Data Science Methods - Assignment 1

M. Alberti, 2020162 N.R. Ceschin

February 21, 2020

Question 1

First we upload all relevant libraries:

```r
library(readxl)
library(ggplot2)
library(ggfortify)
library(dplyr)
library(tidyr)
library(RCurl)
library(ggrepel)
```

Upload dataset:

```r
setwd("C:/Users/Mr Nobody/Desktop/Uni/EME/Data science Methods/Assignments")
#setwd("~/Tilburg/Courses/Data Science Methods/Assignment1/DATA-SCIENCE-ASSIGNMENTS")
data<-read_excel("env_air_emis.xls")
x <- getURL("https://raw.githubusercontent.com/AlbertiMarco/DATA-SCIENCE-ASSIGNMENTS/master/EU%20labels
EU_labels<- read.csv(text = x, header = FALSE ,sep=";") #import country tags to make plots more readabl
rownames(EU_labels)<-EU_labels[[1]]
```

After a quick glimpse of the data we realized that information for the five pollutant are presented in separated consecutive tables, the separation contains some information in the first column and NA cells in the rest. To be sure not to drop NAs in the middle of the dataset, we first proceed to drop all raws containing at least 5 NA values and we assign to *df*:

```r
dim(data)
df<-data[rowSums(is.na(data))<length(data)-5,]
#df<-data[complete.cases(data), ]
```

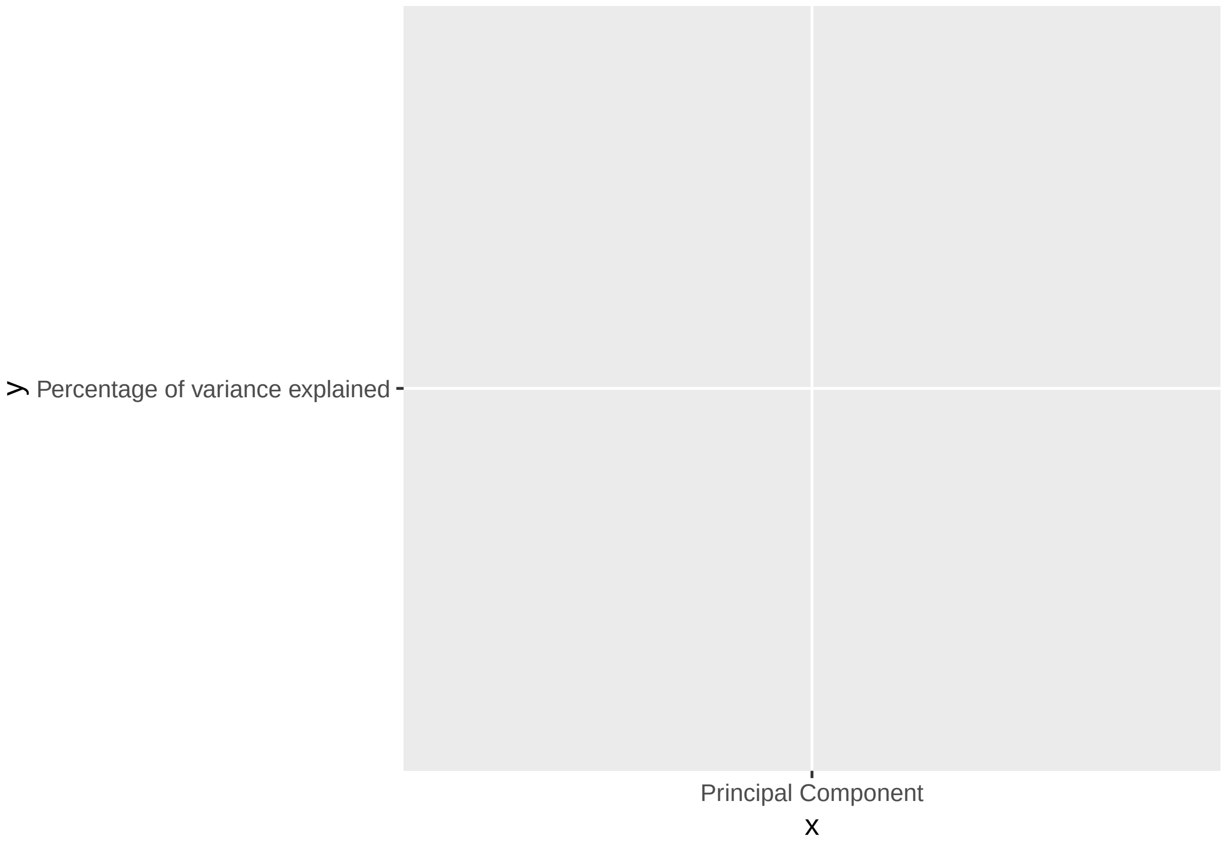
Given the data structure and the subpoints a-c requests, we decided that the optimal approach would be looping over the chunks of data containing information for each pollutant, producing without repeting the code the outputs all in one step. First we create some variables that will be used in the loop:

