Exploring data using R

Kamarul Imran Musa, Wan Nor Arifin

2017-07-07

Contents

Chapter 1

Introduction to R

1.1 Installing R and RStudio

Install R base package: http://www.r-project.org/

Install RStudio: http://www.rstudio.com/

1.2 Getting familiar with the interface

Consists of 4 tabs: 1. Source 2. Console 3. Environment & History 4. Misc. Most important Plots, Packages & Help

1.3 Basic tasks in R

1.3.1 R Script

Text here.

1.3.2 Setting working directory

Text here.

1.3.3 Packages

Text here.

1.3.3.1 Installation

```
install.packages("package.name")
```

1.3.3.2 Loading

```
library("package.name")
```

1.3.4 Data management

Text here.

1.3.4.1 Loading data

```
read.csv("file.name")
```

For SPSS file, need `foreign` package

```
library("foreign")
read.spss("file.name")
```

1.3.4.2 Data dimension

```
dim(data)
```

1.3.4.3 Entering data

text here

1.3.4.4 Editing data

text here

Chapter 2

Textual

In this chapter, we will go through a number of R functions for basic statistics. We will mostly use the builtin functions (from R standard library). Extra packages will be introduced whenever necessary.

2.1 Descriptive statistics

We are going to use builtin datasets in R. You can view the available datasets by

```
data()
```

```
## Data sets in package 'datasets':

## AirPassengers                      Monthly Airline Passenger Numbers 1949-1960
## BJsales                            Sales Data with Leading Indicator
## BJsales.lead (BJsales)             Sales Data with Leading Indicator
## BOD                                Biochemical Oxygen Demand
## CO2                                Carbon Dioxide Uptake in Grass Plants
## ...
```

View the data, for example

```
women
```

```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
## 7      64    132
## 8      65    135
## 9      66    139
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
## 14     71    159
## 15     72    164
```

View the dimension, i.e. number of subjects and variables

```
dim(women)
```

```
## [1] 15  2
```

Obtaining mean

```
mean(women$weight)
```

```
## [1] 136.7333
```

and median

```
median(women$weight)
```

```
## [1] 135
```

and sd

```
sd(women$weight)
```

```
## [1] 15.49869
```

and IQR

```
IQR(women$weight)
```

```
## [1] 23.5
```

There 9 types of IQR in R, the default one is type 7. You may change this to type 6 (Minitab and SPSS),

```
IQR(women$weight, type = 6)
```

```
## [1] 27
```

and minimum, maximum and range

```
min(women$weight)
```

```
## [1] 115
```

```
max(women$weight)
```

```
## [1] 164
```

```
range(women$weight)
```

```
## [1] 115 164
```

However, it is actually simpler to obtain most these in one single command for both weight and height

```
summary(women)
```

```
##      height         weight
##  Min.   :58.0   Min.   :115.0
##  1st Qu.:61.5   1st Qu.:124.5
##  Median :65.0   Median :135.0
##  Mean   :65.0   Mean   :136.7
##  3rd Qu.:68.5   3rd Qu.:148.0
##  Max.   :72.0   Max.   :164.0
```

even simpler, all of the statistics using *psych* package

```
install.packages("psych")
```

```
library(psych)
describe(women)
```

```
##        vars  n   mean    sd median trimmed   mad min max range skew
## height    1 15  65.00  4.47     65   65.00  5.93  58  72    14 0.00
## weight    2 15 136.73 15.50    135  136.31 17.79 115 164    49 0.23
##        kurtosis   se
## height    -1.44 1.15
## weight    -1.34 4.00
```

2.2 Tables

2.2.1 Count, proportion, percentage and cross-tabulation

Use *birthwt* dataset from MASS package.

```
install.packages("MASS")
```

```
library(MASS)
head(birthwt)  # First six subjects
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
## 91   0  21 124    3     0   0  0  0   0 2622
```

Count and proportion,

```
table(birthwt$smoke)
```

```
##
##   0   1
## 115  74
```

```
prop.table(table(birthwt$smoke))
```

```
##
##         0         1
## 0.6084656 0.3915344
```

Cross-tabulation of smoking vs low birth weight baby,

```
table(birthwt$smoke, birthwt$low)  # without row/column labels
```

```
##
##      0  1
##   0 86 29
##   1 44 30
```

```
table("Smoking status" = birthwt$smoke, "Low birth weight" = birthwt$low)  # with row/column labels
```

```
##                Low birth weight
## Smoking status  0  1
##              0 86 29
```

```
##              1 44 30
```

To add value labels to the data for a nicer table, we use *factor*

```
birthwt$smoking = factor(birthwt$smoke, levels = 0:1, labels = c("Non-smoker", "Smoker"))
birthwt$low.weight = factor(birthwt$low, levels = 0:1, labels = c("Low <2.5kg", "Normal >2.5kg"))
head(birthwt)  # we added two new variables with factors
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt    smoking low.weight
## 85   0  19 182    2     0   0  0  1   0 2523 Non-smoker Low <2.5kg
## 86   0  33 155    3     0   0  0  0   3 2551 Non-smoker Low <2.5kg
## 87   0  20 105    1     1   0  0  0   1 2557     Smoker Low <2.5kg
## 88   0  21 108    1     1   0  0  1   2 2594     Smoker Low <2.5kg
## 89   0  18 107    1     1   0  0  1   0 2600     Smoker Low <2.5kg
## 91   0  21 124    3     0   0  0  0   0 2622 Non-smoker Low <2.5kg
```

```
table(birthwt$smoking)
```

```
##
## Non-smoker     Smoker
##        115         74
```

```
prop.table(table(birthwt$smoking))*100  # in percent
```

```
##
## Non-smoker     Smoker
##   60.84656   39.15344
```

```
cbind(n = table(birthwt$smoking), "%" = 100*prop.table(table(birthwt$smoking)))  # using cbind
```

```
##              n        %
## Non-smoker 115 60.84656
## Smoker      74 39.15344
```

```
table(birthwt$smoking, birthwt$low.weight)
```

```
##
##              Low <2.5kg Normal >2.5kg
##   Non-smoker         86            29
##   Smoker             44            30
```

Save table for later view and analysis,

```
smoke.x.weight = table(birthwt$smoking, birthwt$low.weight)
smoke.x.weight
```

```
##
##              Low <2.5kg Normal >2.5kg
##   Non-smoker         86            29
##   Smoker             44            30
```

2.2.2 Entering table data

```
smoking = as.table(rbind(c(15, 5), c(7, 13)))
smoking
```

```
##    A  B
## A 15  5
```

```
## B  7 13
```

```
str(smoking)
```

```
##  table [1:2, 1:2] 15 7 5 13
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2] "A" "B"
##   ..$ : chr [1:2] "A" "B"
```

```
dimnames(smoking) = list(
  Smoking = c("Yes", "No"),
  Lung.CA = c("Yes", "No")
)
smoking
```

```
##        Lung.CA
## Smoking Yes No
##     Yes  15  5
##     No    7 13
```

Chapter 3

Graphical

Test GIT Test GIT 2 - commit

Chapter 4

Reporting results

Chapter 5

Data and variable manipulation

Chapter 6

Grammar of variables

6.1 Prepare folder and data

6.1.1 Set the working directory

This can be done in 2 ways:

1. Using codes
2. Using point and click

To use point and click, use the down arrow button next to *More* . Then click 'Set as working directory'

6.2 Read Data

```
library(foreign)
data_qol<-read.dta('qol.dta',convert.factors = T)
str(data_qol)
```

```
## 'data.frame':    365 obs. of  13 variables:
##  $ id       : num  308 335 94 329 350 22 171 274 332 147 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age      : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ tahundx  : num  14 4 5 10 13 4 4 15 13 3 ...
##  $ tx      : Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
##  $ group    : Factor w/ 2 levels "\"group A\"",..: 2 2 1 2 2 1 1 1 2 1 ...
##  $ complica : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 2 1 ...
##  $ hba1c    : num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
##  $ fbs      : num  6.9 4.8 8 3.6 12.5 8.5 NA NA NA NA ...
##  $ rbs      : num  16.7 7.4 13.2 7.4 NA 7.8 9.4 7.8 NA 12.4 ...
##  $ tg_total : num  0.92 1.66 0.74 0.94 3.01 1.3 NA 1.9 NA NA ...
##  $ choleste : num  7.09 2.91 5.94 3.27 7.1 3.54 NA 5.7 NA NA ...
##  $ ADDQSCORE: num  0 -0.222 -0.333 -0.36 -0.44 ...
##  - attr(*, "datalabel")= chr ""
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%10.0g" "%10.0g" "%10.0g" "%10.0g" ...
##  - attr(*, "types")= int  255 255 255 255 255 255 255 255 255 255 ...
##  - attr(*, "val.labels")= chr  "" "sex" "" "" ...
```

```
##  - attr(*, "var.labels")= chr  "id_no" "sex" "" "" ...
##  - attr(*, "version")= int 8
##  - attr(*, "label.table")=List of 4
##   ..$ sex     : Named int  0 1
##   .. ..- attr(*, "names")= chr  "female" "male"
##   ..$ tx      : Named int  1 2 3 4
##  .. ..- attr(*, "names")= chr  "diet only" "OHA and diet only" "insulin and diet only" "all"
##   ..$ group   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "\"group A\"" "\"group B\""
##   ..$ complica: Named int  0 1
##   .. ..- attr(*, "names")= chr  "no" "yes"
```

6.3 Browse data

1. First few rows
2. Last few rows

```
head(data_qol)
```

```
##    id    sex age tahundx                    tx     group complica hba1c
## 1 308 female  55      14 insulin and diet only "group B"      yes   8.1
## 2 335   male  41       4                   all "group B"       no   8.0
## 3  94 female  50       5     OHA and diet only "group A"       no   7.5
## 4 329 female  47      10                   all "group B"      yes   9.4
## 5 350 female  67      13                   all "group B"       no  11.7
## 6  22   male  57       4     OHA and diet only "group A"      yes   8.1
##    fbs  rbs tg_total choleste  ADDQSCORE
## 1  6.9 16.7     0.92     7.09  0.0000000
## 2  4.8  7.4     1.66     2.91 -0.2222222
## 3  8.0 13.2     0.74     5.94 -0.3333333
## 4  3.6  7.4     0.94     3.27 -0.3600000
## 5 12.5   NA     3.01     7.10 -0.4400000
## 6  8.5  7.8     1.30     3.54 -0.5000000
```

```
tail(data_qol)
```

```
##      id  sex age tahundx                tx     group complica hba1c  fbs
## 360  14 male  45      10 OHA and diet only "group A"       no   9.6 12.6
## 361 170 male  57       4 OHA and diet only "group A"       no    NA   NA
## 362 214 male  48       5 OHA and diet only "group A"       no    NA   NA
## 363 174 male  45       2 OHA and diet only "group A"       no   8.5   NA
## 364 130 male  64      16 OHA and diet only "group A"       no   6.1  3.8
## 365 219 male  46       2         diet only "group A"       no   5.9   NA
##      rbs tg_total choleste ADDQSCORE
## 360   NA       NA       NA -8.833333
## 361  9.4       NA       NA -8.833333
## 362 10.7       NA       NA -9.000000
## 363  9.6       NA       NA -9.000000
## 364  7.9       NA       NA -9.000000
## 365  6.3       NA       NA -9.000000
```

