TMA4315 Generalized linear models H2018

Module 2: MULTIPLE LINEAR REGRESSION

Mette Langaas, Department of Mathematical Sciences, NTNU – with contributions from Oyvind Bakke

30.08 and 06.09 [PL], 31.08 and 07.09 [IL]

Contents

(Latest changes: 09.12 Typos from Scott. 05.09.2018. Typos and added wordcloud code.)

Overview

Learning material

- Textbook: Chapter 2.2, 3 and B.4. (Chapter 3 was on the reading list for TMA4267 Linear statistical 2016-2018, so much of this module is know from before - but not from a GLM point of view!)
- Classnotes 30.08.2018
- Classnotes 06.09.2018

Topics

First week

- Aim of multiple linear regression.
- Define and understand the multiple linear regression model - traditional and GLM way
- parameter estimation with maximum likelihood (and least squares),
- likelihood, score vector and Hessian (observed Fisher information matrix)
- properties of parameter estimators,
- assessing model fit (diagnostic), residuals, QQ-plots,
- design matrix: how to code categorical covariates (dummy or effect coding), and how to handle interactions.

Jump to IL for first week

Second week

- What did we do last week?
- Parameter estimation in practice.
- Statistical inference for parameter estimates
  - confidence intervals,
  - prediction intervals,
  - hypothesis test,
  - linear hypotheses.
- Introducing deviance - and likelihood ratio test
- Analysis of variance decompositions and R^2, sequential ANOVA table.
- Model selection with AIC

Jump to second week and IL for second week

FIRST WEEK

Aim of multiple linear regression

1. Construct a model to help understand the relationship between a response and one or several explanatory variables. [Correlation, or cause and effect?]
2. Construct a model to predict the reponse from a set of (one or several) explanatory variables. [More or less "black box"]

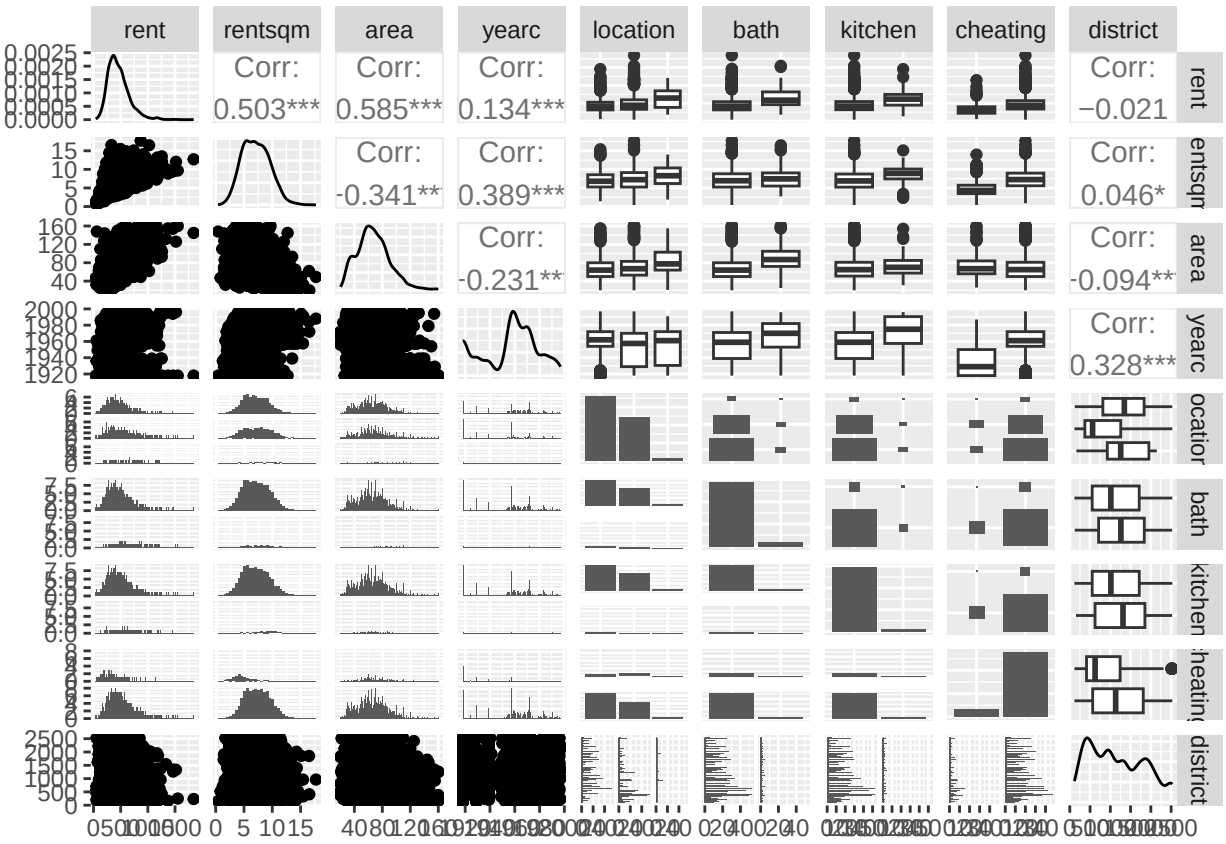
Munich rent index

Munich, 1999: 3082 observations on 9 variables.

- `rent`: the net rent per month (in Euro).
- `rentsqm`: the net rent per month per square meter (in Euro).
- `area`: living area in square meters.
- `yearc`: year of construction.
- `location`: quality of location: a factor indicating whether the location is average location, 1, good location, 2, and top location, 3.
- `bath`: quality of bathroom: a a factor indicating whether the bath facilities are standard, 0, or premium, 1.
- `kitchen`: Quality of kitchen: 0 standard 1 premium.
- `cheating`: central heating: a factor 0 without central heating, 1 with central heating.
- `district`: District in Munich.

More information in Fahrmeir et. al., (2013) page 5.

```
library("gamlss.data")
library(GGally)
ggpairs(rent99, lower = list(combo = wrap(ggally_facethist, binwidth = 0.5)))
```



Interesting questions

1. Is there a relationship between `rent` and `area`?
2. How strong is this relationship?
3. Is the relationship linear?
4. Are also other variables associated with `rent`?
5. How well can we predict the rent of an appartment?
6. Is the effect of `area` the same on `rent` for appartments at average, good and top `location`? (interaction)

Notation

$\mathbf{Y}$: $(n \times 1)$ vector of responses (random variable) [e.g. one of the following: rent, rent pr sqm, weight of baby, ph of lake, volume of tree]

$\mathbf{X}$: $(n \times p)$ design matrix [e.g. location of flat, gestation age of baby, chemical measurement of the lake, height of tree]

$\boldsymbol{\beta}$: $(p \times 1)$ vector of regression parameters (intercept included, so $p = k + 1$)

ε : $(n \times 1)$ vector of random errors. Used in "traditional way".

We assume that pairs $(\mathbf{x}_i^T, y_i)$ $(i = 1, ..., n)$ are measured from sampling units. That is, the observation pair $(\mathbf{x}_1^T, y_1)$ is independent from $(\mathbf{x}_2^T, y_2)$, and so on.

Hands on: Munich rent index — response and covariates

Study the print-out and discuss the following questions:

a) What can be response, and what covariates? (using what you know about rents)
b) What type of response(s) do we have? (continuous, categorical, nominal, ordinal, discrete, factors, ...).
c) What types of covariates? (continuous, categorical, nominal, ordinal, discrete, factors, ...)
d) Explain what the elements of `model.matrix` are. (Hint: coding of `location`)

```
library("gamlss.data")
ds = rent99
colnames(ds)
```

```
## [1] "rent"     "rentsqm"  "area"     "yearc"    "location" "bath"     "kitchen"
## [8] "cheating" "district"
```

```
summary(ds)
```

```
##       rent            rentsqm             area            yearc      location
##  Min.   :  40.51   Min.   : 0.4158   Min.   : 20.00   Min.   :1918   1:1794
##  1st Qu.: 322.03   1st Qu.: 5.2610   1st Qu.: 51.00   1st Qu.:1939   2:1210
##  Median : 426.97   Median : 6.9802   Median : 65.00   Median :1959   3:  78
##  Mean   : 459.44   Mean   : 7.1113   Mean   : 67.37   Mean   :1956
##  3rd Qu.: 559.36   3rd Qu.: 8.8408   3rd Qu.: 81.00   3rd Qu.:1972
##  Max.   :1843.38   Max.   :17.7216   Max.   :160.00   Max.   :1997
##  bath     kitchen  cheating    district
##  0:2891   0:2951   0: 321   Min.   : 113
##  1: 191   1: 131   1:2761   1st Qu.: 561
##                             Median :1025
##                             Mean   :1170
##                             3rd Qu.:1714
##                             Max.   :2529
```

```
dim(ds)
```

```
## [1] 3082    9
```

```
head(ds)
```

```
##       rent   rentsqm area yearc location bath kitchen cheating district
## 1 109.9487  4.228797   26  1918        2    0       0        0      916
## 2 243.2820  8.688646   28  1918        2    0       0        1      813
## 3 261.6410  8.721369   30  1918        1    0       0        1      611
## 4 106.4103  3.547009   30  1918        2    0       0        0     2025
## 5 133.3846  4.446154   30  1918        2    0       0        1      561
## 6 339.0256 11.300851   30  1918        2    0       0        1      541
```

```
str(ds$location)
```

```
##  Factor w/ 3 levels "1","2","3": 2 2 1 2 2 2 1 1 1 2 ...
```

```
contrasts(ds$location)
```

```
##   2 3
## 1 0 0
## 2 1 0
## 3 0 1
```

```r
X = model.matrix(rentsqm ~ area + yearc + location + bath + kitchen +
    cheating + district, data = ds)
head(X)
```

```
##   (Intercept) area yearc location2 location3 bath1 kitchen1 cheating1 district
## 1           1   26  1918         1         0     0        0         0      916
## 2           1   28  1918         1         0     0        0         1      813
## 3           1   30  1918         0         0     0        0         1      611
## 4           1   30  1918         1         0     0        0         0     2025
## 5           1   30  1918         1         0     0        0         1      561
## 6           1   30  1918         1         0     0        0         1      541
```

Model

The traditional way

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon$$

is called a classical linear model if the following is true:

1. $\mathrm{E}(\varepsilon) = \mathbf{0}$.
2. $\mathrm{Cov}(\varepsilon) = \mathrm{E}(\varepsilon\varepsilon^{\mathbf{T}}) = \sigma^{\mathbf{2}}\mathbf{I}$.
3. The design matrix has full rank, $\mathrm{rank}(\mathbf{X}) = k + 1 = p$.

The classical *normal* linear regression model is obtained if additionally

4. $\varepsilon \sim N_n(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{I})$ holds.

For random covariates these assumptions are to be understood conditionally on $\mathbf{X}$.

The GLM way

Independent pairs $(Y_i, \mathbf{x}_i)$ for $i = 1, \ldots, n$.

1. Random component: $Y_i \sim N$ with $\mathrm{E}(Y_i) = \mu_i$ and $\mathrm{Var}(Y_i) = \sigma^2$.
2. Systematic component: $\eta_i = \mathbf{x}_i^T\boldsymbol{\beta}$.
3. Link function: linking the random and systematic component (linear predictor): Identity link and response function. $\mu_i = \eta_i$.

Questions

- Compare the traditional and GLM way. Have we made the same assumptions for both?
- What is the connection between each $\mathbf{x}_i$ and the design matrix?
- What is "full rank"? Why is this needed? Example of rank less than p?
- Why do you think we move from traditional to GLM way? Could we not just let ε be from binomial, Poisson, etc. distribution?

Parameter estimation

In multiple linear regression there are two popular methods for estimating the regression parameters in $\boldsymbol{\beta}$: maximum likelihood and least squares. These two methods give the same estimator when we assume the normal linear regression model. We will in this module focus on maximum likelihood estimation, since that can be used also when we have non-normal responses (modules 3-6: binomial, Poisson, gamma, multinomial).

Likelihood theory (from B.4)

Likelihood $L(\boldsymbol{\beta})$

We assume that pairs of covariates and response are measured independently of each other: $(\mathbf{x}_i, Y_i)$, and Y_i follows the distribution specified above, and $\mathbf{x}_i$ is fixed.

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} L_i(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i; \boldsymbol{\beta})$$

Q: fill in with the normal density for f and the multiple linear regression model.

###Loglikelihood $l(\boldsymbol{\beta})$ The log-likelihood is just the natural log of the likelihood, and we work with the log-likelihood because this makes the mathematics simpler - since we work with exponential families. The main aim with the likelihood is to maximize it to find the maximum likelihood estimate, and since the log is a monotone function the maximum of the log-likelihood will be in the same place as the maximum of the likelihood.

$$l(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ln L_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\beta})$$

Observe that the log-likelihood is a sum of invidual contributions for each observation pair i.

Q: fill in with the normal density for f and the multiple linear regression model.

Repetition: rules for derivatives with respect to vector

Hardle and Simes (2015), page 65.

- Let $\boldsymbol{\beta}$ be a p-dimensional column vector of interest,
- and let $\frac{\partial}{\partial \boldsymbol{\beta}}$ denote the p-dimensional vector with partial derivatives wrt the p elements of $\boldsymbol{\beta}$.
- Let $\mathbf{d}$ be a p-dimensional column vector of constants and
- $\mathbf{D}$ be a $p \times p$ symmetric matrix of constants.

Rule 1:

$$\frac{\partial}{\partial \boldsymbol{\beta}}(\mathbf{d}^T\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}(\sum_{j=1}^{p} d_j\beta_j) = \mathbf{d}$$

Rule 2:

$$\frac{\partial}{\partial \boldsymbol{\beta}}(\boldsymbol{\beta}^T\mathbf{D}\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}(\sum_{j=1}^{p}\sum_{k=1}^{p} \beta_j d_{jk}\beta_k) = 2\mathbf{D}\boldsymbol{\beta}$$

Rule 3: The Hessian of the quadratic form $\boldsymbol{\beta}^T\mathbf{D}\boldsymbol{\beta}$ is

$$\frac{\partial^2 \boldsymbol{\beta}^T\mathbf{D}\boldsymbol{\beta}}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^T} = 2\mathbf{D}$$

###Score function $s(\beta)$

The score function is a $p \times 1$ vector, $s(\beta)$, with the partial derivatives of the log-likelihood with respect to the p elements of the β vector.

$$s(\beta) = \frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{n} \frac{\partial l_i(\beta)}{\partial \beta} = \sum_{i=1}^{n} s_i(\beta)$$

Again, observe that the score function is a sum of individual contributions for each observation pair i.

Q: fill in for the multiple linear regression model.

To find the maximum likelihood estimate $\hat{\beta}$ we solve the set of p equations:

$$s(\hat{\boldsymbol{\beta}}) = 0$$

Q: fill in for the multiple linear regression model. Specify what the *normal equations* are.

For the normal linear regression model, these equations $s(\hat{\beta}) = 0$ have a solution to be written on closed form.

Least squares and maximum likelihood (ML) estimator for β:

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

Q: Least squares is found by minimizing $\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2$. How can you see that least squares and ML gives the same estimator?

Looking ahead: Hessian and Fisher information

But, for other distribution than the normal we get a set of non-linear equations when we look at $s(\hat{\beta}) = 0$, and then we will use the Newton-Raphson or Fisher Scoring iterative methods.

Observed Fisher information matrix $H(\beta)$

$$H(\beta) = -\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T} = -\frac{\partial s(\beta)}{\partial\beta^T}$$

so this is minus the Hessian of the loglikelihood.

- $H(\beta)$ may be considered as a *local measure of information* that the likelihood contains.
- The higher the curvature of the log-likelihood near its maximum the more information is provide by the likelihood about the unknown parameter.

Q: Calculate this for the multiple linear regression model. What is the dimension of $H(\beta)$?

In addition we also use the *expected Fisher information matrix* $F(\beta)$ which we may find in two ways, one is by taking the mean of the observed Fisher information matrix:

$$F(\beta) = E\left(-\frac{\partial^2 l(\beta)}{\partial\beta\partial\beta^T}\right).$$

Q: Calculate this for the multiple linear regression model. What is the dimension of $F(\beta)$?

In Module 3 we need the Fisher information matrix in the Newton-Raphson method, and also to find the (asympotic) covariance matrix of our estimated coefficents $\hat{\beta}$ - so much more about this then.

Hands on: Munich rent index parameter estimates

Explain what the values under `Estimate` mean in practice.

```
fit = lm(rentsqm ~ area + yearc + location + bath + kitchen + cheating,
    data = ds)
summary(fit)
```

```
##
## Call:
## lm(formula = rentsqm ~ area + yearc + location + bath + kitchen +
##     cheating, data = ds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4303 -1.4131 -0.1073  1.3244  8.6452
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -45.475484   3.603775 -12.619  < 2e-16 ***
## area         -0.032330   0.001648 -19.618  < 2e-16 ***
## yearc         0.026959   0.001846  14.606  < 2e-16 ***
## location2     0.777133   0.076870  10.110  < 2e-16 ***
## location3     1.725068   0.236062   7.308 3.45e-13 ***
## bath1         0.762808   0.157559   4.841 1.35e-06 ***
## kitchen1      1.136908   0.183088   6.210 6.02e-10 ***
## cheating1     1.765261   0.129068  13.677  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.031 on 3074 degrees of freedom
## Multiple R-squared:  0.3065, Adjusted R-squared:  0.3049
## F-statistic: 194.1 on 7 and 3074 DF,  p-value: < 2.2e-16
```

Reproduce the values under `Estimate` by calculating without the use of `lm`.

```
X = model.matrix(rentsqm ~ area + yearc + location + bath + kitchen +
    cheating, data = ds)
Y = ds$rentsqm
betahat = solve(t(X) %*% X) %*% t(X) %*% Y
# betahat-fit$coefficients
print(betahat)
```

```
##                     [,1]
```

```
## (Intercept) -45.47548356
## area         -0.03233033
## yearc         0.02695857
## location2     0.77713297
## location3     1.72506792
## bath1         0.76280784
## kitchen1      1.13690814
## cheating1     1.76526110
```

Projection matrices: idempotent, symmetric/orthogonal

(Optional - known from TMA4267)

First, we define predictions as $\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta}$, and inserted the ML (and LS) estimate we get $\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

We define the projection matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$$

called the *hat matrix*. This simplifies the notation for the predictions,

$$\hat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$$

so the hat matrix is putting the hat on the response $\mathbf{Y}$.

In addition we define residuals as

$$\hat{\varepsilon} = \mathbf{Y} - \hat{\mathbf{Y}}$$
$$\hat{\varepsilon} = \mathbf{Y} - \mathbf{H}\mathbf{Y} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$$

so we have a second projection matrix

$$\mathbf{I} - \mathbf{H} = \mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$$

Geometry of Least Squares — involving our two projection matrices

(Optional - known from TMA4267)

- Mean response vector: $\mathrm{E}(\mathbf{Y}) = \mathbf{X}\beta$
- As β varies, $\mathbf{X}\beta$ spans the model plane of all linear combinations. I.e. the space spanned by the columns of $\mathbf{X}$: the column-space of $\mathbf{X}$.
- Due to random error (and unobserved covariates), $\mathbf{Y}$ is not exactly a linear combination of the columns of $\mathbf{X}$.
- LS-estimation chooses $\hat{\beta}$ such that $\mathbf{X}\hat{\beta}$ is the point in the column-space of $\mathbf{X}$ that is closes to $\mathbf{Y}$.
- The residual vector $\hat{\varepsilon} = \mathbf{Y} - \hat{\mathbf{Y}} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$ is perpendicular to the column-space of $\mathbf{X}$.
- Multiplication by $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ projects a vector onto the column-space of $\mathbf{X}$.
- Multiplication by $\mathbf{I} - \mathbf{H} = \mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ projects a vector onto the space perpendicular to the column-space of $\mathbf{X}$.

Restricted maximum likelihood estimator for σ^2

$$\hat{\sigma}^2 = \frac{1}{n-p}(\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) = \frac{\text{SSE}}{n-p}$$

In the generalized linear models setting (remember exponential family from Module 1) we will look at the parameter σ^2 as a nuisance parameter = parameter that is not of interest to us. Our our focus will be on the parameters of interest - which will be related to the mean of the response, which is modelled using our covariate - so the regression parameters β are therefore our prime focus.

However, to perform inference we need an estimator for σ^2.

The maximum likelihood estimator for σ^2 is $\frac{\text{SSE}}{n}$, which is found from maximizing the likelihood inserted our estimate of $\hat{\beta}$

$$L(\hat{\beta}, \sigma^2) = (\frac{1}{2\pi})^{n/2}(\frac{1}{\sigma^2})^{n/2}\exp(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}))$$

$$\begin{aligned} l(\hat{\beta}, \sigma^2) &= \ln(L(\hat{\beta}, \sigma^2)) \\ &= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}) \end{aligned}$$

The score vector with respect to σ^2 is

$$\frac{\partial l}{\partial \sigma^2} = 0 - \frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta})$$

Solving $\frac{\partial l}{\partial \sigma^2} = 0$ gives us the estimator

$$\frac{1}{n}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}) = \frac{\text{SSE}}{n}$$

But, this estimator is biased.

To prove this you may use the trace-formula, that is $\text{E}(\mathbf{Y}^T\mathbf{A}\mathbf{Y}) = \text{tr}(\mathbf{A}\text{Cov}(\mathbf{Y})) + \text{E}(\mathbf{Y})^T\mathbf{A}\text{E}(\mathbf{Y})$, and we use that $\text{SSE} = \mathbf{Y}^T(\mathbf{I} - \mathbf{H})\mathbf{Y}$. This was done in class notes from TMA4267 - lecture 10

But, the estimator is *asympotically* unbiased (unbiased when the sample size n increases to infinity).

When an unbiased version is preferred, it is found using *restricted maximum likelihood* (REML). We will look into REML-estimation in Module 7. In our case the (unbiased) REML estimate is

$$\hat{\sigma}^2 = \frac{1}{n-p}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}) = \frac{\text{SSE}}{n-p}$$

The restricted maximum likelihood estimate is used in `lm`.

Q: What does it mean that the REML estimate is unbiased? Where is the estimate $\hat{\sigma}$ in the regression output? (See output from `lm` for the rent index example.)

Properties for the normal linear model

Distribution

To be able to do inference (=make confidence intervals, prediction intervals, test hypotheses) we need to know about the properties of our parameter estimators in the (normal) linear model.

- Least squares and maximum likelihood estimator for β:

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

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

- Restricted maximum likelihood estimator for σ^2:

$$\hat{\sigma}^2 = \frac{1}{n-p}(\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) = \frac{\text{SSE}}{n-p}$$

  with $\frac{(n-p)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p}$.

- Statistic for inference about β_j, c_{jj} is diagonal element j of $(\mathbf{X}^T\mathbf{X})^{-1}$.

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

  This requires that $\hat{\beta}_j$ and $\hat{\sigma}$ are independent (see below).

However, when we work with *large samples* then $n - p$ becomes large and the t distribution goes to a normal distribution, so we may use the standard normal in place of the t_{n-p}.

Asymptotically we have:

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

and

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\tilde{\sigma}} \sim N(0, 1)$$

where $\tilde{\sigma}^2 = \frac{\text{SSE}}{n}$ (the ML estimator).

Q: Pointing forwards: do you see any connection between the covariance matrix of $\hat{\boldsymbol{\beta}}$ and the Fisher information?

Are $\hat{\beta}$ and SSE are independent? (optional)

Independence: Let $\mathbf{X}_{(p\times 1)}$ be a random vector from $N_p(\mu, \mathbf{\Sigma})$. Then $\mathbf{AX}$ and $\mathbf{BX}$ are independent iff $\mathbf{A\Sigma B}^T = \mathbf{0}$.

- $\mathbf{Y} \sim N_n(\mathbf{X}\beta, \sigma^2\mathbf{I})$
- $\mathbf{AY} = \hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, and
- $\mathbf{BY} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$.
- Now $\mathbf{A}\sigma^2\mathbf{I}\mathbf{B}^T = \sigma^2\mathbf{A}\mathbf{B}^T = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I} - \mathbf{H}) = \mathbf{0}$
- since $\mathbf{X}(\mathbf{I} - \mathbf{H}) = \mathbf{X} - \mathbf{HX} = \mathbf{X} - \mathbf{X} = \mathbf{0}$.
- We conclude that $\hat{\beta}$ is independent of $(\mathbf{I} - \mathbf{H})\mathbf{Y}$,

- and, since SSE=function of $(\mathbf{I}-\mathbf{H})\mathbf{Y}$: SSE=$\mathbf{Y}^T(\mathbf{I}-\mathbf{H})\mathbf{Y}$,
- then $\hat{\beta}$ and SSE are independent, and the result with T_j being t-distributed with $n-p$ degrees of freedom is correct.

Remark: a similar result will exist for GLMs, using the concept of *orthogonal parameters*.

Checking model assumptions

In the normal linear model we have made the following assumptions.

1. Linearity of covariates: $\mathbf{Y}=\mathbf{X}\beta+\varepsilon$. Problem: non-linear relationship?
2. Homoscedastic error variance: $\text{Cov}(\varepsilon)=\sigma^2\mathbf{I}$. Problem: Non-constant variance of error terms
3. Uncorrelated errors: $\text{Cov}(\varepsilon_i,\varepsilon_j)=0$.
4. Additivity of errors: $\mathbf{Y}=\mathbf{X}\beta+\varepsilon$
5. Assumption of normality: $\varepsilon\sim N_n(\mathbf{0},\sigma^2\mathbf{I})$

The same assumptions are made when we do things the GLM way for the normal linear model.

In addtion the following might cause problems:

- Outliers
- High leverage points
- Collinearity

General theory on QQ-plots

Read this for yourself. You do not need to understand this in detail, but is useful to have a basic idea why we look for a straight line in a QQ-plot. There is one question about this in the ILw1.

Go to separate R Markdown or html document: QQ–plot as Rmd or QQ–plot as html

---

Residuals

If we assume the normal linear model then we know that the residuals ($n\times 1$ vector)

$$\hat{\varepsilon}=\mathbf{Y}-\hat{\mathbf{Y}}=(\mathbf{I}-\mathbf{H})\mathbf{Y}$$

has a normal (singular) distribution with mean $\text{E}(\hat{\varepsilon})=\mathbf{0}$ and covariance matrix $\text{Cov}(\hat{\varepsilon})=\sigma^2(\mathbf{I}-\mathbf{H})$ where $\mathbf{H}=\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.

This means that the residuals (possibly) have different variance, and may also be correlated.

Our best guess for the error ε is the residual vector $\hat{\varepsilon}$, and we may think of the residuals as *predictions of the errors*. Be aware: don't mix errors (the unobserved) with the residuals ("observed").

But, we see that the residuals are not independent and may have different variance, therefore we will soon define variants of the residuals that we may use to assess model assumptions after a data set is fitted.

Q: How can we say that the residuals can have different variance and may be correlated? Why is that a problem?

We would like to check the model assumptions - we see that they are all connected to the error terms. But, but we have not observed the error terms ε so they can not be used for this. However, we have made "predictions" of the errors - our residuals. And, we want to use our residuals to check the model assumptions.

That is, we want to check that our errors are independent, homoscedastic (same variance for each observation), and not dependent on our covariates - and we want to use the residuals (observed) in place of the errors (unobserved). Then it would have been great if the residuals have these properties when the underlying errors have. To amend our problem we need to try to fix the residual so that they at least have equal variances. We do that by working with *standardized* or *studentized residuals*.

###Standardized residuals:

$$r_i = \frac{\hat{\varepsilon}_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$$

where h_ii is the ith diagonal element of the hat matrix $\mathbf{H}$.

In R you can get the standardized residuals from an `lm`-object (named `fit`) by `rstandard(fit)`.

###Studentized residuals:

$$r_i^* = \frac{\hat{\varepsilon}_i}{\hat{\sigma}_{(i)}\sqrt{1-h_{ii}}}$$

where $\hat{\sigma}_{(i)}$ is the estimated error variance in a model with observation number i omitted. This seems like a lot of work, but it can be shown that it is possible to calculated the studentized residuals directly from the standardized residuals:

$$r_i^* = r_i(\frac{n-p-1}{n-p-r_i^2})^{1/2}$$

In R you can get the studentized residuals from an `lm`-object (named `fit`) by `rstudent(fit)`.

Plotting residuals - and what to do when assumptions are violated?

Some important plots

1. Plot the residuals, r_i^* against the predicted values, $\hat{y}_i$.
   - Dependence of the residuals on the predicted value: wrong regression model?
   - Nonconstant variance: transformation or weighted least squares is needed?
2. Plot the residuals, r_i^*, against predictor variable or functions of predictor variables. Trend suggest that transformation of the predictors or more terms are needed in the regression.

---

3. Assessing normality of errors: QQ-plots and histograms of residuals. As an additional aid a test for normality can be used, but must be interpreted with caution since for small sample sizes the test is not very powerful and for large sample sizes even very small deviances from normality will be labelled as significant.
4. Plot the residuals, r_i^*, versus time or collection order (if possible). Look for dependence or autocorrelation.

Residuals can be used to check model assumptions, and also to *discover outliers*.

Diagnostic plots in R

More information on the plots here: http://data.library.virginia.edu/diagnostic-plots/ and http://ggplot2.tidyverse.org/reference/fortify.lm.html

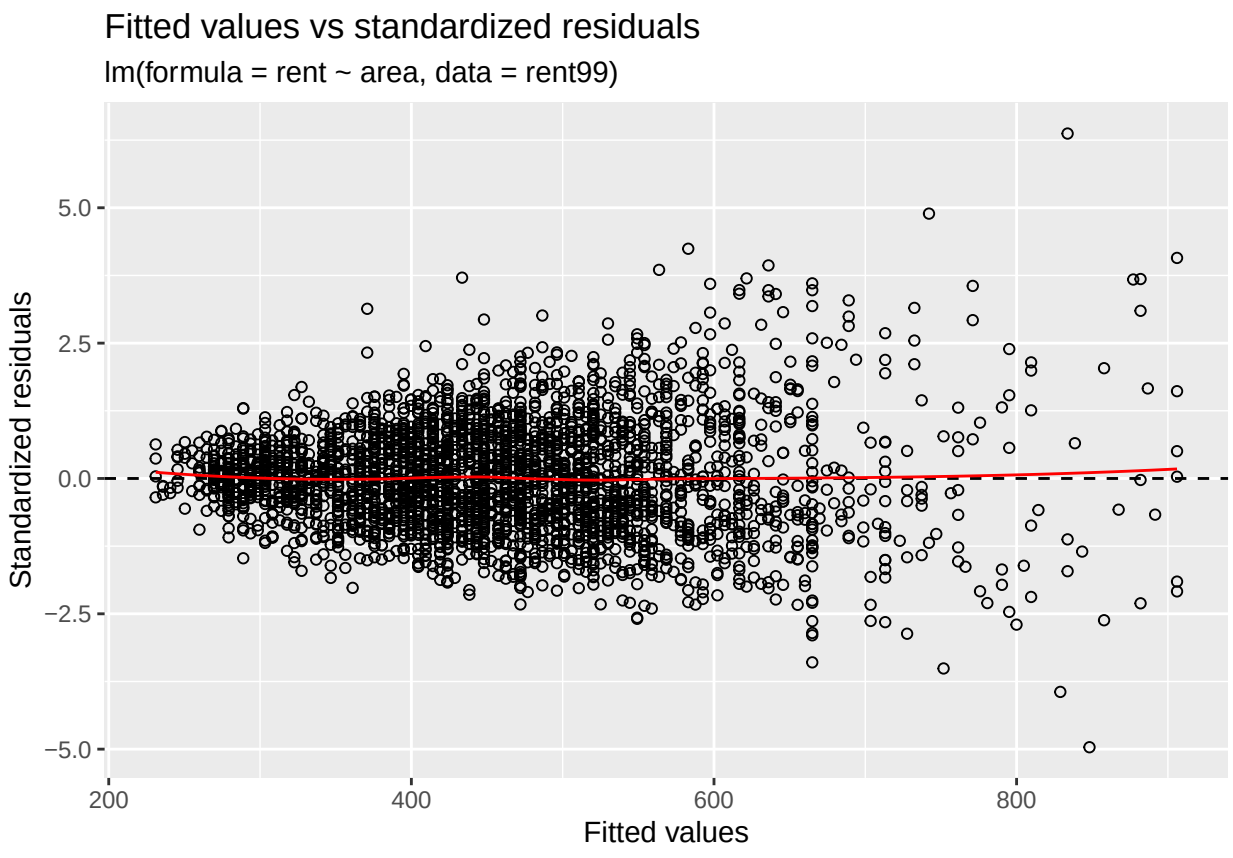
You can use the function `fortify.lm` in `ggplot2` to create a dataframe from an `lm`-object, which `ggplot` uses automatically when given a `lm`-object. This can be used to plot diagnostic plots.

For simplicity we use the Munch rent index with `rent` as response and only `area` as the only covariate. (You may change the model to a more complex one, and rerun the code chunks.)

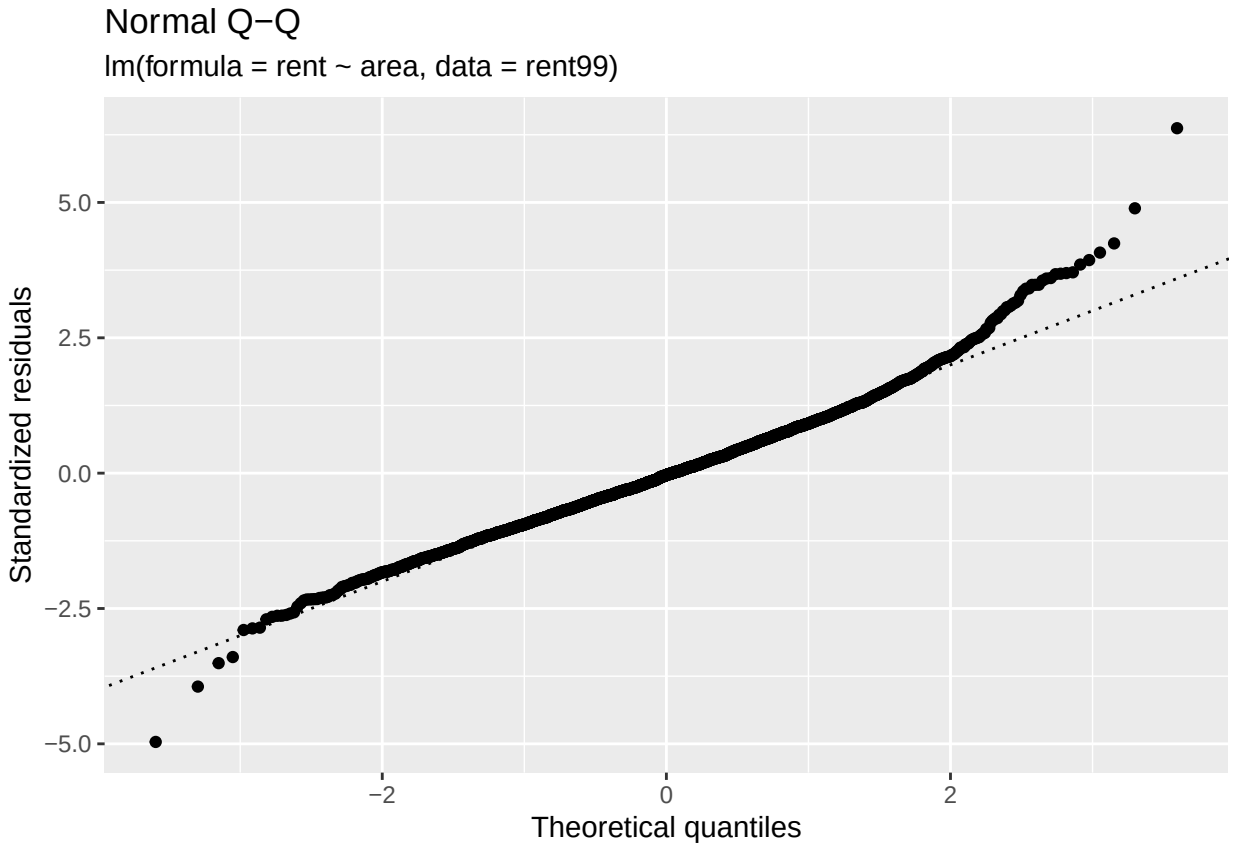
```
##    rent area     .hat .sigma   .cooksd .fitted  .resid .stdresid
## 1 109.9   26 0.001312  158.8 5.870e-04   260.0 -150.00   -0.9454
## 2 243.3   28 0.001219  158.8 1.678e-05   269.6  -26.31   -0.1658
## 3 261.6   30 0.001130  158.8 6.956e-06   279.2  -17.60   -0.1109
## 4 106.4   30 0.001130  158.8 6.711e-04   279.2 -172.83   -1.0891
## 5 133.4   30 0.001130  158.8 4.779e-04   279.2 -145.85   -0.9191
## 6 339.0   30 0.001130  158.8 8.032e-05   279.2   59.79    0.3768
```

Residuals vs fitted values A plot with the fitted values of the model on the x-axis and the residuals on the y-axis shows if the residuals have non-linear patterns. The plot can be used to test the assumption of a linear relationship between the response and the covariates. If the residuals are spread around a horizontal line with no distinct patterns, it is a good indication on no non-linear relationships, and a good model.

Does this look like a good plot for this data set?



Normal Q-Q This plot shows if the residuals are Gaussian (normally) distributed. If they follow a straigt line it is an indication that they are, and else they are probably not.

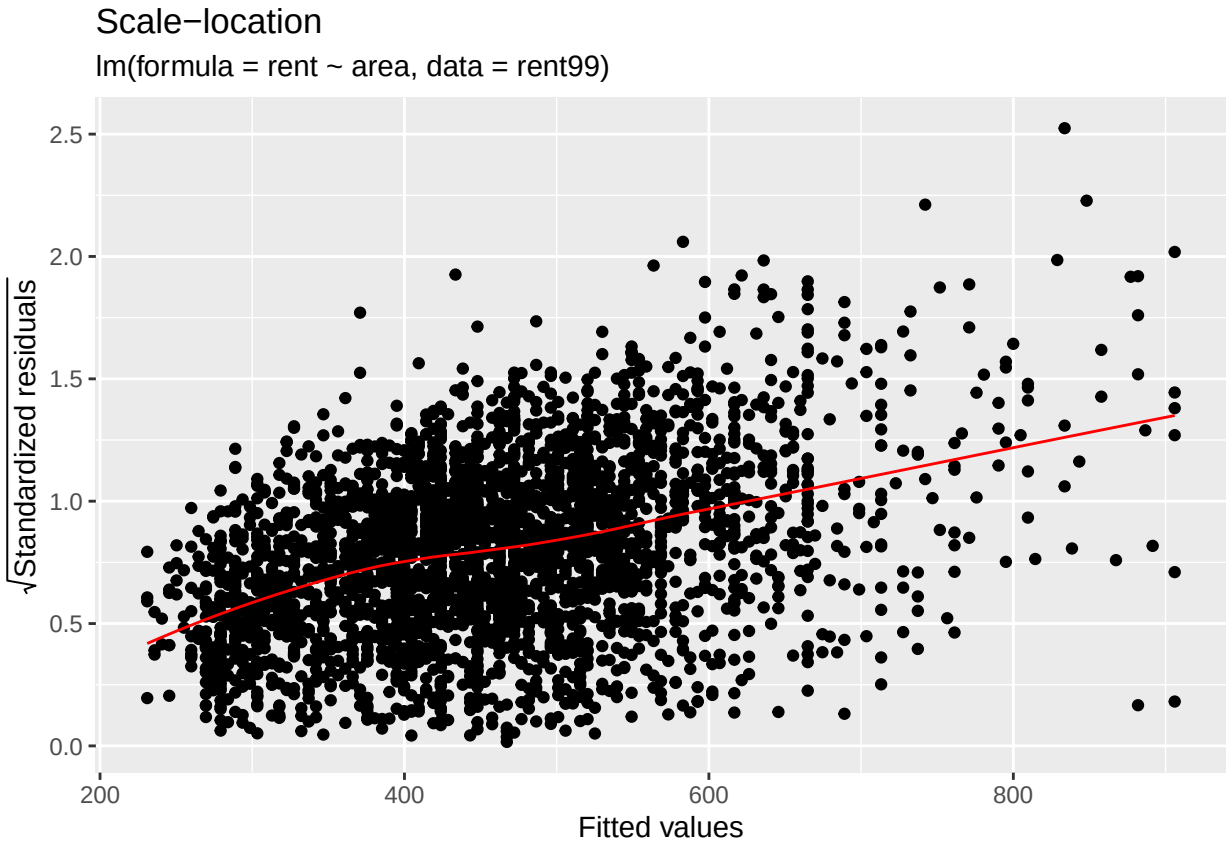


```
library(nortest)
ad.test(rstudent(fit))
```

```
##
##  Anderson-Darling normality test
##
## data:  rstudent(fit)
## A = 6.4123, p-value = 9.809e-16
```

Scale-location This is also called spread-location plot. It shows if the residuals are spread equally along the ranges of predictors. Can be used to check the assumption of equal variance (homoscedasticity). A good plot is one with a horizontal line with randomly spread points.

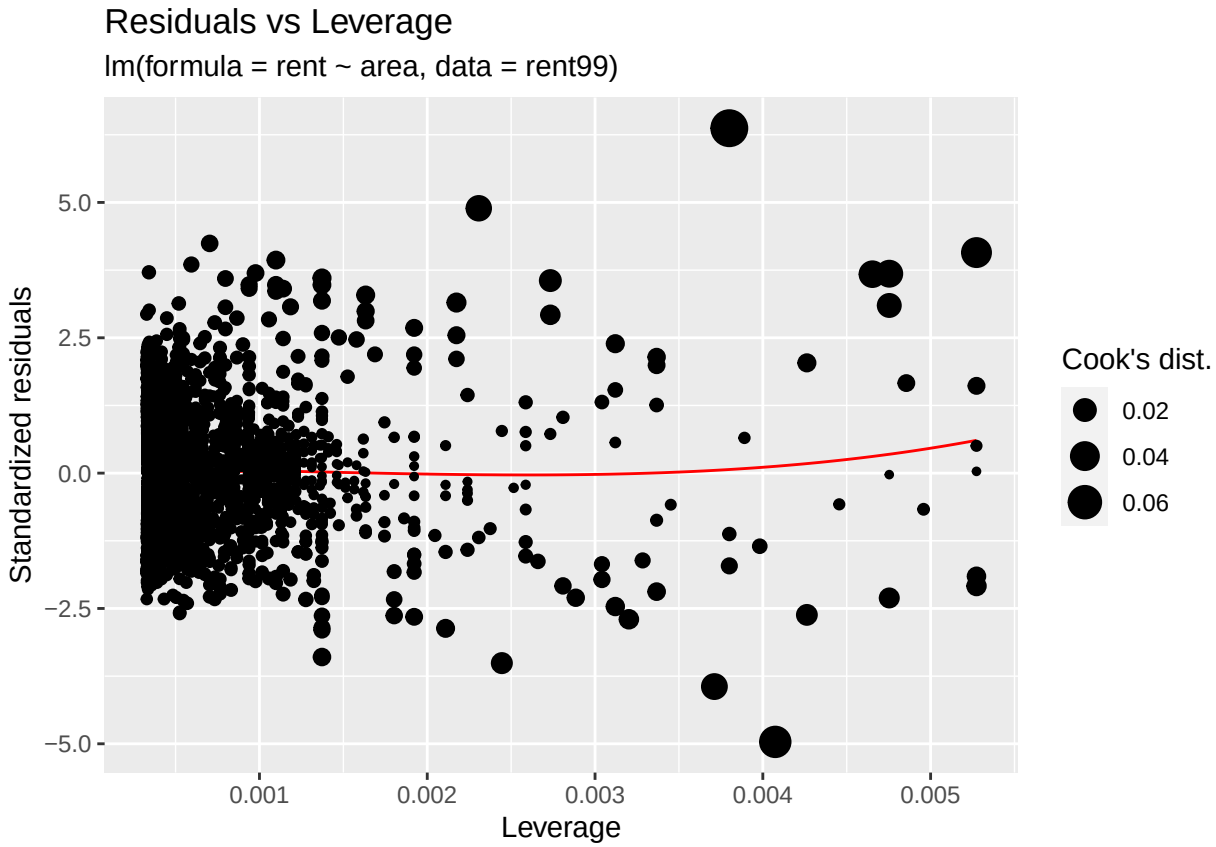
Is this plot good for your data?



Residual vs Leverage This plot can reveal influential outliers. Not all outliers are influential in linear regression; even though data have extreme values, they might not be influential to determine the regression line (the results don't differ much if they are removed from the data set). These influential outliers can be seen as observations that does not get along with the trend in the majority of the observations. In `plot.lm`, dashed lines are used to indicate the Cook's distance, instead of using the size of the dots as is done here.

Cook's distance is the Euclidean distance between the $\hat{\mathbf{y}}$ (the fitted values) and $\hat{\mathbf{y}}_{(i)}$ (the fitted values calculated when the i-th observation is omitted from the regression). This is then a measure on how much the model is influences by observation i. The distance is scaled, and a rule of thumb is to examine observations with Cook's distance larger than 1, and give some attention to those with Cook's distance above 0.5.

Leverage is defined as the diagonal elements of the hat matrix, i.e., the leverage of the i-th data point is h_{ii} on the diagonal of $\mathbf{H} = \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T}$. A large leverage indicated that the observation (i) has a large influence on the estimation results, and that the covariate values $(\mathbf{x}_i)$ are unusual.



(Some observations does not fit our model, but if we fit a more complex model this may change.)

Categorical covariates - dummy and effect coding

(read for yourself - topic of ILw1)

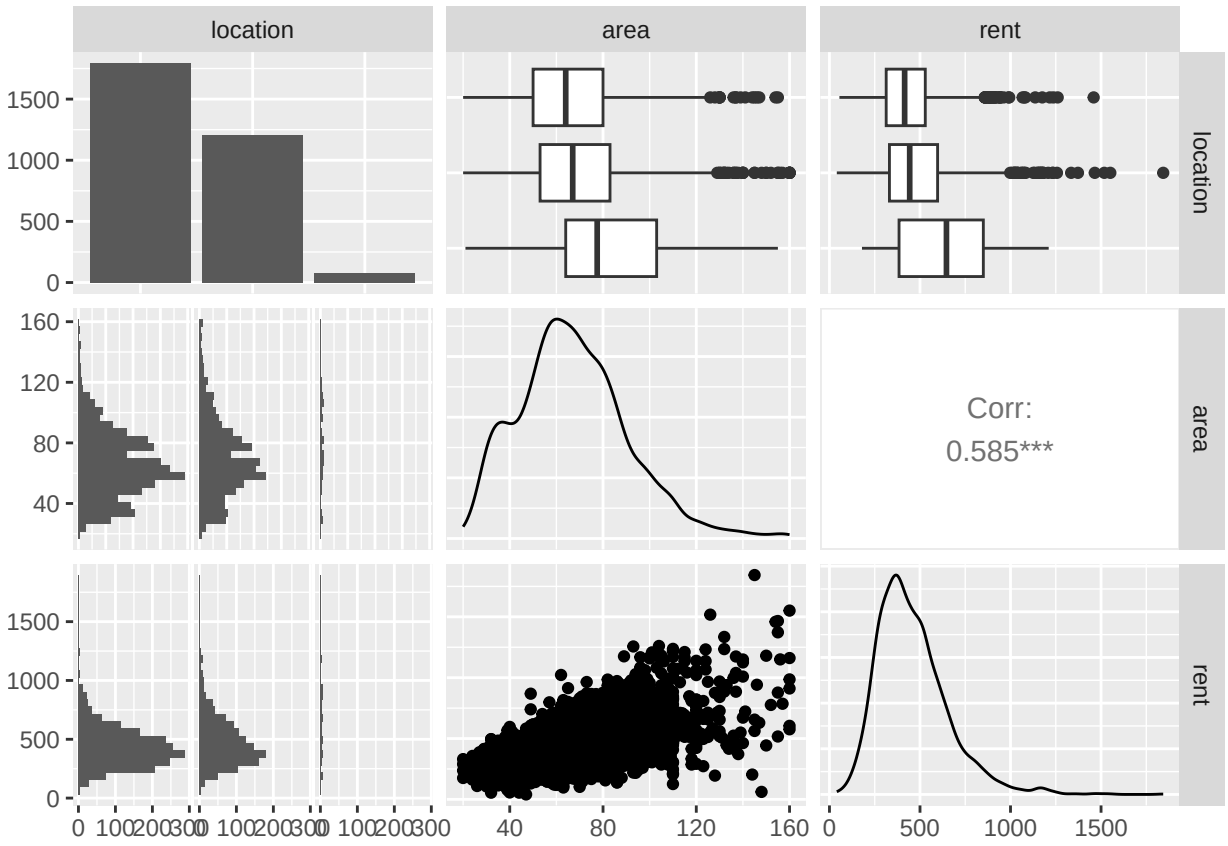
Example: consider our `rent` dataset with `rent` as reponse, and continuous covariate `area` and categorical covariate `location`. Let the `location` be a factor with levels `average, good, excellent`.

```
library(gamlss.data)
library(tidyverse)
library(GGally)
```

```
ds = rent99 %>%
    select(location, area, rent)
levels(ds$location)
```

```
## [1] "1" "2" "3"
```

```
# change to meaningful names
levels(ds$location) = c("average", "good", "excellent")
ggpairs(ds)
```



Q: comment on what you see in the `ggpairs` plot.

Categorical covariates may either be ordered or unordered. We will only consider unordered categories here. In general, we could like to estimate regression coefficients for all levels for the categorical covariates. However, if we want to include an intercept in our model we can only include codings for one less variable than the number of levels we have - or else our design matrix will not have full rank.

Q: Assume you have a categorical variable with three levels. Check for yourself that making a design matrix with one intercept and three columns with dummy (0-1) variable coding will result in a matrix that is singular.

```
# make 'wrong' dummy variable coding with 3 columns
n = length(ds$location)
X = cbind(rep(1, n), ds$area, rep(0, n), rep(0, n), rep(0, n))
X[ds$location == "average", 3] = 1
X[ds$location == "good", 4] = 1
X[ds$location == "excellent", 5] = 1
X[c(1, 3, 69), ]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1   26    0    1    0
## [2,]    1   30    1    0    0
## [3,]    1   55    0    0    1
```

```
library(Matrix)
dim(X)
```

```
## [1] 3082    5
```

```
rankMatrix(X)
```

```
## [1] 4
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.843415e-13
```

This is why we need to instead work with different ways of coding categorical variables. One solution is to not include an intercept in the model, but that is often not what we want. We will look at two other solutions - one where we decide on a reference category (that we not include in the coding, and therefore is kind of included in the intercept - this is called "treatment coding") and one where we require that the the sum of the coeffisients are zero (called "effect coding). This mainly effects how we interpret parameter estimates and communicate our findings to the world.

If we fit a regression model with `lm` to the data with `rent` as response and `area` and `location` as covariates, a model matrix is made - and how to handle the categorical variable is either specified the call to `lm` in `contrasts=list(location="contr.treatment")` (or to model.matrix) or globally for all categorical variables with `options(contrasts=c("contr.treatment","contr.poly"))`- where first element give choice for unordered factor (then treatment contrast is default) and second for ordered (and then this polynomial contrast is default). We will only work with unordered factors now.

–

Dummy variable coding aka treatment contrast

This is the default coding. The reference level is automatically chosen as the "lowest" level (sorted alphabetically). For our example this means that the reference category for location is "average". If we instead wanted "good" to be reference category we could relevel the factor.

```
X1 = model.matrix(~area + location, data = ds)
X1[c(1, 3, 69), ]
```

```
##    (Intercept) area locationgood locationexcellent
## 1            1   26            1                 0
## 3            1   30            0                 0
## 69           1   55            0                 1
```

```
ds$locationRELEVEL = relevel(ds$location, ref = "good")
X2 = model.matrix(~area + locationRELEVEL, data = ds)
X2[c(1, 3, 69), ]
```

```
##    (Intercept) area locationRELEVELaverage locationRELEVELexcellent
## 1            1   26                      0                        0
## 3            1   30                      1                        0
## 69           1   55                      0                        1
```

So, what does this mean in practice? Model 1 has `average` as reference category and model 2 `good`.

```
fit1 = lm(rent ~ area + location, data = ds, contrasts = list(location = "contr.treatment"))
summary(fit1)
```

```
##
## Call:
## lm(formula = rent ~ area + location, data = ds, contrasts = list(location = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -790.98 -100.89   -4.87   94.47 1004.98
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       128.0867     8.6947  14.732  < 2e-16 ***
## area                4.7056     0.1202  39.142  < 2e-16 ***
## locationgood       28.0040     5.8662   4.774 1.89e-06 ***
## locationexcellent 131.1075    18.2614   7.180 8.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.1 on 3078 degrees of freedom
## Multiple R-squared:  0.3555, Adjusted R-squared:  0.3549
## F-statistic:   566 on 3 and 3078 DF,  p-value: < 2.2e-16
```

```
fit2 = lm(rent ~ area + locationRELEVEL, data = ds, contrasts = list(locationRELEVEL = "contr.treatment"))
summary(fit2)
```

```
##
## Call:
## lm(formula = rent ~ area + locationRELEVEL, data = ds, contrasts = list(locationRELEVEL = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -790.98 -100.89   -4.87   94.47 1004.98
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              156.0907     9.4950  16.439  < 2e-16 ***
## area                       4.7056     0.1202  39.142  < 2e-16 ***
## locationRELEVELaverage   -28.0040     5.8662  -4.774 1.89e-06 ***
## locationRELEVELexcellent 103.1034    18.4021   5.603 2.30e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.1 on 3078 degrees of freedom
## Multiple R-squared:  0.3555, Adjusted R-squared:  0.3549
## F-statistic:   566 on 3 and 3078 DF,  p-value: < 2.2e-16
```

Q: Comment on the print-out. How do we interpret the intercept estimate?

Effect coding aka sum-zero-contrast:

This is an equally useful and popular coding - and this is the coding that is preferred when working with analysis of variance in general. The effect coding assumes that the sum of the effects for the levels of the factor sums to zero, and this is done with the following coding scheme (Model 3 with the original location and 4 with the releveled version.)

```
X3 = model.matrix(~area + location, data = ds, contrasts = list(location = "contr.sum"))
X3[c(1, 3, 69), ]
```

```
##    (Intercept) area location1 location2
## 1            1   26         0         1
## 3            1   30         1         0
## 69           1   55        -1        -1
```

```
X4 = model.matrix(~area + locationRELEVEL, data = ds, contrasts = list(locationRELEVEL = "contr.sum"))
X4[c(1, 3, 69), ]
```

```
##    (Intercept) area locationRELEVEL1 locationRELEVEL2
## 1            1   26                1                0
## 3            1   30                0                1
## 69           1   55               -1               -1
```

Observe the coding scheme. This means that when we find "the missing location level estimate" as the negative of the sum of the parameter estimates for the other estimated levels.

So, what does this mean in practice?

```
fit3 = lm(rent ~ area + location, data = ds, contrasts = list(location = "contr.sum"))
summary(fit3)
```

```
##
## Call:
## lm(formula = rent ~ area + location, data = ds, contrasts = list(location = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -790.98 -100.89   -4.87   94.47 1004.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 181.1238    10.6383  17.026  < 2e-16 ***
## area          4.7056     0.1202  39.142  < 2e-16 ***
## location1   -53.0372     6.6428  -7.984 1.98e-15 ***
## location2   -25.0331     6.7710  -3.697 0.000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.1 on 3078 degrees of freedom
## Multiple R-squared:  0.3555, Adjusted R-squared:  0.3549
## F-statistic:   566 on 3 and 3078 DF,  p-value: < 2.2e-16
```

```
fit4 = lm(rent ~ area + locationRELEVEL, data = ds, contrasts = list(locationRELEVEL = "contr.sum"))
summary(fit4)
```

```
##
## Call:
## lm(formula = rent ~ area + locationRELEVEL, data = ds, contrasts = list(locationRELEVEL = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -790.98 -100.89   -4.87   94.47 1004.98
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      181.1238    10.6383  17.026  < 2e-16 ***
## area               4.7056     0.1202  39.142  < 2e-16 ***
## locationRELEVEL1 -25.0331     6.7710  -3.697 0.000222 ***
## locationRELEVEL2 -53.0372     6.6428  -7.984 1.98e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 157.1 on 3078 degrees of freedom
## Multiple R-squared:  0.3555, Adjusted R-squared:  0.3549
## F-statistic:   566 on 3 and 3078 DF,  p-value: < 2.2e-16
```

Q: Comment on the print-out. How do we now interpret the intercept estimate?

Interactions

(read for yourself)

To illustrate how interactions between covariates can be included we use the `ozone` data set from the `ElemStatLearn` library. This data set is measurements from 1973 in New York and contains 111 observations of the following variables:

- `ozone` : ozone concentration (ppm)
- `radiation` : solar radiation (langleys)
- `temperature` : daily maximum temperature (F)
- `wind` : wind speed (mph)

We start by fitting a multiple linear regression model to the data, with `ozone` as our response variable and `temperature` and `wind` as covariates.

ozone	radiation	temperature	wind
41	190	67	7.4
36	118	72	8.0
12	149	74	12.6
18	313	62	11.5
23	299	65	8.6
19	99	59	13.8

```
##
## Call:
## lm(formula = ozone ~ temperature + wind, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.160 -13.209  -3.089  10.588  98.470
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.2008    23.6083  -2.846  0.00529 **
## temperature   1.8265     0.2504   7.293 5.32e-11 ***
## wind         -3.2993     0.6706  -4.920 3.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.72 on 108 degrees of freedom
## Multiple R-squared:  0.5817, Adjusted R-squared:  0.574
```

```
## F-statistic:  75.1 on 2 and 108 DF,  p-value: < 2.2e-16
```

The model can be written as:

$$Y = \beta_0 + \beta_1 x_t + \beta_2 x_w + \varepsilon$$

In this model we have assumed that increasing the value of one covariate is independent of the other covariates. For example: by increasing the `temperature` by one-unit always increases the response value by $\beta_2 \approx 1.651$, regardless of the value of `wind`.

However, one might think that the covariate `wind` (wind speed) might act differently upon `ozone` for different values of `temperature` and vice verse.

$$\begin{aligned} Y &= \beta_0 + \beta_1 x_t + \beta_2 x_w + \beta_3 \cdot (x_t \cdot x_w) + \varepsilon \\ &= \beta_0 + (\beta_1 + \beta_3 x_w) \cdot x_t + \beta_2 x_w + \varepsilon \\ &= \beta_0 + \beta_1 x_t + (\beta_2 + \beta_3 x_t) \cdot x_w + \varepsilon \end{aligned} \quad .$$

We fit this model in `R`. An interaction term can be included in the model using the `*` symbol.

Q: Look at the `summary` below. Is this a better model than without the interaction term? It the term significant?

```
ozone.int = lm(ozone ~ temperature + wind + temperature * wind, data = ozone)
summary(ozone.int)
```

```
##
## Call:
## lm(formula = ozone ~ temperature + wind + temperature * wind,
##     data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.929 -11.190  -3.037   8.209  97.440
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -239.94146   48.59004  -4.938 2.92e-06 ***
## temperature         4.00151    0.59311   6.747 8.02e-10 ***
## wind               13.60882    4.28070   3.179  0.00193 **
## temperature:wind   -0.21747    0.05446  -3.993  0.00012 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.36 on 107 degrees of freedom
## Multiple R-squared:  0.636,  Adjusted R-squared:  0.6258
## F-statistic: 62.31 on 3 and 107 DF,  p-value: < 2.2e-16
```

Below we see that the interaction term is highly significant. The p-value is very small, so that there is strong evidence that $\beta_3 \neq 0$. Furthermore, R^2_{adj} has increased, indicating that more of the variability in the data has been explained by the model (than without the interaction).

Interpretation of the interaction term:

- If we now increase the `temperature` by 10° F, the increase in `wind` speed will be

$$(\hat{\beta}_1 + \hat{\beta}_3 \cdot x_w) \cdot 10 = (4.0 - 0.22 \cdot x_w) \cdot 10 = 40 - 2.2 x_w \text{ units.}$$

- If we increase the `wind` speed by 10 mph, the increase in `temperature` will be

$$(\hat{\beta}_2 + \hat{\beta}_3 \cdot x_t) \cdot 10 = (14 - 0.22 \cdot x_t) \cdot 10 = 140 - 2.2 x_t \text{ units.}$$

The hierarchical principle

It is possible that the interaction term is higly significant, but the main effects are not.

In our `ozone.int` model above: the main effects are `temperature` and `wind`. The hierarchical principle states that if we include an interaction term in our model, the main effects are also to be included, even if they are not significant. This means that if the coefficients $\hat{\beta}_1$ or $\hat{\beta}_2$ would be insignificant, while the coefficient $\hat{\beta}_3$ is significant, $\hat{\beta}_1$ and $\hat{\beta}_2$ should still be included in the model.

There reasons for this is that a model with interaction terms, but without the main effects is hard to interpret.

Interactions between qualitative (discrete) and quantitative (continuous) covariates

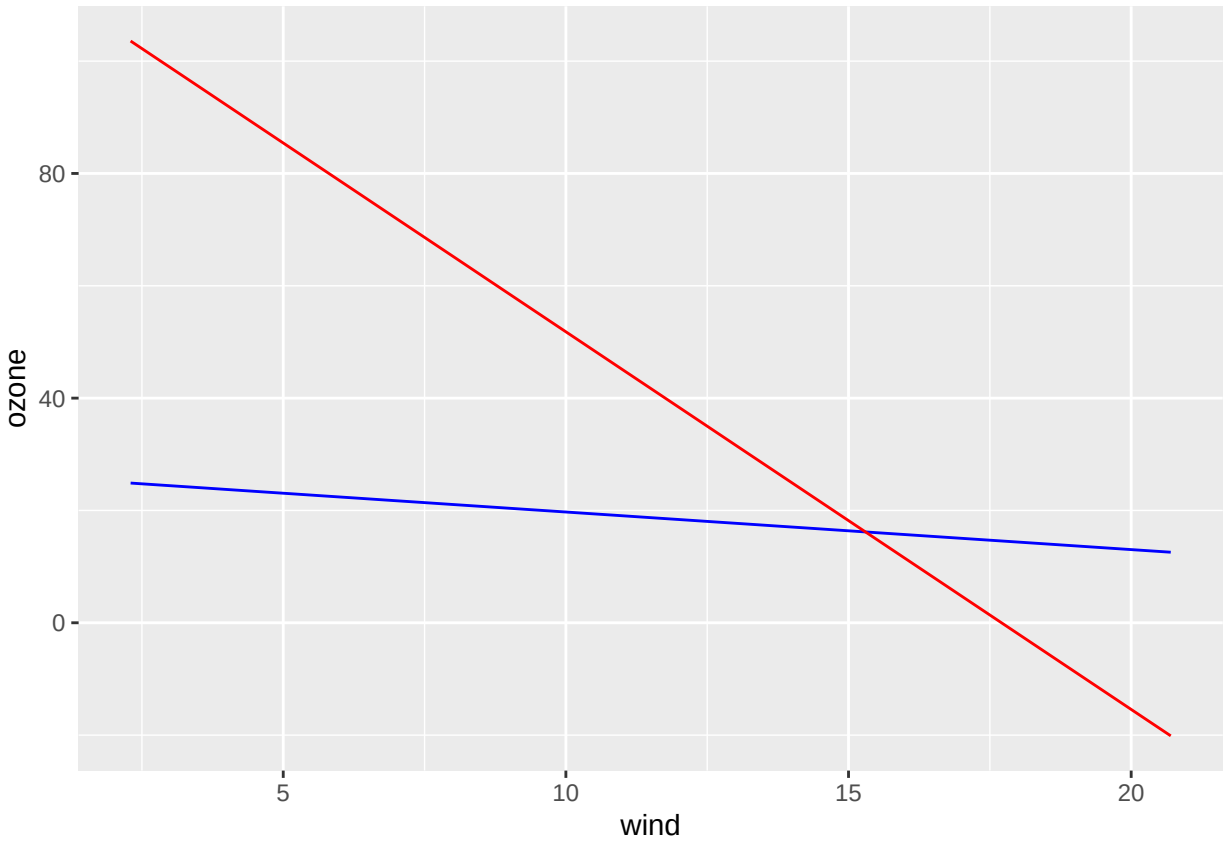
We create a new variable `temp.cat` which is a `temperature` as a qualitative covariate with two levels and fit the model:

$$y = \beta_0 + \beta_1 x_w + \begin{cases} \beta_2 + \beta_3 x_w & \text{if temperature="low"} \\ 0 & \text{if temperature = "high"} \end{cases}$$

$$= \begin{cases} (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \cdot x_w & \text{if temperature="low"} \\ \beta_0 + \beta_1 x_w & \text{if temperature="high""} \end{cases}$$

ozone	radiation	temperature	wind	temp.cat
41	190	67	7.4	low
36	118	72	8.0	low
12	149	74	12.6	low
18	313	62	11.5	low
23	299	65	8.6	low
19	99	59	13.8	low

```
##
## Call:
## lm(formula = ozone ~ wind + temp.cat + temp.cat * wind, data = ozone2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.291  -9.091  -1.307  11.227  71.815
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      119.0450     7.5004  15.872  < 2e-16 ***
## wind              -6.7235     0.8195  -8.204 5.61e-13 ***
## temp.catlow      -92.6316    12.9466  -7.155 1.09e-10 ***
## wind:temp.catlow   6.0544     1.1999   5.046 1.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.26 on 107 degrees of freedom
## Multiple R-squared:  0.6393, Adjusted R-squared:  0.6291
## F-statistic:  63.2 on 3 and 107 DF,  p-value: < 2.2e-16
```



Interactive lectures- problem set first week

Theoretical questions

Problem 1

1. Write down the GLM way for the multiple linear regression model. Explain.
2. Write down the likelihood and loglikelihood. Then define the score vector. What is the set of equations we solve to find parameter estimates? What if we could not find a closed form solution to our set of equations - what could we do then?
3. Define the observed and the expected Fisher information matrix. What dimension does these matrices have? What can these matrices tell us?

---

4. A core finding is $\hat{\beta}$.

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

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

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

5. 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?

6. That is the theoretical intercept and slope of a QQ–plot based on a normal sample? Hint: QQ–plot as html

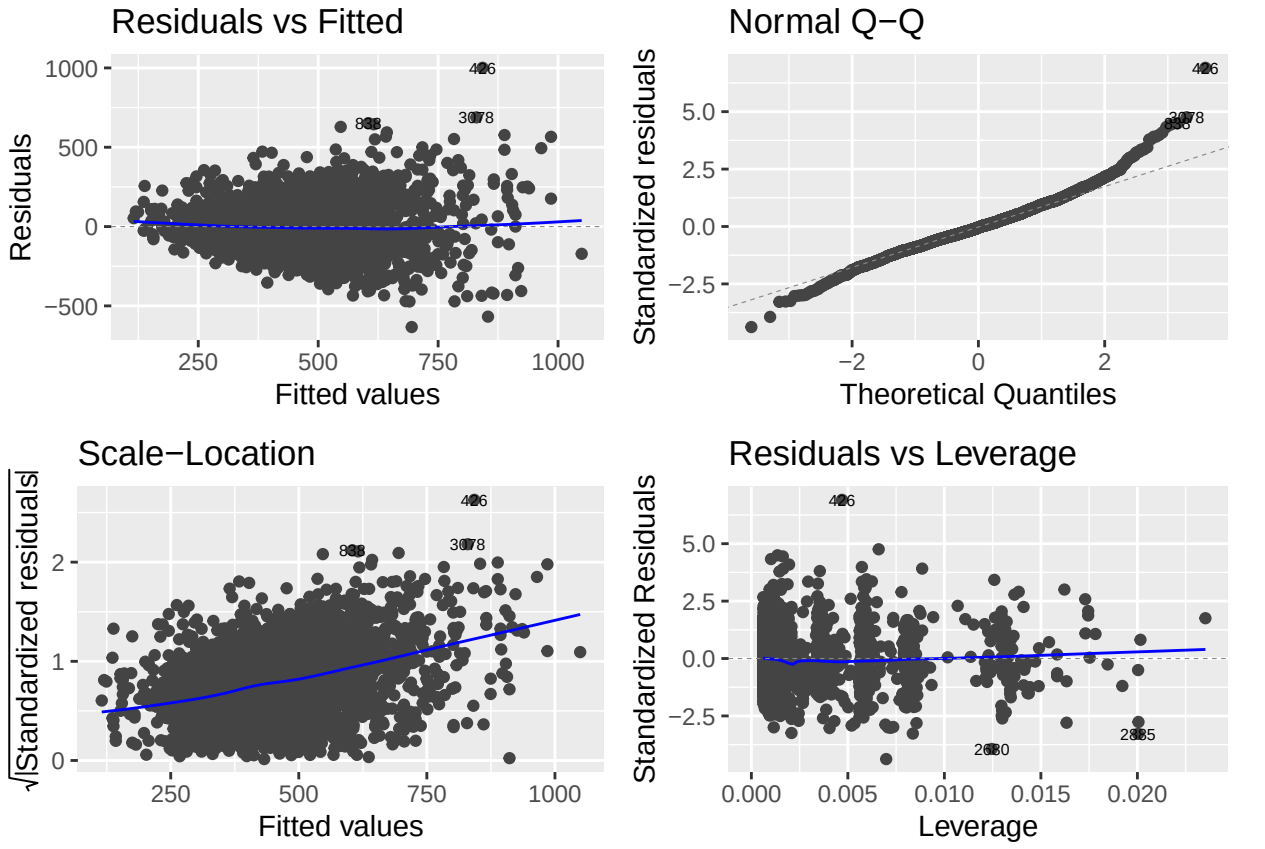
---

Interpretation and understanding

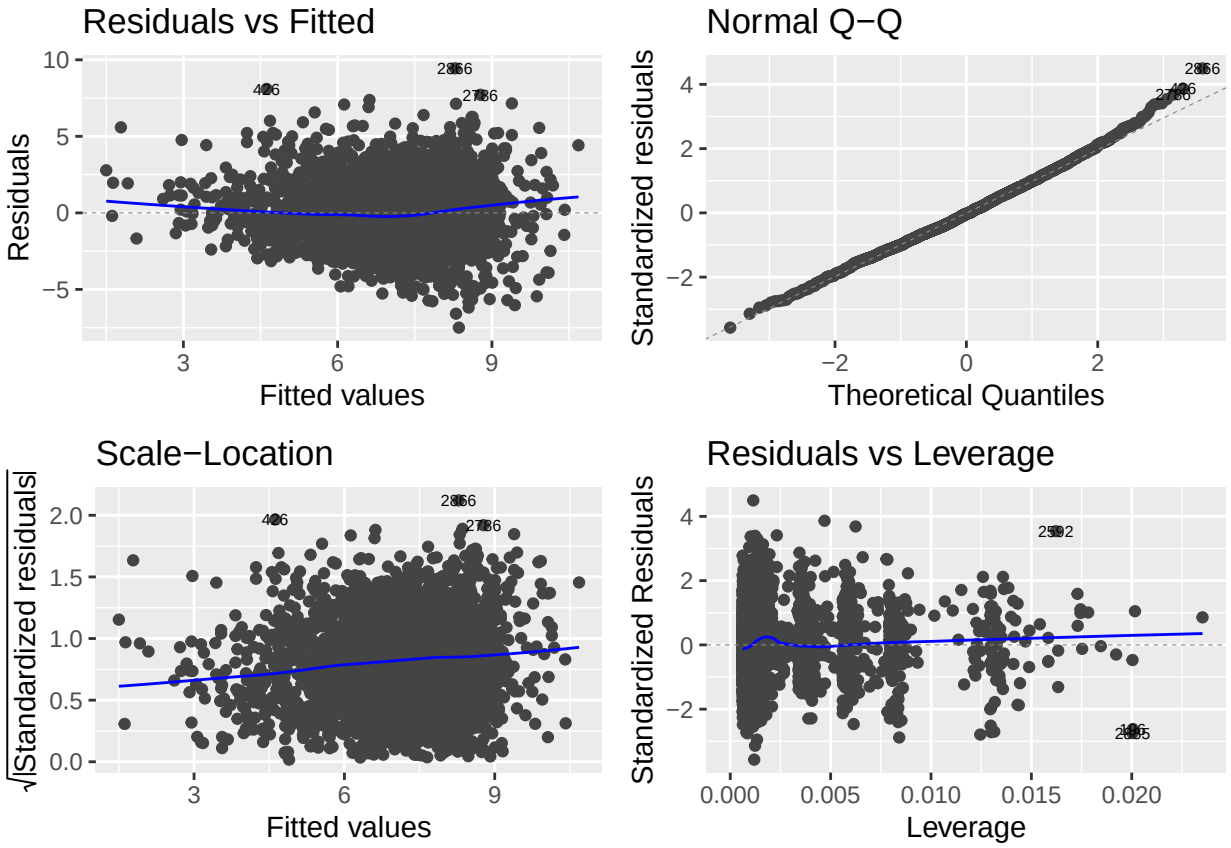
Problem 2: Munich Rent Index data

Fit the regression model with first `rent` and then `rentsqm` as reponse and following covariates: `area`, `location` (dummy variable coding using location2 and location3), `bath`, `kitchen` and `cheating` (central heating).

```
library(gamlss.data)
library(ggfortify)
`?`(rent99)

mod1 <- lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
mod2 <- lm(rentsqm ~ area + location + bath + kitchen + cheating, data = rent99)
autoplot(mod1, label.size = 2)
```




```
autoplot(mod2, label.size = 2)
```



1. Look at diagnostic plots for the two fits. Which response do you prefer?

Consentrate on the response-model you choose for the rest of the tasks.

2. Explain what the parameter estimates mean in practice. In particular, what is the interpretation of the intercept?

```
summary(mod1)
```

```
##
## 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
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = rentsqm ~ area + location + bath + kitchen + cheating,
##     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
```

3. Go through the summary printout and explain the parts you know now, and also observe the parts you don't know yet (on the agenda for next week?).

Next week: more on inference on this data set.

Problem 3: Simple vs. multiple regression

We look at a regression problem where both the response and the covariates are centered - that is, the mean of the response and the mean of each covariate is zero. We do this to avoid the intercept term, which makes things a bit more complicated.

1. In a design matrix (without an intercept column) orthogonal columns gives diagonal $\mathbf{X}^T\mathbf{X}$. What does that mean? How can we get orthogonal columns?
2. If we have orthogonal columns, will then simple (only one covariate) and multiple estimated regression coefficients be different? Explain.
3. What is multicollinearity? Is that a problem? Why (not)?

Problem 4: Dummy vs. effect coding in MLR

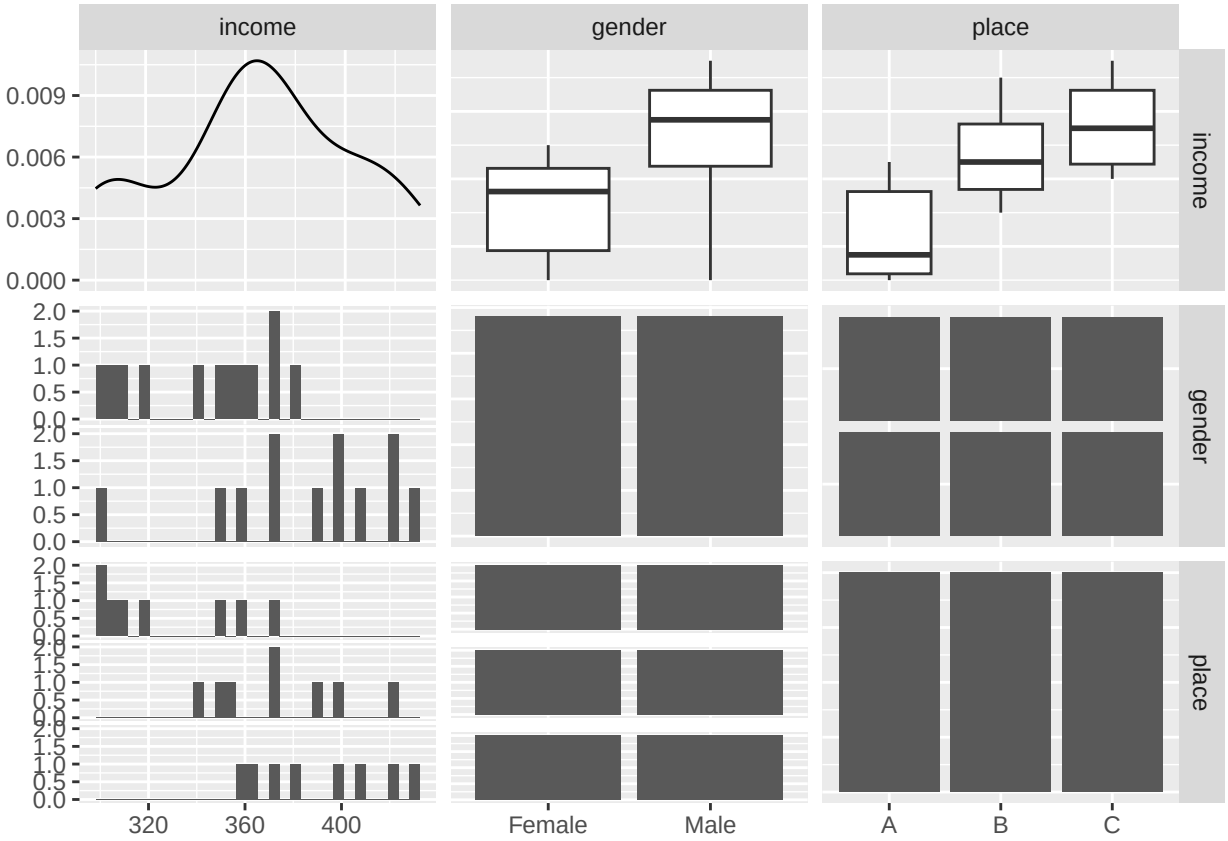
Background material for this task: [Categorical covariates - dummy and effect coding)(#categorical)

We will study a dataset where we want to model `income` as response and two unordered categorical covariates `gender`and `place` (location).

```
income <- c(300, 350, 370, 360, 400, 370, 420, 390, 400, 430, 420, 410,
    300, 320, 310, 305, 350, 370, 340, 355, 370, 380, 360, 365)
gender <- c(rep("Male", 12), rep("Female", 12))
place <- rep(c(rep("A", 4), rep("B", 4), rep("C", 4)), 2)
data <- data.frame(income, gender = factor(gender, levels = c("Female",
    "Male")), place = factor(place, levels = c("A", "B", "C")))
```

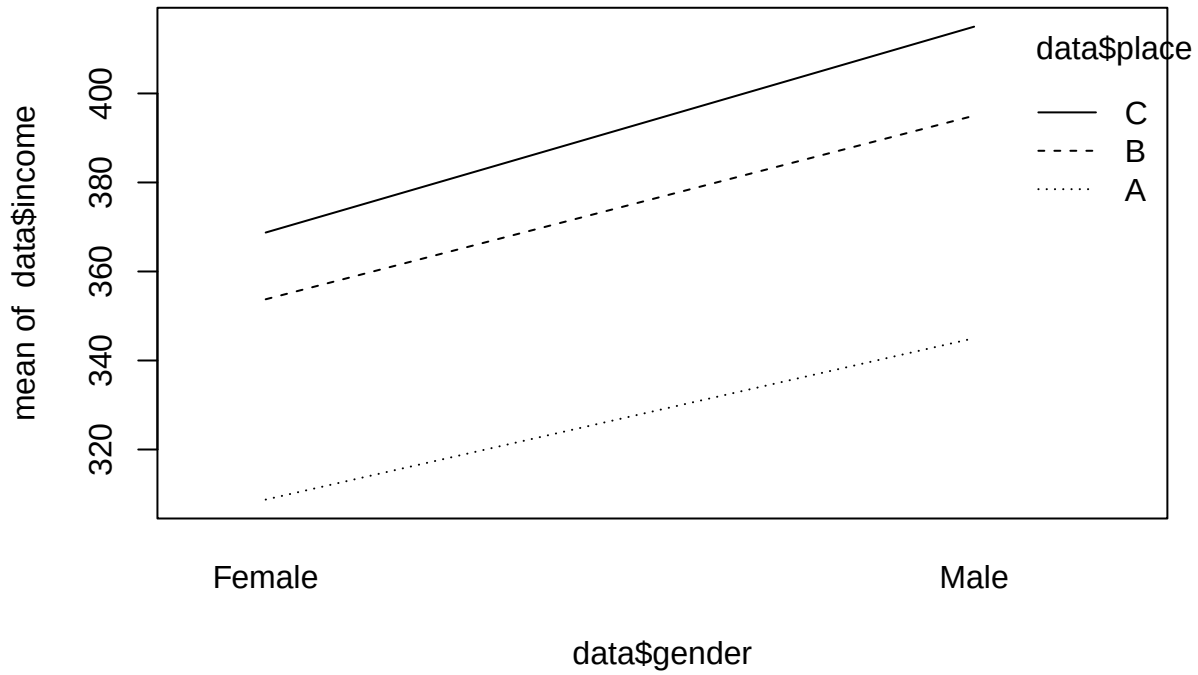
1. First, describe the data set.

```
library(GGally)
GGally::ggpairs(data)
```



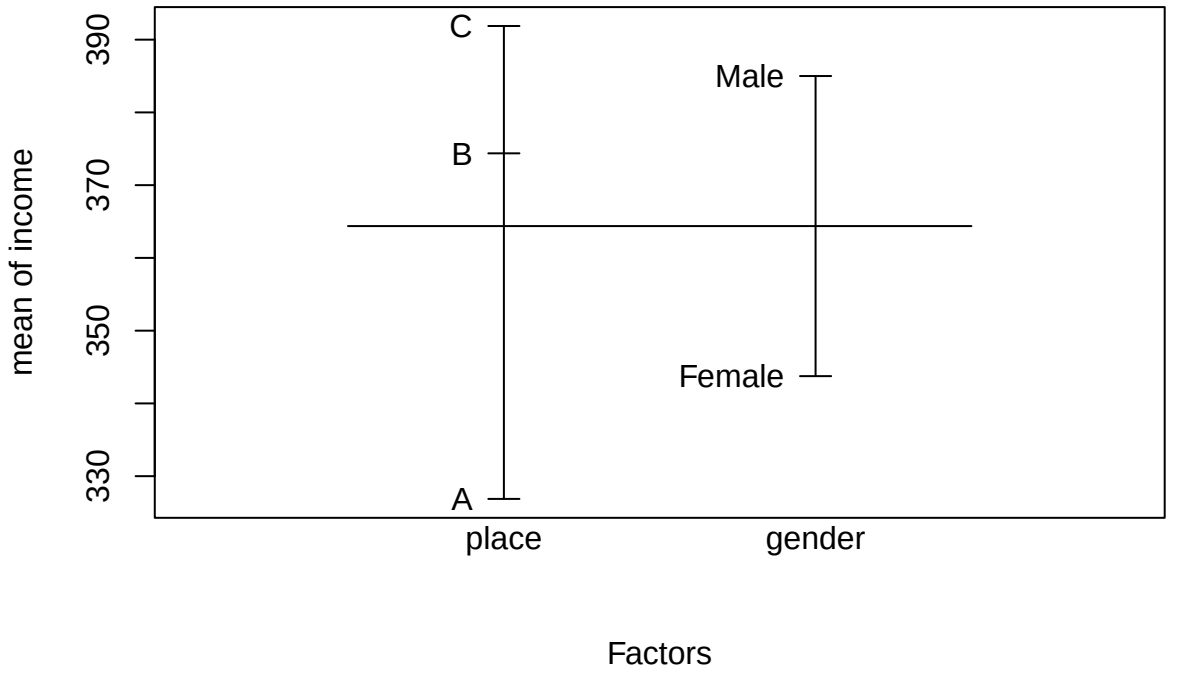
2. Check out the `interaction.plot(data$gender,data$place,data$income)`. What does it show? Do we need an interaction term if we want to model a MLR with `income` as response?

```
interaction.plot(x.factor = data$gender, trace.factor = data$place, response = data$income,
    type = "l")
```



3. Check our `plot.design(income~place+gender, data = data)`. What does it show?

```
plot.design(income ~ place + gender, data = data)
```



4. First, use treatment contrast (dummy variable coding) and fit a MLR with `income` as response and `gender` and `place` as covariates. Explain what your model estimates mean. In particular, what is the interpretation of the intercept estimate?

```
mod3 <- lm(income ~ place + gender, data = data)
mod3
```

```
##
## Call:
```

```
## lm(formula = income ~ place + gender, data = data)
##
## Coefficients:
## (Intercept)       placeB       placeC   genderMale
##      306.25        47.50        65.00        41.25
```

5. Now, turn to sum-zero contrast (effect coding). Explain what your model estimates mean. Now, what is the intercept estimate? Calculate the estimate for `place=C`.

```
mod4 <- lm(income ~ place + gender, data = data, contrasts = list(place = "contr.sum",
    gender = "contr.sum"))
mod4
```

```
##
## Call:
## lm(formula = income ~ place + gender, data = data, contrasts = list(place = "contr.sum",
##     gender = "contr.sum"))
##
## Coefficients:
## (Intercept)       place1       place2      gender1
##      364.38       -37.50        10.00       -20.62
```

```
model.matrix(mod4)
```

```
##    (Intercept) place1 place2 gender1
## 1            1      1      0      -1
## 2            1      1      0      -1
## 3            1      1      0      -1
## 4            1      1      0      -1
## 5            1      0      1      -1
## 6            1      0      1      -1
## 7            1      0      1      -1
## 8            1      0      1      -1
## 9            1     -1     -1      -1
## 10           1     -1     -1      -1
## 11           1     -1     -1      -1
## 12           1     -1     -1      -1
## 13           1      1      0       1
## 14           1      1      0       1
## 15           1      1      0       1
## 16           1      1      0       1
## 17           1      0      1       1
## 18           1      0      1       1
## 19           1      0      1       1
## 20           1      0      1       1
## 21           1     -1     -1       1
## 22           1     -1     -1       1
## 23           1     -1     -1       1
## 24           1     -1     -1       1
## attr(,"assign")
## [1] 0 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$place
## [1] "contr.sum"
##
## attr(,"contrasts")$gender
```

```
## [1] "contr.sum"
```

```
mean(income)
```

```
## [1] 364.375
```

Next week we connect this to linear hypotheses and ANOVA.

Problem 5: Interactions

This part of the module was marked "self-study". Go through this together in the group, and make sure that you understand.

Problem 6: Simulations in R (optional)

(a version this problem was also given as recommended exercise in TMA4268 Statistical learning)

1. For simple linear regression, simulate at data set with homoscedastic errore and with heteroscedastic errors. Here is a suggestion of one solution. Why this? To see how things looks when the model is correct and wrong. Look at the code and discuss what is done, and relate this to the plots of errors (which are usually unobserved) and plots of residuals.

```
# Homoscedastic errors
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))
plot(x, y1, pch = 20)
abline(beta0, beta1, col = 1)
plot(x, e1, pch = 20)
abline(h = 0, col = 2)
plot(x, ehat1, pch = 20)
abline(h = 0, col = 2)

# Heteroscedastic errors
sigma = (0.1 + 0.3 * (x + 3))^2
e2 = rnorm(n, 0, sd = sigma)
y2 = xbeta + e2
ehat2 = residuals(lm(y2 ~ x))
plot(x, y2, pch = 20)
abline(beta0, beta1, col = 2)
plot(x, e2, pch = 20)
abline(h = 0, col = 2)
plot(x, ehat2, pch = 20)
abline(h = 0, col = 2)
```

2. All this fuss about raw, standardized and studentized residuals- does really matter in practice? Below is one example where the raw residuals are rather different from the standardized, but the standardized is identical to the studentized. Can you come up with a simuation model where the standardized and studentized are very different? Hint: what about at smaller sample size?

```r
n = 1000
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)
ehat = residuals(fit)
estand = rstandard(fit)
estud = rstudent(fit)
plot(yhat, ehat, pch = 20)
points(yhat, estand, pch = 20, col = 2)
# points(yhat,estud,pch=19,col=3)
```

SECOND WEEK

What to remember from the first week?

Munich rent index

Munich, 1999: 3082 observations on 9 variables.

- `rent`: the net rent per month (in Euro).
- `rentsqm`: the net rent per month per square meter (in Euro).
- `area`: living area in square meters.
- `yearc`: year of construction.
- `location`: quality of location: a factor indicating whether the location is average location, 1, good location, 2, and top location, 3.
- `bath`: quality of bathroom: a a factor indicating whether the bath facilities are standard, 0, or premium, 1.
- `kitchen`: Quality of kitchen: 0 standard 1 premium.
- `cheating`: central heating: a factor 0 without central heating, 1 with central heating.
- `district`: District in Munich.

More information in Fahrmeir et. al., (2013) page 5.

The GLM way

Independent pairs $(Y_i, \mathbf{x}_i)$ for $i = 1, \ldots, n$.

1. Random component: $Y_i \sim N$ with $\mathrm{E}(Y_i) = \mu_i$ and $\mathrm{Var}(Y_i) = \sigma^2$.
2. Systematic component: $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$.
3. Link function: linking the random and systematic component (linear predictor): Identity link and response function. $\mu_i = \eta_i$.

Likelihood, loglikelihood, score function, observed and expected Fisher information matrix

- Likelihood $L(\beta) = \prod_{i=1}^{n} f(y_i; \beta)$.
- Loglikelihood $l(\beta) = \ln L(\beta)$.

- Score function $s(\beta) = \frac{\partial l(\beta)}{\partial \beta}$. Find ML estimates by solving $s(\hat{\boldsymbol{\beta}}) = \mathbf{0}$.
- Observed $H(\boldsymbol{\beta}) = -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}$ and expected Fisher information $F(\beta) = \mathrm{E}(H(\boldsymbol{\beta}))$

Parameter estimators with properties

- Parameter of interest is β and σ^2 is a nuisance. Maximum likelihood estimator

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

  has distribution: $\hat{\beta} \sim N_p(\beta, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.
- Restricted maximum likelihood estimator for σ^2:

$$\hat{\sigma}^2 = \frac{1}{n-p}(\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) = \frac{\mathrm{SSE}}{n-p}$$

  with $\frac{(n-p)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p}$.

- Statistic for inference about β_j, c_{jj} is diagonal element j of $(\mathbf{X}^T\mathbf{X})^{-1}$.

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

  This requires that $\hat{\beta}_j$ and $\hat{\sigma}$ are independent.
- Asymptotically

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}} \approx N(0,1)$$

Sums of squares of error (SSE): $\mathrm{SSE} = (\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) = \hat{\varepsilon}^T\hat{\varepsilon} = \sum_{i=1}^n (Y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2$.

Parameter estimation in practice

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

Q: How is this done in `lm`?

lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)

Big data

But, what about big data? Big data are characterized by

- volume
- velocity - data collected in a (near) continuous setting
- variation — many types of data: numerical measurements, images, text
- veracity — quality and thrustworthiness
- value — potential in data?

We need analysis tools that are

- efficent from a computational point of view
- large memory capacity

- can be done automatically
- is sensible from a statistics point of view

If the number of observations, n, is large a parallel formulation is valuable.

In the simple case where we want to calculate an average, $\hat{\mu} = \sum_{i=1}^{n} y_i$, we may divide the dataset into G groups (with n_g observations in each group) and calculate sums (or averages) in each group. Group sums: $\hat{\mu}_g = \frac{1}{n_g} \sum_{i:g_i=g} y_i$.

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i = \frac{1}{n} \sum_{g=1}^{G} \sum_{i:g_i=g} y_i = \frac{1}{n} \sum_{g=1}^{G} n_g \hat{\mu}_g$$

This makes it possible to calculate the average in parallell operations and put the result together again.

Q: Can this also be done for $\hat{\beta}$?

Solutions in R

- `lm` requires memory of order $O(np + p^2)$, which causes problems when n is large.
- The solution `biglm` needs memory of the order $O(p^2)$ where computations are performed in blocks.

Remark: for GLM in general we have no closed form solutions to the $s(\hat{\boldsymbol{\beta}}) = \mathbf{0}$ so we will use numerical optimization to handle this, and the ´biglm´ also solves the GLM.

Inference

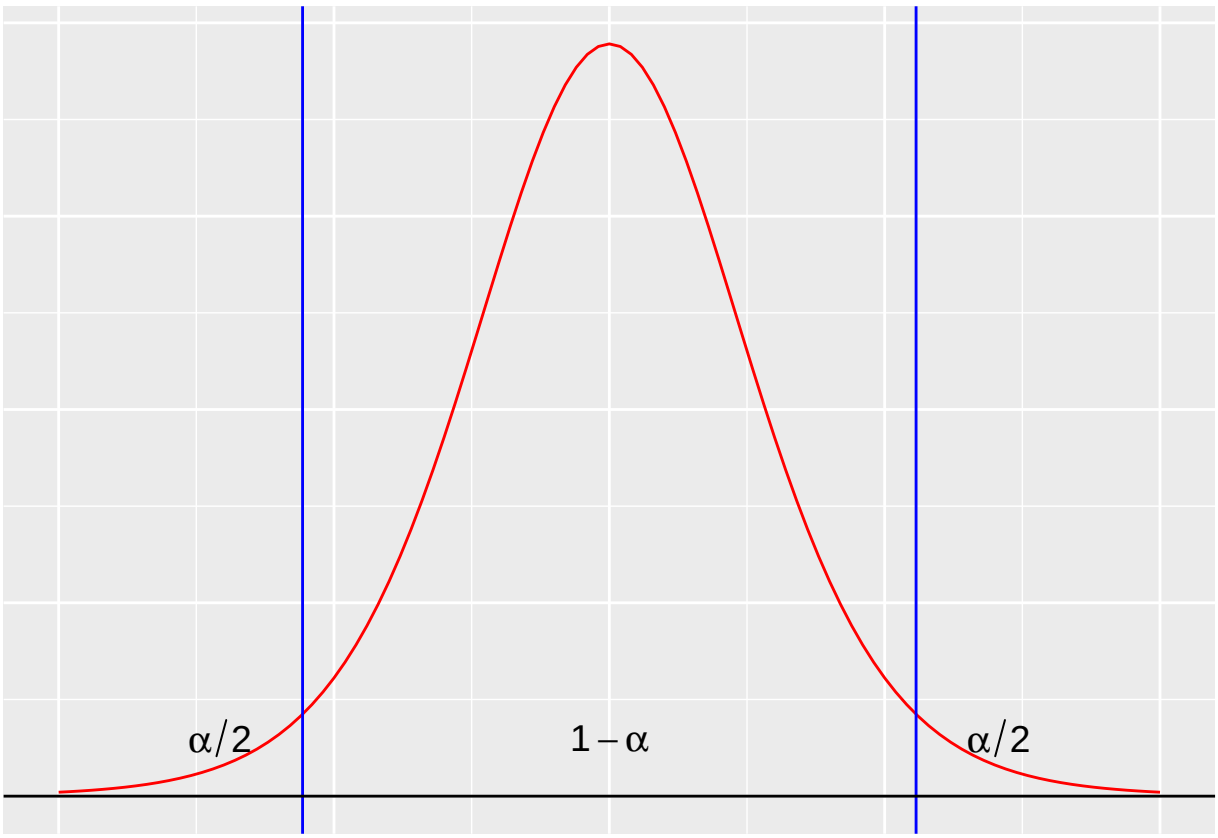
We will consider confidence intervals and prediction intervals, and then test single and linear hypotheses. Most of this should be known to you from earlier regression courses. We will only focus on the results, and you need to read the details in the derivation by yourself.

Confidence intervals (CI)

In addition to providing a parameter estimate for each element of our parameter vector β we should also report a $(1-\alpha)100\%$ confidence interval (CI) for each element. (We will not consider simultanous confidence regions in this course.)

We focus on element j of β, called β_j. It is known that $T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}}$ follows a t-distribution with $n-p$ degrees of freedom. Let $t_{\alpha/2,n-p}$ be such that $P(T_j > t_{\alpha/2,n-p}) = \alpha/2$. REMARK: our textbook would here look at area to the left instead of to the right - but we stick with this notation. Since the t-distribution is symmetric around 0, then $P(T_j < -t_{\alpha/2,n-p}) = \alpha/2$. We may then write

$$P(-t_{\alpha/2,n-p} \leq T_j \leq t_{\alpha/2,n-p}) = 1 - \alpha$$



(Blue lines at $\pm t_{\alpha/2,n-p}$.)

Inserting $T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}}$ and solving so β_j is in the middle gives:

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

A $(1-\alpha)\%$ CI for β_j is when we insert numerical values for the upper and lower limits: $[\hat{\beta}_j - t_{\alpha/2,n-p}\sqrt{c_{jj}}\hat{\sigma}, \hat{\beta}_j + t_{\alpha/2,n-p}\sqrt{c_{jj}}\hat{\sigma}]$.

CIs can be found in R using `confint` on an `lm` object. (Here dummy variable coding is used for `location`, with average as reference location.)

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
confint(fit)
```

```
##                  2.5 %      97.5 %
## (Intercept) -44.825534   0.8788739
## area          4.354674   4.8029443
## location2    28.579849  49.9405909
## location3    92.970636 159.1443278
## bath1        52.076412  96.0311030
## kitchen1     94.907671 145.9621578
## cheating1   144.427555 178.4000215
```

Q (and A):

1. What is the interpretation of a 95% confidence interval?
2. Does the CI for $\hat{\beta}_{\text{area}}$ change if we change the regression model (e.g. not include `cheating`)?
3. How can we in practice find a CI for `location1` (average location) - when that is not printed above? (Yes, may use formula, but in R without maths?)
4. What if we go for an asymptotic confidence interval - what will change?

Prediction intervals

Remember, one aim for regression was to "construct a model to predict the reponse from a set of (one or several) explanatory variables- more or less black box".

Assume we want to make a prediction (of the response - often called Y_0) given specific values for the covariates - often called $\mathbf{x}_0$. An intuitive point estimate is $\hat{Y}_0 = \mathbf{x}_0^T\hat{\beta}$ - but to give a hint of the uncertainty in this prediction we also want to present a prediction interval for the Y_0.

To arrive at such an estimate we start with the difference between the unobserved response Y_0 (for a given covariate vector $\mathbf{x}_0$) and the point prediction $\hat{Y}_0$, $Y_0 - \hat{Y}_0$. First, we assume that the unobserved response at covariate $\mathbf{x}_0$ is independent of our previous observations and follows the same distibution, that is $Y_0 \sim N(\mathbf{x}_0^T\beta, \sigma^2)$. Further,

$$\hat{Y}_0 = \mathbf{x}_0^T\hat{\beta} \sim N(\mathbf{x}_0^T\beta, \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0).$$

Then, for $Y_0 - \mathbf{x}_0^T\hat{\beta}$ we have

$$\mathrm{E}(Y_0 - \mathbf{x}_0^T\hat{\beta}) = 0 \text{ and } \mathrm{Var}(Y_0 - \mathbf{x}_0^T\hat{\beta}) = \mathrm{Var}(Y_0) + \mathrm{Var}(\mathbf{x}_0^T\hat{\beta}) = \sigma^2 + \sigma^2\mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0$$

so that

$$Y_0 - \mathbf{x}_0^T\hat{\beta} \sim N(0, \sigma^2(1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0))$$

—

Inserting our REML-estimate for σ^2 gives

$$T = \frac{Y_0 - \mathbf{x}_0^T\hat{\beta}}{\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}} \sim t_{n-p}.$$

Then, we start with

$$P(-t_{\alpha/2,n-p} \leq \frac{Y_0 - \mathbf{x}_0^T\hat{\beta}}{\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}} \leq t_{\alpha/2,n-p}) = 1 - \alpha$$

and solve so that Y_0 is in the middle, which gives

$$P(\mathbf{x}_0^T\hat{\beta} - t_{\alpha/2,n-p}\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0} \leq Y_0 \leq \mathbf{x}_0^T\hat{\beta} + t_{\alpha/2,n-p}\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}) = 1 - \alpha$$

A $(1-\alpha)$% PI for Y_0 is when we insert numerical values for the upper and lower limits: $[\mathbf{x}_0^T\hat{\beta} - t_{\alpha/2,n-p}\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}, \mathbf{x}_0^T\hat{\beta} + t_{\alpha/2,n-p}\hat{\sigma}\sqrt{1 + \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0}]$.

PIs can be found in R using `predict` on an `lm` object, but make sure that `newdata` is a `data.frame` with the same names as the original data. We want to predict the rent - with PI - for an appartment with area 50, location 2 ("good"), nice bath and kitchen and with central heating.

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
newobs = rent99[1, ]
newobs[1, ] = c(NA, NA, 50, NA, 2, 1, 1, 1, NA)
predict(fit, newdata = newobs, interval = "prediction", type = "response")
```

```
##        fit      lwr      upr
## 1 602.1298 315.5353 888.7243
```

Q (and A):

1. When is a prediction interval of interest?
2. Explain the result from `predict` above.
3. What is the interpretation of a 95% prediction interval?
4. What will change if want an asymptotic interval?

Single hypothesis testing set-up

In single hypothesis testing we are interesting in testing one null hypothesis against an alternative hypothesis. In linear regression the hypothesis is often about a regression parameter β_j:

$$H_0 : \beta_j = 0 \text{ vs. } H_1 : \beta_j \neq 0$$

Remark: we implicitly say that our test is done given that the other variables are present in the model, that is, the other β_is ($j \neq i$) are not zero.

###Two types of errors:

- "Reject H_0 when H_0 is true"="false positives" = "type I error" ="miscarriage of justice". These are our *fake news*, which are very important for us to avoid.
- "Fail to reject H_0 when H_1 is true (and H_0 is false)"="false negatives" = "type II error"= "guilty criminal go free".

We choose to reject H_0 at some significance level α if the p-value of the test (see below) is smaller than the chosen significance level. We say that : Type I error is "controlled" at significance level α, which means that the probability of miscarriage of justice (Type I error) does not exceed α.

Q: Draw a 2 by 2 table showing the connection between

- "truth" (H_0 true or H_0 false) - rows in the table, and
- "action" (reject H_0 and accept H_0) - columns in the table,

and place the two types of errors in the correct position within the table.

What else should be written in the last two cells?

Hypothesis test on β_j (t-test)

In linear regression models our test statistic for testing $H_0 : \beta_j = 0$ is

$$T_0 = \frac{\hat{\beta}_j - 0}{\sqrt{c_{jj}}\hat{\sigma}} \sim t_{n-p}$$

where $c_{jj}\hat{\sigma}^2 = \widehat{\text{Var}}(\hat{\beta}_j)$.

Inserted observed values (and estimates) we have t_0.

We would in a two-sided setting reject H_0 for large values of abs(t_0). We may rely on calculating a p-value.

Q: what if we want an asymptotic test statistics?

The p-value

A p-value is a test statistic satisfying $0 \leq p(\mathbf{Y}) \leq 1$ for every vector of observations $\mathbf{Y}$.

- Small values give evidence that H_1 is true.
- In single hypothesis testing, if the p-value is less than the chosen significance level (chosen upper limit for the probability of committing a type I error), then we reject the null hypothesis, H_0. The chosen significance level is often referred to as α.
- A p-value is *valid* if
$$P(p(\mathbf{Y}) \leq \alpha) \leq \alpha$$
for all α, $0 \leq \alpha \leq 1$, whenever H_0 is true, that is, if the p-value is valid, rejection on the basis of the p-value ensures that the probability of type I error does not exceed α.
- If $P(p(\mathbf{Y}) \leq \alpha) = \alpha$ for all α, $0 \leq \alpha \leq 1$, the p-value is called an *exact* p-value.

In our linear regression we use the t-distibution to calculate p-values for our two-sided test situation $H_0 : \beta_j = 0$ vs. $H_1 : \beta_j \neq 0$. Assume we have observed that our test statistic T_0 takes the numerical value t_0. Since the t-distribution is symmetric around 0 we have

$$p\text{-value} = P(T_0 > \text{abs}(t_0)) + P(T_0 < -\text{abs}(t_0)) = 2 \cdot P(T_0 > \text{abs}(t_0)).$$

We reject H_0 if our calculated p-value is below our chosen signficance level. We often choose as significance level $\alpha = 0.05$.

Q: what if we want an asymptotic p-value?

Munich rent index hypothesis test

We look at print-out using `summary` from fitting `lm`.

```
library(gamlss.data)
colnames(rent99)
```

```
## [1] "rent"     "rentsqm"  "area"     "yearc"    "location" "bath"     "kitchen"
## [8] "cheating" "district"
```

```
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
summary(fit)
```

```
##
## 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
```

Q (and A):

1. Where is hypothesis testing performed here, and which are the hypotheses rejected at level 0.01?
2. Will the test statistics and p-values change if we change the regression model?
3. What is the relationship between performing an hypothesis test and constructing a CI interval? Remember:

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
confint(fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) -44.825534   0.8788739
## area          4.354674   4.8029443
## location2    28.579849  49.9405909
## location3    92.970636 159.1443278
## bath1        52.076412  96.0311030
## kitchen1     94.907671 145.9621578
## cheating1   144.427555 178.4000215
```

Testing linear hypotheses in regression

We study a normal linear regression model with $p = k + 1$ covariates, and refer to this as model A (the larger model). We then want to investigate the null and alternative hypotheses of the following type(s):

$$\begin{aligned} H_0 : \beta_j &= 0 \text{ vs. } H_1 : \beta_j \neq 0 \\ H_0 : \beta_1 &= \beta_2 = \beta_3 = 0 \text{ vs. } H_1 : \text{ at least one of these } \neq 0 \\ H_0 : \beta_1 &= \beta_2 = \cdots = \beta_k = 0 \text{ vs. } H_1 : \text{ at least one of these } \neq 0 \end{aligned}$$

We call the restricted model (when the null hypotesis is true) model B, or the smaller model.

These null hypotheses and alternative hypotheses can all be rewritten as a linear hypothesis

$$H_0 : \mathbf{C}\beta = \mathbf{d} \text{ vs. } \mathbf{C}\beta \neq \mathbf{d}$$

by specifying $\mathbf{C}$ to be a $r \times p$ matrix and $\mathbf{d}$ to be a column vector of length r.

The test statistic for performing the test is called F_{obs} and can be formulated in two ways:

$$F_{obs} = \frac{\frac{1}{r}(SSE_{H_0} - SSE)}{\frac{SSE}{n-p}} \quad (1)$$

$$F_{obs} = \frac{1}{r}(\mathbf{C}\hat{\beta} - \mathbf{d})^{\mathrm{T}}[\hat{\sigma}^2\mathbf{C}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{C}^{\mathrm{T}}]^{-1}(\mathbf{C}\hat{\beta} - \mathbf{d}) \quad (2)$$

where SSE is from the larger model A, SSE_{H_0} from the smaller model B, and $\hat{\beta}$ and $\hat{\sigma}^2$ are estimators from the larger model A.

Testing a set of parameters - what is C and d?

We consider a regression model with intercept and five covariates, $x_1, \ldots, x_5$. Assume that we want to know if the covariates x_3, x_4, and x_5 can be dropped (due to the fact that none of the corresponding β_js are different from zero). This means that we want to test:

$$H_0 : \beta_3 = \beta_4 = \beta_5 = 0 \text{ vs. } H_1 : \text{ at least one of these } \neq 0$$

This means that our $\mathbf{C}$ is a 3×6 matrix and $\mathbf{d}$ a 3×1 column vector

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{d} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Testing one regression parameter

If we set $\mathbf{C} = (0, 1, 0, \cdots, 0)^T$, a row vector with 1 in position 2 and 0 elsewhere, and $\mathbf{d} = (0, 0, \ldots, 0)$, a column vector with 0s, then we test

$$H_0 : \beta_1 = 0 \text{ vs. } H_1 : \beta_1 \neq 0.$$

Now $\mathbf{C}\hat{\beta} = \beta_1$ and $\mathbf{C}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{C}^{\mathrm{T}} = c_{11}$, so that F_{obs} then is equal to the square of the t-statistics for testing a single regression parameter.

$$F_{obs} = (\hat{\beta}_1 - 0)^T[\hat{\sigma}^2 c_{jj}]^{-1}(\hat{\beta}_1 - 0) = T_1^2$$

Repeat the argument with β_j instead of β_1.

Remark: Remember that $T_\nu^2 = F_{1,\nu}$.

Testing "significance of the regression"

If we set $\mathbf{C} = (0, 1, 1, \cdots, 1)^T$, a row vector with 0 in position 1 and 0 elsewhere, and $\mathbf{d} = 0$, a scalar, then we test

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \text{ vs. } H_1 : \text{at least one different from zero.}$$

This means we test if at least one of the regression parameters (in addition to the intercept) is different from 0. The small model is then the model with only the intercept, and for this model the SSE_{H_0} is equal to SST

(sums of squares total, see below). Let SSE be the sums-of-squares of errors for the full model. If we have k regression parameters (in addition to the intercept) then the F-statistic becomes

$$F_{obs} = \frac{\frac{1}{k}(\text{SST} - \text{SSE})}{\frac{\text{SSE}}{n-p}}$$

with k and $n-p$ degrees of freedom under H_0.

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
summary(fit)
```

```
##
## 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
```

Q (and A): Is the regression significant?

Relation to Wald test

Since $\text{Cov}(\hat{\beta}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$, then $\text{Cov}(\mathbf{C}\hat{\beta}) = \mathbf{C}\sigma^2(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{C}^T$, so that $\mathbf{C}\hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{C}^T$ can be seen as an estimate of $\text{Cov}(\mathbf{C}\hat{\beta})$. Therefore, F_{obs} can be written

$$F_{obs} = \frac{1}{r}(\mathbf{C}\hat{\beta} - \mathbf{d})^{\mathrm{T}}[\widehat{\text{Cov}}(\mathbf{C}\hat{\beta})]^{-1}(\mathbf{C}\hat{\beta} - \mathbf{d}) = \frac{1}{r}W$$

where W is a socalled Wald test. It is known that $W \sim \chi^2_r$ asymptotically as n becomes large. We will study the Wald test in more detail later in this course.

Asymptotic result

It can in general be shown that

$$rF_{r,n-p} \overset{n\to\infty}{\longrightarrow} \chi^2_r.$$

That is, if we have a random variable F that is distributed as Fisher with r (numerator) and $n-p$ (denominator) degrees of freedom, then when n goes to infinity (p kept fixed), then rF is approximately χ^2-distributed with r degrees of freedom.

Also, if our error terms are not normally distributed then we can assume that when the number of observation becomes very large then $rF_{r,n-p}$ is approximately χ^2_r.

Introducing deviance

The deviance will replace the SSE (sums of squares of errors, aka residual sums of squares) in the GLM setting, and now we take a first look at the deviance, but to do that we first look at the likelihood ratio test.

The likelihood ratio test

An alternative to the Wald test (based on the F-test shown previously) is the likelihood ratio test (LRT), which compares the likelihood of *two models*.

We use the following notation. A: the larger model (this is H_1) and B: the smaller model (under H_0), and the smaller model is nested within the larger model (that is, B is a submodel of A).

- First we maximize the likelihood for model A (the larger model) and find the maximum likelihood parameter estimates $\hat{\beta}_A$ and $\tilde{\sigma}_A$. The maximum likelihood is achieved at this parameter estimate and is denoted $L(\hat{\beta}_A, \tilde{\sigma}_A)$.
- Then we maximize the likelihood for model B (the smaller model) and find the maximum likelihood parameter estimates $\hat{\beta}_B$ and $\tilde{\sigma}_B$. The maximum likelihood is achieved at this parameter estimate and is denoted $L(\hat{\beta}_B, \tilde{\sigma}_B)$.

The likelihood of the larger model (A) will always be larger or equal to the likelihood of the smaller model (B). Why?

The likelihood ratio statistic is defined as

$$-2\ln\lambda = -2(\ln L(\hat{\beta}_B, \tilde{\sigma}_B) - \ln L(\hat{\beta}_A, \tilde{\sigma}_A))$$

(so, -2 times small minus large).

Under weak regularity conditions the test statistic is approximately χ^2-distributed with degrees of freedom equal the difference in the number of parameters in the large and the small model. This is general - and not related to the GLM! More about this result in TMA4295 Statistical Inference!

P-values are calculated in the upper tail of the χ^2-distribution.

Observe: to perform the test you need to fit both the small and the large model.

Notice: *asymptotically* the Wald and likelihood ratio test statistics have the same distribution, but the value of the test statistics might be different.

Example: Munich rent data

- A (larger): model with `area`, `location` and `bath`.
- B (smaller): model with `area` only.

```
library(lmtest)
fitB <- lm(rent ~ area, data = rent99)
fitA <- update(fitB, . ~ . + location + bath)
lrtest(fitB, fitA)
```

```
## Likelihood ratio test
##
## Model 1: rent ~ area
## Model 2: rent ~ area + location + bath
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1   3 -19990
## 2   6 -19923  3 134.34  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fitB, fitA, test = "Chisq")
```

```
## Analysis of Variance Table
##
## Model 1: rent ~ area
## Model 2: rent ~ area + location + bath
##   Res.Df      RSS Df Sum of Sq  Pr(>Chi)
## 1   3080 77646265
## 2   3077 74334393  3   3311872 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fitB, fitA)
```

```
## Analysis of Variance Table
##
## Model 1: rent ~ area
## Model 2: rent ~ area + location + bath
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1   3080 77646265
## 2   3077 74334393  3   3311872 45.697 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observe that the LRT can be performed using `anova` with 'test="Chisq".

Deviance

The *deviance* (new!) is used to assess model fit and also for model choice, and is based on the likelihood ratio test statistic. It is used for all GLMs in general - and replaces using SSE in multiple linear regression.

First: a covariate pattern is a unique combination of the covariates in our model, for continuous covariates we often have n covariate patterns if we have n observations. Let us assume that for now.

Saturated model: If we were to provide a perfect fit to our data. This means that we have $\hat{\mu}_i = y_i$. So, each observation is given its own parameter.

Candidate model: The model that we are investigated can be thought of as a *candidate* model. Then we maximize the likelihood and get $\hat{\beta}$.

The *deviance* is then defined as the likelihood ratio statistic, where we put the saturated model in place of the larger model A and our candidate model in place of the smaller model B:

$$D = -2(\ln L(\text{candidate model}) - \ln L(\text{saturated model}))$$

The deviance compares the proposed model to the saturated model, and then ask "can we use a more parsimonious model to describe the data as well as the most general model does?".

For the MLR it turns out that the deviance is

$$D = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}})^2 = \frac{\text{SSE}}{\sigma^2}$$

This is sometimes called the *scaled deviance* while the *unscaled deviance* is ϕD, where ϕ is the dispersion parameter. For the normal model the unscaled deviance is thus $\sigma^2 D = \text{SSE}$.

Warning: both the scaled and unscaled deviance is referred to as the deviance, and called D in different sources. Our textbook use the scaled version, while R use the unscaled.

Analysis of variance decomposition and coefficient of determination, R^2

Sums-of-squares decomposition

It is possible to decompose the total variability in the data, called SST (sums-of-squares total), into a part that is explained by the regression SSR (sums-of-squares regression), and a part that is not explained by the regression SSE (sums-of-squares error, or really residual).

Let $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, and $\hat{Y}_i = \mathbf{x}_i^T\hat{\beta}$. Then,

$$\begin{aligned}
\text{SST} &= \text{SSR} + \text{SSE} \\
\text{SST} &= \sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \mathbf{Y}^T(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\mathbf{Y} \\
\text{SSR} &= \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 = \mathbf{Y}^T(\mathbf{H} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\mathbf{Y} \\
\text{SSE} &= \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 = \sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \mathbf{Y}^T(\mathbf{I} - \mathbf{H})\mathbf{Y}.
\end{aligned}$$

The proof can be found in Section 3.5 in our text book Regression.

Based on this decomposition we may define the *coefficient of determination* (R^2) as the ratio between SSR and SST, that is

$$R^2 = \text{SSR}/\text{SST} = 1 - \text{SSE}/\text{SST}$$

1. The interpretation of this coefficient is that the closer it is to 1 the better the fit to the data. If $R^2 = 1$ then all residuals are zero - that is, perfect fit to the data.
2. In a simple linear regression the R^2 equals the squared correlation coefficient between the reponse and the predictor. In multiple linear regression R^2 is the squared correlation coefficient between the observed and predicted response.
3. If we have two models M_1 and M_2, where model M_2 is a submodel of model M_1, then

   $$R^2_{M_1} \geq R^2_{M_2}.$$

   This can be explained from the fact that $\text{SSE}_{M_1} \leq \text{SSE}_{M_2}$.

Analysis of variance tables - with emphasis on sequential Type I ANOVA

It is possible to call the function `anova` on an `lm`-object. What does that function do?

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: rent
##             Df   Sum Sq  Mean Sq  F value    Pr(>F)
## area         1 40299098 40299098 1911.765 < 2.2e-16 ***
## location     2  1635047   817524   38.783 < 2.2e-16 ***
## bath         1  1676825  1676825   79.547 < 2.2e-16 ***
## kitchen      1  2196952  2196952  104.222 < 2.2e-16 ***
## cheating     1  7317894  7317894  347.156 < 2.2e-16 ***
## Residuals 3075 64819547    21080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What is produced is a *sequential* table of *the reductions in residual sum of squares (SSE) as each term in the regression formula is added in turn.* This type of ANOVA is often referred to as "Type I" (not to be confused with type I errors).

We can produce the same table by fitting larger and larger regression models.

```
library(gamlss.data)
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
fit0 <- lm(rent ~ 1, data = rent99)
fit1 <- update(fit0, . ~ . + area)
fit2 <- update(fit1, . ~ . + location)
fit3 <- update(fit2, . ~ . + bath)
fit4 <- update(fit3, . ~ . + kitchen)
fit5 <- update(fit4, . ~ . + cheating)
anova(fit0, fit1, fit2, fit3, fit4, fit5, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: rent ~ 1
## Model 2: rent ~ area
## Model 3: rent ~ area + location
## Model 4: rent ~ area + location + bath
## Model 5: rent ~ area + location + bath + kitchen
## Model 6: rent ~ area + location + bath + kitchen + cheating
##   Res.Df       RSS Df Sum of Sq        F    Pr(>F)
## 1   3081 117945363
## 2   3080  77646265  1  40299098 1911.765 < 2.2e-16 ***
## 3   3078  76011217  2   1635047   38.783 < 2.2e-16 ***
## 4   3077  74334393  1   1676825   79.547 < 2.2e-16 ***
## 5   3076  72137441  1   2196952  104.222 < 2.2e-16 ***
## 6   3075  64819547  1   7317894  347.156 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# anova(fit0,fit1) # compare model 0 and 1 - NOT sequential
# anova(fit0,fit5) # compare model 0 and 5 - NOT sequential
```

If we had changed the order of adding the covariates to the model, then our anova table would also change. You might check that if you want.

###Details on the test `anova(fit)`

When running `anova` on one fitted regression the F-test in `anova` is calculated as for "testing linear hypotheses" - but with a slight twist. Our large model is still the full regression model (from the fitted object), but the smaller model is replaced by the *the change from one model to the next*.

Let SSE be the sums-of-squares-error (residual sums of squares) from the full (large, called A) model - this will be our denominator (as always). For our rent example the denominator will be SSE/(n-p)=64819547/3075 (see print-out above).

However, for the numerator we are not comparing one small model with the full (large) one, we are instead looking at the change in SSE between two (smaller) models (calles model B1 and B2). So, now we have in the numerator the difference in SSE between models B1 and B2, scaled with the difference in number of parameters estimated in model B1 and B2 ="number in B2 minus in B1" (which is the same as the difference in degrees of freedom for the two models).

So, B1 could be the model with only intercept, and B2 could be the model with intercept and area. Then we calculate the SSE for model B1 and for model B2, and keep the difference (here 40299098). Then we count the number of parameters in model B1 and model B2 and compute "number in B2 minus in B1" (here 1). We could istead have calculated the number of degrees of freedom in the smaller model minus the number of degrees of freedom for the larger model (which here is 1). Our numerator is then 40299098/1.

This means that the test statistics we use are:

$$F_0 = \frac{\frac{\text{SSE}_{B1}-\text{SSE}_{B2}}{\text{df}_{B1}-\text{df}_{B2}}}{\frac{\text{SSE}_A}{\text{df}_A}}$$

Remark: notice that the denominator is just the $\hat{\sigma^2}$ from the larger model A.

This makes our F-test statistic: $f_0 = \frac{40299098/1}{64819547/3075} = 1911.765$ (remember that we swap from capital to small letters when we insert numerical values).

To produce a p-value to the test that

$$H_0 : \text{"Model B1 and B2 are equally good" vs } H_1 : \text{"Model B2 is better than B1}$$

and then the $F \sim \text{df}_{B1} - \text{df}_{B2}, \text{df}_A$. In our example we compare to an F-distribution with 1 and 3075 degrees of freedom. The p-value is the "probability of observing a test statistic at least as extreme as we have" so we calculate the p-value as $P(F > f_0)$. This gives a p-value that is practically 0.

If you then want to use the asymptotic version (relating to a chi-square instead of the F), then multiply your F-statistic with $\text{df}_{B1} - \text{df}_{B2}$ and relate to a χ^2 distribution with $\text{df}_{B1} - \text{df}_{B2}$ degrees of freedom, where $\text{df}_{B1} - \text{df}_{B2}$ is the difference in number of parameters in models B1 and B2. In our example $\text{df}_{B1} - \text{df}_{B2} = 1$.

For the anova table we do this sequentially for all models from starting with only intercept to the full model A. This means you need to calculate SSE and df for models of all sizes to calculate lots of these F_0s. Assume that we have 4 covariates that are added to the model, and call the 5 possible models (given the order of adding the covariates)

- model 1: model with only intercept
- model 2: model with intercept and covariate 1

- model 3: model with intercept and covariate 1 and covariate 2
- model 4: model with intercept and covariate 1 and covariate 2 and covariate 3
- model 5: model with intercept and covariate 1 and covariate 2 and covariate 3 and covariate 4

Fit a linear model (lm) for each model 1-5, and store SSE and degrees of freedom=df (number of observations minus number of covariates estimated) for each of the models. Call these SSE_1 to SSE_5 and df_1 to *df_5$.

The anova output has columns: Df Sum Sq Mean Sq F value Pr(>F) and one row for each covariate added to the model.

- model 2 vs model 1: Df=df_1-df_2, Sum Sq=SSE_1-SSE_2, Mean Sq=Sum Sq/Df, F value=(Mean Sq)/(SSE_5/df_5)=f_0, Pr(>F)=pvalue=$P(F > f_0)$.
- model 3 vs model 2: Df=df_2-df_3, Sum Sq=SSE_2-SSE_3, Mean Sq=Sum Sq/Df, F value=(Mean Sq)/(SSE_5/df_5)=f_0, Pr(>F)=pvalue=$P(F > f_0)$.
- model 4 vs model 3: Df=df_3-df_4, Sum Sq=SSE_3-SSE_4, Mean Sq=Sum Sq/Df, F value=(Mean Sq)/(SSE_5/df_5)=f_0, Pr(>F)=pvalue=$P(F > f_0)$.
- model 5 vs model 4: Df=df_4-df_5, Sum Sq=SSE_4-SSE_5, Mean Sq=Sum Sq/Df, F value=(Mean Sq)/(SSE_5/df_5)=f_0, Pr(>F)=pvalue=$P(F > f_0)$.

In R the p-value is calculated as `1-pf(f0,Df)` or as `1-pchisq(Df*f0,Df)` if the asymptotic chisquare distribution is used.

So, this is what is presented - a sequential record of the effect of adding a new covariate.

Q: what if you change the order of the covariates into the model? Yes, then everything changes. That is the drawback of Type I (sequential) thinking.

Q: What if one of the covariates is a factor? Then all parameters of the factor (e.g. all dummy variables) are tested in one step (more in interactive lecture Problem 2).

A competing way of thinking is called *type III ANOVA* and instead of looking sequentially at adding terms, we (like in `summary`) calculated the contribution to a covariate (or factor) given that all other covariates are present in the regression model. Type III ANOVA is available from library `car` as function `Anova` (possible to give type of anova as input).

Check : Take a look at the print-out from `summary` and `anova` and observe that for our rent data the p-values for each covariate are different due to the different nature of the H_0s tested (sequential vs. "all other present").

If we had orthogonal columns for our different covariates the type I and type III ANOVA tables would have been equal.

Optional (beyond the scope of this course) There is also something called a type II ANOVA table, but that is mainly important if we have interactions in our model, so we do not consider that here. If you want to read more this blogplot https://www.r-bloggers.com/anova-%E2%80%93-type-iiiiii-ss-explained/ is a good read. And, in combination with different variants of dummy and effect coding - read this: http://rstudio-pubs-static.s3.amazonaws.com/65059_586f394d8eb84f84b1baaf56ffb6b47f.html. A good read is Langsrud (2003): ANOVA for unbalanced data: Use Type II instead of Type III sums of squares.

Pointing ahead

For GLM the sequential analysis of variance (ANOVA) is replace by analysis of deviance.

Model selection

When we do model selection in the GLM course we will focus on the AIC criterion. The other criteria are added for completeness. We do not use hypothesis tests in model selection.

Quality measures

To assess the quality of the regression we can report the R^2 coefficient of determination. However, since adding covariates to the linear regression can not make the SSE larger, this means that adding covariates can not make the R^2 smaller. This means that SSE and R^2 are only useful measures for comparing models with the same number of regression parameters estimated.

If we consider two models with the same model complexity then SSE can be used to choose between (or compare) these models.

But, if we want to compare models with different model complexity we need to look at other measures of quality for the regression.

R^2 adjusted (corrected)

$$R^2_{\text{adj}} = 1 - \frac{\frac{SSE}{n-p}}{\frac{SST}{n-1}} = 1 - \frac{n-1}{n-p}(1-R^2)$$

Choose the model with the *largest* R^2_{adj}.

###AIC Akaike information criterion

AIC is one of the most widely used criteria, and is designed for likelihood-based inference. Let $l(\hat{\beta}_M, \tilde{\sigma}^2)$ be the maximum of the log-likelihood of the data inserted the maximum likelihood estimates for the regression and nuisance parameter. Further, let $|M|$ be the number of estimated regression parameters (coefficients) in our model, and add 1 if we need to estimate a dispersion parameter (like we do for the normal model).

$$\text{AIC} = -2 \cdot l(\hat{\beta}_M, \tilde{\sigma}^2) + 2(|M| + 1)$$

For a normal regression model this can be further elaborated on:

$$\text{AIC} = n \ln(\tilde{\sigma}^2) + 2(|M| + 1) + C$$

where C is a function of n (will be the same for two models for the same data set). Remark that $\tilde{\sigma}^2 = SSE/n$ - our ML estimator (not our unbiased REML), so that the first term in the AIC is just a function of the SSE. For MLR the AIC and the Mallows Cp gives the same result when comparing models.

Choose the model with the minimum AIC.

###BIC Bayesian information criterion.

The BIC is also based on the likelihood (see notation above).

$$\text{BIC} = -2 \cdot l(\hat{\beta}_M, \tilde{\sigma}^2) + \ln(n) \cdot (|M| + 1)$$

For a normal regression model:

$$\text{BIC} = n \ln(\tilde{\sigma}^2) + \ln(n)(|M| + 1)$$

Choose the model with the minimum BIC.

AIC and BIC are motivated in very different ways, but the final result for the normal regression model is very similar. BIC has a larger penalty than AIC ($\log(n)$ vs. 2), and will often give a smaller model (=more parsimonious models) than AIC. In general we would not like a model that is too complex.

Model selection strategies

- All subset selection: use smart "leaps and bounds" algorithm, works fine for number of covariates in the order of 40.
- Forward selection: choose starting model (only intercept), then add one new variable at each step - selected to make the best improvement in the model selection criteria. End when no improvement is made.
- Backward elimination: choose starting model (full model), then remove one new variable at each step - selected to make the best improvement in the model selection criteria. End when no improvement is made.
- Stepwise selection: combine forward and backward.

Interactive tasks for the second week

Problem 1: Theory

1. What is the interpretation of a 95% confidence interval? Hint: repeat experiment (on Y), on average how many CIs cover the true β_j?
2. Explain in words and with formulas the p-values printed in a `summary` from `lm`.

```
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = rent99)
summary(fit)
```

```
##
## 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
```

3. Explain in words and with formulas the full output (with p-values) printed in an `anova` from `lm`.

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: rent
##             Df   Sum Sq  Mean Sq  F value    Pr(>F)
## area         1 40299098 40299098 1911.765 < 2.2e-16 ***
## location     2  1635047   817524   38.783 < 2.2e-16 ***
## bath         1  1676825  1676825   79.547 < 2.2e-16 ***
## kitchen      1  2196952  2196952  104.222 < 2.2e-16 ***
## cheating     1  7317894  7317894  347.156 < 2.2e-16 ***
## Residuals 3075 64819547    21080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. In particular: why does using `summary` and `anova` on a fitted `lm` give different test statistics and different p-values listed for each covariate. And, why is `summary` listing `location2` and `location3` while `anova` is listing `location`?

Optional: Maybe test out `Anova` in library `car` with type 3 ANOVA to compare?

5. Consider a MLR model A and a submodel B (all parameters in B are in A also). We say that B is nested within A. Assume that regression parameters are estimated using maximum likelihood. Why is the following true: the likelihood for model A will always be larger or equal to the likelihood for model B.

6. How do we define the deviance of model A? What is a *saturated model* in our MLR setting? What does our finding in 5. imply for the deviance (can the deviance both be positive and negative)?

Problem 2: Dummy vs. effect coding in MLR (continued)

We have studied the data set with income, place and gender - with focus on dummy variable coding (with different reference levels) and effect coding and the interpretation of parameter estimates. Now we continue with the same data set, but with focus on hypothesis testing (linear hypotheses) and analysis of variance decomposition.

1. Previously, we have read in the data and fitted linear models - look back to see what we found.

```
income <- c(300, 350, 370, 360, 400, 370, 420, 390, 400, 430, 420, 410,
    300, 320, 310, 305, 350, 370, 340, 355, 370, 380, 360, 365)
gender <- c(rep("Male", 12), rep("Female", 12))
place <- rep(c(rep("A", 4), rep("B", 4), rep("C", 4)), 2)
data <- data.frame(income, gender = factor(gender, levels = c("Female",
    "Male")), place = factor(place, levels = c("A", "B", "C")))
```

2. Fit the following model `model = lm(income~place-1,data=data,x=TRUE)`. Here `x=TRUE` tells the function to calculate the design matrix X, which is stored as `model$x`.

```
model = lm(income ~ place - 1, data = data, x = TRUE)
model$x
```

```
##    placeA placeB placeC
## 1       1      0      0
## 2       1      0      0
## 3       1      0      0
## 4       1      0      0
## 5       0      1      0
## 6       0      1      0
## 7       0      1      0
```

```
## 8       0      1      0
## 9       0      0      1
## 10      0      0      1
## 11      0      0      1
## 12      0      0      1
## 13      1      0      0
## 14      1      0      0
## 15      1      0      0
## 16      1      0      0
## 17      0      1      0
## 18      0      1      0
## 19      0      1      0
## 20      0      1      0
## 21      0      0      1
## 22      0      0      1
## 23      0      0      1
## 24      0      0      1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$place
## [1] "contr.treatment"
```

Examine the results with `summary` and `anova`. What parametrization is used? What is the interpretation of the parameters? Which null hypothesis is tested in the anova-call? What is the result of the hypothesis test?

```
summary(model)
```

```
##
## Call:
## lm(formula = income ~ place - 1, data = data, x = TRUE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## placeA  326.875      9.733   33.58   <2e-16 ***
## placeB  374.375      9.733   38.46   <2e-16 ***
## placeC  391.875      9.733   40.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9944
## F-statistic:  1409 on 3 and 21 DF,  p-value: < 2.2e-16
```

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      3 3204559 1068186  1409.4 < 2.2e-16 ***
## Residuals 21   15916     758
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Fit the models:

```
model1 = lm(income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.treatment"))
head(model1$x)
```

```
##   (Intercept) placeB placeC
## 1           1      0      0
## 2           1      0      0
## 3           1      0      0
## 4           1      0      0
## 5           1      1      0
## 6           1      1      0
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  326.875      9.733  33.583  < 2e-16 ***
## placeB        47.500     13.765   3.451 0.002394 **
## placeC        65.000     13.765   4.722 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.4875
## F-statistic: 11.94 on 2 and 21 DF,  p-value: 0.000344
```

```
model2 = lm(income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.sum"))
head(model2$x)
```

```
##   (Intercept) place1 place2
## 1           1      1      0
## 2           1      1      0
## 3           1      1      0
## 4           1      1      0
## 5           1      0      1
## 6           1      0      1
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  364.375      5.619  64.841  < 2e-16 ***
## place1       -37.500      7.947  -4.719 0.000117 ***
## place2        10.000      7.947   1.258 0.222090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.4875
## F-statistic: 11.94 on 2 and 21 DF,  p-value: 0.000344
```

We have talked about dummy- and effect encoding of categorical covariates. What are the parametrizations used here? What is the interpretation of the parameters and how do the parameter interpretations differ between `model1` and `model2`?

4. We want to test the (one-way ANOVA) null hypothesis that there is no effect of place. Use the F_{obs} to do this both using the dummy-variable and the effect coding of the place-factor. Compare the results from the two coding strategies.

```
model0 = lm(income ~ 1, data = data)
anova(model0, model1)
```

```
## Analysis of Variance Table
##
## Model 1: income ~ 1
## Model 2: income ~ place
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     23 34016
## 2     21 15916  2     18100 11.941 0.000344 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(model0, model2)
```

```
## Analysis of Variance Table
##
## Model 1: income ~ 1
## Model 2: income ~ place
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     23 34016
## 2     21 15916  2     18100 11.941 0.000344 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Suppose now that there are two factors `place`and `gender`.

```
model3 = lm(income ~ place + gender, data = data, x = TRUE, contrasts = list(place = "contr.treatment",
    gender = "contr.treatment"))
summary(model3)
```

```
##
## Call:
## lm(formula = income ~ place + gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.treatment", gender = "contr.treatment"))
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -47.500  -6.250   0.000   9.687  25.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  306.250      6.896  44.411  < 2e-16 ***
## placeB        47.500      8.446   5.624 1.67e-05 ***
## placeC        65.000      8.446   7.696 2.11e-07 ***
## genderMale    41.250      6.896   5.982 7.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.89 on 20 degrees of freedom
## Multiple R-squared:  0.8322, Adjusted R-squared:  0.8071
## F-statistic: 33.07 on 3 and 20 DF,  p-value: 6.012e-08
```

```
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      2 18100.0  9050.0  31.720 6.260e-07 ***
## gender     1 10209.4 10209.4  35.783 7.537e-06 ***
## Residuals 20  5706.2   285.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model4 = lm(income ~ place + gender, data = data, x = TRUE, contrasts = list(place = "contr.sum",
    gender = "contr.sum"))
summary(model4)
```

```
##
## Call:
## lm(formula = income ~ place + gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.sum", gender = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.500  -6.250   0.000   9.687  25.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  364.375      3.448 105.680  < 2e-16 ***
## place1       -37.500      4.876  -7.691 2.13e-07 ***
## place2        10.000      4.876   2.051   0.0536 .
## gender1      -20.625      3.448  -5.982 7.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.89 on 20 degrees of freedom
## Multiple R-squared:  0.8322, Adjusted R-squared:  0.8071
## F-statistic: 33.07 on 3 and 20 DF,  p-value: 6.012e-08
```

```r
anova(model4)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      2 18100.0  9050.0  31.720 6.260e-07 ***
## gender     1 10209.4 10209.4  35.783 7.537e-06 ***
## Residuals 20  5706.2   285.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What are the parameterizations? What is the interpretation of the parameters? Does the ANOVA table look different for the two parametrizations? Hint: orthogonality of design matrix for this balanced design?

6. Finally, fit a model with interactions (model formula is place*gender for both the contrasts and check if the interaction effect is significant.

```r
model5 = lm(income ~ place * gender, data = data, x = TRUE, contrasts = list(place = "contr.treatment",
    gender = "contr.treatment"))
summary(model5)
```

```
##
## Call:
## lm(formula = income ~ place * gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.treatment", gender = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.000  -5.938   1.250  11.250  25.000
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        308.750      8.824  34.989  < 2e-16 ***
## placeB              45.000     12.479   3.606 0.002020 **
## placeC              60.000     12.479   4.808 0.000141 ***
## genderMale          36.250     12.479   2.905 0.009446 **
## placeB:genderMale    5.000     17.648   0.283 0.780168
## placeC:genderMale   10.000     17.648   0.567 0.577963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.65 on 18 degrees of freedom
## Multiple R-squared:  0.8352, Adjusted R-squared:  0.7894
## F-statistic: 18.24 on 5 and 18 DF,  p-value: 1.74e-06
```

```r
anova(model5)
```

```
## Analysis of Variance Table
##
## Response: income
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## place         2 18100.0  9050.0 29.0569 2.314e-06 ***
## gender        1 10209.4 10209.4 32.7793 1.988e-05 ***
## place:gender  2   100.0    50.0  0.1605    0.8529
## Residuals    18  5606.2   311.5
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model6 = lm(income ~ place * gender, data = data, x = TRUE, contrasts = list(place = "contr.sum",
    gender = "contr.sum"))
summary(model6)
```

```
##
## Call:
## lm(formula = income ~ place * gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.sum", gender = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.000  -5.938   1.250  11.250  25.000
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.644e+02  3.602e+00 101.147  < 2e-16 ***
## place1         -3.750e+01  5.095e+00  -7.361 7.86e-07 ***
## place2          1.000e+01  5.095e+00   1.963   0.0653 .
## gender1        -2.062e+01  3.602e+00  -5.725 1.99e-05 ***
## place1:gender1  2.500e+00  5.095e+00   0.491   0.6296
## place2:gender1  1.743e-14  5.095e+00   0.000   1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.65 on 18 degrees of freedom
## Multiple R-squared:  0.8352, Adjusted R-squared:  0.7894
## F-statistic: 18.24 on 5 and 18 DF,  p-value: 1.74e-06
```

```
anova(model6)
```

```
## Analysis of Variance Table
##
## Response: income
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## place         2 18100.0  9050.0 29.0569 2.314e-06 ***
## gender        1 10209.4 10209.4 32.7793 1.988e-05 ***
## place:gender  2   100.0    50.0  0.1605    0.8529
## Residuals    18  5606.2   311.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem 3: Compulsory exercise 1

Introduction to the first compulsory exercise by TA Ingeborg - and an introduction to packages and classes in `R`.

The exercise: https://www.math.ntnu.no/emner/TMA4315/2018h/project1_h18.html

Packages: `ggplot2`, `gamlss.data`, and so on. Some are already loaded when `R` starts (like `stats`), others must be loaded (like `MASS`).

You are going to make your own package, called `mylm`, which performs multiple linear regression and is a smaller version of `lm`.

Show how to create package in R Studio

Classes in R: Something we do not have to think much about, but we use all the time. We are now going to make a new class in R, that we call "test".

```
# takes a word, and returns the index in the alphabet of each
# letter in an object with class 'test'
test <- function(word) {

    x <- 1:nchar(word)
    y <- match(c(strsplit(tolower(word), "")[[1]]), letters[1:26])

    res <- list(x = x, y = y, word = word)  # if you are not familiar with lists, you should read up on
    class(res) <- "test"

    return(res)

}
```

Now we make an object of this class and try to look at it.

```
myname <- test("Ingeborg")
# 'print(myname)' and 'myname' returns the same thing in a script,
# so to simplify we just write 'myname' here
myname  # prints everything
```

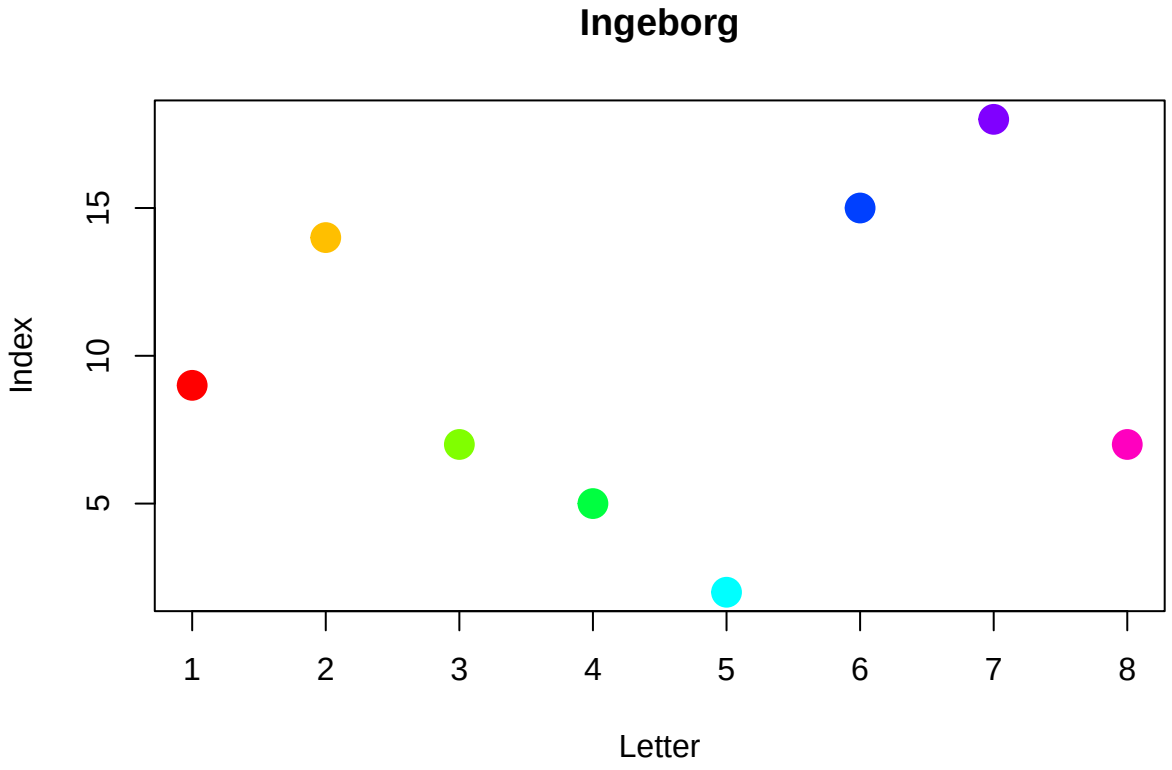
```
## $x
## [1] 1 2 3 4 5 6 7 8
##
## $y
## [1]  9 14  7  5  2 15 18  7
##
## $word
## [1] "Ingeborg"
##
## attr(,"class")
## [1] "test"
```

```
# lets make a print function that only prints the word
print.test <- function(obj) cat(obj$word)

myname  # and now we get only the name
```

```
## Ingeborg
```

Now we want a function that plots objects of this class in a particular way.

```
# important that it is called plot.test with .test at the end!!!
plot.test <- function(obj) plot(obj$x, obj$y, xlab = "Letter", ylab = "Index",
    main = obj$word, col = rainbow(length(obj$x)), pch = 19, cex = 2)

plot(myname)  # we do not have to specify that this is plot.test, because 'myname' is already of class
```




```
# And a summary function
summary.test <- function(obj) {

    cat("Word: ", obj$word, "\n")
    cat("Length of word: ", length(obj$x), " letters\n")
    cat("Occurrence of each letter:")
    print(table(strsplit(tolower(obj$word), "")))

}

summary(myname)
```

```
## Word:  Ingeborg
## Length of word:  8  letters
## Occurrence of each letter:
## b e g i n o r
## 1 1 2 1 1 1 1
```

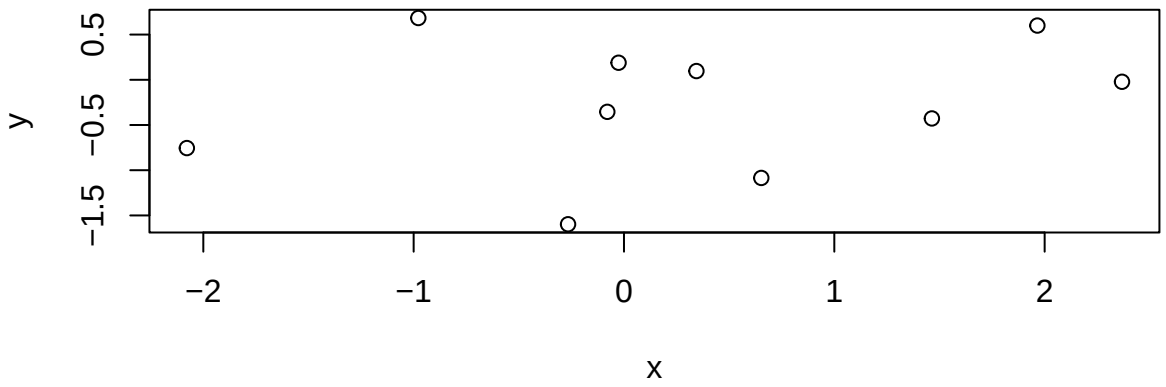
Now we have made a class with a plot, print and summary function, and this is what you do in the exercise! But a bit more advanced...

Let us look at what happens when we use the plotting function on objects with different classes: The function called `plot`. First we make two new objects that can be plotted:

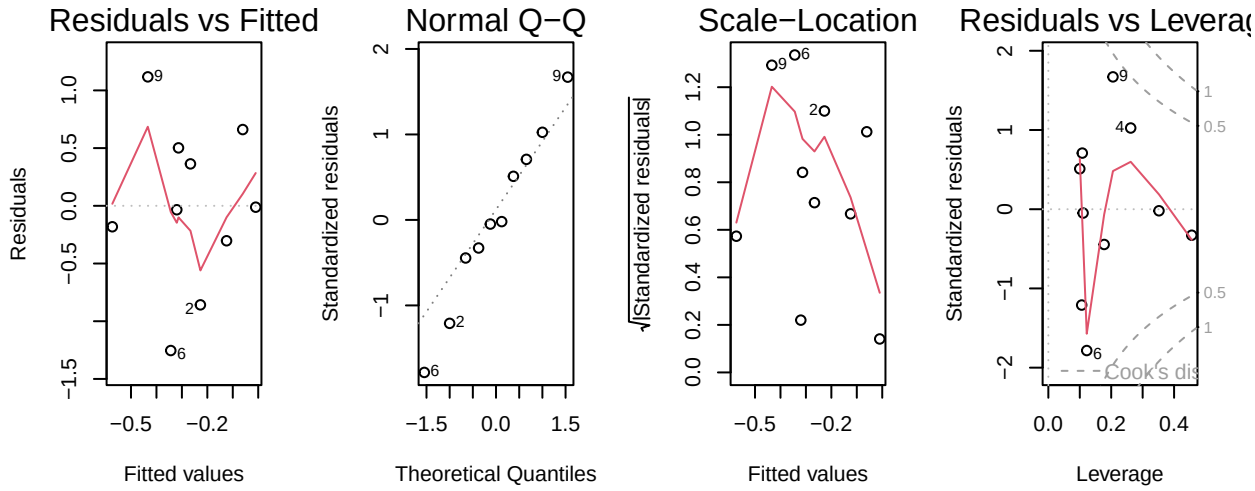
```
data <- data.frame(x = rnorm(10), y = rnorm(10))
mod <- lm(y ~ x, data = data)
```

And then we plot them:

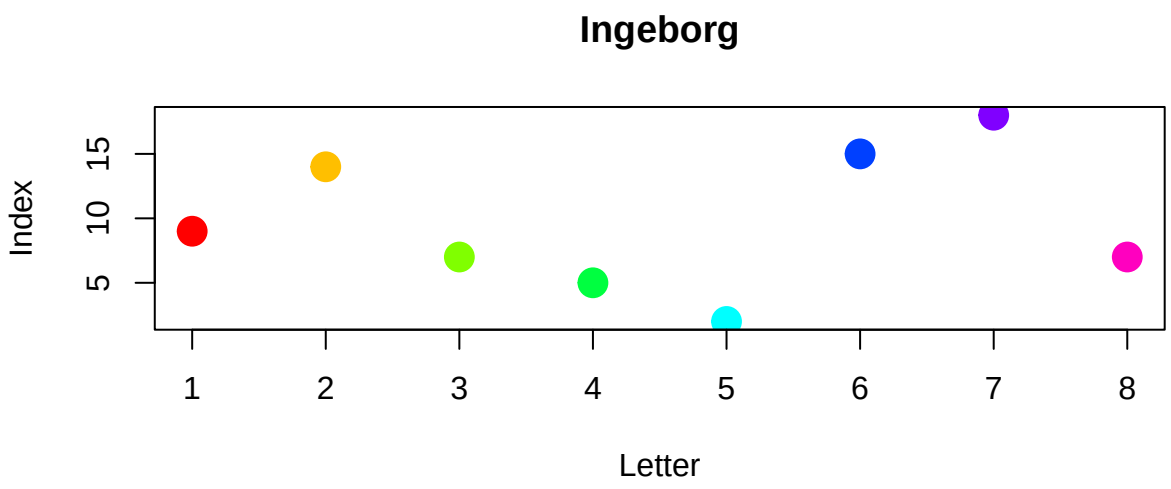
```
plot(data)
```




```
plot(mod)
```




```
plot(myname)
```



What is happening? R reads the class of the objects and uses the plot-function made for that specific class. The user does not have to specify the class as this is already stored in the object!

The different objects we have declared earlier have the following classes:

```
class(data)
class(mod)
class(myname)
```

```
## [1] "data.frame"
## [1] "lm"
## [1] "test"
```

You will make a new class in R called mylm, and R will also then understand which plot-function to use based on the class.

Note that an object can have more than one class.

You write the report using R Markdown, use this template: https://www.math.ntnu.no/emner/TMA4315/2018h/template_glm.Rmd

---

Exercises: Discuss with the group to get a feeling on what to do in the exercise.

1. Go through how to make an R package together in the group, and make the mylm-package.
2. The core is the mylm function. Which formulas are used to
   - calculate parameters estimates?
   - calculate covariance matrix of the estimated regression coefficients?
   - perform type 3 hypothesis tests (remember you need to do the asymptotic normal - so no t-distributions)?
3. You will make print.mylm, plot.mylm and summary.mylm. What should these functions contain?
4. Look at the mylm-template (https://www.math.ntnu.no/emner/TMA4315/2018h/mylm.R) and see if you understand it, or if you have questions about some of the parts. In particular, explore the functions model.frame, model.matrix and model.response.

Problem 4: Munich Rent index (optional)

Last week all groups decided on using rent or rentsqm as response, and in short - there was not really a big difference. So, now use rent as the response.

1. We now want to use model selection to arrive at a good model. Start by defining which covariates you want to include and how to code them (location as dummy or effect coding). What about year of construction - is that a linear covariate? Maybe you want to make intervals in time instead? Linear or categorical for the time? What about the district? We leave that since we have not talked about how to use spatial covariates.

Hint: if you want to test out interval versions of year of construction the function mutate (from dplyr) is useful:

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

More on dplyr: Tutorial: http://genomicsclass.github.io/book/pages/dplyr_tutorial.html and Cheat sheet (data wrangling): https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf and dplyr in particular: https://github.com/rstudio/cheatsheets/raw/master/source/pdfs/data-transformation-cheatsheet.pdf

2. There are many ways to perform model selection in MLR. One possibility is best subsets, which can be done using the regsubsets function from library leaps. You may define x from model.matrix(fit)[,-1] (not including the intercept term), and then run best=regsubsets(x=model.matrix(fit)[,-1],y=rent99$rent) and look at summary(best). Explain the print-out (with all the stars). Using the Mallows Cp (named cp in the list from summary(best)) will give the same result at using AIC (which is not available in this function). What

is your preferred model? Hint: look at the R-code in Problem 2 (Figure 3) from the TMA4267V2017 exam: pdf, and maybe the solutions for the interprtation pdf

3. Check what is done if you use `stepAIC`. Do you get the same model choice as with best subset selection based on AIC? Why, or why not?

Quiz with Kahoot!

One person on each group go to https://kahoot.it on a mobile device or a laptop. (The lecturer will hijack the screen for showing questions so you it is difficult to use the PC.)

Give the pin (shown soon) and then give the team nick name "Group1"-"Group8" or make your own personalized group name. Then - if you want - add nicks for all group members. Work together and only provide *one* answer to each question for each group. In team mode there is a short "team talk" period before you can provide the answer - so you have some time. 1000 points if you answer correctly immediately, 500 if you answer when the time is up, 0 for wrong answers.

Wordclouds are cool?

Run the following code to make the wordcloud. The code can not be run by `knit` because of how the graphics are made - so run and then you need to save the resulting figure as a file (I choose png). Maybe you want to run the code on another document? Please mail Mette.Langaas@ntnu.no if you do cool stuff for others to see!

```
library(wordcloud2)
library(tm)
all = scan("https://www.math.ntnu.no/emner/TMA4315/2018h/2MLR.Rmd", what = "s")

corpus = Corpus(VectorSource(all))
corpus[[1]][1]
corpus = tm_map(corpus, content_transformer(tolower))
corpus = tm_map(corpus, removeNumbers)
corpus = tm_map(corpus, removeWords, stopwords("english"))
corpus = tm_map(corpus, removeWords, c("---", "bf", "boldsymbol", "will",
    "include", "use", "can", "follow", "provide", "using"))
corpus = tm_map(corpus, removePunctuation)
corpus = tm_map(corpus, stripWhitespace)
# corpus=tm_map(corpus,stemDocument)

tdm = TermDocumentMatrix(corpus)
m = as.matrix(tdm)
v = sort(rowSums(m), decreasing = TRUE)
d = data.frame(word = names(v), freq = v)
dim(d)
d[1:10, ]
wordcloud2(d, shape = "cardioid", maxRotation = pi/10, minRotation = -pi/10)
```

R packages

```
install.packages(c("formatR", "gamlss.data", "tidyverse", "ggplot2",
    "GGally", "Matrix", "nortest", "lmtest", "wordcloud2", "tm"))
```

References and further reading

- Slightly different presentation (more focus on multivariate normal theory): Slides and written material from TMA4267 Linear Statistical Models in 2017, Part 2: Regression (by Mette Langaas).
- And, same source, but now [Slides and written material from TMA4267 Linear Statistical Models in 2017, Part 3: Hypothesis testing and ANOVA] (http://www.math.ntnu.no/emner/TMA4267/2017v/TMA4267V2017Part3.pdf)