Hands-on Biostatistics

Introduction to R

17/7/2020

Introduction to R

"R is an environment, based on S plus language, within which many classical and modern statistical techniques have been implemented. A few of these are built into the base R environment, but many are supplied as packages. There are about 25 packages supplied with R and many more are available through the CRAN family of Internet sites (via https://CRAN.R-project.org) and elsewhere. Most classical statistics and much of the latest methodology is available for use with R, but users may need to be prepared to do a little work to find it." (W. N. Venables, D. M. Smith and the R Core Team., 2018) - Check the R-intro document in Open Source Materials/Hands-on Biostatistics in Drive for more infortmation.

Start with R

Working directory

1. Create a sub-directory named "R" in your "Documents" folder.
2. From RStudio, use the menu to change your working directory under Session > Set Working Directory > Choose Directory. Choose the directory you've just created in step 1

You can run the code below to set working directory also.

```
setwd("C:/Users/user/Desktop/HOB(2020)/R") #set working directory

#getwd() # Shows the working directory (wd)
#setwd(choose.dir()) # Select the working directory interactively
#setwd("C:/myfolder/data") # Changes the wd
#setwd("H:\\myfolder\\data") # Changes the wd
```

Installing/loading packages/

R packages are collections of functions and data sets developed by the community.

Getting help

"Before asking others for help, it's generally a good idea for you to try to help yourself. R includes extensive facilities for accessing documentation and searching for help. There are also specialized search engines for accessing information about R on the internet, and general internet search engines can also prove useful" (https://www.r-project.org/help.html).

```
#?plot # Get help for an object. You can also type: help(plot)

#??regression # Search the help pages for anything that has the word "regression". You can also type:

#help.search("regression")
```

```r
#help(package=car) # View documentation in package car. You can also type: library(help="car")
```

Read/import data in R

Option 1

You can add your data as vectors and create the dataframe using the data.frame() function as follows.

```r
# Data in two numeric vectors

women_weight <- c(38.9, 61.2, 73.3, 21.8, 63.4, 64.6, 48.4, 48.8, 48.5)
men_weight <- c(67.8, 60, 63.4, 76, 89.4, 73.3, 67.3, 61.3, 62.4)

# Create a data frame

my_data <- data.frame(
  group = rep(c("Woman", "Man"), each = 9),
  weight = c(women_weight,  men_weight)
)

# Saving all objects to file *.RData
save.image("my_data.RData")
```

Option 2: Import or read your data from excel, csv.

```r
# Read csv file above my_data
my_data2<- read.csv("C:/Users/user/Desktop/HOB(2020)/R/my_data.csv")


#Read excel file
library(readxl) #Load the readxl package, if not installed, do that before.
my_data3<- read_excel("C:/Users/user/Desktop/HOB(2020)/R/my_data.xlsx")
```

```
## New names:
## * `` -> ...1
```

```r
#Data can be imported from Environment-Import dataset as well
```

Option 3: Load data from R

```r
# Load the data saved as RData in option 1
load("my_data.RData") #Add path to data if necessary
```

Export/write the data in other formats

```r
#Write the dataframe as csv file
write.csv(my_data, "C:/Users/user/Desktop/HOB(2020)/R/my_data.csv")


#Write the dataframe as .txt file
write.table(my_data, file = "my_data.txt", sep = "\t")
```

