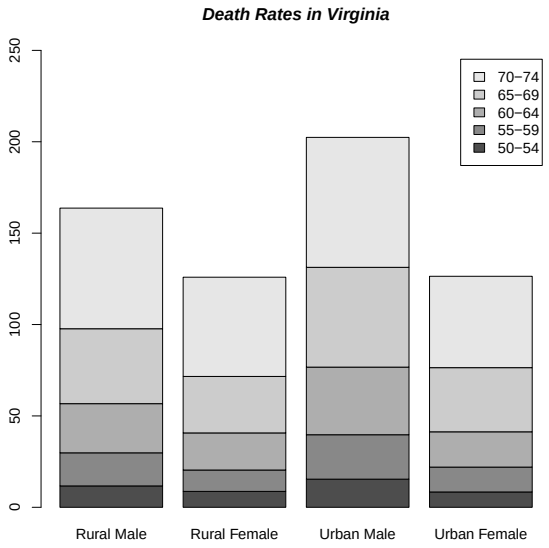
Exercises for the course

"An introduction to R"

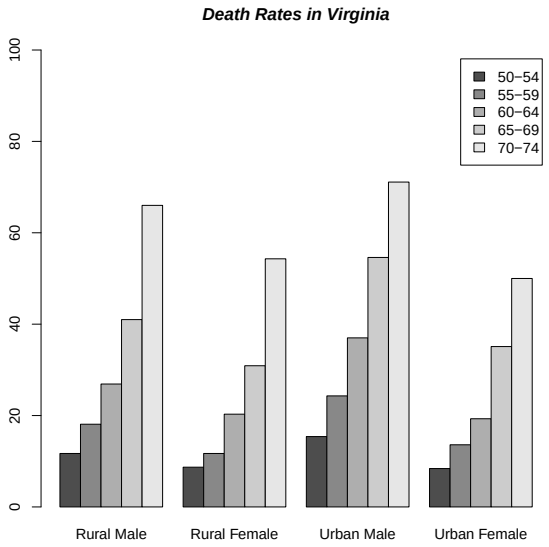
Sheet 08

Exercise 42: *Barplots*

- Download `deaths.txt` from the web page. This file contains the death rates per 1000 in Virginia in 1940. The rates are classified by age group and population group (rural-male, rural-femail, urban-male, urban-female). Read this data into a data frame `deaths` and find out the column names.
- We are going to use the column names in a plot and don't want the dots in the names. Read the data again in. This time the column names shall contain no dots. Hint: `check.names=`.
- Visualize the data with `barplot`. The command `barplot` cannot handle data frames, only matrices. Therefore you need to apply `barplot` to the matrix `as.matrix(deaths)`. Your result should resemble the following figure. Hint: Use the rownames for the legend.



- Visualize the data with `barplot` once more. This time plot the the population groups horizontally next to each other. Your result should resemble the following figure. Hint: `beside=`.



Exercise 43: *Regular expressions*
Assign the object returned by `data()` as `dat`. What is the class of `dat`? What are the names of the elements of `dat`? One element is a matrix with name `results`. Denote the column 'Item' of `results` as `s`. Find all elements of `s` which

- contain "men"
- contain "air"
- start with "euro"
- have an 'a', 'c', 'g' or 't' at the fourth position
- contain both "sect" and "pray".

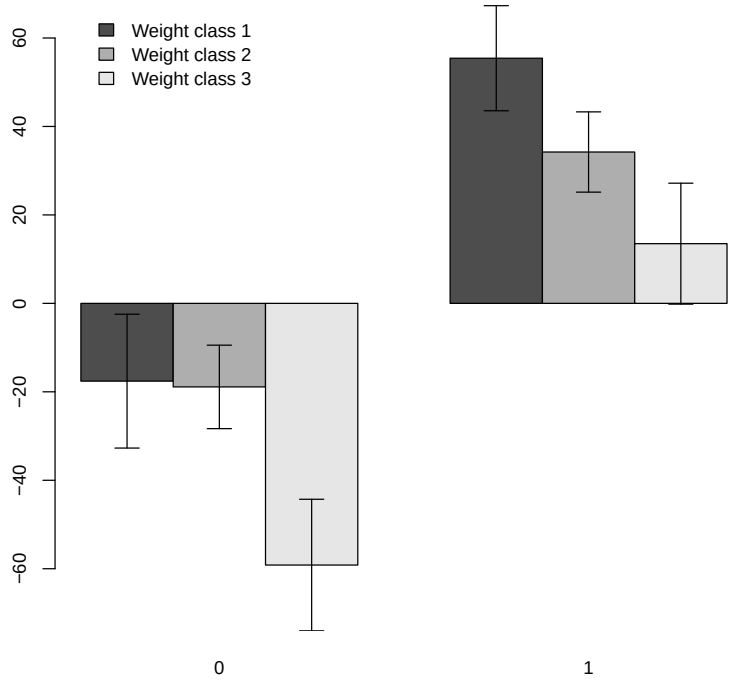
Exercise 44: Sometimes one needs to deal with a lot of files which are numbered consecutively. Here is some practice. Write a for-loop with 100 cycles. In cycle 67, the integer 67^2 is written into the file 'testfile67.txt' using the command `write.table()`. Similarly for other cylces but with 67 replaced by the respective cylce index. Having done that create a vector `vec` consisting of one hundred zeros. Then write a second for-loop with 100 cycles as follows. In cyle 67 the content of 'testfile67.txt' is read into a data frame variable. Extract 67^2 from that data frame and store it into `vec[67]`. Similarly for other cycles. In the end the vector `vec` should be equal to `(1:100)^2`.