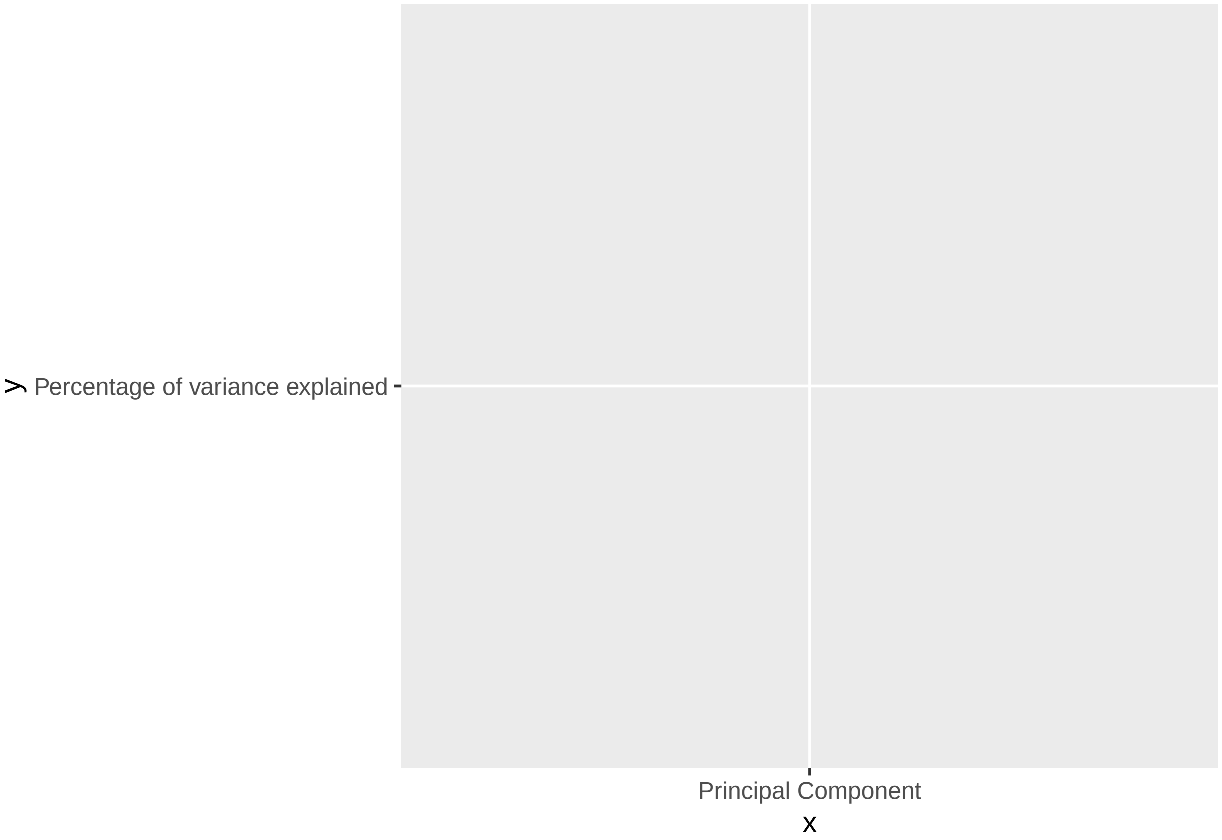
```r
#build 'index' for your loop
interval<-c(1,30,59,88,117)   #number of the first row of each individidual dataset
pollutants<-c("ammonia","nmvoc","smallpart","largepart","sulphur")
index<-data.frame(interval,pollutants)

PC1<-data.frame(matrix(ncol=5,nrow=28))  #empty data frames that will be filled with the scores
PC2<-data.frame(matrix(ncol=5,nrow=28))  # of the PC 1 and 2 for each pollutant and country
```