6.4 Grammar of variables

6.4.1 Select columns

6.5 Select columns

Let us create a new dataframe with only id, sex and hba1c as the variables

```
data_qol2<-subset(data_qol, select = c('sex', 'age', 'hba1c'))
str(data_qol2)
```

```
## 'data.frame':    365 obs. of  3 variables:
##  $ sex  : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age  : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ hba1c: num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
```

alternatively, we can use other subsetting functions

```
data_qol3<-data_qol[,c('sex','age','hba1c')]
str(data_qol3)
```

```
## 'data.frame':    365 obs. of  3 variables:
##  $ sex  : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age  : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ hba1c: num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
```

6.5.1 Select rows

```
data_qol4<-subset(data_qol, age > 30)
str(data_qol4)
```

```
## 'data.frame':    363 obs. of  13 variables:
##  $ id       : num  308 335 94 329 350 22 171 274 332 147 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age      : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ tahundx  : num  14 4 5 10 13 4 4 15 13 3 ...
##  $ tx       : Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
##  $ group    : Factor w/ 2 levels "\"group A\"",..: 2 2 1 2 2 1 1 1 2 1 ...
##  $ complica : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 2 1 ...
##  $ hba1c    : num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
##  $ fbs      : num  6.9 4.8 8 3.6 12.5 8.5 NA NA NA NA ...
##  $ rbs      : num  16.7 7.4 13.2 7.4 NA 7.8 9.4 7.8 NA 12.4 ...
##  $ tg_total : num  0.92 1.66 0.74 0.94 3.01 1.3 NA 1.9 NA NA ...
##  $ choleste : num  7.09 2.91 5.94 3.27 7.1 3.54 NA 5.7 NA NA ...
##  $ ADDQSCORE: num  0 -0.222 -0.333 -0.36 -0.44 ...
```

```
summary(data_qol4$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   32.00   47.00   53.00   52.91   59.00   80.00
```

alternatively, we can use other subsetting functions

```
data_qol5<-data_qol[data_qol$age>30,]
str(data_qol5)
```

```
## 'data.frame':    363 obs. of  13 variables:
##  $ id       : num  308 335 94 329 350 22 171 274 332 147 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age      : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ tahundx  : num  14 4 5 10 13 4 4 15 13 3 ...
##  $ tx      : Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
##  $ group    : Factor w/ 2 levels "\"group A\"",..: 2 2 1 2 2 1 1 1 2 1 ...
##  $ complica : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 2 1 ...
##  $ hba1c    : num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
##  $ fbs      : num  6.9 4.8 8 3.6 12.5 8.5 NA NA NA NA ...
##  $ rbs      : num  16.7 7.4 13.2 7.4 NA 7.8 9.4 7.8 NA 12.4 ...
##  $ tg_total : num  0.92 1.66 0.74 0.94 3.01 1.3 NA 1.9 NA NA ...
##  $ choleste : num  7.09 2.91 5.94 3.27 7.1 3.54 NA 5.7 NA NA ...
##  $ ADDQSCORE: num  0 -0.222 -0.333 -0.36 -0.44 ...
##  - attr(*, "datalabel")= chr ""
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%10.0g" "%10.0g" "%10.0g" "%10.0g" ...
##  - attr(*, "types")= int  255 255 255 255 255 255 255 255 255 255 ...
##  - attr(*, "val.labels")= chr  "" "sex" "" "" ...
##  - attr(*, "var.labels")= chr  "id_no" "sex" "" "" ...
##  - attr(*, "version")= int 8
##  - attr(*, "label.table")=List of 4
##   ..$ sex     : Named int  0 1
##   .. ..- attr(*, "names")= chr  "female" "male"
##   ..$ tx      : Named int  1 2 3 4
##   .. ..- attr(*, "names")= chr  "diet only" "OHA and diet only" "insulin and diet only" "all"
##   ..$ group   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "\"group A\"" "\"group B\""
##   ..$ complica: Named int  0 1
##   .. ..- attr(*, "names")= chr  "no" "yes"
```

```
summary(data_qol5$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   32.00   47.00   53.00   52.91   59.00   80.00
```

6.5.2 Select rows and columns together

```
data_qol6<-subset(data_qol,age>30 & sex=='male', select = c(id, sex, age, group))
str(data_qol6)
```

```
## 'data.frame':    211 obs. of  4 variables:
##  $ id   : num  335 22 332 147 247 185 331 323 314 305 ...
##  $ sex  : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ age  : num  41 57 60 45 59 48 58 69 65 73 ...
##  $ group: Factor w/ 2 levels "\"group A\"",..: 2 1 2 1 1 1 2 2 2 2 ...
```

```
table(data_qol6$sex)
```

```
##
## female   male
##      0    211
```

6.5.3 Generate a new variable

```
data_qol$age_cat<-data_qol$age
View(data_qol)
```

6.5.4 Categorize into new variables

6.5.4.1 From a numerical variable

```
data_qol$age_cat<-cut(data_qol$age_cat,
                      breaks=c(min(data_qol$age),40,60,Inf),
                      labels=c('min-39','40-59','60-above'))
min(data_qol$age)
```

```
## [1] 21
```

```
table(data_qol$age_cat)
```

```
##
##   min-39    40-59 60-above
##       32      259       73
```

```
str(data_qol$age_cat)
```

```
##  Factor w/ 3 levels "min-39","40-59",..: 2 2 2 2 3 2 2 2 2 2 ...
```

6.5.4.2 From a categorical variable

```
table(data_qol$tx)
```

```
##
##             diet only     OHA and diet only insulin and diet only
##                    10                   238                    26
##                   all
##                    91
```

```
str(data_qol$tx)
```

```
##  Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
```

Create a variable with 'Diet only' vs 'Diet+Drug'. This is a little bit complicated

```
data_qol$tx2<-data_qol$tx
str(data_qol$tx2)
```

```
##  Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
```

```
str(data_qol$tx)
```

```
##  Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
```

```
table(data_qol$tx2)
```

```
##
##             diet only     OHA and diet only insulin and diet only
##                    10                   238                    26
```

```
##                   all
##                    91
```

```
library(plyr)
data_qol$tx2<-revalue(data_qol$tx,c('diet only'='diet', 'OHA and diet only'='med',
                                    'insulin and diet only'='med', 'all'='med'))
table(data_qol$tx2)
```

```
##
## diet  med
##   10  355
```

6.5.5 Dealing with missing data

```
data_qol$tx3<-data_qol$tx
str(data_qol$tx3)
```

```
##  Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
```

```
str(data_qol$tx)
```

```
##  Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
```

```
table(data_qol$tx3)
```

```
##
##             diet only     OHA and diet only insulin and diet only
##                    10                   238                    26
##                   all
##                    91
```

6.5.5.1 Replace values with 'NA'

```
data_qol$tx3<-revalue(data_qol$tx,c('diet only'=NA))
table(data_qol$tx3)
```

```
##
##     OHA and diet only insulin and diet only                   all
##                   238                    26                    91
```

```
str(data_qol$tx3)
```

```
##  Factor w/ 3 levels "OHA and diet only",..: 2 3 1 3 3 1 1 1 3 1 ...
```

<<<<<<< HEAD ## Additional packages ======= ## Additional package >>>>>>> b1e7fe217da24c9c49c585f8cb7b29c3a03e88ce

6.5.6 Package 'dplyr'

'dplyr' package is a very useful package that encourage users to use proper verb when manipulating variables (columns) and observations (rows)

It has 9 useful functions 1. filter() 2. arrange() 3. select() 4. distinct() 5. mutate() and transmute() 6. summarise() 7. sample_n() and sample_frac()

Package ‘dplyr’ is very useful when it is combined with another function that is ‘group_by’

‘

Chapter 7

Exploratory data analysis

7.1 Prepare folder and data

7.1.1 Set the working directory

7.2 Prepare folder and data

7.3 Set the working directory

This can be done in 2 ways:

1. Using codes
2. Using point and click

To use point and click, use the down arrow button next to *More* . Then click 'Set as working directory'

7.3.1 List the files inside the working directory

All files will be displayed when you click 'Files'.

Or you can use this code,

```
list.files()
```

```
##  [1] "_book"
##  [2] "_bookdown.yml"
##  [3] "_bookdown_files"
##  [4] "_output.yml"
##  [5] "01-intro.Rmd"
##  [6] "02-text.Rmd"
##  [7] "03-graphical.Rmd"
##  [8] "04-report.Rmd"
##  [9] "05-Grammar_of_Var.Rmd"
## [10] "06-EDA_Graphs.Rmd"
## [11] "07-preparing-R.Rmd"
## [12] "08-reading-statistical-data-in-R.Rmd"
## [13] "09-glm.Rmd"
```

```
## [14] "10-summary.Rmd"
## [15] "11-references.Rmd"
## [16] "book.bib"
## [17] "cholest.csv"
## [18] "cholest.dta"
## [19] "cholest.sav"
## [20] "cholest.xlsx"
## [21] "directory.jpg"
## [22] "export_csv.csv"
## [23] "export_stata.dta"
## [24] "eye.csv"
## [25] "eye.dta"
## [26] "eye.sav"
## [27] "eye.xlsx"
## [28] "index.Rmd"
## [29] "metab1.csv"
## [30] "metab1.dta"
## [31] "myfolder.png"
## [32] "openr.png"
## [33] "packages.bib"
## [34] "packages.jpg"
## [35] "panes.jpg"
## [36] "preamble.tex"
## [37] "qol.csv"
## [38] "qol.dta"
## [39] "qol.sav"
## [40] "qol.xlsx"
## [41] "R book KIM and Arifin.Rmd"
## [42] "R.png"
## [43] "R_book.Rproj"
## [44] "R_book_KIM_and_Arifin.Rmd"
## [45] "R_book_KIM_and_Arifin_files"
## [46] "README.md"
## [47] "rstudio.png"
## [48] "site"
## [49] "style.css"
## [50] "Template_R_bookdown"
## [51] "toc.css"
```

7.3.2 Reading dataset from SPSS file (.sav)

Dataset in SPSS format will end with .sav. To read SPSS data into R we use 'foreign' library.

