Final project: Step 2

Danyu Zhang, Limingrui Wan, Daniel Alonso

January 29th, 2021

Contents

Importing libraries

```
library(dplyr)
library(ggplot2)
library(reshape2)
library(PerformanceAnalytics)
library(gridExtra)
library(stringr)
library(foreach)
library(MASS)
library(andrews)
library(mice)
library(corrplot)
library(plotrix)
library(corpcor)
library(ggpubr)
library(ca)
library(tidyverse)
library(corpcor)
library(RSpectra)
library(factoextra)
library(cluster)
library(mclust)
library(smacof)
```

Cluster Analysis

Pre-processing Data

We define colors for plots

```
color_1 <- "deepskyblue2"
color_2 <- "seagreen2"
color_3 <- "orange2"
color_4 <- "darkorchid4"
color_5 <- "firebrick2"
color_6 <- 'red'
```

As we stated in *step 1*, there are some variables as they are redundant transformations of other columns. For different cases we may need to use standardized data and cases where the model only work with quantitative variables. we need to build a few subsets. And we need to impute the missing values.

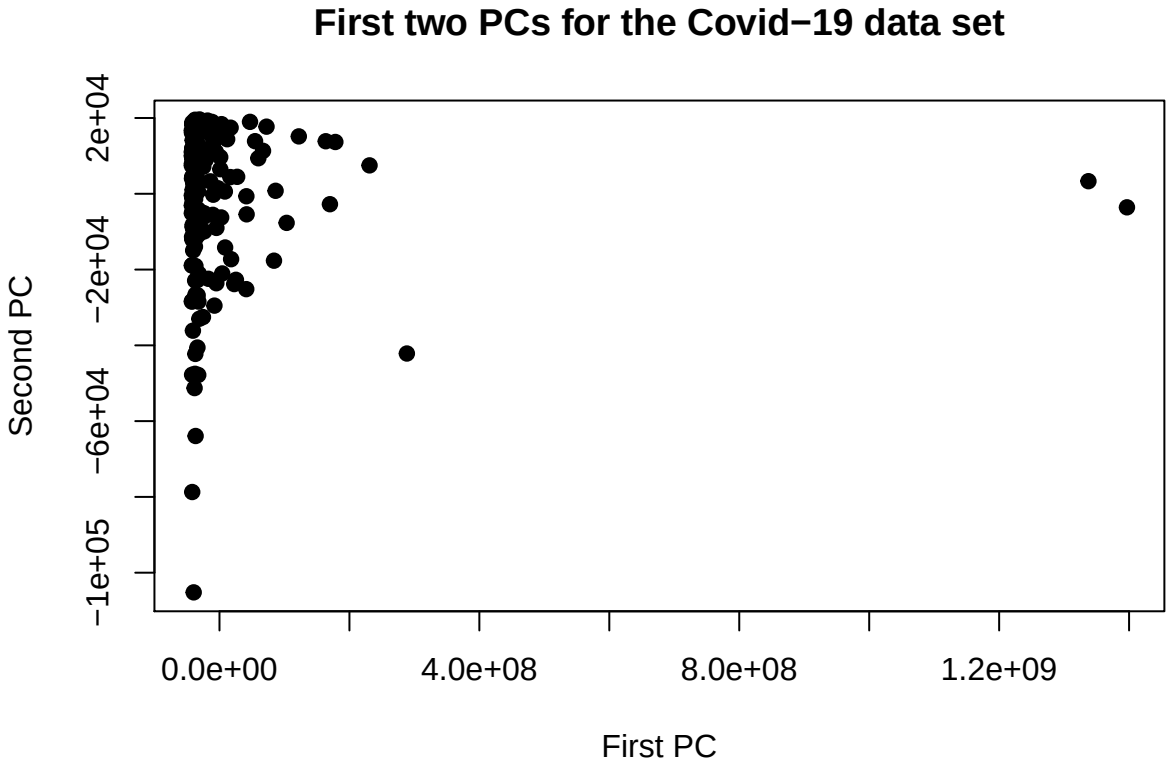
```
data2 <- read.csv('./data/data_imp.csv', header=TRUE)
data <- data2[,2:length(names(data2))]
data$continent=as.factor(data$continent)
data$development=as.factor(data$development)
data_cate <- subset(data, select = c(continent,development,location))
data_quan <- subset(data, select = -c(continent,development,location))
```

PCA analysis

computing PCAs

To visualize the results, we need to obtain the first two PCAs.

```
#> Estimating optimal shrinkage intensity lambda.var (variance vector): 0.3941
#>
#> Estimating optimal shrinkage intensity lambda (correlation matrix): 0.0367
```



We can't tell how many groups from this picture, then we need to find that with multiple methods.

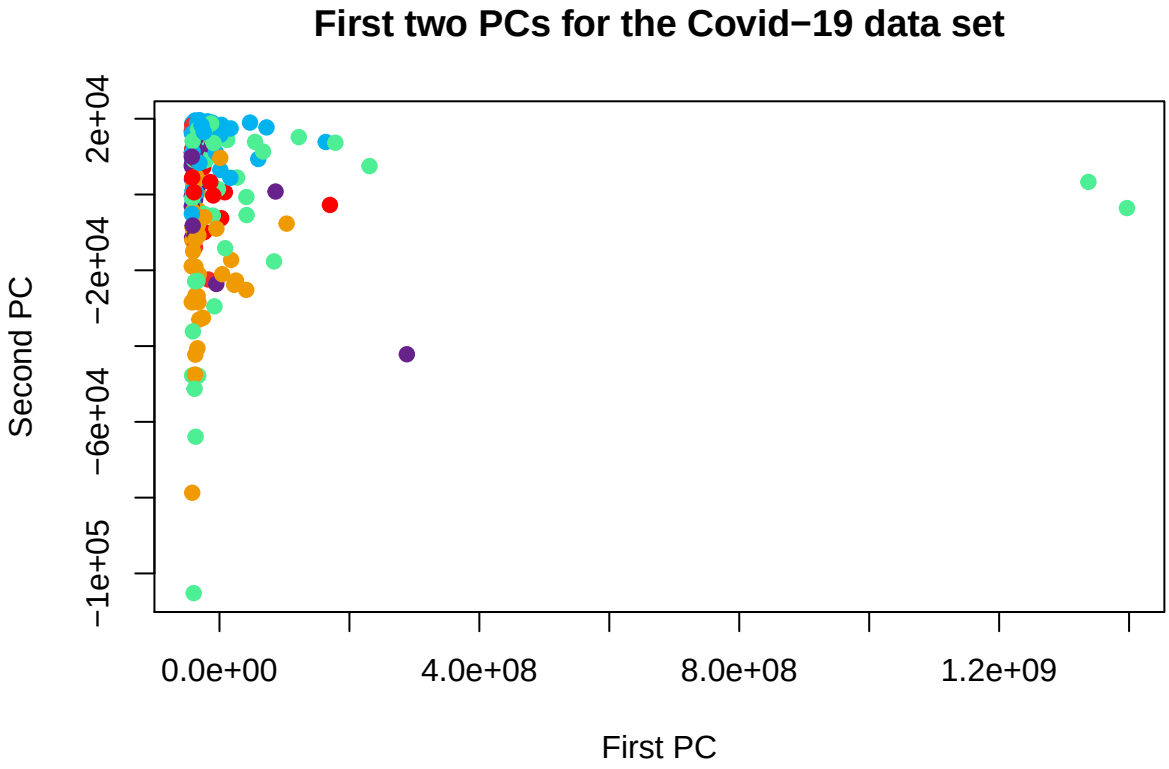
Partitional clustering

Partition of dataset

We firstly check how is the data grouped by the categorical variables.

Continent

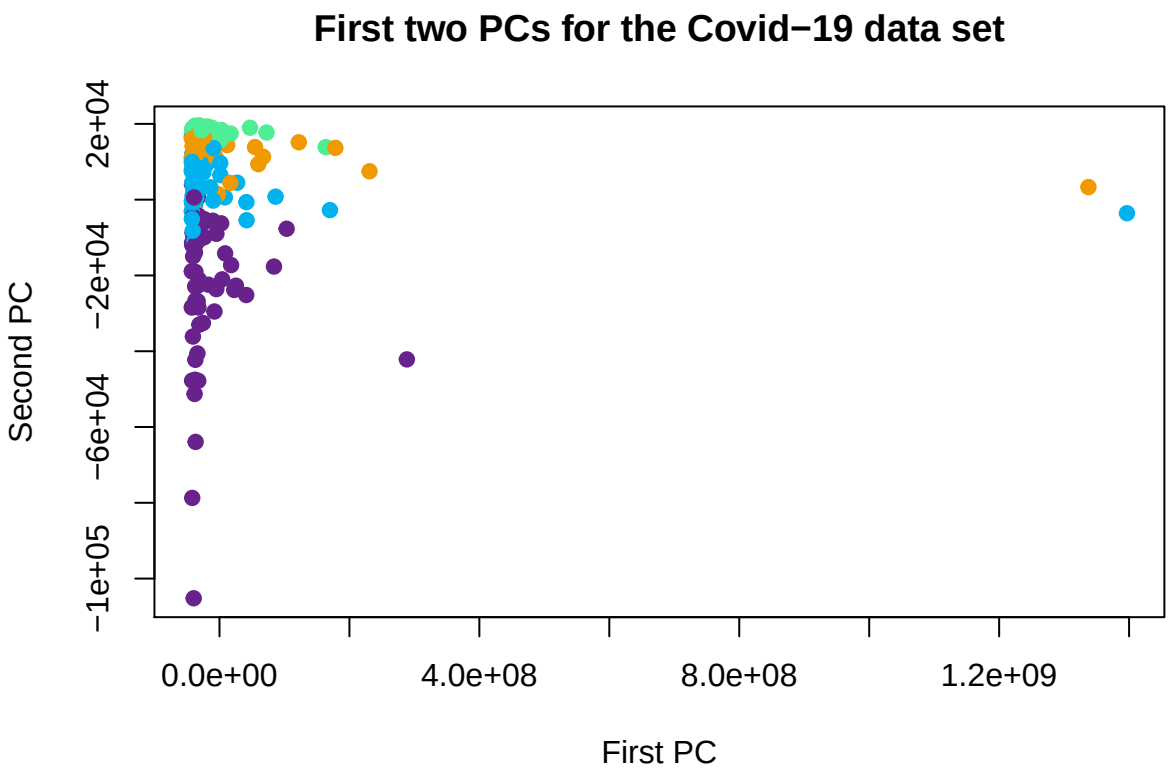
```
#>
#>        Africa          Asia        Europe North America       Oceania
#>            53            45            42            23             6
#> South America
#>            12
```



No sign of groups, i.t. we can't get information by knowing the location of a country.

Development

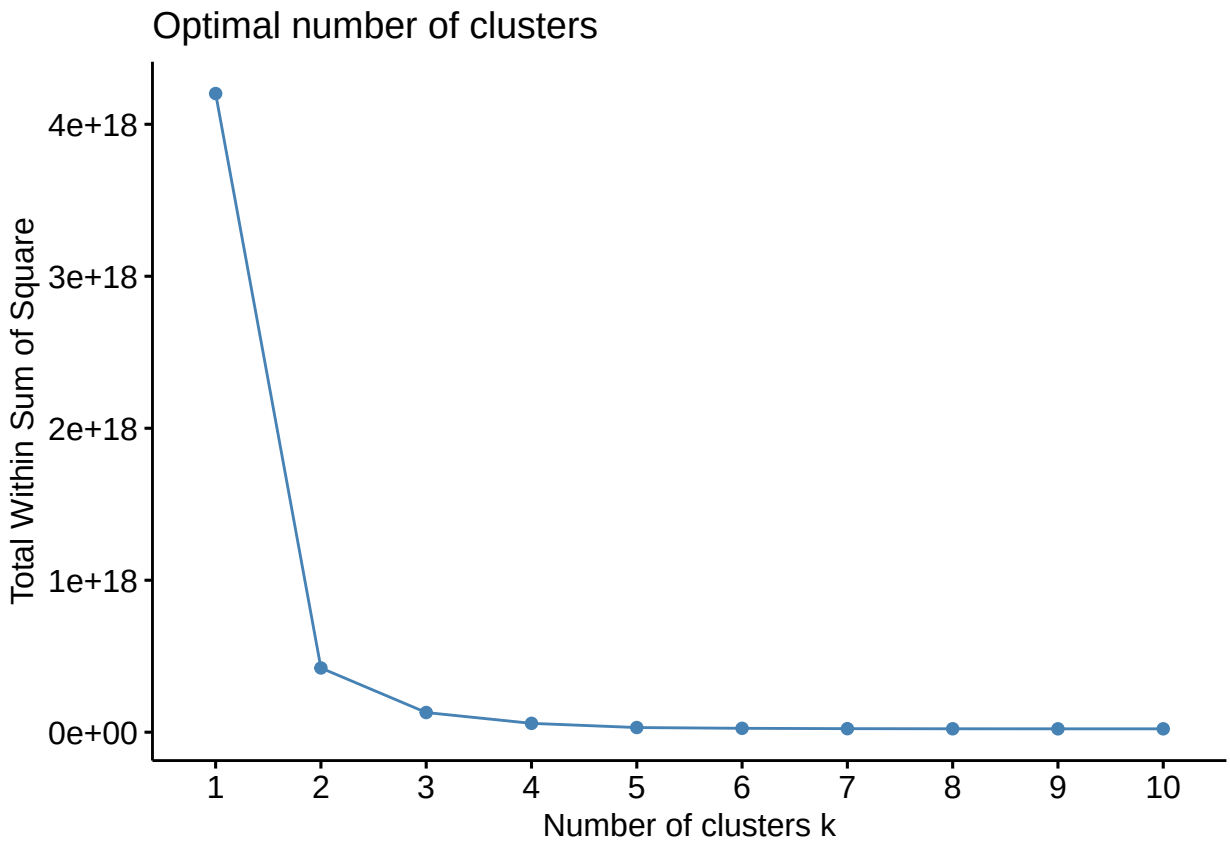
```
#>
#>      high       low    medium very high
#>        49        38        36        58
```



In this plot, groups are not separated well, the borders are not clear.

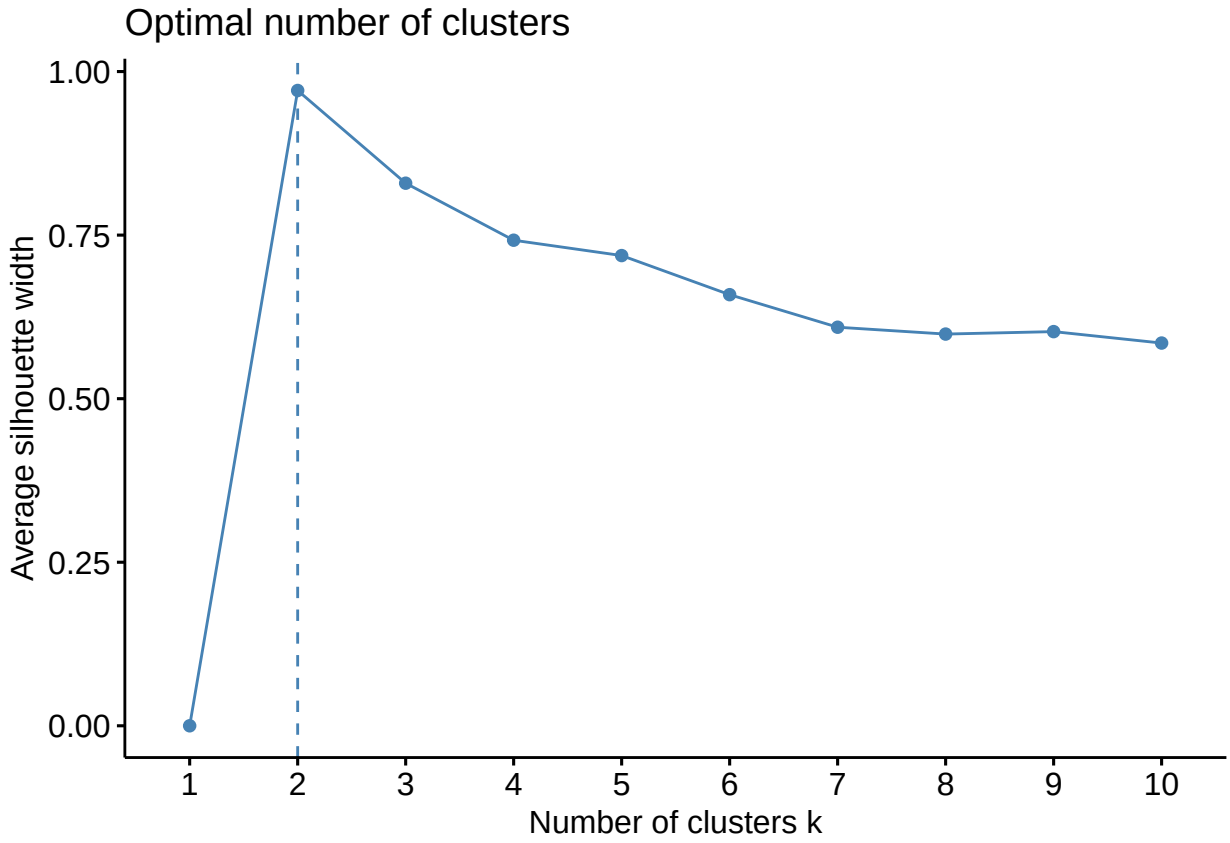
Select k

WSS



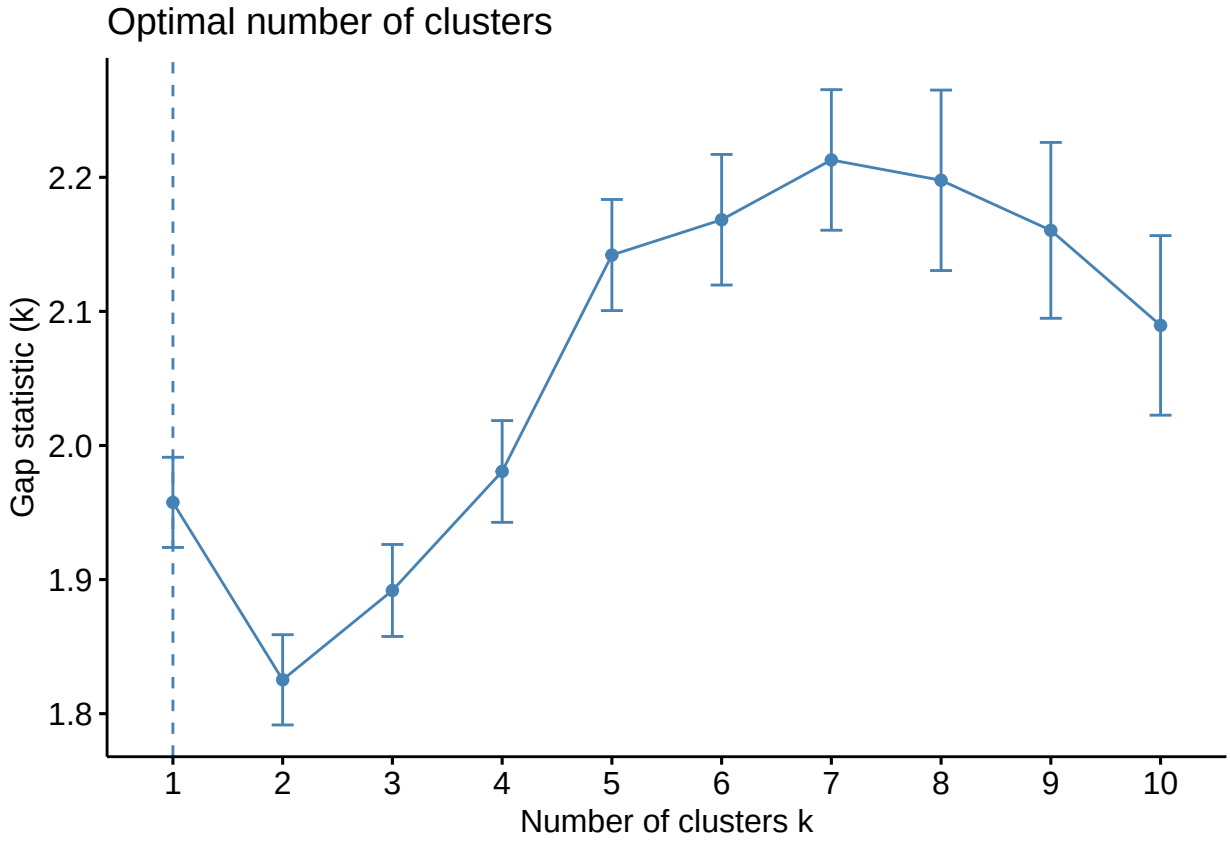
There is no optimal solution from WSS

Silhoutte



It sugguest us to set k into 2.

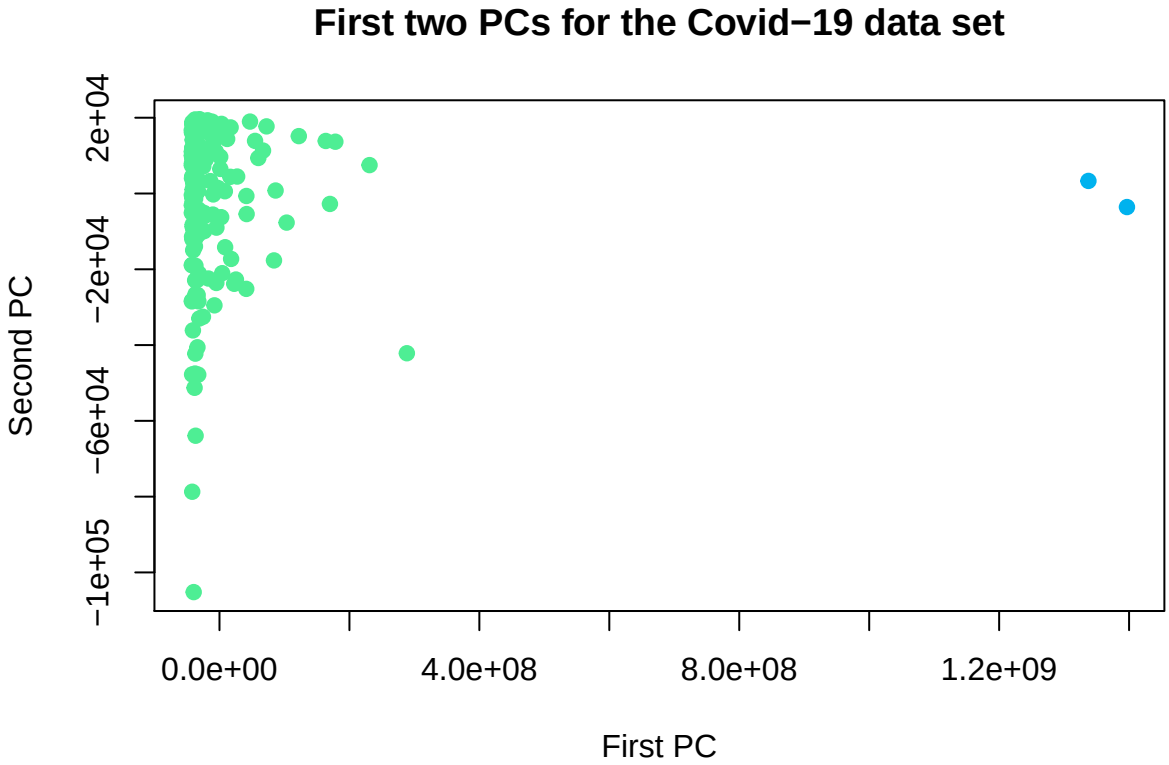
Gap Statistic



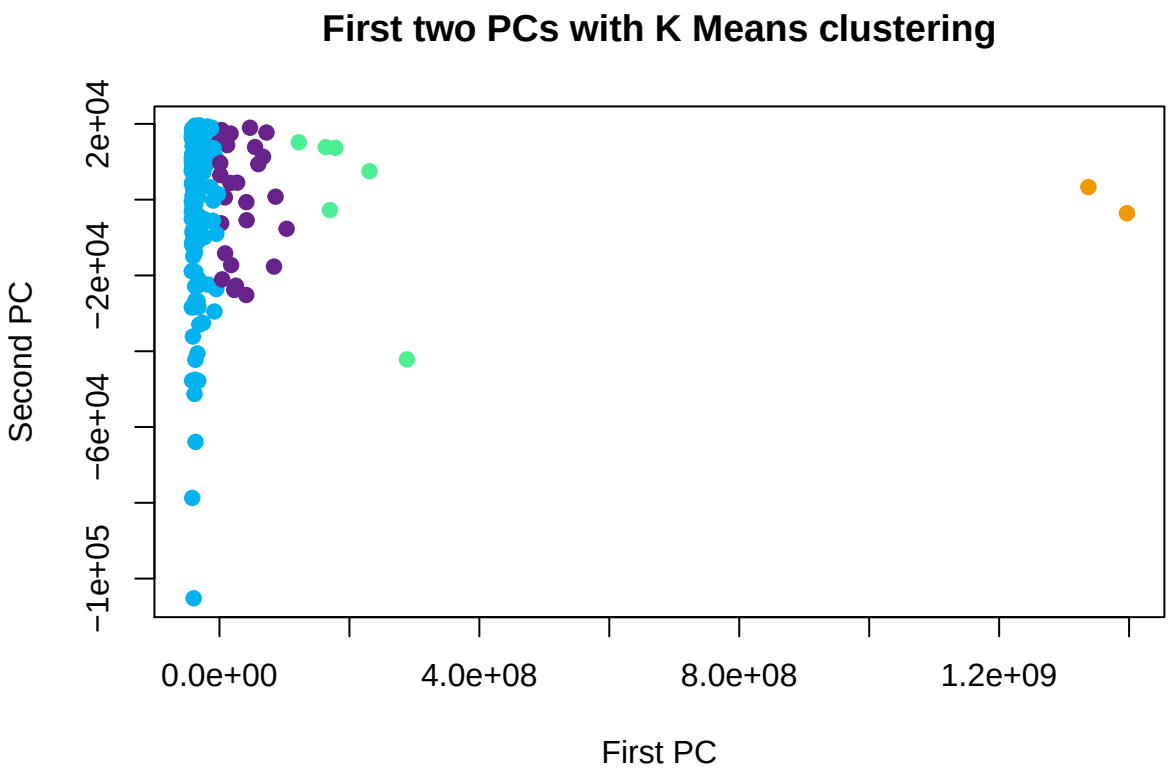
The result is 1, but we can't set the number of cluster to be 1, otherwise, it makes no sense.

The K-means algorithm

Notice that in our data, there are 3 categorical variables: but one of them is the names;one of them is the continent, which is irrelevant;one is development, but it's simple determined by numerical variable *develop*. so we only choose the rest of variables which all are quantitative. and we have only 181, not need to apply **CLARA**.



we can try to increase k=4 because we have a categorical variable *development*, we can check the model with it.



We now have a better clustering result.

K-means Analysis

```
table(kmeans_1$cluster)
#>
#>   1   2
#>   2 179
```

```
#>                                [,1]      [,2]
#> total_cases_per_million    8.367e+03 1.001e+04
#> new_cases_per_million      1.308e+02 5.319e+01
#> total_deaths_per_million   1.365e+02 2.629e+02
#> stringency_index           5.315e+01 5.918e+01
#> population                 9.910e+06 2.348e+08
#> population_density         2.065e+02 3.228e+02
#> median_age                 3.016e+01 2.837e+01
#> aged_65_older              8.590e+00 6.938e+00
#> gdp_per_capita             1.878e+04 1.557e+04
#> extreme_poverty            1.204e+01 1.102e+01
#> cardiovasc_death_rate      2.659e+02 2.623e+02
#> diabetes_prevalence        7.972e+00 7.395e+00
#> hospital_beds_per_thousand 2.695e+00 1.352e+00
#> life_expectancy            7.259e+01 7.017e+01
#> human_development_index    7.061e-01 6.798e-01
```

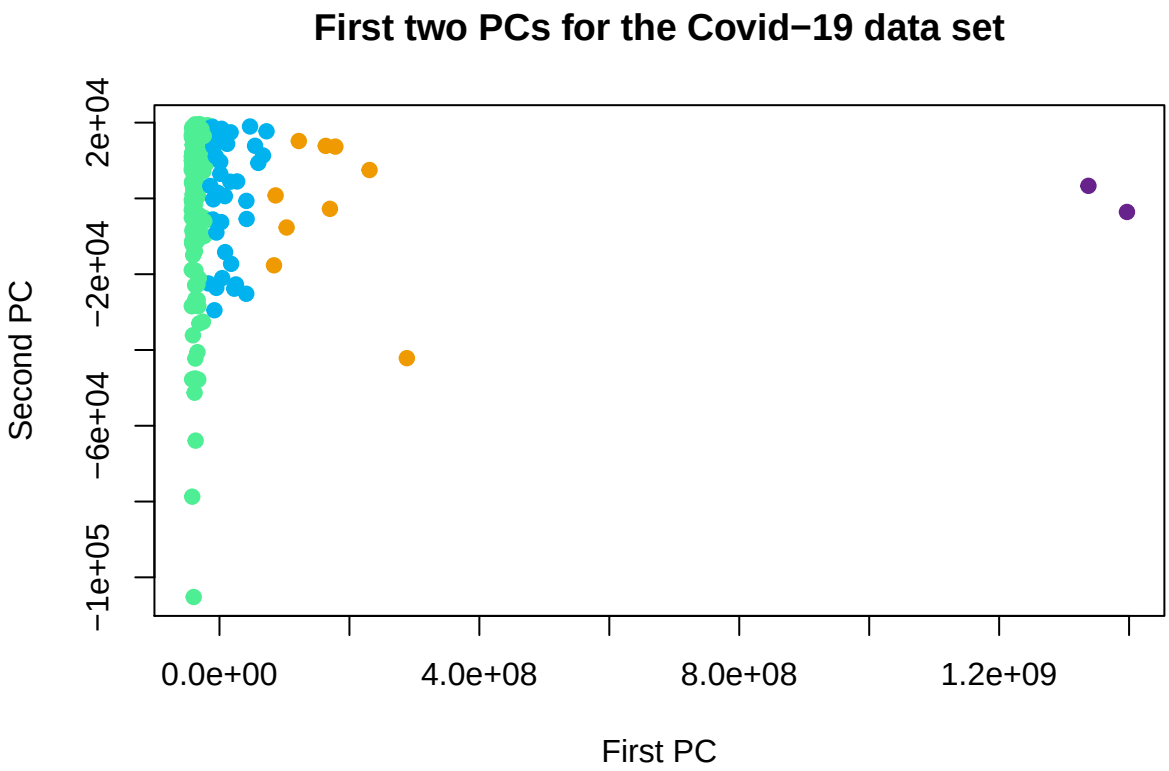
when k = 2, we can see obvious difference between two groups.

After we tuning K into 4, it has a more interesting result, we can also characterize them with some features: from 1 to 4 means from lowest(fewest) to highest(most).

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	3	4	4	3

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster3	3	2	4	2	2	2
cluster4	4	4	2	3	3	4

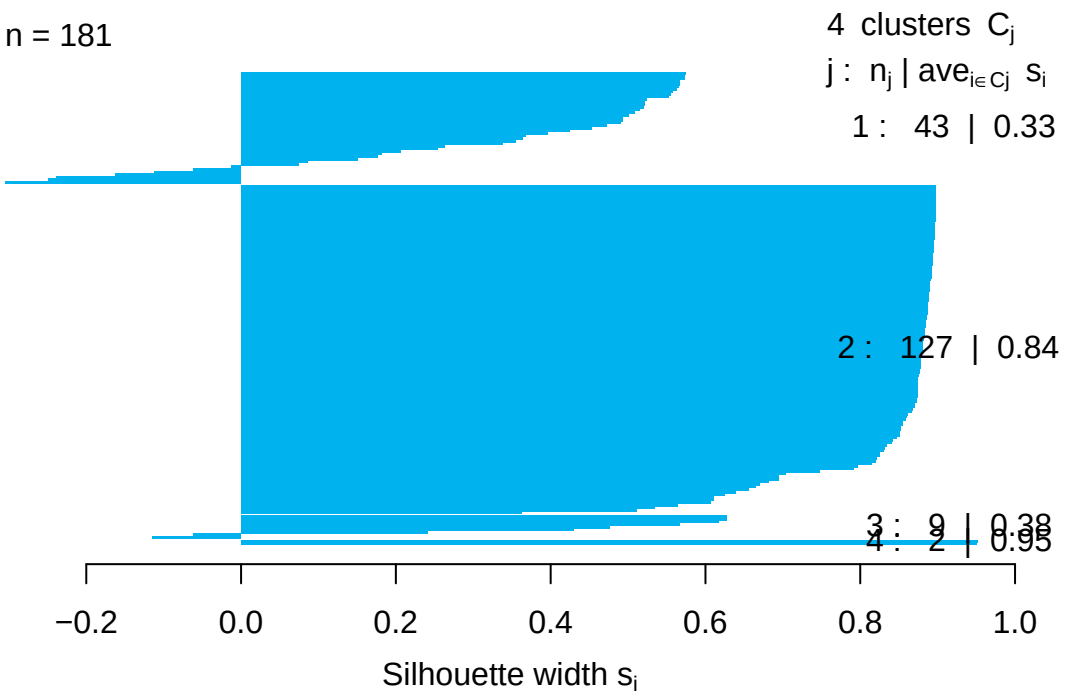
PAM



let's check the mean vector of the results of PAM.

we can also charactsize the clusters as following table: from 1 to 4 means from lowest(fewest) to highest(most).

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	2	3	3	4
cluster3	3	4	3	4	4	3
cluster4	4	2	4	2	2	2



Average silhouette width : 0.7

Here is the plot of silhouette.

Hierarchical clustering

There are multiple choice in this section, we will only accept the models with reasonable clusters. i.t. not too few or too many observations in one cluster.

Agglomerative algorithms

Single linkage

```
man_dist <- daisy(data_quan,metric="manhattan",stand=FALSE)
single = hclust(man_dist,method="single")
cl_single = cutree(single,4)
table(cl_single)
#> cl_single
#>   1   2   3   4
#> 178   1   1   1
```

Single method is an obvious wrong choice.

Complete linkage

```
complete = hclust(man_dist,method="complete")
cl_complete<- cutree(complete,4)
table(cl_complete)
#> cl_complete
#>   1   2   3   4
#> 170   7   2   2
```

Still terrible, only a little bit better.

Average linkage

```
average<- hclust(man_dist,method="average")
cl_average <- cutree(average,4)
table(cl_average)
#> cl_average
#>   1   2   3   4
#> 162  12   5   2
```

Almost same as the previous one, 165 observations in cluster 1, and 16 in others, not a good result.

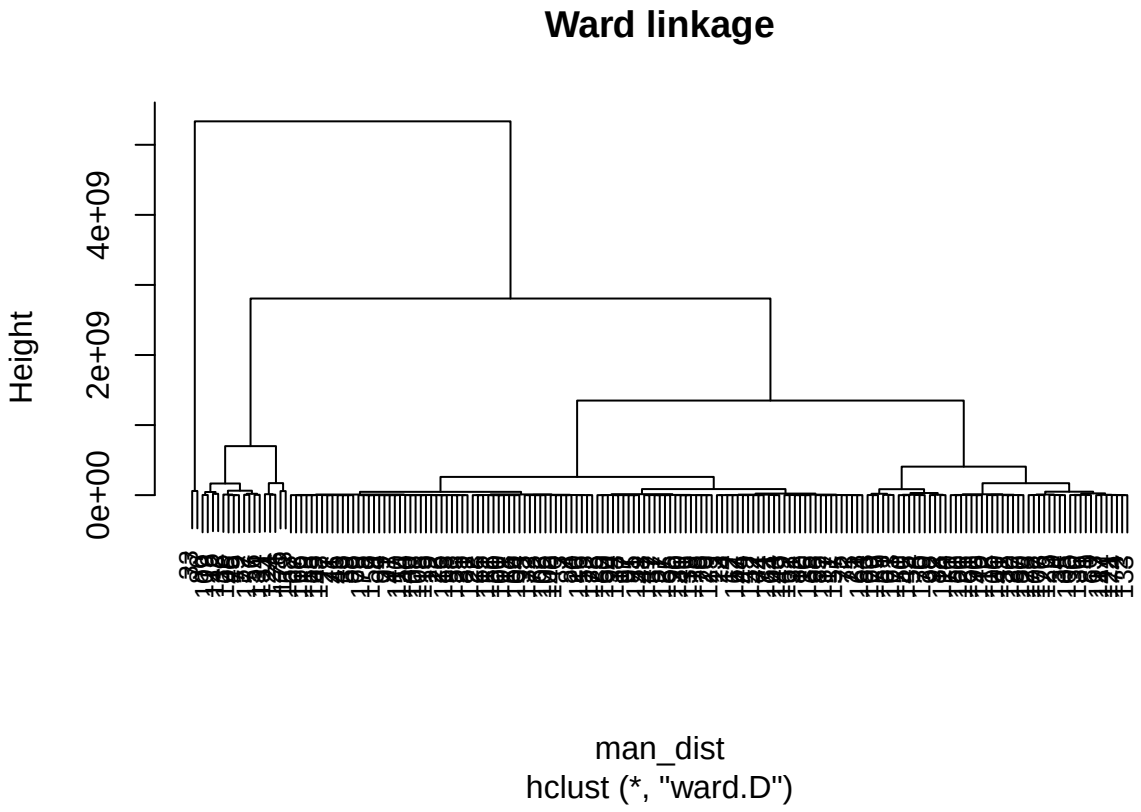
Ward linkage

```
ward <- hclust(man_dist,method="ward")
cl_ward <- cutree(ward,4)
table(cl_ward)
#> cl_ward
#>   1   2   3   4
#>  51 111  17   2
```

This one is acceptable. let's move on and analyze it.

