Chapter 5

Mohammad Imangholiloo

18 November 2020

# About the project

**Welcome to my page**

This is the learning diary for the course ‘Introduction to Open Data Science 2020’. I would like to further develop my data science skills in R and get familiar with Github.  
*My GitHub repository is* [*https://github.com/Imangholiloo/IODS-project*](https://github.com/Imangholiloo/IODS-project)

# Let's get the date and time  
date()

## [1] "Wed Nov 25 19:27:13 2020"

alternatively, print(Sys.time())

## ————————————————————————————

# *Exercise 5:* **Dimensionality reduction techniques**

Mohamamd Imangholiloo  
*My GitHub repository is* [*https://github.com/Imangholiloo/IODS-project*](https://github.com/Imangholiloo/IODS-project)  
Nowadays, it’s common to have huge datasets (like Big Data) in which there could be multiple variables with high correlation with each other, or noise within the data. Meaning that, one variable can explain more of the other one. Thus, it is good to reduce the high dimensionality of our data.  
There are many available methods, but we will learn about principal component analysis (PCA), which reduces any number of measured (continuous) and correlated variables into a few uncorrelated components that collect together as much variance as possible from the original variables. And Multiple correspondence analysis (MCA) which gives us similar possibilities in the world of discrete variables, even nominal scale (classified) variables, by finding a suitable transformation into continuous scales and then reducing the dimensions quite analogously with the PCA. *More info:* [*https://vimeo.com/204164956*](https://vimeo.com/204164956)

## *Part 1) Data wrangling*

#Read the “Human development” file into R  
hd <- read.csv("http://s3.amazonaws.com/assets.datacamp.com/production/course\_2218/datasets/human\_development.csv", stringsAsFactors = F)  
  
#Read the “Gender inequality” file into R  
gii <- read.csv("http://s3.amazonaws.com/assets.datacamp.com/production/course\_2218/datasets/gender\_inequality.csv", stringsAsFactors = F, na.strings = "..")

For data description and more info please visit: <http://hdr.undp.org/en/content/human-development-index-hdi> and <http://hdr.undp.org/sites/default/files/hdr2015_technical_notes.pdf>

#strcutre of data  
str(hd)

## 'data.frame': 195 obs. of 8 variables:  
## $ HDI.Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ Country : chr "Norway" "Australia" "Switzerland" "Denmark" ...  
## $ Human.Development.Index..HDI. : num 0.944 0.935 0.93 0.923 0.922 0.916 0.916 0.915 0.913 0.913 ...  
## $ Life.Expectancy.at.Birth : num 81.6 82.4 83 80.2 81.6 80.9 80.9 79.1 82 81.8 ...  
## $ Expected.Years.of.Education : num 17.5 20.2 15.8 18.7 17.9 16.5 18.6 16.5 15.9 19.2 ...  
## $ Mean.Years.of.Education : num 12.6 13 12.8 12.7 11.9 13.1 12.2 12.9 13 12.5 ...  
## $ Gross.National.Income..GNI..per.Capita: chr "64,992" "42,261" "56,431" "44,025" ...  
## $ GNI.per.Capita.Rank.Minus.HDI.Rank : int 5 17 6 11 9 11 16 3 11 23 ...

#dimension of file (row\* column)  
dim(hd)

## [1] 195 8

#summary of file  
summary(hd)

## HDI.Rank Country Human.Development.Index..HDI.  
## Min. : 1.00 Length:195 Min. :0.3480   
## 1st Qu.: 47.75 Class :character 1st Qu.:0.5770   
## Median : 94.00 Mode :character Median :0.7210   
## Mean : 94.31 Mean :0.6918   
## 3rd Qu.:141.25 3rd Qu.:0.8000   
## Max. :188.00 Max. :0.9440   
## NA's :7   
## Life.Expectancy.at.Birth Expected.Years.of.Education Mean.Years.of.Education  
## Min. :49.00 Min. : 4.10 Min. : 1.400   
## 1st Qu.:65.75 1st Qu.:11.10 1st Qu.: 5.550   
## Median :73.10 Median :13.10 Median : 8.400   
## Mean :71.07 Mean :12.86 Mean : 8.079   
## 3rd Qu.:76.80 3rd Qu.:14.90 3rd Qu.:10.600   
## Max. :84.00 Max. :20.20 Max. :13.100   
##   
## Gross.National.Income..GNI..per.Capita GNI.per.Capita.Rank.Minus.HDI.Rank  
## Length:195 Min. :-84.0000   
## Class :character 1st Qu.: -9.0000   
## Mode :character Median : 1.5000   
## Mean : 0.1862   
## 3rd Qu.: 11.0000   
## Max. : 47.0000   
## NA's :7

As you can see, our columns are all numerical, except two columns of country and GNI\_per\_capita, which are class

#column names  
colnames(hd)

## [1] "HDI.Rank"   
## [2] "Country"   
## [3] "Human.Development.Index..HDI."   
## [4] "Life.Expectancy.at.Birth"   
## [5] "Expected.Years.of.Education"   
## [6] "Mean.Years.of.Education"   
## [7] "Gross.National.Income..GNI..per.Capita"  
## [8] "GNI.per.Capita.Rank.Minus.HDI.Rank"

As you can see, column names are long, so in next phases, we will shorten the names.

#see head and tail of the data  
head(hd)

## HDI.Rank Country Human.Development.Index..HDI. Life.Expectancy.at.Birth  
## 1 1 Norway 0.944 81.6  
## 2 2 Australia 0.935 82.4  
## 3 3 Switzerland 0.930 83.0  
## 4 4 Denmark 0.923 80.2  
## 5 5 Netherlands 0.922 81.6  
## 6 6 Germany 0.916 80.9  
## Expected.Years.of.Education Mean.Years.of.Education  
## 1 17.5 12.6  
## 2 20.2 13.0  
## 3 15.8 12.8  
## 4 18.7 12.7  
## 5 17.9 11.9  
## 6 16.5 13.1  
## Gross.National.Income..GNI..per.Capita GNI.per.Capita.Rank.Minus.HDI.Rank  
## 1 64,992 5  
## 2 42,261 17  
## 3 56,431 6  
## 4 44,025 11  
## 5 45,435 9  
## 6 43,919 11

tail(hd)

## HDI.Rank Country Human.Development.Index..HDI.  
## 190 NA East Asia and the Pacific 0.710  
## 191 NA Europe and Central Asia 0.748  
## 192 NA Latin America and the Caribbean 0.748  
## 193 NA South Asia 0.607  
## 194 NA Sub-Saharan Africa 0.518  
## 195 NA World 0.711  
## Life.Expectancy.at.Birth Expected.Years.of.Education  
## 190 74.0 12.7  
## 191 72.3 13.6  
## 192 75.0 14.0  
## 193 68.4 11.2  
## 194 58.5 9.6  
## 195 71.5 12.2  
## Mean.Years.of.Education Gross.National.Income..GNI..per.Capita  
## 190 7.5 11,449  
## 191 10.0 12,791  
## 192 8.2 14,242  
## 193 5.5 5,605  
## 194 5.2 3,363  
## 195 7.9 14,301  
## GNI.per.Capita.Rank.Minus.HDI.Rank  
## 190 NA  
## 191 NA  
## 192 NA  
## 193 NA  
## 194 NA  
## 195 NA

#Let's do the same for gii dataset  
#strcutre of data  
str(gii)

## 'data.frame': 195 obs. of 10 variables:  
## $ GII.Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ Country : chr "Norway" "Australia" "Switzerland" "Denmark" ...  
## $ Gender.Inequality.Index..GII. : num 0.067 0.11 0.028 0.048 0.062 0.041 0.113 0.28 0.129 0.157 ...  
## $ Maternal.Mortality.Ratio : int 4 6 6 5 6 7 9 28 11 8 ...  
## $ Adolescent.Birth.Rate : num 7.8 12.1 1.9 5.1 6.2 3.8 8.2 31 14.5 25.3 ...  
## $ Percent.Representation.in.Parliament : num 39.6 30.5 28.5 38 36.9 36.9 19.9 19.4 28.2 31.4 ...  
## $ Population.with.Secondary.Education..Female.: num 97.4 94.3 95 95.5 87.7 96.3 80.5 95.1 100 95 ...  
## $ Population.with.Secondary.Education..Male. : num 96.7 94.6 96.6 96.6 90.5 97 78.6 94.8 100 95.3 ...  
## $ Labour.Force.Participation.Rate..Female. : num 61.2 58.8 61.8 58.7 58.5 53.6 53.1 56.3 61.6 62 ...  
## $ Labour.Force.Participation.Rate..Male. : num 68.7 71.8 74.9 66.4 70.6 66.4 68.1 68.9 71 73.8 ...

#dimension of file (row\* column)  
dim(gii)

## [1] 195 10

#summary of file  
summary(gii)

## GII.Rank Country Gender.Inequality.Index..GII.  
## Min. : 1.00 Length:195 Min. :0.0160   
## 1st Qu.: 47.75 Class :character 1st Qu.:0.2030   
## Median : 94.00 Mode :character Median :0.3935   
## Mean : 94.31 Mean :0.3695   
## 3rd Qu.:141.25 3rd Qu.:0.5272   
## Max. :188.00 Max. :0.7440   
## NA's :7 NA's :33   
## Maternal.Mortality.Ratio Adolescent.Birth.Rate  
## Min. : 1.0 Min. : 0.60   
## 1st Qu.: 16.0 1st Qu.: 15.45   
## Median : 69.0 Median : 40.95   
## Mean : 163.2 Mean : 49.55   
## 3rd Qu.: 230.0 3rd Qu.: 71.78   
## Max. :1100.0 Max. :204.80   
## NA's :10 NA's :5   
## Percent.Representation.in.Parliament  
## Min. : 0.00   
## 1st Qu.:12.47   
## Median :19.50   
## Mean :20.60   
## 3rd Qu.:27.02   
## Max. :57.50   
## NA's :3   
## Population.with.Secondary.Education..Female.  
## Min. : 0.9   
## 1st Qu.: 27.8   
## Median : 55.7   
## Mean : 54.8   
## 3rd Qu.: 81.8   
## Max. :100.0   
## NA's :26   
## Population.with.Secondary.Education..Male.  
## Min. : 3.20   
## 1st Qu.: 38.30   
## Median : 60.00   
## Mean : 60.29   
## 3rd Qu.: 85.80   
## Max. :100.00   
## NA's :26   
## Labour.Force.Participation.Rate..Female.  
## Min. :13.50   
## 1st Qu.:44.50   
## Median :53.30   
## Mean :52.61   
## 3rd Qu.:62.62   
## Max. :88.10   
## NA's :11   
## Labour.Force.Participation.Rate..Male.  
## Min. :44.20   
## 1st Qu.:68.88   
## Median :75.55   
## Mean :74.74   
## 3rd Qu.:80.15   
## Max. :95.50   
## NA's :11

As summary shows, all variables, except country, are numerical variable

#column names  
colnames(gii)

## [1] "GII.Rank"   
## [2] "Country"   
## [3] "Gender.Inequality.Index..GII."   
## [4] "Maternal.Mortality.Ratio"   
## [5] "Adolescent.Birth.Rate"   
## [6] "Percent.Representation.in.Parliament"   
## [7] "Population.with.Secondary.Education..Female."  
## [8] "Population.with.Secondary.Education..Male."   
## [9] "Labour.Force.Participation.Rate..Female."   
## [10] "Labour.Force.Participation.Rate..Male."

