Applied Statistics and Data Analysis

Lab 5: Towards multiple linear regression and logistic regression

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

September, 2019

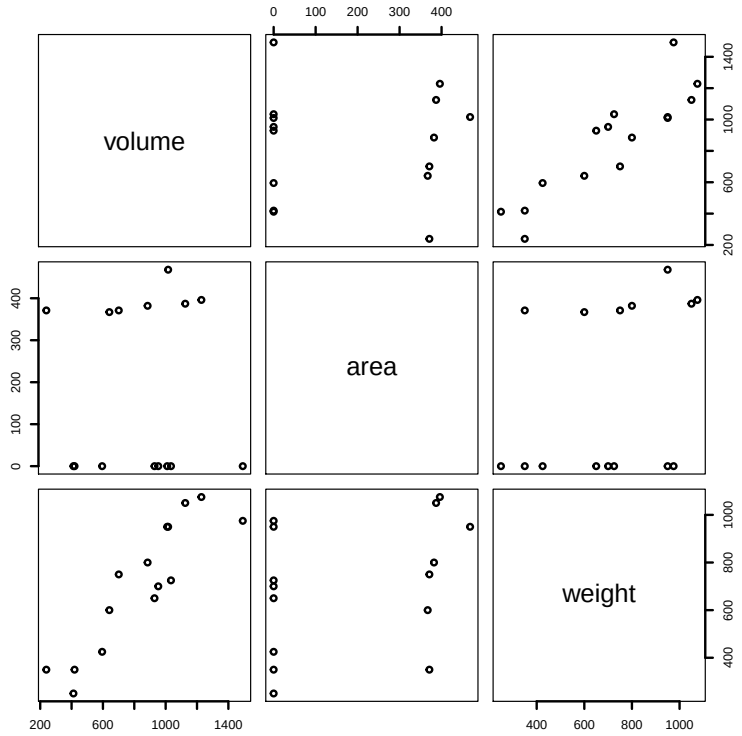
1 Multiple linear regression with continuous responses

1.1 Example: book weight

We consider the data frame `allbacks` of the library `DAAG` which contains measurements on a sample of 15 books, with regard to the variables `volume` (book volumes in cm^3), `area` (hard board cover area in cm^2), `weight` (book weights in gr) and `cover`, a factor with levels hardback (`hb`) and paperback (`pb`).

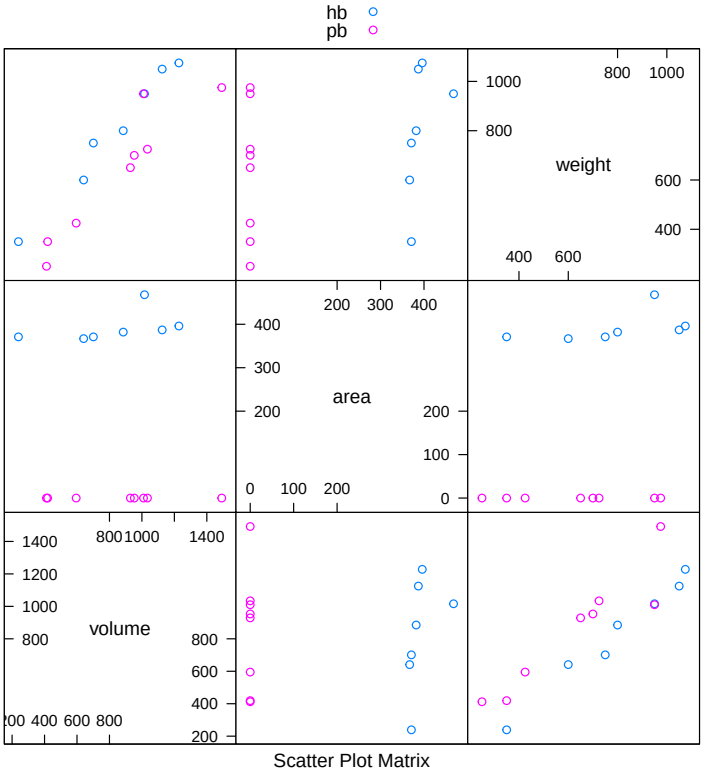
The scatterplot matrix for the three numerical variables is obtained using the function `pairs`. It gives a global representation, in a single graphical device, of the six scatterplots involving each pair of variables.

```
library(DAAG)
pairs(allbacks[,1:3],lwd=2)
```



An enhanced version of the plot can be obtained using the function `splom`, included in the `lattice` library. The first argument specifies the object on which the graphical representation is applied; in this case, the `formula` object `~allbacks[,1:3]` describes the structure and the content of the plot. Furthermore, the `groups` argument is used to specify the classification of the observations according to the factor `cover`, using different colors; the `auto.key = T` option is used to include the legend. The function `splom` is a general function for drawing conditional scatterplot matrices and many additional argument options may be considered.

```
library(lattice)
splom(~allbacks[,1:3],lwd=2, groups = cover,auto.key = T, data = allbacks)
```



Scatter Plot Matrix

A multiple linear regression model is defined in order to study the book `weight` as a function of the `volume` and the `area`; the factor `cover` is not considered, since it provides similar information to cover area. The function `lm` is used to fit also linear models with multiple predictors. In this case, the first argument is the `formula` object `weight ~volume + area`, which describes the model structure (with the intercept included as default). Notice that, in order to include more variables, it is sufficient to sum the additional variables. The result of the fitting procedure is obtained by applying the `summary` function to the `lm` object, which gives a similar output to that specified for simple linear models.

```
allbacks.lm <- lm(weight  ~ volume + area, data=allbacks)
summary(allbacks.lm)
```

```

Call:
lm(formula = weight ~ volume + area, data = allbacks)

Residuals:
    Min      1Q  Median      3Q     Max
-104.06  -30.02  -15.46   16.76  212.30

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 22.41342   58.40247   0.384 0.707858
volume       0.70821    0.06107  11.597 7.07e-08 ***
```

```
area         0.46843    0.10195   4.595 0.000616 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 77.66 on 12 degrees of freedom
Multiple R-squared:  0.9285,Adjusted R-squared:  0.9166
F-statistic: 77.89 on 2 and 12 DF,  p-value: 1.339e-07
```

The low p-values for `volume` and `area` emphasize that they are both important predictors of book `weight`, even if these results should be used informally, since, in case of multiple predictors, the model parameter estimators, and the associated t-tests, are usually not independent. Moreover, the F-test on the overall significance of the regression parameters presents a low p-value too. In this example, both the F-test and the individual t-tests on β_1 and β_2 are strongly significant. However, a significant F-test does not necessarily imply that all the individual t-tests are significant too. Finally, also the values for the multiple R^2 and the adjusted multiple R^2 confirm the goodness of the model.

The estimation results can also be studied by considering the `anova` function, which gives the ANOVA table, although the interpretation of the results requires great care.

```
anova(allbacks.lm)

Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value    Pr(>F)
volume     1 812132  812132 134.659  7.02e-08 ***
area       1 127328  127328  21.112 0.0006165 ***
Residuals 12  72373    6031
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the two F-tests describe, respectively, the contribution of `volume` after fitting the intercept and the contribution of `area` after fitting both the intercept and `volume`. Thus, the p-value for `area` agrees with that obtained using the t-test, since in both cases the model includes `volume`. On the other hand, the p-value for `volume` differs from that obtained using the t-test, because is computed without considering `area` (the difference, in this case, is low because the correlation between `volume` and `area` is close to zero).

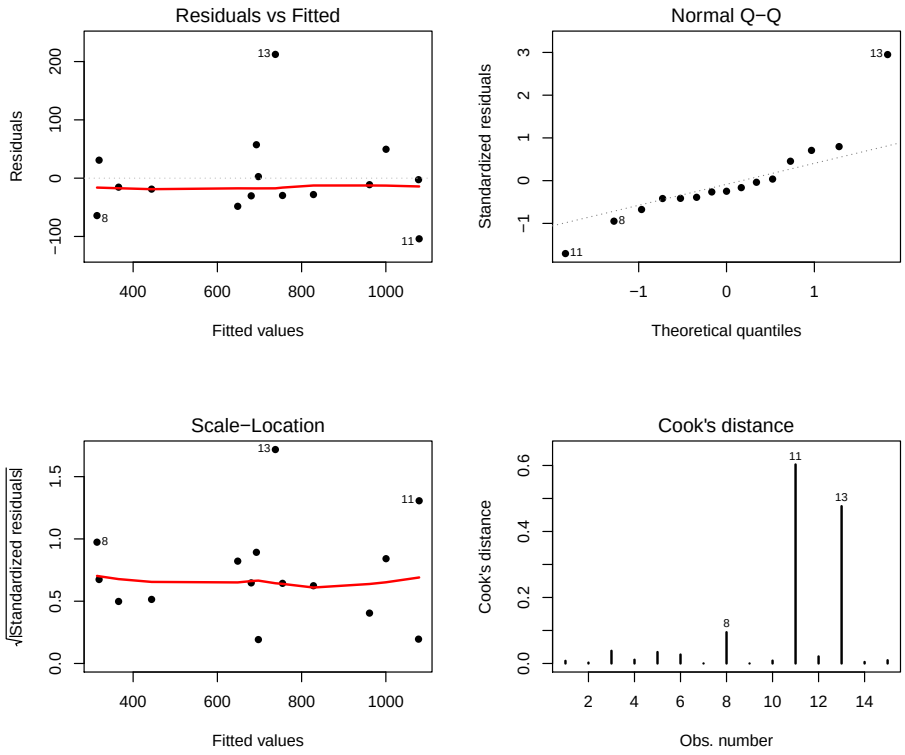
```
cor(allbacks$volume,allbacks$area)

[1] 0.001534791
```

The values for the variance inflation factor of the two explanatory variables confirm that the collinearity issue does not affect the present model. The variance inflation factor can be computed by considering the function `vif` of the library `car` or the function `vif` of the library `DAAG`. The alternative commands `DAAG::vif` and `car::vif` call, respectively, the functions of the `DAAG` and of the `car` library. This expedient is necessary when different libraries use the same name for a specific command.

```
library(car)
car::vif(allbacks.lm)

  volume     area
1.000002 1.000002

DAAG::vif(allbacks.lm)

volume   area
     1      1
```

As for simple linear regression, the diagnostic plots can be obtained by using the `plot` function, with the usual specification for the `which` argument.

```
par(mfrow=c(2,2))
plot(allbacks.lm, which = 1, lwd=2, pch = 16, cex.caption=1)
plot(allbacks.lm, which = 2, xlab="Theoretical quantiles",lwd=2, pch = 16,
     cex.caption=1)
plot(allbacks.lm, which = 3,lwd=2, pch = 16, cex.caption=1)
plot(allbacks.lm, which = 4,lwd=2, pch = 16, cex.caption=1)
```




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

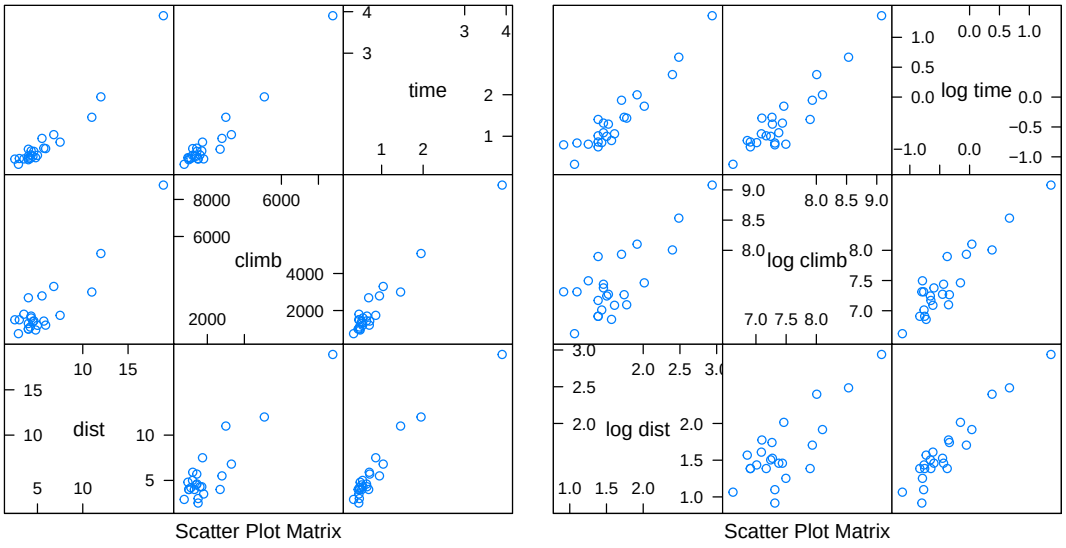
It immediate to see that observations 11 and 13 correspond to the largest residuals, and are influential points. However, they seem legitimate observations and with only 15 points it is better not remove them.

1.2 Example: hill races

The data frame `nihills`, of the library `DAAG`, contains data, obtained from the 2007 calendar for the Northern Ireland Mountain Running Association, on 23 hill races; the variables are: the distance `dist` (miles), the amount of climb `climb` (ft), the male record time `time` (hours) and the female record time `timef` (hours). Considering the male times only, the scatterplot matrices for the original data and for the logarithmic transformation of the data reveal some linear relationships. Taking the log data seems preferable, since the relationship between the pairs of variables is more clear. The two scatterplot matrices are obtained by considering the following steps:

- the two scatterplot matrices are defined using the function `splom` of the library `lattice`.;
- the first plot is printed on the left side of the graphical device, defined by `c(0, 0, 0.5, 1)`;
- the second plot is printed on right side of the graphical device, defined by `c(0.5, 0, 1, 1)`; the `newpage=F` option is used to add the plot to the existing graphical device.

