Lab 01

R and Statistics

CsciE63 Big Data Analytics

Zoran B. Djordjević

Objectives

- Continue to familiarizing ourselves with R;
- Learn most common statistical terminology;
- Learn some of standard routines for establishing most common statistical measures.

Categorical (Qualitative) Data

- A data sample is called categorical, or qualitative, if its values belong to a collection of known, defined, non-overlapping classes. Common examples include student letter grades (A, B, C, D or F), and commercial bond ratings (AAA, AAB,...), human gender (M,F,N,TG)
- R built-in data frame named `painters` is a compilation of scores (grades) on a few classical painters. The data set belongs to the MASS package, and has to be pre-loaded into the R workspace prior to use.

```
> library(MASS)
> head(painters)
              Composition Drawing Colour Expression School
Da Udine               10       8     16          3      A
Da Vinci               15      16      4         14      A
Del Piombo              8      13     16          7      B
Del Sarto              12      16      9          8      A
Fr. Penni               0      15      8          0      D
Guilio Romano          15      16      4         14      A
```


Column `School` contains Categorical Data

- The last column, `School`, contains the information on school classification of the painters. The schools are named as A, B, ...,H, and the School variable is categorical.

```
> school = painters$School; school
 [1] A A A A A A A A A A B B B B B B C C C C C C D D D D D D D
D D D E E E
[36] E E E E F F F F G G G G G G G H H H H
Levels: A B C D E F G H
> help(painters)
```

- This is a subjective assessment, on a 0 to 20 integer scale, of 54 classical painters. The painters were graded on four characteristics: `composition, drawing, colour` and `expression.`
- The school to which a painter belongs, is indicated by a factor level code: `"A": Renaissance; "B": Mannerist; "C": Seicento; "D": Venetian; "E": Lombard; "F": Sixteenth Century; "G": Seventeenth Century; "H": French.`
- `Composition, Drawing, Colour, and Expression` represent subjective measures of individual painters by an art critic, de Piles.

Frequency Distribution of Categorical Data

- In the data set `painters`, the frequency distribution of the School variable is a summary of the number of painters in each school.
- Frequency distribution is determined with R function `table()`

```
> school.freq = table(school)
> school.freq
school
A  B  C  D  E  F  G  H
10  6  6 10  7  4  7  4
```

- To represent the results as a column, use function `cbind()`

```
> cbind(school.freq)
  school.freq
A          10
B           6
C           6
D          10
E           7
F           4
G           7
H           4
```

@Zoran B. Djordjevic 5

Relative Frequency Distribution of Categorical Data

- The relative frequency distribution is the proportion with which a particular category participates in the total population of all samples.
- The relationship between relative frequency and frequency is given by the ratio:

$$Relative\ Frequency = \frac{\boldsymbol{Frequency}}{\boldsymbol{Sample\ Size}}$$

- We find the sample size of data set painters with R function `nrow()`. The relative frequency distribution is then determined :

```
> school.relfreq = school.freq / nrow(painters)
> school.relfreq
 school
         A          B          C          D          E          F
0.636319 0.11111111 0.11111111 0.636319 0.12962963 0.07407407
         G          H
0.12962963 0.07407407
```

- Please note, the sum over all relative frequencies is equal to 1

@Zoran B. Djordjevic 6

Making Relative Frequencies More Readable

- We can print with fewer digits by using function `options()`

```
> old = options(digits=1)
> old        # If you care to know what went into variable old
 $digits
[1] 7
```

- We apply function `cbind()` to print the result in column format.

```
> old = options(digits=1)
> cbind(school.relfreq)
  school.relfreq
A           0.19
B           0.11
C           0.11
D           0.19
E           0.13
F           0.07
G           0.13
H           0.07
> options(old) # Restore old options
```

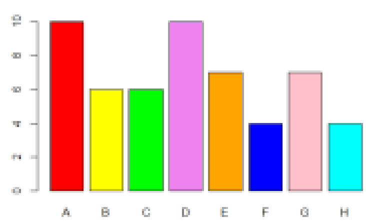

Bar Graph

- The bar graph of the School variable is a collection of vertical bars showing the number of painters in each school.
- We use function `barplot()` to produce the bar graph.

```
> barplot(school.freq)
```

To add color, we create a vector of colors and then add that vector to the `barplot()` as a color palette

```
> colors = c("red", "yellow",
"green", "violet", "orange",
"blue", "pink", "cyan")
> barplot(school.freq, col=colors)
```



Category Statistics, `mean composition`

- Each school can be characterized by its various statistics, such as `means of: composition`, `drawing, coloring` and `expression`.
- Suppose we would like to know which school has the highest mean composition score.
- We would have to first find out the mean composition score of each school.
- Let us find the mean composition score of one school, e.g. school `C`. We do that in 3 steps:

1. Create a logical index vector for school C.
```
c_school = school == "C"   # the logical index vector
```
2. Find the subset of painters for school C.
```
c_painters = painters[c_school, ]  # child data set
```
3. Find the mean composition score of school C.
```
mean(c_painters$Composition)      # mean composition
[1] 13.16667                      # score of school C
```

@Zoran B. Djordjevic

9

`mean composition` score for all schools

- We could calculate mean composition score for all schools one by one or could use R function `tapply()`
```
> mean.scores = tapply(painters$Composition,painters$School,mean)
> mean.scores
       A        B        C        D        E        F        G        H
10.40000 12.16667 13.16667 9.10000 13.57143 7.25000 13.85714 14.0000
```
- Finally, we take an average of those values to get a mean over all schools.
```
> mean(mean.scores)
[1] 11.68899
```
- Function `tapply()` is used to apply a function, here `mean()`, to each group of components of the first argument, here `painters$Composition`, defined by the levels of the second component, here `painters$School`.
- Official Description of `tapply()` : "Applies a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors."

10

Quantitative Data

- Quantitative data, or continuous data, consists of numeric data that support arithmetic operations. This in contrast with categorical data, whose values belong to pre-defined classes with no arithmetic operation allowed.
- A built-in data frame `faithful` consists of a set of observations of the Old Faithful geyser in the USA Yellowstone National Park.

```
> head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
```

- There are two observation variables in the data set. The first one, called `eruptions`, is the duration of the geyser eruptions. The second one, called `waiting`, is the length of waiting period until the next eruption. We want to find out whether there is a correlation between the two variables.

