Solutions to Recommended Exercises

TMA4268 Statistical Learning V2019. Module 3: LINEAR REGRESSION

Thea Roksvåg, Ingeborg Hem and Mette Langaas, Department of Mathematical Sciences, NTNU

week 4 2019

Contents

Last changes: (18.01.2019: first version)

Problem 1

The Framingham Heart Study is a study of the etiology (i.e. underlying causes) of cardiovascular disease, with participants from the community of Framingham in Massachusetts, USA. For more more information about the Framingham Heart Study visit https://www.framinghamheartstudy.org/. The dataset used in here is subset of a teaching version of the Framingham data, used with permission from the Framingham Heart Study.

We will focus on modelling systolic blood pressure using data from n = 2600 persons. For each person in the data set we have measurements of the seven variables

- `SYSBP` systolic blood pressure,
- `SEX` 1=male, 2=female,
- `AGE` age in years at examination,
- `CURSMOKE` current cigarette smoking at examination: 0=not current smoker, 1= current smoker,
- `BMI` body mass index,
- `TOTCHOL` serum total cholesterol, and
- `BPMEDS` use of anti-hypertensive medication at examination: 0=not currently using, 1=currently using.

A multiple normal linear regression model was fitted to the data set with `-1/sqrt(SYSBP)` as response and all the other variables as covariates.

```
library(ggplot2)
#data = read.table("https://www.math.ntnu.no/emner/TMA4268/2018v/data/SYSBPreg3uid.txt")
data = read.table("~/WWWemner/TMA4268/2018v/data/SYSBPreg3uid.txt")
dim(data)
```

```
## [1] 2600    7
```

```
colnames(data)
```

```
## [1] "SYSBP"    "SEX"      "AGE"      "CURSMOKE" "BMI"      "TOTCHOL"
## [7] "BPMEDS"
```

```
modelA=lm(-1/sqrt(SYSBP) ~ .,data = data)
summary(modelA)
```

```
##
## Call:
## lm(formula = -1/sqrt(SYSBP) ~ ., data = data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0207366 -0.0039157 -0.0000304  0.0038293  0.0189747
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.103e-01  1.383e-03 -79.745  < 2e-16 ***
## SEX         -2.989e-04  2.390e-04  -1.251 0.211176
## AGE          2.378e-04  1.434e-05  16.586  < 2e-16 ***
## CURSMOKE    -2.504e-04  2.527e-04  -0.991 0.321723
## BMI          3.087e-04  2.955e-05  10.447  < 2e-16 ***
## TOTCHOL      9.288e-06  2.602e-06   3.569 0.000365 ***
## BPMEDS       5.469e-03  3.265e-04  16.748  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005819 on 2593 degrees of freedom
## Multiple R-squared:  0.2494, Adjusted R-squared:  0.2476
## F-statistic: 143.6 on 6 and 2593 DF,  p-value: < 2.2e-16
```

a) Understanding model output

We name the model fitted above `modelA`.

- Write down the equation for the fitted `modelA`.
- Explain (with words and formula) what the following in the `summary`-output means.

- `Estimate` - in particular interpretation of `Intercept`
- `Std.Error`
- `t value`
- `Pr(>|t|)`
- `Residual standard error`
- `F-statistic`

Answers:

- Model A:

  $$-1/\sqrt{\text{SYSBP}} = \beta_0 + \beta_1\text{SEX} + \beta_2\text{AGE} + \beta_3\text{CURSMOKE} + \beta_4\text{BMI} + \beta_5\text{TOTCHOL} + \beta_6\text{BPMEDS} + \epsilon$$

  with the fitted version

  $$\widehat{1/\sqrt{\text{SYSBP}}} = -0.110 - 0.0003\text{SEX} + 0.0002\text{AGE} - 0.0003\text{CURSMOKE} + 0.0003\text{BMI} + 0.00001\text{TOTCHOL} + 0.0055\text{BPMEI}$$

- The `Estimate` is 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 the response increases by $\hat{\beta}_j$. Example, holding all other variables constant, an increase of BMI from 25 to 26 will increase the response $-1/\sqrt{\text{SYSBP}}$ by 0.00031. Similarily, for the binary variables, the coefficient estimates represents the change in the response when changing levels of the variable with one unit. For a female, the response will hence be reduced by 0.0003 compared to a male (with the same values of all the other covariate). For all variables, negative value of the estimates give reduced response when increasing the corresponding variable, while positive estimates give increased response when increasing the corresponding variable. The intercept, β_0 can be found by setting all other coefficients to zero. This involves also setting the covariate SEX to 0 - which has no meaning since SEX is coded as 1 for male and 2 for female.
- 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 $n = 2600$ and $p = 6$.
- The `t value` is the t-statistic $t = \frac{\hat{\beta}_j - \beta_j}{\hat{SD}(\hat{\beta}_j)}$, when we assume that $\beta_j = 0$.
- The `Pr(>|t|)` is the two-sided p-value for the null hypothesis $\beta_j = 0$. The p-value is calculated as the probability of observing a test staistics equal to $|t|$ or larger in absolute value, assuming that the null hypothesis is true. A p-value less than 0.05 is considered statistically significant at a 5% significance level.
- The residual standard error is the estimate of the standard deviation of ϵ, and is given by $\text{RSS}/(n-p-1)$ where $\text{RSS}=\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.
- 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.

b) Model fit

- What is the proportion of variability explained by the fitted `modelA`? Comment.
- Use diagnostic plots of "fitted values vs. standardized residuals"" and "QQ-plot of standardized residuals" (see code below) to assess the model fit.
- Now fit a model, call this `modelB`, with `SYSBP` 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 the systolic blood pressure?

```
# residuls vs fitted
ggplot(modelA, aes(.fitted, .resid)) + geom_point(pch = 21) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE, col = "red", size = 0.5, method = "loess") +
  labs(x = "Fitted values", y = "Residuals", title = "Fitted values vs. residuals", subtitle = deparse(
  theme_minimal()

# qq-plot of residuals
ggplot(modelA, aes(sample = .stdresid)) +
  stat_qq(pch = 19) +
  geom_abline(intercept = 0, slope = 1, linetype = "dotted") +
  labs(x = "Theoretical quantiles", y = "Standardized residuals", title = "Normal Q-Q", subtitle = depa



# normality test
library(nortest)
ad.test(rstudent(modelA))
```

Answers:

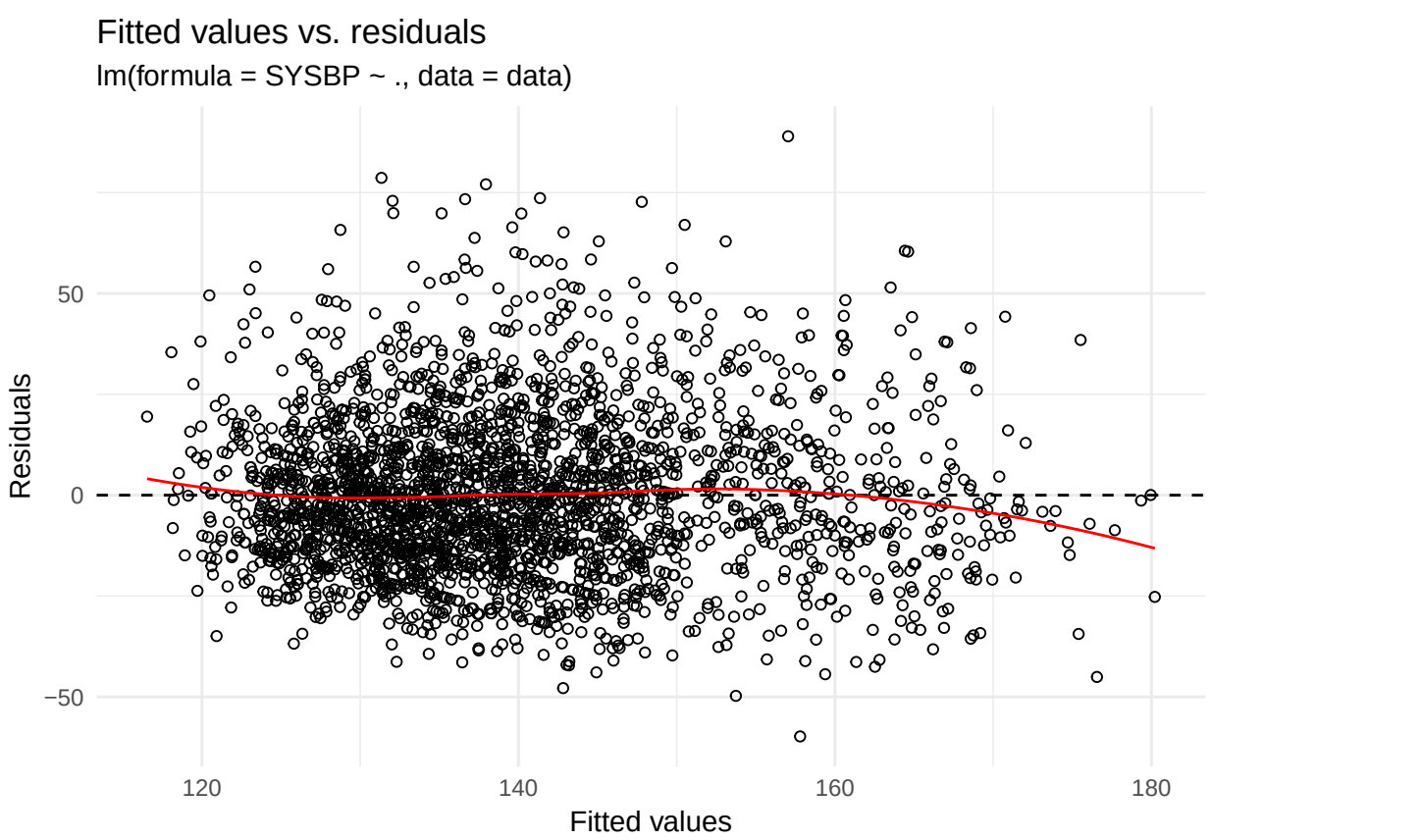
- The R^2 statistic gives the proportion of variance explained by the model. In this model, the proportion of variability in $Y = -1/\sqrt{\text{SYSBP}}$ explained by the data X is 0.2494. Since the range of R^2 is from 0 to 1, where for 1 all the variance in the response is explained by the regression model, we observe a fairly low number and we would have prefered it higher. However, these are medical data with low signal-to-noise ratio.
- Looking at the diagnostic plots, the model fit looks good. The fitted values vs residuals plot is nice with semingly random spread and the QQ-plot looks nice as the plotted values follows the normal line. In addition, the Anderson-Darling normality test does not reject the hypothesis of normality.
- For model B, we no longer model $-1/\sqrt{\text{SYSBP}}$, but rather SYSBP. This makes interpreation easier. However, looking at the diagnostic plots, we see that the QQ-plot looks suspicious at the tails, and the Anderson-Darling test rejects the null hypothesis of normal distribution.

```
modelB = lm(SYSBP ~ .,data = data)
summary(modelB)
```

```
##
## Call:
## lm(formula = SYSBP ~ ., data = data)
##
## Residuals:
```