#see head and tail of the data  
head(gii)

## GII.Rank Country Gender.Inequality.Index..GII. Maternal.Mortality.Ratio  
## 1 1 Norway 0.067 4  
## 2 2 Australia 0.110 6  
## 3 3 Switzerland 0.028 6  
## 4 4 Denmark 0.048 5  
## 5 5 Netherlands 0.062 6  
## 6 6 Germany 0.041 7  
## Adolescent.Birth.Rate Percent.Representation.in.Parliament  
## 1 7.8 39.6  
## 2 12.1 30.5  
## 3 1.9 28.5  
## 4 5.1 38.0  
## 5 6.2 36.9  
## 6 3.8 36.9  
## Population.with.Secondary.Education..Female.  
## 1 97.4  
## 2 94.3  
## 3 95.0  
## 4 95.5  
## 5 87.7  
## 6 96.3  
## Population.with.Secondary.Education..Male.  
## 1 96.7  
## 2 94.6  
## 3 96.6  
## 4 96.6  
## 5 90.5  
## 6 97.0  
## Labour.Force.Participation.Rate..Female.  
## 1 61.2  
## 2 58.8  
## 3 61.8  
## 4 58.7  
## 5 58.5  
## 6 53.6  
## Labour.Force.Participation.Rate..Male.  
## 1 68.7  
## 2 71.8  
## 3 74.9  
## 4 66.4  
## 5 70.6  
## 6 66.4

tail(gii)

## GII.Rank Country Gender.Inequality.Index..GII.  
## 190 NA East Asia and the Pacific 0.328  
## 191 NA Europe and Central Asia 0.300  
## 192 NA Latin America and the Caribbean 0.415  
## 193 NA South Asia 0.536  
## 194 NA Sub-Saharan Africa 0.575  
## 195 NA World 0.449  
## Maternal.Mortality.Ratio Adolescent.Birth.Rate  
## 190 72 21.2  
## 191 28 30.8  
## 192 85 68.3  
## 193 183 38.7  
## 194 506 109.7  
## 195 210 47.4  
## Percent.Representation.in.Parliament  
## 190 18.7  
## 191 19.0  
## 192 27.0  
## 193 17.5  
## 194 22.5  
## 195 21.8  
## Population.with.Secondary.Education..Female.  
## 190 54.7  
## 191 70.8  
## 192 54.3  
## 193 29.1  
## 194 22.1  
## 195 54.5  
## Population.with.Secondary.Education..Male.  
## 190 66.3  
## 191 80.6  
## 192 55.2  
## 193 54.6  
## 194 31.5  
## 195 65.4  
## Labour.Force.Participation.Rate..Female.  
## 190 62.6  
## 191 45.6  
## 192 53.7  
## 193 29.8  
## 194 65.4  
## 195 50.3  
## Labour.Force.Participation.Rate..Male.  
## 190 79.4  
## 191 70.0  
## 192 79.8  
## 193 80.3  
## 194 76.6  
## 195 76.7

**Task 4** : rename the variables with (shorter) descriptive names

colnames(hd)

## [1] "HDI.Rank"   
## [2] "Country"   
## [3] "Human.Development.Index..HDI."   
## [4] "Life.Expectancy.at.Birth"   
## [5] "Expected.Years.of.Education"   
## [6] "Mean.Years.of.Education"   
## [7] "Gross.National.Income..GNI..per.Capita"  
## [8] "GNI.per.Capita.Rank.Minus.HDI.Rank"

colnames(hd) <- c('HDI\_Rank', 'Country', 'HDI', 'Life\_Expec\_Birth',   
 'Expec\_yr\_Edu', 'Mean\_yr\_Edu', 'GNI\_per\_Cap',   
 'GNI\_per\_Cap\_Min\_HDI\_Rank')  
colnames(hd)

## [1] "HDI\_Rank" "Country"   
## [3] "HDI" "Life\_Expec\_Birth"   
## [5] "Expec\_yr\_Edu" "Mean\_yr\_Edu"   
## [7] "GNI\_per\_Cap" "GNI\_per\_Cap\_Min\_HDI\_Rank"

so you can see that column names were shorten.

**Let’s do the same fo gii dataset**

colnames(gii)

## [1] "GII.Rank"   
## [2] "Country"   
## [3] "Gender.Inequality.Index..GII."   
## [4] "Maternal.Mortality.Ratio"   
## [5] "Adolescent.Birth.Rate"   
## [6] "Percent.Representation.in.Parliament"   
## [7] "Population.with.Secondary.Education..Female."  
## [8] "Population.with.Secondary.Education..Male."   
## [9] "Labour.Force.Participation.Rate..Female."   
## [10] "Labour.Force.Participation.Rate..Male."

colnames(gii) <- c('GII\_Rank', 'Country', 'GII', 'Mortality\_R',   
 'Adolescent\_Birth\_R', 'Representation\_Parliament',  
 'Sec\_Edu\_Fem','Sec\_Edu\_Mal',  
 "Lab\_Forc\_Particip\_R\_Fem",  
 "Lab\_Forc\_Particip\_R\_Mal" )  
colnames(gii)

## [1] "GII\_Rank" "Country"   
## [3] "GII" "Mortality\_R"   
## [5] "Adolescent\_Birth\_R" "Representation\_Parliament"  
## [7] "Sec\_Edu\_Fem" "Sec\_Edu\_Mal"   
## [9] "Lab\_Forc\_Particip\_R\_Fem" "Lab\_Forc\_Particip\_R\_Mal"

So you can see that column names were shorten.

**Task 5** . Mutate the “Gender inequality” data and create two new variables

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

gii <- mutate(gii, edu\_R = (Sec\_Edu\_Fem/Sec\_Edu\_Mal))  
head(gii)

## GII\_Rank Country GII Mortality\_R Adolescent\_Birth\_R  
## 1 1 Norway 0.067 4 7.8  
## 2 2 Australia 0.110 6 12.1  
## 3 3 Switzerland 0.028 6 1.9  
## 4 4 Denmark 0.048 5 5.1  
## 5 5 Netherlands 0.062 6 6.2  
## 6 6 Germany 0.041 7 3.8  
## Representation\_Parliament Sec\_Edu\_Fem Sec\_Edu\_Mal Lab\_Forc\_Particip\_R\_Fem  
## 1 39.6 97.4 96.7 61.2  
## 2 30.5 94.3 94.6 58.8  
## 3 28.5 95.0 96.6 61.8  
## 4 38.0 95.5 96.6 58.7  
## 5 36.9 87.7 90.5 58.5  
## 6 36.9 96.3 97.0 53.6  
## Lab\_Forc\_Particip\_R\_Mal edu\_R  
## 1 68.7 1.0072389  
## 2 71.8 0.9968288  
## 3 74.9 0.9834369  
## 4 66.4 0.9886128  
## 5 70.6 0.9690608  
## 6 66.4 0.9927835

# Extra coding 1: alternative to mutate function

instead of using mutate, this is also possible:

gii$edu\_R <- gii$Sec\_Edu\_Fem/gii$Sec\_Edu\_Mal  
  
gii <- mutate(gii, Lab\_Forc\_R = (Lab\_Forc\_Particip\_R\_Fem/Lab\_Forc\_Particip\_R\_Mal))  
head(gii)

## GII\_Rank Country GII Mortality\_R Adolescent\_Birth\_R  
## 1 1 Norway 0.067 4 7.8  
## 2 2 Australia 0.110 6 12.1  
## 3 3 Switzerland 0.028 6 1.9  
## 4 4 Denmark 0.048 5 5.1  
## 5 5 Netherlands 0.062 6 6.2  
## 6 6 Germany 0.041 7 3.8  
## Representation\_Parliament Sec\_Edu\_Fem Sec\_Edu\_Mal Lab\_Forc\_Particip\_R\_Fem  
## 1 39.6 97.4 96.7 61.2  
## 2 30.5 94.3 94.6 58.8  
## 3 28.5 95.0 96.6 61.8  
## 4 38.0 95.5 96.6 58.7  
## 5 36.9 87.7 90.5 58.5  
## 6 36.9 96.3 97.0 53.6  
## Lab\_Forc\_Particip\_R\_Mal edu\_R Lab\_Forc\_R  
## 1 68.7 1.0072389 0.8908297  
## 2 71.8 0.9968288 0.8189415  
## 3 74.9 0.9834369 0.8251001  
## 4 66.4 0.9886128 0.8840361  
## 5 70.6 0.9690608 0.8286119  
## 6 66.4 0.9927835 0.8072289

**Task 6** .Join together the two datasets using the variable Country as the identifier  
**TIP** we use inner join to keep only observations in both data sets

hd\_gii\_joint <- inner\_join(hd, gii,   
 by = 'Country', suffix = c("\_hd", "\_gi"))  
  
head(hd\_gii\_joint)

