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

Exercise 05: Jointly summarising two variables

Statistical Computing – WiSe 2022/23

The exercises today have fewer commands for you to blindly type in, than in previous weeks. For aspects covered in previous workshops/lectures consult the relevant teaching material. In other places hints are given. At the end there is a written exercise for you to practice calculating the correlation.

Preparations

a. Make sure you work inside your RStudio project. Details please see Exercise 01.

b. If not done until now, install the R package `carDate` by running the following code

```
install.packages("carData")
```

Another possibility to install packages is via the *Packages* tab in the lower right pane of RStudio.

c. Open a new R script file (either `Shift + Ctrl + N` or via the menu), type the following comment/code in the first few lines and save the file.

```
#============================================================================#
# Statistical Computing: exercise sheet 5                                    #
# Jointly summarising two variables                                          #
#============================================================================#
rm(list = ls(all.names = TRUE))
```

d. Explain once more(!) what the command `rm(list = ls(all.names = TRUE))` does and why the command is important at the beginning of a new R script.

Contingency tables

a. In the package `carData` there is a data frame called `TitanicSurvival`. Load the package and the data.

```
library(carData)
data("TitanicSurvival")
```

b. Read the documentation for the `TitanicSurvival` dataset[1].

```
help("TitanicSurvival")
```

c. Apply the `summary` command to the data set and check, if the data looks plausible?

```
summary(TitanicSurvival)

# discussion:
# - data looks reasonable
# - missing values for `age`
```

d. Obtain a frequency table to find out how many passengers survived and how many died. Determine also the relative frequencies.

```
# absolute frequencies
table(TitanicSurvival$survived)
# relative frequencies
prop.table(table((TitanicSurvival$survived)))
```

e. Obtain a contingency table with the frequencies for `survived` and `passengerClass`. Use the `dnn` argument for `table()` to label the output appropriate.

```
# If you assign the outcome of `table` to a new object, you can use this
# object for other calculations later on.
freqTab <- table(TitanicSurvival$survived,
                 TitanicSurvival$passengerClass,
                 dnn = c("survived", "Passenger Class"))
# Now have a look a the calculated frequencies:
freqTab
```

f. Calculate he overall relative frequencies, the column and row relative frequencies for passenger survival and class. Hint: see lecture notes and exercise sheet.
Round the values to 3 digits.

```
# If you store the number of digits you want to see in an object, it is
# easier to change it afterwards globally for all output
nDigits <- 3

# relative frequencies
round(prop.table(freqTab), digits = nDigits)

# column relative frequencies
```

[1] NB: There are several data sets with very similar Titanic data. This one has data on the passengers only, others include data on the crew as well.

```
round(prop.table(freqTab, margin = 2), digits = nDigits)

# row relative frequencies
round(prop.table(freqTab, margin = 1), digits = nDigits)
```

g. Use the results from the commands above to answer the following questions.

- What proportion of passengers were 1st class and survived? Solution: 0.153
- Of the 1st class passengers what proportion survived? Solution: 0.619
- Of the 3rd class passengers what proportion survived? Solution: 0.255
- Of those passengers who survived what proportion were 3rd class passengers? Solution: 0.362
  Note that (c) and (d) are not the same thing.

Bar charts

Note: Diagrams always have a title and axis labels, so that a discussion of the relationships shown is possible from the diagram alone. If the graph contains the information of further variables in addition to the X and Y values, it requires a legend (see below).

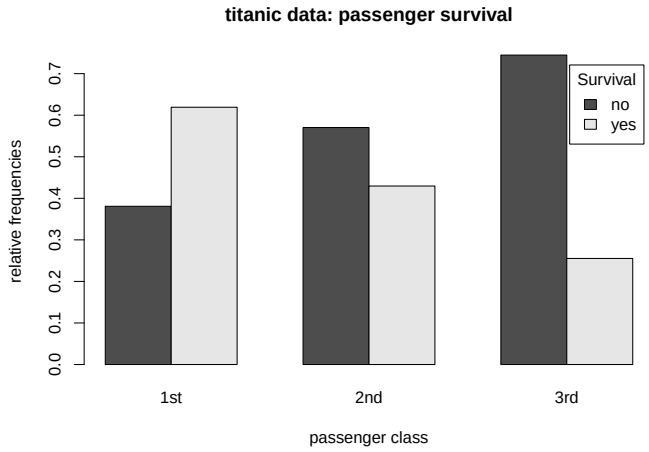
a. Plot a bar chart of passenger survival. Hint `barplot(table(???))`

