Business Statistics

J. Alejandro Gelves

2024-09-06

Table of contents

Introduction

"Whatever you would make habitual, practice it; and if you would not make a thing habitual, do not practice it, but accustom yourself to something else." *Epictetus*

This course companion is designed to help you build mastery in statistics and its applications using R. Through practice, you will develop the skills and confidence needed to apply statistical concepts effectively. Each chapter begins with a list of key concepts to guide your learning, and the problems are crafted to reinforce these ideas through hands-on experience. If you need additional support while learning R, I encourage you to explore Grolemund (2014). Take your time, enjoy the process, and make practice a habit!

Why R?

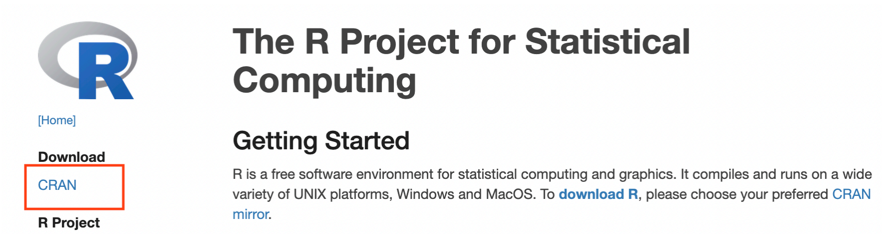
We will be using R to apply the lessons we learn in BUAD 231. R is a language and environment for statistical computing and graphics. There are several advantages to using the R software for statistical analysis and data science. Some of the main benefits include:

- R is a **powerful and flexible programming language** that allows users to manipulate and analyze data in many different ways.
- R has a large and **active community of users**, who have developed a wide range of packages and tools for data analysis and visualization.
- R is **free and open-source**, which makes it accessible to anyone who wants to use it.
- R is **widely used** in academia and industry, which means that there are many resources and tutorials available to help users learn how to use it.
- R is well-suited for working with **large and complex datasets**, and it can handle data from many different sources.
- R can be **easily integrated** with other tools and software, such as databases, visualization tools, and machine learning algorithms.

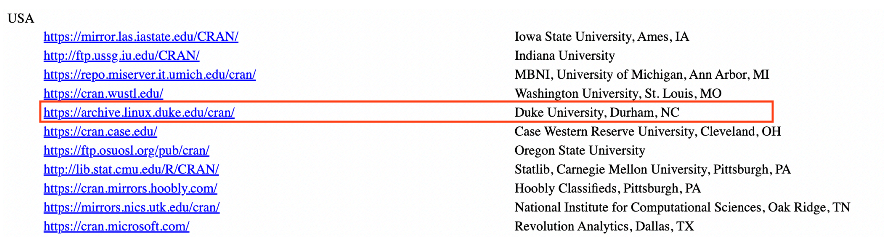
Overall, R is a powerful and versatile tool for data analysis and data science, and it offers many benefits to users who want to work with data.

Installing R.

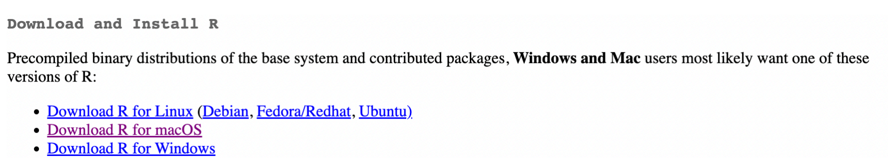
To install R, visit the R webpage at https://www.r-project.org/. Once in the website, click on the CRAN hyperlink.



Here you can select the CRAN mirror. Scroll down until you see USA. You are free to choose any mirror you like, I recommend using the Duke University mirror.



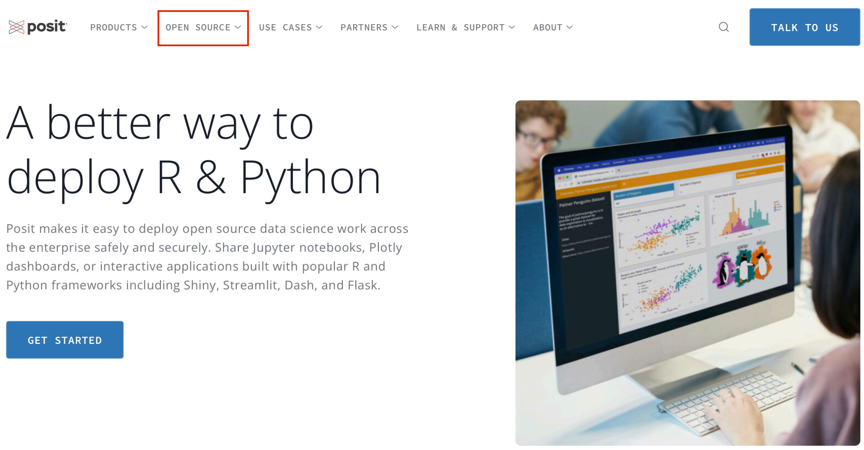
Once you click on the hyperlink, you will be prompted to choose the download for your operating system. Depending on your operating system, choose either a Windows or Macintosh download.



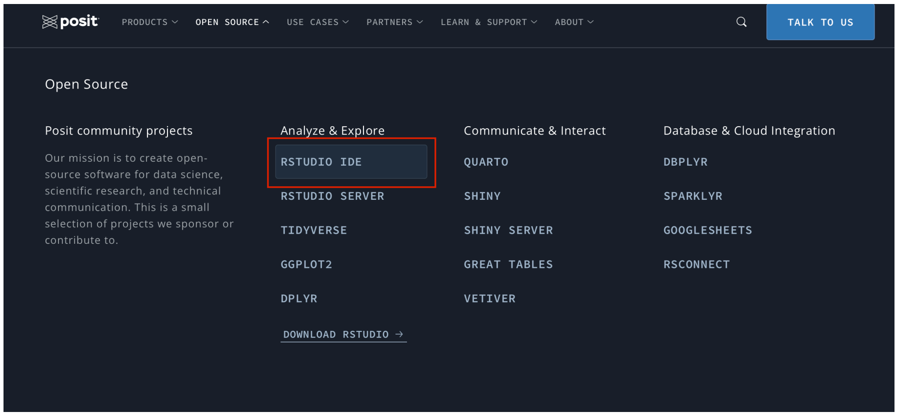
Follow all prompts and complete installation.

Installing RStudio

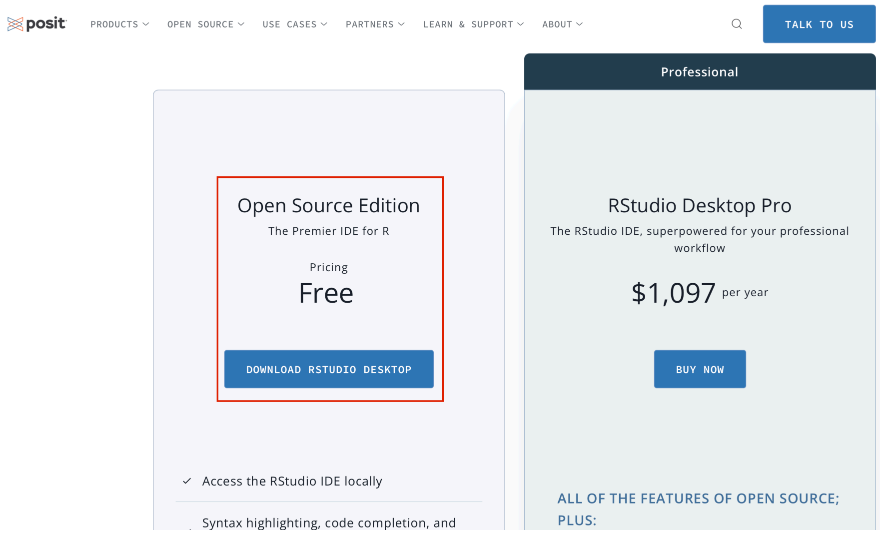
Visit the Posit website at https://posit.co. Once on the website, hover to the top of the screen and select "Open Source" from the drop down menus.



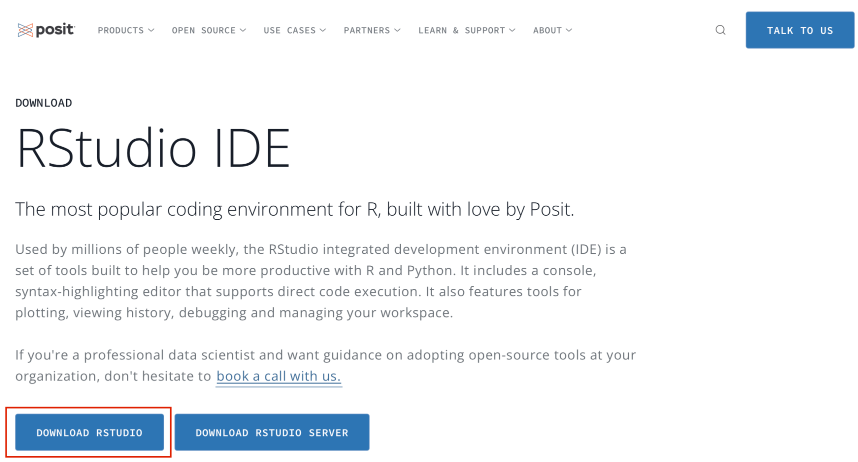
Next, choose "R Studio IDE".



Scroll down until you see the products. You want to download "RStudio Desktop" and make sure it is the free version.



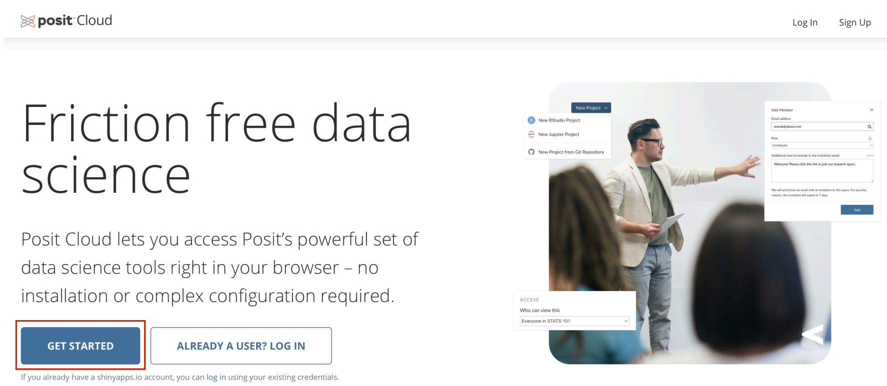
Finally, select "Download RStudio" and follow the instructions for installation.



It is important to note that RStudio will not work if R is not installed. You can think of R as the engine and RStudio as the interface.

Posit Cloud

If you do not wish to install R, you can always use the cloud version. To do this, visit https://posit.cloud/. On the main page click on the "Get Started" button.



Choose the "Cloud Free" option and log in using your Google credentials (if you have a Google account) or sign up if you want to create a new account.

1 Descriptive Stats I

1.1 Motivation

Understanding the nature and classification of data is crucial for effective analysis and decision-making. Data are the building blocks of insights, providing a foundation for businesses, researchers, and policymakers to make informed choices. Whether capturing a snapshot of a specific moment, tracking changes over time, or organizing information in structured or unstructured formats, how data is collected and categorized significantly impacts how it is analyzed and interpreted. This overview highlights key types of data and their unique characteristics to help you better understand their application in various contexts.

1.2 Data and Types of Data

Data are facts and figures collected, analyzed and summarized for presentation and interpretation. Data can be classified as:

- **Cross Sectional Data** refers to data collected at the same (or approximately the same) point in time. *Ex: NFL standings in 1980 or Country GDP in 2015.*
- **Time Series Data** refers to data collected over several time periods. *Ex: U.S. inflation rate from 2000-2010 or Tesla deliveries from 2016-2022.*
- **Structured Data** resides in a predefined row-column format (tidy). *Ex: spreadsheet data.*
- **Unstructured Data** do not conform to a pre-defined row-column format. *Ex: Text, video, and other multimedia.*

Example: Consider a retail store analyzing its sales performance. If the store collects data on the total revenue generated *by each location* on Black Friday, it is cross-sectional data. On the other hand, if the store tracks *weekly sales for the past year* to observe trends, it is time series data. Structured data, like sales figures stored in *spreadsheets*, allows for easy comparison and analysis. Meanwhile, customer feedback gathered from *social media posts and video reviews* represents unstructured data, requiring advanced tools to extract meaningful insights.

1.3 Data Sets

A **data set** contains all data collected for a particular study. Data sets are composed of:

- **Elements** are the entities on which data are collected. *Ex: Football teams, countries, and individuals.*
- **Variables** are a set of characteristics collected for each element. *Ex: Goals scored, GDP, weight.*
- **Observations** are the set of measurements obtained for a particular element. *Ex: Salah, 20 (goals), 15 (assists). US, 2.3 (inflation), 4.5% (federal interest rate).*

Elements	Variable 1	Variable 2
Element 1	#	#
Element 2	#	#
Element 3	#	#
...	...	...

Example: Consider the dataset on electric vehicles (EV's) displayed below:

Make	Model	Range_km	TopSpeed_kmh	Price_pounds	Charge_kmh
Tesla	Model 3	415	201	39990	690
BYD	ATTO 3	330	160	37195	370
Tesla	Model 3 Long Range Dual Motor	500	201	49990	770
Tesla	Model Y Long Range Dual Motor	435	217	52990	670
BYD	SEAL 82.5 kWh AWD Excellence	490	180	48695	540
Tesla	Model Y	350	217	44990	580
MG	MG4 Electric 64 kWh	360	160	29495	630
Renault	Scenic E-Tech EV87 220hp	490	170	40995	510
BYD	DOLPHIN 60.4 kWh	340	160	30195	340
BMW	i4 eDrive40	515	190	57890	800

In this dataset, each row represents an electric vehicle model, making the elements the specific EV models rather than the manufacturers. The variables collected for each model include:

- Make: The manufacturer of the EV.
- Model: The specific name of the EV model.
- Range_km: Driving range in kilometers on a full charge.
- TopSpeed_kmh: Maximum speed in km/h.
- Price_pounds: Price in pounds (£).
- Charge_kmh: Charging speed in kilometers per hour.

An example observation is "Tesla Model 3," with the following data: Make: Tesla, Model: Model 3, Range_km: 415, TopSpeed_kmh: 201, Price_pounds: 39,990, Charge_kmh: 690.

1.4 Scales of Measurement

Understanding scales of measurement is crucial for analyzing and interpreting data effectively in business. By distinguishing between categorical (e.g., marital status, satisfaction ratings) and numerical data (e.g., profits, prices), you'll know what methods to use for analysis. Knowing whether data is nominal, ordinal, interval, or ratio ensures your analysis and conclusions are accurate and relevant.

The **scales of measurements** determine the amount and type of information contained in each variable. In general, variables can be classified as **categorical** or **numerical**.

- **Categorical** (qualitative) data includes labels or names to identify an attribute of each element. Categorical data can be **nominal** or **ordinal**.
  - With **nominal** data, the order of the categories is arbitrary. *Ex: Marital Status, Race/Ethnicity, or NFL division.*
  - With **ordinal** data, the order or rank of the categories is meaningful. *Ex: Rating, Difficulty Level, or Spice Level.*
- **Numerical** (quantitative) include numerical values that indicate how many (discrete) or how much (continuous). The data can be either **interval** or **ratio**.
  - With **interval** data, the distance between values is expressed in terms of a fixed unit of measure. The zero value is arbitrary and does not represent the absence of the characteristic. Ratios are not meaningful. *Ex: Temperature or Dates.*
  - With **ratio** data, the ratio between values is meaningful. The zero value is not arbitrary and represents the absence of the characteristic. *Ex: Prices, Profits, Wins.*

Example: Let's keep using the EV example. Consider the new data set below:

Car	Brand	Range	Rating	Year
Mustang Mach-E	Ford	217	4	2021
E-Tron GT	Audi	250	3	2020
...	...	...	...	
Volt EV	Chevrolet	124	2	2021

The variables can be classified as follows: Car (Categorical - Nominal), consists of names of cars, which are labels used to identify each row. The order of these names does not matter, making it nominal data. Brand (Categorical - Nominal) represents the manufacturer of the car (e.g., Ford, Audi). These are labels with no inherent order, making it nominal data. Range

(Numerical - Ratio), refers to the car's driving range in miles. It is numerical and ratio because it has a meaningful zero (a car with zero range cannot move), and ratios are meaningful (e.g., a car with 250 miles range has double the range of one with 125 miles). Rating (Categorical - Ordinal) represents a rank or score (e.g., 4, 3, 2). The order matters, as higher ratings indicate better performance. However, the intervals between ratings are not consistent, so it is ordinal data. Year (Numerical - Interval) represents a point in time. While numerical, it is interval data because the zero point is arbitrary (e.g., year 0 does not indicate the "absence" of time), and ratios are not meaningful (e.g., 2020 is not "twice as late" as 1010).

1.5 Useful Base R Functions

Understanding and using Base R functions is essential for efficiently managing and analyzing data. Functions like `na.omit()` help clean datasets by removing rows with missing values, ensuring your analyses are accurate and complete. `nrow()` and `ncol()` quickly provide insights into the size of your dataset, while `is.na()` allows you to identify and address missing data. The `summary()` function is a powerful way to generate descriptive statistics and assess the overall structure of your data at a glance. Additionally, coercion functions like `as.integer()`, `as.factor()`, and `as.double()` enable you to convert variables to appropriate data types, ensuring compatibility with different analysis methods.

- The `na.omit()` function removes any observations that have a missing value (NA). The resulting data frame has only complete cases. *Input: A data frame (tibble) or vector.*
- The `nrow()` and `ncol()` functions return the number of rows and columns respectively from a data frame. *Input: A data frame (tibble).*
- The `is.na()` function returns a vector of *True* and *False* that specify if an entry is missing (NA) or not. *Input: A data frame (tibble) or vector.*
- The `summary()` function returns a collection of descriptive statistics from a data frame (or vector). The function also returns whether there are any missing values (NA) in a variable. *Input: A data frame (tibble) or vector.*
- The `as.integer()`, `as.factor()`, `as.double()`, are functions used to coerce your data into a different scale of measurement. *Input: A vector or column of a data frame (tibble).*

1.6 Useful DPLYR Functions

The `dplyr` package has a collection of functions that are useful for data manipulation and transformation. If you are interested in this package you can refer to Wickham (2017). To install, run the following command in the console `install.packages("dplyr")`.

- The `arrange()` function allows you to sort data frames in ascending order. Pair with the `desc()` function to sort the data in descending order.

- The `filter()` function allows you to subset the rows of your data based on a condition.
- The `select()` function allows you to select a subset of variables from your data frame.
- The `mutate()` function allows you to create a new variable.
- The `group_by()` function allows you to group your data frame by categories present in a given variable.
- The `summarise()` function allows you to summarise your data, based on groupings generated by the `goup_by()` function.

1.7 Exercises

The following exercises will help you test your knowledge on the Scales of Measurement. They will also allow you to practice some basic data "wrangling" in R. In these exercises you will:

- Identify numerical and categorical data.
- Classify data according to their scale of measurement.
- Sort and filter data in R.
- Handle missing values (NA's) in R.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

A bookstore has compiled data set on their current inventory. A portion of the data is shown below:

Title	Price	Year Published	Rating
Frankenstein	5.49	1818	4.2
Dracula	7.60	1897	4.0
...	...	...	...
Sleepy Hollow	6.95	1820	3.8

1. Which of the above variables are categorical and which are numerical?

Answer

The "Title" variable represents the names of books. Therefore, this is a categorical variable. "Price" represents the cost of each book in a numeric format, making it a numerical variable. "Year Published" indicates the publication year of each book. It is numerical. If "Rating" represents a numerical score based on a continuous scale (e.g., average user ratings on a platform like

Goodreads), it is numerical because arithmetic operations like averaging or comparing differences are meaningful. If "Rating" represents predefined categories (e.g., "Excellent," "Good," "Fair," "Poor") or is interpreted as ranks without meaningful differences between values, it would be categorical.

2. What is the measurement scale of each of the above variable?

Answer

The measurement scale is nominal for Title since these are labels used to identify each book and do not have a numerical meaning or order. If Rating represents a score (e.g., 4.2, 4.0) given to each book, it is numerical and could be considered interval data because the scale represents a meaningful difference, but it may not have an absolute zero or meaningful ratios (e.g., a book rated 4.0 is not "twice as good" as one rated 2.0). Price is a measurable quantity with a meaningful zero (e.g., a book priced at $0 means it is free), making it ratio data. Year is interval data because the zero point is arbitrary (year 0 does not represent the absence of time) and differences between years are meaningful (e.g., 1897 - 1818 = 79 years).

Exercise 2

A car company tracks the number of deliveries every quarter. A portion of the data is shown below:

Year	Quarter	Deliveries
2016	1	14800
2016	2	14400
...	...	...
2022	3	343840

1. What is the measurement scale of the Year variable? What are the strengths and weaknesses of this type of measurement scale?

Answer

The variable Year is measured on the interval scale because the observations can be ranked, categorized and measured when using this kind of scale. However, there is no true zero point so we cannot calculate meaningful ratios between years.

2. What is the measurement scale for the Quarter variable? What is the weakness of this type of measurement scale?

Answer

The variable Quarter is measured on the ordinal scale, even though it contains numbers. It is the least sophisticated level of measurement because if we are presented with nominal data, all we can do is categorize or group the data.

3. What is the measurement scale for the Deliveries variable? What are the strengths of this type of measurement scale?

Answer

The variable Deliveries is measured on the ratio scale. It is the strongest level of measurement because it allows us to categorize and rank the data as well as find meaningful differences between observations. Also, with a true zero point, we can interpret the ratios between observations.

Exercise 3

Use the **airquality** data set included in R for this problem.

1. Sort the data by *Temp* in descending order. What is the day and month of the first observation on the sorted data?

Answer

The day and month of the first observation is August 28th.

The easiest way to sort in R is by using the `dplyr` *package. Specifically, the* `arrange()` *function within the package. Let's also use the* `desc()` *function to make sure that the data is sorted in descending order. We can use indexing to retrieve the first row of the sorted data set.*

```
library(dplyr)
SortedAQ<-arrange(airquality,desc(Temp))
SortedAQ[1,]
```

```
  Ozone Solar.R Wind Temp Month Day
1    76     203  9.7   97     8  28
```

2. Sort the data only by *Temp* in descending order. Of the 10 hottest days, how many of them were in July?

Answer

We can use the `arrange()` *function one more time for this question. Then we can use indexing to retrieve the top* 10 observations.

```r
SortedAQ2<-arrange(airquality,desc(Temp))
SortedAQ2[1:10,]
```

```
   Ozone Solar.R Wind Temp Month Day
1     76     203  9.7   97     8  28
2     84     237  6.3   96     8  30
3    118     225  2.3   94     8  29
4     85     188  6.3   94     8  31
5     NA     259 10.9   93     6  11
6     73     183  2.8   93     9   3
7     91     189  4.6   93     9   4
8     NA     250  9.2   92     6  12
9     97     267  6.3   92     7   8
10    97     272  5.7   92     7   9
```

3. How many missing values are there in the data set? What rows have missing values for *Solar.R*?

Answer

There are a total of 44 missing values. Ozone has 37 and Solar.R has 7. Rows 5, 6, 11, 27, 96, 97, 98 are missing for Solar.R.

We can easily identify missing values with the `summary()` *function.*

```r
summary(airquality)
```

```
     Ozone           Solar.R           Wind             Temp      
 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
 NA's   :37       NA's   :7                                       
     Month            Day      
 Min.   :5.000   Min.   : 1.0  
 1st Qu.:6.000   1st Qu.: 8.0  
 Median :7.000   Median :16.0  
 Mean   :6.993   Mean   :15.8  
 3rd Qu.:8.000   3rd Qu.:23.0  
 Max.   :9.000   Max.   :31.0  
```

To view the rows that have NA's in them, we can use the ***is.na()*** *function and indexing. Below we see that* 7 values are missing for the Solar.R variable in the months 5 and 8 combined.

```
airquality[is.na(airquality$Solar.R),]
```

```
   Ozone Solar.R Wind Temp Month Day
5     NA      NA 14.3   56     5   5
6     28      NA 14.9   66     5   6
11     7      NA  6.9   74     5  11
27    NA      NA  8.0   57     5  27
96    78      NA  6.9   86     8   4
97    35      NA  7.4   85     8   5
98    66      NA  4.6   87     8   6
```

4. Remove all observations that have a missing values. Create a new object called *CompleteAG*.

Answer

To create the new object of complete observations we can use the ***na.omit()*** *function.*

```
CompleteAQ<-na.omit(airquality)
```

5. When using *CompleteAG*, how many days was the temperature at least 60 degrees?

Answer

There were 107 days where the temperature was at least 60.

