PCA — AirPollution dataset

Hanyu Wang

0 Packages

```
library(readxl)
library(readxl)
library(GGally)
```

```
## Loading required package: ggplot2
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(heplots)
```

```
## Loading required package: broom
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
```

```
## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

1 Read data

```
file_path <- "AirPollution.xls"
dat <- read_excel(file_path)
```

```
# Ensure data are numeric
dat <- as.data.frame(dat)
for (j in 1:ncol(dat)) dat[[j]] <- as.numeric(dat[[j]])
```

```
# Check data
cat("n =", nrow(dat), " p =", ncol(dat), "\n")
```

```
## n = 42  p = 7
```

```
summary(dat)
```

```
##       Wind         Radiation            CO              NO      
##  Min.   : 5.00   Min.   : 30.00   Min.   :2.000   Min.   :1.00  
##  1st Qu.: 6.00   1st Qu.: 68.25   1st Qu.:4.000   1st Qu.:1.00  
##  Median : 8.00   Median : 76.50   Median :4.000   Median :2.00  
##  Mean   : 7.50   Mean   : 73.86   Mean   :4.548   Mean   :2.19  
##  3rd Qu.: 8.75   3rd Qu.: 84.75   3rd Qu.:5.000   3rd Qu.:3.00  
##  Max.   :10.00   Max.   :107.00   Max.   :7.000   Max.   :5.00  
##       NO2              O3               HC       
##  Min.   : 5.00   Min.   : 2.000   Min.   :2.000  
##  1st Qu.: 8.00   1st Qu.: 6.000   1st Qu.:3.000  
##  Median : 9.50   Median : 8.500   Median :3.000  
##  Mean   :10.05   Mean   : 9.405   Mean   :3.095  
##  3rd Qu.:12.00   3rd Qu.:11.000   3rd Qu.:3.000  
##  Max.   :21.00   Max.   :25.000   Max.   :5.000  
```

```
colSums(is.na(dat))
```

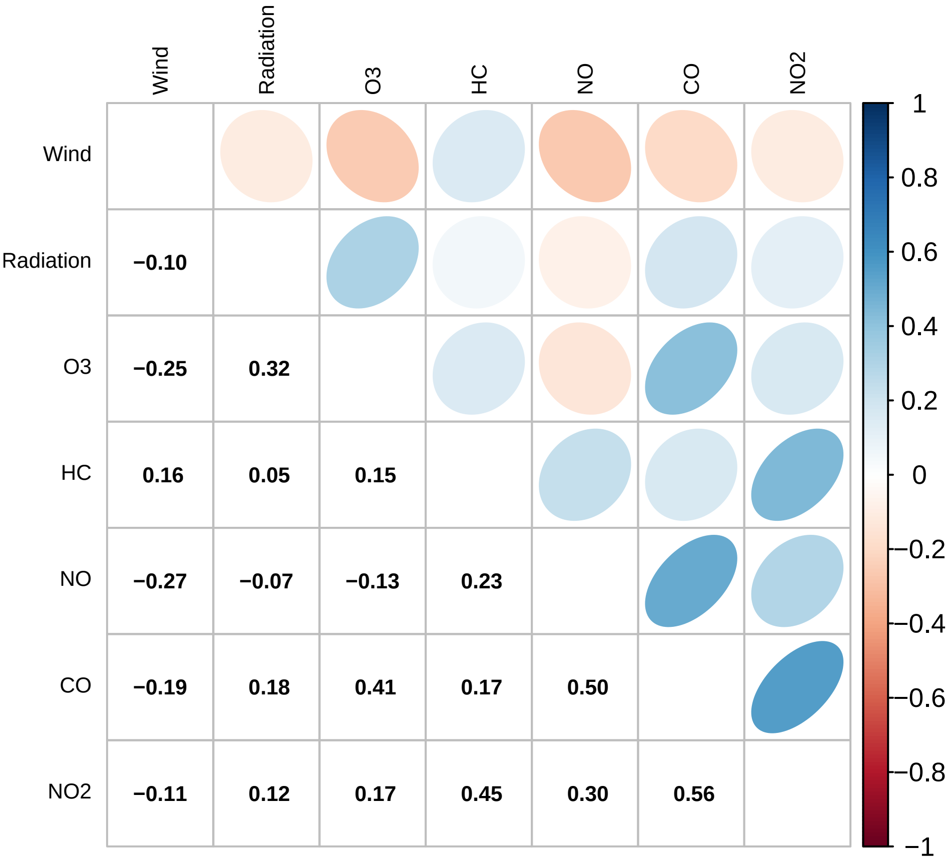
```
##      Wind Radiation        CO        NO       NO2        O3        HC 
##         0         0         0         0         0         0         0 
```

1) First, provide some plots showing relationships between your variables (i.e.scatterplots, etc). Discuss what you see, thinking in particular about high-dimensional linearity.

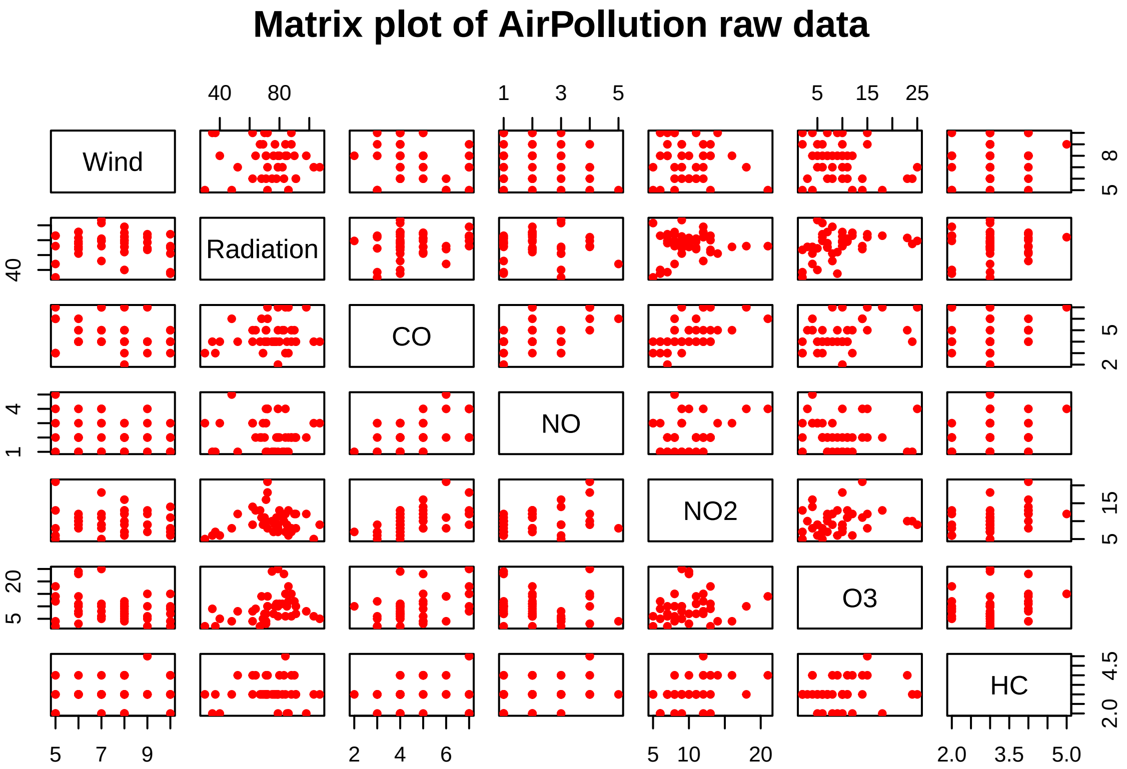
```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
corrplot.mixed(cor(dat), lower.col = "black", upper = "ellipse",
               tl.col = "black", number.cex = .7, order = "hclust",
               tl.pos = "lt", tl.cex = .7)
```



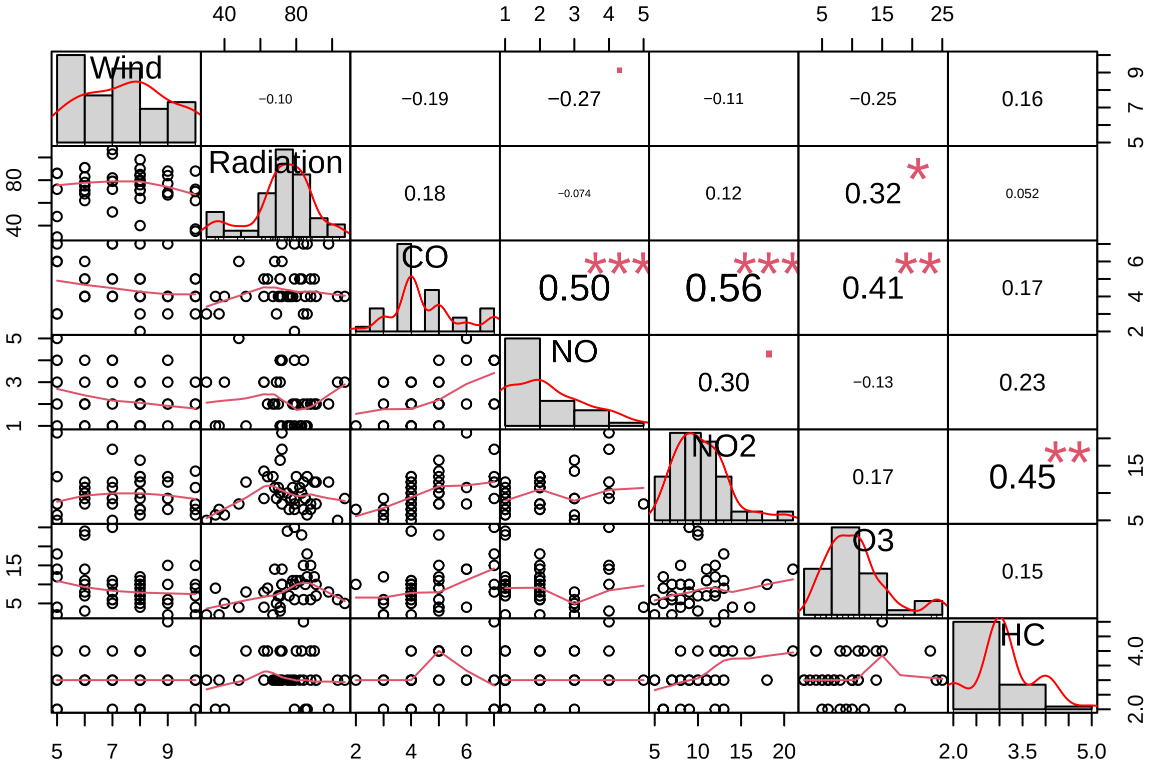
```
#make matrix plot to check for linearity
plot(dat, pch = 19, cex = .7, col = 'red',
     main = "Matrix plot of AirPollution raw data")
```



