SOCIETY OF ACTUARIES
MODELING SECTION

ACTUARIAL TECHNOLOGY TODAY
MAY 2020

Principal Component Analysis Using R

By Soumava Dey

In today's Big Data world, exploratory data analysis has become a stepping stone to discover underlying data patterns with the help of visualization. Due to the rapid growth in data volume, it has become easy to generate large dimensional datasets with multiple variables. However, the growth has also made the computation and visualization process more tedious in the recent era.

The two ways of simplifying the description of large dimensional datasets are the following:

1. Remove redundant dimensions or variables, and
2. retain the most important dimensions/variables.

Principal component analysis (PCA) is the best, widely used technique to perform these two tasks. The purpose of this article is to provide a complete and simplified explanation of principal component analysis, especially to demonstrate how you can perform this analysis using R.

WHAT IS PCA?

In simple words, PCA is a method of extracting important variables (in the form of components) from a large set of variables available in a data set. PCA is a type of unsupervised linear transformation where we take a dataset with too many variables and untangle the original variables into a smaller set of variables, which we called "principal components." It is especially useful when dealing with three or higher dimensional data. It enables the analysts to explain the variability of that dataset using fewer variables.

WHY PERFORM PCA?

The goals of PCA are to:

1. Gain an overall structure of the large dimension data,
2. determine key numerical variables based on their contribution to maximum variances in the dataset,
3. compress the size of the data set by keeping only the key variables and removing redundant variables, and
4. find out the correlation among key variables and construct new components for further analysis.

Note that, the PCA method is particularly useful when the variables within the data set are highly correlated and redundant.

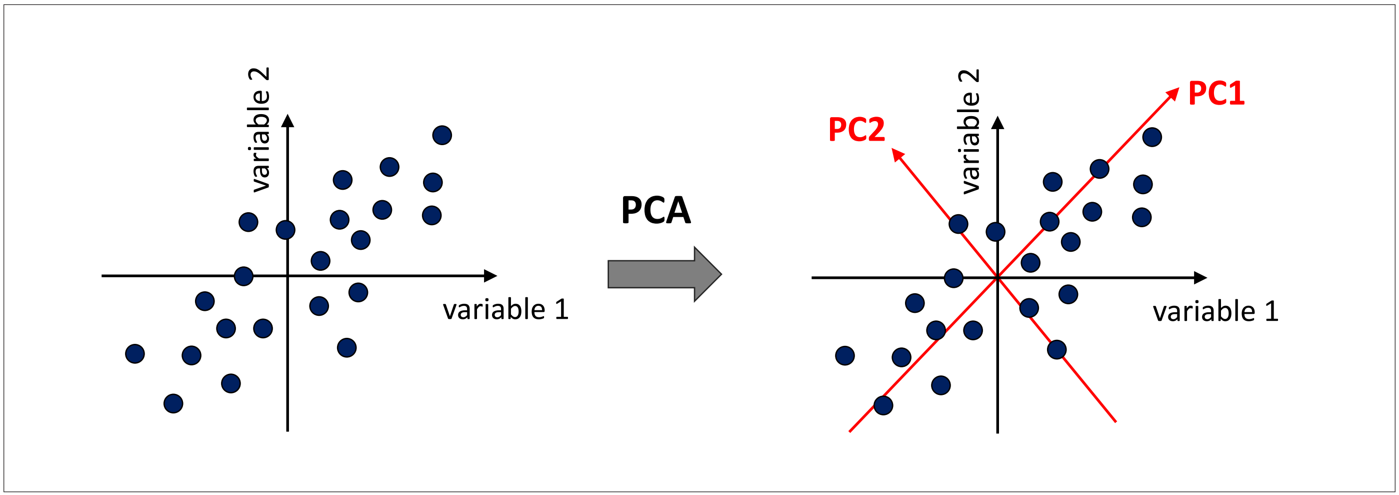
HOW DO WE PERFORM PCA?

Before I start explaining the PCA steps, I will give you a quick rundown of the mathematical formula and description of the principal components.

What are Principal Components?

Principal components are the set of new variables that correspond to a linear combination of the original key variables. The number of principal components is less than or equal to the number of original variables.

Figure 1
Principal Components



Source: ourcodingclub.github.io

In Figure 1, the PC1 axis is the first principal direction along which the samples show the largest variation. The PC2 axis is the second most important direction, and it is orthogonal to the PC1 axis.

The first principal component of a data set $X_1,X_2,...,X_p$ is the linear combination of the features

$$Z_1=\phi_{1,1}X_1+\phi_{2,1}X_2+...+\phi_{p,1}X_p$$

$\Phi_{p,1}$ is the loading vector comprising of all the loadings ($\phi_1 \ldots \phi_p$) of the principal components.

The second principal component is the linear combination of $X_1,\ldots, X_p$ that has maximal variance out of all linear combinations that are uncorrelated with Z_1. The second principal component scores $z_{1,2},z_{2,2},\ldots,z_{n,2}$ take the form

$$Z_2=\phi_{1,2}X_1+\phi_{2,2}X_2+...+\phi_{p,2}X_p$$

It is necessary to understand the meaning of covariance and eigenvector before we further get into principal components analysis.