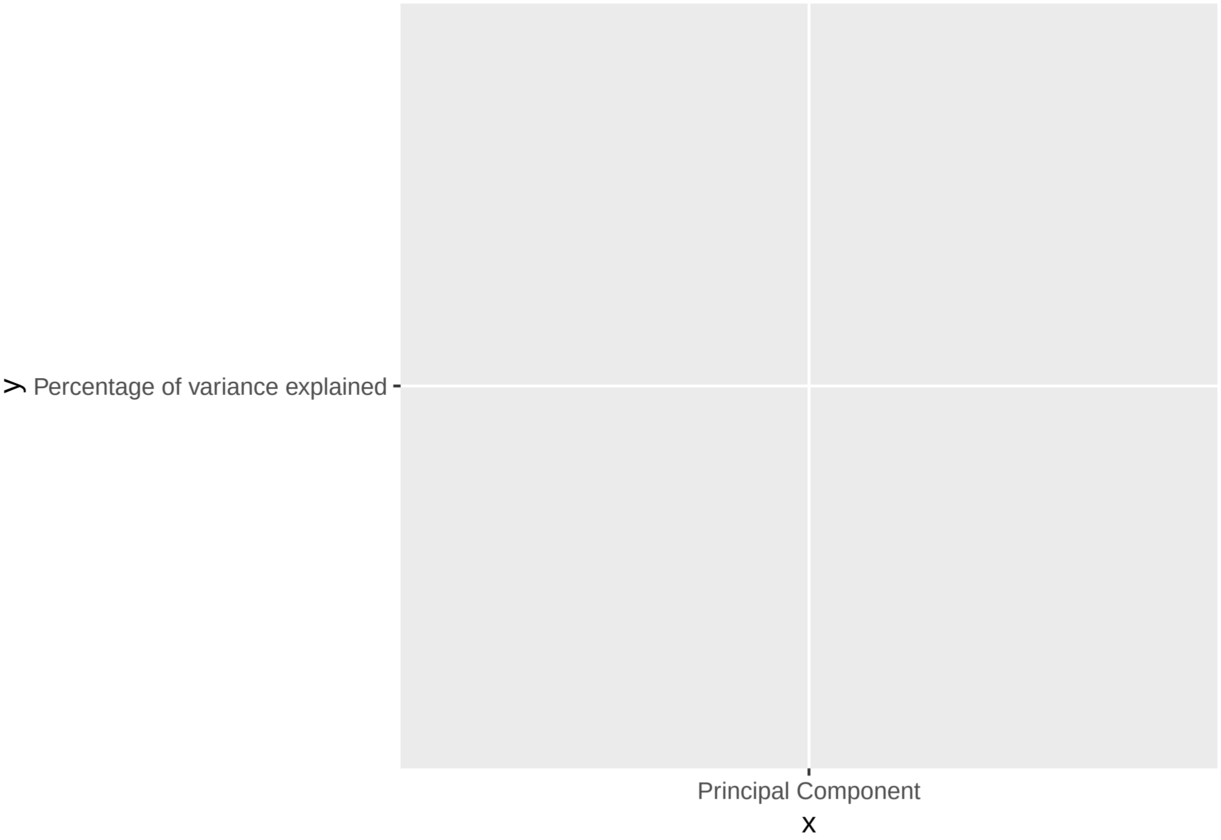
Then we run the loop that: 1) Selects the chunk of the dataframe corresponding to one pollutant and puts it into shape 2) Runs PCA on the reduced dataframe and produces the plots of the first two PCs 3) Produces a scree plot

