Module 2: MULTIPLE LINEAR REGRESSION

TMA4315 Generalized linear models H2018

Mette Langaas, Department of Mathematical Sciences, NTNU –
with contributions from Øyvind Bakke and Ingeborg Hem

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

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

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

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 $s(\hat{\beta}) = \mathbf{0}$.
- Observed $H(\beta) = -\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T}$ and expected Fisher information $F(\beta) = \mathrm{E}(H(\beta))$

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:

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

  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.
- Asymptotically

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

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

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 $s(\hat{\beta}) = \mathbf{0}$ so we will use numerical optimization to handle this, and the ´biglm´ also solves the GLM.

Inference

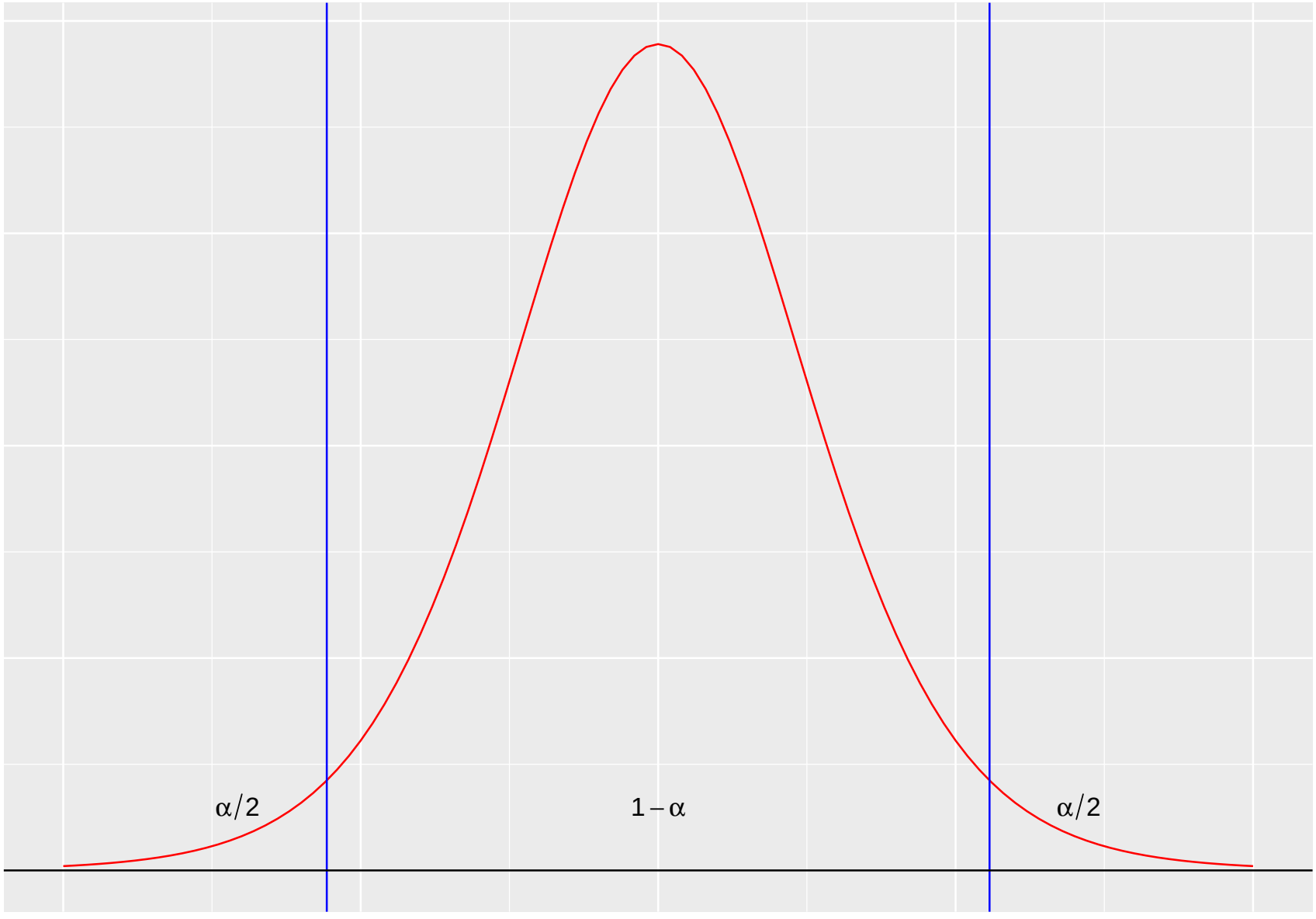
We will consider confidence intervals and prediction intervals, and then test single and linear hypotheses. Most of this should be known to you from earlier regression courses. We will only focus on the results, and you need to read the details in the derivation by yourself.

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, dat
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

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

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

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

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
newobs = rent99[1, ]
newobs[1, ] = c(NA, NA, 50, NA, 2, 1, 1, 1, NA)
predict(fit, newdata = newobs, interval = "prediction", typ
```

```
##        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"
## [7] "kitchen"  "cheating" "district"
```

```
fit = lm(rent ~ area + location + bath + kitchen + cheating, data = ren
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
##
```

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
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 p.

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

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

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

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 $\mathbf{C}$ and $\mathbf{d}$?

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 6×3 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 $\mathbf{C}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{C}^\mathsf{T} = c_{11}$, 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

$H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0$ vs. H_1 : 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.

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

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

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

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

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

(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
```

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:

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,

$$\text{SST} = \text{SSR} + \text{SSE}$$

$$\text{SST} = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 = \mathbf{Y}^T (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\mathbf{Y}$$

$$\text{SSR} = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 = \mathbf{Y}^T (\mathbf{H} - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\mathbf{Y}$$

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
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
```

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
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
```

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).

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

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)

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.

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
summary(fit)
```

```
##
## Call:
## lm(formula = rent ~ area + location + bath + kitchen + c
##     data = rent99)
##
```

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, 43
    300, 320, 310, 305, 350, 370, 340, 355, 370, 380, 360,
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 =
    "Male")), place = factor(place, levels = c("A", "B", "C
```

2. Fit the following model `model =`

Problem 3: Compulsory exercise 1

The TA will give a short introduction to the exercise - and talk about the `lm`-class.

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 =
    2000, 10)), labels = 10 * 1:9))
```

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/2M

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", "boldsy
    "include", "use", "can", "follow", "provide", "using"))
corpus = tm_map(corpus, removePunctuation)
corpus = tm_map(corpus, stripWhitespace)
```

R packages

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

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)