed linguistic information has an immediate effect on processing streams. For instance, manipulations of word category information affect ERP signals at 40-90ms after word onset, and 120-150ms if there is a category violation. Soto (2014) also indicate that the N400 component of the ERP signal can also be influencied by strict linguistic variables.

For this study, three ERP parameters that were extracted from the Soto (2014) experiment. These parameters were:

a) Mean Amplitude Between two fixed latencies, in microvolts;
b) Peak Amplitude, in microvolts, and
c) Peak Latency, in miliseconds.

In linguistic experiments, the values of these parameters are influenced and correlated by the type of stimulus applied to the subjects (SOTO, 2014). Due to this, the expectation for this work is using these parameters in Soto’s (2014) experiment as features in order to try to achieve good results for the classifiers.

II.2 EEG/ERP Data Software toolboxes and Matlab® platform

To calculate and organize the ERP data from an experiment there are several softwares that help in this data mining activity. Concerning this work, EEGLAB® and the ERPLAB®, which are Matlab® toolboxes for processing and analyzing EEG and ERP data were used, and for the digital process and pattern recognition study, Matlab's® Patter Recognition toolbox was used.

EEGLAB® is an interactive Matlab® free toolbox for processing continuous and event-related EEG, Magnetoencephalography (MEG) and other electrophysiological data using independent component analysis (ICA), time/frequency analysis, and other methods. The use of EEGLAB® helped to organize the EEG data from the experiment of Soto (2014) to be treated by the ERPLAB® toolbox. This software was developed by the Swartz Center for Computational Neuroscience of the University of California San Diego (EEGLAB® Tutorial site, consulted on April, 15th, 2016).

In turn, ERPLAB® Toolbox is also a free, open-source Matlab® package for analyzing ERP data. It is tightly integrated with EEGLAB® Toolbox, extending EEGLAB®'s capabilities to provide robust, industrial-strength tools for ERP processing, visualization, and analysis. A graphical user interface makes it easy for beginners to learn. This software was developed by Javier Lopez-Calderon and Steven J. Luck from the Center for Mind and Brain, University of California-Davis, Davis, CA, USA (LOPEZ-CALDERÓN; LUCK, 2014) .

An example of the Graphical Interfaces of the EEGLAB® and ERPLAB® toolboxes can be seen in figure 11 below.

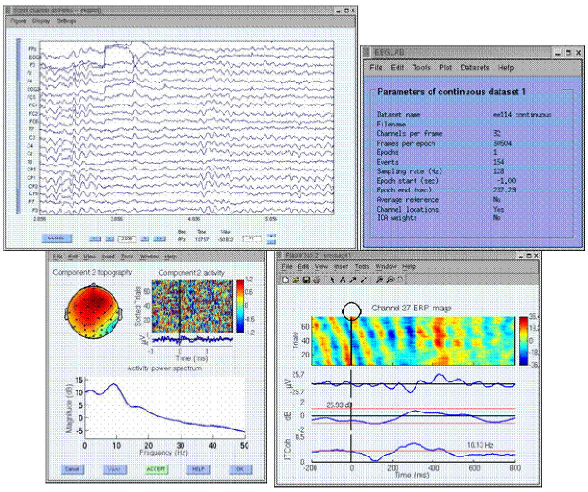


Figure 11 - Example of Graphical Interfaces of the EEGLAB® and ERPLAB® toolboxes (EEGLAB®, 2016) and (ERPLAB®, 2016)

The software Matlab® is a well-known and powerful computing environment developed by MathWorks, which allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python (MATLAB®, 2016a.).

The use of EEGLAB® and ERPLAB® toolboxes associated with Matlab® scripting provides enormous power for intermediate and advanced users.

II.3 Soto's (2014) ERP Experiment

In order to investigate the specific nature of the N400 effects in sentence and word pair contexts, Soto (2014) proposed a sentence and word priming task experiments. An important observation considered in the experiment are the use of Stimulus Onset Asynchrony (SOA), which is the time between the onset of the first stimulus until the onset of the second (following) stimulus. SOA comprises of presentation time plus interval time. In general, it is assumed that if SOAs are short, faster automatic processing can be captured, while longer SOAs capture higher level processing, which packs in more processing information within the longer course of processing. Given that Soto (2014) was interested in capturing as much automaticity in processing as possible, the experiment endeavored to keep the shortest possible SOA value throughout the development of the pilot studies.

II.3.1 Soto's (2014) Materials and Methods

For the sentence task, the experiment investigated the level of information accessible during processing in the 0 to 700ms time window after target onset. The target word was the verb complement. For instance; in (…) *"João dirije a moto"* (….) ("João rides the motorbike"), the brain potential related with motorbike was assessed. And for the word task, the mechanisms for establishing semantic relationships between word pairs via a priming paradigm were examined, either pairs were related semantically by mere association or pairs could additionally be linked semantically by syntactic structuring (SOTO, 2014). Target words were repeated in all conditions so as to increase comparability, i.e. also for the word task, the brain potential related with motorbike in a priming pair was assessed (e.g. CAPACETE moto, "HELMET motorbike"), because it was also the target in the sentence task (SOTO, 2014).

Following from the proposed variables (SOTO, 2014), 4 conditions and one control condition were established for the sentence task: (i) congruous supportive-context (CSC): e.g. "Até sem capacete, João dirige a moto feito louco"; (ii) congruous non-supportive context

(CNSC), e.g. “Todos os dias, João dirige a moto feito louco”; (iii) incongruous supportive-context (ISC): e.g. “Até sem capacete, João dirige a pera feito louco”; and (iv) incongruous non-supportive context (INSC), e.g. “Todos os dias, João dirige a pera feito louco”. For the word pair task (SOTO, 2014), 3 conditions and one control condition were established: (i) associative semantic relation (ASR): e.g.”ÔNIBUS moto” (“BUS motorbike”); (ii) syntactic and semantic relation (SSR): e.g. “CAPACETE moto” (“HELMET motorbike”); (iii) unrelated pair (UR) “FACA nuvem” (“KNIFE cloud”) ; and control 2: a pair with pseudo word (PW) target: e.g. “CARRO garufa” (“CAR garufa”).

For the sentence task, each subject saw 240 sentences: 120 distractors, 30 sentences with and 30 without supporting context, and 60 incongruous sentences of which 30 with and 30 without supporting context. 60 of the distractor sentences were incongruous, to balance YES/NO answers for the incongruence judgment task. After the sentence task, subjects saw 150 word pairs, of which 30 semantically related, 30 syntactically and semantically related, 30 unrelated words and 60 non-words. None of the words were seen more than once by the subjects. Four versions were compiled in which target words were repeated for all conditions except for incongruous sentences and unrelated word pairs (Table 2).

Table 2 - Experimental conditions and sample stimuli for the ERP experiment (SOTO,2014)

<table>
<tr><td colspan="5">Sentence Task</td></tr>
<tr><td>Condition</td><td>context</td><td>congruence</td><td colspan="1">Stimulus example (n=30 for each condition)</td><td>Repeated item</td></tr>
<tr><td>1: CSC</td><td>supportive</td><td>congruous</td><td>Até sem capacete, João dirige ↑ a moto feito louco</td><td>dirige a moto</td></tr>
<tr><td>2: CNSC</td><td>non-supportive</td><td>congruous</td><td>Todos os dias, João dirige ↑ a moto feito louco</td><td>dirige a moto</td></tr>
<tr><td>3: ISC</td><td>supportive</td><td>incongruous</td><td>Até sem capacete, João dirige ↑ a pera feito louco</td><td>dirige -</td></tr>
<tr><td>4: INSC</td><td>non-supportive</td><td>incongruous</td><td>Todos os dias, João dirige ↑ a pera feito louco</td><td>dirige -</td></tr>
<tr><td>Control 1:</td><td>-</td><td>incongruous</td><td>distractor sentences</td><td>-</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="2">Condition</td><td rowspan="2">relation</td><td colspan="2">Stimulus example (n=30 for each condition)</td><td rowspan="2">Repeated item</td></tr>
<tr><td>Prime</td><td>Target</td></tr>
<tr><td>1: SSR</td><td>Syntactic and Semantic</td><td>CAPACETE</td><td>moto</td><td>moto</td></tr>
<tr><td>2: ASR</td><td>Associative Semantic</td><td>ÔNIBUS</td><td>moto</td><td>moto</td></tr>
<tr><td>Control 1: UR</td><td>Unrelated Words</td><td>FACA</td><td>nuvem</td><td>-</td></tr>
<tr><td>Control 2: PW</td><td>Pseudo Word Target</td><td>FILTRO</td><td>garufa</td><td>-</td></tr>
<tr><td colspan="5">Abbreviations: congruous supportive-context (CSC); congruous non-supportive context (CNSC); incongruous supportive-context (ISC); incongruous non-supportive context (INSC); associative semantic relation (ASR); syntactic and semantic relation (SSR); unrelated pair (UR); pair with pseudo word target (PW)</td></tr>
</table>

The arrows in the Table 2 indicate the exactly of extraction of the signal cutout concerning each epoch. It is important to highlignht that each epoch is a fragment of the EEG signal related with each kind of differtent linguistic condition, this is, the classes for the classifiers.

Participants first saw four blocks of sentences (4x60, a total of 240 sentences) in pseudo randomized order; in between blocks there was an interval, the duration of which was determined by the participant. After that, 150 word pairs were presented in pseudo randomized order, in 3 blocks of 50 with 2 intervals. The presentation was programmed and presented with Eprime software, version 2 (developed by Psychology Software Tools, Inc.). Sentences were presented segmented into inseparable linguistic constituents (1-3 words) and word pairs were presented word by word (except for the last segment, which could contain up to 2 words) on a 19 inch screen in white 25 pts times new roman font. The presentation rates can be seen in Table 3 below:

Table 3 - Presentation protocol and SOA for the ERP Experiment (SOTO,2014)

Presentation protocol ERP experiment: sentence task										
Presented:	+	Até sem capacete,	(blank)	João	(...)	a moto	(blank)	feito louco	(blank)	RESPONDA
Action:						*Target*				*Congruent Y or N?*
Timing: (ms)	1500	300	100	250	(...)	250	100	250	350	1500

Presented:	+	(blank)	CAPACETE	(blank)	moto	(blank)	muito veloz	(blank)	RESPONDA
Action:			*Prime*		*Target*				*Lexical Descision Y or N?*
Timing: (ms)	1500	100	250	100	250	100	250	350	1500

II.3.2 Soto's (2014) Procedure

Concerning the experimental setup, 21 university students participated in the study (female =11), distributed evenly over 4 versions, average age 22 years old, all right-handed, with normal or corrected-to-normal vision Participants' judgments were recorded by pressing with one of two fingers of the right hand either a red or a green button on a button box. The position of the green and red buttons, destined for YES and NO responses, was swapped for each participant. Figure 12 shows this setup:

Figure 12 - Electrode set up during recording (SOTO,2014)

The scalp Regions of Interest (ROI) selected by Soto (2014) and the respectively electrode channels identification for along the mid-line were: Frontal (F1-ch34, F2-ch60, FC1-ch7, FC2-ch27, FCz-ch38 and Fz-ch2); Central (C1-ch39, C2-ch56, CP1-ch11 CP2-ch21, CPz-ch52 and Cz-ch22), Parietal (CP1-ch11, CP2-ch21, CPz-ch52, P1-ch43, P2-ch51, and Pz-ch12), and Occipital (O1-ch15, O2-ch17, Oz-ch16, PO3-ch46, PO4-ch48, and POz-ch47). On the left hemisphere, they were Frontal (F3-ch3, F5-ch35, F7-ch4, FC3-37, FC5-ch6 and FT7-ch36); Central (C3-ch8, C5-ch40, CP3-ch42, CP5-ch10, T7-ch9 and TP7-ch41), Parietal (CP3-ch42, CP5-ch10, P3-ch13, P5-ch40, P7-14 and TP7-ch41), and Occipital (P3-ch13, P5-ch44, P7-ch14, PO3-ch46 and PO7-ch45). And on the right hemisphere, they were: Frontal (F4-ch28, F6-ch59, F8-ch29, FC4-ch57, FC6-ch26 and FT8-ch58); Central (C4-ch23, C6-ch55, CP4-ch53, CP6-ch20, T8-ch24 and TP8-ch54), Parietal (CP4-ch53, CP6-ch20, P4-ch18, P6-ch50, P8-ch19 and TP8-ch54), and Occipital (P4-ch18, P6-ch50, P8-ch19, PO4-ch48 and PO8-ch49).

In Figure 13 all scalp ROI based on anatomical proximity are presented.

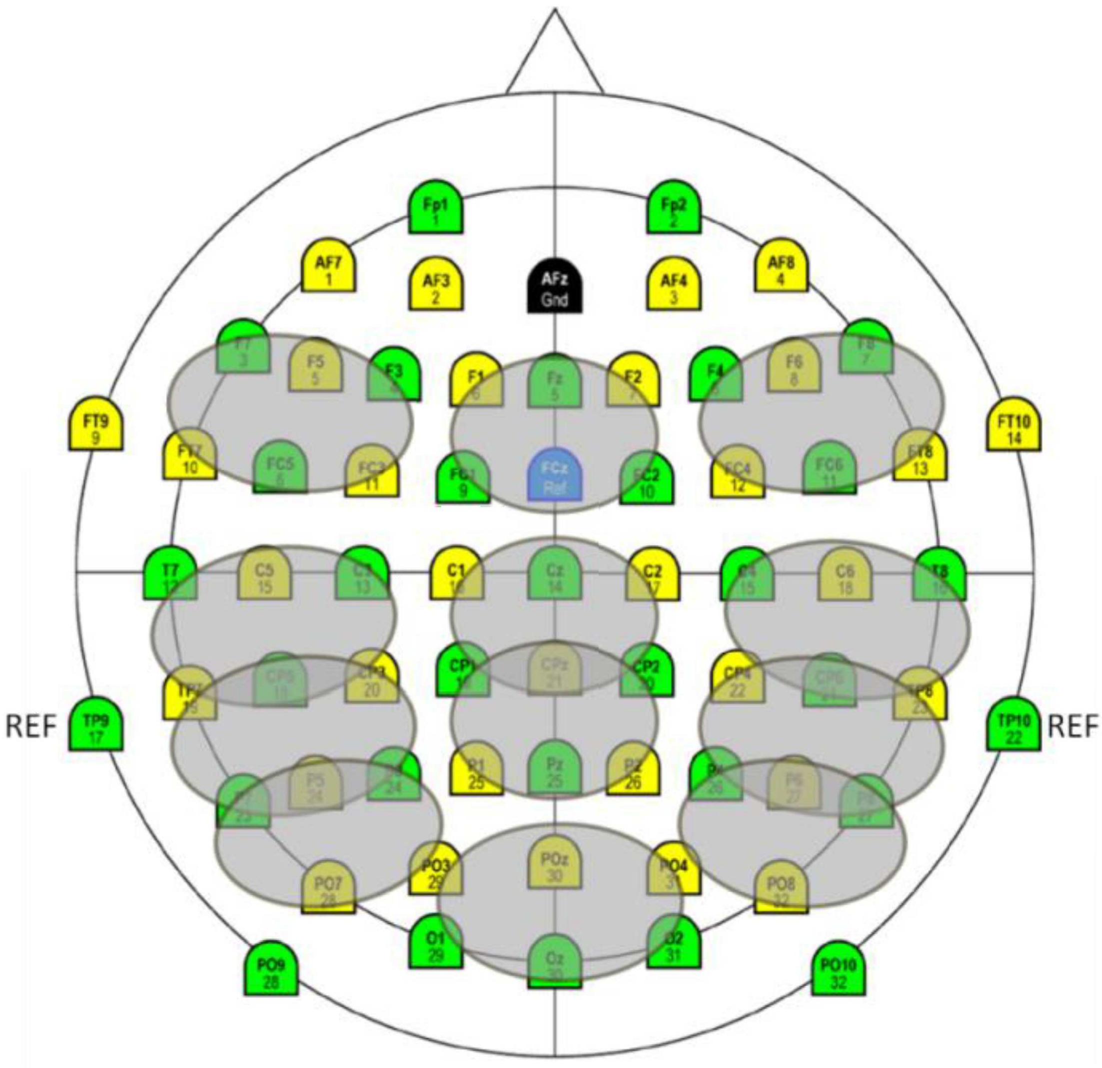


Figure 13 - ROI definition as based on anatomical proximity (SOTO,2014)

For EEG recording, the active electrodes were connected to the ActiCHamp equipment (developed by BrainProducts) which amplifies and digitalizes the analogue signal captured by the electrodes. From the computer that handles the stimulus presentation, the presentation program sends pulses via the parallel port, marking stimulus onset for posterior segmenting of the continuous EEG signal. These pulses (also named triggers) are input for the amplifier and simultaneously recorded on a separate channel along with the other 64 electrode channels (in fact 62 signal electrode channels and 2 for EEG reference on-line to left and right mastoid channels). During the acquisition phase, the signal was filtered with a 100Hz low-pass and 0,01Hz high-pass filter. Data were digitalized at a 500Hz sample frequency by a 24-bit analog-to-digital converter.

The ActiCHamp device was connected to another computer with a software to conFigure recording parameters (sample frequency, reference, etc.), which records and saves the digital signal for later processing by the program Pycorder, an open source acquisition program.

The pulses sent via parallel port were converted into strings (S1, S2, etc.) depending on experimental condition of the stimulus shown (classes for the pattern recognition approach). This enabled the program to separate the continuous EEG signal into epochs, segments of 1200ms, starting 200ms before and ending 1000ms after trigger markers. During segmenting, baseline correction was carried out (-200 to 0 ms).

After these steps, the segmented signals were visually inspected for artifacts, such as eyeblinking, frowning, swallowing and other distorted signals (due to loose electrodes, crosstalking between electrodes, etc.).

II.3.3 Soto's (2014) data acquisition and organization

Finally, the signal was filtered with a low pass Butterworth filter of 30Hz. Now the signal is ready to be averaged separately for each condition as well as subject so that conditions can be treated by the software tools.

The files obtained for each subject are:

a) the EEG raw data file recordings, in format ".eeg";
b) the Epochs markers datafile , in format ".vmrk"; and
c) the header data file, in format ".vhdr".

For instance, for the subject 1, for the word task, the following experimental files were obtained: subj1_palavras.eeg, subj1_palavras.vmrk and subj1_palavras.vhdr.

In other hand, for the sentence task, the following experimental files were obtained: subj1_sentences.eeg, subj1_sentences.vmrk and subj1_sentences.vhdr.

After the use of the EEGLAB® and ERPLAB®, described on the Appendixes B and C, the following database was extracted:

a) For Sentences Task: a matrix with 2880 lines and 7 columns, where the columns corresponding to the following features respectively: Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency, Region of Interest (ROI), ERP Time Range and Subject index. The last column corresponding to the classes.

b) For Words Task: a matrix with 2304 lines and 7 columns, where the columns corresponding to the following features respectively: Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency, Region of Interest (ROI), ERP Time Range and Subject index. The last column corresponding to the classes.

II.4 Pattern Recognition

Pattern recognition is an interdisciplinary subject, covering developments in the areas of statistics, engineering, artificial intelligence, computer science, psychology and physiology, among others. As described by Bishop (2006), pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, and by Theodoridys and Koutrombas (2009), pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes.

Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning or discrimination), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning or clustering). Webb (2002) defines that in supervised classification a set of data samples (each consisting of measurements on a set of variables or attributes or features that can be extracted) are associated with which correspond to the class types. These classes and features are used in the classifier design. In unsupervised classification, the data labels (classes) are not known and it is necessary to seek for groups in the data with the same characteristics, by the features that can distinguish one group (class) from another.

As described by Webb (2002), the data can undergo several separate transformation stages before the outcome is reached. These transformations, as termed preprocessing, feature selection or feature extraction, operate on the data in a way that the objective is to reduce the dimension of the number of features involved. This involves the remotion of redundant or irrelevant information, and the transformation of the data in an appropriate form for the classifier. This can determinate effectively the kind of classes of an input. An oversimplified procedure of pattern recognition is shown in figure 14 with the roles of all data experiment origin and software tools used in this work.

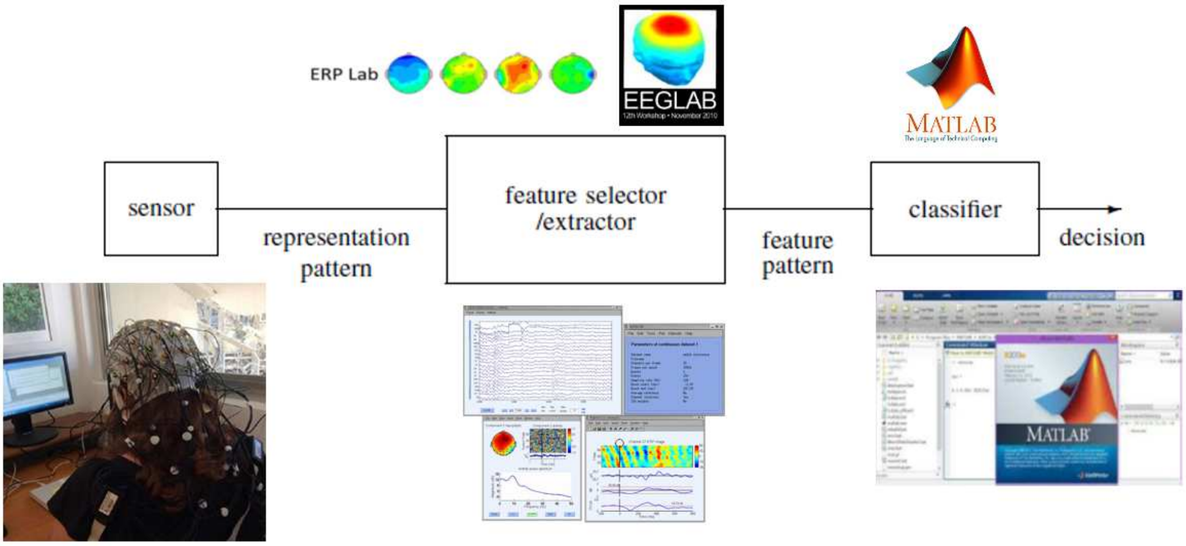


Figure 14 - Pattern Recognition Method and roles of each software tool used (WEBB, 2002)

As shown in figure 14, in this study, the Soto (2014) experiment is related with the Sensor block (in fact, the EEG experiment) and the feature selector/extractor were implemented with the softwares EEGLAB® and ERPLAB®. The classifier box was done with the software Matlab®.

The methodology considered in this work is based on the stages of a pattern recognition problem indicated by Webb (2002). These stages are:

a) Formulation of the problem - it is the origin of the study. Specifiy the boundary conditions and premises in order to “gaining a clear understanding of the aims of the investigation and planning the remaining stages” (WEBB, 2002);

b) Data collection - it is the execution of the experiment, in order to “making measurements on appropriate variables and recording details of the data collection procedure (ground truth)” as defined by Webb (2002);

c) Initial examination of the data - it is the verification of the results obtained with the experiment, in order to “checking the data, calculating summary statistics and producing plots in order to get a feel for the structure” (WEBB, 2002);

d) Feature selection or feature extraction - it is the stage to identify which parameters can contribute decisively in the classification campaign, in order to “selecting variables from the measured set that are appropriate for the task” (WEBB, 2002). These new parameters can be obtained by feature extraction, that is a linear or nonlinear transformation of the original set as described by Webb (2002);

e) Unsupervised pattern classification or clustering - it is the classification scenario where the dataset do not contains the identification of the classes and can be considered “as exploratory data analysis and it may provide a successful conclusion to a study and, on the other hand, it may be a means of preprocessing the data for a supervised classification procedure" (WEBB, 2002);

f) Apply discrimination or regression procedures as appropriate - it is the classification scenario where the dataset contains clearly the identification of the classes and "the classifier is designed using a training set of exemplar patterns", as decribed by Webb (2002);

g) Assessment of results - it is the stage where after the use of the classification, the results are assesses by means of the accuracy or the use of figures of merit as a CVonfusion Matrix or Receiver Operating Characteristic (ROC), that will be explaine later. As Webb (2202) explain, "this may involve applying the trained classifier to an independent test set of labelled patterns"; and

h) Interpretation - it is the final stage where the reasons of the results obtained can be explained and well understood.

Figure 15 shown this procedure in a flowchart way.

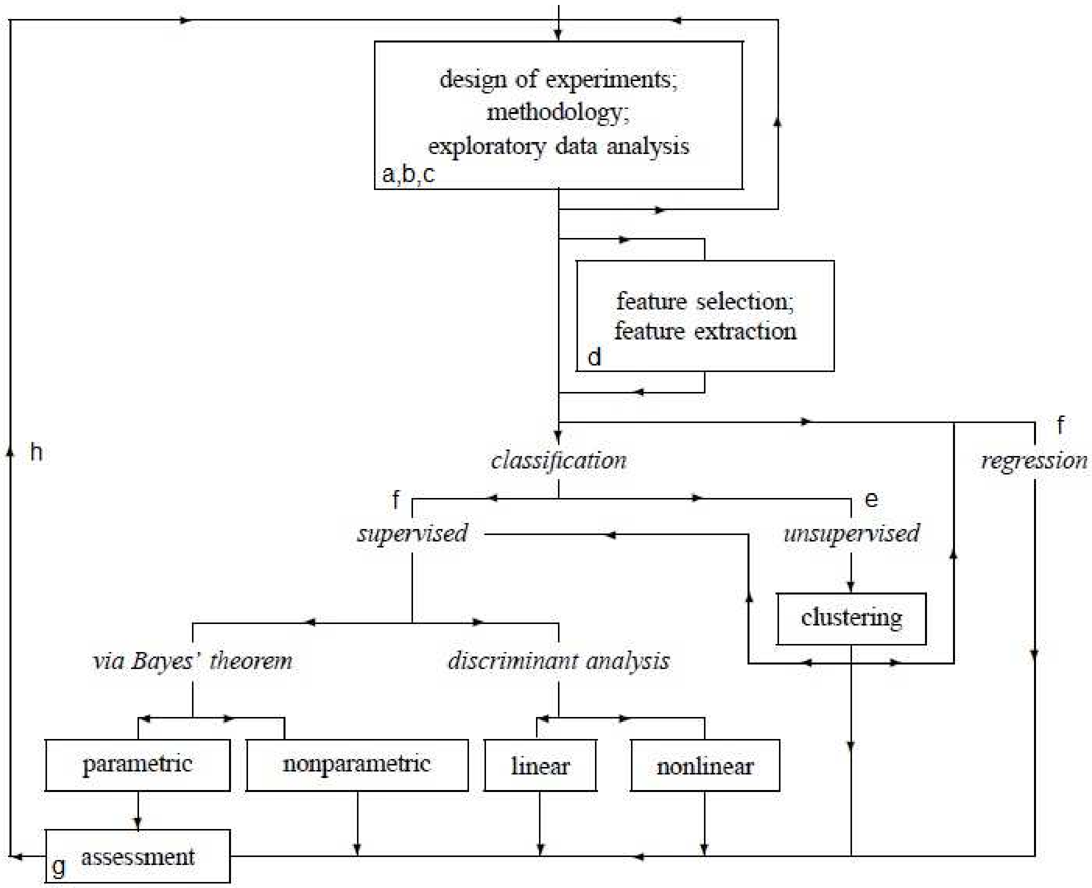


Figure 15 - Pattern Recognition methodology (WEBB, 2002)

As explained by Webb (2002), these steps are an interative process, where the analysis of the results may pose further hypotheses that require further data collection. On the other hand, the questions posed may be answered by an initial examination of the data or it may be concluded that the data cannot answer the initial question and the problem must be reformulated.

Concerning the performance of classifiers is an important aspect of the pattern recognition cycle, because it is the measurement of how good the classifier is in relation to other alternative classification techniques. In this study, the method used to estimate the performance of classifiers was the confusion matrix.

The confusion matrix is a specific table layout that allows visualization of the performance of an algorithm as described by Stehman (1997). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice versa).

For example, in figure 16 shows the confusion matrix for a two class classifier.

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Figure 16 - Example of Confusion Matrix for 2 classes

The entries in the confusion matrix have the following meaning in the context of our study are; “a” is the number of correct predictions that an instance is negative; “b” is the number of incorrect predictions that an instance is positive; “c” is the number of incorrect of predictions that an instance negative, and “d” is the number of correct predictions that an instance is positive.

The accuracy (AC) is the proportion of the total number of predictions that were correct. In the example of figure 17 is determined using the equation:

$$AC = \frac{a+d}{a+b+c+d} \tag{2.1}$$

Other important procedure to consider in pattern recognition performance for classifiers is the dataset split. This kind of approach gives a better accuracy in the error results obtained by a classifier. The more usual method (and used in this work) is known as “houldout estimation”:

The holdout method splits the data into two mutually exclusive sets,

> sometimes referred to as the training and test sets. The classifier is designed using the training set and performance evaluated on the independent test set. The method makes inefficient use of the data (using only part of it to train the classifier) and gives a pessimistically biased error estimate.. However, it is possible to obtain confidence limits on the true error rate given a set of n independent test samples, drawn from the same distribution as the training data. (WEBB,2002).

Although the holdout method usually divides the data into 2 sets (training and test), it is also possible to divide it into a 3 sets (including a validation set, between the training and test sets), thus increasing the confidence of the results.

As already mentioned, the Webb (2002) methodology can be adapted, not being mandatory the use of all stages described. In this study, there are followed the stages:

1. Formulation of the problem, Data collection and Initial examination of the data;
2. Feature selection or feature extraction;
3. Unsupervised pattern classification or clustering;
4. Supervised pattern classification;
5. Assessment of results and Interpretation.

In this study, the Matlab® clustering and unsupervised classification scenarios used were Gaussian Mixture Models, Hierarchical Clustering and k-means. The Matlab® supervised classification scenarios used were Naïve Bayes, Multiclass Support Vector Machine (MSVM), Neural Network and Random Forest classifiers. All these methods are described in details in the Appendix C.

Chapter III - Methodology

As explained in Chapter II, item II.4, the methodology adopted is based on the Webb (2002) proposition described in the flowchart of figure 16, with the following adapted steps:

1. Formulation of the problem, Data collection and Initial examination of the data;
2. Feature selection or feature extraction;
3. Unsupervised pattern classification or clustering;
4. Supervised pattern classification;
5. Assessment of results and Interpretation.

In the next topics, each step will be explained in details for this study. The assessment of results are correspondent to the next Chapter IV, where the results obtained with this methodology will be explained and discussed.

III.1 Formulation of the problem, Data collection and Initial examination of the data

For this work, the proposed question is: if applying the pattern recognition methodology proposed by Webb (2002) in the ERP results from the Soto (2014) data experiment, is it possible to obtain good classification scenarios considering each type of stimulus for the epochs previously labeled (using supervised classification methods) and not labeled (unsupervised classification and clustering methods)?

Considering the visual inspection of the ERP segmented signals for the 21 original subjects done by Soto (2014), mentioned in the item II.3, 7 subjects were eliminated because of the low quality of these signals. Using the sequential order of the experiment, subjects 2, 3, 4, 5, 6, 7, 9, 10, 13, 15, 16, 17, 18, 19, 20 and 21 were used in this work. For each subject, from their EEG raw data, a specific ERPLAB® dataset needed to be created to organize this data in order to allow their treatment and analysis by MATLAB®.

Version v13.6.5b for EEGLAB® and the version v5.0.0.0 for ERPLAB® were used. Using the EEG raw data obtained by Soto experiment (2014), a procedure was performed for treatment of these data by EEGLAB® and ERPLAB® software to create the dataset files with the extensions “*.set” and “*.fdt” that allow the EEGLAB® to load the EEG dataset information for each subject, which steps can be seen in the Appendix A.

III.2 Feature selection or feature extraction

For this study, the data basic model used is as proposed by Webb (2002):

> *"…the term 'pattern' to denote the p-dimensional data vector $x = (x_1, \ldots, x_p)^T$ of measurements (^{T}denotes vector transpose), whose components x_i are measurements of the features of an object. Thus the features are the variables specified by the investigator and thought to be important for classification. In discrimination, we assume that there exist C groups or classes, denoted $\omega_1, \ldots, \omega_C$, and associated with each pattern x is a categorical variable z that denotes the class or group membership; that is, if $z = i$, then the pattern belongs to ω_i, $i \in \{1,\ldots, C)$(WEBB, 2002).*

In this study from Soto (2014) experiment, the features that were extracted for the words and sentences task are the ERP parameters Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency, in addition to the ERP Time Range, the Region of Interest (ROI) and the human subject related to each measurement. Concerning the classes, for the sentences task, they are S1 (CSC), S2 (CNSC), S3 (ISC), S4 (INSC) and S5 (Control) and, for the words task, they are S1 (SSR), S2 (ASR), S3 (Control 1 - UR) and S4 (Control 2 - PW).

To obtain the ERP signals for each ROI, it is necessary to add the contribution of each electrode channel related with the Region and take the arithmetic media. So, considering the electrode distribution of the experiment, the ERP signal for each ROI is obtained by the following equations:

$$\text{Frontal Mid Line (ch63)} = (ch2 + ch7 + ch27 + ch34 + ch38 + ch60)/6 \tag{3.1}$$

$$\text{Central Mid Line (ch64)} = (ch11 + ch21 + ch22 + ch39 + ch52 + ch56)/6 \tag{3.2}$$

$$\text{Pariental Mid Line (ch65)} = (ch11 + ch12 + ch21 + ch43 + ch51 + ch52)/6 \tag{3.3}$$

$$\text{Occipital Mid Line (ch66)} = (ch15 + ch16 + ch17 + ch46 + ch47 + ch48)/6 \tag{3.4}$$

$$\text{Frontal Left Side (ch67)} = (ch3 + ch4 + ch6 + ch35 + ch36 + ch37)/6 \tag{3.5}$$

$$\text{Central Left Side (ch68)} = (ch8 + ch9 + ch10 + ch40 + ch41 + ch42)/6 \tag{3.6}$$

$$\text{Pariental Left Side (ch69)} = (ch10 + ch13 + ch14 + ch41 + ch42 + ch44)/6 \tag{3.7}$$

$$\text{Occipital Left Side (ch70)} = (ch13 + ch14 + ch44 + ch45 + ch46)/5 \tag{3.8}$$

$$\text{Frontal Right Side (ch71)} = (ch26 + ch28 + ch29 + ch57 + ch58 + ch59)/6 \tag{3.9}$$

$$\text{Central Right Side (ch72)} = (ch20 + ch23 + ch24 + ch53 + ch54 + ch55)/6 \tag{3.10}$$

$$\text{Pariental Right Side (ch73)} = (ch18 + ch19 + ch20 + ch50 + ch53 + ch54)/6 \tag{3.11}$$

$$\text{Occipital Right Side (ch74)} = (ch18 + ch19 + ch48 + ch49 + ch50)/5 \tag{3.12}$$

The parameters/features Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency with their respectively values to use in the algorithms were extracted from the softwares EEGLAB® and ERPLAB® and the detailed procedure can be seen in the Appendix B.

For classification purposes, before building the classifier, it is necessary to use numeric single values for the features and classes, to allow the convergence of the classification methods of Matlab®. Features or classes that are not numerical variables, for instance, the Region of Interest (ROI) feature, shall be coded with numerical values with a coherent correspondence with the original string value.

In the tables 4 and 5 below, the data organization and coding of values for the other features and classes for the elaboration of the classifiers are shown. The complete set of data for Words Task are in the Appendix D and for Sentences Task are in the Appendix E.

Table 4 - Words and Sentences Task Organization and coding for features

Features	Real Value	Code for Matlab® algorithm
ERP Time Range	150-300ms	1
	300-500ms	2
	500-700ms	3
Region of Interest (ROI)	Frontal Mid Line	1
	Central Mid Line	2
	Pariental Mid Line	3
	Occiptal Mid Line	4
	Frontal Left Side	5
	Central Left Side	6
	Pariental Left Side	7
	Occiptal Left Side	8
	Frontal Right Side	9
	Central Right Side	10
	Pariental Right Side	11
	Occiptal Right Side	12
Subject	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	9	9
	10	10
	13	13
	15	15
	16	16
	17	17
	18	18
	19	19
	20	20
	21	21

Table 5 - Words and Sentences Task organization and coding for classes

Task	Classes	Code for Matlab® algorithm
Words	S1 (SSR)	1
	S2 (ASR)	2
	S3 (Control 1 - UR)	3
	S4 (Control 2 - PW)	4
Sentences	S1 (CSC)	1
	S2 (CNSC)	2
	S3 (ISC)	3
	S4 (INSC)	4
	S5 (Control)	5

As already mentioned , after the use of the EEGLAB® and ERPLAB®, described on the Appendixes A and B, the following database was extracted:

a) For Sentences Task: a matrix with 2880 lines and 7 columns, where the features corresponding to: column A: Mean Amplitude Between two fixed latencies, column B: Peak Amplitude, column C: Peak Latency, column D: Region of Interest (ROI), column E: ERP Time Range and column F: Subject index. The last column G corresponding to the classes.

b) For Words Task: a matrix with 2304 lines and 7 columns, where the features corresponding to: column A: Mean Amplitude Between two fixed latencies, column B: Peak Amplitude, column C: Peak Latency, column D: Region of Interest (ROI), column E: ERP Time Range and column F: Subject index. The last column G corresponding to the classes.

The Figure 17 shows this input data organization.

	A	B	C	D	E	F	G
1	-2,049	2,227	254	1	1	2	1
2	-1,794	-0,304	196	1	1	3	1
3	1,099	4,893	260	1	1	4	1
4	-0,339	2,548	294	1	1	5	1
5	-2,309	3,372	266	1	1	6	1
6	-0,589	2,96	180	1	1	7	1
7	1,438	5,267	206	1	1	9	1
8	3,277	9,979	274	1	1	10	1
9	1,645	4,17	200	1	1	13	1
10	-0,572	1,157	260	1	1	15	1

Column A - Mean Amplitude Between two fixed latencies
Column B - Peak Amplitude
Column C - Peak Latency
Column D - ROI
Column E - ERP time range
Column F - Subject
Column G - classes

Figure 17 - Input data organization extracted from the EEGLAB® and ERPLAB® for the classifiers.

For both Words and Sentences task, Microsoft Excel® sheets files data.xls and classes.xls for each task were created manually, considering these conversions and organized by an Excel filter putting the sheet in increasing order sequence by classes as shown in figure 18.

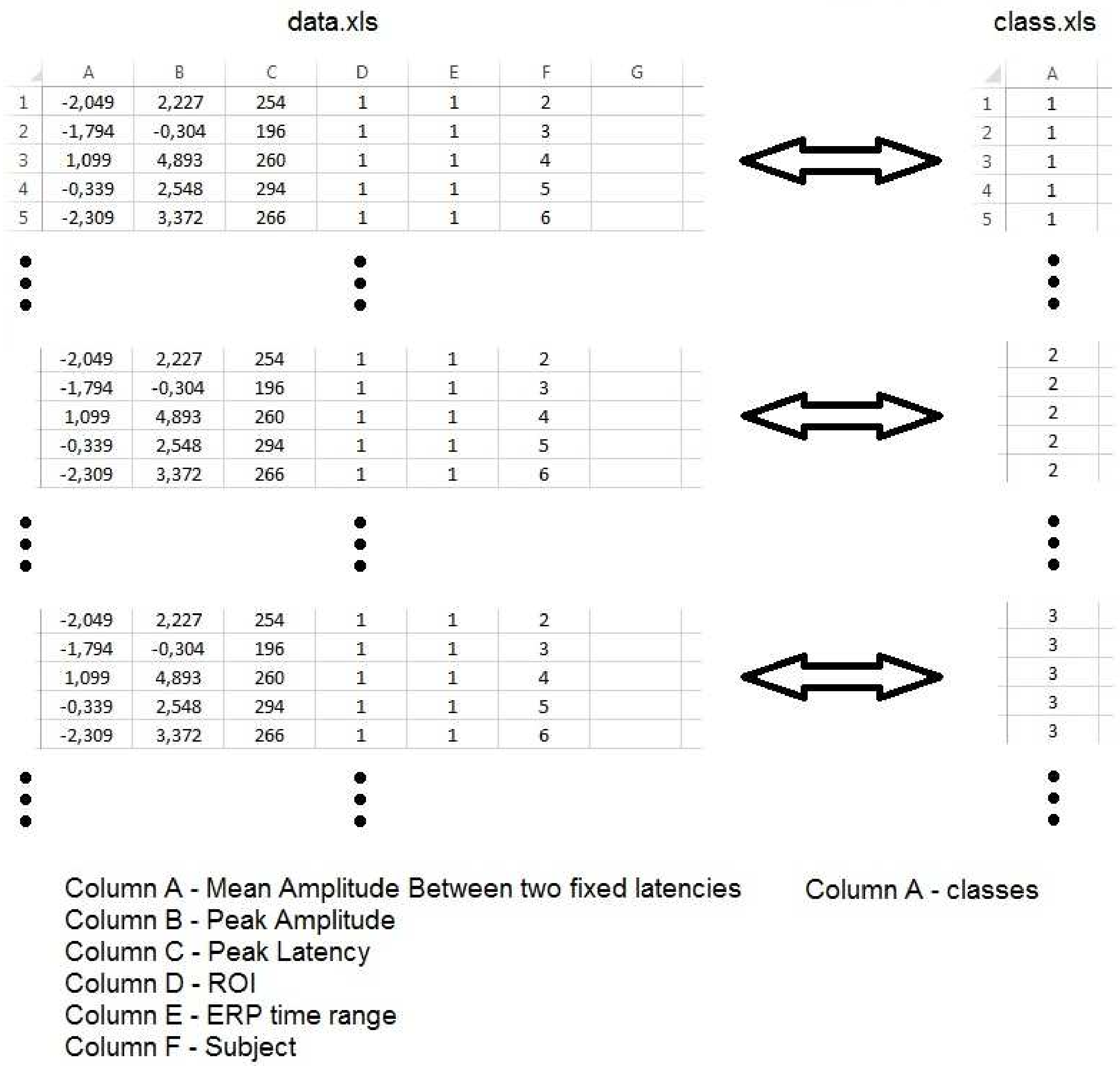


Figure 18 - Microsoft Excel® sheets format with the features (data.xls) and classes (class.xls) for both Words and Sentences task with increasing order by classes

These data are splited in two Microsoft Excel® sheet files data.xls and classes.xls for each task.

Thus, it was plotted the scattering distribution of the features Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency, combined 2-by-2, labelled by classes for each task, to perform the step “initial examination of the data”. The scatter plots are shown in the Chapter IV, for each task. The developed scripts are shown in the Annex F, items F.1 a), for Sentences task e F.1 b), for Words Task.

III.3 Unsupervised pattern classification or clustering

Concerning each dataset matrix (data.xls) for sentences and words tasks, this study consider for the unsupervised classification and clustering, the classifiers one single dataset, not being splitted in subsets as training, validation or test.

For the unsupervised clustering and classification, the Matlab® clustering methods Hierarchical Clustering, k-means and Gaussian Mixture Models was used to create the classifiers algorithms scripts:

a) For Hierarchical Clustering, first, the algorithm finds the similarity or dissimilarity between every pair of objects (points) in the data set using the “pdist” function. After that, the objects were grouped into a binary hierarchical cluster tree, using the “linkage” function. At last, it was determined where to cut the hierarchical tree into clusters. All “pdist” metrics and “linkage” method options, with arbitrary clusters for each task (4 clusters for Words task and 5 clusters for Sentences) were used and combined. Checking the dendograms and the confusion matrixes for all possibilities, many results were obtained. In order to simplify the study, the poor results were discarded and in Chapter IV only the best performance results for each task were considered for presentation, with their confusion matrixes and also their respectively dendograms. The developed scripts are shown in the Annex F, items F.2.1 a), for Sentences task e F.2.1 b), for Words Task. For more details about the algorithm, see the Appendix C.

b) For the k-means, the tests done considered all k-means metric options to obtain the silhouette plots for 2, 3, 4 and 5 clusters, for each task. After that, by the observation of the silhouette plot results, the best metric was verified. Thus, the classifiers were dine for each task with the best k-means metric and, also, for the best number of clusters and for the real number of clusters (4 clusters for Words task and 5 clusters for Sentences task) indicated by Soto (2014). As many results were obtained, Chapter IV presents the silhouettes plots and the confusions matrixes obtained only for the best k-means metrics, for each task. The developed scripts are shown in the Annex F, items F.2.2 a), for Sentences task e F.2.2 b), for Words Task. For more details about the algorithm, see the Appendix C. For more details about the algorithm, see the Appendix C.

c) For the Gaussian Mixture Models (GMM), the method considers just 2 features to do the clustering. Thus, in order to simplify the test, only the 3 ERP features extracted from ERPLAB® (Peak Amplitude, Peak latency and Mean Amplitude Between Two Fixed Latencies) were considered, ignoring the ROI, the Subject and the ERP range, considering the events independently of these features. 3 GMM classifiers were developed, using the combinations 2-bt-2 of the 3 parameters, for Sentences and Word tasks. The developed scripts are shown in the Annex F, items F.2.3 a), for Sentences task e F.2.3 b), for Words Task. For more details about the algorithm, see the Appendix C.

III.4 Supervised pattern classification

For supervised classification, the Words and Sentences datasets (data.xls and class.xls) were splitted in three sets with the same amount of data with all features for each class coming from the Words and Sentence Task. The sets are defined as training set, validation set and test set, respectively, with 1/3 of the total amount.

For supervised classification, in order to assess the performances of the classifiers, for each try, the confusion matrixesa and the accuracies were used.

Concerning the supervised classification, four scripts were developed using Matlab® functions to the following techniques: Naïve Bayes, Multiclass Support Vector Machine (SVM), Neural Network and Random Forest.

a) For the Naïve Bayes technique, all probability distribution functions were tested, i.e., normal (normal (Gaussian) distribution), kernel (kernel smoothing density estimate) and MVMN (multivariate multinomial distribution), for both tasks. For more details about the algorithm, see the Appendix C. The developed scripts are shown in the Annex F, items F.3.1 a), for Sentences task e F.3.1 b), for Words Task. The results are shown in Chapter IV, for each task.

b) For the Multiclass Support Vector Machine, several tests were done changing the parameters Box Constraint, Kernel Function (gaussian or rbf, linear or polynomial), Standardize (true or false), for both tasks, to achieve the best result. For more details about the algorithm, see the Appendix C. The developed scripts are shown in the Annex F, items F.3.2 a), for Sentences task e F.3.2 b), for Words Task. As many results were obtained, it was showed in Chapter IV just the best results achieved, for each task.

c) For the Neural Network, several tests were done changing the parameters Number of hidden layers, Neural Network Input-Output Processing Functions, Divide Mode, Multilayer Neural Network Training Function and Neural Network Performance Function, for both tasks, to achieve the best result. For more details about the algorithm, see the Appendix C. The developed scripts are shown in the Annex F, items F.3.3 a), for Sentences task e F.3.3 b), for Words Task. As many results were obtained, it was showed in the Chapter IV just the best results achieved, for each task.

d) For the Random Forest, the method “bag” was used, with the Learner “Tree” that allows the creation of a random forest classifier. For this study, several tests were done changing the number of Learners up to achieve the best result. As many results were obtained, only the best results were showed in Chapter IV. For more details about the algorithm, see the Appendix C. The developed scripts are shown in the Annex F, items F.3.4 a), for Sentences task e F.3.4 b), for Words Task.

Chapter IV - Results and Discussion

In this Chapter VI, the last step of the method proposed by Webb (2002), the assessment of the results will be addressed. The behavior of the methods used will be also discussed.

IV.1 Initial examination of the data

The first attempt at identifying possible coherent clusters for the features and the classes by doing the analysis of the scattering plotting of the ERP parameters features Peak Latency Peak Amplitude and Mean Amplitude Between two fixed latencies, labelled by the classes of the experiments.

These results, except for the parameters ROI, ERP Time Range and the Subject, which were not presented in thus analysis, are the expected result for all classification method for this study.

In Figure 19 the scatter plot of the Peak Latency with their correspondent Peak Amplitude by classes, for the Sentences Task is shown. A high degree of shuffling concerning the classes observed, not indicating clearly regions to cluster each class.

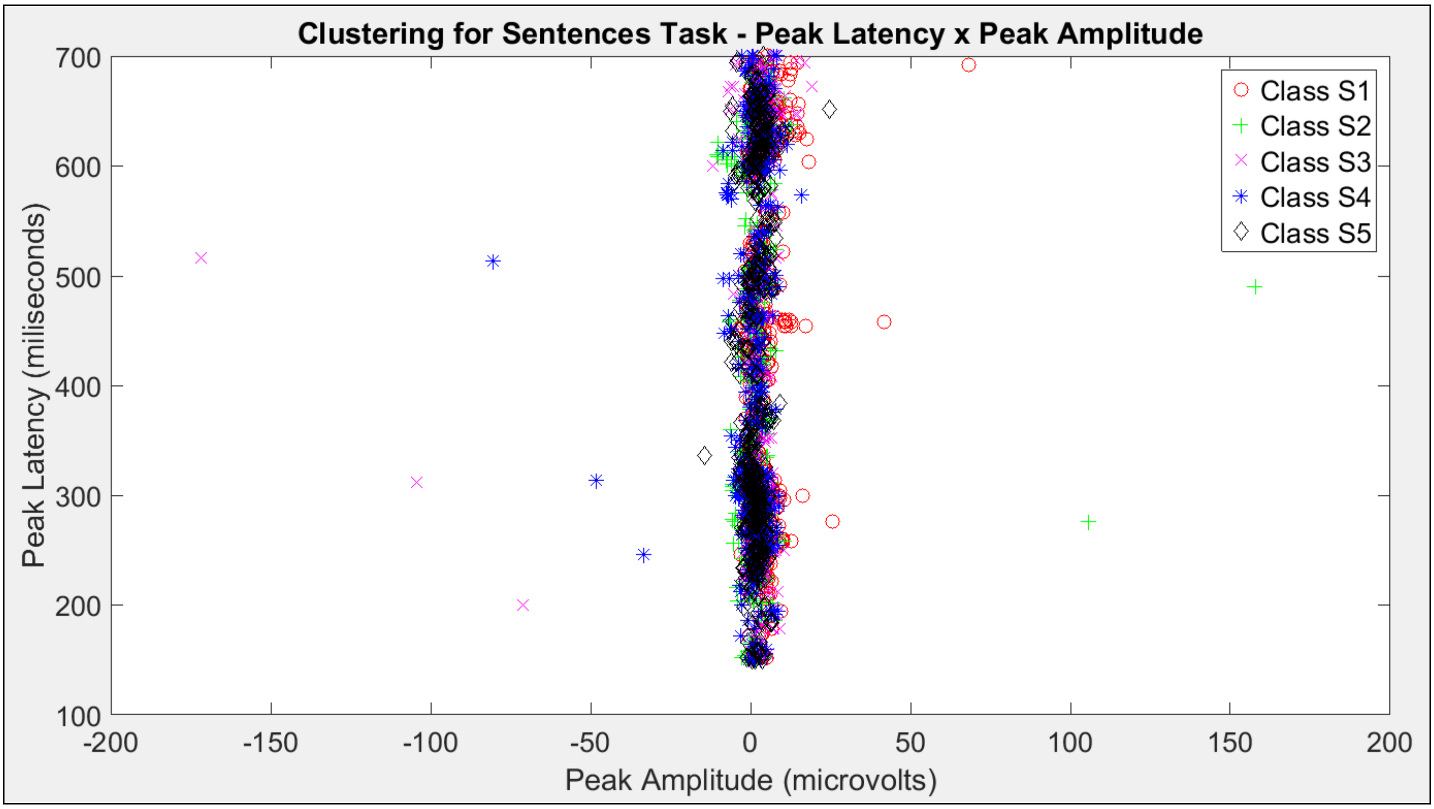


Figure 19- Clustering for Sentences Task - Peak Latency x Peak Amplitude

In Figure 20, the scatter plot of the Peak Latency with their correspondent Mean Amplitude Between two fixed latencies by classes for the Sentences Task is shown.

In this case, a high degree of shuffling concerning the classes is also observed, already observed in the prevous result, not indicating, also, clearly regions to cluster each class.

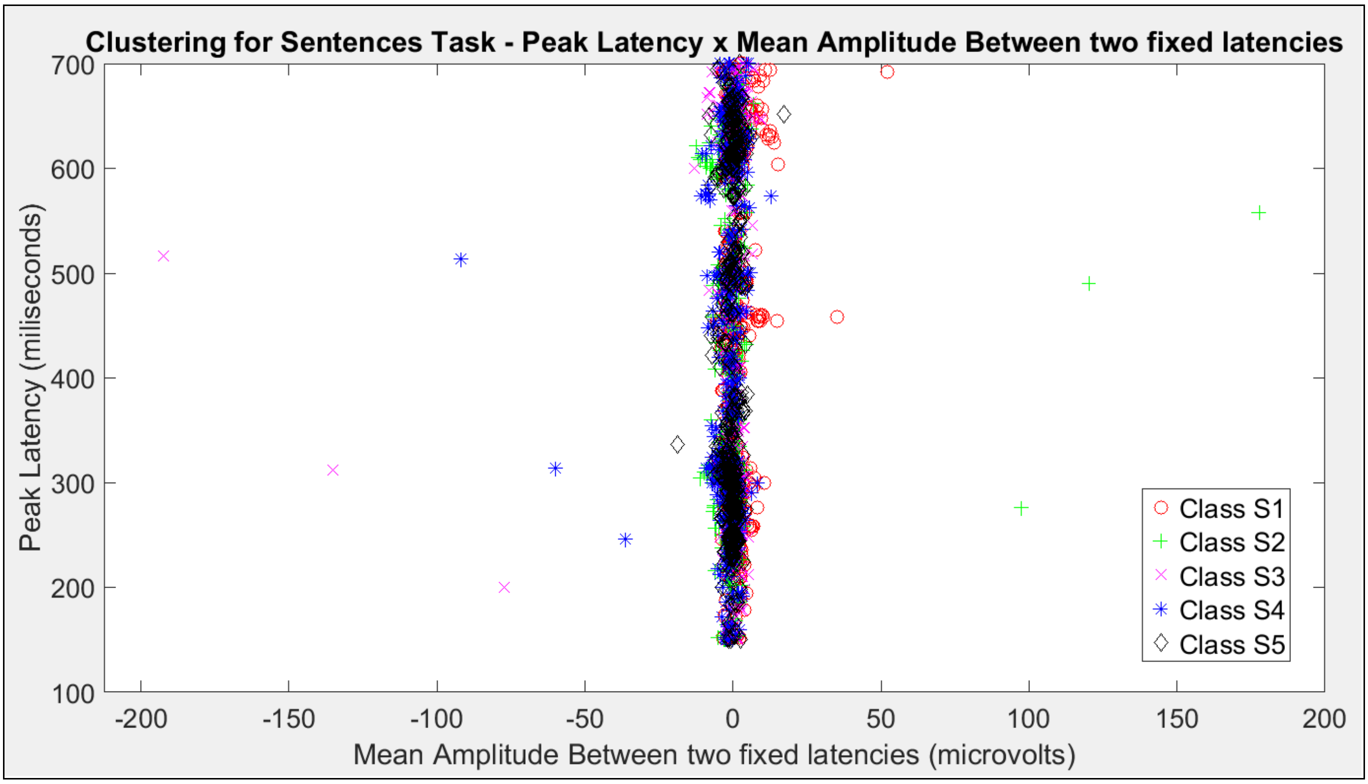


Figure 20- Clustering for Sentences Task - Peak Latency x Mean Amplitude Between two fixed latencies

In Figure 21, the scatter plot of the Peak Amplitude with their correspondent Mean Amplitude Between two fixed latencies by classes for the Sentences Task is shown.

In this case, a high degree of shuffling, concerning the classes is also observed, already observed in the prevous results, not indicating, also, clearly regions to cluster each class.

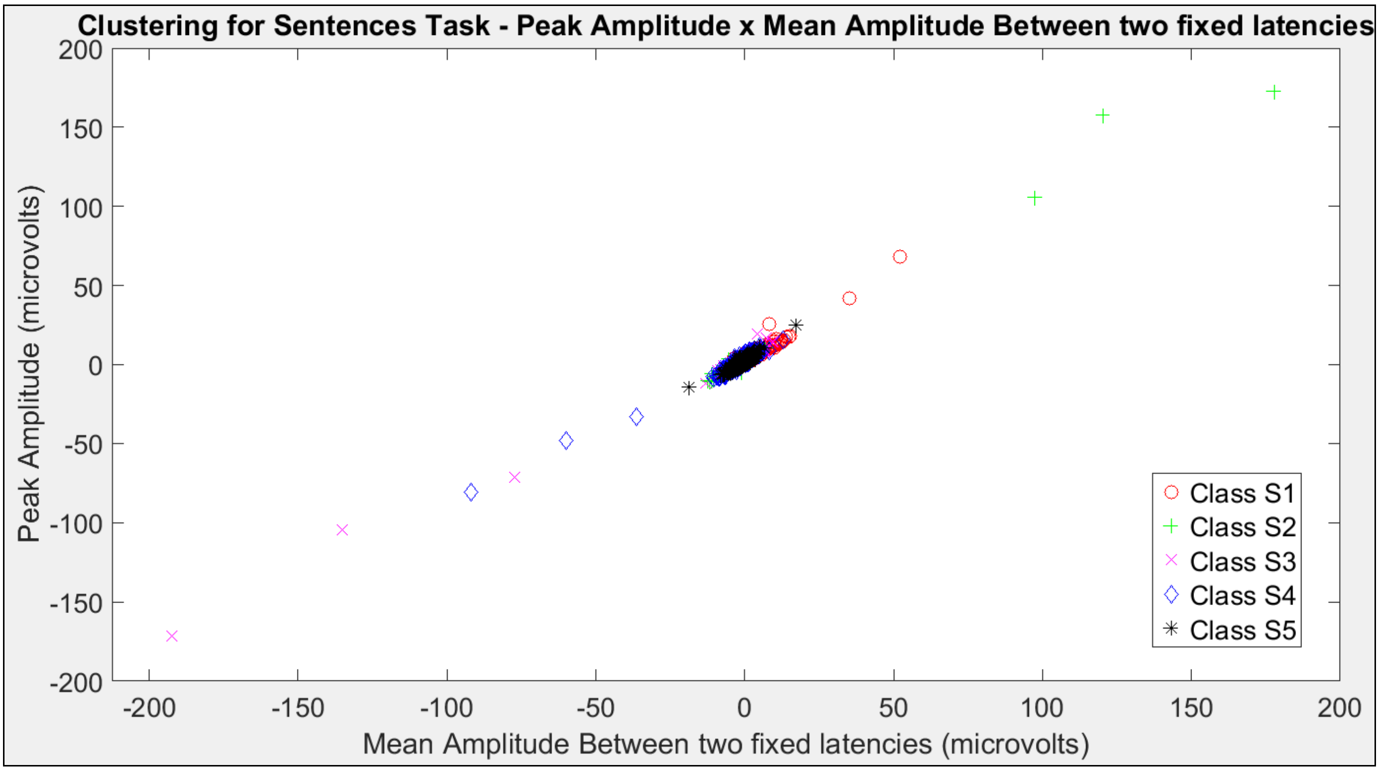


Figure 21 - Clustering for Sentences Task - Peak Amplitude x Mean Amplitude Between two fixed latencies

In Figure 22, the scatter plot of the Peak Latency with their correspondent Peak Amplitude by classes, for the Words Task is shown.

In the same way, a high degree of shuffling concerning the classes is observed, not indicating clearly regions to cluster each class.

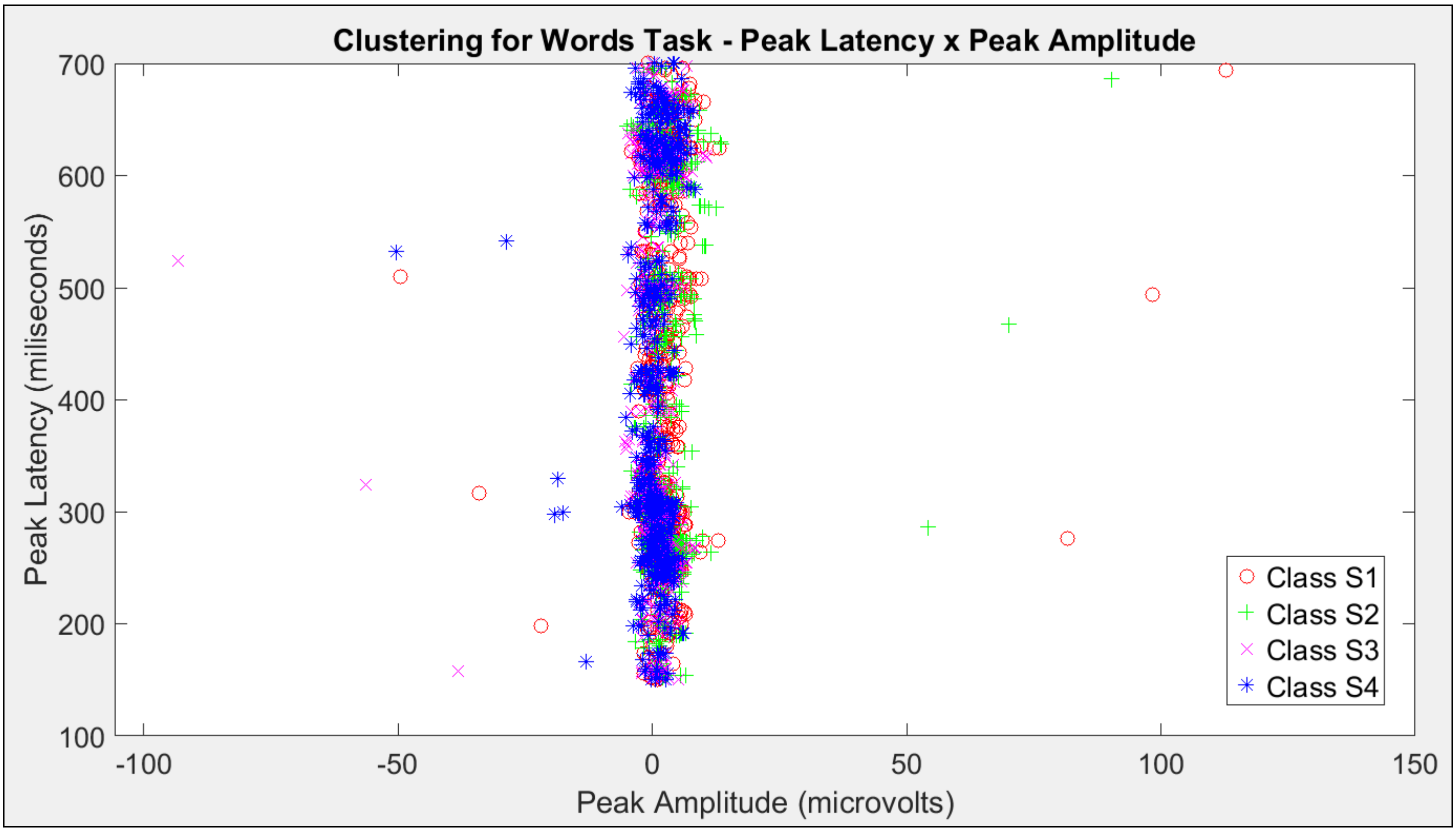


Figure 22 - Clustering for Words Task - Peak Latency x Peak Amplitude

In Figure 23, the scatter plot of the Peak Latency with their correspondent Mean Amplitude Between two fixed latencies by classes for the Words Task is shown.

In this case, a high degree of shuffling concerning the classes is also observed, already observed in the prevous result, not indicating, also, a clearly regions to cluster each class.

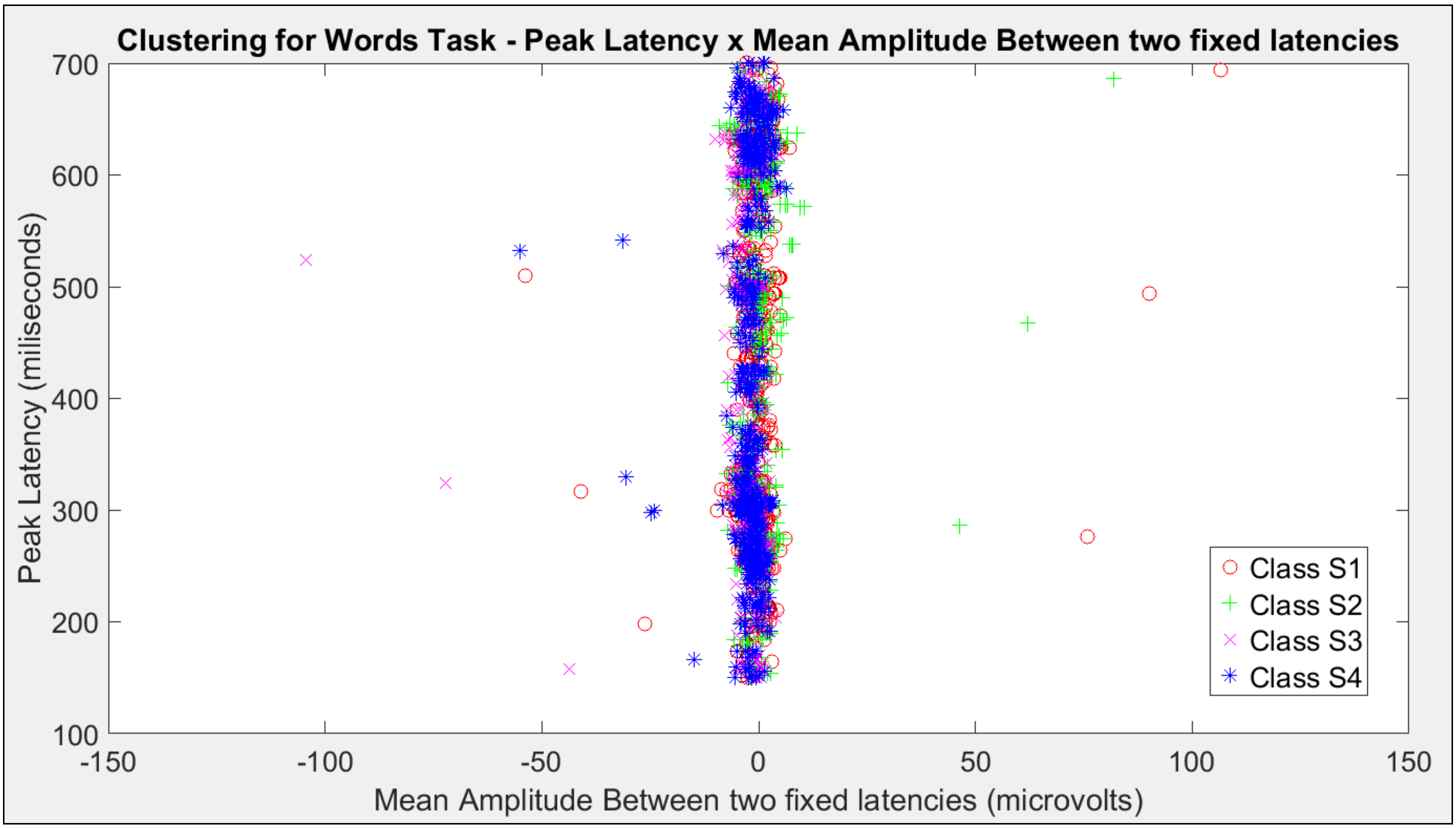


Figure 23 - Clustering for Words Task - Peak Latency x Mean Amplitude Between two fixed latencies

In Figure 24, the scatter plot of the Peak Amplitude with their correspondent Mean Amplitude Between two fixed latencies by classes for the Words Task is shown.

In this case, a high degree of shuffling, concerning the classes, is also observed, as seen in the prevous results, not indicating, also, a clearly regions to cluster each class.

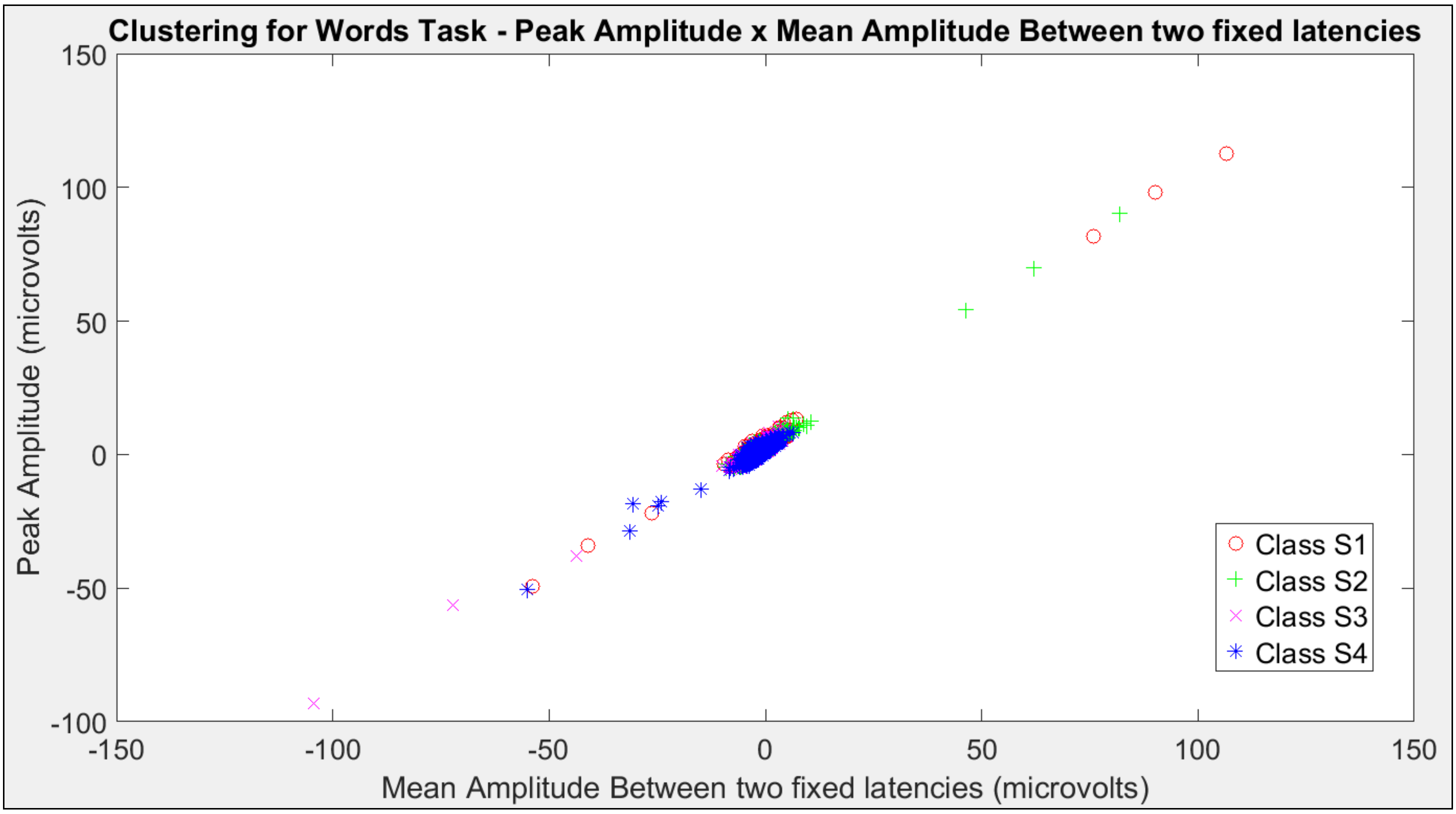


Figure 24 - Clustering for Words Task - Peak Amplitude x Mean Amplitude Between two fixed latencies

The results showed graphically that for the Sentences task as well as the Words task, the level of dispersion and mixing data between the classes is very high, not allowing a direct obviously clustering by the plot location of the combined ERP parameters.

Because of this, the study follows the Webb (2002) methodology, and continued to perform clustering through unsupervised methods.

IV.2 Unsupervised pattern classification and clustering

The clustering and also classification essays were performed to identify if without indicating the labelling of the classes, a good classification method for the Sentences task and Words task could be achieved.

IV.2.1 Hierarchical Clustering and Unsupervised Classifier

For Hierarchical Clustering and Unsupervised Classifiers, the best accuracies results will be presented by means of their confusion matrixes, for all tries, and their respective dendograms.

In Figure 25, the confusion matrixes and accuracies for the best results of Hierarchical Clustering Unsupervised Classifiers for Sentences Task are shown, with a “pdist” metric “cityblock” with a “linkage” method “average” and with a “pdist” metric “cityblock” with a “linkage” method “centroid”, both achieving an accuracy of 21,63%.

Hierarchical Cluster (pdist metric: cityblock Linkage Method: average):
accuracy = 21.63%
Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	311	265	0	0	0
S2	263	311	0	2	0
S3	260	315	1	0	0
S4	262	313	1	0	0
S5	245	331	0	0	0

Hierarchical Cluster (pdist metric: cityblock Linkage Method: centroid)
accuracy = 21.63%
Confusion Matrix for the test

T	S1	S2	S3	S4	S5
S1	311	265	0	0	0
S2	263	311	0	2	0
S3	260	315	1	0	0
S4	263	312	1	0	0
S5	245	331	0	0	0

Figure 25 - Confusion Matrix and accuracy for the best results of Hierarchical Clustering and Unsupervised Classifiers for Sentences Task

In Figure 26, the dendograms for the best results of Hierarchical Clustering Unsupervised Classifiers for Sentences Task are shown, with a “pdist” metric “cityblock” with a “linkage” method “average”, and with a “pdist” metric “cityblock” with a “linkage” method “centroid”.