```
#Here is a cool way to look for non-linearity, get correlation, make histograms all at once.
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```
chart.Correlation(dat)
```

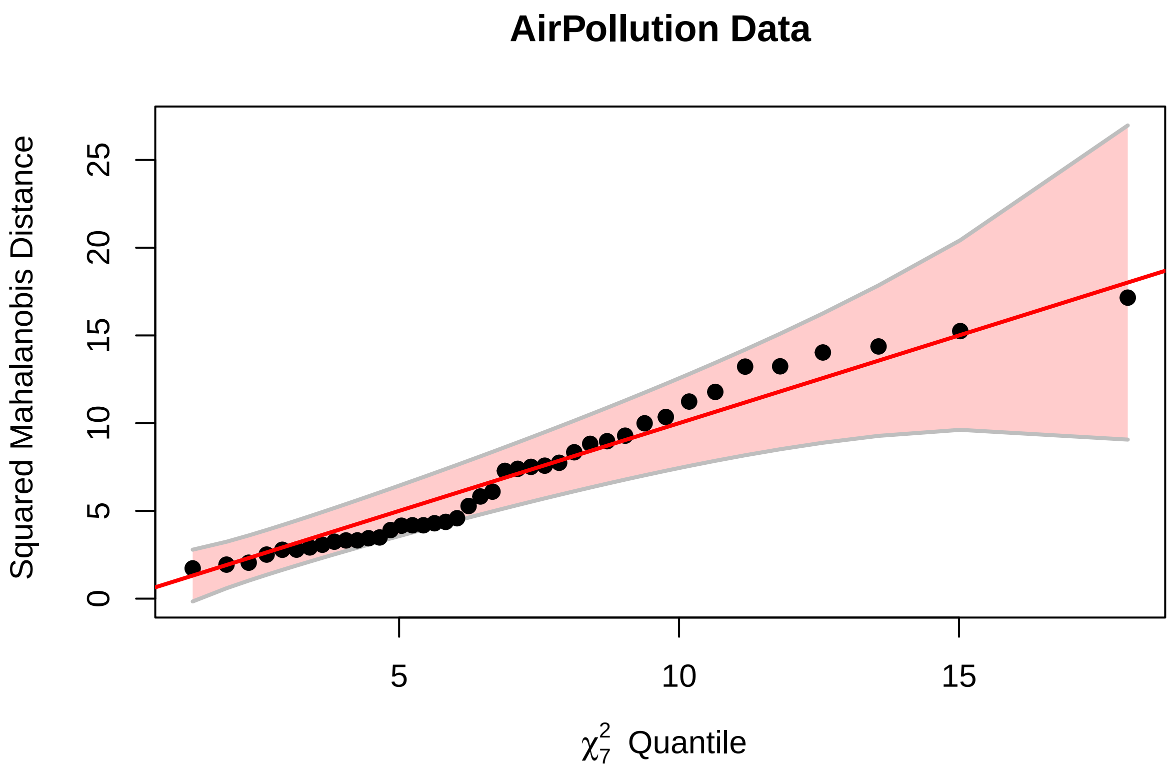


Overall, the variables exhibit a moderate linear correlation structure. Several air pollutant variables (CO, NO, NO2, O3, HC) tend to be positively correlated with one another, while Wind shows negative associations with some pollutant measures. This pattern suggests the presence of one or more underlying linear directions that may explain shared variation among the variables.

From a high-dimensional linearity perspective, the scatterplot matrices do not reveal strong nonlinear patterns such as curved, circular, or multi-modal relationships. Most pairwise relationships can be reasonably approximated as linear. Therefore, from a geometric standpoint, applying PCA to summarize dominant directions of variability is appropriate for this dataset.

Next, examine the multivariate normality of your data – makeplots as appropriate including a chi-square quantile plot.

```r
cqplot(dat, main = "AirPollution Data")
```



In the raw data, most observations align reasonably well with the reference line in the central region, but noticeable deviations occur in the upper tail, indicating departures from strict multivariate normality.

If you decide to make transformations of variables, make at least some post-transformation plots (including a new chi-square quantile plot) and again discuss linearity and multivariate normality. NOTE that multivariate normality is NOT a requirement for PCA to work!

```r
#start with prcomp
comp1 <-prcomp(dat)
summary(comp1)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6    PC7
## Standard deviation     17.443 5.31753 3.38592 1.58881 1.13116 0.72714 0.4578
## Proportion of Variance  0.873 0.08113 0.03289 0.00724 0.00367 0.00152 0.0006
## Cumulative Proportion   0.873 0.95408 0.98697 0.99421 0.99788 0.99940 1.0000
```

```r
#Make output variances not standard deviations

summary.PCA.JDRS <- function(x){
  sum_JDRS <- summary(x)$importance
  sum_JDRS[1, ] <- sum_JDRS[1, ]^2
  attr(sum_JDRS, "dimnames")[[1]][1] <- "Eigenvals (Variance)"
  sum_JDRS
}
```

```
round(summary.PCA.JDRS(comp1), 3)
```

```
##                              PC1    PC2    PC3   PC4   PC5   PC6   PC7
## Eigenvals (Variance)     304.258 28.276 11.464 2.524 1.280 0.529 0.210
## Proportion of Variance     0.873  0.081  0.033 0.007 0.004 0.002 0.001
## Cumulative Proportion      0.873  0.954  0.987 0.994 0.998 0.999 1.000
```

```
#rotation - same as what we calculated by hand
round(comp1$rotation, 3)
```

```
##              PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Wind      -0.010  0.076 -0.031  0.920  0.342  0.012  0.170
## Radiation  0.993  0.116 -0.007  0.000  0.002  0.003  0.002
## CO         0.014 -0.100  0.183 -0.138  0.650 -0.564 -0.444
## NO        -0.005  0.013  0.130 -0.328  0.643  0.498  0.463
## NO2        0.024 -0.150  0.955  0.102 -0.207 -0.009  0.105
## O3         0.112 -0.973 -0.170  0.063  0.000  0.051  0.067
## HC         0.002 -0.024  0.085  0.110  0.062  0.657 -0.738
```

```
#get means just FYI
comp1$center
```

```
##      Wind Radiation        CO        NO       NO2        O3        HC
##  7.500000 73.857143  4.547619  2.190476 10.047619  9.404762  3.095238
```

```
#get total variance
sum(comp1$sdev^2)
```

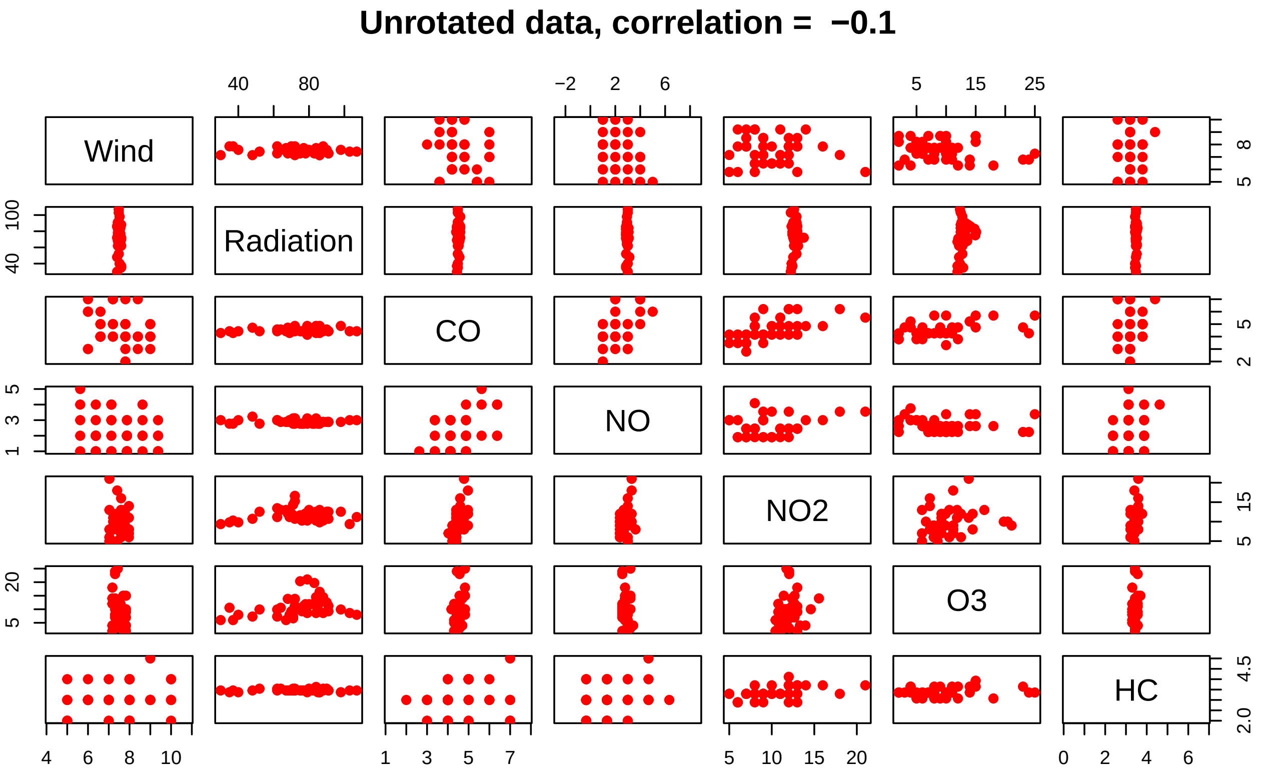
```
## [1] 348.5407
```

```
#confirm this is the same as the total variance of the original variables
sum(apply(dat, 2, var))
```

```
## [1] 348.5407
```

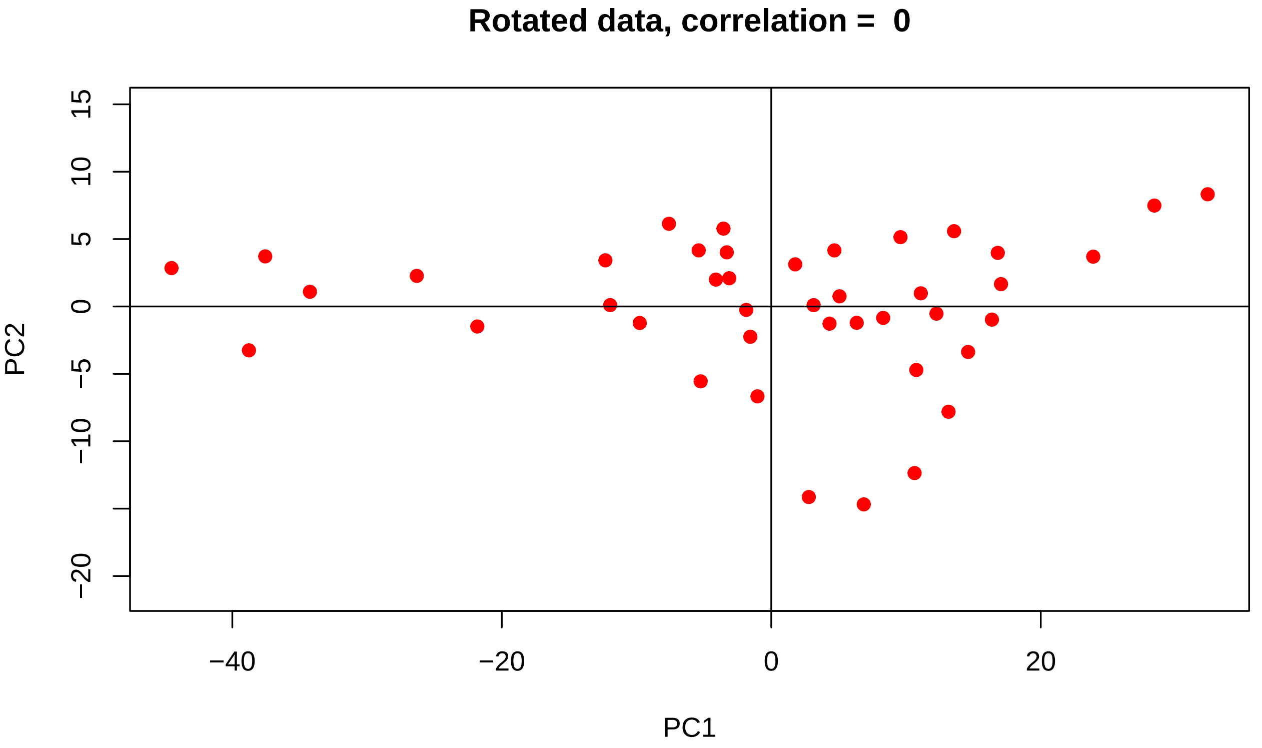
Show plot of data before and after rotation:

```
#plot data before rotation and include correlation
plot(dat, pch = 19, asp = 1, col = 'red',
     main = paste("Unrotated data, correlation = ", round(cor(dat)[1,2],2)))
abline(h = 3, v = 8)
```



Unrotated data, correlation = −0.1

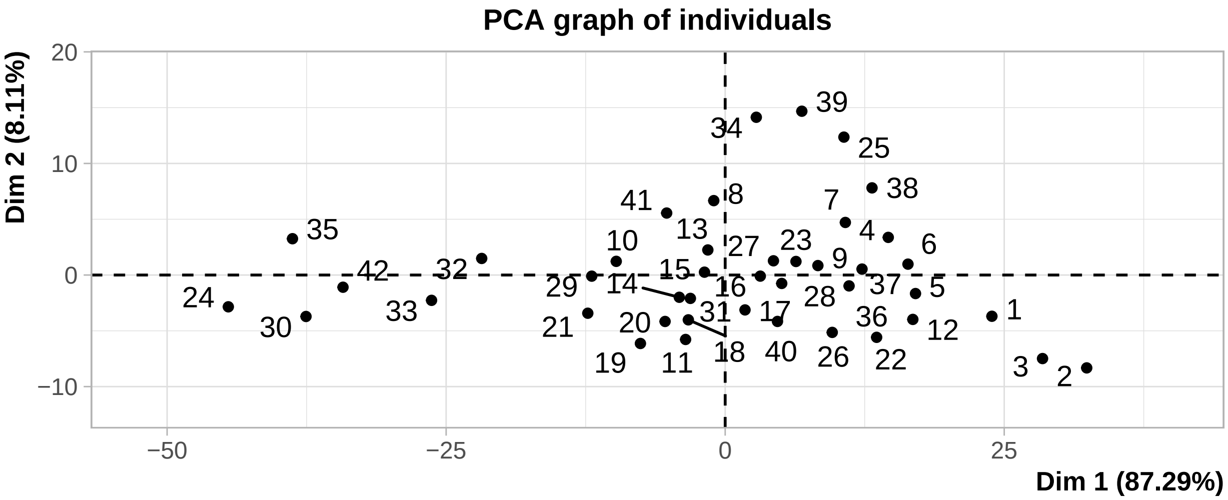
```r
#plot data after rotation and include correlation
plot(comp1$x, pch = 19, asp = 1, col = 'red',
     main = paste("Rotated data, correlation = ", round(cor(comp1$x)[1,2],2)))
abline(h = 0, v = 0)
```

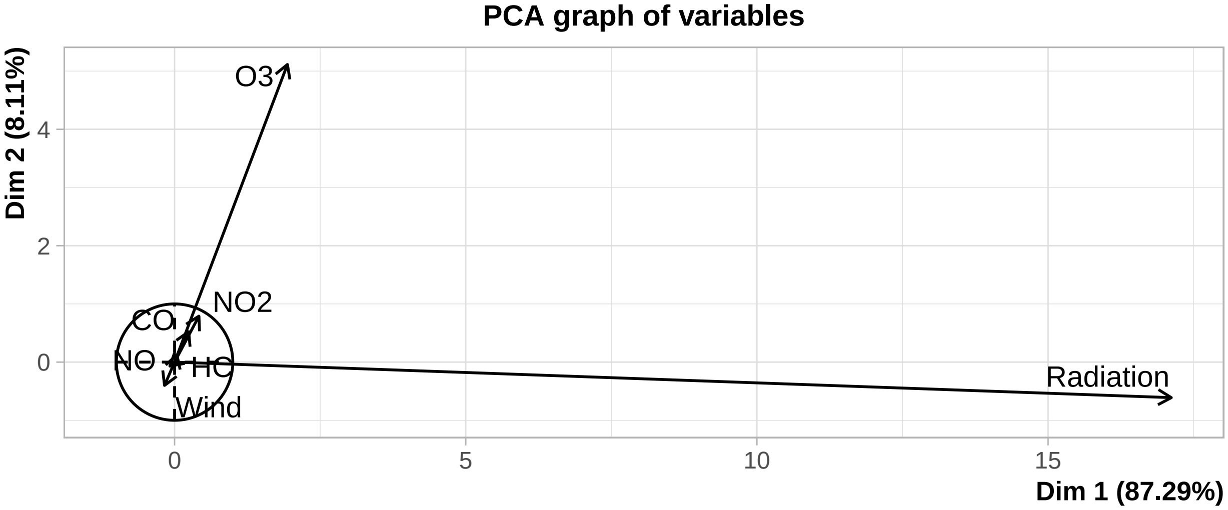


Rotated data, correlation = 0

Let's see what happens if we use the princomp() or PCA() functions.

```r
library(FactoMineR)

#use PCA() function
pc1_a <- PCA(dat, scale.unit = F)
```





```r
#NOTE - eigenvalues are DIFFERENT although proportion explained is the same
summary(pc1_a)
```

```
##
## Call:
## PCA(X = dat, scale.unit = F)
##
##
## Eigenvalues
##                          Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               297.014  27.603  11.192   2.464   1.249   0.516   0.205
## % of var.               87.295   8.113   3.289   0.724   0.367   0.152   0.060
## Cumulative % of var.    87.295  95.408  98.697  99.421  99.788  99.940 100.000
##
## Individuals (the 10 first)
##                 Dist     Dim.1     ctr    cos2     Dim.2     ctr    cos2     Dim.3
## 1          | 24.416 | 23.896  4.577  0.958 | -3.696  1.178  0.023 |  2.259
## 2          | 33.469 | 32.390  8.410  0.937 | -8.324  5.977  0.062 | -0.459
## 3          | 29.792 | 28.433  6.481  0.911 | -7.488  4.836  0.063 | -4.423
## 4          | 15.583 | 14.610  1.711  0.879 |  3.374  0.982  0.047 | -2.942
## 5          | 17.350 | 17.052  2.331  0.966 | -1.660  0.238  0.009 | -2.257
## 6          | 16.506 | 16.377  2.150  0.984 |  0.976  0.082  0.003 |  1.437
## 7          | 12.376 | 10.766  0.929  0.757 |  4.713  1.916  0.145 |  1.648
## 8          | 12.529 | -1.023  0.008  0.007 |  6.668  3.836  0.283 | 10.350
## 9          |  8.464 |  8.307  0.553  0.963 |  0.847  0.062  0.010 |  0.520
## 10         | 10.361 | -9.760  0.764  0.887 |  1.226  0.130  0.014 |  3.074
##                ctr   cos2
## 1            1.086  0.009 |
## 2            0.045  0.000 |
## 3            4.162  0.022 |
## 4            1.841  0.036 |
## 5            1.084  0.017 |
## 6            0.440  0.008 |
## 7            0.578  0.018 |
## 8           22.789  0.682 |
## 9            0.058  0.004 |
## 10           2.010  0.088 |
##
## Variables
##                Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
## Wind      | -0.173  0.010  0.012 | -0.400  0.581  0.066 | -0.103  0.095  0.004
## Radiation | 17.117 98.645  0.999 | -0.610  1.349  0.001 | -0.022  0.004  0.000
## CO        |  0.242  0.020  0.040 |  0.523  0.991  0.184 |  0.612  3.343  0.252
## NO        | -0.081  0.002  0.006 | -0.069  0.017  0.004 |  0.436  1.696  0.164
## NO2       |  0.418  0.059  0.016 |  0.790  2.261  0.056 |  3.196 91.253  0.921
## O3        |  1.938  1.264  0.124 |  5.114 94.743  0.865 | -0.568  2.884  0.011
## HC        |  0.040  0.001  0.003 |  0.125  0.057  0.034 |  0.285  0.726  0.174
##
## Wind      |
## Radiation |
## CO        |
## NO        |
## NO2       |
## O3        |
## HC        |
```

```
#This means sum of eigenvalues is NOT total variance
sum(apply(dat, 2, var))
```

```
## [1] 348.5407
```

```
sum(pc1_a$eig[,1])
```

```
## [1] 340.2421
```

```
names(pc1_a)
```

```
## [1] "eig"  "var"  "ind"  "svd"  "call"
```

```
#Make nice output for loadings
pc1_loads <- data.frame(round(pc1_a$svd$V, 3))
rownames(pc1_loads) <- colnames(dat)
colnames(pc1_loads) <- c("PC1", "PC2")
pc1_loads
```

```
##               PC1    PC2     NA     NA     NA
## Wind      -0.010 -0.076 -0.031  0.920  0.342
## Radiation  0.993 -0.116 -0.007  0.000  0.002
## CO         0.014  0.100  0.183 -0.138  0.650
## NO        -0.005 -0.013  0.130 -0.328  0.643
## NO2        0.024  0.150  0.955  0.102 -0.207
## O3         0.112  0.973 -0.170  0.063  0.000
## HC         0.002  0.024  0.085  0.110  0.062
```

Let's now use the `princomp()` function.

```
#using princomp
pc1_b <- princomp(dat, cor = F)
names(pc1_b)
```

```
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
```

```
summary(pc1_b)
```

```
## Importance of components:
##                            Comp.1     Comp.2     Comp.3      Comp.4      Comp.5
## Standard deviation     17.234083 5.25384279 3.34537279 1.569785488 1.117613433
## Proportion of Variance  0.872948 0.08112714 0.03289281 0.007242569 0.003671092
## Cumulative Proportion   0.872948 0.95407514 0.98696795 0.994210520 0.997881611
##                              Comp.6       Comp.7
## Standard deviation     0.718428856 0.4523547944
## Proportion of Variance 0.001516979 0.0006014096
## Cumulative Proportion  0.999398590 1.0000000000
```

```
#get loadings
pc1_b$loadings
```

```
## 
## Loadings:
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## Wind                            0.920  0.342         0.170
## Radiation -0.993  0.116                                   
## CO                      -0.183 -0.138  0.650 -0.564 -0.444
## NO                      -0.130 -0.328  0.643  0.498  0.463
## NO2              -0.150 -0.955  0.102 -0.207         0.105
## O3        -0.112 -0.973  0.170                            
## HC                              0.110         0.657 -0.738
## 
##                 Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings      1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var   0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var   0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

```
#check variance - again, not total variance
sum(round(pc1_b$sdev^2, 3))
```

```
## [1] 340.243
```

```
sum(apply(dat, 2, var))
```

```
## [1] 348.5407
```

NOW - let's scale variables and see what we get.

```r
#start with prcomp - use the option scale. = T
comp1 <- prcomp(dat, scale. = T)
round(summary.PCA.JDRS(comp1), 3)
```

```
##                          PC1   PC2   PC3   PC4   PC5   PC6   PC7
## Eigenvals (Variance)   2.337 1.386 1.204 0.727 0.653 0.537 0.156
## Proportion of Variance 0.334 0.198 0.172 0.104 0.093 0.077 0.022
## Cumulative Proportion  0.334 0.532 0.704 0.808 0.901 0.978 1.000
```

