Chapter 2: Creating graphs from data

Learning objectives of this chapter:

- Creating a histogram by hand and in R
- Creating and modifying a scatter plot in R
- Creating and modifying a line plot in R
- Creating a box plot in R

Assignment 2.1: Creating a histogram by hand

Suppose that you have the following set of numbers:

1.5 5.5 1.7 7.2 1.2 7.9 1.4 3.6 3.1 3.8 5.9 3.6 5.1 3.2 7.1

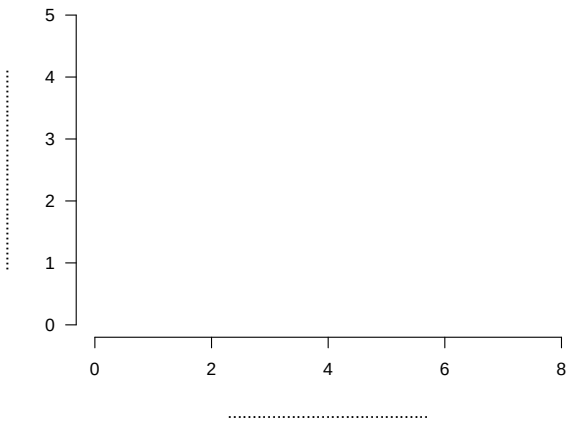
2.1 a) For each of the ranges given below, find out their **frequency** in the data above.

Answer 2.1a:

0 to 2	2 to 4	4 to 6	6 to 8
Frequency: ____	Frequency: ____	Frequency: ____	Frequency: ____

2.1 b) Draw the histogram for these data by hand. The name of the *x-axis* should represent the range of values, and the name of the *y-axis* should represent the **frequency** of the values that fall in the ranges of assignment 2.1a.

Answer 2.1 b:



Assignment 2.2: Creating and modifying a histogram in R

2.2 a) Take the numbers from assignment 2.1a and use the **c()** function to enter these numbers in a vector called **dataset4** . Next, run the following code in R and explain what you see.

```
hist(dataset4)
```

Answer 2.2a:

2.2 b) The graph resulting from the command in 2.2a is not the same as the histogram that you drew in assignment 2.1b. What is the difference between the graph from 2.2a and your graph from 2.1b?

Answer 2.2b:

2.2 c) Try to replicate your exact histogram from 2.1b in R. This requires that you specify the **breaks** argument after a comma in the **hist()** function.

Hint 2.1: You can find more information on the **hist()** *function by running* **?hist** .

R code 2.2c:

2.2 d) Give your histogram a different color by adding and changing the **col** argument after the comma. Improve your histogram further by changing the *x-axis* name, the main title, and the rotation of the *y-axis* labels by adding more arguments.

Hint 2.2: Check Part II of the R help for additional arguments to the **hist()** *function.*

R code 2.2d:

2.2 e) Calculate the **mean** and **median** of the values of **dataset4** . Is the distribution of **dataset4** **negatively skewed** or **positively skewed** ? Explain your answer using the relation between the **mean** and the **median** .

Answer 2.2e:

These data sets are **positively** / **negatively** / **not** skewed.

Explanation:

Assignment 2.3: Creating and modifying a scatter plot in R

In this assignment you are going to work with a data set that is built into R.

The `data()` function gives you access to all the data sets that are built into R, or that are included with downloaded R packages. For example, you can load a data set called `swiss` into the environment by running the R code below:

```
data(swiss)
```

The data are imported as an object called `swiss` , which you can now also see in the environment. This particular data set contains some socio-economic indicators for each of 47 French-speaking provinces of Switzerland. In this assignment, you will focus on the column `Education` (the percentage education beyond primary school) and the column `Agriculture` (the percentage of males involved in agriculture as an occupation).

2.3 a) Extract the values of the two columns using the `$` sign and store them in two new variables called `education` and `agriculture` .

Hint 2.3: You can check the R environment to see the names of the objects that you have currently stored.

R code 2.3a:

```
```

2.3 b) Use the `plot()` function to create a scatter plot of the two variables. Place the percentage of education beyond primary school on the *x-axis* and the percentage of males involved in agriculture as an occupation on the *y-axis*. Rename your axis labels to match the content of the graph.

R code 2.3b:

```
```

2.3 c) Looking at the scatter plot, what can you globally say about the relation between the percentage of education beyond primary school and the percentage of males involved in agriculture as an occupation in the French-speaking provinces of Switzerland?

Answer 2.3c:

2.3 d) Give the points in your scatter plot a different color by changing the `col` argument after the comma. Improve the graph further by changing the main title, the rotation of the *y-axis* labels, the shape of the points, and removing the outer lines.

R code 2.3d:

Assignment 2.4: Creating and modifying a line plot in R

In this assignment, you are going to create a line plot of the daily closing prices of some major European stock indices: Germany DAX, Switzerland SMI, France CAC, and UK FTSE.

You can load this data set by running the code below:

```
data(EuStockMarkets)
stockData <- data.frame(EuStockMarkets)
```

The data is now loaded into the environment as the `EuStockMarkets` data set and immediately transformed to the `stockData` data set (this is because of technical reasons as the original data is in a time-series format). In this assignment, you will work with the `stockData` data set.

2.4 a) Create a line plot where the closing price of the DAX stock is displayed over time. Give your plot appropriate *x-axis* and *y-axis* names.

R code 2.4a:

```

```

2.4 b) Find a function to add a separate line for the closing price of the SMI stock in red. You may try to add the other stocks as well using this function, but remember to adjust the *y-axis* accordingly so that all the lines are all displayed decently.

Hint 2.4: Do not use the `plot()` *function to add a line to an already existing plot.*

R code 2.4 b:

```

```

Assignment 2.5: Creating a box plot in R

For this next assignment, let's return to the **swiss** data set. More specifically, you will now have to focus on the values that are stored in the separate variable **agriculture** .

2.5 a) Find out the **minimum** , **lower quartile** , **median** , **upper quartile** , and **maximum** of the **agriculture** variable.

R code 2.5a:

Answer 2.5a:

Minimum: _____ Upper quartile: _____

Median: _____ Lower quartile: _____

Maximum: _____

2.5 b) Create a boxplot of the **agriculture** variable.

R code 2.5b:

2.5 c) Run the following code in R and explain the table output. How does the code work?

```
educationLevel <- rep('2.Medium', 47)
educationLevel[education <= 6] = '1.Low'
educationLevel[education >= 12] = '3.High'
table(educationLevel)
```

Answer 2.5c:

2.5 d) Create a box plot using the R code below and explain what you see.

```
boxplot(agriculture ~ educationLevel)
```

Answer 2.5d: