UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

Data Analysis using Sparse PCA

Trabajo de integración curricular presentado como requisito para la obtención del título de Matemático

Autor:

Narea Navarrete Fausto Alejandro

Tutor:

Amaro Martín Isidro Rafael, Ph.D.

Urcuquí, Mayo del 2023

Autoría

Yo, **FAUSTO ALEJANDRO NAREA NAVARRETE**, con cédula de identidad 0940286826, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, Mayo del 2023.

Firmado electrónicamente por:
FAUSTO ALEJANDRO NAREA NAVARRETE

Fausto Alejandro Narea Navarrete

CI: 0940286826

Autorización de publicación

Yo, **FAUSTO ALEJANDRO NAREA NAVARRETE**, con cédula de identidad 0940286826, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, Mayo del 2023.

Firmado electrónicamente por:
FAUSTO ALEJANDRO
NAREA NAVARRETE

Fausto Alejandro Narea Navarrete

CI:0940286826

Dedication

To my beloved wife and my beloved daughter.

Fausto Alejandro Narea Navarrete

Acknowledgments

Agradezco a mis familiares especialmente a mis padres, esposa, hija y a mis amigos Guido y Christian que estuvieron presentes durante este largo camino. También a todos mis profesores, en especial al gran Juan Mayorga, el gran Antonio Acosta y a los miembros del gremio de los estadísticos como el gran Isidro Amaro, el gran Saba Infante y el gran Diego Morales quienes me apoyaron mucho en la realización de este trabajo.

Fausto Alejandro Narea Navarrete

Resumen

En el campo de la investigación y la ciencia, los datos de estudio cada vez son más grandes, lo que conlleva a una difícil gestión de estos, es aquí donde surgen muchas técnicas de Análisis Multivariante que nos permiten gestionar estas bases de datos mediante la reducción de dimensión de estas. El método de reducción utilizado en este trabajo se denomina Análisis de Componentes Principales Sparse, el cual se encarga de obtener componentes principales cuya matriz de carga está mayoritariamente conformada por ceros, facilitando su interpretación.

Se aplicaron algunos algoritmos de este método a una base de datos de Pruebas Clínicas COVID-19 de la cual se obtuvo que de las 7 variables, 4 de ellas eran las más importantes ya que con ellas se alcanzaba alrededor del 91% de la varianza explicada. Finalmente, estos algoritmos fueron más efectivos que un PCA clásico ya que, debido a la forma de su matriz de carga, son más fáciles de interpretar. Además, estos no presentan dificultades a la hora de trabajar con outliers y, finalmente, presentan un bajo coste computacional.

Palabras Clave:
Big data, Sparse PCA, Análisis Multivariante, PCA, Shrinkage Methods.

Abstract

In the field of research and science, the study data is getting larger, which leads to difficult management of these, it is here where many Multivariate Analysis techniques arise that allow us to manage these databases by reducing of dimension of these. The reduction method used in this work is called Sparse Principal Component Analysis, which is responsible for obtaining principal components whose loadings matrix is mostly made up of zeros, facilitating its interpretation.

Some algorithms of this method were applied to a Clinical test COVID-19 database from which it was obtained that of the 7 variables, 4 of them were the most important since with them around 91% of the explained variance was reached. Finally, these algorithms were more effective than classic PCA since, due to the structure of their loadings matrix, they are easier to interpret. In addition, they do not present difficulties when working with outliers and, finally, they present a low computational cost.

Keywords:
Big data, Sparse PCA, Multivariate analysis, PCA, Shrinkage Methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

During the last decades, the databases used in many investigations have become more complex. In fact, new terms associated with this context have appeared, such as Big Data, Data Mining and Machine Learning, among others. Some examples of this reality are: in the area of Human Face Recognition [2], in Biomedical Research (gene expression data analysis) [3], in image processing [4], Time Series Analysis [5], etc.

Multivariate Statistical Dimension Reduction techniques have been widely used to handle, process and analyze large data sets. The main objective is to project the original data into a low-dimensional subspace so that it is possible to capture the greatest variability present in the data.

Principal Component Analysis (PCA), introduced by Karl Pearson [6] in 1901, is one of the most extensively used dimension reduction approaches. Jolliffe (2002) [7] is a more recent reference to the method. The PCA generates new variables, which are linear combinations of the original variables and are known as Principal Components (PC's). The coefficients of these linear combinations, known as loads (loadings), are frequently different from zero, which leads to the technique's fundamental disadvantage: the interpretation of the principal components.

There are many cases where the principal components can be easily interpreted, and many bibliographies about it. For example, Jolliffe [7] presented some examples where a straightforward interpretation is possible in chapter 4 of his book "Principal Component

Analysis".

Several alternatives have been proposed to improve the interpretation of the results of a PCA. Some suggest using Rotation Techniques [8] to simplify the structure of the principal components, but more is needed to solve the problem.

Other techniques propose the imposition of restrictions on the charges in the components. Among these strategies are the Regularization Techniques, also known as shrinkage methods. The main objective is to introduce penalties such that each component is a combination of only the relevant variables (making them null or almost zero) so that the interpretation of the results improves significantly. Hausman (1982) [9] proposed to restrict the values of the charges of the Principal Components to the set of integers -1, 0, 1. Vines [10] in 2000 suggested the use of arbitrary integers.

In this sense, Tibshirani [11] developed the Least Absolute Shrinkage and Selection Operator (LASSO) approach in 1996. In it, he paired a regression model with a technique for setting some parameters equal to zero, penalizing the regression coefficients.

Another type of penalty is known as Elastic Net for Regression, which combines Ridge Regularization Techniques (a technique often employed when the Multicollinearity problem occurs in Multiple Regression Analysis) with LASSO, and this was introduced in 2005 by Zou et al [12]. Based on the l_1 and l_2 norms[1], this technique penalizes the magnitude of the regression coefficients.

Zou et al in 2006 [2], proposed a penalty algorithm, called Sparse PCA, which applies Elastic Net regularization and LASSO penalty to efficiently solve the problem. Research on this topic is still open as evidenced by the large number of recent scientific articles reporting new theoretical developments and applications of these techniques.

In this thesis, theoretical aspects of Sparse PCA are studied and the technique is applied to a data set related to "Clinical Tests (COVID-19)".

[1] We consider $\mathbb{K}(\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C})$ and $p \in \mathbb{N}$

- $l^p(\mathbb{K}) = \left\{ x = (x_n)_{n\in\mathbb{N}} \subseteq \mathbb{K} \quad / \quad \sum_{n=1}^{+\infty} |x_n|^p < +\infty \right\}$.
- In $l^p(\mathbb{K})$, we consider the mapping $\|\cdot\|_p : l^p(\mathbb{R}) \to \mathbb{R}$ given by $\|x\|_p = \left(\sum_{n=1}^{+\infty} |x_n|^p \right)^{1/p}$.

1.2 Problem statement

Principal Component Analysis is a Multivariate Statistical technique widely used when one of the researcher's objectives is to reduce the data size. Despite this, this technique has problems with the interpretation of the Principal Components since they are linear combinations of the original variables, which are abstract mathematical concepts and, sometimes, need to be interpretable, or their interpretation requires deep knowledge of the field from which the data came.

Over the years, various techniques have been proposed to reduce these deficiencies. In this sense, in this thesis, the Sparse PCA method is studied, which forces each Principal Component to be a combination of only some of the original variables, hoping to interpret the results better.

In this thesis, we present Sparse Principal Components Analysis. This multivariate data analysis technique seeks to correct the basic deficiency of classic Principal Components Analysis, which is the interpretation of the results. Likewise, the technique is applied to a real-life data set. Therefore, this thesis contributes to explaining the theoretical basis of the technique and how to apply it to solve a particular problem.

1.3 Objectives

1.3.1 General Objective

Analyze the 'Sparse Principal Component Analysis' (Sparse PCA) Multivariate Statistical technique.

1.3.2 Specific Objectives

1. Show the state of the art of the Sparse PCA.
2. Explain the mathematical theory that supports the Sparse PCA.
3. Analyze and compare the results of the application of Sparse PCA to a real life data set with respect to the classic PCA.

Chapter 2

Theoretical Framework

In this chapter, we will address some definitions and properties of linear algebra and multivariate statistical analysis necessary for the understanding of principal components analysis.This section is based on the following references [1], [2],[7], [8], [13], [14],[15], [16], [17], [18],[19], [20],[21],[22],[23], and [24].

2.1 Matrix Algebra

2.1.1 Linear Combination

Let $\mathbf{V}$ be a vector space over a real number field $\mathbb{R}$. Assume S is a nonempty subset of V. Then a $\mathbf{v} \in V$ is said to be a linear combination of the set of vectors in S if S contains vectors $w_1, w_2, ..., w_n$ and scalars $\alpha_1, \alpha_2, ..., \alpha_n$ such that

$$v = \alpha_1 w_1 + \alpha_2 w_2 + \cdots + \alpha_n w_n$$

2.1.2 Eigenvalues

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, we say that $\lambda \in \mathbb{R}$ is an eigenvalue of $\mathbf{A}$ if and only if there is a nonzero vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ such that:

$$\mathbf{Av} = \lambda \mathbf{v} \qquad \mathbf{v} \neq \mathbf{0} \tag{2.1}$$

$\mathbf{v}$ is called the eigenvector associated with λ.

2.1.3 Eigenvectors

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$. Let's consider a nonzero vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$. We say that $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ if and only if there is $\lambda \in \mathbb{R}$ such that:

$$\mathbf{Av} = \lambda \mathbf{v} \qquad \mathbf{v} \neq \mathbf{0} \tag{2.2}$$

λ is called the eigenvalue associated with the eigenvector $\mathbf{v}$.

2.1.4 Orthonormal Basis

If $\mathbf{V}$ is a basis and all of its vectors have a norm of 1 and are pairwise orthogonal, V is said to be orthonormal.

2.1.5 Singular Values

If $\mathbf{A}$ is an $m \times n$ matrix, then the singular values of matrix $\mathbf{A}$ are said to be the square roots of eigenvalues of matrix $\mathbf{A}^T\mathbf{A}$, which is symmetric. These values are denoted as

$$\sigma_1, \sigma_2, ..., \sigma_n \tag{2.3}$$

and for ease are considered

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \tag{2.4}$$

.

2.1.6 Singular Value Decomposition(SVD)

Let $\mathbf{X}$ be a matrix of size $m \times n$ and $rank(\mathbf{X}) = r$, it is said to have a singular value decomposition (SVD) if there exists a diagonal matrix $\mathbf{D} \in \mathbb{R}^{r \times r}$ and the orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ such that

$$\mathbf{X} = \mathbf{UDV}^T \tag{2.5}$$

Proof. Let $\{v_t\}_{t=1}^{r} \subseteq \mathbb{R}^p$ be an orthonormal eigenbasis for $\mathbb{R}^p$ associated with singular values $\lambda_1, \lambda_2, \ldots, \lambda_r$ of $\mathbf{X}$ where

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0.$$

Moreover, we have that the sequence $\{u_l\}_{t=1}^{r} \subseteq \mathbb{R}^n$ where

$$\forall t \in \{1, \ldots, r\} : \quad u_t = \frac{\mathbf{X}v_t}{\lambda_t}$$

is an orthonormal basis for the column space of $\mathbf{X}$ where each u_t is an eigenvector of $\mathbf{X}\mathbf{X}^T$. It implies that

$$\forall t \in \{1, \ldots, r\} : \quad Xv_t = \lambda_t u_t. \tag{2.6}$$

Let $\mathbf{U} = [u_1 \cdots u_r]$ and $\mathbf{V} = [v_1 \cdots v_r]$. From (2.6) and the orthogonality of $\mathbf{U}$ and $\mathbf{V}$, it follows that

$$\mathbf{XV} = \begin{bmatrix} \mathbf{X}v_1 & \cdots & \mathbf{X}v_r \end{bmatrix} = \begin{bmatrix} \lambda_1 u_1 & \cdots & \lambda_r u_r \end{bmatrix}.$$

Also let $\Sigma = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$. Therefore,

$$\mathbf{U}\Sigma = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_r \end{bmatrix} = \begin{bmatrix} \lambda_1 u_1 & \cdots & \lambda_r u_r \end{bmatrix}$$

Therefore, we have hat $\mathbf{XV}=\mathbf{U}\Sigma$. Finally, by the orthogonality of $\mathbf{V}$, we have proved that

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{2.7}$$

. □

2.1.7 Frobenius Norm

Let $\mathbf{X}$ be an $m \times n$ matrix. We define the Frobenius norm $\|\cdot\|_F$ as the square root of the sum of the absolute squares of the elements $\mathbf{X}$ given by

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |x_{ij}|^2} \tag{2.8}$$

This can also be defined as the square root of the trace of $\mathbf{X^T X}$

$$\|\mathbf{X}\|_F := \sqrt{\text{tr}\left(\mathbf{X^T X}\right)}. \tag{2.9}$$

2.2 Principal Components

Principal Component Analysis (PCA) is a multivariate technique that consists of reducing the dimension of the data. It allows extracting n new variables, which we will call Principal Components, from m related variables, with $n << m$, which explain the behavior of the sample in a low-dimensional space. It should be noted that each Principal Component is a linear combination of all the original variables, which makes it difficult to interpret. It must be taken into account that if the original variables are not correlated with each other, then it would not make sense to carry out a PCA analysis.

2.2.1 PCA properties

According to Zou and Hastie (2005)[12], the maximum capture between the columns of X is sequentially captured by the principle components, ensuring that there is little information loss. Also, because major components are unrelated to one another, it is possible to discuss one without mentioning others. The PCA allows for the conversion of the initial, generally correlated variables into new, uncorrelated variables, which aids in the interpretation of the data. It is frequently challenging to interpret the findings of PCA because each principle component is a linear combination of all the original variables.

Maximizes Variability

The principal components are sought that are a linear combination of the original variables, in such a way that they are uncorrelated and that they preserve as much information as possible from the original data matrix $\mathbf{X}$.

Let's consider

$$\mathbf{z} = \mathbf{a}^T\mathbf{X} \tag{2.10}$$

with variance

$$Var(\mathbf{z}) = \mathbf{a}^T\mathbf{S}\mathbf{a}. \tag{2.11}$$

In order to maximize the variance, we need that $\mathbf{a}$ to have the normalization constraint, i.e.,

$$\mathbf{a}^T\mathbf{a} = 1 \tag{2.12}$$

First, the first component is calculated by choosing a_1 so that z_1 has the largest variance and maintaining the condition that $\mathbf{a_1}^T\mathbf{a_1} = 1$. Then

$$Var(z_1) = Var(\mathbf{a_1}^T\mathbf{X}) = \mathbf{a_1}^T\mathbf{S}\mathbf{a_1}. \tag{2.13}$$

Using the Lagrange multipliers to maximize (2.13), we get

$$\begin{aligned}
L(\mathbf{a}_1) &= \mathbf{a}_1^T\mathbf{S}\mathbf{a}_1 - \lambda\left(\mathbf{a}_1^T\mathbf{a}_1 - 1\right) \\
\frac{\partial\left(L(\mathbf{a}_1)\right)}{\partial\mathbf{a}_1} &= 2\mathbf{S}\mathbf{a}_1 - 2\lambda_1\mathbf{a}_1 \\
2\mathbf{S}\mathbf{a}_1 - 2\lambda_1\mathbf{a}_1 &= \mathbf{0} \\
\mathbf{S}\mathbf{a}_1 &= \lambda_1\mathbf{a}_1 \\
(\mathbf{S} - \lambda_1\mathbf{I})\,\mathbf{a}_1 &= \mathbf{0} \\
\mathbf{S}\mathbf{a}_1 - \lambda_1\mathbf{I}\mathbf{a}_1 &= 0 \\
\mathbf{S}\mathbf{a}_1 &= \lambda_1\mathbf{I}\mathbf{a}_1
\end{aligned}$$

Then

$$\begin{aligned}
\text{Var}(z_1) &= \text{Var}\left(\mathbf{a}_1^T\mathbf{x}\right) \\
&= \mathbf{a}_1^T\mathbf{S}\mathbf{a}_1 \\
&= \mathbf{a}_1^T\lambda_1\mathbf{I}\mathbf{a}_1 \\
&= \lambda_1\mathbf{a}_1^T\mathbf{a}_1 \\
&= \lambda_1 \cdot 1 \\
&= \lambda_1
\end{aligned} \tag{2.14}$$

where λ_1 is the largest eigenvalue associated to the eigenvector a_1, which maximizes the

variance of z_1. Therefore, the first principal component is given by

$$\begin{aligned} z_1 &= \mathbf{a}_1^T \mathbf{X} \\ &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p \end{aligned} \tag{2.15}$$

where a_1 corresponds to the eigenvector of $\mathbf{S}$ with the largest eigenvalue. In the same way, the second principal component

$$z_2 = \mathbf{a}_2^T \mathbf{X} \tag{2.16}$$

can be obtained using the same argument as for the first principal component. In addition to maintaining the condition $\mathbf{a_2}^T\mathbf{a_2} = 1.$, it must also be uncorrelated with z_1,i.e,

$$\begin{aligned} \text{Cov}\left(z_1, z_2\right) &= \text{Cov}\left(\mathbf{a}_1^T\mathbf{Y}, \mathbf{a}_2^T\mathbf{Y}\right) \\ &= \mathbf{a}_2^T\mathbf{S}\mathbf{a}_1 \\ &= \mathbf{a}_2^T\lambda_1\mathbf{a}_1 \\ &= \lambda_1\mathbf{a}_2^T\mathbf{a}_1 \\ &= 0 \end{aligned} \tag{2.17}$$

which implies that $\mathbf{a}_2$ and $\mathbf{a}_1$ are orthogonals and $\mathbf{a}_2'\mathbf{S}\mathbf{a}_1 = 0$. Seeking to maximize the variance, maintaining the conditions and applying two Lagrange multipliers, we get

$$\begin{aligned} L\left(\mathbf{a}_2\right) &= \mathbf{a}_2^T\mathbf{S}\mathbf{a}_1 - \lambda\left(\mathbf{a}_2^T\mathbf{a}_1 - 1\right) - \delta\mathbf{a}_2^T\mathbf{a}_1 \\ \frac{\partial\left(L\left(\mathbf{a}_2\right)\right)}{\partial\mathbf{a}_1} &= 2\mathbf{S}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_2 - \delta\mathbf{a}_1 \\ 2\mathbf{S}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_2 - \delta\mathbf{a}_1 &= \mathbf{0} \end{aligned} \tag{2.18}$$

Multiplying (2.18) by $\mathbf{a}_1^T$

$$2\mathbf{a}_1^T\mathbf{S}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_1^T\mathbf{a}_2 - \delta\mathbf{a}_1^T\mathbf{a}_1 = 0 \tag{2.19}$$

$$\tag{2.20}$$

by the conditions, we have

$$2\mathbf{a}_1^T\mathbf{S}\mathbf{a}_2 - \delta = 0 \tag{2.21}$$

$$\tag{2.22}$$

which implies that

$$\delta = 0$$

from (2.18)

$$2\mathbf{S}\mathbf{a}_2 - 2\lambda_2\mathbf{a}_2 = \mathbf{0}$$

$$\mathbf{S}\mathbf{a}_2 = \lambda_2\mathbf{a}_2$$

therefore, λ_2 is the second largest eigenvalue of $\mathbf{S}$ associated to the eigenvector a_2. In general, the p^{th} principal component is given by

$$\mathbf{z}_p = \mathbf{a}_p^T\mathbf{X}$$

whose variance is

$$\text{Var}\left(z_p\right) = \lambda_p.$$

Now, let's call

$$A_{p\times p} = \left[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p\right]$$

as the matrix of the eigenvector a_i, for $i = 1, 2, ...p$. Then, the vector of the principal components can be written as

$$\mathbf{Z}_{p\times 1} = \mathbf{A}_{p\times p}^T\mathbf{X}_{\mathbf{p}\times\mathbf{1}}$$

Singular Value Decomposition

Consider $\mathbf{X}$ the centred data matrix of size $m \times n$ where represents the number of observations and n the number of variables. The data matrix $\mathbf{X}$ can be expressed using its singular value decomposition (SVD), which gives us

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

where **Z=UD** is the principal component, **U** is a unit matrix $n \times n$, **D** is a diagonal matrix $n \times p$ which contains the eigenvalues of **X**, the columns of **V** are the loads of the principal components and the sample variance of the i^{th} principal component is given by:

$$\text{Var}(z_i) = \frac{\lambda_i}{n}.$$

Each principal component can be expressed in the form

$$z_i = u_i \lambda_i.$$

It is important to note that since the eigenvalues are ordered in decreasing order, the first component z_1 will have the largest eigenvalue λ_1 and also the largest variance among all normalized linear combinations of the columns of the **X** matrix.

Minimizes Error

The main purpose of this model is to estimate the load matrix that defines the principal components through error minimization; that is, minimizing the difference between the data of the original matrix, and the new variables (PC) in the original space,i.e.,

$$min||\mathbf{X} - \hat{\mathbf{X}}||^2, \tag{2.23}$$

with the condition $\mathbf{V}\mathbf{V}^T = \mathbf{I}$, where $\hat{\mathbf{X}}$ is the coordinate matrix of the projections onto the PC subspace in the original space. Since **Z=UD=XV**, it implies that

$$\hat{\mathbf{X}} = \mathbf{Z}\mathbf{V}^\mathbf{T},$$
$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{V}\mathbf{V}^\mathbf{T}.$$

Then, the equation (2.23) can be written as

$$min||\mathbf{X} - \mathbf{X}\mathbf{V}\mathbf{V}^\mathbf{T}||^2 \quad \text{with} \quad \mathbf{V}\mathbf{V}^\mathbf{T} = \mathbf{I}. \tag{2.24}$$

The matrix **V** can be obtained from the equation 2.24 by solving a least squares problem. For this, the k two-by-two orthogonal vectors $\mathbf{v_1}, ..., \mathbf{v_k}$ that generate a subspace are calculated and with the help of the Lagrange multipliers, it is proved that these vectors

are the eigenvectors of $\mathbf{X^T X}$ associated to the eigenvalues $\lambda_1, ..., \lambda_k$ which are ordered in descending order.

The principal components obtained are linear combinations of the original variables, which guarantees that there is no correlation between them and, these have decreasing variance, with the first associated component having the highest eigenvalue that implies the highest variance, and so on.

Loadings

PC loadings are defined as the relationship that exists between a PC and a variable. It is worth noting that the sum of the squared correlation coefficients between a variable and all of its components equals 1. Because the squares of the loadings offer a proportion of the variance by the component components, we can more easily analyze them [7].

Interpretation

The interpretation of the PCs depends a lot on the magnitudes of the loadings of the components, since these show us the relationship or the contribution that the original variables have in the PC. This interpretation in some cases is not very easy to obtain, since even if we have some PCs with reduced dimensions, each one of the Principal Components is generated as a linear combination of the original variables, which would complicate their interpretation when the number of new variables is much greater. In order to facilitate the interpretation of the principal components, rotation techniques were developed [7].

Rotation

As is generally known, one of the most difficult aspects of PC analysis is interpreting the PCs. This has resulted in the use of rotation techniques on the components for interpretation in order to arrange the generated data and make it easier to interpret.

Despite the fact that the PC analysis achieves a reduced dimension in k PCs and it is expected that the interpretation of the principal components is simple when using the PC analysis, this does not always occur because the interpretation can be more complicated when the matrix data contains an extremely large number of variables [8].

Orthogonal and oblique rotations stand out among the various types of rotations. One thing to bear in mind is that even if the loading matrix changes, the variance of the model will not change when rotated.

The simplest rotations are orthogonal rotations, which directly indicate the connection between the factors and the beginning variables. The goal of this rotation is to maximize the variances of the squares of the charges in order to scatter the values as evenly as possible, raising the greatest and decreasing the smallest.

In orthogonal rotations, given a matrix of charges $\mathbf{V}$, one seeks to find an orthogonal matrix $\mathbf{Q}$ in such a way that a new matrix of charges can be created of the form

$$\hat{\mathbf{V}} = \mathbf{VQ}$$

make it easier for interpretation. The rotation matrix Q is an orthogonal matrix, which satisfies $\mathbf{QQ} = \mathbf{I}$ and associates the rows to the original axes and the columns to the new axes.

The analytical criterion of the orthogonal rotations is given by the function [22]

$$G = \sum_{m=1}^{k} \sum_{m \neq j=1}^{k} \left[\sum_{i=1}^{p} v_{ij}^2 v_{im}^2 - \frac{\gamma}{p} \sum_{i=1}^{p} v_{ij}^2 \sum_{i=1}^{p} v_{im}^2 \right] \tag{2.25}$$

where $0 \leq \gamma \leq 1$.

The orthogonal rotations that stand out the most are $VARIMAX$, $QUARTIMAX$ and $EQUIMAX$. The most commonly used rotation introduced by Kaiser(1958) [25], is a particular case of the equation 2.25 when $\gamma = 1$ known as $VARIMAX$(VARiance MAXimization)

$$G = \sum_{m=1}^{k} \sum_{m \neq j=1}^{k} \left[\sum_{i=1}^{p} v_{ij}^2 v_{im}^2 - \frac{1}{p} \sum_{i=1}^{p} v_{ij}^2 \sum_{i=1}^{p} v_{im}^2 \right] \tag{2.26}$$

This rotation seeks the least number of variables that have high loadings in each PC, i.e., that in the rotated axes the variables have the greatest number of almost null loadings and few high loadings. This approach utilizes the charge matrix's columns and maximizes the variations of each component's factor loadings.

With a limited number of variables and correlations, the goal is to look for the pres-

ence of components that have strong correlations. With the rest of them, nil. All of this results in a redistribution of the components' variance. As a result, for each element independently, the charge dispersion is maximized.

Finding a matrix **Q** orthogonal of size $k \times k$ such that said quantity is maximum is the basis of the rotational issue of $VARIMAX$ [26].

2.3 Regression Models

Regression analysis is a statistical approach that is used to investigate and model the connection between variables. Regression has many applications in practically every field, including engineering, physical and chemical sciences, economics, management, life and biological sciences, and social sciences. Regression analysis is the most extensively used statistical approach [15].

