FCD Assignment 3

2024-11-24

Authors: - Afonso Vaz 64505 - Lloyd D'silva 64858 - Matei-Alexandru Lupașcu 64471

Assignment 3 -

Option A - CSV File: airpollution.csv

Introduction In this project, we analyze a dataset on air pollution across 41 cities in the USA, containing the following variables:

- **so2**: Sulphur dioxide content of air in micrograms per cubic meter.
- **temp**: Average annual temperature (°F).
- **manuf**: Number of manufacturing enterprises employing 20 or more workers.
- **pop**: Population size (1970 census) in thousands.
- **wind**: Average wind speed in miles per hour.
- **precip**: Average annual precipitation in inches.
- **days**: Average number of days with precipitation per year.

Task 1: Performing Principal Components Analysis

First, we must begin with a brief analysis of the dataset, to know what we are working with.

```
# Import necessary libraries
library(psych)
library(ggplot2)

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

# Load the CSV file and check the first 6 rows
pollution <- read.csv("airpollution.csv")
head(pollution)

##       city so2 temp manuf pop wind precip days
## 1  Phoenix  10 70.3   213 582  6.0   7.05   36
## 2 Little R  13 61.0    91 132  8.2  48.52  100
## 3 San Fran  12 56.7   453 716  8.7  20.66   67
## 4   Denver  17 51.9   454 515  9.0  12.95   86
```

```
## 5 Hartford  56 49.1   412 158  9.0  43.37  127
## 6 Wilmingt  36 54.0    80  80  9.0  40.25  114
```

```r
# Check the data type of the variables
str(pollution)
```

```
## 'data.frame':    41 obs. of  8 variables:
##  $ city  : chr  "Phoenix" "Little R" "San Fran" "Denver" ...
##  $ so2   : int  10 13 12 17 56 36 29 14 10 24 ...
##  $ temp  : num  70.3 61 56.7 51.9 49.1 54 57.3 68.4 75.5 61.5 ...
##  $ manuf : int  213 91 453 454 412 80 434 136 207 368 ...
##  $ pop   : int  582 132 716 515 158 80 757 529 335 497 ...
##  $ wind  : num  6 8.2 8.7 9 9 9 9.3 8.8 9 9.1 ...
##  $ precip: num  7.05 48.52 20.66 12.95 43.37 ...
##  $ days  : int  36 100 67 86 127 114 111 116 128 115 ...
```

To begin, we removed non-numeric variables (e.g., “city”) to retain only numerical data.

```r
# Removal of the variable "city" for the analysis of the dataset
pollution_dataset <- pollution[,2:8]
head(pollution_dataset)
```

```
##   so2 temp manuf pop wind precip days
## 1  10 70.3   213 582  6.0   7.05   36
## 2  13 61.0    91 132  8.2  48.52  100
## 3  12 56.7   453 716  8.7  20.66   67
## 4  17 51.9   454 515  9.0  12.95   86
## 5  56 49.1   412 158  9.0  43.37  127
## 6  36 54.0    80  80  9.0  40.25  114
```

With the new ‘pollution_dataset’ made, we can now make a brief preliminary analysis of the charcateristics of the dataset. We will be checking the sd and the mean, since these are crucial for making a decision on which matrix we will be using for the PCA analysis.

```r
describe(pollution_dataset)
```

```
##        vars  n   mean     sd median trimmed    mad   min    max
range  skew
## so2       1 41  30.05  23.47  26.00   26.00  17.79  8.00  110.0
102.00  1.58
## temp      2 41  55.76   7.23  54.60   55.13   6.23 43.50   75.5
32.00  0.82
## manuf     3 41 463.10 563.47 347.00  353.79 246.11 35.00 3344.0
3309.00  3.48
## pop       4 41 608.61 579.11 515.00  499.67 320.24 71.00 3369.0
3298.00  2.94
## wind      5 41   9.44   1.43   9.30    9.45   1.19  6.00   12.7
6.70  0.00
## precip    6 41  36.77  11.77  38.74   37.71   7.98  7.05   59.8
```

```
52.75 -0.69
## days      7 41 113.90  26.51 115.00  115.09  19.27 36.00  166.0
130.00 -0.55
##        kurtosis    se
## so2        2.26  3.67
## temp       0.09  1.13
## manuf     14.33 88.00
## pop       10.58 90.44
## wind       0.06  0.22
## precip     0.50  1.84
## days       0.72  4.14
```

After observing significant differences in the means and standard deviations of the variables, we chose to conduct PCA using the correlation matrix.