Covariance

Covariance is a measure to find out how much the dimensions may vary from the mean with respect to each other. For example, the covariance between two random variables X and Y can be calculated using the following formula (for population):

$$\text{Cov}(x,y) = \text{SUM}\ [(xi - xm) * (yi - ym)] / (n - 1)$$

- xi = a given x value in the data set
- xm = the mean, or average, of the x values
- yi = the y value in the data set that corresponds with xi
- ym = the mean, or average, of the y values
- n = the number of data points

Both covariance and correlation indicate whether variables are positively or inversely related. Correlation also tells you the degree to which the variables tend to move together.

Eigenvectors

Eigenvectors are a special set of vectors that satisfies the linear system equations:

$Av = \lambda v$

where A is an (n x n)square matrix, v is the eigenvector, and λ is the eigenvalue. Eigenvalues measure the amount of variances retained by the principal components. For instance, eigenvalues tend to be large for the first component and smaller for the subsequent principal components. The number of eigenvalues and eigenvectors of a given dataset is equal to the number of dimensions that dataset has. Depending upon the variances explained by the eigenvalues, we can determine the most important principal components that can be used for further analysis.

GENERAL METHODS FOR PRINCIPAL COMPONENT ANALYSIS USING R

Singular value decomposition (SVD) is considered to be a general method for PCA. This method examines the correlations between individuals,

The functions prcomp ()[“stats” package] and PCA()[“FactoMineR” package] use the SVD.

PCA () function comes from FactoMineR. So, install this package along with another package called Factoextra which will be used to visualize the results of PCA.

In this article, I will demonstrate a sample of SVD method using PCA() function and visualize the variance results.

Dataset Description

I will explore the principal components of a dataset which is extracted from KEEL-dataset repository.

This dataset was proposed in McDonald, G.C. and Schwing, R.C. (1973) “Instabilities of Regression Estimates Relating Air Pollution to Mortality,” *Technometrics*, vol.15, 463-482. It contains 16 attributes describing 60 different pollution scenarios. The attributes are the following:

1. PRECReal: Average annual precipitation in inches
2. JANTReal: Average January temperature in degrees F
3. JULTReal: Same for July
4. OVR65Real: of 1960 SMSA population aged 65 or older
5. POPNReal: Average household size
6. EDUCReal: Median school years completed by those over 22
7. HOUSReal: of housing units which are sound and with all facilities
8. DENSReal: Population per sq. mile in urbanized areas, 1960
9. NONWReal: non-white population in urbanized areas, 1960
10. WWDRKReal: employed in white collar occupations
11. POORReal: of families with income less than $3000
12. HCReal: Relative hydrocarbon pollution potential
13. NOXReal: Same for nitric oxides
14. SO@Real: Same for sulphur dioxide
15. HUMIDReal: Annual average % relative humidity at 1pm
16. MORTReal: Total age-adjusted mortality rate per 100,000

Figure 2
Computer Code for Pollution Scenarios

```
pollution <- read.delim(“pollution.dat”,
header = FALSE,skip = 19, sep = “,”)

colnames(pollution) <- c(“PRECReal”,
”JANTReal”,”JULTReal”,”OVR65Real”,
”POPNReal”,”EDUCReal”,”HOUSReal”,
”DENSReal”,”NONWReal”,”WWDRKReal”,
”POORReal”,”HCReal”,”NOXReal”,”SO@Real”,
”HUMIDReal”,”MORTReal”)

library(dplyr)

pollution <- mutate(pollution,MORTReal_Type
= case_when

  (pollution$MORTReal < 900.0 ~
  “Low Mortality”,
  pollution$MORTReal > 900.0 & MORTReal <
  1000.0 ~ “Medium Mortality”,
  pollution$MORTReal > 1000.0 ~
  “High Mortality”))
```

The code in Figure 2 loads the dataset to an R data frame and names all 16 variables. In order to define a different range of mortality rate, one extra column named “MORTReal_TYPE” has been created in the R data frame. This extra column will be useful to create data visualization based on mortality rates.

Compute Principal Components Using PCA ()

PCA () [*FactoMineR* package] function is very useful to identify the principal components and the contributing variables associated with those PCs. A simplified format is:

```
library(“FactoMineR”)

pollution.PCA <- PCA(pollution[c(-17)],
scale.unit = TRUE, graph = FALSE)
```

- pollution: a data frame. Rows are individuals and columns are numeric variables
- scale.unit: a logical value. If TRUE, the data are scaled to unit variance before the analysis. This standardization to the same scale avoids some variables to become dominant just because of their large measurement units. It makes the variable comparable.
- graph: a logical value. If TRUE a graph is displayed.

The output of the function PCA () is a list that includes the following components:

```
>pollution.pca
““Results for the Principal Component Analysis (PCA)**
The analysis was performed on 60 individuals, described by 16 variables
*The results are available in the following objects:

Name                     description
1  “$eig”                “eigenvalues”
2  “$var”                “results for the variables”
3  “$var$coord”          “coord. for the variables”
4  “$var$cor”            “correlations variables - dimensions”
5  “$var$cos2”           “cos2 for the variables”
6  “$var$contrib”        “contributions of the variables”
7  “$ind”    `           “results for the individuals”
8  “$ind$coord”          “coord.for the individuals”
9  “$ind$cos2”           “cos2 for the individuals”
10 “$ind$contrib”        “contributions of the individuals”
11 “$call”               “summary statistics”
12 “$call$centre”        “mean of the variables”
13 “$call$ecart.type”    “standard error of the variables”
14 “$call$row.w”         “weights for the individuals”
15 “$call$col.w”         “weights for the variables”
```

For better interpretation of PCA, we need to visualize the components using R functions provided in factoextra R package:

get_eigenvalue(): Extract the eigenvalues/variances of principal components
fviz_eig(): Visualize the eigenvalues
fviz_pca_ind(), fviz_pca_var(): Visualize the results individuals and variables, respectively.