2.3.1 Multiple linear regression model

The multiple linear regression model relates p exogenous variables $\mathbf{X}$ with n endogenous variable $\mathbf{y}$ in the form:

$$\mathbf{y} = X\beta + \epsilon \tag{2.27}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

and

$$E(\mathbf{y}|\mathbf{X}) = X\beta, \tag{2.28}$$

and

$$Var(y|x) = S^2. \tag{2.29}$$

An special case of this model is the simple linear regression model with a single regressor x which is related to y through a straight line. The equation of this model is

$$y = \beta_0 + \beta_1 x + \epsilon$$

where β_0 is the intercept, β_1 is the slope and ϵ is the error of estimation, which is considered to have mean zero and variance σ^2 [15]. It should be noted that β_0 and β_1 are unknown parameters, which can be estimated using different methods. Also, we have that the mean and the variance of the model are:

$$E(y|x) = \beta_0 + \beta_1 x,$$

and

$$Var(y|x) = \sigma^2.$$

Least-squares Estimation

This method is used to estimate β using the least squares estimation. So, the least-square estimator $\hat{\beta}$ can be obtained

$$\begin{aligned} S(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \boldsymbol{\varepsilon}_i^2 \\ &= \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} \\ &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{y}^T\mathbf{y} - \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{y}^T\mathbf{y} - 2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}. \end{aligned} \tag{2.30}$$

Since $\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y}$ is an scalar and its transpose is the same scalar , we get

$$\left.\frac{\partial S}{\partial \boldsymbol{\beta}}\right|_{\hat{\beta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0}, \tag{2.31}$$

which implies that

$$2\mathbf{X}^T\mathbf{y} = 2\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}},$$
$$\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y}. \tag{2.32}$$

Then multiplying (2.32) by $(\mathbf{X}^T\mathbf{X})^{-1}$, we have that the least-squares estimator of β is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.33}$$

and the variance of the coefficients $\hat{\boldsymbol{\beta}}$ estimated by least squares is

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \mathbf{X}^T\mathbf{X}\sigma^2$$

where σ^2 corresponds to the variance of the errors of the model ε which are independent and follow a distribution $N\ (0, \sigma^2)$

Thus, the model can be estimated using

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y} \tag{2.34}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the Hat matrix $n \times n$. This matrix is particularly useful in regression analysis since it defines the degrees of freedom of the model as

$$d_f = \text{tr}(\mathbf{H}) = p \tag{2.35}$$

The equation 2.33 gives us the $\hat{\beta}$ coefficients of the regression as long as $(\mathbf{X}^T\mathbf{X})^{-1}$ exists. This does not occur when there are many more variables than observations $p >> n$, and the vector of coefficients cannot be calculated unless we apply some constraint. Shrinkage methods, which impose a penalty on the regression coefficients, play a crucial role in this scenario[24].

2.4 Shrinkage Methods

2.4.1 Ridge Regression

Let´s consider a linear regression model with n observations and p variables of the form

$$\mathbf{y} = X\beta + \epsilon \tag{2.36}$$

where X is the centered data matrix. Ridge regression is a method of estimation of the β_k coefficients through least squares, but with a penalty of the square of the l_2 norm of the β with a scalar $\lambda \geq 0$ [14]. This penalty is given by

$$\|\beta\|_2^2 = \sum_{j=1}^{p} |\beta_j|^2 \tag{2.37}$$

and the ridge coefficients $\hat{\beta}_{ridge}$ are defined as the set of all β such that they minimize the following equation

$$\sum_{i=1}^{n} \left(y_i - x_i\beta\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.38}$$

i.e.,

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}. \tag{2.39}$$

According to Hastie et al. [14], an equivalent way to write the ridge problem is

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2, \qquad \text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t. \tag{2.40}$$

The parameters λ and t shown in (2.39) and (2.40) are one-to-one correspondences. The penalty imposed on the size of the coefficients (2.37) alleviates the problem that occurs when there are numerous connected variables, since when there is a connection, the coefficients might be wrongly estimated and exhibit a high variance. The parameter λ controls the shrinkage, large values of this parameter cause more amount of shrinkage. This type of penalty (2.37) is widely used in neural networks and is known as weight decay [14].

According to Hastie et al.[14], a wildly large positive coefficient on one variable can be canceled by a similarly large negative coefficient on its correlated cousin. The remaining coefficients are computed using the centered x_{ij} via a ridge regression without intercepts. Writing the criterion in matrix form

$$\text{RSS}(\lambda) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}. \tag{2.41}$$

We have that the solution of the ridge regression is given by

$$\hat{\boldsymbol{\beta}}_{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{2.42}$$

where $\mathbf{I}$ is a identity matrix of size $p \times p$. In the equation 2.42, when $\lambda \to 0$ we get the ordinary least squares of the equation 2.33 and when $\lambda \to \infty$ we get $\hat{\boldsymbol{\beta}}_{\text{ridge}} = 0$

One of the advantages of ridge regression over ordinary regressions is that the coefficients $\hat{\boldsymbol{\beta}_{ridge}}$ exist regardless of whether $(\mathbf{X}^T\mathbf{X})^{-1}$ exists, since the penalty 2.37 adds a positive constant to the diagonal of the $(\mathbf{X}^T\mathbf{X})$ matrix so that always there exists $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$ which guarantees the existence and uniqueness of the coefficients $\hat{\boldsymbol{\beta}_{ridge}}$.

The variance of the ridge regression is

$$\text{Var}(\hat{\boldsymbol{\beta}}_{ridge}) = \sigma^2\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}$$

The Ridge regression penalty has the effect of lowering the coefficient estimates toward zero, creating **Bias** but reducing the estimate's variance.

$$\textbf{Bias}(\hat{\boldsymbol{\beta}}_{ridge}) = \lambda\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\beta}$$

The parameter λ of equation (2.39) can be chosen using the AIC (*Akaike Information Criterion*) [27], BIC (*Bayesian Information Criterion*)[28] or GCV (*General Cross Validation*).This allows us to select the model that best fits the data, resulting in a response vector $\mathbf{y}$ with the fewest variables and errors.

In ridge regression, the matrix $\mathbf{H}_{ridge}$ is given by

$$\mathbf{H}_{ridge} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T \tag{2.43}$$

and the degrees of freedom of the model are

$$df_{ridge} = \text{tr}(\mathbf{H}) = \sum_{j=1}^{p} \frac{\lambda_j^2}{\lambda_j^2 + \lambda} \tag{2.44}$$

where λ_j for $j = 1, \cdots, p$, are the eigenvalues of $\mathbf{X}$.

The appropriate value for λ can be chosen using the AIC or BIC

$$AIC = n\log(RSS) + 2df_{ridge}$$
$$BIC = n\log(RSS) + df_{ridge}\log(n).$$

However, this causes an issue since the data must be separated in order to estimate the model and calibrate its explanatory ability, and many times there is not enough data.

As a solution to this problem, the *Cross Validation* is used in practice. As a solution to this problem, the *CrossValidation* is used in practice. This consists of partitioning the data into a folds in such a way that the data is adjusted for $a - 1$ folds and testing the model on the remaining fold. In practice, the most used values for a are 5, 10 and n. The issue with this approach is its high computational cost, which is why we chose to employ *General Cross Validation*, which is defined as [29]

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{y_i} - \hat{\mathbf{y}}_\mathbf{i}}{1 - \text{tr}(\mathbf{H})/n}\right)^2 \tag{2.45}$$

2.4.2 Multicollinearity

One of the main problems that can occur on a regression model is multicollinearity, since this implies that there would be an almost linear dependence on the regressor variables. Multicollinearity is when there is an almost linear dependency between the regressors, i.e., the columns of the data matrix $\mathbf{X}$. The linear dependency between the variables would cause the matrix $\mathbf{X}^T\mathbf{X}$ to be singular, i.e., it would not have an inverse.

This problem greatly influences the regression model since the variances of the regression coefficients would be very large. The non-existence of a linear relationship between the regulating variables, it is said that these are orthogonal to each other, which facilitates an analysis and inferences about them.

Although in most applications there is no orthogonality between the regressors, in

some cases it is not so serious. However, in some cases there may be an almost perfect relationship between the regressor variables, but the inferences based on them are usually wrong.

Effects of Multicollinearity

Multicollinearity has a large number of effects on the estimators of the regression coefficients when these are calculated using least squares. For example, suppose we only have x_1 and x_2 as regressor variables. then the model can be written as

$$y = \beta_1 x_1 + \beta_2 x_2 + \epsilon \tag{2.46}$$

and using the least-squares estimation, we have

$$(\mathbf{X}^T\mathbf{X})\hat{\beta} = \mathbf{X}^T\mathbf{y} \tag{2.47}$$

i.e.,

$$\begin{bmatrix} 1 & r_{12} \\ r_{12} & 1 \end{bmatrix} \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} r_{1y} \\ r_{2y} \end{bmatrix} \tag{2.48}$$

where r_{12} is the simple correlation between x_1 and x_2, also r_{1y} and r_{2y} are the simple correlations between x_1 with y and x_2 with y respectively. The estimators of the regression coefficients are

$$\hat{\beta}_1 = \frac{r_{1y} - r_{12}r_{2y}}{\left(1 - r_{12}^2\right)} \quad \text{and} \quad \hat{\beta}_2 = \frac{r_{2y} - r_{12}r_{1y}}{\left(1 - r_{12}^2\right)}. \tag{2.49}$$

Note that when there is multicollinearity between x_1 and x_2, the coefficient r_{12} will be very large, which results in very high values of the variances and very high covariances of the least squares regression coefficient estimators. Another result of multicollinearity is the generation of estimators $\hat{\beta}_j$ with excessively high magnitudes. In order to appreciate this more clearly, consider the squared distance between $\hat{\beta}$ and β

$$L_1^2 = (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}).$$

Taking the expected value of L_1, we have that

$$\begin{aligned} E\left(L_1^2\right) &= E[(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})] \\ &= E((\hat{\beta}_1-\beta_1)+\cdots+(\hat{\beta}_p-\beta_p))\begin{pmatrix}\hat{\beta}_1-\beta_1\\ \hat{\beta}_2-\beta_2\\ \vdots\\ \hat{\beta}_p-\beta_p\end{pmatrix} \\ &= E\left(\hat{\beta}_1-\beta_1\right)^2+\cdots+\left(\hat{\beta}_p-\beta_p\right)^2 \\ &= \sum_{j=1}^{p} E\left(\hat{\beta}_j-\beta_j\right)^2. \end{aligned} \tag{2.50}$$

Note that

$$\begin{aligned} \operatorname{Var}\left(\hat{\beta}_j\right) &= E[(\hat{\beta}_j)^2]-[E(\hat{\beta}_j)]^2 \\ &= E[(\hat{\beta}_j)^2]-(\hat{\beta}_j)^2, \end{aligned} \tag{2.51}$$

and

$$\begin{aligned} E\left[\left(\hat{\beta}_j-\beta_j\right)^2\right] &= E(\hat{\beta}_j^2)-2E(\hat{\beta}_j\beta_j)+E(\beta_j^2) \\ &= E(\hat{\beta}_j^2)-2\beta_j E(\hat{\beta}_j)+E(\beta_j^2) \\ &= E(\hat{\beta}_j^2)-2\beta_j\beta_j+\beta_j^2 \\ &= E(\hat{\beta}_j^2)-2\beta_j^2+\beta_j^2 \\ &= E(\hat{\beta}_j^2)-\beta_j^2. \end{aligned} \tag{2.52}$$

From (2.51) and (2.52) we have that

$$E\left[\left(\hat{\beta}_j-\beta_j\right)^2\right] = \operatorname{Var}\left(\hat{\beta}_j\right). \tag{2.53}$$

Replacing (2.53) in (2.50), we have that

$$\begin{aligned} E\left(L_1^2\right) &= \sum_{j=1}^{k}\operatorname{Var}\left(\hat{\beta}_j\right) \\ &= \sigma^2\operatorname{tr}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} \end{aligned} \tag{2.54}$$

Some of the eigenvalues of the matrix $\mathbf{X}^T\mathbf{X}$ will be very small when there is multicollinearity and since the trace is defined as the sum of the eigenvalues of the matrix , we can rewrite (2.54) as

$$E\left(L_1^2\right) = \sigma^2 \sum_{j=1}^{k} \frac{1}{\lambda_j} \tag{2.55}$$

where λ_j are the eigenvalues of $\mathbf{X}^T\mathbf{X}$. It is clear from (2.55) that if any of the eigenvalues is very tiny due to multicollinearity, the distance estimated by the least squares defined as $E\left(L_1^2\right)$ may be large. Equivalently, we can show that

$$\begin{aligned} E\left(L_1^2\right) &= E[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] \\ &= E[(\hat{\boldsymbol{\beta}}^T - \boldsymbol{\beta}^T)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] \\ &= E[\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^T\boldsymbol{\beta} - \boldsymbol{\beta}^T\hat{\boldsymbol{\beta}} + \boldsymbol{\beta}^T\boldsymbol{\beta}] \\ &= E[\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}} - 2\hat{\boldsymbol{\beta}}^T\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\beta}] \\ &= E[\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}}] - 2E[\hat{\boldsymbol{\beta}}^T\boldsymbol{\beta}] + E[\boldsymbol{\beta}^T\boldsymbol{\beta}] \\ &= E[\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}}] - 2\boldsymbol{\beta}^T\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= E[\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta}^T\boldsymbol{\beta}. \end{aligned} \tag{2.56}$$

From (2.56) and (2.54), it follows that

$$\begin{aligned} E\left(\hat{\boldsymbol{\beta}}^T\hat{\boldsymbol{\beta}}\right) &= E\left(L_1^2\right) + \boldsymbol{\beta}^T\boldsymbol{\beta} \\ &= \boldsymbol{\beta}^T\boldsymbol{\beta} + \sigma^2 \operatorname{Tr}\left(\mathbf{X}'\mathbf{X}\right)^{-1}. \end{aligned} \tag{2.57}$$

Since vector $\hat{\boldsymbol{\beta}}$ is longer than vector $\boldsymbol{\beta}$, the least squares approach gives predicted regression coefficients that are excessively big in absolute value.

