High Diminsional Statistics-Sheet 5-Exercise 4

Hamed Vaheb

16/05/2022

Consider the orange juice dataset X, called "jusdorange.csv". The dataset contains 7 characteristics of the orange juice, on which we want to perform a principal component analysis.

Prepare

First of all we import the dataset:

```
path <- "/home/hamed/Documents/datasets/jusdorange.csv"
data <- read.table(path, sep=";", header=T)
#drops <- c("Conditionnement","Origine", "Produit")
#df <- data[ , !(names(data) %in% drops)]
X <- data[,sapply(data, is.numeric)]
X
```

```
##   Intensité.odeur Typicité.odeur Caractère.pulpeux Intensité.goût
## 1            2.82           2.53              1.66           3.46
## 2            2.76           2.82              1.91           3.23
## 3            2.83           2.88              4.00           3.45
## 4            2.76           2.59              1.66           3.37
## 5            3.20           3.02              3.69           3.12
## 6            3.07           2.73              3.34           3.54
##   Caractère.acide Caractère.amer Caractère.sucré Glucose Fructose Saccharose
## 1            3.15           2.97            2.60   25.32    27.36      36.45
## 2            2.55           2.08            3.32   17.33    20.00      44.15
## 3            2.42           1.76            3.38   23.65    25.65      52.12
## 4            3.05           2.56            2.80   32.42    34.54      22.92
## 5            2.33           1.97            3.34   22.70    25.32      45.80
## 6            3.31           2.63            2.90   27.16    29.48      38.94
##   Pouvoir.sucrant   pH Titre Acide.citrique Vitamine.C
## 1           89.95 3.59 13.98           0.84      43.44
## 2           82.55 3.89 11.14           0.67      32.70
## 3          102.22 3.85 11.51           0.69      37.00
## 4           90.71 3.60 15.75           0.95      36.60
## 5           94.87 3.82 11.80           0.71      39.50
## 6           96.51 3.68 12.21           0.74      27.00
```

Question 1

Build the centered vector Y starting from X. Compute its covariance matrix Σ and the eigenvectors and eigenvalues of Σ. What do they represent in the principal component analysis?

We create Y by centering X, i.e., subtract mean of each variable (column) from its observations (rows). We then construct covariance matrix of Y and computes its eigenvalues and eigenvectors.

```r
# center with 'sapply()'
Y <- sapply(X, function(x) as.numeric(x) - mean(x))
Sigma <- cov(Y)
#Sigma
#Sigma
#summary(Sigma)

e <- eigen(Sigma)

# eigenvalues
eigvals <- e$values

# eigenvectors
eigvecs <- e$vectors

eigvals
```

```
##  [1]  1.400532e+02  5.813757e+01  3.243695e+01  3.091913e-01  1.345573e-01
##  [6]  1.046674e-14  7.409821e-15  1.398509e-15  8.719297e-17 -2.214281e-18
## [11] -1.453968e-17 -1.895423e-17 -5.913880e-17 -4.022990e-16 -7.692731e-16
```

```r
#eigvecs
#sort(eigvals, decreasing=TRUE)
```

The eigenvectors and eigenvalues of a covariance matrix lie at core of PCA. Since PCA maps our data to a lower dimension space, the contribution of each dimension in explaining the variability in data heavily depends on eigenvalues and eigenvectors. More specifically, the eigenvectors (principal components) determine the directions of the new feature space, and the eigenvalues determine their magnitude. In other words, eigenvectors manifest the direction of spread of our data. In mathematical terms, suppose that β_1 is the first principal component, hence ($||\beta_1||_2 = 1$) and β maximizes $var({\beta_1}^T x)$. Then, we can relate β_1 to the largest eigenvalue, say λ_1, of the covariance matrix Σ as the following:

$$\max_{\beta \in \mathbb{R}^k || \beta ||_2 = 1} {\beta_1}^T \Sigma \beta_1 = \lambda_1$$

It turns out that β_1 is the eigenvector of Σ associated with λ_1.

Question 2

Compute the variance represented by the first axis. How many principal components do we need to explain at least the 90% of the total variance of the dataset?