The height for the correct number of clusters (5 clusters for Sentences) occurred with low level of consistence in the dendograms in the “linkage” method “average” and was verified that is impossible to distinguish graphically for the “linkage” method “centroid”, being 4 clusters the closest value. In fact, as indicated in Figure 26, the better level of consistence for both cases is better with 7 or more clusters.

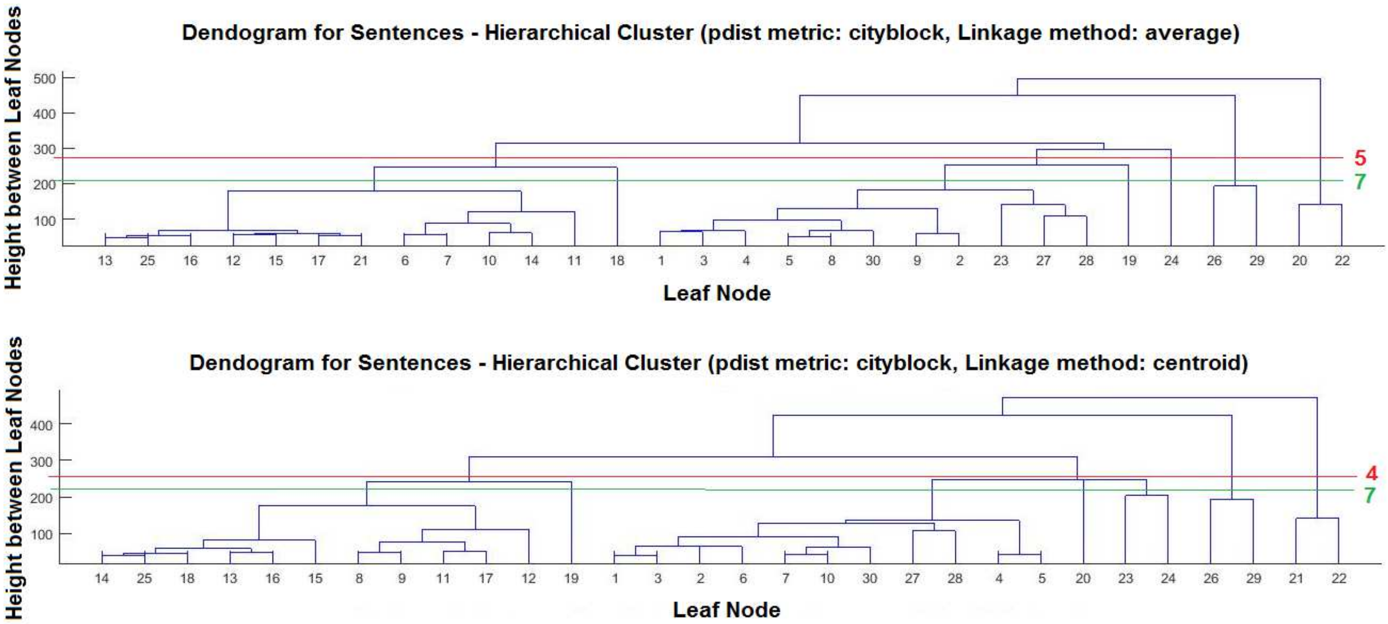
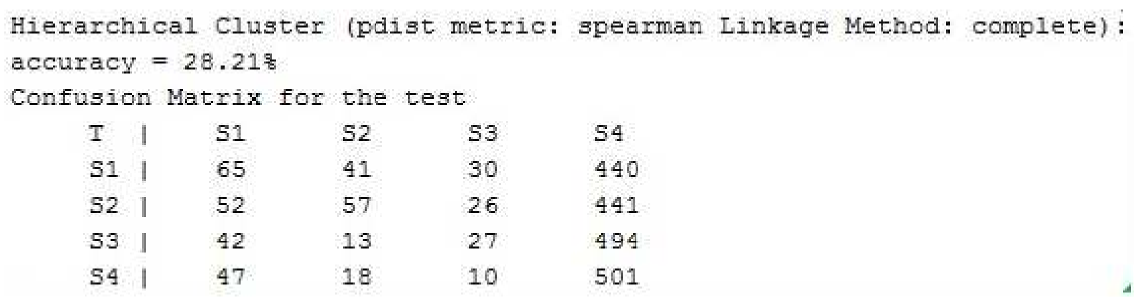


Figure 26 - Dendograms for the best results of Hierarchical Clustering and Unsupervised Classifiers for Sentences Task

In Figure 27, the confusion Matrix and accuracy for the best result of Hierarchical Clustering Unsupervised Classifiers for Words Task are shown, with a “pdist” metric “spearman” with a “linkage” method “single”, achieving an accuracy of 28,21%.



Hierarchical Cluster (pdist metric: spearman Linkage Method: complete):
accuracy = 28.21%
Confusion Matrix for the test

T	S1	S2	S3	S4
S1	65	41	30	440
S2	52	57	26	441
S3	42	13	27	494
S4	47	18	10	501

Figure 27 - Confusion Matrix and accuracy for the best results of Hierarchical Clustering and Unsupervised Classifiers for Words Task

In Figure 28, the dendograms for the best result of Hierarchical Clustering Unsupervised Classifiers for Words Task are shown, with a “pdist” metric “spearman” with a “linkage” method “complete”.

The height for the correct number of clusters (4 clusters for Words) occurred with low level of consistence in the dendogram. In fact, as indicated in Figure 28, this level of consistence is better with 9 or more clusters.

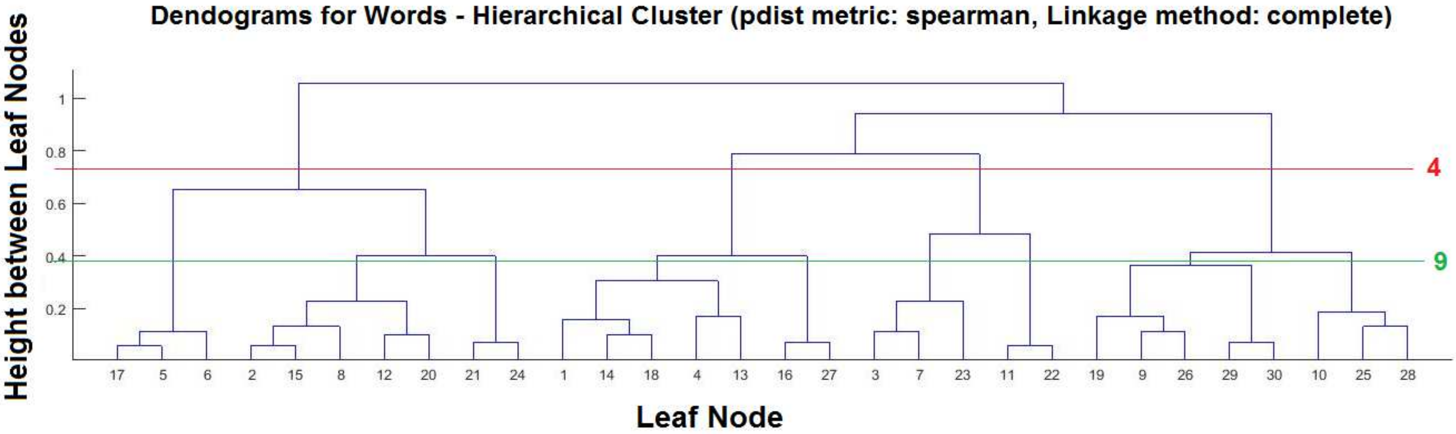


Figure 28 - Dendograms for the best results of Hierarchical Clustering and Unsupervised Classifiers for Words Task

The best results for Sentences Task are for the “pdist” metric “cityblock” and “linkage” methods “average” and “centroid”, with accuracies of 21,63%. The best result for Words Task are for the “pdist” metric “spearman” and “linkage” method “complete”, with accuracy of 28,21%.

The accuracies for both tasks are low, and indicate that this method is not good for clustering and classifying both tasks.

IV.2.2 K-means Clustering and Unsupervised Classifier

For k-means clustering, first, silhouette plots for the best k-means metrics were applied to check the best number of clusters for each Task.

After that, the unsupervised classifiers for the real number of classes for each task and for the best number of clusters indicated in the silhouette plots were done.

In Figure 29, the silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric “cityblock” are shown. It can observed that the best number of clusters achieved is 2 clusters, although the real number of clusters is 5.

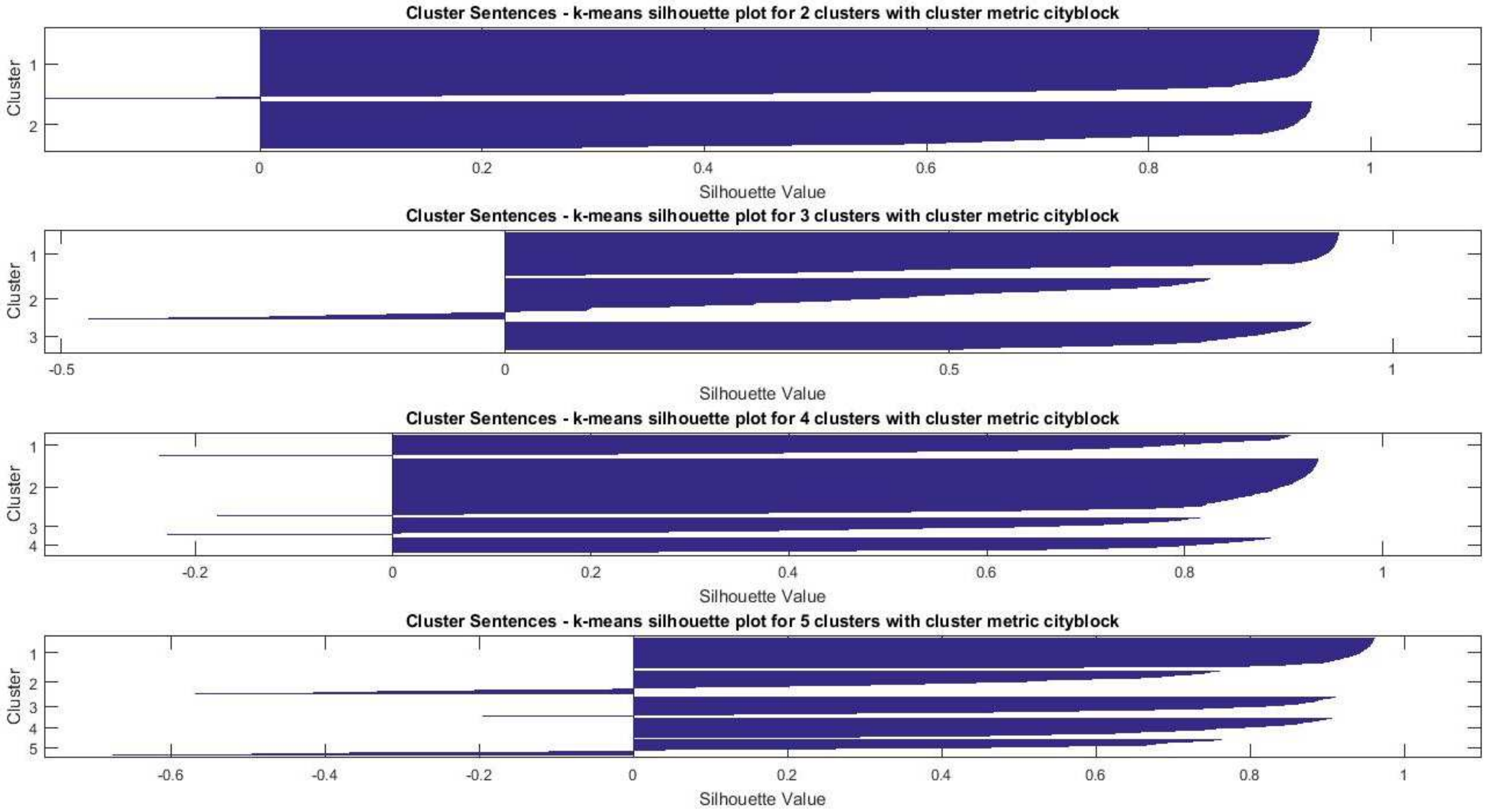


Figure 29 - Silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric "cityblock"

After that, two classifiers considering 2 clusters (silhouette plot result) and 5 clusters (real number of classes for Sentences Task) were done.

For the verification of the 2 clusters and to maintain the coherence with the 5 original classes of Soto(2014) experiment, the classes S1 and S2 (congruous) are joined togheter in the first cluster and S3, S4 and S5 (incongruous) are joined togheter in the second cluster to the verification.

In Figure 30, the confusion Matrix and accuracies for the k-means unsupervised classifiers for 2 clusters (best silhouette result) are shown, achieving an accuracy of 52,92 %, and for 5 clusters (real number of classes) of Sentences Task with k-means metric "cityblock", achieving an accuracy of 19,44%.

```
kmeans cityblock with 2 clusters:
accuracy = 52.92%
Confusion Matrix for the test
     T  |    S1      S2
     S1 |    487     665
     S2 |    691     1037

kmeans cityblock with 5 clusters:
accuracy = 19.44%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    135     129     82      83      147
     S2 |    170     141     72      93      100
     S3 |    184     130     118     60      84
     S4 |    186     120     85      85      100
     S5 |    209     122     74      90      81
```

Figure 30 - K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 5 clusters (real number of classes) of Sentences Task with k-means metric “cityblock”

In Figure 31, the silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric “sqEuclidean” are shown. It can observed that the best number of clusters achieved is also 2 clusters, although the real number of clusters is 5.

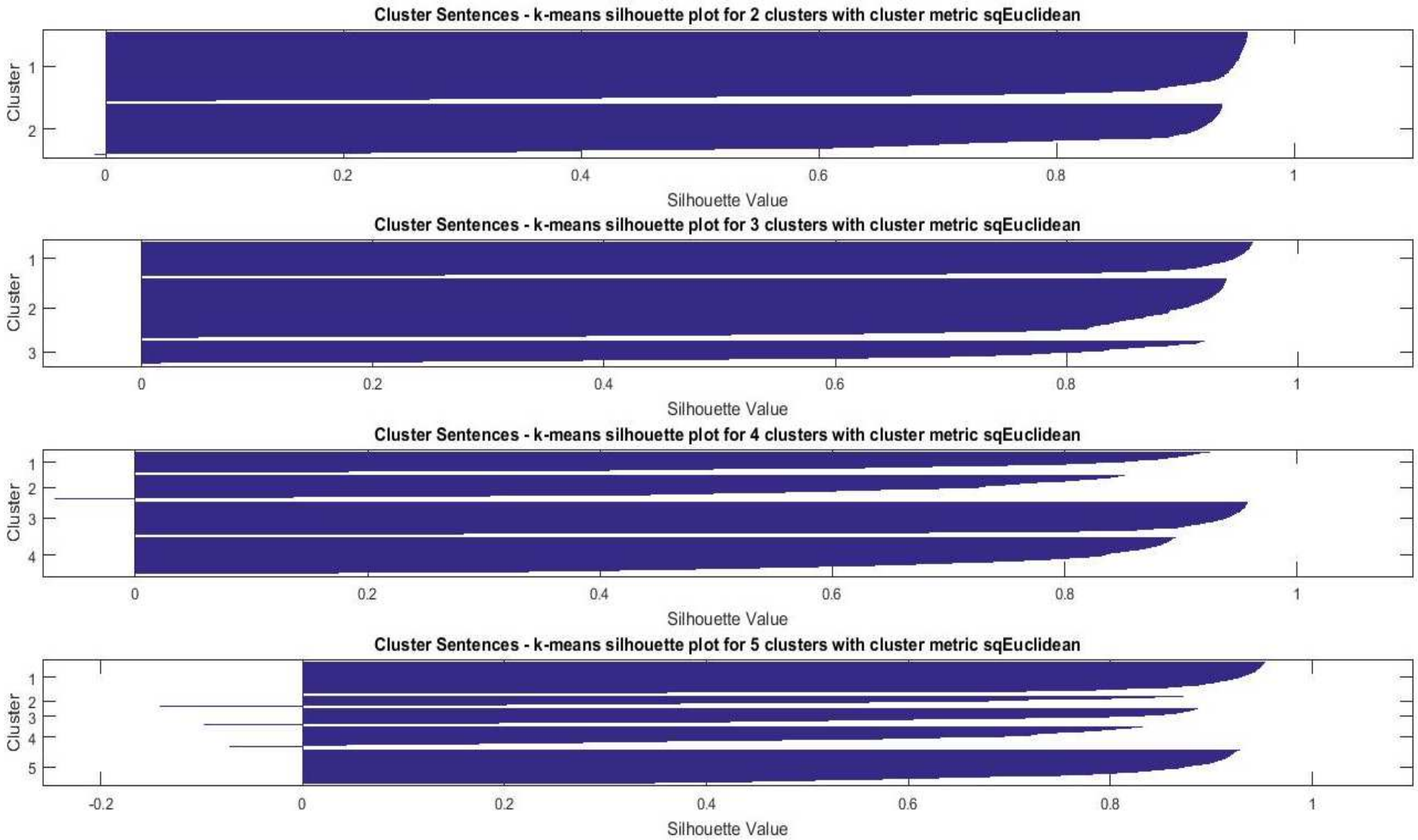


Figure 31 - Silhouette plots for 2, 3, 4 and 5 clusters of Sentences Task with k-means metric “sqEuclidean”

Similarly to k-means metric "cityblock", two classifiers considering 2 clusters (silhouette plot result) and 5 clusters (real number of classes for Sentences Task) were done, with the same adaptation of the classes for the verification of the 2 clusters, as already explained in this topic.

In Figure 32, the confusion Matrix and accuracies for the k-means unsupervised classifiers for 2 clusters (best silhouette result) are shown, achieving an accuracy of 53,33%, and for 5 clusters (real number of classes) of Sentences Task with k-means metric "sqEuclidean", achieving an accuracy of 18,13%.

```
kmeans sqEuclidean with 2 clusters:
accuracy = 53.33%
Confusion Matrix for the test
     T  |     S1      S2
     S1 |     522     630
     S2 |     714     1014

kmeans sqEuclidean with 5 clusters:
accuracy = 18.13%
Confusion Matrix for the test
     T  |     S1      S2      S3      S4      S5
     S1 |     143     158     62      108     105
     S2 |     160     158     51      83      124
     S3 |     180     172     43      63      118
     S4 |     198     162     52      80      84
     S5 |     200     156     55      67      98
```

Figure 32 - K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 5 clusters (real number of classes) of Sentences Task with k-means metric "sqEuclidean"

Now, the results for the Words task will be shown. In Figure 33, the silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric "cityblock" are shown. It is observed that the best number of clusters achieved is 2 clusters, although the real number of clusters is 4.

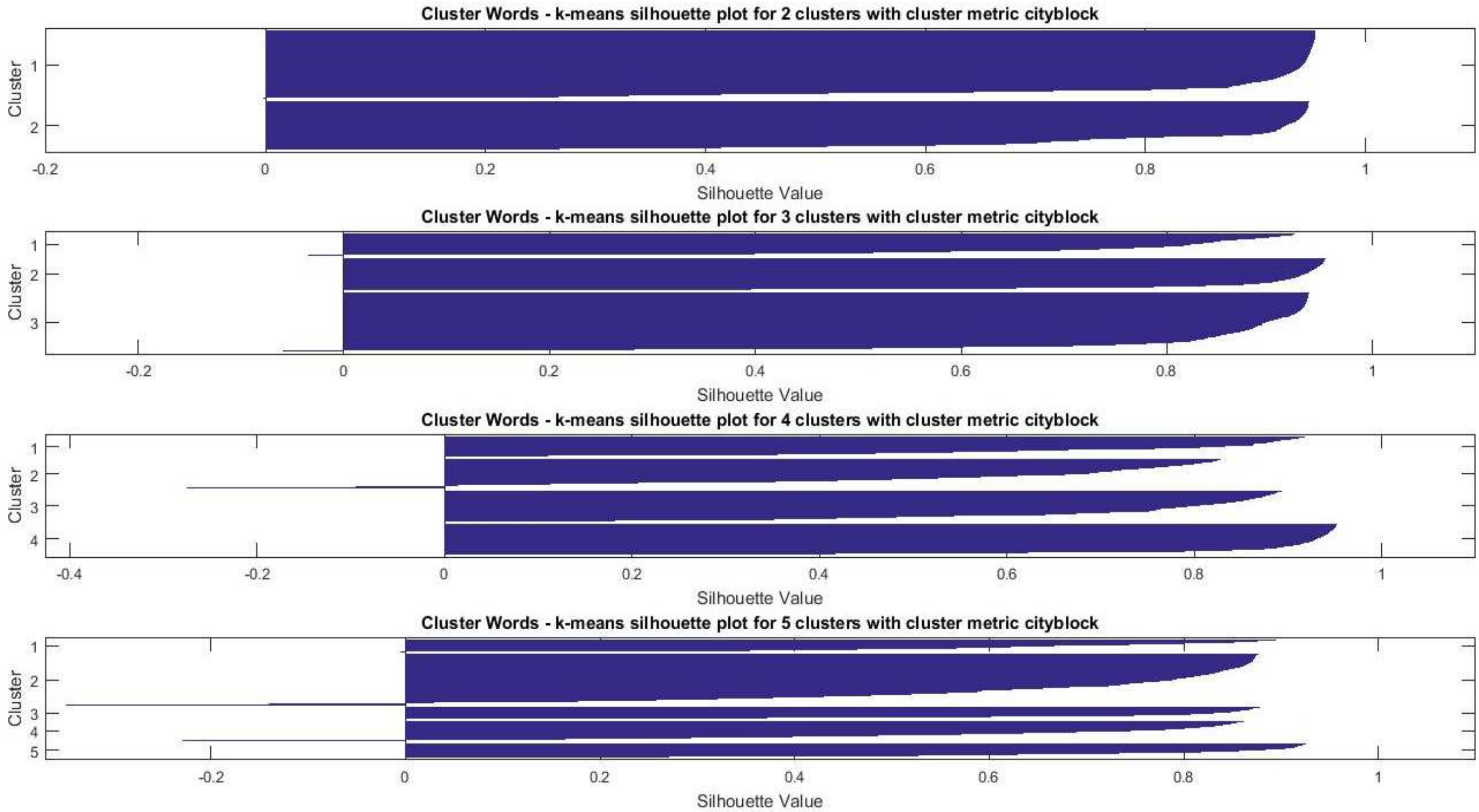


Figure 33 - Silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric "cityblock"

After that, itwo classifiers considering 2 clusters (silhouette plot result) and 4 clusters (real number of classes for WordsTask) were done.

For the verification of the 2 clusters and to maintain the coherence with the 4 original classes of Soto (2014) experiment, the classes S1 and S2 (semantic) are joined togheter in the first cluster and S3 and S4 (no semantic) are joined togheter in the second cluster to the verification.

In Figure 34, the confusion Matrix and accuracies for the k-means unsupervised classifiers for 2 clusters (best silhouette result) are shown, achieving an accuracy of 48,44 %, and for 4 clusters (real number of classes) of Words Task with k-means metric "cityblock", achieving an accuracy of 23,26%.

```
kmeans cityblock with 2 clusters:
accuracy = 48.44%
Confusion Matrix for the test
     T  |     S1      S2
     S1 |     658     494
     S2 |     694     458

kmeans cityblock with 4 clusters:
accuracy = 23.26%
Confusion Matrix for the test
     T  |     S1      S2      S3      S4
     S1 |     103     282     55      136
     S2 |     98      286     67      125
     S3 |     119     319     51      87
     S4 |     93      307     80      96
```

Figure 34 - K-means unsupervised classifiers for 2 clusters (best silhouette result) and for 4 clusters (real number of classes) of Words Task with k-means metric “cityblock”

In Figure 35, the silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric “sqEuclidean” are shown. It can be observed that the best number of clusters achieved is 3 clusters, although the real number of clusters is 4.

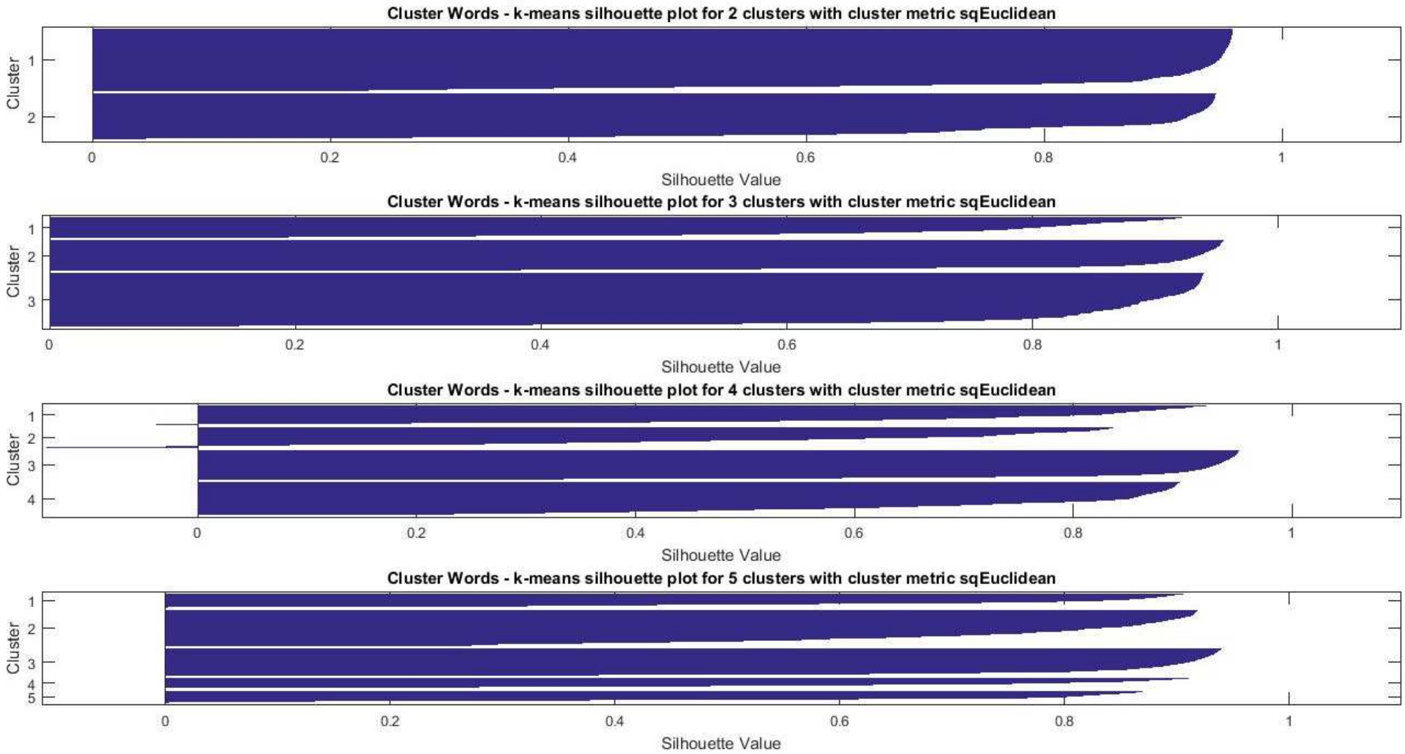


Figure 35 - Silhouette plots for 2, 3, 4 and 5 clusters of Words Task with k-means metric “sqEuclidean”

For the verification of the 3 clusters and to maintain the coherence with the 4 original classes of Soto (2014) experiment and the silhouette plot indication, the classes S1 and S2 (semantic) are maintained separate and S3 and S4 (no semantic) are joined togheter in the third cluster for the verification.

In Figure 36, the confusion Matrix and accuracies for the k-means unsupervised classifiers for 3 clusters (best silhouette result) are shown, achieving an accuracy of 32,68 %, and for 4 clusters (real number of classes) of WordsTask with k-means metric “sqEuclidean”, achieving an accuracy of 25,00%.

```
kmeans sqEuclidean with 3 clusters:
accuracy = 32.68%
Confusion Matrix for the test
     T  |    S1      S2      S3
     S1 |    465     158     145
     S2 |    407     165     196
     S3 |    302     343     123

kmeans sqEuclidean with 4 clusters:
accuracy = 25.00%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4
     S1 |    134     100     158     184
     S2 |    124     91      164     197
     S3 |    88      108     169     211
     S4 |    94      127     173     182
```

Figure 36 - K-Means unsupervised classifiers for 3 clusters (best silhouette result) and for 4 clusters (real number of classes) of Sentences Task with k-means metric “sqEuclidean”

In fact, this clustering method also did not have a good performance for both tasks. The values of accuracies for unsupervised classifiers are near of the equiprobable situation (for instance, a dice), that is, 1/2 (~50%) for 2 clusters, 1/3 (~33%) for 3 clusters, and 1/5 (~20%) for 5 clusters, indicating that this method is not good for these data also.

IV.2.3 Gaussian Mixture Models Clustering and Unsupervised Classifier

The Matlab® GMM clustering method chosen allows the use of just 2 features to cluster and, also, to classify a dataset. Thus, this study considered the 03 ERP parameters (Mean Amplitude Between two fixed latencies, Peak Amplitude, Peak Latency), two-by-two, as features, providing three clustering processes and, consequently, 3 unsupervised classifiers for each Task.

Thus, going on to use the GMM model for unsupervised classification of the data for Sentences task, Figure 37 shows the confusion matrixes and the obtained accuracies for the three classifiers built.

```
Gaussian Mixture Models Cluster for Sentences - MeanAmp2FixedLat and PeakAmp Attributes:
accuracy = 19.24%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    3       64      208     227     74
     S2 |    3       37      188     242     106
     S3 |    4       35      229     217     91
     S4 |    3       40      189     243     101
     S5 |    2       31      143     358     42
Gaussian Mixture Models Cluster for Sentences - MeanAmp2FixedLat and PeakLat Attributes:
accuracy = 21.18%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    158     70      137     209     2
     S2 |    146     65      131     231     3
     S3 |    147     32      157     237     3
     S4 |    151     43      150     229     3
     S5 |    148     51      136     240     1
Gaussian Mixture Models Cluster for Sentences - PeakAmp and PeakLat Attributes:
accuracy = 19.24%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4      S5
     S1 |    202     156     62      150     6
     S2 |    228     158     85      102     3
     S3 |    220     172     94      87      3
     S4 |    217     161     96      98      4
     S5 |    245     155     86      88      2
```

Figure 37 - GMM unsupervised classifiers' confusion matrixes and their accuracies results for Sentences Task

As can be seen in Figure 37, for the Sentences task, the best result was for the features Peak Latency and Mean amplitude Between two fixed latencies with accuracy of 21,18%.

Using the GMM model for unsupervised classification of the data for Words task, Figure 38 shows the confusion matrixes and the obtained accuracies for the three GMM model unsupervised classification classifiers for Words task built. As can be seen, for the Words task, the best accuracy achieved was 24,91% for the features Peak Amplitude and Mean Amplitude Between two fixed latencies.

```
Gaussian Mixture Models Cluster for Words - MeanAmp2FixedLat and PeakAmp Attributes):
accuracy = 24.91%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4
     S1 |    256     96      218     6
     S2 |    268     79      226     3
     S3 |    260     80      233     3
     S4 |    202     52      316     6
Gaussian Mixture Models Cluster for Words - MeanAmp2FixedLat and PeakLat Attributes):
accuracy = 24.78%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4
     S1 |    213     212     145     6
     S2 |    224     195     154     3
     S3 |    239     177     157     3
     S4 |    226     184     160     6
Gaussian Mixture Models Cluster for Words - PeakAmp and PeakLat Attributes):
accuracy = 24.39%
Confusion Matrix for the test
     T  |    S1      S2      S3      S4
     S1 |    205     7       216     148
     S2 |    216     3       200     157
     S3 |    223     3       194     156
     S4 |    213     6       197     160
```

Figure 38 - GMM unsupervised classifiers' confusion matrixes and their accuracies results for Words Task

In fact, as already happened with the previous methods, this clustering method also did not have a good performance for both tasks.

Considering all clustering methods used, the best result was achieved in the Words task with Hierarchical Clustering with a accuracy of 28%, i.e, higher than the equiprobable situation. The results, unfortunately, cannot be considered good enough to be considered as a solution for these datasets.

IV.3 Supervised Pattern Classification

After these bad results for clustering and unsupervised classification of the pattern recognition campaign, following the methodology proposed, the next attempt was the supervised classifiers, that is, using the known labels of the classes in the machine learning of the classifiers.

As already mentioned in the Chapter III, for all supervised methods, the data was split in 3 sets (training, validation and test) with the same size for each class, for both tasks, and the assessment of the results were done by the Confusion Matrix and the accuracies, indicating the performances of the classifiers.

IV.3.1 Naïve Bayes Supervised Classifier

For this classification, three distribution functions were used, as Matlab® method "fitNaiveBayes" allows: normal (normal (Gaussian) distribution), kernel (kernel smoothing density estimate), and MVMN (multivariate multinomial distribution) were used.

For the normal function, the algorithm converges only with a positive within-class variance for each feature. However, given that the feature ERP time range related to the class S1 presents a negative within-class variance, the algorithm did not converge. For multinomial distribution also did not converge, because the dataset are not strings.

Thus, the method continued to be tested with the distribution functions kernel (kernel smoothing density estimate) and MVMN (multivariate multinomial distribution).

In Figure 39, the Naïve Bayes Supervised Classifier with kernel distribution Confusion Matrixes and Accuracies, for Sentences task, are shown.

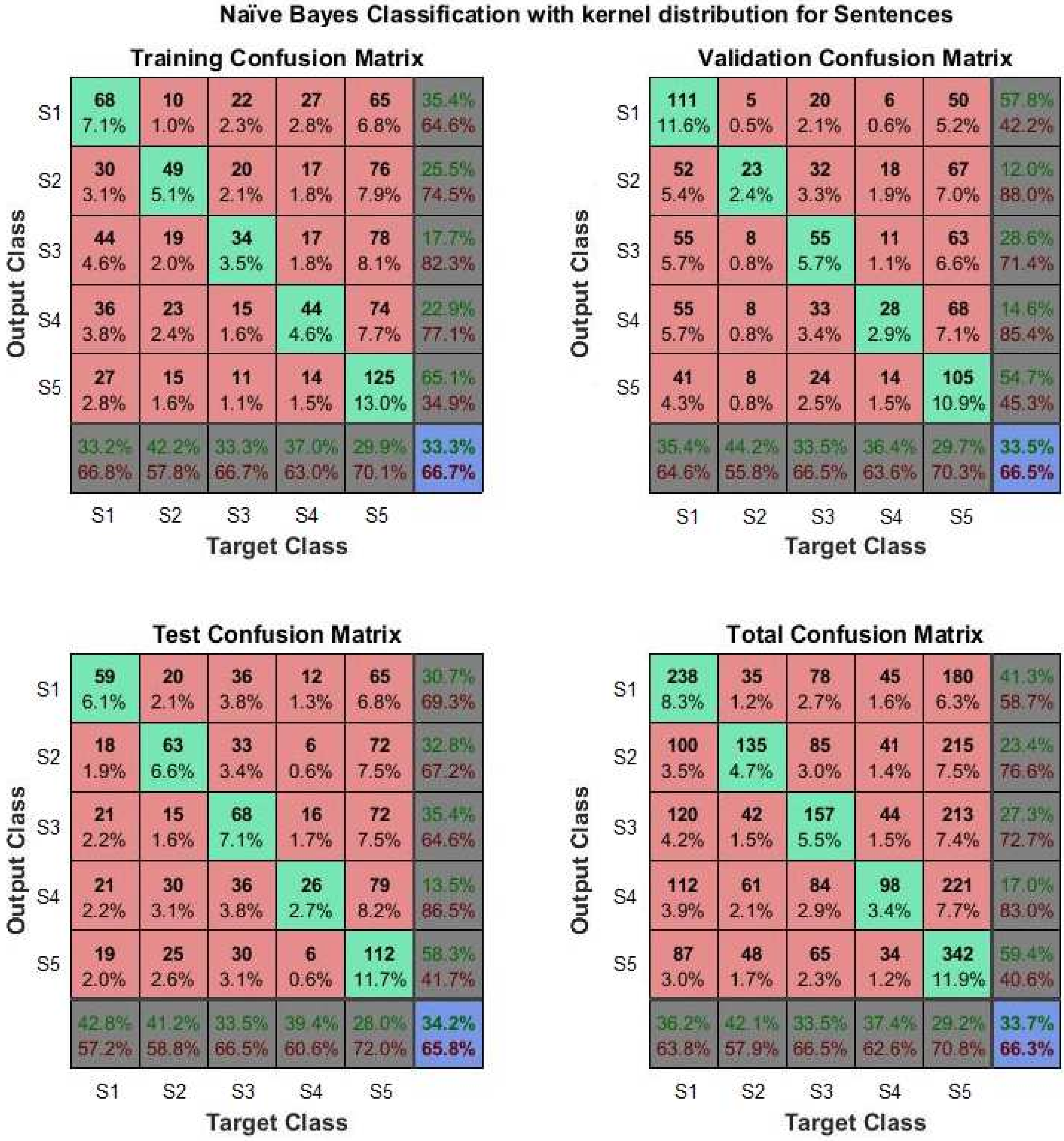


Figure 39 - Naïve Bayes Supervised Classifier with kernel distribution Confusion Matrixes and Accuracies for Sentences Task

The result for this supervised classifier with kernel distribution for Sentences task was not good, with total accuracy of 33,7% .

In Figure 40, the Naïve Bayes Supervised Classifier with MVMN distribution Confusion Matrixes and Accuracies, for Sentences task, are shown.

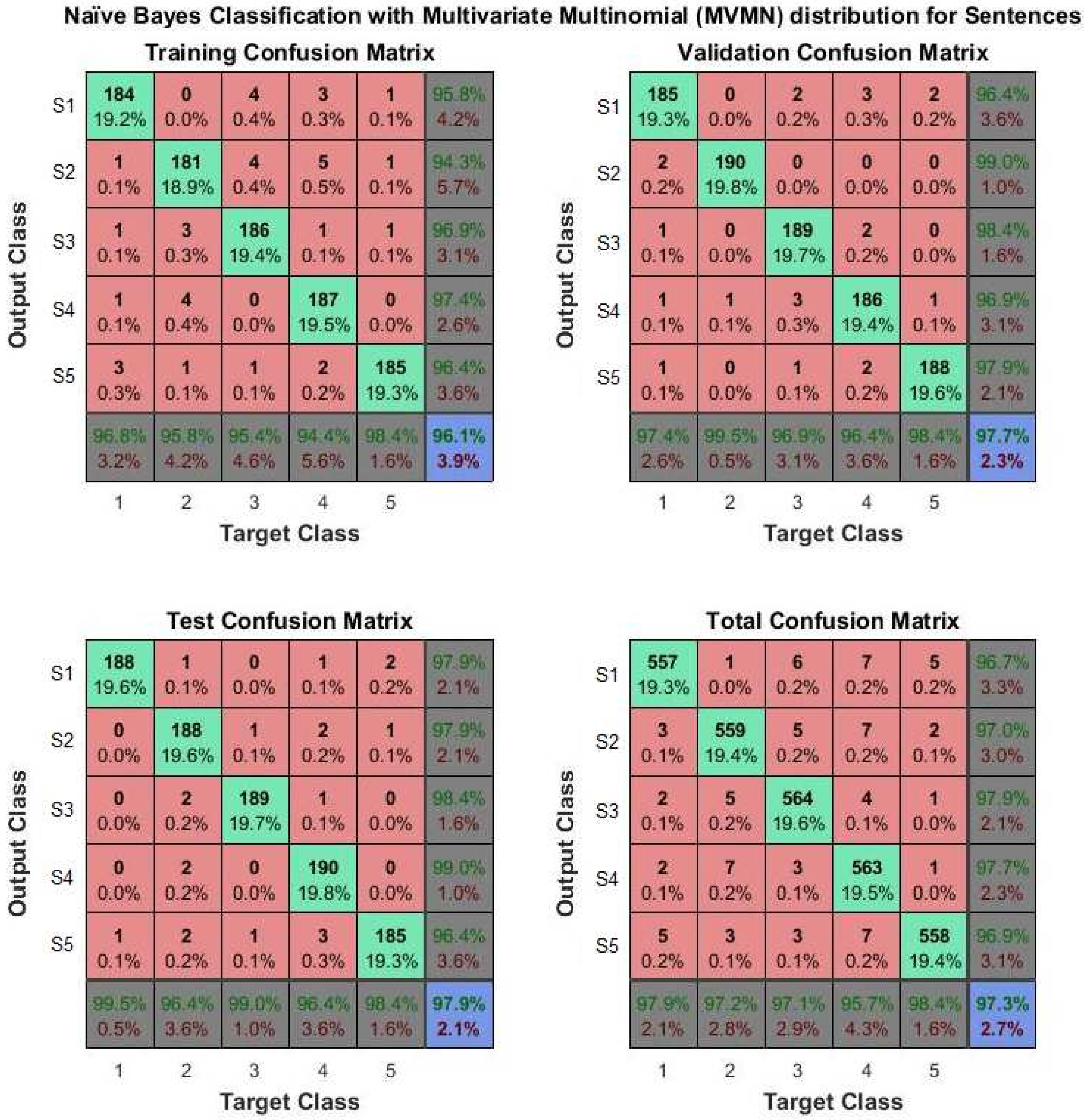


Figure 40 - Naïve Bayes Supervised Classifier with MVMN distribution Confusion Matrixes and Accuracies for Sentences Task

On the other hand, the Multivariate Multinomial distribution was excellent for Sentences task, with total accuracy achieving the value of 97,3%.

After that, the Naïve Bayes classifiers are used with the Words task data set. In Figure 41, the Naïve Bayes Supervised Classifier with MVMN distribution Confusion Matrixes and Accuracies, are shown.

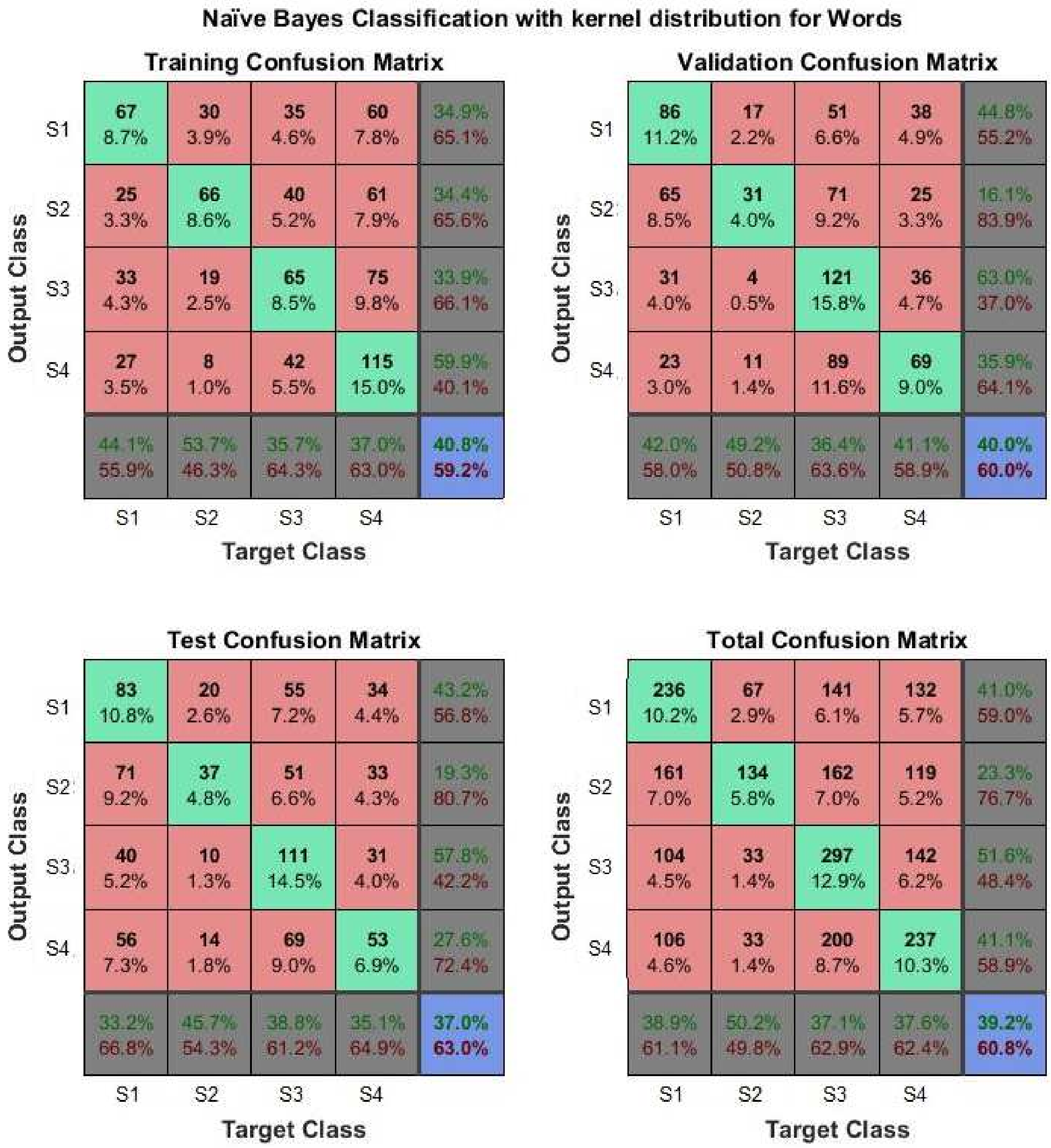


Figure 41 - Naïve Bayes Supervised Classifier with kernel distribution Confusion Matrixes and Accuracies for Words Task

The result for this supervised classifier with kernel distribution for Words task was not good, with total accuracy of 39,2%.

In Figure 42, the Naïve Bayes Supervised Classifier with MVMN distribution Confusion Matrixes and Accuracies, for Words task, are shown.

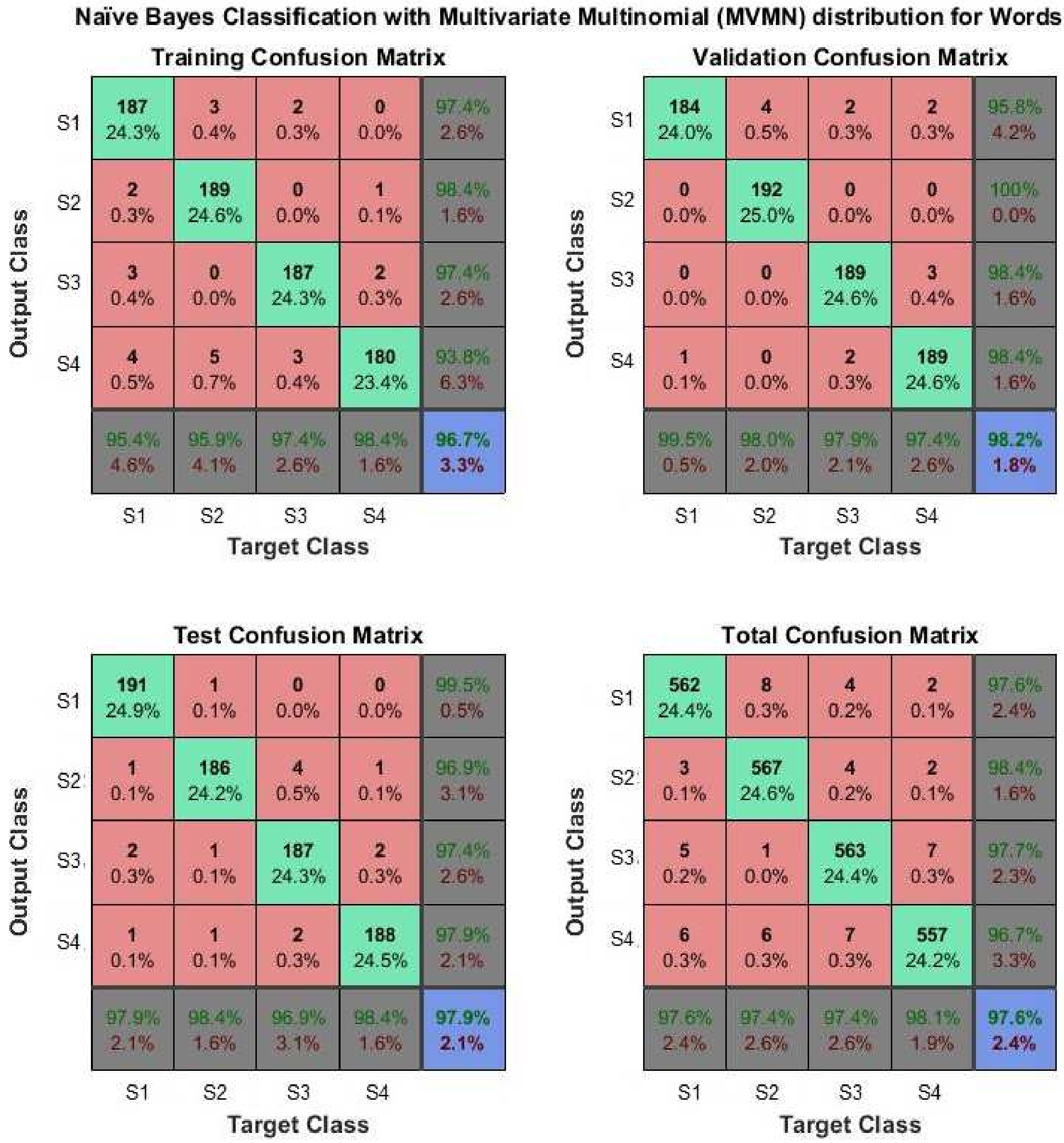


Figure 42 - Naïve Bayes Supervised Classifier with MVMN distribution Confusion Matrixes and Accuracies for Words Task

The Multivariate Multinomial distribution was also excellent for Words Task, with total accuracy achieving the value of 97,6%.

The Naïve Bayes supervised classifier achieved excellent results for the distribution function MVMN (multivariate multinomial distribution), achieving accuracy values higher than 97% for both tasks, indicating that this method is a good option to classify these datasets, but not perfect. An important point to highlight is that this MVMN function characterizes this classifier method as non-linear. Thus, another method was used, the Multiclass Support Vector Machine (SVM) Supervised Classifier.

IV.3.2 Multiclass Support Vector Machine Supervised Classifier

This supervised classifier used a Support Vector machine Template (SVM) for the algorithm Error-correcting output codes (ECOC). The main parameters observed during the experiments were Box Constraint, Kernel Function and Standardize.

Many tests were done changing the kernel distribution functions and parameters for both tasks and the best results were for the following configuration: “BoxConstraint” was 0.01; “KernelFunction” was “Gaussian”; and “Standardize” was “off”’, for both tasks. These best results will be shown in the sequence below.

In Figure 43, the SVM Supervised Classifier Confusion Matrixes and Accuracies for Sentences task are shown.

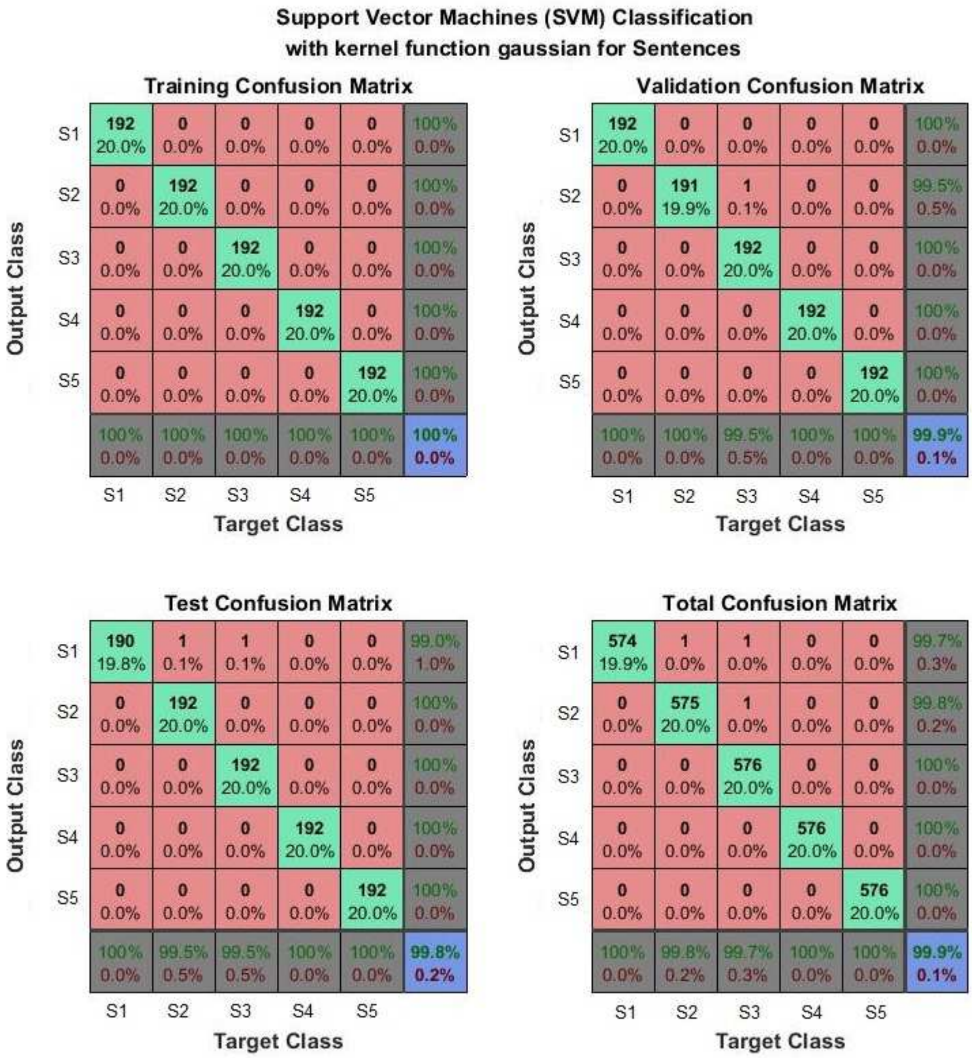


Figure 43 - SVM Supervised Classifier Confusion Matrixes and Accuracies for Sentences Task

For the Sentences task, the results are almost perfect and better than occurred in the Naïve Bayes method, with total accuracy achieving the value of 99,9%.

In Figure 44, the SVM Supervised Classifier Confusion Matrixes and Accuracies for Words task are shown.

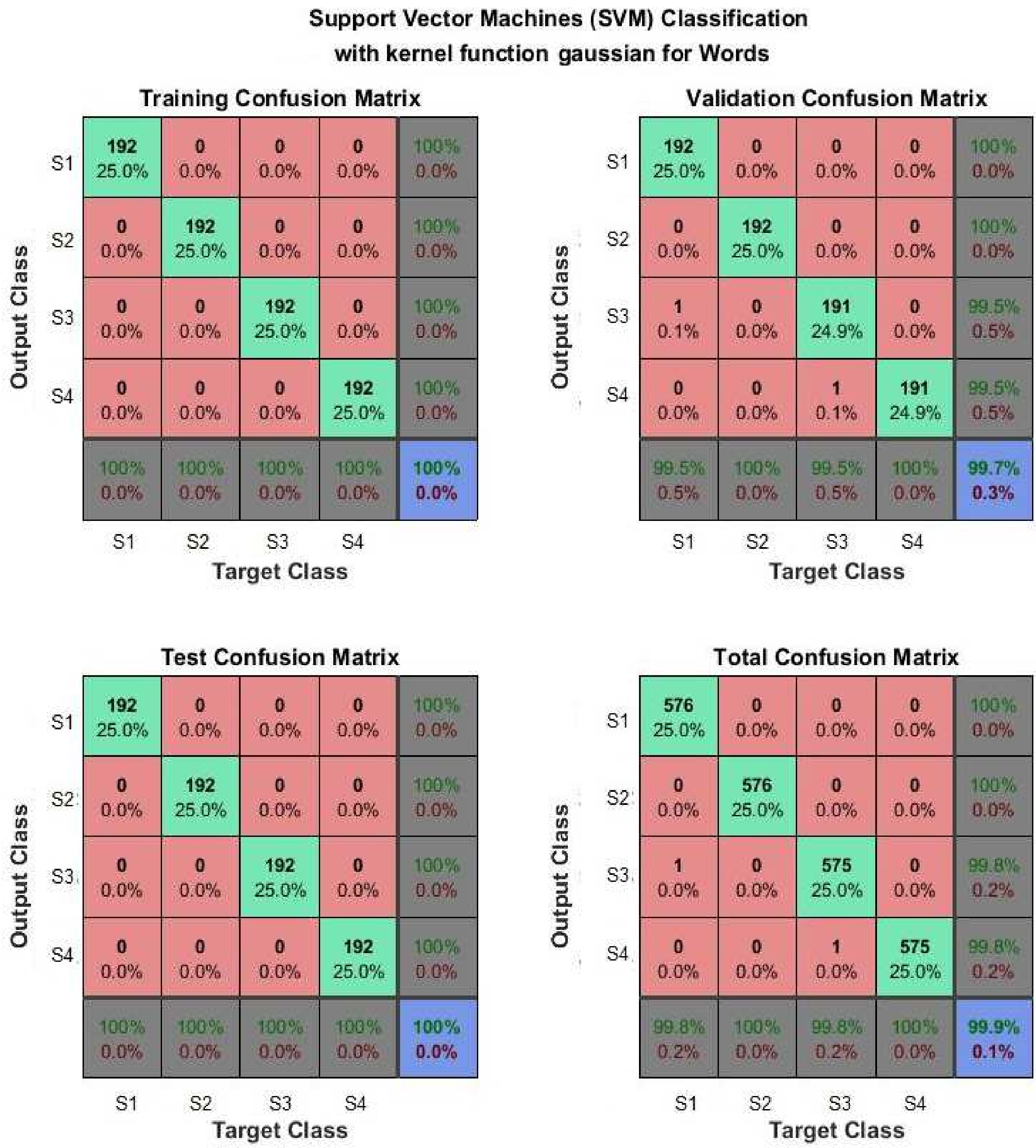


Figure 44 - SVM Supervised Classifier Confusion Matrixes and Accuracies for Words Task

The results are also almost perfect and better than occurred in the Naïve Bayes method, with total accuracy achieving the value of 99,9%, for the Words task.

The Multiclass SVM supervised classifier achieved excellent results, better than the result obtained by the Naïve Bayes method, for both tasks. As the kernel function used is Gaussian, these results are strong indication that the best supervised classification approaches for these datasets are non-linear methods.

IV.3.3 Neural Network Supervised Classifier

For this supervised classifier was used the Matlab®'s Neural Network Toolbox™. For this test were also executed several configurations to achieve good results, but as the number of iterations was variable and not stable for each attempt, the values of accuracies for each one were very near and not good in comparison with Naïve Bayes and SVM.

The Neural Network Toolbox™ parameters' configuration for Neural Network Supervised Classifier for Sentences Task is the following:

a) Number of hidden layers (“hiddenLayerSize”): 10;
b) Neural Network Input Processing Function (“net.input.processFcns”): 'removeconstantrows','mapstd'
c) Neural Network Output Processing Function(“net.output.processFcns”): 'removeconstantrows','mapstd'
d) Setup Division of Data for Training, Validation, Testing (“net.divideFcn“): 'dividerand'
e) Train Ratio ("net.divideParam.trainRatio") = 1/3
f) Validation Ratio ("net.divideParam.valRatio") = 1/3;
g) Test Ratio ("net.divideParam.testRatio") = 1/3
h) Divide Mode (“net.divideMode”): 'sample';
i) Multilayer Neural Network Training Function (“net.trainFcn”):'trainrp'; and
j) Neural Network Performance Function (“net.performFcn”):'mse'

In Figure 45, the best attempt for Neural Network Supervised Classifier confusion matrixes and Accuracies for Sentences Task are shown below.

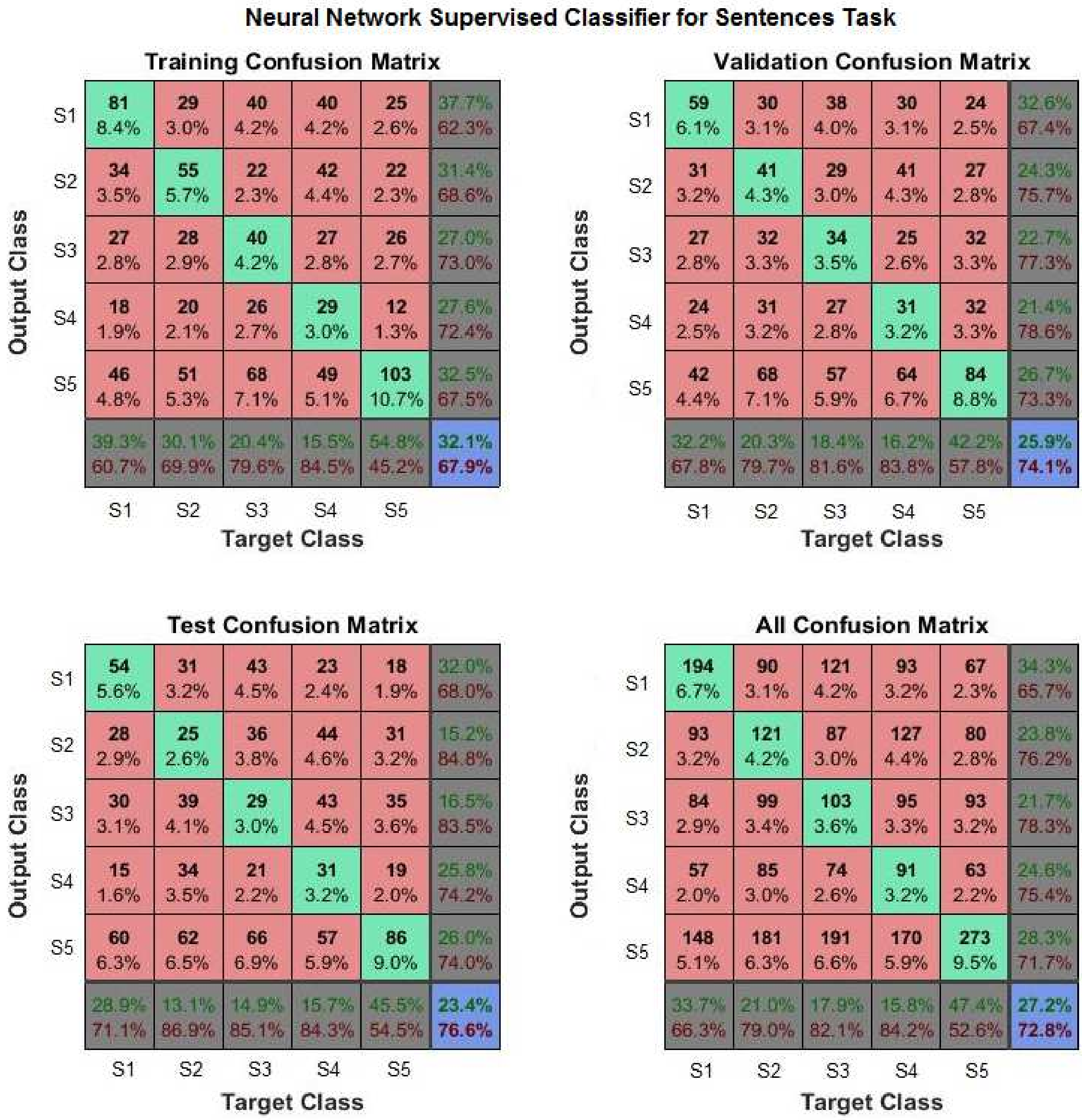


Figure 45 - Neural Network Supervised Classifier Confusion Matrixes and Accuracies for Sentences Task

The result for this supervised classifier for Sentences task was not good, with total accuracy of 27,2%.

The Neural Network Toolbox™ parameters' configuration for the best attempr for Neural Network Supervised Classifier for Words task is shown below.

a) Number of hidden layers ("hiddenLayerSize"): 60;
b) Neural Network Input Processing Function("net.input.processFcns"): 'removeconstantrows','mapstd';

c) Neural Network Output Processing Function("net.output.processFcns"): 'removeconstantrows','mapstd'

d) Setup Division of Data for Training, Validation, Testing ("net.divideFcn"): 'dividerand';

e) Train Ratio ("net.divideParam.trainRatio") = 1/3;

f) Validation Ratio("net.divideParam.valRatio") = 1/3;

g) Test Ratio ("net.divideParam.testRatio") = 1/3;

h) Divide Mode ("net.divideMode"): 'sample';

i) Multilayer Neural Network Training Function ("net.trainFcn"):'trainscg'; and

j) Neural Network Performance Function ("net.performFcn"):'mse'.

In Figure 46, the Neural Network Supervised Classifier confusion matrixes and Accuracies for Words task are shown.

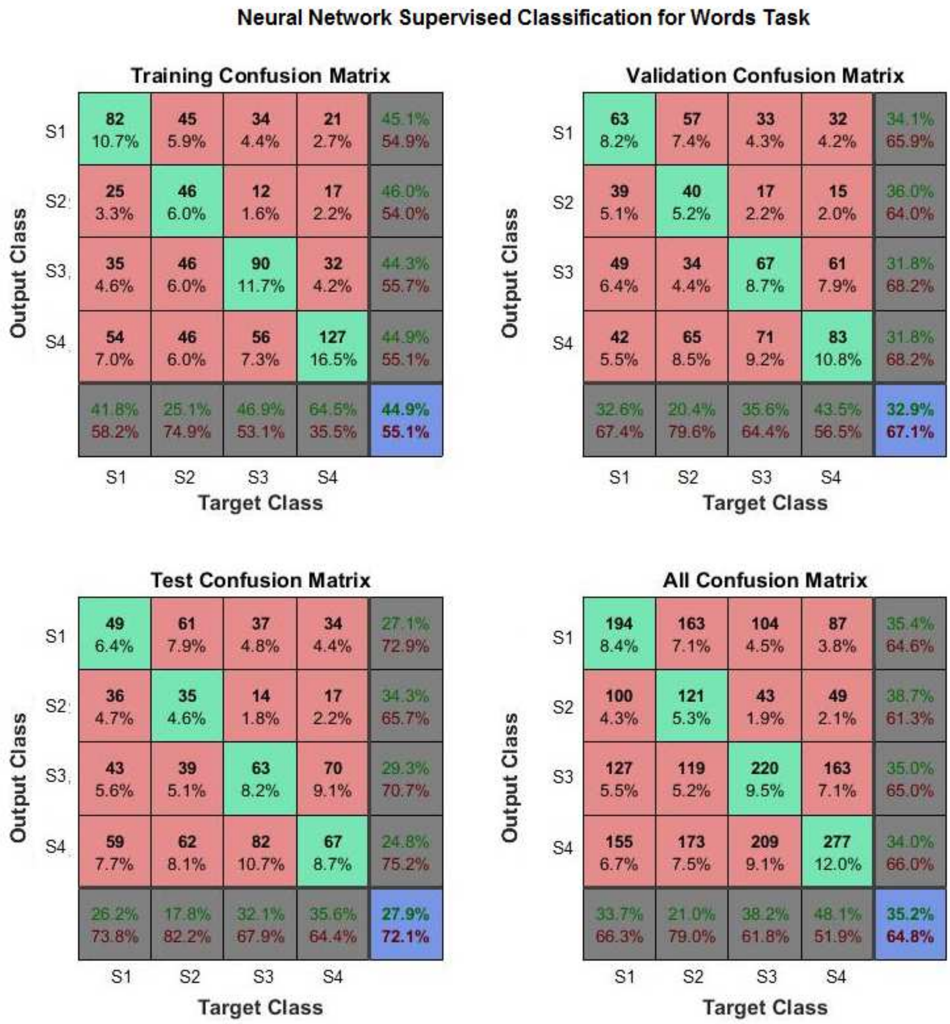


Figure 46 - Neural Network Supervised Classifier Confusion Matrixes and Accuracies for Words Task

The result for this supervised classifier for Sentences task was also not good, with total accuracy of 35,2%.

The results for this Neural Network supervised classifier for both tasks were not good, with total accuracy of 27,2% for the Sentences Task and 35,2% for the Words Task, indicating that this method is not appropriate for this kind of datasets.

Although the results for Naïve Bayes and SVM were excellent, the last supervised classifier, Random Forest, will be explained in the next topic.

IV.3.4 Random Forest Supervised Classifier

Following the Webb (2002) methodology proposed for Supervised Classifiers, finally was used the Random Forest with the Matlab® function “fitensemble”. The simulations were done changing the parameters of this function. The best results were achieved using the following Matlab® function “fitensemble” parameters:

a) 'method' (Ensemble-aggregation method): “bag”;

b) 'NLearn' (Number of ensemble learning cycles): 100;

c) 'Learners' (Weak learners to use in ensemble): “Tree”; and

d) 'Type' (Supervised learning type): 'classification'.

In Figure 47 is shown the Random Forest Supervised Classifier Confusion Matrixes and Accuracies for Sentences Task.

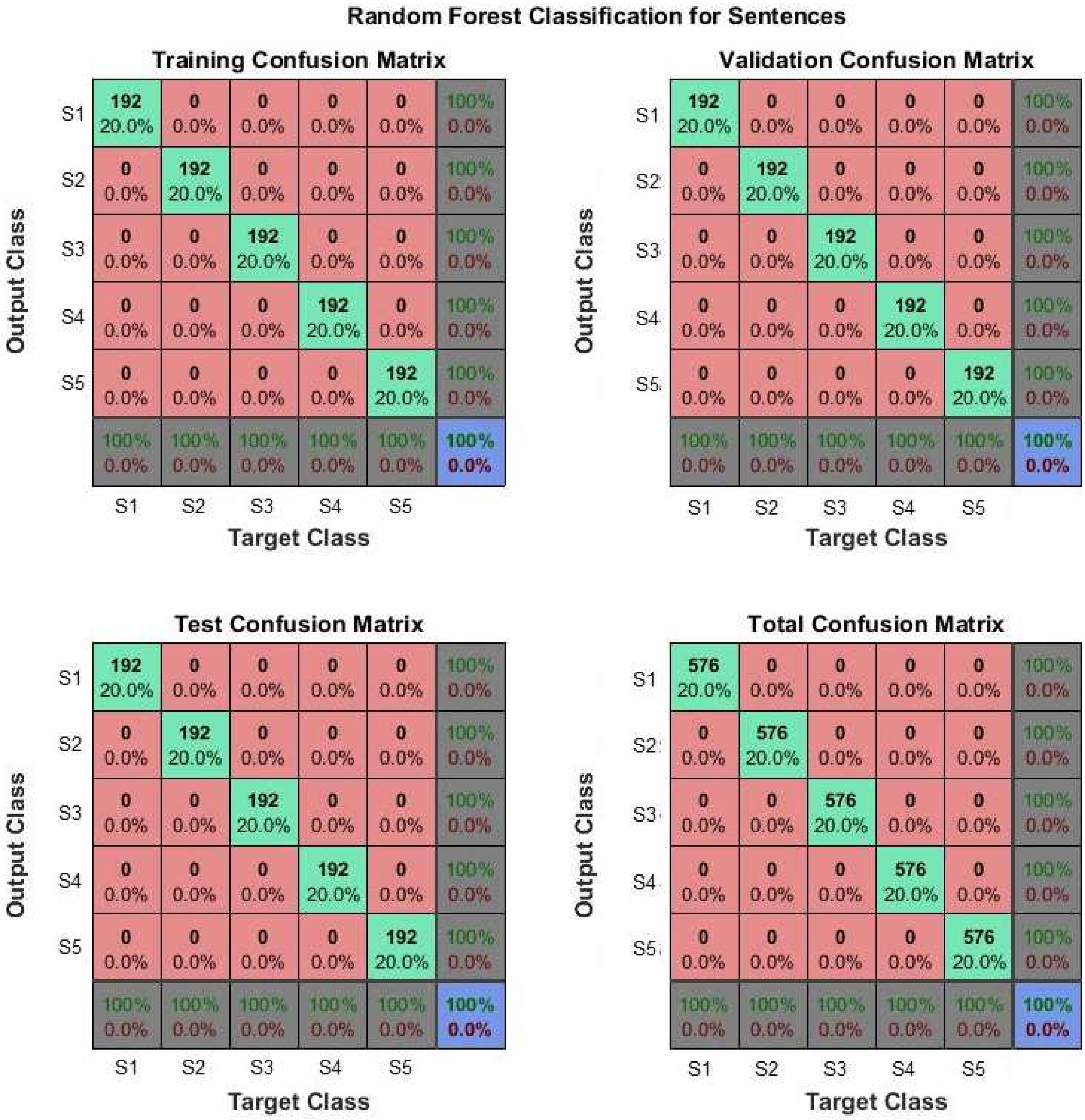


Figure 47 - Random Forest Supervised Classifier Confusion Matrixes and Accuracies for Sentences Task

The results were perfect and better than occurred in the supervised classifiers with Naïve Bayes and Multiclass SVM methods, with total accuracy achieving the value of 100,0%, for the Sentences task.

In Figure 48, the Random Forest Supervised Classifier Confusion Matrixes and Accuracies for Words task are shown.