Using base R we have:

```
nrow(CompleteAQ[CompleteAQ$Temp>=60,])
```

```
[1] 107
```

We can also use ***dplyr*** *for this question. Specifically, using the* ***filter()*** *and* ***nrow()*** *functions we get:*

```
nrow(filter(CompleteAQ,Temp>=60))
```

```
[1] 107
```

6. When using *CompleteAG*, how many days was the temperature within [55,75] degrees and an *Ozone* below 20?

Answer

There were 24 days where the temperature was between 55 and 75 and the ozone level was below 20.

Using base R we have:

```
nrow(CompleteAQ[CompleteAQ$Temp>55 & CompleteAQ$Temp<75 & CompleteAQ$Ozone<20,])
```

```
[1] 24
```

Using the `filter()` function once more we get:

```
nrow(filter(CompleteAQ,Temp>55,Temp<75,Ozone<20))
```

```
[1] 24
```

Exercise 4

Use the **Packers** data set for this problem. You can find the data set at https://jagelves.github.io/Data/Packers.c

1. Remove the any observation that has a missing value with the `na.omit()` function. How many observations are left in the data set?

Answer

There are 84 observations in the complete cases data set.

Let's import the data to R by using the `read.csv()` function.

```
Packers<-read.csv("https://jagelves.github.io/Data/Packers.csv")
```

We can remove any missing observation by using the `na.omit()` function. We can name this new object Packers2.

```
Packers2<-na.omit(Packers)
```

To find the number of observations we can use the `dim()` function. It returns the number of observations and variables.

```r
dim(Packers2)
```

```
[1] 84  8
```

2. Determine the type of the *Experience* variable by using the `typeof()` function. What type is the variable?

Answer

The type is character.

Use the `typeof()` function on the Experience variable.

```r
typeof(Packers2$Experience)
```

```
[1] "character"
```

3. Remove observations that have an "R" and coerce the *Experience* variable to an integer using the `as.integer()` function. What is the total sum of years of experience?

Answer

The total sum of experience is 288.

First, remove any observation with an R by using indexing and logicals.

```r
Packers2<-Packers2[Packers2$Experience!="R",]
```

Now we can coerce the variable to an integer by using the `as.integer()` function.

```r
Packers2$Experience<-as.integer(Packers2$Experience)
```

Lastly, calculate the sum using the `sum()` function.

```r
sum(Packers2$Experience)
```

```
[1] 288
```

2 Descriptive Stats II

Understanding and visualizing data distribution is a fundamental step in data analysis. A frequency distribution organizes a structured data summary into non-overlapping classes, allowing for insights into patterns and trends. Complementary to this, relative frequency, cumulative frequency, and cumulative relative frequency offer deeper perspectives on the proportions and accumulation of data within these classes. Visualization techniques, including bar plots and histograms, play a crucial role in representing these distributions, with bar plots suited for qualitative data and histograms tailored for quantitative data. The R package `ggplot2`, has functions like `geom_bar()` and `geom_hist()` to plot distributions efficiently. By leveraging these methods, data can be transformed into clear and meaningful insights.

2.1 Frequency Distributions (Categorical)

A **frequency distribution** is perhaps the most valuable tool for summarizing categorical data. It illustrates with a table the number of items within distinct, non-overlapping categories. Presenting the data in a tabular format makes it easier to identify patterns and trends within the categories. A key component of this analysis is the **relative frequency**, which quantifies the proportion of items in each category relative to the total number of observations. You can calculate it by taking the frequency of a particular class (f_i/n), and dividing it by the total frequency n. Relative frequency helps contextualize the data by highlighting the significance of each category compared to the whole.

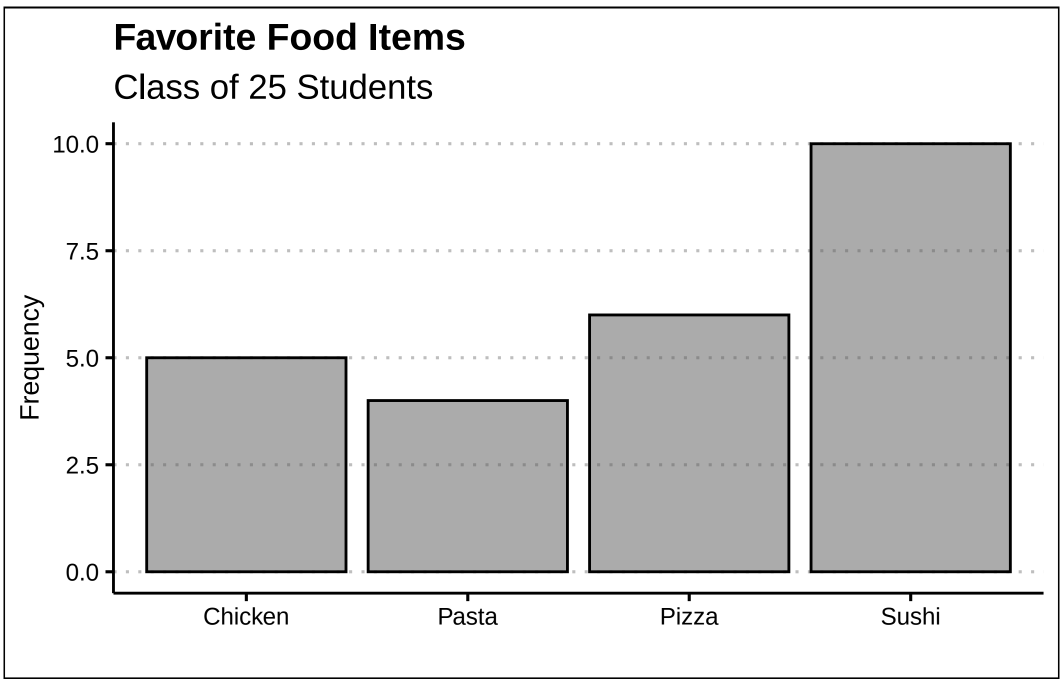
Example: Consider data on students' answers to the question, what is your favorite food? You can see the data below:

PIZZA	SUSHI	SUSHI	CHICKEN	CHICKEN
PASTA	PASTA	PASTA	SUSHI	PASTA
CHICKEN	PIZZA	CHICKEN	SUSHI	PIZZA
SUSHI	SUSHI	SUSHI	SUSHI	PIZZA
PIZZA	CHICKEN	SUSHI	PIZZA	SUSHI

Simply observing raw data can make identifying the most and least popular items challenging. A frequency distribution organizes this information into a clear table, showcasing the popularity of each item. The frequency distribution of the table is displayed below:

Food	Frequency	Relative
Chicken	5	0.20
Pasta	4	0.16
Pizza	6	0.24
Sushi	10	0.40

Each food item is tallied up, and the result is shown in the Frequency column. We can also show the tally result as a ratio of the total food items recorded in the data (i.e., 25). For example, five students liked chicken; out of the 25 students surveyed, this represents 0.2 or 20%. This approach makes it much easier to pinpoint the most popular and the least popular items. Below, you can see the bar graph showing the frequency distribution of the food items data. Note that the visualization is constructed by showing each food item as a bar with a height equal to the frequency.



In sum, the **bar plot** illustrates the frequency distribution of categorical data. It includes the classes in the horizontal axis and frequencies or relative frequencies in the vertical axis and has gaps between each bar.

2.2 Frequency Distributions (Numerical)

- The **cumulative frequency** shows the number of data items with values less than or equal to the upper class limit of each class.
- The **cumulative relative frequency** is given by cf_i/n, where cf_i is the cumulative frequency of class i.

2.3 Plots Using ggplot2

A **bar plot** illustrates the frequency distribution of qualitative data.

- Is an illustration for qualitative data.
- Includes the classes in the horizontal axis and frequencies or relative frequencies in the vertical axis.
- Has gaps between each bar.

A **histogram** illustrates the frequency distribution of quantitative data.

- Is an illustration for quantitative data.
- There are no gaps between the bars.
- The **number**, **width** and **limits** of each class must be determined.
  - The **number** of classes can be determined by the 2^k rule: select k such that 2^k is greater than the number of observations by the smallest amount.
  - The **width** of the class is approximately *range/(# of Classes)*. The value should be rounded up.
  - The **limits** should be chosen so that each point belongs to only one class.

Useful R Functions

The `table()` command generates frequency distributions or contingency tables if two variables are used.

The `prop.table()` command generates relative frequency distributions from an object that contains a table.

The `cut()` function generates class limits and bins used in frequency distributions (and histograms) for quantitative data.

Base R has the `barplot()` function for categorical variable, `histogram()` function for numerical data, and the `plot()` function for line charts or scatter plots. Below are some arguments that are helpful when plotting.

- *main*: used to set the plot's title. The title should be entered as a character.
- *col*: used to set the color of the plot. Hex and RGB values are allowed as inputs. The color should be entered as a character.
- *xlab* and *ylab*: are used to set the labels for the x and y axis respectively. The labels should be entered as characters.
- `legend()` is a function to customize the legend of a graph. This argument may be used with the `plot()`, `barplot()` or `histogram()` functions.
  - x: used to set the location of the legend in the plotting area. Ex: "bottomleft".
  - *legend*: a vector specifying the legend names to be included.
  - *col*: a vector specifying the color of each item in the legend.

2.4 Exercises

The following exercises will help you practice summarizing data with tables and simple graphs. In particular, the exercises work on:

- Developing frequency distributions for both categorical and numerical data.
- Constructing bar charts, histograms, and line charts.
- Creating contingency tables.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

Install the `ISLR2` package in R. You will need the **BrainCancer** data set to answer this question.

1. Construct a frequency and relative frequency table of the *Diagnosis* variable. What was the most common diagnosis? What percentage of the sample had this diagnosis?

Answer

The most common diagnosis is Meningioma, a slow-growing tumor that forms from the membranous layers surrounding the brain and spinal cord. The diagnosis represents about 48.28% *of the sample.*

Start by loading the `ISLR2` package. To construct the frequency distribution table, use the `table()` function.

```r
library(ISLR2)
table(BrainCancer$diagnosis)
```

```

Meningioma  LG glioma  HG glioma      Other 
        42          9         22         14 
```

The relative frequency distribution can be easily retrieved by saving the frequency table in an object and then using the `prop.table()` function.

```r
freq<-table(BrainCancer$diagnosis)
prop.table(freq)
```

```

Meningioma  LG glioma  HG glioma      Other 
 0.4827586  0.1034483  0.2528736  0.1609195 
```

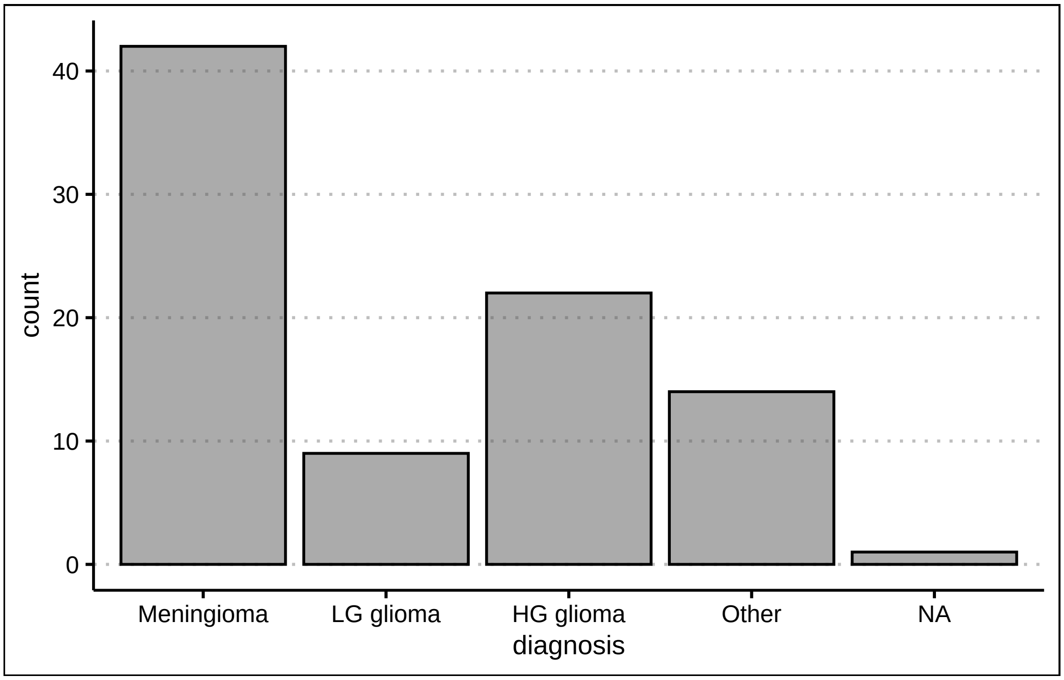
2. Construct a bar chart. Summarize the findings.

Answer

The majority of diagnosis are Meningioma. Low grade glioma is the least common of diagnosis. High grade glioma and other diagnosis have about the same frequency.

To construct the bar chart use the `geom_bar()` function from `tidyverse`.

```r
library(tidyverse)
library(ggthemes)
ggplot(data=BrainCancer) +
  geom_bar(aes(diagnosis), alpha=0.5, col="black") +
  theme_clean()
```



3. Construct a contingency table that shows the *Diagnosis* along with the *Status*. Which diagnosis had the highest number of non-survivals (0)? What was the survival rate of this diagnosis?

Answer

33 people did not survive Meningioma. The survival rate of Meningioma is only 21.43%.

Use the `table()` function one more time to create the contingency table for the two variables.

```
(freq2<-table(BrainCancer$status,BrainCancer$diagnosis))
```

```
   Meningioma LG glioma HG glioma Other
  0         33         5         5     9
  1          9         4        17     5
```

To get the survival rates, we can use the `prop.table()` function once again.

```
prop.table(freq2,margin = 2)
```

```
   Meningioma LG glioma HG glioma     Other
  0  0.7857143 0.5555556 0.2272727 0.6428571
  1  0.2142857 0.4444444 0.7727273 0.3571429
```

Exercise 2

You will need the **airquality** data set (in base R) to answer this question.

1. Construct a frequency distribution for *Temp*. Use five intervals with widths of $50 < x \leq 60$; $60 < x \leq 70$; etc. Which interval had the highest frequency? How many times was the temperature between 50 and 60 degrees?
2. Construct a relative frequency, cumulative frequency and the relative cumulative frequency distributions. What proportion of the time was *Temp* between 50 and 60 degrees? How many times was the *Temp* 70 degrees or less? What proportion of the time was *Temp* more than 70 degrees?
3. Construct the histogram. Is the distribution symmetric? If not, is it skewed to the left or right?

Exercise 3

You will need the **Portfolio** data set from the `ISLR2` package to answer this question.

1. Construct a line chart that shows the returns over time for each portfolio (X and Y) by using two lines each with a unique color. Assume the data is for the period 1901 to 2000. Include also a legend that matches colors to portfolios.

2.5 Answers

Exercise 1

1. The most common diagnosis is Meningioma, a slow-growing tumor that forms from the membranous layers surrounding the brain and spinal cord. The diagnosis represents about 48.28% of the sample.

Start by loading the `ISLR2` package. To construct the frequency distribution table, use the `table()` function.

```
library(ISLR2)
table(BrainCancer$diagnosis)
```

```
Meningioma  LG glioma  HG glioma      Other 
        42          9         22         14 
```

The relative frequency distribution can be easily retrieved by saving the frequency table in an object and then using the `prop.table()` function.

```
freq<-table(BrainCancer$diagnosis)
prop.table(freq)
```

```
Meningioma   LG glioma   HG glioma       Other 
 0.4827586   0.1034483   0.2528736   0.1609195 
```

2. The majority of diagnosis are Meningioma. Low grade glioma is the least common of diagnosis. High grade glioma and other diagnosis have about the same frequency.

To construct the bar chart use the `barplot()` function in R.

```
barplot(freq, col = "#F5F5F5", ylim=c(0,50))
```

3. 33 people did not survive Meningioma. The survival rate of Meningioma is only 21.43%.

Use the `table()` function one more time to create the contingency table for the two variables.

```
(freq2<-table(BrainCancer$status,BrainCancer$diagnosis))
```

```
   
    Meningioma LG glioma HG glioma Other
  0         33         5         5     9
  1          9         4        17     5
```

To get the survival rates, we can use the `prop.table()` function once again.

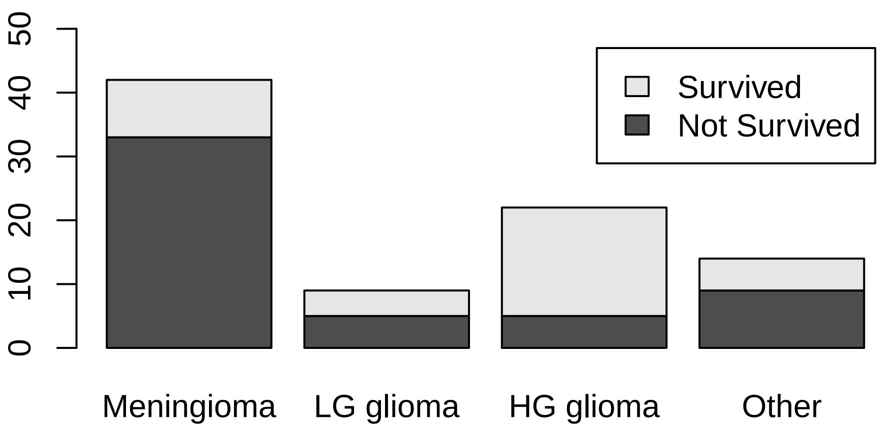
```r
prop.table(freq2,margin = 2)
```

```
    Meningioma LG glioma HG glioma     Other
  0  0.7857143 0.5555556 0.2272727 0.6428571
  1  0.2142857 0.4444444 0.7727273 0.3571429
```

4. Meningioma and not surviving is the most common with 33 occurrences. High grade glioma and surviving is the the second most common.

Use the `barplot()` function one more time to construct the stacked column chart.

```r
barplot(table(BrainCancer$status,BrainCancer$diagnosis),
        legend.text = c("Not Survived","Survived"), ylim=c(0,50))
```



Exercise 2

1. The highest frequency is in the $80 < x \leq 90$ bin. 8 temperatures were between $50 < x \leq 60$ degrees.

Create a vector containing the intervals desired by using the `seq()` function.

```r
intervals <- seq(50, 100, by=10)
```

Next use the `cut()` function to create the cuts for the histogram.

```r
intervals.cut <- cut(airquality$Temp, intervals, left=FALSE, right=TRUE)
```

The frequency distribution can be obtained by using the `table()` function on the *interval.cut* object created above.

```r
table(intervals.cut)
```

```
intervals.cut
 (50,60]  (60,70]  (70,80]  (80,90] (90,100]
       8       25       52       54       14
```

2. The temperature was 5.22% of the time between 50 and 60; The temperature was 70 or less 33 times; The temperature was above 70, 78.43% of the time.

To get the relative frequency table, start by saving the proportion table into an object.Then you can use the `prop.table()` function.

```r
freq<-table(intervals.cut)
prop.table(freq)
```

```
intervals.cut
   (50,60]    (60,70]    (70,80]    (80,90]   (90,100]
0.05228758 0.16339869 0.33986928 0.35294118 0.09150327
```

For the cumulative distribution you can use the `cumsum()` function on the frequency distribution.

```r
cumulfreq<-cumsum(freq)
cumulfreq
```

```
 (50,60]  (60,70]  (70,80]  (80,90] (90,100]
       8       33       85      139      153
```

Lastly, for the relative cumulative distribution table, you can use the `cumsum()` function on the relative frequency table.

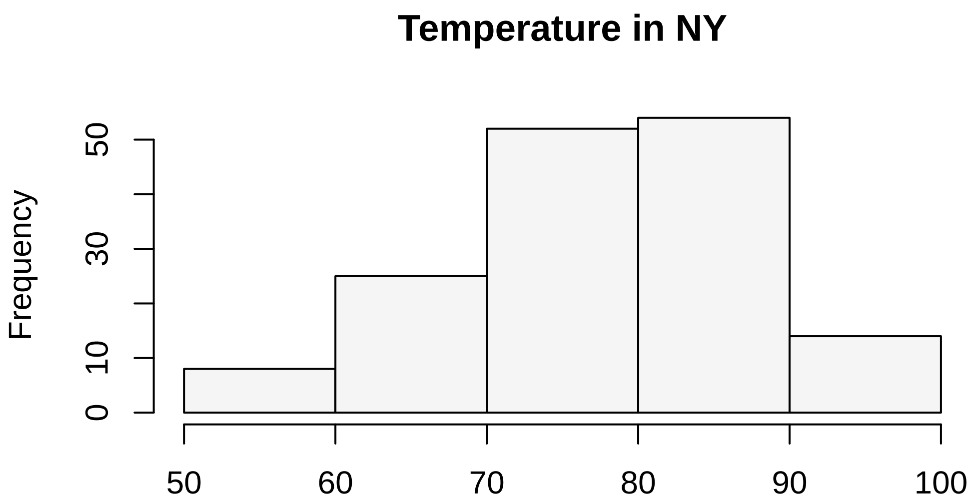
```r
cumsum(prop.table(freq))
```

```
   (50,60]    (60,70]    (70,80]    (80,90]   (90,100]
0.05228758 0.21568627 0.55555556 0.90849673 1.00000000
```

3. The distribution is not perfectly symmetric. It is skewed slightly to the left (see histogram.)

Use the `hist()` function to create the histogram.

```
hist(airquality$Temp, breaks=intervals,
     right=TRUE,col="#F5F5F5", main="Temperature in NY", xlab="")
```



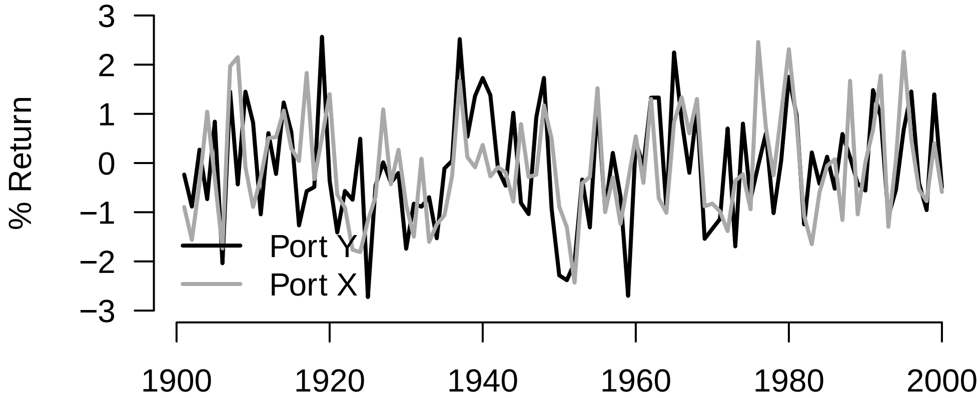
Exercise 3

1. From 1901 through 2000, both portfolios have behaved very similarly. Returns are between -3% and 3%, there is no trend, and positive (negative) returns for X seem to match with positive (negative) returns of Y.

You can use the `plot()` function to create a plot of Portfolio Y. The line for Portfolio X can be added with the `lines()` function.

```
plot(Portfolio$Y,
     x=seq(1901,2000), type="l",
     col="black", xlab="", ylab="% Return", ylim=c(-3,3),
     xlim=c(1901,2000), lwd=2, axes = F)
axis(side=1, labels=TRUE, font=1,las=1)
axis(side=2, labels=TRUE, font=1,las=1)
lines(Portfolio$X, x=seq(1901,2000), type="l",
      col="darkgrey", lwd=2)
legend(x = "bottomleft",
       legend = c("Port Y", "Port X"),
```

```
      lty = c(1, 1),
      col = c("black", "darkgrey"),
      lwd = 2,
      bty="n")
```



3 Descriptive Statistics III

3.1 Concepts

Measures of Central Location

Measures of Central Location determine where the center of a distribution lies.

