Dimensionality Reduction Using PCA and CUR Algorithm for Data on COVID-19 Tests

Marco Enríquez, Samantha Naranjo, Isidro Amaro, and Franklin Camacho
School of Mathematical and Computational Sciences, Yachay Tech University, Urcúquí, Ecuador
naranjosamantha1j@gmail.com

Abstract. In this paper we present the results of two well known analyses, Principal Component Analysis and CUR algorithm, conducted on data related to tests of coronavirus, which were performed from May 17th to June 26th, 2020 in Ibarra, Ecuador. We analyzed the effectiveness of CUR over PCA and found out that, for our data matrix, CUR is more effective than PCA whenever the control parameters of the CUR algorithm $c$ and $k$ are equal. Furthermore, the results of CUR algorithm suggest that the laboratory tests D-dimer, ferritin and PCR are the most important variables.

Keywords: CUR algorithm · Principal Components Analysis · COVID-19

1 Introduction

The World Health Organization declared COVID-19 outbreak as a pandemic on 11 March 2020 and people’s lives have changed dramatically ever since. In Ecuador the number of cases is still increasing and, according to Ministerio de Salud Pública Ecuador, 182,523 tests have been performed until July 12th, 2020. There are two kinds of tests that can show us if a person is infected or not: reverse transcription polymerase chain reaction (RT-PCR) and serology tests (also known as rapid tests). The former is a diagnostic test that, from a respiratory sample of a person suspected of being infected, detects the presence of SARS-CoV-2 virus’ RNA, and the latter, through a blood sample, is able to detect antibodies generated by the immune system.

Generally, people who want to know if they have COVID-19 do not ask clinical laboratories to do the RT-PCR test; they prefer to use the serology tests instead because those are faster and affordable. However, relying solely on these rapid tests can be dangerous. According to Ministerio de Salud Pública Ecuador [1], serology tests may yield a type II error, which means that it is possible that the laboratory concludes some patients are not infected when they are. In some...
cases, people ask laboratories for other tests, which will be named and described in Sect. 2, besides the serology ones. When serology tests are not very reliable or accurate, could the other tests help us detect or at least give more information about the presence of the virus? This question motivates our research.

The purpose of this research is to compare the results given by the application of two multivariate statistical techniques: Principal Components Analysis (PCA) and CUR algorithm (also known as CUR matrix decomposition), to a database related to COVID-19 in Imbabura, Ecuador. This database consists of some laboratory tests. Currently, only two of them are being used to determine if a person is infected with SARS-CoV-2. From this point these laboratory tests will be referred to as variables.

Then, our main objective is to use multivariate statistical techniques to reduce the dimension of the problem to determine which of those variables are more significant. In order to achieve this, we have to analyze our data set using PCA and CUR algorithm, and then interpret the results obtained by both techniques. Moreover, we will determine which technique is more effective to reduce the dimension for our data set. Our hypothesis is that CUR algorithm is more effective than PCA. The effectiveness of these techniques will be measured with the Frobenius Norm.

If we successfully fulfill our main objective, then we will be able to determine which variables play a role in the detection of the SARS-CoV-2, especially when serology tests are not reliable. This research is important because it provides lab technicians with valuable information about the variables that can be used to diagnose this disease.

We took for granted our interaction with other species and as a result the new coronavirus has affected our lifestyle dramatically. For this reason, research on this topic has increased a lot recently. In particular, the multivariate statistical analysis techniques, such as regression tree, cluster analysis or PCA, have been used for the study of COVID-19 disease. For example, Rahaman Khan and Hossain, in [2], used these techniques for a COVID-19 data from 133 countries. For each country they considered the following variables: total confirmed cases, new confirmed cases, total deaths, total recovered patients, total active cases, total seriously critical patients, infection rate per million, death rate per million, total tests conducted, and test rate per million. Another example is the research of Kumar et al. [3]. They use correlation and PCA to study the relationship of coronavirus spread with socio-economic factors of different states of India using the Rstudio software\footnote{https://rstudio.com/}. Nonetheless, we were not able to find research on the use of CUR algorithm in some aspect of the COVID-19 disease. Hence, our research takes a different approach because our COVID-19 data is local from Ibarra, Ecuador, and the use of multivariate statistical analysis techniques, such as CUR algorithm is focused in clinical laboratory tests.

