Anova and Ancova

Josep Franquet

12/03/2025

One-way Anova

Prestige data with factor type

```
library(car)
library(MASS)
library(tidyverse)
library(emmeans)
library(multcomp)
library(multcompView)
library(RcmdrMisc)
```

We load `Prestige` dataset:

```
df <- Prestige
names(df)
```

```
## [1] "education" "income"    "women"     "prestige"  "census"    "type"
```

Our objective is to know if the factor "type" has an effect on the prestige target. We can first do a boxplot and a bit of descriptive analysis:

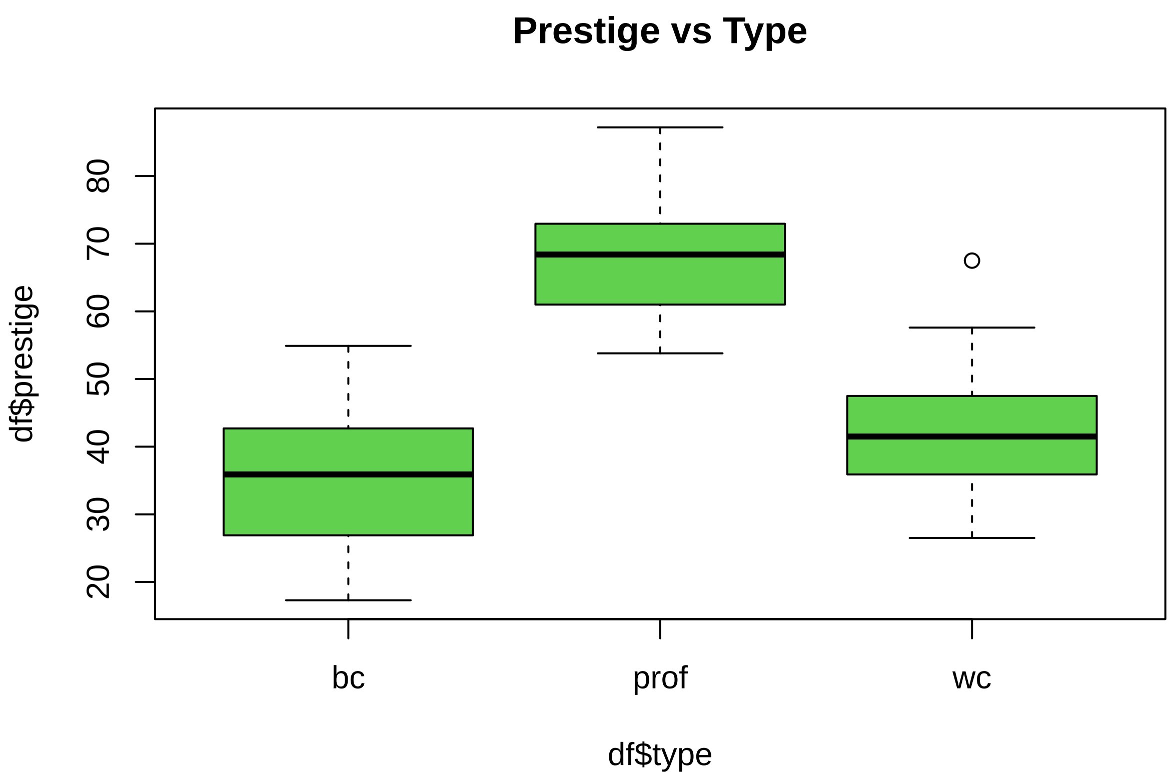
```
summary(df[, c("prestige", "type")])
```

```
##     prestige       type
##  Min.   :14.80   bc  :44
##  1st Qu.:35.23   prof:31
##  Median :43.60   wc  :23
##  Mean   :46.83   NA's: 4
##  3rd Qu.:59.27
##  Max.   :87.20
```

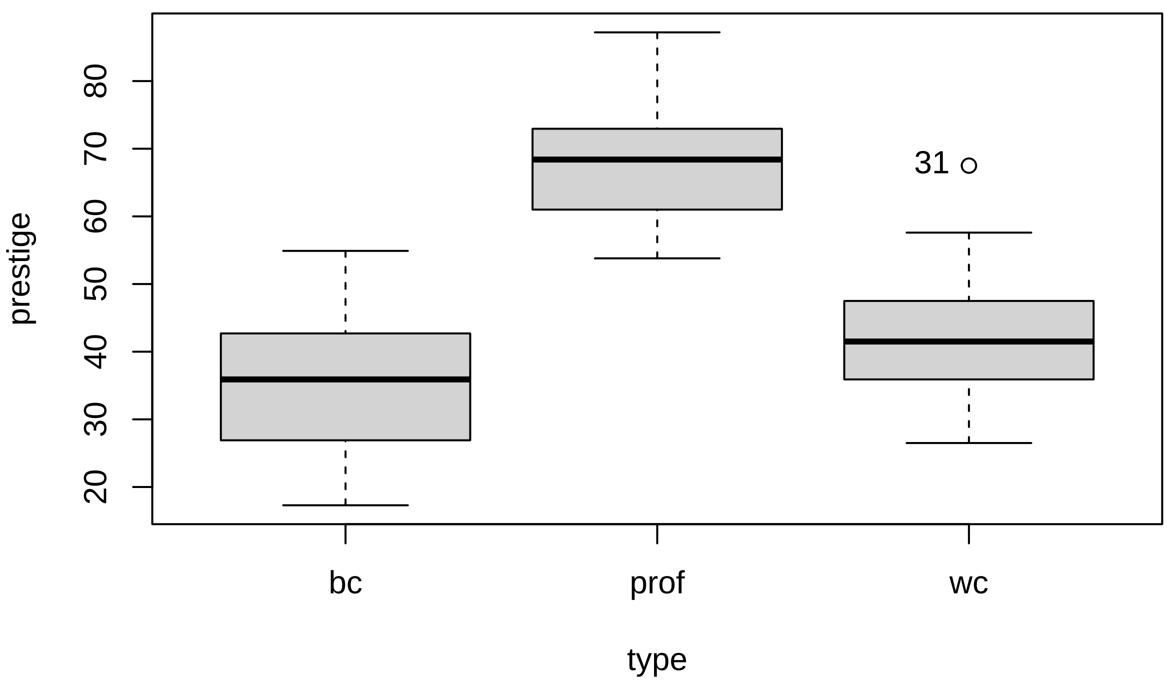
```
# We remove NAs
df <- na.omit(df)
```

To do that, we build the linear model with one factor as explicative variable (type):

```
plot(df$prestige~df$type, main="Prestige vs Type", col=3)
```

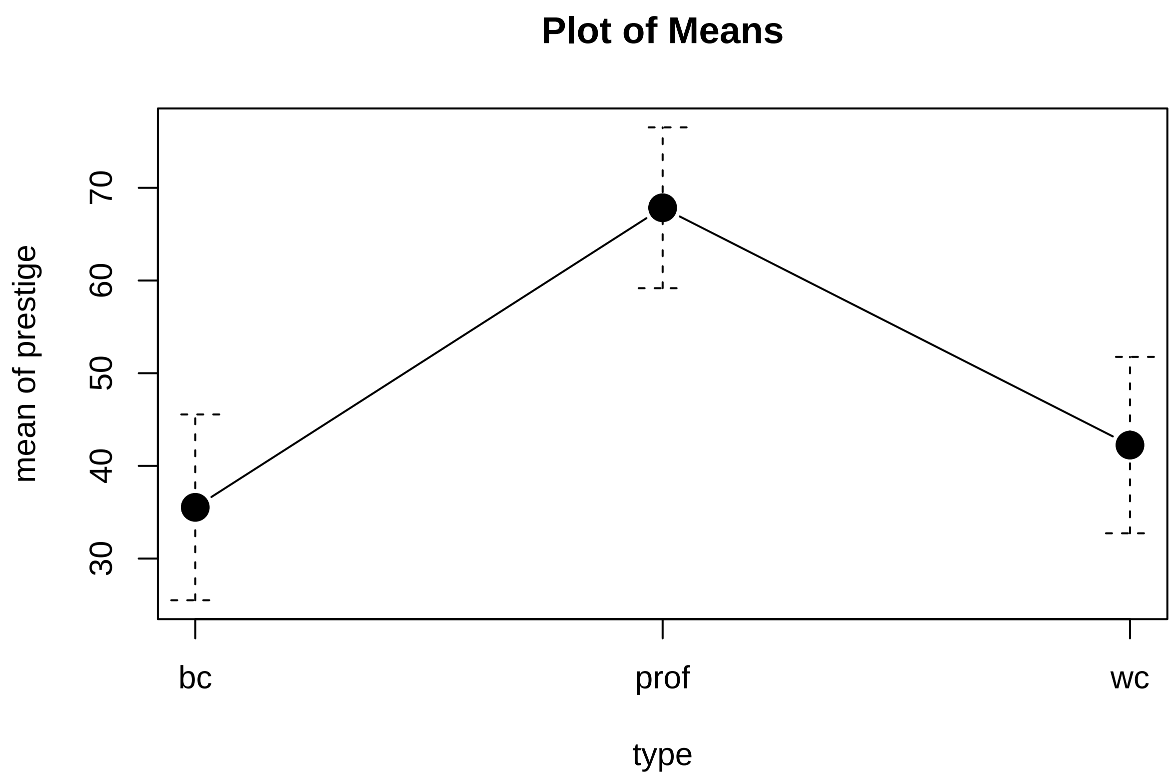



```r
scatterplot(prestige~type,df)
```

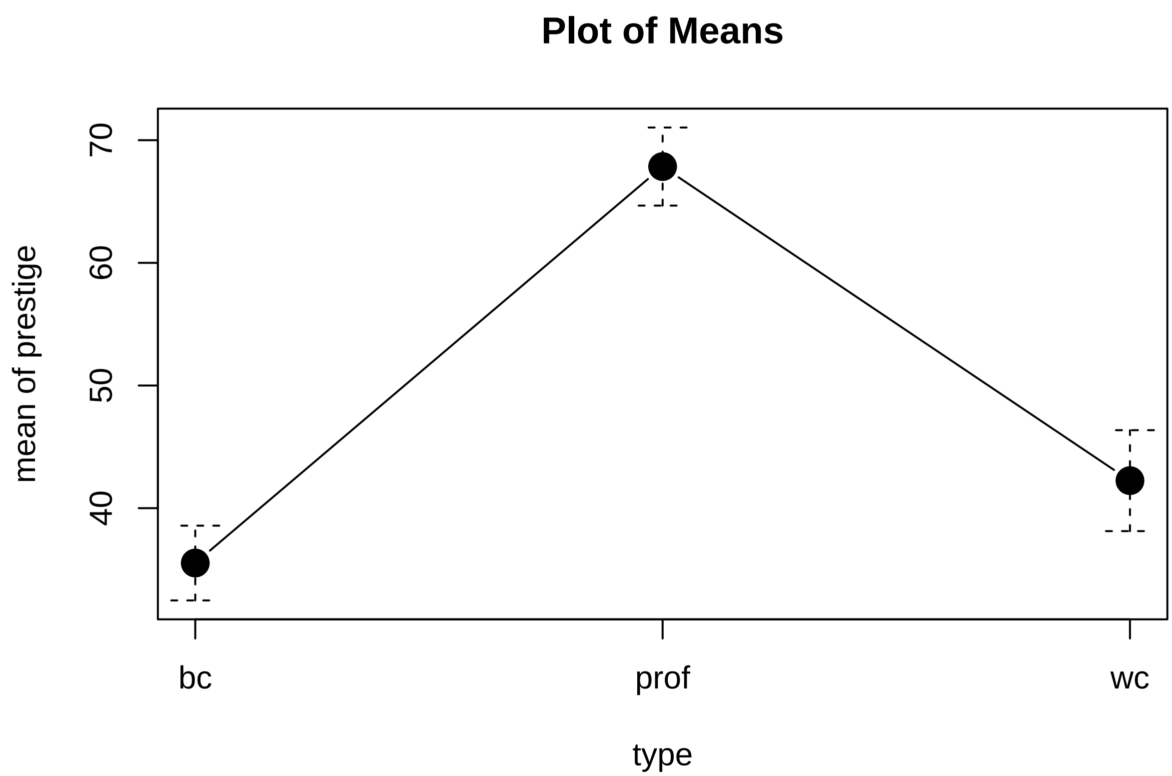



```
## [1] "31"
```

```r
with(df, plotMeans(prestige, type, error.bars = "sd"))
```




```r
with(df, plotMeans(prestige, type, error.bars = "conf.int", level=0.95))
```



We fit the model with two different types of contrasts: treatment and sum.

```r
model_treat <- lm(prestige~type, data = df, contrasts = list(type = "contr.treatment"))
summary(model_treat)
```

```
##
## Call:
## lm(formula = prestige ~ type, data = df, contrasts = list(type = "contr.treatment"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2273  -7.1773  -0.0854   6.1174  25.2565
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.527      1.432  24.810  < 2e-16 ***
## typeprof      32.321      2.227  14.511  < 2e-16 ***
## typewc         6.716      2.444   2.748  0.00718 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.499 on 95 degrees of freedom
## Multiple R-squared:  0.6976, Adjusted R-squared:  0.6913
## F-statistic: 109.6 on 2 and 95 DF,  p-value: < 2.2e-16
```

```r
model_sum <- lm(prestige~type, data = df, contrasts = list(type = "contr.sum"))
summary(model_sum)
```

```
##
## Call:
## lm(formula = prestige ~ type, data = df, contrasts = list(type = "contr.sum"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2273  -7.1773  -0.0854   6.1174  25.2565
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.5397     0.9935   48.86   <2e-16 ***
## type1       -13.0124     1.2925  -10.07   <2e-16 ***
## type2        19.3087     1.3990   13.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.499 on 95 degrees of freedom
## Multiple R-squared:  0.6976, Adjusted R-squared:  0.6913
## F-statistic: 109.6 on 2 and 95 DF,  p-value: < 2.2e-16
```

Can you interpret the parameters in both cases? Are the predictions or errors of the models different?

```r
# Test predictions are the same
cbind(predict(model_treat), predict(model_sum))
```

```
##                                [,1]     [,2]
## gov.administrators         67.84839 67.84839
## general.managers           67.84839 67.84839
## accountants                67.84839 67.84839
## purchasing.officers        67.84839 67.84839
## chemists                   67.84839 67.84839
## physicists                 67.84839 67.84839
## biologists                 67.84839 67.84839
```

```
## architects                67.84839 67.84839
## civil.engineers           67.84839 67.84839
## mining.engineers          67.84839 67.84839
## surveyors                 67.84839 67.84839
## draughtsmen               67.84839 67.84839
## computer.programers       67.84839 67.84839
## economists                67.84839 67.84839
## psychologists             67.84839 67.84839
## social.workers            67.84839 67.84839
## lawyers                   67.84839 67.84839
## librarians                67.84839 67.84839
## vocational.counsellors    67.84839 67.84839
## ministers                 67.84839 67.84839
## university.teachers       67.84839 67.84839
## primary.school.teachers   67.84839 67.84839
## secondary.school.teachers 67.84839 67.84839
## physicians                67.84839 67.84839
## veterinarians             67.84839 67.84839
## osteopaths.chiropractors  67.84839 67.84839
## nurses                    67.84839 67.84839
## nursing.aides             35.52727 35.52727
## physio.therapsts          67.84839 67.84839
## pharmacists               67.84839 67.84839
## medical.technicians       42.24348 42.24348
## commercial.artists        67.84839 67.84839
## radio.tv.announcers       42.24348 42.24348
## secretaries               42.24348 42.24348
## typists                   42.24348 42.24348
## bookkeepers               42.24348 42.24348
## tellers.cashiers          42.24348 42.24348
## computer.operators        42.24348 42.24348
## shipping.clerks           42.24348 42.24348
## file.clerks               42.24348 42.24348
## receptionsts              42.24348 42.24348
## mail.carriers             42.24348 42.24348
## postal.clerks             42.24348 42.24348
## telephone.operators       42.24348 42.24348
## collectors                42.24348 42.24348
## claim.adjustors           42.24348 42.24348
## travel.clerks             42.24348 42.24348
## office.clerks             42.24348 42.24348
## sales.supervisors         42.24348 42.24348
## commercial.travellers     42.24348 42.24348
## sales.clerks              42.24348 42.24348
## service.station.attendant 35.52727 35.52727
## insurance.agents          42.24348 42.24348
## real.estate.salesmen      42.24348 42.24348
## buyers                    42.24348 42.24348
## firefighters              35.52727 35.52727
## policemen                 35.52727 35.52727
## cooks                     35.52727 35.52727
## bartenders                35.52727 35.52727
## funeral.directors         35.52727 35.52727
## launderers                35.52727 35.52727
```

```
## janitors                    35.52727 35.52727
## elevator.operators          35.52727 35.52727
## farm.workers                35.52727 35.52727
## rotary.well.drillers        35.52727 35.52727
## bakers                      35.52727 35.52727
## slaughterers.1              35.52727 35.52727
## slaughterers.2              35.52727 35.52727
## canners                     35.52727 35.52727
## textile.weavers             35.52727 35.52727
## textile.labourers           35.52727 35.52727
## tool.die.makers             35.52727 35.52727
## machinists                  35.52727 35.52727
## sheet.metal.workers         35.52727 35.52727
## welders                     35.52727 35.52727
## auto.workers                35.52727 35.52727
## aircraft.workers            35.52727 35.52727
## electronic.workers          35.52727 35.52727
## radio.tv.repairmen          35.52727 35.52727
## sewing.mach.operators       35.52727 35.52727
## auto.repairmen              35.52727 35.52727
## aircraft.repairmen          35.52727 35.52727
## railway.sectionmen          35.52727 35.52727
## electrical.linemen          35.52727 35.52727
## electricians                35.52727 35.52727
## construction.foremen        35.52727 35.52727
## carpenters                  35.52727 35.52727
## masons                      35.52727 35.52727
## house.painters              35.52727 35.52727
## plumbers                    35.52727 35.52727
## construction.labourers      35.52727 35.52727
## pilots                      67.84839 67.84839
## train.engineers             35.52727 35.52727
## bus.drivers                 35.52727 35.52727
## taxi.drivers                35.52727 35.52727
## longshoremen                35.52727 35.52727
## typesetters                 35.52727 35.52727
## bookbinders                 35.52727 35.52727
```

We stick to the treatment model:

```
model <- model_treat
```

Is the addition of the factor significative?