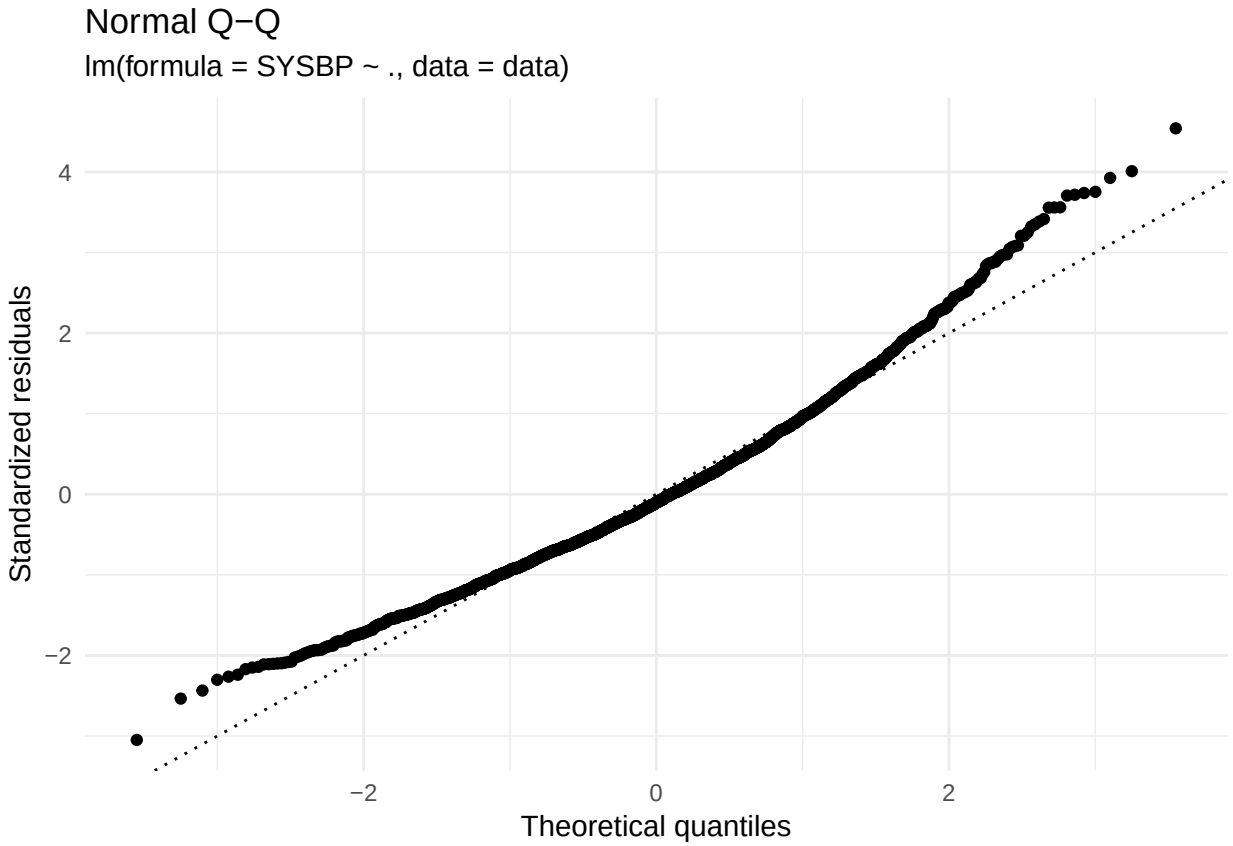
```
##     Min      1Q  Median      3Q     Max
## -59.800 -13.471  -1.982  11.063  88.959
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 56.505170   4.668798  12.103  < 2e-16 ***
## SEX         -0.429973   0.807048  -0.533  0.59424
## AGE          0.795810   0.048413  16.438  < 2e-16 ***
## CURSMOKE    -0.518742   0.853190  -0.608  0.54324
## BMI          1.010550   0.099770  10.129  < 2e-16 ***
## TOTCHOL      0.028786   0.008787   3.276  0.00107 **
## BPMEDS      19.203706   1.102547  17.418  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.65 on 2593 degrees of freedom
## Multiple R-squared:  0.2508, Adjusted R-squared:  0.249
## F-statistic: 144.6 on 6 and 2593 DF,  p-value: < 2.2e-16
```

```r
library(ggplot2)
# residuls vs fitted
ggplot(modelB, aes(.fitted, .resid)) + geom_point(pch = 21) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(se = FALSE, col = "red", size = 0.5, method = "loess") +
  labs(x = "Fitted values", y = "Residuals", title = "Fitted values vs. residuals", subtitle = deparse(
```




```r
# qq-plot of residuals
ggplot(modelB, aes(sample = .stdresid)) +
```

```
  stat_qq(pch = 19) +
  geom_abline(intercept = 0, slope = 1, linetype = "dotted") +
  labs(x = "Theoretical quantiles", y = "Standardized residuals", title = "Normal Q-Q", subtitle = depa
```




```
# normality test
library(nortest)
ad.test(rstudent(modelB))
```

```
##
##  Anderson-Darling normality test
##
## data:  rstudent(modelB)
## A = 13.2, p-value < 2.2e-16
```

c) Confidence interval and hypothesis test

We use `modelA` and focus on addressing the association between BMI and the response.

- What is the estimate $\hat{\beta}_{\text{BMI}}$ (numerically)?
- Explain how to interpret the estimated coefficient $\hat{\beta}_{\text{BMI}}$.
- Construct a 99% confidence interval for β_{BMI} (write out the formula and calculate the interval numerically). Explain what this interval tells you.
- From this confidence interval, is it possible for you know anything about the value of the p-value for the test $H_0 : \beta_{\text{BMI}} = 0$ vs. $H_1 : \beta_{\text{BMI}} \neq 0$? Explain.

Answers:

- $\hat{\beta} = (X^T X)^{-1} X^T Y$. From the summary output we find that $\hat{\beta}_{\text{BMI}} = 0.0003$. This is the average increase in $-1/sqrt(SYSBP)$ for a unit increase in BMI. Hence, keeping all other covariates fixed - having a BMI of 24 instead of 23, the value of $-1/sqrt(SYSBP)$ will on average increase with 0.0003.
- For linear regression where the distribution of the estimated coefficients are assumed to follow a t-distribution, we have that the $(1-\alpha)100\%$-confidence interval is given by

$$\hat{\beta} \pm t_{\alpha/2,df} SD(\hat{\beta})$$

For $\hat{\beta}_{BMI}$ the 99% confidence interval is hence given by

$$[\hat{\beta}_{BMI} - t_{0.005,n-p-1} SD(\hat{\beta}_{BMI}), \hat{\beta}_{BMI} + t_{0.005,n-p-1} SD(\hat{\beta}_{BMI})]$$

This means that before we have collected the data this interval has a 99% chance of covering the true value of β_{BMI}. After the interval is made - now this is [0.00023, 0.00038] the the true value is either within the interval or not. But, colleting new data and making 99% CIs, then 99% of these will on average cover the true β_{BMI}.

- Since the interval does not cover 0, we know that the p-value is less than 0.01.

```
n = dim(data)[1]
p = dim(data)[2]-1
betahat=modelA$coefficients[5]
sdbetahat=summary(modelA)$coeff[5,2]
UCI = betahat + qt(0.005, df = n-p-1, lower.tail = F)*sdbetahat
LCI = betahat - qt(0.005, df = n-p-1, lower.tail = F)*sdbetahat
c(LCI, UCI)
```

```
##          BMI          BMI
## 0.0002325459 0.0003848866
```

d) Prediction

Consider a 56 year old man who is smoking. He is 1.75 meters tall and his weight is 89 kilograms. His serum total cholesterol is 200 mg/dl and he is not using anti-hypertensive medication.

```
names(data)
```

```
## [1] "SYSBP"    "SEX"      "AGE"      "CURSMOKE" "BMI"      "TOTCHOL"
## [7] "BPMEDS"
```

```
new=data.frame(SEX=1,AGE=56,CURSMOKE=1,BMI=89/1.75^2,TOTCHOL=200,BPMEDS=0)
```

- What is your best guess for his `-1/sqrt(SYSBP)`? To get a best guess for his `SYSBP` you may take the inverse function of `-1/sqrt` (this would be a first order Taylor expansion).
- Construct a 90% prediction interval for his systolic blood pressure `SYSBP`. Comment. Hint: first contruct values on the scale of the response `-1/sqrt(SYSBP)` and then transform the upper and lower limits of the prediction interval.
- Do you find this prediction interval useful? Comment.

