Module 3: Recommended Exercises - Solution

TMA4268 Statistical Learning V2020

Martina Hall, Michail Spitieris, Stefanie Muff, Department of Mathematical Sciences, NTNU

January 23, 2020

Last changes: 16.01.2020

We strongly recommend you to work through the Section 3.6 in the course book (Lab on linear regression)

You need to install the following packages in R to run the code in this file.

```
install.packages("knitr")  #probably already installed
install.packages("rmarkdown")  #probably already installed
install.packages("GLMsData")  #data set for Problem 3
install.packages("ggplot2")  #plotting with ggplot
install.packages("nortest")  #test that data comes from normal distribution
install.packages("car")  #qqPlot function
install.packages("ISLR")  #data problem 1
install.packages("ggfortify")  #diagnostic plots for lm objects
```

Problem 1 (Book Ex. 9)

This question involves the use of multiple linear regression on the `Auto` data set from `ISLR` package (you may use `?Auto` to see a description of the data). First we exclude from our analysis the variable `name`.

```
library(ISLR)
Auto = subset(Auto, select = -name)
# Auto$origin = factor(Auto$origin)
summary(Auto)
```

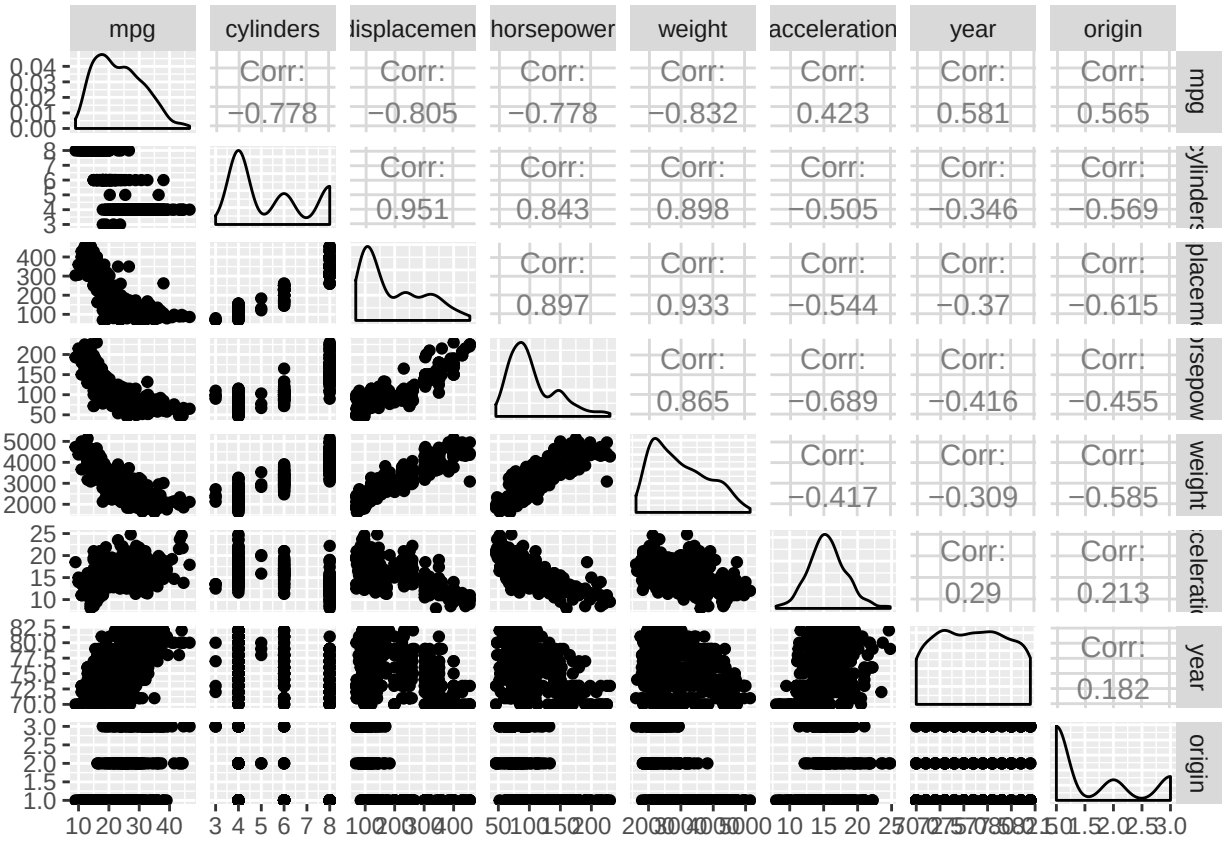
```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##      weight      acceleration        year           origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
```

a)

Use the function `ggpairs()` from `GGally` package to produce a scatterplot matrix which includes all of the variables in the data set.

Answer

```
library(GGally)
ggpairs(Auto)
```



b)

Compute the correlation matrix between the variables.

Answer

```
cor(Auto)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
```

```
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

c)

Use the lm() function to perform a multiple linear regression with `mpg` as the response and all other variables (except `name`) as the predictors. Use the `summary()` function to print the results. Comment on the output. For instance:

i. Is there a relationship between the predictors and the response?

ii. Which predictors show evidence that they are related to the response?

iii. What does the coefficient for the year variable suggest?

Answer

```
fit.lm = lm(mpg ~ ., data = Auto)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

- There is strong evidence for a relationship between the covariates `displacement`, `weight`, `year`, `origin` and the response `mgp`.
- The 0.75 coefficient suggests that a new model has higher mpg compared to an older one

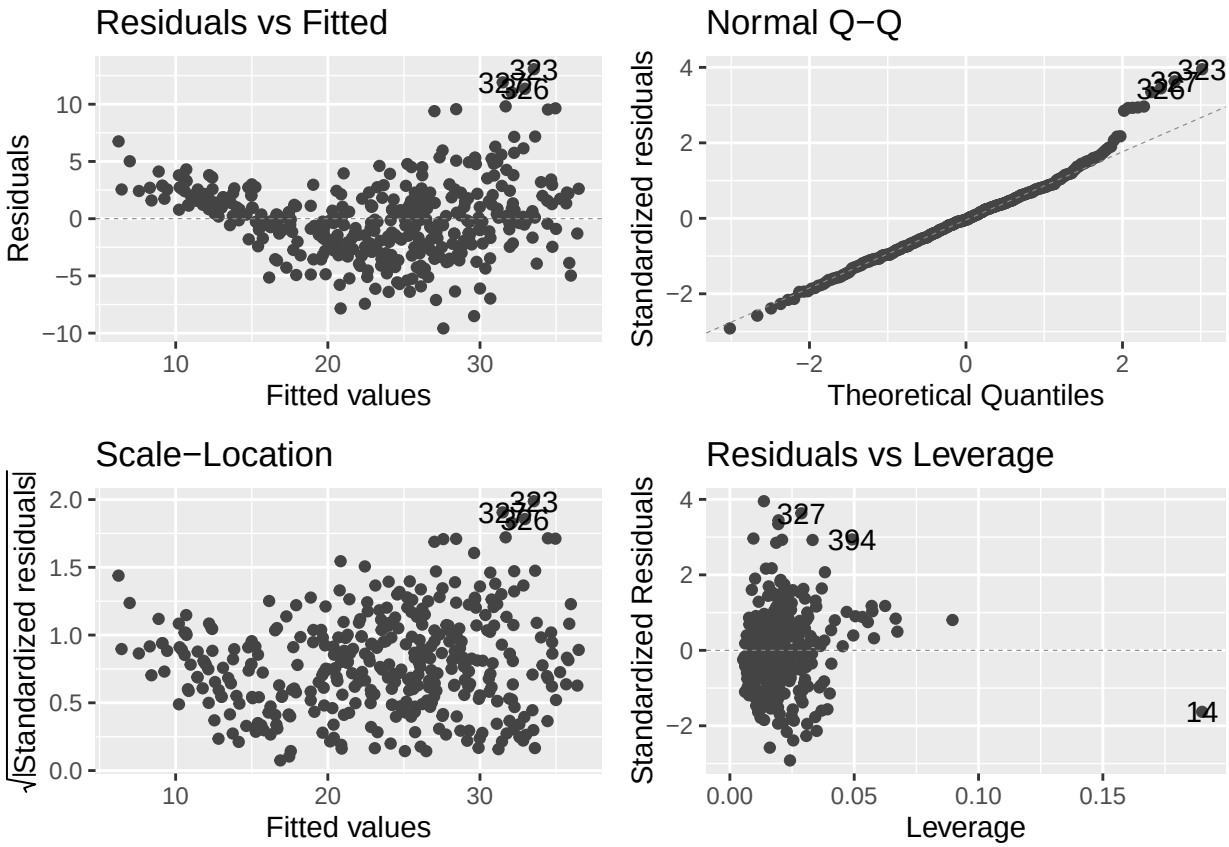
d)

Use the `autoplot()` function from the `ggfortify` package to produce diagnostic plots of the linear regression fit by setting `smooth.colour = NA`, as sometimes the smoothed line can be misleading. Comment on any

problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

Answer

```
library(ggfortify)
autoplot(fit.lm, smooth.colour = NA)
```



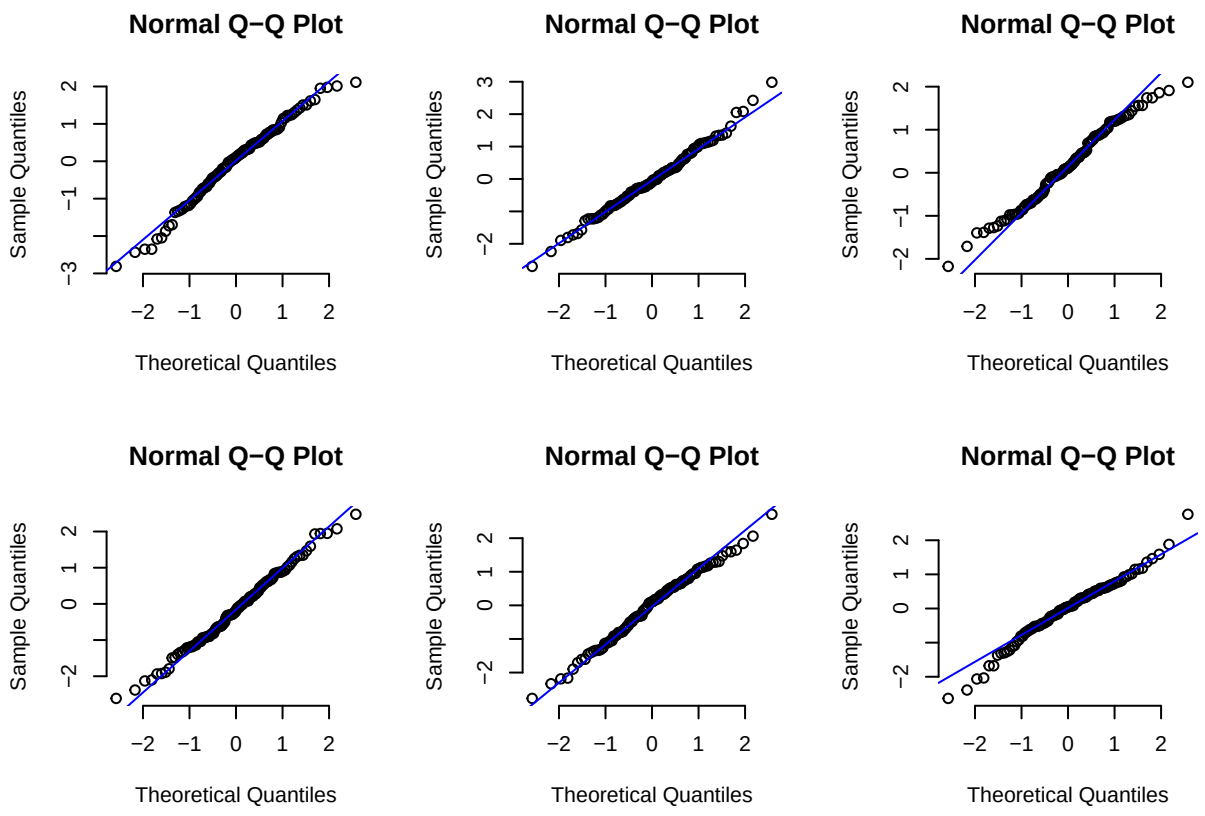
- In the residual vs fitted plot (the so-called Tukey-Anscombe plot) there is evidence of non-linearity.
- Observation 14 has an unusually high leverage. This does not necessarily need to be a problem, but it would be wise to double-check that this observation is not an outlier.

e)

For beginners, it can be difficult to decide whether a certain QQ plot looks "good" or "bad", because we only look at it and do not test anything. A way to get a feeling for how "bad" a QQ plot may look, even when the normality assumption is perfectly ok, we can use simulations: We can simply draw from the normal distribution and plot the QQ plot. Use the following code to repeat this nine times:

```
set.seed(2332)
n = 100

par(mfrow = c(2, 3))
for (i in 1:6) {
    sim = rnorm(n)
    qqnorm(sim, pch = 1, frame = FALSE)
    qqline(sim, col = "blue", lwd = 1)
}
```



f)

Let us look at interactions. These can be included via the `*` or `:` symbols in the linear predictor of the regression function (see Section 3.6.4 in the course book).

Fit the same model as before, but now also include an interaction term between `year` and `origin`. Note that `origin` is encoded as 1, 2, 3, but it is actually a qualitative predictor with three levels (1=American, 2=European, 3=Japanese)! To ensure that R treats it correctly, we first need to convert `origin` into a factor variable (a synonymous for "qualitative predictor"):

```
Auto$origin = factor(Auto$origin)
```

Answer

```
fit.lm1 = lm(mpg ~ displacement + weight + year * origin, data = Auto)
summary(fit.lm1)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + weight + year * origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7710 -2.0204 -0.0207  1.7045 13.0017
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.117e+00  5.259e+00  -0.973 0.331220
## displacement  4.803e-03  5.032e-03   0.955 0.340420
```

```
## weight       -6.685e-03  5.543e-04 -12.060  < 2e-16 ***
## year          6.152e-01  6.614e-02   9.302  < 2e-16 ***
## origin2      -3.735e+01  1.026e+01  -3.642 0.000307 ***
## origin3      -2.532e+01  9.441e+00  -2.682 0.007631 **
## year:origin2  5.187e-01  1.342e-01   3.865 0.000130 ***
## year:origin3  3.564e-01  1.213e-01   2.937 0.003514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.257 on 384 degrees of freedom
## Multiple R-squared:  0.829,  Adjusted R-squared:  0.8259
## F-statistic: 265.9 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
anova(fit.lm1)
```

```
## Analysis of Variance Table
##
## Response: mpg
##               Df  Sum Sq Mean Sq   F value    Pr(>F)
## displacement   1 15440.2 15440.2 1455.6706 < 2.2e-16 ***
## weight         1  1208.5  1208.5  113.9364 < 2.2e-16 ***
## year           1  2601.4  2601.4  245.2550 < 2.2e-16 ***
## origin         2   296.9   148.5   13.9977 1.356e-06 ***
## year:origin    2   198.9    99.5    9.3764 0.0001057 ***
## Residuals    384  4073.1    10.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is very strong evidence that the year-effect depends on the origin of the car, as can be seen by the F-test hat is given by the anova table ($p = 0.0001057$). For European (2) and Japanese (3) cars, it seems that the fuel consumption (`mpg`) has a steeper slope for year: $\beta_{year} = 0.615$ for the reference category 1 (American), wheras $\beta_{year} = 0.615 + 0.519$ and $\beta_{year} = 0.615 + 0.356$ for category 2 (European) and 3 (Japanese), respectively.

Note: For a full understanding of interaction terms, you really do need both the `summary()` and the `anova()` tables.

g)

Try a few different transformations of the variables, such as $\log(X)$, $\sqrt{X}$, X^2. See Section 3.6.5 in the course book for how to do this. Comment on your findings.

```
# try 3 predictor transformations
fit.lm3 = lm(mpg ~ poly(displacement, 2) + weight + year + origin, data = Auto)
fit.lm4 = lm(mpg ~ displacement + I(log(weight)) + year + origin, data = Auto)
fit.lm5 = lm(mpg ~ displacement + I(weight^2) + year + origin, data = Auto)
summary(fit.lm3)
```

```
##
## Call:
## lm(formula = mpg ~ poly(displacement, 2) + weight + year + origin,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8331  -1.8210   0.0548   1.7424  12.7675
##
```

```
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -2.280e+01  3.833e+00  -5.949 6.06e-09 ***
## poly(displacement, 2)1 -1.185e+01  1.062e+01  -1.116    0.265
## poly(displacement, 2)2  2.688e+01  3.705e+00   7.255 2.23e-12 ***
## weight                 -5.448e-03  5.622e-04  -9.690  < 2e-16 ***
## year                    8.183e-01  4.780e-02  17.118  < 2e-16 ***
## origin2                 8.215e-01  5.687e-01   1.444    0.149
## origin3                 7.730e-01  5.489e-01   1.408    0.160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.124 on 385 degrees of freedom
## Multiple R-squared:  0.8422, Adjusted R-squared:  0.8398
## F-statistic: 342.5 on 6 and 385 DF,  p-value: < 2.2e-16
```

```
summary(fit.lm4)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + I(log(weight)) + year + origin,
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1583 -1.9536  0.1523  1.6273 13.2890
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    134.100973  10.980595  12.213  < 2e-16 ***
## displacement     0.011137   0.004233   2.631 0.008853 **
## I(log(weight)) -22.207559   1.460920 -15.201  < 2e-16 ***
## year             0.832519   0.047400  17.564  < 2e-16 ***
## origin2          2.014019   0.516662   3.898 0.000114 ***
## origin3          1.637175   0.497463   3.291 0.001090 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.09 on 386 degrees of freedom
## Multiple R-squared:  0.8453, Adjusted R-squared:  0.8433
## F-statistic: 421.7 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
summary(fit.lm5)
```

```
##
## Call:
## lm(formula = mpg ~ displacement + I(weight^2) + year + origin,
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7042 -2.2711 -0.0008  1.9468 13.6913
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  -2.642e+01  4.440e+00  -5.950 6.01e-09 ***
## displacement -6.441e-03  5.389e-03  -1.195 0.232750
## I(weight^2)  -7.332e-07  9.216e-08  -7.956 1.99e-14 ***
## year          7.520e-01  5.500e-02  13.674  < 2e-16 ***
## origin2       2.333e+00  6.152e-01   3.792 0.000173 ***
## origin3       2.983e+00  5.789e-01   5.153 4.10e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.621 on 386 degrees of freedom
## Multiple R-squared:  0.7875, Adjusted R-squared:  0.7847
## F-statistic: 286.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

Problem 2: Theoretical questions & Simulations

a)

A core finding for the least-squares estiamtor $\hat{\boldsymbol{\beta}}$ of linear regression models is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \ ,$$

with $\hat{\boldsymbol{\beta}} \sim N_p(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.

- Show that $\hat{\boldsymbol{\beta}}$ has this distribution with the given mean and covariance matrix.
- What do you need to assume to get to this result?
- What does this imply for the distribution of the jth element of $\hat{\boldsymbol{\beta}}$?
- In particular, how can we calculate the variance of $\hat{\beta}_j$?

Answer

$$E(\hat{\boldsymbol{\beta}}) = E((X^TX)^{-1}X^TY) = (X^TX)^{-1}X^TE(Y) = (X^TX)^{-1}X^TE(X\boldsymbol{\beta} + \epsilon) \tag{1}$$
$$= (X^TX)^{-1}X^T(X\boldsymbol{\beta} + 0) = (X^TX)^{-1}(X^TX)\boldsymbol{\beta} = I\boldsymbol{\beta} = \boldsymbol{\beta} \tag{2}$$

$$Cov(\hat{\boldsymbol{\beta}}) = Cov((X^TX)^{-1}X^TY) = (X^TX)^{-1}X^TCov(Y)((X^TX)^{-1}X^T)^T \tag{3}$$
$$= (X^TX)^{-1}X^T\sigma^2I((X^TX)^{-1}X^T)^T \tag{4}$$
$$= \sigma^2(X^TX)^{-1} \tag{5}$$
$$\tag{6}$$

We need to assume that Y is multivariate normal. As $\hat{\boldsymbol{\beta}}$ is a linear transformation of a multivariate normal vector Y, $\hat{\boldsymbol{\beta}}$ is also multivariate normal.

All components of a multivariate normal vector are themselves univariate normal. This means that $\hat{\beta}_j$ is normally distributed with expected value given by the β_j and the variance given by the j'th diagonal element of $\sigma^2(X^TX)^{-1}$.

b)

What is the interpretation of a 95% confidence interval? Hint: repeat experiment (on Y), on average how many CIs cover the true β_j? The following code shows an interpentation of a 95% confidence interval. Study and fill in the code where is needed

- Model: $Y = 1 + 3X + \varepsilon$, with $\varepsilon \sim \mathsf{N}(0, 1)$.

```
beta0 = beta1 = true_beta = c(beta0, beta1)  # vector of model coefficients
true_sd = 1  # choosing true sd
X = runif(100, 0, 1)  # simulate the predictor variable X
Xmat = model.matrix(~X, data = data.frame(X))  # create design matrix


ci_int = ci_x = 0  # Counts how many times the true value is within the confidence interval
nsim = 1000
for (i in 1:nsim) {
    y = rnorm(n = 100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
    mod = lm(y ~ x, data = data.frame(y = y, x = X))
    ci = confint(mod)
    ci_int[i] = ifelse(, 1, 0)  # if true value of beta0 is within the CI then 1 else 0
    ci_x[i] = ifelse(, 1, 0)  # if true value of beta_1 is within the CI then 1 else 0
}

c(mean(ci_int), mean(ci_x))
```

Answer

Fix covariates X. *Collect Y, create CI using $\hat{\beta}$ and $\hat{\sigma}$*, repeat from * to * many times. 95 % of the times the CI contains the true β. Collect Y means simulate it with the true β as parameter(s).

```
# CI for beta_j
beta0 = 1
beta1 = 3
true_beta = c(beta0, beta1)  # vector of model coefficients
true_sd = 1  # choosing true sd
X = runif(100, 0, 1)
Xmat = model.matrix(~X, data = data.frame(X))  # Design Matrix


ci_int = ci_x = 0  # Counts how many times the true value is within the confidence interval
nsim = 1000
for (i in 1:nsim) {
    y = rnorm(n = 100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
    mod = lm(y ~ x, data = data.frame(y = y, x = X))
    ci = confint(mod)
    ci_int[i] = ifelse(true_beta[1] >= ci[1, 1] && true_beta[1] <= ci[1, 2],
        1, 0)
    ci_x[i] = ifelse(true_beta[2] >= ci[2, 1] && true_beta[2] <= ci[2, 2], 1,
        0)
}

c(mean(ci_int), mean(ci_x))
```

```
## [1] 0.955 0.947
```

c)

What is the interpretation of a 95% prediction interval? Hint: repeat experiment (on Y) for a given $\boldsymbol{x}_0$. Write R code that shows the interpretation of a 95% PI. Hint: In order to produce the PIs use the data point $x_0 = 0.4$. Furthermore you may use a similar code structure as in b).

Answer

Same idea. Fix covariates X and x_0. * Collect Y, create PI using $\hat{\beta}$ and $\hat{\sigma}$, simulate Y_0*, repeat from * to * many times. 95 % of the times the PI contains Y_0. Collect Y and Y_0 means simulate it with the true β as parameter(s). Y_0 should not be used to estimate β or σ.

```
# PI for Y_0
beta0 = 1
beta1 = 3
true_beta = c(beta0, beta1)  # vector of model coefficients
true_sd = 1  # choosing true sd
X = runif(100, 0, 1)
Xmat = model.matrix(~X, data = data.frame(X))  # Design Matrix

x0 = c(1, 0.4)

# simulating and fitting models many times
pi_y0 = 0
nsim = 1000
for (i in 1:nsim) {
    y = rnorm(n = 100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
    mod = lm(y ~ x, data = data.frame(y = y, x = X))
    y0 = rnorm(n = 1, mean = x0 %*% true_beta, sd = true_sd)
    pi = predict(mod, newdata = data.frame(x = x0[2]), interval = "predict")[2:3]
    pi_y0[i] = ifelse(y0 >= pi[1] && y0 <= pi[2], 1, 0)
}

mean(pi_y0)
```

```
## [1] 0.945
```

d)

Construct a 95% CI for $\boldsymbol{x}_0^T\beta$. Explain what is the connections between a CI for β_j, a CI for $\boldsymbol{x}_0^T\beta$ and a PI for Y at $\boldsymbol{x}_0$.

Answer

95 % CI for $\boldsymbol{x}_0^T\beta$: Same idea as for β_j. Use that $\boldsymbol{x}_0^T\hat{\beta} \sim N(\boldsymbol{x}_0^T\beta, \boldsymbol{x}_0^T\text{Var}(\hat{\beta})\boldsymbol{x}_0)$ and do as for β_j. Note that $\boldsymbol{x}_0$ is a vector. The connection between CI for β, $\boldsymbol{x}_0^T\beta$ and PI for Y at $\boldsymbol{x}_0$: The first is CI for a parameter, the second is CI for the expected regression line in the point x_0 (when you only have one covariate, this may be more intuitive), and the last is the PI for the response Y_0. The difference between the two latter is that Y are the observations, and $\boldsymbol{x}_0^T\beta$ is the expected value of the observations and hence a function of the model parameters (NOT an observation).

e)

Explain the difference between *error* and *residual*. What are the properties of the raw residuals? Why don't we want to use the raw residuals for model check? What is our solution to this?

Answer

We have a model on the form $Y = X\beta + \epsilon$ where ϵ is the error. The error of the model is unknown and unobserved, but we can estimate it by what we call the residuals. The residuals are given by the difference between the true response and the predicted value

$$\hat{\epsilon} = Y - \hat{Y} = (I - X(X^TX)^{-1}X^T)Y.$$

Properties of raw residuals: Normally distributed with mean 0 and covariance $Cov(\hat{\epsilon}) = \sigma^2(I - X(X^TX)^{-1}X^T)$. This means that the residuals may have different variance (depending on X) and may also be correlated.

In a model check, we want to check that our errors are independent, homoscedastic (same variance for each observation) and not dependent on the covariates. As we don't know the true error, we use the residuals as predictors, but as mentioned, the residuals may have different variances and may be correlated. This is why we don't want to use the raw residuals for model check.