Determining the Number of Principal Components

To follow up with the PCA, we need to determine how many PC's we should consider for the analysis.

We employed three standard methods to decide how many components to retain:

1. Kaiser Criteria: Retain components with eigenvalues > 1.
2. Proportion of Variance Explained: Retain components contributing to a cumulative variance > 80%.
3. Scree Plot: Visual inspection of the plot indicating an "elbow" point.

We can start with Kaiser Criteria ->

```
### Kaiser Criteria ###

# Obtain Eigenvalues and Eigenvectors. Check which are > 1

## 1st) Determine the correlation matrix

cor_pollution <- cor(pollution_dataset)
cor_pollution

##                so2        temp       manuf         pop        wind
precip
## so2     1.00000000 -0.43360020  0.64476873  0.49377958  0.09469045
0.05429434
## temp   -0.43360020  1.00000000 -0.19004216 -0.06267813 -0.34973963
0.38625342
## manuf   0.64476873 -0.19004216  1.00000000  0.95526935  0.23794683 -
0.03241688
## pop     0.49377958 -0.06267813  0.95526935  1.00000000  0.21264375 -
0.02611873
## wind    0.09469045 -0.34973963  0.23794683  0.21264375  1.00000000 -
0.01299438
## precip  0.05429434  0.38625342 -0.03241688 -0.02611873 -0.01299438
```

```
1.00000000
## days    0.36956363 -0.43024212  0.13182930  0.04208319  0.16410559
0.49609671
##               days
## so2     0.36956363
## temp   -0.43024212
## manuf   0.13182930
## pop     0.04208319
## wind    0.16410559
## precip  0.49609671
## days    1.00000000
```

```
## 2nd) Obtain eigenvalues and eigenvectors

eigen_pollution <- eigen(cor_pollution)
eigen_pollution
```

```
## eigen() decomposition
## $values
## [1] 2.72811968 1.51233485 1.39497299 0.89199129 0.34677866 0.10028759
0.02551493
##
## $vectors
##               [,1]        [,2]       [,3]        [,4]       [,5]
[,6]
## [1,]  0.4896988171 -0.08457563 -0.0143502  0.40421007  0.7303942 -
0.18334573
## [2,] -0.3153706901  0.08863789 -0.6771362 -0.18522794  0.1624652 -
0.61066107
## [3,]  0.5411687028  0.22588109 -0.2671591 -0.02627237 -0.1641011
0.04273352
## [4,]  0.4875881115  0.28200380 -0.3448380 -0.11340377 -0.3491048
0.08786327
## [5,]  0.2498749284 -0.05547149  0.3112655 -0.86190131  0.2682549 -
0.15005378
## [6,]  0.0001873122 -0.62587937 -0.4920363 -0.18393719  0.1605988
0.55357384
## [7,]  0.2601790729 -0.67796741  0.1095789  0.10976070 -0.4399698 -
0.50494668
##              [,7]
## [1,]  0.149529278
## [2,] -0.023664113
## [3,] -0.745180920
## [4,]  0.649125507
## [5,]  0.015765377
## [6,] -0.010315309
## [7,]  0.008217393
```

Taking a look at the eigenvalues, there are 3 values that are > 1; in that case, we will choose the 1st 3 eigenvalues according to the Kaiser Criteria.

