Exercises for the course

"An introduction to R"

Exercise session 4: Friday, March 7 2013

Exercise 4.1: *The role of the heartbeat in the relation between mother and infant.*
Download the file 'heartbeats.txt' from the course web page. Read the data from 'heartbeats.txt' into a data frame named `heartbeats`. Get an overview of the structure of the data frame. The group of newborns which had the heartbeat treatment is referred to as 'heartbeat group'. Calculate

- the mean of the increase in weight of all newborns,
- the mean of the weight increase of the control group,
- the mean of the weight increase in the heartbeat group,
- the mean of the weight increase in each weight class of the control group,
- the mean of the weight increase in each weight class of the heartbeat group,
- the standard deviation of the weight increase of all newborns.

Are the means in the heartbeat group higher than in the control group?

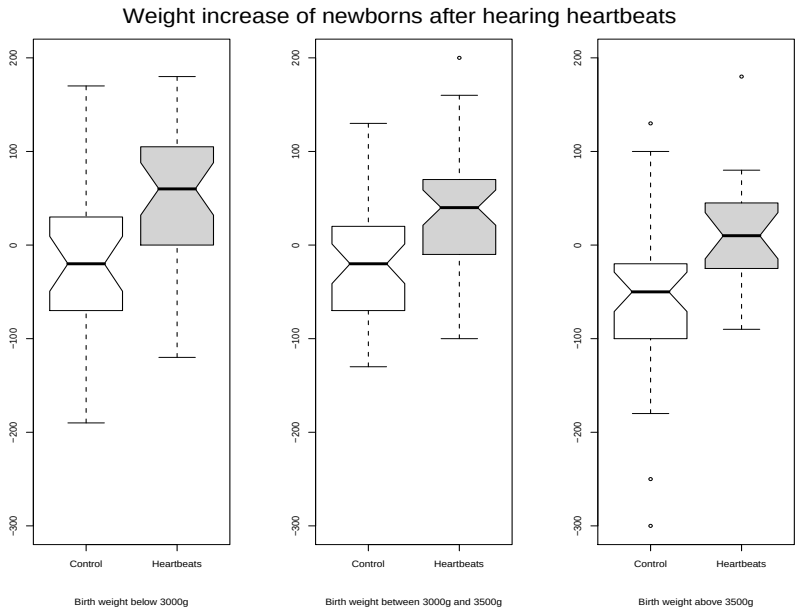
Newborns often loose weight in the first days. Let's have a closer look at that with the data at hand. Define a new vector `negwghtincr` as the elements of `wghtincr` which are ≤ 0.

- Boxplot `negwghtincr` both for the heartbeat group and for the control group.
- Plot an histogram of `negwghtincr` of the first weight class of the heartbeat group.
- Plot an histogram of `negwghtincr` of the first weight class of the control group.

Exercise 4.2: Recall `heartbeats` from Exercise 4.1. Produce a figure 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

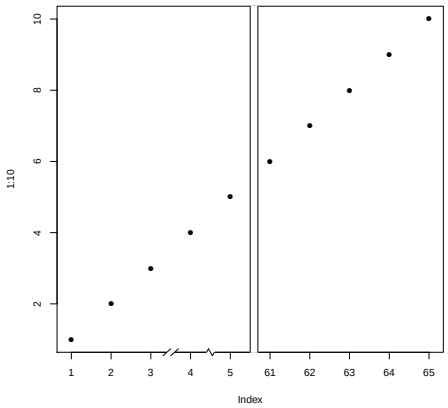
```
boxplot(weight~treatment, data=L'1',ylim=c(-300,200),
        col=c("white","lightgrey"), ...)
```

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.



Exercise 4.3: *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.

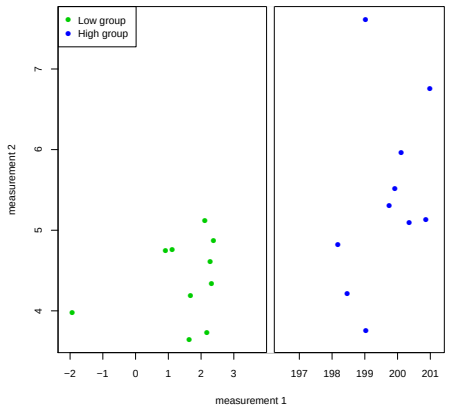
- Use `axis.break` to produce a plot similar to the following plot. First plot $1:10$ with `plot()` and with `xaxt=n`. Then add the x-axis with suitable ticks and add the axis breaks.



- Execute the commands

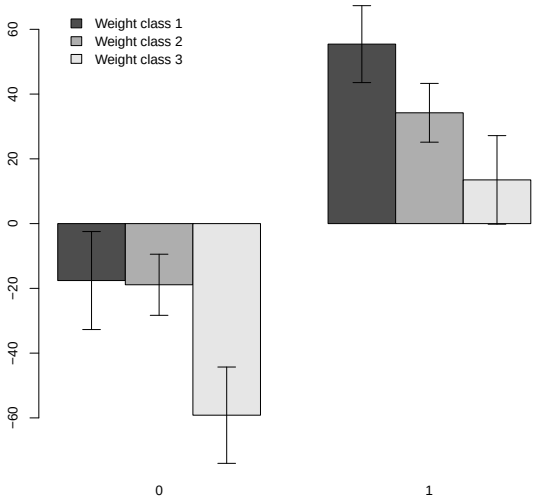
```
set.seed(1111)
twogrp.x <- c( rnorm(10)+1,rnorm(10)+200 )
twogrp.y <- c( rnorm(10,sd=0.5)+4,rnorm(10)+5 )
```

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.



Use the color 'green' for the points to the left of the axis break and 'blue' for the points to the right of the axis break.

Exercise 4.4: *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 call `heartbeats` again. 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 of 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 4.5: *Functions to handle NAs*
Write a function `which.NA()` which returns the vector of indices at which the function argument has `NA`s. Here is how it should work:

```
> which.NA(c(1,2,NA,7,NA,6))
[1] 3 5
```

Hint: `is.na()`.

Write a function `rm.NA()` which returns its argument without `NA`s.

```
> rm.NA(c(1,2,NA,7,NA,6))
[1] 1 2 7 6
```

Exercise 4.6: *The infinite loop*
Execute the following commands:

```
x <- 0
while (x < 1) {
x <- 0
}
```

Hint: Use Strg + C to stop the execution.
What happened?
Invent other cases of infinite loops.