To amend our problem we need to try to fix the residuals so that they at least have equal variances. We do that by working with standardized or studentized residuals.

Problem 3 (Compulsory 1, 2019)

The lung capacity data `lungcap` (from the `GLMsData` R package) gives information on health and on smoking habits of a sample of 654 youths, aged 3 to 19, in the area of East Boston during middle to late 1970s.

We will focus on modelling forced expiratory volume `FEV`, a measure of lung capacity. For each person in the data set we have measurements of the following 5 variables:

- `FEV` the forced expiratory volume in litres, a measure of lung capacity; a numeric vector,
- `Age` the age of the subject in completed years; a numeric vector,
- `Ht` the height in inches; a numeric vector,
- `Gender` the gender of the subjects: a numeric vector with females coded as 0 and males as 1,
- `Smoke` the smoking status of the subject: a numeric vector with non-smokers coded as 0 and smokers as 1

First we transform the height from inches to cm. Then a multiple normal linear regression model is fitted to the data set with `log(FEV)` as response and the other variables as covariates. The following R code may be used.

```
library(GLMsData)
data("lungcap")
lungcap$Htcm = lungcap$Ht * 2.54
modelA = lm(log(FEV) ~ Age + Htcm + Gender + Smoke, data = lungcap)
summary(modelA)
```

```
##
## Call:
## lm(formula = log(FEV) ~ Age + Htcm + Gender + Smoke, data = lungcap)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63278 -0.08657  0.01146  0.09540  0.40701
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.943998   0.078639 -24.721  < 2e-16 ***
## Age          0.023387   0.003348   6.984  7.1e-12 ***
## Htcm         0.016849   0.000661  25.489  < 2e-16 ***
## GenderM      0.029319   0.011719   2.502   0.0126 *
## Smoke       -0.046067   0.020910  -2.203   0.0279 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1455 on 649 degrees of freedom
```

```
## Multiple R-squared:  0.8106, Adjusted R-squared:  0.8095
## F-statistic: 694.6 on 4 and 649 DF,  p-value: < 2.2e-16
```

We call the model fitted above `modelA`.

a)

Write down the equation for the fitted `modelA`.

Answer

Model A:

$$\log(\text{FEV}) = \beta_0 + \beta_1 \text{Age} + \beta_2 \text{Htcm} + \beta_3 \text{Gender} + \beta_4 \text{Smoke} + \epsilon$$

with the fitted version $\log(\hat{\text{FEV}})$ = -1.9439982 + 0.0233872 Age + 0.0168487 Htcm + 0.0293194 Gender + -0.0460675 Smoke

b)

Explain (with words and formulas) what the following in the `summary`-output means, use the `Age` and/or the `Smoke` covariate for numerical examples.

- `Estimate` - in particular interpretation of `Intercept`
- `Std.Error`
- `Residual standard error`
- `F-statistic`

Answer

- The `Estimate` column give the estimated regression coefficients, and are given by $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. The interpretation of $\hat{\beta}_j$ is that when all other covariates are kept constant and the covariate x_j is increased to from x_j to $x_j + 1$ then on average the response increases by $\hat{\beta}_j$. For example, an increase in height of one cm is associated with an increase in the mean log(FEV) by 0.016849, keeping all other variables constant. The quantitative variable Age can be interpreted in the same way. Parameter estimates for qualitative covariates indicate how much the value of explanatory variable changes compared to the reference level. For example the value of log(FEV) will change by a factor of -0.046067 for smokers (Smoke=1), compared to non-smokers (Smoke=0).
- The `Std.Error` $\hat{SD}(\hat{\beta}_j)$ of the estimated coefficients is given by the square root of the diagonal entries of $(\mathbf{X}^T\mathbf{X})^{-1}\hat{\sigma^2}$, where $\hat{\sigma} = \text{RSS}/(n-p-1)$. Here $n = 654$ and $p = 4$.
- The residual standard error is the estimate of the standard deviation of ϵ, and is given by $\sqrt{\text{RSS}/(n-p-1)}$ where RSS=$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.
- The `F-statistic` is used test the hypothesis that all regression coefficients are zero,

$$\begin{aligned} H_0 &: \beta_1 = \beta_2 = \cdots = \beta_p = 0 \quad \text{vs} \\ H_1 &: \text{at least one } \beta \text{ is} \neq 0 \end{aligned}$$

and is computed by

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

where $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$, $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, n is the number of observations and p is the number of covariates (and $p+1$ the number of estimated regression parameters). If the p-value is less than 0.05, we reject the hypothesis that there are no coefficients with effect on the outcome in the model.

c)

