# Machine Learning for Biostatistics

Module 1

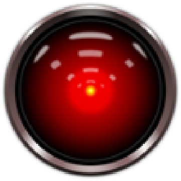
Armando Teixeira-Pinto

2021-07-18

# Contents

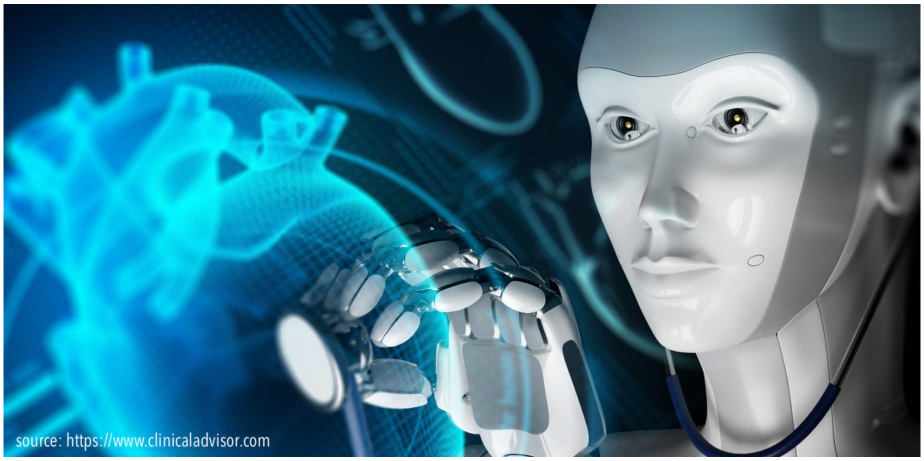
# What is machine learning?

We came a long way since the term *artificial intelligence* was first used by John McCarthy in the 50's. We still don't have computers capable of conversations such as Hal 9000 from the movie 2001: A space odyssey, but we can easily identify some traces of *synthetic intelligence* in many of our interactions with electronic devices.



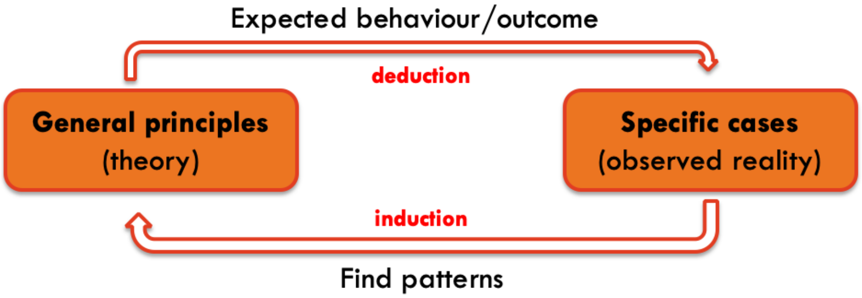
When we ask something to *Siri* or *Alexa*, when we do a search in the web, when we get recommendations for movies or shopping, when our car "reacts" to the proximity of other objects, when our spam email is filtered, when we play chess against a computer or when we are automatically identified in photos posted in social media, these are some simple examples of sophisticated systems that, one way or another, interpret the environment and take actions or make decision that maximize their chances of success.

Several examples can be also found in medical practice, such as patients' access to healthcare (e.g., Babylon), cancer diagnosis (e.g, PathAI), medical imaging (e.g., Zebra Medical Vision), diagnostic support systems (e.g., Buoy Health) and drugs' development.

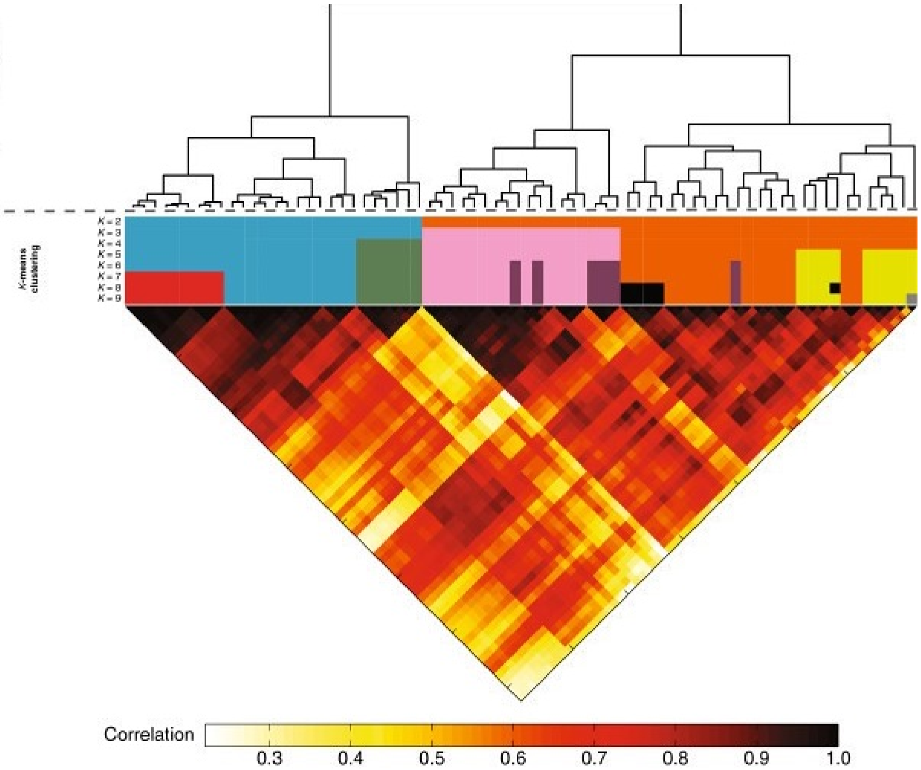

source: https://www.clinicaladvisor.com

In parallel with the development and dissemination of AI, we have also witness a a true data revolution in the past 30 years and a paradigm shift. Data used to be limited and "expensive" to acquire, whereas nowadays we produce more data than we can process. Many of our daily actions are stored in different databases around the world.