In this paper, in Sect. 2, we first describe our data matrix and its variables, then we give a brief description of PCA and CUR algorithm and how to determine which of these techniques is more effective. Second, in Sect. 3, we present
the main results of our work, which was done using R language\(^2\). Finally, in Sect. 4, we summarize the article.

## 2 Materials and Methods

In this section, first we describe the data set used, including information about the number of individuals and the type of variables. Then we explain the two techniques used for the analysis and the effectiveness of one against another.

### 2.1 Description of Data

The data set used in this research\(^3\) was obtained from a clinical laboratory in Ibarra, Ecuador, called Laboratorio Clínico - Clínica Ibarra. The tests of 256 individuals were performed from May 17th, 2020 to June 26th, 2020. The variables in this data are: Edad, IGG, IGM, Dimero D, Ferritina, PCT and PCR, but only two of them are specialized for COVID-19. Since this clinical laboratory is private, we asked its owner permission to use this data. We decided not to include the ID of the individuals because of privacy matters.

First, let’s explain the variables of our data set that are used to confirm or rule out the new strain of coronavirus: IGG and IGM. According to Portillo et al. [4], “rapid tests would be a solution that would speed up the identification and isolation of infected individuals, estimating more accurately the number of infections”. For that reason, in this study we consider the variables IGG and IGM, which are the serology tests. They both detect the immune response of the individual and thus the COVID-19 IgG and COVID-19 IgM antibodies. The Laboratorio Clínico - Clínica Ibarra applies a technique known as immunofluorescence for these tests.

The difference between IGG and IGM is that IGM reveals the immune response against active infection, while IGG indicates the presence of non-current infections; i.e., those that have already developed a secondary immune response. The time it takes to get the results, from when the patient’s blood is drawn, is the same for IGG and IGM, 1 h. That is why they are called rapid tests.

The results of these tests are qualitative, but they can be expressed in numerical form against a cut off index (COI). The range of IGG and IGM is greater than or equal to 0. According to Laboratorio Clínico - Clínica Ibarra, both test are negative, which means that the patient does not have COVID-19 when their values are less than or equal to 0.9. If the values of IGG and IGM are between 0.9 and 1.1, the result of the tests are indeterminate; i.e., it is not possible to conclude whether or not the patient has COVID-19. If the values of IGG and IGM are greater than or equal to 1.1, then the result of the test is positive and the patient has COVID-19.

\(^2\) [http://www.r-project.org/](http://www.r-project.org/).

\(^3\) If you need the data matrix, you can email the corresponding author.
Now, let’s briefly explain the other five variables. The variable *Edad* is the age of the individual and we considered it because in medical research, it has been shown to be an important variable, particularly in coronavirus studies. The variable *Dimero D* is a test that looks for the D-dimer in the blood, a protein fragment that is produced when a blood clot dissolves in the body. The method used in this test is heterogeneous immunoassay and its reference value range is $[0, 500]$ ng/mL. The variable *Ferritina* is a test that measures the level of ferritin in the blood. Ferritin is a protein that stores iron in cells. The immunofluorescence method was used to identified this protein and its reference value range is $[30, 350]$ ng/mL. The variable *PCT* is the procalcitonin test that measures the level of procalcitonin in the blood. A high level could be a sign of a serious bacterial infection, such as sepsis. As for this method, immunofluorescence was used. *PCT* is negative when is less than 0.5 ng/mL: the risk of severe sepsis and/or septic shock is low if *PCT* is between from 0.5 to 2 ng/mL, moderate if *PCT* is between from 2 to 10 ng/mL and high if *PCT* is greater or equal than 10 ng/mL. Finally, the variable *PCR*, also known as Ultrasensitive CRP (that must not be mistaken for RT-PCR), tell us if C-reactive protein is secreted by a current infection or inflammation. The method used for this test is the automated photometry. Ultrasensitive CRP is measured in mg/L. The reference values for this test is $[0, 5]$ mg/L.

### 2.2 Techniques for the Analysis

In many disciplines and data analysis applications, the use of data matrices with a huge number of columns or rows is common. But sometimes, the studies with a large dimension of the data matrix has dissimilar or contradictory results. In such cases, it is necessary to employ some dimensionality reduction method in order to avoid wrong conclusions. In this research, we use only two of these techniques: PCA and CUR algorithm. So, before to review them, let’s first briefly recall two main definitions.

