Categorical regression

Anton Rask Lundborg
arl@math.ku.dk

Copenhagen Causality Lab
Department of Mathematical Sciences

December 6, 2023

Why do statistics?

Brief summary of Day 1 and 2

1. Is there an effect?
   - Answered by hypothesis testing via *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 be discussed later in the course.

3. What is the effect?
   - Answered by estimates with confidence intervals, and by prediction intervals.
   - Power as a remedy for risk of Type S error and size of Type M error (Gelman & Carlin, 2014).
   - 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	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	Sons		
(row pct)	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?

Categorical regression

Properties of good statistical models

- **Valid (not falsified)**
  - “All models are wrong”, but a statistical model must be valid, i.e. not falsified. This means that the probabilistic properties implied by the model are met 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 becomes (and the more often it may be invalid).

What is regression analysis?

Here is 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	11–24
Binomial	Logistic regression	25–40
Nominal	Multinomial logistic regression	43
Ordinal	Proportional odds model	41–45
Counts	Poisson regression	46–51

- 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.com`. May be installed via these steps:
    - @Windows users: Must first install Rtools bundle (not an R package!)
    - @All: `install_packages("devtools")`
    - @All: `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 – Dose-response experiment

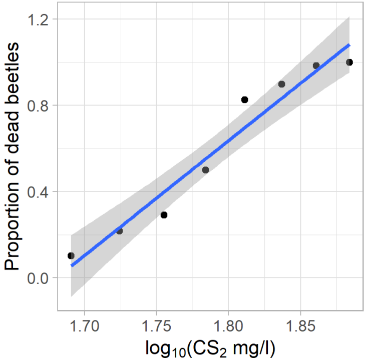
481 beetles were exposed to 8 different doses of carbon-disulfide (CS_2) for 5 hours. Mortality in each dose group was registered:

CS_2 mg/l	alive	dead	total	$\hat{p}_{\text{dead}}$
49.06	53	6	59	0.10
52.99	47	13	60	0.21
56.91	44	18	62	0.29
60.84	28	28	56	0.50
64.76	11	52	63	0.82
68.69	6	53	59	0.89
72.61	1	61	62	0.98
76.54	0	60	60	1.00
total	190	291	481	0.60

Variables used in the R analysis:

$n = \texttt{total}$, $y = \texttt{dead}$, $x = \log_{10}(\texttt{dose})$

Linear regression:



Is linear regression sensible here?

Probit regression

Probit assumptions and interpretation

- Suppose the ith beetle has a tolerance value T_i for $\log(\text{CS}_2)$, i.e. the beetle dies if log-dose 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 ith beetle is exposed to log-dose of CS_2 of size x_i.

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

Then the probability that the ith beetle dies equals

$$p_i = \mathbb{P}(T_i \leq x_i) = \Phi\Big(\frac{x_i - \mu}{\sigma}\Big)$$

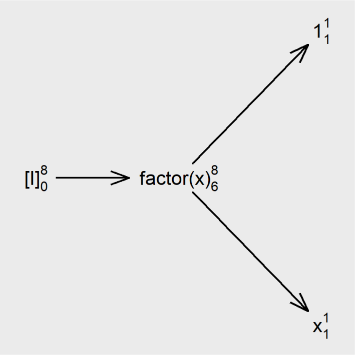
This implies a straight line with intercept $\alpha = -\frac{\mu}{\sigma}$ and slope $\beta = \frac{1}{\sigma}$:

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

Mortality of beetles – Table-of-Variables & Overview of design

Variable	Type	Range	Usage
n=total	Integer	[56; 63]	parameter
y=dead	Binomial count	[0; 53]	response
$x = \log_{10}(\texttt{dose})$	Numerical	[1.691; 1.884]	fixed effect

- dose will be used on log-scale as this gives a better fit to the data.
- Since the numerical variable dose 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

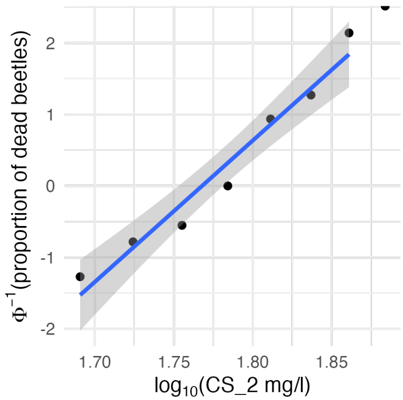
```
# Use dataset from dobson package
library(dobson)
data(beetle)

# Perform probit regression
m1 <- glm(cbind(y, n - y) ~ x,
  data = beetle,
  family = binomial(link = "probit")
)
```