In order to determine how many PCA componentes are required, we need to measure the propoertion of variance for each component. In other words, each component explains some part of variability in data, such tat the first component explaining the most (as it is corresponded to the biggest eigenvalue of Σ) and the last making the least contribution.
As we need 90% of the total variance, we will derive the cumulative sum proportion of variance for all components and determine the component at which this sum reaches 90 percent. All components prior to that component with that component are required to explain the 90% proportion.

The fact that first components contribute the most, results in having least number of components required for the explainability we require (90% in this case).

Using the PCA model which is defined by applying `prcomp` function on our dataset, we will compute proportion of variance using the two methods, one is just by observing values provided by the summary of the model, and another to compute them, as they are normalized standard deviation.

```
## method1: PCA model
#eig.val <- get_eigenvalue(pca)
pca <- prcomp(Y, center = FALSE, scale = TRUE)
summary(pca)
```

Determine Nnumer of Components

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5       PC6
## Standard deviation     2.9828 1.6074 1.2685 1.00585 0.9479 1.717e-15
## Proportion of Variance 0.5931 0.1722 0.1073 0.06745 0.0599 0.000e+00
## Cumulative Proportion  0.5931 0.7654 0.8727 0.94010 1.0000 1.000e+00
```

```
## method2: computation
PoV <- pca$sdev^2/sum(pca$sdev^2)
label_percent()(PoV)
```

