# Data science in foods

Zhijun Wang

19/09/2021

## 1 Introduction

This is an R Markdown document to summary the necessary Data science in food applications.

The datasets were from built-in dataset in R packages, and the open-access resources from Quadram Institute. Example Datasets for Download.These datasets are distributed under the terms of **the Creative Commons Attribution License**, which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

## 2 Using R for maps

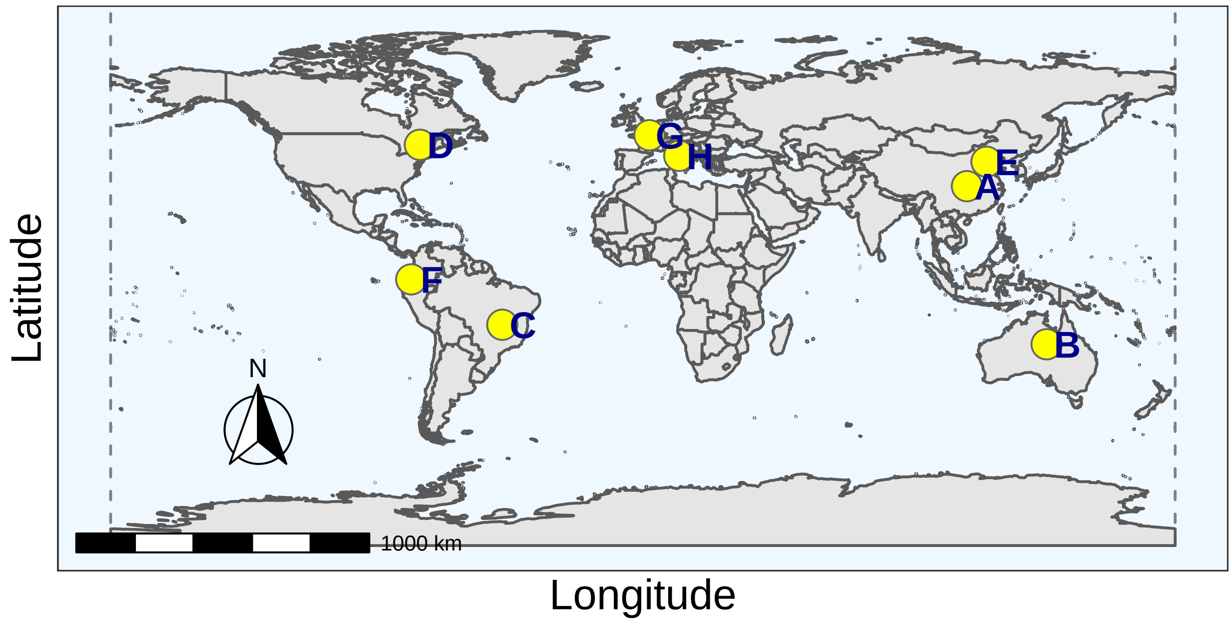
Many articles need to use maps to display some data. Professional drawing software such as ARCGIS is more expensive and takes a long time to learn. The advantage of drawing with R is that everyone can install R and R maps have no copyright issues and are easy to use. The main R packages used are **ggplot2**, **sf**, **rnaturalearth**, **rnaturalearthdata**, etc. When you need to modify the picture, you only need to modify the code, which eliminates the cumbersome modification of the picture.

> To determine the origin of the sample, so samples from different countries are collected, and the location of the origin needs to be marked on the map.

```
library(rnaturalearth)
library(rnaturalearthdata)
library(ggspatial)
library(sf)
library(ggplot2)
library(gpclib)
library(maptools)
library(maps)
library(mapdata)
library(sp)
library(raster)
library(RColorBrewer)
library(rgeos)
```

The sampling sits could be created by R:

The sampling sits created by R




## 3 Data visualisation

Examples : **The weight of chicks from different months and feed diet**.

Firstly, have a look at dataset. The **structure of dataset** is following:

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   578 obs. of  4 variables
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

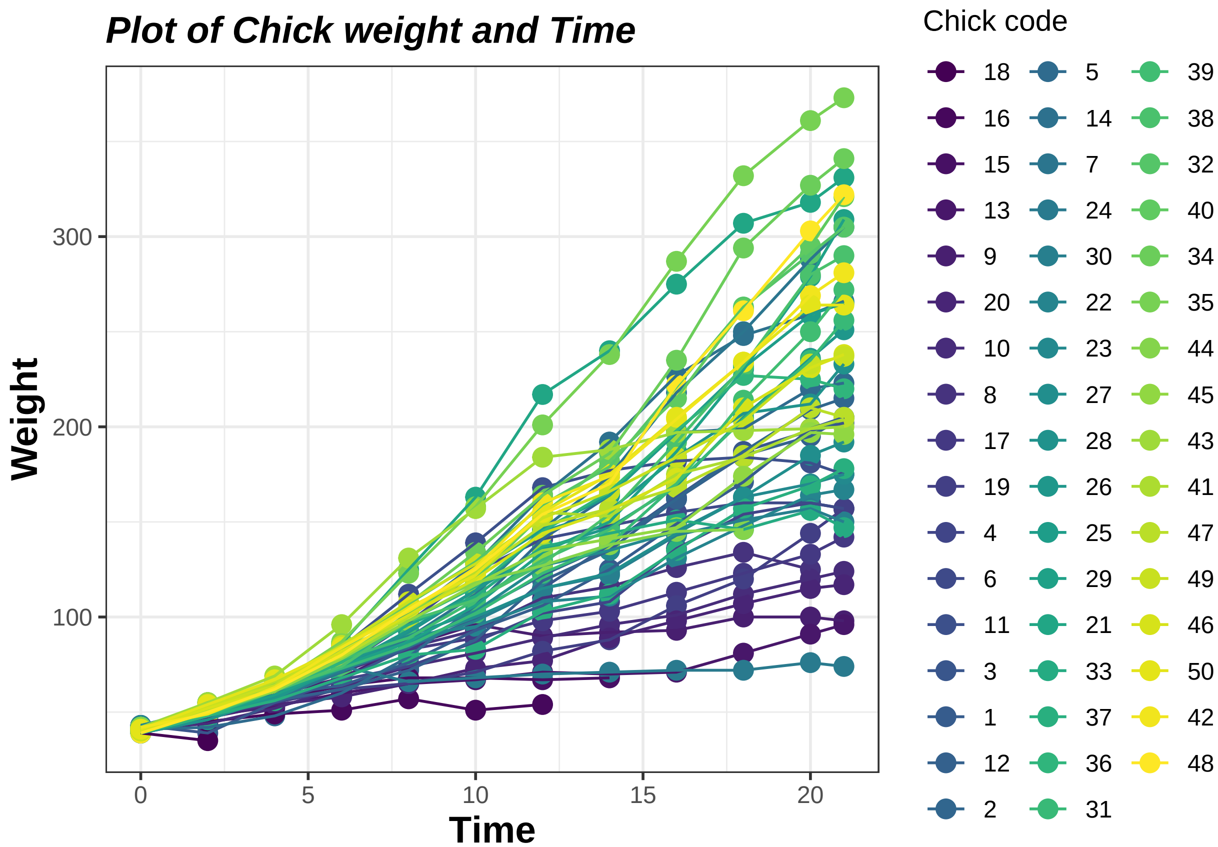
Then, have a look at the first 5 lines in datasets, we could find the dataset includes **weight**, **chick numbers**, **diet** and **times** variables.

```
##   weight Time Chick Diet
## 1     42    0     1    1
```

```
## 2     51    2     1     1
## 3     59    4     1     1
## 4     64    6     1     1
## 5     76    8     1     1
```

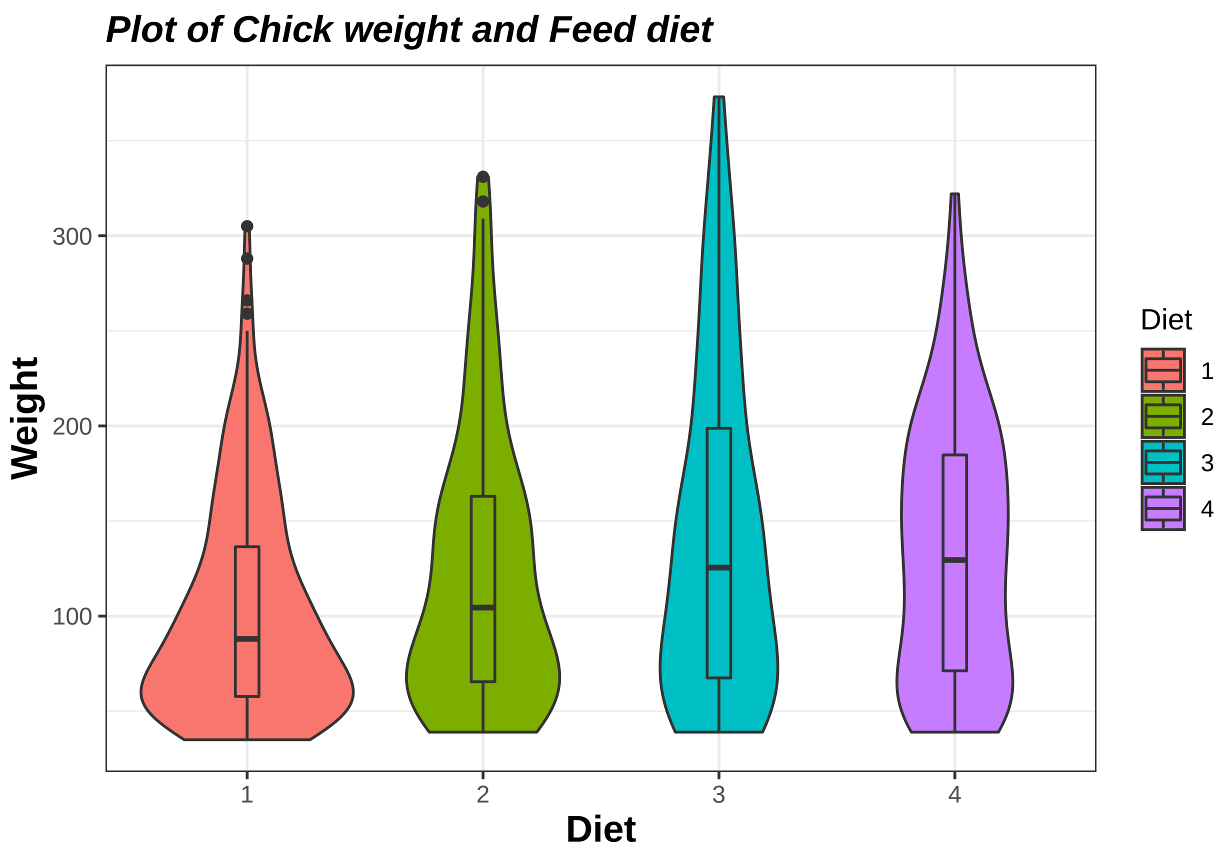
### 3.1 line plot

**To explore the relationship between weight and time using package ggplot2**




The results from plot indicated that the weight incresed with the increasing times.

