TMA4315 Generalized linear models H2018

Module 3: BINARY REGRESSION

Mette Langaas, Department of Mathematical Sciences, NTNU - with contributions from Øyvind Bakke,

13.09 and 20.09 [PL], 14.09 and 21.09 [IL]

Contents

(Latest changes: 09.11: clarified one sentence on the devianc. 23.09: score test moved to M4. 20.09: typos, and added solutions to Qs in class. 18.09: typos and added sententence ILw2Problem3c. 16.09: edited and added material for week 2, 13.09 moved material not lectured to after the ILw1, and added one sentence to Problem 5 ILw1.)

Overview

Learning material

- Textbook: Fahrmeir et al (2013): Chapter 2.3, 5.1, B4.1-3
- Classnotes 13.09.2018
- Classnotes 20.09.2018

Topics

First week

- aim of binary regression
- how to model a binary response
- three ingredients of a GLM model
- the logit model: logistic regression
- interpreting the logit model - with odds
- grouped vs. individual data

- parameter estimation with maximum likelihood
  - likelihood, log-likelihood,
  - score function

Jump to interactive lecture (week 1)

Second week

- Parameter estimation
  - score function- and mean and covariance thereof,
  - observed and expected information matrix
- comparison with the normal distribution - score function and Fisher information
- exponential family and canonical link
- iterative calculation of ML estimator (Newton-Raphson and Fisher scoring) - and in R with `optim`
- asymptotic properties of ML estimators - how to use in inference?
- statistical inference
  - confidence intervals
  - hypothesis testing: Wald, and likelihood ratio
- deviance: definition, analysis of deviance, deviance residuals
- model fit and model choice
- overdispersion and estimating overdispersion parameter
- sampling stragegy: cohort, but also case-control data good for logit model

Jump to interactive lecture (week 2)

FIRST WEEK

Aim of binary regression

Two aims

1. Construct a model to help understand the relationship between a "success probability" and one or several explanatory variables. The response measurements are binary (present/absent, true/false, healthy/diseased).
2. Use the model for estimation and prediction of success probabilites.

Two running examples: mortality of beetles and probability of respiratory infant disease.

Example: Mortality of beetles

A total of 481 beetles were exposed to 8 different consentration of CS_2 (data on log10-dose). Yes, only one consentration tried for each beetle. For each beetle is was recorded if the beetle was alive or killed at the given concentration.

Data for beetle i: $Y_i = 0$ if beetle i was alive and $Y_i = 1$ if it was killed, and x_i is then the log10-dose beetle i was given.

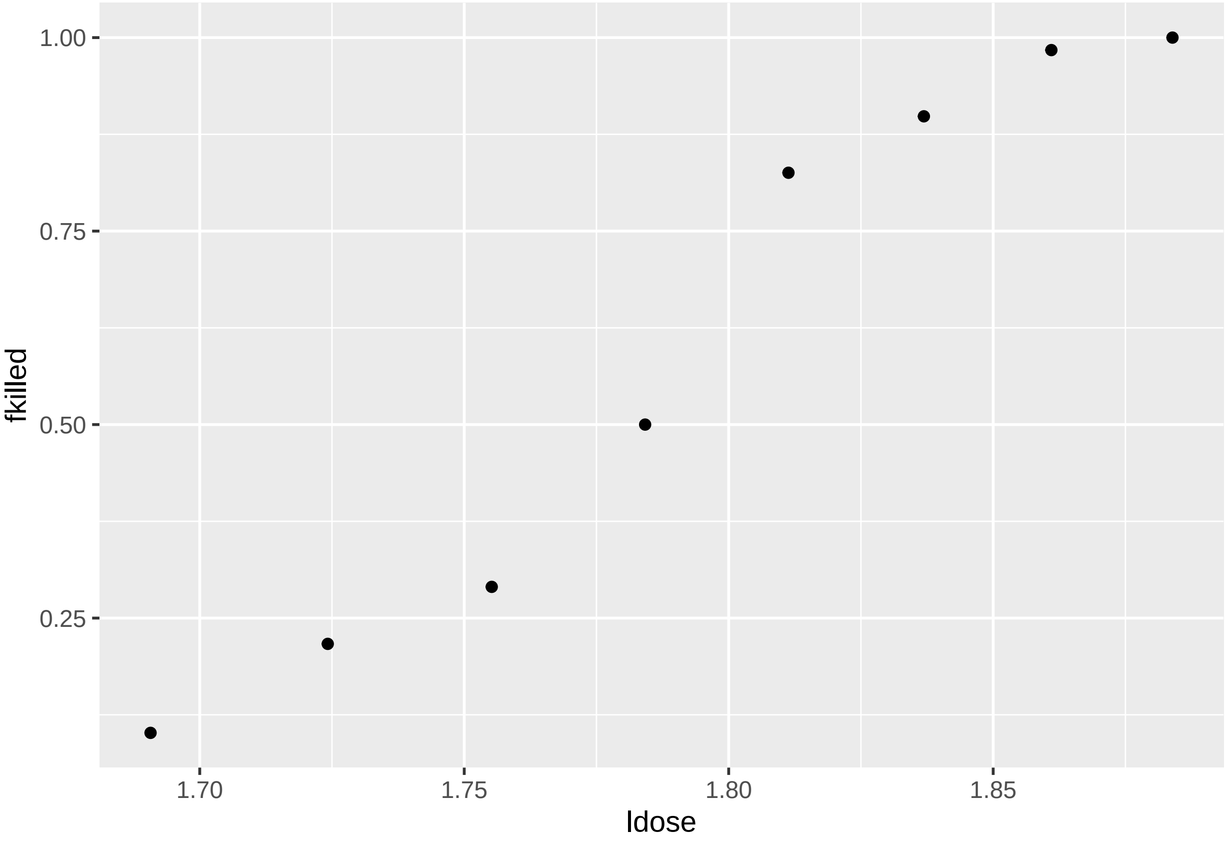
The table below shows the 8 values of the log10-dose against the number of beetles alive and killed. The plot shows log10-dose on the horisontal axis and fraction of beetles killed (killed/total) for each log10-dose.

```
library(investr)
# from aggregated to individual data (because these data were aggregated)
ldose=rep(beetle$ldose,beetle$n)
y=NULL; for (i in 1:8) y=c(y,rep(0,beetle$n[i]-beetle$y[i]),rep(1,beetle$y[i]))
```

```
beetleds=data.frame("killed"=y,"ldose"=ldose)
table(beetleds)
```

```
##       ldose
## killed 1.6907 1.7242 1.7552 1.7842 1.8113 1.8369 1.861 1.8839
##      0     53     47     44     28     11      6     1      0
##      1      6     13     18     28     52     53    61     60
```

```
# plot from aggregated data
frac=beetle$y/beetle$n
dss=data.frame(fkilled=frac,ldose=beetle$ldose)
ggplot(dss,aes(ldose,fkilled))+
  geom_point()
```



Q:

a. What might be the effect (mathematical function) of the log10-dose on the probability of killing a beetle?
b. How can this curve be part of a regression model?

```
## Answers

## a) logistic, sigmoid, normal cdf?.

## b) see item 3 below - the response function connects the mean of the respons to the linear predictor
```

How to model a binary response?

In multiple linear regression we have

1. Random component: Distribution of response: $Y_i \sim N(\mu_i, \sigma^2)$, where μ_i is *parameter of interest* and σ^2 is *nuisance*.
2. Systematic component: Linear predictor: $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$. Here $\mathbf{x}_i$ is our fixed (not random) p-dimensional column vector of covariates (intercept included).
3. Link: Connection between the linear predictor and the mean (parameter of interest): $\mu_i = \eta_i$.

In addition we have independent pairs $(\mathbf{x}_i, Y_i)$, and assume that the design matrix - with the covariates for all observation in a $n \times p$ matrix - has full rank p.

- It would not make sense to fit the continuous linear regression to Y_i when $Y_i = \{0, 1\}$ - since Y_i is not a continuous random variable, and Y_i is not normal.
- So, we need to change 1. We keep 2. And, we make 3. more general.

<id="binary"> Binary regression

1. $Y_i \sim \text{bin}(n_i, \pi_i)$. First we study the case that $n_i = 1$, that is, each individual observation comes from a Bernoulli process with $n_i = 1$ trials (this version of the binomial distribution is called the Bernoulli distribution). (Remark: later we will also look at "grouped" data with $n_i > 1$.) Our parameter of interest is π_i which is the mean $\text{E}(Y_i) = \mu_i = \pi_i$.

For a generalized linear model (GLM) we require that the distribution of the response is an exponential family. We have seen in M1 that the binomial distribution is an exponential family.

---

2. Linear predictor: $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$.
3. We will consider different relationships between the mean $\mu_i = \pi_i$ and the linear predictor η_i, and define the *link function* g as
$$g(\mu_i) = \eta_i$$
and the inverse of the link function, called the *response function*, and denoted by
$$h(\eta_i) = g^{-1}(\eta_i) = \mu_i$$

We thus also have to require that the link function is monotone, and we will soon see that we also need to require that it is twice differential.

These three ingredients (exponential family distribution of reponse, linear predictor and choice of reponse or link function) give the core of our GLM model.

Popular choices for the respons function for binary regression are based on selecting a cumulative distribution function (cdf) as the response function. The cdf will always be within [0,1], and the cdf is monotone - which will help us to interpret results.

The most popular response functions are:

- *logistic cdf* (with corresponding *logit* link function) referred to as the *logit model*, followed by the
- *normal cdf* - (with corresponding *probit* link function) referred to as the *probit model* , and
- the *extreme minimum-value cdf* (with corresponding *complementary log-log* link function) referred to as the *complementary log-log model*.

In this module we focus on the logit model.

The logit model aka logistic regression

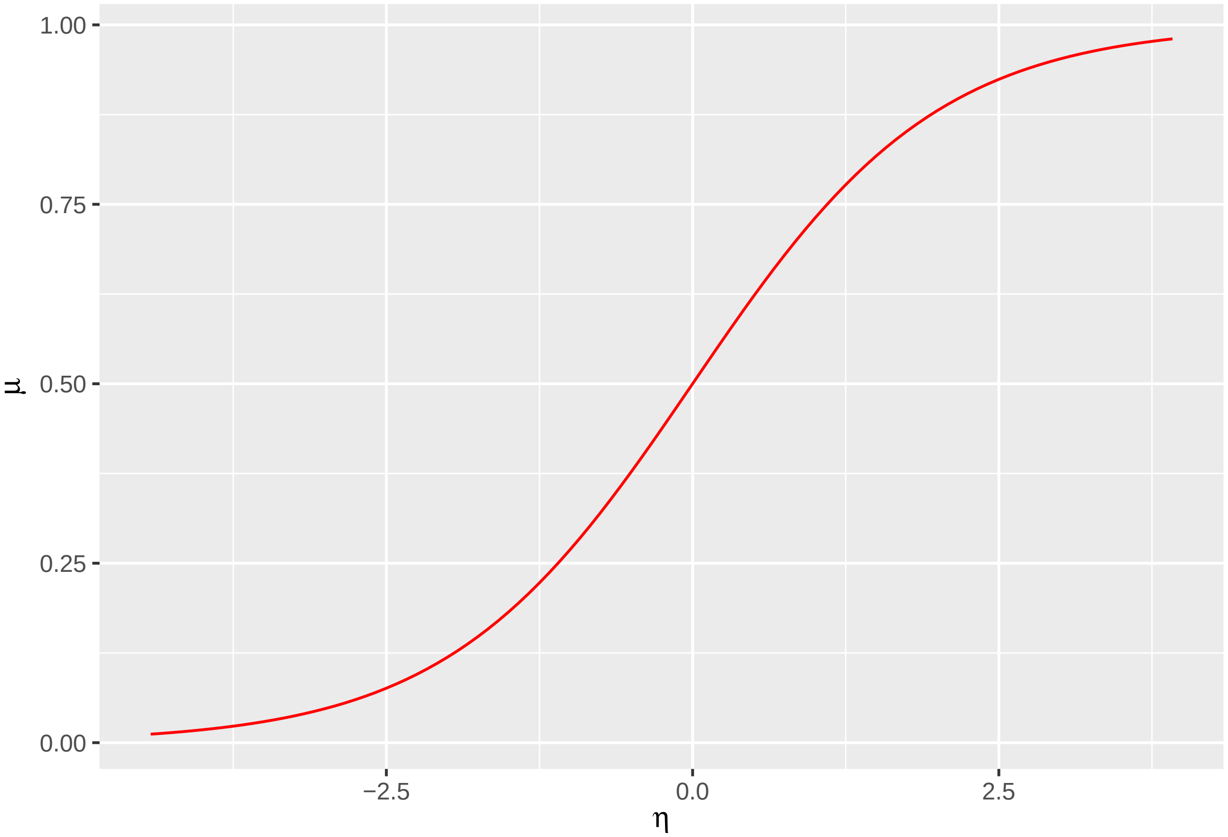
Beetle mortality: response function

In the beetle example we have a simple linear predictor: $\eta_i = \beta_0 + \beta_1 x_i$ where x_i is the log10-dose for beetle i.

Assume that $\beta_0 =$ -60.7 and $\beta_1 =$ 34.3. (These values are estimates from our data, and we will see later how to find these estimates using maximum likelihood estimation.)

Below the response function is plotted for $\eta_i =$-60.7+34.3x_i.

```r
library(ggplot2)
#print(c(beta0, beta1))
xrange=range(beetleds$ldose)
xrange[1]=xrange[1]-0.05
etas=beta0+beta1*seq(xrange[1],xrange[2],length=100)
ggplot(data.frame(eta=range(etas),mu=c(0,1)), aes(eta,mu))+
  xlab(expression(eta))+
  ylab(expression(mu))+
  stat_function(fun=function(eta) exp(eta)/(1+exp(eta)), geom="line", colour="red")
```



R: more on greek letters with ggplot: https://github.com/tidyverse/ggplot2/wiki/Plotmath

---

Q: Explain to your neighbour what is on the x- and y-axis of this plot. Where are the observed log10-doses in this graph?

A:

pivec	0.10	0.20	0.30	0.40	0.5	0.6	0.70	0.8	0.9
odds	0.11	0.25	0.43	0.67	1.0	1.5	2.33	4.0	9.0

On the x-axis is the linear predictor $\eta = \beta_0 + \beta_1 x_1$ for the given values of β_0 and β_1 and values for x_1 is chosen from the range of the log10-dose values. On the y-axis is the model for the mean of the response, which also here is the probability of success. This is our non-linear relationship between the linear predictor and the mean = our response function.

Link and reponse function

The logit model is based on the logistic cdf as the response function, given as

$$\mu_i = \pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$$

or alternatively as the link function (the inverse of the response function)

$$g(\mu_i) = h^{-1}(\mu_i) = \ln(\frac{\mu_i}{1-\mu_i}) = \ln(\frac{\pi_i}{1-\pi_i})$$

Hands-on: show this for yourself.

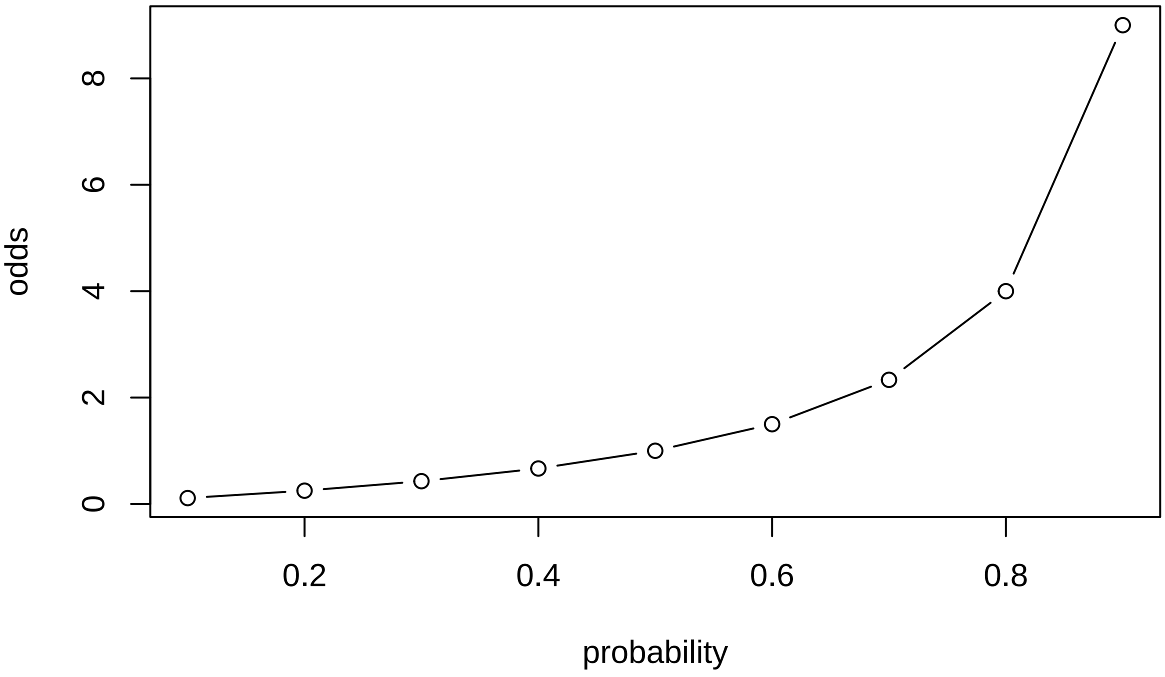
Interpreting the logit model

If the value of the linear predictor η_i changes to $\eta_i + 1$ the probability π increases non-linearly from $\frac{\exp(\eta_i)}{1+\exp(\eta_i)}$ to $\frac{\exp(\eta_i+1)}{1+\exp(\eta_i+1)}$, as shown in the graph above.

Before we go further: do you know about the odds? The ratio $\frac{P(Y_i=1)}{P(Y_i=0)} = \frac{\pi_i}{1-\pi_1}$ is called the *odds*. If $\pi_i = \frac{1}{2}$ then the odds is 1, and if $\pi_i = \frac{1}{4}$ then the odds is $\frac{1}{3}$. We may make a table for probability vs. odds in R:

```r
library(knitr)
library(kableExtra)
pivec=seq(0.1,0.9,0.1)
odds=pivec/(1-pivec)
kable(t(data.frame(pivec,odds)),digits=c(2,2))%>%
  kable_styling()
```

```r
library(knitr)
library(kableExtra)
pivec=seq(0.1,0.9,0.1)
odds=pivec/(1-pivec)
plot(pivec,odds,type="b",ylab="odds",xlab="probability")
```



R: Pretty table with kableExtra

Odds may be seen to be a better scale than probability to represent chance, and is used in betting. In addition, odds are unbounded above.

We look at the link function (inverse of the response function). Let us assume that our linear predictor has k covariates present

$$\begin{aligned}
\eta_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} \\
\pi_i &= \frac{\exp(\eta_i)}{1+\exp(\eta_i)} \\
\eta_i &= \ln(\frac{\pi_i}{1-\pi_i}) \\
\ln(\frac{\pi_i}{1-\pi_i}) &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} \\
\frac{\pi_i}{1-\pi_i} =& \frac{P(Y_i=1)}{P(Y_i=0)} = \exp(\beta_0) \cdot \exp(\beta_1 x_{i1}) \cdots \exp(\beta_k x_{ik})
\end{aligned}$$

We have a *multiplicative model* for the odds.

So, what if we increase x_{1i} to $x_{1i}+1$?

If the covariate x_{1i} increases by one unit (while all other covariates are kept fixed) then the odds is multiplied by $\exp(\beta_1)$:

$$\begin{aligned}\frac{P(Y_i=1\mid x_{i1}+1)}{P(Y_i=0)\mid x_{i1}+1)} &= \exp(\beta_0)\cdot\exp(\beta_1(x_{i1}+1))\cdots\exp(\beta_k x_{ik})\\ &= \exp(\beta_0)\cdot\exp(\beta_1 x_{i1})\exp(\beta_1)\cdots\exp(\beta_k x_{ik})\\ &= \frac{P(Y_i=1\mid x_{i1})}{P(Y_i=0\mid x_{i1})}\cdot\exp(\beta_1)\end{aligned}$$

This means that if x_{i1} increases by 1 then: if $\beta_1 < 0$ we get a decrease in the odds, if $\beta_1 = 0$ no change, and if $\beta_1 > 0$ we have an increase. In the logit model $\exp(\beta_1)$ is easier to interpret than β_1.

So, to sum up: for the linear predictor we interpret effects in the same way as for the linear model (in Module 2), then we transform this linear effect in η into a nonlinear effect for $\pi = \frac{\exp(\eta)}{1+\exp(\eta)}$, and use the odds to interpret changes.

##Infant respitory disease : interpretation of parameter estimates (This example is taken from Faraway (2006): "Extending the linar model with R")

We select a sample of newborn babies (girls and boys) where the parents had decided on the method of feeding (bottle, breast, breast with some supplement), and then monitored the babies during their first year to see if they developed infant respiratory disease (the event we want to model).

We fit a logistic regression to the data, and focus on the parameter estimates.

```
library(faraway)
data(babyfood)
babyfood
```

```
##   disease nondisease  sex   food
## 1      77        381  Boy Bottle
## 2      19        128  Boy  Suppl
## 3      47        447  Boy Breast
## 4      48        336 Girl Bottle
## 5      16        111 Girl  Suppl
## 6      31        433 Girl Breast
```

```
xtabs(disease/(disease+nondisease)~sex+food,babyfood)
```

```
##       food
## sex        Bottle     Breast      Suppl
##   Boy  0.16812227 0.09514170 0.12925170
##   Girl 0.12500000 0.06681034 0.12598425
```

```
fit=glm(cbind(disease, nondisease)~sex+food,family=binomial(link=logit),data=babyfood)
summary(fit)
```

```
##
## Call:
## glm(formula = cbind(disease, nondisease) ~ sex + food, family = binomial(link = logit),
##     data = babyfood)
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.6127     0.1124 -14.347  < 2e-16 ***
## sexGirl      -0.3126     0.1410  -2.216   0.0267 *
## foodBreast   -0.6693     0.1530  -4.374 1.22e-05 ***
## foodSuppl    -0.1725     0.2056  -0.839   0.4013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 26.37529  on 5  degrees of freedom
## Residual deviance:  0.72192  on 2  degrees of freedom
## AIC: 40.24
##
## Number of Fisher Scoring iterations: 4
```

```
exp(fit$coefficients)
```

```
## (Intercept)     sexGirl  foodBreast   foodSuppl
##   0.1993479   0.7315770   0.5120696   0.8415226
```

```
predict(fit,type="response") #gives predicted probabilites
```

```
##          1          2          3          4          5          6
## 0.16621356 0.14365654 0.09262485 0.12727653 0.10931093 0.06948992
```

Q: Observe that the two factors by default is coded with dummy variable coding, and that `sexBoy` is the reference category for sex and `foodBottle` the reference category for feeding method.

1: Explain how to interpret the `Estimate` for `sexGirl`, `foodBreast` and `foodSuppl`.

2: What are the 6 values given by the call to `predict`? What is the least favourable combination of sex and method of feeding? And the most favourable?

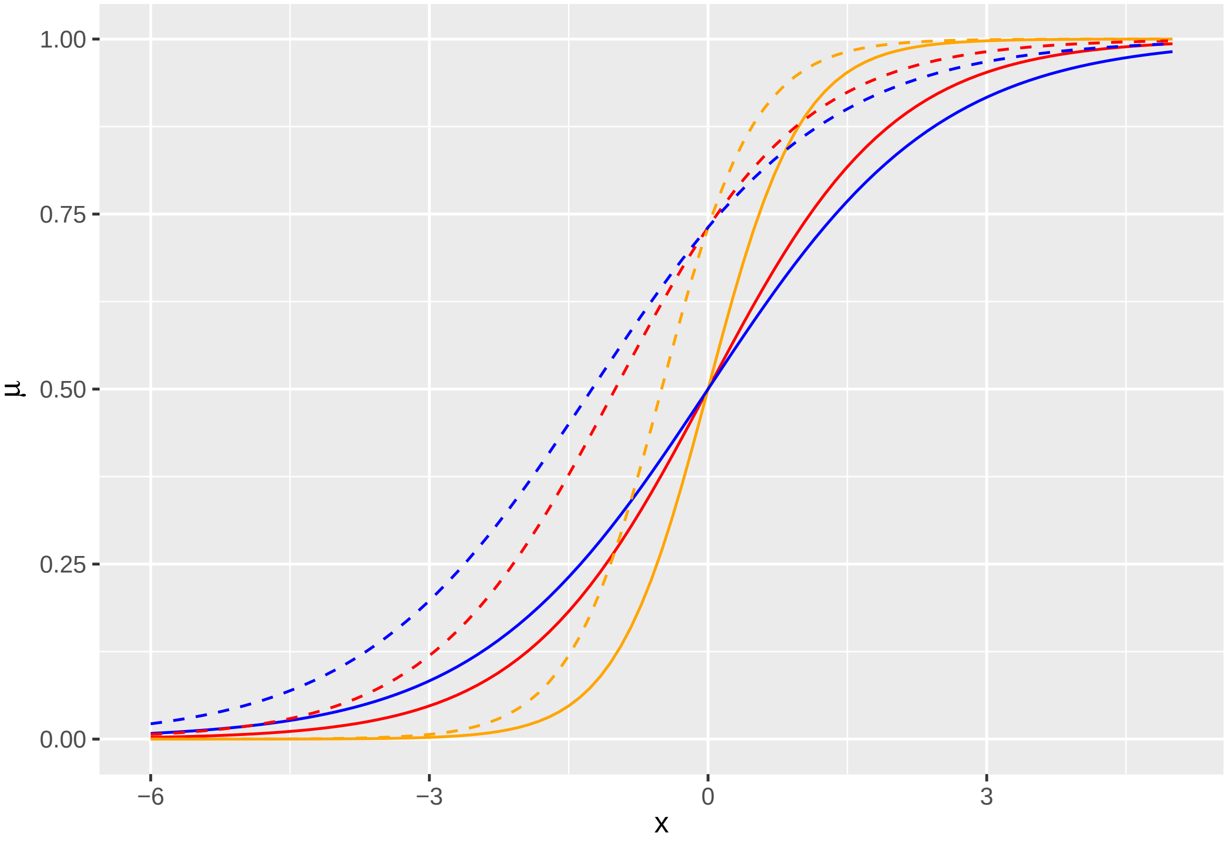
Comment: we have here fitted an additive model in the two covariates, but we could also include an interaction term. This will be discussed later.

More response function plots for the logit model

The response function as a function of the covariate x and not of η. Solid lines: $\beta_0 = 0$ and β_1 is 0.8 (blue), 1 (red) and 2 (orange), and dashed lines with $\beta_0 = 1$.

```
library(ggplot2)
ggplot(data.frame(x=c(-6,5)), aes(x))+
  xlab(expression(x))+
  ylab(expression(mu))+
    stat_function(fun=function(x) exp(x)/(1+exp(x)), geom="line", colour="red")+
    stat_function(fun=function(x) exp(2*x)/(1+exp(2*x)), geom="line", colour="orange")+
          stat_function(fun=function(x) exp(0.8*x)/(1+exp(0.8*x)), geom="line", colour="blue")+
    stat_function(fun=function(x) exp(1+x)/(1+exp(1+x)), geom="line", colour="red",linetype="dashed")+
    stat_function(fun=function(x) exp(1+2*x)/(1+exp(1+2*x)), geom="line", colour="orange",linetype="das
          stat_function(fun=function(x) exp(1+0.8*x)/(1+exp(1+0.8*x)), geom="line", colour="blue",linet
  scale_colour_manual("0+k x",values = c("red", "orange","blue"),labels=c("1","2","0.8"))
```

disease	nondisease	sex	food
77	381	Boy	Bottle
19	128	Boy	Suppl
47	447	Boy	Breast
48	336	Girl	Bottle
16	111	Girl	Suppl
31	433	Girl	Breast



Grouped vs. individual data

So far we have only mentioned individual (ungrouped) data. That is, every Y_i is 0 or 1 and has a corresponding covariate vector $\mathbf{x}_i$ - and together they form *one* unit.

$$Y_i \sim \text{bin}(n_i = 1, \pi_i)$$

However, in both the examples we have looked at some covariate vectors are *identical* (rows in the design matrix are identical). We call these unique combinations of covariates *covariate patterns*, and say we have *grouped* data.

```r
library(kableExtra)
knitr::kable(babyfood) %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Here we have 6 groups of covariate patterns. The first group has covariates `Boy` and `Bottle`, there are 77+381= 458 babies with this combination and 77 of these got the disease.

We prefer to group data if possible. Grouping is good because then data can be kept in a condenced form, it will speed up computations and makes model diagnosis easier (than for individual data).

For the grouped data we still have a binomial distribution, and possible generalization is to let

- $n_j \bar{Y_j}$ be the number of successes in group j,
- which means that $\bar{Y_j} = \frac{1}{n_j} \sum Y_i$ where the sum is over all i in group j.

Further

$$n_j \bar{Y_j} \sim \text{bin}(n_j, \pi_j)$$

such that $\text{E}(n_j \bar{Y_j}) = n_j \pi_j$ and $\text{Var}(n_j \bar{Y_j}) = n_j \pi_j (1 - \pi_j)$, and $\text{E}(\bar{Y_j}) = \pi_j$ and $\text{Var}(\bar{Y_j}) = \frac{1}{n_j} \pi_j (1 - \pi_j)$

We then keep the linear predictor, and the link function is still $\eta_j = \ln(\frac{\pi_j}{1-\pi_j})$. That is, we do not model the mean $n_j \pi_j$ but π_j directly.

Q: What is a covariate pattern?

```
## Answer

## A set of values for covariates that are the same for a group of individual data. For the beetle examp
```

Likelihood and derivations thereof

Our parameter of interest is the vector $\boldsymbol{\beta}$ of regression coefficients, and we have no nuicance parameters (because the variance is related directly to the π_j and n_j is known).

We would like to estimate $\boldsymbol{\beta}$ from maximizing the likelihood, but we will soon see that we have no closed form solution. First we look at the likelihood, the log-likelihood and first and second derivatives thereof.

For simplicity we do the derivations for the case where $n_i = 1$, but then include the results for the case where we have G covariate patterns with n_j observations of each pattern.

Assumptions:

1. $Y_i \sim \text{bin}(n_i = 1, \pi_i)$, and $\text{E}(Y_i) = \mu_i = \pi_i$, and $\text{Var}(Y_i) = \pi_i(1 - \pi_i)$.
2. Linear predictor: $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$.
3. Logit link

   $$\eta_i = \ln(\frac{\pi_i}{1 - \pi_i}) = g(\mu_i)$$

   and (inverse thereof) logistic response function

   $$\mu_i = \pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} = h(\eta_i)$$

We will also need:

$$(1 - \pi_i) = 1 - \frac{\exp(\eta_i)}{1 + \exp(\eta_i)} = \frac{1 + \exp(\eta_i) - \exp(\eta_i)}{1 + \exp(\eta_i)} = \frac{1}{1 + \exp(\eta_i)}.$$

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.

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

Q: What is the interpretation of the likelihood? Is it not a misuse of notation to write $L(\boldsymbol{\beta})$ when the right hand side does not involve $\boldsymbol{\beta}$?

A:

The likelihood is the joint distribution of the responses, but with focus on the unknown parameter vector. Yes, a slight misuse of notation, need to fill in $\pi_i = h(\eta_i)$ and $\eta_i = \mathbf{x}^T\boldsymbol{\beta}$ to write out $L(\boldsymbol{\beta})$, really. But we keep this version because when we want to take the partical derivatives we will use the chain rule.

Loglikelihood $l(\boldsymbol{\beta})$

The log-likelihood is the natural log of the likelihood, and 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.

$$\begin{aligned}
l(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) &= \sum_{i=1}^n \ln L_i(\boldsymbol{\beta}) = \sum_{i=1}^n l_i(\boldsymbol{\beta}) && (1)\\
&= \sum_{i=1}^n [y_i \ln \pi_i + (1-y_i)\ln(1-\pi_i)] && (2)\\
&= \sum_{i=1}^n [y_i \ln(\frac{\pi_i}{1-\pi_i}) + \ln(1-\pi_i)] && (3)
\end{aligned}$$

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

The log-likelihood is now expressed as a function of π_i, but we want to make this a function of $\boldsymbol{\beta}$ and the connection between π_i and $\boldsymbol{\beta}$ goes through η_i. We have that $\pi = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$ and in our log-likelihood we need

$$(1-\pi_i) = \frac{1}{1+\exp(\eta_i)} = \frac{1+\exp(\eta_i)-\exp(\eta_i)}{1+\exp(\eta_i)} = \frac{1}{1+\exp(\eta_i)}$$

and

$$\ln(\frac{\pi_i}{1-\pi_1}) = \eta_i$$

(the last is our logit link function). Then we get:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^n [y_i\eta_i + \ln(\frac{1}{1+\exp(\eta_i)})] = \sum_{i=1}^n [y_i\eta_i - \ln(1+\exp(\eta_i))]$$

which is now our function of η_i.

Finally, since $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$,

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \mathbf{x}_i^T \boldsymbol{\beta} - \ln(1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta}))].$$

Q: What does the graph of l look like as a function of $\boldsymbol{\beta}$?

If we look at the beetle example we only have one covariate (in addition to the intercept) - so this means that we have $\boldsymbol{\beta} = (\beta_0, \beta_1)$. Plotting the log-likelihood (for the beetle data set) will be one of the tasks for the interactive lecture.

But, next we take partial derivatives, and then we will (instead of using this formula) look at $l_i(\boldsymbol{\beta}) = l_i(\eta_i(\boldsymbol{\beta}))$ and use the chain rule.

Score function $s(\boldsymbol{\beta})$

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

Q: Write down the rules for derivatives: chain rule, product rule, fraction rule, and in particular derivative of $\ln(x)$, $\exp(x)$ and $\frac{1}{x}$, you will need them now.

A: Chain rule: $\frac{df(u(x))}{du} = \frac{df}{du} \cdot \frac{du}{dx}$, product rule: $(u \cdot v)' = u' \cdot v + u \cdot v'$, fraction rule: $(\frac{u}{v})' = \frac{u' \cdot v - u \cdot v'}{v^2}$, $\frac{d\ln(x)}{dx} = \frac{1}{x}$, $\frac{d\exp(x)}{dx} = \exp(x)$ and $\frac{d(\frac{1}{x})}{dx} = -\frac{1}{x^2}$.

Here we go:

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

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

We will use the chain rule to calculate $s_i(\boldsymbol{\beta})$.

$$s_i(\boldsymbol{\beta}) = \frac{\partial l_i(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial l_i(\boldsymbol{\beta})}{\partial \eta_i} \cdot \frac{\partial \eta_i}{\partial \boldsymbol{\beta}} = \frac{\partial [y_i \eta_i - \ln(1 + \exp(\eta_i))]}{\partial \eta_i} \cdot \frac{\partial [\mathbf{x}_i^T \boldsymbol{\beta}]}{\partial \boldsymbol{\beta}}$$

$$s_i(\boldsymbol{\beta}) = (y_i - \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}) \cdot \mathbf{x}_i = (y_i - \pi_i)\mathbf{x}_i$$

Here we have used the general rule for all partial derivatives of scalar with respect to vector (also used in TMA4267):

$$\frac{\partial \mathbf{a}^T \mathbf{b}}{\partial \mathbf{b}} = \mathbf{a}$$

and later we will also need

$$\frac{\partial \mathbf{a}^T \mathbf{b}}{\partial \mathbf{b}^T} = (\frac{\partial \mathbf{a}^T \mathbf{b}}{\partial \mathbf{b}})^T = \mathbf{a}^T.$$

The score function is given as:

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} s_i(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i (y_i - \pi_i) = \sum_{i=1}^{n} \mathbf{x}_i (y_i - \frac{\exp(\mathbf{x}_i^T \boldsymbol{\beta})}{1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta})})$$

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

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

We will soon see how we can do that using the Newton-Raphson or Fisher Scoring iterative methods, but first we will work on finding the mean and covariance matrix of the score vector - and the derivatives of the score vector (the Hessian, which is minus the observed Fisher matrix).

Remark: in Module 5 we will see that the general formula for GLMs is:

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} [\frac{y_i - \mu_i}{\text{Var}(Y_i)} \mathbf{x}_i \frac{\partial \mu_i}{\partial \eta_i}] = \sum_{i=1}^{n} [\frac{y_i - \mu_i}{\text{Var}(Y_i)} \mathbf{x}_i h'(\eta_i)] = \mathbf{X}^T \mathbf{D} \Sigma^{-1} (\mathbf{y} - \mu)$$

where $\mathbf{X}$ is the $n \times p$ design matrix, $\mathbf{D} = \text{diag}(h'(\eta_1), h'(\eta_2), \ldots, h'(\eta_n))$ is a diagonal matrix with the derivatives of the response function evaluated at each observation. Further, $\Sigma = \text{diag}(\text{Var}(Y_1), \text{Var}(Y_2), \ldots, \text{Var}(Y_n))$ is a diagonal matrix with the variance for each response, and $\mathbf{y}$ is the observed $n \times 1$ vector of responses and μ is the $n \times 1$ vector of individual expectations $\mu_i = \text{E}(Y_i) = h(\eta_i)$.

More in Module 5.

Interactive lecture - first week

Theoretical questions - with and without use of R

Problem 1: Model assumptions

a) What are the model assumptions for a binary regression?
b) Which link function and response function is used for the logit model?
c) What is the difference between the logit model and a logistic regression?

Problem 2: Log-likelihood.

a) What is the definition of the log-likelihood?

b) For the logit model the log-likelihood is

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{G} [\ln \binom{n_j}{y_j} + y_j \ln \pi_j - y_j \ln(1-\pi_j) + n_j \ln(1-\pi_j)]$$

for grouped data. Explain how we have arrived at this formula?

c) Write the version of the loglikelihood for individual data (i.e. $n_j = 1$ and $G = n$).

d) Where is $\boldsymbol{\beta}$ in the loglikelihood in c? Rewrite this to be a function of $\boldsymbol{\beta}$.

e) Why can we ignore the normalizing constant (what is the constant?) in the case of $n_j = 1 \ \forall j$? Considering what the log-likelihood is used for, why can we ignore the normalizing constant in all cases (i.e., also when $n_j \neq 1$)?

f) What does this graph of l look like as a function of $\boldsymbol{\beta}$ for the beetle data? First discuss shortly and then to aid you in answering this we look at the loglikelihood for the beetle data. Read the R code, discuss what is done and work on interpreting the final graph - in particular comment on the yellow ridge in the plot.

The beetle data has only one covariate (in addition to the intercept) - so this means that we have $\boldsymbol{\beta} = (\boldsymbol{\beta}_0, \boldsymbol{\beta}_1)$. Look at the following code and explain what is done - remark: we have used the $n_i = 1$ version of the loglikelihood here.

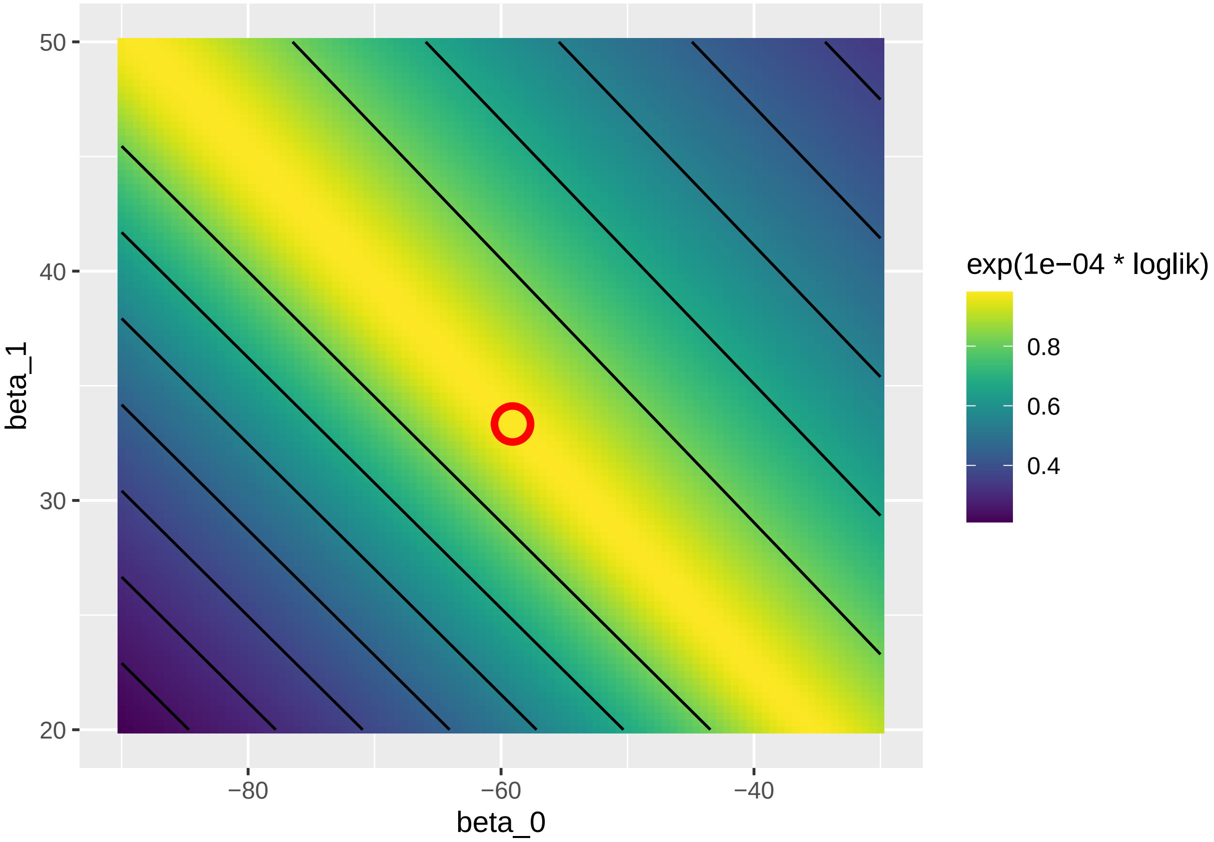
```
library(investr)
library(ggplot2)
library(viridis)

# from aggregated to individual data (because these data were aggregated)
ldose <- rep(investr::beetle$ldose, investr::beetle$n)
y <- NULL
for (i in 1:8) y = c(y, rep(0, investr::beetle$n[i] - investr::beetle$y[i]), rep(1, investr::beetle$y[i
beetleds = data.frame(killed = y, ldose = ldose)

loglik <- function(par, args){
  y <- args$y; x <- args$x; n <- args$n
  res <- sum(y*x%*%par - n*log(1 + exp(x%*%par)))
  return(res)
}

loglik(c(1,1), args = list(y = beetleds$killed,
                           x = cbind(rep(1, nrow(beetleds)), beetleds$ldose),
                           n = rep(1, nrow(beetleds))))
```

```
## [1] -549.2543
```

```
loglikmat <- matrix(NA, nrow = 100, ncol = 100)
loglikframe <- data.frame()
beta_0 <- seq(-90,-30, length.out = 100)
beta_1 <- seq(20, 50, length.out = 100)

for (i in 1:length(beta_0)){
  for (j in 1:length(beta_1)){

    loglikmat[i,j] <- loglik(c(beta_0[i], beta_1[j]), args = list(y = beetleds$killed,
                                                                   x = cbind(rep(1, nrow(beetleds)), bee
                                                                   n = rep(1, nrow(beetleds))))

    loglikframe <- rbind(loglikframe, c(beta_0[i], beta_1[j], loglikmat[i,j]))

  }
}
names(loglikframe) <- c("beta_0", "beta_1", "loglik")
head(loglikframe)
```

```
##   beta_0   beta_1    loglik
## 1    -90 20.00000 -15545.83
## 2    -90 20.30303 -15384.56
## 3    -90 20.60606 -15223.28
## 4    -90 20.90909 -15062.01
## 5    -90 21.21212 -14900.73
## 6    -90 21.51515 -14739.46
```

```
ggplot(data = loglikframe, mapping = aes(x = beta_0, y = beta_1, z = loglik)) + geom_raster(aes(fill = 
  geom_point(data = loglikframe[which.max(loglikframe$loglik),], mapping = aes(x = beta_0, y = beta_1),
             size = 5, col = "red", shape = 21, stroke = 2) + scale_shape(solid = FALSE) +
  scale_fill_viridis() + geom_contour(col = "black")
```



Comments to the code: for the `loglik` function we have two arguments: par= the parameters to be estimated, and args=a list with data. The reason for only having these two arguments is that it is easier to use when we later perform optimization (with `optim`) of the loglikelihood to find the ML estimates.

Problem 3: Score function

a) What is the definition of the score function? What is the dimension of the score function?
b) Derive the score function for the logit model (individual data). The result should be

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i(y_i - \pi_i) = \sum_{i=1}^{n} \mathbf{x}_i(y_i - \frac{\exp(\mathbf{x}_i^T\boldsymbol{\beta})}{1+\exp(\mathbf{x}_i^T\boldsymbol{\beta})})$$

c) What do we need the score function for?

Problem 4: Fisher information.

(We did not cover this in the lecture week 1, but we know one of the definitions from Module 2. Either you skip Problem 4 and move to Problem 5, or you look at the section "Properties of the score function", and "The expected Fisher information matrix $F(\boldsymbol{\beta})$" together.)

a) What is the definition of the expected (and the observed) Fisher information matrix? What is the dimension of thise matrix (matrices)?

b) What is the role of these matrices in ML estimation?

c) For the logit model with grouped data the expected and the observed Fisher information matrix are equal and given as

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1 - \pi_j)$$

Where is $\boldsymbol{\beta}$ in this expression?

d) Write the version of the expected Fisher information for individual data (i.e. $n_j = 1$ and $G = n$).

Problem 5: Maximum likelihood

To find the ML estimate for $\boldsymbol{\beta}$ we may either use the function `glm` or optimize the log-likelihood manually. We will do both.

a) First we use the `glm` function in R, and we also check that the individual and the grouped data give the same parameter estimates for the $\boldsymbol{\beta}$. Read the R-code, notice the different input structures and check the results.

```
# the beetle.ds was made above
fitind=glm(killed ~ ldose, family = "binomial", data = beetleds) # individual data
summary(fitind)
```

```
##
## Call:
## glm(formula = killed ~ ldose, family = "binomial", data = beetleds)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      5.181  -11.72   <2e-16 ***
## ldose         34.270      2.912   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.44  on 480  degrees of freedom
## Residual deviance: 372.47  on 479  degrees of freedom
## AIC: 376.47
##
## Number of Fisher Scoring iterations: 5
```

```
fitgrouped=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle) # grouped data. resp
summary(fitgrouped)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ ldose, family = "binomial", data = investr::beetle)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      5.181  -11.72   <2e-16 ***
## ldose         34.270      2.912   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
##
## Number of Fisher Scoring iterations: 4
```

b) What is the default convergence criterion for the glm? (Note: IRWLS used in `glm` - more in Module 5.)

c) We implemented the log-likelihood as a function in item 2 above. Now we will use this together with the `optim` function on the beetle data set to optimize the loglikelihood. Read the R-code, take notice of how we put data into the `args` slot and how the optimization is called with `optim`. (In Compulsory exercise 2 you will use this in a Poisson regression.)

```
loglik_gr <- function(par, args) {

    y <- args$y
    x <- args$x
    n <- args$n

    res <- y %*% x - t(t(n * x) %*% ((1 + exp(-x %*% par))^(-1)))
    return(res)
}

opt <- optim(c(-60, 30), fn = loglik, gr = loglik_gr, args = list(y = beetleds$killed,
    x = cbind(rep(1, nrow(beetleds)), beetleds$ldose), n = rep(1, nrow(beetleds))),
    control = list(fnscale = -1), hessian = TRUE,method="BFGS")
opt
```

```
## $par
## [1] -60.71748  34.27034
##
## $value
## [1] -186.2354
##
## $counts
## function gradient
##       24        9
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##            [,1]      [,2]
## [1,]  -58.48417 -104.0105
## [2,] -104.01047 -185.0941
```

```
sqrt(diag(-solve(opt$hessian))) # calculate the standard deviations of the parameters
```

