Michael Poghosyan

CS 108 - Statistics

(lecture notes)

AUA, Fall 2018

Yerevan, 2018

Exploratory Data Analysis for Univariate Data: Graphical Summaries

So why we need Descriptive Statistics? You can ask - if we have the data, if we have all observations we want, why we need to describe that data? Say, if we have, for a particular student, all his/her grades for all courses, why we need to describe them? And why people calculate the GPA? Hopefully, you will not give the answer to me - you will answer this question by yourself.

Well, of course, having all the datapoints is very nice. But...

Example, Dataset: Look at the following data:
29.70, 29.72, 30.37, 30.19, 30.12, 30.28, 29.88, 29.72, 29.80, 29.99, 29.56, 29.55, 28.99, 29.22, 28.27, 28.05, 27.45, 27.88, 27.94, 29, 27.57, 27.01, 26.15, 26.78, 25.91, 25.53, 25.78, 25.20, 24.55, 24.49, 25.14, 23.82, 23.93, 24.87, 24.18, 24.39, 22.98, 23.83, 20.79, 20.14, 20.49, 18.21, 18.19, 17.88, 17.71, 17.82, 17.5, 17.45, 18.54, 18.76, 16.26, 16.13, 15.64, 15.64

Well, I can guess (and even bet on that for a Bounty chocolate stick) that you have not read all the data values ◡̈ (don't even try to go back to read and win the bet ◡̈). In fact, this is a real example of a data - the weekly close prices (closing prices) for the time period Feb 03, 2017 - Feb 03, 2018 for the General Electric Company stock. Of course, having all the values is very valuable, but if you will show this to your clients, nobody will be happy. So describing - visualizing the data or summarizing a data is important for communicating, and maybe not a trivial task.

For example, we can display the above GE Stock price dataset in a graphical form, and (hopefully) everybody will get the picture: see the Figure 2.1 below.

R code, GE Stock Price: To obtain the GE Stock price Figure 2.1, use:

```
x <- c(29.70, 29.72, 30.37, 30.19, 30.12, 30.28, 29.88, 29.72, 29.80, 29.99, 29.56,
       29.55, 28.99, 29.22, 28.27, 28.05, 27.45, 27.88, 27.94, 29, 27.57, 27.01,
       26.15, 26.78, 25.91, 25.53, 25.78, 25.20, 24.55, 24.49, 25.14, 23.82, 23.93,
       24.87, 24.18, 24.39, 22.98, 23.83, 20.79, 20.14, 20.49, 18.21, 18.19, 17.88,
       17.71, 17.82, 17.5, 17.45, 18.54, 18.76, 16.26, 16.13, 15.64, 15.64)

plot(x, type = "l", lwd = 2, col = "blue", main = "GE Weekly closing prices,
     Feb 03, 2017 - Feb 03, 2018", xlab = "No. of the week in the inverse order",
     ylab = "GE Stock Price")
```

Of course, it will be much better to draw like in the[1] Fig. 2.2.

[1] You can find more info about the "quantmod" package at https://www.quantmod.com/examples/intro/.

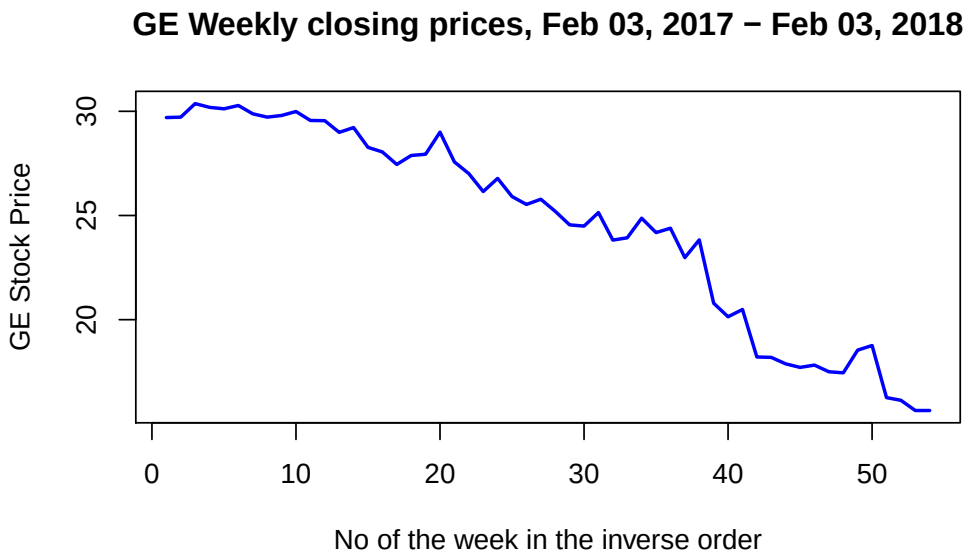


Fig. 2.1: GE Stock Price Data

Fig. 2.2: GE Stock Price Data, using the **R**'s quantmod package

R CODE, GE STOCK PRICE, QUANTMODE PACKAGE: To obtain the GE Stock price Figure 2.2, you need to first install the "quantmod" package. To that end, uncomment the "install.packages" line below and run it.

```
#GE Stock Prices
#You need to install quantmod package, if it is not already installed
#install.packages("quantmod")
library(quantmod) # Some kind of include command from C++
start <- as.Date("2017-02-03")  #Time period beginning
end <- as.Date("2018-02-03")    #Time period end
```

```
getSymbols("GE", src="yahoo", from = start, to = end)
barChart(GE)
```

By the way, in the above example we have a so-called *time series data*. Right this moment we have a course on Times Series at AUA.

The moral of the story: data description and representation is important!

So let's start. Here, for the beginning, we will assume that we have a univariate numerical data (observations, dataset), $x_1, x_2, ..., x_n$, where $x_i \in \mathbb{R}$. In this case we will say that we are given a (univariate, 1D) dataset x. Here x_i-s can be the values of some variable for different observations:

EXAMPLE, 1D DATASET: Say, we are given a dataset x, where x_k is the AUA Math Test grade for the student k, $k = 1, ..., 100$.

EXAMPLE, 1D DATASET: Say, we are doing a salary size study for Math professors in Armenia. We choose a sample of size 10 (some portion of Math Professors ☺), ask them about their salary, and record them as x_1, x_2, ..., x_{10}, where x_1 is the salary of the first Prof, x_2 for the second Prof etc.

Btw, will it be a nice study, if we will choose at random 10 CS students, give them an instruction to ask one of their Math instructors about the salary?

As we have seen above, without any organization of the data it is difficult to get a sense of what is going on in the data: what is the the most typical value, or how concentrated or spread are values about that typical value, or if there are gaps in values etc. So we want to have some simple characteristics that will help to make summaries. There are various graphical representation methods and numerical summaries that help us to draw conclusion about the density/frequency distribution of our data.

So we will give below:

- Graphical Summaries, Graphical Representation Methods;
- Numerical Summaries, Numerical Description Methods.

2.1 Frequency Tables

Before doing a graphical representation, we can get an information about the data by calculating how frequently different numbers appear in our dataset. Of course, this will be meaningful in the case of discrete variable, since for a continuous variable, mostly you will have every datapoint appearing only once[2].

We assume that we have a univariate discrete numerical data $x_1, x_2, ..., x_n$, where $x_i \in \mathbb{R}$, for $i = 1, ..., n$.

[2]Say, if x is a variable showing the lifetime for a light bulb, then, if you are not making rough rounding, say, if you fix the exact (as precise as you can) lifetime of a bulb - something like 2 years, 1 months, 12 days, 23 hours, 22 minutes, 45 seconds and 210 milliseconds, for some bulb, then it is highly unprobable that you will have 2 bulbs with the same lifetime.

Definition 2.1. *The **frequency** of a value t in observations $x_1, x_2, ..., x_n$ is the number of times t occurs in observations:*

$$\textit{Frequency of } t = \textit{number of occurrences of } t \textit{ in data.}$$

Definition 2.2. *The **relative frequency** (or percentage) of a value t in observations $x_1, x_2, ..., x_n$ is the ratio of frequency of t divided by the total number of observations, n:*

$$\textit{Relative Frequency of } t = \frac{\textit{Frequency of } t}{\textit{Total Number of Observations}} = \frac{\textit{Frequency of } t}{n}.$$

Example, Frequency and Relative Frequency Table: Consider the following dataset: the number of A students in my different courses:

$$9, 18, 20, 11, 15, 13, 14, 12, 10, 8, 26, 6, 14, 6$$

To construct the Frequency or Relative Frequency tables, it is convenient to sort our dataset[3]. The sorted dataset is

$$6, 6, 8, 9, 10, 11, 12, 13, 14, 14, 15, 18, 20, 26.$$

Now, the frequency of 6 is 2, since we have 2 sixes in or dataset. The frequency of 8 is 1 etc. The relative frequency of 6 is $\frac{2}{14} = \frac{1}{7}$, since we have 14 observations. Similarly, the relative frequency of 8 is $\frac{1}{14}$ and so on. The result will be:

Table 2.1: The Frequency/Relative Frequency Table

Value	Frequency	Relative Frequency
6	2	2/14
8	1	1/14
9	1	1/14
10	1	1/14
11	1	1/14
12	1	1/14
13	1	1/14
14	2	2/14
15	1	1/14
18	1	1/14
20	1	1/14
26	1	1/14

REMARK, FREQUENCY/RELATIVE FREQUENCY PROPERTIES: Please note that the sum of all frequencies will give the number of all observations:

$$\sum_{x_k \in x} \text{Frequency of } x_k = n,$$

and the sum of all relative frequencies will sum up to 1:

$$\sum_{x_k \in x} \text{Relative Frequency of } x_k = 1.$$

R CODE, FREQUENCY TABLE: To get the frequency table in **R**, you can use the *table* command:

```
#Frequency Table
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
table(x)
```

Here the unique data values and frequencies are not accessible (say, if I want to add all frequencies, or to calculate the relative frequencies). Another way of doing this is the following:

```
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
y <- as.data.frame(table(x))
y  #Displaying the Frequency Table
a <- as.numeric(y$x) #Choosing the values of x
b <- y$Freq #Choosing the frequencies
c <- b/length(x) #Calculating Relative frequencies
y$RelFreq <- c   #Adding a new column to the data frame (table)
y  #Displaying the result
```

2.2 Graphical Representation of Data: Histogram for a Discrete Variable

Now, to give a graphical representation for a Discrete Variable frequency table, one draws the frequency histogram or the relative frequency histogram.

Assume we have some observations of a particular discrete variable: $x_1, x_2, ..., x_n$, with $x_k \in \mathbb{R}$.

Definition 2.3. *The **frequency histogram** for this data is the function* $h : \mathbb{R} \to \mathbb{N} \cup \{0\}$ *defined by*

$$h_{freq}(x) = \text{the number of occurences of } x \text{ in our dataset} = \text{the frequency of } x, \qquad x \in \mathbb{R}.$$

Mathematically, one can define

$$h_{freq}(x) = \sum_{i=1}^{n} \mathbb{1}(x_i = x), \qquad x \in \mathbb{R},$$

where $\mathbb{1}(\cdot)$ is the truth indicator function defined by

$$\mathbb{1}(\text{TRUE}) = 1 \qquad \text{and} \qquad \mathbb{1}(\text{FALSE}) = 0.$$

In other words, this is just the same as the frequency, but in the form of a function defined on $\mathbb{R}$[4].

EXAMPLE, INDICATOR FUNCTION: As an example how the Truth Indicator Function works, assume $t = 3$. Then

$$\mathbb{1}(t < 5) = \mathbb{1}(\text{TRUE}) = 1, \qquad \text{but} \qquad \mathbb{1}(t > 4) = 0.$$

Now, we can define the relative frequency histogram in a similar way:

Definition 2.4. *The **relative frequency histogram** for the above dataset is the function* $h : \mathbb{R} \to [0, +\infty)$ *defined by*

$$h_{relfreq}(x) = \text{the relative frequency of } x = \frac{1}{n} \cdot \sum_{i=1}^{n} \mathbb{1}(x_i = x), \qquad x \in \mathbb{R}.$$

The following properties are very easy to prove:

Proposition 2.1. *For any dataset* $x_1, ..., x_n$,

a. $0 \leqslant h_{freq}(x) \leqslant n$ *for any* $x \in \mathbb{R}$;

b. $\sum_{x \in \mathbb{R}} h_{freq}(x) = n$[5].

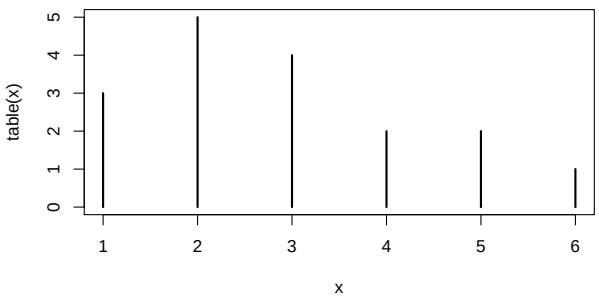
c. $0 \leqslant h_{relfreq}(x) \leqslant 1$ *for any* $x \in \mathbb{R}$;

d. $\sum_{x \in \mathbb{R}} h_{relfreq}(x) = 1$.

Now, having the histogram (or the frequency table), we can represent it on the Cartesian plane by using the Line Graph or a Bar Graph. We put the values of our variable on the OX axis, and for each value, we draw a line of height equal to the frequency (or the relative frequency, if we want to have these values) of that value in the dataset. Below are some examples for the Frequency Histogram, and you can easily update for the Relative ones.

R CODE, LINE GRAPH: The following simple code draws a Line Graph:

```
#Line Graph, Simple Version
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
plot(table(x))
```

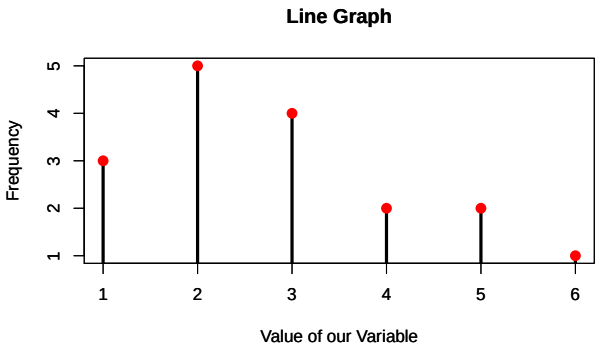


[4]Well, you can say that the above definition of the frequency was also a definition of a function. I will have nothing to argue in this case. Btw, people sometimes define the frequency of x to be 0, if x is not in our dataset.

[5]In fact, the sum involves only finitely many number of non-zero values, so the sum is correctly defined.

R code, Line Graph: Another way to draw a Line Graph:

```
#Line Graph, Another Version
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
y <- as.data.frame(table(x))
a <- as.numeric(y$x)
b <- y$Freq
plot(a,b,type="h", lwd=3,  xlab = "Value of
our Variable", ylab = "Frequency", main
= "Line Graph")
points(a,b, pch=16, cex=1.4, col="red")
```



R code, BarPlot: Barplot is almost the same as the Line Graph above with one difference: it plots rectangles (bars) of small width instead of lines. Here is an example how to do a barplot for the data above. The plot is given in Fig. 2.3.

```
#BarPlot
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
z <- table(x)
barplot(z, main = "BarPlot Example")
```

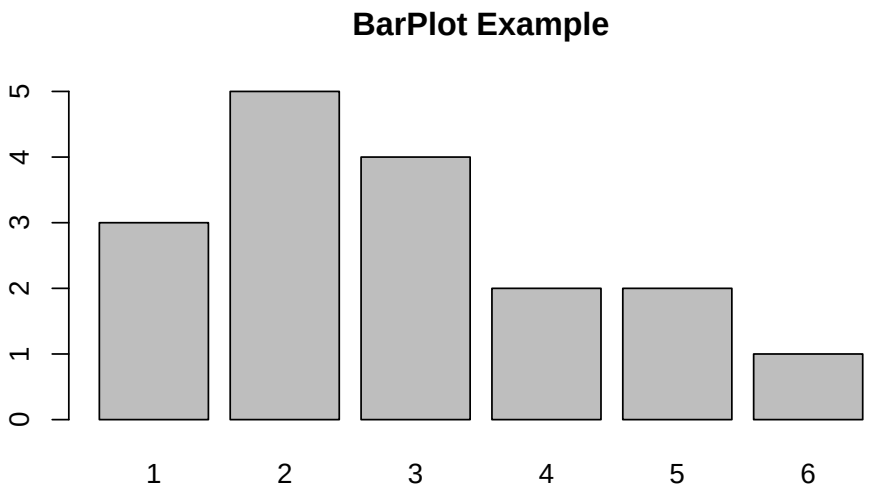


Fig. 2.3: Barplot Example

R CODE, FREQUENCY POLYGON: Another way to graph the frequency table is to give the Frequency Polygon. We plot the points with the value of our variable as the x-coordinate of a point, and the frequency of x as the y-coordinate, we join that points by line segments, and this is the Frequency Polygon.

```
#Frequency Polygon
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
y<-as.data.frame(table(x))
a<-as.numeric(y$x)
b<-y$Freq
plot(a,b,type = "l", lwd = 3, main = "Frequency Polygon", xlab = "Value of our
Variable", ylab = "Frequency")
points(a,b, pch=16, cex=1.4, col="red")
```

The result is in Fig. 2.4.

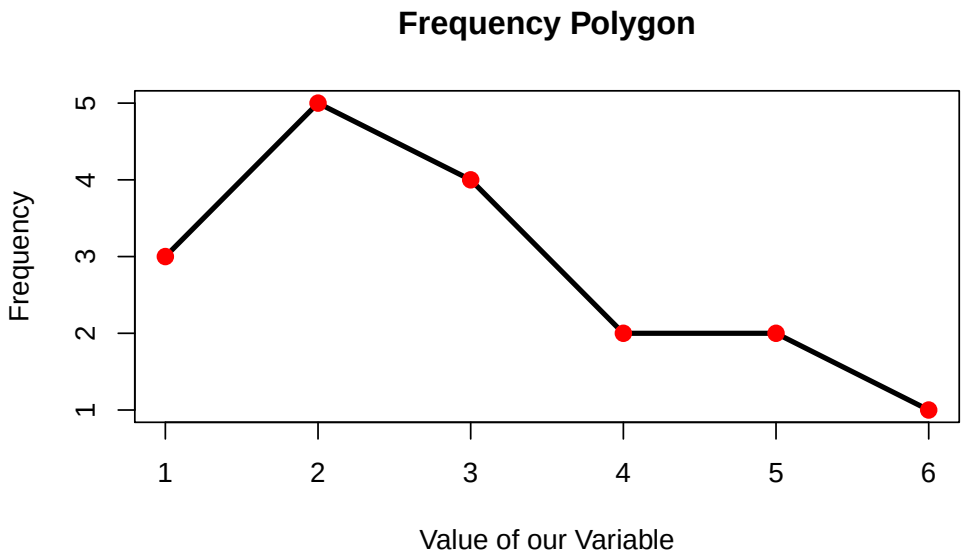


Fig. 2.4: Frequency Polygon Example

2.3 Graphical Representation of Data: Histogram for a Grouped Data / Continuous Variable

Now assume we have some finite number of observations on some continuous variable, $x_1, ..., x_n$. The general idea is the following: we want to group our data into some subgroups, and then visualize.

EXAMPLE, GROUPING: Say, our variable is an age, and we want to group to: ages between 0 and 6 years, between 6 and 15 years, 15 and 120 years; or, our variable is the blood (systolic) pressure, we

group into categories: pressure lower than 80, between 80 and 100, between 100 and 130, higher than 130; or, our variable is the wage of a person, and we group them into: wage between 30,000AMD and 100,000AMD, between 100,000AMD and 700,000AMD, higher than 700,000AMD etc.

To that end, generally, we first fix an interval (usually, a closed interval) containing all observations: $x_i \in I$, for any $i = 1, ..., n$. For example, we can choose $I = [\min x_k, \max x_k]$ or a larger interval. Then we choose a finite partition of the interval I: we choose a finite number of disjoint intervals $I_0, I_1, ..., I_k$, which are called **class intervals** or **bins**, in such a way that

$$I_p \cap I_q = \varnothing, \text{ for } p \neq q, \qquad \text{and} \qquad \bigcup_{j=0}^{k} I_j = I.$$

We will assume that I_j includes its left endpoint but not the right endpoint[6] (except the case when I_j is the rightmost interval - in that case I_j includes also the right endpoint).

EXAMPLE, CLASS INTERVALS OR BINS: Assume our observations lie in the interval $I = [-3, 22]$. Then we can choose the following partition:

$$I_1 = [-3, 1), \ I_2 = [1, 4), \ I_3 = [4, 12), \ I_4 = [12, 22].$$

Or, we can choose for the same example,

$$I_1 = [-3, 10), \quad I_2 = [10, 20), \quad I_3 = [20, 22].$$

EXAMPLE, CLASS INTERVALS OR BINS: For the ages example above, we can take $I = [0, 120]$ and the following partition as bins:

$$I_1 = [0, 6), \ I_2 = [6, 15), \ I_3 = [15, 120].$$

Now, having the class intervals (bins), we calculate the number n_j of datapoints x_i lying in I_j:

$$n_j = \text{the number of data points in } I_j = \sum_{i=1}^{n} \mathbb{1}(x_i \in I_j), \qquad j = 0, 1, 2, ..., k.$$

Definition 2.5. *The **frequency histogram** of our continuous (or a grouped) data $x_1, ..., x_n$ is the piecewise constant function*

$$h_{freq}(x) = n_j, \qquad \forall x \in I_j, \quad j = 1, 2, ..., k.$$

In other words, frequency histogram shows the number of observations in our dataset in each bin, in each class interval. One also defines $h_{freq}(x) = 0$ for all $x \notin I$.

[6] This is just a convention, the left-end inclusion convention. Sometimes people choose the right-end inclusion convention. Important is to know and to state in your example which convention are you using.

EXAMPLE, FREQUENCY HISTOGRAM FOR A CONTINUOUS OR A GROUPED DATA: Consider the following example: or dataset is

$$1,3,4,2,5,6,7,5,3,4,3,5,6,4,5,6,5,7,6,7,8.$$

We choose $I = [0,10]$, which covers the range of our datapoints. Next, we choose some partition of I, say,

$$I_1 = [0,4), \quad I_2 = [4,7), \quad I_3 = [7,10].$$

Then we calculate n_j-s. We have 3 bins, so we need to calculate n_1, n_2 and n_3. For our ease, it will be convenient to sort our dataset:

$$1,2,3,3,3,4,4,4,5,5,5,5,5,6,6,6,6,7,7,7,8$$

- n_1 = the number of observations in $I_1 = [0,4) = 5$;
- n_2 = the number of observations in $I_2 = [4,7) = 12$;
- n_3 = the number of observations in $I_3 = [7,10] = 4$.

Now, the frequency histogram for our data is:

$$h_{freq}(x) = \begin{cases} 5, & x \in I_1 = [0,4) \\ 12, & x \in I_2 = [4,7) \\ 4, & x \in I_3 = [7,10] \\ 0, & \text{otherwise} \end{cases}$$

See below for the code and the graph.

R CODE, HISTOGRAM OF A CONTINUOUS OR A GROUPED DATA:

```
#Frequency Histogram
x <- c(1,3,4,2,5,6,7,5,3,4,3,5,6,4,5,6,5,7,6,7,8)
bins <- c(0,4,7,10)
hist(x, breaks = bins, freq = T, right = F)
# right = F is for excluding the right-endpoints,
# freq = T is to draw the frequencies
```

The result is in Fig. 2.5.

Similarly, we define the relative frequency histogram for a continuous data:

Definition 2.6. *The **relative frequency histogram** of our continuous data $x_1, ..., x_n$ is the piecewise constant function*

$$h_{relfreq}(x) = \frac{n_j}{n}, \qquad \forall x \in I_j, \quad j = 1,2,...,k.$$

In other words, the relative frequency histogram shows the relative frequency (the percentage) of observations in our dataset in each bin, in each class interval.

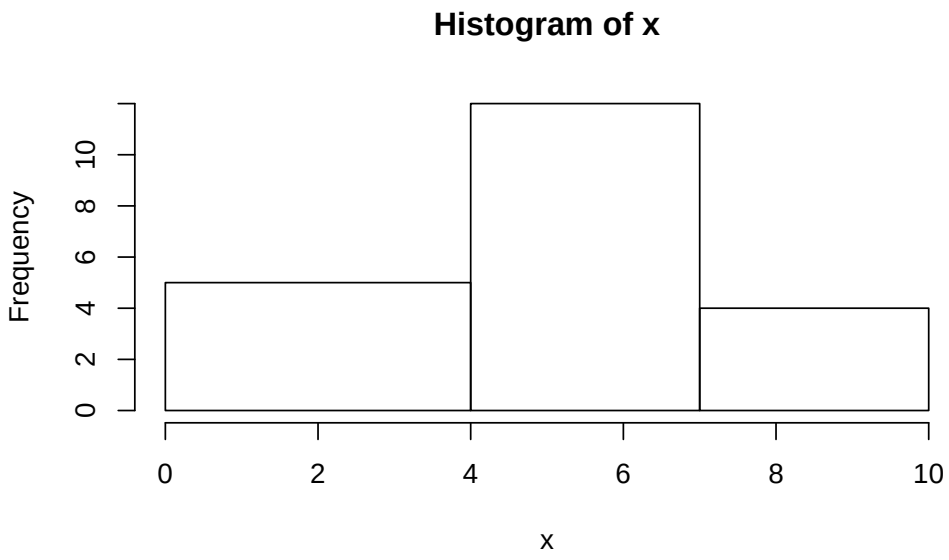


Fig. 2.5: Frequency Histogram

EXAMPLE, RELATIVE FREQUENCY HISTOGRAM FOR A CONTINUOUS OR A GROUPED DATA: For the above example, we will have (note that the number of observations is $n = 21$):

$$h_{relfreq}(x) = \begin{cases} 5/21, & x \in I_1 = [0,4) \\ 12/21, & x \in I_2 = [4,7) \\ 4/21, & x \in I_3 = [7,10] \\ 0, & \text{otherwise} \end{cases}$$

The graph of this function will be the scaled version of the frequency histogram.

For the grouped data case, one defines also the density histogram, which is an important characteristic for our data, as we will see soon:

Definition 2.7. *The **density histogram** or the **normalized relative frequency histogram** of our data $x_1, ..., x_n$ is the piecewise constant function*

$$h_{dens}(x) = \frac{n_j}{n} \cdot \frac{1}{length(I_j)}, \qquad \forall x \in I_j.$$

Here $length(I_j)$ is the length of the interval I_j. Also we define $h(x) = 0$, if $x \notin I$.

In other words,

$$h_{dens}(x) = \frac{\text{Number of datapoints in } I_j}{\text{Total number of elements}} \cdot \frac{1}{length(I_j)}, \qquad \forall x \in I_j.$$

Example, Density Histogram for a Continuous or a Grouped Data: Again, for the above example, we will have

$$\text{length}(I_1) = 4-0 = 4, \quad \text{length}(I_2) = 7-4 = 3, \quad \text{length}(I_3) = 10-7 = 3,$$

so

$$h_{dens}(x) = \begin{cases} \frac{5}{21}\cdot\frac{1}{4}, & x \in I_1 = [0,4) \\ \frac{12}{21}\cdot\frac{1}{3}, & x \in I_2 = [4,7) \\ \frac{4}{21}\cdot\frac{1}{3}, & x \in I_3 = [7,10] \\ 0, & \text{otherwise} \end{cases}$$

The graph of the density histogram function will NOT be, in general, the scaled version of the frequency histogram (unless the lengths of all class intervals are the same). See below for the code and graph.

R code, Density Histogram:

```
#Density Histogram
x <- c(1,3,4,2,5,6,7,5,3,4,3,5,6,4,5,6,5,7,6,7,8)
bins <- c(0,4,7,10)
hist(x, breaks = bins, right = F)
# right = F is for excluding the right-endpoints
```

The result is in Fig. 2.6.

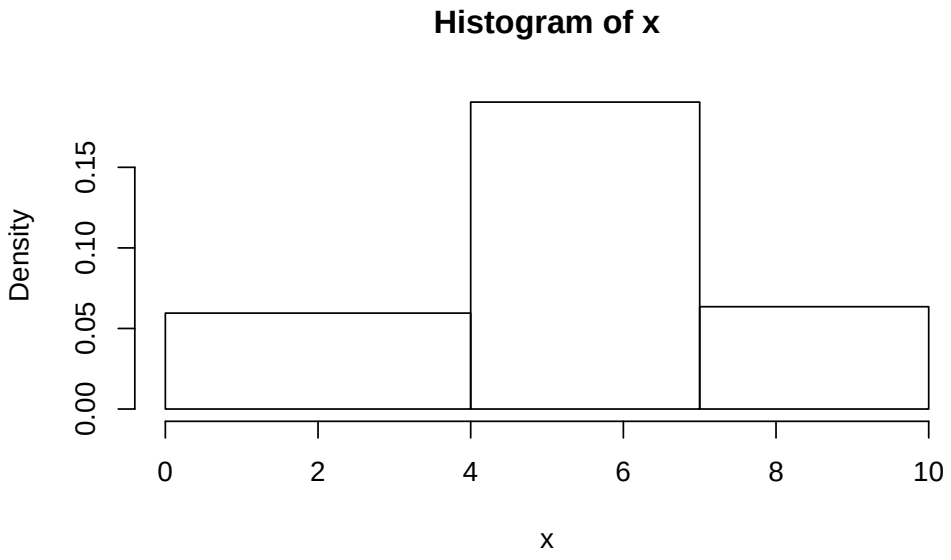


Fig. 2.6: Density Histogram

The idea of the normalization is that in this case we will have that the area under the graph of

the density histogram is 1, i.e., the sum of areas of rectangles represented in the histogram is equal to 1. This makes the density histogram to be some kind of PDF for our data. In fact, when the data comes from some continuous distribution, then the density histogram is an approximation for the PDF of that distribution. We will talk about this again later.

R CODE, HISTOGRAMS: Here are some methods to draw histograms in **R**:

```
#Histogram, basic version
#R itself chooses the bins
x <- rexp(200, rate = 2) #Generates 200 (pseudo)random numbers from the Exp(2) distrib
hist(x) #Basic histogram

#Histogram, v2
bins=c(0,1,3,5) #creates a bin-endpoints vector
hist(x, breaks = bins) #Histogram with custom bins: our bins are [0,1], (1,3], (3,5]
hist(x, breaks = bins, right = F) #Histogram with custom bins:
                                  #our bins are [0,1), [1,3), [3,5]

#Histogram, v3
bins=seq(from=0,to=4, by=0.1) #creates a vector c(0, 0.1, 0.2, ..., 3.9, 4)
                              #equivalent to Matlab's bins = 0:0.1:4
hist(x, breaks = bins)        #Histogram with custom bins:
                              #our bins are [0,0.1], (0.1,0.2],... [3.9, 4]

#Caution: bins need to cover the range:
bins = c(0,1,2)
hist(x, breaks = bins) #This will sometimes give an error
#The point is that we will sometimes obtain numbers x outside the range [0,2]
#So take care to include all datapoints in bins.

#Histogram, v4
x <- rnorm(100) #generating a sample of 100 random numbers from the Standard Normal
hist(x) #Basic Histogram
bins=seq(from=min(x)-1,to=max(x)+1,by=0.8) #Bins with length 0.8,
   #covering the whole range of x's
hist(x, breaks = bins) #Histogram with our custom bins

#Histogram, v5
x <- c(1,2,3,1,2,4,5,3,6,2,3,4,1,5,2,2,3)
hist(x) #Please note that the y-axis shows the frequencies
#another example
x <- rnorm(1400)
hist(x)  #Here again, the y-axis shows the frequencies
hist(x, freq = TRUE)  #The freq parameter controls whether the frequencies
                      #will be displayed or the densities
                      #Here you will have the same result as without the freq parameter
hist(x, freq = FALSE) #This will give the Density histogram
hist(x, freq = FALSE, col = "green") #Guess what is doing this :)
```

Now, let us be back to our density histograms, and talk about the effect of normalization. Recall that any PDF $f(x)$ of a random variable X satisfies the following conditions:

- $f(x) \geqslant 0$, for any $x \in \mathbb{R}$ (in fact, a.e. in $\mathbb{R}$);
- $\int_{-\infty}^{+\infty} f(x)dx = 1$.