**To explore the relationship between weight and time using violin plot**




### 3.2 the simple prediction

**To predict the weight based on time and diet types**

```
library(performance)

chick_model <- glm(weight ~ Diet + Chick + Time, family = poisson, data = Chick_data)

model_performance(chick_model)
```

```
## # Indices of model performance
##
## AIC      |      BIC | Nagelkerke's R2 |   RMSE | Sigma | Score_log | Score_spherical
## -----------------------------------------------------------------------------------
## 5282.750 | 5505.088 |           1.000 | 17.106 | 1.652 |    -4.482 |           0.034
```

The RMSE value of glm model is quite high, which indicated the model should be improved using more data.

# 4 The one-way analysis of variance (ANOVA)

## 4.1 Defination of ANOVA

Analysis of variance (ANOVA) is an analysis tool used in statistics that splits an observed aggregate variability found inside a data set into two parts: systematic factors and random factors. The systematic factors have a statistical influence on the given data set, while the random factors do not. Analysts use the ANOVA test to determine the influence that independent variables have on the dependent variable in a regression study– cited from **WILL KENTON**(https://www.investopedia.com/terms/a/anova.asp)

### 4.2 Normal distribution

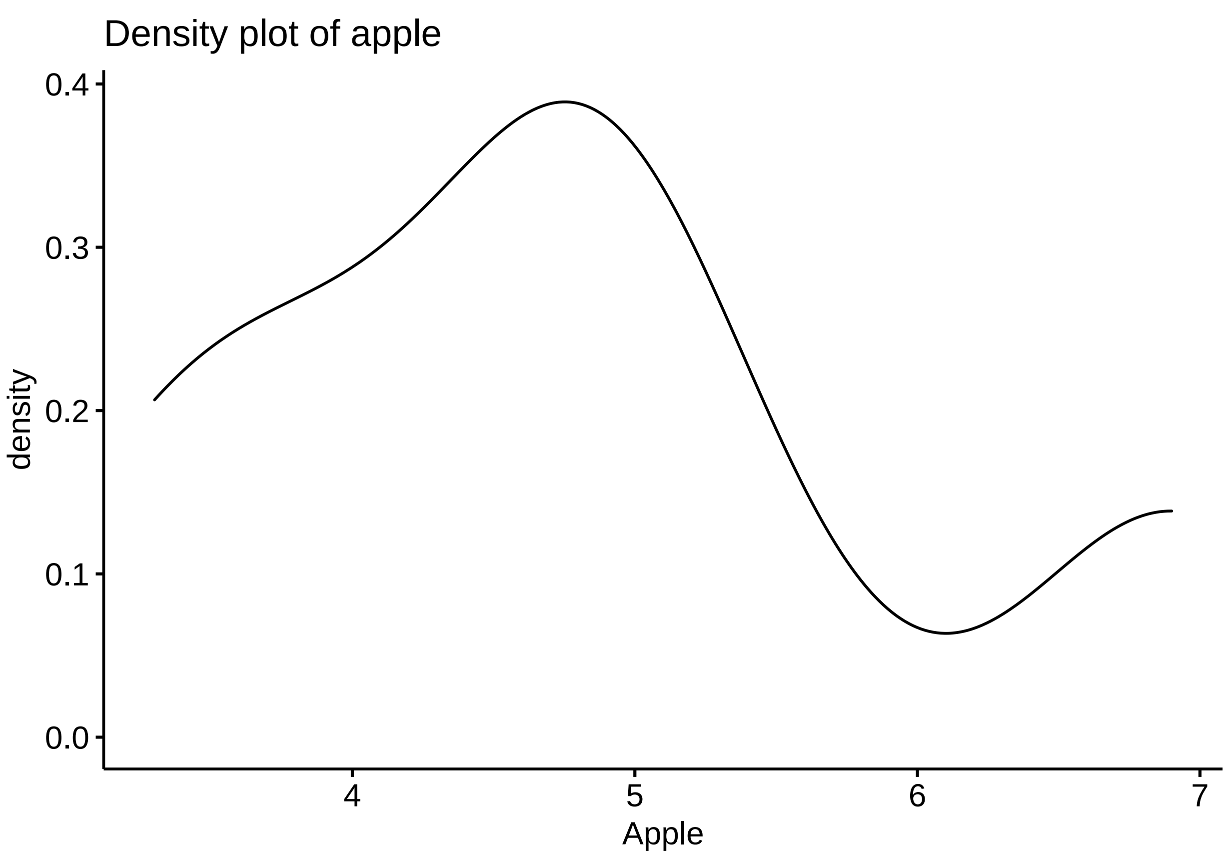
The data of most food experiments will be analyzed for significance. In the writing method section of many articles, we will mention what method is used for significance analysis. If P is less than 0.05, the difference between variables is considered to be significant. However, before the significance analysis, some preprocessing shoule carried out to make sure Normal distribution, otherwise the result is inaccurate. The following summarizes the data preprocessing that can be performed in R to meet the requirements of saliency analysis.

*Before statistical analysis of data such as significance analysis, the data should meet normally distributed.*

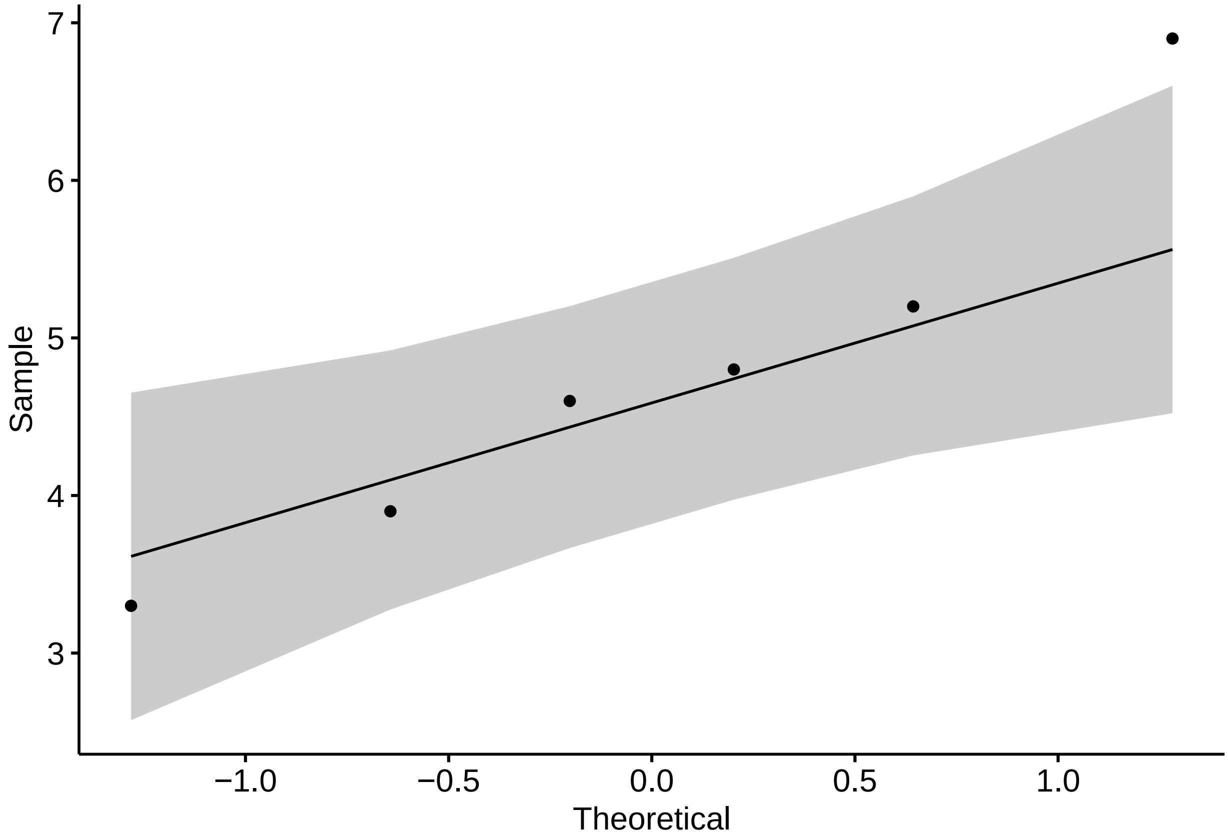
```
Apple <- data[1:6,4]
shapiro.test(Apple)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Apple
## W = 0.94874, p-value = 0.7301
```

```
library(ggpubr)
ggdensity(Apple,
          main = "Density plot of apple",
          xlab = "Apple")
```



```
ggqqplot(Apple)
```