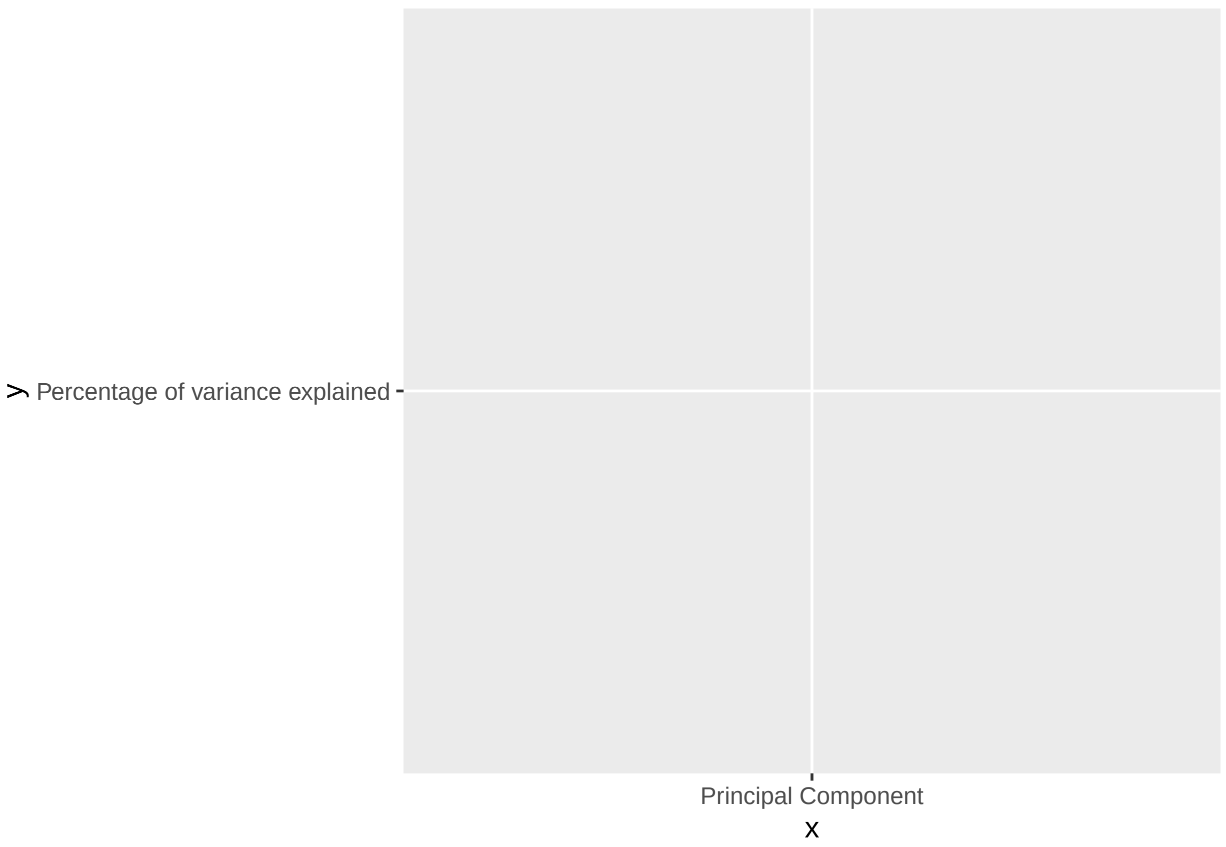
4) Computes the Bayesian Informatio Criteria and prints the optimal number of PCs according to this method
5) Stores plots and loadings for future use