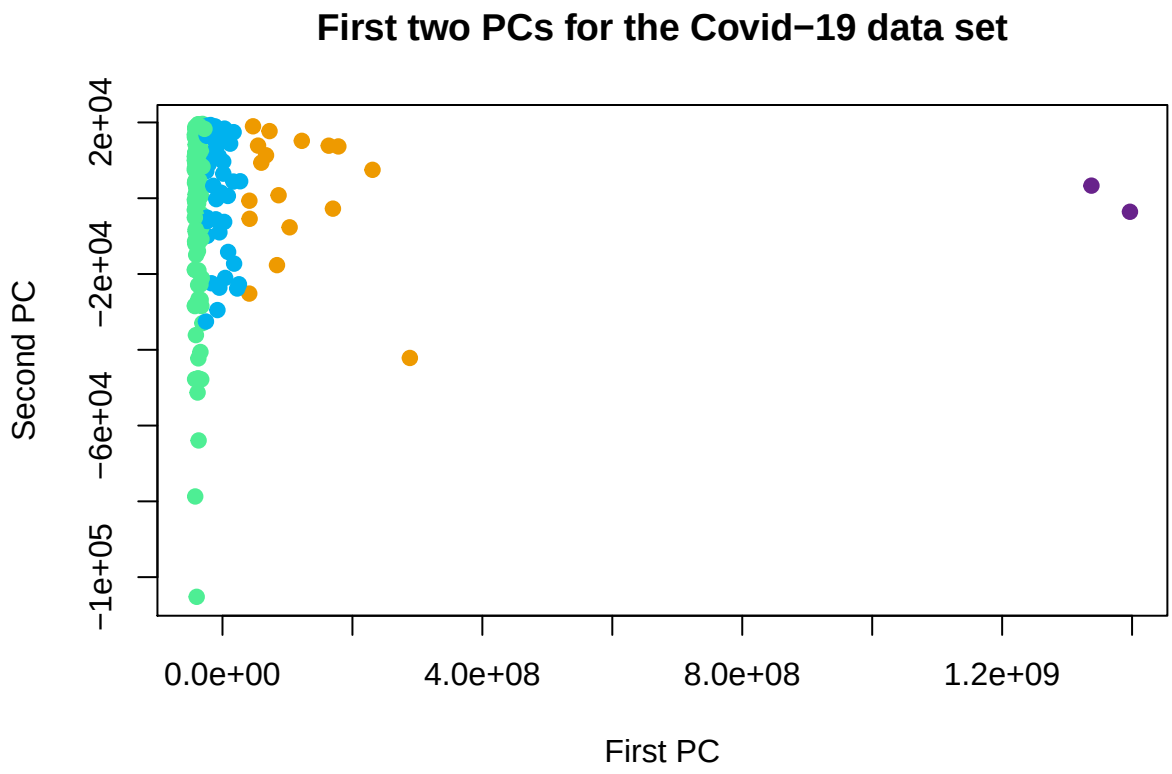
Analysis

```
plot(ward,main="Ward linkage",cex=0.8)
```



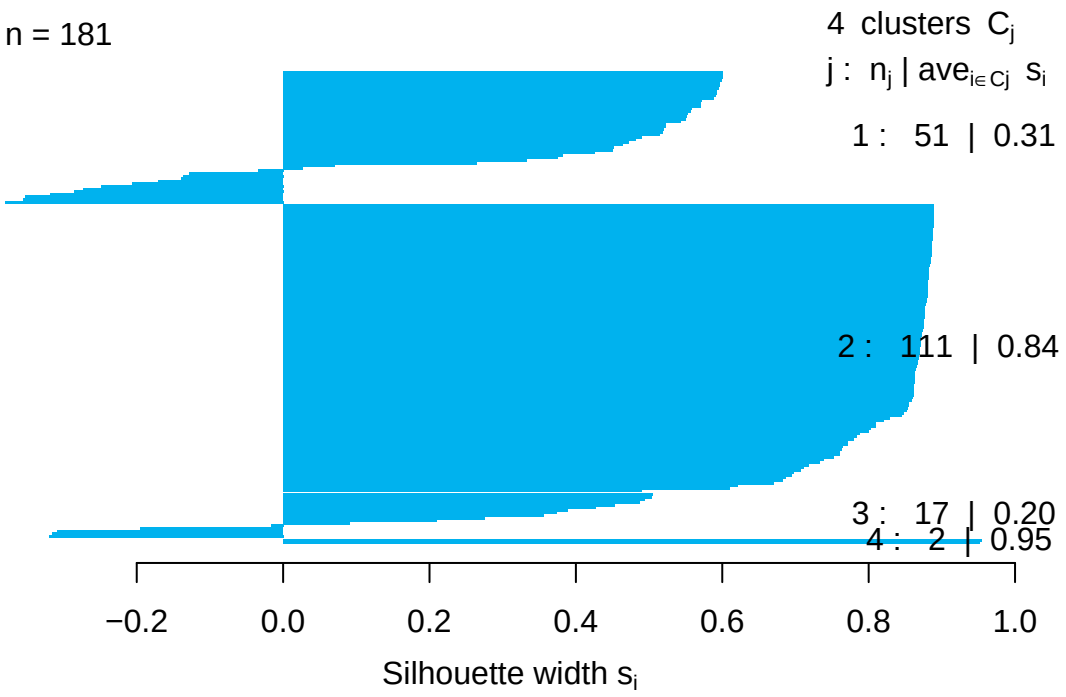
Since our assumed K is 4, we need to cut the highest connection, then we can have our clusters.

```
colors_ward <- c(color_1,color_2,color_3,color_4)[cl_ward]
plot(Z,pch=19,col=colors_ward,main="First two PCs for the Covid-19 data set",xlab="First PC",ylab="Second PC")
```



Then we can check the silhouette plot:

```
sil_ward <- silhouette(cl_ward,man_dist)
plot(sil_ward,main='silhouette',col=color_1)
```



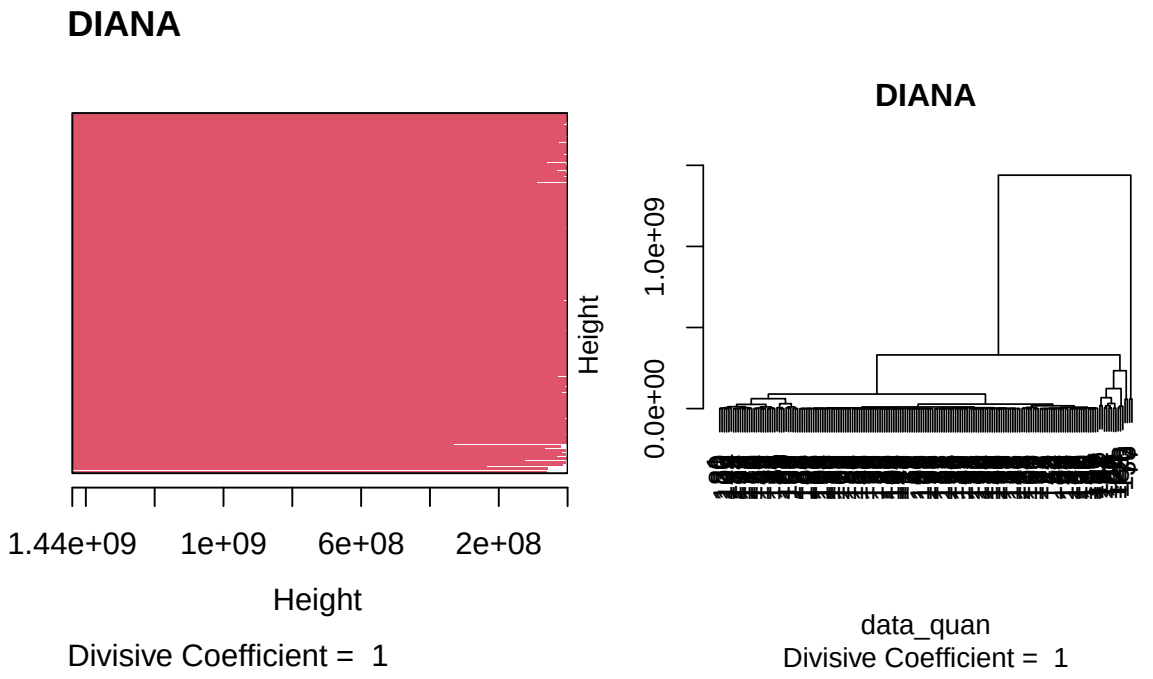
We can also charactsize the clusters as following table: from 1 to 4 means from lowest(fewest) to highest(most).

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	3	4	4	3
cluster3	3	2	4	2	2	2
cluster4	4	4	2	3	3	4

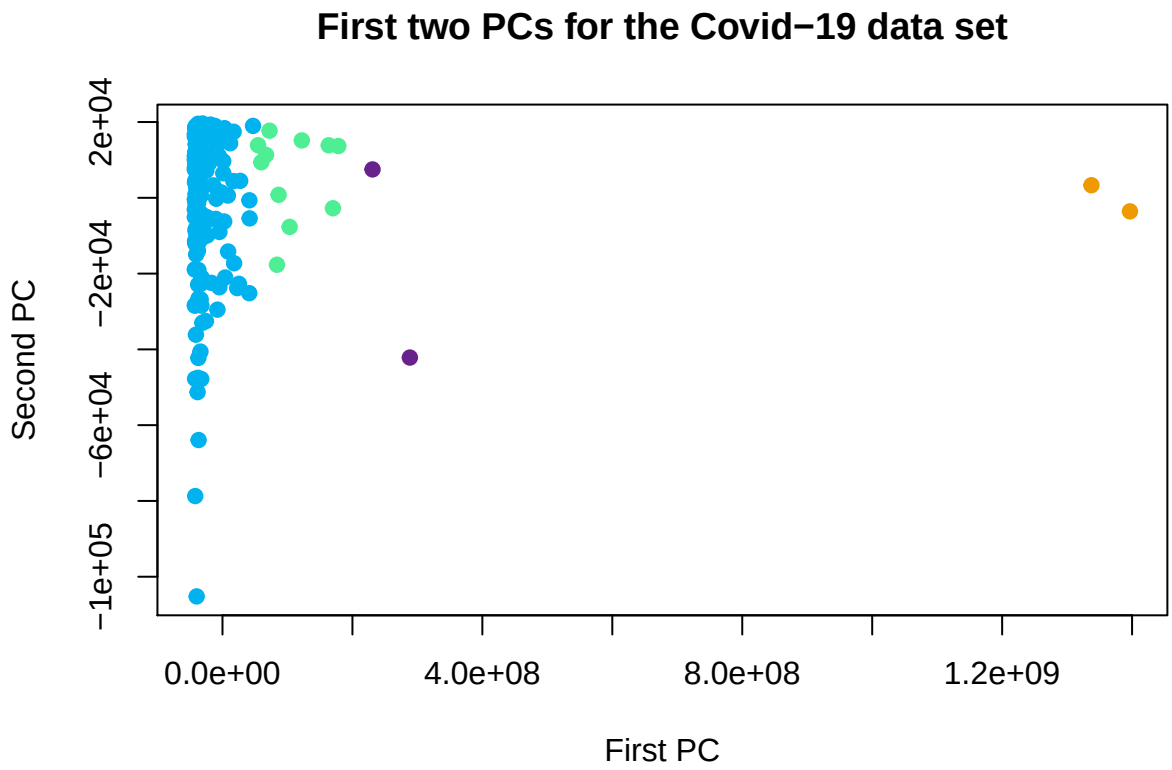
Divisive algorithms

```
diana <- diana(data_quan,metric="manhattan")
cl_diana <- cutree(diana,4)
table(cl_diana)
#> cl_diana
#>   1   2   3   4
#> 166  11   2   2
```

```
plot(diana,main="DIANA")
```




```
colors_diana <- c(color_1,color_2,color_3,color_4)[cl_diana]
plot(Z,pch=19,col=colors_diana,main="First two PCs for the Covid-19 data set",xlab="First PC",ylab="Second PC")
```



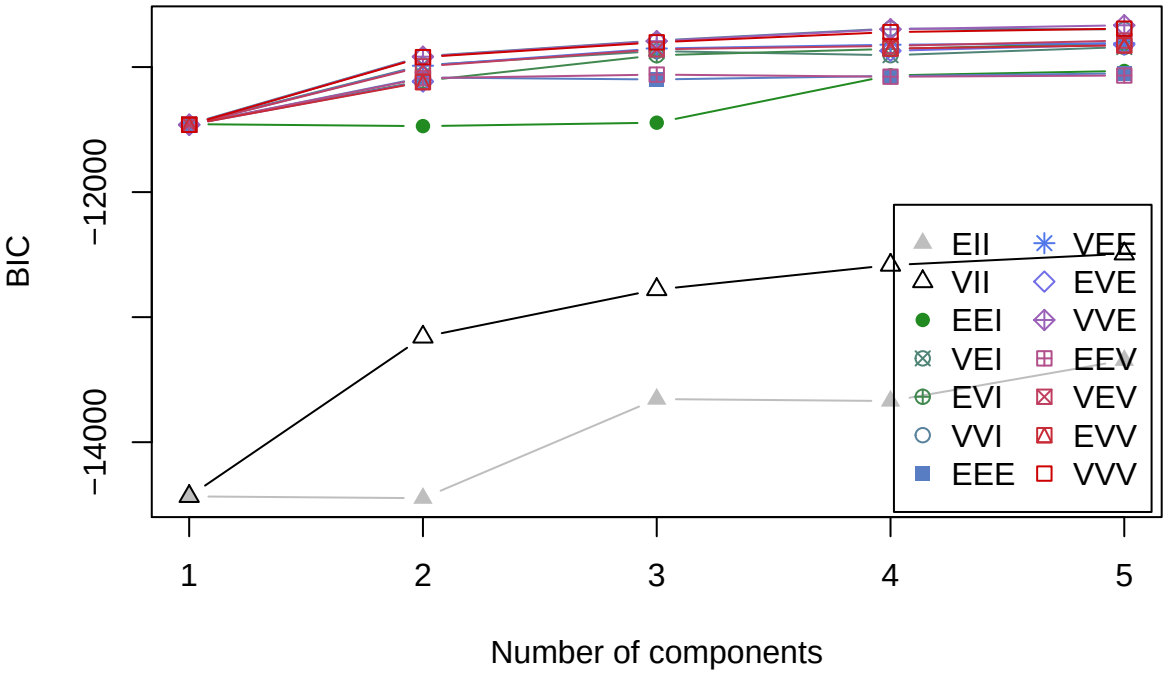
There are too many entries of cluster 1, we can hardly say that it a good one.

So among all **Hierarchical clusterings** we will choose the result of **Ward**.

Model-based clustering

BIC

```
BIC <- mclustBIC(Z,G=1:5)
#> fitting ...
#>   |                                                                      |
```



Model

```
Mclust <- Mclust(Z,x=BIC)
summary(Mclust)
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust VVE (ellipsoidal, equal orientation) model with 5 components:
#>
#>  log-likelihood   n df    BIC    ICL
#>           -5268 181 25 -10666 -10711
#>
#> Clustering table:
#>  1  2  3  4  5
#> 36 46 49  3 47
Mclust$classification
#>   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
#>   1   1   2   3   3   5   2   2   5   3   3   1   3   1   1   5   3   3   2   2
#>  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
#>   3   2   3   5   2   3   2   2   1   5   3   5   4   1   1   5   1   5   2   2
#>  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
#>   3   3   2   3   5   2   2   3   3   5   5   5   1   5   2   5   3   2   5   2
#>  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
#>   5   2   1   1   2   2   2   3   2   5   2   3   3   1   3   5   4   3   5   5
#>  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
#>   3   3   5   2   3   5   5   5   3   1   2   5   3   3   3   1   3   2   3   5
#> 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
#>   2   2   3   2   5   3   1   2   5   2   2   1   3   5   2   2   1   1   2   1
#> 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
#>   5   2   1   5   1   3   3   1   3   3   5   3   5   5   1   5   3   3   3   3
```

```
#> 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
#>   5   5   1   5   5   1   3   2   1   3   3   1   2   2   3   2   3   2   2   1
#> 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
#>   1   1   5   1   2   2   3   5   5   1   5   2   4   5   2   5   5   1   5   1
#> 181
#>   1
```

Parameters

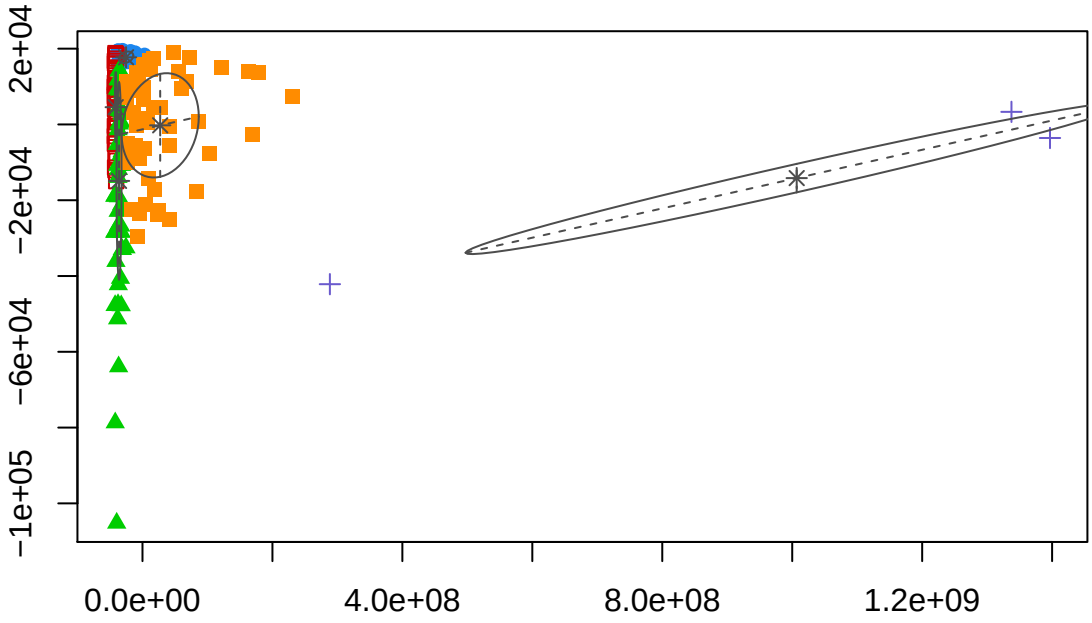
Here is the parameters' probability and mean vector

```
Mclust$parameters$pro
#> [1] 0.19897 0.22021 0.28389 0.01658 0.28035
```

```
Mclust$parameters$mean
#>           [,1]      [,2]      [,3]       [,4]       [,5]
#> [1,] -25285823 -41187534 -36080502 1006910621 27266760.2
#> [2,]     17725      4556    -14897     -14157     -235.2
```

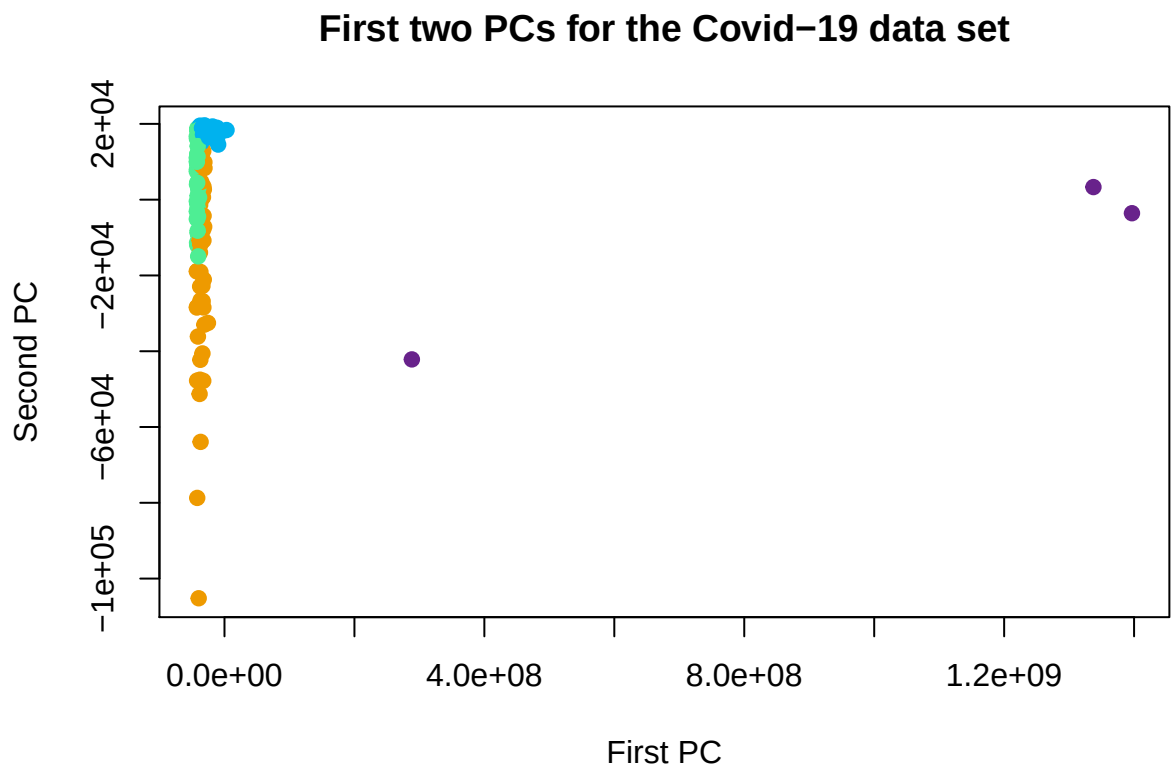
Mclust plot

```
plot(Mclust,what="classification")
```

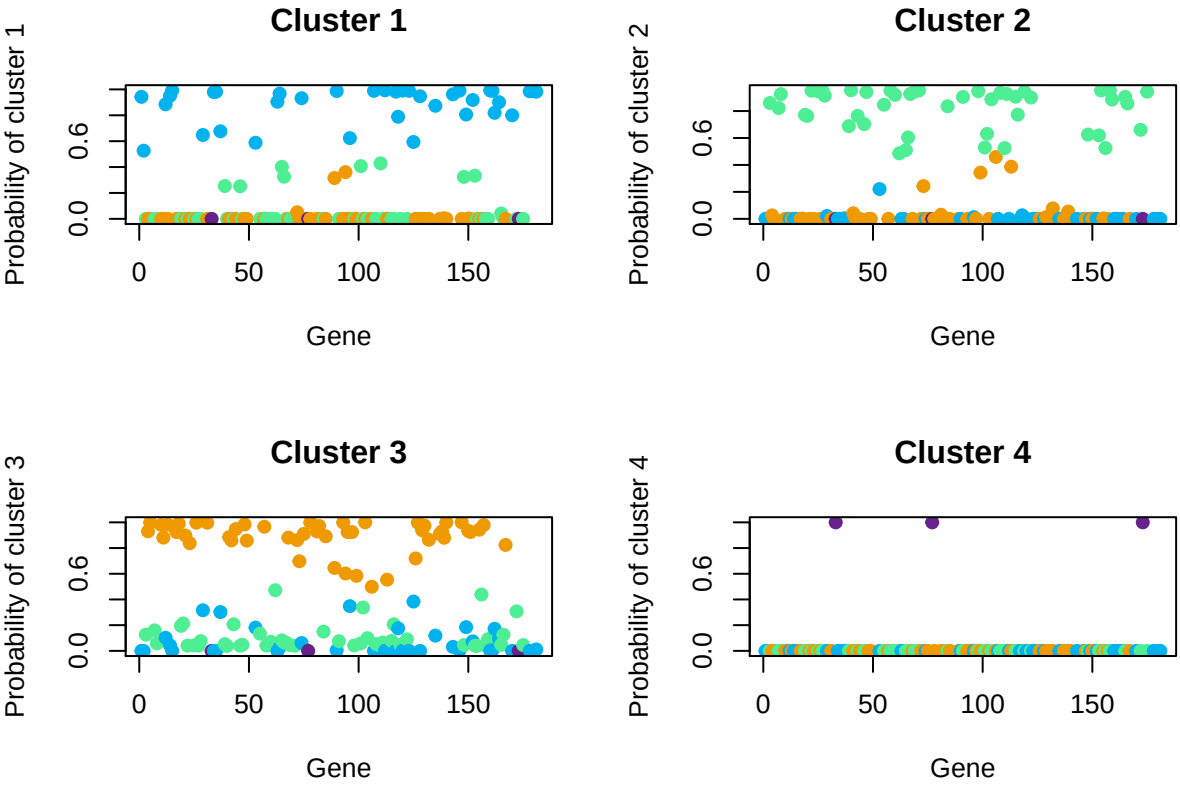


PCA plot

```
colors_Mclust <- c(color_1,color_2,color_3,color_4)[Mclust$classification]
plot(Z,pch=19,col=colors_Mclust,main="First two PCs for the Covid-19 data set",xlab="First PC",ylab="Second PC")
```

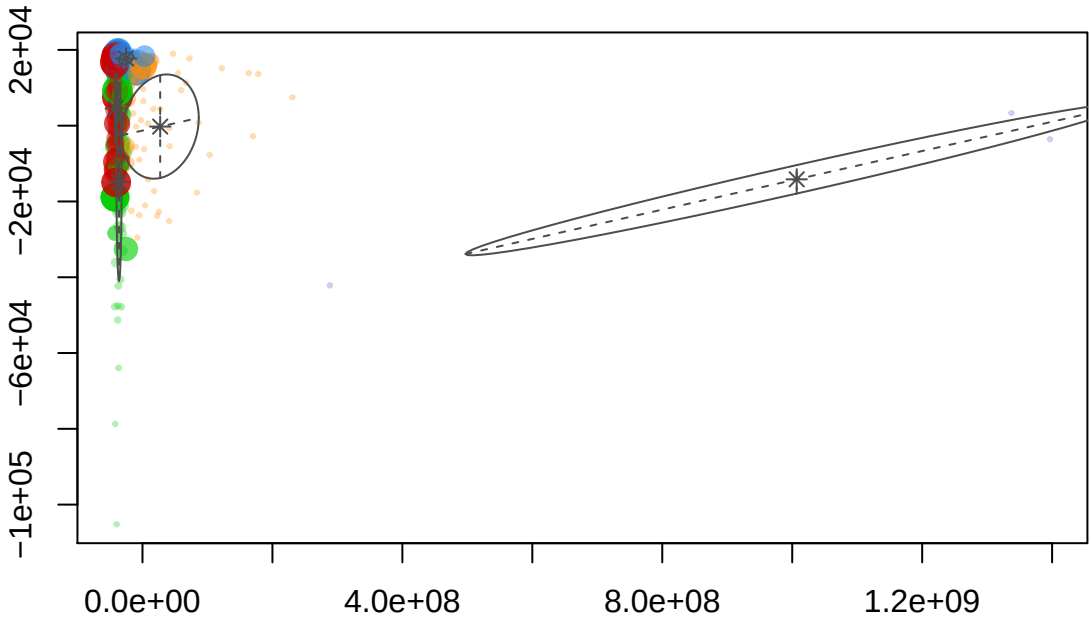


Probability plot



These four plots show the probability of the observations locate in the specific cluster. We can see that each cluster has a fairly good performance. it is reliable.

```
par(mfrow=c(1,1))
plot(Mclust,what="uncertainty")
```



And here we can check the plot of those observations labeled with **uncertainty**

Analysis

We can also characterize the clusters as following table:

From 1 to 4 means from lowest(fewest) to highest(most).

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	2	2	1	1	1	1
cluster2	3	3	2	3	3	4
cluster3	4	4	3	4	4	3
cluster4	1	1	4	2	2	2

Analysis of the results

We set the K into 4, i.t. we wish the algorithm can split the dataset into 4 clusters with clear border with the others. And there shouldn't be too many or too few observations in one cluster.

Hence we present the result from *K-Means,PAM,Agglomerative algorithms with ward linkage,Model-based*. And here we can put all mean vectors together.

1. K-Means:

We can check the cluster number and which countries are in the same cluster, but the table would be to long to show it.

from 1 to 4 means from lowest(fewest) to highest(most).

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	3	4	4	3
cluster3	3	2	4	2	2	2
cluster4	4	4	2	3	3	4

2. PAM:

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	2	3	3	4
cluster3	3	4	3	4	4	3
cluster4	4	2	4	2	2	2

3. Ward—linkage Hierarchical clustering

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	1	1	1	1	1	1
cluster2	2	3	3	4	4	3
cluster3	3	2	4	2	2	2
cluster4	4	4	2	3	3	4

4. Model_based

cluster	cases	death rate	economic	average age	medical resources	stringency
cluster1	2	2	1	1	1	1
cluster2	3	3	2	3	3	4
cluster3	4	4	3	4	4	3
cluster4	1	1	4	2	2	2

Factor Analysis

Our objective is to find out which are the main characteristics of the countries in our dataset using factor analysis, based on all these 18 numeric variables that we have. Each factor is a summary of some correlated variables, these factors are not correlated and are all equally important.

We have two aims doing factor analysis. One is to find the factors and try to understand them, the other is to estimate the value of the factor for each observation. To check whether these factors are good or not, we use communalities, which are values between 0 and 1, and they represent the percentage of variance explained.

As the countries are very different from one to anothers, we have divided the countries into four following groups to do the factor analysis by using the variable Human Development Index: low HDI, medium HDI, high medium and very high HDI in order to perform the factor analysis.

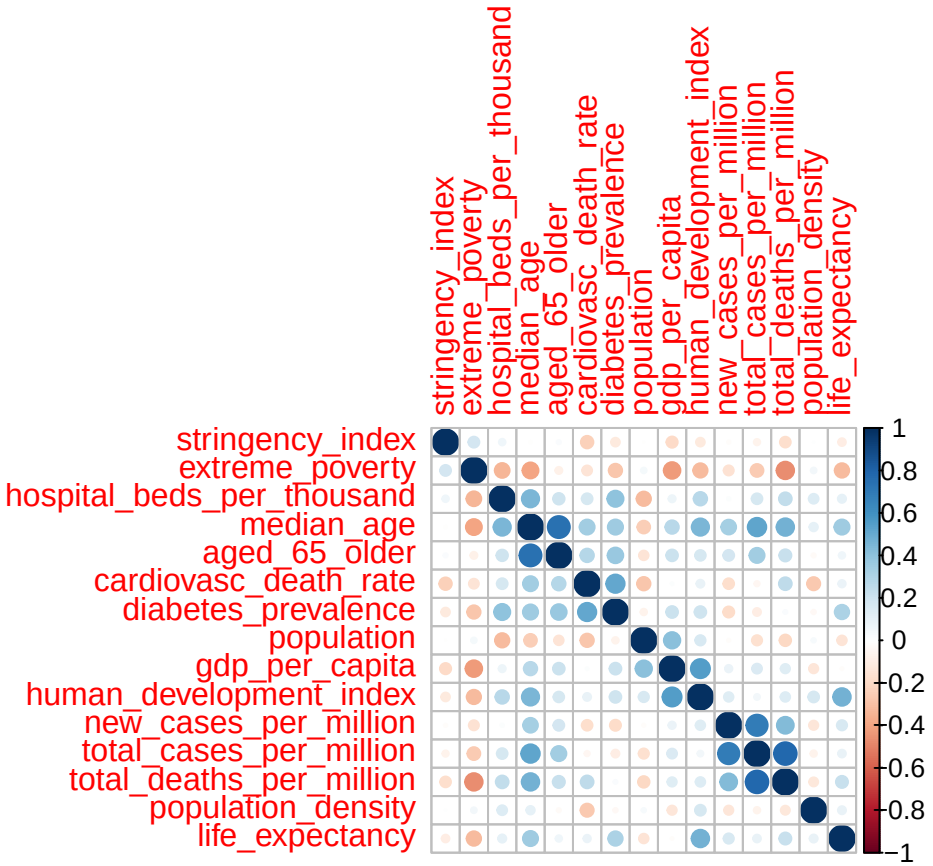
```
rm(list = ls())
options(digits=4)

X <- read.csv("./data/data_imp.csv",header = TRUE)
X <-as.data.frame(X[,-1])
XX <- X[,-(1:2)]

XX_low <- XX %>% filter(development=="low") %>% dplyr::select(-development)
XX_medium <- XX %>% filter(development=="medium") %>% dplyr::select(-development)
XX_high <- XX %>% filter(development=="high") %>% dplyr::select(-development)
XX_veryhigh <- XX %>% filter(development=="very high") %>% dplyr::select(-development)
```

1. Low HDI:

- *PCFA* Our very first step is to obtain a correlation plot, it is the most important visual analysis for the variables in dataset, and is critical for factor analysis.

We sorted the variables using their correlations, and we can see that there are groups of variables that are highly correlated. For instance, we have a group of variables which is relatad to the situation of covid of the country: new cases per million, total cases per million and total deaths per million; and another group of variables that are more or less highly correlated, median age and aged 65 or older.

```
# Sample size and dimension of the personality data set
n <- nrow(XX_low)
p <- ncol(XX_low)

corrplot(cor(XX_low),order="hclust")
```

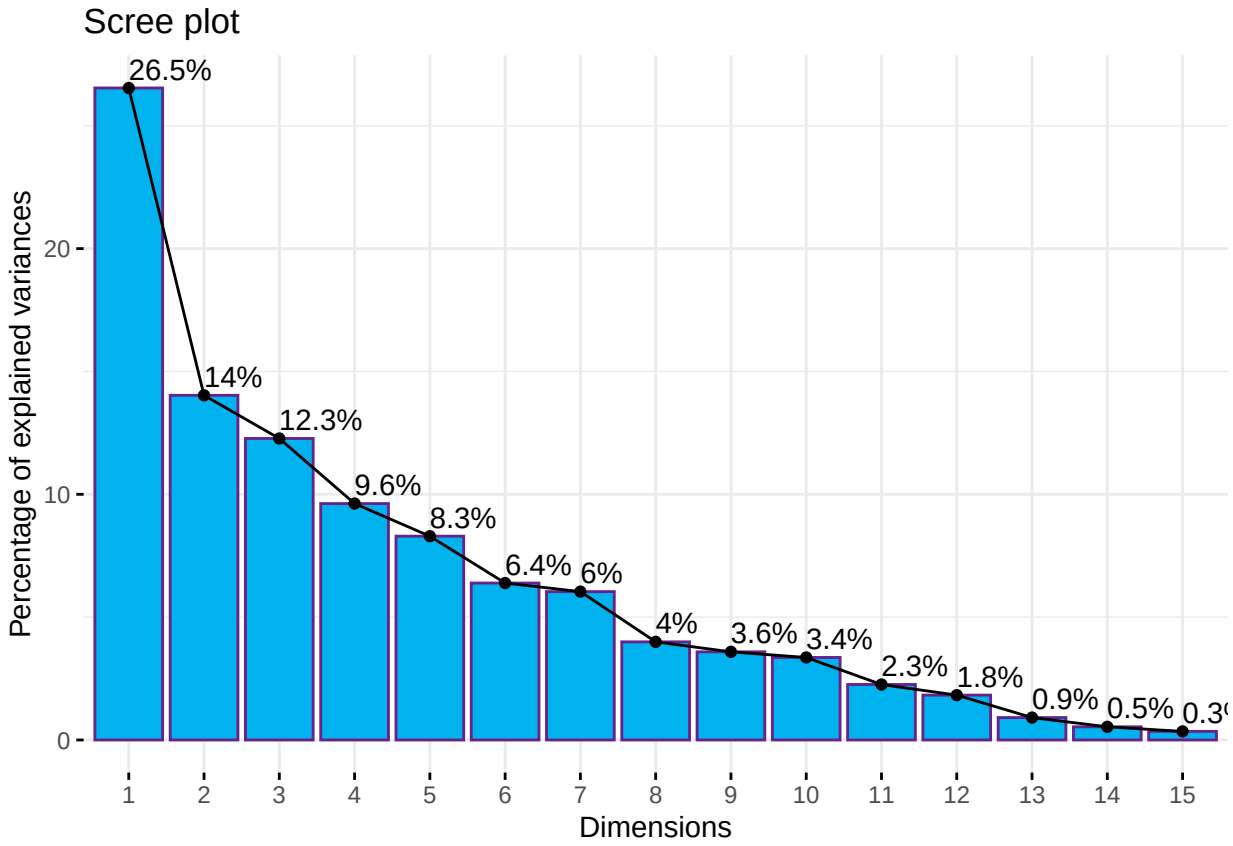



```
# There are groups of correlated variables that may suggest a factor structure
```

To start the analysis, we scale the variables in order to obtain better results.

We can check here the variance explained by each principal component, e.g. the first principal component explains 26.5% of the total variability and the second principal component explains 14%.

```
# Principal Component Factor Analysis
# Obtain the PCs of the univariate standardized variables
Y <- scale(XX_low)
Y_pcs <- prcomp(Y)

# Screeplot with all the eigenvalues
fviz_eig(Y_pcs,ncp=p,addlabels=T,barfill=color_1,barcolor=color_4)
```



Now it is necessary to check the eigenvalues and the cumulative and the cumulative percentage of explained variance, we have decided to take 5 principal components, by doing that, we will be using the 5/15, which is the 33% of the variables and will be keeping the 70.76% of the total information.

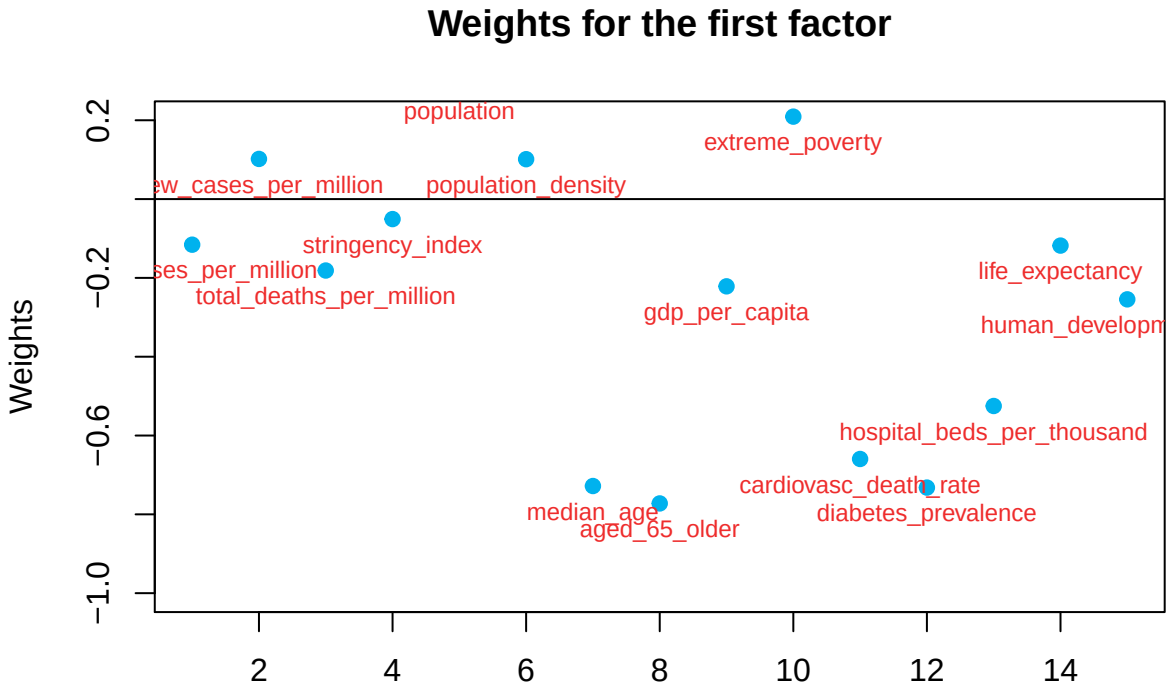
```
get_eigenvalue(Y_pcs)
#>        eigenvalue variance.percent cumulative.variance.percent
#> Dim.1    3.98070          26.5380                       26.54
#> Dim.2    2.10420          14.0280                       40.57
#> Dim.3    1.84078          12.2719                       52.84
#> Dim.4    1.44343           9.6229                       62.46
#> Dim.5    1.24421           8.2947                       70.76
#> Dim.6    0.95808           6.3872                       77.14
#> Dim.7    0.90579           6.0386                       83.18
#> Dim.8    0.59906           3.9937                       87.18
#> Dim.9    0.53809           3.5872                       90.76
#> Dim.10   0.50375           3.3583                       94.12
#> Dim.11   0.33860           2.2574                       96.38
#> Dim.12   0.27378           1.8252                       98.20
#> Dim.13   0.13680           0.9120                       99.12
#> Dim.14   0.08035           0.5356                       99.65
#> Dim.15   0.05238           0.3492                      100.00
```

From now on, let us focus on the first five PCs.

```
r <- 5
# Estimate the matrix M and use the varimax rotation for interpretability
M_pcfa <- Y_pcs$rotation[,1:r] %*% diag(Y_pcs$sdev[1:r])
M_pcfa <- varimax(M_pcfa)
M_pcfa <- loadings(M_pcfa)[1:p,1:r]
```

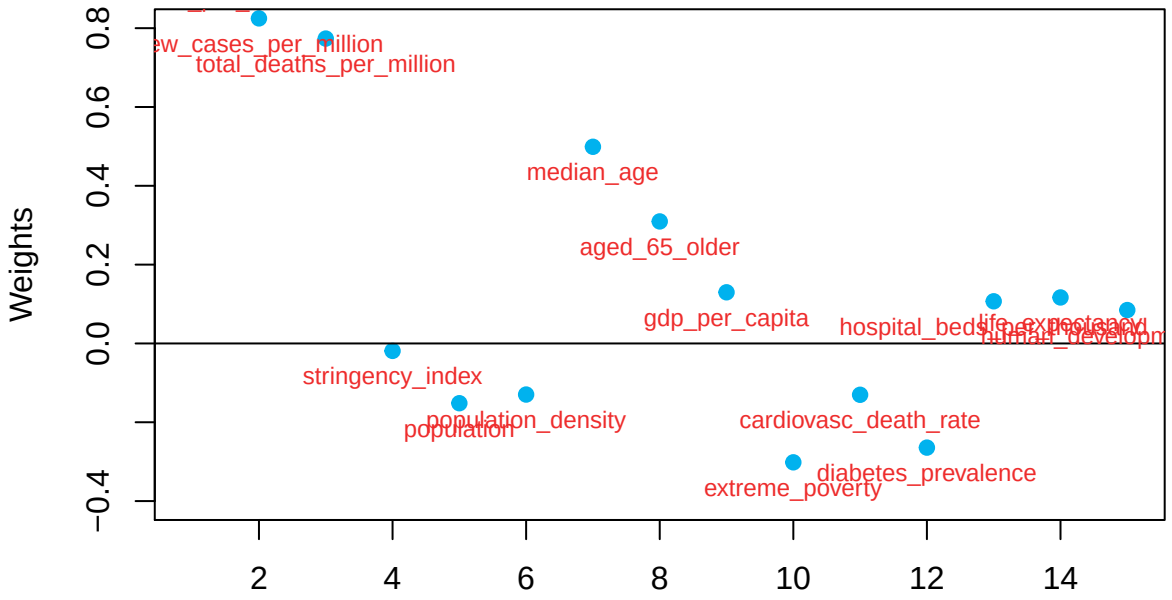
We can observe here that the first factor appears to be an index of the mixture of the inverse of disease rate and the age variables, as we can see that the variable diabetes prevalence, cardiovascular rate, median age and aged 65 or older have a very low negative value (around -0.8).

```
plot(1:p,M_pcfa[,1],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,0.2),main="Weights for the first factor")
abline(h=0)
text(1:p,M_pcfa[,1],labels=colnames(XX_low),pos=1,col=color_5,cex=.75)
```



However, the second factor seems to be an index that measures the situation of the pandemic in each country as the variables cases per million, new cases per million and total deaths per million are very highly weighted, around 0.8.

