Chapter 5: Dimensionality reduction techniques

```
# Part 1. Graphical overview and summaries of the data


library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(readr)
library(corrplot)

## corrplot 0.92 loaded

library(tibble)
library(GGally)

## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

human <- read_csv("human.csv")

## New names:
## • `` -> `...1`

## Rows: 155 Columns: 10
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Country
## dbl (9): ...1, Edu2.FM, Labo.FM, Life.Exp, Edu.Exp, GNI, Mat.Mor, Ado.Birth,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

keep <- c("Country", "Edu2.FM", "Labo.FM", "Life.Exp", "Edu.Exp", "GNI",
"Mat.Mor", "Ado.Birth", "Parli.F")
```

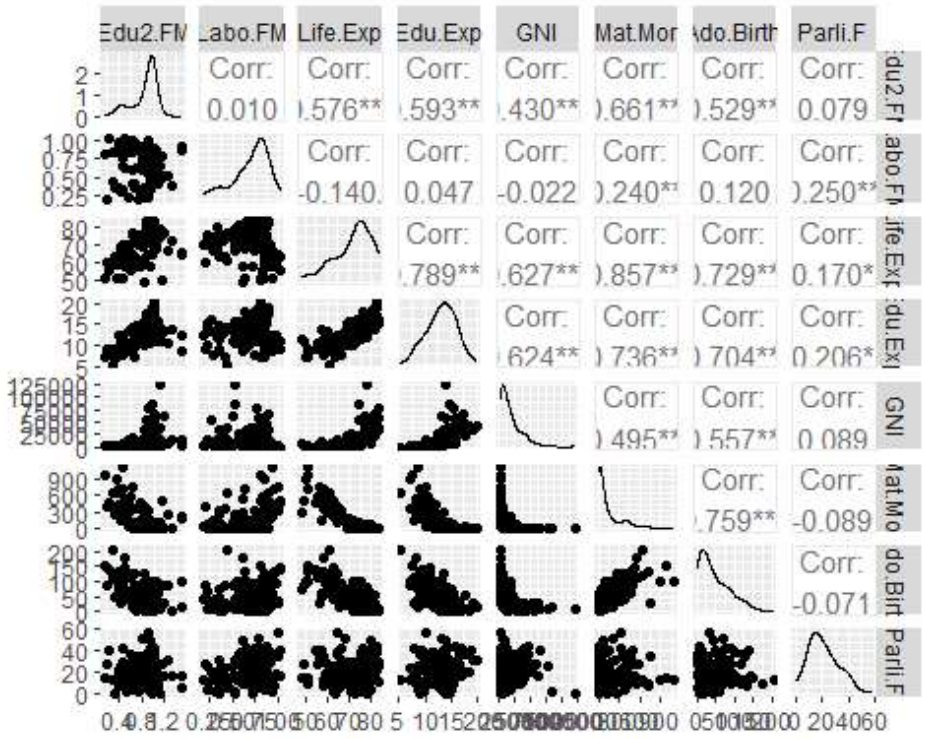
```
human <- select(human, one_of(keep))
dim(human)
```

```
## [1] 155   9
```

```
colnames(human)
```

```
## [1] "Country"   "Edu2.FM"   "Labo.FM"   "Life.Exp"  "Edu.Exp"   "GNI"
## [7] "Mat.Mor"   "Ado.Birth" "Parli.F"
```

```
# Moving the country names to rownames
human_ <- column_to_rownames(human, "Country")

# Visualizing the 'human_' variables
ggpairs(human_, progress=FALSE)
```




```
colnames(human)
```

```
## [1] "Country"   "Edu2.FM"   "Labo.FM"   "Life.Exp"  "Edu.Exp"   "GNI"
## [7] "Mat.Mor"   "Ado.Birth" "Parli.F"
```

```
# looking at summaries of the data set
summary(human_)
```