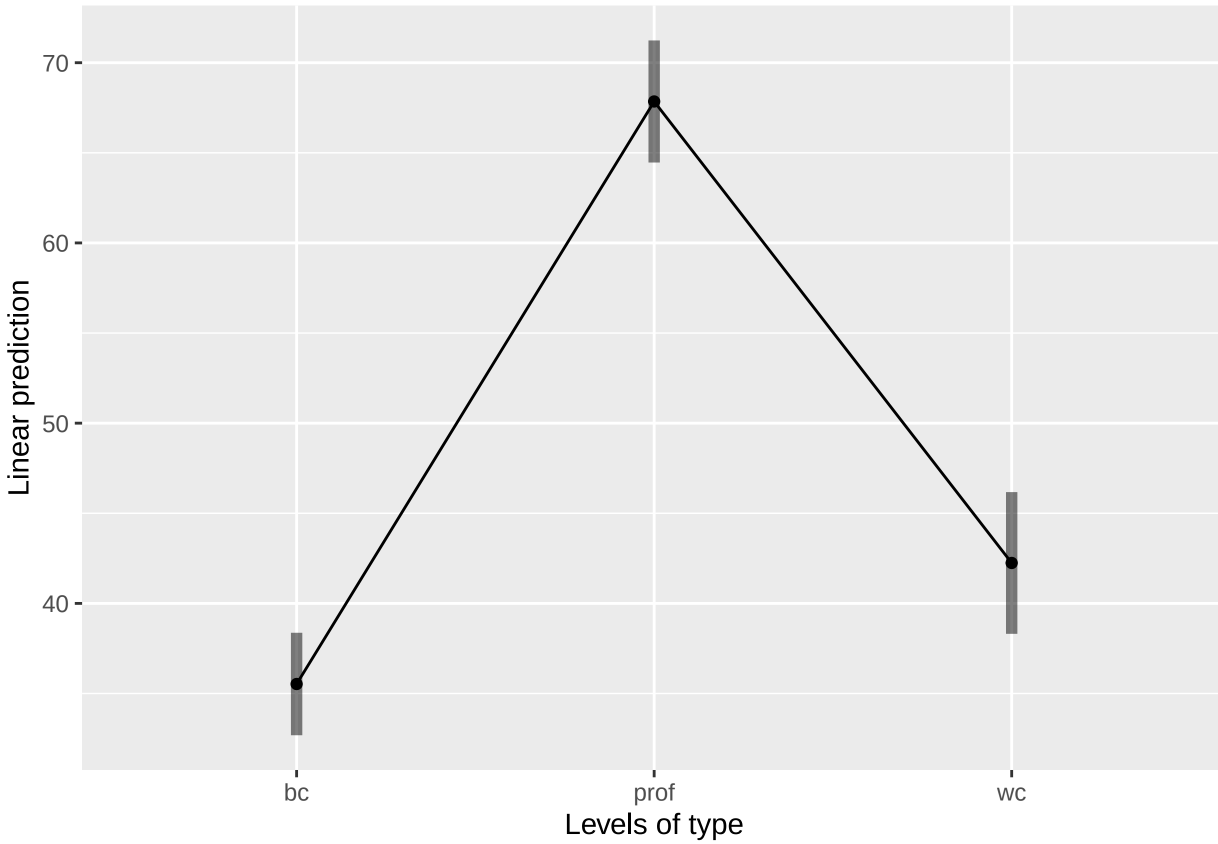
```
Anova(model)
```

```
## Anova Table (Type II tests)
##
## Response: prestige
##            Sum Sq Df F value    Pr(>F)
## type      19775.6  2  109.59 < 2.2e-16 ***
## Residuals  8571.3 95
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

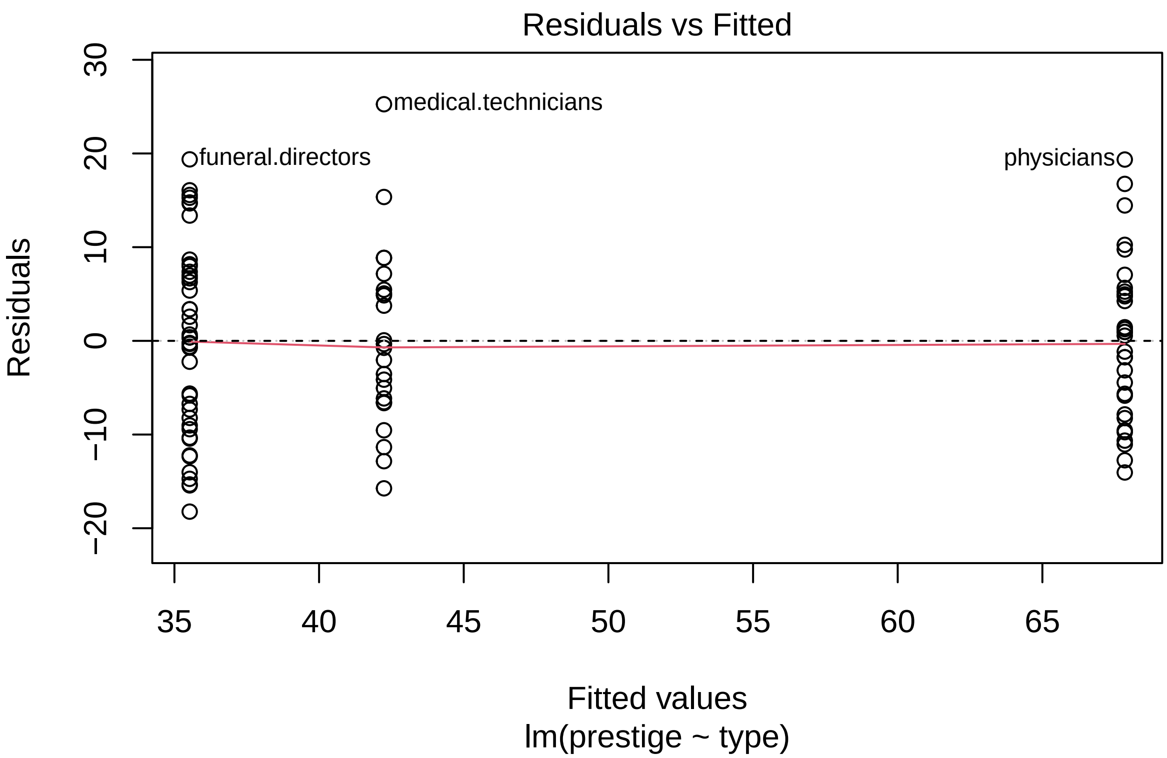
```
model_0 <- lm(prestige~1, df)
anova(model_0, model)
```

```
## Analysis of Variance Table
##
## Model 1: prestige ~ 1
## Model 2: prestige ~ type
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     97 28346.9
## 2     95  8571.3  2     19776 109.59 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

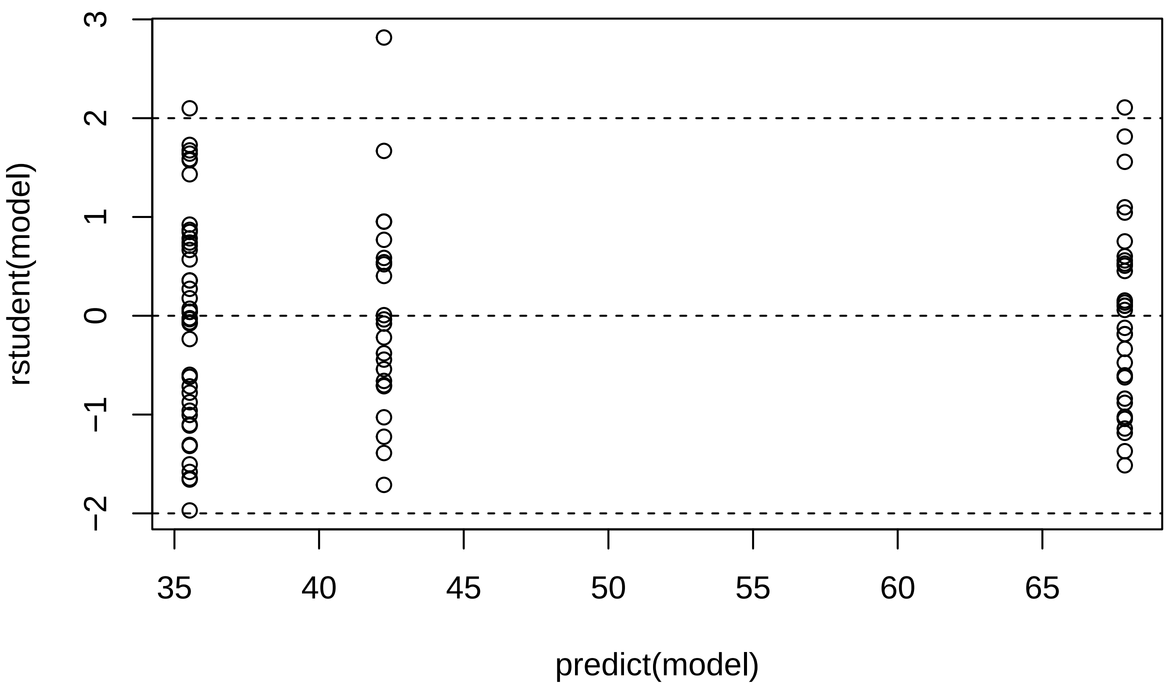
```r
emmip(model, ~type, CIs=T)
```



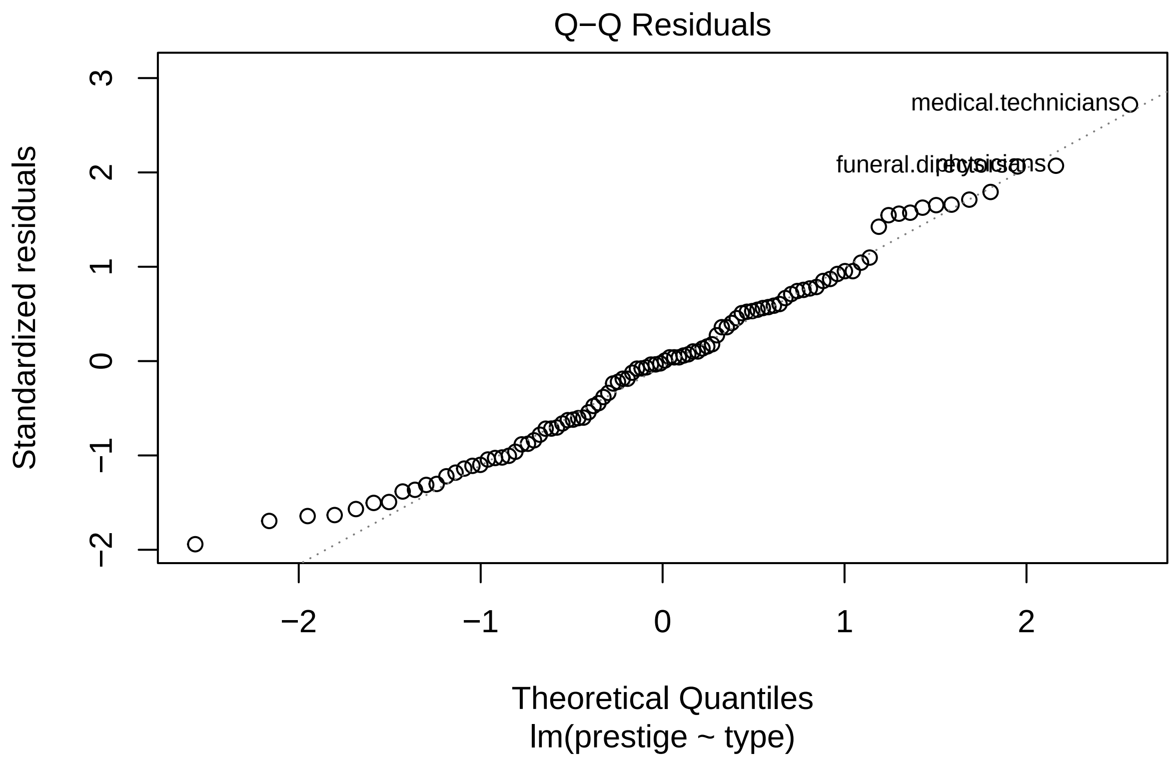

```r
# Diagnostics
plot(model, which=1)
abline(h=0, lty=2)
```




```r
plot(predict(model), rstudent(model))
abline(h=c(-2, 0, 2), lty=2)
```




```r
plot(model, which=2)
```



Two-way Anova

Ancova

Prestige data with factor type

We load `Prestige` dataset:

```
df <- Prestige
names(df)
```

```
## [1] "education" "income"    "women"     "prestige"  "census"    "type"
```

We ended up with this final model:

```
model_final <- lm(prestige ~ education + log(income) + type, data = df)
summary(model_final)
```

```
##
## Call:
## lm(formula = prestige ~ education + log(income) + type, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.511  -3.746   1.011   4.356  18.438
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -81.2019    13.7431  -5.909 5.63e-08 ***
## education     3.2845     0.6081   5.401 5.06e-07 ***
## log(income)  10.4875     1.7167   6.109 2.31e-08 ***
```

```
## typeprof      6.7509     3.6185   1.866   0.0652 .
## typewc       -1.4394     2.3780  -0.605   0.5465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.637 on 93 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8555, Adjusted R-squared:  0.8493
## F-statistic: 137.6 on 4 and 93 DF,  p-value: < 2.2e-16
```

We can now extend this model to explore interactions of the type:

```
model_final_ext <- lm(prestige ~ (education + log(income))*type, data = df)
summary(model_final_ext)
```

```
##
## Call:
## lm(formula = prestige ~ (education + log(income)) * type, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.970  -4.124   1.206   3.829  18.059
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -120.0459    20.1576  -5.955 5.07e-08 ***
## education                2.3357     0.9277   2.518  0.01360 *
## log(income)             15.9825     2.6059   6.133 2.32e-08 ***
## typeprof                85.1601    31.1810   2.731  0.00761 **
## typewc                  30.2412    37.9788   0.796  0.42800
## education:typeprof       0.6974     1.2895   0.541  0.58998
## education:typewc         3.6400     1.7589   2.069  0.04140 *
## log(income):typeprof    -9.4288     3.7751  -2.498  0.01434 *
## log(income):typewc      -8.1556     4.4029  -1.852  0.06730 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.409 on 89 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.871,  Adjusted R-squared:  0.8595
## F-statistic: 75.15 on 8 and 89 DF,  p-value: < 2.2e-16
```

There are some interactions that seem significative. We can check if this model is better with the interaction (factorial) or it does not improve and we stick to additive efects (additive):