## HDI\_Rank Country HDI Life\_Expec\_Birth Expec\_yr\_Edu Mean\_yr\_Edu  
## 1 1 Norway 0.944 81.6 17.5 12.6  
## 2 2 Australia 0.935 82.4 20.2 13.0  
## 3 3 Switzerland 0.930 83.0 15.8 12.8  
## 4 4 Denmark 0.923 80.2 18.7 12.7  
## 5 5 Netherlands 0.922 81.6 17.9 11.9  
## 6 6 Germany 0.916 80.9 16.5 13.1  
## GNI\_per\_Cap GNI\_per\_Cap\_Min\_HDI\_Rank GII\_Rank GII Mortality\_R  
## 1 64,992 5 1 0.067 4  
## 2 42,261 17 2 0.110 6  
## 3 56,431 6 3 0.028 6  
## 4 44,025 11 4 0.048 5  
## 5 45,435 9 5 0.062 6  
## 6 43,919 11 6 0.041 7  
## Adolescent\_Birth\_R Representation\_Parliament Sec\_Edu\_Fem Sec\_Edu\_Mal  
## 1 7.8 39.6 97.4 96.7  
## 2 12.1 30.5 94.3 94.6  
## 3 1.9 28.5 95.0 96.6  
## 4 5.1 38.0 95.5 96.6  
## 5 6.2 36.9 87.7 90.5  
## 6 3.8 36.9 96.3 97.0  
## Lab\_Forc\_Particip\_R\_Fem Lab\_Forc\_Particip\_R\_Mal edu\_R Lab\_Forc\_R  
## 1 61.2 68.7 1.0072389 0.8908297  
## 2 58.8 71.8 0.9968288 0.8189415  
## 3 61.8 74.9 0.9834369 0.8251001  
## 4 58.7 66.4 0.9886128 0.8840361  
## 5 58.5 70.6 0.9690608 0.8286119  
## 6 53.6 66.4 0.9927835 0.8072289

dim(hd\_gii\_joint)

## [1] 195 19

str(hd\_gii\_joint)

## 'data.frame': 195 obs. of 19 variables:  
## $ HDI\_Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ Country : chr "Norway" "Australia" "Switzerland" "Denmark" ...  
## $ HDI : num 0.944 0.935 0.93 0.923 0.922 0.916 0.916 0.915 0.913 0.913 ...  
## $ Life\_Expec\_Birth : num 81.6 82.4 83 80.2 81.6 80.9 80.9 79.1 82 81.8 ...  
## $ Expec\_yr\_Edu : num 17.5 20.2 15.8 18.7 17.9 16.5 18.6 16.5 15.9 19.2 ...  
## $ Mean\_yr\_Edu : num 12.6 13 12.8 12.7 11.9 13.1 12.2 12.9 13 12.5 ...  
## $ GNI\_per\_Cap : chr "64,992" "42,261" "56,431" "44,025" ...  
## $ GNI\_per\_Cap\_Min\_HDI\_Rank : int 5 17 6 11 9 11 16 3 11 23 ...  
## $ GII\_Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ GII : num 0.067 0.11 0.028 0.048 0.062 0.041 0.113 0.28 0.129 0.157 ...  
## $ Mortality\_R : int 4 6 6 5 6 7 9 28 11 8 ...  
## $ Adolescent\_Birth\_R : num 7.8 12.1 1.9 5.1 6.2 3.8 8.2 31 14.5 25.3 ...  
## $ Representation\_Parliament: num 39.6 30.5 28.5 38 36.9 36.9 19.9 19.4 28.2 31.4 ...  
## $ Sec\_Edu\_Fem : num 97.4 94.3 95 95.5 87.7 96.3 80.5 95.1 100 95 ...  
## $ Sec\_Edu\_Mal : num 96.7 94.6 96.6 96.6 90.5 97 78.6 94.8 100 95.3 ...  
## $ Lab\_Forc\_Particip\_R\_Fem : num 61.2 58.8 61.8 58.7 58.5 53.6 53.1 56.3 61.6 62 ...  
## $ Lab\_Forc\_Particip\_R\_Mal : num 68.7 71.8 74.9 66.4 70.6 66.4 68.1 68.9 71 73.8 ...  
## $ edu\_R : num 1.007 0.997 0.983 0.989 0.969 ...  
## $ Lab\_Forc\_R : num 0.891 0.819 0.825 0.884 0.829 ...

#setwd("./data")  
write.csv(hd\_gii\_joint, "./data/human.csv")

## *Data wrangling* (for this week)

Now in this week, lets continue to data wrangling and analysis in theme of dimensionality reduction

human <- read.csv("./data/human.csv")  
#altrnatively, you can read directly from web:  
human <- read.table("http://s3.amazonaws.com/assets.datacamp.com/production/course\_2218/datasets/human1.txt", sep =",", header = T)  
  
# check column names of the data  
names(human)

## [1] "HDI.Rank" "Country" "HDI" "Life.Exp"   
## [5] "Edu.Exp" "Edu.Mean" "GNI" "GNI.Minus.Rank"  
## [9] "GII.Rank" "GII" "Mat.Mor" "Ado.Birth"   
## [13] "Parli.F" "Edu2.F" "Edu2.M" "Labo.F"   
## [17] "Labo.M" "Edu2.FM" "Labo.FM"

#dimention of dataset  
dim(human)

## [1] 195 19

**Data description:**   
This data originates from the United Nations Development Programme. It gives us information about Human Development status in different countries by different variables.  
The data combines several indicators from most countries in the world. *More info in* [*http://hdr.undp.org/en/content/human-development-index-hdi*](http://hdr.undp.org/en/content/human-development-index-hdi)

The variables names and their descriptions are as following: “Country” = Country name

Health and knowledge  
“GNI” = Gross National Income per capita  
“Life.Exp” = Life expectancy at birth  
“Edu.Exp” = Expected years of schooling  
“Mat.Mor” = Maternal mortality ratio  
“Ado.Birth” = Adolescent birth rate

Empowerment  
“Parli.F” = Percentage of female representatives in parliament  
“Edu2.F” = Proportion of females with at least secondary education  
“Edu2.M” = Proportion of males with at least secondary education  
“Labo.F” = Proportion of females in the labour force  
“Labo.M” " Proportion of males in the labour force

“Edu2.FM” = Edu2.F / Edu2.M  
“Labo.FM” = Labo2.F / Labo2.M"

NOTE: Since I personally do not like dots (.) to be in the file or columns names, I like to change it to underscore (\_)

colnames(human) <- c("HDI\_Rank", "Country", "HDI", "Life\_Exp",   
 "Edu\_Exp", "Edu\_Mean", "GNI", "GNI\_Minus\_Rank",  
 "GII\_Rank","GII","Mat\_Mor","Ado\_Birth",  
 "Parli\_F","Edu2\_F","Edu2\_M","Labo\_F",  
 "Labo\_M","Edu2\_FM","Labo\_FM")  
#Source: https://raw.githubusercontent.com/TuomoNieminen/Helsinki-Open-Data-Science/master/datasets/human\_meta.txt   
  
# Check Column names again that changed successfully  
colnames(human)

## [1] "HDI\_Rank" "Country" "HDI" "Life\_Exp"   
## [5] "Edu\_Exp" "Edu\_Mean" "GNI" "GNI\_Minus\_Rank"  
## [9] "GII\_Rank" "GII" "Mat\_Mor" "Ado\_Birth"   
## [13] "Parli\_F" "Edu2\_F" "Edu2\_M" "Labo\_F"   
## [17] "Labo\_M" "Edu2\_FM" "Labo\_FM"

# look at the structure of human  
str(human)

## 'data.frame': 195 obs. of 19 variables:  
## $ HDI\_Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ Country : chr "Norway" "Australia" "Switzerland" "Denmark" ...  
## $ HDI : num 0.944 0.935 0.93 0.923 0.922 0.916 0.916 0.915 0.913 0.913 ...  
## $ Life\_Exp : num 81.6 82.4 83 80.2 81.6 80.9 80.9 79.1 82 81.8 ...  
## $ Edu\_Exp : num 17.5 20.2 15.8 18.7 17.9 16.5 18.6 16.5 15.9 19.2 ...  
## $ Edu\_Mean : num 12.6 13 12.8 12.7 11.9 13.1 12.2 12.9 13 12.5 ...  
## $ GNI : chr "64,992" "42,261" "56,431" "44,025" ...  
## $ GNI\_Minus\_Rank: int 5 17 6 11 9 11 16 3 11 23 ...  
## $ GII\_Rank : int 1 2 3 4 5 6 6 8 9 9 ...  
## $ GII : num 0.067 0.11 0.028 0.048 0.062 0.041 0.113 0.28 0.129 0.157 ...  
## $ Mat\_Mor : int 4 6 6 5 6 7 9 28 11 8 ...  
## $ Ado\_Birth : num 7.8 12.1 1.9 5.1 6.2 3.8 8.2 31 14.5 25.3 ...  
## $ Parli\_F : num 39.6 30.5 28.5 38 36.9 36.9 19.9 19.4 28.2 31.4 ...  
## $ Edu2\_F : num 97.4 94.3 95 95.5 87.7 96.3 80.5 95.1 100 95 ...  
## $ Edu2\_M : num 96.7 94.6 96.6 96.6 90.5 97 78.6 94.8 100 95.3 ...  
## $ Labo\_F : num 61.2 58.8 61.8 58.7 58.5 53.6 53.1 56.3 61.6 62 ...  
## $ Labo\_M : num 68.7 71.8 74.9 66.4 70.6 66.4 68.1 68.9 71 73.8 ...  
## $ Edu2\_FM : num 1.007 0.997 0.983 0.989 0.969 ...  
## $ Labo\_FM : num 0.891 0.819 0.825 0.884 0.829 ...

