Prof. Dr. Steffen Wagner
Angewandte Statistik
Fachbereich II
Berliner Hochschule für Technik

Exercise 02: Frequency data

Statistical Computing – WiSe 2022/23

Preparations

a. Make sure you work inside a RStudio project. Details please see Exercise 01.
b. Create a folder named `data` inside your RStudio project folder.
c. Download the data set *GU.csv* from Moodle[1] an save it in the data folder just created.

The Gotham City data set

Work through this worksheet entering the commands and examples as you did last week. You should type in all your commands into the script file. In many of the code examples the output is not included, you should always see what the output is and make sure you understand it. Please add comments (in your own words) to your code so that it will make sense to you when you return several days, weeks or years later.

Please do not copy and paste the commands from this PDF file, type them in yourself. Copy/paste is the quickest way to forget what you have just done!

Reading text data

a. Read the data from `GU.csv` using the command `read.csv` via

```
Gotham <- read.csv(file = "data/GU.csv")
```

If this causes an error check that you have typed in the command exactly as above, check that the file is in your project in the `data` directory, and that you have not saved the file in another format e.g. in Excel format.

b. For further information regarding *reading and writing data* work through the slide deck *Programming with R (see https://lms.bht-berlin.de/mod/resource/view.php?id=944673) slides 57 - 64.

c. Start investigating the Gotham data set by running

```
nrow(Gotham)
ncol(Gotham)
dim(Gotham)
```

Comment the given code so that a reader not experienced with R understands what the code does.

Variable names and types

a. `Gotham` is stored in R as a `data.frame` object:

```
class(Gotham)
```

b. You have already found out how many variables are in this data set. Use `names(Gotham)` to find the names of the variables in the data set.

c. You can access a variable (= column) in a data frame easily using the dollar sign

```
Gotham$Income
```

d. To find out the variable type of the values stored in `Gotham$Income` use

[1] https://lms.bht-berlin.de/mod/folder/view.php?id=969045

```
class(Gotham$Income)
```

e. Determine the data types of all the variables in the data set.

f. Run `str(Gotham)` and discuss the outcome.

Frequencies for a nominal variable

As nominal variable we choose `Gotham$DegreeSubject`.

a. Table of **absolute frequencies**:

```
table(Gotham$DegreeSubject)
```

b. Table of **relative frequencies**:

```
prop.table(
  table(Gotham$DegreeSubject)
)
```

c. The amount of decimal places is a bit overkill! Let's round the proportions to 3 decimal places and present them as percentages

```
round(
  prop.table(
    table(Gotham$DegreeSubject)
  ), digits = 3) * 100
```

d. We can present both the absolute and relative frequences in one object using the function `cbind`:

```
cbind(table(Gotham$DegreeSubject),
      round(
      prop.table(
        table(Gotham$DegreeSubject)
      ), digits = 3) * 100)
```

e. What do you think the function `rbind()` does? Try it out. Check the help `?rbind`.

f. Table of **cumulative frequencies**:

```
cumsum(
  table(Gotham$DegreeSubject)
)
```

Interprete the outcome. Does it make sense?

Graphics for frequency data

a. Presenting frequency data is usually done using a bar chart:

```
barplot(
  table(Gotham$DegreeSubject)
)
```

Click on the *Zoom button* in the *Plots* tab to see an enlarged version of the plot.

There are two purposes for obtaining graphics for our data:

i) to investigate the data by eye or
ii) to communicate the properties of the data to others

b. For i) this plot is sufficient. For ii) we need to add arguments for the titles, colours or a legend. As an example

```
barplot(
  table(Gotham$DegreeSubject),
  main="26 Gotham City Uni Students",
  sub="Degree Subject",
  ylab="Frequency",
  col=c("lightgreen", "lightpink", "lightblue", "orange")
)
```

c. For continuous variables a bar chart is not a good idea. To find out why, plot a bar chart for the variable *Income*.

d. Now we go for a histogram[2]:

```
hist(Gotham$Income)
```

e. Improve the layout of the histogram using the arguments you have learned by improving the bar chart.

Exercise: Frequency table

a. Obtain the four types frequency for the variable `number of siblings`, in the GCU data and present them all in one table.
b. Obtain a bar chart for the number of siblings.

Tydying up

a. Make sure your script file has sensible comments. At least one comment for each main section.
b. Save your script file. Use another folder inside your RStudio project folder named *scripts*
c. Leave RStudio and **do not** save the workspace!

[2]You will learn the statistical details of a histogram in coming weeks.

Written exercises

Here are three easy exercises for you to work on with pen, paper and maybe a calculator (i.e. not using R). These exercises are to reinforce the concepts covered in the lecture.

Complete the gaps . . .

in the following text:

At the end of the 2021/22 school year, five-hundred students at a college were asked about their decision to take a course at this college. There are in total 1251 students. The population consists of all 1251 students, whereas the students who took part in the survey is a sample. The sample size is $n = 500$.

Frequency tables by hand

A tyre replacement garage asked 20 customers for feedback about their satisfaction with the work done. The responses are coded as *s*= satisfied, *a*= average, *u*=unsatisfied and as follows:

```
s, a, s, s, u, s, s, a, a, s, s, a, u, a, a, s, u, a, a, u, a, a, s, s, a
```

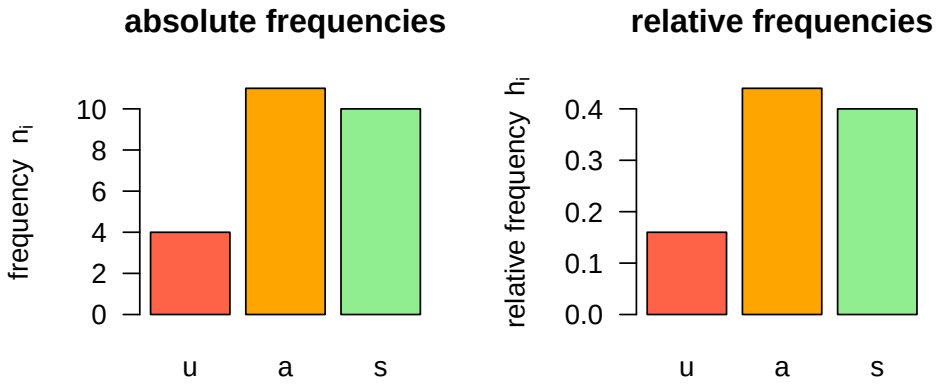
a. Obtain the four types of frequencies covered in the lecture and present them in a table.
   - absolute frequency n_i
   - relative frequency h_i
   - absolute cumulative frequency N_i
   - relative cumulative frequency H_i

b. Plot the absolute frequencies in a bar chart.

c. Plot the relative frequencies in a bar chart. What is different between the two charts.

Solution:

rating	n_i	h_i	N_i	H_i
u	4	0.16	4	0.16
a	11	0.44	15	0.60
s	10	0.40	25	1.00



Variable types

A comparison of PC-monitors considers the following variables:

- Manufacturer: nominal
- type of screen: nominal
- diagonal dimension (inches): continuous
- response time (miliseconds): continuous
- connections (e.g. HDMI): nominal
- number of USB ports: discrete
- refresh rate (Hz): continuous
- built in speakers (yes /no): nominal/binary
- features: nominal
- colour: nominal
- overall customer satisfaction (1 to 5 stars): ordinal

Assign the following data types to these variables: *nominal*, *binary*, *ordinal*, *discrete* or *continuous*.