- The **mean** is the average value for a numerical variable. The sample statistic is estimated by $\bar{x} = \sum x_i/n$, where x_i is observation i, and n is the number of observations. The population parameter is defined as $\mu = \sum x_i/N$.
- The **median** is the value in the middle when data is organized in ascending order. When n is even, the median is the average between the two middle values.
- The **mode** is the value with highest frequency from a set of observations.
- The **weighted mean** uses weights to determine the importance of each data point of a variable. It is calculated by $\frac{\sum w_i x_i}{\sum w_i}$, where w_i are the weights associated to the values x_i.
- The **geometric mean** is a multiplicative average that is less sensitive to outliers. It is used to average growth rates or rated of return. It is calculated by $\sqrt[n]{(1+r_1)*(1+r_2)...(1+r_n)}-1$, where $\sqrt[n]{}$ is the n_{th} root, and r_i are the returns or growth rates.

Useful R functions

Base R has a collection of functions that calculate measures of central location.

- The `mean()` function calculates the average of a vector of values.
- The `median()` function returns the median of a vector of values.
- The `table()` function provides us with a frequency distribution. We can then identify the mode(s) of the vector provided.
- The `summary()` function returns a collection of descriptive statistics for a vector or data frame.

3.2 Exercises

The following exercises will help you practice the measures of central location. In particular, the exercises work on:

- Calculating the mean, median, and the mode.
- Calculating the weighted average.
- Applying the geometric mean for growth rates and returns.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results using R functions when possible.

1. Use the following observations to calculate the mean, the median, and the mode.

8	10	9	12	12

2. Use following observations to calculate the mean, the median, and the mode.

-4	0	-6	1	-3	-4

3. Use the following observations, calculate the mean, the median, and the mode.

20	15	25	20	10	15	25	20	15

Exercise 2

Download the `ISLR2` package. You will need the **OJ** data set to answer this question.

1. Find the mean price for Country Hill (*PriceCH*) and Minute Maid (*PriceMM*).
2. Find the mean price of Country Hill (*PriceCH*) in store 1 and store 2 (*StoreID*). Which store had the better price?
3. Find the mean price paid by Country Hill (*PriceCH*) purchasers (*Purchase*) in store 1 (*StoreID*)? How about store 2? Which store had the better price?

Exercise 3

1. Over the past year an investor bought TSLA. She made these purchases on three occasions at the prices shown in the table below. Calculate the average price per share.

Date	Price Per Share	Number of Shares
February	250.34	80
April	234.59	120
Aug	270.45	50

2. What would have been the average price per share if the investor would have bought equal amounts of shares each month?

Exercise 4

1. Consider the following observations for the consumer price index (CPI). Calculate the inflation rate (Growth Rate of the CPI) for each period.

1.0	1.3	1.6	1.8	2.1

2. Suppose that you want to invest $1000 dollars in a stock that is predicted to yield the following returns in the next four years. Calculate both the arithmetic mean and the geometric mean. Use the geometric mean to estimate how much money you would have by the end of year 4.

Year	Annual Return
1	17.3
2	19.6
3	6.8
4	8.2

3.3 Answers

Exercise 1

1. To find the mean we will use the following formula ($\frac{1}{n}\sum_{i=i}^{n} x_i$). The summation of the values is 51 and the number of observations is 5. The mean is $51/5 = 10.2$.

The median is found by locating the middle value when data is sorted in ascending order. The median in this example is 10.

The mode is the value with the highest frequency. In this example the mode is 12 since it is repeated twice and all other numbers appear only once.

The mean can be easily verified in R by using the `mean()` function:

```r
mean(c(8,10,9,12,12))
```

```
[1] 10.2
```

Similarly, the median is easily verified by using the `median()` function:

```r
median(c(8,10,9,12,12))
```

```
[1] 10
```

We can use the `table()` function to calculate frequencies and easily identify the mode.

```r
table(c(8,10,9,12,12))
```

```

 8  9 10 12 
 1  1  1  2 
```

2. The mean is -2.67, the median is -3.5, the mode is -4.

These mean is verified in R:

```r
mean(c(-4,0,-6,1,-3,-4))
```

```
[1] -2.666667
```

The median in R:

```r
median(c(-4,0,-6,1,-3,-4))
```

```
[1] -3.5
```

Finally, the mode in R:

```r
table(c(-4,0,-6,1,-3,-4))
```

```
-6 -4 -3  0  1 
 1  2  1  1  1 
```

3. The mean is 18.33, the median is 20, the data is bimodal with both 15 and 20 being modes.

These mean is verified in R:

```r
mean(c(20,15,25,20,10,15,25,20,15))
```

```
[1] 18.33333
```

The median in R:

```r
median(c(20,15,25,20,10,15,25,20,15))
```

```
[1] 20
```

The frequency distribution identifies the modes:

```r
table(c(20,15,25,20,10,15,25,20,15))
```

```
10 15 20 25 
 1  3  3  2 
```

Exercise 2

1. The mean price for Country Hill is 1.87. The mean price for Minute Maid is 2.09.

The means can be easily found with the `mean()` function:

```r
library(ISLR2)
mean(OJ$PriceCH)
```

```
[1] 1.867421
```

```r
mean(OJ$PriceMM)
```

```
[1] 2.085411
```

2. The mean price at store 1 for Country Hill is 1.80 vs. 1.84 for store 2. The juice is cheaper at store 1.

The means for each store can be found by using indexing and a logical statement. The Country Hill mean price at store 1 is given by:

```r
mean(OJ$PriceCH[OJ$StoreID==1])
```

```
[1] 1.803758
```

The Country Hill mean price at store 2 is given by:

```r
mean(OJ$PriceCH[OJ$StoreID==2])
```

```
[1] 1.841216
```

3. Purchasers of Country Hill at store 1 paid and average of 1.80 for Country Hill juice. At store 2 they paid 1.86. Once again the average price was lower at store 1.

The mean for Country Hill purchasers at store 1 is given by:

```r
mean(OJ$PriceCH[OJ$StoreID==1 & OJ$Purchase=="CH"])
```

```
[1] 1.797176
```

The mean for Country Hill purchasers at store 2 is:

```r
mean(OJ$PriceCH[OJ$StoreID==2 & OJ$Purchase=="CH"])
```

```
[1] 1.857383
```

Exercise 3

1. The average price of sale is found by using the weighted average formula. $\frac{\sum w_i x_i}{\sum w_i}$ The weights (w_i) are given by the number of shares bought and the values (x_i) are the prices. The weighted average is 246.802.

In R you can create two vectors. One holds the share price and the other one the number of shares bought.

```
PricePerShare<-c(250.34,234.59,270.45)
NumberOfShares<-c(80,120,50)
```

Next, you can multiply the *PricePerShare* and *NumberOfShares* vectors to find the numerator and then use `sum()` function to find the denominator. The weighted average is:

```
(WeightedAverage<-
  sum(PricePerShare*NumberOfShares)/sum(NumberOfShares))
```

```
[1] 246.802
```

2. The average if equal shares were bought would be 251.7933.

In R you can use the `mean()` function on the PricePerShare vector.

```
(Average<-mean(PricePerShare))
```

```
[1] 251.7933
```

Exercise 4

1. The inflation rate for each period is shown in the table below:

30%	23.08%	12.5%	16.67%

In R create an object to store the values of the CPI:

```
CPI<-c(1,1.3,1.6,1.8,2.1)
```

Next use the `diff()` function to find the difference between the end value and start value. Divide the result by a vector of starting value and multiply times 100.

```
(Inflation<-100*diff(CPI)/CPI[1:4])
```

```
[1] 30.00000 23.07692 12.50000 16.66667
```

2. At the end of 4 years it is predicted that you would have 1621.17 dollars. Each year you would have gained 12.84% on average.

In R include the annual rates in a vector:

```
growth<-c(0.173,0.196,0.068,0.082)
```

The arithmetic mean is:

```
100*mean(growth)
```

```
[1] 12.975
```

The geometric mean is:

```
(geom<-((prod(1+growth))^(1/4)-1)*100)
```

```
[1] 12.8384
```

At the end of the four years we would have:

```
1000*(1+geom/100)^4
```

```
[1] 1621.167
```

4 Descriptive Stats IV

4.1 Concepts

Measures of Dispersion

Measures of dispersion are used to determine the spread (variability) of the data.

- The **range** is calculated by $largest - smallest$. It ignores the variability of the data between the largest and smallest values.
- The **variance** calculates the dispersion around the mean. It uses squared deviations. The population parameter is given by $\sigma^2 = \frac{\sum(x_i-\mu)^2}{N}$, while the sample statistic is $s^2 = \frac{\sum(x_i-\bar{x})^2}{n-1}$.
- The **standard deviation** measures the average deviation around the mean. It is calculated as the square root of the variance. For the population parameter use $\sigma = \sqrt{\sigma^2}$ and $s = \sqrt{s^2}$ for the sample statistic.
- The **Mean Absolute Deviation** (MAD) measures the average deviation from the mean. This measure uses absolute deviations. It is calculated by $MAD = \frac{\sum|x_i-\mu|}{N}$ for the population and $mad = \frac{\sum|x_i-\bar{x}|}{n}$ for the sample.
- The **coefficient of variation** $CV = s/\bar{x}$ adjusts the standard deviation for differences in units of measure or scale.

Portfolio Assesment

To asses the performance of a portfolio calculate:

- The mean return of the portfolio $\alpha\bar{R}_1 + (1-\alpha)\bar{R}_2$, where α is the weight of investment 1 in the portfolio and $\bar{R}_i$ is the average return of investment $i \in$\{1,2\}.
- The variance of the portfolio is given by $\begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}^T \begin{bmatrix} s_x^2 & s_{xy} \\ s_{xy} & s_y^2 \end{bmatrix} \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}$

- The **Sharpe ratio** quantifies the excess return of an investment over the risk free return. It is calculated by $\frac{\bar{R_p}-R_f}{s}$, where $\bar{R_p}$ is the mean return of the portfolio, R_f is the risk free return, and s is the standard deviation.

Useful R Functions

The `range()` function returns the maximum and minimum of a vector of values.

The `diff()` function finds the first difference of a vector.

The `var()` function calculates the sample variance for a vector of values. To calculate the population variance, adjust the result by a factor of $(n-1)/n$.

The `sd()` function calculates the sample standard deviation.

The `matrix()` function defines a matrix.

When dealing with matrices, the `t()` function transposes a vector or matrix, and the operator `%*%` performs matrix multiplication.

4.2 Exercises

The following exercises will help you practice the measures of dispersion. In particular, the exercises work on:

- Calculating the range, MAD, variance, and the standard deviation.
- Using R to calculate measures of dispersion.
- Calculating and using the Sharpe ratio to select investments.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results using R functions when possible. Make sure to calculate the deviations from the mean.

1. Use the following observations to calculate the Range, MAD, Variance and Standard Deviation. Assume that the data below is the entire population.

70	68	4	98

2. Use the following observations to calculate the Range, MAD, Variance and Standard Deviation. Assume that the data below is a sample from the population.

-4	0	-6	1	-3	0

Exercise 2

You will need the **Stocks** data set to answer this question. You can find this data at https://jagelves.github.io/Data/Stocks.csv The data is a sample of daily stock prices for ticker symbols TSLA (Tesla), VTI (S&P 500) and GBTC (Bitcoin).

1. Calculate the standard deviations for each stock. Which stock had the lowest standard deviation?
2. Calculate the MAD. Does your answer in 1. remain the same?
3. Finally, calculate the coefficient of variation. Any changes to your conclusions?

Exercise 3

Install the `ISLR2` package. You will need the **Portfolio** data set to answer this question. The data has 100 records of the returns of two stocks.

1. Calculate the mean and standard deviation for each stock. Which investment has higher returns on average? Which investment is safest as measured by the standard deviation?
2. Use a Risk Free rate of return of 3.5% to calculate the Sharpe ratio for each stock. Which stock would you recommend?
3. Calculate the average return for a portfolio that has 30% of stock X and 70% of stock Y. What is the standard deviation of the portfolio?

4.3 Answers

Exercise 1

1. The mean is 60, the Range is 94, the MAD is 28, the variance is 1186 and the variance is 34.44.

Start by crating a vector to hold the values:

```
Ex1<-c(70,68,4,98)
```

The range can be calculated by using the `range()` and `diff()` functions in R.

```r
(Range<-diff(range(Ex1)))
```

```
[1] 94
```

Next, we can create a table by hand that captures the deviations from the mean. Let's calculate the mean first:

```r
(Average1<-mean(Ex1))
```

```
[1] 60
```

Now we can use the mean to fill out a table of deviations:

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$	$\lvert x_i - \bar{x} \rvert$
70	10	100	10
68	8	64	8
4	-56	3136	56
98	38	1444	38

The variance averages out the squared deviations $(x_i - \bar{x})^2$, the MAD averages out the absolute deviations $|x_i - \bar{x}|$, and the standard deviation is the square root of the variance.

Let's verify the variance in R:

```r
SquaredDeviations1<-(Ex1-Average1)^2
AverageDeviations1<-mean(SquaredDeviations1)
var(Ex1)*3/4
```

```
[1] 1186
```

Note that R calculates the sample variance. Hence, we must multiply the result by 3/4 to get the population variance. The standard deviation is just the square root of the variance:

```r
sqrt(AverageDeviations1)
```

```
[1] 34.43835
```

Lastly, the MAD is calculated by averaging the absolute deviations $|x_i - \bar{x}|$.

```r
AbsoluteDeviations1<-abs(Ex1-Average1)
mean(AbsoluteDeviations1)
```

```
[1] 28
```

2. The mean is -2, Range is 7, the MAD is 2.33, the variance is 7.6 and the standard deviation is 2.76.

Here is the table of deviations from the mean:

x_i	$x_i - \bar{x}$	$(x_i - \bar{x})^2$	$\|x_i - \bar{x}\|$
-4	-2	4	2
0	2	4	2
-6	-4	16	4
1	3	9	3
-3	-1	1	1
0	2	4	2

We can check the results in R. Let's start with the variance:

```r
Ex2<-c(-4,0,-6,1,-3,0)
var(Ex2)
```

```
[1] 7.6
```

The standard deviation can be found with the `sd()` function:

```r
sd(Ex2)
```

```
[1] 2.75681
```

The MAD is given by:

```r
(MAD<-mean(abs(Ex2-mean(Ex2))))
```

```
[1] 2.333333
```

Lastly, the range:

```r
diff(range(Ex2))
```

```
[1] 7
```

Exercise 2

1. For the sample taken, GBTC has the less variation. The standard deviation of GBTC is 9.43, which is less than 16.57 for VTI or 50.38 for TSLA.

Start by loading the data set from the website. Since the file is in csv format, we will use the `read.csv()` function.

```r
StockPrices<-read.csv("https://jagelves.github.io/Data/Stocks.csv")
```

Let's start with the standard deviation of the Tesla stock. The standard deviation is given by:

```r
sd(StockPrices$TSLA)
```

```
[1] 50.38092
```

Next, let's find the standard deviation for the S&P 500 or VTI. The standard deviation is given by:

```r
sd(StockPrices$VTI)
```

```
[1] 16.5731
```

Finally, let's calculate the standard deviation for GBTC or Bitcoin.

```r
sd(StockPrices$GBTC)
```

```
[1] 9.434213
```

2. The answer is the same, since the MAD for GBTC is 8.46 which is lower than 14.27 for VTI or 41.67 for TSLA.

To calculate the MAD for TSLA we can use the following command:

```r
(MADTSLA<-mean(abs(StockPrices$TSLA-mean(StockPrices$TSLA))))
```

```
[1] 41.67163
```

The MAD for VTI is:

```r
(MADVTI<-mean(abs(StockPrices$VTI-mean(StockPrices$VTI))))
```

```
[1] 14.27169
```

The MAD for GBTC is:

```r
(MADGBTC<-mean(abs(StockPrices$GBTC-mean(StockPrices$GBTC))))
```

```
[1] 8.458029
```

3. By considering the magnitudes of the stock prices, it seems like VTI is the less volatile stock. VTI has a CV of 0.08 which is lower than 0.44 for GBTC or 0.18 for TSLA. In fact, by CV Bitcoin seems to be the most risky asset.

The coefficients of variations are as follows. For TSLA the CV is:

```r
(CVTSLA<-sd(StockPrices$TSLA)/mean(StockPrices$TSLA))
```

```
[1] 0.1793755
```

For VTI the CV is:

```r
(CVVTI<-sd(StockPrices$VTI)/mean(StockPrices$VTI))
```

```
[1] 0.07970004
```

For GBTC we get:

```r
(CVGBTC<-sd(StockPrices$GBTC)/mean(StockPrices$GBTC))
```

```
[1] 0.4442497
```

Exercise 3

1. The best performing stock on average is stock X. It has an average return of -0.078% vs. 0.097% for stock Y. The safest stock is stock X as well, since the standard deviation is 1.062 percentage points vs. 1.14 percentage points for stock Y.

Start by loading the `ISLR2` package:

```
library(ISLR2)
```

Next, calculate the mean for stock X:

```
mean(Portfolio$X)
```

```
[1] -0.07713211
```

and stock Y.

```
mean(Portfolio$Y)
```

```
[1] -0.09694472
```

Then, calculate the standard deviation for stock X

```
sd(Portfolio$X)
```

```
[1] 1.062376
```

and stock Y.

```
sd(Portfolio$Y)
```

```
[1] 1.143782
```

2. The Sharpe Ratio measures the excess return per unit of risk taken. Stock X has the better Sharpe Ratio. -0.106 vs. -0.115. Stock X is recommended since it provides a higher excess return per unit of risk taken.

To calculate Sharpe Ratios use both the average return, and the standard deviation. For stock X, the Sharpe Ratio is:

```r
(mean(Portfolio$X)-0.035)/sd(Portfolio$X)
```

```
[1] -0.1055484
```

The Sharpe Ratio for stock Y:

```r
(mean(Portfolio$Y)-0.035)/sd(Portfolio$Y)
```

```
[1] -0.1153583
```

3. The portfolio has an average return of -0.091 which is worse than stock X but better than stock Y. The standard deviation is 1.00. This is better than stock X and Y separately. The Sharpe ratio of -0.091 is also better for the portfolio than for each stock individually.

The mean of the portfolio is given by:

```r
(mean_return=0.3*mean(Portfolio$X)+0.7*mean(Portfolio$Y))
```

```
[1] -0.09100094
```

The covariance matrix is given by:

```r
(risk<-cov(Portfolio))
```

```
          X         Y
X 1.1286424 0.6263583
Y 0.6263583 1.3082375
```

Using the matrix we can now calculate the standard deviation:

```r
(standard<-sqrt(t(c(0.3,0.7)) %*% (risk %*% c(0.3,0.7))))
```

```
         [,1]
[1,] 1.002838
```

Finally, the Sharpe ration for the portfolio is:

```r
mean_return/standard[1]
```

```
[1] -0.09074338
```

5 Descriptive Stats V

5.1 Concepts

Quantiles and Percentiles

A **quantile** is a location within a set of ranked numbers (or distribution), below which a certain proportion, q, of that set lie. Ex: 0.25 of the data lies below the 0.25 quantile.

Percentiles express quantiles in percentage form. Ex: 25% of the data lies below the 25th percentile. To calculate a percentile:

- Sort the data in ascending order.
- Compute the location of the percentile desired using $L_p = \frac{(n+1)P}{100}$ where L_p is the location of the P_{th} percentile, and P is the percentile desired.
- The value at L_p, is the the P_{th} percentile.

Chevyshev's Theorem and Empirical Rule

Chevyshev's Theorem states that at least $1 - 1/z^2\%$ of the data lies between z standard deviations from the mean. This result does not depend on the shape of the distribution.

The **Empirical Rule** or (68,95,99.7 rule) states that 68%, 95%, and 99.7% of the data lies between 1, 2, and 3 standard deviations from the mean respectively. The rule depends on the data being normally distributed.

Five Point Summary and Outliers

A popular way to summarize data is by calculating the minimum, first quartile, median, third quartile and maximum (five point summary).

The **interquartile range** (IQR) is the difference between the third quartile and the first quartile.

Outliers are extreme deviations from the mean. They are values that are not "normal". To calculate outliers:

- Use a **z-score** to measure the distance from the mean in units of standard deviation. $z_i = \frac{x_i - \bar{x}}{s_x}$. z-scores above 3 are suspected outliers.
- Calculate $Q_1 - 1.5(IQR)$ and $Q_3 + 1.5(IQR)$, where Q_1 is the first quartile, Q_3 is the third quartile, and IQR is the interquartile range. If x_i is less than $Q_1 - 1.5(IQR)$ or greater than $Q_3 + 1.5(IQR)$, then it is considered an outlier.

A **box plot** is a graph that shows the five point summary, outliers (if any), and the distribution of data.

To determine if the data is skewed, calculate the **Pearson's Coefficient of Skew**. $Sk = \frac{3(\bar{x} - Median)}{s_x}$. The distribution is skewed to the left if $Sk < 0$, skewed to the right is $Sk > 0$, and symmetric if $Sk = 0$.

Useful R Functions

The `quantile()` function returns the five point summary when no arguments are specified. For a specific quantile, specify the *probs* argument.

The `boxplot()` command returns a box plot for a vector of values.

5.2 Exercises

The following exercises will help you practice other statistical measures. In particular, the exercises work on:

- Constructing a five point summary and a boxplot.
- Applying Chebyshev's Theorem.
- Identifying skewness.
- Identifying outliers.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results using R functions when possible.

1. Use the following observations to calculate the minimum, the first, second and third quartiles, and the maximum. Are there any outliers? Find the IQR to answer the question.

3	10	4	1	0	30	6

2. Confirm your finding of an outlier by calculating the z-score. Is 30 an outlier when using a z-Score?

3. Use Chebyshev's theorem to determine what percent of the data falls between the z-score found in 2.

Exercise 2

You will need the **Stocks** data set to answer this question. You can find this data at https://jagelves.github.io/Data/Stocks.csv The data is a sample of daily stock prices for ticker symbols TSLA (Tesla), VTI (S&P 500) and GBTC (Bitcoin).

1. Construct a boxplot for Stock A. Is the data skewed or symmetric?
2. Create a histogram of the data. Include a vertical line for the mean and median. Explain how the mean and median indicates a skew in the data. Calculate the skewness statistic to confirm your result.
3. Use a line chart to plot your data. Can you explain why the data has a skew?

Exercise 3

You will need the **mtcars** data set to answer this question. This data set is part of R. You don't need to download any files to access it.

1. Construct a boxplot for the *hp* variable. Write a command in R that retrieves the outlier. Which car is the outlier?
2. Create a histogram of the data. Is the data skewed? Include a vertical line for the mean and median. Calculate the skewness statistic to confirm your result.
3. Transform the data by taking a natural logarithm. Specifically, create a new variable called *Loghp*. Repeat the procedure in 2. Is the skew still there?

5.3 Answers

Exercise 1

1. The minimum is 0, the first quartile is 2, second quartile is 4, third quartile is 8, and maximum is 30. 30 is an outlier since it is beyond $Q_3 + 1.5 * IQR$.

Quartiles are calculated using the percentile formula $(n+1)P/100$. The data set has seven numbers. The first quartile's location is $8/4 = 2$, the second quartile's location is $8/2 = 4$ and the third quartile's location is $24/4 = 6$. The values at these location, when data is organized in ascending order, are 1, 4, and 10.

In R we can get the five number summary by using the `quantile()` function. Since there are various rules that can be used to calculate percentiles, we specify type 6 to match our rules.

```
Ex1<-c(3,10,4,1,0,30,6)
quantile(Ex1,type = 6)
```

```
  0%  25%  50%  75% 100% 
   0    1    4   10   30 
```

The interquartile range is needed to determine if there are any outliers. The IQR for this data set is $Q_3 - Q_1 = 9$. This reveals that 30 is and outlier, since $10 + 1.5 * 9 = 23.5$. Everything beyond 23.5 is an outlier.

2. If we use the z-score instead we find that 30 is not an outlier since the z-score is $Z_{30} = 2.15$. This observation is only 2.15 standard deviations away from the mean.