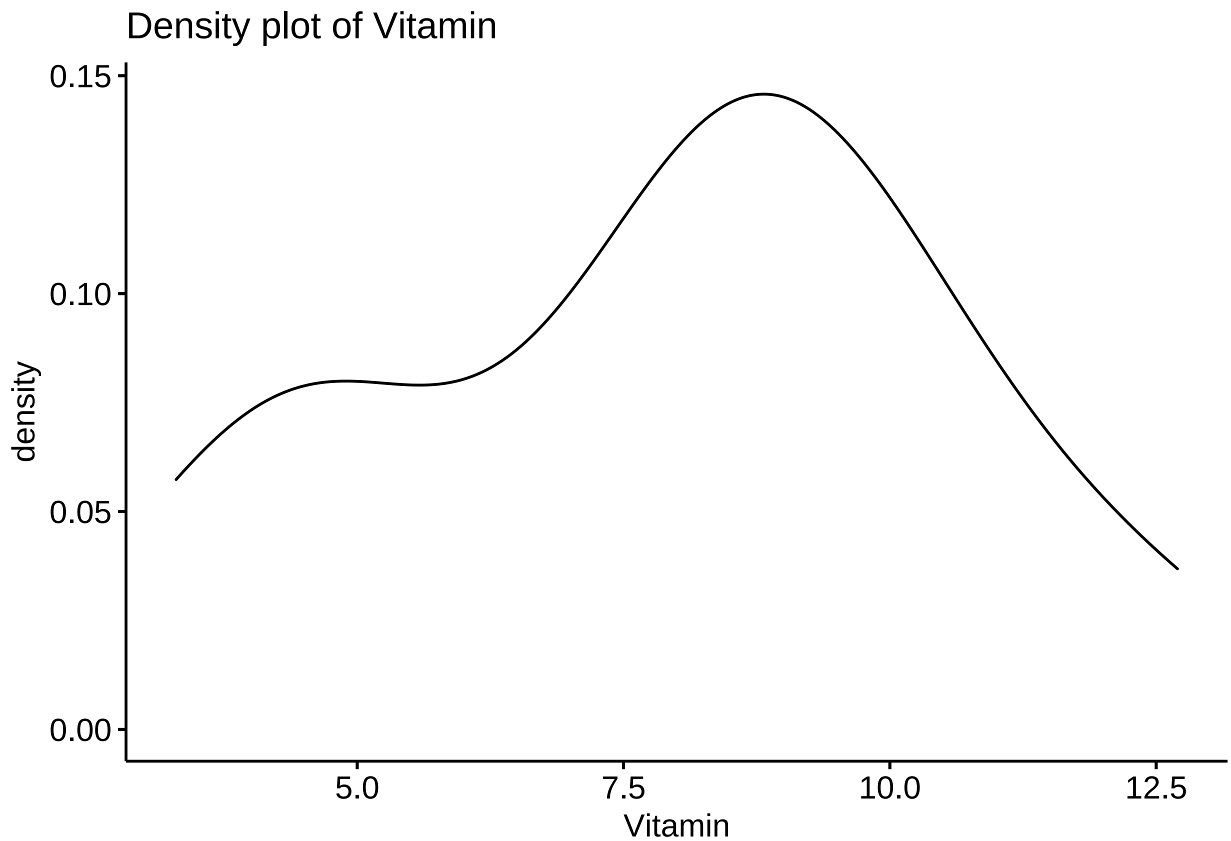



Shapiro-Wilk normality test

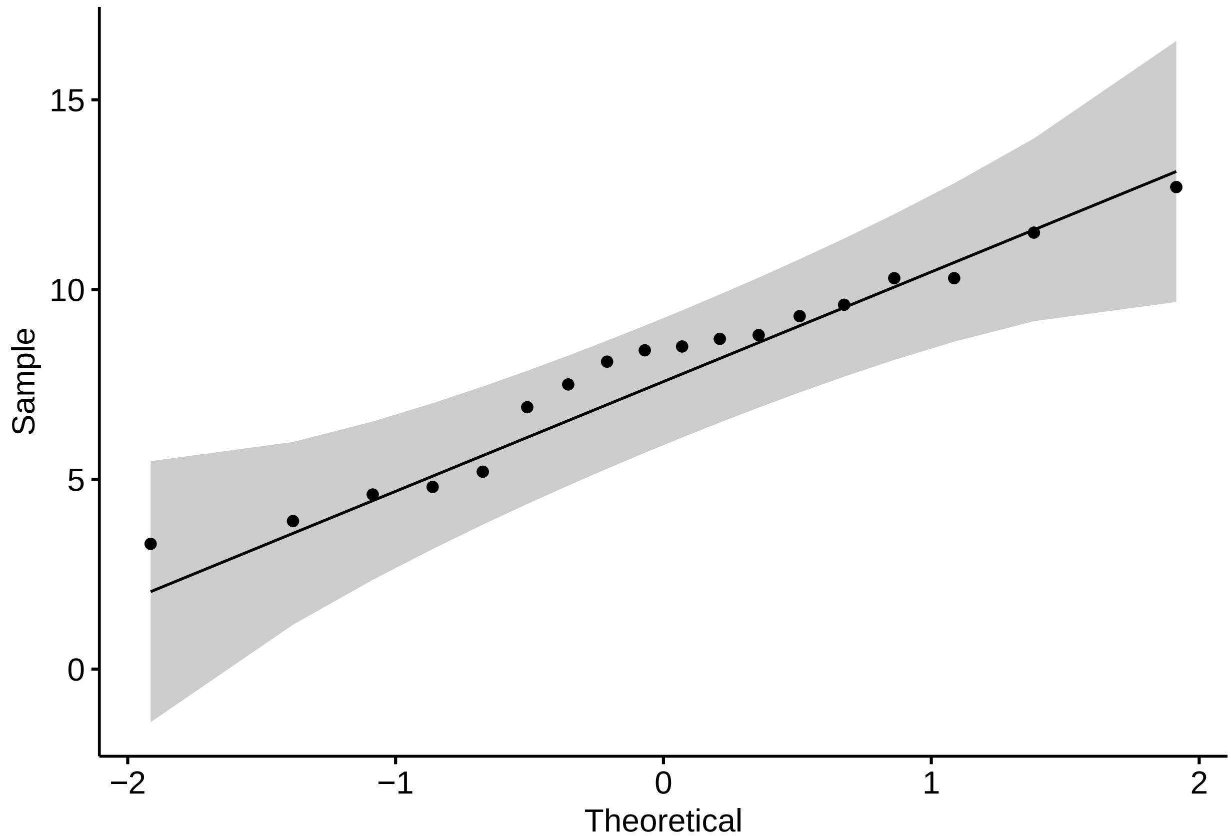
data: data$Vitamin W = 0.96026, p-value = 0.6066 When the P value here is greater than 0.05, it represents a normal distribution.

You can also observe the normal distribution graph:

```
library("ggpubr")
ggdensity(data$Vitamin,
          main = "Density plot of Vitamin",
          xlab = "Vitamin")
```




```
ggqqplot(data$Vitamin)
```




### 4.3 The examples of ANOVA using fruit dataset

```
##   Number Fruit Repeat Vitamin
## 1      1 Apple     A1     4.6
## 2      2 Apple     A2     3.9
## 3      3 Apple     A3     5.2
## 4      4 Apple     A4     6.9
## 5      5 Apple     A5     4.8
## 6      6 Apple     A6     3.3
```

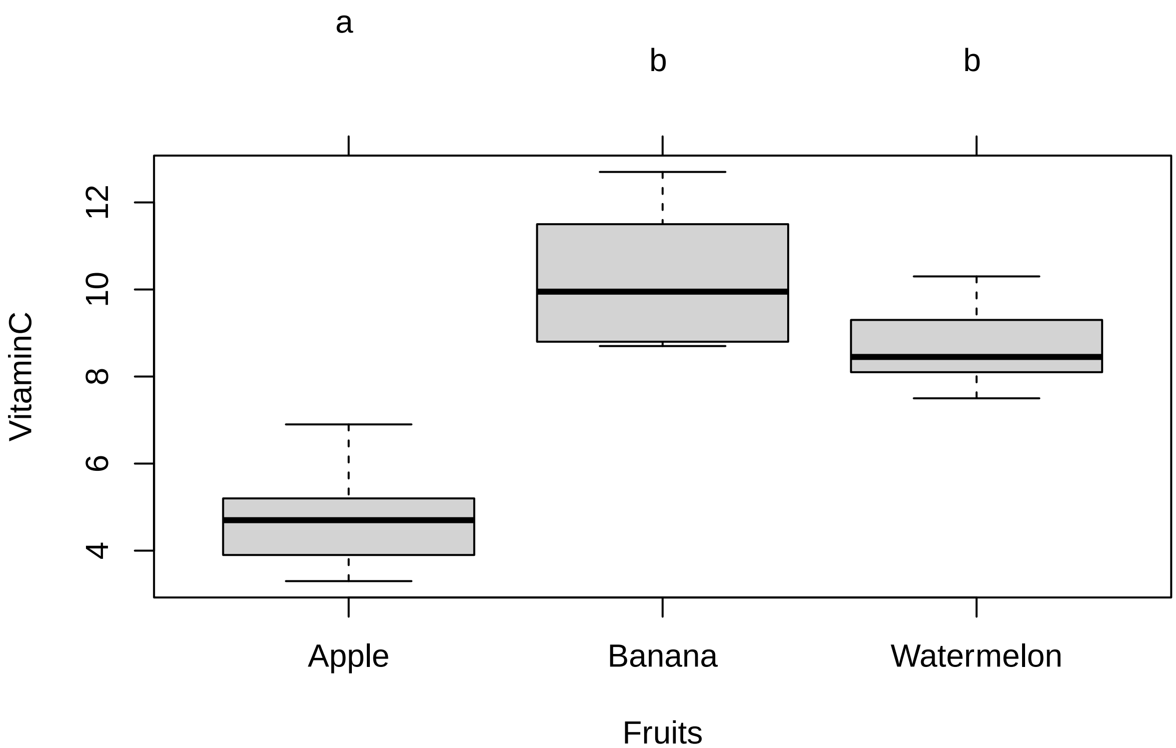
To compare the vitamin C contents of different fruits

```
##      Group.1         x
## 1      Apple  4.783333
## 2     Banana 10.266667
## 3 Watermelon  8.683333
```

```
##      Group.1         x
## 1      Apple 1.2384130
## 2     Banana 1.5807171
## 3 Watermelon 0.9847165
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Fruits       2  95.57   47.78   28.66 7.52e-06 ***
## Residuals   15  25.01    1.67
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = VitaminC ~ Fruits)
##
## $Fruits
##                       diff       lwr       upr     p adj
## Banana-Apple       5.483333  3.546906 7.4197605 0.0000068
## Watermelon-Apple   3.900000  1.963573 5.8364272 0.0002819
## Watermelon-Banana -1.583333 -3.519761 0.3530938 0.1184352
```




To save the ANOVA results after calculations

```
## # A tibble: 3 x 4
##   `data$Fruit` count  mean    sd
##   <fct>        <int> <dbl> <dbl>
## 1 Apple            6  4.78 1.24
## 2 Banana           6 10.3  1.58
## 3 Watermelon       6  8.68 0.985
```

# 5 Principal component analysis(PCA)

From this section, the machine learning will be discussed by foods data, the data from Mid-infrared spectroscopy will be used for Geographical origin of Extra Virgin Olive Oils (Quadram open dataset,2003).