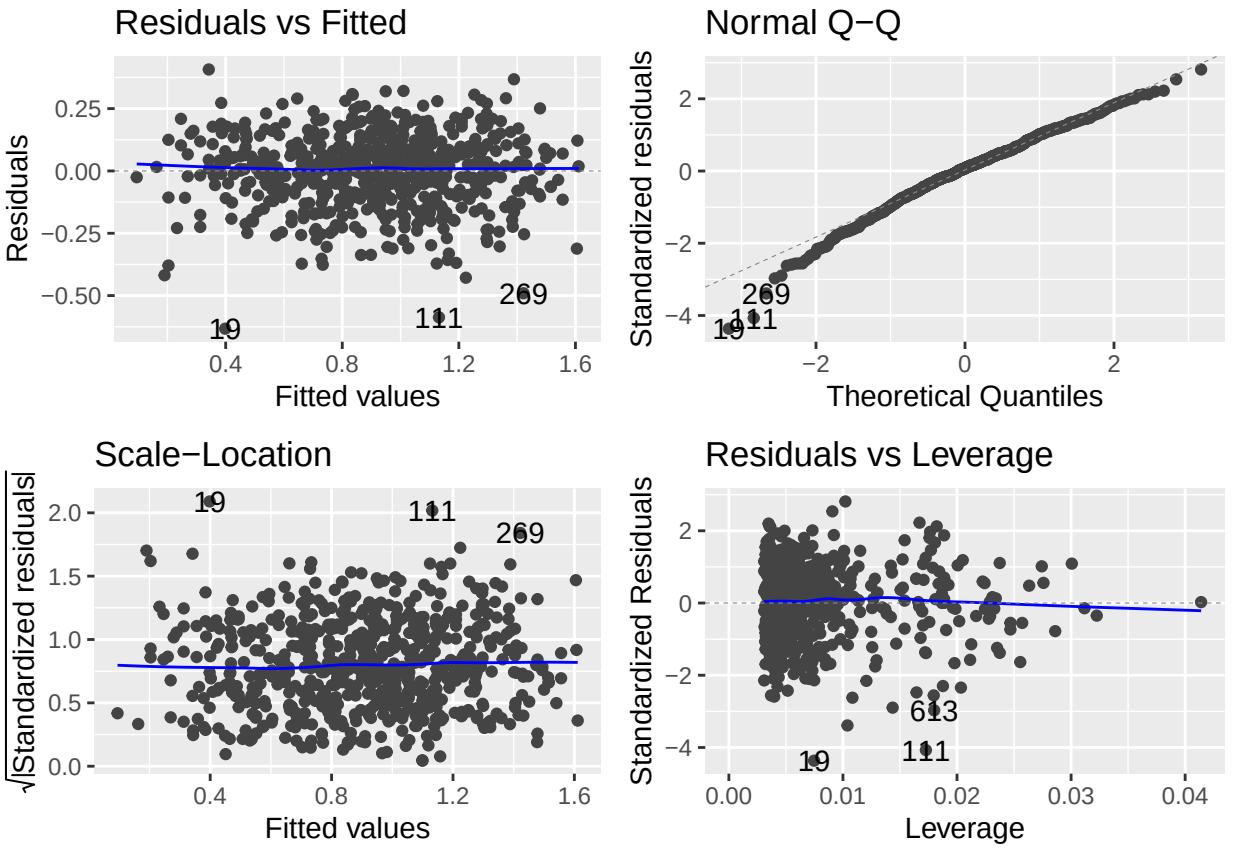
What is the proportion of variability explained by the fitted `modelA`? Comment.

Answer * The R^2 statistic gives the proportion of variance explained by the model. In this model, the proportion of variability in $Y = \log(\text{FEV})$ explained by the data X is 0.8106.

d)

Run the code below to produce diagnostic plots of "fitted values vs. standardized residuals" and "QQ-plot of standardized residuals" to assess the model fit. Comment on what you see.

```
autoplot(modelA)
```



Answer

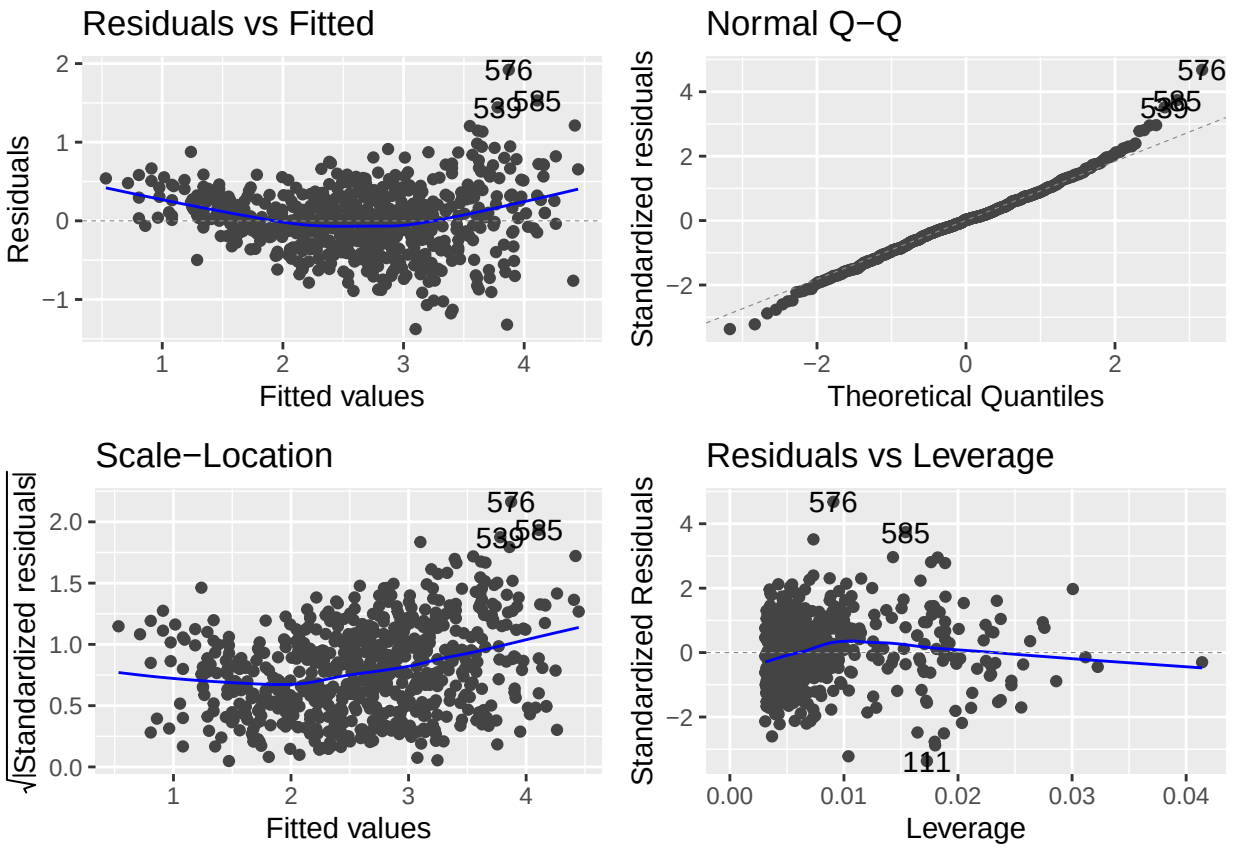
- The fitted values vs residuals plot is nice with semingly random spread but from the QQ-plot looks like the plotted values don't follow the normal line perfectly, but we know from problem 2e) that QQ-plots do typically not look perfect, even if the assumption of the normal distribution is fulfilled.
- Note: We usually avoid testing for normality. These tests have often very little power. In addition, we usually would want to show that the normal distribution is fulfilled, so we would need to formulate a test where H_0 would be "The normal distribution is violated", which we would then want to reject.