In R we can make a quick calculation of the z-Score to confirm our results. The z-score is given by $Z_i = \frac{x_{30} - \mu}{\sigma}$.

```
(Z30<-(30-mean(Ex1))/sd(Ex1))
```

```
[1] 2.148711
```

3. Chebyshev's theorem states that $1 - \frac{1}{z_2}$ of the data lies between z standard deviation from the mean.

Substituting the z-score found in 2. we get 78.34%. In R:

```
1-1/(Z30)^2
```

```
[1] 0.7834073
```

Exercise 2

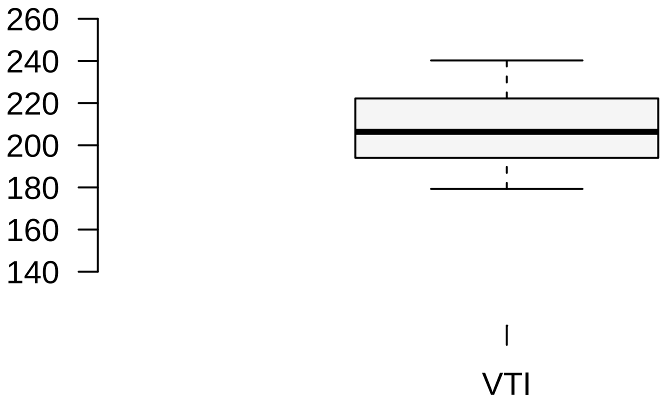
1. The data is skewed to the right.

Start by loading the data set:

```
StockPrices<-read.csv("https://jagelves.github.io/Data/Stocks.csv")
```

To construct the boxplot in R, use the `boxplot()` command.

```
boxplot(as.numeric(StockPrices$VTI),axes=F, ylim=c(120,260),
        cex=1.5, col="#F5F5F5",pch=21,bg="red")
axis(side=1, labels=c("VTI"), at=seq(1))
axis(side=2, labels=TRUE, at=seq(140,260,20),font=1,las=1)
```



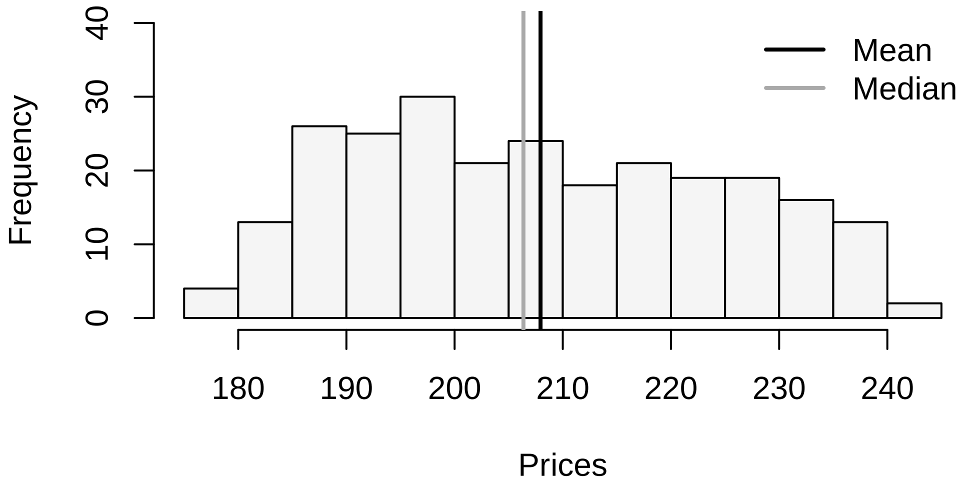
The boxplot shows that there are no outliers. The data also looks like it has a slight skew to the right.

2. The mean is more sensitive to outliers than the median. Hence, when the data is skewed to the right we expect that the mean is larger than the median.

Let's construct a histogram in R to search for skewness.

```
hist(StockPrices$VTI,main="", ylim=c(0,40),
     xlab="Prices", col="#F5F5F5")
abline(v=mean(StockPrices$VTI),col="black",lwd=2)
abline(v=median(StockPrices$VTI),col="darkgrey",lwd=2)
legend(x = "topright",
       legend = c("Mean", "Median"),
       lty = c(1, 1),
       col = c("black", "darkgrey"),
```

```
       lwd = 2,
       bty="n")
```



The lines are close to each other but the mean is slighlty larger than the median. Let's confirm with the skewness statistic $3(mean - median)/sd$.

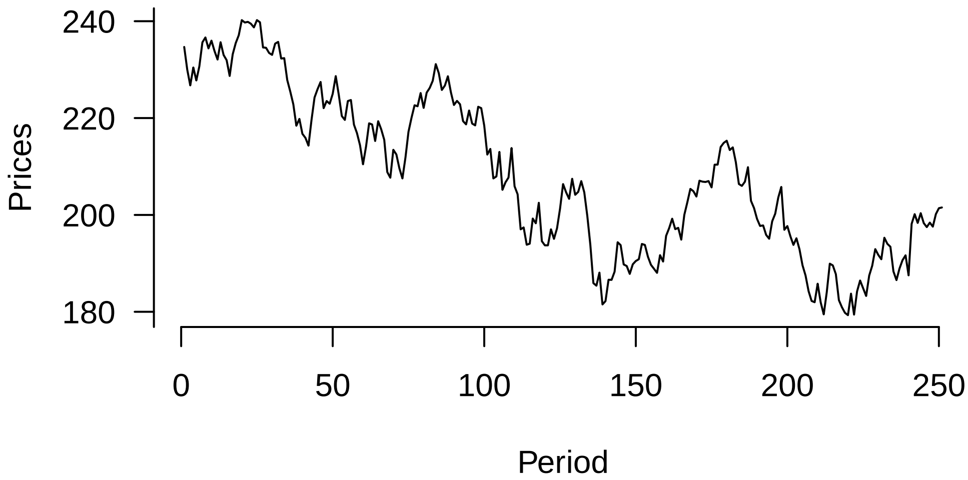
```
(skew<-3*(mean(StockPrices$VTI-median(StockPrices$VTI))/sd(StockPrices$VTI)))
```

```
[1] 0.2856304
```

This indicates that there is a slight skew to the right of the data.

3. The line chart indicates that the data has a downward trend in the early periods. This creates a few points that are large. In later periods the stock price stabilizes to lower levels.

```
plot(y=StockPrices$VTI,x=seq(1,length(StockPrices$VTI)),
     type="l", ylab="Prices", xlab="Period", axes=F)
axis(side=1, labels=TRUE, at=seq(0,250,50),font=1,las=1)
axis(side=2, labels=TRUE, at=seq(0,300,20),font=1,las=1)
```

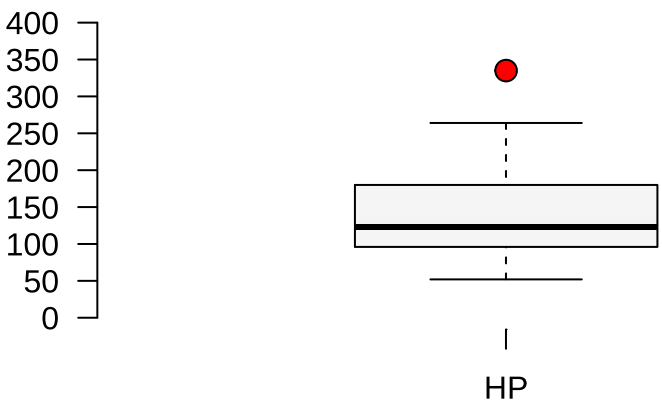


Exercise 3

1. The outlier is the Masserati Bora. The horse power is 335.

In R we can construct a boxplot with the following command:

```
boxplot(mtcars$hp,axes=F, ylim=c(0,400),
        cex=1.5, col="#F5F5F5",pch=21,bg="red")
axis(side=1, labels=c("HP"), at=seq(1))
axis(side=2, labels=TRUE, at=seq(0,400,50),font=1,las=1)
```



From the graph it seems like the outlier is beyond a horsepower of 275. Let's write an R command to retrieve the car.

```
mtcars[mtcars$hp>275,]
```

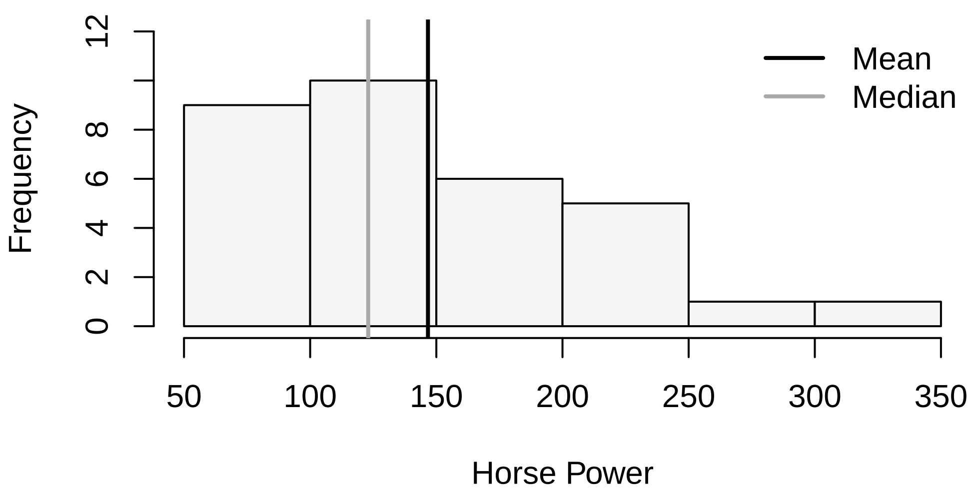
```
              mpg cyl disp  hp drat   wt qsec vs am gear carb
Maserati Bora  15   8  301 335 3.54 3.57 14.6  0  1    5    8
```

It's the Masserati Bora!

2. The histogram looks skewed to the right. This is confirmed by the estimation of a Pearson coefficient fo skewness of 1.04.

In R we can construct a histogram with vertical lines for the mean and median wit the following code:

```
hist(mtcars$hp,main="", ylim=c(0,12), xlab="Horse Power",
     col="#F5F5F5")
abline(v=mean(mtcars$hp),col="black",lwd=2)
abline(v=median(mtcars$hp),col="darkgrey",lwd=2)
legend(x = "topright",
       legend = c("Mean", "Median"),
       lty = c(1, 1),
       col = c("black", "darkgrey"),
       lwd = 2,
       bty="n")
```



The histogram looks skewed to the right. Pearson's Coefficient of Skewness is:

```
(SkewHP<-3*(mean(mtcars$hp)-median(mtcars$hp))/sd(mtcars$hp))
```

```
[1] 1.036458
```

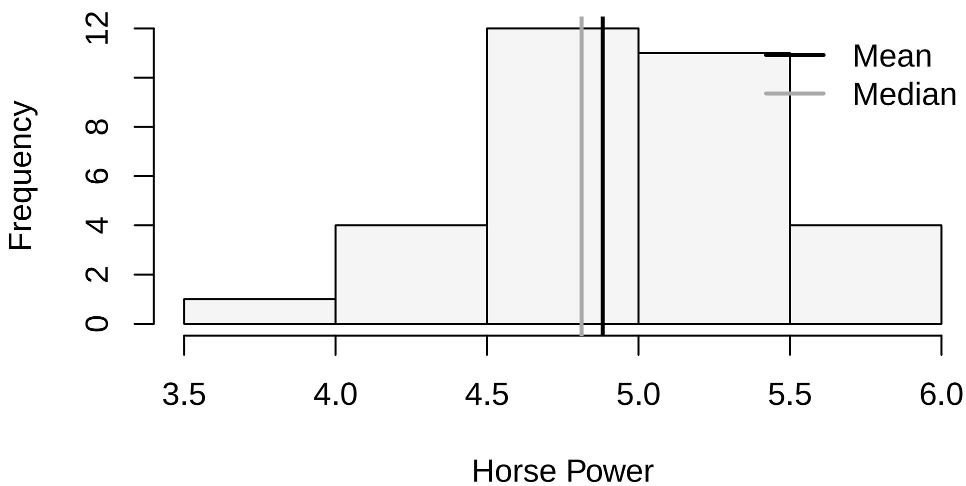
3. The skew is still there, but the distribution now look more symmetrical and the Skew coefficient has decreased to 0.44.

In R we can create an new variable that captures the log transformation. The `log()` function takes the natural logarithm of a number or vector.

```
LogHP<-log(mtcars$hp)
```

Let's use this new variable to create our histogram:

```
hist(LogHP,main="", ylim=c(0,12), xlab="Horse Power",
     col="#F5F5F5")
abline(v=mean(LogHP),col="black",lwd=2)
abline(v=median(LogHP),col="darkgrey",lwd=2)
legend(x = "topright",
       legend = c("Mean", "Median"),
       lty = c(1, 1),
       col = c("black", "darkgrey"),
       lwd = 2,
       bty="n")
```



The mean and the variance now look closer together. The tail of the distribution (skew) now also looks diminished. The Skewness coefficient has decreased significantly:

```
(SkewLogHP<-3*(mean(LogHP)-median(LogHP))/sd(LogHP))
```

```
[1] 0.4402212
```

6 Regression I

6.1 Concepts

Measures of Association

Measures of association determine whether there is a linear relationship between two variables. They also determine the strength of the relationship.

- The **covariance** is a measure that determines the direction of the relationship between two variables. It is calculated by $s_{xy} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2}$. If $s_{xy} > 0$ there is a direct relationship, if $s_{xy} < 0$ there is an inverse relationship, and if $s_{xy} = 0$ there is no relationship.
- The **correlation** measures the strength of the linear relationship. It is calculated by $r = \frac{s_{xy}}{s_x s_y}$. The correlation coefficient is between $[-1, 1]$. When the correlation coefficient is 1 (-1), there is a perfect direct (inverse) relationship between the two variables.
- The **coefficient of determination** or R^2, measures the percent of variation in y explained by variations in x. It is calculated by $R^2 = r^2$. The next chapter expands on this measure.
- A **scatter plot** displays pairs of [x,y] as points on the Cartesian plane. The plot provides a visual aid to determine the relationship between two variables.

Useful R Functions

To calculate the covariance use the `cov()` function.

The correlation coefficient can be calculated using the `cor()` function.

The `plot()` function will create scatter plots.

6.2 Exercises

The following exercises will help you understand statistical measures that establish the relationship between two variables. In particular, the exercises work on:

- Calculating covariance and correlation.
- Using R to plot scatter diagrams.
- Calculating the coefficient of determination.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results using R functions when possible.

1. Consider the data below. Calculate the covariance and correlation coefficient by finding deviations from the mean. Use R to verify your result. Is there a direct or inverse relationship between the two variables? How strong is the relationship?

x	20	21	15	18	25
y	17	19	12	13	22

2. Consider the data below. Calculate the covariance and correlation coefficient by finding deviations from the mean. Use R to verify your result. Is there a direct or inverse relationship between the two variables? How strong is the relationship?

w	19	16	14	11	18
z	17	20	20	16	18

Exercise 2

You will need the **mtcars** data set to answer this question. This data set is part of R. You don't need to download any files to access it.

1. Calculate the correlation coefficient between *hp* and *mpg*. Explain the results. Specifically, the direction of the relationship and the strength given the context of the problem.

2. Create a scatter diagram of the two variables. Is the scatter diagram what you expected after you calculated the correlation coefficient?
3. Calculate the coefficient of determination. How close is it to one? What else could be explaining the variation in the *mpg*? Let your dependent variable be *mpg*.

Exercise 3

You will need the **College** data set to answer this question. You can find this data set here: https://jagelves.github.io/Data/College.csv

1. Create a scatter diagram between *GRAD_DEBT_MDN* (Median Debt) and *MD_EARN_WNE_P10* (Median Earnings). What type of relationship do you observe between the variables?
2. Calculate the correlation coefficient and the coefficient of determination. According to the data, are higher debts correlated with higher earnings?

6.3 Answers

Exercise 1

1. The covariance is 14.9 and the correlation is 0.96. The results indicate that there is a strong direct relationship between the two variables.

Let's start by finding the deviations from the mean for the x variable in R.

```
x<-c(20,21,15,18,25)
(devx<-x-mean(x))
```

```
[1]  0.2  1.2 -4.8 -1.8  5.2
```

We will do the same with y:

```
y<-c(17,19,12,13,22)
(devy<-y-mean(y))
```

```
[1]  0.4  2.4 -4.6 -3.6  5.4
```

Note that when the deviations in x are negative (positive), they are also negative (positive) in y. This is indicative of a direct relationship between the two variables. The covariance is given by:

```
(Ex1Cov<-sum(devx*devy)/(length(devx)-1))
```

```
[1] 14.9
```

We can verify this by using `cov()` function in R.

```
cov(x,y)
```

```
[1] 14.9
```

The correlation coefficient is found by dividing the covariance over the product of standard deviations. In R:

```
(Ex1Cor<-Ex1Cov/(sd(x)*sd(y)))
```

```
[1] 0.9678386
```

We can once more verify the result in R with the built in function `cor()`.

```
cor(x,y)
```

```
[1] 0.9678386
```

2. The covariance is 0.85 and the correlation is 0.148. The results indicate that there is a very weak direct relationship between the two variables. They might be unrelated.

Let's start with w and finding the deviations from the mean in R.

```
w<-c(19,16,14,11,18)
(devw<-w-mean(w))
```

```
[1]  3.4  0.4 -1.6 -4.6  2.4
```

We will do the same with z:

```
z<-c(17,20,20,16,18)
(devz<-z-mean(z))
```

```
[1] -1.2  1.8  1.8 -2.2 -0.2
```

The covariance is given by:

```
(Ex2Cov<-sum(devw*devz)/(length(devz)-1))
```

```
[1] 0.85
```

We can verify this with the `cov()` function in R.

```
cov(w,z)
```

```
[1] 0.85
```

The correlation coefficient is found by dividing the covariance over the product of standard deviations. In R:

```
(Ex2Cor<-Ex2Cov/(sd(z)*sd(w)))
```

```
[1] 0.1480558
```

We can once more verify the result in R with the built in function `cor()`.

```
cor(w,z)
```

```
[1] 0.1480558
```

Exercise 2

1. The correlation coefficient is -0.78. This is indicative of a moderately strong inverse relationship between *mpg* and *mp*.

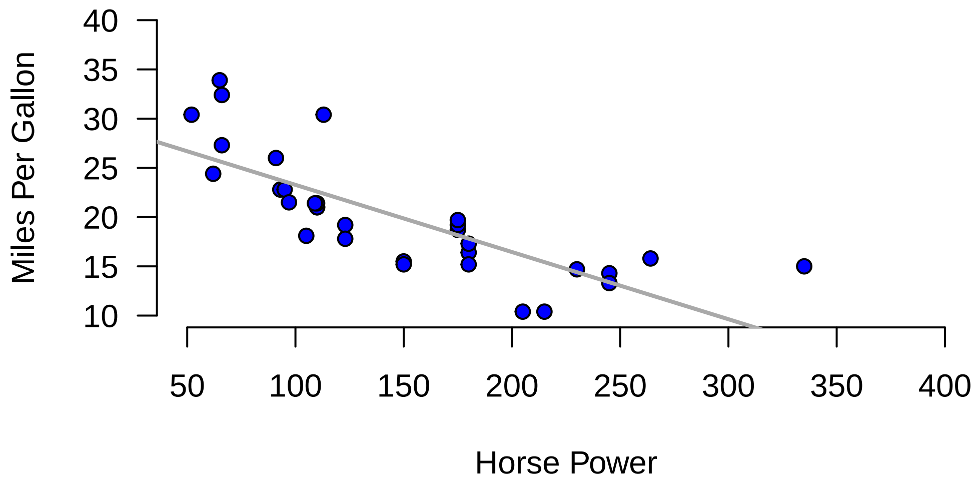
In R we can easily calculate the correlation coefficient with the `cor()` function.

```
cor(mtcars$mpg,mtcars$hp)
```

```
[1] -0.7761684
```

2. The scatter diagram is downward sloping. Most points are close to the trend line. It is what was expected from a correlation coefficient of -0.78.

```
plot(y=mtcars$mpg,x=mtcars$hp, main="",
     axes=F,pch=21, bg="blue",
     xlab="Horse Power",
     ylab="Miles Per Gallon", ylim=c(10,40),xlim=c(50,400))
axis(side=1, labels=TRUE, font=1,las=1)
axis(side=2, labels=TRUE, font=1,las=1)
abline(lm(mtcars$mpg~mtcars$hp),
       col="darkgray",lwd=2)
```



3. The coefficient of determination is 0.6. This value is not very close to one. This is expected since miles per gallon can also vary because of the cars weight, and fuel efficiency. It makes sense that the *hp* only explains 60% of the total variation.

In R we can calculate the coefficient of determination by squaring the correlation coefficient.

```
cor(mtcars$mpg,mtcars$hp)^2
```

```
[1] 0.6024373
```

Exercise 3

1. It seems like there is a direct relationship between both variables. The more debt you take, the higher the salary.

Start by loading the data. We'll use the `read.csv()` function:

```
College<-read.csv("https://jagelves.github.io/Data/College.csv")
```

The two variables of interest are *GRAD_DEBT_MDN* and *MD_EARN_WNE_P10*. The following code creates the scatter plot:

2. The correlation coefficient shows a moderate direct relationship between earnings and debt 0.43. The coefficient of determination indicates that only 19% of the variation in earnings can be explained by debt.

In R we can start with the correlation coefficient:

The coefficient of determination is:

7 Regression II

7.1 Concepts

The Regression Line

The regression line is fitted so that the average distance between the line and the sample points is as small as possible. The line is defined by a **slope** (β) and an **intercept** (α). Mathematically, the regression line is expressed as $\hat{y}_i = \hat{\alpha} + \hat{\beta}x_i$, where $\hat{y}_i$ are the predicted values of y given the x's.

- The **slope** determines the steepness of the line. The estimate quantifies how much a unit increase in x changes y. The estimate is given by $\hat{\beta} = \frac{s_{xy}}{s_x^2}$.
- The **intercept** determines where the line crosses the y axis. It returns the value of y when x is zero. The estimate is given by $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$.

Goodness of Fit

There are a couple of popular measures that determine the goodness of fit of the regression line.

- The **coefficient of determination** or R^2 is the percent of the variation in y that is explained by changes in x. The higher the R^2 the better the explanatory power of the model. The R^2 is always between [0,1]. To calculate use $R^2 = SSR/SST$.
  - SSR (Sum of Squares due to Regression) is the part of the variation in y explained by the model. Mathematically, $SSR = \sum(\hat{y}_i - \bar{y})^2$.
  - SSE (Sum of Squares due to Error) is the part of the variation in y that is unexplained by the model. Mathematically, $SSE = \sum(y_i - \hat{y}_i)^2$.
  - SST (Sum of Squares Total) is the total variation of y with respect to the mean. Mathematically, $SST = \sum(y_i - \bar{y})^2$.
  - Note that $SST = SSR + SSE$.

- The **adjusted** R^2 recognizes that the R^2 is a non-decreasing function of the number of explanatory variables in the model. This metric penalizes a model with more explanatory variables relative to a simpler model. It is calculated by $1-(1-R^2)\frac{n-1}{n-k-1}$, where k is the number of explanatory variables used in the model and n is the sample size.
- The **Residual Standard Error** estimates the average dispersion of the data points around the regression line. It is calculated by $s_e = \sqrt{\frac{SSE}{n-k-1}}$.

Useful R Functions

The `lm()` function to estimates the linear regression model.

The `predict()` function uses the linear model object to predict values. New data is entered as a data frame.

The `coef()` function returns the model's coefficients.

The `summary()` function returns the model's coefficients, and goodness of fit measures.

7.2 Exercises

The following exercises will help you get practice on Regression Line estimation and interpretation. In particular, the exercises work on:

- Estimating the slope and intercept.
- Calculating measures of goodness of fit.
- Prediction using the regression line.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results using R functions when possible.

1. Consider the data below. Calculate the deviations from the mean for each variable and use the results to estimate the regression line. Use R to verify your result. On average by how much does y increase per unit increase of x?

x	20	21	15	18	25
y	17	19	12	13	22

2. Calculate *SST*, *SSR*, and *SSE*. Confirm your results in R. What is the R^2? What is the Standard Error estimate? Is the regression line a good fit for the data?
3. Assume that x is observed to be *32*, what is your prediction of y? How confident are you in this prediction?

Exercise 2

You will need the **Education** data set to answer this question. You can find the data set at https://jagelves.github.io/Data/Education.csv . The data shows the years of education (*Education*), and annual salary in thousands (*Salary*) for a sample of 100 people.

1. Estimate the regression line using R. By how much does an extra year of education increase the annual salary on average? What is the salary of someone without any education?
2. Confirm that the regression line is a good fit for the data. What is the estimated salary of a person with 16 years of education?

Exercise 3

You will need the **FoodSpend** data set to answer this question. You can find this data set at https://jagelves.github.io/Data/FoodSpend.csv .