Answers:

Find best guess by prediction, and 90% prediction interval.

```
pred = predict(modelA, newdata = new)
pred
```

```
##           1
## -0.08667246
```

```
f.inv = function(x) 1/x^2
sys = f.inv(pred)
#pred. interval
f.ci = predict(modelA, newdata = new, level = 0.9, interval = "prediction")
f.ci
```

```
##           fit         lwr         upr
## 1 -0.08667246 -0.09625664 -0.07708829
```

```
sys.ci = f.inv(f.ci)
sys.ci
```

```
##        fit      lwr      upr
## 1 133.1183 107.9291 168.2764
```

This prediction interval is very large and doesn't really tell us much. A person with our characteristics on average has a 90% chance of having a systolic blood pressure between 108 and 168, and looking at the table given in http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/KnowYourNumbers/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp#.WnLqWOYo_AI, we sse that this interval covers almost all the levels from normal to high blood pressure. It seems our model is better for inference than prediction.

Problem 2

a)

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

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

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.

The components of a multivariate normal vector, is 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)

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). See also `R` code below.

c)

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. See also `R` code below.

d)

95 % CI for $\mathbf{x}_0^T\beta$: Same idea as for β_j. Use that $\mathbf{x}_0^T\hat{\beta} \sim N(\mathbf{x}_0^T\beta, \mathbf{x}_0^T\text{Var}(\hat{\beta})\mathbf{x}_0)$ and do as for β_j. Note that $\mathbf{x}_0$ is a vector. The connection between CI for β, $\mathbf{x}_0^T\beta$ and PI for Y at $\mathbf{x}_0$: 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 $\mathbf{x}_0^T\beta$ is the expected value of the observations and hence a function of the model parameters (NOT an observation).

e)

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.

f)

RSS(small) $\geq$ RSS(large) since RSS will be smaller with more covariates explaining the variation (and for a covariate that is completly unrelated to the data it might not be a large change, but the RSS will not increase). R^2 is directly related to RSS: R^2 = 1 - RSS/TSS, and TSS does not change when the model changes.

Problem 3: Munich Rent index

a)

```
library(ggplot2)
library(gamlss.data)
library(dplyr)
data("rent99")

rent99$location=as.factor(rent99$location)
formula1 <- rent ~ area +location + bath + kitchen + cheating
formula2 <- rentsqm ~ area + location + bath + kitchen + cheating

rent1 <- lm(formula1,data=rent99)
rent2<-lm(formula2,data=rent99)
```

Look at the summary

```
summary(rent1)
```

```
##
## Call:
## lm(formula = formula1, data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -633.41  -89.17   -6.26   82.96 1000.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.9733    11.6549  -1.885   0.0595 .
## area          4.5788     0.1143  40.055  < 2e-16 ***
## location2    39.2602     5.4471   7.208 7.14e-13 ***
## location3   126.0575    16.8747   7.470 1.04e-13 ***
## bath1        74.0538    11.2087   6.607 4.61e-11 ***
## kitchen1    120.4349    13.0192   9.251  < 2e-16 ***
## cheating1   161.4138     8.6632  18.632  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 145.2 on 3075 degrees of freedom
## Multiple R-squared:  0.4504, Adjusted R-squared:  0.4494
## F-statistic:   420 on 6 and 3075 DF,  p-value: < 2.2e-16
```

```
summary(rent2)
```

```
##
## Call:
## lm(formula = formula2, data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4959 -1.4084 -0.0733  1.3847  9.4400
##
## Coefficients:
```

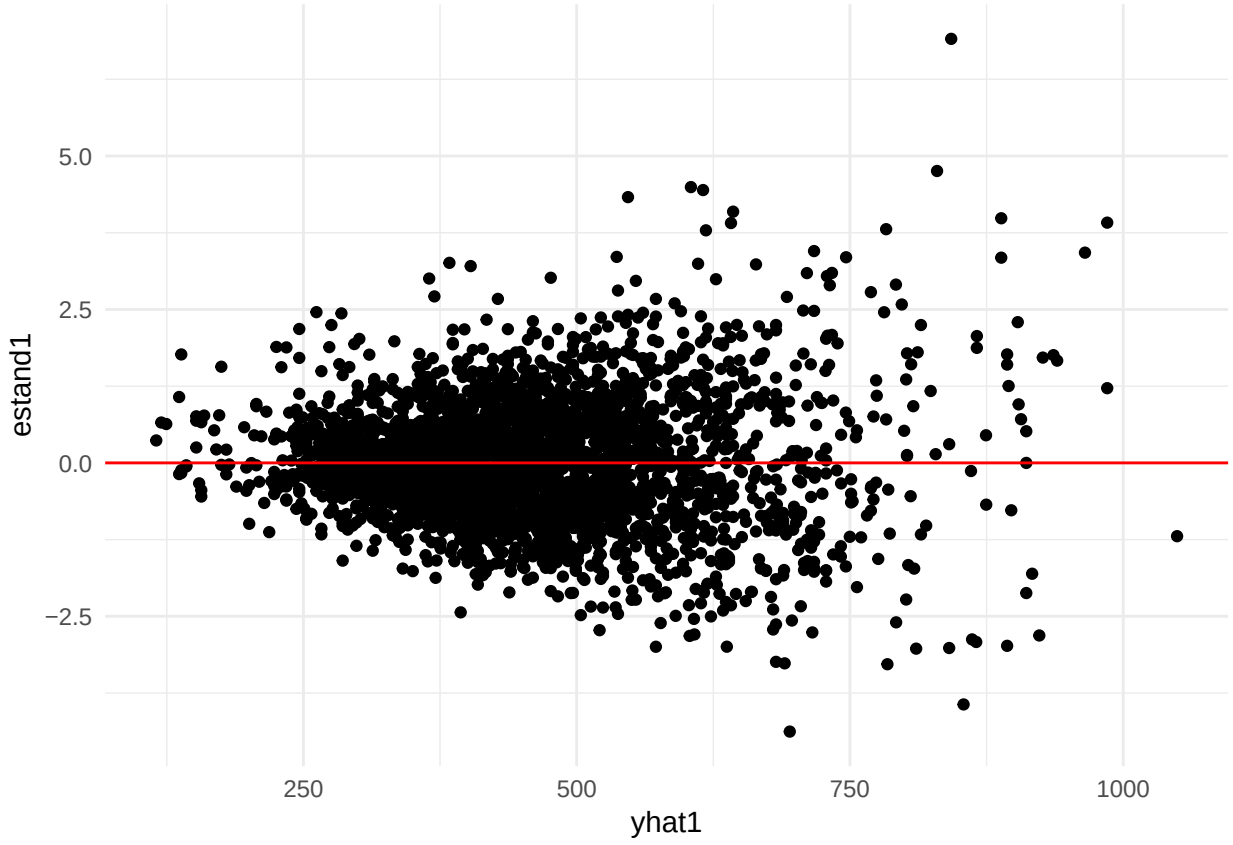
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.108319   0.168567  42.169  < 2e-16 ***
## area        -0.038154   0.001653 -23.077  < 2e-16 ***
## location2    0.628698   0.078782   7.980 2.04e-15 ***
## location3    1.686099   0.244061   6.909 5.93e-12 ***
## bath1        0.989898   0.162113   6.106 1.15e-09 ***
## kitchen1     1.412113   0.188299   7.499 8.34e-14 ***
## cheating1    2.414101   0.125297  19.267  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.1 on 3075 degrees of freedom
## Multiple R-squared:  0.2584, Adjusted R-squared:  0.2569
## F-statistic: 178.6 on 6 and 3075 DF,  p-value: < 2.2e-16
```

Consider residual plots. We plot standardized residuals against fitted values.

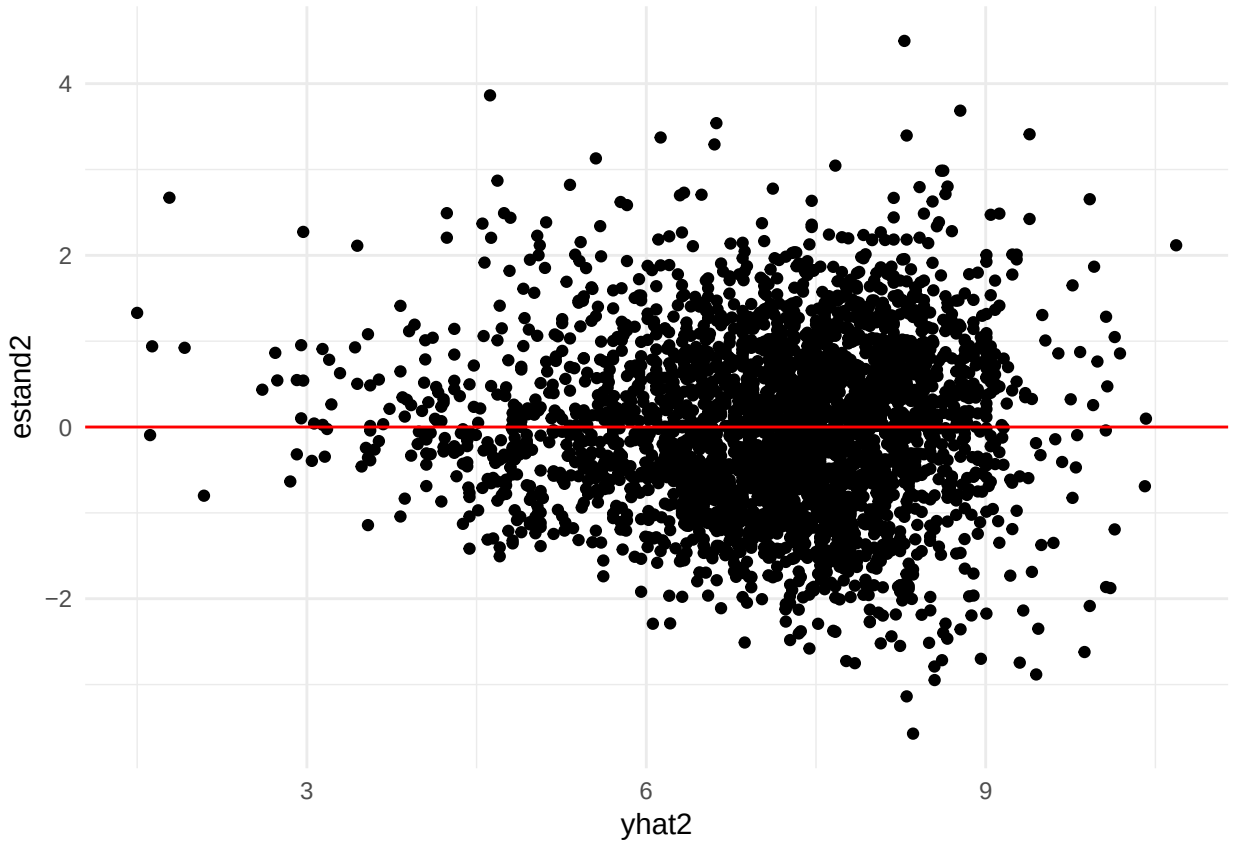
```
yhat1=predict(rent1)
yhat2=predict(rent2)