# print out summaries of the variables  
summary(human)

## HDI\_Rank Country HDI Life\_Exp   
## Min. : 1.00 Length:195 Min. :0.3480 Min. :49.00   
## 1st Qu.: 47.75 Class :character 1st Qu.:0.5770 1st Qu.:65.75   
## Median : 94.00 Mode :character Median :0.7210 Median :73.10   
## Mean : 94.31 Mean :0.6918 Mean :71.07   
## 3rd Qu.:141.25 3rd Qu.:0.8000 3rd Qu.:76.80   
## Max. :188.00 Max. :0.9440 Max. :84.00   
## NA's :7   
## Edu\_Exp Edu\_Mean GNI GNI\_Minus\_Rank   
## Min. : 4.10 Min. : 1.400 Length:195 Min. :-84.0000   
## 1st Qu.:11.10 1st Qu.: 5.550 Class :character 1st Qu.: -9.0000   
## Median :13.10 Median : 8.400 Mode :character Median : 1.5000   
## Mean :12.86 Mean : 8.079 Mean : 0.1862   
## 3rd Qu.:14.90 3rd Qu.:10.600 3rd Qu.: 11.0000   
## Max. :20.20 Max. :13.100 Max. : 47.0000   
## NA's :7   
## GII\_Rank GII Mat\_Mor Ado\_Birth   
## Min. : 1.00 Min. :0.0160 Min. : 1.0 Min. : 0.60   
## 1st Qu.: 47.75 1st Qu.:0.2030 1st Qu.: 16.0 1st Qu.: 15.45   
## Median : 94.00 Median :0.3935 Median : 69.0 Median : 40.95   
## Mean : 94.31 Mean :0.3695 Mean : 163.2 Mean : 49.55   
## 3rd Qu.:141.25 3rd Qu.:0.5272 3rd Qu.: 230.0 3rd Qu.: 71.78   
## Max. :188.00 Max. :0.7440 Max. :1100.0 Max. :204.80   
## NA's :7 NA's :33 NA's :10 NA's :5   
## Parli\_F Edu2\_F Edu2\_M Labo\_F   
## Min. : 0.00 Min. : 0.9 Min. : 3.20 Min. :13.50   
## 1st Qu.:12.47 1st Qu.: 27.8 1st Qu.: 38.30 1st Qu.:44.50   
## Median :19.50 Median : 55.7 Median : 60.00 Median :53.30   
## Mean :20.60 Mean : 54.8 Mean : 60.29 Mean :52.61   
## 3rd Qu.:27.02 3rd Qu.: 81.8 3rd Qu.: 85.80 3rd Qu.:62.62   
## Max. :57.50 Max. :100.0 Max. :100.00 Max. :88.10   
## NA's :3 NA's :26 NA's :26 NA's :11   
## Labo\_M Edu2\_FM Labo\_FM   
## Min. :44.20 Min. :0.1717 Min. :0.1857   
## 1st Qu.:68.88 1st Qu.:0.7284 1st Qu.:0.5987   
## Median :75.55 Median :0.9349 Median :0.7514   
## Mean :74.74 Mean :0.8541 Mean :0.7038   
## 3rd Qu.:80.15 3rd Qu.:0.9968 3rd Qu.:0.8513   
## Max. :95.50 Max. :1.4967 Max. :1.0380   
## NA's :11 NA's :26 NA's :11

The summary shows the minimum, maximum, mean, and 1st and 3rd quartiles of the data together with median. It also shows if there as NA cells in our columns. Moreover, we can see that all variables, except Country and GNI index, are numerical

**Task 1** : Mutate the data: transform the Gross National Income (GNI) variable to numeric"

# access the stringr package  
library(stringr)  
  
# look at the structure of the GNI column in 'human'  
str(human$GNI)

## chr [1:195] "64,992" "42,261" "56,431" "44,025" "45,435" "43,919" "39,568" ...

class(human$GNI)

## [1] "character"

As you can see, this GNI column is not numerical (is character class), so we shall change it as numeric.

# remove the commas from GNI and print out a numeric version of it  
human$GNI <- str\_replace(human$GNI, pattern=",", replace ="") %>% as.numeric  
  
str(human$GNI)

## num [1:195] 64992 42261 56431 44025 45435 ...

class(human$GNI)

## [1] "numeric"

**Task 2**: keep only the columns matching the following variable names"

# columns to keep  
keep <- c("Country", "Edu2\_FM", "Labo\_FM", "Life\_Exp", "Edu\_Exp",   
 "GNI", "Mat\_Mor", "Ado\_Birth", "Parli\_F")  
  
# select the 'keep' columns  
human <- dplyr::select(human, one\_of(keep))  
  
length(keep)

## [1] 9

ncol(human)

## [1] 9

So the number of columns of dataset is now equal to the number of variables we like to keep.

**Task 3** : Remove all rows with missing values

# print out a completeness indicator of the 'human' data  
complete.cases(human)

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
## [13] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [25] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE  
## [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [49] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE  
## [61] TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [73] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE  
## [85] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE  
## [97] TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [109] FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
## [121] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE  
## [133] FALSE TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE  
## [145] TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE TRUE  
## [157] FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE  
## [169] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE  
## [181] TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
## [193] TRUE TRUE TRUE

complete.cases function Return a logical vector indicating which cases are complete, i.e., have no missing values

# print out the data along with a completeness indicator as the last column  
data.frame(human[-1], comp = complete.cases(human))