For a matrix $M \in \mathbb{R}^{n \times p}$ the Frobenius norm is defined as

$$\|M\|_F = \sqrt{tr(MM^T)},$$

where $tr(A)$ is the trace of a square matrix $A \in \mathbb{R}^{p \times p}$ defined by $tr(A) = \sum_{i=1}^{p} A_{i,i}$, the sum of its diagonal elements.

Also, let’s recall Singular Value Decomposition (SVD). Given $X \in \mathbb{R}^{n \times p}$, there exists orthogonal matrices $U = [u^1 | u^2 | \cdots | u^n] \in \mathbb{R}^{n \times n}$ and $V = [v^1 | v^2 | \cdots | v^p] \in \mathbb{R}^{p \times p}$, where $\{u^i\}_{i=1}^{n} \subseteq \mathbb{R}^{n}$ and $\{v^j\}_{i=1}^{p} \subseteq \mathbb{R}^{p}$ are such that

$$U^T XV = \Sigma = diag(\sigma_1, \cdots, \sigma_\rho),$$

where $\Sigma \in \mathbb{R}^{n \times p}$, $\rho = \min\{n, p\}$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\rho \geq 0$, and $diag(\cdot)$ represents a diagonal matrix with the specified elements on the diagonal.

**Principal Component Analysis.** This is one of the most popular multivariate statistics technique to reduce the data dimension and boils down to the SVD.
We use PCA in our data set because it not only reduces the data dimension but also gives us information about the behavior of the variables. According to Rencher and Christensen [5], the idea of this method is to create a new kind of variables called principal components (PC). The principal components associated with the largest singular values have strong optimality properties, and they can often be quite useful as a tool to summarize and identify major patterns of the data. The $i^{th}$, $i = 1, \ldots, p$, principal component is defined by

$$PC_i = Xv_i = v_{1,i}\text{col}(1, X) + v_{2,i}\text{col}(2, X) + \cdots + v_{p,i}\text{col}(p, X),$$

where $v_{j,i}$ is the $j^{th}$ component of the $i^{th}$ right singular vector the data matrix $X \in \mathbb{R}^{n \times p}$, which contains $n$ observations of a random vector $\mathbf{X} = (X_1, \ldots, X_p)^T$.

Note that $X_i$ is the random variable whereas $\text{col}(i, X)$ is a collection of $n$ observations of this random variable. Thus the columns of our new data matrix $D$ with lower dimension are the observations of the random variables

$$PC_i = v_{1,i}X_1 + v_{2,i}X_2 + \cdots + v_{p,i}X_p.$$ (4)

However, the ability to interpret the random variables $X_i$, or equivalently the columns of $X$, in no way guarantees an ability to interpret linear combinations of them.

**CUR Algorithm.** The CUR matrix decomposition is also a dimensionality reduction method that was developed by Mahoney and Drineas [6]. We use this technique in our data set because it can solve the disadvantage of PCA with the interpretation. The idea behind CUR is to decompose the data matrix $X$ into the product of three matrices $C$, $U$ and $R$ such that

$$X \approx CUR,$$ (5)

where the $C$ matrix contains a small number of columns from $X$, the $R$ matrix contains a small number of rows of $X$ and the matrix $U$ is the product of their Moore-Penrose pseudo inverses with $X$; that is $U = C^+XR^+$. The role of $U$ is to guarantee that the product $CUR$ is close to $X$. Since matrix $C$ is built from the original variables of $X$, CUR matrix decomposition can be easily interpreted by practitioners of the field from which the data is drawn.

Although CUR matrix decomposition and CUR algorithm refer to the same, in this paper we establish a difference between them. Here, the R code implementation of the CUR matrix decomposition is referred to as CUR algorithm. This implementation is in the rCUR package, which is available for free in an article of Bodor et al. [7]. In the rCUR package, a primary function that makes the decomposition called ‘CUR’ is implemented. The inputs of this function are the data matrix $X$, the column parameter $c$, the row parameter $r$ and the rank parameter $k$. The function ‘CUR’ returns a CURobj-class object.