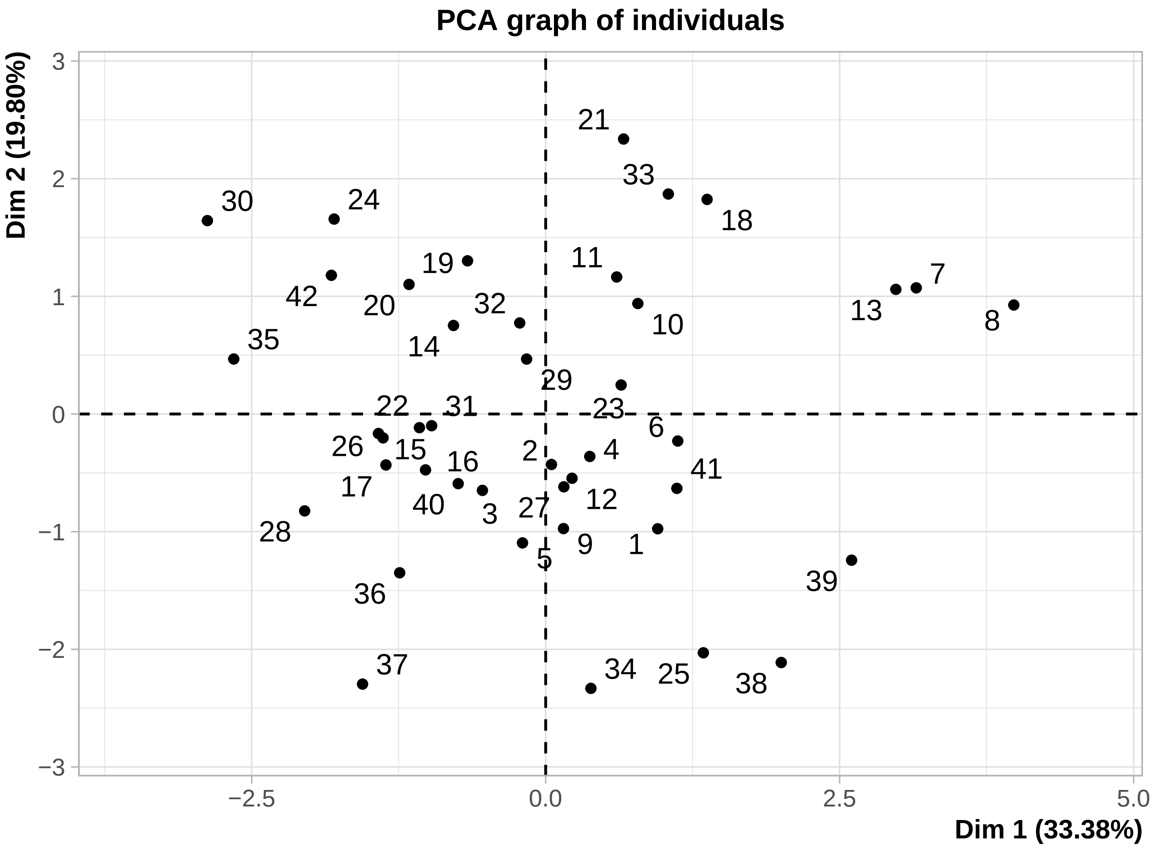
```r
#rotation - clearly 45 deg
round(comp1$rotation, 3)
```

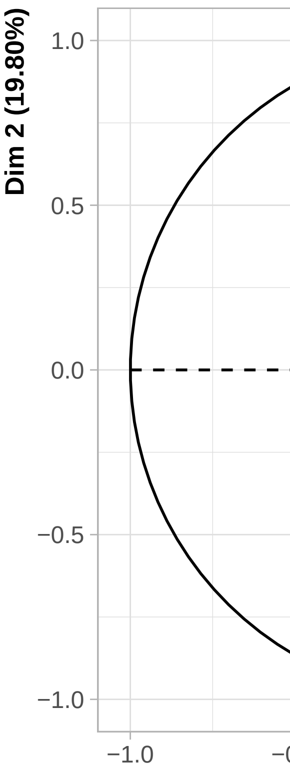
```
##              PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Wind      -0.237  0.278  0.643  0.173 -0.561  0.224 -0.241
## Radiation  0.206 -0.527  0.224  0.778  0.156  0.006 -0.011
## CO         0.551 -0.007 -0.114  0.005 -0.573  0.110  0.585
## NO         0.378  0.435 -0.407  0.291  0.057  0.450 -0.461
## NO2        0.498  0.200  0.197 -0.042 -0.050 -0.745 -0.338
## O3         0.325 -0.567  0.160 -0.508 -0.080  0.331 -0.417
## HC         0.319  0.308  0.541 -0.143  0.566  0.266  0.314
```

```r
#get total variance
sum(comp1$sdev^2)
```

```
## [1] 7
```

```r
#use PCA() function - default is to scale variables
pc1_a <- PCA(dat)
```





```
#NOTE - eigenvalues are now SAME
summary(pc1_a)
```

```
##
## Call:
## PCA(X = dat)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               2.337   1.386   1.204   0.727   0.653   0.537   0.156
## % of var.             33.383  19.800  17.201  10.387   9.335   7.667   2.227
## Cumulative % of var.  33.383  53.183  70.384  80.771  90.106  97.773 100.000
##
## Individuals (the 10 first)
##               Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## 1         |  3.024 |  0.953  0.925  0.099 | -0.975  1.634  0.104 | -0.427
## 2         |  2.319 |  0.049  0.002  0.000 | -0.429  0.316  0.034 | -0.293
## 3         |  2.543 | -0.538  0.295  0.045 | -0.649  0.724  0.065 | -0.552
## 4         |  2.565 |  0.375  0.143  0.021 | -0.361  0.224  0.020 |  2.003
## 5         |  1.602 | -0.197  0.040  0.015 | -1.096  2.062  0.468 | -0.449
## 6         |  1.865 |  1.123  1.286  0.363 | -0.230  0.091  0.015 |  1.354
## 7         |  4.160 |  3.151 10.119  0.574 |  1.072  1.975  0.066 |  1.622
## 8         |  4.483 |  3.981 16.148  0.789 |  0.926  1.474  0.043 | -0.379
## 9         |  1.371 |  0.152  0.023  0.012 | -0.974  1.629  0.505 |  0.337
## 10        |  1.774 |  0.784  0.626  0.195 |  0.939  1.515  0.280 |  0.985
##              ctr   cos2
## 1          0.360  0.020 |
## 2          0.169  0.016 |
## 3          0.602  0.047 |
## 4          7.935  0.610 |
## 5          0.398  0.079 |
## 6          3.627  0.527 |
## 7          5.204  0.152 |
## 8          0.284  0.007 |
## 9          0.224  0.060 |
## 10         1.920  0.309 |
##
## Variables
##              Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
## Wind      | -0.362  5.608  0.131 |  0.328  7.753  0.107 |  0.706 41.406  0.499
## Radiation |  0.314  4.226  0.099 | -0.620 27.732  0.384 |  0.246  5.039  0.061
## CO        |  0.842 30.369  0.710 | -0.008  0.005  0.000 | -0.125  1.291  0.016
## NO        |  0.577 14.259  0.333 |  0.512 18.894  0.262 | -0.447 16.573  0.200
## NO2       |  0.761 24.802  0.580 |  0.235  3.991  0.055 |  0.216  3.863  0.047
## O3        |  0.496 10.533  0.246 | -0.667 32.146  0.446 |  0.175  2.555  0.031
## HC        |  0.488 10.202  0.238 |  0.362  9.479  0.131 |  0.594 29.273  0.352
##
## Wind      |
## Radiation |
## CO        |
## NO        |
## NO2       |
## O3        |
## HC        |
```

```r
round(pc1_a$svd$V, 3)
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] -0.237  0.278  0.643  0.173  0.561
## [2,]  0.206 -0.527  0.224  0.778 -0.156
## [3,]  0.551 -0.007 -0.114  0.005  0.573
## [4,]  0.378  0.435 -0.407  0.291 -0.057
## [5,]  0.498  0.200  0.197 -0.042  0.050
## [6,]  0.325 -0.567  0.160 -0.508  0.080
## [7,]  0.319  0.308  0.541 -0.143 -0.566
```

```r
#use princomp function - use option cor = T
pc1_b <- princomp(dat, cor = T)
#now results are the same
pc1_b$sdev^2
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 2.3367826 1.3860007 1.2040659 0.7270865 0.6534765 0.5366888 0.1558989
```

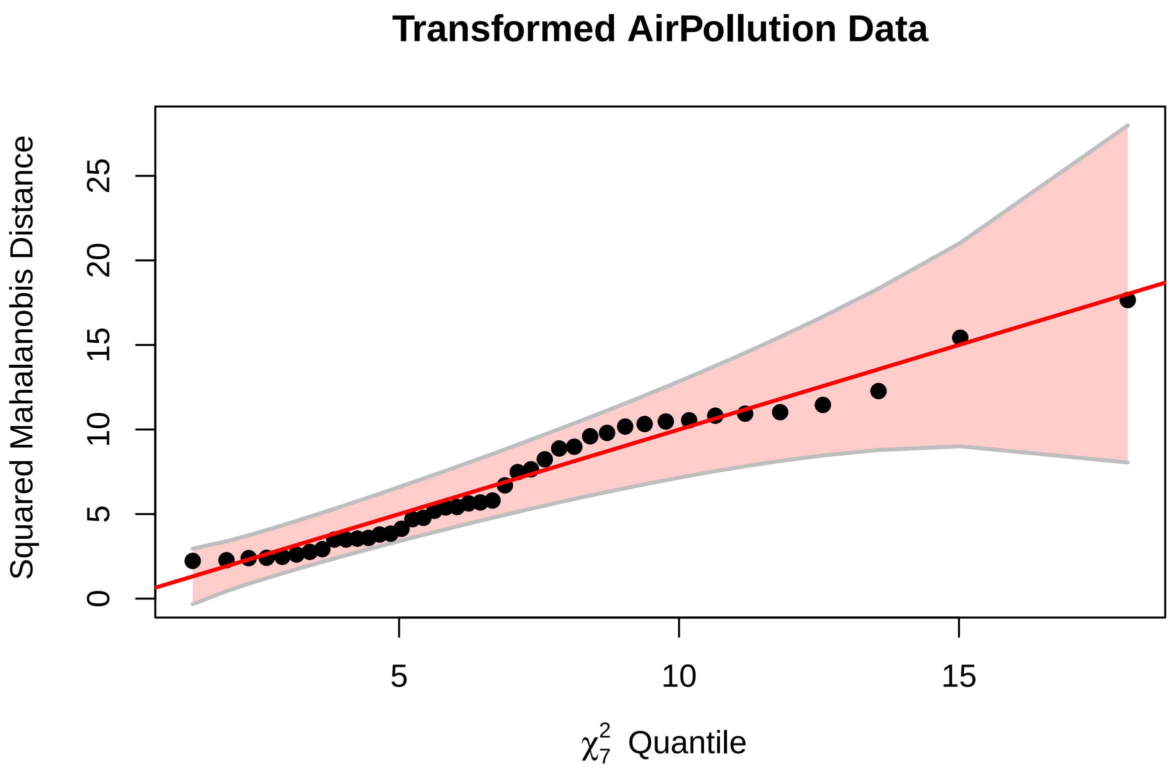
```r
#get loadings
pc1_b$loadings
```

```
##
## Loadings:
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## Wind       0.237  0.278  0.643  0.173  0.561  0.224  0.241
## Radiation -0.206 -0.527  0.224  0.778 -0.156
## CO        -0.551        -0.114         0.573  0.110 -0.585
## NO        -0.378  0.435 -0.407  0.291         0.450  0.461
## NO2       -0.498  0.200  0.197                -0.745  0.338
## O3        -0.325 -0.567  0.160 -0.508         0.331  0.417
## HC        -0.319  0.308  0.541 -0.143 -0.566  0.266 -0.314
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
## Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000
```

```r
str(dat)
```