## Edu2\_FM Labo\_FM Life\_Exp Edu\_Exp GNI Mat\_Mor Ado\_Birth Parli\_F comp  
## 1 1.0072389 0.8908297 81.6 17.5 64992 4 7.8 39.6 TRUE  
## 2 0.9968288 0.8189415 82.4 20.2 42261 6 12.1 30.5 TRUE  
## 3 0.9834369 0.8251001 83.0 15.8 56431 6 1.9 28.5 TRUE  
## 4 0.9886128 0.8840361 80.2 18.7 44025 5 5.1 38.0 TRUE  
## 5 0.9690608 0.8286119 81.6 17.9 45435 6 6.2 36.9 TRUE  
## 6 0.9927835 0.8072289 80.9 16.5 43919 7 3.8 36.9 TRUE  
## 7 1.0241730 0.7797357 80.9 18.6 39568 9 8.2 19.9 TRUE  
## 8 1.0031646 0.8171263 79.1 16.5 52947 28 31.0 19.4 TRUE  
## 9 1.0000000 0.8676056 82.0 15.9 42155 11 14.5 28.2 TRUE  
## 10 0.9968520 0.8401084 81.8 19.2 32689 8 25.3 31.4 TRUE  
## 11 0.9148148 0.7616580 83.0 15.4 76628 6 6.0 25.3 TRUE  
## 12 0.9116162 0.7566372 84.0 15.6 53959 NA 3.3 NA FALSE  
## 13 NA NA 80.0 15.0 79851 NA NA 20.0 FALSE  
## 14 0.9908362 0.8880707 82.2 15.8 45636 4 6.5 43.6 TRUE  
## 15 0.9989990 0.8107715 80.7 16.2 39267 8 25.8 23.5 TRUE  
## 16 0.9934498 0.9108527 82.6 19.0 35182 4 11.5 41.3 TRUE  
## 17 0.8641975 0.6948682 81.9 16.9 33890 27 2.2 16.3 TRUE  
## 18 0.9667812 0.8379161 82.4 16.0 30676 2 7.8 22.5 TRUE  
## 19 1.0000000 0.7848297 81.7 13.9 58711 11 8.3 28.3 TRUE  
## 20 1.0139860 0.6931818 83.5 15.3 36927 6 5.4 11.6 TRUE  
## 21 0.9348613 0.8010118 80.8 16.3 41187 6 6.7 42.4 TRUE  
## 22 0.9375000 0.8230519 82.2 16.0 38056 12 5.7 25.7 TRUE  
## 23 1.0000000 0.8064993 81.4 15.7 43869 4 4.1 30.3 TRUE  
## 24 1.0000000 0.8703125 80.8 17.1 38695 4 9.2 42.5 TRUE  
## 25 0.9775510 0.8275316 80.4 16.8 27852 7 0.6 27.7 TRUE  
## 26 0.9138167 0.7978723 82.6 17.3 32045 4 10.6 38.0 TRUE  
## 27 0.8844720 0.6655462 83.1 16.0 33030 4 4.0 30.1 TRUE  
## 28 1.0020060 0.7481698 78.6 16.4 26660 5 4.9 18.9 TRUE  
## 29 0.8880597 0.7072000 80.9 17.6 24524 5 11.9 21.0 TRUE  
## 30 1.0000000 0.8156749 76.8 16.5 25214 11 16.8 19.8 TRUE  
## 31 0.9424779 0.6985392 78.8 14.5 72570 27 23.0 NA FALSE  
## 32 0.9302326 0.7876231 80.2 14.0 28633 10 5.5 12.5 TRUE  
## 33 1.1305085 0.5319372 78.2 13.8 123124 6 9.5 0.0 TRUE  
## 34 1.0040568 NA 81.3 13.5 43978 NA NA 50.0 FALSE  
## 35 0.9959799 0.7448980 76.3 15.1 25845 7 15.9 18.7 TRUE  
## 36 0.9286550 0.7534669 77.4 15.5 23177 3 12.2 22.1 TRUE  
## 37 0.9448568 0.8291233 73.3 16.4 24500 11 10.6 23.4 TRUE  
## 38 0.8772379 0.5716440 80.6 14.4 27930 9 18.2 13.0 TRUE  
## 39 0.8605974 0.2579821 74.3 16.3 52821 16 10.2 19.9 TRUE  
## 40 0.9774306 0.6333333 76.3 17.9 22050 69 54.4 36.8 TRUE  
## 41 1.1944444 0.5054348 77.0 13.3 60868 8 27.6 17.5 TRUE  
## 42 0.9594241 0.6577540 81.7 15.2 21290 22 55.3 15.8 TRUE  
## 43 0.9896266 0.8293051 80.9 16.3 25757 8 12.6 31.3 TRUE  
## 44 0.9918946 0.7466667 75.2 15.4 22916 14 12.1 10.1 TRUE  
## 45 1.1031128 0.4510932 76.6 14.4 38599 22 13.8 15.0 TRUE  
## 46 0.9989899 0.8121302 74.2 15.2 22281 13 13.5 18.0 TRUE  
## 47 0.9081197 0.7654110 77.3 14.8 19409 13 12.7 25.8 TRUE  
## 48 0.9875666 0.5246691 74.4 14.7 83961 14 14.5 1.5 TRUE  
## 49 0.8891235 0.7504363 76.2 15.2 14558 7 15.2 17.3 TRUE  
## 50 0.9436009 0.7939778 71.3 15.7 16676 1 20.6 30.1 TRUE  
## 51 0.9686486 0.7963738 70.1 14.7 22352 24 25.7 14.5 TRUE  
## 52 0.8266200 0.3510896 76.8 13.6 34858 11 10.6 9.6 TRUE  
## 53 0.9358696 0.7503852 74.7 14.2 18108 33 31.0 12.0 TRUE  
## 54 1.0815109 0.7239583 77.2 15.5 19283 14 58.3 11.5 TRUE  
## 55 1.0410959 0.8738966 75.4 12.6 21336 37 28.5 16.7 TRUE  
## 56 0.9645749 0.8690629 69.4 15.0 20867 26 29.9 20.1 TRUE  
## 57 1.0205245 0.8603133 75.6 15.4 12488 52 48.4 19.6 TRUE  
## 58 NA NA 76.1 14.0 20070 NA 49.3 25.7 FALSE  
## 59 0.9717868 0.8118644 74.2 14.4 15596 5 35.9 20.4 TRUE  
## 60 NA NA 72.7 13.7 13496 NA NA 10.3 FALSE  
## 61 1.0821643 0.5990220 77.6 13.3 18192 85 78.5 19.3 TRUE  
## 62 0.9130435 0.5880795 74.7 12.7 22762 29 5.7 14.2 TRUE  
## 63 0.8517241 0.5876011 74.4 15.6 17470 73 30.9 11.6 TRUE  
## 64 1.0045045 NA 73.1 13.4 23300 NA 56.3 43.8 FALSE  
## 65 0.9802956 0.7019868 70.4 12.3 26090 84 34.8 24.7 TRUE  
## 66 0.7934783 0.7307061 74.9 14.4 12190 16 16.9 34.0 TRUE  
## 67 0.9428934 0.6200000 79.4 13.8 7301 80 43.1 48.9 TRUE  
## 68 0.9566787 0.3286319 79.3 13.8 16509 16 12.0 3.1 TRUE  
## 69 1.0039604 0.5898734 79.4 13.9 13413 38 60.8 33.3 TRUE  
## 70 0.9201183 0.2255435 75.4 15.1 15440 23 31.6 3.1 TRUE  
## 71 1.1141732 0.6452020 74.2 14.2 16159 110 83.2 17.0 TRUE  
## 72 0.6500000 0.4152542 75.3 14.5 18677 20 30.9 14.4 TRUE  
## 73 0.9515707 0.4600262 74.9 13.7 9779 29 16.9 5.8 TRUE  
## 74 0.9191419 0.5644556 76.8 13.1 16056 49 63.4 37.1 TRUE  
## 75 1.0419847 0.7351485 74.5 15.2 15175 69 70.8 9.6 TRUE  
## 76 0.9676375 0.7523302 74.9 13.8 7164 41 46.8 11.3 TRUE  
## 77 NA NA 73.8 12.9 20805 NA NA 6.7 FALSE  
## 78 0.9620123 0.9037356 70.8 11.9 16428 26 40.0 15.6 TRUE  
## 79 NA NA 73.4 15.8 10939 23 35.4 25.0 FALSE  
## 80 0.8853503 0.2342342 74.0 13.5 11365 50 26.5 11.6 TRUE  
## 81 0.7230216 0.6385185 75.4 13.4 11780 7 18.3 33.3 TRUE  
## 82 0.9562044 0.7952167 71.0 15.1 8178 23 25.7 11.8 TRUE  
## 83 0.8612903 0.2105263 74.8 14.0 13054 89 10.0 25.7 TRUE  
## 84 0.8517398 0.8080569 74.6 13.1 11015 89 50.7 22.3 TRUE  
## 85 0.9306030 0.6854962 77.8 11.8 9943 21 15.3 20.7 TRUE  
## 86 0.9894737 0.7465565 74.7 12.3 8124 29 27.1 10.7 TRUE  
## 87 0.6432665 0.5951134 76.5 13.6 9638 8 15.1 19.3 TRUE  
## 88 1.0177665 0.6614268 75.9 14.2 10605 87 77.0 41.6 TRUE  
## 89 NA 0.8228346 75.1 12.6 9765 34 56.3 20.7 FALSE  
## 90 0.8164117 0.8160920 75.8 13.1 12547 32 8.6 23.6 TRUE  
## 91 0.9953488 0.5208333 70.0 15.7 7493 59 42.8 14.0 TRUE  
## 92 1.0142687 0.8167388 69.4 14.6 10729 68 18.7 14.9 TRUE  
## 93 0.8750000 0.7967782 74.4 13.5 13323 26 41.0 6.1 TRUE  
## 94 1.2801724 NA 77.8 12.7 9994 NA NA 21.9 FALSE  
## 95 1.3245823 0.3926702 71.6 14.0 14911 15 2.5 16.0 TRUE  
## 96 0.7114967 0.3540197 74.8 14.6 10404 46 4.6 31.3 TRUE  
## 97 1.0233813 0.7001255 74.0 13.5 12040 83 68.5 20.9 TRUE  
## 98 NA 0.7141026 72.9 13.4 9937 45 54.5 13.0 FALSE  
## 99 1.0541311 0.7912553 75.7 12.4 7415 80 70.1 16.7 TRUE  
## 100 0.9909400 0.7171582 72.8 14.7 5069 120 18.1 0.0 TRUE  
## 101 1.0079156 0.5978129 70.0 13.6 7614 45 71.4 13.3 TRUE  
## 102 1.0470810 0.6526718 73.5 13.1 11883 100 99.6 19.1 TRUE  
## 103 0.9469214 0.5886628 71.1 12.7 15617 130 35.2 11.8 TRUE  
## 104 0.8348624 0.7251613 76.8 13.0 12328 31 4.2 5.9 TRUE  
## 105 1.0716667 0.4023973 73.4 12.9 5327 58 28.3 6.1 TRUE  
## 106 0.9448010 0.8811275 64.5 12.5 16646 170 44.2 9.5 TRUE  
## 107 0.9689441 0.8506787 71.6 11.9 5223 21 29.3 20.8 TRUE  
## 108 0.7244224 0.3168449 71.1 13.5 10512 45 43.0 2.2 TRUE  
## 109 NA 0.6098830 65.6 10.8 13066 61 18.0 25.8 FALSE  
## 110 1.4930748 0.8593272 64.4 12.5 16367 240 103.0 16.2 TRUE  
## 111 0.8109756 0.6104513 68.9 13.0 9788 190 48.3 17.1 TRUE  
## 112 0.8558140 0.6568396 72.9 11.9 7643 110 67.0 16.8 TRUE  
## 113 0.9074074 0.2319277 72.9 13.0 4699 NA 45.8 NA FALSE  
## 114 NA 0.6362434 68.4 11.5 5567 36 38.8 16.4 FALSE  
## 115 1.0345369 0.6411543 68.2 11.3 7915 120 46.8 27.1 TRUE  
## 116 0.8440367 0.6050633 73.0 12.3 7349 69 76.0 27.4 TRUE  
## 117 0.9578393 0.7355372 57.4 13.6 12122 140 50.9 40.7 TRUE  
## 118 0.8342697 0.8880779 75.8 11.9 5092 49 29.0 24.3 TRUE  
## 119 0.8054146 0.7935723 68.3 13.2 5760 200 71.9 51.8 TRUE  
## 120 0.9762397 0.7044025 70.6 12.5 3044 75 29.3 23.3 TRUE  
## 121 0.5537849 0.2134670 69.4 10.1 14003 67 68.7 26.5 TRUE  
## 122 NA 0.6152927 73.3 13.5 6094 53 70.6 20.8 FALSE  
## 123 NA NA 69.1 11.7 3432 96 18.6 0.0 FALSE  
## 124 1.2615063 0.5291925 66.4 10.3 6522 250 88.5 31.3 TRUE  
## 125 1.0287206 0.5902864 74.9 11.5 4457 100 100.8 39.1 TRUE  
## 126 0.6854305 0.3496042 74.0 11.6 6850 120 35.8 11.0 TRUE  
## 127 0.9680233 0.8587127 64.8 11.3 9418 130 54.9 37.7 TRUE  
## 128 0.9439655 0.5589569 71.8 10.7 6929 140 97.2 13.3 TRUE  
## 129 1.0427632 0.7639429 69.4 11.2 2517 44 42.8 15.2 TRUE  
## 130 0.4770318 0.3379224 68.0 11.7 5497 190 32.8 12.2 TRUE  
## 131 1.0852713 0.5162847 73.1 11.1 3938 120 84.0 25.8 TRUE  
## 132 0.9855072 0.8639896 69.5 12.6 7176 120 40.9 8.3 TRUE  
## 133 NA 0.4842520 68.2 11.7 5363 270 52.2 38.5 FALSE  
## 134 0.7283951 0.1856946 69.6 12.3 2728 49 41.6 12.4 TRUE  
## 135 NA 0.7687500 71.9 10.6 2803 86 44.8 0.0 FALSE  
## 136 0.8446809 0.9383562 62.3 11.1 6012 410 126.7 11.5 TRUE  
## 137 NA NA 66.0 12.3 2434 130 16.6 8.7 FALSE  
## 138 NA 0.8752711 57.6 9.0 21056 290 112.6 19.7 FALSE  
## 139 0.5863636 0.8539720 60.1 13.5 3734 280 125.4 12.7 TRUE  
## 140 0.6986090 0.9425770 61.4 11.5 3852 380 58.4 10.9 TRUE  
## 141 0.6189189 0.9646018 66.2 10.6 4680 NA 65.0 25.0 FALSE  
## 142 0.8256659 0.6825208 71.6 10.0 3191 170 80.6 20.0 TRUE  
## 143 0.4323144 0.9109827 68.4 10.9 2949 170 44.3 19.0 TRUE  
## 144 NA 0.5822622 66.5 11.3 2918 210 65.1 18.2 FALSE  
## 145 0.8057325 0.8591160 61.6 11.0 2762 400 93.6 20.8 TRUE  
## 146 0.4633508 0.9173364 69.6 12.4 2311 190 73.7 29.5 TRUE  
## 147 0.4186551 0.2967431 66.2 7.8 4866 170 27.3 19.7 TRUE  
## 148 1.4967320 0.9137303 65.9 8.6 4608 200 12.1 4.7 TRUE  
## 149 NA 0.8231469 52.3 11.4 6822 460 170.2 36.8 FALSE  
## 150 0.8423077 0.6131285 49.0 11.3 5542 310 72.0 14.7 TRUE  
## 151 0.5894737 0.9767184 65.0 9.2 2411 410 122.7 36.0 TRUE  
## 152 NA 0.7566719 52.8 9.0 5341 560 119.6 6.6 FALSE  
## 153 0.6103152 0.8307292 55.5 10.4 2803 590 115.8 27.1 TRUE  
## 154 NA 0.9569061 65.1 10.3 1328 440 122.8 20.5 FALSE  
## 155 0.7854839 0.9275362 57.5 10.9 1615 470 60.3 35.1 TRUE  
## 156 0.3971292 0.3628319 63.1 8.5 3560 320 73.3 22.2 TRUE  
## 157 NA 0.6759494 67.9 9.2 1540 130 64.9 2.0 FALSE  
## 158 0.5241379 0.9527027 62.6 9.9 2463 220 62.1 2.7 TRUE  
## 159 NA 0.4394507 63.3 11.5 1456 350 51.1 3.0 FALSE  
## 160 0.3220974 0.3518006 63.8 9.2 3519 270 47.0 0.7 TRUE  
## 161 1.1526316 0.8027211 49.8 11.1 3306 490 89.4 26.8 TRUE  
## 162 0.3995037 0.9913899 59.7 12.2 1228 450 91.5 17.6 TRUE  
## 163 0.6363636 0.8577465 62.8 8.7 1669 380 42.0 3.5 TRUE  
## 164 0.9090909 1.0128957 64.2 10.3 1458 320 33.6 57.5 TRUE  
## 165 0.6835821 0.9570707 58.5 9.8 1613 360 126.6 35.0 TRUE  
## 166 0.4185185 0.8633461 59.6 11.1 1767 340 90.2 8.4 TRUE  
## 167 0.6648352 0.4118421 63.5 7.0 3809 360 84.0 23.8 TRUE  
## 168 NA 0.5361891 62.0 6.4 3276 230 18.6 12.7 FALSE  
## 169 NA NA 55.7 7.6 2332 730 75.3 24.3 FALSE  
## 170 0.4675325 0.7500000 66.5 7.9 2188 320 94.4 42.7 TRUE  
## 171 0.1979866 0.1987421 60.4 9.3 1885 400 86.8 27.6 TRUE  
## 172 0.4651163 0.6437346 51.5 8.9 3171 720 130.3 9.2 TRUE  
## 173 0.5138889 1.0380368 62.8 10.8 747 510 144.8 16.7 TRUE  
## 174 0.4285714 0.8756999 64.1 8.5 1428 420 78.4 25.5 TRUE  
## 175 0.5523810 0.8709288 60.2 8.8 1507 430 115.8 9.4 TRUE  
## 176 0.3950617 0.9658470 58.7 9.8 680 730 135.3 8.2 TRUE  
## 177 0.3918575 0.8981481 60.9 9.5 805 640 117.4 10.7 TRUE  
## 178 NA 0.8687898 55.2 9.0 1362 560 99.3 13.7 FALSE  
## 179 0.5099338 0.6240786 58.0 8.4 1583 550 175.6 9.5 TRUE  
## 180 0.2258065 1.0326087 55.1 9.3 1123 480 137.8 39.6 TRUE  
## 181 0.4608295 0.9521739 50.9 8.6 1780 1100 100.7 12.4 TRUE  
## 182 NA 0.8378033 58.8 8.7 1096 650 131.0 21.9 FALSE  
## 183 0.2812500 0.8566667 58.7 7.8 1591 400 115.4 13.3 TRUE  
## 184 0.6385542 1.0158537 56.7 10.1 758 740 30.3 34.9 TRUE  
## 185 0.1717172 0.8080808 51.6 7.4 2085 980 152.0 14.9 TRUE  
## 186 NA 0.8908686 63.7 4.1 1130 380 65.3 22.0 FALSE  
## 187 0.3782772 0.8531140 50.7 7.2 581 880 98.3 12.5 TRUE  
## 188 0.3076923 0.4459309 61.4 5.4 908 630 204.8 13.3 TRUE  
## 189 0.7289916 0.3081009 70.6 12.0 15722 155 45.4 14.0 TRUE  
## 190 0.8250377 0.7884131 74.0 12.7 11449 72 21.2 18.7 TRUE  
## 191 0.8784119 0.6514286 72.3 13.6 12791 28 30.8 19.0 TRUE  
## 192 0.9836957 0.6729323 75.0 14.0 14242 85 68.3 27.0 TRUE  
## 193 0.5329670 0.3711083 68.4 11.2 5605 183 38.7 17.5 TRUE  
## 194 0.7015873 0.8537859 58.5 9.6 3363 506 109.7 22.5 TRUE  
## 195 0.8333333 0.6558018 71.5 12.2 14301 210 47.4 21.8 TRUE

