TMA4315 Generalized linear models H2018

Module 4: Count and continuous positive response data (Poisson and gamma regression)

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

27.09.2018 and 04.10.2018 [PL], 28.09.2018 and 05.10.2018 [IL]

(Latest changes: 06.10: solutions added. 01.10: small changes for second week. 27.09: added one Problem for ILw1, moved stuff to w2, added a few dimensions to score test.)

Overview

Learning material

- Textbook: Fahrmeir et al (2013): Chapter 5.2, 5.3.
- Classnotes 27.09.2018
- Classnotes 04.10.2018

Topics

First week

- examples of count data
- the Poisson distribution
- regression with count data
- Poisson regression with log-link
- parameter estimation (ML): log-likelihood, score vector, information matrix to give iterative calculations
- asymptotic MLE properties
- confidence intervals and hypothesis tests (Wald, score and LRT)

Jump to interactive (week 1)

Second week

- Count data with Poisson regression (continued)
- deviance, model fit and model choice
- overdispersion
- rate models and offset
- Modelling continuous response data: lognormal and gamma
- the gamma distribution
- the gamma GLM model
- gamma likelihood and derivations thereof
- dispersion parameter: scaled and unscaled deviance

SECOND WEEK

Poisson regression for count data

What did we do last week?

Examples — GLM model — loglikelihood, score function and Fisher information matrix — asympototic results for $\widehat{\beta}$ and Wald, score and LRT.

Deviance

The deviance is the difference in model fit between 2 models.
Three models are helpful:

- **Null model**: model with only an intercept
- **Candidate model:** The model that we are investigating. We maximize the likelihood and get $\widehat{\beta}$
- **Saturated model:** A model with a parameter for every λ_i by the observed count for observation i.

Notation:

$\widehat{\lambda}_i = \widehat{y}_i$: prediction for the candidate model for observation i
$\widetilde{\lambda}_i = y_i$: prediction for the saturated model for observation i

Model assessment and model choice: using the deviance

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

The (log-) likelihood ratio between two models is

$$
\begin{aligned}
l(\bar{\bar{\theta}}) - l(\bar{\theta}) &= \sum_{i=1}^{n} [y_i \ln(\bar{\bar{y}}_i) - \bar{\bar{y}}_i - \ln(y!)] - \sum_{i=1}^{n} [y_i \ln(\bar{y}_i) - \bar{y}_i - \ln(y!)] \\
&= \sum_{i=1}^{n} [y_i \left(\ln(\bar{\bar{y}}_i) - \ln(\bar{y}_i)\right) - \bar{\bar{y}}_i - \bar{y}_i] \\
&= \sum_{i=1}^{n} [y_i \ln(\bar{\bar{y}}_i / \bar{y}_i) - (\bar{\bar{y}}_i - \bar{y}_i)]
\end{aligned}
$$

The Saturated Deviance is

$$
D = -2l(\theta) = -2 \sum_{i=1}^{n} [y_i \ln(y_i) - y_i - \ln(y!)]
$$

because $\widehat{y}_i = y_i$ for this model

Deviance test

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

Likelihood Ratio Tests with Deviance

The likelihood ratio test can be performed using the difference between two deviances:

$$\begin{aligned} LRT &= D_1 - D_2 = -2(l_1(\hat{\pi}_1) - l(\tilde{\pi})) - (-2(l_2(\hat{\pi}_2) - l(\tilde{\pi}))) \\ &= -2(l_1(\hat{\pi}_1) - l_2(\tilde{\pi}_2)) \end{aligned}$$

This follows a χ^2_p distribution with k equal to the difference in number of parameters

Deviance test

We may use the deviance test presented in Module 3 to test if the model under study is preferred compared to the saturated model.

$$D = 2\sum_{i=1}^{n}[y_i \ln(\frac{y_i}{\widehat{y}_i}) - (y_i - \widehat{y}_i)]$$

Remark: if $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \widehat{y}_i$ then deviance will be equal to

$$D = 2\sum_{i=1}^{n} y_i \ln(\frac{y_i}{\widehat{y}_i})$$

The deviance statistic approximately follows a χ^2_{n-p}, at least when the counts are not low.

Pearson test

The Pearson χ^2-goodness of fit statistic is given as the sum of the squared Pearson residuals

$$X_P^2 = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} \frac{(y_j - \hat{y}_i)^2}{\hat{y}_i}$$

where $\hat{y}_i = \hat{\lambda}_i = \exp(\mathbf{x}_i^T \hat{\beta})$. The Pearson χ^2 statistic is asymptotically equivalent to the deviance statistic and thus is asymptotically χ^2_{n-p} (proof: do a Taylor series expansion of the deviance).

Remarks

The asymptotic distribution of both statistics (deviance and Pearson) are questionable when there are many low counts. Agresti (1996, page 990) suggest analysing grouped data, for example by grouping by width in the horseshoe crab example. The Pearson statistic is also used for testing independence in contingency tables - we will do that in Compulsory Exercise 2.

Example: goodness of fit with female horseshoe crabs

Comment on the analysis. Is this a good fit? What might a bad fit be due to?

```
model3 = glm(Sa ~ W + C, family = poisson(link = log), data
# summary(model3)
1 - pchisq(model3$deviance, model3$df.residual)
Xp = sum(residuals(model3, type = "pearson")^2)
Xp
1 - pchisq(Xp, model3$df.residual)
```

```
## [1] 0
## [1] 543.249
## [1] 0
```

AIC

Identical to Module 3 - we may use the Akaike informations criterion. Let p be the number of regression parameters in our model.

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

A scaled version of AIC, standardizing for sample size, is sometimes preferred. And, we may also use the BIC, where $2p$ is replaced by $\log(n) \cdot p$.

Analysis of deviance

Identical to Module 3 we may also sequentially compare models, and use analysis of deviance for this.

Residuals

Two types of residuals are popular: *deviance* and *Pearson*. These are based on the deviance and χ^2 statistics: basically they are the contribution of each data point to that statistic.