This is of particular relevance because AI systems, similar to the way our brain works, rely on two principles to "learn": deduction and induction (I could also include abduction but trying to maintain things simple). Initial AI systems relied heavily in deductive methods. The system was given (taught) several rules and based on these rules and logic principles the system could act. A complementary approach is to "learn" by recognising patterns in the data fed to the system.



The challenges brought by new problems and the availability of data in different format (image, video, text, . . . ) required new approaches outside of the traditional statistical methods. Scientists with computer science and engineering background, as well as statisticians, tackled these problems and developed new methods and algorithms that take advantage of both large amounts of data, and computational power. Despite the clear overlap with statistics, the rapid development of this area, the major contribution from non-statisticians to these methods, and the specificity of some of the problems, fostered the creation of an independent scientific subject: **Machine Learning**. Coming from statistical background (and maybe with narrower focus) we could also call it **Statistical Learning**. In fact, you may notice that the latter is used in the book we will follow.



An interesting perspective about Machine Learning vs Statistics is presented by M. Stewart.

## Datasets used in the examples

The file bmd.csv contains 169 records of bone densitometries (measurement of bone mineral density). The following variables were collected:

- id – patient's number

- age – patient's age
- fracture – hip fracture (fracture / no fracture)
- weight_kg – weight measured in Kg
- height_cm – height measure in cm

- waiting_time – time the patient had to wait for the densitometry (in minutes)
- bmd – bone mineral density measure in the hip

# Chapter 1

# Supervised and unsupervised learning

## 1.1 Introduction

The methods in **machine learning** can be broadly divided into **supervised** and **unsupervised** learning methods.

In **supervised learning**, we have an outcome variable $Y$ (also called output, response, dependent variable) and a set of predictors $\mathbf{X}$ (also called features, dependent variables, covariates). The objective is to estimate the function $f(\mathbf{X})$ that connects $Y$ and $\mathbf{X}$. For example,

$$Y = f(\mathbf{X}) + \varepsilon$$

(if $Y$ is categorical we generally consider the association of $\mathbf{X}$ and the $Pr(Y = k)$)

Having a dataset with observations of $Y$ and $X$ (called training set), we can use these data to find an estimate $\hat{f}(\mathbf{X})$ according to some optimisation principle.

If we specify a functional for $f(\mathbf{X})$, such as the linear regression model:$f(\mathbf{X}) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$, we say that we are using a *parametric* method. If instead we allow the data to estimate the functional form of $f(\mathbf{X})$, such as k-nearest neighbourhood regression, we call the method *non-parametric.*

In **unsupervised learning** , the goal is to model the underlying structure or distribution in the data without having the outcome $Y$. In other words, we want to analyse how the features $\mathbf{X}$ are *clustered* or *associated.* Examples of these methods are *principle components analysis* and *k-means clustering.*

## 1.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 2.1 What is statistical learning

If you are using R for the first time, I encourage you to complete the section

- 2.3 Introduction to R

## 1.3 R review

### Task 1 - Read a dataset and create a new variable

Read the bmd.csv dataset in R. You can download and read the data or use the path *https://www.dropbox.com/s/7wjsfdaf0wt2kg2/bmd.csv?dl=1*

```
bmd.data <- read.csv("https://www.dropbox.com/s/7wjsfdaf0wt2kg2/bmd.csv?dl=1")
```

If you have downloaded the file, you need to substitute the *url* above by the path to file in your computer, e.g., “c:\my files\bmd.csv”

The variable **id** is the identification of the subjects in the dataset, and the variable **age** is their age. What is the age of the subject id=197?

`bmd.data$age` is the vector with the ages of all the subjects. Within that vector we need to find the one where `bmd.data$id == 197`. Notice that double equal sign “==” is a logic statement.

If we write `bmd.data$id = 197` we are assigning the value 197 to the variable **id** and every subject in the dataset gets a new value 197 for that variable.

```
bmd.data$age[bmd.data$id == 197]
```

```
## [1] 69.72845
```

```
#or equivalent
bmd.data[bmd.data$id == 197, "age"]
```

```
## [1] 69.72845
```

```
#or
bmd.data[bmd.data$id == 197, 2]  #age is the second variable
```

```
## [1] 69.72845
```

Let's now create a new variable **age.months** which is age in months (the original is in years).

```
# we just need to multiply the original by 12
bmd.data$age.months <- bmd.data$age * 12

# we can use = instead of <-
bmd.data$age.months = bmd.data$age * 12
```

Finally, let's recode the variable **age** into age categories <=50, (50,60], (60,70] and >60.

```
# we can use the function cut() to
#create the categories
bmd.data$age.cat <- cut(bmd.data$age,
                        c(0,50,60,70, 100))

# alternative
bmd.data$age.cat2[bmd.data$age<50]                     <- 1
bmd.data$age.cat2[bmd.data$age>=50 & bmd.data$age<60] <- 2
bmd.data$age.cat2[bmd.data$age>=60 & bmd.data$age<70] <- 3
bmd.data$age.cat2[bmd.data$age>=70]                   <- 4

#frequencies of the categories
table(bmd.data$age.cat)
```

```
##
##   (0,50]  (50,60]  (60,70] (70,100]
##       24       45       48       52
```

**TRY IT YOURSELF:**

1) Get the ages for all the male subjects, i.e., `bmd.data$sex == "M"`.

See the solution code

```
bmd.data$age[bmd.data$sex == "M"]
```

2) What is the length of the vector above? Or, in other words, how many subjects are male?