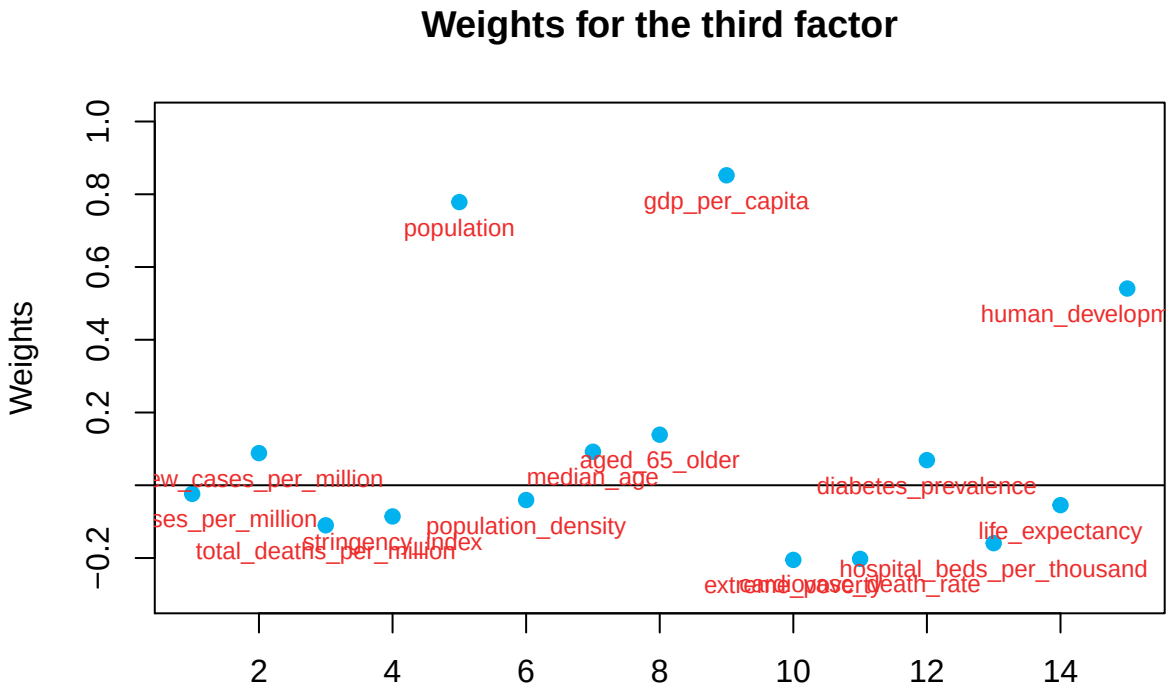
```
plot(1:p,M_pcfa[,2],pch=19,col=color_1,xlab="",ylim=c(-0.4,0.8),ylab="Weights",main="Weights for the second factor")
abline(h=0)
text(1:p,M_pcfa[,2],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



The third factor appears to be an index of how developed a country is, because the variables gdp per capita,

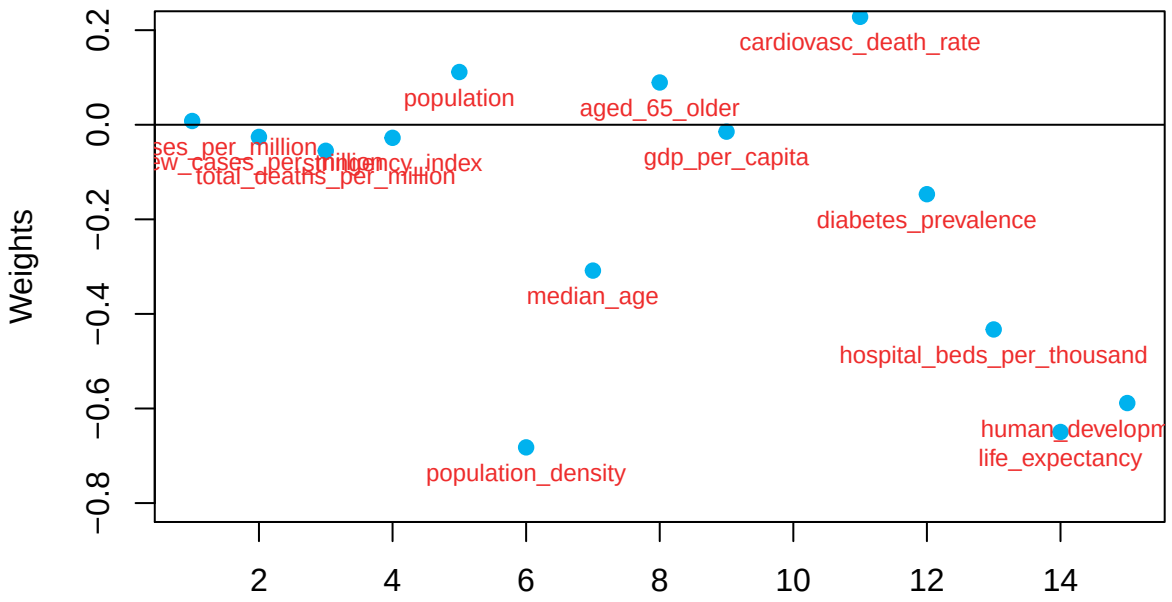
population and human development index have very high weights.

```
plot(1:p,M_pcfa[,3],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.3,1),main="Weights for the third factor")
abline(h=0)
text(1:p,M_pcfa[,3],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



The fourth factor appears to be an index of the inverse of how developed a country is, as the variables life expectancy and HDI are negative with also a population density with a very low negative value.

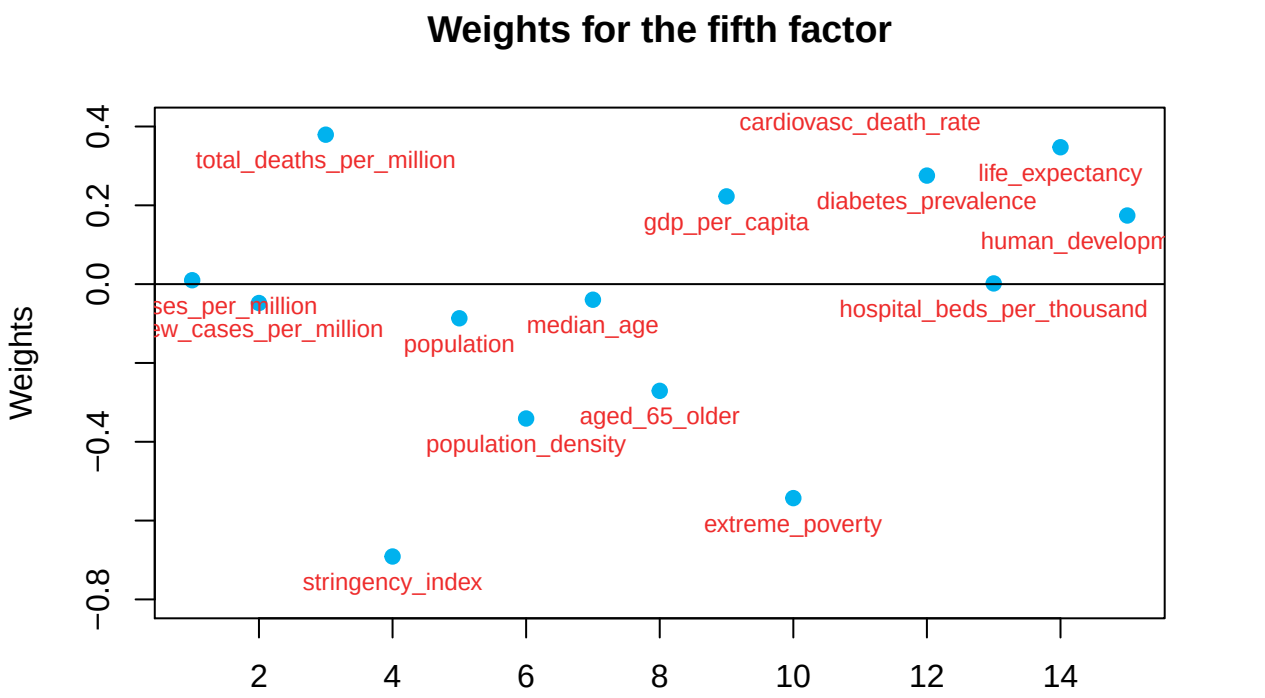
```
plot(1:p,M_pcfa[,4],pch=19,col=color_1,ylim=c(-0.8,0.2),xlab="",ylab="Weights",main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_pcfa[,4],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



The fifth factor appears to be an index of a mixture of all the variables, but the main variable of this factor

is stringency index and extreme poverty with negative values.

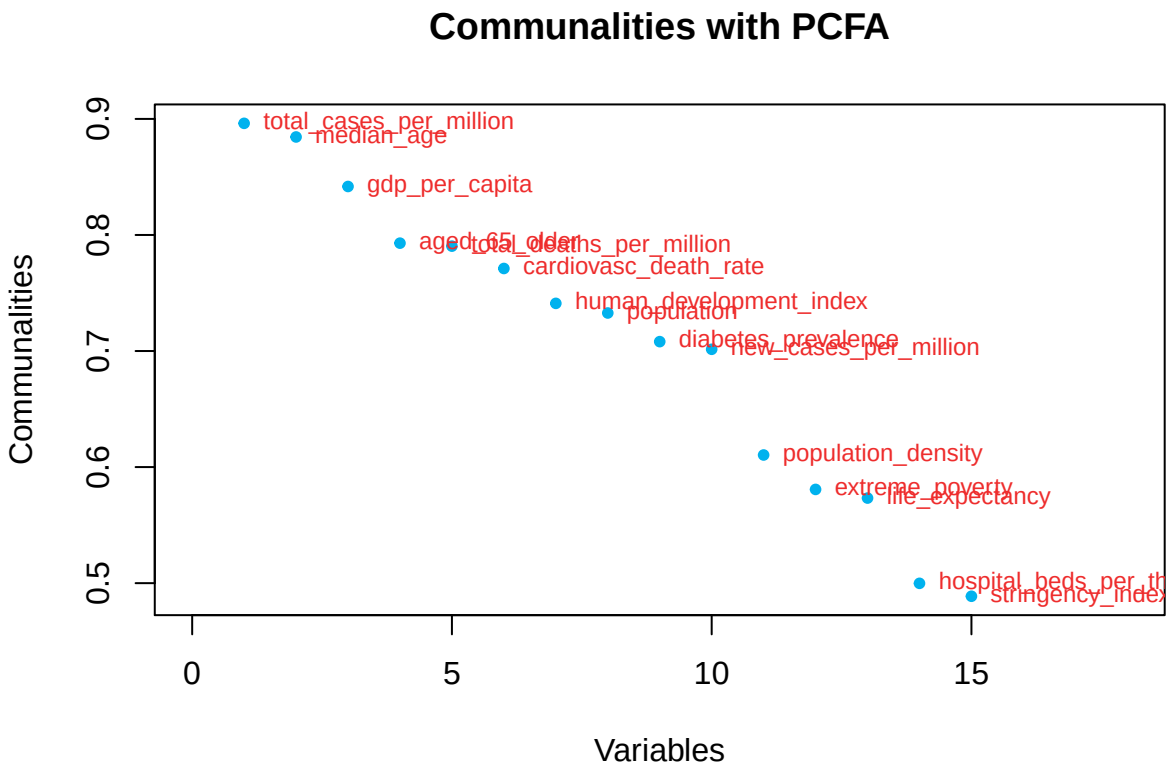
```
plot(1:p,M_pcfa[,5],pch=19,ylim=c(-0.8,0.4),col=color_1,xlab="",ylab="Weights",main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_pcfa[,5],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```




```
# Estimate the covariance matrix of the errors
Sigma_nu_pcfa <- diag(diag(cov(Y) - M_pcfa %*% t(M_pcfa)))
```

Communalities are values that range from 0 to 1, higher the value, higher the percentage of variabilities explained by the factor model. We plot the values of communalities and we can observe that the aspects of countries that are better explained by the factors are total covid cases per million, median age and gdp per capita (more then 80%). And the variables that are not explained that well are hospital beds per thousand and stringency index (50% or below).

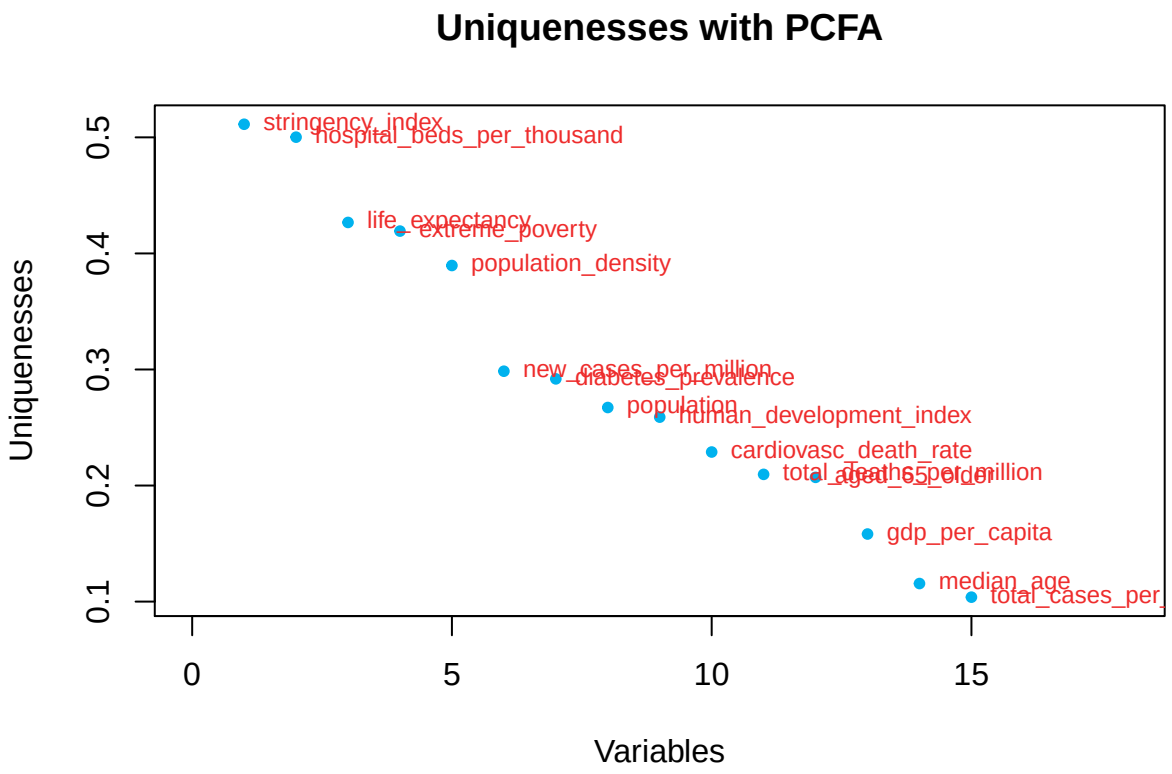
```
# Communalities and uniquenesses
comm_pcfa <- diag(M_pcfa %*% t(M_pcfa))
comm_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.8962                     0.7015
#>   total_deaths_per_million           stringency_index
#>                     0.7904                     0.4887
#>                 population         population_density
#>                     0.7328                     0.6104
#>                 median_age               aged_65_older
#>                     0.8845                     0.7929
#>             gdp_per_capita            extreme_poverty
#>                     0.8418                     0.5808
#>      cardiovasc_death_rate         diabetes_prevalence
#>                     0.7711                     0.7081
#> hospital_beds_per_thousand             life_expectancy
#>                     0.4998                     0.5733
#>    human_development_index
#>                     0.7411
plot(1:p,sort(comm_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Communalities",
     main="Communalities with PCFA")
text(1:p,sort(comm_pcfa,decreasing=TRUE),labels=names(sort(comm_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```



The values of uniqueness are the same, but the other way around (1-Communality).

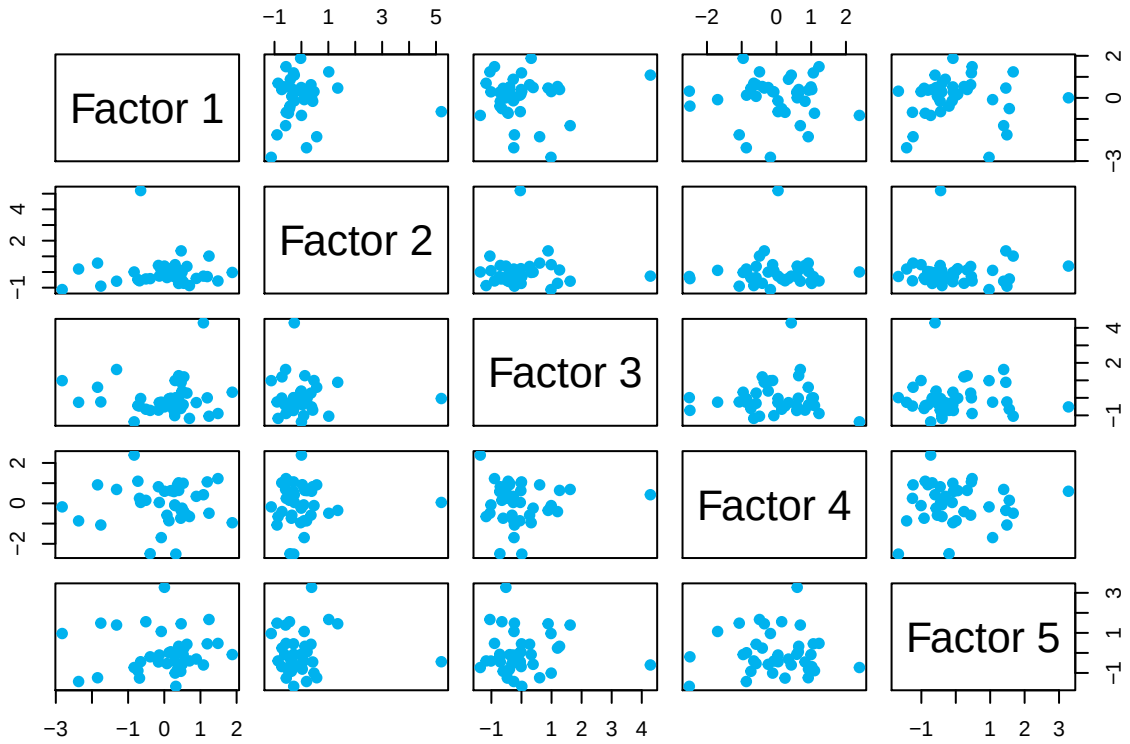
```
uniq_pcfa <- 1 - comm_pcfa
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.1038                     0.2985
#>   total_deaths_per_million          stringency_index
#>                     0.2096                     0.5113
#>                 population         population_density
#>                     0.2672                     0.3896
#>                 median_age              aged_65_older
#>                     0.1155                     0.2071
#>             gdp_per_capita           extreme_poverty
#>                     0.1582                     0.4192
#>      cardiovasc_death_rate        diabetes_prevalence
#>                     0.2289                     0.2919
#> hospital_beds_per_thousand            life_expectancy
#>                     0.5002                     0.4267
#>    human_development_index
#>                     0.2589
names(uniq_pcfa) <- names(comm_pcfa)
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.1038                     0.2985
#>   total_deaths_per_million          stringency_index
#>                     0.2096                     0.5113
#>                 population         population_density
#>                     0.2672                     0.3896
#>                 median_age              aged_65_older
#>                     0.1155                     0.2071
#>             gdp_per_capita           extreme_poverty
#>                     0.1582                     0.4192
#>      cardiovasc_death_rate        diabetes_prevalence
#>                     0.2289                     0.2919
#> hospital_beds_per_thousand            life_expectancy
#>                     0.5002                     0.4267
#>    human_development_index
#>                     0.2589
```

```
plot(1:p,sort(uniq_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Uniquenesses",
     main="Uniquenesses with PCFA")
text(1:p,sort(uniq_pcfa,decreasing=TRUE),labels=names(sort(uniq_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```

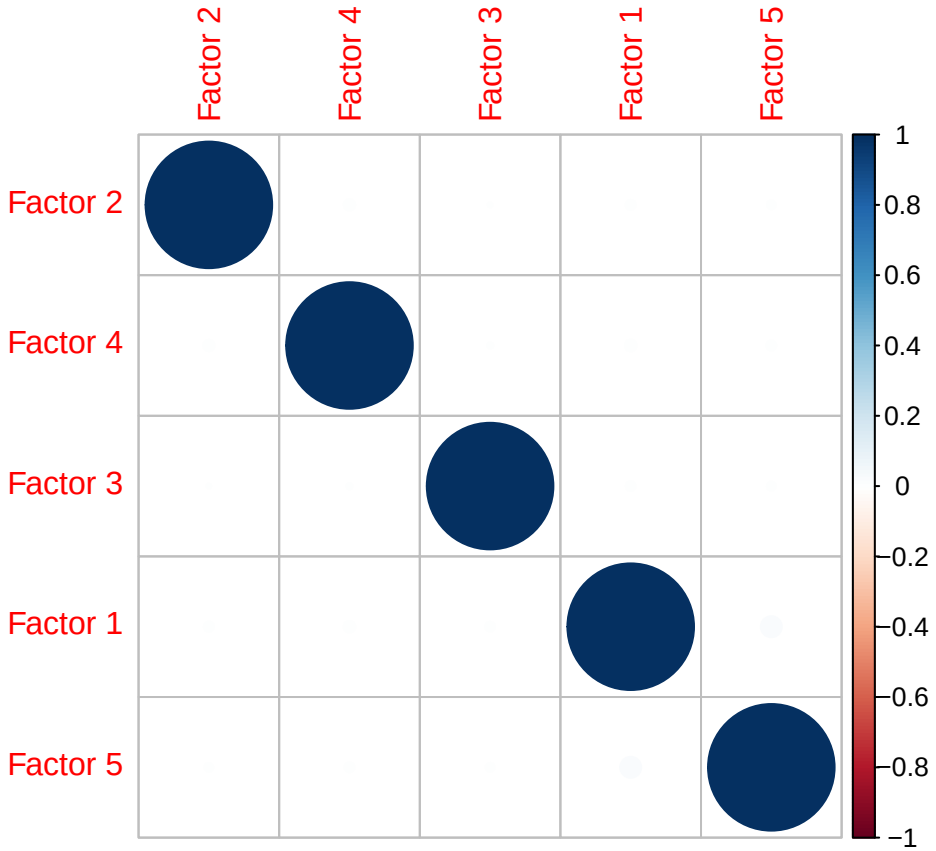


From the following plot we can tell that all the factors are uncorrelated.

```
# Estimate the factor scores
F_pcfa <- Y %*% solve(Sigma_nu_pcfa) %*% M_pcfa %*% solve(t(M_pcfa) %*% solve(Sigma_nu_pcfa) %*% M_pcfa)
colnames(F_pcfa) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")

pairs(F_pcfa,pch=19,col=color_1)
```



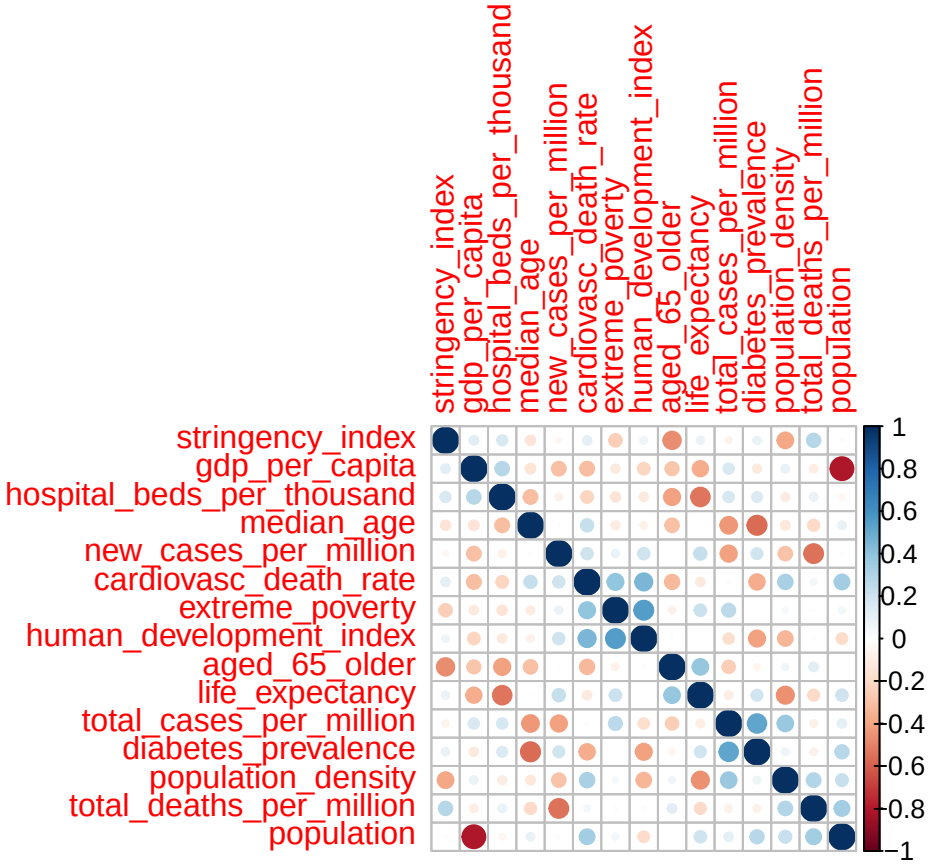

```
corrplot(cor(F_pcfa),order="hclust")
```



We plot the correlations between the residuals of the model. We can see that in the following plot that the correlations outside the diagonals the correlations are very low, except the correlation between population and

gdp per capita, which means that if we include another factor, we might be able to explain the correlation, but as we only have 18 variables, we will only keep 5 factors.

```
# Estimate the residuals
Nu_pcfa <- Y - F_pcfa %*% t(M_pcfa)
corrplot(cor(Nu_pcfa),order="hclust")
```



- *PFA*: It is a method of refinement of PCFA. First step is to obtain the sample correlation matrix of our observations, and this is our sample covariance matrix. Then we obtain the eigenvector and eigenvalues.

```
# Obtain the sample correlation matrix of X, that is the sample covariance matrix of Y
R_X <- cor(XX_low)
# Obtain R_X - Sigma_nu_pcfa, its eigenvectors and eigenvalues
MM <- R_X - Sigma_nu_pcfa
MM_eig <- eigen(MM)
MM_values <- MM_eig$values
MM_vectors <- MM_eig$vectors
```

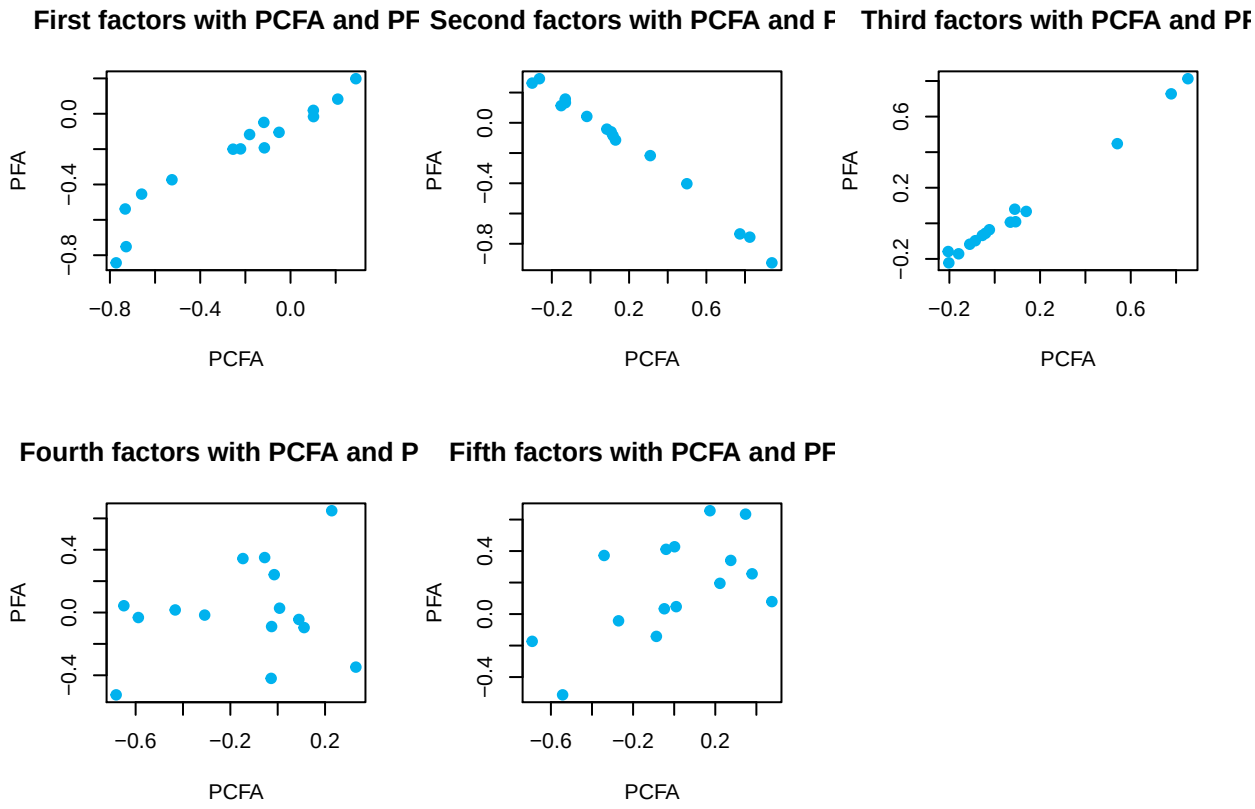
Then we estimate the matrix M using the varimax rotaion.

```
# Estimate the matrix M and use the varimax rotation for interpretability
M_pfa <- MM_eig$vectors[,1:r] %*% diag(MM_eig$values[1:r])^(1/2)
M_pfa <- varimax(M_pfa)
M_pfa <- loadings(M_pfa)[1:p,1:r]
```

The objective here is to compare the results obtained from PCFA and PFA, and it is easy to observe from the following plot that the first three factors of PCFA and PFA are very similar, but the last two factors are quite different, so we plot the estimated weights for those two specific factors.

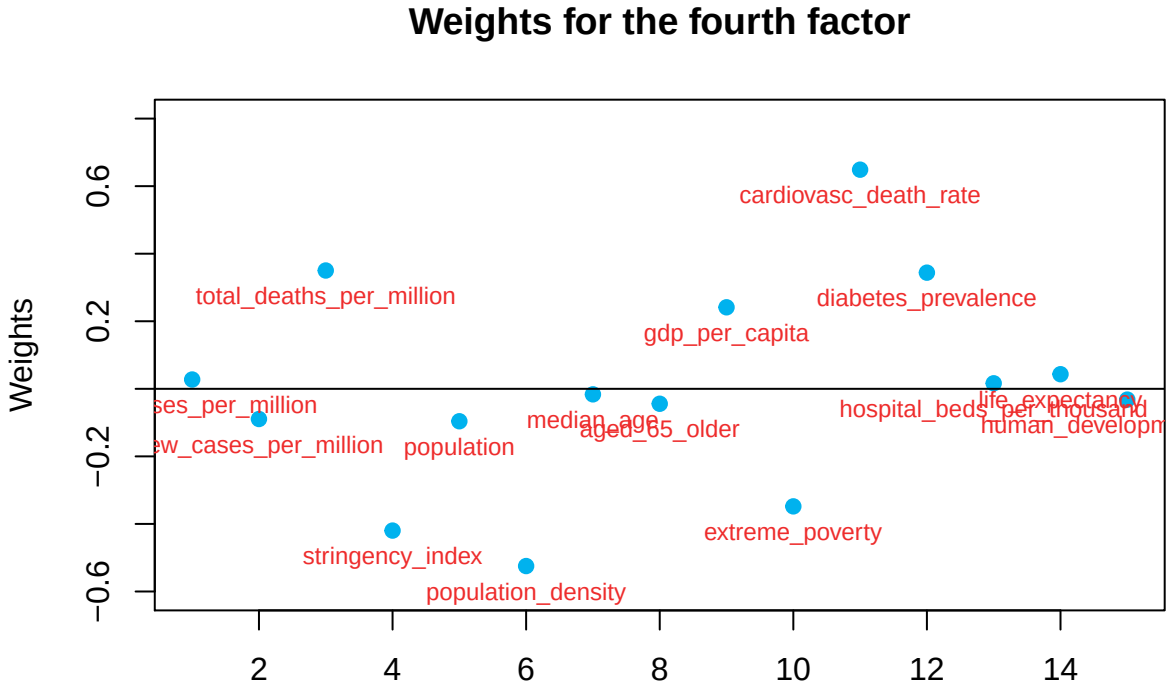
```
# Compare PCFA and PFA estimates of M
par(mfrow=c(2,3))
plot(M_pcfa[,1],M_pfa[,1],pch=19,col=color_1,main="First factors with PCFA and PFA",xlab="PCFA",ylab="PFA")
plot(M_pcfa[,2],M_pfa[,2],pch=19,col=color_1,main="Second factors with PCFA and PFA",xlab="PCFA",ylab="PFA")
plot(M_pcfa[,3],M_pfa[,3],pch=19,col=color_1,main="Third factors with PCFA and PFA",xlab="PCFA",ylab="PFA")
plot(M_pcfa[,4],M_pfa[,4],pch=19,col=color_1,main="Fourth factors with PCFA and PFA",xlab="PCFA",ylab="PFA")
```

```
plot(M_pcfa[,5],M_pfa[,5],pch=19,col=color_1,main="Fifth factors with PCFA and PFA",xlab="PCFA",ylab="PFA")
```



The fourth factor might be a index of the disease rate.

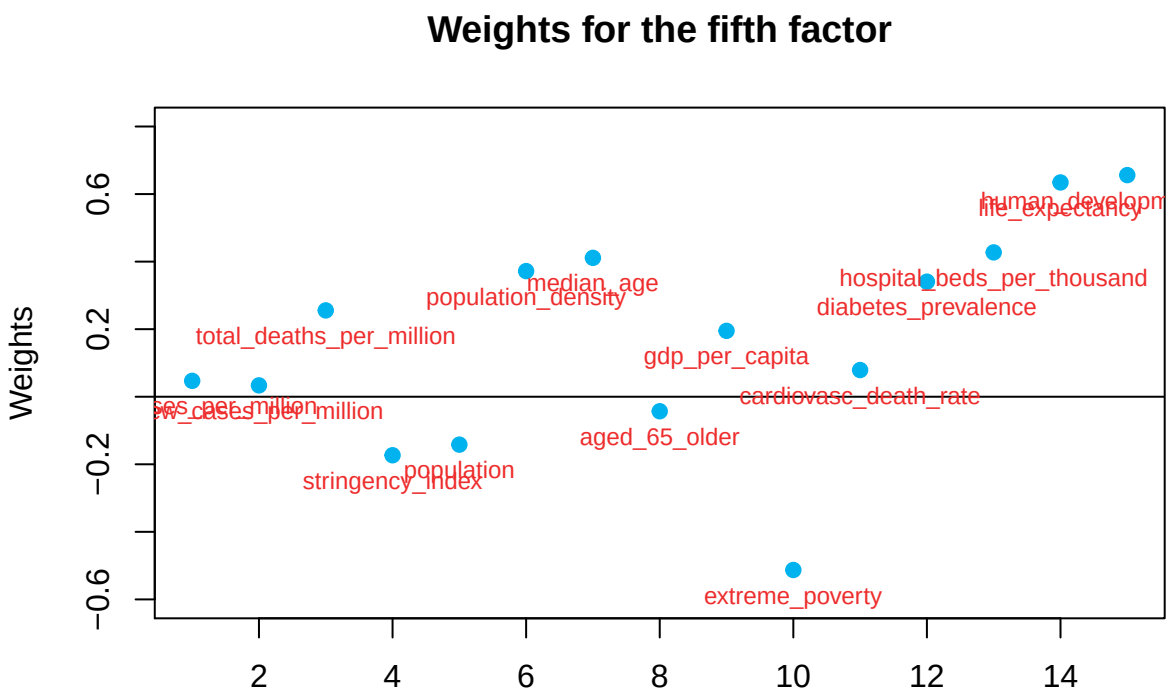
```
plot(1:p,M_pfa[,4],pch=19,col=color_1,xlab="",ylim=c(-0.6,0.8),ylab="Weights",main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_pfa[,4],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



The fifth factor appears to be an index of how developed a country is, as we have high positive weights for

HDI and life expectancy and low negative weight for extreme poverty.

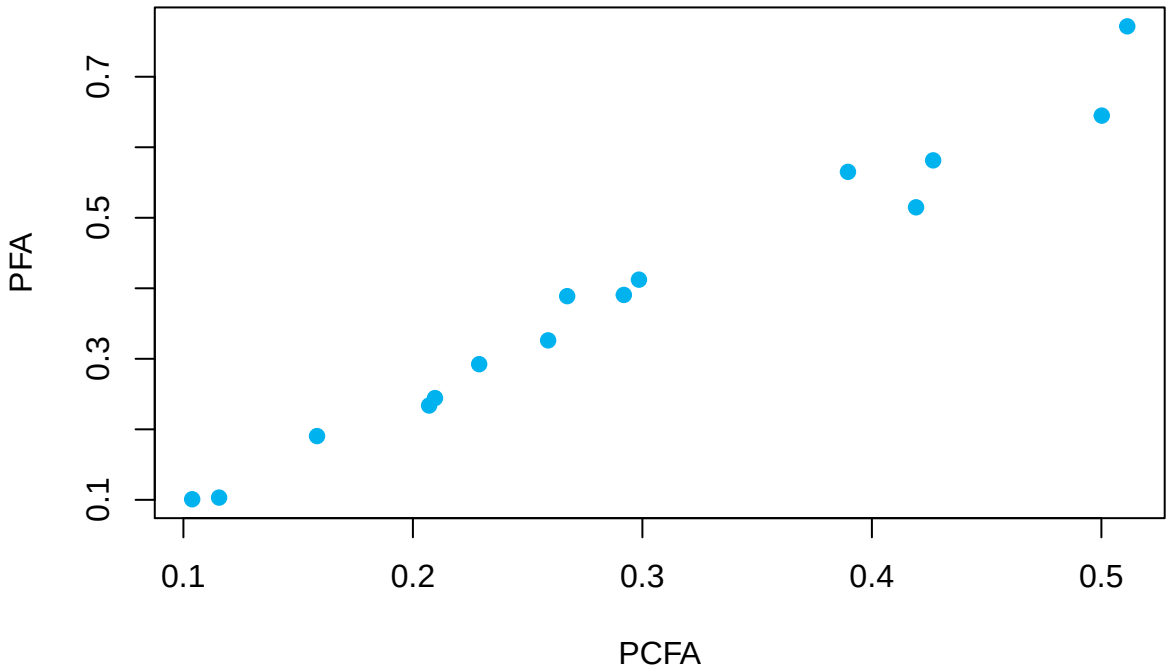
```
plot(1:p,M_pfa[,5],pch=19,col=color_1,xlab="",ylim = c(-0.6,0.8), ylab="Weights",main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_pfa[,5],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



Now we compare the covariance matrix of the errors for PCFA and PFA, to do so, it is necessary to estimate the covariance matrix of the errors, and then plot them with the errors of PCFA method.