estand1=rstandard(rent1);
estand2=rstandard(rent2)

ggplot(data.frame(yhat1,estand1),aes(yhat1,estand1))+geom_point(pch=19)+geom_abline(intercept=0,slope=0
```

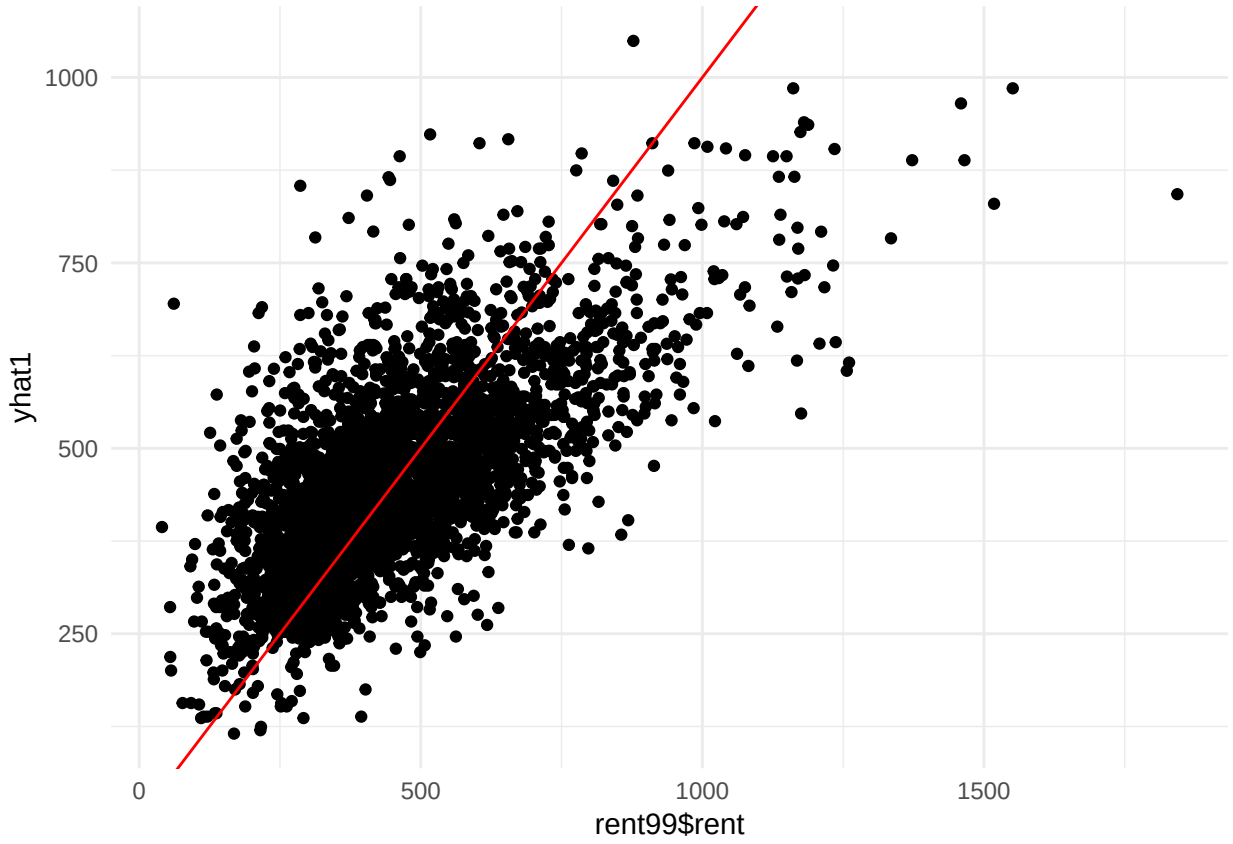



```
ggplot(data.frame(yhat2,estand2),aes(yhat2,estand2))+geom_point(pch=19)+geom_abline(intercept=0,slope=0
```

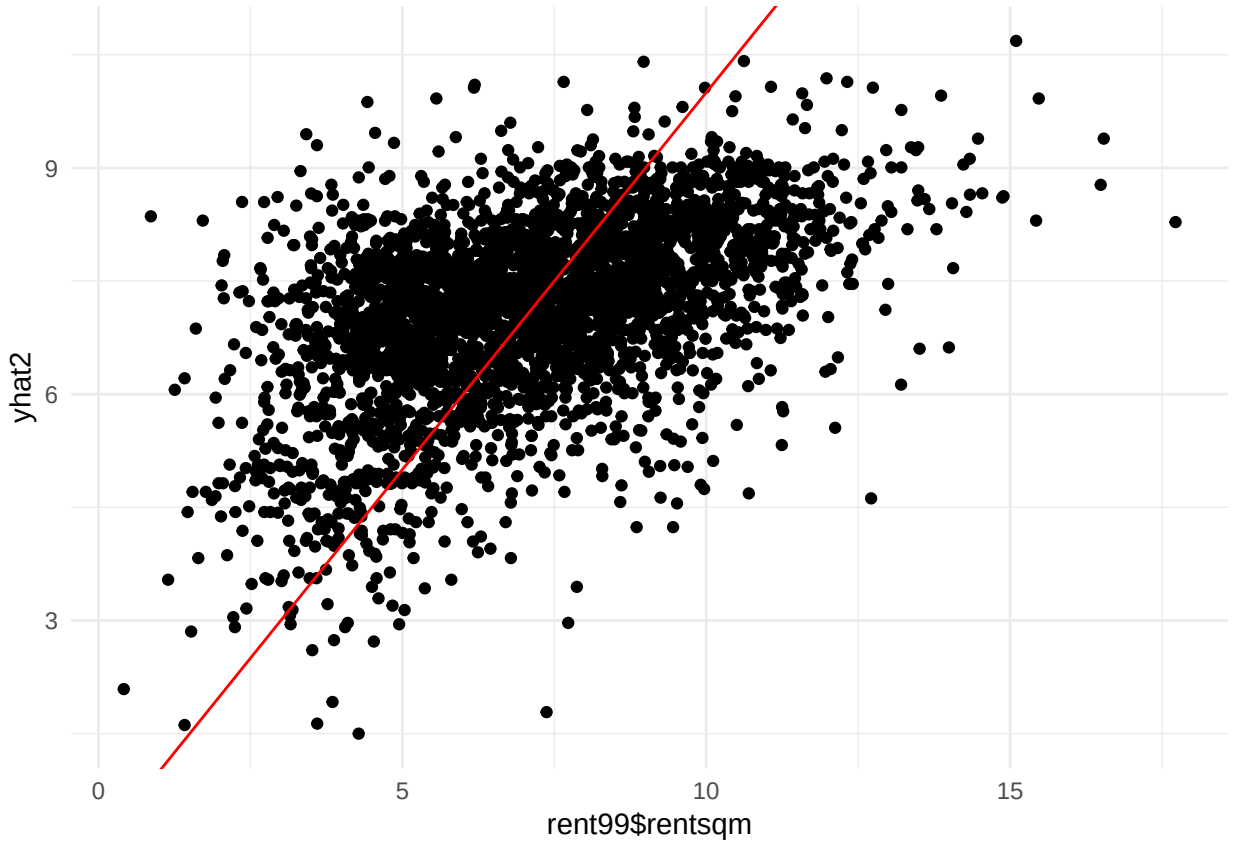


We plot the true observations against the fitted values

```
ggplot(data.frame(yhat1,rent99),aes(rent99$rent,yhat1))+geom_point(pch=19)+geom_abline(intercept=0,slop
```

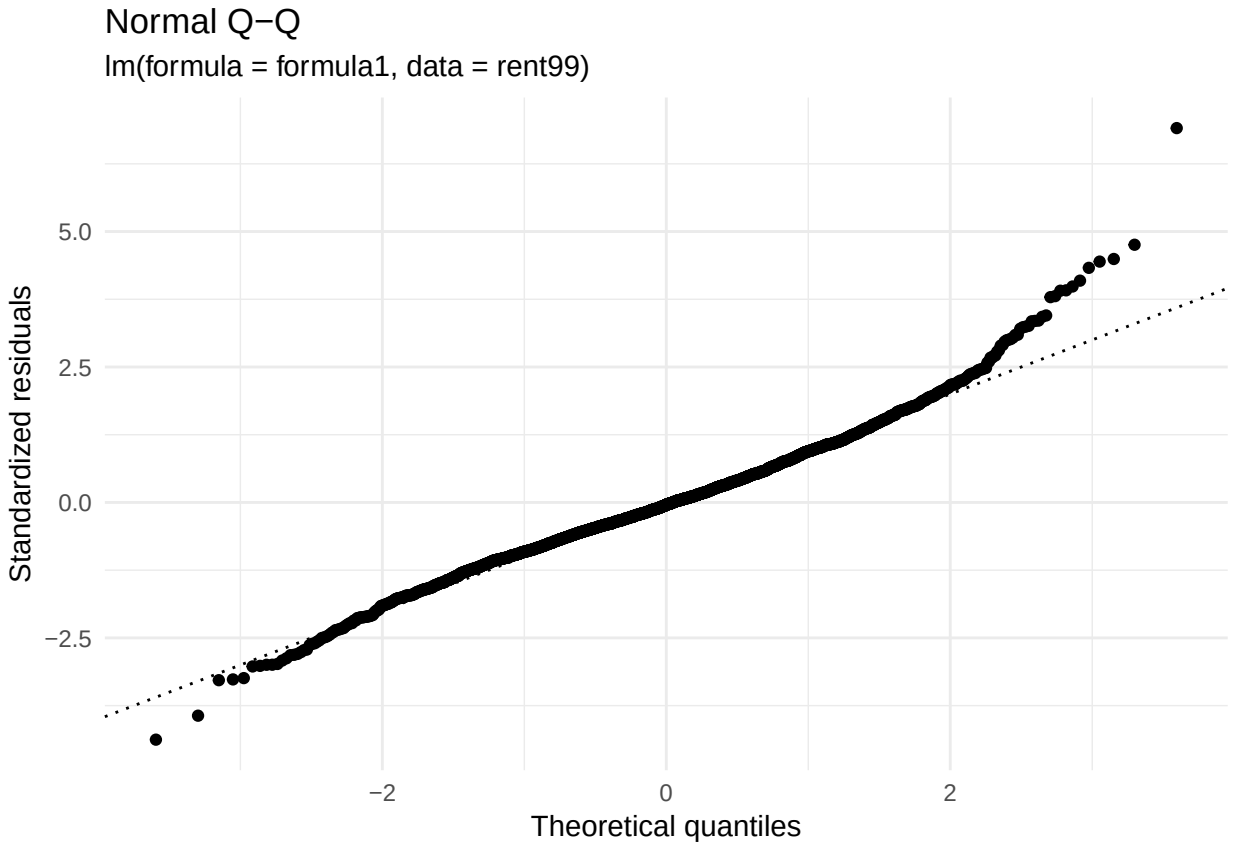



```
ggplot(data.frame(yhat2,rent99),aes(rent99$rentsqm,yhat2))+geom_point(pch=19)+geom_abline(intercept=0,s
```

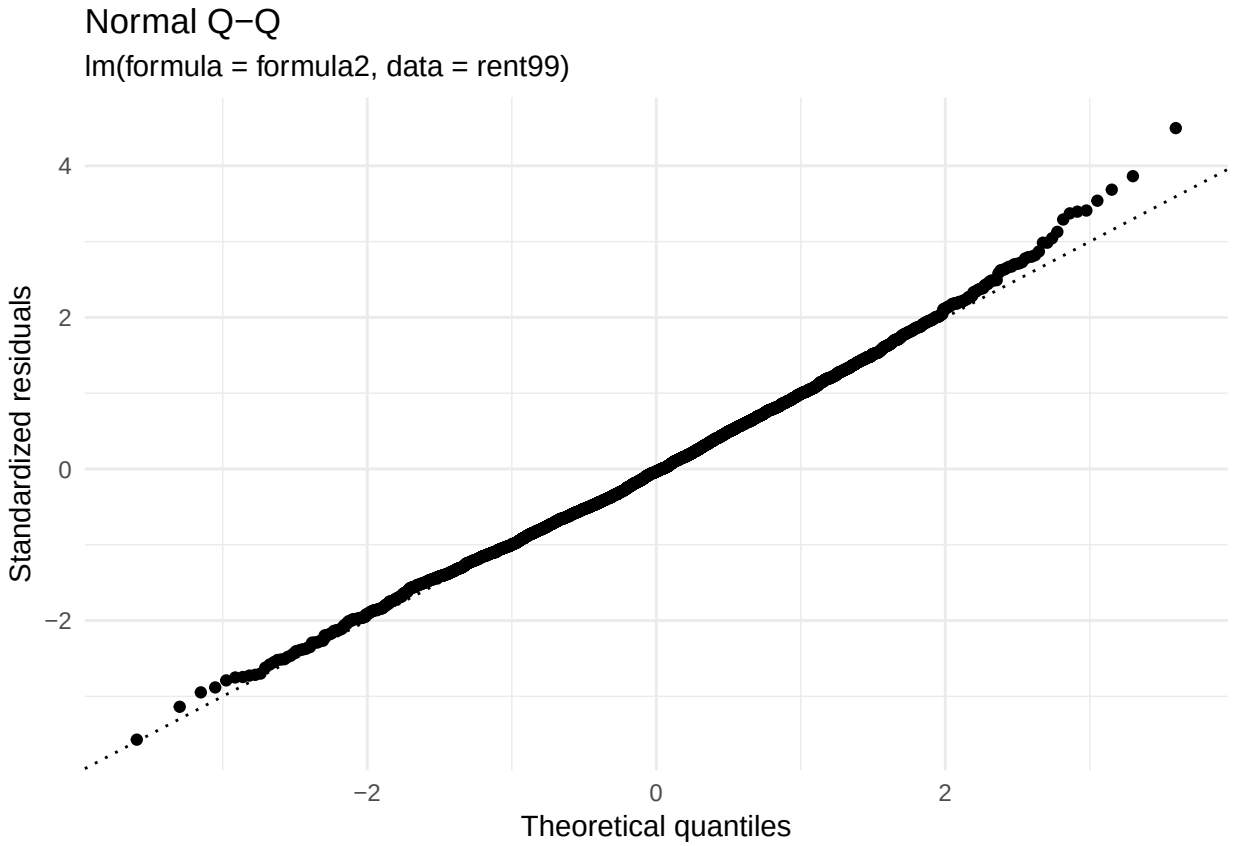


Normal Q-Q

```
ggplot(rent1, aes(sample = .stdresid)) +
  stat_qq(pch = 19) +
  geom_abline(intercept = 0, slope = 1, linetype = "dotted") +
  labs(x = "Theoretical quantiles", y = "Standardized residuals", title = "Normal Q-Q", subtitle = depa
```




```
ggplot(rent2, aes(sample = .stdresid)) +
  stat_qq(pch = 19) +
  geom_abline(intercept = 0, slope = 1, linetype = "dotted") +
  labs(x = "Theoretical quantiles", y = "Standardized residuals", title = "Normal Q-Q", subtitle = depa
```



Can't really see that one response is better than the other, so we proceed with *rent*.

b&c)

We consider the summary

```
summary(rent1)
```

```
##
## Call:
## lm(formula = formula1, data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -633.41  -89.17   -6.26   82.96 1000.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.9733    11.6549  -1.885   0.0595 .
## area          4.5788     0.1143  40.055  < 2e-16 ***
## location2    39.2602     5.4471   7.208 7.14e-13 ***
## location3   126.0575    16.8747   7.470 1.04e-13 ***
## bath1        74.0538    11.2087   6.607 4.61e-11 ***
## kitchen1    120.4349    13.0192   9.251  < 2e-16 ***
## cheating1   161.4138     8.6632  18.632  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 145.2 on 3075 degrees of freedom
## Multiple R-squared:  0.4504, Adjusted R-squared:  0.4494
## F-statistic:   420 on 6 and 3075 DF,  p-value: < 2.2e-16
```

The column *estimate* gives us the $\hat{\beta}$'s. For example if the living area of the house increases qith 1 square meter, the predicted response, net rent per month, increases with 4.5788 euro. The column std.error gives the standard deviation of the $\hat{\beta}$'s. The remaining columns report the t value (*t value*) with corresponding p-value (*Pr(>|t|)*). The *RSE*, R^2, R^2_{adj} and F-statistic are also reported as *Residual standard error*, *Multiple R-squared*, *Adjusted R-Squared* and *F-statistic* respectively. See module page for definition of these terms.

Interpretation of the intercept: This is the intercept of a average location (location=1). (If location=2, the intercept is -21.9733+39.2602, and if location=3 the intercept is -21.9733+126.0575.)

d)

```
orgfit=lm(rent~area+location+bath+kitchen+cheating,data=rent99)
summary(orgfit)
set.seed(1) #to be able to reproduce results
n=dim(rent99)[1]
IQ=rnorm(n,100,16)
fitIQ=lm(rent~area+as.factor(location)+bath+kitchen+cheating+IQ,data=rent99)
summary(fitIQ)

summary(orgfit)$sigma
summary(fitIQ)$sigma

summary(orgfit)$r.squared
summary(fitIQ)$r.squared
summary(orgfit)$adj.r.squared
summary(fitIQ)$adj.r.squared
```

```
##
## Call:
## lm(formula = rent ~ area + location + bath + kitchen + cheating,
##     data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -633.41  -89.17   -6.26   82.96 1000.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -21.9733    11.6549  -1.885   0.0595 .
## area          4.5788     0.1143  40.055  < 2e-16 ***
## location2    39.2602     5.4471   7.208 7.14e-13 ***
## location3   126.0575    16.8747   7.470 1.04e-13 ***
## bath1        74.0538    11.2087   6.607 4.61e-11 ***
## kitchen1    120.4349    13.0192   9.251  < 2e-16 ***
## cheating1   161.4138     8.6632  18.632  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 145.2 on 3075 degrees of freedom
```

```
## Multiple R-squared:  0.4504, Adjusted R-squared:  0.4494
## F-statistic:   420 on 6 and 3075 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = rent ~ area + as.factor(location) + bath + kitchen +
##     cheating + IQ, data = rent99)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -630.95  -89.50   -6.12   82.62  995.76
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -41.3879    19.5957  -2.112   0.0348 *
## area                   4.5785     0.1143  40.056  < 2e-16 ***
## as.factor(location)2  39.2830     5.4467   7.212 6.90e-13 ***
## as.factor(location)3 126.3356    16.8748   7.487 9.18e-14 ***
## bath1                 74.1979    11.2084   6.620 4.23e-11 ***
## kitchen1             120.0756    13.0214   9.221  < 2e-16 ***
## cheating1            161.4450     8.6625  18.637  < 2e-16 ***
## IQ                     0.1940     0.1574   1.232   0.2179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 145.2 on 3074 degrees of freedom
## Multiple R-squared:  0.4507, Adjusted R-squared:  0.4494
## F-statistic: 360.3 on 7 and 3074 DF,  p-value: < 2.2e-16
##
## [1] 145.1879
## [1] 145.1757
## [1] 0.4504273
## [1] 0.4506987
## [1] 0.449355
## [1] 0.4494479
```

R^2 will always increase (or stay the same) if we add a parameter to the model. Thus, we cannot use this alone for model selection. However, the adjusted R^2_{adj} is "punished" based on the number of parameters in the model and will not necessarily increase if we add a covariate to the model.

Sigma (or RSE) is given by $\hat{\sigma} = \sqrt{\frac{1}{n-p-1}RSS}$. Multiple R-squared is given by $R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\hat{\sigma}^2(n-p-1)}{TSS}$.

e)

```
formula <- rent ~ area + location + bath + kitchen + cheating
rent1 <- lm(formula, data = rent99)#, contrasts = list(location = "contr.sum"))

rent99 <- rent99 %>% mutate(yearc.cat = cut(yearc, breaks = c(-Inf, seq(1920,2000,10)), labels = 10*1:9)

formula <- rent ~ area + location + bath + kitchen + cheating + yearc.cat
rent2 <- lm(formula, data = rent99)#, contrasts = list(location = "contr.sum"))

rent99 <- rent99 %>% mutate(yearc.cat2 = cut(yearc, breaks = c(-Inf, seq(1920,2000,20)), labels = c(20,4
```

```r
formula <- rent ~ area + location + bath + kitchen + cheating + yearc.cat2
rent3 <- lm(formula, data = rent99)#,contrasts = list(location = "contr.sum"))
```

f)

```r
library(MASS)
library(leaps)
best <- regsubsets(model.matrix(rent3)[,-1], y = rent99$rent,method="exhaustive")
summary(best)
```

```
## Subset selection object
## 10 Variables  (and intercept)
##               Forced in Forced out
## area              FALSE      FALSE
## location2         FALSE      FALSE
## location3         FALSE      FALSE
## bath1             FALSE      FALSE
## kitchen1          FALSE      FALSE
## cheating1         FALSE      FALSE
## yearc.cat240      FALSE      FALSE
## yearc.cat260      FALSE      FALSE
## yearc.cat280      FALSE      FALSE
## yearc.cat20       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          area location2 location3 bath1 kitchen1 cheating1 yearc.cat240
## 1  ( 1 ) "*"  " "       " "       " "   " "      " "       " "
## 2  ( 1 ) "*"  " "       " "       " "   " "      " "       " "
## 3  ( 1 ) "*"  " "       " "       " "   " "      "*"       " "
## 4  ( 1 ) "*"  " "       "*"       " "   " "      "*"       " "
## 5  ( 1 ) "*"  "*"       "*"       " "   " "      "*"       " "
## 6  ( 1 ) "*"  "*"       "*"       " "   " "      "*"       " "
## 7  ( 1 ) "*"  "*"       "*"       " "   "*"      "*"       " "
## 8  ( 1 ) "*"  "*"       "*"       "*"   "*"      "*"       " "
##          yearc.cat260 yearc.cat280 yearc.cat20
## 1  ( 1 ) " "          " "          " "
## 2  ( 1 ) " "          " "          "*"
## 3  ( 1 ) " "          " "          "*"
## 4  ( 1 ) " "          " "          "*"
## 5  ( 1 ) " "          " "          "*"
## 6  ( 1 ) " "          "*"          "*"
## 7  ( 1 ) " "          "*"          "*"
## 8  ( 1 ) " "          "*"          "*"
```

A selection method is used (you will learn more later). The output shows the best model of each size (1-8 covariates). The best model with one covariate uses only area, the best model with two covariates uses area and yearc.cat20 and so on.

```r
summary(best)$cp
```

```
## [1] 1015.979023  540.680039  228.460243  184.179283  125.679898   75.667739
## [7]   32.571877    9.418625
```

Model 8 gives the lowest Mallows Cp and is the preferred model.

Problem 4: Simulations in R

a)

```
# CI for beta_j

true_beta <- c(3.14, 10, 0.8) # choosing true betas
true_sd <- 10 # choosing true sd
set.seed(345); X <- matrix(c(rep(1, 100), runif(100, 2, 5), sample(1:100, 100, replace = TRUE)),
            nrow = 100, ncol = 3) # fixing X. set.seed() is used to produce same X every time this code

# simulating and fitting models many times
ci_int <- ci_x1 <- ci_x2 <- 0; nsim <- 1000
for (i in 1:nsim){
  y <- rnorm(n = 100, mean = X%*%true_beta, sd = rep(true_sd, 100))
  mod <- lm(y ~ x1 + x2, data = data.frame(y = y, x1 = X[,2], x2 = X[,3]))
  ci <- confint(mod)
  ci_int[i] <- ifelse(true_beta[1] >= ci[1,1] && true_beta[1] <= ci[1,2], 1, 0)
  ci_x1[i] <- ifelse(true_beta[2] >= ci[2,1] && true_beta[2] <= ci[2,2], 1, 0)
  ci_x2[i] <- ifelse(true_beta[3] >= ci[3,1] && true_beta[3] <= ci[3,2], 1, 0)
}

c(mean(ci_int), mean(ci_x1), mean(ci_x2))
```

```
## [1] 0.952 0.944 0.945
```

b)

```
# PI for Y_0

true_beta <- c(3.14, 10, 0.8) # choosing true betas
true_sd <- 10 # choosing true sd
set.seed(345);
X <- matrix(c(rep(1, 100), runif(100, 2, 5),
              sample(1:100, 100,replace = TRUE)), nrow = 100, ncol = 3) # fixing X.

#set.seed() is used to produce same X every time this code is used

x0 <- c(1,3,50)

# simulating and fitting models many times
pi_y0 <- 0; nsim <- 1000
for (i in 1:nsim){
  y <- rnorm(n = 100, mean = X%*%true_beta, sd = rep(true_sd, 100))
  mod <- lm(y ~ x1 + x2, data = data.frame(y = y, x1 = X[,2], x2 = X[,3]))
  y0 <- rnorm(n = 1, mean = x0%*%true_beta, sd = true_sd)
  pi <- predict(mod, newdata = data.frame(x1 = x0[2], x2 = x0[3]), interval = "predict")[,2:3]
  pi_y0[i] <- ifelse (y0 >= pi[1] && y0 <=pi[2], 1, 0)
}
```

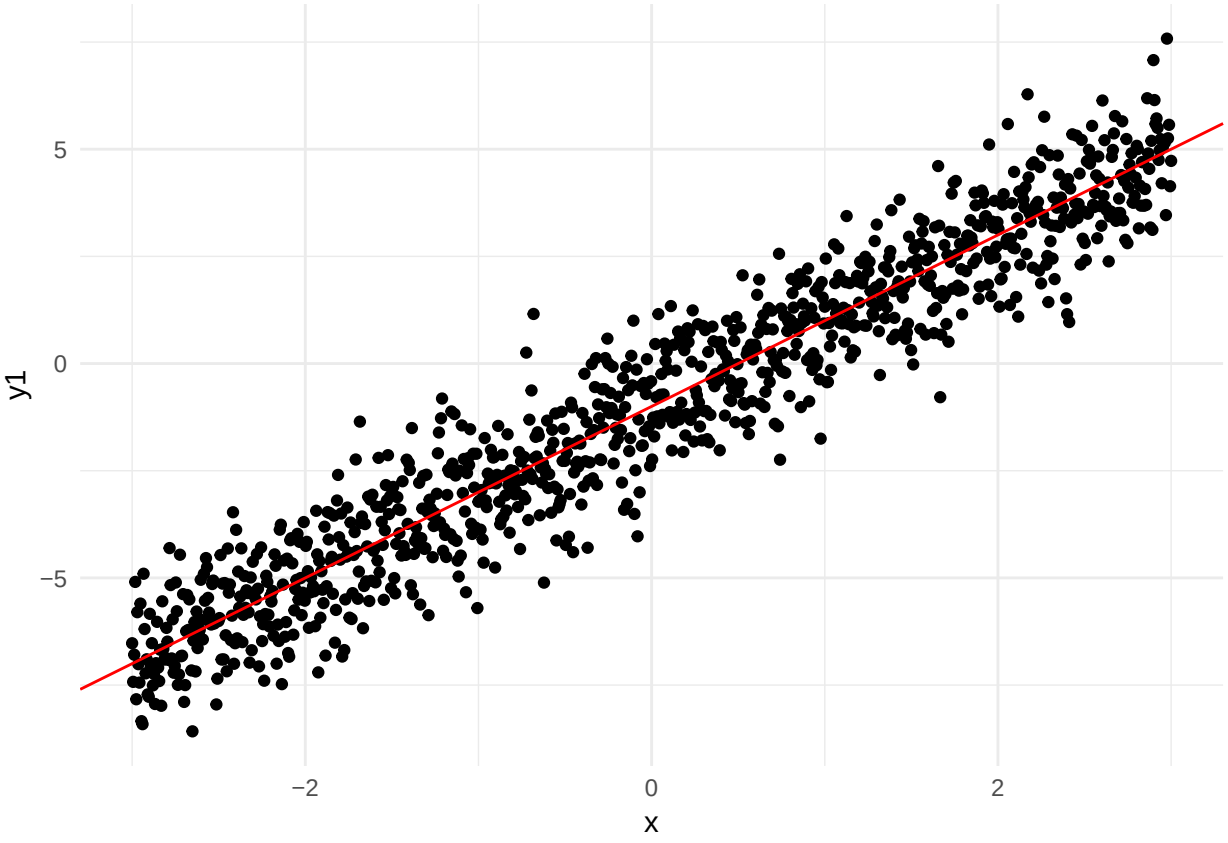
```r
mean(pi_y0)
```

```
## [1] 0.958
```

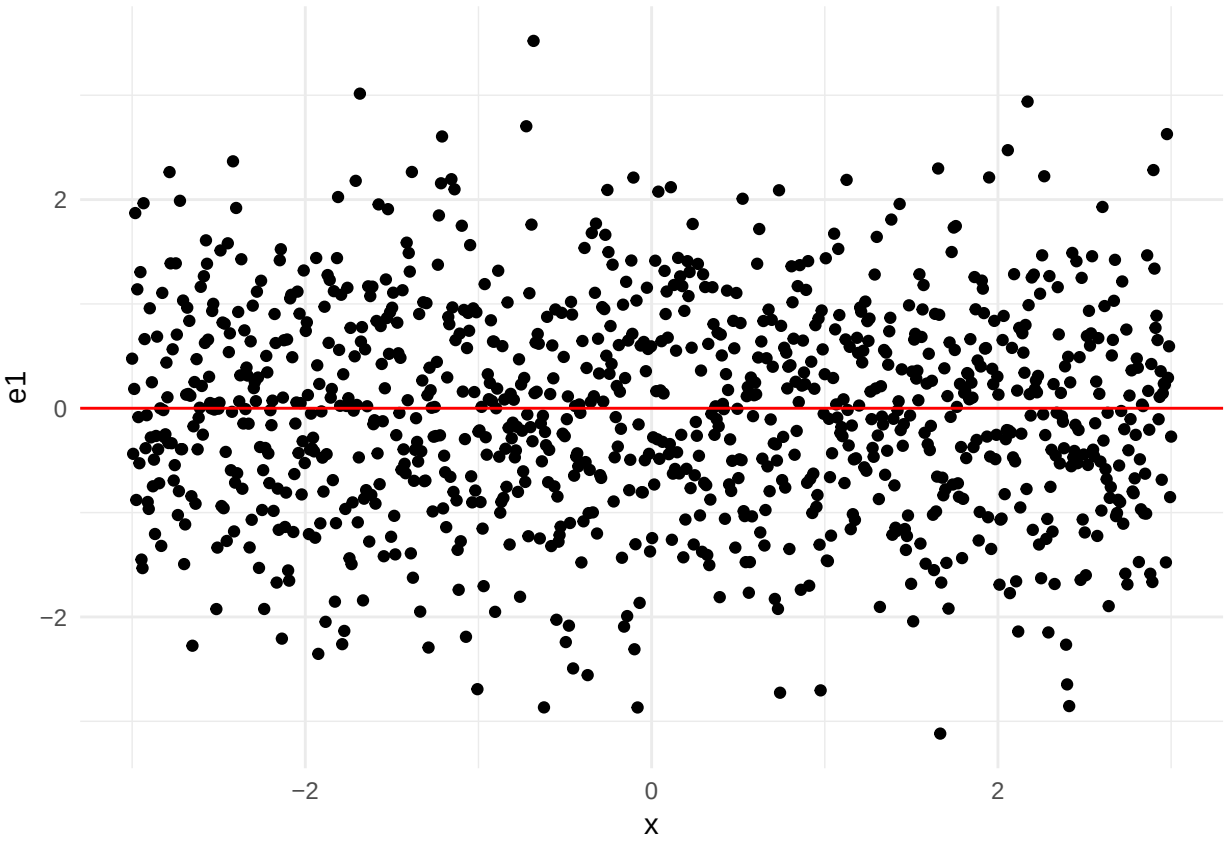
c)

```r
library(ggplot2)
#Homoscedastic error
n=1000
x=seq(-3,3,length=n)
beta0=-1
beta1=2
xbeta=beta0+beta1*x
sigma=1
e1=rnorm(n,mean=0,sd=sigma)
y1=xbeta+e1
ehat1=residuals(lm(y1~x))

#ggplot-solution
ggplot(data.frame(x=x,y1=y1),aes(x,y1)) +
  geom_point(pch =19)+geom_abline(slope=beta1,intercept=beta0,col="red")+  theme_minimal()
```



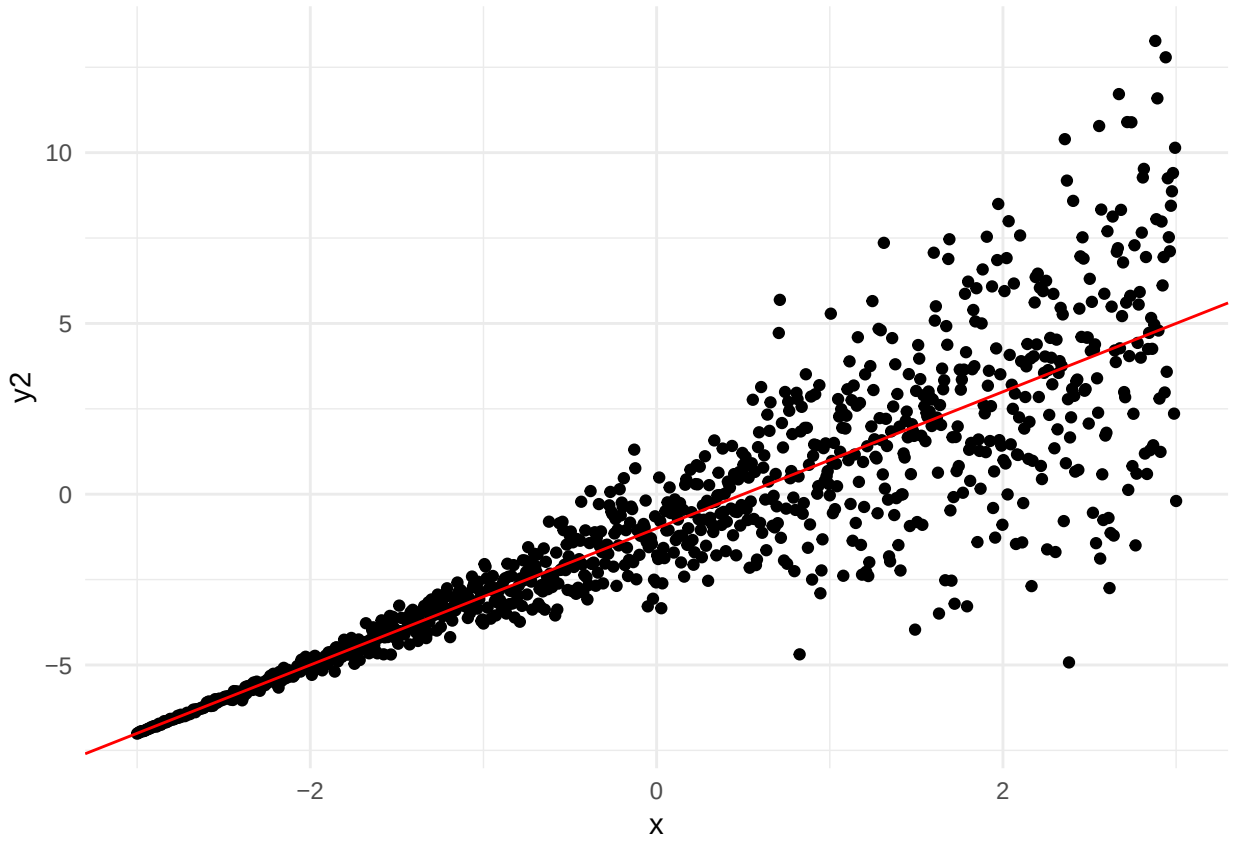