---

4 We are using the version 1.3 of rCUR that was updated on 2012-07-02.
Recall that $C$ matrix contains a small number of columns from $X$. To select the columns of $X$ to include in $C$ (and similarly for the row selection of $R$), we have to compute an importance score for each columns of $X$. Then, by using the rank parameter $k$ and the data matrix $X$, the CUR algorithm randomly samples a small number of columns from $X$ using that score as an importance sampling probability distribution. That is, for the $i^{th}$ column $X$, ‘CUR’ associates a number $l_i$ such that the collection $\{l_i\}_{i=1}^p$ forms a probability distribution. These $l_i$ are called the normalized statistical leverage scores. If $V_{i,j}$ is the $i^{th}$ element of the $j^{th}$ right singular vector $X$, then the leverage scores equal

$$l_i = \frac{1}{k} \sum_{j=1}^k V_{i,j}^2,$$  \hspace{1cm} (6)

for all $i = 1, \ldots, p$. The function ‘leverage’ retrieve column leverage scores from CURobj-class.

Using the leverage scores the algorithm makes a pass over the columns and selects the $i^{th}$ columns with a probability $\min\{cl_i, 1\}$. So the algorithm takes the leverage scores and scales them by the column parameter. If $cl_i < 1$ then the algorithm chooses that column as important with probability $cl_i$. If $cl_i > 1$ then the algorithm always chooses that column as important, i.e., with probability 1. Once the algorithm has chosen a set of columns it creates a matrix $C$ whose columns are those selected columns. The function ‘getC’ retrieve matrix $C$ from CURobj-class.

According to Drineas, Mahoney and Muthukrishnan [8], with probability at least 99%, this choice of columns satisfies the following

$$\|X - P_cX\|_F \leq (1 + \epsilon/2) \|X - X_k\|_F,$$  \hspace{1cm} (7)

where $P_c$ denotes a projection matrix onto the column space of $C$ and $\epsilon$ is an error parameter. The proof of (7) (see [8]) depends crucially on the use of (6). Thus, by using (7), Mahoney and Drineas [6] prove with probability at least 98% the following

$$\|X - CUR\|_F \leq (2 + \epsilon) \|X - X_k\|_F,$$  \hspace{1cm} (8)

where $X_k = \sum_{i=1}^k \sigma_i u_i v_i^T$.

In [7], Bodor et al. make a suggestion to improve efficiency by switching off the computation of components that are not used. So we make that the row parameter $r$ be the set to almost the actual numbers of rows of $X$. The function ‘getR’ retrieve matrix $R$ from CURobj-class.

Hence, based on Hunt [9], the steps of the CUR algorithm can be written as follows:

1. Get data matrix $X$ and the two parameters $c$ and $k$.
2. Calculate the leverage scores $\{l_i\}_{i=1}^p$ using $X$ and $k$.
3. Make a pass over the columns of the matrix and keep the $i^{th}$ columns with a probability $\min\{cl_i, 1\}$. 


4. If we have chosen at least one column, return a matrix $C$ consisting of those chosen columns. If we have not chosen any column, then go back to step 3.

Similarly to PCA, the way in which one decides how to choose dimensionality we should reduce the data is not a question CUR algorithm can answer. However, one of the conclusions of Hunt [9] is that, if we want to use CUR algorithm in place of retaining $m$ principal components, we should run the function CUR with $c = m$ and $k = m$.

### 2.3 Effectiveness

The performance of PCA and CUR algorithm depends on the data matrix $X$ and on the control parameters $c$ and $k$ of the CUR algorithm. So, we have to know which of both techniques is better for our data set. To reduce the dimensionality of the data matrix implies to throw away part of the data. Hence, to measure this loss, Hunt [9] propose a statistic to measure the performance of CUR algorithm relative to PCA by “computing how much variance we lose, or equivalently, how much variance we retain” for each method.

Recall that PCA re-expresses the data in terms of a new data matrix $D$. By using (4), we construct $D$, the comprised matrix of $k$ principal components of $X$. Thus, the effectiveness of CUR algorithm relative to PCA, by using the Frobenius norm, is

$$e = \frac{\|C\|_F^2 - \|D\|_F^2}{\|X\|_F^2}.$$  \hspace{1cm} (9)

Since “neither CUR algorithm nor PCA can do better than retaining 100% more variance than the other”, then $-1 < e < 1$. Therefore, if $e < 0$ then $\|C\|_F^2 < \|D\|_F^2$ which means that CUR algorithm retains less of the variance than PCA. Similarly, if $e > 0$ then $\|C\|_F^2 > \|D\|_F^2$ which means that CUR algorithm retains more of the variance than PCA.