We can observe that they are very similar, there are only some small differences.

```
# Estimate the covariance matrix of the errors
Sigma_nu_pfa <- diag(diag(R_X - M_pfa %*% t(M_pfa)))
# Compare with the estimate with the PCFA method
par(mfrow=c(1,1))
plot(diag(Sigma_nu_pcfa),diag(Sigma_nu_pfa),pch=19,col=color_1,main="Noise variances with PCFA and PFA",
     xlab="PCFA",ylab="PFA")
```



Here we compare the values of communalities of the different methods. And we found out that the values of communalities are higher in the method of PFA, which means that this model explains more variabilities of those variables that appear in the following chunk.

The variables best explained by the factors using method PFA are total cases per million, median age and gdp per capita (explained by more than 80% of variabilities).

```
# Communalities and uniquenesses
comm_pfa <- diag(M_pfa %*% t(M_pfa))
names(comm_pfa) <- colnames(Y)

sort(comm_pfa,decreasing=TRUE)
#>    total_cases_per_million                 median_age
#>                     0.8991                     0.8968
#>             gdp_per_capita              aged_65_older
#>                     0.8096                     0.7662
#>   total_deaths_per_million      cardiovasc_death_rate
#>                     0.7557                     0.7076
#>    human_development_index                 population
#>                     0.6739                     0.6112
#>        diabetes_prevalence      new_cases_per_million
#>                     0.6094                     0.5877
#>            extreme_poverty         population_density
#>                     0.4851                     0.4349
#>            life_expectancy hospital_beds_per_thousand
#>                     0.4186                     0.3551
#>          stringency_index
#>                     0.2286
sort(comm_pcfa,decreasing=TRUE)
#>    total_cases_per_million                 median_age
#>                     0.8962                     0.8845
#>             gdp_per_capita              aged_65_older
#>                     0.8418                     0.7929
#>   total_deaths_per_million      cardiovasc_death_rate
#>                     0.7904                     0.7711
#>    human_development_index                 population
#>                     0.7411                     0.7328
#>        diabetes_prevalence      new_cases_per_million
```

```
#>                   0.7081                   0.7015
#>        population_density            extreme_poverty
#>                   0.6104                   0.5808
#>           life_expectancy hospital_beds_per_thousand
#>                   0.5733                   0.4998
#>          stringency_index
#>                   0.4887
```

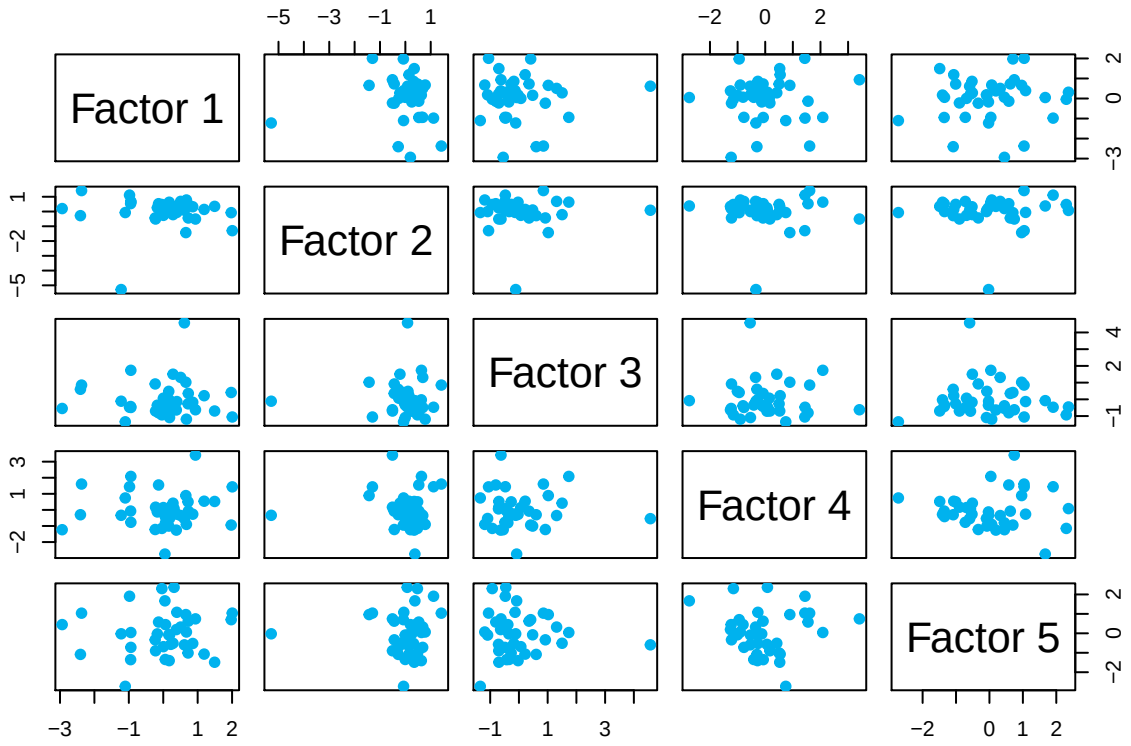
We can observe that the most of the values of uniqueness of the method PFA are slightly higher than the method PFCA in the following chunk.

The variables worst explained by the factors using method PFA are stringency index, hospital beds per thousand, life expectancy, population density and extreme poverty (less than 50%).

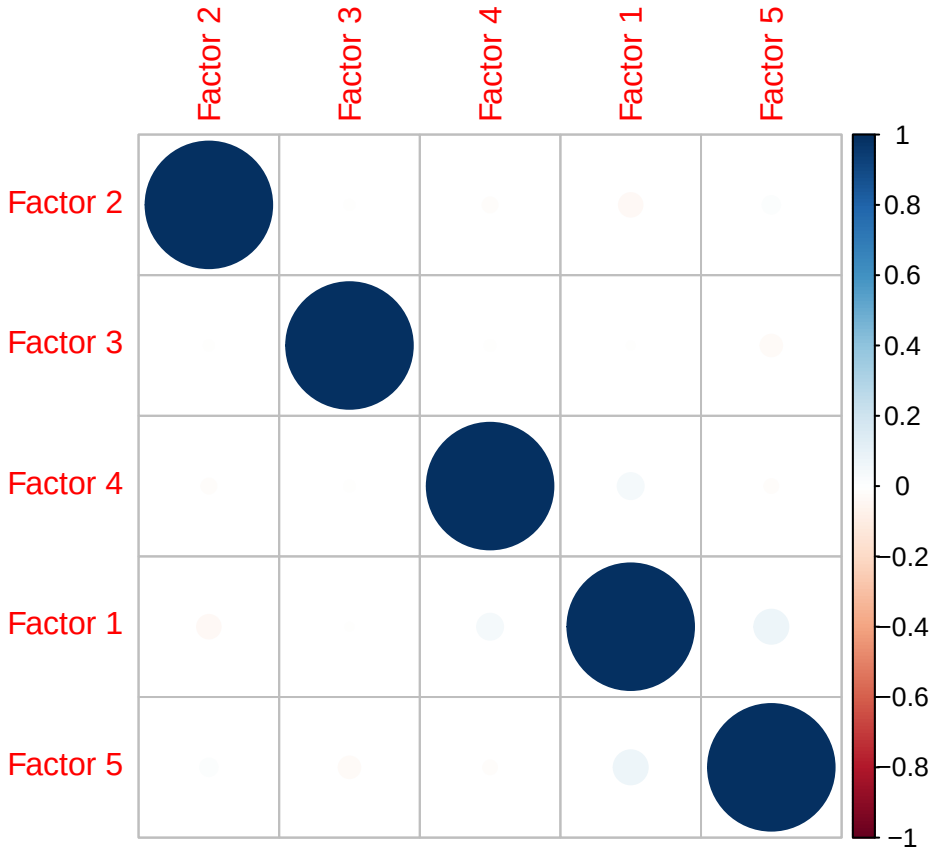
```
uniq_pfa <- diag(Sigma_nu_pfa)
names(uniq_pfa) <- names(comm_pfa)
sort(uniq_pfa,decreasing=TRUE)
#>          stringency_index hospital_beds_per_thousand
#>                   0.7714                   0.6449
#>           life_expectancy         population_density
#>                   0.5814                   0.5651
#>           extreme_poverty      new_cases_per_million
#>                   0.5149                   0.4123
#>       diabetes_prevalence                population
#>                   0.3906                   0.3888
#>    human_development_index      cardiovasc_death_rate
#>                   0.3261                   0.2924
#>   total_deaths_per_million              aged_65_older
#>                   0.2443                   0.2338
#>            gdp_per_capita                 median_age
#>                   0.1904                   0.1032
#>    total_cases_per_million
#>                   0.1009
sort(uniq_pcfa,decreasing=TRUE)
#>          stringency_index hospital_beds_per_thousand
#>                   0.5113                   0.5002
#>           life_expectancy            extreme_poverty
#>                   0.4267                   0.4192
#>         population_density      new_cases_per_million
#>                   0.3896                   0.2985
#>       diabetes_prevalence                population
#>                   0.2919                   0.2672
#>    human_development_index      cardiovasc_death_rate
#>                   0.2589                   0.2289
#>   total_deaths_per_million              aged_65_older
#>                   0.2096                   0.2071
#>            gdp_per_capita                 median_age
#>                   0.1582                   0.1155
#>    total_cases_per_million
#>                   0.1038
```

As for the method of PCFA, we can see here that the factors are uncorrelated.

```
# Estimate the factor scores
F_pfa <- Y %*% solve(Sigma_nu_pfa) %*% M_pfa %*% solve(t(M_pfa) %*% solve(Sigma_nu_pfa) %*% M_pfa)
colnames(F_pfa) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")
# See that the factors are uncorrelated
pairs(F_pfa,pch=19,col=color_1)
```

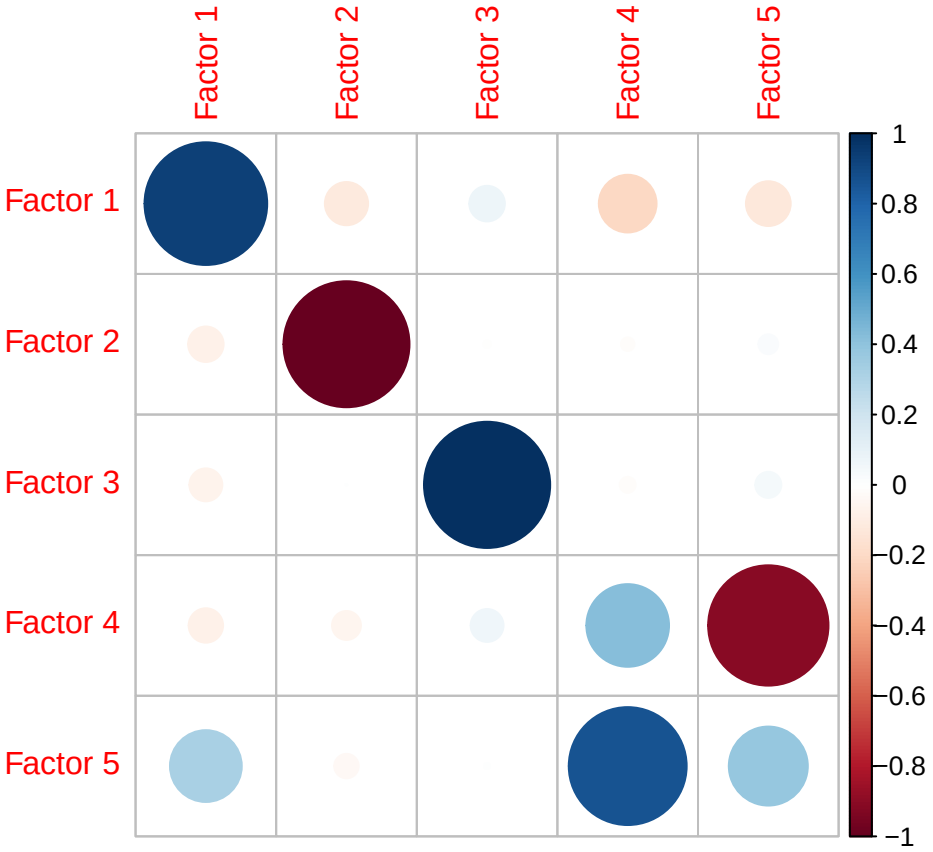



```
corrplot(cor(F_pfa),order="hclust")
```



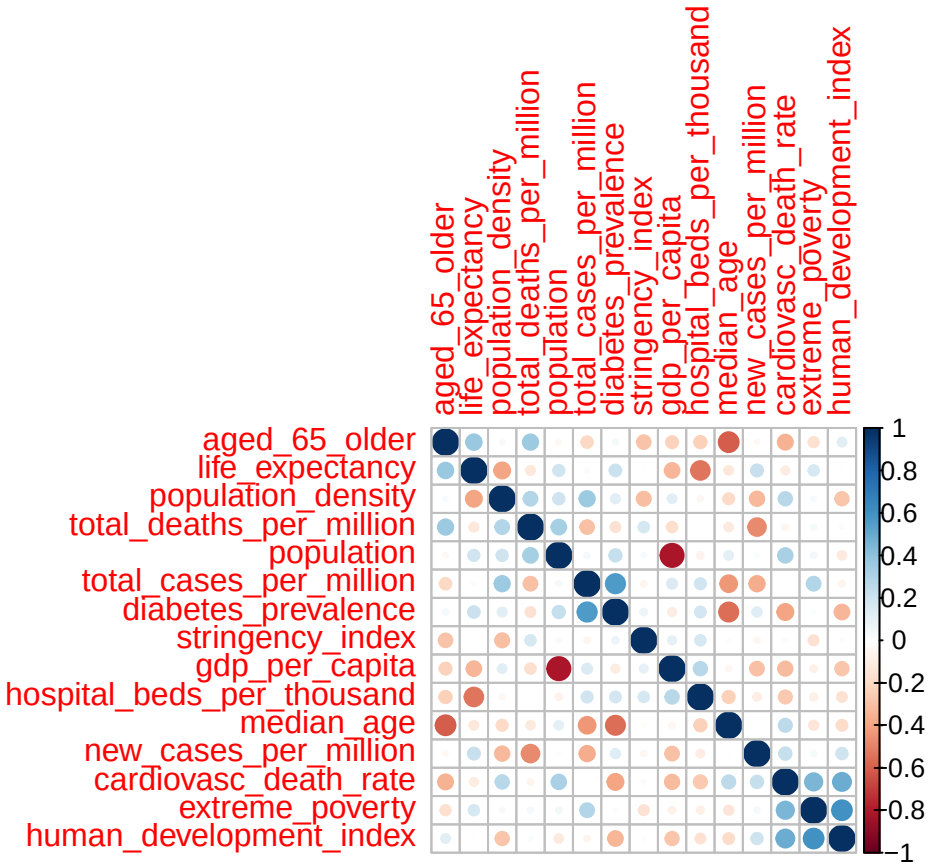
As the results that we have obtained before, we can see that the factors are quite similar except for the last two, they kind of exchanged the position.

```
# Obtain the correlation matrix between the PCFA and PFA estimates
cor(F_pcfa,F_pfa)
#>          Factor 1   Factor 2  Factor 3 Factor 4 Factor 5
#> Factor 1  0.93944 -0.1163977  0.078859 -0.20777 -0.12492
#> Factor 2 -0.07850 -0.9908598 -0.003416 -0.01125  0.02426
#> Factor 3 -0.06750  0.0001708  0.993806 -0.01562  0.04225
#> Factor 4 -0.07371 -0.0520921  0.067326  0.42635 -0.90357
#> Factor 5  0.32158 -0.0371333  0.002114  0.86756  0.38932
corrplot(cor(F_pcfa,F_pfa))
```



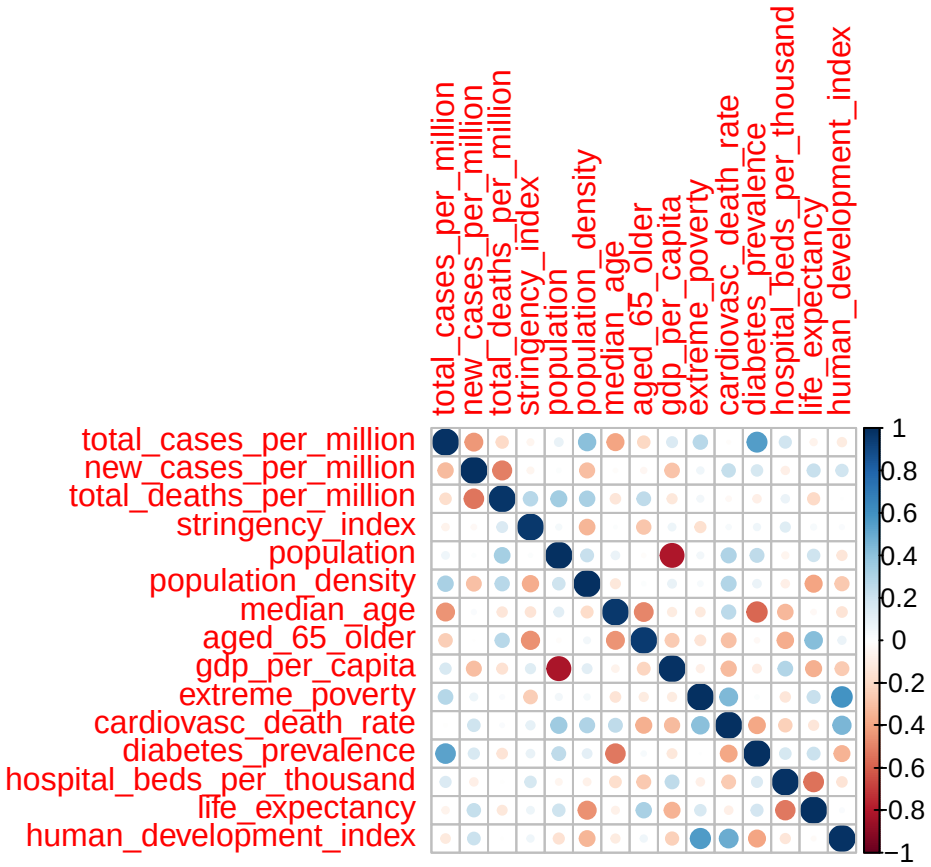
Now we plot the residuals. As before, the residuals show some some correlations that the model is not able to explain, e.g. the correlation between gdp per capita and population.

```
# Estimate the residuals
Nu_pfa <- Y - F_pfa %*% t(M_pfa)
corrplot(cor(Nu_pfa),order="hclust")
```



Here we obtain the correlation matrix of residuals between the PCFA and PFA estimates, we can see that they are highly correlated.

```
# Obtain the correlation matrix between the PCFA and PFA estimates
corrplot(cor(Nu_pcfa,Nu_pfa))
```



Here we are trying to find the optimal number of factors that we should consider, but we can see that for 5 factors null hypothesis is not rejected, so we choose five factors as what we have done before.

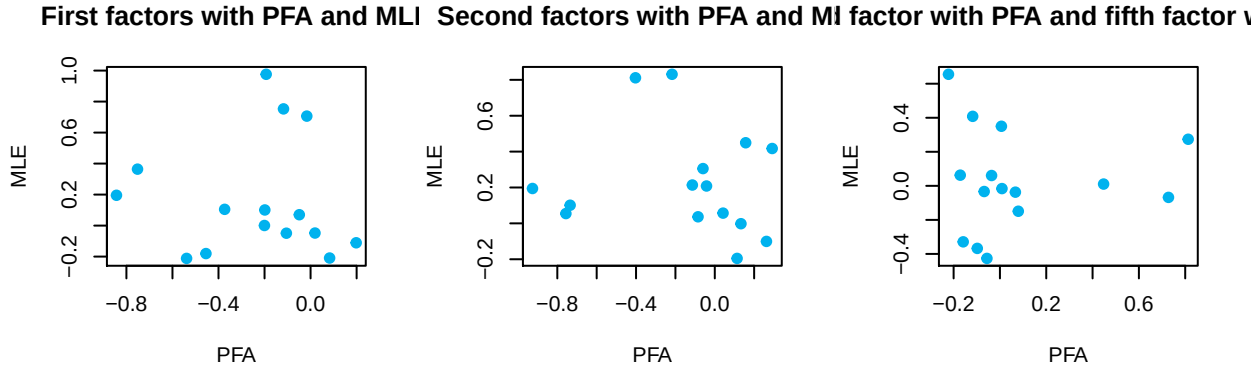
```
# Maximum likelihood estimation

# Start with one factor
Y_mle_1 <- factanal(Y,factors=1,rotation="varimax",scores="Bartlett")
Y_mle_1$STATISTIC
#> objective
#>     177.1
Y_mle_1$PVAL
#> objective
#> 1.226e-07

# two factors
Y_mle_2 <- factanal(Y,factors=2,rotation="varimax",scores="Bartlett")
Y_mle_2$STATISTIC
#> objective
#>     125.4
Y_mle_2$PVAL
#> objective
#>  0.000312

# three factors
Y_mle_3 <- factanal(Y,factors=3,rotation="varimax",scores="Bartlett")
Y_mle_3$STATISTIC
#> objective
#>     96.91
Y_mle_3$PVAL
#> objective
#>  0.003899

# four factors
Y_mle_4 <- factanal(Y,factors=4,rotation="varimax",scores="Bartlett")
```

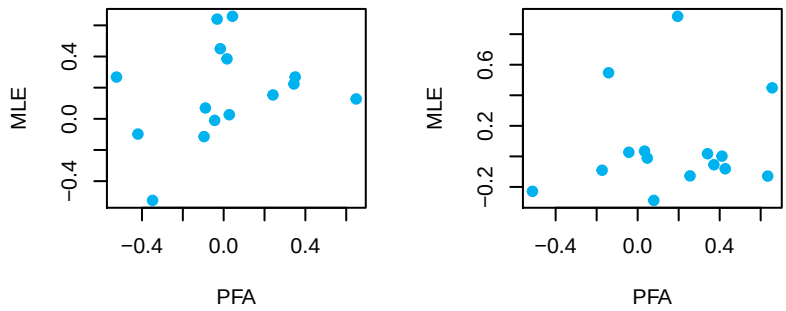
```
Y_mle_4$STATISTIC
#> objective
#>     72.16
Y_mle_4$PVAL
#> objective
#>   0.02714

# five factors
Y_mle_5 <- factanal(Y,factors=5,rotation="varimax",scores="Bartlett")
Y_mle_5$STATISTIC
#> objective
#>     46.53
Y_mle_5$PVAL
#> objective
#>    0.2214
```

```
# Get the loading matrix
M_mle <- loadings(Y_mle_5)[1:p,1:r]
```

We can see here that the factors are very different. So we plot the weights of each factor.

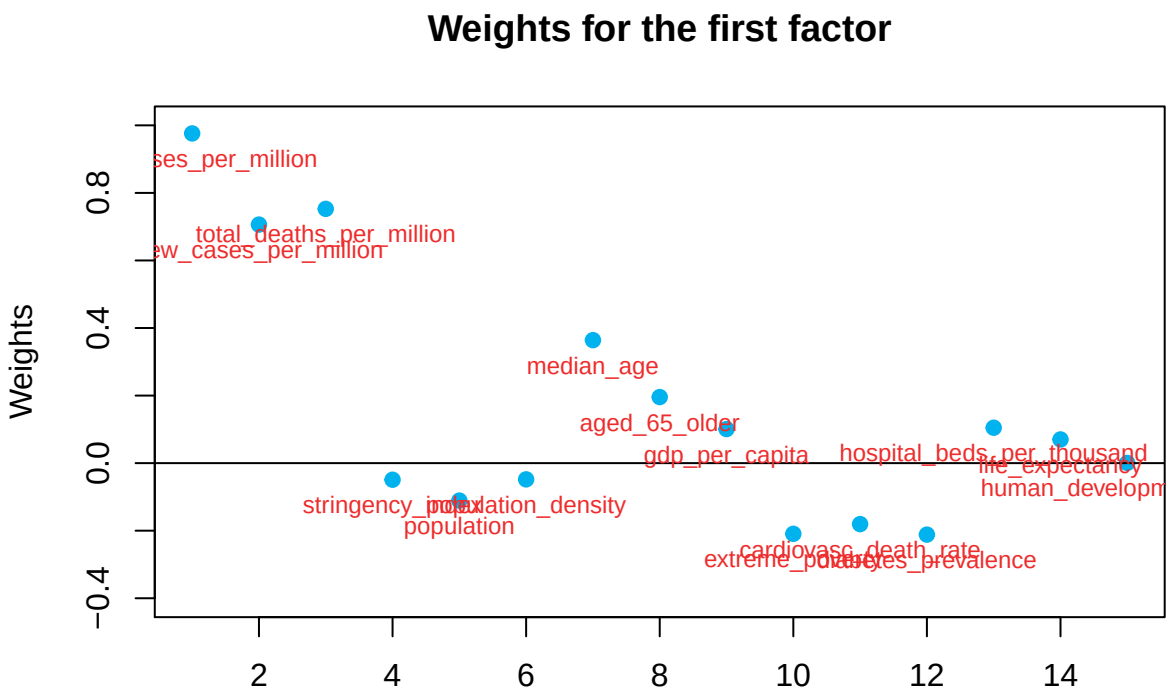
```
# Compare with PFA estimates
par(mfrow=c(2,3))
plot(M_pfa[,1],M_mle[,1],pch=19,col="deepskyblue2",main="First factors with PFA and MLE",xlab="PFA",ylab="MLE")
plot(M_pfa[,2],M_mle[,2],pch=19,col="deepskyblue2",main="Second factors with PFA and MLE",xlab="PFA",ylab="MLE")
plot(M_pfa[,3],M_mle[,5],pch=19,col="deepskyblue2",main="Third factor with PFA and fifth factor with MLE",xlab="PFA",ylab="MLE")
plot(M_pfa[,4],M_mle[,3],pch=19,col="deepskyblue2",main="Fourth factor with PFA and third factor with MLE",xlab="PFA",ylab="MLE"
plot(M_pfa[,5],M_mle[,4],pch=19,col="deepskyblue2",main="Fifth factors with PFA and fourth factor with MLE",xlab="PFA",ylab="MLE
```





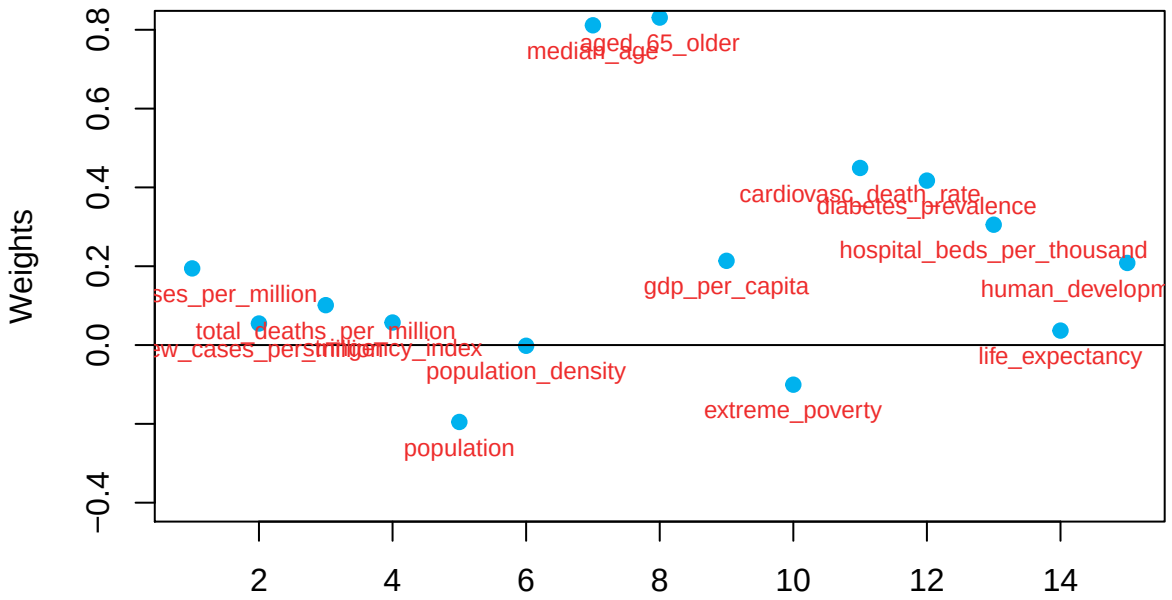
We can observe here that the first factor appears to be an index that explains the situation of pandemic for each country.

```
plot(1:p,M_mle[,1],pch=19,col=color_1,xlab="",ylim=c(-0.4,1),ylab="Weights",main="Weights for the first factor")
abline(h=0)
text(1:p,M_mle[,1],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



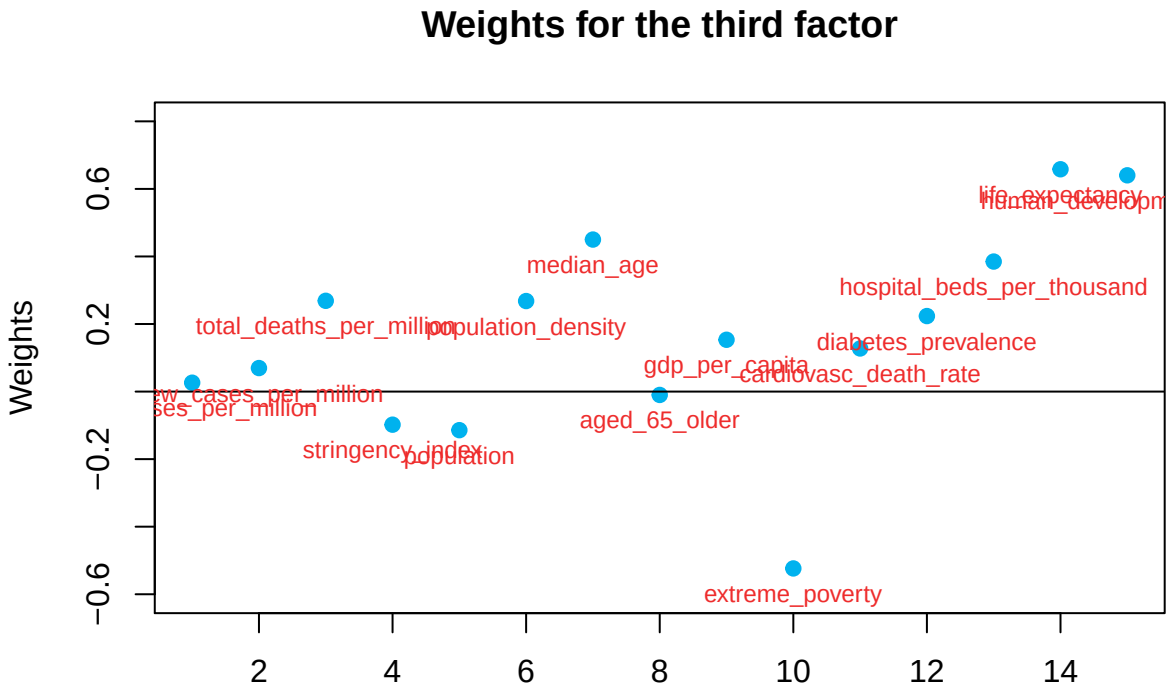
However, the second factor seems to be an index of age.

```
plot(1:p,M_mle[,2],pch=19,col=color_1,xlab="",ylim=c(-0.4,0.8),ylab="Weights",main="Weights for the second factor")
abline(h=0)
text(1:p,M_mle[,2],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



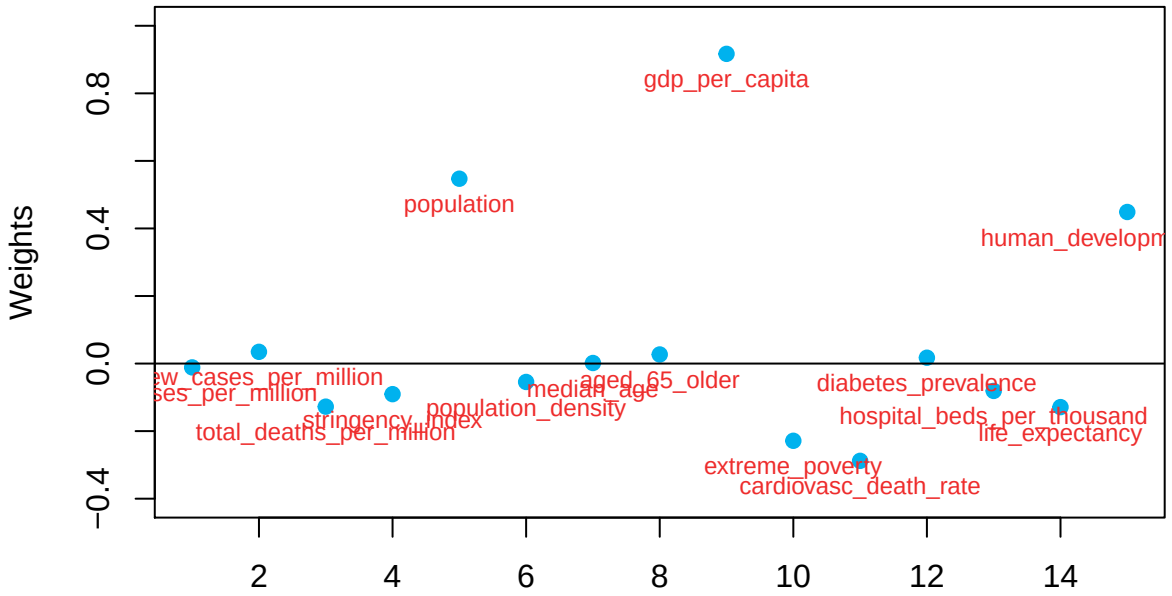
The third factor appears to be an index of development, as it has high positive weights for life expectancy and HDI and low negative values of extreme poverty.

```
plot(1:p,M_mle[,3],pch=19,col=color_1,xlab="",ylim=c(-0.6,0.8),ylab="Weights",main="Weights for the third factor")
abline(h=0)
text(1:p,M_mle[,3],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



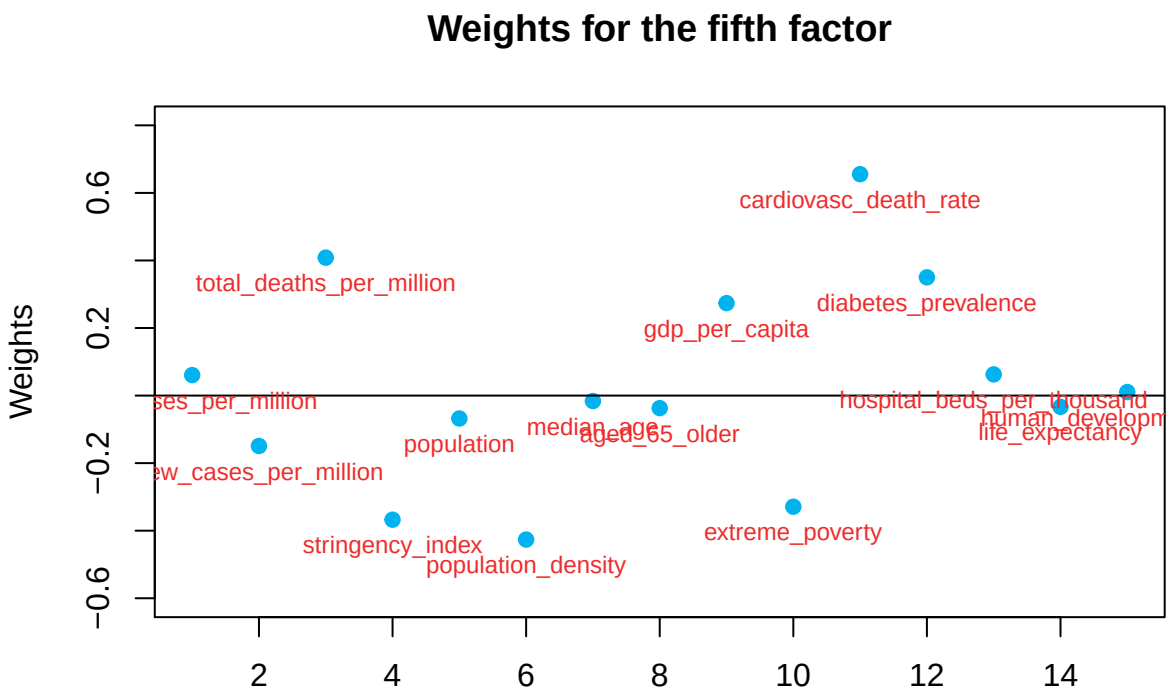
The fourth factor seems to be an index of how developed a country is.