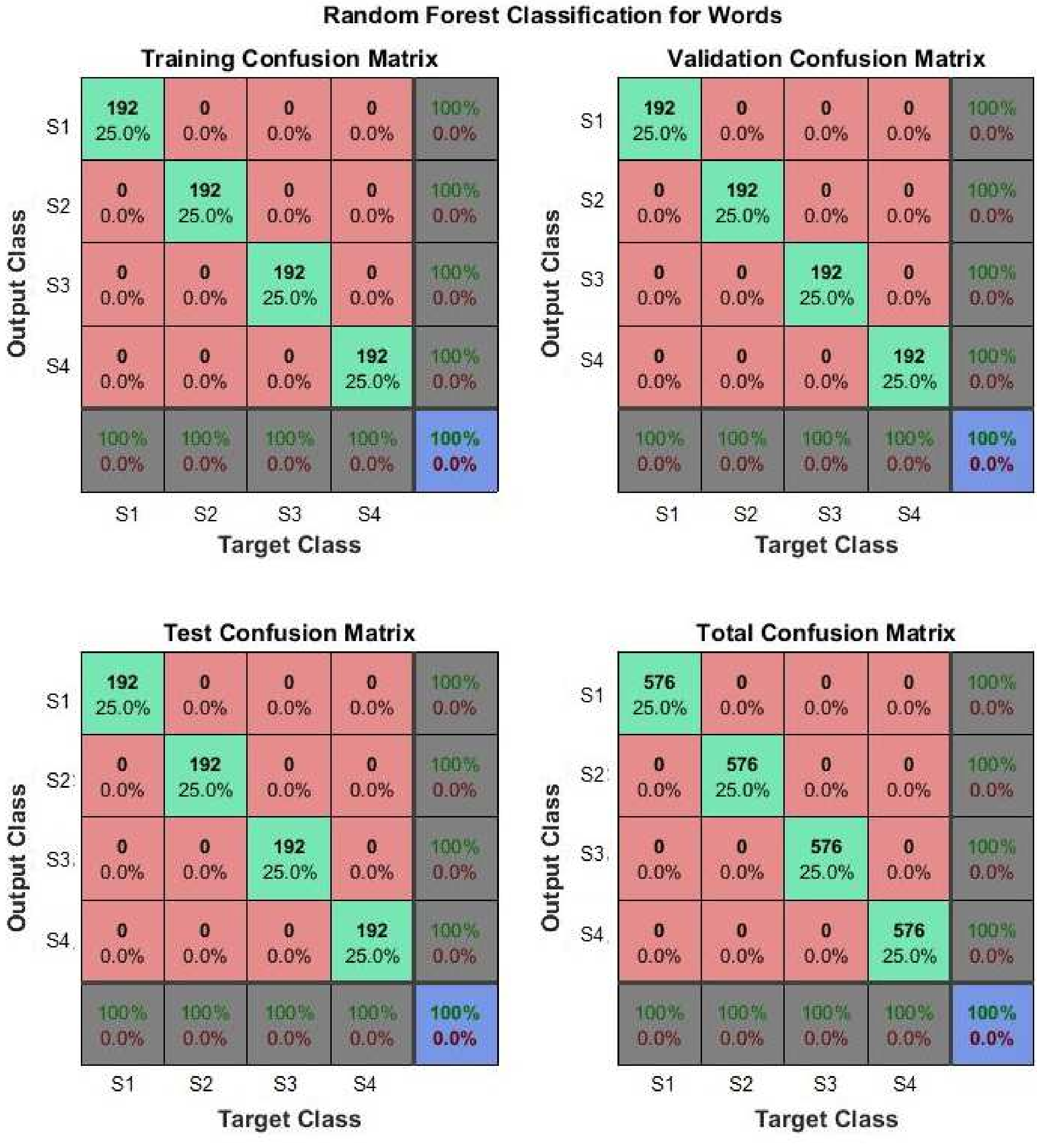


Figure 48 - Random Forest Supervised Classifier Confusion Matrixes and Accuracies for Words Task

The results were also perfect, with total accuracy achieving the value of 100,0%, for the Sentences task.

The results were perfect and better than occurred in the Naïve Bayes and Multiclass SVM supervised classifiers methods, with total accuracy achieving the value of 100,0%, for both tasks, indicating that the method Random Forest was the best and more apprppriate classification scenario for these datasets, and also, a non-linear classifier.

IV.4. Summarized Results

The results obtained are summarized in the Table 6, for unsupervised classifiers, and Table 7, for supervised classifiers.

Table 6 - Unsupervised Classifiers Results Summary

Classifier	Accuracies for Sentences Task	Observation concerning the Parameters used	Accuracies for Words Task	Observation concerning the Parameters used
Hierarchical Clustering	21,63 %	“pdist” metric “cityblock” with a “linkage” method “average” “pdist” metric “cityblock” with a “linkage” method “centroid”	28,21%	“pdist” metric “spearman” with a “linkage” method “single”
K-means	52,92 %	2 clusters with k-means metric “cityblock”	48,44 %	2 clusters with k-means metric “cityblock”
	19,44 %	5 clusters with k-means metric “cityblock”	23,26 %	4 clusters with k-means metric “cityblock”
	53,33 %	2 clusters with k-means metric “sqEuclidean”	32,68 %	3 clusters with k-means metric “sqEuclidean”
	18,13 %	5 clusters with k-means metric “sqEuclidean”	25,00 %	4 clusters with k-means metric “sqEuclidean”
Gaussian Mixture Models	19,24 %	Features used: Mean Amplitude Between two fixed latencies and Peak Amplitude	24,91 %	Features used: Mean Amplitude Between two fixed latencies and Peak Amplitude
	21,18 %	Features used: Mean Amplitude Between two fixed latencies and Peak Latency	24,78 %	Features used: Mean Amplitude Between two fixed latencies and Peak Latency
	19,24 %	Features used: Peak Amplitude and Peak Latency	24,39 %	Features used: Peak Amplitude and Peak Latency

Table 7 - Supervised Classifiers Results Summary

Classifier	Accuracies for Sentences Task	Observation concerning the Parameters used	Accuracies for Words Task	Observation concerning the Parameters used
Naïve Bayes	33,7 %	Distribuiton function “kernel”	39,2 %	Distribuiton function “kernel”
	97,3 %	Distribuiton function “MVMN”	97,6 %	Distribuiton function “MVMN”
Multiclass Support Vector Machine	99,9 %	“BoxConstraint”:0.01 “KernelFunction”:“Gaussian” “Standardize”:“off”’	99,9 %	“BoxConstraint”:0.01 “KernelFunction”:“Gaussian” “Standardize”:“off”’

Neural Network	27,2 %	a) Number of hidden layers (“hiddenLayerSize”): 10
		b) Neural Network Input Processing Function (“net.input.processFcns”): 'removeconstantrows','mapstd'
		c) Neural Network Output Processing Function(“net.output.processFcns”): 'removeconstantrows','mapstd'
		d) Setup Division of Data for Training, Validation, Testing (“net.divideFcn“): 'dividerand'
		e) Train Ratio ("net.divideParam.trainRatio") = 1/3
		f) Validation Ratio ("net.divideParam.valRatio") = 1/3;
		g) Test Ratio ("net.divideParam.testRatio") = 1/3
		h) Divide Mode (“net.divideMode”): 'sample'
		i) Multilayer Neural Network Training Function (“net.trainFcn”):'trainrp'
		j) Neural Network Performance Function (“net.performFcn”):'mse'
			35,2 %	a) Number of hidden layers (“hiddenLayerSize”): 60
				b) Neural Network Input Processing Function(“net.input.processFcns”): 'removeconstantrows','mapstd'
				c) Neural Network Output Processing Function(“net.output.processFcns”): 'removeconstantrows','mapstd'
				d) Setup Division of Data for Training, Validation, Testing (“net.divideFcn“): 'dividerand'
				e) Train Ratio ("net.divideParam.trainRatio") = 1/3;
				f) Validation Ratio ("net.divideParam.valRatio") = 1/3;
				g) Test Ratio ("net.divideParam.testRatio") = 1/3;
				h) Divide Mode (“net.divideMode”): 'sample';
				i) Multilayer Neural Network Training Function (“net.trainFcn”):'trainscg'
				j) Neural Network Performance Function (“net.performFcn”):'mse'

Random Forest	100,0 %	a) 'method' (Ensemble-aggregation method): “bag” b) 'NLearn' (Number of ensemble learning cycles): 100 c) 'Learners' (Weak learners to use in ensemble): “Tree” d) 'Type' (Supervised learning type): 'classification'	100,0 %	a) 'method' (Ensemble-aggregation method): “bag” b) 'NLearn' (Number of ensemble learning cycles): 100 c) 'Learners' (Weak learners to use in ensemble): “Tree” d) 'Type' (Supervised learning type): 'classification'

IV.5 Additional Tests

The test campaign considered all complete datasets for Sentences and Words Task. At the end of the proposed methodology, in order to deepen the investigation of the features used in classification, specifically concerning the subjects (the persons subjecte to the experiment), classification tests without split were performed, with the datasets for all supervised classifiers and for Random Forest method, for both tasks, but doing the classification campaign, without retrainning, running again the algorithms with only the data for each subject.

Since the behavior of the brain between individuals may be quite different (although the profile of the people who participate in the experiments are similar), the objective is to check if the results for individuals can be different in relation with the complete datasets. The results can be seen in the Table 8, for the Sentences Task, and Table 9, for the Words Task.

Table 8 - Classifiers Results for Individuals - Sentences Task

Applying Discrmination (Supervised Classifiers)						Regression method	
NAÏVE BAYES MVMN		SVM		Neural Network		Random Forest	
Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %
2	100.00%	2	100.00%	2	87,80%	2	100.00%
3	98.89%	3	99.44%	3	23,90%	3	100.00%
4	100.00%	4	99.44%	4	32,20%	4	100.00%
5	100.00%	5	100.00%	5	52,80%	5	100.00%
6	100.00%	6	100.00%	6	34,40%	6	100.00%
7	100.00%	7	100.00%	7	20,00%	7	100.00%
9	99.44%	9	100.00%	9	44,40%	9	100.00%
10	100.00%	10	100.00%	10	23,30%	10	100.00%
13	100.00%	13	100.00%	13	33,90%	13	100.00%
15	100.00%	15	100.00%	15	19,40%	15	99.44%
16	100.00%	16	100.00%	16	45,00%	16	100.00%
17	99.44%	17	100.00%	17	25,60%	17	100.00%
18	100.00%	18	100.00%	18	43,30%	18	100.00%
19	100.00%	19	100.00%	19	61,70%	19	100.00%
20	100.00%	20	100.00%	20	36,10%	20	100.00%
21	100.00%	21	100.00%	21	29,40%	21	100.00%
Classifier accuracy with complete dataset %	97.26%	Classifier accuracy with complete dataset %	99.90%	Classifier accuracy with complete dataset %	27,20%	Classifier accuracy with complete dataset %	100.00%

Table 9 - Classifiers Results for Individuals - WordsTask

Applying Discrmination (Supervised Classifiers)						Regression method	
NAÏVE BAYES MVMN		SVM		Neural Network		Random Forest	
Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %	Subject	Accuracy %
2	100.00%	2	100.00%	2	26,40%	2	100.00%
3	100.00%	3	99.31%	3	34,70%	3	100.00%
4	100.00%	4	99.31%	4	37,50%	4	100.00%
5	100.00%	5	100.00%	5	25,70%	5	100.00%
6	99.31%	6	100.00%	6	34,70%	6	100.00%
7	100.00%	7	100.00%	7	30,60%	7	100.00%
9	100.00%	9	100.00%	9	50,00%	9	100.00%
10	100.00%	10	100.00%	10	28,50%	10	100.00%
13	100.00%	13	99.31%	13	36,10%	13	100.00%
15	100.00%	15	100.00%	15	25,00%	15	100.00%
16	100.00%	16	100.00%	16	27,80%	16	100.00%
17	100.00%	17	100.00%	17	34,70%	17	100.00%
18	100.00%	18	100.00%	18	52,10%	18	100.00%
19	100.00%	19	100.00%	19	32,60%	19	100.00%
20	100.00%	20	100.00%	20	41,00%	20	100.00%
21	100.00%	21	100.00%	21	21,50%	21	100.00%
Classifier accuracy with complete dataset %	97,60%	Classifier accuracy with complete dataset %	99,90%	Classifier accuracy with complete dataset %	35,20%	Classifier accuracy with complete dataset %	100.00%

The results indicated that the all individuals, independently, achieved 100% of accuracy for the Random Forest classifier, except for one individual (subject 15) in the Sentence Task that achieved 99,44%. For Naîve Bayes and SVM a few subjects (1 or 2) presented different accuracies but, even so, with values higher than 98%. The worst case was the Neural Network classifier for which all individuals presented different and low accuracies.

Chapter V - Conclusions and Final Considerations

The objective of this work that was to investigate Webb's (2002) pattern recognition methodology in ERP results from Soto's (2014) data experiment to classify correctly different patterns was considered as achieved.

The software tools EEGLAB®, ERPLAB® and Matlab® perform the extraction and treatment of the focused EEG data and the pattern recognition algorithms proposed properly. As demonstrated in this dissertation simulations and results, the "clustering and unsupervised classification" is not appropriate for both tasks. On the other hand, the proposed methodology by Webb (2002) allowed us to obtain good results to support the goal of this work with excellent results for supervised classification.

The Random Forest classification scenario was the best supervised classifier method for these data sets which a total accuracy of 100%. Other good results are achieved with the supervised classifiers SVM and Naïve Bayes, with total accuracies higher than 96%. These results also indicated that for these ERP datasets, for both Sentences and Words Tasks, non-linear approaches were more suitable to classify the data from Soto (2014) experiment configuration. This result is valid for each subject and also for group of subjects.

Even with these good results, the next step is to continue the studies in relation to the analysis of the classifiers proposed with the separation of the other features, especially the ROI and the ERP time range features. Maybe they cannot only allow a more specific way to classify the data, but also could indicate which ERP features can be more influent in the classification process.

Other methods of classification and, especially, for the clustering and unsupervised classification shall be considered in order to promote the advance in this dataset study not only for pattern recognition, but also for their use as a possible method for neuro linguistics and medicine areas.

Chapter VI - Bibliographic References

BREIMAN, Leo. **Random Forests**. Machine Learning. October 2001, Volume 45, Issue 1, pp 5-32. doi:10.1023/A:1010933404324.

CONG, Fengyu, RISTANIEMI, Tapani, LYYTINEN, Heikki, **Advanced Signal Processing on Brain Event-Related Potentials: Filtering ERPs in Time, Frequency and Space Domains Sequentially and Simultaneously**, Singapore, Ed. World scientific Publlishing, 2015, ISBN 978-981-4623-08-7.

EEGLAB®. **EEGLAB® Tutorial**. Available on: <http://sccn.ucsd.edu/wiki/Getting_Started>, accessed in April, 15th,2016.

ERPLAB®. **ERPInfo-ERPLAB® Toolbox**. Available on:<http://www.erpinfo.org/erplab.html>, consulted on April, 15th,2016.

GESUALDI, Aline da Rocha; FRANÇA, Aniela Improta. **Event-related brain potentials (ERP): an overview**. Revista Linguística / Revista do Programa de Pós-Graduação em Linguística da Universidade Federal do Rio de Janeiro. Volume 7, número 2, dezembro de 2011. ISSN 1808-835X 1. [http://www.letras.ufrj.br/poslinguistica/revistalinguistica]

HANDY, T. C. **Event Related Potentials: A Methods Handbook**. Cambridge, MA: Bradford/MIT Press., 2005.

HASTIE, T., TIBSHIRANI, R., and FRIEDMAN, J. **The Elements of Statistical Learning.** Second Edition. NY: Springer, 2008.

KAMEL, Nidal, MALIK, Aamir Saeed, **EEG/ERP analysis: methods and apllications**, Boca Raton/FL, USA, Ed. CRC Press, 2015. ISBN 978-1-4822-2469-6.

LOPEZ-CALDERÓN, Javier, LUCK, Styeven J. **ERPLAB: an open-source toolbox for the analysis of event-related potentials.** Frontiers in Human Neuroscience. Volume 8. Article 213. p. 2-14. April 2014.

LOUPPE, Gilles. **Understanding Random Forests: From Theory to Practice**. Doctoral thesis. Université de Liège. 2014.

LUCK, Steven J., **An Introduction to the Event-Related Potential Technique**, Massachusetts Institute of Technology MIT Press books, 2nd Ed., 2014, ISBN 978-0-262-52585-5

MATLAB®. **MATLAB - The Language of Technical Computing**. Available on: <https://www.mathworks.com/products/matlab.html>, consulted on April, 15th, 2016a.

MATLAB®. **Hierarchical Clustering**. Available on: <https://www.mathworks.com/help/stats/hierarchical-clustering.html>, consulted on April, 15th, 2016b.

MATLAB®. **pdist - Pairwise distance between pairs of objects**. Available on: <https://www.mathworks.com/help/stats/pdist.html>, consulted on April, 15th, 2016c.

MATLAB®. **linkage - Agglomerative hierarchical cluster tree.** Available on: <https://www.mathworks.com/help/stats/linkage.html>, consulted on April, 15th, 2016d.

MATLAB®. **k-Means Clustering**. Available on: <https://www.mathworks.com/help/stats/k-means-clustering.html>, consulted on April, 15th, 2016e.

MATLAB®. **k-means.** Available on: <https://www.mathworks.com/help/stats/kmeans.html>, consulted on April, 15th, 2016f.

MATLAB®. **Clustering Using Gaussian Mixture Models**. Available on: <https://www.mathworks.com/help/stats/clustering-using-gaussian-mixture-models.html>, consulted on April, 15th, 2016g.

MATLAB®. **Cluster Data from Mixture of Gaussian Distributions**. Available on: <https://www.mathworks.com/help/stats/cluster-data-from-mixture-of-gaussian-distributions.html>, consulted on April, 15th, 2016h.

MATLAB®. **Naive Bayes Classification**. Available on: <https://www.mathworks.com/help/stats/naive-bayes-classification.html>, consulted on April, 15th, 2016i.

MATLAB®. **Train multiclass naive Bayes model**. Available on: <https://www.mathworks.com/help/stats/fitcnb.html>, consulted on April, 15th, 2016j.

MATLAB®. **Support Vector Machines for Binary Classification**. Available on: <https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>, consulted on April, 15th, 2016k.

MATLAB®. **ClassificationECOC class - Multiclass model for support vector machines or other classifiers**. Available on: <https://www.mathworks.com/help/stats/classificationecoc-class.html>, consulted on April, 15th, 2016l.

MATLAB®. **templateSVM - Support vector machine template**. Available on: <https://www.mathworks.com/help/stats/templatesvm.html>, consulted on April, 15th, 2016m.

MATLAB®. **Neural Network Toolbox**. Available on: <https://www.mathworks.com/products/neural-network.html>, consulted on April, 15th, 2016n.

MATLAB®. **Neural Network Object Properties**. Available on: <https://www.mathworks.com/help/nnet/ug/neural-network-object-properties.html>, consulted on April, 15th, 2016o.

MATLAB®. **Choose Neural Network Input-Output Processing Functions**. Available on: <https://www.mathworks.com/help/nnet/ug/choose-neural-network-input-output-processing-functions.html>, consulted on April, 15th, 2016p.

MATLAB®. **Divide Data for Optimal Neural Network Training**. Available on: <https://www.mathworks.com/help/nnet/ug/divide-data-for-optimal-neural-network-training.html>, consulted on April, 15th, 2016q.

MATLAB®. **Ensemble Methods**. Available on: <https://www.mathworks.com/help/stats/ensemble-methods.html>, consulted on April, 15th, 2016r.

MATLAB®. **Fit ensemble of learners for classification and regression**. Available on: <https://www.mathworks.com/help/stats/fitensemble.html>, consulted on April, 15th, 2016s.

ROKACH, Lior, and ODED, Maimon. **Clustering methods. Data mining and knowledge discovery handbook**. Springer US, 2005. 321-352.

SOTO, Marije, **ERP and fMRI Evidence of Compositional Differences between Linguistic Computations for Words and Sentences**. Marije Soto - Rio de Janeiro:UFRJ./Faculdade de Letras, 2014.

STEHMAN, Stephen V. **Selecting and interpreting measures of thematic classification accuracy**. Remote Sensing of Environment. 62 (1): 77-89. 1997. doi:10.1016/S0034-4257(97)00083-7.

THEODORIDIS, Segios, KOUTROMBAS, Konstantinos, **Pattern Recogniton**, 4th Edition, Academic Press, Elsevier Inc, .2009, ISBN: 978-1-59749-272-0

WEBB, Andrew R., **Statistical Pattern Recognition**, Second Edition. John Wiley & Sons, Ltd. 2012. ISBNs: 0-470-84513-9 (HB); 0-470-84514-7 (PB)

WIKIPEDIA, **Components of ERP**. Available on: <https://en.wikipedia.org/wiki/Event-related_potential#/media/File:ComponentsofERP.svg>, consulted on April, 15th,2016.

WOODMAN, Geoffrey F.. **A Brief Introduction to the Use of Event-Related Potentials (ERPs) in Studies of Perception and Attention**. Atten Percept Psychophys. 2010 November ; 72(8): . doi:10.3758/APP.72.8.2031.

Appendix A - EEGLAB® and ERPLAB® steps for subjects dataset creation

a) Reading of the EEG header file vhdr

a.1) With MATLAB program has already been runned, it is necessary to start the EEGLAB® program by the command “eeglab” as shown in figure A.1.

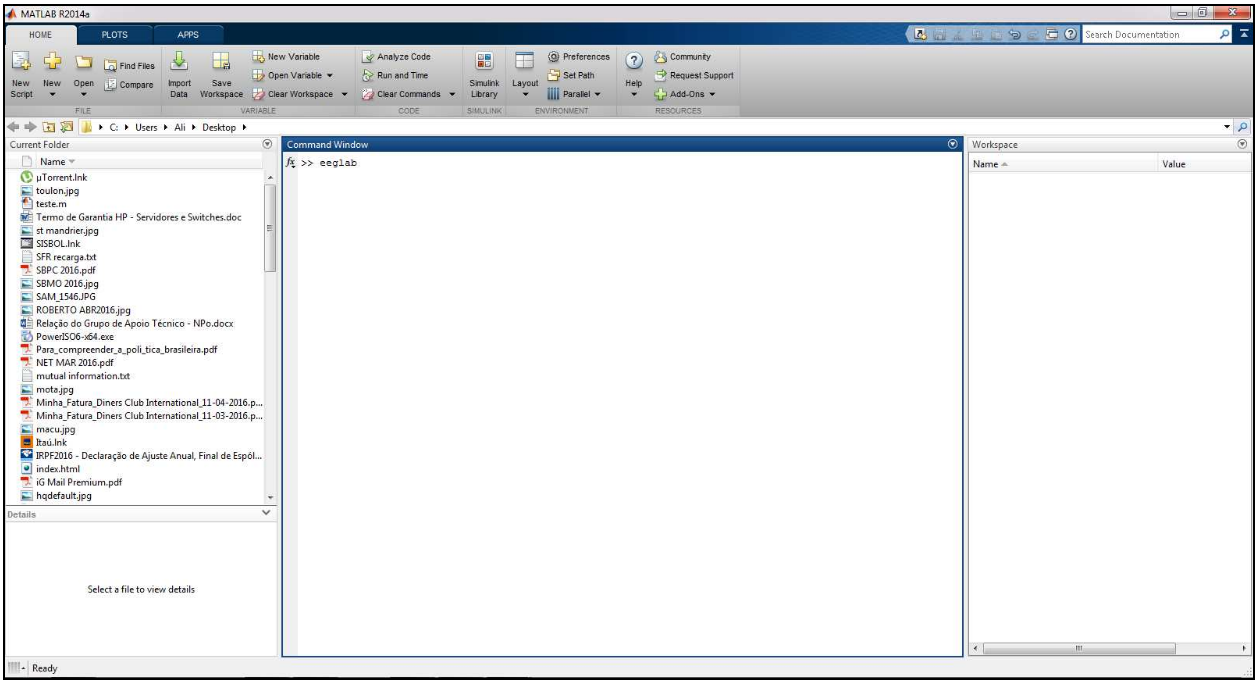


Figure A.1 - Initializing the program EEGLAB® in the MATLAB®

a.2) After the EEGLAB® window open, access the reading file by the following command tab sequence: “File/Import Data/Using the FILE-IO Interface”, as shown in figure A.2.

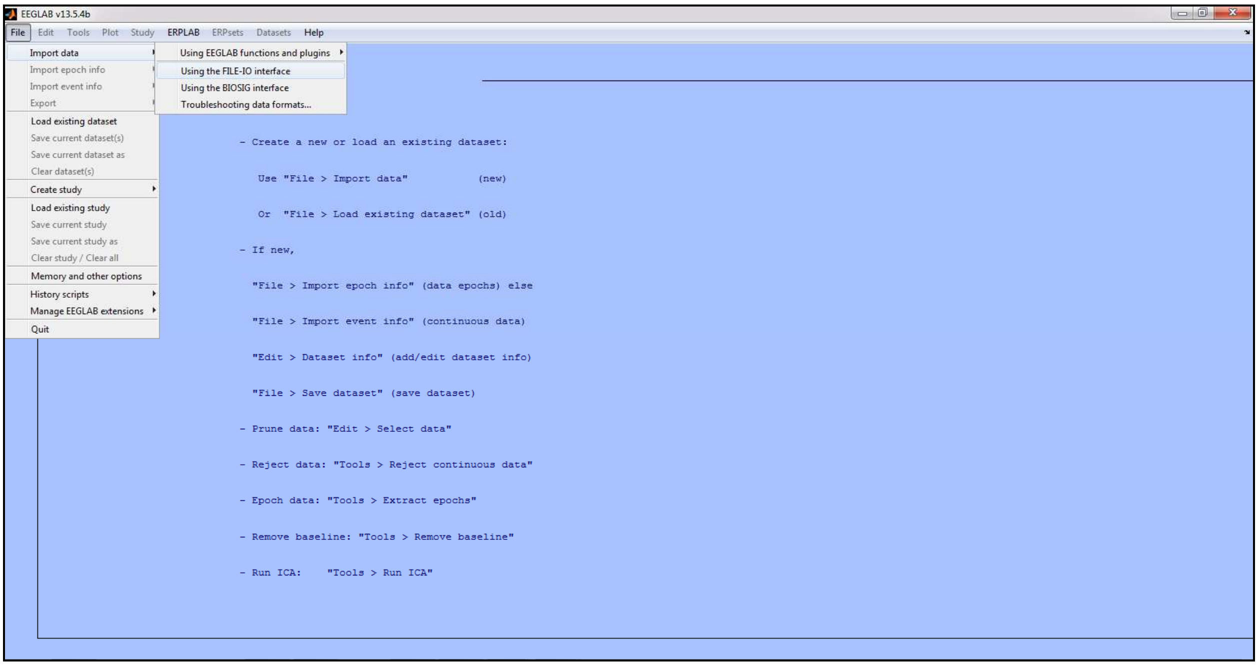


Figure A.2 - Opening the EEG header file “vhdr” in the EEGLAB®

a.3) In the EEGLAB® “FILE-IO Import” window, press the button “File” as shown in figure A.3.

Figure A.3 - EEGLAB® “FILE-IO Import” window

a.4) The EEGLAB® window “Choose a file or header file - pop_fileio ()” will be appears and it is necessary to select and open the EEG header file “vhdr” as shown in figure A.4.

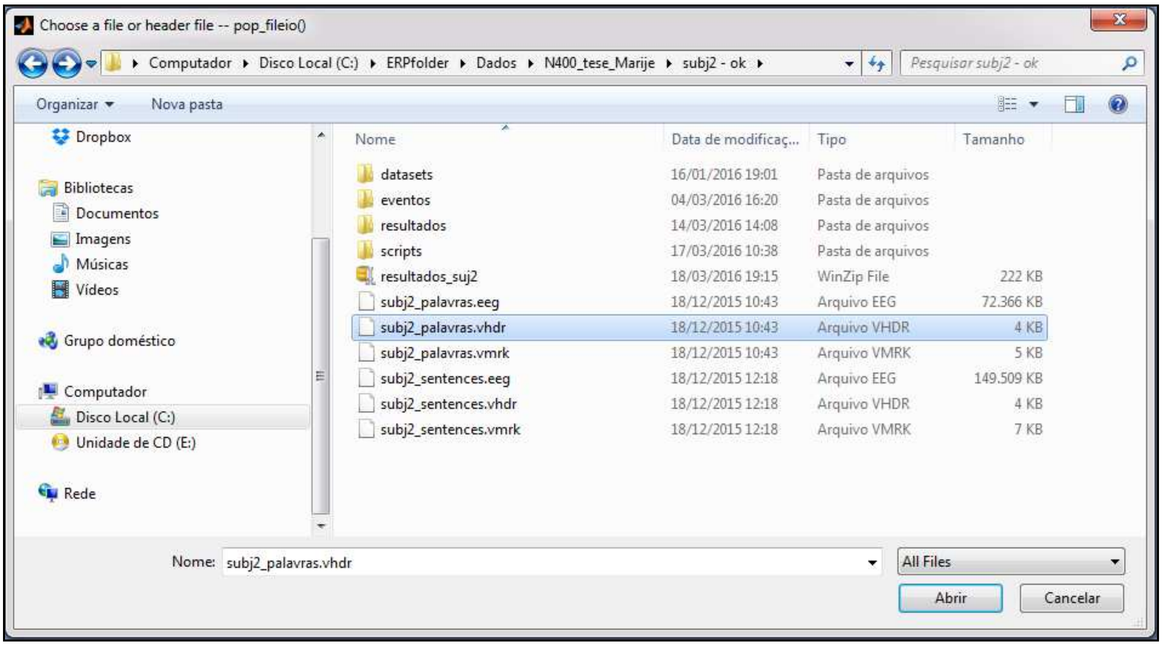


Figure A.4 - EEGLAB® “Choose a file or header file - pop_field()” window and the “vhdr” file to be opened

a.5) For the remaining windows, select OK, to maintain the EEGLAB® default parameters as shown in figures A.5 and A.6.

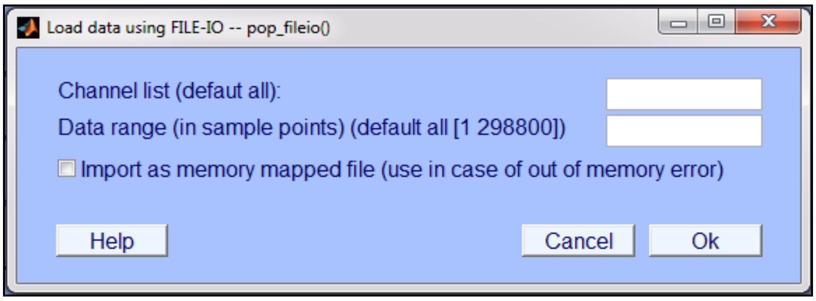


Figure A.5 - EEGLAB® “Load data using FILE-IO --- pop_fileio ()” window

Figure A.6 - EEGLAB® “Dataset info --- pop_newset ()” window

As a result, EEGLAB® acquire the EEG data preliminary parameters for the selected subject and experiment task as shown in figure A.7.

Figure A.7 - EEG data preliminary parameters for the selected subject and experiment task in the EEGLAB® main window

b) The EEG Channel Locations insertion

b.1) In the EEGLAB® window, select the following command tab sequence: Edit/Channel Locations as shown in figure A.8.

Figure A.8 - Channel Locations selection in EEGLAB®

b.2) For the remaining windows, select OK, to maintain the EEGLAB® default parameters as shown in figures A.9 and A.10.

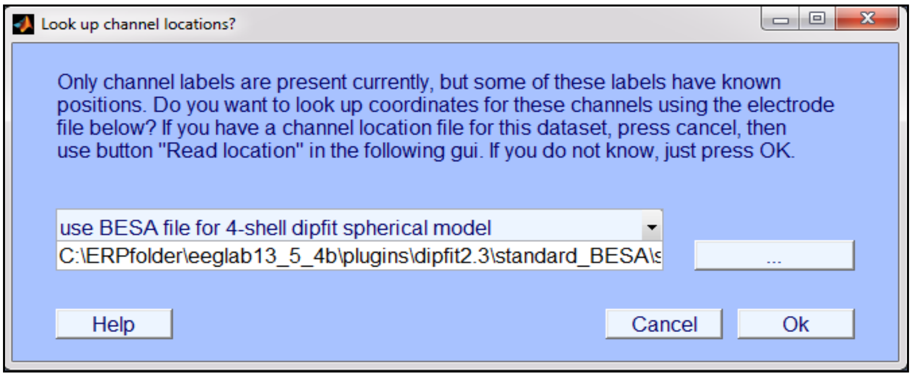


Figure A.9 - EEGLAB® “Look up channel locations” window

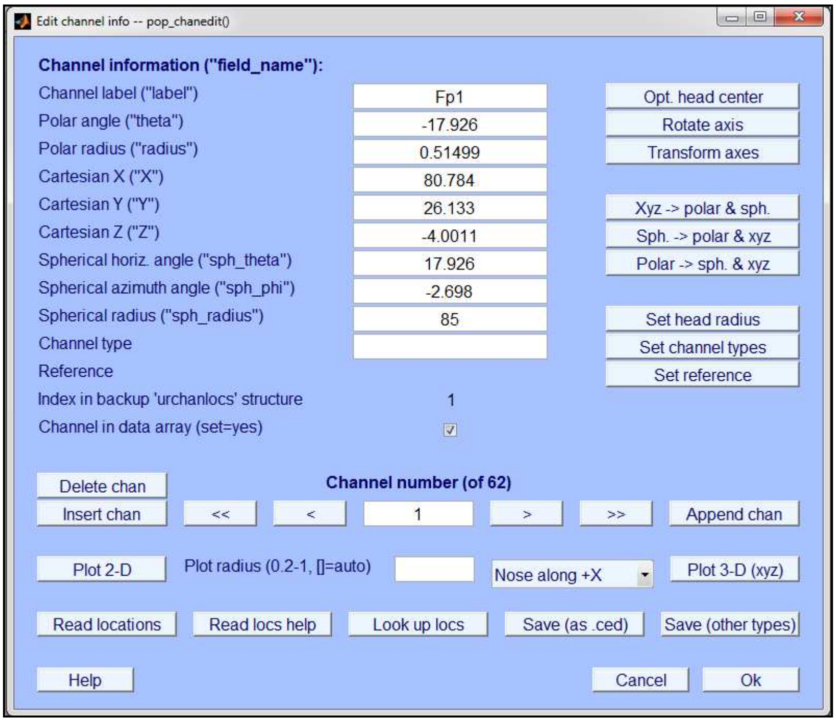


Figure A.10 - EEGLAB® “Edit channel info --- pop_chanedit” window

b.3) As a result, EEGLAB® acquire the default channel location and modify the EEG parameter Channel Location for “Yes” as shown in figure A.11.

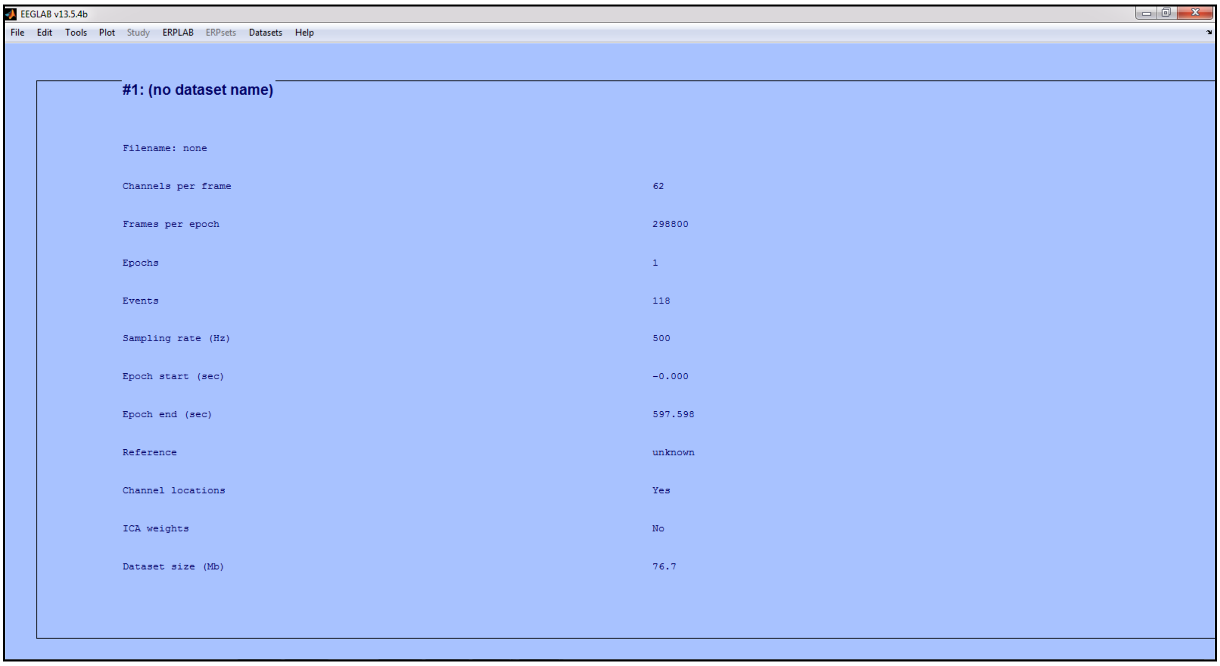


Figure A.11 - EEG data preliminary parameters after the insertion of the channel locations

c) Create the Event List for the experiment using ERPLAB®

c.1) In the EEGLAB® main window, select “ERPLAB” tab and the following command tabs sequence: “Event List/Create EEG EVENT LIST” as shown in figure A.12.

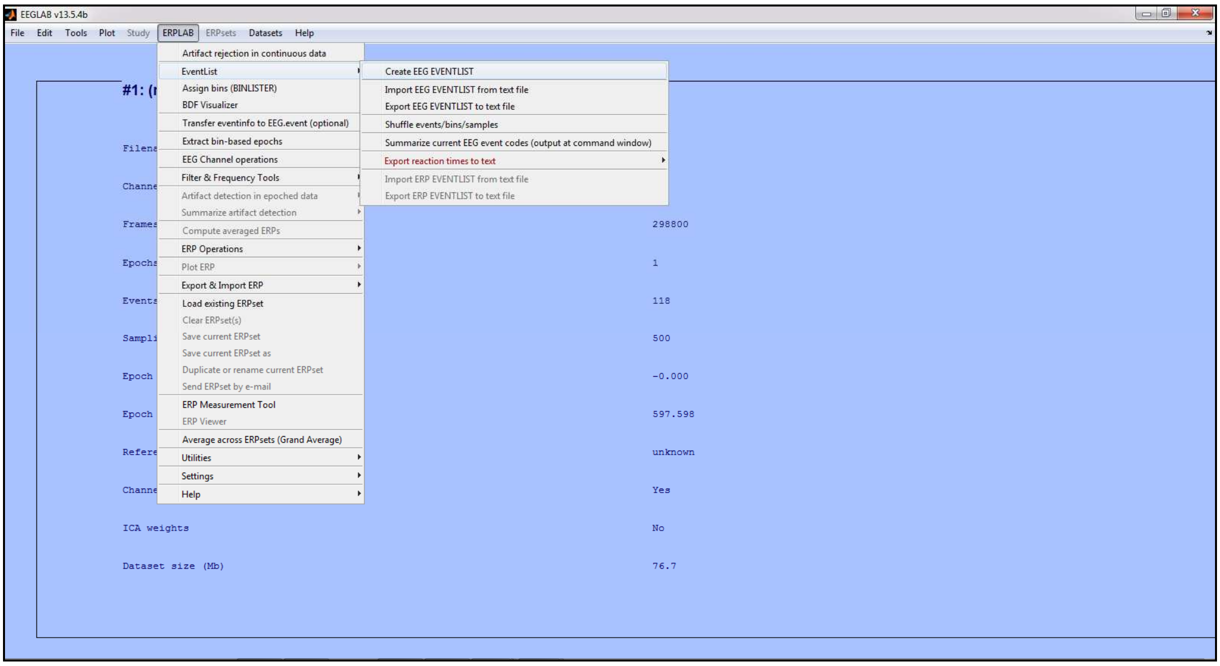


Figure A.12 - Creating the EEG Event List in ERPLAB® command tab

An ERPLAB Warning window about the existence of non-numerical code in the event list will appears. Just press the button “Continue” as shown in figures A.13.

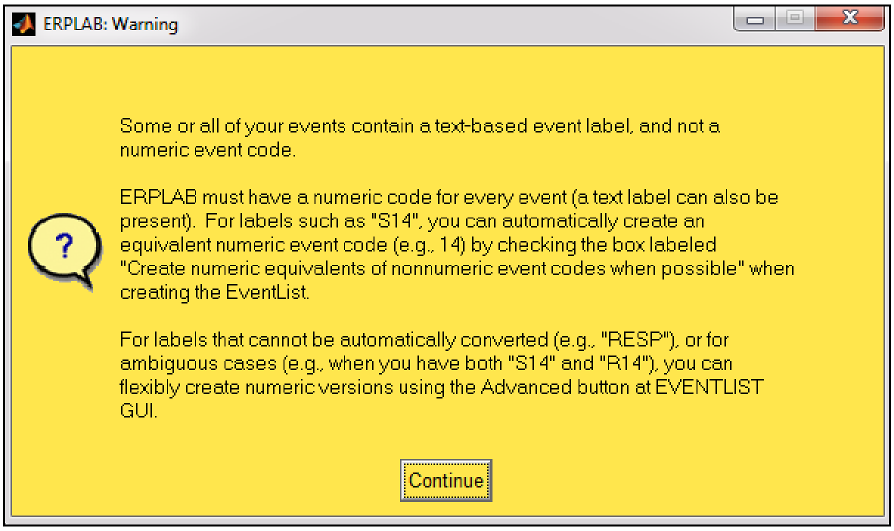


Figure A.13 - ERPLAB® Warning window

c.2) The ERPLAB® window “CREATE A BASIC EVENTLIST GUI” will appears and press the button “Advanced” as shown in figure A.14.

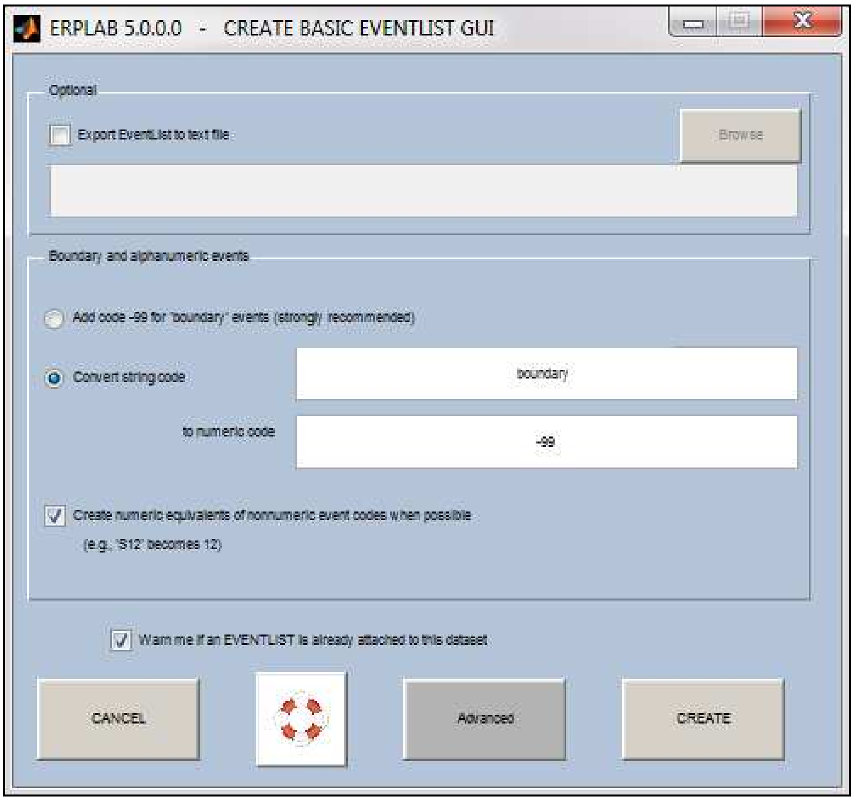


Figure A.14 - ERPLAB® “CREATE A BASIC EVENTLIST GUI” window

c.3) The ERPLAB® window “CREATE ADVANCED EVENTLIST GUI” will appears as shown in figure A.15.

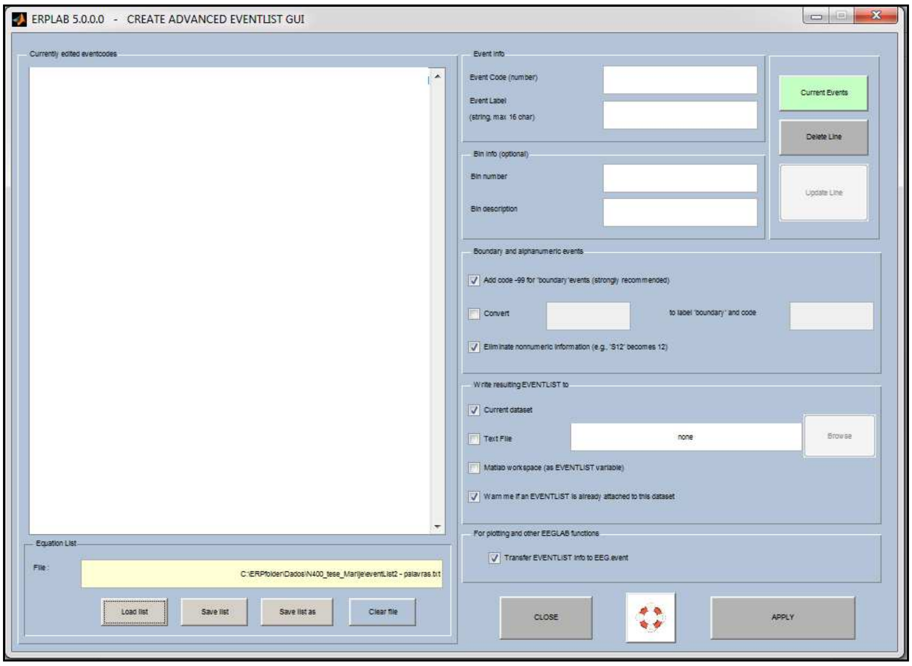


Figure A.15 - ERPLAB® “CREATE ADVANCED EVENTLIST GUI” window

c.4) It is necessary to create the Event List text file containing the types of epochs for each specific experiment task. In this case, as the experiment is the word task, it is created a file containing the 4 types of epochs previewed for the word task as described in the item II.3. The file will have the content as below:

1	"S1"	1	"S1"
2	"S2"	2	"S2"
3	"S3"	3	"S3"
4	"S4"	4	"S4"

Therefore, it is possible to use the program Notepad to create the text file and named as “eventList2 - palavras.txt”, for instance.

c.5) After the creation of this text file, press the button “Load List” in the ERPLAB® window “CREATE ADVANCED EVENTLIST GUI” and load the Event List text file created in the previously step as shown in figure A.16.

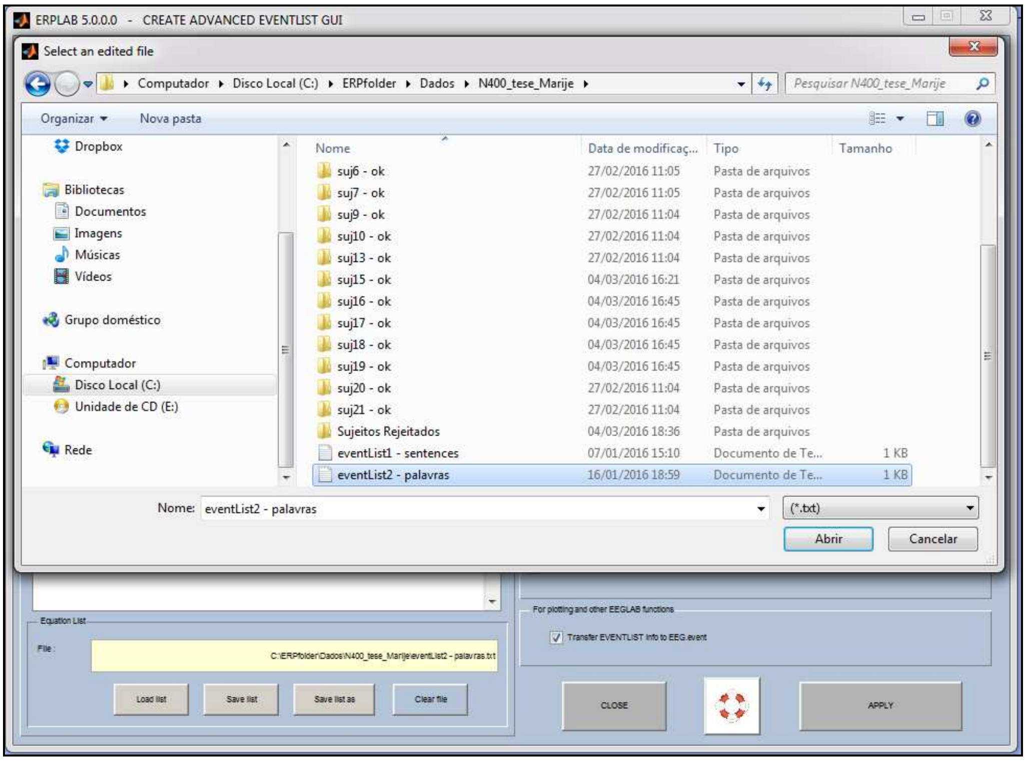


Figure A.16 - ERPLAB® “CREATE ADVANCED EVENTLIST GUI” window and the load of the Event List text file

c.6) After that, it will appears in the ERPLAB® “CREATE ADVANCED EVENTLIST GUI”, in the field “Currently edited eventcodes” the codification established in the Event List text file, as shown in figure A.17.

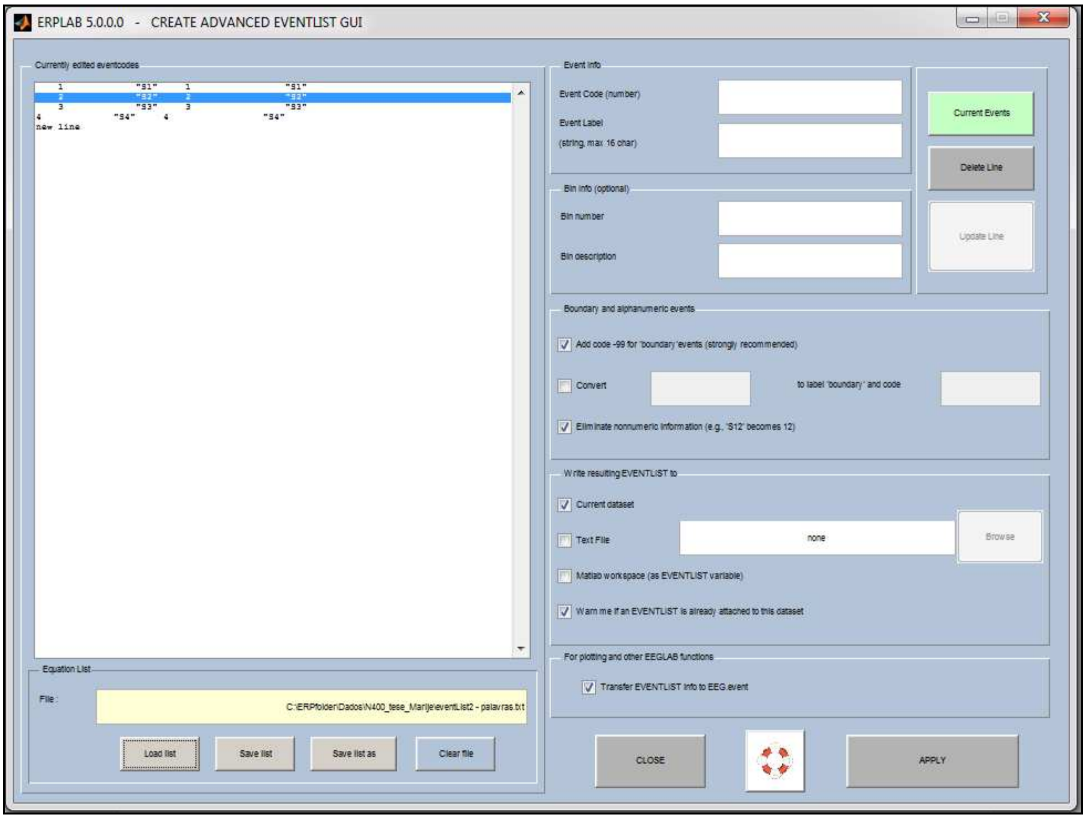


Figure A.17 - ERPLAB® “CREATE ADVANCED EVENTLIST GUI” window with the evet list inserted in the field “Currently edited eventcodes”

c.7) After the insertion of the Event List text file, select the button “Apply”. The ERPLAB® “Modify EEG.event GUI” will appears. To update the Event List database file “EEG.event” of EEGLAB® and ERPLAB®, maintain the selection as “Code Labels” and press the button “Apply” as shown in figure A:18.

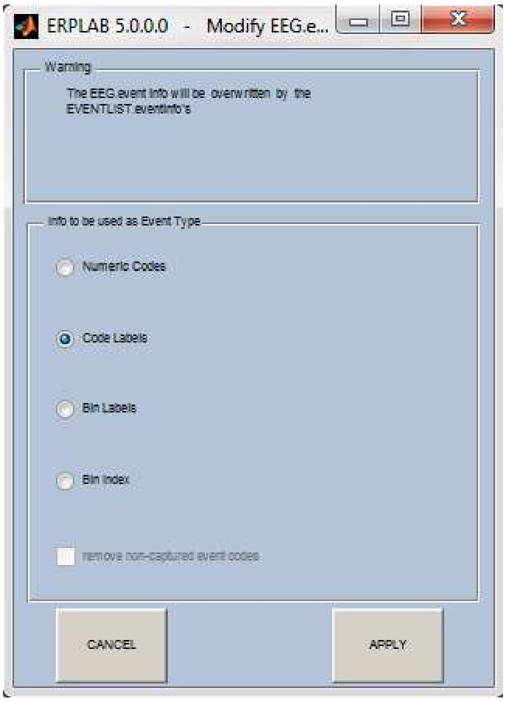


Figure A.18 - ERPLAB® “Modify EEG.event GUI” window

c.8) The ERPLAB® “Dataset info --- pop_newset ()” window will appears. Press the button “Ok” as shown in figure A.19. Then, the “eeg.event” database file is updated.

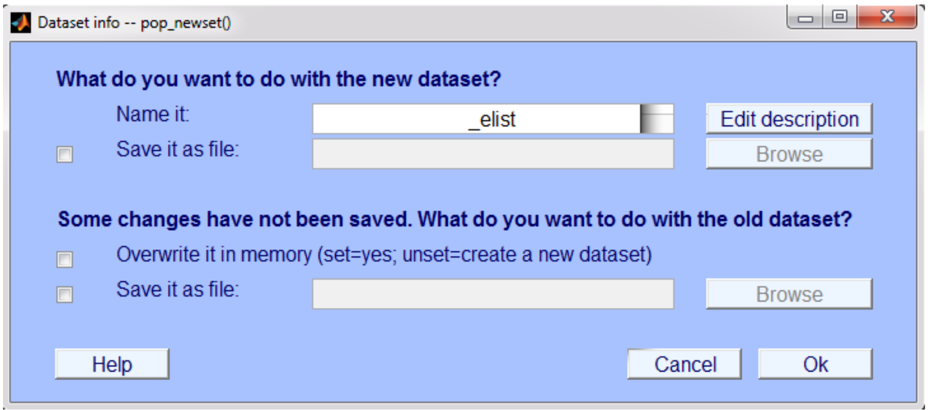


Figure A.19 - ERPLAB® “Dataset info --- pop_newset ()” window

d) Applying the 30 Hz Low-Pass Butterworth Filter in the EEG data using ERPLAB® in order to reduce noise.

d.1) In the EEGLAB® main window, select “ERPLAB” tab and the following command tabs sequence: “Filters and frequency tools//Filters for EGG Data” as shown in figure A.20.

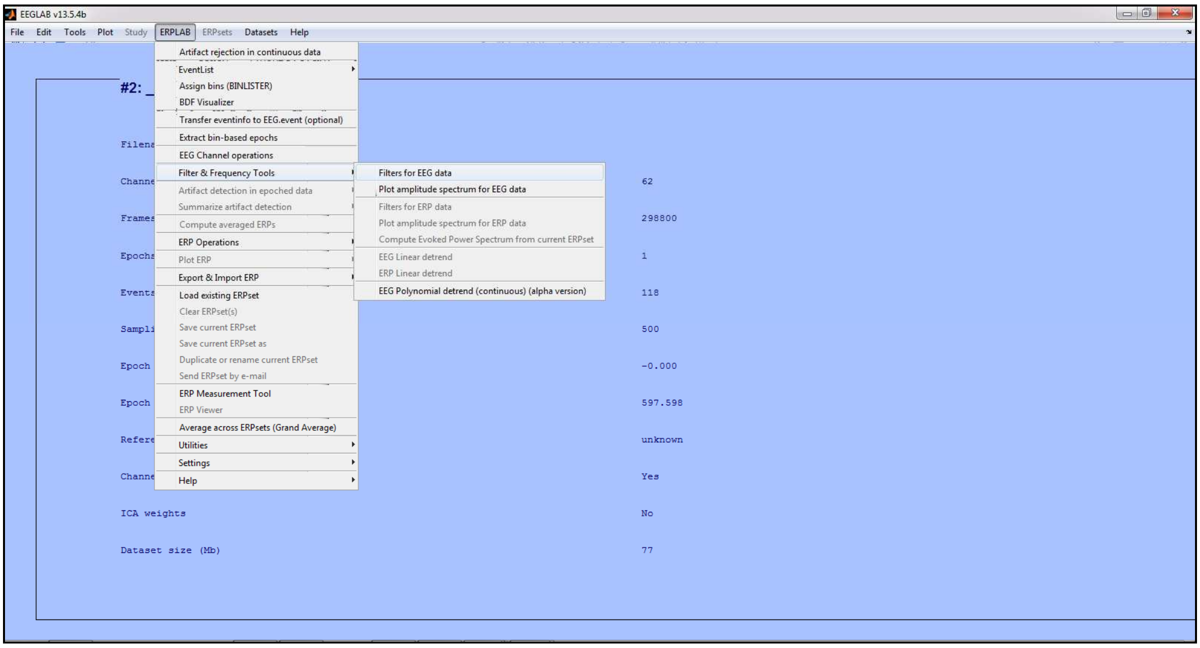


Figure A.20 - Filtering the EEG data

d.2) The ERPLAB “Basic Filter GUI for continuous EEG” will appears. Then, select the options for a filter 3’ Hz Butterworth order 2 and press the button “Apply” as shown in figure A.21.

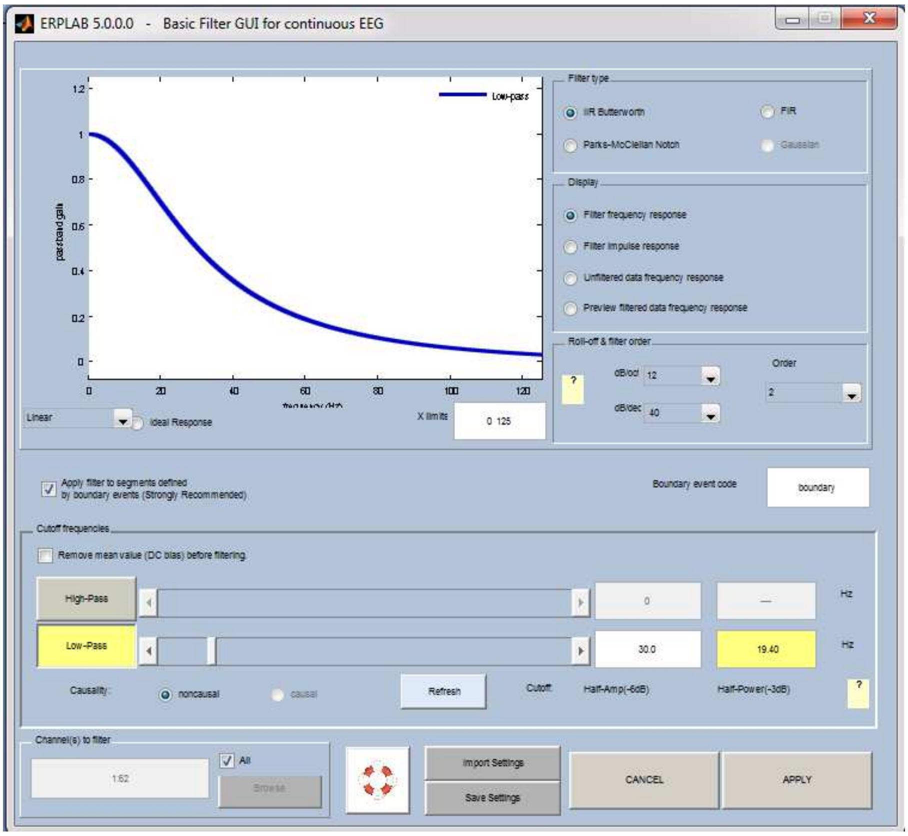


Figure A.21 - ERPLAB® “Basic Filter GUI for continuous EEG” window d.3) The ERPLAB® “Dataset info --- pop_newset ()” window will appears. Press the button “Ok” as shown in figure A.22. Then, the EEG data is filtered.

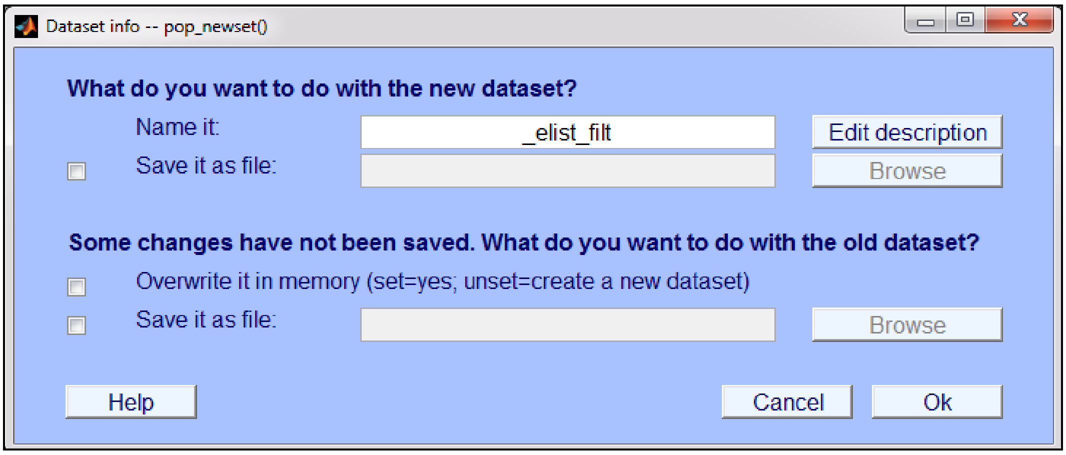


Figure A.22 - ERPLAB® “Dataset info --- pop_newset ()” window

e) Extract the bin-based epochs from the Event List updated

e.1) In the EEGLAB® main window, select “ERPLAB” tab and the command “Extract bin-based epochs” as shown in figure A.23

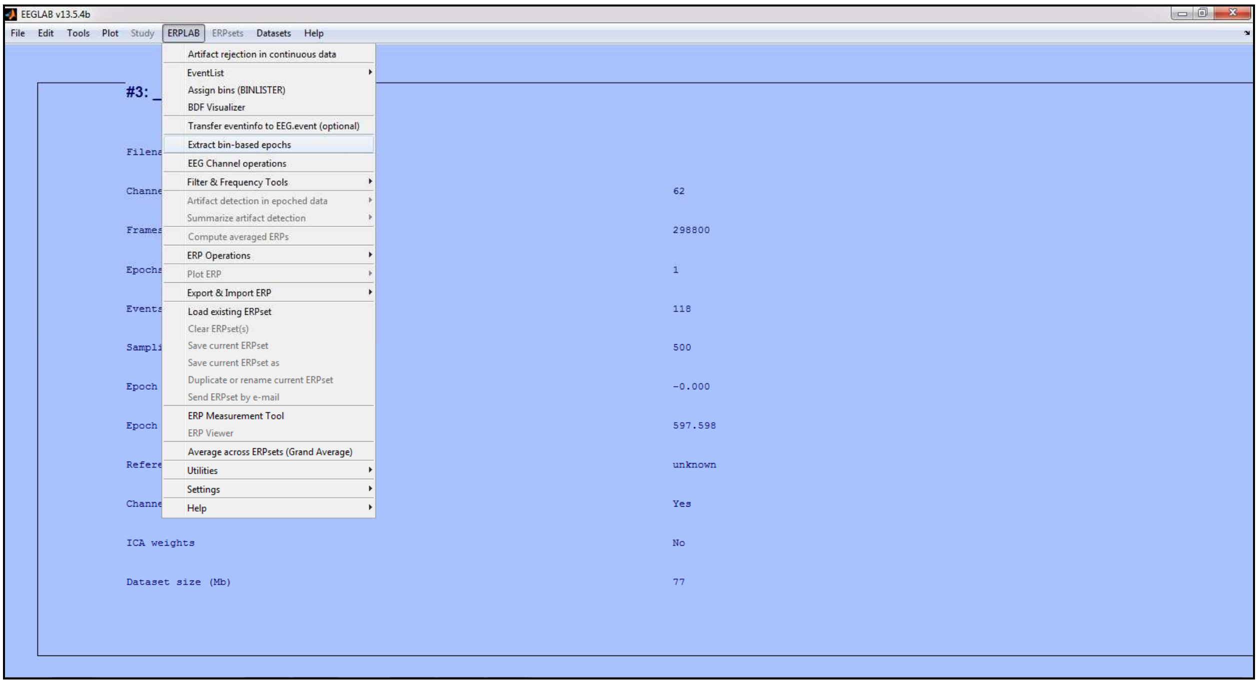


Figure A.23 - Extracting the bin-based epochs with ERPLAB®

e.2) The ERPLAB® “EXTRACT BINEPOCH” window will appears. Then, it is necessary to insert the time limits of the range of the bin-based epochs in the field “Bin-based epoch time range (ms)” of -200 ms to 1000 ms as describes in the item II.3. After the insertion, press the button “RUN” as shown in figure A.24.

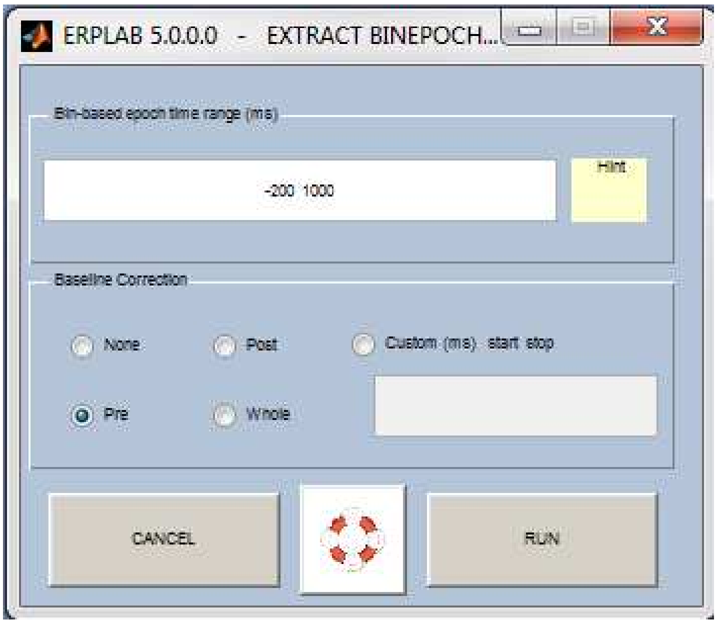


Figure A.24 - ERPLAB® "EXTRACT BINEPOCH" window

e.3) The ERPLAB® "Dataset info --- pop_newset ()" window will appears. In this step, it is necessary to name the dataset created in the field "Name it:" The name chosen in this work is "Suj1_Palavras_FiltPB30Hz_Epocas", for the subject 1, for instance. After the insertion of the dataset name, press the button "Ok", as shown in figure A.52. Then, the EEG dataset is read to be worked and the dataset for each subject can be accessed by the files Suj1_Palavras_FiltPB30Hz_Epocas.set and Suj1_Palavras_FiltPB30Hz_Epocas.fdt.

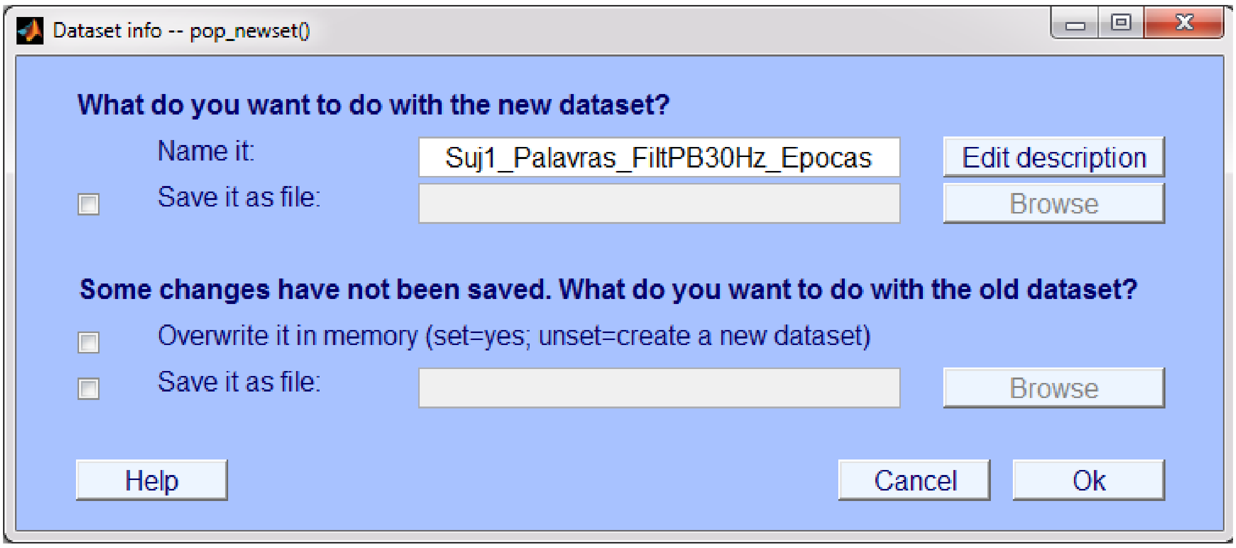


Figure A.25 - ERPLAB® "Dataset info --- pop_newset ()" window

Appendix B - EEGLAB® and ERPLAB® steps for ERP features extraction for each subject dataset

a) Load the EEGLAB dataset for an subject

a.1) With EEGLAB® program already opened, in the EEGLAB® main window, select “File” tab and the command “Load existing dataset” as shown in figure B1.