2.4.3 LASSO Regression

Let´s consider a linear regression model with n observations and p variables of the form

$$\mathbf{y} = X\beta + \epsilon \tag{2.58}$$

where X is the centered data matrix. LASSO (LeastAbsolute Shrinkage and Selection Operator) regression is a method of estimation of the β_k coefficients through least squares,

but with a penalty of the square of the l_1 norm of the β with a scalar $\lambda \geq 0$ [14].

This penalty is given by

$$\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \tag{2.59}$$

and the LASSO coefficients $\hat{\beta}_{LASSO}$ are defined as the set of all β such that they minimize the following equation

$$\sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j. \tag{2.60}$$

The solution of the problem (2.60) is defined by

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \tag{2.61}$$

subject to

$$\sum_{j=1}^{p} \left|\beta_j\right| \leq t. \tag{2.62}$$

According to Hastie et al.[14], we can also write the lasso problem in the equivalent Lagrangian form

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \right\}. \tag{2.63}$$

The coefficients predicted by LASSO are constantly reduced to zero in order to enhance model prediction via variance and bias. Since of their penalty (2.59), certain coefficients become zero for high values of λ, and this approach is beneficial for variable selection since LASSO creates both a sparse and precise model.

Since the LASSO estimator is not a linear estimator, the hat matrix $\mathbf{H}_{LASSO}$ cannot be defined such that

$$\hat{\boldsymbol{y}} = \mathbf{H}_{LASSO}\, \boldsymbol{y} \tag{2.64}$$

this makes it difficult for us to calculate the degrees of freedom. For this, the non-null coefficients in the regression model can be used to quantify the degrees of freedom and to be able to implement the *AIC*, the *BIC* or the *General Cross Validation* as selection criteria of λ.

As in ridge regression, the most reliable method of estimating λ is through general cross validation.

2.4.4 Elastic-Net Regression

This regression model combines the LASSO and ridge penalties in such a way that it preserves the individual advantages of each method and in turn overcomes the problems of each [12].

For non-negative λ_1 and λ_2, the model coefficients $\hat{\boldsymbol{\beta}}_{EN}$ are defined as

$$\hat{\boldsymbol{\beta}}_{EN} = (1+\lambda_2)\left\{\arg\min_{\beta}\left\|Y - \sum_{j=1}^{p} X_j \dot{\beta} j\right\|^2 + \lambda_2 \sum_{j=1}^{p}\left|\beta_j\right|^2 + \lambda_1 \sum_{j=1}^{p}\left|\beta_j\right|\right\}. \tag{2.65}$$

In the equation (2.65), the term $\lambda_2 \sum_{j=1}^{p}\left|\beta_j\right|^2$ allows variables with a high correlation coefficient to have similar coefficients, while the term $\lambda_1 \sum_{j=1}^{p}\left|\beta_j\right|$ allows Sparse solutions. Note that for $\lambda_2 = 0$, the equation (2.65) becomes in the LASSO equation.

For the case $p >> n$, the solution is given for $\lambda_2 > 0$ which eliminates the main disadvantage of LASSO including all the variables in the model [2].

Chapter 3

Sparse Principal Component Analysis

3.1 Sparse Principal component analysis (SPCA)

Principal Component Analysis (PCA) was developed to improve interpretation of Principal Components, help with non-uniqueness and some inconsistencies that arise from loading. For this reason, the sparse PCA was originated, which seeks to achieve that a large part of the coefficients of the charge matrix are zero, thus facilitating the understanding of the principal components[30].

It must be taken into account that this technique has applications in practically all areas of science: machine learning, image processing, engineering, genetics, neurocomputing, chemistry, meteorology, control theory, computer networks, etc[31]. The SPCA solution criterion is defined from the row vectors $\mathbf{x_i}$ of the matrix $\mathbf{X}$.

Hastie, Tibshirani and Friedman (2009)[14] transform the PCA into a regression problem imposing the ridge penalty by means of the following theorem

Theorem 1 *Let* $\mathbf{A_{p\times k}}$ *and* $\mathbf{B_{n\times k}}$ *be matrices. If* $\lambda_2 > 0$ *and*

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \underset{\mathbf{A},\mathbf{B}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| \mathbf{x_i} - \mathbf{A}\mathbf{B}^T\mathbf{x_i} \right\|^2 + \lambda_2 \sum_{j=1}^{k} \left\| \beta_j \right\|^2 \quad \textit{subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k} \tag{3.1}$$

then, $\hat{\beta}_j \propto V_j$ *for* $j = 1, 2, \ldots, k$.

They also propose obtaining sparse loadings through the LASSO penalty, which gives

rise to the equation 3.2

$$(\hat{A}, \hat{B}) = \underset{A,B}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| x_i - AB^T x_i \right\|^2 + \lambda_2 \sum_{j=1}^{r} \left\| \beta_j \right\|^2 + \sum_{j=1}^{r} \lambda_{1,j} \left\| \beta_j \right\|_1 \quad \text{subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}_{kxk} \tag{3.2}$$

The steps to solve the problem 3.2 can be described as follows

Algorithm 1: General Sparse PCA Algorithm

1 Initialize the loadings of the k principal components $\mathbf{A}$ at $\mathbf{V}[, 1:k]$.
2 Solve the problem

$$\beta_j = \arg\min_{\beta} \left(\alpha_j - \beta\right)^T \mathbf{X}^T\mathbf{X} \left(\alpha_j - \beta\right) + \lambda\|\beta\|^2 + \lambda_{1,j}\|\beta\|_1$$

with a fixed $\mathbf{A} = [\alpha_1, \ldots, \alpha_k]$.
3 Calculate the singular value decomposition of $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ with $\mathbf{B} = [\beta_1, \cdots, \beta_k]$ and let´s pick $\mathbf{A} = \mathbf{U}\mathbf{V}^T$.
4 Steps 2-3 must be repeated until convergence is achieved.
5 Normalize $\widehat{V}_j = \frac{\beta_j}{\|\beta_j\|}, j = 1, \ldots, k$.

3.1.1 Sparse PCA via Lasso

The SCoTLASS algorithm (Simplified Component Technique subject to LASSO) is one of the most important when talking about Sparse PCA. This algorithm proposed by Jolliffe, Trendafilov, and Uddin (2003) modifies the principal components by imposing the LASSO penalty in such a way that it allows obtaining many null charges. Let's consider a data matrix $\mathbf{X}_{n\times p}$. This method solves the problem 3.3

$$\mathbf{a_k}^T \left(\mathbf{X}^T\mathbf{X}\right) \mathbf{a_k}, \tag{3.3}$$

subject to

$$\mathbf{a_k}^T\mathbf{a_k} = 1 \quad \text{and} \quad (\text{for} \quad k \geq 2\,) \quad \mathbf{a_h}^T\mathbf{a_k} = 0, \quad h < k. \tag{3.4}$$

It maximizes the equation 3.3 using the LASSO penalty

$$\|\mathbf{a_k}\|_1 = \sum_{j=1}^{p} \left|a_{kj}\right| \leq t.$$

This problem can also be rewritten as

$$\max_{\|\boldsymbol{a}\|=1, \boldsymbol{a} \perp \boldsymbol{a}_{1,\ldots,}, \boldsymbol{a} \perp \boldsymbol{a}_{j-1}} \boldsymbol{a}^t \left(\mathbf{X}^T \mathbf{X}\right) \boldsymbol{a} - \lambda_1 \|\boldsymbol{a}\|_1. \tag{3.5}$$

According to Zou, Hastie and Tibshirani (2006) "The high computational cost of SCoTLASS makes this an impractical solution. This high computational cost is probably due to the fact that SCoTLASS is not a convex optimization problem"[2]. The parameter t has a great influence on the loadings, since sufficiently small values of t produce exactly zero loadings. In SCoTLASS, the value of the parameter t is very important, which is why there are certain values that give us information (see figure 3.1).

As t is reduced from $\sqrt{p}$, we will not get the PCA directly and get a solution that has only one non-zero charge on each component for the variable, while the other variables will be reduced along with t to zero.

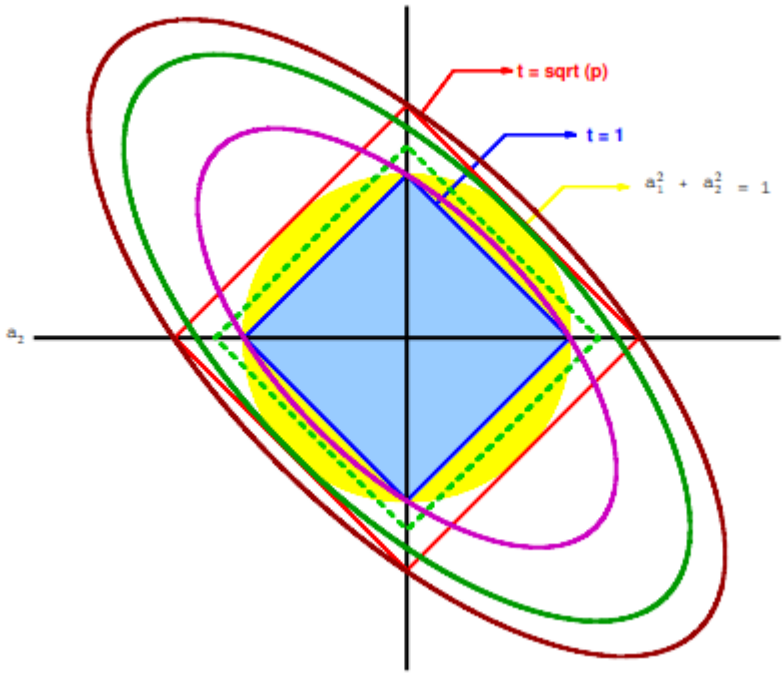

Figure 3.1: The two-Dimensional SCoTLASS [1].

(a) For values of $t \geq \sqrt{p}$, we can carry out the Principal Component Analysis.

(b) For values of $1 < t < \sqrt{p}$, the SCoTLASS is limited by the part of unit circle $a_1^T a_1 = 1$ inside the green dotted square $\sum_{j=1}^{2} \left|a_{1j}\right| \leq t$ (see Figure 3.1).

(c) For values of $t < 1$, the problem 3.5 does not have solution.

(d) For $t = 1$, we have one nonzero a_{kj} for each k. These correspond to the shaded square (see Figure 3.1) and the solutions are only on the axes

3.1.2 Robust Sparse Principal Component Analysis Via Variable Projection (ROBSPCA)

It is a technique used for matrix decomposition and dimensionality reduction, where the goal is to separate a matrix into two components: a low-rank component and a sparse component. The low-rank component represents the underlying structure or signal of the data, while the sparse component represents the noise or outliers. Robust sparse PCA is one of the few methods that combines robustness and sparseness. It focuses on PCs projection and search, where PCs are extracted from the data by searching for directions that maximize a robust measure of variance of the projected data. Using this variance-robust method prevents PCs from being attracted to outliers that inflate the standard variance.

On the one hand, sparsity can be imposed on PCA directions by adding an $l1$ penalty in the objective function as we can see in the equation 3.3. So, one way of robust sparse PCA is to replace the empirical covariance matrix with a robust covariance estimator, as is often done in robust multivariate data analyzes [32]. On the other hand, to avoid errors and problems in the estimator, this method proposes a projection-search approach, where the PCs are obtained directly using a previous estimation of covariance.

$$\max_{\|\boldsymbol{a}\|=1, \boldsymbol{a} \perp \boldsymbol{a}_1, \ldots, \boldsymbol{a} \perp \boldsymbol{a}_{j-1}} \boldsymbol{a}^t \hat{\boldsymbol{\Sigma}} \boldsymbol{a}, \quad \text{subject to } \|\boldsymbol{a}\|_1 \leq t \tag{3.6}$$

where $\hat{\mathbf{a}}_\mathbf{j}$ is the sparse PCA direction and λ_1 controls sparcity. Note that

- $\lambda_1 = 0$ results in the first unrestricted PCA direction a_1.
- $\lambda_1 > 0$ sparsity gains importance.

The j^{th} sparse PCA direction is defined by $(1 < j \leq p)$. The projection-pursuit approach reduces the computation time of this algorithm since the estimators are computed se-

quentially and it would not make sense to calculate all the principal components but just a few [33]. Finally, the advantage of the robust sparse PCA over the SCoTLASS, PCA with rotation and Sparse PCA via elasticnet methods is that this method is efficient with respect to outliers due to its robustness[34].

3.1.3 Randomized sparse principal component analysis Via Variable Projection (RSPCA)

This method combines the concept of low-rank matrix and random methods, which allows to build a low-dimensional scheme that captures essential information from the initial data. Then we consider

(a) This equation denotes the randomized value function

$$v(\mathbf{B}) := \min_{\mathbf{A}} \frac{1}{2} \left\| \widetilde{\mathbf{X}} - \widetilde{\mathbf{X}}\mathbf{B}\mathbf{A}^\top \right\|_F^2 \quad \text{subject to } \mathbf{A}^\top\mathbf{A} = \mathbf{I},$$

where the low-dimensional scheme of $\mathbf{X} \in \mathbb{R}^{n\times p}$ is given by $\widetilde{\mathbf{X}} \in \mathbb{R}^{l\times p}$. In this case l is bigger than k.