### 5.1 Load dataset

```
##                 V1          V2          V3          V4          V5
## 1  Sample Number:           1           1           2           2
## 2     Group Code:           1           1           1           1
## 3     Wavenumbers      Greece      Greece      Greece      Greece
## 4         798.892 0.127523009 0.126498181 0.130411785 0.130022227
## 5        800.8215 0.127949615 0.127130974 0.130675401 0.130406662
## 6         802.751 0.129282219 0.128510777  0.13201661 0.132018029
## 7        804.6805 0.131174169  0.13033991 0.133824061 0.134007275
## 8          806.61 0.133590328 0.132527221 0.136095296 0.136270568
## 9        808.5395 0.136425525 0.135308508 0.138943757  0.13887477
## 10        810.469 0.139357827  0.13835292 0.141722779 0.141481132
```

### 5.2 Data wrangling

The raw data form are not suitable for R programming, therefore we need to transpose it ti suitable form.

```
## data transpose
oil_transpose <- as.data.frame(t(oil))

print(oil_transpose[1:10,1:5])
```

```
##                  V1          V2          V3          V4          V5
## V1  Sample Number: Group Code: Wavenumbers     798.892    800.8215
## V2                1           1      Greece 0.127523009 0.127949615
## V3                1           1      Greece 0.126498181 0.127130974
## V4                2           1      Greece 0.130411785 0.130675401
## V5                2           1      Greece 0.130022227 0.130406662
## V6                3           1      Greece 0.128601989 0.128789565
## V7                3           1      Greece 0.128217254 0.128282253
## V8                4           1      Greece 0.126174933 0.126732773
## V9                4           1      Greece 0.126466053 0.126915413
## V10               5           1      Greece 0.127060105 0.127551128
```

```
## rename some col names to make data analysis easier

colnames(oil_transpose)=oil_transpose[1,]
oil_transpose <- oil_transpose[-1,]
library(dplyr)
oil_transpose <- rename(oil_transpose,c("Group" ="Group Code:",
                                        "Number" ="Sample Number:",
                                        "Countries" ="Wavenumbers"))
## Get transposed dataset

oil.data <- oil_transpose

print(oil.data[1:10,1:5])
```

```
##     Number Group Countries     798.892    800.8215
## V2       1     1    Greece 0.127523009 0.127949615
## V3       1     1    Greece 0.126498181 0.127130974
## V4       2     1    Greece 0.130411785 0.130675401
## V5       2     1    Greece 0.130022227 0.130406662
## V6       3     1    Greece 0.128601989 0.128789565
## V7       3     1    Greece 0.128217254 0.128282253
```

```
## V8        4      1     Greece 0.126174933 0.126732773
## V9        4      1     Greece 0.126466053 0.126915413
## V10       5      1     Greece 0.127060105 0.127551128
## V11       5      1     Greece 0.126812707 0.127460743
```

## 5.3 Principal Component Analysis

The PCA is used to reduce the dimensionality of the spectral value, and initially explore the distribution of the sample.

The original data and related transformated data (MSC, SNV and Savitzky-Golay filtering) are used to obtain the PCA plots.

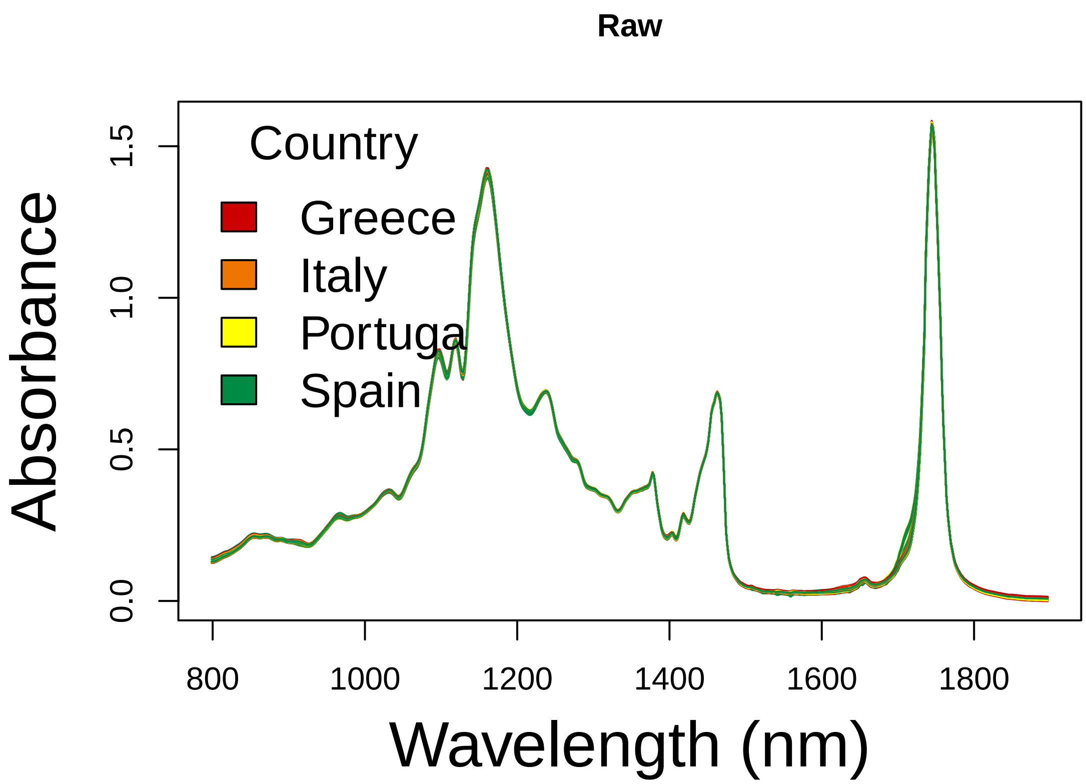
```
library(missMDA)
library(factoextra)
library(FactoMineR)
library(ggrepel)
library(ggplot2)
library(mixOmics)
library(EMSC)
library(pls)
library(prospectr)
```

## 5.4 Raw data PCA

Have a look at spectral of different pre-processing for MIR.

```
colors = c("#CD0000", "#EE7600", "#FFFF00", "#008B45")
col <- as.factor(oil.data$Countries)
group <- c("Greece","Italy","Portuga", "Spain")
```

**RAW spectral**



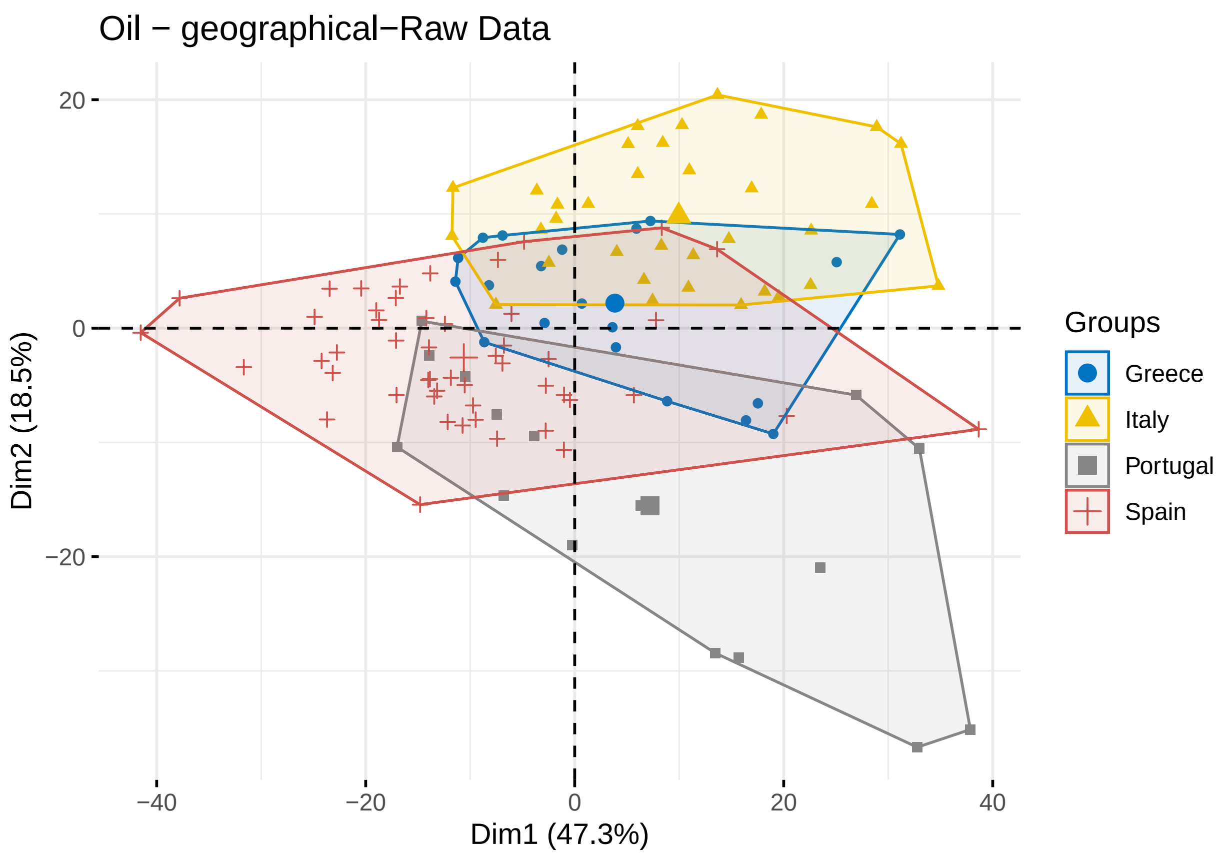

```
df_raw <- oil.data[,4:573]

df <- as.data.frame(apply(df_raw, 2, as.numeric))
print(df[1:10,1:5])
```

```
##      798.892  800.8215   802.751  804.6805    806.61
## 1  0.1275230 0.1279496 0.1292822 0.1311742 0.1335903
## 2  0.1264982 0.1271310 0.1285108 0.1303399 0.1325272
## 3  0.1304118 0.1306754 0.1320166 0.1338241 0.1360953
## 4  0.1300222 0.1304067 0.1320180 0.1340073 0.1362706
## 5  0.1286020 0.1287896 0.1300223 0.1320119 0.1344266
## 6  0.1282173 0.1282823 0.1296366 0.1317986 0.1340615
## 7  0.1261749 0.1267328 0.1282438 0.1298927 0.1317546
## 8  0.1264661 0.1269154 0.1282541 0.1299583 0.1320672
## 9  0.1270601 0.1275511 0.1289000 0.1306090 0.1329558
## 10 0.1268127 0.1274607 0.1287653 0.1306390 0.1331313
```