b. Plot a bar chart of passenger survival and class. `barplot(table(???,???))`

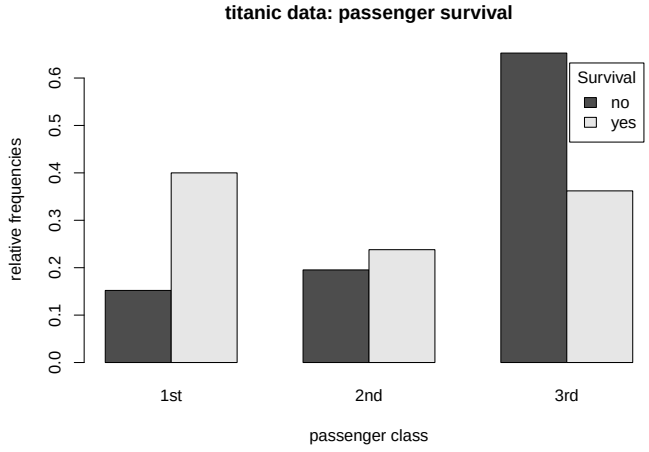
c. Bar plot accepts the arguments `beside=TRUE` and `legend=TRUE`. Investigate the effect of these two arguments.

d. What effect does reversing the order of `survived` and `passengerClass` in `table()` have on the diagram?

e. You want to visualise the relative survival within each passenger class as a bar chart. Create the following diagram:



f. Create the following bar chart and describe the information depicted with your own words.
Hint: the proportions for *survived* passengers sum up to 1 as well as the proportions for *non-survived* passengers do.



```r
#=============================================================================#
# RUN THE CODE and inspect the results                                        #
#=============================================================================#

# load the data or recreate `freqTab` if required
# more on control structures later in this course
if(!exists("TitanicSurvival") || !exists("freqTab")){
  data("TitanicSurvival")
  freqTab <- table(TitanicSurvival$survived,
                   TitanicSurvival$passengerClass)
}

# exercise A2.a:
barplot(table(TitanicSurvival$survived),
        main = "titanic data: passenger survival",
        xlab = "survival",
        ylab = "absolute frequencies")

# exercise A2.b:
# Notice that the color coding in the plot cannot be interpreted without legend
barplot(freqTab,
        main = "titanic data: passenger survival",
        xlab = "passenger class",
        ylab = "absolute frequencies")

# exercise A2.c:
# Notice that it is not so clear to which variable the `yes`/`no` color
# coding in the legend corresponds to
barplot(freqTab,
        beside = TRUE,
        legend.text = TRUE,
        main = "titanic data: passenger survival",
        xlab = "passenger class",
        ylab = "absolute frequencies")

# better: usage of a legend title via `args.legend = list(title = "survival")`
barplot(freqTab,
        beside = TRUE,
        legend.text = TRUE,
        args.legend = list(title = "survival"),
        main = "titanic data: passenger survival",
        xlab = "passenger class",
        ylab = "absolute frequencies")

# exercise A2.d:
# changing the order in `table()` changes the row and column values,
barplot(table(TitanicSurvival$passengerClass, TitanicSurvival$survived),
```

```
        beside = TRUE,
        legend.text = TRUE,
        args.legend = list(title = "passenger class"),
        main = "titanic data: passenger survival",
        xlab = "survival",
        ylab = "absolute frequencies")
```

```
# exercise A2.e:
barplot(prop.table(freqTab, margin = 2),
        beside = TRUE,
        legend.text = TRUE,
        args.legend = list(title = "Survival"),
        main = "titanic data: passenger survival",
        xlab = "passenger class",
        ylab = "relative frequencies")
```

```
# exercise A2.f:
# this bar chart shows the distribution of passengers over the passenger
# classes for the passenger groups `survived` and `dead` separately.
barplot(prop.table(freqTab, margin = 1),
        beside = TRUE,
        legend.text = TRUE,
        args.legend = list(title = "Survival"),
        main = "titanic data: passenger survival",
        xlab = "passenger class",
        ylab = "relative frequencies")
```

Jointly summarising a qualitative and a quantitative variable

You have already learnt a good graphical method for comparing a quantitative variable across different levels of a qualitative variable: a box plot.