```r
ggplot(data.frame(x=x,e1=e1),aes(x,e1))+geom_point(pch=19)+geom_hline(yintercept=0,col="red")+  theme_m
```



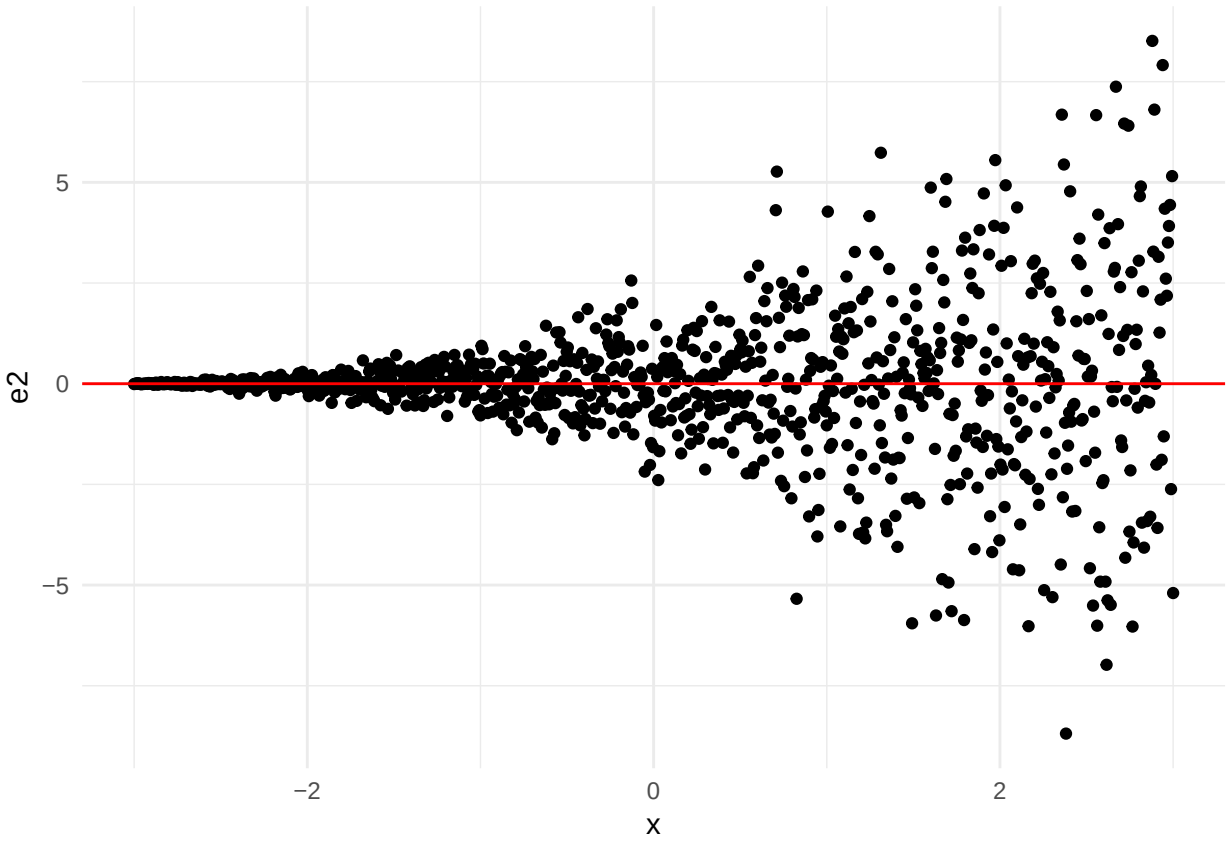
Correct model: We don't see any pattern in the residual plot. The variance seems to be independent of the covariate x.

```
#Heteroscedastic errors
sigma=(0.1+0.3*(x+3))^2
e2=rnorm(n,0,sd=sigma)
y2=xbeta+e2
ehat2=residuals(lm(y2~x))

#ggplot-solution
ggplot(data.frame(x=x,y2=y2),aes(x,y2)) +
  geom_point(pch =19)+geom_abline(slope=beta1,intercept=beta0,col="red")+  theme_minimal()
```




```
ggplot(data.frame(x=x,e2=e2),aes(x,e2))+geom_point(pch=19)+geom_hline(yintercept=0,col="red")+  theme_m
```



Wrong model: The variance of the residuals increases as a function of x.

d)

Reduce the sample size to for example $n = 10$. Then we see a difference between the standardized and the studentized residuals (red and blue). The expressions for standardized and studentized residuals for an observation y_i are identical, except that the latter estimates $\hat{\sigma}$ without using observation number i. When the sample size is large, it typically doesn't matter if we include y_i in the estimation or not.

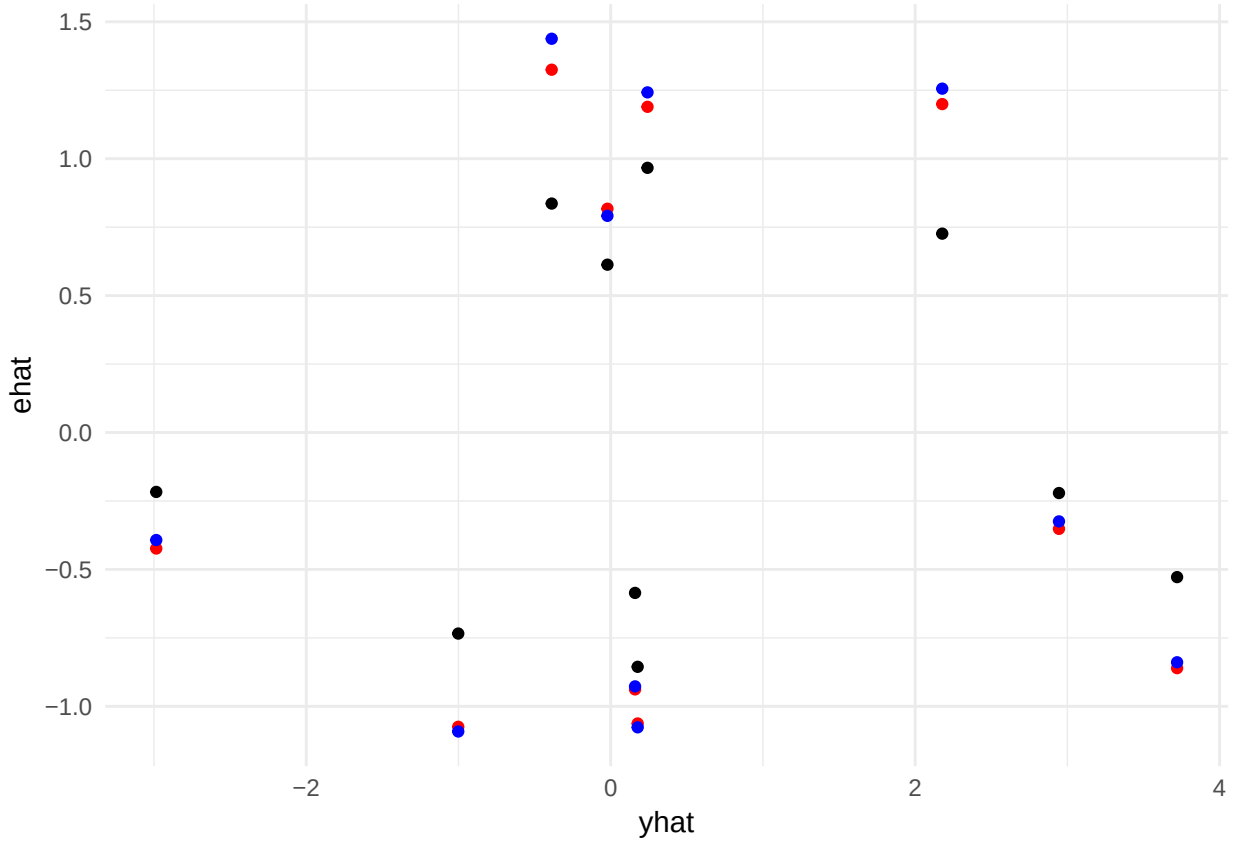
```
n=10
beta=matrix(c(0,1,1/2,1/3),ncol=1)
set.seed(123)
x1=rnorm(n,0,1); x2=rnorm(n,0,2); x3=rnorm(n,0,3)
X=cbind(rep(1,n),x1,x2,x3)
y=X%*%beta+rnorm(n,0,2)
fit=lm(y~x1+x2+x3)
yhat=predict(fit)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8557 -0.5714 -0.2191  0.6980  0.9667
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4958     0.3067   1.617  0.15706
## x1            1.8135     0.3744   4.844  0.00287 **
## x2            0.3260     0.1909   1.708  0.13853
## x3            0.2076     0.1268   1.638  0.15262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8686 on 6 degrees of freedom
## Multiple R-squared:  0.8852, Adjusted R-squared:  0.8278
## F-statistic: 15.43 on 3 and 6 DF,  p-value: 0.003162
```

```
ehat=residuals(fit); estand=rstandard(fit); estud=rstudent(fit)

ggplot(data=data.frame(ehat,yhat,estand,estud),aes(yhat,ehat))+geom_point(pch=19)+geom_point(aes(yhat,es
```



R packages

```
install.packages("gamlss.data")
install.packages("tidyverse")
install.packages("GGally")
install.packages("Matrix")
install.packages("ggpubr")
install.packages("nortest")
```

```r
install.packages("MASS")
```