But the sign has to be included in the deviance residual

Deviance residuals

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

$$d_i = \text{sign}(y_i - \widehat{y}_i) \cdot \left\{ 2[y_i \ln(\frac{y_i}{\widehat{y}_i}) - (y_i - \widehat{y}_i)] \right\}^{1/2}$$

where the term $\text{sign}(y_i - \widehat{y}_i)$ makes negative residuals possible - and we get the same sign as the *Pearson residuals*

Pearson residuals

The Pearson residuals are given as

$$r_i = \frac{o_i - e_i}{\sqrt{e_i}}$$

where o_i is the observed count for observation i and e_i is the estimated expected count for observation i. We have that $o_i = y_i$ and $e_i = \hat{y}_i = \hat{\lambda}_i = \exp(\mathbf{x}_i^T \hat{\beta})$.

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

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.

Normality of residuals is not assumed, but for large counts can be reasonable, and can be checked using qq-plots as for the MLR in Module 2.

Below - notice the trend in the residuals, this is due to the discrete nature of the response. The plot with different shades of blue shows that the structures are for equal values of y.

R Code

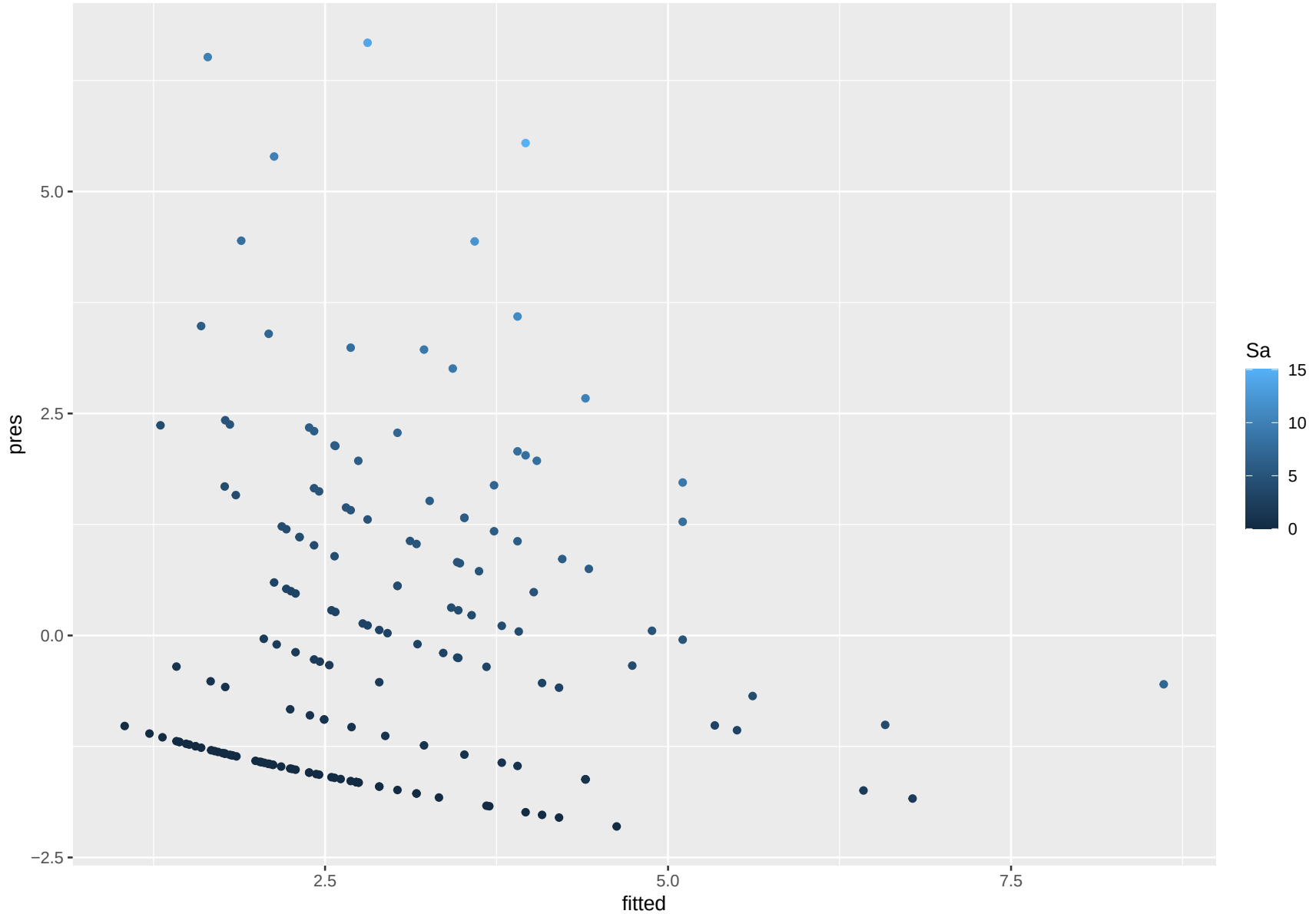
```
model3 = glm(Sa ~ W + C, family = poisson(link = log), data
    S = "contr.sum"))
df = data.frame(Sa = crab$Sa, fitted = model3$fitted.values
    type = "deviance"), pres = residuals(model3, type = "pe

library(ggplot2)
# create the plot
gg1 = ggplot(df) + geom_point(aes(x = fitted, y = dres, col
    y = pres, color = "pearson"))
```