Frequency Distribution of Quantitative Data

- The frequency distribution of a quantitative variable can be presented as a summary of occurrences of data in a collection of non-overlapping categories.
- This means that we will break the range of values over which a variable of interest varies into a set of intervals (usually of equal duration) and then count how many times values in our sample fall in each of those intervals.
- In what follows we will find the frequency distribution of the eruption durations in `faithful` data set. We do it in several steps:

1. We first find the range of eruption durations.
2. Break the range into non-overlapping intervals.
3. Classify the eruption durations according to which interval they fall into.
4. "Compute the frequency of eruptions in each interval" or count the number of eruption durations in each interval.

Frequency Distribution of Quantitative Data

- We first find the range of eruption durations with the range function.
- Observed eruptions are between 1.6 and 5.1 minutes in duration .

```
duration = faithful$eruptions;
range(duration)
[1] 1.6 5.1
```

- Break the range into non-overlapping intervals by defining a sequence of equal distance break points.
- We come up with the interval `[1.5, 5.5]`.
- We set the break points to be the half-integer sequence `{ 1.5, 2.0, 2.5,..}`

```
> breaks = seq(1.5, 5.5, by=0.5);   # half-integer sequence
  breaks
[1] 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5
```

- Next we classify the eruption durations according to the intervals.

13

Frequency Distribution of Quantitative Data

- We need to assign values from vector `duration` to the intervals delimited by sequence `breaks`. That is done by function `cut()`.

```
> duration.cut = cut(duration, breaks, right=FALSE)
```

- `cut()` accepts a vector that will be converted to a factor. In our case, `duration`.
- The second argument of `cut()` are labels for the factor levels of the resulting category. By default, labels are constructed as intervals of the form "`(a,b]`". Values of `a-s` and b-s are taken from the supplied vector containing labels, here `breaks`.
- As the intervals are to be closed on the left, and open on the right, what is reverse from the default, we set `right= FALSE`.
- Frequency of eruptions in each interval is calculated with `table()`.

```
> duration.freq = table(duration.cut);
  duration.freq
duration.cut
[1.5,2)  [2,2.5)  [2.5,3)  [3,3.5)  [3.5,4)  [4,4.5)  [4.5,5)  [5,5.5)
     51       41        5        7       30       73       61        4
```

14

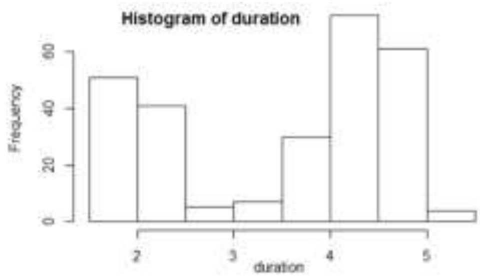
Histogram

- We use function `cbind()` to print the result in the column format.

```
> cbind(duration.freq)
duration.freq
[1.5,2) 51
[2,2.5) 41
[2.5,3) 5
[3,3.5) 7
[3.5,4) 30
[4,4.5) 73
[4.5,5) 61
[5,5.5) 4
```

It appears that `hist()` function is an efficient mechanism for finding and displaying frequency distributions.

```
duration = faithful$eruptions
hist(duration, # apply the hist()
right=FALSE)   # intervals closed
               # on the left
```



Relative Frequency Distribution Quantitative Data

- The relative frequency distribution of a data variable is the proportion of frequencies falling into a collection of non-overlapping categories (intervals)
- To find the relative frequency distribution of the eruption `duration`s, we first find the frequency distribution of the eruption durations

```
> duration.freq = table(duration.cut)
```

- Next we divide the frequency distribution with the sample size established by `nrow()`. `nrow()` tells us how many measurements are in the whole `faithful` sample.
- The relative frequency distribution is then calculated as

```
> duration.relfreq = duration.freq / nrow(faithful);
  duration.relfreq
 duration.cut
 [1.5,2)  [2,2.5)  [2.5,3)  [3,3.5)  [3.5,4)  [4,4.5) [4.5,5) [5,5.5)
0.187500 0.150735 0.018382 0.025735 0.110294 0.268382 0.22426 0.01470
```

Relative Frequency Distribution Quantitative Data

- We can print with fewer digits and make results more readable by setting the digits option.

```
> old = options(digits=3)
```

- We then apply the `cbind()` function to print both the frequency distribution and relative frequency distribution in parallel columns.

```
> old = options(digits=1) ;
  cbind(duration.freq, duration.relfreq);
        duration.freq duration.relfreq
[1.5,2)            51             0.19
[2,2.5)            41             0.15
[2.5,3)             5             0.02
[3,3.5)             7             0.03
[3.5,4)            30             0.11
[4,4.5)            73             0.27
[4.5,5)            61             0.22
[5,5.5)             4             0.01
> options(old)    # restore the old option
```


Scatter Plot of Old Faithful Data

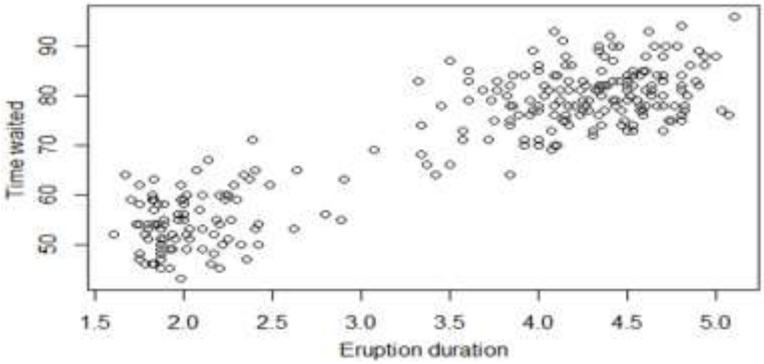
- A scatter plot pairs up values of two quantitative variables in a data set and display them as geometric points on a diagram.
- We pair up the `eruptions` and `waiting` values in the same observation as `(x,y)` coordinates.
- We plot the points in the Cartesian plane.
- The eruption data value pairs with help of function `cbind()`

```
duration = faithful$eruptions;     # the eruption
  waiting = faithful$waiting;         # the waiting interval
  head(cbind(duration, waiting));
      duration waiting
 [1,]    3.600      79
 [2,]    1.800      54
 [3,]    3.333      74
 [4,]    2.283      62
 [5,]    4.533      85
 [6,]    2.883      55
```

Scatter Plot of Old Faithful Data

- We apply the `plot` function to compute the scatter plot of eruptions and waiting.

```
> plot(duration, waiting, xlab="Eruption duration",
   ylab="Time waited")
```



We do see “correlation” between variables. If you increase one, on average the other also increases

Scatter Plot of Old Faithful Data

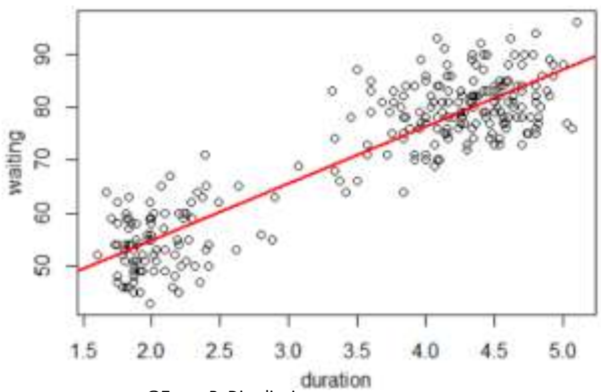
- To establish the “best possible” linear relationship between two variable, we generate the so called linear model, or linear regression, using function `lm()` :

```
> model = lm(waiting ~ duration, data = faithful)
```

- and draw the trend line with function `abline()`:

```
> abline(model, col='red', lwd = 2)
```

- Parameter `lwd` determines the line width.
- We will discuss function `lm()` at length, later.