a. Load the `Prestige` data set (also package `carData`) to the global environment and have another look to the data documentation if necessary.

```
data("Prestige", package = "carData")
```

b. Create a box plot for `income` split by `type`. Consider the remarks regarding plots in the beginning of exercise .

```
boxplot(income ~ type, data = Prestige,
        main = "Prestige data: income vs. type",
        ylab = "income US$",
        xlab = "type of occupation")
```

c. To obtain statistics for `income` split by `type` we use the function `tapply()` (see lecture notes). We can read the following call to the `tapply()` function as "apply the function `median()` to the variable `income` split by `type`".

```
tapply(Prestige$income, INDEX = Prestige$type, FUN = median)
```

Observe that the values for the median in each group matches with the box plot.

d. Many functions in R take a data variable as the first argument and then optional arguments to specify further settings. For example:

```
mean(Prestige$income)
mean(Prestige$income, trim = 0.1)
```

Read the help page for `mean` to find out what a trimmed mean is.

e. Run the following code and discuss the results.

```
# Applying the trimmed mean ignores the extreme values for `income` for
# occupation type 'prof'. Therefore the trimmed mean is not biased due the
# extreme values and much smaller and not so different compared to the
# median anymore.

# The trimmed mean for the other two occupation type doesn't differ that
# much from the non-trimmed mean and median, since there are no extreme
# values for those two groups.

# Trimming the mean is a way of robustification the mean against extreme
# values.
```

```
tapply(Prestige$income, INDEX = Prestige$type, FUN = mean)
tapply(Prestige$income, INDEX = Prestige$type, FUN = mean, trim = 0.1)
```

f. The function `quantile(x, probs = c(0.25, 0.75))` returns the lower and upper quartiles for the variable `x`. Obtain the lower and upper quartiles for `income` split by `type` using `tapply()`. Note the slightly different output format. Observe that the values for the quantile in each group matches the edges of each box in the box plot.

```
tapply(Prestige$income,
       INDEX = Prestige$type,
       FUN = quantile, probs = c(0.25, 0.75))
```

Jointly summarising two quantitative variables

a. You work again with the `Prestige` data set (also package `carData`).

```
data("Prestige", package = "carData")
```

b. Plot the two variables `education` and `prestige` in a scatter plot. Which of the two variables should you use for the x and which for the y axis?

```
# x-axis: `education` since this assumed to explain `prestige`
# y-axis: `prestige` since this depends presumably on `education`
```

c. If not done already, enhance your plot to a good plot considering the remarks at the beginning of .

```
plot(prestige ~ education,
     data = Prestige,
     main = "Prestige data: prestige vs. education",
     xlab = "education (years)",
     ylab = "prestige score")
```

d. Is there a linear relationship between these two variables? If yes,

- is it positive or negative?
- is it strong, weak or middling?
- have a guess at estimating the correlation coefficient before you calculate it.

```
# A strong positive correlation is obvious.
```

e. The covariance and (Pearson's) correlation between two variables is found using the functions `cov()` and `cor()` respectively. Obtain the covariance and correlation for `education` and `prestige`. Calculate the correlation in addition using the covariance and the standard deviations.

```
# covariance
cov(Prestige$prestige, Prestige$education)
# correlation
cor(Prestige$prestige, Prestige$education)
# correlation by hand
cov(Prestige$prestige, Prestige$education) /
  (sd(Prestige$prestige) * sd(Prestige$education))
```

f. Plot a scatter plot of `education` and `income` and obtain the Pearson correlation for these two. Notice that there are a couple of outliers in the variable `income`.

g. Is your scatter plot for `education` and `income` informative enough? Does your choice of x and y variable reflect the concept of explainatory and dependent variable?

```
plot(income ~ education,
     data = Prestige,
     main = "Prestige data: income vs. education",
     xlab = "education (years)",
     ylab = "income (US$)")
```

```
cor(Prestige$income, Prestige$education)
```

h. The function `cor()` takes an argument `method = "spearman"`, to calculate Spearman's rank correlation coefficient. Find this for `education` and `income`. What do you conclude?

```
# The two types of correlation coefficient give almost he same value.
# It makes no practical difference which statistic we choose.
cor(Prestige$income, Prestige$education, method = "spearman")
```

Missing values

In the `TitanicSurvival` data set, there is a variable called `age`.

a. Make sure that the Titanic data is still available in the global environment.

b. Calculate the mean age of the Titanic passengers. Is there a problem? What is the problem?