Create a object to represent the SPSS data that we will read into R.

```
library(foreign)
dataSPSS<-read.spss('qol.sav', to.data.frame = TRUE)
```

```
## re-encoding from UTF-8
```

7.4 Describing data

Let us examine the data

```
str(dataSPSS)
```

```
## 'data.frame':    365 obs. of  13 variables:
##  $ id       : num  308 335 94 329 350 22 171 274 332 147 ...
##  $ sex      : Factor w/ 2 levels "female","male": 1 2 1 1 1 2 1 1 2 2 ...
##  $ age      : num  55 41 50 47 67 57 60 54 60 45 ...
##  $ tahundx  : num  14 4 5 10 13 4 4 15 13 3 ...
##  $ tx      : Factor w/ 4 levels "diet only","OHA and diet only",..: 3 4 2 4 4 2 2 2 4 2 ...
##  $ group    : Factor w/ 2 levels "\"group A\"",..: 2 2 1 2 2 1 1 1 2 1 ...
##  $ complica : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 2 1 ...
##  $ hba1c    : num  8.1 8 7.5 9.4 11.7 8.1 7.5 9.2 NA NA ...
##  $ fbs      : num  6.9 4.8 8 3.6 12.5 8.5 NA NA NA NA ...
##  $ rbs      : num  16.7 7.4 13.2 7.4 NA 7.8 9.4 7.8 NA 12.4 ...
##  $ tg_total : num  0.92 1.66 0.74 0.94 3.01 1.3 NA 1.9 NA NA ...
##  $ choleste : num  7.09 2.91 5.94 3.27 7.1 3.54 NA 5.7 NA NA ...
##  $ ADDQSCORE: num  0 -0.222 -0.333 -0.36 -0.44 ...
##  - attr(*, "variable.labels")= Named chr  "id_no" "sex" "" "" ...
##   ..- attr(*, "names")= chr  "id" "sex" "age" "tahundx" ...
##  - attr(*, "codepage")= int 65001
```

Now, let us summarize our data

```
summary(dataSPSS)
```

```
##        id            sex           age           tahundx
##  Min.   :  1.0   female:153   Min.   :21.00   Min.   : 1.000
##  1st Qu.:126.0   male  :212   1st Qu.:47.00   1st Qu.: 4.000
##  Median :227.0                Median :53.00   Median : 7.000
##  Mean   :221.5                Mean   :52.75   Mean   : 8.795
##  3rd Qu.:325.0                3rd Qu.:59.00   3rd Qu.:12.000
##  Max.   :416.0                Max.   :80.00   Max.   :38.000
##
##                       tx            group     complica      hba1c
##  diet only             : 10   "group A":248   no :225   Min.   : 4.100
##  OHA and diet only     :238   "group B":117   yes:140   1st Qu.: 7.500
##  insulin and diet only: 26                              Median : 9.050
##  all                   : 91                             Mean   : 9.301
##                                                         3rd Qu.:10.775
##                                                         Max.   :19.900
##                                                         NA's   :111
##       fbs              rbs            tg_total        choleste
##  Min.   : 2.700   Min.   : 3.900   Min.   :0.380   Min.   : 2.020
##  1st Qu.: 5.700   1st Qu.: 7.925   1st Qu.:1.125   1st Qu.: 4.308
##  Median : 8.000   Median :11.300   Median :1.570   Median : 5.210
##  Mean   : 9.003   Mean   :12.045   Mean   :2.002   Mean   : 5.437
##  3rd Qu.:11.900   3rd Qu.:15.000   3rd Qu.:2.385   3rd Qu.: 6.423
##  Max.   :29.200   Max.   :31.500   Max.   :8.020   Max.   :13.100
##  NA's   :178      NA's   :83       NA's   :191     NA's   :181
##    ADDQSCORE
##  Min.   :-9.000
##  1st Qu.:-5.590
##  Median :-3.944
##  Mean   :-4.179
##  3rd Qu.:-2.556
##  Max.   : 0.000
```

##

7.5 Graphing or Plotting data

You must ask yourselves these: 1. Which variable do you want to plot? 2. What is the type of that variable? Factor? Numerical? 3. Are you going to plot another variable together?

7.5.1 One variable: A categorical or factor variable

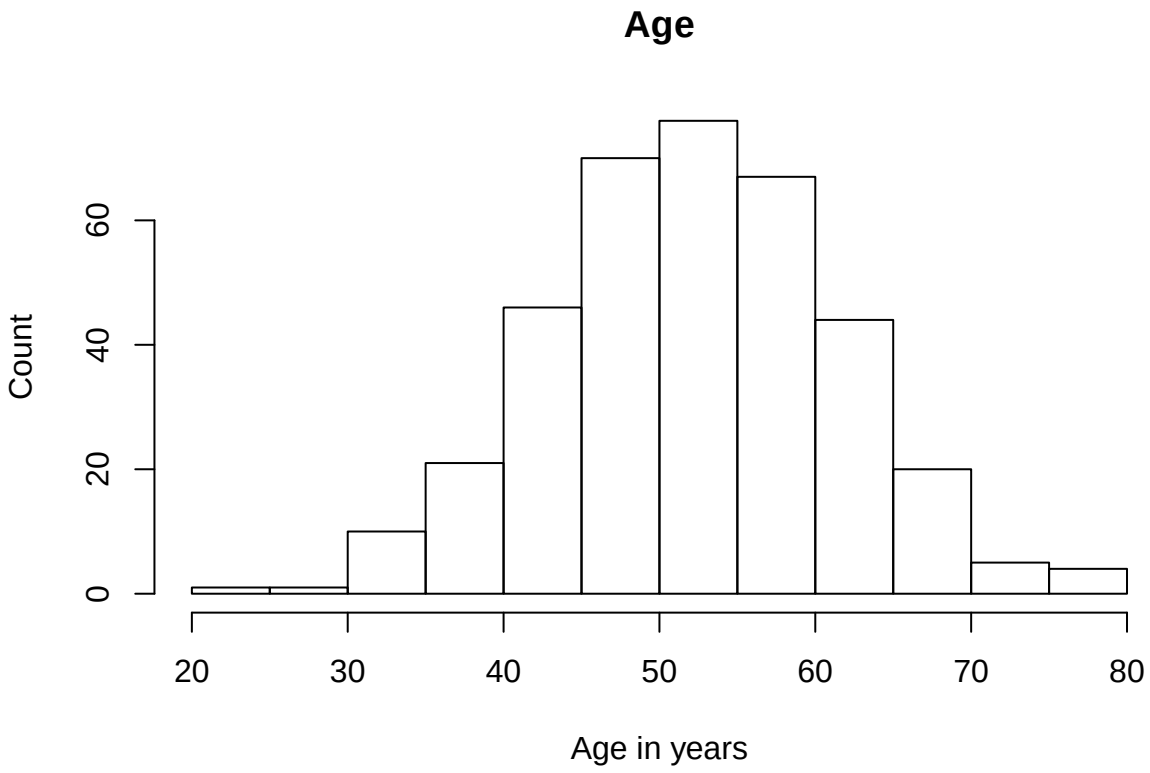
We can create a simple barchart

```
dist.sex<-table(dataSPSS$sex)
barplot(dist.sex,
        main='Sex distribution',
        xlab='Sex')
```

7.5.2 One variable: A numerical variable

histogram

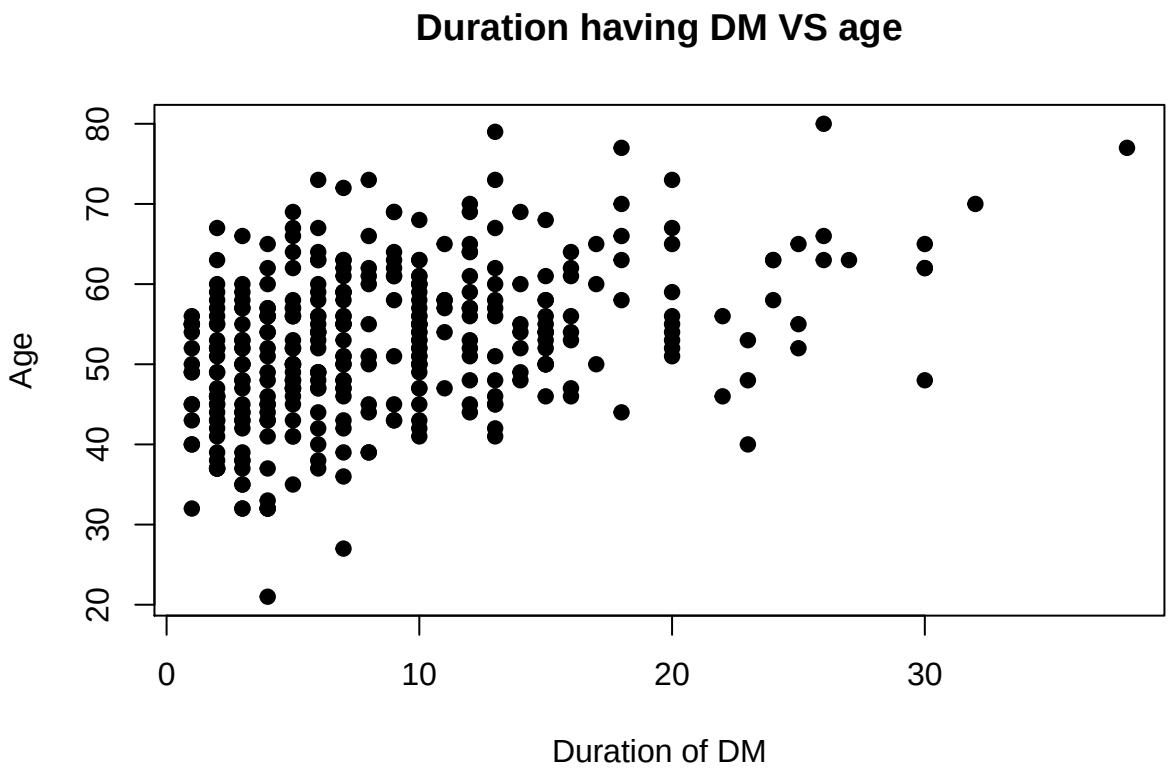
```
hist(dataSPSS$age, main = 'Age',
     xlab='Age in years',
     ylab='Count')
```