EIGENVALUES

As described in the previous section, eigenvalues are used to measure the variances retained by the principal components. First principal component keeps the largest value of eigenvalues and the subsequent PCs have smaller values. To determine the eigenvalues and proportion of variances held by different PCs of a given data set we need to rely on the R function get_eigenvalue() that can be extracted from the factoextra package.

```
library(“factoeextra”)

eig.val <- get_eigenvalue(pollution.PCA)
eig.val
```

```
  “eig.val”
  eigenvalue variance.percent cumulative.variance.percent
  Dim.1  4.878595616      30.49122260                   30.49122
  Dim.2  2.766574422      17.29109013                   47.78231
  Dim.3  2.292475683      14.32797302                   62.11029
  Dim.4  1.351660343       8.44787715                   70.55816
  Dim.5  1.223507408       7.64692130                   78.20508
  Dim.6  1.086738477       6.79211548                   84.99720
  Dim.7  0.661476260       4.13422662                   89.13143
  Dim.8  0.479425447       2.99640904                   92.12784
  Dim.9  0.407500850       2.54688031                   94.67472
  Dim.10 0.244819892       1.53012432                   96.20484
  Dim.11 0.194097702       1.21311064                   97.41795
  Dim.12 0.156401959       0.97751224                   98.39546
  Dim.13 0.116810134       0.73006334                   99.12553
  Dim.14 0.089284390       0.55802744                   99.68355
  Dim.15 0.045962000       0.28726250                   99.97082
  Dim.16 0.004669417       0.02918386                  100.00000
```

The sum of all the eigenvalues gives a total variance of 16.

The proportion of all the eigenvalues is demonstrated by the second column “variance.present.” For example, if you divide 4.878 by 16 equals to 0.304875, i.e., almost 30.49 percent variance explained by the first component/dimension. Based on the output of eig.val object, we can derive the fact that the first six eigenvalues keep almost 82 percent of total variances existed in the dataset.

As an alternative approach, we can also examine the pattern of variances using a scree plot which showcases the order of eigenvalues from largest to smallest. In order to produce the scree plot (see Figure 3), we will use the function fviz_eig() available in factoextra() package:

```
fviz_eig(pollution.pca, addlabels = TRUE,
hjust = -0.3, ylim = c(0,35))
```

From the scree plot above, we might consider using the first six components for the analysis because 82 percent of the whole dataset information is retained by these principal components.

VARIABLES CONTRIBUTION GRAPH

The next step is to determine the contribution and the correlation of the variables that have been considered as principal components of the dataset. In order to extract the relationship of the variables from a PCA object we need to use the function get_pca_var () which provides a list of matrices containing all the results for the active variables (coordinates, correlation between variables, squared cosine and contributions).

```
var_pollution <- get_pca_var(pollution.PCA)
var_pollution
```

```
> var_pollution.pca
Principal Component Analysis Results for
varibles

Name          description
1 “$coord”    “Coordinates for the variables”
2 “$cor”      “Correlations between variables
               and dimensions”
3 “$cos2”     “Cos2 for the variables”
4 “$contrib”  “contributions of the variables”
```