c. Many of the R functions, which return a statistic, accept an argument called `na.rm`. The default value is`FALSE`, but if set to `TRUE`, the missing values are simply ignored. To obtain the mean age for those passengers with known age, use

```
mean(TitanicSurvival$age, na.rm = TRUE)
```

d. Find the mean age of the passengers who survived and who died, using `tapply`.

```
tapply(TitanicSurvival$age,
       INDEX = TitanicSurvival$survived,
       mean, na.rm = TRUE)
```

e. Find the mean age of the passengers who survived and who died within each passenger class by altering the `INDEX` argument of `tapply`. Investigate the following example from the `tapply` help page, to solve this task.

```
ind <- list(c(1, 2, 2), c("A", "A", "B"))
table(ind)
tapply(1:3, ind) #-> the split vector
tapply(1:3, ind, sum)
```

```
tapply(TitanicSurvival$age,
       INDEX = list(TitanicSurvival$survived,
                    TitanicSurvival$passengerClass),
       mean, na.rm = TRUE)
```

f. Now you are interested in a new variable carrying the information if the `age` information is missing. Create this new variable via

```
TitanicSurvival$ageMissing <- is.na(TitanicSurvival$age)
```

g. Create a contingency table considering `ageMissing` and `passengerClass` and calculate the proportion of missing ages within each passenger class. Discuss the results!

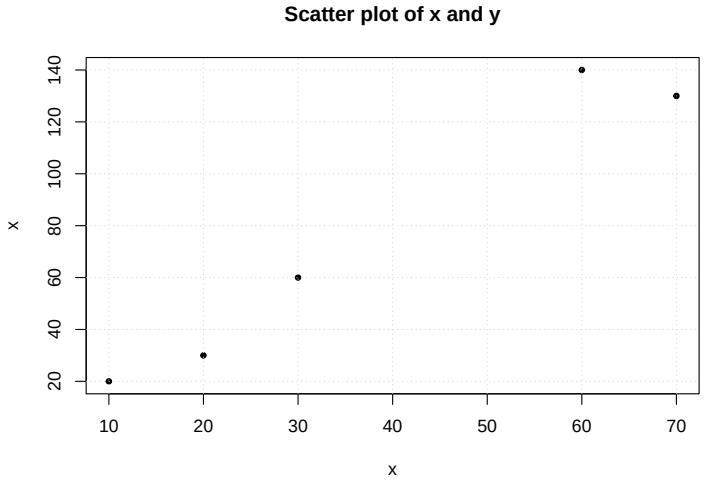
```
# The proportion of 3rd class passengers whose ages are unknown is much
# higher (0.29) than the proportion of other passengers whose ages
# are unknown
prop.table(
  table(TitanicSurvival$ageMissing, TitanicSurvival$passengerClass),
  margin = 2)
```

Warning! Be very careful when just ignoring missing values. If the *missingness* is dependent on another variable, you can easily bias the results of an analysis.

Written exercise: Calculating the correlation coefficient

For the following two (small!) samples X and Y, plot the variables in a scatter plot (using pencil and paper).

x	y
10	20
60	140
70	130
20	30
30	60



a. Is there a linear relationship between these two variables? If yes,
   - is it positive or negative?
   - is it strong, weak or middling?
   - have a guess at estimating the correlation coefficient before you calculate it.

b. Calculate the mean, variance and standard deviation of X and Y and the covariance and correlation coefficient for X and Y.
   Use the following table to help with the calculations.

Solution

With $\overline{x} = 38$ and $\overline{y} = 76$ one obtains

x	y	$(x_i - \overline{x})$	$(x_i - \overline{x})^2$	$(y_i - \overline{y})$	$(y_i - \overline{y})^2$	$(x_i - \overline{x})(y_i - \overline{y})$
10	20	-28	784	-56	3136	-87808
60	140	22	484	64	4096	90112
70	130	32	1024	54	2916	93312
20	30	-18	324	-46	2116	-38088
30	60	-8	64	-16	256	-2048

resulting in

$$s_{x^2} = \frac{1}{5-1}\sum_{i=1}^{5}(x_i - \overline{x})^2 = \frac{2680}{4} = 670$$

$$s_x = \sqrt{s_{x^2}} = 25.88$$

$$s_{y^2} = \frac{1}{5-1}\sum_{i=1}^{5}(y_i - \overline{y})^2 = \frac{2680}{4} = 3130$$

$$s_y = \sqrt{s_{y^2}} = 55.95$$

$$s_{x,\,y} = \frac{1}{5-1}\sum_{i=1}^{5}(x_i - \overline{x})(y_i - \overline{y}) = \frac{55480}{4} = 1415$$

$$r_{x,\,y} = \frac{s_{x,\,y}}{s_x s_y} = 0.98$$