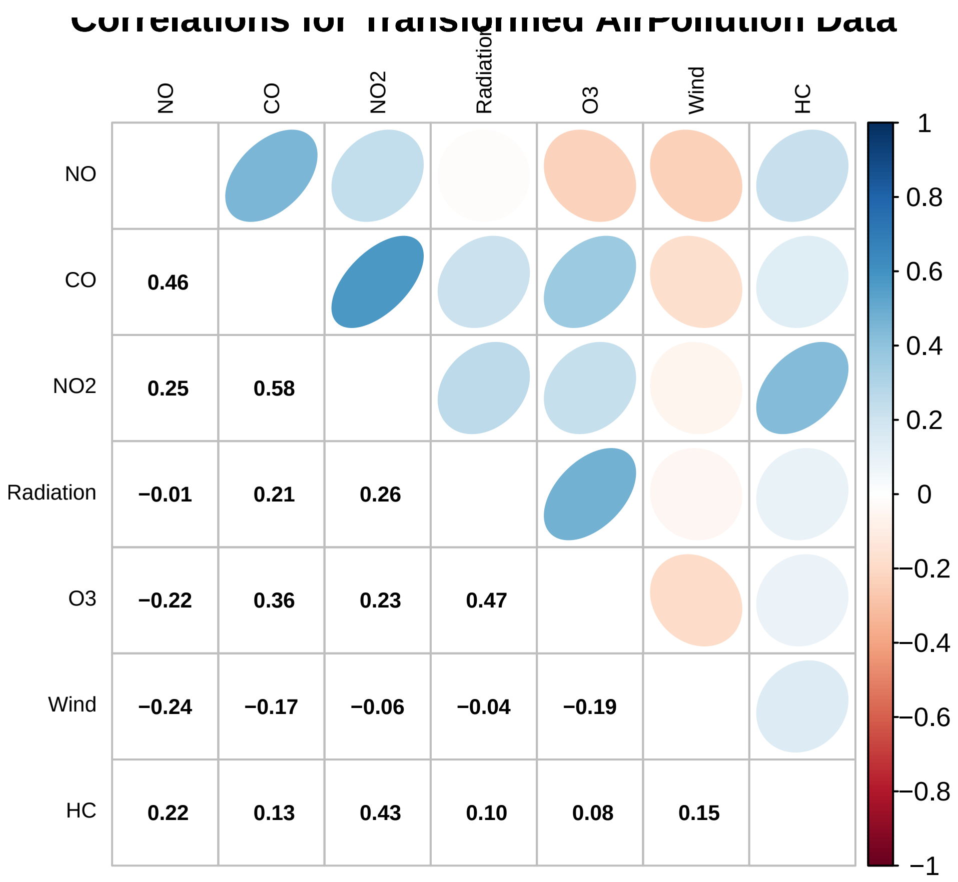
```
## 'data.frame':    42 obs. of  7 variables:
##  $ Wind     : num  8 7 7 10 6 8 9 5 7 8 ...
##  $ Radiation: num  98 107 103 88 91 90 84 72 82 64 ...
##  $ CO       : num  7 4 4 5 4 5 7 6 5 5 ...
##  $ NO       : num  2 3 3 2 2 2 4 4 1 2 ...
##  $ NO2      : num  12 9 5 8 8 12 12 21 11 13 ...
##  $ O3       : num  8 5 6 15 10 12 15 14 11 9 ...
##  $ HC       : num  2 3 3 4 3 4 5 4 3 4 ...
```

```r
dattrans <- log(dat)
dattrans <- dattrans[complete.cases(dattrans),]

#run the function
cqplot(dattrans, main = "Transformed AirPollution Data")
```



Because several variables display skewness and tail deviations from multivariate normality, a log transformation was applied to all variables. After transformation, the scatterplot matrix still shows approximately linear relationships among variables. The chi-square quantile plot for the transformed data appears sightly closer to linearity than that of the raw data, suggesting a little improvement in multivariate normality. The transformed data still do not perfectly follow a multivariate normal distribution.

```r
corrplot.mixed(cor(dattrans), lower.col = "black", upper = "ellipse",
               tl.col = "black", number.cex=.7, order = "hclust",
               tl.pos = "lt", tl.cex=.7,
               main="Correlations for Transformed AirPollution Data")
```

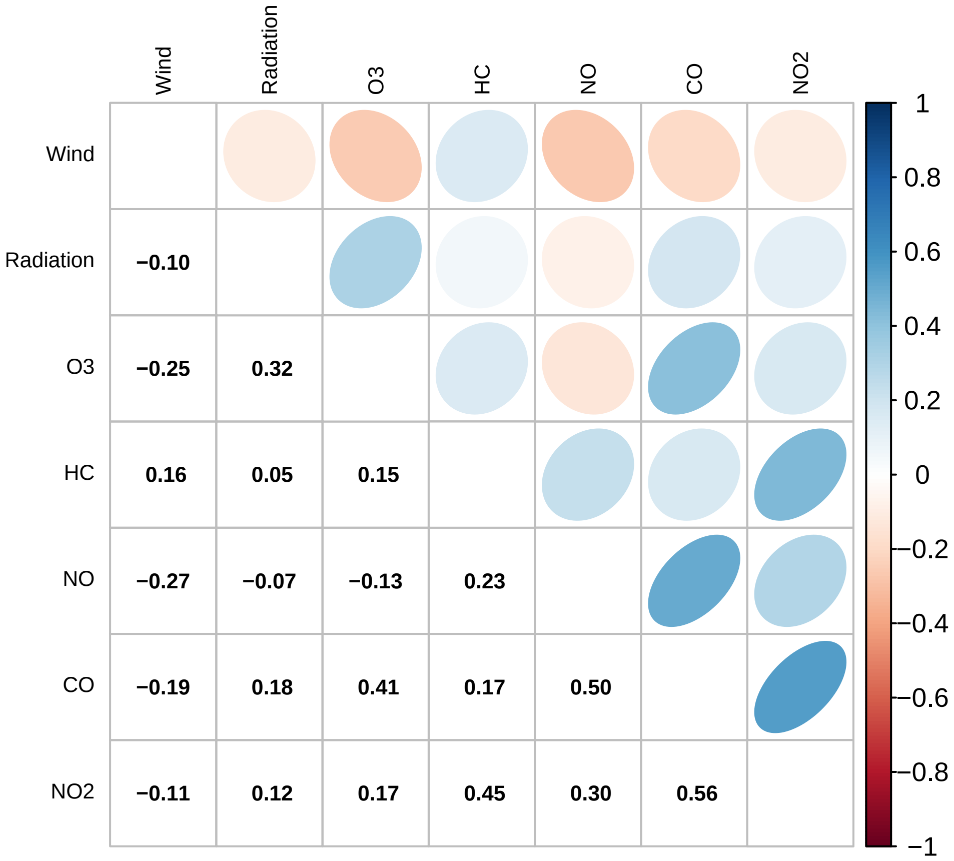


2) Examine the correlations among all of your variables. Include your results in table/graph form as you deem appropriate.

```
cor(dat)
```

```
##                 Wind   Radiation          CO          NO        NO2          O3
## Wind       1.0000000 -0.10144191 -0.1938032 -0.26954261 -0.1098249 -0.2535928
## Radiation -0.1014419  1.00000000  0.1827934 -0.07356907  0.1157320  0.3191237
## CO        -0.1938032  0.18279338  1.0000000  0.50215246  0.5565838  0.4109288
## NO        -0.2695426 -0.07356907  0.5021525  1.00000000  0.2968981 -0.1339521
## NO2       -0.1098249  0.11573199  0.5565838  0.29689814  1.0000000  0.1666422
## O3        -0.2535928  0.31912373  0.4109288 -0.13395214  0.1666422  1.0000000
## HC         0.1560979  0.05201044  0.1660323  0.23470432  0.4477678  0.1544506
##                   HC
## Wind      0.15609793
## Radiation 0.05201044
## CO        0.16603235
## NO        0.23470432
## NO2       0.44776780
## O3        0.15445056
## HC        1.00000000
```

```
corrplot.mixed(cor(dat), lower.col = "black", upper = "ellipse",
               tl.col = "black", number.cex = .7, order = "hclust",
               tl.pos = "lt", tl.cex = .7)
```



Based on the correlation matrix and correlation plots, the variables in the AirPollution dataset exhibit a pattern of moderate overall correlation with localized stronger associations. Several pollutant variables (e.g., CO, NO, NO2, O3, and HC) show moderate to relatively strong positive correlations with one another, while Wind tends to be negatively correlated with some pollutant measures. In contrast, Radiation displays weaker correlations with several of the pollutant variables.

Comment on how well you think PCA will work on your data.

The correlation structure suggests that the data contain shared variation that PCA can exploit, and at least one principal component is expected to capture a general "overall pollution" or "co-varying pollutant" axis. However, because correlations are not uniformly strong across all variables, PCA is unlikely to achieve very aggressive dimensionality reduction. Instead, multiple principal components will likely be required to explain a large proportion of the total variance, rather than only one or two components accounting for most of the variability.

In addition, provide a discussion of sample size relative to the number of variables in your dataset.

The dataset consists of 42 observations ($n = 42$) measured on 7 variables ($p = 7$), yielding a ratio of approximately $n/p \approx 6$. According to common empirical guidelines in multivariate analysis, PCA is generally considered acceptable when the sample size is on the order of 4p to 10p. The current sample size falls within this recommended range, suggesting that the estimation of principal components should be reasonably stable. Nevertheless, because the sample size does not reach the most conservative 10p criterion, interpretations of later components (such as PC3, PC4, and beyond) should be made with appropriate caution.

3) Perform Principal components analysis using the Correlation matrix (standardized variables). Think about how many principal components to retain. To make this decision look at:

• Total variance explained by a given number of principle components • The 'eigenvalue > 1' criteria • The 'scree plot elbow' method (turn in the scree plot) • Parallel Analysis: think about whether this is appropriate based on what you discover in question 1.

#####FIRST, use prcomp() on the untransformed data (scaled yes, transformed no)

```
#scale. = TRUE means run on the correlation matrix, i.e. standardize the variables.
pc1 <- prcomp(dat, scale. = TRUE)
```

```
pc1_trans <- prcomp(dattrans, scale. = T)

#Here are variances
round(summary.PCA.JDRS(pc1_trans), 2)
```

```
##                         PC1  PC2  PC3  PC4  PC5  PC6  PC7
## Eigenvals (Variance)   2.32 1.45 1.23 0.70 0.65 0.46 0.18
## Proportion of Variance 0.33 0.21 0.18 0.10 0.09 0.07 0.03
## Cumulative Proportion  0.33 0.54 0.71 0.81 0.91 0.97 1.00
```

```
#print results -
#Here are eigenvalues
round(summary.PCA.JDRS(pc1),2)
```

