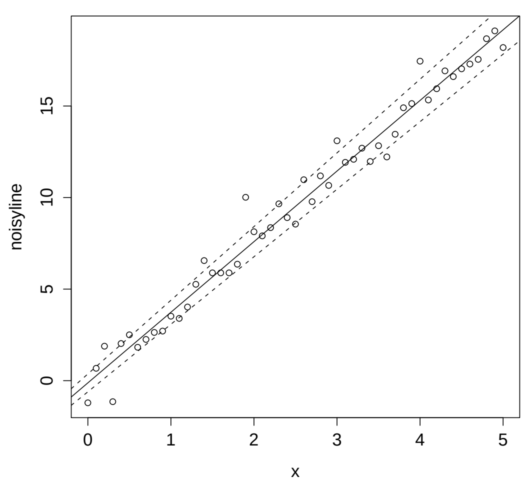
Exercises for the course

"An introduction to R"

Sheet 08

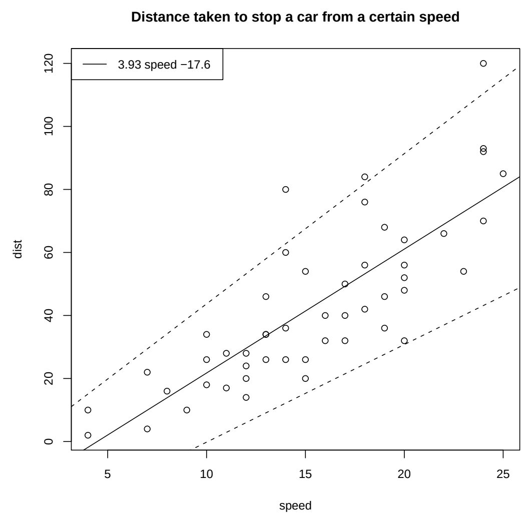
Exercise 37: Recall the Chill Coma Recovery Time (CCRT) data set from Exercise 28. Load the data into the variable `data.ccrt` and copy it into the search path. Calculate the sample mean and the sample standard deviation of `ccrt`. Then caculate the sample mean and the sample standard deviation for the two subvectors of `ccrt` corresponding to flies from Bangkok and Kathmandu, respectively. Is the difference of these two means significantly different from zero? Choose a suitable test and justify its usage. Furthermore check with a one sample test that both sample means are significantly different from `mean(ccrt)`. Finally calculate by hand the 95% confidence interval for the true mean of the CCRT of the Bangkok population and the 95% confidence interval for the true mean of the CCRT of the Kathmandu population. (4 points)

Exercise 38: Set the seed to 1234 to get the same picture. Define two vectors `x <- seq(from=0,to=5,by=0.1)` and `noisyline <- 2*x+4 + rnorm(length(x))`. Explain the variable `noisyline` with a linear model of the independent variable `x`. Use the command `lm()` for this. Denote the returned object as `regr`. What is the Anova table of this regression? Read off the intercept and the slope of the fitted line with `coef()`. Extract the p-values of the intercept and of the slope from `summary(regr)`. What is the fraction of the total variation in `noisyline` that is explained by the regression? This fraction is called 'r squared' and is printed by `summary(regr)` under 'Multiple R-squared'. Check that this value is equal to $\big(\mathrm{cor}(\mathrm{x}, \mathrm{noisyline})\big)^2$. In order to visualize the linear model, plot `noisyline` against `x`. Add the regression line with `abline(regr)`. Next calculate confidence intervals of the fitted parameters with `confint()` applied to `regr`. Denote the object returned by this command as `cf`. Enter `cf` to view it. The two columns of `cf` define two lines. Add these two lines to the plot with line type "dashed". Your result should resemble the following figure.

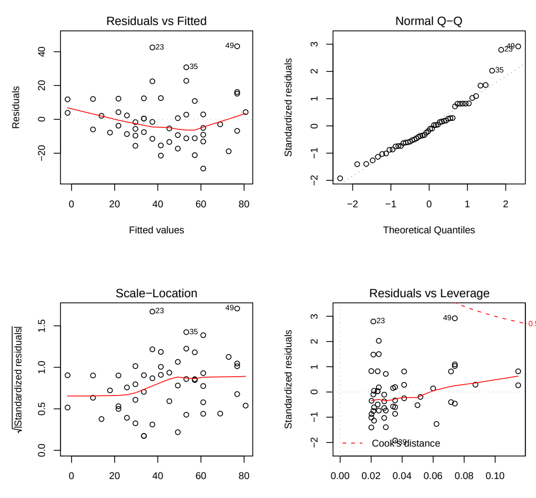


(4 points)

Exercise 39: Recall the data set `cars` from Exercise 21. Plot `speed` as a function of `dist`. The plot suggests that `speed` depends linearly on `dist`. Find this linear dependence with a linear regression in which `dist` is the response variable. Store the returned regression object into the variable `regr`. What is the Anova table of this regression? Read off the p-values of the intercept and of the slope from `summary(regr)`. What is the fraction of the total variation in `dist` that is explained by the regression, that is, what is r-squared? Then add the regression line to the plot. Extract intercept and slope of the regression line from `regr` and round both values to 3 significant digits. These values are used for the legend of the following figure. In addition add the two lines which you get from the confidence intervals for the intercept and for the slope of the regression line.



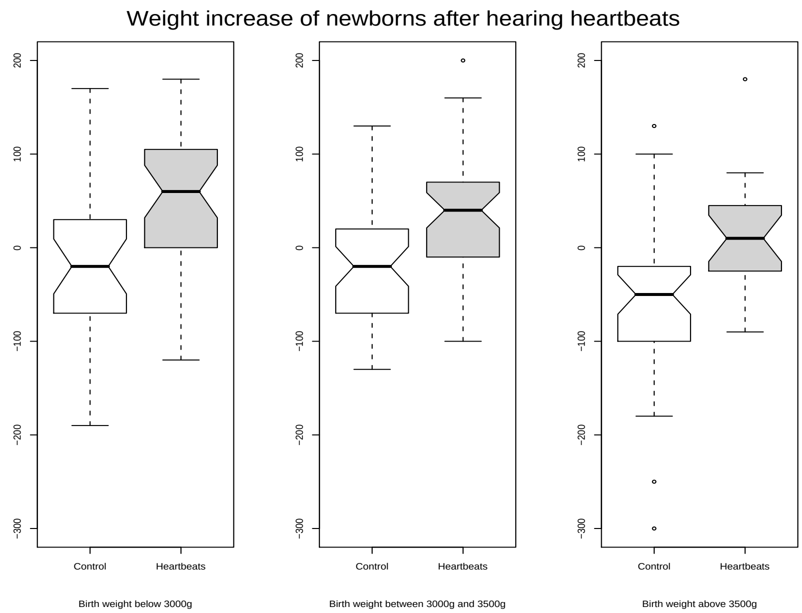
Finally start a 2 by 2 multi-figure and enter `plot(regr)` to obtain the following 4 figures which can be used to check the linear model. (4 points)



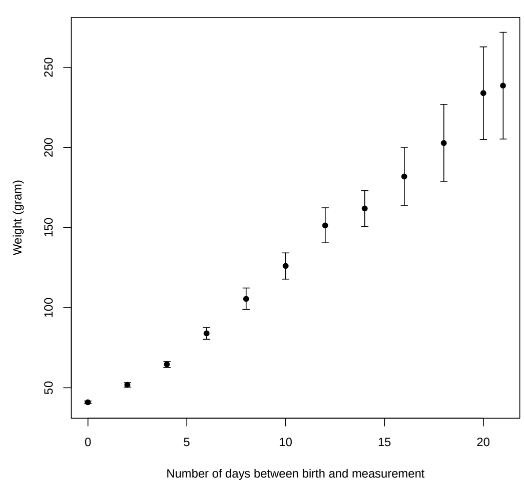
Exercise 40: Recall `heartbeats` from Exercise 12. Produce a picture which resembles the multi-figure below. Hints: One way to boxplot `wghtincr` as a function of `treatment` for each weight class is as follows. Split `heartbeats` according to `wghtcls` and denote the resulting list of data frames as `L`. Then use the command

```
boxplot(weight~treatment, data=L$'1',ylim=c(-300,200),
        col=c("white","lightgrey"), ...)
```

to produce the left boxplot and adapt the above command to produce the other boxplots. The option `ylim=c(-300,200)` ensures that all y-axes have the same range. Moreover the main title is magnified with factor 1.5. You can change the ratio of height and width of your multi-figure by using the mouse to change the plotting window. (3 points)



Exercise 41: Recall the ChickWeight data from Exercise 26. Define a subvector `weight4` of `weight` corresponding to Diet 4. As in Exercise 26 calculate the mean of `weight4` for each day. In addition calculate the confidence intervals for these means. Represent the confidence intervals through vectors `top4` and `bot4` which contain upper and lower interval boundaries. Plot the vector of means and the confidence intervals using the command `errbar()` from the library `sfsmisc`. Your result should resemble the following figure.



Hint: The point character used for `errbar()` is 16 in this plot. (4 points)