Applied Statistics and Data Analysis
Lab 4: Linear regression with a single predictor

Luca Grassetti and Paolo Vidoni
Department of Economics and Statistics, University of Udine

September, 2019

1 Example: roller data

The `roller` data set of the package `DAAG` is used to exemplify the main features of a regression analysis. Function `lm` fits a linear model with a single predictor or with multiple predictors. The main arguments are:

- `formula`, which is an object giving the symbolic description of the model to be fitted; in case of a simple linear regression, the specification is `response~predictor` (the intercept is implicitly considered and, to remove this, the formula is `response~predictor-1`);
- `data`, which gives the data frame, if required;
- `weights`, which is an optional argument giving the vector of weights to be used in the fitting procedure.

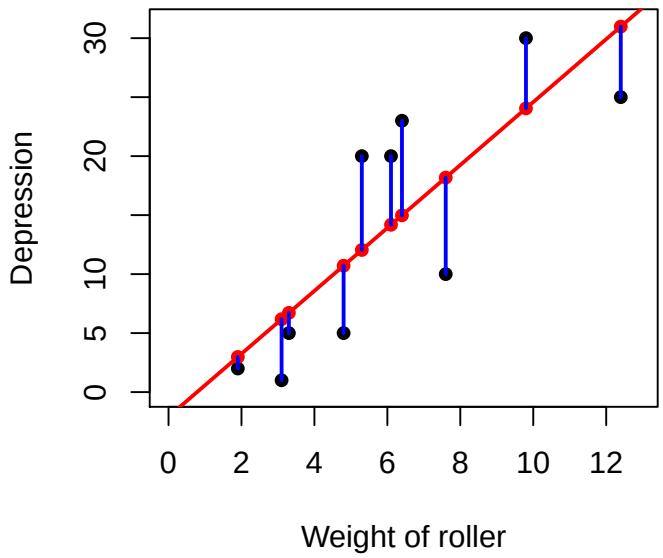
The function `lm` returns an object of class `lm`, which is a list containing a number of elements. In particular, `coefficients` gives the estimates of the parameters of the regression line, `fitted.values` specifies the fitted mean values, namely the prediction of the observed points on the fitted line, and `residuals` gives the observed residuals, namely the observed response minus the fitted values.

```
library(DAAG)
roller.lm <- lm(depression ~ weight,data=roller)
attributes(roller.lm)

$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
```

```
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"
```

At first we represent the data and the estimated regression line. Moreover, using function **ppoints**, we add to the existing graph the projections (in red) of the observed data points on the regression line. Indeed, function **segments** is used to represent the observed residuals as the blue segments connecting the observed data points and their projections on the regression line.

```
plot(depression ~ weight, data = roller,
     xlim=c(0,1.04*max(weight)),ylim=c(0,1.04*max(depression)),
     xlab = 'Weight of roller', ylab = 'Depression', pch = 16)
roller.lm <- lm(depression ~ weight,data=roller)
abline(roller.lm,col='red',lwd=2)
points(roller$weight,fitted.values(roller.lm), pch = 16,
       col="red", lwd=2)
segments(roller$weight,roller$depression,roller$weight,
         fitted.values(roller.lm), col="blue", lwd=2)
```



The results of the fitting procedure can be obtained by applying function **summary** to the object **roller.lm**.

```
attributes(summary(roller.lm))
```

```
$names
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

$class
[1] "summary.lm"
```

```
summary(roller.lm)
```

```

Call:
lm(formula = depression ~ weight, data = roller)

Residuals:
   Min     1Q Median     3Q    Max
-8.180 -5.580 -1.346  5.920  8.020

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.0871     4.7543  -0.439  0.67227
weight        2.6667     0.7002   3.808  0.00518 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.735 on 8 degrees of freedom
Multiple R-squared:  0.6445,Adjusted R-squared:  0.6001
F-statistic:  14.5 on 1 and 8 DF,  p-value: 0.005175
```

In particular, we obtain a summary of the vector of the observed residuals and the element `coefficients`, which gives the estimates of the regression parameters, with the associated standard errors, and the observed values of the t-test statistics for the nullity of the coefficients, with the corresponding p-values. In this case, the p-value for the slope is in accordance with the evident linear trend. Indeed, the residual standard error provides an estimate for the standard deviation σ of the error term, while the R-squared and the adjusted R-squared describe the proportion of the total variability of the response explained by the model. Finally, the result of the F-test is in accordance to the result of the testing procedure on the slope parameter.

The attributes of the object `summary(roller.lm)` can be used to obtain confidence intervals for the regression parameters. In particular, for the slope parameter, we consider the standard error given by the element `[2,2]` of the matrix `coefficients`.

```
SEb <- summary(roller.lm)$coefficients[2, 2]
SEb

[1] 0.7002426
```

Then, the 95% confidence interval for the slope parameter is simply obtained with the following command, where function `qt` is considered for computing the required Student's t quantiles.

```
coef(roller.lm)[2] + qt(c(0.025,.975), 8)*SEb

[1] 1.051984 4.281508
```

In order to obtain confidence intervals for the mean values, namely for the fitted values, and prediction intervals for the response variable, with specified values for the predictor variable, we may consider the R function `predict`. This function is a generic function for making predictions, based on a number of fitted models. The specific predictive procedure is determined by the object-class of the first argument. With regard to predictions and mean estimates based on a linear model, the main arguments are:

- `object`, which is an object of class `lm` giving the fitted model;
- `newdata`, which corresponds to the optional data frame giving values for the predictor variables to be used for prediction and estimation (if omitted, the original predictor values and the associated fitted values are considered);
- `se.fit`, which is a logical argument (the default is `FALSE`) indicating whether the standard errors are required as output;
- `interval`, with values `"none"`, `"confidence"` and `"prediction"`, which specifies the computation of confidence or prediction intervals (the default value is `"none"` and it specifies the computation of just the point estimates/predictions);
- `level`, which gives the confidence or the coverage level (the default value is `0.95`).

The function `predict` returns a list with the following elements:

- `fit`, which is a vector of point estimates/predictions (with the option `interval="none"`) or a matrix with column names `fit` (point estimates/predictions), `lwr` (interval lower bound), and `upr` (interval upper bound), if the confidence or prediction intervals are set;
- `se.fit`, which gives the standard error of the estimated means;
- `residual.scale`, which is the residual standard error and it provides an estimate for the standard deviation σ of the error term, if the original data are considered;
- `df`, which specifies the degrees of freedom for the residuals.

This function produces estimates/predictions obtained by evaluating the regression function in the data frame `newdata`. If `newdata` is omitted, the estimates/predictions are based on the data used for the fit of the model.

The estimates for the mean values, related to the original data, correspond to the fitted values saved in `roller.lm` and they can be computed, in the same way, using function `predict`. The associated estimated standard errors may be obtained with the option `se.fit=TRUE`.

```
roller.lm$fitted.values

       1        2        3        4        5        6        7
 2.979669  6.179765  6.713114 10.713233 12.046606 14.180002 14.980026
       8        9       10
18.180121 24.046962 30.980502

roller.pred <- predict(roller.lm, se.fit=TRUE)
roller.pred$fit

       1        2        3        4        5        6        7
 2.979669  6.179765  6.713114 10.713233 12.046606 14.180002 14.980026
       8        9       10
18.180121 24.046962 30.980502

roller.pred$se.fit

 [1] 3.614297 2.976896 2.880798 2.308147 2.197133 2.130050 2.142445
 [8] 2.384221 3.370270 4.917728
```

Differently, the standard errors of prediction include also the estimated variance of the random term and they can be obtained with the following command.

```
se.pred <- sqrt(roller.pred$se.fit^2+roller.pred$residual.scale^2)
se.pred

 [1] 7.643943 7.364009 7.325689 7.119990 7.084781 7.064265 7.068012
 [8] 7.145014 7.531629 8.339710
```

It is possible to represent the 95% pointwise confidence bounds for the mean values (that is, the fitted regression line) and the 95% pointwise prediction bounds for the response variable with respect to different values for the predictor variable. The procedure is based on the following steps:

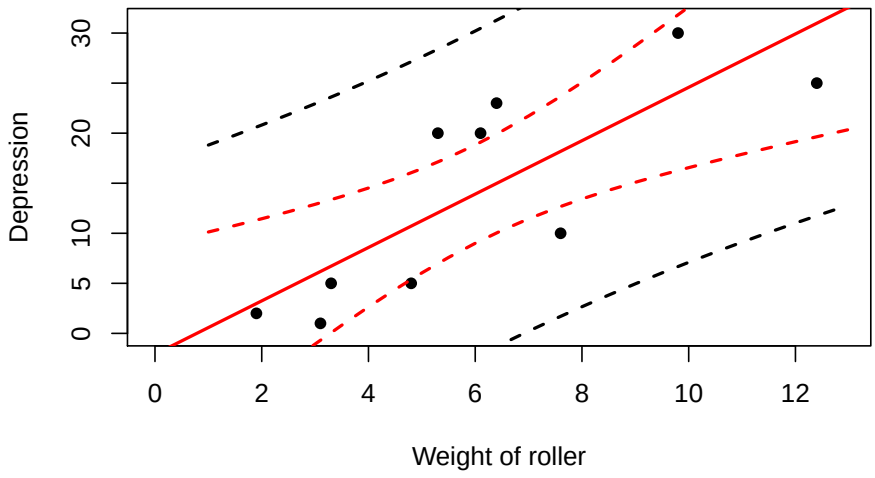
1. The scatterplot is obtained and the regression line, in red, is included in the graph;
2. A data frame, containing new values for the predictor variable `weight`, is created by considering points in the interval $[1, 13]$; function `pretty` is used to compute a sequence of $20 + 1$

equally spaced, round values, which cover the required range (the same result can be obtained by using `seq(1,13,0.5)`);

3. The point estimates for the mean values, associated to the new values of the predictor `weight`, are computed; the confidence bounds are also computed, and represented as red dashed curves;

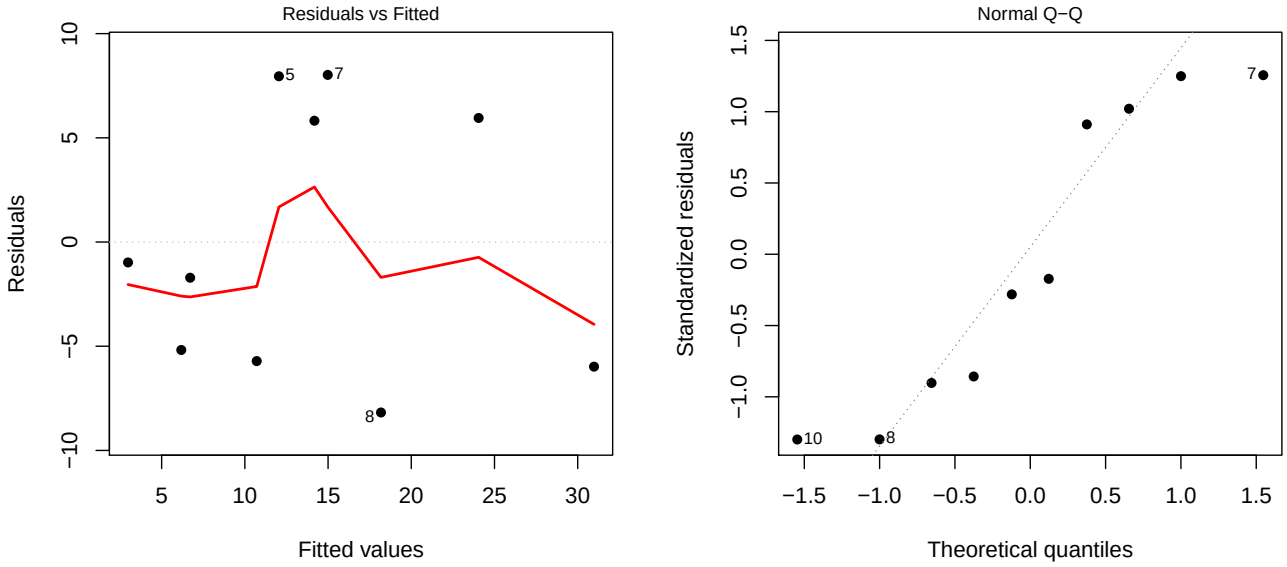
4. The point predictions for the response variable, associated to the new values of the predictor `weight`, are computed; the prediction bounds are also computed, and represented as black dashed curves.

```
# Step 1
plot(depression ~ weight, data = roller,
     xlim=c(0,1.04*max(weight)),ylim=c(0,1.04*max(depression)),
xlab = 'Weight of roller', ylab = 'Depression', pch = 16)
roller.lm <- lm(depression ~ weight,data=roller)
abline(roller.lm,col='red',lwd=2)
# Step 2
xy <- data.frame(weight = pretty(seq(1,13,1), 20))
# Step 3
yhat <- predict(roller.lm, newdata = xy, interval="confidence")
ci <- data.frame(lower=yhat[, "lwr"], upper=yhat[, "upr"])
lines(xy$weight, ci$lower, lty = 2, lwd=2, col="red")
lines(xy$weight, ci$upper, lty = 2, lwd=2, col="red")
# Step 4
yhatob <- predict(roller.lm, newdata = xy, interval="prediction")
ciob <- data.frame(lower=yhatob[, "lwr"], upper=yhatob[, "upr"])
lines(xy$weight, ciob$lower, lty = 2, lwd=2)
lines(xy$weight, ciob$upper, lty = 2, lwd=2)
```



In order to judge if the estimated model is adequate for the data, we may consider some graphical diagnostic tools. The `plot` function, applied to the object `roller.lm`, produces the well-known diagnostic plots. In particular, with the option `which=1`, we obtain the plot of the observed residuals against the fitted values, which is useful for checking the presence of systematic patterns. Indeed, the option `which=2` gives the normal probability plot for checking the normality assumption.

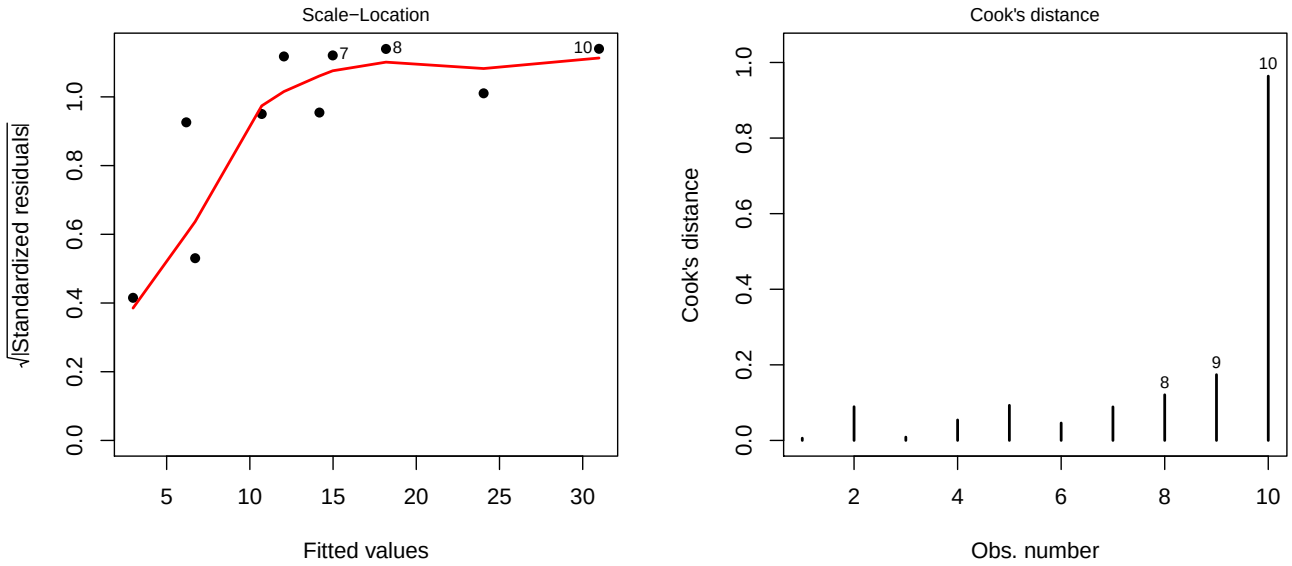
```
par(mfrow=c(1,2))
plot(roller.lm, which = 1, lwd=2, pch = 16, cex.caption=0.8)
plot(roller.lm, which = 2, xlab="Theoretical quantiles",
     lwd=2, pch = 16, cex.caption=0.8)
```




```
par(mfrow=c(1,1))
```

Two further diagnostic plots can be considered. The plot of the square root of absolute values of the observed residuals against the fitted values, for checking if the variance is constant (option `which=3`) and the plot with the Cook's distance (option `which=4`), for detecting influential points. Observation 10 has a large Cook's distance, although its residual is relatively small.

```
par(mfrow=c(1,2))
plot(roller.lm, which = 3, lwd=2, pch = 16, cex.caption=0.8)
plot(roller.lm, which = 4, lwd=2, pch = 16, cex.caption=0.8)
```




```
par(mfrow=c(1,1))
```

Finally, the adequacy of the fitted model can also be analyzed with the `anova` function, which gives the Analysis of Variance table. The variability of the response variable is decomposed in the part accounted for by the model and the residual part. Since the associated F-test presents a rather small p-value, the linear model with the predictor `weight` is a plausible model for describing the variability of the response variable.

```
anova(roller.lm)

Analysis of Variance Table

Response: depression
          Df Sum Sq Mean Sq F value   Pr(>F)
weight     1 657.97  657.97  14.503 0.005175 **
Residuals  8 362.93   45.37
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the table it is immediate to obtain the total sum of squares $657.97 + 362.93 = 1020.90$ and then the coefficient of determination $R^2 = 1 - 362.93/1020.90 = 0.64$. The adjusted version is obtained by dividing the sum of squares by the corresponding degrees of freedom, namely 8 and 9, and it corresponds to 0.60.

2 Example: paper resistance

We consider a data set on paper resistance and wood fibre concentration in a pulp. The aim is to study the relation between these two variables. There are 4 different levels of concentration (5%, 10%, 15%, 20%), and 6 trials are made at each level (so that the data are balanced). Firstly, a data frame `paper`, with the observed values of `resistance` and treatment `trt` (wood fibre concentration), is created. The data set includes a first variable that is defined as a vector of values concatenated by the `c` function. The treatment values are obtained with function `rep`, which is used to replicate 6 times the values included in the character vector `c("5\%", "10\%", "15\%", "20\%")`. Indeed, the treatment variable is redefined by using the `relevel` function. This function allows to define the reference level of a factor, which is used as a benchmark in all the subsequent analysis.

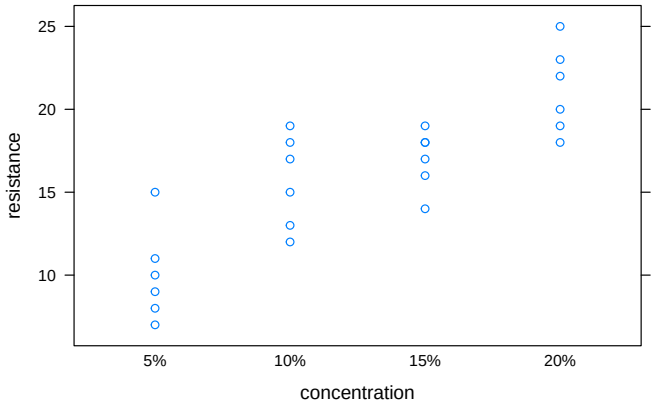
```
paper <- data.frame(resistance =
c(7, 8, 15,  11, 9, 10,# 5%
12, 17, 13, 18, 19, 15,# 10%
14, 18, 19, 17, 16, 18,# 15%
19, 25, 22, 23, 18, 20), # 20%
trt = rep(c("5%", "10%", "15%", "20%"),
c(6, 6, 6, 6)))
paper$trt <- relevel(paper$trt, ref="5%")
paper
```

```
   resistance trt
1           7  5%
2           8  5%
3          15  5%
4          11  5%
5           9  5%
6          10  5%
7          12 10%
8          17 10%
9          13 10%
10         18 10%
11         19 10%
12         15 10%
13         14 15%
14         18 15%
15         19 15%
16         17 15%
17         16 15%
18         18 15%
19         19 20%
20         25 20%
21         22 20%
22         23 20%
```

```
23         18 20%
24         20 20%
```

The `stripplot` function of the library `lattice` produces a one-dimensional scatterplot where the response variable is plotted conditional on a factor variable. The argument `aspect` is used to specify the shape of the panel (the default value is `1`, which corresponds to squared panels).

```
library(lattice)
stripplot(resistance~trt, aspect=0.6, data=paper, xlab="concentration",
          ylab="resistance")
```



The stripplot shows *within-group* variability and it gives an indication about the differences among the group means of the response variable `resistance`. In this case there is a single explanatory variable (predictor) `concentration`, which is treated as a factor with 4 different levels. Since the variances of the response variable, for the four different levels of `trt`, seem similar, we may use an analysis of variance (ANOVA) procedure in order to study how the mean level of a continuous response variable depends on the level of the factor. In fact, we consider a linear model with a categorical-type regressor. In general, one-way ANOVA is a set of techniques to compare the means of several groups, generalizing two-sample comparisons. Indeed, it can be extended to more than one factor, obtaining two-way or multi-way ANOVA procedures, where also the interaction among factors is taken int account.

The use of an ANOVA procedure (which, in this case, corresponds to the fitting of a particular linear model) enables an overall analysis of the potential differences in the means of the response variable due to the different treatments, namely the levels of the factor regressor. We usually assume that the observations of the response variable are independent and follow a Gaussian distribution with a common variance σ^2 and a mean $\mu + \tau_h$, where μ is the general mean and τ_h is the treatment effect related to the h-th group, $h = 1, \ldots, 4$.

By considering a suitable F-test, it is possible to test the null hypothesis that all the means are equal (that is, $\tau_h = 0$ for each h). The test statistic is specified as the ratio of two estimates of

the variance σ^2, namely that one obtained by the between-group sum of squares divided by the treatment degrees of freedom, and that one given by the residuals sum of squares, divided by the residual degrees of freedom.

In order to fit an ANOVA model to the data set `paper`, we use the function `aov`, which main arguments are similar to those of the function `lm`. The result is an object of class `c("aov","lm")`. The main difference from `lm` is that, applying functions `anova` or `summary` to the output, the results are represented in the traditional language of the analysis of variance (`Analysis of Variance Table`) rather than that of linear models.

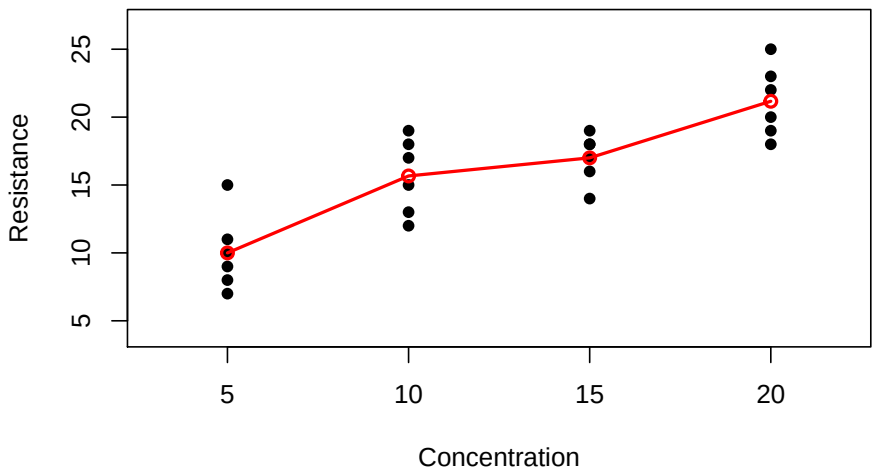
```
paper.aov <- aov(resistance~trt,data=paper)
anova(paper.aov) # the same result is given by summary(paper.aov)

Analysis of Variance Table

Response: resistance
          Df Sum Sq Mean Sq F value    Pr(>F)
trt        3 382.79 127.597  19.605 3.593e-06 ***
Residuals 20 130.17   6.508
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The observed value of the F-statistic gives a low p-value, leading to a substantial evidence against the null hypothesis. Then, the ANOVA test is significant, so that at least one mean must differ from the others. In this case, it is recommendable to investigate the results by considering a suitable post-hoc analysis in order to test which effects are significantly different.

The following lines of code are used to obtain the stripplot related to the data set (the treatment is now specified as the numerical vector `concentration`), to add the mean values of each group (using function `points`) and to draw the segments connecting them (using function `lines`). The mean values of the groups are obtained as the sum of the benchmark level (the mean in the first group) and the additional effects related to the other groups. These values are extracted from the element `coefficients` of the output given by function `summary.lm` applied to the object `paper.aov`.

```
concentration <- c(5, 5, 5, 5, 5, 5, 10, 10, 10, 10, 10, 10, 15, 15, 15, 15, 15,
         15, 20, 20, 20, 20, 20, 20) # treatment specified as numeric vector
plot(paper$resistance ~ concentration,xlab = 'Concentration', ylab = 'Resistance',
     xlim=c(3,22), ylim=c(4,27), pch = 16)

trt1 <- summary.lm(paper.aov)$coefficients[1,1] # mean first group
trt1

[1] 10
```

```
trt2 <- trt1 + summary.lm(paper.aov)$coefficients[2,1] # mean second group
trt2

[1] 15.66667

trt3 <- trt1 + summary.lm(paper.aov)$coefficients[3,1] # mean third group
trt3

[1] 17

trt4 <- trt1 + summary.lm(paper.aov)$coefficients[4,1] # mean fourth group
trt4

[1] 21.16667

points(c(5,10,15,20), c(trt1,trt2,trt3,trt4), col='red',lwd=2)
lines(c(5,10,15,20), c(trt1,trt2,trt3,trt4), col='red',lwd=2)
```

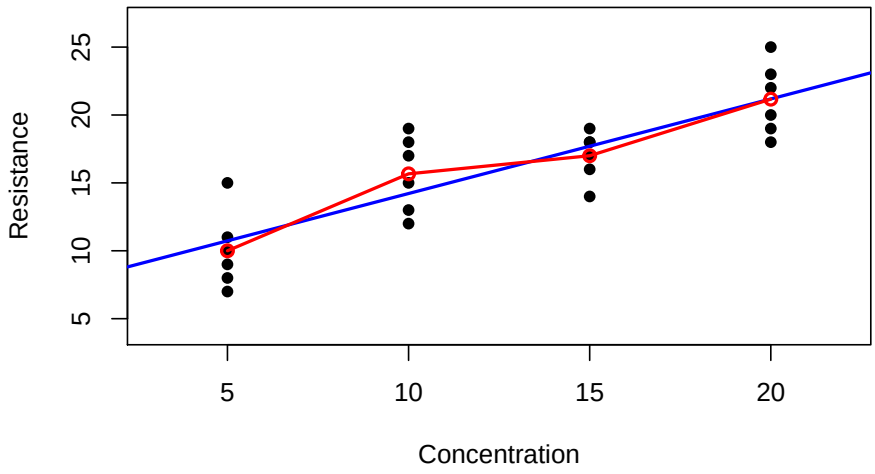


Finally, since in this application the levels of the factor `trt` are in fact quantitative, it is possible to consider the explanatory variable `concentration` as a numerical predictor in a simple linear regression model. In the ANOVA framework the statistical tests for differences between the treatment mean effects ignore the fact that the levels are quantitative. Thus, fitting a line or a curve, where this is possible, rather than fitting an analysis of variance model that has a separate parameter for each separate level of the explanatory variable, takes proper advantage of the data structure. Moreover, we may obtain a more convenient description for the pattern of the response variable and the testing result on the effectiveness of the linear trend is more powerful than that one given by an analysis of variance test, that treats the levels as qualitatively different levels (the p-values tend to be be smaller, on average).

```
paper.lm2 <- lm(paper$resistance ~ concentration)
summary(paper.lm2)


Call:
lm(formula = paper$resistance ~ concentration)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7333 -1.8458 -0.2167  1.4292  4.7833

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.25000    1.30100   5.573 1.33e-05 ***
concentration 0.69667    0.09501   7.332 2.43e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.602 on 22 degrees of freedom
Multiple R-squared:  0.7096,Adjusted R-squared:  0.6964
F-statistic: 53.76 on 1 and 22 DF,  p-value: 2.43e-07
```

The p-value for the slope parameter β is $2.43 \cdot 10^{-7}$, whereas that one found by the ANOVA analysis is $3.59 \cdot 10^{-6}$. Although both values suggest a strong relation, the p-value for the linear model is about 10 times smaller. However, in this case, the comparison between the two models show that the difference in terms of goodness of fit is minimal, as we may conclude by looking at the graphical comparison between the regression line in blue and the red line connecting the group means.

```
plot(paper$resistance ~ concentration,xlab = 'Concentration', ylab = 'Resistance',
     xlim=c(3,22), ylim=c(4,27), pch = 16)
abline(paper.lm2, col='blue',lwd=2)
points(c(5,10,15,20), c(trt1,trt2,trt3,trt4), col='red',lwd=2)
lines(c(5,10,15,20), c(trt1,trt2,trt3,trt4), col='red',lwd=2)
```



3 Example: cars

The `cars` data set, available in the R system libraries, contains data, recorded in the 1920s, on the speed of cars (mph) and on the distances taken to stop of 50 cars. This example is considered in order to discuss the linearity issue, with regard to the linear regression model. Firstly, the model is fitted and the inferential results are summarized.

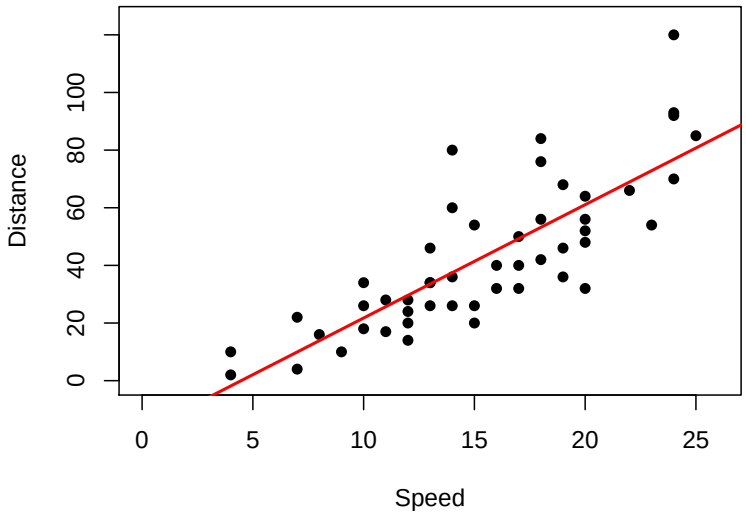
```
cars.lm <- lm(dist ~ speed, data = cars)
summary(cars.lm)
```

```

Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,Adjusted R-squared:  0.6438
```

```
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

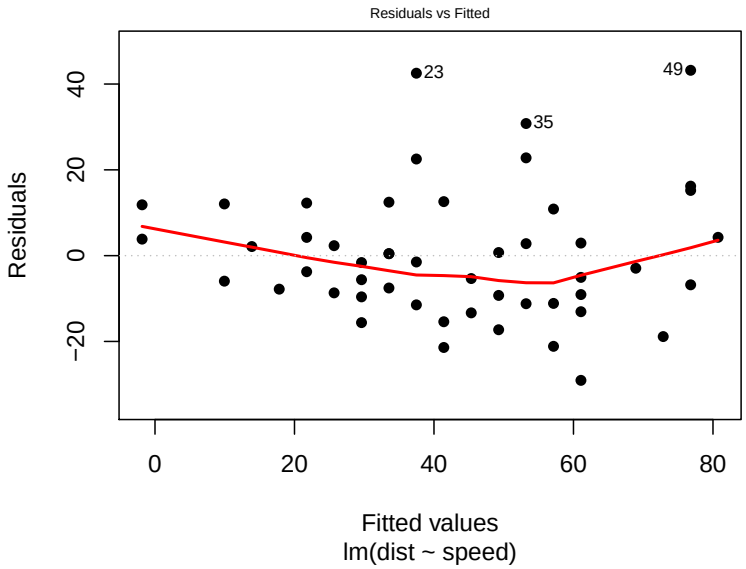
Although the p-values of the t-test on the slope parameter (and of the F-test) is closed to 0, the graphical representation below suggests a non-linear pattern in the data, which are also characterized by an increasing variability.

```
plot(dist ~ speed, data = cars, xlim=c(0,1.04*max(speed))
     ,ylim=c(0,1.04*max(dist)),xlab = 'Speed', ylab = 'Distance', pch = 16)
abline(cars.lm,col='red',lwd=2)
```



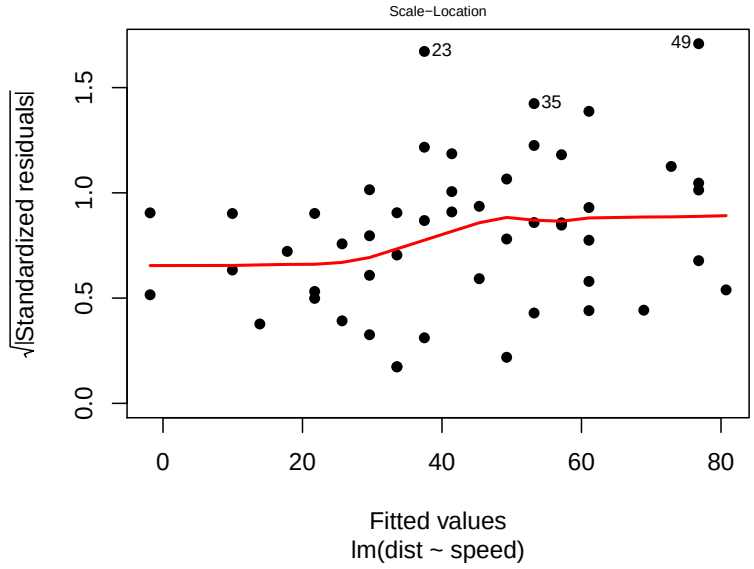
In order to evaluate more formally the adequacy of the model, we may consider the graphical diagnostic tools given by the `plot` function applied to the object `cars.lm`. The first plot, obtained with the option `which=1`, emphasizes that the observed residuals seem to be correlated, since there is a quadratic relation with the fitted values.

```
plot(cars.lm, which = 1, lwd=2, pch = 16, cex.caption=0.6)
```



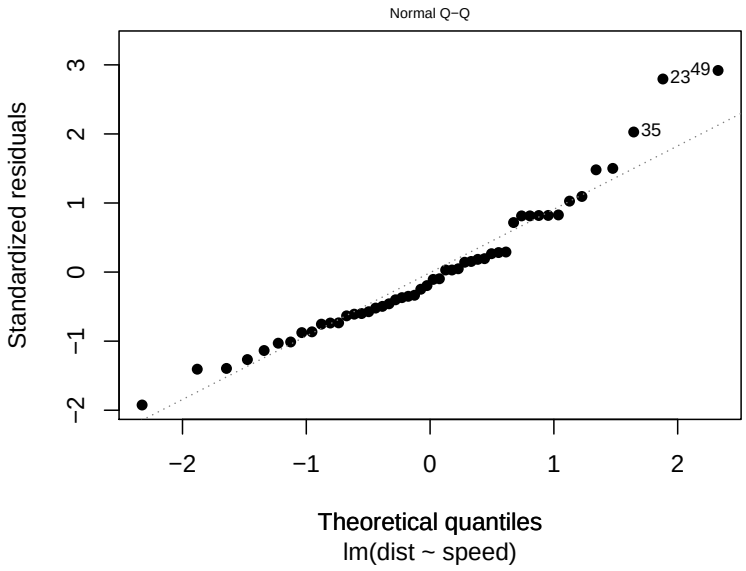
Indeed, the variance of the fitted residuals is non constant and this is also confirmed by the following diagnostic plot (obtained with the option `which=3`), where the square root of the standardized residuals is plotted against the fitted values.

```
plot(cars.lm, which = 3, lwd=2, pch = 16, cex.caption=0.6)
```



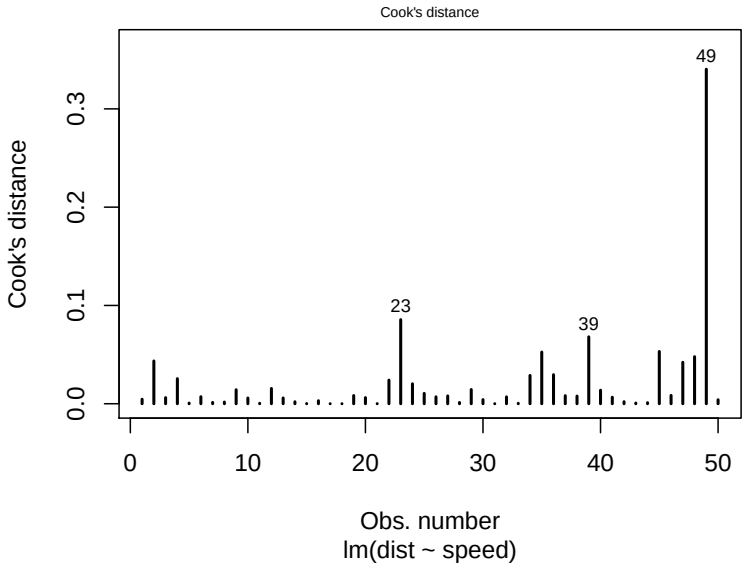
Moreover, the quantile-quantile plot (obtained with the option `which=2`) reveals a small evidence of skewness.

```
plot(cars.lm, which = 2, xlab="Theoretical quantiles",
     lwd=2, pch = 16, cex.caption=0.6)
```



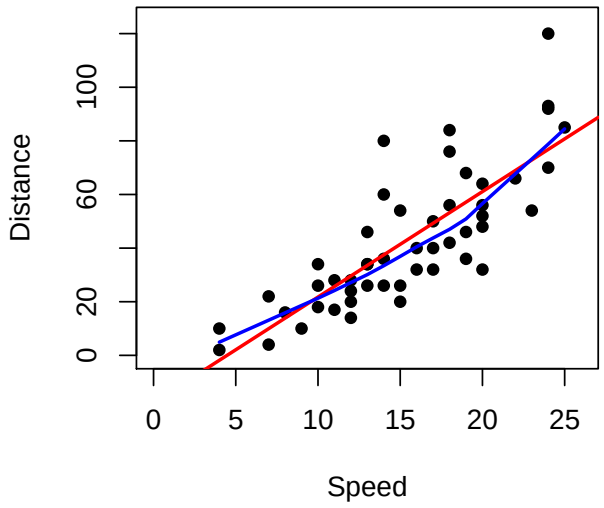
Finally, some outliers are identified by means of the Cook's distance (obtained with the option `which=4`). Although there are not points with a distance greater than 1, the 49-th observation is substantially larger than the others and this fact requires additional investigation since it could be an influential point.

```
plot(cars.lm, which = 4,lwd=2, pch = 16, cex.caption=0.6)
```



The results obtained by fitting a regression line to the `cars` data may be compared with those obtained by fitting a smooth non-linear curve. This objective can be achieved using function `lowess`, which uses locally-weighted polynomial regression in order to approximate the relationship between two variables. Its first two arguments are the vectors giving the coordinates of the points in the scatterplot, but alternatively they can be substituted by a `formula`, as for the `lm` function. Additional arguments are: `f`, which is the smoother span giving the proportion of points in the plot which influence the smooth at each value (the default value is `2/3` and larger values give more smoothness); `iter` and `delta`, which can be used, respectively, to adopt an iterative smoothing procedure and to define a different distance between two points to be considered in the smoothing function computation, in order to speed up calculation. The `lowess` function returns a list containing components `x` and `y` which give the coordinates of the smooth curve. Using function `lines` the curve can be added to an existing plot. With the following commands, a scatterplot with the fitted regression line in red and the fitted smooth curve in blue is obtained.

```
plot(dist ~ speed, data = cars, xlim=c(0,1.04*max(speed)),
     ylim=c(0,1.04*max(dist)),
xlab = 'Speed', ylab = 'Distance', pch = 16)
abline(cars.lm,col='red',lwd=2)
with(cars, lines(lowess(dist ~ speed, f=.7), lwd=2, col='blue'))
```



An alternative approach, in order to account for the non-linearity and the heteroschedasticity issues, is to consider a variance-stabilizing transformation for the response variable, such as the square-root. Then, a new linear model, with `sqrt(dist)` as response, is considered and compared with the original linear model. The comparison, based on the analysis of the results given by function `summary`, assures that the relation between `dist` and `speed` is stronger in the new model, since the p-values for the t-test and the F-test are lower and the values for the coefficients of determination increase.

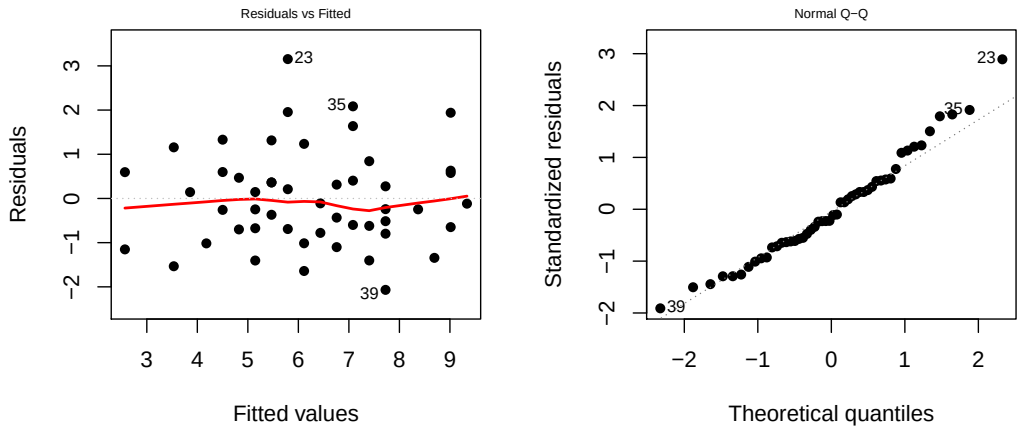
```
sqrtcars.lm <- lm(sqrt(dist) ~ speed, data = cars)
summary(cars.lm)


Call:
lm(formula = dist ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
speed         3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

summary(sqrtcars.lm)


Call:
lm(formula = sqrt(dist) ~ speed, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0684 -0.6983 -0.1799  0.5909  3.1534

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.27705    0.48444   2.636   0.0113 *
speed        0.32241    0.02978  10.825 1.77e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.102 on 48 degrees of freedom
Multiple R-squared:  0.7094,Adjusted R-squared:  0.7034
F-statistic: 117.2 on 1 and 48 DF,  p-value: 1.773e-14
```

The same evidence can be obtained by considering the ANOVA tables.

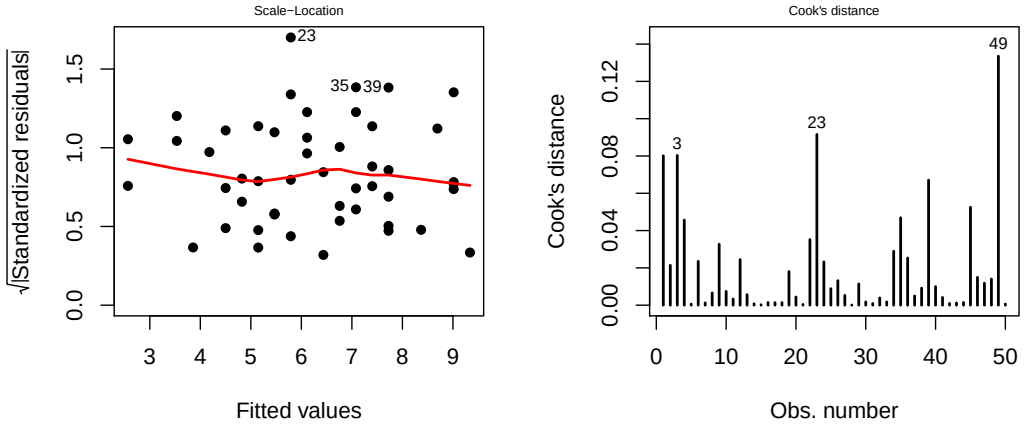
```
anova(cars.lm)

Analysis of Variance Table

Response: dist
          Df Sum Sq Mean Sq F value   Pr(>F)
speed      1  21186 21185.5  89.567 1.49e-12 ***
Residuals 48  11354   236.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(sqrtcars.lm)

Analysis of Variance Table

Response: sqrt(dist)
          Df  Sum Sq Mean Sq F value    Pr(>F)
speed      1 142.411 142.411  117.18 1.773e-14 ***
Residuals 48  58.334   1.215
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Moreover, the analysis of the diagnostic plots confirms the enhanced goodness of fit.

```
par(mfrow=c(1,2))
plot(sqrtcars.lm, which=1,lwd=2, pch = 16, cex.caption=0.6)
plot(sqrtcars.lm, which=2, xlab="Theoretical quantiles",
     lwd=2, pch = 16, cex.caption=0.6)
```

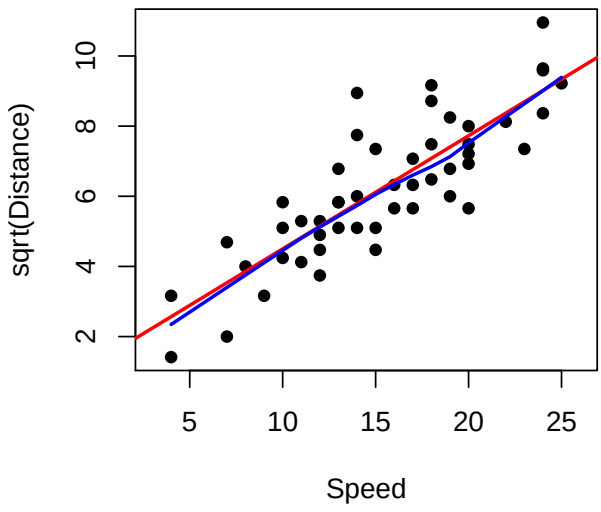


```
par(mfrow=c(1,2))
plot(sqrtcars.lm, which=3,lwd=2, pch = 16, cex.caption=0.6)
plot(sqrtcars.lm, which=4,lwd=2, pch = 16, cex.caption=0.6)
```



Finally, the graphical comparison of the fitted regression line with `sqrt(dist)` as response in red and the fitted smooth curve in blue shows that the difference is not as relevant as before.

```
plot(sqrt(dist) ~ speed, data = cars, xlim=c(3,26),
xlab = 'Speed', ylab = 'sqrt(Distance)', pch = 16)
abline(sqrtcars.lm,col='red',lwd=2)
with(cars, lines(lowess(sqrt(dist) ~ speed, f=.7),
                 lwd=2, col='blue'))
```

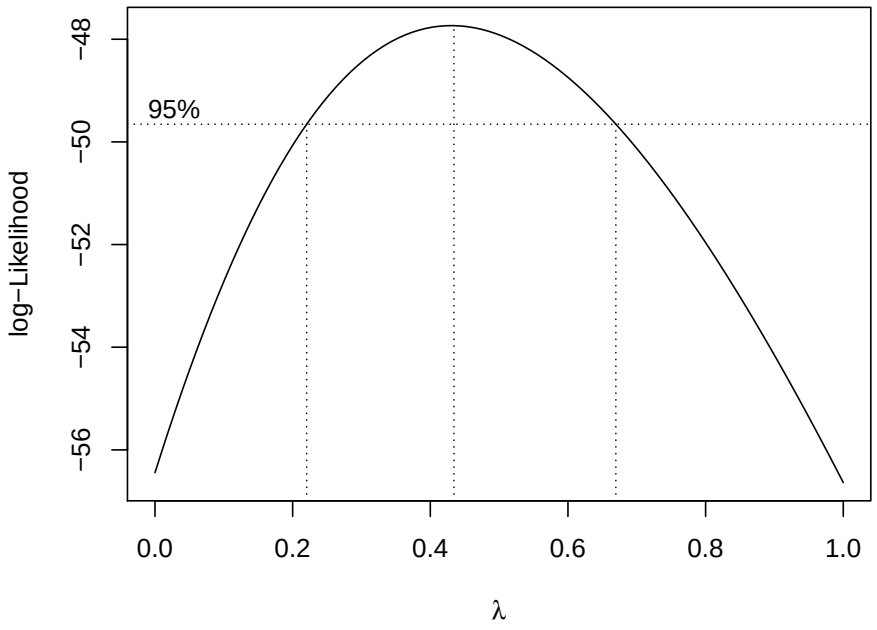


Choosing a suitable transformation for the response variable can not be easy and it might be convenient to consider semi-automatic procedures. Under this respect, the Box-Cox power transformation can be useful for linear models with positive responses. The `boxcox` function of the library `MASS` can be applied to an `lm` object (or to a `formula` object) in order to identify the value for the parameter λ which maximizes the (profile) log-likelihood for the transformed linear model. The main arguments are:

- `lambda`, which defines the vector of values of λ, where the profile log-likelihood is computed (the default value is `seq(-2,2,0.1)`);
- `interp`, which is a logical value determining the use of a spline interpolation for the lambda values computation (the default is `TRUE` if `lambda` has length less than 100);
- `plotit`, which is a logical value determining whether the results should be plotted (the default is `TRUE`).

The value of function `boxcox` is a list with the `lambda` vector and the computed profile log-likelihood vector. If `plotit=TRUE` the list is not visible and the outcome is a graphical representation of the profile log-likelihood as a function of `lambda`, with the indication of a 95% confidence interval about the maximum observed value of `lambda`.

```
library(MASS)
lambdares <- boxcox(cars.lm, lambda = seq(0, 1, 0.05))
```



In order to obtain the `lambda` vector and the computed profile log-likelihood vector, without interpolation and avoiding the graphical representation, we consider the following commands

```
lambdares <- boxcox(cars.lm, lambda = seq(0, 1, 0.05), plotit=F, interp=F)
lambdares

$x
 [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65
[15] 0.70 0.75 0.80 0.85 0.90 0.95 1.00

$y
 [1] -56.44525 -54.44791 -52.71917 -51.25976 -50.06651 -49.13255 -48.44768
 [8] -47.99897 -47.77145 -47.74884 -47.91427 -48.25086 -48.74223 -49.37287
[15] -50.12839 -50.99563 -51.96275 -53.01918 -54.15560 -55.36382 -56.63671
```

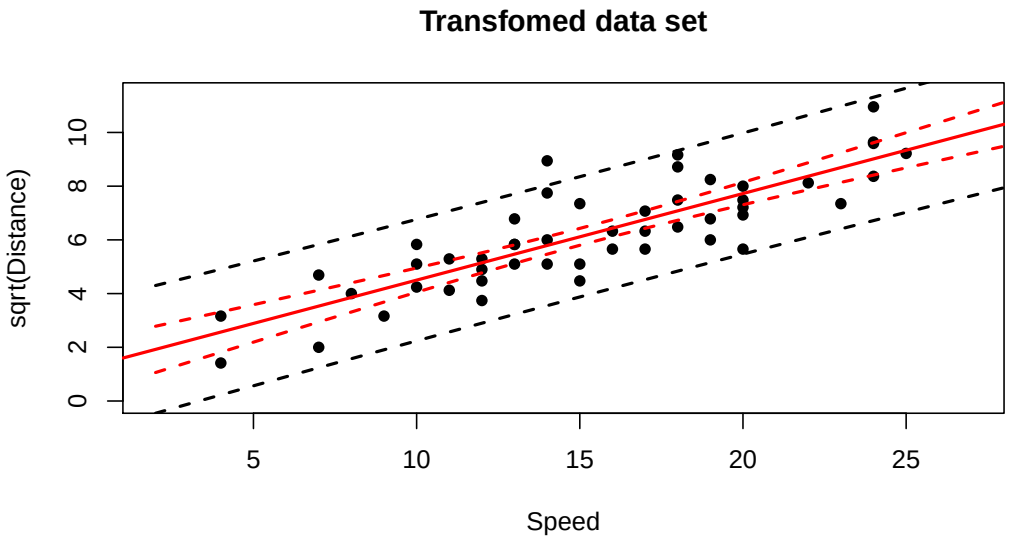
We find out that the value $\lambda = 0.5$, which corresponds to the square-root transformation, is among the most supported values.

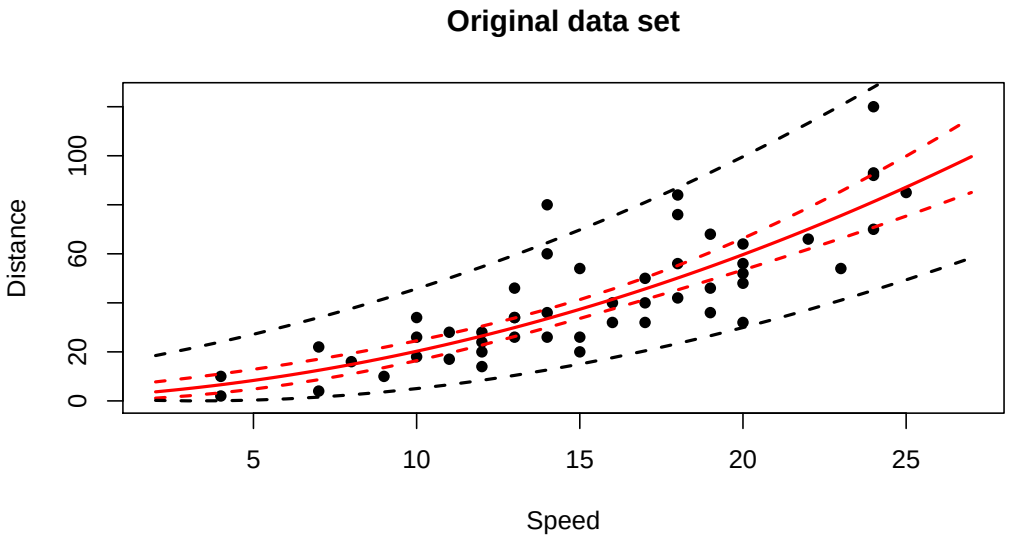
For the linear model taking `sqrt(dist)` as response it is possible to obtain the 95% confidence intervals for the mean values, namely for the fitted values, and the 95% prediction intervals for the response variable, with specified values for the predictor variable `speed`. The results are reported both on the transformed scale and on the original scale. The latter choice gives a more effective representation of the interest phenomenon. The procedure considers the R function `predict` and it is based on the following steps:

1. The scatterplot is obtained and the regression line, in red, is included in the graph; the confidence intervals (red dashed lines) and the prediction intervals (black dashed lines) are represented by specifying a set of commands similar to those defined for the `roller` data set;
2. The lower and upper bounds of the confidence and of the prediction intervals are converted in the original scale;
3. The original scatterplot is represented and the regression line is represented in the original scale, in red; the converted confidence intervals (red dashed lines) and the converted prediction intervals (black dashed lines) are drawn.

```
# Step 1
par(mfrow=c(2,1))
plot(sqrt(dist) ~ speed, data = cars, xlim=c(2,27),
     ylim=c(0,1.04*max(sqrt(dist))), xlab = 'Speed',
     ylab = 'sqrt(Distance)', pch = 16, main="Transfomed data set")
sqrtcars.lm <- lm(sqrt(dist) ~ speed, data=cars)
abline(sqrtcars.lm,col='red',lwd=2) # regression line
xy <- data.frame(speed = pretty(seq(2,28,1), 25))
yhat <- predict(sqrtcars.lm, newdata = xy, interval="confidence") # ci
ci <- data.frame(lower=yhat[, "lwr"], upper=yhat[, "upr"])
```

```
lines(xy$speed, ci$lower, lty = 2, lwd=2, col="red")
lines(xy$speed, ci$upper, lty = 2, lwd=2, col="red")
yhatob <- predict(sqrtcars.lm, newdata = xy, interval="prediction") # pi
ciob <- data.frame(lower=yhatob[, "lwr"], upper=yhatob[, "upr"])
lines(xy$speed, ciob$lower, lty = 2, lwd=2)
lines(xy$speed, ciob$upper, lty = 2, lwd=2)
# Step 2
xy <- data.frame(speed = pretty(seq(2,27,1), 25))
yhat <- predict(sqrtcars.lm, newdata = xy, interval="confidence")
yhat <- yhat^2 # converted ci
ci <- data.frame(lower=yhat[, "lwr"], upper=yhat[, "upr"])
yhatob <- predict(sqrtcars.lm, newdata = xy, interval="prediction")
yhatob <- yhatob^2 # converted pi
ciob <- data.frame(lower=yhatob[, "lwr"], upper=yhatob[, "upr"])
# Step 3
plot(dist ~ speed, data = cars, xlim=c(2,27),ylim=c(0,1.04*max(dist)),
      xlab = 'Speed', ylab = 'Distance', pch = 16,
      main="Original data set") # original scatterplot
lines(seq(2,27,0.05),(sqrtcars.lm$coefficients[1]+
      seq(2,27,0.05)*sqrtcars.lm$coefficients[2])^2,
      col='red',lwd=2) # converted regression line
lines(xy$speed, ci$lower, lty = 2, lwd=2, col="red") # converted ci
lines(xy$speed, ci$upper, lty = 2, lwd=2, col="red")
lines(xy$speed, ciob$lower, lty = 2, lwd=2) # converted pi
lines(xy$speed, ciob$upper, lty = 2, lwd=2)
```



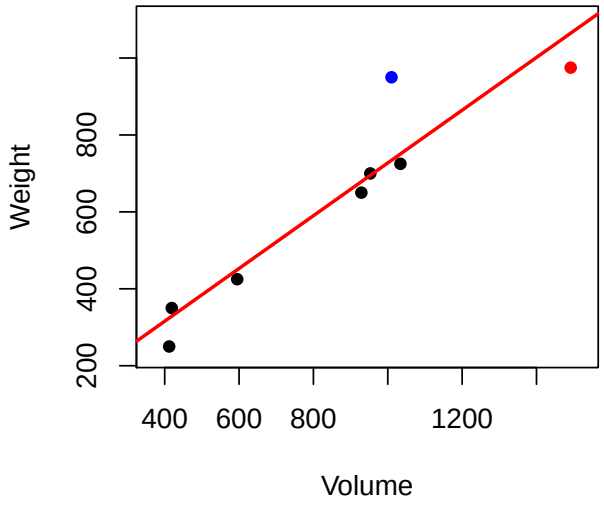



```
par(mfrow=c(1,1))
```

4 Example: books

The measurements of the volume (cm^3) and of the weight (gr) of 8 paperback books are given in the data set `softbacks` of the library `DAAG`. Two observations present a large influence on the model estimation: observation 4 (the red point in the scatterplot) and observation 6 (the blue point in the scatterplot).

```
library(DAAG)
softbacks.lm <- lm(weight ~ volume, data=softbacks)
plot(softbacks$volume[-c(4,6)], softbacks$weight[-c(4,6)], xlab = 'Volume',
     ylab = 'Weight', xlim=c(370,1520), ylim=c(230,1100), pch = 16)
points(softbacks$volume[4], softbacks$weight[4], pch=16, lwd=2, col='red')
points(softbacks$volume[6], softbacks$weight[6], pch=16, lwd=2, col='blue')
abline(softbacks.lm,col='red',lwd=2)
```



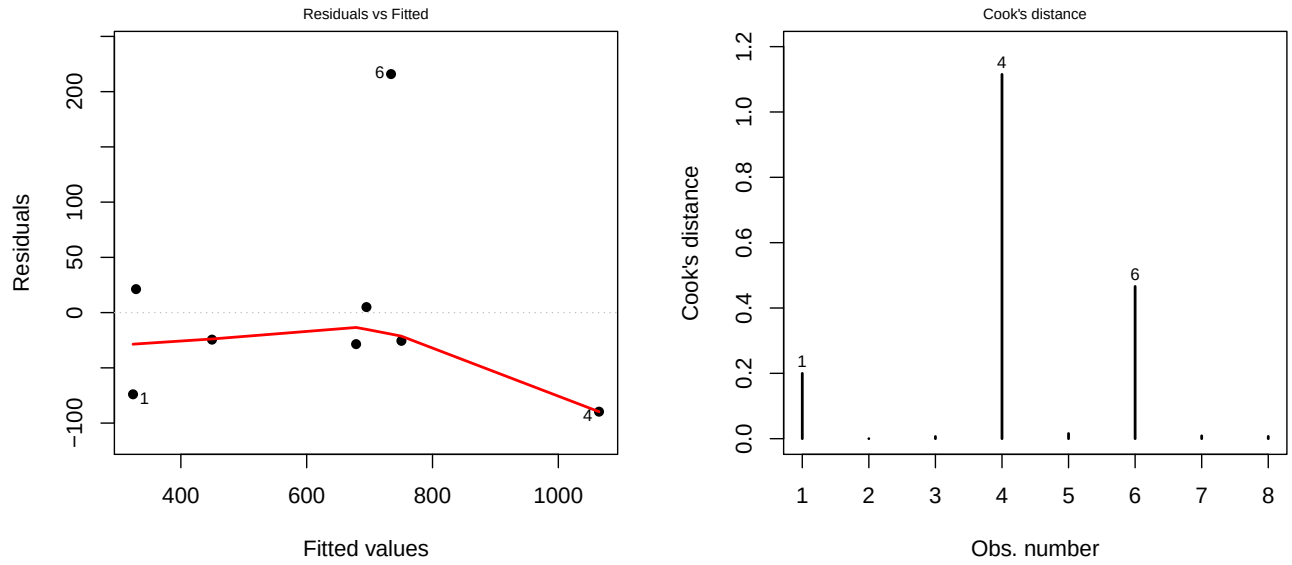
Indeed, the inferential results of the fitted model are summarized below.

```
summary(softbacks.lm)


Call:
lm(formula = weight ~ volume, data = softbacks)

Residuals:
    Min      1Q  Median      3Q     Max
-89.674 -39.888 -25.005   9.066 215.910

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.3725    97.5588   0.424 0.686293
volume        0.6859     0.1059   6.475 0.000644 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 102.2 on 6 degrees of freedom
Multiple R-squared:  0.8748,Adjusted R-squared:  0.8539
F-statistic: 41.92 on 1 and 6 DF,  p-value: 0.0006445
```

The following commands return the diagnostic plots specified by the options `which=1` and `which=4` and they are used to represent the magnitude of the residuals and the Cook's distance associated to each data point. In particular, this confirms that observation 4 (the red point in the scatterplot) and observation 6 (the blue point in the scatterplot) are influential points; observation 4 is also a leverage point.

```
par(mfrow=c(1,2))
plot(softbacks.lm, which = 1, lwd=2, pch = 16, cex.caption=0.7)
plot(softbacks.lm, which = 4, lwd=2, pch = 16, cex.caption=0.7)
```

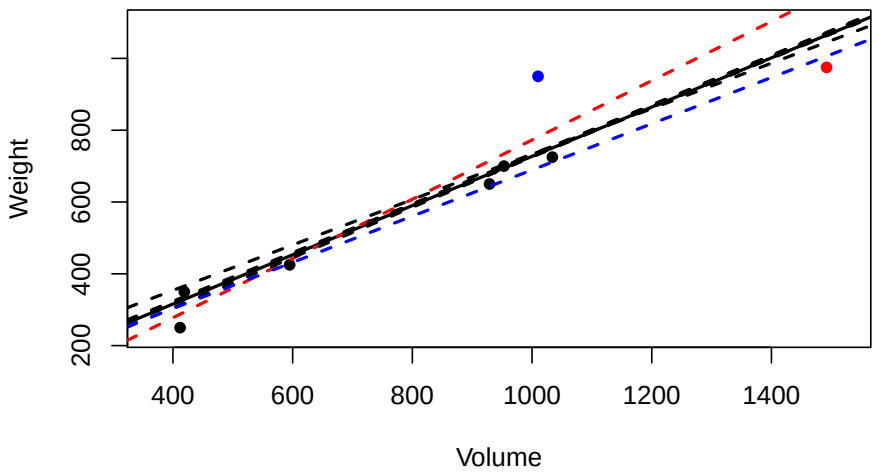



```
par(mfrow=c(1,1))
```

Although observation 6 has the largest residual, its Cook's distance is relatively small. On the other hand observation 4, which residual is not large, has the largest Cook's distance. In part, this is motivated by the fact that this point has a high leverage and, since its y-value is lower than would be predicted by the line, it pulls the line downward.

Points 4 and 6 are both candidates for omission, however with only eight observations, it would not make sense to omit any of them. Nevertheless, using a leave-one-out strategy, we may verify the behavior of the estimated linear model when a single observation is in turn omitted.

```r
softbacks.lm <- lm(weight ~ volume, data=softbacks)
plot(softbacks$volume[-c(4,6)], softbacks$weight[-c(4,6)],
     xlab = 'Volume', ylab = 'Weight', xlim=c(370,1520),
     ylim=c(230,1100), pch = 16)
points(softbacks$volume[4], softbacks$weight[4], pch=16,
       lwd=2, col='red')
points(softbacks$volume[6], softbacks$weight[6], pch=16,
       lwd=2, col='blue')
abline(softbacks.lm,col='black',lwd=2)
for(i in 1:8)
{
  cols <- 'black'
  cols <- ifelse(i==4, 'red', cols)
  cols <- ifelse(i==6, 'blue', cols)
  mod <- lm(weight ~ volume, data=softbacks[-i,])
  abline(mod, lty=2, lwd=2, col=cols)
}
```



The blue and red dashed lines correspond to the models estimated excluding the fourth and sixth observation respectively, while the black dashed lines represent the models estimated excluding the other observations. It is immediate to see that, as expected, observations 4 and 6 determine a substantial influence on the fitted model.