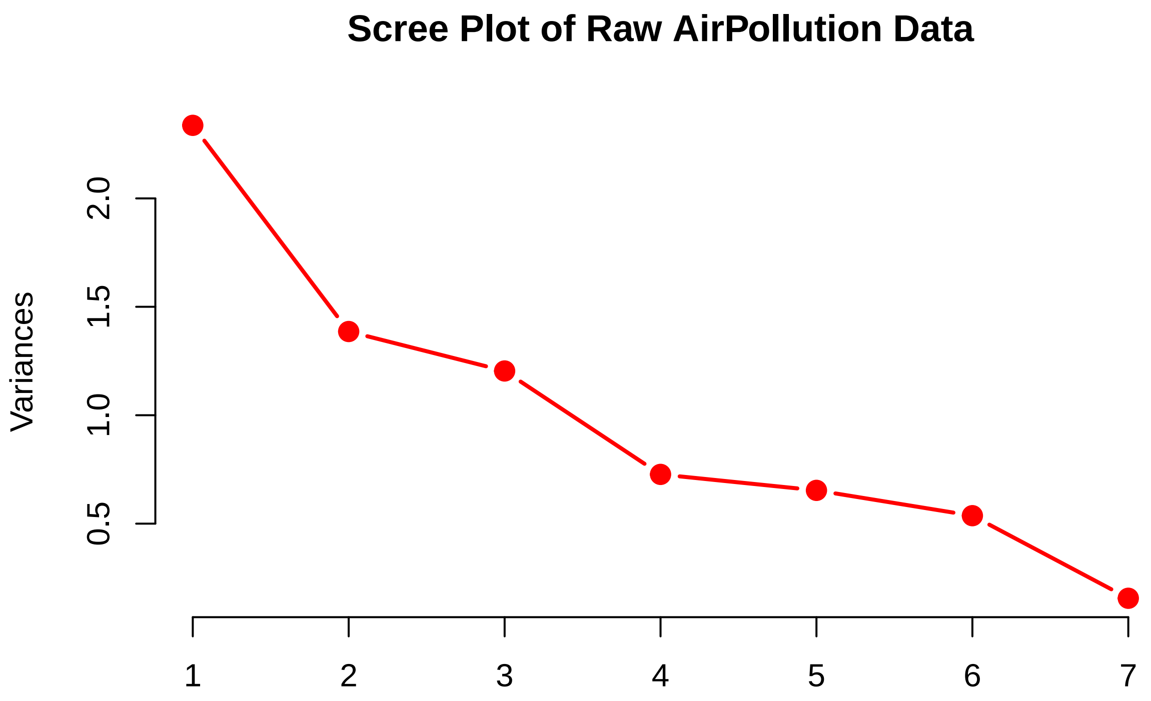
```
##                         PC1  PC2  PC3  PC4  PC5  PC6  PC7
## Eigenvals (Variance)   2.34 1.39 1.20 0.73 0.65 0.54 0.16
## Proportion of Variance 0.33 0.20 0.17 0.10 0.09 0.08 0.02
## Cumulative Proportion  0.33 0.53 0.70 0.81 0.90 0.98 1.00
```

```
#Get loadings
round(pc1$rotation,2)
```

```
##             PC1   PC2   PC3   PC4   PC5   PC6   PC7
## Wind      -0.24  0.28  0.64  0.17 -0.56  0.22 -0.24
## Radiation  0.21 -0.53  0.22  0.78  0.16  0.01 -0.01
## CO         0.55 -0.01 -0.11  0.01 -0.57  0.11  0.59
## NO         0.38  0.43 -0.41  0.29  0.06  0.45 -0.46
## NO2        0.50  0.20  0.20 -0.04 -0.05 -0.74 -0.34
## O3         0.32 -0.57  0.16 -0.51 -0.08  0.33 -0.42
## HC         0.32  0.31  0.54 -0.14  0.57  0.27  0.31
```

Make a screeplot

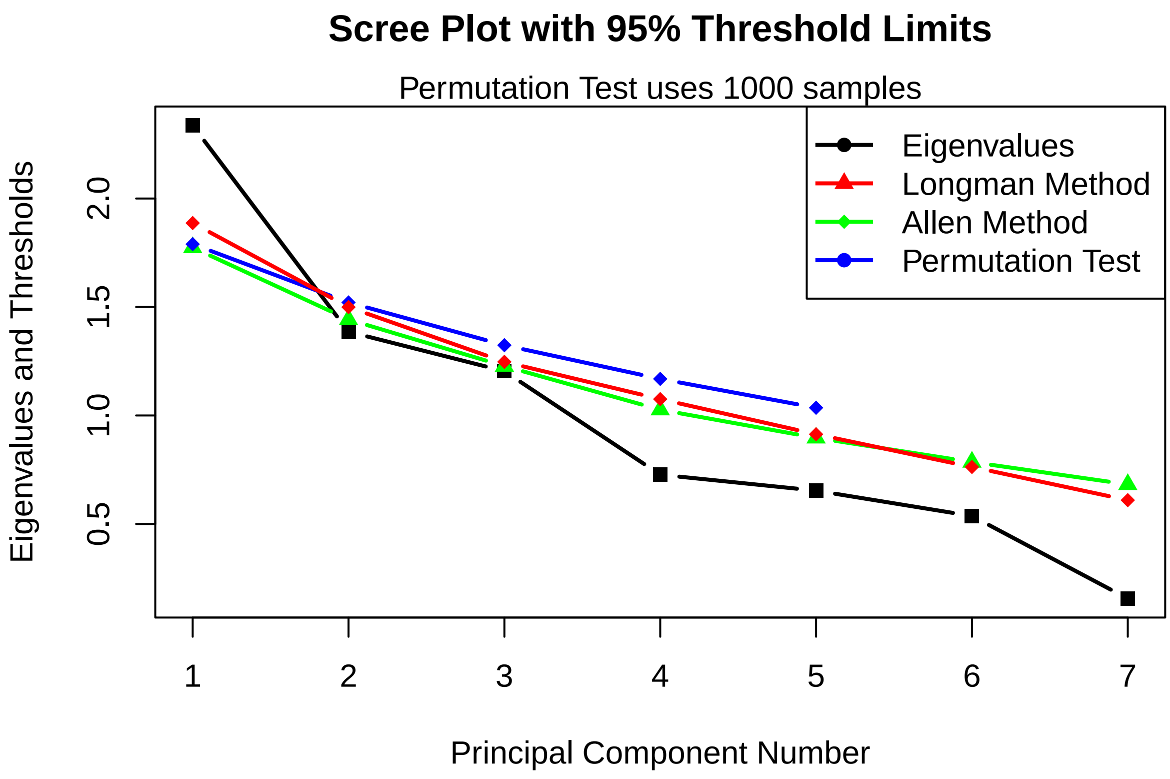
```
screeplot(pc1, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of Raw AirPollution Data")
```



Perform parallel analysis and permutation test analysis.

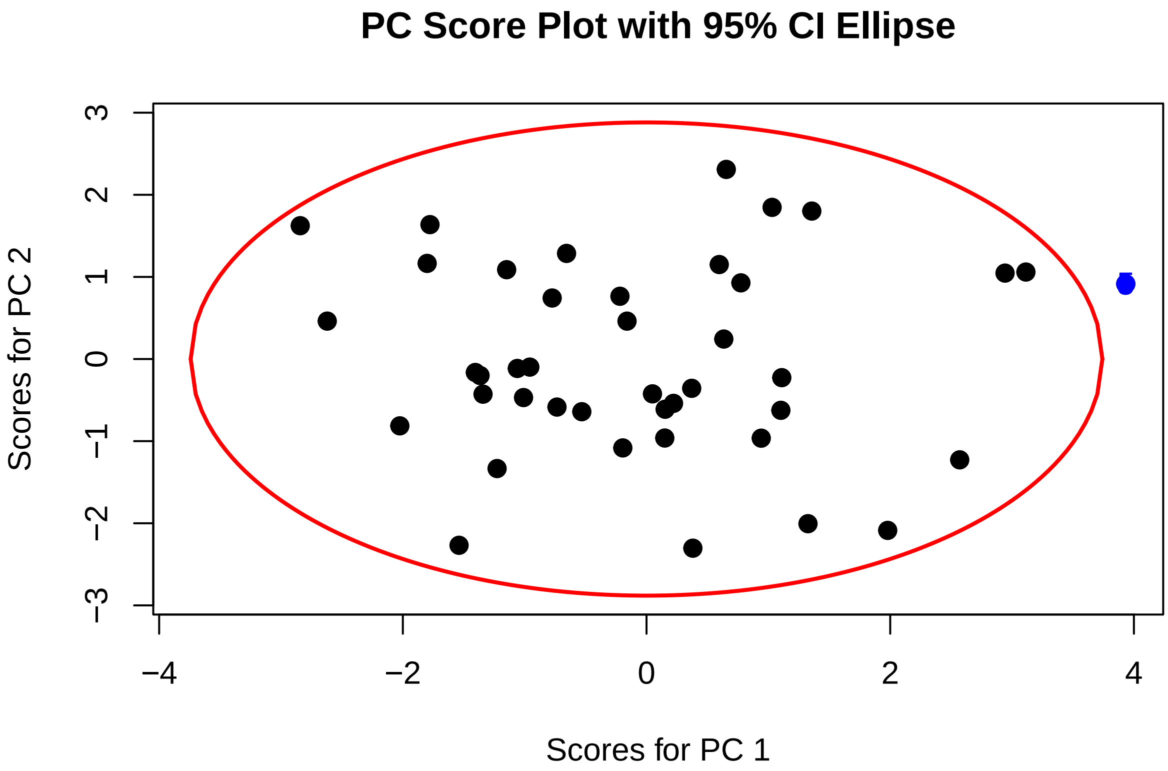
```
#get functions for score plots with boundary and PCA threshold analysis.
source("https://raw.githubusercontent.com/jreuning/sds363_code/refs/heads/main/pcaThreshold.r.txt")

#make the threshold analysis plot using the pcaThreshold function
pcaThreshold(dat)
```

```
##   pcompnum longman allen permutation
## 1        1    1.77  1.79        1.89
## 2        2    1.44  1.52        1.50
## 3        3    1.23  1.32        1.25
## 4        4    1.03  1.17        1.08
## 5        5    0.90  1.04        0.91
## 6        6    0.79    NA        0.76
## 7        7    0.68    NA        0.61
```

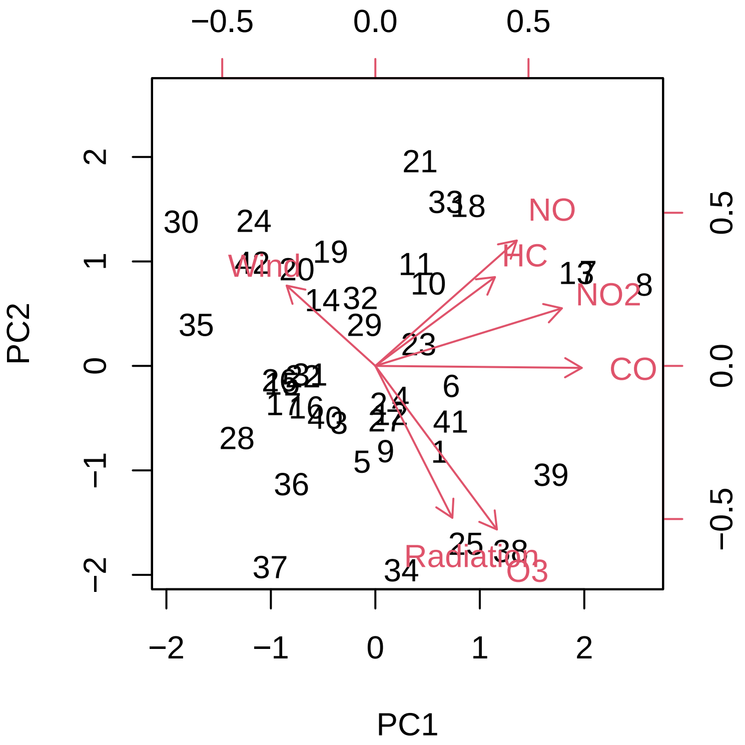


Make scoreplot with confidence ellipse as well as a biplot.

```r
#  c(1, 2) specifies to use components 1 and 2
#get function from online
source("https://raw.githubusercontent.com/jreuning/sds363_code/refs/heads/main/ciscoreplot.r.txt")

#run the function
ciscoreplot(pc1, c(1, 2), dat[, 1])
```



```
#make a biplot for first two components
biplot(pc1, choices = c(1, 2), pc.biplot = T)
```



First, based on the cumulative proportion of variance explained, the first two principal components account for approximately 53% of the total variance, while the first three components explain about 70%, and the first four components explain roughly 80%. Thus, using a typical heuristic threshold of 70–80% cumulative variance suggests retaining between three and four components.

Second, according to the eigenvalue greater than 1, the first three principal components have eigenvalues exceeding 1, whereas the fourth and subsequent components do not. This criterion therefore supports retaining three principal components.

Third, inspection of the scree plot reveals a clear elbow around the third component, after which the eigenvalues decrease more gradually. This visual criterion is also consistent with retaining three components.