gg1

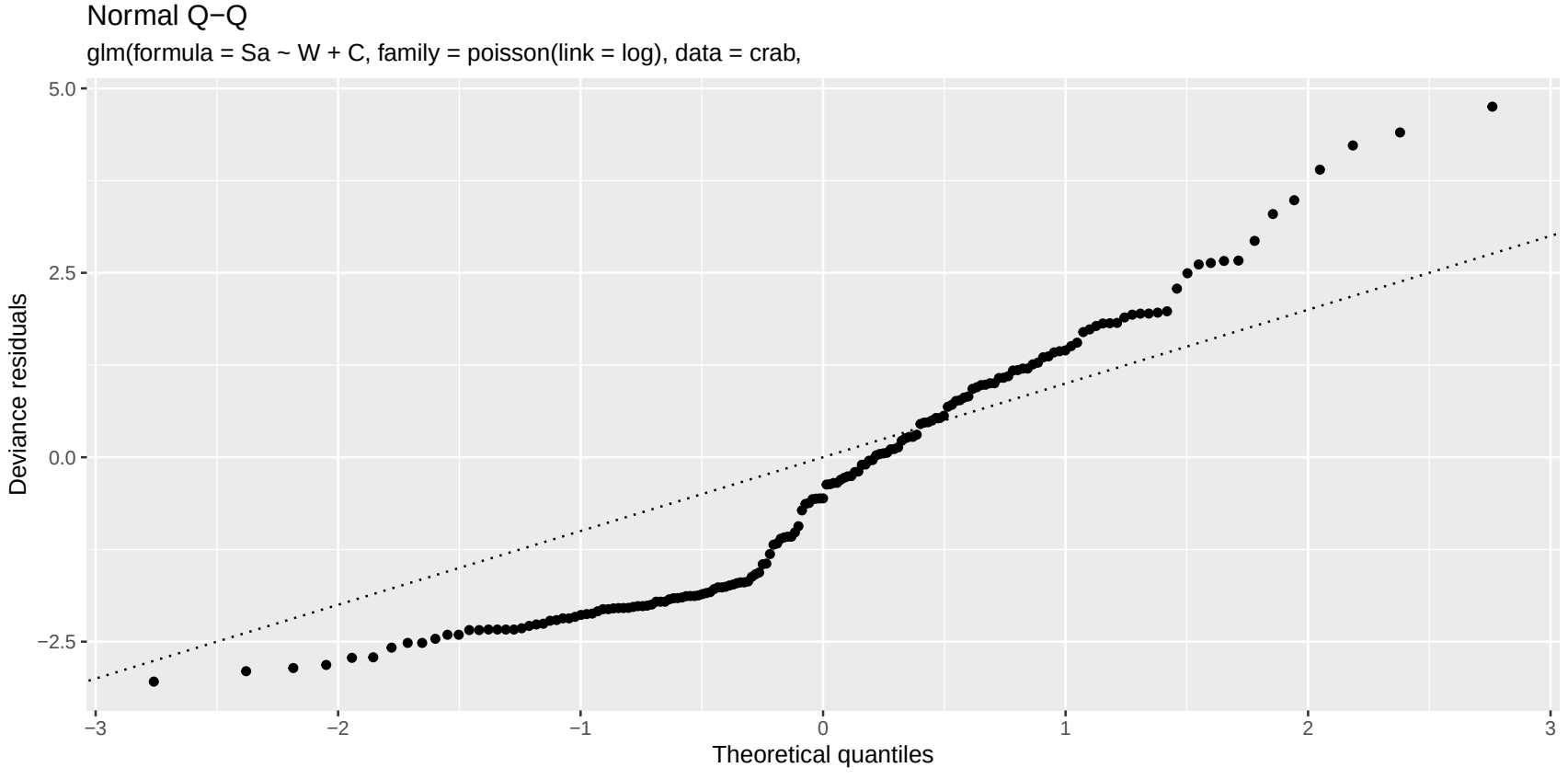
Pearson Residuals

```
gg2 = ggplot(df) + geom_point(aes(x = fitted, y = pres, col
gg2
```

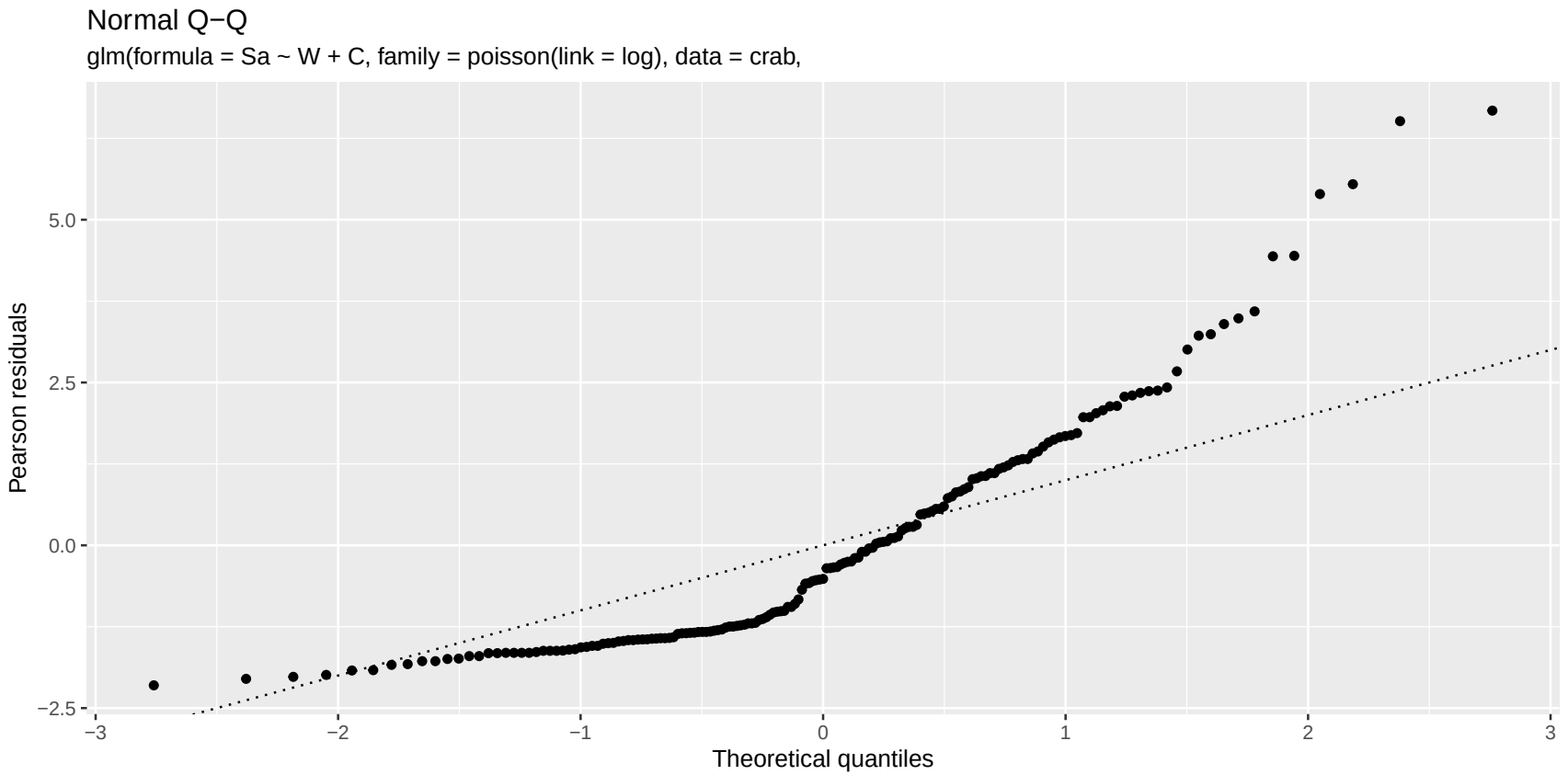


Normal Prabability Plots

```
dff = data.frame(devres = residuals(model3, type = "devianc
    type = "pearson"))
ggplot(dff, aes(sample = devres)) + stat_qq(pch = 19) + geo
    slope = 1, linetype = "dotted") + labs(x = "Theoretical
    title = "Normal Q-Q", subtitle = deparse(model3$call))
```



```
ggplot(dff, aes(sample = pearsonres)) + stat_qq(pch = 19) +
    slope = 1, linetype = "dotted") + labs(x = "Theoretical
    title = "Normal Q-Q", subtitle = deparse(model3$call))
```



Overdispersion

Count data might show greater variability in the response counts than we would expect if the response followed a Poisson distribution. This is called *overdispersion*.

Example: newspaper sales with tourist bus.

Our model states that the variance $\text{Var}(Y_i) = \lambda_i$. If we change the model to $\text{Var}(Y_i) = \phi\lambda_i$ we may allow for an increased variance due to heterogeneity among subjects.

Or, we can miss several covariate, then any daya point is a mixture of several Poisson populations, each with its own mean for the response.

This heterogeniety may give an overall response distribution where the variance is greater than the standard Poisson variance.

The overdispersion parameter can be estimated from the Pearson statistic or deviance

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

where D is the deviance. Note that similarity to $\hat{\sigma^2} = 1/(n-p) \cdot \text{SSE}$ in the MLR.

$\text{Cov}(\hat{\beta})$ can then be changed to $\hat{\phi} F^{-1}(\hat{\beta})$, so we multiply the standard error by the square root of $\hat{\phi}_D$.

Estimating Overdispersion

(more on quasipoisson later in the course)

```
model.od = glm(Sa ~ W, family = poisson(link = log), data =
model.disp = glm(Sa ~ W, family = quasipoisson(link = log)
# summary.glm(model.od)
summary.glm(model.disp)$dispersion

(OverDisp.Dev <- model.od$deviance/model.od$df.residual)
Chi2 <- sum((crab$Sa - fitted(model.od))^2/fitted(model.od)
(OverDisp.Chi2 <- Chi2/model.od$df.residual)
```

```
## [1] 3.182205
## [1] 3.320927
## [1] 3.182205
```

Rate models

In the Poisson process we might analyse an event that occurs within a time interval or region in space, and therefore it is often of interest to model the *rate* at which events occur.

Examples:

- crime rates in cities
- death rate for smokers vs. non-smokers
- rate of auto thefts in cities

We model the rates by using an *offset* to convert to counts.

We don't want a model for Y_i but for Y_i/t_i:

- Let t_i denote the index (population size in the example) associated with observation i.
- We still assume that Y_i follows a Poisson distribution, but we now include the index in the modelling and focus on Y_i/t_i.
- The expected value of Y_i/t_i would then be $\mathrm{E}(Y_i)/t_i = \lambda_i/t_i$.

A log-linear model would be

$$\log(\lambda_i/t_i) = \mathbf{x}_i^T\beta$$

We may equivalently write the model as

$$\log(\lambda_i) - \log(t_i) = \mathbf{x}_i^T\beta$$

This adjustment term is called an *offset* and is a known quantity. Equivalently we have $\log(\lambda_i) = \mathbf{x}_i^T\beta + \log(t_i)$

The expected number of outcomes will then satisfy

$$\mathrm{E}(Y_i) = \lambda_i = t_i \exp(\mathbf{x}_i^T\beta).$$

Example: British doctors and rate models

British doctors sent a questionnaire (in 1951) about whether they smoked tobacco, and later information about their deaths were collected.

Research questions:

1) Is the death rate higher for smokers than for non-smokers?
2) If so, by how much?
3) How is this related to age?

```
library(boot)
data(breslow)
# n=person-year, ns=smoker-years, y=number of deaths due t
breslow$age <- factor(breslow$age)  #age=midpoint 10 year
breslow$smoke <- factor(breslow$smoke)  # smoke=smoking st
```

Writing an offset

Here our count depends on n

(we are actually using a Poisson approximation to hte binomial)

There are 2 ways of coding an offset in R:

```
# first age and smoke (but not interaction thereof)
fit1 <- glm(y ~ age + smoke, offset = log(n), family = pois
fit1a <- glm(y ~ age + smoke + offset(log(n)), family = poi
```

Do we need an interaction?

```r
# do we need interaction?
fit2 <- update(fit1, . ~ . + smoke * age)
anova(fit1, fit2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ age + smoke
## Model 2: y ~ age + smoke + age:smoke
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         4     12.132
## 2         0      0.000  4   12.132  0.01639 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

The final model

Number of deaths per 1000 doctors.

```
# year 40 nonsmokers should only be the intercept
exp(fit2$coefficients[1]) * 1000
# 80 year olds who smoke
1000 * exp(sum(fit2$coefficients[c(1, 5, 6, 10)]))
```

```
## (Intercept)
##   0.1064396
## [1] 19.18375
```

(you can also use `predict()` to do this)

Modelling continuous positive response data

Examples of continuous positive responses

- Insurance: Claim sizes
- Medicine: Time to blood coagulation (main example)
- Biology: Time in various development stages for fruit fly
- Meteorology: Amount of precipitation (interactive session - exam question 2012)

This is also covered in **survival analysis**, but that often sidesteps modelling the actual distributions

Models for continuous positive responses

- Lognormal distribution on response
- Gamma distribution on response
- Inverse Gaussian distribution on response (we will not consider this here)

Time to blood coagulation

The data is clotting time of blood (in seconds) `y` for normal plasma diluted to nine different percentage concentrations `u` with prothrombin-free plasma (whatever that is!).
To induce the clotting a chemical called thromboplasting was used, and in the experiment two different lots of the chemical were used - denoted `lot`. Our aim is to investigate the relationship between the clotting time and the dilution percentage, and look at differences between the lots.

```r
clot = read.table("https://www.math.ntnu.no/emner/TMA4315/2
clot$lot = as.factor(clot$lot)
summary(clot)
```

```
##        u               time        lot
##  Min.   :  5   Min.   : 12.0   1:9
##  1st Qu.: 15   1st Qu.: 18.0   2:9
##  Median : 30   Median : 23.0
##  Mean   : 40   Mean   : 32.5
##  3rd Qu.: 60   3rd Qu.: 35.0
##  Max.   :100   Max.   :118.0
```

Lognormal distribution

Let Y_i be the response on the original scale, where $Y_i > 0$.

Transform the response to a logaritmic scale: $Y_i^* = \ln(Y_i)$. Then, assume that transformed responses follow a normal distribution (or follows approximately) and use ordinary MLR. This means we have a GLM with normal response and identity link (on logarithmic scale of reponse).

1. $Y_i^* \sim N(\mu_i^*, \sigma^{*2})$
2. $\eta_i = \mathbf{x}_i^T \beta$
3. $\mu_i^* = \eta_i$ (identity link)

There are two ways of looking at this,

1. either this is just a transformation to achieve approximate normality, or
2. we assume that the original data follows a lognormal distribution.

In genomics one usually assume the former, and reports back results on the exponential scale - just say that the mean of original data is $\exp(\mu_i^*)$.

However, if on instead assume that the original data really comes from a lognormal distribution, then it can be shown that

$$\mathsf{E}(Y_i) = \exp(\mu_i^*) \cdot \exp(\sigma^{*2}/2)$$
$$\mathsf{Var}(Y_i) = \exp(\sigma^{*2} - 1) \cdot \mu_i^2$$

i.e. standard deviation proportional to expectation.

Gamma regression

The gamma distribution

We have seen that a gamma distributed variable may be the result of the time between events in a Poisson process. The well known χ^2_δ-distribution is a special case of the gamma distribution ($\frac{\nu}{\mu_i} = 2$, $\nu = \frac{\delta}{2}$).

There are many parameterization for the gamma distribution, but we will stick with the one used in our textbook (page 643):

$Y_i \sim Ga(\mu_i, \nu)$ with density

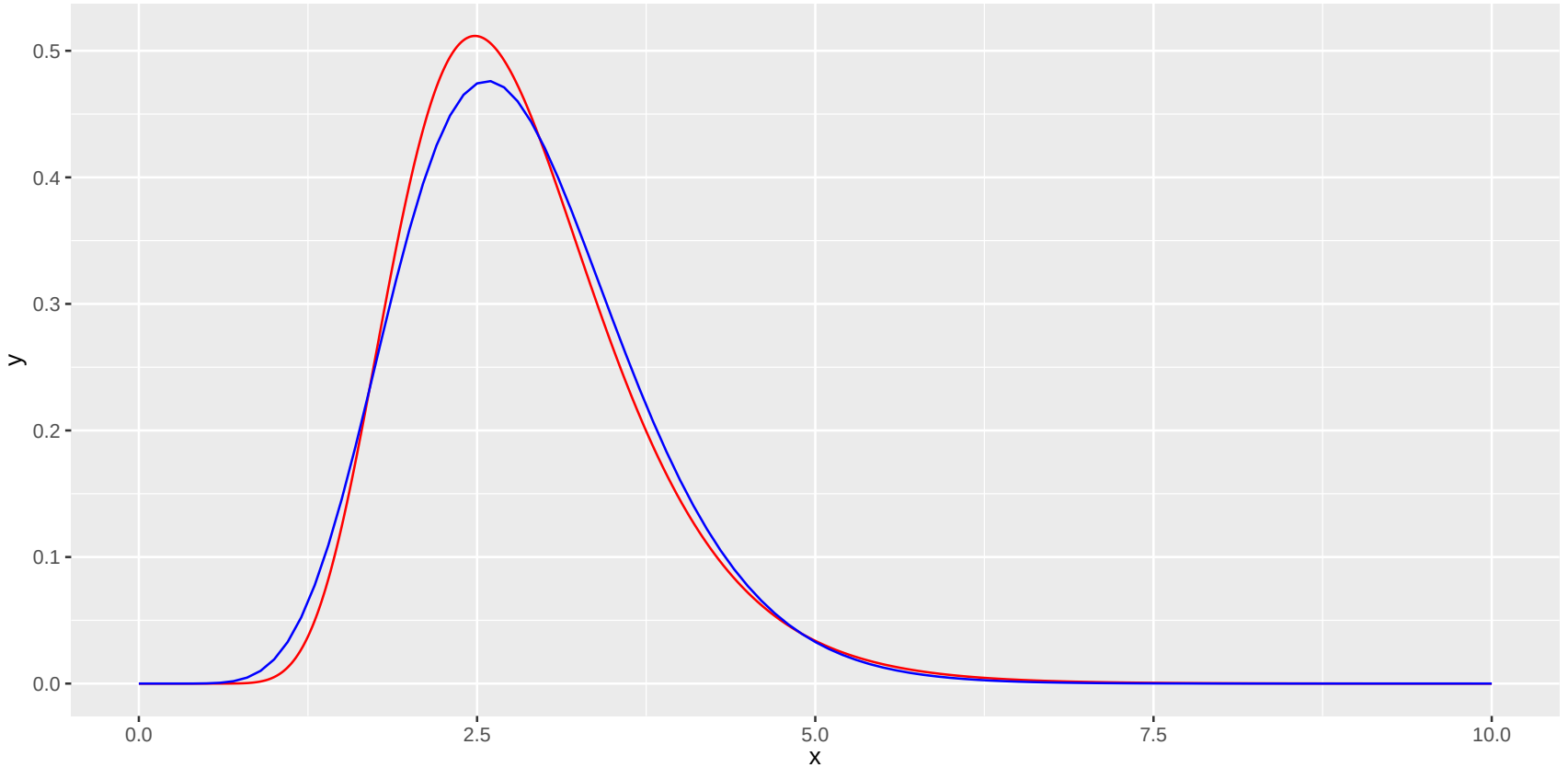
$$f(y_i) = \frac{1}{\Gamma(\nu)}(\frac{\nu}{\mu_i})^\nu y_i^{\nu-1} \exp(-\frac{\nu}{\mu_i} y_i) \text{ for } y_i > 0$$

Comparing the lognormal and gamma

```
orgmu = 1
orgsd = 0.3  # normal mean and sd
mu = exp(orgmu + orgsd^2/2)  # = shape*scale
scale = (exp(orgsd^2) - 1) * mu
shape = mu/scale
library(ggplot2)
xrange = range(0, 10)
```

These should have the same mean and variance

```
ggplot(data.frame(x = xrange), aes(xrange)) + xlab(expressi
    args = list(meanlog = orgmu, sdlog = orgsd), geom = "li
    stat_function(fun = dgamma, args = list(shape = shape,
        colour = "blue")
```



We found in Module 1 that the gamma distribution is an exponential family, with

- $\theta_i = -\frac{1}{\mu_i}$ is the canonical parameter
- $\phi = \frac{1}{\nu}$,
- $w_i = 1$
- $b(\theta_i) = -\ln(-\theta_i)$
- $\mathsf{E}(Y_i) = b'(\theta_i) = -\frac{1}{\theta_i} = \mu_i$
- $\mathsf{Var}(Y_i) = b''(\theta_i)\frac{\psi}{w_i} = \frac{\mu_i^2}{\nu}$

(if you don't remember, work it out!)

For a GLM model we have canonical link if

$$\theta_i = \eta_i$$

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

$$\theta_i = g(\mu_i) = -\frac{1}{\mu_i}$$

saying that with the canonical link is $-\frac{1}{\mu_i}$.

However, the most commonly used link is $g(\mu_i) = \ln(\mu_i)$, and the identity link is also used.

Q: Discuss the implications on η_i when using the canonical link. Why might the log-link be preferred?

Remark: often the inverse and not the negative inverse is used, and since

$$g(\mu_i) = -\frac{1}{\mu_i} = \mathbf{x}_i^T \beta$$

then

$$\frac{1}{\mu_i} = -\mathbf{x}_i^T \beta = \mathbf{x}_i^T \beta^*$$

where $\beta^* = -\beta$.

Gamma GLM model

1. $Y_i \sim Ga(\mu_i, \nu)$
2. $\eta_i = \mathbf{x}_i^T \beta$
3. Popular link functions:
   - $\eta_i = \mu_i$ (identity link)
   - $\eta_i = \frac{1}{\mu_i}$ (inverse link)
   - $\eta_i = \ln(\mu_i)$ (log-link)

Remark: In our model the parameter μ_i varies with i but ν is the same for all observations.

Example: Time to blood coagulation

A simple model to start with is as follows (dosages often analysed on log scale):

```
fit1 = glm(time ~ lot + log(u), data = clot, family = Gamma(link = log))
summary(fit1)
```

```
##
## Call:
## glm(formula = time ~ lot + log(u), family = Gamma(link = log),
##     data = clot)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.44660    0.13453   40.48  < 2e-16 ***
## lot2        -0.47034    0.07095   -6.63 8.02e-06 ***
## log(u)      -0.58476    0.03772  -15.50 1.22e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.02265072)
##
##     Null deviance: 7.7087  on 17  degrees of freedom
## Residual deviance: 0.3211  on 15  degrees of freedom
## AIC: 104.28
##
## Number of Fisher Scoring iterations: 5
```

Q: describe what you see in the print-out.

Gamma regression: likelihood and derivations thereof

Likelihood:

$$L(\beta) = \prod_{i=1}^{n} \exp(-\frac{\nu y_i}{\mu_i} - \nu \ln \mu_i + \nu \ln \nu + (\nu - 1) \ln y_i - \ln(\Gamma(\nu)))$$

Log-likelihood:

$$l(\beta) = \sum_{i=1}^{n} [-\frac{\nu y_i}{\mu_i} - \nu \ln \mu_i + \nu \ln \nu + (\nu - 1) \ln y_i - \ln(\Gamma(\nu))]$$

Observe that we now- for the first time - have a nuisance parameter ν here.

Fitting the Model

To produce numerical estimates for the parameter of interest β we may proceed to the score function, and solve using Newton Raphson or Fisher scoring. If we do not have the canonical link the observed and expected Fisher information matrix may not be equal.

What about $\phi = 1/\nu$? Also estimated using maximum likelihood.

Further analyses: as before we use asymptotic distribution of parameter estimates, and of Wald, LRT and score test.

Scaled and unscaled deviance

We have defined the deviance as

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

This is often called the *scaled deviance*.

The *unscaled deviance* is then defined as ϕD, but is sadly sometimes also called the deviance - for example by `R`.

1. For the normal model the
   - scaled deviance is $D = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2$, while
   - unscaled deviance is $\phi D = \sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2$
2. For the binomial and Poisson model $\phi = 1$ so the scaled and unscaled deviance are equal.
3. What about the Gamma model?

Some calculations - see IL week 2, problem 2: 1b.

$$D = \frac{-2\sum_{i=1}^{n}[\ln(\frac{y_i}{\hat{\mu}_i}) - \frac{y_i - \hat{\mu}_i}{\hat{\mu}_i}]}{\phi}$$

and unscaled as $\phi D = -2\sum_{i=1}^{n}[\ln(\frac{y_i}{\hat{\mu}_i}) - \frac{y_i - \hat{\mu}_i}{\hat{\mu}_i}]$.

Compare to print-out from R: the deviance in R is the *unscaled deviance*.