1. Omit any NA's that the data has. Create a dummy variable that is equal to 1 if an individual owns a home and 0 if the individual doesn't. Find the mean of your dummy variable. What proportion of the sample owns a home?
2. Run a regression with *Food* being the dependent variable and your dummy variable as the independent variable. What is the interpretation of the intercept and slope?
3. Now run a regression with *Food* being the independent variable and your dummy variable as the dependent variable. What is the interpretation of the intercept and slope? Hint: you might want to plot the scatter diagram and the regression line.

Exercise 4

You will need the **Population** data set to answer this question. You can find this data set at https://jagelves.github.io/Data/Population.csv .

1. Run a regression of *Population* on *Year*. How well does the regression line fit the data?
2. Create a prediction for Japan's population in 2030. What is your prediction?
3. Create a scatter diagram and include the regression line. How confident are you of your prediction after looking at the diagram?

7.3 Answers

Exercise 1

1. The regression lines is $\hat{y} = -4.93 + 1.09x$. For each unit increase in x, y increases on average 1.09.

Start by generating the deviations from the mean for each variable. For x the deviations are:

```
x<-c(20,21,15,18,25)
(devx<-x-mean(x))
```

```
[1]  0.2  1.2 -4.8 -1.8  5.2
```

Next, find the deviations for y:

```
y<-c(17,19,12,13,22)
(devy<-y-mean(y))
```

```
[1]  0.4  2.4 -4.6 -3.6  5.4
```

For the slope we need to find the deviation squared of the x's. This can easily be done in R:

```
(devx2<-devx^2)
```

```
[1]  0.04  1.44 23.04  3.24 27.04
```

The slope is calculated by $\frac{\sum_{i=i}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=i}^{n}(x_i-\bar{x})^2}$. In R we can just find the ratio between the summations of *(devx)(devy)* and *devx2*.

```
(slope<-sum(devx*devy)/sum(devx2))
```

```
[1] 1.087591
```

The intercept is given by $\bar{y}-\beta(\bar{x})$. In R we find that the intercept is equal to:

```
(intercept<-mean(y)-slope*mean(x))
```

```
[1] -4.934307
```

Our results can be easily verified by using the `lm()` and `coef()` functions in R.

```
fitEx1<-lm(y~x)
coef(fitEx1)
```

```
(Intercept)           x
  -4.934307    1.087591
```

2. SST is 69.2, SSR is 64.82 and SSE is 4.38 (note that $SSR+SSE=SST$). The R^2 is just $\frac{SSR}{SST}=0.94$ and the Standard Error estimate is 1.21. They both indicate a great fit of the regression line to the data.

Let's start by calculating the SST. This is just $\sum(y_i-\bar{y})^2$.

```
(SST<-sum((y-mean(y))^2))
```

```
[1] 69.2
```

Next, we can calculate SSR. This is calculated by the following formula $\sum(\hat{y}_i-\bar{y})^2$. To obtain the predicted values in R, we can use the output of the `lm()` function. Recall our *fitEx1* object created in Exercise 1. It has *fitted.values* included:

```
(SSR<-sum((fitEx1$fitted.values-mean(y))^2))
```

```
[1] 64.82044
```

The ratio of SSR to SST is the R^2:

```r
(R2<-SSR/SST)
```

```
[1] 0.9367115
```

Finally, let's calculate $SSE \sum(y_i - \hat{y}_i)^2$:

```r
(SSE<-sum((y-fitEx1$fitted.values)^2))
```

```
[1] 4.379562
```

With the SSE we can calculate the Standard Error estimate:

```r
sqrt(SSE/3)
```

```
[1] 1.208244
```

We can confirm these results using the `summary()` function.

```r
summary(fitEx1)
```

```

Call:
lm(formula = y ~ x)

Residuals:
      1       2       3       4       5 
 0.1825  1.0949  0.6204 -1.6423 -0.2555 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -4.9343     3.2766  -1.506  0.22916   
x             1.0876     0.1632   6.663  0.00689 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.208 on 3 degrees of freedom
Multiple R-squared:  0.9367,	Adjusted R-squared:  0.9156 
F-statistic:  44.4 on 1 and 3 DF,  p-value: 0.00689
```

3. If $x = 32$ then $\hat{y} = 29.87$. The regression is a good fit, so we can feel good about our prediction. However, we would be concerned about the sample size of the data.

In R we can obtain a prediction by using the `predict()` function. This function requires a data frame as an input for new data.

```
predict(fitEx1, newdata = data.frame(x=c(32)))
```

```
       1 
29.86861 
```

Exercise 2

1. An extra year of education increases the annual salary about 5,300 dollars (slope). A person that has no education would be expected to earn 17,2582 dollars (intercept).

Start by loading the data in R:

```
Education<-read.csv("https://jagelves.github.io/Data/Education.csv")
```

Next, let's use the `lm()` function to estimate the regression line and obtain the coefficients:

```
fitEducation<-lm(Salary~Education, data = Education)
coefficients(fitEducation)
```

```
(Intercept)   Education 
  17.258190    5.301149 
```

2. The R^2 is 0.668 and the standard error is 21. The line is a moderately good fit. If someone has 16 years of experience, the regression line would predict a salary of 102,000 dollars.

Let's get the R^2 and the Standard Error estimate by using the `summary()` function and *fitEx1* object.

```
summary(fitEducation)
```

```
Call:
lm(formula = Salary ~ Education, data = Education)

Residuals:
    Min      1Q  Median      3Q     Max 
-62.177  -9.548   1.988  15.330  45.444 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  17.2582     4.0768   4.233  5.2e-05 ***
Education     5.3011     0.3751  14.134  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.98 on 98 degrees of freedom
Multiple R-squared:  0.6709,    Adjusted R-squared:  0.6675 
F-statistic: 199.8 on 1 and 98 DF,  p-value: < 2.2e-16
```

Lastly, let's use the regression line to predict the salary for someone who has 16 years of education.

```
predict(fitEducation, newdata = data.frame(Education=c(16)))
```

```
       1 
102.0766 
```

Exercise 3

1. Approximately, 36% of the sample owns a home.

Start by loading the data into R and removing all NA's:

```
Spend<-read.csv("https://jagelves.github.io/Data/FoodSpend.csv")
Spend<-na.omit(Spend)
```

To create a dummy variable for *OwnHome* we can use the `ifelse()` function:

```
Spend$dummyOH<-ifelse(Spend$OwnHome=="Yes",1,0)
```

The average of the dummy variable is given by:

```r
mean(Spend$dummyOH)
```

```
[1] 0.3625
```

2. The intercept is the average food expenditure of individuals without homes (6417). The slope, is the difference in food expenditures between individuals that do have homes minus those who don't. We then conclude that individuals that do have a home spend about -2516 less on food than those who don't have homes.

To run the regression use the `lm()` function:

```r
lm(Food~dummyOH,data=Spend)
```

```
Call:
lm(formula = Food ~ dummyOH, data = Spend)

Coefficients:
(Intercept)      dummyOH
       6473        -3418
```

3. The scatter plot shows that most of the points for home owners are below 6000. For non-home owners they are mainly above 6000. The line can be used to predict the likelihood of owning a home given someones food expenditure. The intercept is above one, but still it gives us the indication that it is likely that low food expenditures are highly predictive of owning a home. The slope tells us how that likelihood changes as the food expenditures increase by 1. In general, the likelihood of owning a home decreases as the food expenditure increases.

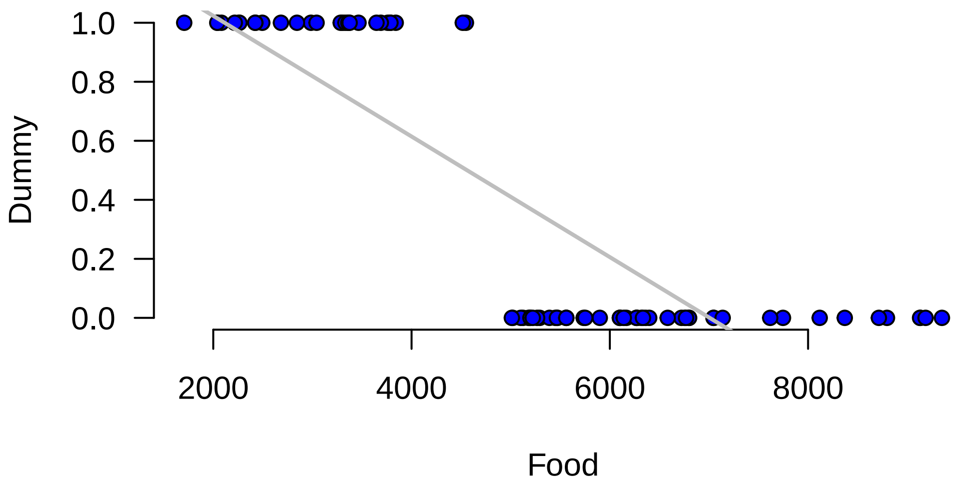
Run the `lm()` function once again:

```r
fitFood<-lm(dummyOH~Food,data=Spend)
coefficients(fitFood)
```

```
  (Intercept)          Food
 1.4320766616 -0.0002043632
```

For the scatter plot use the following code:

```
plot(y=Spend$dummyOH,x=Spend$Food,
     main="", axes=F, pch=21, bg="blue",
     xlab="Food",ylab="Dummy")
axis(side=1, labels=TRUE, font=1,las=1)
axis(side=2, labels=TRUE, font=1,las=1)
abline(fitFood,
       col="gray",lwd=2)
```



Exercise 4

1. If we follow the $R^2 = 0.81$ the model fits the data very well.

Let's load the data from the web:

```
Population<-read.csv("https://jagelves.github.io/Data/Population.csv")
```

Now let's filter the data so that we can focus on the population for Japan.

```
library(dplyr)
```

```
Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
Japan<-filter(Population,Country.Name=="Japan")
```

Next, we can run the regression of *Population* against the *Year*. Let's also run the `summary()` function to obtain the fit and the coefficients.

```
fit<-lm(Population~Year,data=Japan)
summary(fit)
```

```
Call:
lm(formula = Population ~ Year, data = Japan)

Residuals:
     Min       1Q   Median       3Q      Max 
-9583497 -4625571  1214644  4376784  5706004 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -988297581   68811582  -14.36   <2e-16 ***
Year            555944      34569   16.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4871000 on 60 degrees of freedom
Multiple R-squared:  0.8117,    Adjusted R-squared:  0.8086 
F-statistic: 258.6 on 1 and 60 DF,  p-value: < 2.2e-16
```

2. The prediction for 2030 is about 140 million people.

Let's use the `predict()` function:

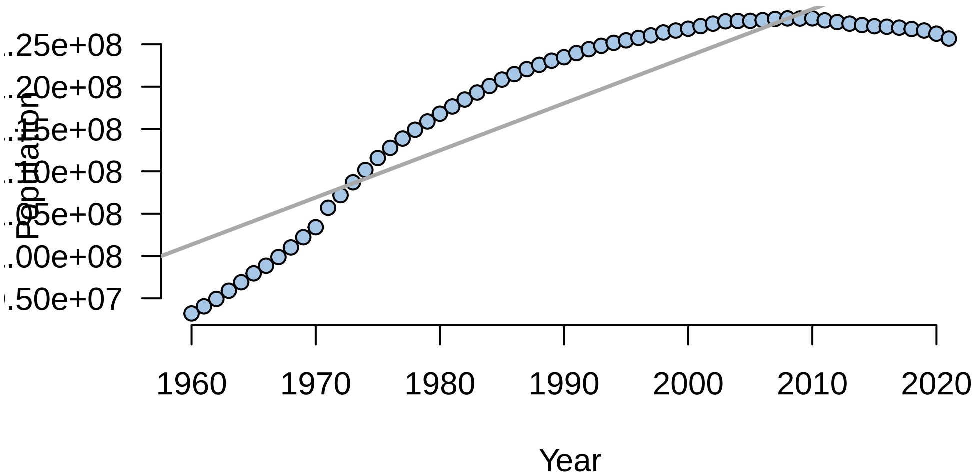
```
predict(fit,newdata=data.frame(Year=c(2030)))
```

```
        1 
140268585 
```

3. After looking at the scatter plot, it seems unlikely that the population in Japan will hit 140 million. Population has been decreasing in Japan!

Use the `plot()` and `abline()` functions to create the figure.

```
plot(y=Japan$Population,x=Japan$Year, main="",
     axes=F,pch=21, bg="#A7C7E7",
     xlab="Year",
     ylab="Population")
axis(side=1, labels=TRUE, font=1,las=1)
axis(side=2, labels=TRUE, font=1,las=1)
abline(fit,
       col="darkgray",lwd=2)
```



8 Probability I

8.1 Concepts

Frequentist Vs. Bayesian

The **frequentist** interpretation assumes that probabilities represent proportions of specific events occurring over infinitely identical trials.

The **Bayesian** interpretation assumes that probabilities are subjective beliefs about the relative likelihood of events.

Experiments and Sets

An **experiment** is a process that leads to one of several outcomes. Ex: Tossing a Die, Tossing a Coin, Drawing a Card, etc.

An **outcome** is the result of an experiment. Ex: A coin landing on heads, drawing the ace of spades.

The **sample space** (S) of an experiment contains all possible outcomes of the experiment. Ex: S={1,2,3,4,5,6} is the sample space for tossing a die.

An **event** is a subset of the sample space. A={2,4,6} is the event of tossing an even number when rolling a die.

Basic Probability Concepts

A **probability** is a numerical value that measures the likelihood that an event occurs.

To calculate **probabilities**, find the ratio between favorable outcomes and total outcomes. $p = favorable/total$.

- The probability of any event A is a value between 0 and 1 inclusive. Formally, $0 \leq P(A) \leq 1$.
- When the probability of the event is 0 then the event is impossible. When the probability is 1 then the event is certain.

- The sum of the probabilities of a list of **mutually exclusive** and **exhaustive** events equals 1. Formally, $\sum P(x_i) = 1$.
  - **Mutually exclusive** events do not share any common outcomes. The occurrence of one event precludes the occurrence of others.
  - **Exhaustive** events include all outcomes in the sample space.

To assign probabilities you can use the Empirical, Classical, or Subjective Methods.

- **Empirical**: calculated as a relative frequency of occurrence.
- **Classical**: based on logical analysis.
- **Subjective**: calculated by drawing on personal and subjective judgement.

Probability Rules

The **Complement Rule**: $P(A^c) = 1 - P(A)$, where A^c is the complement of A.

The **Addition Rule**: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, where $\cap$ is intersection and $\cup$ is union.

The **Multiplication Rule**:

- if events are dependent $P(A \cap B) = P(A|B)P(B)$, where $P(A|B)$ is the conditional probability.
- if events are independent $P(A \cap B) = P(A)P(B)$.

The **Law of Total Probability**: $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$.

Bayes' Theorem: $P(A|B) = P(B|A)P(A)/P(B)$.

Counting Rules

The **Combination** function counts the number of ways to choose x objects from a total of n objects. The order in which the x objects are listed does not matter.

- If repetition is not allowed use $C_n^x = \frac{n!}{(n-x)!x!}$.
- If repetition is allowed use $\frac{(x+n-1)!}{(n-1)!x!}$.

The **Permutation** function also counts the number of ways to choose x objects from a total of n objects. However, the order in which the x objects are listed does matter.

- If repetition is not allowed use $P_n^x = \frac{n!}{(n-x)!}$.

- If repetition is allowed use n^x.

Useful R Functions

The `table()` function can be used to construct frequency distributions.

The `factorial()` function returns the factorial of a number.

The `gtools` package contains the `combinations()` and `permutations()` functions used to calculate combinations and permutations. Use the *repeats.allowed* argument to specify counting with repetition or no repetition. The v argument allows you to specify a vector of elements.

8.2 Exercises

The following exercises will help you practice some probability concepts and formulas. In particular, the exercises work on:

- Calculating simple probabilities.
- Applying probability rules.
- Using counting rules.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results with a calculator or R.

1. A sample space S yields five equally likely events, A, B, C, D, and E. Find $P(D)$, $P(B^c)$, and $P(A \cup C \cup E)$.
2. Consider the roll of a die. Define A as {1,2,3}, B as {1,2,3,5,6}, C as {4,6}, and D as {4,5,6}. Are the events A and B mutually exclusive, exhaustive, both or none? What about events A and D?
3. A recent study suggests that 33.1% of the adult U.S. population is overweight and 35.7% obese. What is the probability that a randomly selected adult in the U.S. is either obese or overweight? What is the probability that their weight is normal? Are the events mutually exclusive and exhaustive?

Exercise 2

For the following exercises, make your calculations by hand and verify results with a calculator or R.

1. Let $P(A) = 0.65$, $P(B) = 0.3$, and $P(A|B) = 0.45$. Calculate $P(A \cap B)$, $P(A \cup B)$, and $P(B|A)$.
2. Let $P(A) = 0.4$, $P(B) = 0.5$, and $P(A^c \cap B^c) = 0.24$. Calculate $P(A^c|B^c)$, $P(A^c \cup B^c)$, and $P(A \cup B)$.
3. Stock A will rise in price with a probability of 0.4, stock B will rise with a probability of 0.6. If stock B rises in price, then A will also rise with a probability of 0.5. What is the probability that at least one of the stocks will rise in price? Prove that events A and B are (are not) mutually exclusive (independent).

Exercise 3

1. Create a joint probability table from the contingency table below. Find $P(A)$, $P(A \cap B)$, $P(A|B)$, and $P(B|A^c)$. Determine whether the events are independent or mutually exclusive.

	B	B^c
A	26	34
A^c	14	26

Exercise 4

You will need the **Crash** data set and R to answer this question. The data shows information on several car crashes. Specifically, if the crash was Head-On or Not Head-On and whether there was Daylight or No Daylight. You can find the data here: https://jagelves.github.io/Data/Crash.csv

1. Create a contingency table.
2. Find the probability that a) a car crash is Head-On, b) a car crash is in daylight c) a car crash is Head-On given that there is daylight.
3. Show that Crashes and Light are dependent.

Exercise 5

1. Use Bayes' Theorem in the following question. Let $P(A) = 0.7$, $P(B|A) = 0.55$, and $P(B|A^c) = 0.10$. Find $P(A^c)$, $P(A \cap B)$, $P(A^c \cap B)$, $P(B)$, and $P(A|B)$.
2. Some find tutors helpful when taking a course. Julia has a 40% chance to fail a course if she does not have a tutor. With a tutor, the probability of failing is only 10%. There is a 50% chance that Julia finds an available tutor. What is the probability that Julia will fail the course? If she ends up failing the course, what is the probability that she had a tutor?

Exercise 6

1. Calculate the following values and verify your results using R. a) 3!, b) 4!, c) C_6^8, d) P_6^8.
2. There are 10 players in a local basketball team. If we chose 5 players to randomly start a game, in how many ways can we select the five players if order doesn't matter? What if order matters?

8.3 Answers

Exercise 1

1. $P(D) = 1/5 = 0.2$ since all events are equally likely. $P(B^c) = 4/5 = 0.8$, and $P(A \cup C \cup E) = P(A + C + E) = 3/5 = 0.6$.
2. Events A and B are not mutually exclusive since they share some of the same elements. They are not exhaustive since the union of both doesn't create the sample space.
3. The probability is 68.8%. The events are mutually exclusive. If someone is classified as obese, the person is not classified again as overweight. The events are not exhaustive since there are people in the U.S. that have a normal weight. The probability that the person drawn has normal weight is 31.2%.

Exercise 2

1. From the multiplication rule, $P(A|B) * P(B) = P(A \cap B)$.
   Substituting values yields, $P(A \cap B) = 0.45 * 0.3 = 0.135$.
   From the addition rule, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
   Substituting yields, $P(A \cup B) = 0.65 + 0.3 - 0.135 = 0.815$.
   From the multiplication rule once again, $P(B|A) = \frac{P(A \cap B)}{P(A)}$. Substituting yields, $P(B|A) = 0.135/0.65 = 0.2076923$.

2. From the complement rule we have that $P(A^c) = 0.6$ and $P(B^c) = 0.5$.
   Using the multiplication rule, $P(A^c|B^c) = \frac{P(A^c \cap B^c)}{P(B^c)}$. Substituting yields $P(A^c|B^c) = 0.24/0.5 = 0.48$.
   From the addition rule $P(A^c \cup B^c) = P(A^c) + P(B^c) - P(A^c \cap B^c)$.
   Substituting yields $P(A^c \cup B^c) = 0.6 + 0.5 - 0.24 = 0.86$.
   The event that has no elements of A or B is given by $P(A^c \cap B^c)$. Therefore $P(A \cup B) = 1 - 0.24 = 0.76$ has all the elements of A and B.

3. In short the problem states $P(A) = 0.4$, $P(B) = 0.6$, and $P(A|B) = 0.5$. Where A and B are events of stocks rising in price. The question asks for $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
   Using the multiplication rule $P(A \cap B) = 0.5 * 0.6 = 0.3$.
   Hence, $P(A \cup B) = 0.4 + 0.6 - 0.3 = 0.7$.
   The events are not mutually exclusive since $P(A \cap B) = 0.3 \neq 0$.
   The events are also not independent since $P(A|B) = 0.5 \neq 0.4 = P(A)$.

Exercise 3

1. Below is the joint probability table. The $P(A) = 0.26 + 0.34 = 0.6$, $P(A \cap B) = 0.26$, $P(A|B) = 0.26/0.4 = 0.65$, and $P(B|A^c) = 0.14/0.4 = 0.35$. Events A and B are not independent since $P(A) \neq P(A|B)$. The events are not mutually exclusive since $P(A \cap B) = 0.26 \neq 0$.

	B	B^c	Total
A	0.26	0.34	0.6
A^c	0.14	0.26	0.4
Total	0.4	0.6	1

Exercise 4

1. The probability of a Head-On crash is $(166 + 108)/4858 = 0.056$. The probability of a daylight crash is $(166 + 3258)/4858 = 0.70$. The probability that the car crash is Head-On given daylight is $166/(166 + 3258) = 0.048$.

Start by loading the data into R.

```
Crash<-read.csv("https://jagelves.github.io/Data/Crash.csv")
```

To create a contingency table use the `table()` command in R.

```
(freq<-table(Crash$Crash.Type,Crash$Light.Condition))
```

```
             Daylight Not Daylight
Head-on           166          108
Not Head-On      3258         1326
```

This table is used to calculate probabilities. We can pass it through the `prop.table()` function to get the contingency table.

```
round(prop.table(freq),2)
```

```
             Daylight Not Daylight
Head-on          0.03         0.02
Not Head-On      0.67         0.27
```

2. The two variables are dependent since $P(Head-On|Daylight) \neq P(Head-On)$, that is $0.048 \neq 0.56$.

Exercise 5

1. $P(A^c) = 1 - P(A) = 1 - 0.7 = 0.3$, $P(A \cap B) = P(B|A)P(A) = 0.55(0.70) = 0.385$, $P(A^c \cap B) = P(B|A^c)P(A^c) = 0.10(0.30) = 0.03$, $P(B) = P(A \cap B) + P(A^c \cap B) = 0.385 + 0.03 = 0.415$, and $P(A|B) = \frac{P(A \cap B)}{P(B)} = 0.385/0.415 = 0.9277$.
2. Let the event of failing be F, the event of not failing be NF, the event of having a tutor be T, and the event of not having a tutor be NT. The probability of failing the course is 0.25. $(F) = P(F \cap T) + P(F \cap T^c) = P(F|T)P(T) + P(F|T^c)P(T^c) = 0.10(0.50) + 0.40(0.50) = 0.05 + 0.20 = 0.25$ The probability of not having a tutor, given that she failed the course is 0.2. $P(T|F) = \frac{P(F \cap T)}{P(F \cap T) + P(F \cap T^c)} = 0.05/0.25 = 0.20$

Exercise 6

1. $3! = 3 \times 2 \times 1 = 6$, $4! = 6 \times 4 = 24$, $C^8_6 = 28$, and $P^8_6 = 20,160$

In R we can just use the factorial command. So 3! is:

```r
factorial(3)
```

```
[1] 6
```

and 4! is:

```r
factorial(4)
```

```
[1] 24
```

For combinations and permutations we can use the `gtools` package:

```r
library(gtools)
C<-combinations(8,6)
nrow(C)
```

```
[1] 28
```

2. If order doesn't matter, there are 252 ways. If order matters, then there are $30,240$ ways.

In R we can once more use the combination and permutation functions:

```r
B1<-combinations(10,5)
nrow(B1)
```

```
[1] 252
```

```r
B2<-permutations(10,5)
nrow(B2)
```

```
[1] 30240
```

9 Probability II

9.1 Concepts

Random Variables

A **random variable** associates a numerical value with each possible experimental outcome. Specifically, the random variable takes on a value with some probability.