Finally, parallel analysis indicates that only the first principal component clearly exceeds the corresponding threshold across methods, while the second and third components lie close to the threshold. This suggests that the first component is the most statistically robust, whereas later components primarily serve an exploratory and descriptive role. However, parallel analysis is known to be conservative and is most appropriate under approximate multivariate normality. As shown in Question 1, the AirPollution data do not strictly satisfy the multivariate normality assumption, even after transformation. Accordingly, the results of parallel analysis should be interpreted with caution.

Taking all criteria together, I ultimately chose to retain three principal components for subsequent analysis. The first component captures the dominant and most stable source of variation, while the second and third components represent secondary but still interpretable patterns in the data.

4) For principal components you decide to retain, examine the loadings (principalcomponents) and think about an interpretation for each retained component.

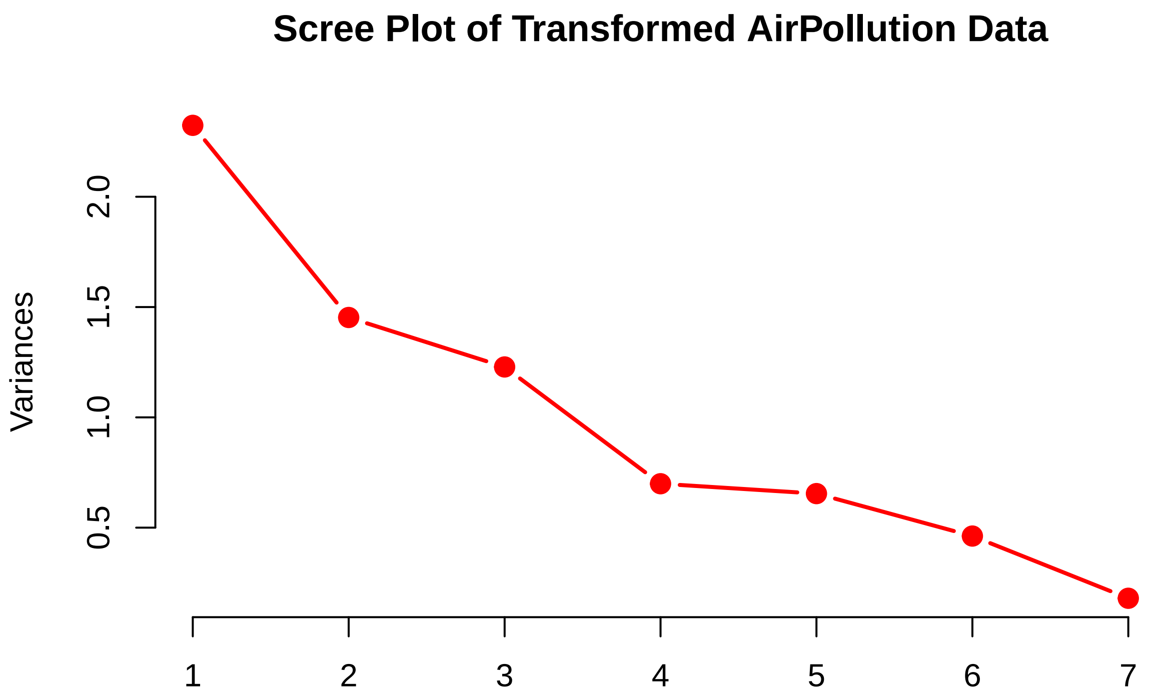
```
pc1_trans <- prcomp(dattrans, scale. = T)

#Here are variances
round(summary.PCA.JDRS(pc1_trans), 2)
```

```
##                           PC1  PC2  PC3  PC4  PC5  PC6  PC7
## Eigenvals (Variance)     2.32 1.45 1.23 0.70 0.65 0.46 0.18
## Proportion of Variance 0.33 0.21 0.18 0.10 0.09 0.07 0.03
## Cumulative Proportion  0.33 0.54 0.71 0.81 0.91 0.97 1.00
```

```
#Get loadings
round(pc1_trans$rotation, 2)
```

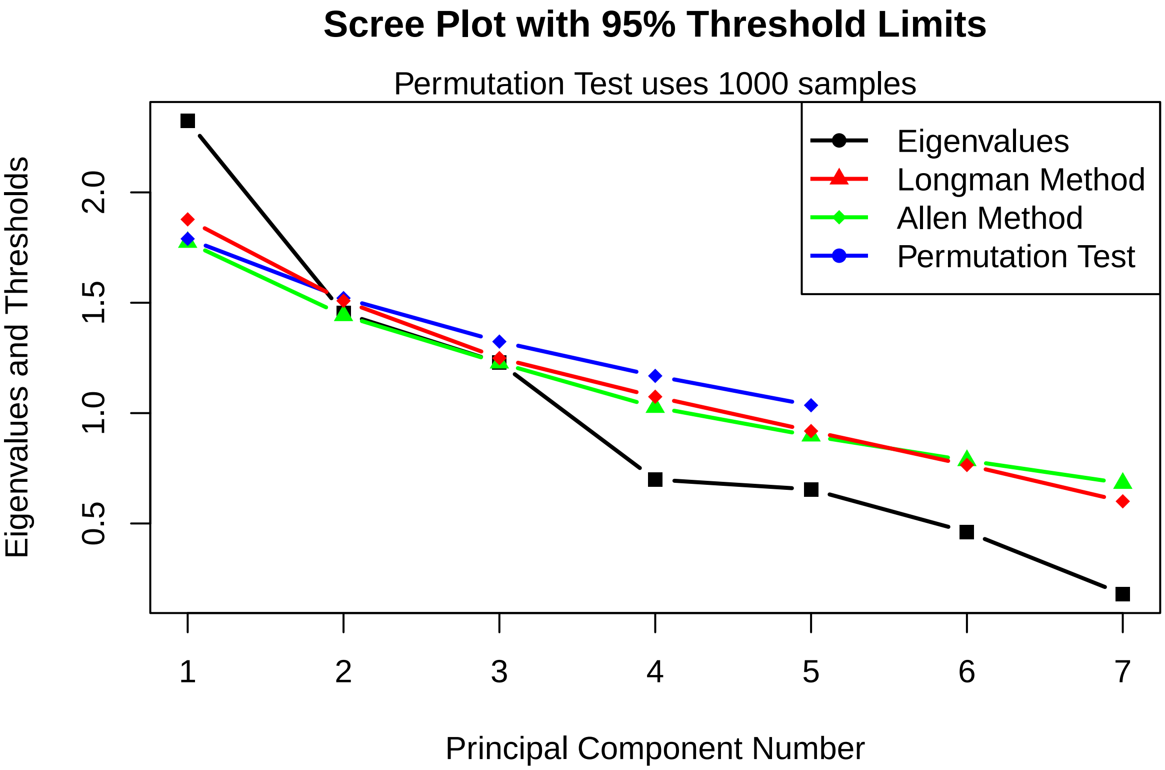
```
##             PC1   PC2   PC3   PC4   PC5   PC6   PC7
## Wind      -0.17  0.01  0.73  0.59  0.13 -0.22  0.18
## Radiation  0.34  0.47  0.06 -0.03  0.76  0.21 -0.18
## CO         0.53 -0.12 -0.17  0.51 -0.20 -0.22 -0.57
## NO         0.30 -0.59 -0.23  0.03  0.44 -0.31  0.46
## NO2        0.52 -0.11  0.23  0.08 -0.28  0.68  0.35
## O3         0.35  0.59 -0.07 -0.07 -0.28 -0.49  0.45
## HC         0.30 -0.22  0.57 -0.62 -0.06 -0.26 -0.27
```

```
#make a screeplot
screeplot(pc1_trans, type = "lines", col = "red", lwd = 2, pch = 19, cex = 1.2,
          main = "Scree Plot of Transformed AirPollution Data")
```

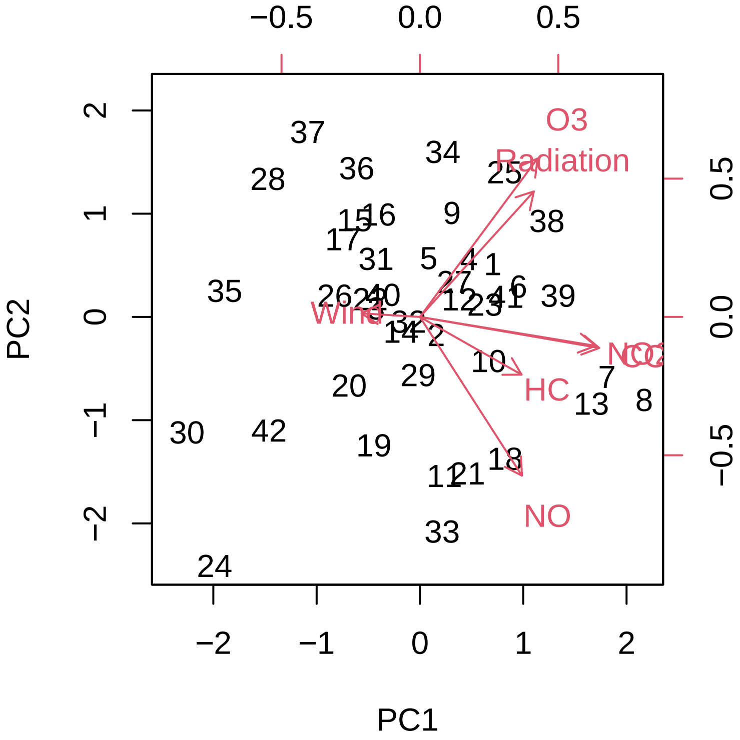


```
#make the threshold analysis plot
pcaThreshold(dattrans)
```

```
##   pcompnum longman allen permutation
## 1        1    1.77  1.79        1.88
## 2        2    1.44  1.52        1.51
## 3        3    1.23  1.32        1.25
## 4        4    1.03  1.17        1.07
## 5        5    0.90  1.04        0.92
## 6        6    0.79    NA        0.76
## 7        7    0.68    NA        0.60
```



```
#make a biplot for first two components
biplot(pc1_trans, choices = c(1, 2), pc.biplot = T)
```



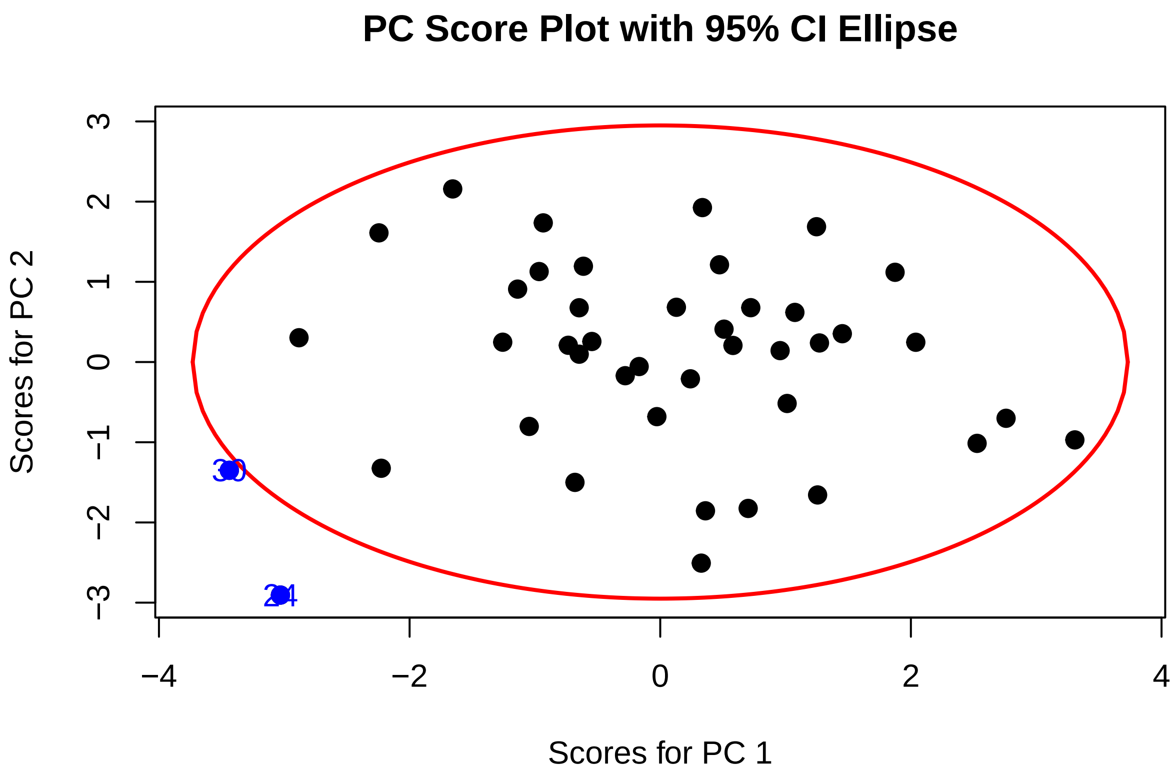
PC1 has relatively large positive loadings on several pollutant variables, including CO, NO2, O3, HC, and NO, as well as a positive loading on Radiation, while Wind shows a negative loading. This component represents a pattern in which multiple pollutants increase together, particularly under conditions of lower wind speed when dispersion is limited. Therefore, PC1 can be interpreted as an overall pollution burden or co-varying pollutant axis, capturing the dominant source of variation in air pollution levels.