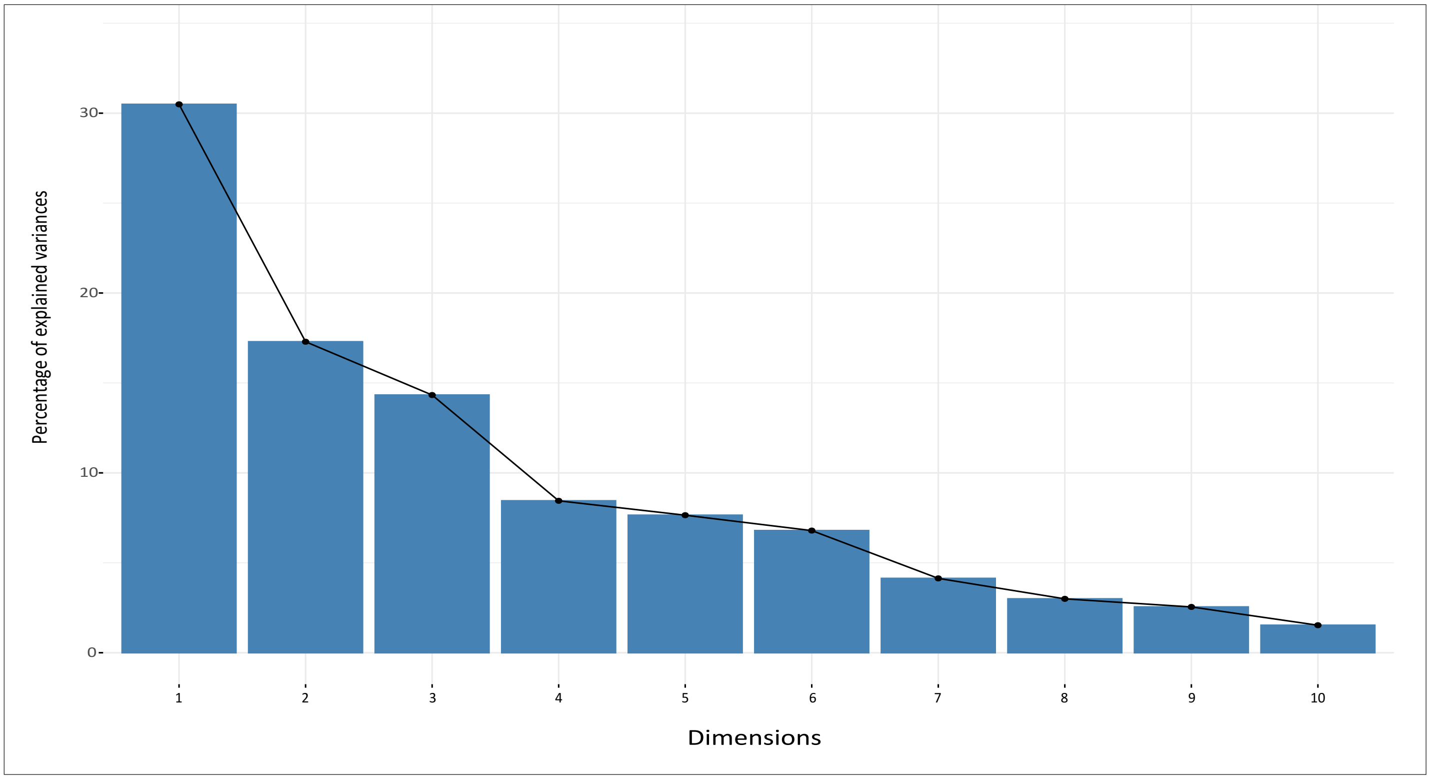
CORRELATION CIRCLE PLOT

We can apply different methods to visualize the SVD variances in a correlation plot in order to demonstrate the relationship between variables. The correlation between a variable and a principal component (PC) is used as the coordinates of the variable on the PC.

```
# Coordinates of Variables

head(var_pollution$contrib)
```

Figure 3
Scree Plot



```
> head(var_pollution$conrib)
                Dim.1      Dim.2        Dim.3      Dim.4      Dim.5
PRECReal  11.3363777  1.2901207 2.320962e-04 10.5955205  1.6692242
JANTReal   0.3312333 21.4926762 3.234525e+00 10.8905057  0.6850515
JULTReal  10.2768749  2.7936599 2.838199e+00  0.1211819 15.3922039
OVR65Real  2.4116845 11.8740295 3.795171e+00 27.4577926  0.1384696
POPNReal   8.1452038  0.5965791 3.132326e-03 30.1085931  5.6756771
EDUCReal   8.6313043  2.4816964 1.212178e+01  2.0276052
```

To plot all the variables we can use fviz_pca_var() :

```
fviz_pca_var(pollution.PCA,col.var = "black")
```

Figure 4
Relationship Between Variables

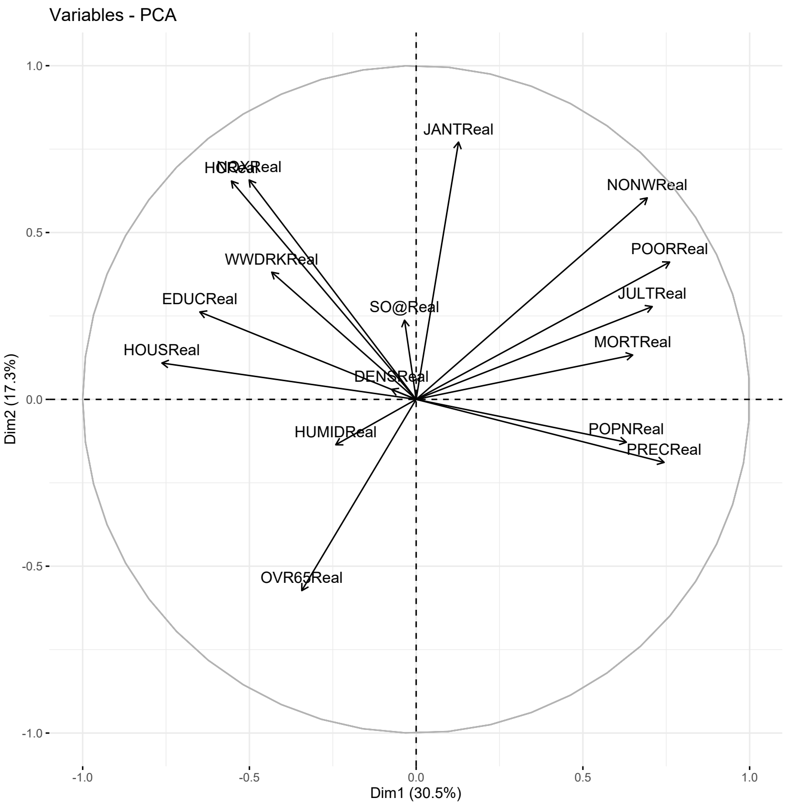


Figure 4 shows the relationship between variables in three different ways:

- Positively correlated variables are grouped together.
- Negatively correlated variables are located on opposite sides of the plot origin
- The distance between variables and the origin measures the quality of the variables on the factor map. Variables that are away from the origin are well represented on the factor map.