### 3 Results and Discussions

In this section we summarize the results that were found after running PCA and CUR algorithm on our data matrix in R language\(^5\).

#### 3.1 Principal Component Analysis on Data Matrix

Implementation of PCA in R is straightforward. We have applied PCA on the original data matrix $X$, which has 7 columns and 256 rows. In order to run PCA, R program uses the correlation matrix. In our case, the determinant of the correlation matrix is $0.02606074$ which is relatively close to 0. This means that the variables of our data are correlated. We can see that the cumulative proportion of the third principal component amounts to $0.8023$ (see Table 1) of

\(^5\) If you want to replicate the results, you can email the corresponding author.
the total variance. Since the criteria we are using to retain components is to account 80% of the variance, we take PC1, PC2, PC3. We get the same result if instead of using the previous criteria, we take principal components such that their eigenvalues are greater than or equal to 1 (see Fig. 1). In any case, we retain three principal components.

**Table 1. Summary of PCA on the original data matrix**

|                  | PC1   | PC2   | PC3   | PC4   | PC5   | PC6   | PC7   |
|------------------|-------|-------|-------|-------|-------|-------|-------|
| Standard deviation| 1.8355| 1.1247| 0.9908| 0.8408| 0.59853| 0.4443| 0.34868|
| Proportion of variance | 0.4813 | 0.1807 | 0.1403 | 0.1010 | 0.05118 | 0.0282 | 0.01737 |
| Cumulative proportion | 0.4813 | 0.6620 | 0.8023 | 0.9032 | 0.95443 | 0.9826 | 1.00000 |

**Fig. 1.** Scree plot: the numbers in the abscissa represent the PC

PCA also gives us the coordinates of each of the seven variables using the principal components, which are given in Table 2. These coordinates, along with the coordinates of each individual and the retained PC (see Table 3), are used to get the plot shown in Fig. 2.

In Fig. 2 we can see that people on the left hand side of the plot are similar to each other and that they are people who tested negative for COVID-19. We can also notice that all of the seven variables are positively correlated; i.e. if one of the variables is high and far from its normal value, then it is more likely that the other variables have high values as well. On one hand, there is a high positive correlation among Ferritina, Dimero D and IGG; similarly among PCT, PCR and IGM. On the other hand, Edad has a high correlation among Ferritina, Dimero D and IGG, but it has a relatively low correlation among PCT and PCR. Moreover, the correlation between Edad and IGM is significantly low; that is, its value is rather close to zero. This shows that the higher the age the higher the values of these variables. We can further notice that people on the
Table 2. Coordinates of each variable in terms of PC

|       | PC1     | PC2     | PC3     | PC4     | PC5     | PC6     | PC7     |
|-------|---------|---------|---------|---------|---------|---------|---------|
| Edad  | 0.0516  | 0.2320  | 0.9687  | −0.0206 | 0.0584  | −0.0086 | 0.0344  |
| IGM   | 0.3161  | −0.6276 | 0.1367  | 0.3134  | −0.1580 | −0.5612 | 0.2218  |
| IGG   | 0.4502  | 0.2878  | −0.1151 | −0.4238 | −0.1915 | −0.0039 | 0.6962  |
| Dimero D | 0.4588 | 0.3574  | −0.0779 | −0.1447 | −0.2774 | −0.4098 | −0.6242 |
| Ferritina | 0.2807 | 0.4338  | −0.1113 | 0.8193  | 0.0977  | 0.1342  | 0.1471  |
| PCT   | 0.4516  | −0.1460 | −0.0403 | −0.1705 | 0.8554  | 0.0296  | 0.1066  |
| PCR   | 0.4486  | −0.3633 | 0.0976  | 0.0139  | −0.3415 | 0.7057  | −0.2052 |

Table 3. Matrix $D$ obtained from PCA: we only show the first three and last three rows of the matrix

|       | PC1     | PC2     | PC3     |
|-------|---------|---------|---------|
| 1     | −0.4112 | $2.5987 \times 10^{-1}$ | −0.5766 |
| 2     | −0.7519 | $1.3912 \times 10^{-1}$ | 0.2170  |
| 3     | −1.2401 | $4.3013 \times 10^{-1}$ | −0.4711 |
| ...   |         |         |         |
| 254   | −1.0526 | −0.6707 | −0.3653 |
| 255   | −0.4939 | −0.0055 | −0.0225 |
| 256   | 6.6718  | 2.1108  | −0.6041 |

Fig. 2. PCA on original matrix
upper right of the graph have high values of Ferritina, Dimero D and IGG while people on the bottom right have high values of PCT, PCR and IGM.