```
plot(1:p,M_mle[,4],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.4,1),main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_mle[,4],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



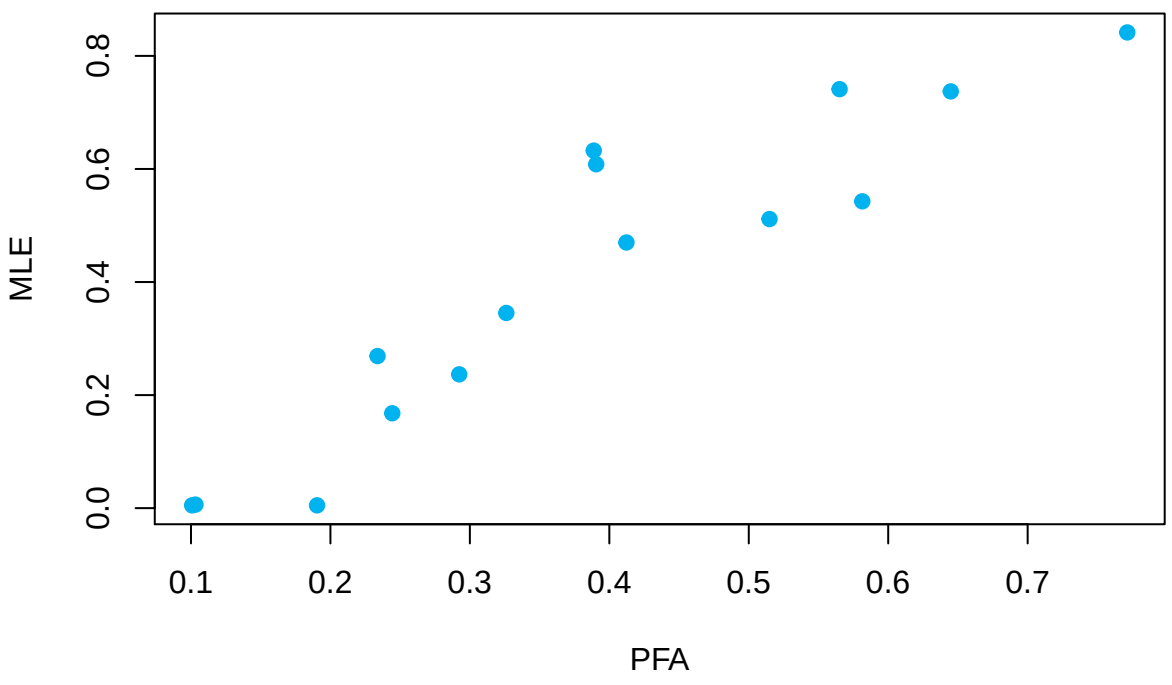
The fifth factor appears to be an index of a mixture between all the variables that we cannot understand.

```
plot(1:p,M_mle[,5],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.6,0.8),main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_mle[,5],labels=colnames(XX_low),pos=1,col=color_5,cex=0.75)
```



By the following plot we can observe that there are only some small differences between the covariance matrix of errors between MLE and PFA.

```
# Estimate the covariance matrix of the errors
Sigma_nu_mle <- diag(diag(cov(Y) - M_mle %*% t(M_mle)))

# Compare with the estimate with the PFA method
par(mfrow=c(1,1))
plot(diag(Sigma_nu_pfa),diag(Sigma_nu_mle),pch=19,col=color_1,main="Noise variances with PFA and MLE",
     xlab="PFA",ylab="MLE")
```



There are changes in the sorting, the communalities are quite different from the method PFA.

The variables best explained by the factors are gdp per capita, total cases per million and median age (almost 100% explained).

```
# Communalities and uniquenesses
comm_mle <- diag(M_mle %*% t(M_mle))
names(comm_mle) <- colnames(Y)
sort(comm_mle,decreasing=TRUE)
#>             gdp_per_capita    total_cases_per_million
#>                     0.9950                     0.9950
#>                 median_age   total_deaths_per_million
#>                     0.9937                     0.8322
#>      cardiovasc_death_rate              aged_65_older
#>                     0.7633                     0.7310
#>    human_development_index      new_cases_per_million
#>                     0.6547                     0.5300
#>            extreme_poverty            life_expectancy
#>                     0.4886                     0.4572
#>        diabetes_prevalence                 population
#>                     0.3917                     0.3676
#> hospital_beds_per_thousand         population_density
#>                     0.2628                     0.2588
#>           stringency_index
#>                     0.1585
sort(comm_pfa,decreasing=TRUE)
#>    total_cases_per_million                 median_age
#>                     0.8991                     0.8968
#>             gdp_per_capita              aged_65_older
#>                     0.8096                     0.7662
#>   total_deaths_per_million      cardiovasc_death_rate
#>                     0.7557                     0.7076
#>    human_development_index                 population
#>                     0.6739                     0.6112
#>        diabetes_prevalence      new_cases_per_million
#>                     0.6094                     0.5877
#>            extreme_poverty         population_density
#>                     0.4851                     0.4349
#>            life_expectancy hospital_beds_per_thousand
#>                     0.4186                     0.3551
#>           stringency_index
#>                     0.2286
```

There are also changes in the sorting.

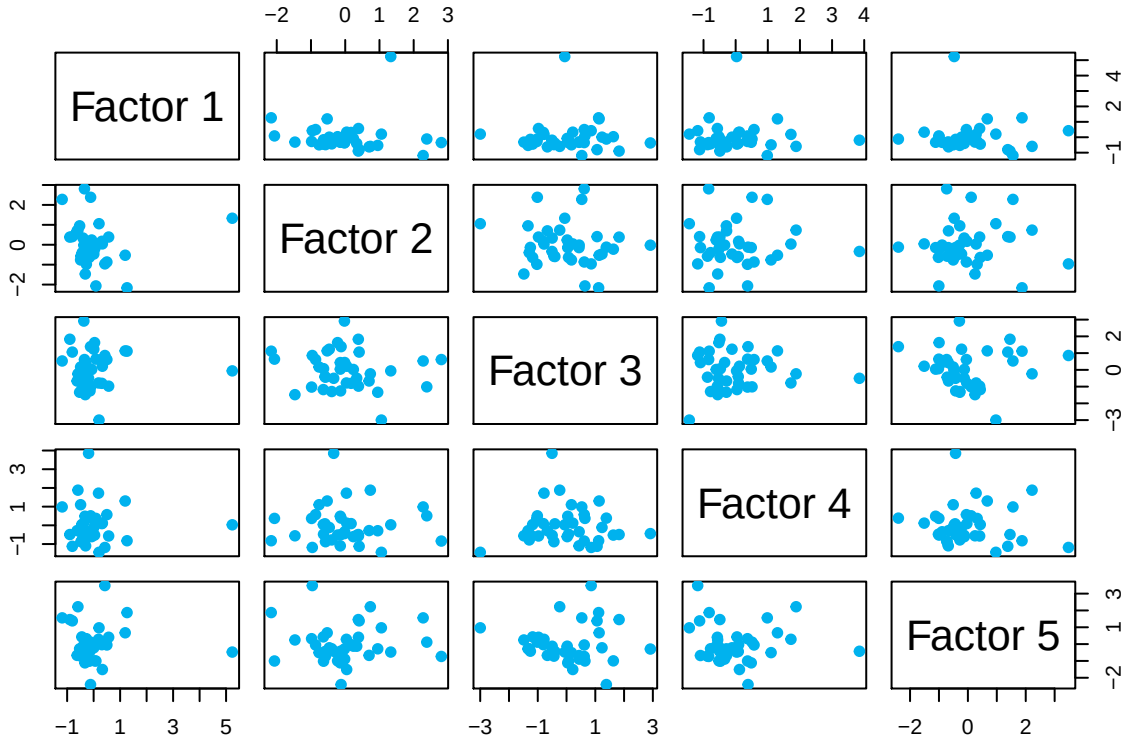
The variables worst explained by the factors are stringency index, population density and hospital beds per thousand.

```
uniq_mle <- diag(Sigma_nu_mle)
names(uniq_mle) <- names(comm_mle)
sort(uniq_mle,decreasing=TRUE)
#>           stringency_index         population_density
#>                   0.841492                   0.741169
#> hospital_beds_per_thousand                 population
#>                   0.737154                   0.632436
#>        diabetes_prevalence            life_expectancy
#>                   0.608302                   0.542791
#>            extreme_poverty      new_cases_per_million
#>                   0.511384                   0.469959
#>    human_development_index              aged_65_older
#>                   0.345269                   0.269019
#>      cardiovasc_death_rate   total_deaths_per_million
#>                   0.236700                   0.167833
#>                 median_age    total_cases_per_million
#>                   0.006277                   0.004993
#>             gdp_per_capita
#>                   0.004989
sort(uniq_pfa,decreasing=TRUE)
#>           stringency_index hospital_beds_per_thousand
#>                     0.7714                     0.6449
```

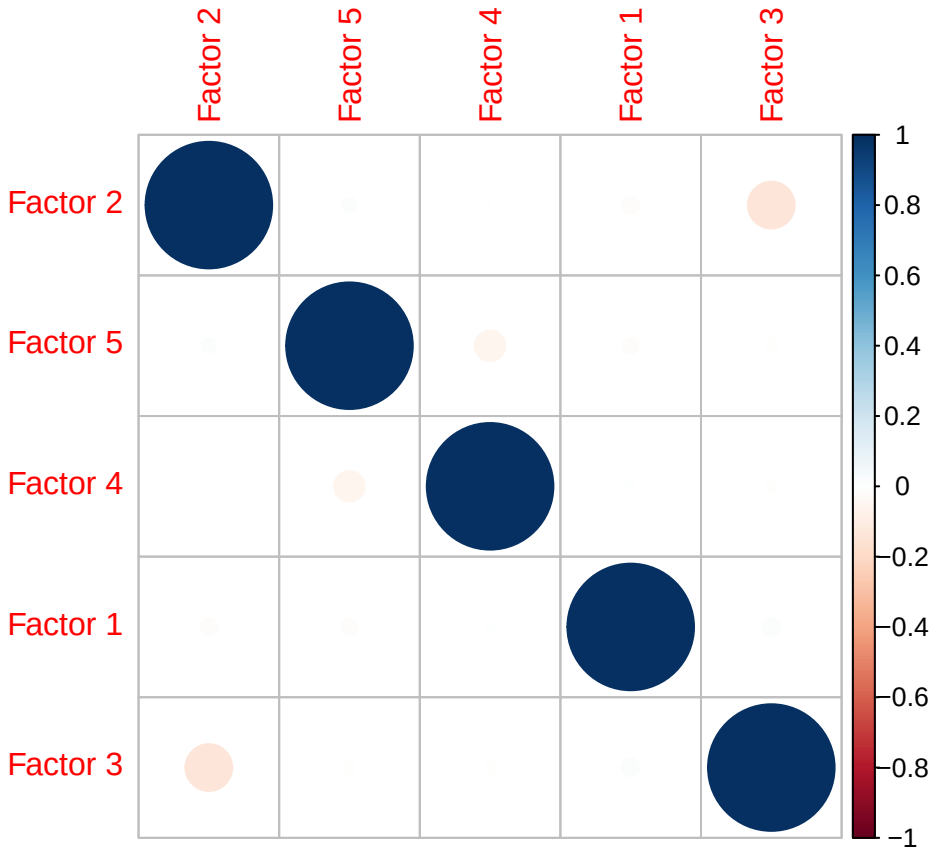
```
#>            life_expectancy         population_density
#>                     0.5814                     0.5651
#>            extreme_poverty       new_cases_per_million
#>                     0.5149                     0.4123
#>        diabetes_prevalence                 population
#>                     0.3906                     0.3888
#>    human_development_index       cardiovasc_death_rate
#>                     0.3261                     0.2924
#>   total_deaths_per_million             aged_65_older
#>                     0.2443                     0.2338
#>             gdp_per_capita                 median_age
#>                     0.1904                     0.1032
#>    total_cases_per_million
#>                     0.1009
```

The factors are uncorrelated in this method as as in other two methods.

```
# Estimate the factor scores
F_mle <- Y %*% solve(Sigma_nu_mle) %*% M_mle %*% solve(t(M_mle) %*% solve(Sigma_nu_mle) %*% M_mle)
colnames(F_mle) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")
pairs(F_mle,pch=19,col="deepskyblue2")
```

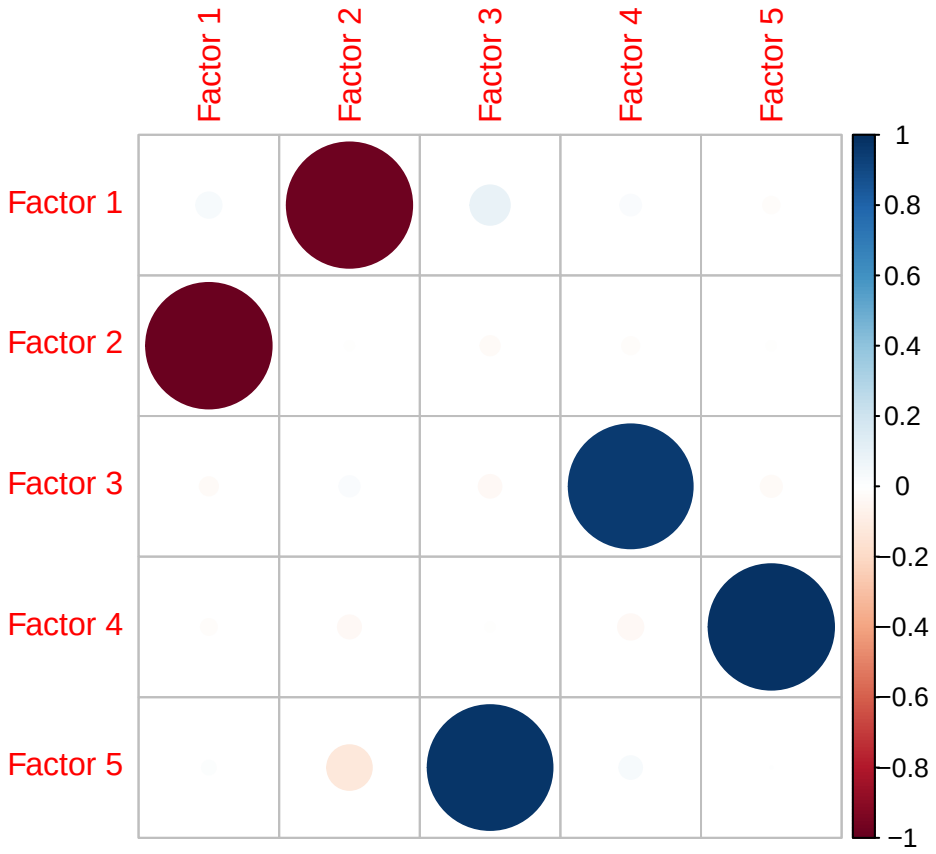



```
corrplot(cor(F_mle),order="hclust")
```



The correlations are not the same, the have exchanged the positions.

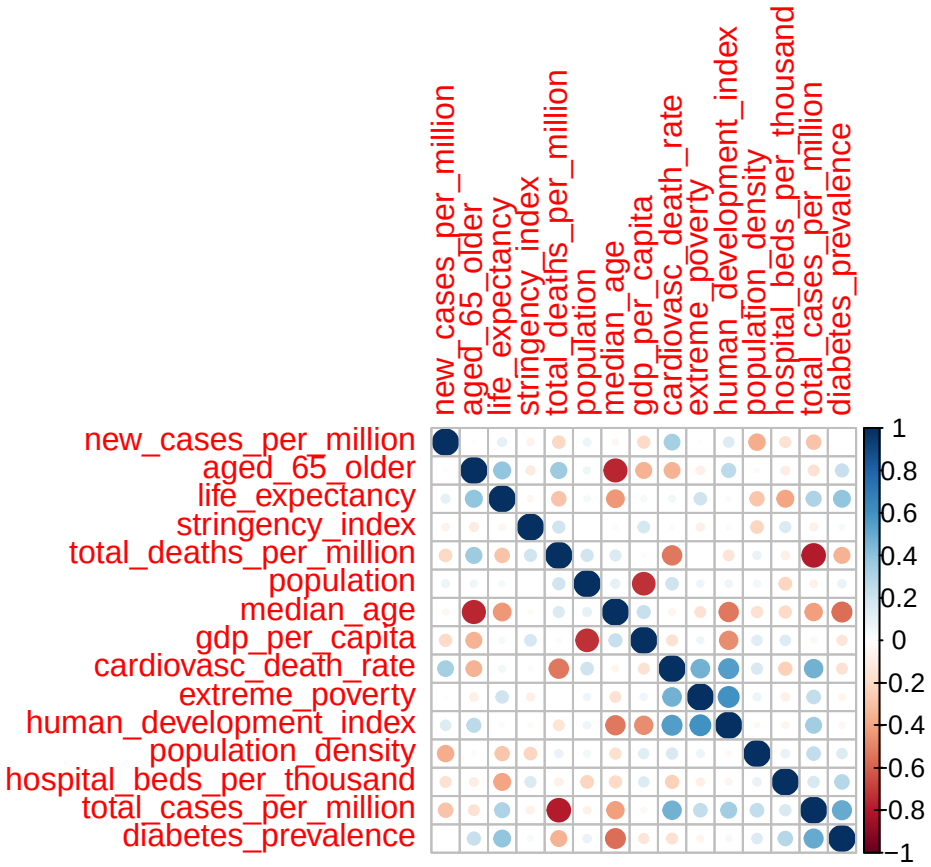
```
# Obtain the correlation matrix between the PFA and MLE estimates
cor(F_pfa,F_mle)
#>          Factor 1  Factor 2 Factor 3 Factor 4   Factor 5
#> Factor 1  0.03988 -0.979853  0.09666  0.02681 -0.0169512
#> Factor 2 -0.98335 -0.005228 -0.02297 -0.01835 -0.0054662
#> Factor 3 -0.02057  0.025508 -0.03206  0.95811 -0.0279128
#> Factor 4 -0.01570 -0.033152 -0.00536 -0.03989  0.9815884
#> Factor 5  0.01185 -0.124078  0.97172  0.03202  0.0001983
corrplot(cor(F_pfa,F_mle))
```



As before, the residuals show some correlations that the model is not able to explain, but we can see that the correlations in this method are much higher than the previous methods, it seems to be the worst method if we use the residual criterion to rank.

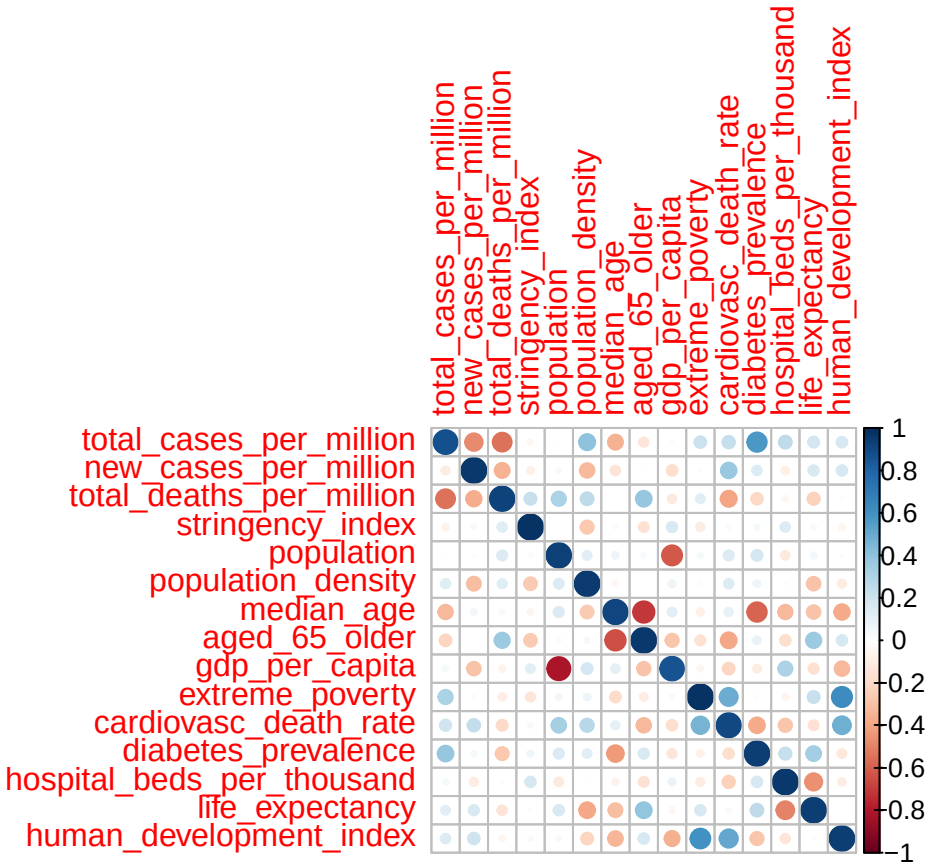
So we will only use the first method for the countries with medium HDI, high HDI and very high HDI.

```
# Estimate the residuals
Nu_mle <- Y - F_mle %*% t(M_mle)
corrplot(cor(Nu_mle),order="hclust")
```



Now we obtain the correlation matrix between the PFA and MLE estimates.

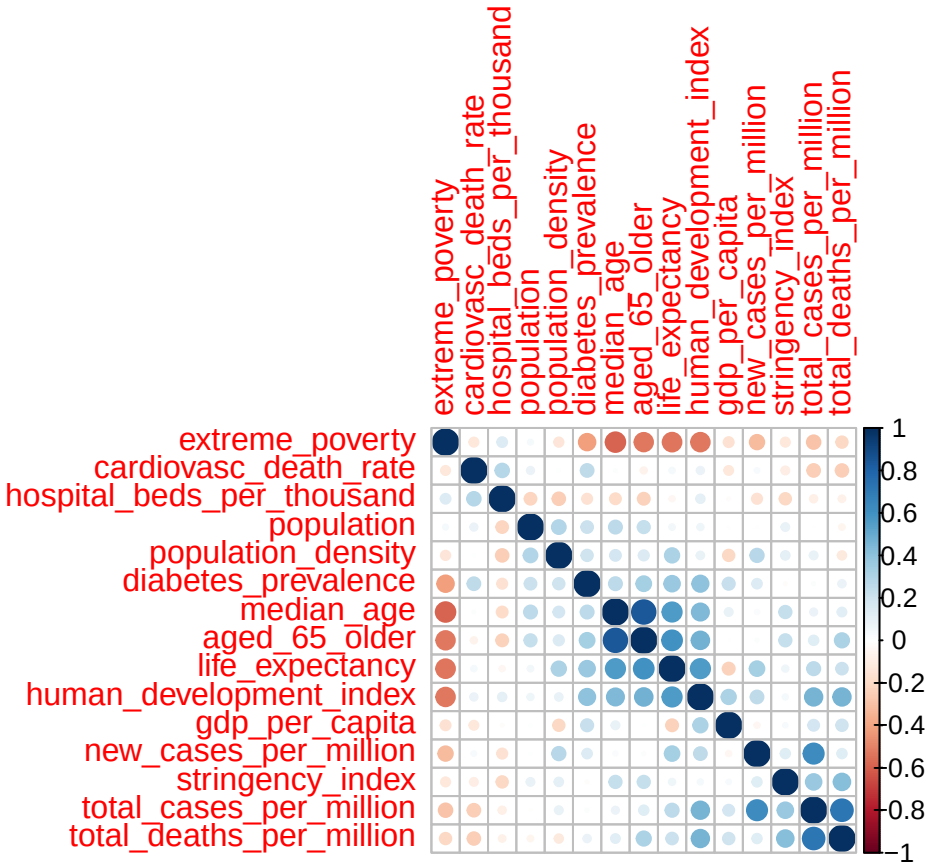
```
# Obtain the correlation matrix between the PFA and MLE estimates
Nu_pfa <- Y - F_pfa %*% t(M_pfa)
corrplot(cor(Nu_pfa,Nu_mle))
```



2. Medium HDI:

The variables are sorted as before using their correlations, and we can observe that there are some groups of variables that are highly correlated. For instance, we have a group of variables which is relatad to how developed a country is: median age, aged 65 or older, life expectancy and human development index, and these variables are negatively correlated to extreme poverty; and another group of variables that are highly correlated are total cases per million and total deaths per million.

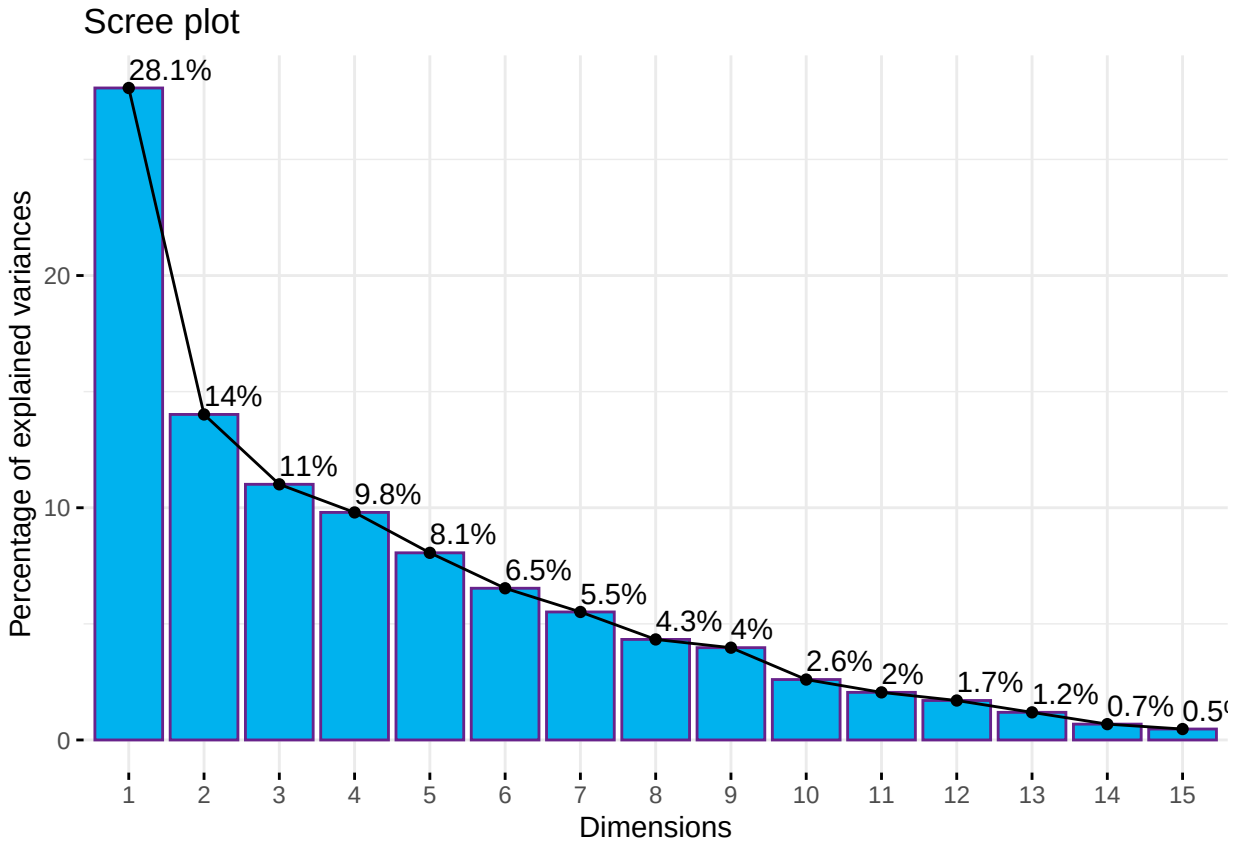
```
# Sample size and dimension of the personality data set
n <- nrow(XX_medium)
p <- ncol(XX_medium)

corrplot(cor(XX_medium),order="hclust")
```




```
# There are groups of correlated variables that may suggest a factor structure
```

Checking that the variance explained by each principal component, e.g. the first principal component explains 28.1% of the total variability and the secon principal component explains 14%.

```
# Principal Component Factor Analysis
# Obtain the PCs of the univariate standardized variables
Y <- scale(XX_medium)
Y_pcs <- prcomp(Y)
# Screeplot with all the eigenvalues
fviz_eig(Y_pcs,ncp=p,addlabels=T,barfill=color_1,barcolor=color_4)
```



Now we check the eigenvalues and the cumulative and the cumulative percentage of explained variance, and we will take the 5 principal components, we will be using the 5/15, which is 33% of the variables and will be keeping the 70.96% of the total information.

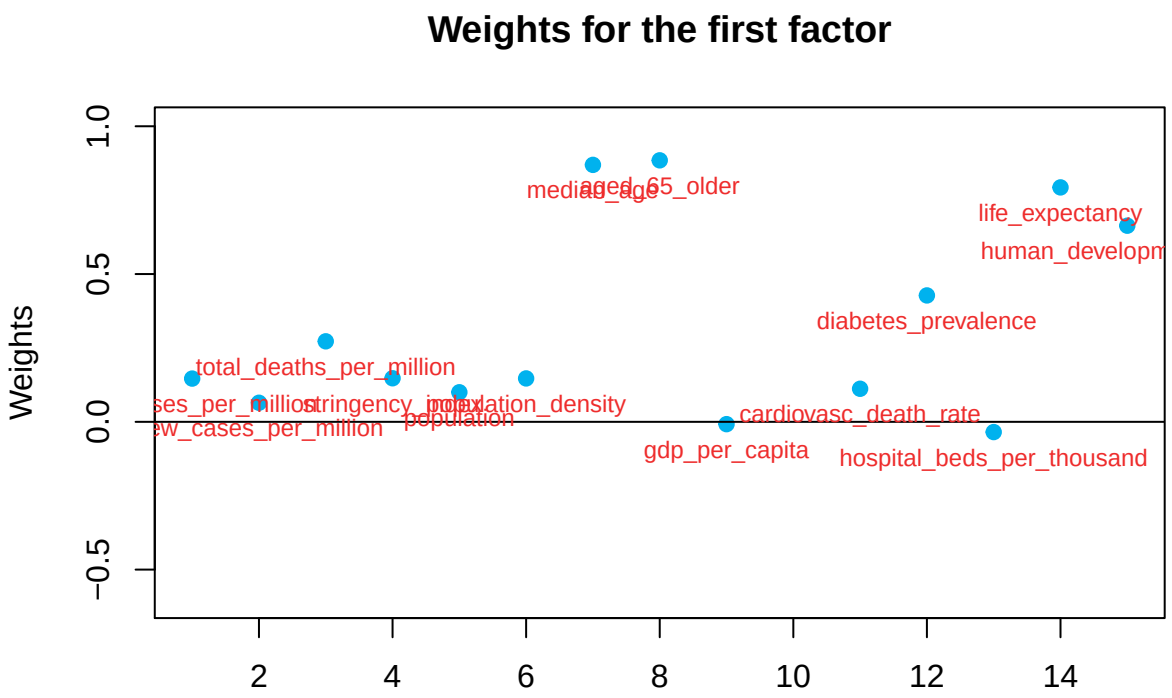
```
get_eigenvalue(Y_pcs)
#>        eigenvalue variance.percent cumulative.variance.percent
#> Dim.1     4.21127          28.0751                       28.08
#> Dim.2     2.10262          14.0175                       42.09
#> Dim.3     1.65127          11.0085                       53.10
#> Dim.4     1.46971           9.7981                       62.90
#> Dim.5     1.20889           8.0593                       70.96
#> Dim.6     0.97988           6.5326                       77.49
#> Dim.7     0.82686           5.5124                       83.00
#> Dim.8     0.64940           4.3293                       87.33
#> Dim.9     0.59609           3.9739                       91.31
#> Dim.10    0.39024           2.6016                       93.91
#> Dim.11    0.30742           2.0495                       95.96
#> Dim.12    0.25496           1.6997                       97.66
#> Dim.13    0.17842           1.1895                       98.85
#> Dim.14    0.10232           0.6822                       99.53
#> Dim.15    0.07064           0.4709                      100.00
```

From now on, let us focus on the first five PCs.

```
r <- 5
# Estimate the matrix M and use the varimax rotation for interpretability
M_pcfa <- Y_pcs$rotation[,1:r] %*% diag(Y_pcs$sdev[1:r])
M_pcfa <- varimax(M_pcfa)
M_pcfa <- loadings(M_pcfa)[1:p,1:r]
```

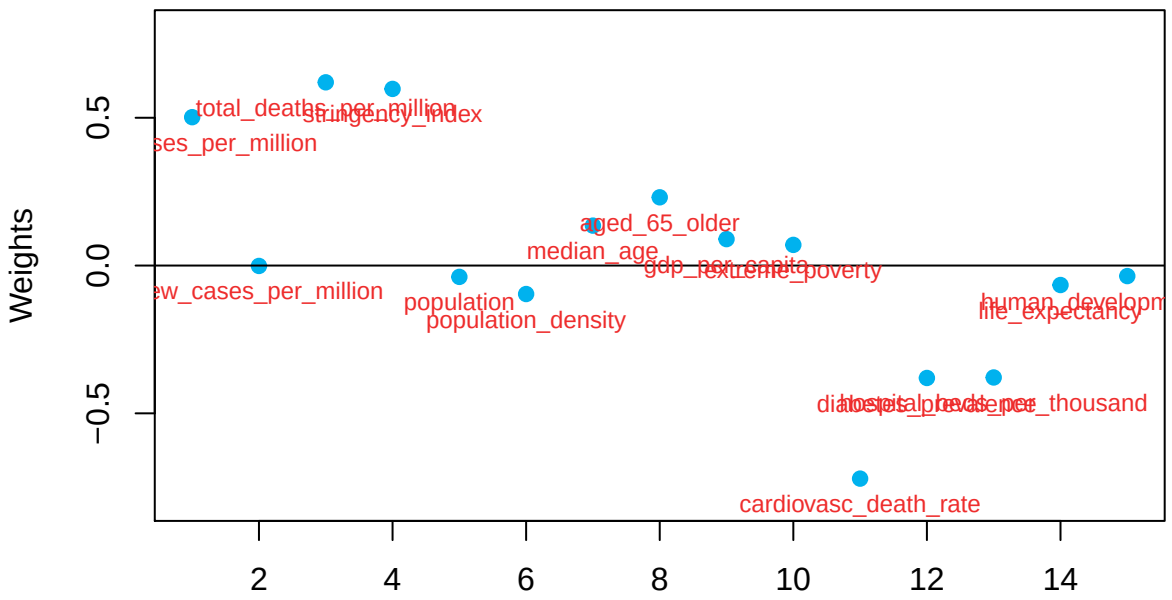
We can observe here that the first factor appears to be an index of how developed a country is, as we can see that the median age, aged 65 or older, life expectancy and HDI have very high positive weights.

```
plot(1:p,M_pcfa[,1],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.6,1),main="Weights for the first factor")
abline(h=0)
text(1:p,M_pcfa[,1],labels=colnames(XX_medium),pos=1,col=color_5,cex=0.75)
```



However, the second factor seems to be an index related to the situation of covid of each country as the variables total cases, total deaths and stringency index are very highly weighted, alse it combines the index with some negative value of cardiovascular death and diabetes prevalence.

```
plot(1:p,M_pcfa[,2],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.8,0.8),main="Weights for the second factor")
abline(h=0)
text(1:p,M_pcfa[,2],labels=colnames(XX_medium),pos=1,col=color_5,cex=0.75)
```

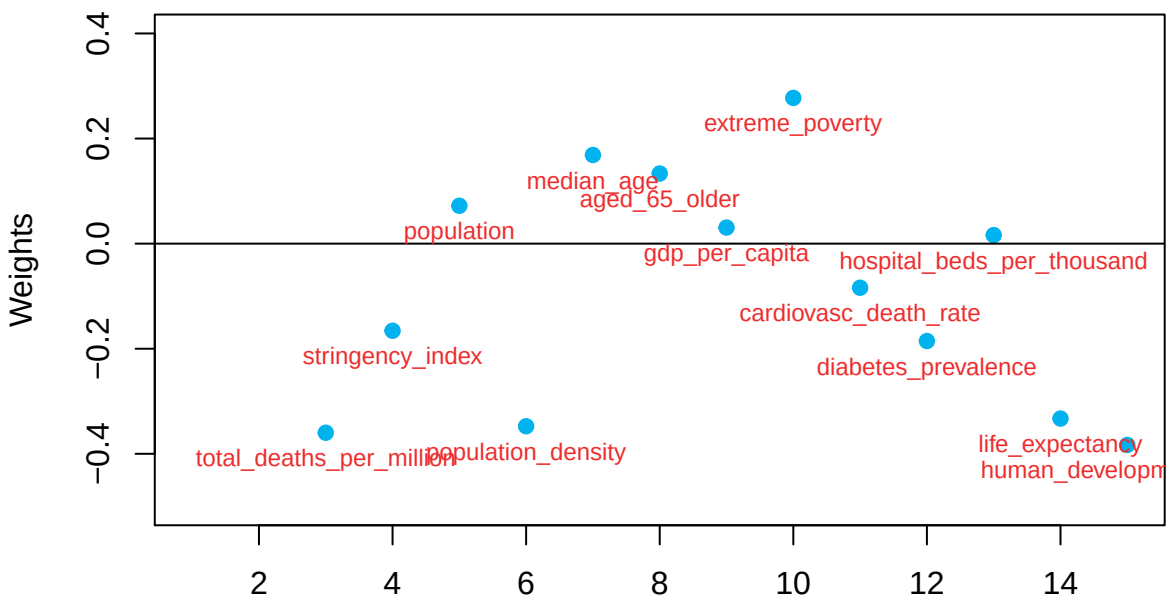


The third factor appears to be another index of population with some negative value of hospital beds per thousand.

```
plot(1:p,M_pcfa[,3],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.8,0.8),main="Weights for the third factor")
abline(h=0)
text(1:p,M_pcfa[,3],labels=colnames(XX_medium),pos=1,col=color_5,cex=0.75)
```

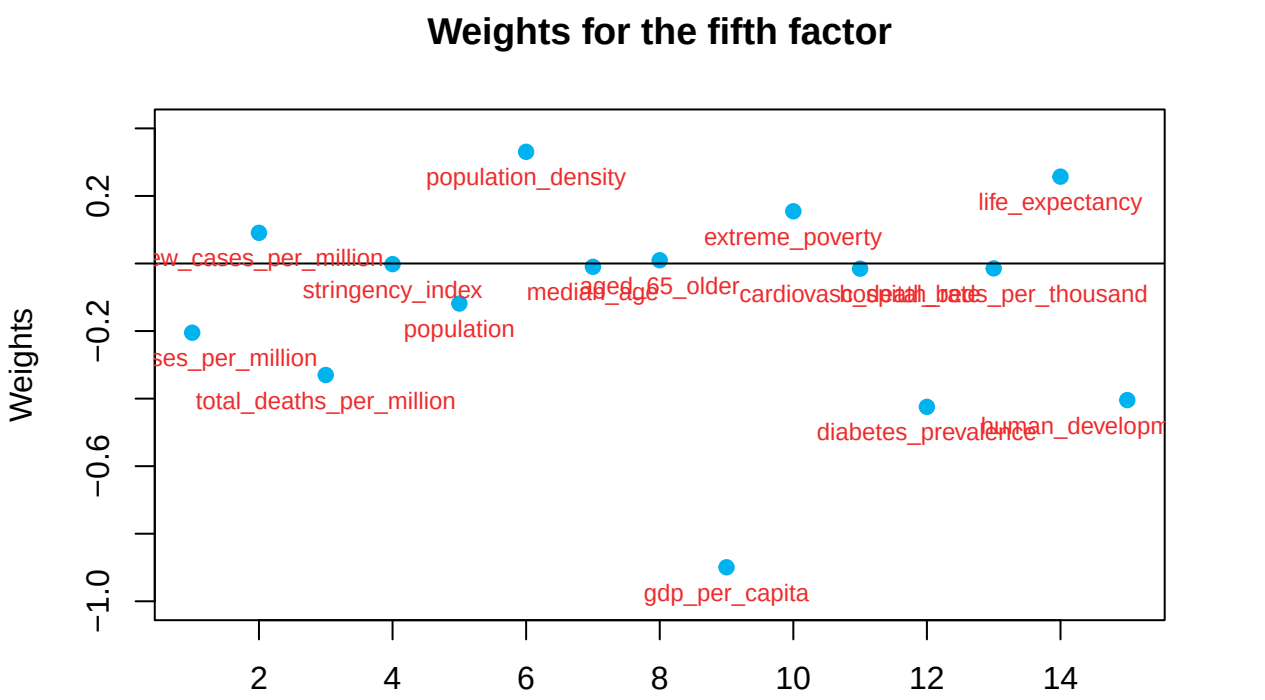
The fourth factor seems to be an index of how undeveloped a country is.

```
plot(1:p,M_pcfa[,4],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.5,0.4),main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_pcfa[,4],labels=colnames(XX_medium),pos=1,col=color_5,cex=0.75)
```



The fifth factor appears to be an index of how undeveloped a country is (with negative value of gdp per capita).

```
plot(1:p,M_pcfa[,5],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,0.4),main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_pcfa[,5],labels=colnames(XX_medium),pos=1,col=color_5,cex=0.75)
```

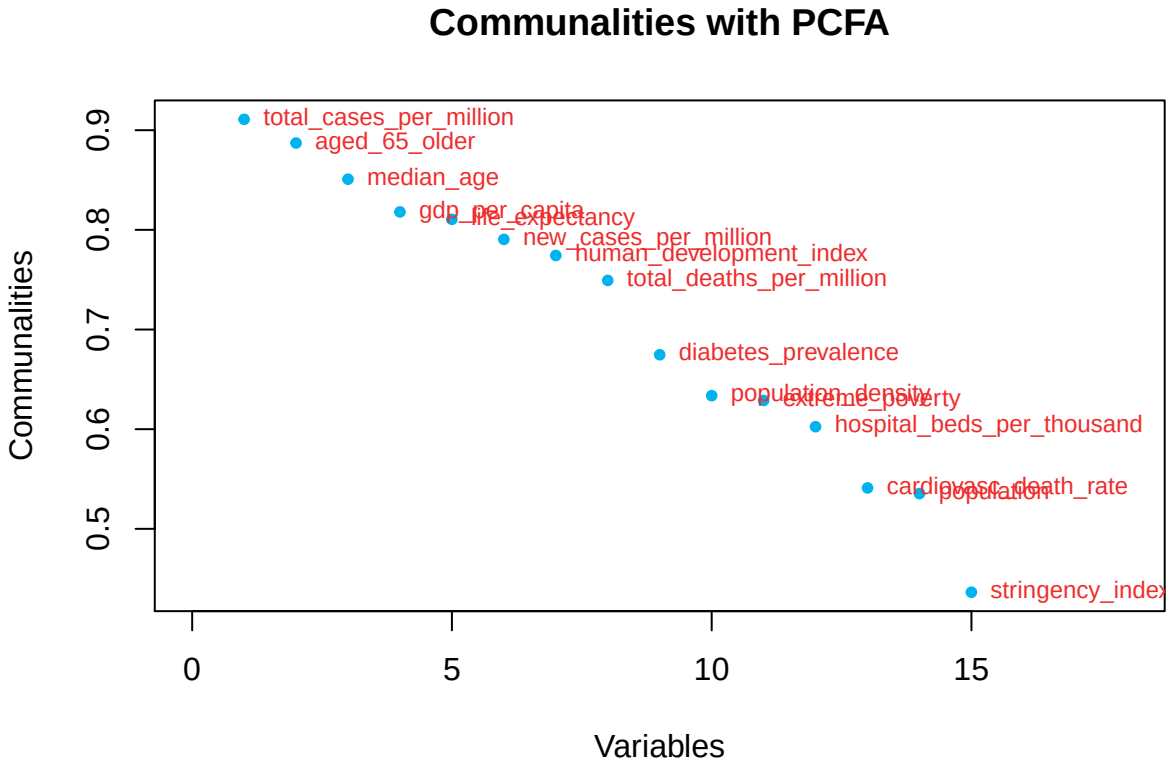



```
# Estimate the covariance matrix of the errors
Sigma_nu_pcfa <- diag(diag(cov(Y) - M_pcfa %*% t(M_pcfa)))
```

Here we plot the values of communalities and we can observe that the aspects of countries that are better explained by the factors are total covid cases per million, median age and gdp per capita (more or less 90%). And the variables that are not explained that well isstringency index (50% or below).