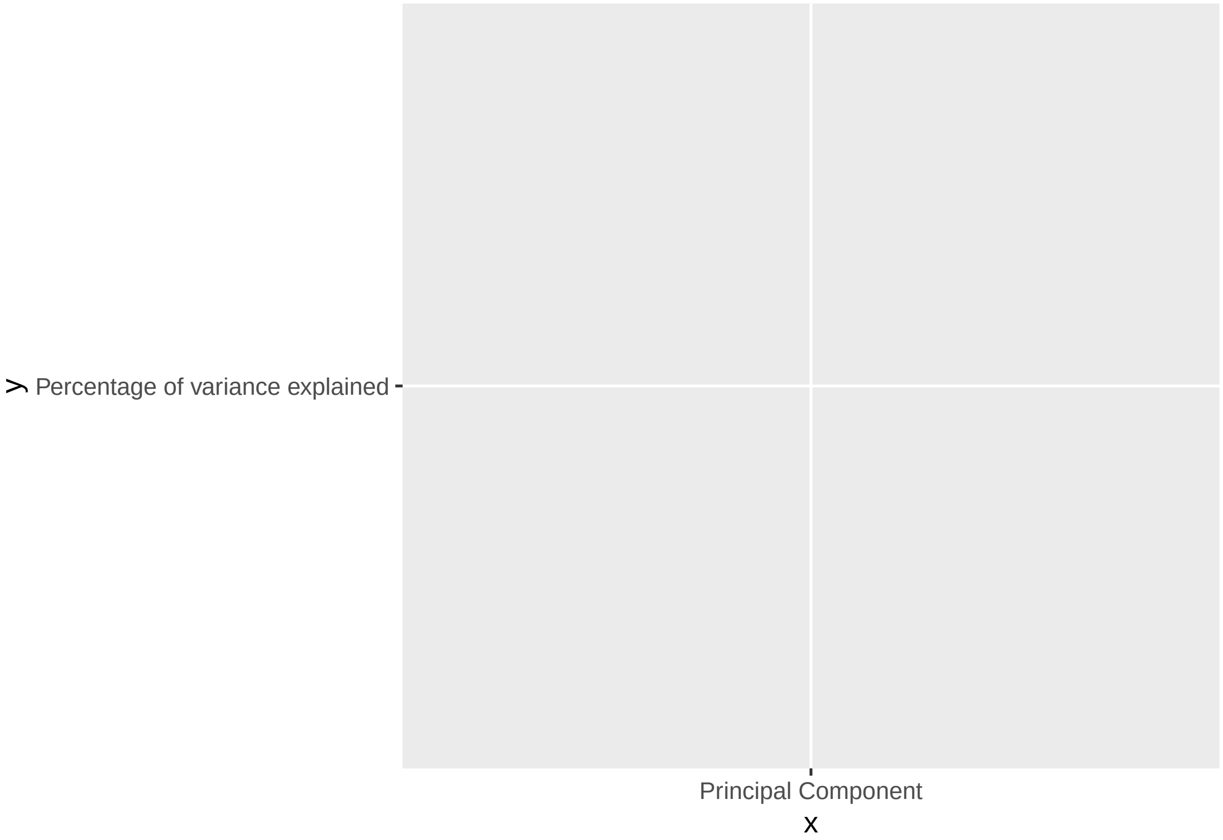
```
mytheme <- theme(plot.title= element_text(face="bold",colour = "antiquewhite4",size = (16),hjust = 0.5)

for (i in 1:5){
  #data chunk preparation
  begin<-index[i,1]
  end<-index[i,1]+28      #each chunk has 27 countries plus the first raw with years
  dfx<-df[begin:end,] #slice portion of the dataframe, 'according to begin' and 'end'
  dfx<-as.data.frame(dfx) #rename first column with the name of the pollutant
  colnames(dfx)<-dfx[1,]         #set first column as observations' names and first row as       variab
  rownames(dfx)<-dfx[,1]
  dfx<-dfx[c(2:29),c(2:29)]      #drop first column and obtain the final datset
  if (sum(mapply(grepl,rownames(EU_labels),rownames(dfx)))==length(dfx)) {
  rownames(dfx)<- EU_labels[[2]]
    } #Substitute name with short labels of the appropriate country
  dfx<-as.data.frame(t(dfx))     #convert factor columns into numeric to apply prcomp
  indx <- sapply(dfx, is.factor)
  dfx[indx] <- lapply(dfx[indx], function(x) as.numeric(as.character(x)))



  #Principal Component Analysis
  pr.out<-prcomp(dfx, scale=TRUE)
  #print(pr.out$rotation[,1:2])           # print first two PC loadings and plot first two PC
  graph<-autoplot(pr.out,variance_percentage=FALSE,loadings=TRUE,
           loadings.label=TRUE,loadings.label.repel=TRUE,loadings.colour="coral",loadings.label.size=3,l

  pve= 100* (pr.out$sdev ^2)/ sum(pr.out$sdev ^2)  #screeplot
  scree<- ggplot(data.frame(pve),aes("Principal Component", "Percentage of variance explained"))
  print(scree)
    #plot(pve , type ="o", ylab="PVE ", xlab=" Principal Component ",
       #col =" blue",axes = F)
  #axis(side = 1, at = seq(from=0,to=30))
  #axis(side = 2, at = seq(from=0,to=100,by=5))
  #title(paste("Scree plot for PCs of",toString(index[i,2]),"pollutant"))
  #grid()

  #compute vector of BIC for first 27 principal components
  BIC<-c(1:27)   #initialize a numeric vector to be filled with BIC(k) values. set max k=p-1
  for (j in 1:27) {
    f<-t(pr.out$rotation[,1:j])%*%pr.out$x[,1:j] #compute aF in X=aF+e
    res_mat<-dfx[,1:j]-f                  #compute matrix of residuals
    res_mat_sq<-res_mat*res_mat                  #square residuals
    if (j==1){
      res<-(sum(res_mat_sq)/(dim(dfx)[1]*dim(dfx)[2]))  #did separate for k=1 because of "rowSums"
    } else{
    res<-(sum(rowSums(res_mat_sq))/(dim(dfx)[1]*dim(dfx)[2]))         #residuals sum of squares
    }
    k<-j
    BICk<-log(res)+k*(log(28^2))/(28^2) #BIC for each k
    BIC[j]<-BICk                                  #fill BIC vector at each iteration
    }
  min<-min(BIC)
  num_pc<-match(min,BIC)                        #find and print k, the index of the min of BIC
  cat("According to the BIC criterion, the optimal number of principal components is ", num_pc)


  #save first two PC in separate dataset for point d)
  PC1[i]<-pr.out$x[,1]
  colnames(PC1)[i]<-as.character(index[i,2])
  PC2[i]<-pr.out$x[,2]
  colnames(PC2)[i]<-as.character(index[i,2])

  #save relevant objects with their respective name
  assign(paste0("BIC_", index[i,2]), BIC)
  assign(paste0("df_", index[i,2]), dfx)
  assign(paste0("prcomp_",index[i,2]),pr.out)
  assign(paste0("Screeplot_",index[i,2]),scree)
  assign(paste0("PC1-PC2_",index[i,2]),graph)
  #remove non relevant objects
  rm(dfx)
  rm(BIC)
  rm(pr.out)
}
```




