Exercises for the course

# "An introduction to R"

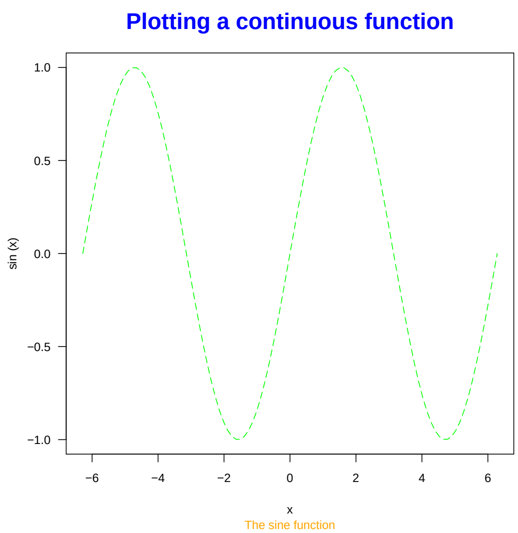
Sheet 04

**Exercise 15:** *Apply functions directly to vectors if possible. If this is not possible, then apply() and its relatives might help to avoid slow loops.*

- Use the command `lapply()` to apply the function `sqrt()` to each element of the vector `1:9`. Repeat this with `lapply()` replaced by `sapply()` and compare the output of the two commands. (Of course `sqrt(1:9)` does the same.)
- Define the variable `line` to be a vector which increases from 0 to 1 and has 1000 elements. Generate 1000 values with `rnorm()` and add `line`. Denote the resulting vector as `x`. Then generate a factor named `groups` with 10 levels and total length 1000 with the command `gl()`. Calculate the mean of `x` for each level in `groups`. Plot the resulting vector of means.

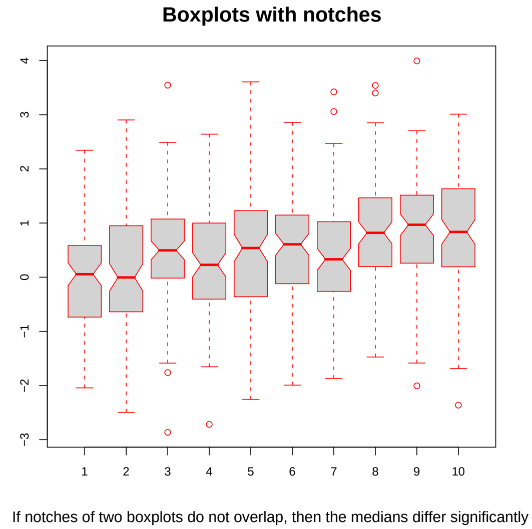
(3 points)

**Exercise 16:** 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. (4 points)

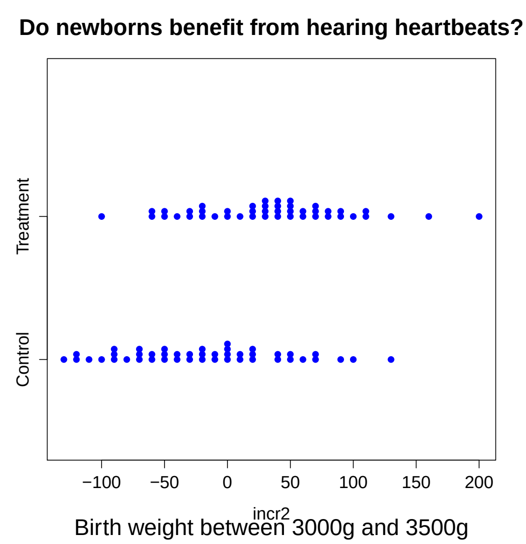
**Exercise 17:** Set the seed to 1111 to obtain the same plot as below. Define a vector `x` and a factor `groups` as in the last part of Exercise 15. Use either `plot()` or `boxplot()` to boxplot `x` grouped according to `groups`. Produce the following figure:




The text of the main title is magnified by 1.8, the text of the subtitle is magnified by 1.3. The colours used in the plot are 'red' for the border and 'lightgray' for the background.

(3 points)

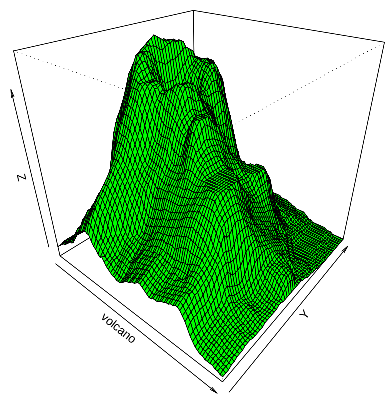
**Exercise 18:** *If the data size is small, then scatter plots produced by* `stripchart()` *are a good alternative to boxplots.*
Read the data from the file 'heartbeats.txt' (see web page) into a data frame named 'heartbeats' and attach the data frame. Here we only consider weight class '2'. Let `incr2` be the subvector of `wghtincr` of newborns of this weight class. Moreover let `treat2` be the subvector of `treatment` of all newborns of weight class '2'. Grouping `incr2` according to `treat2`, use `stripchart()` to produce the following figure.



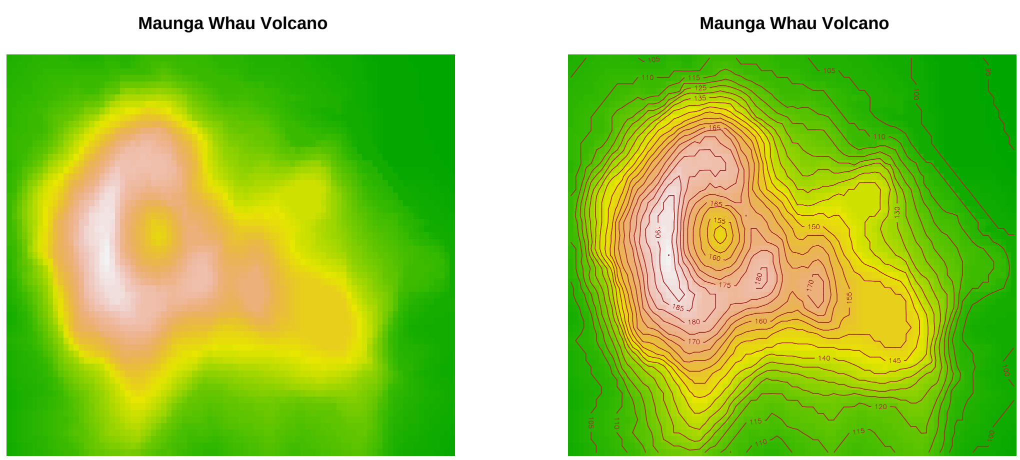

Here are hints. To avoid points being plotted at the same spot, let them be plotted on top of each other with `method="stack"`. The labels on the y-axis are set with `group.names=`. The point character to be used is 19. Both the main title and the subtitle should be magnified by a suitable factor. Use `ylim=` to suitably scale the y-axis.

(4 points)

**Exercise 19:** *The following high-level plotting commands produce pretty plots. We will not go into detail here as such plots are less important for biological data.*
Enter `data(volcano)` to load the volcano data into the variable `volcano`. Plot this variable with the command `persp`. Set the angles `theta=40` and `phi=30` to define a suitable viewing direction. Let the color be `col="green"`.




Next plot the variable `volcano` with the command `image`. By default, `image` uses `heat.colors(12)` to visualize the third dimension with colours. As we have landscape data it is better to use `col=terrain.colors(100)`. The axes have no meaning here so supress both axes. The main title of the following figure (left-hand side) is magnified by a factor 1.5.




To improve the impression of the third dimension, add contour lines with the command `contour()`. The contour lines are added to the previous plot if you specify `add=TRUE`. Specify the levels of the contour plot with `levels=` ; use the vector which increases from 90 to 200 in steps of size 5 for the levels. Use `col=` to change the colour of the level lines to 'brown'. You could also try `heat.colors()` or `topo.colors()` instead of `terrain.colors()`.

(4 points)