Diagnostics and partial residuals in a linear model

Brice Ozenne

September 21, 2020

Summary

In the document we provide a brief introduction to the linear model and its underlying assumptions. We then show how some of these hypotheses can be checked and what to do when there is evidence that one or several assumptions are not met. Finally we introduce the notion of partial residuals and explain how to compute and display them.

To be able to run the R code you will need to load the following packages:

```r
library(lava) ## install.packages("lava")
library(car) ## install.packages("car")
library(nlme) ## install.packages("nlme")
library(butils) ## devtools::install_github("bozenne/butils")
library(gof) ## devtools::install_github("kkholst/gof")
library(ggfortify) ## install.packages("ggfortify")
```

The first is hosted on CRAN and can be install directly. The next two are hosted on Gihub so you should first install the package devtools (using `install.packages`) and then you will be able to install these packages via the command `install_gitub` as shown above (after the `##`).

1 Data

We will use the dataset generated by the following commands:

```
set.seed(10)
m.lvm <- lvm(Y[100:sigma2]~beta*AgeC+BMI2+Gene)
categorical(m.lvm, K = 3) <- ~Gene
distribution(m.lvm, ~Age) <- uniform.lvm(20,50)
distribution(m.lvm, ~BMI) <- gaussian.lvm(mean = 24)
transform(m.lvm, AgeC~Age) <- function(x, ...){x-35}
transform(m.lvm, Id~Age) <- function(x, ...){1:NROW(x)}
transform(m.lvm, BMI2~BMI) <- function(x, ...){(x-24) + (x-24)^2}
latent(m.lvm) <- ~AgeC+BMI2
d <- rbind(cbind(lava::sim(n = 1e2, m.lvm, latent=FALSE, p = c(beta =
    1, sigma2 = 1)), Gender = "Male"),
           cbind(lava::sim(n = 1e2, m.lvm, latent=FALSE, p = c(beta =
    2, sigma2 = 2)), Gender = "Female")
           )

d$Gender <- as.factor(d$Gender)
d$Gene <- factor(d$Gene, labels = c("A","B","C"))
d$Y <- round(d$Y,1)
d$Age <- round(d$Age,1)
d$BMI <- round(d$BMI,1)
head(d)
```

```
      Y Gene  Age  BMI Id Gender
1 115.7    A 48.0 25.2  1   Male
2 108.7    B 42.4 24.3  2   Male
3 108.6    A 41.7 25.4  3   Male
4 104.4    C 36.4 24.9  4   Male
5  93.3    A 27.9 22.9  5   Male
6  97.3    C 29.2 24.5  6   Male
```

2 Short introduction to the linear model

Imagine we would like to model the age effect on the outcome, but accounting for a possible gender and gene effect. In **R** we would use the `lm` function:

```
e.lm <- lm(Y~Gender+Age+Gene+BMI, data = d)
e.lm
```

```

Call:
lm(formula = Y ~ Gender + Age + Gene + BMI, data = d)

Coefficients:
 (Intercept)  GenderFemale           Age         GeneB         GeneC           BMI
     21.3988        0.9778        1.5326        1.3783        2.6682        1.0351
```

Denote for the $i-th$ patient its outcome value by Y_i (can be any real number), its gender value by $Gender_i$ (can be "Male" or "Female"), its gene value by $Gene_i$ (can be "A", "B", or "C"), and its BMI value by BMI_i. Mathematically, this linear model can be written:

$$\begin{aligned} Y_i =& \alpha + \beta_{Gender} * \mathbb{1}_{Gender_i="Female"} + \beta_{Age} * Age_i + \beta_{GeneB} * \mathbb{1}_{Gene_i="B"} + \beta_{GeneC} * \mathbb{1}_{Gene_i="C"} \\ &+ \beta_{BMI} * BMI_i + \varepsilon_i \end{aligned}$$

where $\boldsymbol{\beta} = (\alpha, \beta_{Gender}, \beta_{Age}, \beta_{GeneB}, \beta_{GeneC}, \beta_{BMI})$ is the vector of model parameters. Their value is shown just above (e.g. $\alpha = 21.3988$). Here $\mathbb{1}_.$ denotes the indicator function taking value 1 if "." is true and 0 otherwise. ε_i is the residual error, i.e. the difference between the observed value and the fitted value. Consider for instance the first individual:

```
d[1,]
```

```

      Y Gene Age  BMI Id Gender
1 115.7    A  48 25.2  1   Male
```

its observed value is 115.7 and we can computed its fitted value as:

$$\begin{aligned} \hat{Y}_1 &= \alpha + \beta_{Gender} * 0 + \beta_{Age} 48 + \beta_{GeneB} * 0 + \beta_{GeneC} * 0 \\ &= 21.3988 + 0.9778 * 0 + 1.5326 * 48 + 1.3783 * 0 + 2.6682 * 0 + 1.0351 * 25.2 \\ &= 121.048 \end{aligned}$$

This can also be obtained using the `fitted` method in **R** (the discrepancy comes from rounding the coefficient values at the 4th digit):

```
fitted(e.lm)[1]
```

```
       1
121.0474
```

Often, for conciseness, this linear model can be abbreviated as:

$$Y_i = X_i\boldsymbol{\beta} + \varepsilon_i$$

where $X_i = (1, \mathbb{1}_{Gender_i="Female"}, Age_i, \mathbb{1}_{Gene_i="B"}, \mathbb{1}_{Gene_i="C"})$ and $X_i\boldsymbol{\beta}$ is the matrix product between the row vector X_i and the column vector $\boldsymbol{\beta}$. More generally, i.e. at the population level instead of the individual level, we also write $Y = X\boldsymbol{\beta} + \varepsilon$ to describe the relationship between the random variables Y, X, ε.

3 Assumptions

A linear model $Y = X\beta + \varepsilon$ is a model studying the effects (β) of covariates (X) on the expected value of the outcome Y. Maximum likelihood (ML) estimation leads to unbiased estimates of β if the following assumptions are satisfied:

- **(A0)**: no unobserved confounders.
- **(A1)**: $\mathbb{E}\left[Y_i|X\right] = X_i\beta$ correct specification of the functional form of the covariates.
- **(A2)**: identically distributed and **(A3)** independent residuals.
  Under the normality assumption, it simplifies to **(A2)** homoschedasticity $\mathbb{V}ar\left[Y_i|X\right] = \sigma^2$ and **(A3)** uncorrelatedness $\forall i \neq j$, $\mathbb{C}ov\left[Y_i, Y_j|X\right] = 0$.

While not needed per se, the assumption of:

- **(A4)**: normally distributed residuals is often mentioned since (i) normality of the estimates holds exactly in finite samples (instead of asymptotically) i.e. p-values/CIs are reliable even in small samples, (ii) it ensures that MLE is the best estimation procedure, (iii) checking **(A2)** and **(A3)** is simplified.

Additional assumptions are typically necessary to ensure reliable and interpretable estimates:

- **(A4-bis)**: approximately symmetric and unimodal - otherwise modeling the expected value (aka the mean value) may not be very relevant.
- **(A5)**: absence of outliers - otherwise the estimates may be very sensitive to the value of a few observations which is often undesirable.

4 Checking assumptions made when fitting a linear model

4.1 (A0): no unobserved confounders

(A0) is in general impossible to check.

