Practical Exercises for **Day 4**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Exercise 11

- Apply the two-sided two sample t-test to suitable variables of the data set `ToothGrowth`.

```r
?t.test
t.test(ToothGrowth$len ~ ToothGrowth$supp)
t.test(len ~ supp, data = ToothGrowth)
# p-value = 0.06039 (borderline) significant, close to 0.05
# p-value says the difference is not (borderline) significant
# however, the boxplot do somehow look different
boxplot(ToothGrowth$len ~ ToothGrowth$supp)
# change the default setting of var.equal
t.test(ToothGrowth$len ~ ToothGrowth$supp, var.equal = TRUE)
t.test(ToothGrowth$len ~ ToothGrowth$supp, var.equal = FALSE) # DEFAULT!

# Define subset
sub.OJ <- subset(ToothGrowth, supp == "OJ")
sub.VC <- subset(ToothGrowth, supp == "VC")
# Drop (unused) levels for each subset
sub.VC$supp <- droplevels(sub.VC$supp)
levels(sub.VC$supp) # check that levels are dropped
sub.OJ$supp <- droplevels(sub.OJ$supp)
levels(sub.OJ$supp) # check that levels are dropped
# Additional option for comparing lengths between the two groups:
# Compare the two vectors of lengths
t.test(sub.VC$len, sub.OJ$len)
```

- Interpret the results.

```r
# t = 1.9153
# df = 55.309
# p-value = 0.06063
# 95 percent confidence interval: -0.1710156  7.5710156
# sample mean in group OJ: 20.66333
# sample mean in group VC: 16.96333
# Also with the lm(...) function  for "linear model" you get the same sample means:
lm.mod0 <- lm(ToothGrowth$len ~ ToothGrowth$supp - 1)
coef(lm.mod)
```

- Apply the two-sided t-test to the `perulang_ems` data set

```r
# two-sided t-test of fev1 vs respsymptoms
t.test(lung$fev1 ~ lung$respsymptoms)
t.test(fev1 ~ respsymptoms, data = lung)
# Define linear model
mod.fev.resp.0 <- lm(lung$fev1 ~ lung$respsymptoms)
summary(mod.fev.resp.0)
mod.fev.resp.1 <- lm(lung$fev1 ~ lung$respsymptoms - 1)
summary(mod.fev.resp.1)
# Coefficients of linear model
coef(mod.fev.resp.0)
coef(mod.fev.resp.1)
# Anova
anova(mod.fev.resp.0)
anova(mod.fev.resp.1)
# two-sided t-test of fev1 vs sex
t.test(lung$fev1 ~ lung$sex)
t.test(fev1 ~ sex, data = lung)
# Define linear model
mod.fev.sex.0 <- lm(lung$fev1 ~ lung$sex)
mod.fev.sex.1 <- lm(lung$fev1 ~ lung$sex - 1)
# Coefficients of linear model
coef(mod.fev.sex.0)
coef(mod.fev.sex.1)
# Anova
```

```r
anova(mod.fev.sex.0)
anova(mod.fev.sex.1)
```

Exercise 12

- Apply the Chi-square Test and the fisher exact test to the whole `bacteria` data set.

```r
library("MASS")
data(bacteria)
summary(bacteria)
subbac <- subset(bacteria, week == 2)
bacteria$trt.new <- bacteria$trt
levels(bacteria$trt.new) <- c("placebo", "drug", "drug")
bacteria$trt.new <- droplevels(bacteria$trt.new)
# Ordering of the variables does not matter
chisq.test(table(bacteria$trt, bacteria$y))
chisq.test(table(bacteria$y, bacteria$trt))
chisq.test(bacteria$trt, bacteria$y)
my.table <- table(bacteria$trt, bacteria$y)
chisq.test(my.table)
table(subbac$trt, subbac$y)
chisq.test(table(subbac$trt, subbac$y))
fisher.test(table(subbac$trt, subbac$y))
fisher.test(table(subbac$trt.new, subbac$y))
# Chi-squared test with trt and y
chisq.test(table(bacteria$trt, bacteria$y))
# Fisher test with trt and y
fisher.test(table(bacteria$trt, bacteria$y))
```

- Apply the Chi-square Test and the fisher exact test to the subset of `bacteria` containing only the observations taken in week 2. Are there any issues?

```r
subbac <- subset(bacteria, week == 2)
# Chi-squared test with trt and y
chisq.test(table(subbac$trt, subbac$y))
# --> NOT RELIABLE RESULTS: at least 5 observations per group.
# Fisher test with trt and y
fisher.test(table(subbac$trt, subbac$y))
```

- Repeat this exercise by using the (previously defined) combined `trt.new` variable with the two levels `treated` and `drug`.

```r
# WHOLE DATA SET
# Chi-squared test with trt.new and y
chisq.test(table(bacteria$trt.new, bacteria$y))
# Fisher test with trt.new and y
fisher.test(table(bacteria$trt.new, bacteria$y))
# SUB DATA SET only observations from week 2
# Chi-squared test with trt.new and y
chisq.test(table(subbac$trt.new, subbac$y))
# --> NOT RELIABLE RESULTS: at least 5 observations per group.
# Fisher test with trt.new and y
fisher.test(table(subbac$trt.new, subbac$y))
```

- Could you also obtain the odds ratios?

```r
fisher.test(table(subbac$trt.new, subbac$y))
fisher.test(bacteria$y, bacteria$ap)
my.logreg <- glm(y ~ ap, data = bacteria, family = "binomial")
summary(my.logreg)
exp(0.8473 )
coef(my.logreg)
exp(coef(my.logreg))
```

- Try also a logistic regression in R. Ask Google for help!

```r
model.logreg <- glm(bacteria$y ~ bacteria$trt.new, family = "binomial")
model.logreg <- glm(y ~ trt.new, data = bacteria, family = "binomial")
summary(model.logreg)
anova(model.logreg)
coef(model.logreg)
exp(coef(model.logreg))
```

Exercise 13A: Outside plot frame

- 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))
```

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

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

- 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")
```

- 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
```

- 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)
boxplot(sub.casein$weight ~ sub.casein$feed, ylim = c(100, 410))
boxplot(sub.horsebean$weight ~ sub.horsebean$feed, ylim = c(100, 410))
```

- 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?

```
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)
```

- 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")))
```

- Read `http://www.statmethods.net/advgraphs/parameters.html`.

Exercise 13B: Inside the square of the plot

- Type `demo(graphics)` in your console and press enter.
- Add a legend to the following barplot. Are there several different solutions for this?

```
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))
```

- 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)
```

- 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)
```

```r
# See
# https://stackoverflow.com/questions/24173468/r-print-equation-of-linear-regression-on-the-p
# 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)
```

- Color the points in the following plot according to the sex variable.

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

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

- Add two linear regression lines separately for female and maleto the following plot.

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

```r
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)
```

- 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))
```

- Read `http://www.statmethods.net/advgraphs/parameters.html`.