Data exploration

```
summary(my_data) # Provides basic descriptive statistics and frequencies.
```

```
##     group               weight
##  Length:18          Min.   :21.80
##  Class :character   1st Qu.:51.60
##  Mode  :character   Median :62.90
##                     Mean   :60.54
##                     3rd Qu.:67.67
##                     Max.   :89.40
```

```
edit(my_data) # Open data editor
```

```
##    group weight
## 1  Woman   38.9
## 2  Woman   61.2
## 3  Woman   73.3
## 4  Woman   21.8
## 5  Woman   63.4
## 6  Woman   64.6
## 7  Woman   48.4
## 8  Woman   48.8
## 9  Woman   48.5
## 10   Man   67.8
## 11   Man   60.0
## 12   Man   63.4
## 13   Man   76.0
## 14   Man   89.4
## 15   Man   73.3
## 16   Man   67.3
## 17   Man   61.3
## 18   Man   62.4
```

```
str(my_data) # Provides the structure of the dataset
```

```
## 'data.frame':    18 obs. of  2 variables:
##  $ group : chr  "Woman" "Woman" "Woman" "Woman" ...
##  $ weight: num  38.9 61.2 73.3 21.8 63.4 64.6 48.4 48.8 48.5 67.8 ...
```

```
names(my_data) # Lists variables in the dataset
```

```
## [1] "group"  "weight"
```

```
head(my_data) # First 6 rows of dataset
```

```
##   group weight
## 1 Woman   38.9
## 2 Woman   61.2
## 3 Woman   73.3
## 4 Woman   21.8
## 5 Woman   63.4
## 6 Woman   64.6
```

```
head(my_data, n=4)# First 4 rows of dataset
```

```
##   group weight
## 1 Woman   38.9
```

```
## 2 Woman   61.2
## 3 Woman   73.3
## 4 Woman   21.8
```

```
head(my_data, n= -3) # All rows but the last 3
```

```
##    group weight
## 1  Woman   38.9
## 2  Woman   61.2
## 3  Woman   73.3
## 4  Woman   21.8
## 5  Woman   63.4
## 6  Woman   64.6
## 7  Woman   48.4
## 8  Woman   48.8
## 9  Woman   48.5
## 10   Man   67.8
## 11   Man   60.0
## 12   Man   63.4
## 13   Man   76.0
## 14   Man   89.4
## 15   Man   73.3
```

```
tail(my_data) # Last 6 rows
```

```
##    group weight
## 13   Man   76.0
## 14   Man   89.4
## 15   Man   73.3
## 16   Man   67.3
## 17   Man   61.3
## 18   Man   62.4
```

```
tail(my_data, n=5) # Last 5 rows
```

```
##    group weight
## 14   Man   89.4
## 15   Man   73.3
## 16   Man   67.3
## 17   Man   61.3
## 18   Man   62.4
```

```
tail(my_data, n= -5) # All rows but the first 5
```

```
##    group weight
## 6  Woman   64.6
## 7  Woman   48.4
## 8  Woman   48.8
## 9  Woman   48.5
## 10   Man   67.8
## 11   Man   60.0
## 12   Man   63.4
## 13   Man   76.0
## 14   Man   89.4
## 15   Man   73.3
## 16   Man   67.3
## 17   Man   61.3
```

```
## 18   Man   62.4
```

```
my_data[1:5, ] # First 5 rows
```

```
##   group weight
## 1 Woman   38.9
## 2 Woman   61.2
## 3 Woman   73.3
## 4 Woman   21.8
## 5 Woman   63.4
```

```
my_data[1:5,1:2] # First 5 rows of data of the first 2 variables
```

```
##   group weight
## 1 Woman   38.9
## 2 Woman   61.2
## 3 Woman   73.3
## 4 Woman   21.8
## 5 Woman   63.4
```

Missing data

```
rowSums(is.na(my_data)) # Number of missing per row
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
colSums(is.na(my_data)) # Number of missing per column/variable
```

```
##  group weight
##      0      0
```

```
# Convert to missing data
my_data[my_data$weight=="& ","weight"] <- NA # NOTE: Notice hidden spaces.
my_data[my_data$weight==999,"weight"] <- NA

# The function complete.cases() returns a logical vector indicating which cases are complete.
# list rows of data that have missing values
my_data[!complete.cases(my_data),]
```

```
## [1] group  weight
## <0 rows> (or 0-length row.names)
```

```
# The function na.omit() returns the object with listwise deletion of missing values.
# Creating a new dataset without missing data
my_data1 <- na.omit(my_data)
```

Value labels/recode variables

```
#Read data from ecxel.
library(readxl)
my_data_num <- read_excel("C:/Users/user/Desktop/HOB(2020)/R/my_data.num.xlsx")
```

```
## New names:
## * `` -> ...1
```

```
# Use factor() for nominal data
my_data_num$group <- factor(my_data_num$group, levels = c(1,2), labels = c("male", "female"))
```

```
# Use ordered() not factor() for ordinal data
```