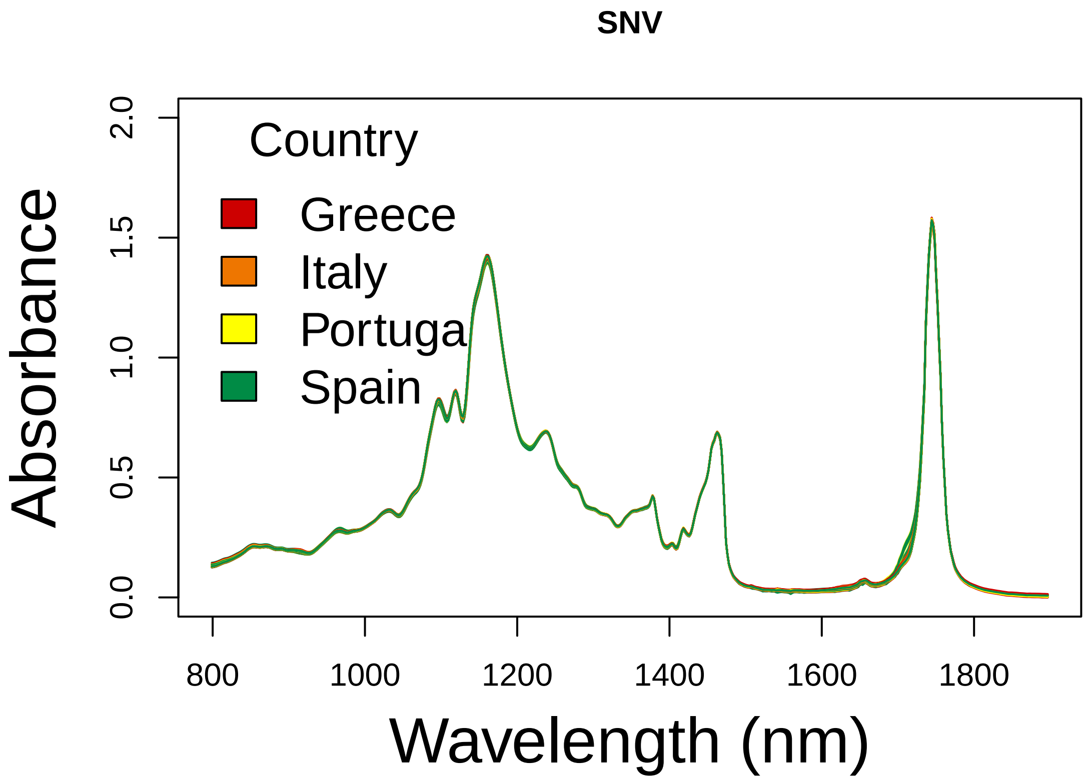
```
oil.pca <- PCA(df,scale.unit = TRUE,ncp = 5,graph = TRUE)
```

The PCA plot of raw Spectral




### 5.5 Standard Normal Variate

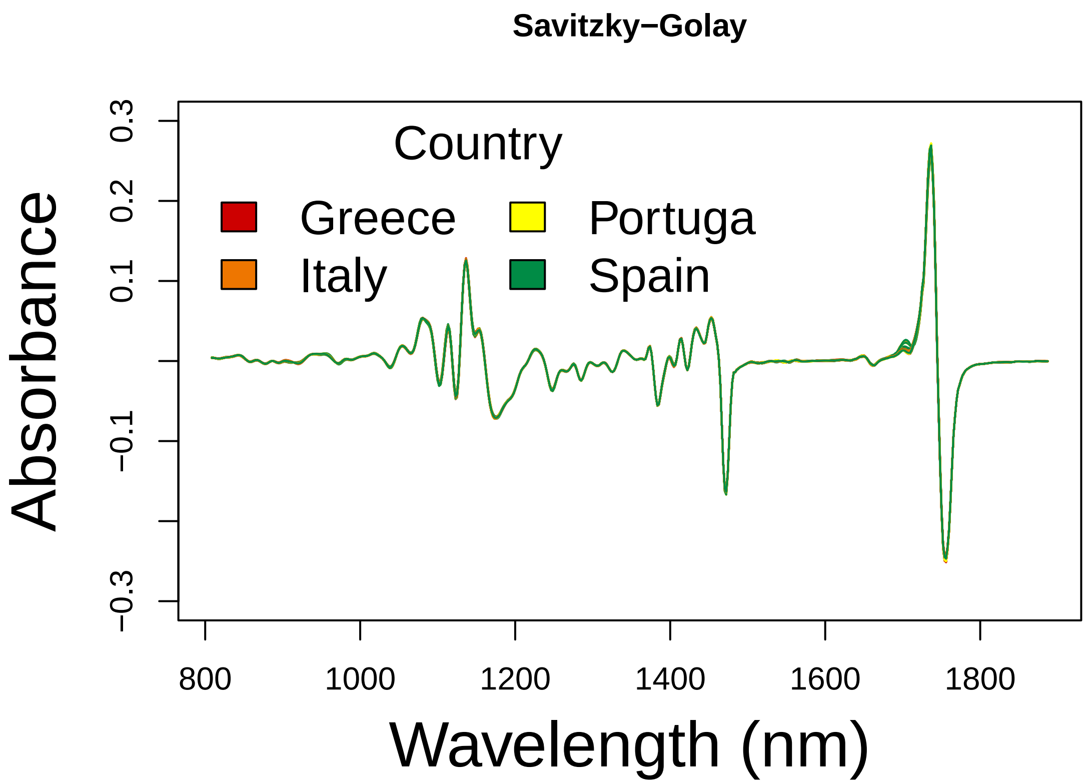
#### SNV spectral




p3 The PCA plot of SNV Spectral

### 5.6 Savitzky-Golay filtering

#### Savitzky-Golay spectral

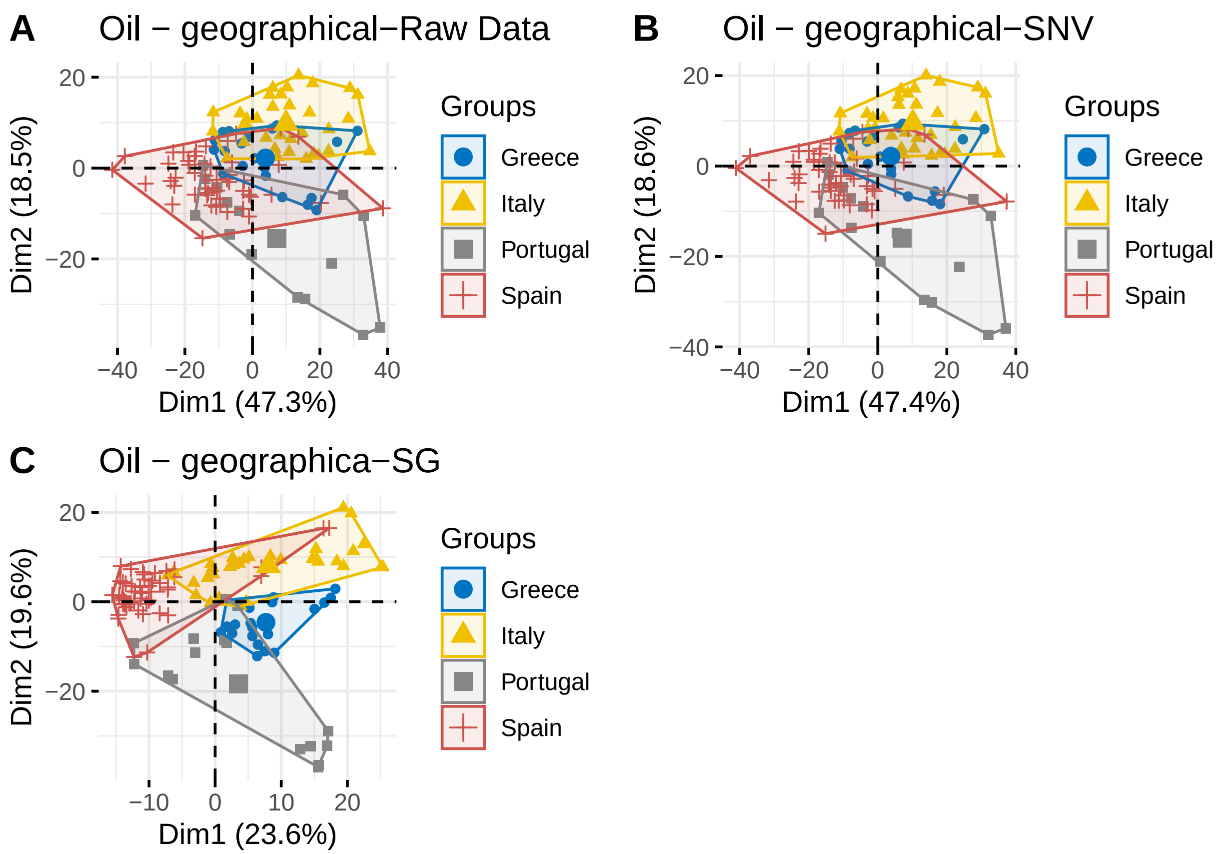



p4

The PCA plot of Savitzky-Golay Spectral:

g4

### 5.7 PCA plots




According to PCA plots, the best option is use RAW-SNV data set, therefore only **RAW-snv transformated data** will be used in the following data analysis such as classification models.