It is a notable (and important) fact that the sum of the areas of all rectangles (boxes) of the density histogram, i.e., the area under the density histogram, is 1. In fact:

Proposition 2.2. *If $h_{dens}(x)$ is the density histogram for the dataset $x_1, x_2, ..., x_n$, constructed using the bins $I_1, I_2, ..., I_m$, then*

- $h_{dens}(x) \geqslant 0$, *for all* $x \in \mathbb{R}$;
- $\int_{-\infty}^{+\infty} h_{dens}(x)dx = 1$.

Proof. The first part is obvious, and the second part can be obtained in the following way:

$$\int_{-\infty}^{+\infty} h_{dens}(x)dx = \int_{I} h_{dens}(x)dx = \sum_j \int_{I_j} h_{dens}(x)dx = \sum_j \int_{I_j} \frac{n_j}{n}\frac{1}{|I_j|}dx = \sum_j \frac{n_j}{n}\frac{1}{|I_j|}\cdot|I_j| = \sum_j \frac{n_j}{n} = 1.$$

Also, you can get this result by calculating the sum of the areas of all rectangles representing the histogram. □

Now, if you have the density histogram for some dataset, then it is not too hard to find the relative frequencies of datapoint in each bin:

The relative frequency of datapoints in each class interval (bin) is the **area** of the rectangle over that interval

This is very easy to see, since if we are considering the class interval I_j, then the relative frequency of datapoints lying in I_j is $\frac{n_j}{n}$. On the other hand,

$$\text{Area of the rectangle over the interval } I_j = \text{height}(I_j)\cdot\text{width}(I_j) =$$

$$= \frac{n_j}{n}\frac{1}{\text{length}(I_j)}\cdot\text{length}(I_j) = \frac{n_j}{n}.$$

Example, Reading relative frequencies from the Histogram: Fig. 2.7 shows the density histogram for some dataset (in fact, this is the **R** dataset *state.x77*, the *Murder* column, see the dataset and its documentation). The relative frequency of datapoints in $[10, 12)$ is the area of the rectangle over that interval, approximately, $0.12\cdot(12-10) = 0.24$. In other words, approximately 24% of our data is in $[10, 12)$. Below please find the code drawing the histogram and checking that exactly 24% of all datapoints are in $[10, 12)$.

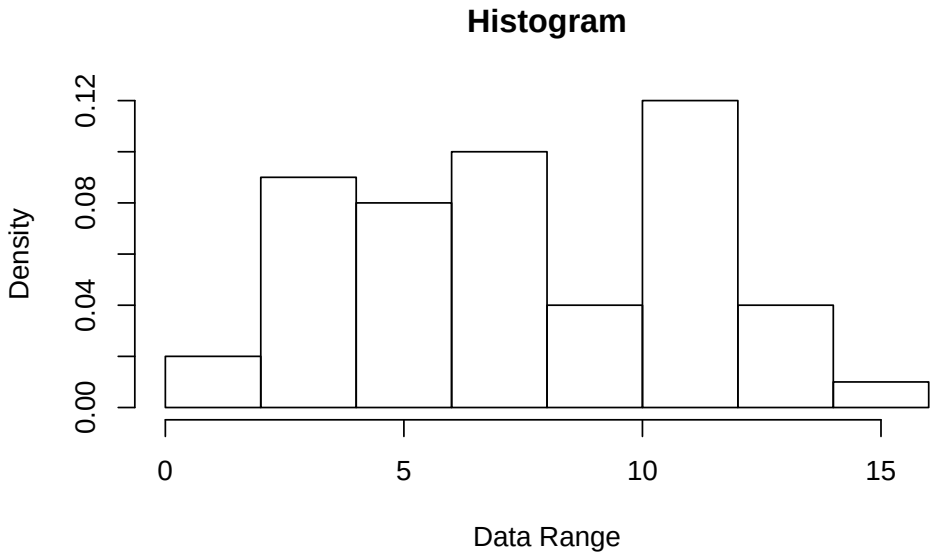


Fig. 2.7: Density Histogram for some Data

R code, Reading relative frequencies from the Histogram:

```
#Reading relative frequencies from the Histogram
x <- data.frame(state.x77) #Transforming the dataset into a data frame
y <- x$Murder # y will be the column (feature) Murder
hist(y, freq = FALSE, main = "Histogram", xlab = "Data Range")
n <- length(y) # total number of datapoints
n.j <- length(y[(y>=10)&(y<12)]) #the number of datapoints in [10,12)
rel.freq <- n.j/n # the ratio of n.j and n, the relative frequency
rel.freq
```

The result is: 0.24

Remark, Histogram and PDF, similarities: Assume we have a dataset $x_1, ..., x_n$, and $h_{dens}(x)$ is its density histogram, constructed using the class intervals I_j, $j = 1, ..., m$. As we have stressed above, h_{dens} is some sort of PDF for our data. And, in some sense, it has many properties of the PDF. Besides the above two properties, we can get the following: recall that if $f(x)$ is the PDF of the r.v. X, then

$$\mathbb{P}(X \in [a, b]) = \int_{[a,b]} f(x)dx.$$

For the density histogram, the above property of calculation of the relative frequencies can be written as:

$$\mathbb{P}(x_k \in I_j) = \int_{I_j} h_{dens}(x)dx = \frac{n_j}{n}.$$

The meaning of this is that if we will make a new r.v. taking the values $x_1, x_2, ..., x_n$, with uniform

probabilities $\frac{1}{n}$, then the probability that this new r.v. will be inside I_j is the integral of h_{dens} over that interval. Or, put in other way, if we are picking at random (uniformly!) one of the points x_k, then the probability that we will get a number inside I_j is $\int_{I_j} h_{dens}(x)dx =$ the relative frequency of our datapoints in I_j.

In some sense, the density histogram gives us an approximation and estimation of the PDF behind the data. The idea is that if we will assume that our data comes from some probability distribution, then, under some mild conditions, the density histogram will approximate the PDF of that distribution.

R code, Density Histogram: Here is the example of PDF estimation by a histogram: we generate 1000 random samples from $Exp(1)$ distribution, and compare the normalized histogram with its PDF.

```
#Histogram approximation of the PDF
#First we plot the graph of Exp(1) distribution PDF
plot(dexp, lwd = 3, col = 'red', xlim = c(0,3), ylim = c(0,1))
#Now we generate a sample of size 1000 from that distribution
x <- rexp(1000)
par(new = TRUE) #This is to keep the previous plot, and to draw over that plot
hist(x, breaks = seq(-3,10,0.2), freq = FALSE, xlim = c(0,3), ylim = c(0,1))
```

The obtained picture is given in Fig. 2.8. By the way, every time you will run this code, it will generate new sample of size 1000, so the picture will not be the same. You can try to run this part many times to see that, in general, Density histogram gives an approximation for the PDF. Also, you can try to increase the number of random numbers, and to decrease the bin lengths.

Please note also that in the Fig. 2.8 the y-labels overlap. you can correct this by giving, say, the same *ylab = 'Density'* parameter in the *plot* and *hist* commands. Try that by yourself. You can try also to run without *xlim* or *ylim* parameters to see why I am using them.

It turns out that to have a good representation for a histogram, one needs to choose the class intervals wisely[7]. The next example is an example of small bins.

R code, Small Bin Sizes:

```
#Histogram with small bins
x <- 0.3*rnorm(300)+0.7*runif(300, -1,1)
hist(x, breaks = seq(min(x)-1,max(x)+1, by = 0.01))
```

See the result in Fig. 2.9.

In fact, if you have a lot of data, then using small bins is something like to consider all your datapoints, one by one. Something like to say: "I have 1 datapoint with the value 0.002, 1 datapoint with the value 0.0019, another one with 0.0022, etc...". The idea of histogram is to group wisely your

[7]Experiment with http://www.shodor.org/interactivate/activities/Histogram/

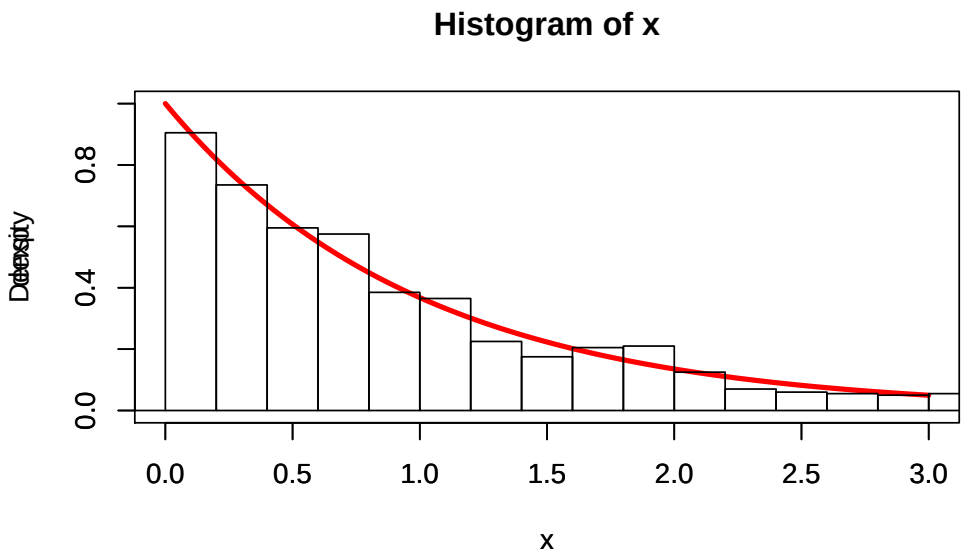


Fig. 2.8: Density Histogram approximation of PDF

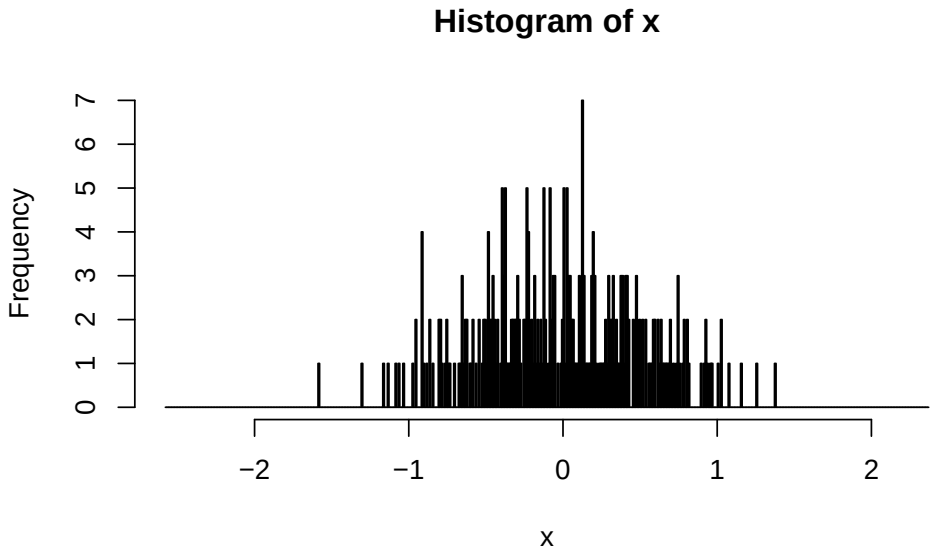


Fig. 2.9: Histogram with small bins

datapoints to keep all important features and information. So taking bin widths small enough will keep the structure of your dataset, and give a lot of information about your dataset, but choosing very small bins will be an overdose[8] ☺. Usually, one takes the lengths of all class intervals to be the same number h, and in that case h is called the bin width. You can consult the literature or Wiki page https://en.wikipedia.org/wiki/Histogram for the choice of bin width h.

[8] An anecdote comes to my mind: students ask their professor: what is the best length for a final project paper? Professor replies - your papers need to be like bikinis: long enough to cover the subject, but short enough to be interesting ☺

So what can be seen using histograms? A lot of things. Among them, histograms can be used to determine if the data:

- is symmetric about some point or is skewed to left or right
- is spread out or concentrated at some point
- has some gaps
- has values far apart from others, has outliers (anomalies)
- is unimodal, bimodal or multimodal[9]

Also, if using the frequency histogram, one can also see the most frequent values range.

R code, Histogram for Precipitation Data:

```
#precip is a standard dataset in R, showing the average amount of precipitation
#(rainfall) in inches for each of 70 United States (and Puerto Rico) cities.
hist(precip, breaks = seq(0,70,5), col = "blue")
```

The result is represented in Fig. 2.10. You can see two gaps - between 0 and 5 and 60 and 65 (so there was no city with average precipitation between 60 and 65, say). The most frequent average precipitation is between 35 and 40 (we have 13 cities having this much average precipitations). The data is 3-modal (?), most of the values are between 30 and 50, and also that one of the cities was unlucky with a flood ☺

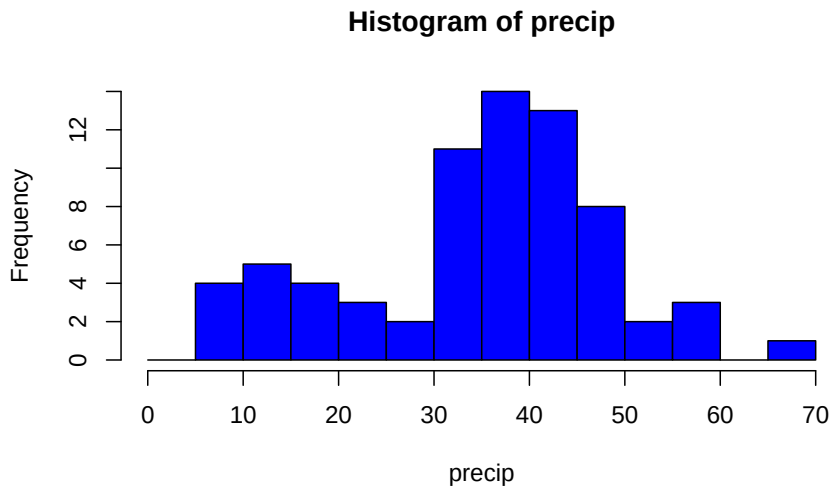


Fig. 2.10: Precipitation data histogram

[9]Well, you can ask - why we need to know if our data is unimodal or bimodal etc. The idea is that later we will try to fit a distribution to a data: say, we have a data, and we want to find a model, theoretical distribution best describing the data. If the data is, say, bimodal, then it is not a clever idea to try to use the Normal distribution (because Normal Distribution is unimodal). What can be done for, say, bimodal dataset? - We can try, for example, the mixture of 2 Gaussian (i.e., Normal) distributions - some weighted sum of two Normal distributions.

REMARK, HISTOGRAM VS BARPLOT: Please note that Barplot and Density Histogram are different things. Barplot is not grouping the data, it just makes a rectangle over the datapoint with the height, which is the frequency (or the relative frequency) of the value. Also, the widths of that rectangles are not important. But this is not the case for Histograms.

2.4 Supplementary: Histograms in Image Analysis

One of the usage of Histograms is in Digital Photography and Image Analysis - the Image Histogram. To make the Histogram for a, say, grayscale image, we take as a variable the level of gray (tone) in the image, ranging from 0 to 255 (0-black, 255-white), and the frequency of the value $x \in \{0, 1, 2, ..., 255\}$ shows how many pixels in our image have the gray level (tone) x. You can find histograms in Digital Cameras, and in software like Adobe Photoshop, IrfanView etc.

Fig. 2.14 gives an example of image histogram. This image is taken from the book **Gonzalez, R.C. and Woods, R.E.**, *Digital Image Processing*, 2008, Pearson/Prentice Hall. See Section 3.3 of that book for Histogram Processing methods. You can also read about the Image Histogram and about the Exposure and Contrast at http://www.cambridgeincolour.com/tutorials/histograms1.htm.

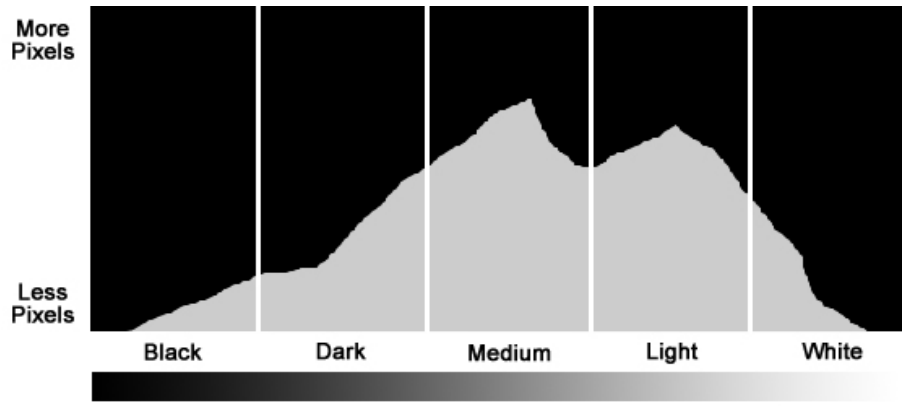


Fig. 2.11: Image Histograms, Explanation: Very few Black areas; An increase in Dark areas; Much more Medium areas; Around the same amount of Light areas; Very few White areas. Source: http://seeinginmacro.com/exposure-histogram-in-photography/

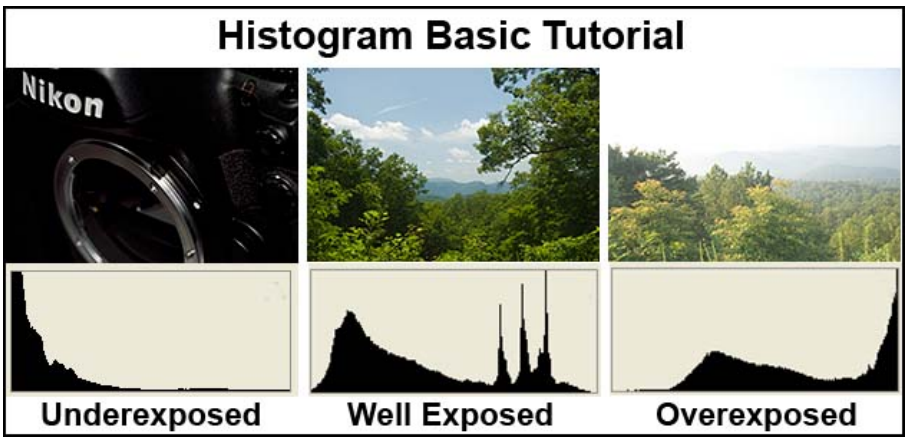


Fig. 2.12: Image Histograms, Explanation

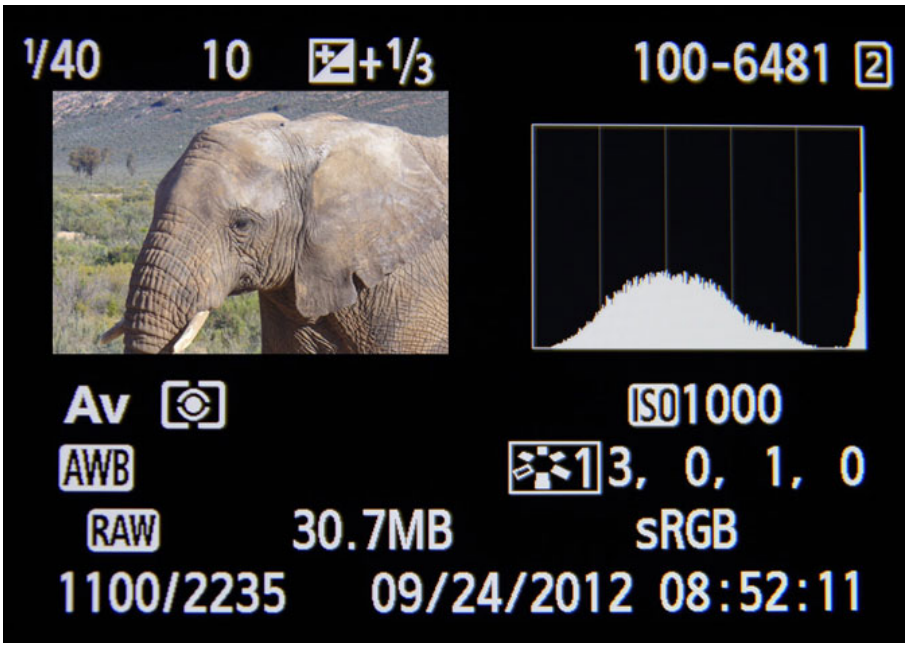


Fig. 2.13: Camera Image Histogram

2.5 Stem-and-Leaf Plots

Stem-and-Leaf plot is another visual representation for a univariate numerical dataset. In fact, it can be viewed as some kind of inverted (rotated) histogram for a small or moderate size dataset.

Assume we have a numerical dataset $x_1, x_2, ..., x_n$, and n is not too large. Drawing the Stem-and-Leaf plot starts by dividing our numbers x_k into Stems and Leafs, and then we will represent each number x_k in the form

$$\mathrm{Stem}\,|\,\mathrm{Leaf}$$

The *Leaf* needs to be a single digit, and *Stem* needs to be an integer (with not too many digits). And each Stem-and-Leaf plots need to contain a key explaining how the datapoints are divided into Stems and Leafs, or, which is the same, how to "recover" the dataset using the Stem-and-Leaf plot. I am using "recover" in quotes, since sometimes we do some rounding before plotting the Stem-and-Leaf plot, so exact recovery will not be possible in that case.

Now, let me explain how to construct Stem-and-Leaf plots, and why the Leaf needs to be a single digit.

EXAMPLE, STEM-AND-LEAF PLOT: Assume we have the following dataset:

$$20, 21, 34, 21, 45, 20, 23, 21, 32, 33, 33, 31, 20, 21, 23$$

Our dataset consists of 2-digit integers, so it is natural to take last digits as Leafs, and first digits as Stems. Say, the number 20 will be represented as

$$2\,|\,0$$

The key here will be that $2\,|\,0$ will be the number 20, or the Leaf shows the units digit, and the Stem shows the decimal digit.

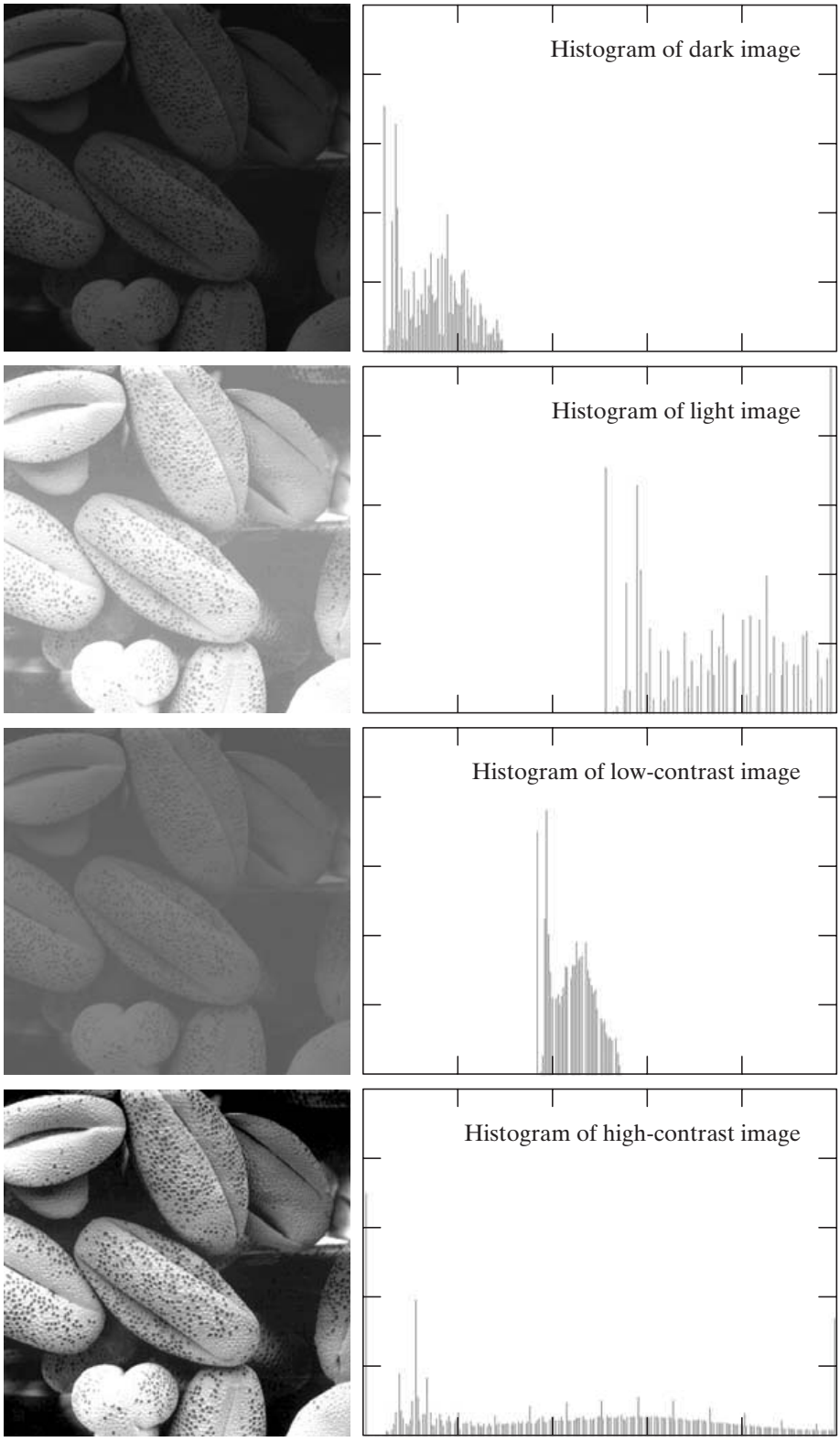


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

Fig. 2.14: Image Histogram, From Gonzalez, Woods, Digital Image Processing

Now, we need some grouping of data. For the Stems we choose, in some sense, bins: say, all datapoints from the interval $[20,30)$ will be displayed in one row, all datapoints from the interval $[30,40)$ will be on the next row etc. Say, the next datapoint 21 will be represented as

$$2|1$$

and we represent the pair $20,21$ in the following way:

$$2|01$$

We will read this "back" as 20 and 21 - this is why we need to choose Leafs to have only one digit. So we have 1 stem here - 2, and we have 2 Leafs here - 0 and 1, so we recover 20 and 21.

Now, say, the next number is 34. So the Stem is 3 and the Leaf for this number is 4. Because our number is not in the first "bin", is not in $[20,30)$, we will write it on the next row. So the Stem-and-Leaf plot for 20, 21 and 34 will be

$$\begin{array}{l} 2 \mid 01 \\ 3 \mid 4 \end{array}$$

Now, hope things are clear. To obtain the Stem-and-Leaf plot for our dataset, we first sort it:

$$20,20,20,21,21,21,21,23,23,31,32,33,33,34,45$$

Now, the Stem-and-Leaf plot will be:

$$\begin{array}{l} 2 \mid 000111133 \\ 3 \mid 12334 \\ 4 \mid 5 \end{array}$$

and the Key is that $2|0$ is 20. **R** is giving the Key in other way, see the code below.

R CODE, STEM-AND-LEAF PLOT: Let us construct the Stem-and-Leaf plot for the above dataset using **R**:

```
#Stem-and-Leaf Plot
x <- c(20, 21, 34, 21, 45, 20, 23, 21, 32, 33, 33, 31, 20, 21, 23)
stem(x)
```

The result will be:

```
  The decimal point is 1 digit(s) to the right of the |

  2 | 000111133
  3 | 12334
  4 | 5
```

So **R** is giving the place of the decimal point - it is 1 digit to the right of |, i.e., for $2|0$ we will have "20." .

Now, another example:

EXAMPLE, STEM-AND-LEAF PLOT: Say, we want to give the Stem-and-Leaf plot for the dataset

$$3.172, 3.145, 3.442, 3.124, 3.456, 2.312, 2.443, 2.111, 2$$

First we sort our dataset:

$$2, 2.111, 2.312, 2.443, 3.124, 3.145, 3.172, 3.442, 3.456$$

The we need to choose Stems and Leafs. Say, we can choose the Leaf to be the last digit, and all other digits will be in the Stem. For example, the Stem-and-Leaf plot representation for 3.172 will be

$$317\,|\,2$$

with the key that this is our number 3.172, or, in the **R** format, we need to say that the decimal point is 2 units left to |. But this is not a good choice for Leafs and Stems. Usually, one first rounds the numbers (well, if it is important not to do a rounding, if it is important to keep the original numbers, choose Stems and Leafs as above): say, we can round to

$$2, 2.1, 2.3, 2.4, 3.1, 3.1, 3.2, 3.4, 3.5$$

Now, it is very natural to choose Leafs the digits after the decimal point, and Stems will be the numbers before the decimal point. Say, for the datapoint 2 we will have $2\,|\,0$ (since $2 = 2.0$), and for the datapoint 2.1 we will have $2\,|\,1$. Next, we need to choose groupings or bins: say, naturally, we can choose to represent in the same row the number in $[2, 3)$ and in $[3, 4)$: in this case our dataset Stem-and-Leaf plot will be:

$$\begin{array}{l} 2 \mid 0134 \\ 3 \mid 11245 \end{array}$$

and the Key is that $2\,|\,0$ is 2.0.

If we will choose another bins, say, we will group in a row datapoints in $[2, 2.5)$, $[2.5, 3)$, $[3, 3.5)$ and $[3.5, 4)$, then we will have the following Stem-and-Leaf plot for our dataset:

$$\begin{array}{l} 2 \mid 0134 \\ 2 \mid \\ 3 \mid 1124 \\ 3 \mid 5 \end{array}$$

and again the Key is that $2\,|\,0$ is 2.0. Below please find the **R** code for this example - **R** will do this last grouping.

R CODE, STEM-AND-LEAF PLOT: Let us construct the Stem-and-Leaf plot for the above dataset using **R**:

```
#Stem-and-Leaf Plot
x <- c(3.172, 3.145, 3.442, 3.124, 3.456, 2.312, 2.443, 2.111, 2)
stem(x)
```

The result will be:

```
  The decimal point is at the |
```

```
  2 | 0134
  2 |
  3 | 1124
  3 | 5
```

So **R** is giving the place of the decimal point - it is at the |, i.e., for 2|0 we will have "2.0".

R code, Stem-and-Leaf Plot: You can control the Stem-and-Leaf Plot (choice of grouping "bins") in **R**. Please run and compare the following code results, for the above example:

```
#Stem-and-Leaf Plot, different scale parameters
x <- c(3.172, 3.145, 3.442, 3.124, 3.456, 2.312, 2.443, 2.111, 2)
stem(x)
stem(x, scale = 1)
stem(x, scale = 2)
stem(x, scale = 0.5)
```

Now, lets do the inverse thing: read the data from the Stem-and-Leaf plot:

Example, Reading Stem-and-Leaf plot: Consider the following Stem-and-Leaf plot obtained in **R**:

```
  The decimal point is 1 digit(s) to the right of the |

   0 | 24004678
   2 | 002466668822244466
   4 | 002668024466
   6 | 046806
   8 | 04523
  10 |
  12 | 0
```

The Key says how to read the data: say, 0|2 means $02 = 2$, and 6|4 means 64. So we can recover our dataset (if no roundings were made): the first elements are 02=2 and 04=4, but the next 0 is confusing - it is not in the increasing order. But the point is that we need to take into account our bins - here **R** is grouping with class intervals (look at the stems!) $[0, 20)$, $[20, 40)$, $[40, 60)$,..., $[120, 140)$. So in the first row, after 2,4, the third element 0 means 10 (!!). Then again 10, then 14, 16, etc. So the first row of our Stem-and-Leaf plot can be decoded as

$$2, 4, 10, 10, 14, 16, 17, 18.$$

The second row ten will be decoded as

$$20, 20, 22, 24, 26, 26, 26, 26, 28, 28, 32, 32, 32, 34, 34, 34, 36, 36$$

The row with Stem 8 is obtained from the datapoints

$$80, 84, 85, 92, 93$$

and the last datapoint in our dataset is 120 (think: why not 130? ◡̈). The above Stem-and-Leaf plot was obtained using the code give below.

Uff!