Creating ids/sequence of numbers

```
# Creating a variable with a sequence of numbers from 1 to n (where 'n' is the total number of observat
my_data$id <- seq(dim(my_data)[1])
my_data
```

```
##    group weight id
## 1  Woman   38.9  1
## 2  Woman   61.2  2
## 3  Woman   73.3  3
## 4  Woman   21.8  4
## 5  Woman   63.4  5
## 6  Woman   64.6  6
## 7  Woman   48.4  7
## 8  Woman   48.8  8
## 9  Woman   48.5  9
## 10   Man   67.8 10
## 11   Man   60.0 11
## 12   Man   63.4 12
## 13   Man   76.0 13
## 14   Man   89.4 14
## 15   Man   73.3 15
## 16   Man   67.3 16
## 17   Man   61.3 17
## 18   Man   62.4 18
```

Recoding variables/creating categories

```
library(car)
my_data$weight.rec <- recode(my_data$weight,
"30:50='30-50';
51:70='51-70';
71:90='71-90'")
my_data$weight.rec <- as.factor(my_data$weight.rec)
```

Sort data; Deleting variables

```
#Sort data by weight.rec
my_data.sorted <- my_data[order(my_data$weight.rec),]

#Delete variables
my_data$weight.rec <- NULL
```

Subseting the data

```
mydata3 <- subset(my_data, weight >= 20 & weight <= 50)
mydata4 <- subset(my_data, weight >= 20 & weight <= 50, select=c(id, weight))

mydata5 <- subset(my_data, group=="Woman" & weight >= 70)
mydata6 <- subset(my_data, group=="Woman" & weight == 70)
```

Categorical data: Frequencies/Crosstabs

```
#Frequencies
table(my_data$group)
```

```
##
##   Man Woman
##     9     9
```

```
# Two-way tables/ crosstabs

library(readr)
data <- read_csv("C:/Users/user/Desktop/HOB(2020)/R/chi-square.csv",
    col_types = cols(id = col_number(), improvement = col_character(),
        treatment = col_character()))

#Create crosstab
dt<- table(data$treatment, data$improvement)
dt
```

```
##
##               improved not-improved
##   not-treated       26           29
##   treated           35           15
```

```
addmargins(dt) # Adding row/col margins
```

```
##
##               improved not-improved Sum
##   not-treated       26           29  55
##   treated           35           15  50
##   Sum               61           44 105
```

```
#Calculate proportions
round(prop.table(dt,1), 2) # Round col prop to 2 digits
```

```
##
##               improved not-improved
##   not-treated     0.47         0.53
##   treated         0.70         0.30
```

```
round(100*prop.table(dt,1), 2) # Round col prop to 2 digits (percents)
```

```
##
##               improved not-improved
##   not-treated    47.27        52.73
##   treated        70.00        30.00
```

```
addmargins(round(prop.table(dt,1), 2),2) # Round col prop to 2 digits
```

```
##
##               improved not-improved  Sum
##   not-treated     0.47         0.53 1.00
##   treated         0.70         0.30 1.00
```

```
round(prop.table(dt,2), 2) # Round column prop to 2 digits
```

```
##
##               improved not-improved
```

```
##   not-treated     0.43         0.66
##   treated         0.57         0.34
```

```
round(100*prop.table(dt,2), 2) # Round column prop to 2 digits (percents)
```

```
##
##               improved not-improved
##   not-treated    42.62        65.91
##   treated        57.38        34.09
```

```
addmargins(round(100*prop.table(dt,2), 2),1) # Round col prop to 2 digits
```

```
##
##               improved not-improved
##   not-treated    42.62        65.91
##   treated        57.38        34.09
##   Sum           100.00       100.00
```

```
round(prop.table(dt),2) # Tot proportions rounded
```

```
##
##               improved not-improved
##   not-treated     0.25         0.28
##   treated         0.33         0.14
```

```
round(100*prop.table(dt),2) # Tot proportions rounded
```

```
##
##               improved not-improved
##   not-treated    24.76        27.62
##   treated        33.33        14.29
```

