Exercises for the course

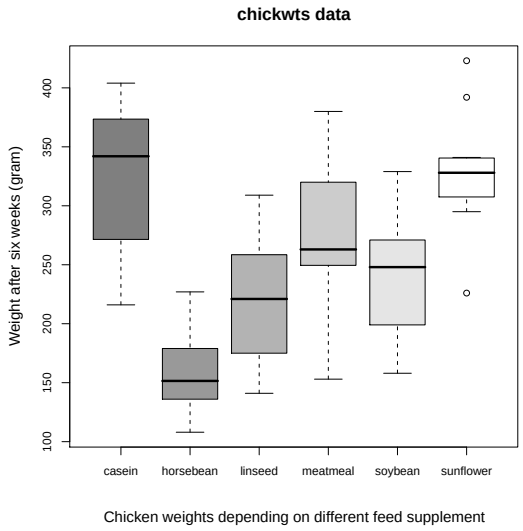
"An introduction to R"

Sheet 03

Exercise 13: *The normal distribution is very important. So let's study it.*

- Plot the density of the (standard) normal distribution between -3 and 3.
- In statistics, null hypotheses are rejected if a certain probability is below 5%. Thus it is important to know which centered interval supports 95% of the mass of the normal distribution. Find the value x such that `pnorm(x)-pnorm(-x)`$= 0.95$. Recall the symmetry of the (standard) normal density and explain why this value x is equal to `qnorm(0.975)`. Round the value x to 3 significant digits and remember this number.
- Sample 1000 random values from the standard normal distribution and denote the vector of these values as `x`. Calculate the mean, the variance, the standard deviation and the quartiles of this vector. Then visualize the quartiles of `x` with a boxplot. Finally plot the histogram and the empirical distribution function of `x`.

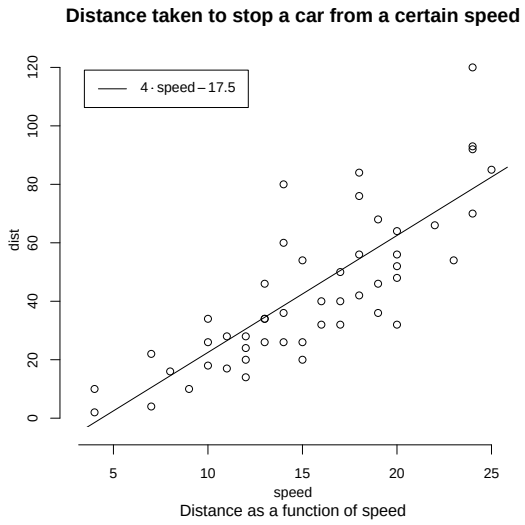
Exercise 14: *We learned that there are many pre-defined data sets you can use in R. The data frame* `chickwts` *is one of those. It contains measured growth rates of chickens in dependence of various feed supplements.*
If `chickwts` is not known, then say `data(chickwts)` in order to load it. Produce the following figure:



Use the colors 'grey50', 'grey60', 'grey70', 'grey80', 'grey90' and 'grey100'. The main title is magnified with 1.5, the subtitle and the label of the y-axis are magnified with 1.3.

Exercise 15: *The data frame* `cars` *gives the distance taken to stop from a certain speed. Note that the data were recorded in the 1920s.* If `cars` is not known, then say `data(chickwts)` in order to load it. Plot `dist` as a function of `speed` without titles and without axes (`axes=FALSE`). Add a line with slope 4 and intercept -17.5. Furthermore add axes with the command `axis()`. Specify

the positions of the ticks with `at=`. 'Pretty' positions of the ticks are generated with `pretty(speed)` and `pretty(dist)`, respectively. You might want to draw the axis into the margins by one line (`line=1`). Add a legend with `legend()`. The command locator(1) might help you to find a good position for the legend. The text in the legend is generated with the command `expression()`. Furthermore add a main title and a subtitle with the command `title()`.