7.5.3 Two variables : A numerical with another numerical variable

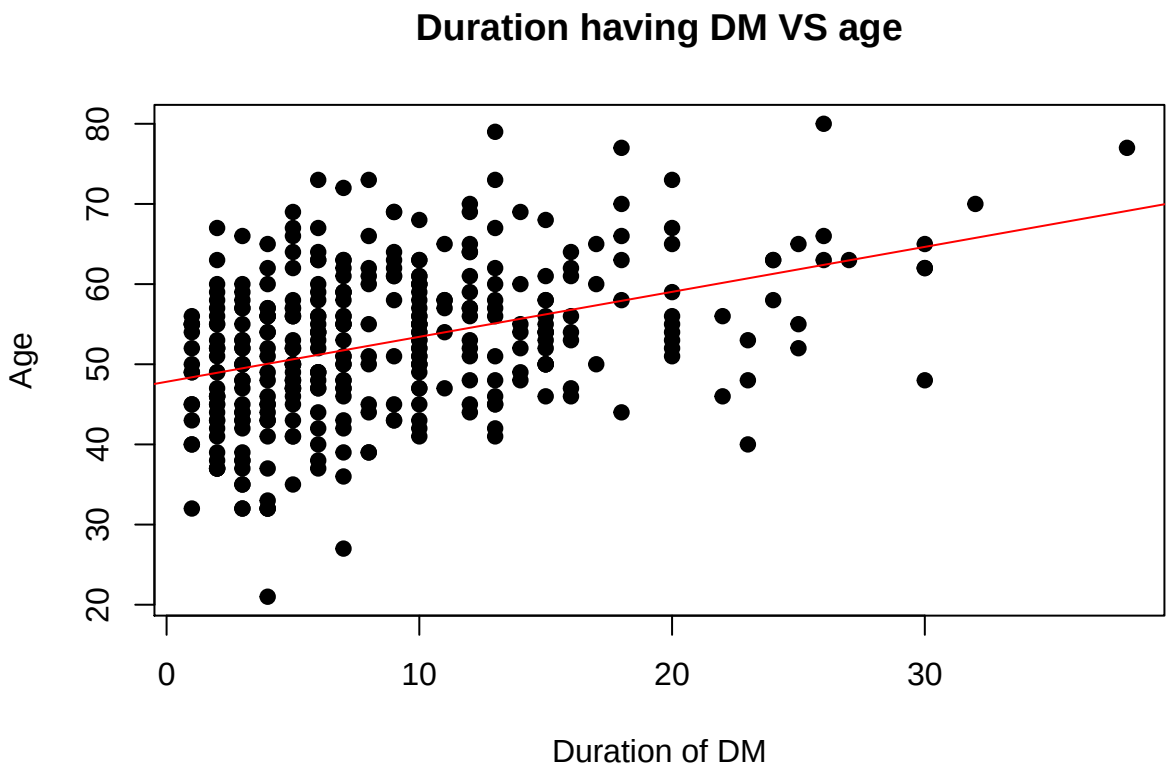
We will use *scatterplot* to plot

```
plot(dataSPSS$tahundx, dataSPSS$age,
     main = 'Duration having DM VS age',
     xlab = 'Duration of DM', ylab = 'Age',
     pch = 19)
```



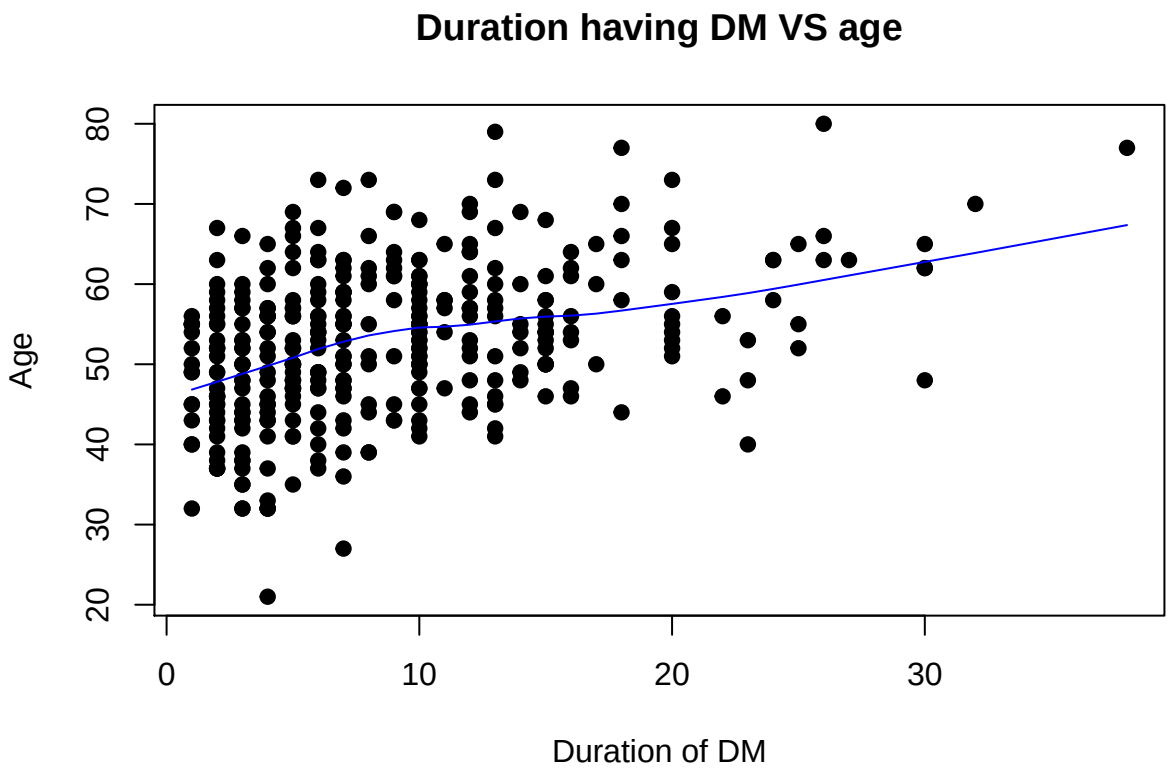
Let us make a fit line

```
plot(dataSPSS$tahundx, dataSPSS$age,
     main = 'Duration having DM VS age',
     xlab = 'Duration of DM', ylab = 'Age',
     pch = 19)
abline(lm(dataSPSS$age~dataSPSS$tahundx), col = 'red')
```



and a lowess

```
plot(dataSPSS$tahundx, dataSPSS$age,
     main = 'Duration having DM VS age',
     xlab = 'Duration of DM', ylab = 'Age',
     pch = 19)
lines(lowess(dataSPSS$tahundx,dataSPSS$age), col = 'blue')
```



7.5.4 Two variables : A categorical variable with a categorical variable

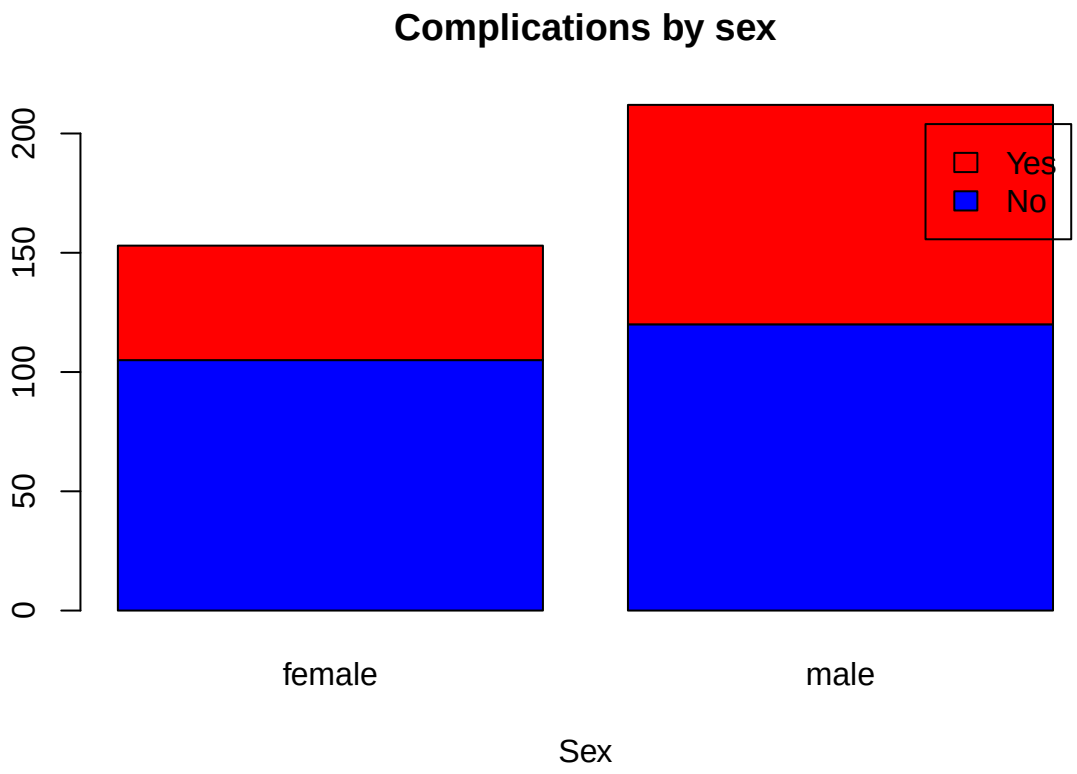
Now, we will plot 2 categorical variables simultenously.

First, we will use stacked barchart

```
compl.sex<-table(dataSPSS$complica,dataSPSS$sex)
compl.sex
```

```
##
##       female male
##   no     105  120
##   yes     48   92
```

```
barplot(compl.sex,
        main='Complications by sex',
        xlab='Sex',
        col=c('blue','red'),
        legend=c('No','Yes'))
```

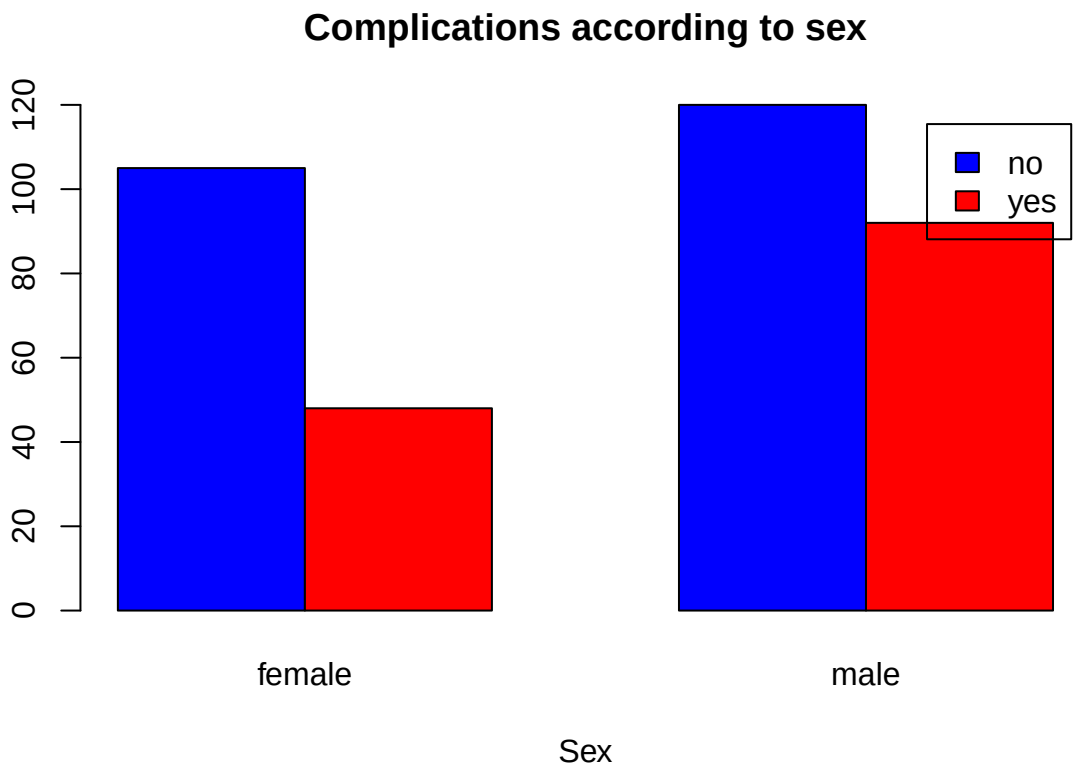


Next, we will use grouped barchart

```
compl.sex
```

```
##
##       female male
##   no     105  120
##   yes     48   92
```

```
barplot(compl.sex,
        main = 'Complications according to sex',
        xlab = 'Sex',
        col = c('blue','red'),
        legend = c('no','yes'),
        beside = TRUE)
```


Complications according to sex
0
20
40
60
80
100
120
no
yes
female
male
Sex

Chapter 8

Preparing R

8.1 Objectives

The objectives of this lecture are:

1. To ensure that the installation of R is correct
2. To ensure that the installation of RStudio is correct
3. To be able to install R packages
4. To be able to create a working directory

8.1.1 Installation of R

- The latest version of R is R version 3.4.1 (2017-06-30), Single Candle.
- R can be installed inside Linux, Mac OS and Windows (of course)
- The installation files (tar.gz, binaries) can be downloaded from https://cran.r-project.org/
- Users can install different versions of R in a same machine or computer
- There is no need to uninstall if you want to upgrade currently installed R

8.1.1.1 Starting R

Double click on R icon and you should get this

8.1.2 Installation of RStudio

First, make sure you have RStudio successfully installed.