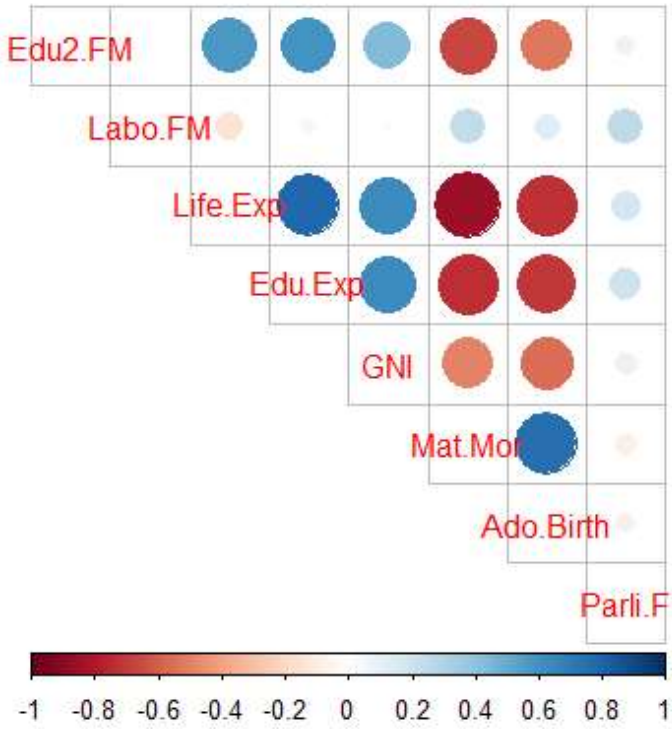
```
##     Edu2.FM          Labo.FM          Life.Exp        Edu.Exp
##  Min.   :0.1717   Min.   :0.1857   Min.   :49.00   Min.   : 5.40
##  1st Qu.:0.7264   1st Qu.:0.5984   1st Qu.:66.30   1st Qu.:11.25
##  Median :0.9375   Median :0.7535   Median :74.20   Median :13.50
```

```
##  Mean   :0.8529   Mean   :0.7074   Mean   :71.65   Mean   :13.18
##  3rd Qu.:0.9968   3rd Qu.:0.8535   3rd Qu.:77.25   3rd Qu.:15.20
##  Max.   :1.4967   Max.   :1.0380   Max.   :83.50   Max.   :20.20
##       GNI            Mat.Mor         Ado.Birth         Parli.F
##  Min.   :   581   Min.   :   1.0   Min.   :  0.60   Min.   : 0.00
##  1st Qu.:  4198   1st Qu.:  11.5   1st Qu.: 12.65   1st Qu.:12.40
##  Median : 12040   Median :  49.0   Median : 33.60   Median :19.30
##  Mean   : 17628   Mean   : 149.1   Mean   : 47.16   Mean   :20.91
##  3rd Qu.: 24512   3rd Qu.: 190.0   3rd Qu.: 71.95   3rd Qu.:27.95
##  Max.   :123124   Max.   :1100.0   Max.   :204.80   Max.   :57.50
```

```
# Computing the correlation matrix and visualizing it with corrplot
cor_matrix <- cor(human_)
cor_matrix
```

```
##                Edu2.FM      Labo.FM   Life.Exp     Edu.Exp         GNI
## Edu2.FM    1.000000000  0.009564039  0.5760299  0.59325156  0.43030485
## Labo.FM    0.009564039  1.000000000 -0.1400125  0.04732183 -0.02173971
## Life.Exp   0.576029853 -0.140012504  1.0000000  0.78943917  0.62666411
## Edu.Exp    0.593251562  0.047321827  0.7894392  1.00000000  0.62433940
## GNI        0.430304846 -0.021739705  0.6266641  0.62433940  1.00000000
## Mat.Mor   -0.660931770  0.240461075 -0.8571684 -0.73570257 -0.49516234
## Ado.Birth -0.529418415  0.120158862 -0.7291774 -0.70356489 -0.55656208
## Parli.F    0.078635285  0.250232608  0.1700863  0.20608156  0.08920818
##              Mat.Mor  Ado.Birth     Parli.F
## Edu2.FM   -0.6609318 -0.5294184  0.07863528
## Labo.FM    0.2404611  0.1201589  0.25023261
## Life.Exp  -0.8571684 -0.7291774  0.17008631
## Edu.Exp   -0.7357026 -0.7035649  0.20608156
## GNI       -0.4951623 -0.5565621  0.08920818
## Mat.Mor    1.0000000  0.7586615 -0.08944000
## Ado.Birth  0.7586615  1.0000000 -0.07087810
## Parli.F   -0.0894400 -0.0708781  1.00000000
```

```
corrplot(cor_matrix, method="circle", type = "upper", cl.pos = "b", tl.pos = "d")
```



Part 2-4. principal component analysis (PCA)