This values of communalities are very similar comparing to the countries with low HDI, although a lit bit higher.

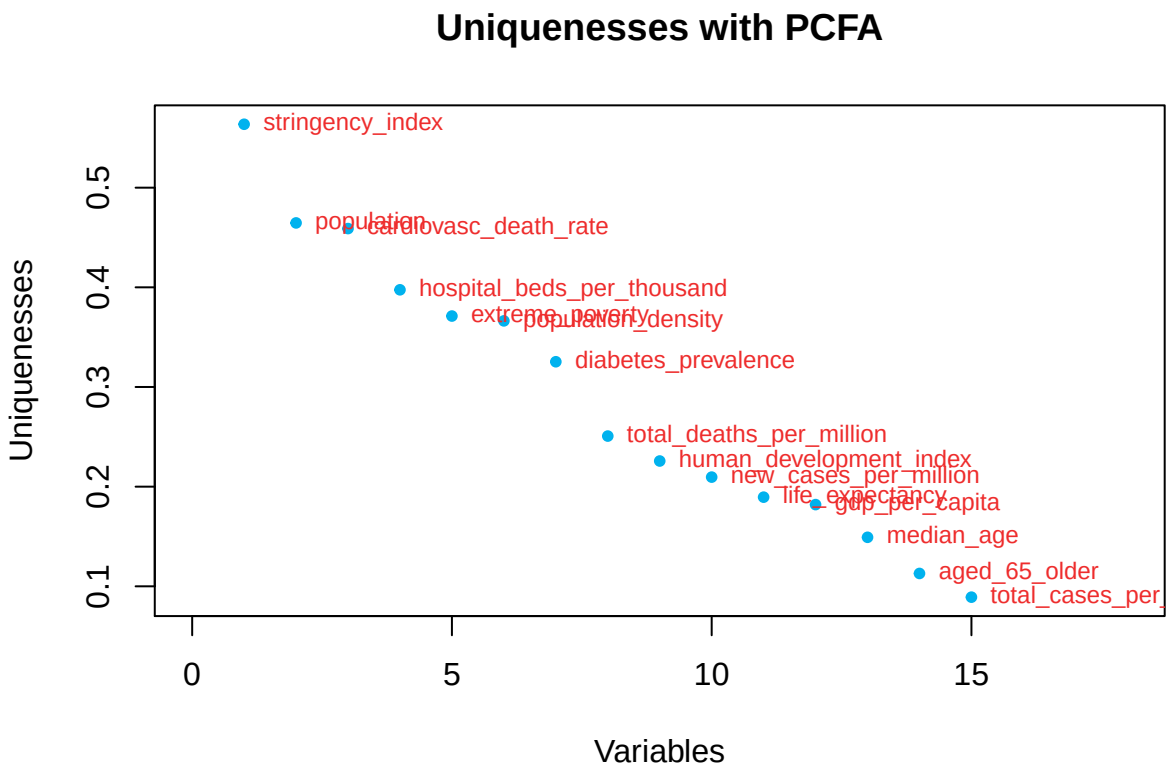
```
# Communalities and uniquenesses
comm_pcfa <- diag(M_pcfa %*% t(M_pcfa))
comm_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.9109                     0.7904
#>   total_deaths_per_million          stringency_index
#>                     0.7492                     0.4364
#>                 population         population_density
#>                     0.5354                     0.6336
#>                 median_age              aged_65_older
#>                     0.8508                     0.8871
#>            gdp_per_capita           extreme_poverty
#>                     0.8180                     0.6288
#>      cardiovasc_death_rate        diabetes_prevalence
#>                     0.5410                     0.6747
#> hospital_beds_per_thousand           life_expectancy
#>                     0.6025                     0.8106
#>    human_development_index
#>                     0.7743
plot(1:p,sort(comm_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Communalities",
     main="Communalities with PCFA")
text(1:p,sort(comm_pcfa,decreasing=TRUE),labels=names(sort(comm_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```



The values of uniqueness provide the same information, but the other way around.

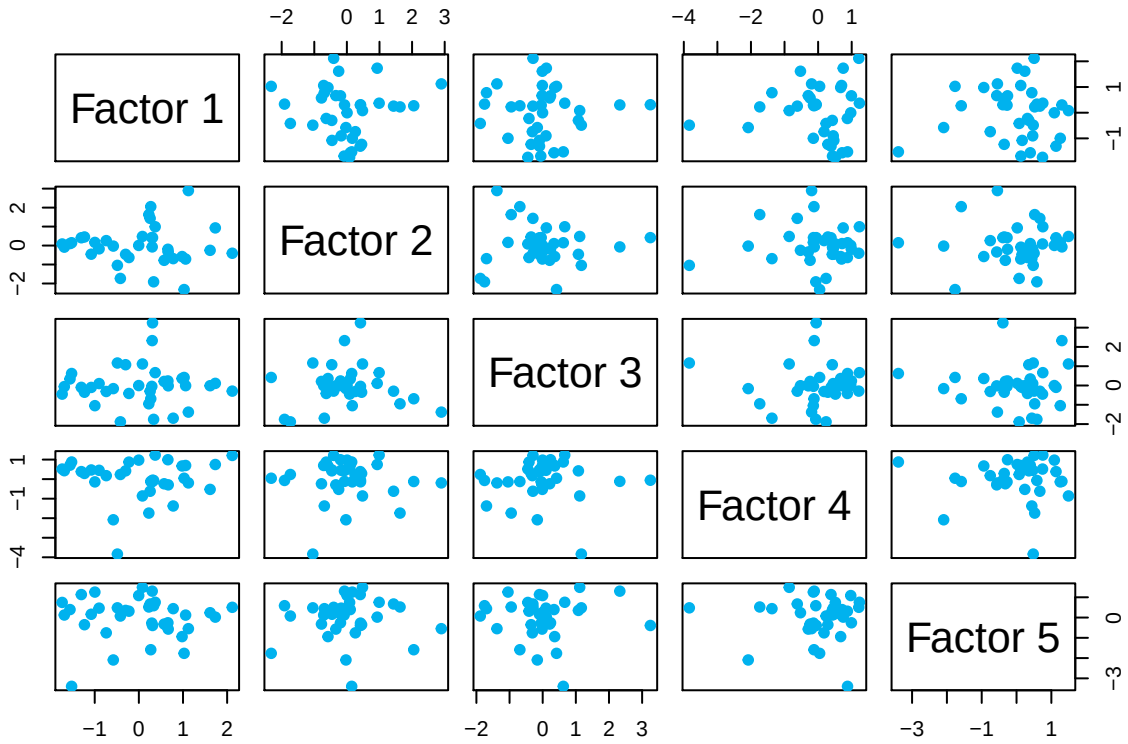
```
uniq_pcfa <- 1 - comm_pcfa
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.08912                    0.20956
#>   total_deaths_per_million           stringency_index
#>                    0.25076                    0.56361
#>                 population          population_density
#>                    0.46461                    0.36635
#>                 median_age               aged_65_older
#>                    0.14918                    0.11291
#>             gdp_per_capita            extreme_poverty
#>                    0.18199                    0.37119
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.45896                    0.32531
#> hospital_beds_per_thousand             life_expectancy
#>                    0.39750                    0.18945
#>    human_development_index
#>                    0.22574
names(uniq_pcfa) <- names(comm_pcfa)
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.08912                    0.20956
#>   total_deaths_per_million           stringency_index
#>                    0.25076                    0.56361
#>                 population          population_density
#>                    0.46461                    0.36635
#>                 median_age               aged_65_older
#>                    0.14918                    0.11291
#>             gdp_per_capita            extreme_poverty
#>                    0.18199                    0.37119
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.45896                    0.32531
#> hospital_beds_per_thousand             life_expectancy
#>                    0.39750                    0.18945
#>    human_development_index
#>                    0.22574
```

```
plot(1:p,sort(uniq_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Uniquenesses",
     main="Uniquenesses with PCFA")
text(1:p,sort(uniq_pcfa,decreasing=TRUE),labels=names(sort(uniq_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```

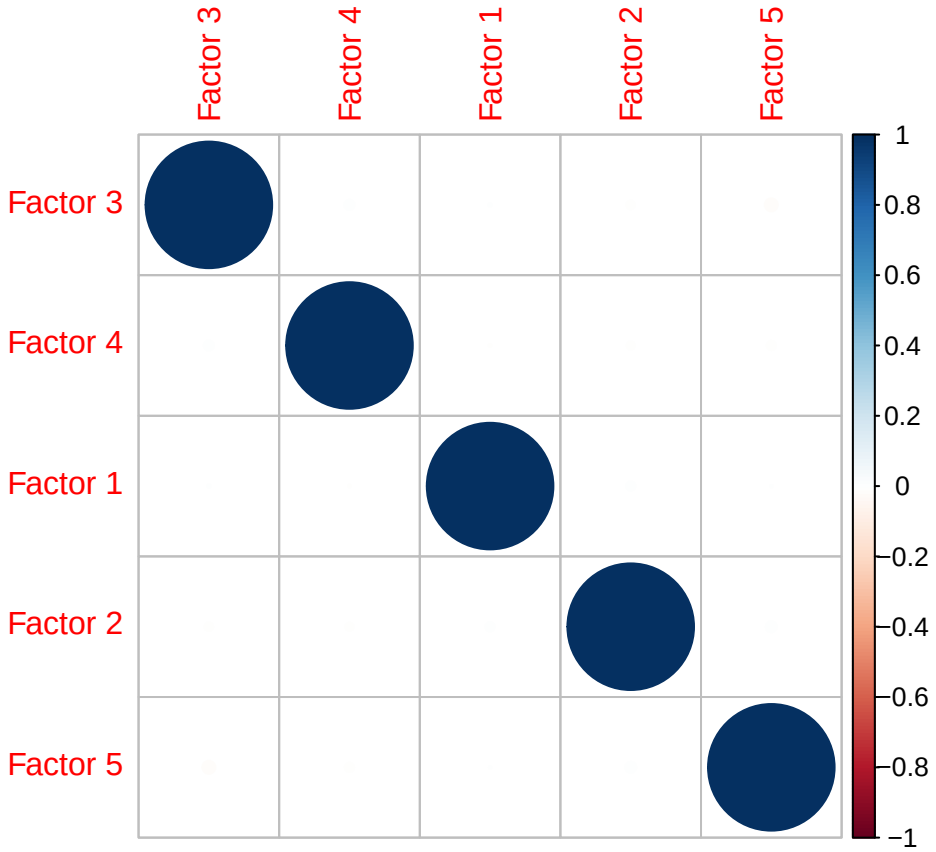


From the following plot we can tell that the factors are uncorrelated.

```
# Estimate the factor scores
F_pcfa <- Y %*% solve(Sigma_nu_pcfa) %*% M_pcfa %*% solve(t(M_pcfa) %*% solve(Sigma_nu_pcfa) %*% M_pcfa)
colnames(F_pcfa) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")

pairs(F_pcfa,pch=19,col=color_1)
```



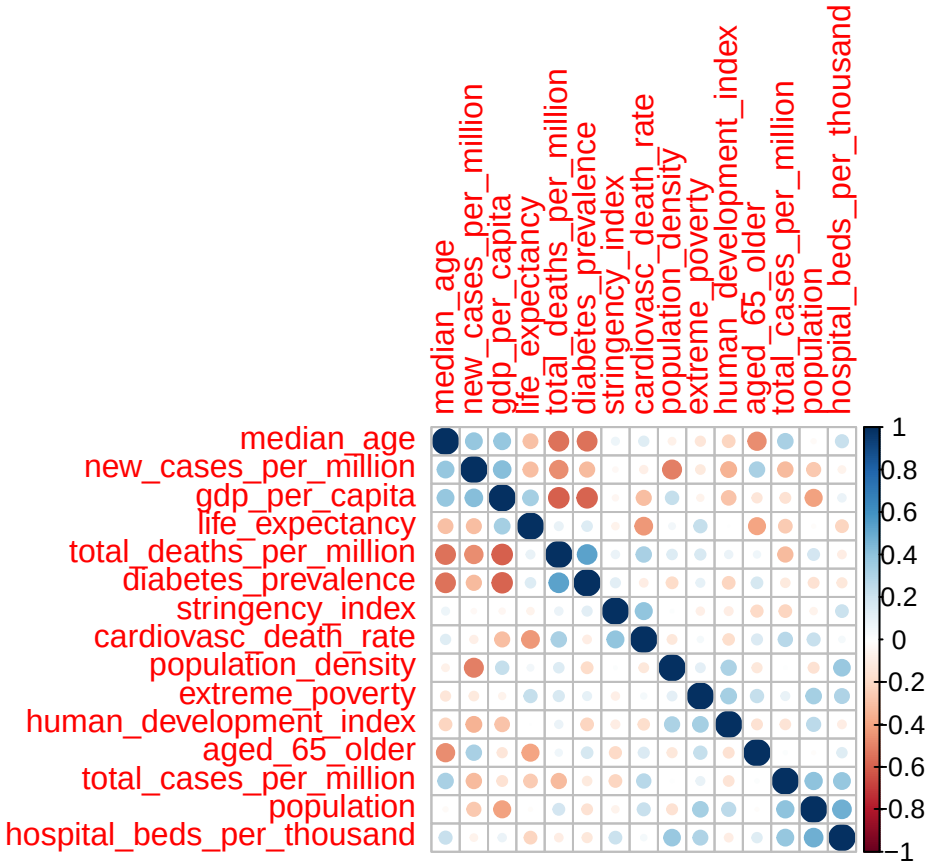

```
corrplot(cor(F_pcfa),order="hclust")
```



We plot the correlations between the residuals. We can see that in the following plot the correlations outside the diagonals the correlations are very low, except the correlation between total death due to covid with

median age and gdp per capita; and diabetes prevalence with median age and gdp per capita. It is possible that if we include another factor, the model will be able to explain them, but as there are only have 18 variables, we will only keep 5 factors.

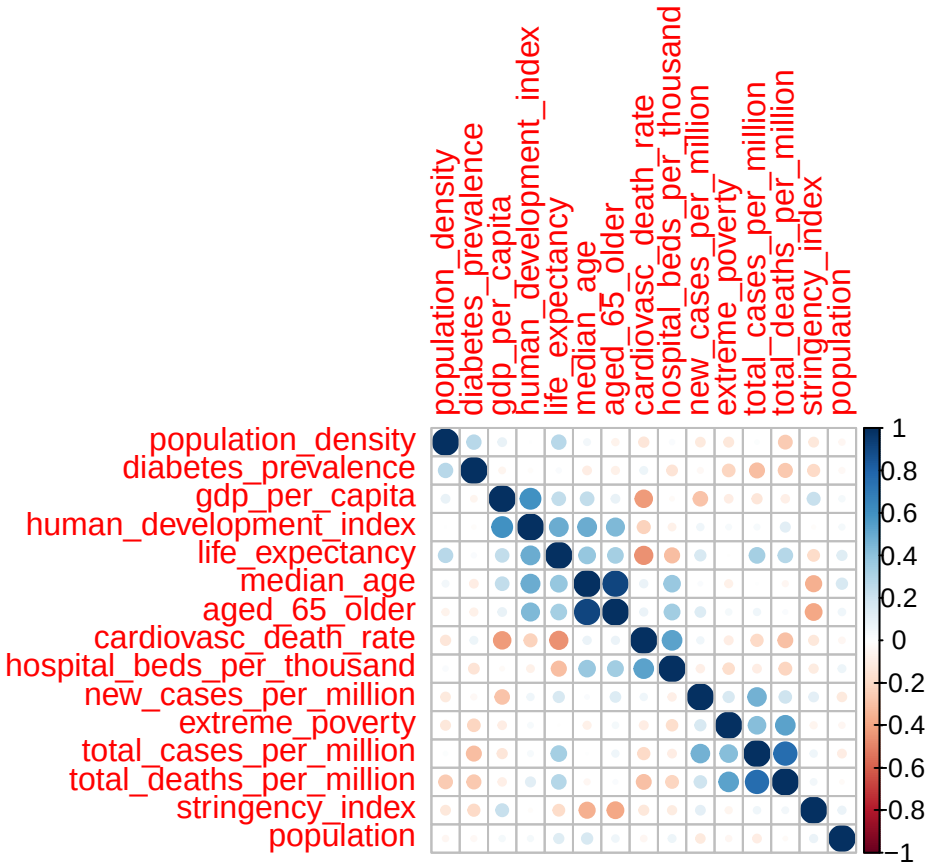
```
# Estimate the residuals
Nu_pcfa <- Y - F_pcfa %*% t(M_pcfa)
corrplot(cor(Nu_pcfa),order="hclust")
```



3. High HDI:

We sorted the variables using their correlations, and we can see that there are few groups of variables that are highly correlated. For instance, we have a group of variables that are followings: extreme poverty, total cases per million and total deaths per million; and another group of variables that are more or less correlated are human development index, life expectancy, median age and aged 65 or older, which is related to how developed a country is.

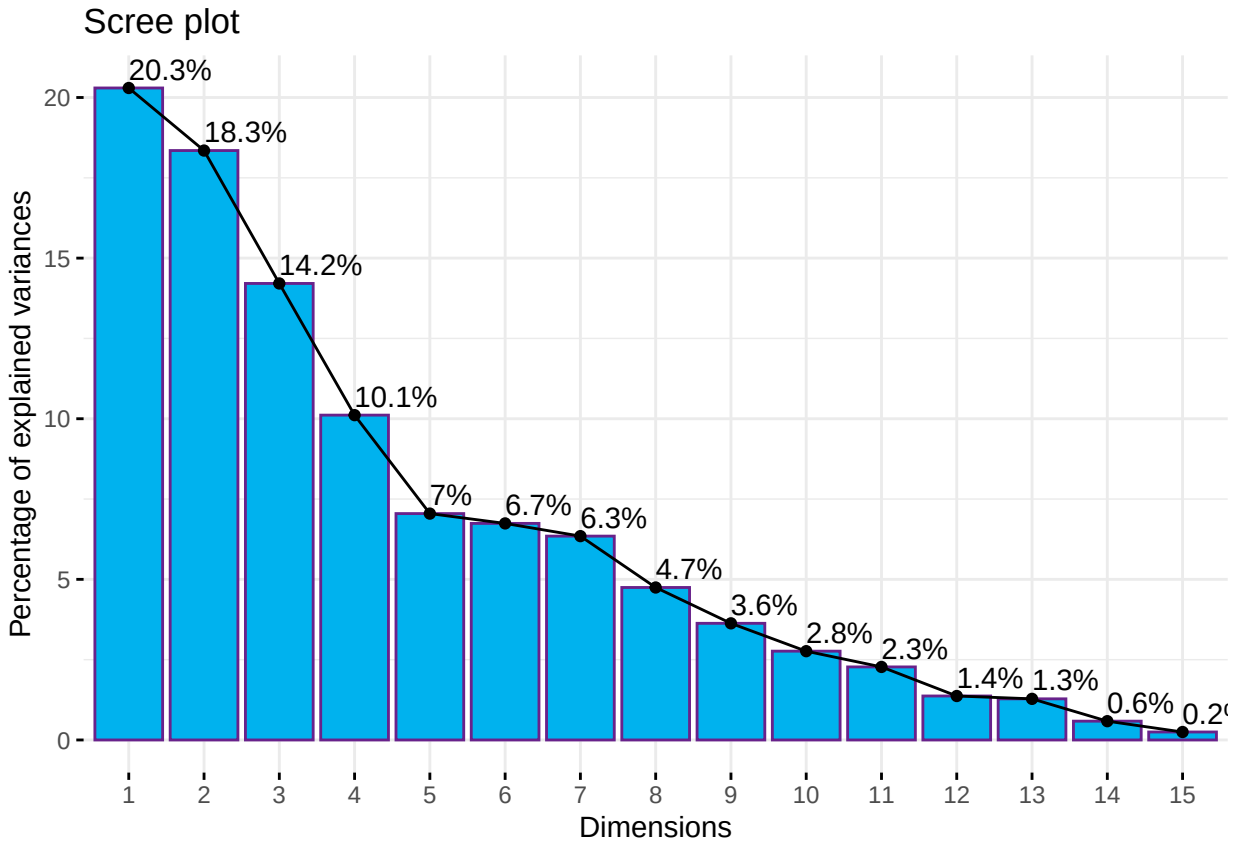
```
# Sample size and dimension of the personality data set
n <- nrow(XX_high)
p <- ncol(XX_high)

corrplot(cor(XX_high),order="hclust")
```




```
# There are groups of correlated variables that may suggest a factor structure
```

We can check here that the variance explained by each principal component, e.g. the first principal component explains 20.3% of the total variability and the secon principal component explains 18.3%. This perhaps is the best group to use the factor analysis as the first 5 factors explain very similar percentage of the variances.

```
# Principal Component Factor Analysis

# Obtain the PCs of the univariate standardized variables
Y <- scale(XX_high)
Y_pcs <- prcomp(Y)

# Screeplot with all the eigenvalues
fviz_eig(Y_pcs,ncp=p,addlabels=T,barfill=color_1,barcolor=color_4)
```



Now we check the eigenvalues and the cumulative and the cumulative percentage of explained variance, and we have decided to take the 5 principal components, we will be using the 5/15, which is 33% of the variables and will be keeping the 70.01% of the total information.

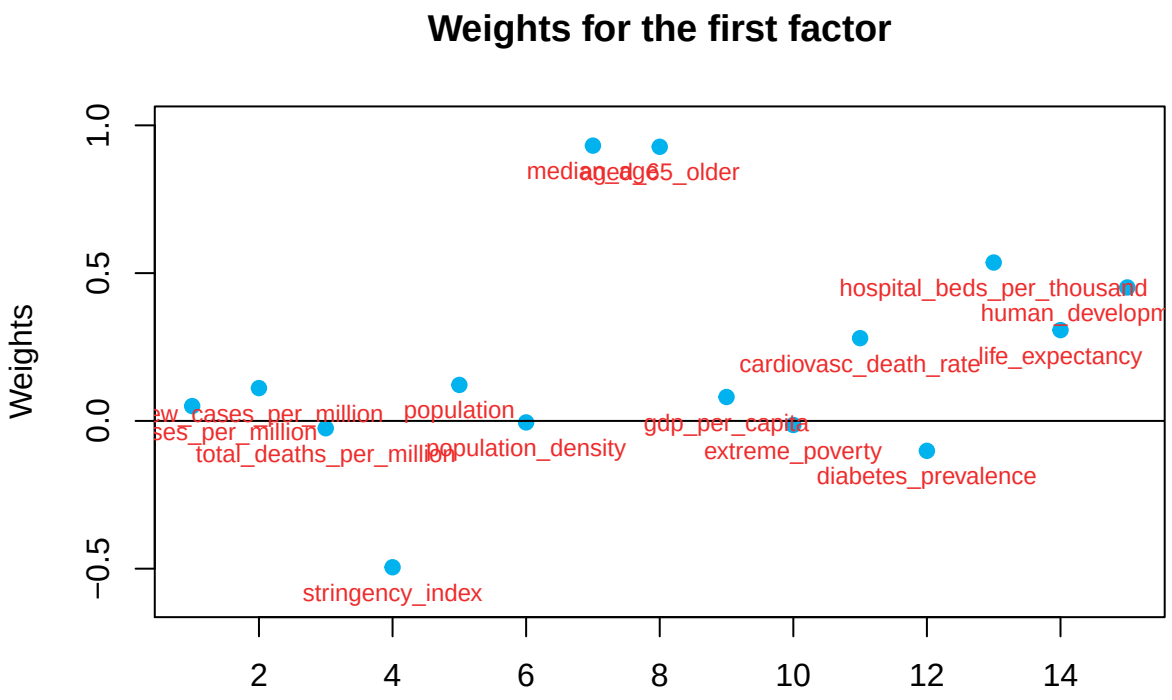
```
get_eigenvalue(Y_pcs)
#>        eigenvalue variance.percent cumulative.variance.percent
#> Dim.1      3.0444          20.2959                       20.30
#> Dim.2      2.7522          18.3482                       38.64
#> Dim.3      2.1316          14.2110                       52.86
#> Dim.4      1.5168          10.1118                       62.97
#> Dim.5      1.0571           7.0472                       70.01
#> Dim.6      1.0112           6.7412                       76.76
#> Dim.7      0.9519           6.3457                       83.10
#> Dim.8      0.7120           4.7468                       87.85
#> Dim.9      0.5447           3.6310                       91.48
#> Dim.10     0.4146           2.7641                       94.24
#> Dim.11     0.3411           2.2741                       96.52
#> Dim.12     0.2054           1.3693                       97.89
#> Dim.13     0.1920           1.2803                       99.17
#> Dim.14     0.0878           0.5853                       99.75
#> Dim.15     0.0372           0.2480                      100.00
```

From now on, let us focus on the first five PCs.

```
r <- 5
# Estimate the matrix M and use the varimax rotation for interpretability
M_pcfa <- Y_pcs$rotation[,1:r] %*% diag(Y_pcs$sdev[1:r])
M_pcfa <- varimax(M_pcfa)
M_pcfa <- loadings(M_pcfa)[1:p,1:r]
```

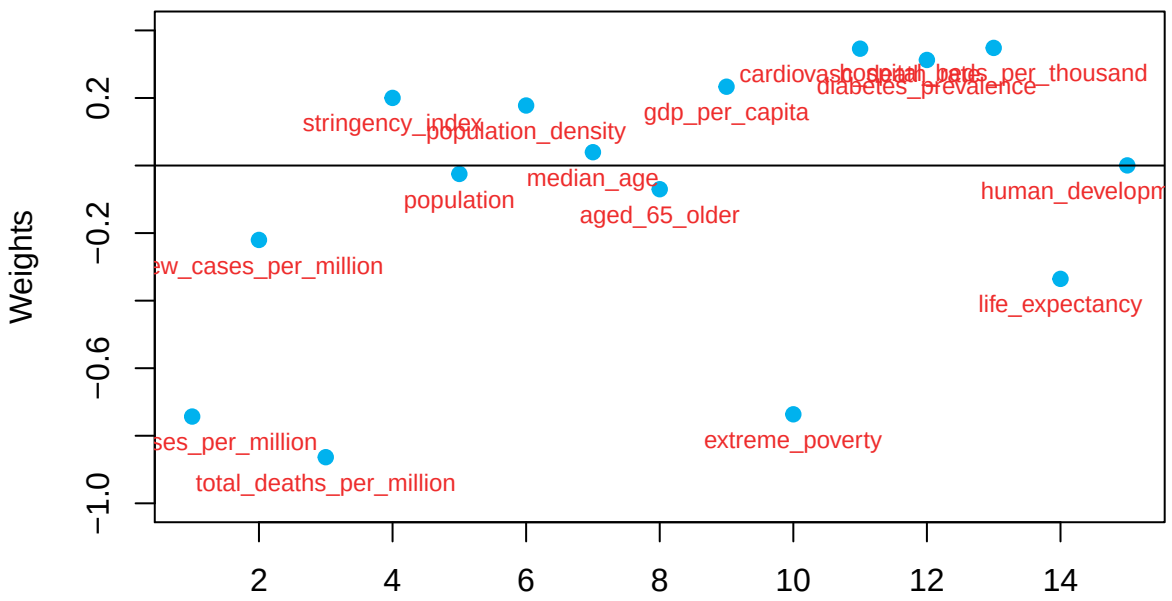
We can observe here that the first factor appears to be an index of age.

```
plot(1:p,M_pcfa[,1],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.6,1),main="Weights for the first factor")
abline(h=0)
text(1:p,M_pcfa[,1],labels=colnames(XX_high),pos=1,col=color_5,cex=0.75)
```



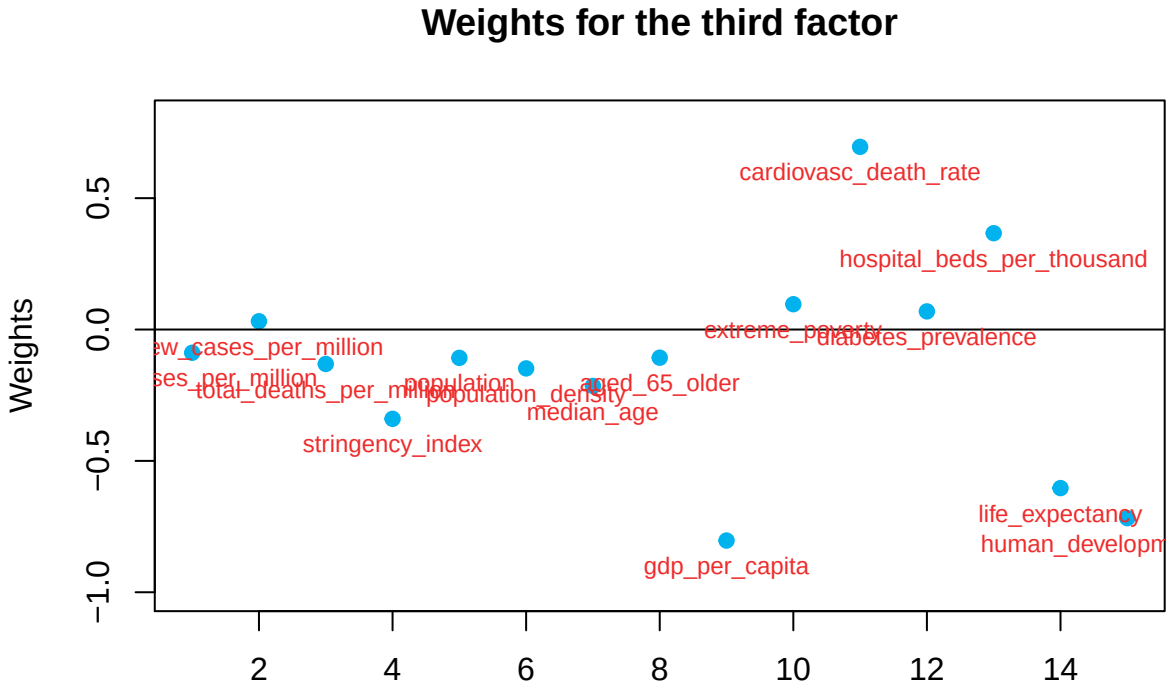
However, the second factor seems to be an index of covid situation of each country.

```
plot(1:p,M_pcfa[,2],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,0.4),main="Weights for the second factor")
abline(h=0)
text(1:p,M_pcfa[,2],labels=colnames(XX_high),pos=1,col=color_5,cex=0.75)
```



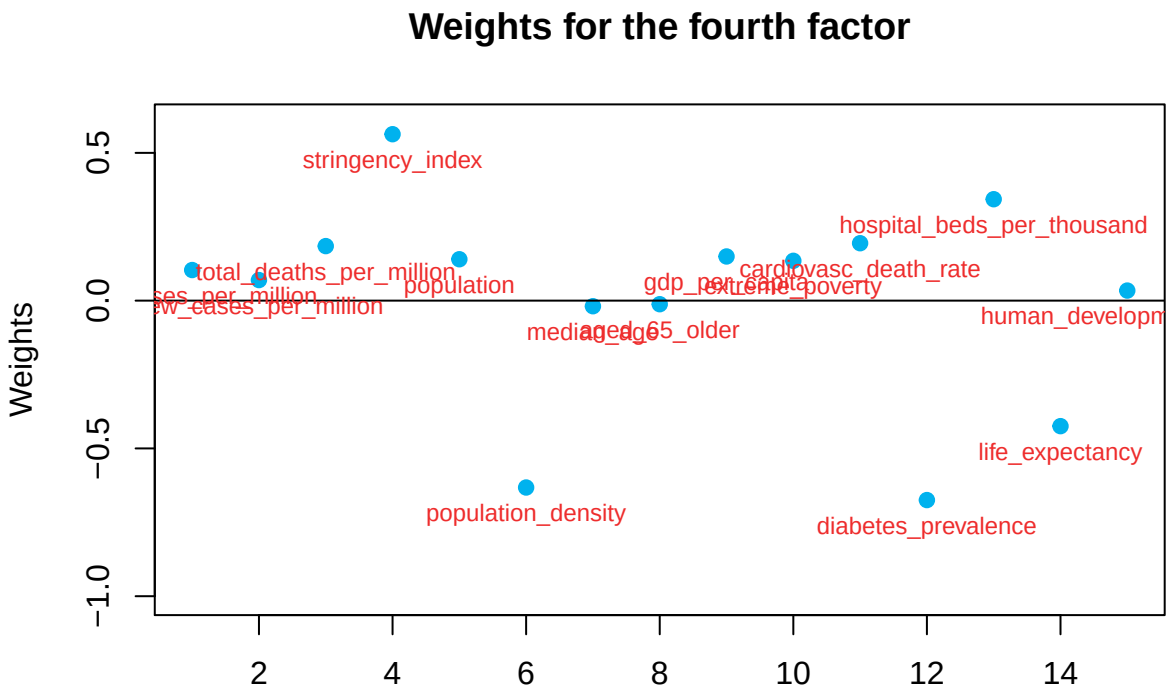
The third factor appears to be an index of how undeveloped a country is (positive cardiovascular death rate weight and negative human development index and life expectancy).

```
plot(1:p,M_pcfa[,3],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,0.8),main="Weights for the third factor")
abline(h=0)
text(1:p,M_pcfa[,3],labels=colnames(XX_high),pos=1,col=color_5,cex=0.75)
```



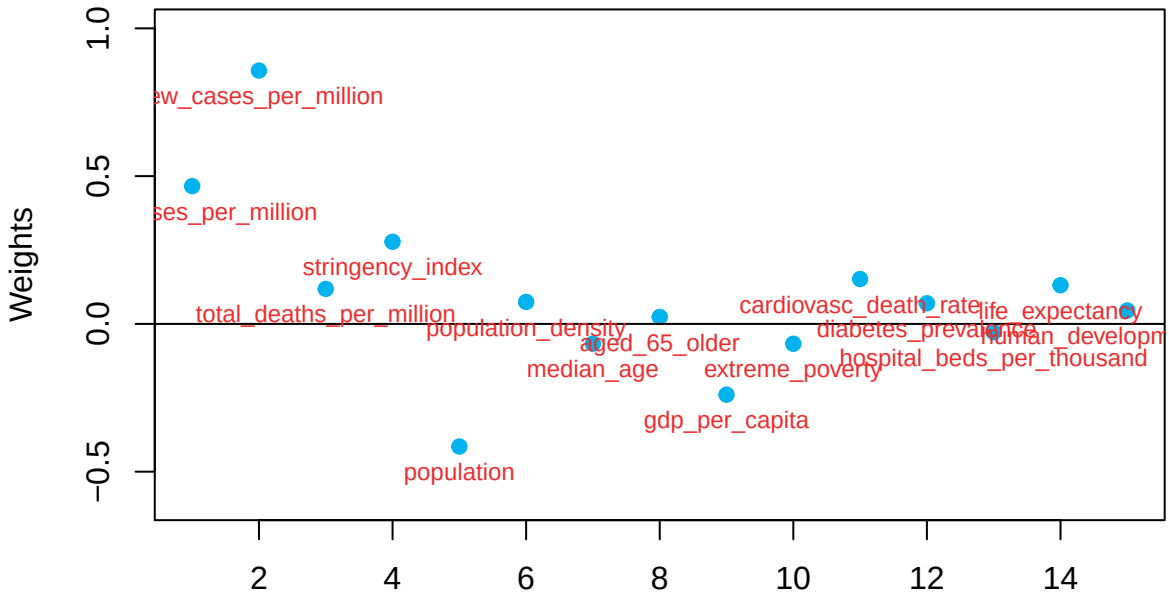
The fourth factor seems to be an index of mixture of variables stringency index with positive weight and population density and diabetes prevalence with negative weight.

```
plot(1:p,M_pcfa[,4],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,0.6),main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_pcfa[,4],labels=colnames(XX_high),pos=1,col=color_5,cex=0.75)
```



The fifth factor appears to be an index of the situation of covid (majorly explained by new cases per million).