# 6 Discrimination analysis

**Using PLS-DA for geographical origin of oils**

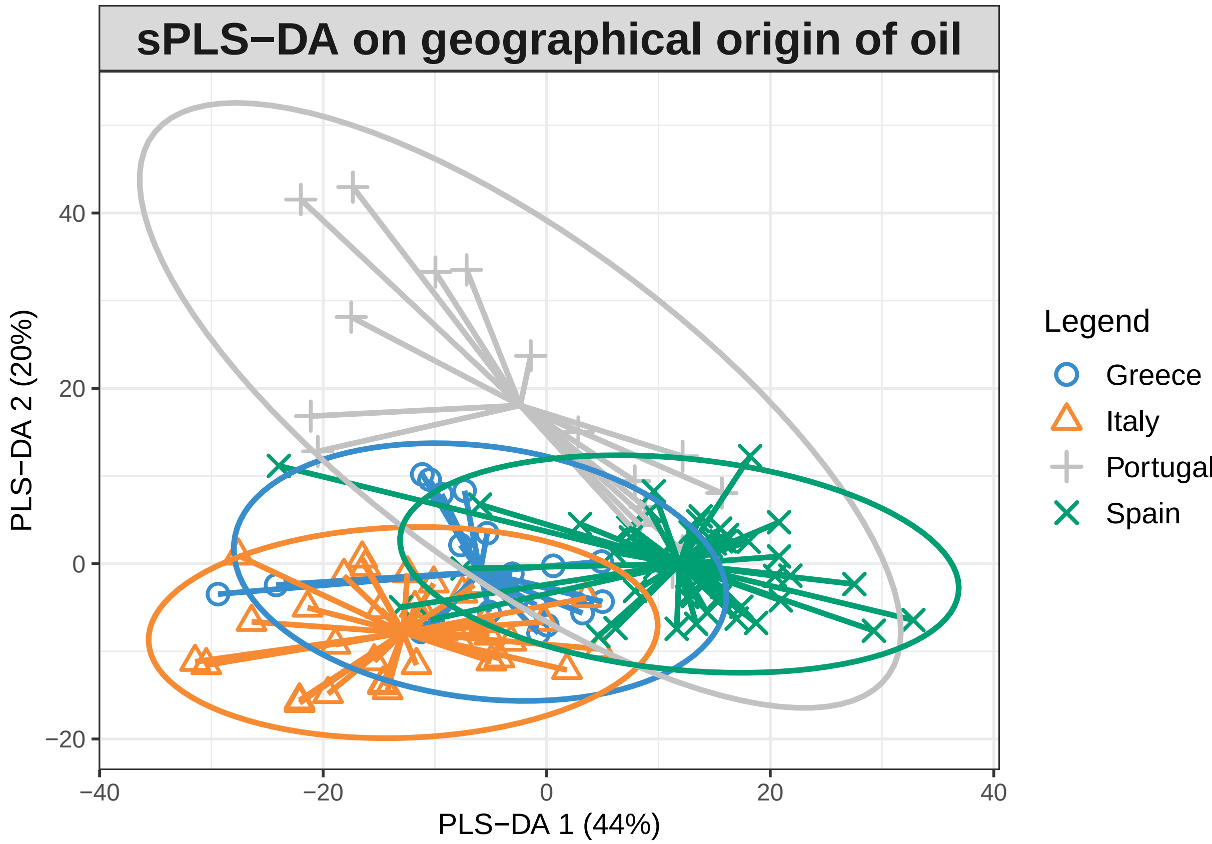
## 6.1 Data preprocess

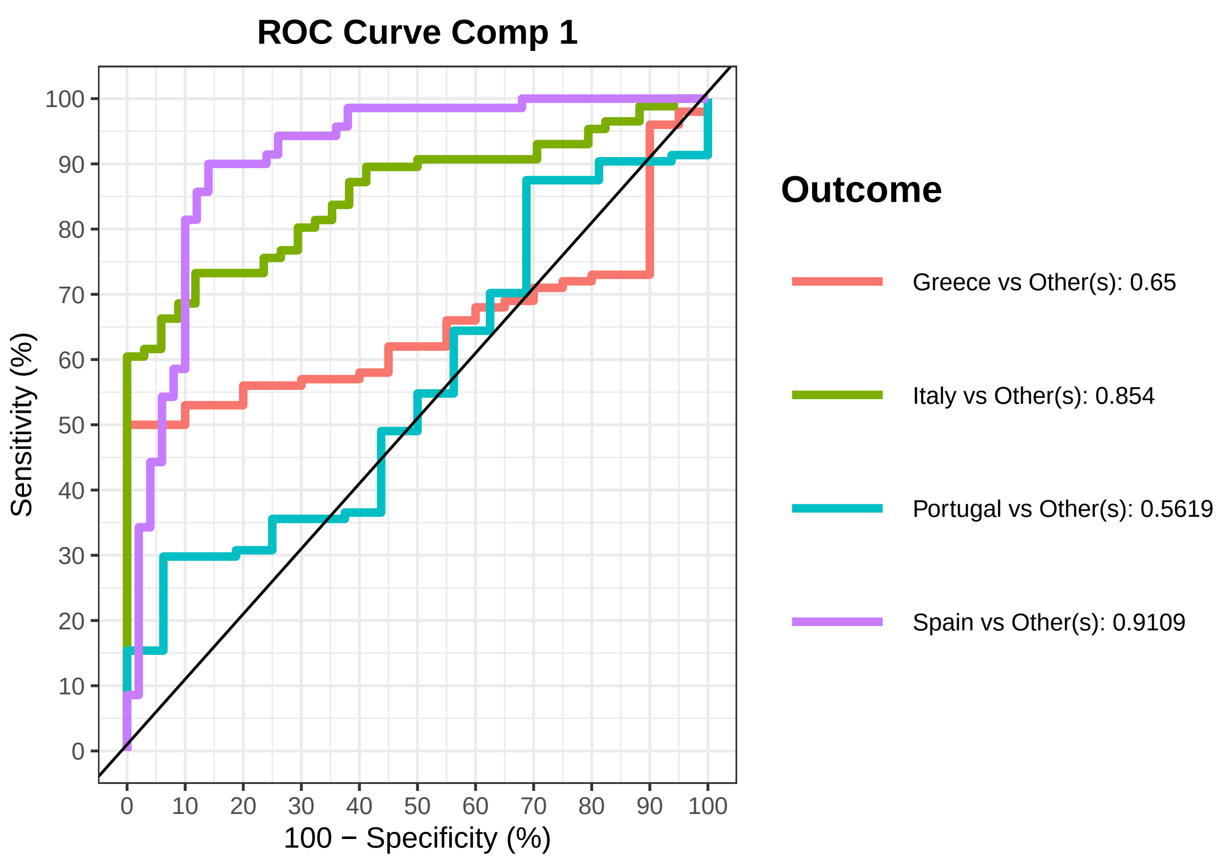
```
library(prospectr)

oil_data <- oil.snv[,4:573]
X <- oil_data

Country <- oil.snv[,3]
Y <- as.factor(Country)
```

### 6.2 To plot a simple pls-da plot

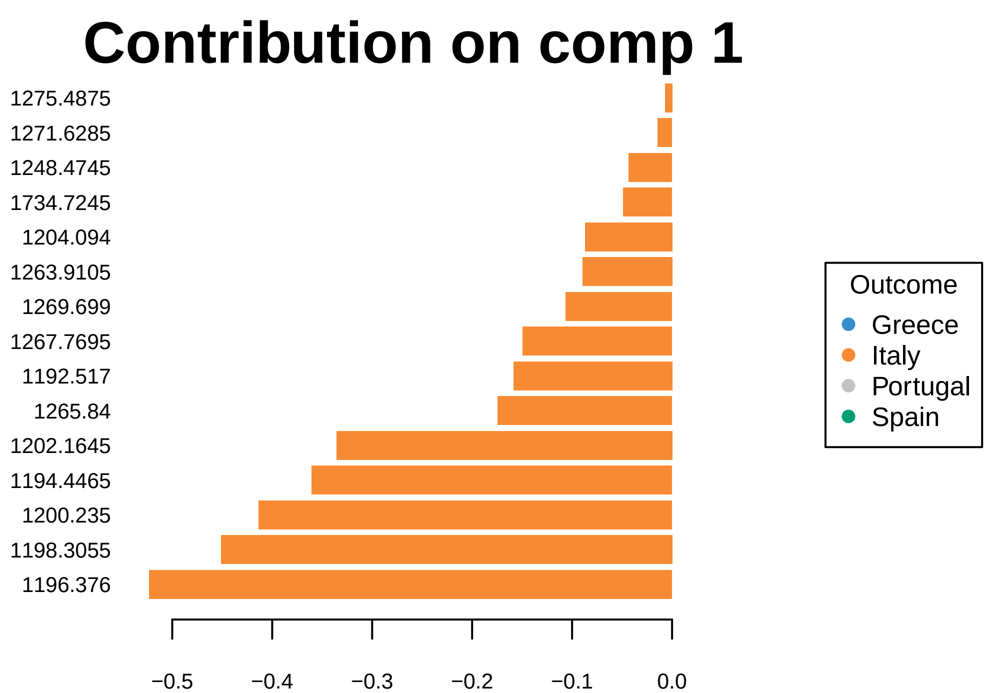


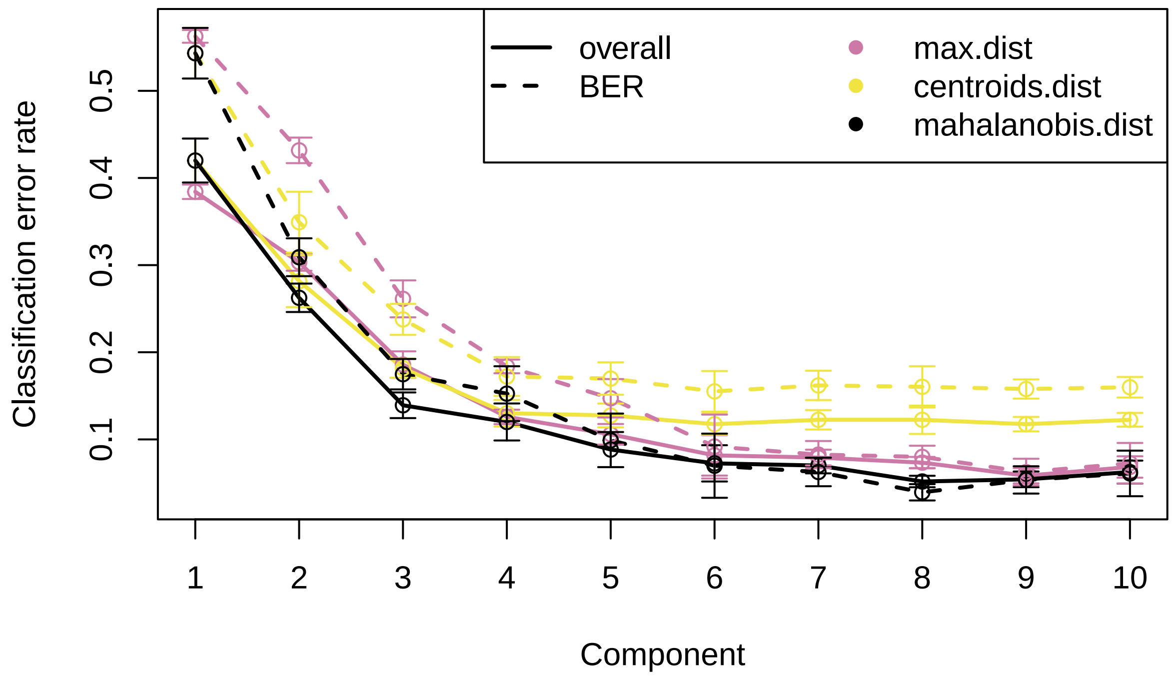
```
## $Comp1
##                         AUC   p-value
## Greece vs Other(s)   0.6500 3.464e-02
## Italy vs Other(s)    0.8540 1.663e-09
## Portugal vs Other(s) 0.5619 4.265e-01
## Spain vs Other(s)    0.9109 1.932e-14
##
## $Comp2
##                         AUC   p-value
## Greece vs Other(s)   0.6510 3.345e-02
## Italy vs Other(s)    0.9504 1.710e-14
## Portugal vs Other(s) 0.9339 2.491e-08
## Spain vs Other(s)    0.9106 2.021e-14
```