See the solution code

```
length(bmd.data$age[bmd.data$sex == "M"])

#or
table(bmd.data$sex)
```

3) Using the variables **weight__kg** and **height__cm**, compute the body mass index? $\frac{\text{weight in Kg}}{(\text{heigh in m})^2}$

See the solution code

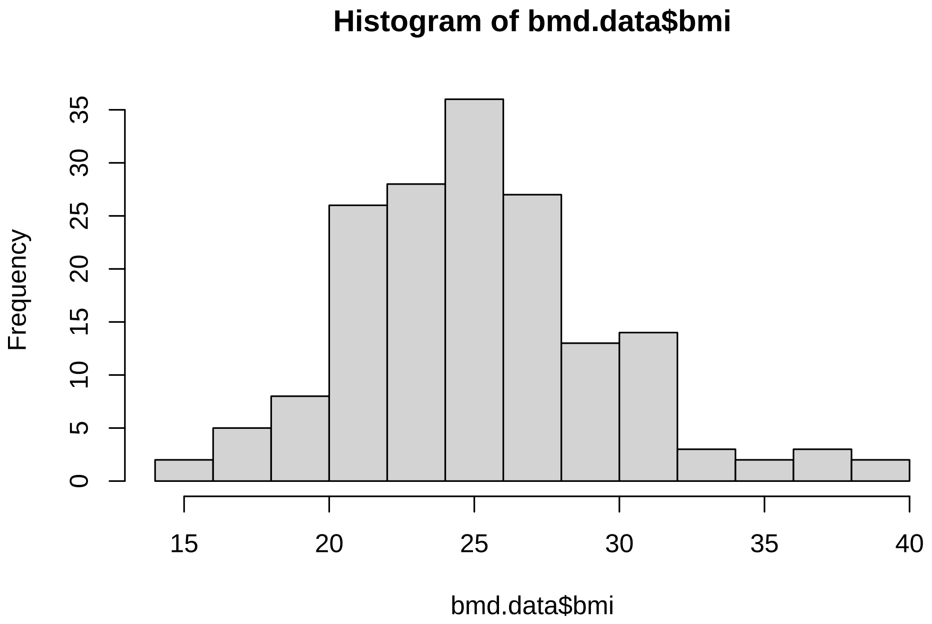
```
bmd.data$bmi <- bmd.data$weight_kg /(bmd.data$height_cm/100)**2
```

## Task 2 - Histogram

There are several packages specific to graphs `ggplot2`, `lattice`, `plotly`, `highcharter`, `sunburstR`, `dygraphs`, `rgl`,… The R Graph Gallery is an excellent learning resource for complex plots and visualisation

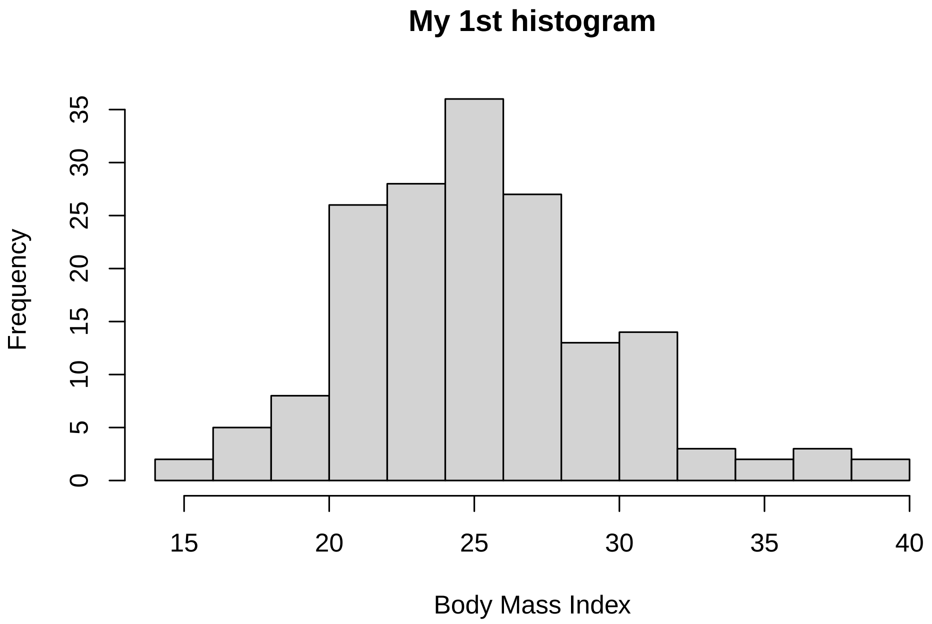
We will start with basic functions. Let's produce the histogram of the variable **bmi** created in task 1

```
hist(bmd.data$bmi)
```