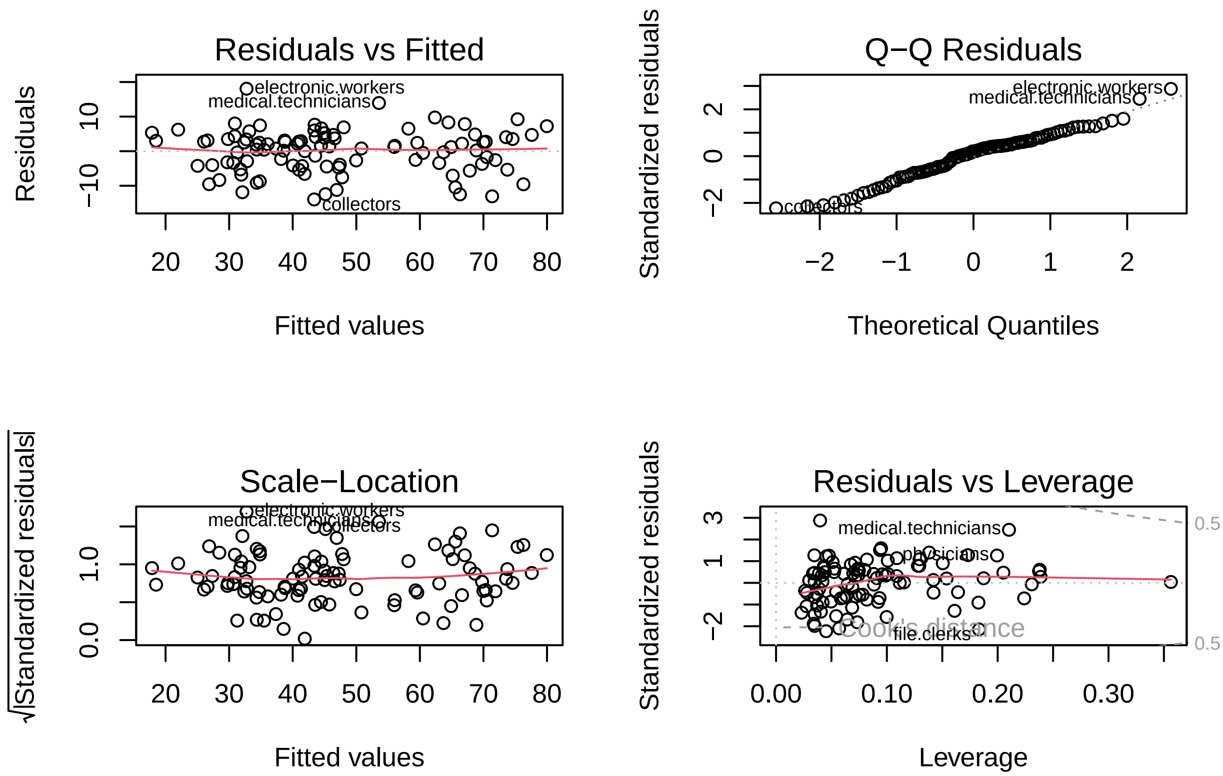
```
anova(model_final, model_final_ext)
```

```
## Analysis of Variance Table
##
## Model 1: prestige ~ education + log(income) + type
## Model 2: prestige ~ (education + log(income)) * type
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     93 4096.3
## 2     89 3655.4  4    440.89 2.6836 0.03646 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

At 5% signification it is better to add interaction.

Standard model plots:

```r
par(mfrow=c(2,2))
plot(model_final_ext)
```



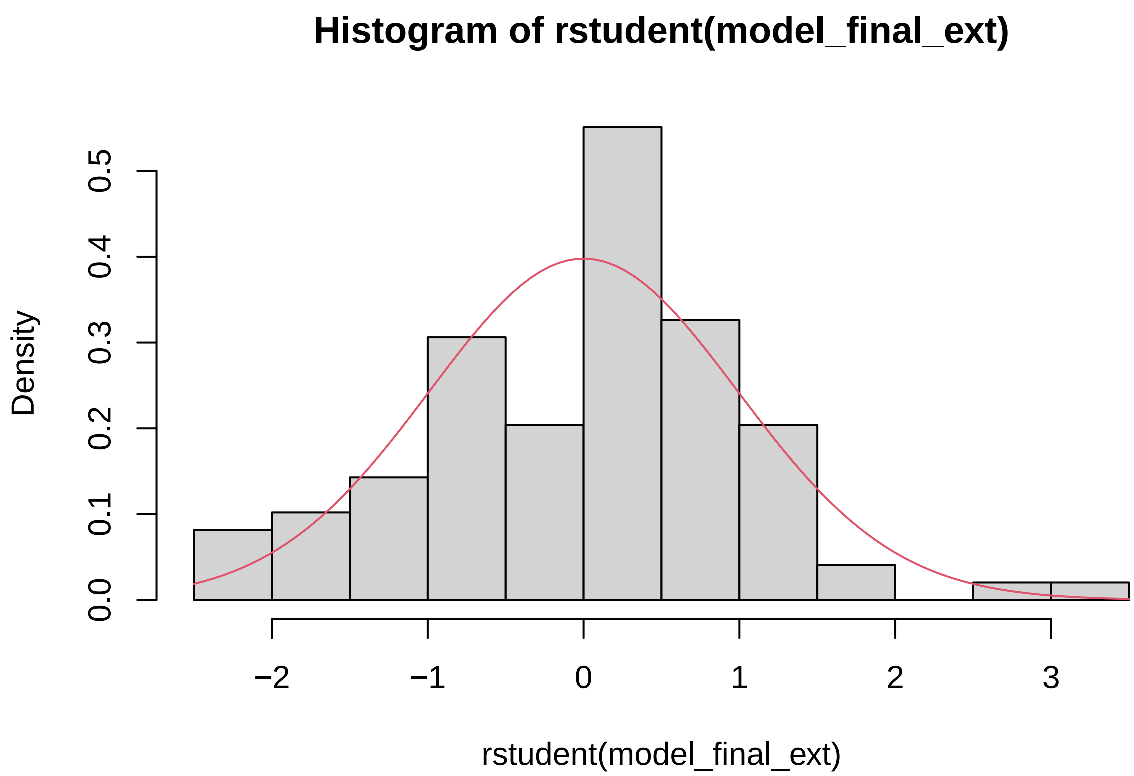

```r
par(mfrow=c(1,1))
```

Model Validation

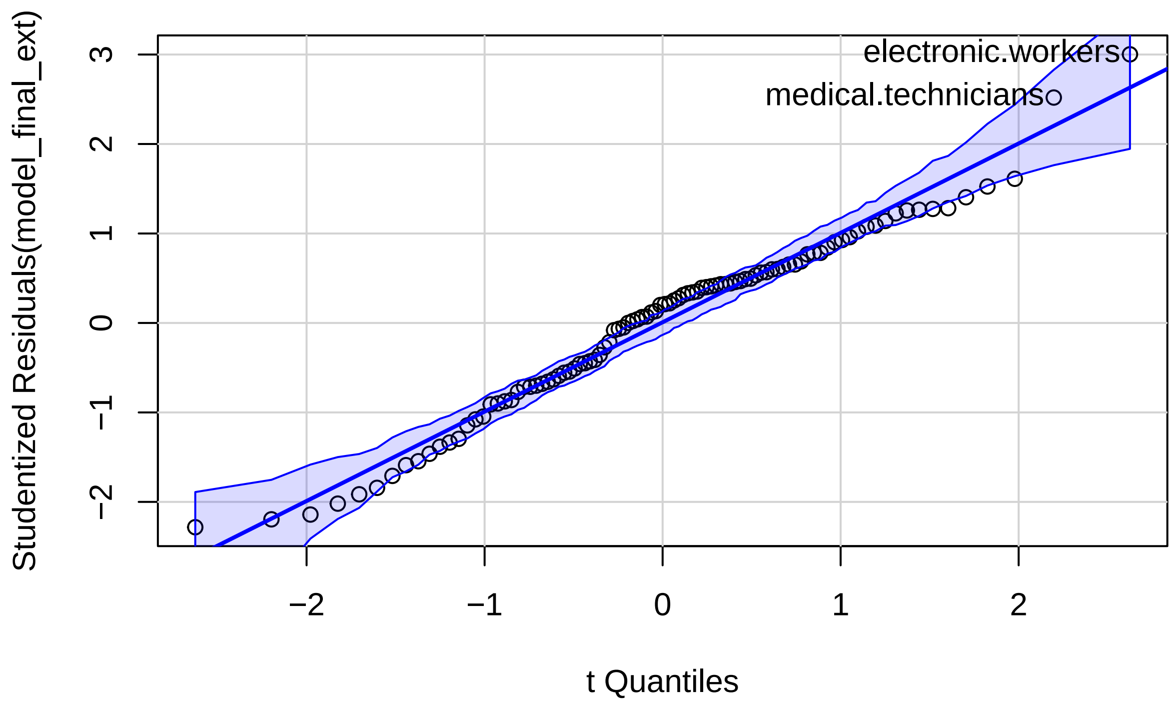
Residual analysis constitutes a practical tool for graphically assessing model fitting and satisfaction of optimal hypothesis for OLS estimates.

Usual plots:

```r
# Histogram of studentized residuals
hist(rstudent(model_final_ext), freq=F)
curve(dt(x, model_final_ext$df), col=2, add=T)
```



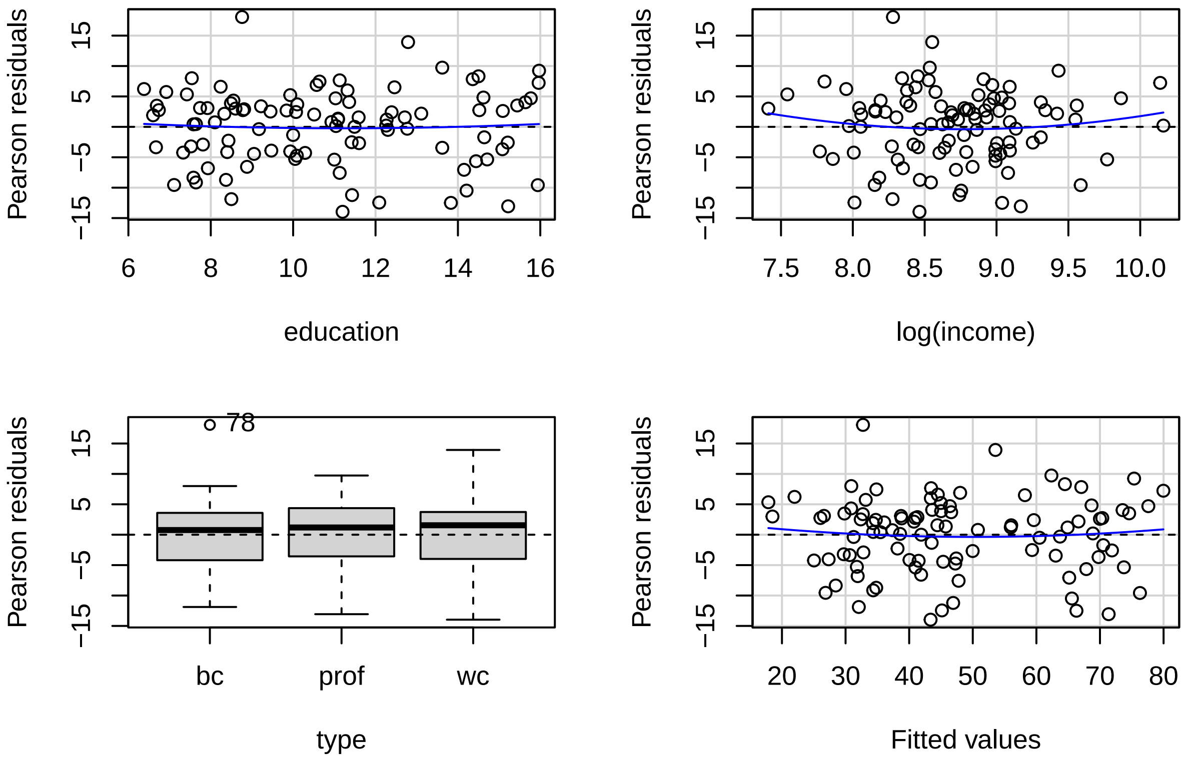

```r
## QQ Plot for normality
qqPlot(model_final_ext, simulate=T, labels=F)
```




```
## medical.technicians  electronic.workers
##                  31                  82
```

We have more functions to check linearity satisfaction and homoskedastic hypothesis. The horizontal band indicates them:

```r
residualPlots(model_final_ext)
```




```
##              Test stat Pr(>|Test stat|)
## education       1.5863          0.11627
## log(income)     1.8719          0.06455 .
## type
## Tukey test      2.1569          0.03101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have as well a Homoskedastic Hypothesis Test - Breusch-Pagan test in package lmtest might be of interes:

```r
library(lmtest)
```

```
## S'està carregant el paquet requerit: zoo

##
## S'està adjuntant el paquet: 'zoo'

## Els següents objectes estan emmascarats des de 'package:base':
##
##     as.Date, as.Date.numeric
```

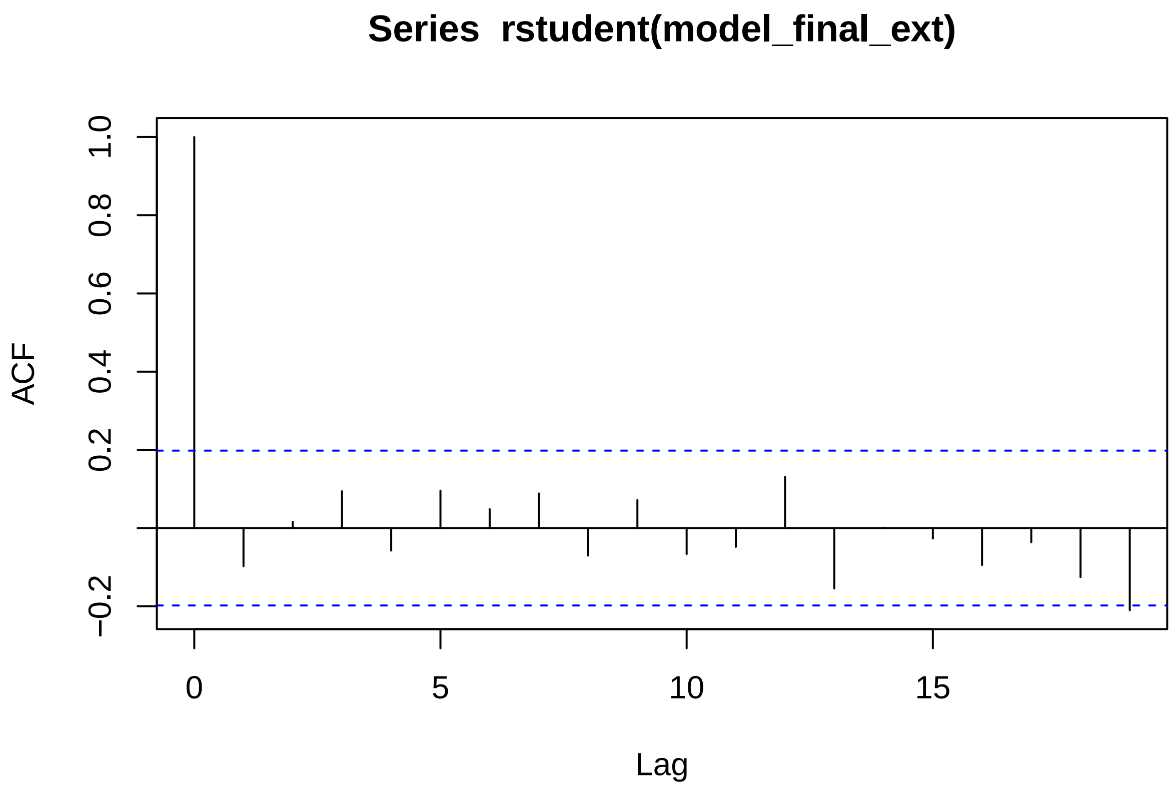
```r
bptest(model_final_ext)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_final_ext
## BP = 12.265, df = 8, p-value = 0.1398
```

In this case, we can't reject homoskedasticity.

To test uncorrelation of the residuals (residual vs time/order or any omitted variable in the model suspected to affect hypothesis) we can use acf:

```r
acf(rstudent(model_final_ext))
```

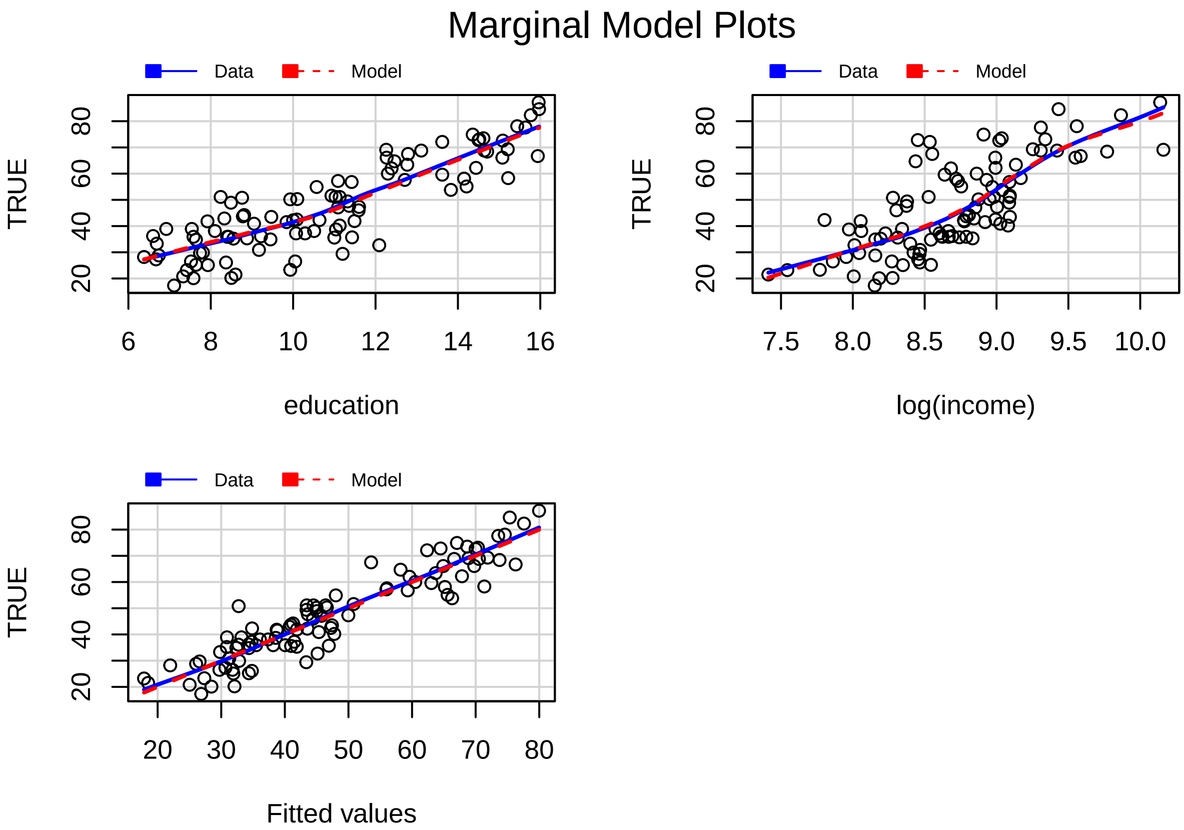


Model transformations on Y or X

Use `marginalModelPlots(model)` method in package car for R. Lack of fit between data smoother and current model behavior for one variable indicates that transformation on selected regressor is needed.

