Categorical regression

Anders Tolver (slides by Bo Markussen)
tolver@math.ku.dk

Department of Mathematical Sciences

December 2, 2020

Why do statistics?

Brief summary of Day 1 and 2

1. Is there an effect?
   - Answered by p-values.
   - Power vs. Risk of False Positives (Sterne & Smith, 2001).
   - Discussed on Day 1 and 2.
2. Where is the effect?
   - Answered by p-values from post hoc analyses.
   - Will first be discussed later in the course.
3. What is the effect?
   - Answered by confidence and prediction intervals.
   - Power vs. Risk of Type S error + Size of Type M error (Gelman & Carlin, 2014).
   - Briefly discussed on Day 1 and 2.
4. Can the conclusions be trusted?
   - Answered by model validation.
   - Briefly discussed on Day 1 and 2.

Summary: Chi-squared vs. McNemar test

Exercise 2.4: The 2×2 table in the exercise contains row and column marginals. Thus, the actual cross-tabulation of the 85 sibling pairs is this:

		Control: Tonsillectomy	Control: No tonsillectomy	Total
Hodgkin:	Tonsillectomy	26	15	41
	No tonsillectomy	7	37	44
	Total	33	52	85

- **Chi-squared:** $p = 0.00002$. Strong evidence of *correlation* between siblings, which might be due to genetic heritability.
- **McNemar:** $p = 0.1326$. Still no evidence of association between Hodgkin's disease and risk of tonsillectomy.

Thus, both tests make sense. But please note the different interpretations of the (possibly significant) results.

Solution to Exercise 2.7

Categorization of the continuous height measurements results in the following table:

Count (row pct)	Sons		
	Small	Tall	Total
Parents: small	247 (62%)	152 (38%)	399 (100%)
tall	189 (34%)	364 (66%)	553 (100%)
Total	436	516	952

Chi-square test for association: $\chi^2 = 70.6704$, df=1, $p < 2.2 \cdot 10^{-16}$:

```
chisq.test(matrix(c(247,189,152,364),2,2))
```

Thus, the association is highly significant. Inspection of the row percentages shows that tall parents tend to get tall sons.

- In this situation McNemar's test is non-significant (p=0.05123). But what does this mean?

Properties of good statistical models

- **Valid**
  - "All models are wrong", but a statistical model must be valid. This means that the probabilistic properties implied by the model are meet by the data within statistical uncertainty.

- **Interpretable**
  - Often different valid models can be formulated for a given dataset. The interpretation of these models and their parameters may, however, be different. It is preferable to have an interpretation that matches the scientific question under investigation.

- **Powerful**
  - Some models and tests are better at detecting deviations from the null hypothesis than others. Loosely said, the more assumptions you put into a model the more powerful it is (and the more often it may be invalid).

What is regression analysis?

Here the "popular" answer

Simple linear regression

Relates a response variable to an explanatory variable via a straight line.

Multiple linear regression

Relates a response variable to several explanatory variables via a "web" of straight lines.

Categorical response variable: Examples of the main types

- **Binary** ($\sim$ Bernoulli distribution, i.e. binomial with n=1):
  - No, Yes.
- **Binomial** ($\sim$ binomial distribution):
  - Number of weeks with weight loss out of 8 weeks on some diet.
- **Nominal** ($\sim$ multinomial distribution):
  - Red, Green, Blue, Yellow, Purple.
- **Ordinal** ($\sim$ multinomial distribution):
  - No symptoms, Mild symptoms, Severe symptoms, Dead by disease.
- **Counts** ($\sim$ Poisson distribution):
  - 0, 1, 2, 3, ...

Overview: Categorical regression analysis

- Models and theory:

Response	Model	See slides
Binomial	Probit analysis	9–20
Binomial	Logistic regression	25–27, 32–36
Nominal	Multinomial logistic regression	39
Ordinal	Proportional odds model	38–41
Counts	Poisson regression	42–46

- R analysis:
  - Binary, binomial, counts responses: `glm()`
  - Nominal, ordinal responses: I recommend `ordinal::clm()`
  - Model validation: `gof::cumres()`. Unfortunately, the `gof` package is only available on github, and may installed using
    - ★ `devtools::install_github("kkholst/gof")`

- The main example in this lecture is binomial regression.
  - Please pay attention to the interpretation of the different models.

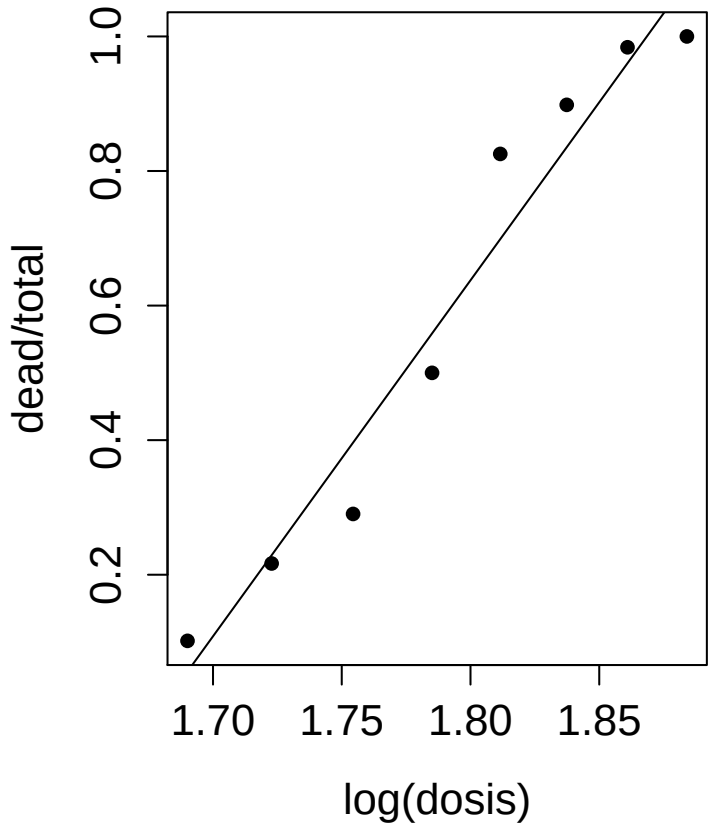
Data example 1: Mortality of beetles

Dosis-response experiment

481 beetles were exposed to 8 different doses of carbon-disulfide (CS_2) for 5 hours. Mortality in each dose group was registered:

dosis	alive	dead	total	$\hat{p}_{\text{dead}}$
5.42	53	6	59	0.10
5.60	47	13	60	0.21
5.78	44	18	62	0.29
5.96	28	28	56	0.50
6.12	11	52	63	0.82
6.28	6	53	59	0.89
6.43	1	61	62	0.98
6.58	0	60	60	1.00
total	190	291	481	0.60

Variables used in the R analysis:
dead, **alive**, **dosis**



Probit analysis and it's interpretation

Let $\Phi(x) = P(Z \leq x)$ be the cumulative distribution function of $\mathcal{N}(0, 1)$

- Suppose the i'th beetle has a tolerance value T_i for $\log(CS_2)$, i.e. the beetle dies if log-dosis is above the tolerance and survives otherwise.
- Suppose the distribution of tolerance values in the population of beetles is normal with mean μ and standard deviation σ.
- Suppose the i'th beetle is exposed to log-dosis of CS_2 of size x_i.

Then the probability that the i'th beetle dies equals

$$p_i = P(T_i < x_i) = \Phi\left(\frac{x_i - \mu}{\sigma}\right)$$

This implies a straight line with intercept $\alpha = -\frac{\mu}{\sigma}$ and slope $\beta = \frac{1}{\sigma}$:

$$\Phi^{-1}(p_i) = -\frac{\mu}{\sigma} + \frac{1}{\sigma} \cdot x_i = \alpha + \beta \cdot x_i$$

Overview of R code for beetle example

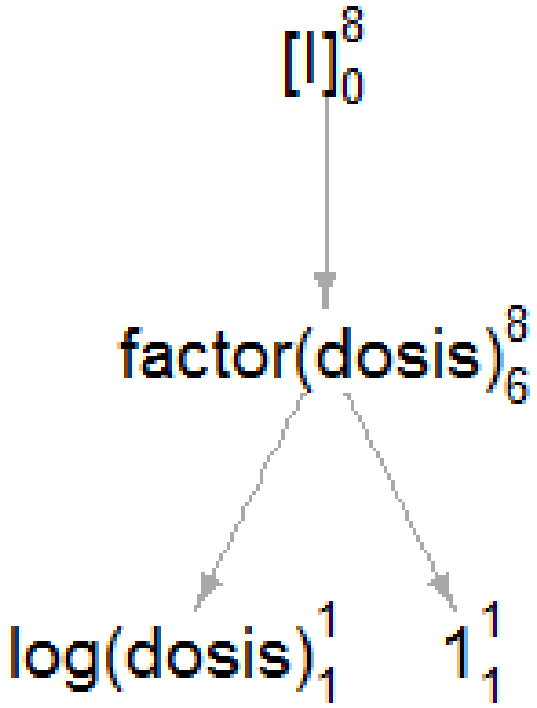
Slide 13	Fitting the model to the data.
Slide 15	Cumulative residuals and associated Goodness-of-Fit tests (Can the conclusions be trusted?)
Slide 12, 16	Lack-of-Fit test, which only is available in some situations (Can the conclusions be trusted?)
Slide 17	Hypothesis tests (Is there an effect?)
Slide 18	Estimates and confidence intervals (What is the effect?)
Slide 19, 20	Backtransformation in order to get estimates for the parameters in the tolerance distribution (What is the effect?)

Mortality of beetles

Table-of-Variables & Overview of design

Variable	Type	Range	Usage
dead	Binomial	6, 13, . . . , 61	response
alive	Binomial	0, 1, . . . , 53	response
dosis	Numerical	[5.42; 6.58]	fixed effect

- `dosis` will be used on log-scale as this gives a better fit to the data.
- Since the numerical variable `dosis` only takes 8 different values we may perform a Lack-of-Fit test. This is illustrated in the Design Diagram shown to the right.



Mortality of beetles

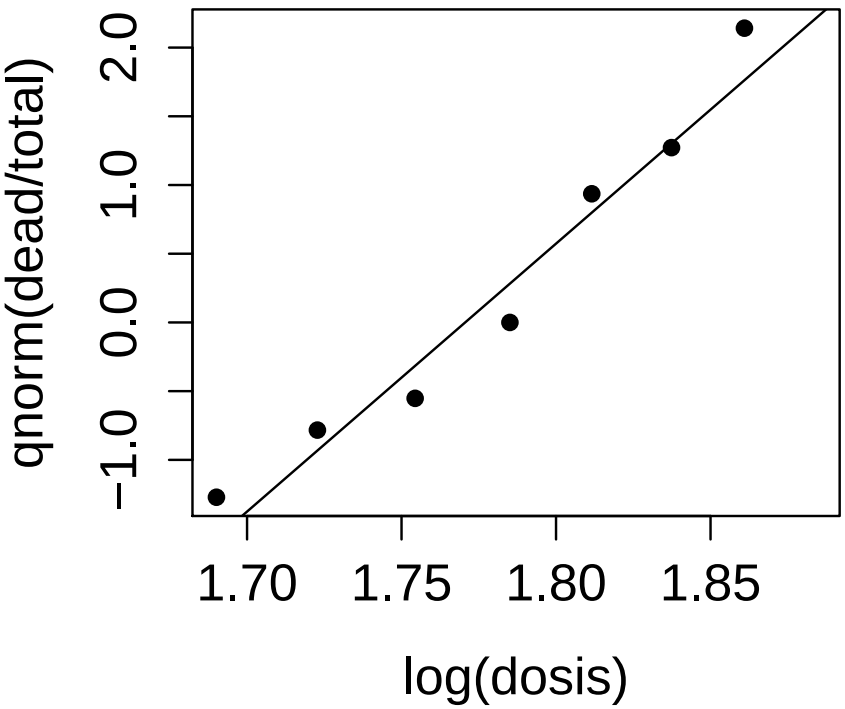
Fitting the probit model in R

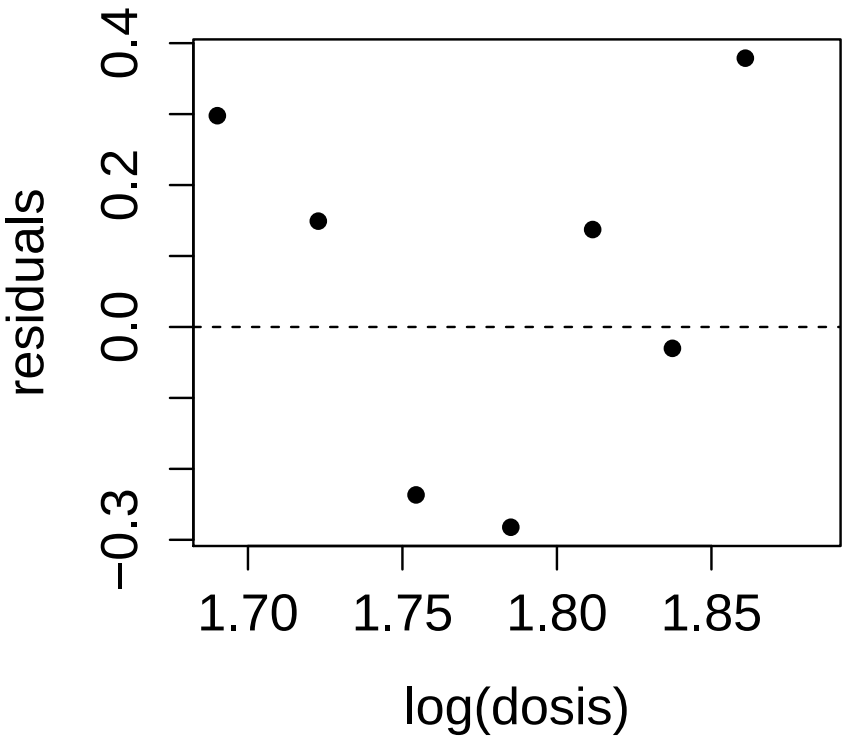
```
# Read data from text file
beetle <- read.delim("beetle.txt")

# Make probit regression
m1 <- glm(cbind(dead,alive)~log(dosis),data=beetle,
          family=binomial(link="probit"))
```

- The response consists of number of successes (dead beetles) and failures (alive beetles). These are combined column-wise, that is as variables, using `cbind()`.
- Note that we have used a logarithmic transformation of the explanatory variable.

Does the probit model fit the beetle data?

Note that qnorm(x) is the R code for $\Phi^{-1}(x)$

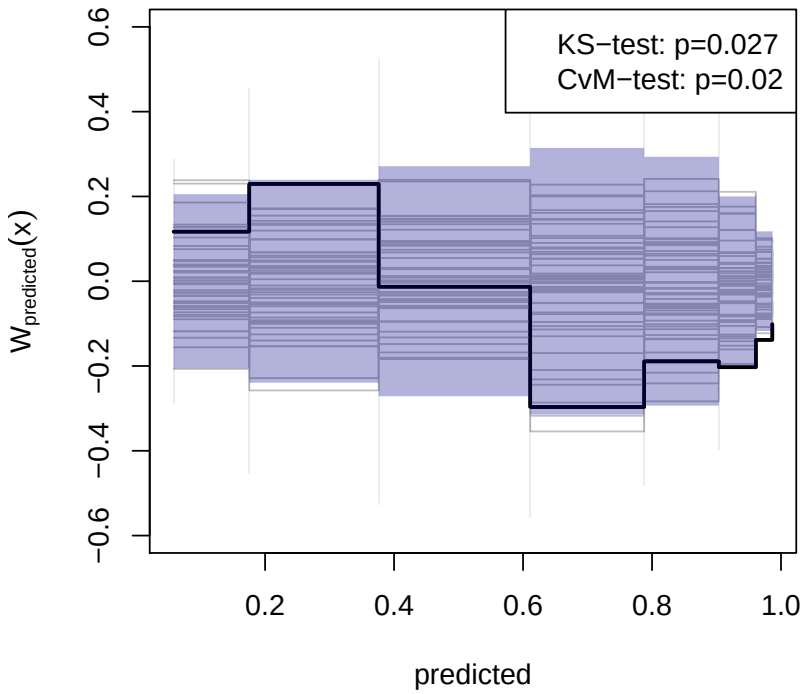


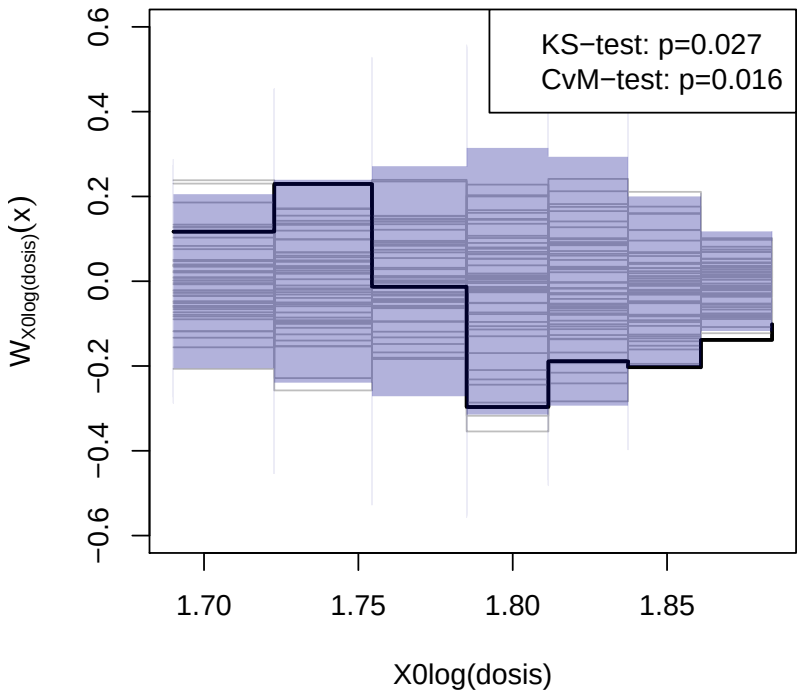


- Here we define the residuals as the deviation of the raw estimates (the points) from the model prediction (the line).
- A valid model should have random residuals, ie. without structure.

Idea: Instead investigate the cumulated residuals

And use associated Goodness-of-Fit tests (two given by the gof-package)





- One plot for the model + One plot for each for the continuous explanatory variables (here log(dosis) to the right). R code:

```
library(gof)
plot(cumres(m1))
```

- Goodness-of-Fit tests are based on simulations.

More model validation: Lack-of-Fit test

May be done since only a "few" different dosis were used

```
# Make a model where dosis is used as a categorical factor
m0 <- glm(cbind(dead,alive)~factor(dosis),data=beetle,
          family=binomial(link="probit"))

# Lack-of-Fit test: Test m1 as a hypothesis against m0
anova(m1,m0,test="Chisq")
```

More model validation: Lack-of-Fit test

May be done since only a "few" different dosis were used

```
# Make a model where dosis is used as a categorical factor
m0 <- glm(cbind(dead,alive)~factor(dosis),data=beetle,
          family=binomial(link="probit"))

# Lack-of-Fit test: Test m1 as a hypothesis against m0
anova(m1,m0,test="Chisq")
```

```
Analysis of Deviance Table

Model 1: cbind(dead, alive) ~ log(dosis)
Model 2: cbind(dead, alive) ~ factor(dosis)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6     10.509
2         0      0.000  6   10.509   0.1048
```

- What is the conclusion?

Hypothesis tests: Is there an effect?

Despite model is invalid by GoF-tests (see slide 15), we continue the analysis.
Your comments on this?

- Hypothesis tests may be done using the `anova()` function as demonstrated in the R guide.
- However, I recommend the `drop1()` function. The R code is easy:

```
drop1(m1,test="Chisq")
```

Hypothesis tests: Is there an effect?

Despite model is invalid by GoF-tests (see slide 15), we continue the analysis.
Your comments on this?

- Hypothesis tests may be done using the `anova()` function as demonstrated in the R guide.
- However, I recommend the `drop1()` function. The R code is easy:

```
drop1(m1,test="Chisq")
```

```
Single term deletions

Model:
cbind(dead, alive) ~ log(dosis)
           Df Deviance     AIC    LRT  Pr(>Chi)
<none>          10.509  40.707
log(dosis)  1  284.202 312.400 273.69 < 2.2e-16 ***
```

Parameter estimates and confidence intervals

- The parameter estimates can be extracted in many ways, e.g. `m1`, `summary(m1)`, `coef(m1)`.
- Confidence intervals may be found by `confint(m1)`.
- If preferred the output may be combined like this:

```
cbind(estimate=coef(m1),confint(m1))
```

Parameter estimates and confidence intervals

- The parameter estimates can be extracted in many ways, e.g. `m1, summary(m1), coef(m1)`.
- Confidence intervals may be found by `confint(m1)`.
- If preferred the output may be combined like this:

```
cbind(estimate=coef(m1),confint(m1))
```

```
Waiting for profiling to be done...
             estimate     2.5 %    97.5 %
(Intercept) -34.52600 -39.79798 -29.59097
log(dosis)   19.49966  16.72642  22.46531
```

Interpretation via tolerance distribution

Probit analysis: $p = \Phi\big(\alpha + \beta \cdot \log(\text{dosis})\big)$

The relation between the parameters in the linear model and the parameters in the tolerance distribution is as follows:

Parameter	Interpretation
$\mu = -\alpha/\beta$	Lethal Dosis 50% = mean in tolerance distribution
$\sigma = 1/\beta$	Scale = standard deviation in tolerance distribution

- Confidence interval for σ may be found by $1/x$-transforming the ditto for $\hat{\beta} = 19.4997$. Interpretation of $\beta < 0$ via, cf. slide 10,

$$P(T_i \geq x_i) = 1 - P(T_i < x_i) = 1 - \Phi\big(\alpha + \beta \cdot x_i\big)$$
$$= \Phi\big(-\alpha - \beta \cdot x_i\big)$$

- To find confidence interval for μ is more tricky since this is given as a non-linear combination of the parameters in the probit regression.
  - However, this may the done using the `deltaMethod()` function from the car-package.
  - Remember to install and load the package.

Backtransformation and confidence intervals

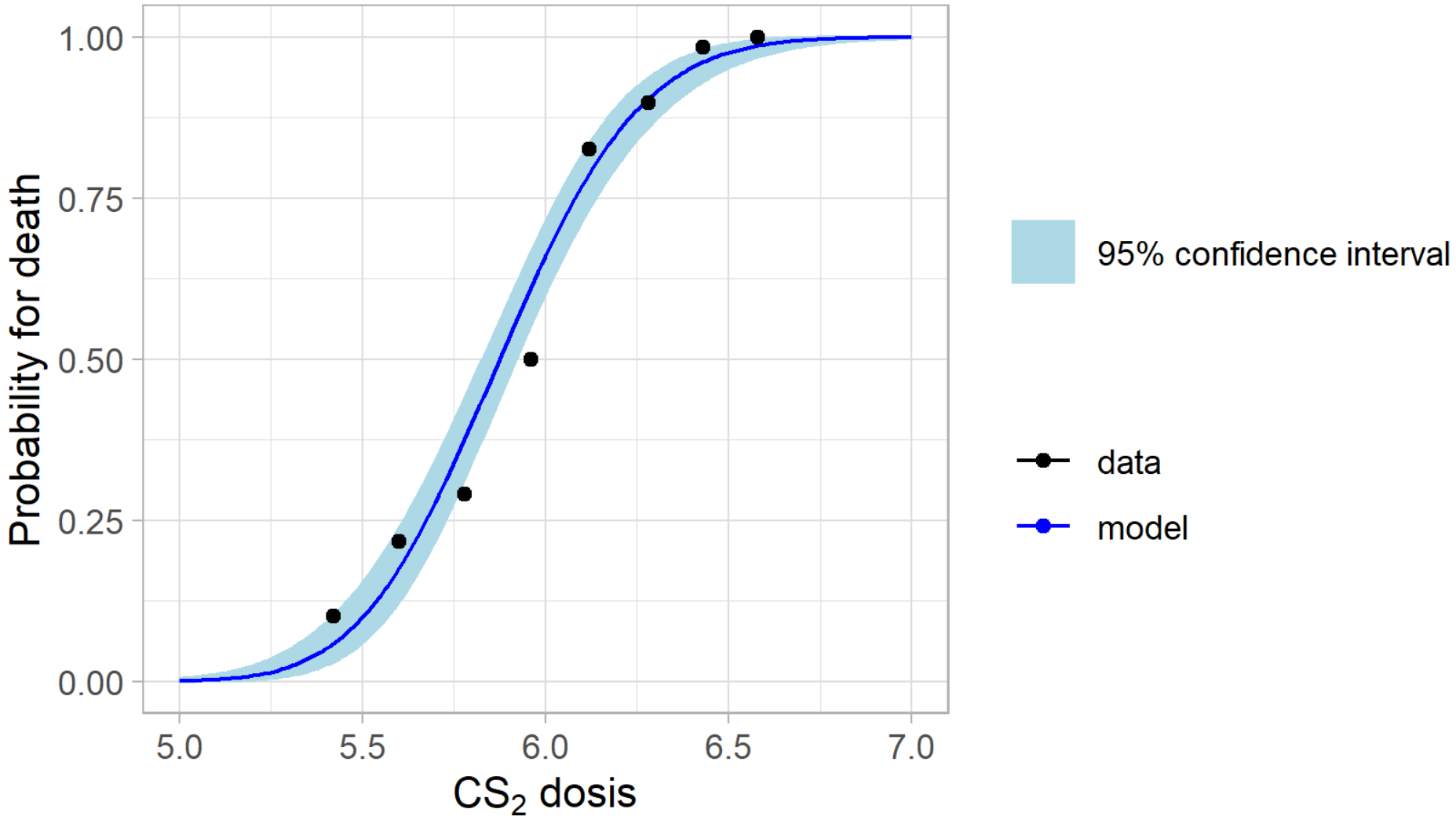
```
# Scale parameter in the tolerance distribution
1/cbind(estimate=coef(m1),confint(m1))[2,]

# Mean parameter in the tolerance distribution
deltaMethod(m1,"-alpha/beta",
            parameterNames=c("alpha","beta"))
```

```
Waiting for profiling to be done...
  estimate      2.5 %     97.5 %
0.05128295 0.05978567 0.04451306

              Estimate         SE     2.5 % 97.5 %
-alpha/beta 1.7705949 0.0038475 1.7630540 1.7781
```

Summary of beetle example

- A probit analysis of the death probability against the logarithm of the CS_2 dosis was performed.
- Model validity was investigated by cumulative residuals and associated Goodness-of-Fit tests, as well as a Lack-of-Fit test.
  - In practice I for this example probably wouldn't do the Lack-of-Fit test.
  - In this example the model was actually invalidated by the cumulative residuals (CvM gof-test gave p=0.02). So in principle, we shouldn't proceed with the analysis done on slides 17 – 20.
- Effect of CS_2 was highly significant.
- Estimates and confidence intervals were found for the parameters in the tolerance distribution, which provides the canonical interpretation of a probit analysis.

Graphical display of fitted model



Questions?

- And then a break.
- After the break we discuss logistic regression as an alternative to probit analysis.

Data example 2: Risk of company default

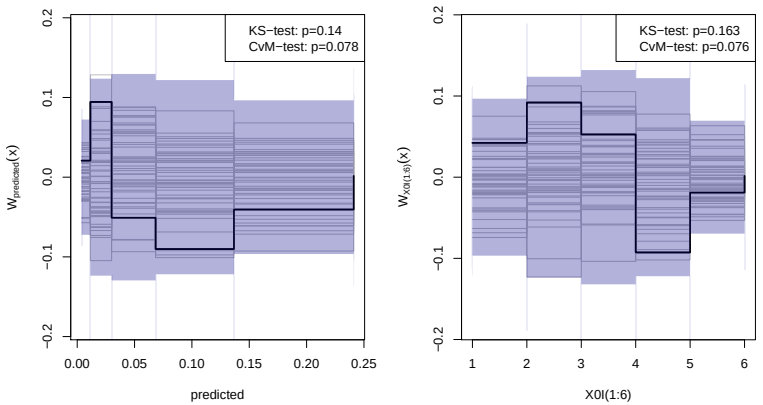
Danske Bank Business Analytics Challenge (2017)

Prediction of Default within next year using public available data. In this lecture we look at equity of start up's, that is companies less the 1 year old:

Default within next year	Equity group (numeric) 1	2	3	4	5	6	Total
Yes	3	4	5	11	5	1	29
No	4	14	86	506	231	92	933
Total	7	18	91	517	236	93	962

Results from a probit analysis (What is the effect? + Model validation):

Tolerance distribution for Default	Estimate (95% CI)
mean μ	-0.79 $(-2.98$; $1.40)$
standard deviation σ	2.55 $(1.75$; $4.55)$



- Quiz: What is your opinion about this analysis?

Odds and Odds-ratio

Towards logistic regression

- The interpretation via tolerance distribution is somewhat awkward for the "Default within next year" example.
- The answer to the following question (which is ill-defined in the probit model) might have a more natural interpretation:

How more likely are start up's to default within the next year compared to start up's in 1 higher Equity group (e.g. 2 vs. 3)?

- A possible answer could be formulated via the odds $= \frac{P(\text{event})}{P(\text{no event})}$:

$$\text{Odds}_{\text{group}=2} = \frac{P(\text{Default}|\text{Group}=2)}{P(\text{no-Default}|\text{group}=2)}$$

$$\text{Odds}_{\text{group}=3} = \frac{P(\text{Default}|\text{group}=3)}{P(\text{no-Default}|\text{group}=3)}$$

And the odds ratio: $\text{OR}_{2:3} = \frac{\text{Odds}_{\text{group}=2}}{\text{Odds}_{\text{group}=3}}$

Data example 2: Start up's defaults revisited

Default within next year	Equity group (numeric) 1	2	3	4	5	6	Total
Yes	3	4	5	11	5	1	29
No	4	14	86	506	231	92	933
Total	7	18	91	517	236	93	962

Odds	3/4	4/14	5/86	11/506	5/231	1/92	29/933

Odds-ratio	$\frac{3/4}{4/14}$	$\frac{4/14}{5/86}$	$\frac{5/86}{11/506}$	$\frac{11/506}{5/231}$	$\frac{5/231}{1/92}$	—	—
	2.625	4.914	2.674	1.004	1.991	—	—

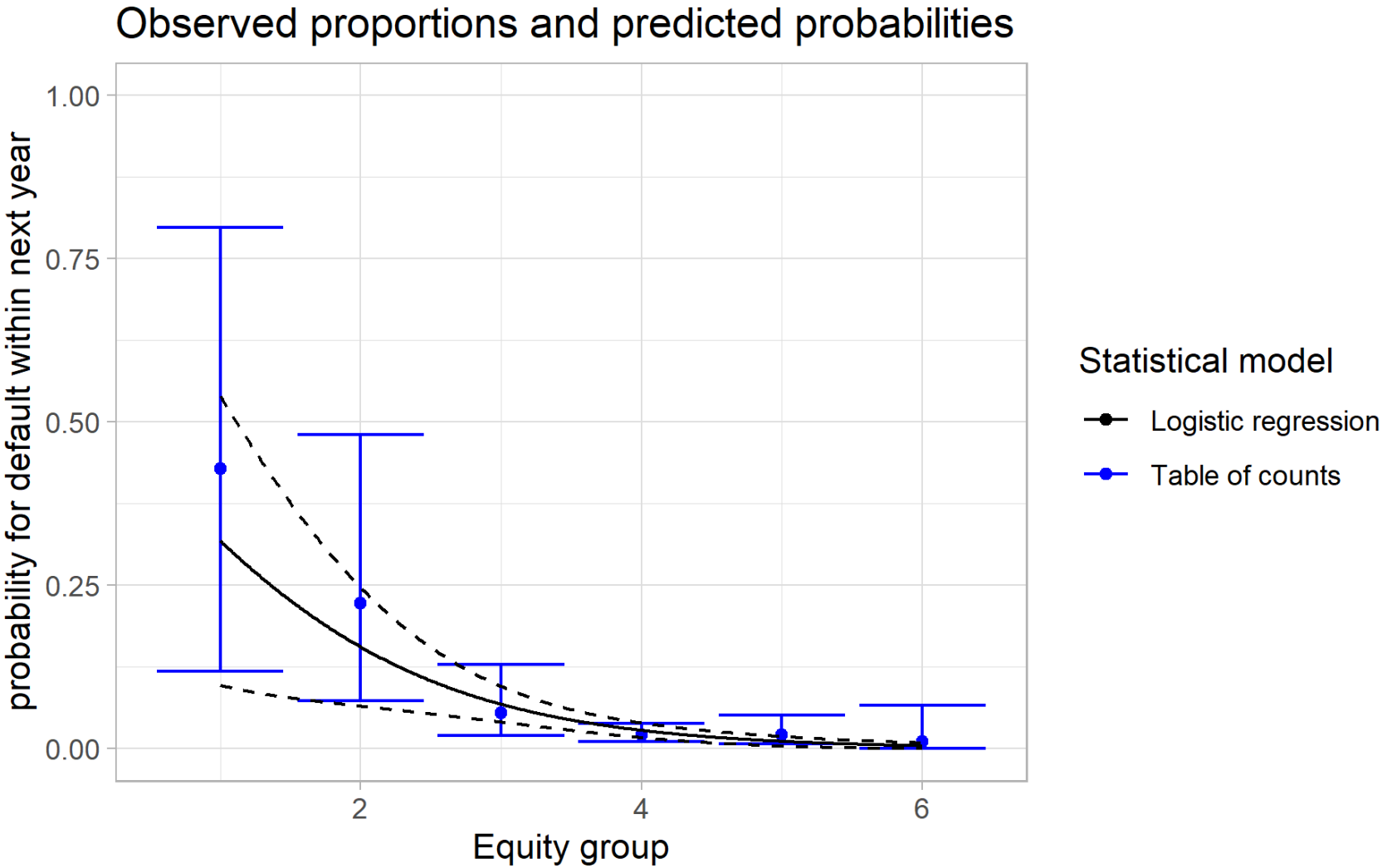
- Logistic regression models the log(odds) by a line:

$$\log(\text{odds}) = \alpha + \beta \cdot \text{group}$$

- This implies constant odds ratios:

$$\log(\text{OR}_{g:g+1}) = \log(\text{Odds}_g) - \log(\text{Odds}_{g+1}) = \alpha + \beta \cdot g - \alpha - \beta \cdot (g+1) = -\beta$$

3 advantages of the linear model



- Quiz: What are the advantages of a logistic regression (today) over the analysis via a table of counts (last week)?

Lack-of-Fit test

The examples given so far may be represented in a table of counts (ie. the topic of Day 2). The saturated model assigns an event probability to each group. Typically, the regression models have fewer parameters:

Example	Parameters in full model	Parameters in regression model
Beetle	8 (= levels of CS_2)	2 (intercept, logdose)
Default	6 (= number of Equity groups)	2 (intercept, group)

- The null hypothesis of the Lack-of-Fit test is validity of the regression model. This is tested against the saturated model.
- The Lack-of-Fit test is a Goodness-of-Fit test.

Questions?

- If needed, then let's have a break.
- Thereafter we discuss model selection and methods of answering the question What is the effect?

Data example 3: Hypertension (yes/no) for 433 men

Explanatory categorical variables: smoking, obese, snoring

Smoking	Obese	Snoring	Hypertension	No hypertension
no	no	no	5	55
yes	no	no	2	15
no	yes	no	1	7
yes	yes	no	0	2
no	no	yes	35	152
yes	no	yes	13	72
no	yes	yes	15	36
yes	yes	yes	8	15

- All interactions between 3 factors on 2 levels: $2^3 = 8$ parameters, i.e. the saturated model. In particular, Lack-of-Fit test is meaningless.

Model selection (digression from today's main topic)

How to find the "best" model, e.g. select variables

There are disagreements about how to approach this. The following 3 possibilities go from *"wrong + practicable"* to *"correct + impracticable"*:

- **Backward model selection:** Start from a valid model, and remove non-significant effects one-by-one, preferably the least significant first, until all remaining effects are significant.
- **Best subset selection:** Try all possible submodels, and select the best model according to some criterion. In practice the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) often are used.
  - R: preferably done automatically using `step()`, or possibly `MASS::stepAIC()` or `MuMIn::dredge()`.
  - Actually, `MuMIn::dredge()` as default uses a biased-corrected version of AIC known as AIC_c. This is always preferable over AIC.
- **Don't:** Instead choose model based on other knowledge.

Automated model selection

```
# Load libraries. And read data from text file
library(gof); hypertension <- read.delim("hypertension.txt")

# Make saturated logistic regresion
m1 <- glm(cbind(yes,no)~snoring*obese*smoking,
          data=hypertension,family=binomial)

# Automated model selection using AIC
step(m1,direction="both")

# Investigation of selected model
m2 <- glm(cbind(yes,no)~snoring+obese,
          data=hypertension,family=binomial)
drop1(m2,test="Chisq")
plot(cumres(m2))
exp(cbind(OR=coef(m2),confint(m2)))
```

Results of analysis

- Final model contains main effects of **snoring** and **obese**.
- Effects preferably reported as odds ratios found by taking the exponential of the parameter estimates:

Comparison	Odds Ratio	Lower 95% CL	Upper 95% CL
Snoring vs. non-snoring	2.3761	1.1514	5.5609
Obese vs. non-obese	2.0045	1.1336	3.4792

- Odds ratios are multiplicative, i.e. the OR for hypertension of a snoring, obese man against a non-snoring, non-obese man is:

$$\text{OR} = 2.3761 * 2.0045 = 4.7629$$

How to report estimates of model parameters?

Exemplified by logistic regression for hypertension data: Final model has 3 parameters.

```
> cbind(log_odds=coef(m2),confint(m2))
Waiting for profiling to be done...
              log_odds      2.5 %    97.5 %
(Intercept) -2.3920763 -3.2101098 -1.718094
snoringyes   0.8654583  0.1410076  1.715763
obeseyes     0.6954188  0.1254244  1.246789
```

- This model is so simple that parameters "easily" can be combined and backtransformed to interpretable statements.
- In general, however, dealing with model parametrizations is highly technical.
- When parameters have a specific interpretation by themselves, you may of course use this. Otherwise, I recommend that you use the `emmeans`-package.
- Name refers to estimated marginal means. Corresponds to *least squares means* for normally distributed responses, but the methodology is generally applicable.

Interpretation of parameters in logistic regressions

Using the emmeans-package

Predictions in the linear models are of logit = log odds. Thus, backtransformation by "expit" function leads to probabilities.

Interpretation of parameters in logistic regressions

Using the emmeans-package

Predictions in the linear models are of logit = log odds. Thus, backtransformation by "expit" function leads to probabilities.

Here's how to do this in R:

```
> emmeans(m2,~snoring*obese,type="response")
 snoring obese       prob          SE  df  asymp.LCL asymp.UCL
 no      no    0.08377892 0.02884212 Inf 0.04194600 0.1603493
 yes     no    0.17848906 0.02293162 Inf 0.13786495 0.2279191
 no      yes   0.15490233 0.05750643 Inf 0.07191419 0.3024487
 yes     yes   0.30339158 0.05174310 Inf 0.21231081 0.4130561

Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

- Option `type="response"` requests backtransformation.
- Output `df=Inf` suggests that confidence interval are made using a normal approximation (a technicality you may ignore).

Interpretation of parameters in logistic regressions

Using the emmeans-package

Contrasts between parameters = differences of log odds = log odds ratios.
Thus, backtransformation by "exp" function lead to odds ratios.

Interpretation of parameters in logistic regressions

Using the emmeans-package

Contrasts between parameters = differences of log odds = log odds ratios.
Thus, backtransformation by "exp" function lead to odds ratios.

```
> confint(pairs(emmeans(m2,~snoring*obese,type="response"),reverse=TRUE))
 contrast          odds.ratio        SE  df asymp.LCL asymp.UCL
 yes,no / no,no     2.3760948 0.9425066 Inf 0.8576329  6.583034
 no,yes / no,no     2.0045485 0.5714244 Inf 0.9637539  4.169337
 no,yes / yes,no    0.8436315 0.4259395 Inf 0.2305902  3.086488
 yes,yes / no,no    4.7629972 2.2456531 Inf 1.4185464 15.992528
 yes,yes / yes,no   2.0045485 0.5714244 Inf 0.9637539  4.169337
 yes,yes / no,yes   2.3760948 0.9425066 Inf 0.8576329  6.583034

Confidence level used: 0.95
Conf-level adjustment: tukey method for comparing a family of 4 estimates
Intervals are back-transformed from the log odds ratio scale
```

- Option `reverse=TRUE` switches reference level from `"yes"` to `"no"`.
- Adjustment of confidence intervals allows simultaneous interpretation. If you don't want this, then use option `adjust="none"`.

Questions?

- And then a break.
- After the break we discuss ordinal regression (using the proportional odds model) and Poisson regression.

Data example 4: Taste of Cheeses

Proportional odds model for ordinal regression

Cheese additive	Taste score (1=worst, 9=best)									Total
	1	2	3	4	5	6	7	8	9	
A	0	0	1	7	8	8	19	8	1	52
B	6	9	12	11	7	6	1	0	0	52
C	1	1	6	8	23	7	5	1	0	52
D	0	0	0	1	3	7	14	16	11	52

- Depending of the taste requirements we might say that a cheese is tasty if its score is at least j (for some j=1,. . . ,9).
- The proportional odds model assumes that the odds ratios for being tasty between the cheeses do not depend on the cut-off point j.

Data example 4: Taste of Cheeses

Proportional odds model for ordinal regression

Cheese additive	Taste score (1=worst, 9=best)									Total
	1	2	3	4	5	6	7	8	9	
A	0	0	1	7	8	8	19	8	1	52
B	6	9	12	11	7	6	1	0	0	52
C	1	1	6	8	23	7	5	1	0	52
D	0	0	0	1	3	7	14	16	11	52

- Depending of the taste requirements we might say that a cheese is tasty if its score is at least j (for some j=1,. . . ,9).
- The proportional odds model assumes that the odds ratios for being tasty between the cheeses do not depend on the cut-off point j.
- Table of variables for the data in `cheese.txt`:

Variable	**Type**	**Range**	**Usage**
cheese	Nominal	A, B, C, D	Fixed effect
taste	Ordinal	$1 < 2 < \cdots < 9$	Response
count	Count	[0 ; 23]	Frequency variable

Cheese example: R analysis (I)

Numerical problems in the multinomial regression solved using non-default optimizer