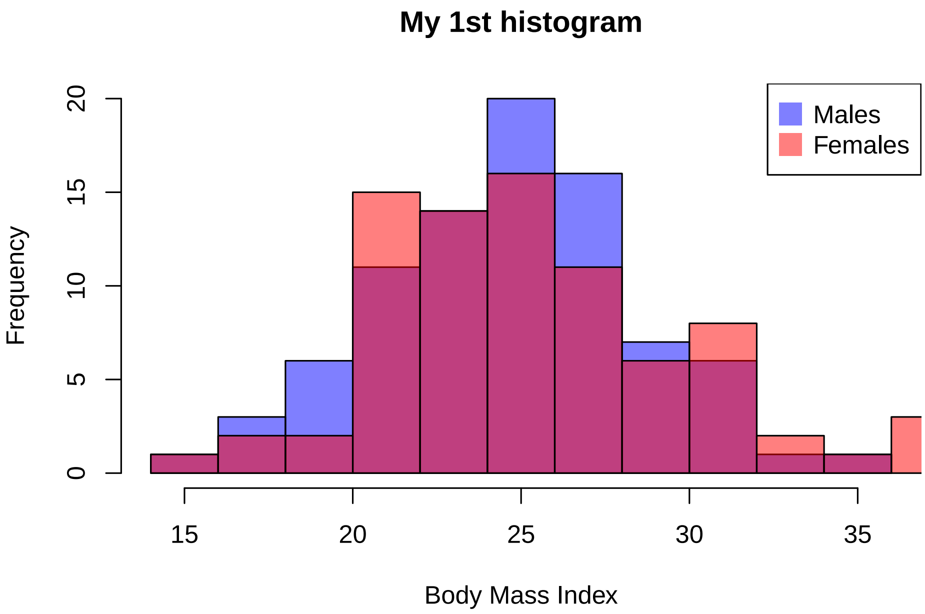
Let's edit the title and x-axis label

```
hist(bmd.data$bmi,
     main = "My 1st histogram",
     xlab = "Body Mass Index")
```



And we can also plot the distribution for males and females, separately

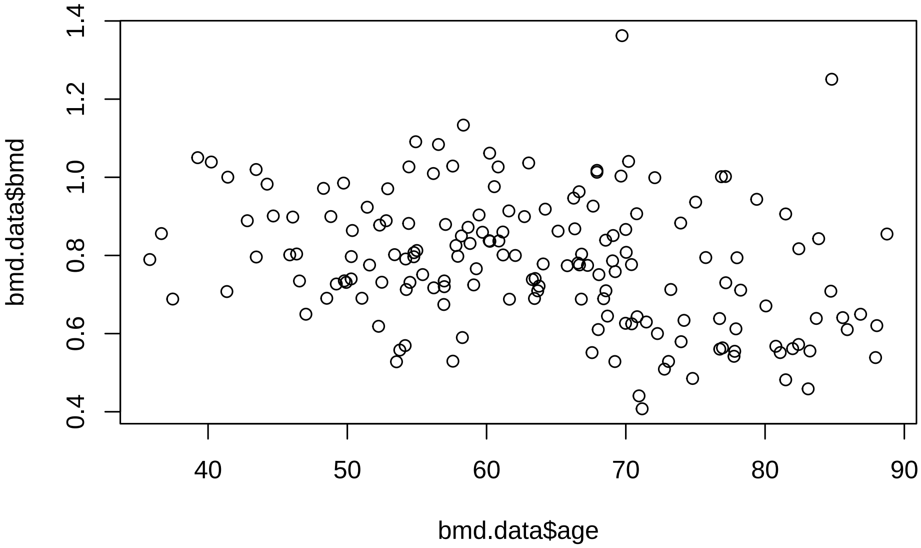
```
#histogram for males
hist(bmd.data$bmi[bmd.data$sex=="M" ],
     breaks = 10,                  # number of cutoff for the bars
     main= "My 1st histogram",
     xlab= "Body Mass Index",
     col=rgb(0,0,1,.5))           # color of the bars

#histogram for females
hist(bmd.data$bmi[bmd.data$sex=="F" ],
      breaks = 10,
     main= "My 1st histogram",
     xlab= "Body Mass Index",
     col=rgb(1,0,0,.5),
     add=T)                      # superimposes the histograms

# Add legend
legend("topright",
       legend=c("Males","Females"),
       col=c(rgb(0,0,1,0.5),  rgb(1,0,0,0.5)),
       pt.cex=2, pch=15 )
```



### Task 3 - Scatter and boxplot

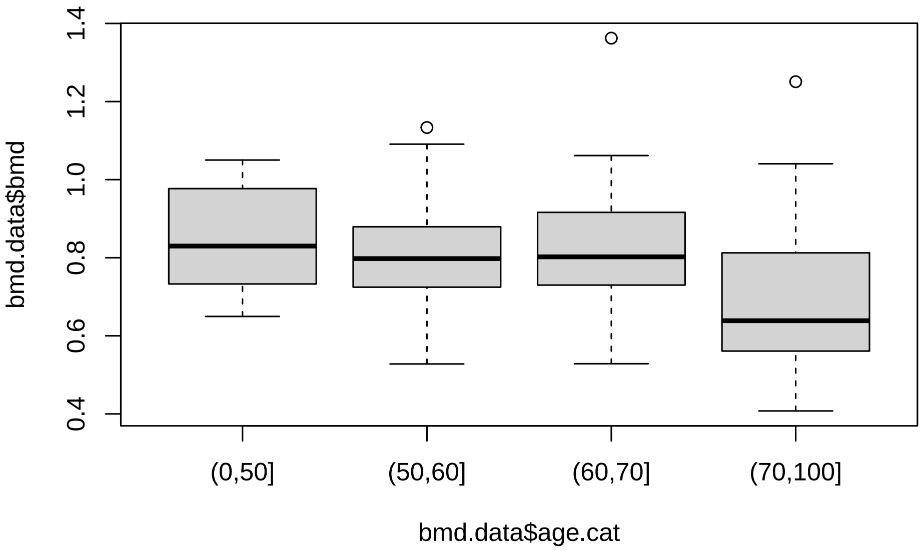
To see the relation between bone mineral density (**BMD**) and **age** we can produce a scatter plot:

```
# BMD
plot(bmd.data$age,bmd.data$bmd)
```



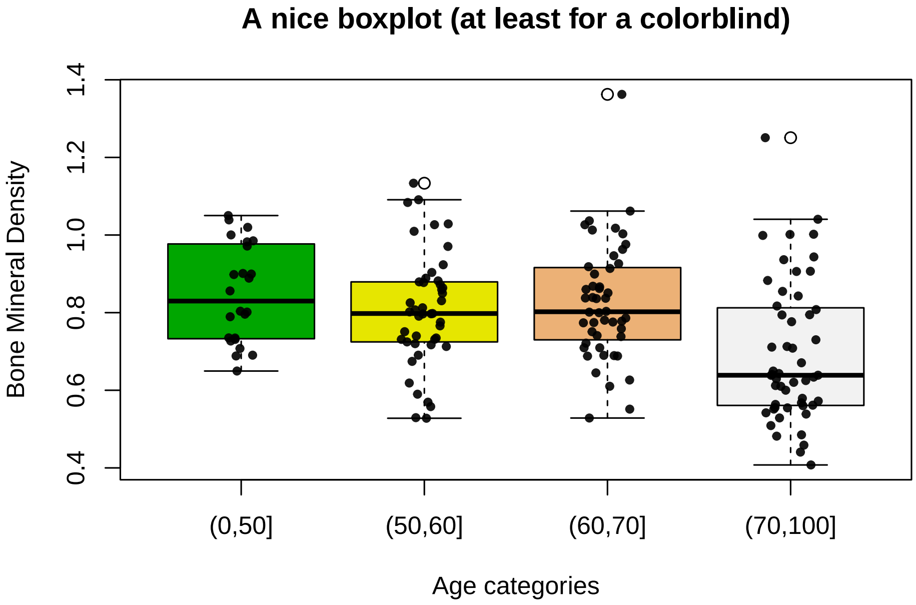
Or a boxplot between bone mineral density (**BMD**) by categories of age created in task 1

```
# BMD
boxplot(bmd.data$bmd ~ bmd.data$age.cat)
```



And with a bit more work:

```
# BMD
boxplot(bmd.data$bmd ~ bmd.data$age.cat,
        main = "A nice boxplot (at least for a colorblind)",
        xlab = "Age categories",
        ylab = "Bone Mineral Density",
        col  = terrain.colors(4) )

# Add data points
mylevels         <- levels(bmd.data$age.cat)
levelProportions <- summary(bmd.data$age.cat)/nrow(bmd.data)

for(i in 1:length(mylevels)){
  thislevel  <- mylevels[i]
  thisvalues <- bmd.data[bmd.data$age.cat == thislevel, "bmd"]
  #take the x-axis indices and add a jitter,
  #proportional to the N in each level
  myjitter   <- jitter(rep(i, length(thisvalues)),
                       amount=levelProportions[i]/2)
  points(myjitter, thisvalues,
    pch=20, col=rgb(0,0,0,.9))
}
```

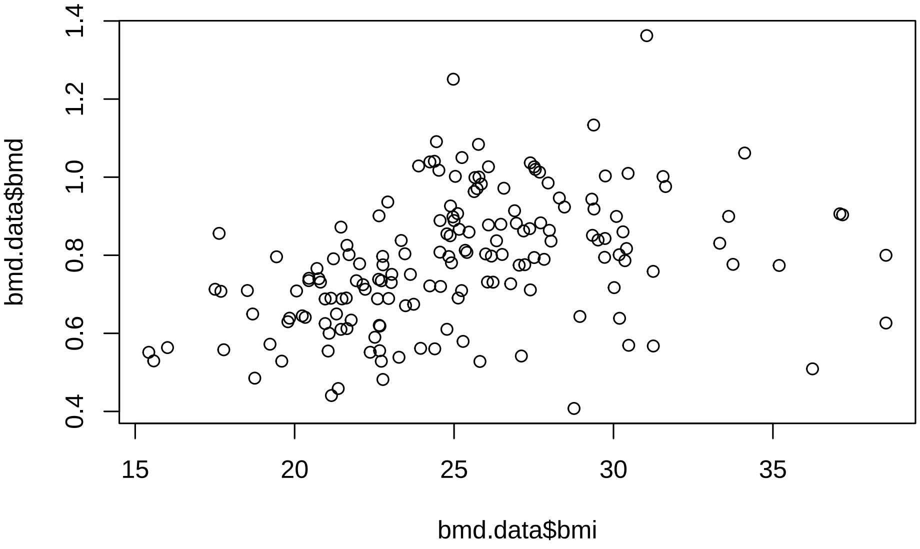


**TRY IT YOURSELF:**

1) Plot a scatter for the relation of **bmi** and **bmd**.

See the solution code

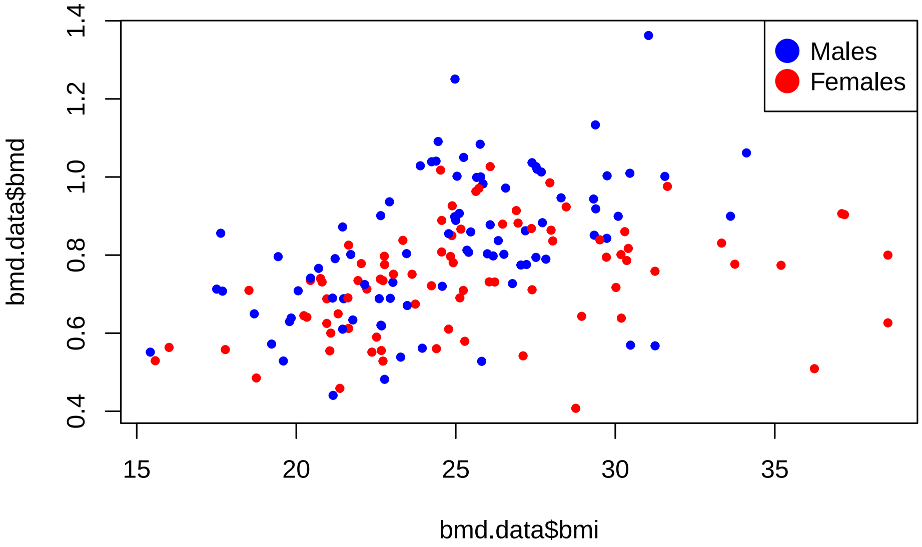
```
plot(bmd.data$bmi,bmd.data$bmd)
```



2) Plot a scatter for the relation of **bmi** and **bmd** with the dots painted by the variable **sex**.