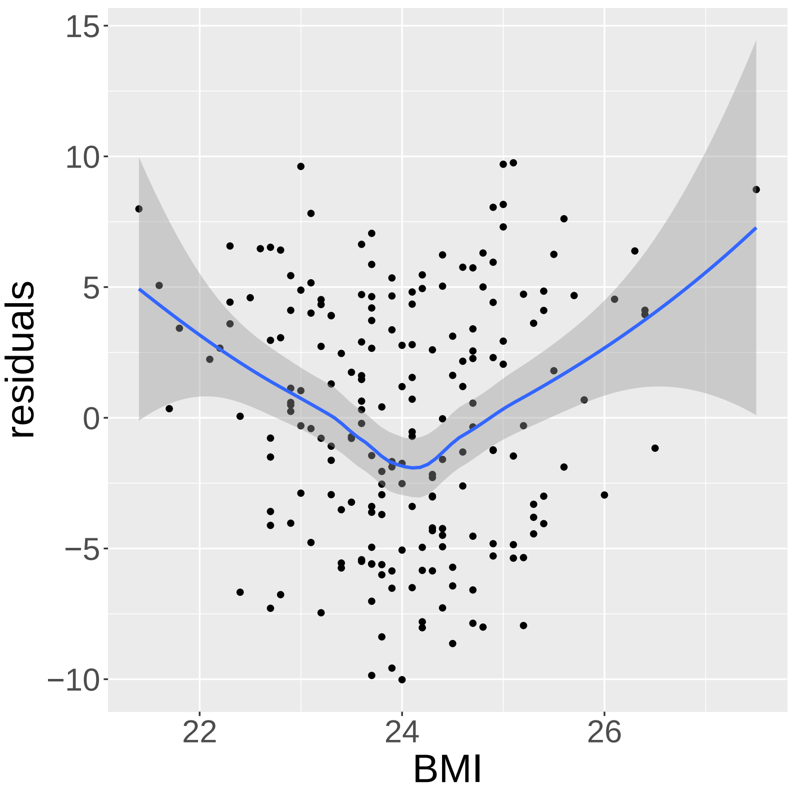
4.2 (A1): correct specification of the functional

(A1) can be (artificially) decomposed into two part:

- in absence of interaction, **is the effect of the continuous variables correctly modeled?** Typically it is modeled as a linear effect and the question is is there a non-linear effect. We can look at the plot of the covariate vs. the residuals and search for any trend:

```
gg <- ggplot(d, aes(x = BMI, y = residuals(e.lm)))
gg <- gg + geom_point() + geom_smooth() + ylab("residuals")
gg
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
    figures/A1-BMI.pdf")
```

```
‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



(similar plots can be automatically generated using the `crPlots` or `ceresPlots` function from the car package). A p-value for testing the correct specification of the functional form for the covariate can be obtained using the `cumres` function from the gof package:

```
cumres(e.lm, variable = "BMI")
```

```

Kolmogorov-Smirnov-test: p-value=0.002
Cramer von Mises-test: p-value=0
Based on 1000 realizations. Cumulated residuals ordered by BMI-variable.
---
```

Remedies: if a trend is found, a possible remedy is to use splines to model the non-linear relationship, e.g.

```
e.gam <- mgcv::gam(Y ~ Gender + Age + Gene + s(BMI), data = d)
```

In this simple example, it looks like a quadratic function of BMI would be enough:

```
e.lm.1 <- lm(Y ~ Gender + Age + Gene + BMI + I(BMI^2), data = d)
cumres(e.lm.1, variable = "BMI")
```

```

Kolmogorov-Smirnov-test: p-value=0.2
Cramer von Mises-test: p-value=0.164
Based on 1000 realizations. Cumulated residuals ordered by BMI-variable.
---
```

Note that this type of test is not appropriate to detect missing interaction:

```
cumres(e.lm.1, variable = "Age")
```

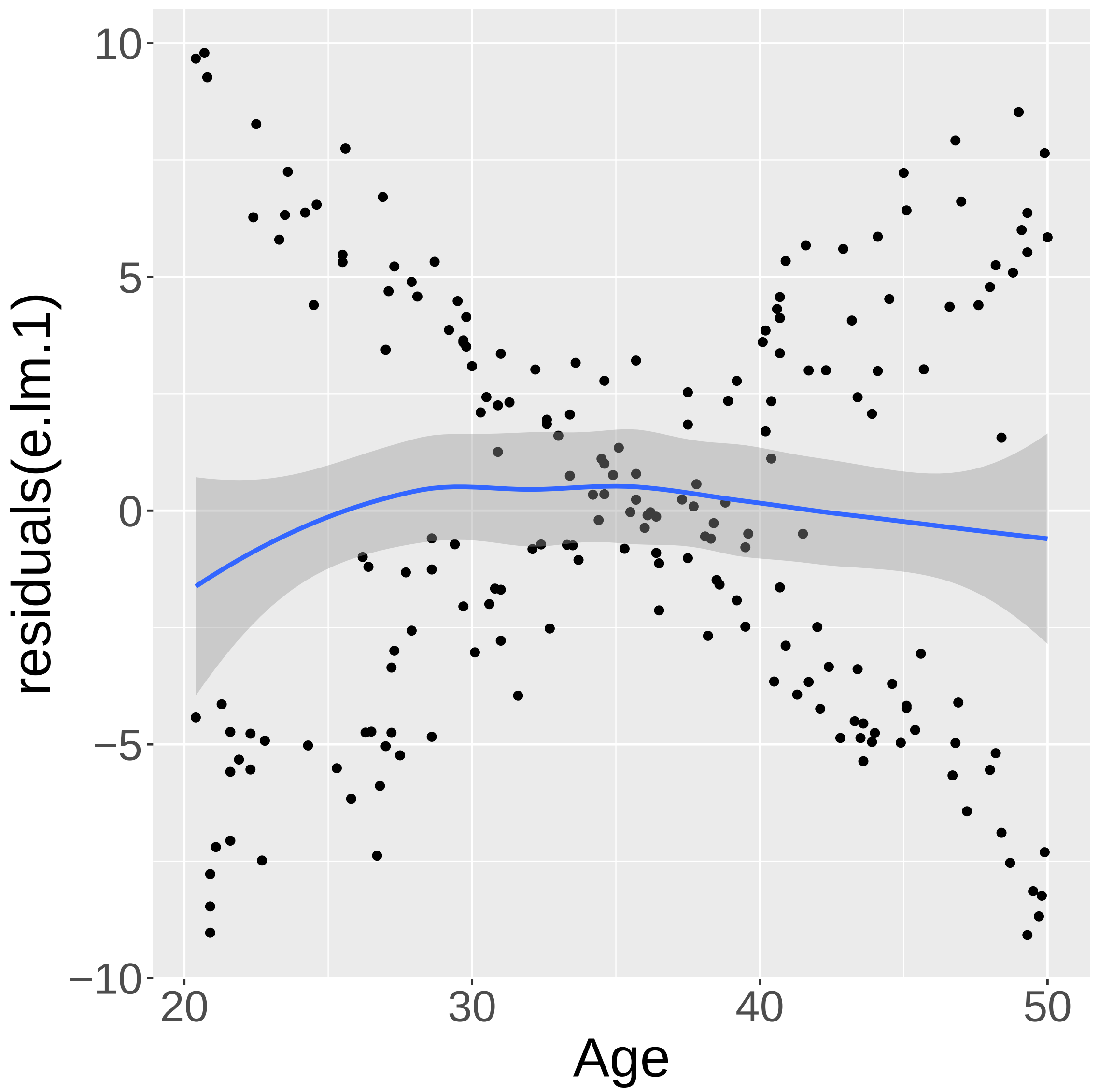
```

Kolmogorov-Smirnov-test: p-value=0.274
Cramer von Mises-test: p-value=0.332
Based on 1000 realizations. Cumulated residuals ordered by Age-variable.
---
```

while the display of the residuals can be informative

```
gg <- ggplot(d, aes(x = Age, y = residuals(e.lm.1))) + geom_point() +
    geom_smooth()
gg
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
    figures/A1-Age.pdf")
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



- **checking for interactions** is hard because the number of possible interactions grows quickly with the number of covariates. A typical test would be to compare a model with interactions to a model without interactions:

```
e.lm.2 <- update(e.lm, Y ∼ Gender*Age + Gene + BMI + I(BMI^2))
anova(e.lm.1, e.lm.2)
```

```
Analysis of Variance Table

Model 1: Y ~ Gender + Age + Gene + BMI + I(BMI^2)
Model 2: Y ~ Gender + Age + Gene + BMI + I(BMI^2) + Gender:Age
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
```

```
1    193 4037.2
2    192  318.7  1    3718.5 2240.4 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that in that case a test on the cumulative residuals process would not detect any issue:

```
cumres(e.lm.1, variable = "predicted")
```

```
Kolmogorov-Smirnov-test: p-value=0.46
Cramer von Mises-test: p-value=0.642
Based on 1000 realizations. Cumulated residuals ordered by predicted-variable.
---
```

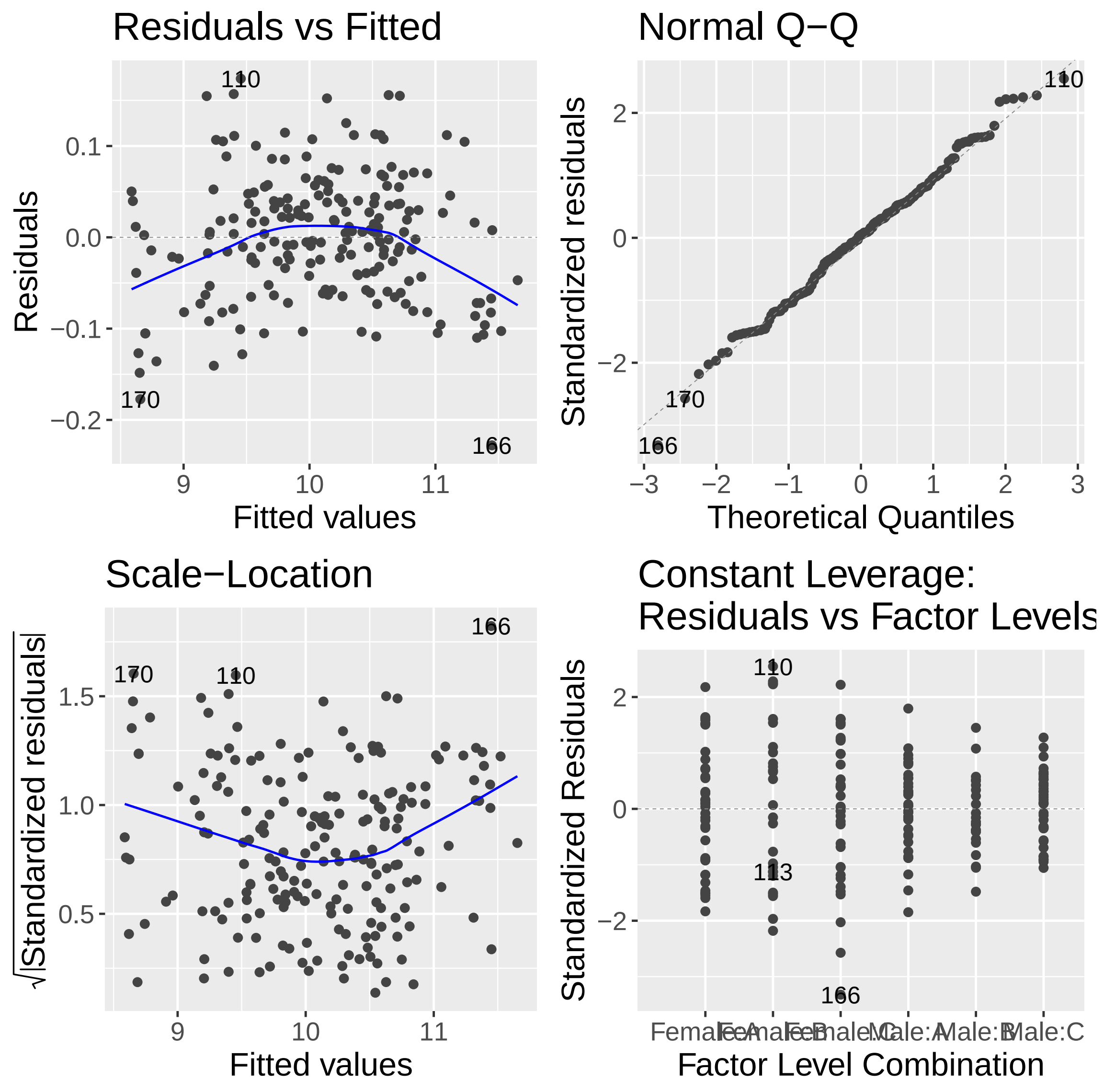
Remedies: this is a harder situation. When only few interactions are considered, a possible strategy would be to include all of them and perform backward selection. Otherwise adding all possible interactions and use a group-lasso penalty, or use more flexible but less interpretable models (e.g. random forest).

- A last possible issue arise when the **outcome variable is not studied on the right scale**. Consider the model using a square root transformation:

```
e.sqrt.lm <- lm(sqrt(Y) ~ Gender*Age + Gene + BMI + I(BMI^2), data = d)
```

Diagnostic plots indicates lack of fit (first line, first plot) and heteroschedasticity (second line first plot):

```
autoplot(e.sqrt.lm)
## ggsave(autoplot(e.sqrt.lm) + theme(text = element_text(size=15)),
   filename = "./figures/A1-scale.pdf")
```



We can use cumres and see that the link function seems inappropriate:

```
cumres(e.sqrt.lm, variable = "predicted")
```

```
Kolmogorov-Smirnov-test: p-value=0.001
Cramer von Mises-test: p-value=0
Based on 1000 realizations. Cumulated residuals ordered by predicted-variable.
---
```

In that case a box-cox transformation can be useful as it suggests to square the outcome:

```
M <- MASS::boxcox(e.sqrt.lm, lambda = seq(-1,4,by=0.1))
M$x[which.max(M$y)]
```

```
[1] 2.181818
```

Note that it seems to sometimes also suggest weird transformations:

```
M <- MASS::boxcox(lm(log(Y) ~ Gender*Age + Gene + BMI + I(BMI^2), data
    = d), lambda = seq(-10,10,by=0.1))
M$x[which.max(M$y)]
```

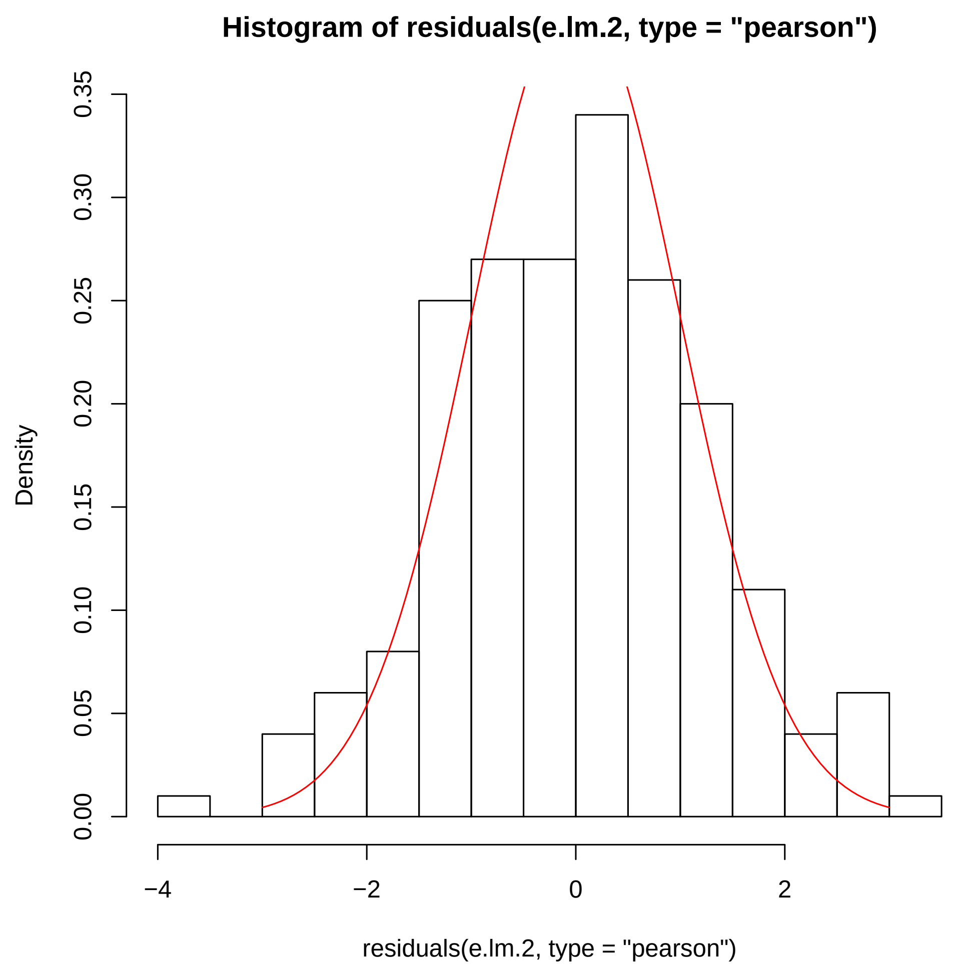
```
[1] 6
```

(the results should be 0)

4.3 (A4): normal distribution

(A4) can be tested using an histogram of the standardized residuals:

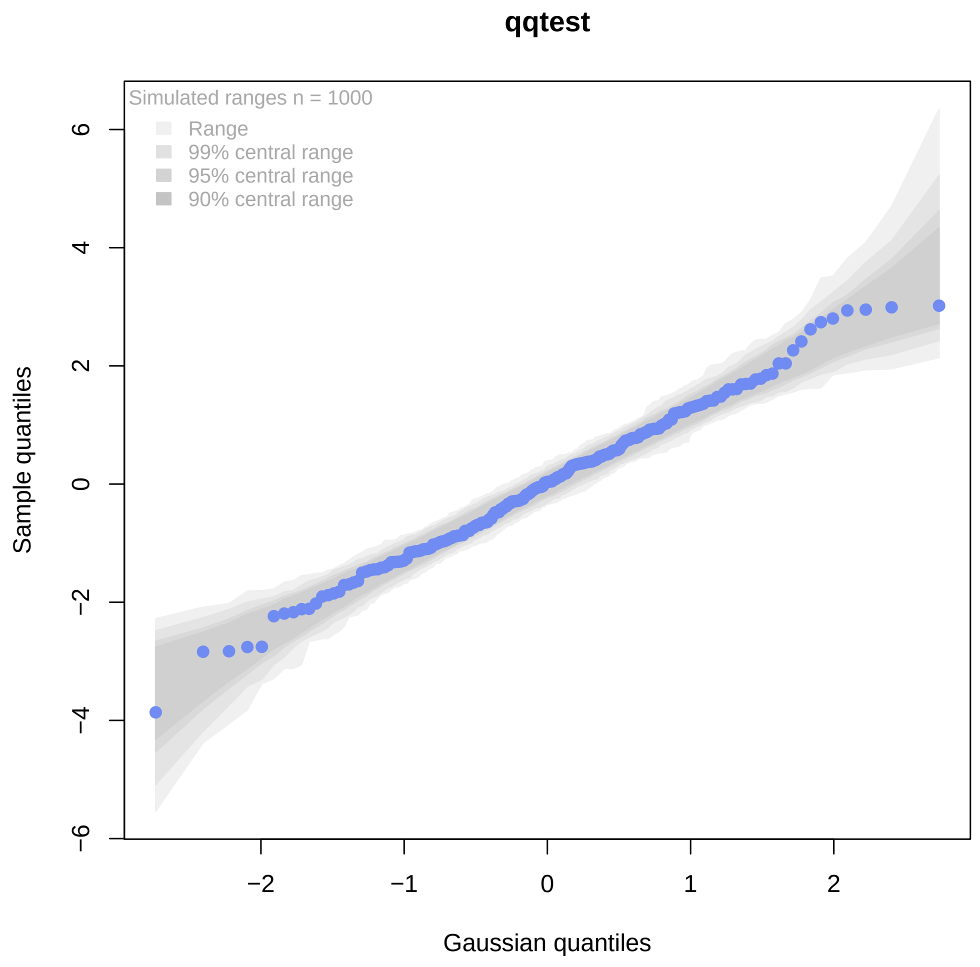
```
hist(residuals(e.lm.2, type = "pearson"), freq = FALSE, breaks = 10)
curve(dnorm,-3,3,add =TRUE,col = "red")
```



where the histogram should be close to the shape of the standard normal distribution (red curve). We could reject **(A4)** but accept **(A4-bis)** in the case where the distribution has heavy tails but is still unimodal and symmetric. While intuitive, this method is sensitive to the discretization of the residuals values (argument break) and a qq-plot is often preferred:

```
qqtest::qqtest(residuals(e.lm.2, type = "pearson"))
```

```
00:00:00 left
```



Here the points should follow a straight line and be within the shaded area. We could reject **(A4)** but accept **(A4-bis)** in the case where deviation to the straight line mostly arise in the tails. Statistical test (like a shapiro test) are not recommended since they do not enable us to know whether we reject **(A4)** or **(A4bis)**.

Remedies: when **(A4)** is rejected but not **(A4-bis)**, the main concern is about the validity of the traditional asymptotic results. This is not critical in a linear

regression where our variance estimator is consistent and the central limit theorem ensures asymptotic normality: instead of having exact p-values/CI they are only asymptotically valid. If the sample size is not too small they will hold; otherwise permutation test are a good alternative. In more complex models, robust standard errors or non-parametric bootstrap can be used for large enough samples to obtain p-values/CI robust to deviation to the normal distribution.
A more serious problem arises when **(A4-bis)** is rejected. In that case one should consider whether the expected outcome is really relevant. Alternative approaches include transformation of the outcome or use of alternative regression models (quantile regression, probability index models, finite mixture models).

Note 1: the `type` argument indicates the type of residuals we want to extract. Raw residuals are $\hat{\varepsilon} = Y - \hat{Y}$, i.e. the observed minus the fitted values. In models more complex than a univariate linear regression, the raw residuals may not be iid. This makes it difficult to assess the validity of the assumptions. In such cases we display instead diagnostics for normalized residuals that, if the assumptions of the model are correct, should follow a standard normal distribution.

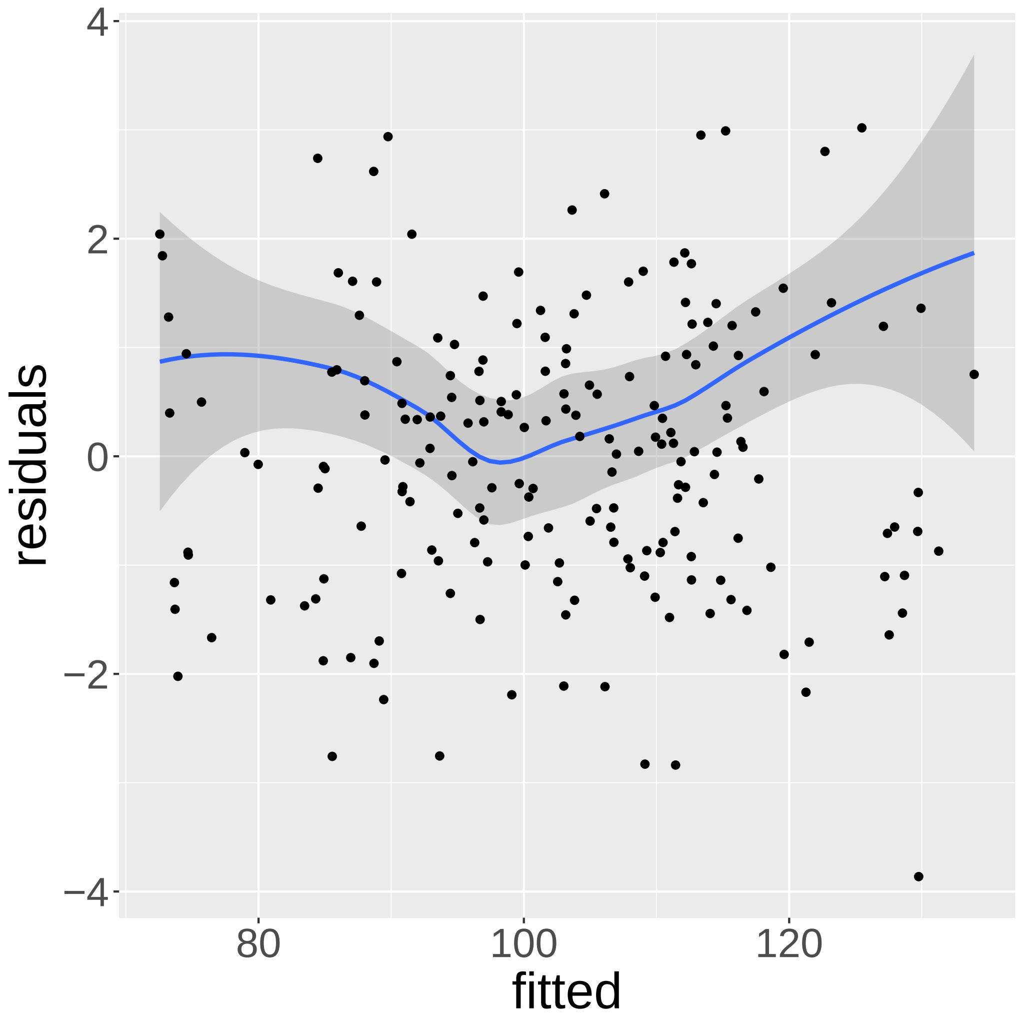
Note 2: an alternative to the `qqtest` function is the `qqPlot` function from the car package.

4.4 (A2): Homeschedasticity

Homoschedasticity can be inspected by displaying the residuals along the fitted values:

```
d$residuals <- residuals(e.lm.2, type = "pearson")
d$fitted <- fitted(e.lm.2)
gg <- ggplot(d, aes(x = fitted)) + ylab("residuals")
gg <- gg + geom_smooth(aes(y = residuals^2-1))
gg <- gg + geom_point(aes(y = residuals))
gg
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
   figures/A2-smooth.pdf")
```

```
‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



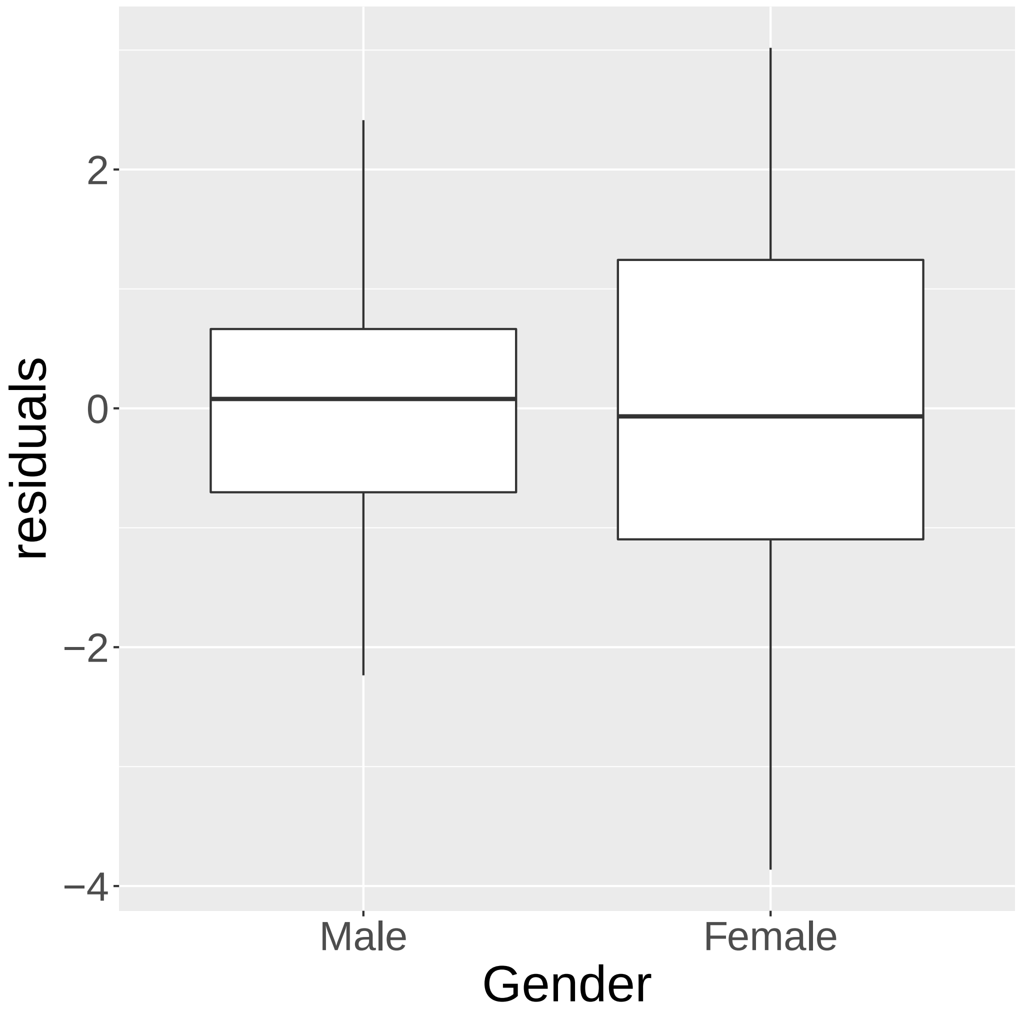
(see also the function `spreadLevelPlot` from the car package). It is also possible to have a global statistical test (Breusch-Pagan test):

```
ncvTest(e.lm.2)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.6009815, Df = 1, p = 0.4382
```

Alternatively one can look along a specific regressor:

```
gg <- ggplot(d, aes(x = Gender, y = residuals)) + ylab("residuals")
gg <- gg + geom_boxplot()
gg
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
   figures/A2-boxplot.pdf")
```



or investigate look how the squared residuals relates to the regressors:

```
summary(lm(residuals(e.lm.2)^2 ~ Gender + Age + Gene + BMI, data = d))$
   coef
```

```
                 Estimate Std. Error    t value     Pr(>|t|)
(Intercept)    6.45227607 3.65565988  1.7650100 7.913549e-02
GenderFemale   1.44221328 0.29848355  4.8318015 2.742530e-06
Age            0.01348642 0.01758548  0.7669068 4.440692e-01
GeneB          0.23754662 0.38188217  0.6220417 5.346448e-01
GeneC          0.03181216 0.34758915  0.0915223 9.271720e-01
BMI           -0.25529930 0.14902626 -1.7131161 8.828871e-02
```

Remedies: in presence of global heteroschadasticity (first graph), transforming the outcome can be a solution. Otherwise one should reflect about possible source of heteroschadasticity (e.g. correlated observations, mixture of populations) and model them. When the heteroschadasticity is related to a single variable, one can for instance use the `gls` function to model this variance:

```
e.gls <- gls(Y ~ Gender + Age + Gene + BMI + I(BMI^2) + Gender:Age,
             data = d,
             weight = varIdent(form=~1|Gender))
summary(e.gls$modelStruct)
```

```
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | Gender
 Parameter estimates:
    Male   Female
1.000000 1.650464
```

```
summary(
    lm(residuals(e.gls, type = "normalized")^2 ~ Gender + Age + Gene +
   BMI, data = d)
)$coef
```

```
                 Estimate Std. Error     t value  Pr(>|t|)
(Intercept)   2.845507264 2.08650094  1.36376994 0.1742203
GenderFemale  0.015137857 0.17036219  0.08885691 0.9292873
Age           0.005236824 0.01003707  0.52174821 0.6024408
GeneB         0.068737967 0.21796270  0.31536573 0.7528230
GeneC        -0.069620340 0.19838965 -0.35092728 0.7260237
BMI          -0.086243866 0.08505809 -1.01394080 0.3118739
```

4.5 (A5): Influential observations

The `influence` method can be used to output what is the impact of each observation on each estimated parameter:

```
if.lme <- influence(e.lm.2)
if.lme$coefficient[1:6,1:4]
```

```
  (Intercept) GenderFemale           Age         GeneB
1  0.03478943  0.018050611  6.597433e-04 -5.046560e-03
2 -0.05849442  0.001113552  4.583238e-05  9.488884e-04
3 -0.97841870 -0.018612231 -1.105912e-03  1.833443e-02
4  0.33314244  0.004040188  3.325704e-06 -2.090461e-05
5  0.34719463 -0.020540159 -4.416311e-04 -4.518752e-03
6 -0.33837887 -0.014621030 -3.528092e-04  1.299324e-04
```

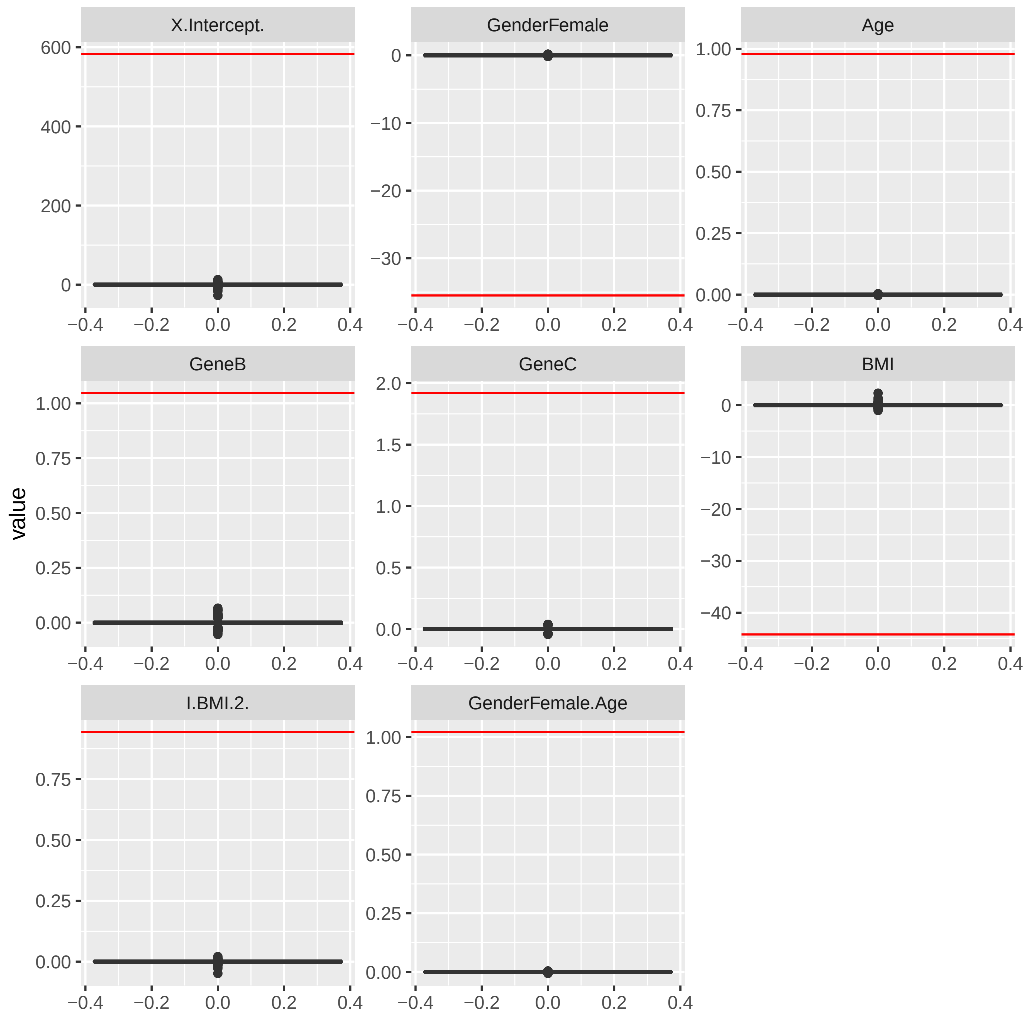
Here the value in the first line and third column indicates by how much is changed the Age effect when removing the first observation.

```
coef(update(e.lm.2,data=d[-1,]))-coef(e.lm.2)
```

```
  (Intercept)     GenderFemale              Age            GeneB            GeneC 
-0.0347894306    -0.0180506110    -0.0006597433     0.0050465601     0.0049566971 
          BMI        I(BMI^2) GenderFemale:Age 
 0.0063071805    -0.0001723731     0.0005968902 
```

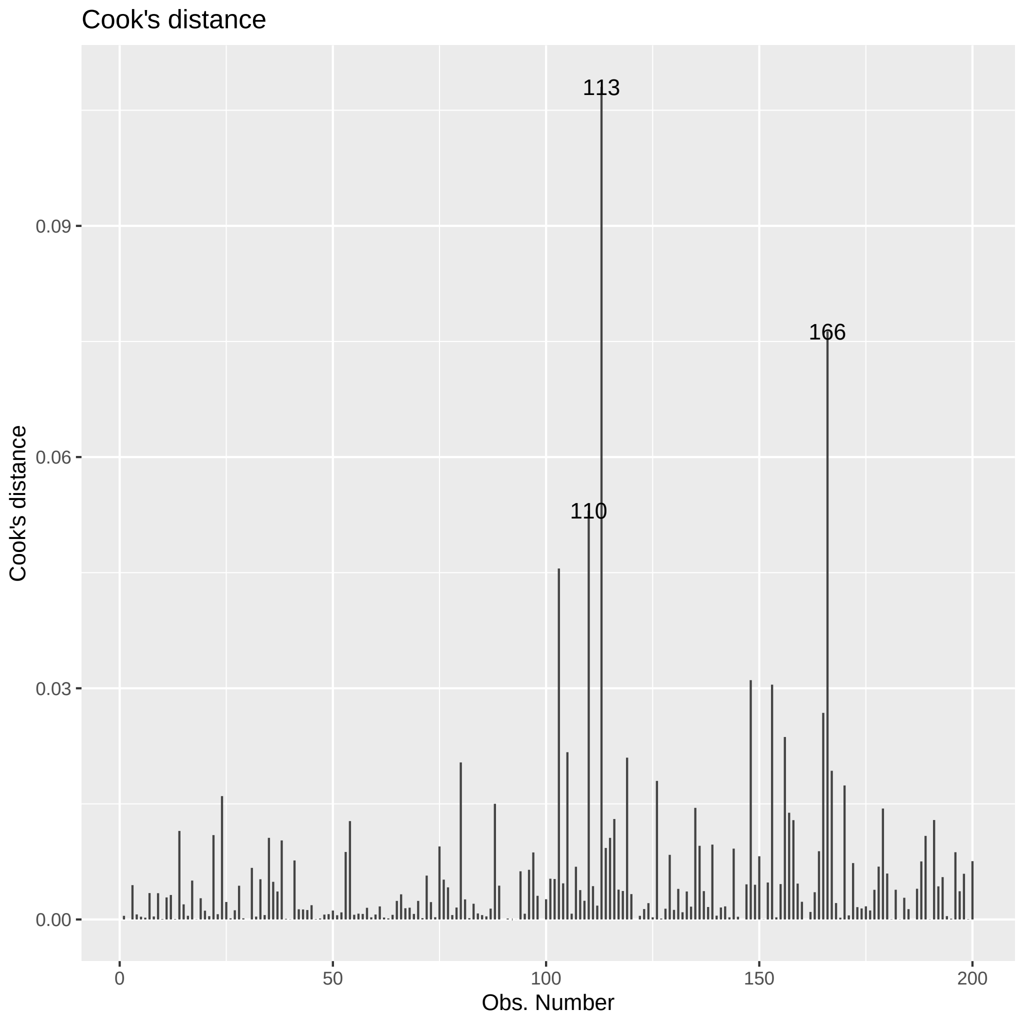
Large values (positive or negative) indicate influential observations. The following plot displaying in red the coefficient value and in black the influence of each individual can be useful:

```
dfW1.gg <- data.frame(id = "true", as.data.frame(t(coef(e.lm.2))))
dfW2.gg <- data.frame(id = as.character(1:NROW(d)), if.lme$coefficient)
dfL1.gg <- reshape2::melt(dfW1.gg, id.vars = "id")
dfL2.gg <- reshape2::melt(dfW2.gg, id.vars = "id")
gg.inf <-  ggplot() + facet_wrap(~variable, scales = "free")
gg.inf <- gg.inf + geom_boxplot(data = dfL2.gg, aes(y = value))
gg.inf <- gg.inf + geom_hline(data = dfL1.gg, aes(yintercept = value),
    color = "red")
gg.inf
## ggsave(gg.inf, filename = "./figures/A5-boxplot.pdf")
```



When the aim is to perform prediction, global influence metrics such as Cook's distance can be useful:

```
autoplot(e.lm.2, which = 4)
## ggsave(autoplot(e.lm.2, which = 4), filename = "./figures/A5-cook.
   pdf")
```



4.6 Others [no recommanded unless specific reasons]

Some people recommand to check the correlation between the explanatory variables, with the argument that when very correlated it is difficult to disantangle effects and thus to interpret the regression coefficients. The VIF (variance inflation factor) is typically recommanded to check that with values higher than 5 considered as high:

```
car::vif(e.lm.2)
```

```
                  GVIF Df GVIF^(1/(2*Df))
Gender       18.757045  1        4.330940
Age           2.210228  1        1.486683
Gene          1.026260  2        1.006501
BMI        1031.164279  1       32.111747
I(BMI^2)   1031.061224  1       32.110142
Gender:Age   19.413821  1        4.406112
```

I personnally don't recommand this as an automatic check since in many settings co-linearity can be better assessed from the meaning of the variables than from a statistical test. It is also quite unclear to me why 5 is a good cut-off and we see in this example that we get values close to five (or higher) even though there is no issue.

5 Partial residuals

5.1 With respect to one variable

The partial residuals with respect to age are defined by removing the effect of all the covariates but age on the outcome:

$$\hat{\varepsilon}_i^{Age} = Y_i - (\alpha + \beta_{Gender}\mathbb{1}_{Gender_i="Female"} + \beta_{GeneB}\mathbb{1}_{Gene_i="B"} + \beta_{GeneC}\mathbb{1}_{Gene_i="C"} + \beta_{BMI}BMI_i)$$

So for instance for the first individual:

$$\begin{aligned}\hat{\varepsilon}_1^{Age} &= 115.7 - (21.3988 + 0.9778 * 0 + 1.3783 * 0 + 2.6682 * 0 + 1.0351 * 25.2)\\ &= 115.7 - 47.48 = 68.22\end{aligned}$$

At the dataset level, this type of partial residual is centered around the expected value of the covariate times its effect (here $0.9814 * 36.078 \approx 35$). These partial residuals can be computed using the `partialResidual` function from the butils package:

```
pRes.noI <- partialResiduals(e.lm, var = "Age", keep.intercept = FALSE)
head(pRes.noI)
```

```
       Y Gene  Age  BMI Id Gender     pFit ranef pResiduals
1: 115.7    A 48.0 25.2  1   Male 47.48357     0   68.21643
2: 108.7    B 42.4 24.3  2   Male 47.93024     0   60.76976
3: 108.6    A 41.7 25.4  3   Male 47.69059     0   60.90941
4: 104.4    C 36.4 24.9  4   Male 49.84120     0   54.55880
5:  93.3    A 27.9 22.9  5   Male 45.10282     0   48.19718
6:  97.3    C 29.2 24.5  6   Male 49.42716     0   47.87284
```

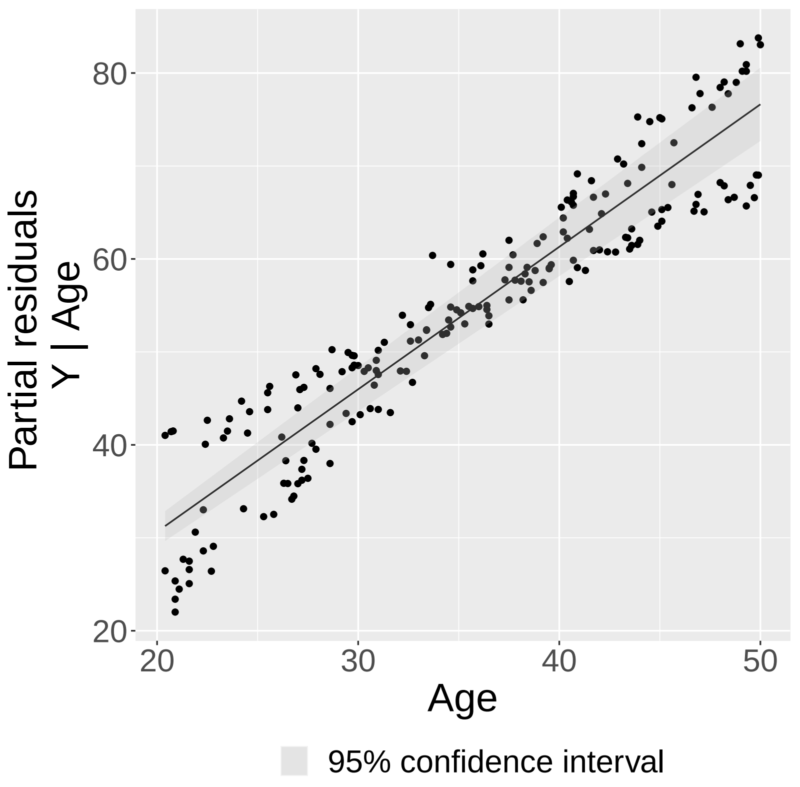
or manually:

```
keep.coef <- c("(Intercept)","GenderFemale","GeneB","GeneC","BMI")
d$Y[1] - model.matrix(e.lm)[1,keep.coef] %*% coef(e.lm)[keep.coef]
```

```
         [,1]
[1,] 68.21643
```

A graphical display can be obtained using the `autoplot` function (require the ggplot2 package):

```
gg <- autoplot(pRes.noI)
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
   figures/fig-butils-plotConf-noI.pdf")
```



- An alternative definition do not remove the intercept effect:

$$\hat{\varepsilon}_i^{Age,\alpha} = Y_i - (\beta_{Gender}\mathbb{1}_{Gender_i="Female"} + \beta_{GeneB}\mathbb{1}_{Gene_i="B"} + \beta_{GeneC}\mathbb{1}_{Gene_i="C"} + \beta_{BMI}BMI_i)$$

so now the residuals are centered around the intercept plus the expected value of age times the age effect (here approximately 0). As before the partial residuals can either be obtained via the `partialResiduals` function:

```
pRes.I <- partialResiduals(e.lm, var = "Age", keep.intercept = TRUE)
head(pRes.I)
```

```
       Y Gene  Age  BMI Id Gender     pFit ranef pResiduals
1: 115.7    A 48.0 25.2  1   Male 26.08478     0   89.61522
2: 108.7    B 42.4 24.3  2   Male 26.53145     0   82.16855
3: 108.6    A 41.7 25.4  3   Male 26.29181     0   82.30819
4: 104.4    C 36.4 24.9  4   Male 28.44242     0   75.95758
5:  93.3    A 27.9 22.9  5   Male 23.70403     0   69.59597
6:  97.3    C 29.2 24.5  6   Male 28.02837     0   69.27163
```

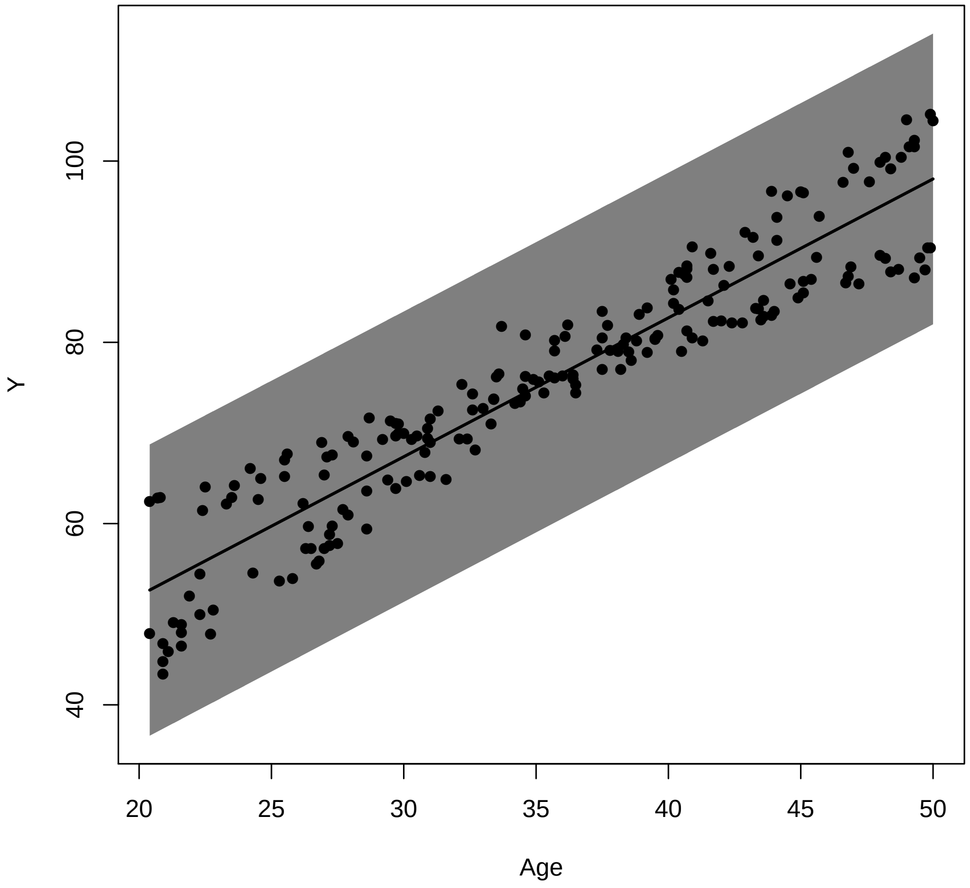
or manually:

```
keep.coef <- c("GenderFemale","GeneB","GeneC","BMI")
d$Y[1] - model.matrix(e.lm)[1,keep.coef] %*% coef(e.lm)[keep.coef]
```

```
        [,1]
[1,] 89.61522
```