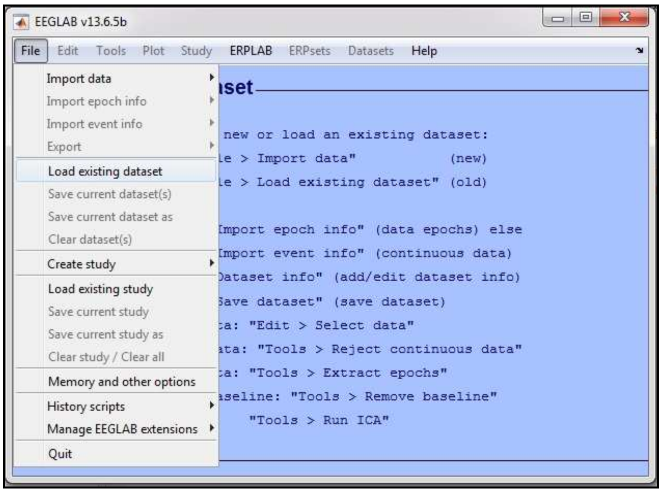


Figure B.1 - Load existing dataset with EEGLAB®

a.2) The EEGLAB® “Load dataset(s) --- pop_newset ()” window will appears. Then select the file with the extension *.set related do the dataset, which the ERP parameters will be extracted. For instance, the file of Words Task for the subject 1 is Suj1_Palavras_FiltPB30Hz_Epocas.set, as shown in figure B.2. Then, press the button “Open” and dataset will be loaded as shown in figure B.3

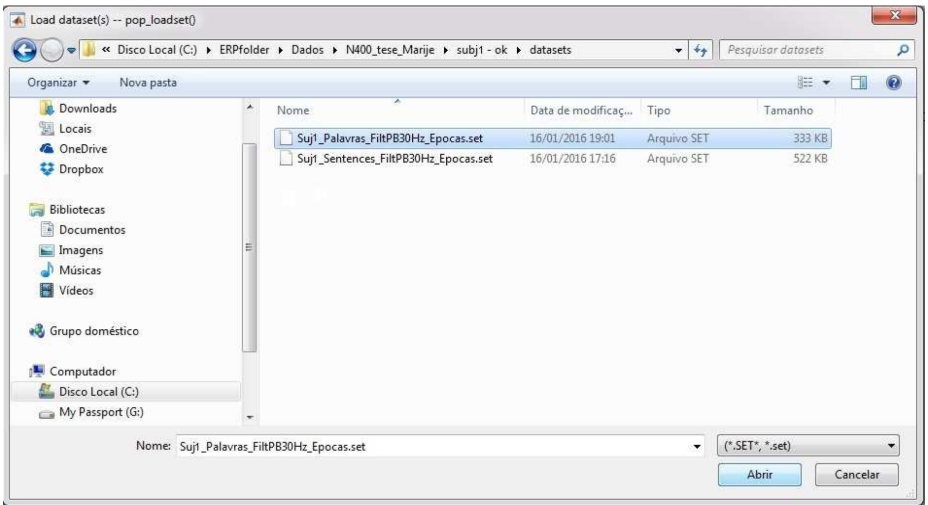


Figure B.2 - EEGLAB® “Load dataset(s) --- pop_newset ()” window

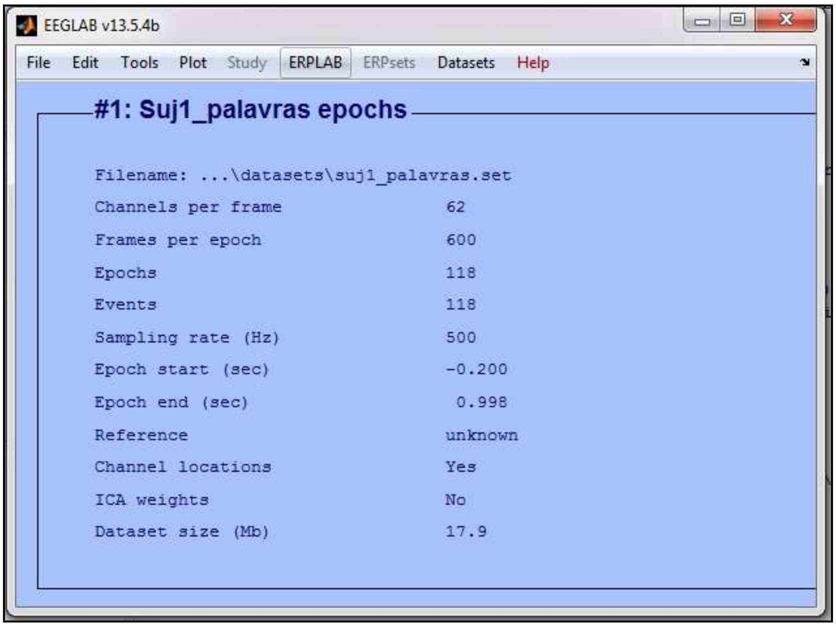


Figure B.3 - EEGLAB® main window with the indication of the dataset loaded with all EEG dataset parameters

b) Set the EEG channel operations

b.1) In the EEGLAB® main window, select "ERPLAB®" tab and the command "EEG channel operations" as shown in figure B.4.

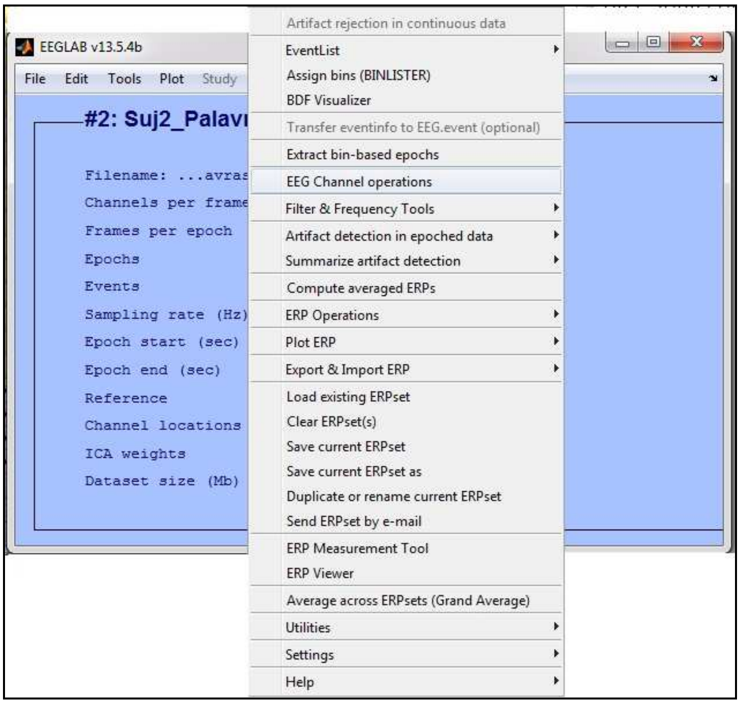


Figure B.4 - Open EEG Channel operations with ERPLAB®

b.2) The ERPLAB® "ERPLAB 5.5.0 - Channel Operation GUI for EEG" window will appears. In this step, it is necessary to create the ERP extra channels that correspond to the ERP signals of the ROIs. These signals are according to the sum of the individual EEG channels as explained in the item II.3 and showed in figure 6 of this dissertation. Then, it is necessary to create a *.txt file with the sum operations of the individual channels, for instance by the Notepad. The file's name can be "ERPChannelList-txt", for instance. The content of this file is:

ch63 = (ch2 + ch7 + ch27 + ch34 + ch38+ ch60)/6 label FrontalMidLine

ch64 = (ch11 + ch21 + ch22 + ch39 + ch52 + ch56)/6 label CentralMidLine

ch65 = (ch11 + ch12 + ch21 + ch43 + ch51 + ch52)/6 label ParientalMidLine

ch66 = (ch15 + ch16 + ch17 + ch46 + ch47 + ch48)/6 label OccipitalMidLine

ch67 = (ch3 + ch4 + ch6 + ch35 + ch36 + ch37)/6 label FrontalLeftSide

ch68 = (ch8 + ch9 + ch10 + ch40 + ch41 + ch42)/6 label CentralLeftSide

ch69 = (ch10 + ch13 + ch14 + ch41 + ch42 + ch44)/6 label ParientalLeftSide

ch70 = (ch13 + ch14 + ch44 + ch45 + ch46)/5 label OccipitalLeftSide

ch71 = (ch26 + ch28 + ch29 + ch57 + ch58 + ch59)/6 label FrontalRightSide

ch72 = (ch20 + ch23 + ch24 + ch53 + ch54 + ch55)/6 label CentralRightSide

ch73 = (ch18 + ch19 + ch20 + ch50 + ch53 + ch54)/6 label ParientalRightSide

ch74 = (ch18 + ch19 + ch48 + ch49 + ch50)/5 label OccipitalRightSide

As can be seen above, each new ERP channel correspond to the sum of the individual EEG channels, according each ROI and which respective label is written after their correspondent operation. After the creation of the *.txt file, he shall be loaded in the ERPLAB® "ERPLAB 5.5.0 - Channel Operation GUI for EEG" window. To do that, it is necessary to fill the path of the file location in the field "File:", to press the button "Load List" (where will appear the ERP channels computations) and to press the button "Run", as shown in figure B.5. After that, all the ERP channels are computed with their correspondent ERP ROI signal and the prompt of Matlab® will indicate the end of the process by the message "Done." Create all "erpsets", for all subjects, for both tasks, in this point.

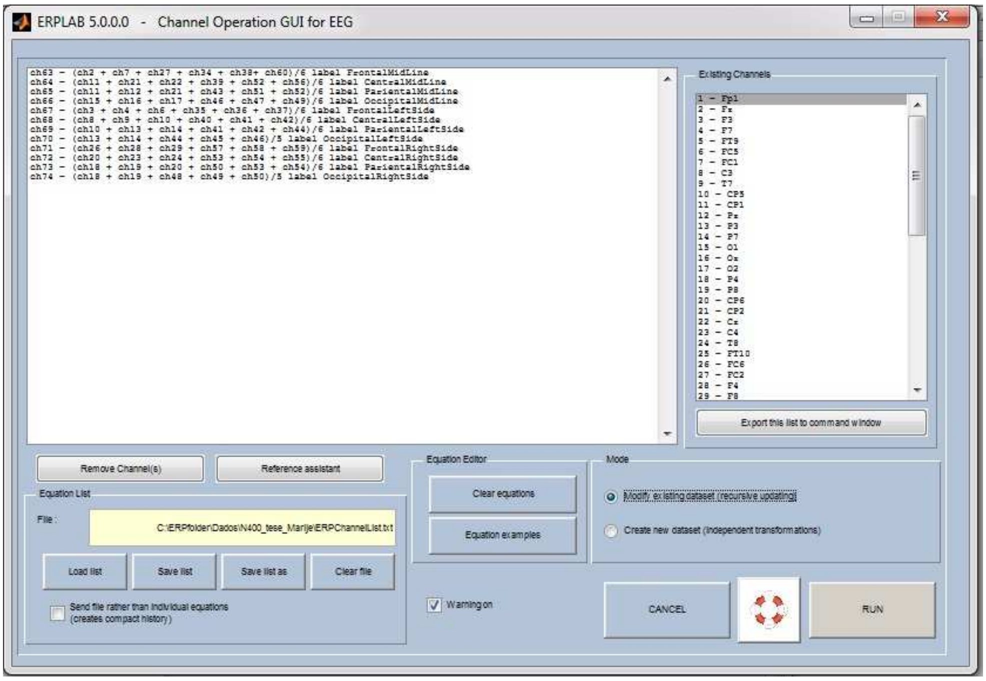


Figure B.5 - ERPLAB® “ERPLAB 5.5.0 - Channel Operation GUI for EEG” window

c) Compute averaged ERPs

c.1) In the EEGLAB® main window, select “ERPLAB®” tab and the command “Compute averaged ERPs” as shown in figure B.6.

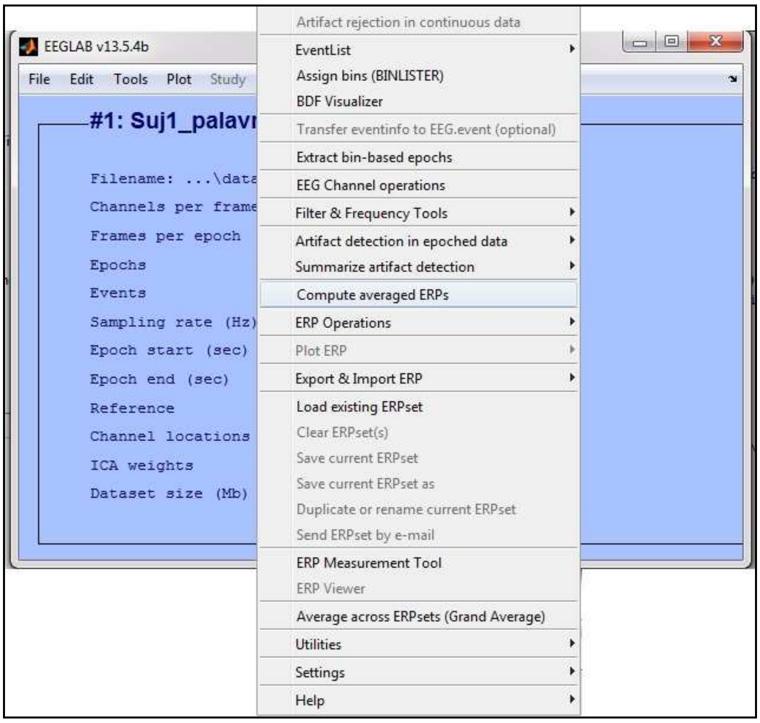


Figure B.6 - Compute averaged ERPs with ERPLAB®

c.2) The ERPLAB® “ERPLAB 5.5.0 - WEIGHTED AVERAGE GUI” window will appears. Then select the button “Run”, as shown in figure B.7.

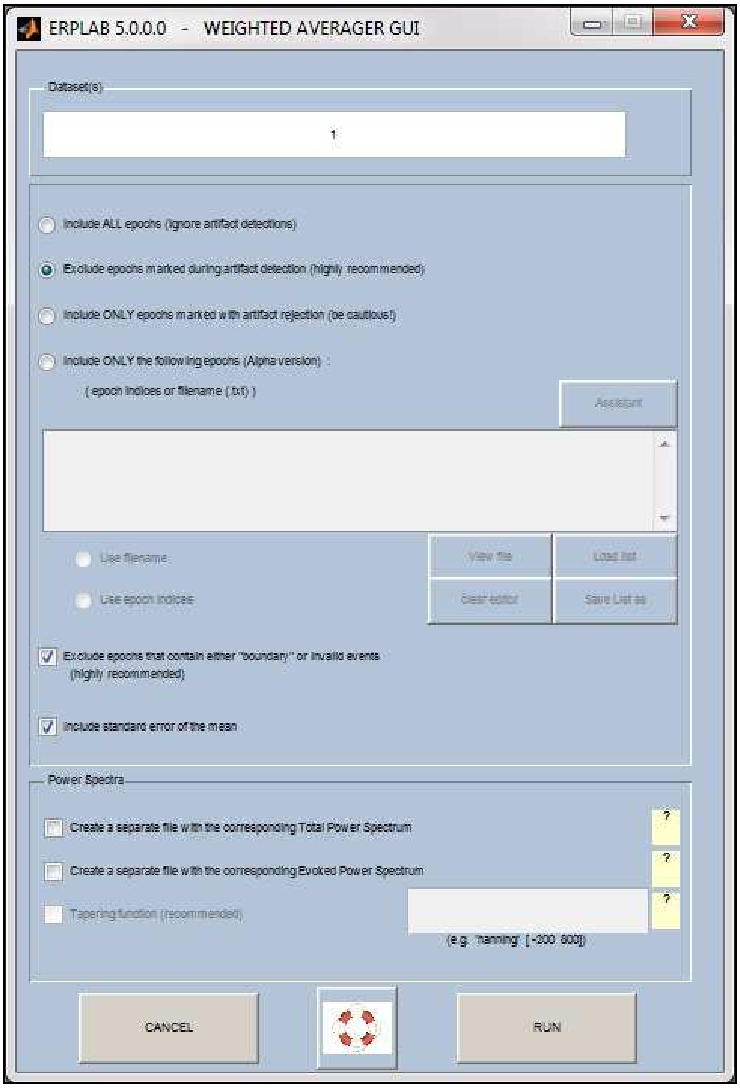


Figure B.7 - ERPLAB® “ERPLAB 5.5.0 - WEIGHTED AVERAGE GUI” window

c.3) The ERPLAB® “ERPLAB 5.5.0 - save ERPset GUI” window will appears. In this step, it is necessary to name the ERPset created in the field “erpname”. The name chosen in this work is “subject2_words”, for the subject 1, for instance. After the insertion of the dataset name, press the button “Ok” as shown in figure B.8. Then the averaged ERP for all channels will be computed and saved in the respectively “erpset”.

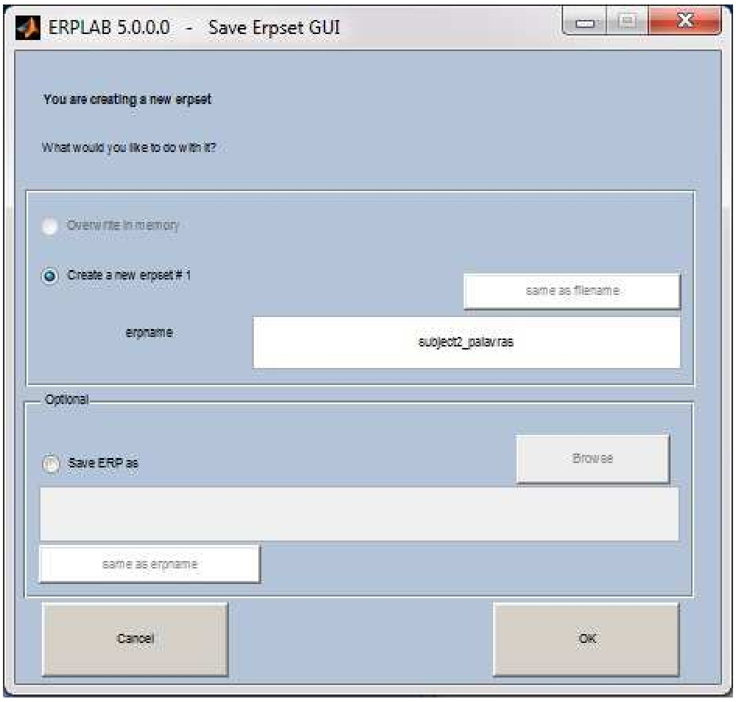


Figure B.8 - ERPLAB® “ERPLAB 5.5.0 - save ERPset GUI” window

d) Extract the ERP parameters (features) by the ERPLAB® ERP measurement Tool

d.1) In the EEGLAB® main window, select “ERPLAB®” tab and the command “ERP Measurement Tool” as shown in figure B.9.

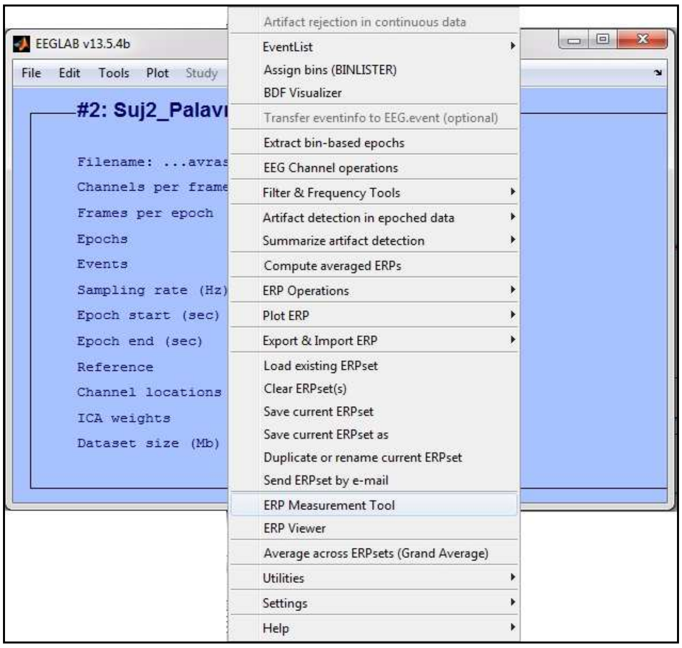


Figure B.9 - Open the ERPLAB® ERP measurement Tool

d.2) The ERPLAB® “ERPLAB 5.5.0 - ERP Measurements GUI” window will appears, as shown in figure B.10.

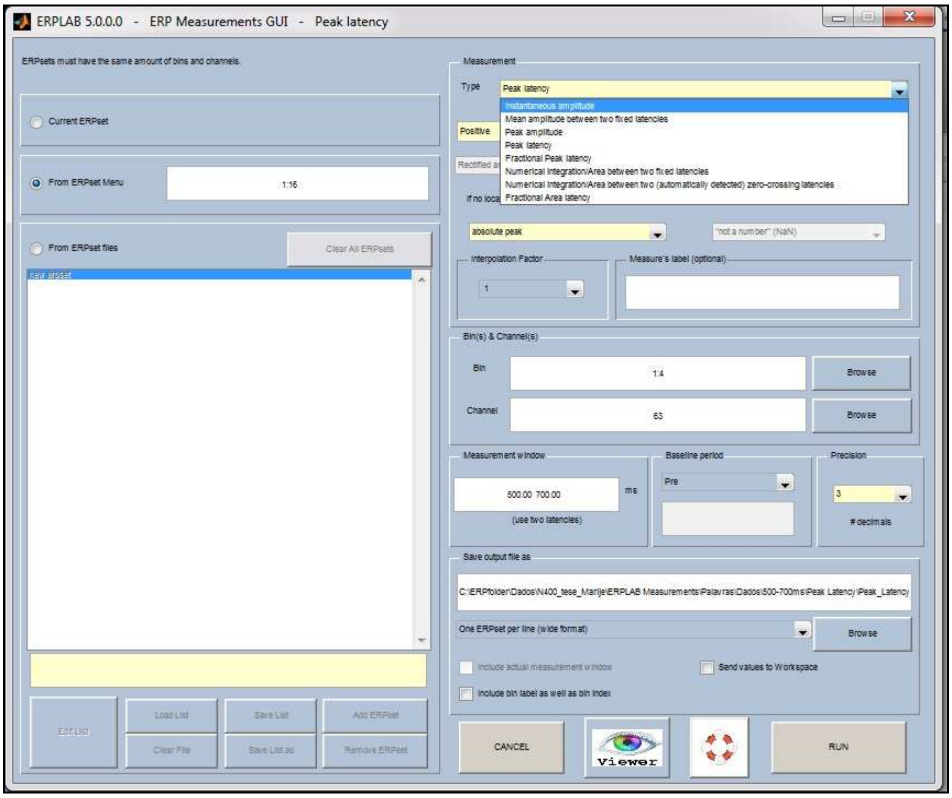


Figure B.10 - ERPLAB® “ERPLAB 5.5.0 - ERP Measurements GUI” window

Then, it is necessary to set the following fields, in the presented order, to extract the ERP parameters (features) wished, as described below:

i) Field “From the ERPset Menu” - to select all the “erpsets” previously created and loaded that will be treated. As for this study we have 16 subjects, it will be created 16 “eprsets”. Then, this field shall be filled with “1:16” (“erpset” 1 to 16), as the example of figure B.10;

ii) Area “Measurement”, field “Type” - to select which feature will be extracted. There are many features available to be selected, but for this study will be extracted only the “Mean Amplitude Between two fixed latencies”, “Peak Amplitude”, or “Peak Latency” (as selected in the example of figure B.10);

iii) Area “Measurement”, field “Bin”- to indicate the number of the classes (bins) for each task. For Sentences Task will be filled “1:5”. For Words Task will be filled “1:4” (as filled in the example of figure B.10);

iv) Area “Measurement”, field “Channel” - to select which ERP extra channel created to calculate the feature selected. The values can be 63 to 74, as shown in item b. For instance, in the example of figure B.10, the channel 63 (ROI “Frontal Mid Line”) was filled in the field. This step shall be repeated for all other channels up to 74, to extract all the related feature in their relative output *.txt file;

v) Area “Measurement”, subarea “Measurement Window”, field “ms” - to select the time range of the ERP calculation. For this study, it were chosen three ranges that can be filled in this field. For 150 to 300 milliseconds (ms), it will be filled “150.00 300.00”. For 300 to 500 ms, it will be filled “300.00 500.00”. And for 500 to 700 ms, it will be filled “500.00 700.00” (as filled in the example of figure B.10);

vi) Area “Measurement”, field “Save output file as:” - to fill with the path and the name of the output *.txt file that will contain the values of the feature calculated for all “erpsets” loaded for the ERP channel chosen. This file will have information with the example format shown in figure B.11. In this Figure, for instance, the extracted feature is the “Mean Amplitude Between two fixed latencies” feature, from the ROI “Central Left Side”, Words Task.

bin1_CentralLeftSide	bin2_CentralLeftSide	bin3_CentralLeftSide	bin4_CentralLeftSide	ERPset
0,509	0,605	0,598	0,191	subject2_words
0,559	0,557	0,392	0,162	subject3_words
2,296	2,203	2,014	1,315	subject4_words
1,051	-1,284	-1,262	-1,430	subject5_words
1,487	-0,777	-0,656	-0,570	subject6_words
-1,207	-0,783	-1,888	-1,668	subject7_words
2,474	1,816	1,010	2,494	subject9_words
5,016	3,183	2,126	1,009	subject10_words
4,128	2,853	0,604	1,952	subject13_words
0,599	1,410	1,288	-0,029	subject15_words
3,296	4,020	1,004	0,178	subject16_words
1,298	1,906	0,506	-0,450	subject17_words
-0,584	-2,126	-1,157	0,199	subject18_words
-3,226	4,989	-1,109	-0,183	subject19_words
-1,361	-0,057	-1,585	-1,369	subject20_words
1,448	-0,683	-0,002	0,830	subject21_words

Figure B.11 - Output file example with the extracted ERP feature values for all classes and subjects involved. In this case, “Mean Amplitude Between two fixed latencies” feature, from the ROI “Central Left Side”, and Words Task

In figure B.11, the bin1 is correspondent to the class S1, the bin2 is correspondent to the class S2, the bin3 is correspondent to the class S3, and , the bin3 is correspondent to the class S4. The related subject is indicate by the name of the “erpset” beside the correspondent row.

d.3) After to obtain all output *.txt files with the features extracted, they shall be organized by a spreadsheet software as Microsoft Excel® to allow the use of these data for the Matlab®.

Appendix C - Extracts from the MATLAB® help sites of the pattern recognition functions used

All classification scenarios explanations were extracted from the references and the Matlab® links are mentioned in the bibliography.

C.1 Unsupervised pattern classification and clustering

C.1.1 Hierarchical Clustering and Unsupervised Classifier

Hierarchical clustering analysis (HCA) is a method of cluster analysis in order to build a hierarchy of clusters. As described by Webb (2002), hierarchical clustering procedures are the most commonly used method of summarizing data structure. A hierarchical tree is a nested set of partitions represented by a tree diagram or dendrogram. Sectioning a tree at a particular level produces a partition into g disjoint groups. If two groups are chosen from different partitions (the results of partitioning at different levels) then either the groups are disjoint or one group wholly contains the other.

Normally, as described by Rokach et al. (2005), there are two strategies for hierarchical clustering. The first is called agglomerative, where each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy, a "bottom up" approach. The second one is called divisive, where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy, a "top down" approach. Explaining better:

> *There are several different algorithms for finding a hierarchical tree. An agglomerative algorithm begins with n subclusters, each containing a single data point, and at each stage merges the two most similar groups to form a new cluster, thus reducing the number of clusters by one. The algorithm proceeds until all the data fall within a single cluster.*
> *A divisive algorithm operates by successively splitting groups, beginning with a single group and continuing until there are n groups, each of a single individual. Generally,divisive algorithms are computationally inefficient (except where most of the variables are binary attribute variables) (WEBB,2002)."*

In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric (a measure of distance between pairs of observations), and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets. The choice of an appropriate metric will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations.

As described in the MATLAB® (2016a), hierarchical clustering do the agroupments with the data over a variety of scales by creating a cluster tree diagram more known as a dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. This allows you to decide the level or scale of clustering that is most appropriate for your application. It incorporates the “pdist”, “linkage”, and “cluster” functions, which may be used separately for more detailed analysis. The “dendrogram” function plots the cluster tree.

In this study, it was used an agglomerative hierarchical cluster analysis using Matlab® Statistics and Machine Learning Toolbox functions, following the procedure above (MATLAB®, 2016b):

a) Step 1 - Find the similarity or dissimilarity between every pair of objects in the data set - In this step, you calculate the distance between objects using the “pdist” function. The “pdist” function supports many different ways to compute this measurement. The metric available are shown in Figure C.1

Metric	Description
`'euclidean'`	Euclidean distance (default).
`'squaredeuclidean'`	Squared Euclidean distance. (This option is provided for efficiency only. It does not satisfy the triangle inequality.)
`'seuclidean'`	Standardized Euclidean distance. Each coordinate difference between rows in X is scaled by dividing by the corresponding element of the standard deviation `S=nanstd(X)`. To specify another value for `S`, use `D = pdist(X,'seuclidean',S)`.
`'cityblock'`	City block metric.
`'minkowski'`	Minkowski distance. The default exponent is 2. To specify a different exponent, use `D = pdist(X,'minkowski',P)`, where `P` is a scalar positive value of the exponent.
`'chebychev'`	Chebychev distance (maximum coordinate difference).
`'mahalanobis'`	Mahalanobis distance, using the sample covariance of `X` as computed by `nancov`. To compute the distance with a different covariance, use `D = pdist(X,'mahalanobis',C)`, where the matrix `C` is symmetric and positive definite.
`'cosine'`	One minus the cosine of the included angle between points (treated as vectors).
`'correlation'`	One minus the sample correlation between points (treated as sequences of values).
`'spearman'`	One minus the sample Spearman's rank correlation between observations (treated as sequences of values).
`'hamming'`	Hamming distance, which is the percentage of coordinates that differ.
`'jaccard'`	One minus the Jaccard coefficient, which is the percentage of nonzero coordinates that differ.

custom distance function	A distance function specified using @:
	`D = pdist(X,@distfun)`
	A distance function must be of form
	`d2 = distfun(XI,XJ)`
	taking as arguments a 1-by-*n* vector `XI`, corresponding to a single row of `X`, and an *m*2-by-*n* matrix `XJ`, corresponding to multiple rows of `X`. `distfun` must accept a matrix `XJ` with an arbitrary number of rows. `distfun` must return an *m*2-by-1 vector of distances `d2`, whose *k*th element is the distance between `XI` and `XJ(k,:)`.

Figure C.1 - “pdist” function metrics (MATLAB®, 2016c)

The “pdist” function is used to calculate the distance between every pair of objects in a data set. For a data set made up of m objects, there are m*(m - 1)/2 pairs in the data set. The result of this computation is commonly known as a distance or dissimilarity matrix. For example, consider a data set, X, made up of five objects where each object is a set of x,y coordinates, as can be seen below:

Object 1: 1, 2

Object 2: 2.5, 4.5

Object 3: 2, 2

Object 4: 4, 1.5

Object 5: 4, 2.5

Figure C.2 shows the plot of this data in a plan X,Y:

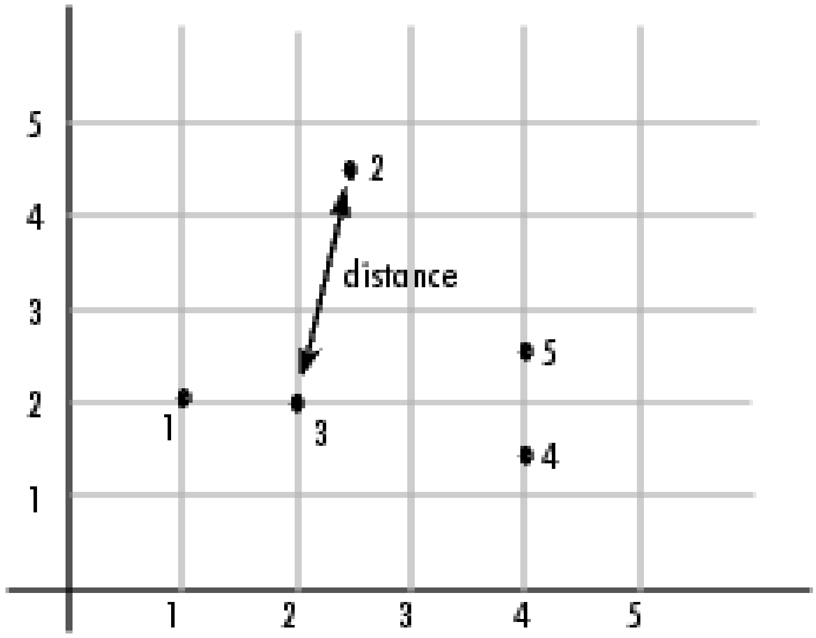


Figure C.2 - Distance Information (MATLAB®, 2016b)

The “pdist” function returns this distance information in a vector, Y, where each element contains the distance between a pair of objects.

For the example of figure 9:

Y = pdist(X)

Y =

Columns 1 through 7

2.9155   1.0000   3.0414   3.0414   2.5495   3.3541   2.5000

Columns 8 through 10

2.0616   2.0616   1.0000

To make it easier to see the relationship between the distance information generated by “pdist” and the objects in the original data set, you can reformat the distance vector into a matrix using the squareform function. In this matrix, element i,j corresponds to the distance between object i and object j in the original data set. In the following example, element 1,1 represents the distance between object 1 and itself (which is zero). Element 1,2 represents the distance between object 1 and object 2, and so on.

squareform(Y):

ans =

0	2.9155	1.0000	3.0414	3.0414
2.9155	0	2.5495	3.3541	2.5000
1.0000	2.5495	0	2.0616	2.0616
3.0414	3.3541	2.0616	0	1.0000
3.0414	2.5000	2.0616	1.0000	0

b) Step 2 - Group the objects into a binary, hierarchical cluster tree - In this step, you link pairs of objects that are in close proximity using the “linkage” function. The “linkage” function uses the distance information generated in step 1 to determine the proximity of objects to each other. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.

Once the proximity between objects in the data set has been computed, you can determine how objects in the data set should be grouped into clusters, using the “linkage” function. The “linkage” function takes the distance information generated by “pdist” and links pairs of objects that are close together into binary clusters (clusters made up of two objects). The “linkage” function then links these newly formed clusters to each other and to other objects to create bigger clusters until all the objects in the original data set are linked together in a hierarchical tree.

For example, given the distance vector Y generated by “pdist” from the sample data set of x- and y-coordinates, the “linkage” function generates a hierarchical cluster tree, returning the “linkage” information in a matrix, Z.

Z = linkage(Y)

Z =

4.0000	5.0000	1.0000
1.0000	3.0000	1.0000
6.0000	7.0000	2.0616
2.0000	8.0000	2.5000

In this output, each row identifies a link between objects or clusters. The first two columns identify the objects that have been linked. The third column contains the distance between these objects. For the sample data set of x- and y-coordinates, the “linkage” function begins by grouping objects 4 and 5, which have the closest proximity (distance value = 1.0000). The “linkage” function continues by grouping objects 1 and 3, which also have a distance value of 1.0000.

The third row indicates that the “linkage” function grouped objects 6 and 7. If the original sample data set contained only five objects, what are objects 6 and 7? Object 6 is the newly formed binary cluster created by the grouping of objects 4 and 5. When the “linkage” function groups two objects into a new cluster, it must assign the cluster a unique index value, starting with the value m + 1, where m is the number of objects in the original data set. (Values 1 through m are already used by the original data set). Similarly, object 7 is the cluster formed by grouping objects 1 and 3.

“linkage” function uses distances to determine the order in which it clusters objects. The distance vector Y contains the distances between the original objects 1 through 5. But “linkage” must also be able to determine distances involving clusters that it creates, such as objects 6 and 7. By default, “linkage” function uses a method known as “single” linkage. However, there are a number of different methods available in Matlab®, as shown in figure C.3.

Method	Description
'average'	Unweighted average distance (UPGMA)
'centroid'	Centroid distance (UPGMC), appropriate for Euclidean distances only
'complete'	Furthest distance
'median'	Weighted center of mass distance (WPGMC), appropriate for Euclidean distances only
'single'	Shortest distance
'ward'	Inner squared distance (minimum variance algorithm), appropriate for Euclidean distances only
'weighted'	Weighted average distance (WPGMA)

Figure C.3 - “linkage” function methods (MATLAB®, 2016d)

As the final cluster, the “linkage” function grouped object 8, the newly formed cluster made up of objects 6 and 7, with object 2 from the original data set.

The following Figure C.4 graphically illustrates the way “linkage” groups the objects into a hierarchy of clusters.

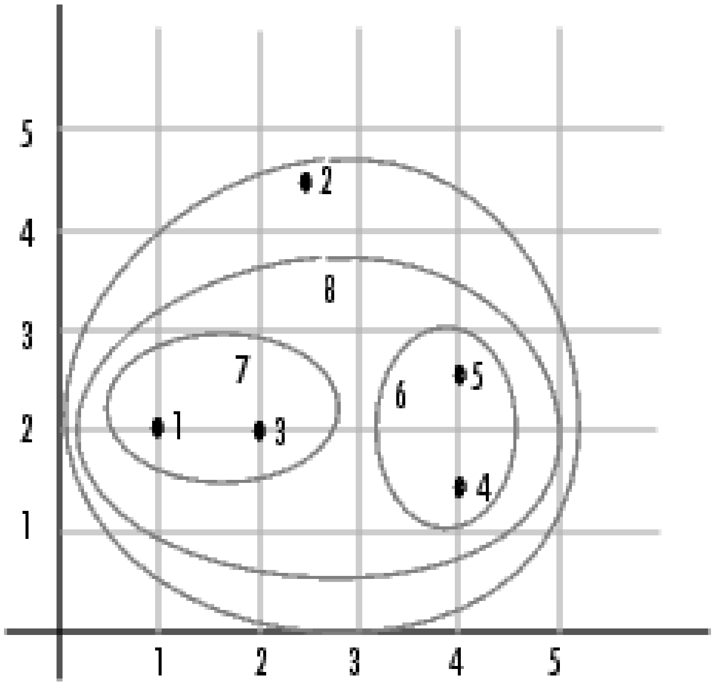


Figure C.4 - Linkage (MATLAB®, 2016b)

The hierarchical, binary cluster tree created by the “linkage” function is most easily understood when viewed graphically, using the “dendogram” function, as can be seen in figure C.5.

dendrogram(Z)

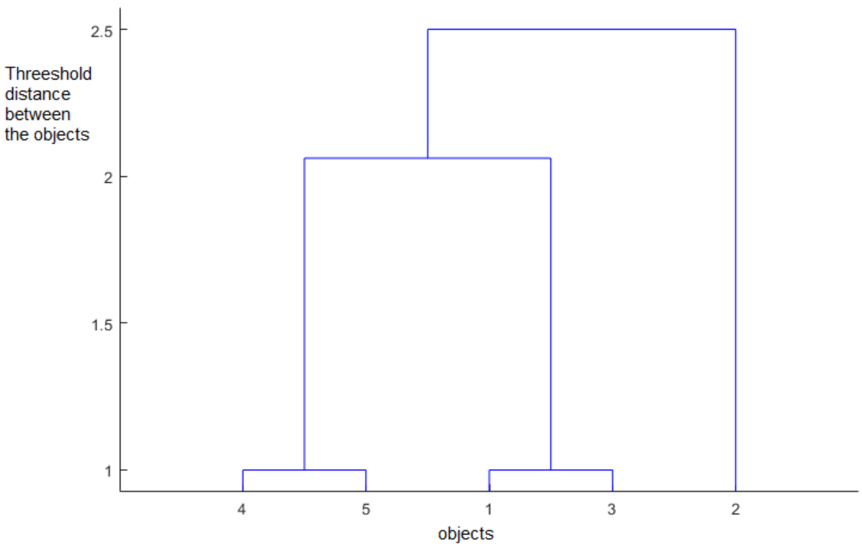


Figure C.5 - Dendogram (MATLAB®, 2016b)

In figure C.5, the numbers along the horizontal axis represent the indices of the objects in the original data set. The links between objects are represented as upside-down U-shaped lines. The height of the U indicates the threeshold distance between the objects. For example, the link representing the cluster containing objects 1 and 3 has a height of 1. The link representing the cluster that groups object 2 together with objects 1, 3, 4, and 5, (which are already clustered as object 8) has a height of 2.5. The height represents the threshold distance “linkage” computes between objects 2 and 8.

After linking the objects in a data set into a hierarchical cluster tree, the distances (that is, heights) in the tree reflect the original distances accurately, indicating the level of dissimilarity between the objects. In a hierarchical cluster tree, any two objects in the original data set are eventually linked together at some level. The height of the link represents the distance between the two clusters that contain those two objects. This height is known as the cophenetic distance between the two objects. One way to measure how well the cluster tree generated by the “linkage” function reflects your data is to compare the cophenetic distances with the original distance data generated by the “pdist” function. If the clustering is valid, the linking of objects in the cluster tree should have a strong correlation with the distances between objects in the distance vector. The “cophenet” function compares these two sets of values and computes their correlation, returning a value called the cophenetic correlation coefficient. The closer the value of the cophenetic correlation coefficient is to 1, the more accurately the clustering solution reflects your data.

In addition, it is possible to investigate natural divisions that exist among links between objects (verify consistency). One way to determine the natural cluster divisions in a data set is to compare the height of each link in a cluster tree with the heights of neighboring links below it in the tree. A link that is approximately the same height as the links below it indicates that there are no distinct divisions between the objects joined at this level of the hierarchy. These links are said to exhibit a high level of consistency, because the distance between the objects being joined is approximately the same as the distances between the objects they contain. On the other hand, a link whose height differs noticeably from the height of the links below it indicates that the objects joined at this level in the cluster tree are much farther apart from each other than their components were when they were joined. This link is said to be inconsistent with the links below it. In cluster analysis, inconsistent links can indicate the border of a natural division in a data set. The following dendrogram illustrates inconsistent links. Note how the objects in the dendrogram fall into two groups that are connected by links at a much higher level in the tree. These links are inconsistent when compared with the links below them in the hierarchy, as shown in figure C.6.

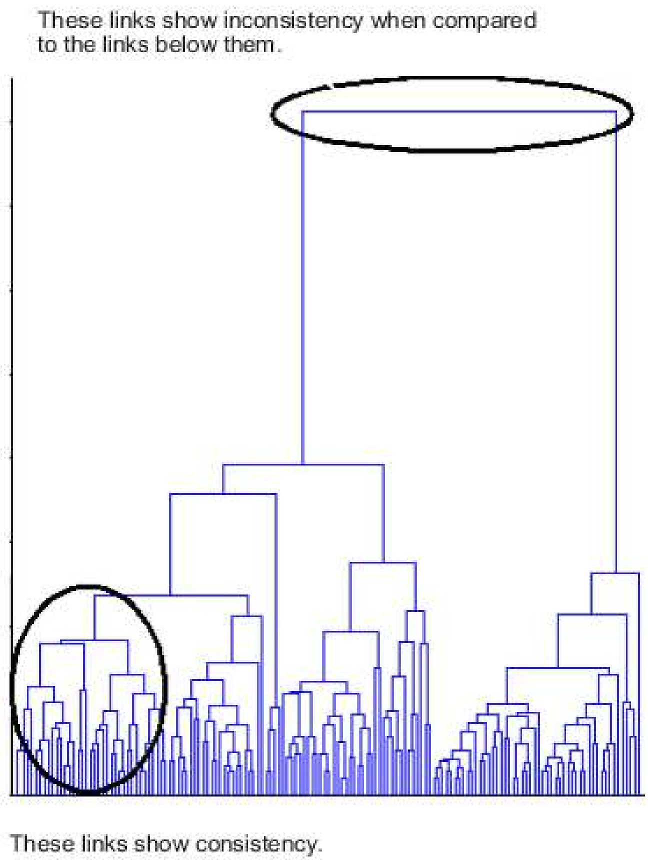


Figure C.6 - Consistency in a dendogram (MATLAB®, 2016b)

c) Step 3 - Determine where to cut the hierarchical tree into clusters. In this step, you use the “cluster” function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster. This creates a partition of the data. The “cluster” function can create these clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point. After you create the hierarchical tree of binary clusters, you can prune the tree to partition your data into clusters using the “cluster” function. The “cluster” function lets you create clusters in two ways, as discussed in the following sections: Find Natural Divisions in Data and Specify Arbitrary Clusters.

Find Natural Divisions in Data - The hierarchical cluster tree may naturally divide the data into distinct, well-separated clusters. This can be particularly evident in a dendrogram diagram created from data where groups of objects are densely packed in certain areas and not in others. The inconsistency coefficient of the links in the cluster tree can identify these divisions where the similarities between objects change abruptly. You can use this value to determine where the “cluster” function creates cluster boundaries.

To evaluate the best number of cluesters, it is interesting to specify arbitrary clusters. Instead of letting the “cluster” function create clusters determined by the natural divisions in the data set, you can specify the number of clusters you want created. For example, you can specify that you want the “cluster” function to partition the sample data set into two clusters. In this case, the “cluster” function creates one cluster containing objects 1, 3, 4, and 5 and another cluster containing object 2. To help you visualize how the cluster function determines these clusters, the following Figure C.7 (a) shows the dendrogram of the hierarchical cluster tree. The horizontal dashed line intersects two lines of the dendrogram, corresponding to setting 'maxclust' to 2. These two lines partition the objects into two clusters: the objects below the left-hand line, namely 1, 3, 4, and 5, belong to one cluster, while the object below the right-hand line, namely 2, belongs to the other cluster. On the other hand, if you set 'maxclust' to 3, the “cluster” function groups objects 4 and 5 in one cluster, objects 1 and 3 in a second cluster, and object 2 in a third cluster. This time, the “cluster” function cuts off the hierarchy at a lower point, corresponding to the horizontal line that intersects three lines of the dendrogram in figure C.7 (b).

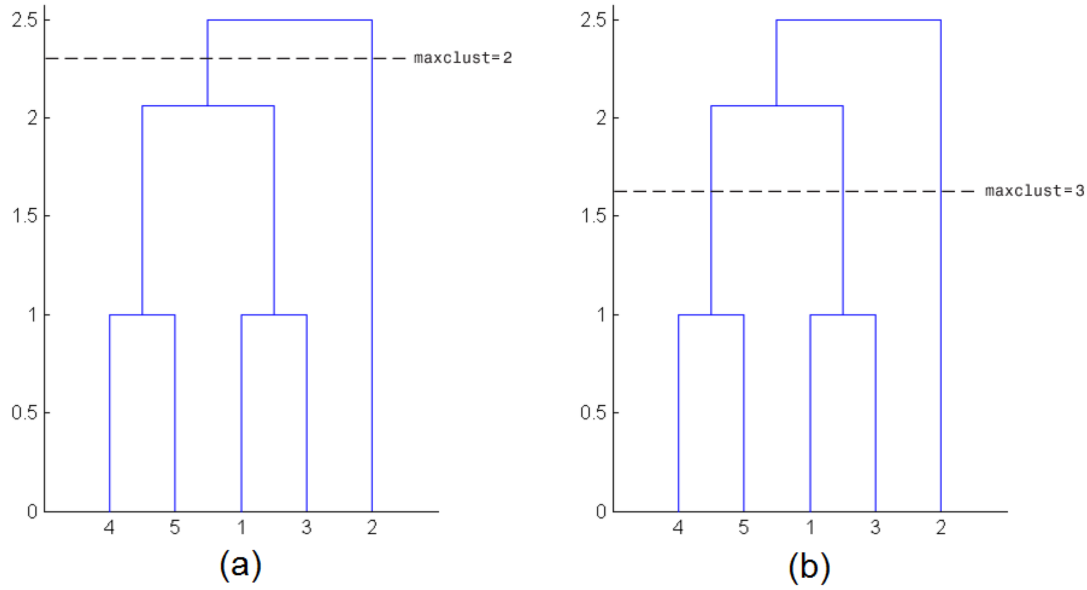


Figure C.7 - Examples of Arbitrary Clusters for: a) 2 clusters; and b) 3 clusters, respectively (MATLAB®, 2016b)

C.1.2 K-means Clustering and Unsupervised Classifier

K-means clustering is a partitioning method of clustering (MATLAB®, 2016e). The “kmeans” function do the partitions of data into k mutually exclusive clusters, and returns the index of the cluster to which it has assigned each observation.

Unlike hierarchical clustering, k-means clustering operates on actual observations (rather than the larger set of dissimilarity measures), and creates a single level of clusters. The distinctions mean that k-means clustering is often more suitable than hierarchical clustering for large amounts of data.

“kmeans” function treats each observation in your data as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. You can choose from five different distance measures, depending on the kind of data you are clustering.

Each cluster in the partition is defined by its member objects and by its centroid, or center. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized. “kmeans” function computes cluster centroids differently for each distance measure, to minimize the sum with respect to the measure that you specify.

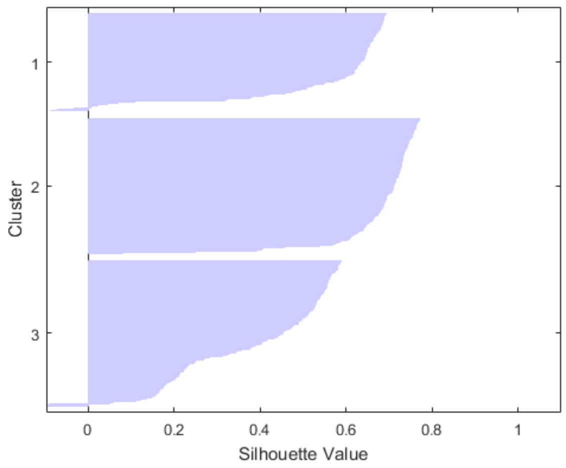
It is possible to control the details of the minimization using several optional input parameters to “kmeans” function , including ones for the initial values of the cluster centroids, and for the maximum number of iterations.

By default, “kmeans” function uses the k-means++ algorithm for cluster center initialization and the squared Euclidean metric to determine distances. In figure C.8 is shown the metric available in Matlab®.

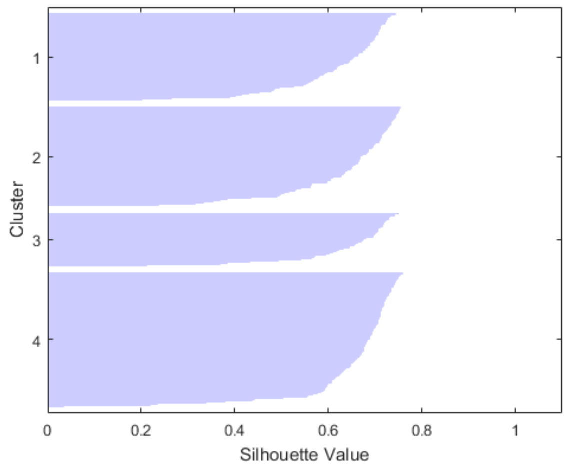
Distance Measure	Description	Formula
'sqeuclidean'	Squared Euclidean distance (default). Each centroid is the mean of the points in that cluster.	$d(x,c) = (x-c)(x-c)'$
'cityblock'	Sum of absolute differences, i.e., the *L*1 distance. Each centroid is the component-wise median of the points in that cluster.	$d(x,c) = \sum_{j=1}^{p} \lvert x_j - c_j \rvert$
'cosine'	One minus the cosine of the included angle between points (treated as vectors). Each centroid is the mean of the points in that cluster, after normalizing those points to unit Euclidean length.	$d(x,c) = 1 - \frac{xc'}{\sqrt{(xx')(cc')}}$
'correlation'	One minus the sample correlation between points (treated as sequences of values). Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.	$d(x,c) = 1 - \frac{(x-\vec{\bar{x}})(c-\vec{\bar{c}})'}{\sqrt{(x-\vec{\bar{x}})(x-\vec{\bar{x}})'}\sqrt{(c-\vec{\bar{c}})(c-\vec{\bar{c}})'}}$, where • $\vec{\bar{x}} = \frac{1}{p}\left(\sum_{j=1}^{p} x_j\right)\vec{1}_p$ • $\vec{\bar{c}} = \frac{1}{p}\left(\sum_{j=1}^{p} c_j\right)\vec{1}_p$ • $\vec{1}_p$ is a row vector of *p* ones.
'hamming'	This measure is only suitable for binary data. It is the proportion of bits that differ. Each centroid is the component-wise median of points in that cluster.	$d(x,y) = \frac{1}{p}\sum_{j=1}^{p} I\{x_j \neq y_j\}$, where *I* is the indicator function.

Figure C.8 - “kmeans” function metrics (MATLAB®, 2016f)

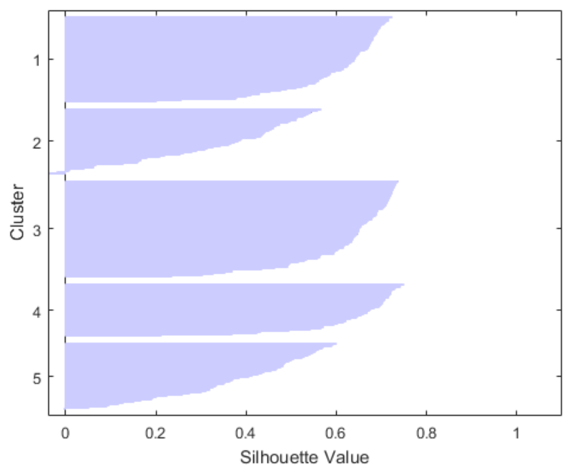
To get an idea of how well-separated the resulting clusters are, you can make a silhouette plot using the cluster indices output from “kmeans” function. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters. This measure ranges from +1, indicating points that are very distant from neighboring clusters, through 0, indicating points that are not distinctly in one cluster or another, to -1, indicating points that are probably assigned to the wrong cluster. silhouette returns these values in its first output. Figure C.9 a), b) and c) show an example of the k-means for 3,4 and 5 clusters for the same data.



(a)



(b)



(c)

Figure C.9 - Silhouette for k-means clustering example for: a) 3 clusters; b) 4 clusters; and c) 5 clusters (MATLAB®, 2016e)

The silhouette plot for 3 clusters allow to infer that most points in the second cluster have a large silhouette value, greater than 0.6, indicating that the cluster is somewhat separated from neighboring clusters. However, the third cluster contains many points with low silhouette values. The first and also the third cluster contain a few points with negative values, indicating that those two clusters are not well separated.

The silhouette plot for 4 clusters indicates that these four clusters are better separated than the three in the previous solution, because the fours clusters have the almost the same silhouette value and all clusters have no negative value.

The silhouette plot for 5 clusters indicates that this is probably not the right number of clusters, since two of the clusters contain points with mostly low silhouette values. Without some knowledge of how many clusters are really in the data, it is a good idea to experiment with a range of values for k.

C.1.3 Gaussian Mixture Models Clustering and Unsupervised Classifier

In the mixture method of clustering, each different group in the population is assumed to be described by a different probability distribution that may belong to the same family but differ in the values they take for the parameters of the distribution. In this study, it was used a Matlab® gaussian probability distribution model.

As described in MATLAB® (2016g), Gaussian mixture models (GMM)

> *...are often used for data clustering. Usually, fitted GMMs cluster by assigning query data points to the multivariate normal components that maximize the component posterior probability given the data. That is, given a fitted GMM, "gmdistribution.cluster" assigns query data to the component yielding the highest posterior probability. This method of assigning a data point to exactly one cluster is called hard clustering. Wherever, GMM clustering is more flexible because you can view it as a fuzzy or soft clustering method. Soft clustering methods assign a score to a data point for each cluster. The value of the score indicates the association strength of the data point to the cluster. As opposed to hard clustering methods, soft clustering methods are flexible in that they can assign a data point to more than one cluster. When clustering with GMMs, the score is the posterior probability. Moreover, GMM clustering can accommodate clusters that have different sizes and correlation structures within them. Because of this, GMM clustering can be more appropriate to use than, e.g, k-means clustering.*

Gaussian mixture models can be used for clustering data, by realizing that the multivariate normal components of the fitted model can represent clusters. For example, in figure C.10, there is a simulate data from a mixture of two bivariate Gaussian distribution. The next step is to fit the Simulated Data to a Gaussian Mixture Model.

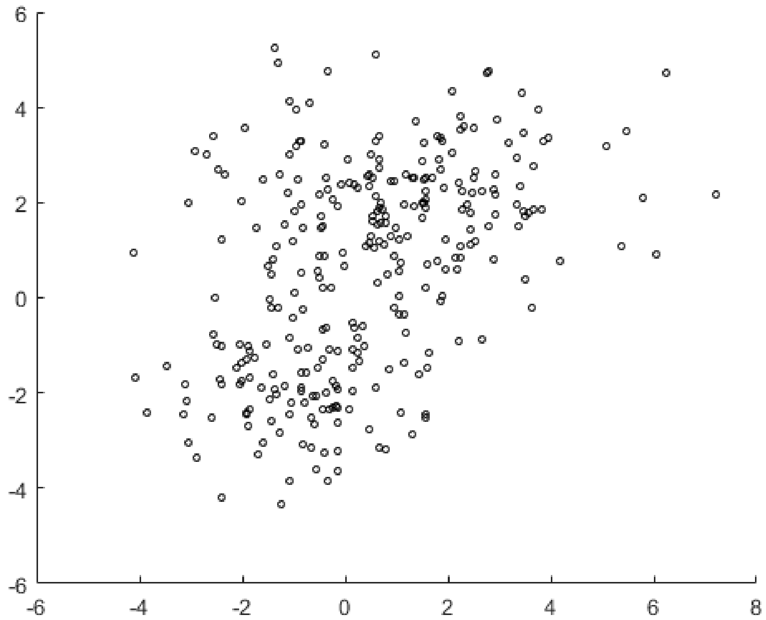


Figure C.10 - Simulate data from a mixture of two bivariate Gaussian distribution (MATLAB®, 2016h)

Fit a two-component Gaussian mixture model (GMM). Here, you know the correct number of components to use. In practice, with real data, this decision would require comparing models with different numbers of components. Also, request to display the final iteration of the expectation-maximization fitting routine.

options = statset('Display','final');

gm = fitgmdist(X,2,'Options',options)

26 iterations, log-likelihood = -1210.59

gm =

Gaussian mixture distribution with 2 components in 2 dimensions

Component 1:

Mixing proportion: 0.629514

Mean: 1.0756 2.0421

Component 2:

Mixing proportion: 0.370486

Mean: -0.8296 -1.8488

The step 2 is plot the estimated probability density contours for the two-component mixture distribution. The two bivariate normal components overlap, but their peaks are distinct. This suggests that the data could reasonably be divided into two clusters, as shown in figure C.11.

```
hold on
ezcontour(@(x,y)pdf(gm,[x y]),[-8 6],[-8 6]);
title('Scatter Plot and Fitted GMM Contour')
hold off
```

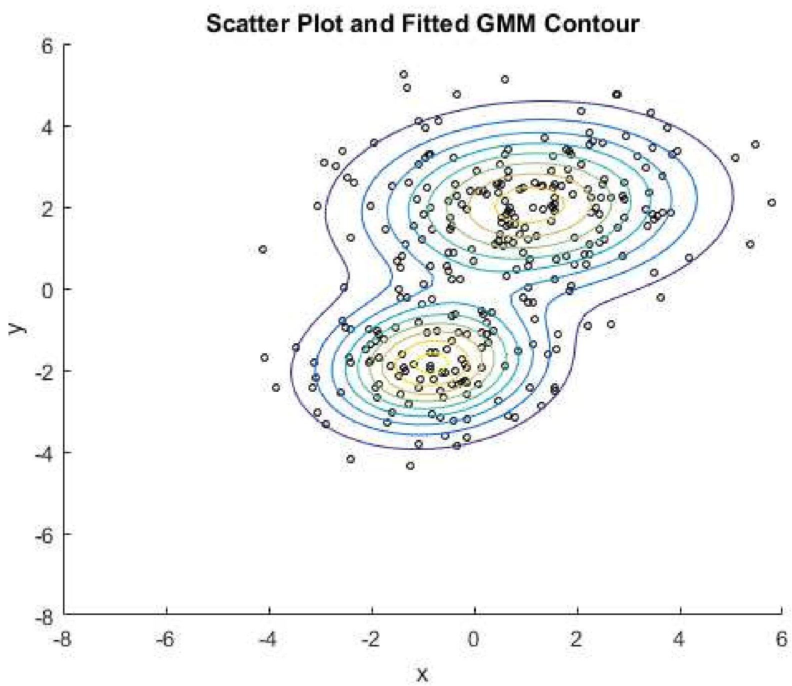


Figure C.11 - Scatter Plot and Fitted GMM Contour (MATLAB®, 2016h)

The step 3 is to cluster the Data Using the Fitted GMM. “gmdistribution.cluster” implements "hard clustering", a method that assigns each data point to exactly one cluster. For GMM, cluster assigns each point to one of the two mixture components in the GMM. The center of each cluster is the corresponding mixture component mean. Partition the data into clusters by passing the fitted GMM and the data to cluster.

```
idx = cluster(gm,X);
```

```
cluster1 = (idx == 1); % |1| for cluster 1 membership
cluster2 = (idx == 2); % |2| for cluster 2 membership
Figure;
gscatter(X(:,1),X(:,2),idx,'rb','+o');
legend('Cluster 1','Cluster 2','Location','NorthWest');
```

In figure C.12, it is shown the result of the scattering plot of the GMM fitted clusters.

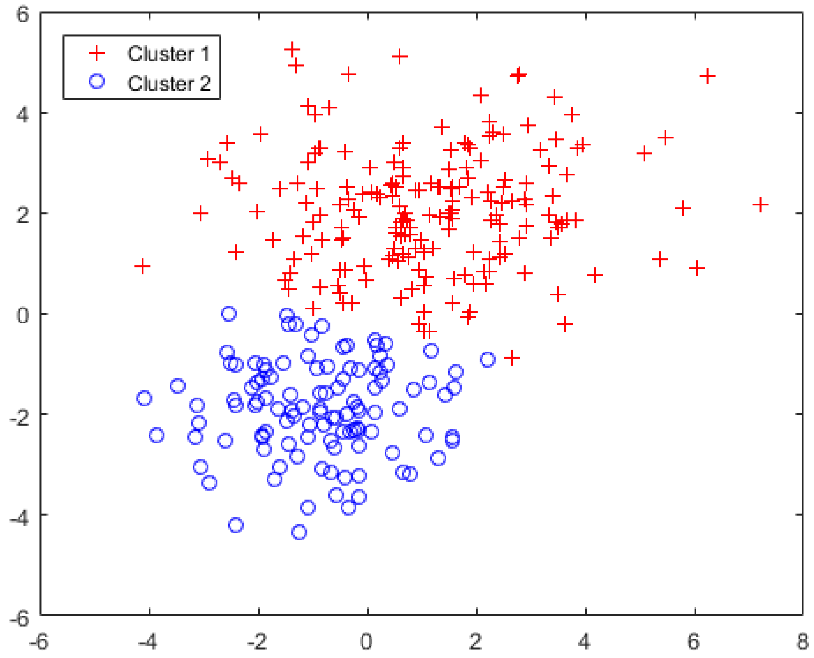


Figure C.12 - Scattering plot of the GMM fitted clusters (MATLAB®, 2016h)

Each cluster corresponds to one of the bivariate normal components in the mixture distribution. cluster assigns data to clusters based on a cluster membership score. Each cluster membership scores is the estimated posterior probability that the data point came from the corresponding component. cluster assigns each point to the mixture component corresponding to the highest posterior probability.

C.2 Supervised Classification

C.2.1 Naïve Bayes Supervised Classifier

Naïve Bayes classifiers is a simple probabilistic classifier method based on Bayes' theorem assuming a strong (naïve) independence between the features. The naïve Bayes Matlab® classifier (MATLAB®, 2016i) is

> *...designed for use when predictors are independent of one another within each class, but it appears to work well in practice even when that independence assumption is not valid. It classifies data in two steps:*
> *Step 1 - Training step: Using the training data, the method estimates the parameters of a probability distribution, assuming predictors are conditionally independent given the class.*
> *Step 2 - Prediction step: For any unseen test data, the method computes the posterior probability of that sample belonging to each class. The method then classifies the test data according the largest posterior probability.*
> *The class-conditional independence assumption greatly simplifies the training step since you can estimate the one-dimensional class-conditional density for each predictor individually. While the class-conditional independence between predictors is not true in general, research shows that this optimistic assumption works well in practice. This assumption of class-conditional independence of the predictors allows the naive Bayes classifier to estimate the parameters required for accurate classification while using less training data than many other classifiers. This makes it particularly effective for data sets containing many predictors.*

The Naive Bayes model (“fitNaiveBayes” function) used in Matlab® use the Bag-of-Tokens Model where the value of predictor j is the nonnegative number of occurrences of token j in this observation. The number of categories (bins) in this multinomial model is the number of distinct tokens, that is, the number of predictors (MATLAB®, 2016j).

Naive Bayes is a classification algorithm that applies density estimation to the data and leverages Bayes theorem, and (naively) assumes that the predictors are conditionally independent, given the class, as already menioned. Though the assumption is usually violated in practice, naive Bayes classifiers tend to yield posterior distributions that are robust to biased class density estimates, particularly where the posterior is 0.5 (the decision boundary), as described by Hastie et al. (2008).

Naive Bayes classifiers assign observations to the most probable class (in other words, the maximum a posteriori decision rule). Explicitly, the algorithm:

a. Estimates the densities of the predictors within each class;

b. Models posterior probabilities according to Bayes rule. That is, for all k = 1,...,K,

$$\hat{P}(Y = k \mid X_1, .., X_P) = \frac{\pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)}{\sum_{k=1}^{K} \pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)}, \qquad (C.1)$$

where:

Y is the random variable corresponding to the class index of an observation.

$X_1,...,X_P$ are the random predictors of an observation.

$\pi(Y=k)$ is the prior probability that a class index is k.

c. Classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

The Naive Bayes model (“fitNaiveBayes” function) used in Matlab® can be performed with normal, gaussian, multinomial, or kernel predictors as shown in figure C.13.

Value	Description
`'kernel'`	Kernel smoothing density estimate.
`'mn'`	Multinomial distribution. If you specify `mn`, then all features are components of a multinomial distribution. Therefore, you cannot include `'mn'` as an element of a cell array of character vectors.
`'mvmn'`	Multivariate multinomial distribution.
`'normal'`	Normal (Gaussian) distribution.

Figure C.13 - Probability distribution values for “fitNaiveBayes” function (MATLAB®, 2016i)

Normal (Gaussian) Distribution - The 'normal' distribution (specify using 'normal') is appropriate for predictors that have normal distributions in each class. For each predictor you model with a normal distribution, the naive Bayes classifier estimates a separate normal distribution for each class by computing the mean and standard deviation of the training data in that class.

Kernel Distribution - The 'kernel' distribution (specify using 'kernel') is appropriate for predictors that have a continuous distribution. It does not require a strong assumption such as a normal distribution and you can use it in cases where the distribution of a predictor may be skewed or have multiple peaks or modes. It requires more computing time and more memory than the normal distribution. For each predictor you model with a kernel distribution, the naive Bayes classifier computes a separate kernel density estimate for each class based on the training data for that class. By default the kernel is the normal kernel, and the classifier selects a width automatically for each class and predictor. The software supports specifying different kernels for each predictor, and different widths for each predictor or class.

Multivariate Multinomial Distribution (MVMN) - The multivariate multinomial distribution (specify using 'mvmn') is appropriate for a predictor whose observations are categorical. Naive Bayes classifier construction using a multivariate multinomial predictor is described below. To illustrate the steps, consider an example where observations are labeled 0, 1, or 2, and a predictor the weather when the sample was conducted.

MN - The multinomial distribution (specify using 'mn') is appropriate when, given the class, each observation is a multinomial random variable.

C.2.2 Multiclass Support Vector Machine(SVM) Supervised Classifier

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other task. As describe in Webb (2002), they map pattern vectors to a highdimensional feature space where a 'best' separating hyperplane (the maximal margin hyperplane) is constructed.

Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

A briefly explanation (MATLAB®, 2016k) about a support vector machine (SVM) it can be done when your data is linearly separable and has exactly two classes. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points.

The support vectors are the data points that are closest to the separating hyperplane; these points are on the boundary of the slab. The following Figure C.14 illustrates these definitions, with + indicating data points of type 1, and - indicating data points of type -1.

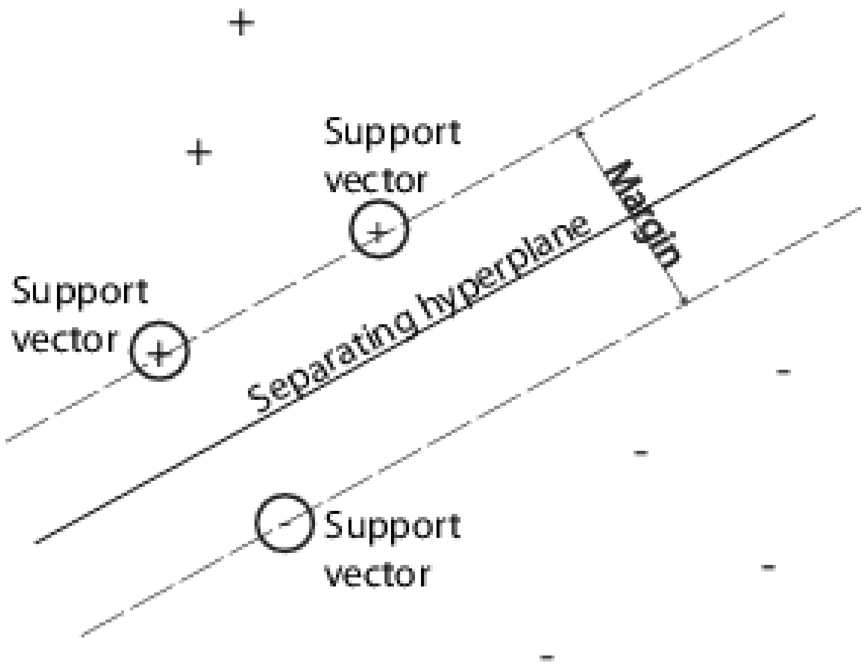


Figure C.14 - Support Vectors (MATLAB®, 2016k)

In the case of nonseparable data, your data might not allow for a separating hyperplane. In that case, SVM can use a soft margin, meaning a hyperplane that separates many, but not all data points.

Some binary classification problems do not have a simple hyperplane as a useful separating criterion. For those problems, there is a variant of the mathematical approach that retains nearly all the simplicity of an SVM separating hyperplane. This approach uses these results from the theory of reproducing kernels:

There is a class of functions $G(x_1,x_2)$ with the following property. There is a linear space S and a function φ mapping x to S such that $G(x_1,x_2) = <\varphi(x_1),\varphi(x_2)>$.

The dot product takes place in the space S.

This class of functions includes:

Linear: $G(x_1,x_2)=x_1'x_2$;

Polynomials: For some positive integer p, $G(x_1,x_2) = (1 + x_1'x_2)^p$;

Radial basis function (rbf or Gaussian): $G(x_1,x_2) = exp(-\|x_1-x_2\|^2)$; and

Multilayer perceptron or sigmoid (neural network): For a positive number p_1 and a negative number p_2. (This is not implemented in ECOC).

Support vector machines (WEBB, 2002) can also be applied in multiclass problems with two optinons: by using the binary classifier in a one-against-all or one-against-one situation, or by constructing linear discriminant functions simultaneously.

In Matlab®, for greater accuracy and kernel-function choices on low- through medium-dimensional data sets, train a binary SVM model or a multiclass error-correcting output codes (ECOC) model containing SVM binary learners using the Classification Learner app. For greater flexibility, use the command-line interface to train a binary SVM model using “fitcsvm” or train a multiclass ECOC model composed of binary SVM learners using “fitcecoc”.

For reduced computation time on high-dimensional data sets that fit in the MATLAB® Workspace, efficiently train a binary, linear classification model, such as a linear SVM model, using fitclinear or train a multiclass ECOC model composed of SVM models using “fitcecoc”. To develop a multiclass SVM classifier, this study used the Matlab® function Error-correcting output codes (ECOC) associate with a method templateSVM.

ClassificationECOC class (MATLAB®, 2016l) has a function which is an error-correcting output codes (ECOC) classifier for multiclass learning by reduction to multiple, binary classifiers such as support vector machines (SVMs). Train a ClassificationECOC classifier using “fitcecoc” and the training data. Trained ClassificationECOC classifiers store the training data, parameter values, prior probabilities, and coding matrices.

t = templateSVM() returns a support vector machine (SVM) learner template suitable for training error-correcting output code (ECOC) multiclass models. If you specify a default template, then the software uses default values for all input arguments during training. Specify t as a binary learner, or one in a set of binary learners, in fitcecoc to train an ECOC multiclass classifer.

t = templateSVM(Name,Value) returns a template with additional options specified by one or more name-value pair arguments. For example, you can specify the box constraint, the kernel function, or whether to standardize the predictors.

The mainly parameters (MATLAB®, 2016m) used in this study were:

- Box Constraint - A parameter that controls the maximum penalty imposed on margin-violating observations, and aids in preventing overfitting (regularization). If you increase the box constraint, then the SVM classifier assigns fewer support vectors. However, increasing the box constraint can lead to longer training times.

- Kernel Function - Kernel function is used to compute the Gram matrix, specified as the comma-separated pair consisting of ‘KernelFunction’. The Gram matrix of a set of n vectors $\{x_1,..,x_n; x_j \in R^p\}$ is an n-by-n matrix with element (j,k) defined as $G(x_j,x_k) = <\phi(x_j),\phi(x_k)>$ an inner product of the transformed predictors using the kernel function ϕ. For nonlinear SVM, the algorithm forms a Gram matrix using the predictor matrix columns. The dual formalization replaces the inner product of the predictors with corresponding elements of the resulting Gram matrix (called the “kernel trick”). Subsequently, nonlinear SVM operates in the transformed predictor space to find a separating hyperplane. The kernel functions available for this method are ‘linear’, ‘gaussian’ or ‘rbf’, and ‘polynomial’; and

- Standardize - This parameter is a flag to standardize the predictor data, specified as the comma-separated pair consisting of ‘Standardize’ and true (1) or false (0). If you set ‘Standardize’,true, the software centers and scales each column of the predictor data (X) by the weighted column mean and standard deviation, respectively. MATLAB® does not standardize the data contained in the dummy variable columns generated for categorical predictors. The software trains the classifier using the standardized predictor matrix, but stores the unstandardized data in the classifier property X.

Concerning the Kernel Function, figure C.15 shows the options available.

Value	Description	Formula
'gaussian' or 'rbf'	Gaussian or Radial Basis Function (RBF) kernel, default for one-class learning	$G(x_1, x_2) = \exp(-\|x_1 - x_2\|^2)$
'linear'	Linear kernel, default for two-class learning	$G(x_1, x_2) = x_1' x_2$
'polynomial'	Polynomial kernel. Use 'PolynomialOrder',*p* to specify a polynomial kernel of order *p*.	$G(x_1, x_2) = (1 + x_1' x_2)^p$

Figure C.15 - “templateSVM” function kernel functions (Matlab®, 2016m)

C.2.3 Neural Network Supervised Classifier

Neural networks are based on a large collection of neural units loosely modeling the way a biological brain solves problems with large clusters of biological neurons connected by axons. Each neural unit is connected with many others, and links can be enforcing or inhibitory in their effect on the activation state of connected neural units. Each individual neural unit may have a summation function which combines the values of all its inputs together.

There may be a threshold function or limiting function on each connection and on the unit itself such that it must surpass it before it can propagate to other neurons. These systems are self-learning and trained rather than explicitly programmed and excel in areas where the solution or feature detection is difficult to express in a traditional computer program.