R code, The above example code: To obtain the above Stem-and-Leaf plot, I have used the *cars* dataset form **R**:

```
#Stem-and-Leaf plot example
cars  #This will show the standard loaded dataset cars
x <- cars$dist # we take the dist column of cars dataset
x #this will show the vector x
sort(x) #sort the dataset, so it will be easy to compare with the Stem-and-Leaf plot
stem(x) #Stem-and-Leaf plot with default scale 1
```

Voila!

Next, you can try, as a continuation of the same example, to run

```
stem(x, scale = 2)
```

R code, Stem-and-Leaf plot vs Histogram: Let us get the Stem-and-Leaf plot and Histogram for the same dataset *faithful*, which is one of the standard embedded datasets in **R** (it shows the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.)

```
#Stem-and-Leaf plot vs. histogram
faithful #faithful dataset of R
duration = faithful$eruptions #the eruptions column data of the dataset
stem(duration)
hist(duration, breaks = seq(from = 1.4,to = 5.8, by =0.2))
#stem(duration, scale = 2)  #you can try this also
```

The results are given in Fig. 2.16-2.15. Note the shape similarity (well, if you will rotate one of the graphs ◡̈)!!

Remark, Difference between SaL and Hist: One advantage of Stem-and-Leaf plot against the histogram is that it shows the original dataset (after, maybe, some rounding), shows all the datapoints (so in case of Stem-and-Leaf plot you can "recover" all your datapoints, but not for the histogram). The disadvantage is that Stem-and-Leaf plot will not work for a large dataset, and, also, you need to calculate by hand, if necessary, the frequencies, relative frequencies etc, in this case.

2.6 Supplementary: Kernel Density Estimation

As we have seen above, we can get an information about the density of our dataset using the density histogram or Stem-and-Leaf plots. Stem-and-Leaf plot works only for small size datasets, and one

```
The decimal point is 1 digit(s) to the left of the |

16 | 070355555588
18 | 000022233333335577777777888822335777888
20 | 00002223378800035778
22 | 0002335578023578
24 | 00228
26 | 23
28 | 080
30 | 7
32 | 2337
34 | 250077
36 | 0000823577
38 | 2333335582225577
40 | 000000335778888800233555577778
42 | 033355557788002333335555777778
44 | 022223355577800000000233333577788888
46 | 0000233357700000023578
48 | 00000022335800333
50 | 0370
```

Fig. 2.15: Stem-and-Leaf plot

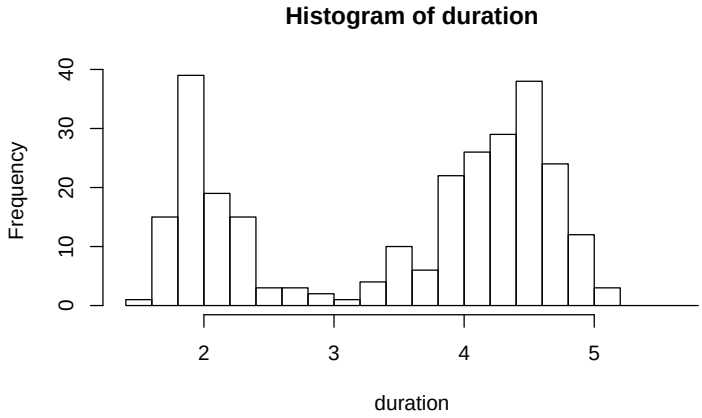


Fig. 2.16: Histogram

of the drawbacks of the histogram is that it is not smooth - it is a piecewise constant (stepwise) function. Another issue with histograms is that it depends on the choice of the class intervals. Usually, one fixes a step-size h and takes class intervals to be adjacent intervals of width h, starting from some point a, i.e., intervals of the form $[a, a+h]$, $(a+h, a+2h]$, $(a+2h, a+3h]$, ... But in this case the form of the histogram will depend on a and h. If, say, we will fix h, then by changing a, we will obtain different histograms. This is why (and not only ☺) people try to construct a smoothed version of the histogram. Kernel Density Estimate[10] (KDE) is some kind of smoothed histogram.

To explain the idea of KDE, and where it comes from, first let me give another type of histogram. Let $x_1, x_2, ..., x_n$ be our datapoints. First, I want to give our ordinary Density histogram, but for equal-length bins: choose an origin a (this can be, for example, $\min\{x_k\}$), choose a bin size $h > 0$, and take

$$I_1 = [a, a+h], \quad I_2 = (a+h, a+2h], \quad I_3 = (a+2h, a+3h], ...$$

as our class intervals (bins). Then

$$h_{dens}(x) = \frac{\text{Number of datapoints in the same bin as } x}{\text{Total number of points}} \cdot \frac{1}{length(I_j)} =$$

$$= \frac{\text{Number of datapoints in the same bin as } x}{nh}.$$

Now, this definition depends on the choice of a, the starting point. Another way to define a histogram is the following:

[10] It is called an Estimate, because it is an estimate for the PDF behind our data.

Definition 2.8. *The **Naive Estimator** of the PDF is the following function*[11]

$$h_{naive}(x) = \frac{\textit{Number of datapoints in } \left(x-\frac{h}{2}, x+\frac{h}{2}\right]}{nh}.$$

This will again produce a piecewise constant function.

EXAMPLE, NAIVE PDF ESTIMATOR: Here we need to have an example. Write that by yourself!

Next, let us rewrite the definition of the Naive Estimator using the following function (called a rectangular kernel with bandwidth 1):

$$k(t) = \begin{cases} 1, & \text{if } t \in \left(-\frac{1}{2}, \frac{1}{2}\right] \\ 0, & \text{otherwise.} \end{cases}$$

Then[12]

$$h_{naive}(x) = \frac{1}{nh} \cdot \sum_{i=1}^{n} k\left(\frac{x-x_i}{h}\right).$$

Now, to obtain a smooth version of this, one chooses a smooth Kernel function $K(t)$, here, a function with

$$K(t) \geqslant 0, t \in \mathbb{R}, \qquad \text{and} \qquad \int_{-\infty}^{+\infty} K(t)dt = 1.$$

For example, we can take the Gaussian Kernel

$$K(t) = \frac{1}{\sqrt{2\pi}} \cdot e^{-t^2/2}, \qquad t \in \mathbb{R},$$

or any other PDF.

Next, one defines the Kernel Density Estimator with Kernel K as

$$KDE_K(x) = KDE(x) = \frac{1}{nh} \cdot \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right).$$

It is easy to see that $KDE(x)$ will give a PDF, i.e., will be nonnegative and will integrate to 1:

$$\int_{-\infty}^{+\infty} KDE(x)dx = \frac{1}{nh} \cdot \sum_{i=1}^{n} \int_{-\infty}^{+\infty} K\left(\frac{x-x_i}{h}\right) dx = \frac{1}{n} \cdot \sum_{i=1}^{n} \int_{-\infty}^{+\infty} K\left(\frac{x-x_i}{h}\right) d\frac{x-x_i}{h} \overset{u=\frac{x-x_i}{h}}{=\!=\!=}$$

[11] The idea of this comes from the following consideration: Assume that our data comes from a realizations of a random variable X, with PDF $f(x)$. We want to estimate f, having that realizations $x_1, ..., x_n$. Now, we have

$$\mathbb{P}\left(x-\frac{h}{2} < X < x+\frac{h}{2}\right) = \int_{x-\frac{h}{2}}^{x+\frac{h}{2}} f(t)dt \approx f(x) \cdot h.$$

Hence,

$$f(x) \approx \frac{\mathbb{P}\left(x-\frac{h}{2} < X < x+\frac{h}{2}\right)}{h},$$

and also

$$\mathbb{P}\left(x-\frac{h}{2} < X < x+\frac{h}{2}\right) \approx \text{The relative frequency of our data in } \left(x-\frac{h}{2}, x+\frac{h}{2}\right] = \frac{\text{Number of datapoints in } \left(x-\frac{h}{2}, x+\frac{h}{2}\right]}{\text{Total Number of datapoints}}.$$

[12] Check this!

$$= \frac{1}{n} \cdot \sum_{i=1}^{n} \int_{-\infty}^{+\infty} K(u)du = \frac{1}{n} \cdot \sum_{i=1}^{n} 1 = 1.$$

Like in the case of the Density histogram, where that histogram was depending on the bins choice, the KDE depends on the choice of $h > 0$. h is called the **bandwidth**, and its estimation is another story.

R CODE, KDE: To construct the KDE for a dataset, one can use **R** command *density*:

```
#Kernel Density Estimator
x <- c(1,1,1,2,2,2,2,3,3,3,3,3,3,3,3,3,4,4,4,5,6,7,8,9,10)
d <- density(x)
plot(d, lwd = 2)
```

The result is given in Fig. 2.17 (and the default kernel **R** is using is Gaussian, you can change that by giving the parameter *kernel*). Note that we have a lot of 3-s in our dataset, so KDE is giving high value close to 3.

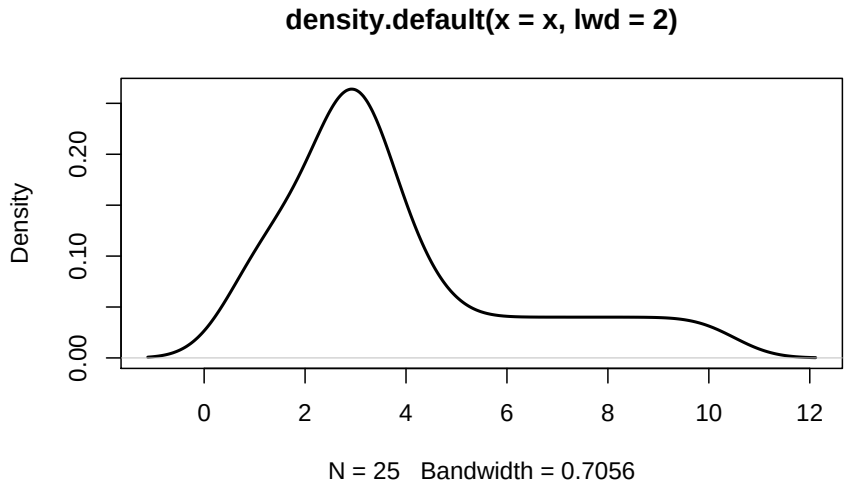


Fig. 2.17: Kernel Density Estimator

R CODE, DENSITY HISTOGRAM VS KDE: Now, lets us draw a Density Histogram vs KDE:

```
#Kernel Density Estimator vs Density Histogram
x <- c(1,1,1,2,2,2,2,3,3,3,3,3,3,3,3,3,4,4,4,5,6,7,8,9,10)
d <- density(x)
hist(x, freq = FALSE, xlim = c(-1,11), ylim = c(0,0.3), col = "blue", main = "")
par(new = TRUE)
plot(d, lwd = 3, col = "red", xlim = c(-1,11), ylim = c(0,0.3), main = "")
```

The result is given in Fig. 2.18.

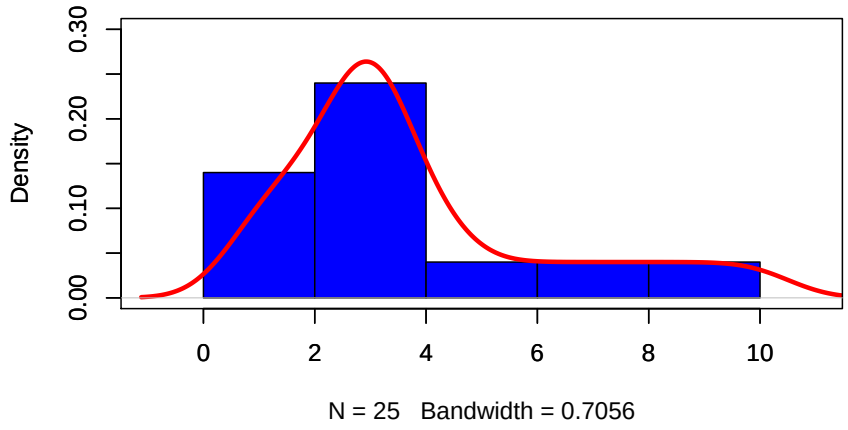


Fig. 2.18: Density Histogram vs KDE

2.7 Empirical Distribution Function

The density histogram and KDE were to estimate the probability density, the PDF of the distribution behind the data. Now we want to estimate the CDF driving the data. And even if we do not have some distribution behind our data, we can construct so kind of CDF function, which will give us another graphical representation for our dataset.

Assume our dataset consists of points $x_1, x_2, ..., x_n$.

Definition 2.9. *The **Empirical Distribution Function, ECDF** or the **Cumulative Histogram** $\mathrm{ecdf}(x)$ of our data $x_1, ..., x_n$ is defined by*

$$\mathrm{ecdf}(x) = \frac{\textit{number of elements in our dataset} \leqslant x}{\textit{the total number of elements in our dataset}} = \\ = \frac{\textit{number of elements in our dataset} \leqslant x}{n} = \frac{\sum_{i=1}^{n} \mathbb{1}(x_i \leqslant x)}{n}, \qquad \forall x \in \mathbb{R}.$$

Example, ECDF: Assume our dataset is:

$$-1, 3, 2, 1, 3, 1, 0.$$

In order to construct the ECDF, first we rearrange these numbers in the increasing order (this will simplify our calculations):

$$-1, 0, 1, 1, 2, 3, 3.$$

Here, n, the total number of elements in our dataset is $n = 7$. Let us calculate first, say, $ecdf(-1.3)$:

$$\mathrm{ecdf}(-1.3) = \frac{\text{number of elements} \leqslant -1.3}{n} = \frac{0}{7} = 0.$$

In fact, if $x < -1$, then $\mathrm{ecdf}(x) = 0$, since no element in our dataset is less than x. Now, if $x = -1$, then

$$\mathrm{ecdf}(-1) = \frac{\text{number of elements} \leqslant -1}{n} = \frac{1}{7},$$

since only one element in our dataset is $\leqslant -1$. Similarly, for any $x \in [-1, 0)$, we will have $ecdf(x) = \frac{1}{7}$. In the same vain we can obtain the value of our ECDF at any point:

$$ecdf(x) = \begin{cases} 0, & \text{if } x < -1 \\ \frac{1}{7}, & \text{if } -1 \leqslant x < 0 \\ \frac{2}{7}, & \text{if } 0 \leqslant x < 1 \\ \frac{4}{7}, & \text{if } 1 \leqslant x < 2 \\ \frac{5}{7}, & \text{if } 2 \leqslant x < 3 \\ 1, & \text{if } x \geqslant 3 \end{cases}$$

Remark, ECDF and Discrete r.v. CDF: In some sense, the ECDF is the CDF of the r.v. X which takes the value x_i with the probability $\frac{1}{n}$ (if two or more values x_i coincide, then one needs to count every occurrence of x_i). In other way, we make a r.v. X with values x_i (taking only distinct values x_i), and with probabilities

$$\mathbb{P}(X = x_i) = \text{the relative frequency of } x_i.$$

Say, for the example above, for the dataset

$$-1, 3, 2, 1, 3, 1, 0$$

we define the following r.v. X:

Values of X	-1	0	1	2	3
$\mathbb{P}(X = x)$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{2}{7}$	$\frac{1}{7}$	$\frac{2}{7}$

since we have $n = 7$ datapoints, and -1, 0 and 2 appear only once, and 1 and 3 appear twice in our dataset. Now, the Empirical CDF of our dataset $ecdf(x)$ will coincide with the CDF $F_X(x)$ of X:

$$ecdf(x) = F_X(x), \qquad x \in \mathbb{R}.$$

Example, ECDF and Discrete r.v. CDF: The Empirical CDF for the dataset $1, 5, 5, 9$ is the same as the CDF for the r.v. X with values $1, 5, 9$ and with probabilities

$$\mathbb{P}(X = 1) = \frac{1}{4}, \qquad \mathbb{P}(X = 5) = \frac{2}{4}, \qquad \mathbb{P}(X = 9) = \frac{1}{4}.$$

i.e., for the r.v. X with the PMF

Values of X	1	5	9
$\mathbb{P}(X = x)$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{1}{4}$

R CODE, ECDF: The R Code to graph the ECDF for the data above is

```
#Empirical CDF
x <- c(-1, 3 , 2, 1, 3, 1, 0)
f<-ecdf(x)
plot(f)
```

The result is given in Fig. 2.19.

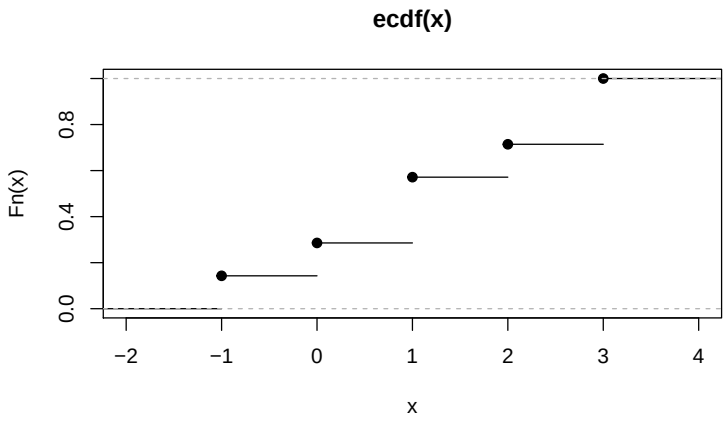


Fig. 2.19: Empirical CDF

The remarkable property of ECDF $ecdf(x)$ is that if our dataset $x_1, ..., x_n$ is a sample obtained from a r.v. with the CDF $F(x)$, then $ecdf(x)$ is a good estimate (approximation) for $F(x)$, and, in fact, $ecdf(x)$ approaches to $F(x)$[13]. Below I am giving one example, and you can try to run that for different distributions, and for different (increasing) number of sample points:

R CODE, ECDF APPROXIMATION OF THE THEORETICAL CDF:

```
#Empirical CDF approximation of Theoretical CDF
#First we plot the CDF of the Standard Normal Distribution
plot(pnorm, lwd = 3, col = 'red', xlim = c(-3,3), ylim = c(0,1), ylab = "ecdf and CDF")
x <- rnorm(30) #Now we are taking a sample of size 30 from N(0,1)
f<-ecdf(x) #f will be the ECDF of our data x
par(new = TRUE) #this is to keep the previous graph and to draw over it
plot(f, xlim = c(-3,3), ylim = c(0,1), ylab = "ecdf and CDF")
```

The result is presented in Fig. 2.20. Try to rerun several times (since every time **R** will generate a new sample), increase the number of sample points, and try for different distributions!

Later we will talk about BoxPlots, another graphical Representation (Description) of our data.

[13] This is the famous **Glivenko-Cantelli Theorem**.

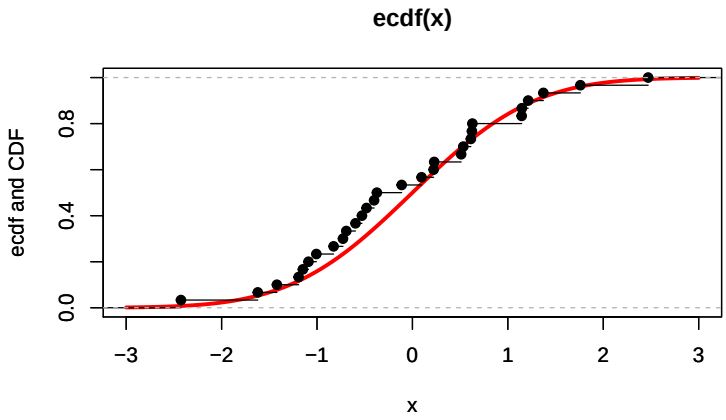


Fig. 2.20: Empirical CDF approximation of the $\mathcal{N}(0,1)$ CDF

Exploratory Data Analysis for Univariate Data: Numerical Summaries

Usually, one uses the word *Statistics* with three (maybe more?) meanings:

- Statistics is a Course, Scientific Subject name, topic we are studying at Universities
- Statistics is a numerical characteristic for a numerical dataset
- Statistics is a r.v. obtained from a random sample (will be discussed later)

Here we want to give some numerical summaries, characterizations for our data. The previous part of our course was devoted to graphical representation of the data, and here we want to talk about some numerical representation for our data.

Definition 3.1. *Given a numerical dataset* $x_1, ..., x_n$*, we will call a* ***Statistics*** *for that dataset any function of* $x_1, ..., x_n$.

Well, of course, this is a formal definition, and in the reality we will choose some meaningful and descriptive Statistics for our dataset, to get some useful information about our data.

Concerning the notations, I will use the following convention: in our Probability course, I was denoting by capital letters r.v.'s, say X was a r.v.. Here, in this section, by x I will denote a dataset - a collection of real numbers, values of some variable. Say, if I have a dataset $x_1, x_2, ..., x_n$, I will say that we have a dataset x. If y is another dataset, then y consists of some numbers (or 2D points) $y_1, y_2, ..., y_m$ etc.

In our Probability course, for a r.v. X we have defined some numerical characteristics: e.g.,

$\mathbb{E}(X)$ - the Expected Value or the Expectation of X;

$Var(X)$ - the Variance of X;

$SD(X)$ - the Standard Deviation,

and for a pair of r.v. we have defined

$Cov(X, Y)$ - the covariance between X and Y;

$Cor(X, Y)$ - the correlation coefficient between X and Y etc.

For numerical datasets we will define the analogous characteristics, and we will use small letters, and denote the corresponding quantities as $var(x)$, $sd(x)$ or $cov(x, y)$, say.

In this part we will talk only about univariate numerical data, about observations concerning one feature. In the next part, we will consider some numerical characteristics for the relationship of two numerical datasets.

3.1 Order Statistics (Ranks)

First, we want to introduce a useful notion and notation:

Definition 3.2. *For a dataset $x_1, x_2, ..., x_n$, let $x_{(j)}$ be the jth sample value, when our data is ordered in the increasing order. Then $x_{(j)}$ is called the* ***j-th order statistics*** *of our dataset.*

In other words, to obtain the j-th order statistics, one needs to arrange our data into the increasing order, and then calculate the j-th element in this arranged set[1].

So, by the definition, the dataset $x_{(1)}, x_{(2)}, ..., x_{(n)}$ coincides with $x_1, ..., x_n$, and

$$x_{(1)} \leqslant x_{(2)} \leqslant ... \leqslant x_{(n-1)} \leqslant x_{(n)}.$$

In particular,

$$x_{(1)} = \min\{x_1, x_2, ..., x_n\} \qquad \text{and} \qquad x_{(n)} = \max\{x_1, x_2, ..., x_n\}.$$

EXAMPLE, ORDER STATISTICS: Here is a sample showing dayly number of informative emails (that I am reading and replying to) in my AUA acoount:

$$10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6.$$

Now, in order to find, say 7th order statistic, we need to sort our data in the increasing order:

$$1, 2, 2, 2, 2, 4, 5, 5, 6, 6, 7, 7, 8, 8, 10, 10.$$

The 7th order statistic is the 7th element in this sorted list from the left: $x_{(7)} = 5$. Similarly, $x_{(1)} = 1$, $x_{(2)} = x_{(3)} = x_{(4)} = x_{(5)} = 2$, $x_{(6)} = 4$, $x_{(11)} = x_{(12)} = 7$, etc.

Computational Challenge: Given a numerical dataset $x_1, x_2, ..., x_n$,

- Sort the dataset
- for a given j, find the j-th order statistic of that dataset

R CODE, ORDER STATISTICS: In order to find the j-th order statistics in the dataset in **R**, one can use the *sort* command. Say, I want to find the 7th order statistics in the dataset above:

```
#Order Statistics, My Emails Data
my.emails <- c(10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6)
my.emails.sorted <- sort(my.emails)
my.emails.sorted[7]
```

3.2 Statistics for the central tendency

Here we want to give some measures, estimates of location for the typical observed value, estimates for the location of the center of the data. We will give different statistics for that.

[1] counting from the left to the right ☺

3.2.1 The Sample Mean

Definition 3.3. *For a sample* $x_1, x_2, ..., x_n$*, the sample mean is*

$$\bar{x} = mean(x) = \frac{x_1 + x_2 + ... + x_n}{n}.$$

Geometric Interpretation: If we will put our data into the real line, and if we will put equal weights at that points, then the sample mean or the average of our set is the equilibrium position of that system, of that real line with the weights at our data points.

EXAMPLE, SAMPLE MEAN: Consider the dataset x:

$$1, 1, 2, 3, -4, 5.$$

The Sample Mean for dataset is

$$\bar{x} = mean(x) = \frac{1+1+2+3+(-4)+5}{6} = \frac{4}{3}.$$

EXAMPLE, SAMPLE MEAN: Looking at my emails dataset above, if I want to give someone the idea about the number of emails I am receiving daily, it will not be a good idea to give the daily email numbers, say, "one day I have received 10 emails, the other day I have received again 10 email, and for the next day the number of emails was 2" etc. Instead, if I will calculate the Sample Mean for my email number dataset:

$$mean(emails) = \frac{10+10+2+5+7+5+1+2+7+2+6+8+2+4+8+6}{16} = \frac{85}{16} = 5.3125$$

and now I can describe: the average number of my daily emails is 5.3.

By the way, sometimes, when stating some average numbers, that can be a little bit confusing. Say, one is stating that the average number of children in families in a country is 1.8. This can be confusing at the first sight, since the number of children cannot be non-integer. But you need to take this as the example above: the average number of emails is 5.3, but I will never get 5.3 emails (well, I will, if the internet connection will be interrupted during the 6th email download process ☺).

R CODE, SAMPLE MEAN, DAILY EMAILS DATASET: Here is the **R** code to calculate the Sample Mean for the Daily Emails dataset:

```
#Sample Mean Calculation
x <- c(10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6)
mean(x)
```

REMARK, IDEA BEHIND THE SAMPLE MEAN: The idea of the Sample Mean is very intuitive, so maybe there is no necessity to give the idea behind that. But let me give an explanation to make a link between the Probability Theory and Sample Statistics. And you will find this type of links also in the rest of this part.

Assume we are given a numerical dataset $x_1, x_2, ..., x_n$. We then make a random variable X out of this dataset, by giving equal probability to each of this data points. We have made the same thing when considering the Empirical CDF. Say, if our dataset is $-1, 1, 1$, then we make a r.v. X with

Values of X	-1	1
$\mathbb{P}(X = x)$	$\frac{1}{3}$	$\frac{2}{3}$

Now, the Sample Mean of the dataset x is just the Expected Value of our r.v., $\mathbb{E}(X)$. Say, for the above example,

$$\mathbb{E}(X) = \frac{1}{3} \cdot (-1) + \frac{2}{3} \cdot 1 = \frac{-1+1+1}{3} = mean(x).$$

Now, if our dataset x is given by the frequency table: the frequency of the value x_k is f_k, for $k = 1, 2, ..., m$, then the Sample Mean of x can be calculated by

$$\bar{x} = mean(x) = \frac{f_1 \cdot x_1 + f_2 \cdot x_2 + ... + f_m \cdot x_m}{f_1 + f_2 + ... + f_m}.$$

To see why this formula is true, we just need to "recover" the whole dataset using the frequency table: our dataset will be

$$\underbrace{x_1, x_1, ..., x_1}_{f_1 \text{ times}}, \underbrace{x_2, x_2, ..., x_2}_{f_2 \text{ times}}, ..., \underbrace{x_m, x_m, ..., x_m}_{f_m \text{ times}}.$$

EXAMPLE, SAMPLE MEAN BY FREQUENCIES: Considering again my emails dataset (in the sorted form),

$$1, 2, 2, 2, 2, 4, 5, 5, 6, 6, 7, 7, 8, 8, 10, 10$$

we can write it in the frequency table form:

No. of daily emails	1	2	4	5	6	7	8	10
Frequency	1	4	1	2	2	2	2	2

So the mean number of my daily emails can be calculated by the formula above:

$$mean(emails) = \frac{1 \cdot 1 + 4 \cdot 2 + 1 \cdot 4 + 2 \cdot 5 + 2 \cdot 6 + 2 \cdot 7 + 2 \cdot 8 + 2 \cdot 10}{1 + 4 + 1 + 2 + 2 + 2 + 2 + 2} = \frac{85}{16},$$

which gives, of course, the same result as above.

R CODE, SAMPLE MEAN BY FREQUENCIES: To calculate the Sample Mean by Frequencies, we can use the following two **R** codes:

```
#Sample Mean Calculation with Frequencies, v. 1
```

```
x.unique <- c(1, 2, 4, 5, 6, 7, 8, 10) #Unique x values
x.freq <- c(1, 4, 1, 2, 2, 2, 2, 2) #The corresponding Frequencies
mean_by_freq <- (x.unique%*%x.freq)/sum(x.freq) #The dot product over the sum of frequencies
mean_by_freq <- as.numeric(mean_by_freq) #Transforming the result to a number
```

Here the command $a\%*\%b$ calculates the dot product of a and b. In our case, we calculate the dot product of two vectors $x.unique$ and $x.freq$, and the result is a 1×1 matrix. Try to run the command $x.unique\%*\%x.freq$. To convert it to a number, we use the $as.numeric$ command.

The second version is more straightforward:

```
#Sample Mean Calculation with Frequencies, v. 2
x.unique <- c(1, 2, 4, 5, 6, 7, 8, 10)
x.freq <- c(1, 4, 1, 2, 2, 2, 2, 2)
mean_by_freq <- sum(x.unique*x.freq)/sum(x.freq)
```

Here the command $x.unique * x.freq$ returns a vector of corresponding elements products, in the above terms, it returns the vector $(f_1 \cdot x_1, f_2 \cdot x_2, ..., f_m \cdot x_m)$. We calculate the sum of the elements of this vector by $sum(x.unique * x.freq)$.

The Sample Mean is very easy to calculate, and, of course, usually serves as a good representation for the typical value, the location for the center for our data. But, unfortunately, it has a drawback, weak side: it is sensitive to extreme (very large or very small) values, to outliers.

EXAMPLE, SAMPLE MEAN SENSITIVITY TO OUTLIERS: We have seen above that I can say that typically, I am replying to 5.3 emails daily. This will give a good sense of how busy am I[2] ☺ Now, assume that the next day of my above observation, I have received 100 emails. Now, if I will use this number too, then the number of daily emails will be

$$10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6, 100,$$

and the Sample Mean will be

$$mean(\text{daily emails}) = \frac{10+10+2+5+7+5+1+2+7+2+6+8+2+4+8+6+100}{17} \approx 10.88$$

And if I will state that my daily email number is almost 11, that will not give the correct picture. And this is because of just one enormously email-busy day!

EXAMPLE, SAMPLE MEAN SENSITIVITY TO OUTLIERS: Assume you apply to a newly established programming company, and you ask about the mean salary in that company. And you get the response that the mean salary is 300,000AMD. And you are happily thinking that if you will be their employee, then you will get something around 300,000. Well, plus/minus something, of course, based on the experience. But, let us see that the number can be misleading: assume the company has 21 employees, and 20 of them are receiving 165,000 and one gets 3,000,000. Then the mean will be, using the frequency form calculation,

$$mean(salary) = \frac{20 * 165,000 + 3,000,000}{20+1} = 300,000.$$

But the real picture is that almost all employees, except the boss, get 165,000.

So, when using the Sample Mean, please take an attention to outliers.

In fact, there are some methods trying to solve the problem of outliers sensitivity of the Sample Mean: the following methods are more robust in the sense of sensitiveness to extremes.

Trimmed (Truncated) Sample Mean: First we choose a natural number p, satisfying $2p < n$. Next we sort our data in the increasing order, we drop p lowest and p highest values[3], and then we calculate the sample mean of the rest of the data:

$$\text{trimmed sample mean}(x) = \bar{x}_{trimmed} = \frac{x_{(p+1)} + x_{(p+2)} + ... + x_{(n-p-1)} + x_{(n-p)}}{n-2p} = \frac{\sum_{k=p+1}^{n-p} x_{(k)}}{n-2p}.$$

This trimmed means eliminates the influence of extreme values.

Remark, Trimmed Mean: Usually, one is giving the proportion of elements to be dropped from the beginning and end of the sorted array. So one gives $r \in [0, 0.5)$. In that case we calculate the number of elements to be dropped, $p = [r \cdot n]$, and then do our trimming.

Example, Trimmed Sample Mean: Say, sometimes, when calculating the score of some international competition, one drops the highest and lowest values of the grades of judges, and then takes the trimmed mean (because one of the judges can give very high grades for participants from its own country or very low grades for "not-so-friendly" country participants).