See the solution code

```
plot(bmd.data$bmi,bmd.data$bmd,
     pch=20,
     col= ifelse(bmd.data$sex=="M",   #blue for males
                 "blue",              #red for females
                 "red")
     )

# Add legend
legend("topright",
       legend=c("Males","Females"),
       col=c("blue",  "red"),
       pt.cex=3, pch=20 )
```



# Chapter 2

# Model Accuracy

## 2.1 Introduction

In **machine learning** there is a big emphasis in the prediction ability of the model. We will see several measures of performance for different methods but one commonly used measure (in particular when $Y$ is continuous) is the **mean squared error (MSE)**.

The MSE is defined as:

$$MSE = E\big[(\hat{Y} - Y)^2\big]$$

where $\hat{Y} = \hat{f}(\mathbf{X})$.

We can estimate the MSE using the same data (training data) that we have used to obtain $\hat{f}(\mathbf{X})$ (the training MSE). If we have $n$ observations,

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

and

$$\mathbf{x} = \begin{pmatrix} x_{11} & \dots & x_{p1} \\ x_{12} & \dots & x_{p2} \\ \vdots & \vdots & \vdots \\ x_{1n} & \dots & x_{pn} \end{pmatrix}$$

the estimate of MSE based on the data would then be:

$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_{1i},\ldots,x_{pi})\right)^2$

However, this MSE tends to overestimate the true predictive ability, given that the model is optimised to the training data. Ideally, we would like to evaluate the performance of the model in an independent dataset (test data) with $y^{new}$ and $\mathbf{x}^{new}$.

One important concept associated with the MSE that we will be talking later in the coming modules, is the *bias-variance tradeoff*. The MSE can be decomposed into *bias* and *variance*:

$$MSE = \mathrm{E}\big[(\hat{Y}-Y)^2\big] = \mathrm{E}\big(\hat{Y}^2\big)+\mathrm{E}\big(Y^2\big)-2\mathrm{E}(\hat{Y}Y) = \mathrm{E}\big(\hat{Y}^2\big)+Y^2-2Y\mathrm{E}(\hat{Y})+\mathrm{E}^2(\hat{Y})-\mathrm{E}^2(\hat{Y}) = \underbrace{\big[\mathrm{E}(\hat{Y}) - Y\big]^2}_{bias^2} \tag{2.1}$$

If we use a method that produces unbiased predictions for $Y$, such as the ordinary least squares (OLS) for linear regression, then $\mathrm{E}(\hat{Y}) - Y = 0$. In this case, the MSE simplifies to the $var(\hat{Y})$.

We can see from the decomposition, that an unbiased estimation (prediction) of $Y$ does not lead necessarily to the lowest MSE possible. Once again, the OLS is a good example of this. In the case of high colinearity of the predictors $\mathbf{x}$, we know that the OLS becomes quite "unstable" or in other words, the OLS will have a high variance. In this situation, it may be better to choose a different methods that can produce some bias but will have a much lower variance, resulting in a lower MSE. This is the case of the ridge estimator, as an alternative for the OLS when this estimator has high variance (we will talk about ridge regression in module 4).

Several methods that we will discuss use this principle of exchanging variance for bias.

## 2.2 Readings

Read the following chapters of *An introduction to statistical learning*:

- 2.2 Assessing Model Accuracy

## 2.3 R review

### 2.3.1 Task 1 - Using libraries

Currently, there are more that 16,000 packages (also called libraries) available at CRAN (Comprehensive R Archive Network). Packages are the fundamental units of reproducible R code; they include reusable R functions, the documentation that describes how to use them, and sample data.

We will install and use a library that produces tables similar to the ones used for publication. This library is called tableone‘.

```
install.packages("tableone",
                 repos = "https://cran.rstudio.com/" ) #downloads the package
```

```
##
## The downloaded binary packages are in
##  /var/folders/nj/00dlv8r97qvczd7trdqw283w0000gn/T//RtmplFmhRc/downloaded_packages
```

```
library(tableone)            #loads the package
```

Read the bmd.csv dataset in R and use the function `CreateTableOne()` from the package `tableone` to describe the variable **age**, **bmd** and **sex**.

```
bmd.data <- read.csv("https://www.dropbox.com/s/7wjsfdaf0wt2kg2/bmd.csv?dl=1")
CreateTableOne(c("age", "bmd", "sex"), data=bmd.data)
```

```
##
##                    Overall
##   n                  169
##   age (mean (SD)) 63.63 (12.36)
##   bmd (mean (SD))  0.78 (0.17)
##   sex = M (%)        86 (50.9)
```

Let’s repeat the table above but now stratified by fracture status (variable **fracture**)

```
CreateTableOne(c("age", "bmd", "sex"), data=bmd.data, strata = "fracture")
```

```
##                  Stratified by fracture
##                   fracture      no fracture   p      test
##   n                  50            119
##   age (mean (SD)) 69.77 (13.38) 61.05 (10.97) <0.001
##   bmd (mean (SD))  0.62 (0.10)   0.85 (0.14)  <0.001
##   sex = M (%)        25 (50.0)     61 (51.3)   1.000
```

### 2.3.2 Task 2 - Using ggplot

`ggplot2` is a powerful library that implements a “grammar of graphics” developed by Wilkinson in 1999.