PC2 is characterized by strong positive loadings on O3 and Radiation, accompanied by a negative loading on NO. This loading pattern reflects a contrast between ozone and nitrogen monoxide under varying radiation conditions, which is consistent with photochemical processes in the atmosphere. As such, PC2 can be interpreted as a photochemical or ozone–nitrogen oxide contrast axis, representing variation driven by solar radiation and related chemical reactions.

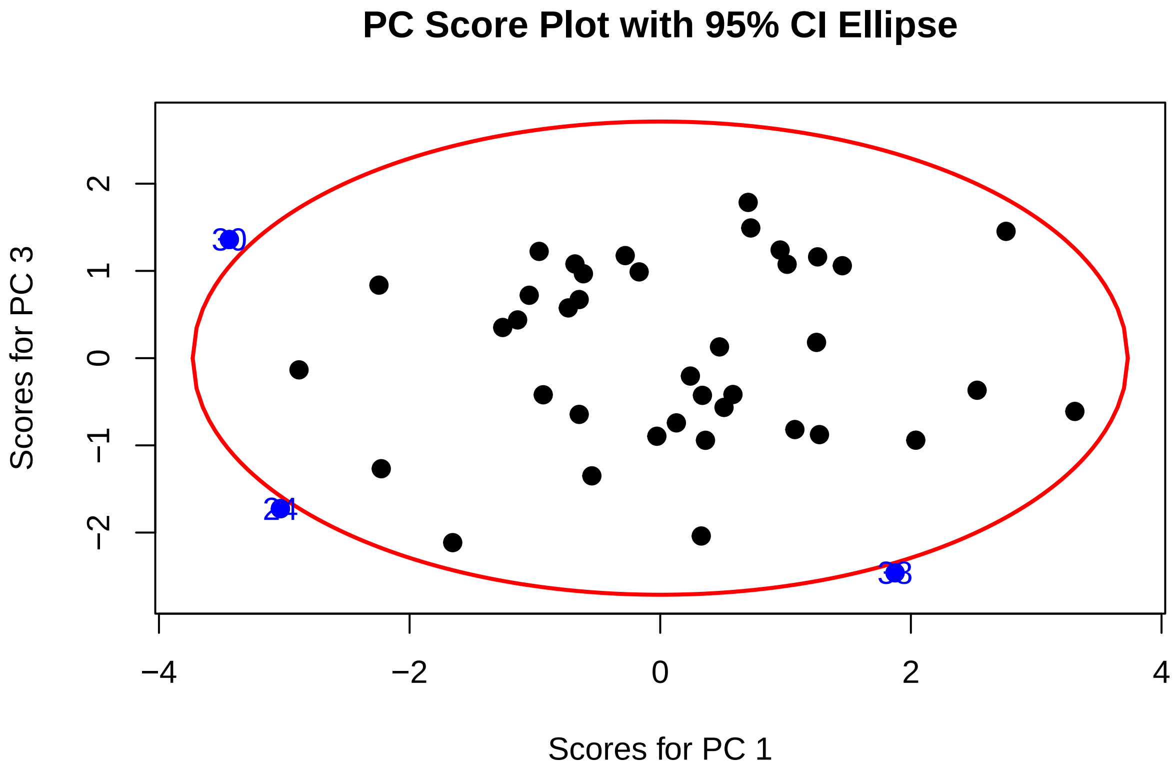
PC3 shows relatively large positive loadings on Wind and HC, with negative loadings on some other pollutants such as NO and CO. This component appears to capture a secondary pattern related to meteorological influences, particularly ventilation effects associated with wind speed. Because PC3 explains a smaller proportion of the total variance than PC1 and PC2, its interpretation should be made more cautiously. Nevertheless, it can be viewed as a meteorologically driven secondary modulation axis.

5) Make a score plot of the scores for at least one pair of component scores (one and two, one and three, two and three, etc). Discuss any trends/groupings you observe (probably, this will be 'none'). In addition, make a 95% CI ellipse for two of the retained components. Discuss whether it makes sense to use this as an outlier detection method and describe what you observe. If possible, include a biplot as well and discuss what you observe.

```r
Day <- 1:nrow(dat)

#make scoreplot with confidence ellipse :

#  c(1,2) specifies to use components 1 and 2
ciscoreplot(pc1_trans, c(1, 2), Day)
```

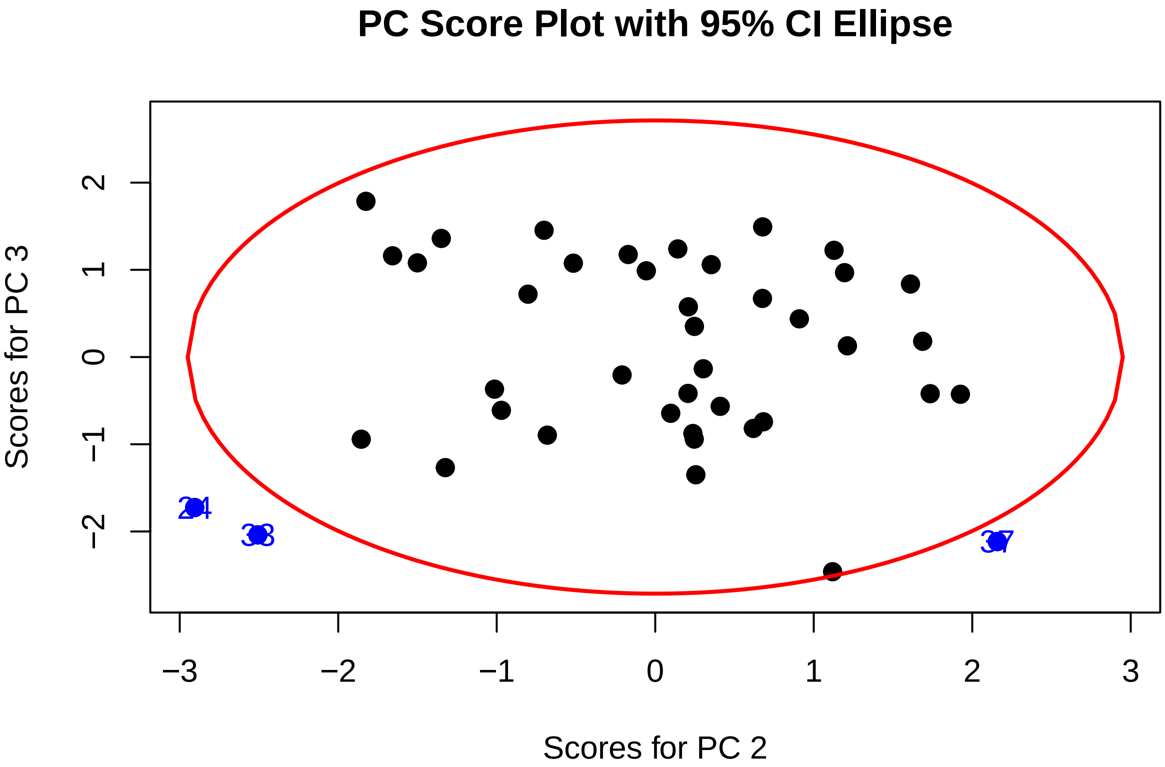


```
#  c(1,3) specifies to use components 1 and 3
#run the function
ciscoreplot(pc1_trans, c(1, 3), Day)
```



```
#  c(2,3) specifies to use components 2 and 3
#run the function
```

```
ciscoreplot(pc1_trans, c(2, 3), Day)
```



In the score plots of the retained components (PC1–PC2, PC1–PC3, and PC2–PC3), the observations form a largely continuous cloud with no clear evidence of distinct clusters or groupings. The 95% confidence ellipse in the PC1–PC2 plot flags Day 24 and Day 30 as clear outliers (lying outside the ellipse), indicating unusually extreme scores along the dominant variation axis. In the PC1–PC3 plot, Day 24 and Day 30 are again flagged, and Day 33 is additionally identified as an outlier. In the PC2–PC3 plot, Day 24 and Day 33 are again flagged, and Day 37 is additionally identified as an outlier, suggesting these days are unusual with respect to the secondary structure captured by PC2 and PC3.

Using the 95% confidence ellipse as an outlier detection method can be useful for exploratory quality control, but it is not a formal test unless the multivariate normality assumption holds reasonably well. As shown in Question 1, the AirPollution data do not strictly follow a multivariate normal distribution (even after log transformation), so these outliers should be interpreted as potentially unusual days rather than definitive statistical anomalies.

6) Write a paragraph summarizing your findings and your opinions about the effectiveness of using PCA on your data. Include evidence based on scatterplots of linearity in higher dimensional space, note any multivariate outliers in your score plot, interpretation of components, etc.

Overall, PCA provides a useful exploratory summary of the AirPollution dataset, but its dimensionality-reduction efficiency is moderate rather than extreme. The scatterplot matrices and correlation results indicate that most pairwise relationships are approximately linear and that several pollutant variables (e.g., CO, NO, NO2, O3, and HC) share moderate positive correlations, while Wind tends to be negatively associated with some pollutant measures. This structure supports the geometric assumptions behind PCA (linear projections capturing major variance directions), and it also implies that at least one dominant “co-varying pollution" axis should exist. Using PCA on the correlation matrix (standardized variables) yields a first component that captures a general pollution burden pattern (multiple pollutants loading in the same direction), with subsequent components reflecting secondary contrast patterns (e.g., a photochemical/radiation–ozone contrast versus nitrogen oxides and other meteorological modulation). However, because correlations are not uniformly high across all variables, PCA does not compress the data into only one or two components without substantial information loss; several components are needed to reach typical cumulative variance thresholds. Score plots with 95% confidence ellipses show no clear clustering of days (consistent with continuous environmental variation), but they do flag a small number of potentially unusual observations—most notably Day 24 and Day 30 in the PC1–PC2 space and Day 33 in the PC2–PC3 space—suggesting days with relatively extreme combinations of pollutant and meteorological conditions. Finally, because the multivariate normality diagnostics in Question 1 suggest departures from strict multivariate normality (even after transformation), the confidence-ellipse "outlier" results should be interpreted as exploratory quality-control signals rather than formal anomaly tests. Taken together, PCA is effective here as a descriptive tool for summarizing dominant patterns and identifying potentially extreme days, but conclusions about later components and outliers should be made cautiously and supported by the underlying variables.

Note: Chatgpt is used for grammar correction.