See, for example, how the Diving Competition is scored at https://en.wikipedia.org/wiki/Diving#Scoring_the_dive.

Example, Trimmed Sample Mean: When doing grading for quizzes at AUA, we are usually dropping the lowest grade - this is some kind of half-trimming, since we are not dropping the highest grade (fortunately ?) ☺.

R code, Trimmed Sample Mean: Here is the Code to calculate the Trimmed Mean:

```
#Trimmed Mean Calculation
x <- c(10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6)
mean(x, trim = 0.2)
```

The parameter `trim` can take values from 0 to 0.5 (the default is 0), and shows the fraction of observations to be trimmed from each end of the dataset before the mean is computed.

[3]Sometimes people take a number p from 1 to 100 and drop the lowest and highest p% of the data.

Winsorized Sample Mean: Another variation is the Winsorized Mean - we sort our data in the increasing order, and, for a fixed number p, we replace the first (i.e., smallest) p numbers by $x_{(p+1)}$, and the last p numbers (i.e., largest p numbers) by $x_{(n-p-1)}$. Then we calculate the sample mean of the obtained dataset:

$$\text{winsorized mean}(x) = \frac{x_{(p+1)} + x_{(p+1)} + ... + x_{(p+1)} + x_{(p+1)} + x_{(p+2)} + ... + x_{(n-p-2)} + x_{(n-p-1)} + x_{(n-p-1)} + ... +}{n}$$

$$= \frac{(p+1)\cdot x_{(p+1)} + \sum_{k=p+2}^{n-p-2} x_{(k)} + (p+1)\cdot x_{(n-p-1)}}{n}.$$

REMARK, WINSORIZED MEAN: Here also, as in the Trimming case, one is giving the proportion of elements to be replaced from the beginning and end of the sorted array. So one gives $r \in [0, 0.5)$. In that case we calculate the number of elements to be replaced, $p = [r \cdot n]$, and then do our calculations.

REMARK, WINSORIZED MEAN: The difference between Trimming and Winsorizing is that in the latter case we calculate the mean of the same number of elements, n, as in the original Dataset x.

Weighted Sample Mean: Assume we want to calculate the mean of the dataset $x_1, x_2, ..., x_n$. We take nonnegative weights w_k's, such that $\sum_{k=1}^{n} w_k \neq 0$, and we calculate

$$\text{weighted sample mean} = \bar{x}_w = \frac{\sum_{k=1}^{n} w_k x_k}{\sum_{k=1}^{n} w_k}.$$

The weight of data x_k is then $\frac{w_k}{\sum_{i=1}^{n} w_i}$. This is to give more weight to some data, and give less weight to some others (if, say, we are unsure in the correctness of some that data). Say, if we are collecting data from different sources, and we trust some of our sources and not too much to others, we can give larger weights to the results from the trusted sources and small weights to the others (if we do not want to completely dismiss the results/observations of the latters). Another situation is, for example, when calculating the mean daily price of some stock, to make predictions about the price in the future, one can calculate the weighted mean of daily prices, giving more weights to recent information, and less weight to the old information. The idea is that we think that recent prices contain more information about the price, useful for prediction, rather than the old prices. But, of course, old prices do contain *some* information, so we do not want to dismiss them completely.

The Weighted Sample Mean is not helping us too much concerning the extremes sensitiveness, but is helping when we want to make a difference between the datapoints.

R CODE, WEIGHTED MEAN: To calculate the Weighted Sample Mean of a dataset x with weights w, one can use the **R**'s function *weighted.mean()*:

```
#Weighted Arithmetic Mean
x <- c(-1, 0, 3, 2, -2, 3, 2, 3, 2, 3)
w <- c( 2, 2, 2, 1, 1,  1, 2, 0, 1, 5)
weighted.mean(x, w)
```

Here **R** will produce 1.647059. To check the result is true, we can try

```
sum(x*w)/sum(w)
```

which will give the same result.

REMARK, ON THE DEFINITION OF THE SAMPLE MEAN: Assume we have a dataset $x : x_1, x_2, ..., x_n$. For any point $a \in \mathbb{R}$ we ca consider the differences $x_k - a$, the deviations from the point a. It is easy to see that the Sample Mean $mean(x)$ is the only point with the property that the sum of all deviations is 0, i.e.

$$a = mean(x) \qquad \Leftrightarrow \qquad \sum_{k=1}^{n} \Big(x_k - a\Big) = 0.$$

REMARK, THEORETICAL AND SAMPLE MEANS AS MINIMIZERS: Hope everybody remembers that the Theoretical Mean of a distribution (or a r.v. X with that distribution) is defined as the Expected Value of the r.v. X.

Then one can prove the following assertions:

a. m is the Mean of the r.v. X if and only if

$$m \in \underset{a\in\mathbb{R}}{\operatorname{argmin}}\, \mathbb{E}\Big((X-a)^2\Big);$$

b. m is the mean of the dataset $x_1, x_2, ..., x_n$ if and only if

$$m \in \underset{a\in\mathbb{R}}{\operatorname{argmin}}\, \frac{1}{n}\cdot\sum_{k=1}^{n}(x_k-a)^2.$$

Of course, here we assume that the expectation above is defined.

3.2.2 The Sample Median

The other measure of central tendency, "central value" of the dataset, more robust to outliers than the Sample Mean, is the **Median** of a dataset. The Median is a point (not necessarily from our dataset) such that half of the observations (datapoints) are less than or equal to that number, and half of the observations are greater than or equal to that number. In fact, different authors use different definitions, but the idea is the one above. The point is that when we have an odd number of datapoints, then we can find a unique datapoint dividing our dataset into two halves (that datapoint itself is included in both halves). So there is no ambiguity in the definition of the Median in this case.

EXAMPLE, MEDIAN OF A DATASET, ODD NUMBER OF ELEMENTS: Assume our dataset is

$$1, 3, 2, 1, 2, 1, 2, 3, 4, 5, 2$$

First we sort our dataset:

$$1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 5$$

Now, the sixth element in this sorted list (one of the 2's) divides our sorted dataset into two equal parts:

$$1,1,1,2,2,\color{red}{2},2,3,3,4,5$$

(there are 6 elements less than or equal to the red 2, and 6 elements larger than or equal to the same number). So the median is 2 for this dataset.

But for the case when we have an even number of datapoints, then we can have different approaches.

EXAMPLE, MEDIAN OF A DATASET, EVEN NUMBER OF ELEMENTS: Assume our dataset is

$$1,3,2,1$$

We want to find a point (number) such that half of the datapoints are less than or equal to that point, and half of the observations are greater than or equal to that point. Again, we sort our dataset:

$$1,1,2,3$$

Now, any point between 1 and 2 will have the above property. Say, half of the datapoints are $\leqslant 1.3$ and half of the datapoints are $\geqslant 1.3$. So we can use 1.3 as a Median for this dataset. Similarly, 1.4 will work as well. Usually, people take the midpoint of these numbers, 1.5.

Note here that we do not require our Median to be a datapoint, to be in our dataset!

Now, one of the widely used definitions of the Median is the following (and we will use this one in the rest of the text):

Definition 3.4. *The Sample Median* $\mathtt{median}(x)$ *of the dataset* $x_1, x_2, ..., x_n$ *is the number dividing the sorted dataset*

$$x_{(1)}, x_{(2)}, ..., x_{(n)}$$

to two equal parts, more precisely:

$$\textit{If } n \textit{ is odd}, \quad \mathtt{median}(x) = x_{\left(\frac{n+1}{2}\right)};$$

$$\textit{If } n \textit{ is even}, \quad \mathtt{median}(x) = \frac{1}{2} \cdot \left(x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}\right).$$

EXAMPLE, SAMPLE MEDIAN: Continuing to consider the number of daily emails dataset,

$$10,10,2,5,7,5,1,2,7,2,6,8,2,4,8,6.$$

let us calculate the median of it. To that end, we first sort our data in the increasing order:

$$1,2,2,2,2,4,5,5,6,6,7,7,8,8,10,10.$$

Here we have $n = 16$ observations, an even number of observations, so, by our definition above,

$$\mathtt{median}(\text{daily emails}) = \frac{1}{2} \cdot \left(x_{(8)} + x_{(9)}\right) = \frac{1}{2} \cdot (5+6) = 5.5$$

So I can state that the "average" number of emails I am receiving daily is 5.5.

R code, Sample Median, Emails Dataset: Here is the code to calculate the Sample Median for the daily emails dataset:

```
#Sample Median Calculation
x <- c(10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6)
median(x)
```

Once again, the Sample Median is a number dividing our set of observations into two equal-length parts. On the Fig. 3.1 you can see on red the Sample Median - we have 50 data points here (in Black), and 25 of them are to the left of the Median and 25 of them are to the right.

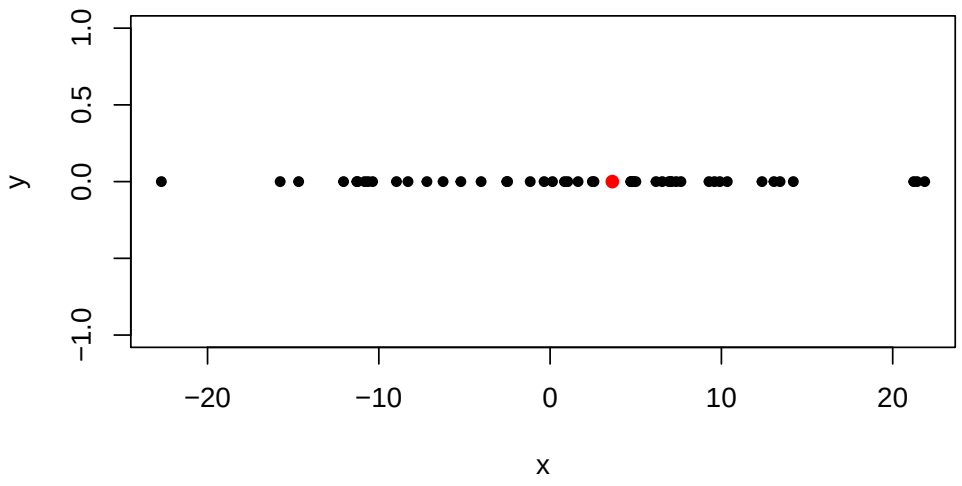


Fig. 3.1: Sample Median Example: Here we have a dataset of 50 points (in Black). The red point is the Median.

R code, Sample Median: Fig. 3.1 is obtained by running the code:

```
#Sample Median calculation
x <- rnorm(50, mean = 2, sd = 10)
m <- median(x)
y <- rep(0,50)
plot(x,y, pch = 20, cex = 1.1)
points(m,0, pch = 16, col = "red", cex = 1.1)
```

Here the command $\texttt{rnorm}(n, \texttt{mean} = a, \texttt{sd} = b)$ is generating a sample of size n from the Normal Distribution with a mean a and Standard Deviation b, i.e., we are getting 50 possible realizations of $X \sim \mathcal{N}(a, b^2)$. The command $\texttt{plot}(x, y)$ runs as follows: if we have 2 vectors of the same size $x = (x_1, x_2, ...)$ and $y = (y_1, y_2, ...)$, it draws the corresponding points (x_i, y_i). Here we use $y < -\texttt{rep}(0, 50)$: the command rep stands for "replicate", it copies 0 fifty times. So in the result, y will be the zero vector of size 50. Try to run, say, $\texttt{rep}(\texttt{c}(1, 2), 10)$ - this will copy 1,2 ten times. Going

back to our goats, $\texttt{plot}(x,y)$ will draw the points $(x_i,0)$ for all i. $\texttt{pch}$ is for point character - try changing the value to see the effect, and $\texttt{cex}$ is for the point size - again play with the values to see the effect.

It can be seen, that Sample Median is not affected by outliers, by extreme observations.

EXAMPLE, SAMPLE MEDIAN: If we will consider the above example with fake email numbers dataset,

$$10,10,2,5,7,5,1,2,7,2,6,8,2,4,8,6,100,$$

or, in the sorted form,

$$1,2,2,2,2,4,5,5,6,6,7,7,8,8,10,10,100,$$

then the Sample Median will be (we have $n=17$ observations)

$$\texttt{median}(\text{daily emails}) = x_{(\frac{17+1}{2})} = x_{(9)} = 6,$$

which is much more relevant (realistic ?) than the one obtained using the Sample Mean.

Here let us compare the Sample Median vs the Sample Mean: the good point of the Sample Median is that it is not affected by very large or small values, by outliers, but the Sample Mean is affected. The bad news is that Sample Median is not taking into account the actual values of our observations, rather than their ordering, but the Sample Mean is taking into account all values. So you need to take these pros/cons when using Sample Medians and Means to describe your dataset.

REMARK, SAMPLE MEDIAN DEFINITION: Sometimes people use another definition of the sample median. As we have seen above, our definition of the Median can produce a number which is not in our dataset. In some cases this is not acceptable, so one uses another definition of the Median producing a datapoint, a number in our dataset. For example, one can define (for any case of n),

$$\texttt{median}(x) = x_{(\lceil \frac{n}{2} \rceil)},$$

where $\lceil a \rceil$ is the smallest integer $\geqslant a$.

REMARK, THEORETICAL MEDIAN: One can also define the theoretical median of a distribution. Assuming X is a r.v. with a CDF $F(x)$ (and, in the case if X is continuous, with a PDF $f(x)$), we call a number $m \in \mathbb{R}$ to be a median of that distribution, if

$$\mathbb{P}(X \leqslant m) \geqslant \frac{1}{2} \qquad \mathbb{P}(X \geqslant m) \geqslant \frac{1}{2}.$$

In other words, median is a number m such that the probability that $X \leqslant m$ and $X \geqslant m$ are not less that 50%.

It is easy to see that if our distribution is continuous with PDF $f(x)$, then m is a median if and only if

$$F(m) = \frac{1}{2} \qquad \text{or, equivalently,} \qquad \int_{-\infty}^{m} f(x)dx = \int_{m}^{+\infty} f(x)dx = \frac{1}{2}.$$

EXAMPLE, THEORETICAL MEDIAN: Here let us find the Theoretical Median for the distribution with the PDF

$$f(x) = \begin{cases} 2x \cdot e^{-x^2}, & x \geqslant 0 \\ 0, & x < 0. \end{cases}$$

We have a continuous distribution here, and in this case we will calculate the CDF $F(x)$ (which is easy to do for this example) and solve $F(m) = \frac{1}{2}$ to find the median m. From the Probability course, we have

$$F(x) = \int_{-\infty}^{x} f(t)dt = \begin{cases} 0, & x < 0 \\ 1 - e^{-x^2}, & x \geqslant 0. \end{cases}$$

Clearly, we cannot have $F(m) = \frac{1}{2}$ for $m < 0$. So in our case, $m \geqslant 0$. Then we need to have

$$F(m) = 1 - e^{-m^2} = \frac{1}{2},$$

yielding $m = \sqrt{\ln 2}$.

If we will interpret this geometrically, then:

- For the PDF $f(x)$, m is a point on the OX axis such that the line $x = m$ divides the area under the graph of $y = f(x)$ into two equal parts: the area under f is exactly 0.5 for $x \in (-\infty, m]$ and exactly 0.5 for $x \in [m, +\infty)$, see Fig. 3.2
- For the CDF $F(x)$, m is an intersection point of the graph of $y = F(x)$ and the horizontal line $y = \frac{1}{2}$, see Fig. 3.3

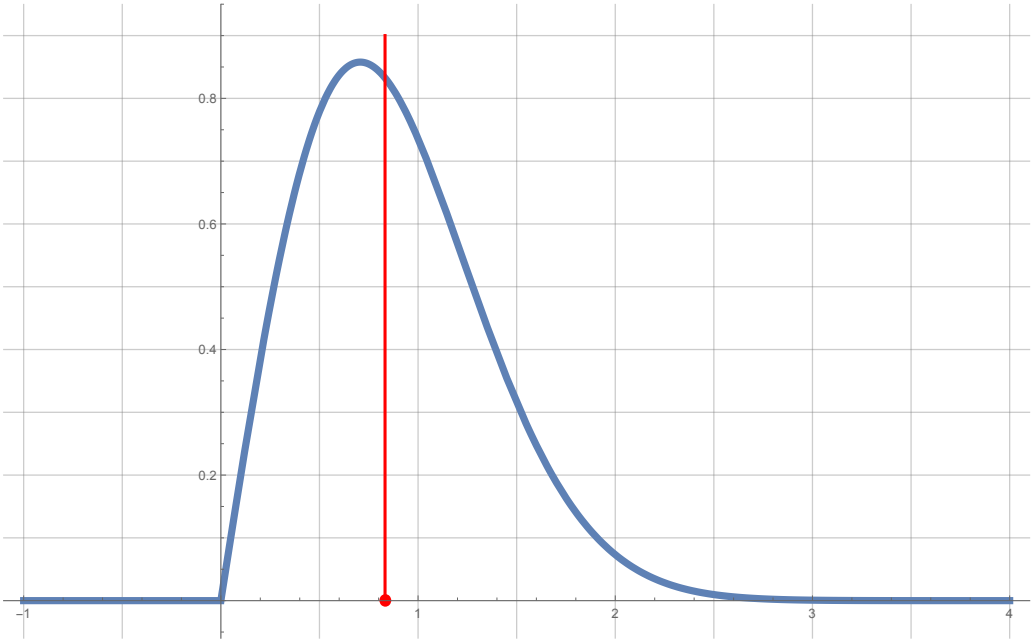


Fig. 3.2: Theoretical Median Example: Here we have the PDF graph. The line passing through the Median (red point on the OX axis) divides the area under the graph of PDF into two equal parts.

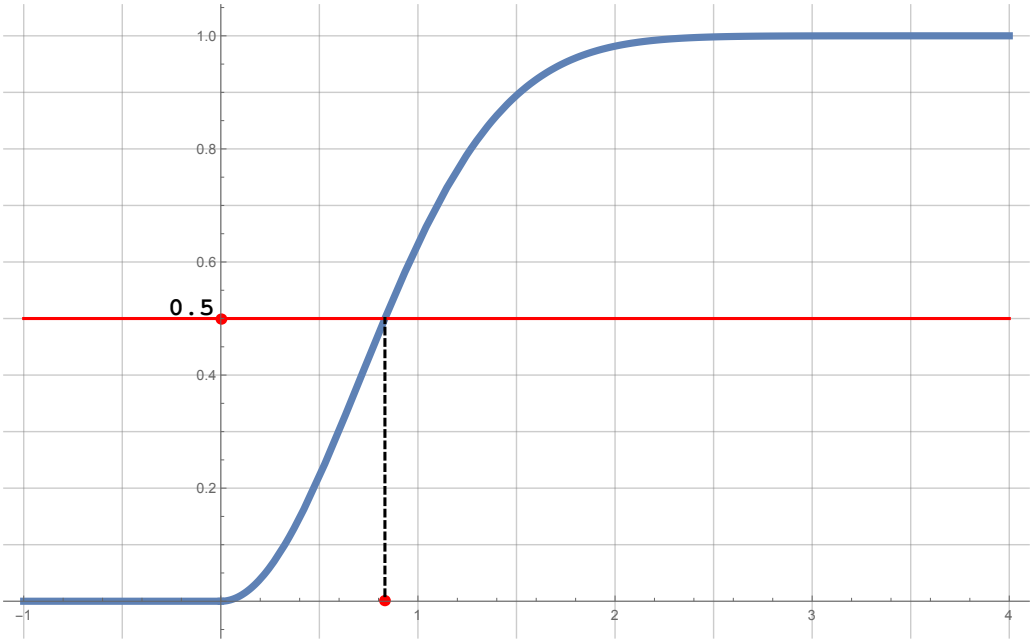


Fig. 3.3: Theoretical Median Example: Here we have the CDF graph. Median lies on the intersection of the graph of the CDF and the horizontal line $y = 0.5$.

EXAMPLE, THEORETICAL MEDIAN: If we will try to find the Theoretical Median for the distribution with the PDF

$$f(x) = \begin{cases} 0.5, & x \in [-1, 0] \cup [2, 3] \\ 0, & \text{otherwise.} \end{cases}$$

then we will have infinitely many points m, such that the vertical line $x = m$ is dividing the area under the PDF into two equal-size parts. In fact, any point $m \in [0, 2]$ will work. So in this case we have infinitely many Theoretical Medians, see Fig. 3.4.

This can be seen also by the graph of the CDF also: the intersection of CDF with the horizontal line $y = 0.5$ is the whole interval $[0, 2]$, see Fig. 3.5.

REMARK, THEORETICAL AND SAMPLE MEDIANS AS MINIMIZERS: Try to prove the following assertions:

a. m is a median of the r.v. X (i.w., of the theoretical distribution behind X) if and only if

$$m \in \operatorname*{argmin}_{a \in \mathbb{R}} \mathbb{E}(|X - a|)$$

b. If m is a median of the dataset $x_1, x_2, ..., x_n$, then

$$m \in \operatorname*{argmin}_{a \in \mathbb{R}} \frac{1}{n} \cdot \sum_{k=1}^{n} |x_k - a|. \tag{3.1}$$

Inversely, if m satisfies (3.1), then m divides our dataset into two equal-length parts.

Again, here we assume that the expectation above is defined.

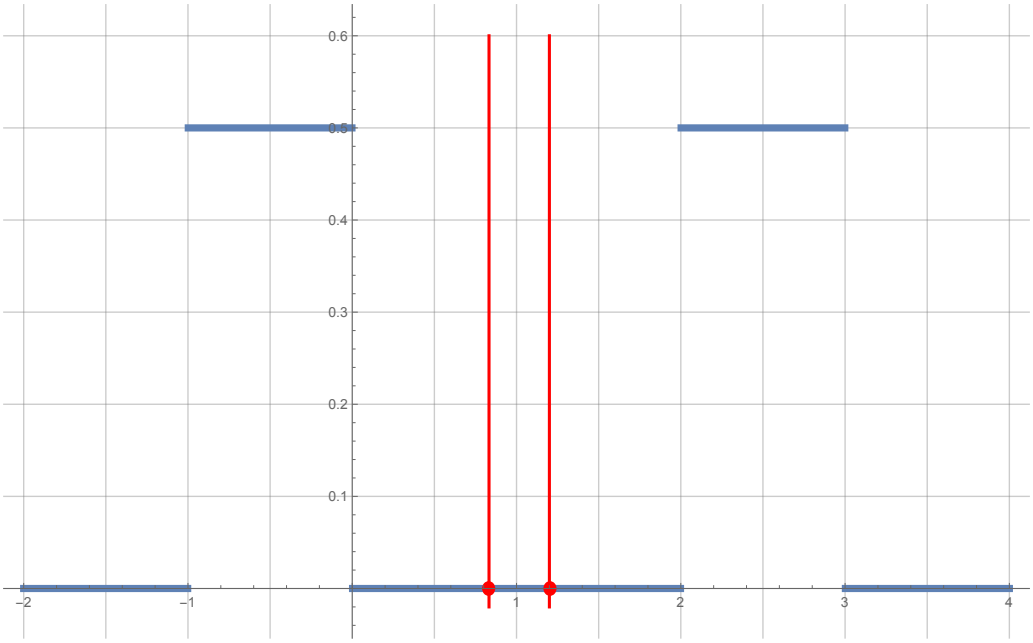


Fig. 3.4: Theoretical Median Example: Here we have the PDF graph. There are infinitely many points on the OX such that the vertical line passing through that points (say, 2 red points on the OX axis) divides the area under the graph of PDF into two equal parts.

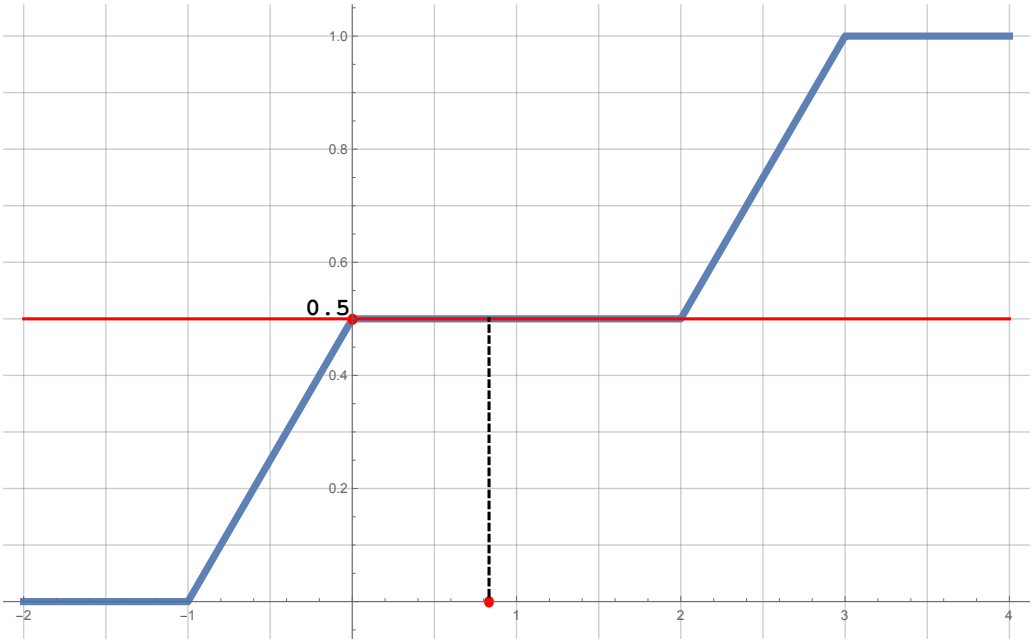


Fig. 3.5: Theoretical Median Example: Here we have the CDF graph. Median lies on the intersection of the graph of the CDF and the horizontal line $y = 0.5$. We have here that the intersection is the interval $[0, 2]$, and just one of the medians is shown

REMARK, SAMPLE MEDIAN AND ECDF: Assume we have a dataset x, with data points $x_1, x_2, ..., x_n$. Then, as above, we can construct the r.v. X, taking the values x_k, with probabilities $\frac{1}{n}$ (if some x_k-s coincide, we add the corresponding probabilities).

Vnimanie, the question: Find the relation between the CDF of X, the ECDF of the dataset, the Theoretical Median of the distribution of X and the Sample Median.

Remark, Sample Median, use in DIP: See Gonzalez, Woods, Digital Image Processing, 3rd Ed, part 3.5.2, 5.3.2. Also find a lot of statistics in this books!

3.2.3 The Sample Mode

Another representative for the dataset is its most frequent element(s):

Definition 3.5. *The **Sample Mode** of the dataset is the value which occurs most frequently in our dataset.*

Example, Sample Mode: Let us consider the dataset of daily emails,

$$10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6.$$

To find the mode, it is convenient first to form the frequency table:

No. of daily emails	1	2	4	5	6	7	8	10
Frequency	1	4	1	2	2	2	2	2

From this table it is clear that the most frequent value is 2 (the frequency is 4, the highest), so

$$\mathrm{mode}(emails) = 2.$$

Remark, Sample Mode: We can define the Sample Mode of the Dataset x as:

$$\mathrm{mode}(x) \in \underset{x}{\mathrm{argmax}} \Big(\mathrm{Frequency}(x)\Big).$$

It is important that the Sample Mode can be non-unique. For example, the dataset $1, 2, 2, 3, 4, 4$ has two modes: 2 and 4.

In the case if we have a discrete dataset, where every value is unique, then we will say that all elements are modes of that dataset. For example, all elements of the dataset $1, 2, 3, 4$ are modes. Some authors define that in this case there is no mode at all. Also, if one is dealing with a continuous-valued dataset, then (since there is almost no chance that the values will be repeated) one first group the data into bins, calculate the frequencies for the bins, and take the bin with the highest frequency (or, usually, the midpoint of that bin) as the mode of the dataset. In other words, one calculates the argmax for the frequency histogram, to find the mode of the continuous dataset.

Well, although the Mode can be far from the "center" of the data, from the "middle" of our data (and hence, not showing the central tendency), but sometimes it is used as a good representative for our data points. The most frequent data point is sometimes very important.

REMARK, UNIMODAL, BIMODAL DATASETS: In Descriptive Statistics, if a dataset has a unique mode, then one calls that type of datasets Unimodal. In case we have 2 modes, one calls that type of datasets Bimodal. Multimodal datasets have more that 2 modes.

REMARK, THEORETICAL MODE: If we have a distribution given by its PMF or PDF, then the Mode of that distribution is the $\mathtt{argmax}$ for the PMF or PDF function. Please look at the textbooks and other papers for the Theoretical Mode, Unimodal and Bimodal distributions.

R CODE, SAMPLE MODE: Unfortunately (or fortunately), **R**'s *mode(x)* command is not calculating the Sample Mode of x, rather it is giving the *type or storage mode* of x. Say, if x is a vector of numbers, for example,

```
x <- c(0,1,2,3)
```

then the result of *mode(x)* will be *"numeric"*. So let us write a function to calculate the Sample Mode of a dataset.

We create a function with a name *my.mode*:

```
#Sample Mode Calculation
my.mode <- function(x){
    y <- table(x)   #table of frequencies
    nam <- as.numeric(names(y)) # this will give unique values in x
    freq <- as.numeric(y)  #this will give the frequencies of the above values
    return(nam[freq == max(freq)]) #this will give all modes
}
```

Note that here the command *table(x)* returns a named array: names are the unique values and the elements are the corresponding frequencies. For example, if we will run

```
x <- c(0,4,9,0,9,0)
table(x)
```

Then the result will be:

```
x
0 4 9
3 1 2
```

Here the upper row is the vector of unique values in x: 1,2,3, and the bottom row show the corresponding frequencies: the frequency of 0 is 3, the frequency of 4 is 1, and the frequency of 9 is 2. Now, to extract the vector of names, we can use *names(table(x))*, and the result will be:

```
[1] "0" "4" "9"
```

These are not numbers, so in order to transform back to numbers we use *as.numeric(names(table(x)))*, and this will give the desired result. Obtaining frequencies is much easier: we use *as.numeric(table(x))* (since the actual elements of *table(x)* are the frequencies, and the values are just their names. We just need to convert that frequencies, stored as strings, back to numbers).

Next, try to make experiments/search Google to see what the code *x == max(x)* does with an array *x*, and the code *x(x == max(x))* is returning. Hope after that the code above will be clear.

Now, to test the code, we can run:

```
x <- c(10, 10, 2, 5, 7, 5, 1, 2, 7, 2, 6, 8, 2, 4, 8, 6)
my.mode(x)
```

or

```
x <- c(1,2,3,1,2,3,1,2,3,4)
my.mode(x)
```

Yeah, I know programming! Google, I am waiting for your offer!

REMARK, MS EXCEL AND DESCRIPTIVE STAT: By the way, there are many Statistical functions implemented in MS Excel. For example, you can calculate the Mean, Median, Trimmed Mean, Mode a sample using Excel. And you can generate random numbers, and many more. Try to explore the real power of Excel!

3.2.4 Other Statistics for the Central Tendency

One can define also other Statistics to measure the central tendency, the "center" of the dataset. For example, given a dataset x:

$$x_1, x_2, ..., x_n,$$

one can calculate the MidRange[4]

$$midrange(x) = \frac{x_{(1)} + x_{(n)}}{2}.$$

REMARK, MIDRANGE AS A MINIMIZER: We have seen above that the Sample Mean and the Sample Median are minimizers for some deviation measure. Here, for the MidRange, one can prove that if m is the MidRange of the dataset x, then

$$m \in \underset{a\in\mathbb{R}}{argmin} \max_k |x_k - a|,$$

so it minimizes the maximum absolute deviation.

You can find a lot of other measures at https://en.wikipedia.org/wiki/Central_tendency. By the way, there is a nice relationship between the Sample Mean, Mode and Median for a unimodal distribution, see https://en.wikipedia.org/wiki/Central_tendency#Relationships_between_the_mean,_median_and_mode.

3.3 Statistics for the Spread/Variability

Of course, the Sample Mean or the Median are the first descriptors for a univariate numerical dataset. But, you can easily guess that this numbers are not enough to give some picture about the dataset. Say, if I will state that the mean salary for two persons is 200K, then that will not give the idea about their salaries: they both can get 200K, or one can get 150K and the other one - 250K, or,

[4]See, for example, https://en.wikipedia.org/wiki/Mid-range

it can happen that one receives 0, and the other one - 400K. So knowing the center is not enough in most cases, we need to know how spread are the values about that center.