8.1.2.1 Starting RStudio

You can double click on RStudio icon and you will see this:

8.1.2.2 Why RStudio?

- Working with R GNU is alright.
- But for many people, they prefer to communicate with R using a GUI
- RStudio is a popular GUI

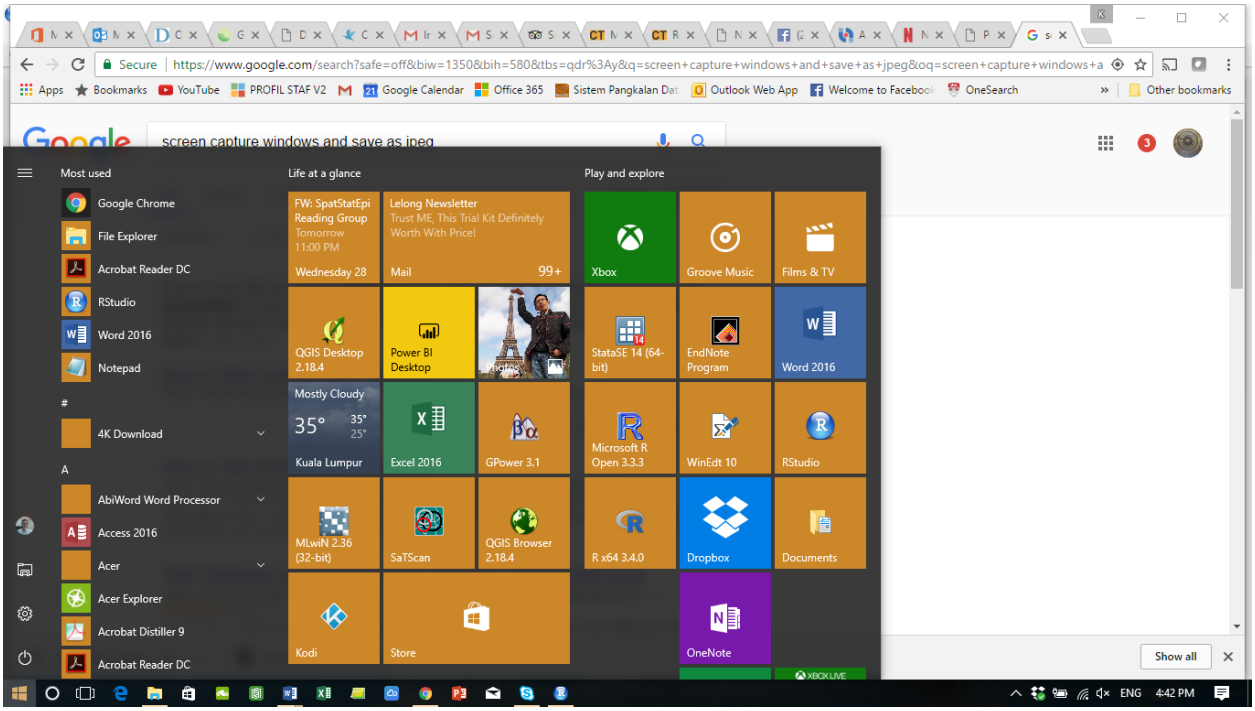


Figure 8.1: Succesful R installation

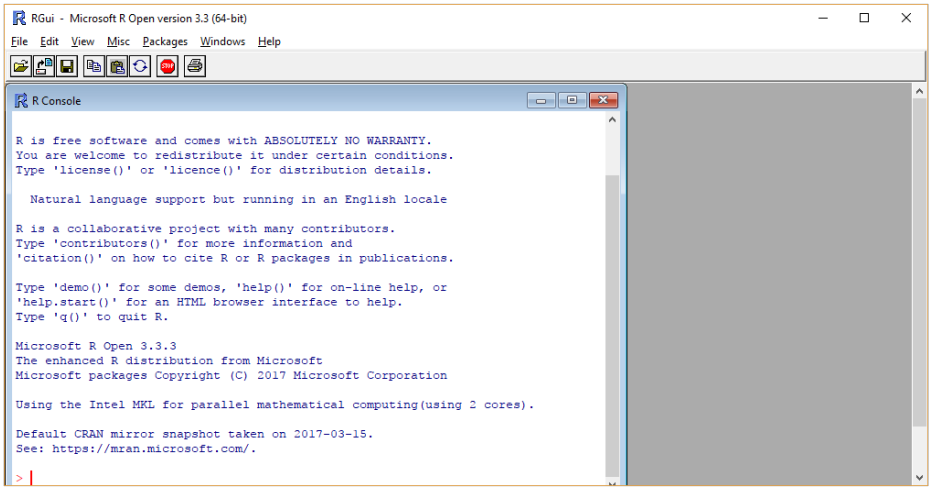


Figure 8.2: R in action

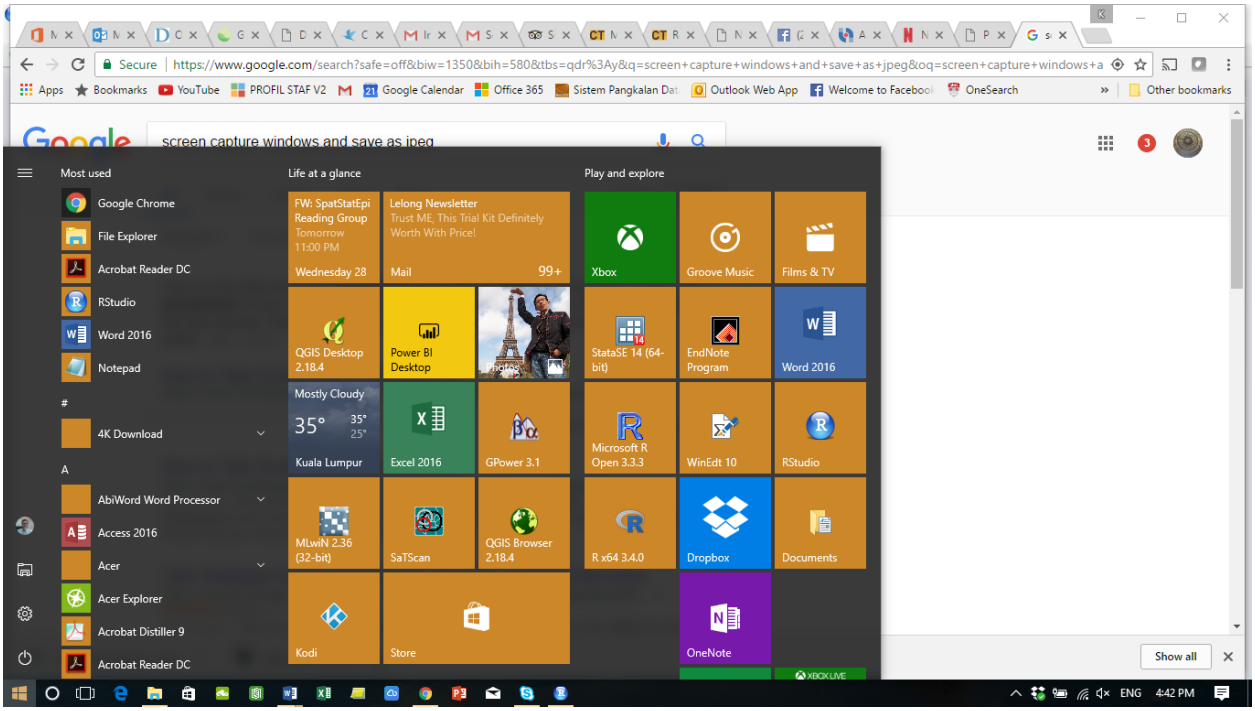


Figure 8.3: Succesful RStudio installation

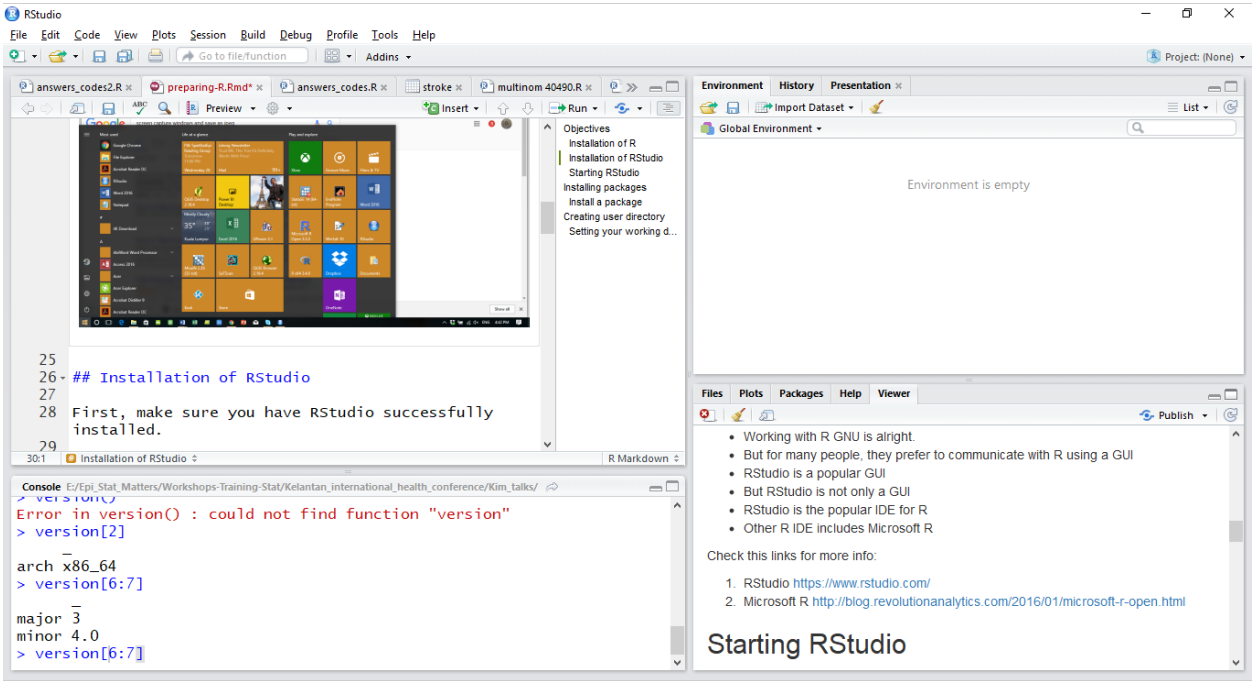


Figure 8.4: RStudio sucessfully open

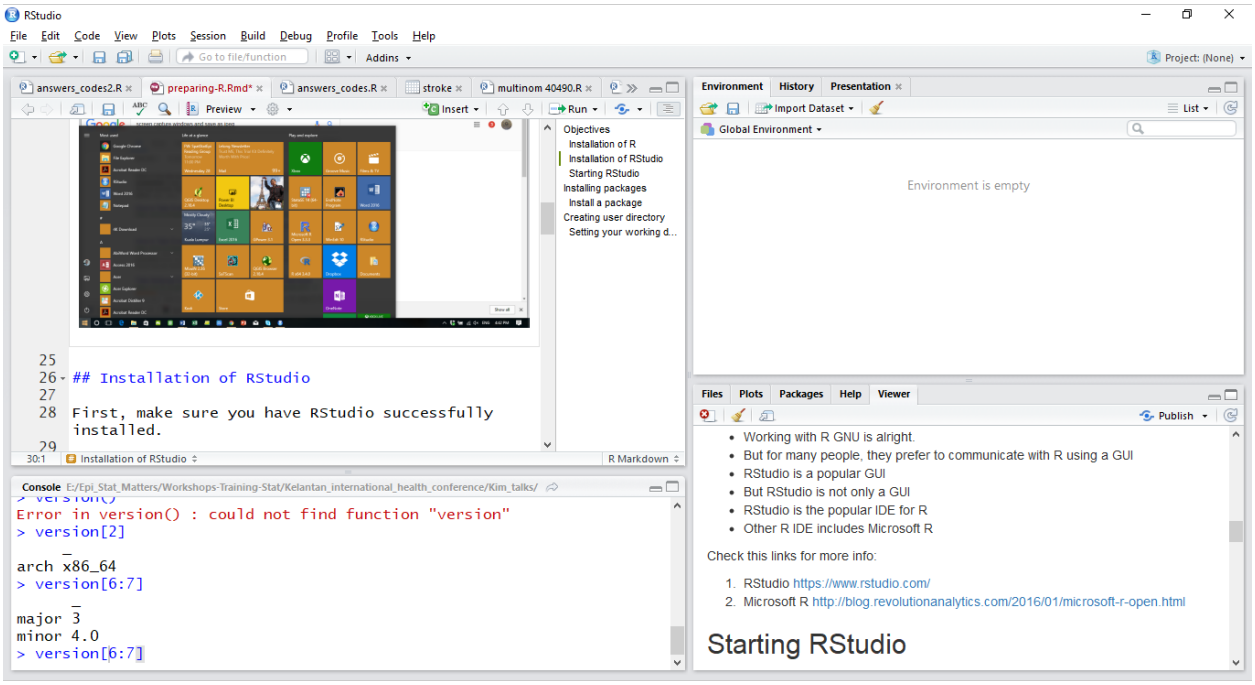


Figure 8.5: RStudio sucessfully open

- But RStudio is not only a GUI
- RStudio is the popular IDE for R
- Other R IDE includes Microsoft R

Check this links for more info:

1. RStudio https://www.rstudio.com/
2. Microsoft R http://blog.revolutionanalytics.com/2016/01/microsoft-r-open.html

8.1.2.3 RStudio interface

- Depending on your OS, you may start RStudio differently.
- Here we assume you are working with Windows OS
- You should be able to see 4 panes in the layout.

8.1.2.4 RStudio panes

You should see that - the lower left pane tells you about your R information (the console pane) - the upper left pane is to show files that are open - the upper right for the 'Environment, History and Presentation' pane - the lower right pane is for to list file names, show plots, show packages, display help document and view outputs (such as html file)

8.2 Installing packages

R uses packages to perform its tasks.

There are two common packages:

1. `base` packages
2. `user-contributed` packages

- The base packages come with the installation of R
- The base package provides basic but adequate functions to perform many standard data management, visualization and analysis.
- However, user needs to install user-contributed packages if they need to perform functions (tasks) not available in the base package
- User-contributed packages allow users to perform more advanced and more complicated functions
- There are more than 10200 packages as of March 2017

For a complete list of packages, see https://cran.r-project.org/web/packages/

8.2.1 Package installation

You can install user-contributed packages through:

1. internet (to cran)
2. Github packages
3. local zip files

In this session, we will learn to install a few small packages. I have installed them. For those who have not,

1. put your cursor in the CONSOLE pane
2. type the codes below

```
install.packages(foreign)
install.packages(haven)
```

3. click ENTER

8.3 Workflow

We propose that you always have these steps as your workflow when working with R:

1. Set working directory
2. Read data
3. Explore + Clean data
4. Build Model
5. Check Model

8.4 Working directory

This is a good practice.

User must create or specify the working directory to work with R.

The working directory:

1. stores all the outputs such as the plots, html files, pdf files
2. contains your data

Creating a working directory is a simple BUT an important step.

Unfortunately, many users do not pay attention to this and forget to set it. So, pay attention so you will not get lost.

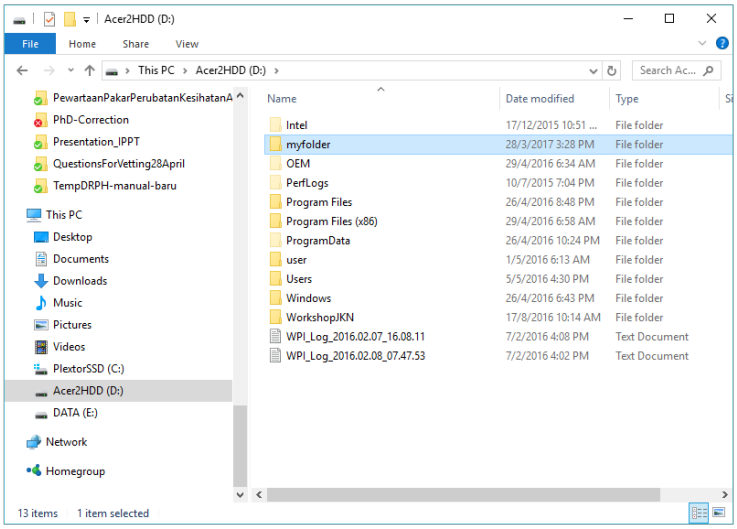


Figure 8.6: myfolder

8.4.1 Creating your working directory

Follow this steps:

1. Make a folder in D directory D:

2. Name it as *myfolder*

You will see this:

8.4.2 Setting your working directory

To set your working directory:

1. Go back to RStudio pane
2. In the FILE pane, click the *three small dots*
3. Navigate to *myfolder*
4. Click *More*
5. Click *Set as working directory*

- or simply use *setwd* to do so.

```
setwd('D:/myfolder')
```

- type *getwd* to confirm

```
getwd()
```

```
## [1] "E:/R_book"
```

Chapter 9

Reading statistical data

9.1 Objectives

At the end of the lecture, participants are:

1. able to read various data formats into R
2. able to export data from R into various data format
3. able to create R-markdown document

9.2 Data formats

R can read datasets from different formats.

The `base` package enables us to read `.txt` and `.csv` files.

You can also use the point and click method to read data. Go to File -> Import Dataset -> From …

To read datasets from SPSS (`.sav`), Stata (`.dta`), Excel (`xlsx`) and SAS, we need to load special packages.

There are more than 1 packages you can use to read and write data from/to different spreadsheet or statistical software.

The packages include:

1. `haven`
2. `foreign`
3. `readxl`
4. `readr`

9.3 Reading data into R

9.3.1 Reading csv file

Let us read a `.csv` files names as `metab.csv`.

We would like to create an object named as `met_data` to represent the read `metab.csv`. We can do this:

```
met_data <- read.csv('metab1.csv', header = TRUE)
```

header = TRUE means that you will read the first row as the variable names.

After you have done this, you will see a new object named as `met_data` inside the *environment* pane.

9.3.2 Reading dataset from MS Excel file

You can read dataset from Excel using 2 methods

1. point and click method. File - Import Dataset - from Excel
2. `readxl` package

To read data using specific packages such as `readxl` package:

1. you need to install the library first using `install.packages()`
2. After that, load the library using `library(readxl)`
3. Type `read_excel()` with relevant arguments to read the data into RStudio

```
library(readxl)
dataexcel <- read_excel('eye.xlsx', sheet = 1)
```

The example above means that we read an MS Excel file named as `eye.xlsx` and named it as `dataexcel`.

9.3.3 Reading dataset from SPSS file (.sav)

Dataset in SPSS format will end with `.sav`.

To read SPSS data into R we may use `foreign` or `haven` packages.

After reading the `.sav` file, an object will be created based on what we named it.

The example below shows that an object named as `dataSPSS`represents the SPSS data `cholest.sav`that we just read into RStudio.

```
library(foreign)
dataSPSS <- read.spss('cholest.sav', to.data.frame = TRUE)
```

```
## re-encoding from UTF-8
```

9.3.4 Reading dataset from Stata (.dta)

Dataset in Stata format will end with `.dta`.

To read Stata data into R we may use `foreign` or `haven` packages.

After reading the `.dta` file, similarly an object will be created - in this case, we named it as `dataSTATA`.

`dataSTATA` is an object that represents the Stata data (now in the memory) that we just read `metab1.dta`into RStudio.

```
datastata <- read.dta('metab1.dta', convert.factors = TRUE)
```

9.3.5 Alternative methods or package

1. You can go to File - Read Datasets
2. You may use **haven** package to read SAS, SPSS and Stata file.

```
library(haven)
dataSPSS2 <- read_sav('cholest.sav')
dataSTATA2 <- read_dta('cholest.dta')
```

9.3.6 Other data format

Other important data formats that might be useful in epidemiology and statistics are:

1. shapefile `.shp`
2. text file `.txt`
3. text file `.dat`
4. `XML` file
5. images file `DICOM`

We will not cover these today.

9.4 Exporting data from R

You can also export data into various formats using similar packages.

For example,

1. to export data into a *comma separated version* (.csv) file, we can use `write.csv` function.
2. to export data into stata format, we can use `write.dta` function

```
export_csv <- write.csv(dataSPSS, 'export_csv.csv')
export_stata <- write.dta(dataexcel, 'export_stata.dta')
```

Chapter 10

GLM

10.0.1 Objectives

At the end of the lecture, participants are

1. able to perform linear regression
2. able to perform logistic regression
3. able to perform Cox proportional hazard regression

10.1 Set working directory

Set your working directory.