Up till now, we have retained three principal components. Nonetheless, we still do not know which variables we should get rid off, which is precisely why we need to implement CUR algorithm.

### 3.2 CUR Algorithm on Data Matrix

Unlike PCA, we must provide CUR algorithm the values of $c$ and $k$. To improve the efficiency of CUR algorithm with respect to PCA, we ran the function ‘CUR’ with input parameters $c = k = 3$, following the suggestion of Hunt [9] explained before. Recall that the data matrix is decomposed into three matrices $C$, $U$, and $R$. We can see that the function ‘getR’ returned a matrix $R$ of dimension $134 \times 7$ (see Table 4); that is, 119 individuals of the original matrix were omitted because they do not provide as much information (low row leverage scores) as the other entries. In fact, this can be seen in Fig. 3, which is the plot of the row leverage scores.

|   | Edad | IGM | IGG | Dimero D | Ferritina | PCT  | PCR   |
|---|------|-----|-----|----------|-----------|------|-------|
| 4 | 52   | 1.6 | 0.00| 864.20   | 246.50    | 0.50 | 50.12 |
| 5 | 33   | 1.2 | 0.00| 2456.50  | 368.45    | 1.50 | 145.20|
| 8 | 34   | 0.1 | 0.00| 321.45   | 302.20    | 0.20 | 0.22  |
|   |      |     |     |          |           |      |       |
| 250|70   | 0.2 | 0.00| 385.62   | 318.56    | 0.20 | 1.50  |
| 251|12   | 0.3 | 0.00| 107.20   | 145.41    | 0.09 | 2.60  |
| 256|50   | 0.5 | 0.00| 6285.84  | 380.10    | 2.00 | 35.10 |

When it comes to matrix $C$, we can see that it provides us with more information about dimensionality reduction. Figure 4 is a plot of the column leverage scores that were retrieved using the function ‘leverage’. On one hand, Edad, Dimero D, Ferritina and PCR have the highest values. On the other hand, PCT, IGM and IGG have the lowest values. This fact proves that the variables IGM and IGG are not very reliable clinical tests. Note that the leverage score of Edad is greater than that of PCR. As we choose the column parameter $c = 3$, the most natural thing would be to think that the CUR algorithm will choose the variables with highest leverage score: Edad, Dimero D and Ferritina. However, the function ‘getC’ returns a $256 \times 3$ matrix, whose columns are Dimero D, Ferritina and PCR (see Table 5). The variable Edad was omitted by the algorithm. This happens because when PCA was performed Ferritina, Dimero D, and PCR
turned out to be more important for the first PC while Edad is more important for the second PC.

In order to show that the choice of the three variables was correct, we run a PCA on the matrix $C$ and the matrix given by the four variables (see Fig. 5 and Fig. 6), and computed the cophenetic correlation coefficients. With the former matrix we got a coefficient of 0.995, but with the later the coefficient is 0.955. This implies that Dimero D, Ferritina and PCR are the most important variables. Furthermore, on Fig. 5 we see that Ferritina, Dimero D and PCR are positively correlated and that the proportion of variance is high. Unlike Fig. 5, Fig. 6 shows that when we add Edad the variance is lower and that this variable is barely related with the other three variables.
Table 5. Matrix $C$ for our data matrix: we only show the first three and last three rows of the matrix

|   | Dimero D | Ferritina | PCR |
|---|----------|-----------|-----|
| 1 | 265.85   | 246.15    | 4.57|
| 2 | 145.65   | 102.23    | 4.63|
| 3 | 41.90    | 45.00     | 0.00|
|   | ...      |           |     |
| 254| 165.20  | 45.00     | 0.90|
| 255| 87.10   | 216.80    | 3.68|
| 256| 6285.84 | 380.10    | 35.10|

Fig. 5. Plot of PCA on matrix C

3.3 Effectiveness

We have already run PCA and CUR algorithm on the data matrix. It remains to see which method was more effective. Recall that here matrix $X$ is the data matrix and $D$ is the matrix obtained by performing PCA. Using (9) we get that $e = 0.9992723$, which is rather close to 1. This implies that, for our data matrix, CUR algorithm excels in dimensionality reduction, i.e., it is better than PCA.
Fig. 6. Plot of PCA on the matrix with four variables: Edad, Dimero D, Ferritina and PCR