Now we can consider the proportion of the components ->

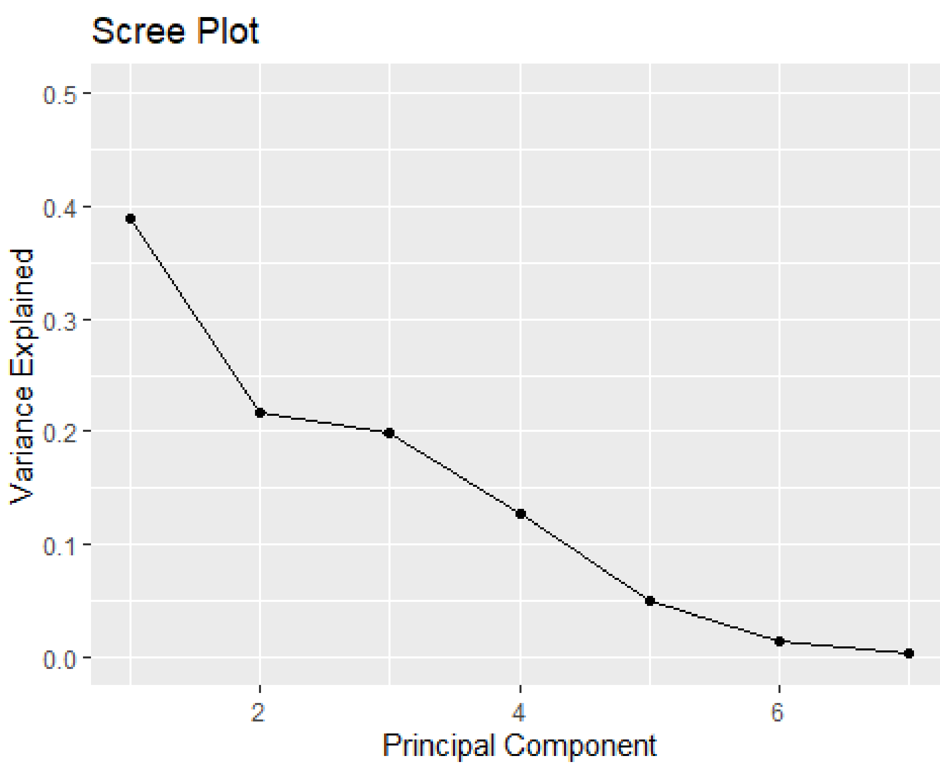
```
### Proportion of the components ###

# Perform PCA

pca_pollution <- princomp(pollution_dataset,cor = TRUE)

print(summary(pca_pollution),loadings = TRUE)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4
Comp.5
## Standard deviation     1.6517021 1.2297702 1.1810897 0.9444529
0.58887916
## Proportion of Variance 0.3897314 0.2160478 0.1992819 0.1274273
0.04953981
## Cumulative Proportion  0.3897314 0.6057792 0.8050611 0.9324884
0.98202821
##                           Comp.6      Comp.7
## Standard deviation     0.3166822 0.159733920
## Proportion of Variance 0.0143268 0.003644989
## Cumulative Proportion  0.9963550 1.000000000
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## so2     0.490                0.404  0.730  0.183  0.150
## temp   -0.315         0.677 -0.185  0.162  0.611
## manuf   0.541 -0.226  0.267        -0.164        -0.745
## pop     0.488 -0.282  0.345 -0.113 -0.349         0.649
## wind    0.250        -0.311 -0.862  0.268  0.150
## precip         0.626  0.492 -0.184  0.161 -0.554
## days    0.260  0.678 -0.110  0.110 -0.440  0.505

# Choose the components with the best SD (should be higher than 80%).
```

Taking a look at the Standard Deviation, there are 4 values that are above 80%, however, we will stick with 3, since there are 3 standard deviation values above 1 and that correlates with the amount of PCs chosen in the Kaiser's criteria.

Now we will try with the Scree-Plot ->

```
### Scree-Plot ###

# Calculate the total variance explained by each principal component
var_explained_pollution = pca_pollution$sdev^2  /
sum(pca_pollution$sdev^2)

qplot(c(1:7), var_explained_pollution) + geom_line() + xlab("Principal
Component") + ylab("Variance Explained") + ggtitle("Scree Plot") +
ylim(0,0.5)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning
was
## generated.
```



Taking a look at this Scree-Plot, we can see there's a big slope in the PC 3, so, according to the Scree-Plot, we should choose 3 PCs

Based on all the possible methodologies, we should choose 3 Principal Components.

Task 2: Selection of Principal Components

```
summary(pca_pollution)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4
Comp.5
## Standard deviation     1.6517021 1.2297702 1.1810897 0.9444529
0.58887916
## Proportion of Variance 0.3897314 0.2160478 0.1992819 0.1274273
0.04953981
## Cumulative Proportion  0.3897314 0.6057792 0.8050611 0.9324884
0.98202821
##                           Comp.6      Comp.7
## Standard deviation     0.3166822 0.159733920
## Proportion of Variance 0.0143268 0.003644989
## Cumulative Proportion  0.9963550 1.000000000
```

The retained components collectively explained a substantial proportion of the variance in the dataset. The breakdown is as follows:

- PC1: Largest contribution to variance (~40%).
- PC2: Moderate contribution (~22%).
- PC3: Moderate contribution (~20%).