Cumulative Relative Freq. Distribution

- The cumulative relative frequency distribution of a quantitative variable is a summary of frequency proportions below a given level. Formally, cumulative relative frequency distribution is the integral of the relative frequency distribution from the beginning of the range to the observation point (interval, level).

$$Cumulative\ Rel.Freq.Distribution\ (\boldsymbol{l}) = \int_0^{l} Rel.Freq.Distribution(\boldsymbol{i})\ d\boldsymbol{i}$$

- Find the frequency distribution of the eruption durations as follows:

```
> duration = faithful$eruptions ;
  breaks = seq(1.5, 5.5, by=0.5);
  duration.cut = cut(duration, breaks, right=FALSE);
  duration.freq = table(duration.cut)
```

- We then apply `cumsum()` function to compute the cumulative frequency distribution.

```
> duration.cumfreq = cumsum(duration.freq)
```

- The sample size of faithful is found with `nrow()`, and the cumulative relative frequency distribution is given by:

```
> duration.cumrelfreq = duration.cumfreq / nrow(faithful)
```


Cumulative Relative Freq. Distribution

- We then apply the `cbind()` function to print both the cumulative frequency distribution and relative cumulative frequency distribution in parallel columns.

```
> old = options(digits=2);
  cbind(duration.cumfreq, duration.cumrelfreq);
        duration.cumfreq duration.cumrelfreq
[1.5,2)               51                0.19
[2,2.5)               92                0.34
[2.5,3)               97                0.36
[3,3.5)              104                0.38
[3.5,4)              134                0.49
[4,4.5)              207                0.76
[4.5,5)              268                0.99
[5,5.5)              272                1.00
> options(old)
```

Cumulative Relative Frequency Distribution

- We could plot the cumulative relative frequency of durations of eruptions starting with zero element.

```
> cumrelfreq0 = c(0, duration.cumrelfreq);
  plot(breaks, cumrelfreq0,
main="Old Faithful Eruptions",      # main title
xlab="Duration minutes",
ylab="Cumulative eruption distribution");
  lines(breaks, cumrelfreq0);         # join the points
```

- Please note that cumulative distributions always range from 0 to 1

Probability Distributions

Probability Distribution

- The Histogram of the durations of Old Faithful Eruptions and the subsequent Cumulative Relative Frequency Distribution are telling us how particular events are distributed along a particular parameter axis or space.
- Such distributions are of great interest in probability and statistics and are usually studied under the term of **Probability Distributions**.
- For example, the collection of all possible outcomes of a sequence of coin tossing will turn out to be a distribution, known as the binomial distribution.
- The means of sufficiently large samples of a data population are known to resemble the normal distribution.
- The characteristics of these and other theoretical distributions are well understood. They can be used to make statistical inferences on data populations which they represent well.

Binomial Distribution

- The **binomial distribution** is a discrete probability distribution. It describes the outcome of *n* independent trials in an experiment. Each trial is assumed to have only two outcomes, either success or failure. If the probability of a successful trial is *p*, then the probability of having *k* successful outcomes in an experiment of *n*-independent trials is equal to .

$$f(k) = \binom{n}{k} p^k (1 - p)^{(n-k)}, \text{ where } k = 0, 1, 2, \ldots, n$$

Factor $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is referred to as the binomial coefficient

Practical Problem

- Suppose there are twelve multiple choice questions (trials) in an English class quiz. Each question has five possible answers, and only one of them is correct. Find the probability of having four or less correct answers if a student attempts to answer every question at random.

Binomial Problem, Solution

- If only one out of five possible answers is correct, the probability of answering a question correctly by random is $p=1/5=0.2$.
- Probability for answering incorrectly is $1\text{-}p = 4/5 = 0.8$.
- From $f(x)$ we can find the probability of having exactly k=4 correct answers in 12 random attempts :

$f(4,12) = \frac{12!}{4!8!}0.2^4 0.8^8$. We could also use R function $choose(n,k)$, i.e. $choose(12{,}4)0.2^4 0.8^{12}$ or use R function `dbinom(k, size, prob)` `dbinom(4,size=12,prob=0.2);` all giving the result `0.1328756`

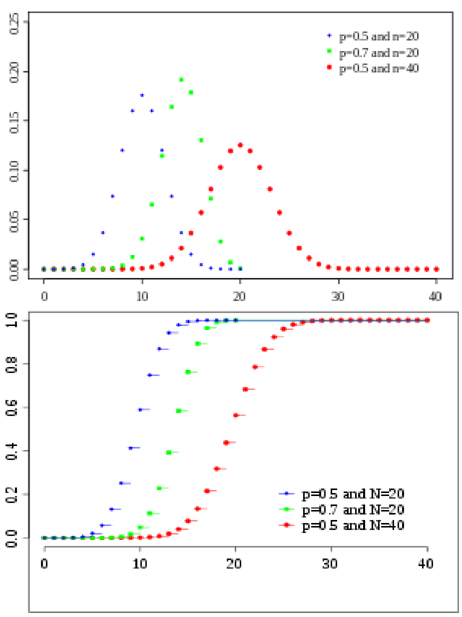
To find the probability of having four or less correct answers in 12 random attempts, we apply function dbinom() with k = 0,…,4.

```
> dbinom(0, size=12, prob=0.2) +
+ dbinom(1, size=12, prob=0.2) +
+ dbinom(2, size=12, prob=0.2) +
+ dbinom(3, size=12, prob=0.2) +
+ dbinom(4, size=12, prob=0.2);
[1] 0.9274
```


Mass Function vs. Cumulative Distribution

- The probability distribution is some times referred to as the probability mass function.
- On the right we see variation of the binomial distribution with k (number of successes) out of n = 20 and n = 40 trials.
- The bottom diagram presents cumulative binomial distribution, i.e. the probability that there were k or less successes in n = 20 and 40 trials.
- The cumulative binomial distribution is calculated as

```
> pbinom(4,12,0.2);
[1] 0.9274445
```

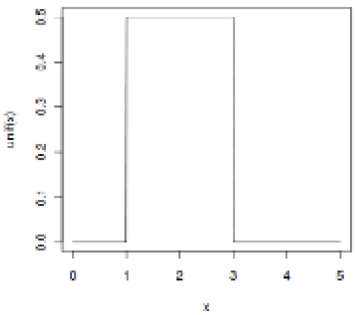


Continuous Uniform Distribution

- The **continuous uniform distribution** is the probability distribution of random number selection from the continuous interval between *a* and *b*. Its density function is defined by:

$$f(x) = \begin{cases} \frac{1}{b-a} & when\ a \le x \le b \\ 0 & when\ x < a\ or\ x > b \end{cases}$$

- Below is a graph of continuous uniform distribution with *a*= 1,*b*= 3.