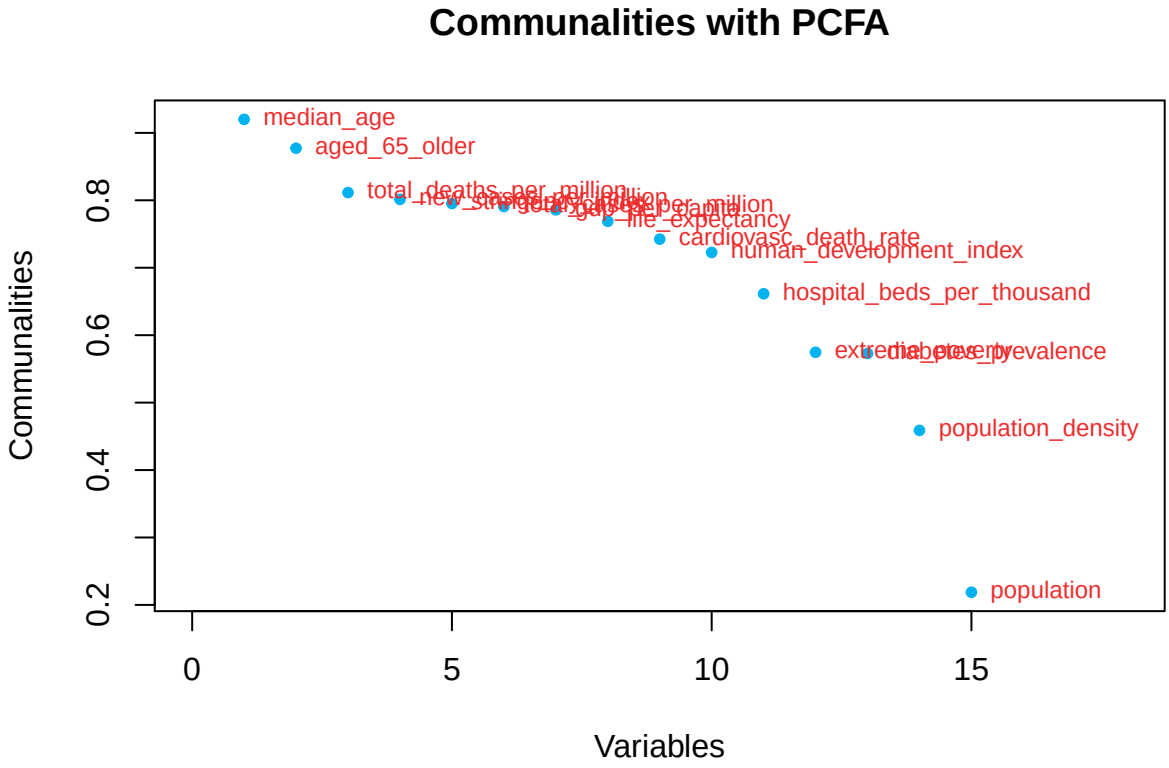
```
plot(1:p,M_pcfa[,5],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.6,1),main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_pcfa[,5],labels=colnames(XX_high),pos=1,col=color_5,cex=0.75)
```




```
# Estimate the covariance matrix of the errors
Sigma_nu_pcfa <- diag(diag(cov(Y) - M_pcfa %*% t(M_pcfa)))
```

Here we plot the values of communalities and we can observe that the aspects of countries that are better explained by the factors are median age, aged 65 or older (more or less 90%). And the variables that are not explained that well is population (20%).

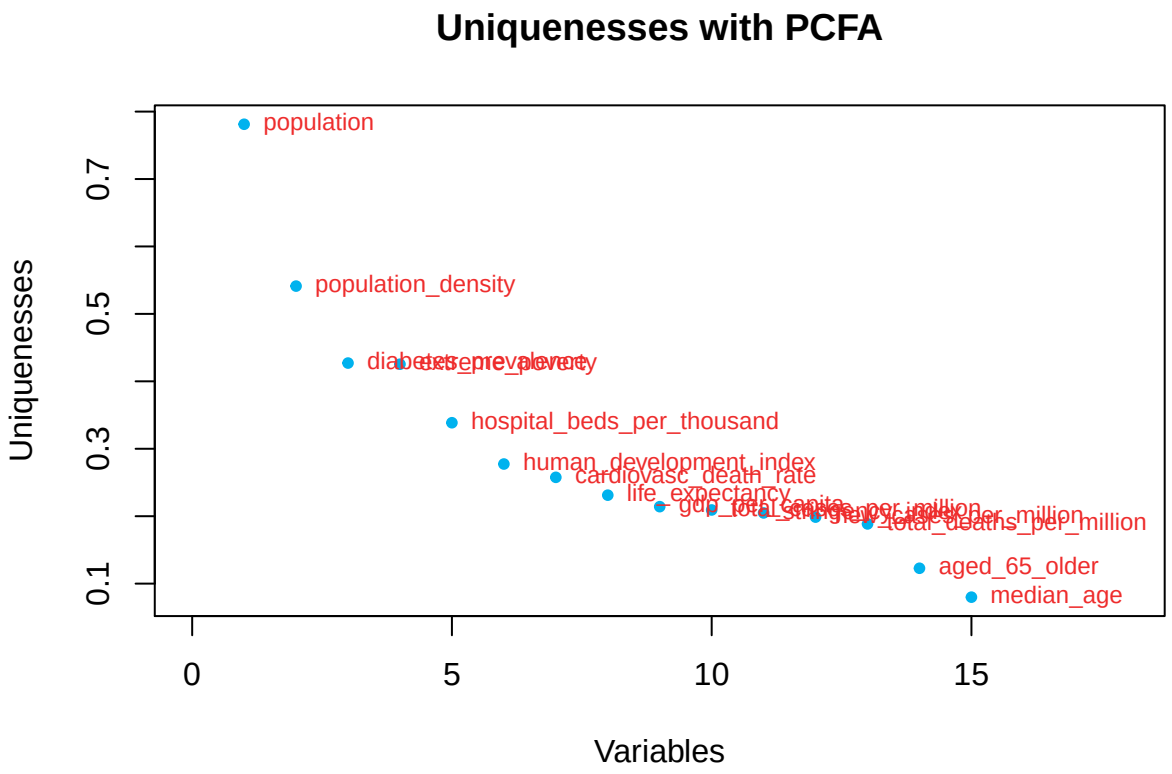
```
# Communalities and uniquenesses
comm_pcfa <- diag(M_pcfa %*% t(M_pcfa))
comm_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.7907                     0.8014
#>   total_deaths_per_million          stringency_index
#>                     0.8114                     0.7952
#>                 population         population_density
#>                     0.2188                     0.4588
#>                 median_age              aged_65_older
#>                     0.9201                     0.8772
#>            gdp_per_capita           extreme_poverty
#>                     0.7859                     0.5746
#>      cardiovasc_death_rate        diabetes_prevalence
#>                     0.7423                     0.5729
#> hospital_beds_per_thousand           life_expectancy
#>                     0.6614                     0.7688
#>    human_development_index
#>                     0.7226
plot(1:p,sort(comm_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Communalities",
     main="Communalities with PCFA")
text(1:p,sort(comm_pcfa,decreasing=TRUE),labels=names(sort(comm_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```



The values of uniqueness are the same, but the other way around.

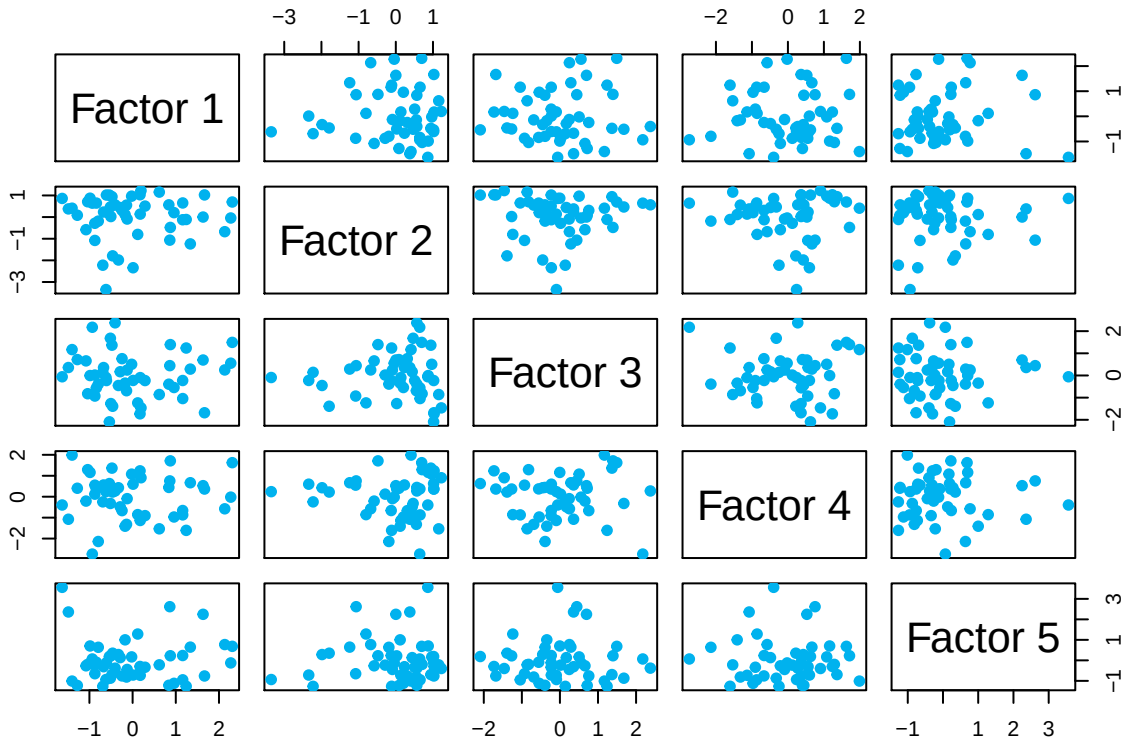
```
uniq_pcfa <- 1 - comm_pcfa
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.20925                    0.19863
#>   total_deaths_per_million           stringency_index
#>                    0.18857                    0.20484
#>                 population          population_density
#>                    0.78119                    0.54121
#>                 median_age               aged_65_older
#>                    0.07994                    0.12280
#>             gdp_per_capita            extreme_poverty
#>                    0.21415                    0.42540
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.25765                    0.42710
#> hospital_beds_per_thousand             life_expectancy
#>                    0.33857                    0.23119
#>    human_development_index
#>                    0.27738
names(uniq_pcfa) <- names(comm_pcfa)
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.20925                    0.19863
#>   total_deaths_per_million           stringency_index
#>                    0.18857                    0.20484
#>                 population          population_density
#>                    0.78119                    0.54121
#>                 median_age               aged_65_older
#>                    0.07994                    0.12280
#>             gdp_per_capita            extreme_poverty
#>                    0.21415                    0.42540
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.25765                    0.42710
#> hospital_beds_per_thousand             life_expectancy
#>                    0.33857                    0.23119
#>    human_development_index
#>                    0.27738
```

```r
plot(1:p,sort(uniq_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Uniquenesses",
     main="Uniquenesses with PCFA")
text(1:p,sort(uniq_pcfa,decreasing=TRUE),labels=names(sort(uniq_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```

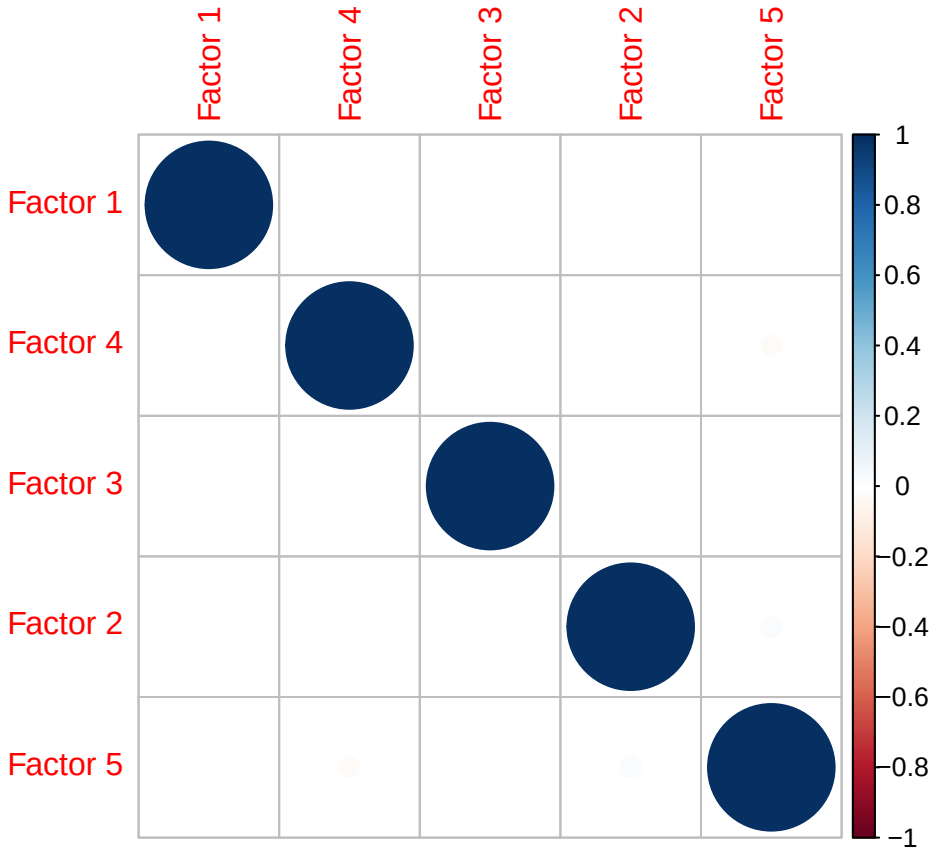


From the following plot we can tell that the factors are uncorrelated at all.

```r
# Estimate the factor scores
F_pcfa <- Y %*% solve(Sigma_nu_pcfa) %*% M_pcfa %*% solve(t(M_pcfa) %*% solve(Sigma_nu_pcfa) %*% M_pcfa)
colnames(F_pcfa) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")

pairs(F_pcfa,pch=19,col=color_1)
```



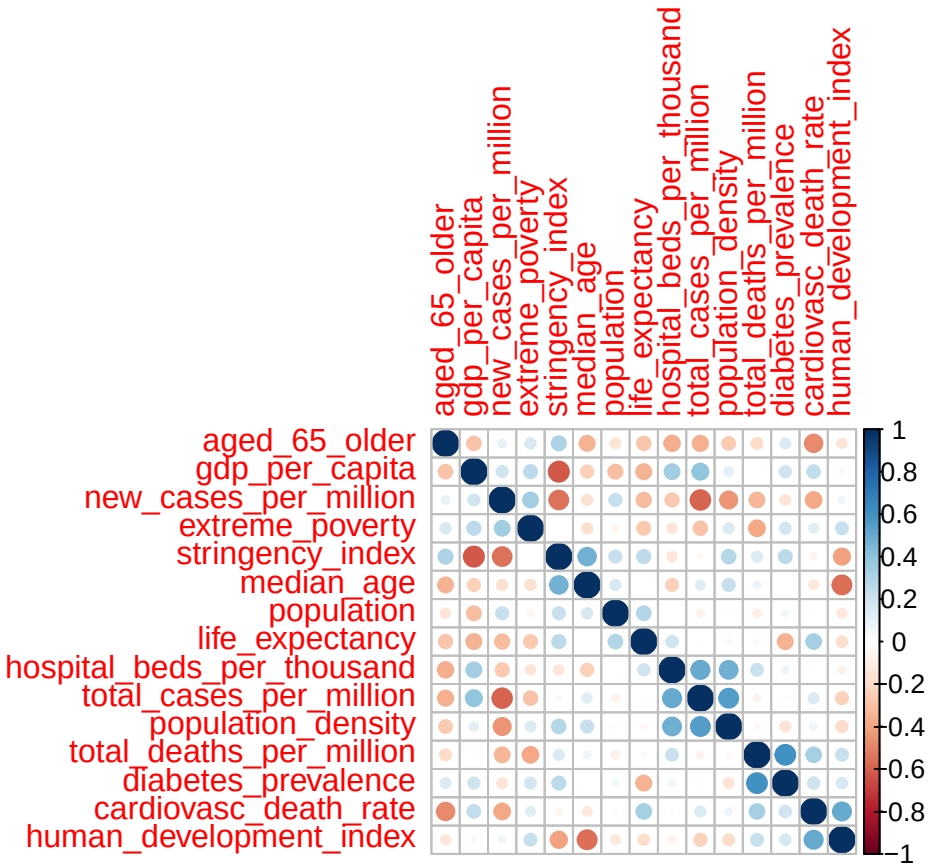

```
corrplot(cor(F_pcfa),order="hclust")
```



We plot the correlations between the residuals. We can see that in the following plot the correlations outside the diagonals the correlations are very low, except some of them, e.g., the correlation between stringency

index and gdp per capita and between stringency index and new cases per million. They might be explained using more factors.

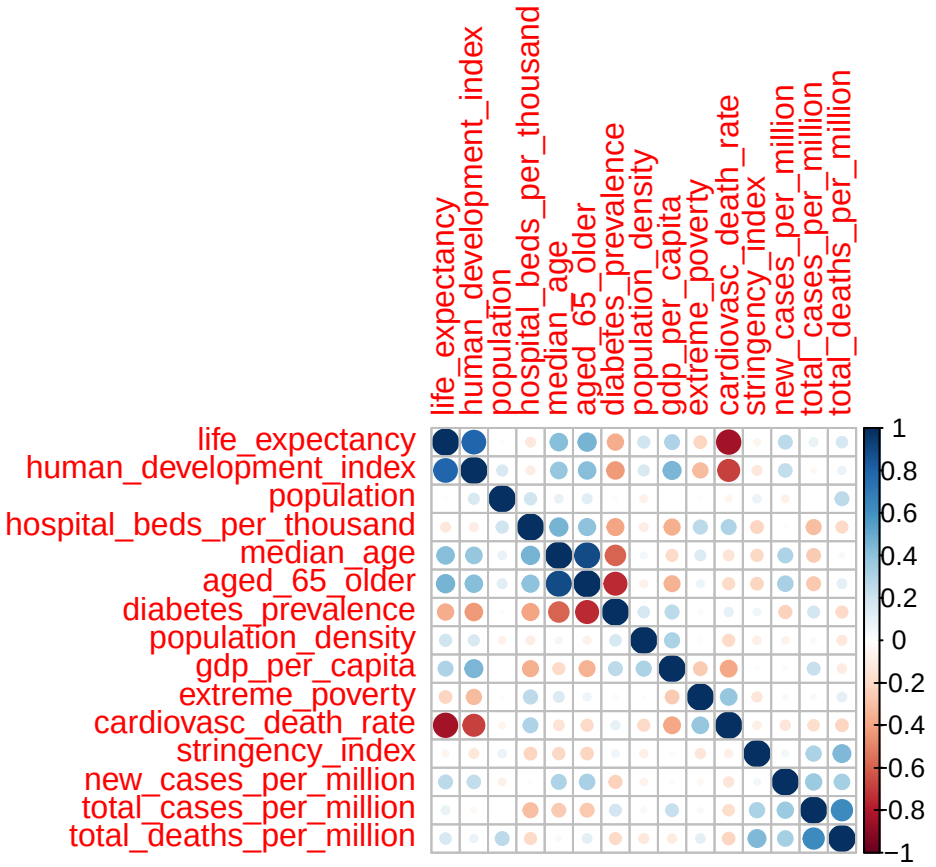
```
# Estimate the residuals
Nu_pcfa <- Y - F_pcfa %*% t(M_pcfa)
corrplot(cor(Nu_pcfa),order="hclust")
```



4. Very High HDI:

We sorted the variables using their correlations, and we can see that there are groups of variables that are highly correlated. For instance, we have a group of variables which is relatad to how developed a country is: median age, aged 65 or older, and these variables are negatively correlated to diabetes prevalence; and another group of variables that are highly correlated are life expectancy and human development index, and they are negatively correlated to cardiovascular death rate.

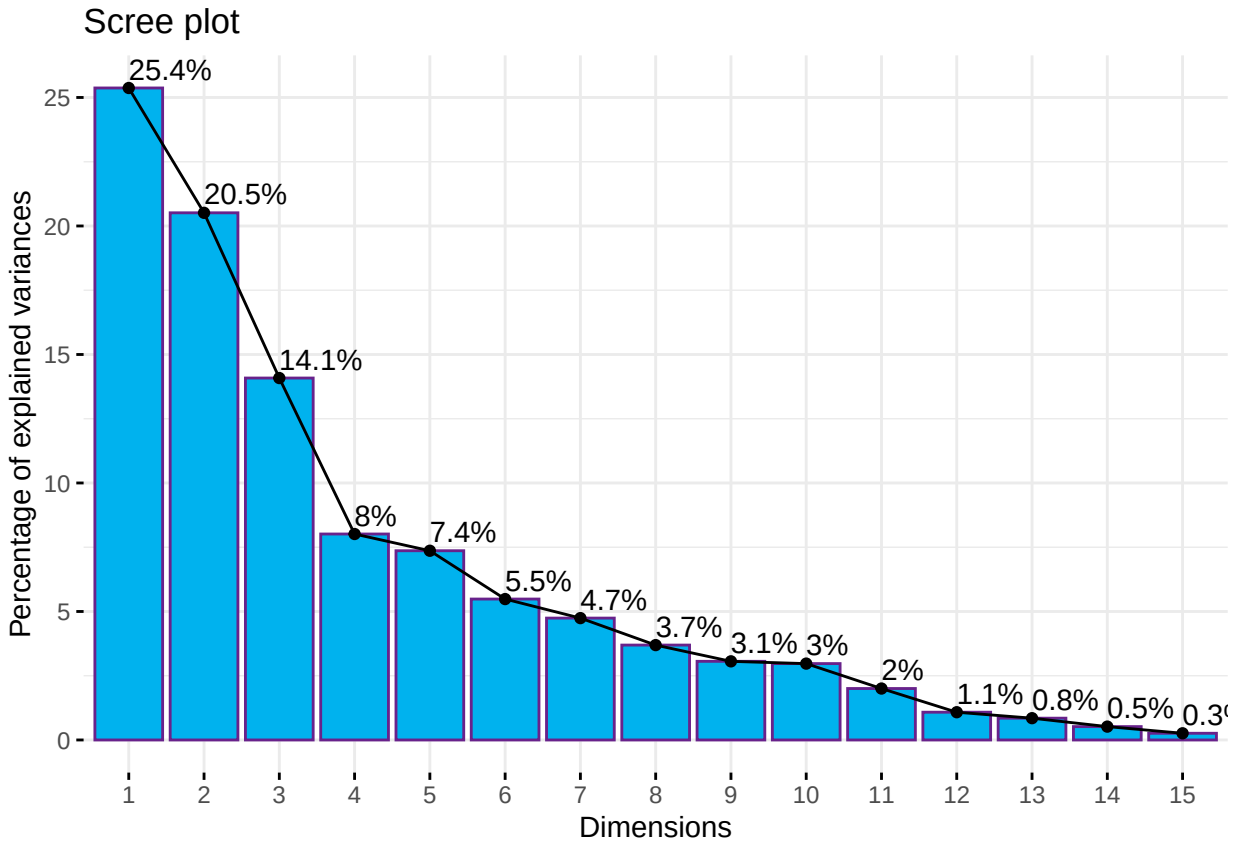
```
# Sample size and dimension of the personality data set
n <- nrow(XX_veryhigh)
p <- ncol(XX_veryhigh)

corrplot(cor(XX_veryhigh),order="hclust")
```




```
# There are groups of correlated variables that may suggest a factor structure
```

We can check here that the variance explained by each principal component, e.g. the first principal component explains 25.4% of the total variability and the secon principal component explains 20.5%.

```
# Principal Component Factor Analysis

# Obtain the PCs of the univariate standardized variables
Y <- scale(XX_veryhigh)
Y_pcs <- prcomp(Y)

# Screeplot with all the eigenvalues
fviz_eig(Y_pcs,ncp=p,addlabels=T,barfill=color_1,barcolor=color_4)
```



Now we check the eigenvalues and the cumulative and the cumulative percentage of explained variance, and we have decided to take the 5 principal components, we will be using the 5/15, which is 33% of the variables and will be keeping the 75.34% of the total information.

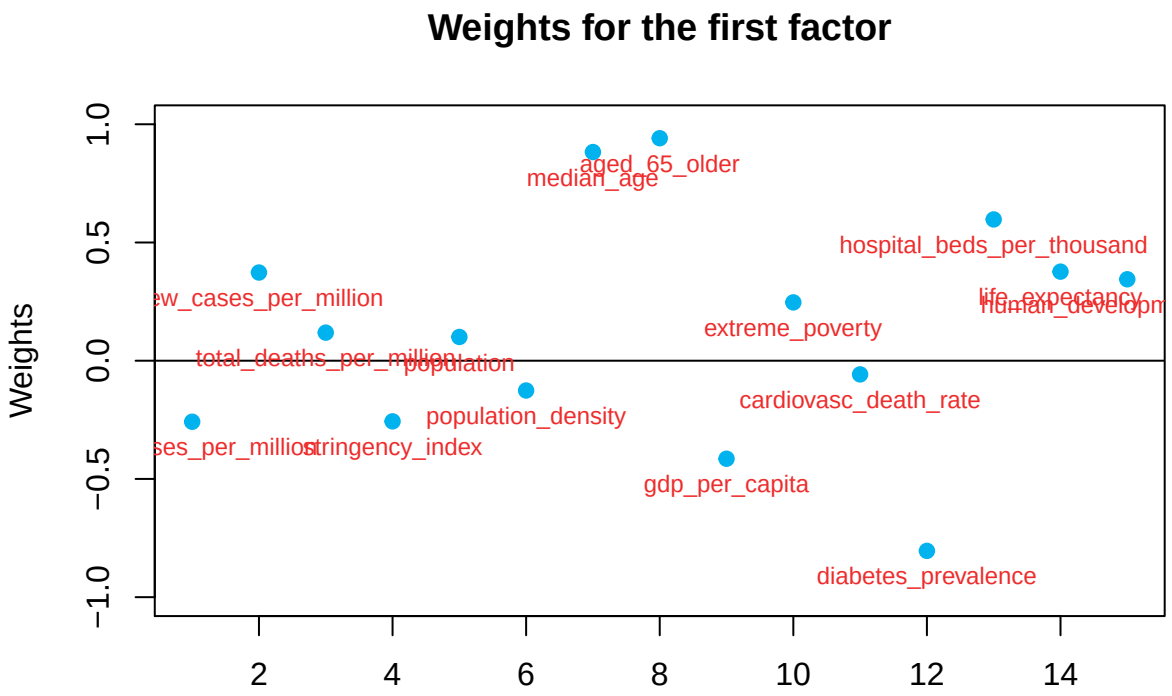
```
get_eigenvalue(Y_pcs)
#>        eigenvalue variance.percent cumulative.variance.percent
#> Dim.1     3.80537          25.3691                       25.37
#> Dim.2     3.07679          20.5119                       45.88
#> Dim.3     2.11249          14.0832                       59.96
#> Dim.4     1.20213           8.0142                       67.98
#> Dim.5     1.10443           7.3629                       75.34
#> Dim.6     0.82249           5.4833                       80.82
#> Dim.7     0.71132           4.7421                       85.57
#> Dim.8     0.55377           3.6918                       89.26
#> Dim.9     0.45898           3.0599                       92.32
#> Dim.10    0.44573           2.9716                       95.29
#> Dim.11    0.30018           2.0012                       97.29
#> Dim.12    0.16200           1.0800                       98.37
#> Dim.13    0.12724           0.8483                       99.22
#> Dim.14    0.07798           0.5199                       99.74
#> Dim.15    0.03910           0.2607                      100.00
```

From now on, let us focus on the first five PCs.

```
r <- 5
# Estimate the matrix M and use the varimax rotation for interpretability
M_pcfa <- Y_pcs$rotation[,1:r] %*% diag(Y_pcs$sdev[1:r])
M_pcfa <- varimax(M_pcfa)
M_pcfa <- loadings(M_pcfa)[1:p,1:r]
```

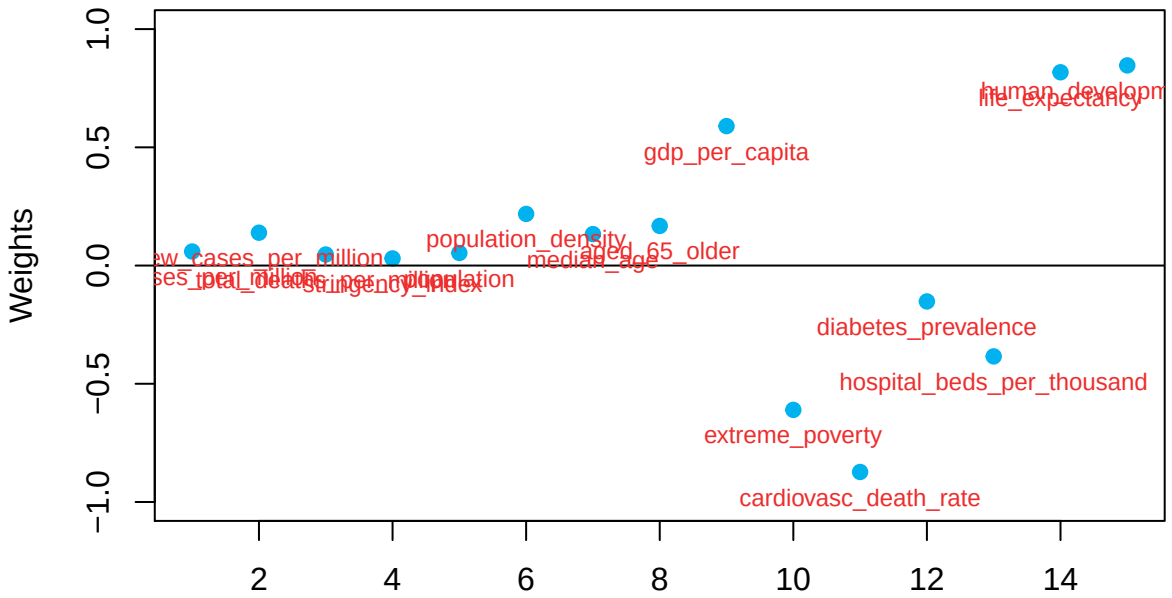
We can observe here that the first factor appears to be an index related of a mixture of age and diabetes prevalence, as we can see that the median age, aged 65 or older have really high positive weights (close to 1) and diabetes prevalence have really low negative value.

```
plot(1:p,M_pcfa[,1],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,1),main="Weights for the first factor")
abline(h=0)
text(1:p,M_pcfa[,1],labels=colnames(XX_veryhigh),pos=1,col=color_5,cex=0.75)
```



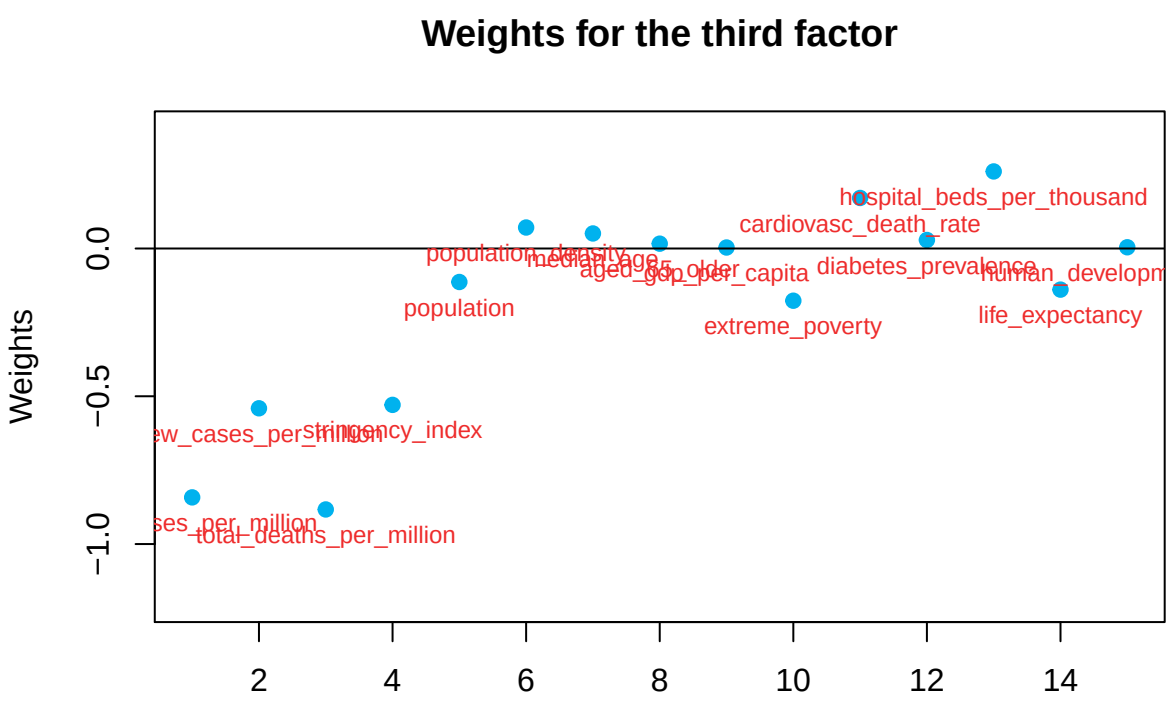
However, the second factor appears to be an index related how developed a country is, HDI, life expectancy and gdp per capita are to be very highly weighted while the cardiovascular death rate and extreme poverty rate are negative (near -1).

```
plot(1:p,M_pcfa[,2],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1,1),main="Weights for the second factor")
abline(h=0)
text(1:p,M_pcfa[,2],labels=colnames(XX_veryhigh),pos=1,col=color_5,cex=0.75)
```



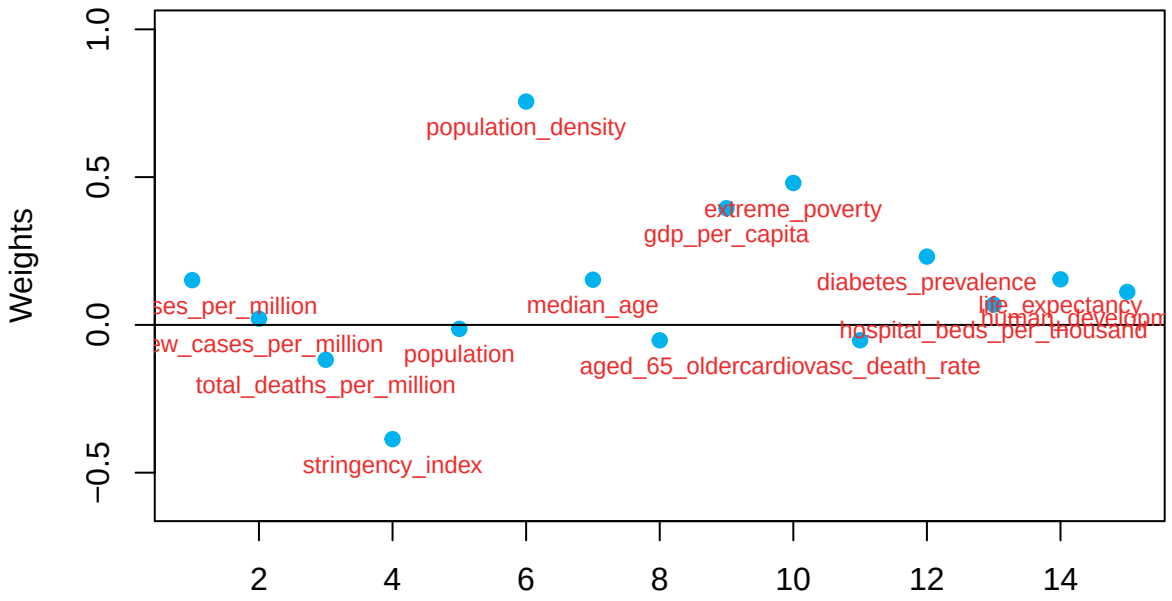
The third factor appears to be a index related to the situation of the pandemic of a country, the better the situation, the higher the value, as this index depends total deaths per million, cases per million, new cases per million (with negative values).

```
plot(1:p,M_pcfa[,3],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1.2,0.4),main="Weights for the third factor")
abline(h=0)
text(1:p,M_pcfa[,3],labels=colnames(XX_veryhigh),pos=1,col=color_5,cex=0.75)
```



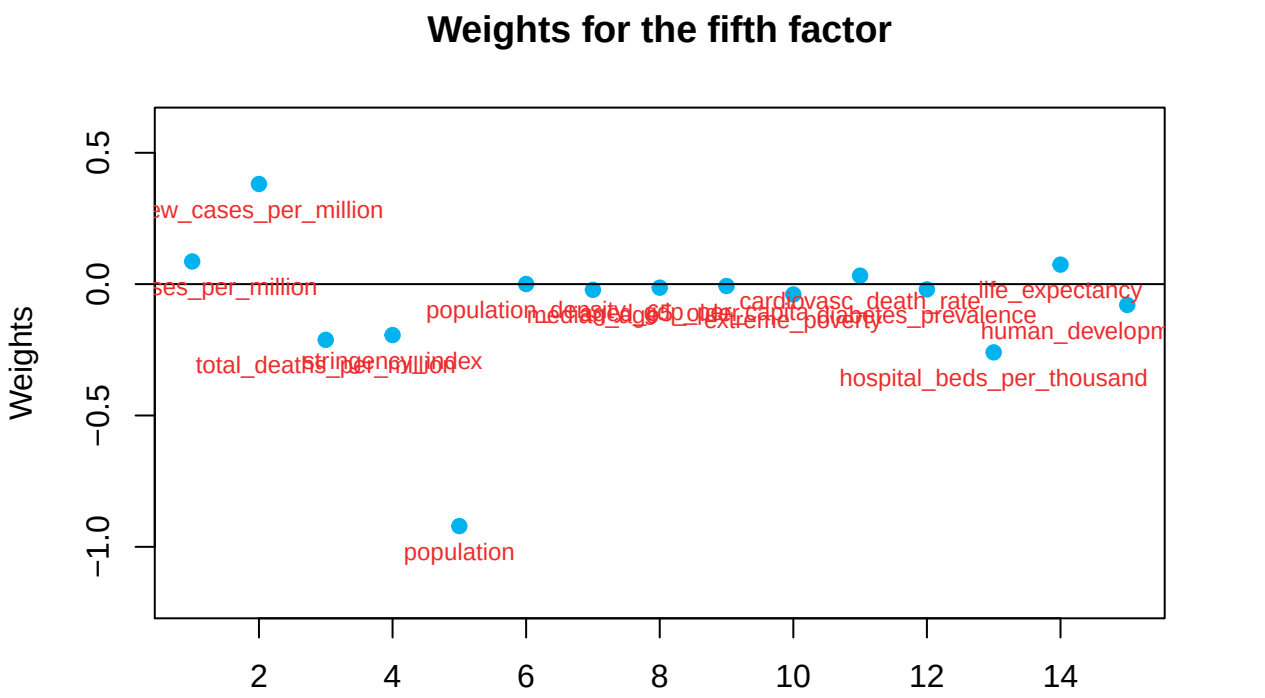
The fourth factor seems to be an index of population density of a country.

```
plot(1:p,M_pcfa[,4],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-0.6,1),main="Weights for the fourth factor")
abline(h=0)
text(1:p,M_pcfa[,4],labels=colnames(XX_veryhigh),pos=1,col=color_5,cex=0.75)
```



The fifth factor appears to be an index of population.