Now we want to give some measures for the concentration and spread, dispersion of our dataset. We will give again different measures for that.

Assume our observation, our dataset is $x_1, ..., x_n$.

3.3.1 Deviations

Definition 3.6. *If $\bar{x}$ is the Sample Mean of the dataset $x_1, ..., x_n$, then the differences*

$$x_k - \bar{x}, \qquad k = 1, ..., n$$

are called deviations of our dataset from the mean.

We can get a lot of information about the spread of our dataset using the deviations from the mean. Say, if the deviations are close to zero, then our dataset is concentrated about the mean, or if the deviations are symmetric about zero, then our dataset is symmetric about the mean. So deviations are good characteristics for our dataset spread, but, unfortunately, for a large dataset we will have a lot of numbers (deviations), so that will not help us to get the picture about the spread or variability. Instead, we want to describe the spread by just one number.

For example, we can try to find the mean of all the deviations, but, unfortunately, this will not give an information to us, because of the following Exercise:

Exercise: Prove that the sum of all deviations (and also the mean of all deviations) is 0.

Remark, Deviations from the Median: In fact, having a notion of the center of a dataset (say, Sample Mean, Median, Trimmed Mean or Weighted Mean), we can calculate the deviations from that center to estimate the variability and spread. For example, we can calculate deviations from the Sample Median of our dataset:

$$x_k - median(x), \qquad k = 1, 2, ..., n.$$

Remark, Absolute Deviations: One is defining also the Absolute Deviations from the Mean as the dataset

$$|x_k - \bar{x}|, \qquad k = 1, 2, ..., n.$$

3.3.2 The Range

One of the simplest measures for the spread is the Range:

Definition 3.7. *The **range** of the dataset x is the difference*

$$Range(x) = (\textit{the largest element in } x_k) \text{ - } (\textit{the smallest element in } x_k) = x_{(n)} - x_{(1)}.$$

R code, Remark On the *range* Function: Please note that in **R**, the range(x) function is returning the minimum and maximum values of the dataset x, but not the difference of that maximum and minimum. Here is an example:

```
#Range
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
range(x)
```

The result is:

```
[1] -3  5
```

To calculate the Range in our sense, we can use:

```
#Our Range, v1
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
r <- range(x)
my.range <-r[2]-r[1]
```

or just

```
#Our Range, v2
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
my.range <- max(x)-min(x)
```

Now, if we want to have a function to calculate the Range, we can write

```
#Our Range, v3, as a function
my.range <- function(x){
  return(max(x)-min(x))
}
```

and then try to use it for some dataset:

```
x <- c(2,3,9,0,1,0)
my.range(x)
```

Another method is:

```
#Our Range, v4, as a function, another version
my.range <- function(x){
  return(diff(range(x)))
}
```

And you can check by the above example dataset that this gives the same value.

3.3.3 The Sample Variance and Sample Standard Deviation

The above measure for the spread, the Sample Range, in fact, is not giving a good information for the spread, since we can have some outliers, and the rest of our dataset can be concentrated close to some point. One of the widely used measures for the spread or variability is the Sample Variance, and its square root, the Standard Deviation:

Definition 3.8. *The **Sample Variance** of our dataset is*

$$var(x) = s^2 = \frac{\sum_{k=1}^{n} (x_k - \bar{x})^2}{n},$$

where $\bar{x}$ is the sample mean of our dataset:

$$\bar{x} = mean(x) = \frac{1}{n} \cdot \sum_{k=1}^{n} x_k.$$

In many Statistics books we will find also the following definition for the Sample Variance:

Definition 3.9. *The **Sample Variance** of our dataset is*

$$var(x) = s^2 = \frac{\sum_{k=1}^{n} (x_k - \bar{x})^2}{n-1}.$$

There are some reasons why people prefer this last definition (later we will see that this is an *unbiased* estimate for the population variance[5]). In practice, if n is large, then these two definitions give very close results, so you can use either of them to calculate the Sample Variance.

Definition 3.10. *The **Sample Standard Deviation** is the square root of the Sample Variance:*

$$sd(x) = s = \sqrt{var(x)}.$$

So, in fact, we will have 2 values for the Sample Standard Deviation - with either n and $n-1$ in the denominator of the Sample Variance.

Example, Sample Variance and SD: Let us calculate the Sample Variance and the Sample Standard Deviation for the dataset x:

$$-1, 2, 1, 3, 0, 2, 1$$

The number of observations here is $n = 7$.

First, we calculate the Sample mean:

$$\bar{x} = \frac{-1+2+1+3+0+2+1}{7} = \frac{8}{7}.$$

Then we calculate the Sample Variance with $n = 7$ in the denominator:

$$var(x) = \frac{1}{7} \cdot \left[\left(-1 - \frac{8}{7}\right)^2 + \left(2 - \frac{8}{7}\right)^2 + \left(1 - \frac{8}{7}\right)^2 + \left(3 - \frac{8}{7}\right)^2 + \left(0 - \frac{8}{7}\right)^2 + \left(2 - \frac{8}{7}\right)^2 + \left(1 - \frac{8}{7}\right)^2 \right] =$$

$$= \frac{532}{7^3} \approx 1.5510$$

[5] And sometimes taking $n-1$ instead of n in the Sample Variance calculation is referred to as Bessel's correction.

and the Sample Standard Deviation is in this case

$$sd(x) = \sqrt{var(x)} = \sqrt{\frac{532}{7^3}} \approx 1.2454.$$

Now, if we want to calculate the Sample Variance and Standard Deviation with $n-1$ in the denominator, we just need to multiply the above var by $\frac{n}{n-1} = \frac{7}{6}$:

$$var(x) = \frac{532}{7^3} \cdot \frac{7}{6} = \frac{266}{3 \cdot 7^2} \approx 1.8095$$

and the Standard Deviation will be in this case

$$sd(x) = \sqrt{var(x)} = \sqrt{\frac{266}{3 \cdot 7^2}} \approx 1.3452.$$

Now, we give another formula to calculate the Sample Variance for the case when the denominator is n:

Proposition 3.1. *The Sample Variance (with the denominator n) can be calculated by the following formula*

$$var(x) = \frac{\sum_{k=1}^{n} x_k^2}{n} - \left(\frac{\sum_{k=1}^{n} x_k}{n}\right)^2 = \frac{\sum_{k=1}^{n} x_k^2}{n} - \left(\bar{x}\right)^2.$$

We can write this, using an analogy with the r.v. Variance[6],

$$var(x) = mean(x^2) - \left(mean(x)\right)^2,$$

where x^2 is the dataset $x_1^2, x_2^2, ..., x_n^2$. Just remember to use this in the case when the Sample Variance is with the denominator n !

The proof of the above Proposition is straightforward: one just need to do simple calculations. We leave the calculation joy to the interested reader ☺

REMARK, SAMPLE VARIANCE INTERPRETATION: Let us note here the link between the Sample Variance and a Variance of a r.v. . If we will define a r.v. X, which will take the values x_1, x_2, ..., x_n with the equal probabilities $\frac{1}{n}$, then the sample variance of the dataset $x_1, ..., x_n$, with the denominator n, will be exactly the variance of the r.v. X, i.e., we will have, for this X,

$$mean(x) = \mathbb{E}(X) \qquad \text{and} \qquad var(x) = Var(X) = \mathbb{E}((X - \mathbb{E}(X))^2).$$

[6]Recall that for a r.v. X,

$$Var(X) = \mathbb{E}(X^2) - \left(\mathbb{E}(X)\right)^2.$$

In case we define the Sample Variance by taking $n-1$ in the denominator, then the formula above will look like:

$$var(x) = \frac{\sum_{k=1}^{n}(x_k - \bar{x})^2}{n-1} = \frac{\sum_{k=1}^{n} x_k^2}{n-1} - \frac{\left(\sum_{k=1}^{n} x_k\right)^2}{n(n-1)}.$$

R code, Variance: In **R**, one can calculate the sample variance by using the command $var(x)$ for a dataset x.

```
#Variance Calculation
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
var(x)
```

This will give the result:

```
[1] 6.727273
```

Note that **R** is using $n-1$ for the denominator. For example, if $n = 1$, i.e., we have only one observation, **R** will produce **NA**, i.e., Not Available. Try:

```
#Variance is calculated by (n-1) in the denominator
x <- c(2)
var(x)
```

The result is:

```
[1] NA
```

R code, Implementation of the Variance: The following code will give the same result as $var(x)$:

```
#Variance by Sephakan Dzerqer
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
x.bar <- mean(x)
sum.of.deviations <-sum((x- x.bar)^2)
my.var <- sum.of.deviations/(length(x)-1)
```

We can also make a variance calculation as a function:

```
#Variance by Sephakan Dzerqer, v2
my.var <- function(x){
  return(sum((x-mean(x))^2)/(length(x)-1))
}
```

Now, to check the result, run

```
x <- c(-2,1,3,2,4,2,5,3,0,-1,-2,-3)
my.var(x)
```

EXAMPLE, COMPARING DATASETS WITH THE SAME MEAN AND DIFFERENT VARIANCES: Now let us consider the following datasets:

$$x: \quad -3, -4, 2, 0, -1, 2, 3, 4, 8, 2, -2, -2$$

$$y: \quad 13, 19.5, 0, -30, -10, -25, -30, 40, 20, 10$$

It is easy to calculate that the Sample Mean for both of these datasets are the same:

$$mean(x) = mean(y) = 0.75.$$

But the Standard Deviations are not the same (we use the **R**'s sd, with $n-1$ in the denominator):

$$sd(x) \approx 3.41 \qquad \text{and} \qquad sd(y) \approx 23.96$$

Clearly, y is more spread out than x. See the Fig. 3.6. By the way, please note that the picture is somewhat misleading - we have different datapoints with the same values - they will represent by the same point on the graph.

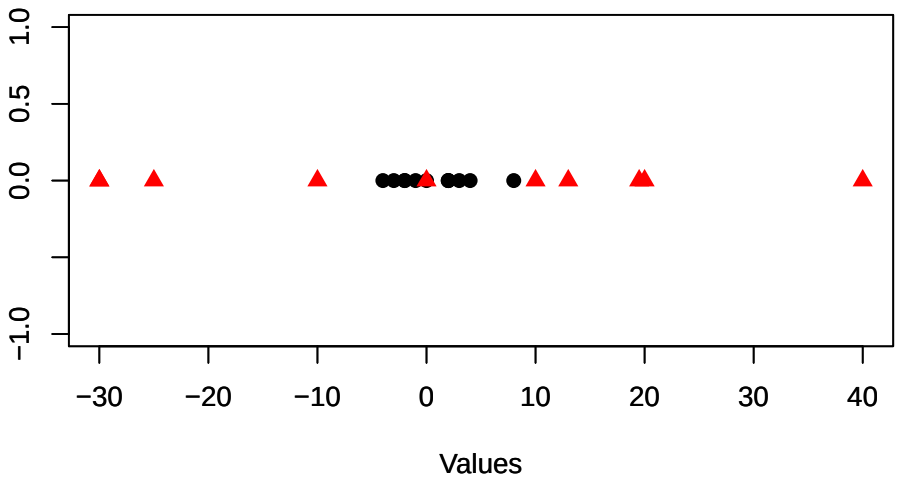


Fig. 3.6: The Datasets x (blue dots) and y (red triangles). They have the same Sample Mean, but the Sample Standard Deviation of y is larger than the x's one.

R CODE, COMPARING DATASETS WITH THE SAME MEAN AND DIFFERENT VARIANCES: The code for the above example is the following:

```
#Comparing two datasets with the same mean and different SDs
x <- c(-3, -4, 2, 0, -1, 2, 3, 4, 8, 2, -2, -2)
xbar <- mean(x)
xsdev <- sd(x)
y <- c(13, 19.5, 0, -30, -10, -25, -30, 40, 20, 10)
```

```
ybar <- mean(y)
ysdev <- sd(y)
z1 <- rep(0, length(x))
z2 <- rep(0, length(y))
plot(x,z1, pch = 16, cex = 1.2, xlim = c(min(y), max(y)), xlab = "Values", ylab = "")
par(new = TRUE)
plot(y,z2, pch = 17, col = "red", cex = 1.2, xlim = c(min(y), max(y)),
 xlab = "Values", ylab = "")
```

Now, let us give some properties for the Sample Variance. They are the analogues for the properties of the variance for a r.v..

Proposition 3.2. *Assume* x *is the dataset* $x_1, x_2, ..., x_n$, *and* $\alpha, \beta \in \mathbb{R}$ *are constants. If we will denote by* $\alpha \cdot x$ *the dataset* $\alpha \cdot x_1, \alpha \cdot x_2, ..., \alpha \cdot x_n$, *and by* $x + \beta$ *the dataset* $x_1 + \beta, x_2 + \beta, ..., x_n + \beta$, *then*

a. $var(x) \geqslant 0$;

b. $var(x) = 0$ *if and only if* $x_k = x_j$ *for any* k, j;

c. $var(\alpha \cdot x) = \alpha^2 \cdot var(x)$;

d. $var(x + \beta) = var(x)$.

Proof. I suggest you to check these properties by yourself. □

The weak side of the Sample Variance is that it is very sensitive to outliers: if we will have an outlier, then the deviation will be large, and the squared deviation will be even larger. That's why people say that Sample Variance is not a **robust** Statistics for a spread, it is not resistant to outliers.

Remark, Sample Variance and a Minimum Problem: Above, when talking about the Sample Mean, we gave a remark that the Mean is the number minimizing the sum of squared distances. Namely,

a. m is the Mean of the r.v. X if and only if

$$m \in \operatorname*{argmin}_{a \in \mathbb{R}} \mathbb{E}\Big((X - a)^2\Big);$$

b. m is the mean of the dataset $x_1, x_2, ..., x_n$ if and only if

$$m \in \operatorname*{argmin}_{a \in \mathbb{R}} \frac{1}{n} \cdot \sum_{k=1}^{n} (x_k - a)^2.$$

Here we can add the following (we will use $var(x)$ with the denominator n):

a.

$$Var(X) = \min_{a \in \mathbb{R}} \mathbb{E}\Big((X - a)^2\Big);$$

b.

$$var(x) = \min_{a\in\mathbb{R}} \frac{1}{n}\cdot\sum_{k=1}^{n}(x_k-a)^2.$$

So, in fact, the Mean is the minimum point of the sum of squared distances, and the Variance is the minimum value of that sum of squared distances.

Btw, here again you can see the analogy for the r.v.'s and finite datasets: if we will make a r.v. from the dataset $x_1,...,x_n$ by giving equal probabilities to each data point, then the discrete case can be obtained from the r.v. case.

REMARK, VARIANCE FROM A MEDIAN ETC.: Note that if we have a measure for the central tendency, some descriptor for the typical element in our dataset (say, the mean, median, mode, weighted mean, trimmed mean,...), then we can define the Sample Variance from that typical element. For example, we can define

$$\text{Variance from the Median}(x) = \frac{1}{n}\cdot\sum_{k=1}^{n}\Big(x_k-\text{Median}(x)\Big)^2.$$

3.3.4 The Mean Absolute Deviation (MAD) from the Mean and Median

Another way of measuring the spread of a dataset is the Mean Absolute Deviation from the Mean (or Median). Of course, you can define, by analogy, the Mean Absolute Deviation from any measure of a location (say, Trimmed or Weighted Mean).

Definition 3.11. *The Mean Absolute Deviation (MAD) from the Mean for the dataset* $x_1,...,x_n$ *is*

$$mad(x) = mad(x, mean) = \frac{\sum_{k=1}^{n}|x_k-\bar{x}|}{n}.$$

This is, of course the mean of Absolute Deviations from the Mean. Analogously,

Definition 3.12. *The Mean Absolute Deviation (MAD) from the Median for the dataset* $x_1,...,x_n$ *is*

$$mad(x) = mad(x, median) = \frac{\sum_{k=1}^{n}|x_k-median(x)|}{n}$$

REMARK, THEORETICAL MAD: Clearly, the above definitions can be generalized for any theoretical distribution, for random variables. Hope you got the trick with making a r.v. from a dataset.

Say, the analogue for the MAD from the Mean for a r.v. X will be

$$MAD(X) = \mathbb{E}(|X-\mathbb{E}(X)|).$$

REMARK, MAD IN OTHER WAY: One can also define the *Median absolute deviations from the Mean or Median*. Try to give the definitions!

R CODE, MAD: Here we need to have **R** codes for functions to calculate MADs of a dataset. But we do not. Why? Do not ask! Write it by yourself ☺

3.3.5 Some Experiments with a Sample Standard Deviation

Here, using **R**, I am giving some graphical plots for the Sample Standard Deviation. We will plot frequency histograms for some datasets, and show on the same graph the Sample Mean (using a thick vertical red line passing through the Sample Mean), and show the points

$$\text{Sample Mean} \pm \text{Sample Standard Deviation}$$

using vertical thin red lines passing through that points.

R CODE, SAMPLE STANDARD DEVIATION: Here are the codes:

```
#Sample Variance Experiments
set.seed(100)
x <- rnorm(1000, mean = 3, sd = 5)
bins <- seq(min(x)-1, max(x)+1, 1)
hist(x, breaks = bins)
xbar <- mean(x)
abline(v = xbar, col = "red", lwd = 3)
sdev <- sd(x)
abline(v = xbar - sdev, col = "red", lwd = 1)
abline(v = xbar + sdev, col = "red", lwd = 1)

set.seed(100)
x <- rnorm(1000, mean = 3, sd = 15)
bins <- seq(min(x)-1, max(x)+1, 1)
hist(x, breaks = bins)
xbar <- mean(x)
abline(v = xbar, col = "red", lwd = 3)
sdev <- sd(x)
abline(v = xbar - sdev, col = "red", lwd = 1)
abline(v = xbar + sdev, col = "red", lwd = 1)

set.seed(121)
x <- rnorm(1000, mean = 3, sd = 5)
bins <- seq(min(x)-1, max(x)+1, 1)
hist(x, breaks = bins)
xbar <- mean(x)
abline(v = xbar, col = "red", lwd = 3)
sdev <- sd(x)
abline(v = xbar - sdev, col = "red", lwd = 1)
```

```
abline(v = xbar + sdev, col = "red", lwd = 1)

set.seed(100)
x <- rexp(1000, rate = 2)
bins <- seq(min(x)-1, max(x)+1, 0.2)
hist(x, breaks = bins)
xbar <- mean(x)
abline(v = xbar, col = "red", lwd = 3)
sdev <- sd(x)
abline(v = xbar - sdev, col = "red", lwd = 1)
abline(v = xbar + sdev, col = "red", lwd = 1)
```

Here the command `set.seed(n)` fixes, in some sense, the random number generator: every time you will run the code, you will get the same random numbers. I am suggesting you to read about `set.seed` in the help documentation, and also to read about Random Number Generation methods.

The results are show in Fig. 3.7-3.10. If you will run the above codes, you will get *exactly* the same pictures.

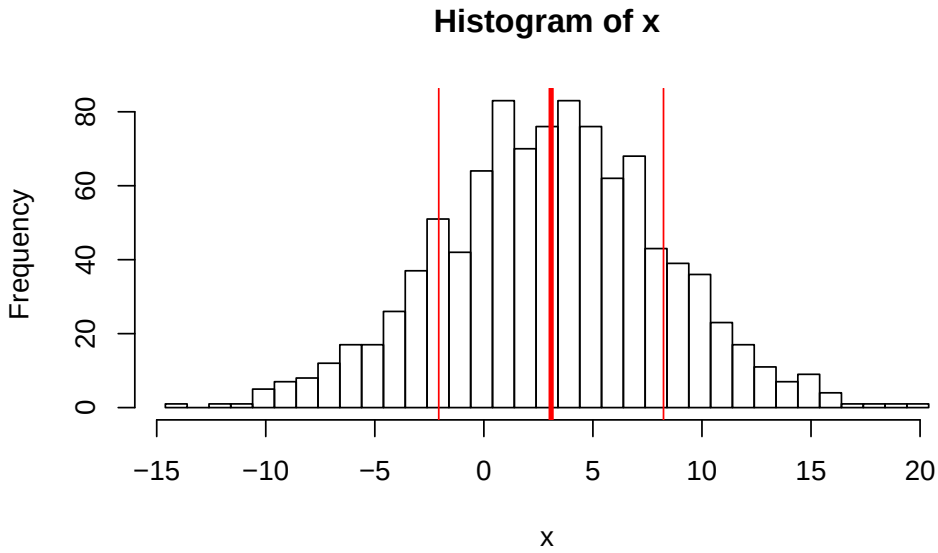


Fig. 3.7: 1000 samples from $\mathcal{N}(3, 5^2)$, from the seed 100. The Sample Mean is $\bar{x} \approx 3.084$, the Sample Standard Deviation is $sd(x) \approx 5.153$.

3.3.6 Other Measures for the Spread/Variability

There are a lot of other measures for a Spread/Variability of a dataset. In the next section we will give the Inter-Quartile Range, which is one of the important Spread descriptors. Another one is the Winsorized Sample Variance and the Standard Deviation, which can be found in *Rand R. Wilcox, Basic Statistics: Understanding Conventional Methods and Modern Insights, Oxford University Press, 2009, p. 27-28.*

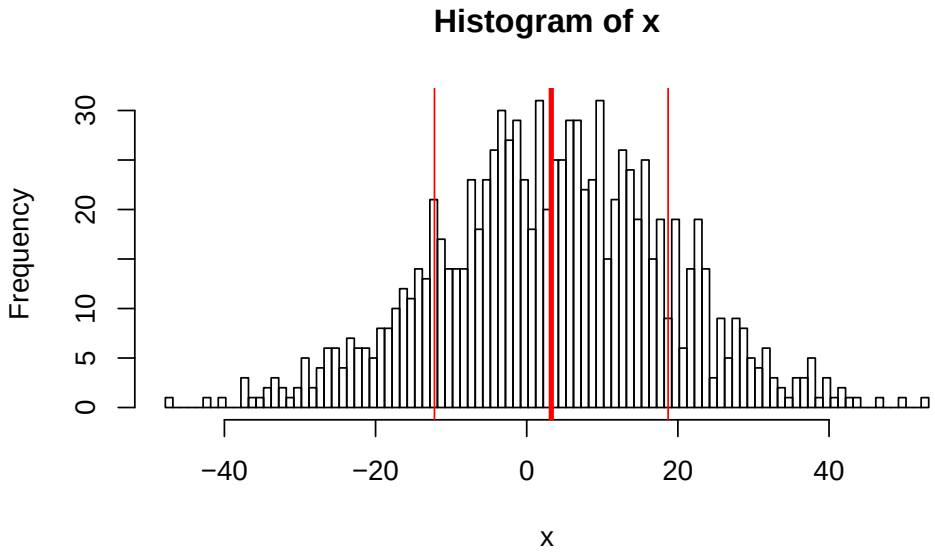


Fig. 3.8: 1000 samples from $\mathcal{N}(3, 15^2)$, from the seed 100. The Sample Mean is $\bar{x} \approx 3.252$, the Sample Standard Deviation is $sd(x) \approx 15.459$.

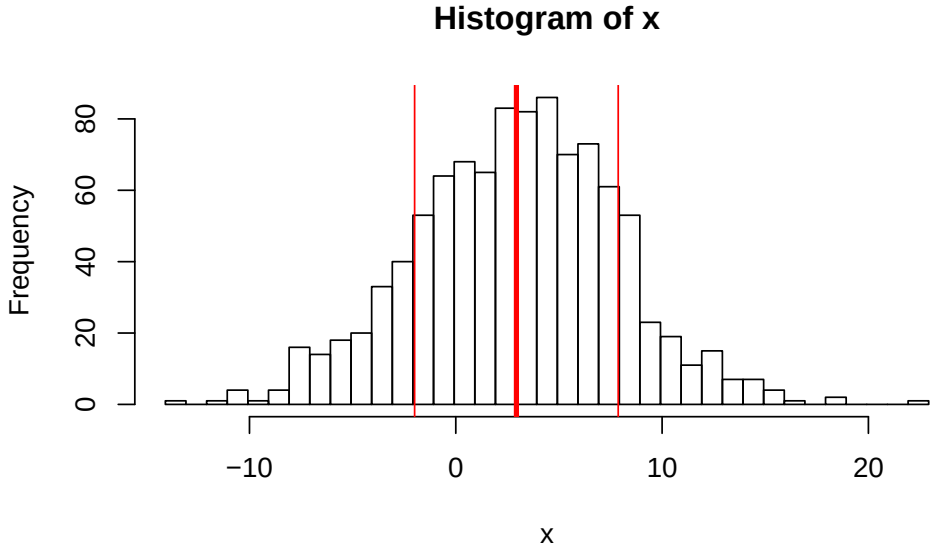


Fig. 3.9: 1000 samples from $\mathcal{N}(3, 5^2)$, from the seed 121. The Sample Mean is $\bar{x} \approx 2.939$, the Sample Standard Deviation is $sd(x) \approx 4.934$.

You can find others at https://en.wikipedia.org/wiki/Deviation_(statistics) and at https://en.wikipedia.org/wiki/Statistical_dispersion.

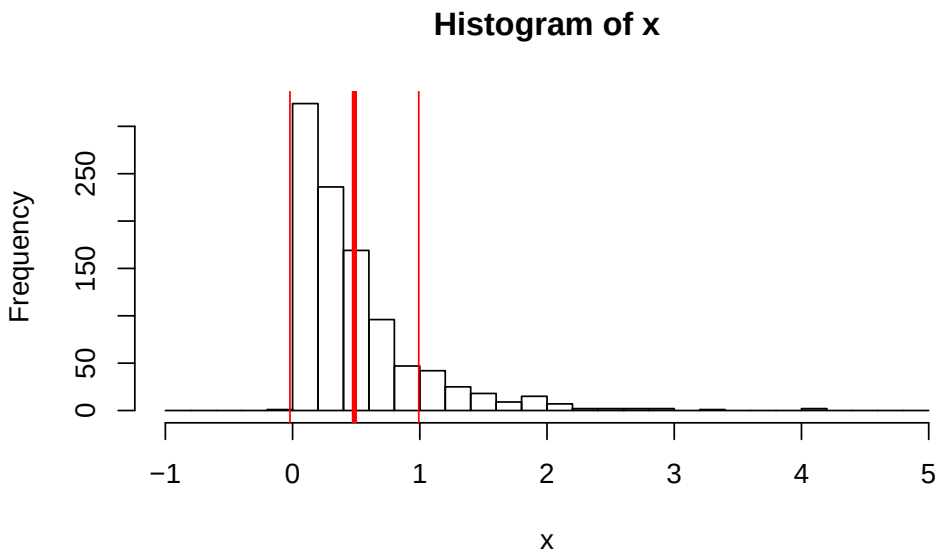


Fig. 3.10: 1000 samples from $\mathrm{Exp}(2)$, from the seed 100. The Sample Mean is $\bar{x} \approx 0.486$, the Sample Standard Deviation is $\mathrm{sd}(x) \approx 0.506$.

3.4 Other Numerical Summaries

Besides describing a dataset through the location and variability, people also give other numerical summaries. Widely used summaries are the shape parameters: Sample Kurtosis and Sample Skewness. See, for example, https://en.wikipedia.org/wiki/Kurtosis and https://en.wikipedia.org/wiki/Skewness.

Exploratory Data Analysis for Univariate Data: Quantiles and BoxPlots

4.0.1 Quartiles and Interquartile Range, IQR

Let

$$x_{(1)} \leqslant x_{(2)} \leqslant ... \leqslant x_{(n-1)} \leqslant x_{(n)}$$

be the sorted dataset obtained from $x_1, ..., x_n$. Assume Med is the Sample Median of our dataset.

Definition 4.1.

- *The first (or lower) quartile,* Q_1*, is the Median of the ordered dataset of all observations to the left of* Med_x *(including* Med_x*, if it is a data point)*
- *The second (or middle) quartile,* Q_2*, is the Median of our dataset;*
- *The third (or upper) quartile,* Q_3*, is the Median of the ordered dataset of all observations to the right of* Med_x *(including* Med_x*, if it is a data point);*
- *The InterQuartile Range,* $\mathrm{IQR} = Q_3 - Q_1$.

!!! See https://en.wikipedia.org/wiki/Quartile for different methods to calculate the quartiles

Roughly, 25% of all observations are less than or equal to Q_1, and 25% of all observations are greater than or equal to Q_3. So quartiles divide our sorted dataset into four equal parts. Also, about 50% of observations lie in $[Q_1, Q_3]$. In some sense, $[Q_1, Q_3]$ is the most central interval containing 50% of observations. The length of $[Q_1, Q_3]$, IQR shows how spread is the central portion of our data.

The following R code gives a simple summary statistics for a data:

```
#summary
x<-c(1,1,1,2,3,1,3,4,1,5,1,6)
summary(x)
fivenum(x)
```

Definition 4.2. *We will say that* x_i *is an* ***outlier****, if*

$$x_i \notin \left[Q_1 - \frac{3}{2}\mathrm{IQR}, Q_3 + \frac{3}{2}\mathrm{IQR} \right].$$

There are other definitions of outliers, but we will not touch this topic.

Question: Will the set of outliers be changed if we will change the scale, change the units?

Question: Why $\frac{3}{2}$ in the definition of the IQR and outliers? This comes from the Normal Distribution. Explain!!

4.1 BoxPlot

There is another convenient way of visualizing our data: boxplots. Assume $x_1, ..., x_n$ is our dataset.

To obtain the BoxPlot, we calculate:

- M, the Median of our dataset, and the lower and upper quartiles Q_1, Q_3;
- the lower and upper fences $W_1 = \min\{x_i : x_i \geqslant Q_1 - 1.5 \cdot IQR\}$ and $W_2 = \max\{x_i : x_i \leqslant Q_3 + 1.5 \cdot IQR\}$, i.e., the first and last observations lying in $\left[Q_1 - \frac{3}{2}IQR, Q_3 + \frac{3}{2}IQR\right]$; the line joining that fences to corresponding quartiles are the whiskers;
- the set of all outliers $O = \left\{x_i : x_i \notin \left[Q_1 - \frac{3}{2}IQR, Q_3 + \frac{3}{2}IQR\right]\right\}$

Then we draw the points W_1, Q_1, M, Q_3, W_2 on the real line and add all outliers, and make a box over $[Q_1, Q_3]$:

The R code:

```
#Boxplot
x <- rnorm(20, 1, 1)
x <- c(x, 3)
boxplot(x)
boxplot(x, horizontal = TRUE)
boxplot.stats(x)
```

and

```
#Boxplot Example 2
x <- rnorm(20)
par(mfrow=c(1,2))
boxplot(x, horizontal = TRUE)
y <- rep(0, 20)
plot(x,y)
```

and

```
#Boxplot Example 3
mtcars
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data", xlab="Number of Cylinders", ylab="Miles Pe
```

Example: Plot different examples: Histogram vs BoxPlot (symmetric, skewed left, right etc)

R CODE, BOXPLOT EXAMPLES, COMPARISON OF DATASETS:

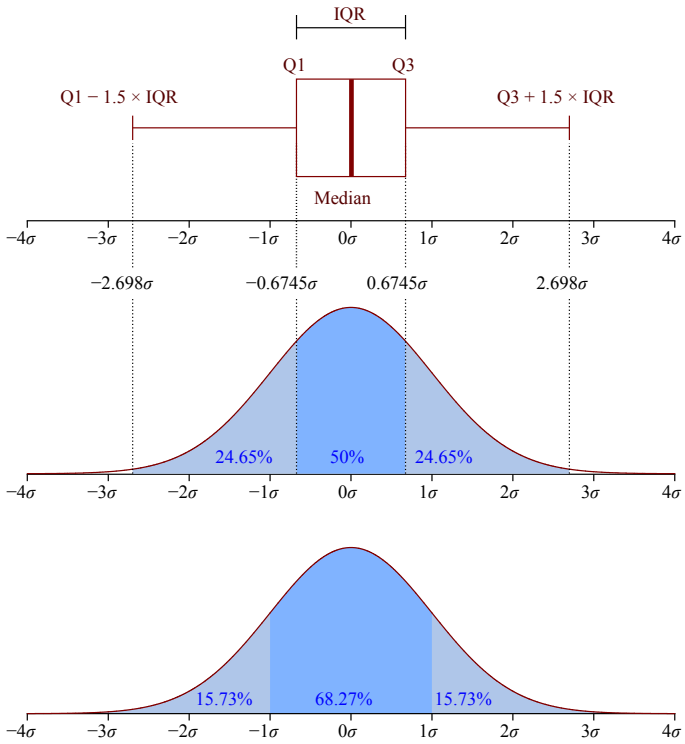


Fig. 4.1: Boxplot vs PDF for the Normal Distribution

```
discoveries
#boxplot(discoveries, horizontal = T, notch = T, plot =T, col = "gray", range = 1.5)
boxplot(discoveries, horizontal = T, col = "gray")
boxplot.stats(discoveries)
ts.plot(discoveries)

Orange
boxplot(circumference~Tree, data = Orange)

PlantGrowth
head(PlantGrowth)
boxplot(weight~group, data = PlantGrowth)

head(mtcars)
boxplot(mpg ~ cyl, data = mtcars)
```

Graphical Example:

4.2 Outliers

We are given a dataset $x_1, x_2, ..., x_n$. We want to separate atypical elements in our dataset, the outliers. Other term for describing the outliers is anomalies.