This is a folder that contains your dataset and objects created by R.

10.2 Read data

We will read a stata data into R. This file will read using **foreign** package.

We will name the object as **data1** as an object that represent the dataset.

This object is a `data.frame` object

The **data1** object will remain in the memory unless you close your RStudio.

```
library(foreign)
data1 <- read.dta('metab1.dta', convert.factors = TRUE)
head(data1)
```

```
##   id2 age sex race marital dm  bmi2 waist hip hba1c  fbs totchol   msbp
## 1   1  42   2    1       1  0 21.31    72  89   5.0 5.41    4.80 113.25
## 2   2  63   2    1       3  1 36.00   125  95   7.2 8.39    8.09 209.00
## 3   3  54   2    1       1  0 37.17   100 118   5.6 5.46    4.32 160.00
## 4   4  46   1    1       1  0 27.34    89  97   5.3 5.82    5.27 134.25
## 5   5  40   2    1       1  0 26.94    79  99   4.8 4.67    6.38 100.00
## 6   6  43   1    1       1  0 28.82   101 112   5.1 4.82    7.48 159.00
##     mdbp gender crural racecat       whr
## 1  73.50 female      0       0 0.8089887
## 2  94.00 female      1       0 1.3157895
```

```
## 3  76.75 female      0       0 0.8474576
## 4  87.75   male      1       0 0.9175258
## 5  65.00 female      1       0 0.7979798
## 6 104.00   male      0       0 0.9017857
```

We use `head()` function to list the first 6 observations in the dataset.

10.3 Explore and clean data

Next we will describe the data and visualize

1. Descriptive
2. Visualization

10.3.1 Descriptive analysis

We will check basic descriptive statistics from our data

```
library(psych)
describe(data1)
```

```
##         vars   n   mean     sd median trimmed    mad   min    max  range
## id2        1 500 250.50 144.48 250.50  250.50 185.32  1.00 500.00 499.00
## age        2 500  49.04  14.09  48.50   49.06  14.08 18.00  84.00  66.00
## sex        3 500   1.58   0.49   2.00    1.60   0.00  1.00   2.00   1.00
## race       4 500   1.46   0.99   1.00    1.21   0.00  1.00   5.00   4.00
## marital    5 498   1.29   0.65   1.00    1.12   0.00  1.00   3.00   2.00
## dm         6 500   0.11   0.31   0.00    0.01   0.00  0.00   1.00   1.00
## bmi2       7 500  26.28   5.33  25.71   25.95   4.81 14.30  56.08  41.78
## waist      8 500  86.40  12.79  86.00   86.08  11.86 52.00 154.50 102.50
## hip        9 500  98.54  10.75  98.00   98.31  10.38 62.00 153.50  91.50
## hba1c     10 496   5.86   1.47   5.40    5.56   0.44  0.20  13.20  13.00
## fbs       11 486   5.94   2.30   5.39    5.54   1.14  2.59  21.11  18.52
## totchol   12 496   5.83   1.29   5.80    5.76   1.20  2.47  12.29   9.82
## msbp      13 500 134.72  22.77 132.00  132.92  21.68 82.50 225.00 142.50
## mdbp      14 500  80.03  11.51  79.50   79.86  11.49 45.00 120.00  75.00
## gender*   15 500   1.42   0.49   1.00    1.40   0.00  1.00   2.00   1.00
## crural    16 500   0.51   0.50   1.00    0.51   0.00  0.00   1.00   1.00
## racecat   17 500   0.45   0.94   0.00    0.21   0.00  0.00   3.00   3.00
## whr       18 500   0.88   0.11   0.86    0.87   0.08  0.65   1.42   0.77
##           skew kurtosis   se
## id2       0.00    -1.21 6.46
## age       0.01    -0.46 0.63
## sex      -0.31    -1.90 0.02
## race      2.00     2.77 0.04
## marital   1.98     2.28 0.03
## dm        2.52     4.35 0.01
## bmi2      0.97     2.50 0.24
## waist     0.61     2.03 0.57
## hip       0.46     2.21 0.48
## hba1c     2.49     7.80 0.07
## fbs       2.88    11.33 0.10
## totchol   0.73     1.74 0.06
```

```
## msbp     0.80     0.68 1.02
## mdbp     0.18    -0.05 0.51
## gender*  0.31    -1.90 0.02
## crural  -0.03    -2.00 0.02
## racecat  1.85     1.87 0.04
## whr      1.72     5.09 0.00
```

10.3.2 Visualization

We do not have time to cover for data vizualition.

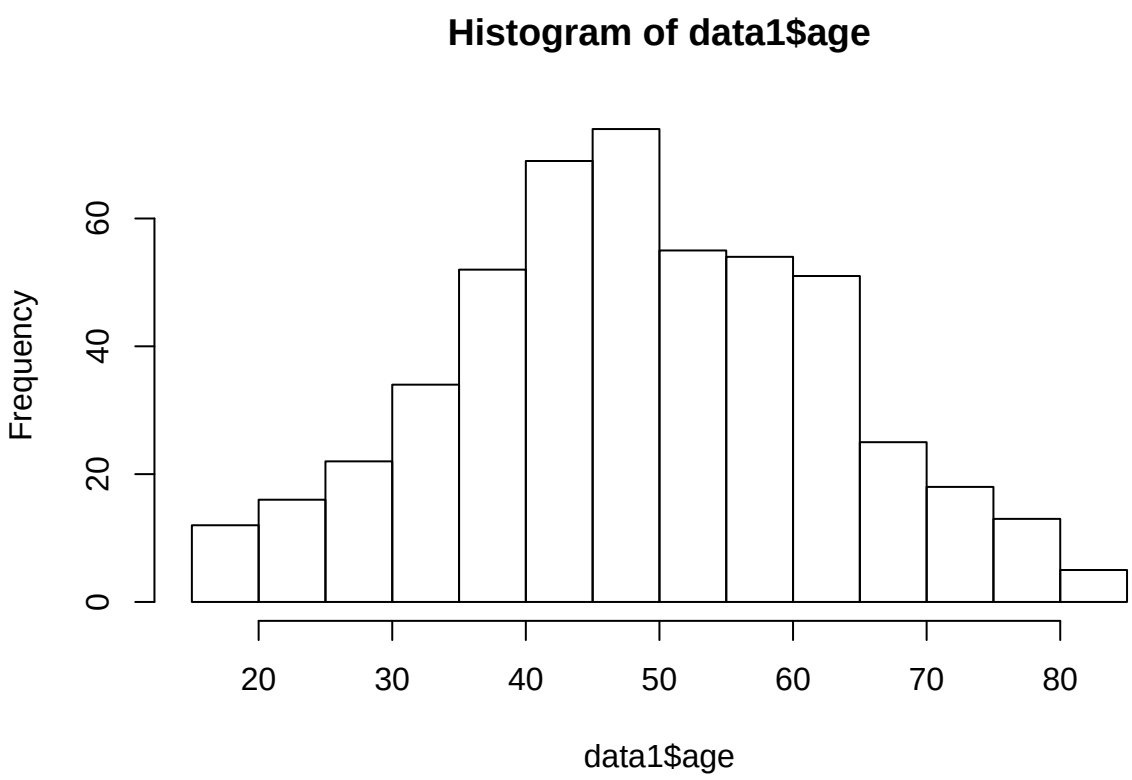
But for here we would do

1. histogram
2. bar charts
3. box-plots
4. scatterplots

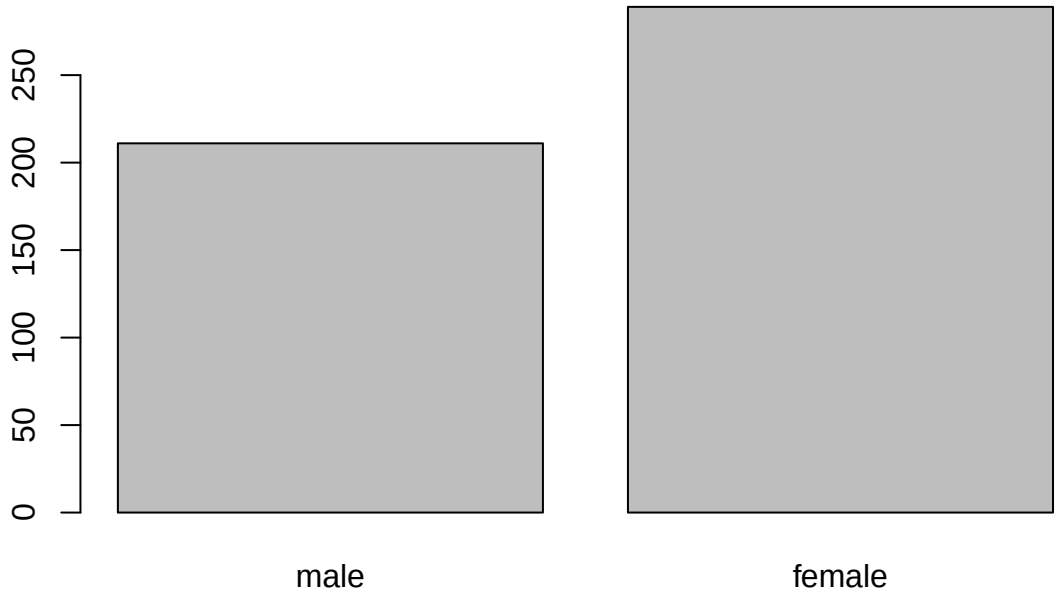
To examine the distribution of our data.

Briefly:

```
hist(data1$age)
```




```
cts_sex <- table(data1$sex)
barplot(cts_sex, names.arg = c('male','female'))
```




```
cor(data1[,c(2,7:14 )], use = 'complete.obs')
```

```
##                  age       bmi2      waist          hip     hba1c
## age     1.0000000000 0.01333764 0.20366927 0.0007456519 0.1637614
## bmi2    0.0133376398 1.00000000 0.76631090 0.8333520043 0.1894581
## waist   0.2036692689 0.76631090 1.00000000 0.6481527376 0.2607116
## hip     0.0007456519 0.83335200 0.64815274 1.0000000000 0.1393534
## hba1c   0.1637613510 0.18945815 0.26071159 0.1393533901 1.0000000
## fbs     0.0744876967 0.14530370 0.16664931 0.1081298717 0.6201932
## totchol 0.1680643316 0.03611592 0.07409306 0.0314817073 0.1919677
## msbp    0.4973145998 0.29310141 0.34845937 0.2088822226 0.1012963
## mdbp    0.2395562888 0.39672237 0.38554987 0.3421091043 0.1149683
##                fbs    totchol      msbp       mdbp
## age     0.0744877 0.16806433 0.4973146 0.23955629
## bmi2    0.1453037 0.03611592 0.2931014 0.39672237
## waist   0.1666493 0.07409306 0.3484594 0.38554987
## hip     0.1081299 0.03148171 0.2088822 0.34210910
## hba1c   0.6201932 0.19196773 0.1012963 0.11496827
## fbs     1.0000000 0.23526402 0.0976063 0.11685121
## totchol 0.2352640 1.00000000 0.1286784 0.09181019
## msbp    0.0976063 0.12867845 1.0000000 0.69994756
## mdbp    0.1168512 0.09181019 0.6999476 1.00000000
```

10.4 Linear regression

We perform linear regression when we assume that distribution of the outcome variables is normally distributed as a function of certain covariates (independent variables)

10.4.1 Estimation

To perform the estimation for linear regression, we can use `lm()` function.