```
## [1] "59.31%" "17.23%" "10.73%" "6.74%"  "5.99%"  "0.00%"
```

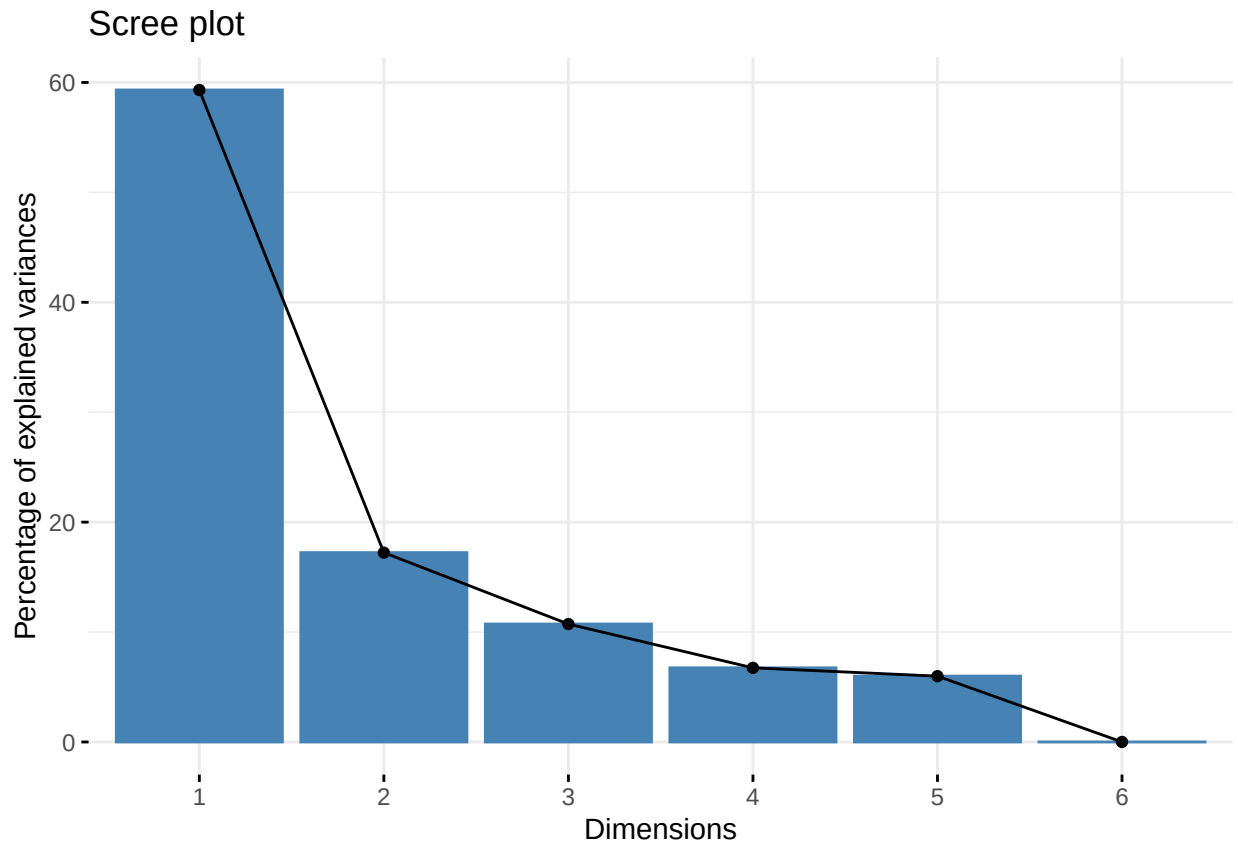
In the summary of our PCA model, we can see that the cumulative variance is equal to 94, therefore reaches 90% at PC4 (4th principal component). However, at PC3 it is also very close to 90%. If we choose to be stringent, then we should choose 4th component.

Plot proportion of variance For the purpose of illustration, I plotted both proportion of variance and cummulative proportion of variance, which validates the statement I made about number of components required.

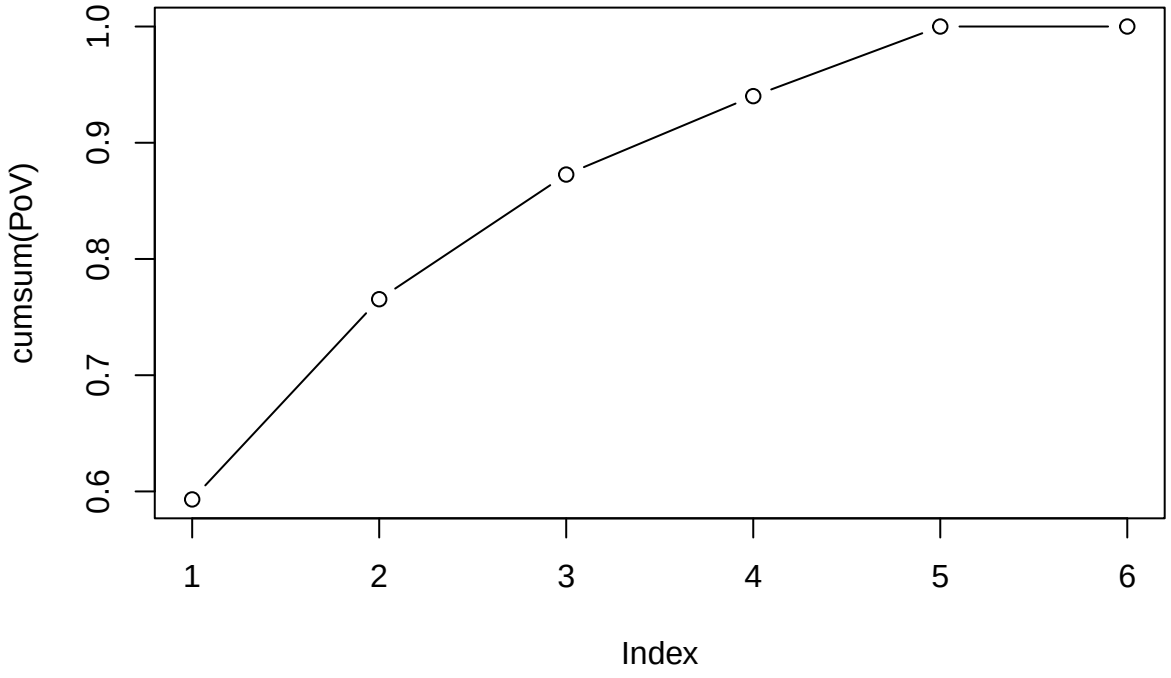
```
# compute PCA
get_eig(pca)
```

```
##         eigenvalue variance.percent cumulative.variance.percent
## Dim.1 8.897003e+00     5.931335e+01                     59.31335
## Dim.2 2.583801e+00     1.722534e+01                     76.53869
## Dim.3 1.609019e+00     1.072679e+01                     87.26548
## Dim.4 1.011732e+00     6.744883e+00                     94.01036
## Dim.5 8.984454e-01     5.989636e+00                    100.00000
## Dim.6 2.948015e-30     1.965343e-29                    100.00000
```