y
Percentage of variance explained
Principal Component
x


y
Percentage of variance explained
Principal Component
x


y
Percentage of variance explained
Principal Component
x



Now to comment on the results we can called the corresponding stored object.

```
print("PC1-PC2_sulphur")
```

```
## [1] "PC1-PC2_sulphur"
```

d)

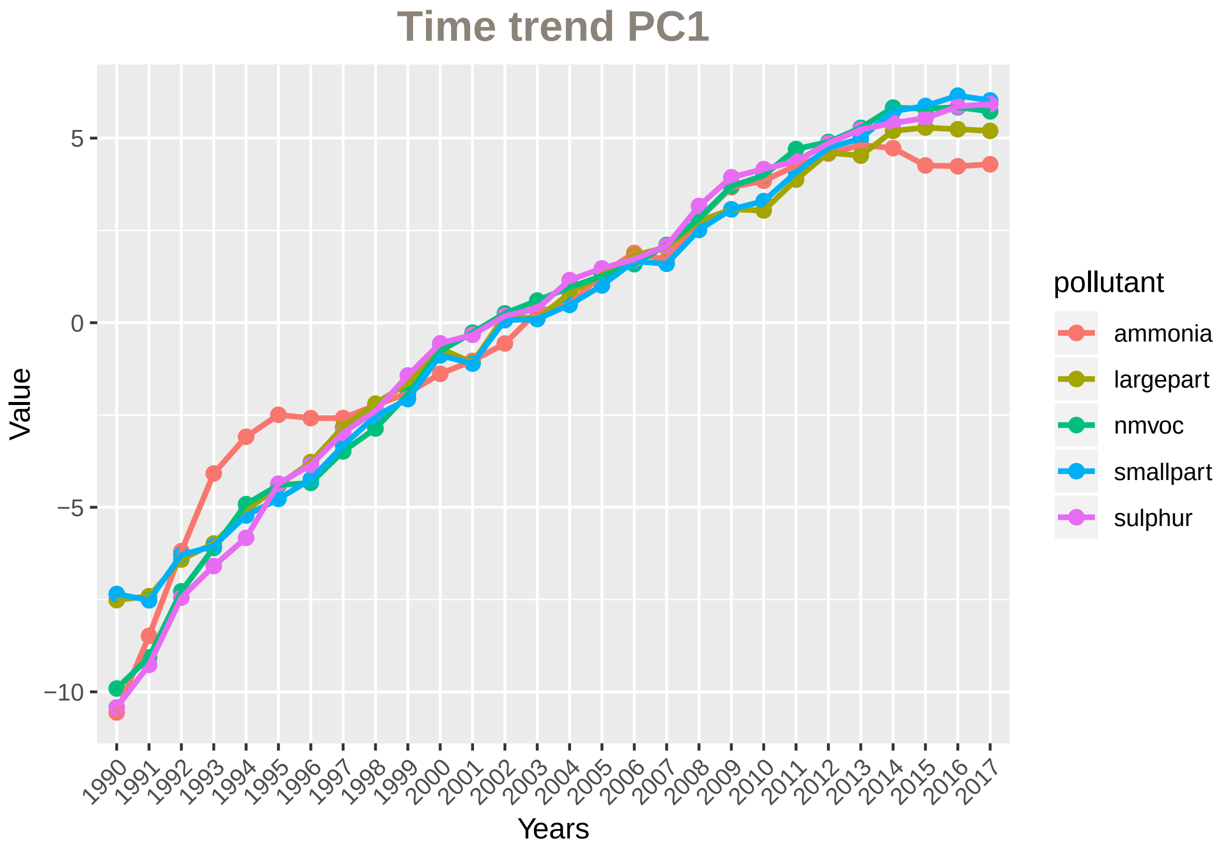
We plot the first 2 principal components for all the five pollutants over the time interval.