```
library(DAAG)
library(lattice)
gr1 <- splom(~nihills[,c("dist","climb","time")],lwd=2)
gr2 <- splom(~log(nihills[,c("dist","climb","time")]),
             varnames=c("log dist", "log climb", "log time"),lwd=2)
print(gr1, position=c(0, 0, 0.5, 1))
print(gr2, position=c(0.5, 0, 1, 1),  newpage=FALSE)
```



The linear model can be estimated by considering the logarithmic transformation of the data; the formula specified in the first argument of function `lm` involves the `log` function applied to the response variable `time` and to the explanatory variables `dist` and `climb`.

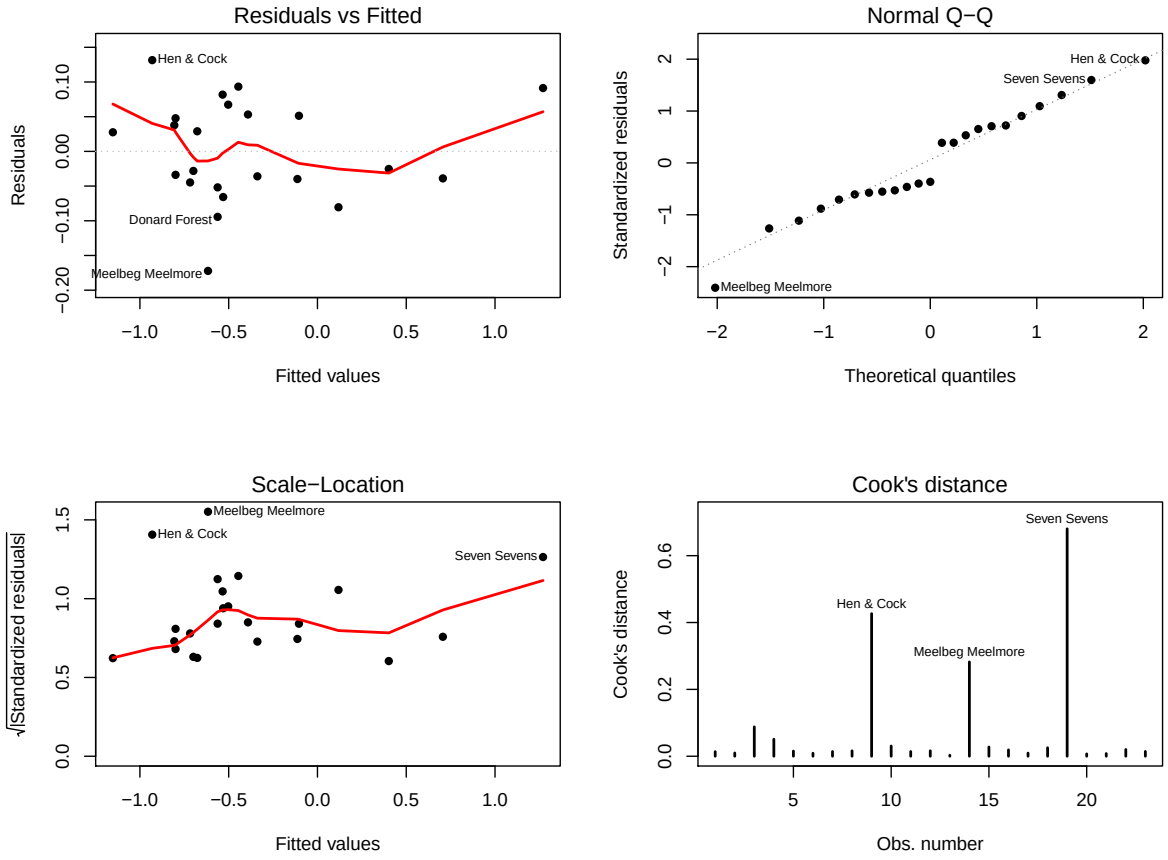
```
nihills.lm <- lm(log(time) ~ log(dist) + log(climb), data = nihills)
summary(nihills.lm)
```

```
Call:
lm(formula = log(time) ~ log(dist) + log(climb), data = nihills)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17223 -0.04229 -0.02538  0.05222  0.13150
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.96113    0.27387  -18.11 7.09e-14 ***
log(dist)    0.68136    0.05518   12.35 8.19e-11 ***
log(climb)   0.46576    0.04530   10.28 1.98e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0766 on 20 degrees of freedom
Multiple R-squared:  0.9831,Adjusted R-squared:  0.9814
F-statistic: 582.7 on 2 and 20 DF,  p-value: < 2.2e-16
```

The low p-values for `log(dist)` and `log(climb)` emphasize that they are both important predictors of `log(time)`. Indeed, the global significance of the regression coefficients is confirmed by the low p-value of the F-test. Moreover, the values for the multiple R^2 and the adjusted multiple R^2 confirm the goodness of the model. The main diagnostic plots can be obtained using, as usual, the function `plot` with specific `which` options. The graphical analysis does not reveal any problem, except a slight non-linear pattern for the residuals and a moderately large residual associated with the Meelbeg Meelmore race.

```
par(mfrow=c(2,2))
plot(nihills.lm, which = 1, lwd=2, pch = 16, cex.caption=1)
plot(nihills.lm, which = 2, xlab="Theoretical quantiles",lwd=2, pch = 16,
     cex.caption=1)
plot(nihills.lm, which = 3,lwd=2, pch = 16, cex.caption=1)
plot(nihills.lm, which = 4,lwd=2, pch = 16, cex.caption=1)
```




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

The same analysis could be performed by considering a new data frame called `lognihills`, obtained by applying the logarithmic transformation to the entire original data set. Here, the names of the columns are changed adding `log` to the former names.

```
lognihills <- log(nihills)
names(lognihills) <- paste("log", names(nihills), sep="")
str(lognihills)

'data.frame': 23 obs. of  4 variables:
 $ logdist : num  2.01 1.44 1.77 1.92 1.61 ...
 $ logclimb: num  7.46 7.01 7.1 8.1 7.09 ...
 $ logtime : num  -0.1528 -0.7621 -0.3523 0.0379 -0.6141 ...
 $ logtimef: num  0.0625 -0.4731 -0.12 0.1941 -0.4502 ...
```

A new regression model, which is equivalent to the previous one, can be defined by considering the explanatory variable log(`climb/dist`), instead of log(`climb`). This new variable is added to the data set `lognihills` and the corresponding linear model is estimated.

```
lognihills$logGrad <- with(nihills, log(climb/dist))
nihillsG.lm <- lm(logtime ~ logdist + logGrad, data=lognihills)
summary(nihillsG.lm)


Call:
lm(formula = logtime ~ logdist + logGrad, data = lognihills)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17223 -0.04229 -0.02538  0.05222  0.13150

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.9611     0.2739  -18.11 7.09e-14 ***
logdist       1.1471     0.0346   33.16  < 2e-16 ***
logGrad       0.4658     0.0453   10.28 1.98e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0766 on 20 degrees of freedom
Multiple R-squared:  0.9831,Adjusted R-squared:  0.9814
F-statistic: 582.7 on 2 and 20 DF,  p-value: < 2.2e-16
```

Notice that the two models provide the same fit (the p-value for the F-test and the values for the multiple R^2 and the adjusted multiple R^2 are the same), since they are different mathematical formulations of the same underlying model. Interpretation issues and application-specific considerations will drive the choice of a particular model form. In the second model, the amount of climb is evaluated with respect to the distance. A further benefit, in addition to interpretation, is that the correlation between the two regressors is very low. Although the original model specification does not present multicollinarity, the interpretation of the coefficients is very difficult, given the high correlation between `climb` and `dist`.

```
with(lognihills, cor(cbind(logGrad, logclimb), logdist))

               [,1]
logGrad  -0.06529222
logclimb  0.78006703
```

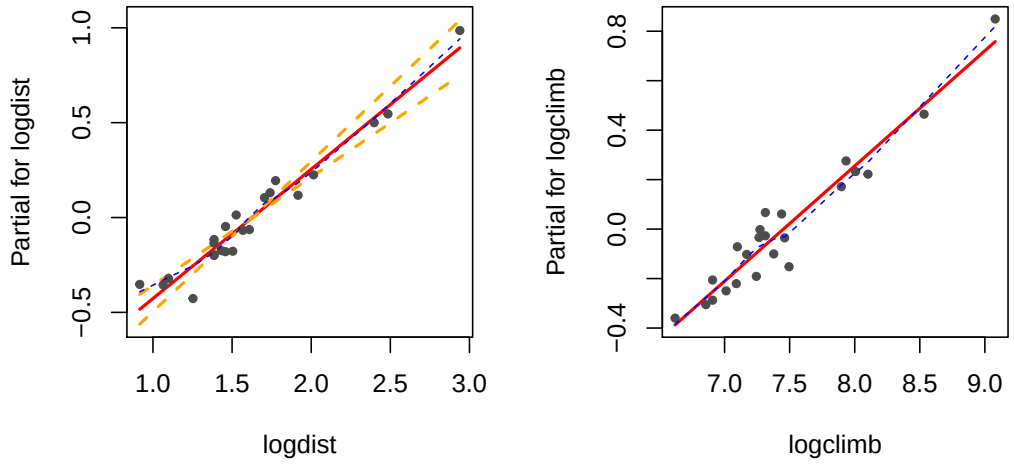
Another transformation that helps the model interpretation, without modifying the goodness of fit, is the centering of the covariates. In the following specification of the `formula` object, the function `I()` can be used to bracket those portions of a model formula where the operators are used in their arithmetic sense.

```
nihillsG.lm_demean <- lm(logtime ~ I(logdist-mean(logdist)) +
        I(logGrad-mean(logGrad)), data=lognihills)
summary(nihillsG.lm_demean)


Call:
lm(formula = logtime ~ I(logdist - mean(logdist)) + I(logGrad -
    mean(logGrad)), data = lognihills)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17223 -0.04229 -0.02538  0.05222  0.13150

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                -0.38199    0.01597  -23.92 3.46e-16 ***
I(logdist - mean(logdist))  1.14712    0.03460   33.16  < 2e-16 ***
I(logGrad - mean(logGrad))  0.46576    0.04530   10.28 1.98e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0766 on 20 degrees of freedom
Multiple R-squared:  0.9831,Adjusted R-squared:  0.9814
F-statistic: 582.7 on 2 and 20 DF,  p-value: < 2.2e-16
```

The only difference is that, in this formulation, the estimate of the intercept corresponds to the sample mean of the response variable `logtime`. Furthermore, using the single regression terms, we may represent the partial residual plot for the corresponding covariate, given all the others. This accounts for the part of the response that is not explained by the other covariates and it assesses whether this part can be approximated by a linear function of the interest covariate.

The function `termplot` draws regression terms against their predictors, optionally with standard errors and partial residuals added. Its first argument is a fitted model object (in this case an `lm` object) and the argument `term` indicates the regression term to be considered. Then, with the option `partial.resid=TRUE` the partial residuals are plotted, `lwd.term` and `col.res` are used to specify the color of the residuals plot and the width of the term regression line, `lwd.se` sets the line width for the twice-standard-error curve (it works if `se=T`), `pch` is used to specify the plotting character used to represent the partial residuals, `col.res` defines their color and `smooth=panel.smooth` draws a smooth curve through the partial residuals. With the option `plot = F`, the plot is not produced and it is returned a list containing the data that would otherwise have been plotted. The following plots show the behavior of the two partial residuals associated to the model `nihills.lm`, which follows an almost linear pattern. For the first plot, we set the option `se=T`.

```
nihills.lm <- lm(logtime ~ logdist + logclimb, data=lognihills)
par(mfrow=c(1,2))
termplot(nihills.lm, terms=1,partial.resid=TRUE, lwd.term=2,lwd.se=2,pch=20,
         smooth=panel.smooth, col.smth='blue', col.res="gray30", se=T)
termplot(nihills.lm, terms=2,partial.resid=TRUE, lwd.term=2,lwd.se=2,pch=20,
         smooth=panel.smooth, col.smth='blue', col.res="gray30")
```

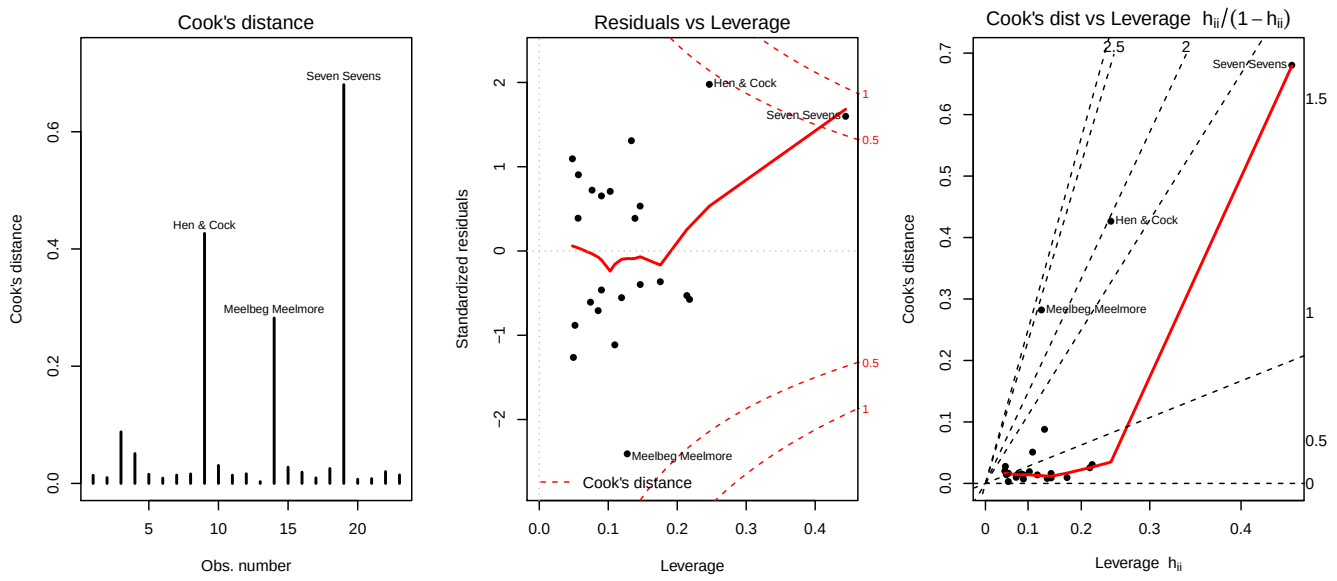



```
par(mfrow=c(1,1))
```

When there are several explanatory variables, the outliers are treated as in the simple linear regression models but their detection could be more complicated. The effect of each single observation on the estimation results can be evaluated graphically by considering the following diagnostic plots, obtained by means of the function `plot`:

- the plot of the Cook's distance associated to the observations (option `which=4`);
- the plot of the standardized residuals against leverage values, with contours of equal Cook's distance (option `which=5`);
- the plot of the Cook's distance against leverage/(1-leverage), with contours of standardized residuals that are equal in magnitude (option `which=6`).