Numerical data

Descriptive Statistics

```
summary(my_data) # Summary of all numeric variables
```

```
##     group               weight           id
##  Length:18          Min.   :21.80   Min.   : 1.00
##  Class :character   1st Qu.:51.60   1st Qu.: 5.25
##  Mode  :character   Median :62.90   Median : 9.50
##                     Mean   :60.54   Mean   : 9.50
##                     3rd Qu.:67.67   3rd Qu.:13.75
##                     Max.   :89.40   Max.   :18.00
```

```
mean(my_data$weight) #mean
```

```
## [1] 60.54444
```

```
median(my_data$weight) #median
```

```
## [1] 62.9
```

```
var(my_data$weight) # Variance
```

```
## [1] 231.3414
```

```
sd(my_data$weight) # Standard deviation
```

```
## [1] 15.20991
```

```r
max(my_data$weight) # Max value
```

```
## [1] 89.4
```

```r
min(my_data$weight) # Min value
```

```
## [1] 21.8
```

```r
range(my_data$weight) # Range
```

```
## [1] 21.8 89.4
```

```r
quantile(my_data$weight) # Quantiles 25%
```

```
##     0%    25%    50%    75%   100%
## 21.800 51.600 62.900 67.675 89.400
```

```r
quantile(my_data$weight, c(.3,.6,.9)) # Customized quantiles
```

```
##   30%   60%   90%
## 60.12 63.64 74.11
```

```r
length(my_data$weight) # Num of observations when a variable is specify
```

```
## [1] 18
```

```r
length(my_data$weight) # Number of variables when a dataset is specify
```

```
## [1] 18
```

```r
table(my_data$group)
```

```
##
##   Man Woman
##     9     9
```

```r
names(sort(-table(my_data$group)))[1]
```

```
## [1] "Man"
```

Descriptive statistics by groups

```r
# Descriptive statistics by groups using --tapply--
mean <- tapply(my_data$weight,my_data$group, mean, na.rm=TRUE)
sd <- tapply(my_data$weight,my_data$group, sd)
median <- tapply(my_data$weight,my_data$group, median)
max <- tapply(my_data$weight,my_data$group, max)

table <- round(cbind(mean, median, sd, max),digits=1)
table
```

```
##       mean median   sd  max
## Man   69.0   67.3  9.4 89.4
## Woman 52.1   48.8 15.6 73.3
```

Confidence intervals for the mean

```r
#install.packages(distributions3)

library(distributions3) # load package
```

```
##
## Attaching package: 'distributions3'
```

```
## The following objects are masked from 'package:stats':
##
##     Gamma, quantile
```

```
## The following object is masked from 'package:grDevices':
##
##     pdf
```

```
# The data
x <- c(8.5, 9.3, 7.9, 9.2, 10.3)
n<-length(x)

# t-student with 4 degrees of freedom
T_4 <- StudentsT(df = 4)

# 95% CI
L1= mean(x) - quantile(T_4, 1-0.05 / 2) * sd(x) / sqrt(n)
L2= mean(x) + quantile(T_4, 1 - 0.05 / 2) * sd(x) / sqrt(n)
#

L1 #7.917
```

```
## [1] 7.916997
```