Thus, the three components together account for approximately 80.5% of the total variance, making them sufficient to summarize the dataset effectively.

Task 3: Explain the importance of the variables for the explanation of each of the principal components retained.

We analyzed the variable loadings (correlations between variables and components) to interpret the roles of individual variables:

```
cor(pollution_dataset,pca_pollution$scores)

##               Comp.1      Comp.2      Comp.3      Comp.4      Comp.5
## so2     0.8088365434  0.10400859  0.01694887  0.38175738  0.43011391
## temp   -0.5208984175 -0.10900423  0.79975860 -0.17493906  0.09567234
## manuf   0.8938494595 -0.27778184  0.31553891 -0.02481301 -0.09663572
## pop     0.8053502866 -0.34679989  0.40728458 -0.10710452 -0.20558056
## wind    0.4127189331  0.06821718 -0.36763244 -0.81402520  0.15796972
## precip  0.0003093839  0.76968782  0.58113903 -0.17372001  0.09457328
## days    0.4297383099  0.83374415 -0.12942257  0.10366381 -0.25908903
##             Comp.6       Comp.7
## so2     0.05806232  0.023884898
## temp    0.19338547 -0.003779961
## manuf  -0.01353294 -0.119030670
## pop    -0.02782473  0.103687362
## wind    0.04751936  0.002518265
## precip -0.17530696 -0.001647705
## days    0.15990761  0.001312596
```

Firstly, we apply the rule to the 1st P.C.

```
# We apply a rule for the 1st P.C.
sqrt(eigen_pollution$values [1]/7)

## [1] 0.6242847
```

1. PC1: Strongly influenced by **so2**, **manuf**, and **pop**, indicating that this component captures industrial activity and population density.

Now, we apply the rule to the 2nd P.C.

```
# We apply a rule for the 2nd P.C.
sqrt(eigen_pollution$values [2]/7)

## [1] 0.4648095
```

2. PC2: Dominated by **precip** and **days**, representing weather conditions and precipitation patterns.

Now, we apply the rule to the 3rd P.C.

```
# We apply a rule for the 3rd P.C.
sqrt(eigen_pollution$values [3]/7)

## [1] 0.44641
```

3. PC3: Primarily related to **temp**, emphasizing temperature variations along with precipitation.

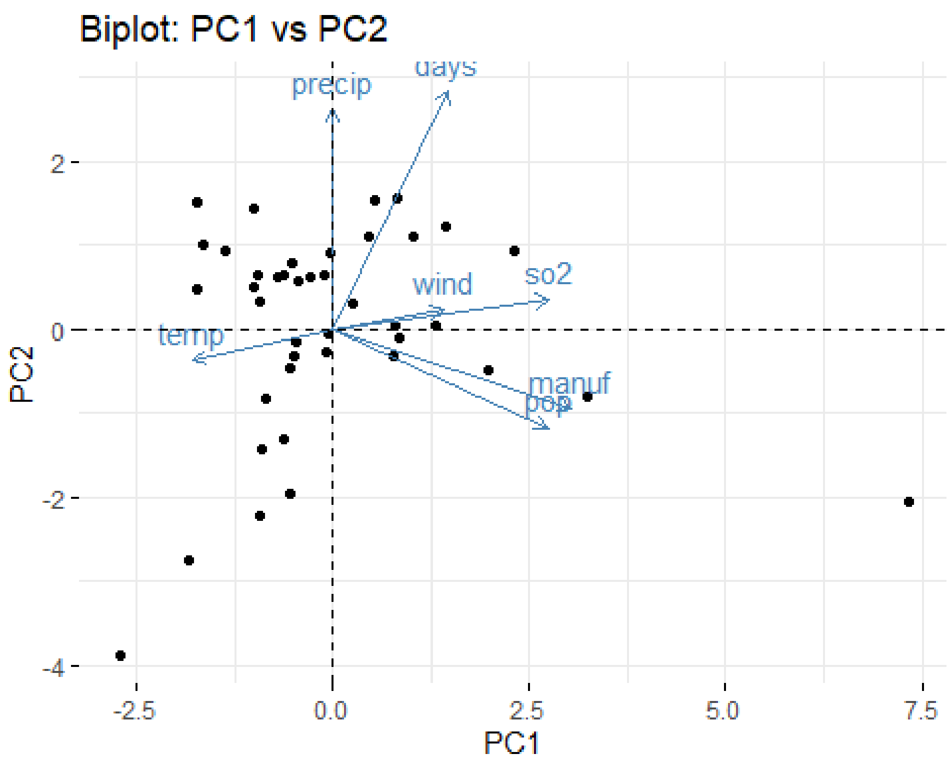
Importance of Variables

- **PC1**: Variables related to industrialization (e.g., **so2**) are critical for understanding urban pollution dynamics.
- **PC2**: Precipitation-related variables highlight environmental factors affecting air quality.
- **PC3**: Temperature influences seasonal patterns and interactions with precipitation.