(b) Now, a new sample matrix $\mathbf{Y} \in \mathbb{R}^{n\times p}$ is created such that

$$\mathbf{Y} = \mathbf{X}\mathbf{\Omega}$$

where $\mathbf{\Omega} \in \mathbb{R}^{p\times l}$ is a randomly generated test matrix.

(c) We compute the $\mathbf{QR}$ factorization of the sample matrix $\mathbf{Y}$, to obtain an orthonormal basis matrix

$$\mathbf{Y} = \mathbf{Q}\mathbf{R}. \tag{3.7}$$

(d) Finally we create the low-dimensional schema by projecting the data matrix onto the range of $\mathbf{Y}$,i.e.

$$\tilde{\mathbf{X}} = \mathbf{Q}^\top\mathbf{X}. \tag{3.8}$$

This algorithm has computational advantage and it becomes significant when the range of the data is small compared to the dimension of the measurement space[34].

3.1.4 Sparse PCA for $p >> n$ (Gene Array)

This is a special Sparse PCA model that is applied when we have many more variables than observations ($p >> n$). For example, in the case of gene expression array the number of genes (variables) we have is much larger than the number of samples (observations).

The SPCA general algorithm can be adapted to this case by using $\lambda_1 > 0$. A major disadvantage found in this algorithm is that by having many variables, it will have to search for a large number of non-zero loads, which considerably increases its computational cost [2]. Since a positive $\lambda_1 > 0$ is required, a saving solution is proposed that is $\lambda_1 \to \infty$ which generates the following theorem

Theorem 2 *Let* $\widehat{V}_j(\lambda) = \frac{\hat{\beta}_j}{\|\hat{\beta}_j\|}(j = 1, \ldots, k)$ *be the loadings obtained from the equation 3.2. Then* $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ *is the solution of the problem:*

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \underset{\mathbf{A},\mathbf{B}}{\arg\min} - 2\,\mathrm{tr}\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right) + \sum_{j=1}^{k}\left\|\beta_j\right\|^2 + \sum_{j=1}^{k}\lambda_{1,j}\left\|\beta_j\right\|_1 \qquad (3.9)$$
$$\textit{subject to } \mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}.$$

when

$$\lambda \to \infty \quad \textit{and} \quad \widehat{V}_j(\lambda) \to \frac{\hat{\beta}_j}{\left\|\hat{\beta}_j\right\|}.$$

The algorithm to carry out this type of Sparse PCA is the same as the one mentioned above 1, only with a variation in step 2 that would now be as follows:

For $j = 1, \ldots, k$

$$\beta_j = \left(\left|\alpha_j^T\mathbf{X}^T\mathbf{X}\right| - \frac{\lambda_{1,j}}{2}\right)_+ \mathrm{Sign}\left(\alpha_j^T\mathbf{X}^T\mathbf{X}\right). \qquad (3.10)$$

This operation is known as Soft-Thresholding.

Chapter 4

Methodology

There are several algorithms to carry out the Sparse PCA, however in this thesis we take the Sparse PCA via variable projection as a reference and we apply the SPCA, RSPCA and ROBSPCA algorithms, since these present significant advantages and are efficient when working with outliers and the computational cost of this is relatively low when we have data of low dimension

4.1 Sparse PCA Steps

In a simplified way we can explain that the Sparse PCA procedure can be carried out following the steps described below:

(a) **Pre-Processing:** Before the analysis can begin, the input data is pre-processed to eliminate any noise or outliers that can skew the results.

(b) **Covariance Matrix:** The pre-processed data's covariance matrix is computed.

(c) **Eigenvalue Decomposition:** To determine the eigenvectors and eigenvalues of the covariance matrix, the eigenvalue decomposition is carried out.

(d) **Sparsity Constraint:** A sparsity constraint is imposed on the eigenvectors to make sure that only a small number of coefficients are non-zero.

(e) **Modified Power Iteration:** A modified power iteration algorithm is used to find the eigenvectors that maximize variance subject to the sparsity constraint.

(f) **Iteration:** The algorithm continues iterating until convergence, i.e., when the covariance matrix of the residual errors is below a certain threshold.

(g) **Selection of Sparse Components:** The sparse components with the highest variance are selected as the principal components.

The following stages make up the sparse PCA algorithm:

4.2 Algorithm Design

For the `R` implementation of the Sparse Principal Component Analysis (SPCA) algorithm it was necessary to install Package `sparsepca` focuses on finding sparse weight vectors (loadings), with only a few non-zero values. This approach provides better interpretability of principal components in high-dimensional data sets. This is because the principal components are formed as a linear combination of only a few of the original variables. This package provides efficient routines for SPCA. Specifically, a variable projection solver is used to compute the sparse solver. In addition, a fast random accelerated SPCA routine and a robust SPCA routine are provided. Robust SPCA allows you to capture severely corrupted entries in the data. The methods are discussed in [34].

4.3 Packages

There are a few different packages available in RStudio for performing sparse PCA (principal component analysis). Here are a few options:

4.3.1 Elasticnet package

The `elasticnet` package offers functions for performing sparse PCA using an elastic net penalty. The `epca` function can be used to run the PCA, and the `print.epca` function can be used to view the results.

4.3.2 PcaPP package

The `pcaPP` package includes a `spca` function for sparse PCA using a penalized covariance matrix.

4.3.3 PcaMethods package

The `pcaMethods` package includes a `pca` function that can be used with the `sparsePCA` method. The `plotLoadings` function can be used to view the resulting loadings.

4.4 Implementation

Robust Sparse Principal Component Analysis (ROBSPCA)

The principal components are produced as a linear combination of only a few of the original variables, this technique improves model interpretability. Furthermore, SPCA avoids overfitting in a high-dimensional data configuration with p variables higher than n observations. This parsimonious model is obtained by introducing prior information as regularizers that favor scarcity. More specifically, given a data matrix $\mathbf{X}_{n\times p}$, the robust SPCA tries to minimize the following objective function:

$$f(\mathbf{A},\mathbf{B}) = \frac{1}{2}\left\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top - \mathbf{S}\right\|_F^2 + \psi(\mathbf{B}) + \gamma\|\mathbf{S}\|_1 \tag{4.1}$$

where:

- $\mathbf{B}$ is the sparse weight matrix (loadings)
- $\mathbf{A}$ is an orthonormal matrix.
- ψ denotes a sparsity inducing regularizer such as the LASSO or the elastic net.
- The matrix $\mathbf{S}$ captures grossly corrupted outliers in the data.
- $\mathbf{Z} = \mathbf{X}\mathbf{B}$
- And the data can be approximately rotated back as

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{A}^\top \tag{4.2}$$

Randomized sparse principal component analysis (RSPCA)

This parsimonious model is obtained by introducing prior information as regularizers that promote scarcity. More specifically, given a data matrix $\mathbf{X}_{n\times p}$, SPCA tries to mini-

mize the following objective function

$$f(\mathbf{A}, \mathbf{B}) = \frac{1}{2} \left\| \mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top \right\|_F^2 + \psi(\mathbf{B}) \tag{4.3}$$

where

- $\mathbf{B}$ is the sparse weight matrix (loadings)
- $\mathbf{A}$ is an orthonormal matrix.
- ψ denotes a sparsity inducing regularizer such as the LASSO or the elastic net.
- $\mathbf{Z} = \mathbf{X}\mathbf{B}$
- And the data can be approximately rotated back as

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{A}^\top \tag{4.4}$$

Sparse Principal Component Analysis (SPCA)

SPCA avoids overfitting in a high-dimensional data setup where the number of variables p is greater than the number of observations n. This parsimonious model is obtained by introducing prior information as regularizers that promote sparsity. More specifically, given a data matrix $\mathbf{X}_{n \times p}$, SPCA tries to minimize the following objective function

$$f(\mathbf{A}, \mathbf{B}) = \frac{1}{2} \left\| \mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top \right\|_F^2 + \psi(\mathbf{B}) \tag{4.5}$$

where

- $\mathbf{B}$ is the sparse weight matrix (loadings)
- $\mathbf{A}$ is an orthonormal matrix.
- ψ denotes a sparsity inducing regularizer such as the LASSO (l_1 norm) or the elastic net (a combination of the l_1 norm and l_2 norm).
- $\mathbf{Z} = \mathbf{X}\mathbf{B}$

- And the data can be approximately rotated back as

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{A}^{\top} \tag{4.6}$$

We now apply our SPCA framework to a Clinical Tests (COVID-19) data Set from various clients of the Ibarra clinical laboratory. These real data examples capture many challenges driving new algorithms, with high-dimensional measures and low-dimensional structures across multiple scales. In this case we will apply the three algorithms: SPCA, ROBSPCA, RSPCA. These will reduce the dimension of the data to facilitate the understanding of the principal components obtained. In this case, the variables to consider in the data set are IGM, IGG, PCT, Dimer, Ferritin, Age and CRP. Finally, we project the input data in new directions, known as principal components (PCs), which absorb as much information as possible and thus be able to eliminate those variables that contribute less variability.

Chapter 5

Data Description

5.1 Data Description

This section provides a description of the data sets that were used. The dimension of the data is not very large since they have 7 variables, but the most interesting thing about this data set is that the observations of each variable are not on the same scale, i.e., there are very large values and others that are too small.

5.1.1 Clinical Tests (COVID-19) Data Set

The Clinical Laboratory of Clínica Ibarra, located in Ibarra, Ecuador, provided the first data set for this thesis. This data set was used for the first time by Enríquez et al [35] who applied the "random" CUR algorithm technique. Permission to use this data was obtained from the laboratory owner because this clinical facility is private. For privacy reasons, we choose not to include people's IDs. A total of 255 people were evaluated between May 17 and June 26, 2020. The variables used were: Age, IGG, IGM, D-dimer, Ferritin, PCT and PCR, however only two of them are focused on the COVID-19.[35]

First, we will explain the IGG and IGM variables. These are responsible for detecting an infection in our respiratory system. In our data set they are used to confirm or rule out respiratory conditions caused by COVID-19. In this case, the COVID-19 IgG antibody was in charge of detecting the presence of old infections in the individual. While the COVID-19 IgM antibody provides us with information about the presence of a current respiratory infection, since these show the individual's immune response. The Clinical

Laboratory - Clínica Ibarra performs these tests using a method called immunofluorescence. Obtaining the results of both the IGG and the IGM analyzes takes approximately 1 hour from the patient's blood sample. The results of this type of examination are qualitative, can be quantitative and expressed numerically. That is, the IGG and IGM variables have a range between 0-10. Negative results are shown between 0-0.9, indeterminate or without much information are shown between 0.9-1.1 and IGG and IGM levels must be greater than or equal to 1.1 for the test to be considered positive. Finally, we are going to explain the other five variables. Then, we get

(1) Variable age: it is the one that shows the age of the individual.

(2) Variable D-dimer: looks for D-dimer in the blood, a piece of protein that is produced when a blood clot dissolves in the body [value range is 0, 500 ng/mL].

(3) Variable Ferritin: is a test that measures the level of ferritin in the blood [value range is 30, 350 ng/mL].

(4) Variable PCT: is the procalcitonin test that measures the level of procalcitonin in the blood (High levels indicate serious infections)

(5) Ultrasensitive CRP variable: measured in mg/l, indicates C-reactive protein values [reference values for this test are 0.5 mg/L]

Chapter 6

Results and Discussion

In this section we will apply 3 sparse principal component analysis (SPCA) algorithms to the "Clinical Tests (COVID-19)" dataset. First, an exploratory analysis of the data was carried out, which allows us to have a prior idea of how the data behaves before applying the SPCA to them. Then we will apply the SPCA , ROBSPCA , and RSPCA algorithms described in the section 4.4 to obtain the sparse principal components of the data set. Finally, we will compare which of the applied sparse PCA algorithms manages to capture the greatest amount of information with fewer principal components.

6.1 Exploratory Analysis of Data

First, summary measures for each variable were computed, yielding the following values for each variable's means and variances (see table 6.1)

	Age	**IGG**	**IGM**	**D-Dimer**	**Ferritin**	**PCT**	**CRP**
min	1	0	0	20.73	20.5	0	0
mean	42.04	0.1914	1.684	674.70	188.4	0.2578	10.68
variance	374.0772	0.2160	53.5764	2130105.6895	11497.8003	0.1714	1046.3900
max	94	3.7	47.10	9562.45	578.4	3.4	199.60

Table 6.1: Summary of variables.

Also, for this data set we have its correlation matrix 6.2 which is given by

$$\begin{bmatrix} 1.0000 & -0.0052 & 0.0586 & 0.1054 & 0.0581 & 0.0157 & 0.0552 \\ -0.0052 & 1.0000 & 0.1717 & 0.2222 & 0.1154 & 0.5008 & 0.7184 \\ 0.0586 & 0.1717 & 1.0000 & 0.9160 & 0.4014 & 0.6232 & 0.5385 \\ 0.1054 & 0.2222 & 0.9160 & 1.0000 & 0.4334 & 0.5867 & 0.5608 \\ 0.0581 & 0.1154 & 0.4014 & 0.4334 & 1.0000 & 0.2736 & 0.2593 \\ 0.0157 & 0.5008 & 0.6232 & 0.5867 & 0.2736 & 1.0000 & 0.6504 \\ 0.0552 & 0.7184 & 0.5385 & 0.5608 & 0.2593 & 0.6504 & 1.0000 \end{bmatrix}$$

Table 6.2: Correlation Matrix of Data set

and its determinant is equal to 0.01592. Note that this value is close to zero which implies that the variables (simultaneously) of the data set are correlated and we can apply this multivariate technique (SPCA).