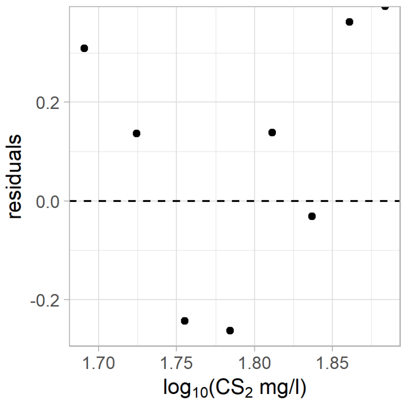
- The response consists of number of successes (dead beetles) and failures (alive beetles). These are combined column-wise, that is as variables, using `cbind()`.
- Note that we are using $x = \log_{10}(\texttt{dose})$ as the explanatory variable.

Does the probit model fit the beetle data?

Fitted model:



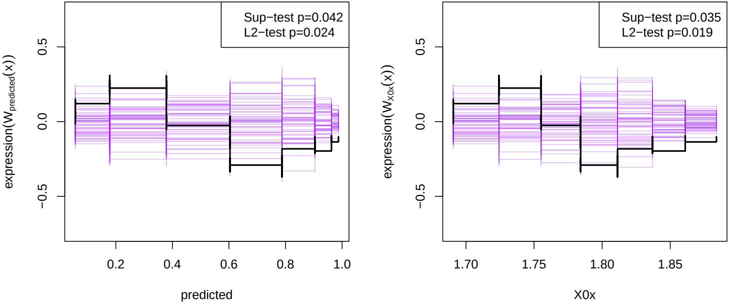
Residuals against log-dose:



- 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, i.e. without structure.

Idea: Investigate cumulative residuals

Cumulative residuals also behave in certain ways when model is correct. Black line should look similar to purple lines:



- One plot for the model + One plot for each for the continuous explanatory variables (here $x = \log_{10}(\texttt{dose})$ to the right). R code:

```
plot(gof::cumres(m1))
```

- Goodness-of-Fit tests (p-values in corner of plot) are based on simulations. (Default number of simulations is low!)

More model validation: Lack-of-Fit test

The following is only possible since “few” different doses were used:

```
# Construct a model where dose is used as a categorical factor
m0 <- glm(cbind(y, n - y) ~ factor(x),
  data = beetle,
  family = binomial(link = "probit")
)
# Lack-of-Fit test: Test m1 as a hypothesis in m0
anova(m1, m0, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: cbind(y, n - y) ~ x
Model 2: cbind(y, n - y) ~ factor(x)
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6      10.12
2         0       0.00  6    10.12   0.1197
```

- What is the conclusion?

Checkpoint

- Questions?
- After the break we discuss hypothesis testing and confidence intervals in the probit model.
- We then discuss logistic regression as an alternative to probit analysis.

Time for a break!

Hypothesis tests: Is there an effect?

Despite model being invalidated by GoF-tests (see slide 15), we continue the analysis.

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

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

```
Single term deletions

Model:
cbind(y, n - y) ~ x
        Df Deviance     AIC    LRT  Pr(>Chi)
<none>       10.12  40.318
x       1   284.20 312.400 274.08 < 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))
```

```
Waiting for profiling to be done...
                estimate     2.5 %    97.5 %
(Intercept) -34.93527 -40.28936 -29.92940
x            19.72794  16.91488  22.73983
```

Interpretation via tolerance distribution: $p = \Phi\big(\alpha + \beta \cdot x\big)$

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 dose 50% = mean in tolerance distribution
$\sigma = 1/\beta$	Scale = standard deviation in tolerance distribution

- Confidence interval for σ may be found by $1/z$-transforming the interval for $\hat{\beta} = 19.72794$. Interpretation of $\beta < 0$ via, cf. slide 10,

$$\begin{aligned}\mathbb{P}(T_i \geq x_i) = 1 - \mathbb{P}(T_i < x_i) &= 1 - \Phi\big(\alpha + \beta \cdot x_i\big)\\ &= \Phi\big(-\alpha - \beta \cdot x_i\big)\end{aligned}$$

- 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, the `emmeans_ED()` function from the `LabApplStat`-package can be used to find confidence intervals using the so-called Delta-method.
    - Alternatively the `deltaMethod` from the car-package might be used.