Exercise 45: *Reliability of the t-test for non-normally distributed and dependent samples*

- In contrast to the bell-shaped normal distribution, we now consider a U-shaped distribution. Here we choose `dbeta(,0.5,0.5)`. Look at the density hereof by plotting `dbeta(x,0.5,0.5)` as a function of `x<-seq(from=0,to=1,by=0.01)`. Convince yourself that the mean of this distribution is 0.5.
- Repeat Exercise 40 with `rnorm(100)` being replaced by `rbeta(100,0.5,0.5)` and with the null hypothesis being now $H_0 : \mu = 0.5$ (again 10.000 loops and confidence level 5%. Also try different sample lengths. What is the result now?

- So far we used samples of independent values. Now we use a dependent sample. Execute `sample(rnorm(10),20,replace=TRUE)` and convince yourself that this command produces a sample of dependent standard normally distributed values. The true mean of the underlying distribution is 0. Repeat Exercise 40 with `rnorm(100)` being replaced by `sample(rnorm(10),20,replace=TRUE)` and with the null hypothesis being $H_0 : \mu = 0$ (again 10.000 loops and confidence level 5%. Also try different sample lengths. What is the result now?

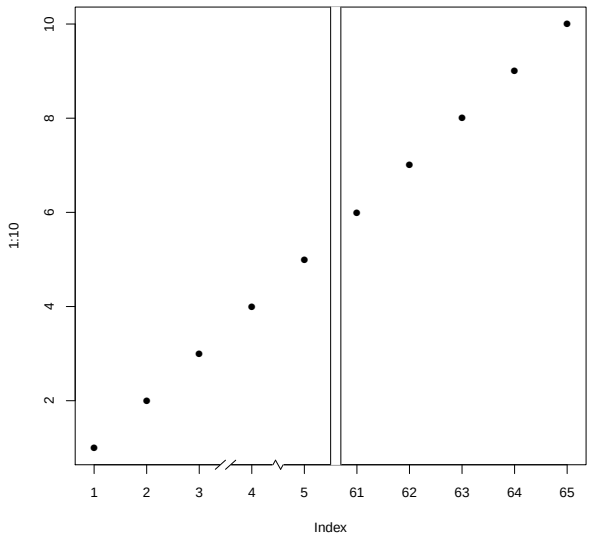
Exercise 46: *How to produce barplots with error bars*
Install and load the library `sciplot`. Look at the help page `?bargraph.CI` and the example with `ToothGrowth` at the end of the help page. Then load the data `heartbeats.txt` from the web page. Use `bargraph.CI` from the library `sciplot` to barplot `weightincr` for every `treatment` and grouped according to `wghtcls`. Your result should resemble the following figure.



Then make a complete figure out this (what information is missing so far?). Hints: Change the location of the legend with `x.leg=1` and change the labels of the legend.

Exercise 47: *How to produce gaps on the axis*
`axis.break()` and `gap.plot()` are commands in the library `plotrix`. If you cannot load this library, then install it first.

- Use `axis.break` to produce a plot similar to the following plot. (The data vector is 1:10).



- Execute the commands

```
set.seed(1111)
twogrp.x <- c( rnorm(10)+1,rnorm(10)+200 )
twogrp.y <- c( rnorm(10,sd=0.5)+4,rnorm(10)+5 )
```

Plot `twogrp.y` against `twogrp.x` to see how it looks like without gap. Then use `gap.plot()` to produce a plot similar to the following plot.