There are different methods to define outliers:

Definition 4.3. *We will say that x_i is an outlier, if*

Classical 1 -

$$|x_i - \bar{x}| \geqslant 2 \cdot sd(x)$$

Classical 2 -

$$|x_i - \bar{x}| \geqslant 3 \cdot sd(x)$$

BoxPlot Method -

$$x_i < Q_1 - 1.5 \cdot IQR \quad \text{or} \quad x_i > Q_3 + 1.5 \cdot IQR,$$

that is, if

$$x_i \notin \Big[Q_1 - 1.5 \cdot IQR,\ Q_3 + 1.5 \cdot IQR\Big].$$

The idea behind these definitions is the following. It is easy to calculate, using some Math software, that if $X \sim \mathcal{N}(\mu, \sigma^2)$, i.e., if X is r.v. with a Normal Distribution with mean μ and standard deviation σ, then[1]

$$\mathbb{P}(|X - \mu| \leqslant 2\sigma) \xrightarrow{Z=\frac{X-\mu}{\sigma}} \mathbb{P}(|Z| \leqslant 2) \approx 0.954499,$$

so

$$\mathbb{P}(|X - \mu| > 2\sigma) \approx 0.04550026.$$

R CODE, NORMAL DISTRIBUTION:

```
#We calculate the probability that Z\in[-2,2]
pnorm(2)-pnorm(-2)
```

This means that if X is normally distributed with a mean μ and standard deviation σ, then the probability that X will be in $[\mu - 2\sigma, \mu + 2\sigma]$ is 95.4%

Similarly, for $X \sim \mathcal{N}(\mu, \sigma^2)$,

$$\mathbb{P}(|X - \mu| \leqslant 3\sigma) \approx 0.9973,$$

so with more that 99.7% probability, X will be in $[\mu - 3\sigma, \mu + 3\sigma]$. Now, if **we will assume that our data points, our observations x_k, come from the Normal Distribution**[2], then the chance that x_k will be more than 3σ away from μ is veeeery small, only 0.3%. So if x_k is not lying in $[\mu - 3\sigma, \mu + 3\sigma]$ we can call it an outlier (or a black swan, or a white raven,...), for sure. Please note that in the definition above we are using $\bar{x}$ and $sd(x)$, but not μ and σ, because we do not have that μ and σ, and, moreover, we even do not know that our data comes from the Normal Distribution.

[1] Recall the Normalization or the Z-score!

[2] Veeeery strong assumption.

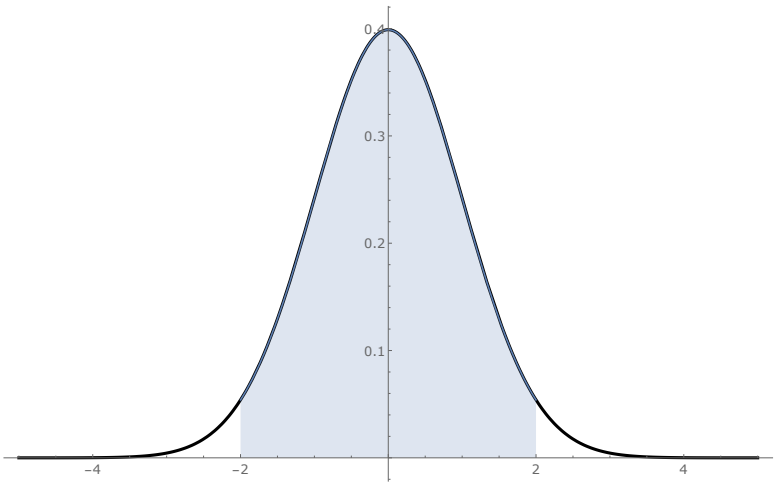


Fig. 4.2: The area under the graph of the Standard Normal Distribution from -2 to 2, the shaded region area, is 0.954

Similarly, for $X \sim \mathcal{N}(0,1)$, the lower and upper (theoretical) quartiles are

$$Q_1 \approx -0.6744898 \qquad \text{and} \qquad Q_3 \approx 0.6744898,$$

so

$$IQR = Q_3 - Q_1 \approx 1.34898.$$

Now,

$$\mathbb{P}(X \in [Q_1 - 1.5 \cdot IQR, Q_3 + 1.5 \cdot IQR]) \approx 0.9930234$$

meaning that if our data comes from a Normal Standard Distribution (you can translate this, by using again the normalization argument, to Normal r.v.s with other mean and variance), then the chances that a datapoint will be out of the interval $[Q_1 - 1.5 \cdot IQR, Q_3 + 1.5 \cdot IQR]$ is only 0.7%. So if it is not in this interval, it is an outlier.

R code, Probabilities for Standard Normal r.v.:

```
Q1=qnorm(0.25) #the 25% quantile, i.e., the lower quartile for Standard Normal RV
Q3=qnorm(0.75) #the 75% quantile, i.e., the upper quartile for Standard Normal RV
IQR = Q3 - Q1  #the IQR
pnorm(Q3+1.5*IQR)-pnorm(Q1-1.5*IQR)
```

4.3 Numerical Summaries

R: fivenum, summary,...

4.4 Interesting Things

Problem from AMM, No 11962: *Proposed by Elton Hsu, Northwestern University, Evanston, IL.* Let $\{X_n\}$, $n \geqslant 1$ be a sequence of independent and identically distributed random variables each taking the values ± 1 with probability $1/2$. Find the distribution of the random variable

$$\sqrt{\frac{1}{2}+\frac{X_1}{2}\sqrt{\frac{1}{2}+\frac{X_2}{2}\sqrt{\frac{1}{2}+....}}}$$

Exploratory Data Analysis for Bivariate Data: Quantiles and Q-Q Plots

Here we define the notion of a quantile, and give the Q-Q Plots ideas.

The idea of quantile is a generalization of the Median idea. The idea of the Sample Median was to give a number dividing the dataset into to parts, such that half of the data is to the left (or equal to) of that number, and half of the data - to the right (or equal to).

The idea of a Sample Quantile is a straightforward generalization of the Median idea: if we want to define the α-order quantile, or the α-quantile, for $\alpha \in (0,1)$, then we want to find a number that will divide our dataset into the proportions α and $1-\alpha$, i.e., we want to find a number such that $100\alpha\%$ of our datapoints will be to the left of that number (or equal to), and the rest, i.e., $100(1-\alpha)\%$ of the datapoints will be to the right (or equal to) of that number.

Analogously, quantiles can be defined also for theoretical distributions. Here, in this section, we define the quantiles (or percentiles) for a dataset and for a distribution, and then compare the quantiles using the Q-Q Plot.

5.1 Theoretical Quantiles, Quantiles for a Distribution

Assume we have a CDF $F(x)$ for some distribution.

Definition 5.1. *For $\alpha \in (0,1)$, the α-th quantile q_α of that distribution is defined by*

$$q_\alpha = q_\alpha^F = \inf\{x \in \mathbb{R} : F(x) \geqslant \alpha\}. \tag{5.1}$$

If F is strictly increasing and continuous, then the α-th quantile q_α is defined to be the unique solution to

$$F(q_\alpha) = \alpha,$$

or, in terms of the PDF $f(x)$, we will have

$$\int_{-\infty}^{q_\alpha} f(x)dx = \alpha.$$

In other words, if q_α is the α-th quantile for F, which is continuous and strictly increasing, and if X is a r.v. with CDF $F(x)$, then

$$\mathbb{P}(X \leqslant q_\alpha) = \alpha \qquad \text{and} \qquad \mathbb{P}(X \geqslant q_\alpha) = 1-\alpha.$$

So for the α-th quantile q_α, and for r.v. X with CDF $F(x)$, we will have that with probability α the values of X are to the left than or equal to q_α, and with the probability $1-\alpha$, the values of X are larger than q_α. Or, which is the same, the area under the PDF of that distribution in the region $(-\infty, q_\alpha]$ is equal to α, i.e., the line $x = q_\alpha$ divides the area under the graph of PDF into the portions α (left portion) and $1-\alpha$ (right portion), see Fig. 5.1.

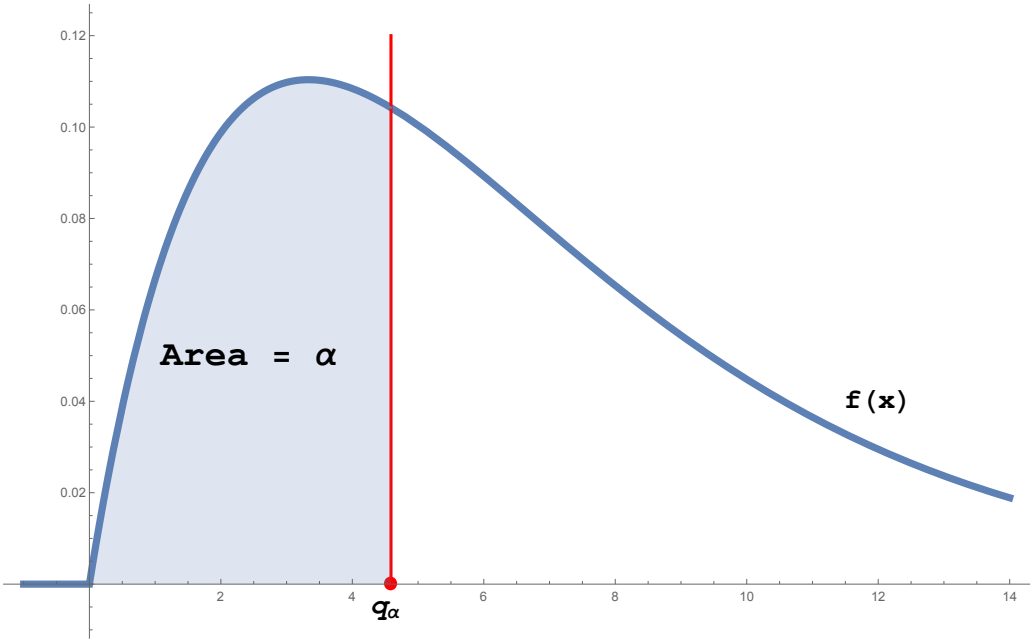


Fig. 5.1: Theoretical distribution's α-quantile, q_α, using the PDF graph. The area under the PDF left to the vertical line passing through the point q_α is exactly α

To explain the notion of the α-th quantile geometrically on the CDF graph - q_α is the leftmost point on the x-axis, where the graph of our CDF $F(x)$ crosses or jumps over[1] the value α, see Fig. 5.2.

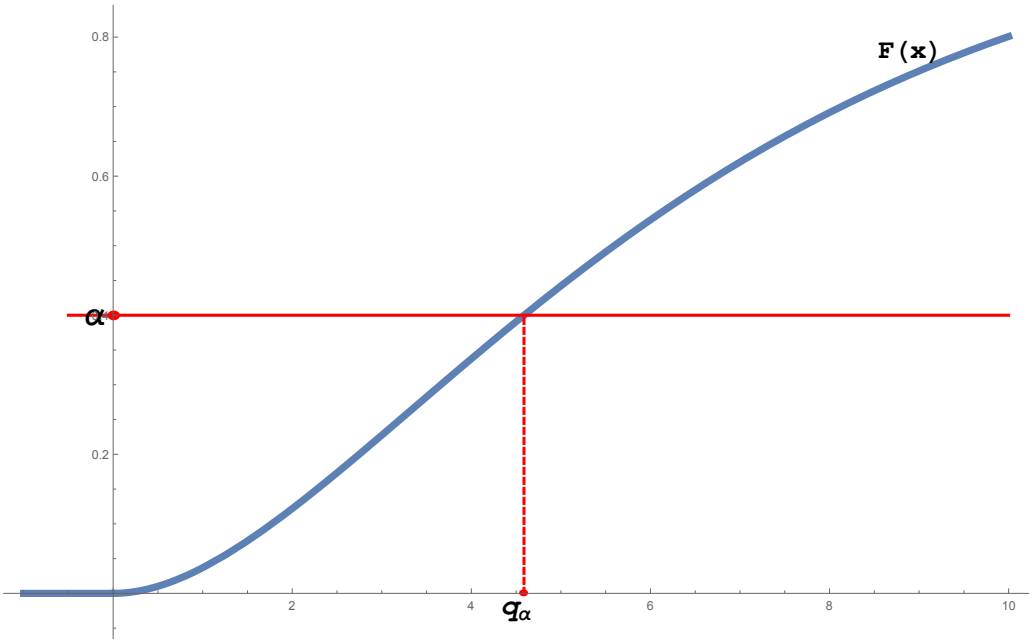


Fig. 5.2: Theoretical distribution's α-quantile, q_α, using the CDF graph. If we will consider the line $y = \alpha$ (solid red), then the (leftmost) intersection point with the CDF is $x = q_\alpha$

[1]For the continuous distribution case, it will cross for sure, but not for the discrete case, in general

EXAMPLE, QUANTILES FOR A THEORETICAL DISTRIBUTION, CONTINUOUS CASE: Consider the distribution (called a Pareto Distribution, with some parameters) with the following PDF:

$$f(x) = \begin{cases} \frac{2}{x^3}, & x \geqslant 1 \\ 0, & x < 1 \end{cases}$$

It is easy to check that f is a PDF, i.e., $\int_{-\infty}^{\infty} f(x)dx = 1$, and $f(x) \geqslant 0$ for any $x \in \mathbb{R}$. The graph of f is given in Fig. 5.3.

Now, let us calculate the 40% quantile for this distribution, i.e., calculate $q_{0.4}$. To that end, we need to find a point on the x-axis such that the vertical line passing through that point will divide the total area under the PDF graph into the portions 0.4 (to the left to that line) and 0.6 (to the right). So we need to have that the area under the PDF $f(x)$ from $-\infty$ to the quantile $q_{0.4}$ needs to be equal to 0.4, i.e.

$$\text{Area under the PDF curve from } -\infty \text{ to } q_{0.4} = \int_{-\infty}^{q_{0.4}} f(x)dx = 0.4$$

In other words, we need to solve

$$\int_{-\infty}^{q_{0.4}} f(x)dx = 0.4$$

to find $q_{0.4}$. Of course, $q_{0.4}$ cannot be less than 1, otherwise the integral will give 0. So $q_{0.4} > 1$. In that case,

$$0.4 = \int_{-\infty}^{q_{0.4}} f(x)dx = \int_{1}^{q_{0.4}} f(x)dx = \int_{1}^{q_{0.4}} \frac{2}{x^3}dx \overset{\text{Show this!}}{=\!=\!=} 1 - \frac{1}{(q_{0.4})^2}.$$

This gives that $\frac{1}{(q_{0.4})^2} = 0.6$, hence, recalling that $q_{0.4} > 1$, we will get

$$q_{0.4} = \sqrt{\frac{5}{3}},$$

see Fig. 5.3.

Now, let us solve the general problem of calculation of all α-quantiles for any $\alpha \in (0,1)$. As above, we need to solve the equation

$$\text{Area under the PDF curve from } -\infty \text{ to } q_\alpha = \int_{-\infty}^{q_\alpha} f(x)dx = \alpha$$

to calculate the α-quantile q_α. So we need to solve

$$\int_{-\infty}^{q_\alpha} f(x)dx = \alpha$$

for q_α. Again, q_α cannot be less than 1, since our PDF is 0 for any $x < 1$. So $q_\alpha > 1$. Then,

$$\alpha = \int_{-\infty}^{q_\alpha} f(x)dx = \int_{1}^{q_\alpha} f(x)dx = \int_{1}^{q_\alpha} \frac{2}{x^3}dx = 1 - \frac{1}{(q_\alpha)^2}.$$

This gives $\frac{1}{(q_\alpha)^2} = 1 - \alpha$, implying that

$$q_\alpha = \pm\sqrt{\frac{1}{1-\alpha}}.$$

Using the condition $q_\alpha > 1$ (and hence, $q_\alpha > 0$), we will get that

$$q_\alpha = \sqrt{\frac{1}{1-\alpha}}.$$

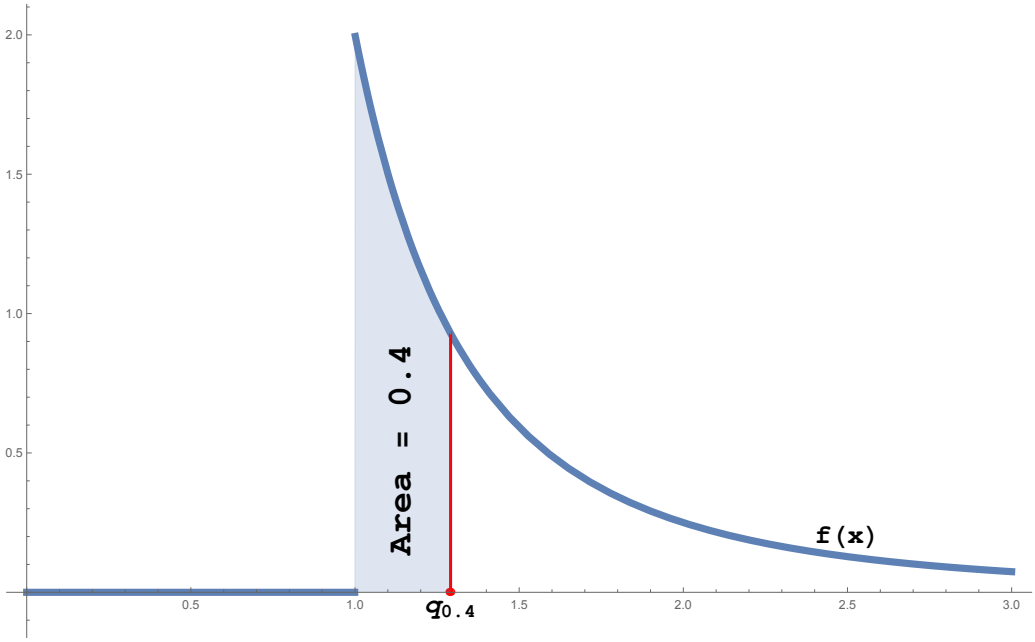


Fig. 5.3: Pareto Distribution PDF and its 40% quantile.

Example, Quantiles for a Theoretical Distribution, Continuous case: Here let us give another example of quantile calculation using the CDF. The aim of this example is to show why we use the infimum in the quantile definition (5.1).

Consider the distribution given by its CDF $F(x)$ as in Fig. 5.4, and let $\alpha = 0.6$, so we want to find the 60% quantile of this distribution. Clearly,

$$\{x \in \mathbb{R} : \ F(x) \geqslant \alpha\} = [2, +\infty),$$

and $F(q) = \alpha$ has infinitely many solutions, namely, any number $q \in [2, 3]$ will satisfy. So we cannot find a unique point q with $F(q) = \alpha$. Of course, any such point q will divide the range of our distribution into two pieces, and the probability that the values of a r.v. with this distribution will be less or equal to q will be α:

$$\mathbb{P}(X \leqslant q) = \alpha,$$

where X is a r.v. with CDF $F(x)$. So, in theory, this q can serve as a quantile. But usually people take the minimal such q (in fact, the infimum of such q-s). So in our case, $q_\alpha = 2$.

Example, Theoretical Quantiles, Discrete case:

Now let us consider the following r.v.:

X	1	3	5
$\mathbb{P}(X = x)$	0.3	0.5	0.2

We want to calculate the $\alpha = 0.6$-th quantile for the distribution of X.

The CDF of X, $F(x)$ is given in Fig. 5.5. From the graph it is clear that

$$\{x \in \mathbb{R} : F(x) \geqslant \alpha\} = \{x \in \mathbb{R} : F(x) \geqslant 0.6\} = [3, +\infty),$$

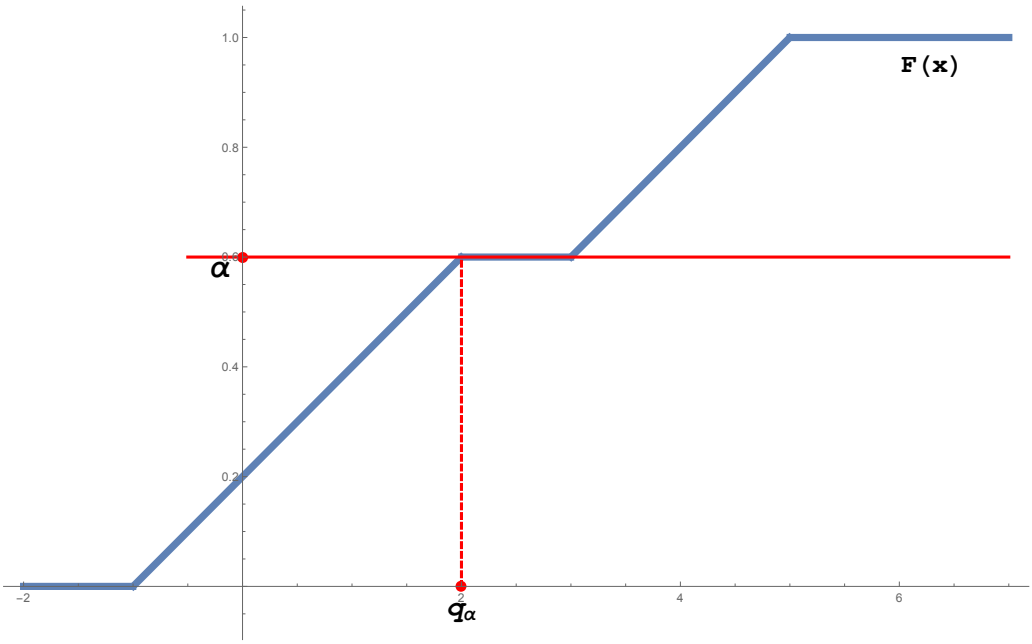


Fig. 5.4: CDF for some distribution and its $\alpha = 0.6$ quantile.

so $q_{0.6} = q_\alpha = 3$.

Similarly, for any $\alpha \in (0.3, 0.8]$, $q_\alpha = 3$. But, say, for $\alpha = 0.81$, $q_\alpha = 5$.

Note that for the Continuous Disctribution case, for any $\alpha \in (0, 1)$, we will always have a $q \in \mathbb{R}$ (unique or not) with $F(q) = \alpha$. But in the Discrete Distribution case, not for all α we will have such q-s. Say, for our example, no q exists with $F(q) = 0.6$. That's why in the definition of the quantiles (5.1) we are not using $\inf\{x \in \mathbb{R} : F(x) = \alpha\}$, but $\inf\{x \in \mathbb{R} : F(x) \geqslant \alpha\}$, with inequality sign.

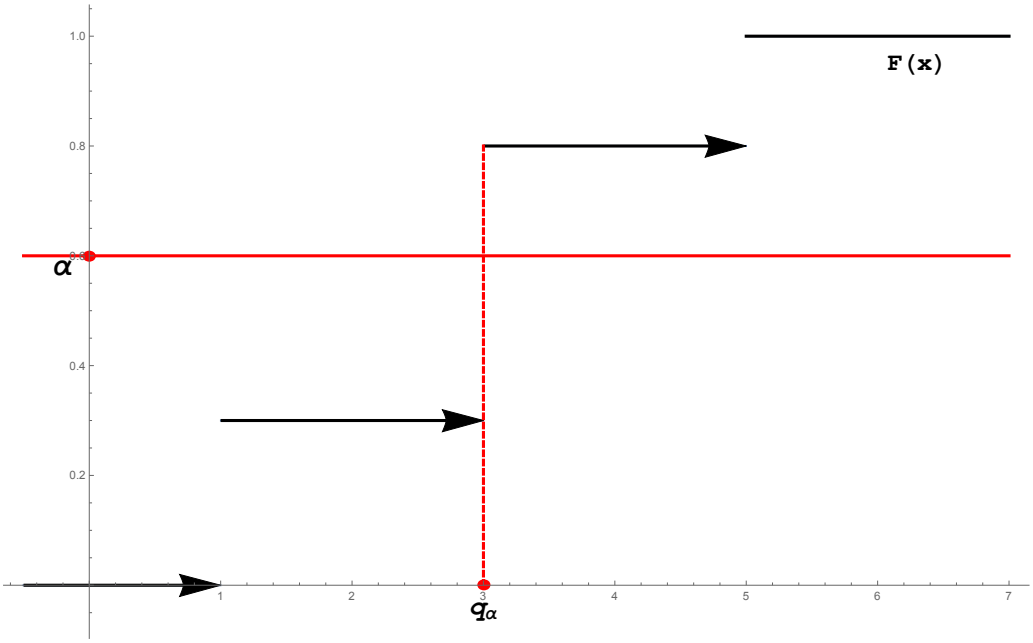


Fig. 5.5: CDF for a discrete distribution and its $\alpha = 0.6$ quantile.

R CODE, QUANITILES IN R: In **R**, any distribution has 4 standard functions ([name] is the name of the distribution in **R**):

r[name] - generates random numbers from the distribution [name]. Say, rnorm(100) will generate 100 random (pseudo-random) numbers from the Standard Normal distribution, and rpois(100, lambda = 3) will generate 100 random numbers from Pois(3) distribution.

p[name] - gives the CDF of the distribution [name]. For example, punif(0.2) will give the value of the CDF at the point 0.2 for the Standard Uniform distribution, Unif[0, 1]. Also, punif(0.2, min = 2, max = 5) will give the value of the CDF at the point 0.2 for the Uniform distribution Unif[2, 5] (can you guess the value?).

d[name] - gives the PDF of the distribution [name]. For example, curve(dexp, 0, 4) will draw the Exp(1) distribution's PDF in [0, 4], and dexp(1, rate = 2) will return the value of the Exp(2) distribution's PDF at $x = 1$.

q[name] - gives the quantiles of the distribution [name]. For example, qcauchy(0.3) will give the 30% quantile for the Cauchy Distribution.

Now, the command

```
qnorm(0.2)
```

will return -0.8416212, so the 20% quantile of the Standard Normal Distribution $\mathcal{N}(0, 1)$ is

$$q^{\mathcal{N}}_{0.2} = -0.8416212,$$

(well, after doing some rounding, since the actual number will have infinitely many digits after the period - you can find more digits in the "Environment" tab in RStudio). This means that for a r.v. $X \sim \mathcal{N}(0, 1)$,

$$\mathbb{P}(X \leqslant -0.8416212) = 0.2$$

We can check this in **R** by running the command

```
pnorm(-0.8416212)
```

or, we can even run

```
pnorm(qnorm(0.2))
```

To show the result graphically, we can run the following code:

```
#Quantile, Geometrically, on the CDF
alpha <- 0.2 #alpha = 0.2
qalpha <- qnorm(alpha) #20% quantile for the Standard Normal Distrib
plot(pnorm, xlim = c(-5,5), lwd = 2)  #The graph of the Standard Normal Distrib's CDF
abline(h = alpha, xlim = c(-5,5), lwd = 2, col="red") #horizontal line y=alpha
abline(v = qalpha, lwd = 2, lty = 2, col = "red") #vertical line through the quantile
qalpha # the value of the quantile
```

We can do similar thing with the PDF:

```
#Quantile, Geometrically, on the PDF
alpha <- 0.2 #alpha = 0.2
qalpha <- qnorm(alpha) #20% quantile for the Standard Normal Distrib
plot(dnorm, xlim = c(-5,5), lwd = 2)  #The graph of the Standard Normal Distrib's CDF
abline(v = qalpha, lwd = 2, lty = 2, col = "red") #vertical line through the quantile
integrate(dnorm,-Inf, qalpha) #The area under the PDF left to the quantile,
                              #the integral of the PDF over (-Infinity, qalpha]
qalpha # the value of the quantile
```

R code, Quantiles in R: Let us calculate the quantiles of orders $0.1, 0.25, 0.5, 0.75, 0.9$ for the distribution $\mathcal{N}(-2, 5^2)$:

```
#Quantiles for N(-2,5^2)
alpha <- c(0.1, 0.25, 0.5, 0.75, 0.9) #the vector of quantile orders
qnorm(alpha, mean = -2, sd = 5)
```

The result is

```
[1] -8.407758 -5.372449 -2.000000  1.372449  4.407758
```

Here the idea is that you can calculate several quantiles for the same distribution simultaneously, by passing the vector of the quantile orders to the quantile function of that distribution.

Another example,

```
#Quantiles for Exp(1)
qexp(c(0.2, 0.5, 0.7))
```

will return the quantiles of orders $0.2, 0.5$ and 0.7 for the distribution $\mathrm{Exp}(1)$, and the result will be

```
[1] 0.2231436 0.6931472 1.2039728
```

5.2 Quantiles for a Dataset

Now, if we have a dataset x, then the α-th quantile is the number for which approximately $100 \cdot \alpha\%$ of data is below that number, and the rest are above it: say, if $\alpha = 0.3$, then the 0.3-quantile is the "point" below which we will have 30% of our observations[2], and above which will be 70% of all observations.

Now, to define for any $\alpha \in (0, 1)$ the α-th quantile of a dataset (or the order α quantile), we will use the following definition[3].

Definition 5.2. *For a dataset x and $\alpha \in (0, 1)$, the quantile of order α is defined by*

$$q_\alpha = q_\alpha^x = x_{([\alpha \cdot n])}. \tag{5.2}$$

[2] Approximately 30%, since for 7 point dataset, we need to have 2.1 points are to the left or equal to the quantile $q_{0.3}$, but what it means "2.1 points"?

[3] Please note that there are different definitions of a sample quantile, and they give slightly different values. For example, you can read the help file of R package to find the description of 9 types of quantiles[4]

EXAMPLE, DATASET QUANTILES:

... Here you can put your ad ...

REMARK, QUANTILE DEFINITION: Another possible definition is (from Wesserman's book):

Definition 5.3. *For a dataset x and $\alpha \in (0,1)$, the quantile of order α is defined by*

$$q_\alpha = q_\alpha^x = \inf\{x : \mathrm{ECDF}(x) \geqslant \alpha\}.$$

Here the idea is one of the standard methods in Statistics: we have the definition of the theoretical quantile given in (5.1):

$$q_\alpha = q_\alpha^F = \inf\{x \in \mathbb{R} : F(x) \geqslant \alpha\}.$$

Now, for a dataset x, instead of theoretical CDF $F(x)$ we take the Empirical CDF $\mathrm{ECDF}(x)$! Nice and clear! If you will have some theoretical notion defined in terms of CDF, for a dataset replace CDF by ECDF, and that's it ˙·.◡

Btw, is this definition giving the same values for quantiles as our one?

REMARK, QUANTILES AND QUARTILES: Using our quantile definition above, one will not obtain that $q_{\frac{1}{2}}$ is always the Median, or $q_{1/4}$ and $q_{3/4}$ are the quartiles Q_1 and Q_3, respectively, for any datasets. This is because, say, by our definition, quantiles are always datapoints, are from our dataset, but Median or Quartiles can be midpoints of datapoints, not elements from our dataset.

We have that Median divides our sorted list of observations into two equal-length parts, so we could define the 0.5-quantile to be our Median. And we could take as $\frac{1}{4}$ and $\frac{3}{4}$-quantiles our first and third quartiles Q_1 and Q_3, as we have talked before that the 25% of observations are to the left of Q_1 and the rest are to the right of Q_1. And in some textbooks this is the case. But here, for the sake of simple formula for quantiles, we will use the definition (5.2) above, which can produce the described effect where 0.25, 0.5 and 0.75 quantiles are not the quartiles.