Neural networks typically consist of multiple layers or a cube design, and the signal path traverses from front to back. Back propagation is where the forward stimulation is used to reset weights on the "front" neural units and this is sometimes done in combination with training where the correct result is known. More modern networks are a bit more free flowing in terms of stimulation and inhibition with connections interacting in a much more chaotic and complex fashion. Dynamic neural networks are the most advanced in that they dynamically can, based on rules, form new connections and even new neural units while disabling others.

The goal of the neural network is to solve problems in the same way that the human brain would, although several neural networks are much more abstract. Modern neural network projects typically work with a few thousand to a few million neural units and millions of connections, which is still several orders of magnitude less complex than the human brain and closer to the computing power of a worm.

In this study, it was used the Matlab®'s Neural Network Toolbox™. This toolbox provides

> ...algorithms, functions, and apps to create, train, visualize, and simulate neural networks. You can perform classification, regression, clustering, dimensionality reduction, time-series forecasting, and dynamic system modeling and control. The toolbox includes convolutional neural network and autoencoder deep learning algorithms for image classification and feature learning task. To speed up training of large data sets, you can distribute computations and data across multicore processors, GPUs, and computer clusters using Parallel Computing Toolbox™.
>
> Neural Network Toolbox™ supports a variety of supervised and unsupervised network architectures. With the toolbox's modular approach to building networks, you can develop custom network architectures for your specific problem. You can view the network architecture including all inputs, layers, outputs, and interconnections.
>
> Supervised neural networks are trained to produce desired outputs in response to sample inputs, making them particularly well-suited to modeling and controlling dynamic systems, classifying noisy data, and predicting future events. Neural Network Toolbox includes four types of supervised networks: feedforward, radial basis, dynamic and learning vector quantization (Matlab®, 2016n).

The archictecture of the neural network provided by the tollbox is shown n figure C.16.

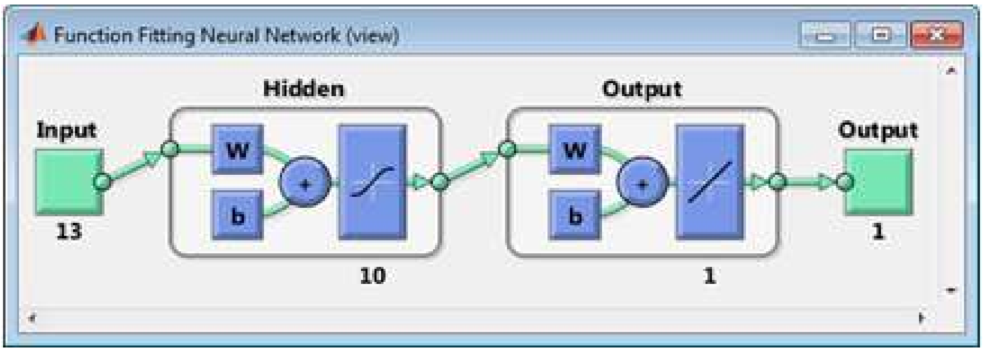


Figure C.16 - A two-layer feedforward network with sigmoid hidden neurons and linear output neurons. This type of network can fit multidimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer (Matlab®, 2016n)

The mainly parameters (MATLAB®, 2016o) used in this study were:

a) Number of hidden layers (“hiddenLayerSize”) - This property defines the number of hidden neurons of the neural network;

b) Neural Network Input-Output Processing Functions (“net.input.processFcns” and “net.output.processFcns”) - This property defines the pre-processing and pos-processing functions for the neural network. The options are in figure C.17.

Function	Algorithm
mapminmax	Normalize inputs/targets to fall in the range [−1, 1]
mapstd	Normalize inputs/targets to have zero mean and unity variance
processpca	Extract principal components from the input vector
fixunknowns	Process unknown inputs
removeconstantrows	Remove inputs/targets that are constant

Figure C.17 - Neural Network pre-processing and postprocessing functions (MATLAB®, 2016p).

c) Setup Division of Data for Training, Validation, Testing (“net.divideFcn “) - This property defines the data division function to be used when the network is trained using a supervised algorithm, such as backpropagation. The options are in figure C.18.

Function	Algorithm
dividerand	Divide the data randomly (default)
divideblock	Divide the data into contiguous blocks
divideint	Divide the data using an interleaved selection
divideind	Divide the data by index

Figure C.18 - Neural Network Setup Division of Data for Training, Validation, Testing functions (MATLAB®, 2016q).

d) Divide Mode (“net.divideMode”) - This property defines the target data dimensions which to divide up when the data division function is called. Its default value is 'sample' for static networks and 'time' for dynamic networks. It may also be set to 'sampletime' to divide targets by both sample and timestep, 'all' to divide up targets by every scalar value, or 'none' to not divide up data at all (in which case all data is used for training, none for validation or testing).

e) Set up Division of Data for Training, Validation, Testing (“net.divideParam.trainRatio”, “net.divideParam.valRatio”, and “net.divideParam.testRatio”) - This property defines the size proportion in relation of the total amount of data for the Training, Validation and Test essays. As already mentioned, for all classifiers it was used 1/3 for all essays.

f) Multilayer Neural Network Training Function (“net.trainFcn”) - This property defines the function that will be used to train the neural network. The options available are:

“trainlm” - Levenberg-Marquardt backpropagation;

“trainbr” - Bayesian Regulation backpropagation;

“trainbfg”- BFGS quasi-Newton backpropagation;

“traincgb”- Conjugate gradient backpropagation with Powell-Beale restarts;

“traincgf”- Conjugate gradient backpropagation with Fletcher-Reeves updates;

“traincgp” - Conjugate gradient backpropagation with Polak-Ribiere updates;

“traingd” - Gradient descent backpropagation;

“traingda” - Gradient descent with adaptive lr backpropagation;

“traingdm” - Gradient descent with momentum;

“traingdx” - Gradient descent w/momentum & adaptive lr backpropagation;

“trainoss” - One step secant backpropagation;

“trainrp” - RPROP (resilient backpropagation) backpropagation;

“trainscg” - Scaled conjugate gradient backpropagation.

“trainb” - Batch training with weight & bias learning rules;

“trainc” - Cyclical order weight/bias training;

“trainr” - Random order weight/bias training;

“trains” - Sequential order weight/bias training;

“trainbu” - Unsupervised batch training with weight & bias learning rules; and

“trainru” - Unsupervised random order weight/bias training; and

g) Neural Network Performance Function (“net.performFcn”) - This property calculates a network performance given targets and outputs, with optional performance weights and other parameters. The options available are:

“crossentropy” - cross entropy function;

“mae” - Mean absolute error performance function;

“mse” - Mean squared normalized error performance function;

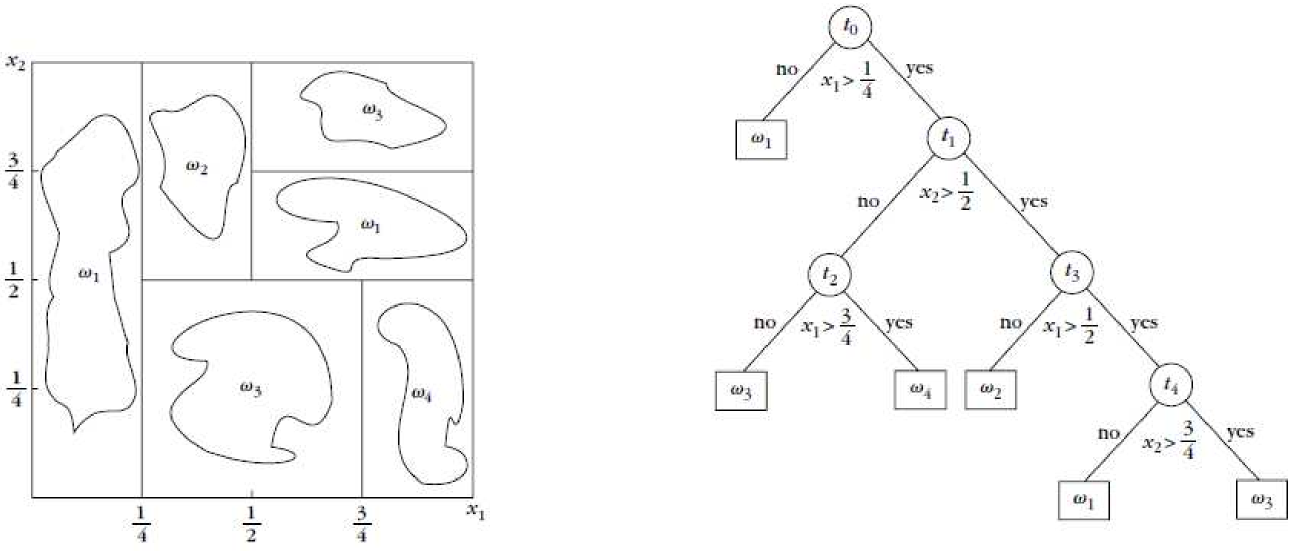
“sse” - Sum squared error performance function; and

“sae” - Sum absolute error performance function.

C.2.4 Decision Tree and Random Forest methods

Decision tree learning uses a decision tree as a predictive model which maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves).

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making as can be seen in figure C.19.



(a) (b)

Figure C.19 - a) Example case; and their b) decision tree (THEODORIDIS, 2009)

The example ilustred (THEODORIDIS, 2009) in figure C.19 is a simple one in the two-dimensional space.There is 2 variable (x_1 and x_2) and 6 cluster areas with 4 classes (ω_1, ω_2, ω_3 and ω_4). The thresholds used for the binary splits at each node of the tree were defined by the observation of the geometry of the clusters. However, this may be not possible in complex examples with higher dimensional spaces.

Decision trees learning comes closest to meeting the requirements for serving as an off-the-shelf procedure for data mining (HASTIE et al., 2008), because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate. In particular, trees that are grown very deep tend to learn highly irregular patterns: they overfit their training sets, having low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance of the final model.

The general method of random decision forests was first proposed by Ho (1995), who established that forests of trees splitting with oblique hyperplanes, if randomly restricted to be sensitive to only selected feature dimensions, can gain accuracy as they grow without suffering from overtraining. The introduction of random forests proper was first made in a paper by Leo Breiman (2001), using out-of-bag error as an estimate of the generalization error and measuring variable importance through permutation.

An example of this kind of approach is the classification and regression tree (CART) model that uses an expansion into indicator functions of multidimensional rectangles. In this study, the CART classifier method used is the ensemble method of Random Forest. Random forests or random decision forests are an ensemble learning method for classification, regression and other task, that operate by constructing a big amount of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Another definition provided by Louppe (2015) is

> *"Random forests form a family of methods that consist in building an ensemble (or forest) of decision trees grown from a randomized variant of the tree induction algorithm. Decision trees are indeed ideal candidates for ensemble methods since they usually have low bias and high variance, making them very likely to benefit from the averaging process."*

In Matlab®, there are several methods for melding results from many weak learners into one high-quality ensemble predictor. These methods closely follow the same syntax, so you can try different methods with minor changes in your commands.

In Matlab®, to create an ensemble for classification or regression is usedthe "fitensemble" function (Matlab®, 2016r). To train an ensemble using "fitensemble", the syntax used (Matlab®, 2016s) is:

Model = fitensemble(X, Y, method, NLearns, learners, type)

Where:

- X is the matrix of data. Each row contains one observation, and each column contains one predictor variable.

- Y is the vector of responses, with the same number of observations as the rows in X.

- “method” is a character vector, naming the type of ensemble.

- “NLearns” is the number of ensemble learning cycles, specified as a positive integer.

“- learners” is either a character vector, naming a weak learner, a weak learner template, or a cell array of such templates. Weak learners to use in the ensemble, specified as a weak-learner name, weak-learner template object, or cell array of weak-learner template objects.

- “type”: is the supervised learning type. In this study, the option is 'classification'.

In figure C.20 is shown the information you need to create an ensemble:

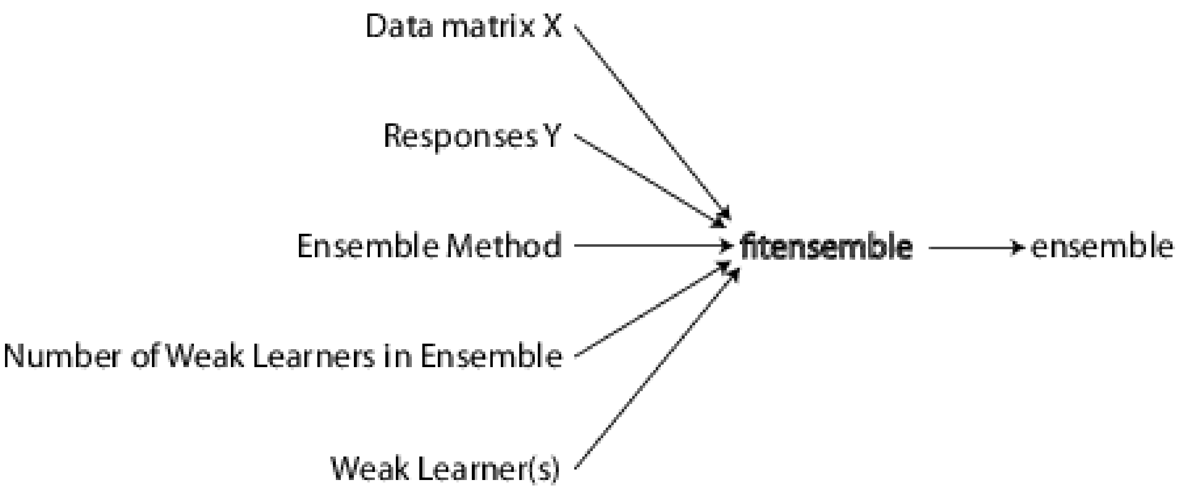


Figure C.20 - Information to create an ensemble (Matlab®, 2016r).

Appendix D - Words Task Data Organization and Coding

Mean Amplitude Between two fixed latencies	Peak Amplitude	Peak Latency	ROI	ERP time range	Subject	Class
-2,049	2,227	254	1	1	2	1
-1,794	-0,304	196	1	1	3	1
1,099	4,893	260	1	1	4	1
-0,339	2,548	294	1	1	5	1
-2,309	3,372	266	1	1	6	1
-0,589	2,96	180	1	1	7	1
1,438	5,267	206	1	1	9	1
3,277	9,979	274	1	1	10	1
1,645	4,17	200	1	1	13	1
-0,572	1,157	260	1	1	15	1
-0,231	5,191	300	1	1	16	1
-4,697	3,02	264	1	1	17	1
0,569	5,816	286	1	1	18	1
-3,697	0,597	268	1	1	19	1
-1,901	4,044	286	1	1	20	1
-5,053	-1,599	174	1	1	21	1
-1,269	5,488	242	1	1	2	2
-1,494	0,134	276	1	1	3	2
2,331	5,738	260	1	1	4	2
-2,397	1,224	268	1	1	5	2
-4,363	-1,763	276	1	1	6	2
2,271	7,378	276	1	1	7	2
-0,098	4,054	248	1	1	9	2
4,338	11,488	264	1	1	10	2
1,906	5,742	192	1	1	13	2
0,754	2,667	284	1	1	15	2
-0,892	3,874	276	1	1	16	2
-2,577	4,971	264	1	1	17	2
-7,044	-2,534	282	1	1	18	2
1,761	5,636	274	1	1	19	2
0,347	3,441	286	1	1	20	2
-5,779	-3,377	184	1	1	21	2
-1,457	2,997	238	1	1	2	3
-3,11	-0,693	300	1	1	3	3
1,816	6,794	254	1	1	4	3
-3,471	-0,145	206	1	1	5	3
-5,461	-1,73	282	1	1	6	3
-1,106	3,226	294	1	1	7	3
-2,022	1,784	256	1	1	9	3
0,886	8,144	268	1	1	10	3
1,686	5,304	198	1	1	13	3

-1,03	1,036	258	1	1	15	3
-2,924	1,81	298	1	1	16	3
-3,012	4,155	254	1	1	17	3
-5,307	-2,033	234	1	1	18	3
-3,711	0,62	282	1	1	19	3
-2,1	-0,184	254	1	1	20	3
-4,976	-2,148	188	1	1	21	3
-2,106	2,47	244	1	1	2	4
-3,4	-0,606	270	1	1	3	4
-0,167	3,328	250	1	1	4	4
-3,744	-2,241	256	1	1	5	4
-5,743	-0,514	278	1	1	6	4
-1,939	1,662	280	1	1	7	4
-0,617	1,326	202	1	1	9	4
-0,496	5,208	258	1	1	10	4
2,164	5,896	192	1	1	13	4
-2,047	0,012	252	1	1	15	4
-3,906	1,926	294	1	1	16	4
-4,19	2,522	256	1	1	17	4
-2,102	2,436	268	1	1	18	4
-1,307	1,526	214	1	1	19	4
-2,476	-0,558	268	1	1	20	4
-4,877	-0,689	174	1	1	21	4
-1,816	1,544	252	2	1	2	1
0,509	1,69	280	2	1	3	1
1,977	5,518	254	2	1	4	1
0,87	3,215	292	2	1	5	1
-0,984	4,302	266	2	1	6	1
-2,821	0,202	180	2	1	7	1
2,331	5,747	212	2	1	9	1
2,418	7,444	264	2	1	10	1
2,933	4,62	248	2	1	13	1
-0,307	1,027	196	2	1	15	1
0,835	5,498	300	2	1	16	1
-2,401	4,255	262	2	1	17	1
0,165	2,328	288	2	1	18	1
-2,158	0,114	282	2	1	19	1
-1,951	2,329	284	2	1	20	1
-1,882	1,89	190	2	1	21	1
0,354	5,796	244	2	1	2	2
0,756	1,906	276	2	1	3	2
2,791	5,851	228	2	1	4	2
-2,291	0,73	296	2	1	5	2
-3,869	-0,939	274	2	1	6	2
-1,058	1,951	296	2	1	7	2
2,626	6,066	252	2	1	9	2
2,304	7,847	262	2	1	10	2
1,781	4,704	190	2	1	13	2

0,472	1,995	202	2	1	15	2
0,461	4,205	274	2	1	16	2
-0,342	6,041	260	2	1	17	2
-4,962	-1,649	248	2	1	18	2
3,061	5,684	274	2	1	19	2
0,444	2,074	232	2	1	20	2
-2,795	0,898	182	2	1	21	2
-0,473	3,267	238	2	1	2	3
-0,04	1,16	288	2	1	3	3
1,64	5,728	238	2	1	4	3
0,039	2,357	204	2	1	5	3
-4,02	-0,689	282	2	1	6	3
-0,691	3,55	294	2	1	7	3
-0,249	2,45	210	2	1	9	3
0,546	5,179	250	2	1	10	3
1,243	3,683	214	2	1	13	3
-0,161	0,701	204	2	1	15	3
-0,593	2,88	300	2	1	16	3
-0,698	5,586	252	2	1	17	3
-2,99	-1,737	288	2	1	18	3
-2,563	0,649	266	2	1	19	3
-1,658	-0,139	236	2	1	20	3
-1,867	1,537	192	2	1	21	3
-1,855	2,081	242	2	1	2	4
-0,343	1,826	272	2	1	3	4
0,409	3,267	246	2	1	4	4
-3,291	-1,851	214	2	1	5	4
-3,736	0,735	278	2	1	6	4
-1,481	0,078	294	2	1	7	4
2,095	4,328	256	2	1	9	4
-0,634	4,274	248	2	1	10	4
2,208	4,392	212	2	1	13	4
-1,763	-0,293	256	2	1	15	4
-0,779	4,136	296	2	1	16	4
-2,142	3,603	254	2	1	17	4
-1,655	1,669	250	2	1	18	4
-0,592	1,649	214	2	1	19	4
-2,306	-1,347	252	2	1	20	4
-3,037	2,233	174	2	1	21	4
-1,515	1,111	252	3	1	2	1
1,366	2,334	280	3	1	3	1
2,028	4,962	254	3	1	4	1
1,501	3,865	292	3	1	5	1
-0,096	4,975	266	3	1	6	1
-2,787	1,692	300	3	1	7	1
2,284	4,585	214	3	1	9	1
1,601	5,956	266	3	1	10	1
3,592	5,38	248	3	1	13	1

-26,187	-21,824	198	3	1	15	1
1,916	5,457	300	3	1	16	1
-1,372	4,373	260	3	1	17	1
0,22	1,799	234	3	1	18	1
-1,757	-0,169	202	3	1	19	1
-1,747	1,432	284	3	1	20	1
-0,642	3,399	190	3	1	21	1
0,821	4,869	246	3	1	2	2
1,322	2,306	278	3	1	3	2
2,809	5,389	228	3	1	4	2
-2,501	0,472	296	3	1	5	2
-2,438	0,912	258	3	1	6	2
-1,049	2,557	298	3	1	7	2
2,739	5,385	254	3	1	9	2
1,222	6,226	262	3	1	10	2
2,782	5,406	190	3	1	13	2
4,227	5,606	288	3	1	15	2
0,681	3,797	286	3	1	16	2
0,145	6,199	258	3	1	17	2
-4,109	-1,551	246	3	1	18	2
2,752	4,761	274	3	1	19	2
0,46	1,993	230	3	1	20	2
-2,657	1,379	180	3	1	21	2
-0,164	2,402	240	3	1	2	3
0,86	1,488	272	3	1	3	3
1,634	4,967	238	3	1	4	3
0,11	2,05	200	3	1	5	3
-2,565	0,289	282	3	1	6	3
-1,142	2,734	296	3	1	7	3
-0,179	1,853	212	3	1	9	3
0,254	3,976	250	3	1	10	3
1,405	3,971	228	3	1	13	3
4,006	5,03	204	3	1	15	3
0,056	2,832	278	3	1	16	3
0,455	6,043	252	3	1	17	3
-1,777	0,094	168	3	1	18	3
-2,165	-0,137	280	3	1	19	3
-2,158	-0,399	234	3	1	20	3
-1,567	1,906	170	3	1	21	3
-1,696	1,558	242	3	1	2	4
0,713	2,411	274	3	1	3	4
0,572	2,854	244	3	1	4	4
-3,137	-2,273	212	3	1	5	4
-2,148	2,013	276	3	1	6	4
-1,752	-0,041	298	3	1	7	4
2,436	4,611	256	3	1	9	4
-0,684	3,531	246	3	1	10	4
2,19	4,436	222	3	1	13	4

-14,977	-13,056	166	3	1	15	4
0,016	3,547	298	3	1	16	4
-1,351	3,706	254	3	1	17	4
-1,588	2,403	244	3	1	18	4
-0,288	1,575	214	3	1	19	4
-2,817	-1,998	234	3	1	20	4
-3,057	2,725	174	3	1	21	4
-2,239	-0,674	168	4	1	2	1
1,065	3,175	298	4	1	3	1
0,581	1,888	228	4	1	4	1
2,376	4,842	292	4	1	5	1
-1,101	2,541	258	4	1	6	1
-5,267	-2,074	300	4	1	7	1
-0,774	3,059	298	4	1	9	1
-4,814	-0,839	294	4	1	10	1
2,411	5,237	248	4	1	13	1
-1,338	0,361	240	4	1	15	1
0,581	1,812	230	4	1	16	1
0,603	3,659	262	4	1	17	1
-0,759	2,868	234	4	1	18	1
-3,109	-2,323	282	4	1	19	1
-3,639	-0,186	152	4	1	20	1
-0,955	3,957	290	4	1	21	1
-0,925	1,205	274	4	1	2	2
0,504	1,885	164	4	1	3	2
1,078	2,577	224	4	1	4	2
-2,622	0,451	162	4	1	5	2
-2,286	0,12	252	4	1	6	2
-3,311	0,341	296	4	1	7	2
0,203	3,705	272	4	1	9	2
-5,624	-1,966	282	4	1	10	2
1,84	3,649	268	4	1	13	2
-1,508	0,001	162	4	1	15	2
0,143	2,058	152	4	1	16	2
-0,375	3,617	258	4	1	17	2
-1,853	0,905	242	4	1	18	2
2,076	5,62	154	4	1	19	2
-3,155	-2,037	204	4	1	20	2
-3,125	1,675	294	4	1	21	2
-0,04	1,119	276	4	1	2	3
0,467	2,608	160	4	1	3	3
0,343	1,703	236	4	1	4	3
0,203	3,07	152	4	1	5	3
-1,435	-0,106	284	4	1	6	3
-3,935	-3,014	254	4	1	7	3
-2,091	-0,497	298	4	1	9	3
-5,221	-1,656	282	4	1	10	3
-0,527	2,134	228	4	1	13	3

-1,46	0,486	152	4	1	15	3
1,468	2,39	216	4	1	16	3
1,569	4,319	254	4	1	17	3
-0,608	1,011	166	4	1	18	3
-3,173	-0,117	160	4	1	19	3
-4,222	-0,208	164	4	1	20	3
-3,066	0,921	290	4	1	21	3
-2,451	-1,521	258	4	1	2	4
0,531	1,811	286	4	1	3	4
-0,576	0,219	230	4	1	4	4
-1,687	0,695	150	4	1	5	4
-2,402	0,531	266	4	1	6	4
-4,407	-2,891	300	4	1	7	4
0,376	3,326	288	4	1	9	4
-5,105	-2,163	274	4	1	10	4
-0,55	2,933	252	4	1	13	4
-1,888	-0,463	258	4	1	15	4
-0,562	-0,255	230	4	1	16	4
-1,319	1,133	256	4	1	17	4
-2,378	1,569	242	4	1	18	4
-1,566	-1,473	258	4	1	19	4
-5,436	-0,188	150	4	1	20	4
-5,116	-1,209	160	4	1	21	4
0,117	3,795	254	5	1	2	1
-1,602	-0,334	196	5	1	3	1
1,665	4,722	260	5	1	4	1
-0,206	3,719	294	5	1	5	1
-1,36	2,506	268	5	1	6	1
0,758	3,415	266	5	1	7	1
2,789	6,627	208	5	1	9	1
6,099	12,968	274	5	1	10	1
2,639	5,821	200	5	1	13	1
75,748	81,724	276	5	1	15	1
0,663	6,324	300	5	1	16	1
-1,421	4,746	248	5	1	17	1
-0,071	2,168	286	5	1	18	1
-4,578	-1,094	284	5	1	19	1
-1,351	2,896	286	5	1	20	1
-1,774	1,004	256	5	1	21	1
-1,066	4,002	242	5	1	2	2
-1,641	-0,158	242	5	1	3	2
2,963	5,97	262	5	1	4	2
-2,851	0,962	300	5	1	5	2
-2,758	0,238	280	5	1	6	2
0,501	4,344	274	5	1	7	2
-0,153	2,602	200	5	1	9	2
4,72	9,925	278	5	1	10	2
2,123	6,498	192	5	1	13	2

46,409	54,314	286	5	1	15	2
1,924	5,942	274	5	1	16	2
0,034	6,106	246	5	1	17	2
-5,397	-2,336	248	5	1	18	2
1,037	4,748	274	5	1	19	2
-0,465	2,309	286	5	1	20	2
-4,053	-2,033	200	5	1	21	2
1,755	5,128	252	5	1	2	3
-2,287	-0,16	300	5	1	3	3
2,606	6,887	270	5	1	4	3
-3,907	-1,996	204	5	1	5	3
-4,768	-1,714	282	5	1	6	3
-1,396	2,683	294	5	1	7	3
0,37	3,165	256	5	1	9	3
2,598	8,431	268	5	1	10	3
1,215	5,025	200	5	1	13	3
-43,784	-38,101	158	5	1	15	3
-1,777	2,154	298	5	1	16	3
-1,319	4,64	254	5	1	17	3
-4,881	-1,239	220	5	1	18	3
-3,349	0,708	282	5	1	19	3
-2,095	-0,442	254	5	1	20	3
-3,655	-1,121	254	5	1	21	3
0,98	4,383	244	5	1	2	4
-2,964	-0,662	268	5	1	3	4
0,984	3,758	250	5	1	4	4
-2,886	-0,86	256	5	1	5	4
-3,265	1,352	280	5	1	6	4
0,458	3,684	280	5	1	7	4
1,453	3,635	238	5	1	9	4
1,472	5,896	258	5	1	10	4
2,606	6,209	192	5	1	13	4
-24,901	-19,076	298	5	1	15	4
-2,335	4,24	296	5	1	16	4
-2,518	2,532	256	5	1	17	4
-1,816	0,817	280	5	1	18	4
-1,256	1,749	214	5	1	19	4
-1,684	0,113	270	5	1	20	4
-1,31	1,101	174	5	1	21	4
0,509	2,593	252	6	1	2	1
0,559	1,861	266	6	1	3	1
2,296	4,942	260	6	1	4	1
1,051	2,974	294	6	1	5	1
1,487	4,823	266	6	1	6	1
-1,207	0,649	180	6	1	7	1
2,474	5,087	212	6	1	9	1
5,016	9,456	264	6	1	10	1
4,128	6,451	210	6	1	13	1

0,599	1,875	242	6	1	15	1
3,296	5,936	300	6	1	16	1
1,298	5,694	246	6	1	17	1
-0,584	0,885	188	6	1	18	1
-3,226	-0,359	282	6	1	19	1
-1,361	0,692	286	6	1	20	1
1,448	4,196	196	6	1	21	1
0,605	2,921	244	6	1	2	2
0,557	1,481	274	6	1	3	2
2,203	4,589	270	6	1	4	2
-1,284	0,851	298	6	1	5	2
-0,777	1,751	272	6	1	6	2
-0,783	1,778	276	6	1	7	2
1,816	3,554	270	6	1	9	2
3,183	6,744	264	6	1	10	2
2,853	5,601	192	6	1	13	2
1,41	2,685	202	6	1	15	2
4,02	6,77	274	6	1	16	2
1,906	6,654	260	6	1	17	2
-2,126	-0,713	266	6	1	18	2
4,989	8,583	274	6	1	19	2
-0,057	1,189	234	6	1	20	2
-0,683	1,586	200	6	1	21	2
0,598	2,194	252	6	1	2	3
0,392	1,209	266	6	1	3	3
2,014	4,773	270	6	1	4	3
-1,262	-0,027	152	6	1	5	3
-0,656	1,819	282	6	1	6	3
-1,888	0,356	294	6	1	7	3
1,01	3,424	258	6	1	9	3
2,126	5,836	268	6	1	10	3
0,604	3,961	214	6	1	13	3
1,288	2,449	258	6	1	15	3
1,004	3,353	300	6	1	16	3
0,506	5,314	252	6	1	17	3
-1,157	0,829	234	6	1	18	3
-1,109	1,495	264	6	1	19	3
-1,585	-0,687	220	6	1	20	3
-0,002	2,174	206	6	1	21	3
0,191	2,326	242	6	1	2	4
0,162	2,142	270	6	1	3	4
1,315	3,229	264	6	1	4	4
-1,43	-0,021	284	6	1	5	4
-0,57	2,822	268	6	1	6	4
-1,668	-0,66	296	6	1	7	4
2,494	4,109	238	6	1	9	4
1,009	4,103	242	6	1	10	4
1,952	3,405	192	6	1	13	4

-0,029	1,208	256	6	1	15	4
0,178	3,454	298	6	1	16	4
-0,45	3,118	256	6	1	17	4
0,199	2,168	250	6	1	18	4
-0,183	1,671	216	6	1	19	4
-1,369	-0,847	254	6	1	20	4
0,83	3,79	196	6	1	21	4
-0,42	0,775	264	7	1	2	1
0,645	1,821	266	7	1	3	1
1,425	3,591	260	7	1	4	1
1,839	4,037	292	7	1	5	1
1,696	4,775	264	7	1	6	1
-2,78	-0,479	178	7	1	7	1
1,658	3,839	280	7	1	9	1
3,029	5,965	266	7	1	10	1
3,577	5,424	248	7	1	13	1
-0,066	1,1	242	7	1	15	1
3,563	5,021	298	7	1	16	1
1,273	4,464	248	7	1	17	1
-1,145	1,002	222	7	1	18	1
-3,049	-0,622	248	7	1	19	1
-1,468	0,1	152	7	1	20	1
1,609	4,168	194	7	1	21	1
0,33	2,147	274	7	1	2	2
0,619	1,526	276	7	1	3	2
1,365	3,336	274	7	1	4	2
-1,695	0,77	298	7	1	5	2
-1,017	1,345	256	7	1	6	2
-2,293	-0,322	276	7	1	7	2
1,762	4,311	270	7	1	9	2
1,069	3,611	264	7	1	10	2
2,49	3,997	190	7	1	13	2
0,822	2,339	154	7	1	15	2
4,267	6,23	274	7	1	16	2
1,324	5,436	260	7	1	17	2
-2,368	-0,537	186	7	1	18	2
5,64	8,678	274	7	1	19	2
0,289	1,894	226	7	1	20	2
-0,631	1,462	202	7	1	21	2
0,129	1,422	290	7	1	2	3
0,552	1,278	250	7	1	3	3
0,916	2,991	272	7	1	4	3
-0,716	2,242	150	7	1	5	3
0,199	2,215	284	7	1	6	3
-3,577	-2,025	158	7	1	7	3
0,116	1,787	258	7	1	9	3
1,028	3,588	266	7	1	10	3
-0,155	2,452	214	7	1	13	3

1,188	1,844	158	7	1	15	3
2,38	3,682	300	7	1	16	3
0,953	4,98	252	7	1	17	3
-0,534	1,973	172	7	1	18	3
-0,763	1,03	264	7	1	19	3
-1,881	-1,004	220	7	1	20	3
0,071	1,9	204	7	1	21	3
-0,764	0,569	242	7	1	2	4
0,68	2,164	272	7	1	3	4
0,336	1,433	264	7	1	4	4
-1,034	-0,016	286	7	1	5	4
0,2	3,168	266	7	1	6	4
-2,988	-2,134	198	7	1	7	4
2,214	3,853	256	7	1	9	4
0,29	2,63	244	7	1	10	4
1,069	2,369	224	7	1	13	4
-0,52	0,418	252	7	1	15	4
1,12	1,676	226	7	1	16	4
-0,469	2,212	256	7	1	17	4
-0,914	1,556	246	7	1	18	4
0,257	1,559	214	7	1	19	4
-2,153	-1,865	222	7	1	20	4
-0,172	2,548	196	7	1	21	4
-1,037	0,613	266	8	1	2	1
0,578	2,242	298	8	1	3	1
0,849	2,728	262	8	1	4	1
1,897	4,919	292	8	1	5	1
0,864	4,071	258	8	1	6	1
-4,216	-1,683	156	8	1	7	1
0,102	3,328	280	8	1	9	1
0,136	2,387	268	8	1	10	1
3,458	5,731	248	8	1	13	1
-1,703	-0,282	236	8	1	15	1
3,002	4,144	164	8	1	16	1
1,367	3,887	262	8	1	17	1
-1,687	1,838	220	8	1	18	1
-2,485	-0,881	248	8	1	19	1
-2,563	1,082	150	8	1	20	1
1,457	3,579	192	8	1	21	1
0,057	2,128	274	8	1	2	2
0,185	1,196	280	8	1	3	2
1,072	2,721	274	8	1	4	2
-2,193	1,599	162	8	1	5	2
-1,16	1,636	252	8	1	6	2
-3,445	-1,266	298	8	1	7	2
0,907	3,928	272	8	1	9	2
-1,833	0,177	264	8	1	10	2
2,034	3,203	270	8	1	13	2

-0,902	1,071	154	8	1	15	2
3,519	4,866	274	8	1	16	2
0,767	4,365	258	8	1	17	2
-2,096	-0,192	244	8	1	18	2
4,779	6,598	274	8	1	19	2
-0,646	0,843	226	8	1	20	2
-1,724	0,642	150	8	1	21	2
-0,198	1,964	292	8	1	2	3
0,108	1,116	160	8	1	3	3
0,183	1,989	284	8	1	4	3
0,659	5,178	150	8	1	5	3
0,175	2,141	284	8	1	6	3
-4,493	-2,155	156	8	1	7	3
-1,425	-0,167	278	8	1	9	3
-1,119	0,588	268	8	1	10	3
-0,598	2,194	228	8	1	13	3
-0,63	0,515	154	8	1	15	3
3,188	4,026	272	8	1	16	3
1,695	5,031	252	8	1	17	3
0,187	2,726	170	8	1	18	3
-1,35	-0,367	160	8	1	19	3
-3,22	-2,551	218	8	1	20	3
-0,342	1,719	152	8	1	21	3
-1,577	-0,559	274	8	1	2	4
0,312	1,772	286	8	1	3	4
-0,429	0,446	280	8	1	4	4
-0,579	2,662	150	8	1	5	4
0,198	2,997	266	8	1	6	4
-4,257	-3,758	198	8	1	7	4
1,198	3,417	288	8	1	9	4
-1,352	0,052	252	8	1	10	4
0,361	2,607	254	8	1	13	4
-2,346	-1,506	158	8	1	15	4
1,321	3,137	156	8	1	16	4
-0,629	1,486	274	8	1	17	4
-2,099	1,6	242	8	1	18	4
0,084	0,8	258	8	1	19	4
-3,592	-3,146	220	8	1	20	4
-2,091	0,765	160	8	1	21	4
-0,765	2,391	238	9	1	2	1
-2,421	-1,027	196	9	1	3	1
1,214	4,236	258	9	1	4	1
-0,487	1,888	294	9	1	5	1
-2,581	2,243	270	9	1	6	1
0,697	3,172	270	9	1	7	1
2	5,625	208	9	1	9	1
0,505	6,188	274	9	1	10	1
1,402	2,818	184	9	1	13	1

1,678	3,613	258	9	1	15	1
0,624	5,459	300	9	1	16	1
-4,095	2,292	262	9	1	17	1
0,744	6,672	288	9	1	18	1
-2,212	2,785	282	9	1	19	1
-0,314	4,476	286	9	1	20	1
-3,293	-1,146	256	9	1	21	1
0,908	6,284	244	9	1	2	2
-2,088	-0,482	294	9	1	3	2
1,738	5,216	260	9	1	4	2
-1,191	1,518	294	9	1	5	2
-3,691	-1,694	178	9	1	6	2
0,065	2,948	276	9	1	7	2
0,084	3,06	248	9	1	9	2
2,173	8,228	264	9	1	10	2
0,81	3,42	192	9	1	13	2
0,918	3,163	284	9	1	15	2
0,279	3,564	274	9	1	16	2
-2,141	3,66	262	9	1	17	2
-5,074	-1,121	282	9	1	18	2
0,871	4,583	274	9	1	19	2
1,312	4,143	270	9	1	20	2
-3,739	-1,821	266	9	1	21	2
-1,996	1,275	238	9	1	2	3
-3,198	-1,412	286	9	1	3	3
2,572	6,617	254	9	1	4	3
-4,14	-1,786	204	9	1	5	3
-5,102	-1,897	282	9	1	6	3
0,687	5,919	288	9	1	7	3
-1,724	1,391	258	9	1	9	3
0,134	7,364	268	9	1	10	3
1,951	4,406	282	9	1	13	3
-0,497	1,606	272	9	1	15	3
-1,575	3,302	298	9	1	16	3
-2,341	3,236	268	9	1	17	3
-5,903	-1,701	286	9	1	18	3
-1,457	2,903	280	9	1	19	3
-0,51	1,503	286	9	1	20	3
-2,564	-0,202	258	9	1	21	3
-1,412	1,947	244	9	1	2	4
-4,158	-2,121	270	9	1	3	4
0,098	3,034	250	9	1	4	4
-3,637	-2,412	254	9	1	5	4
-4,255	0,361	300	9	1	6	4
-0,452	3,134	282	9	1	7	4
0,938	2,932	218	9	1	9	4
-0,78	4,361	260	9	1	10	4
1,237	3,937	208	9	1	13	4

-1,313	0,991	262	9	1	15	4
-0,456	4,3	296	9	1	16	4
-3,179	1,795	256	9	1	17	4
-2,182	1,068	266	9	1	18	4
-0,21	2,045	278	9	1	19	4
-1,758	-0,071	270	9	1	20	4
-3,107	-0,853	190	9	1	21	4
-0,841	0,95	248	10	1	2	1
0,984	1,864	196	10	1	3	1
1,691	3,878	252	10	1	4	1
-0,392	0,451	222	10	1	5	1
0,369	4,478	266	10	1	6	1
-1,669	1,486	300	10	1	7	1
1,433	3,66	214	10	1	9	1
-0,378	3,538	264	10	1	10	1
1,991	3,482	212	10	1	13	1
-0,136	1,033	212	10	1	15	1
1,182	2,106	230	10	1	16	1
-3,008	1,758	262	10	1	17	1
2,128	6,442	288	10	1	18	1
-2,608	0,079	282	10	1	19	1
0,456	3,678	282	10	1	20	1
0,632	2,383	276	10	1	21	1
0,529	4,155	246	10	1	2	2
0,352	1,191	186	10	1	3	2
2,433	4,863	260	10	1	4	2
-2,016	-0,572	268	10	1	5	2
-2,517	-1,433	274	10	1	6	2
-0,99	1,153	280	10	1	7	2
2,628	5,687	236	10	1	9	2
0,481	4,943	262	10	1	10	2
0,834	2,133	208	10	1	13	2
0,391	2,501	288	10	1	15	2
0,89	4,226	296	10	1	16	2
-1,378	2,671	262	10	1	17	2
-2,398	1,139	282	10	1	18	2
2,86	5,416	272	10	1	19	2
1,846	3,188	266	10	1	20	2
-0,072	2,266	198	10	1	21	2
-1,046	1,816	240	10	1	2	3
0,642	1,723	272	10	1	3	3
2,085	5,159	240	10	1	4	3
-1,333	0,175	204	10	1	5	3
-2,552	-0,314	284	10	1	6	3
1,196	5,714	294	10	1	7	3
-0,328	2,016	248	10	1	9	3
-0,853	2,809	270	10	1	10	3
1,247	3,24	264	10	1	13	3

-0,6	0,414	262	10	1	15	3
0,631	2,945	282	10	1	16	3
-1,241	2,623	270	10	1	17	3
-2,119	0,638	288	10	1	18	3
-2,066	0,548	280	10	1	19	3
0,271	1,09	300	10	1	20	3
0,614	2,568	192	10	1	21	3
-1,54	0,837	242	10	1	2	4
-0,595	0,528	270	10	1	3	4
0,582	2,62	250	10	1	4	4
-3,477	-2,454	220	10	1	5	4
-1,939	1,246	280	10	1	6	4
-0,835	1,318	288	10	1	7	4
2,61	4,248	238	10	1	9	4
-1,279	2,569	258	10	1	10	4
0,276	2,437	220	10	1	13	4
-1,903	-0,163	260	10	1	15	4
-0,155	2,336	286	10	1	16	4
-2,201	1,597	256	10	1	17	4
-0,67	1,855	262	10	1	18	4
-0,43	1,356	278	10	1	19	4
-1,399	-0,628	234	10	1	20	4
-0,674	1,598	174	10	1	21	4
-1,801	-0,302	260	11	1	2	1
0,998	1,793	198	11	1	3	1
0,803	2,673	260	11	1	4	1
0,196	1,165	292	11	1	5	1
-0,102	3,361	266	11	1	6	1
-2,988	0,389	300	11	1	7	1
0,407	3,194	262	11	1	9	1
-2,612	0,891	272	11	1	10	1
1,728	3,708	246	11	1	13	1
-1,867	-0,868	278	11	1	15	1
0,15	1,485	230	11	1	16	1
-2,167	1,733	264	11	1	17	1
1,907	4,07	290	11	1	18	1
-3,045	-0,637	282	11	1	19	1
0,107	2,04	282	11	1	20	1
0,4	2,927	282	11	1	21	1
0,022	2,927	246	11	1	2	2
0,807	1,586	184	11	1	3	2
1,937	3,989	260	11	1	4	2
-2,142	-0,979	298	11	1	5	2
-2,49	-1,321	274	11	1	6	2
-0,887	1,443	280	11	1	7	2
2,116	4,843	270	11	1	9	2
-2,112	1,54	264	11	1	10	2
0,512	2,353	272	11	1	13	2

-0,497	1,562	288	11	1	15	2
-0,043	2,704	296	11	1	16	2
-0,994	2,53	262	11	1	17	2
-1,736	0,623	284	11	1	18	2
3,363	5,985	272	11	1	19	2
1,345	2,888	216	11	1	20	2
-0,278	2,624	156	11	1	21	2
-0,831	1,218	240	11	1	2	3
0,731	2,043	162	11	1	3	3
1,025	3,514	252	11	1	4	3
-0,982	0,338	198	11	1	5	3
-1,767	0,066	284	11	1	6	3
0,06	4,482	296	11	1	7	3
-0,674	1,23	246	11	1	9	3
-2,801	0,563	272	11	1	10	3
0,043	2,132	264	11	1	13	3
-0,792	0,302	278	11	1	15	3
-0,084	1,56	212	11	1	16	3
-0,215	3,173	272	11	1	17	3
-0,773	1,444	290	11	1	18	3
-3,313	-1,639	280	11	1	19	3
-0,412	1,027	166	11	1	20	3
0,2	2,446	156	11	1	21	3
-2,126	-0,062	244	11	1	2	4
0,149	0,732	272	11	1	3	4
-0,384	1,01	242	11	1	4	4
-3,188	-2,783	198	11	1	5	4
-1,914	0,856	280	11	1	6	4
-1,365	0,513	294	11	1	7	4
1,726	3,902	256	11	1	9	4
-3,261	0,38	264	11	1	10	4
-0,272	2,429	252	11	1	13	4
-2,2	-0,525	260	11	1	15	4
-1,225	0,39	288	11	1	16	4
-1,707	1,582	258	11	1	17	4
-1,356	1,02	252	11	1	18	4
-1,196	-0,19	276	11	1	19	4
-2,187	-0,212	150	11	1	20	4
-1,451	1,288	172	11	1	21	4
-2,533	-1,045	170	12	1	2	1
0,705	2,23	298	12	1	3	1
0,386	2,053	256	12	1	4	1
0,951	2,476	292	12	1	5	1
-1,59	1,472	264	12	1	6	1
-3,381	0,635	300	12	1	7	1
-0,372	3,035	278	12	1	9	1
-5,002	-0,612	296	12	1	10	1
1,145	3,953	246	12	1	13	1

-3,843	-2,715	272	12	1	15	1
-0,846	0,757	230	12	1	16	1
-1,136	2,557	282	12	1	17	1
1,149	2,945	250	12	1	18	1
-3,515	0,397	162	12	1	19	1
-1,227	0,683	152	12	1	20	1
-0,159	3,529	290	12	1	21	1
-0,662	1,705	272	12	1	2	2
0,544	1,335	168	12	1	3	2
1,578	3,192	260	12	1	4	2
-2,309	-0,171	298	12	1	5	2
-3,051	-1,687	274	12	1	6	2
-1,696	1,626	294	12	1	7	2
1,347	5,238	272	12	1	9	2
-4,838	-0,524	282	12	1	10	2
1,023	3,518	272	12	1	13	2
-2,354	-0,766	292	12	1	15	2
-1,218	0,858	254	12	1	16	2
-1,207	2,218	260	12	1	17	2
-1,418	-0,222	246	12	1	18	2
2,757	6,54	154	12	1	19	2
-0,634	1,109	208	12	1	20	2
-1,322	2,987	154	12	1	21	2
-0,687	0,424	240	12	1	2	3
0,403	2,391	160	12	1	3	3
0,451	2,703	250	12	1	4	3
-0,15	1,522	152	12	1	5	3
-2,174	-0,522	284	12	1	6	3
-1,825	2,517	296	12	1	7	3
-1,213	0,853	296	12	1	9	3
-5,013	-0,843	278	12	1	10	3
-0,725	2,002	246	12	1	13	3
-2,011	-0,88	278	12	1	15	3
-0,64	0,886	214	12	1	16	3
0,317	3,385	272	12	1	17	3
-0,801	0,495	292	12	1	18	3
-3,854	-0,639	162	12	1	19	3
-2,304	0,777	164	12	1	20	3
-0,815	2,753	154	12	1	21	3
-2,424	-0,666	244	12	1	2	4
0,105	1,372	152	12	1	3	4
-0,936	0,057	242	12	1	4	4
-2,593	-1,959	168	12	1	5	4
-2,617	0,225	268	12	1	6	4
-2,486	-0,245	296	12	1	7	4
0,929	3,876	288	12	1	9	4
-5,469	-1,056	274	12	1	10	4
-0,928	2,727	252	12	1	13	4

-2,874	-1,122	258	12	1	15	4
-2,233	-1,564	290	12	1	16	4
-1,552	1,427	258	12	1	17	4
-1,84	1,024	244	12	1	18	4
-2,019	-1,823	260	12	1	19	4
-3,633	-3,104	222	12	1	20	4
-3,076	1,116	158	12	1	21	4
-2,107	0,609	320	1	2	2	1
-2,351	-0,089	430	1	2	3	1
-0,092	2,291	326	1	2	4	1
1,43	3,774	362	1	2	5	1
-2,507	0,632	368	1	2	6	1
-0,314	2,843	414	1	2	7	1
-1,306	1,599	440	1	2	9	1
0,575	2,593	478	1	2	10	1
-0,493	4,403	452	1	2	13	1
-2,452	0,172	360	1	2	15	1
-3,116	5,191	300	1	2	16	1
-5,941	-1,163	314	1	2	17	1
-1,16	2,015	384	1	2	18	1
-8,599	-1,87	318	1	2	19	1
-0,797	0,959	404	1	2	20	1
-5,773	-1,526	440	1	2	21	1
-1,692	0,785	356	1	2	2	2
-2,992	-1,099	476	1	2	3	2
2,189	3,699	428	1	2	4	2
-1,484	1,277	302	1	2	5	2
-6,749	-3,446	376	1	2	6	2
5,127	8,716	458	1	2	7	2
-1,839	1,683	464	1	2	9	2
-0,07	1,963	480	1	2	10	2
-2,458	0,112	306	1	2	13	2
-1,422	1,689	302	1	2	15	2
-2,68	2,717	306	1	2	16	2
-5,722	-1,104	312	1	2	17	2
-7,413	-2,942	332	1	2	18	2
0,278	5,07	306	1	2	19	2
-0,118	2,672	452	1	2	20	2
-4,286	-0,161	366	1	2	21	2
-3,604	-0,801	340	1	2	2	3
-3,854	-0,693	300	1	2	3	3
1,053	3,615	304	1	2	4	3
-5,346	-1,594	304	1	2	5	3
-7,257	-5,462	362	1	2	6	3
-0,391	1,033	412	1	2	7	3
-5,42	-2,468	340	1	2	9	3
-2,217	-0,6	484	1	2	10	3
-3,458	-0,922	482	1	2	13	3

-2,399	0,28	322	1	2	15	3
-5,761	-0,062	312	1	2	16	3
-6,083	-0,942	304	1	2	17	3
-5,03	1,562	420	1	2	18	3
-7,411	-4,144	314	1	2	19	3
-3,918	-1,379	302	1	2	20	3
-7,383	-4,218	390	1	2	21	3
-3,41	-1,219	426	1	2	2	4
-4,767	-2,588	300	1	2	3	4
-1,713	-0,394	466	1	2	4	4
-2,86	0,183	306	1	2	5	4
-3,999	-0,725	360	1	2	6	4
0,851	4,347	444	1	2	7	4
-3,793	-1,208	352	1	2	9	4
-2,648	-1,853	472	1	2	10	4
-1,282	2,315	306	1	2	13	4
-3,863	-3,014	464	1	2	15	4
-1,754	0,155	376	1	2	16	4
-5,261	-4,26	406	1	2	17	4
-1,785	0,898	414	1	2	18	4
-5,333	-2,542	328	1	2	19	4
-2,354	-0,319	470	1	2	20	4
-7,358	-5,16	384	1	2	21	4
-2,466	-0,661	436	2	2	2	1
0,524	2,277	430	2	2	3	1
0,857	2,394	326	2	2	4	1
3,067	4,951	358	2	2	5	1
-2,201	0,155	366	2	2	6	1
-0,472	2,937	406	2	2	7	1
-0,098	1,823	376	2	2	9	1
-1,442	1,875	480	2	2	10	1
0,014	3,293	464	2	2	13	1
-2,023	-0,498	328	2	2	15	1
-1,112	5,498	300	2	2	16	1
-4,762	-1,162	312	2	2	17	1
-1,839	0,116	436	2	2	18	1
-6,301	-1,43	318	2	2	19	1
-3,022	-1,733	416	2	2	20	1
-0,002	3,335	440	2	2	21	1
0,066	1,591	356	2	2	2	2
-0,625	0,493	448	2	2	3	2
1,63	2,679	456	2	2	4	2
-0,248	1,266	412	2	2	5	2
-6,687	-3,569	308	2	2	6	2
-0,217	1,317	410	2	2	7	2
1,384	3,383	334	2	2	9	2
-1,298	1,251	494	2	2	10	2
-1,459	0,955	492	2	2	13	2

-1,841	0,268	302	2	2	15	2
-1,387	0,02	422	2	2	16	2
-4,34	-0,334	310	2	2	17	2
-5,318	-2,171	300	2	2	18	2
2,136	4,91	340	2	2	19	2
-1,51	0,589	452	2	2	20	2
-0,103	3,639	384	2	2	21	2
-3,412	-0,675	416	2	2	2	3
-0,386	1,27	302	2	2	3	3
0,148	2,638	302	2	2	4	3
-2,447	0,911	300	2	2	5	3
-6,545	-5,242	356	2	2	6	3
-1,381	0,498	498	2	2	7	3
-4,157	-1,237	328	2	2	9	3
-3,795	-0,593	500	2	2	10	3
-4,218	-1,554	310	2	2	13	3
-2,014	0,507	342	2	2	15	3
-2,949	2,88	300	2	2	16	3
-5,074	-0,296	318	2	2	17	3
-4,307	-0,634	390	2	2	18	3
-6,325	-3,052	332	2	2	19	3
-3,578	-2,635	480	2	2	20	3
-1,57	1,983	422	2	2	21	3
-3,958	-2,016	426	2	2	2	4
-1,598	-0,287	318	2	2	3	4
-1,18	0,223	470	2	2	4	4
-2,554	-0,573	312	2	2	5	4
-3,048	-0,224	358	2	2	6	4
-2,215	-0,606	414	2	2	7	4
0,929	2,627	354	2	2	9	4
-4,109	-1,88	488	2	2	10	4
-1,015	2,017	310	2	2	13	4
-4,1	-3,219	300	2	2	15	4
0,547	3,262	424	2	2	16	4
-4,57	-3,515	418	2	2	17	4
-2,578	-1,184	352	2	2	18	4
-3,971	-1,494	328	2	2	19	4
-3,381	-1,551	470	2	2	20	4
-4,163	-0,918	408	2	2	21	4
-1,796	0,191	436	3	2	2	1
1,482	2,928	416	3	2	3	1
1,042	2,167	476	3	2	4	1
3,822	5,206	358	3	2	5	1
-1,748	0,001	350	3	2	6	1
-0,426	2,883	406	3	2	7	1
0,586	2,111	392	3	2	9	1
-1,416	3,098	482	3	2	10	1
0,392	2,719	452	3	2	13	1

-40,926	-34,02	316	3	2	15	1
0,517	5,457	300	3	2	16	1
-4,299	-0,699	310	3	2	17	1
-1,673	0,584	436	3	2	18	1
-5,484	-1,522	332	3	2	19	1
-3,891	-0,808	314	3	2	20	1
2,181	5,386	376	3	2	21	1
0,341	1,321	366	3	2	2	2
0,172	1,091	448	3	2	3	2
1,706	2,883	456	3	2	4	2
-0,6	1,396	498	3	2	5	2
-5,442	-2,429	308	3	2	6	2
-0,194	1,653	400	3	2	7	2
2,349	4,67	320	3	2	9	2
-0,427	3,486	494	3	2	10	2
0,243	3,117	492	3	2	13	2
4,222	5,848	456	3	2	15	2
-1,767	-0,145	422	3	2	16	2
-3,701	-0,081	308	3	2	17	2
-5,1	-1,864	312	3	2	18	2
1,985	4,059	340	3	2	19	2
-1,913	0,813	300	3	2	20	2
1,044	5,84	390	3	2	21	2
-3,177	-0,892	416	3	2	2	3
0,747	1,84	316	3	2	3	3
0,005	2,712	302	3	2	4	3
-2,415	0,75	302	3	2	5	3
-5,654	-4,958	360	3	2	6	3
-1,866	0,573	500	3	2	7	3
-3,157	-0,459	328	3	2	9	3
-3,098	0,542	500	3	2	10	3
-3,996	-1,588	310	3	2	13	3
2,618	4,541	326	3	2	15	3
-2,276	2,725	302	3	2	16	3
-4,101	0,781	318	3	2	17	3
-3,108	-0,295	396	3	2	18	3
-5,861	-2,91	334	3	2	19	3
-5,258	-2,667	314	3	2	20	3
0,106	3,898	408	3	2	21	3
-3,822	-2,226	426	3	2	2	4
-0,097	1,379	302	3	2	3	4
-0,608	0,797	472	3	2	4	4
-2,473	-0,582	312	3	2	5	4
-2,193	-0,333	360	3	2	6	4
-2,063	-0,365	498	3	2	7	4
2,373	4,029	306	3	2	9	4
-3,919	-0,522	490	3	2	10	4
-0,605	1,985	310	3	2	13	4

-24,014	-17,596	300	3	2	15	4
1,043	4,027	424	3	2	16	4
-3,975	-3,09	418	3	2	17	4
-2,454	-0,866	412	3	2	18	4
-3,412	-1,335	342	3	2	19	4
-3,968	-2,46	484	3	2	20	4
-2,739	1,101	406	3	2	21	4
-2,946	-0,988	472	4	2	2	1
1,001	3,357	310	4	2	3	1
-0,432	1,274	478	4	2	4	1
4,868	6,805	474	4	2	5	1
-2,557	-0,042	492	4	2	6	1
-5,458	-2,074	300	4	2	7	1
0,033	0,921	394	4	2	9	1
-2,473	3,59	488	4	2	10	1
-0,015	3,098	500	4	2	13	1
-1,943	-0,231	322	4	2	15	1
0,684	4,154	432	4	2	16	1
-1,965	1,328	346	4	2	17	1
-0,249	2,067	446	4	2	18	1
-3,815	-1,255	498	4	2	19	1
-6,897	-4,465	300	4	2	20	1
1,851	3,1	448	4	2	21	1
-2,423	-0,735	480	4	2	2	2
-0,174	1,353	308	4	2	3	2
0,18	1,887	478	4	2	4	2
-0,974	2,585	498	4	2	5	2
-5,256	-1,667	306	4	2	6	2
-3,432	-1,252	386	4	2	7	2
2,783	4,693	304	4	2	9	2
-1,618	3,562	490	4	2	10	2
0,458	5,788	494	4	2	13	2
-3,231	-1,343	464	4	2	15	2
-0,984	1,437	420	4	2	16	2
-3,014	-1,468	312	4	2	17	2
-1,943	0,327	312	4	2	18	2
4,856	8,222	476	4	2	19	2
-5,973	-4,053	336	4	2	20	2
-0,01	2,483	394	4	2	21	2
-2,733	-0,15	342	4	2	2	3
0,477	2,117	314	4	2	3	3
-1,237	1,288	300	4	2	4	3
-0,849	2,032	300	4	2	5	3
-4,427	-3,058	496	4	2	6	3
-5,113	-0,855	308	4	2	7	3
-2,015	-0,21	310	4	2	9	3
-2,946	1,792	494	4	2	10	3
-4,002	1,799	496	4	2	13	3

-2,935	-0,327	324	4	2	15	3
1,369	5,115	500	4	2	16	3
-2,452	1,279	318	4	2	17	3
-1,269	0,478	398	4	2	18	3
-4,498	-0,818	498	4	2	19	3
-7,631	-4,906	498	4	2	20	3
0,21	2,505	350	4	2	21	3
-4,619	-3,263	308	4	2	2	4
0,561	2,044	304	4	2	3	4
-1,182	0,696	496	4	2	4	4
-0,477	1,026	312	4	2	5	4
-3,697	-0,167	302	4	2	6	4
-5,422	-1,034	498	4	2	7	4
2,898	3,875	308	4	2	9	4
-4,08	0,425	488	4	2	10	4
-2,752	2,812	498	4	2	13	4
-3,137	-1,948	456	4	2	15	4
1,013	3,673	424	4	2	16	4
-3,72	-1,783	346	4	2	17	4
-1,507	0,825	412	4	2	18	4
-2,6	-0,79	344	4	2	19	4
-8,372	-6,047	304	4	2	20	4
-2,762	-0,543	360	4	2	21	4
-1,414	1,69	320	5	2	2	1
-2,25	-0,205	430	5	2	3	1
0,808	2,739	326	5	2	4	1
1,163	3,94	312	5	2	5	1
-1,169	1,022	370	5	2	6	1
0,997	3,489	316	5	2	7	1
0,622	3,718	442	5	2	9	1
3,704	5,368	442	5	2	10	1
1,549	5,912	466	5	2	13	1
90,233	98,321	494	5	2	15	1
0,37	6,324	300	5	2	16	1
-3,823	0,241	314	5	2	17	1
-3,033	-1,084	420	5	2	18	1
-9,558	-3,432	300	5	2	19	1
-0,325	1,437	404	5	2	20	1
-5,052	-0,43	458	5	2	21	1
-2,496	-0,418	324	5	2	2	2
-3,274	-1,826	308	5	2	3	2
3,11	4,646	444	5	2	4	2
-2,013	0,962	300	5	2	5	2
-4,341	-1,291	378	5	2	6	2
0,783	3,844	442	5	2	7	2
-1,105	2,37	466	5	2	9	2
0,729	4,072	310	5	2	10	2
-2,717	0,704	308	5	2	13	2

62,157	70,085	468	5	2	15	2
0,213	4,51	306	5	2	16	2
-2,825	1,786	312	5	2	17	2
-7,171	-4,189	414	5	2	18	2
-0,524	4,381	306	5	2	19	2
-0,577	1,787	452	5	2	20	2
-5,309	-2,211	464	5	2	21	2
2,252	4,356	416	5	2	2	3
-3,456	-0,16	300	5	2	3	3
1,758	4,233	304	5	2	4	3
-4,65	-1,6	302	5	2	5	3
-6,461	-4,548	364	5	2	6	3
-0,553	1,907	426	5	2	7	3
-2,739	0,038	444	5	2	9	3
0,255	1,764	386	5	2	10	3
-2,974	0,434	466	5	2	13	3
-72,286	-56,204	324	5	2	15	3
-4,183	1,659	312	5	2	16	3
-3,686	-0,342	320	5	2	17	3
-6,822	-3,226	420	5	2	18	3
-6,881	-3,513	314	5	2	19	3
-2,526	-0,707	454	5	2	20	3
-7,862	-5,564	456	5	2	21	3
-0,739	0,947	428	5	2	2	4
-3,976	-2,239	416	5	2	3	4
-0,525	0,648	466	5	2	4	4
-1,519	0,503	304	5	2	5	4
-0,799	1,052	300	5	2	6	4
-1,111	1,879	312	5	2	7	4
-1,293	1,034	452	5	2	9	4
0,016	1,218	438	5	2	10	4
0,545	3,385	308	5	2	13	4
-30,508	-18,519	330	5	2	15	4
-2,421	-1,131	426	5	2	16	4
-3,658	0,446	304	5	2	17	4
-3,606	-1,117	314	5	2	18	4
-5,232	-2,849	326	5	2	19	4
-1,734	-0,499	422	5	2	20	4
-4,707	-1,543	458	5	2	21	4
-0,439	1,392	320	6	2	2	1
0,28	1,856	430	6	2	3	1
1,338	2,768	326	6	2	4	1
2,805	4,117	360	6	2	5	1
1,236	3,104	456	6	2	6	1
-0,446	0,83	408	6	2	7	1
1,271	3,188	458	6	2	9	1
2,727	4,816	478	6	2	10	1
2,006	5,203	462	6	2	13	1

-1,798	0,014	326	6	2	15	1
3,146	5,936	300	6	2	16	1
-1,399	0,579	448	6	2	17	1
-2,673	0,952	436	6	2	18	1
-7,476	-2,698	316	6	2	19	1
-2,041	-1,226	418	6	2	20	1
-0,567	2,472	458	6	2	21	1
-0,614	0,617	440	6	2	2	2
-0,993	0,217	310	6	2	3	2
1,677	2,527	456	6	2	4	2
0,152	1,101	416	6	2	5	2
-2,546	-1,745	376	6	2	6	2
-0,671	1,001	308	6	2	7	2
1,521	4,021	466	6	2	9	2
0,478	3,331	494	6	2	10	2
0,346	2,549	458	6	2	13	2
-1,78	0,465	304	6	2	15	2
2,049	2,991	424	6	2	16	2
-0,892	1,623	312	6	2	17	2
-3,953	-1,278	416	6	2	18	2
4,785	7,601	304	6	2	19	2
-1,364	0,602	452	6	2	20	2
-1,125	1,393	452	6	2	21	2
-0,837	0,628	338	6	2	2	3
-0,338	1,464	302	6	2	3	3
1,04	3,57	304	6	2	4	3
-2,942	-0,537	322	6	2	5	3
-2,756	-1,929	364	6	2	6	3
-2,397	-0,386	310	6	2	7	3
-1,671	-0,454	326	6	2	9	3
-0,08	2,414	486	6	2	10	3
-2,397	0,507	482	6	2	13	3
-0,863	1,389	342	6	2	15	3
-1,62	3,353	300	6	2	16	3
-2,464	-0,016	320	6	2	17	3
-3,704	-2,392	490	6	2	18	3
-3,961	-1,272	312	6	2	19	3
-2,96	-1,169	302	6	2	20	3
-2,693	-0,692	424	6	2	21	3
-2,257	-1	308	6	2	2	4
-0,959	0,873	302	6	2	3	4
-0,037	1,058	314	6	2	4	4
-0,649	0,778	308	6	2	5	4
-0,242	1,322	360	6	2	6	4
-3,111	-0,622	310	6	2	7	4
0,844	1,948	304	6	2	9	4
-1,476	0,165	488	6	2	10	4
0,145	2,297	468	6	2	13	4

-3,072	-0,908	314	6	2	15	4
-0,168	1,685	424	6	2	16	4
-2,068	-1,597	406	6	2	17	4
-2,728	-1,114	446	6	2	18	4
-3,275	-1,391	326	6	2	19	4
-2,662	-1,673	488	6	2	20	4
-1,976	0,322	454	6	2	21	4
-0,132	0,802	318	7	2	2	1
0,719	1,885	432	7	2	3	1
0,67	1,923	326	7	2	4	1
3,622	5,625	494	7	2	5	1
1,329	2,918	456	7	2	6	1
-2,484	-1,351	408	7	2	7	1
1,612	5,027	302	7	2	9	1
2,653	6,168	490	7	2	10	1
1,576	4,341	464	7	2	13	1
-1,824	-0,236	410	7	2	15	1
3,532	6,467	418	7	2	16	1
-0,918	0,549	398	7	2	17	1
-2,982	0,667	436	7	2	18	1
-6,301	-2,092	332	7	2	19	1
-2,628	-1,089	488	7	2	20	1
1,066	3,035	374	7	2	21	1
-0,202	1,007	354	7	2	2	2
-0,337	0,717	484	7	2	3	2
0,845	1,764	492	7	2	4	2
-0,24	2,608	498	7	2	5	2
-3,173	-0,352	306	7	2	6	2
-2,362	-0,934	306	7	2	7	2
2,554	4,673	468	7	2	9	2
1,404	5,747	494	7	2	10	2
1,224	3,893	492	7	2	13	2
-2,119	-0,44	304	7	2	15	2
2,725	4,045	422	7	2	16	2
-0,869	0,828	312	7	2	17	2
-4,042	-1,119	418	7	2	18	2
6,326	8,247	472	7	2	19	2
-1,157	0,092	452	7	2	20	2
0,215	2,409	392	7	2	21	2
-1,127	0,722	340	7	2	2	3
0,326	1,584	302	7	2	3	3
-0,125	2,213	302	7	2	4	3
-2,372	0,471	300	7	2	5	3
-2,267	-1,906	362	7	2	6	3
-4,749	-3,231	308	7	2	7	3
-1,489	-0,309	312	7	2	9	3
0,782	4,564	500	7	2	10	3
-2,549	1,443	494	7	2	13	3

-0,734	1,725	342	7	2	15	3
0,337	3,682	300	7	2	16	3
-1,824	0,671	318	7	2	17	3
-3,565	-2,217	318	7	2	18	3
-3,49	-1,573	330	7	2	19	3
-3,651	-1,768	302	7	2	20	3
-1,077	0,995	424	7	2	21	3
-2,626	-0,933	308	7	2	2	4
0,046	1,552	302	7	2	3	4
-0,535	0,661	494	7	2	4	4
-0,388	0,863	320	7	2	5	4
-0,456	0,482	362	7	2	6	4
-4,338	-2,891	310	7	2	7	4
2,192	3,229	304	7	2	9	4
-1,154	1,696	488	7	2	10	4
-0,518	2,474	476	7	2	13	4
-3,147	-1,141	302	7	2	15	4
1,226	3,57	424	7	2	16	4
-1,905	-1,482	406	7	2	17	4
-3,778	-2,523	318	7	2	18	4
-2,275	-0,767	344	7	2	19	4
-3,789	-2,506	300	7	2	20	4
-1,853	-0,418	372	7	2	21	4
-0,11	1,177	468	8	2	2	1
1,075	2,202	308	8	2	3	1
0,03	1,495	490	8	2	4	1
3,847	7,417	494	8	2	5	1
-0,347	1,673	318	8	2	6	1
-4,206	-1,872	496	8	2	7	1
0,505	4,416	302	8	2	9	1
2,247	7,594	492	8	2	10	1
1,612	5,081	482	8	2	13	1
-3,33	-1,612	410	8	2	15	1
2,88	6,651	428	8	2	16	1
-0,656	0,776	398	8	2	17	1
-2,707	0,906	436	8	2	18	1
-4,79	-1,781	332	8	2	19	1
-4,79	-2,813	310	8	2	20	1
2,512	4,17	380	8	2	21	1
0,054	1,454	486	8	2	2	2
-0,491	0,8	310	8	2	3	2
0,501	1,8	492	8	2	4	2
-0,401	4,482	498	8	2	5	2
-3,534	-0,077	312	8	2	6	2
-3,475	-2,258	320	8	2	7	2
2,74	4,562	470	8	2	9	2
2,025	7,679	494	8	2	10	2
1,512	6,37	492	8	2	13	2

-4,318	-3,019	456	8	2	15	2
2,578	4,395	422	8	2	16	2
-1,132	0,239	396	8	2	17	2
-3,155	0,049	418	8	2	18	2
5,721	8,516	470	8	2	19	2
-2,38	-1,539	302	8	2	20	2
0,154	2,785	394	8	2	21	2
-1,563	0,684	340	8	2	2	3
0,233	1,462	312	8	2	3	3
-1,077	1,595	300	8	2	4	3
-0,475	3,172	500	8	2	5	3
-2,006	1,717	304	8	2	6	3
-5,865	-4,304	310	8	2	7	3
-2,015	-0,357	312	8	2	9	3
1,172	6,409	500	8	2	10	3
-2,764	3,704	496	8	2	13	3
-2,663	-0,114	342	8	2	15	3
1,812	4,427	426	8	2	16	3
-1,017	1,74	318	8	2	17	3
-2,419	-0,945	318	8	2	18	3
-4,302	-2,504	348	8	2	19	3
-5,769	-3,38	496	8	2	20	3
0,345	2,391	352	8	2	21	3
-2,916	-1,025	308	8	2	2	4
0,138	1,488	302	8	2	3	4
-1,025	0,732	496	8	2	4	4
0,411	1,724	314	8	2	5	4
-1,133	1,856	300	8	2	6	4
-5,329	-3,197	496	8	2	7	4
2,609	3,679	306	8	2	9	4
-0,991	3,26	490	8	2	10	4
-1,323	3,705	494	8	2	13	4
-5,105	-3,062	302	8	2	15	4
1,828	4,675	424	8	2	16	4
-2,146	-1,221	344	8	2	17	4
-3,777	-2,61	424	8	2	18	4
-1,908	-0,469	346	8	2	19	4
-5,95	-4,259	306	8	2	20	4
-2,33	-0,758	370	8	2	21	4
-0,862	0,669	440	9	2	2	1
-4,182	-2,785	428	9	2	3	1
-0,09	1,317	344	9	2	4	1
0,504	1,625	494	9	2	5	1
-1,18	1,808	370	9	2	6	1
2,887	5,049	314	9	2	7	1
0,306	3,406	388	9	2	9	1
-0,911	3,486	308	9	2	10	1
0,593	2,6	412	9	2	13	1