4 Conclusions

Throughout this paper, we focused on dimensionality reduction techniques and how we can use them to contribute more information about the detection of the virus SARS-CoV-2, which yields in COVID-19 disease. We considered two multivariate statistical techniques, PCA and CUR algorithm. In order reduce the dimension, PCA creates new variables called principal components while CUR algorithm removes the less important variables based on their leverage scores. Unlike PCA and whenever it is possible, CUR algorithm allows us to have a better interpretation of the results because matrix $C$ is built from the original variables.

After applying PCA to our data matrix, we found out that it is possible to retain three principal components, but they do not give us much information about the variables. Then we ran CUR algorithm with input parameters $c = k = 3$. Three variables were chosen by the CUR algorithm. This variables are Dimero D, Ferritina and PCR. Furthermore, for our data matrix, CUR retains more variance than PCA. This means that CUR algorithm was more effective than PCA.

This exploratory data analysis led us to the conclusion that the laboratory tests D-dimer, ferritin and CRP are the most significant in our data set and they come in handy in the detection of SARS-CoV-2. For that reason and taking into account that COVID-19 is a new disease, we suggest for clinical research related to the detection of SARS-CoV-2 to consider them. We did not appraise other values of $c$ and $k$ other than the number of retained principal components.
Analyzing how the effectiveness of both techniques behaves when those values vary and using a bigger data matrix may constitute the object of future studies.

Acknowledgements. The data set was thankfully provided by Laboratorio Clínico - Clínica Ibarra. This research would not have been possible without their help. The authors also thank Vicecancillería de Investigación of Yachay Tech University for its partial support through the Project Decisiones Cualitivas.

References

1. Avilés, E., et al.: Protocolo de uso de pruebas rápidas para detección de anticuerpos contra SARS-COV-2/COVID-19. Ministerio de Salud Pública Ecuador (2020). https://www.salud.gob.ec/wp-content/uploads/2020/04/Protocolo-de-uso-de-pruebas-r%C3%A1pidas-para-detecci%C3%B3n-de-anticuerpos-contra-Sars-Cov-2Covid-19_v2.20.04.2020.pdf. Accessed 21 Aug 2020

2. Rahaman Khan, H., Hossain, A.: Countries are clustered but number of tests is not vital to predict global COVID-19 confirmed cases: A machine learning approach. medRxiv (2020). https://doi.org/10.1101/2020.04.24.20078238. https://www.medrxiv.org/content/early/2020/04/29/2020.04.24.20078238

3. Kumar, A., Rani, P., Kumar, R., Sharma, V., Purohit, S.R.: Data-driven modelling and prediction of COVID-19 infection in India and correlation analysis of the virus transmission with socio-economic factors. Diabetes Metab. Synd. Clin. Res. Rev. 14(5), 1231–1240 (2020). https://doi.org/10.1016/j.dsx.2020.07.008. http://www.sciencedirect.com/science/article/pii/S187140212030254X

4. Portillo, R.V., et al.: Pruebas rápidas para COVID-19, la mejor alternativa para Ecuador. Bionatura (2020). https://doi.org/10.21931/RB/2020.05.03.21

5. Rencher, A., Christensen, W.: Methods of Multivariate Analysis, 3rd edn. John Wiley & Sons, Inc., New York (2012). https://doi.org/10.1002/9781118391686

6. Mahoney, M.W., Drineas, P.: Cur matrix decompositions for improved data analysis. Proc. Nat. Acad. Sci. 106(3), 697–702 (2009). https://doi.org/10.1073/pnas.0803205106. https://www.pnas.org/content/106/3/697

7. Bodor, A., Csabai, I., Mahoney, M.W., Solymosi, N.: rCUR: an R package for CUR matrix decomposition. BMC Bioinf. 13(1), 1–6 (2012)

8. Drineas, P., Mahoney, M.W., Muthukrishnan, S.: Relative-error cur matrix decompositions. SIAM J. Matrix Anal. Appl. 30(2), 844–881 (2008). https://doi.org/10.1137/07070471X

9. Hunt, G.J.: CUR: An interpretable alternative to principal components analysis. Ph.D. thesis, Drew University (2013)