Tutorial 7. Statistical tests and linear models in R

In this tutorial we will learn how to specify statistical models and perform some statistical tests. We will use the stat_modelling.csv data set.

First load the data set:

```
stat.modelling <- read.table("stat_modelling.csv", sep = ",", head = T)
```

Check that the data are loaded correctly:

```
class(stat.modelling)
```

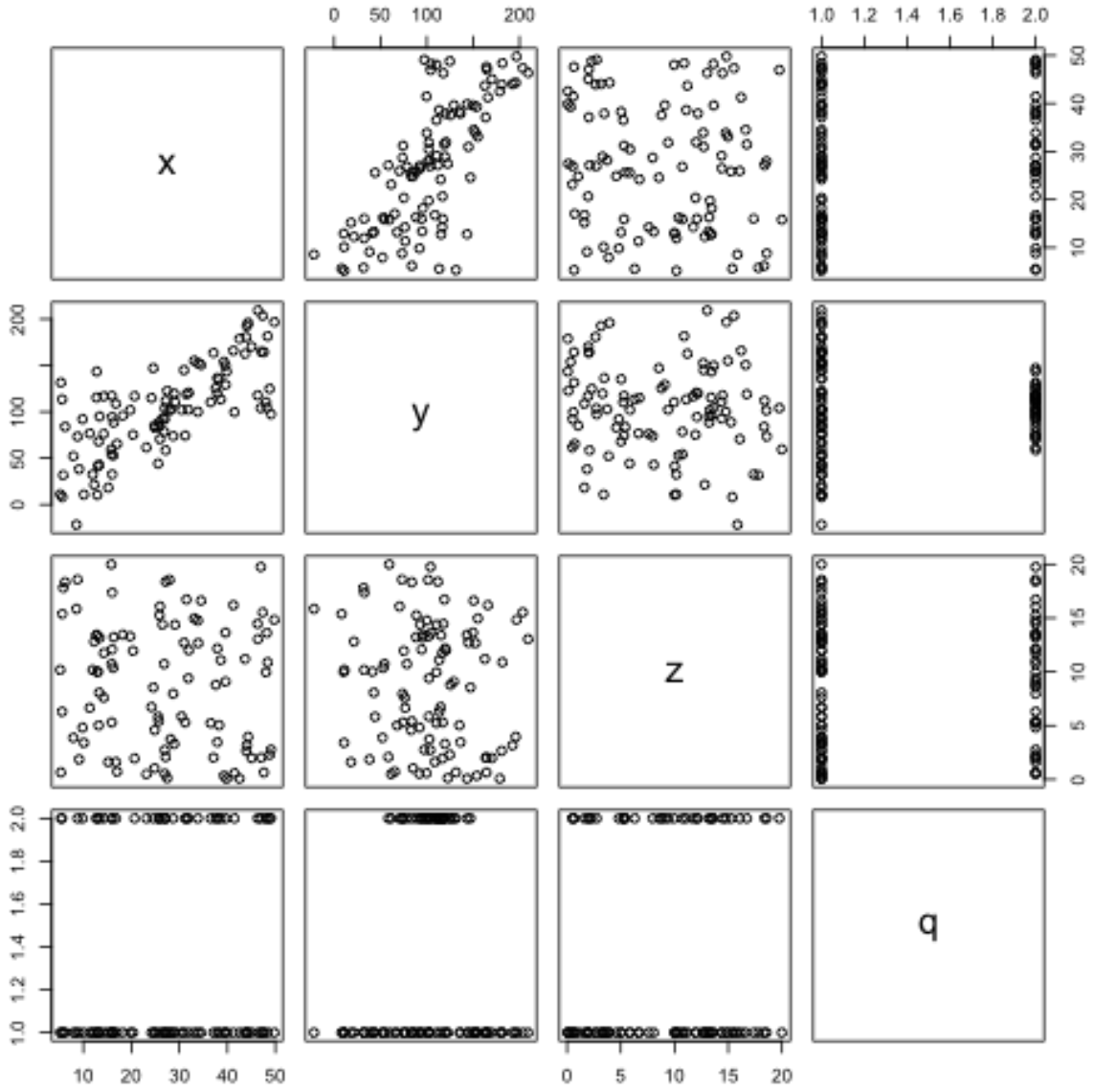
```
## [1] "data.frame"
```

```
str(stat.modelling)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ x: num  24.83 20.67 5.19 16.32 16.01 ...
##  $ y: num  85.1 116.9 11 88.3 32.5 ...
##  $ z: num  1.04 1.95 10.18 13.22 17.36 ...
##  $ q: Factor w/ 2 levels "A","B": 1 2 1 1 1 2 1 2 1 2 ...
```

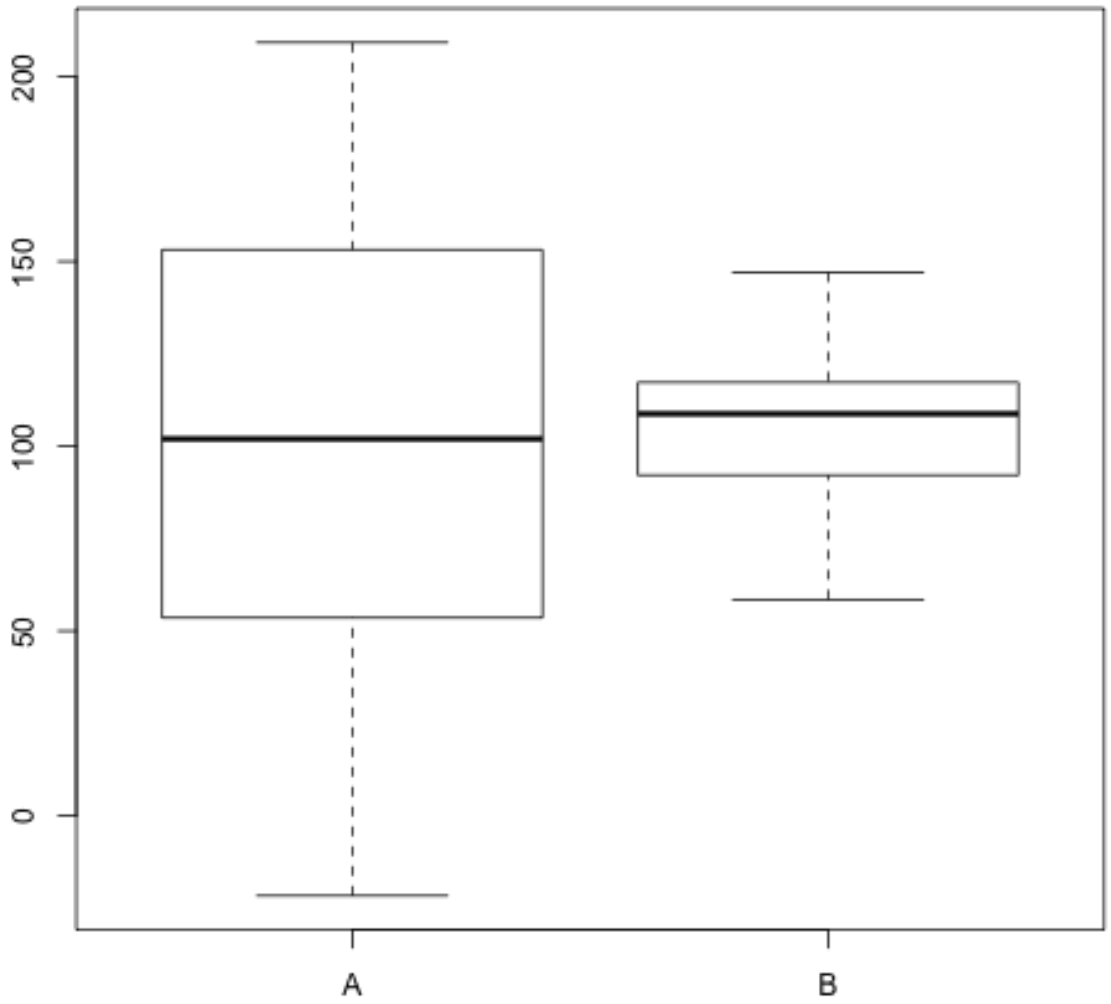
With data frames the plot function plots a pairs plot, which can be useful to inspect for possible association between variables:

```
plot(stat.modelling)
```



We will test whether the measurements of y for levels A and B of variable q are significantly different. This could be tested by specifying a linear model for an ANOVA or a t-test. We can inspect these data by specifying a formula object in the boxplot funcion:

```
boxplot(stat.modelling$y ~ stat.modelling$q)
```



It seems that the variance is different for A and B. This can be tested with the var.test function as follows:

```
# note that we need some indexing to select factors A and B because var.test
# uses two vectors:
y.A <- stat.modelling$y[stat.modelling$q == "A"]
y.B <- stat.modelling$y[stat.modelling$q == "B"]
var.test(y.A, y.B)
```

```
##
##  F test to compare two variances
##
## data:  y.A and y.B
## F = 7.52, num df = 62, denom df = 36, p-value = 4.454e-09
## alternative hypothesis: true ratio of variances is not equal to 1
```

```
## 95 percent confidence interval:
##   4.079 13.236
## sample estimates:
## ratio of variances
##               7.52
```

The variances are not equal, so it is more appropriate to use a t-test for unequal variances:

```
t.test(y.A, y.B, equal.var = F)
```

```
##
##  Welch Two Sample t-test
##
## data:  y.A and y.B
## t = -0.1376, df = 85.69, p-value = 0.8909
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -17.53  15.26
## sample estimates:
## mean of x mean of y
##     102.8     103.9
```

We could confim this with an ANOVA

This requires specifying a formula in the linear model, which is used in the anova function:

```
model.ab <- lm(stat.modelling$y ~ stat.modelling$q)
anova(model.ab)
```

```
## Analysis of Variance Table
##
## Response: stat.modelling$y
##                   Df Sum Sq Mean Sq F value Pr(>F)
## stat.modelling$q   1     30      30    0.01   0.91
## Residuals         98 233302    2381
```

Variables x and y are continuous, and there appears to be a linear relationship between them. We can conduct a linear regression as follows:

```
model.xy <- lm(stat.modelling$y ~ stat.modelling$x)
```

In this case we use the summary function to obtain the regression coefficients and their significance:

```
summary(model.xy)
```

```
##
## Call:
## lm(formula = stat.modelling$y ~ stat.modelling$x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -75.66 -21.94   1.19  22.48  86.02
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        31.251      7.570    4.13  7.7e-05 ***
## stat.modelling$x    2.659      0.251   10.59  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.3 on 98 degrees of freedom
## Multiple R-squared:  0.534,  Adjusted R-squared:  0.529
## F-statistic:  112 on 1 and 98 DF,  p-value: <2e-16
```

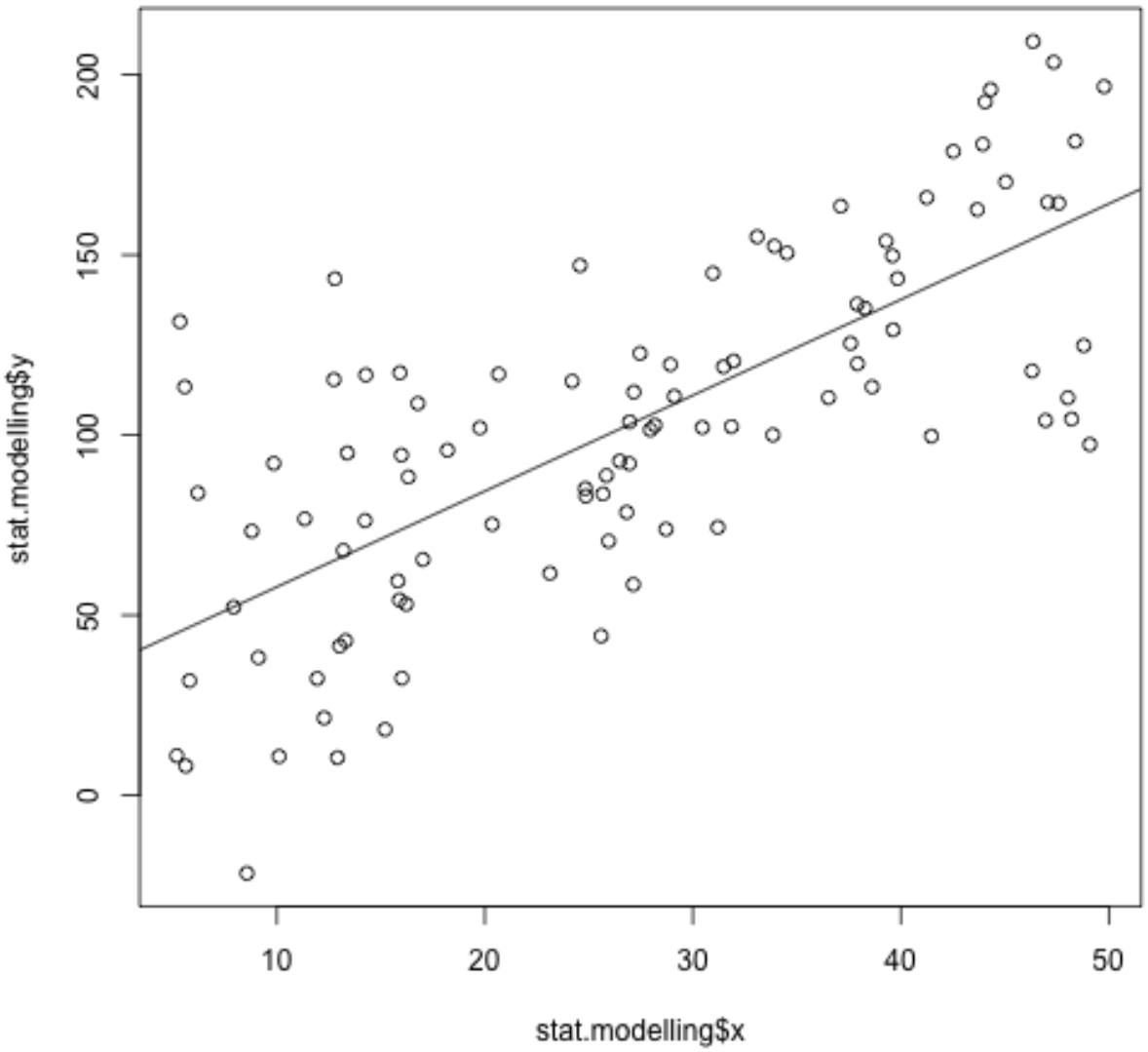
We are mostly interested in the output under *Coefficients*. The column *Estimates* is the estimated coefficients. In this case it is the intercept and the slope (*stat.modelling$x*). The next columns are the standard errors in the coefficients, the t statistic, and the P value. The null hypothesis is that the coefficients are not different from 0. The R squared and other statistics are also displayed.

The regression model (model.xy) can also be used for plotting. Use the following code for some diagnostic plots:

```
plot(model.xy)
```

An other use of the linear regression model object to showing the regression line:

```
plot(stat.modelling$x, stat.modelling$y)
abline(model.xy)
```


stat.modelling$y
stat.modelling$x
0
50
100
150
200
10
20
30
40
50