0,625	3,194	358	9	2	15	1
-1,389	5,459	300	9	2	16	1
-4,196	0,483	314	9	2	17	1
0,222	3,732	386	9	2	18	1
-5,1	0,73	300	9	2	19	1
0,992	3,057	302	9	2	20	1
-5,06	-2,69	390	9	2	21	1
0,912	3,2	324	9	2	2	2
-2,938	-0,862	308	9	2	3	2
1,905	3,099	426	9	2	4	2
-0,94	0,237	316	9	2	5	2
-6,063	-3,401	376	9	2	6	2
1,489	4,136	460	9	2	7	2
0,622	2,324	450	9	2	9	2
-1,549	1,042	328	9	2	10	2
-2,6	-0,558	308	9	2	13	2
-0,159	2,366	300	9	2	15	2
0,08	4,03	304	9	2	16	2
-3,537	0,188	312	9	2	17	2
-4,189	-0,914	334	9	2	18	2
0,118	4,753	306	9	2	19	2
0,947	2,189	318	9	2	20	2
-3,761	0,092	366	9	2	21	2
-3,829	-1,652	342	9	2	2	3
-4,4	-1,618	314	9	2	3	3
2,014	4,519	306	9	2	4	3
-4,396	-2,713	340	9	2	5	3
-5,385	-3,111	422	9	2	6	3
1,097	3,073	324	9	2	7	3
-3,706	-0,785	326	9	2	9	3
-2,973	0,78	316	9	2	10	3
-2,105	0,945	466	9	2	13	3
-0,931	1,285	322	9	2	15	3
-1,917	0,797	330	9	2	16	3
-3,415	0,516	302	9	2	17	3
-3,796	0,775	418	9	2	18	3
-2,022	0,812	414	9	2	19	3
-0,886	1,23	302	9	2	20	3
-5,211	-2,239	390	9	2	21	3
-1,54	-0,117	410	9	2	2	4
-5,319	-3,732	302	9	2	3	4
-0,936	0,231	348	9	2	4	4
-3,426	-1,302	306	9	2	5	4
-1,028	1,187	360	9	2	6	4
-0,588	1,526	310	9	2	7	4
-0,097	1,057	354	9	2	9	4
-1,999	-1,268	366	9	2	10	4
-1,296	2,035	308	9	2	13	4

-0,944	-0,216	334	9	2	15	4
0,748	2,562	362	9	2	16	4
-3,206	0,704	304	9	2	17	4
-0,736	1,65	364	9	2	18	4
-2,415	-0,664	328	9	2	19	4
-0,867	0,281	420	9	2	20	4
-5,869	-3,006	374	9	2	21	4
-1,44	-0,178	444	10	2	2	1
-0,403	0,868	314	10	2	3	1
0,393	1,942	476	10	2	4	1
0,799	2,688	494	10	2	5	1
0,614	2,809	366	10	2	6	1
-0,331	3,156	400	10	2	7	1
0,24	2,407	390	10	2	9	1
-2,262	-0,537	480	10	2	10	1
0,528	3,401	412	10	2	13	1
-1,006	1,089	330	10	2	15	1
-0,336	4,387	304	10	2	16	1
-4,119	-0,625	312	10	2	17	1
0,9	2,5	322	10	2	18	1
-3,73	-0,672	332	10	2	19	1
0,427	2,538	316	10	2	20	1
0,717	3,117	376	10	2	21	1
0,503	2,653	322	10	2	2	2
-1,273	-0,581	326	10	2	3	2
1,186	2,02	480	10	2	4	2
-1,834	-0,214	486	10	2	5	2
-4,228	-2,269	322	10	2	6	2
-0,217	0,587	408	10	2	7	2
3,926	5,247	422	10	2	9	2
-2,052	0,119	490	10	2	10	2
-0,659	1,016	304	10	2	13	2
-0,41	2,092	304	10	2	15	2
-0,508	0,891	420	10	2	16	2
-4,186	-0,975	310	10	2	17	2
-1,833	1,477	330	10	2	18	2
3,907	6,345	354	10	2	19	2
0,388	2,358	318	10	2	20	2
1,177	4,79	396	10	2	21	2
-3,11	-1,505	342	10	2	2	3
-0,634	0,522	372	10	2	3	3
0,351	2,619	304	10	2	4	3
-2,579	-0,575	300	10	2	5	3
-3,732	-2,178	318	10	2	6	3
1,023	2,139	396	10	2	7	3
-1,775	0,81	326	10	2	9	3
-3,613	-0,719	314	10	2	10	3
-2,716	-0,935	464	10	2	13	3

-1,766	0,753	324	10	2	15	3
-0,03	3,958	302	10	2	16	3
-3,995	-0,673	320	10	2	17	3
-1,356	1,208	396	10	2	18	3
-3,156	-1,077	348	10	2	19	3
-0,788	1,09	300	10	2	20	3
0,915	3,456	390	10	2	21	3
-1,839	-0,594	344	10	2	2	4
-2,329	-0,625	302	10	2	3	4
-0,797	0,324	312	10	2	4	4
-3,777	-2,233	308	10	2	5	4
-0,534	1,562	302	10	2	6	4
-1,356	-0,151	498	10	2	7	4
2,821	3,967	306	10	2	9	4
-3,185	0,576	308	10	2	10	4
-1,464	0,915	310	10	2	13	4
-2,573	-1,72	322	10	2	15	4
0,015	1,213	426	10	2	16	4
-4,211	-4,036	450	10	2	17	4
0,292	2,095	364	10	2	18	4
-1,664	0,322	342	10	2	19	4
-1,445	-0,548	484	10	2	20	4
-2,302	0,696	410	10	2	21	4
-2,39	-1,236	446	11	2	2	1
0,117	1,652	312	11	2	3	1
-0,334	1,462	304	11	2	4	1
1,721	5,061	494	11	2	5	1
-0,696	0,835	490	11	2	6	1
-2,154	0,887	398	11	2	7	1
0,296	2,053	392	11	2	9	1
-3,454	-0,032	482	11	2	10	1
0,06	2,889	500	11	2	13	1
-2,284	-0,284	330	11	2	15	1
-1,096	2,234	304	11	2	16	1
-3,55	-1,027	344	11	2	17	1
1,001	3,189	324	11	2	18	1
-3,61	-0,83	332	11	2	19	1
-1,061	1,368	316	11	2	20	1
2,018	4,089	374	11	2	21	1
-0,094	1,944	322	11	2	2	2
-0,494	0,96	310	11	2	3	2
0,632	1,806	478	11	2	4	2
-1,989	1,225	498	11	2	5	2
-4,593	-1,639	322	11	2	6	2
-0,036	1,581	384	11	2	7	2
4,047	5,914	320	11	2	9	2
-2,338	1,281	488	11	2	10	2
-0,351	2,627	494	11	2	13	2

-1,21	0,728	304	11	2	15	2
-1,75	-0,522	418	11	2	16	2
-3,727	-1,015	314	11	2	17	2
-1,505	2,264	316	11	2	18	2
5,495	7,807	354	11	2	19	2
-0,217	1,903	318	11	2	20	2
1,833	5,748	394	11	2	21	2
-2,89	-0,836	342	11	2	2	3
-0,027	1,248	372	11	2	3	3
-0,83	1,524	304	11	2	4	3
-2,777	-0,222	306	11	2	5	3
-3,665	-1,753	494	11	2	6	3
-0,083	1,478	500	11	2	7	3
-1,539	1,07	310	11	2	9	3
-4,173	-1,28	316	11	2	10	3
-3,729	-1,866	312	11	2	13	3
-1,258	1,188	326	11	2	15	3
-0,817	2,461	304	11	2	16	3
-3,461	0,074	320	11	2	17	3
-0,184	1,816	392	11	2	18	3
-4,145	-1,317	330	11	2	19	3
-2,246	-0,365	488	11	2	20	3
1,875	3,969	390	11	2	21	3
-2,741	-1,298	344	11	2	2	4
-1,055	0,463	302	11	2	3	4
-1,464	-0,337	312	11	2	4	4
-3,313	-1,466	330	11	2	5	4
-1,596	1,05	314	11	2	6	4
-1,663	0,499	496	11	2	7	4
2,984	4,61	306	11	2	9	4
-4,62	-1,756	490	11	2	10	4
-2,234	0,389	324	11	2	13	4
-2,71	-1,975	334	11	2	15	4
-0,975	0,077	426	11	2	16	4
-3,858	-3,854	372	11	2	17	4
-0,231	1,098	392	11	2	18	4
-2,024	-0,105	342	11	2	19	4
-2,983	-1,813	368	11	2	20	4
-1,539	1,278	408	11	2	21	4
-3,458	-1,801	472	12	2	2	1
0,147	2,216	310	12	2	3	1
-0,773	1,259	304	12	2	4	1
3,202	6,595	494	12	2	5	1
-2,836	-0,221	490	12	2	6	1
-2,86	0,635	300	12	2	7	1
0,755	2,494	306	12	2	9	1
-4,258	0,947	490	12	2	10	1
-0,809	3,845	500	12	2	13	1

-4,483	-2,157	334	12	2	15	1
-1,661	0,102	304	12	2	16	1
-2,894	0,482	344	12	2	17	1
1,242	3,57	326	12	2	18	1
-3,196	-0,076	498	12	2	19	1
-3,828	-1,229	316	12	2	20	1
2,67	4,649	372	12	2	21	1
-1,334	0,761	322	12	2	2	2
-0,437	1,186	312	12	2	3	2
0,306	1,844	486	12	2	4	2
-1,365	2,326	496	12	2	5	2
-5,915	-2,008	314	12	2	6	2
-1,056	1,1	386	12	2	7	2
4,054	5,947	322	12	2	9	2
-2,557	2,293	488	12	2	10	2
0,333	5,53	494	12	2	13	2
-3,699	-2,058	330	12	2	15	2
-2,678	1,005	302	12	2	16	2
-4,242	-1,858	312	12	2	17	2
-1,145	2,437	314	12	2	18	2
5,389	8,22	490	12	2	19	2
-2,84	-0,086	336	12	2	20	2
1,571	5,366	394	12	2	21	2
-2,968	-0,621	342	12	2	2	3
0,159	1,616	316	12	2	3	3
-1,396	1,079	302	12	2	4	3
-1,719	1,412	318	12	2	5	3
-4,998	-2,254	494	12	2	6	3
-2,084	0,492	500	12	2	7	3
-1,148	1,156	310	12	2	9	3
-4,553	-0,853	494	12	2	10	3
-4,448	-0,381	496	12	2	13	3
-2,869	0,107	326	12	2	15	3
-1,072	3,274	500	12	2	16	3
-3,543	0,432	318	12	2	17	3
-0,207	1,491	392	12	2	18	3
-4,437	-1,744	496	12	2	19	3
-5,07	-2,412	500	12	2	20	3
1,896	4,051	342	12	2	21	3
-3,698	-2,118	324	12	2	2	4
-0,383	1,142	304	12	2	3	4
-1,798	-0,508	480	12	2	4	4
-2,152	-0,023	330	12	2	5	4
-3,331	0,37	314	12	2	6	4
-2,625	0,954	496	12	2	7	4
3,379	4,739	308	12	2	9	4
-5,664	-1,485	490	12	2	10	4
-3,141	-0,473	310	12	2	13	4

-3,697	-2,536	326	12	2	15	4
-1,477	-0,436	426	12	2	16	4
-4,09	-2,439	320	12	2	17	4
-0,387	1,278	394	12	2	18	4
-2,574	-0,965	342	12	2	19	4
-5,521	-3,136	348	12	2	20	4
-1,398	1,413	358	12	2	21	4
-1,66	2,191	604	1	3	2	1
-0,132	3,995	644	1	3	3	1
1,768	5,621	610	1	3	4	1
3,85	7,127	678	1	3	5	1
-3,403	1,946	634	1	3	6	1
-1,992	0,221	664	1	3	7	1
-2,206	1,411	574	1	3	9	1
5,059	12,087	624	1	3	10	1
1,347	4,988	554	1	3	13	1
-0,286	3,032	610	1	3	15	1
1,012	7,25	668	1	3	16	1
-3,968	3,144	598	1	3	17	1
0,765	5,908	638	1	3	18	1
-6,136	-0,988	634	1	3	19	1
-1,504	0,82	620	1	3	20	1
-1,609	4,509	574	1	3	21	1
-0,139	3,318	608	1	3	2	2
-2,125	0,734	660	1	3	3	2
3,624	7,659	610	1	3	4	2
-0,618	1,774	634	1	3	5	2
-9,01	-3,619	644	1	3	6	2
4,13	6,94	624	1	3	7	2
0,459	3,634	550	1	3	9	2
5,212	13,498	630	1	3	10	2
2,424	7,604	592	1	3	13	2
-0,712	3,064	618	1	3	15	2
1,226	6,959	658	1	3	16	2
-3,731	3,882	596	1	3	17	2
-6,03	0,37	632	1	3	18	2
5	9,368	574	1	3	19	2
2,525	5,05	670	1	3	20	2
-2,59	3,465	552	1	3	21	2
-1,767	3,841	618	1	3	2	3
-2,286	0,718	616	1	3	3	3
0,956	5,13	622	1	3	4	3
-3,587	0,095	656	1	3	5	3
-9,938	-4,246	632	1	3	6	3
1,328	3,509	624	1	3	7	3
-4,91	-0,247	558	1	3	9	3
2,996	10,555	618	1	3	10	3
1,61	6,727	598	1	3	13	3

-2,05	1,223	622	1	3	15	3
1,01	6,794	698	1	3	16	3
-6,443	-1,634	604	1	3	17	3
-2,686	2,088	668	1	3	18	3
-5,694	-0,604	582	1	3	19	3
-4,152	-1,154	604	1	3	20	3
-6,203	-0,046	556	1	3	21	3
-3,579	0,687	610	1	3	2	4
-3,609	-1,053	636	1	3	3	4
-0,135	4,136	614	1	3	4	4
-4,115	-1,758	682	1	3	5	4
-6,321	-1,971	660	1	3	6	4
-0,767	1,07	612	1	3	7	4
-2,301	0,919	568	1	3	9	4
1,144	6,738	620	1	3	10	4
2,615	5,582	640	1	3	13	4
-2,928	-0,229	630	1	3	15	4
0,477	5,204	642	1	3	16	4
-3,495	3,036	622	1	3	17	4
-3,028	1,591	632	1	3	18	4
-1,184	2,605	630	1	3	19	4
-0,221	2,442	668	1	3	20	4
-3,011	2,696	556	1	3	21	4
-1,557	1,972	604	2	3	2	1
2,468	5,642	636	2	3	3	1
2,643	5,891	604	2	3	4	1
4,939	7,68	624	2	3	5	1
-2,018	2,71	666	2	3	6	1
-0,689	0,799	598	2	3	7	1
-1,432	1,449	592	2	3	9	1
2,302	7,408	624	2	3	10	1
1,24	4,128	556	2	3	13	1
-0,394	1,539	594	2	3	15	1
1,18	5,606	668	2	3	16	1
-3,951	2,37	596	2	3	17	1
0,42	3,947	636	2	3	18	1
-4,094	-1,234	584	2	3	19	1
-3,039	-1,345	550	2	3	20	1
0,375	7,05	558	2	3	21	1
1,245	3,978	590	2	3	2	2
-0,25	1,874	660	2	3	3	2
3,085	6,081	608	2	3	4	2
0,588	1,743	594	2	3	5	2
-7,355	-2,019	642	2	3	6	2
-1,19	0,553	624	2	3	7	2
2,193	3,487	602	2	3	9	2
2,41	8,442	612	2	3	10	2
2,814	7,292	588	2	3	13	2

-2,466	-0,131	604	2	3	15	2
1,267	5,213	650	2	3	16	2
-2,537	4,836	594	2	3	17	2
-4,881	-0,637	646	2	3	18	2
6,761	10,38	574	2	3	19	2
1,779	3,955	684	2	3	20	2
-1,652	5,106	550	2	3	21	2
-1,284	4,151	614	2	3	2	3
1,519	3,814	652	2	3	3	3
0,235	3,437	620	2	3	4	3
-1,805	0,902	632	2	3	5	3
-7,76	-3,636	630	2	3	6	3
0,187	2,705	620	2	3	7	3
-4,199	-0,999	560	2	3	9	3
0,496	6,414	616	2	3	10	3
-0,002	3,33	614	2	3	13	3
-2,038	0,307	616	2	3	15	3
3,344	7,766	662	2	3	16	3
-6,041	-1,674	604	2	3	17	3
-2,257	2,903	670	2	3	18	3
-4,837	-1,63	582	2	3	19	3
-3,826	-1,545	602	2	3	20	3
-3,854	1,932	566	2	3	21	3
-3,513	-0,31	610	2	3	2	4
0,795	3,158	652	2	3	3	4
0,673	3,688	612	2	3	4	4
-4,034	-1,878	682	2	3	5	4
-3,601	0,557	660	2	3	6	4
-2,988	-1,246	674	2	3	7	4
2,279	3,724	558	2	3	9	4
0,201	5,517	614	2	3	10	4
4,245	6,866	590	2	3	13	4
-3,017	-1,241	634	2	3	15	4
3,22	6,794	636	2	3	16	4
-2,674	3,122	620	2	3	17	4
-2,411	0,384	700	2	3	18	4
-0,832	1,826	578	2	3	19	4
-1,751	0,209	668	2	3	20	4
-1,638	4,74	558	2	3	21	4
-0,585	2,186	602	3	3	2	1
3,425	6,199	636	3	3	3	1
2,677	5,264	604	3	3	4	1
5,16	7,649	624	3	3	5	1
-0,351	4,066	666	3	3	6	1
-0,144	1,087	598	3	3	7	1
-0,816	1,681	594	3	3	9	1
1,319	4,84	606	3	3	10	1
1,012	3,893	588	3	3	13	1

-53,845	-49,407	510	3	3	15	1
2,303	6,022	668	3	3	16	1
-4,355	0,992	596	3	3	17	1
0,089	2,345	636	3	3	18	1
-3,748	-1,116	568	3	3	19	1
-3,589	-1,196	550	3	3	20	1
-0,488	7,025	540	3	3	21	1
1,73	4,058	584	3	3	2	2
0,608	2,294	660	3	3	3	2
3,099	5,48	606	3	3	4	2
0,184	1,077	512	3	3	5	2
-5,46	-0,379	642	3	3	6	2
-1,177	1,161	508	3	3	7	2
2,96	5,208	670	3	3	9	2
2,209	6,21	612	3	3	10	2
4,321	7,975	588	3	3	13	2
5,927	8,204	590	3	3	15	2
0,551	3,5	650	3	3	16	2
-2,286	4,301	594	3	3	17	2
-4,996	-2,364	616	3	3	18	2
6,119	9,51	574	3	3	19	2
0,784	2,72	696	3	3	20	2
-3,364	3,168	548	3	3	21	2
-0,954	3,837	614	3	3	2	3
2,819	4,955	656	3	3	3	3
0,054	2,948	618	3	3	4	3
-2,259	-0,048	630	3	3	5	3
-6,182	-2,709	628	3	3	6	3
-0,161	1,758	618	3	3	7	3
-3,304	-1,59	628	3	3	9	3
0,112	4,531	614	3	3	10	3
0,193	3,047	614	3	3	13	3
3,986	5,734	584	3	3	15	3
2,566	6,19	676	3	3	16	3
-5,443	-2,005	604	3	3	17	3
-1,791	3,425	672	3	3	18	3
-4,915	-2,909	610	3	3	19	3
-6,069	-3,78	600	3	3	20	3
-4,613	1,137	536	3	3	21	3
-2,468	-0,073	610	3	3	2	4
2,739	4,875	652	3	3	3	4
1,31	3,596	612	3	3	4	4
-4,036	-2,275	682	3	3	5	4
-1,311	2,589	660	3	3	6	4
-2,189	0,264	680	3	3	7	4
3,774	5,17	626	3	3	9	4
0,383	4,675	612	3	3	10	4
5,067	7,587	590	3	3	13	4

-31,337	-28,61	542	3	3	15	4
4,14	7,047	656	3	3	16	4
-2,415	2,695	620	3	3	17	4
-1,237	1,917	698	3	3	18	4
-0,808	2,088	578	3	3	19	4
-2,955	-1,592	668	3	3	20	4
-2,581	3,259	560	3	3	21	4
-2,07	0,019	534	4	3	2	1
1,887	3,826	632	4	3	3	1
0,512	1,548	574	4	3	4	1
4,926	8,581	508	4	3	5	1
-0,669	1,371	662	4	3	6	1
-5,191	-2,337	504	4	3	7	1
-1,849	1,063	664	4	3	9	1
-4,044	-0,421	678	4	3	10	1
0,022	3,098	500	4	3	13	1
-2,669	-0,52	622	4	3	15	1
-0,401	2,468	694	4	3	16	1
-3,413	-1,52	630	4	3	17	1
-1,158	0,473	604	4	3	18	1
-4,041	-1,279	532	4	3	19	1
-5,858	-1,59	532	4	3	20	1
-4,283	2,023	528	4	3	21	1
-1,95	-0,765	526	4	3	2	2
0,197	1,774	660	4	3	3	2
0,798	1,863	604	4	3	4	2
-1,264	2,708	512	4	3	5	2
-4,769	-2,461	642	4	3	6	2
-4,249	-0,708	502	4	3	7	2
1,171	4,43	506	4	3	9	2
-4,709	0,349	694	4	3	10	2
2,291	4,989	596	4	3	13	2
-4,972	-3,145	582	4	3	15	2
0,823	2,35	604	4	3	16	2
-3,136	0,12	594	4	3	17	2
-2,97	-0,282	608	4	3	18	2
7,615	9,947	538	4	3	19	2
-4,627	-1,547	518	4	3	20	2
-6,939	-0,884	500	4	3	21	2
-1,147	0,594	616	4	3	2	3
2,177	4,124	658	4	3	3	3
-1,153	0,109	618	4	3	4	3
-1,073	0,744	500	4	3	5	3
-3,015	-1,469	652	4	3	6	3
-3,728	-1,618	512	4	3	7	3
-2,945	-0,934	646	4	3	9	3
-4,576	-1,708	688	4	3	10	3
-0,94	1,407	608	4	3	13	3

-3,449	-2,343	614	4	3	15	3
3,523	5,87	678	4	3	16	3
-3,719	-2,793	642	4	3	17	3
-2,224	0,711	670	4	3	18	3
-4,87	-1,284	510	4	3	19	3
-8,343	-4,685	532	4	3	20	3
-6,223	-1,576	506	4	3	21	3
-2,656	-0,807	556	4	3	2	4
3,188	4,618	652	4	3	3	4
0,24	1,469	642	4	3	4	4
-2,109	0,399	502	4	3	5	4
-1,666	1,048	668	4	3	6	4
-4,858	-3,345	672	4	3	7	4
2,754	4,07	658	4	3	9	4
-4,332	-1,557	686	4	3	10	4
1,083	3,515	600	4	3	13	4
-2,77	-1,114	598	4	3	15	4
3,546	5,848	686	4	3	16	4
-3,016	-0,14	624	4	3	17	4
-0,885	1,748	608	4	3	18	4
-1,368	1,38	524	4	3	19	4
-8,13	-4,683	530	4	3	20	4
-6,02	-4,045	536	4	3	21	4
-0,919	2,528	606	5	3	2	1
-0,999	2,239	644	5	3	3	1
2,26	5,572	610	5	3	4	1
3,524	8,17	662	5	3	5	1
-0,887	3,309	654	5	3	6	1
-0,308	2,409	662	5	3	7	1
0,713	3,631	574	5	3	9	1
7,065	13,097	624	5	3	10	1
3,664	7,686	554	5	3	13	1
106,504	112,733	694	5	3	15	1
3,283	10,102	666	5	3	16	1
-2,036	3,566	614	5	3	17	1
-0,253	3,452	636	5	3	18	1
-6,551	-1,102	634	5	3	19	1
-0,794	0,552	572	5	3	20	1
0,134	5,554	606	5	3	21	1
-0,253	2,751	608	5	3	2	2
-2,016	0,442	660	5	3	3	2
4,305	7,914	610	5	3	4	2
-0,374	1,99	584	5	3	5	2
-6,712	-2,948	646	5	3	6	2
0,921	4,99	624	5	3	7	2
1,139	3,436	650	5	3	9	2
6,404	13,706	628	5	3	10	2
-0,094	3,078	558	5	3	13	2

82,012	90,337	686	5	3	15	2
4,044	9,312	658	5	3	16	2
-1,355	4,472	612	5	3	17	2
-5,603	-1,316	632	5	3	18	2
4,558	9,094	640	5	3	19	2
-0,009	1,548	588	5	3	20	2
-0,578	3,173	550	5	3	21	2
4,661	9,517	618	5	3	2	3
-2,464	0,024	612	5	3	3	3
1,82	5,516	622	5	3	4	3
-3,863	-0,538	636	5	3	5	3
-7,784	-2,884	634	5	3	6	3
0,232	2,412	628	5	3	7	3
-1,914	2,062	558	5	3	9	3
4,069	10,702	616	5	3	10	3
-0,251	3,442	582	5	3	13	3
-104,33	-93,171	524	5	3	15	3
1,267	6,682	662	5	3	16	3
-3,239	0,608	606	5	3	17	3
-5,703	-2,656	636	5	3	18	3
-5,203	-0,83	598	5	3	19	3
-2,972	-0,899	604	5	3	20	3
-3,395	1,038	572	5	3	21	3
0,475	3,651	612	5	3	2	4
-3,674	-1,85	652	5	3	3	4
0,959	4,626	616	5	3	4	4
-2,835	-1,267	654	5	3	5	4
-2,33	1,131	664	5	3	6	4
-1,513	1,17	662	5	3	7	4
1,187	3,834	602	5	3	9	4
2,958	7,533	624	5	3	10	4
2,334	5,396	640	5	3	13	4
-55,04	-50,354	532	5	3	15	4
1,238	6,298	644	5	3	16	4
-1,804	3,658	622	5	3	17	4
-4,8	-1,753	632	5	3	18	4
-2,512	0,859	628	5	3	19	4
0,201	1,999	670	5	3	20	4
0,5	3,699	552	5	3	21	4
0,324	3,344	622	6	3	2	1
1,144	3,391	632	6	3	3	1
2,763	5,54	608	6	3	4	1
4,433	6,167	628	6	3	5	1
2,359	5,065	656	6	3	6	1
-0,624	0,692	616	6	3	7	1
0,527	2,674	620	6	3	9	1
4,734	8,148	624	6	3	10	1
3,103	6,078	586	6	3	13	1

-0,656	1,821	610	6	3	15	1
4,575	8,402	668	6	3	16	1
-0,183	4,548	612	6	3	17	1
-0,567	1,749	588	6	3	18	1
-5,106	-2,177	634	6	3	19	1
-2,27	-0,497	586	6	3	20	1
1,059	5,916	564	6	3	21	1
1,223	3,626	598	6	3	2	2
-0,817	0,466	660	6	3	3	2
2,578	4,954	626	6	3	4	2
1,648	2,774	584	6	3	5	2
-3,47	-0,052	644	6	3	6	2
-1,4	0,307	692	6	3	7	2
3,04	4,438	650	6	3	9	2
4,266	8,831	612	6	3	10	2
3,439	6,371	558	6	3	13	2
-2,058	-0,002	604	6	3	15	2
4,762	7,613	672	6	3	16	2
0,969	5,018	612	6	3	17	2
-2,316	0,286	646	6	3	18	2
8,767	11,566	638	6	3	19	2
-0,623	0,598	588	6	3	20	2
0,997	5,402	564	6	3	21	2
1,625	5,115	616	6	3	2	3
1,074	2,869	608	6	3	3	3
1,208	3,697	620	6	3	4	3
-2,437	-0,272	630	6	3	5	3
-2,837	0,069	628	6	3	6	3
-2,882	-1,443	632	6	3	7	3
-1,265	1,435	626	6	3	9	3
3,333	7,761	604	6	3	10	3
-0,144	2,517	600	6	3	13	3
0,159	2,68	614	6	3	15	3
2,233	5,433	662	6	3	16	3
-2,572	0,873	604	6	3	17	3
-2,401	0,081	570	6	3	18	3
-2,719	-0,558	612	6	3	19	3
-3,914	-2,387	602	6	3	20	3
-2,04	1,734	556	6	3	21	3
-0,726	1,362	610	6	3	2	4
0,209	1,859	652	6	3	3	4
1,447	3,927	630	6	3	4	4
-2,218	-1,63	616	6	3	5	4
-0,195	2,815	658	6	3	6	4
-4,094	-2,33	678	6	3	7	4
3,065	4,884	624	6	3	9	4
2,696	6,269	620	6	3	10	4
3,367	6,207	660	6	3	13	4

-1,782	-0,023	634	6	3	15	4
3,887	7,389	656	6	3	16	4
-0,271	3,875	622	6	3	17	4
-2,239	-0,801	628	6	3	18	4
-0,787	0,933	630	6	3	19	4
-0,991	0,176	588	6	3	20	4
0,771	3,65	572	6	3	21	4
0,413	2,761	632	7	3	2	1
1,287	3,275	632	7	3	3	1
1,607	3,575	610	7	3	4	1
4,775	7,347	508	7	3	5	1
3,055	5,269	666	7	3	6	1
-2,185	-1,25	630	7	3	7	1
0,506	2,75	660	7	3	9	1
3,565	5,52	512	7	3	10	1
1,912	4,956	586	7	3	13	1
-1,317	0,538	610	7	3	15	1
4,191	7,376	682	7	3	16	1
-0,431	3,46	610	7	3	17	1
-2,134	0,327	604	7	3	18	1
-4,294	-2,586	616	7	3	19	1
-2,197	0,116	534	7	3	20	1
-0,186	3,799	558	7	3	21	1
1,04	2,672	622	7	3	2	2
-0,71	0,458	662	7	3	3	2
1,309	2,829	608	7	3	4	2
0,621	3,575	514	7	3	5	2
-3,587	-0,793	646	7	3	6	2
-3,308	-1,601	692	7	3	7	2
3,194	4,875	652	7	3	9	2
3,185	5,798	510	7	3	10	2
3,437	5,619	558	7	3	13	2
-2,825	-1,104	588	7	3	15	2
4,975	6,795	672	7	3	16	2
0,912	4,061	598	7	3	17	2
-3,638	-1,512	600	7	3	18	2
10,548	12,657	572	7	3	19	2
-0,644	0,64	520	7	3	20	2
-0,067	2,954	564	7	3	21	2
0,752	4,121	630	7	3	2	3
1,859	3,61	606	7	3	3	3
-0,142	1,924	620	7	3	4	3
-1,991	-0,206	500	7	3	5	3
-1,951	-0,115	624	7	3	6	3
-5,14	-3,679	630	7	3	7	3
-1,478	1,044	626	7	3	9	3
2,881	5,496	602	7	3	10	3
-0,554	1,631	576	7	3	13	3

-0,365	1,52	614	7	3	15	3
3,712	5,793	662	7	3	16	3
-2,236	0,083	604	7	3	17	3
-2,79	0,573	584	7	3	18	3
-2,579	-0,819	612	7	3	19	3
-4,585	-2,266	534	7	3	20	3
-2,39	1,054	536	7	3	21	3
-1,535	-0,063	626	7	3	2	4
1,722	3,13	652	7	3	3	4
0,746	2,371	634	7	3	4	4
-2,008	-0,676	518	7	3	5	4
1,151	3,96	660	7	3	6	4
-4,857	-3,378	696	7	3	7	4
3,932	5,805	628	7	3	9	4
2,807	4,97	614	7	3	10	4
3,121	5,156	660	7	3	13	4
-1,858	-0,316	600	7	3	15	4
5,703	8,007	658	7	3	16	4
-0,195	3,035	622	7	3	17	4
-3,364	-2,296	618	7	3	18	4
0,376	1,68	578	7	3	19	4
-2,222	-1,29	522	7	3	20	4
-0,547	1,548	572	7	3	21	4
0,03	2,471	502	8	3	2	1
1,252	3,187	614	8	3	3	1
0,83	1,989	610	8	3	4	1
4,517	9,781	508	8	3	5	1
1,594	3,328	666	8	3	6	1
-3,66	-1,861	506	8	3	7	1
-1,115	1,627	666	8	3	9	1
1,377	6,827	510	8	3	10	1
1,554	4,699	588	8	3	13	1
-3,807	-2,217	610	8	3	15	1
2,691	5,917	696	8	3	16	1
-0,539	2,605	610	8	3	17	1
-2,842	-0,135	606	8	3	18	1
-3,345	-2,018	532	8	3	19	1
-4,414	-0,35	530	8	3	20	1
-1,034	3,329	516	8	3	21	1
0,882	1,818	622	8	3	2	2
-1,031	0,233	662	8	3	3	2
0,618	1,992	640	8	3	4	2
-0,033	5,231	514	8	3	5	2
-3,289	-1,145	646	8	3	6	2
-4,649	-2,573	508	8	3	7	2
2,349	5,295	504	8	3	9	2
1,305	7,574	508	8	3	10	2
3,109	5,18	588	8	3	13	2

-6,006	-4,338	588	8	3	15	2
4,592	5,78	670	8	3	16	2
0,358	3,084	596	8	3	17	2
-3,551	-0,684	600	8	3	18	2
9,568	11,034	572	8	3	19	2
-1,996	0,961	520	8	3	20	2
-2,68	1,39	512	8	3	21	2
-0,221	2,615	630	8	3	2	3
1,636	3,484	606	8	3	3	3
-1,272	0,417	618	8	3	4	3
-0,248	3,172	500	8	3	5	3
-1,197	0,207	622	8	3	6	3
-5,345	-4,181	620	8	3	7	3
-2,662	-0,821	626	8	3	9	3
1,085	6,409	500	8	3	10	3
-0,442	2,26	612	8	3	13	3
-3,567	-1,872	616	8	3	15	3
5,034	6,647	592	8	3	16	3
-1,532	-0,252	604	8	3	17	3
-2,413	1,71	582	8	3	18	3
-3,655	-1,789	546	8	3	19	3
-6,837	-3,017	522	8	3	20	3
-3,094	0,358	524	8	3	21	3
-1,783	-0,558	640	8	3	2	4
2,103	3,349	652	8	3	3	4
0,136	1,384	642	8	3	4	4
-1,218	1,556	502	8	3	5	4
1,592	4,289	668	8	3	6	4
-5,409	-4,121	674	8	3	7	4
3,551	5,214	628	8	3	9	4
1,618	3,894	508	8	3	10	4
2,784	4,808	602	8	3	13	4
-4,768	-3,49	598	8	3	15	4
6,4	8,369	588	8	3	16	4
-0,642	1,995	622	8	3	17	4
-3,624	-2,277	616	8	3	18	4
0,652	1,911	578	8	3	19	4
-4,917	-2,345	522	8	3	20	4
-2,488	-0,883	572	8	3	21	4
-0,019	3,268	604	9	3	2	1
-1,914	1,821	644	9	3	3	1
1,431	4,413	608	9	3	4	1
2,505	5,511	644	9	3	5	1
-1,861	2,831	636	9	3	6	1
2,138	4,457	666	9	3	7	1
0,562	3,142	574	9	3	9	1
3,283	9,636	626	9	3	10	1
3,365	5,831	648	9	3	13	1

2,511	5,564	628	9	3	15	1
3,158	8,551	650	9	3	16	1
-2,156	2,752	630	9	3	17	1
1,836	5,414	672	9	3	18	1
-3,583	0,774	650	9	3	19	1
-0,152	3,129	636	9	3	20	1
-0,024	5,151	608	9	3	21	1
1,821	4,716	610	9	3	2	2
-1,563	1,537	660	9	3	3	2
3,368	6,787	610	9	3	4	2
0,881	3,729	640	9	3	5	2
-9,029	-4,904	644	9	3	6	2
0,899	3,147	626	9	3	7	2
2,725	4,612	550	9	3	9	2
3,327	10,171	630	9	3	10	2
1,571	6,163	592	9	3	13	2
1,696	5,112	634	9	3	15	2
2,626	7,701	658	9	3	16	2
-2,107	3,509	596	9	3	17	2
-2,242	3,506	646	9	3	18	2
2,632	7,483	640	9	3	19	2
1,652	4,845	670	9	3	20	2
-0,18	3,381	636	9	3	21	2
-2,396	1,627	602	9	3	2	3
-3,53	-0,552	652	9	3	3	3
1,87	5,072	622	9	3	4	3
-2,733	0,287	656	9	3	5	3
-7,976	-4,821	638	9	3	6	3
2,254	4,513	668	9	3	7	3
-3,321	0,798	560	9	3	9	3
0,742	6,859	630	9	3	10	3
1,225	6,252	600	9	3	13	3
0,437	2,882	640	9	3	15	3
2,918	7,231	662	9	3	16	3
-3,165	0,767	604	9	3	17	3
-2,807	1,531	654	9	3	18	3
-0,732	3,361	614	9	3	19	3
-1,917	0,293	688	9	3	20	3
-3,238	-0,312	568	9	3	21	3
-2,597	0,381	612	9	3	2	4
-4,199	-2,279	636	9	3	3	4
-0,111	3,074	616	9	3	4	4
-4,261	-2,264	684	9	3	5	4
-3,752	-0,867	654	9	3	6	4
-0,509	1,193	616	9	3	7	4
1,546	4,078	568	9	3	9	4
1,142	5,567	630	9	3	10	4
0,503	3,406	634	9	3	13	4

-0,46	2,156	628	9	3	15	4
2,292	6,527	644	9	3	16	4
-1,926	3,275	622	9	3	17	4
-0,989	3,044	664	9	3	18	4
1,574	5,011	628	9	3	19	4
-1,068	1,141	654	9	3	20	4
-2,096	1,473	554	9	3	21	4
-0,97	1,237	608	10	3	2	1
0,895	2,69	640	10	3	3	1
2,307	4,348	604	10	3	4	1
2,254	3,692	690	10	3	5	1
1,73	4,838	666	10	3	6	1
-0,459	2,154	666	10	3	7	1
-0,297	1,603	578	10	3	9	1
0,521	3,961	626	10	3	10	1
1,343	3,886	606	10	3	13	1
-0,027	2,566	696	10	3	15	1
1,353	4,352	638	10	3	16	1
-3,51	-0,388	630	10	3	17	1
2,578	5,293	672	10	3	18	1
-2,742	-0,795	700	10	3	19	1
-0,343	1,288	636	10	3	20	1
2,885	5,486	540	10	3	21	1
0,586	2,533	614	10	3	2	2
-0,646	1,026	630	10	3	3	2
2,735	4,829	610	10	3	4	2
-0,833	0,796	680	10	3	5	2
-4,821	-1,766	642	10	3	6	2
-1,012	0,876	632	10	3	7	2
4,829	6,583	670	10	3	9	2
0,445	3,604	632	10	3	10	2
2,052	5,403	594	10	3	13	2
-0,679	1,685	638	10	3	15	2
0,546	3,637	662	10	3	16	2
-4,266	0,175	594	10	3	17	2
-0,216	3,617	646	10	3	18	2
6,765	8,902	638	10	3	19	2
1,62	4,723	670	10	3	20	2
0,829	3,648	548	10	3	21	2
-2,081	1,119	602	10	3	2	3
0,32	2,675	656	10	3	3	3
1,077	3,483	606	10	3	4	3
-2,019	-0,482	658	10	3	5	3
-4,263	-2,402	634	10	3	6	3
1,921	3,677	618	10	3	7	3
-2,26	-0,169	560	10	3	9	3
-1,468	1,157	616	10	3	10	3
-0,963	1,996	630	10	3	13	3

-1,642	-0,01	626	10	3	15	3
2,646	6,004	676	10	3	16	3
-4,979	-2,166	618	10	3	17	3
-0,482	3,903	670	10	3	18	3
-3,195	-1,262	696	10	3	19	3
-1,948	0,012	700	10	3	20	3
0,332	2,341	566	10	3	21	3
-2,313	-0,639	612	10	3	2	4
-0,453	1,675	648	10	3	3	4
0,349	2,482	614	10	3	4	4
-4,443	-2,7	684	10	3	5	4
-0,574	2,127	666	10	3	6	4
-0,898	1,3	676	10	3	7	4
3,74	5,002	604	10	3	9	4
-0,493	2,422	614	10	3	10	4
0,897	2,805	622	10	3	13	4
-2,707	-1,289	636	10	3	15	4
1,13	3,169	656	10	3	16	4
-2,967	0,906	622	10	3	17	4
1,159	4,248	668	10	3	18	4
0,231	2,368	658	10	3	19	4
-1,725	-0,261	654	10	3	20	4
0,023	2,942	558	10	3	21	4
-2,055	-0,694	592	11	3	2	1
1,146	2,617	636	11	3	3	1
1,031	2,31	622	11	3	4	1
2,575	4,687	508	11	3	5	1
0,538	3,307	666	11	3	6	1
-2,517	-0,715	666	11	3	7	1
-0,972	0,879	660	11	3	9	1
-2,446	-0,448	678	11	3	10	1
0,476	2,889	500	11	3	13	1
-2,089	0,441	696	11	3	15	1
-1,112	1,271	514	11	3	16	1
-3,824	-1,611	630	11	3	17	1
2,102	4,758	670	11	3	18	1
-3,301	-0,9	516	11	3	19	1
-1,175	0,042	534	11	3	20	1
1,554	5,364	528	11	3	21	1
-0,284	0,843	622	11	3	2	2
-0,294	0,809	630	11	3	3	2
1,916	3,558	610	11	3	4	2
-1,195	0,185	512	11	3	5	2
-4,947	-2,253	642	11	3	6	2
-1,177	0,591	508	11	3	7	2
3,676	6,715	670	11	3	9	2
-2,234	1,208	676	11	3	10	2
2,102	5,19	594	11	3	13	2

-2,193	-0,321	638	11	3	15	2
-1,036	0,982	664	11	3	16	2
-3,422	0,611	596	11	3	17	2
-0,676	1,663	648	11	3	18	2
7,853	10,275	538	11	3	19	2
1,595	4,182	670	11	3	20	2
-1,086	2,028	532	11	3	21	2
-2,306	-0,058	618	11	3	2	3
0,919	3,053	658	11	3	3	3
-0,402	1,554	620	11	3	4	3
-2,999	-1,524	670	11	3	5	3
-3,035	-1,786	624	11	3	6	3
0,691	2,265	680	11	3	7	3
-2,689	-1,518	678	11	3	9	3
-3,618	-1,284	666	11	3	10	3
-1,848	0,957	626	11	3	13	3
-0,979	0,32	690	11	3	15	3
0,715	3,938	676	11	3	16	3
-4,736	-2,805	618	11	3	17	3
0,001	4,823	672	11	3	18	3
-4,999	-2,91	696	11	3	19	3
-3,004	-1,999	532	11	3	20	3
-0,639	2,066	504	11	3	21	3
-2,719	-1,361	558	11	3	2	4
1,045	2,96	650	11	3	3	4
-0,395	1,091	632	11	3	4	4
-4,277	-2,648	682	11	3	5	4
-0,358	2,863	668	11	3	6	4
-1,13	1,667	678	11	3	7	4
3,009	4,221	656	11	3	9	4
-2,802	-0,332	662	11	3	10	4
0,625	2,084	602	11	3	13	4
-2,723	-1,794	648	11	3	15	4
0,284	1,815	672	11	3	16	4
-2,432	0,775	622	11	3	17	4
1,118	4,372	700	11	3	18	4
-0,493	0,793	524	11	3	19	4
-3,011	-0,893	506	11	3	20	4
-0,754	1,685	654	11	3	21	4
-3,192	-2,376	584	12	3	2	1
1,043	2,662	660	12	3	3	1
0,281	1,279	622	12	3	4	1
3,704	6,803	508	12	3	5	1
-1,731	0,84	666	12	3	6	1
-3,383	-1,025	504	12	3	7	1
-1,144	0,98	662	12	3	9	1
-4,829	-1,037	678	12	3	10	1
-0,39	3,845	500	12	3	13	1

-5,382	-4,122	622	12	3	15	1
-2,785	1,648	512	12	3	16	1
-4,023	-2,207	630	12	3	17	1
1,503	3,619	532	12	3	18	1
-3,443	0,622	516	12	3	19	1
-2,841	0,099	530	12	3	20	1
-0,517	5,291	526	12	3	21	1
-1,484	-0,272	546	12	3	2	2
-0,434	0,7	660	12	3	3	2
1,418	2,831	654	12	3	4	2
-1,183	1,571	512	12	3	5	2
-6,363	-4,097	640	12	3	6	2
-2,854	-1,491	678	12	3	7	2
2,837	6,457	670	12	3	9	2
-4,715	0,413	696	12	3	10	2
2,67	5,801	594	12	3	13	2
-5,573	-4,211	646	12	3	15	2
-1,837	0,025	504	12	3	16	2
-4,131	-0,689	594	12	3	17	2
-1,087	0,746	608	12	3	18	2
7,234	10,438	538	12	3	19	2
-0,566	1,7	504	12	3	20	2
-4,041	1,087	512	12	3	21	2
-2,248	-0,128	630	12	3	2	3
1,102	3,143	658	12	3	3	3
-1,14	0,349	634	12	3	4	3
-2,233	-0,194	500	12	3	5	3
-3,951	-2,969	624	12	3	6	3
-1,434	0,492	500	12	3	7	3
-2,524	-0,806	680	12	3	9	3
-5,394	-2,188	666	12	3	10	3
-2,194	0,551	626	12	3	13	3
-3,062	-2,12	688	12	3	15	3
0,012	3,274	500	12	3	16	3
-4,921	-3,654	640	12	3	17	3
-0,637	3,745	670	12	3	18	3
-5,593	-1,811	512	12	3	19	3
-5,541	-2,412	500	12	3	20	3
-2,501	2,086	504	12	3	21	3
-2,799	-1,022	556	12	3	2	4
1,736	3,262	652	12	3	3	4
-0,591	0,837	636	12	3	4	4
-3,691	-1,273	502	12	3	5	4
-1,768	1,663	674	12	3	6	4
-2,175	0,728	680	12	3	7	4
3,387	5,03	658	12	3	9	4
-5,17	-2,262	670	12	3	10	4
0,212	2,369	502	12	3	13	4

-4,226	-3,016	508	12	3	15	4
-0,139	1,805	512	12	3	16	4
-2,765	0,251	622	12	3	17	4
1,337	4,173	700	12	3	18	4
-1,41	1,78	524	12	3	19	4
-5,037	-1,532	516	12	3	20	4
-2,496	0,284	656	12	3	21	4

Appendix E - Sentences Task Data Organization and Coding

Mean Amplitude Between two fixed latencies	Peak Amplitude	Peak Latency	ROI	ERP time range	Subject	Class
2,695	6,322	238	1	1	2	1
1,209	5,036	280	1	1	3	1
0,184	3,899	246	1	1	4	1
3,15	8,482	298	1	1	5	1
-0,924	5,058	270	1	1	6	1
-3,033	-0,933	278	1	1	7	1
2,056	5,035	256	1	1	9	1
3,433	10,117	260	1	1	10	1
4,443	9,369	194	1	1	13	1
6,888	12,654	258	1	1	15	1
2,252	6,141	294	1	1	16	1
1,234	7,474	250	1	1	17	1
3,783	8,775	272	1	1	18	1
4,194	8,646	290	1	1	19	1
2,136	5,03	216	1	1	20	1
0,293	4,854	206	1	1	21	1
-0,257	3,401	248	1	1	2	2
0,578	2,933	280	1	1	3	2
0,499	5,086	250	1	1	4	2
0,864	4,585	290	1	1	5	2
1,635	5,877	262	1	1	6	2
-3,391	-0,759	278	1	1	7	2
2,327	6,219	288	1	1	9	2
1,251	5,499	256	1	1	10	2
2,34	4,71	188	1	1	13	2
-5,824	-2,673	256	1	1	15	2
0,366	6,362	296	1	1	16	2
3,715	10,627	258	1	1	17	2
1,25	5,498	276	1	1	18	2
-1,476	2,936	270	1	1	19	2
1,837	4,195	234	1	1	20	2
2,147	7,598	186	1	1	21	2
-0,44	4,119	246	1	1	2	3
1,138	4,622	284	1	1	3	3
1,038	5,345	256	1	1	4	3
5,128	7,207	248	1	1	5	3
-1,043	4,624	278	1	1	6	3

0,164	4,629	290	1	1	7	3
5,094	8,493	212	1	1	9	3
0,196	6,347	266	1	1	10	3
2,222	3,451	182	1	1	13	3
-3,148	-0,615	252	1	1	15	3
-0,21	5,248	298	1	1	16	3
2,881	10,383	250	1	1	17	3
0,127	5,157	296	1	1	18	3
-0,422	4,107	262	1	1	19	3
3,794	6,552	288	1	1	20	3
2,397	9,211	178	1	1	21	3
-0,534	2,862	270	1	1	2	4
0,764	2,916	268	1	1	3	4
-0,201	3,686	264	1	1	4	4
2,74	5,357	290	1	1	5	4
0,059	5,549	286	1	1	6	4
-5,534	-0,773	288	1	1	7	4
3,225	5,846	276	1	1	9	4
2,333	8,389	254	1	1	10	4
-1,232	2,119	222	1	1	13	4
-4,298	-1,582	264	1	1	15	4
-3,027	3,909	284	1	1	16	4
2,395	8,378	266	1	1	17	4
0,858	4,841	282	1	1	18	4
-2,056	0,605	256	1	1	19	4
1,895	5,725	296	1	1	20	4
1,452	8,553	194	1	1	21	4
0,939	4,372	250	1	1	2	5
0,779	3,075	280	1	1	3	5
1,308	4,576	246	1	1	4	5
1,347	3,404	272	1	1	5	5
0,024	4,62	280	1	1	6	5
2,359	5,948	298	1	1	7	5
2,841	4,859	274	1	1	9	5
2,119	7,136	264	1	1	10	5
1,337	2,908	222	1	1	13	5
-3,803	-1,823	266	1	1	15	5
-2,144	0,512	290	1	1	16	5
-1,282	4,87	256	1	1	17	5
0,399	5,787	300	1	1	18	5
-0,414	2,814	276	1	1	19	5
0,532	1,891	298	1	1	20	5
1,787	6,614	184	1	1	21	5

0,848	4,216	238	2	1	2	1
0,55	3,01	266	2	1	3	1
-1,556	1,264	256	2	1	4	1
2,301	5,906	298	2	1	5	1
-0,873	4,21	270	2	1	6	1
-1,783	-0,685	278	2	1	7	1
-0,186	1,786	260	2	1	9	1
0,469	5,053	248	2	1	10	1
2,269	5,113	208	2	1	13	1
5,994	9,749	258	2	1	15	1
0,055	2,643	294	2	1	16	1
0,21	5,495	248	2	1	17	1
2,256	4,721	256	2	1	18	1
3,874	6,501	222	2	1	19	1
1,334	5,009	218	2	1	20	1
-2,599	2,793	188	2	1	21	1
-2,302	0,348	248	2	1	2	2
-0,613	0,884	280	2	1	3	2
-0,949	2,091	246	2	1	4	2
1,523	4,807	286	2	1	5	2
1,589	5,207	260	2	1	6	2
-1,935	0,041	282	2	1	7	2
-1,21	1,346	292	2	1	9	2
-0,429	3,448	248	2	1	10	2
2,033	4,812	222	2	1	13	2
-6,079	-5,328	256	2	1	15	2
0,335	3,931	294	2	1	16	2
1,168	6,222	256	2	1	17	2
-1,778	0,616	240	2	1	18	2
-2,133	0,822	218	2	1	19	2
-0,239	1,971	234	2	1	20	2
-0,728	4,887	202	2	1	21	2
-1,135	2,742	246	2	1	2	3
0,975	3,426	292	2	1	3	3
0,465	3,713	236	2	1	4	3
3,149	4,762	250	2	1	5	3
-0,736	4,457	292	2	1	6	3
-0,758	1,671	296	2	1	7	3
2,787	4,622	252	2	1	9	3
-0,689	3,41	246	2	1	10	3
1,018	3,85	230	2	1	13	3
-3,509	-1,63	226	2	1	15	3
-1,692	2,878	296	2	1	16	3

1,341	6,804	250	2	1	17	3
-0,213	3,075	232	2	1	18	3
-0,511	1,897	262	2	1	19	3
2,884	4,912	220	2	1	20	3
1,577	8,133	192	2	1	21	3
-1,606	0,877	268	2	1	2	4
0,929	2,256	268	2	1	3	4
-0,57	2,56	242	2	1	4	4
1,549	3,938	236	2	1	5	4
-1,088	3,094	286	2	1	6	4
-2,505	0,082	292	2	1	7	4
1,852	3,494	282	2	1	9	4
1,309	6,819	252	2	1	10	4
-2,287	0,944	222	2	1	13	4
-5,062	-3,377	218	2	1	15	4
-1,436	3,263	284	2	1	16	4
1,186	6,546	254	2	1	17	4
0,528	3,211	300	2	1	18	4
-2,057	0,401	226	2	1	19	4
0,967	2,968	296	2	1	20	4
0,338	7,587	192	2	1	21	4
0,456	3,238	250	2	1	2	5
0,631	2,172	282	2	1	3	5
0,685	3,561	240	2	1	4	5
1,225	3,06	270	2	1	5	5
-0,805	2,891	276	2	1	6	5
-1,916	-0,724	282	2	1	7	5
1,878	3,24	244	2	1	9	5
1,638	6,126	244	2	1	10	5
-0,14	2,101	228	2	1	13	5
-3,423	-2,506	222	2	1	15	5
-0,254	1,282	288	2	1	16	5
-1,257	3,974	250	2	1	17	5
-0,264	3,764	300	2	1	18	5
0,171	2,038	232	2	1	19	5
0,015	1,014	298	2	1	20	5
1,66	6,48	186	2	1	21	5
0,564	3,266	238	3	1	2	1
0,231	2,156	272	3	1	3	1
-2,166	-0,083	244	3	1	4	1
2,227	5,603	300	3	1	5	1
-1,167	3,174	268	3	1	6	1
-1,848	-1,203	278	3	1	7	1

-1,09	0,571	226	3	1	9	1
-0,597	2,812	240	3	1	10	1
2,498	5,726	224	3	1	13	1
10,418	16,416	300	3	1	15	1
-0,273	2,27	258	3	1	16	1
-0,548	3,915	246	3	1	17	1
0,834	4,005	238	3	1	18	1
3,37	5,571	224	3	1	19	1
0,556	4,424	228	3	1	20	1
-2,858	3,321	174	3	1	21	1
-2,466	-0,599	248	3	1	2	2
-0,612	0,681	254	3	1	3	2
-1,532	0,627	230	3	1	4	2
1,033	3,647	290	3	1	5	2
1,72	5,222	244	3	1	6	2
-1,915	0,03	282	3	1	7	2
-1,725	1,075	294	3	1	9	2
0,098	3,128	250	3	1	10	2
1,26	4,218	224	3	1	13	2
-5,952	-4,447	216	3	1	15	2
0,719	3,318	292	3	1	16	2
0,043	4,127	254	3	1	17	2
-3,06	-0,667	238	3	1	18	2
-1,801	1,056	206	3	1	19	2
-1,927	-0,218	234	3	1	20	2
-1,56	3,522	186	3	1	21	2
-1,416	1,477	246	3	1	2	3
1,208	3,371	290	3	1	3	3
0,319	2,784	234	3	1	4	3
2,033	2,996	250	3	1	5	3
-0,714	3,599	292	3	1	6	3
-1,118	-0,204	260	3	1	7	3
1,947	3,956	252	3	1	9	3
-0,638	2,475	236	3	1	10	3
0,745	4,33	230	3	1	13	3
-0,123	2,103	226	3	1	15	3
-1,91	2,325	296	3	1	16	3
-0,257	3,956	248	3	1	17	3
-1,078	2,915	232	3	1	18	3
-0,85	0,53	246	3	1	19	3
1,711	4,042	220	3	1	20	3
0,98	6,974	178	3	1	21	3
-1,275	0,733	268	3	1	2	4

1,163	2,057	274	3	1	3	4
-0,689	1,826	242	3	1	4	4
0,605	2,259	236	3	1	5	4
-0,538	3,667	286	3	1	6	4
-2,221	-0,091	292	3	1	7	4
1,247	3,295	300	3	1	9	4
1,35	5,638	250	3	1	10	4
-2,237	0,893	222	3	1	13	4
6,226	8,886	290	3	1	15	4
-2,306	0,933	284	3	1	16	4
0,08	4,518	252	3	1	17	4
-0,078	2,59	250	3	1	18	4
-2,338	-0,41	210	3	1	19	4
-0,025	1,588	226	3	1	20	4
-0,58	6,454	192	3	1	21	4
0,66	2,738	250	3	1	2	5
1,397	3,16	294	3	1	3	5
0,307	2,479	240	3	1	4	5
1,383	2,933	270	3	1	5	5
-0,864	2,315	274	3	1	6	5
-1,658	-0,573	292	3	1	7	5
1,845	3,254	290	3	1	9	5
1,712	5,715	242	3	1	10	5
-0,407	2,293	234	3	1	13	5
3,611	4,769	224	3	1	15	5
-0,523	0,736	250	3	1	16	5
-1,547	2,895	248	3	1	17	5
-1,301	0,206	278	3	1	18	5
0,241	1,897	230	3	1	19	5
-0,664	0,127	216	3	1	20	5
1,734	6,139	184	3	1	21	5
0,383	2,395	290	4	1	2	1
0,109	1,302	160	4	1	3	1
-1,264	-0,267	276	4	1	4	1
0,904	2,793	276	4	1	5	1
-1,209	0,911	254	4	1	6	1
-2,184	-1,334	240	4	1	7	1
-1,686	0,813	294	4	1	9	1
-3,193	1,742	300	4	1	10	1
1,311	4,589	226	4	1	13	1
6,098	9,063	254	4	1	15	1
-0,602	0,694	254	4	1	16	1
-0,423	1,383	232	4	1	17	1

-0,832	2,933	236	4	1	18	1
2,257	5,118	152	4	1	19	1
-0,285	2,59	164	4	1	20	1
-3,18	2,314	152	4	1	21	1
-1,155	-0,252	282	4	1	2	2
-0,289	1,177	252	4	1	3	2
-1,64	-0,361	230	4	1	4	2
0,319	1,569	170	4	1	5	2
1,08	3,732	244	4	1	6	2
-1,641	-1,216	238	4	1	7	2
-2,035	1,454	294	4	1	9	2
-1,154	-4,088	204	4	1	10	2
0,621	3,014	226	4	1	13	2
-5,233	-2,625	152	4	1	15	2
1,566	3,534	240	4	1	16	2
-2,138	-0,751	244	4	1	17	2
-3,848	-1,57	238	4	1	18	2
-2,663	1,92	150	4	1	19	2
-2,248	1,364	152	4	1	20	2
-2,245	1,896	154	4	1	21	2
-0,853	1,05	300	4	1	2	3
0,549	1,732	294	4	1	3	3
0,098	1,431	284	4	1	4	3
-0,492	2,312	160	4	1	5	3
0,213	3,216	298	4	1	6	3
-1,706	-1,041	250	4	1	7	3
1,13	3,035	152	4	1	9	3
-2,016	1,76	150	4	1	10	3
1,212	4,865	232	4	1	13	3
-2,203	-0,191	228	4	1	15	3
-0,236	1,875	298	4	1	16	3
-2,886	-1,065	230	4	1	17	3
-0,29	2,553	242	4	1	18	3
-0,929	2,779	156	4	1	19	3
-0,485	3,072	168	4	1	20	3
-1,75	3,807	160	4	1	21	3
-0,093	1,872	290	4	1	2	4
0,944	1,588	236	4	1	3	4
-0,121	1,279	280	4	1	4	4
-1,543	1,096	150	4	1	5	4
-0,196	2,788	294	4	1	6	4
-2,239	-0,585	150	4	1	7	4
1,055	4,24	300	4	1	9	4

-1,368	-1,311	252	4	1	10	4
-1,672	0,862	238	4	1	13	4
-4,423	-2,973	212	4	1	15	4
-2,651	-1,096	232	4	1	16	4
-0,869	0,549	270	4	1	17	4
-0,974	2,985	242	4	1	18	4
-2,494	0,883	152	4	1	19	4
-1,862	1,372	162	4	1	20	4
-2,42	2,279	296	4	1	21	4
0,888	2,905	294	4	1	2	5
0,932	2,075	260	4	1	3	5
0,11	1,036	238	4	1	4	5
0,215	2,4	156	4	1	5	5
-0,804	0,899	280	4	1	6	5
-1,682	-0,793	248	4	1	7	5
1,194	3,208	292	4	1	9	5
-0,359	3,274	294	4	1	10	5
-0,134	3,369	234	4	1	13	5
-2,892	-2,175	234	4	1	15	5
0,721	2,148	234	4	1	16	5
-1,089	1,036	236	4	1	17	5
-1,376	-0,024	242	4	1	18	5
-0,907	0,087	184	4	1	19	5
-1,187	1,708	164	4	1	20	5
0,523	5,671	300	4	1	21	5
1,88	4,805	256	5	1	2	1
0,806	3,947	280	5	1	3	1
0,987	3,9	246	5	1	4	1
2,268	6,672	278	5	1	5	1
-2,404	2,375	272	5	1	6	1
2,623	6,042	278	5	1	7	1
2,283	4,896	256	5	1	9	1
4,753	9,759	260	5	1	10	1
2,438	6,13	210	5	1	13	1
8,236	25,525	276	5	1	15	1
0,747	4,566	292	5	1	16	1
-0,706	4,564	250	5	1	17	1
4,134	7,717	272	5	1	18	1
1,192	5,341	288	5	1	19	1
0,546	3,58	284	5	1	20	1
2,458	5,756	190	5	1	21	1
0,619	4,138	248	5	1	2	2
2,141	4,814	282	5	1	3	2

1,483	4,888	266	5	1	4	2
0,512	4,004	286	5	1	5	2
2,317	5,839	282	5	1	6	2
1,58	4,795	278	5	1	7	2
2,461	5,057	284	5	1	9	2
3,073	6,979	254	5	1	10	2
1,545	4,375	188	5	1	13	2
97,235	105,732	276	5	1	15	2
-1,225	4,362	294	5	1	16	2
3,16	8,783	258	5	1	17	2
0,582	3,154	258	5	1	18	2
-0,015	4,637	286	5	1	19	2
1,58	3,4	234	5	1	20	2
3,082	7,058	202	5	1	21	2
0,66	3,556	246	5	1	2	3
0,529	3,882	284	5	1	3	3
0,666	5,077	256	5	1	4	3
3,098	4,576	294	5	1	5	3
-0,572	4,687	280	5	1	6	3
1,8	6,608	290	5	1	7	3
3,764	6,818	198	5	1	9	3
0,396	5,108	266	5	1	10	3
2,059	4,692	184	5	1	13	3
-77,247	-71,067	200	5	1	15	3
0,366	4,514	280	5	1	16	3
1,683	7,991	250	5	1	17	3
1,474	4,394	296	5	1	18	3
1,484	5,986	278	5	1	19	3
3,195	5,722	290	5	1	20	3
3,382	7,174	178	5	1	21	3
-0,254	2,738	268	5	1	2	4
0,3	2,27	266	5	1	3	4
-0,49	2,806	262	5	1	4	4
2,656	5,657	272	5	1	5	4
1,388	5,555	292	5	1	6	4
1,313	7,539	288	5	1	7	4
4,117	6,734	264	5	1	9	4
3,49	8,069	270	5	1	10	4
-1,612	0,739	224	5	1	13	4
-36,41	-33,332	246	5	1	15	4
-3,235	2,039	284	5	1	16	4
1,873	6,293	252	5	1	17	4
1,223	5,279	266	5	1	18	4

-2,644	0,334	270	5	1	19	4
1,199	4,3	298	5	1	20	4
2,852	7,354	194	5	1	21	4
1,326	4,192	250	5	1	2	5
0,61	2,508	264	5	1	3	5
1,171	3,53	246	5	1	4	5
4,687	6,813	274	5	1	5	5
0,874	5,088	286	5	1	6	5
1,336	4,059	282	5	1	7	5
2,131	3,836	244	5	1	9	5
2,028	5,577	264	5	1	10	5
1,055	2,502	208	5	1	13	5
-3,659	-0,187	200	5	1	15	5
-3,192	-0,514	290	5	1	16	5
-1,355	3,055	256	5	1	17	5
0,762	4,23	266	5	1	18	5
0,272	3,387	282	5	1	19	5
0,531	1,89	300	5	1	20	5
2,314	4,795	200	5	1	21	5
0,235	1,451	260	6	1	2	1
0,643	2,857	270	6	1	3	1
-0,245	1,842	278	6	1	4	1
1,757	4,844	298	6	1	5	1
-0,169	3,525	272	6	1	6	1
0,036	2,621	278	6	1	7	1
0,528	2,57	258	6	1	9	1
1,378	4,353	264	6	1	10	1
2,683	5,661	210	6	1	13	1
5,671	9,809	258	6	1	15	1
-1,433	0,562	292	6	1	16	1
-0,44	3,413	248	6	1	17	1
2,224	4,843	240	6	1	18	1
2,463	5,179	288	6	1	19	1
0,068	1,776	232	6	1	20	1
1,789	4,399	188	6	1	21	1
-0,193	2,561	248	6	1	2	2
0,263	1,749	264	6	1	3	2
0,423	2,67	266	6	1	4	2
0,714	2,166	286	6	1	5	2
2,254	5,165	280	6	1	6	2
-1,306	0,549	212	6	1	7	2
0,991	3,12	258	6	1	9	2
0,935	3,191	252	6	1	10	2

1,117	3,174	194	6	1	13	2
-4,898	-3,69	256	6	1	15	2
0,465	3,5	294	6	1	16	2
1,33	5,135	258	6	1	17	2
-1,187	1,587	238	6	1	18	2
-1,375	0,983	216	6	1	19	2
0,493	2,183	230	6	1	20	2
3,118	6,405	202	6	1	21	2
-0,461	1,728	244	6	1	2	3
0,357	2,855	288	6	1	3	3
0,651	3,285	254	6	1	4	3
2,512	3,337	290	6	1	5	3
0,447	4,348	282	6	1	6	3
-0,478	1,983	258	6	1	7	3
2,981	4,475	250	6	1	9	3
-0,667	1,487	234	6	1	10	3
1,798	3,859	182	6	1	13	3
-2,056	-0,929	250	6	1	15	3
0,262	2,536	298	6	1	16	3
0,327	4,353	264	6	1	17	3
-0,103	3,188	232	6	1	18	3
0,619	3,395	260	6	1	19	3
2,034	3,349	254	6	1	20	3
3,191	6,315	210	6	1	21	3
-0,949	0,616	270	6	1	2	4
0,994	2,305	264	6	1	3	4
-0,179	1,802	262	6	1	4	4
1,734	3,136	238	6	1	5	4
2,046	4,983	292	6	1	6	4
-1,494	0,927	288	6	1	7	4
2,091	4,793	268	6	1	9	4
2,399	5,21	252	6	1	10	4
0,128	2,474	190	6	1	13	4
-3,311	-2,352	268	6	1	15	4
-1,871	1,253	282	6	1	16	4
0,65	4,466	252	6	1	17	4
1,046	4,651	264	6	1	18	4
-2,669	-0,361	268	6	1	19	4
0,752	2,427	228	6	1	20	4
3,326	7,182	192	6	1	21	4
0,638	2,151	278	6	1	2	5
0,474	1,672	280	6	1	3	5
0,849	2,392	240	6	1	4	5

0,384	1,862	272	6	1	5	5
0,724	4,542	284	6	1	6	5
-0,582	1,154	264	6	1	7	5
1,748	3,171	244	6	1	9	5
1,198	3,362	244	6	1	10	5
0,575	2,425	224	6	1	13	5
-2,485	-1,94	198	6	1	15	5
-0,943	0,215	252	6	1	16	5
-1,038	2,445	250	6	1	17	5
0,067	1,655	264	6	1	18	5
0,053	2,061	282	6	1	19	5
-0,285	0,701	232	6	1	20	5
2,46	4,487	198	6	1	21	5
-0,744	0,249	286	7	1	2	1
0,798	2,705	272	7	1	3	1
-1,213	0,266	278	7	1	4	1
1,791	5,044	300	7	1	5	1
-0,7	1,546	270	7	1	6	1
-1,015	0,759	276	7	1	7	1
-0,056	1,813	258	7	1	9	1
0,41	1,847	268	7	1	10	1
1,922	4,844	210	7	1	13	1
6,59	10,284	258	7	1	15	1
-1,417	1,067	156	7	1	16	1
0,029	2,981	248	7	1	17	1
1,045	3,781	238	7	1	18	1
2,739	4,948	236	7	1	19	1
-0,128	2,064	224	7	1	20	1
0,955	3,785	174	7	1	21	1
-0,419	1,312	248	7	1	2	2
0,061	1,354	254	7	1	3	2
-0,748	1,153	266	7	1	4	2
1,129	1,948	266	7	1	5	2
1,823	4,605	282	7	1	6	2
-2,114	-0,231	212	7	1	7	2
0,245	2,894	294	7	1	9	2
0,929	1,941	254	7	1	10	2
0,594	2,641	224	7	1	13	2
-5,04	-4,317	216	7	1	15	2
1,163	3,223	242	7	1	16	2
0,378	3,11	256	7	1	17	2
-1,525	1,319	238	7	1	18	2
-1,662	0,146	216	7	1	19	2

-0,319	1,169	216	7	1	20	2
2,03	4,763	154	7	1	21	2
-0,698	0,68	300	7	1	2	3
0,663	2,783	290	7	1	3	3
0,019	1,831	254	7	1	4	3
1,904	2,364	294	7	1	5	3
0,271	4,091	296	7	1	6	3
-0,733	0,665	256	7	1	7	3
2,247	3,609	250	7	1	9	3
-0,866	0,155	234	7	1	10	3
1,17	3,512	230	7	1	13	3
-1,978	-0,83	228	7	1	15	3
0,467	3,483	298	7	1	16	3
-0,5	2,493	264	7	1	17	3
0,433	4,936	242	7	1	18	3
0,098	2,301	244	7	1	19	3
1,542	2,895	222	7	1	20	3
2,847	4,674	210	7	1	21	3
-0,933	1,033	284	7	1	2	4
1,164	2,28	264	7	1	3	4
-0,782	0,724	248	7	1	4	4
0,869	1,828	260	7	1	5	4
1,343	4,036	292	7	1	6	4
-0,761	1,211	286	7	1	7	4
1,963	4,288	268	7	1	9	4
1,831	3,188	252	7	1	10	4
-0,467	1,799	210	7	1	13	4
-3,516	-2,733	200	7	1	15	4
-1,36	1,354	232	7	1	16	4
0,918	4,204	252	7	1	17	4
1,153	3,953	262	7	1	18	4
-3,24	-1,35	284	7	1	19	4
0,036	1,492	226	7	1	20	4
2,073	4,965	160	7	1	21	4
0,204	1,856	280	7	1	2	5
0,722	1,805	282	7	1	3	5
-0,185	1,146	238	7	1	4	5
0,871	1,989	158	7	1	5	5
0,232	3,43	286	7	1	6	5
-0,894	0,549	280	7	1	7	5
1,898	3,379	272	7	1	9	5
0,898	2,338	244	7	1	10	5
0,251	2,322	228	7	1	13	5

-2,075	-1,583	222	7	1	15	5
0,075	1,082	238	7	1	16	5
-0,752	2,002	248	7	1	17	5
-1,079	0,382	244	7	1	18	5
-0,116	1,109	252	7	1	19	5
-0,485	0,415	218	7	1	20	5
2,4	3,747	150	7	1	21	5
-0,755	0,849	286	8	1	2	1
0,702	2,391	270	8	1	3	1
-1,636	-0,713	288	8	1	4	1
1,781	3,82	278	8	1	5	1
-1,529	0,106	292	8	1	6	1
-1,542	-0,597	278	8	1	7	1
-0,248	2,098	292	8	1	9	1
-0,859	3,284	152	8	1	10	1
1,809	5,342	226	8	1	13	1
5,428	8,638	256	8	1	15	1
0,069	0,908	254	8	1	16	1
0,494	2,959	256	8	1	17	1
0,536	4,183	238	8	1	18	1
2,806	4,325	230	8	1	19	1
-0,452	1,941	224	8	1	20	1
0,149	3,676	154	8	1	21	1
-0,672	1,281	278	8	1	2	2
-0,13	1,238	252	8	1	3	2
-1,763	-0,278	266	8	1	4	2
0,784	2,719	168	8	1	5	2
1,484	4,388	286	8	1	6	2
-1,81	-0,743	212	8	1	7	2
-0,353	3,793	294	8	1	9	2
0,683	2,317	298	8	1	10	2
0,815	3,446	224	8	1	13	2
-4,488	-2,664	218	8	1	15	2
1,448	3,313	242	8	1	16	2
-0,693	1,5	244	8	1	17	2
-2,048	0,791	238	8	1	18	2
-1,176	0,17	166	8	1	19	2
-1,278	2,154	152	8	1	20	2
0,719	4,801	152	8	1	21	2
-0,65	1,313	298	8	1	2	3
0,73	2,681	292	8	1	3	3
-0,472	0,692	252	8	1	4	3
1,205	2,093	294	8	1	5	3

