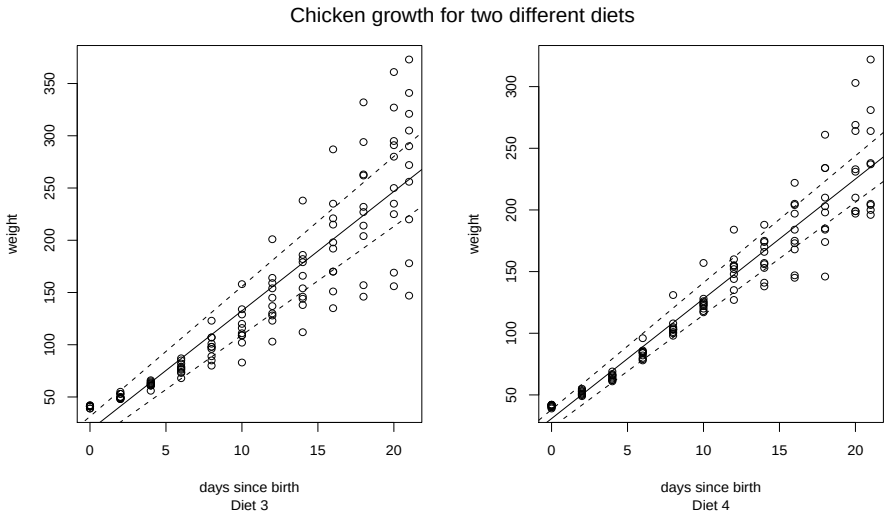
Exercises for the course

"An introduction to R"

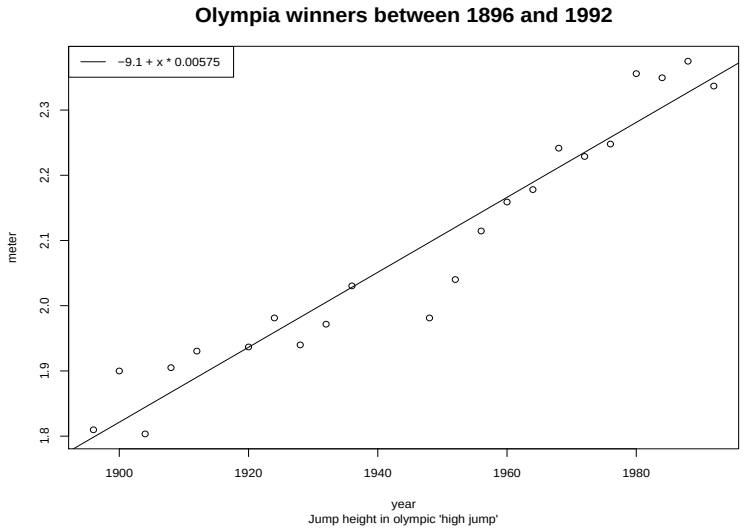
Sheet 09

Exercise 48: Recall the ChickWeight data. We consider only Diet 3 and Diet 4 here. Store the weight vector corresponding to Diet 3 into `w3`. Store the time vector corresponding to Diet 3 into `t3`. Find the linear model which best explains `w3` as function of `t3`. Furthermore extract the confidence intervals for the regression line. The corresponding lines are added to the figure below with line type "dashed". Proceed similarly for Diet 4. What are the values of r squared of the two linear regressions. Produce a picture which resembles the following figure.



Hint: The main title is magnified with 1.5. The ratio between the axis can be changed by using the mouse to resize the plotting window.

Exercise 49: The file "olympics.meter.txt" contains the jump height and the leap distance of the olympia winners between 1896 and 1992 in the disciplines 'high jump' and 'long jump'. All measurements are given in the unit 'meter'. Plot the data points for the discipline 'high jump' against the years (true years, not years since 1900) and add the regression line. Your result should resemble the following figure.



Hints: The main title is magnified with factor 1.8. The values in the legend are rounded to 3 significant digits. It is useful to first define a vector of the true years.

Exercise 50: Define `cols <- colors()` to be the vector of color names. Using regular expressions find the following elements of `cols`:

- All color names which contain "yellow".
- All colors whose name starts with "yellow".
- All colors whose name starts with "yellow" or with "orange".
- All color names whose third character is 'c' (This is not interesting for color names. However when looking at DNA data one might want to find sequences with a certain polymorphism at a given locus).

Exercise 51: *Using regular expressions to identify mRNA*
Download the file `sequences.txt` from the web page. Use the command `readLines()` to read the lines of the file as strings into the vector `sequences` of strings. We want to find out which of the sequences could be mRNA. Identify the sequences which satisfy:

- Contains a ribosomal binding site just before the start codon with the consensus sequence: GCC(G or A)CC
- Immediately followed by a start codon AUG
- An arbitrary number of codons (a codon has exactly three letters)
- A stop codon, that is, one of the sequences UAG, UGA or UAA

The output of the command also tells you where such a pattern occurs the first time and the length of the matches. (This exercise is adapted from the Pearl course of Stephan Hutter).

Exercise 52: The sample mean is sensitive to outliers, e.g. `mean(1:4)` is 2.5 whereas `mean(c(1:4,1000))` is 202. To reduce the impact of outliers write a function `trim.mean()` which removes the smallest and the largest element of its argument and returns the mean of the remaining vector. Assume that the argument is numeric and contains no `NA`s. Here is how it should work:

```
> trim.mean(c(-100,1:4,1000))
[1] 2.5
> mean(1:4)
[1] 2.5
```

Exercise 53: Define a class 'species'. Each object of that class shall have the following three elements: a species *name*, a sample *dna* sequence and a variable which indicates whether the species is *endangered* or not. For simplicity let's use the S3 approach. Objects are created as lists and by setting the `class` attribute. Create a first object and let for simplicity "atcg" be the value of *dna*. Then define the print method for the class. Here is an example how the output of the print command could look like:

```
Species:        Elephant
DNA:            atcg
Endangered:     No
```

Hint: You need tabulators (\t) and newlines (\n) to produce such an output.