In the same way we can see easier the variables that are correlated in the figure 6.1. Note that according to figure 6.1, the variables D_Dimer and IGG have a high positive correlation due to the correlation coefficient is 0.92. Another variables that has a significant positive correlation are CRP and IGM, PCT and CRP, D_Dimer and PCT with a correlation coefficient of 0.72, 0.65 and 0.59, respectively.

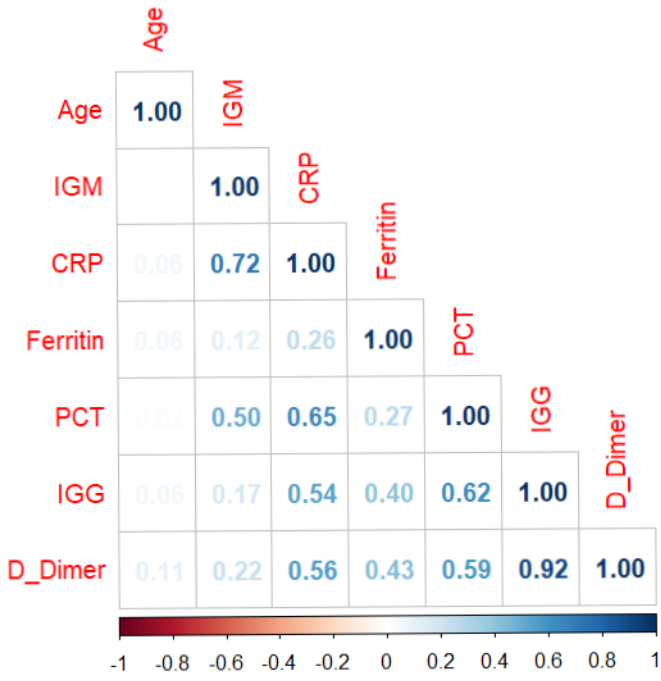

Figure 6.1: Graphical representation of the Correlation matrix

Furthermore we have another graph obtained from the data set called *Violin plots* (figure 6.2), which helps us to visualize the distribution of numerical data of different variables. As can be seen, the regions with the widest distribution curve correspond to the greatest presence of data in that region. Also due to the scale variety of the data, the curve generated for the D_Dimer variable is much larger than the others. Additionally, in each density curve of the variables, there is a cloud of points of the values of each variable and it is observed that most of the points are inside the curves.

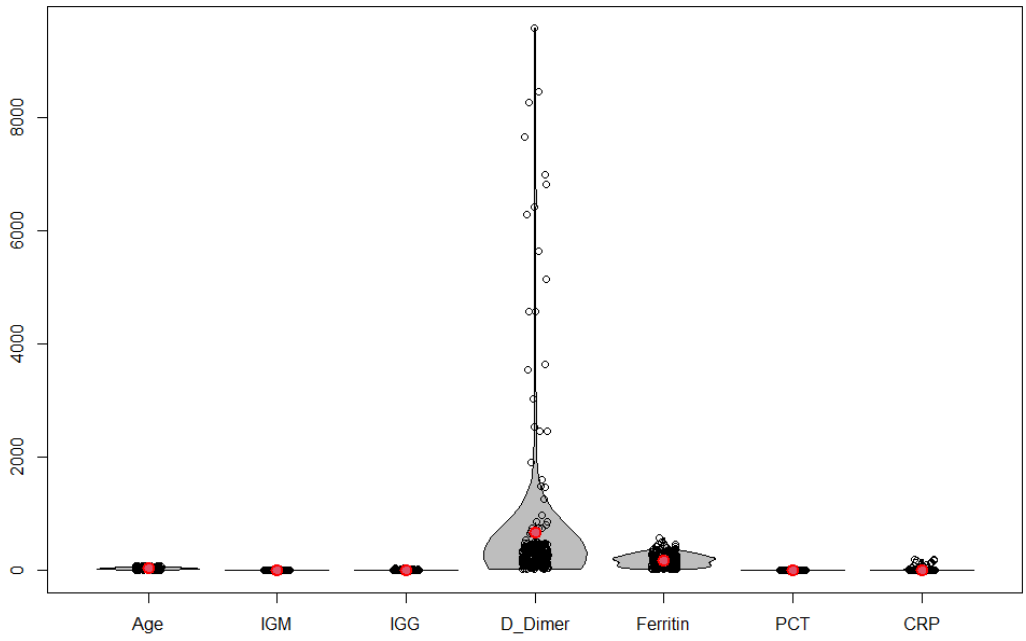

Figure 6.2: Violin plot of Real Data

In order to get a better observation of the violin plot, we plotted on with the standardized data (figure 6.3). In this case, given that the variables are in similar dimensions and that they have a mean of 0 and a variance of 1, it is possible to better appreciate the distributions together with the observations of each variable.

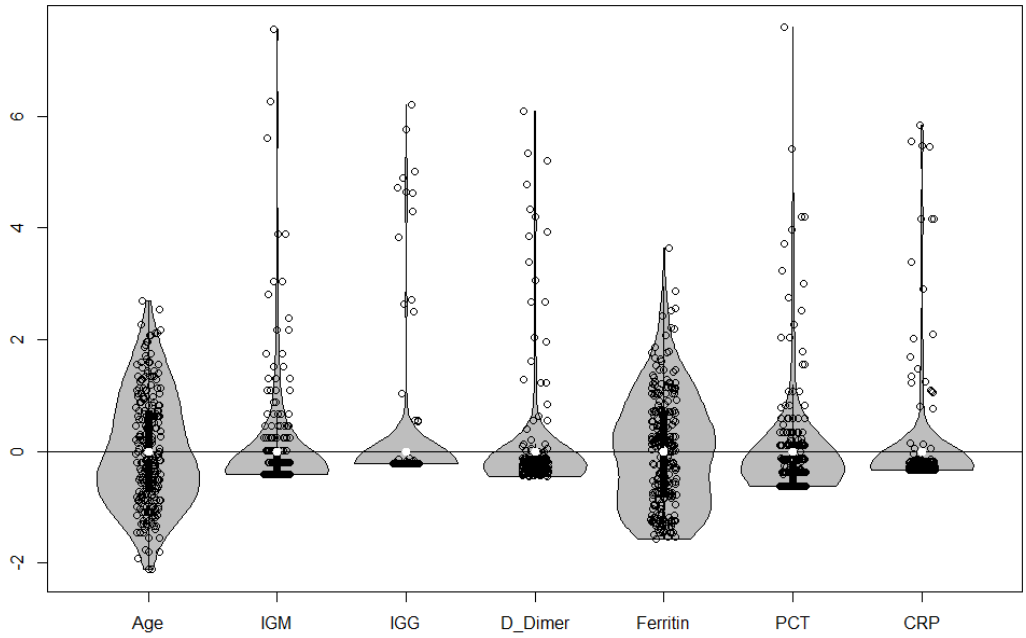

Figure 6.3: Violin plot of Standardized Data

It is important to take into account that based on the data from the IGG and IGM variables, the possible positive and negative results of the covid tests can be determined. Following the ranges of values to determine the absence or presence of the disease provided by the laboratory, we have the following results:

IGG	IGM	Percentage of Individuals	Clinical interpretation
≤ 0.9	≤ 0.9	89.80%	Negative
≥ 1.1	≤ 0.9	3.92%	Positive
≤ 0.9	≥ 1.1	1.18%	Positive
≥ 1.1	≥ 1.1	5.10%	Positive

Table 6.3: Data Interpretation.

From the Table 6.3, we have that 89.80% of the tests carried out were negative, i.e., these patients have never had COVID-19. while 10.20% were positive either because the patient was in the early phase of the COVID-19 infection, in the active phase or in its final phase.This may even be because this is a recurring infection or an old infection.

6.1.1 Sparse Principal Component Analysis (SPCA)

We used R-studio to apply the SPCA algorithm on Clinical Tests (COVID-19) Data Set through the following function

```
SPCA <- spca(X, k=4, alpha=1e-3, beta=1e-6,center = TRUE, scale
    = TRUE, verbose=0).
```

For this function, several values for alpha and beta were tested, which are the Sparsity controlling parameter and the Amount of ridge shrinkage to apply in order to improve conditioning and it was found that with $alpha = 1 \times 10^{-}3$ and $beta = 1 \times 10^{-}6$ the greatest amount of variance is reached, for which we worked with those values.

In this case we chose 4 principal components since with it a cumulative proportion of the variance of 0.907 is reached. As you can see in the table 6.4, the first principal component has an explained variance of 3.42, which is equivalent to a variance proportion of 0.482, while the second, third, and fourth principal components have an explained variance of 1.226, 0.977 and 0.724, which represents 0.175 , 0.140, and 0.103 of the proportion of variance respectively.

	PC1	PC2	PC3	PC4
Explained variance	3.420	1.226	0.977	0.724
Standard deviations	1.849	1.107	0.988	0.851
Proportion of variance	0.489	0.175	0.140	0.103
Cumulative proportion	0.489	0.664	0.803	0.907

Table 6.4: SPCA Summary

In the figure 6.4 we have on the left side the graph of the variances for each principal component. It is easy to see that as the number of principal components increases, the variance associated with each one decreases, with the first component having the maximum variance and the last component the minimum variance. While on the right side we have the graph of the proportion of variances by principal component.

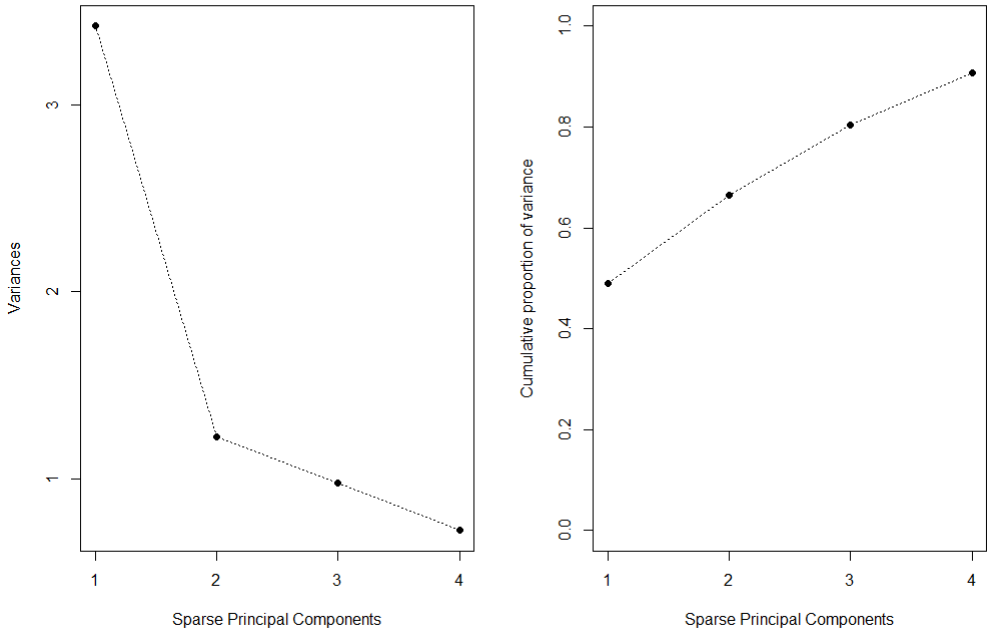

Figure 6.4: Variances of the Principal components

As with the variances, each principal component has an associated eigenvalue (see table 6.5) where the first component has the largest eigenvalue and the fourth component has the smallest eigenvalue.

	PC1	PC2	PC3	PC4
Eigenvalues	3.420	1.226	0.977	0.724

Table 6.5: Eigenvalues of Principal Components

The loadings help determine which variables are mostly contained in each of the principal components. The advantage of these sparse loadings (see table 6.6) obtained from the sparse pca is that most of them are very close to 0, which facilitates the interpretation of the principal components.

From the table 6.6 we can interpret that the age variable is contained in the third principal component since it has a loading value of 0.996 in this component. The IGG, D_Dimer, PCT, CRP variables are mostly contained in the first principal component since they have a higher loading value in that component than in the others. The IGM variable is mostly contained in the second principal component with a loading of -0.744 which is by magnitude the largest of all loadings. Finally, the Ferritin variable is in the fourth principal component since it has a loading of 0.994.

	PC1	PC2	PC3	PC4
Age	0.000	0.000	0.996	0.000
IGM	0.230	-0.744	0.000	0.017
IGG	0.524	0.394	0.000	0.000
D_Dimer	0.512	0.348	0.013	0.012
Ferritin	0.023	0.000	0.000	0.994
PCT	0.474	-0.104	-0.034	-0.071
CRP	0.425	-0.385	0.018	0.000

Table 6.6: Sparse Loadings

The table 6.7 contains the value of the 4 principal components for each of the 255 observations. For ease, only the first 3 rows and the last 4 rows are shown.