- A set of numbers uniformly distributed between 1 and 3 could be generated with a call to R function

```
> runif(10,min=1, max=3)
[1] 2.032381 1.792425 1.805124 1.733175
[5] 1.642199 1.830730 1.183520 1.251148
[9] 2.372529 2.625160
```


Statistical Measures

Statistical Measures

- If we know the probability (statistical) distribution of a process, i.e. a random variable, we could describe results of measurements involving that process (variable) most accurately.
- There are situations when we cannot rely on distribution functions:
  - We do not posses the full knowledge of the behavior of a random variable.
    - We posses no extensive data set illustrating the behavior nor
    - We have a simple (or complex) formula for the probability distribution
  - We need to transmit information about a process using a few numbers rather than an extended data set or a formula.
- There exists a set of standard descriptions or measures of statistical and probability distributions

Statistical Measures

- Mean
- Median
- Quartile
- Percentile
- Range
- Interquartile Range
- Box Plot
- Variance
- Standard Deviation
- Covariance
- Correlation Coefficient
- Central Moment
- Skewness
- Kurtosis

Mean

- The **mean** of an observation variable is a numerical measure of the central location of data values. It is the sum of its data values divided by data count. It corresponds to the center of gravity.
- Hence, for a data sample of size *n,* its **sample mean** is defined as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Problem
- Find the mean eruption duration in the `faithful` data set.

Solution
- We apply the mean function to compute the values of `eruptions`.

```
➢ duration = faithful$eruptions # the eruption durations
➢ mean(duration) # apply the mean function
  [1] 3.4878
```


Median

- The **median** of an observation variable is the value at the middle when the data is sorted in ascending order. It is an ordinal measure of the central location of the data values.
- If you have 11 measurements and you order them from the lowest to the highest, the median is the 6^{th} measurement in the ordered set

Problem
- Find the median of the eruption duration in the data set faithful.

Solution
- We apply the median function to compute the median value of eruptions.

```
> duration = faithful$eruptions; # eruption durations
  median(duration); # apply the median function
[1] 4
```

Quartile

- There are several **quartiles** of an observation variable. The **first quartile**, or **lower quartile**, is the value that cuts off the first 25% of the data when data is sorted in an ascending order. The **second quartile**, or **median**, is the value that cuts off the first 50%. The **third quartile**, or **upper quartile**, is the value that cuts off the first 75%.

Problem

- Find the quartiles of the eruption durations in the `faithul` data set.

Solution

- We apply the quantile function to compute the quartiles of eruptions.

```
> duration = faithful$eruptions; # the eruption durations
  quantile(duration) # apply the quantile function
0% 25% 50% 75% 100%
1.6000 2.1627 4.0000 4.4543 5.1000
```

Answer

- The first, second and third quartiles of the eruption duration are 2.1627, 4.0000 and 4.4543 minutes respectively.

Percentile

- The n^{th} **percentile** of an observation variable is the value that cuts off the first *n*-percent of the data values when the data set is sorted in ascending order.

Problem

- Find the 32^{nd}, 57^{th} and 98^{th} percentiles of the eruption durations in the data set faithful.

Solution

- We apply the `quantile` function to compute the percentiles of eruptions with the desired percentage ratios.

```
> duration = faithful$eruptions # eruption durations
> quantile(duration, c(.32, .57, .98));
32% 57% 98%
2.3952 4.1330 4.9330
```

Answer

- The 32^{nd}, 57^{th} and 98^{th} percentiles of the eruption duration are 2.3952, 4.1330 and 4.9330 minutes respectively.

Range

- The **range** of an observation variable is the difference of its largest and smallest data values. It is a measure of how far apart the entire data spreads in value.

Problem

- Find the range of the eruption durations in the `faithful` data set.

Solution

- We apply the max and min function to compute the largest and smallest values of eruptions, then take the difference.

```
> duration = faithful$eruptions # eruption durations
> max(duration) − min(duration) # apply the max and min
                                # functions
[1] 3.5
```

Answer

- The range of the eruption duration is 3.5 minutes.

Interquartile Range

- The **interquartile range** of an observation variable is the difference of its upper and lower quartiles. It is a measure of how far apart the middle portion of data spreads in value.

Problem

- Find the interquartile range of eruption duration in the data set faithful.

Solution

- We apply the IQR function to compute the interquartile range of eruptions.

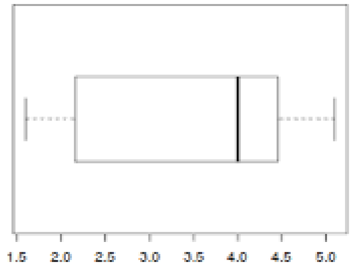
```
> duration = faithful$eruptions; # eruption durations
  IQR(duration)                  # apply the IQR function
[1] 2.2915
```

Answer

- The interquartile range of eruption duration is 2.2915 minutes.

Box Plot

- The **box plot** of an observation variable is a graphical representation based on its quartiles, as well as its smallest and largest values. It attempts to provide a visual shape of the data distribution.



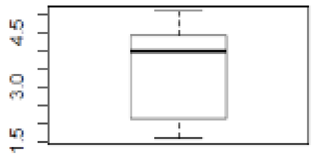
Problem

- Find the box plot of the eruption duration in the data set `faithful`.