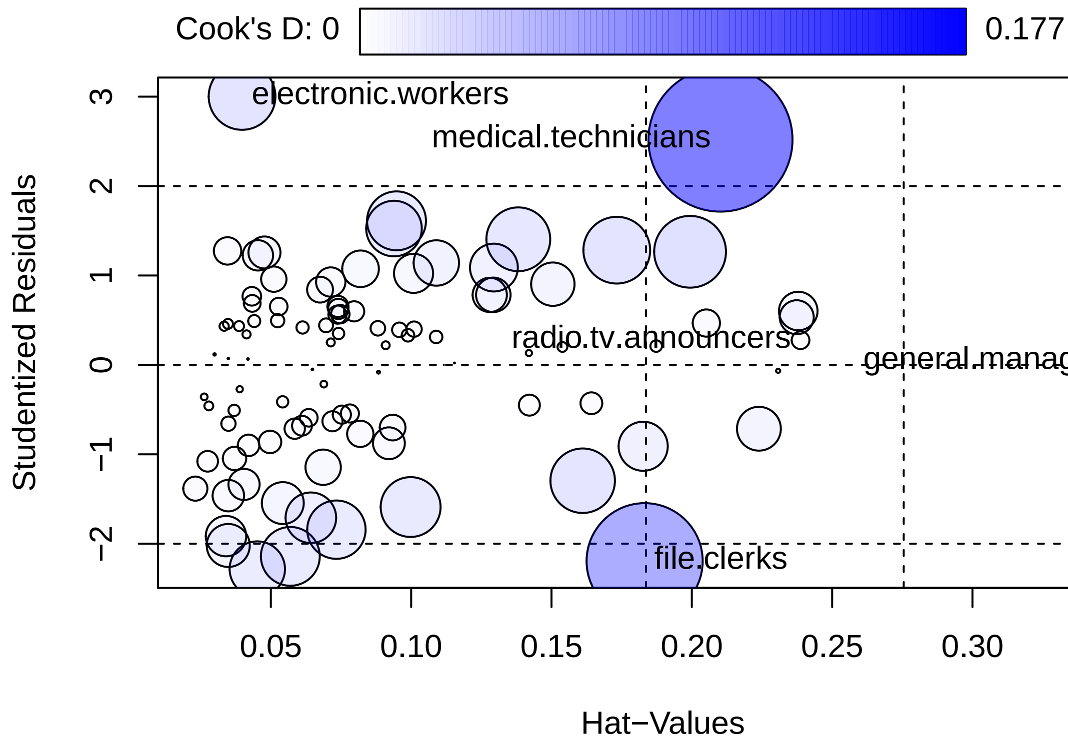
```r
marginalModelPlots(model_final_ext)
```

```
## Warning in mmps(...): Interactions and/or factors skipped
```



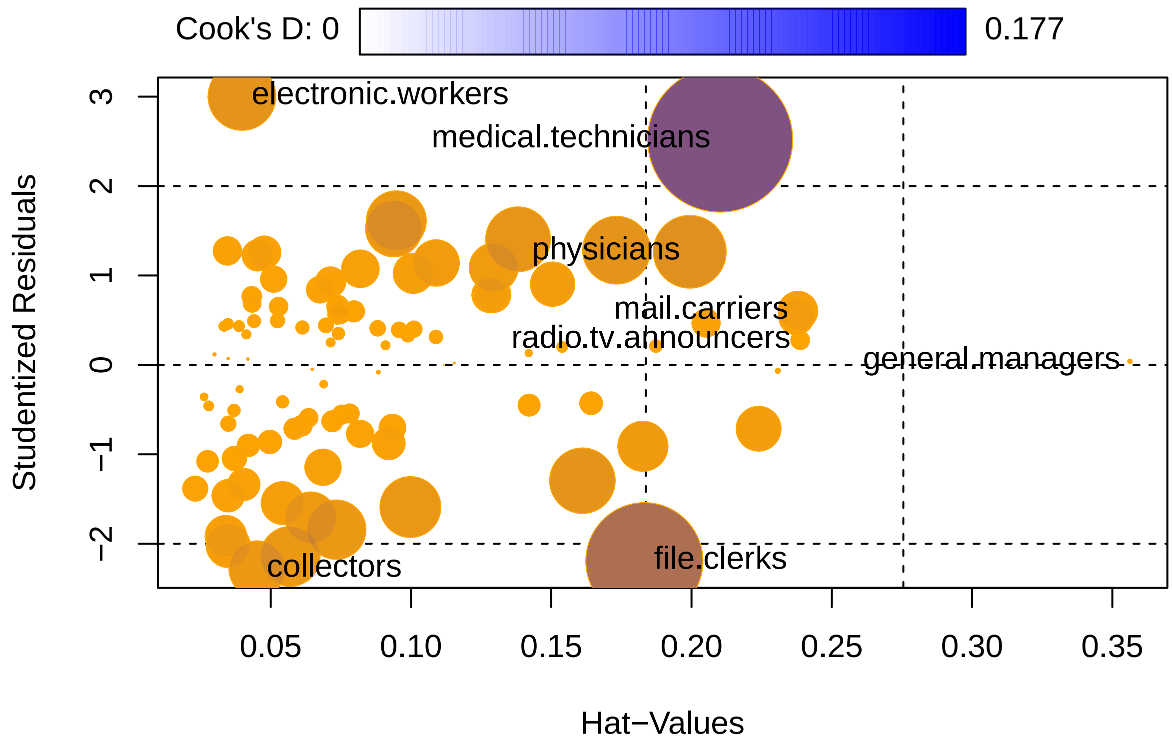
Unusual and influential data

```
influencePlot(model_final_ext)
```




```
##                            StudRes        Hat        CookD
## general.managers     0.04009503 0.35629278 9.999001e-05
## medical.technicians  2.51921523 0.21021056 1.770499e-01
## radio.tv.announcers  0.27620307 0.23874092 2.686209e-03
## file.clerks         -2.19476523 0.18319130 1.151014e-01
## electronic.workers   3.00234280 0.03974919 3.803444e-02
```

```
influencePlot(model_final_ext,
              col="orange",
              pch=19,
              id=list(method="noteworthy",n=3))
```




```
##                          StudRes        Hat        CookD
## general.managers      0.04009503 0.35629278 9.999001e-05
## physicians            1.26527876 0.19938371 4.400203e-02
## medical.technicians   2.51921523 0.21021056 1.770499e-01
## radio.tv.announcers   0.27620307 0.23874092 2.686209e-03
## file.clerks          -2.19476523 0.18319130 1.151014e-01
## mail.carriers         0.60271435 0.23793137 1.269277e-02
## collectors           -2.28296893 0.04517471 2.616058e-02
## electronic.workers    3.00234280 0.03974919 3.803444e-02
```

Influential observations imply that the inclusion of the data in OLS modify the vector of estimated parameter and the fitted values.

DFBetas

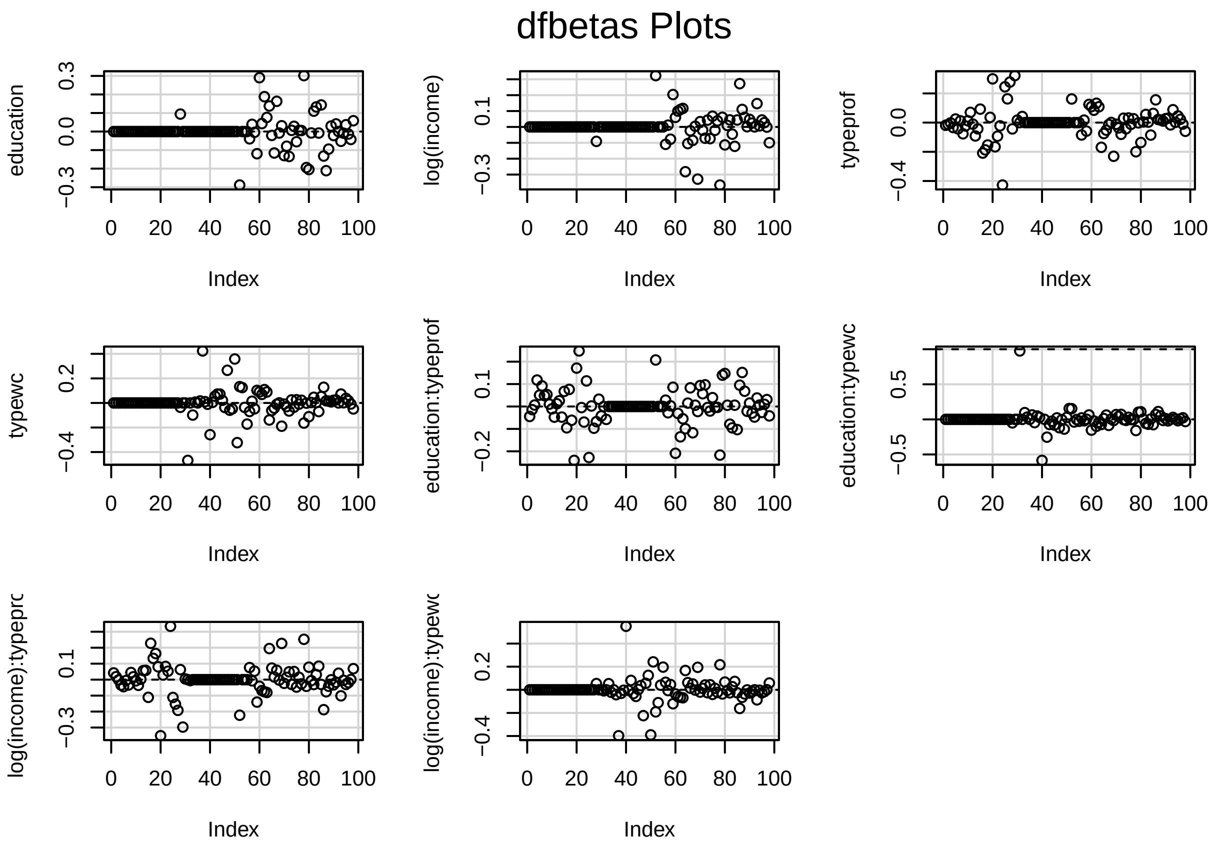
The most direct approach to assessing influence is to assess how the regression coefficients change if outliers are omitted from the model. We can use DFBetas_ij). Use `dfbetas(model)` in R.

```
head(dfbetas(model_final_ext))
```

```
##                        (Intercept)     education   log(income)     typeprof
## gov.administrators     6.538858e-15  1.489033e-15 -6.472561e-15 -0.018525957
## general.managers      -5.850257e-16 -3.291786e-17  5.363924e-16 -0.012210572
## accountants           -1.213078e-16 -4.071971e-18  1.107616e-16 -0.001862753
## purchasing.officers    2.433797e-15 -4.696393e-16 -2.053816e-15 -0.033828887
## chemists              -1.093594e-16  6.043572e-17  9.039659e-17  0.024676244
## physicists            -1.650530e-16 -4.575344e-17  1.453720e-16 -0.042763392
##                             typewc education:typeprof education:typewc
## gov.administrators    -5.775739e-15       -0.044470499    -8.548841e-17
## general.managers       2.583071e-16       -0.014028578     8.435965e-17
## accountants            8.329421e-17        0.006479325     1.948746e-17
```

```
## purchasing.officers -5.312145e-15        0.118006677     1.201035e-15
## chemists             8.096171e-17        0.050621176    -5.947008e-16
## physicists           2.952486e-16        0.092002165     5.214072e-17
##                     log(income):typeprof log(income):typewc
## gov.administrators           0.042254793       5.819927e-15
## general.managers             0.018681921      -2.960812e-16
## accountants                 -0.002033284      -9.437731e-17
## purchasing.officers         -0.034287273       4.802797e-15
## chemists                    -0.044159148       2.120784e-16
## physicists                  -0.005700716      -3.186343e-16
```

```r
dfbetasPlots(model_final_ext)
```



Cook's D

To overcome the problem of having a 2D object we have Cook's Dthat presents a single summary measure for each observation. Use `cooks.distance(model)` in R.

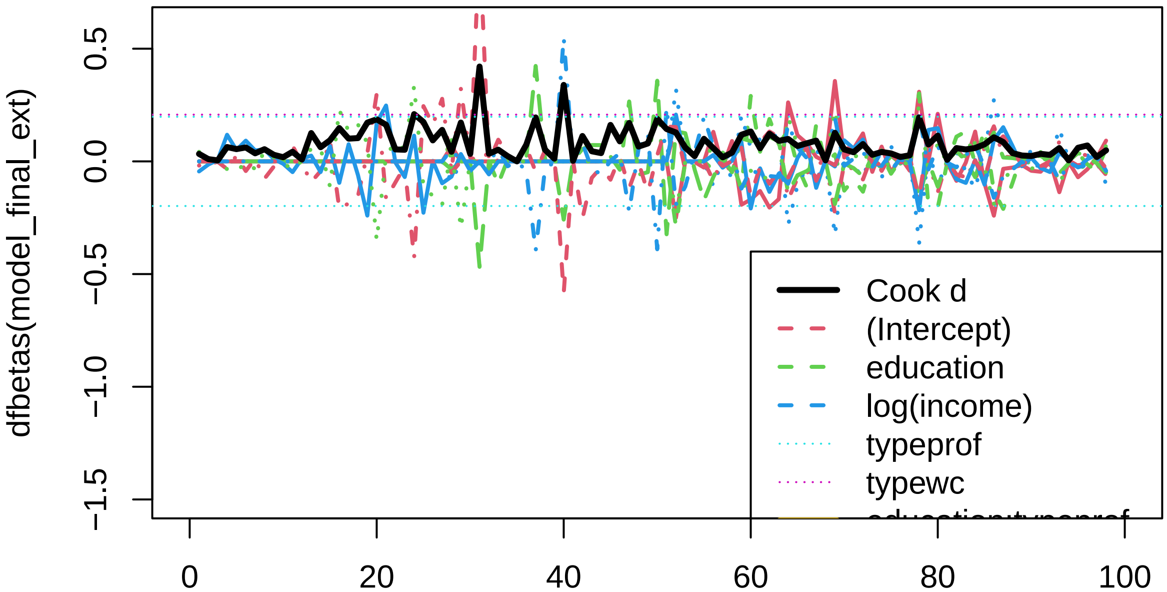
```r
head(cooks.distance(model_final_ext))
```

```
##  gov.administrators    general.managers         accountants purchasing.officers
##        1.108628e-03        9.999001e-05        1.951420e-05        4.050968e-03
##            chemists          physicists
##        2.969913e-03        3.813927e-03
```

We can plot both together and see the relationship:

```r
matplot(dfbetas(model_final_ext), type = "l",
        col=2:4, lwd=2, xlim = c(0, 100), ylim = c(-1.5, 0.6))
```

```r
lines(sqrt(cooks.distance(model_final_ext)), col=1, lwd=3)
abline(h = 2/sqrt(dim(df)[1]), lty=3, lwd=1, col=5)
abline(h = -2/sqrt(dim(df)[1]), lty=3, lwd=1, col=5)
abline(h = sqrt(4/(dim(df)[1]-length(names(coef(model_final_ext))))),
       lty=3, lwd=1, col=6)
llegenda <- c("Cook d", names(coef(model_final_ext)), "DFBETA Cut-off", "Ch-H Cut-off")
# legend(locator(n=1), legend=llegenda,
#        col=1:length(llegenda), lty=c(1,2,2,2,3,3), lwd=c(3,2,2,2,1,1))
legend(x = 60, y = -0.4, legend=llegenda,
       col=1:length(llegenda), lty=c(1,2,2,2,3,3), lwd=c(3,2,2,2,1,1))
```



DFFits

One can argue that if the final objective is rather predictive than explicative, one can use the difference in the fitted values rather than in the beta parameters. DFFits are related to Cook's distance and combine studentized residuals and leverages. Use `dffits(model)` in R.

