```
#################################################################
### The aim of this practical is to get familiar with multiple ###
### regression analysis using R                                ###
#################################################################
# Clear workspace
rm(list=ls())

# Read data in
d <- read.csv("F:\\Opetus\\AAM2017\\Harjoitukset\\AirTemperatureData.csv", sep=";")

# Test whether air temperature are related to elevation
model1 <- lm(temp ~ elev + I(elev^2), data=d)
anova(model1, test="F")

# Is the relationship positive or negative?
summary(model1)

# Test whether air temperature are related to lake variable
model2 <- lm(temp ~ lake + I(lake^2), data=d)
anova(model2, test="F")

# Is the relationship positive or negative?
summary(model2)

# How well geographical location (x and y) explain air temperatures?
model3 <- lm(temp ~ x + y + I(x^2) + I(y^2), data=d)
summary(model3)
anova(model3, test="F")

# Does the model improve if we add elevation into it?
# First, let's check if multicollinearity is going to cause problems...
# Calculte Spearmann's correlation between the predictors
cor(d[,c("elev", "x", "y")], method = "spearman")

model4 <- lm(temp ~ elev + I(elev^2) + x + y + I(x^2) + I(y^2), data=d)
summary(model4)
anova(model4, test="F")

# Predict air temperatures using different models
# Note that you must specify same predictors as in the models!
predict(model1, data.frame(elev=1000)) # model with only elevation
predict(model4, data.frame(elev=1000, x=max(d$x), y=min(d$y))) # Model with elevation, x &
y

### Modelling interactions ###
ia1 <- lm(temp~x:y, data=d); summary(ia1)

# Notice the difference due to syntax!
ia2 <- lm(temp~x*y, data=d); summary(ia2)

############################################
###### Let's model precipitation sums ######
############################################
d2 <- read.csv("F:\\Opetus\\AAM2017\\Harjoitukset\\PrecipitationData.csv", sep=";")

# Find the most parsimonious model explaining precipitation sums
# Based on first and second order polynomial terms
# using backward stepwise term selection
model5 <- lm(prec ~ elev + I(elev^2) + lake + I(lake^2) +
               sea + I(sea^2) + x + I(x^2) + y + I(y^2) +
               x:y, data=d2)
summary(model5)
anova(model5, test="F")

# Let's remove the most insignificant term (x:y interaction)
model6 <- lm(prec ~ elev + I(elev^2) + lake + I(lake^2) +
               sea + I(sea^2) + x + I(x^2) + y + I(y^2), data=d2)
summary(model6)
anova(model6, test="F")

# Omit sea^2
model7 <- lm(prec ~ elev + I(elev^2) + lake + I(lake^2) +
               sea + x + I(x^2) + y + I(y^2), data=d2)
summary(model7)
```

```
anova(model7, test="F")

# Omit sea
model8 <- lm(prec ~ elev + I(elev^2) + lake + I(lake^2) +
               x + I(x^2) + y + I(y^2), data=d2)
anova(model8, test="F")
summary(model8)

# Omit x^2
model9 <- lm(prec ~ elev + I(elev^2) + lake + I(lake^2) +
               x + y + I(y^2), data=d2)
anova(model9, test="F")
summary(model9)

# Omit lake^2
model10 <- lm(prec ~ elev + I(elev^2) + lake +
               x + y + I(y^2), data=d2)
anova(model10, test="F")
summary(model10)

# Omit lake
model11 <- lm(prec ~ elev + I(elev^2) +
                x + y + I(y^2), data=d2)
anova(model11, test="F")
summary(model11)
```