QUALITY OF REPRESENTATION

This shows the quality of representation of the variables on the factor map called cos2, which is multiplication of squared cosine and squared coordinates. The previously created object var_pollution holds cos2 value:

```
head(var_pollution$cos2)
```

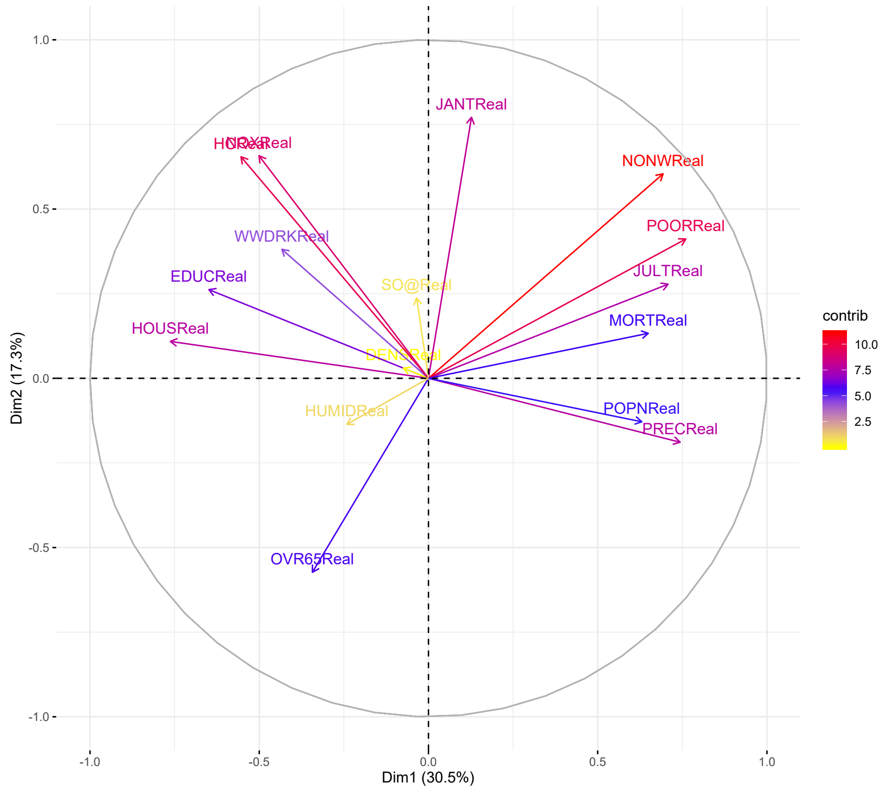
```
head(var_pollution$cos2)
                Dim.1      Dim.2        Dim.3       Dim.4       Dim.5
PRECReal   0.55305602 0.03569215 5.320748e-06 0.143215449 0.020423082
JANTReal   0.01615953 0.59461088 7.415070e-02 0.147202647 0.008381656
JULTReal   0.50136717 0.07728868 6.506502e-02 0.001637968 0.188324755
OVR65Real  0.11765633 0.32850386 8.700336e-02 0.371136093 0.001694186
POPNReal   0.39737155 0.01650481 7.180782e-05 0.406965913 0.069442330
EDUCReal   0.42108643 0.06865798 2.778888e-01 0.027406335 0.025652309
```

A high cos2 indicates a good representation of the variable on a particular dimension or principal component. Whereas, a low cos2 indicates that the variable is not perfectly represented by PCs.

Cos2 values can be well presented using various aesthetic colors in a correlation plot. For instance, we can use three different colors to present the low, mid and high cos2 values of variables that contribute to the principal components.

```
fviz_pca_var(pollution.PCA,col.var = "cos2",
             gradient.cols = c("green","blue","red"),
             repel = TRUE # Avoid text overlapping
```

Figure 5
Variables—PCA



Variables that are closed to circumference (like NONWReal, POORReal and HCReal) manifest the maximum representation of the principal components. However, variables like HUMIDReal, DENSReal and SO@Real show week representation of the principal components.

CONTRIBUTION OF VARIABLES TO PCS

After observing the quality of representation, the next step is to explore the contribution of variables to the main PCs. Variable contributions in a given principal component are demonstrated in percentage.

Key points to remember:

- Variables with high contribution rate should be retained as those are the most important components that can explain the variability in the dataset.
- Variables with low contribution rate can be excluded from the dataset in order to reduce the complexity of the data analysis.

The function fviz_contrib() [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code (see code 1 and Figures 6 and 7) below shows the top 10 variables contributing to the principal components:

Code 1

```
#Contributions of variables to PC1
fviz_contrib(pollution.PCA, choice = “var”,
axes = 1, top = 10)
#Contribution of variables to PC2
fviz_contrib(pollution.PCA, choice = “var”,
axes = 2, top = 10)
```

The most important (or, contributing) variables can be highlighted on the correlation plot as in code 2 and Figure 8.

