```
##################################################
### This practical provides you basic tools to ###
### conduct exploratory data analysis using R  ###
##################################################
# Clear workspace, (be careful, deletes all objects!)
rm(list=ls())

# Read air temperature data into R
# Modify relevant directory path
d <- read.csv("F:\\Opetus\\AAM2017\\Harjoitukset\\AirTemperatureData.csv", sep=";")

# Basic data checks
names(d)
str(d)

# Use summary()- function to get a quick overlook at your data
summary(d)
summary(d$temp) # Just one variable

### Let's practice of calculating "basic" statistics
# Measures of average
mean(d$elev)
median(d$elev)
round(mean(d$sea), 1) # report the mean using one decimal precision

# Measures of spread/deviation
var(d$temp) # variance
sd(d$temp) # standard deviation
sqrt(var(d$temp))

# range of variation
range(d$temp)

# Let's calculate the length of a range
minim <- min(d$temp) # minimum temperature
maxim <- max(d$temp) # maximum temperature
maxim - minim

# ... or use range() -function
range(d$temp)[2] - range(d$temp)[1]

# Let's plot histograms
hist(d$temp) # basic histogram
hist(d$temp, breaks = 30) # increase the number of breaks
hist(d$temp, breaks = c(0, 5, 10, 15)) # define your own breaks
hist(d$temp, main="Histogram", col="grey", xlab = "Air temperature")

# Add a line depicting mean and median values
abline(v=mean(d$temp), col="red", lty=2, lwd=2)
abline(v=median(d$temp), col="blue", lty=2, lwd=2)

# Calculate quantiles
quantile(d$temp)
quantile(d$temp, probs = 0.5) # 50th percentile i.e. median
quantile(d$temp, probs = seq(0, 1, 0.1))

# add lines depicting 2.5 % and 97.5 % points to the histogram
abline(v=quantile(d$temp, probs = 0.025)) # 2.5 %
abline(v=quantile(d$temp, probs = 0.975)) # 97.5 %

# Plot boxplots
boxplot(d$temp)
boxplot(d$temp, main = "Boxplot", ylab = "Air temperature",
        col = "yellow")
boxplot(d$temp, main = "Horizontal boxplot", xlab = "Air temperature",
        horizontal = TRUE, col = "yellow")

### Calculate statistics over groups
# First, let's create two elevation groups (as factors)
d$elev_class <- cut(d$elev, c(-0.1, 150, 1000))

# Rename the factor levels
levels(d$elev_class) <- c("low", "high")
```

```
# Plot the variation in air temperatures at each group
boxplot(d$temp ~ d$elev_class, ylab="Air temperatures")
boxplot(d$temp ~ d$elev_class, ylab="Air temperatures",
        col = heat.colors(2))

# Calculate statistics over the groups
tapply(d$temp, d$elev_class, mean) # mean
tapply(d$temp, d$elev_class, sd) # standard deviation

# Is the difference between the groups' mean values statistically significant?
t.test(d$temp~d$elev_class) # two sample, two sided t-test
t.test(d$temp~d$elev_class, alternative = "greater") # one-sided t-test

# An example of how to run a paired t-test
t.test(temperature_yesterday, temperature_today, paired = T) # not run

# Non-parametric Mann-whitney's U-test for non-normal data
wilcox.test(d$temp~d$elev_class) # same options available as with t-test

### Assessing statistical dependency
plot(d$elev, d$temp) # Scatter plot
cov(d$temp, d$elev) # Covariance between two variables, doesn't tell you much...
cor(d$temp, d$elev) # Pearson's correlation coefficient
cor(d$temp, d$elev, method = "spearman") # Spearman's rank correlation coefficient

# cor.test() -function quantifies the significance of the correlation
cor.test(d$temp, d$elev) # Pearson's correlation coefficient
cor.test(d$temp, d$elev, method = "spearman") # Spearman's rank correlation coefficient

# Pairwise correlation matrix
cor(d)
cor(d[, c("elev", "lake", "sea", "temp")]) # Pearson
cor(d[, c("elev", "lake", "sea", "temp")], method = "spearman") # Spearman
pairs(d[, c("elev", "lake", "sea", "temp")]) # Plot a scatter plot matrix
```