A random variable is fully characterized by its **probability density function** (PDF) if continuous or the **probability mass function** (PMF) if discrete.

Expected Value and Variance

When summarizing a random variable, we are mostly interested in the variable's central tendency (Expected Value) and dispersion (Variance).

The **expected value** (mean) is a measure of central location. For a discrete random variable it is given by $E(x) = \mu = \sum xf(x)$, where $f(x)$ is the probability mass function. For a continuous random variable it is given by $E(x) = \int_{-\infty}^{\infty} xf(x)dx$, where $f(x)$ is the probability density function.

The **variance** summarizes the deviation of the values of the random variable from the mean. It is calculated by $var(x) = E[(x - E(x))^2] = E[x^2] - E[x]^2$. Note that this formula can be used for both discrete and continuous random variables.

Discrete Uniform Distribution

The **discrete uniform distribution** is a probability distribution that assigns equal probability to each outcome in a finite set of possible outcomes. In other words, each outcome in the set is equally likely to occur.

The **probability mass function** is given by $f(x) = 1/n$, where n is the number of elements in the sample space (all possible outcomes).

The **expected value** is given by $E(x) = \frac{\sum x_i}{n}$, where x_i are the possible values, and n is the number of possible values.

The **variance** is given by $var(x) = \frac{\sum(x_i - E(x))^2}{n-1}$.

Binomial Distribution

The binomial distribution is a probability distribution that describes the outcome of a sequence of n independent Bernoulli trials. In a Bernoulli trial, there are only two possible outcomes: "success" and "failure". The probability of success is denoted by p, and the probability of failure is denoted by $q = 1 - p$. In a sequence of n independent Bernoulli trials, the number of successes (x) is a random variable that follows a binomial distribution.

The **probability mass function** is given by $f(x) = C_x^n(p^x)(1-p)^{n-x}$, where n is the number of trials, x is the number of successes, p is the probability of success, and C_x^n is the number of ways there can be x successes in n trials.

The **expected value** of the binomial distribution is $E(x) = np$.

The **variance** of the binomial distribution is $var(x) = np(1 - p)$.

The Hypergeometric Distribution

The **hypergeometric distribution** is a probability distribution that describes the outcome of drawing a sample from a population without replacement. It is used to calculate the probability of drawing a certain number of successes (x) in a sample of a given size (n), where the success or failure of each individual draw is not dependent on the success or failure of other draws.

The **hypergeometric** experiment differs from the binomial since:

- trials are not independent.
- the probability of success changes from trial to trial.

The **probability mass function** is given by $f(x) = \frac{C_x^r C_{n-x}^{N-r}}{C_n^N}$, where n is the number of trials, x is the number of successes, r is the number of elements in the population labeled as success, and N is the number of elements in the population.

The **expected value** of the hypergeometric distribution is $E(x) = n\frac{r}{N}$.

The **variance** of the hypergeometric distribution is $var(x) = n\frac{r}{N}(1 - \frac{r}{N})(\frac{N-n}{N-1})$.

Poisson Distribution

The **Poisson distribution** estimates the number of successes (x) over a specified interval of time or space.

The **probability mass function** is given by $f(x) = \frac{\mu e^{-x}}{x!}$, where μ is the expected number of successes in any given interval and also the variance, and e is Euler's number (2.71828…).

An experiment satisfies a Poisson process if:

- The number of successes with a specified time or space interval equals any integer between zero and infinity.
- The number of successes counted in non-overlapping intervals are independent.
- The probability of success in any interval is the same for all intervals of equal size and is proportional to the size of the interval.

Useful R Functions

To calculate probabilities based on discrete random variables use the `pbinom()`, `phyper()`, and `ppois()` functions. For the uniform distribution use the `extraDistr` package and the `pdunif()` function.

To calculate cumulative probabilities use the `dbinom()`, `dhyper()`, `dpois()`, and `ddunif()` functions.

To calculate quantiles use the `qbinom()`, `qhyper()`, `qpois()`, and `qdunif()` functions.

To generate random numbers use the `rbinom()`, `rhyper()`, `rpois()`, and `rdunif()` functions.

9.2 Exercises

The following exercises will help you practice some probability concepts and formulas. In particular, the exercises work on:

- Calculating probabilities for discrete random variables.
- Calculating the expected value and standard deviation.
- Applying the binomial, Poisson and hypergeometric probability distributions.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results with a calculator or R.

1. Consider the table below. Calculate the mean and standard deviation. What is the probability that $x < 15$?

x	5	10	15	20
$P(X = x)$	0.35	0.3	0.2	0.15

2. Consider the table below. Calculate the mean and standard deviation. What is the probability that $x \geq -9$?

y	-23	-17	-9	-3
$P(Y = y)$	0.5	0.25	0.15	0.1

3. The returns on a couple of funds depends on the state of the economy. The economy is expected to be Good with a probability of 20%, Fair with probability of 50% and Poor with probability of 30%. Which fund would you choose if you want to maximize your return? What would you choose if you really dislike risk?

State of Economy	Fund 1	Fund 2
Good	20	40
Fair	10	20
Poor	-10	-40

Exercise 2

1. Use the table below. A portfolio has 200,000 dollars invested in Asset X and 300,000 dollars in asset Y. If the correlation coefficient between the two investments is 0.4, what is the expected return and standard deviation of the portfolio?

Measure	X	Y
Expected Return (%)	8	12
Standard Deviation (%)	12	20

Exercise 3

1. Let Z be a binomial random variable with $n = 5$ and $p = 0.35$ use the binomial formula to find $P(Z = 1)$, $P(Z \geq 2)$. What is the expected value and standard deviation of Z?
2. Let W be a binomial random variable with $n = 200$ and $p = 0.77$ use the binomial formula to find $P(W > 160)$, $P(155 \leq W \leq 165)$. What is the expected value and standard deviation of W?
3. Sixty percent of a firm's employees are men. Suppose four of the firm's employees are randomly selected. What is more likely, finding three men and one woman, or two men and one woman? Does your answer change if the proportion falls to 50%?

Exercise 4

1. Assume that S is a Poisson process with mean of $\mu = 1.5$. Calculate $P(S = 2)$ and $P(S \geq 2)$. What is the mean and standard deviation of S?
2. Assume that T is a Poisson process with mean of $\mu = 20$. Calculate $P(T = 14)$ and $P(18 \leq T \leq 23)$.
3. A local pharmacy administers on average 84 Covid-19 vaccines per week. The vaccines shots are evenly administered across all days. Find the probability that the number of vaccine shots administered on a Wednesday is more than eight but less than 12.

Exercise 5

1. Assume that X is a hypergeometric random variable with $N = 25$, $S = 3$, and $n = 4$. Calculate $P(X = 0)$, $P(X = 1)$, and $P(X \leq 1)$.
2. Compute the probability of at least eight successes in a random sample of 20 items obtained from a population of 100 items that contains 25 successes. What are the expected value and standard deviation of the number of successes?
3. For 1 dollar a player gets to select six numbers for the base game of Powerball. In the game, five balls are randomly drawn from 59 consecutively numbered white balls. One ball, called the Powerball, is randomly drawn from 39 consecutively numbered red balls. What is the probability that a player is able to match two out of five randomly drawn white balls? What is the probability of winning the jackpot?

9.3 Answers

Exercise 1

1. The expected value is 10.75 and the standard deviation is 5.31. The probability of $x < 15$ is 0.65.

In R we can create vectors for both x and the probabilities $P(X = x)$.

```
x<-c(5,10,15,20)
px<-c(0.35,0.3,0.2,0.15)
```

The expected value is the sum product of probabilities and values. Formally, $\sum_{i=1}^{n} x_i p_i$ and in R:

```
(ex<-sum(x*px))
```

```
[1] 10.75
```

The standard deviation is given by $\sqrt{\sum_{i=1}^{n}(x_i - \mu)^2 p_i}$. We can calculate it in R with the following code:

```
(sd<-sqrt(sum((x-ex)^2*px)))
```

```
[1] 5.30919
```

2. The expected value is -17.4 and the standard deviation is 6.86. The probability of is 0.25.

Let's create the vectors once more in R.

```
y<-c(-23,-17,-9,-3)
py<-c(0.5,0.25,0.15,0.1)
```

The expected value is given by:

```
(ey<-sum(y*py))
```

```
[1] -17.4
```

The standard deviation is given by:

```r
(sdy<-sqrt(sum((y-ey)^2*py)))
```

```
[1] 6.858571
```

3. Both funds have the same expected return of 6. The safest return comes from fund 1 since the standard deviation is only 11.14 vs. 31.05 for fund 2.

In R we can create a data frame with probabilities and the performance of the funds.

```r
funds<-data.frame(probs=c(0.2,0.5,0.3),fund1=c(20,10,-10), fund2=c(40,20,-40))
```

Let's create a function for the expected value and standard deviation. For the expected value:

```r
Expected_Value<-function(x,p){
  sum(x*p)
}
```

Now we can use the formula to calculate the expected value of fund1:

```r
Expected_Value(funds$fund1,funds$probs)
```

```
[1] 6
```

and fund 2:

```r
Expected_Value(funds$fund2,funds$probs)
```

```
[1] 6
```

For the standard deviation we can create another function:

```r
Standard_Deviation<-function(x,p){
  sqrt(sum((x-Expected_Value(x,p))^2*p))
}
```

Using the function to get the standard deviation of fund 1 we get:

```r
Standard_Deviation(funds$fund1,funds$probs)
```

```
[1] 11.13553
```

and for fund 2:

```r
Standard_Deviation(funds$fund2,funds$probs)
```

```
[1] 31.04835
```

Exercise 2

1. The expected return of the portfolio is 10.4 and the standard deviation is 14.60.

In R we can start by calculating the expected return. This is given by the formula $\alpha R_1+\beta R_2$:

```r
(ER<-(2/5)*8+(3/5)*12)
```

```
[1] 10.4
```

Next we can find the standard deviation with the formula $\sqrt{\alpha^2\sigma_1^2+\beta^2\sigma_2+\alpha\beta\rho\sigma_1\sigma_2}$:

```r
(Risk<-sqrt(0.4^2*12^2 + 0.6^2*20^2+2*0.4*0.6*0.4*12*20))
```

```
[1] 14.59863
```

Exercise 3

1. $P(Z=1)=0.31$, and $P(Z\geq 2)=0.57$. The expected value is $np=1.75$ and the standard deviation is $\sqrt{np(1-p)}=1.067$.

Let's use R and the `dbinom()` function to find $P(Z=1)$.

```r
dbinom(1,5,0.35)
```

```
[1] 0.3123859
```

We can now use `pbinom()` to find the cumulative distribution. Since we want the right tale of the distribution, we will specify this with an argument.

```r
pbinom(1,5,0.35, lower.tail=F)
```

```
[1] 0.571585
```

2. $P(W > 160) = 0.14$, and $P(155 \leq W \leq 165) = 0.45$. The expected value is $np = 154$ and the standard deviation is $\sqrt{np(1-p)} = 5.95$.

Using the `pbinom()` function we find that $P(W > 160)$.

```r
pbinom(160,200,0.77, lower.tail = F)
```

```
[1] 0.136611
```

We make two calculations to find the probability. First, $P(W \leq 165)$ and then $P(W \geq 154)$. The difference between these two, gives us the desired outcome.

```r
pbinom(165,200,0.77, lower.tail=T)-pbinom(154,200,0.77, lower.tail=T)
```

```
[1] 0.4487104
```

3. The probabilities are the same. Each event has a probability of 0.3456. If the probability changes to 0.5 now the event of two women and two men is more likely.

Let's calculate the probabilities in R. First, the probability of three men and one woman.

```r
dbinom(3,4,0.6)
```

```
[1] 0.3456
```

Now the probability of two men and two women.

```r
dbinom(2,4,0.6)
```

```
[1] 0.3456
```

Changing the probabilities reveals that:

```
dbinom(3,4,0.5)
```

```
[1] 0.25
```

```
dbinom(2,4,0.5)
```

```
[1] 0.375
```

Having two of each is the most likely outcome.

Exercise 4

1. The $P(S = 2) = 0.25$ and $P(S \geq 2) = 0.44$. The expected value and the variance is 1.5.

In R we will make use of the `dpois()` function:

```
dpois(2,1.5)
```

```
[1] 0.2510214
```

For the second probability we will use `ppois()`:

```
ppois(1,1.5, lower.tail=F)
```

```
[1] 0.4421746
```

2. The $P(T = 14) = 0.039$ and $P(18 \leq T \leq 23) = 0.49$.

Using the `dpois()` function once more:

```
dpois(14,20)
```

```
[1] 0.03873664
```

For the second probability we will find the difference between two probabilities:

```r
ppois(23,20, lower.tail=T)-ppois(17,20, lower.tail=T)
```

```
[1] 0.4904644
```

3. The probability of administering more than 8 but less than 12 shots is 0.3.

Let's first note that if 84 shots are administered on average weekly, then 12 are administered daily. Now we can use this average and the `ppois()` function to find the probability:

```r
ppois(11,12)-ppois(8,12)
```

```
[1] 0.3065696
```

Exercise 5

1. $P(X = 0) = 0.58$, $P(X = 1) = 0.37$, and $P(X \leq 1) = 0.94$.

In R we can use the `dhyper()` function

```r
dhyper(0,3,22,4)
```

```
[1] 0.5782609
```

once more for the second probability:

```r
dhyper(1, 3, 22, 4)
```

```
[1] 0.3652174
```

For the last probability we can add the previous probabilities or use the `phyper()` function:

```r
phyper(1, 3, 22, 4)
```

```
[1] 0.9434783
```

2. The probability is 0.545.

In R we use the `dhyper()` function once more:

```r
dhyper(0, 2, 10, 3)
```

```
[1] 0.5454545
```

3. The probability of matching two white balls is 5. Winning the jackpot is extremely unlikely! A probability of 0.00000000512. It is more likely to be struck by lightning according to the CDC.

In R use the `dhyper()` function:

```r
dhyper(2, 5, 54, 5)
```

```
[1] 0.04954472
```

For the jackpot we first calculate the probability of getting all of the white balls.

```r
options(digits = 5,scipen=999)
dhyper(5, 5, 54, 5)
```

```
[1] 0.00000019974
```

Now the probability of getting the Powerball.

```r
dhyper(1, 1, 38, 1)
```

```
[1] 0.025641
```

Since the two events are independent, we can multiply them to find the probability of a jackpot.

```r
dhyper(5, 5, 54, 5)*dhyper(1, 1, 38, 1)
```

```
[1] 0.0000000051217
```

10 Probability III

10.1 Concepts

Continuous Random Variables

Continuous random variables are characterized by their probability density function $f(x)$. The probability density function does not directly provide probabilities!

The probability of a continuous random variable assuming a single value is zero. Instead, probabilities are defined for intervals. These are calculated by areas under the PDF curve (integral).

Uniform Distribution

The **uniform** probability density function is given by $f(x) = \frac{1}{b-a}$ when $a \leq x \leq b$ and 0 otherwise.

The **expected value** of the uniform distribution is $E(x) = \frac{a+b}{2}$.

The **variance** of the uniform distribution is $var(x) = \frac{(b-a)^2}{12}$

Normal Distribution

The **normal** PDF is given by $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{x-\mu}{\sigma})}$, where μ is the mean, σ is the standard deviation, π is 3.1415... , and e is 2.7282... . The normal distribution has the following properties:

- The normal curve is symmetrical about the mean μ.
- The mean is at the middle and divides the area of the distribution into halves.
- The total area under the curve is equal to 1.
- The distribution is completely determined by its mean and standard deviation.

The **standard normal** distribution has a mean of 0 and a standard deviation of 1.

Exponential Distribution

The **exponential distribution** is useful in computing probabilities for the time it takes to complete a task. It describes the time between events in a Poisson process.

The probability density function is given by $f(x) = \frac{1}{\mu}e^{\frac{-x}{\mu}}$.

Triangular Distribution

The **triangular distribution** is characterized by a single mode (the peak of the distribution) and two boundaries. It is often used in situations where the lower and upper bounds of a potential outcome are known, but the exact likelihood of the outcome is uncertain.

The probability density function is given by $f(x) = \frac{2(x-a)}{(b-a)(c-a)}$ for $a \leq x < c$; $f(x) = \frac{2}{(b-a)}$ for $x = c$; $f(x) = \frac{2(b-x)}{(b-a)(b-c)}$ for $c < x \leq b$, and $f(x) = 0$ otherwise.

The **expected value** of the distribution is $E(x) = \frac{a+b+c}{3}$.
The **variance** of the triangular distribution is $var(x) = \frac{a^2+b^2+c^2-ab-ac-bc}{18}$.

Useful R Functions

To calculate the density of continuous random variables use the `dunif()`, `dnorm()`, and `dexp()` functions. For the triangular distribution use the extraDistr package and the `dtriang()` function.

To calculate probabilities of continuous random variables use the `punif()`, `pnorm()`, `pexp()`, and `ptriang()` functions.

To calculate quartiles of continuous random variables use the `qunif()`, `qnorm()`,`qexp()`, and `qtriang()` functions.

To calculate generate random variables based on continuous random variables use the `runif()`, `rnorm()`, `rexp()`, and `rtriang()` functions.

10.2 Exercises

The following exercises will help you practice some probability concepts and formulas. In particular, the exercises work on:

- Calculating probabilities for continuous random variables.
- Calculating the expected value and standard deviation.

- Applying the uniform, normal, and exponential distributions.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

For the following exercises, make your calculations by hand and verify results with a calculator or R.

1. A random variable X follows a continuous uniform distribution with minimum of -2 and maximum of 4. Determine the height of the density function $f(x)$, the mean, the standard deviation, and calculate $P(X \leq -1)$.
2. Your internet provider will arrive sometime between 10:00 am and 12:00 pm. Suppose you have to run a quick errand at 10:00 am. If it takes 15 minutes to run the errand, what is the probability that you will be back before the internet provider arrives? What if you take 30 minutes?

Exercise 2

1. A random variable Z follows a standard normal distribution. Find $P(-0.67 \leq Z \leq -0.23)$, $P(0 \leq Z \leq 1.96)$, $P(-1.28 \leq Z \leq 0)$ and $P(Z > 4.2)$.
2. Let Y be normally distributed with $\mu = 2.5$ and $\sigma = 2$. Find $P(Y > 7.6)$, $P(7.4 \leq Y \leq 10.6)$, a y such that $P(Y > y) = 0.025$, and a y such that $P(y \leq Y \leq 2.5) = 0.4943$.
3. Assume that football game times are normally distributed with a mean of 3 hours and a standard deviation of 0.4 hour. What is the probability that the game lasts at most 2.5 hours? Find the maximum value for a game to be in the bottom 1% of the distribution.

Exercise 3

1. Random variable S is exponentially distributed with mean of 0.1. What is the standard deviation of S? What is $P(0.10 \leq S \leq 0.2)$?
2. A tollbooth operator has observed that cars arrive randomly at a rate of 360 cars per hour. What is the mean time between car arrivals? What is the probability that the next car will arrive within ten seconds?

10.3 Answers

Exercise 1

1. The height of the density function $f(x) = 0.1667$, the mean is 1, standard deviation is 1.73, and $P(X \leq -1) = 0.1667$.

$f(x)$ can be easily estimated by using the formula of the continuous uniform random variable. $f(x) = \frac{1}{b-a}$. Using R as a calculator we find:

```
1/(4-(-2))
```

```
[1] 0.1666667
```

The mean is given by $\mu = \frac{a+b}{2}$. In R we determine that the mean is:

```
(-2+4)/2
```

```
[1] 1
```

The standard deviation is $\sigma = \sqrt{\frac{(b-a)^2}{12}}$. Using R we find:

```
sqrt((4-(-2))^2/12)
```

```
[1] 1.732051
```

Finally, we can find the probability of Z being less than -1 by using the `punif()` function:

```
punif(-1,-2,4)
```

```
[1] 0.1666667
```

2. The probability that you will arrive on time is 0.875. If the time of the errand is 30 minutes, then the probability goes down to 0.75.

There is a 120 minute interval in which the IP can arrive. The density function is given by $f(x) = 1/120$. Using R we can find $P(X > 15)$:

```r
punif(15,0,120,lower.tail=F)
```

```
[1] 0.875
```

Once more we can find $P(X > 30)$:

```r
punif(30,0,120,lower.tail=F)
```

```
[1] 0.75
```

Exercise 2

1. $P(-0.67 \leq Z \leq -0.23) = 0.158$, $P(0 \leq Z \leq 1.96) = 0.475$, $P(-1.28 \leq Z \leq 0) = 0.4$ and $P(Z > 4.2) \approx 0$.

Use the `pnorm()` function to find the probabilities. $P(-0.67 \leq Z \leq -0.23)$:

```r
pnorm(-0.23)-pnorm(-0.67)
```

```
[1] 0.157617
```

$P(0 \leq Z \leq 1.96)$

```r
pnorm(1.96)-pnorm(0)
```

```
[1] 0.4750021
```

$P(-1.28 \leq Z \leq 0)$

```r
pnorm(0)-pnorm(-1.28)
```

```
[1] 0.3997274
```

$P(Z > 4.2)$

```
options(scipen=999)
pnorm(4.2,lower.tail = F)
```

```
[1] 0.00001334575
```

2. $P(Y > 7.6) = 0.005386$, $P(7.4 \leq Y \leq 10.6) = 0.0071$, a y such that $P(Y > y) = 0.025$ is 6.42, and a y such that $P(y \leq Y \leq 2.5)$ is -2.56.

Let's use once more the `pnorm()` function in R.

$P(Y > 7.6)$

```
pnorm(7.6,2.5,2,lower.tail = F)
```

```
[1] 0.005386146
```

$P(7.4 \leq Y \leq 10.6)$

```
pnorm(10.6,2.5,2)-pnorm(7.4,2.5,2)
```

```
[1] 0.007117202
```

y such that $P(Y > y) = 0.025$

```
qnorm(0.025,2.5,2,lower.tail = F)
```

```
[1] 6.419928
```

y such that $P(y \leq Y \leq 2.5) = 0.4943$. Note that 2.5 is the mean. Hence we are looking for a y that has $0.5 - 0.4943 = 0.0057$ on the left:

```
qnorm(0.0057,2.5,2)
```

```
[1] -2.560385
```

3. The probability is 10.56%. A game lasting no more than 2.069 hours would be in the bottom 1%.

Let's use `pnorm()` once more in R.

```r
pnorm(2.5,3,0.4)
```

```
[1] 0.1056498
```

For the threshold we can use `qnorm()`

```r
qnorm(0.01,3,0.4)
```

```
[1] 2.069461
```

Exercise 3

1. The standard deviation is equal to the mean 0.1. $P(0.10 \leq S \leq 0.2) = 0.2325$

Let's use `pexp()` in R:

```r
pexp(0.2,rate = 10)-pexp(0.1,rate = 10)
```

```
[1] 0.2325442
```

2. The mean time between car arrivals is $1/360 = 0.002778$. The probability that the next car will arrive within the next 10 seconds is 0.6321.

Once more we use `pexp()` in R

```r
pexp(1/360,360)
```

```
[1] 0.6321206
```

11 Inference I

11.1 Concepts

Statistical Inference

The goal of statistical inference is gain insight on a **population parameter** by using a **sample statistic**. It is required that the sample statistic be calculated from a random sample from the population where each element is selected independently.

A sample mean is used to infer the population mean. Some properties of the sample mean are:

- The expected value of the sample means is equal to the population mean (i.e., the sample mean is unbiased). Formally, $E(\bar{x}_i) = \mu$.
- The standard deviation of the sample means is lower than the population standard deviation. $\sigma_{\bar{x}} = \sigma/\sqrt{n}$. We call this measure the **standard error**.
- If the population is normally distributed, then the sample means ($\bar{x}$'s) are normally distributed.
- If the population is not normally distributed, the the sample means are also normally distributed if the sample size is large (i.e., $n > 30$). This is the **central limit theorem**.

Proportions

Recall that the **binomial distribution** describes the number of successes x in n trials of a Bernoulli process where p is the probability of success. Here, x/n is the proportion of successes.