```
# Load library we will be using
library(ordinal)

# Read data from text file
cheese <- read.delim("cheese.txt")

# Recode 'taste' as a factor. Otherwise clm() doesn't work
cheese$taste <- factor(cheese$taste)

# Fit multinomial and proportional odds model
m0 <- clm(taste~1,nominal=~cheese,data=cheese,
          weights=count,control=list(method="nlminb"))
m1 <- clm(taste~cheese,data=cheese,weights=count)
```

Cheese example: R analysis (II)

```
# Lack-of-Fit test for proportional odds assumption
anova(m1,m0)

# Significance test for effect of 'cheese'
drop1(m1,test="Chisq")

# Estimates for confidence intervals for OR's
# for being tasty between cheeses
exp(cbind("OR vs cheese A"=coef(m1)[9:11],confint(m1)))

# emmeans-package can be used for clm-objects, but
# automatic backtransformation is not available!?
library(emmeans)
confint(pairs(emmeans(m1,~cheese),reverse=TRUE))
```

Results from analysis

Proportional odds assumption & Is there an effect?: Likelihood ratio tests

- Proportional odds assumption: $\chi^2 = 20.308$, df=21, p=0.5018
- Effect of cheese: $\chi^2 = 148.45$, df=3, $p < 2.2 * 10^{-16}$
- Estimated odds ratios for being more tasty:

	OR vs cheese A	2.5 %	97.5 %
cheeseB	0.0350	0.0148	0.0796
cheeseC	0.1809	0.0862	0.3708
cheeseD	5.0168	2.4095	10.7474

- Thus, cheese D is the most tasty. It is 5 times as tasty as cheese A ($\sim$ the second most tasty additive).

Data example 5: Number of greenflies on lettuce leaves

System (conventional/ecological), Week (1 or 2 before harvest), Leave (inner/outer)

Number of greenflies	2 weeks before		1 week before	
	outer	inner	outer	inner
conventional	5	2	29	39
ecological	32	22	38	46

- What is the relation between number of greenflies and the factors **system**, **week** and **leave**?
- The response variable **number** contains counts, and may take the values 0,1,2,. . .

Poisson regression

- The standard probability model for counts is the Poisson distribution, which may be parametrized by the intensity $\lambda > 0$:

$$P(\text{count} = y) = \frac{\lambda^y}{y!}\mathrm{e}^{-\lambda}, \qquad \text{mean count} = \lambda$$

- Poisson regression models the log-intensity as a linear function f of the explanatory variables, i.e. for the greenflies example:

$$\text{number} \sim \text{Poiss}(\lambda), \qquad \log(\lambda) = f(\text{system}, \text{week}, \text{leave})$$

- Significant effects are often reported in relative risks:

$$\text{RR}_{1:2} = \frac{\lambda_1}{\lambda_2}, \qquad \log(\text{RR}_{1:2}) = \underbrace{\log(\lambda_1) - \log(\lambda_2)}_{=f(\lambda_1)-f(\lambda_2)}$$

Number of greenflies: Poisson regression

```
# Load libraries. And read data from text file
library(gof); greenflies <- read.delim("greenflies.txt")

# Make saturated Poisson regresion
m1 <- glm(number~system*week*leave,
          data=greenflies,family=poisson)

# Automated model selection. NB: AIC based.
step(m1,direction="both")

# Investigation of selected model
m2 <- glm(number~system+week+leave+system:week+week:leave,
    data=greenflies,family=poisson)
drop1(m2,test="Chisq")
plot(cumres(m2))
exp(cbind(RR=coef(m2),confint(m2)))
```

Greenflies on lettuce leaves: Presentation of results

- A stepwise model selection using the Akaike Information Criterion was made starting from the saturated model given by the main effects and interactions (up-to third order) of the factors **system**, **week** and **leave**.

- The final model is given by the 3 main effects, and the 2-way interactions **system:week** and **week:leave**.

- Some estimated relative-risks in the final model are:

Ecological vs. Conventional	Estimate	Lower-CL	Upper-CL
at 1 week before harvest	1.2353	0.8937	1.7074
at 2 weeks before harvest	7.7143	3.4767	17.1169

But how are these estimates derived?

Ecological vs. Conventional, at 2 week before harvest

log(relative risk) = f(condition 1) - f(condition 2)

The parameters in the final model and the weights needed to construct the above contrast are:

	logRR	weight
(Intercept)	3.6382784	1-1=0
systemecological	0.2113091	1
week2 before	-2.6251883	1-1=0
leaveouter	-0.2379586	0
systemecological:week2 before	1.8317648	1
week2 before:leaveouter	0.6708227	0

Ecological vs. Conventional, at 2 week before harvest

log(relative risk) = f(condition 1) - f(condition 2)

The parameters in the final model and the weights needed to construct the above contrast are:

	logRR	weight
(Intercept)	3.6382784	1-1=0
systemecological	0.2113091	1
week2 before	-2.6251883	1-1=0
leaveouter	-0.2379586	0
systemecological:week2 before	1.8317648	1
week2 before:leaveouter	0.6708227	0

But it's more easy to let `emmeans()` do this:

```
> library(emmeans)
> confint(pairs(lsmeans(m2,~system|week),reverse=TRUE),
          type="response")
```

Summary (I)

- For regression of binary (yes/no) responses special attention was given to the model interpretation:
  - Probit analysis is adequate for dosis-response experiments.
  - Logistic regression is adequate to quantify risk factors.
- Model validation was done using two methods:
  - Cumulative residuals and associated Goodness-of-Fit tests. This should be a standard tool. Unfortunately the method is not (yet!) available for the proportional odds model.
  - Lack-of-Fit tests against a saturated model. In particular, this is useful to test the proportional odds assumption.

Summary (II)

- Backtransformation of model parameters was discussed:
  - In the categorical regressions the parameters are often given on a logarithmic scale. Thus, we backtransform parameter contrasts by the exponential function to go from **log(odds)** to **odds**.
  - For the probit analysis a non-linear combination of the model parameters was needed to get the LD50. Confidence intervals were found using the so-called Delta method.
  - The `emmeans`-package in many cases can do much of this work.

- In this lecture we didn't discuss the important concept of overdispersion. This will be discussed on Day 5.