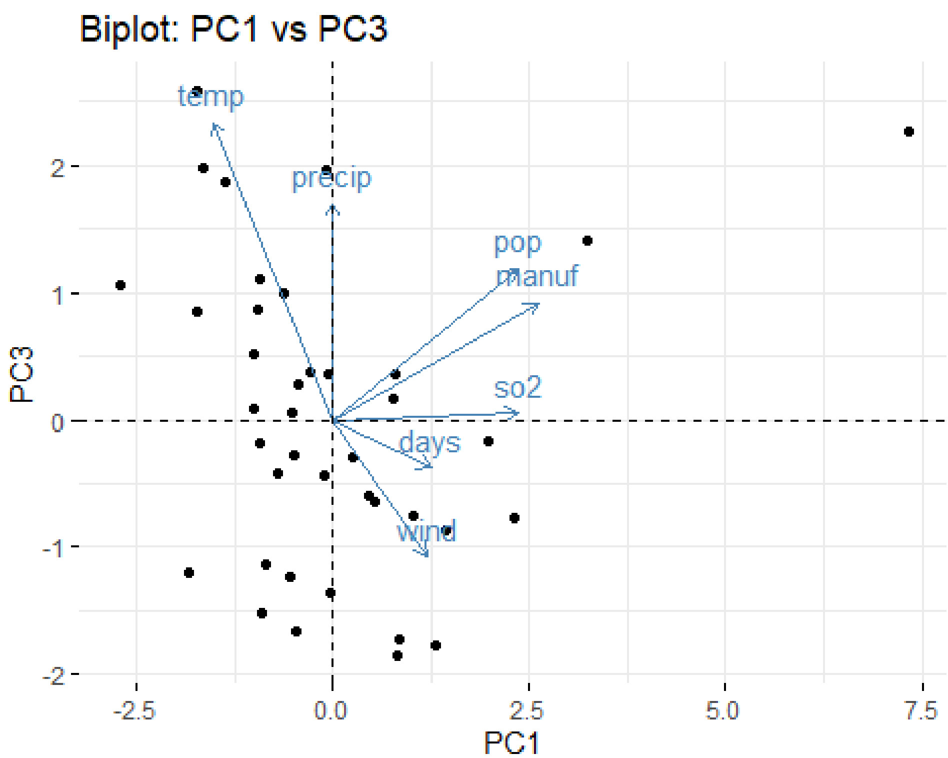
Task 4: Make a graphical representation of the principal components and present relevant results.

We created multiple different Biplot representations, as we considered 3 relevant principal components. Therefore, we analyzed the 3 possible combinations between each principal component, as following: PC1 vs PC2, PC1 vs PC3, PC2 vs PC3.