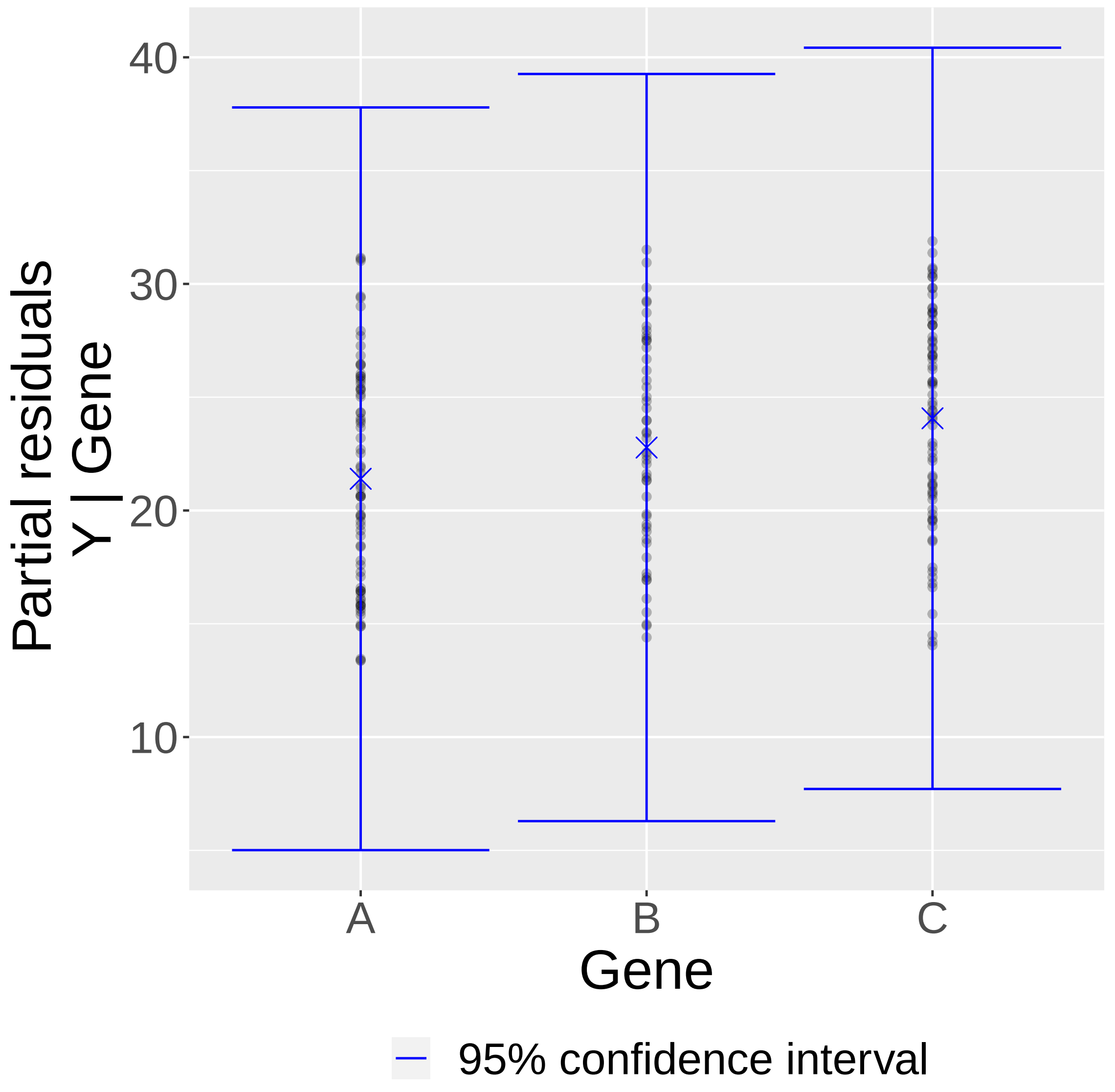
This corresponds to what the `plotConf` function is displaying (R package lava available on CRAN):

```
lava::plotConf(e.lm, var1 = "Age")
```



Note that it is also possible to display the partial residuals for a categorical variable:

```
gg <- autoplot(partialResiduals(e.lm, var = "Gene", keep.intercept =
   TRUE))
gg
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
   figures/fig-butils-plotConf-categorical.pdf")
```



Partial residuals
Y | Gene
40
30
20
10
A
B
C
Gene
95% confidence interval

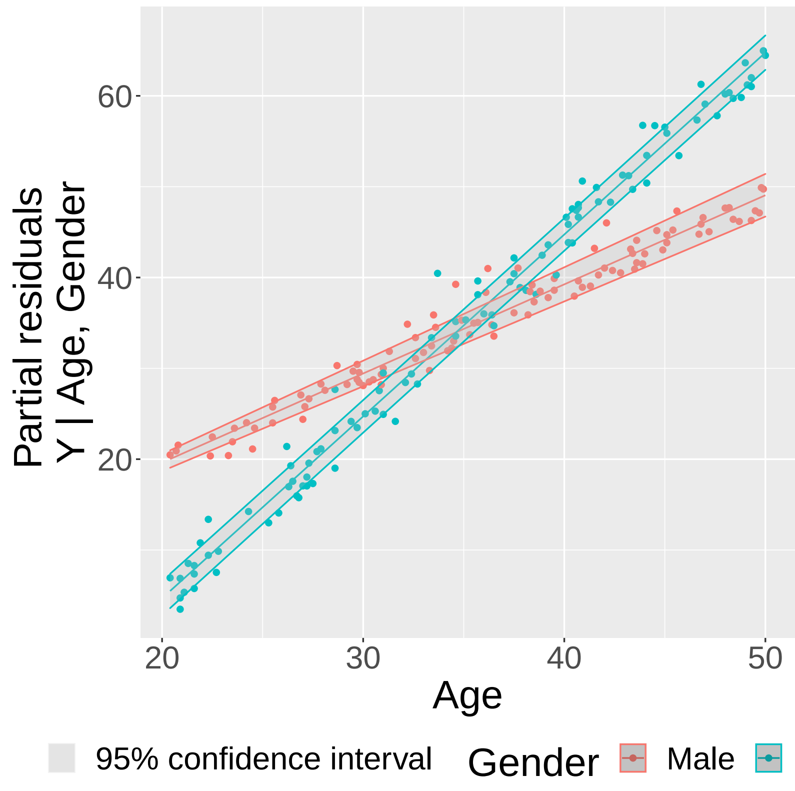
5.2 With respect to an interaction between two variables (one continuous, one categorical)

Consider now a model where the age effect can be different for males and females:

```
e.lmI <- lm(Y~Gender*Age+Gene+BMI, data = d)
```

The partial residuals can be defined in a similar way as before. Here the effect of Age and Gender (and their interaction) are not substracted from the outcome:

```
gg <- autoplot(partialResiduals(e.lmI, var = c("Age","Gender")))
## ggsave(gg + theme(text = element_text(size=25)), filename = "./
   figures/fig-butils-plotConf-interaction.pdf")
```



5.3 Customizing a partial residual plot

The autoplot function returns the ggplot object:

```
gg <- autoplot(partialResiduals(e.lm, var = "Gene", keep.intercept =
    TRUE))
class(gg)
```

```
[1] "gg"     "ggplot"
```

So it can be easily customized, e.g. the text can be made bigger by doing:

```
gg + theme(text = element_text(size=25))
```