# filter out all rows with NA values  
human\_filterd <- filter(human, complete.cases(human))  
dim(human\_filterd)

## [1] 162 9

dim(human)

## [1] 195 9

So you can see that the original human dataset had 195 rows, an now has 162.

# Extra coding 2: alternatively, you could do with the following code too

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## <U+221A> ggplot2 3.3.2 <U+221A> readr 1.4.0  
## <U+221A> tibble 3.0.4 <U+221A> purrr 0.3.4  
## <U+221A> tidyr 1.1.2 <U+221A> forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

human\_filterd\_new = human %>% drop\_na()  
dim(human\_filterd\_new)

## [1] 162 9

#Or alternatively same   
human\_ = human %>% drop\_na(colnames(human))  
dim(human\_)

## [1] 162 9

**Task 4**: Remove the observations which relate to regions instead of countries

#look at the last 10 observations  
tail(human\_filterd, 10)

## Country Edu2\_FM Labo\_FM Life\_Exp Edu\_Exp GNI  
## 153 Chad 0.1717172 0.8080808 51.6 7.4 2085  
## 154 Central African Republic 0.3782772 0.8531140 50.7 7.2 581  
## 155 Niger 0.3076923 0.4459309 61.4 5.4 908  
## 156 Arab States 0.7289916 0.3081009 70.6 12.0 15722  
## 157 East Asia and the Pacific 0.8250377 0.7884131 74.0 12.7 11449  
## 158 Europe and Central Asia 0.8784119 0.6514286 72.3 13.6 12791  
## 159 Latin America and the Caribbean 0.9836957 0.6729323 75.0 14.0 14242  
## 160 South Asia 0.5329670 0.3711083 68.4 11.2 5605  
## 161 Sub-Saharan Africa 0.7015873 0.8537859 58.5 9.6 3363  
## 162 World 0.8333333 0.6558018 71.5 12.2 14301  
## Mat\_Mor Ado\_Birth Parli\_F  
## 153 980 152.0 14.9  
## 154 880 98.3 12.5  
## 155 630 204.8 13.3  
## 156 155 45.4 14.0  
## 157 72 21.2 18.7  
## 158 28 30.8 19.0  
## 159 85 68.3 27.0  
## 160 183 38.7 17.5  
## 161 506 109.7 22.5  
## 162 210 47.4 21.8

# last indice we want to keep  
last <- nrow(human\_filterd) - 7  
  
# choose everything until the last 7 observations  
human\_ <- human\_filterd[1:last, ]

**Task 5**. Define the row names of the data by the country names

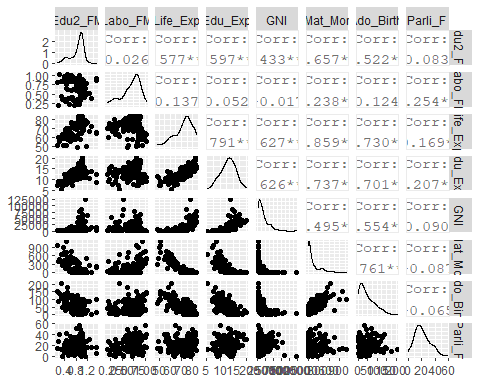
# add countries as rownames  
rownames(human\_filterd) <- human\_filterd$Country  
  
# remove the Country variable  
human\_ <- select(human\_filterd, -Country)  
  
# load library  
library(GGally)

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

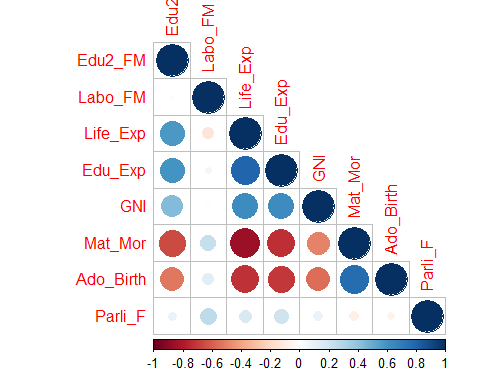
library(corrplot)

## corrplot 0.84 loaded

# visualize the 'human\_' variables  
ggpairs(human\_)

 As the graph shows, there is strong correlation between some variables such as ado\_birth abd mat\_mor and edu\_exp. similarly, GNI and life\_exp and edu\_exp.

# compute the correlation matrix and visualize it with corrplot  
cor(human\_) %>% corrplot(type = "lower")

 As graph shows, there is strong negative correlation between Mat\_Mor and Life\_Exp, which means by increase of mmat\_mor, life\_exp increases. Same for Life\_Exp and Abo\_birth.  
Also, some positive correlation between abo\_birth and Mat\_Mor. Parli\_F and labo\_FM are not correlating with other variables.  
**TIP:** by including ´(type = “lower”)` you set it to plot lower triangular part of correlation matirx.

write.csv(human\_, "human\_new.csv")

## *Part 2) Data analysis*

human <- read.csv("human\_new.csv")  
# Alternatively, yu can read directly from web  
human <- read.csv("http://s3.amazonaws.com/assets.datacamp.com/production/course\_2218/datasets/human2.txt", sep = ",", stringsAsFactors = F)  
dim(human)

## [1] 155 8

**Task 1** : Show a graphical overview

# standardize the variables  
human\_std <- scale(human)

As center = True (by default), now, this scale is done now by cantering which is done by subtracting the column means (omitting NAs) of x from their corresponding columns.  
We standardize the data, because PCA is sensitive to relative scaling od the orignal features.

# print out summaries of the standardized variables  
summary(human\_std)

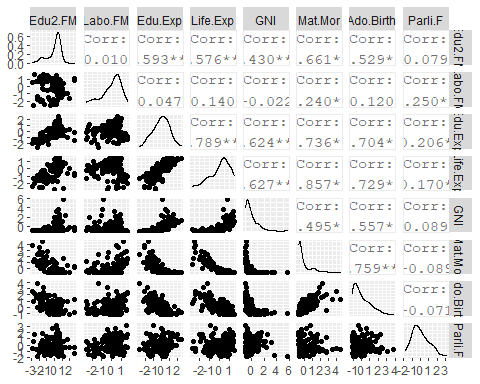
## Edu2.FM Labo.FM Edu.Exp Life.Exp   
## Min. :-2.8189 Min. :-2.6247 Min. :-2.7378 Min. :-2.7188   
## 1st Qu.:-0.5233 1st Qu.:-0.5484 1st Qu.:-0.6782 1st Qu.:-0.6425   
## Median : 0.3503 Median : 0.2316 Median : 0.1140 Median : 0.3056   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.5958 3rd Qu.: 0.7350 3rd Qu.: 0.7126 3rd Qu.: 0.6717   
## Max. : 2.6646 Max. : 1.6632 Max. : 2.4730 Max. : 1.4218   
## GNI Mat.Mor Ado.Birth Parli.F   
## Min. :-0.9193 Min. :-0.6992 Min. :-1.1325 Min. :-1.8203   
## 1st Qu.:-0.7243 1st Qu.:-0.6496 1st Qu.:-0.8394 1st Qu.:-0.7409   
## Median :-0.3013 Median :-0.4726 Median :-0.3298 Median :-0.1403   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.3712 3rd Qu.: 0.1932 3rd Qu.: 0.6030 3rd Qu.: 0.6127   
## Max. : 5.6890 Max. : 4.4899 Max. : 3.8344 Max. : 3.1850

summary(human)