There are seven grammatical elements: * Data - The dataset * Aesthetics - How the variables in the data are mapped to visual properties (aesthetics) of geoms * Geometries - The visual element used for plotting the data * Statistics - Representation of the data to help understand relationships * Facets - Split in multiple plots * Coordinates - Systems of coordinates * Themes - Color schemes, font sizes,. . . .

The combination of these elements, following certain rules, produces the plot

Suppose we want to plot the scatter for **bmd** and **age**.

First we get and load the library

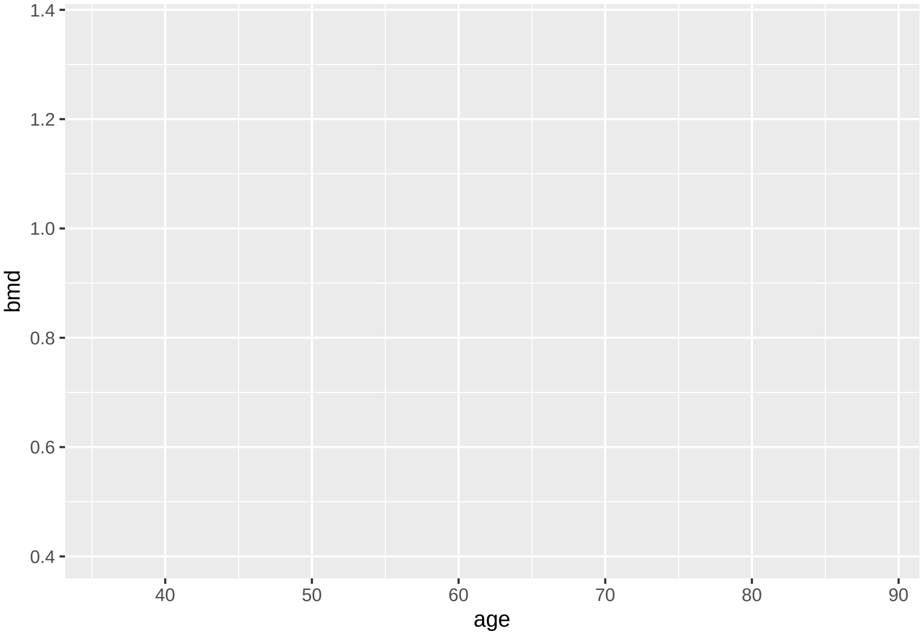
```
install.packages("ggplot2",
                 repos = "https://cran.rstudio.com/" )
```

```
##
## The downloaded binary packages are in
##  /var/folders/nj/00dlv8r97qvczd7trdqw283w0000gn/T//RtmplFmhRc/downloaded_packages
```

```
library(ggplot2)
```

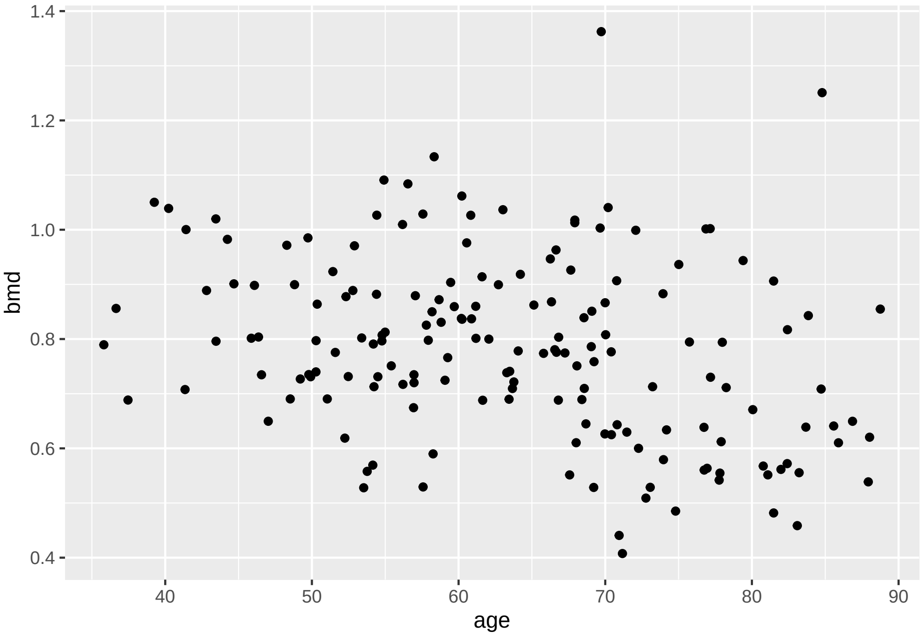
We start by defining the data and aesthetics

```
ggplot(bmd.data, aes(x=age, y=bmd))
```



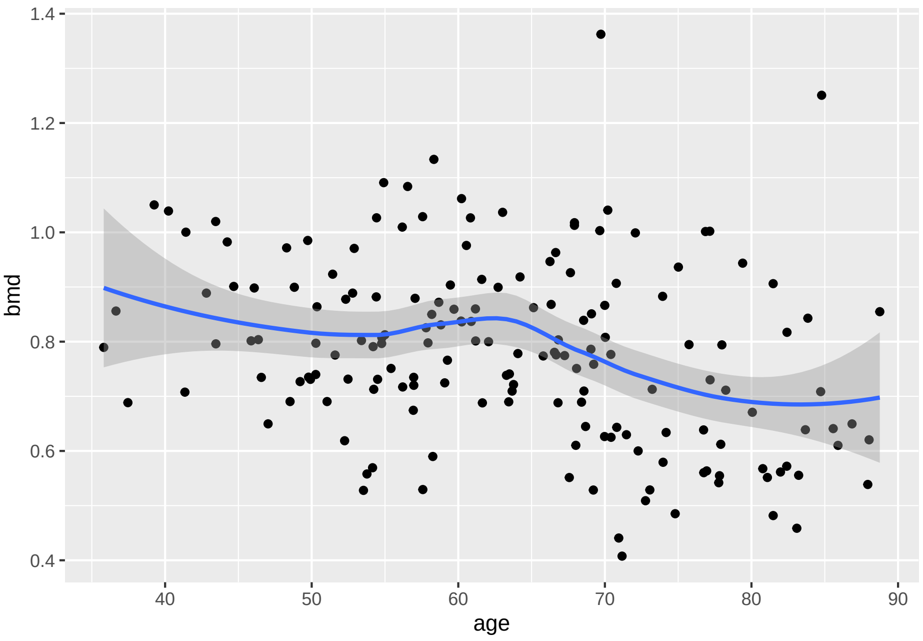
now, we add the geometry (in this case, dots)

```
ggplot(bmd.data, aes(x=age, y=bmd)) +
        geom_point()
```