Solution

- We apply the `boxplot()` function to produce the box plot of `eruptions`.

```
> duration = faithful$eruptions
# eruption durations
 boxplot(duration, horizontal=FALSE)
# vertical box plot
```



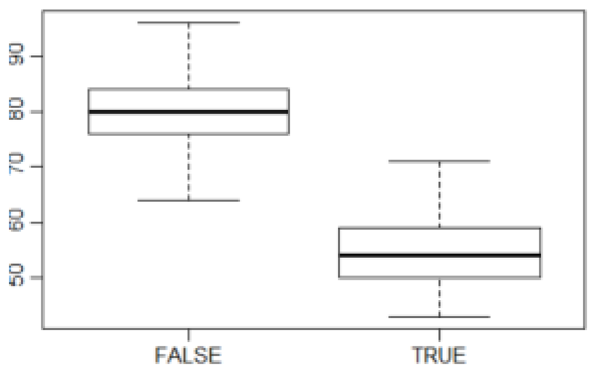
Old Faithful Measures

- Let us add a new column to the faithful dataset

```
> faithful$type <- duration < 3.1;
  type <- faithful$type
```

- Present waiting times measures for two different types:

```
➢ boxplot(waiting ~ type, data = faithful)
```



Variance

- The **variance** is a numerical measure of how the data values are dispersed around the mean. In particular, the **sample variance** is defined as:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Similarly, the **population variance** is defined in terms of the population mean μ and population size N as:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

Problem

- Find the variance of the eruption duration in `faithful` data set.

Solution

- We apply the `var()` function to compute the variance of eruptions.

```
> duration = faithful$eruptions;
  var(duration) # apply the var function
[1] 1.3027
```


Function $var()$

- Function $var(x)$, for variance, acts on sample vector x and calculates:

  $var(x) = sum((x - mean(x))\text{^}2)/(length(x) - 1)$
- or sample variance.
- If the argument to $var()$ is an n by p matrix the value is a p by p sample *Covariance Matrix* obtained by regarding the rows as independent p-variate (p-dimensional) sample vectors.

In the Case you Noticed

- One of the greatest mysteries of Statistics as a science is the factor `(n-1)` in the formula for sample variance.
- The POPULATION VARIANCE $\boldsymbol{\sigma^2}$ is a PARAMETER of the population.
- The SAMPLE VARIANCE $\boldsymbol{s^2}$ is a STATISTIC of the sample.
- We use the sample statistic to estimate the population parameter
- The sample variance $\boldsymbol{s^2}$ is an estimate of the population variance $\boldsymbol{\sigma^2}$
- Suppose we have a population with N individuals or items.
- Suppose that we want to take samples of size `n` from that population.

43

`(n-1)` Denominator Mystery

- If we could list all possible samples of `n` items that could be selected from the population of N items, then we could find the sample variance for each possible sample.
- We would want the following to be true:
  - The average of the sample variances for all possible samples to equal the population variance.
- This is a logical proposition and a reasonable thing to expect.
- If the sample variance satisfies this requirement, we say it is "unbiased"
- When we divide by `(n -1)` in calculating the sample variance, it turns out that the average of the sample variances for all possible samples is equal the population variance.
- Such sample variance is an unbiased estimate of the population variance. Should in formula for s^2 on slide 39, we divide by n, the above will not be true.
- This assertion could be proven in a rigorous way.

44

Standard Deviation

- The **standard deviation** of an observation variable is the square root of its variance.

Problem

- Find the standard deviation of the eruption duration in the `faithful` data set

Solution

- We apply the `sd()` function to compute the standard deviation of eruptions.

```
> duration = faithful$eruptions; # eruption durations
  sd(duration) # apply the sd function
[1] 1.1414
```


Covariance

- The **covariance** of two variables x and y in a data sample measures how or whether two variables are (linearly) related.
- A positive covariance indicates a positive linear relationship between the variables, and a negative covariance indicates the opposing relationship.
- The **sample covariance** is defined in terms of the sample means as:

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

- Similarly, the **population covariance** is defined in terms of the population means μ_x, μ_y as:

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

Problem

- Find the covariance of the eruption duration and waiting time in the data set faithful. Observe if there is any linear relationship between the two variables.

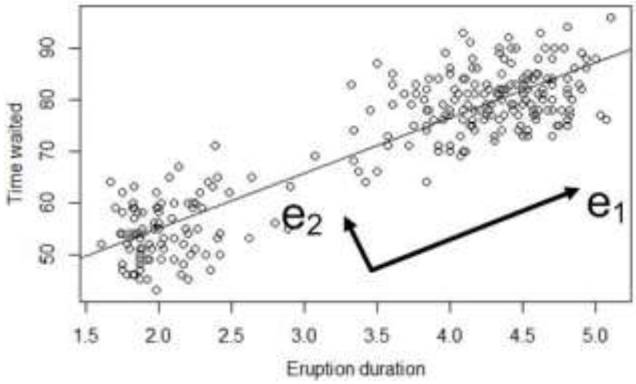
Solution

- We apply the `cov()` function to compute the covariance of eruptions and waiting.

```
> duration = faithful$eruptions; # the eruption durations
  waiting = faithful$waiting;    # the waiting period
  cov(duration, waiting)         # apply the cov function
[1] 13.978
```

Eigen Vectors of Covariance Matrix

- Eigen vectors of covariance matrix or its normalized form provide important insight in the behavior of our data.
- The largest eigen vectors of that matrix point into directions of strongest variation of underlying variables.



Correlation Coefficient

- The **correlation coefficient** of two variables in a data sample is their covariance divided by the product of their individual standard deviations . It is a normalized measurement of how the two are (linearly) related.
- Formally, the **sample correlation coefficient** is defined by the following formula, where s_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

- Similarly, the **population correlation coefficient** is defined as:

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

- Where σ_x and σ_y are the population standard deviations, and σ_{xy} is the population covariance.
- If the correlation coefficient is close to 1, it indicates that the variables are positively linearly related and the scatter plot falls almost along a straight line with a positive slope.
- Correlation coefficient of -1 indicates that the variables are negatively linearly related and the scatter plot almost falls along a straight line with negative slope.
- Correlation coefficient of 0 (zero) indicates a very weak linear relationship between the variables, or absence of a relationship between variables.

Correlation Coefficient

Problem

- Find the correlation coefficient of the eruption duration and waiting time in the `faithful` data set. Observe if there is any linear relationship between two variables.

Solution

- We apply the `cor()` function to compute the correlation coefficient of `eruptions` and `waiting`.

```
> duration = faithful$eruptions; # eruption durations
  waiting = faithful$waiting;    # waiting period
  cor(duration, waiting);       # apply the cor function
[1] 0.90081
```

- The correlation coefficient of the eruption duration and waiting time is 0.90081.
- The correlation coefficient is close to 1, and we can conclude that eruption duration and the waiting time are positively linearly correlated.

Central Moment

- The k^{th} **central moment** (or moment about the mean) of a data population is:

$$\mu_k = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^k$$

- Similarly, the k^{th} central moment of a data sample is:

$$m_k = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^k$$

- The second central moment of a sample population is its variance.