Code 2

```
fviz_pca_var (pollution.pca, col.var =
“contrib”,
      Gradient.cols = c(“yellow”, “blue”,
      “red”)
```

Figures 6 and 7

Top 10 Variables Contributing to Principal Components

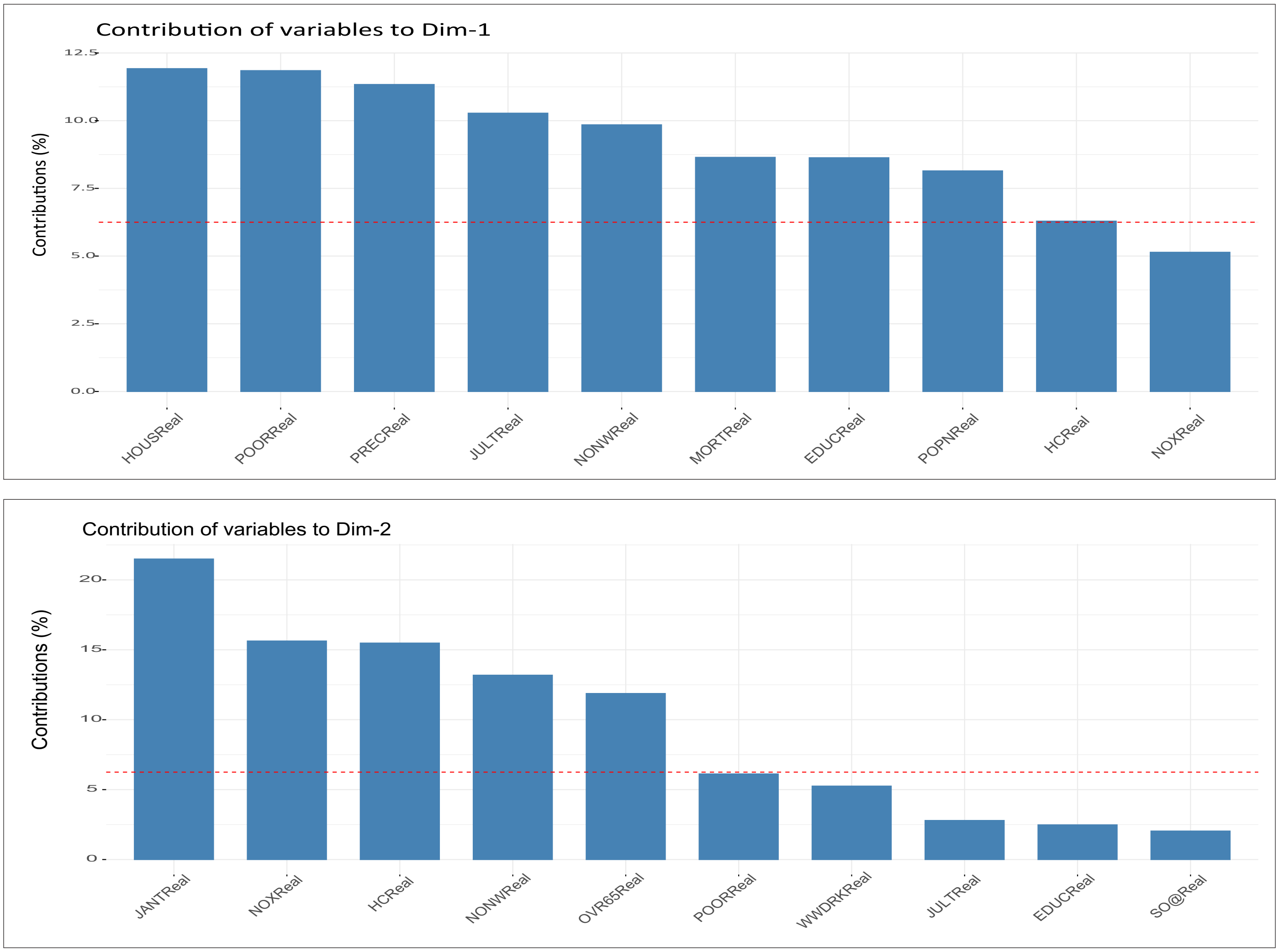
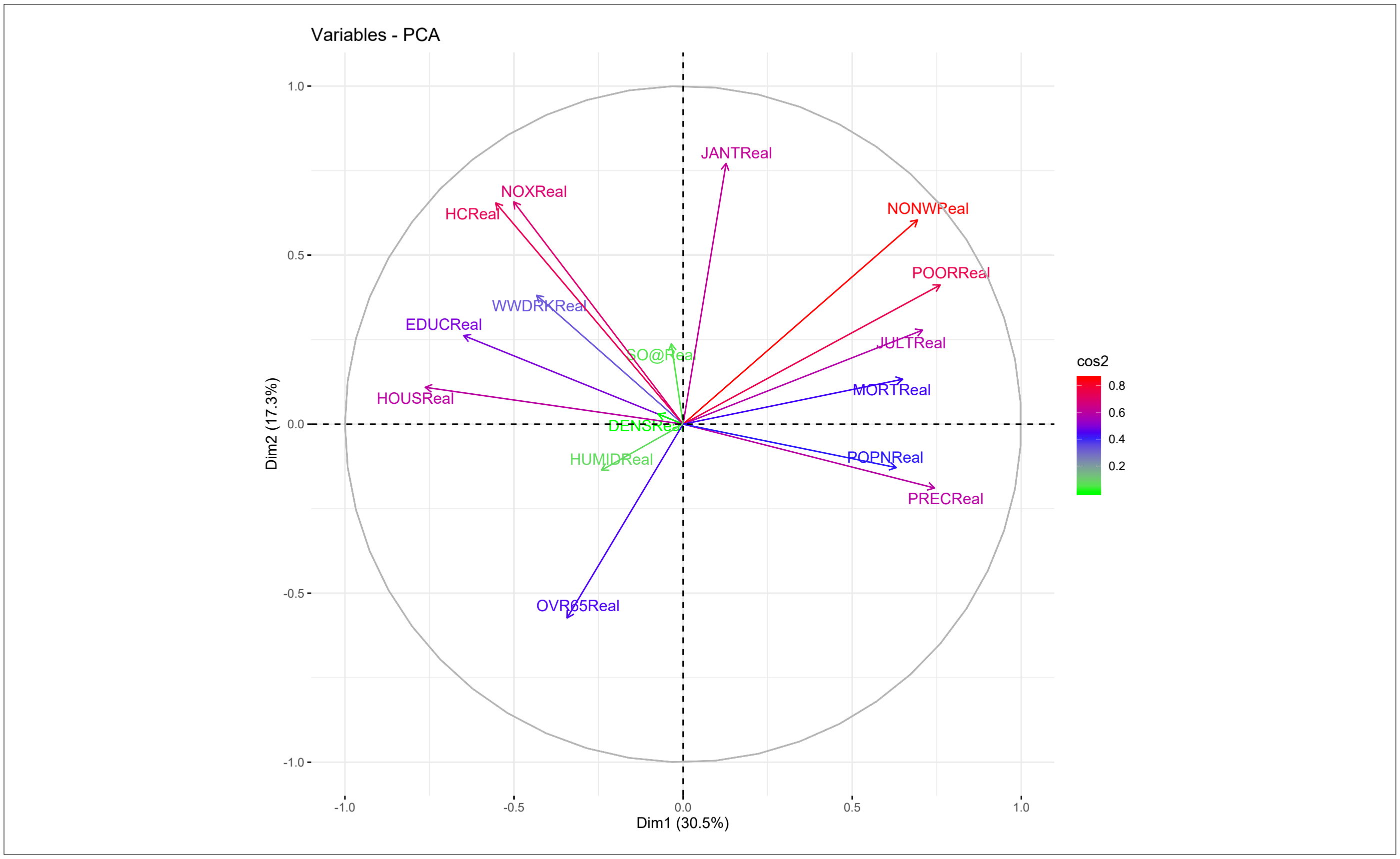