```
deviance(fit1)
(nu1 = 1/summary(fit1)$dispersion)
(D = -2 * nu1 * sum(log(fit1$y/fit1$fitted.values) - ((fit1
deviance(fit1) * nu1
```

```
## [1] 0.3210963
## [1] 44.14871
## [1] 14.17599
## [1] 14.17599
```

Comparing models

Comparing models based on deviance

```
fit2 = glm(time ~ lot + log(u) + lot:log(u), data = clot, f
anova(fit1, fit2)
```

```
## Analysis of Deviance Table
##
## Model 1: time ~ lot + log(u)
## Model 2: time ~ lot + log(u) + lot:log(u)
##   Resid. Df Resid. Dev Df  Deviance
## 1         15    0.32110
## 2         14    0.31576  1 0.0053352
```

The deviance table does not include ϕ, so the unscaled deviance is reported. If significance testing is done, the estimated ϕ from the largest model is used, and p-values are based on the scaled deviance.

```r
anova(fit1, fit2, test = "Chisq")
1 - pchisq((deviance(fit1) - deviance(fit2))/summary(fit2)$
    fit2$df.residual)
anova(fit1, fit2, test = "F")
1 - pf((deviance(fit1) - deviance(fit2))/summary(fit2)$disp
    fit2$df.residual, fit2$df.residual)
```

```
## Analysis of Deviance Table
##
## Model 1: time ~ lot + log(u)
## Model 2: time ~ lot + log(u) + lot:log(u)
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1        15    0.32110
## 2        14    0.31576  1 0.0053352   0.6355
## [1] 0.6355477
## Analysis of Deviance Table
##
## Model 1: time ~ lot + log(u)
## Model 2: time ~ lot + log(u) + lot:log(u)
##   Resid. Df Resid. Dev Df  Deviance      F Pr(>F)
```

Comparing models based on AIC

```
AIC(fit1, fit2)
```

```
##       df       AIC
## fit1  4 104.2763
## fit2  5 105.9738
```

Q: would you prefer `fit1` or `fit2`?

AIC can also be used when we compare models with different link functions (models that are not nested).

The literature suggests to plot y_i vs. each covariate to get a hint about which link function or transformation to use.

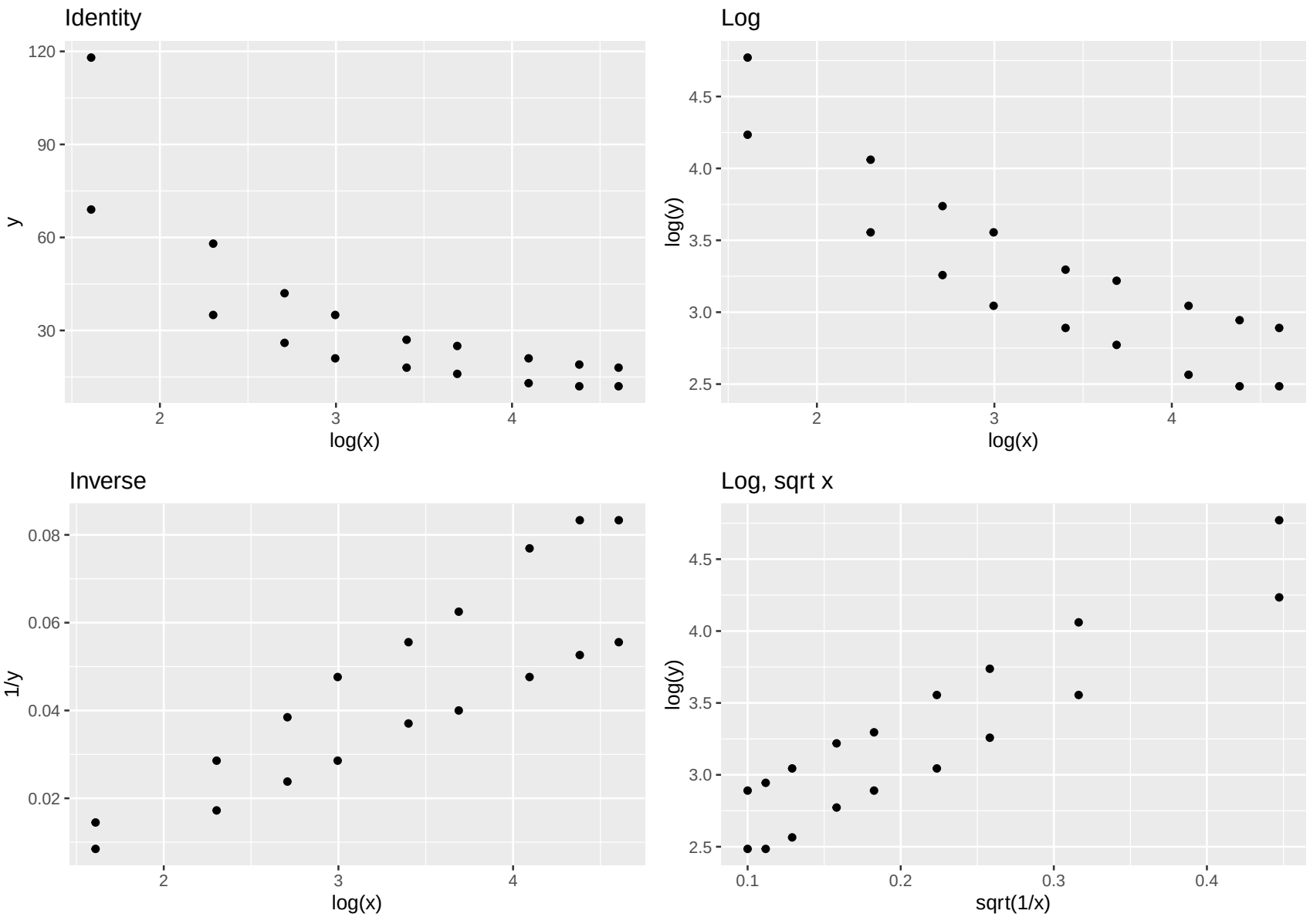
- Identity: Plot of y_i vs x_i should be close to linear
- $\ln$: Plot of $\ln(y_i)$ vs x_i should be close to linear
- Inverse (reciprocal): Plot of $1/y_i$ vs x_i should be close to linear

Compare link functions