Hopefully, this will not cause much problems.

5.3 Quantile-Quantile (Q-Q) Plots

One of the standard and important problems of the Statistics is to check if the given data comes from some fixed distribution, say, from the Normal Distribution[5]. Or, another problem is to check if two datasets are generated from the same distribution.

In our course, we will consider two methods to check this kind of things. The first one is the graphical one, where we will get the answer visually. And the second method will be to do some test to check that - and we will talk about this later, when considering hypotheses testing topics.

Here, in this section we will describe one of the non-parametric ways to solve the described problems - the graphical method, Q-Q Plot method.

We will consider the following problems:

[5]Later, in the Inferential Statistics part of our course, we will deal with the Parametric Statistics. That is, we will assume that our data comes from some parametric family of distributions, and our aim will be to estimate that parameters. But where from we can guess that our data comes from that parametric family? - Here the Q-Q Plot method can be used!

Problem 1: We have 2 theoretical distributions (say, $\mathcal{N}(3,4)$ and $\mathrm{Exp}(2)$). We want to compare these distributions to see which one has fatter tails. Another problem is to see if the distributions are close to each other (say, $t(50)$ is close to $\mathcal{N}(0,1)$).

Problem 2: We have a theoretical distribution and a dataset. We want to see if the dataset is generated from the theoretical distribution.

Problem 3: We have two datasets, possibly, of different sizes. We want to see if the datasets are generated from the same distribution.

To approach the above problems, we will use the Q-Q Plot graphical method.

5.3.1 Q-Q Plot for two Distributions

Assume we are given two distributions, by their CDF-s $F(x)$ and $G(x)$.

Definition 5.4. *The **Q-Q Plot** for $F(x)$ and $G(x)$ is the plot of all points (q_α^F, q_α^G), where $\alpha \in (0,1)$, and q_α^F and q_α^G are the α-th quantiles of F and G, respectively.*

It is clear that in the case when F coincides with G, $F \equiv G$, the quantiles will be the same, so all points (q_α^F, q_α^G), $\alpha \in (0,1)$ will give the portion of the bisector $y = x$ on the graph.

Let us see what will happen if we will do the Q-Q Plot for the same family distributions.

EXAMPLE, Q-Q PLOT FOR UNIFORM VS UNIFORM: Let us do the Q-Q Plot for $\mathcal{A} = \mathtt{Unif}[0,1]$ and $\mathcal{B} = \mathtt{Unif}[0,5]$.

So we fix $\alpha \in (0,1)$, and calculate $q_\alpha^{\mathcal{A}}$ and $q_\alpha^{\mathcal{B}}$.

The quantile of order α of the $\mathcal{A} = \mathtt{Unif}[0,1]$ is the point that divides the area under the PDF curve into α and $1-\alpha$ portions (to the left and right, respectively). The PDF of the $\mathtt{Unif}[0,1]$ is the function $f_{\mathcal{A}}(x) = 1$ for $x \in [0,1]$ and $f_{\mathcal{A}}(x) = 0$ for $x \notin [0,1]$. So the α quantile will be $q_\alpha^{\mathcal{A}} = \alpha$. This can be seen also geometrically, using the graph of the PDF.

Another way to see this is to use the CDF of the $\mathtt{Unif}[0,1]$: the CDF has the form

$$F_{\mathcal{A}}(x) = \begin{cases} 0, & x < 0 \\ x, & x \in [0,1] \\ 1, & x > 1. \end{cases}$$

Now, the quantile of order α is the leftmost point q with $F_{\mathcal{A}}(q) = \alpha$, i.e., the intersection point of the line $y = \alpha$ and the graph of CDF $y = F_{\mathcal{A}}(x)$. Obviously, the intersection point will be unique: $q = \alpha$, since $F_{\mathcal{A}}(\alpha) = \alpha$ (as $F_{\mathcal{A}}(x) = x$ for $x \in [0,1]$).

Summarizing, $q_\alpha^{\mathcal{A}} = \alpha$.

Now, let us calculate the α quantile for the distribution $\mathcal{B} = \mathtt{Unif}[0,5]$.

Again we will use geometric ideas. We first find the PDF of $\mathtt{Unif}[0,5]$, which is

$$f_{\mathcal{B}}(x) = \begin{cases} \dfrac{1}{5-0} = \dfrac{1}{5}, & x \in [0,5] \\ 0, & x \notin [0,5]. \end{cases}$$

We want to find the point q such that the area left to the line $x = q$ under the PDF curve will be α. Clearly, $q \in [0,5]$ (think why?). Then the area to the left to $x = q$ will be

$$\text{Area} = \text{height} \times \text{width} = \frac{1}{5} \cdot (q - 0) = \alpha,$$

so $q = 5\alpha$. Hence, $q^{\mathcal{B}}_{\alpha} = 5\alpha$.

This means that for the Q-Q Plot we need to draw the points $(q^{\mathcal{A}}_{\alpha}, q^{\mathcal{B}}_{\alpha}) = (\alpha, 5\alpha)$ for $\alpha \in (0,1)$, which will give the line $y = 5x$ on the graph, see Fig. 5.6.

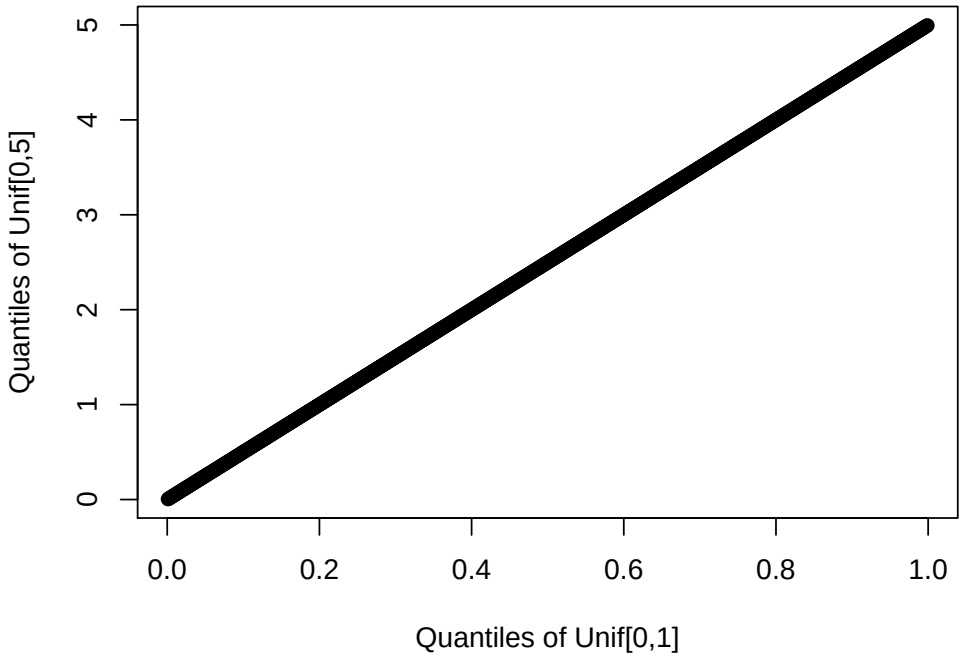


Fig. 5.6: Q-Q Plot for $\mathrm{Unif}[0,5]$ vs $\mathrm{Unif}[0,1]$

R code, Q-Q Plot for the $\mathrm{Unif}[0,5]$ vs $\mathrm{Unif}[0,1]$:

```
#Q-Q plot, Theoretical vs Theoretical
alpha <- seq(from = 0.001, to = 0.999, by = 0.001)
              #alpha is running from 0.001 to 0.999 with stepsize 0.001
x <- qunif(alpha, min = 0, max = 1)
              #quantiles of orders alpha for the above alpha-s for Unif[0,1]
y <- qunif(alpha, min = 0, max = 5)
              #quantiles of orders alpha for the above alpha-s for Unif[0,5]
plot(x,y, pch = 19, xlab = "Quantiles of Unif[0,1]", ylab = "Quantiles of Unif[0,5]")
```

Example, Q-Q Plot for Uniform vs Uniform, v2: Now, let us do the Q-Q Plot for the $\mathcal{A} = \mathrm{Unif}[1,4]$ and $\mathcal{B} = \mathrm{Unif}[3,9]$.

We again take $\alpha \in (0,1)$. Now, one can easily check (do the calculations!), using the ideas above,

that

$$q_\alpha^{\mathcal{A}} = 1 + 3\alpha \qquad \text{and} \qquad q_\alpha^{\mathcal{B}} = 3 + 6\alpha.$$

Now, the Q-Q Plot will consists of all points

$$(q_\alpha^{\mathcal{A}}, q_\alpha^{\mathcal{B}}) = (1 + 3\alpha, 3 + 6\alpha), \qquad \alpha \in (0, 1).$$

To describe this parametric graph, let us denote by $x = 1 + 3\alpha$ and $y = 3 + 6\alpha$. Then $\alpha = \frac{1}{3} \cdot (x - 1)$, so $y = 3 + 6\alpha = 3 + 6 \cdot \frac{1}{3} \cdot (x - 1) = 2x + 1$. So the graph will be some portion of the line (can you guess where the coefficients 2 and 1 come from in the line $y = 2x + 1$? Note that $4 - 1 = 3$ and $9 - 3 = 6$, so $9 - 3 = 2 \cdot (4 - 1)$ ☺)

$$y = 2x + 1,$$

for $x \in (1, 4)$ (since $\alpha \in (0, 1)$ and $x = 1 + 3\alpha$, then $x \in (1, 4)$). See Fig. 5.7.

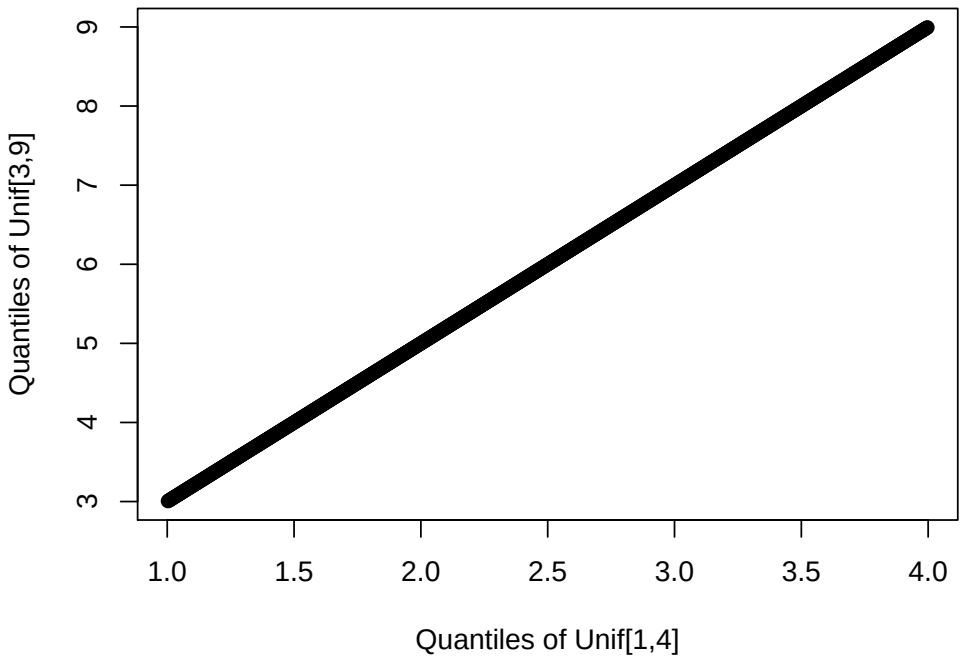


Fig. 5.7: Q-Q Plot for Unif$[1, 4]$ vs Unif$[3, 9]$

R code, Q-Q Plot, for Unif$[1, 4]$ vs Unif$[3, 9]$:

```
#Q-Q plot, Theoretical vs Theoretical
alpha <- seq(from = 0.001, to = 0.999, by = 0.001)
    #alpha is running from 0.001 to 0.999 with stepsize 0.001
x <- qunif(alpha, min = 1, max = 4)
    #quantiles of orders alpha for the above alpha-s for Unif[0,1]
```

```
y <- qunif(alpha, min = 3, max = 9)
    #quantiles of orders alpha for the above alpha-s for Unif[0,5]
plot(x,y, pch = 19, xlab = "Quantiles of Unif[1,4]", ylab = "Quantiles of Unif[3,9]")
```

Now, to give some idea about the interpretation of the Q-Q Plot, let us consider the example of Standard Normal Distribution $\mathcal{N}(0,1)$ and Cauchy Distribution[6] with the parameters $(0,1)$, $Cauchy(0,1)$. First we draw the PDFs of these distribution on the same figure, see Fig. 5.8. The code is:

R code, PDFs of $\mathcal{N}(0,1)$ and $Cauchy(0,1)$:

```
# PDFs for N(0,1) and Cauchy(0,1)
curve(dcauchy, xlim = c(-5,5), ylim = c(0,0.4), lwd = 2, col = "red",
      ylab = "PDFs of N(0,1) and Cauchy(0,1)")
par(new = TRUE)
curve(dnorm, xlim = c(-5,5),  ylim = c(0,0.4), lwd = 2, col = "blue",
      ylab = "PDFs of N(0,1) and Cauchy(0,1)")
legend(1.8, 0.38, c("N(0,1)", "Cauchy(0,1)"), lty = c(2,2), lwd = c(2,2),
       col = c("blue", "red"))
```

Clearly,

- Both distributions are symmetric around 0;
- The Cauchy Distribution PDF has fatter tails, meaning that it tends to 0 as $x \to \pm\infty$ much slower that the PDF of Standard Normal[7]

Now, let us calculate the $0.5, 0.7, 0.8$ and 0.9 quantiles for both distributions and plot the pairs of quantiles $(q_\alpha^{\mathcal{N}}, q_\alpha^{C})$ for $\alpha = 0.5, 0.7, 0.8, 0.9$. The **R** code is here:

R code, Quantiles for $\mathcal{N}(0,1)$ vs $Cauchy(0,1)$:

```
alpha = c(0.5, 0.7, 0.8, 0.9)
xx <- qnorm(alpha)
yy <- qcauchy(alpha)
xx
yy
plot(xx,yy, pch = 19, cex = 1.2, xlim = c(0,3.5), ylim = c(0,3.5),
    xlab = "Quantiles of N(0,1)", ylab = "Quantiles of C(0,1)")
abline(0,1, col = "green", lwd = 2)
```

The last command adds the line $y = x$ (in green) to the graph. The result is in Fig. 5.9.

[6] See https://en.wikipedia.org/wiki/Cauchy_distribution

[7] In fact, Cauchy distribution's PDF is given by $f_{Cauchy}(x) = \frac{1}{\pi(1+x^2)}$, which tends to 0 like $\frac{1}{x^2}$, as $x \to \pm\infty$, and the Standard Normal Distribution's PDF is given by $f_{StdNormal}(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-x^2/2}$, which tends to 0 exponentially fast.

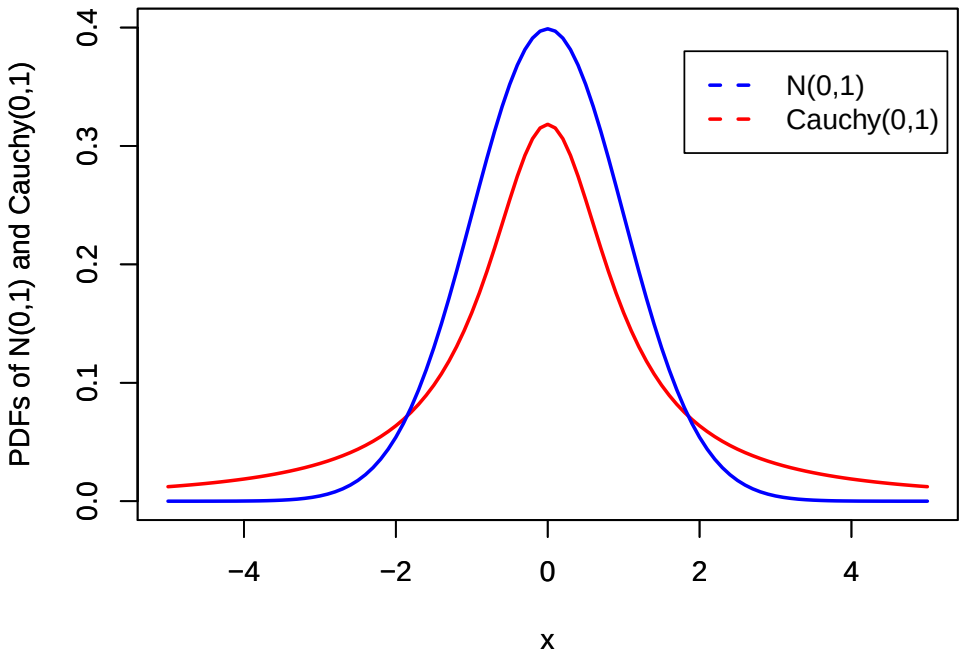


Fig. 5.8: PDFs of the $\mathcal{N}(0,1)$ (blue) and $Cauchy(0,1)$ (red)

You can see that $q^{\mathcal{N}}_{0.5} = q^{C}_{0.5} = 0$, $q^{\mathcal{N}}_{0.7} < q^{C}_{0.7}$, $q^{\mathcal{N}}_{0.8} = q^{C}_{0.8}$ and $q^{\mathcal{N}}_{0.9} = q^{C}_{0.9}$, and, in fact, $q^{\mathcal{N}}_{\alpha}$ grows much slower that q^{C}_{α} (can you explain what this statement means?). This means that the shape of the Q-Q Plot on the left will be *convex*. To explain why the quantiles of the Standard Normal grow much slower than the ones for Cauchy, we draw the CDF's and corresponding α levels, see Fig. 5.10. On the Fig. 5.10, the quantiles are the x-coordinates of the corresponding line $y = \alpha$ with the CDFs. Visually, when α increases, the quantile of the Cauchy Distribution (the intersection point of the line $y = \alpha$ with the red curve) grows faster that the quantile of the Standard Normal Distribution.

This phenomenon is specific for fatter-tailed distributions: if we draw the Q-Q Plot for two distributions $\mathcal{A}$ and $\mathcal{B}$, both distributions have right tails (are non-zero in $[a, +\infty)$ for some a), and the tails of $\mathcal{A}$ are thinner than the tails of $\mathcal{B}$, then the right-hand side of the Q-Q Plot (assuming that the quantiles of $\mathcal{A}$ are on the x-axis) will be convex-shaped. The inverse is true for the left-tailed distributions: if the left tail of $\mathcal{B}$ is fatter than the left tail of $\mathcal{A}$, then on the left-hand side of the Q-Q Plot (again assuming that the quantiles of $\mathcal{A}$ are on the x-axis) we will have a concave-shaped graph.

EXAMPLE, COMPARISON OF DISTRIBUTION TAILS WITH Q-Q PLOT: The Fig. 5.11 shows the Q-Q Plot for $\mathcal{N}(0,1)$ (quantiles are on the x-axis) and $Cauchy(0,1)$ (on the y-axis). $Caucy(0,1)$ has fatter tails on the left and right hand sides, so the Q-Q Plot shape is convex on the right-hand side, and concave on the left one. Also, you can clearly see the symmetry of quantiles (because of the symmetry of distributions).

Another example is in Fig. 5.12, where the Q-Q plot of $\mathcal{N}(0,1)$ vs $Exp(1)$ is given. Here again, $Exp(1)$ has fatter tails on the right hand side, hence the convex shape on the right. $Exp(1)$ does not have left tail (the PDF is 0, if $x < 0$), so the quantiles of $Exp(1)$ will approach $0+$, as $\alpha \to 0$, and the

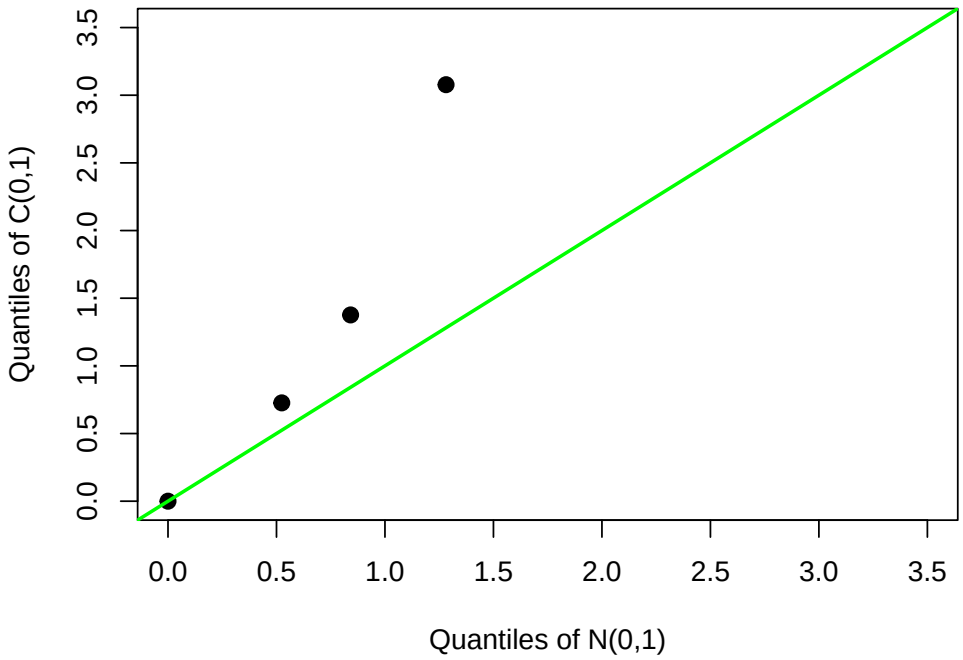


Fig. 5.9: $0.5, 0.7, 0.8$ and 0.9 quantiles of $\mathcal{N}(0,1)$ vs $Cauchy(0,1)$

quantiles of $\mathcal{N}(0,1)$ will tend to $-\infty$, when $\alpha \downarrow 0$ (can you explain this?).

Yet another example is in Fig. 5.13, where the Q-Q plot of $\mathrm{Exp}(1)$ vs $\mathrm{LogNormal}(0,1)$ is given (see https://en.wikipedia.org/wiki/Log-normal_distribution for the definition and properties of the LogNormal distribution). Here, LogNormal distribution has fatter tails compared to the Exponential.

R code, Q-Q Plot of $\mathcal{N}(0,1)$ vs $Cauchy(0,1)$:

```
# Q-Q Plot for N(0,1) and C(0,1), alpha runs from 0.01 to 0.99
alpha <- seq(from = 0.01, to = 0.99, by = 0.001)
xx <- qnorm(alpha)
yy <- qcauchy(alpha)
plot(xx,yy, type = "l", lwd = 3, xlab = "Quantiles of N(0,1)",
     ylab = "Quantiles of Cauchy(0,1)")
par(new = TRUE)
abline(0,1, col = "green", lwd = 3)
```

R code, Q-Q Plot of $\mathcal{N}(0,1)$ vs $Cauchy(0,1)$:

```
# Q-Q Plot for N(0,1) and Exp(1), alpha runs from 0.001 to 0.999
```

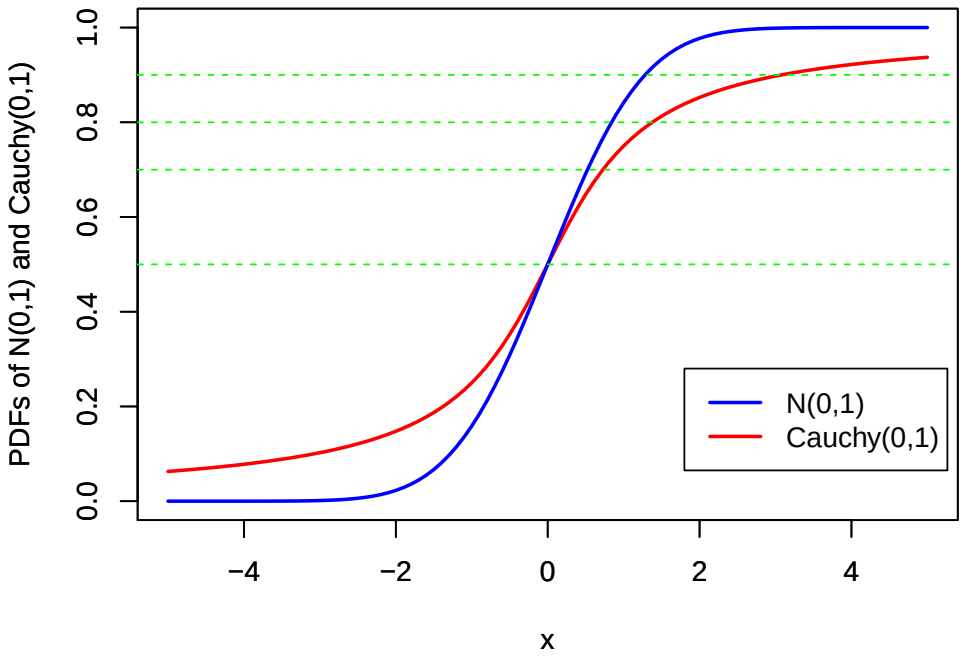


Fig. 5.10: The CDFs of $\mathcal{N}(0,1)$ vs $\mathrm{Cauchy}(0,1)$, and the lines $\alpha = 0.5, 0.7, 0.8$ and 0.9

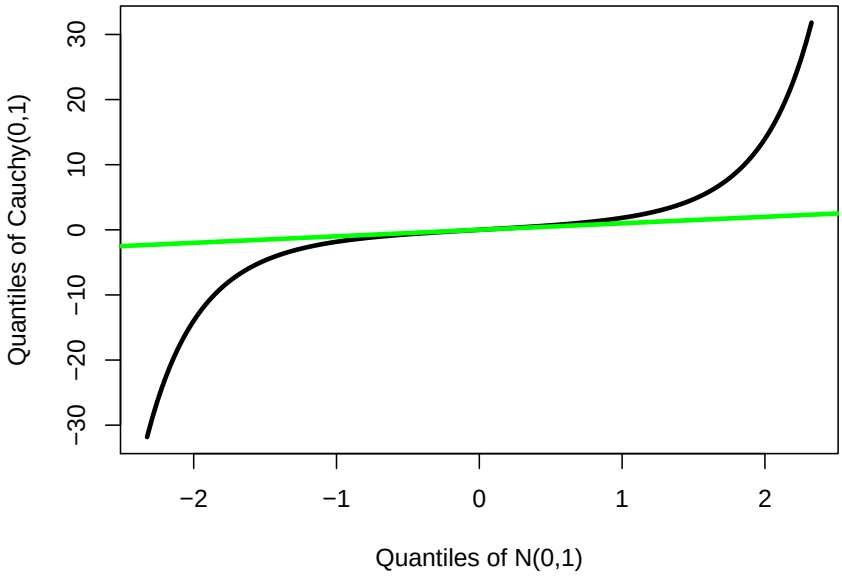


Fig. 5.11: Q-Q Plot of $\mathcal{N}(0,1)$ vs $\mathrm{Cauchy}(0,1)$

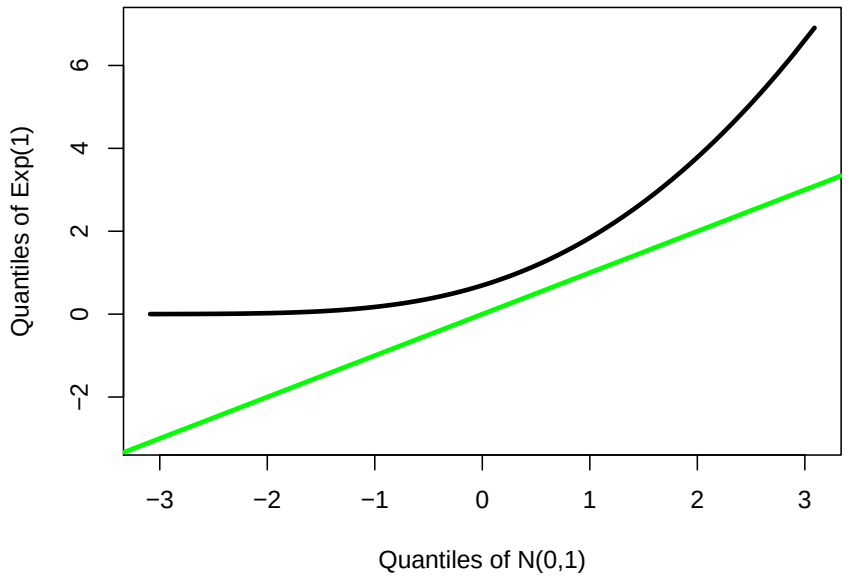


Fig. 5.12: Q-Q Plot of $\mathcal{N}(0,1)$ vs $\mathrm{Exp}(1)$

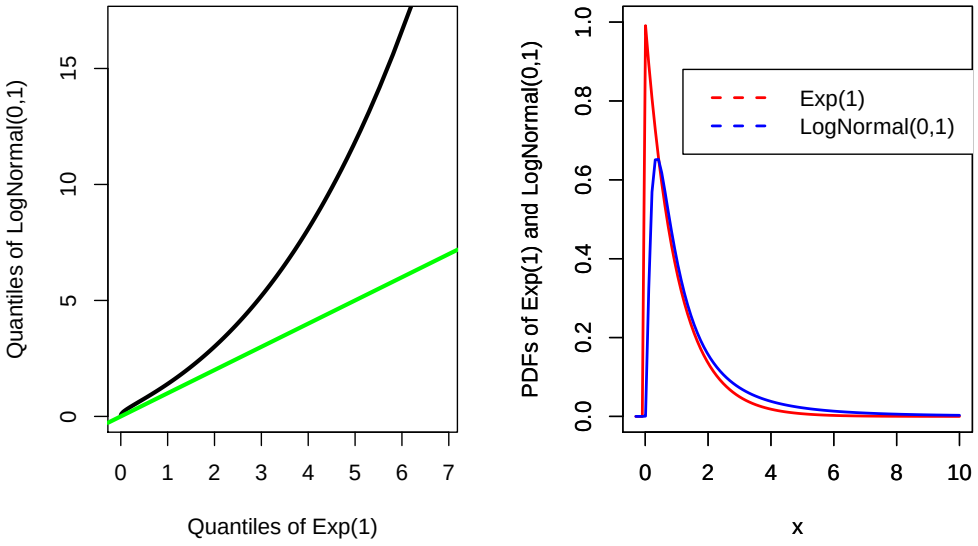


Fig. 5.13: Q-Q and PDF Plots of $\mathrm{Exp}(1)$ vs $\mathrm{LogNormal}(0,1)$

```
alpha <- seq(from = 0.001, to = 0.999, by = 0.001)
xx <- qnorm(alpha)
yy <- qexp(alpha)
plot(xx,yy, type = "l", lwd = 3, ylim = c(-3,7), xlab = "Quantiles of N(0,1)",
```

```
     ylab = "Quantiles of Exp(1)")
abline(0,1, col = "green", lwd = 3)
```

R code, Q-Q Plot of $\mathcal{N}(0,1)$ vs Cauchy$(0,1)$:

```
# Q-Q Plot for Exp(1) and LogNormal(0,1)
alpha <- seq(from = 0.001, to = 0.999, by = 0.001)
xx <- qexp(alpha)
yy <- qlnorm(alpha)
par(mfrow = c(1,2))
plot(xx,yy, type = "l", lwd = 3, ylim = c(0,17), xlab = "Quantiles of Exp(1)",
     ylab = "Quantiles of LogNormal(0,1)")
abline(0,1, col = "green", lwd = 3)
curve(dexp, xlim = c(-0.3,10), ylim = c(0,1), lwd = 2, col = "red",
      ylab = "PDFs of Exp(1) and LogNormal(0,1)")
par(new = TRUE)
curve(dlnorm, xlim = c(-0.3,10),  ylim = c(0,1), lwd = 2, col = "blue",
      ylab = "PDFs of Exp(1) and LogNormal(0,1)")
legend(1.2, 0.88, c("Exp(1)", "LogNormal(0,1)"), lty = c(2,2), lwd = c(2,2),
       col = c("red", "blue"))
```

Remark, Q-Q Plot:

- It is easy to see that the Q-Q Plot has the shape of a graph of an increasing function;
- Unfortunately, on the Q-Q Plot, one cannot identify some specific quantile, say, having the Q-Q Plot, one cannot find the 20% quantile for the distributions, or, say, the Medians. Unfortunately, we are not showing α on the graph

Example, t-distribution and Standard Normal Distribution: Student's t Distribution is one of the important distributions in Statistics. We will define and meet this distribution a lot of times in the rest of our Stat course. t distribution comes with a parameter called the degrees of freedom, $t(n)$ is the t distribution with n degrees of freedom[8] Here we want to compare $t(n)$ with $\mathcal{N}(0,1)$. The idea is that for large n, $t(n)$ is very close to $\mathcal{N}(0,1)$, for example, as a rule of thumb (that you will find in many Stat textbooks), if $n \geqslant 30$, then one is using $\mathcal{N}(0,1)$ as an approximation of $t(n)$.