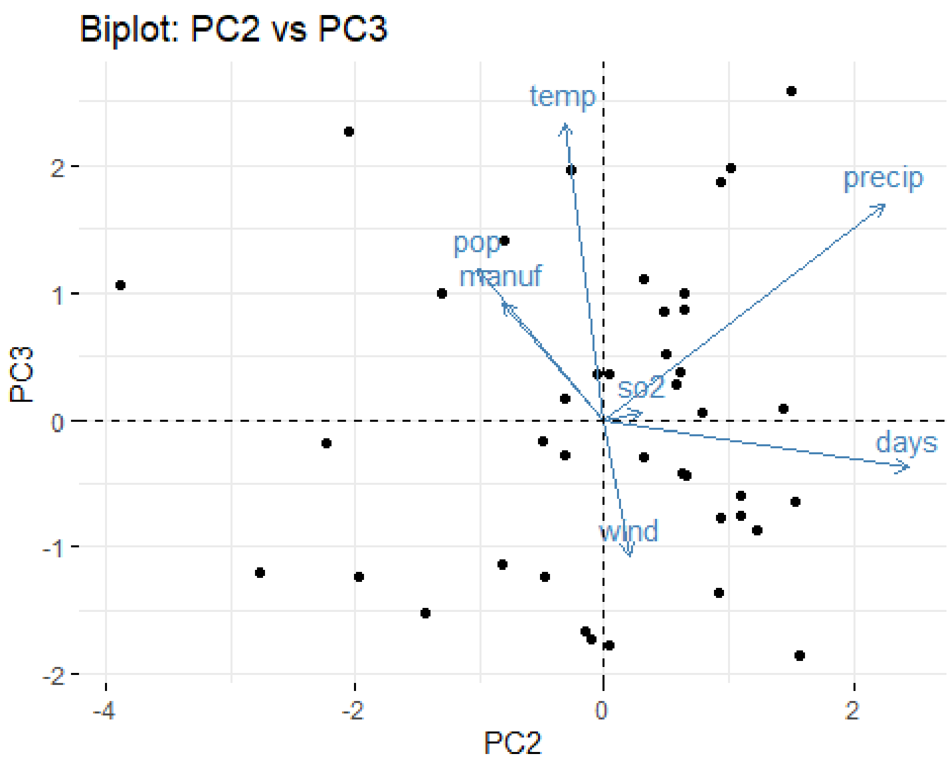
```
# Graphical representation of the principal components

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.4.2

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

# Biplot for PC1 vs PC2
fviz_pca_biplot(pca_pollution, axes = c(1, 2), geom.ind = "point", label
= "var") +
  xlab("PC1") +
  ylab("PC2") +
  ggtitle("Biplot: PC1 vs PC2")
```




```
# Biplot for PC1 vs PC3
fviz_pca_biplot(pca_pollution, axes = c(1, 3), geom.ind = "point", label
= "var") +
  xlab("PC1") +
  ylab("PC3") +
  ggtitle("Biplot: PC1 vs PC3")
```




```
# Biplot for PC2 vs PC3
fviz_pca_biplot(pca_pollution, axes = c(2, 3), geom.ind = "point", label
= "var") +
  xlab("PC2") +
  ylab("PC3") +
  ggtitle("Biplot: PC2 vs PC3")
```



Based on the conclusions regarding the influence of each variable on the 3 principal components, we considered the following data as relevant for each Biplot representation:

- **PC1 vs PC2**: **so2**, **manuf**, **pop**, **precip**, **days**
- **PC1 vs PC3**: **so2**, **manuf**, **pop**, **temp**
- **PC2 vs PC3**: **precip**, **days**, **temp**

The first plot indicates a strong positive correlation between **manuf** and **pop**, a strong positive correlation **wind** and **so2**, a strong negative correlation between **temp** and **wind** and a strong negative correlation between **temp** and **so2.** The second plot indicates a strong positive correlation between **manuf** and **pop** and a strong negative correlation between **temp** and **wind.** The third plot indicates a strong positive correlation between **manuf** and **pop**, a strong positive correlation between **days** and **so2** and a strong negative correlation between **temp** and **wind**.

Task 5: Perform a k-means clustering.

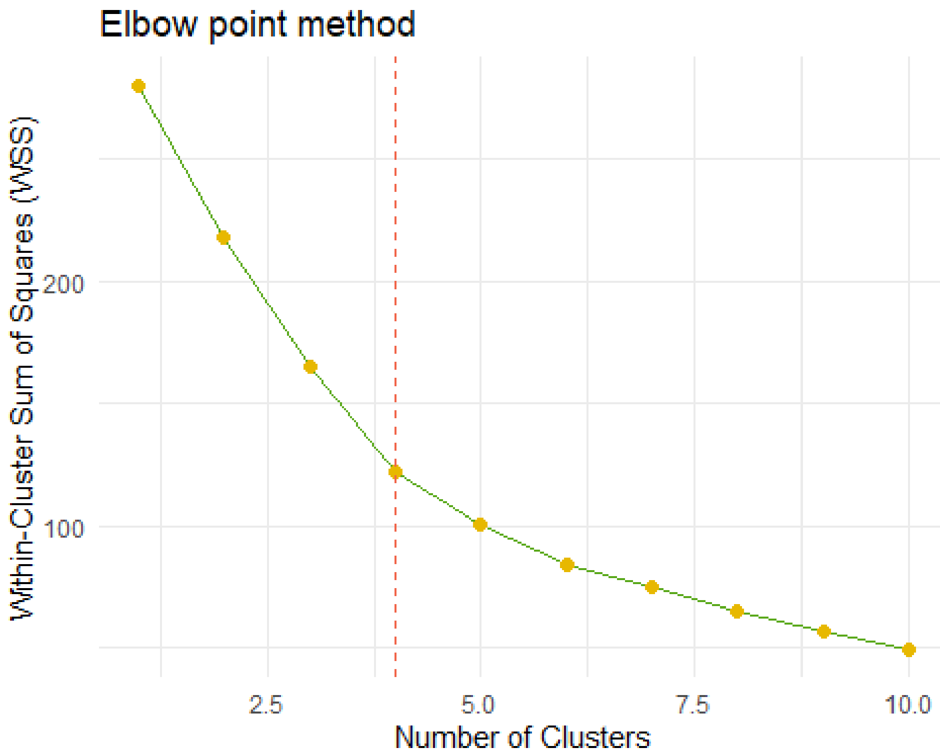
As the data range has a high variability, the first step is standardizing the data.

```
# Data scaling

pollution_scaled <- scale(pollution_dataset)
```