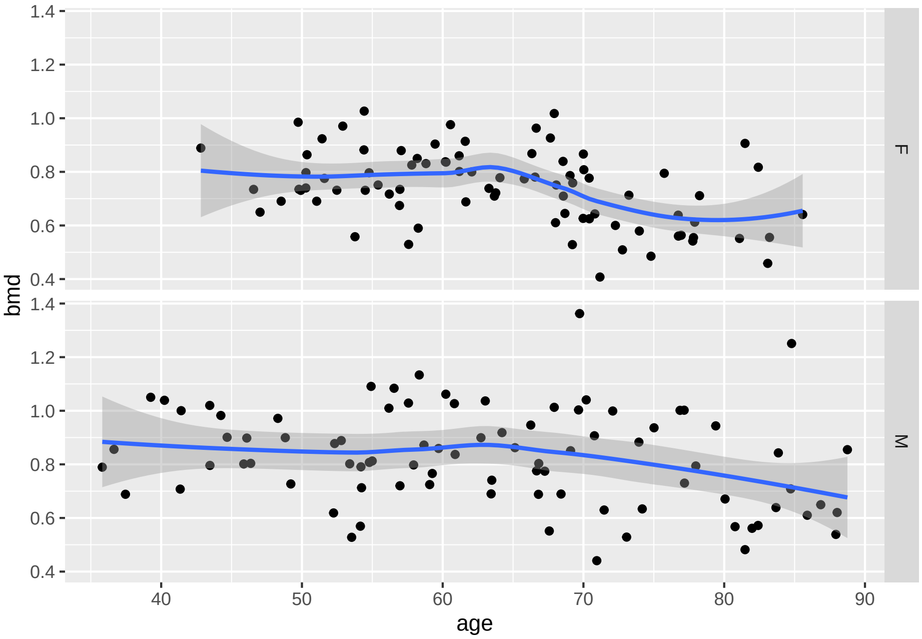
let's add a smooth line (statistics)

```
ggplot(bmd.data, aes(x=age, y=bmd)) +
        geom_point() +
        stat_smooth()
```



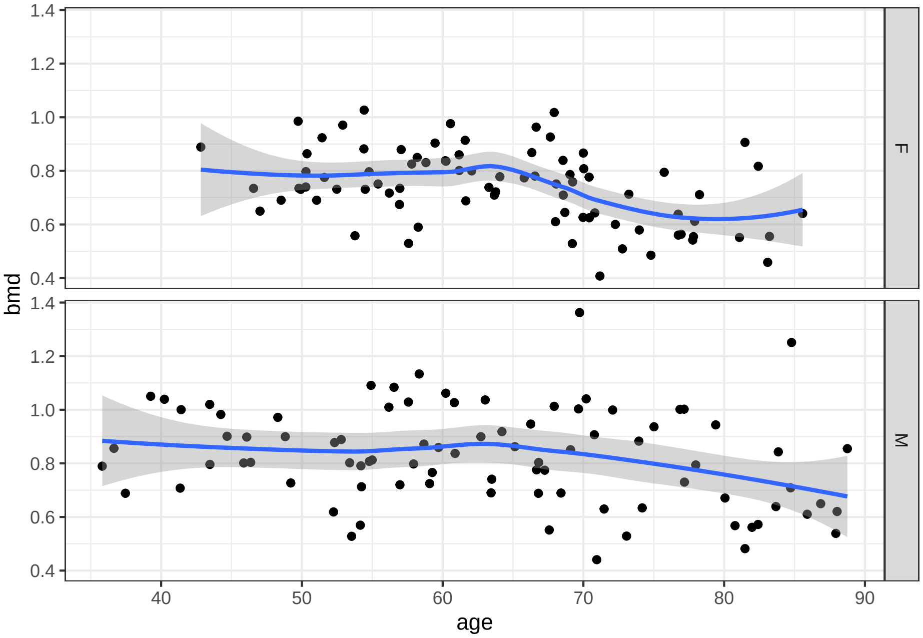
and split by sex (facets)

```
ggplot(bmd.data, aes(x=age, y=bmd)) +
        geom_point() +
        stat_smooth() +
        facet_grid(sex~.)
```



Finally, change the theme

```
ggplot(bmd.data, aes(x=age, y=bmd)) +
        geom_point() +
        stat_smooth() +
        facet_grid(sex~.) +
        theme_bw()
```



### 2.3.3 Task 3 - Writing a function

Write a function that computes the Body Mass Index = weight(kg)/height$^2$(m) using weight and height as arguments

```
bmi.func <- function(W, H){
        bmi <- W/H^2
        bmi <- round(bmi,1) #rounds 1 decimal place
        return(paste("The BMI is ", bmi))
}
```

What is the BMI for an individual with 1.83m weighting 89Kg?

```
bmi.func(89, 1.83)
```

```
## [1] "The BMI is  26.6"
```