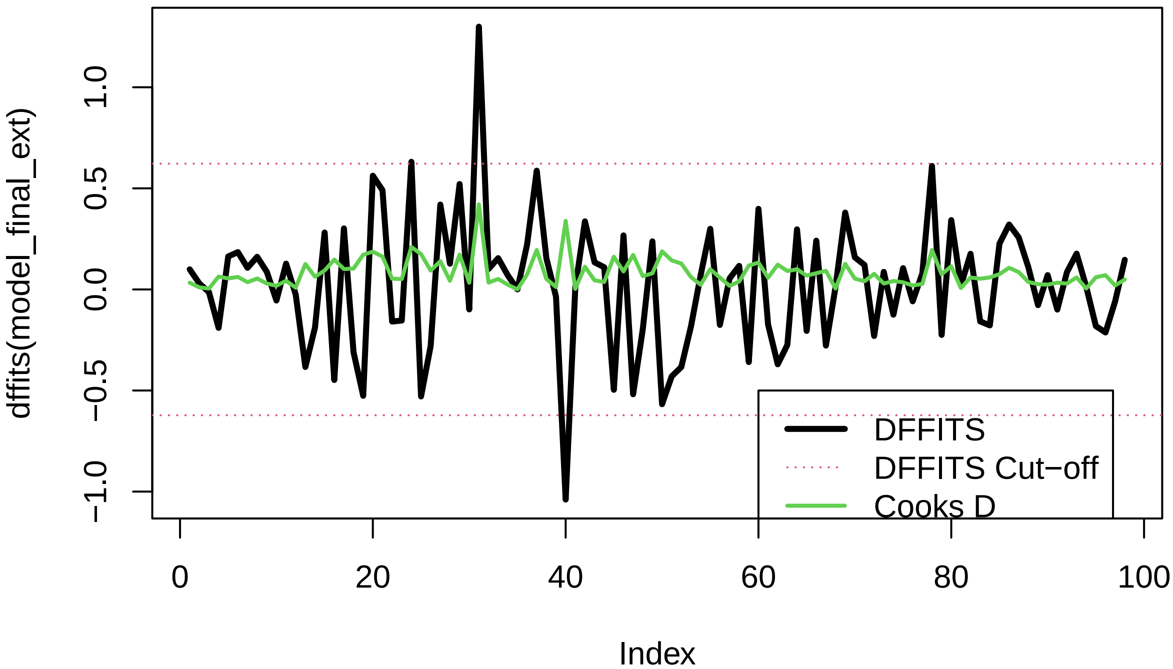
```r
head(dffits(model_final_ext))
```

```
##  gov.administrators    general.managers         accountants purchasing.officers
##          0.09939505          0.02982977         -0.01317799         -0.19006414
##            chemists          physicists
##          0.16311255          0.18467396
```

```r
# influence(m2)
```

```r
plot(dffits(model_final_ext), type="l", lwd=3)
pp = length(names(coef(model_final_ext)))
lines(sqrt(cooks.distance(model_final_ext)), col=3, lwd=2)
abline(h = 2*(sqrt(pp/(nrow(df)-pp))), lty=3, lwd=1, col=2)
abline(h = -2*(sqrt(pp/(nrow(df)-pp))),lty=3, lwd=1, col=2)
llegenda <- c("DFFITS", "DFFITS Cut-off", "Cooks D")
# legend(locator(n=1), legend = llegenda,
#        col=1:3, lty=c(1,3,1), lwd=c(3,1,2))
legend(x = 60, y = -0.5, legend = llegenda,
```

```
col=1:3, lty=c(1,3,1), lwd=c(3,1,2))
```



Best Model Selection

The best regression equation for Y given the regressors $(X_1, \dots, X_p)$ might contain dummy variables, transformations of the original variables and terms related to polynomial regression (higher order rather than linear for covariate variables) for the original variables $(Z_1, \dots, Z_q)$. Model selection should satisfy trade-off between simplicity and goodness of fit, often called parsimony criteria.

1. As many regressors as necessary to make good predictions, on average and with the highest precision in confidence interval.
2. Many variables are expensive to obtain (data collection) and difficult to maintain.

The elements available to assess the quality of a particular multiple regression (goodness of fit) model are:

1. Determination coefficient R^2.
2. Stability of the standard error of regression estimate.
3. Residual analysis.
4. Unusual and influential data analysis.
5. Information Criteria:

- Akaike Information Criteria (AIC) $AIC = 2(-l(\hat{\beta}, y) + p)$. Models with lower values of AIC indicator are preferred.
- Bayesian Information Criteria (BIC) $BIC = -2l(\hat{\beta}, y) + p \log n$. Models with lower values of BIC indicator are preferred. where extra parameters are penalized.

In R, for AIC on model objects for which a log-likelihood value can be obtained and `AIC(model)`. For BIC, `AIC(model, k=log(nrow(data.frame)))`.

Stepwise regression

- Backward elimination is a heuristic strategy to select the best model given a number of regressors and a maximal model built from them. It is a robust method that suppresses insignificant terms from the maximal model to the point that all the terms maintained are statistically significant and cannot be removed. It has been proven to be very effective for polynomial regression.

- Forward inclusion is a heuristic strategy to select the best model given a set of regressors from the null model by iteratively adding terms and regressors to the target set. It is not a robust procedure and it is not recommended as an automatic procedure to find the best model for a data set and regressor terms.
- Stepwise regression is a forward strategy that builds on the starting model but, at each iteration, regressor terms are checked for statistical significance.

R software implements these heuristics in a sophisticated way in the method `step(model, target model)` based on AIC criteria for model selection at each step.

```
lm0 <- lm(prestige~1, data = df)
step(lm0, ~income+education+women, direction = "forward", data=df)
```

```
## Start:  AIC=581.41
## prestige ~ 1
##
##             Df Sum of Sq   RSS    AIC
## + education  1   21608.4  8287 452.54
## + income     1   15279.3 14616 510.42
## <none>                   29895 581.41
## + women      1     418.6 29477 581.97
##
## Step:  AIC=452.54
## prestige ~ education
##
##          Df Sum of Sq    RSS    AIC
## + income  1   2248.14 6038.9 422.26
## + women   1    876.71 7410.3 443.14
## <none>                8287.0 452.54
##
## Step:  AIC=422.26
## prestige ~ education + income
##
##         Df Sum of Sq    RSS    AIC
## <none>               6038.9 422.26
## + women  1    5.2806 6033.6 424.17

##
## Call:
## lm(formula = prestige ~ education + income, data = df)
##
## Coefficients:
## (Intercept)    education       income
##   -6.847779     4.137444     0.001361
```

```
lm1 <- lm(prestige~education + income + women + type, data = df)
step(lm1, direction = "backward", data=df)
```

```
## Start:  AIC=390.86
## prestige ~ education + income + women + type
##
##             Df Sum of Sq    RSS    AIC
## - women      1      2.29 4681.3 388.90
## <none>                   4679.0 390.86
## - type       2    583.08 5262.1 398.36
```

```
## - income      1    803.92 5482.9 404.39
## - education  1   1635.49 6314.5 418.23
##
## Step:  AIC=388.9
## prestige ~ education + income + type
##
##             Df Sum of Sq    RSS    AIC
## <none>                    4681.3 388.90
## - type       2    591.16 5272.4 396.56
## - income     1   1058.77 5740.0 406.89
## - education  1   1655.47 6336.7 416.58
```

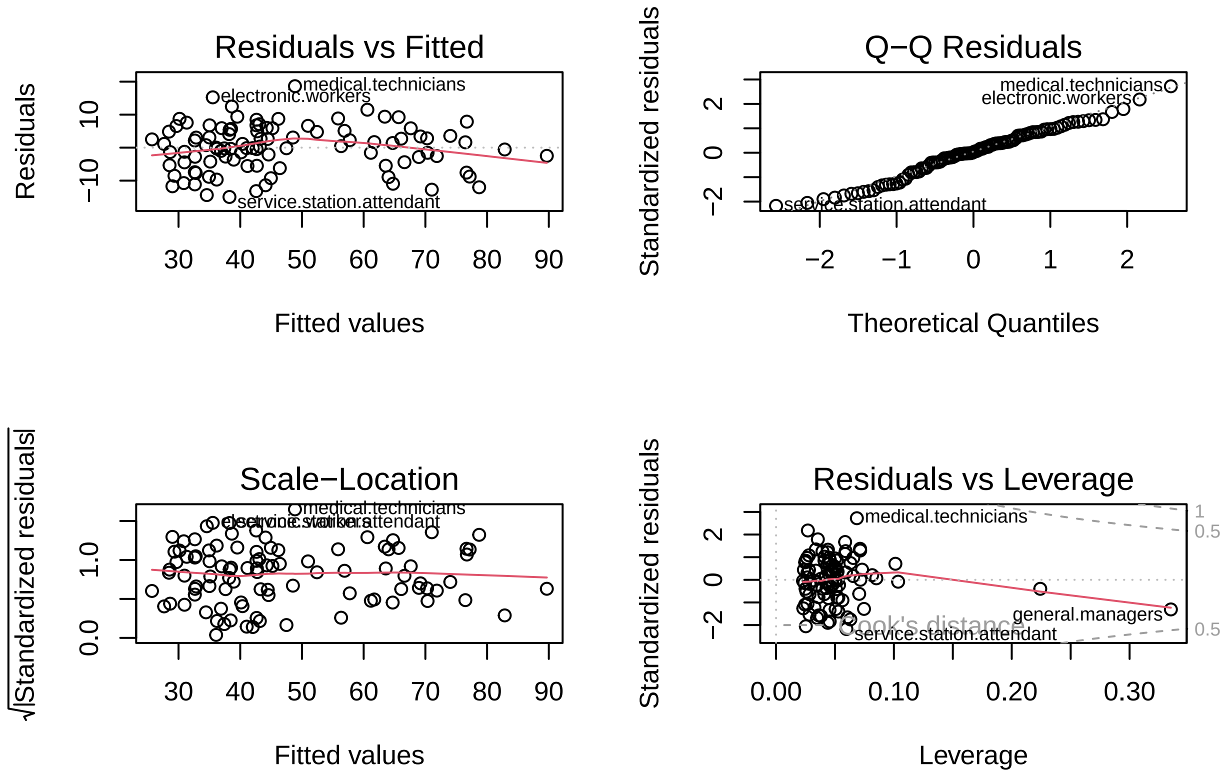
```
##
## Call:
## lm(formula = prestige ~ education + income + type, data = df)
##
## Coefficients:
## (Intercept)    education       income     typeprof       typewc
##   -0.622929     3.673166     0.001013     6.038971    -2.737231
```

Using all data available, we define a final model:

```
model_final <- lm(prestige~education + income + type, data = df)
summary(model_final)
```

```
##
## Call:
## lm(formula = prestige ~ education + income + type, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9529  -4.4486   0.1678   5.0566  18.6320
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.6229292  5.2275255  -0.119    0.905
## education    3.6731661  0.6405016   5.735 1.21e-07 ***
## income       0.0010132  0.0002209   4.586 1.40e-05 ***
## typeprof     6.0389707  3.8668551   1.562    0.122
## typewc      -2.7372307  2.5139324  -1.089    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.095 on 93 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8349, Adjusted R-squared:  0.8278
## F-statistic: 117.5 on 4 and 93 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model_final)
```




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

```r
Anova(model_final)
```

```
## Anova Table (Type II tests)
##
## Response: prestige
##            Sum Sq Df F value    Pr(>F)
## education 1655.5  1 32.8882 1.205e-07 ***
## income    1058.8  1 21.0339 1.405e-05 ***
## type       591.2  2  5.8721  0.003966 **
## Residuals 4681.3 93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_no_type <- lm(prestige~education + income, data = df)
# anova(model_no_type, model_final)  # Falla, type té NAs!!

which(is.na(df), arr.ind = T)  # Which type are these?
```

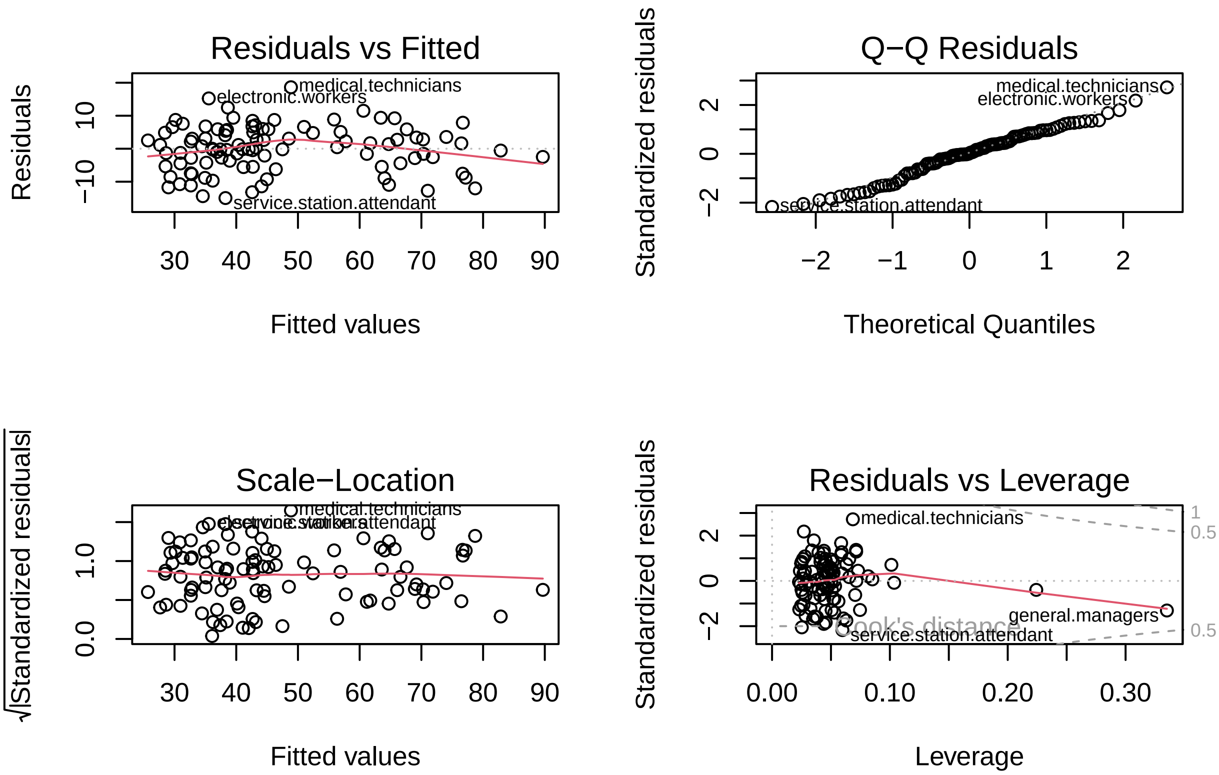
```
##             row col
## athletes     34   6
## newsboys     53   6
## babysitters  63   6
## farmers      67   6
```

```r
# We try to remove them:
df2 <- df %>% na.omit()

model_final2 <- lm(prestige~education + income + type, data = df2)
summary(model_final2)
```

```
##
## Call:
## lm(formula = prestige ~ education + income + type, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9529  -4.4486   0.1678   5.0566  18.6320
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.6229292  5.2275255  -0.119    0.905
## education    3.6731661  0.6405016   5.735 1.21e-07 ***
## income       0.0010132  0.0002209   4.586 1.40e-05 ***
## typeprof     6.0389707  3.8668551   1.562    0.122
## typewc      -2.7372307  2.5139324  -1.089    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.095 on 93 degrees of freedom
## Multiple R-squared:  0.8349, Adjusted R-squared:  0.8278
## F-statistic: 117.5 on 4 and 93 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model_final2)
```