0,309	3,923	296	8	1	6	3
-1,142	-0,287	224	8	1	7	3
1,648	3,659	150	8	1	9	3
-0,898	0,171	296	8	1	10	3
0,594	3,957	230	8	1	13	3
-2,284	-0,614	154	8	1	15	3
0,729	3,619	298	8	1	16	3
-1,124	1,185	250	8	1	17	3
0,007	4,844	242	8	1	18	3
0,01	1,629	176	8	1	19	3
0,413	2,983	164	8	1	20	3
1,845	5,62	156	8	1	21	3
-0,453	1,924	284	8	1	2	4
1,304	2,355	264	8	1	3	4
-1,276	0,097	246	8	1	4	4
-0,262	2,978	150	8	1	5	4
0,589	3,579	294	8	1	6	4
-0,955	0,451	288	8	1	7	4
2,029	4,959	298	8	1	9	4
1,16	0,943	250	8	1	10	4
-1,359	0,976	222	8	1	13	4
-3,812	-3,059	172	8	1	15	4
-1,229	1,316	232	8	1	16	4
0,793	3,454	254	8	1	17	4
0,813	3,607	248	8	1	18	4
-3,069	-2,278	282	8	1	19	4
-0,901	0,077	222	8	1	20	4
0,856	4,827	156	8	1	21	4
0,099	2,182	296	8	1	2	5
0,776	1,913	264	8	1	3	5
-0,819	0,301	238	8	1	4	5
0,627	3,593	154	8	1	5	5
-0,258	2,262	284	8	1	6	5
-1,069	0,195	280	8	1	7	5
2,184	4,174	288	8	1	9	5
0,835	1,647	292	8	1	10	5
-0,258	2,696	230	8	1	13	5
-2,031	-0,97	152	8	1	15	5
1,027	2,326	158	8	1	16	5
-0,606	1,553	240	8	1	17	5
-1,582	0,434	244	8	1	18	5
-0,309	0,322	244	8	1	19	5
-0,754	0,122	208	8	1	20	5

1,769	3,437	300	8	1	21	5
1,939	5,194	238	9	1	2	1
0,336	3,305	280	9	1	3	1
-0,385	2,941	258	9	1	4	1
3,67	7,015	298	9	1	5	1
-1,234	4,195	272	9	1	6	1
1,212	4,394	278	9	1	7	1
1,029	3,578	256	9	1	9	1
2,176	7,318	260	9	1	10	1
3,73	6,571	178	9	1	13	1
5,411	9,314	256	9	1	15	1
1,929	5,48	294	9	1	16	1
-0,056	4,615	264	9	1	17	1
3,378	8,081	288	9	1	18	1
0,056	5,427	288	9	1	19	1
1,345	5,507	282	9	1	20	1
-0,484	1,389	272	9	1	21	1
-0,464	2,01	248	9	1	2	2
1,849	4,383	278	9	1	3	2
0,728	4,834	248	9	1	4	2
0,433	2,448	242	9	1	5	2
-0,528	2,126	266	9	1	6	2
-0,604	2,504	280	9	1	7	2
-0,091	2,741	290	9	1	9	2
-0,652	3,487	274	9	1	10	2
3,154	5,282	226	9	1	13	2
-6,676	-4,157	272	9	1	15	2
0,015	3,565	296	9	1	16	2
2,262	7,58	256	9	1	17	2
0,442	5,731	290	9	1	18	2
-1,135	3,665	284	9	1	19	2
0,764	3,117	286	9	1	20	2
1,057	3,713	202	9	1	21	2
-0,022	2,833	276	9	1	2	3
0,076	2,294	284	9	1	3	3
0,048	2,885	238	9	1	4	3
4,232	5,981	292	9	1	5	3
-2,788	0,888	278	9	1	6	3
0,194	3,948	292	9	1	7	3
3,276	5,743	264	9	1	9	3
0,053	6,033	264	9	1	10	3
2,471	4,596	234	9	1	13	3
-4,433	-2,358	248	9	1	15	3

-1,386	1,466	282	9	1	16	3
1,671	7,309	264	9	1	17	3
-0,339	4,619	298	9	1	18	3
1,333	7,129	294	9	1	19	3
2,409	6,432	290	9	1	20	3
1,62	6,368	192	9	1	21	3
0,027	3,03	270	9	1	2	4
-0,352	1,716	282	9	1	3	4
-0,678	2,601	264	9	1	4	4
2,857	4,654	290	9	1	5	4
-1,702	3,206	286	9	1	6	4
-2,299	1,057	288	9	1	7	4
2,293	3,829	294	9	1	9	4
0,889	6,244	256	9	1	10	4
-0,287	2,064	238	9	1	13	4
-4,58	-1,986	280	9	1	15	4
-1,275	4,166	284	9	1	16	4
0,51	5,092	256	9	1	17	4
0,907	5,26	300	9	1	18	4
-1,299	2,598	236	9	1	19	4
0,594	4,495	298	9	1	20	4
1,759	6,298	194	9	1	21	4
0,374	3,195	250	9	1	2	5
0,374	2,6	280	9	1	3	5
0,454	3,551	246	9	1	4	5
0,702	2,425	270	9	1	5	5
-0,536	2,704	276	9	1	6	5
-0,401	1,728	294	9	1	7	5
1,612	3,259	272	9	1	9	5
0,025	5,014	264	9	1	10	5
0,972	2,468	226	9	1	13	5
-4,256	-2,58	266	9	1	15	5
-2,155	0,875	292	9	1	16	5
-1,105	3,751	258	9	1	17	5
0,384	5,22	298	9	1	18	5
0,844	3,695	284	9	1	19	5
-0,287	1,814	298	9	1	20	5
0,872	3,753	184	9	1	21	5
0,809	3,109	236	10	1	2	1
0,419	2,061	266	10	1	3	1
-1,179	0,841	256	10	1	4	1
1,558	3,605	300	10	1	5	1
-0,473	3,248	272	10	1	6	1

-1,436	0,52	212	10	1	7	1
-1,125	0,718	276	10	1	9	1
-1,222	2,489	246	10	1	10	1
1,645	3,216	226	10	1	13	1
6,779	10,549	296	10	1	15	1
0,328	2,731	294	10	1	16	1
0,281	2,628	246	10	1	17	1
2,609	5,525	286	10	1	18	1
1,92	3,984	288	10	1	19	1
0,57	2,846	298	10	1	20	1
-2,675	-0,451	172	10	1	21	1
-1,942	-0,271	248	10	1	2	2
0,241	1,888	254	10	1	3	2
-0,47	1,95	248	10	1	4	2
0,401	1,899	280	10	1	5	2
0,434	2,993	246	10	1	6	2
-1,643	0,271	164	10	1	7	2
-3,083	-0,795	294	10	1	9	2
-2,123	0,93	246	10	1	10	2
1,335	3,803	226	10	1	13	2
-6,405	-5,16	276	10	1	15	2
0,362	2,38	296	10	1	16	2
1,303	4,859	256	10	1	17	2
-1,72	-1,32	220	10	1	18	2
-1,103	0,199	266	10	1	19	2
-0,524	0,739	236	10	1	20	2
-1,523	0,922	204	10	1	21	2
-1,368	1,194	276	10	1	2	3
0,164	1,639	238	10	1	3	3
-0,047	1,716	236	10	1	4	3
2,196	3,647	256	10	1	5	3
-0,403	1,877	278	10	1	6	3
-0,961	0,942	268	10	1	7	3
0,852	2,331	252	10	1	9	3
-1,48	1,738	262	10	1	10	3
0,051	3,802	234	10	1	13	3
-2,892	-1,339	224	10	1	15	3
-1,834	-0,002	284	10	1	16	3
0,609	4,08	250	10	1	17	3
-1,554	-0,111	212	10	1	18	3
-1,662	0,249	276	10	1	19	3
1,167	2,748	290	10	1	20	3
-0,607	3,272	178	10	1	21	3

-1,129	1,028	266	10	1	2	4
0,157	1,85	276	10	1	3	4
-0,124	1,778	230	10	1	4	4
0,998	2,127	246	10	1	5	4
-0,112	3,086	284	10	1	6	4
-2,004	-0,359	290	10	1	7	4
0,23	1,296	298	10	1	9	4
-0,911	3,075	256	10	1	10	4
-2,115	0,821	222	10	1	13	4
-4,283	-1,941	286	10	1	15	4
-0,767	2,287	286	10	1	16	4
-0,074	2,799	266	10	1	17	4
0,836	1,554	252	10	1	18	4
-2,146	-0,91	268	10	1	19	4
-0,573	2,019	294	10	1	20	4
-0,271	3,373	194	10	1	21	4
-0,311	2,118	252	10	1	2	5
0,352	1,934	296	10	1	3	5
0,457	2,271	246	10	1	4	5
0,131	1,259	248	10	1	5	5
-0,337	2,186	274	10	1	6	5
-1,283	-0,774	166	10	1	7	5
0,167	1,683	290	10	1	9	5
-0,459	2,739	264	10	1	10	5
-0,736	0,685	228	10	1	13	5
-3,715	-2,716	270	10	1	15	5
-0,477	1,426	284	10	1	16	5
-1,163	1,878	252	10	1	17	5
-0,729	3,767	300	10	1	18	5
0,017	1,388	280	10	1	19	5
-0,646	0,688	300	10	1	20	5
0,449	3,328	186	10	1	21	5
0,435	2,527	238	11	1	2	1
0,354	1,711	266	11	1	3	1
-1,605	-0,008	274	11	1	4	1
1,089	3,34	300	11	1	5	1
-0,803	2,116	270	11	1	6	1
-2,216	-0,948	212	11	1	7	1
-1,35	1,237	292	11	1	9	1
-2,579	-0,103	294	11	1	10	1
1,743	3,242	224	11	1	13	1
5,828	9,161	298	11	1	15	1
-1,341	-0,092	294	11	1	16	1

0,13	1,718	234	11	1	17	1
1,459	3,055	154	11	1	18	1
0,724	2,654	170	11	1	19	1
0,421	2,1	298	11	1	20	1
-3,256	0,086	152	11	1	21	1
-2,493	-1,376	248	11	1	2	2
-0,125	1,327	254	11	1	3	2
-1,134	0,592	248	11	1	4	2
0,297	1,245	268	11	1	5	2
0,117	3,082	246	11	1	6	2
-2,251	-0,322	164	11	1	7	2
-2,665	0,229	294	11	1	9	2
-2,093	-0,006	262	11	1	10	2
1,108	3,19	226	11	1	13	2
-6,353	-5,519	278	11	1	15	2
0,491	2,245	296	11	1	16	2
0,556	3,011	256	11	1	17	2
-3,274	-1,741	154	11	1	18	2
-2,05	0,711	150	11	1	19	2
-1,66	-0,963	240	11	1	20	2
-2,174	1,25	154	11	1	21	2
-1,834	0,365	278	11	1	2	3
0,477	1,943	238	11	1	3	3
-0,252	1,364	220	11	1	4	3
1,058	1,866	162	11	1	5	3
0,041	1,782	296	11	1	6	3
-1,749	1,149	156	11	1	7	3
0,963	2,847	252	11	1	9	3
-2,164	-0,28	264	11	1	10	3
-0,036	4,342	236	11	1	13	3
-3,333	-1,746	226	11	1	15	3
-2,135	0,221	298	11	1	16	3
-0,582	1,745	250	11	1	17	3
-2,134	-0,711	214	11	1	18	3
-0,867	2,139	158	11	1	19	3
0,008	0,572	152	11	1	20	3
-1,499	2,732	164	11	1	21	3
-1,152	0,734	268	11	1	2	4
0,474	1,526	272	11	1	3	4
-0,375	1,184	230	11	1	4	4
-0,435	0,328	296	11	1	5	4
-0,425	2,368	292	11	1	6	4
-2,373	-1,215	292	11	1	7	4

0,303	2,841	300	11	1	9	4
-1,406	1,293	252	11	1	10	4
-1,553	0,945	222	11	1	13	4
-5,227	-3,278	300	11	1	15	4
-1,364	1,146	298	11	1	16	4
-0,263	1,462	268	11	1	17	4
-0,189	2,168	164	11	1	18	4
-2,511	-0,797	186	11	1	19	4
-1,022	1,132	294	11	1	20	4
-1,546	1,253	178	11	1	21	4
-0,412	1,356	252	11	1	2	5
0,39	1,806	298	11	1	3	5
0,329	1,587	276	11	1	4	5
0,277	1,236	278	11	1	5	5
-0,753	1,334	274	11	1	6	5
-2,133	-1,919	230	11	1	7	5
0,488	2,488	290	11	1	9	5
-0,26	2,372	264	11	1	10	5
-0,496	1,425	238	11	1	13	5
-2,663	-1,83	270	11	1	15	5
-0,428	0,588	286	11	1	16	5
-1,109	1,265	248	11	1	17	5
-1,569	-1,055	152	11	1	18	5
-0,731	0,595	150	11	1	19	5
-0,958	-0,785	270	11	1	20	5
0,491	3,012	166	11	1	21	5
0,055	1,719	238	12	1	2	1
0,13	1,213	266	12	1	3	1
-1,423	0,016	274	12	1	4	1
1,02	2,649	276	12	1	5	1
-1,298	1,311	256	12	1	6	1
-2,94	-2,197	280	12	1	7	1
-1,652	1,345	294	12	1	9	1
-3,775	-2,954	246	12	1	10	1
1,891	3,765	224	12	1	13	1
5,555	8,763	300	12	1	15	1
-1,806	-0,856	256	12	1	16	1
0,11	1,502	232	12	1	17	1
0,55	3,729	152	12	1	18	1
1,231	4,2	154	12	1	19	1
0,179	1,264	216	12	1	20	1
-3,54	2,064	152	12	1	21	1
-2,946	-2,243	300	12	1	2	2

-0,447	0,893	252	12	1	3	2
-1,454	0,33	282	12	1	4	2
0,353	1,133	268	12	1	5	2
-0,121	2,906	244	12	1	6	2
-2,692	-0,824	158	12	1	7	2
-2,336	1,37	294	12	1	9	2
-2,096	0,086	262	12	1	10	2
0,668	2,879	246	12	1	13	2
-5,335	-4,708	284	12	1	15	2
0,98	2,664	154	12	1	16	2
-0,037	1,591	256	12	1	17	2
-4,032	-1,231	152	12	1	18	2
-2,964	1,427	150	12	1	19	2
-2,482	-2,79	242	12	1	20	2
-2,525	2,106	154	12	1	21	2
-2,019	0,008	278	12	1	2	3
0,467	1,84	238	12	1	3	3
-0,134	1,369	218	12	1	4	3
-0,177	1,771	168	12	1	5	3
-0,009	2,011	242	12	1	6	3
-2,552	1,47	150	12	1	7	3
1,482	3,712	152	12	1	9	3
-2,87	0,496	300	12	1	10	3
0,613	5,162	244	12	1	13	3
-3,556	-2,084	228	12	1	15	3
-1,626	1,099	296	12	1	16	3
-1,508	0,062	282	12	1	17	3
-1,925	0,045	158	12	1	18	3
-1,626	3,5	158	12	1	19	3
-0,873	1,542	166	12	1	20	3
-2,422	3,42	160	12	1	21	3
-0,976	0,984	290	12	1	2	4
0,582	1,224	236	12	1	3	4
-0,135	1,449	282	12	1	4	4
-1,566	-0,073	296	12	1	5	4
-0,889	2,119	294	12	1	6	4
-3,146	-2,708	264	12	1	7	4
1,349	5,05	300	12	1	9	4
-2,075	-0,503	250	12	1	10	4
-2,063	0,143	236	12	1	13	4
-5,065	-3,635	298	12	1	15	4
-1,888	0,125	298	12	1	16	4
-0,192	1,398	282	12	1	17	4

-0,795	3,097	162	12	1	18	4
-2,774	0,576	152	12	1	19	4
-1,63	0,373	164	12	1	20	4
-2,546	1,496	150	12	1	21	4
-0,422	1,477	294	12	1	2	5
0,448	1,756	300	12	1	3	5
0,502	1,682	274	12	1	4	5
0,15	1,697	300	12	1	5	5
-1,236	0,554	276	12	1	6	5
-2,98	-2,533	234	12	1	7	5
0,879	3,415	290	12	1	9	5
-0,46	3,225	292	12	1	10	5
-0,154	2,657	254	12	1	13	5
-2,322	-1,606	270	12	1	15	5
-0,013	0,94	236	12	1	16	5
-0,666	1,257	246	12	1	17	5
-1,372	0,107	152	12	1	18	5
-0,902	0,431	182	12	1	19	5
-1,317	1,473	150	12	1	20	5
0,381	4,104	156	12	1	21	5
-0,062	2,55	340	1	2	2	1
0,887	2,998	448	1	2	3	1
-1,46	0,273	312	1	2	4	1
5,906	7,882	462	1	2	5	1
-2,314	1,593	436	1	2	6	1
-4,224	-0,874	428	1	2	7	1
0,396	3,745	456	1	2	9	1
-1,614	0,516	442	1	2	10	1
1,238	3,552	470	1	2	13	1
10,147	12,785	458	1	2	15	1
-1,501	4,548	308	1	2	16	1
-4,16	-1,361	316	1	2	17	1
0,275	4,28	304	1	2	18	1
2,947	6,983	306	1	2	19	1
3,457	6,097	448	1	2	20	1
-3,627	1,37	388	1	2	21	1
-2,068	0,406	330	1	2	2	2
0,047	1,15	400	1	2	3	2
-1,542	1,855	300	1	2	4	2
1,542	5,349	336	1	2	5	2
1,386	3,017	314	1	2	6	2
-2,602	2,584	460	1	2	7	2
1,524	3,276	404	1	2	9	2

-5,022	0,297	490	1	2	10	2
-3,653	-0,719	414	1	2	13	2
-8,716	-4,164	306	1	2	15	2
3,813	6,632	312	1	2	16	2
-4,208	2,143	306	1	2	17	2
-5,551	3,778	304	1	2	18	2
-3,089	1,773	302	1	2	19	2
4,622	7,771	432	1	2	20	2
-0,541	6,721	370	1	2	21	2
-3,154	-1,345	426	1	2	2	3
0,244	1,642	348	1	2	3	3
-1,158	2,709	456	1	2	4	3
3,631	6,801	320	1	2	5	3
-3,121	1	426	1	2	6	3
-1,705	3,384	306	1	2	7	3
2,732	6,796	312	1	2	9	3
-6,763	-3,102	316	1	2	10	3
-2,846	-1,974	468	1	2	13	3
-6,989	-1,835	316	1	2	15	3
3,398	6,798	464	1	2	16	3
-5,049	0,729	316	1	2	17	3
-1,604	4,096	312	1	2	18	3
-3,218	2,126	312	1	2	19	3
4,041	5,81	304	1	2	20	3
-1,346	2,585	376	1	2	21	3
-0,713	1,766	420	1	2	2	4
-1,167	2,487	300	1	2	3	4
-1,223	1,619	464	1	2	4	4
2,005	4,022	304	1	2	5	4
-0,477	2,813	320	1	2	6	4
-3,797	-0,122	454	1	2	7	4
2,482	4,038	462	1	2	9	4
-3,023	2,697	302	1	2	10	4
-5,329	-3,304	476	1	2	13	4
-7,944	-3,441	314	1	2	15	4
-3,268	-0,647	452	1	2	16	4
-6,791	-6,251	450	1	2	17	4
-2,017	1,622	316	1	2	18	4
-5,711	-2,287	322	1	2	19	4
4,41	6,87	464	1	2	20	4
-1,339	4,355	376	1	2	21	4
-1,487	2,187	300	1	2	2	5
-0,663	0,115	346	1	2	3	5

-0,66	1,266	312	1	2	4	5
1,057	3,666	308	1	2	5	5
-0,28	0,388	368	1	2	6	5
4,098	6,075	432	1	2	7	5
1,952	3,128	378	1	2	9	5
-3,744	-3,055	470	1	2	10	5
-2,256	-1,007	468	1	2	13	5
-5,827	-3,72	334	1	2	15	5
-2,313	-0,278	306	1	2	16	5
-7,23	-5,619	458	1	2	17	5
-1,759	5,787	300	1	2	18	5
-1,937	-0,838	354	1	2	19	5
1,037	1,848	464	1	2	20	5
1,377	5,471	368	1	2	21	5
-2,524	0,264	340	2	2	2	1
-0,804	0,28	448	2	2	3	1
-2,619	-0,931	458	2	2	4	1
4,22	5,578	462	2	2	5	1
-2,476	1,022	436	2	2	6	1
-0,957	1,946	416	2	2	7	1
-0,775	1,137	460	2	2	9	1
-3,215	-0,686	478	2	2	10	1
-0,281	2,018	440	2	2	13	1
8,573	10,838	458	2	2	15	1
-2,68	-1,437	390	2	2	16	1
-4,72	-2,87	454	2	2	17	1
-0,381	2,45	422	2	2	18	1
3,12	4,467	324	2	2	19	1
0,069	2,041	446	2	2	20	1
-3,629	1,537	388	2	2	21	1
-4,495	-2,52	330	2	2	2	2
-0,994	-0,029	378	2	2	3	2
-2,75	-0,012	312	2	2	4	2
2,212	5,141	334	2	2	5	2
-0,883	1,844	310	2	2	6	2
-1,902	0,018	450	2	2	7	2
-1,786	-0,281	406	2	2	9	2
-6,892	-2,072	488	2	2	10	2
-3,739	-0,575	300	2	2	13	2
-8,609	-5,582	308	2	2	15	2
1,799	4,37	476	2	2	16	2
-6,644	-6,201	460	2	2	17	2
-7,123	-0,607	304	2	2	18	2

-3,855	-0,555	338	2	2	19	2
1,22	2,642	438	2	2	20	2
-1,505	4,98	370	2	2	21	2
-3,576	0,198	300	2	2	2	3
-0,083	1,27	346	2	2	3	3
-1,954	1,406	300	2	2	4	3
2,052	4,268	308	2	2	5	3
-2,574	0,251	424	2	2	6	3
-2,323	-0,952	424	2	2	7	3
1,226	3,902	352	2	2	9	3
-8,141	-5,364	484	2	2	10	3
-5,216	-3,22	480	2	2	13	3
-7,103	-2,48	312	2	2	15	3
0,652	3,601	462	2	2	16	3
-7,163	-2,242	314	2	2	17	3
-1,282	3,594	316	2	2	18	3
-2,872	-0,079	344	2	2	19	3
1,667	3,134	306	2	2	20	3
-0,548	3,43	394	2	2	21	3
-2,438	-0,94	310	2	2	2	4
-1,737	1,727	300	2	2	3	4
-2,15	0,408	308	2	2	4	4
1,275	3,377	438	2	2	5	4
-3,001	0,36	320	2	2	6	4
-1,353	0,043	410	2	2	7	4
1,698	3,253	398	2	2	9	4
-5,153	-3,785	440	2	2	10	4
-6,724	-3,987	302	2	2	13	4
-9,117	-5,404	314	2	2	15	4
-2,212	-0,782	466	2	2	16	4
-8,359	-8,112	448	2	2	17	4
-2,008	3,211	300	2	2	18	4
-4,959	-2,56	324	2	2	19	4
1,505	3,34	462	2	2	20	4
0,221	6,644	376	2	2	21	4
-2,624	-1,259	434	2	2	2	5
-0,732	-0,093	346	2	2	3	5
-1,069	0,832	308	2	2	4	5
0,524	2,541	308	2	2	5	5
-1,656	0,43	316	2	2	6	5
-1,832	-0,416	418	2	2	7	5
2,229	3,404	380	2	2	9	5
-4,439	-2,894	492	2	2	10	5

-3,959	-1,817	312	2	2	13	5
-5,774	-3,415	324	2	2	15	5
-2,004	-0,309	324	2	2	16	5
-7,345	-6,032	440	2	2	17	5
-1,768	3,764	300	2	2	18	5
-0,907	1,24	316	2	2	19	5
-0,321	0,172	462	2	2	20	5
3,219	7,339	368	2	2	21	5
-2,815	-0,517	338	3	2	2	1
-1,365	-0,745	414	3	2	3	1
-2,84	-1,276	474	3	2	4	1
4,442	5,844	492	3	2	5	1
-3,319	-0,002	434	3	2	6	1
-0,421	2,06	420	3	2	7	1
-0,556	1,288	494	3	2	9	1
-2,275	1,093	482	3	2	10	1
0,371	1,999	440	3	2	13	1
14,919	17,23	454	3	2	15	1
-2,185	-0,412	424	3	2	16	1
-4,827	-3,745	452	3	2	17	1
-0,924	3,033	420	3	2	18	1
3,089	4,535	474	3	2	19	1
-1,493	0,067	314	3	2	20	1
-1,145	4,012	388	3	2	21	1
-4,533	-2,962	426	3	2	2	2
-0,798	0,2	378	3	2	3	2
-2,941	-1,033	312	3	2	4	2
1,854	4,147	334	3	2	5	2
-1,183	2,369	310	3	2	6	2
-1,349	-0,157	312	3	2	7	2
-0,667	1,052	408	3	2	9	2
-4,932	0,041	490	3	2	10	2
-3,799	-1,299	302	3	2	13	2
-8,628	-5,301	314	3	2	15	2
0,768	2,998	488	3	2	16	2
-6,728	-6,703	458	3	2	17	2
-7,389	-2,136	304	3	2	18	2
-3,365	-0,244	340	3	2	19	2
-0,999	-0,071	304	3	2	20	2
0,017	6,032	370	3	2	21	2
-4,035	-0,127	304	3	2	2	3
0,165	1,527	338	3	2	3	3
-1,882	-0,135	452	3	2	4	3

1,107	2,996	308	3	2	5	3
-2,931	1,657	306	3	2	6	3
-1,855	1,447	310	3	2	7	3
1,264	4,006	352	3	2	9	3
-6,339	-3,209	330	3	2	10	3
-5,302	-2,82	482	3	2	13	3
-1,553	1,847	312	3	2	15	3
-0,131	2,375	310	3	2	16	3
-7,996	-3,196	314	3	2	17	3
-1,258	3,127	316	3	2	18	3
-2,684	-0,625	330	3	2	19	3
0,202	1,414	304	3	2	20	3
1,331	6,464	408	3	2	21	3
-2,435	-0,431	310	3	2	2	4
-1,661	0,132	332	3	2	3	4
-2,33	0,2	306	3	2	4	4
0,378	2,221	438	3	2	5	4
-1,81	0,705	320	3	2	6	4
-1,028	0,497	408	3	2	7	4
2,25	3,633	400	3	2	9	4
-3,574	1,183	302	3	2	10	4
-6,97	-4,198	316	3	2	13	4
8,181	8,773	300	3	2	15	4
-3,333	-0,927	312	3	2	16	4
-8,638	-8,444	498	3	2	17	4
-1,537	2,215	316	3	2	18	4
-4,831	-2,386	326	3	2	19	4
0,1	2,088	460	3	2	20	4
1,697	7,913	378	3	2	21	4
-2,522	-1,749	434	3	2	2	5
0,957	1,6	346	3	2	3	5
-1,12	0,457	308	3	2	4	5
0,63	2,761	300	3	2	5	5
-1,963	0,496	316	3	2	6	5
-1,4	-0,337	306	3	2	7	5
2,96	4,053	380	3	2	9	5
-3,063	-0,655	492	3	2	10	5
-3,982	-1,981	314	3	2	13	5
3,635	4,69	326	3	2	15	5
-2,446	-0,75	326	3	2	16	5
-7,147	-5,92	422	3	2	17	5
-2,729	1,449	302	3	2	18	5
-0,41	1,412	324	3	2	19	5

-1,353	0,139	312	3	2	20	5
4,807	9,109	384	3	2	21	5
-1,783	0,493	352	4	2	2	1
-1,646	0,716	302	4	2	3	1
-0,839	0,752	492	4	2	4	1
4,617	6,734	494	4	2	5	1
-2,484	-1,619	434	4	2	6	1
-0,292	1,707	338	4	2	7	1
1,344	3,831	494	4	2	9	1
1,947	6,597	492	4	2	10	1
0,306	3,038	492	4	2	13	1
9,129	12,701	454	4	2	15	1
-0,132	2,177	426	4	2	16	1
-2,167	-0,712	334	4	2	17	1
0,39	4,454	418	4	2	18	1
3,33	7,131	488	4	2	19	1
-1,681	-0,273	318	4	2	20	1
1,727	4,585	404	4	2	21	1
-2,667	-1,79	478	4	2	2	2
-0,683	0,958	316	4	2	3	2
-1,898	-0,161	482	4	2	4	2
2,769	3,944	416	4	2	5	2
-0,379	0,27	354	4	2	6	2
-0,578	1,974	318	4	2	7	2
1,477	2,588	408	4	2	9	2
1,177	5,366	490	4	2	10	2
-1,595	2,418	498	4	2	13	2
-6,882	-4,233	316	4	2	15	2
1,855	6,726	494	4	2	16	2
-4,745	-2,923	338	4	2	17	2
-5,972	-2,974	408	4	2	18	2
-3,818	-0,558	346	4	2	19	2
-2,252	-0,006	350	4	2	20	2
1,778	4,45	356	4	2	21	2
-3,163	1,05	300	4	2	2	3
-0,713	0,84	332	4	2	3	3
-1,25	0,581	482	4	2	4	3
0,474	1,756	410	4	2	5	3
-0,64	0,06	372	4	2	6	3
-1,168	2,446	318	4	2	7	3
3,557	5,747	352	4	2	9	3
0,007	4,292	486	4	2	10	3
-2,098	2,623	496	4	2	13	3

-3,928	-1,154	312	4	2	15	3
1,761	6,489	496	4	2	16	3
-6,255	-3,004	300	4	2	17	3
0,983	4,347	414	4	2	18	3
-1,247	2,497	494	4	2	19	3
-1,26	1,659	368	4	2	20	3
2,406	5,489	406	4	2	21	3
-1,683	0,918	308	4	2	2	4
-1,903	1,015	306	4	2	3	4
-1,056	-0,034	480	4	2	4	4
0,323	1,848	448	4	2	5	4
0,551	1,831	320	4	2	6	4
-0,368	2,271	322	4	2	7	4
4,759	6,482	484	4	2	9	4
1,341	5,588	494	4	2	10	4
-4,272	0,37	498	4	2	13	4
-6,464	-4,594	344	4	2	15	4
-2,462	-1,367	466	4	2	16	4
-5,749	-3,435	350	4	2	17	4
-0,005	2,926	424	4	2	18	4
-4,009	-1,214	320	4	2	19	4
-2,193	-0,724	360	4	2	20	4
1,801	3,768	362	4	2	21	4
-1,486	-0,737	358	4	2	2	5
-0,24	1,794	300	4	2	3	5
-0,61	0,456	476	4	2	4	5
0,751	1,923	302	4	2	5	5
-1,279	1,234	318	4	2	6	5
-1,36	0,152	324	4	2	7	5
4,153	6,281	486	4	2	9	5
1,048	4,124	492	4	2	10	5
-2,445	0,308	492	4	2	13	5
-4,297	-2,586	336	4	2	15	5
0,526	2,191	410	4	2	16	5
-4,29	-3,244	410	4	2	17	5
-1,176	1,786	410	4	2	18	5
-0,836	0,752	352	4	2	19	5
-2,48	-0,966	348	4	2	20	5
4,073	6,071	368	4	2	21	5
0,06	2,089	424	5	2	2	1
0,598	2,698	448	5	2	3	1
-0,821	1,768	312	5	2	4	1
4,656	6,62	312	5	2	5	1

-3,273	-1,27	438	5	2	6	1
2,848	5,398	460	5	2	7	1
-1,292	1,91	422	5	2	9	1
-0,725	0,63	442	5	2	10	1
0,456	2,422	462	5	2	13	1
35,198	41,676	458	5	2	15	1
-0,212	3,114	308	5	2	16	1
-5,285	-2,438	454	5	2	17	1
1,791	5,546	422	5	2	18	1
-1,208	3,974	306	5	2	19	1
2,588	4,314	450	5	2	20	1
-1,47	2,181	372	5	2	21	1
-1,211	0,631	428	5	2	2	2
1,346	2,438	400	5	2	3	2
-0,542	3,454	300	5	2	4	2
0,981	5,019	336	5	2	5	2
3,39	5,381	316	5	2	6	2
1,574	4,548	462	5	2	7	2
-0,188	1,686	438	5	2	9	2
-1,935	-0,017	476	5	2	10	2
-1,868	0,349	308	5	2	13	2
120,288	157,934	490	5	2	15	2
1,252	5,379	312	5	2	16	2
-2,763	1,668	306	5	2	17	2
-5,508	0,909	304	5	2	18	2
-1,428	4,4	302	5	2	19	2
3,783	6,257	434	5	2	20	2
-1,515	2,838	386	5	2	21	2
-1,467	0,203	440	5	2	2	3
-0,314	0,649	450	5	2	3	3
-1,497	1,464	456	5	2	4	3
1,48	4,239	310	5	2	5	3
-0,947	1,869	412	5	2	6	3
0,746	5,374	306	5	2	7	3
0,84	4,734	314	5	2	9	3
-4,905	-1,601	314	5	2	10	3
-2,149	2,182	302	5	2	13	3
-135,212	-104,227	312	5	2	15	3
2,55	5,907	464	5	2	16	3
-3,664	1,269	300	5	2	17	3
-1,552	2,404	312	5	2	18	3
-0,009	4,703	312	5	2	19	3
3,412	5,45	306	5	2	20	3

-2,309	1,89	476	5	2	21	3
-1,583	1,097	420	5	2	2	4
-1,062	1,453	300	5	2	3	4
-1,167	1,183	330	5	2	4	4
2,04	5,237	320	5	2	5	4
0,718	3,619	320	5	2	6	4
2,186	6,561	304	5	2	7	4
2,434	4,505	444	5	2	9	4
-0,877	0,847	470	5	2	10	4
-4,657	-2,941	332	5	2	13	4
-60,042	-48,163	314	5	2	15	4
-4,056	-1,209	316	5	2	16	4
-4,66	-3,848	420	5	2	17	4
-2,43	0,016	382	5	2	18	4
-6,618	-1,925	302	5	2	19	4
1,317	3,495	464	5	2	20	4
-2,3	1,986	394	5	2	21	4
-0,52	2,348	300	5	2	2	5
-0,837	-0,342	348	5	2	3	5
-0,115	1,906	462	5	2	4	5
3,188	6,307	308	5	2	5	5
0,72	2,127	320	5	2	6	5
2,223	3,527	446	5	2	7	5
-0,224	0,755	312	5	2	9	5
-1,644	-0,568	462	5	2	10	5
-2,313	-0,992	458	5	2	13	5
-18,802	-14,423	336	5	2	15	5
-2,965	-1,152	306	5	2	16	5
-6,311	-1,847	306	5	2	17	5
-2,135	0,223	330	5	2	18	5
-0,632	1,495	316	5	2	19	5
1,319	2,284	402	5	2	20	5
-0,135	2,401	368	5	2	21	5
-1,74	-0,213	340	6	2	2	1
-0,341	1,388	448	6	2	3	1
-0,868	0,309	310	6	2	4	1
3,631	5,09	310	6	2	5	1
-0,336	1,392	436	6	2	6	1
-0,151	2,363	462	6	2	7	1
-0,586	1,94	458	6	2	9	1
-0,879	1,144	462	6	2	10	1
1,627	4,274	442	6	2	13	1
7,718	9,981	460	6	2	15	1

-1,366	0,87	376	6	2	16	1
-4,134	-1,994	458	6	2	17	1
1,644	5,685	438	6	2	18	1
1,619	4,021	304	6	2	19	1
0,47	1,797	446	6	2	20	1
-0,226	3,068	388	6	2	21	1
-1,484	-0,033	300	6	2	2	2
-0,311	0,244	454	6	2	3	2
-0,612	1,069	312	6	2	4	2
1,201	3,164	336	6	2	5	2
2,328	4,605	300	6	2	6	2
-1,963	0,668	310	6	2	7	2
-0,19	0,921	460	6	2	9	2
-2,684	0,571	488	6	2	10	2
-0,961	0,722	304	6	2	13	2
-7,339	-6,3	360	6	2	15	2
1,994	4,074	312	6	2	16	2
-2,885	-2,088	460	6	2	17	2
-5,945	-2,608	306	6	2	18	2
-3,057	0,981	318	6	2	19	2
0,799	2,569	450	6	2	20	2
-0,312	3,195	402	6	2	21	2
-1,838	0,559	306	6	2	2	3
-0,694	-0,116	414	6	2	3	3
-1,079	0,212	456	6	2	4	3
1,662	2,883	310	6	2	5	3
-0,004	1,53	426	6	2	6	3
-1,224	1,304	306	6	2	7	3
0,868	3,645	300	6	2	9	3
-4,837	-2,512	500	6	2	10	3
-1,108	1,035	452	6	2	13	3
-5,543	-1,407	300	6	2	15	3
1,95	5,051	462	6	2	16	3
-5,236	-1,918	314	6	2	17	3
-2,181	0,242	410	6	2	18	3
-1,042	2,722	342	6	2	19	3
1,28	2,838	306	6	2	20	3
-0,563	2,322	474	6	2	21	3
-2,12	-0,698	318	6	2	2	4
-1,023	0,954	300	6	2	3	4
-0,955	0,849	314	6	2	4	4
1,303	2,728	438	6	2	5	4
1,525	3,96	320	6	2	6	4

-1,816	0,198	302	6	2	7	4
1,591	2,546	466	6	2	9	4
-1,684	1,029	488	6	2	10	4
-2,061	0,207	480	6	2	13	4
-7,212	-4,548	314	6	2	15	4
-2,141	-0,784	334	6	2	16	4
-4,514	-3,699	500	6	2	17	4
-1,929	0,31	416	6	2	18	4
-5,15	-1,402	322	6	2	19	4
0,171	1,708	464	6	2	20	4
-0,187	3,411	394	6	2	21	4
-1,446	-0,961	434	6	2	2	5
-0,612	-0,378	346	6	2	3	5
0,015	1,41	460	6	2	4	5
-0,615	0,037	340	6	2	5	5
0,342	0,51	432	6	2	6	5
-0,393	0,385	364	6	2	7	5
1,157	2,054	312	6	2	9	5
-2,34	-0,67	470	6	2	10	5
-1,441	0,558	314	6	2	13	5
-5,102	-3,363	310	6	2	15	5
-1,802	-0,237	326	6	2	16	5
-5,492	-4,904	442	6	2	17	5
-1,851	1,577	300	6	2	18	5
-0,774	1,57	316	6	2	19	5
-0,32	0,695	486	6	2	20	5
1,048	3,181	382	6	2	21	5
-2,219	-0,53	340	7	2	2	1
0,003	1,562	300	7	2	3	1
-1,536	-0,64	306	7	2	4	1
4,561	6,07	490	7	2	5	1
-1,311	-0,193	368	7	2	6	1
-0,991	1,075	460	7	2	7	1
0,775	2,749	460	7	2	9	1
1,619	4,796	494	7	2	10	1
1,914	3,692	442	7	2	13	1
9,25	11,975	460	7	2	15	1
-0,559	1,446	374	7	2	16	1
-2,582	-1,256	458	7	2	17	1
1,058	5,452	422	7	2	18	1
2,725	5,143	488	7	2	19	1
-0,172	0,802	414	7	2	20	1
1,182	4,346	406	7	2	21	1

-0,965	0,137	330	7	2	2	2
0,091	0,789	318	7	2	3	2
-1,561	-0,869	484	7	2	4	2
2,512	3,541	306	7	2	5	2
1,625	4,462	300	7	2	6	2
-2,382	0,348	310	7	2	7	2
1,208	2,307	476	7	2	9	2
-0,261	3,325	488	7	2	10	2
-0,348	1,96	486	7	2	13	2
-6,816	-5,582	308	7	2	15	2
1,932	3,539	492	7	2	16	2
-2,736	-1,181	326	7	2	17	2
-5,465	-2,512	306	7	2	18	2
-3,46	-0,057	300	7	2	19	2
-0,279	1,318	450	7	2	20	2
0,581	3,876	420	7	2	21	2
-1,975	0,68	300	7	2	2	3
0,161	1,207	412	7	2	3	3
-1,492	-0,461	488	7	2	4	3
1,675	2,881	408	7	2	5	3
-0,123	0,288	426	7	2	6	3
-1,206	1,182	422	7	2	7	3
1,791	3,597	332	7	2	9	3
-2,108	1,429	486	7	2	10	3
-1,042	0,695	482	7	2	13	3
-4,817	-1,522	310	7	2	15	3
2,644	5,14	496	7	2	16	3
-4,997	-2,085	314	7	2	17	3
-0,764	1,942	412	7	2	18	3
-1,879	1,701	342	7	2	19	3
0,495	1,216	308	7	2	20	3
1,519	3,482	410	7	2	21	3
-1,727	-0,185	368	7	2	2	4
-0,927	1,332	318	7	2	3	4
-1,877	-1,538	394	7	2	4	4
1,2	2,552	438	7	2	5	4
0,642	2,894	320	7	2	6	4
-0,585	1,813	368	7	2	7	4
3,502	5,004	498	7	2	9	4
0,389	4,514	488	7	2	10	4
-1,703	1,951	494	7	2	13	4
-7,098	-4,662	300	7	2	15	4
-1,24	-0,028	380	7	2	16	4

-3,314	0,277	314	7	2	17	4
-0,699	1,93	416	7	2	18	4
-5,357	-2,107	322	7	2	19	4
-0,357	1,026	464	7	2	20	4
0,253	2,858	396	7	2	21	4
-1,53	-1,053	332	7	2	2	5
0,124	0,484	412	7	2	3	5
-0,981	-0,065	476	7	2	4	5
0,449	0,942	340	7	2	5	5
-0,243	2,242	316	7	2	6	5
-0,332	0,884	362	7	2	7	5
2,91	4,338	488	7	2	9	5
-0,706	1,791	490	7	2	10	5
-1,172	0,827	492	7	2	13	5
-4,301	-3,159	308	7	2	15	5
-0,59	0,466	328	7	2	16	5
-4,451	-1,62	306	7	2	17	5
-2,507	-0,479	300	7	2	18	5
-0,921	1,107	316	7	2	19	5
-1,031	0,955	500	7	2	20	5
2,792	4,496	386	7	2	21	5
-1,793	0,471	372	8	2	2	1
-0,161	1,682	302	8	2	3	1
-1,73	-0,997	446	8	2	4	1
5,497	6,31	440	8	2	5	1
-2,7	-1,498	370	8	2	6	1
-1,2	0,805	494	8	2	7	1
2,724	4,426	466	8	2	9	1
4,237	9,156	492	8	2	10	1
1,988	4,802	492	8	2	13	1
8,045	11,133	454	8	2	15	1
0,97	2,939	426	8	2	16	1
-1,454	-0,156	402	8	2	17	1
1,904	6,746	418	8	2	18	1
3,456	6,542	488	8	2	19	1
-0,967	-0,078	410	8	2	20	1
2,536	5,633	406	8	2	21	1
-0,709	0,213	310	8	2	2	2
0,244	1,345	318	8	2	3	2
-2,481	-1,307	484	8	2	4	2
3,204	4,194	426	8	2	5	2
0,848	4,319	302	8	2	6	2
-1,082	0,977	310	8	2	7	2

2,911	4,118	492	8	2	9	2
2,768	7,154	488	8	2	10	2
-0,207	3,851	488	8	2	13	2
-5,774	-4,193	434	8	2	15	2
2,023	5,008	494	8	2	16	2
-2,904	-1,015	326	8	2	17	2
-5,124	-2,521	306	8	2	18	2
-2,712	0,286	344	8	2	19	2
-1,508	0,013	486	8	2	20	2
1,282	3,781	422	8	2	21	2
-1,866	-0,516	324	8	2	2	3
0,327	1,677	410	8	2	3	3
-1,947	-0,85	486	8	2	4	3
1,701	2,848	410	8	2	5	3
-0,101	0,463	372	8	2	6	3
-1,436	0,364	422	8	2	7	3
3,199	4,365	334	8	2	9	3
1,16	6,183	486	8	2	10	3
-1,282	2,589	486	8	2	13	3
-4,701	-1,741	310	8	2	15	3
3,294	6,7	496	8	2	16	3
-4,565	-1,238	300	8	2	17	3
-0,383	2,45	412	8	2	18	3
-1,738	0,992	342	8	2	19	3
-0,891	1,153	500	8	2	20	3
2,535	5,154	410	8	2	21	3
-0,857	1,111	368	8	2	2	4
-0,909	1,753	318	8	2	3	4
-2,685	-0,618	308	8	2	4	4
1,303	3,066	484	8	2	5	4
0,172	2,317	320	8	2	6	4
0,11	2,511	368	8	2	7	4
5,291	7,907	500	8	2	9	4
3,177	8,64	490	8	2	10	4
-2,596	2,719	494	8	2	13	4
-7,067	-6,065	354	8	2	15	4
-0,285	0,434	398	8	2	16	4
-2,856	-3,798	416	8	2	17	4
0,398	3,556	414	8	2	18	4
-4,843	-1,961	324	8	2	19	4
-1,201	-0,19	464	8	2	20	4
1,361	3,2	398	8	2	21	4
-1,405	-0,557	498	8	2	2	5

0,357	1,698	310	8	2	3	5
-1,578	-0,544	492	8	2	4	5
0,603	1,942	300	8	2	5	5
-0,815	2,037	316	8	2	6	5
-0,451	0,799	362	8	2	7	5
4,647	7,157	488	8	2	9	5
1,748	1,929	336	8	2	10	5
-1,398	2,262	492	8	2	13	5
-3,812	-3,066	322	8	2	15	5
0,738	1,979	408	8	2	16	5
-3,753	-1,273	308	8	2	17	5
-2,252	-0,784	312	8	2	18	5
-0,862	0,663	326	8	2	19	5
-1,836	-1,017	316	8	2	20	5
3,684	5,006	366	8	2	21	5
0,733	2,844	338	9	2	2	1
0,344	1,488	364	9	2	3	1
-2,119	-0,778	312	9	2	4	1
5,808	7,438	314	9	2	5	1
-2,611	0,663	436	9	2	6	1
2,525	5,428	312	9	2	7	1
-0,875	1,761	306	9	2	9	1
-3,442	-2,84	342	9	2	10	1
0,416	3,936	472	9	2	13	1
7,138	9,099	304	9	2	15	1
-3,719	3,783	308	9	2	16	1
-4,802	-1,462	316	9	2	17	1
1,772	4,782	322	9	2	18	1
-1,507	3,96	322	9	2	19	1
3,002	4,436	448	9	2	20	1
-3,552	-0,106	388	9	2	21	1
-2,348	-0,893	428	9	2	2	2
0,519	1,889	368	9	2	3	2
-1,239	2,139	316	9	2	4	2
0,257	2,995	370	9	2	5	2
-0,039	1,167	364	9	2	6	2
-0,041	3,139	446	9	2	7	2
-1,395	-0,053	404	9	2	9	2
-5,173	-3,402	478	9	2	10	2
-0,85	1,503	470	9	2	13	2
-11,067	-5,668	304	9	2	15	2
0,504	3,349	312	9	2	16	2
-3,858	0,864	306	9	2	17	2

-2,425	5,222	304	9	2	18	2
-1,415	3,187	300	9	2	19	2
2,601	4,546	304	9	2	20	2
-1,116	4,757	368	9	2	21	2
-2,087	-0,114	306	9	2	2	3
-0,399	1,577	348	9	2	3	3
-1,799	0,636	458	9	2	4	3
3,804	6,325	306	9	2	5	3
-3,273	-0,742	426	9	2	6	3
-0,303	3,203	304	9	2	7	3
0,656	4,388	314	9	2	9	3
-7,138	-0,869	314	9	2	10	3
-0,144	1,23	466	9	2	13	3
-9,079	-3,626	314	9	2	15	3
-0,905	0,791	462	9	2	16	3
-2,636	1,262	316	9	2	17	3
-1,563	4,534	312	9	2	18	3
0,497	5,471	310	9	2	19	3
4,397	6,236	304	9	2	20	3
-1,918	1,101	376	9	2	21	3
-0,254	1,611	438	9	2	2	4
-1,485	1,078	300	9	2	3	4
-1,505	0,821	464	9	2	4	4
1,04	3,708	304	9	2	5	4
-0,957	0,865	320	9	2	6	4
0,341	2,238	418	9	2	7	4
0,888	2,944	312	9	2	9	4
-3,56	4,168	302	9	2	10	4
-3,616	-2,094	480	9	2	13	4
-7,179	-2,287	312	9	2	15	4
-2,471	1,67	314	9	2	16	4
-6,456	-6,105	452	9	2	17	4
0,29	5,26	300	9	2	18	4
-3,315	0,622	302	9	2	19	4
2,152	3,863	464	9	2	20	4
-0,363	3,85	394	9	2	21	4
-1,113	0,685	300	9	2	2	5
-0,794	0,656	348	9	2	3	5
-0,699	1,451	312	9	2	4	5
0,865	2,475	308	9	2	5	5
-1,233	-0,062	366	9	2	6	5
0,635	2,385	430	9	2	7	5
0,696	2,196	312	9	2	9	5

-3,639	-3,142	366	9	2	10	5
-1,813	-0,425	468	9	2	13	5
-5,737	-3,072	320	9	2	15	5
-1,656	-0,152	358	9	2	16	5
-4,836	-3,584	440	9	2	17	5
-1,089	0,102	384	9	2	18	5
0,35	2,825	318	9	2	19	5
0,609	1,339	384	9	2	20	5
0,683	4,067	382	9	2	21	5
-0,743	1,525	336	10	2	2	1
-0,608	0,691	300	10	2	3	1
-1,534	0,119	474	10	2	4	1
2,66	3,938	490	10	2	5	1
-2,497	-0,075	436	10	2	6	1
0,446	4,153	408	10	2	7	1
-1,377	0,142	408	10	2	9	1
-3,994	-1,61	320	10	2	10	1
-0,88	1,587	470	10	2	13	1
9,752	10,856	460	10	2	15	1
-2,09	-0,693	328	10	2	16	1
-3,633	-0,498	316	10	2	17	1
1,032	4,047	324	10	2	18	1
1,577	3,194	304	10	2	19	1
0,78	1,814	314	10	2	20	1
-3,675	-0,651	356	10	2	21	1
-3,417	-1,915	426	10	2	2	2
-0,973	0,6	366	10	2	3	2
-1,544	0,529	314	10	2	4	2
0,684	2,683	370	10	2	5	2
-0,201	1,789	322	10	2	6	2
-1,387	1,734	452	10	2	7	2
-3,07	-1,102	312	10	2	9	2
-5,365	-2,261	500	10	2	10	2
-2,411	-0,28	300	10	2	13	2
-9,726	-5,735	310	10	2	15	2
0,736	2,594	484	10	2	16	2
-4,097	-3,351	340	10	2	17	2
-3,534	2,008	302	10	2	18	2
-1,731	0,332	318	10	2	19	2
0,524	2,179	350	10	2	20	2
-1,794	2,853	356	10	2	21	2
-3,343	-0,461	306	10	2	2	3
-0,569	1,4	348	10	2	3	3

-1,735	0,316	306	10	2	4	3
1,851	3,617	318	10	2	5	3
-1,636	1,067	306	10	2	6	3
-1,617	0,433	308	10	2	7	3
-0,347	2,24	352	10	2	9	3
-6,929	-2,442	316	10	2	10	3
-2,933	-0,199	398	10	2	13	3
-5,579	-1,594	306	10	2	15	3
-1,335	-0,234	312	10	2	16	3
-4,393	-1,027	316	10	2	17	3
-2,516	3,154	312	10	2	18	3
-2,409	-0,04	310	10	2	19	3
1,943	2,939	384	10	2	20	3
-1,626	1,571	378	10	2	21	3
-1,766	-0,698	310	10	2	2	4
-1,677	0,576	300	10	2	3	4
-1,22	-0,19	464	10	2	4	4
-0,029	1,191	366	10	2	5	4
-0,463	1,306	320	10	2	6	4
0,029	1,902	388	10	2	7	4
0,221	1,574	384	10	2	9	4
-4,832	0,279	306	10	2	10	4
-5,809	-2,563	312	10	2	13	4
-6,2	-2,031	300	10	2	15	4
-1,921	-0,835	486	10	2	16	4
-6,743	-6,811	464	10	2	17	4
0,848	5,721	316	10	2	18	4
-4,417	-1,729	322	10	2	19	4
0,997	2,144	378	10	2	20	4
-0,426	3,39	394	10	2	21	4
-1,423	-0,33	434	10	2	2	5
-0,504	0,672	346	10	2	3	5
-0,759	0,84	310	10	2	4	5
-0,039	1,063	308	10	2	5	5
-0,943	0,383	316	10	2	6	5
-0,725	0,913	324	10	2	7	5
0,799	1,817	380	10	2	9	5
-3,653	0,537	306	10	2	10	5
-3,263	-1,438	312	10	2	13	5
-5,084	-2,395	332	10	2	15	5
-0,838	0,793	308	10	2	16	5
-4,992	-4,498	438	10	2	17	5
-1,646	3,767	300	10	2	18	5

0,575	1,794	318	10	2	19	5
-0,413	0,688	300	10	2	20	5
1,186	4,295	370	10	2	21	5
-1,099	1,288	340	11	2	2	1
-0,846	0,533	302	11	2	3	1
-1,543	0,245	476	11	2	4	1
2,778	5,431	492	11	2	5	1
-2,354	-0,828	436	11	2	6	1
0,368	3,259	408	11	2	7	1
0,28	3,333	494	11	2	9	1
-2,024	0,85	494	11	2	10	1
0,171	1,822	490	11	2	13	1
8,721	10,515	458	11	2	15	1
-2,513	-0,134	496	11	2	16	1
-2,634	0,437	316	11	2	17	1
1,109	4,49	326	11	2	18	1
0,524	2,402	470	11	2	19	1
-0,338	1,382	314	11	2	20	1
-1,706	1,582	356	11	2	21	1
-3,67	-2,256	300	11	2	2	2
-1,02	0,522	366	11	2	3	2
-1,694	-0,123	312	11	2	4	2
1,419	3,092	366	11	2	5	2
-1,111	0,936	322	11	2	6	2
-1,533	-0,229	452	11	2	7	2
-0,871	0,566	492	11	2	9	2
-2,855	1,64	500	11	2	10	2
-1,707	1,12	488	11	2	13	2
-8,344	-4,932	316	11	2	15	2
0,784	3,91	492	11	2	16	2
-4,115	-2,686	340	11	2	17	2
-4,533	-0,024	306	11	2	18	2
-2,365	0,528	350	11	2	19	2
-0,463	2,161	350	11	2	20	2
-0,397	4,225	356	11	2	21	2
-3,526	-0,516	306	11	2	2	3
-0,21	1,508	338	11	2	3	3
-1,921	-0,626	484	11	2	4	3
1,127	2,517	318	11	2	5	3
-1,156	1,909	308	11	2	6	3
-2,128	0,139	340	11	2	7	3
1,78	4,463	352	11	2	9	3
-4,526	-0,388	330	11	2	10	3

-2,891	0,765	496	11	2	13	3
-5,743	-2,085	310	11	2	15	3
-0,968	0,652	494	11	2	16	3
-5,272	-1,654	316	11	2	17	3
-1,746	3,601	316	11	2	18	3
-1,079	1,429	492	11	2	19	3
0,627	2,199	368	11	2	20	3
0,409	3,372	360	11	2	21	3
-1,834	0,003	310	11	2	2	4
-1,46	0,845	302	11	2	3	4
-1,398	-0,399	478	11	2	4	4
-0,453	0,602	500	11	2	5	4
-0,626	1,32	322	11	2	6	4
-0,157	1,873	388	11	2	7	4
2,309	3,948	498	11	2	9	4
-2,878	1,916	318	11	2	10	4
-4,909	-1,585	312	11	2	13	4
-7,215	-3,278	300	11	2	15	4
-2,245	0,938	312	11	2	16	4
-6,427	-6,683	498	11	2	17	4
0,359	5,387	318	11	2	18	4
-3,788	-1,545	320	11	2	19	4
0,534	2,599	498	11	2	20	4
0,529	3,599	378	11	2	21	4
-1,695	-0,714	330	11	2	2	5
-0,234	0,951	346	11	2	3	5
-0,734	0,813	308	11	2	4	5
0,151	1,407	306	11	2	5	5
-1,384	0,723	320	11	2	6	5
-1,781	-0,301	314	11	2	7	5
2,463	4,071	488	11	2	9	5
-1,673	2,138	308	11	2	10	5
-2,523	-0,905	494	11	2	13	5
-3,817	-1,056	332	11	2	15	5
-0,855	0,354	306	11	2	16	5
-4,913	-4,972	440	11	2	17	5
-2,096	2,338	302	11	2	18	5
0,239	1,548	352	11	2	19	5
-1,191	0,282	352	11	2	20	5
2,476	5,702	370	11	2	21	5
-1,831	0,918	306	12	2	2	1
-1,326	0,324	304	12	2	3	1
-1,081	0,831	486	12	2	4	1

3,528	6,242	500	12	2	5	1
-2,518	-0,64	496	12	2	6	1
0,11	2,282	402	12	2	7	1
1,618	5,616	494	12	2	9	1
-0,444	3,788	492	12	2	10	1
0,636	3,582	492	12	2	13	1
8,437	10,639	454	12	2	15	1
-2,019	-1,237	338	12	2	16	1
-1,521	1,292	332	12	2	17	1
1,373	4,02	332	12	2	18	1
2,17	4,821	488	12	2	19	1
-1,553	0,912	312	12	2	20	1
0,425	3,309	358	12	2	21	1
-4,325	-2,243	300	12	2	2	2
-1,227	0,037	376	12	2	3	2
-1,78	-0,196	482	12	2	4	2
2,261	3,64	360	12	2	5	2
-1,546	0,56	322	12	2	6	2
-1,567	0,017	336	12	2	7	2
0,99	2,817	492	12	2	9	2
-0,779	4,311	492	12	2	10	2
-1,451	3,928	500	12	2	13	2
-6,736	-3,178	318	12	2	15	2
1,28	6,138	494	12	2	16	2
-3,957	-1,867	338	12	2	17	2
-4,983	-1,309	306	12	2	18	2
-2,533	0,987	348	12	2	19	2
-1,311	2,259	350	12	2	20	2
1,106	5,482	356	12	2	21	2
-3,936	-0,349	304	12	2	2	3
-0,518	1,396	332	12	2	3	3
-1,761	-0,109	482	12	2	4	3
0,434	1,407	500	12	2	5	3
-1,281	2,266	308	12	2	6	3
-2,259	1,175	330	12	2	7	3
3,921	6,465	352	12	2	9	3
-2,708	1,395	330	12	2	10	3
-2,239	3,447	496	12	2	13	3
-5,558	-2,082	312	12	2	15	3
0,197	4,416	496	12	2	16	3
-5,335	-2,06	314	12	2	17	3
0,008	3,601	316	12	2	18	3
-1,524	1,563	492	12	2	19	3

-0,767	2,077	366	12	2	20	3
1,819	4,766	348	12	2	21	3
-1,953	0,436	310	12	2	2	4
-1,638	0,896	302	12	2	3	4
-0,884	0,443	478	12	2	4	4
-0,398	1,319	500	12	2	5	4
-0,435	1,739	348	12	2	6	4
-0,892	1,342	388	12	2	7	4
5,024	7,627	500	12	2	9	4
-1,161	3,467	320	12	2	10	4
-5,577	-2,039	498	12	2	13	4
-6,955	-3,708	324	12	2	15	4
-2,198	0,114	312	12	2	16	4
-5,748	-3,183	352	12	2	17	4
0,304	4,721	316	12	2	18	4
-3,34	-1,546	368	12	2	19	4
-0,424	1,94	364	12	2	20	4
1,404	4,525	362	12	2	21	4
-2,302	-1,429	354	12	2	2	5
-0,246	1,756	300	12	2	3	5
-0,396	1,022	308	12	2	4	5
0,333	1,697	300	12	2	5	5
-1,844	1,113	320	12	2	6	5
-2,568	-0,876	310	12	2	7	5
4,037	6,605	488	12	2	9	5
-0,167	3,811	310	12	2	10	5
-2,043	1,037	494	12	2	13	5
-3,684	-0,931	334	12	2	15	5
-0,178	0,65	324	12	2	16	5
-4,187	-4,438	422	12	2	17	5
-1,283	1,967	310	12	2	18	5
0,118	1,54	352	12	2	19	5
-2,216	0,05	352	12	2	20	5
3,565	6,745	370	12	2	21	5
1,993	6,558	608	1	3	2	1
5,383	9,809	632	1	3	3	1
0,103	3,803	612	1	3	4	1
9,602	14,517	648	1	3	5	1
-1,231	4,396	652	1	3	6	1
-2,28	1,147	628	1	3	7	1
4,11	8,279	638	1	3	9	1
2,829	9,886	626	1	3	10	1
4,18	10,177	558	1	3	13	1

13,726	17,687	624	1	3	15	1
1,103	4,139	660	1	3	16	1
-0,032	6,669	616	1	3	17	1
1,549	6,911	622	1	3	18	1
9,832	14,919	656	1	3	19	1
3,771	6,682	616	1	3	20	1
-0,399	4,244	540	1	3	21	1
-1,288	3,849	614	1	3	2	2
1,209	5,402	636	1	3	3	2
-0,254	5,533	632	1	3	4	2
1,628	4,495	624	1	3	5	2
0,827	5,528	644	1	3	6	2
-3,658	-0,334	646	1	3	7	2
3,435	5,975	638	1	3	9	2
5,496	11,788	638	1	3	10	2
-2,315	-1,137	576	1	3	13	2
-8,142	-4,375	624	1	3	15	2
6,886	10,379	662	1	3	16	2
-2,804	2,255	624	1	3	17	2
-4,963	-2,097	590	1	3	18	2
-1,82	2,916	602	1	3	19	2
4,821	7,608	584	1	3	20	2
0,796	4,259	636	1	3	21	2
-1,943	1,392	608	1	3	2	3
2,756	6,454	632	1	3	3	3
1,715	7,845	624	1	3	4	3
4,716	7,571	678	1	3	5	3
-4,762	1,003	660	1	3	6	3
4,095	19,276	672	1	3	7	3
6,083	10,754	664	1	3	9	3
3,452	9,926	618	1	3	10	3
0,899	4,63	672	1	3	13	3
-8,693	-6,261	652	1	3	15	3
9,417	14,401	646	1	3	16	3
0,634	6,464	632	1	3	17	3
-2,781	4,559	648	1	3	18	3
0,298	5,933	646	1	3	19	3
3,865	6,073	672	1	3	20	3
1,059	6,897	560	1	3	21	3
0,329	4,668	602	1	3	2	4
2,14	6,532	634	1	3	3	4
-0,691	3,817	664	1	3	4	4
2,711	6,94	622	1	3	5	4

-2,297	2,486	650	1	3	6	4
-2,353	3,372	654	1	3	7	4
5,654	8,475	562	1	3	9	4
5,034	11,419	620	1	3	10	4
-2,21	0,522	656	1	3	13	4
-8,949	-5,506	614	1	3	15	4
2,371	6,546	650	1	3	16	4
-1,671	5,405	604	1	3	17	4
-2,005	3,122	650	1	3	18	4
-2,645	1,726	622	1	3	19	4
4,724	9,254	596	1	3	20	4
1,295	4,033	628	1	3	21	4
-0,638	2,849	616	1	3	2	5
0,987	3,549	646	1	3	3	5
1,958	5,569	612	1	3	4	5
3,026	5,633	640	1	3	5	5
-0,314	4,837	642	1	3	6	5
6,552	10,593	630	1	3	7	5
4,577	6,91	614	1	3	9	5
4,965	12,034	630	1	3	10	5
2,177	4,698	550	1	3	13	5
-7,281	-5,577	632	1	3	15	5
-0,234	4,378	642	1	3	16	5
-1,927	4,589	626	1	3	17	5
-0,742	3,689	666	1	3	18	5
1,362	4,861	642	1	3	19	5
1,841	3,466	584	1	3	20	5
2,433	7,612	550	1	3	21	5
-2,153	1,623	592	2	3	2	1
2,964	6,215	634	2	3	3	1
-2,223	0,72	608	2	3	4	1
7,948	11,546	648	2	3	5	1
-0,973	4,274	652	2	3	6	1
-1,581	-0,127	530	2	3	7	1
1,084	3,501	658	2	3	9	1
0,283	5,976	618	2	3	10	1
1,845	6,644	558	2	3	13	1
12,141	14,55	632	2	3	15	1
-1,595	0,422	654	2	3	16	1
-0,751	5,245	604	2	3	17	1
-0,218	4,964	622	2	3	18	1
9,136	12,65	688	2	3	19	1
1,039	2,846	598	2	3	20	1

-2,876	1,515	540	2	3	21	1
-2,797	1,877	612	2	3	2	2
-0,124	2,104	638	2	3	3	2
-2,08	2,091	630	2	3	4	2
2,294	4,569	654	2	3	5	2
0,564	4,703	644	2	3	6	2
-3,736	-1,945	648	2	3	7	2
-1,364	0,513	626	2	3	9	2
1,678	7,414	620	2	3	10	2
-1,728	0,1	580	2	3	13	2
-9,382	-7,138	608	2	3	15	2
4,547	7,237	558	2	3	16	2
-4,469	-0,229	608	2	3	17	2
-7,297	-4,242	640	2	3	18	2
-3,112	-0,371	602	2	3	19	2
1,546	4,42	580	2	3	20	2
-1,921	1,801	536	2	3	21	2
-2,375	0,754	608	2	3	2	3
1,375	4,255	632	2	3	3	3
0,196	4,576	618	2	3	4	3
2,878	4,917	686	2	3	5	3
-2,896	2,99	658	2	3	6	3
-2,584	0,131	664	2	3	7	3
3,502	7,135	700	2	3	9	3
0,862	6,792	616	2	3	10	3
-0,928	2,227	614	2	3	13	3
-8,831	-6,891	668	2	3	15	3
6,335	9,815	648	2	3	16	3
-1,428	3,463	616	2	3	17	3
-0,64	3,669	630	2	3	18	3
0,347	4,144	646	2	3	19	3
1,758	3,63	594	2	3	20	3
0,178	5,365	560	2	3	21	3
-2,268	1,234	600	2	3	2	4
0,684	3,898	622	2	3	3	4
-1,742	1,85	608	2	3	4	4
2,558	5,098	620	2	3	5	4
-4,403	0,56	650	2	3	6	4
-2,189	0,829	644	2	3	7	4
2,749	4,652	564	2	3	9	4
2,3	7,527	618	2	3	10	4
-2,707	-0,685	656	2	3	13	4
-10,689	-7,371	574	2	3	15	4

2,319	5,635	632	2	3	16	4
-3,363	2,297	604	2	3	17	4
-1,959	1,674	650	2	3	18	4
-2,535	0,367	622	2	3	19	4
1,255	4,255	594	2	3	20	4
-0,184	3,662	542	2	3	21	4
-0,914	1,89	614	2	3	2	5
0,652	2,641	634	2	3	3	5
0,512	3,054	608	2	3	4	5
2,14	4,193	622	2	3	5	5
-1,162	3,432	636	2	3	6	5
-1,671	0,149	664	2	3	7	5
2,707	3,982	628	2	3	9	5
2,846	8,819	616	2	3	10	5
0,252	2,51	574	2	3	13	5
-7,282	-5,436	584	2	3	15	5
-1,052	1,925	644	2	3	16	5
-2,373	2,989	622	2	3	17	5
-1,552	1,854	678	2	3	18	5
1,938	4,118	596	2	3	19	5
0,813	2,497	598	2	3	20	5
2,386	7,606	548	2	3	21	5
-2,784	0,613	592	3	3	2	1
2,682	5,327	634	3	3	3	1
-2,809	-0,633	608	3	3	4	1
7,965	11,125	654	3	3	5	1
-1,348	3,987	668	3	3	6	1
-1,029	0,733	530	3	3	7	1
0,574	2,538	660	3	3	9	1
0,486	4,864	618	3	3	10	1
2,933	6,497	558	3	3	13	1
15,294	18,174	604	3	3	15	1
-1,915	-0,066	524	3	3	16	1
-1,187	4,112	602	3	3	17	1
-1,089	4,34	620	3	3	18	1
8,615	11,764	678	3	3	19	1
-0,33	1,465	532	3	3	20	1
-2,438	1,951	540	3	3	21	1
-2,422	1,337	620	3	3	2	2
0,292	2,4	610	3	3	3	2
-2,366	0,612	630	3	3	4	2
1,778	3,933	652	3	3	5	2
0,981	4,841	678	3	3	6	2

-2,783	-0,427	508	3	3	7	2
-0,617	1,66	626	3	3	9	2
1,617	6,146	620	3	3	10	2
-1,488	0,314	580	3	3	13	2
-10,341	-8,919	612	3	3	15	2
3,553	6,183	542	3	3	16	2
-4,597	-0,778	624	3	3	17	2
-7,366	-5,15	600	3	3	18	2
-2,762	-0,627	604	3	3	19	2
-0,447	2,279	578	3	3	20	2
-2,045	2,861	536	3	3	21	2
-2,487	-0,052	608	3	3	2	3
1,636	4,31	634	3	3	3	3
-0,415	2,742	652	3	3	4	3
1,95	3,692	686	3	3	5	3
-2,591	3,144	658	3	3	6	3
-1,708	0,37	666	3	3	7	3
2,976	6,4	668	3	3	9	3
1,38	6,289	616	3	3	10	3
-1,245	2,96	612	3	3	13	3
-1,196	1,478	676	3	3	15	3
4,923	7,309	648	3	3	16	3
-2,905	1,124	616	3	3	17	3
-0,38	3,224	630	3	3	18	3
0,382	2,504	596	3	3	19	3
0,271	2,209	684	3	3	20	3
1,014	5,329	560	3	3	21	3
-2,378	0,695	600	3	3	2	4
0,872	3,765	622	3	3	3	4
-1,956	0,997	608	3	3	4	4
1,373	3,356	652	3	3	5	4
-2,393	2,358	652	3	3	6	4
-1,457	1,485	656	3	3	7	4
3,257	4,946	666	3	3	9	4
2,762	7,228	606	3	3	10	4
-2,676	-1,154	586	3	3	13	4
12,874	15,835	574	3	3	15	4
0,588	3,081	632	3	3	16	4
-3,86	0,645	604	3	3	17	4
-1,529	0,07	648	3	3	18	4
-2,889	-0,578	694	3	3	19	4
-0,194	1,942	594	3	3	20	4
-0,155	4,956	542	3	3	21	4

-0,574	1,546	614	3	3	2	5
1,815	3,486	636	3	3	3	5
-0,027	1,952	608	3	3	4	5
1,983	3,267	622	3	3	5	5
-1,337	2,948	636	3	3	6	5
-1,133	0,474	664	3	3	7	5
2,862	3,944	630	3	3	9	5
2,813	7,502	614	3	3	10	5
-0,013	2,219	574	3	3	13	5
4,197	6,351	582	3	3	15	5
-2,087	-0,136	642	3	3	16	5
-3,009	1,595	606	3	3	17	5
-3,185	-0,318	682	3	3	18	5
2,251	4,353	580	3	3	19	5
-0,51	0,953	600	3	3	20	5
2,549	8,06	534	3	3	21	5
-2,315	-0,633	592	4	3	2	1
1,714	3,286	634	4	3	3	1
-0,605	0,778	526	4	3	4	1
6,301	8,938	658	4	3	5	1
-0,591	3,143	680	4	3	6	1
-1	1,695	510	4	3	7	1
1,34	3,6	662	4	3	9	1
-0,084	2,156	698	4	3	10	1
2,197	3,988	608	4	3	13	1
12,688	14,574	636	4	3	15	1
-1,804	3,815	508	4	3	16	1
-0,502	2,024	600	4	3	17	1
-0,52	2,958	618	4	3	18	1
5,401	8,34	684	4	3	19	1
-0,848	3,522	532	4	3	20	1
-2,326	3,058	658	4	3	21	1
-1,342	1,361	654	4	3	2	2
0,751	2,691	610	4	3	3	2
-1,456	-0,344	644	4	3	4	2
2,354	4,792	686	4	3	5	2
2,064	5,491	676	4	3	6	2
-2,089	1,831	508	4	3	7	2
0,075	3,539	694	4	3	9	2
-0,223	1,771	682	4	3	10	2
0,478	2,928	598	4	3	13	2
-9,163	-7,22	602	4	3	15	2
3,731	7,856	524	4	3	16	2

-3,448	-0,887	628	4	3	17	2
-7,915	-4,218	600	4	3	18	2
-5,678	-1,999	500	4	3	19	2
-1,064	1,951	548	4	3	20	2
-1,81	1,862	504	4	3	21	2
-1,788	-0,527	690	4	3	2	3
0,863	2,776	632	4	3	3	3
-0,465	1,002	668	4	3	4	3
0,813	2,647	530	4	3	5	3
0,961	4,638	658	4	3	6	3
-0,909	0,976	504	4	3	7	3
3,938	7,587	670	4	3	9	3
1,44	4,93	680	4	3	10	3
1,871	6,719	614	4	3	13	3
-5,39	-4,068	694	4	3	15	3
4,01	7,868	516	4	3	16	3
-2,883	-1,06	616	4	3	17	3
0,359	4,09	598	4	3	18	3
0,687	3,097	694	4	3	19	3
-0,476	1,556	502	4	3	20	3
1,411	5,358	664	4	3	21	3
-1,678	-0,653	600	4	3	2	4
0,493	2,899	622	4	3	3	4
0,048	1,189	604	4	3	4	4
-0,004	2,387	534	4	3	5	4
1,762	4,406	672	4	3	6	4
-0,321	1,71	516	4	3	7	4
4,596	7,404	700	4	3	9	4
1,458	5,598	682	4	3	10	4
-0,956	1,604	602	4	3	13	4
-8,084	-6,036	576	4	3	15	4
0,168	4,648	514	4	3	16	4
-2,985	-1,148	650	4	3	17	4
-0,296	1,004	536	4	3	18	4
-4,787	-2,994	520	4	3	19	4
-1,23	2,773	528	4	3	20	4
-1,399	2,524	538	4	3	21	4
0,167	1,724	652	4	3	2	5
0,992	2,147	626	4	3	3	5
0,054	0,891	606	4	3	4	5
0,46	2,556	522	4	3	5	5
-0,488	1,916	616	4	3	6	5
-1,534	1,383	506	4	3	7	5