Problem

- Find the third central moment of eruption duration in the `faithful` data set

Solution

- We apply the function moment from the e1071 package. Package e1071 is not in the core R library, and has to be installed and loaded into the R workspace.

```
> library(e1071); # load e1071
  duration = faithful$eruptions; # eruption durations
  moment(duration, order=3, center=TRUE);
[1] -0.6149    #  The third central moment of eruption
                # duration is -0.6149.
```

Note on Loading Packages

- If you need a package, you usually type:

```
> install.packages("package_name")
> library(package_name)
```

- And you are done. R goes to `cran.r-project.org`, finds the package installs it for you and you are done.
- That did not work for me for the package e1071.
- I still had to go to `cran.r-project.org`, find the link to Packages and look for the packages you need, e.g. e1071.
- You will have the option to download a ZIP or a tar file.
- Expand that archive and drop resulting directory in the subdirectory library of the installation directory of your R. In my case that directory is
- `C:\Program Files\R\R-3.0.2\library`

Skewness

- The **skewness** of a data population is defined by a specific ratio of $\mu_2$and μ_3 are the second and third central moments.

$$\gamma_1 = {}^{\mu_3}\!/_{\mu_2{}^{3/2}}$$

- Intuitively, the skewness is a measure of symmetry.
- As a rule, negative skewness indicates that the mean of the data values is less than the median, and the data distribution is *left-skewed*. Positive skewness would indicates that the mean of the data values is larger than the median, and the data distribution is *right-skewed*.

Problem

- Find the skewness of eruption duration in the data set faithful.
- You will do it for your homework

Kurtosis

- The **kurtosis** of a univariate population is defined by the following formula, where μ_2 and μ_4 are the second and fourth central moments

$$\gamma_2 = {}^{\mu_4}/_{{\mu_2}^2}$$

- Intuitively, the *kurtosis* is a measure of the "*peakedness"* of the data distribution. Negative kurtosis would indicates a *flat* data distribution, which is said to be **platykurtic**.
- Positive kurtosis would indicates a *peaked* distribution, which is said to be **leptokurtic**. Incidentally, the normal distribution has zero kurtosis, and is said to be **mesokurtic**.

Problem

- Find the kurtosis of eruption duration in the data set faithful

Solution

- You will do it for your homework

Normal and Other Distributions

Normal Distribution

- The **normal distribution** is defined by the following probability density function, where μ is the population mean and σ^2 is the variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

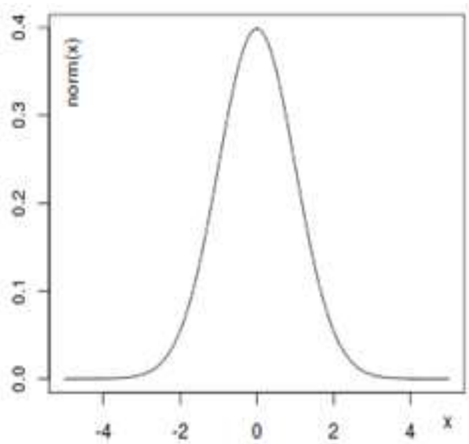
- If a random variable *X* follows the normal distribution, then we write:

$$X \sim N(\mu, \sigma^2)$$

- In particular, the normal distribution with $\mu = 0$ and $\sigma = 1$ is called the ***standard normal distribution***, and is denoted as *N*(0,1).
- The graph on the next pages shows a standard normal distribution. The "normal" normal distribution looks very much the same. It is just shifted to point $\boldsymbol{x} = \boldsymbol{\mu}$ and expanded $\boldsymbol{\sigma}$ times.

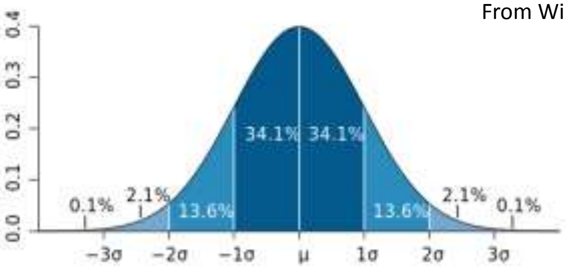
Graph, Gaussian Distribution, Central Limit Theorem

- The normal distribution is also called Gaussian distribution.
- The normal distribution is used in many situations because of the fact that when you combine a large number of random variables, each with a fairly arbitrary distribution, to produce a new average variable (value), that resulting variable has a normal distribution.
- That fact is described by the **Central Limit Theorem**



Standard Deviation σ

From Wikipedia



±1*σ	68.2 %
±2*σ	95.4 %
±3*σ	99.6 %

- Standard deviation provides some convenience reference points on the distribution curve. For example, more that 2/3 of all samples fall in the interval of width 2*σ around the mean.
- Similarly, 95.4% of all samples fall in the interval of width 4*σ around the mean.
- Now you know where the term 6σ is coming from.

Previous Plot in R, from Wikipedia

```
svg(filename = "Standard deviation diagram.svg", width = 7, height = 3.5)
par(mar = c(2, 2, 0, 0))
# # External package to generate four shades of blue
library(RColorBrewer)
# cols <- rev(brewer.pal(4, "Blues"))
cols <- c("#2171B5", "#6BAED6", "#BDD7E7", "#EFF3FF")  # Sequence between -4 and 4 with 0.1 steps
x <- seq(-4, 4, 0.1)    # Plot an empty chart with tight axis boundaries, and axis lines on bottom and left
plot(x, type="n", xaxs="i", yaxs="i", xlim=c(-4, 4), ylim=c(0, 0.4),
    bty="l", xaxt="n", xlab="", ylab="")
# Function to plot each coloured portion of the curve, between "a" and "b" as a
# polygon; the function "dnorm" is the normal probability density function
polysection <- function(a, b, col, n=11){
   dx <- seq(a, b, length.out=n)
   polygon(c(a, dx, b), c(0, dnorm(dx), 0), col=col, border=NA)     # draw a white vertical line on "inside" side to separate each section
   segments(a, 0, a, dnorm(a), col="white")
}  # Build the four left and right portions of this bell curve
for(i in 0:3){
   polysection(   i, i+1,  col=cols[i+1]) # Right side of 0
   polysection(-i-1,  -i,  col=cols[i+1]) # Left right of 0
} # Black outline of bell curve
lines(x, dnorm(x))   # Bottom axis values, where sigma represents standard deviation and mu is the mean
axis(1, at=-3:3, labels=expression(-3*sigma, -2*sigma, -1*sigma, mu,
                          1*sigma,  2*sigma,  3*sigma))
# Add percent densities to each division (rounded to 1 decimal place), between x and x+1
pd <- sprintf("%.1f%%", 100*(pnorm(1:4) - pnorm(0:3)))
text(c((0:3)+0.5,(0:-3)-0.5), c(0.16, 0.05, 0.04, 0.02), pd, col=c("white","white","black","black"))
segments(c(-2.5, -3.5, 2.5, 3.5), dnorm(c(2.5, 3.5)), c(-2.5, -3.5, 2.5, 3.5), c(0.03, 0.01))
dev.off()
```