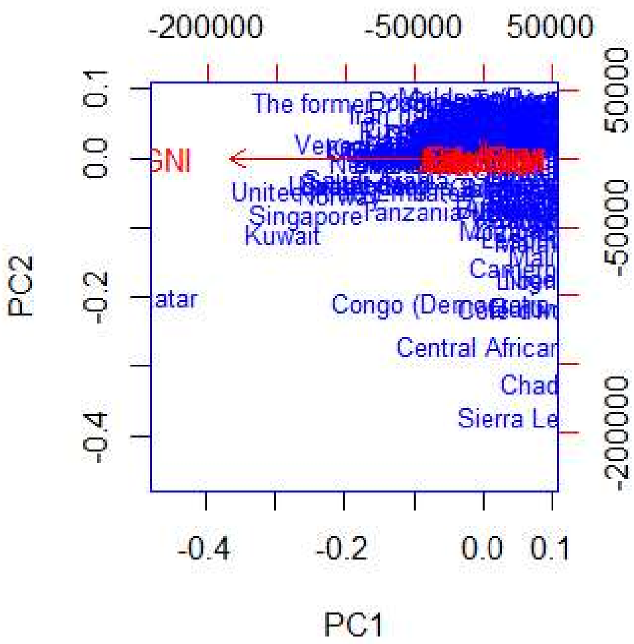
Principal component analysis (PCA) is a technique for analyzing large datasets containing a high number of dimensions. PCA helps reduce the dimensionality of a dataset by linearly transforming the data into a new coordinate system where most of the variation in the data can be described with fewer dimensions.

A biplot can be used to visualize connections between two representations of the same data. Here, the two principal components are vizualized for PC1 coordinate in x-axis and PC2 coordinate in y-axis. The arrows in the graph showcase connections between the original variables and the PC's.

The angle between the arrows can be interpreted as the correlation between the variables. The angle between a variable and a PC axis can be interpreted as the correlation between the two. The length of the arrows are proportional to the standard deviations of the variables.

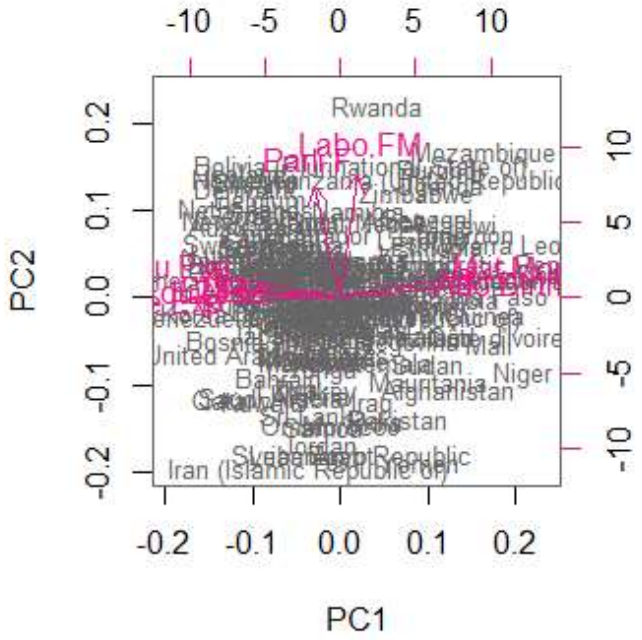
```
library(tibble)
library(readr)

# perform principal component analysis on unscaled data

pca_human <- prcomp(human_)

# draw a biplot of the principal component representation and the original
variables
biplot(pca_human, choices = 1:2, col = c("blue", "red"), cex = c(0.8, 1))
```

```
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```




```
# create and print out a summary of pca_human
s <- summary(pca_human)

# rounded percentanges of variance captured by each PC
pca_pr <- round(1*s$importance[2, ], digits = 5)

pca_pr

##    PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## 0.9999 0.0001 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
# For the unscaled data, the PCA output creates large weight to the first PC with
close to no weight on the others.


# Perform principal component analysis on scaled data

human_std <- scale(human_)
pca_human2 <- prcomp(human_std)

# draw a biplot of the principal component representation and the original
variables
biplot(pca_human2, choices = 1:2, col = c("grey40", "deeppink2"), cex = c(0.8, 1))
```




```
# create and print out a summary of pca_human
s2 <- summary(pca_human2)