### 6.3 Variable selection outputs

The contribution of selected variables for PLS-DA model:



### 6.4 Tuning parameters and numerical outputs

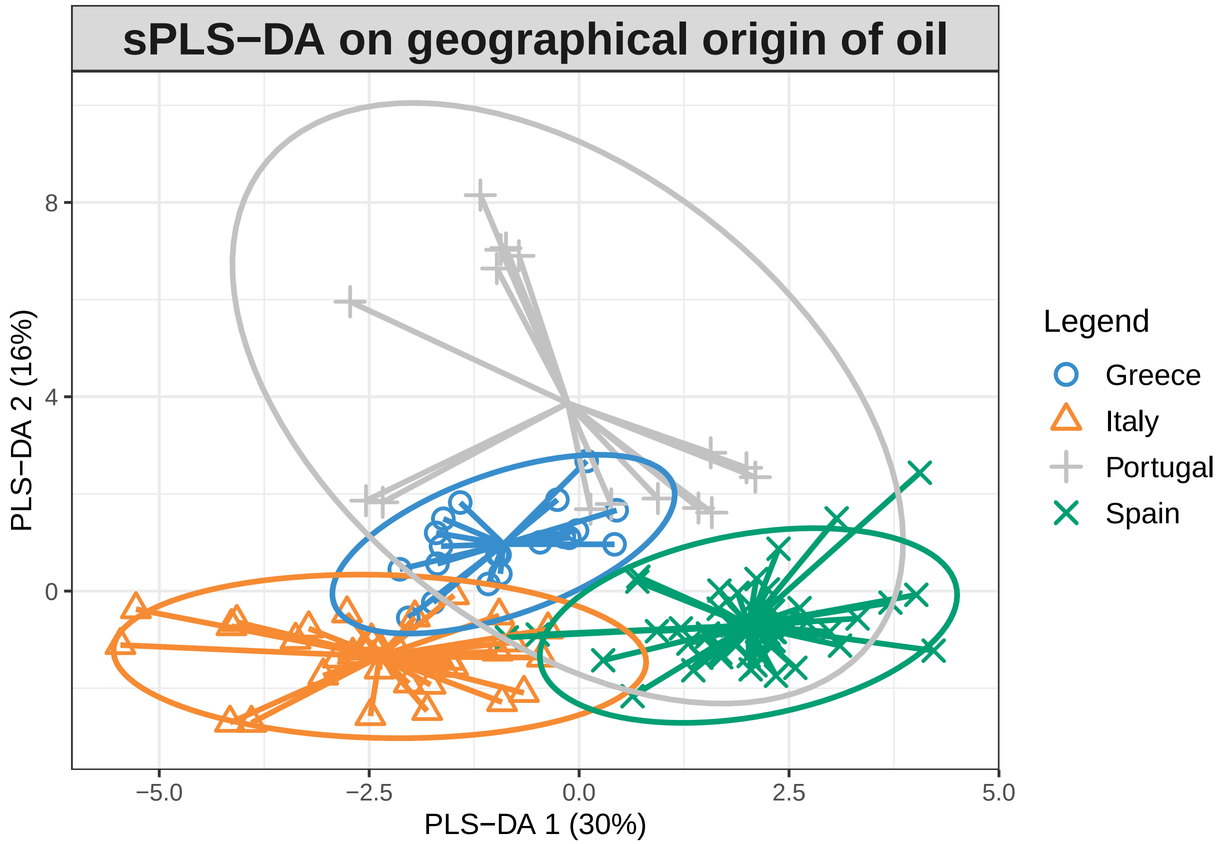



```
##   [1]    5    6    7    8    9   10   20   30   40   50   60   70   80   90  100
```

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

```
## comp1 comp2 comp3 comp4 comp5 comp6 comp7
##     7     5    50    10     5    80    40
```

### 6.5 The final pls-da model using tuning setting




### 6.6 Discrimination accuracy based on traing and testing data

```
library(caret)
oildata <- cbind(Country, oil_data)

set.seed(100)

# Step 1: Get row numbers for the training data
trainRowNumbers <- createDataPartition(oildata$Country, p=0.7, list=FALSE)

# Step 2: Create the training  dataset
trainData <- oildata[trainRowNumbers,]

# Step 3: Create the test dataset
testData <- oildata[-trainRowNumbers,]

plsda_model <- caret::plsda(trainData[,2:571],factor(trainData$Country), ncomp = 6, probMethod = "Bayes


C1 <- confusionMatrix(predict(plsda_model, trainData[,2:571]),as.factor(trainData$Country))

C2 <- confusionMatrix(predict(plsda_model, testData[,2:571]),as.factor(testData$Country))
```

The Confusion Matrix for geographical origin using training data

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Greece Italy Portugal Spain
##   Greece       14     0        0     0
##   Italy         0    24        0     3
##   Portugal      0     0       12     0
##   Spain         0     0        0    32
##
## Overall Statistics
##
##                Accuracy : 0.9647
##                  95% CI : (0.9003, 0.9927)
##     No Information Rate : 0.4118
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9502
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Greece Class: Italy Class: Portugal Class: Spain
## Sensitivity                 1.0000       1.0000          1.0000       0.9143
## Specificity                 1.0000       0.9508          1.0000       1.0000
## Pos Pred Value              1.0000       0.8889          1.0000       1.0000
## Neg Pred Value              1.0000       1.0000          1.0000       0.9434
## Prevalence                  0.1647       0.2824          0.1412       0.4118
## Detection Rate              0.1647       0.2824          0.1412       0.3765
## Detection Prevalence        0.1647       0.3176          0.1412       0.3765
## Balanced Accuracy           1.0000       0.9754          1.0000       0.9571
```

The Confusion Matrix for geographical origin using testing data

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Greece Italy Portugal Spain
##   Greece        5     0        0     1
##   Italy         0    10        0     0
##   Portugal      0     0        4     1
##   Spain         1     0        0    13
##
## Overall Statistics
##
##                Accuracy : 0.9143
##                  95% CI : (0.7694, 0.982)
##     No Information Rate : 0.4286
##     P-Value [Acc > NIR] : 2.195e-09
##
##                   Kappa : 0.8778
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                        Class: Greece Class: Italy Class: Portugal Class: Spain
## Sensitivity                   0.8333       1.0000          1.0000       0.8667
## Specificity                   0.9655       1.0000          0.9677       0.9500
## Pos Pred Value                0.8333       1.0000          0.8000       0.9286
## Neg Pred Value                0.9655       1.0000          1.0000       0.9048
## Prevalence                    0.1714       0.2857          0.1143       0.4286
## Detection Rate                0.1429       0.2857          0.1143       0.3714
## Detection Prevalence          0.1714       0.2857          0.1429       0.4000
## Balanced Accuracy             0.8994       1.0000          0.9839       0.9083
```

The overall accuracy for training set is **96%** and for testing set is **91%**.

# 7 Regression for prediction

In this part, the data set *Orange* is used to show how regression could be used for prediction in food science.

## 7.1 Load data

Have a look at *dataset Orange*:

```
## Grouped Data: circumference ~ age | Tree
##   Tree  age circumference
## 1    1  118            30
## 2    1  484            58
## 3    1  664            87
## 4    1 1004           115
## 5    1 1231           120
## 6    1 1372           142
```

## 7.2 Linear regression model

To establish the relationship **Linear regression model** between the *circumference* and *age*.

```
##
## Call:
## lm(formula = circumference ~ age, data = orange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.310 -14.946  -0.076  19.697  45.111
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.399650   8.622660   2.018   0.0518 .
## age          0.106770   0.008277  12.900 1.93e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.74 on 33 degrees of freedom
## Multiple R-squared:  0.8345, Adjusted R-squared:  0.8295
## F-statistic: 166.4 on 1 and 33 DF,  p-value: 1.931e-14
```

To plot the **Linear regression model**:

```
coeff=coefficients(lm_model)

coeff
```

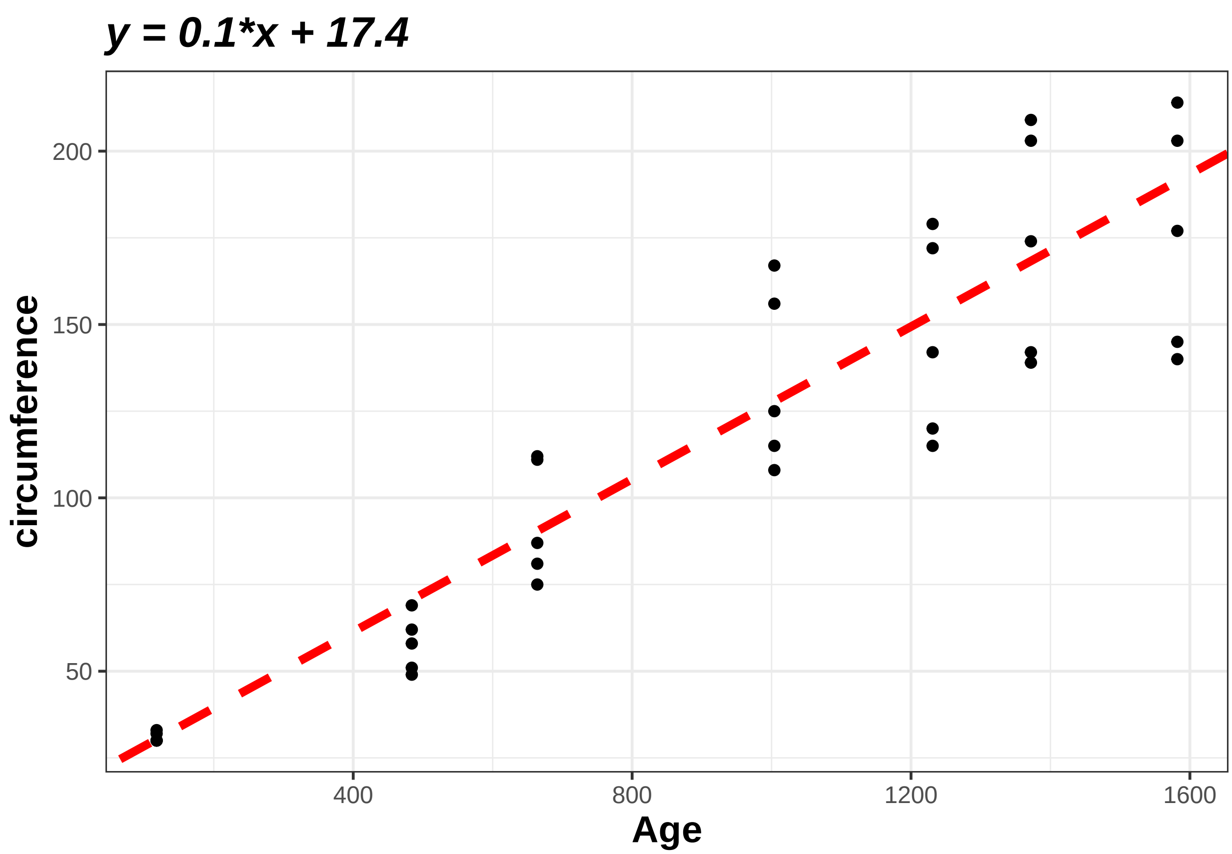
```
## (Intercept)         age
##  17.3996502   0.1067703
```

```
# Equation of the line :
eq = paste0("y = ", round(coeff[2],1), "*x + ", round(coeff[1],1))

eq
```

```
## [1] "y = 0.1*x + 17.4"
```

### 7.3 Linear regression model plot




## 8 Correlation analysis

The correlation heatmap is quite helpful to show correlations between sifferent variables such as contents of compositions.