Central Limit Theorem formally, Arbitrary Distributions

- Let $X_1, X_2, \ldots, X_N$ be a set of N independent random variables and each X_i has an arbitrary probability distribution *$P(X_i)$* with mean μ_i and a finite variance $\sigma_i{}^2$ Then, the variable

$$X_{Norm} = \frac{1}{N} \left(\sum_{i=1}^{N} X_i - \sum_{i=1}^{N} \mu_i\right)$$

- i.e., the variation of the sum of variables $\boldsymbol{X_i}$ from the sum of their means, in the case when N (the number of random variables) is large, approaches a distribution function which is normally distributed. The mean of the resulting distribution is $\boldsymbol{\mu} = \boldsymbol{0}$ and the variance is equal to $\boldsymbol{\sigma_X} = {}^1/_N \sqrt{\sum_{i=1}^{N} \sigma_i^2}$
- Note that variances add as if random processes are vectors in an N-dimensional vector space.

CLT, Collection of Identical Distributions

- If all variables $\{X_i\}$ have the same probability distribution with identical variance σ_x, and mean μ_x then the average variable

$$X_{Norm} = \frac{1}{N} \left(\sum_{i=1}^{N} X_i\right)$$

- is normally distributed with $\mu_X = \mu_x$ and variance $\sigma_X = \sigma_x / \sqrt{N}$

[1] The **mean** of the population of random variable is always equal to the mean of the parent population from which the population samples were drawn.

[2] The **standard deviation** of the population of means is always equal to the standard deviation of the parent population divided by the square root of the sample size (N).

[3] Most importantly, **the distribution of means** will increasingly approximate a **normal distribution** as the size N of samples increases.

- The Central Limit Theorem explains the ubiquity of the famous bell-shaped "Normal distribution" (or "Gaussian distribution") in the all kinds of measurements.

Application of Normal Distribution

Problem

- Assume that the test scores of a college entrance exam fits a normal distribution. Furthermore, the mean test score is 72, and the standard deviation is 15.2. What is the percentage of students scoring 84 or more in the exam?

Solution

- We apply the function `pnorm()` of the normal distribution with mean 72 and standard deviation 15.2. Since we are looking for the percentage of students scoring higher than 84, we are interested in the *upper tail* of the normal distribution.

```
> pnorm(84, mean=72, sd=15.2, lower.tail=FALSE)
[1] 0.21492
```

- The percentage of students scoring 84 or more in the college entrance exam is 21.5%.

> ?pnorm

- Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd. Usage:

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

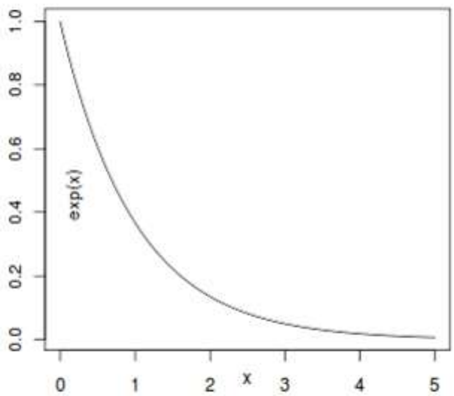
`x,q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	`if TRUE,` probabilities `p` given as `log(p)`.
`lower.tail`	`if TRUE` (default), probabilities are `P[X ≤ x] otherwise, P[X > x]`.

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

Exponential Distribution

- The **exponential distribution** describes the arrival time of a randomly recurring independent event sequence. If μ is the mean waiting time for the next event recurrence, its probability density function is give by:

$$f(x) = \begin{cases} \frac{1}{\mu} e^{-x/\mu} & when\ x \geq 0 \\ 0 & when\ x < 0 \end{cases}$$



Graph to the right corresponds to the exponential distribution with μ = 1.

Exponential Distribution

Problem

- Suppose the mean checkout time of a supermarket cashier is three minutes. Find the probability of a customer checkout being completed by the cashier in less than two minutes.

Solution

- The checkout processing rate is equals to one divided by the mean checkout completion time. Hence the processing rate is 1/3 checkouts per minute. We then apply the function `pexp()` of the exponential distribution with rate=1/3.

```
> pexp(2, rate=1/3)
[1] 0.48658
```

Answer

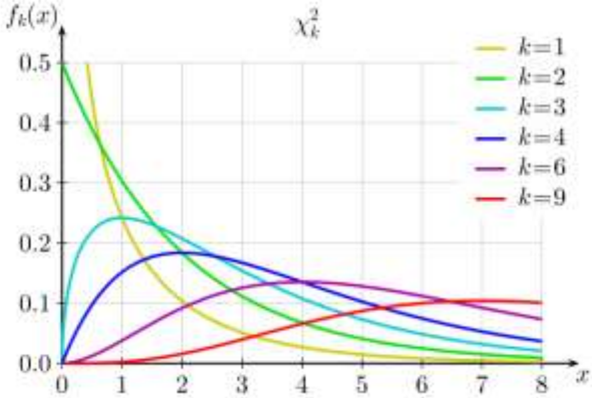
- The probability of finishing a checkout in under two minutes by the cashier is 48.7%

Chi-squared Distribution

- If $X_1, X_2, \ldots, X_m$ be a set of ***m*** independent random variables each having the standard normal distribution, then variable V, defined as the sum of squares of $\{X_i\}$ -s

$$V = X^2{}_1 + X_2{}^2 + \ldots + X^2{}_m$$

Follows a Chi-Squared Distribution with m-degrees of freedom.



The mean value of variable V is ***m*** and its variance is equal to ***2m***

Chi-Square Distribution

Problem

- Find the 95th percentile of the Chi-Squared distribution with 7 degrees of freedom.