2,901	3,847	668	4	3	9	5
0,021	3,645	692	4	3	10	5
-0,287	1,92	612	4	3	13	5
-5,08	-3,782	596	4	3	15	5
0,122	3,438	520	4	3	16	5
-2,612	-0,556	602	4	3	17	5
-2,422	0,918	600	4	3	18	5
-0,232	2,299	502	4	3	19	5
-1,858	1,318	530	4	3	20	5
0,775	4,858	518	4	3	21	5
3,23	7,563	606	5	3	2	1
3,998	7,387	632	5	3	3	1
0,107	3,389	628	5	3	4	1
6,629	10,5	646	5	3	5	1
-1,255	3,77	642	5	3	6	1
3,719	7,219	646	5	3	7	1
2,184	6,245	622	5	3	9	1
3,1	9,371	626	5	3	10	1
3,46	8,786	558	5	3	13	1
52,078	68,24	692	5	3	15	1
2,466	5,998	608	5	3	16	1
-1,106	4,525	652	5	3	17	1
4,833	9,604	622	5	3	18	1
2,864	7,84	640	5	3	19	1
3,195	5,364	600	5	3	20	1
3,394	7,636	624	5	3	21	1
-0,243	4,136	630	5	3	2	2
2,641	6,653	644	5	3	3	2
0,071	4,684	632	5	3	4	2
1,137	3,629	652	5	3	5	2
3,451	7,434	646	5	3	6	2
3,5	6,768	646	5	3	7	2
2,492	6,134	638	5	3	9	2
6,226	12,665	638	5	3	10	2
-0,871	1,386	676	5	3	13	2
178,013	173,01	558	5	3	15	2
4,242	7,11	662	5	3	16	2
-1,025	3,697	624	5	3	17	2
-5,512	-2,216	590	5	3	18	2
0,325	5,007	602	5	3	19	2
4,224	6,332	584	5	3	20	2
2,576	8,101	620	5	3	21	2
-0,083	2,272	608	5	3	2	3

2,261	5,324	634	5	3	3	3
1,156	6,187	656	5	3	4	3
1,672	4,57	692	5	3	5	3
-0,881	4,416	660	5	3	6	3
1,61	8,052	658	5	3	7	3
3,409	7,639	664	5	3	9	3
2,391	7,887	634	5	3	10	3
1,047	4,223	670	5	3	13	3
-192,526	-171,811	516	5	3	15	3
9,276	14,257	648	5	3	16	3
0,516	5,579	666	5	3	17	3
-2,192	3,736	630	5	3	18	3
6,845	14,312	696	5	3	19	3
3,362	5,37	658	5	3	20	3
2,17	6,704	576	5	3	21	3
-0,692	2,941	636	5	3	2	4
1,783	5,415	632	5	3	3	4
-0,534	3,779	664	5	3	4	4
1,647	5,597	622	5	3	5	4
0,844	4,821	642	5	3	6	4
2,012	7,499	670	5	3	7	4
5,671	9,071	628	5	3	9	4
5,378	10,817	636	5	3	10	4
-3,618	-0,843	658	5	3	13	4
-92,094	-80,487	514	5	3	15	4
1,471	6,888	632	5	3	16	4
-0,841	4,99	620	5	3	17	4
-0,586	4,969	650	5	3	18	4
-4,667	-0,457	654	5	3	19	4
2,847	6,002	596	5	3	20	4
3,214	9,577	628	5	3	21	4
0,304	3,237	616	5	3	2	5
0,377	2,315	646	5	3	3	5
1,735	4,752	630	5	3	4	5
3,262	5,333	622	5	3	5	5
1,404	5,818	648	5	3	6	5
4,223	7,546	650	5	3	7	5
2,164	4,536	614	5	3	9	5
4,782	10,287	632	5	3	10	5
0,182	2,25	552	5	3	13	5
17,045	24,789	652	5	3	15	5
-1,077	3,099	656	5	3	16	5
-2,005	3,491	626	5	3	17	5

-0,318	3,109	666	5	3	18	5
3,133	6,396	642	5	3	19	5
1,941	3,483	666	5	3	20	5
3,371	6,648	552	5	3	21	5
-0,254	2,472	592	6	3	2	1
2,349	4,706	634	6	3	3	1
-0,985	0,702	612	6	3	4	1
5,355	8,218	648	6	3	5	1
2,2	5,945	670	6	3	6	1
-0,944	1,795	630	6	3	7	1
1,483	3,694	624	6	3	9	1
0,558	4,335	624	6	3	10	1
2,906	6,687	558	6	3	13	1
11,932	13,97	628	6	3	15	1
-0,359	2,622	610	6	3	16	1
-2,489	1,274	614	6	3	17	1
4,609	8,313	620	6	3	18	1
5,569	8,846	654	6	3	19	1
0,901	2,922	534	6	3	20	1
2,335	5,099	540	6	3	21	1
-0,728	2,443	632	6	3	2	2
0,055	1,935	642	6	3	3	2
-0,212	2,453	632	6	3	4	2
1,345	2,89	638	6	3	5	2
3,594	6,827	646	6	3	6	2
-1,931	0,551	676	6	3	7	2
0,957	3,048	628	6	3	9	2
3,341	7,51	622	6	3	10	2
0,439	2,599	574	6	3	13	2
-7,062	-4,933	608	6	3	15	2
3,607	6,157	578	6	3	16	2
-1,849	1,415	624	6	3	17	2
-5,536	-2,649	592	6	3	18	2
-2,569	-0,106	600	6	3	19	2
0,709	2,932	600	6	3	20	2
2,198	5,143	604	6	3	21	2
-0,851	1,083	608	6	3	2	3
0,515	2,8	636	6	3	3	3
0,73	3,728	656	6	3	4	3
2,148	4,127	688	6	3	5	3
0,319	4,427	658	6	3	6	3
-0,163	2,66	672	6	3	7	3
3,218	6,527	700	6	3	9	3

0,938	4,694	634	6	3	10	3
2,298	4,956	644	6	3	13	3
-6,297	-4,38	652	6	3	15	3
6,888	8,947	648	6	3	16	3
-2,324	1,306	616	6	3	17	3
-3,251	0,154	630	6	3	18	3
2,015	6,243	644	6	3	19	3
1,468	2,939	592	6	3	20	3
3,056	6,592	574	6	3	21	3
-1,301	0,98	602	6	3	2	4
0,902	3,741	622	6	3	3	4
-0,104	2,903	662	6	3	4	4
0,857	2,322	622	6	3	5	4
2,223	5,967	674	6	3	6	4
-2,909	-0,089	654	6	3	7	4
3,38	5,308	630	6	3	9	4
3,307	6,521	618	6	3	10	4
-0,472	2,168	658	6	3	13	4
-7,556	-5,78	622	6	3	15	4
1,711	5,387	632	6	3	16	4
-1,822	1,696	604	6	3	17	4
-0,211	3,52	650	6	3	18	4
-3,789	-0,936	618	6	3	19	4
1,034	3,354	596	6	3	20	4
2,891	5,573	542	6	3	21	4
-0,219	1,258	600	6	3	2	5
0,373	1,705	614	6	3	3	5
1,45	3,362	612	6	3	4	5
0,633	1,752	624	6	3	5	5
1,758	5,673	648	6	3	6	5
-0,303	1,628	664	6	3	7	5
2,167	3,668	616	6	3	9	5
1,925	5,362	616	6	3	10	5
0,517	2,635	576	6	3	13	5
-5,968	-4,641	592	6	3	15	5
-0,657	1,286	642	6	3	16	5
-2,635	1,353	624	6	3	17	5
-1,584	0,85	578	6	3	18	5
1,719	3,715	640	6	3	19	5
0,278	1,679	600	6	3	20	5
2,696	5,171	550	6	3	21	5
-1,873	0,211	592	7	3	2	1
2,618	4,561	634	7	3	3	1

-2,31	-1,047	612	7	3	4	1
6,536	8,837	650	7	3	5	1
0,651	4,175	670	7	3	6	1
-2,305	-0,455	644	7	3	7	1
1,829	3,361	624	7	3	9	1
1,36	3,247	626	7	3	10	1
2,642	5,43	558	7	3	13	1
13,351	15,482	630	7	3	15	1
-0,437	2,15	608	7	3	16	1
-1,37	1,904	608	7	3	17	1
2,564	6,443	620	7	3	18	1
6,502	9,18	686	7	3	19	1
0,519	3,486	534	7	3	20	1
1,732	4,36	656	7	3	21	1
-0,689	1,992	646	7	3	2	2
0,447	2,018	610	7	3	3	2
-1,592	0,22	634	7	3	4	2
2,679	4,311	688	7	3	5	2
3,218	6,427	680	7	3	6	2
-2,969	-0,183	506	7	3	7	2
1,483	3,248	638	7	3	9	2
2,867	4,909	620	7	3	10	2
0,814	2,447	576	7	3	13	2
-7,657	-6,112	606	7	3	15	2
3,676	6,103	578	7	3	16	2
-1,815	0,846	624	7	3	17	2
-5,366	-2,649	594	7	3	18	2
-3,571	-1,477	600	7	3	19	2
-0,186	1,612	600	7	3	20	2
1,195	4,325	536	7	3	21	2
-1,375	0,391	690	7	3	2	3
1,3	3,312	634	7	3	3	3
-0,608	1,353	656	7	3	4	3
2,131	2,961	512	7	3	5	3
0,526	3,95	658	7	3	6	3
-0,06	2,031	674	7	3	7	3
3,715	8,037	700	7	3	9	3
2,53	4,661	646	7	3	10	3
1,767	4,416	632	7	3	13	3
-5,503	-3,97	622	7	3	15	3
6,697	8,177	546	7	3	16	3
-1,877	1,042	616	7	3	17	3
-1,472	1,566	628	7	3	18	3

0,105	3,164	644	7	3	19	3
0,801	2,175	574	7	3	20	3
3,876	5,601	608	7	3	21	3
-1,461	0,337	700	7	3	2	4
0,844	3,343	622	7	3	3	4
-1,426	0,502	656	7	3	4	4
1,262	2,484	684	7	3	5	4
1,266	5,111	674	7	3	6	4
-1,191	1,185	652	7	3	7	4
4,686	6,359	630	7	3	9	4
3,826	5,493	608	7	3	10	4
0,446	2,304	668	7	3	13	4
-7,612	-6,007	570	7	3	15	4
2,187	4,822	596	7	3	16	4
-0,442	2,069	604	7	3	17	4
0,496	2,989	648	7	3	18	4
-4,692	-2,787	618	7	3	19	4
0,76	2,439	580	7	3	20	4
1,521	4,329	540	7	3	21	4
-0,502	0,884	650	7	3	2	5
0,964	2,126	616	7	3	3	5
-0,297	0,89	612	7	3	4	5
1,349	2,639	526	7	3	5	5
0,813	4,246	648	7	3	6	5
-0,281	1,074	664	7	3	7	5
3,035	4,013	624	7	3	9	5
1,796	3,656	614	7	3	10	5
0,505	2,391	574	7	3	13	5
-5,045	-3,858	592	7	3	15	5
0,258	1,271	606	7	3	16	5
-1,992	1,085	606	7	3	17	5
-3,083	-0,713	580	7	3	18	5
1,049	2,045	600	7	3	19	5
-0,699	0,955	500	7	3	20	5
3,331	5,998	520	7	3	21	5
-2,278	-0,632	590	8	3	2	1
2,62	4,426	634	8	3	3	1
-2,631	-1,782	504	8	3	4	1
7,568	10,029	522	8	3	5	1
-1,258	2,474	682	8	3	6	1
-1,95	0,097	510	8	3	7	1
3,037	5,394	508	8	3	9	1
2,209	2,62	626	8	3	10	1

2,433	4,673	558	8	3	13	1
11,145	13,385	632	8	3	15	1
0,1	3,084	520	8	3	16	1
-0,212	2,481	608	8	3	17	1
2,465	6,369	620	8	3	18	1
7,31	9,583	684	8	3	19	1
-0,201	4,108	532	8	3	20	1
1,351	5,635	656	8	3	21	1
-0,877	2,216	654	8	3	2	2
0,776	2,318	610	8	3	3	2
-2,739	-1,324	644	8	3	4	2
2,907	5,454	514	8	3	5	2
2,77	6,585	680	8	3	6	2
-2,002	2,316	508	8	3	7	2
2,436	4,822	694	8	3	9	2
2,727	5,021	524	8	3	10	2
0,967	2,597	634	8	3	13	2
-6,725	-5,015	602	8	3	15	2
3,778	7,173	524	8	3	16	2
-1,832	0,46	624	8	3	17	2
-5,853	-2,303	598	8	3	18	2
-2,674	-1,392	552	8	3	19	2
-1,036	1,02	542	8	3	20	2
0,47	4,341	518	8	3	21	2
-1,266	0,649	690	8	3	2	3
1,495	3,408	634	8	3	3	3
-1,669	-0,35	602	8	3	4	3
1,795	4,669	510	8	3	5	3
0,947	4,351	676	8	3	6	3
-0,751	0,705	638	8	3	7	3
4,774	8,104	670	8	3	9	3
4,53	6,928	698	8	3	10	3
1,375	4,896	614	8	3	13	3
-5,187	-3,738	624	8	3	15	3
6,456	8,917	518	8	3	16	3
-0,986	1,175	614	8	3	17	3
-0,964	2,545	598	8	3	18	3
0,528	2,38	694	8	3	19	3
-0,386	1,153	500	8	3	20	3
3,796	5,746	646	8	3	21	3
-1,079	0,777	700	8	3	2	4
0,918	3,318	620	8	3	3	4
-2,6	-1,046	594	8	3	4	4

1,233	4,571	502	8	3	5	4
0,993	4,593	672	8	3	6	4
0,101	2,04	652	8	3	7	4
6,028	7,907	500	8	3	9	4
4,357	7,323	688	8	3	10	4
-0,214	2,092	586	8	3	13	4
-8,269	-6,778	574	8	3	15	4
2,948	5,685	564	8	3	16	4
0,449	2,026	602	8	3	17	4
1,263	3,228	662	8	3	18	4
-4,376	-2,884	700	8	3	19	4
0,008	2,276	528	8	3	20	4
0,828	4,448	524	8	3	21	4
-0,493	1,01	650	8	3	2	5
1,111	2,274	632	8	3	3	5
-1,361	-0,631	612	8	3	4	5
0,912	3,925	522	8	3	5	5
0,068	2,964	648	8	3	6	5
-0,575	0,988	504	8	3	7	5
4,165	6,532	504	8	3	9	5
2,145	6,084	506	8	3	10	5
0,125	1,925	592	8	3	13	5
-4,319	-3,035	596	8	3	15	5
1,419	3,229	524	8	3	16	5
-1,451	1,155	604	8	3	17	5
-3,66	-0,336	598	8	3	18	5
0,719	1,441	578	8	3	19	5
-1,626	1,529	504	8	3	20	5
2,971	6,616	518	8	3	21	5
2,031	5,591	608	9	3	2	1
3,884	7,049	632	9	3	3	1
0,604	3,72	626	9	3	4	1
8,177	12,448	660	9	3	5	1
-2,95	0,68	642	9	3	6	1
1,702	4,873	694	9	3	7	1
2,399	5,18	656	9	3	9	1
0,19	5,304	626	9	3	10	1
2,862	6,878	610	9	3	13	1
9,873	12,742	638	9	3	15	1
-1,329	2,891	658	9	3	16	1
-1,591	3,301	618	9	3	17	1
3,605	8,546	622	9	3	18	1
1,47	7,288	654	9	3	19	1

2,76	5,271	700	9	3	20	1
-1,436	2,505	656	9	3	21	1
-1,024	2,858	610	9	3	2	2
2,029	6,204	638	9	3	3	2
0,619	5,868	618	9	3	4	2
1,551	4,098	632	9	3	5	2
-2,104	-0,026	648	9	3	6	2
-1,252	2,774	646	9	3	7	2
0,164	2,39	638	9	3	9	2
1,648	6,414	622	9	3	10	2
0,812	2,388	600	9	3	13	2
-12,39	-10,028	622	9	3	15	2
4,209	8,402	662	9	3	16	2
-2,458	1,806	624	9	3	17	2
-2,553	0,806	640	9	3	18	2
-0,079	5,134	668	9	3	19	2
2,079	4,215	636	9	3	20	2
0,244	4,141	636	9	3	21	2
-1,822	1,136	610	9	3	2	3
2,297	5,02	632	9	3	3	3
1,103	6,781	624	9	3	4	3
4,493	6,354	612	9	3	5	3
-5,37	-2,18	660	9	3	6	3
-1,18	3,768	672	9	3	7	3
2,158	5,14	664	9	3	9	3
-0,392	4,212	632	9	3	10	3
2,565	4,768	600	9	3	13	3
-13,036	-11,921	600	9	3	15	3
4,164	9,164	648	9	3	16	3
0,775	5,229	650	9	3	17	3
-1,95	4,027	648	9	3	18	3
7,243	16,9	694	9	3	19	3
3,38	6,195	672	9	3	20	3
-0,36	3,945	560	9	3	21	3
0,613	4,252	602	9	3	2	4
1,493	5,061	634	9	3	3	4
-1,178	2,727	664	9	3	4	4
1,579	5,092	620	9	3	5	4
-3,334	-0,749	644	9	3	6	4
-0,263	3,468	654	9	3	7	4
4,047	6,174	564	9	3	9	4
1,518	5,852	620	9	3	10	4
-0,843	1,456	670	9	3	13	4

-10,538	-8,489	614	9	3	15	4
0,555	4,834	650	9	3	16	4
-3,013	2,152	604	9	3	17	4
-1,264	2,771	650	9	3	18	4
-1,277	3,458	620	9	3	19	4
1,799	4,475	598	9	3	20	4
0,768	3,665	660	9	3	21	4
-0,381	3,089	616	9	3	2	5
0,717	2,884	646	9	3	3	5
1,33	4,928	630	9	3	4	5
2,534	4,568	638	9	3	5	5
-2,24	0,564	646	9	3	6	5
1,862	4,922	664	9	3	7	5
2,452	4,528	644	9	3	9	5
2,075	7,341	632	9	3	10	5
1,948	4,007	580	9	3	13	5
-8,022	-6,18	650	9	3	15	5
0,234	4,274	644	9	3	16	5
-0,821	4,051	636	9	3	17	5
0,636	5,473	666	9	3	18	5
3,202	6,416	668	9	3	19	5
0,296	2,409	668	9	3	20	5
2,324	6,134	548	9	3	21	5
-1,137	0,88	620	10	3	2	1
2,048	4,388	634	10	3	3	1
-0,066	1,616	638	10	3	4	1
5,189	8,395	656	10	3	5	1
-2,358	0,026	670	10	3	6	1
-0,991	1,206	696	10	3	7	1
-0,818	0,741	660	10	3	9	1
-2,178	0,925	618	10	3	10	1
0,344	3,464	610	10	3	13	1
12,557	14,767	694	10	3	15	1
-0,738	2,116	662	10	3	16	1
-1,698	1,575	592	10	3	17	1
1,692	5,842	622	10	3	18	1
4,153	6,624	686	10	3	19	1
0,763	2,585	680	10	3	20	1
-3,429	-0,684	690	10	3	21	1
-2,367	0,335	608	10	3	2	2
-0,078	2,311	640	10	3	3	2
-0,058	2,8	618	10	3	4	2
1,467	3,298	630	10	3	5	2

0,151	1,617	642	10	3	6	2
-2,877	-0,111	546	10	3	7	2
-2,822	-0,602	638	10	3	9	2
-1,065	1,456	622	10	3	10	2
-1,19	0,757	600	10	3	13	2
-11,595	-10,398	610	10	3	15	2
3,195	5,663	662	10	3	16	2
-3,35	-0,295	626	10	3	17	2
-3,336	-0,253	634	10	3	18	2
-2,962	-1,084	650	10	3	19	2
0,091	1,141	632	10	3	20	2
-2,932	-0,545	502	10	3	21	2
-3,231	-1,154	612	10	3	2	3
0,589	2,81	632	10	3	3	3
0,568	3,994	658	10	3	4	3
2,468	3,872	612	10	3	5	3
-1,915	0,875	646	10	3	6	3
-2,661	-0,202	664	10	3	7	3
0,166	2,875	650	10	3	9	3
-2,26	1,141	616	10	3	10	3
-2,004	2,678	594	10	3	13	3
-8,187	-6,3	672	10	3	15	3
2,546	5,733	650	10	3	16	3
-1,229	1,861	616	10	3	17	3
-1,238	2,543	648	10	3	18	3
-0,013	3,347	644	10	3	19	3
1,255	3,315	684	10	3	20	3
-1,451	1,142	544	10	3	21	3
-2,444	-0,387	600	10	3	2	4
0,119	3,247	650	10	3	3	4
-0,291	1,979	664	10	3	4	4
0,422	2,707	596	10	3	5	4
-1,462	1,624	648	10	3	6	4
-0,769	2,271	668	10	3	7	4
1,216	2,968	564	10	3	9	4
-1,694	1,072	624	10	3	10	4
-3,181	-1,335	686	10	3	13	4
-8,5	-6,837	584	10	3	15	4
0,397	2,681	650	10	3	16	4
-3,774	-0,668	630	10	3	17	4
-0,673	1,166	650	10	3	18	4
-4,361	-2,081	654	10	3	19	4
0,189	0,92	510	10	3	20	4

-1,765	0,789	662	10	3	21	4
-0,787	1,383	614	10	3	2	5
0,504	2,074	638	10	3	3	5
0,686	2,528	612	10	3	4	5
0,917	1,893	624	10	3	5	5
-0,469	1,941	648	10	3	6	5
-0,744	0,765	666	10	3	7	5
0,418	1,445	626	10	3	9	5
-0,295	2,993	616	10	3	10	5
-0,499	1,141	592	10	3	13	5
-6,895	-5,681	654	10	3	15	5
-0,148	2,104	644	10	3	16	5
-1,989	1,212	620	10	3	17	5
-0,957	1,937	678	10	3	18	5
2,313	4,095	700	10	3	19	5
-0,564	0,494	684	10	3	20	5
0,924	4,326	544	10	3	21	5
-2,315	-0,952	622	11	3	2	1
1,752	3,814	634	11	3	3	1
-0,656	0,382	610	11	3	4	1
5,585	7,997	658	11	3	5	1
-1,677	1,474	670	11	3	6	1
-1,782	0,55	528	11	3	7	1
0,582	2,444	660	11	3	9	1
-1,719	-0,192	670	11	3	10	1
1,322	4,158	610	11	3	13	1
10,858	12,415	694	11	3	15	1
-2,44	0,445	524	11	3	16	1
-0,655	2,008	596	11	3	17	1
1,518	5,68	620	11	3	18	1
2,496	4,809	686	11	3	19	1
-0,271	1,515	514	11	3	20	1
-3,157	1,048	690	11	3	21	1
-2,983	-1,1	636	11	3	2	2
-0,288	1,236	638	11	3	3	2
-0,512	1,704	630	11	3	4	2
2,148	3,558	654	11	3	5	2
-0,083	1,151	666	11	3	6	2
-4,017	-1,686	546	11	3	7	2
-1,03	1,299	694	11	3	9	2
-0,96	1,64	500	11	3	10	2
0,611	2,196	634	11	3	13	2
-10,934	-10,086	608	11	3	15	2

2,506	4,774	526	11	3	16	2
-3,448	-0,864	626	11	3	17	2
-4,353	-1,777	632	11	3	18	2
-4,695	-1,69	500	11	3	19	2
-0,692	0,573	504	11	3	20	2
-3,098	1,51	502	11	3	21	2
-3,308	-1,055	692	11	3	2	3
0,718	2,57	632	11	3	3	3
-0,175	2,543	668	11	3	4	3
1,891	3,496	674	11	3	5	3
-0,455	2,507	646	11	3	6	3
-3,404	-1,725	666	11	3	7	3
2,161	5,461	668	11	3	9	3
-1,192	0,981	616	11	3	10	3
-1,403	3,046	596	11	3	13	3
-7,712	-5,439	672	11	3	15	3
2,278	4,244	660	11	3	16	3
-1,811	0,874	618	11	3	17	3
-0,483	2,643	696	11	3	18	3
0,31	3,028	694	11	3	19	3
0,363	2,536	684	11	3	20	3
0,166	2,528	510	11	3	21	3
-3,042	-1,535	688	11	3	2	4
0,05	2,742	650	11	3	3	4
-0,394	1,383	664	11	3	4	4
-0,13	1,581	654	11	3	5	4
-1,357	1,839	648	11	3	6	4
-1,309	1,386	668	11	3	7	4
2,606	4,766	666	11	3	9	4
-0,959	1,47	674	11	3	10	4
-2,475	-0,697	686	11	3	13	4
-9,362	-7,668	576	11	3	15	4
-0,354	2,01	516	11	3	16	4
-3,358	-0,827	636	11	3	17	4
-1,335	0,698	700	11	3	18	4
-4,075	-1,905	686	11	3	19	4
0,275	2,861	510	11	3	20	4
-1,761	1,055	662	11	3	21	4
-1,031	0,259	614	11	3	2	5
0,633	2,012	636	11	3	3	5
0,379	1,73	612	11	3	4	5
0,803	1,726	502	11	3	5	5
-0,908	1,222	650	11	3	6	5

-2,399	-0,729	506	11	3	7	5
1,527	2,628	648	11	3	9	5
-0,196	2,07	662	11	3	10	5
0,276	1,443	592	11	3	13	5
-5,354	-4,487	694	11	3	15	5
-1,157	0,316	520	11	3	16	5
-2,528	-0,276	608	11	3	17	5
-2,346	0,615	682	11	3	18	5
1,11	1,597	568	11	3	19	5
-0,92	0,531	510	11	3	20	5
1,123	4,512	518	11	3	21	5
-3,308	-1,588	692	12	3	2	1
1,306	3,007	636	12	3	3	1
-0,375	0,469	608	12	3	4	1
6,015	8,353	654	12	3	5	1
-1,124	2,413	670	12	3	6	1
-2,418	0,69	510	12	3	7	1
1,735	4,142	510	12	3	9	1
-1,55	0,994	696	12	3	10	1
2,05	4,809	608	12	3	13	1
10,309	12,329	684	12	3	15	1
-2,954	2,761	508	12	3	16	1
0,779	3,024	598	12	3	17	1
1,483	5,346	620	12	3	18	1
4,309	7,079	686	12	3	19	1
-1,238	2,077	514	12	3	20	1
-2,647	2,157	690	12	3	21	1
-3,665	-1,641	646	12	3	2	2
-0,338	1,216	608	12	3	3	2
-0,813	0,751	632	12	3	4	2
2,85	4,725	686	12	3	5	2
0,227	2,461	676	12	3	6	2
-4,971	-0,794	508	12	3	7	2
0,352	3,58	694	12	3	9	2
-1,076	0,167	680	12	3	10	2
1,782	3,928	500	12	3	13	2
-8,901	-7,665	606	12	3	15	2
2,769	6,427	524	12	3	16	2
-2,873	-0,481	626	12	3	17	2
-5,091	-2,449	618	12	3	18	2
-5,194	-1,237	500	12	3	19	2
-0,764	1,427	506	12	3	20	2
-2,714	3,2	500	12	3	21	2

-3,344	-1,091	692	12	3	2	3
0,432	1,944	632	12	3	3	3
-0,205	2,455	668	12	3	4	3
1,171	3,201	678	12	3	5	3
-0,17	3,285	656	12	3	6	3
-3,655	-1,354	518	12	3	7	3
4,781	8,351	652	12	3	9	3
-0,721	2,099	666	12	3	10	3
0,199	4,719	614	12	3	13	3
-7,033	-4,536	692	12	3	15	3
3,161	5,562	516	12	3	16	3
-1,525	0,897	632	12	3	17	3
0,881	3,894	692	12	3	18	3
-0,146	2,256	694	12	3	19	3
-0,466	1,929	506	12	3	20	3
0,934	4,504	510	12	3	21	3
-2,999	-1,403	646	12	3	2	4
-0,098	2,05	622	12	3	3	4
0,471	1,976	664	12	3	4	4
-0,302	1,319	500	12	3	5	4
-0,473	2,77	648	12	3	6	4
-2,383	0,78	504	12	3	7	4
5,284	8,076	700	12	3	9	4
-0,548	3,211	674	12	3	10	4
-3,4	-2,132	604	12	3	13	4
-8,94	-7,258	576	12	3	15	4
-0,209	4,306	514	12	3	16	4
-2,604	-0,186	648	12	3	17	4
-1,028	0,835	698	12	3	18	4
-4,464	-2	520	12	3	19	4
0,184	3,954	510	12	3	20	4
-1,687	1,88	536	12	3	21	4
-1,482	-0,07	664	12	3	2	5
0,652	1,842	636	12	3	3	5
0,636	1,757	656	12	3	4	5
0,555	2,136	502	12	3	5	5
-1,269	0,86	658	12	3	6	5
-3,592	-0,517	506	12	3	7	5
2,903	4,171	662	12	3	9	5
-0,304	2,806	674	12	3	10	5
0,49	2,137	616	12	3	13	5
-4,913	-4,121	696	12	3	15	5
-0,984	2,636	510	12	3	16	5

-1,978	-0,319	600	12	3	17	5
-1,809	0,981	684	12	3	18	5
0,517	2,884	506	12	3	19	5
-1,399	1,385	510	12	3	20	5
0,908	5,722	516	12	3	21	5

Appendix F - Matlab® scripts

General Remark: It is necessary to create the Microsoft Excel® files data.xls, class.xls, and classL.xls (class.xls converted to binary format for Neural Network classifier) for each task, as described in the chapter III of this dissertation.

F.1 - Initial examination of the data - Scattering plotting of the features Mean Amplitude Between two fixed latencies, Peak Amplitude, and Peak Latency in matching pairs with their related result with classes.

a) Sentences

```
clc
close all

%Carregar base de dados
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');


% CLASSES:
[classes]=xlsread('class.xlsx','A1:A2880');

%Montar matriz de características:

caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];

X = caract(:,1:6);
Y = classes;
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

figure
gscatter(X(:,1),X(:,2),Y,'rgmbk','o+xd*',10);
h = legend('Class S1','Class S2','Class S3','Class S4','Class S5');
set(h,'FontSize',20)
title('Clustering for Sentences Task - Peak Amplitude x Mean Amplitude
Between two fixed latencies')
xlabel('Mean Amplitude Between two fixed latencies
(microvolts)','FontSize', 20);
ylabel('Peak Amplitude (microvolts)','FontSize', 20);
set(gca,'fontsize',20)

figure
gscatter(X(:,1),X(:,3),Y,'rgmbk','o+x*d',10);
h = legend('Class S1','Class S2','Class S3','Class S4','Class S5');
```

```
set(h,'FontSize',20)
title('Clustering for Sentences Task - Peak Latency x Mean Amplitude
Between two fixed latencies')
xlabel('Mean Amplitude Between two fixed latencies
(microvolts)','FontSize', 20);
ylabel('Peak Latency (miliseconds)','FontSize', 20);
set(gca,'fontsize',20)

figure
gscatter(X(:,2),X(:,3),Y,'rgmbk','o+x*d',10);
h = legend('Class S1','Class S2','Class S3','Class S4','Class S5');
set(h,'FontSize',20)
title('Clustering for Sentences Task - Peak Latency x Peak Amplitude')
xlabel('Peak Amplitude (microvolts)','FontSize', 20);
ylabel('Peak Latency (miliseconds)','FontSize', 20);
set(gca,'fontsize',20);
```

b) Words

```
clc
close all

%Carregar base de dados
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');


% CLASSES:
[classes]=xlsread('class.xlsx','A1:A2304');

%Montar matriz de características:

caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];

X = caract(:,1:6);
Y = classes;
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

figure
gscatter(X(:,1),X(:,2),Y,'rgmb','o+x*',10);
h = legend('Class S1','Class S2','Class S3','Class S4');
set(h,'FontSize',20);
title('Clustering for Words Task - Peak Amplitude x Mean Amplitude
Between two fixed latencies','FontSize', 20);
xlabel('Mean Amplitude Between two fixed latencies
(microvolts)','FontSize', 20);
ylabel('Peak Amplitude (microvolts)','FontSize', 20);
set(gca,'fontsize',20)

figure
```

```
gscatter(X(:,1),X(:,3),Y,'rgmb','o+x*',10);
h = legend('Class S1','Class S2','Class S3','Class S4');
set(h,'FontSize',20);
title('Clustering for Words Task - Peak Latency x Mean Amplitude
Between two fixed latencies')
xlabel('Mean Amplitude Between two fixed latencies
(microvolts)','FontSize', 20);
ylabel('Peak Latency (miliseconds)','FontSize', 20);
set(gca,'fontsize',20)

figure
gscatter(X(:,2),X(:,3),Y,'rgmb','o+x*',10);
h = legend('Class S1','Class S2','Class S3','Class S4');
set(h,'FontSize',20);
title('Clustering for Words Task - Peak Latency x Peak Amplitude')
xlabel('Peak Amplitude (microvolts)','FontSize', 20);
ylabel('Peak Latency (miliseconds)','FontSize', 20);
set(gca,'fontsize',20)
```

F.2 - Unsupervised pattern classification or clustering

F.2.1 - Hierarchical Clustering and Unsupervised Classifier

a) Sentences task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% Execute the classification
Y1 = pdist(X,'euclidean');
% pdist options
% 'euclidean'
% 'squaredeuclidean'
% 'seuclidean'
% 'cityblock'
% 'chebychev'
```

```
% 'mahalanobis'
% 'cosine'
% 'correlation'
% 'spearman'
% 'hamming'
% 'jaccard'

Z11 = linkage(Y1,'average');
% linkage options
% 'average'
% 'centroid'
% 'complete'
% 'median'
% 'single'
% 'ward'
% 'weighted'



C11 = cluster(Z11,'maxclust',5);
% 'maxclust' constructs a maximum of n clusters using the 'distance'
% criterion. cluster finds the smallest height at which a horizontal
% cut through the tree leaves n or fewer clusters.

% Print the Confusion Matrix and the accuracies of the classifier

[ConfusionMat11,labels] = confusionmat(Y,C11);
acc11 = 100*sum(diag(ConfusionMat11))./sum(ConfusionMat11(:));
fprintf('Hierarchical Cluster (pdist metric: euclidian Linkage Method:
average):\naccuracy = %.2f%%\n', acc11);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat11(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat11(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat11(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat11(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat11(5,:));
fprintf ('\n');

% Plot the Dendogram
Figure
dendrogram(Z1)
title ('Dendogram for Sentences - Hierarchical Cluster (pdist metric:
euclidian Linkage Method: average)
ylabel ('Height between Leaf Nodes')
xlabel ('Leaf Node')
```

b) Words task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')

% Execute the classification
Y1 = pdist(X,'euclidean');
% pdist options
% 'euclidean'
% 'squaredeuclidean'
% 'seuclidean'
% 'cityblock'
% 'chebychev'
% 'mahalanobis'
% 'cosine'
% 'correlation'
% 'spearman'
% 'hamming'
% 'jaccard'

Z11 = linkage(Y1,'average');
% linkage options
% 'average'
% 'centroid'
% 'complete'
% 'median'
% 'single'
% 'ward'
% 'weighted'

C11 = cluster(Z11,'maxclust',4);
% 'maxclust' constructs a maximum of n clusters using the 'distance'
% (pdist)criterion. cluster finds the smallest height at which a
% horizontalmcut through the tree leaves n or fewer clusters.

% Print the Confusion Matrix and the accuracies of the classifier
```

```
[ConfusionMat11,labels] = confusionmat(Y,C11);
acc11 = 100*sum(diag(ConfusionMat11))./sum(ConfusionMat11(:));
fprintf('Hierarchical Cluster (pdist metric: euclidian Linkage Method:
average):\naccuracy = %.2f%%\n', acc11);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat11(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat11(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat11(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat11(4,:));
fprintf ('\n');

% Plot the Dendogram
Figure
dendrogram(Z1)
title ('Dendogram for Words - Hierarchical Cluster (pdist metric:
euclidian Linkage Method: average)
ylabel ('Height between Leaf Nodes')
xlabel ('Leaf Node')
```

F.2.2 - K-means Clustering and Unsupervised Classifier

a) Sentences task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% Silhouette plots for 2, 3, 4 and 5 clusters
rng default  % For reproducibility
```

```
% Silhouette plot for 2 clusters
idx2 = kmeans(X,2,'distance','correlation');

 % cluster by distance's metric options:
 %'sqEuclidean'
 %'cityblock'
 %'cosine'
 %'correlation'

Figure
subplot(4,1,1)
silhouette(X,idx2)
title('Cluster Sentences - k-means silhouette plot for 2 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 3 clusters
idx3 = kmeans(X,3,'distance','correlation');
subplot(4,1,2)
silhouette(X,idx3)
title('Cluster Sentences - k-means silhouette plot for 3 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 4 clusters
idx4 = kmeans(X,4,'distance','correlation');
subplot(4,1,3)
silhouette(X,idx4)
title('Cluster Sentences - k-means silhouette plot for 4 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 5 clusters
idx5 = kmeans(X,5,'distance','correlation');
subplot(4,1,4)
silhouette(X,idx5)
title('Cluster Sentences - k-means silhouette plot for 5 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Print the Confusion Matrix and the accuracies of the k-means
% classifiers for 2, 3, 4 and 5 clusters
[ConfusionMat11,labels1] = confusionmat(Y2,idx2);
acc11 = 100*sum(diag(ConfusionMat11))./sum(ConfusionMat11(:));
fprintf('K-means Sentences wit metric cityblock with 2
clusters:\naccuracy = %.2f%%\n', acc11);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \n');
fprintf('\t S1 | \t %d   \t %d  \n',ConfusionMat11(1,:));
fprintf('\t S2 | \t %d   \t %d  \n',ConfusionMat11(2,:));
```

```
fprintf ('\n');

[ConfusionMat12,labels2] = confusionmat(Y1,idx3);
acc12 = 100*sum(diag(ConfusionMat12))./sum(ConfusionMat12(:));
fprintf(' K-means Sentences wit metric sqEuclidean with 3
clusters:\naccuracy = %.2f%%\n', acc12);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \n',ConfusionMat12(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \n',ConfusionMat12(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \n',ConfusionMat12(3,:));
fprintf ('\n');

[ConfusionMat13,labels3] = confusionmat(Y,idx4);
acc13 = 100*sum(diag(ConfusionMat13))./sum(ConfusionMat13(:));
fprintf(' K-means Sentences wit metric sqEuclidean with 4
clusters:\naccuracy = %.2f%%\n', acc13);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(4,:));
fprintf ('\n');


[ConfusionMat14,labels4] = confusionmat(Y,idx5);
acc14 = 100*sum(diag(ConfusionMat14))./sum(ConfusionMat14(:));
fprintf(' K-means Sentences wit metric cityblock with 5
clusters:\naccuracy = %.2f%%\n', acc14);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(5,:));
fprintf ('\n');
```

b) Words task

```
clc
clear
close all
```

```
% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')


% Silhouette plots for 2, 3, 4 and 5 clusters
rng default  % For reproducibility

% Silhouette plot for 2 clusters
idx2 = kmeans(X,2,'distance','correlation');

 % cluster by distance's metric options:
 %'sqEuclidean'
 %'cityblock'
 %'cosine'
 %'correlation'



Figure
subplot(4,1,1)
silhouette(X,idx2)
title('Cluster Words - k-means silhouette plot for 2 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 3 clusters
idx3 = kmeans(X,3,'distance','correlation');
subplot(4,1,2)
silhouette(X,idx3)
title('Cluster Words - k-means silhouette plot for 3 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 4 clusters
idx4 = kmeans(X,4,'distance','correlation');
subplot(4,1,3)
silhouette(X,idx4)
```

```
title('Cluster Words - k-means silhouette plot for 4 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Silhouette plot for 5 clusters
idx5 = kmeans(X,5,'distance','correlation');
subplot(4,1,4)
silhouette(X,idx5)
title('Cluster Words - k-means silhouette plot for 5 clusters with
cluster metric correlation')
xlabel ('Silhouette Value')
ylabel ('Cluster')

% Print the Confusion Matrix and the accuracies of the k-means
% classifiers for 2, 3, 4 and 5 clusters
[ConfusionMat11,labels1] = confusionmat(Y2,idx2);
acc11 = 100*sum(diag(ConfusionMat11))./sum(ConfusionMat11(:));
fprintf('K-means Words with metric cityblock with 2
clusters:\naccuracy = %.2f%%\n', acc11);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \n');
fprintf('\t S1 | \t %d   \t %d  \n',ConfusionMat11(1,:));
fprintf('\t S2 | \t %d   \t %d  \n',ConfusionMat11(2,:));
fprintf ('\n');

[ConfusionMat12,labels2] = confusionmat(Y1,idx3);
acc12 = 100*sum(diag(ConfusionMat12))./sum(ConfusionMat12(:));
fprintf('K-means Words with metric sqEuclidean with 3
clusters:\naccuracy = %.2f%%\n', acc12);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \n',ConfusionMat12(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \n',ConfusionMat12(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \n',ConfusionMat12(3,:));
fprintf ('\n');

[ConfusionMat13,labels3] = confusionmat(Y,idx4);
acc13 = 100*sum(diag(ConfusionMat13))./sum(ConfusionMat13(:));
fprintf('K-means Words with metric sqEuclidean with 4
clusters:\naccuracy = %.2f%%\n', acc13);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d
\n',ConfusionMat13(4,:));
fprintf ('\n');

[ConfusionMat14,labels4] = confusionmat(Y,idx5);
```

```
acc14 = 100*sum(diag(ConfusionMat14))./sum(ConfusionMat14(:));
fprintf('K-means Words with metric cityblock with 5
clusters:\naccuracy = %.2f%%\n', acc14);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat14(5,:));
fprintf ('\n');
```

F.2.3 - Gaussian Mixture Models Clustering and Unsupervised Classifier

a) Sentences task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% Features and classes matrixes creation
caract1 =[MeanAmp2FixedLat,PeakAmp];
caract2 =[MeanAmp2FixedLat,PeakLat];
caract3 =[PeakAmp,PeakLat];

X1 = caract1(:,1:2);
X2 = caract2(:,1:2);
```

```
X3 = caract3(:,1:2);
Y = classes;

% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Sentences to MeanAmp2FixedLat and PeakAmp features
Figure;
subplot (2,1,1)
scatter(X1(:,1),X1(:,2),10,'ko')
options1 = statset('Display','final');
gm1 = fitgmdist(X1,5,'Options',options1);
hold on
ezcontour(@(MeanAmpTwoFixedLat,PeakAmp)pdf(gm1,[MeanAmpTwoFixedLat
PeakAmp]),[-200 200],[-200 250]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Sentences')

% Cluster indication
idx1 = cluster(gm1,X1);
cluster11 = (idx1 == 1); % |1| for cluster 1 membership
cluster21 = (idx1 == 2); % |2| for cluster 2 membership
cluster31 = (idx1 == 3); % |3| for cluster 3 membership
cluster41 = (idx1 == 4); % |4| for cluster 4 membership
cluster51 = (idx1 == 5); % |5| for cluster 5 membership

subplot (2,1,2)
gscatter(X1(:,1),X1(:,2),idx1,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Sentences')
xlabel('MeanAmpTwoFixedLa')
ylabel('PeakAmp')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster
5','Location','NorthWest');
hold off

% Print the Confusion Matrix and the accuracies of the GMM for
% Sentences with MeanAmp2FixedLat and PeakAmp features
[ConfusionMat1,labels1] = confusionmat(Y,idx1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Gaussian Mixture Models Cluster for Sentences -
MeanAmp2FixedLat and PeakAmp Attributes):\naccuracy = %.2f%%\n',
acc1);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(5,:));
fprintf ('\n');
```

```
% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Sentences to MeanAmp2FixedLat and PeakLat features
Figure;
subplot (2,1,1)
scatter(X2(:,1),X2(:,2),10,'ko')
options2 = statset('Display','final');
gm2 = fitgmdist(X2,5,'Options',options2);
hold on
ezcontour(@(MeanAmpTwoFixedLat,PeakLat)pdf(gm2,[MeanAmpTwoFixedLat
PeakLat]),[-250 250],[-200 900]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Sentences')

% Cluster indication
idx2 = cluster(gm2,X2);
cluster12 = (idx2 == 1); % |1| for cluster 1 membership
cluster22 = (idx2 == 2); % |2| for cluster 2 membership
cluster32 = (idx2 == 3); % |3| for cluster 3 membership
cluster42 = (idx2 == 4); % |4| for cluster 4 membership
cluster52 = (idx2 == 5); % |5| for cluster 5 membership
subplot (2,1,2)
gscatter(X2(:,1),X2(:,2),idx2,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Sentences')
xlabel('MeanAmpTwoFixedLat')
ylabel('PeakLat')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster
5','Location','NorthWest');
hold off

% Print the Confusion Matrix and the accuracies of the GMM for
% Sentences with MeanAmp2FixedLat and PeakLat features
[ConfusionMat2,labels2] = confusionmat(Y,idx2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Gaussian Mixture Models Cluster for Sentences -
MeanAmp2FixedLat and PeakLat Attributes):\naccuracy = %.2f%%\n',
acc2);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(5,:));
fprintf ('\n');

% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Sentences to PeakAmp and PeakLat features
Figure;
subplot (2,1,1)
```

```
scatter(X3(:,1),X3(:,2),10,'ko')
options3 = statset('Display','final');
gm3 = fitgmdist(X3,5,'Options',options3);
hold on
ezcontour(@(PeakAmp,PeakLat)pdf(gm3,[PeakAmp PeakLat]),[-250 250],[-
200 900]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Sentences')

% Cluster indication
idx3 = cluster(gm3,X3);
cluster13 = (idx3 == 1); % |1| for cluster 1 membership
cluster23 = (idx3 == 2); % |2| for cluster 2 membership
cluster33 = (idx3 == 3); % |3| for cluster 3 membership
cluster43 = (idx3 == 4); % |4| for cluster 4 membership
cluster53 = (idx3 == 5); % |5| for cluster 5 membership
subplot (2,1,2)
gscatter(X3(:,1),X3(:,2),idx3,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Sentences')
xlabel('PeakAmp')
ylabel('PeakLat')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster
5','Location','NorthWest');
hold off

% Print the Confusion Matrix and the accuracies of the GMM for
% Sentences with PeakAmp and PeakLat features
[ConfusionMat3,labels3] = confusionmat(Y,idx3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('Gaussian Mixture Models Cluster for Sentences - PeakAmp and
PeakLat Attributes):\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(5,:));
fprintf ('\n');
```

b) Words task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
```

```
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')

% Features and classes matrixes creation
caract1 =[MeanAmp2FixedLat,PeakAmp];
caract2 =[MeanAmp2FixedLat,PeakLat];
caract3 =[PeakAmp,PeakLat];
X1 = caract1(:,1:2);
X2 = caract2(:,1:2);
X3 = caract3(:,1:2);
Y = classes;



% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Words to MeanAmp2FixedLat and PeakAmp features
Figure;
subplot (2,1,1)
scatter(X1(:,1),X1(:,2),10,'ko')
options1 = statset('Display','final');
gm1 = fitgmdist(X1,4,'Options',options1);
hold on
ezcontour(@(MeanAmpTwoFixedLat,PeakAmp)pdf(gm1,[MeanAmpTwoFixedLat
PeakAmp]),[-200 200],[-200 250]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Words')

% Cluster indication
idx1 = cluster(gm1,X1);
cluster11 = (idx1 == 1); % |1| for cluster 1 membership
cluster21 = (idx1 == 2); % |2| for cluster 2 membership
cluster31 = (idx1 == 3); % |3| for cluster 3 membership
cluster41 = (idx1 == 4); % |4| for cluster 4 membership
subplot (2,1,2)
gscatter(X1(:,1),X1(:,2),idx1,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Words')
xlabel('MeanAmpTwoFixedLa')
ylabel('PeakAmp')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Location','NorthWest');
hold off
```

```
% Print the Confusion Matrix and the accuracies of the GMM for
% Sentences with MeanAmp2FixedLat and PeakAmp features
[ConfusionMat1,labels1] = confusionmat(Y,idx1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Gaussian Mixture Models Cluster for Words - MeanAmp2FixedLat
and PeakAmp Attributes):\naccuracy = %.2f%%\n', acc1);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(4,:));
fprintf ('\n');


% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Words to MeanAmp2FixedLat and PeakLat features
Figure;
subplot (2,1,1)
scatter(X2(:,1),X2(:,2),10,'ko')
options2 = statset('Display','final');
gm2 = fitgmdist(X2,4,'Options',options2);
hold on
ezcontour(@(MeanAmpTwoFixedLat,PeakLat)pdf(gm2,[MeanAmpTwoFixedLat
PeakLat]),[-250 250],[-200 900]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Words')



% Cluster indication
idx2 = cluster(gm2,X2);
cluster12 = (idx2 == 1); % |1| for cluster 1 membership
cluster22 = (idx2 == 2); % |2| for cluster 2 membership
cluster32 = (idx2 == 3); % |3| for cluster 3 membership
cluster42 = (idx2 == 4); % |4| for cluster 4 membership

subplot (2,1,2)
gscatter(X2(:,1),X2(:,2),idx2,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Words')
xlabel('MeanAmpTwoFixedLat')
ylabel('PeakLat')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Location','NorthWest');
hold off

% Print the Confusion Matrix and the accuracies of the GMM for
% Words with MeanAmp2FixedLat and PeakLat features
[ConfusionMat2,labels2] = confusionmat(Y,idx2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Gaussian Mixture Models Cluster for Words - MeanAmp2FixedLat
and PeakLat Attributes):\naccuracy = %.2f%%\n', acc2);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
```

```
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(4,:));
fprintf ('\n');

% Scatter Plot and Fitted Gaussian Mixture Models Contour for
% Words to PeakAmp and PeakLat features
Figure;
subplot (2,1,1)
scatter(X3(:,1),X3(:,2),10,'ko')
options3 = statset('Display','final');
gm3 = fitgmdist(X3,4,'Options',options3);
hold on
ezcontour(@(PeakAmp,PeakLat)pdf(gm3,[PeakAmp PeakLat]),[-250 250],[-
200 900]);
title('Scatter Plot and Fitted Gaussian Mixture Models Contour for
Words')

% Cluster indication
idx3 = cluster(gm3,X3);
cluster13 = (idx3 == 1); % |1| for cluster 1 membership
cluster23 = (idx3 == 2); % |2| for cluster 2 membership
cluster33 = (idx3 == 3); % |3| for cluster 3 membership
cluster43 = (idx3 == 4); % |4| for cluster 4 membership
subplot (2,1,2)
gscatter(X3(:,1),X3(:,2),idx3,'rbgc','+ox*');
title('Gaussian Mixture Models clustering for Words')
xlabel('PeakAmp')
ylabel('PeakLat')
legend('Cluster 1','Cluster 2','Cluster 3','Cluster
4','Location','NorthWest');
hold off



% Print the Confusion Matrix and the accuracies of the GMM for
% Words with PeakAmp and PeakLat features
[ConfusionMat3,labels3] = confusionmat(Y,idx3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('Gaussian Mixture Models Cluster for Words - PeakAmp and
PeakLat Attributes):\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(4,:));
fprintf ('\n');
```

F.3 - Apply discrimination (Supervised Classification)

F.3.1 - Naïve Bayes Supervised Classifier

a) Sentences Task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:); caract(2305:2496,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:); classes(2305:2496,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% Naive Bayesian Network classifier for the training
NBModel1 = fitNaiveBayes(Xtr,Ytr,...
   'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});

% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});
```

```
% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Training
predictLabels1 = predict(NBModel1,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);

acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Naïve Bayes Network MVMN Sentences training:\naccuracy =
%.2f%%\n', acc1);
fprintf('Confusion Matrix for the training\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(5,:));
fprintf ('\n');

% VALIDATION SET (1/3 of the total dataset)
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:); caract(2497:2688,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:); classes(2497:2688,:)];
% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');

% Naive Bayesian Network classifier for the validation
NBModel2 = fitNaiveBayes(Xval,Yval,...
   'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});

% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Validation
predictLabels2 = predict(NBModel2,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Naïve Bayes Network MVMN Sentences validation:\naccuracy =
%.2f%%\n', acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
```

```
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(5,:));
fprintf ('\n');

% TEST SET (1/3 of the total dataset)

x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:); caract(2689:2880,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:); classes(2689:2880,:)];
% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');

% Naive Bayesian Network classifier for the Test
NBModel3 = fitNaiveBayes(Xtst,Ytst,...
       'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});

% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Test
predictLabels3 = predict(NBModel3,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('Naïve Bayes Network MVMN Sentences test:\naccuracy =
%.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(5,:));
```

```
fprintf ('\n');

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Total dataset
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('Naïve Bayes Network MVMN Sentences Total:\naccuracy =
%.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(5,:));
fprintf ('\n');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
```

```
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);

% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'Naïve Bayes Classification with Multivariate
Multinomial (MVMN) distribution for
Sentences','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
h.YTickLabel = [num2cell(isLabels); {''}];
hold off
```

```
% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for Naïve
Bayes Classification with MVMN distribution for
Sentences','FontSize',12,'FontWeight','bold')
hold off
```

b) Words Task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% Naive Bayesian Network classifier for the training
NBModel1 = fitNaiveBayes(Xtr,Ytr,...
   'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
```

```
% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Training
predictLabels1 = predict(NBModel1,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Naïve Bayes Network MVMN Words training:\naccuracy =
%.2f%%\n', acc1);
fprintf('Confusion Matrix for the training\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(4,:));
fprintf ('\n');

% VALIDATION SET (1/3 of the total dataset)
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:)];

% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');

% Naive Bayesian Network classifier for the validation
NBModel2 = fitNaiveBayes(Xval,Yval,...
   'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});

% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Validation
predictLabels2 = predict(NBModel2,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Naïve Bayes Network MVMN Words validation:\naccuracy =
%.2f%%\n', acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat2(1,:));
```

```
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(4,:));
fprintf ('\n');

% TEST SET (1/3 of the total dataset)

x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:)];

% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');



% Naive Bayesian Network classifier for the Test
NBModel3 = fitNaiveBayes(Xtst,Ytst,...
       'Distribution',{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});

% Distribution function options for each feature:
%{'normal','normal','normal','normal','normal','normal'});
%{'kernel','kernel','kernel','kernel','kernel','kernel'});
%{'mvmn','mvmn','mvmn','mvmn','mvmn','mvmn'});
%{'mn','mn','mn','mn','mn','mn'});

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Test
predictLabels3 = predict(NBModel3,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('Naïve Bayes Network MVMN Words test:\naccuracy = %.2f%%\n',
acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(4,:));
fprintf ('\n');

% Print the Confusion Matrix and the accuracies of the Naïve Bayes
% Network for Total dataset
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('Naïve Bayes Network MVMN Sentences Total:\naccuracy =
%.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(2,:));
```

```
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(4,:));
fprintf ('\n');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');


% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
```

```
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Naïve Bayes Network for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);


% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'Naïve Bayes Classification with Multivariate
Multinomial (MVMN) distribution for
Words','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
h.YTickLabel = [num2cell(isLabels); {''}];
hold off

% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for Naïve
Bayes Classification with (MVMN) distribution for
Words','FontSize',12,'FontWeight','bold')
hold off
```

F.3.2 - Multiclass Support Vector Machine Supervised Classifier

a) Sentences Task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:); caract(2305:2496,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:); classes(2305:2496,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% ECOC SVM classifier for the training
t1 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
ecocModel1 = fitcecoc(Xtst,Ytst,'Learners',t1);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.html

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Training
predictLabels1 = predict(ecocModel1,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('ECOC SVM Sentences for training:\naccuracy = %.2f%%\n',
acc1);
fprintf('Confusion Matrix for the training\n');
```

```
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(5,:));
fprintf ('\n');

% VALIDATION SET (1/3 of the total dataset)
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:); caract(2497:2688,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:); classes(2497:2688,:)];
% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');

% ECOC SVM classifier for the validation
t2 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
ecocModel2 = fitcecoc(Xtst,Ytst,'Learners',t2);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.html

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Validation
predictLabels2 = predict(ecocModel2,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('ECOC SVM Sentences for validation:\naccuracy = %.2f%%\n',
acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(5,:));
fprintf ('\n');
```

```
% TEST SET (1/3 of the total dataset)

x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:); caract(2689:2880,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:); classes(2689:2880,:)];

% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');

% ECOC SVM classifier for the validation
t3 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
ecocModel3 = fitcecoc(Xtst,Ytst,'Learners',t3);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Test
predictLabels3 = predict(ecocModel3,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('ECOC SVM Sentences for test:\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(5,:));
fprintf ('\n');

% Print the Confusion Matrix and the accuracies of the ECOC SVM Total
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('ECOC SVM Sentences for Total:\naccuracy = %.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(3,:));
```

```
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(5,:));
fprintf ('\n');



% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
```

```
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);

% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'ECOC SVM classifier for
Sentences','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
h.YTickLabel = [num2cell(isLabels); {''}];
hold off

% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for ECOC SVM
for Sentences','FontSize',12,'FontWeight','bold')
hold off
```

b) Words Task

```
clc
clear
close all
```

```
% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');


% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% ECOC SVM classifier for the training
t1 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
ecocModel1 = fitcecoc(Xtst,Ytst,'Learners',t1);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.html

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Training
predictLabels1 = predict(ecocModel1,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('ECOC SVM Words for training:\naccuracy = %.2f%%\n', acc1);
fprintf('Confusion Matrix for the training\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(4,:));
fprintf ('\n');
```

```
% VALIDATION SET (1/3 of the total dataset)
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:)];

% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');


% ECOC SVM classifier for the validation
t2 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
ecocModel2 = fitcecoc(Xtst,Ytst,'Learners',t2);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.html

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Validation
predictLabels2 = predict(ecocModel2,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('ECOC SVM Words for validation:\naccuracy = %.2f%%\n', acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(4,:));
fprintf ('\n');

% TEST SET (1/3 of the total dataset)
x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:)];

% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');

% ECOC SVM classifier for the validation
t3 =
templateSVM('BoxConstraint',0.01,'KernelFunction','gaussian','Standard
ize','off');
```

```
ecocModel3 = fitcecoc(Xtst,Ytst,'Learners',t3);
% vide templateSVM options on
% https://www.mathworks.com/help/stats/templatesvm.

% Print the Confusion Matrix and the accuracies of the ECOC SVM
% for Test
predictLabels3 = predict(ecocModel3,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('ECOC SVM Words for test:\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(4,:));
fprintf ('\n');


% Print the Confusion Matrix and the accuracies of the ECOC SVM Total
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('ECOC SVM Words for Total:\naccuracy = %.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(4,:));
fprintf ('\n');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);
```

```
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);


% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the ECOC SVM for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);

% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
```

```
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'ECOC SVM classifier for
Words','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
h.YTickLabel = [num2cell(isLabels); {''}];
hold off

% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for ECOC SVM
for Words','FontSize',12,'FontWeight','bold')
hold off
```

F.3.3 - Neural Network Supervised Classifier

a) Sentences and Words Task (it is necessary to load the respectively Microsoft Excel® files data.xls and classL.xls for each task.

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created Mon Oct 24 15:59:18 BRST 2016
%
% This script assumes these variables are defined:
%
%   data - input data.
%   target - target data.

clc;
clear;

[features] = xlsread('data.xlsx');
[classes]=xlsread('classL.xlsx');

data = features; % data - input data.
target = classes;% target - target data.

x = data';
t = target';

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess

% net.input.processFcns and net.output.processFcns options:

%net.input.processFcns = {'removeconstantrows','mapminmax'};
%net.output.processFcns = {'removeconstantrows','mapminmax'};

net.input.processFcns = {'removeconstantrows','mapstd'};
net.output.processFcns = {'removeconstantrows','mapstd'};


%net.input.processFcns = {'removeconstantrows','fixunknowns'};
%net.output.processFcns = {'removeconstantrows','fixunknowns'};

%net.input.processFcns = {'removeconstantrows','processpca'};
%net.output.processFcns = {'removeconstantrows','processpca'};

%net.input.processFcns = {'removerows','mapminmax'};
%net.output.processFcns = {'removerows','mapminmax'};

%net.input.processFcns = {'removerows','mapstd'};
%net.output.processFcns = {'removerows','mapstd'};
```

```
%net.input.processFcns = {'removerows','fixunknowns'};
%net.output.processFcns = {'removerows','fixunknowns'};

%net.input.processFcns = {'removerows','processpca'};
%net.output.processFcns = {'removerows','processpca'};


% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
% net.divideFcn options:
%dividerand Divide the data randomly (default)
%divideblock Divide the data into contiguous blocks
%divideint Divide the data using an interleaved selection
%divideind Divide the data by index


net.divideMode = 'sample';  % Divide up every sample
% net.divideMode options:
%'sample' for static networks
% 'time' for dynamic networks
% 'sampletime' to divide targets by both sample and timestep
% 'all' to divide up targets by every scalar value
%'none' to not divide up data at all (in which case all data is used
for training, none for validation or testing)

% Split of the sets
net.divideParam.trainRatio = 1/3;
net.divideParam.valRatio = 1/3;
net.divideParam.testRatio = 1/3;

% For help on training function 'trainscg' type: help trainscg
% For a list of all training functions type: help nntrain
net.trainFcn = 'trainrp';  % Scaled conjugate gradient

% net.trainFcn options:
%trainlm   - Levenberg-Marquardt backpropagation.
%trainbr   - Bayesian Regulation backpropagation.
%trainbfg  - BFGS quasi-Newton backpropagation.
%traincgb  - Conjugate gradient backpropagation with Powell-Beale
%restarts.
%traincgf  - Conjugate gradient backpropagation with Fletcher-Reeves
%updates.
%traincgp  - Conjugate gradient backpropagation with Polak-Ribiere
%updates.
%traingd   - Gradient descent backpropagation.
%traingda  - Gradient descent with adaptive lr backpropagation.
%traingdm  - Gradient descent with momentum.
%traingdx  - Gradient descent w/momentum & adaptive lr
backpropagation.
%trainoss  - One step secant backpropagation.
```

```
%trainrp   - RPROP backpropagation.
%trainscg  - Scaled conjugate gradient backpropagation.
%trainb    - Batch training with weight & bias learning rules.
%trainc    - Cyclical order weight/bias training.
%trainr    - Random order weight/bias training.
%trains    - Sequential order weight/bias training.

%trainbu   - Unsupervised batch training with weight & bias learning
%rules.
%trainru   - Unsupervised random order weight/bias training.

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
%net.performFcn = 'crossentropy';
%net.performFcn ='mae';
net.performFcn ='mse';
%net.performFcn ='sae';
%net.performFcn ='sse';

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression','plotroc', 'plotconfusion', 'plotfit'};


% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
performance = perform(net,t,y);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t  .* tr.valMask{1};
testTargets = t  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y);
valPerformance = perform(net,valTargets,y);
testPerformance = perform(net,testTargets,y);

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%Figure, plotperform(tr)
%Figure, plottrainstate(tr)
%Figure, plotconfusion(t,y)
%Figure, plotroc(t,y)
```

```
%Figure, ploterrhist(e)

% Deployment
% Change the (false) values to (true) to enable the following code
blocks.
if (false)
  % Generate MATLAB function for neural network for application
deployment
  % in MATLAB scripts or with MATLAB Compiler and Builder tools, or
simply
  % to examine the calculations your trained neural network performs.
  genFunction(net,'myNeuralNetworkFunction');
  y = myNeuralNetworkFunction(x);
end
if (false)
  % Generate a matrix-only MATLAB function for neural network code
  % generation with MATLAB Coder tools.
  genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
  y = myNeuralNetworkFunction(x);
end
if (false)
  % Generate a Simulink diagram for simulation or deployment with.
  % Simulink Coder tools.
  gensim(net);
end
```

F.3.4 - Random Forest Supervised Classifier

a) Sentences Task

```
clc
clear
close all

% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2880');
[PeakAmp]=xlsread('data.xlsx','B1:B2880');
[PeakLat]=xlsread('data.xlsx','C1:C2880');
[ROI]=xlsread('data.xlsx','D1:D2880');
[range]=xlsread('data.xlsx','E1:E2880');
[subject]=xlsread('data.xlsx','F1:F2880');
[classes]=xlsread('class.xlsx','A1:A2880');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#BASE DE DADOS TOTAL\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:); caract(2305:2496,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:); classes(2305:2496,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% Random Forest classifier for the training
rng(1) % For reproducibility
Btr = fitensemble(Xtr,Ytr,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the Random Forest
% for Training
predictLabels1 =  predict(Btr,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Random Forest Sentences for training:\naccuracy = %.2f%%\n',
acc1);
fprintf('Confusion Matrix for the training\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
```

```
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat1(5,:));
fprintf ('\n');

% VALIDATION SET (1/3 of the total dataset)
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:); caract(2497:2688,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:); classes(2497:2688,:)];
% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');

% Random Forest classifier for the validation
Bval =
fitensemble(Xval,Yval,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the random Forest
% for Validation
predictLabels2 =  predict(Bval,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Random Forest Sentences for validation:\naccuracy =
%.2f%%\n', acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat2(5,:));
fprintf ('\n');

% TEST SET (1/3 of the total dataset)
```

```
x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:); caract(2689:2880,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:); classes(2689:2880,:)];



% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');

% Random Forest classifier for the test
Btst =
fitensemble(Xtst,Ytst,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the Random Forest
% for Test
predictLabels3 =  predict(Btst,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('ECOC SVM Sentences for test:\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(4,:));
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat3(5,:));
fprintf ('\n');

% Print the Confusion Matrix and the accuracies of the Random Forest
% Total
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('ECOC SVM Sentences for Total:\naccuracy = %.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \t S5 \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(4,:));
```

```
fprintf('\t S5 | \t %d   \t %d  \t %d  \t %d  \t %d
\n',ConfusionMat4(5,:));
fprintf ('\n');


% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
```

```
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];



% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);

% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'Random Forest classifier for
Sentences','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
h.YTickLabel = [num2cell(isLabels); {''}];
hold off

% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for Random
Forest for Sentences','FontSize',12,'FontWeight','bold')
hold off
```

b) Words Task

```
clc
clear
close all
```

```
% Load dataset
[MeanAmp2FixedLat]=xlsread('data.xlsx','A1:A2304');
[PeakAmp]=xlsread('data.xlsx','B1:B2304');
[PeakLat]=xlsread('data.xlsx','C1:C2304');
[ROI]=xlsread('data.xlsx','D1:D2304');
[range]=xlsread('data.xlsx','E1:E2304');
[subject]=xlsread('data.xlsx','F1:F2304');
[classes]=xlsread('class.xlsx','A1:A2304');

% Create the features and classes matrixes
caract =[MeanAmp2FixedLat,PeakAmp,PeakLat,ROI,range,subject];
X = caract(:,1:6);
Y = classes;

% Print the total dataset
fprintf ('#TOTAL DATASET:\n');
tabulate(Y);
fprintf ('\n')

% TRAINING SET (1/3 of the total dataset)
x_tr = [caract(1:192,:); caract(577:768,:); caract(1153:1344,:);
caract(1729:1920,:)];
y_tr = [classes(1:192,:); classes(577:768,:); classes(1153:1344,:);
classes(1729:1920,:)];
% Print the training dataset
Xtr = x_tr(:,1:6);
Ytr = y_tr;
fprintf ('#TREINO \n');
tabulate(Ytr);
fprintf ('\n');

% Random Forest classifier for the training
rng(1) % For reproducibility
Btr = fitensemble(Xtr,Ytr,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the Random Forest
% for Training
predictLabels1 =  predict(Btr,Xtr);
[ConfusionMat1,labels1] = confusionmat(Ytr,predictLabels1);
acc1 = 100*sum(diag(ConfusionMat1))./sum(ConfusionMat1(:));
fprintf('Random Forest Words for training:\naccuracy = %.2f%%\n',
acc1);
fprintf('Confusion Matrix for the training\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat1(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat1(4,:));
fprintf ('\n');

% VALIDATION SET (1/3 of the total dataset)
```

```
x_val = [caract(193:384,:); caract(769:960,:); caract(1345:1536,:);
caract(1921:2112,:)];
y_val = [classes(193:384,:); classes(769:960,:); classes(1345:1536,:);
classes(1921:2112,:)];

% Print the validation dataset
Xval = x_val(:,1:6);
Yval = y_val;
fprintf ('#VALIDAÇAO\n');
tabulate(Yval);
fprintf ('\n');

% Random Forest classifier for the validation
Bval =
fitensemble(Xval,Yval,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the random Forest
% for Validation
predictLabels2 =  predict(Bval,Xval);
[ConfusionMat2,labels2] = confusionmat(Yval,predictLabels2);
acc2 = 100*sum(diag(ConfusionMat2))./sum(ConfusionMat2(:));
fprintf('Random Forest Words for validation:\naccuracy = %.2f%%\n',
acc2);
fprintf('Confusion Matrix for the validation\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat2(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat2(4,:));
fprintf ('\n');

% TEST SET (1/3 of the total dataset)
x_tst = [caract(385:576,:); caract(961:1152,:); caract(1537:1728,:);
caract(2113:2304,:)];
y_tst = [classes(385:576,:); classes(961:1152,:);
classes(1537:1728,:); classes(2113:2304,:)];

% Print the Test dataset
Xtst = x_tst(:,1:6);
Ytst = y_tst;
fprintf ('#TESTE\n');
tabulate(Ytst);
fprintf ('\n');

% Random Forest classifier for the test
Btst =
fitensemble(Xtst,Ytst,'bag',100,'Tree','Type','classification');
% vide fitensemble options on
% https://www.mathworks.com/help/stats/fitensemble.html

% Print the Confusion Matrix and the accuracies of the Random Forest
% for Test
```

```
predictLabels3 =  predict(Btst,Xtst);
[ConfusionMat3,labels3] = confusionmat(Ytst,predictLabels3);
acc3 = 100*sum(diag(ConfusionMat3))./sum(ConfusionMat3(:));
fprintf('Random Forest Words for test:\naccuracy = %.2f%%\n', acc3);
fprintf('Confusion Matrix for the test\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat3(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat3(4,:));
fprintf ('\n');

% Print the Confusion Matrix and the accuracies of the ECOC SVM Total
ConfusionMat4 = ConfusionMat1 + ConfusionMat2 + ConfusionMat3;
acc4 = 100*sum(diag(ConfusionMat4))./sum(ConfusionMat4(:));
fprintf('Random Forest for Total:\naccuracy = %.2f%%\n', acc4);
fprintf('Confusion Matrix Total\n');
fprintf('\t T  | \t S1 \t S2  \t S3  \t S4  \n');
fprintf('\t S1 | \t %d \t %d  \t %d  \t %d  \n',ConfusionMat4(1,:));
fprintf('\t S2 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(2,:));
fprintf('\t S3 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(3,:));
fprintf('\t S4 | \t %d   \t %d  \t %d  \t %d  \n',ConfusionMat4(4,:));
fprintf ('\n');


% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Training dataset
[n,p] = size(Xtr);
isLabels = unique(Ytr);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels1,isLabels);
predictLabels1Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels1Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytr,isLabels);
YMatTr = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTr(idxLinearY) = 1;
etr = gsubtract(predictLabels1Mat',YMatTr');

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Validation dataset
[n,p] = size(Xval);
isLabels = unique(Yval);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels2,isLabels);
predictLabels2Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
```

```
predictLabels2Mat(idxLinear) = 1;
[~,grpY] = ismember(Yval,isLabels);
YMatVal = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatVal(idxLinearY) = 1;
eval = gsubtract(predictLabels2Mat,YMatVal);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Test dataset
[n,p] = size(Xtst);
isLabels = unique(Ytst);
nLabels = numel(isLabels);

% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels3,isLabels);
predictLabels3Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels3Mat(idxLinear) = 1;
 [~,grpY] = ismember(Ytst,isLabels);
YMatTst = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTst(idxLinearY) = 1;
etst = gsubtract(predictLabels3Mat,YMatTst);

% Convert dataset to plot the Confusion Matrix and the accuracies of
% the Random Forest for Total dataset
[n,p] = size(X);
isLabels = unique(Y);
nLabels = numel(isLabels);
predictLabels4 = [predictLabels1;predictLabels2;predictLabels3];
Ytot = [Ytr;Yval;Ytst];
% Convert the integer label vector to a class-identifier matrix.
[~,grpOOF] = ismember(predictLabels4,isLabels);
predictLabels4Mat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
% Flags the row corresponding to the class
predictLabels4Mat(idxLinear) = 1;
[~,grpY] = ismember(Ytot,isLabels);
YMatTot = zeros(nLabels,n);
idxLinearY = sub2ind([nLabels n],grpY,(1:n)');
YMatTot(idxLinearY) = 1;
etot = gsubtract(predictLabels4Mat,YMatTot);

% All Confusion Matrixes and Accuracies plot
t = 0;
Figure;
plotconfusion(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-12,-8.7,'Random Forest classifier for
Words','FontSize',12,'FontWeight','bold')
h = gca;
h.XTickLabel = [num2cell(isLabels); {''}];
```

```
h.YTickLabel = [num2cell(isLabels); {''}];
hold off

% All ROC plot
Figure;
plotroc(predictLabels1Mat,YMatTr,'Training',
predictLabels2Mat,YMatVal,'Validation',predictLabels3Mat,YMatTst,'Test
',predictLabels4Mat,YMatTot,'Total');
text(-2.30,2.55,'Receiver Operating Characteristic(ROC) for Random
Forest for Words','FontSize',12,'FontWeight','bold')
hold off
```