Practical Exercises for **Day 5**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2018

Exercise 13A: Outside plot frame

(a) Type `demo(graphics)` in your console and press enter. This command shows you a nice demonstration of possible R graphics.

```
# After the demonstration us the following commands:
dev.off()
par(mfrow=c(1,1))
```

(b) Change the x-axis and y-axis labelling of a boxplot plotting the `len` variable of the `ToothGrowth` data set.

```
data("ToothGrowth")
boxplot(ToothGrowth$len)
```

```
boxplot(ToothGrowth$len, xlab = "Length of Teeth",
        ylab = "Length in mm")
```

(c) How do you set a main title for your above plot?

```
# OPTION 1:
boxplot(ToothGrowth$len, xlab = "Length of Teeth",
        ylab = "Length in mm",
        main = "Boxplot of Tooth Length")
# OPTION 2:
boxplot(ToothGrowth$len, xlab = "Length of Teeth",
        ylab = "Length in mm")
title("Boxplot of Tooth Length")
```

(d) What does the following command do?

```
par(mfrow=c(2,2))
```

```
# With the par(...) function, you can include the option
# mfrow=c(nrows, ncols) to create a matrix of nrows x ncols plots
# that are filled in by row.
par(mfrow=c(2,2)) # 2 rows, 2 columns
par(mfrow=c(4,3)) # 4 rows, 3 columns
# DO NOT FORGET TO CHANGE IT BACK TO:
par(mfrow=c(1, 1)) # the default
```

(e) We have six different feed types in `chickwts`. Try to plot two separate boxplots for `casein` and `horsebean` and set the same minimum and maximum for the y-axis. Use the function `subset` for doing so.

```
sub.casein <- subset(chickwts, feed == "casein")
sub.casein <- droplevels(sub.casein)
sub.horsebean <- subset(chickwts, feed == "horsebean")
sub.horsebean <- droplevels(sub.horsebean)
```

```
sub.casein <- subset(chickwts, feed == "casein")
sub.casein <- droplevels(sub.casein)
sub.horsebean <- subset(chickwts, feed == "horsebean")
sub.horsebean <- droplevels(sub.horsebean)
summary(sub.casein$weight)
summary(sub.horsebean$weight)
par(mfrow=c(1, 2))
boxplot(sub.casein$weight ~ sub.casein$feed, ylim = c(100, 410))
boxplot(sub.horsebean$weight ~ sub.horsebean$feed, ylim = c(100, 410))
```

(f) How do you enlarge the font size of the axis as well as the axis labels of the following plot with the `perulung` data set?

```
lung <- read.csv("perulung_ems.csv", sep=";")
par(mfrow=c(1,1))
plot(lung$fev1, lung$height)
```

```
plot(lung$fev1, lung$height, cex.axis = 1.5, cex.lab = 1.5)
plot(lung$fev1, lung$height, cex.axis = 1.5, cex.lab = 1.5, las = 1)
```

(g) Label the x-axis of the following plot with "Vitamin C in μg". Use the greek letter for μ.

```
plot(ToothGrowth$dose, ToothGrowth$len)
```

```
plot(ToothGrowth$dose, ToothGrowth$len,
     xlab = expression(paste("Vitamin C in ", mu, "g")))
```

(h) Read `http://www.statmethods.net/advgraphs/parameters.html`.

Exercise 13B: Inside the square of the plot

(a) Type `demo(graphics)` in your console and press enter. This command shows you a nice demonstration of possible R graphics.

```
# After the demonstration us the following commands:
dev.off()
par(mfrow=c(1,1))
```

(b) Add a legend to the following barplot. Are there several different solutions for this?

```
library("MASS")
data("bacteria")
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1),
        beside=FALSE, ylim = c(0,0.8))
```

```
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1), beside=FALSE,
        ylim = c(0,0.8), legend.text = levels(bacteria$y))
# Helen's solution (THANK YOU!):
barplot(prop.table(table(bacteria$y, bacteria$trt), margin=1),
        beside=FALSE, ylim = c(0,0.8), col = topo.colors(2),
        ylab = "y", xlab = "treatments",
        main = "bacteria")
legend("topright", legend = c("yes", "no"), fill = topo.colors(2))
```

(c) Add a density line to this histogram.

```
hist(ToothGrowth$len, prob = TRUE, col = "grey", ylim = c(0, 0.05))
```

```
hist(ToothGrowth$len, prob = TRUE, col = "grey", ylim = c(0, 0.05))
# add a density estimate with defaults
lines(density(ToothGrowth$len), col="blue", lwd = 4)
# add a density estimate with adjustments
lines(density(ToothGrowth$len, adjust=2), lty="dotted", col="darkgreen",
      lwd = 4)
```

(d) Add a **dotted red** linear regression line to the following plot.

```
plot(lung$height, lung$fev1)
```

```
plot(lung$height, lung$fev1)
abline(lm(lung$fev1 ~ lung$height), col = "red",
       lwd = 3, lty = 2)
# See
# https://stackoverflow.com/questions/24173468/r-print-equation-of-linear-regression-on-the-plo
# to learn how to print equation of linear regression on the plot
## rounded coefficients for better output
lm.mod <- lm(lung$fev1 ~ lung$height)
cf <- round(coef(lm.mod), 2)
## sign check to avoid having plus followed by minus for negative coefficients
eq <- paste0("fev1 = ", cf[1],
             ifelse(sign(cf[2])==1, " + ", " - "), abs(cf[2]), " height ")
## printing of the equation
mtext(eq, 3, line=-2)
```

(e) Color the points in the following plot according to the `sex` variable.

```
plot(lung$height, lung$fev1)
```

```
plot(lung$height, lung$fev1, col = as.numeric(lung$sex))
```

(f) Add two linear regression lines separately for `female` and `male`to the following plot.

```
plot(lung$height, lung$fev1)
```

```
plot(lung$height, lung$fev1, col = as.numeric(lung$respsymptoms))
abline(lm(lung$fev1 ~ lung$height,
          data = subset(lung, sex == "female")),
       col  = "black")
abline(lm(lung$fev1 ~ lung$height,
          data = subset(lung, sex == "male")),
       col  = "red")
# library("graphics")
# coplot(fev1 ~ height | sex, data = lung, panel = panel.smooth)
# coplot(fev1 ~ height | respsymptoms, data = lung, panel = panel.smooth)
```

(g) Color the points in the following plot according to the `supp` variable. Use different point characters (`pch`) based on the `supp` variable.

```
plot(ToothGrowth$len, ToothGrowth$dose)
```

```
plot(ToothGrowth$len, ToothGrowth$dose,
     pch = levels(ToothGrowth$supp),
     col = as.numeric(ToothGrowth$supp))
```

(h) Read `http://www.statmethods.net/advgraphs/parameters.html`.

Exercise 14

(a) Load the below data set and for further information check the command `?water`.

```
# install.packages("HSAUR3")
library("HSAUR3")
data("water")
```

```
str(water)
head(water)
summary(water)
```

(b) Try to plot the variables `mortality` against `hardness` from the `water` data set.

```
par(mfrow=c(1,1))
plot(x = water$hardness, y = water$mortality)
plot(mortality ~ hardness, data = water)
```

(c) Add a main title to the above plot (`mortality` against `hardness`).

```
plot(x = water$hardness, y = water$mortality,
     main = "Calcium concentration against mortality")
plot(mortality ~ hardness, data = water,
     main = "Calcium concentration against mortality")
```

(d) Change the ...

(a) font size of the axis annotation

(b) font size of the x- and y-axis labels

(c) the point sizes within the plot

... of the above plot (`mortality` against `hardness`).

```
# cex: number indicating the amount by which plotting text and symbols
# should be scaled
# cex.axis: magnification of axis annotation relative to cex
plot(x = water$hardness, y = water$mortality,
     cex.axis = 1.5, # (1) enlarge number of the axis
     cex.lab = 1.5, # (2) enlarge font size of axis labels
     cex = 1.5, # (3) enlarge point size within plot
     main = "Calcium concentration vs. mortality")
plot(mortality ~ hardness, data = water,
     cex.axis = 1.5, # enlarge number of the axis
     cex = 1.5, # enlarge point size within plot
     cex.lab = 1.5, # enlarge font size of axis labels
     main = "Calcium concentration vs. mortality")
```

(e) Looking at the above plot: Do you think the two variables `hardness` and `mortality` correlate? What function do you use to find out the correlation coefficient? Do they have a positive or a negative correlation coefficient? How do you interpret the correlation coefficient in your own words?

```
cor(x = water$hardness, y = water$mortality) # -0.6548486
cor.test(x = water$hardness, y = water$mortality)
# negative correlation of -0.65 with confidence interval of [-0.78, -0.48]:
# the higher the calcium concentration (hardness),
# the smaller the averaged annual mortality per 100.000 male
# inhabitants (mortality)
```

(f) In the `water` data set, can you graphically find out if there is a difference between the the two variables `hardness` and `mortality` conditional on the `location` (`North`, `South`).

```
plot(x = water$hardness, y = water$mortality,
     col = as.numeric(water$location),
     pch = 16, cex.axis = 1.5,
     cex = 1.5, cex.lab = 1.5)
library("graphics")
coplot(mortality ~ hardness | location, data = water, panel = panel.smooth)
```

(g) Add a legend to the above plot so that you can easily differentiate the locations (`North` or `South`) of the observations.

```
plot(x = water$hardness, y = water$mortality,
     col = as.numeric(water$location),
     pch = 16, cex.axis = 1.5,
     cex = 1.5, cex.lab = 1.5)
legend("topright", legend = levels(water$location),
       col = c("black", "red"), pch = 16, cex = 1.5)
```

(h) Do a barplot of the variable `location` from the `water` data set.

```
barplot(table(water$location))
```

(i) ADDITIONAL: Try if any of these following plotting functions can be applied to the data sets `perulung` or `ToothGrowth`.

```r
install.packages("graphics")
library("graphics")
?coplot
#
# install.packages("lattice")
library("lattice")
?xyplot
#
?interaction.plot
```

```r
# PERULUNG DATA SET
coplot(fev1 ~ height | sex, data = lung, panel = panel.smooth)
coplot(fev1 ~ height | respsymptoms, data = lung, panel = panel.smooth)

xyplot(fev1 ~ height | sex, data = lung)
xyplot(fev1 ~ height | respsymptoms, data = lung)

# ToothGrowth DATA SET
interaction.plot(ToothGrowth$dose,
                 ToothGrowth$supp,
                 ToothGrowth$len,
                 fixed = TRUE)
```