	PC1	PC2	PC3	PC4
[1]	-0.493178803	0.205505522	-0.519122479	0.534950777
[2]	-0.564959676	0.176131843	0.149302952	-0.800133607
[3]	-0.903131364	0.256824953	-0.610051226	-1.297055002
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
[252]	-0.813972443	0.099986926	1.038883298	-1.105554036
[253]	-0.748976974	-0.044547725	-0.556990363	-1.288587220
[254]	-0.589171728	-0.121423416	-0.049442101	0.286173100
[255]	6.949755323	1.972385485	0.327976351	1.533082847

Table 6.7: SPCA values per observation

6.1.2 Robust Sparse Principal Component Analysis (ROBSPCA) on Clinical Tests (COVID-19) Data Set

We used R-studio to apply the ROBSPCA algorithm on Clinical Tests (COVID-19) Data Set through the following function

```
ROBSPCA <- robspca(X, k=4, alpha=1e-4, beta=1e-6, gamma=1,
    center = TRUE, scale = TRUE, verbose=0)
```

For this function, several values for alpha, beta and gamma were tested, which are the Sparsity controlling parameter, the Amount of ridge shrinkage to apply in order to improve conditioning and Sparsity controlling parameter for the error matrix **S** respec-

tively. It was found that with `alpha` $= 1 \times 10^{-3}$, `beta` $= 1 \times 10^{-6}$ and `gamma` $= 1$. The greatest amount of variance is reached, for which we worked with those values.

In this case we chose 4 principal components since with it a cumulative proportion of the variance of 0.906 is reached. As you can see in the table 6.8, the first principal component has an explained variance of 3.423, which is equivalent to a variance proportion of 0.489, while the second, third, and fourth principal components have an explained variance of 1.218, 0.979 and 0.723, which represents 0.174 , 0.140, and 0.103 of the proportion of variance respectively.

	PC1	PC2	PC3	PC4
Explained variance	3.423	1.218	0.979	0.723
Standard deviations	1.850	1.104	0.990	0.850
Proportion of variance	0.489	0.174	0.140	0.103
Cumulative proportion	0.489	0.663	0.803	0.906

Table 6.8: ROBSPCA Summary

In the figure 6.5 we have on the left side the graph of the variances for each principal component. It is easy to see that as the number of principal components increases, the variance associated with each one decreases, with the first component having the maximum variance and the last component the minimum variance. While on the right side we have the graph of the proportion of variances by principal component.

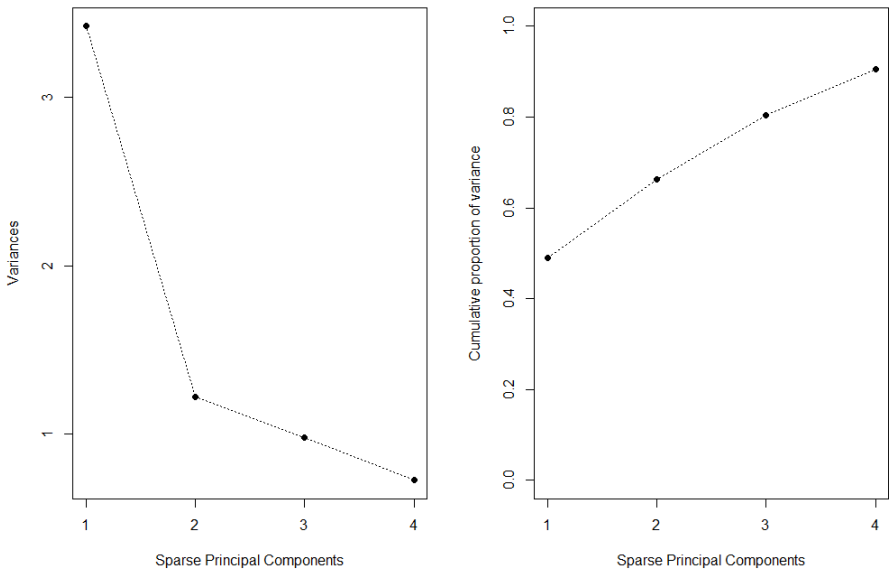

Figure 6.5: Variances of the Principal components

As with the variances, each principal component has an associated eigenvalue (see table 6.9) where the first component has the largest eigenvalue and the fourth component has the smallest eigenvalue.

	PC1	PC2	PC3	PC4
Eigenvalues	3.423	1.218	0.979	0.723

Table 6.9: Eigenvalues of Principal Components

From the table 6.10 we can interpret that the age variable is contained in the third principal component since it has a loading value of 0.963 in this component. The IGG, D_Dimer, PCT, CRP variables are mostly contained in the first principal component since they have a higher loading value in that component than in the others. The IGM variable is mostly contained in the second principal component with a loading of -0.644 which is by magnitude the largest of all loadings. Finally, the Ferritin variable is in the fourth principal component since it has a loading of 0.871.

	PC1	PC2	PC3	PC4
Age	0.049	0.253	0.963	-0.018
IGM	0.315	-0.644	0.156	0.241
IGG	0.458	0.339	-0.136	-0.311
D_Dimer	0.465	0.328	-0.078	-0.240
Ferritin	0.272	0.382	-0.104	0.871
PCT	0.437	-0.149	-0.033	-0.158
CRP	0.452	-0.348	0.100	0.028

Table 6.10: Sparse Loadings

The table 6.11 contains the value of the 4 principal components for each of the 255 observations. For ease, only the first 3 rows and the last 4 rows are shown.

	PC1	PC2	PC3	PC4
[1]	-0.391211539	0.256381409	-0.581263199	0.534812154
[2]	-0.760147753	-0.113767187	0.212418854	-0.626841260
[3]	-1.252122404	-0.415225577	-0.471617330	-0.989049775
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
[252]	-1.017234109	-0.060005649	1.135507744	-0.811658844
[253]	-1.061947263	-0.661356367	-0.358546495	-0.905743936
[254]	-0.481728571	-0.002339508	-0.022733616	0.458489501
[255]	6.842567451	2.328974985	-0.693119636	-1.321060452

Table 6.11: ROBSPCA values per observation

6.1.3 Randomized Sparse Principal Component Analysis (RSPCA) on Clinical Tests (COVID-19) Data Set

We used R-studio to apply the RSPCA algorithm on Clinical Tests (COVID-19) Data Set through the following function

```
RSPCA <- rspca(X, k=4, alpha=1e-4, beta=1e-6, center = TRUE,
   scale = T,verbose=0).
```

For this function, several values for alpha and beta were tested, which are the Sparsity controlling parameter and the Amount of ridge shrinkage to apply in order to improve conditioning and it was found that with `alpha` $= 1 \times 10^{-3}$ and `beta` $= 1 \times 10^{-6}$ the greatest amount of variance is reached, for which we worked with those values.

In this case we chose 4 principal components since with it a cumulative proportion of the variance of 0.910 is reached. As you can see in the table 6.12, the first principal component has an explained variance of 3.428, which is equivalent to a variance proportion of 0.49, while the second, third, and fourth principal components have an explained variance of 1.232, 0.981 and 0.726, which represents 0.176 , 0.140, and 0.104 of the proportion of variance respectively.

	PC1	PC2	PC3	PC4
Explained variance	3.428	1.232	0.981	0.726
Standard deviations	1.851	1.110	0.991	0.852
Proportion of variance	0.490	0.176	0.140	0.104
Cumulative proportion	0.490	0.666	0.806	0.910

Table 6.12: RSPCA Summary

In the figure 6.6 we have on the left side the graph of the variances for each principal component. It is easy to see that as the number of principal components increases, the variance associated with each one decreases, with the first component having the maximum variance and the last component the minimum variance. While on the right side we have the graph of the proportion of variances by principal component.

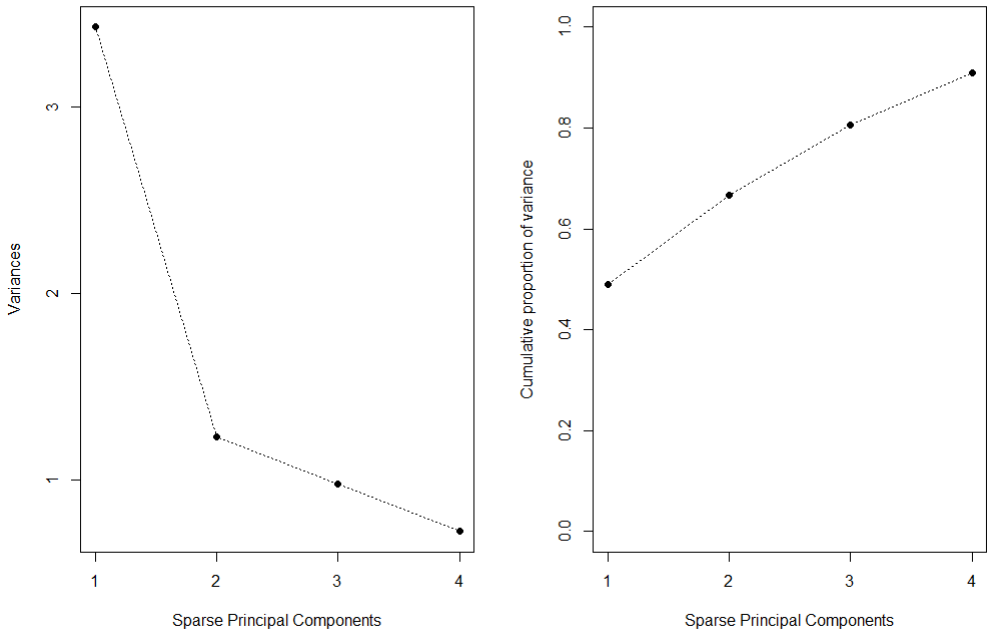

Figure 6.6: Variances of the Principal components

As with the variances, each principal component has an associated eigenvalue (see table 6.13) where the first component has the largest eigenvalue and the fourth component has the smallest eigenvalue.

	PC1	PC2	PC3	PC4
Eigenvalues	3.428	1.232	0.981	0.726

Table 6.13: Eigenvalues of Principal Components

From the table 6.14 we can interpret that the age variable is contained in the third principal component since it has a loading value of -0.963 in this component, which is the largest in magnitude of all the loadings associated with this variable. The IGG, D_Dimer, PCT, CRP variables are mostly contained in the first principal component since they have a higher loading value in that component than in the others. The IGM variable is mostly contained in the second principal component with a loading of -0.648 which is by magnitude the largest of all loadings. Finally, the Ferritin variable is in the fourth principal component since it has a loading of -0.871.

	PC1	PC2	PC3	PC4
Age	-0.050	0.253	-0.963	0.018
IGM	-0.313	-0.648	-0.157	-0.244
IGG	-0.457	0.341	0.137	0.311
D_Dimer	-0.463	0.331	0.077	0.239
Ferritin	-0.273	0.382	0.104	-0.871
PCT	-0.445	-0.151	0.036	0.163
CRP	-0.450	-0.348	-0.102	-0.029

Table 6.14: Sparse Loadings

The table 6.15 contains the value of the 4 principal components for each of the 255 observations. For ease, only the first 3 rows and the last 4 rows are shown

	PC1	PC2	PC3	PC4
[1]	0.3901113	0.256891946	0.581803159	-0.533765938
[2]	0.7596963	-0.114215135	-0.212188864	0.627268628
[3]	1.256437	-0.415400658	0.470605094	0.986619071
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
[252]	1.021162	-0.060467513	-1.136850643	0.808977039
[253]	1.067673	-0.662675231	0.356996872	0.902098419
[254]	0.4836932	-0.003275687	0.022091328	-0.459852910
[255]	-6.867467	2.340572065	0.703300016	1.338834810

Table 6.15: RSPCA values per observation

6.1.4 Principal Component Analysis

We used R-studio to apply the PCA algorithm on Clinical Tests (COVID-19) Data Set through the following function

```
pca <- prcomp(X, scale = TRUE)
```

In this case we chose 4 principal components since with it a cumulative proportion of the variance of 0.907 is reached. As you can see in the table 6.16, the first principal component has an explained variance of 3.429, which is equivalent to a variance proportion of 0.4898, while the second, third, and fourth principal components have an explained variance of 1.233, 0.981 and 0.726, which represents 0.1761 , 0.1402, and 0.1037 of the proportion of variance respectively.

	PC1	PC2	PC3	PC4
Explained variance	3.429	1.233	0.981	0.726
Standard deviations	1.8517	1.1102	0.9906	0.8520
Proportion of variance	0.4898	0.1761	0.1402	0.1037
Cumulative proportion	0.4898	0.6659	0.8061	0.9098

Table 6.16: PCA Summary

In the figure 6.7 we have the plot of the percentage of explained variances for each principal component. It is easy to see that as the number of principal components increases, the percentage of explained variance associated with each one decreases, with the first component having the maximum percentage of explained variance and the last component the minimum percentage of explained variance.

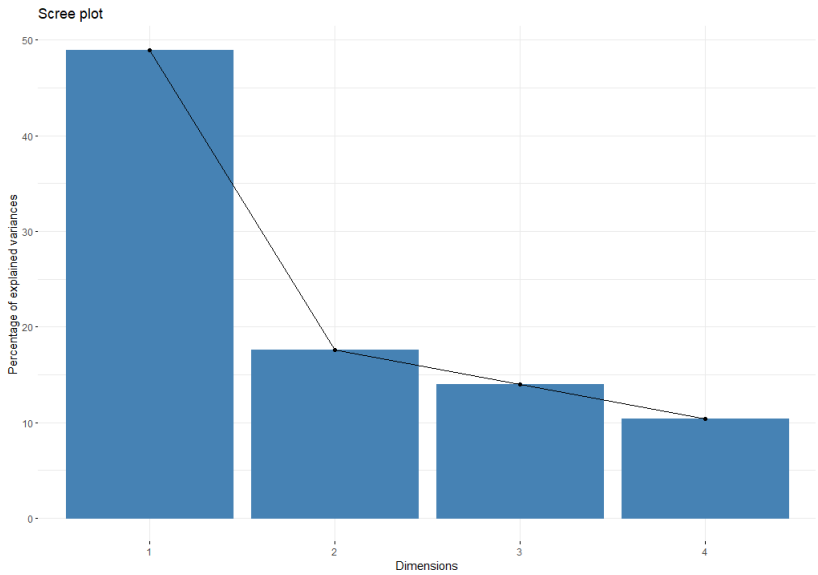

Figure 6.7: Percentage of explained variances of the Principal components

As with the variances, each principal component has an associated eigenvalue (see table 6.17) where the first component has the largest eigenvalue and the fourth component has the smallest eigenvalue.