Code to produce the plots:

```
theme2<-theme(axis.text.x = element_text(angle = 45, hjust=1))
PC1["years"]<-c(1990:2017)
PC1<-gather(PC1, `ammonia`, `largepart`, `nmvoc`,`smallpart`, `sulphur`, key = "pollutant", value = "val
PC1_plot<-ggplot(PC1,aes(x=factor(years),y=value, group=pollutant,color=pollutant))+
  geom_point(size = 2.25) +  geom_line(size = 1) +theme2 +mytheme+
  labs(title = "Time trend PC1",x="Years",y="Value")

PC2["years"]<-c(1990:2017)
PC2<-gather(PC2, `ammonia`, `largepart`, `nmvoc`,      #transform data to be plotted
            `smallpart`, `sulphur`, key = "pollutant", value = "value")
PC2_plot<-ggplot(PC2,aes(x=factor(years),y=value, group=pollutant,color=pollutant))+
  geom_point(size = 2.25) +  geom_line(size = 1) +theme2 +mytheme+
  labs(title = "Time trend PC2",x="Years",y="Value")
```

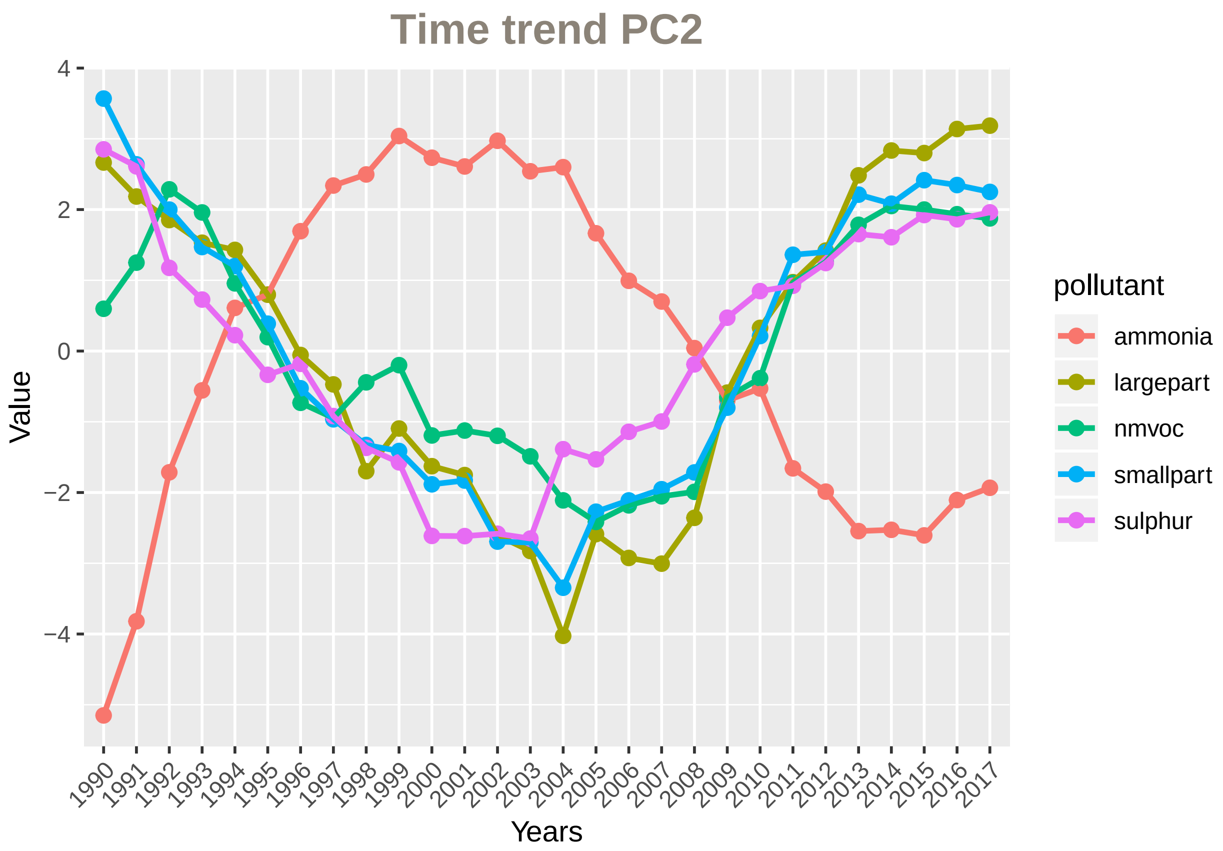
Here is the plot of the first principal component over time for the five pollutant in the dataset:

```
print(PC1_plot)
```