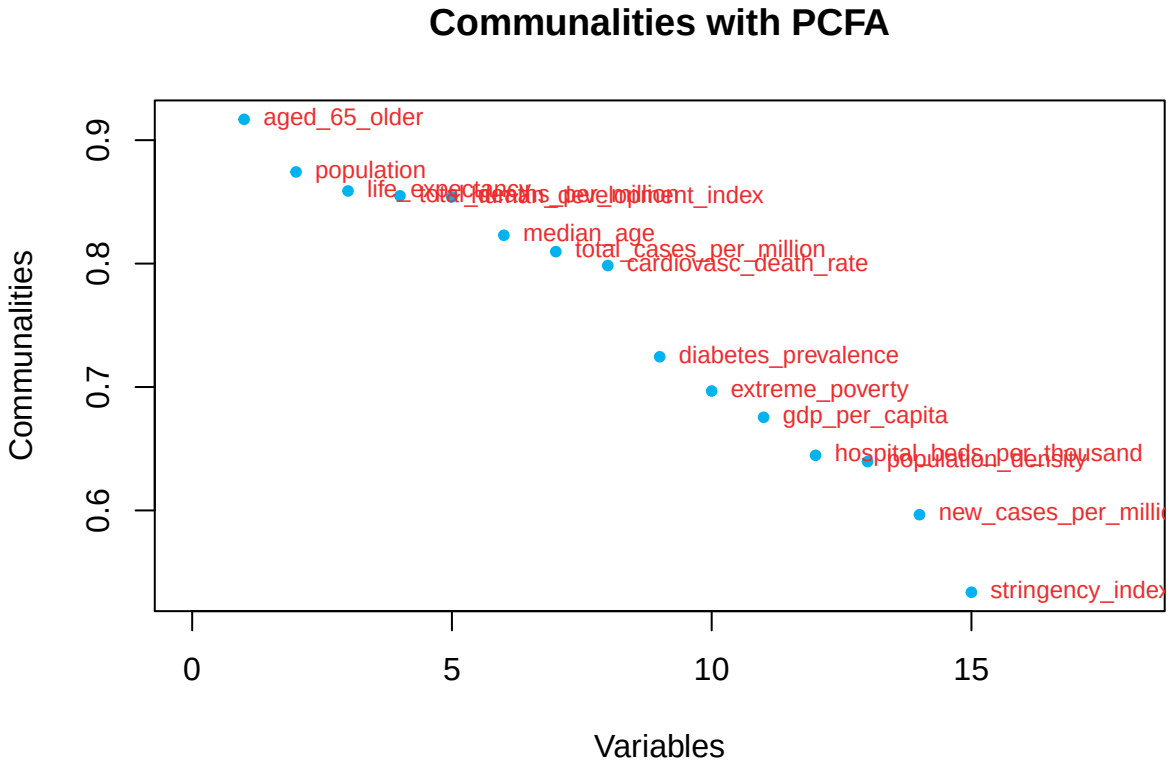
```
plot(1:p,M_pcfa[,5],pch=19,col=color_1,xlab="",ylab="Weights",ylim=c(-1.2,0.6),main="Weights for the fifth factor")
abline(h=0)
text(1:p,M_pcfa[,5],labels=colnames(XX_veryhigh),pos=1,col=color_5,cex=0.75)
```




```
# Estimate the covariance matrix of the errors
Sigma_nu_pcfa <- diag(diag(cov(Y) - M_pcfa %*% t(M_pcfa)))
```

Here we plot the values of communalities and we can observe that the aspect of countries that is better explained by the factors is aged 65 or older (more or less 90%). And the variables that are not explained that well is stringency index.

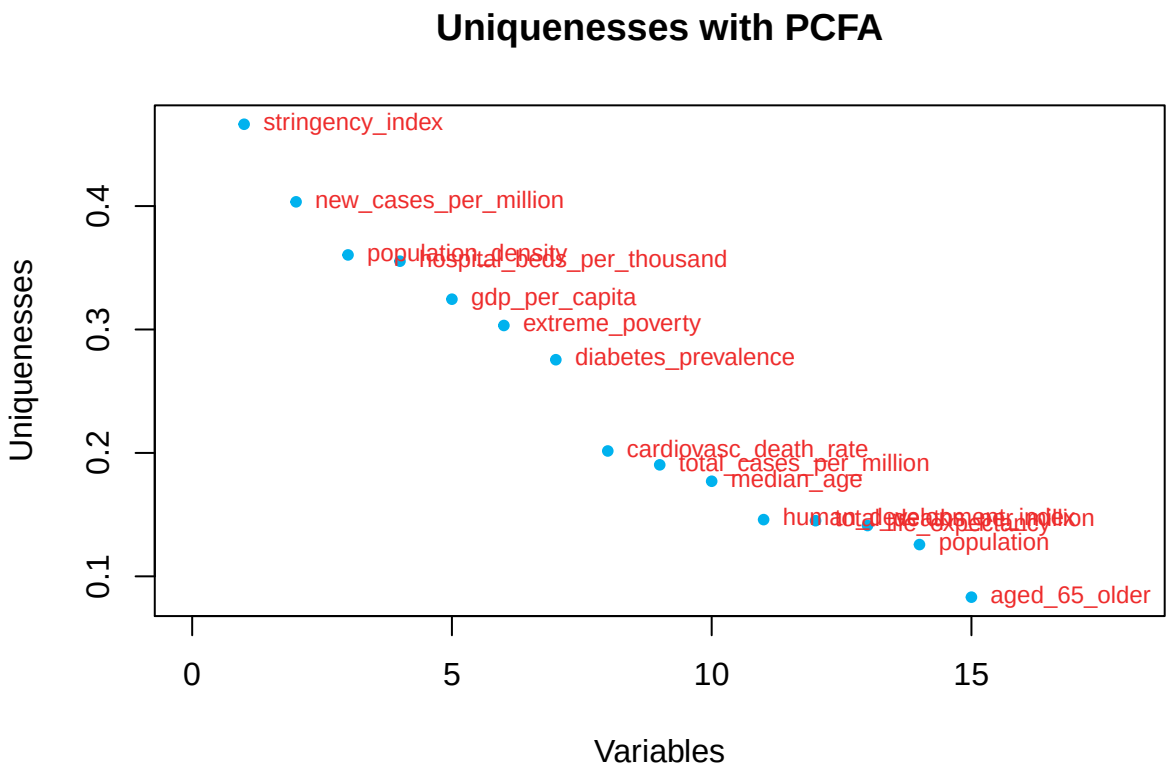
```
# Communalities and uniquenesses
comm_pcfa <- diag(M_pcfa %*% t(M_pcfa))
comm_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                     0.8097                     0.5966
#>   total_deaths_per_million          stringency_index
#>                     0.8549                     0.5337
#>                 population         population_density
#>                     0.8743                     0.6396
#>                 median_age              aged_65_older
#>                     0.8229                     0.9169
#>            gdp_per_capita           extreme_poverty
#>                     0.6754                     0.6968
#>      cardiovasc_death_rate        diabetes_prevalence
#>                     0.7984                     0.7245
#> hospital_beds_per_thousand           life_expectancy
#>                     0.6447                     0.8588
#>    human_development_index
#>                     0.8540
plot(1:p,sort(comm_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Communalities",
     main="Communalities with PCFA")
text(1:p,sort(comm_pcfa,decreasing=TRUE),labels=names(sort(comm_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```



The values of uniqueness are the same, but the other way around.

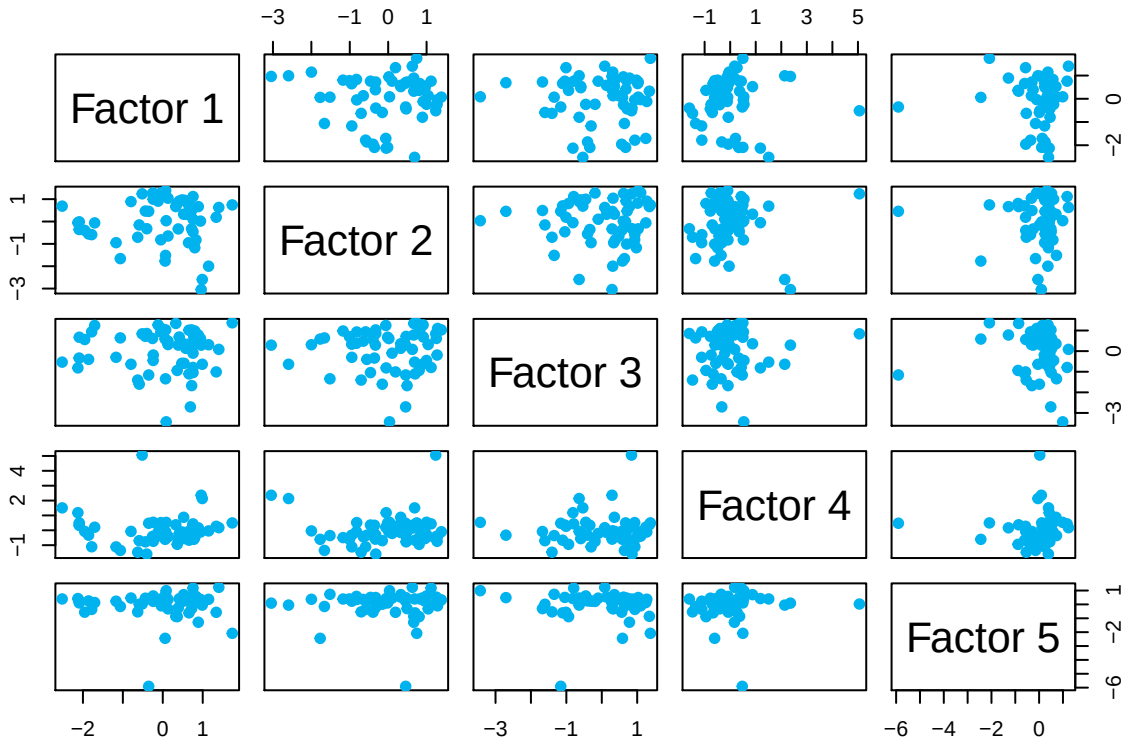
```
uniq_pcfa <- 1 - comm_pcfa
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.19034                    0.40342
#>   total_deaths_per_million           stringency_index
#>                    0.14510                    0.46627
#>                 population          population_density
#>                    0.12574                    0.36041
#>                 median_age               aged_65_older
#>                    0.17706                    0.08314
#>             gdp_per_capita            extreme_poverty
#>                    0.32455                    0.30324
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.20157                    0.27551
#> hospital_beds_per_thousand             life_expectancy
#>                    0.35533                    0.14116
#>    human_development_index
#>                    0.14595
names(uniq_pcfa) <- names(comm_pcfa)
uniq_pcfa
#>    total_cases_per_million      new_cases_per_million
#>                    0.19034                    0.40342
#>   total_deaths_per_million           stringency_index
#>                    0.14510                    0.46627
#>                 population          population_density
#>                    0.12574                    0.36041
#>                 median_age               aged_65_older
#>                    0.17706                    0.08314
#>             gdp_per_capita            extreme_poverty
#>                    0.32455                    0.30324
#>      cardiovasc_death_rate         diabetes_prevalence
#>                    0.20157                    0.27551
#> hospital_beds_per_thousand             life_expectancy
#>                    0.35533                    0.14116
#>    human_development_index
#>                    0.14595
```

```r
plot(1:p,sort(uniq_pcfa,decreasing=TRUE),pch=20,col=color_1,xlim=c(0,18),xlab="Variables",ylab="Uniquenesses",
     main="Uniquenesses with PCFA")
text(1:p,sort(uniq_pcfa,decreasing=TRUE),labels=names(sort(uniq_pcfa,decreasing=TRUE)),pos=4,col=color_5,cex=0.75)
```

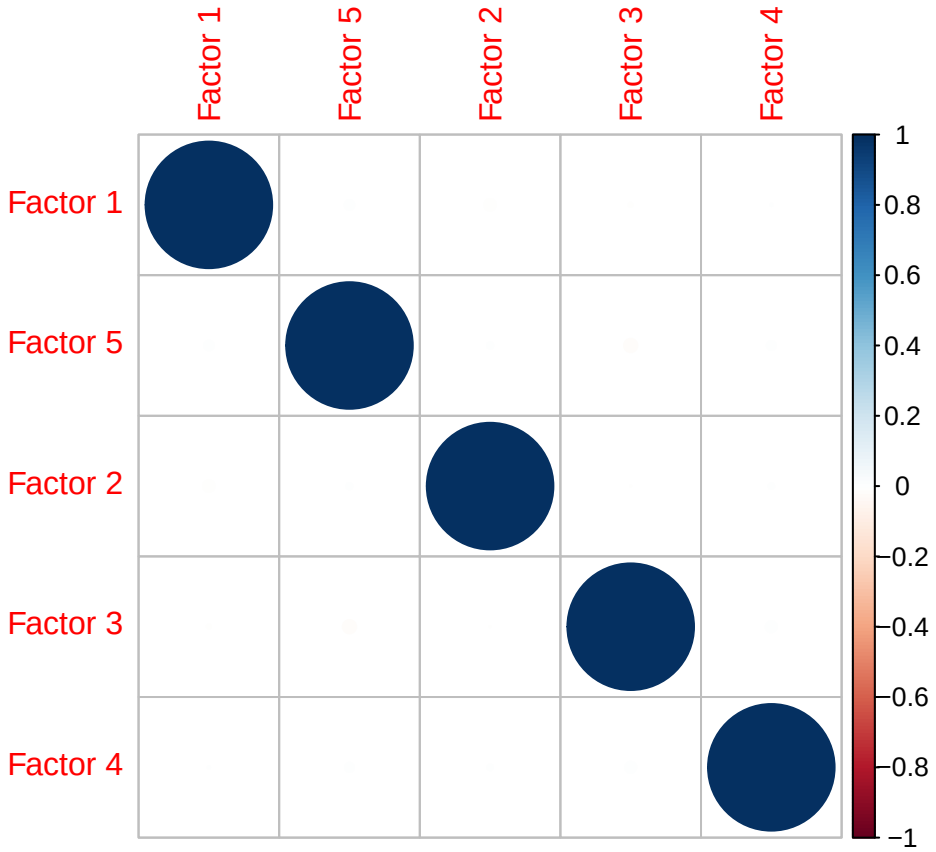


From the following plot we can tell that the factors are uncorrelated.

```r
# Estimate the factor scores
F_pcfa <- Y %*% solve(Sigma_nu_pcfa) %*% M_pcfa %*% solve(t(M_pcfa) %*% solve(Sigma_nu_pcfa) %*% M_pcfa)
colnames(F_pcfa) <- c("Factor 1","Factor 2","Factor 3","Factor 4","Factor 5")

pairs(F_pcfa,pch=19,col=color_1)
```



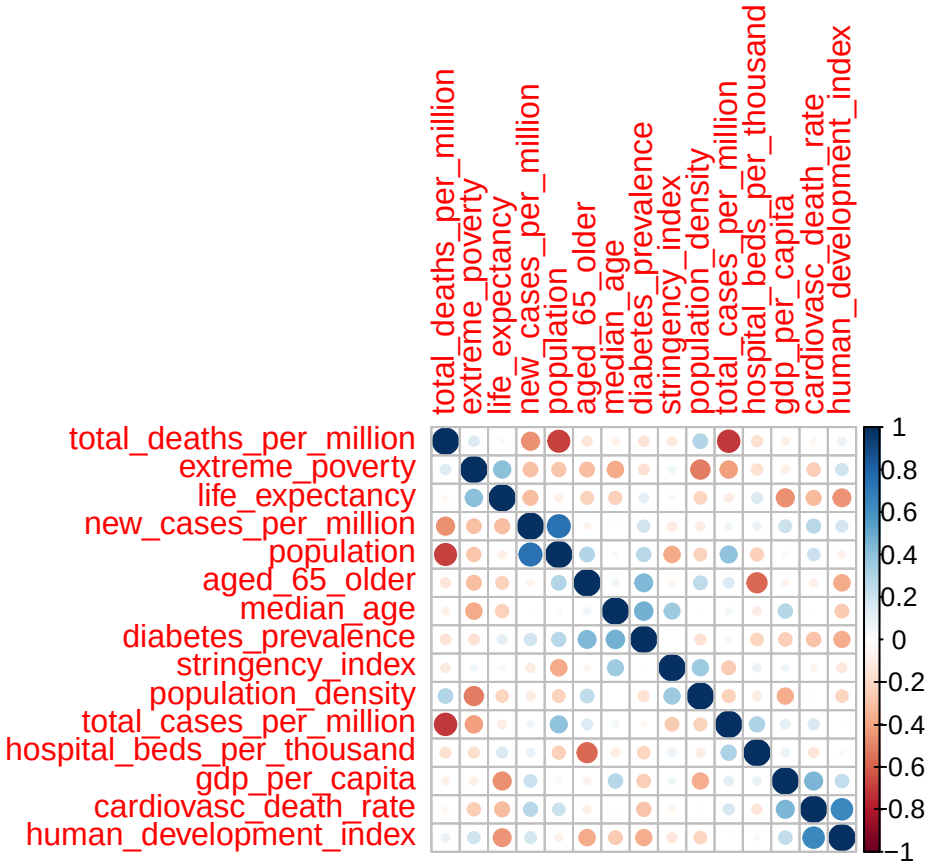

```
corrplot(cor(F_pcfa),order="hclust")
```



We plot the correlations between the residuals. We can see that in the following plot the correlations outside the diagonals the correlations are very low, except some of them, for instance the correlation between

populaion and new cases per million; total cases per million and total deaths per million. It could be that if we include another factor, we are going to be able to explain them, but as we only have 18 variables, we will only keep 5 factors.

```
# Estimate the residuals
Nu_pcfa <- Y - F_pcfa %*% t(M_pcfa)
corrplot(cor(Nu_pcfa),order="hclust")
```



Multidimensional Scaling

Dataset: Similarity of cocktails' popularity in a certain hotel's bar

The dataset contains how similar cocktails are in terms of popularity, the higher the similarity (between 1 and 0) the most similarly popular 2 drinks are.

A similarity of 1 = the cocktails are equally popular, similarity closer to 0 = one of the cocktails is significantly more popular than the other.

```
ctails <- read.csv('./data/cocktails.csv')
ctails <- ctails[,2:length(names(ctails))]
```

Transform similarity matrix into dissimilarity matrix

```
d_ctails <- sim2diss(ctails,method="reverse",to.dist=TRUE)
```

The highest dissimilarity pair is old fashioned and mojito (1), which is the largest possible dissimilarity.

Multidimensional scaling using cmdscale

We obtain the principal coordinates for $k = n - 1 = 8$

```
mds <- cmdscale(d_ctails, k=8, eig=TRUE)
```

We have 4 positive eigen values, 4 negative eigenvalues and 1 eigen value which is roughly 0.

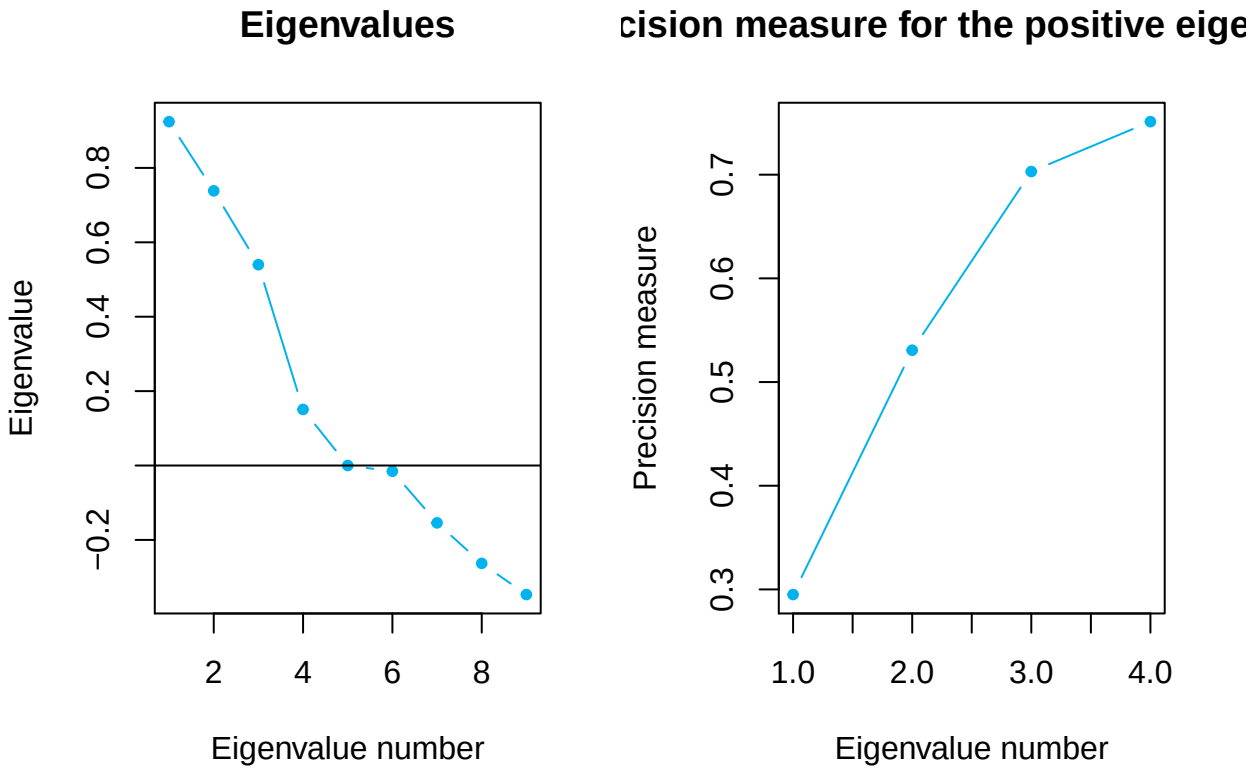
Here we see them rounded:

```
round(mds$eig,2)
#> [1]  0.92  0.74  0.54  0.15  0.00 -0.02 -0.15 -0.26 -0.35
```

We then obtain the precision measure for our positive eigen values (the first 4):

```
mds.m <- cumsum(mds$eig[1:4]/sum(abs(mds$eig)))
mds.m
#> [1] 0.2950 0.5307 0.7030 0.7512
```

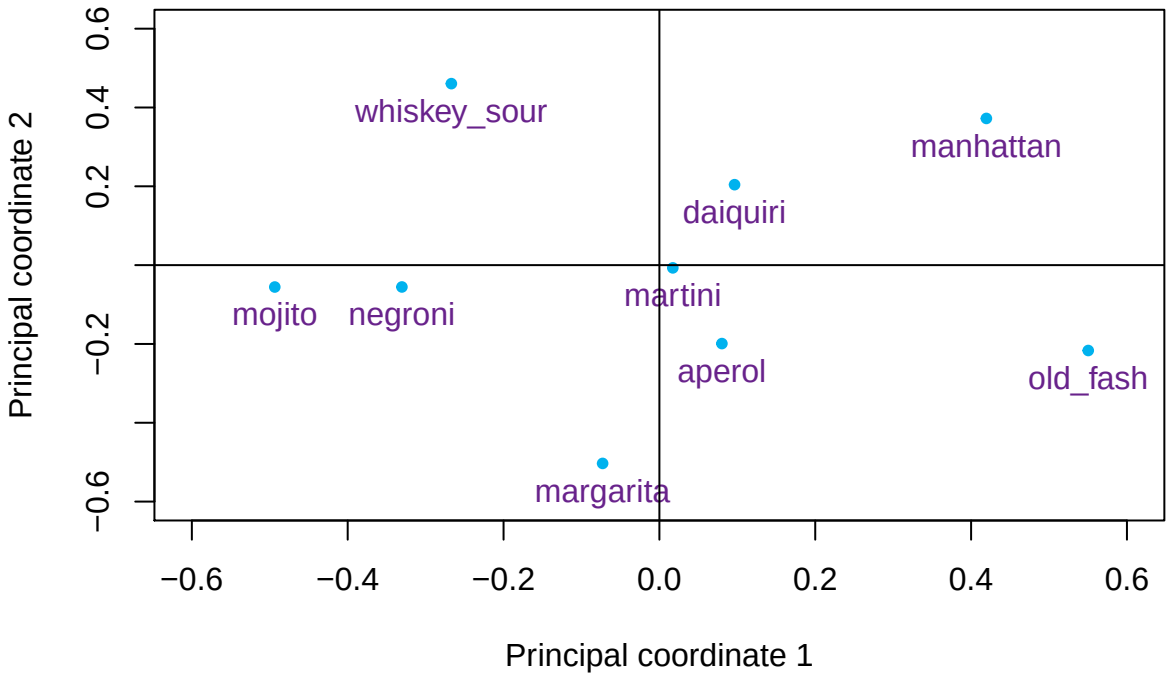
Then we plot the eigenvalues and the precision measure calculated previously:



We see that the first 3 PCs explain 70% of the variability, if we include the 4th PC, our explained variability goes up significantly, but not enough to reach 80%. Roughly ~76% of the variability.

In this case, unfortunately, we would want to preserve more PCs than we'd like to, making representing the data graphically difficult.

However, given that the first 2 PCs explain ~55% of the variability, we still take the liberty to plot the perceptual map of these two PCs:



We see that our cocktails are all quite spread around the plot, where some drinks like whiskey sour or manhattan are starkly separated along the PC2 axis from margarita, which might indicate that the popularity of the drinks might come from a different subset of clients. Perhaps allows bundling of some of these cocktails to target different groups of clients.

Old fashioned is also very particular, where its only similarity is shown in the PC2 with mojito but in a completely opposite manner in the PC1. Maybe this tells us that there's only a very particular type of customer that consumes the drink, a customer with a certain probability of also ordering mojito, but not necessarily doing so.

We could also interpret that the closer a drink is to the origin, the lesser the difference it'll have with all the other drinks and maybe will construct an image that random clients will try, hinting there might not be a specific group preference for them. This is, for example (according to our data), the case of martini, where anyone would probably try it as it does not exactly share identical levels of popularity with other drinks (with the exception of old fashioned, or in the opposite case, margarita). Either way, we could simply say that it is consistently popular.

Correspondence analysis

Given the following contigency table of each pair of classes corresponding to each variable (age group and health status), we will perform correspondence analysis:

Table 9: health table

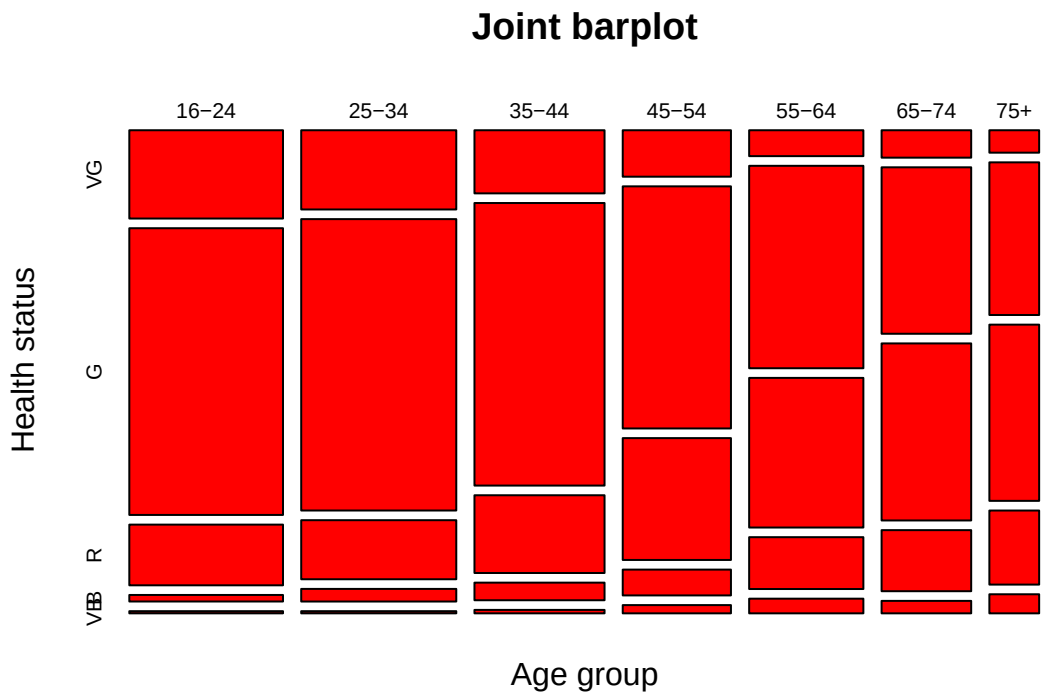
	VG	G	R	B	VB	Sum
16-24	243	789	167	18	6	1223
25-34	220	809	164	35	6	1234
35-44	147	658	181	41	8	1035
45-54	90	469	236	50	16	861
55-64	53	414	306	106	30	909
65-74	44	267	284	98	20	713
75+	20	136	157	66	17	396

	VG	G	R	B	VB	Sum
Sum	817	3542	1495	414	103	6371

Visual analysis of the data

We can see a graphical representation of the contingency table as follows:

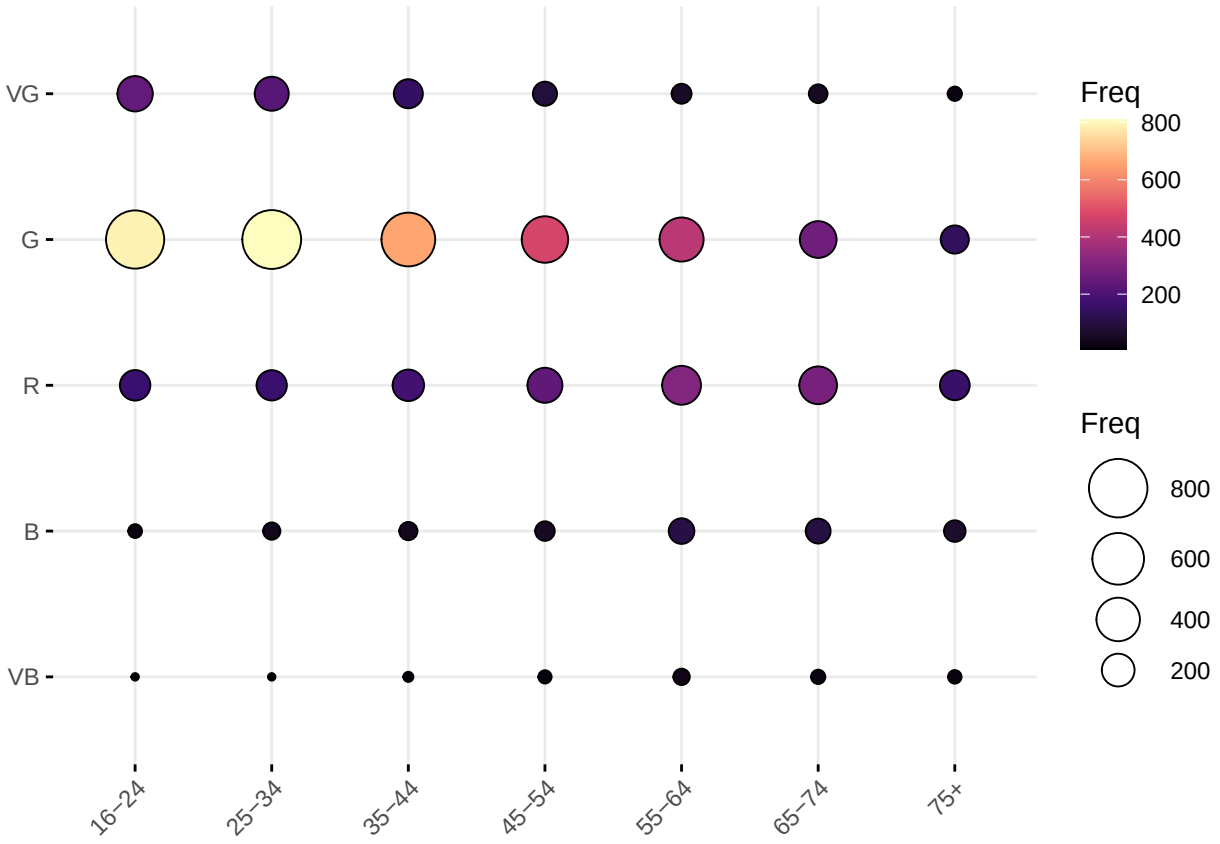
```
plot(health,xlab="Age group",ylab="Health status",col='red',main="Joint barplot")
```



In this joint barplot we can notice that the age groups are all more or less the same, with a small trend, where younger age groups tend to have a larger amount of individuals than older age groups.

We also notice that people with good health status are more abundant than the rest.

```
ggballoonplot(as.data.frame(health),fill="value")+scale_fill_viridis_c(option="A")
```



Our balloon plot tells us much of the same, the larger age groups are hte younger ones and, in proportion, there's a significantly larger amount of people with good/very good health status in younger age groups, than those with worse status within the same age group. The older the age group gets, the lesser the amount of individuals with good/very good health status, and the more with regular/bad health status.

Very bad health status individuals, while a very small subset of the general sample, are increasingly more common the older the age group is.

Therefore, there are differences in the sizes of groups, in general. But the differences are not too dramatic for age group sizes, the differences are more significant for health status groups. And there's definitely some relationship between the variables, or so we can infer from the plots.

Testing for independency between the variables

Relative proportion table (observed):

Table 10: health table

	VG	G	R	B	VB	Sum
16-24	0.0381	0.1238	0.0262	0.0028	0.0009	0.1920
25-34	0.0345	0.1270	0.0257	0.0055	0.0009	0.1937
35-44	0.0231	0.1033	0.0284	0.0064	0.0013	0.1625
45-54	0.0141	0.0736	0.0370	0.0078	0.0025	0.1351
55-64	0.0083	0.0650	0.0480	0.0166	0.0047	0.1427
65-74	0.0069	0.0419	0.0446	0.0154	0.0031	0.1119
75+	0.0031	0.0213	0.0246	0.0104	0.0027	0.0622
Sum	0.1282	0.5560	0.2347	0.0650	0.0162	1.0000

Here we can see how different groups are, the distribution of age groups is more even, however, for health status groups, "good" and "regular" gobble up over 70% of the observations.

Chi squared test (observed vs expected):

```
chisq.test(health)
#>
#>  Pearson's Chi-squared test
#>
#> data:  health
#> X-squared = 895, df = 24, p-value <2e-16
```

We get a p-value of <2e-16, which means that there's a significant dependence between the age group and health status variables.

Correspondence analysis for the data matrix

First of all we calculate the total relative frequencies for rows/cols:

```
rel_freq_rows <- rowSums(health_rf)
rel_freq_cols <- colSums(health_rf)
```

We create a matrix of zeros where the diagonal is the sum of the rows of our relative frequency matrix and we do the same for the columns.

```
diag_rs <- diag(rel_freq_rows)
diag_cs <- diag(rel_freq_cols)
```

We the compute the matrices of row and column profiles:

```
prof_rs <- solve(diag_rs) %*% health_rf
apply(prof_rs, 1, sum)
#> [1] 1 1 1 1 1 1 1
prof_cs <- solve(diag_cs) %*% t(health_rf)
apply(prof_cs, 1, sum)
#> [1] 1 1 1 1 1
```

We compute the matrix M and its SVD:

```
M <- diag(1/sqrt(rel_freq_rows)) %*% (health_rf - rel_freq_rows %*% t(rel_freq_cols)) %*% diag(1/sqrt(r
M_svd <- svd(M)
```

We then define the Lambda, Gamma and Theta matrices:

```
Lambda_M <- diag(M_svd$d)
Gamma_M <- M_svd$u
Theta_M <- M_svd$v
```

And we obtain each matrix:

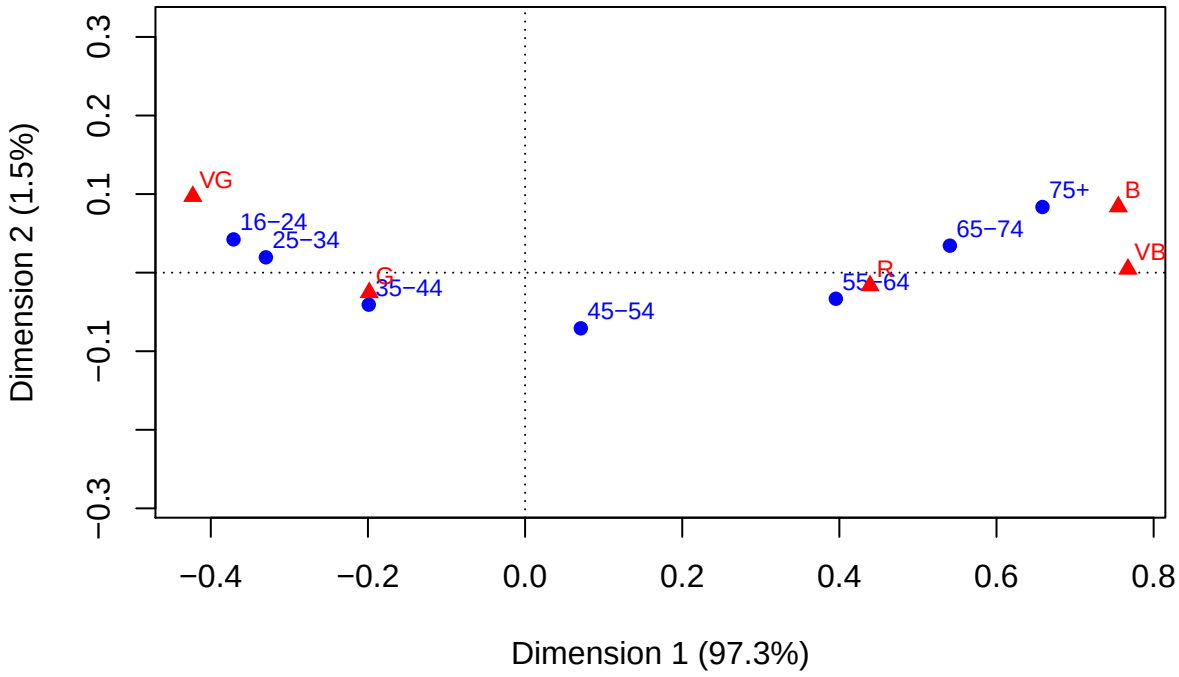
```
X_r <- diag(1/sqrt(rel_freq_rows)) %*% Gamma_M[,1:2] %*% Lambda_M[1:2,1:2]
X_r
#>          [,1]     [,2]
#> [1,] -0.37107  0.04231
#> [2,] -0.32988  0.01951
#> [3,] -0.19895 -0.04075
#> [4,]  0.07091 -0.07086
#> [5,]  0.39552 -0.03325
#> [6,]  0.54064  0.03435
#> [7,]  0.65849  0.08357
```

```
X_c <- diag(1/sqrt(rel_freq_cols)) %*% Theta_M[,1:2] %*% Lambda_M[1:2,1:2]
X_c
#>          [,1]      [,2]
#> [1,] -0.4229  0.097119
#> [2,] -0.1983 -0.025361
#> [3,]  0.4391 -0.016546
#> [4,]  0.7550  0.083936
#> [5,]  0.7673  0.004554
```

Library 'ca' and conclusions

Utilizing the library 'ca' we can perform the same analysis in a more speedy manner:

```
ca_ages_status <- ca(health)
plot(ca_ages_status)
```



From this we can see a few things:

- Clearly, most of these classes are dependent on each other
- Very good health status is strongly dependent on the respondant being younger (16-24 and 25-34)
- Good health status is also strongly dependent on people being relatively younger, but perhaps more than anything it's closer to group 35-44. Either way though, we can't underestimate good/very good's health status' dependence on youth overall.
- Bad and very bad health status are often strongly dependent on older ages, especially 75+
- Regular health status has a significant dependence on the respondants being 55-64 years of age. It seems like a decent way for this group to differentiate itself from the rest, where we can especulate that the respondants are not confident on their health status enough to say that they're in good or bad condition. We can also infer that many long-term health conditions that are mildly deteriorating are already somewhat developed by this age, conditions like vision issues (i.e. developed myopia), arthritis, osteoporosis and some heart conditions are either starting to be developed around this age or are already developed to significantly developed, therefore maybe skewing the individuals' perspective of their own health status.
- Age group 45-54 seems to be in a midpoint where no particular health status is dependent on it in any

significant way, these people may or may not consider themselves in good health, but overall, it's a bit of a tossup between people with regular health status and good health status among individuals in this group.