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

Anova(model_final2)
```

```
## Anova Table (Type II tests)
```

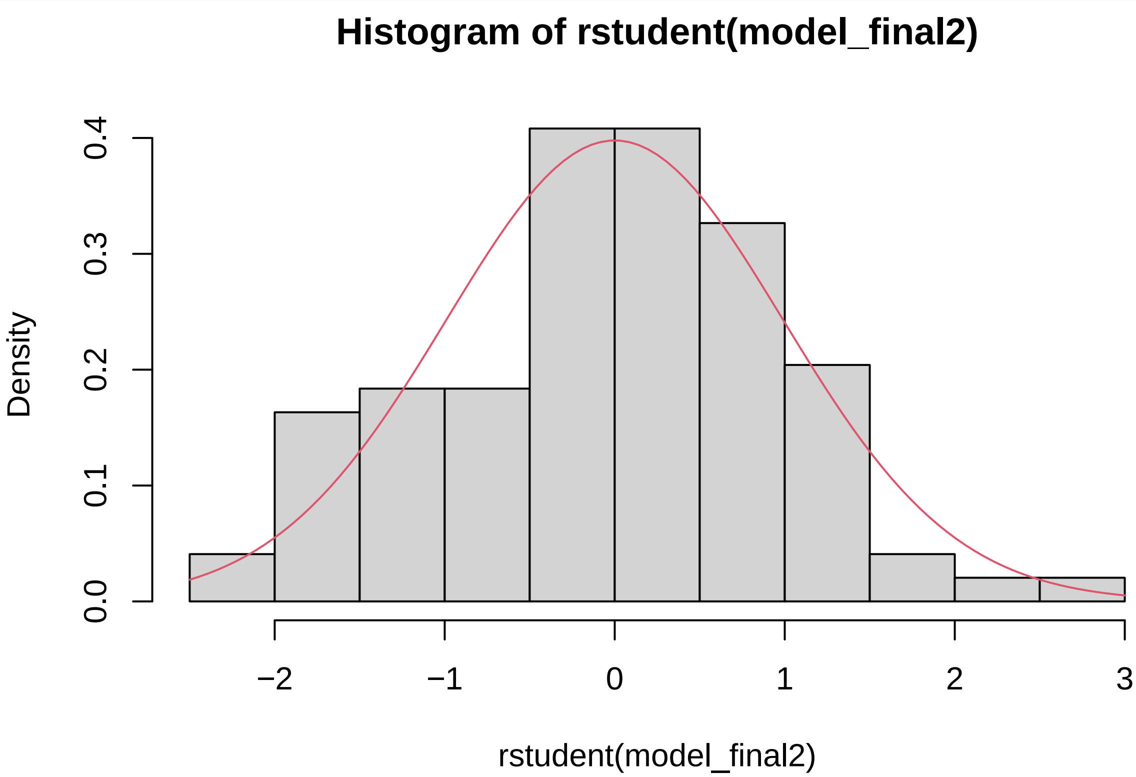
```
##
## Response: prestige
##           Sum Sq Df F value    Pr(>F)
## education 1655.5  1 32.8882 1.205e-07 ***
## income    1058.8  1 21.0339 1.405e-05 ***
## type       591.2  2  5.8721  0.003966 **
## Residuals 4681.3 93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_no_type2 <- lm(prestige~education + income, data = df2)
anova(model_no_type2, model_final2)  # Falla, type té NAs!!
```

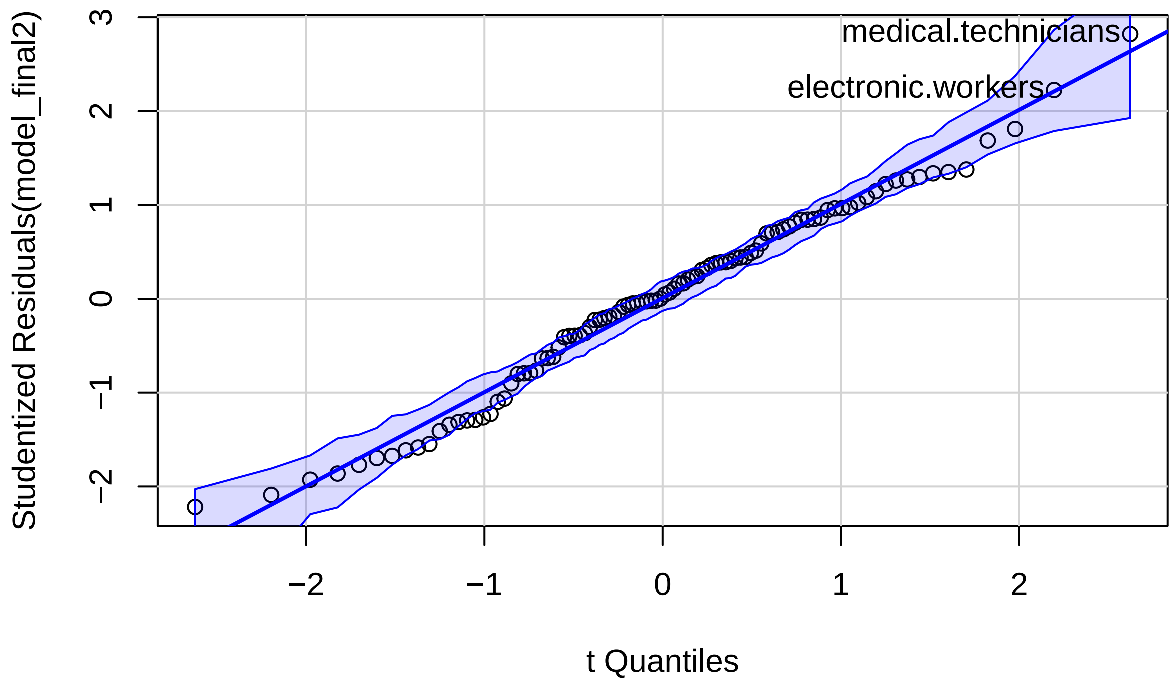
```
## Analysis of Variance Table
##
## Model 1: prestige ~ education + income
## Model 2: prestige ~ education + income + type
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     95 5272.4
## 2     93 4681.3  2    591.16 5.8721 0.003966 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Usual plots:

```r
# Histogram of studentized residuals
hist(rstudent(model_final2), freq=F)
curve(dt(x, model_final2$df), col=2, add=T)
```



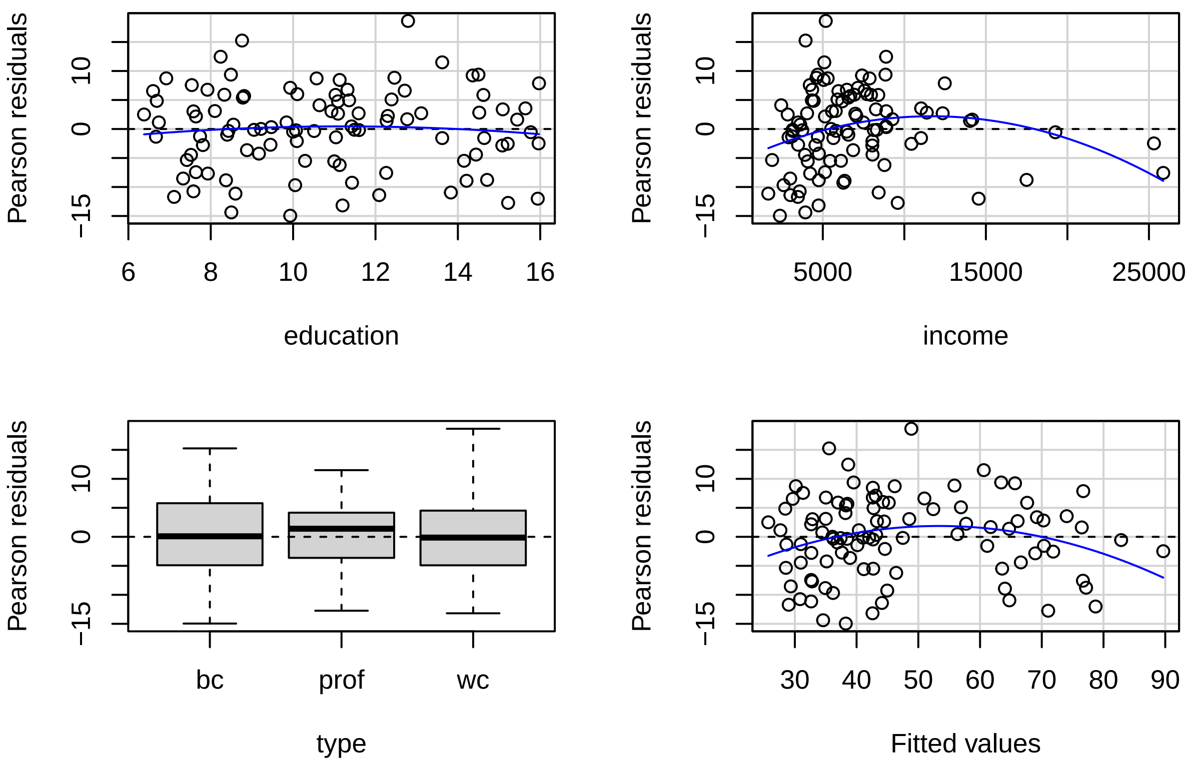

```r
## QQ Plot for normality
qqPlot(model_final2, simulate=T, labels=F)
```




```
## medical.technicians  electronic.workers
##                  31                  78
```

We have more functions to check linearity satisfaction and homoskedastic hypothesis. The horizontal band indicates them:

```r
residualPlots(model_final2)
```




```
##           Test stat Pr(>|Test stat|)
## education   -0.6836         0.495942
```

```
## income       -2.8865         0.004854 **
## type
## Tukey test   -2.6104         0.009043 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have as well a Homoskedastic Hypothesis Test - Breusch-Pagan test in package lmtest might be of interes:

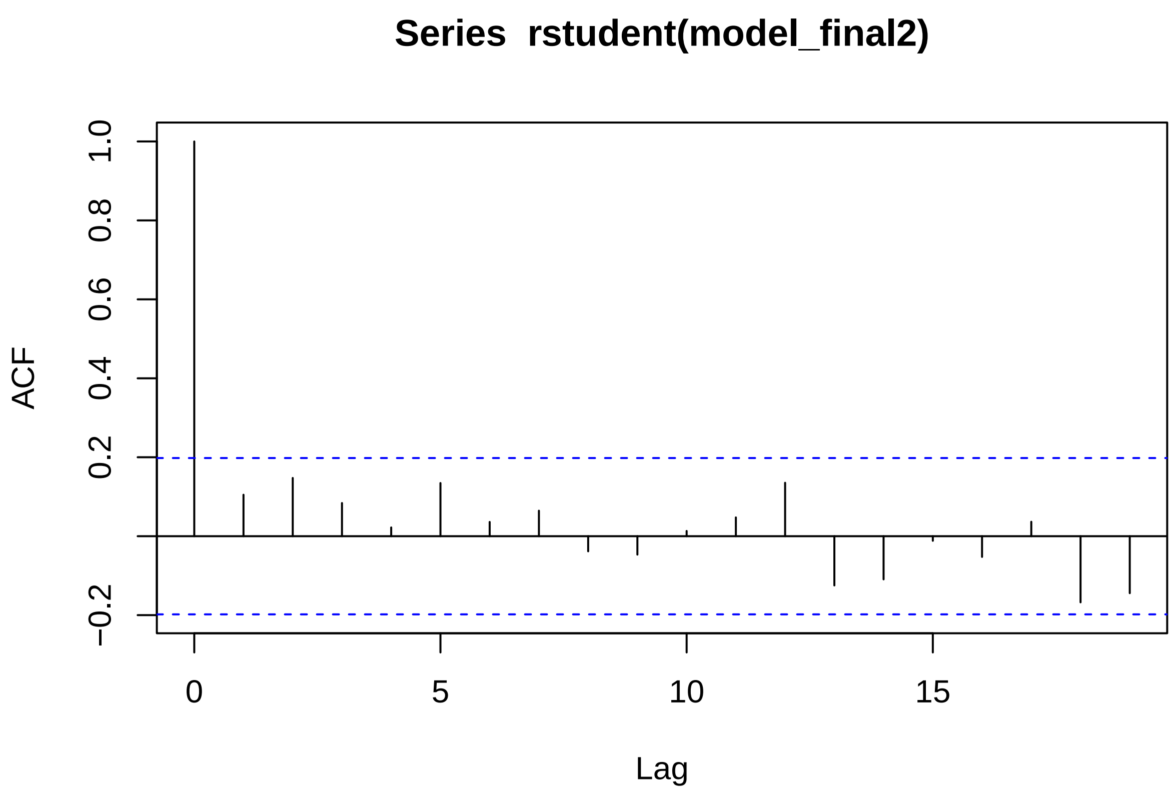
```r
# library(lmtest)
bptest(model_final2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_final2
## BP = 7.0719, df = 4, p-value = 0.1321
```

In this case, we can't reject homoskedasticity.

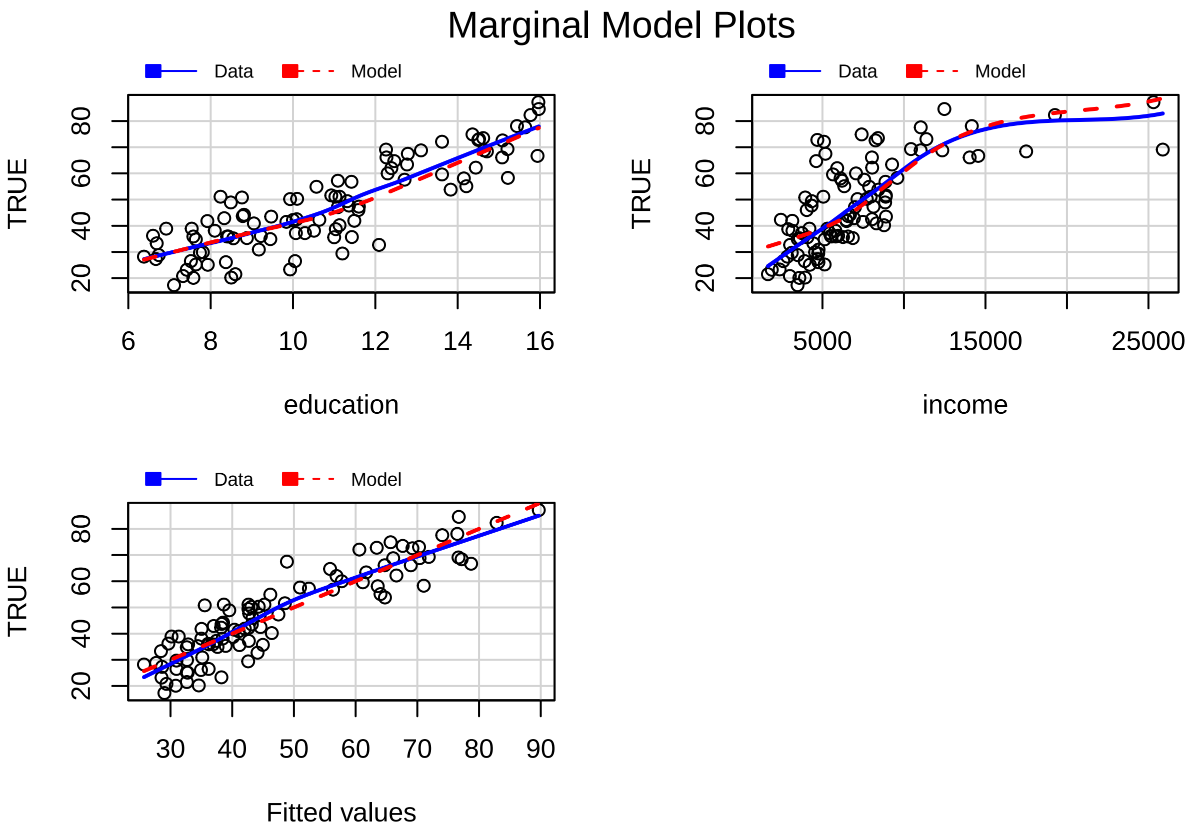
To test uncorrelation of the residuals (residual vs time/order or any omitted variable in the model suspected to affect hypothesis) we can use acf:

```r
acf(rstudent(model_final2))
```




```r
marginalModelPlots(model_final2)
```

```
## Warning in mmps(...): Interactions and/or factors skipped
```



Use `poly(varname, n)` to model linear and up to n-terms on varname regressor.

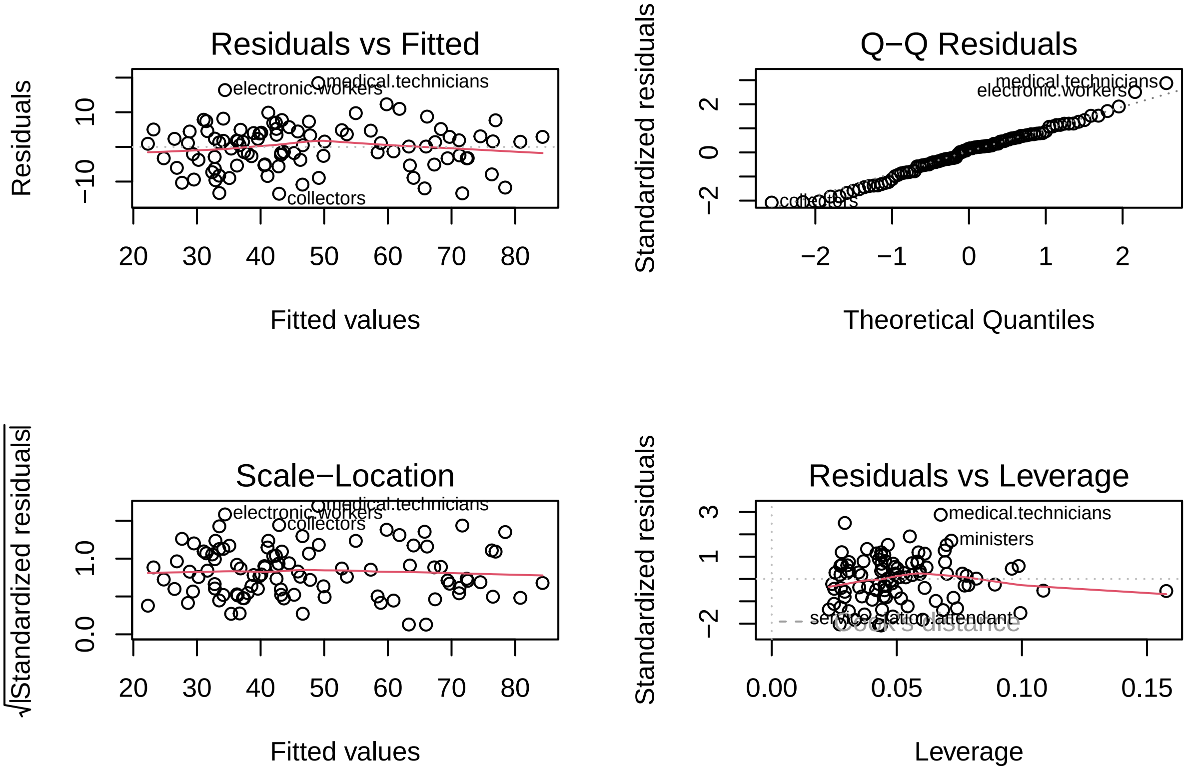
```
model_final22 <- lm(prestige ~ education + income + log(income) + type, data = df2)
summary(model_final22)
```