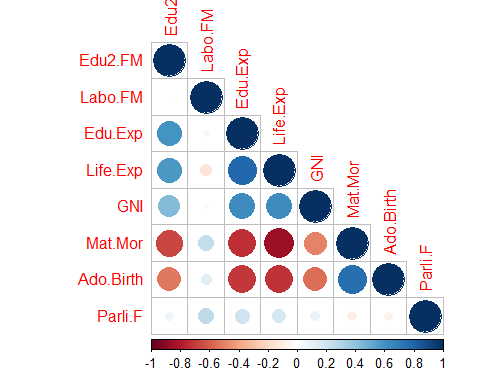
## Edu2.FM Labo.FM Edu.Exp Life.Exp   
## Min. :0.1717 Min. :0.1857 Min. : 5.40 Min. :49.00   
## 1st Qu.:0.7264 1st Qu.:0.5984 1st Qu.:11.25 1st Qu.:66.30   
## Median :0.9375 Median :0.7535 Median :13.50 Median :74.20   
## Mean :0.8529 Mean :0.7074 Mean :13.18 Mean :71.65   
## 3rd Qu.:0.9968 3rd Qu.:0.8535 3rd Qu.:15.20 3rd Qu.:77.25   
## Max. :1.4967 Max. :1.0380 Max. :20.20 Max. :83.50   
## 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

As you can see from the summary of scaled and not-scaled data, the scaled one is standardized based on mean of each column, so data are not heteroscedasticity effect. So, we are not ready for further analysis.

ggpairs(as.data.frame(human\_std))

 This visualises the data distribution for each column and comparing with other columns.

cor(as.data.frame(human\_std)) %>% corrplot(type = "lower")

 The correction matrix (same as above) is given here, same interpretation as in data wrangling part

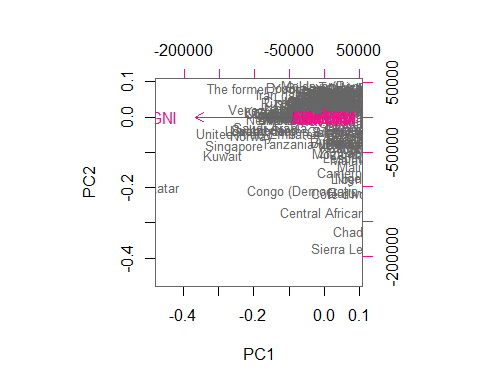
**Task 2** Perform principal component analysis (PCA) with the SVD method  
As instructed, we do PCA on not standardized human data.

pca\_human <- prcomp(human)  
summary(pca\_human)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8  
## Standard deviation 1.854e+04 185.5219 25.19 11.45 3.766 1.566 0.1912 0.1591  
## Proportion of Variance 9.999e-01 0.0001 0.00 0.00 0.000 0.000 0.0000 0.0000  
## Cumulative Proportion 9.999e-01 1.0000 1.00 1.00 1.000 1.000 1.0000 1.0000

# Draw a biplot displaying the observations by the first two principal components (PC1 coordinate in x-axis, PC2 coordinate in y-axis)  
biplot(pca\_human, choices = 1:2, cex = c(0.8, 1), col = c("grey40", "deeppink2"))

## 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

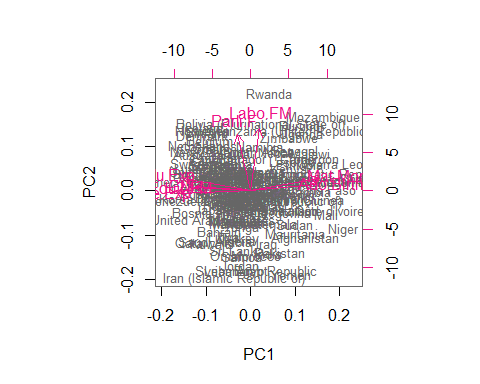
 The arrows show the original variables (there are overlapping on each other).

**Task 3** Perfom PCa with standardize variables

pca\_human <- prcomp(human\_std)  
summary(pca\_human)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 2.0708 1.1397 0.87505 0.77886 0.66196 0.53631 0.45900  
## Proportion of Variance 0.5361 0.1624 0.09571 0.07583 0.05477 0.03595 0.02634  
## Cumulative Proportion 0.5361 0.6984 0.79413 0.86996 0.92473 0.96069 0.98702  
## PC8  
## Standard deviation 0.32224  
## Proportion of Variance 0.01298  
## Cumulative Proportion 1.00000

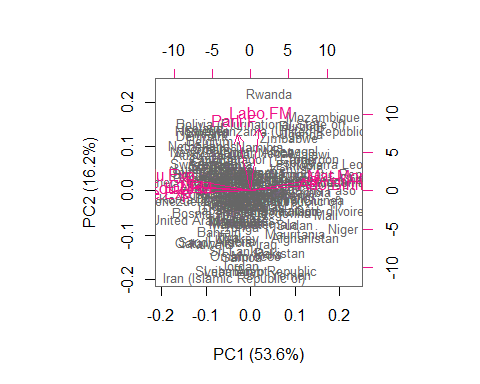
# Draw a biplot displaying the observations by the first two principal components (PC1 coordinate in x-axis, PC2 coordinate in y-axis)  
biplot(pca\_human, choices = 1:2, cex = c(0.8, 1), col = c("grey40", "deeppink2"))

 The arrows show the original variables

# rounded percentages of variance captured by each PC  
pca\_pr <- round(100\*summary(pca\_human)$importance[2,], digits = 1)   
pca\_pr

## PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8   
## 53.6 16.2 9.6 7.6 5.5 3.6 2.6 1.3

# create object pc\_lab to be used as axis labels  
pc\_lab <- paste0(names(pca\_pr), " (", pca\_pr, "%)")  
  
# draw a biplot  
biplot(pca\_human, cex = c(0.8, 1), col = c("grey40", "deeppink2"), xlab = pc\_lab[1], ylab = pc\_lab[2])



**Interpretation** In the biplot small angle between arrows means high correlation between two variables. Thus, as Parli\_F and Edu.Exp abd life\_exp has high angle, thus don’t have high correlation (as also shown earlier in above correlation graphs).  
On the other hand, Edu.Exp and GNI and Edu2.FM has very narrow angle, thus very high correlation (again, explain in above correlation plots).  
Additionally, we can see that Principal component 2 (PC2), in Y axis, has same direction to Parli\_F and Labo\_FM variables, thus, PC2 is contributing to that features (dimensions), while all other variables has contributing to PC1. *More info in* [*https://vimeo.com/204165420*](https://vimeo.com/204165420)

As for PCA components printed by summary function, the results differ, which is as expected. Because we standardised the data to have not heteroscedasticity. Moreover, since the data were standardized, the Standard deviation and Proportion of Variance will be different with non-standardised dataset.  
We shall remember the PCA is sensitive to scaling of variables.  
To interpret PCA, we could look at Proportion of Variance and its cumulative proportion. For example, in the standardized data, the PCA shows that PCA1 contributes to 53.6% of the variance in the data, and PCA2 contributes to 16.3%. Comparing this result with non-standardised result, the data and results has changed which is as expected because there were some heterostasity in the data which CA is sensitive for. PCA1 already could contribute over 99% of variance which looks doubtful to me. Thus, standardising the data is very effective and helpful to use before PCA analysis and getting the dimension of data reduced!  
It makes sense, because in actual phenomenon’s, Gross National Income per capita (GNI) is higher if Expected years of schooling (Edu\_Exp) is higher, and therefore, Life expectancy (Life\_Exp) will be higher.

**Task 4:** I personally think that using PCA is very useful because cool and very useful also in my own research in forest data science where I often have tens of variables. PCA 1 and PCA 2 together show/contribute to nearly 70% (53.6+16.2) of the variance in the original data.

# Extra coding 3: Plot PCA in 3D

*inspired from* [*https://cran.r-project.org/web/packages/pca3d/vignettes/pca3d.pdf*](https://cran.r-project.org/web/packages/pca3d/vignettes/pca3d.pdf)

library(pca3d)  
gr<-factor(rownames(human\_std))  
#summary(gr)  
pca3d(pca\_human, group=gr)

## [1] 0.12143416 0.06251040 0.05134851  
## Creating new device

snapshotPCA3d(file="first\_plot.png")

**feel free to zoom on it or rotate it**.  
Every item in 3D is an observation (country) in 3D space created by PC1, PC2 and PC3.

## Multiple Correspondence Analysis

**Task 5:**  
Let’s do Multiple Correspondence Analysis (MCA) using the Factominer package, which contains functions dedicated to multivariate explanatory data analysis. It contains for example methods (Multiple) Correspondence analysis, Multiple Factor analysis as well as PCA.  
We will do only Multiple Correspondence Analysis (MCA) with this data.  
In this part I am going to use the tea dataset. The dataset contains the answers of a questionnaire on tea consumption.

library(FactoMineR)  
library(ggplot2)  
data(tea)  
View(tea)  
  
# column names to keep in the dataset  
keep\_columns <- c("Tea", "How", "how", "sugar", "where", "lunch")  
  
# select the 'keep\_columns' to create a new dataset  
tea\_time <- dplyr::select(tea, one\_of(keep\_columns))  
  
# look at the summaries and structure of the data  
summary(tea\_time)

## Tea How how sugar   
## black : 74 alone:195 tea bag :170 No.sugar:155   
## Earl Grey:193 lemon: 33 tea bag+unpackaged: 94 sugar :145   
## green : 33 milk : 63 unpackaged : 36   
## other: 9   
## where lunch   
## chain store :192 lunch : 44   
## chain store+tea shop: 78 Not.lunch:256   
## tea shop : 30   
##

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 ...

As you can see, the data include information on tea consumption, e.g. 74 of black tea, 193 Earl Grey and 33 green tea, and how people drink the tea etc. We are doing to plot them and see visally ( as following).