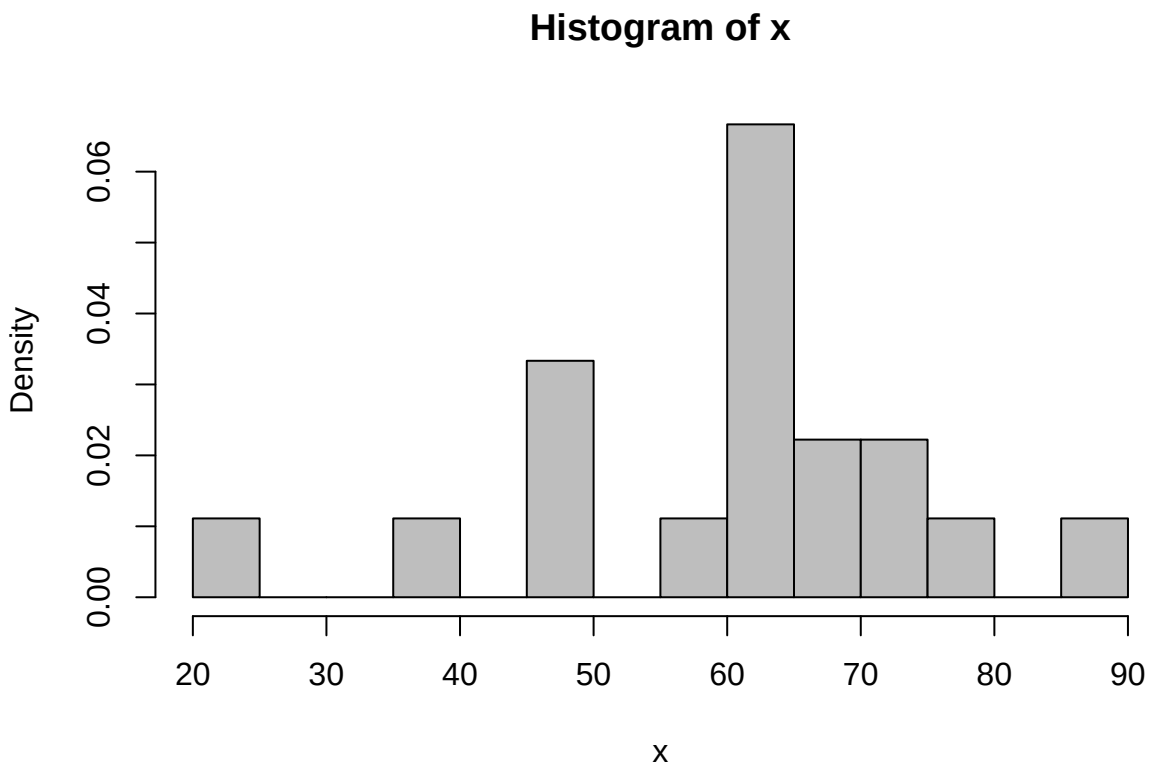
```
L2 # 10.163
```

```
## [1] 10.163
```

Graphs

Histograms

```
x <- my_data$weight
hist(x, freq=F, col="gray", breaks = 10)
```

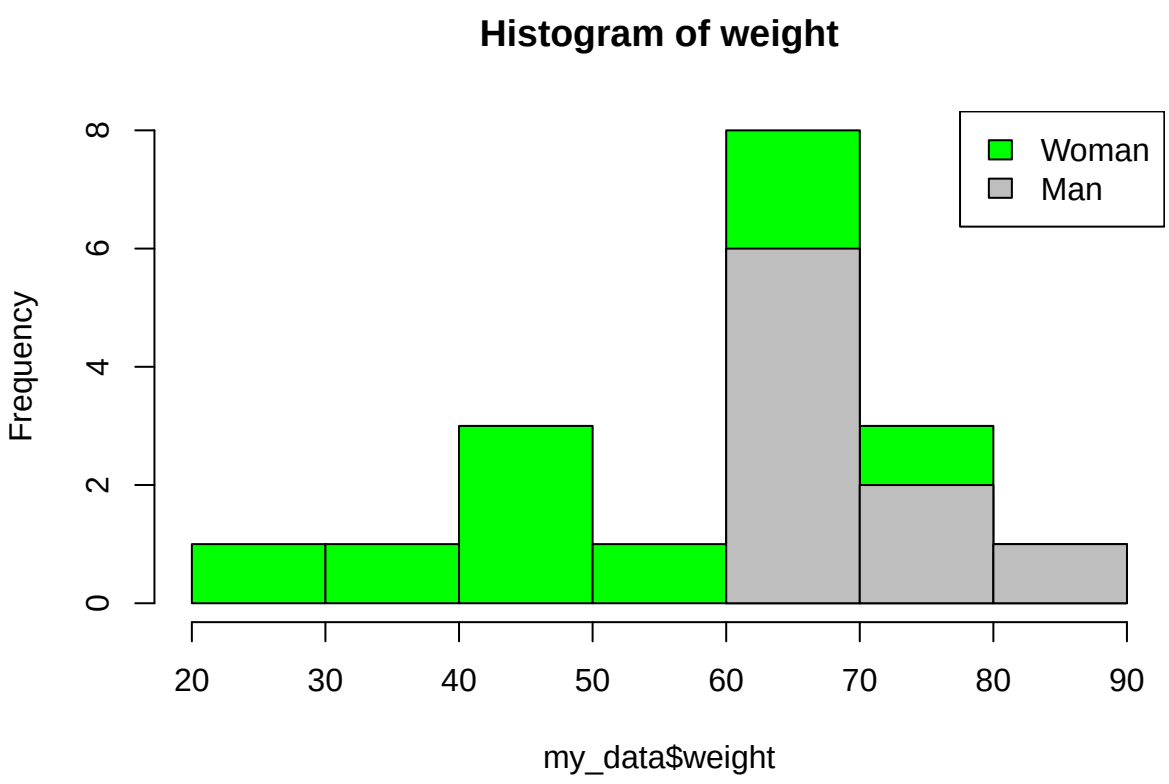



```
help("hist")
```

```
## starting httpd help server ... done
```