Although it isn't the most accurate technique, we decided to use the widely used Elbow point method in order to obtain the number of clusters that we want to obtain.

For implementing this, we started by computing the Within-Cluster Sum of Squares (WSS), and plotted it against the number of potential clusters for the algorithm.

```r
# Elbow point method

set.seed(3123) # For reproducibility
wss <- sapply(1:10, function(k) {
  kmeans(pollution_scaled, centers = k, nstart = 25)$tot.withinss
})

elbow_plot <- data.frame(
  Clusters = 1:10,
  WSS = wss
)

ggplot(elbow_plot, aes(x = Clusters, y = WSS)) +
  geom_line(color = "#59A81A", linewidth = 0.5) + # Line for WSS
  geom_point(color = "#E7B800", size = 2) + # Points for each WSS value
  geom_vline(xintercept = 4, linetype = "dashed", color = "#F06449") + #
Optional elbow point
  labs(
    title = "Elbow point method",
    x = "Number of Clusters",
    y = "Within-Cluster Sum of Squares (WSS)"
  ) +
  theme_minimal()
```



As it observed in the graphical representation of WWS(Number of Clusters), we determined the first elbow point as corresponding to 4 clusters, therefore we will use this number within the K-means Algorithm.

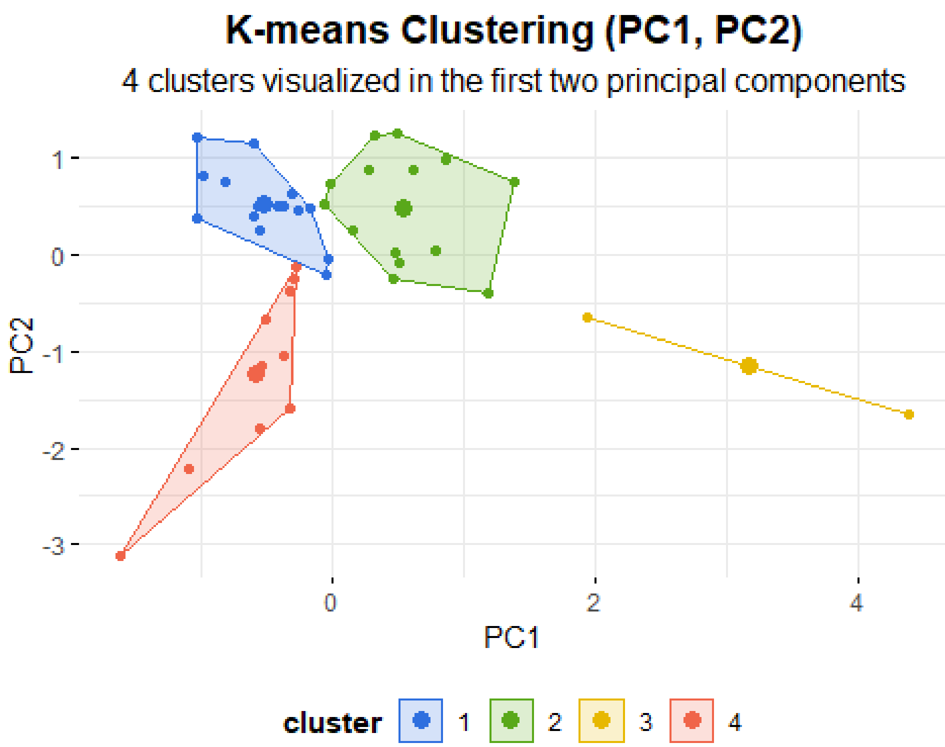
```
# K-means algorithm

set.seed(3123) # Setting a seed for the algorithm grants reproducibility
kmean <- kmeans(pollution_scaled, 4) #
kmean

## K-means clustering with 4 clusters of sizes 15, 14, 2, 10
##
## Cluster means:
##          so2        temp      manuf          pop       wind
precip
## 1 -0.3883496  0.75034106 -0.3564156 -0.277337496 -0.5533702
0.84364215
## 2  0.5426374 -0.80926380  0.1012386 -0.001916303  0.4092474 -
0.06223444
## 3  2.5328276 -0.43767833  3.6468455  3.541433470  0.3892485
0.03533737
## 4 -0.6837336  0.09499341 -0.3364797 -0.289597625  0.1792592 -
1.18540249
##         days
## 1  0.1244061
## 2  0.7662992
## 3  0.1734509
```

```
## 4 -1.2941182
##
## Clustering vector:
##  [1] 4 1 4 4 2 1 1 1 1 1 3 2 4 4 1 1 2 2 2 4 2 4 4 2 2 1 2 2 3 2 2 1 1
4 1 4 1 1
## [39] 2 1 2
##
## Within cluster sum of squares by cluster:
## [1] 34.505046 39.301901  9.278848 40.473268
##  (between_SS / total_SS =  55.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Task 6: Make a graphical representation of the clusterings obtained.

Even if we decided that there are 3 principal components that are relevant for our data, we graphically represented the 4 clusters using, initially, only the first 2 principal components, for a 2D Representation.