```
library(ggplot2)
library(ggpubr)
y = clot$time
x = clot$u

df = data.frame(y = y, x = x)
gg1 = ggplot(df) + geom_point(aes(x = log(x), y = y)) + gg
gg2 = ggplot(df) + geom_point(aes(x = log(x), y = log(y)))
gg3 = ggplot(df) + geom_point(aes(x = log(x), y = 1/y)) + g
gg4 = ggplot(df) + geom_point(aes(x = sqrt(1/x), y = log(y
```

```
ggarrange(gg1, gg2, gg3, gg4)
```



```r
fit4 = glm(time ~ lot + sqrt(1/u), data = clot, family = Ga
AIC(fit1, fit4)
```

```
##       df       AIC
## fit1   4 104.27633
## fit4   4  45.01688
```

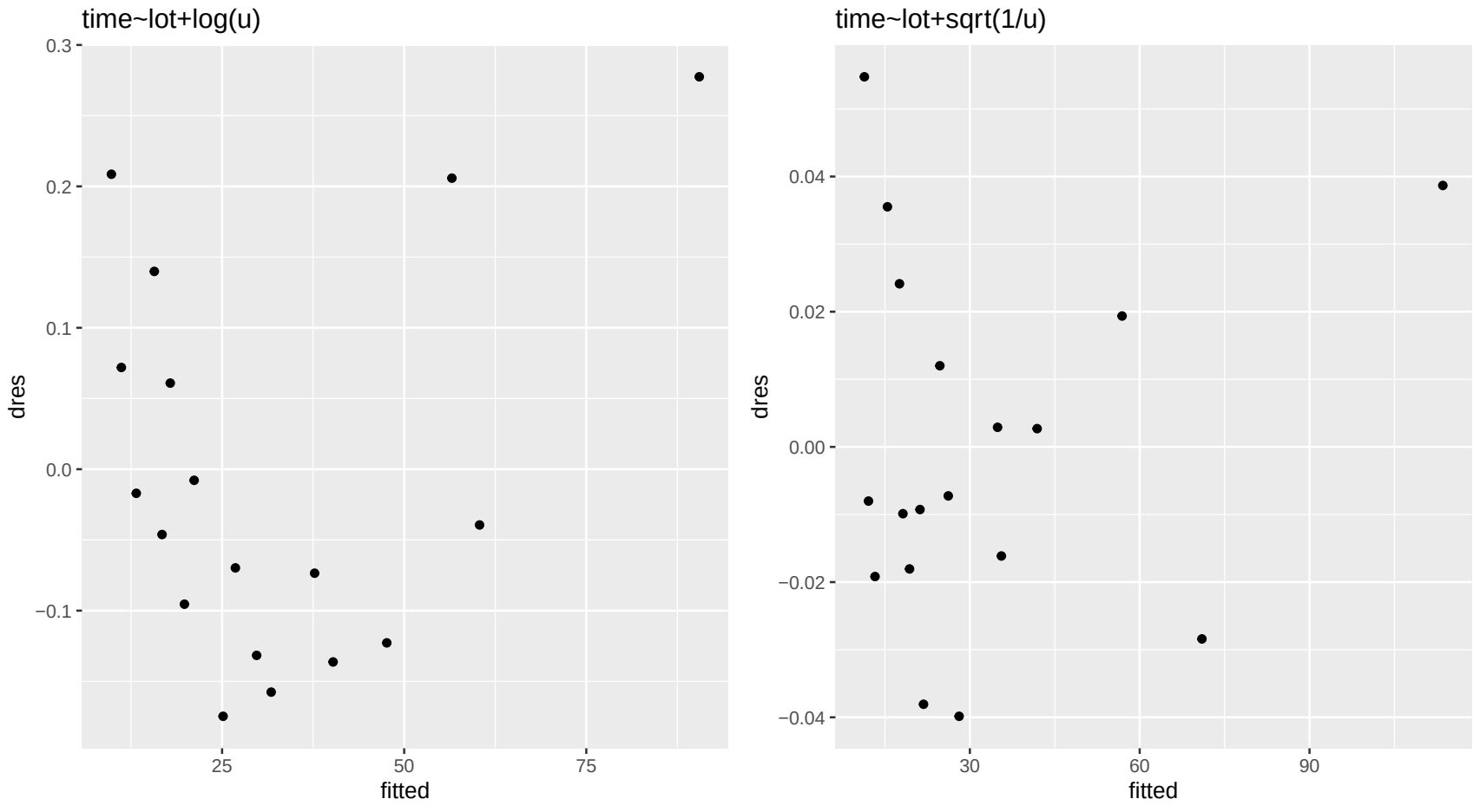
Code for Residual Plots

```
df4 = data.frame(fitted = fit4$fitted.values, dres = residu
gg4 = ggplot(df4) + geom_point(aes(x = fitted, y = dres)) +
    ggtitle("time~lot+sqrt(1/u)")
df1 = data.frame(fitted = fit1$fitted.values, dres = residu
gg1 = ggplot(df1) + geom_point(aes(x = fitted, y = dres)) +
    ggtitle("time~lot+log(u)")
```

The Plots

Are these good?

```
ggarrange(gg1, gg4)
```



R packages

```
install.packages(c("tidyverse", "ggplot2", "statmod", "corr
    "boot"))
```

Further reading

- A. Agresti (1996): "An Introduction to Categorical Data Analysis".
- 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.