e)

Now fit a model, call this `modelB`, with `FEV` as response, and the same covariates as for `modelA`. Would you prefer to use `modelA` or `modelB` when the aim is to make inference about `FEV`? Explain what you base your conclusion on.

```
modelB = lm(FEV ~ Age + Htcm + Gender + Smoke, data = lungcap)

# residual analysis
autoplot(modelB)
```



Answer

Now we have a problem, because there seems to be a pattern in the Tukey-Anscombe plot (residuals vs fitted), and the scale-location plot indicates that the variance increases for larger fitted values. This is a very typical pattern when using the log-transformation of the response solves the problem. So clearly, modelA is the model to be used when we want to make inference.

f)

Construct a 95% and a 99% confidence interval for β_{Age} (write out the formula and calculate the interval numerically). Explain what these intervals tell you.

Answer

Using

$$T_j = \frac{\hat{\beta}_j}{\sqrt{c_{jj}}\hat{\sigma}} \sim t_{n-p-1}$$

$$P(\hat{\beta}_j - t_{\alpha/2,n-p-1}\sqrt{c_{jj}}\hat{\sigma} \leq 0 \leq \hat{\beta}_j + t_{\alpha/2,n-p-1}\sqrt{c_{jj}}\hat{\sigma}) = 1 - \alpha$$

A $(1-\alpha)\%$ CI for β_j is when we insert numerical values for the upper and lower limits:

$$\hat{\beta}_j - t_{\alpha/2,n-p-1}\sqrt{c_{jj}}\hat{\sigma}, \hat{\beta}_j + t_{\alpha/2,n-p-1}\sqrt{c_{jj}}\hat{\sigma}$$

. In this case $\alpha = 0.05$ or $\alpha = 0.01$ and $j = 2$. This means that before we have collected the data this interval has a 95% (99%) chance of covering the true value of β_{Age}. After the interval is made, the the true value is either within the interval or not.

In R, the confidence intervals for the slope estimates can easily be extracted as follows:

```
confint(modelA, level = 0.95)
```

```
##                    2.5 %       97.5 %
## (Intercept) -2.098414941 -1.789581413
## Age          0.016812109  0.029962319
## Htcm         0.015550757  0.018146715
## GenderM      0.006308481  0.052330236
## Smoke       -0.087127344 -0.005007728
```

```
confint(modelA, level = 0.99)
```

```
##                     0.5 %       99.5 %
## (Intercept) -2.1471551289 -1.740841225
## Age          0.0147367391  0.032037689
## Htcm         0.0151410623  0.018556409
## GenderM     -0.0009546847  0.059593401
## Smoke       -0.1000874831  0.007952411
```

g)

Consider a 16 year old male. He is 170 cm tall and not smoking.

```
new = data.frame(Age = 16, Htcm = 170, Gender = "M", Smoke = 0)
```

What is your best guess for his `log(FEV)`? Construct a 95% prediction interval for his forced expiratory volume `FEV`. Comment. Hint: first contruct values on the scale of the response `log(FEV)` and then transform the upper and lower limits of the prediction interval. Do you find this prediction interval useful? Comment.

Answer

```
new = data.frame(Age = 16, Htcm = 170, Gender = "M", Smoke = 0)
pred = predict(modelA, newdata = new)
pred  #Best guess for log(FEV)
```

```
##        1
## 1.323802
```

```
# the inverse of log (natural) is exp
fev = exp(pred)
fev
```

```
##       1
## 3.75768
```

```
logf.ci = predict(modelA, newdata = new, level = 0.95, interval = "prediction")
fev.ci = exp(logf.ci)
fev.ci
```

```
##       fit      lwr      upr
## 1 3.75768 2.818373 5.010038
```

We see that the interval is rather wide - so it gives us limited information. https://en.wikipedia.org/wiki/Spirometry#/media/File:Normal_values_for_FVC,_FEV1_and_FEF_25-75.png