```
fviz_eig(pca)
```




```
plot(cumsum(PoV), type="b")
```



Question 3

Draw the plot of the first two axes with samples of various juice characteristics displayed by different color and symbol.

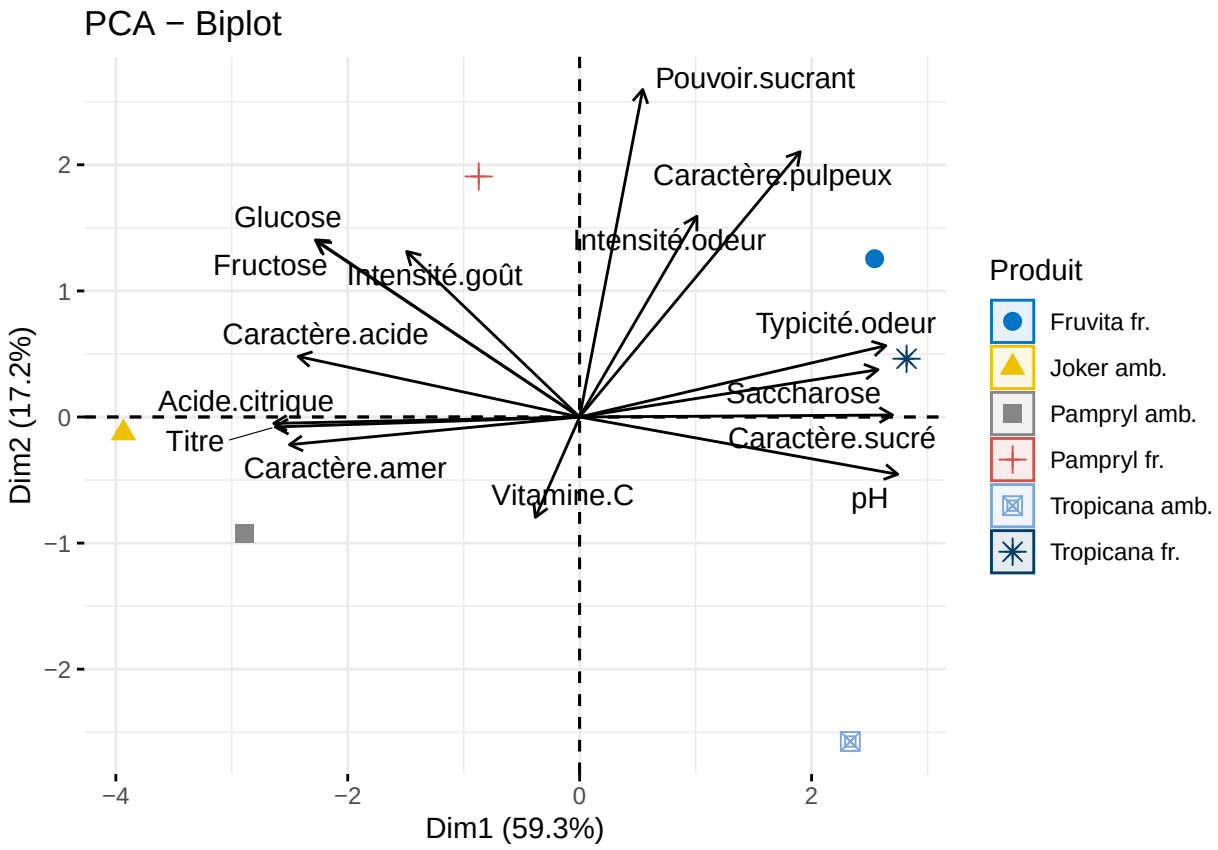
We use two representations to plot the components.

Representation 1: This plot includes the points along with the components. Positive correlated variables point to the same side of the plot. Negative correlated variables point to opposite sides of the graph. Based on the "produit" column of the original dataframe, we have categorized the data and each point is colored based on the category it belongs to. Moreover, each category is represented using a different shape.

```
# store "produit" categores in "labels"
labels <- data[,1]
rownames(X) <- labels
#ggbiplot(pca, labels = labels, obs.scale = 2, var.scale = 1)

fviz_pca_biplot(pca,
        col.ind = labels, palette = "jco",
        gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
        addEllipses = TRUE, label = "var",
        col.var = "black", repel = TRUE,
        legend.title = "Produit")
```

```
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```



Representation 2: This plot includes the points along with the components but with additional aesthetics. Each component is colored based on its contribution.

```
fviz_pca_var(pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```