# column names to keep in the dataset  
keep\_columns <- c("Tea", "How", "how", "sugar", "where", "lunch")  
  
# select the 'keep\_columns' to create a new dataset  
tea\_time <- select(tea, one\_of(keep\_columns))  
  
# look at the summaries and structure of the data  
summary(tea\_time)

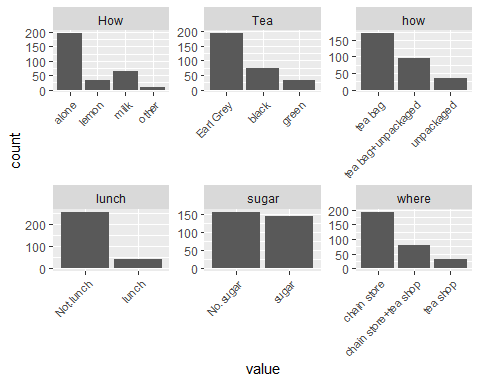
## Tea How how sugar   
## black : 74 alone:195 tea bag :170 No.sugar:155   
## Earl Grey:193 lemon: 33 tea bag+unpackaged: 94 sugar :145   
## green : 33 milk : 63 unpackaged : 36   
## other: 9   
## where lunch   
## chain store :192 lunch : 44   
## chain store+tea shop: 78 Not.lunch:256   
## tea shop : 30   
##

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 ...

# visualize the dataset  
gather(tea\_time) %>% ggplot(aes(value)) + facet\_wrap("key", scales = "free") + geom\_bar() + theme(axis.text.x = element\_text(angle = 45, hjust = 1, size = 8))

## Warning: attributes are not identical across measure variables;  
## they will be dropped

 So you can see the data that include information on tea consumption, multiple columns about how it was used, what type of tea where and when they drink it.

We use some of the columns for MCA analysis

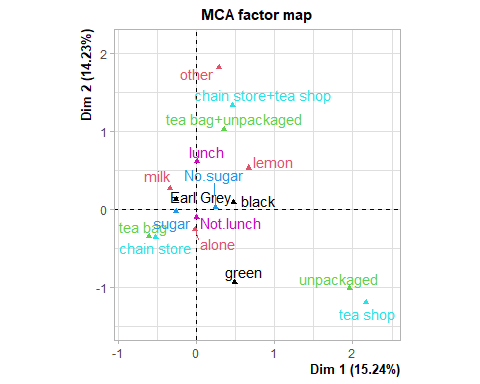
mca <- MCA(tea\_time, graph = FALSE)  
  
#summary of the model  
summary(mca)

##   
## Call:  
## MCA(X = tea\_time, graph = FALSE)   
##   
##   
## Eigenvalues  
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6 Dim.7  
## Variance 0.279 0.261 0.219 0.189 0.177 0.156 0.144  
## % of var. 15.238 14.232 11.964 10.333 9.667 8.519 7.841  
## Cumulative % of var. 15.238 29.471 41.435 51.768 61.434 69.953 77.794  
## Dim.8 Dim.9 Dim.10 Dim.11  
## Variance 0.141 0.117 0.087 0.062  
## % of var. 7.705 6.392 4.724 3.385  
## Cumulative % of var. 85.500 91.891 96.615 100.000  
##   
## Individuals (the 10 first)  
## Dim.1 ctr cos2 Dim.2 ctr cos2 Dim.3  
## 1 | -0.298 0.106 0.086 | -0.328 0.137 0.105 | -0.327  
## 2 | -0.237 0.067 0.036 | -0.136 0.024 0.012 | -0.695  
## 3 | -0.369 0.162 0.231 | -0.300 0.115 0.153 | -0.202  
## 4 | -0.530 0.335 0.460 | -0.318 0.129 0.166 | 0.211  
## 5 | -0.369 0.162 0.231 | -0.300 0.115 0.153 | -0.202  
## 6 | -0.369 0.162 0.231 | -0.300 0.115 0.153 | -0.202  
## 7 | -0.369 0.162 0.231 | -0.300 0.115 0.153 | -0.202  
## 8 | -0.237 0.067 0.036 | -0.136 0.024 0.012 | -0.695  
## 9 | 0.143 0.024 0.012 | 0.871 0.969 0.435 | -0.067  
## 10 | 0.476 0.271 0.140 | 0.687 0.604 0.291 | -0.650  
## ctr cos2   
## 1 0.163 0.104 |  
## 2 0.735 0.314 |  
## 3 0.062 0.069 |  
## 4 0.068 0.073 |  
## 5 0.062 0.069 |  
## 6 0.062 0.069 |  
## 7 0.062 0.069 |  
## 8 0.735 0.314 |  
## 9 0.007 0.003 |  
## 10 0.643 0.261 |  
##   
## Categories (the 10 first)  
## Dim.1 ctr cos2 v.test Dim.2 ctr cos2  
## black | 0.473 3.288 0.073 4.677 | 0.094 0.139 0.003  
## Earl Grey | -0.264 2.680 0.126 -6.137 | 0.123 0.626 0.027  
## green | 0.486 1.547 0.029 2.952 | -0.933 6.111 0.107  
## alone | -0.018 0.012 0.001 -0.418 | -0.262 2.841 0.127  
## lemon | 0.669 2.938 0.055 4.068 | 0.531 1.979 0.035  
## milk | -0.337 1.420 0.030 -3.002 | 0.272 0.990 0.020  
## other | 0.288 0.148 0.003 0.876 | 1.820 6.347 0.102  
## tea bag | -0.608 12.499 0.483 -12.023 | -0.351 4.459 0.161  
## tea bag+unpackaged | 0.350 2.289 0.056 4.088 | 1.024 20.968 0.478  
## unpackaged | 1.958 27.432 0.523 12.499 | -1.015 7.898 0.141  
## v.test Dim.3 ctr cos2 v.test   
## black 0.929 | -1.081 21.888 0.382 -10.692 |  
## Earl Grey 2.867 | 0.433 9.160 0.338 10.053 |  
## green -5.669 | -0.108 0.098 0.001 -0.659 |  
## alone -6.164 | -0.113 0.627 0.024 -2.655 |  
## lemon 3.226 | 1.329 14.771 0.218 8.081 |  
## milk 2.422 | 0.013 0.003 0.000 0.116 |  
## other 5.534 | -2.524 14.526 0.197 -7.676 |  
## tea bag -6.941 | -0.065 0.183 0.006 -1.287 |  
## tea bag+unpackaged 11.956 | 0.019 0.009 0.000 0.226 |  
## unpackaged -6.482 | 0.257 0.602 0.009 1.640 |  
##   
## Categorical variables (eta2)  
## Dim.1 Dim.2 Dim.3   
## Tea | 0.126 0.108 0.410 |  
## How | 0.076 0.190 0.394 |  
## how | 0.708 0.522 0.010 |  
## sugar | 0.065 0.001 0.336 |  
## where | 0.702 0.681 0.055 |  
## lunch | 0.000 0.064 0.111 |

**Interpreation:**  MCA is another approach for dimensionality reduction. MCA works well with categerical variables, as PCA for numerical variables. #As summary show, the MCA gives us for example Eigenvalues, which include Variance, percentage of variance and cumulative percentage of variance for each dimension. As it shows, dimention 1, contains/contributes to 15.238 of variance in the data, Dim 2, 14% and Dim 3 nearly 12%. It gradually declines as I plot the in the following extra coding parts.  
Additionally, next part of summary shows the individuals (10 first here) for each dimension, contribution of each dimension (ctr), and squared correlation of each dimension (cos2). For example, Dim1, which the coordinated is -0.298, has contribution of 0.106 and squared correlation is 0.086. Note that Dim here is equalling to each principal component (PC) in the PCA. So, both approached (MCA and PCA) we transform the original data.  
The next part, shows Categories (the 10 first), has v-test values (like normal t-test) in addition to the same stuff explained above. v.test value for black tea is 4.677 which means the coordinates is significantly different from zero. all values below or above +/-1.96 means that.  
Final part of the summary shows Categorical variables (eta2) which shows squared correlation between each variable and dimension. The closer to 1, the stronger the correlation is. For example, how category and Dim.1 has strong correlation (0.708), which lunch and Dim.1 has no correlation (0.0). *For more info please check* [*https://vimeo.com/204251158*](https://vimeo.com/204251158)

Visualize MCA by biplot

plot(mca, invisible=c("ind"), habillage = "quali")

 **Interpreation** the plot shows the MCA factor map, which is made by plotting Dim1 and Dim2 in x and y axises, respectively. It made easy ro see the possible variables patterns. The shorter distance between the variables the more similar they are, e.g. No sugar and No lunch in middle of the graph, which makes a good sense :). But tea bag and unpacked are far from each other which shows dissimilarities between those.

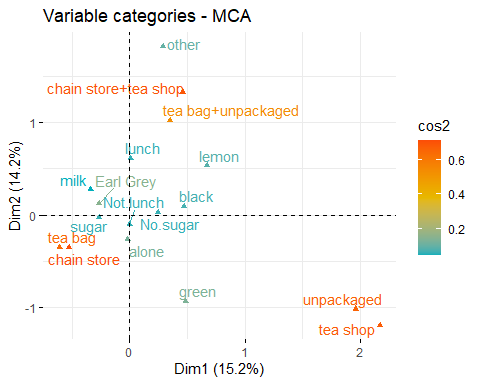
# Extra coding 4: other plotting options for multiple correspondence analysis

*inspritred from;* [*http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/114-mca-multiple-correspondence-analysis-in-r-essentials/*](http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/114-mca-multiple-correspondence-analysis-in-r-essentials/)

library("factoextra")

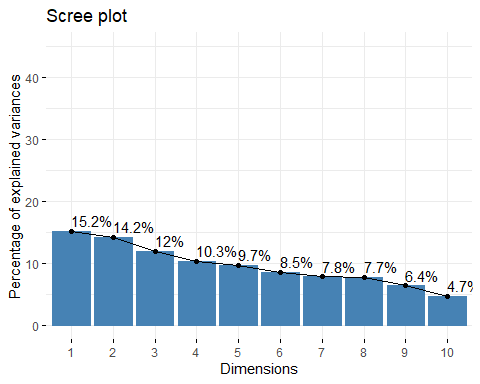
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

# Color by cos2 values: quality on the factor map  
fviz\_mca\_var(mca, col.var = "cos2",  
 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),   
 repel = TRUE, # Avoid text overlapping  
 ggtheme = theme\_minimal())

 The graph shows the squared correlation (cos2) in the variables. Which is a new thig to this is the same graph as above.

**Or yet another way of plotting**

factoextra::fviz\_screeplot(mca, addlabels = TRUE, ylim = c(0, 45))

 **Interpreation:**  We can see that by increasing the dimension, the percentage of explained variance is declining. It means that the first variables contribute to the variance of the data and make it kind of saturated, that for example 10th dimension has less to contribute to variance, because most of variance it already filled and shown by earlier dimensions.