```
## [1] 5.180709 2.912138
```

Problem 6: Interpreting results

a) Interpret the estimated $\boldsymbol{\beta}$´s. Odds ratio is useful for this.
b) Plot the predicted probability of a beetle dying against the dosage and discuss what you see. (Yes, since this is the last question you may try to program by yourself!)

SECOND WEEK

Remember the beetle and infant respitory disease examples?

First, we look back at the model requirements for the binary regression - and the loglikelihood and score function.

Likelihood and derivations thereof - continued

Individual data (not grouped):

Loglikelihood:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n}[y_i \ln \pi_i - y_i \ln(1-\pi_i) + \ln(1-\pi_i)]$$

Score function:

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i (y_i - \pi_i)$$

Properties of the score function

Since the score function depends on $Y_i = y_i$ we may regard the score function as a random vector. We will now calculate the mean and covariance matrix for the score function. The expected value is

$$E(s(\boldsymbol{\beta})) = E(\sum_{i=1}^{n}(Y_i - \pi_i)\mathbf{x}_i) = \sum_{i=1}^{n} E((Y_i - \pi_i)\mathbf{x}_i) = \sum_{i=1}^{n}(E(Y_i) - \pi_i)\mathbf{x}_i = 0$$

as $E(Y_i) = \pi_i$. We also see that $E(s_i(\boldsymbol{\beta})) = E((Y_i - \pi_i)\mathbf{x}_i) = 0 \; \forall i$.

The expected Fisher information matrix $F(\boldsymbol{\beta})$

The covariance of $s(\boldsymbol{\beta})$ is called the expected Fisher information matrix, $F(\boldsymbol{\beta})$ and is given by

$$\begin{aligned}
F(\boldsymbol{\beta}) = \text{Cov}(s(\boldsymbol{\beta})) &= \sum_{i=1}^{n} \text{Cov}(s_i(\boldsymbol{\beta})) && (4)\\
&= \sum_{i=1}^{n} E\left[\Big(s_i(\boldsymbol{\beta}) - E(s_i(\boldsymbol{\beta}))\Big)\Big(s_i(\boldsymbol{\beta}) - E(s_i(\boldsymbol{\beta}))\Big)^T\right] && (5)\\
&= \sum_{i=1}^{n} E(s_i(\boldsymbol{\beta}) s_i(\boldsymbol{\beta})^T) = \sum_{i=1}^{n} F_i(\boldsymbol{\beta}) && (6)
\end{aligned}$$

where it is used that the responses Y_i and Y_j are independent, and that $E(s_i(\boldsymbol{\beta})) = 0 \; \forall i$.

Remember that $s_i(\boldsymbol{\beta}) = (Y_i - \pi_i)\mathbf{x}_i$, then:

$$\begin{aligned} F_i(\boldsymbol{\beta}) &= E(s_i(\boldsymbol{\beta})s_i(\boldsymbol{\beta})^T) = E((Y_i - \pi_i)\mathbf{x}_i(Y_i - \pi_i)\mathbf{x}_i^T) \\ &= \mathbf{x}_i\mathbf{x}_i^T E((Y_i - \pi_i)^2) = \mathbf{x}_i\mathbf{x}_i^T \pi_i(1 - \pi_i) \end{aligned}$$

where $E((Y_i - \pi_i)^2) = \text{Var}(Y_i) = \pi_i(1 - \pi_i)$ is the variance of Y_i. Thus

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T \pi_i(1 - \pi_i).$$

A useful relationship: Under mild regularity conditions (so we can change the order of $\int$ and $\frac{\partial}{\partial \boldsymbol{\beta}}$):

$$\text{Cov}(s(\boldsymbol{\beta}) = F(\boldsymbol{\beta}) = E\left(-\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T}\right) = E(-\text{Hessian matrix of } l)$$

which relates the expected to the observed Fisher information matrix.

Do you want to see an explanation?

Observed Fisher information matrix $H(\boldsymbol{\beta})$

$$H(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^T} = \frac{\partial}{\partial \boldsymbol{\beta}^T}\left[\sum_{i=1}^{n}(\pi_i - y_i)\mathbf{x}_i\right]$$

because $s(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \pi_i)\mathbf{x}_i$ and hence $-s(\boldsymbol{\beta}) = \sum_{i=1}^{n}(\pi_i - y_i)\mathbf{x}_i$. Note that $\pi_i = \pi_i(\boldsymbol{\beta})$.

$$H(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\beta}^T}[\mathbf{x}_i\pi_i - \mathbf{x}_i y_i] = \sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\beta}^T}\mathbf{x}_i\pi_i = \sum_{i=1}^{n} \mathbf{x}_i \frac{\partial \pi_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \boldsymbol{\beta}^T}$$

Use that

$$\frac{\partial \eta_i}{\partial \boldsymbol{\beta}^T} = \frac{\partial \mathbf{x}_i^T \boldsymbol{\beta}}{\partial \boldsymbol{\beta}^T} = \left(\frac{\partial \mathbf{x}_i^T \boldsymbol{\beta}}{\partial \boldsymbol{\beta}}\right)^T = \mathbf{x}_i^T$$

and

$$\begin{aligned} \frac{\partial \pi_i}{\partial \eta_i} = \frac{\partial \left(\frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right)}{\partial \eta_i} &= \frac{(1 + \exp(\eta_i))\exp(\eta_i) - \exp(\eta_i)\exp(\eta_i)}{(1 + \exp(\eta_i))^2} \\ &= \pi_i(1 - \pi_i). \end{aligned}$$

And thus

$$H(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i\pi_i(1 - \pi_i)\mathbf{x}_i^T = \sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^T \pi_i(1 - \pi_i).$$

Note that the observed and the expected Fisher information matrix are equal (see below - canonical link - that this is a general finding). This is not the case for the probit or complementary log-log models.

Overview of the results for individual and grouped data

- Individual data: $i = 1, \ldots, n$, and pairs $(\mathbf{x}_i, y_i)$.
- Grouped data: $j = 1, \ldots, G$ with n_j observations for group j, and $Y_j = \sum Y_i$ for all i member of group j. In total $\sum_{j=1}^{G} n_j$ observations. For each pair $(\mathbf{x}_j, y_j)$, where $\mathbf{x}_j$ the covariate pattern for group j.

NB: we keep that $\eta_i = \ln(\frac{\pi_i}{1-\pi_i})$ - not changed for grouped data (but now $\mu_j = n_j \pi_j$).

Log-likelihood:

Individual:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \ln \pi_i - y_i \ln(1-\pi_i) + \ln(1-\pi_i)]$$

Grouped:

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{G} [y_j \ln \pi_j - y_j \ln(1-\pi_j) + n_j \ln(1-\pi_j) + \ln \binom{n_j}{y_j}]$$

The last part is usually not include in calculations.

Score function:

Individual:

$$s(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i (y_i - \pi_i)$$

Grouped:

$$s(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j (y_j - n_j \pi_j)$$

Expected Fisher information matrix:

Individual:

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T \pi_i (1-\pi_i)$$

Grouped:

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1-\pi_j)$$

The observed Fisher information matrix equals the expected Fisher information matrix - because the logit model is the *canonical link* for the binomial distribution.

Look back at MLR - what is $s(\boldsymbol{\beta})$ and $F(\boldsymbol{\beta})$ then?

1. $Y_i \sim \mathrm{N}(\mu_i, \sigma^2)$
2. $\eta_j = x_i^T \boldsymbol{\beta}$
3. $\mu_i = \eta_i$ (identity response function and link function)

Likelihood:

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

Loglikelihood:

$$l(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - \mathbf{X}\boldsymbol{\beta})^T(y - \mathbf{X}\boldsymbol{\beta})$$

Since $(y - \mathbf{X}\boldsymbol{\beta})^T(y - \mathbf{X}\boldsymbol{\beta}) = Y^TY - 2Y^T\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}$, then

$$s(\boldsymbol{\beta}) = \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\frac{1}{2\sigma^2}(2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^TY) = \frac{1}{\sigma^2}(\mathbf{X}^TY - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta})$$

and $s(\hat{\boldsymbol{\beta}}) = 0$ gives $\mathbf{X}^TY - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = 0$ which can be solved on closed form giving $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^TY$. So, no need for iterative methods.

Finally, observed Fisher information matrix.

$$H(\boldsymbol{\beta}) = \frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^T} = -\frac{\partial}{\partial \boldsymbol{\beta}_T}(\frac{1}{\sigma^2}\mathbf{X}^TY - \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta}) = \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}$$

which is independent on $\boldsymbol{\beta}$, and also we see that $F(\boldsymbol{\beta}) = \mathrm{E}(H(\boldsymbol{\beta})) = H(\boldsymbol{\beta})$ since no random variables are present. The identity link is also the canonical link. Finally, the (asymptotic) covariance of the ML estimate is $F^{-1}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$ which we know as $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$.

Exponential family - and canonical link

In Module 1 we introduced distributions of the Y_i, that could be written in the form of a *univariate exponential family*

$$f(y_i \mid \theta_i) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi} \cdot w_i + c(y_i, \phi, w_i)\right)$$

where

- θ_i is called the canonical parameter and is a parameter of interest
- ϕ is called a nuisance parameter (and is not of interest to us=therefore a nuisance (plage))
- w_i is a weight function, in most cases $w_i = 1$
- b and c are known functions.

It can be shown that $\mathrm{E}(Y_i) = b'(\theta_i)$ and $\mathrm{Var}(Y_i) = b''(\theta_i) \cdot \frac{\phi}{w_i}$.

In Module 1 we found that the binomial distribution $Y_i \sim \text{bin}(1, \pi_i)$ is an exponential family (derivation from Module 1: https://www.math.ntnu.no/emner/TMA4315/2017h/Module1ExponentialFamily.pdf)

and that

- $\theta_i = \ln(\frac{\pi_i}{1-\pi_i})$ is the canonical parameter
- $\phi = 1$, no nuisance
- $w_i = 1$
- $b(\theta_i) = \ln(1 + \exp(\theta_i))$

Recall that in a GLM we choose a link function g, linking the linear predictor and the mean: $\eta_i = g(\mu_i)$. For the logit model we had that $\eta_i = \ln(\frac{\pi_i}{1-\pi_i})$.

Now (new to us) - every exponential family has a unique *canonical link function* such that

$$\theta_i = \eta_i$$

Since $\eta_i = g(\mu_i)$ this means to us that we need

$$g(\mu_i) = \theta_i$$

to have a canonical link.

Q: Is the logit link the canonical link for the binary model?

A:

Yes, since $\theta_i = \ln(\frac{\pi_i}{1-\pi_i}) = g(\pi_i)$ then the logit link is the canonical link for the binary regression.

##Properties of a GLM with canonical link

1. The log-likelihod is always concave so that the ML estimated is always unique (given that it exists).
2. The observed Fisher information matrix $H(\boldsymbol{\beta})$ *equals* the expected Fisher information matrix $F(\boldsymbol{\beta})$. That is,
$$-\frac{\partial^2 l}{\partial \boldsymbol{\beta} \boldsymbol{\beta}^T} = \text{E}(-\frac{\partial^2 l}{\partial \boldsymbol{\beta} \boldsymbol{\beta}^T})$$

Proving this is beyond the scope of this course.

Parameter estimation - in practise

To find the ML estimate $\hat{\boldsymbol{\beta}}$ we need to solve

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

We have that the score function for the logit model is:

$$s(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j (y_j - n_j \pi_j)$$

where $\pi_j = \frac{\exp(\mathbf{x}_j^T \boldsymbol{\beta})}{1+\exp(\mathbf{x}_j^T \boldsymbol{\beta})}$. Observe that this is a non-linear function in $\boldsymbol{\beta}$, and has no closed form solution.

##Iterative gradient-based methods

Back to the general case - we may use a first order multivariate Taylor approximation for $s(\hat{\boldsymbol{\beta}})$, around some chosen reference value $\boldsymbol{\beta}^{(0)}$:

$$s(\hat{\boldsymbol{\beta}}) \approx s(\boldsymbol{\beta}^{(0)}) + \frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(0)}}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{(0)})$$

Let $H(\boldsymbol{\beta}^{(0)}) = -\frac{\partial s(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\big|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(0)}}$. Setting $s(\hat{\boldsymbol{\beta}}) = 0$ solving for $\hat{\boldsymbol{\beta}}$ gives

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(0)} + H(\boldsymbol{\beta}^{(0)})^{-1} s(\boldsymbol{\beta}^{(0)})$$

where $H(\boldsymbol{\beta}^{(0)})^{-1}$ is the matrix inverse of $H(\boldsymbol{\beta}^{(0)})$.

If we start with some value $\boldsymbol{\beta}^{(0)}$ and then find a new value $\boldsymbol{\beta}^{(1)}$ by applying this equation, and then continue applying the equation until convergence we have the *Newton-Raphson* method:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + H(\boldsymbol{\beta}^{(t)})^{-1} s(\boldsymbol{\beta}^{(t)})$$

Replacing the observed Fisher information matrix $\mathbf{H}$ with the expected Fisher information matrix $\mathbf{F}$ yields the *Fisher-scoring* method.

For the logit model these two methods are the same since the observed and expected Fisher information matrix is the same for canonical link functions (like the logit is for binary regression).

This algorithm is run until the relative difference in Euclidean distance between two iterations "(new-old)/old" is smaller than some chosen constant.

##Requirements for convergence

For the Newton-Raphson algorithm we see that the observed Fisher information matrix H needs to be invertible for all $\boldsymbol{\beta}$, alternatively for the Fisher scoring algorithm the expected Fisher information matrix F needs to be invertible.

In our logit model

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1 - \pi_j)$$

Let $\mathbf{X}$ be the design matrix, where the rows are $\mathbf{x}_j^T$. Then $\mathbf{X}^T\mathbf{X} = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T$.

If we require that the design matrix has full rank (G) then also $\mathbf{X}^T\mathbf{X}$ will have full rank (it will also be positive definite) and and in addition $\pi_j(1-\pi_j) > 0$ for all $\pi_j \in (0,1)$, so then $F(\boldsymbol{\beta})$ will be positive definite and all is good.

Why is $F(\boldsymbol{\beta})$ positive definite if we require that the design matrix has full rank? Formally, let $\mathbf{X}$ be a $n \times p$ matrix and Λ a $n \times n$ diagonal matrix where all the diagonal elements are positive (like our $\pi_j(1-\pi_j)$, yes, put them on the diagonal). Let $\mathbf{X}$ have independent columnes (full rank) $\Leftrightarrow$ $\mathbf{X}^T\Lambda\mathbf{X}$ is positive definite.

Proof: $\Rightarrow$: Let $\mathbf{v}$ be a p dimensional column vector. Assume $0 = \mathbf{v}^T\mathbf{X}^T\Lambda\mathbf{X}\mathbf{v} = (\Lambda^{1/2}\mathbf{X}\mathbf{v})^T(\Lambda^{1/2}\mathbf{X}\mathbf{v}) = \sum_{i=1}^{n} w_i^2$ where $\mathbf{W} = \Lambda^{1/2}\mathbf{X}\mathbf{v}$. Then, w must be 0, that is $\Lambda^{1/2}\mathbf{X}\mathbf{v} = 0$ since multiplication with $\Lambda^{1/2}$ is to multiply each element in $\mathbf{X}\mathbf{v}$ with a number different from 0. That is, we must have $\mathbf{v} = \mathbf{0}$ since $\mathbf{X}$ has independent columns.

$\Leftarrow$: Assume that $\mathbf{X}\mathbf{v} = \mathbf{0}$. Then $\mathbf{v}^T\mathbf{X}^T\Lambda\mathbf{X}\mathbf{v} = \mathbf{0}$ so $\mathbf{v} = \mathbf{0}$ since $\mathbf{X}^T\Lambda\mathbf{X}$ is positive definite. This is, $\mathbf{X}$ has independent columns.

End of proof

Therefore, for GLMs we will also - as for the multiple linear regression model in Module 2 - assume that the design matrix has full rank!

We will see in Module 5 that this is the requirement needed for GLMs in general.

However, it is possible that the algorithm does not converge. This may happen for "unfavorable" data configurations (especially for small samples). Accoring to our text book, Fahrmeir et al (2013), page 284, the conditions for uniqueness and existence of ML estimators are very complex, and the authors suggest that the GLM user instead checks for convergence in practice by performing the iterations.

Asymptotic properties of ML estimates

Results

Under some (weak) regularity conditions (including that $\boldsymbol{\beta}$ falls in the interior of the parameter space and p is fixed that n increases, Agresti (2015) page 125):

Let $\hat{\boldsymbol{\beta}}$ be the maximum likelihood (ML) estimate in the GLM model. As the total sample size increases, $n \to \infty$:

1. $\hat{\boldsymbol{\beta}}$ exists
2. $\hat{\boldsymbol{\beta}}$ is consistent (convergence in probability, yielding asymptotically unbiased estimator, variances goes towards 0)
3. $\hat{\boldsymbol{\beta}} \approx N_p(\boldsymbol{\beta}, F^{-1}(\hat{\boldsymbol{\beta}}))$

Observe that this means that asymptotically $\text{Cov}(\hat{\boldsymbol{\beta}}) = F^{-1}(\hat{\boldsymbol{\beta}})$: the inverse of the expected Fisher information matrix evaluated at the ML estimate.

Observe: The result requires that the *total* sample size goes to infinity (not the individual n_j for the covariate patterns).

The *proof* (for the univariate case) is given in the course TMA4295 Statistical Inference course, Casella and Berger (2002):"Statistical inference", page 472. It starts by a first order Taylor expansion of the score function (derivative of loglikelihood) around the true parameter, and utilizes the fact that the maximum likelihood estimate is defined as the zero of the score function.

The following is not a formal proof, but a sketch - and I use the parameter of interest $\boldsymbol{\theta}$ in the exponential family version of the distribution (and then there is a connection to the mean $\boldsymbol{\mu}$ and then to $\boldsymbol{\eta}$ and finally $\boldsymbol{\beta}$):

We start with the multivariate version of the first order Taylor expansion around the true parameter value $\boldsymbol{\theta}$:

$$\mathbf{0} = s(\hat{\boldsymbol{\theta}}) \approx s(\boldsymbol{\theta}) - \mathbf{H}(\boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$$

We assume that $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate, and there the score function is $\mathbf{0}$ so we get:

$$s(\boldsymbol{\theta}) \approx \mathbf{H}(\boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$$

And premultiplying with $\mathbf{H}^{-1}(\boldsymbol{\theta})$ gives

$$(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \approx \mathbf{H}^{-1}(\boldsymbol{\theta})s(\boldsymbol{\theta})$$

Then, to use the central limit theorem we need some smart manipulations with n, so we start by multiplying with $\sqrt{n}$ and split that into n and $\frac{1}{\sqrt{n}}$.

$$\sqrt{n}(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}) \approx \sqrt{n}\mathbf{H}^{-1}(\boldsymbol{\theta})s(\boldsymbol{\theta}) = (\frac{1}{n}\mathbf{H}(\boldsymbol{\theta}))^{-1}\frac{1}{\sqrt{n}}s(\boldsymbol{\theta})$$

From the central limit theorem:

1) $\frac{1}{n}\mathbf{H}(\boldsymbol{\theta})$ goes to the expected value which is $\mathbf{F}(\boldsymbol{\theta})$ (in probability),
2) the part $\frac{1}{\sqrt{n}}s(\boldsymbol{\theta})$ asymptoticalle goes to a random variable W that follows a multivariate normal with
   - mean $\text{E}(\frac{1}{\sqrt{n}}s(\boldsymbol{\theta}))=\mathbf{0}$ and the
   - covariance matrix is $\text{Cov}(\frac{1}{\sqrt{n}}s(\boldsymbol{\theta}))=\frac{1}{n}\mathbf{F}(\boldsymbol{\theta})$

$$\mathbf{W} \sim N(\mathbf{0}, \frac{1}{n}\mathbf{F}(\boldsymbol{\theta}))$$

$$\sqrt{n}(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}) \approx \mathbf{F^{-1}}(\boldsymbol{\theta})\mathbf{W}$$

On the right side here we have a multivariate normal distributed random variable $\mathbf{F^{-1}}(\boldsymbol{\theta})\mathbf{W}$ with mean $\mathbf{0}$ and covariance matrix

$$\text{Cov}(\mathbf{F}^{-1}(\boldsymbol{\theta})\mathbf{W}) = \mathbf{F}^{-1}(\boldsymbol{\theta})\frac{1}{n}\mathbf{F}(\boldsymbol{\theta})\mathbf{F}^{-1}(\boldsymbol{\theta}) = \frac{1}{n}\mathbf{F}^{-1}(\boldsymbol{\theta})$$

This leads to the wanted result:

$$\hat{\boldsymbol{\theta}} \approx N(\boldsymbol{\theta}, \mathbf{F}^{-1}(\boldsymbol{\theta}))$$

Due to the Slutsky theorem (from TMA4295 Statistical inference) this also holds when $\mathbf{F}^{-1}(\boldsymbol{\theta}))$ is replaced by $\mathbf{F}^{-1}(\hat{\boldsymbol{\theta}}))$.

Parameter estimation

Parameter estimation can be based on grouped data - so now we use $Y_j \sim \text{bin}(n_j, \pi_j)$ from 1 above, but keep 2 and 3 unchanged. The number of groups is G and the total number of observations is $\sum_{j=1}^{G} n_j$.

- Likelihood=joint distribution, exponential family.

$$f(y \mid \theta) = \exp\left(\frac{y\theta - b(\theta)}{\phi}\cdot w + c(y,\phi,w)\right)$$

  where we have that $\theta = \ln(\frac{\pi}{1-\pi})$ for the binomial distribution, which means that our logit model is gives the canonical link (remember, good properties!).

- Log-likelihood

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{G}[y_j \ln \pi_j - y_j \ln(1-\pi_j) + n_j \ln(1-\pi_j) + \ln \binom{n_j}{y_j}]$$

- Score function=vector of partial derivatives of log-likelihood. Find ML by solving $s(\hat{\boldsymbol{\beta}})) = 0$ - but no closed form solutions.

$$s(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j (y_j - n_j \pi_j)$$

- Expected Fisher information matrix

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1 - \pi_j)$$

- $\hat{\boldsymbol{\beta}}$ found iteratively using Newton-Raphson or Fisher scoring

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + F(\boldsymbol{\beta}^{(t)})^{-1} s(\boldsymbol{\beta}^{(t)})$$

- $\hat{\boldsymbol{\beta}} \approx N_p(\boldsymbol{\beta}, F^{-1}(\hat{\boldsymbol{\beta}}))$

Further statistical inference

Our further statistical inference (confidence intervals and hypotheses tests) are based on the asymptotic distribution of the parameter estimates

$$\hat{\boldsymbol{\beta}} \approx N_p(\boldsymbol{\beta}, F^{-1}(\hat{\boldsymbol{\beta}}))$$

where $F^{-1}(\hat{\boldsymbol{\beta}}))$ is the inverse of the expected Fisher information matrix inserted $\hat{\boldsymbol{\beta}}$.

For the logit model we found that

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1 - \pi_j)$$

So we would need to do $\pi_j = \frac{\exp(\eta - j)}{1 + \exp(\eta_j)}$ and $\eta_j = \mathbf{x}_j^T \boldsymbol{\beta}$ as "usual", and then replace $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$.

The asymptotic distribution still holds when we replace $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$ in $\mathbf{F}$.

If we make a diagonal matrix $\mathbf{W}$ with $n_j \pi_j (1 - \pi_j)$ on the diagonal, then we may write the matrix $F(\boldsymbol{\beta})$ in matrix notation. As before $\mathbf{X}$ is the $G \times p$ design matrix.

$$F(\boldsymbol{\beta}) = \sum_{j=1}^{G} \mathbf{x}_j \mathbf{x}_j^T n_j \pi_j (1 - \pi_j) = \mathbf{X}^T \mathbf{W} \mathbf{X}.$$

which means that $\text{Cov}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1}$ for the binomial model (remember that $\hat{\boldsymbol{\beta}}$ comes in with $\hat{\pi}_j$ in $\mathbf{W}$).

Q: How is this compared to the normal case?

A: $F(\boldsymbol{\beta}) = \frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X}$, and the inverse $\text{Cov}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2$.

Let $\mathbf{A}(\boldsymbol{\beta}) = F^{-1}(\hat{\boldsymbol{\beta}})$, and $a_{kk}(\hat{\boldsymbol{\beta}})$ is diagonal element number k.

For one element of the parameter vector:

$$Z_k = \frac{\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_k}{\sqrt{\hat{a}_{kk}(\boldsymbol{\beta})}}$$

is asymptotically standard normal. We will use this now!

Q: Where can you find $\hat{\boldsymbol{\beta}}$ and $F^{-1}(\hat{\boldsymbol{\beta}})$ in the print-out below?

```r
library(investr)
fitgrouped=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle)
summary(fitgrouped)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ ldose, family = "binomial", data = investr::beetle)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      5.181  -11.72   <2e-16 ***
## ldose         34.270      2.912   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
##
## Number of Fisher Scoring iterations: 4
```

```r
summary(fitgrouped)$cov.scaled
```

```
##             (Intercept)      ldose
## (Intercept)    26.83966 -15.082090
## ldose         -15.08209   8.480525
```

```r
sqrt(diag(summary(fitgrouped)$cov.scaled))
```

```
## (Intercept)       ldose
##    5.180701    2.912134
```

Note: `cov.unscaled` is the estimated covariance matrix for the estimated coefficients, and `cov.scaled` is the the `cov.unscaled` scaled with the dispersion parameter. For the binomial the dispersion is equal to 1, so no difference between the two.

Confidence intervals

In addition to providing a parameter estimate for each element of our parameter vector $\boldsymbol{\beta}$ we should also report a $(1-\alpha)100\%$ confidence interval (CI) for each element.

We focus on element k of $\boldsymbol{\beta}$, called $\boldsymbol{\beta}_k$. It is known that asympotically

$$Z_k = \frac{\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_k}{\sqrt{a_{kk}(\hat{\beta})}}$$

is asymptotically standard normal. We use that to form confidence intervals.

Let $z_{\alpha/2}$ be such that $P(Z_k > z_{\alpha/2}) = \alpha/2$. REMARK: our textbook would here look at area to the left instead of to the right - but we stick with this notation.

We then use

$$P(-z_{\alpha/2} \leq Z_k \leq z_{\alpha/2}) = 1 - \alpha$$

insert Z_k and solve for β_k to get

$$P(\hat{\beta}_k - z_{\alpha/2}\sqrt{a_{kk}(\hat{\beta})} \leq \beta_k \leq \hat{\beta}_k - z_{\alpha/2}\sqrt{a_{kk}(\hat{\boldsymbol{\beta}})}) = 1 - \alpha$$

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

Q: We write $a_{kk}(\hat{\boldsymbol{\beta}})$. Why not $a_{kk}(\hat{\beta}_{kk})$?

Example with the beetle data

Again, we study our beetle data - in the grouped version.

Here we calculate the upper and lower limits of the confidence interval using the formula. Then, there is also an R function `confint.glm` that can be used. This function may give a slightly different answer to our calculations because here an extra "profiling" step is done to check the convergence of the glm, and to recalculate the estimated covariance matrix for the regression parameter estimate.

```r
fitgrouped=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle)
coeff=fitgrouped$coefficients
sds=sqrt(diag(summary(fitgrouped)$cov.scaled))
alpha=0.05
lower=coeff-qnorm(1-alpha/2)*sds
upper=coeff+qnorm(1-alpha/2)*sds
cbind(lower,upper)
```

```
##                  lower     upper
## (Intercept) -70.87144 -50.56347
## ldose        28.56265  39.97800
```

```r
confint(fitgrouped)
```

```
##                  2.5 %    97.5 %
## (Intercept) -71.44263 -51.07902
## ldose        28.85403  40.30069
```

Q: Explain what is done in the R-print-out.

Hypothesis testing

There are three methods that are mainly used for testing hypotheses in GLMs - these are called Wald test, likelihood ratio test and score test. We will look at the first two.

First, look at linear hypotheses: We study a binary regression model with $p = k + 1$ covariates, and refer to this as model A (the larger model). As for the multiple linear 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

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

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

The Wald test

The Wald test statistic is given as:

$$w = (\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})^{\mathrm{T}}[\mathbf{C}F^{-1}(\hat{\boldsymbol{\beta}})\mathbf{C}^{\mathrm{T}}]^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})$$

and measures the distance between the estimate $\mathbf{C}\hat{\boldsymbol{\beta}}$ and the value under then null hypothesis $\mathbf{d}$, weighted by the asymptotic covariance matrix of $\mathbf{C}\hat{\boldsymbol{\beta}}$. Remember: $\text{Cov}(\mathbf{C}\hat{\boldsymbol{\beta}}) = \mathbf{C}F^{-1}(\hat{\boldsymbol{\beta}})\mathbf{C}^{\mathrm{T}}$.

Asymptotically it is found that w under the null hypothesis follows a χ^2 distribution with r degrees of freedom (where r is the number of hypotheses tested). Why is that?

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

Observe: to perform the test you only need to fit the larger model (and not the smaller).

For the special case that we only test one regression parameter, for example β_k:

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

Now $\mathbf{C}\hat{\boldsymbol{\beta}} = \beta_k$ and $\mathbf{C}[F(\hat{\boldsymbol{\beta}})]^{-1}\mathbf{C}^{\mathrm{T}} = \mathbf{C}\mathbf{A}(\boldsymbol{\beta})\mathbf{C}^{\mathrm{T}} = a_{kk}(\boldsymbol{\beta})$, and the Wald test becomes

$$(\hat{\beta}_k - \beta_k)[a_{kk}(\hat{\boldsymbol{\beta}})]^{-1}(\hat{\beta}_k - \beta_k) = \left(\frac{\hat{\beta}_k - \beta_k}{\sqrt{a_{kk}(\hat{\boldsymbol{\beta}})}}\right)^2 = Z_k^2$$

so, asymptotically the square of the standard normal, which we know follows a χ^2-distribution with 1 degree of freedom.

Q: Explain what you find in the columns named `z value` and `Pr(>|z|)` below, and which hypothesis tests these are related to. Are the hypothesis tests performed using the Wald test?

```r
library(investr)
fitgrouped=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle)
summary(fitgrouped)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ ldose, family = "binomial", data = investr::beetle)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      5.181  -11.72   <2e-16 ***
## ldose         34.270      2.912   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
##
## Number of Fisher Scoring iterations: 4
```

The likelihood ratio test

An alternative to the Wald test is the likelihood ratio test (LRT), which compares the likelihood of two models.

We stick with the notation of A: the larger model 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 parameter estimate $\hat{\boldsymbol{\beta}}_A$. The maximum likelihood is achieved at this parameter estimate and is denoted $L(\hat{\boldsymbol{\beta}}_A)$.
- Then we maximize the likelihood for model B (the smaller model) and find the parameter estimate $\hat{\boldsymbol{\beta}}_B$. The maximum likelihood is achieved at this parameter estimate and is denoted $L(\hat{\boldsymbol{\beta}}_B)$.

The likelihood of the larger model (A) will always be larger or equal to the likelihood of the smaller mode (B). Why? How is this compared to our result for SSE for small and large model in MLR?

The likelihood ratio statistic is defined as

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

(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 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. How different?

For the beetle data we compare model A=model with `ldose` as covariate with model B=model with only intercept. We use the loglikelihood-function that we made for the lecture session for week 2.

```r
library(investr)
fitgrouped=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle)
fitnull=glm(cbind(y, n-y) ~ 1, family = "binomial", data = investr::beetle)

loglik <- function(par, args){
  y <- args$y; x <- args$x; n <- args$n
  res <- sum(y*x%*%par - n*log(1 + exp(x%*%par)))
  return(res)
}

# call this with parameters estimated under model A=larger model
beetleargs = list(y = investr::beetle$y,
            x = cbind(rep(1, nrow(investr::beetle)), investr::beetle$ldose),
            n = investr:: beetle$n)

llA=loglik(matrix(fitgrouped$coefficients,ncol=1),args=beetleargs)

# then the smaller model, then we set the coeff for ldose to 0. B=smaller model
llB=loglik(matrix(c(fitnull$coefficients,0),ncol=1),args=beetleargs)
lrt=-2*(llB-llA)
lrt
```

```
## [1] 272.9702
```

```
pchisq(lrt,df=1)
```

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

```
anova(fitgrouped,fitnull,test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(y, n - y) ~ ldose
## Model 2: cbind(y, n - y) ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1         6     11.232
## 2         7    284.202 -1  -272.97 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q and A: Here the small model is the model with only intercept and the large is the one with dose as covariate. This means that the null hypothesis is that "the small model is preferred" and our p-value is very small, so we reject the null hyptheses and stick with the model with dose as covariate. Observe that the LRT can be performed using `anova`.

Deviance

The *deviance* is used to assess model fit and also for model choice, and is based on the likelihood ratio test statistic.

The derivation assumes that data can be grouped into covariate patterns, with G groups and n_j observations in each group (individual data later).

Saturated model: If we were to provide a perfect fit to our data then we would estimate π_j by the observed frequency for the group: $\tilde{\pi}_j = \frac{y_j}{n_j}$. Then $\tilde{\pi}$ is a G-dimensional column vector with the elements $\tilde{\pi}_j$.

This "imaginary model" is called the *saturated* model. This would be a model where each group was 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 $\hat{\beta}$ which through our linear predictor and link function we turn into estimates for each group $\hat{\pi}_j$. Then $\hat{\pi}$ is a G-dimensional column vector with the elements $\hat{\pi}_j$.

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:

$$D = -2(\ln L(\text{candidate model}) - \ln L(\text{saturated model})) = -2(l(\hat{\pi}) - l(\tilde{\pi})) = -2\sum_{j=1}^{G}(l_j(\hat{\pi}_j) - l_j(\tilde{\pi}_j))$$

For our logit model this can be written as (after some maths):

$$D = 2\sum_{j=1}^{G}[y_j \ln(\frac{y_j}{n_j\hat{\pi}_j}) + (n_j - y_j)\ln(\frac{n_j - y_j}{n_j - n_j\hat{\pi}_j})]$$

Verify this by yourself.

The reasoning behind this is that if our model is good, it should not be too far from the saturated model, and we measure this distance by the deviance.

If we want to investigate the null hypothesis that "our model fits the data well" to the negation, it is useful to know that asymptotically D is distributed as χ^2 with $G - p$ degrees of freedom (same reason as for the likelihood ratio test statistic).

This result depends on that n_j is large, hard to say how large (at least 5 is a rule of thumb).

The deviance is in `summary.glm` outputted as "Residual deviance", which we read off as 11.2322311. Let's check for our beetle example by computing the formula for D directly:

```
D=deviance(fitgrouped)
D
```

```
## [1] 11.23223
```

```
G=dim(investr::beetle)[1]
G
```

```
## [1] 8
```

```
p=2
1-pchisq(D,G-p)
```

```
## [1] 0.08145881
```

So, do we have a good fit?

The null hypothesis is that the candiate model is equally good as the saturated model. We do not reject this hypothesis at level 0.05. This means that we are satisfied with the candidate model.

In the summary from glm also the socalled *NULL deviance* is given. This is the deviance when the candicate model is the model with only intercept term present. This deviance asymptotically distributed as χ^2 with $G - 1$ degrees of freedom.

```
summary(fitgrouped)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ ldose, family = "binomial", data = investr::beetle)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      5.181  -11.72   <2e-16 ***
## ldose         34.270      2.912   11.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
##
## Number of Fisher Scoring iterations: 4
```

Q: where is the deviance(s) here and how do we use these?

Analysis of deviance

In MLR we have seen that we may produce a sequential analysis of variance (Type I) by adding more and more terms to the model and comparing the scaled decrease in SSE by the scaled SSE of a full model.

For the binary regression we may adapt a similar strategy, but with using the scaled change in deviance instead of the SSE.

We use the infant respiratory disease data as an example

```r
library(faraway)
fit=glm(cbind(disease, nondisease)~sex*food,family=binomial(link=logit),data=babyfood)
summary(fit)
```

```
##
## Call:
## glm(formula = cbind(disease, nondisease) ~ sex * food, family = binomial(link = logit),
##     data = babyfood)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.59899    0.12495 -12.797  < 2e-16 ***
## sexGirl            -0.34692    0.19855  -1.747 0.080591 .
## foodBreast         -0.65342    0.19780  -3.303 0.000955 ***
## foodSuppl          -0.30860    0.27578  -1.119 0.263145
## sexGirl:foodBreast -0.03742    0.31225  -0.120 0.904603
## sexGirl:foodSuppl   0.31757    0.41397   0.767 0.443012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.6375e+01  on 5  degrees of freedom
## Residual deviance: 4.2144e-13  on 0  degrees of freedom
## AIC: 43.518
##
## Number of Fisher Scoring iterations: 3
```

```r
anova(fit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
```

```
##
## Response: cbind(disease, nondisease)
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        5    26.3753
## sex       1   5.4761        4    20.8992   0.01928 *
## food      2  20.1772        2     0.7219 4.155e-05 ***
## sex:food  2   0.7219        0     0.0000   0.69701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q: is it recommended (from the test) to add an interaction term to the model? What does it mean that the Residual deviance is 0 for the `sex*food` model?

Deviance residuals

The deviance residuals are given by a signed version of each element in the sum for the deviance, that is

$$d_k = \text{sign}(y_k - n_k\hat{\pi}_k) \cdot \left\{ 2[y_k \ln(\frac{y_k}{n_k\hat{\pi}_k}) + (n_k - y_k)\ln(\frac{n_k - y_k}{n_k - n_k\hat{\pi}_k})] \right\}^{1/2}$$

where the term $\text{sign}(y_k - n_k\hat{\pi}_k)$ makes negative residuals possible.

Model assessment and choice

The fit of the model can be assessed based on goodness of fit statistics (and related tests) and by residual plots. Model choice can be made from analysis of deviance, or by comparing the AIC for different models.

Deviance test for grouped data

We may use the deviance test presented before to test if the model under study is preferred compared to the saturated model.

Pearson test and residuals

An alternative to the deviance test is the Pearson test. We will look in more detail at this test in a Module 4. The Pearson test statistic can be written as a function of the Pearson residuals, which for the binomial regression is given as:

$$r_j = \frac{y_j - n_j\hat{\pi}_j}{\sqrt{n_j\hat{\pi}_j(1-\hat{\pi}_j)}}$$

Remark: A standardized version scales the Pearson residuals with $\sqrt{1-h_{kk}}$ similar to the standardized residuals for the normal model. Here h_{kk} is the diagonal element number k in the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$.

The Pearson χ^2-goodness of fit statistic is given as

$$X_P^2 = \sum_{j=1}^{G} r_j^2 = \sum_{j=1}^{G} \frac{(y_j - n_j\hat{\pi}_j)^2}{n_j\hat{\pi}_j(1-\hat{\pi}_j)}$$

The Pearson χ^2 statistic is asymptotically equivalent to the deviance statistic and thus is asymptotically χ^2_{G-p}.

The Pearson χ^2 statistic is not a good choice if any of the groups have a low expected number of observations, i.e. $n_j\hat{\pi}_j$ is small (below 1).

Model assessment with continuous covariates

If data have continuous covariates it is possible to form groups based making intervals for continuous covariates. Alternatively grouping on predicted probabilites can be done.

For continuous data the Hosmer Lemeshow test can be used - not on our reading list.

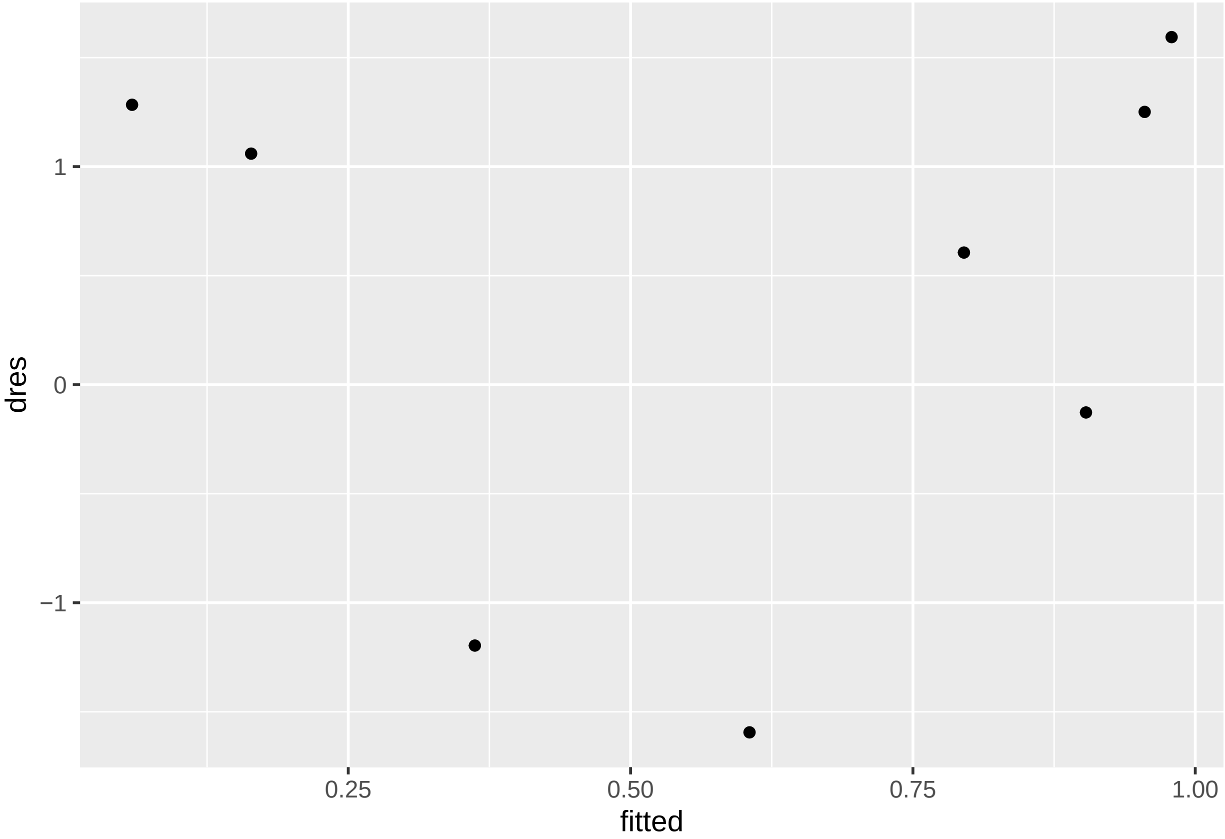
Plotting residuals

Deviance and Pearson residuals can be used for checking the fit of the model, by plotting the residuals against fitted values and covariates.

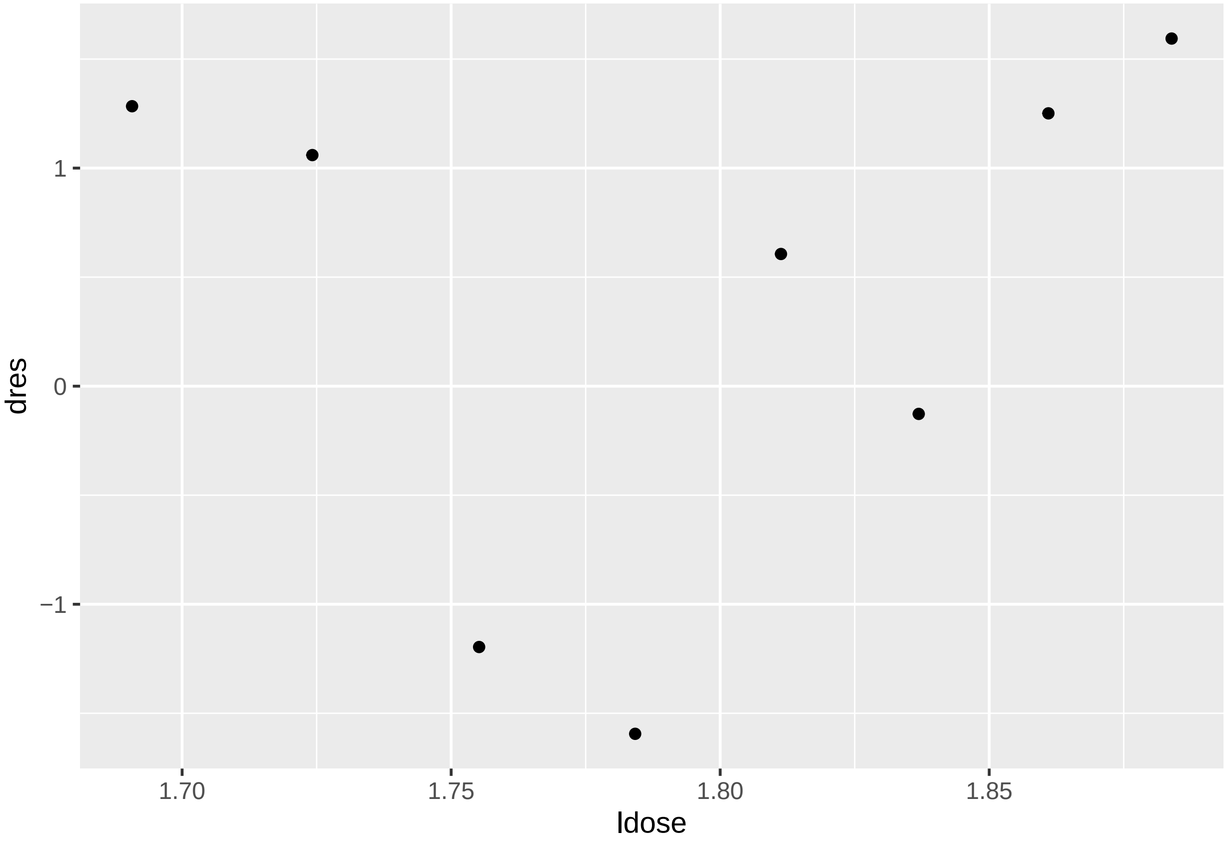
If n_j is small for the covariate patterns the residual plots may be relatively uninformative.

Residual plots for the logistics regression - and for the GLM in general - is highly debated, and we will not put much emphasis on residual plots for this module.

```
df=data.frame("fitted"=fitgrouped$fitted.values,"dres"=residuals(fitgrouped,type="deviance"),"ldose"=inv
ggplot(df,aes(x=fitted,y=dres))+geom_point()
```



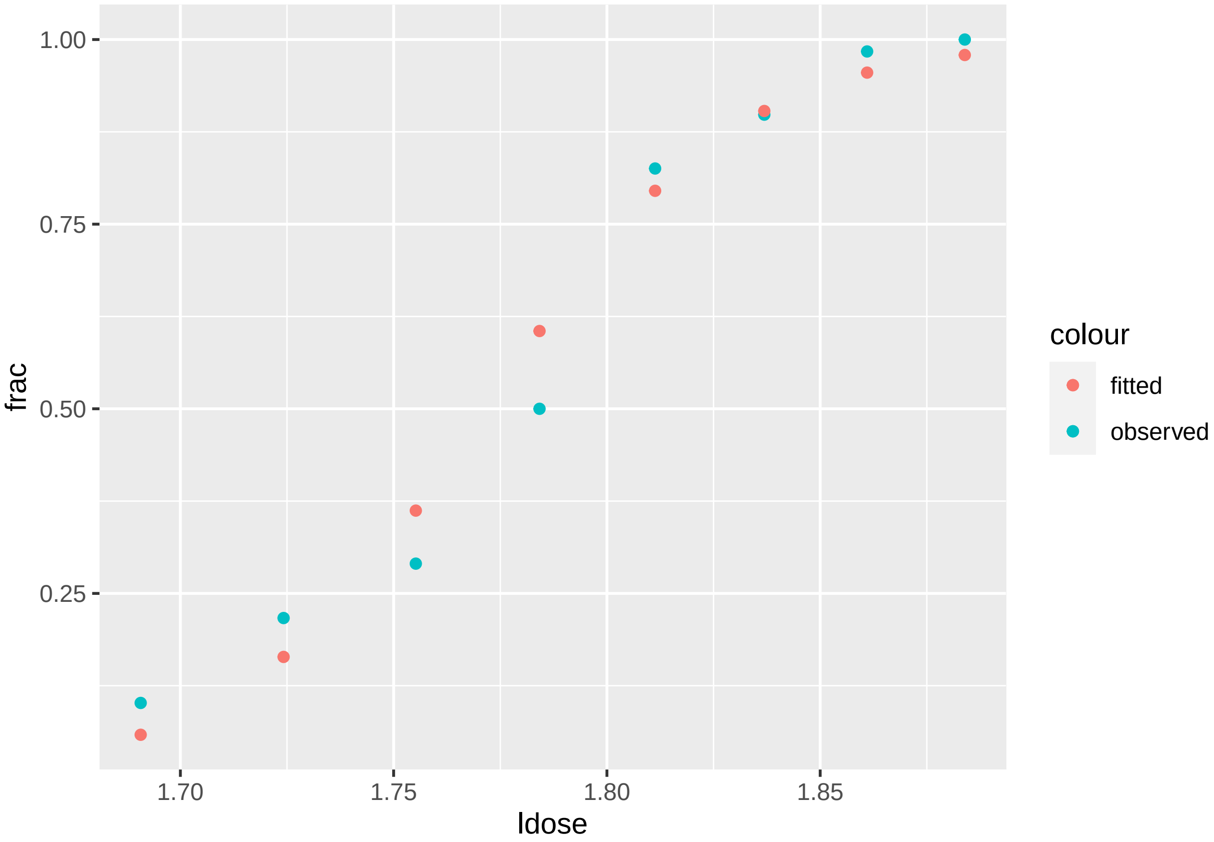
```
ggplot(df,aes(x=ldose,y=dres))+geom_point()
```



Other plots

A useful plot is to show observed and fitted proportions (grouped data) plotted against the linear predictor or covariates.

```
df=data.frame("fitted"=fitgrouped$fitted.values,"dres"=residuals(fitgrouped,type="deviance"),"ldose"=inv
ggplot(df,aes(x=ldose))+geom_point(aes(y=frac,colour="observed"))+geom_point(aes(y=fitted,colour="fitte
```



AIC

It is known to us from multiple linear regression that if a model is chosen based on a goodness of fit statistic (like the SSE or R^2 in multiple linear regression) will in general result in us choosing a to big model (to many parameters fit). The Akaike informations criterion - that we studied for multiple linear regression - can also be used for binary regression: Let p be the number of regression parameters in our model.

$$\text{AIC} = -2 \cdot l(\hat{\boldsymbol{\beta}}) + 2p$$

A scaled version of AIC, standardizing for sample size, is sometimes preferred.

To use AIC for model selection you use the model with the *smallest* AIC.

We may also use the BIC, where $2p$ is replaced by $\log(G) \cdot p$ or $\log(n) \cdot p$.

```
library(faraway)
fit1=glm(cbind(disease, nondisease)~1,family=binomial(link=logit),data=babyfood)
fit2=glm(cbind(disease, nondisease)~sex,family=binomial(link=logit),data=babyfood)
fit3=glm(cbind(disease, nondisease)~food,family=binomial(link=logit),data=babyfood)
fit4=glm(cbind(disease, nondisease)~food+sex,family=binomial(link=logit),data=babyfood)
fit5=glm(cbind(disease, nondisease)~food*sex,family=binomial(link=logit),data=babyfood)
AIC(fit1,fit2,fit3,fit4,fit5)
```

```
##      df      AIC
## fit1  1 59.89324
## fit2  2 56.41710
## fit3  3 43.21693
## fit4  4 40.23987
## fit5  6 43.51795
```

Q: Which of these 5 models would you prefer?

Overdispersion and estimating overdispersion parameter

When we have grouped data: $Y_j \sim \text{Bin}(n_j, \pi_j)$ and $\text{Var}(Y_j) = n_j\pi_j(1-\pi_j)$.

It is possible to estimate the variance (within a group) by $n_j\bar{y_j}(1-\bar{y_j})$ where $\bar{y_j} = y_j/n_j$ (this is an estimate of π_j for group j). We call this the *empirical variance*.

In a logistic regresson we estimate $\hat{\pi}_j = h(\mathbf{x}_j^T\hat{\boldsymbol{\beta}})$ ($h(\cdot)$ is the inverse link function) which is

$$\hat{\pi_j} = \frac{\exp(x_j^T\hat{\boldsymbol{\beta}})}{1+\exp(x_j^T\hat{\boldsymbol{\beta}})}$$

for a logistic regression. This would give the *estimated binomial variance* for Y_j as $n_j\hat{\pi_j}(1-\hat{\pi_j})$.

Some times the empirical variance is much larger than the estimated binomial variance of the model. This is called *overdispersion* and may occur when the individual responses within the groups are correlated, or when the model could be improved upon (missing/unobserved covariates?).

Positively correlated binary variables will give a variance of the sum that is larger than for uncorrelated variables, e.g.

$$\text{Var}(\sum_{k=1}^{K} Y_k) = \sum_{k=1}^{K}\text{Var}(Y_k) + 2\sum_{k<l}\text{Cov}(Y_k, Y_l).$$

This can be handeled by including an *overdispersion parameter*, named ϕ, in the variance formula:

$$\text{Var}(Y_j) = \phi n_j\pi_j(1-\pi_j)$$

The overdispersion parameter can be estimated as the average Pearson statistic or average deviance

$$\hat{\phi}_D = \frac{1}{G-p}D$$

where D is the deviance. Note that similarity to $\hat{\sigma^2} = 1/(n-p)\cdot\text{SSE}$ in the MLR. The $\text{Cov}(\hat{\boldsymbol{\beta}})$ can then be changed to $\hat{\phi}F^{-1}(\hat{\boldsymbol{\beta}})$.

Remark: We are now moving from likelihood to quasi-likelihood theory, where only $\text{E}(Y_j)$ and $\text{Var}(Y_j)$ - and not the distribution of Y_j - are used in the estimation.

In Modules 7 and 8 we will look at using multilevel models to handle correlated observations.

```r
library(investr)
estpi=investr::beetle$y/investr::beetle$n
empvars=investr::beetle$n*estpi*(1-estpi)
fit=glm(cbind(y, n-y) ~ ldose, family = "binomial", data = investr::beetle)
modelestvar=investr::beetle$n*fit$fitted.values*(1-fit$fitted.values)
cbind(empvars,modelestvar)
```

```
##     empvars modelestvar
## 1  5.389831    3.254850
## 2 10.183333    8.227364
## 3 12.774194   14.321308
## 4 14.000000   13.378891
## 5  9.079365   10.261038
## 6  5.389831    5.156652
## 7  0.983871    2.653383
## 8  0.000000    1.230704
```

```
est.dispersion=fit$deviance/fit$df.residual
est.dispersion
```

```
## [1] 1.872039
```

```
summary(fit,dispersion=est.dispersion,correlation=TRUE)
```

```
##
## Call:
## glm(formula = cbind(y, n - y) ~ ldose, family = "binomial", data = investr::beetle)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -60.717      7.088  -8.566   <2e-16 ***
## ldose         34.270      3.984   8.601   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1.872039)
##
##     Null deviance: 284.202  on 7  degrees of freedom
## Residual deviance:  11.232  on 6  degrees of freedom
## AIC: 41.43
##
## Number of Fisher Scoring iterations: 4
##
## Correlation of Coefficients:
##       (Intercept)
## ldose -1.00
```

```
fitquasi=glm(cbind(y, n-y) ~ ldose, family = "quasibinomial", data = investr::beetle)
# preferred method of estimation is to use quasilikelihood
summary(fitquasi)$dispersion
```

```
## [1] 1.671141
```

Prospective vs. retrospective sampling

This section is optional - but it is very useful if you will work within biostatistics. (Examples motivated by Faraway (2006), Extending the linear model with R, Section 2.6.)

In a *prospective sampling* strategy we sample individuals from a population (covariates are then fixed) and wait for a predefined period to check if an event has happend (e.g. disease). This is also called a *cohort study*. An example might be that we select a sample of newborn babies (girls and boys) where the parents had decided on the method of feeding (bottle, breast, breast with some supplement), and then monitored the babies during their first year to see if they developed infant respiratory disease (the event we want to model).

For rare events the sample may then include few individuals with success (disease), which might lead to wide confidence intervals for parameters.

An alternative strategy is called *retrospective sampling*. Here we have access to medical registers and select a sample of n_1 babies where we know that they developed infant respiratory disease (the *cases*) and a sample of n_2 babies where we know that they did not develop infant respiratory disease (the *controls*). We require that the samples are selected independently of the values of the covariates (here: sex and food=method of feeding). This is called a *case–control study*.

If we aim to *model the relationship between the covariates and the response* (here: infant respiratory disease) we would naturally by default choose the cohort design, however, we will now show that the case–control design is also a valid design *if* we use the logit model to analyse the data.

To illustrate this we use a data set from a prospective study,

```
library(faraway)
data(babyfood)
head(babyfood)
```

```
##   disease nondisease  sex   food
## 1      77        381  Boy Bottle
## 2      19        128  Boy  Suppl
## 3      47        447  Boy Breast
## 4      48        336 Girl Bottle
## 5      16        111 Girl  Suppl
## 6      31        433 Girl Breast
```

```
xtabs(disease/(disease+nondisease)~sex+food,babyfood)
```

```
##       food
## sex        Bottle     Breast      Suppl
##   Boy  0.16812227 0.09514170 0.12925170
##   Girl 0.12500000 0.06681034 0.12598425
```

and for simplicity we focus on girls who were breast or bottle fed:

```
babyfood[c(4,6),]
```

```
##   disease nondisease  sex   food
## 4      48        336 Girl Bottle
## 6      31        433 Girl Breast
```

Notice - we now look at the observed data within the different covariate patterns. This would be the same as fitting a binomial glm with sex, food and the interaction thereof as covariates.

We read off: given that the infant is breast fed: 31/433= 0.07 is the estimated odds of respiratory disease, and the log-odds is -2.64. And, given that the infant is bottle fed: 48/336= 0.14 is the estimated odds of respiratory disease, and the log-odds is -1.95. The difference between these two log-odds represents the increased risk of infant respiratory disease caused by bottle feeding relative to breast feeding:

Δ= -1.95- -2.64= 0.69.

But, what if we instead had performed a case–control study, and then wanted to compute the log odds difference between feeding type given respiratory disease status.

We read off: given that the infant got the disease (case): 48/31= 1.55 is the estimated odds for bottle compared to breast, and the log-odds is 0.44. Given that the infant did not get the disease (control): 336/433= 0.78 is the estimated odds for bottle compared to breast, and the log-odds is -0.25. The difference between these two log-odds represents the “increased risk” of bottle feeding relative to breast feeding “caused by respiratory disease”.

Δ^*=0.44- -0.25=0.69.

Observe that Δ and Δ^* are equal, and shows (with a numerical example) why a retrospective design can be use to estimate change in log-odds. But, this argumentation is only valid for our logit model (where we estimate changes in odds), and not for the probit or complementary-log-log model.

We would like to use the retrospective case–control design instead of the prospective cohort design because it is more convenient to get a sample of cases on the same size as a sample of controls - which give us better properties of the parameter estimators. However, to use the case–control design it is important that covariates are reliably measured retrospective (rely on records).

In a mathematical comparison between logit models for pro- and retrospective designs we will find that the regression parameters $\beta_1, \dots, \beta_k$ will be the same in the two models, but the intercept terms will differ.

In the prospectictive study we model π_i, the probability of sucess (e.g. respiratory disease) for covariates $x_{i1}, x_{i2}, \dots, x_{ik}$ (e.g. sex and food). Using the logit model in the proseptive study, the regression model is

$$\ln(\frac{\pi_i}{1-\pi_i}) = \eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}$$

and we write

$$\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}.$$

In a retrospective study we consider ρ_i, the probability of sucess for covariates $x_{i1}, x_{i2}, \dots, x_{ik}$, given that individual i was sampled to be part of the case-control study.

Let S denote the event 'sampled', and let D denote event 'disease' (or sucess). Then, using Bayes' rule and the law of total probability, we have

$$\rho_i = P_i(D|S) = \frac{P_i(D \cap S)}{P_i(S)} = \frac{P_i(S|D)P_i(D)}{P_i(S|D)P_i(D) + P_i(S|D^c)P_i(D^c)}$$

where $P_i(D) = \pi_i$ and $P_i(D^c) = 1 - \pi_i$.

In the retrospective study, the probability of being sampled is equal for any individual with the disease; $\tau_1 = P(S|D) = P_i(S|D)$ and similarly for individuals who do not have the disease; $\tau_0 = P(S|D^c) = P_i(S|D^c)$. Then,

$$\rho_i = \frac{\tau_1 \pi_i}{\tau_1 \pi_i + \tau_0(1-\pi_i)} = \frac{\tau_1 \frac{\exp(\eta_i)}{1+\exp(\eta_i)}}{\tau_1 \frac{\exp(\eta_i)}{1+\exp(\eta_i)} + \tau_0 \frac{1}{1+\exp(\eta_i)}}$$

Dividing by τ_0 and multiplying by $1 + \exp(\eta_i)$ in the numerator and denominator gives

$$\rho_i = \frac{(\tau_1/\tau_0)\exp(\eta_i)}{(\tau_1/\tau_0)\exp(\eta_i) + 1} = \frac{\exp(\ln(\tau_1/\tau_0) + \eta_i)}{\exp(\ln(\tau_1/\tau_0) + \eta_i) + 1}$$

Using the logit model for ρ_i in the retrospective model, the regression model is

$$\ln(\frac{\rho_i}{1-\rho_i}) = \ln(\tau_1/\tau_0) + \eta_i = \ln(\tau_1/\tau_0) + \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}$$

We see that the only difference between the parameters of interest in the retrospective study and the parameters of interest in the prospective study is the intercept of the regression model. All other covariates are equal.

Interactive lecture - second week

We will use a data set on contraceptive use in Fiji (data from 1975). The data is to be analysed with "current use of contraceptive" as response and some (or all of) "age", "level of education", "desire for more children" as covariates

The data is available at https://grodri.github.io/datasets/cuse.dat with the following description:

- Contraceptive use: yes (`using`) or no (`notUsing`)
- `age`: categorical variable with 5 levels: "<25", "25-29", "30-39", "40-49"
- `education`: categorical variable with 2 levels giving highest level of education obtained: Lower and Upper
- `wantsMore`: Desires more children: yes or no

```r
ds=read.table("https://grodri.github.io/datasets/cuse.dat",header=TRUE)
names(ds)
```

```
## [1] "age"       "education" "wantsMore" "notUsing"  "using"
```

```r
summary(ds)
```

```
##      age             education          wantsMore            notUsing     
##  Length:16          Length:16          Length:16          Min.   :  8.00  
##  Class :character   Class :character   Class :character   1st Qu.: 31.00  
##  Mode  :character   Mode  :character   Mode  :character   Median : 56.50  
##                                                           Mean   : 68.75  
##                                                           3rd Qu.: 85.75  
##                                                           Max.   :212.00  
##      using      
##  Min.   : 4.00  
##  1st Qu.: 9.50  
##  Median :29.00  
##  Mean   :31.69  
##  3rd Qu.:49.00  
##  Max.   :80.00
```

```r
dim(ds)
```

```
## [1] 16  5
```

```r
head(ds)
```

```
##     age education wantsMore notUsing using
## 1   <25       low       yes       53     6
## 2   <25       low        no       10     4
## 3   <25      high       yes      212    52
## 4   <25      high        no       50    10
## 5 25-29       low       yes       60    14
## 6 25-29       low        no       19    10
```

We will study binary regression using the logit model, and we will work with grouped data.

Plan: Start with Problem 2, then move to 3 and 4, and if you have time you look at Problem 1. We will do a team Kahoot! in the end of the IL - on one device go to kahoot.it or use an app version.

Problem 1: The null model - no covariates included.

(This is the most theoretical of the problems and rather technical - but with some cool results on the null model.)

a) Fit the null model and call it `fit0`. Explain what you see.

```
fit0=glm(cbind(using,notUsing)~1,data=ds,family=binomial(link=logit))
summary(fit0)
```

```
##
## Call:
## glm(formula = cbind(using, notUsing) ~ 1, family = binomial(link = logit),
##     data = ds)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.77455    0.05368  -14.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.77  on 15  degrees of freedom
## Residual deviance: 165.77  on 15  degrees of freedom
## AIC: 239.28
##
## Number of Fisher Scoring iterations: 4
```

Observe: When we only have an intercept in the model, the model matrix will be an $n \times 1$-matrix with only ones. Then $\mathbf{x}_j^T\boldsymbol{\beta} = \beta_0$. The log-likelihood can be written as (let $j = 1, \dots, G$)

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{G}(y_j\mathbf{x}_j^T\boldsymbol{\beta} - n_j\log(1+\exp(\mathbf{x}_j^T\boldsymbol{\beta}))) = \sum_{j=1}^{G}(y_j\beta_0 - n_j\log(1+\exp(\beta_0))) \tag{7}$$

$$= \beta_0\sum_{j=1}^{G}y_j - \log(1+\exp(\beta_0))\sum_{j=1}^{G}n_j = \beta_0 N_1 - \log(1+\exp(\beta_0))N \tag{8}$$

where $N_1 = \sum_{j=1}^{G} y_j$ is the total number of successes and $N = \sum_{j=1}^{G} n_j$ is the total number of trials (over all covariates patterns, that is, here the number of individuals in the data set). Also $N_2 = N - N_1$ is the total number of failures.

We will use this loglikelihood in the next question.

b) What is the relationship between your estimated coefficient and the proportion in the data set using contraception (N_1/N)? (Hint 0: What would be your intuitive estimator for π (common to all groups). Hint 1: Find the derivative of the log-likelihood with respect to β_0, and use this to find the MLE for β_0. Hint 2: maybe easier to see what $\hat{\pi}$ is, where $\hat{\pi} = \frac{\exp\hat{\beta}_0}{1+\exp\hat{\beta}_0}$ (so `plogis`), and then $\hat{\boldsymbol{\beta}}_0 = \text{logit}(\hat{\pi})$ (so `qlogis`). Hint 3: You can verify by using the estimate from `glm`.)

```
N = sum(ds$using + ds$notUsing)
N
```

```
## [1] 1607
```

```
N1 = sum(ds$using)
N1
```

```
## [1] 507
```

```
N2 = N - N1
N2
```

```
## [1] 1100
```

```
qlogis(N1/N)
```

```
## [1] -0.7745545
```

```
fit0$coefficients
```

```
## (Intercept)
##  -0.7745545
```

c) We know that the (asymptotic) estimated covariance matrix of the ML estimate is the inverse of the expected Fisher information matrix, here the matrix is only a scalar and the covariance matrix is just the variance. Find the mathematical expression for the estimated variance of our estimated intercept. (Hint 1: We have $F(\boldsymbol{\beta}) = \sum_{j=1}^{G} x_j x_j^T n_j \pi_j (1-\pi_j)$, and then insert $x_j = 1$ and $\pi_j(1-\pi_j) = \pi(1-\pi)$, and hopefully you found above that $\hat{\pi} = N_1/N$ in our model with only intercept. Hint 2: $\frac{N}{N_1 \cdot N_2} = \frac{1}{N_1} + \frac{1}{N_2}$ to make things prettier.)

d) What is the estimated (numerical value) standard deviation of the parameter estimate? Hint: `vcov(fit0)`. Did your calculation above gives the same result?

```
vcov(fit0)
```

```
##             (Intercept)
## (Intercept) 0.002881477
```

e) What is the asymptotic distribution of the estimated regression coefficient? Use this to write down the formula for the 95% confidence interval for the intercept, and calculate the interval in R. Compare numerically to `confint.default` and `confint`.

```
ci=confint.default(fit0)
ci
```

```
##                  2.5 %     97.5 %
## (Intercept) -0.8797641 -0.6693448
```

```
confint(fit0)
```

```
##      2.5 %     97.5 %
## -0.8804716 -0.6700014
```

f) Translate the 95% CI to probability scale (Hint: `plogis` is the inverse logit.)

```
plogis(ci)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2932267 0.3386436
```

```
fit0$family$linkinv(ci)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2932267 0.3386436
```

Problem 2: We then study the effect of the covariate "wantsMore"

(a little less maths)

a) Fit a regression with `wantsMore` as covariate, and call this `fit1`. Explain what the estimated coefficient (β_1) means. Hint: Interpretation using odds – if ‘wantsMore´ goes from 0=no to 1=yes, then…

```r
fit1=glm(cbind(using,notUsing)~wantsMore,data=ds,family=binomial)
summary(fit1)
```

```
##
## Call:
## glm(formula = cbind(using, notUsing) ~ wantsMore, family = binomial,
##     data = ds)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.18636    0.07971  -2.338   0.0194 *
## wantsMoreyes -1.04863    0.11067  -9.475   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  74.098  on 14  degrees of freedom
## AIC: 149.61
##
## Number of Fisher Scoring iterations: 4
```

```r
exp(fit1$coefficients)
```

```
##  (Intercept) wantsMoreyes
##    0.8299712    0.3504178
```

b) Is this covariate significant? Write down the null and alternative hypothesis to test. Then test using the Wald test - write down the formula yourself. Use `vcov(fit1)` to access the inverse of the Fisher information matrix. What is the degrees of freedom for this test? Compare to the print-out from `summary`.

```r
vcov(fit1)
```

```
##               (Intercept) wantsMoreyes
## (Intercept)   0.006354067 -0.006354067
## wantsMoreyes -0.006354067  0.012248298
```

c) Alternatively, the likelihood ratio test can be used. Write down the formula for the likelihood ratio test statistic. Use this in combination with the `fit0` and `fit1` objects to calculate the likelihood ratio statistic. Then calculate the *p*-value.

```r
loglik <- function(par, args) {
    y <- args$y
    x <- args$x
    n <- args$n
    res <- sum(y * x %*% par - n * log(1 + exp(x %*% par)))
    return(res)
}

args0 <- list(y = ds$using, n = ds$using + ds$notUsing, x = cbind(rep(1, nrow(ds))))
args1 <- list(y = ds$using, n = ds$using + ds$notUsing, x = cbind(rep(1, nrow(ds)),
    as.numeric(ds$wantsMore) - 1))

betas0 <- fit0$coefficients
betas1 <- fit1$coefficients
```

```
ll0 <- loglik(betas0, args0)
ll0
```

```
## [1] -1001.847
```

```
ll1 <- loglik(betas1, args1)
ll1
```

```
## [1] NA
```

d) The likelihood ratio test statistic can alternatively be calculated using the residual deviance in `fit1` and `fit0`, and is given as `fit0$deviance-fit1$deviance`. Do you see why?

```
fit0$deviance-fit1$deviance
```

```
## [1] 91.6744
```

e) Are the two test statistics (Wald and LRT) equal? Do the two tests give the same conclusions?

Problem 3: Now two covariates - deviance and model comparison

(no maths - only definitions and print-out)

Now we study the response together with `age` and `wantsMore`. We will consider the following 5 models. See R-code and print-out below. **a)** Explain what each of these models include.

b) What is the definition of the deviance and df. What can we use the deviance for? Optional: derive the formula for the deviance for the logit model (the derivation is not given on the module pages.)

c) Perform a likelihood ratio test - based on the deviance results given in the data frame in the R chunk-to compare the additive and interact models. First write down the null and alternative hypothesis you are testing. Which of the two models would you prefer?

d) What if you use the AIC for model selection, which model (out of the 5) will you then use?

```
ds$Y <- cbind(ds$using, ds$notUsing)
models <- list(
  null     = glm(Y ~ 1, family=binomial, data=ds),
  age      = glm(Y ~ age, family=binomial, data=ds),
  desire   = glm(Y ~ wantsMore, family=binomial, data=ds),
  additive = glm(Y ~ age + wantsMore, family=binomial, data=ds),
  interact = glm(Y ~ age*wantsMore, family=binomial, data=ds)
)
models
```

```
## $null
##
## Call:  glm(formula = Y ~ 1, family = binomial, data = ds)
##
## Coefficients:
## (Intercept)
##     -0.7746
##
## Degrees of Freedom: 15 Total (i.e. Null);  15 Residual
## Null Deviance:       165.8
## Residual Deviance: 165.8     AIC: 239.3
##
## $age
##
## Call:  glm(formula = Y ~ age, family = binomial, data = ds)
```

```
##
## Coefficients:
## (Intercept)     age25-29     age30-39     age40-49
##     -1.5072       0.4607       1.0483       1.4246
##
## Degrees of Freedom: 15 Total (i.e. Null);  12 Residual
## Null Deviance:       165.8
## Residual Deviance: 86.58     AIC: 166.1
##
## $desire
##
## Call:  glm(formula = Y ~ wantsMore, family = binomial, data = ds)
##
## Coefficients:
##  (Intercept)  wantsMoreyes
##      -0.1864       -1.0486
##
## Degrees of Freedom: 15 Total (i.e. Null);  14 Residual
## Null Deviance:       165.8
## Residual Deviance: 74.1  AIC: 149.6
##
## $additive
##
## Call:  glm(formula = Y ~ age + wantsMore, family = binomial, data = ds)
##
## Coefficients:
##  (Intercept)      age25-29      age30-39      age40-49  wantsMoreyes
##      -0.8698        0.3678        0.8078        1.0226       -0.8241
##
## Degrees of Freedom: 15 Total (i.e. Null);  11 Residual
## Null Deviance:       165.8
## Residual Deviance: 36.89     AIC: 118.4
##
## $interact
##
## Call:  glm(formula = Y ~ age * wantsMore, family = binomial, data = ds)
##
## Coefficients:
##           (Intercept)               age25-29               age30-39
##               -1.4553                 0.6354                 1.5411
##              age40-49           wantsMoreyes  age25-29:wantsMoreyes
##                1.7643                -0.0640                -0.2672
## age30-39:wantsMoreyes  age40-49:wantsMoreyes
##               -1.0905                -1.3671
##
## Degrees of Freedom: 15 Total (i.e. Null);  8 Residual
## Null Deviance:       165.8
## Residual Deviance: 20.1  AIC: 107.6
```

```
lapply(models,summary)
```

```
## $null
##
## Call:
## glm(formula = Y ~ 1, family = binomial, data = ds)
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.77455    0.05368  -14.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.77  on 15  degrees of freedom
## Residual deviance: 165.77  on 15  degrees of freedom
## AIC: 239.28
##
## Number of Fisher Scoring iterations: 4
##
##
## $age
##
## Call:
## glm(formula = Y ~ age, family = binomial, data = ds)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.5072     0.1303 -11.571  < 2e-16 ***
## age25-29      0.4607     0.1727   2.667  0.00765 **
## age30-39      1.0483     0.1544   6.788 1.14e-11 ***
## age40-49      1.4246     0.1940   7.345 2.06e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  86.581  on 12  degrees of freedom
## AIC: 166.09
##
## Number of Fisher Scoring iterations: 4
##
##
## $desire
##
## Call:
## glm(formula = Y ~ wantsMore, family = binomial, data = ds)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.18636    0.07971  -2.338   0.0194 *
## wantsMoreyes -1.04863    0.11067  -9.475   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  74.098  on 14  degrees of freedom
## AIC: 149.61
##
## Number of Fisher Scoring iterations: 4
##
##
## $additive
##
## Call:
## glm(formula = Y ~ age + wantsMore, family = binomial, data = ds)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.8698     0.1571  -5.536 3.10e-08 ***
## age25-29       0.3678     0.1754   2.097    0.036 *
## age30-39       0.8078     0.1598   5.056 4.27e-07 ***
## age40-49       1.0226     0.2039   5.014 5.32e-07 ***
## wantsMoreyes  -0.8241     0.1171  -7.037 1.97e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  36.888  on 11  degrees of freedom
## AIC: 118.4
##
## Number of Fisher Scoring iterations: 4
##
##
## $interact
##
## Call:
## glm(formula = Y ~ age * wantsMore, family = binomial, data = ds)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.4553     0.2968  -4.903 9.43e-07 ***
## age25-29                0.6354     0.3564   1.783  0.07463 .
## age30-39                1.5412     0.3183   4.842 1.29e-06 ***
## age40-49                1.7643     0.3435   5.136 2.80e-07 ***
## wantsMoreyes           -0.0640     0.3303  -0.194  0.84637
## age25-29:wantsMoreyes  -0.2672     0.4091  -0.653  0.51366
## age30-39:wantsMoreyes  -1.0905     0.3733  -2.921  0.00349 **
## age40-49:wantsMoreyes  -1.3672     0.4834  -2.828  0.00468 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  20.099  on  8  degrees of freedom
## AIC: 107.61
##
```

```
## Number of Fisher Scoring iterations: 4
```

```
data.frame(deviance=round(unlist(lapply(models,deviance)),2),df=unlist(lapply(models,df.residual)),aic=
```

```
##          deviance df aic
## null       165.77 15 239
## age         86.58 12 166
## desire      74.10 14 150
## additive    36.89 11 118
## interact    20.10  8 108
```

Note: The function `lapply` (list apply) will apply a function to each element of a list.

###Problem 4: Plotting (yes, mainly R and trying to understand)

Finally, we want to use the additive model with all three covariates: `Y~age+education+wantsMore` to study different plots.

```
fit3add=glm(cbind(using,notUsing)~age+education+wantsMore,data=ds,family=binomial)
summary(fit3add)
```

```
##
## Call:
## glm(formula = cbind(using, notUsing) ~ age + education + wantsMore,
##     family = binomial, data = ds)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.8082     0.1590  -5.083 3.71e-07 ***
## age25-29       0.3894     0.1759   2.214  0.02681 *
## age30-39       0.9086     0.1646   5.519 3.40e-08 ***
## age40-49       1.1892     0.2144   5.546 2.92e-08 ***
## educationlow  -0.3250     0.1240  -2.620  0.00879 **
## wantsMoreyes  -0.8330     0.1175  -7.091 1.33e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.772  on 15  degrees of freedom
## Residual deviance:  29.917  on 10  degrees of freedom
## AIC: 113.43
##
## Number of Fisher Scoring iterations: 4
```

```
exp(fit3add$coefficients)
```

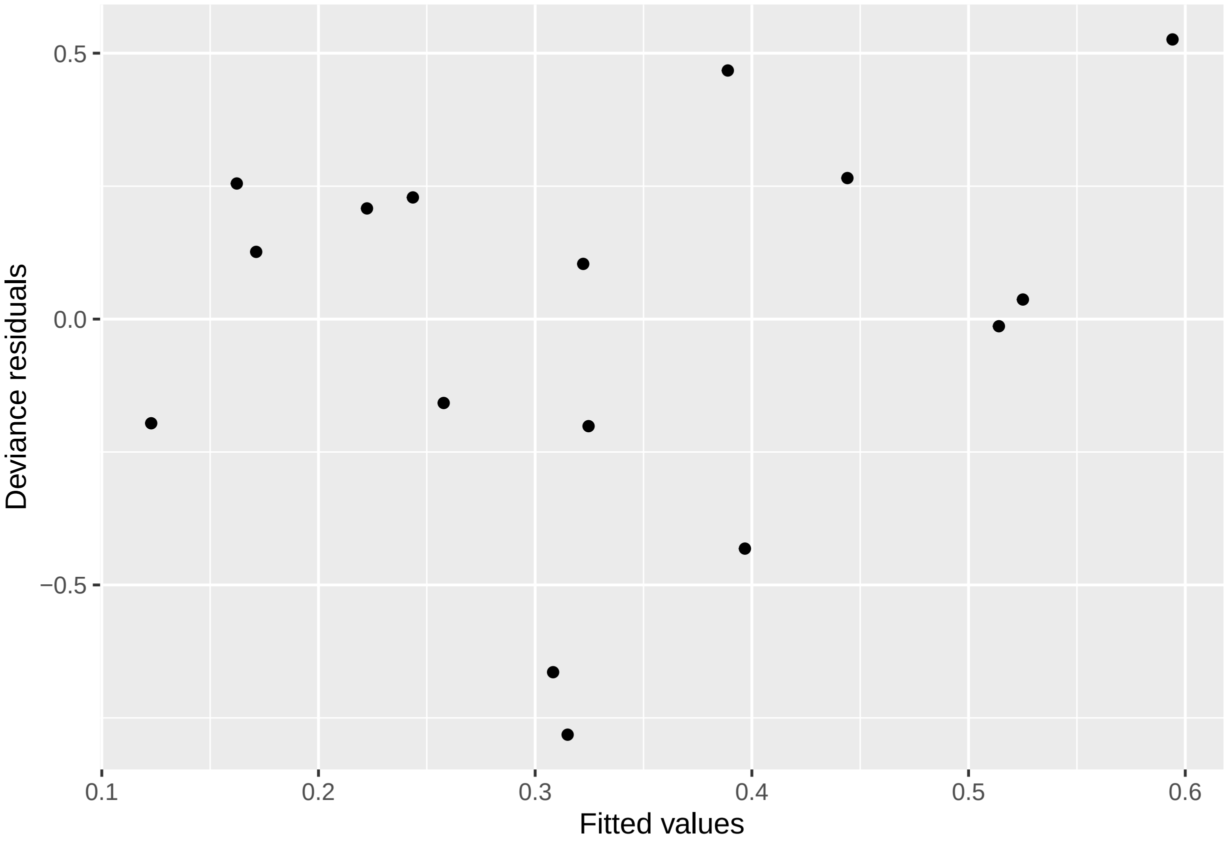
```
##  (Intercept)     age25-29     age30-39     age40-49 educationlow wantsMoreyes
##    0.4456506    1.4760678    2.4808804    3.2845805    0.7225312    0.4347628
```

a) Comment on the output from `summary`. Use the deviance to test if this is a good model. Which model do you then compare to (what is the saturated model here)?.

b) Look at the plot of fitted values vs. deviance residuals. Do you see a trend?

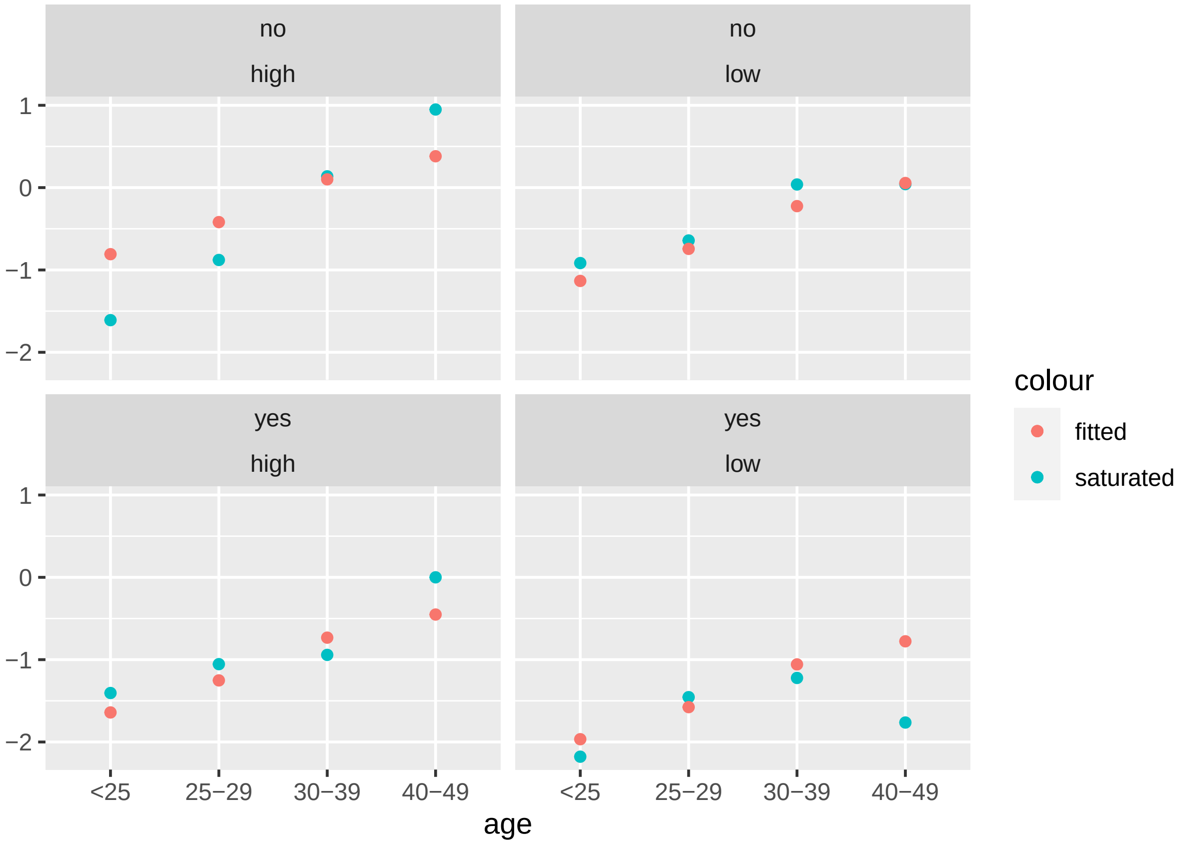
```
library(ggplot2)
plotdf <- data.frame(dres = fit3add$residuals, fitted = fit3add$fitted.values,
    age = ds$age)
```

```r
ggplot(plotdf, aes(x = fitted, y = dres)) + geom_point() + labs(x = "Fitted values",
    y = "Deviance residuals")
```



c) Here is a code to produce a plot of fitted values for the 16 covariate patterns together with a saturated model (called frac) - on the logit scale. How can you see from the dots for fitted values that we are assuming an additive model on the logit scale (that is, the pattern for each panel for logit fitted vs age is the same for each panel if we have an additive model on the logit scale - you see that from the fitted points)? Do you also see that for the observed values? Implication? Any other observations?

```r
library(ggplot2)
frac=ds$using/(ds$using+ds$notUsing)
logitdf=data.frame(lfitted=qlogis(fit3add$fitted),lfrac=qlogis(frac),age=ds$age,wantsMore=ds$wantsMore,
ggplot(logitdf,aes(x=age))+geom_point(aes(y=lfrac,colour="saturated"))+geom_point(aes(y=lfitted,colour=
```



To see what Rodríguez have found to be the best model, look at the bottom of http://data.princeton.edu/wws509/R/c3s6.html.

Note: This exercise is based on the excellent notes of Germán Rodríguez at Princeton, see in particular the R logs at http://data.princeton.edu/wws509/R/c3s1.html.

Exam questions

For this module the following are exam questions to work with

- 2012 – Problem 1
- 2011 – Problem 1

In addition these essay-type exam questions are closely related to this module.

December 2014

There are two main asymptotic results that are behind essentially everything we have done with respect to inference for generalized linear models. These are

1. asymptotic normality of maximum likelihood estimators (MLE), and
2. asymptotic result for likelihood ratio tests (LRT).

State, describe and discuss the main assumptions behind these two asymptotic results, how these results are used in practice to do inference for parameters, testing various hypotheses and comparing nested models, for logistic regression.

December 2016

We will consider (binomial or binary) logistic regression, where we have independent observations

$$Y_i \sim \text{Binomial}(n_i, \pi_i) \; i = 1, \ldots, n$$

so that

$$P(Y_i) = \binom{n_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i}, \; y_i = 0, 1, \ldots, n_i$$

The linear predictor is

$$\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$$

and

$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

or

$$\text{logit}(\pi_i) = \eta_i.$$

Here, $\mathbf{x}_i$ is a vector of the p covariates for the ith observation y_i with size (number of trials) n_i, and $\boldsymbol{\beta}$ is the vector of p unknown regression coefficients.

Write an introduction to logistic regression and its practical usage, for a student with a good background in statistics, but no knowledge about Generalized Linear Models (GLM). Topics you may want to consider, are

- When to use it? Underlying assumptions.
- Parameter estimation, limiting results for the MLE, Fisher information and observed Fisher information, confidence intervals and hypothesis testing.
- Output analysis, residual plots (when it is possible) and interpretation of the $\boldsymbol{\beta}$-coefficients
- Deviance and its usage.

R packages

```r
install.packages(c("tidyverse",
                   "investr",
                   "knitr",
                   "kableExtra",
                   "faraway",
                   "viridis",
                   "statmod"))
```

References for further reading

- A. Agresti (2015): “Foundations of Linear and Generalized Linear Models.” Wiley.
- A. J. Dobson and A. G. Barnett (2008): “An Introduction to Generalized Linear Models”, Third edition.
- J. Faraway (2015): “Extending the Linear Model with R”, Second Edition. http://www.maths.bath.ac.uk/~jjf23/ELM/
- P. McCullagh and J. A. Nelder (1989): “Generalized Linear Models”. Second edition.