```
par(mfrow=c(1,3))
plot(nihills.lm, which=4, lwd=2, pch = 16, cex.caption=0.8)
plot(nihills.lm, which=5, lwd=2, pch = 16, cex.caption=0.8)
plot(nihills.lm, which=6, lwd=2, pch = 16, cex.caption=0.8)
```




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

The diagnostic plots of the fitted model `nihills.lm` do not indicate serious problems, apart a point with a Cook's distance larger than 0.5. Moreover, one can consider the `dfbetas` and the `influence.measures` functions (with argument an `lm` object) to identify the potential effect that the outliers or leverage points have on the model estimation. In particular, function `dfbetas` measures the standardized variation in the coefficient estimation when a single observation is, in turn, omitted. Evaluation of the (standardized) effect of each observation on the estimates shows that none of the three observations with the largest Cook's distance has a relevant effect.

```
dfbetas(nihills.lm)
```

```
                      (Intercept)      logdist    logclimb
Binevenagh            -0.10661714 -0.159599041  0.12320626
Slieve Gullion        -0.11109739 -0.040353991  0.09473110
Glenariff Mountain     0.40732276  0.386941221 -0.41918081
Donard & Commedagh     0.27403082  0.107614807 -0.26263917
McVeigh Classic       -0.15597632 -0.112976114  0.14936493
Tollymore Mountain    -0.13897791 -0.098176548  0.13469155
Slieve Martin          0.01948203  0.078122276 -0.04545794
Moughanmore           -0.08412298 -0.178926201  0.12139575
Hen & Cock            -0.57433374 -1.108374377  0.78503557
Annalong Horseshoe     0.14085271 -0.076325513 -0.10027373
Monument Race          0.14641842  0.052635600 -0.12611836
Loughshannagh Horseshoe -0.05253904 -0.106370952  0.08004940
Rocky                  0.02560530 -0.016878526 -0.01241825
Meelbeg Meelmore       0.54747634  0.862521912 -0.70286110
```

```
Donard Forest           -0.10042606 -0.010001568  0.07114291
Slieve Donard           -0.16991734 -0.122330083  0.17764042
Flagstaff to Carling    -0.01729453 -0.117852253  0.04148600
Slieve Bearnagh          0.20796330  0.221842871 -0.23347317
Seven Sevens            -0.73612113  0.435358828  0.50350543
Lurig Challenge         -0.10315686 -0.037083609  0.08885476
Scrabo Hill Race         0.09654115 -0.008559602 -0.07201400
Slieve Gallion           0.07896209  0.011762410 -0.05606389
BARF Turkey Trot         0.12754748  0.124683695 -0.12930587
```

Indeed, function `influence.measures` produces a more detailed output on the influence of each observation on the fitted model. In fact, it computes the leave-one-out deletion diagnostics for linear (and generalized linear) models and it gives, in addition to the results of function `dfbetas`, the standardized variation in the regression function at a given observation when this observation is omitted, the covariance ratios (which measures the effect of an observation on the covariance matrix of the regression coefficient estimates), the Cook's distances and the diagonal elements of the hat matrix (which correspond to the leverage values).

Finally, as an alternative to the model `nihills.lm`, we may consider a further model, based again on the log data, with an additional quadratic term for `log(dist)` included.

```
nihills.lm <- lm(logtime ~ logdist + logclimb, data = lognihills)
nihills2.lm <- lm(logtime ~ logdist + logclimb + I(logdist^2), data = lognihills)

summary(nihills.lm)


Call:
lm(formula = logtime ~ logdist + logclimb, data = lognihills)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17223 -0.04229 -0.02538  0.05222  0.13150

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.96113    0.27387  -18.11 7.09e-14 ***
logdist      0.68136    0.05518   12.35 8.19e-11 ***
logclimb     0.46576    0.04530   10.28 1.98e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0766 on 20 degrees of freedom
```

```
Multiple R-squared:  0.9831,Adjusted R-squared:  0.9814
F-statistic: 582.7 on 2 and 20 DF,  p-value: < 2.2e-16
```

```
summary(nihills2.lm)
```

```

Call:
lm(formula = logtime ~ logdist + logclimb + I(logdist^2), data = lognihills)

Residuals:
     Min       1Q   Median       3Q      Max
-0.17021 -0.03935 -0.02481  0.05409  0.10315

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.38002    0.41110 -10.654 1.88e-09 ***
logdist       0.31358    0.20856   1.504   0.1491
logclimb      0.42706    0.04786   8.924 3.19e-08 ***
I(logdist^2)  0.10681    0.05864   1.821   0.0843 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07251 on 19 degrees of freedom
Multiple R-squared:  0.9856,Adjusted R-squared:  0.9834
F-statistic: 434.6 on 3 and 19 DF,  p-value: < 2.2e-16
```

The values for the multiple R^2 and the adjusted multiple R^2 are slightly better in the second model, while the p-values for the F-tests are both very low. Indeed, both the AIC and the BIC values are smaller for the second model. Thus, including the quadratic term seems a good idea, even if the p-value for the quadratic coefficient is not sufficiently low.

```
AIC(nihills.lm,nihills2.lm)
```

```
            df       AIC
nihills.lm   4 -48.12639
nihills2.lm  5 -49.82797
```

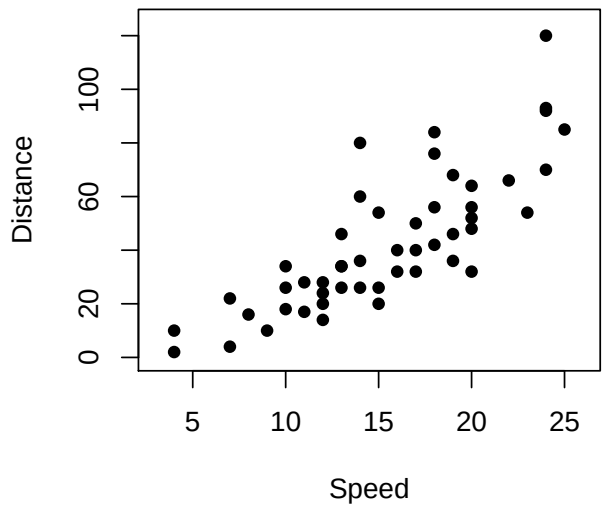
```
BIC(nihills.lm,nihills2.lm)
```

```
            df       BIC
nihills.lm   4 -43.58441
nihills2.lm  5 -44.15050
```

1.3 Example: cars

The `cars` data set, available in the R system libraries, contains data concerning 50 cars, with regard to the `speed` (mph) and the distances (`dist`) taken to stop (ft). A preliminary statistical analysis, supported by physical considerations, suggests that the distance taken to stop should be a non-linear function of the speed.

```
plot(dist ~ speed, data = cars, xlim=c(3,1.04*max(speed)),
     ylim=c(0,1.04*max(dist)), xlab = 'Speed', ylab = 'Distance', pch = 16)
```



A plausible model for this phenomenon could be a polynomial regression where the quadratic effect of the explanatory variable `speed` is taken into account. Thus, a multiple regression model with the covariates `speed` and $\texttt{speed}^2$ is defined.

```
cars2.lm <- lm(dist ~ speed + I(speed^2),data=cars)
```

In order to draw the fitted regression function for this polynomial regression model, the fitted values have to be computed by considering a new data frame, including further specific observations for the covariate `speed`. Then, using the `predict` function the corresponding fitted/predicted values are obtained.

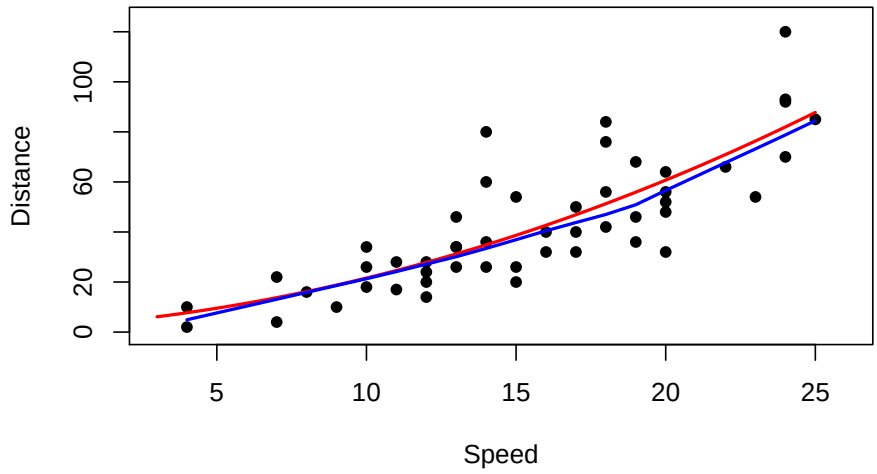
```
dat <- data.frame(speed = seq(3,25,length=100))
fv <- predict(cars2.lm,newdata=dat,se=TRUE)
```

Notice that the same results can be obtained by means of the following command which gives the values of the estimated regression function associated to the values `seq(3,25,length=100)` for the variable `speed`.

```
fitvalues <- cars2.lm$coef[1] + cars2.lm$coef[2]*seq(3,25,length=100) +
       cars2.lm$coef[3]*seq(3,25,length=100)^2
```

The scatterplot of the data, with the fitted regression function in red and the fitted smooth curve in blue (which is the locally-weighted polynomial regression function obtained using function `lowess`), confirms the validity of the polynomial regression model.

```
plot(dist ~ speed, data = cars, xlim=c(3,1.04*max(speed)),
     ylim=c(0,1.04*max(dist)), xlab = 'Speed', ylab = 'Distance', pch = 16)
lines(dat$speed,fv$fit,lwd=2, col='red')
with(cars, lines(lowess(dist ~ speed, f=.7), lwd=2, col='blue'))
```



The inferential results, given by the function `summary`, could be not easy to interpret, as they seem, apparently, inconsistent.

```
summary(cars2.lm)


Call:
lm(formula = dist ~ speed + I(speed^2), data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-28.720  -9.184  -3.188   4.628  45.152

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47014   14.81716   0.167    0.868
```

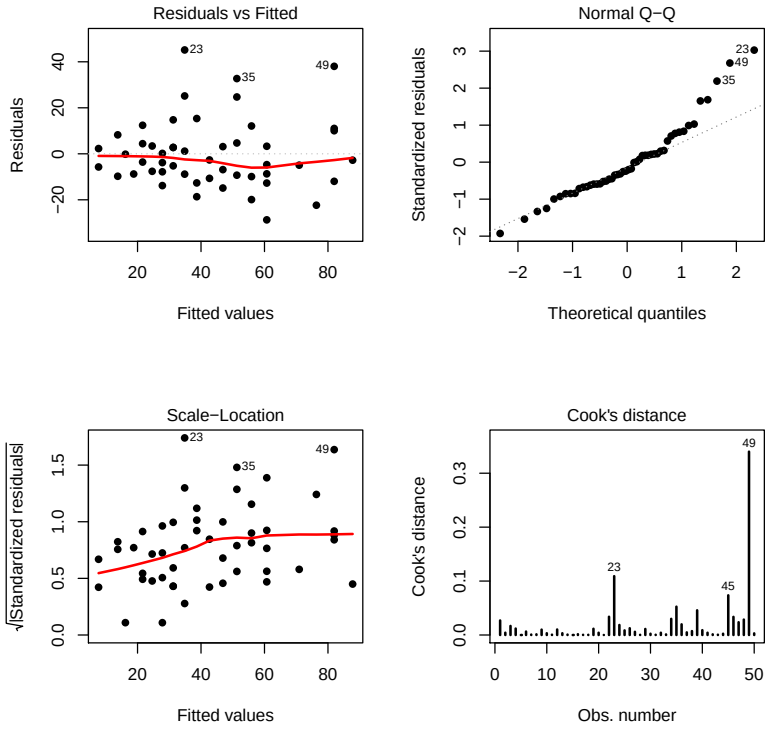
```
speed        0.91329    2.03422   0.449    0.656
I(speed^2)   0.09996    0.06597   1.515    0.136

Residual standard error: 15.18 on 47 degrees of freedom
Multiple R-squared:  0.6673,Adjusted R-squared:  0.6532
F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

Since the p-value for the F-test is very low, there is a strong evidence that the assumed model is better than the constant one, but, unexpectedly, the p-values for all the model coefficients are very high. However, in this case the p-values cannot be taken as an indication that all the terms can be dropped, since there is an evident lack of independence between the coefficient estimators, which creates difficulties in the interpretation of the results. The correlation between parameter estimators arises, as in this example, when there is a correlation between the corresponding covariates. In this case, it is not possible to entirely separate out their effects on the response by examining the results of model fitting. Clearly, in the present polynomial regression model, the two covariates are strongly related, so that there is an evident collinearity, as confirmed by the variance inflation factor values obtained using the function `vif` of the library `DAAG`. The values are very high (with respect to a plausible threshold of 5), which means that there is a severe collinearity and then the model coefficients turns out to be poorly estimated.

```
library(DAAG)
vif(cars2.lm)

     speed I(speed^2)
  24.61489   24.61489
```

The diagnostic plots show some indication of non-constant variance (top left and bottom left) and of a departure from normality in the residuals (top right). In particular, the variability seems to increase with the increasing of `speed`.

```
par(mfrow=c(2,2))
cars2.lm <- lm(dist ~ speed + I(speed^2),data=cars)
plot(cars2.lm, which = 1, lwd=2, pch = 16, cex.caption=0.8)
plot(cars2.lm, which = 2, xlab="Theoretical quantiles",lwd=2, pch = 16,
     cex.caption=0.8)
plot(cars2.lm, which = 3,lwd=2, pch = 16, cex.caption=0.8)
plot(cars2.lm, which = 4,lwd=2, pch = 16, cex.caption=0.8)
```




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

In order to account for heteroschedasticity, we may consider the weighted least squares method of estimation, which can be applied by considering the argument `weights` of the function `lm`, for specifying a weighting system in the optimization procedure. In this case, since the variance of the residuals increases with speed, we may set the option `weights=1/speed`, so that the values for the squared residuals are divided by the corresponding value for the speed.

```
cars2w.lm <- lm(dist ~ speed + I(speed^2),data=cars,weights=1/speed)
summary(cars2w.lm)
```

```

Call:
lm(formula = dist ~ speed + I(speed^2), data = cars, weights = 1/speed)

Weighted Residuals:
   Min     1Q Median     3Q    Max
-6.434 -2.345 -0.880  1.775 11.955

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.73975    9.18580  -0.081    0.936
speed        1.39665    1.47145   0.949    0.347
```

```
I(speed^2)   0.08396    0.05374   1.562    0.125

Residual standard error: 3.757 on 47 degrees of freedom
Multiple R-squared:  0.7124,Adjusted R-squared:  0.7002
F-statistic: 58.22 on 2 and 47 DF,  p-value: 1.906e-13
```

We consider three alternative nested regression models for the `cars` data set, with the aim of comparing them using both the ANOVA procedure and the information criteria for model selection. More precisely, the first model includes the linear effect of `speed` and the intercept, the second model considers the quadratic effect of `speed`, without the intercept, and the third one is the full model. The second and the third models are estimated by considering the weighted least squares method, using function `lm` with the option `weights=1/speed`.

```
cars0.lm <- lm(dist ~ speed, data = cars)
cars1w.lm <- lm(dist ~ I(speed^2) -1, data=cars, weights=1/speed)
cars2w.lm <- lm(dist ~ speed + I(speed^2), data=cars, weights=1/speed)
```

The comparison between two or among more than two nested regression models can be obtained by using the `anova` function, with more than one `lm` object as argument. It is conventional to list the models from smallest to largest, but this is not mandatory.

```
anova(cars0.lm,cars2w.lm)

Analysis of Variance Table

Model 1: dist ~ speed
Model 2: dist ~ speed + I(speed^2)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     48 11353.5
2     47   663.4  1     10690 757.34 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(cars1w.lm,cars2w.lm)

Analysis of Variance Table

Model 1: dist ~ I(speed^2) - 1
Model 2: dist ~ speed + I(speed^2)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     49 756.11
2     47 663.42  2    92.693 3.2834 0.04626 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this case, the ANOVA comparison of the first and the third linear models, which are nested, gives a very low p-value for the F-test, indicating strong evidence against the first model. Indeed, the comparison between the second and the third models gives a p-value which indicates only some slight evidence against the second model.

In order to select the significant variables in a regression model, the `drop1` function can also be adopted. This function evaluates the effect, in terms of changes of fit, of dropping one single regression term from the overall model. A similar function `add1` can be used to evaluate the effect of adding one single term to the base model. In addition to the fitted model object, the main arguments of these functions are:

- `scope`, which is a formula giving the terms to be considered for adding or dropping;
- `test`, which is a logical argument stating whether a test statistic, relative to the original model, has to be included in the output;
- `k`, which is the penalty constant to be considered in the information criteria (the default is 2, corresponding to the AIC).

By applying function `drop1` to the fitted model `cars2w.lm`, we conclude that, according to the AIC (note that here the AIC is computed as in `extractAIC`, giving values different from `AIC`), the best model is that one obtained by removing the variable `speed`. Moreover, each F-test expresses if the model without the corresponding variable is significantly different from the overall model. In this context, the results correspond exactly to those obtained with the command `summary(cars2w.lm)`, and related to the t-test for the coefficients. As emphasized before, since the two covariates are highly correlated, these results can not be considered as an indication that all the terms have to be dropped.

```
drop1(cars2w.lm, test = "F")

Single term deletions

Model:
dist ~ speed + I(speed^2)
           Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>                  663.42 135.27
speed       1    12.717 676.14 134.22  0.9009 0.3474
I(speed^2)  1    34.445 697.86 135.80  2.4402 0.1250
```

Finally, the three models specified before can be compared using the AIC and the BIC, with the aim of selecting the best model specification. Using the functions `AIC` and `BIC`, we can compute the values for this two information criteria, with regard to the fitted model objects considered as arguments.

```
AIC(cars0.lm,cars1w.lm,cars2w.lm)

          df      AIC
cars0.lm   3 419.1569
cars1w.lm  2 414.8026
cars2w.lm  4 412.2635

BIC(cars0.lm,cars1w.lm,cars2w.lm)

          df      BIC
cars0.lm   3 424.8929
cars1w.lm  2 418.6266
cars2w.lm  4 419.9116
```

The AIC statistic suggests the larger model, while the BIC statistic points to the second model, penalizing more the larger model. The alternative models can be compared also by considering the full model description given by the function `summary`. Then, we find that the second model has larger values for the multiple R^2 and the adjusted multiple R^2.

```
summary(cars0.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(cars1w.lm)


Call:
lm(formula = dist ~ I(speed^2) - 1, data = cars, weights = 1/speed)
```

```
Weighted Residuals:
    Min      1Q  Median      3Q     Max
-6.8881 -2.1208  0.1093  2.4735 13.1561

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
I(speed^2) 0.157012   0.007935   19.79   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.928 on 49 degrees of freedom
Multiple R-squared:  0.8888,Adjusted R-squared:  0.8865
F-statistic: 391.6 on 1 and 49 DF,  p-value: < 2.2e-16
```

```
summary(cars2w.lm)
```

```

Call:
lm(formula = dist ~ speed + I(speed^2), data = cars, weights = 1/speed)

Weighted Residuals:
   Min     1Q Median     3Q    Max
-6.434 -2.345 -0.880  1.775 11.955

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.73975    9.18580  -0.081    0.936
speed        1.39665    1.47145   0.949    0.347
I(speed^2)   0.08396    0.05374   1.562    0.125

Residual standard error: 3.757 on 47 degrees of freedom
Multiple R-squared:  0.7124,Adjusted R-squared:  0.7002
F-statistic: 58.22 on 2 and 47 DF,  p-value: 1.906e-13
```

2 Covariates: selection and multicollinearity

2.1 Example: coxite

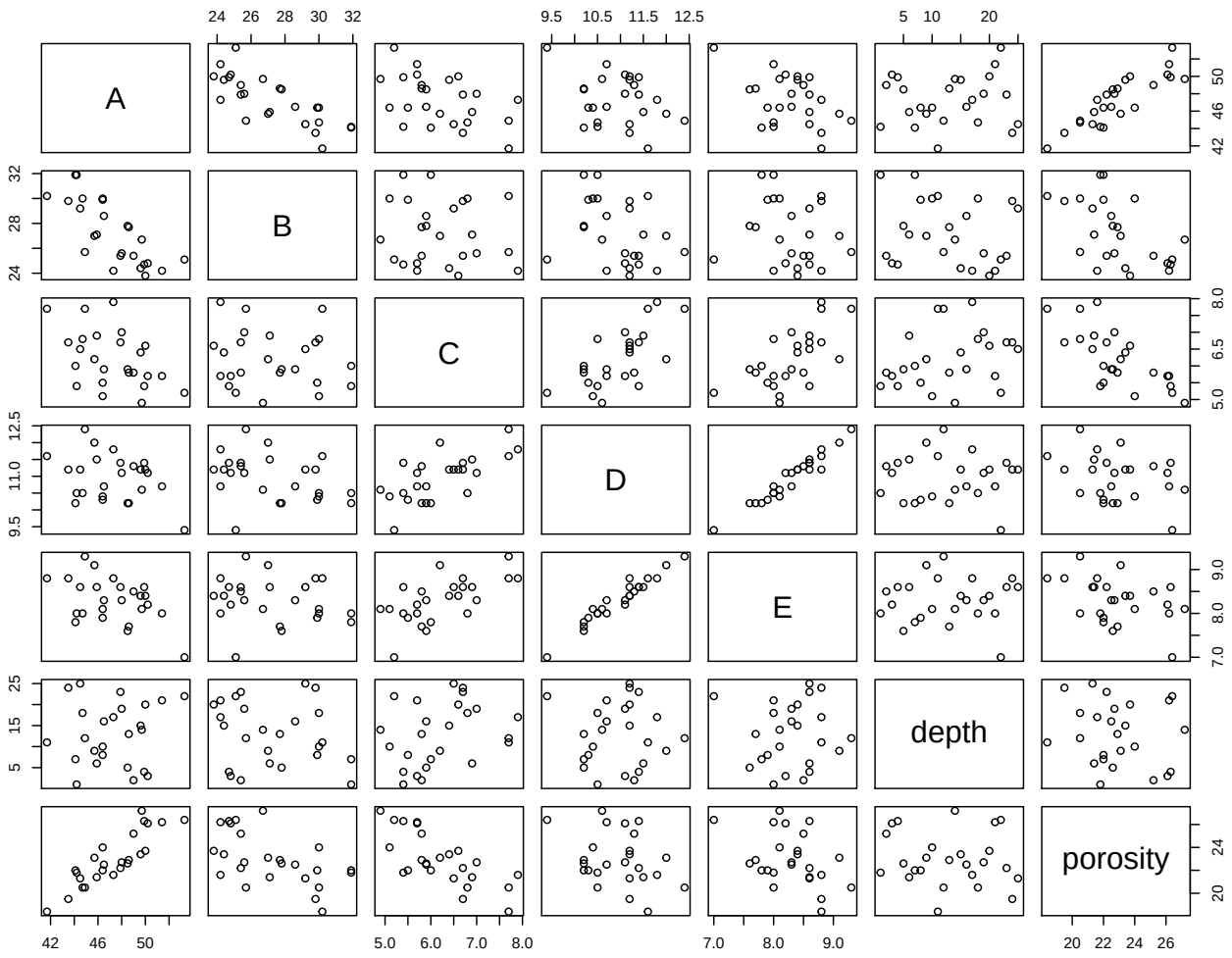
The data set `Coxite` of the library `compositions` contains the mineral compositions of 25 rock specimens of coxite type. Each composition consists of the percentage by weight of five minerals, namely, albite `A`, blandite `B`, cornite `C`, daubite `D`, endite `E` (all row percentage sums to 100). Indeed, further data concerns the recorded `depth` (m) of location of each specimen and the `porosity` (the percentage of void space that the specimen contains). The object `Coxite` is a matrix and it is

redefined as a data frame with the function `as.data.frame`. The aim of the subsequent analysis to explain the response variable `porosity` as a function of mineral composition and `depth`. However, the specification of the regression model requires some care.

```
library(compositions)
data(Coxite)
coxite <- as.data.frame(Coxite)
```

At first the relationship among variables is described graphically using the multiple graphical representation given by the following scatterplot matrix, which shows that `D` and `E` are strongly linearly related.

```
pairs(coxite)
```



Fitting the model with all the six explanatory variables gives a coefficient for `E` equal to `NA`, since the five percentage sum to 100 and then `E` adds no additional information.

```
coxiteAll.lm <- lm(porosity ~ A+B+C+D+E+depth, data=coxite)
summary(coxiteAll.lm)
```

```

Call:
lm(formula = porosity ~ A + B + C + D + E + depth, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93042 -0.46984  0.02421  0.35219  1.18217

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -217.74660  253.44389  -0.859    0.401
A              2.64863    2.48255   1.067    0.299
B              2.19150    2.60148   0.842    0.410
C              0.21132    2.22714   0.095    0.925
D              4.94922    4.67204   1.059    0.303
E                   NA         NA      NA       NA
depth          0.01448    0.03329   0.435    0.668

Residual standard error: 0.6494 on 19 degrees of freedom
Multiple R-squared:  0.9355,Adjusted R-squared:  0.9186
F-statistic: 55.13 on 5 and 19 DF,  p-value: 1.185e-10
```

Then, we consider the model where the explanatory variable `E` (or one of `A`, `B`, `C`, `D`) is omitted.

```
coxite0.lm <- lm(porosity ~ A+B+C+D+depth, data=coxite)
summary(coxite0.lm)
```

```

Call:
lm(formula = porosity ~ A + B + C + D + depth, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93042 -0.46984  0.02421  0.35219  1.18217

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -217.74660  253.44389  -0.859    0.401
A              2.64863    2.48255   1.067    0.299
B              2.19150    2.60148   0.842    0.410
C              0.21132    2.22714   0.095    0.925
```

```
D             4.94922    4.67204   1.059    0.303
depth         0.01448    0.03329   0.435    0.668

Residual standard error: 0.6494 on 19 degrees of freedom
Multiple R-squared:  0.9355,Adjusted R-squared:  0.9186
F-statistic: 55.13 on 5 and 19 DF,  p-value: 1.185e-10
```

The outcome of the fitting procedure reveals that none of the individual coefficients is significantly different from zero, since all the p-values are greater than 0.3. However, both the multiple R^2 measures are very high and the F-test of global effectiveness of all the regression terms is highly significant. These are clear symptoms of multicollinearity and this is the reason why none of the individual coefficients can be estimated meaningfully. The collinearity is confirmed by the variance inflation factor (VIF) values obtained using the function `vif` of the library `DAAG`. The values are very high, emphasizing that multicollinearity is an important issue in this model specification.

```
library(DAAG)
vif(coxite0.lm)

          A           B           C           D       depth
2717.815152 2484.976310  192.588948  566.143622    3.416583
```

In order to specify a suitable regression model, a simple preliminary procedure could be to select those explanatory variables that are, individually, most strongly correlated with the response variable `porosity`.

```
cor(coxite$porosity, coxite[,-7])

             A          B          C          D          E      depth
[1,] 0.8690284 -0.5511044 -0.7233127 -0.3199149 -0.4075911 -0.1467961
```

The model based on the covariates `A`, `B` and `C`, which present highest absolute values for the correlation coefficients, is estimated and the VIF values are computed.

```
coxite1.lm <- lm(porosity ~ A+B+C, data=coxite)
summary(coxite1.lm)


Call:
lm(formula = porosity ~ A + B + C, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.98137 -0.37455  0.02294  0.41742  1.27272

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 53.30463   13.22186   4.032 0.000603 ***
A           -0.01246    0.15580  -0.080 0.937003
B           -0.58668    0.15134  -3.876 0.000873 ***
C           -2.21880    0.33886  -6.548 1.74e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6425 on 21 degrees of freedom
Multiple R-squared:  0.9302,Adjusted R-squared:  0.9203
F-statistic: 93.35 on 3 and 21 DF,  p-value: 2.639e-12
```

```
vif(coxite1.lm)
```

```
        A         B         C
10.936093  8.592363  4.555127
```

Although this model improves the previous one, the regression coefficient for covariate `A` is not significantly different from zero and yet the VIF values are all larger than 4, indicating the presence of a relevant multicollinearity. Finally, the model with only `B` and `C` as explanatory variable is estimated. In this case, the fitted model passes all the diagnostic checks and the VIF values are both around 1. Furthermore, the corresponding value for the AIC statistic is effectively lower that one obtained for the full model.

```
coxite2.lm <- lm(porosity ~ B+C, data=coxite)
summary(coxite2.lm)
```

```


Call:
lm(formula = porosity ~ B + C, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.98353 -0.37851 -0.00347  0.41783  1.26453

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.25713    1.78312   29.31  < 2e-16 ***
B           -0.57531    0.05078  -11.33 1.19e-10 ***
C           -2.19490    0.15617  -14.05 1.81e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6278 on 22 degrees of freedom
Multiple R-squared:  0.9302,Adjusted R-squared:  0.9239
```

```
F-statistic: 146.6 on 2 and 22 DF,  p-value: 1.909e-13

vif(coxite2.lm)

       B        C
1.013249 1.013249

AIC(coxite0.lm,coxite2.lm)

           df      AIC
coxite0.lm  7 56.49973
coxite2.lm  4 52.47331
```

3 Factors as explanatory variables

3.1 Example: paper resistance

In a regression model, explanatory variables are not always numeric, and actually factors are very common in many applied fields. To include a factor in a model it is necessary to code its levels, and one possibility is to use dummy variables, which are regressors that assume only two values, usually, zero and one. More precisely, coding a factor with h levels (categories) requires the usage of $(h-1)$ dummy variables. This particular codification of a factor is known as *treatment contrasts* and it requires that one of the factor levels has to be set as a reference, with the effects of the other levels measured from that baseline. Alternative contrast specifications are also possible; although they lead to exactly the same model fit, the coding based on treatment contrasts is the easiest to interpret and it is the natural choice for observational data.. The analysis of variance (ANOVA) models are just a special case of linear regression models where all the explanatory variables are factors. In one-way ANOVA, there is only one factor, in multi-way ANOVA there are several factors.

The paper resistance example, already considered in the Lab 4 notes, is used here to show the correspondence between the ANOVA model estimation and the linear model estimation when the explanatory variable is a factor. The data frame `paper`, with the observed values of `resistance` and treatment `trt` (wood fibre concentration), is created. Indeed, the function `factor` is used to order the treatment levels, since without this command the levels would be considered in alphabetic order. Alternatively, the treatment variable can redefined by using the `relevel` function, which allows the specification of the reference level to be 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$trt <- factor(paper$trt,levels=c("5%", "10%", "15%", "20%"))
```

As in the Lab 4 notes, an analysis of variance model is fitted by using the `aov` function, which main arguments are similar to those of the function `lm`. A further important argument is `contrasts`, which can be used to define the list of contrasts to test (the default option corresponds to treatment contrasts).

```
paper.aov <- aov( resistance ~ trt , data=paper)
summary(paper.aov)

            Df Sum Sq Mean Sq F value   Pr(>F)
trt          3  382.8  127.60   19.61 3.59e-06 ***
Residuals   20  130.2    6.51
---
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 that all the mean values, related to the alternative factor levels, are equal.
An alternative analysis can be performed by using function `lm`, in order to define a linear model where the explanatory variable `trt` is now specified as a factor. If `trt` were defined as a numerical vector in the data frame `paper`, it can be considered as a factor by setting `factor(trt)` instead of `trt` in the `lm` syntax. Since, in this case the treatment variable has been redefined by setting a reference level, the model is estimated by considering the level `5%` as the benchmark level and by specifying 3 dummy variables which identify the second, the third and the fourth concentration levels. Then, contrary to the ANOVA procedure, the mean of the response variable is defined as a linear function of the dummy variables; namely, the intercept corresponds to the mean value in the first group (related to the benchmark level `5%`) and the three coefficients specify the additional differential effects of passing from the level `5%` to one of the other three levels of `trt`.

```
paper.lm1 <- lm(resistance ~ trt,data=paper)
summary(paper.lm1)


Call:
lm(formula = resistance ~ trt, data = paper)

Residuals:
   Min     1Q Median     3Q    Max
-3.667 -2.042  0.000  1.458  5.000
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   10.000      1.041   9.602 6.24e-09 ***
trt10%         5.667      1.473   3.847 0.001005 **
trt15%         7.000      1.473   4.753 0.000122 ***
trt20%        11.167      1.473   7.581 2.65e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.551 on 20 degrees of freedom
Multiple R-squared:  0.7462,Adjusted R-squared:  0.7082
F-statistic: 19.61 on 3 and 20 DF,  p-value: 3.593e-06
```

The results of the model fitting procedure is specified using the command `summary(paper.lm1)`, which gives the estimated coefficients and the associated standard errors. Furthermore, the individual t-tests strongly support the conclusion that all the model parameters are significantly different from zero and the global F-test gives, in this particular case, a result which corresponds to that of the ANOVA procedure.

The same result can be obtained using function `summary.lm`. The output of the `aov` command is interpreted as a linear model object so that the ANOVA results are presented by considering the corresponding linear model, where there are three regressors specifying the differential effect of the factor levels `10%`, `15%` and `20%`, with respect to the baseline `5%`.

```
summary.lm(paper.aov)


Call:
aov(formula = resistance ~ trt, data = paper)

Residuals:
   Min     1Q Median     3Q    Max
-3.667 -2.042  0.000  1.458  5.000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   10.000      1.041   9.602 6.24e-09 ***
trt10%         5.667      1.473   3.847 0.001005 **
trt15%         7.000      1.473   4.753 0.000122 ***
trt20%        11.167      1.473   7.581 2.65e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.551 on 20 degrees of freedom
Multiple R-squared:  0.7462,Adjusted R-squared:  0.7082
F-statistic: 19.61 on 3 and 20 DF,  p-value: 3.593e-06
```

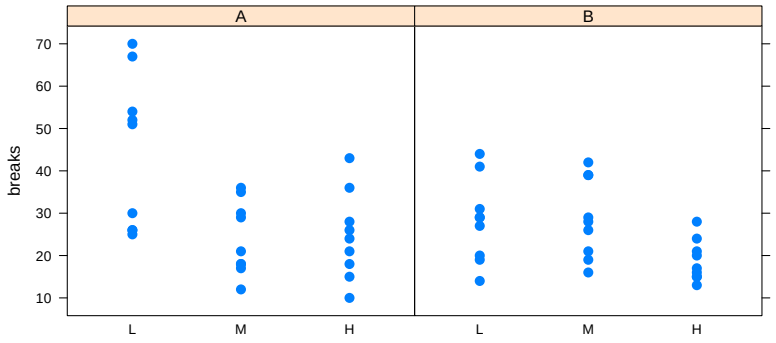
Finally, we may state that ANOVA models are just linear models when the explanatory variables are all categorical. It is not essential to recognize this fact, but in this framework there are some advantages, since all the techniques developed for linear models (for example, model selection techniques, model diagnostics and prediction methods) may be used also for ANOVA models. Moreover, in this context, numerical explanatory variables can be also introduced in an ANOVA setting.

3.2 Example: warp breaks

In a linear regression model with a factor regressor, the regression coefficients are used to represent the effect of a given factor, which corresponds to the *main effect*. In case of two or more than two factors, beside their main effect it is possible to introduce in the model formulation their *interaction* effect. The interaction is expressed by the product of the variables used to represent these factors.

We consider the data frame `warpbreaks`, available in the base R distribution. The data set gives, as response variable, the number of warp breaks (`breaks`) per a fixed length of yarn during weaving and two experimental factors: the type of `wool`, with two levels `A` and `B`, and the level of `tension`, with three levels `L` (low), `M` (medium) and `H` (high). There are 9 replications for each of the 6 combinations of the factors levels (the data are balanced) and the total sample size is then $2 \times 3 \times 9 = 54$. The `stripplot` function of the library `lattice` produces two conditional scatterplots where the response variable is plotted conditional on the factor `tension`, for the two levels of the factor `wool`.

```
library(lattice)
data(warpbreaks)
stripplot(breaks ~ tension | wool, warpbreaks, cex=1.2, pch=16)
```



A two-way analysis of variance model is fitted by using the `aov` function. Here, the transformed response observations `sqrt(breaks)` are taken into account instead of the original ones, as the conditional plot shows some evidence of non-constant variance (ANOVA models assume a constant

variance for the observations). Indeed, in order to consider both the main effects of the two factors and their interaction, the `formula` specification is `sqrt(breaks) ~tension*wool`. Notice that, the `*` operator denotes factor crossing, so that : `tension*wool=tension+wool+tension:wool`, where the operator : is interpreted as the interaction of all the variables and factors appearing in the term. With the argument `sqrt(breaks) ~tension+wool`, only the two main effects are specified.

```
breaks.aov<-aov(sqrt(breaks) ~ tension*wool, warpbreaks)
anova(breaks.aov)

Analysis of Variance Table

Response: sqrt(breaks)
             Df Sum Sq Mean Sq F value   Pr(>F)
tension       2 15.892  7.9458  8.2752 0.000817 ***
wool          1  2.902  2.9019  3.0222 0.088542 .
tension:wool  2  7.201  3.6007  3.7500 0.030674 *
Residuals    48 46.089  0.9602
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The same result can be obtained by comparing, with the `anova` function, a suitable sequence of ordered nested ANOVA models.

```
breaks0.aov<-aov(sqrt(breaks) ~ 1, warpbreaks)
breaks1.aov<-aov(sqrt(breaks) ~ tension, warpbreaks)
breaks2.aov<-aov(sqrt(breaks) ~ tension+wool, warpbreaks)
anova(breaks0.aov,breaks1.aov,breaks2.aov,breaks.aov)

Analysis of Variance Table

Model 1: sqrt(breaks) ~ 1
Model 2: sqrt(breaks) ~ tension
Model 3: sqrt(breaks) ~ tension + wool
Model 4: sqrt(breaks) ~ tension * wool
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     53 72.084
2     51 56.193  2   15.8916 8.2752 0.000817 ***
3     50 53.291  1    2.9019 3.0222 0.088542 .
4     48 46.089  2    7.2014 3.7500 0.030674 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the F-test on the interaction effect is 0.031, showing an appreciable interaction effect, though not very large. Moreover, we may immediately state that the two main effects are present as-well, since both the two factors influence the mean of the response variable. Thus, in

case the data support the presence of an interaction effect, it makes little sense to investigate about the presence of main effects. On the other hand, in case the interaction effect is not significant, the test on the two main effects has to be performed.

As in the previous example, the `summary.lm` function can be used to obtain the results in the linear regression framework. The model with only the main effects would have $2+1=3$ dummy variables (2 for the levels of factor `tension` and 1 for the levels of factor `wool`) and 4 regression coefficients (including that one related to the baseline specified by `wool=A` and `tension=L`). Instead, the model considered below, which includes the interaction effect, requires $2 \times 1 = 2$ additional parameters, corresponding to the products between the two dummy variables related to `tension` and that one associated to `wool`.

```
summary.lm(breaks.aov)


Call:
aov(formula = sqrt(breaks) ~ tension * wool, data = warpbreaks)

Residuals:
     Min       1Q   Median       3Q      Max
-1.69410 -0.70129  0.01772  0.66046  1.81902

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.5476     0.3266  20.046  < 2e-16 ***
tensionM       -1.7216     0.4619  -3.727 0.000511 ***
tensionH       -1.6912     0.4619  -3.661 0.000625 ***
woolB          -1.3094     0.4619  -2.835 0.006692 **
tensionM:woolB  1.7821     0.6533   2.728 0.008874 **
tensionH:woolB  0.7553     0.6533   1.156 0.253350
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9799 on 48 degrees of freedom
Multiple R-squared:  0.3606,Adjusted R-squared:  0.294
F-statistic: 5.415 on 5 and 48 DF,  p-value: 0.0004998
```

The estimated coefficients, with the associated standard errors, and the results for the individual t-tests are provided. The global significance of all the regression coefficients is confirmed by the F-test, which gives a low p-value. In this case, the F-test does not correspond to those ones considered in the ANOVA procedure, focusing on the main and the interaction effects. The same result, concerning the F-test, can be obtained using the `anova` function for comparing the full model with that one specified by the intercept term only.

```
anova(breaks0.aov,breaks.aov)

Analysis of Variance Table

Model 1: sqrt(breaks) ~ 1
Model 2: sqrt(breaks) ~ tension * wool
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     53 72.084
2     48 46.089  5    25.995 5.4145 0.0004998 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.3 Example: leaf and air temperatures

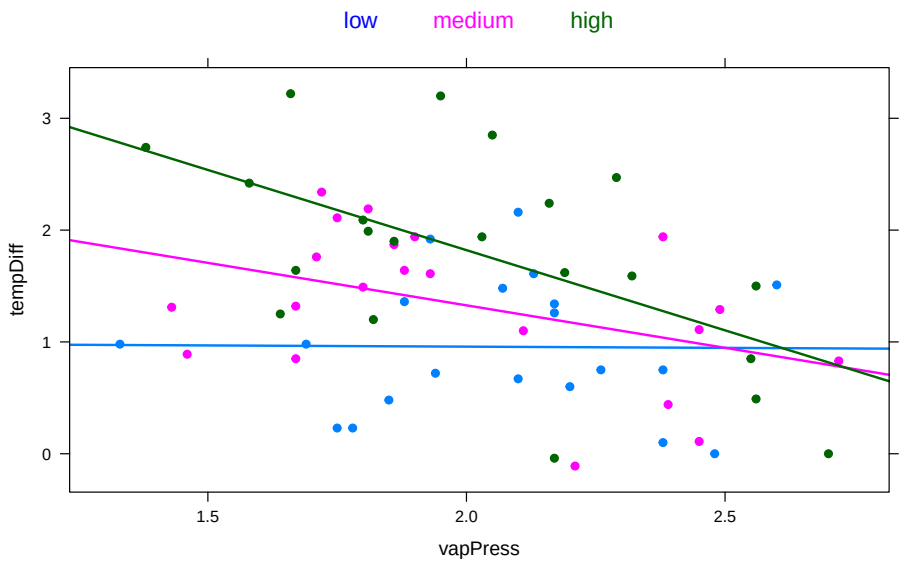
In many applications, we have to deal with both categorical and numerical explanatory variables. By using a suitable factor coding for the categorical variables, it is possible to include the two different types of regressors in the model specification. These models are known as analysis of covariance models, and they correspond essentially to fit multiple regression lines.

In the present example, we analyze the data frame `leaftemp` of the library `DAAG`. These data consist of 62 measurements of vapor pressure (`vapPress`) and of the difference between leaf and air temperature (`tempDiff`), for three different levels (`low`, `medium`, `high`) of carbon dioxide (`CO2level`). There is also an additional variable `BtempDiff`, which is not used in the application.

Using the function `xyplot` of the library `lattice`, it is possible to represent the scatterplot of the variables `vapPress` and `tempDiff`, where different colors indicate the different levels of the factor `CO2level`. Indeed, for each group of bivariate observations, the regression line is reported with the corresponding color. The first and the second arguments of function `xyplot` indicate the variables and the data set taken into account, while the option `groups=CO2level` states that the analysis has to be performed conditionally to the levels of factor `CO2level`. The complete list of the arguments can be obtained from the help page. In this application, we use the option `key=simpleKey()` in order to to define a title written on three columns (`columns=3`), with three different colors (`col=c('blue','magenta','darkgreen')`) and arranged on the top of the plot (`space="top"`). Moreover, with the option `type=c("p","r")`, we state that the plot has to include both the points and the regression lines.

```
library(DAAG)
library(lattice)
xyplot(tempDiff ~ vapPress, leaftemp, groups=CO2level, pch=19,
       key=simpleKey(text=c('low','medium','high'),space="top",
       columns=3,points=FALSE,col=c('blue','magenta','darkgreen'),cex=1.2),
```

```
       type=c("p","r"),lwd=2)
```



A more formal analysis is based on the comparison of the following four alternative models:

- the null model `leaf.lm1`, which specifies a constant mean response;
- the model `leaf.lm2` based on the numerical explanatory variable `vapPress` (a single regression line);
- the model `leaf.lm3` based on `vapPress` and on the factor `CO2level` (three parallel regression lines);
- the model `leaf.lm4` based on `vapPress`, `CO2level` and their interaction (three separate regression lines).

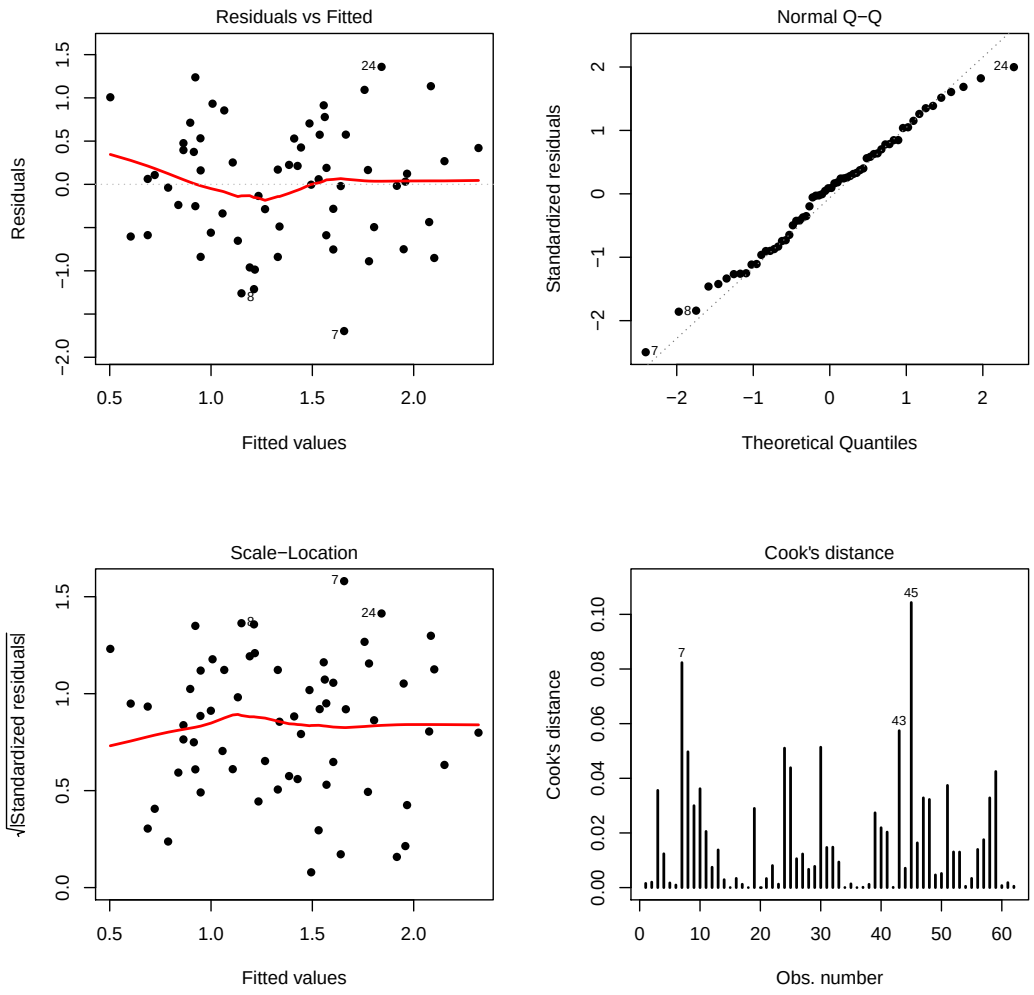
```
leaf.lm1 <- lm(tempDiff ~ 1 , data = leaftemp)
leaf.lm2 <- lm(tempDiff ~ vapPress, data = leaftemp)
leaf.lm3 <- lm(tempDiff ~ CO2level + vapPress, data = leaftemp)
leaf.lm4 <- lm(tempDiff ~ CO2level + vapPress + vapPress:CO2level, data = leaftemp)
anova(leaf.lm1, leaf.lm2, leaf.lm3, leaf.lm4)

Analysis of Variance Table

Model 1: tempDiff ~ 1
Model 2: tempDiff ~ vapPress
Model 3: tempDiff ~ CO2level + vapPress
Model 4: tempDiff ~ CO2level + vapPress + vapPress:CO2level
  Res.Df    RSS Df Sum of Sq       F   Pr(>F)
1     61 39.999
2     60 34.727  1    5.2720 11.3305 0.001383 **
```

```
3     58 28.183  2    6.5441  7.0322 0.001885 **
4     56 26.056  2    2.1262  2.2848 0.111205
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The comparison of the four nested linear regression models is obtained using function `anova`, which suggests the use of the model with the parallel regression lines, since the reduction in the mean square from `leaf.lm3` to `leaf.lm4` has a p-value equal to 0.1112. The inferential summary on the selected model is given below. Indeed, also the diagnostic check does not show any particular problem.

```
summary(leaf.lm3)


Call:
lm(formula = tempDiff ~ CO2level + vapPress, data = leaftemp)

Residuals:
     Min       1Q   Median       3Q      Max
-1.69683 -0.54299  0.06076  0.46371  1.35854

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      2.6849     0.5596   4.798 1.16e-05 ***
CO2levelmedium   0.3199     0.2185   1.464 0.148615
CO2levelhigh     0.7931     0.2179   3.640 0.000582 ***
vapPress        -0.8392     0.2610  -3.216 0.002129 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6971 on 58 degrees of freedom
Multiple R-squared:  0.2954,Adjusted R-squared:  0.259
F-statistic: 8.106 on 3 and 58 DF,  p-value: 0.0001352
```

```
par(mfrow=c(2,2))
plot(leaf.lm3, which = 1, lwd=2, pch = 16, cex.caption=0.8)
plot(leaf.lm3, which = 2, lwd=2, pch = 16, cex.caption=0.8)
plot(leaf.lm3, which = 3, lwd=2, pch = 16, cex.caption=0.8)
plot(leaf.lm3, which = 4, lwd=2, pch = 16, cex.caption=0.8)
```

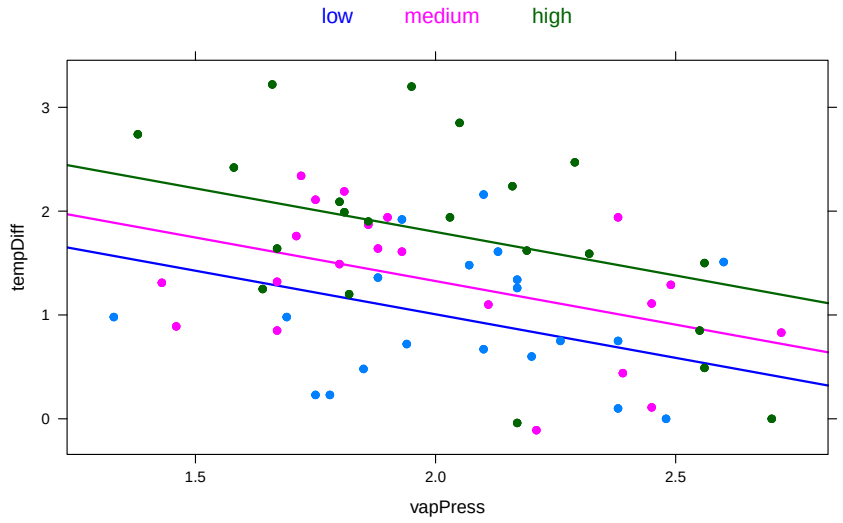



```
par(mfrow=c(1,1))
```

Furthermore, this model specification can be clearly represented by exploiting a specific opportunity of the `lattice` library. Three objects are obtained by applying function `xyplot`. These objects contains the same scatterplot, where the points have three different colors according to the factor levels, and, respectively, a different regression line related to each of the three subgroups of data. The lines, added to the plot using the function `abline`, differ only in the intercept values, obtained as the sum of the associated parameter estimates. The three plots are then jointly plotted with the command `plot1+plot2+plot3`.

```
plot1 <- xyplot(tempDiff ~ vapPress, leaftemp, groups=CO2level,pch=19,
abline = list(a=2.6849,b=-0.8392,col='blue',lwd=2),
key=simpleKey(text=c('low','medium','high'),space="top", columns=3,
points=FALSE,col=c('blue','magenta','darkgreen'),cex=1.2), lwd=2)

plot2 <- xyplot(tempDiff ~ vapPress, leaftemp, groups=CO2level,pch=19,
```

```
abline = list(a=2.6849+0.3199,b=-0.8392,col='magenta',lwd=2),
key=simpleKey(text=c('low','medium','high'),space="top", columns=3,
points=FALSE,col=c('blue','magenta','darkgreen'),cex=1.2), lwd=2)

plot3 <- xyplot(tempDiff ~ vapPress, leaftemp, groups=CO2level,pch=19,
abline = list(a=2.6849+0.7931,b=-0.8392,col='darkgreen',lwd=2),
key=simpleKey(text=c('low','medium','high'),space="top", columns=3,
points=FALSE,col=c('blue','magenta','darkgreen'),cex=1.2), lwd=2)

plot1+plot2+plot3
```



4 Regression models with discrete responses

4.1 Example: teaching program

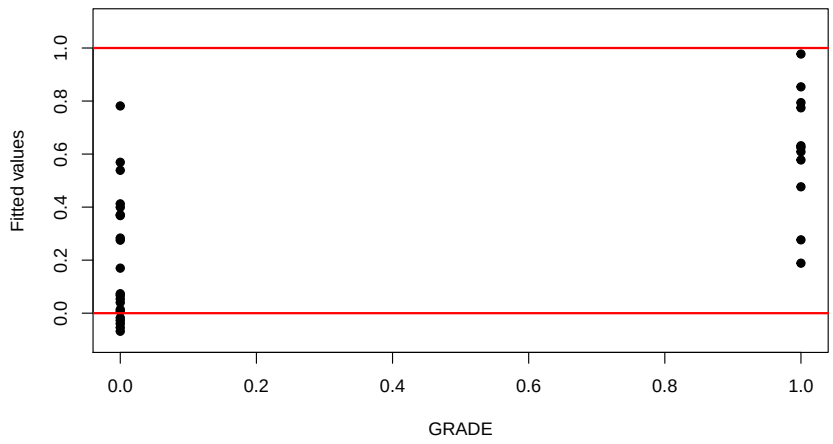
We consider a data set on the effectiveness of a teaching program; for 32 students, the following variables are observed: the grade point average for the period (`GPA`), the score on economics test (`TUCE`), the participation in the teaching program (`PSI`, with values `1`, yes, and `0`, no) and the grade increase indicator (`GRADE`, with values `1`, increase, and `0`, decrease). With the following commands, a 32×5 matrix containing the observed data is defined, the columns names are specified and the matrix is transformed in the data frame `program`.

```
program <- matrix(c(1,2.66,20,0,0,2,2.89,22,0,0,3,3.28,24,0,0,4,2.92,12,0,
                    0,5,4.00,21,0,1,6,2.86,17,0,0,7,2.76,17,0,0,8,2.87,21,
                    0,0,9,3.03,25,0,0,10,3.92,29,0,1,11,2.63,20,0,0,12,3.32,
                    23,0,0,13,3.57,23,0,0,14,3.26,25,0,1,15,3.53,26,0,0,16,
                    2.74,19,0,0,17,2.75,25,0,0,18,2.83,19,0,0,19,3.12,23,1,
                    0,20,3.16,25,1,1,21,2.06,22,1,0,22,3.62,28,1,1,23,2.89,
```

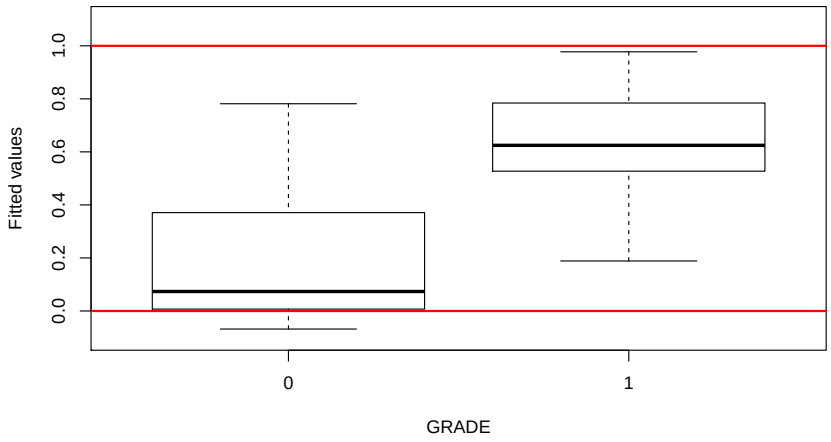
```
                    14,1,0,24,3.51,26,1,0,25,3.54,24,1,1,26,2.83,27,1,1,27,
                    3.39,17,1,1,28,2.67,24,1,0,29,3.65,21,1,1,30,4.00,23,1,
                    1,31,3.10,21,1,0,32,2.39,19,1,1), nrow=32, byrow=T)
colnames(program) <- c("OBS","GPA","TUCE","PSI","GRADE")
program <- as.data.frame(program)
```

In order to study the relationship between the grade increase indicator and the other three explanatory variables, a multiple linear regression model is defined. The results of the fitting procedure are represented by computing the scatterplot of the fitted values and the observed values of the response variable `GRADE`, and also the corresponding boxplots. It is immediate to conclude that the model is badly specified since some fitted values are negative. This is clearly unacceptable, as the mean value of the response variable is the probability of `GRADE=1`.

```
mod0.lm <- lm(GRADE ~ GPA + TUCE + PSI , data = program)
plot(program$GRADE,fitted(mod0.lm),pch=19,ylim=c(-0.1,1.1),xlab="GRADE",
     ylab="Fitted values")
abline(0,0,col='red',lwd=2)
abline(1,0,col='red',lwd=2)
```




```
boxplot(fitted(mod0.lm)~program$GRADE,xlab="GRADE",ylim=c(-0.1,1.1),
        ylab="Fitted values")
abline(0,0,col='red',lwd=2)
abline(1,0,col='red',lwd=2)
```



The logistic multiple regression (logit) model can be introduced in order to extend the linear model specification to dichotomous response variables. In this context, the log odds is modeled as a linear function of the explanatory variables, since the logistic (logit) link function is considered. Other choices for the link function may be adopted. Logistic regression models belong to the wide class of generalized linear models, which extend linear regression models so that a more general form of expression for the mean response is allowed (using suitable link functions) and various types of distributions for the response can be considered.

The `glm` function can be used to estimate different kinds of generalized linear models, specified by giving a symbolic description of the linear predictor and the specification for the error distribution. Many of the available arguments correspond to those of the function `lm`, while some further important arguments are:

- `family`, which describes the error distribution and the link function to be used in the model; the default specification is `gaussian(link = "identity")` but different model distributions can be defined (such as `binomial`, `poisson`, `Gamma`, `inverse.gaussian` and so on), with alternative link functions;
- `start`, `etastart` and `mustart`, which can be used to define the starting values, respectively, for the regression coefficients, the linear predictor and the vector of means to be considered in the iterative estimation procedure;
- `offset`, which can be used to specify an *a priori* known component to be included in the linear predictor during the fitting procedure;
- `method`, which defines the method to be used in fitting the model.

Function `glm` returns an object of class `"glm"` and, as for `"lm"` objects, the function `summary` (namely, `summary.glm`) can be used to obtain a summary of the fitting results and the function `anova` (namely, `anova.glm`) can be considered to produce an analysis of variance table.

With regard to the data set on the effectiveness of a teaching program, we first consider a logistic regression model for the the binary response `GRADE` (grade increase) and the factor predictor `PSI` (participation in program). The argument `family = binomial` is specified, with the default link function `link = "logit"`. The inferential results are summarized using the function `summary`. According to the p-value for the z-test, the actual significance of `PSI` seems to be not so effective.

```
mod.glm <- glm(GRADE ~ PSI , family = binomial, data = program)
summary(mod.glm)
```

```

Call:
glm(formula = GRADE ~ PSI, family = binomial, data = program)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3018  -0.6039  -0.6039   1.0579   1.8930

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.6094     0.6324  -2.545   0.0109 *
PSI           1.8971     0.8317   2.281   0.0225 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 41.183  on 31  degrees of freedom
Residual deviance: 35.342  on 30  degrees of freedom
AIC: 39.342

Number of Fisher Scoring iterations: 3
```

The present model, which involves a single factor predictor, corresponds to the study of the so-called odds ratios. Firstly, the contingency table related to the dichotomous variables `PSI` and `GRADE` can be obtained and the two odds can be defined, together with their logarithmic transformation and, finally, their ratio. Since the observed proportion of `GRADE`=1 is $3/18 = 0.167$, for students with `PSI`=0, and $8/14 = 0.571$, for students with `PSI`=1, the two corresponding odds are $0.167/(1 - 0.167) = 0.2$ and $0.571/(1 - 0.571) = 1.33$.

```
conttable <- table(program$PSI, program$GRADE)
conttable
```

```
    0  1
  0 15  3
  1  6  8

odds <- conttable[,2]/conttable[,1]
odds

       0        1
0.200000 1.333333

log(odds)

         0          1
-1.6094379  0.2876821

OR <- odds[2]/odds[1]
OR

       1
6.666667
```

Notice that the log odds correspond, respectively, to the first regression coefficient and to the sum of the two regression coefficients, as estimated within the previous logistic regression model. Indeed, the estimate of the second regression coefficient corresponds to the logarithmic transformation of the odds ratio.

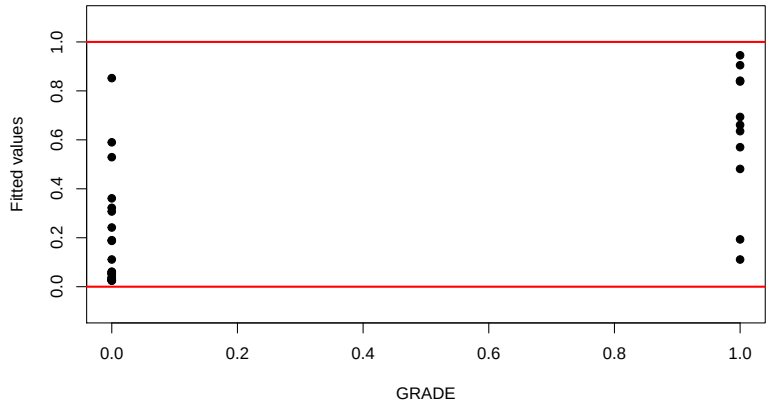
```
mod.glm$coef[1]

(Intercept)
  -1.609438

mod.glm$coef[1]+mod.glm$coef[2]

(Intercept)
  0.2876821

mod.glm$coef[2]

    PSI
1.89712

log(OR)

      1
1.89712
```

The full model, where also the effects due to `GPA` and `TUCE` are taken into account, can then be introduced by considering the following logistic multiple regression model with both numerical and factor predictors. From the output of function `summary`, we may conclude that not all the predictors induce a significant effect.

```
mod.glm.all <- glm(GRADE ~ PSI + TUCE + GPA, family = binomial, data = program)
summary(mod.glm.all)


Call:
glm(formula = GRADE ~ PSI + TUCE + GPA, family = binomial, data = program)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9551  -0.6453  -0.2570   0.5888   2.0966

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.02135    4.93127  -2.641  0.00828 **
PSI           2.37869    1.06456   2.234  0.02545 *
TUCE          0.09516    0.14155   0.672  0.50143
GPA           2.82611    1.26293   2.238  0.02524 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 41.183  on 31  degrees of freedom
Residual deviance: 25.779  on 28  degrees of freedom
AIC: 33.779

Number of Fisher Scoring iterations: 5
```

In this case, the scatterplot of the fitted values (namely, the estimated probability of `GRADE=1`) and the observed values of the response variable `GRADE` presents values in the y-axis which are all within the valid range $[0, 1]$. The fitted values correspond to the element `fitted.values` of the object `mod.glm.all`, which gives the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.

```
plot(program$GRADE,mod.glm.all$fitted.values,pch=19,ylim=c(-0.1,1.1),xlab="GRADE",
     ylab="Fitted values")
abline(0,0,col='red',lwd=2)
abline(1,0,col='red',lwd=2)
```



In this framework, it can be useful to compare the observed data for the response variable `GRADE` with the predictions obtained using the estimated logit model. To this end, we consider the estimated values for the linear predictor, with respect to the observed values of the explanatory variables, and then their transformation using the inverse of the logistic link function. These values correspond to the estimated probabilities of `GRADE=1` and they can be obtained with the command `mod.glm.all$fitted.values`. Alternatively, they can be calculated by applying the inverse of the link function to the fitted (predicted) values of the linear predictor, given by `predict(mod.glm.all)`. Finally, in order to transform the estimated probabilities in dichotomous predictions, we consider the threshold 0.5, so that values equal or greater than 0.5 correspond to `GRADE=1` and values lower than 0.5, to `GRADE=0`. The R function `as.numeric` is used to transform the logical values `TRUE` and `FALSE` in the numerical values `1` and `0`, respectively.

```
pred <- as.numeric(exp(predict(mod.glm.all))/(1+exp(predict(mod.glm.all)))>0.5)
```

The contingency table given below is useful for comparing the observed and the predicted values of the response variable `GRADE`. An evaluation of the predictive accuracy of the estimated logistic model is given by the percentage of correct classifications, which can be obtained by computing the proportion of cases lying in the diagonal.

```
table(pred,program$GRADE)


pred  0  1
   0 18  3
   1  3  8

(18+8)/(18+3+3+8)

[1] 0.8125
```

Since the same data are used twice, for estimating the model and for evaluating its predictive ability, the predictive performance is over-estimated. A more correct predictive assessment uses a

cross-validation procedure. The R function `CVbinary` of the library `DAAG` can be used to compute the cross-validation accuracy for the fitted logit model `mod.glm.all`.

```
library(DAAG)
CVbinary(mod.glm.all)


Fold:  9 3 6 2 8 5 4 7 10 1
Internal estimate of accuracy = 0.812
Cross-validation estimate of accuracy = 0.812
```

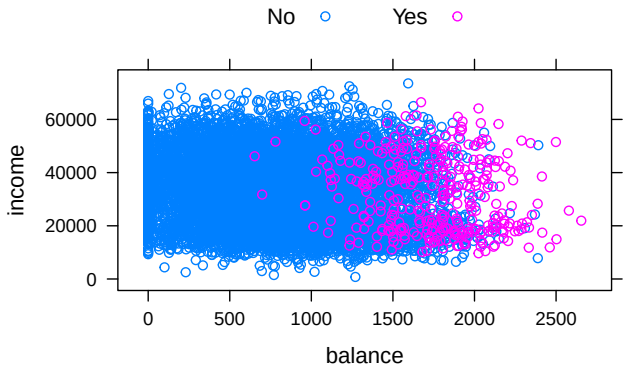
4.2 Example: credit card

We consider the data frame `Default` of the library `ISLR`, containing information on the defaults on credit card payments. For 10000 customers, the following four variables are observed:

- `default`, which is a factor with levels `No` and `Yes` indicating whether the customer defaulted on their debt in a given period;
- `student`, which is a factor with levels `No` and `Yes` indicating whether the customer is a student;
- `income`, which is the annual income of the customer;
- `balance`, which is the monthly credit card balance of the customer.

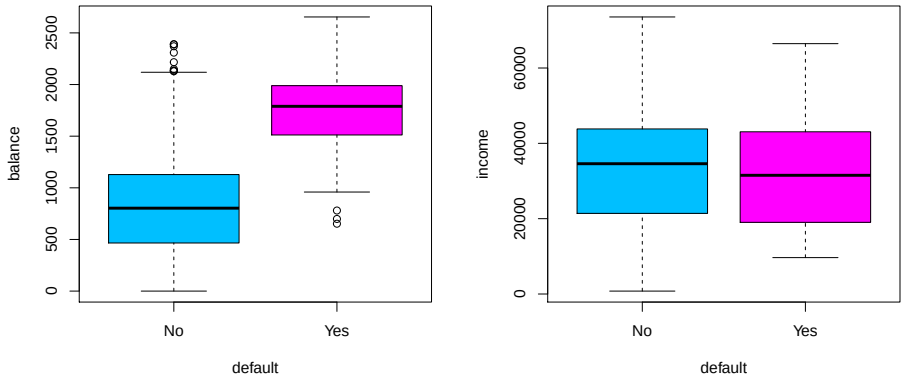
In order to describe the potential effect of `income` and `balance` on the binary response variable `default`, we consider the following scatterplot obtained with the function `xyplot` of the library `lattice`. For obtaining two different colors, for individuals who defaulted and individuals who did not, the `groups` argument is specified, while the `auto.key` argument is used to define automatically the title printed in two columns. We find out that only about 3% of people in the data set actually default and individuals who defaulted tended to have higher credit card balances than those who did not. On the other hand, the income of the costumer does not seem relevant

```
library(ISLR)
library(lattice)
attach(Default)
xyplot(income ~ balance, groups=default,data=Default,auto.key=list(columns=2))
```



The same conclusion can be reached by considering the following boxplot conditional representation.

```
par(mfrow=c(1,2))
boxplot(balance~default, data=Default,xlab="default",ylab="balance",
        col=c("deepskyblue","magenta"))
boxplot(income~default, data=Default,xlab="default",ylab="income",
        col=c("deepskyblue","magenta"))
```




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

A first model for describing the binary response variable `default` is the (simple) logistic regression model based on the explanatory variable `balance`. The model fitting outcomes, given by the function `summary`, confirms that the actual significance of `balance` is effective, since the p-value for the corresponding z-test is rather low.

```
credit1<-glm(default~balance,family="binomial", data=Default)
summary(credit1)


Call:
glm(formula = default ~ balance, family = "binomial", data = Default)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1596.5  on 9998  degrees of freedom
AIC: 1600.5

Number of Fisher Scoring iterations: 8
```

Moreover, the estimate for the regression coefficient of `balance` is 0.0055 and then we may conclude that an increase in `balance` produces an increase in the probability of default: more precisely, a one-unit increase in `balance` increases the log(odds) of `default` by 0.0055 units.

Once the coefficients are estimated, it is possible to make predictions on the variable `default` by computing the probability of `default=Yes` for an individual with a given credit card balance. The predicted probability of default can be obtained using the function `predict`, applied to the object `credit1`, with the new values for the variable `balance` defined in a suitable data frame. The option `type = c("response")` assures that the function will give the predicted probabilities, while, the default option gives predictions on the scale of the linear predictor (namely, the log odds, which are probabilities on logit scale). With the following command we compute the predicted probability of default for individuals with balances of \$1000 and \$2000, respectively.

```
predict(credit1,data.frame(balance=c(1000,2000)),type = c("response"))

          1           2
0.005752145 0.585769370
```

An alternative (simple) logistic regression model can be considered using, as explanatory variable, the factor `student`, coded as a dummy variable. Then, the categorical variable `student` is transformed into a numerical dummy variable `studentD` with values `1` and `0`, if `student="Yes"` and `student="No"`, respectively. A new column is added to the data frame `Default`.

```
# new dummy variable for student
Default$studentD<-0
Default$studentD[Default$student=="Yes"]<-1
credit2<-glm(default~studentD,family="binomial", data=Default)
summary(credit2)


Call:
glm(formula = default ~ studentD, family = "binomial", data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2970  -0.2970  -0.2434  -0.2434   2.6585

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
studentD     0.40489    0.11502    3.52 0.000431 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 2908.7  on 9998  degrees of freedom
AIC: 2912.7

Number of Fisher Scoring iterations: 6
```

The coefficient associated with the dummy variable has a p-value which is statistically significant. Indeed, since its estimated value is positive, students tend to have higher default probabilities than non-students. The following command computes the default probabilities for students and non-students, respectively.

```
predict(credit2,data.frame(studentD=c(1,0)),type = c("response"))

         1          2
0.04313859 0.02919501
```

In the present example, the same fitted logistic model can be obtained also using the command `glm(default~student,family="binomial", data=Default)`, without specifying the numerical dummy variable `studentD`.

The full model, which considers the joint effects of `income`, `balance` and `student`, can be described using the multiple logistic regression model specified below.

```
credit3<-glm(default~income+balance+studentD,family="binomial", data=Default)
summary(credit3)


Call:
glm(formula = default ~ income + balance + studentD, family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
studentD    -6.468e-01  2.363e-01  -2.738  0.00619 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

Since the estimated coefficient for the dummy variable `studentD` is negative, we may conclude that, for a fixed value of `income` and `balance`, students are less likely to default than non-students. This seems in contradiction with the previous conclusion, stating that the overall student default rate is higher. This is a confounding phenomenon due to the fact that `student` and `balance` are slightly correlated and then students tend to hold higher levels of credit card balances, which is associated with higher default rates.

4.3 Example: UCB admissions

We analyze the data set on the admission at the six largest departments of the University of California at Berkeley in the fall of 1973, classified by gender. The data are saved in the 3-dimensional array `UCBAdmissions`, available in the base R libraries, resulting from cross-tabulating 4526 observations on the following 3 categorical variables:

- `Admit`, which is a factor with levels `Admitted` and `Rejected`;
- `Gender`, which is a factor with levels `Male` and `Female`;
- `Dept`, which is a factor identifying the departments, with 6 levels (`A`, `B`, `C`, `D`, `E`, `F`).

It is well-known that the aggregate data and the department level data tell opposite stories about gender bias. Most departments have a slight female bias, while the difference on overall application and admission rates causes the aggregate bias to point in the other direction (the Simpson's paradox). In order to emphasize this fact, we define the function `odds.ratio`, which computes the odds ratio associated to a 2×2 contingency table. This function is applied to the 2×2 contingency tables related to the six departments. We conclude that the odds of males being accepted is lower than that of females for departments `B`, `D` and `F`, and strongly lower for department `A`. On the other hand, if we consider the marginal 2×2 contingency table, we obtain a marginal odds ratio of 1.84108, meaning that the odds for men being accepted is higher.

```
odds.ratio<-function(x){(x[1,1]*x[2,2])/(x[1,2]*x[2,1])}
apply(UCBAdmissions,3,odds.ratio)
```

```
        A         B         C         D         E         F
0.3492120 0.8025007 1.1330596 0.9212838 1.2216312 0.8278727
```

```
margin.table(UCBAdmissions,c(2,1))
```

```
        Admit
Gender   Admitted Rejected
  Male       1198     1493
  Female      557     1278
```

```
odds.ratio(margin.table(UCBAdmissions,c(2,1)))
```

```
[1] 1.84108
```

In order to analyze the data set using a suitable logistic regression model, the first step is to reorganize the data into a data frame. We use the function `as.data.frame.table`, which converts an array-based representation of a contingency table into a data frame containing the classifying factors and the corresponding entries. Firstly, the admission frequencies are extracted and the corresponding column is called `admit`. Secondly, the rejection frequencies are also extracted and saved in the further column `reject`. Indeed, the level `Male` is set as the reference level for the factor `Gender`. Finally, the total number of observations in each category and the proportion of admitted students are computed and saved in the last two columns of the data frame.

```
UCB <- as.data.frame.table(UCBAdmissions["Admitted", , ])
names(UCB)[3] <- "admit"
UCB$reject <- as.data.frame.table(UCBAdmissions["Rejected", , ])$Freq
UCB$Gender <- relevel(UCB$Gender, ref="Male")
```

```
UCB$total <- UCB$admit + UCB$reject
UCB$p <- UCB$admit/UCB$total
UCB
```

```
   Gender Dept admit reject total          p
1    Male    A   512    313   825 0.62060606
2  Female    A    89     19   108 0.82407407
3    Male    B   353    207   560 0.63035714
4  Female    B    17      8    25 0.68000000
5    Male    C   120    205   325 0.36923077
6  Female    C   202    391   593 0.34064081
7    Male    D   138    279   417 0.33093525
8  Female    D   131    244   375 0.34933333
9    Male    E    53    138   191 0.27748691
10 Female    E    94    299   393 0.23918575
11   Male    F    22    351   373 0.05898123
12 Female    F    24    317   341 0.07038123
```

A logistic regression model is defined in order to describe the probability of admission. The two factor explanatory variables `Dept` and `Gender` are set in this order, without considering the interaction effect. The reference level for `Dept` is `A`. The logit model is estimated considering the proportion `p` of admitted students as response variable and using the `weight` argument to define the total number of observations for each category. The same result can be obtained by considering, as observed values for the response variable, the frequencies of admitted and rejected students `cbind(admit, reject)`; in this case the `weight` argument is not required.

```
UCB.glm1 <- glm(p ~ Dept+Gender, family=binomial, data=UCB, weight=total)
summary(UCB.glm1)
```

```

Call:
glm(formula = p ~ Dept + Gender, family = binomial, data = UCB,
    weights = total)

Deviance Residuals:
      1       2       3       4       5       6       7
-1.2487  3.7189 -0.0560  0.2706  1.2533 -0.9243  0.0826
      8       9      10      11      12
-0.0858  1.2205 -0.8509 -0.2076  0.2052

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.58205    0.06899   8.436   <2e-16 ***
DeptB        -0.04340    0.10984  -0.395    0.693
DeptC        -1.26260    0.10663 -11.841   <2e-16 ***
```

```
DeptD        -1.29461    0.10582 -12.234   <2e-16 ***
DeptE        -1.73931    0.12611 -13.792   <2e-16 ***
DeptF        -3.30648    0.16998 -19.452   <2e-16 ***
GenderFemale  0.09987    0.08085   1.235    0.217
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 877.056  on 11  degrees of freedom
Residual deviance:  20.204  on  5  degrees of freedom
AIC: 103.14

Number of Fisher Scoring iterations: 4
```

Since the estimate of the regression coefficient related to the `GenderFemale` term corresponds to the log odds ratio, assuming `Male` as the reference level for `Gender`, the reciprocal of its exponential transformation gives the odds ratio of acceptance with gender, conditional on the department.

```
1/exp(UCB.glm1$coefficients[7])

GenderFemale
    0.904955
```

The conditional odds ratio of acceptance with gender is 0.905, which is significantly lower than the marginal value 1.841 given previously. Thus, conditional on the department, the odds of males being accepted is slightly lower than that of females. This holds in general for each department, since in this case we consider only the main effect of `Gender`, without the interaction with `Dept`.

Furthermore, using the `anova` function with the option `test="Chisq"`, we may compare the nested models by means of a sequence of χ^2 tests, according to the order of the explanatory variables considered in `UCB.glm1`. Then, we conclude that there is a clear effect of `Dept` on the admission rate, while there is no detectable effect of `Gender`, when effect of `Dept` is taken into account.

```
anova(UCB.glm1, test="Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: p

Terms added sequentially (first to last)
```

```
       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                      11     877.06
Dept    5   855.32         6      21.74   <2e-16 ***
Gender  1     1.53         5      20.20   0.2159
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If we modify the order of the two regressors, we obtain the same model fitting as before, but the ANOVA table does not detect properly the effects of `Gender` and `Dept`.

```
UCB.glm2 <- glm(p ~ Gender+Dept, family=binomial, data=UCB, weight=total)
summary(UCB.glm2)


Call:
glm(formula = p ~ Gender + Dept, family = binomial, data = UCB,
    weights = total)

Deviance Residuals:
      1        2        3        4        5        6        7
-1.2487   3.7189  -0.0560   0.2706   1.2533  -0.9243   0.0826
      8        9       10       11       12
-0.0858   1.2205  -0.8509  -0.2076   0.2052

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.58205    0.06899   8.436   <2e-16 ***
GenderFemale  0.09987    0.08085   1.235    0.217
DeptB        -0.04340    0.10984  -0.395    0.693
DeptC        -1.26260    0.10663 -11.841   <2e-16 ***
DeptD        -1.29461    0.10582 -12.234   <2e-16 ***
DeptE        -1.73931    0.12611 -13.792   <2e-16 ***
DeptF        -3.30648    0.16998 -19.452   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 877.056  on 11  degrees of freedom
Residual deviance:  20.204  on  5  degrees of freedom
AIC: 103.14

Number of Fisher Scoring iterations: 4

anova(UCB.glm2, test="Chisq")
```

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: p

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                      11     877.06
Gender  1    93.45        10     783.61 < 2.2e-16 ***
Dept    5   763.40         5      20.20 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Finally, we define a more general logistic regression model with the two factor explanatory variables `Dept` and `Gender` and also their interaction effect. As emphasized before, it is important to fit `Dept`, thus adjusting for different admission rates in different departments, before fitting `Gender`.

```
UCB.glm <- glm(p ~ Dept*Gender, family=binomial, data=UCB, weight=total)
summary(UCB.glm)


Call:
glm(formula = p ~ Dept * Gender, family = binomial, data = UCB,
    weights = total)

Deviance Residuals:
 [1]  0  0  0  0  0  0  0  0  0  0  0  0

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.49212    0.07175   6.859 6.94e-12 ***
DeptB               0.04163    0.11319   0.368  0.71304
DeptC              -1.02764    0.13550  -7.584 3.34e-14 ***
DeptD              -1.19608    0.12641  -9.462  < 2e-16 ***
DeptE              -1.44908    0.17681  -8.196 2.49e-16 ***
DeptF              -3.26187    0.23120 -14.109  < 2e-16 ***
GenderFemale        1.05208    0.26271   4.005 6.21e-05 ***
DeptB:GenderFemale -0.83205    0.51039  -1.630  0.10306
DeptC:GenderFemale -1.17700    0.29956  -3.929 8.53e-05 ***
DeptD:GenderFemale -0.97009    0.30262  -3.206  0.00135 **
DeptE:GenderFemale -1.25226    0.33032  -3.791  0.00015 ***
```

```
DeptF:GenderFemale -0.86318    0.40267  -2.144  0.03206 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8.7706e+02  on 11  degrees of freedom
Residual deviance: 1.4744e-13  on  0  degrees of freedom
AIC: 92.94

Number of Fisher Scoring iterations: 3
```

The first six coefficients are relate to overall admission rates, for males, in the six department. Indeed, the strongly significant positive coefficient for `GenderFemale` indicates that the log odds is increased by 1.05, in department `A`, for females relative to males. On the other hand, in departments `C`, `D`, `E` and `F`, the log odds is reduced for females, relative to males. Finally, the ANOVA table based on sequential χ^2 test confirms that there is a clear effect of `Dept` on the admission rate, while there is no detectable main effect of `Gender`. Moreover, the significant interaction term suggests that there are department-specific gender biases, which average out to reduce the main effect of `Gender` to close to zero.

```
anova(UCB.glm, test="Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: p

Terms added sequentially (first to last)


            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                           11     877.06
Dept         5   855.32         6      21.74 < 2.2e-16 ***
Gender       1     1.53         5      20.20  0.215928
Dept:Gender  5    20.20         0       0.00  0.001144 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```