Now, let us give the Q-Q Plots for $t(n)$ vs $\mathcal{N}(0,1)$ for different n-s. You can find the plots in Fig. 5.14-5.19.

5.3.2 Q-Q Plot for a Dataset vs Distribution

Assume here that we have a dataset x, and a fixed distribution given by its CDF $F(x)$. Our task is to check if the dataset is coming from the distribution defined by F or not. To check this graphically, we are using the Q-Q Plot defined below:

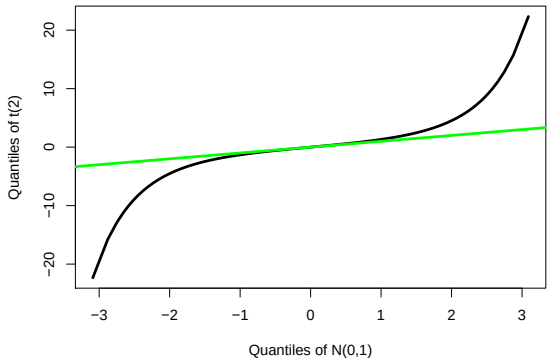


Fig. 5.14: Q-Q Plot: $t(2)$ vs $\mathcal{N}(0,1)$

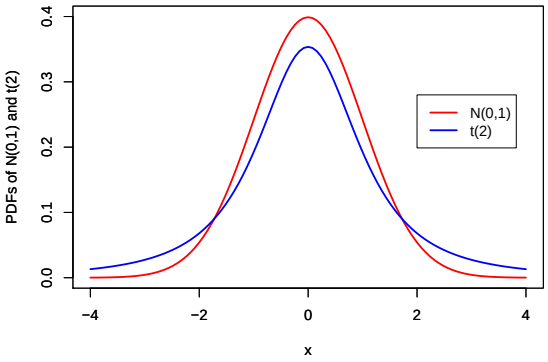


Fig. 5.15: PDF Plot: Plot: $t(2)$ vs $\mathcal{N}(0,1)$

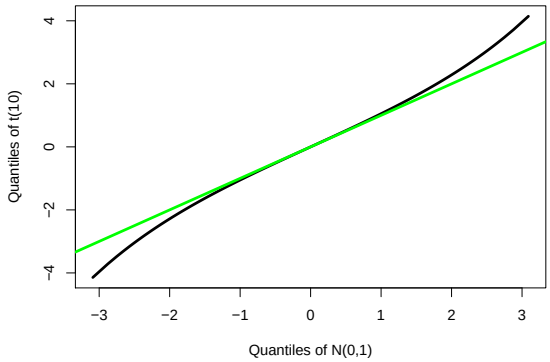


Fig. 5.16: Q-Q Plot: $t(10)$ vs $\mathcal{N}(0,1)$

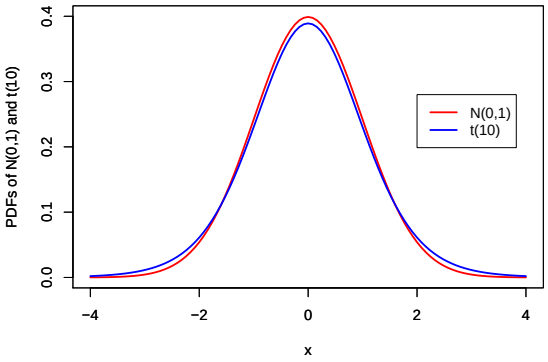


Fig. 5.17: PDF Plot: Plot: $t(10)$ vs $\mathcal{N}(0,1)$

Definition 5.5. *The **Q-Q Plot** for the dataset x and distribution $F(x)$ is the plot of all points (q^F_α, q^x_α), where α runs over some values in $(0,1)$, and q^F_α and q^x_α are the α-th quantiles of the distribution F and the dataset x, respectively.*

Usually, one uses $\alpha = \frac{k}{n}$-th quantiles, or $\alpha = \frac{k}{n+1}$-th $(k = 1, ..., n)$ quantiles[9] or $\alpha = \frac{k-0.5}{n}$-th quantiles, where n is the size of our dataset[10].

The interpretation of the Q-Q Plot obtained this way is just the same as for the previous case, Theoretical-Theoretical Q-Q Plot.

R code, Financial Returns are not Normally Distributed: It is somehow classical fact now that the returns of stocks are not following a Normal Distribution.

Let us consider here an example to explain the claim: we consider some stock, say, FB stock, and we want to see if their weekly rates of returns are Normally Distributed or not. To that end we want to use the Q-Q Plot.

We start by downloading the data: we navigate to `http://finance.yahoo.com/`. In the search bar we enter "FB" and chose the "FB, Facebook, Inc.". The first page will show some basic info about

[9] This is the most used one, for other choices see Wikipedia, `https://en.wikipedia.org/wiki/Q-Q_plot`

[10] No need to take more that n quantiles, since if we have n data points, then taking more than n quantiles will not give new ones.

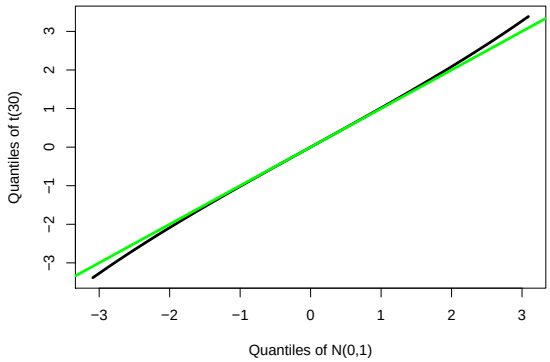


Fig. 5.18: Q-Q Plot: $\mathrm{t}(30)$ vs $\mathcal{N}(0,1)$

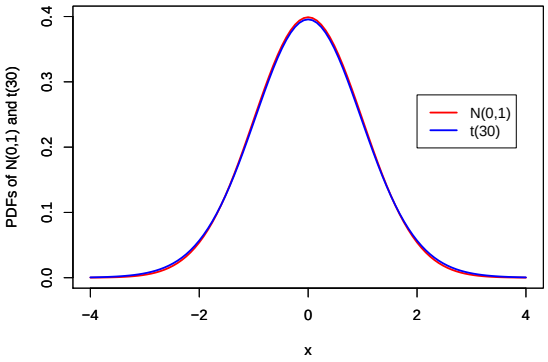


Fig. 5.19: PDF Plot: Plot: $\mathrm{t}(30)$ vs $\mathcal{N}(0,1)$

the company and stock, and, particularly, the current price for 1 share. Then we go to "Historical Data", choose the Time Period "Max" (this will download all available data), choose the Frequency "Weekly" (because we want to calculate weekly returns), hit "Apply", and then choose "Download Data". This will download the historical price data for FB stock to *FB.csv* file (.csv stands for the Comma Separated File). This file can be viewed in Excel. It has a header, the top row, with the names of variables (features) - "Date", "Open", "High", "Low", "Close", "Adj Close", "Volume". Here "Date" is the date ☺, "Open" is the price at the very beginning of that week, "Close" is the price at the very end of the week, "High" is the highest price during that week, "Low" is the lowest price for that week, "Adj Close" is the price at the end of the week, adjusted, if dividend payments or splits happened during that week. "Volume" is the number of shares traded (bought or sold) that week. We will use the "Adj Close" Prices, and will do our calculations in **R**.

So first, we read import the dataset into **R**. To that end, we will use the command

```
xx <- read.csv(file.choose(), header = TRUE)
```

Here *read.csv* is obviously to read the .csv file, *file.choose()* is to open the "Open" dialog to choose the .csv file (otherwise, you need to specify the path to that file), and *header = TRUE* is to indicate that in our .csv file the first row is the header, it shows the variables names.

After running this command, *xx* will be a *data frame* having the same structure as the .csv file. You can see the content of *xx* just by clicking on it in the R-Studio's Environment tab.

Now, we select the "Adj Close" column values:

... To Be Continued ....

```
#Financial returns Q-Q plot and non-Normality
dataset <- read.csv(file.choose(), header = TRUE)
adjcloseprices <- dataset$Adj.Close
rate_of_ret <- diff(adjcloseprices)/adjcloseprices[1:(length(adjcloseprices)-1)]
hist(rate_of_ret)
qqnorm(rate_of_ret)
qqline(rate_of_ret)
```

5.3.3 Q-Q Plot for two Datasets

The third possible Q-Q plot is the plot for two datasets. The problem is that we want to check how similar are our datasets, and we want to check if they are coming from the same distribution. To that end, we plot in 2D the quantiles of the first dataset vs the quantiles of the second one.

Definition 5.6. *The **Q-Q Plot** for datasets x and y is the plot of all points (q^y_α, q^x_α), where α runs over some values in $(0,1)$, and q^y_α and q^x_α are the α-th quantiles of the datasets y and the x, respectively.*

As above, usually one uses $\alpha = \frac{k}{n}$-th quantiles, or $\alpha = \frac{k}{n+1}$-th ($k = 1, ..., n$) quantiles, where n is the minimum of the lengths of datasets x and y. In fact, for Q-Q Plot, we can have that x contains more datapoints then y or vice-versa.

The interpretation of the Q-Q Plot is similar to the previous cases. If the datasets are coming from the same distribution, then the Q-Q Plot will show points well-aligned with the line $q^y = q^x$, the bisector. If the Q-Q Plot is well-aligned with some line which is parallel to $q^y = q^x$, then the datasets, most probably[11], have the same distribution but with some shifted location parameter (e.g., one is from $\mathcal{N}(0,1)$, and the other is from $\mathcal{N}(-2,1)$). And if the datasets are well aligned around some other line, then, most probably, they are from the same distributions, but with different scale and location parameters. The convex or concave shapes on the right or left are signaling about the heavier or lighter tails.

Fig. 5.20-5.23 below show some experiments in **R** for the Q-Q Plots: the datasets are from the same distributions. Fig. 5.24 is a Q-Q Plot for datasets from different distributions. The code is given below[12]

R code, Experiments with Q-Q Plot for 2 Datasets:

```
#Q-Q Plot for 2 Datasets, experiment no. 1
x <- rnorm(100, mean = 0, sd = 1)
y <- rnorm(200, mean = 0, sd = 1)
qqplot(x,y, pch = 16)
abline(0,1, col = "red", lwd = 2)

#Q-Q Plot for 2 Datasets, experiment no. 2
x <- rnorm(100, mean = 0, sd = 1)
y <- rnorm(200, mean = -0.7, sd = 1)
qqplot(x,y, pch = 16)
abline(0,1, col = "red", lwd = 2)
abline(-0.7, 1, col = "green", lwd = 2)

#Q-Q Plot for 2 Datasets, experiment no. 3
x <- rnorm(100, mean = 0, sd = 1)
y <- rnorm(200, mean = -0.7, sd = 5)
qqplot(x,y, pch = 16)
abline(0,1, col = "red", lwd = 2)
abline(-0.7,5, col = "green", lwd = 2)
```

[11] I am using "most probably", because I cannot say for sure, and nobody can say for sure!

[12] of course, if you will run these codes, you will not get exactly the same picture, because every time computer generates different random samples. We can fix the random sample by using the command *set.seed(n)*, where n is some number.

```
#Q-Q Plot for 2 Datasets, experiment no. 4
x <- rexp(100, rate = 4)
y <- rexp(100, rate = 10)
qqplot(x,y, pch = 16)
abline(0,1, col = "red", lwd = 2)
abline(0,0.4, col = "green", lwd = 2)

#Q-Q Plot for 2 Datasets, experiment no. 5
x <- rnorm(200)
y <- rexp(100, rate = 4)
qqplot(x,y, pch = 16)
```

Please note that the lengths of x and y differ in our code, except the 4th experiment, when, after doing a copy-paste, I forgot to change the number of samples ☺

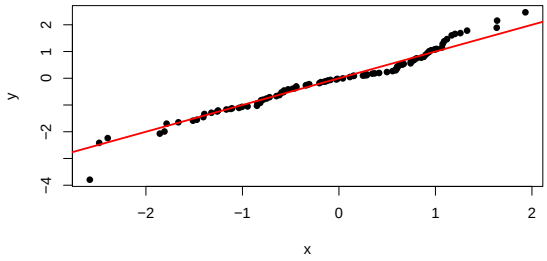


Fig. 5.20: Q-Q Plot: x and y are from $\mathcal{N}(0,1)$, red line is the line $q^y = q^x$, the bisector

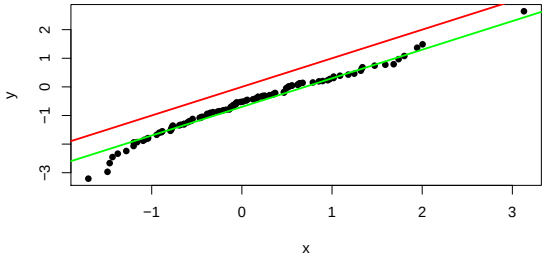


Fig. 5.21: Q-Q Plot: x is generated from $\mathcal{N}(0,1)$, y is from $\mathcal{N}(-0.7,1)$. Red line is the line $q^y = q^x$, and green line is $q^y = q^x - 0.7$

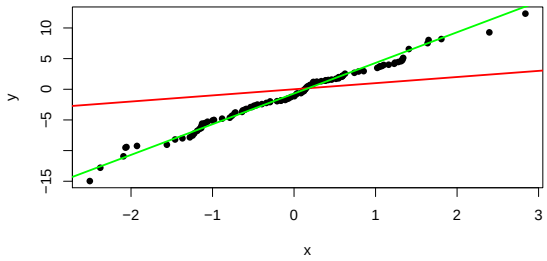


Fig. 5.22: Q-Q Plot: x is generated from $\mathcal{N}(0,1)$, y is from $\mathcal{N}(-0.7,5^2)$. Red line is the line $q^y = q^x$, and green line is $q^y = 5 \cdot q^x - 0.7$

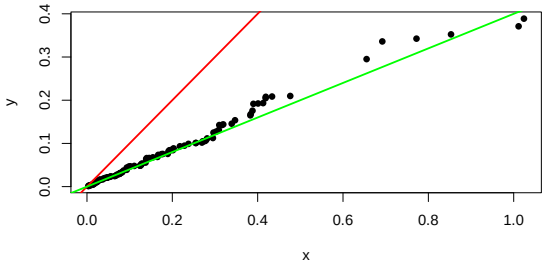


Fig. 5.23: Q-Q Plot: x is generated from $\mathrm{Exp}(4)$, y is from $\mathrm{Exp}(10)$. Red line is the line $q^y = q^x$, and green line is $q^y = \frac{4}{10} q^x$.

REMARK, Q-Q PLOTS:

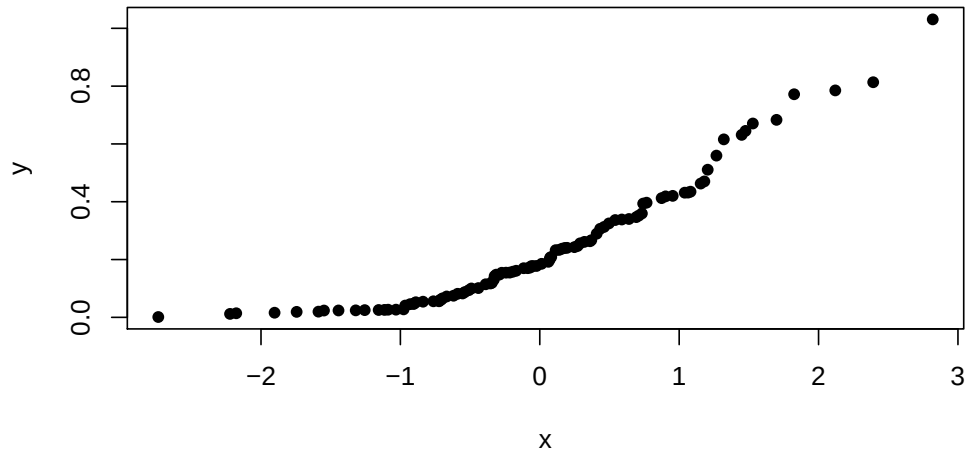


Fig. 5.24: Q-Q Plot: x is generated from $\mathcal{N}(0,1)$, y is from $\mathrm{Exp}(4)$

- If we have 2 datasets of the same size n, x and y, then, if we will use many α-s for the quantile orders, in some sense the Q-Q plot of that dataset will be the plot of points $(x_{(i)}, y_{(i)})$, $i = 1, ..., n$.
- Please note that in Q-Q Plot we are not graphing the data values, rather we plot quantiles. So you will not be able to recover data values from the Q-Q Plot.

REMARK, INTERPRETATION OF Q-Q PLOTS : Nice interpretation for Q-Q Plots is given at http://stats.stackexchange.com/questions/101274/how-to-interpret-a-qq-plot.

REMARK, INTERPRETATION OF Q-Q PLOTS : It can be easily seen that for a (continuous) distribution with a PDF $f(x)$, which has a compact support[13], say, $[a, b]$, then the quantiles of that distribution will be in $[a, b]$, and, moreover,

$$\lim_{\alpha \to 0+} q_\alpha = a, \qquad \lim_{\alpha \to 1-} q_\alpha = b.$$

!!Give here for the Uniform and Beta distributions

Please note also that the quantiles are increasing function of the quantile order, i.e., if α increases, then q_α increases too. So in any case, the limit of q_α, when $\alpha \to 1-$, exists (either finite or infinite). If $\lim_{\alpha \to 1-} q_\alpha = +\infty$, then the distribution has a right tail (extends to $+\infty$, in some sense), and if the limit is finite, then the distribution's support is bounded above (by that limit). The same is true also for the limit $\lim_{\alpha \to 0+} q_\alpha$, which exists and is either $-\infty$ or a finite number.

Exercise: Write **R** functions **qqunif**, **qqexp** that will do similar things like **qqnorm**.

Exploratory Data Analysis for Bivariate Data: Covariance and Correlation

Now assume we have two datasets of the same size for the variables x and y:

$$x_1, ..., x_n \qquad \text{and} \qquad y_1, ..., y_n,$$

and we want to explore the dependencies between that variables x and y. Say, x is the height of a person and y is the width of the same person (or the salary ⌣̈. Btw, how to calculate the width of a person ? ⌣̈). Or, say, we want to find a relationship between the time spent in FaceBook and Statistics Grade, or the (stroong!) relationship between the number of missed Stat classes and Stat Grade.

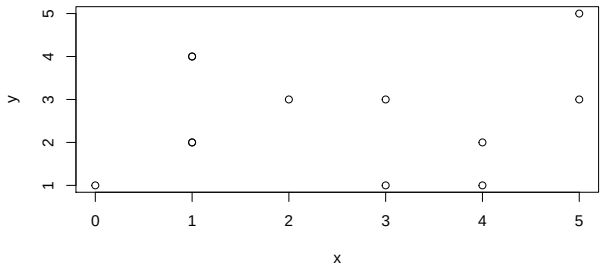
Here, like in 1D case, we will describe two methods - Geometric methods to visualize the relationship, and some numerical measures for that.

6.1 Visualizing the Data: ScatterPlot or the Point Cloud

One of the natural methods to visualize the data is to draw y vs x, i.e., to draw the points (x_i, y_i) for $i = 1, ..., n$:

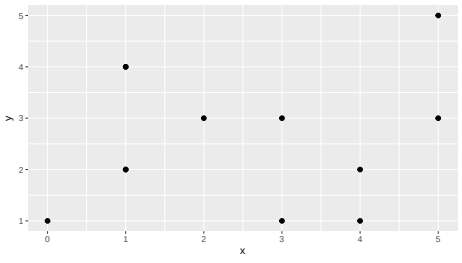
R code, Scatter Plot:

```
#Scatterplot or Points Cloud
x <- c(0,1,2,3,1,4,1,5,1,5,4,3)
y <- c(1,2,3,3,2,1,4,5,4,3,2,1)
plot(x,y)
```



R code, Scatter Plot, with ggplot2:

```
#Scatterplot or Points Cloud with ggplot2 library
library(ggplot2)
x <- c(0,1,2,3,1,4,1,5,1,5,4,3)
y <- c(1,2,3,3,2,1,4,5,4,3,2,1)
z <- data.frame(x,y)
ggplot(z, aes(x=x, y=y)) + geom_point(size=2)
```



In this case we assume that the observation x_i is related somehow (or maybe unrelated) to y_i, **with the same index** i. Say, x_1 and y_1 are two features of the same object (e.g., the height and age

of the person no. 1; or the stock price for General Electric Stock at some time instant and the value of the DJIA Index at the same time). So we plot y_i vs x_i. And if you will shuffle the datasets, the scatter plot will not be the same!

6.2 Sample Covariance and the Correlation Coefficient

Now we want to give some numerical measure of relationship between our datasets x and y. Recall from the Probability course that the covariance and the correlation coefficient are measures for the linear relationship between two r.v.'s. Now, for our observations, we define similar notions[1]:

Definition 6.1. *The Sample Covariance of the datasets x and y is*

$$\operatorname{cov}(x,y) = s_{xy} = \frac{\sum_{k=1}^{n}(x_k-\overline{x})\cdot(y_k-\overline{y})}{n}$$

or

$$\operatorname{cov}(x,y) = s_{xy} = \frac{\sum_{k=1}^{n}(x_k-\overline{x})\cdot(y_k-\overline{y})}{n-1}$$

Here $\overline{x}$ and $\overline{y}$ are the sample means for the datasets x and y.

Definition 6.2. *We say that datasets x and y are **uncorrelated**, if $\operatorname{cov}(x,y)=0$.*

Remark, Uncorrelatedness and Independence: In probability theory, we also have the notion of independence. And we are then describing the relationship between these two notions: independence and uncorrelatedness. And in Probability Theory, if X and Y are independent r.v.s, then they are also uncorrelated. The inverse statement is not true in the general case.

Here, for the datasets, the notion of independence is not defined.

You can see that, as in the case of the sample variance, we introduce 2 different formulas for the sample covariance. And, as in the case of the sample variance, different authors use either the first or the second one. Later we will explain why sometimes it is preferable to choose $n-1$ as a denominator instead of n.

Example, Sample Covariance: Assume we are given the following datasets:

$$x: 1,2,3,1,2,3,4,3,2,4,5, \qquad \text{and} \qquad y: -1,2,3,-1,-1,0,0,2,3,4,1.$$

Then you can surely calculate the covariance between x and y ☺

One of the drawbacks in covariance is that it can be any number, anything from $-\infty$ to $+\infty$, and, when comparing the relationships between 2 pairs of datasets, we cannot use covariances. I

[1] Recall again, that there is no uncertainty in our case here, there is no anything probabilistic in our observation yet: we just have some numbers recorded. Of course, we can make from that numbers a r.v. taking the recorded values with equal probabilities. This will explain the introduction of the sample covariance and sample correlation coefficient.

mean, if we have 2 pairs of datasets, (x, y) is the first pair, and (z, t) is the other pair of datasets, and we know that $cov(x, y)$ is very large compared to $cov(z, t)$, that will not show that the relationship between x and y is stronger that the relationship between z and t. Even worse, if z and t will be the same as x and y, respectively, but with other units of measurements, then $cov(x, y)$ will not be equal to $cov(z, t)$.

EXAMPLE, COVARIANCES FOR TWO DATASET PAIRS: Here we need to have an example of calculation.

The normalized version of the covariance is the correlation coefficient.

Definition 6.3. *The Sample Correlation Coefficient of the datasets x and y is*

$$cor(x, y) = \rho_{xy} = \frac{cov(x, y)}{\sqrt{Var(x) \cdot Var(y)}} = \frac{cov(x, y)}{sd(x) \cdot sd(y)} = \frac{s_{xy}}{s_x \cdot s_y},$$

where s_x and s_y are the standard deviations for x and y, respectively.

If $s_x = 0$ or $s_y = 0$, then we take $cor(x, y) = 0$ by definition.

Important is to remember to take the same denominator for the covariance and standard deviations - either n everywhere or $n - 1$ everywhere. So it is not correct to calculate the covariance using n in the denominator, then take $n - 1$ when calculating the standard deviations, and then calculate the correlation coefficient.

In both cases, when one calculates Standard Deviations and Covariance by using n simultaneously or $n - 1$ simultaneously in the denominator, we will obtain

$$cor(x, y) = \rho_{xy} = \frac{\sum_{k=1}^{n} (x_k - \overline{x}) \cdot (y_k - \overline{y})}{\sqrt{\sum_{k=1}^{n} (x_k - \overline{x})^2 \cdot \sum_{k=1}^{n} (y_k - \overline{y})^2}}$$

Another formula to calc the correlation coefficient is

$$cor(x, y) = \rho_{xy} = \frac{\sum_{k=1}^{n} x_k y_k - n \cdot \overline{x} \cdot \overline{y}}{\sqrt{\sum_{k=1}^{n} x_k^2 - n \cdot (\overline{x})^2} \cdot \sqrt{\sum_{k=1}^{n} y_k^2 - n \cdot (\overline{y})^2}}.$$

Note: Again Cov and Cor can be interpreted as the cov and cor for r.v.'s X and Y taking the values x_i and y_i, correspondingly, with the probabilities $\frac{1}{n}$ (or $\frac{1}{n-1}$).

What are measuring covariance and correlation coefficient - they are giving us some "measure of linear dependence" between x and y, a "measure of joint linear variability", the strength and the direction of the linear relationship between the data. If, say, we get $\rho_{xy} = 0$, then x and y are **uncorrelated**, and we mean that there is no (linear) relationship between x and y. Soon we will see that if $\rho_{xy} = 1$, then there is an exact linear and increasing relationship between x and y, and if ρ_{xy} is very close to 1, then there is a strong linear increasing relationship between x and y.

```
#Covariance and Correlation
x <- rnorm(40)
y <- rnorm(40)
plot(x,y)
cov(x,y)
cor(x,y)
```

Now, if $cov(x, y)$ or ρ_{xy} are positive, the we say that x and y are positively correlated. This means, roughly[2], if $x_k > \bar{x}$, then also y_k tends to be larger than $\bar{y}$. So there is a tendency: if x increases, then y tends to increase also.

```
#Covariance and Correlation, positive correlation
x <- rnorm(40)
e <- rnorm(40)
y <- 2.5*x+e
plot(x,y)
cov(x,y)
cor(x,y)
```

```
#Covariance and Correlation, negative correlation
x <- rnorm(40)
e <- rnorm(40)
y <- -1.4*x+e
plot(x,y)
cov(x,y)
cor(x,y)
```

Example:

```
#Covariance and Correlation, real data
state.x77
state <- as.data.frame(state.x77)
str(state) #structure of the dataset state
head(state)
tail(state)
x <- state$Illiteracy
y <- state$Murder
plot(x,y)
cov(x,y)
cor(x,y)
```

and

```
#Covariance and Correlation, real data, cont
state <- as.data.frame(state.x77)
x <- state$Illiteracy
y <- state$‘Life Exp‘
plot(x,y)
cov(x,y)
cor(x,y)
```

[2]Veery roughly!

Example: See https://en.wikipedia.org/wiki/Pearson_correlation_coefficient or https://en.wikipedia.org/wiki/Correlation_and_dependence for some graphical examples.

The difference between cov and cor is that cor is normalized, in the sense that

Proposition 6.1. *For any datasets* x, y,

$$-1 \leqslant \rho_{xy} \leqslant 1.$$

Moreover,

- $\rho_{xy} = 1$ *iff there exists a constant* $a > 0$ *and* $b \in \mathbb{R}$ *such that*[3] $y_i = a \cdot x_i + b$ *for any* $i = 1, ..., n$.
- $\rho_{xy} = -1$ *iff there exists a constant* $a < 0$ *and* $b \in \mathbb{R}$ *such that*[4] $y_i = a \cdot x_i + b$ *for any* $i = 1, ..., n$.

Exercise: Prove this Proposition.

Example: For example, the correlation coefficient of a dataset x with itself gives 1, i.e.,

$$cor(x, x) = \rho_{xx} = 1,$$

and the covariance of a dataset with itself is the variance:

$$cov(x, x) = s_{xx} = var(x) = s_x^2. \blacksquare$$

Another important aspect of the correlation coefficient is that it is dimensionless, it is independent on the units we are calculating the data x and y. Say, if we measure the weight x in Kg's and the height y of a person in meters, then we will obtain some number for the covariance between x and y. If we will change our units to grams for x and centimeters for y, the covariance will be another number (some multiple of the previous one). But in these both cases, the correlation coefficient will be the same.

REMARK, CORRELATION AND CAUSATION: It is important to note that high correlation between two datasets x and y doesn't mean that there is a causal relationship. In general, it is not true that x influences y or y influences x. It may be the case that some other feature, called a latent feature, is influencing both x and y.

See https://en.wikipedia.org/wiki/Correlation_does_not_imply_causation.

6.3 Appendix: Sample Statistics and Random Variable characteristics

As we have seen above, many descriptive statistics measures are the analogues for the corresponding ones form the Probability Theory. For example, the idea of the Sample Mean:

Sample Covariance: Recall that for the jointly distributed r.v. X and Y, the covariance $Cov(X, Y)$ is defined as

$$Cov(X, Y) = \mathbb{E}\big((x - \mathbb{E}(X))(Y - \mathbb{E}(Y))\big).$$

Now, let us obtain from this the sample covariance formula. Assume we have 2 datasets $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_n$.

Let X be a discrete r.v. taking the values $x_1, x_2, ..., x_n$ (it is possible, of course, that some x_i's coincide, but this is OK for us), and Y be a discrete r.v. with values $y_1, y_2, ..., y_n$. Now, we need to

[3] Or $x_i = a \cdot y_i + b$ for any $i = 1, ..., n$ (maybe for another a and b).
[4] Or $x_i = a \cdot y_i + b$ for any $i = 1, ..., n$ (maybe for another a and b).

describe the probabilities of taking that values. If we will describe marginal (individual) PMF's of X and Y, that wil not be enough for calculating the $\mathrm{Cov}(X,Y)$. For this calculation, we need to have the Joint PMF of X and Y. We define

Table 6.1: The PMF of X and Y

$Y \setminus X$	x_1	x_2	$\cdots$	x_n
y_1	$\frac{1}{n}$	0	$\cdots$	0
y_2	0	$\frac{1}{n}$	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	0	0	$\cdots$	$\frac{1}{n}$

So we give the equal probabilities for the value (x_k, y_k), $k = 1, ..., n$, and also we assume that the event $X = x_1$, $Y = y_3$ is impossible - x_k and y_k are linked to each other, if we observe x_1, then we observe y_1 (say, x_1 is the year of study of a person, and y_1 is his/her salary). This can be written also in the form:

$$\mathbb{P}(X = x_i, Y = y_j) = \begin{cases} \frac{1}{n}, & \text{if} \quad i = j \\ 0, & \text{if} \quad i \neq j. \end{cases}$$

Now, clearly,

$$\mathbb{E}(X) = \frac{\sum_{k=1}^{n} x_k}{n} = \overline{x}, \qquad \text{and} \qquad \mathbb{E}(Y) = \frac{\sum_{k=1}^{n} y_k}{n} = \overline{y}$$

and if we will calculate the covariance $\mathrm{Cov}(X,Y)$, then we will obtain

$$\mathrm{Cov}(X,Y) = \sum_{i,j=1}^{n} (x_i - \overline{x})(y_j - \overline{y}) \cdot \mathbb{P}(X = x_i, Y = y_j) = \frac{1}{n} \cdot \sum_{k=1}^{n} (x_k - \overline{x})(y_k - \overline{y}).$$