- To estimate the **population proportion** use the **sample proportion** $\bar{p} = x/n$. This estimate is unbiased (i.e., $E(\bar{p}) = P$), where P is the population proportion.
- The **standard error** of the estimate is $se(\bar{P}) = \sqrt{\frac{p(1-p)}{n}}$, where p is the sample proportion, and n is the sample size.
- By the central limit theorem, the **sampling distribution** of $\bar{p}$ is approximately normal when $np \geq 5$ and $n(1-p) \geq 5$.

Useful R Functions

Here are some functions that are handy when simulating data in R.

The `pnorm()` and `punif()` functions calculate probabilities for the normal and uniform distributions, respectively.

The `rnorm()` and `runif()` functions generate random numbers from a normal and uniform distribution, respectively.

The `for()` function creates a loop that repeats a procedure a specified amount of times.

The `set.seed()` function is used to create reproducible results in R when random numbers are used.

11.2 Exercises

The following exercises will help you test your knowledge on the Inference. In particular, the exercises work on:

- The Central Limit Theorem.
- Sampling Distribution for means.
- Sampling Distribution for proportions.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

In this exercise we will be simulating the central limit theorem. You will need R to complete this problem.

1. Create a random sample of 1000 data points and store it in an object called *Population*. Use the uniform distribution with min of 100 and max of 200 to generate the sample. Calculate the mean and standard deviation of the random sample and call *PopMean* and *PopSD*, respectively.
2. Create a for loop (with 1000 iterations) that takes a sample of 10 points from *population*, calculate the mean, and then store the result in a vector called *SampleMeans*. Calculate the mean of the *SampleMeans* object. How does this mean compare to *PopMean*? How does the standard deviation compare to *PopSD*?
3. Create a histogram for the sample means. Is the distribution uniform? Is it normal? What is the probability that the sample mean is between 140 and 160?

Exercise 2

1. A random sample of $n = 100$ is taken from a population with mean $\mu = 80$ and standard deviation $\sigma = 14$. Calculate the expected value and standard error for the sampling distribution of the sampling means. What is the probability that the sample mean falls between 77 and 85?
2. Assume that miles-per-gallons of combustion cars are normally distributed with mean of 33.8 and standard deviation of 3.5. What is the probability that the mean mpg of four randomly selected cars is more than 35? What is the probability that all four selected cars have mpg greater than 35?

Exercise 3

1. A random sample of $n = 200$ is taken from a population with a proportion of $p = 0.75$. Calculate the expected value and standard error of the proportion sampling distribution. What is the probability that the sample proportion is between 0.7 and 0.8?

2.Twenty-three percent of employees at a fintech firm work from home. If we take a sample of 50 employees, what is the probability that more than 20% of them are working from home? What if the sample increases to 200? Why does the probability change?

Exercise 4

1. A production process for energy drinks is being evaluated. The machine that fills the cans is calibrated so that each can has 350ml of drink with a standard deviation of 10ml. Every hour, ten cans are sampled and the average amount of drink is recorded (see table below). Is the machine working properly?

1	2	3	4	5	6	7	8
$\bar{x} = 310$	$\bar{x} = 315$	$\bar{x} = 325$	$\bar{x} = 330$	$\bar{x} = 328$	$\bar{x} = 347$	$\bar{x} = 339$	$\bar{x} = 350$

2. The production of Good Guy dolls has a 1% defective rate. A quality inspector takes five samples of size 1000. The proportions are shown in the table below. Is the production process under control?

1	2	3	4	5
$\bar{p} = 0.009$	$\bar{p} = 0.012$	$\bar{p} = 0.008$	$\bar{p} = 0.011$	$\bar{p} = 0.0102$

11.3 Answers

Exercise 1

Let's start by creating the random sample. We can use the `runif()` function in R to do this. We will set a seed so that results are reproducible.

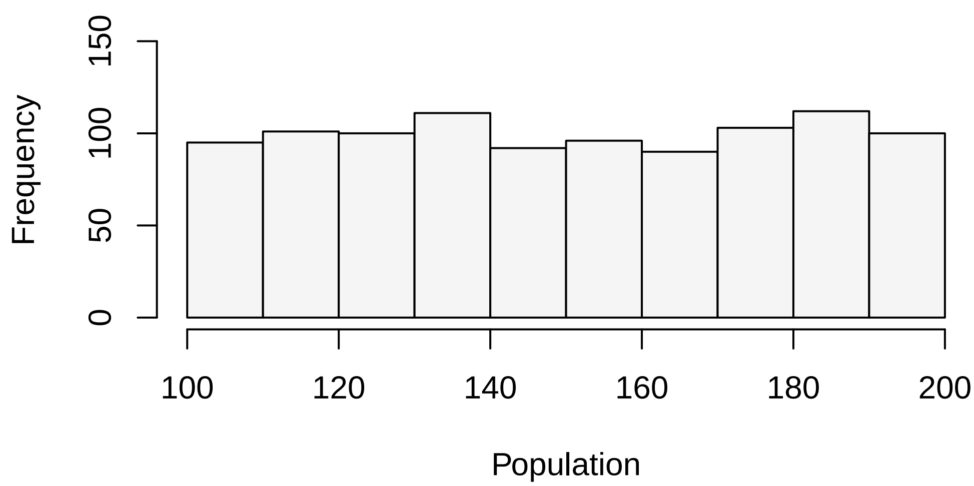
```
set.seed(10)
Population<-runif(1000,100,200)
```

Next, we can save the mean and the standard deviation of the population in two different object:

```
PopMean<-mean(Population)
PopSD<-sd(Population)
```

The mean and standard deviation are 150.53 and 29.2. Let's quickly create a histogram of population, so that we can convince ourselves that the data is uniformly distributed.

```
hist(Population, main="", ylim=c(0,160), col="#F5F5F5")
```



2. Now let's create a for loop that allows us to sample the population several times. In fact, we will sample the population 1000 times and record the mean of the samples.

```r
nrep<-1000
SampleMeans<-c()
for (i in 1:nrep){
  x<-sample(Population,10,replace=T)
  SampleMeans<-c(SampleMeans,mean(x))
}
```

Now we can calculate the mean of the sample means in R:

```r
mean(SampleMeans)
```

```
[1] 150.4177
```

Note that the mean is very close to *PopMean*. In the limit (that is if we take many more samples), these two values are equal to each other. Now let's calculate the standard deviation of the sample means.

```r
sd(SampleMeans)
```

```
[1] 9.134147
```

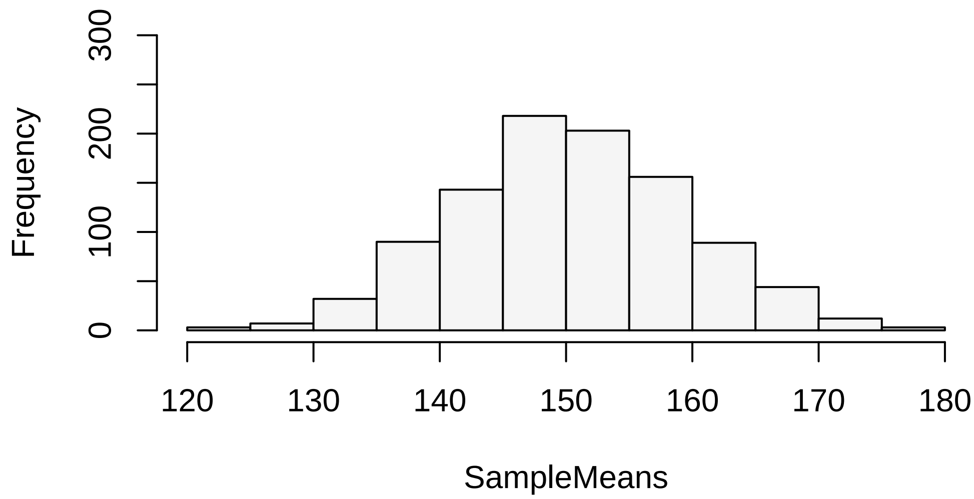
As you can see, the standard deviation is much lower. In fact, if we take *PopSD* and divide by 10 (the size of the sample), we should get close to the standard deviation of the sample means.

```r
PopSD/sqrt(10)
```

```
[1] 9.233644
```

3. To create the histogram we use the `hist()` function once more:

```r
hist(SampleMeans, main="", ylim=c(0,300), col="#F5F5F5")
```



The distribution looks normal. To be clear, if the population follows a uniform distribution, we have shown that the distribution of the sample means is normal with a mean equal to the population mean and a smaller standard deviation.
We can use the distribution of the sample means to calculate the probability. Noting the the distribution is normal:

```
pnorm(160,mean(SampleMeans),sd(SampleMeans))-pnorm(140,mean(SampleMeans),sd(SampleMeans))
```

```
[1] 0.7258913
```

There is a 72.59% probability that the sample mean is between 140 and 160.

Exercise 2

1. The expected value is 80 since it is equal to the mean of the population. The standard error is 1.4. The probability is 98.38%.

We can use R as a calculator to find the standard error.

```
14/sqrt(100)
```

```
[1] 1.4
```

We can use `pnorm()` to find the probability:

```
pnorm(85,80,1.4)-pnorm(77,80,1.4)
```

```
[1] 0.9837602
```

2. The probabilities are 24.66% and 1.8%.

For the first probability we can use a sample size of 4 and use the standard error in the `pnorm()` function.

```
pnorm(35,33.8,3.5/sqrt(4),lower.tail = F)
```

```
[1] 0.2464466
```

For the second probability we can first calculate the probability that a randomly selected car has mpg greater than 35. In R:

```
(p35<-pnorm(35,33.8,3.5,lower.tail = F))
```

```
[1] 0.365853
```

Since draws are independent we get:

```
p35^4
```

```
[1] 0.01791539
```

Exercise 3

1. The expected value is 0.75, the same as the population. The standard error is $\sqrt{p(1-p)/n} = 0.03$. The probability for a sample of 200 is 0.8975.

The standard error is given by:

```
sqrt(0.75*0.25/200)
```

```
[1] 0.03061862
```

In R we can use the `pnorm()` function one more time to find the probability.

```
pnorm(0.8,0.75,sqrt(0.75*0.25/200))-pnorm(0.7,0.75,sqrt(0.75*0.25/200))
```

```
[1] 0.8975296
```

2. The probability with a sample of 50 is 69.29%. When the sample is 200 the probability is 84.33%. As the sample size increases the standard error goes down. This means that the distribution of the sample proportions gets tighter and there is more area to the right of $\bar{p} = 0.2$.

In R we can use the `pnorm()` function one more time with a mean of 0.2 and $n = 50$.

```
pnorm(0.2,0.23,sqrt(0.23*0.77/50),lower.tail = F)
```

```
[1] 0.6928964
```

Updating the code so that $n = 200$ yields:

```
pnorm(0.2,0.23,sqrt(0.23*0.77/200),lower.tail = F)
```

```
[1] 0.8433098
```

Exercise 4

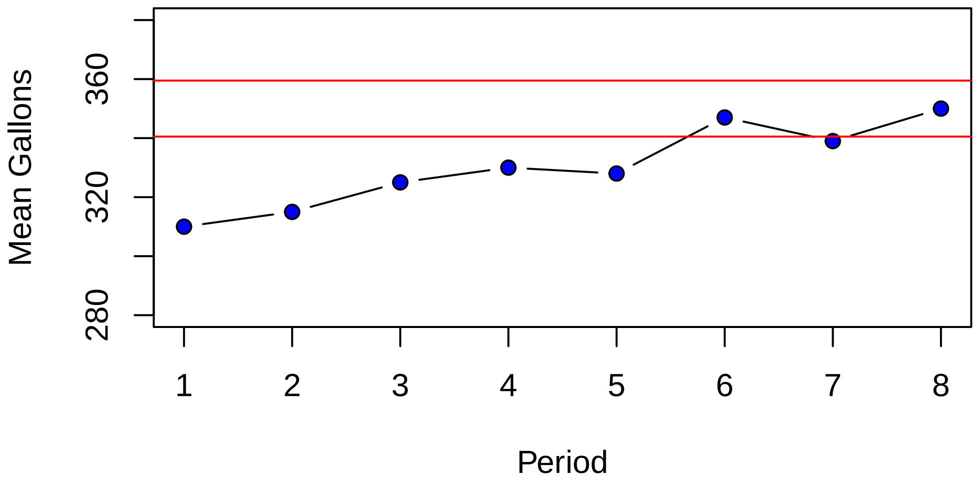
1. The process seems to be out of control. In the early samples, the machine is not filling the cans with enough drink. Although, in the later periods the machine reverts back to the expected performance, it seems unlikely that it will remain functioning correctly.

Let's start by calculating the upper and lower limits in R.

```
dataEx1<-c(310,315,325,330,328,347,339,350)
ulEx1<-350+3*(10/sqrt(10))
llEx1<-350-3*(10/sqrt(10))
```

We can graph the samples and the limits to determine the stability of the production process.

```
plot(dataEx1, type="b", ylab="Mean Gallons",
     xlab="Period", pch=21, bg="blue",ylim=c(280,380))
abline(h=ulEx1,col="red")
abline(h=llEx1,col="red")
```



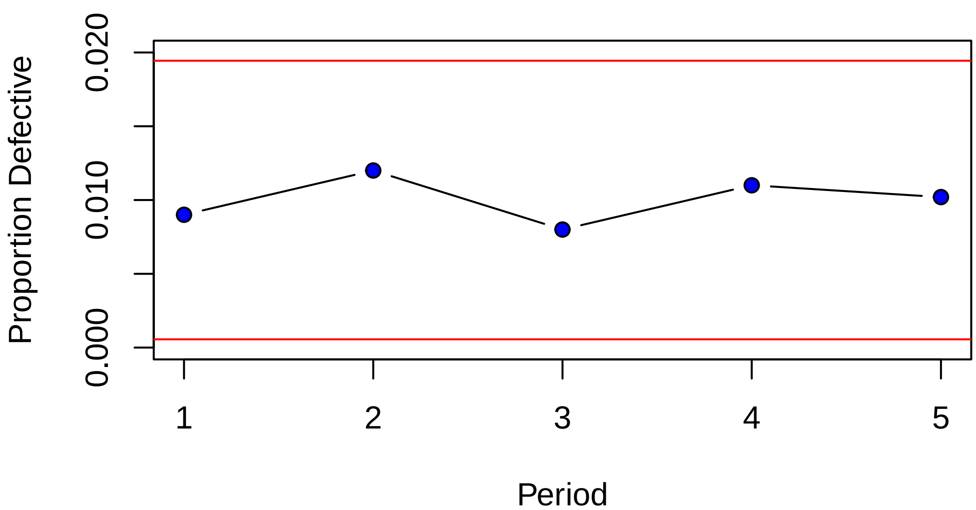
2. Good Dolls production looks good. All proportions fall between three standard errors of the mean.

Once more we can calculate upper and lower limits for the proportions.

```
dataEx2<-c(0.009,0.012,0.008,0.011,0.0102)
ulEx2<-0.01+3*sqrt(0.01*0.99/1000)
llEx2<-0.01-3*sqrt(0.01*0.99/1000)
```

Graphing the results in R we can observe the production process and the sample proportions.

```
plot(dataEx2, type="b", ylab="Proportion Defective",
     xlab="Period", pch=21, bg="blue",ylim=c(0,0.02))
abline(h=ulEx2,col="red")
abline(h=llEx2,col="red")
```



12 Inference II

12.1 Concepts

Confidence Intervals

A **confidence interval** provides a range of values that, with a certain level of confidence, contains the population parameter of interest. For proper confidence intervals ensure that the sampling distributions are normal.

A 95% **confidence level**, indicates that if the interval were constructed many times (from independent samples of the population), it would include the true population parameter 95% of the time.

A **significance level** (α) of 5%, means that the confidence interval would would not include the true population parameter 5% of the time.

The interval for the population mean when the population standard deviation is unknown is given by $\bar{x} \pm t_{\alpha/2,df}\frac{s}{\sqrt{n}}$, where $\bar{x}$ is the point estimate, $t_{a/2,df}\frac{s}{\sqrt{n}}$ is the margin of error, α is the allowed probability that the interval does not include μ, and df are the degrees of freedom $n-1$.

The interval for the population proportion mean is given by $\bar{p} \pm z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}$.

Useful R Functions

The `qnorm()` and `qt()` functions calculate quartiles for the normal and t distributions, respectively.

The `if()` function creates a conditional statement in R.

12.2 Exercises

The following exercises will help you test your knowledge on Statistical Inference. In particular, the exercises work on:

- Simulating confidence intervals.
- Estimating confidence intervals in R.
- Estimating confidence intervals for proportions.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

In this exercise you will be simulating confidence intervals.

1. Set the seed to 9. Create a random sample of 1000 data points and store it in an object called *Population*. Use the exponential distribution with rate of 0.02 to generate the data. Calculate the mean and standard deviation of *Population* and call them *PopMean* and *PopSD* respectively. What are the mean and standard deviation of *Population*?
2. Create a for loop (with 10,000 iterations) that takes a sample of 50 points from *Population*, calculates the mean, and then stores the result in a vector called *SampleMeans*. What is the mean of the *SampleMeans*?
3. Create a 90% confidence interval using the first data point in the *SampleMeans* vector. Does the confidence interval include *PopMean*?
4. Now take the minimum of the *SampleMeans* vector. Create a new 90% confidence interval. Does the interval include *PopMean*? Out of the $10,000$ intervals that you could construct with the vector *SampleMeans*, how many would you expect to include *PopMean*?

Exercise 2

1. A random sample of 24 observations is used to estimate the population mean. The sample mean is 104.6 and the standard deviation is 28.8. The population is normally distributed. Construct a 90% and 95% confidence interval for the population mean. How does the confidence level affect the size of the interval?
2. A random sample from a normally distributed population yields a mean of 48.68 and a standard deviation of 33.64. Compute a 95% confidence interval assuming a) that the sample size is 16 and b) the sample size is 25. What happens to the confidence interval as the sample size increases?

Exercise 3

You will need the **sleep** data set for this problem. The data is built into R, and displays the effect of two sleep inducing drugs on students. Calculate a 95% confidence interval for group 1 and for group 2. Which drug would you expect to be more effective at increasing sleeping times?

Exercise 4

1. A random sample of 100 observations results in 40 successes. Construct a 90% and 95% confidence interval for the population proportion. Can we conclude at either confidence level that the population proportion differs from 0.5?

2. You will need the **HairEyeColor** data set for this problem. The data is built into R, and displays the distribution of hair and eye color for 592 statistics students. Construct a 95 confidence interval for the proportion of Hazel eye color students.

12.3 Answers

Exercise 1

1. The mean of *Population* is 48.61. The standard deviation is 47.94.

Start by generating values from the exponential distribution. You can use the `rexp()` function in R to do this. Setting the seed to 9 yields:

```
set.seed(9)
Population<-rexp(1000,0.02)
```

The population mean is:

```
(PopMean<-mean(Population))
```

```
[1] 48.61053
```

The standard deviation is:

```
(PopSD<-sd(Population))
```

```
[1] 47.94411
```

2. The mean is very close to the population mean 48.83. The standard deviation is 6.83.

In R you can use a for loop to create the vector of sample means.

```
nrep<-10000
SampleMeans<-c()
for (i in 1:nrep){
  x<-sample(Population,50,replace=T)
  SampleMeans<-c(SampleMeans,mean(x))
}
```

The mean of *SampleMeans* is:

```
(xbar<-mean(SampleMeans))
```

```
[1] 48.7005
```

The standard deviation is:

```
(Standard<-sd(SampleMeans))
```

```
[1] 6.827595
```

3. The confidence interval is [47.71,70.17]. Since the population mean is equal to 48.61, the confidence interval does include the population mean.

Let's construct the upper an lower limits of the interval in R.

```
(ll<-SampleMeans[1]+qnorm(0.05)*Standard)
```

```
[1] 47.71385
```

```
(ul<-SampleMeans[1]-qnorm(0.05)*Standard)
```

```
[1] 70.17464
```

4. The confidence interval is [14.86,37.32]. This interval does not include the population mean of 48.61. Out of the $10,000$ confidence intervals, one would expect about $9,000$ to include the population mean.

Let's find the confidence interval limits using R.

```r
(Minll<-min(SampleMeans)+qnorm(0.05)*Standard)
```

```
[1] 14.85631
```

```r
(Minul<-min(SampleMeans)-qnorm(0.05)*Standard)
```

```
[1] 37.31709
```

We can confirm in R that about $9,000$ of the intervals include *PopMean*. Once more, let's use a for loop to construct confidence intervals for each element in *SampleMeans* and check whether the *PopMean* is included. The count variable keeps track of how many intervals include the population mean.

```r
count=0

for (i in SampleMeans){
  (ll<-i+qnorm(0.05)*Standard)
  (ul<-i-qnorm(0.05)*Standard)
  if (PopMean<=ul & PopMean>=ll){
    count=count+1
  }
}

count
```

```
[1] 8978
```

Exercise 2

1. The 90% confidence interval is [94.52,114.67] and the 95% confidence interval is [114.68,116.76]. The larger the confidence level, the larger the interval.

Let's construct the intervals using R. Since the population standard deviation is unknown we will use the t-distribution. The interval is constructed as $\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$.

```r
(ul90<-104.6-qt(0.05,23)*28.8/sqrt(24))
```

```
[1] 114.6755
```

```r
(ll90<-104.6+qt(0.05,23)*28.8/sqrt(24))
```

```
[1] 94.52453
```

For the 95% confidence interval we adjust the significance level accordingly.

```r
(ul95<-104.6-qt(0.025,23)*28.8/sqrt(24))
```

```
[1] 116.7612
```

```r
(ll95<-104.6+qt(0.025,23)*28.8/sqrt(24))
```

```
[1] 92.43883
```

2. The confidence interval for a sample size of 16 is [30.75,66.61]. The confidence interval when the sample size is 25 is [34.79,62.57]. As the sample size gets larger, the confidence interval gets narrower and more precise.

Let's use R again to calculate the confidence interval. For a sample size of 16 the interval is:

```r
(ul16<-48.68-qt(0.025,15)*33.64/sqrt(16))
```

```
[1] 66.60549
```

```r
(ll16<-48.68+qt(0.025,15)*33.64/sqrt(16))
```

```
[1] 30.75451
```

Increasing the ample size to 25 yields:

```r
(ul25<-48.68-qt(0.025,24)*33.64/sqrt(25))
```

```
[1] 62.56591
```

```r
(ll25<-48.68+qt(0.025,24)*33.64/sqrt(25))
```

```
[1] 34.79409
```

Exercise 3

1. The 95% confidence interval for group 1 is $[-0.53, 2.03]$.

Let's first calculate the standard error for group 1.

```
(se1<-sd(sleep$extra[sleep$group==1])/sqrt(length(sleep$extra[sleep$group==1])))
```

```
[1] 0.5657345
```

We can now use the standard error to estimate the lower and upper limits of the confidence interval.

```
(ll1<-mean(sleep$extra[sleep$group==1])+qt(0.025,9)*se1)
```

```
[1] -0.5297804
```

```
(ul1<-mean(sleep$extra[sleep$group==1])-qt(0.025,9)*se1)
```

```
[1] 2.02978
```

2. The 95% confidence interval for group 2 is $[0.90, 3.76]$.

Let's repeat the procedure for group 2. Start by finding the standard error.

```
(se2<-sd(sleep$extra[sleep$group==2])/sqrt(length(sleep$extra[sleep$group==2])))
```

```
[1] 0.6331666
```

Using the standard error we can complete the confidence interval.

```
(ll2<-mean(sleep$extra[sleep$group==2])+qt(0.025,9)*se2)
```

```
[1] 0.8976775
```

```
(ul2<-mean(sleep$extra[sleep$group==2])-qt(0.025,9)*se2)
```

```
[1] 3.762322
```

3. Drug 2. Drug 2 does not include zero in the interval, and the interval is to the right of zero. It is unlikely, that drug 2 has no effect on students sleeping time. Additionally, Drug 2's mean increase in sleeping hours is 2.33 vs. 0.75 for drug 1.

Exercise 4

1. The 90% and 95% confidence intervals are [0.319,0.481], and [0.304,0.496] respectively. Since they do not include 0.5, we can conclude that the population proportion is significantly different from 0.5.