# rounded percentanges of variance captured by each PC
pca_pr2 <- round(1*s2$importance[2, ], digits = 5)

pca_pr2
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
## 0.53605 0.16237 0.09571 0.07583 0.05477 0.03595 0.02634 0.01298
```

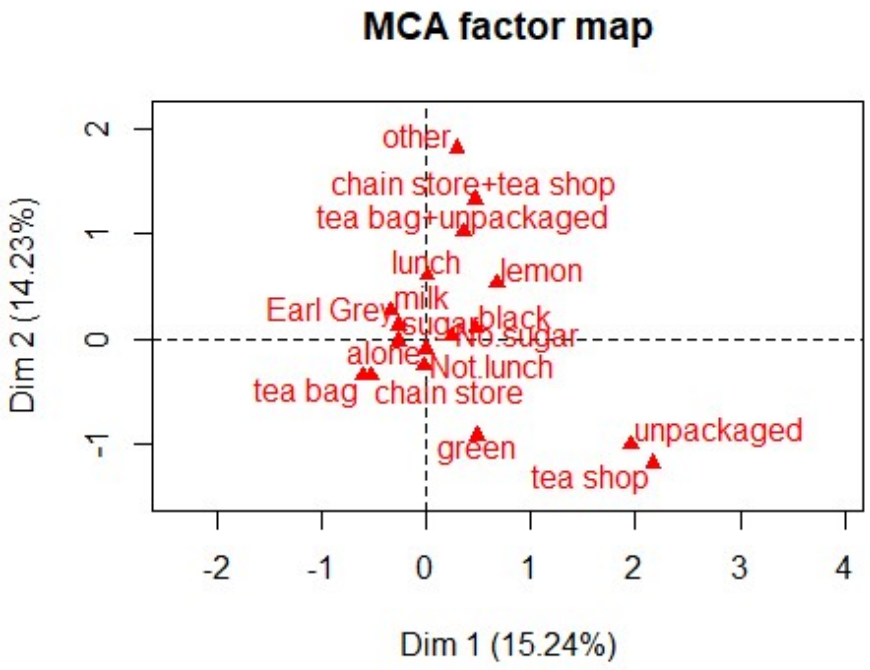
```
# PCA for scaled data shows that more half (54 %) of the variability in the data
is included in the PC1 and around 16 percent in the PCA2.
```

Chapter 5: MCA

```
## Part 5: Multiple Correspondence Analysis (MCA) on the tea data

#Multiple correspondence analysis (MCA) is a data analysis technique for
categorical data, which can be used to detect and represent underlying structures
in a data set. MCA can be considered to be the counterpart of PCA for categorical
data.



library(dplyr)
library(tidyr)
library(ggplot2)
tea_time <- read.csv("https://raw.githubusercontent.com/KimmoVehkalahti/Helsinki-
Open-Data-Science/master/datasets/tea_time.csv", stringsAsFactors = TRUE)



library(FactoMineR)

# Looking at the tea dataset
dim(tea_time)

## [1] 300   6

str(tea_time)

## 'data.frame':    300 obs. of  6 variables:
##  $ Tea  : Factor w/ 3 levels "black","Earl Grey",..: 1 1 2 2 2 2 2 1 2 1 ...
##  $ How  : Factor w/ 4 levels "alone","lemon",..: 1 3 1 1 1 1 1 3 3 1 ...
##  $ how  : Factor w/ 3 levels "tea bag","tea bag+unpackaged",..: 1 1 1 1 1 1 1 1
2 2 ...
##  $ sugar: Factor w/ 2 levels "No.sugar","sugar": 2 1 1 2 1 1 1 1 1 1 ...
##  $ where: Factor w/ 3 levels "chain store",..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ lunch: Factor w/ 2 levels "lunch","Not.lunch": 2 2 2 2 2 2 2 2 2 2 ...

# The dataset includes categorical variables

# multiple correspondence analysis
mca <- MCA(tea_time, graph = FALSE)

# summary of the model
mca

## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 300 individuals, described by 6 variables
## *The results are available in the following objects:
##
##    name              description
## 1  "$eig"            "eigenvalues"
## 2  "$var"            "results for the variables"
## 3  "$var$coord"      "coord. of the categories"
```

```
## 4  "$var$cos2"       "cos2 for the categories"
## 5  "$var$contrib"    "contributions of the categories"
## 6  "$var$v.test"     "v-test for the categories"
## 7  "$var$eta2"       "coord. of variables"
## 8  "$ind"            "results for the individuals"
## 9  "$ind$coord"      "coord. for the individuals"
## 10 "$ind$cos2"       "cos2 for the individuals"
## 11 "$ind$contrib"    "contributions of the individuals"
## 12 "$call"           "intermediate results"
## 13 "$call$marge.col" "weights of columns"
## 14 "$call$marge.li"  "weights of rows"
```

```
# visualize MCA
plot(mca, invisible=c("ind"), graph.type = "classic")
```




```
date()
```

```
## [1] "Mon Dec  4 13:32:49 2023"
```