Let us model body mass index *bmi* as a function of hip circumference, mean systolic blood pressure *msbp*, mean diastolic blood pressure *mdbp* and *gender*

```
modl <- lm(bmi2 ~ hip + msbp + mdbp
           + gender, data = data1)
summary(modl)
```

```
##
## Call:
## lm(formula = bmi2 ~ hip + msbp + mdbp + gender, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3285 -1.8325 -0.3378  1.6164 15.4678
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.065630   1.314528 -12.982  < 2e-16 ***
## hip           0.388818   0.012708  30.596  < 2e-16 ***
## msbp          0.018379   0.007898   2.327  0.02036 *
## mdbp          0.036431   0.016230   2.245  0.02523 *
## gendermale   -0.846502   0.260951  -3.244  0.00126 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.856 on 495 degrees of freedom
## Multiple R-squared:  0.7149, Adjusted R-squared:  0.7126
## F-statistic: 310.4 on 4 and 495 DF,  p-value: < 2.2e-16
```

From the results, we can see that:

1. 71.5% of variation in the expected *bmi* is explained by the covariates
2. all covariates are siginificantly ($p < 0.05$) predictive of *bmi*

10.4.2 Inference

Now, let us calculate the 95% of the expected mean of bmi

```
confint(modl)
```

```
##                     2.5 %       97.5 %
## (Intercept) -19.648371574 -14.48288755
## hip           0.363849346   0.41378699
## msbp          0.002861941   0.03389611
## mdbp          0.004542767   0.06831958
## gendermale   -1.359210684  -0.33379268
```

10.4.3 Prediction

10.5 Logistic regression

In logistic regression, we model an outcome variables which is assumed to follow binomial distribution as a function of a set of covariates (independent variables).

In R, we use `glm()` function to perform **Generalized Linear Regression** analysis.

But there are other R packages that can do similar analysis. Based on our experience, the `glm()` in the **base** package is good enough.

10.5.1 Estimation

Let us estimate the expected log odds for having diabetes mellitus *dm* as a function os these covariates: body mass index *bmi2* , age , total cholesterol *totchol* and mean systolic blood pressure *msbp*.

We specify the family of *binomial* and use the *logit* link.

```
modlog <- glm(dm ~ bmi2 + age + totchol + msbp, family = binomial(link = logit), data = data1)
summary(modlog)
```

```
##
## Call:
## glm(formula = dm ~ bmi2 + age + totchol + msbp, family = binomial(link = logit),
##     data = data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1768  -0.5281  -0.4016  -0.2914   2.5278
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.285660   1.284589  -4.893 9.92e-07 ***
## bmi2         0.049137   0.027323   1.798 0.072115 .
## age          0.047859   0.013178   3.632 0.000281 ***
## totchol     -0.041156   0.113628  -0.362 0.717205
## msbp         0.004154   0.007351   0.565 0.571993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 341.40  on 495  degrees of freedom
## Residual deviance: 315.92  on 491  degrees of freedom
##   (4 observations deleted due to missingness)
## AIC: 325.92
##
## Number of Fisher Scoring iterations: 5
```

From the results, we estimate that the log odds for diabetes mellitus:

1. increase by 0.049 for each increase in bmi, adjusted for other covariates (p-value 0.072)
2. increase by 0.048 for each increase in age, controlled for other covariates (p-value < 0.001)

3. are not significantly predicted by either the total cholesterol and mean systolic blood pressure (p-value 0.717 and 0.572, respectively)

Odds ratio

To obtain the odds ratios, we use the `exp()` function

We then exponential the coefficients of the model estimated from `glm()` function

```
exp(modlog$coefficients)
```

```
## (Intercept)        bmi2         age     totchol        msbp
## 0.001862827 1.050364000 1.049023206 0.959679727 1.004162959
```

10.5.2 Inference

To estimate the 95% confidence intervals for the log odds (CI for log odds), we use the `confint()` function to all the regression parameters or the β_p

```
confint(modlog)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %      97.5 %
## (Intercept) -8.875644415 -3.82283778
## bmi2        -0.005510115  0.10254631
## age          0.022458291  0.07427335
## totchol     -0.269610946  0.17709212
## msbp        -0.010565995  0.01836516
```

And to estimate the CI for odds ratios, we exponentiate the lower bound and upper bound of the regression parameters β_p

```
exp(confint(modlog))
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %     97.5 %
## (Intercept) 0.0001397515 0.02186566
## bmi2        0.9945050382 1.10798861
## age         1.0227123770 1.07710119
## totchol     0.7636765479 1.19374105
## msbp        0.9894896289 1.01853484
```

10.6 Cox proportional hazard regression

In a cohort study, a time-to-event data is common. Such data has a time from the start of study until one point of time.

This point of time can be either:

1. participant develop an outcome of interest
2. participant do not develop the outcome of interest after the maximum time of follow up

In R, the most common packacge for typical time-to-event data is **survival** package

To perform time-to-event data analysis, we need to load the **survival** library.

We will use a data frame using the built-in lung cancer dataset that ships with the survival package. :

1. inst: Institution code
2. time: Survival time in days
3. status: censoring status 1=censored, 2=dead
4. age: Age in years
5. sex: Male=1 Female=2
6. ph.ecog: ECOG performance score (0=good 5=dead)
7. ph.karno: Karnofsky performance score as rated by physician
8. pat.karno: Karnofsky performance score as rated by patient
9. meal.cal: Calories consumed at meals
10. wt.loss: Weight loss in last six months

Load the library

```
library(survival)
```

Describe data

```
library(psych)
lung <- lung
describe(lung)
```

```
##           vars   n   mean     sd median trimmed    mad min  max range
## inst         1 227  11.09   8.30   11.0   10.36   8.90   1   33    32
## time         2 228 305.23 210.65  255.5  281.09 160.86   5 1022  1017
## status       3 228   1.72   0.45    2.0    1.78   0.00   1    2     1
## age          4 228  62.45   9.07   63.0   62.88   9.64  39   82    43
## sex          5 228   1.39   0.49    1.0    1.37   0.00   1    2     1
## ph.ecog      6 227   0.95   0.72    1.0    0.93   1.48   0    3     3
## ph.karno     7 227  81.94  12.33   80.0   82.73  14.83  50  100    50
## pat.karno    8 225  79.96  14.62   80.0   80.72  14.83  30  100    70
## meal.cal     9 181 928.78 402.17  975.0  914.81 296.52  96 2600  2504
## wt.loss     10 214   9.83  13.14    7.0    8.45  10.38 -24   68    92
##              skew kurtosis    se
## inst         0.66    -0.22  0.55
## time         1.08     0.86 13.95
## status      -0.99    -1.02  0.03
## age         -0.37    -0.40  0.60
## sex          0.43    -1.82  0.03
## ph.ecog      0.14    -0.85  0.05
## ph.karno    -0.57    -0.20  0.82
## pat.karno   -0.60     0.13  0.97
## meal.cal     1.00     3.35 29.89
## wt.loss      1.17     2.33  0.90
```

and declare the time to event

```
tte <- Surv(time = lung$time, event = lung$status==2)
tte
```

```
##   [1]  306   455  1010+  210   883  1022+  310   361   218   166   170
##  [12]  654   728    71   567   144   613   707    61    88   301    81
##  [23]  624   371   394   520   574   118   390    12   473    26   533
##  [34]  107    53   122   814   965+   93   731   460   153   433   145
##  [45]  583    95   303   519   643   765   735   189    53   246   689
##  [56]   65     5   132   687   345   444   223   175    60   163    65
##  [67]  208   821+  428   230   840+  305    11   132   226   426   705
##  [78]  363    11   176   791    95   196+  167   806+  284   641   147
```

```
##  [89]  740+  163   655   239    88   245   588+   30   179   310   477
## [100]  166   559+  450   364   107   177   156   529+   11   429   351
## [111]   15   181   283   201   524    13   212   524   288   363   442
## [122]  199   550    54   558   207    92    60   551+  543+  293   202
## [133]  353   511+  267   511+  371   387   457   337   201   404+  222
## [144]   62   458+  356+  353   163    31   340   229   444+  315+  182
## [155]  156   329   364+  291   179   376+  384+  268   292+  142   413+
## [166]  266+  194   320   181   285   301+  348   197   382+  303+  296+
## [177]  180   186   145   269+  300+  284+  350   272+  292+  332+  285
## [188]  259+  110   286   270    81   131   225+  269   225+  243+  279+
## [199]  276+  135    79    59   240+  202+  235+  105   224+  239   237+
## [210]  173+  252+  221+  185+   92+   13   222+  192+  183   211+  175+
## [221]  197+  203+  116   188+  191+  105+  174+  177+
```

10.6.1 Estimation and inference

perform Cox PH model to estime the log hazard and the hazard ratios.

1. constant only model

```
cox_mod <- coxph(tte ~ 1, data = lung)
summary(cox_mod)
```

```
## Call:  coxph(formula = tte ~ 1, data = lung)
##
## Null model
##   log likelihood= -749.9098
##   n= 228
```

2. multivariable model

```
cox_mod2 <- coxph(tte ~ 1 + ph.ecog + factor(sex) + ph.karno, data = lung)
summary(cox_mod2)
```

```
## Call:
## coxph(formula = tte ~ 1 + ph.ecog + factor(sex) + ph.karno, data = lung)
##
##   n= 226, number of events= 163
##    (2 observations deleted due to missingness)
##
##                     coef exp(coef)  se(coef)      z Pr(>|z|)
## ph.ecog       0.640357  1.897158  0.178127  3.595 0.000324 ***
## factor(sex)2 -0.568822  0.566192  0.168845 -3.369 0.000755 ***
## ph.karno      0.011055  1.011116  0.009535  1.159 0.246292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## ph.ecog         1.8972     0.5271    1.3381    2.6898
## factor(sex)2    0.5662     1.7662    0.4067    0.7883
## ph.karno        1.0111     0.9890    0.9924    1.0302
##
## Concordance= 0.634  (se = 0.026 )
## Rsquare= 0.122   (max possible= 0.999 )
## Likelihood ratio test= 29.37  on 3 df,   p=1.876e-06
## Wald test            = 29.18  on 3 df,   p=2.056e-06
```

```
## Score (logrank) test = 29.54  on 3 df,   p=1.726e-06
```

Chapter 11

Final Words

We have finished a nice book.

Bibliography