```
#2D Representation of the clusters

fviz_cluster(
  kmean,
  data = pca_pollution$scores[, 1:2],
  geom = "point",
  ellipse.type = "convex",
  palette = c("#2E6FDF", "#59A81A", "#E7B800", "#F06449"),
  pointsize = 1.5,
  ellipse.alpha = 0.2,
  ggtheme = theme_minimal(),
  shape = 16
) +
  labs(
    title = "K-means Clustering (PC1, PC2)",
    subtitle = "4 clusters visualized in the first two principal
components",
    x = "PC1",
    y = "PC2"
  ) +
  theme(
    plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    legend.title = element_text(face = "bold"),
    legend.position = "bottom"
  )
```



For a 3D Representation of the clusters, we represented the data using all of the 3 principal components that we considered above.

```
#3D Representation of the clusters

library(plotly)

## Warning: package 'plotly' was built under R version 4.4.2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

plot_ly(
  x = ~pca_pollution$scores[, 1], # PC1
  y = ~pca_pollution$scores[, 2], # PC2
  z = ~pca_pollution$scores[, 3], # PC3
  color = ~factor(kmean$cluster),
```

```r
  colors = c("#2E6FDF", "#59A81A", "#E7B800", "#F06449"),
  type = "scatter3d",
  mode = "markers"
) %>%
  layout(
    title = "K-means 3D Cluster Visualization (PC1, PC2, PC3)",
    scene = list(
      xaxis = list(title = "PC1"),
      yaxis = list(title = "PC2"),
      zaxis = list(title = "PC3")
    )
  )
```

```
## PhantomJS not found. You can install it with
webshot::install_phantomjs(). If it is installed, please make sure the
phantomjs executable can be found via the PATH variable.
```

Task 7: Write a brief description of each cluster.

Based on the information that we gathered from clustering the data using the K-means algorithm, we determined that:

Cluster 1: (Blue)

Moderate levels of SO2 pollution Cool temperatures High manufacturing activity Medium-sized population High wind speeds on average Moderate precipitation and a high number of rainy days

Cluster 2: (Green)

Low SO2 levels Warmer temperatures Medium manufacturing activity Medium-sized population Moderate wind speeds on average High precipitation and frequent rainy days

Cluster 3: (Orange)

Extremely high SO2 levels Cool temperatures Very high manufacturing activity Very large populations High wind speeds on average Moderate precipitation and a moderate number of rainy days

Cluster 4: (Red)

Low SO2 levels Mild temperatures Low manufacturing activity Smaller populations Lower wind speeds on average Low precipitation and fewer rainy days