Figure 8
Graphical Display of the Eigen Vector and Their Relative Contribution



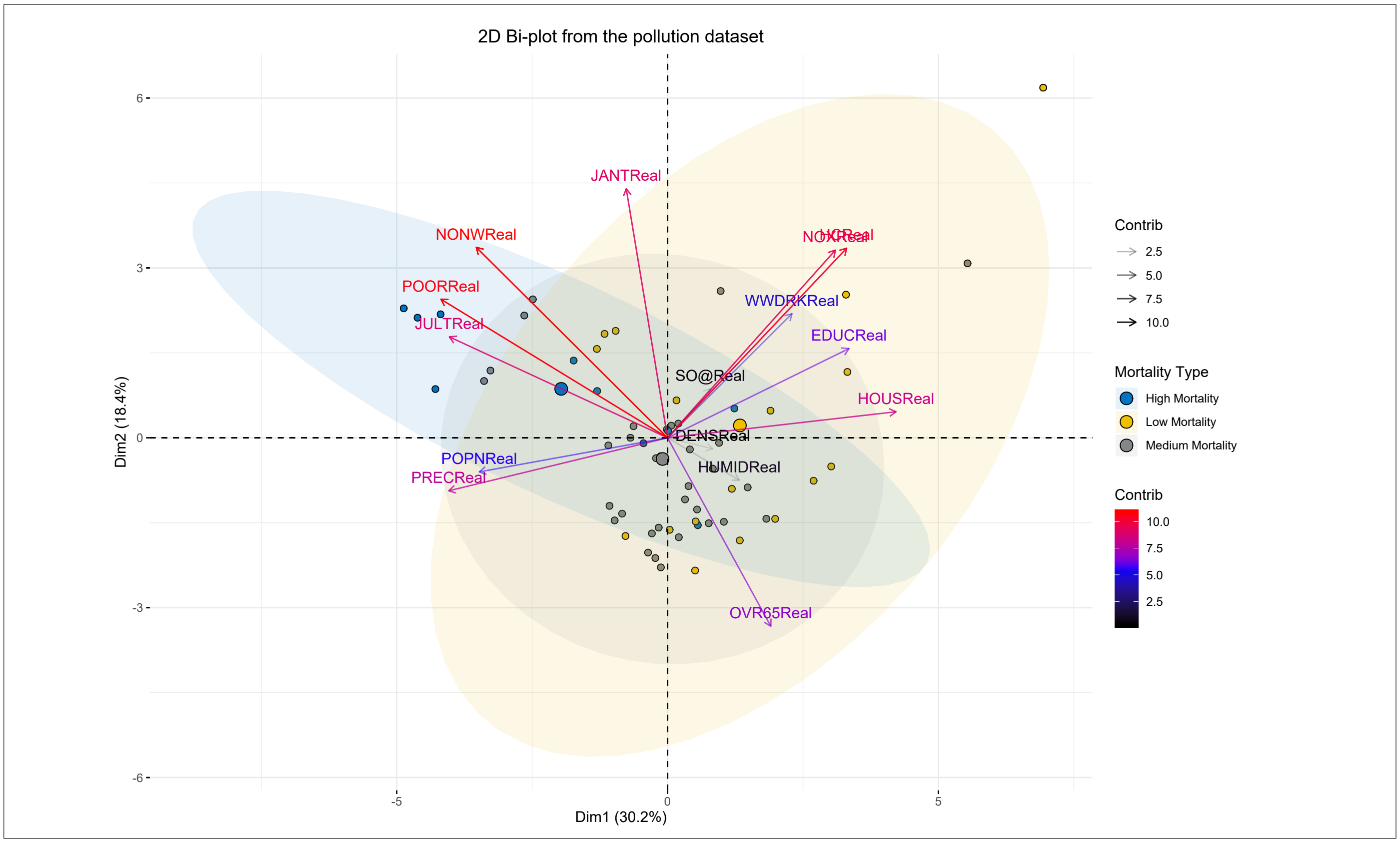
BIPLOT

To make a simple biplot of individuals and variables, type this:

Code 3

```
fviz_biplot (pollution.pca,
      col.ind = pollution$MORTReal_TYPE, palette = "jco",
      addEllipses = TRUE, label = "var"
      col.var = "black", repel = TRUE,
      legend.title = "Mortality_Range")
```

Figure 9
Mortality Rate Value and Corresponding Key Variables Grouped



In Figure 9, column "MORTReal_TYPE" has been used to group the mortality rate value and corresponding key variables.

SUMMARY

PCA analysis is unsupervised, so this analysis is not making predictions about pollution rate, rather simply showing the variability of dataset using fewer variables. Key observations derived from the sample PCA described in this article are:

1. Six dimensions demonstrate almost 82 percent variances of the whole data set.

2. The following variables are the key contributors to the variability of the data set:
 NONWReal, POORReal, HCReal, NOXReal, HOUSReal and MORTReal.

3 Correlation plots and Bi-plot help to identify and interpret correlation among the key variables.

For Python Users

To implement PCA in python, simply import PCA from sklearn library. The code interpretation remains the same as explained for R users above.

INDUSTRY APPLICATION USE

PCA is a very common mathematical technique for dimension reduction that is applicable in every industry related to STEM (science, technology, engineering and mathematics). Most importantly, this technique has become widely popular in areas of quantitative finance. For instance, fund portfolio managers often use PCA to point out the main mathematical factors that drive the movement of all stocks. Eventually, that helps in forecasting portfolio returns, analyzing the risk of large institutional portfolios and developing asset allocation algorithms for equity portfolios.

PCA has been considered as a multivariate statistical tool which is useful to perform the computer network analysis in order to identify hacking or intrusion activities. Network traffic data is typically high-dimensional making it difficult to analyze and visualize. Dimension reduction technique and Bi-plots are helpful to understand the network activity and provide a summary of possible intrusions statistics. Based on a study conducted by UC Davis, PCA is applied to selected network attacks from the DARPA 1998 intrusion detection datasets namely: Denial-of-Service and Network Probe attacks.

Multidimensional reduction capability was used to build a wide range of PCA applications in the field of medical image processing such as feature extraction, image fusion, image compression, image segmentation, image registration and de-noising of images.

Using the multivariate analysis feature of PCS efficient properties it can identify patterns in data of high dimensions and can serve applications for pattern recognition problems. For example, one type for PCA is the Kernel principal component analysis (KPCA) which can be used for analyzing ultrasound medical images of liver cancer (Hu and Gui, 2008). Compared with the experiments of wavelets, the experiment of KPCA showed that KPCA is more effective than wavelets especially in the application of ultrasound medical images.

CONCLUSION

This tutorial gets you started with using PCA. Many statistical techniques, including regression, classification, and clustering can be easily adapted to using principal components.

PCA helps to produce better visualization of high dimensional data. The sample analysis only helps to identify the key variables that can be used as predictors for building the regression model for estimating the relation of air pollution to mortality. My article does not outline the model building technique, but the six principal components can be used to construct some kind of model for prediction purposes.

Further Reading

PCA using prcomp() and princomp() (tutorial). *http://www.sthda.com/english/wiki/pca-using-prcomp-and-princomp*

PCA using ade4 and factoextra (tutorial). *http://www.sthda.com/english/wiki/pca-using-ade4-and-factoextra* ■

Soumava Dey is an actuarial systems analyst at AIG. He can be contacted at soumava.dey@aig.com.

REFERENCE

Husson, Francois, Sebastien Le, and Jérôme Pagès. 2017. Exploratory Multivariate Analysis by Example Using R. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. *http://factominer.free.fr/bookV2/index.html.*

Abdi, Hervé, and Lynne J. Williams. 2010. “Principal Component Analysis.” John Wiley and Sons, Inc. WIREs Comp Stat 2: 433–59. *http://staff.ustc.edu.cn/~zwp/teach/MVA/abdi-awPCA2010.pdf.*

KEEL-dataset citation paper: J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. “KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework.” Journal of *Multiple-Valued Logic and Soft Computing* 17:2-3 (2011) 255-287.

Khaled Labib and V. Rao Vemuri.“An Application of Principal Component Analysis to the Detection and Visualization of Computer Network Attacks.” *https://web.cs.ucdavis.edu/~vemuri/papers/pcaVisualization.pdf*

Libin Yang. 2015. “An Application of Principal Component Analysis to Stock Portfolio Management.” *https://ir.canterbury.ac.nz/bitstream/handle/10092/10293/thesis.pdf*

https://www.researchgate.net/publication/272576742_Principal_Component_Analysis_in_Medical_Image_Processing_A_Study

https://rdrr.io/cran/factoextra/man/fviz_pca.html