Here is the same plot for the second principal component:

```
print(PC2_plot)
```



Question 2

A factor model is defined as:

$$X_{ij} = \alpha_{j1} f_{i1} + \alpha_{j2} f_{i2} + ... + \alpha_{jr} f_{ir} + \epsilon_{ij} = \alpha_j^T f_i + \epsilon_{ij}$$

with i=1,. . . ,n ; j=1,. . . ,p and r number of factors $f_i = \begin{bmatrix} f_{i1} \\ f_{i2} \\ \vdots \\ f_{ir} \end{bmatrix}$ and $\alpha_j = \begin{bmatrix} \alpha_{j1} & .. & \alpha_{jr} \end{bmatrix}$

In matrix form we can write:

$$\underset{n\times p}{X} = \underset{n\times r}{F}\underset{r\times p}{B^T} + \underset{n\times p}{\epsilon}$$

In general the expression above is not identified since we don't observe neither B nor F. If we consider a particular rotation matrix H, with $H^T H = I_r$ then we can rewrite the equation above as:

$$X = \underbrace{FH}_{F^*}\underbrace{H^T B^T}_{B^*} + \epsilon$$

In the particular case of PCA, the NIPALS algorithm can find a unique solution because PCA imposes a specific rotation matrix H. In particular, the rotation matrix imposes restrictions on B, on one hand $\frac{r(1-r)}{2}$ restrictions of orthogonality of the loadings and r on the norm of the loadings (normalization):

(i) $a_i a_j = 0$ when $a_i \neq a_j$

(ii) $a_i^T a_i = 1$

Of course the PCA solution is just one of the possibles, any restrictions that identify the model can lead to a unique solution(for example restrictions on the factors instead).

In a factor model we believe that observed variables (Xs) are linear combinations of a limited number of underlying and unique factors (Zs); whereas in PCA component scores (Zs) are a linear combination of the observed variables (Xs) weighted by eigenvectors. The relationship between the two can be summarized by:

$$XH = UDH^T H = UD = Z$$

with $U^T H = I_p$, $H^T H = HH^T = I_p$, $D = diag(d)$ of size $p \times p$. This shows that there is a transformation matrix (inverse of H, which in the case of orthogonal H is equal to H^T) for which the explanatory variables are linear combinations of the unobserved unique factors Z.

Concistency of NIPALS Algorithm and factor analysis

In class we have seen how the NIPALS algorithm can be used to consistently estimate the scores in a principal components analysis (PCA) when the traditional optimization approach fails or is inefficient given the large number of parameters (p). In particular, we limited ourselves to the case in which $n > p$ and the matrix D of eigenvalues has rank p.
The starting point is

$$X = F^* B^* + \epsilon$$

Under the assumptions:

1. ϵ_{ij} are iid over i and j (can be relaxed to weakly dependent)
2. $$E||f_i||^4 < M$$ $$\frac{1}{n}\sum_{i=1}^{\infty} f_i f_i^T \xrightarrow{p} \Sigma_F$$ which is a positive definite matrix of constants. This assumption is related to saying that the matrix D of eigenvalues has rank p, since positive definiteness for a symmetric matrix can be defined as having a diagonal of non-zero elements
3. f_{ij}, ϵ_{ij} are independent for all j (Exogeneity Assumption)
4. $\underset{r \times r}{H}$ is defined as above (r are restrictions)

Given these assumptions we can consistently estimate $F^* = FH$ by $\hat{F}$ meaning that we have:

1. $\hat{f}_i \xrightarrow{p} Hf_i$ pointwise consistency
2. $min(\sqrt{n}, \sqrt{p})[\hat{f}_i - Hf_i] \xrightarrow{d} N(0, HV(f_i)H^T)$ for $min(\sqrt{n}, \sqrt{p}) \to \infty$

If the assumptions 1-4 are met, also NIPALS method with the PCA restrictions can identify a factor model up to a rotation matrix H. Notice that for consistency to hold, we need both the number of observations n and the number of variables p to be large.

Question 3

a) $p_1(x) = \frac{Pr(Y=1)Pr(X=x|Y=1)}{\sum_{l=1}^{K} Pr(Y=l)Pr(X=x|Y=1)}$

In this case we have that: - $\pi_k = Pr(Y = k)$ is prior probability for class k - $X|Y = k \sim f_k(x) = N(\mu_k, \sigma^2)$ is the distribution of X for each class k

Using the Bayes rule to rewrite: