Practical Exercises for **ALL EXERCISES**

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Exercise 1

- Open R Studio
- Open a new R-Script
- Load data set `chickwts`

```r
data(chickwts)
head(chickwts)
# ?chickwts
```

- Do summary statistic (numerically and graphically)

```r
summary(chickwts)
tapply(chickwts$weight, chickwts$feed, mean)
tapply(chickwts$weight, chickwts$feed, median)
tapply(chickwts$weight, chickwts$feed, sd)
table(chickwts$feed)
barplot(table(chickwts$feed))
boxplot(chickwts$weight ~ chickwts$feed)
# boxplot(weight ~ feed, data = chickwts)
hist(chickwts$weight)
boxplot(weight ~ feed, data = chickwts, col = "lightgray",
        varwidth = TRUE, notch = TRUE, main = "chickwt data",
        ylab = "Weight at six weeks (gm)")
```

Anova, lm, which groups differ, Bonferroni, Tukey-Anscombe Histogram with density line Normally distributed weights

Exercise 2

- Create a data frame with 3 columns.

```r
a <- c(1, 2, 3, 4)
b <- c("d", "h", "h", "d")
c <- factor(c("male", "female", "male", "female"),
            levels = c("female", "male"))
dat <- data.frame(a, b, c)
dat
```

Exercise 3

- Install package `MASS`.

```r
# install.packages("MASS")
library("MASS")
```

- Load data set `bacteria`.

```r
data(bacteria)
head(bacteria)
# ?bacteria
```

- Do summary statistic (numerically and graphically).

```r
summary(bacteria)
table(bacteria$week)
barplot(table(bacteria$week))
barplot(table(bacteria$trt))
table(bacteria$trt, bacteria$ap)
table(bacteria$trt, bacteria$y)
%
fisher.test(table(bacteria$trt, bacteria$y))
%
prop.table(table(bacteria$trt, bacteria$y))
prop.table(table(bacteria$trt, bacteria$y), margin = 1)
prop.table(table(bacteria$trt, bacteria$y), margin = 2)
%
plot(prop.table(table(bacteria$trt, bacteria$y)))
mosaicplot(~trt + y, data = bacteria)
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1), beside=TRUE)
barplot(prop.table(table(bacteria$trt, bacteria$y),margin=1), beside=TRUE)
barplot(prop.table(table(bacteria$y, bacteria$trt),margin=1), beside=FALSE)
barplot(prop.table(table(bacteria$trt, bacteria$y),margin=1), beside=FALSE)
?barplot
```

- Select only observations collected during the second week.

```r
subset(bacteria, week == 2)
ss <- subset(bacteria, week == 2)
summary(ss)
# Check if we only have observations of week 2.
table(bacteria$week)
table(ss$week)
```

Exercise 4

What is conceptionally the difference between these bracket types ([...], (...))?

```r
chickwts[, 2]
summary(aov(weight ~ feed, data = chickwts))
```

Exercise 5

- How many levels has the factor variable `trt` from `bacteria`?

```r
str(bacteria)
head(bacteria$trt)
table(bacteria$trt)
levels(bacteria$trt)
nlevels(bacteria$trt)
```

- Define a new variable `trt.new` in which you combine the levels `drug` and `drug+` into one single level and label it as `treated`. The new variable `trt.new` should in the end have two levels: `placebo` and `treated`.

```r
table(bacteria$trt)
# OPTION 1:
# Test how many levels are in the variable "trt"?
levels(bacteria$trt)
bacteria$trt.new <- bacteria$trt
# Overwrite the levels "placebo", "drug", "drug+" with new
# levels called "placebo", "drug", "drug" --> combine "drug" and "drug+"
levels(bacteria$trt.new) <- c("placebo", "drug", "drug")
# Do table for variable "trt" and "trt.new" to see if you combined correctly
table(bacteria$trt)
table(bacteria$trt.new)
# Rename the levels from "placebo", "drug" to "placebo", "treated"
levels(bacteria$trt.new) <- c("placebo", "treated")
# Do another table to check if you did everything correctly:
table(bacteria$trt.new)
```

- Do summary statistics for `placebo` and `treated` group.

```r
summary(bacteria)
table(bacteria$trt.new)
barplot(table(bacteria$trt.new))
table(bacteria$trt.new, bacteria$ap)
table(bacteria$trt.new, bacteria$y)
plot(table(bacteria$trt.new, bacteria$y))
```

Exercise 6

- Load data set `ToothGrowth`.

```r
data(ToothGrowth)
str(ToothGrowth)
head(ToothGrowth)
```

- Do summary statistic (numerically and graphically).

```r
table(ToothGrowth$supp)
tapply(ToothGrowth$len, ToothGrowth$supp, mean)
tapply(ToothGrowth$len, ToothGrowth$supp, median)
tapply(ToothGrowth$len, ToothGrowth$supp, sd)
%
tapply(ToothGrowth$len, ToothGrowth$dose, mean)
tapply(ToothGrowth$len, ToothGrowth$dose, median)
tapply(ToothGrowth$len, ToothGrowth$dose, sd)
%
barplot(table(ToothGrowth$supp))
hist(ToothGrowth$len)
# install.packages("graphics")
library("graphics")
coplot(len ~ dose | supp, data = ToothGrowth, panel = panel.smooth,
       xlab = "ToothGrowth data: length vs dose, given type of supplement")
```

- Define additional column `dose.fac` by converting the numeric variable `dose` into a factor variable.

```r
table(ToothGrowth$dose)
class(ToothGrowth$dose)
ToothGrowth$dose.fac <- factor(ToothGrowth$dose, levels = c("0.5", "1", "2"))
class(ToothGrowth$dose.fac)
table(ToothGrowth$dose.fac)
```

- Are the tooth length measurements normally distributed within the treatment (`supp`: VC or OJ) and within in the different doses (`dose`: 0.5, 1, 2)?

```r
# supp: VC, OJ
sub.OJ <- subset(ToothGrowth, supp == "OJ")
sub.VC <- subset(ToothGrowth, supp == "VC")
# graphically
qqnorm(sub.OJ$len)
qqline(sub.OJ$len)
qqnorm(sub.VC$len)
qqline(sub.VC$len)
```

```r
# with a statistical test
shapiro.test(sub.OJ$len)
shapiro.test(sub.VC$len)
# dose: 0.5, 1, 2
sub.0.5 <- subset(ToothGrowth, dose.fac == "0.5")
sub.1 <- subset(ToothGrowth, dose.fac == "1")
sub.2 <- subset(ToothGrowth, dose.fac == "2")
# graphically
qqnorm(sub.0.5$len)
qqline(sub.0.5$len)
qqnorm(sub.1$len)
qqline(sub.1$len)
qqnorm(sub.2$len)
qqline(sub.2$len)
# with a statistical test
shapiro.test(sub.0.5$len)
shapiro.test(sub.1$len)
shapiro.test(sub.2$len)
```

Exercise 7

- Import the data set `perulung_ems.csv` (taken from Kirkwood and Sterne, 2nd edition) into R.

  Data from a study of lung function among children living in a deprived suburb of Lima, Peru.

  Variables:

  - `fev1`: in liter, ”Forced Expiratory Volume in 1 second” measured by a spirometer. This is the maximum volume of air which the children could breath out in 1 second
  - `age`: in years
  - `height`: in cm
  - `sex`: 0 = girl, 1 = boy
  - `respsymp`: respiratory symptoms experienced by the child over the previous 12 months

- What *delimiter* do you need to choose?

```
# OPTION 1:
# install.packages("readr")
library("readr")
lung <- read_delim("~/Dropbox/201710_Makerere/03_Exercises/data/perulung_ems.csv",
                    ";", escape_double = FALSE, trim_ws = TRUE)
lung <- data.frame(lung)
# OPTION 2:
# Import .csv file with the help of the read.csv function
# Be sure to add sep = ";" so that we separate the columns.
lung <- read.csv("C:\\Users\\Exercises\\data\\perulung_ems.csv", sep = ";")
head(lung)
str(lung)
```

- Do all variables have the correct data type (numeric, integer, factor)? If not, do correct and / or define them.

```
head(lung)
str(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
# levels(lung$sex) <- c("female", "male")
# levels(lung$sex)[levels(lung$sex)=="0"] <- "female"
# levels(lung$sex)[levels(lung$sex)=="1"] <- "male"
# tapply(lung$fev1, lung$sex, mean)
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))
```

```
library(usdm)
# check for multicollinearity by using variance inflation factors
# cerate a dataframe just with the three continuous/numeric variables fevs, age and height
try.vif <- lung[,c("fev1","height","age")];
# perform scatterplots for these three variables
pairs(try.vif)
# get the three VIF, as a rule of thumb they should be < 3
vif(try.vif)
```

Check for heteroscedascity or homogeneity of variances

```r
?bartlett.test
data("chickwts")
bartlett.test(weight ~ feed, data = chickwts)
```

Exercise 8

Apply the summary statistics to the `perulung_ems` and `ToothGrowth` data set.

```r
# Read in .csv data
lung <- read.csv("C:\\Users\\Exercises\\data\\perulung_ems.csv", sep = ";")
head(lung)
str(lung)
summary(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
levels(lung$sex) <- c("female", "male")
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))
# Continuous and factor
tapply(lung$height, lung$sex, mean)
tapply(lung$height, lung$respsymptoms, mean)
# Factor and factor
table(lung$respsymptoms, lung$sex)
prop.table(table(lung$respsymptoms, lung$sex))
# Continuous and factor
tapply(lung$age, lung$sex, mean)
tapply(lung$age, lung$respsymptoms, mean)
# Continuous and factor
tapply(lung$fev1, lung$sex, mean)
tapply(lung$fev1, lung$respsymptoms, mean)
# Continuous and continuous
pairs(lung)
cor.test(lung$fev1, lung$age, method = "pearson")
cor.test(lung$fev1, lung$height, method = "pearson")
# ToothGrowth
summary(ToothGrowth)
table(ToothGrowth$supp)
tapply(ToothGrowth$len, ToothGrowth$supp, mean)
tapply(ToothGrowth$len, ToothGrowth$supp, median)
tapply(ToothGrowth$len, ToothGrowth$supp, sd)
table(ToothGrowth$dose)
tapply(ToothGrowth$len, ToothGrowth$dose, mean)
tapply(ToothGrowth$len, ToothGrowth$dose, median)
tapply(ToothGrowth$len, ToothGrowth$dose, sd)
```

Exercise 9A: Plausibility Checks

- What can go wrong?
- Identify different strategies for spotting these potential errors.
  - Logical errors
  - Spelling mistakes
- Import the data set `bacteria_plausibility_check.csv` to R.

```
# OPTION 1:
# install.packages("readr")
library("readr")
bac <- read_delim("~/Dropbox/201710_Makerere/03_Exercises/data/bacteria_plausibility_check.cs
                  ";", escape_double = FALSE, trim_ws = TRUE)
bac <- data.frame(bac)
# OPTION 2:
# Import .csv file with the help of the read.csv function
# Be sure to add sep = ";" so that we separate the columns.
bac <- read.csv("~/Dropbox/201710_Makerere/03_Exercises/data/bacteria_plausibility_check.csv"
head(bac)
str(bac)
summary(bac)
```

- Detect the **six** errors in the imported data set `bacteria_plausibility_check.csv` in R.

```
str(bac)
table(bac$y) # We have wrong factor levels: 0, 1
table(bac$ap)
table(bac$hilo) # We have a spelling mistake: Hi.
table(bac$week) # There's only ONE observation in week 20.
table(bac$ID)
table(bac$trt) # We have wrong factor levels: drug++, penicillin+
summary(bac$child_weight) # child weight of 302.8 kg is impossible --> comma
```

- Find possible solutions in R how to handle these challenges.

```
bac$y[which(bac$y == 0)] <- "n"
# bac$y[bac$y == 0] <- "n"
bac$y[which(bac$y == 1)] <- "y"
# Delete the unused levels with the function droplevels(...)
bac$y <- droplevels(bac$y)
bac$hilo[bac$hilo == "Hi"] <- "hi"
bac$hilo[which(bac$hilo == "Hi")] <- "hi"
levels(bac$hilo) <- c("hi", "hi", "lo")
summary(bac)
bac <- bac[-which(bac$week == 20), ] # dim(bac)
bac$trt[bac$trt == "drug++"] <- "drug+"
bac$trt[bac$trt == "penicillin+"] <- "drug+"
table(bac$trt) # We have wrong factor levels: drug++, penicillin+
bac$child_weight[bac$child_weight == 302.8] <- 30.28
summary(bac)
```

- Do all variables have the correct data type (numeric, integer, factor)? - If not, do correct / define them.

```
bac$y <- factor(bac$y, levels = c("n", "y"))
bac$hilo[bac$hilo == "Hi"] <- "hi"
bac$ID <- factor(bac$ID)
bac$trt <- factor(bac$trt)
```

Exercise 9B: Missing Values

- Check out the difference between the different missing values

```
y1 <- c(2, 4, 3, NA, 6, 1)
y2 <- c("diseased", "healthy", NA, "NA")
y3 <- c(1, "NA", 0, 1, NaN)
%
is.na(y1)
which(is.na(y1))
is.na(y2)
which(is.na(y2))
is.na(y3)
which(is.na(y3))
is.nan(y3)
```

- Create a vector with missing values and determine the mean and median

```
myvector <- c(1:3,NA,NA,1:3)
mean(myvector)
mean(myvector,na.rm=TRUE) # calculates c(1, 2, 3, 1, 2, 3)
median(myvector,na.rm=TRUE)
```

- If `x = c (22,3,7,NA,NA,67)` what will be the output for the R statement `length(x)`?

```
x <- c (22,3,7,NA,NA,67)
length(x)
```

- If `x = c(NA,3,14,NA,33,17,NA,41)` which line of R code removes all occurrences of NA in x.

```
x <- c(NA,3,14,NA,33,17,NA,41)
x[!is.na(x)]
x[is.na(x)]
x[which(is.na(x))] <- 0
```

- If `y = c(1,3,12,NA,33,7,NA,21)` what R statement will replace all occurrences of NA with 11?

```
y <- c(1,3,12,NA,33,7,NA,21)
y[y=="NA"] <- 11
y[is.na(y)] <- 11
y[y==11] <- NA
```

- If `x = c(34,33,65,37,89,NA,43,NA,11,NA,23,NA)` then what will count the number of occurrences of NA in `x`?

```
x <- c(34,33,65,37,89,NA,43,NA,11,NA,23,NA)
sum(x=="NA")
sum(x == "NA", is.na(x))
sum(is.na(x))
```

- Create a vector and find the number of missing values and their position

```
x1 <- c(rnorm(10,5,2),NA,5:12,NA,6,7.5,NA)
is.na(x1)
summary(x1)
sum(is.na(x1))
which(is.na(x1))
```

- Now, create the vector x2 and assess the difference to x1

```
x2 <- c(rnorm(10,5,2),NA,5:12,NA,6,7.5,NA,log(-2))
x2
```

- What is the meaning of "NA" versus "NaN"?
- Replace the missing values in x1 with a 0, and check that no NAs are present try two different commands to coerce the NAs into 0

```
x1[is.na(x1)] <- 0
is.na(x1)
# or
ifelse(is.na(x1),0,x1)
```

Exercise 10

- Import the data set `water_errors.csv` to R: A data frame with 61 observations on the following 6 variables.
  - **location**: a factor with levels `North` and `South` indicating whether the town is as north as Derby.
  - **town**: the name of the town.

- **mortality**: averaged annual mortality per 100.000 male inhabitants.
- **hardness**: calcium concentration (in parts per million).
- **smoker**: If there are any smokers living in town.
- **num.of.cig**: In case, smokers live in town, what number of cigarettes do they smoke per day.

```
# H2O_err <- read_csv("C:\\Users\\admin\\Dropbox\\201710_Makerere\\03_Exercises\\data\\water_
# str(H2O_err)
# H2O_err <- data.frame(H2O_err)
# str(H2O_err)
# BEST SOLUTION how to read it in:
# Try to use the "read.csv(...)" function to read data in!
# use the separator sep=";" or sep="," - which ever works better.
H2O_err <- read.csv("C:\\Users\\admin\\Dropbox\\201710_Makerere\\03_Exercises\\data\\water_er
str(H2O_err)
%
# H2O_err <- read_csv("~/Dropbox/201710_Makerere/03_Exercises/data/water_errors.csv")
H2O_err <- read.csv("~/Dropbox/201710_Makerere/03_Exercises/data/water_errors.csv", sep=",")
%
%
C:\Users\admin\Dropbox\201710_Makerere\03_Exercises\data
H2O_err <- data.frame(H2O_err)
str(H2O_err)
head(H2O_err)
```

- Detect the errors in the imported data set `water_errors.csv` in R.

```
str(H2O_err)
table(H2O_err$location) # Only one N and only one West observation.
table(H2O_err$town) # LIVERPOOL is in capital letter.
summary(H2O_err$mortality)
summary(H2O_err$hardness) # hardness of -2 does not make sense, two NA's
table(H2O_err$num.of.cig) # only one "zero" observation (wrong coding / level)
table(H2O_err$smoker, H2O_err$num.of.cig) # non-smokers who smoke more than 20?
```

- Find possible solutions in R how to handle these challenges.

```
str(H2O_err)
which(H2O_err$location == "N") # 6th row
which(H2O_err$location == "West") # 9th row
H2O_err$location[H2O_err$location == "N"] <- "North"
H2O_err$location[H2O_err$location == "West"] <- NA # Option 1: Set to NA.
dim(H2O_err)
H2O_err <- H2O_err[-which(H2O_err$location == "West"), ] # Option 2: Remove from data.
dim(H2O_err)
# H2O_err$town[H2O_err$town == "LIVERPOOL"] <- "Liverpool"
# H2O_err <- H2O_err$town[-which(H2O_err$town == "LIVERPOOL"), ]
which(is.na(H2O_err$hardness))
H2O_err$hardness[which(is.na(H2O_err$hardness))] <- NA
H2O_err$hardness[which(H2O_err$hardness == -2)] <- NA
# H2O_err$hardness[which(H2O_err$hardness == -2)] <- 2
summary(H2O_err$hardness)
# Check levels of varibale num.of.cig
levels(H2O_err$num.of.cig)
table(H2O_err$num.of.cig)
# Change the zero level to none
H2O_err$num.of.cig[H2O_err$num.of.cig == "zero"] <- "none"
# Drop unused levels
H2O_err$num.of.cig <- droplevels(H2O_err$num.of.cig)
# levels(droplevels(H2O_err$num.of.cig))
table(H2O_err$num.of.cig)
%
which.F.morethan20 <- which(H2O_err$smoker == FALSE & H2O_err$num.of.cig == "more than 20")
H2O_err[which.F.morethan20, ]
# OPTION 1:
H2O_err$num.of.cig[which.F.morethan20] <- NA
# OPTION 2:
H2O_err$smoker[which.F.morethan20] <- TRUE
# check again, that we corrected it right
H2O_err[which.F.morethan20, ]
table(H2O_err$smoker, H2O_err$num.of.cig) # check again!
%
which(H2O_err$smoker == FALSE & H2O_err$num.of.cig == "more than 20")
```

```
which.T.none <- which(H20_err$smoker == TRUE & H20_err$num.of.cig == "none")
H20_err[which.T.none, ]
H20_err$smoker[which.T.none] <- FALSE
table(H20_err$smoker, H20_err$num.of.cig)
```

- Do all variables have the correct data type (numeric, integer, factor)? - If not, do correct / define them.

```
str(H20_err)
levels(H20_err$location)
H20_err$location <- factor(H20_err$location, levels = c("North", "South", NA),
                           exclude = NULL)
levels(H20_err$smoker)
H20_err$smoker <- factor(H20_err$smoker, levels = c("FALSE", "TRUE"))
table(H20_err$num.of.cig)
H20_err$num.of.cig <- factor(H20_err$num.of.cig,
                             levels = c("none", "less than 5", "5 to 20", "more than 20"),
                             ordered = TRUE)
table(H20_err$num.of.cig)
levels(H20_err$num.of.cig)
levels(H20_err$num.of.cig) <- c("none", "1 to less than 5", "5 to 20", "more than 20")
table(H20_err$num.of.cig)
levels(H20_err$num.of.cig)
```

Exercise 11

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

```
?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.

```
# 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

```
# 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)
```

```r
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
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?

```
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`.

```
# 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?

```
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!

```
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)
# 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.

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

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

- 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)
```

- 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`.

Exercise 14

- 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)
```

- Try to plot the variables `mortality` against `hardness` from the `water` data set.

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

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

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

- Change the ...

  1. font size of the axis annotation
  2. font size of the x- and y-axis labels
  3. the point sizes within the plot

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

```r
# 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")
```

- 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)
```

- 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)
```

- 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)
```

- Do a barplot of the variable `location` from the `water` data set.

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

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

```
install.packages("graphics")
library("graphics")
```

```r
?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)
```

Exercise 15

- Download the .R file `ANOVA_with_chickwts.R` from the switch drive and have another look on how we applied the anova to the `chickwts` data set.
- Load the `ToothGrowth` data set into R and encode the numeric variable `dose` as a factor variable. Define the new factor variable as `dose.fac` with the three levels `low`, `med` and `high` and add it to the data frame of `ToothGrowth`.

```r
data(ToothGrowth)
str(ToothGrowth)
head(ToothGrowth)
ToothGrowth$dose.fac <- factor(ToothGrowth$dose, levels = c(0.5, 1.0, 2.0),
                               labels = c("low", "med", "high"))
table(ToothGrowth$dose.fac)
```

- Visualize the variable `len` per `dose` level in a boxplot.

```
boxplot(ToothGrowth$len ~ ToothGrowth$dose.fac)
```

- With the help of the R-commands written in the `ANOVA_with_chickwts.R` file, apply a analysis of variance (ANOVA) to the data set `ToothGrowth`

```
# aov.mod <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)
# What objects can we extract from a anova model?
objects(aov.mod)
#
summary(aov.mod)

# What are residuals?
ToothGrowth$residuals <- residuals(aov.mod)
tapply(ToothGrowth$len, ToothGrowth$dose.fac, mean)
ToothGrowth[c(1:3),]
# Save residuals to an objects and check mean of residuals
aov.mod.resid <- residuals(aov.mod)
mean(aov.mod.resid)
round(mean(aov.mod.resid), 3)

par(mfrow=c(1,1))
qqnorm(aov.mod.resid)
qqline(aov.mod.resid, col = "red", lwd = 3, lty = 2)
# Shapiro-Wilk test (dependent on sample size --> limited use)
shapiro.test(aov.mod.resid)
# a <- rnorm(100, 20, 3)
# qqnorm(a)
# qqplot(a)
# shapiro.test(a)

# Bartlett Test
bartlett.test(ToothGrowth$len ~ ToothGrowth$dose.fac)

# Levene's Test
```

```r
# install.packages("Rcmdr")
# library("Rcmdr")
# levene.test(ToothGrowth$len ~ ToothGrowth$dose.fac)

# Plot fitted against residual values
objects(aov.mod)
plot(fitted.values(aov.mod), residuals(aov.mod))

# Plot fitted against residual values
par(mfrow=c(1,2), pty="s", mar = c(1, 4, 1, 2))
plot(fitted.values(aov.mod), residuals(aov.mod))
abline(h = 0, col = "red", lwd = 3, lty = 2)
plot(aov.mod, which=1)

# Plot fitted against residual values
# Cut-off at 3 (y-axis)

# observations above 3 are regarded as having high
# influence to the analysis - have a closer look at them:
# outliers? delete them from the data set?
# why are these observations so influencial?
# everything below 3 is okay for the model
par(mfrow=c(1,1), pty="s", mar = c(5, 4, 4, 2))
plot(aov.mod, which=4)
# ToothGrowth[c(22, 23, 32),]

par(mfrow=c(1,3), pty="s")
plot(fitted.values(aov.mod), residuals(aov.mod))
abline(h = 0, col = "red", lwd = 3, lty = 2)
# Plot residuals against variables from the model
plot(ToothGrowth$len, residuals(aov.mod), ylab = "residuals")
plot(ToothGrowth$dose.fac, residuals(aov.mod),
     xlab = "ToothGrowth$dose.fac", ylab = "residuals")

par(mfrow=c(2, 2))
plot(aov.mod)
```

```
# # HOW TO RELEVEL FACTORS?
# # How to change the reference category of a factor variable?
# # Use the relevel(...) function
# # Make "sunflower" as reference category
# chickwts$feed <- relevel(chickwts$feed, "sunflower")
# levels(chickwts$feed)
# # Make "linseed" as reference category
# chickwts$feed <- relevel(chickwts$feed, "linseed")
# levels(chickwts$feed)
# chickwts$feed <- relevel(chickwts$feed, "casein")
# levels(chickwts$feed)

aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)

# aov.mod1 <- aov(len ~ dose.fac, data = ToothGrowth)
# aov.mod2 <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
# summary(aov.mod1)
# summary(aov.mod2)

# DO NOT USE THIS COMMAND, OTHERWISE THE LINEAR FUNCTION WITHIN
# DUNNETT AND TUKEY DOES NOT WORK!
# --> specify the data at the end of the aov model
# aov.mod <- aov(ToothGrowth$len ~ ToothGrowth$dose.fac)
summary(aov.mod)
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.fac, p.adj = "none")
pairwise.t.test(ToothGrowth$len, ToothGrowth$dose.fac, p.adj = "bonferroni")

# install the package first (one time)
# install.packages("multcomp")
# load the library (every single time you use it!)
library("multcomp")
# compares always to baseline levels (here: casein) --> saves degrees of freedom
# make sure you saved the aov.mod as:
# aov.mod <- aov(len ~ dose.fac, data = ToothGrowth)
dunnett <- glht(aov.mod, linfct = mcp(dose.fac = "Dunnett"))
```

```
summary(dunnett)


library("multcomp")
# compares all factor levels
tukey <- glht(aov.mod, linfct = mcp(dose.fac = "Tukey"))
summary(tukey)
# summary(tukey)          # standard display
tukey.cld <- cld(tukey)   # letter-based display
# the cld(...) function sets up a compact letter display of all pair-wise comparisons
?par
par(mfrow=c(1,1), mar=c(5,4,8,2))
plot(tukey.cld)
```

Exercise 16

- Download the .R file `LM_with_water.R` from the switch drive and have another look on how we applied the linear model to the `water` data set.

- Reuse these commands to fit a simple as well as multiple linear regression model to the data set of `perulung_ems`. Use `fev1` as your response variable y.

```
lung <- read.csv("~/Dropbox/data/perulung_ems.csv", sep = ";")
head(lung)
str(lung)
lung$sex <- factor(lung$sex, levels = c("0", "1"))
levels(lung$sex) <- c("female", "male")
lung$respsymptoms <- factor(lung$respsymptoms, levels = c("0", "1"))


# MODEL 1
# mod.age <- lm(fev1 ~ age, data = lung)
mod.age <- lm(lung$fev1 ~ lung$age)
summary(mod.age)
coef(mod.age)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age)
```

```
# MODEL 2
# mod.height <- lm(fev1 ~ height, data = lung)
mod.height <- lm(lung$fev1 ~ lung$height)
summary(mod.height)
coef(mod.height)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.height)

# MODEL 3
mod.age.height <- lm(fev1 ~ age + height, data = lung)
summary(mod.age.height)
coef(mod.age.height)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height)

# MODEL 4
mod.age.height.sex <- lm(fev1 ~ age + height + sex, data = lung)
summary(mod.age.height.sex)
coef(mod.age.height.sex)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height.sex)

# MODEL 5
mod.age.height.sex.resp <- lm(fev1 ~ age + height + sex + respsymptoms,
                              data = lung)
summary(mod.age.height.sex.resp)
coef(mod.age.height.sex.resp)
# Check model assumptions graphically
par(mfrow=c(2,2))
plot(mod.age.height.sex.resp)

mod1 <- lm(lung$fev1 ~ lung$age)
```

```
mod2 <- lm(lung$fev1 ~ lung$height)
mod3 <- lm(fev1 ~ age + height, data = lung)
mod4 <- lm(fev1 ~ age + height + sex, data = lung)
mod5 <- lm(fev1 ~ age + height + sex + respsymptoms,
           data = lung)
summary(mod5)

# MODEL SELECTION
AIC(mod1, mod2, mod3, mod4, mod5)
round(AIC(mod1, mod2, mod3, mod4, mod5), 2)
# Which of the models is best?
par(mfrow=c(2,2))
plot(mod5)
```

Exercise 17

- Load the `ToothGrowth` data set and run the following four linear regression models.

```
mod1 <- lm(len ~ dose.fac, data = ToothGrowth)
mod2 <- lm(len ~ supp, data = ToothGrowth)
mod3 <- lm(len ~ dose.fac + supp, data = ToothGrowth)
```

- Have a look at the summary of these models.

```
mod1 <- lm(len ~ dose.fac, data = ToothGrowth)
mod2 <- lm(len ~ supp, data = ToothGrowth)
mod3 <- lm(len ~ dose.fac + supp, data = ToothGrowth)
# mod4 <- lm(len ~ dose.fac + supp + dose.fac:supp, data = ToothGrowth)
mod4 <- lm(len ~ dose.fac*supp, data = ToothGrowth)
summary(mod1)
summary(mod2)
summary(mod3)
summary(mod4)
# Check model assumptions
par(mfrow=c(2, 2))
plot(mod1)
```

```r
plot(mod2)
plot(mod3)
plot(mod4)
```

- How do you interpret the model coefficients?
- Which model is best?

```r
AIC(mod1, mod2, mod3, mod4)
# t.test(ToothGrowth$len ~ ToothGrowth$supp) # not significant
# mod4 is the best model, because it has the smallest AIC.
# THE SMALLER THE AIC, THE BETTER THE MODEL!
```

Exercise 18

- Load the `water` data set and fit a multiple linear regression model. Use `mortality` as your response variable and add `hardness` and `location` as an explanatory variable.

```r
library("HSAUR3")
data("water")
str(water)
head(water)
lm.mod <- lm(mortality ~ hardness + location,
             data = water)
summary(lm.mod)
```

- Check the underlying model assumptions.

```r
par(mfrow=c(2,2))
plot(lm.mod)
```

- Add an interaction term between `hardness` and `location` to the above estimated multiple linear regression model.

```r
lm.mod.int <- lm(mortality ~ hardness + location + hardness:location,
             data = water)
```

```
summary(lm.mod.int)


mod1 <- lm(mortality ~ hardness + location,
             data = water)
mod2 <- lm(mortality ~ hardness + location + hardness:location,
             data = water)
AIC(mod1, mod2)
```

- Interpret the interaction coefficient `hardness:locationSouth`.
- Check the underlying model assumptions.

```
par(mfrow=c(2,2))
plot(lm.mod.int)
```

- Which one is the better model? With or without the interaction term?

```
AIC(lm.mod, lm.mod.int)
summary(lm.mod)
summary(lm.mod.int)
```

- How to derive confidence intervals for the regression coefficient of `hardness` and `location`?

```
confint(lm.mod)
```

Exercise 19

Hypothetical example - from Kirkwood and Sterne, Medical Statistics, 2nd ed., p. 177

- Read in the data set `lepto`. This study presents a serology survey of leptospira sero-prevalence in rural and urban areas of the west indies.

```
lepto <- read.csv("~/Dropbox/201710_Makerere/03_Exercises/data/lepto.csv", sep = ";")
# SONJA
lepto <- read.csv("C:\\Users\\admin\\Dropbox\\201710_Makerere\\03_Exercises\\data\\lepto.csv"
sep = ";")
head(lepto)
str(lepto)
```

- Encode the numeric variable `antibodies` as a factor with levels 0 and 1.

```r
table(lepto$antibodies)
class(lepto$antibodies)
lepto$antibodies <- factor(lepto$antibodies, level = c(0, 1),
                           labels = c("no", "yes"))
# Many different ways how to encode a numeric variable into a factor:
# lepto$antibodies <- factor(lepto$antibodies,
#                             levels = c(0, 1),
#                             labels = c("no", "yes"))
# lepto$antibodies <- factor(lepto$antibodies,
#                             levels = c(0, 1),
#                             labels = c("NO", "YES"))
# lepto$antibodies <- factor(lepto$antibodies,
#                             levels = c(0, 1),
#                             labels = c("Ugandian", "Kenian"))
table(lepto$antibodies)
class(lepto$antibodies)
```

- Make a crosstable with the risk factor `exposure` and `antibodies`.

```r
table(lepto$exposure, lepto$antibodies)
```

- Run a Chi-squared test, a Fisher's exact test and a logistic regression (`glm`) to assess if the `exposure` (living in rural vs. urban areas) is a risk factor.

```r
chisq.test(lepto$exposure, lepto$antibodies)
fisher.test(lepto$exposure, lepto$antibodies)
# fisher.test(table(lepto$exposure, lepto$antibodies))
glm.mod <- glm(antibodies ~ exposure, data = lepto,
               family = "binomial")
summary(glm.mod)
confint(glm.mod)
```

- Create a subset for `male` and `female` based on the variable `gender`.

```r
lepto.fem <- subset(lepto, gender == "female")
lepto.male <- subset(lepto, gender == "male")
```

- Repeat the crosstable, Chi-squared test, Fisher's exact test and a logistic regression (`glm`) for the subsets **separately**.

```r
# FEMALES
lepto.fem <- subset(lepto, gender == "female")
table(lepto.fem$exposure, lepto.fem$antibodies)
chisq.test(lepto.fem$exposure, lepto.fem$antibodies)
fisher.test(lepto.fem$exposure, lepto.fem$antibodies)
glm.mod.fem <- glm(antibodies ~ exposure, data = lepto.fem,
               family = "binomial")
summary(glm.mod.fem)
confint(glm.mod.fem)
# MALES
lepto.male <- subset(lepto, gender == "male")
table(lepto.male$exposure, lepto.male$antibodies)
chisq.test(lepto.male$exposure, lepto.male$antibodies)
fisher.test(lepto.male$exposure, lepto.male$antibodies)
glm.mod.male <- glm(antibodies ~ exposure, data = lepto.male,
               family = "binomial")
summary(glm.mod.male)
confint(glm.mod.male)
```

- Does the conclusion of your research question change with the analysis of the subsets? (Research question: Is the `exposure` (rural and urban areas) a risk factor?)
- Fit a logistic regression model (`glm`) with `exposure` and `gender` as explanatory variables.

```r
glm.mod.final <- glm(antibodies ~ exposure + gender, data = lepto,
               family = "binomial")
summary(glm.mod.final)
# Check that exposure and gender is also associated.
glm.exp.gen <- glm(exposure ~  gender, data = lepto,
               family = "binomial")
summary(glm.exp.gen)
```

- **SPECIAL FOR GUMA**: Is `exposure` being from an urban area a risk factor?