	PC1	PC2	PC3	PC4
Eigenvalues	3.42889178	1.23260141	0.98128957	0.72596439

Table 6.17: Eigenvalues of Principal Components

In this case, note that the values of the PCA loadings are not very close to zero (see table 6.18), which would make it difficult for us to interpret the principal components.

From the table 6.18 we can interpret that the age variable is contained in the third principal component since it has a loading value of 0.9633 in this component. The IGG, D_Dimer, PCT, CRP variables are mostly contained in the first principal component since they have a higher loading value in that component than in the others. The IGM variable is mostly contained in the second principal component with a loading of -0.6476 which is by magnitude the largest of all loadings. Finally, the Ferritin variable is in the fourth principal component since it has a loading of 0.8710.

	PC1	PC2	PC3	PC4
Age	0.05013917	0.2532077	0.96333062	-0.0179284
IGM	0.31265592	-0.6476940	0.15706579	0.2437451
IGG	0.45734890	0.3408251	-0.13675849	-0.3112044
D_Dimer	0.46357749	0.3315088	-0.07750017	-0.2391182
Ferritin	0.27268890	0.3825627	-0.10452665	0.8710209
PCT	0.44544959	-0.1510817	-0.03651889	-0.1634349
CRP	0.45041720	-0.3480497	0.10176369	0.0292033

Table 6.18: Loadings

The table 6.19 contains the value of the 4 principal components for each of the 255 observations. For ease, only the first 3 rows and the last 4 rows are shown

	PC1	PC2	PC3	PC4
[1]	-0.3902631073	0.256920130	-0.581970768	0.533968268
[2]	-0.7599061552	-0.114307083	0.212392772	-0.627409720
[3]	-1.2569242315	-0.415636288	-0.470347709	-0.986626156
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
[252]	-1.0212839382	-0.060373000	1.137380276	-0.809228463
[253]	-1.0680870016	-0.662952464	-0.356664045	-0.902047658
[254]	-0.4837753506	-0.003235085	-0.021988480	0.460062101
[255]	6.8690540559	2.341329572	-0.705770519	-1.340739167

Table 6.19: PCA values per observation

Despite the fact that there is no significant difference between the variances achieved with four principal components by the SPCA, ROBSPCA and RSPCA algorithms with respect to the PCA for this data set, the three sparse PCA algorithms present a great advantage over the PCA algorithm which It can be evidenced when calculating the loadings since these will show more null or close to zero loads than the PCA.

	min	lq	mean	median	uq	max	neval
SPCA	37.3108	40.04265	**56.959806**	45.11960	67.96215	155.7481	100
RSPCA	1.7048	1.82175	**2.636254**	2.11740	3.08285	11.1738	100
ROBSPCA	1.1232	1.25655	**2.063722**	1.41335	2.21515	16.9791	100

Table 6.20: Computation time of the algorithms

The SPCA, RSPCA and ROBSPCA algorithms were executed in Rstudio using a 7^{th} *generation Intel core i5* processor computer with *8gb of RAM memory* and a *256gb solid state drive*. In the table 6.20 we can see that using the aforementioned computer, the SPCA algorithm was the one that had the longest running time with 56.9598ms. While the RSPCA and ROBSPCA algorithms have similar running time averages of 2.6362ms and 2.0637ms respectively. However, the RSPCA presents less dispersion, which gives it a great advantage in computational terms.

Chapter 7

Conclusions and Future Work

As a conclusion, we have that the Multivariate Data Analysis Techniques applied throughout this thesis, such as Sparse PCA and PCA, are very useful to deal with a huge amount of data, since they help us to reduce the size of the data and allow us to facilitates interpretation.

We applied with R the SPCA, ROBSPCA and RSPCA algorithms to the data set related to clinical tests of COVID-19 where approximately 91% of the initial data with only 4 principal components was obtained. Similar results were also obtained with the PCA algorithm. However, the Sparse PCA algorithms had a slight advantage over the PCA algorithm for these data since they generated more null loadings, which facilitated the interpretation of the principal components.

An interesting idea to be able to see the potential of Sparse PCA would be to apply it to a data set with more variables, such as gene arrays, which have been extensively worked in recent times. By having more variables and observations, it would make more sense to carry out dimension reduction analysis techniques such as the Sparse PCA in this case.

When there is multicollinearity problems in the variables, the Sparse PCA helps us to eliminate this problem by creating uncorrelated principal components. As future work, we could implement a generalized linear logistic model that helps us determine based on the principal components. the result, either positive or negative, of the clinical tests for COVID-19.

Bibliography

[1] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the lasso," *Journal of computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003.

[2] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[3] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proceedings of the National Academy of Sciences*, vol. 97, no. 18, pp. 10 101–10 106, 2000.

[4] K. S. Gurumoorthy, A. Rajwade, A. Banerjee, and A. Rangarajan, "A method for compact image representation using sparse matrix and tensor projections onto exemplar orthonormal bases," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 322–334, 2009.

[5] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE signal processing magazine*, vol. 8, no. 4, pp. 14–38, 1991.

[6] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[7] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[8] ——, "Rotation of principal components: choice of normalization constraints," *Journal of Applied Statistics*, vol. 22, no. 1, pp. 29–35, 1995.

[9] R. Hausman Jr, "Constrained multivariate analysis. optimisation in statistics (zanckis, sh and rustagi, js, eds.), 137-151," 1982.

[10] S. Vines, "Simple principal components," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 49, no. 4, pp. 441–451, 2000.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[12] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[13] J. Cadima and I. T. Jolliffe, "Loading and correlations in the interpretation of principle compenents," *Journal of applied Statistics*, vol. 22, no. 2, pp. 203–214, 1995.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2009. [Online]. Available: https://books.google.com.ec/books?id=eBSgoAEACAAJ

[15] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[16] J. Jackson and A. Edward, "User's guide to principal components. john willey sons," *Inc., New York*, vol. 40, 1991.

[17] A. C. Rencher and W. F. Christensen. (2012) Methods of multivariate analysis. Hoboken, New Jersey. [Online]. Available: http://www.amazon.de/Methods-Multivariate-Analysis-Probability-Statistics/dp/0470178965

[18] G. Tapia-Riera, L. Riera-Segura, C. Calle-Cárdenas, I. R. Amaro, and S. Infante, "Assessing the covid-19 vaccination process via functional data analysis," in *Information and Communication Technologies*, J. Herrera-Tapia, G. Rodriguez-Morales, E. R. Fonseca C., and S. Berrezueta-Guzman, Eds. Cham: Springer International Publishing, 2022, pp. 152–170.

[19] L. Riera-Segura, G. Tapia-Riera, I. R. Amaro, S. Infante, and H. Marin-Calispa, "Hj-biplot andÂ clustering toÂ analyze theÂ covid-19 vaccination process ofÂ american andÂ european countries," in *Smart Technologies, Systems and Applications*, F. R. Narváez, J. Proaño, P. Morillo, D. Vallejo, D. González Montoya, and G. M. Díaz, Eds. Cham: Springer International Publishing, 2022, pp. 383–397.

[20] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[21] M. Cubilla-Montilla, A. B. Nieto-Librero, M. P. Galindo-Villardón, and C. A. Torres-Cubilla, "Sparse hj biplot: A new methodology via elastic net," *Mathematics*, vol. 9, no. 11, p. 1298, 2021.

[22] C. M. Cuadras, *Nuevos métodos de análisis multivariante*. CMC Editions Barcelona, Spain, 1996.

[23] S. Infante, L. Sánchez, and F. Cedeño, "Filtros para predecir incertidumbre de lluvia y clima," *Revista de Climatología*, vol. 12, pp. 33–48, 2012.

[24] S. Castro, "Analisis de datos en grandes dimensiones. estimacion y seleccion de variables en regresion." *Instituto de Estad ıstica (IESTA) y Departamento de M etodos Cuantitativo*, 2012.

[25] H. F. Kaiser, "The varimax criterion for analytic rotation in factor analysis," *Psychometrika*, vol. 23, no. 3, pp. 187–200, 1958.

[26] N. T. Trendafilov, "From simple structure to sparse components: a review," *Computational Statistics*, vol. 29, pp. 431–454, 2014.

[27] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[28] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773–795, 1995. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476572

[29] G. H. Golub and U. von Matt, "Generalized cross-validation for large-scale problems," *Journal of Computational and Graphical Statistics*, vol. 6, no. 1, pp. 1–34, 1997.

[30] R. Guerra-Urzola, K. Van Deun, J. C. Vera, and K. Sijtsma, "A guide for sparse pca: Model comparison and applications," *psychometrika*, vol. 86, no. 4, pp. 893–919, 2021.

[31] M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre, "Generalized power method for sparse principal component analysis." *Journal of Machine Learning Research*, vol. 11, no. 2, 2010.

[32] M. Hubert, P. J. Rousseeuw, and S. Van Aelst, "High-breakdown robust multivariate methods," 2008.

[33] N. T. Trendafilov and I. T. Jolliffe, "Projected gradient approach to the numerical solution of the scotlass," *Computational Statistics & Data Analysis*, vol. 50, no. 1, pp. 242–253, 2006.

[34] N. B. Erichson, P. Zheng, K. Manohar, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, "Sparse principal component analysis via variable projection," *SIAM Journal on Applied Mathematics*, vol. 80, no. 2, pp. 977–1002, jan 2020. [Online]. Available: https://doi.org/10.1137%2F18m1211350

[35] M. Enríquez, S. Naranjo, I. Amaro, and F. Camacho, "Dimensionality reduction using pca and cur algorithm for data on covid-19 tests," in *Artificial Intelligence, Computer and Software Engineering Advances: Proceedings of the CIT 2020 Volume 1*. Springer, 2021, pp. 121–134.

[36] N. T. Trendafilov, S. Unkel, and W. Krzanowski, "Exploratory factor and principal component analyses: some new aspects," *Statistics and Computing*, vol. 23, pp. 209–220, 2013.

[37] N. González-García and A. B. Nieto-Librero, "Sparse pca vs descomposición cur," *XXVI Simposio Internacional de Estadística*, 2016.

Appendices

Appendix A

R Codes for Results

A.1 Exploratory Analysis of Data

```
#Data
View(X)
summary(X)
var(X)
cov(X)
means<-c(mean(X$Age),mean(X$IGM),mean(X$IGG),mean(X$D_Dimer),mean(X$
    Ferritin),mean(X$PCT),mean(X$CRP))
means

#Correlation Matrix
M<-cor(X)
M
det(M)

#Standardized Data
library(dplyr)
set.seed(1)
X_STANDARDIZED <- X %>% mutate_all(~(scale(.) %>% as.vector))
X_STANDARDIZED
summary(X_STANDARDIZED)
var(X_STANDARDIZED)
means_STANDARDIZED<-c(mean(X_STANDARDIZED$Age), mean(X_STANDARDIZED$
    IGM),mean(X_STANDARDIZED$IGG), mean(X_STANDARDIZED$D_Dimer),mean(X_
    STANDARDIZED$Ferritin),mean(X_STANDARDIZED$PCT),mean(X_STANDARDIZED
    $CRP))
means_STANDARDIZED


```

```
#Violinplot
library("vioplot")
#Standardized Data
vioplot(X_STANDARDIZED, col = "gray", border = "black")
stripchart(X_STANDARDIZED, vertical = TRUE, method = "jitter",
           pch = 1, add = TRUE, col = "black")
abline(h = 0)
points(means_STANDARDIZED, col = "white", pch = 21, cex = 1, bg = "
    white", lwd = 2)

#real data
vioplot(X, col = "gray", border = "black")
stripchart(X, vertical = TRUE, method = "jitter",
           pch = 1, add = TRUE, col = "black")
points(means, col = "red", pch = 21, cex = 1.5, bg = 2, lwd = 2)

#Multicolinearity
library(ggplot2)
library(grid)
library(gridExtra)
library(corrplot)
corrplot(M, method="number", order="hclust", type="lower")
```

A.2 SPCA Algorithm on Clinical Tests (COVID-19) Data Set

```
# Compute SPCA
library(sparsepca)
out <- spca(X, k=4, alpha=1e-3, beta=1e-6, center = TRUE, scale = TRUE
    , verbose=0)
print(out)
summary(out)
SPCA<-out$scores
```

A.3 ROBSPCA Algorithm on Clinical Tests (COVID-19) Data Set

```
# Compute ROBSPCA
out2 <- robspca(X, k=4, alpha=1e-4, beta=1e-6, gamma=1, center = TRUE,
     scale = TRUE, verbose=0)
print(out2)
summary(out2)
ROBSPCA<-out2$scores
```

A.4 RSPCA Algorithm on Clinical Tests (COVID-19) Data Set

```
# Compute RSPCA
out3 <- rspca(X, k=4, alpha=1e-4, beta=1e-6, center = TRUE, scale = T,
     verbose=0)
print(out3)
summary(out3)
RSPCA<-out3$scores
```