Solution

- We apply the quantile function `qchisq()` of the Chi-Squared distribution against the decimal values 0.95.

```
> qchisq(.95, df=7) # 7 degrees of freedom
[1] 14.067
```

Answer

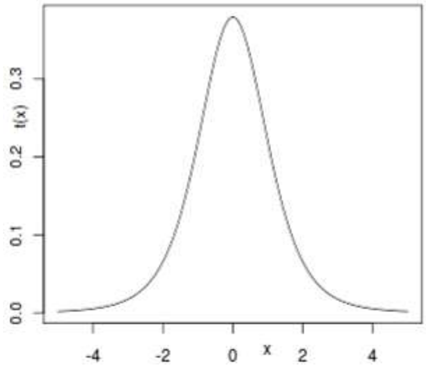
- The 95th percentile of the Chi-Squared distribution with 7 degrees of freedom is 14.067.

Student t Distribution

- Assume that a random variable *Z* has the standard normal distribution, and another random variable *V* has the Chi-Squared distribution with *m-degrees of freedom*.
- Assume further that *Z* and *V* are independent, then variable t defined as:

$$t = \frac{Z}{\sqrt{V/m}} \sim t(m)$$

- follows a **Student t distribution** with *m* degrees of freedom.



Student *t* Distribution

Problem

- Find the 2.5*th* and 97.5*th* percentiles of the Student t distribution with 5 degrees of freedom.

Solution

- We apply the quantile function `qt()` of the Student t distribution against the decimal values 0.025 and 0.975.

```
> qt(c(.025, .975), df=5) # 5 degrees of freedom
[1] -2.5706 2.5706
```

Answer

- The 2.5*th* and 97.5*th* percentiles of the Student t distribution with 5 degrees of freedom are -2.5706 and 2.5706 respectively.

Interval Estimation

- It is a common requirement to efficiently estimate population parameters based on simple random sample data. We will demonstrate how to compute the estimates. We will illustrate steps with a built-in data frame named `survey`. It is the outcome of a survey of Statistics students in an Australian university.
- The data set belongs to the MASS package, which has to be pre-loaded into the R workspace prior to use.

```
> library(MASS) # load the MASS package
```

```
> head(survey)
     Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height
1 Female   18.5   18.0 Right  R on L    92    Left Some Never 173.00
2   Male   19.5   20.5  Left  R on L   104    Left None Regul 177.80
3   Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA
4   Male   18.8   18.9 Right  R on L    NA Neither None Never 160.00
5   Male   20.0   20.0 Right Neither    35   Right Some Never 165.00
6 Female   18.0   17.7 Right  L on R    64   Right Some Never 172.72
```


Point Estimate of Population Mean

- For any particular random sample, we can always compute its sample mean. Although most often it is not the actual population mean, it does serve as a good **point estimate**. For example, in the data set survey, the survey is performed on a sample of the student population. We can compute sample mean and use it as an estimate of the corresponding population parameter.

Problem

- Find a point estimate of mean university student height with the sample `survey`.

Solution

- We save the survey data of student heights in a variable `height.survey`.

```
> library(MASS)      # load the MASS package
> height.survey = survey$Height
```

- It turns out not all students have answered all question, and we must filter out the missing values. We apply the `mean()` function with the `"na.rm"` argument as TRUE.

```
> mean(height.survey, na.rm=TRUE) # skip missing values
[1] 172.38
```

Answer

- A point estimate of the mean student height is 172.38 centimeters.

Interval Estimate of Population Mean with Known Variance

- After we found a **point estimate of the population mean**, we would need a way to quantify its accuracy. Here, we assume that the population variance σ^2 is known.
- Let us denote the $100(1 - \alpha/2)$ percentile of the standard normal distribution as $z_{\alpha/2}$. For random sample of sufficiently large size, the end points of the **interval estimate** at $(1 - \alpha)$ confidence level is given as:

$$\bar{x} \pm z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Problem

- Assume the population standard deviation **σ** of the student height in the survey is 9.48. Find the margin of error and interval estimate at 95% confidence level.

Interval Estimate of Population Mean, Known Variance

Solution

- We first filter out missing values in survey$Height with the na.omit function, and save it in height.response.

```
> library(MASS) # load the MASS package
> height.response = na.omit(survey$Height)
```

- Then we compute the standard error of the mean.

```
> n = length(height.response)
> sigma = 9.48 # population standard deviation
> sem = sigma/sqrt(n); sem # standard error of the mean
[1] 0.65575
```

- Since there are two tails of the normal distribution, the 95% confidence level would imply the 97.5^{th} percentile of the normal distribution at the upper tail. Therefore, $z_{\alpha/2}$ is given by qnorm(.975). We multiply it with the standard error of the mean `sem` and get the margin of error.

```
>E = qnorm(.975)*sem; E # margin of error
[1] 1.2852
```

- We then add it up with the sample mean, and find the confidence interval.

```
> xbar = mean(height.response) # sample mean
> xbar + c(-E, E)
[1] 171.10 173.67
```

Answer

- Assuming the population standard deviation σ being `9.48`, the margin of error for the student height survey at `95%` confidence level is `1.2852` centimeters. The confidence interval is between `171.10` and `173.67` centimeters.

Interval Estimate of Population Mean with Unknown Variance

- After we found a point estimate of the population mean, we would need a way to quantify its accuracy. Now, let us assume that the population variance is not known.
- Let us denote the $100(1-\alpha/2)$ percentile of the Student t distribution with $n-1$ degrees of freedom as $t_{\alpha/2}$.
- For random samples of sufficiently large size, and with standard deviation *s*, the end points of the **interval estimate** at $(1-\alpha)$ confidence level is given as :

$$\bar{x} \pm t_{\alpha/2} \frac{s}{\sqrt{n}}$$

Problem

- Without assuming the population standard deviation of the student height in survey, find the margin of error and interval estimate at 95% confidence level.

Interval Estimate, Unknown Variance

Solution

- We first filter out missing values in `survey$Height` with the `na.omit` function, and save it in `height.response`.

```
> library(MASS) # load the MASS package
> height.response = na.omit(survey$Height)
```

- Then we compute the sample standard deviation.

```
> n = length(height.response)
> s = sd(height.response) # sample standard deviation
> SE = s/sqrt(n); SE # standard error estimate
[1] 0.68117
```

- Since there are two tails of the Student t distribution, the 95% confidence level would imply the 97.5^{th} percentile of the Student t distribution at the upper tail. Therefore, $t_{\alpha/2}$ is given by `qt(.975, df=n-1)`. We multiply it with the standard error estimate SE and get the margin of error.

```
> E = qt(.975, df=n−1)*SE; E # margin of error
[1] 1.3429
```

- We then add it up with the sample mean, and find the confidence interval.

```
> xbar = mean(height.response) # sample mean
> xbar + c(−E, E)
[1] 171.04 173.72
```

Answer

- Without assumption on the population standard deviation, the margin of error for the student height survey at 95% confidence level is 1.3429 centimeters. The confidence interval is between 171.04 and 173.72 centimeters.