```
##
## Call:
## lm(formula = prestige ~ education + income + log(income) + type,
##     data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7732  -3.9665   0.8793   4.2276  18.2334
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.015e+02  2.742e+01  -3.701 0.000366 ***
## education    3.284e+00  6.090e-01   5.393 5.34e-07 ***
## income      -3.603e-04  4.217e-04  -0.854 0.395079
## log(income)  1.310e+01  3.503e+00   3.738 0.000322 ***
## typeprof     6.939e+00  3.630e+00   1.911 0.059074 .
## typewc      -1.408e+00  2.382e+00  -0.591 0.555952
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.646 on 92 degrees of freedom
## Multiple R-squared:  0.8566, Adjusted R-squared:  0.8488
## F-statistic: 109.9 on 5 and 92 DF,  p-value: < 2.2e-16
```

```r
model_final3 <- lm(prestige ~ education + log(income) + type, data = df2)
summary(model_final3)
```

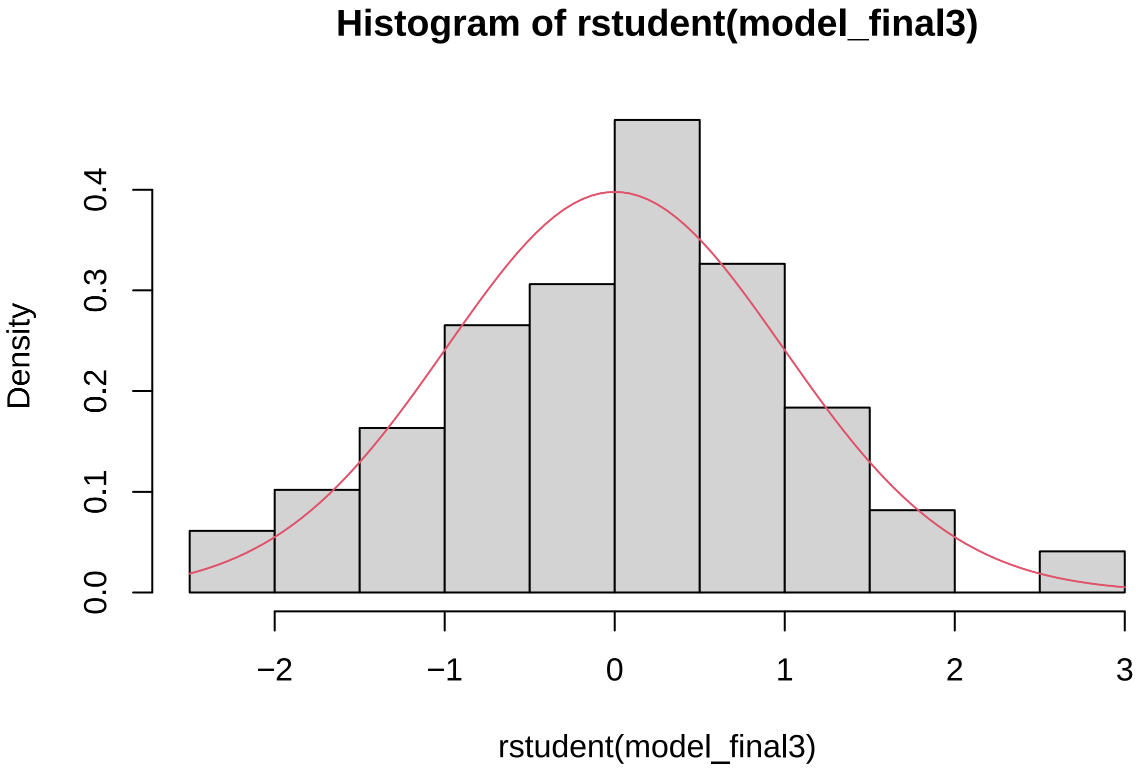
```
##
## Call:
## lm(formula = prestige ~ education + log(income) + type, data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.511  -3.746   1.011   4.356  18.438
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -81.2019    13.7431  -5.909 5.63e-08 ***
## education     3.2845     0.6081   5.401 5.06e-07 ***
## log(income)  10.4875     1.7167   6.109 2.31e-08 ***
## typeprof      6.7509     3.6185   1.866   0.0652 .
## typewc       -1.4394     2.3780  -0.605   0.5465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.637 on 93 degrees of freedom
## Multiple R-squared:  0.8555, Adjusted R-squared:  0.8493
## F-statistic: 137.6 on 4 and 93 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(model_final3)
```

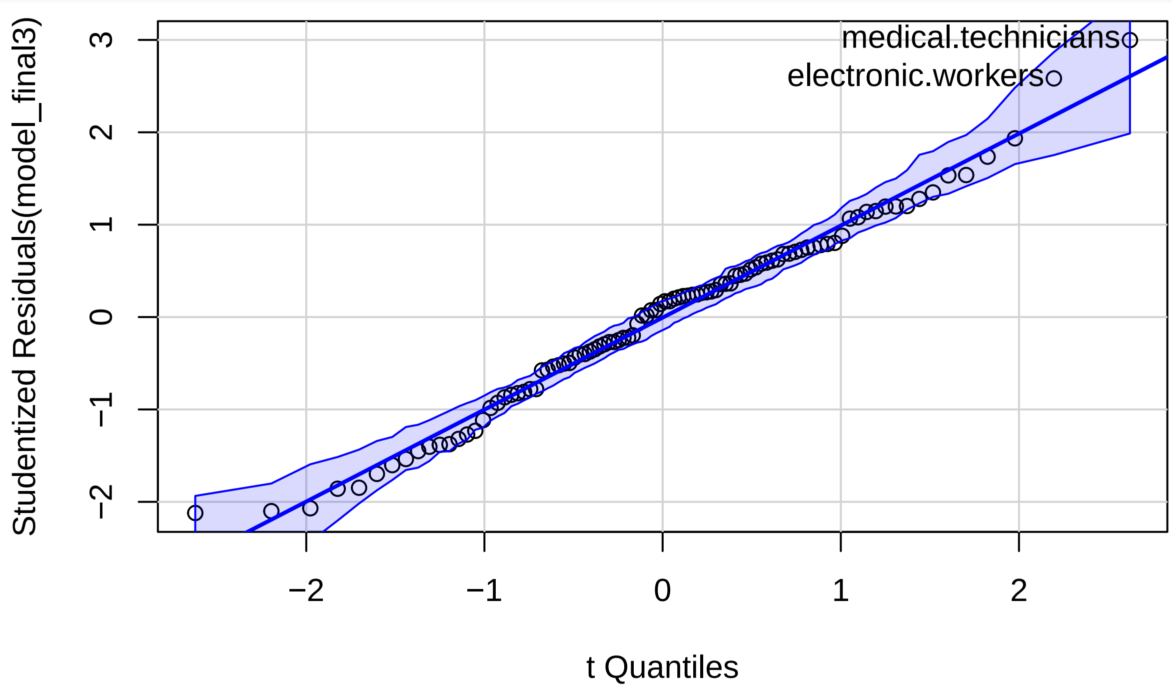



```r
par(mfrow=c(1,1))
```

```r
# Histogram of studentized residuals
hist(rstudent(model_final3), freq=F)
curve(dt(x, model_final3$df), col=2, add=T)
```



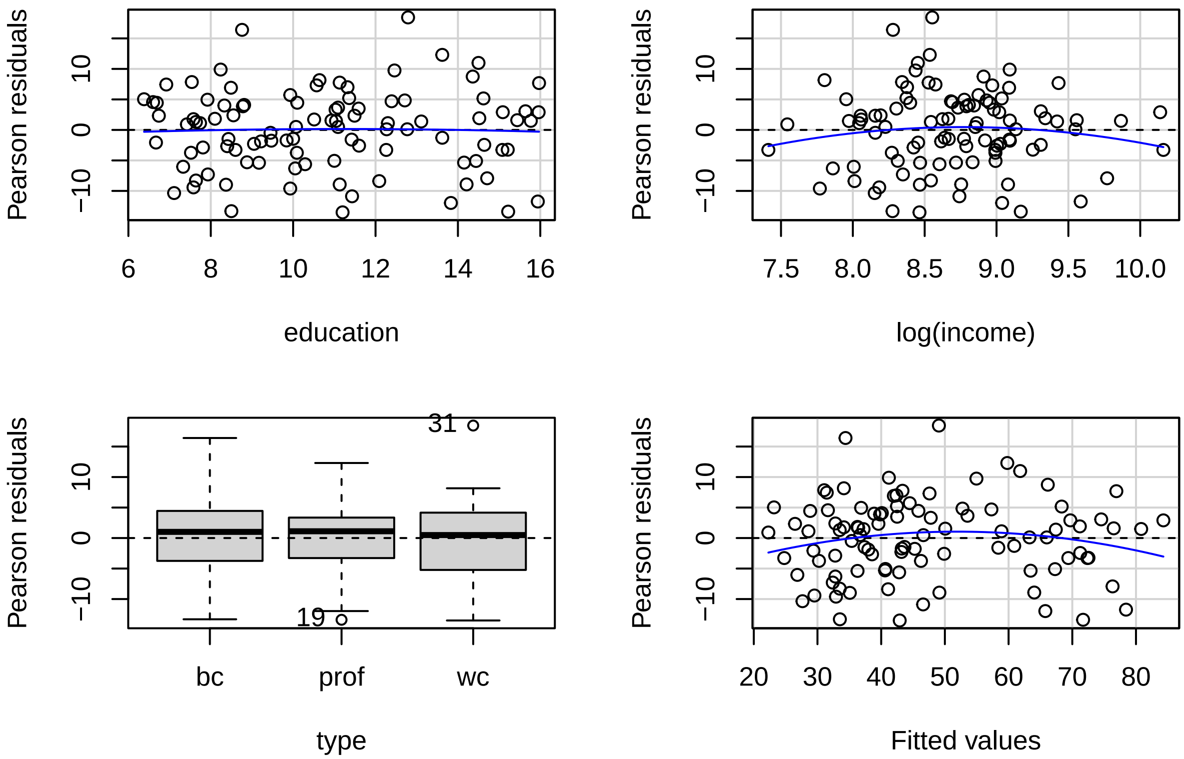

```r
## QQ Plot for normality
qqPlot(model_final3, simulate=T, labels=F)
```




```
## medical.technicians   electronic.workers
```

```
##                    31                   78
```

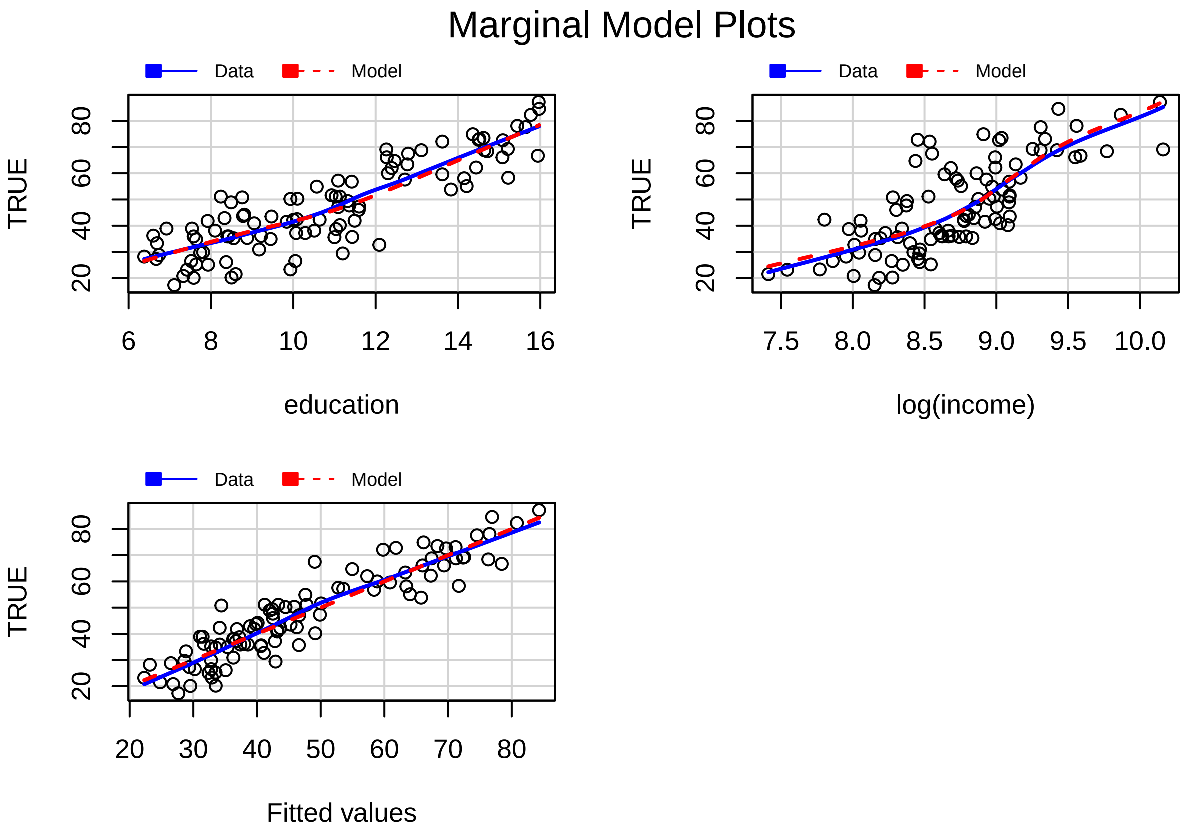
```
residualPlots(model_final3)
```




```
##               Test stat Pr(>|Test stat|)
## education       -0.2372           0.8130
## log(income)     -1.0444           0.2990
## type
## Tukey test      -1.4460           0.1482
```

```
marginalModelPlots(model_final3)
```

```
## Warning in mmps(...): Interactions and/or factors skipped
```



Box-Cox transformation on Y

The Box-Cox transformation of Y functions to normalize the error distribution, stabilize the error variance and straighten the relationship of Y to the Xs. Basic transformations are log(Y), 1/Y, sqrt(Y):

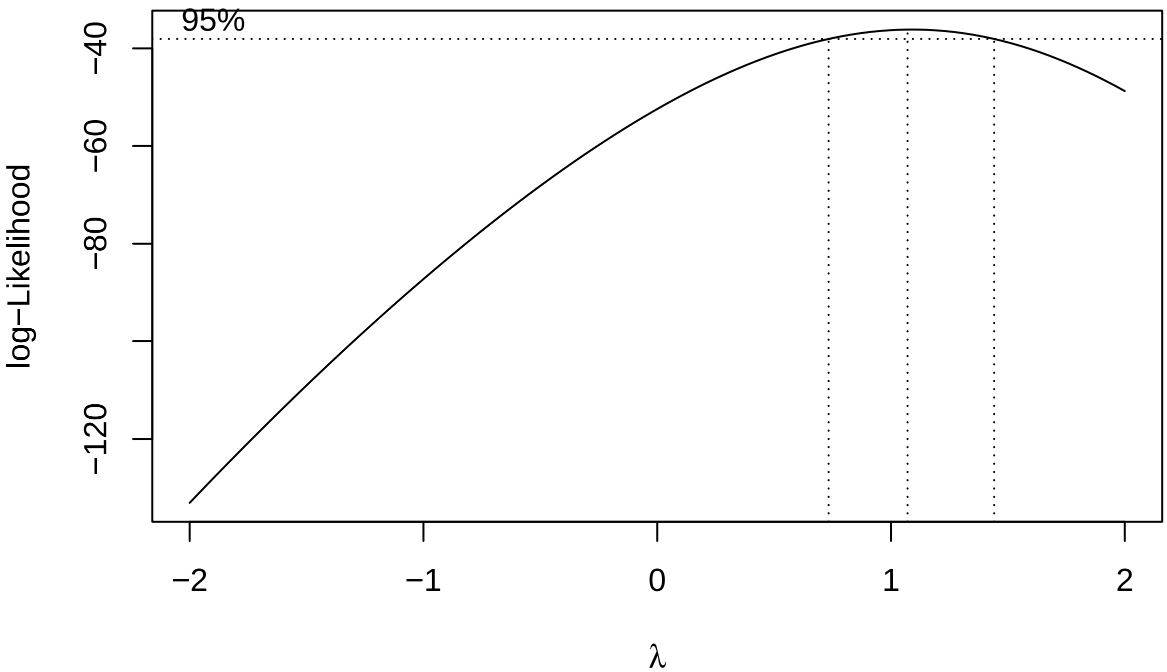
```
bcm <- lm(formula = prestige ~ boxCoxVariable(prestige) + log(income) + education + type, data = df2)
summary(bcm)
```