Backtransformation and confidence intervals

```
# Scale parameter in the tolerance distribution
1 / cbind(estimate = coef(m1), confint(m1))[2, c(1, 3, 2)]

# Mean parameter in the tolerance distribution
emmeans_ED(m1, p = 0.5, tran = "log10")
```

```
Waiting for profiling to be done...
    estimate      97.5 %       2.5 %
0.05068954 0.04397570 0.05911953

grid    estimate       SE  df asymp.LCL asymp.UCL
overall    1.771 0.003803 Inf     1.763     1.778

Results are given on the log10 (not the response) scale.
Confidence level used: 0.95
```

Summary of beetle example – see `beetle.R`

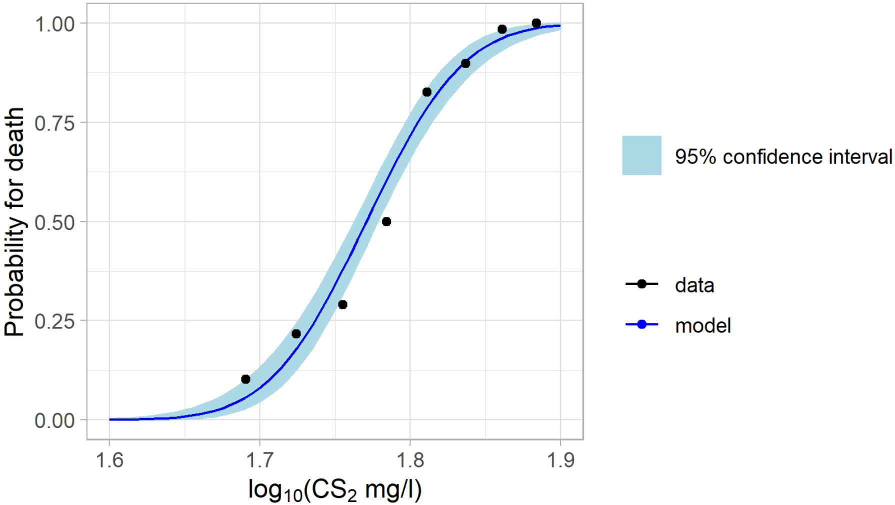
- A probit analysis of the death probability against $x = \log_{10}(\mathrm{CS}_2 \text{ dose})$ was performed.

- Model validity was investigated by cumulative residuals and associated Goodness-of-Fit tests, as well as a Lack-of-Fit test.
    - In this example the model was actually invalidated by the cumulative residuals (L2 gof-test gave $p = 0.02$). So in principle, we should not 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



Logistic regression

Data example 2: Danske Bank Business Analytics Challenge (2017)

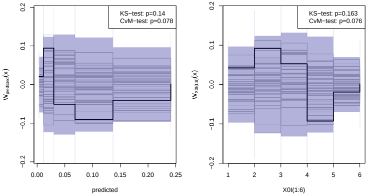
Prediction of Default within next year using publicly available data. In this lecture we look at equity of startups (=companies less than 1 year old):

Default within	Equity group (numeric)						
next year	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: Is this a valid analysis? Is this an interpretable analysis?



Odds and odds-ratios – Towards logistic regression

- The interpretation via a 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 much more likely are startups to default within the next year compared to startups in an Equity group one higher (e.g. 2 vs. 3)?

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

$$\text{Odds}_{\text{group}=2} = \frac{P(\text{Default}|\text{group}=2)}{P(\text{no-Default}|\text{group}=2)}, \quad \text{Odds}_{\text{group}=3} = \frac{P(\text{Default}|\text{group}=3)}{P(\text{no-Default}|\text{group}=3)}$$

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

Data example 2: Startup 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	—	—

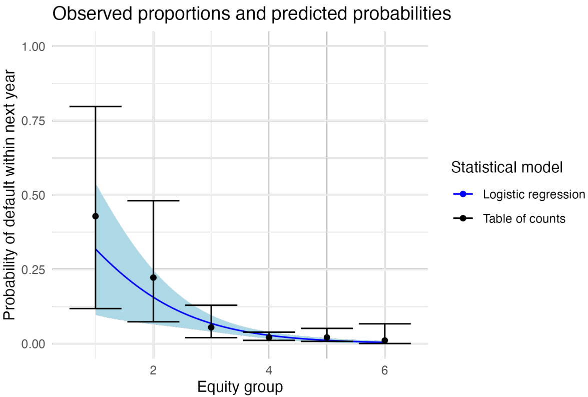
- Logistic regression models the log(odds) by a line:

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

- This implies constant odds ratios:

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

Three advantages of the linear model



- interpolation between groups
- smaller confidence intervals
- using that the probability is a smooth function of group

- 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 (i.e. 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, x)
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.

R code for company default example – see company_default.R

```
# Data is loaded in data.frame "young"

# Logistic regression
m1 <- glm(Default_nextyear ~ group, data = young, family = binomial())

# Fit saturated model
m0 <- glm(Default_nextyear ~ factor(group), data = young,
          family = binomial())

# Lack-of-fit test
anova(m1, m0, test = "Chisq")
```

Checkpoint

- Questions?
- After the break we discuss model selection and methods for answering the question of What is the effect?

Time for a break!

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 – 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 + practical" to "correct + impractical":

- **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) are often 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.
- **Do not use data:** Instead choose model based on other knowledge.

Automated model selection – see hypertension.R

```
# Load library and import data
library(gof); hypertension <- read.delim("hypertension.txt")

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

# Automated model selection. (AIC-based)
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, R=10000)) # R increases number of simulations
exp(cbind(OR = coef(m2), confint(m2)))
```

Results of analysis

- Final model contains main effects of `snoring` and `obese` (3 total parameters).
- 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:

$$\mathrm{OR} = 2.3761 \cdot 2.0045 = 4.7629$$

How to report estimates of model 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 can "easily" be combined and back-transformed 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, it is recommended that you use the `emmeans`-package.
- Name refers to estimated marginal means. Corresponds to means of the response for different combinations of covariates.

Interpretation of parameters in logistic regressions – using `emmeans`

Predictions in the linear models are of logit = log odds. Thus, back-transformation 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 back-transformation.
- Output `df=Inf` suggests that confidence intervals are made using a normal approximation (a technicality you may ignore).

Interpretation of parameters in logistic regressions – using emmeans

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         2.376 0.943 Inf     0.858      6.58
    no yes / no no         2.005 0.571 Inf     0.964      4.17
    no yes / yes no        0.844 0.426 Inf     0.231      3.09
    yes yes / no no        4.763 2.246 Inf     1.419     15.99
    yes yes / yes no       2.005 0.571 Inf     0.964      4.17
    yes yes / no yes       2.376 0.943 Inf     0.858      6.58

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 do not want this, then use option `adjust = "none"`.

Checkpoint

- Questions?
- After the break we discuss ordinal regression (using the proportional odds model) and Poisson regression.

Time for a break!

Proportional odds model

Data example 4: Tasting cheeses – proportional odds for ordinal regression

Cheese	Taste score (1=worst, 9=best)									
additive	1	2	3	4	5	6	7	8	9	Total
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 on the taste requirements we might say that a cheese is tasty if its score is at least j (for some $j = 1, \ldots, 9$).
- The proportional odds model assumes that the odds ratios for being tasty between the cheeses does 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) – see cheese.R

Below we use a non-default optimizer for clm to avoid a numerical issue.

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

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

# The response variable (taste) should be recoded as a factor for clm() to work
cheese$taste <- factor(cheese$taste)
str(cheese)

# 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) – see cheese.R

```
# 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?

- Proportional odds assumption: $\chi^2 = 20.308$, df $= 21$, $p = 0.5018$

- Effect of cheese: $\chi^2 = 148.45$, df $= 3$, $p < 2.2 \cdot 10^{-16}$

- 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 (the second most tasty additive).

Poisson regression

Data example 5: Number of greenflies on lettuce leaves

Explanatory variables: System (conventional/ecological), Week (1 or 2 before harvest), Leaf (inner/outer)

Number of	2 weeks before		1 week before	
greenflies	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 `leaf`?

- 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 $\lambda > 0$:

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

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

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

- 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 – see greenflies.R

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

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

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

# Investigation of selected model
m2 <- glm(number ~ system + week + leaf + system:week + week:leaf,
  data = greenflies, family = poisson()
)
drop1(m2, test = "Chisq")
plot(cumres(m2, R=10000)) # R controls the number of simulations
exp(cbind(RR = coef(m2), confint(m2)))
```

Greenflies on lettuce leaves: Presentation of results

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

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

- 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 weeks before harvest, inner leaf

We use the formula log(relative risk) = f(condition 1) - f(condition 2):

$$f(\text{Ecological, 2 weeks before, inner leaf}) = \texttt{Intercept} + \texttt{system ecological}$$

$$+\,\texttt{week2 before} + \texttt{systemecological:week2 before}$$

$$f(\text{Conventional, 2 weeks before, inner leaf}) = \texttt{Intercept} + \texttt{week2 before},$$

so the relative risk is

$$\exp(\texttt{system ecological} + \texttt{systemecological:week2 before}).$$

It is much easier to let emmeans do this:

```
> library(emmeans)
> confint(pairs(emmeans(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 appropriate for dose-response experiments.
  - Logistic regression is appropriate 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)

- Back-transformation of model parameters was discussed:
  - In the categorical regressions the parameters are often given on a logarithmic scale. E.g. 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 did not discuss the important concept of overdispersion. This will be discussed on Day 5.