The example of **EU Population Prospects** will be used for correlation analysis of Quality of life index.

### 8.1 Load data

```
##     Countries Total.population..thousands. Birth.rate Mortality.rate
## 1     Austria                     9043.070      9.999          9.972
```

```
## 2     Belgium                    11632.300     10.659          9.775
## 3      France                    65426.200     10.947          9.460
## 4     Germany                    83900.500      9.464         11.570
## 5  Luxembourg                      634.814     10.355          7.074
## 6 Netherlands                    17173.100     10.119          9.043
##   Life.expectancy Infant.mortality.rate Number.of.children.per.woman
## 1          81.813                 2.578                        1.562
## 2          81.942                 2.482                        1.719
## 3          82.916                 2.773                        1.841
## 4          81.634                 2.471                        1.614
## 5          82.556                 2.556                        1.417
## 6          82.567                 2.201                        1.667
##   Growth.rate Population.aged.65.and.more..thousands.
## 1        3.34                                  1761.5
## 2        3.38                                  2276.9
## 3        2.38                                 13796.3
## 4        0.59                                 18438.9
## 5       12.93                                    93.0
## 6        2.23                                  3512.0
```

### 8.2 Calculation significants and correlations coefficients

```r
library(Hmisc)

res <- rcorr(as.matrix(dat))
# Extract the correlation coefficients

res$r
```

```
##                                         Total.population..thousands. Birth.rate
## Total.population..thousands.                              1.00000000 -0.1664787
## Birth.rate                                               -0.16647866  1.0000000
## Mortality.rate                                            0.54201100 -0.4354210
## Life.expectancy                                          -0.22762820  0.3436773
## Infant.mortality.rate                                    -0.02444783  0.2414875
## Number.of.children.per.woman                              0.40503563  0.4664838
## Growth.rate                                              -0.48033379  0.2174427
## Population.aged.65.and.more..thousands.                   0.99982299 -0.1785649
##                                         Mortality.rate Life.expectancy
## Total.population..thousands.                 0.5420110     -0.22762820
## Birth.rate                                  -0.4354210      0.34367729
## Mortality.rate                               1.0000000     -0.65121537
## Life.expectancy                             -0.6512154      1.00000000
## Infant.mortality.rate                       -0.2913914      0.71768987
## Number.of.children.per.woman                 0.5007717     -0.07491628
## Growth.rate                                 -0.8872834      0.39471555
## Population.aged.65.and.more..thousands.      0.5502695     -0.23449257
##                                         Infant.mortality.rate
## Total.population..thousands.                      -0.02444783
## Birth.rate                                         0.24148750
## Mortality.rate                                    -0.29139140
## Life.expectancy                                    0.71768987
## Infant.mortality.rate                              1.00000000
## Number.of.children.per.woman                      -0.06790686
```

```
## Growth.rate                                          0.27830654
## Population.aged.65.and.more..thousands.            -0.02806558
##                                         Number.of.children.per.woman
## Total.population..thousands.                              0.40503563
## Birth.rate                                                0.46648378
## Mortality.rate                                            0.50077173
## Life.expectancy                                          -0.07491628
## Infant.mortality.rate                                    -0.06790686
## Number.of.children.per.woman                              1.00000000
## Growth.rate                                              -0.74128367
## Population.aged.65.and.more..thousands.                   0.40127427
##                                         Growth.rate
## Total.population..thousands.             -0.4803338
## Birth.rate                                0.2174427
## Mortality.rate                           -0.8872834
## Life.expectancy                           0.3947155
## Infant.mortality.rate                     0.2783065
## Number.of.children.per.woman             -0.7412837
## Growth.rate                               1.0000000
## Population.aged.65.and.more..thousands.  -0.4835130
##                                         Population.aged.65.and.more..thousands.
## Total.population..thousands.                                         0.99982299
## Birth.rate                                                          -0.17856495
## Mortality.rate                                                       0.55026954
## Life.expectancy                                                     -0.23449257
## Infant.mortality.rate                                               -0.02806558
## Number.of.children.per.woman                                         0.40127427
## Growth.rate                                                         -0.48351304
## Population.aged.65.and.more..thousands.                              1.00000000
```

```
# Extract p-values
res$P
```

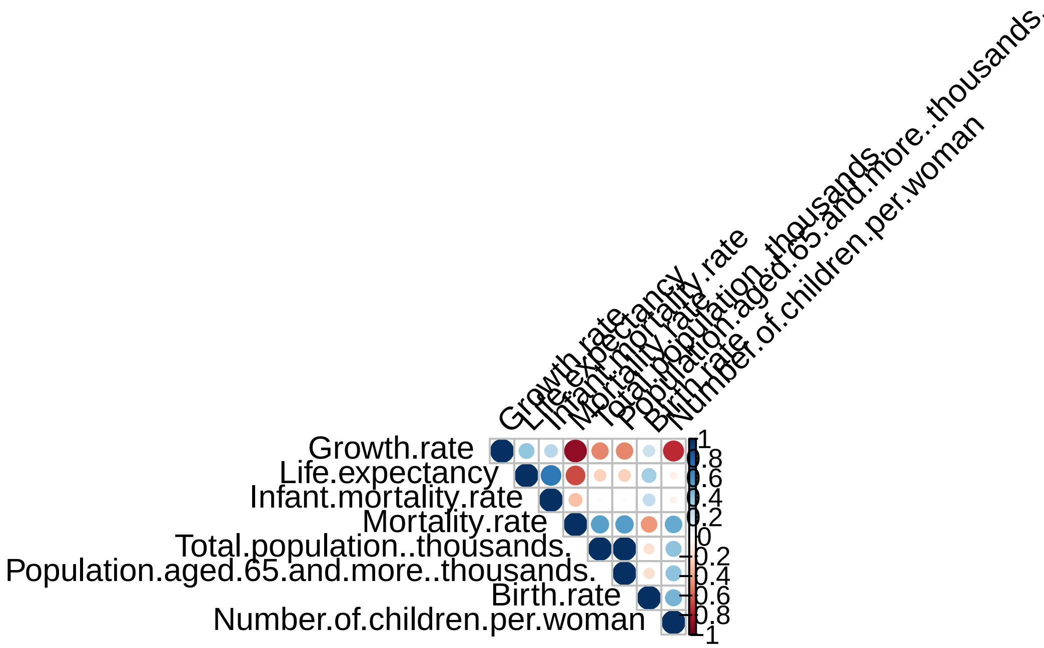
```
##                                         Total.population..thousands. Birth.rate
## Total.population..thousands.                                      NA  0.6935720
## Birth.rate                                              6.935720e-01         NA
## Mortality.rate                                          1.652249e-01  0.2809065
## Life.expectancy                                         5.877110e-01  0.4045485
## Infant.mortality.rate                                   9.541786e-01  0.5645063
## Number.of.children.per.woman                            3.195299e-01  0.2439472
## Growth.rate                                             2.283143e-01  0.6049639
## Population.aged.65.and.more..thousands.                 1.386447e-11  0.6722397
##                                         Mortality.rate Life.expectancy
## Total.population..thousands.               0.165224895      0.58771102
## Birth.rate                                 0.280906497      0.40454851
## Mortality.rate                                      NA      0.08026243
## Life.expectancy                            0.080262432              NA
## Infant.mortality.rate                      0.483780501      0.04501218
## Number.of.children.per.woman               0.206218157      0.86005667
## Growth.rate                                0.003284339      0.33318642
## Population.aged.65.and.more..thousands.    0.157599816      0.57617804
##                                         Infant.mortality.rate
## Total.population..thousands.                       0.95417858
## Birth.rate                                         0.56450626
## Mortality.rate                                     0.48378050
```

```
## Life.expectancy                                     0.04501218
## Infant.mortality.rate                                       NA
## Number.of.children.per.woman                        0.87306551
## Growth.rate                                         0.50449426
## Population.aged.65.and.more..thousands.             0.94740466
##                                         Number.of.children.per.woman
## Total.population..thousands.                              0.31952989
## Birth.rate                                                0.24394715
## Mortality.rate                                            0.20621816
## Life.expectancy                                           0.86005667
## Infant.mortality.rate                                     0.87306551
## Number.of.children.per.woman                                      NA
## Growth.rate                                               0.03532671
## Population.aged.65.and.more..thousands.                   0.32447618
##                                         Growth.rate
## Total.population..thousands.            0.228314326
## Birth.rate                              0.604963872
## Mortality.rate                          0.003284339
## Life.expectancy                         0.333186421
## Infant.mortality.rate                   0.504494257
## Number.of.children.per.woman            0.035326712
## Growth.rate                                      NA
## Population.aged.65.and.more..thousands. 0.224800623
##                                         Population.aged.65.and.more..thousands.
## Total.population..thousands.                                       1.386447e-11
## Birth.rate                                                         6.722397e-01
## Mortality.rate                                                     1.575998e-01
## Life.expectancy                                                    5.761780e-01
## Infant.mortality.rate                                              9.474047e-01
## Number.of.children.per.woman                                       3.244762e-01
## Growth.rate                                                        2.248006e-01
## Population.aged.65.and.more..thousands.                                      NA
```

### 8.3 Correlation plots

```r
library(corrplot)

corrplot(res$r, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```


Growth.rate
Life.expectancy
Infant.mortality.rate
Mortality.rate
Total.population..thousands.
Population.aged.65.and.more..thousands.
Birth.rate
Number.of.children.per.woman
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1