```
##
## Call:
## lm(formula = prestige ~ boxCoxVariable(prestige) + log(income) +
##     education + type, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0526  -4.0006   0.9314   4.2926  18.9225
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -86.6420    16.3860  -5.288 8.32e-07 ***
## boxCoxVariable(prestige)  -0.1446     0.2353  -0.615   0.5404
## log(income)               10.3361     1.7400   5.940 5.02e-08 ***
## education                  3.3651     0.6241   5.392 5.36e-07 ***
## typeprof                   6.9124     3.6402   1.899   0.0607 .
## typewc                    -1.8481     2.4769  -0.746   0.4575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

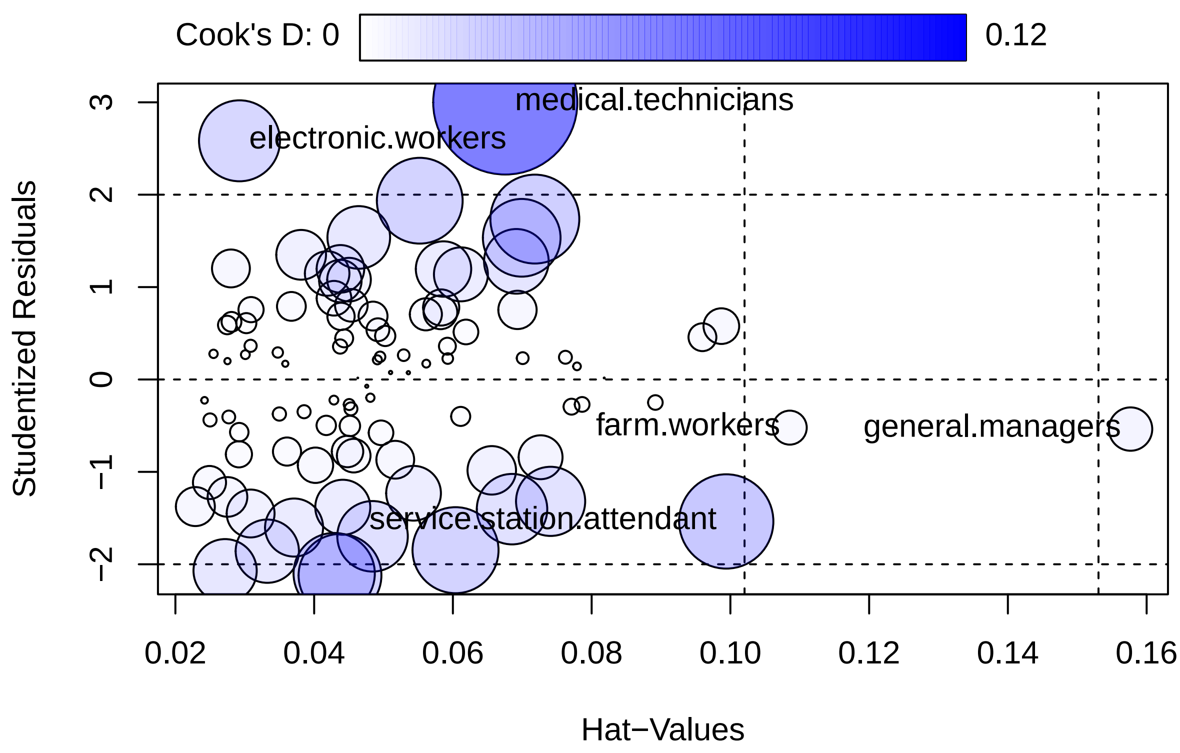
```
##
## Residual standard error: 6.659 on 92 degrees of freedom
## Multiple R-squared:  0.8561, Adjusted R-squared:  0.8483
## F-statistic: 109.5 on 5 and 92 DF,  p-value: < 2.2e-16
```

In this case we don't need any transformation.

```r
boxcox(prestige ~ log(income) + education + type, data = df2)
```

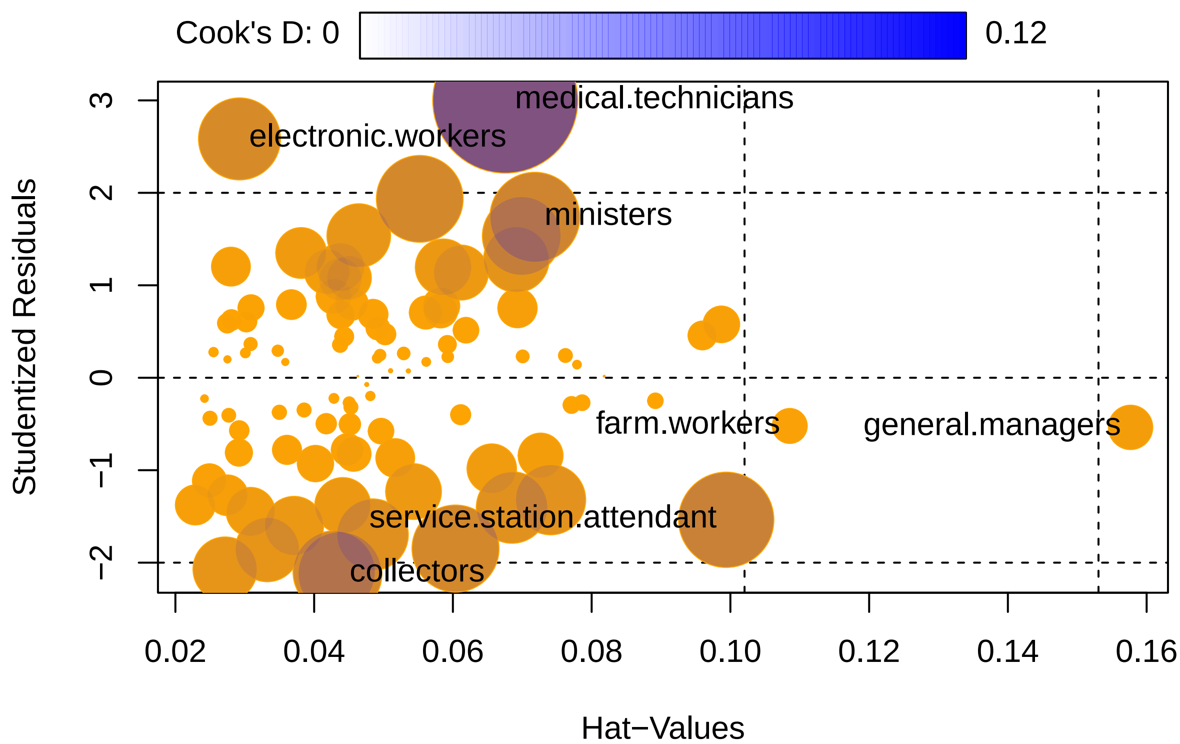



```r
influencePlot(model_final3)
```



```
##                             StudRes        Hat       CookD
## general.managers          -0.5367544 0.15768430 0.010870062
## medical.technicians        2.9980603 0.06754835 0.119925278
## service.station.attendant -1.5365460 0.09940089 0.051365372
## farm.workers              -0.5213202 0.10855861 0.006671519
## electronic.workers         2.5833033 0.02923909 0.037889157
```

```
influencePlot(model_final3,
              col="orange",
              pch=19,
              id=list(method="noteworthy",n=3))
```




```
##                             StudRes        Hat       CookD
## general.managers          -0.5367544 0.15768430 0.010870062
## ministers                  1.7361004 0.07181201 0.045649488
## medical.technicians        2.9980603 0.06754835 0.119925278
## collectors                -2.1205904 0.04372622 0.039634449
## service.station.attendant -1.5365460 0.09940089 0.051365372
## farm.workers              -0.5213202 0.10855861 0.006671519
## electronic.workers         2.5833033 0.02923909 0.037889157
```

Influential observations imply that the inclusion of the data in OLS modify the vector of estimated parameter and the fitted values.

DFBetas

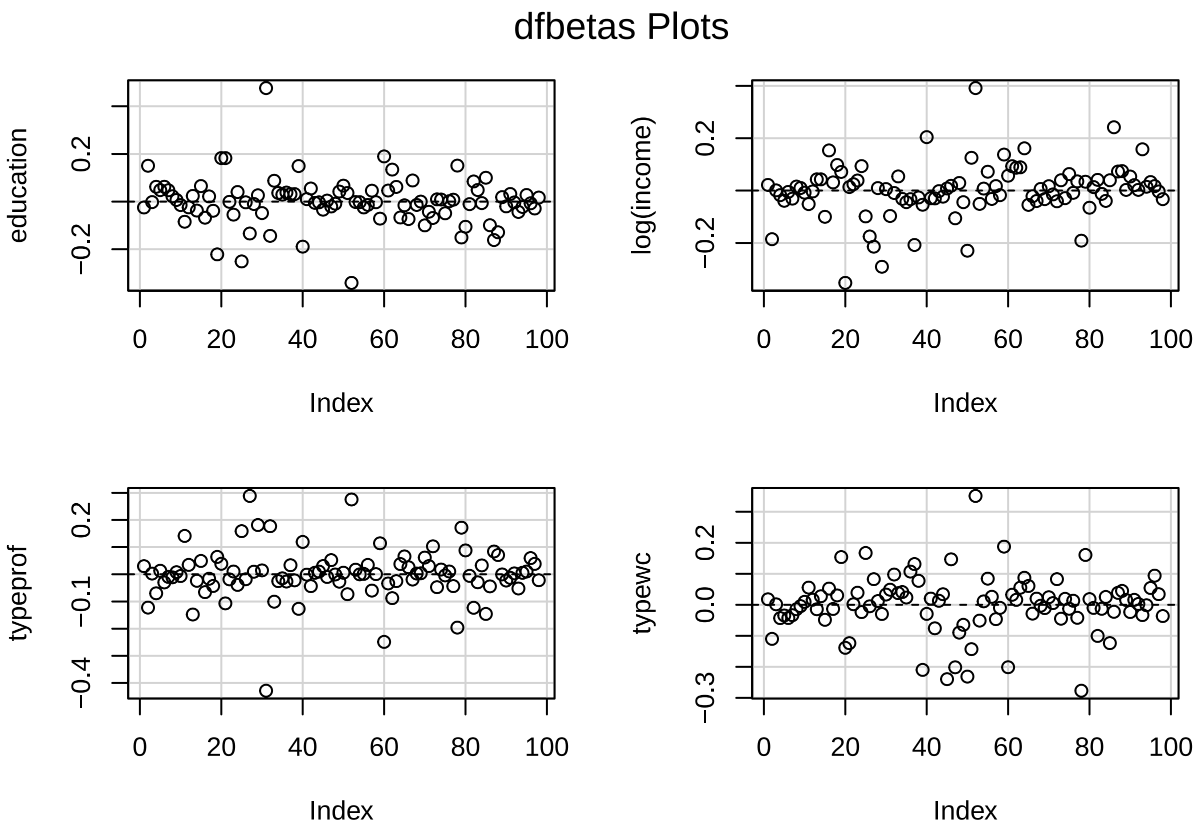
The most direct approach to assessing influence is to assess how the regression coefficients change if outliers are omitted from the model. We can use DFBetas_ij). Use `dfbetas(model)` in R.

```
head(dfbetas(model_final3))
```

```
##                         (Intercept)    education    log(income)      typeprof
```

```
## gov.administrators  -1.390206e-02 -0.024635407  0.0216433448  0.029981713
## general.managers     1.418582e-01  0.150406078 -0.1857267486 -0.122448927
## accountants          9.715269e-05 -0.002022608  0.0006121935  0.002769585
## purchasing.officers -4.351735e-03  0.062199298 -0.0175433476 -0.069930235
## chemists             2.359544e-02  0.048338908 -0.0390045426  0.013123389
## physicists          -1.770880e-02  0.062381406 -0.0050452428 -0.029866497
##                           typewc
## gov.administrators   0.017650150
## general.managers    -0.109932134
## accountants          0.001401876
## purchasing.officers -0.043058491
## chemists            -0.034492189
## physicists          -0.042675674
```

```r
dfbetasPlots(model_final3)
```



Cook's D

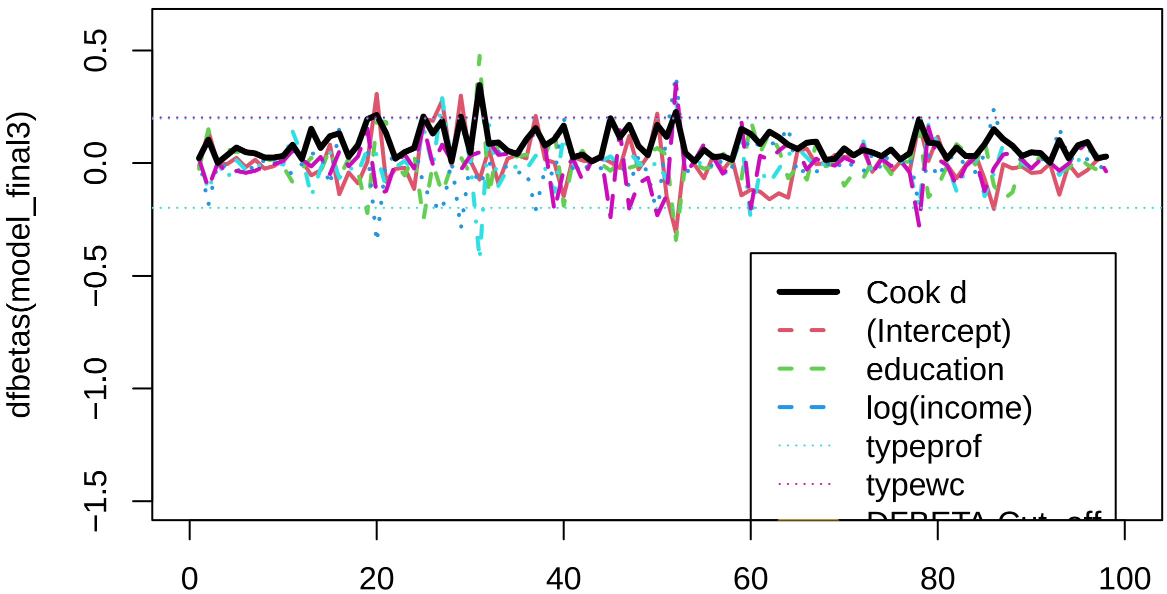
To overcome the problem of having a 2D object we have Cook's Dthat presents a single summary measure for each observation. Use `cooks.distance(model)` in R.

```r
head(cooks.distance(model_final3))
```

```
##   gov.administrators    general.managers         accountants purchasing.officers
##         4.722351e-04        1.087006e-02        2.733529e-06        1.230362e-03
##             chemists          physicists
##         4.790623e-03        2.368733e-03
```

We can plot both together and see the relationship:

```r
matplot(dfbetas(model_final3), type = "l",
        col=2:7, lwd=2, xlim = c(0, 100), ylim = c(-1.5, 0.6))
lines(sqrt(cooks.distance(model_final3)), col=1, lwd=3)
abline(h = 2/sqrt(dim(df)[1]), lty=3, lwd=1, col=5)
abline(h = -2/sqrt(dim(df)[1]), lty=3, lwd=1, col=5)
abline(h = sqrt(4/(dim(df)[1]-length(names(coef(model_final3))))),
       lty=3, lwd=1, col=6)
llegenda <- c("Cook d", names(coef(model_final3)), "DFBETA Cut-off", "Ch-H Cut-off")
# legend(locator(n=1), legend=llegenda,
#        col=1:length(llegenda), lty=c(1,2,2,2,3,3), lwd=c(3,2,2,2,1,1))
legend(x = 60, y = -0.4, legend=llegenda,
       col=1:length(llegenda), lty=c(1,2,2,2,3,3), lwd=c(3,2,2,2,1,1))
```



DFFits

One can argue that if the final objective is rather predictive than explicative, one can use the difference in the fitted values rather than in the beta parameters. DFFits are related to Cook's distance and combine studentized residuals and leverages. Use `dffits(model)` in R.

```r
head(dffits(model_final3))
```

```
##  gov.administrators    general.managers         accountants purchasing.officers
##         0.048341859        -0.232237528         0.003677053        -0.078037005
##            chemists          physicists
##         0.154456285         0.108372435
```

```r
# influence(m2)
```

```r
plot(dffits(model_final3), type="l", lwd=3)
pp = length(names(coef(model_final3)))
lines(sqrt(cooks.distance(model_final3)), col=3, lwd=2)
abline(h = 2*(sqrt(pp/(nrow(df)-pp))), lty=3, lwd=1, col=2)
abline(h = -2*(sqrt(pp/(nrow(df)-pp))),lty=3, lwd=1, col=2)
llegenda <- c("DFFITS", "DFFITS Cut-off", "Cooks D")
# legend(locator(n=1), legend = llegenda,
```

```r
#         col=1:3, lty=c(1,3,1), lwd=c(3,1,2))
legend(x = 60, y = -0.5, legend = llegenda,
       col=1:3, lty=c(1,3,1), lwd=c(3,1,2))
```