We can create an object that stores the sample proportion and sample in R:

```
(p<-0.4)
```

```
[1] 0.4
```

```
(n<-100)
```

```
[1] 100
```

The 90% confidence interval is given by:

```
(Ex1ll90<-p+qnorm(0.05)*sqrt(p*(1-p)/100))
```

```
[1] 0.319419
```

```
(Ex1ul90<-p-qnorm(0.05)*sqrt(p*(1-p)/100))
```

```
[1] 0.480581
```

The 95% confidence interval is:

```
(Ex1ll90<-p+qnorm(0.025)*sqrt(p*(1-p)/100))
```

```
[1] 0.3039818
```

```
(Ex1ul90<-p-qnorm(0.025)*sqrt(p*(1-p)/100))
```

```
[1] 0.4960182
```

2. The 90% confidence interval is [0.132,0.182].The 95% confidence interval is [0.128,0.186].

The data can easily be viewed by calling `HairEyeColor` in R.

```r
HairEyeColor
```

```
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

Note that there are three dimensions to this table (Hair, Eye, Sex). We can calculate the proportion of Hazel eye colored students with the following command that makes use of indexing:

```r
(p<-(sum(HairEyeColor[,3,1])+sum(HairEyeColor[,3,2]))/sum(HairEyeColor))
```

```
[1] 0.1570946
```

Now we can use this proportion to construct the intervals. Recall that for proportions the interval is calculated by $\bar{p} \pm z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}$. The 90% confidence interval is given by:

```r
(Ex2ll90<-p+qnorm(0.05)*sqrt(p*(1-p)/592))
```

```
[1] 0.1324945
```

```r
(Ex2ul90<-p-qnorm(0.05)*sqrt(p*(1-p)/592))
```

```
[1] 0.1816947
```

The 95% confidence interval is:

```
(Ex2ll95<-p+qnorm(0.025)*sqrt(p*(1-p)/592))
```

```
[1] 0.1277818
```

```
(Ex2ul95<-p-qnorm(0.025)*sqrt(p*(1-p)/592))
```

```
[1] 0.1864074
```

13 Inference II

13.1 Concepts

Confidence Intervals

A **confidence interval** provides a range of values that, with a certain level of confidence, contains the population parameter of interest. For proper confidence intervals ensure that the sampling distributions are normal.

A 95% **confidence level**, indicates that if the interval were constructed many times (from independent samples of the population), it would include the true population parameter 95% of the time.

A **significance level** (α) of 5%, means that the confidence interval would would not include the true population parameter 5% of the time.

The interval for the population mean when the population standard deviation is unknown is given by $\bar{x} \pm t_{\alpha/2,df}\frac{s}{\sqrt{n}}$, where $\bar{x}$ is the point estimate, $t_{a/2,df}\frac{s}{\sqrt{n}}$ is the margin of error, α is the allowed probability that the interval does not include μ, and df are the degrees of freedom $n-1$.

The interval for the population proportion mean is given by $\bar{p} \pm z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}$.

Useful R Functions

The `qnorm()` and `qt()` functions calculate quartiles for the normal and t distributions, respectively.

The `if()` function creates a conditional statement in R.

13.2 Exercises

The following exercises will help you test your knowledge on Statistical Inference. In particular, the exercises work on:

- Simulating confidence intervals.
- Estimating confidence intervals in R.
- Estimating confidence intervals for proportions.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

In this exercise you will be simulating confidence intervals.

1. Set the seed to 9. Create a random sample of 1000 data points and store it in an object called *Population*. Use the exponential distribution with rate of 0.02 to generate the data. Calculate the mean and standard deviation of *Population* and call them *PopMean* and *PopSD* respectively. What are the mean and standard deviation of *Population*?
2. Create a for loop (with 10,000 iterations) that takes a sample of 50 points from *Population*, calculates the mean, and then stores the result in a vector called *SampleMeans*. What is the mean of the *SampleMeans*?
3. Create a 90% confidence interval using the first data point in the *SampleMeans* vector. Does the confidence interval include *PopMean*?
4. Now take the minimum of the *SampleMeans* vector. Create a new 90% confidence interval. Does the interval include *PopMean*? Out of the $10,000$ intervals that you could construct with the vector *SampleMeans*, how many would you expect to include *PopMean*?

Exercise 2

1. A random sample of 24 observations is used to estimate the population mean. The sample mean is 104.6 and the standard deviation is 28.8. The population is normally distributed. Construct a 90% and 95% confidence interval for the population mean. How does the confidence level affect the size of the interval?
2. A random sample from a normally distributed population yields a mean of 48.68 and a standard deviation of 33.64. Compute a 95% confidence interval assuming a) that the sample size is 16 and b) the sample size is 25. What happens to the confidence interval as the sample size increases?

Exercise 3

You will need the **sleep** data set for this problem. The data is built into R, and displays the effect of two sleep inducing drugs on students. Calculate a 95% confidence interval for group 1 and for group 2. Which drug would you expect to be more effective at increasing sleeping times?

Exercise 4

1. A random sample of 100 observations results in 40 successes. Construct a 90% and 95% confidence interval for the population proportion. Can we conclude at either confidence level that the population proportion differs from 0.5?
2. You will need the **HairEyeColor** data set for this problem. The data is built into R, and displays the distribution of hair and eye color for 592 statistics students. Construct a 95 confidence interval for the proportion of Hazel eye color students.

13.3 Answers

Exercise 1

1. The mean of *Population* is 48.61. The standard deviation is 47.94.

Start by generating values from the exponential distribution. You can use the `rexp()` function in R to do this. Setting the seed to 9 yields:

```
set.seed(9)
Population<-rexp(1000,0.02)
```

The population mean is:

```
(PopMean<-mean(Population))
```

```
[1] 48.61053
```

The standard deviation is:

```
(PopSD<-sd(Population))
```

```
[1] 47.94411
```

2. The mean is very close to the population mean 48.83. The standard deviation is 6.83.

In R you can use a for loop to create the vector of sample means.

```
nrep<-10000
SampleMeans<-c()
for (i in 1:nrep){
  x<-sample(Population,50,replace=T)
  SampleMeans<-c(SampleMeans,mean(x))
}
```

The mean of *SampleMeans* is:

```
(xbar<-mean(SampleMeans))
```

```
[1] 48.7005
```

The standard deviation is:

```
(Standard<-sd(SampleMeans))
```

```
[1] 6.827595
```

3. The confidence interval is [47.71,70.17]. Since the population mean is equal to 48.61, the confidence interval does include the population mean.

Let's construct the upper an lower limits of the interval in R.

```
(ll<-SampleMeans[1]+qnorm(0.05)*Standard)
```

```
[1] 47.71385
```

```
(ul<-SampleMeans[1]-qnorm(0.05)*Standard)
```

```
[1] 70.17464
```

4. The confidence interval is [14.86,37.32]. This interval does not include the population mean of 48.61. Out of the $10,000$ confidence intervals, one would expect about $9,000$ to include the population mean.

Let's find the confidence interval limits using R.

```r
(Minll<-min(SampleMeans)+qnorm(0.05)*Standard)
```

```
[1] 14.85631
```

```r
(Minul<-min(SampleMeans)-qnorm(0.05)*Standard)
```

```
[1] 37.31709
```

We can confirm in R that about $9,000$ of the intervals include *PopMean*. Once more, let's use a for loop to construct confidence intervals for each element in *SampleMeans* and check whether the *PopMean* is included. The count variable keeps track of how many intervals include the population mean.

```r
count=0

for (i in SampleMeans){
  (ll<-i+qnorm(0.05)*Standard)
  (ul<-i-qnorm(0.05)*Standard)
  if (PopMean<=ul & PopMean>=ll){
    count=count+1
  }
}

count
```

```
[1] 8978
```

Exercise 2

1. The 90% confidence interval is [94.52,114.67] and the 95% confidence interval is [114.68,116.76]. The larger the confidence level, the larger the interval.

Let's construct the intervals using R. Since the population standard deviation is unknown we will use the t-distribution. The interval is constructed as $\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$.

```r
(ul90<-104.6-qt(0.05,23)*28.8/sqrt(24))
```

```
[1] 114.6755
```

```r
(ll90<-104.6+qt(0.05,23)*28.8/sqrt(24))
```

```
[1] 94.52453
```

For the 95% confidence interval we adjust the significance level accordingly.

```r
(ul95<-104.6-qt(0.025,23)*28.8/sqrt(24))
```

```
[1] 116.7612
```

```r
(ll95<-104.6+qt(0.025,23)*28.8/sqrt(24))
```

```
[1] 92.43883
```

2. The confidence interval for a sample size of 16 is [30.75,66.61]. The confidence interval when the sample size is 25 is [34.79,62.57]. As the sample size gets larger, the confidence interval gets narrower and more precise.

Let's use R again to calculate the confidence interval. For a sample size of 16 the interval is:

```r
(ul16<-48.68-qt(0.025,15)*33.64/sqrt(16))
```

```
[1] 66.60549
```

```r
(ll16<-48.68+qt(0.025,15)*33.64/sqrt(16))
```

```
[1] 30.75451
```

Increasing the ample size to 25 yields:

```r
(ul25<-48.68-qt(0.025,24)*33.64/sqrt(25))
```

```
[1] 62.56591
```

```r
(ll25<-48.68+qt(0.025,24)*33.64/sqrt(25))
```

```
[1] 34.79409
```

Exercise 3

1. The 95% confidence interval for group 1 is [−0.53,2.03].

Let's first calculate the standard error for group 1.

```
(se1<-sd(sleep$extra[sleep$group==1])/sqrt(length(sleep$extra[sleep$group==1])))
```

```
[1] 0.5657345
```

We can now use the standard error to estimate the lower and upper limits of the confidence interval.

```
(ll1<-mean(sleep$extra[sleep$group==1])+qt(0.025,9)*se1)
```

```
[1] -0.5297804
```

```
(ul1<-mean(sleep$extra[sleep$group==1])-qt(0.025,9)*se1)
```

```
[1] 2.02978
```

2. The 95% confidence interval for group 2 is [0.90,3.76].

Let's repeat the procedure for group 2. Start by finding the standard error.

```
(se2<-sd(sleep$extra[sleep$group==2])/sqrt(length(sleep$extra[sleep$group==2])))
```

```
[1] 0.6331666
```

Using the standard error we can complete the confidence interval.

```
(ll2<-mean(sleep$extra[sleep$group==2])+qt(0.025,9)*se2)
```

```
[1] 0.8976775
```

```
(ul2<-mean(sleep$extra[sleep$group==2])-qt(0.025,9)*se2)
```

```
[1] 3.762322
```

3. Drug 2. Drug 2 does not include zero in the interval, and the interval is to the right of zero. It is unlikely, that drug 2 has no effect on students sleeping time. Additionally, Drug 2's mean increase in sleeping hours is 2.33 vs. 0.75 for drug 1.

Exercise 4

1. The 90% and 95% confidence intervals are [0.319,0.481], and [0.304,0.496] respectively. Since they do not include 0.5, we can conclude that the population proportion is significantly different from 0.5.

We can create an object that stores the sample proportion and sample in R:

```
(p<-0.4)
```

```
[1] 0.4
```

```
(n<-100)
```

```
[1] 100
```

The 90% confidence interval is given by:

```
(Ex1ll90<-p+qnorm(0.05)*sqrt(p*(1-p)/100))
```

```
[1] 0.319419
```

```
(Ex1ul90<-p-qnorm(0.05)*sqrt(p*(1-p)/100))
```

```
[1] 0.480581
```

The 95% confidence interval is:

```
(Ex1ll90<-p+qnorm(0.025)*sqrt(p*(1-p)/100))
```

```
[1] 0.3039818
```

```
(Ex1ul90<-p-qnorm(0.025)*sqrt(p*(1-p)/100))
```

```
[1] 0.4960182
```

2. The 90% confidence interval is [0.132,0.182].The 95% confidence interval is [0.128,0.186].

The data can easily be viewed by calling `HairEyeColor` in R.

```r
HairEyeColor
```

```
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

Note that there are three dimensions to this table (Hair, Eye, Sex). We can calculate the proportion of Hazel eye colored students with the following command that makes use of indexing:

```r
(p<-(sum(HairEyeColor[,3,1])+sum(HairEyeColor[,3,2]))/sum(HairEyeColor))
```

```
[1] 0.1570946
```

Now we can use this proportion to construct the intervals. Recall that for proportions the interval is calculated by $\bar{p} \pm z_{\alpha/2}\sqrt{\frac{\bar{p}(1-\bar{p})}{n}}$. The 90% confidence interval is given by:

```r
(Ex2ll90<-p+qnorm(0.05)*sqrt(p*(1-p)/592))
```

```
[1] 0.1324945
```

```r
(Ex2ul90<-p-qnorm(0.05)*sqrt(p*(1-p)/592))
```

```
[1] 0.1816947
```

The 95% confidence interval is:

```
(Ex2ll95<-p+qnorm(0.025)*sqrt(p*(1-p)/592))
```

```
[1] 0.1277818
```

```
(Ex2ul95<-p-qnorm(0.025)*sqrt(p*(1-p)/592))
```

```
[1] 0.1864074
```

14 Regression and Inference

14.1 Concepts

Correlation Significance

To determine the statistical significance of the correlation coefficient we test:

- $H_o : \rho \geq 0$; $H_a : \rho < 0$ left tail
- $H_o : \rho \leq 0$; $H_a : \rho > 0$ right tail
- $H_o : \rho = 0$; $H_a : \rho \neq 0$ two tails

The test statistic for the correlation is given by $t_{df} = \frac{r_{xy}\sqrt{n-2}}{\sqrt{1-r_{xy}^2}}$, where $df = n - 2$ and r_{xy} is the sample correlation coefficient.

Run the `cor.test()` function to perform the test on two vectors. Here is a list of arguments to use:

- *alternative*: is a choice between "two.sided", "less" and "greater".
- *conf.level*: sets the confidence level. Enter as a decimal and not percentage.

Difference of Means Tests

Tests for inference about the difference of two population means.

- The test for unpaired mean differences (not equal variances) is given by $t_{df} = \frac{(\bar{x}_1 - \bar{x}_2) - \bar{d}_o}{\sqrt{\frac{s_1^2}{n_1} \frac{s_2^2}{n_2}}}$.
- The test for unpaired mean difference (equal variances) is given by $t_{df} = \frac{(\bar{x}_1 - \bar{x}_2) - \bar{d}_o}{\sqrt{s_p^2(\frac{1}{n_1} + \frac{1}{n_2})}}$.
- The test for paired mean difference is given by $t_{df} = \frac{\bar{d} - d_o}{\frac{s}{\sqrt{n}}}$.

Run these test in R by using the `t.test()` function. Here is a list of arguments to use:

- *paired*: use True for paired, False for independent. The default is False.

- *var.equal*: use True for equal variances, False for unequal. The default is False.
- *mu*: a value that indicate the hypothesized value of the mean or mean difference.
- *alternative*: is a choice between "two.sided", "less" and "greater".
- *conf.level*: sets the confidence level. Enter as a decimal and not percentage.

Regression Inference

When running regression a couple of test can be performed on the coefficients to determine significance:

- The first test competing hypothesis are $H_o : \beta_j = 0$; $H_a : \beta_j \neq 0$. The test statistic for the intercept (slope) coefficient is given by $t_{df} = \frac{b_j}{se(b_j)}$.
- The second test competing hypothesis are $H_o : \beta_1 = \beta_2 = ...\beta_k = 0$; H_a : *at least one* $\beta_i \neq 0$. The joint test of significance is given by $F_{df_1,df_2} = \frac{SSR/k}{SSE/(n-k-1)} = \frac{MSR}{MSE}$. The Anova table below shows more detail on this test.

Anova	df	SS	MS	F	Significance
Regression	k	SSR	$MSR = \frac{SSR}{k}$	$F_{df_1,df_2} = \frac{MSR}{MSE}$	$P(F) \geq \frac{MSR}{MSE}$
Residual	$n-k-1$	SSE	$MSE = \frac{SSE}{n-k-1}$		
Total	$n-1$	SST			

To conduct these tests, save the `lm()` model into an object. The `summary()` function can then be used to retrieve the results of the tests on the model's parameters. Use the `anova()` function to obtain the Anova table.

14.2 Exercises

The following exercises will help you test your knowledge on Regression and Inference. In particular, the exercises work on:

- Determining the significance of correlations.
- Conduct paired and unpaired test of means and proportions.
- Determining the significance of the slope and intercept estimates both individually and jointly.
- Developing prediction intervals.

Answers are provided below. Try not to peak until you have a formulated your own answer and double checked your work for any mistakes.

Exercise 1

1. Consider the following competing hypothesis: $H_o : \rho = 0$, $H_a : \rho \neq 0$. A sample of 25 observations reveals that the correlation coefficient between two variables is 0.15. At a 5% confidence level, can we reject the null hypothesis?
2. Install the `ISLR2` package in R. Use the **Hitters** data set to look at the relationship between *Hits* and *Salary*. Specifically, calculate the correlation coefficient and test the competing hypothesis $H_o : \rho = 0$, $H_a : \rho \neq 0$ at the 1% significance level.

Exercise 2

1. Install the `ISLR2` package in R. Use the **Hitters** data set to investigate if the average hits were significantly different between the two divisions (American and National). Use the *NewLeague* and *Hits* variables to test the hypothesis at the 5% significance level. Is there reason to believe that the population variances are different?
2. Use the `ISLR2` package for this question. Particularly, use the **BrainCancer** data set to test whether males have a higher average survival time than women. Use the *sex* and *time* variables to test the hypothesis at the 5% significance level. Is there reason to believe that the population variances are different?

Exercise 3

1. Use the **sleep** data set included in R. At the 1% significance level, is there an effect of the drug on the 10 patients? Assume that the *group* variable denotes before (1) the drug is administered and after (2) the drug is administered.

Exercise 4

1. Install the `ISLR2` package in R. Use the **Hitters** data set to investigate the effect of *HmRun*,*RBI*, and *Years* on a players *Salary*. Which variables are statistically different from zero? Are the variables jointly significant? Does the R^2 suggest a good fit of the data to the model?
2. José Altuve had 28 home runs, 57 RBI's, and has been in the league for 12 years. What is the model's predicted salary for him? What is the 95% prediction interval? Note: The model predicts his salary if he played in 1987.

14.3 Answers

Exercise 1

1. At the 5% significance level, we can not reject the null since the p-value is $0.47 > 0.05$.

Recall that the t-stat is calculated by $\frac{r_{xy}\sqrt{n-2}}{\sqrt{1-r_{xy}^2}}$. We can use R as a calculator to calculate this value:

```
rxy<-0.15
n<-25
(tstat<-(rxy*sqrt(n-2))/(sqrt(1-rxy^2)))
```

```
[1] 0.7276069
```

Now, we can estimate the p-value using the `pt()` function:

```
2*pt(tstat,n-2,lower.tail = F)
```

```
[1] 0.4741966
```

2. The estimated correlation of 0.44 and the t-value is 7.89. Since the p-value is approximately 0 we reject the null hypothesis $H_o : \rho = 0$.

Once the `ISLR2` package is downloaded, it can be loaded to R using the `library()` function. The `cor.test()` function conducts the appropriate test of significance.

```
library(ISLR2)
cor.test(Hitters$Salary,Hitters$Hits, conf.level = 0.95)
```

```

	Pearson's product-moment correlation

data:  Hitters$Salary and Hitters$Hits
t = 7.8863, df = 261, p-value = 8.531e-14
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3355210 0.5314332
sample estimates:
      cor 
0.4386747 
```

Exercise 2

1. There is no reason to believe that the population variances are different. Players are recruited from what seems to be a common pool. At a 5% significance level, the difference of the two means is not significantly different from zero. We can't reject the null hypothesis.

We will use the `t.test()` function in R to test the hypothesis. We note that the test is not paired, two sided and of equal variances in the population.

```
t.test(Hitters$Hits[Hitters$NewLeague=="A"],
       Hitters$Hits[Hitters$NewLeague=="N"],paired = F,
       alternative = "two.sided",mu = 0,var.equal = T,
       conf.level = 0.95 )
```

```

    Two Sample t-test

data:  Hitters$Hits[Hitters$NewLeague == "A"] and Hitters$Hits[Hitters$NewLeague == "N"]
t = 1.0862, df = 320, p-value = 0.2782
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.581286 15.875028
sample estimates:
mean of x mean of y
103.58523  97.93836
```

2. There might be reason to believe that the population variances are different. Women and men are known to have medical differences. At a 5% significance level, the average survival time of men seems not to be larger than that of women. We can't reject the null hypothesis $H_o : \bar{x_1} - \bar{x_2} \leq 0$.

Once more use the `t.test()` function in R to test the hypothesis. Note that the test is not paired, right-tailed and of different variances in the population.

```
t.test(BrainCancer$time[BrainCancer$sex=="Male"],
       BrainCancer$time[BrainCancer$sex=="Female"],paired = F,
       alternative = "greater",mu = 0, var.equal = F,
       conf.level = 0.95 )
```

```

    Welch Two Sample t-test
```

```
data:  BrainCancer$time[BrainCancer$sex == "Male"] and BrainCancer$time[BrainCancer$sex == "F
t = -0.30524, df = 84.867, p-value = 0.6195
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -8.504999       Inf
sample estimates:
mean of x mean of y
 26.78302  28.10200
```

Exercise 3

1. There drug seems to have an effect as we can reject the null hypothesis $H_o : \bar{d} = 0$. The difference of means seems to be statistically different from zero.

Use the `t.test()` function once more in R. Make sure to note that the test is paired, and two-tailed.

```r
t.test(sleep$extra[sleep$group==1],
       sleep$extra[sleep$group==2], paired=T,
       alternative = "two.sided", mu=0, conf.level = 0.99)
```

```

    Paired t-test

data:  sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true mean difference is not equal to 0
99 percent confidence interval:
 -2.8440519 -0.3159481
sample estimates:
mean difference
          -1.58
```

Exercise 4

1. Both *RBI* and *Years* are statistically significant and the salary of a player increases as they gain more experience and have more RBI's. Home runs do not seem to have an impact on the salary of a player according to the data. The F-Statistics reveals that the coefficients are jointly significant since the p-value is approximately zero. Both the Multiple and Adjusted R^2 suggest that the model only accounts for 32% of the variation

in *Salary*. We might have to include more variable in our model to better explain the salary of a player.

We can run a linear regression in R by using the `lm()` function. We'll use the `summary()` function to get more details on the model's performance.

```r
fit<-lm(Salary~HmRun+RBI+Years,data=Hitters)
summary(fit)
```

```
Call:
lm(formula = Salary ~ HmRun + RBI + Years, data = Hitters)

Residuals:
    Min      1Q  Median      3Q     Max 
-752.31 -197.27  -66.80   97.73 2151.78 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -90.086     61.142  -1.473    0.142    
HmRun         -7.346      4.972  -1.478    0.141    
RBI            9.156      1.685   5.432 1.28e-07 ***
Years         32.818      4.838   6.783 7.97e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 372.2 on 259 degrees of freedom
  (59 observations deleted due to missingness)
Multiple R-squared:  0.3269,	Adjusted R-squared:  0.3191 
F-statistic: 41.93 on 3 and 259 DF,  p-value: < 2.2e-16
```

2. The predicted salary is 619.93 and the 95% prediction interval is $[-129.89, 1369.7]$.

```r
new<-data.frame(HmRun=28,RBI=57,Years=12)
predict(fit,newdata=new,level=0.95,interval="prediction")
```

```
       fit       lwr      upr
1 619.9268 -129.8905 1369.744
```

15 Objects and Vectors

15.1 Concepts

Objects

An **object** is a data structure that stores a value or a set of values, along with information about the type of data and any associated attributes. Objects are usually created by assigning a value to a variable name. You can assign values by using either `=` or `<-`.
When naming objects in R use *PascalCase*, *camelCase*, *snake_case* or *dot.case*.

```
ScreenTime<-120
```

Vectors

A **vector** is a one-dimensional array that can hold elements of any data type.
Some common data types are numeric, character, logical, and complex.
Use the `c()` function to concatenate (combine) elements and store them in a vector.

```
ScreenTimeDays<-c(110,115,120,98,60)
```

Functions

In general, **functions** relate an input (arguments) to an output. For example, the `sum()` function takes as an input a vector with numeric values and returns the sum of the elements.

```
SleepingHours<-c(10,9,6,8)
sum(SleepingHours)
```

```
[1] 33
```

To learn more about a function you can use `?`. For example, to learn more about the `sum()` function, write `?sum` in the console.

Data Types

The main data types are numeric, character, logical, date, and complex. To identify the data type stored in a vector use the `class()` function.

```
class(SleepingHours)
```

```
[1] "numeric"
```

Useful R Functions

15.2 Exercises

References

Grolemund, Garret. 2014. "Hands-on Programming with r." https://jjallaire.github.io/hopr/.
Wickham, Hadley. 2017. "R for Data Science." https://r4ds.hadley.nz.