Grouped histograms

```
hist(my_data$weight, breaks="FD", col="green", main="Histogram of weight")
hist(my_data$weight[my_data$group=="Man"], breaks="fd", col="gray", add=TRUE)
legend("topright", c("Woman","Man"), fill=c("green","gray"))
```



Other graphs

See other lectures where the data is visualized with appropriate graphs before each statistical model.

Exploring the workspace

```
objects() # Lists the objects in the workspace
```

```
##  [1] "data"           "dt"             "L1"             "L2"
##  [5] "max"            "mean"           "median"         "men_weight"
##  [9] "my_data"        "my_data.sorted" "my_data_num"    "my_data1"
## [13] "my_data2"       "my_data3"       "mydata3"        "mydata4"
## [17] "mydata5"        "mydata6"        "n"              "sd"
## [21] "T_4"            "table"          "women_weight"   "x"
```

```
ls() # Same as objects()
```

```
##  [1] "data"           "dt"             "L1"             "L2"
##  [5] "max"            "mean"           "median"         "men_weight"
##  [9] "my_data"        "my_data.sorted" "my_data_num"    "my_data1"
## [13] "my_data2"       "my_data3"       "mydata3"        "mydata4"
## [17] "mydata5"        "mydata6"        "n"              "sd"
## [21] "T_4"            "table"          "women_weight"   "x"
```

```
remove() # Remove objects from the workspace
rm(list=ls()) #clearing memory space
search() # Shows the loaded packages
```

```
##  [1] ".GlobalEnv"             "package:distributions3" "package:readr"
##  [4] "package:readxl"         "package:car"            "package:carData"
##  [7] "package:stats"          "package:graphics"       "package:grDevices"
## [10] "package:utils"          "package:datasets"       "package:methods"
## [13] "Autoloads"              "package:base"
```

```
library() # Shows the installed packages
dir() # show files in the working directory
```

```
##  [1] "acup_data.csv"                      "acup_data.txt"
##  [3] "ageandheight.xls"                   "chi-square.csv"
##  [5] "Chi_square_Linear_regression.docx"  "Chi_square_Linear_regression.log"
##  [7] "Chi_square_Linear_regression.pdf"   "Chi_square_Linear_regression.tex"
##  [9] "Group_project.pdf"                  "Group_project.Rmd"
## [11] "HOB-2020-_EDA.pdf"                  "HOB-2020-_EDA.Rmd"
## [13] "HOB-2020-_EDA_files"                "HOB-2020-_Intro_R.docx"
## [15] "HOB(2020)_chisquare_regression.Rmd" "HOB(2020)_EDA.Rmd"
## [17] "HOB(2020)_T-test_ANOVA.Rmd"         "HOB_2020_Lecture 2.pdf"
## [19] "my_data.csv"                        "my_data.num.xlsx"
## [21] "my_data.RData"                      "my_data.txt"
## [23] "my_data.xlsx"                       "mywork.RData"
## [25] "one-way-anova.csv"                  "one-way-anova.xlsx"
## [27] "R-4.0.2-win.exe"                    "rtools40-x86_64.exe"
## [29] "T-test_and_ANOVA_R.pdf"
```

References

(https://www.rdocumentation.org/packages)

http://www.sthda.com/english/wiki/running-rstudio-and-setting-up-your-working-directory-easy-r-programming

http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html

http://addictedtor.free.fr/graphiques/

http://addictedtor.free.fr/graphiques/thumbs.php?sort=votes

http://www.statmethods.net/advgraphs/layout.html

http://socserv.mcmaster.ca/jfox/

Quick R http://www.statmethods.net/

UCLA Resources to learn and use R http://www.ats.ucla.edu/stat/R/

https://www.r-bloggers.com/confidence-intervals-for-proportions/