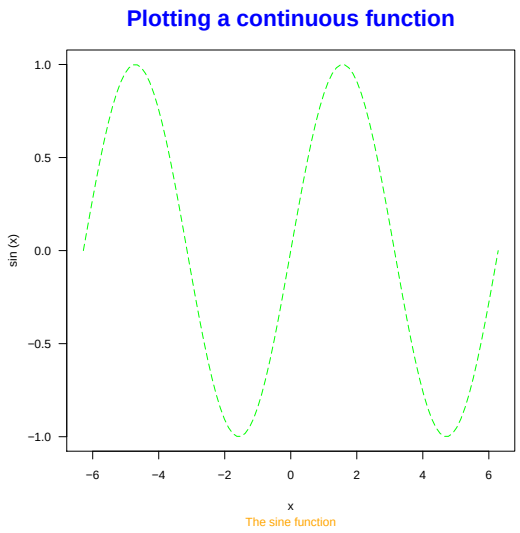
Exercise 16: Before reading data, you need to carefully prepare the data file. Consider the following self-generated data which is supposed to be obtained from measuring mice of the two species 'mus musculus' and 'mus spretus'. The following table contains the tail length measured in centimeter.

mus musculus	11,3	10,6	12,1	9,5	11,7	10,8
mus spretus	5,4	5,6	6,1	4,9	5,3	4,6

Prepare a data file as described in Subsection 3.5 of the script using Excel or a text editor. Then read the data file into a data frame.

- Calculate the mean tail length of all mice.
- Calculate the mean tail length of all mice of the species mus musculus.
- Calculate the mean tail length of all mice of the species mus spretus.

Exercise 17: Use `plot()` to produce the following plot of the sine function on the interval $[-2\cdot\pi, 2\cdot\pi]$:



The text of the main title (`main=`) is magnified (`cex.main=`) by a factor 2. The line type is 'longdash'. The colours used in the plot are 'blue', 'green' and 'orange'. Furthermore you need the option `las=` to obtain horizontal axis labels.

Exercise 18: *This exercise is to acquaint you with multi-figures and margins. Multi-figures are quite useful to illustrate different features of data.* Enter the command `op <- par(mfrow=c(2,1), oma=c(2,3,1,0)`. Subsequent figures will be drawn in a 2 by 1 multi-figure and the outer margins are as follows: 2 lines at the bottom, 3 lines on the left-hand side, 1 line on top and no outer margin on the right-hand side ((bottom,left,top,right)=(2,3,1,0)). The setting before this parameter change is stored into the variable `op`. Now enter a plot command with `plot()` which plots the sine-function between -2π and 2π. Then plot the cosine function. Then plot the tangent-function between -1.5 and 1.5. What happend to the plots? Recall that every call of `plot.new()` starts a new plot and every high-level plotting command such as `plot()` calls `plot.new()`. In a multi-figure `plot.new()` causes an advance to the next plotting region. Have a try: Enter `plot.new()` and then plot again the sine-function. What happend to the plot? Now restore the old parameter setting by entering `par(op)`. Plot the sine-function again to check that.
Now let's see where low-level potting commands go to in a multi-figure. Enter the command `par(mfrow=c(2,1),oma=c(2,3,1,0)`. Plot the sine-function and add a main title with `title()`. Then plot the cosine-function and add a main title with `title()`. Explain the behaviour of the two calls to `title()`. Next we wish to add text to the sine-plot. For this specify with `par(mfg=c(1,1))` that the next low-level plotting commands affect the figure on top. Add the text 'sine' at a suitable position. Then change to the cosine-plot and add 'cosine' at a suitable position.
Finally mark the margins and the plotting area. Change back to the sine-plot with `mfg=`. The plotting area is surrounded by a green box by entering `box(which="plot",col="green")`. The figure area is surrounded by a turquoise box by entering `box(which="figure",col="turquoise")`. Then enter `box(which="inner",col="blue")` and `box(which="outer",col="red")` to mark the boundaries of the inner and of the outer margins, respectively. Use `mtext()` to write text into the margins. The command `mtext("The sine- and the cosine-function",side=2,line=1,cex=2)` produces the text on the left-hand side. Find the command which produces the text on the right-hand side.
Having finished the multiplot, restore the parameter setting with `par(op)`.

The sine− and the cosine−function
The sine−function
sin (x)
1.0
0.0
−1.0
sine
−6
−4
−2
0
2
4
6
x
The cosine−function
cos (x)
1.0
0.0
−1.0
cosine
−6
−4
−2
0
2
4
6
x
Inner margin