Introduction to statistics: Generalized linear models (logistic regression)

Shravan Vasishth

Universität Potsdam
vasishth@uni-potsdam.de
http://www.ling.uni-potsdam.de/∼vasishth

April 19, 2020

Logistic regression

We start with an example data-set that appears in the Dobson et al book: the Beetle dataset.
This data-set shows the number of beetles killed when they were exposed to different doses of some toxic chemical.

```
(beetle<-read.table("data/beetle.txt",header=TRUE))
```

```
##      dose number killed
## 1  1.6907     59      6
## 2  1.7242     60     13
## 3  1.7552     62     18
## 4  1.7842     56     28
## 5  1.8113     63     52
## 6  1.8369     59     53
## 7  1.8610     62     61
## 8  1.8839     60     60
```

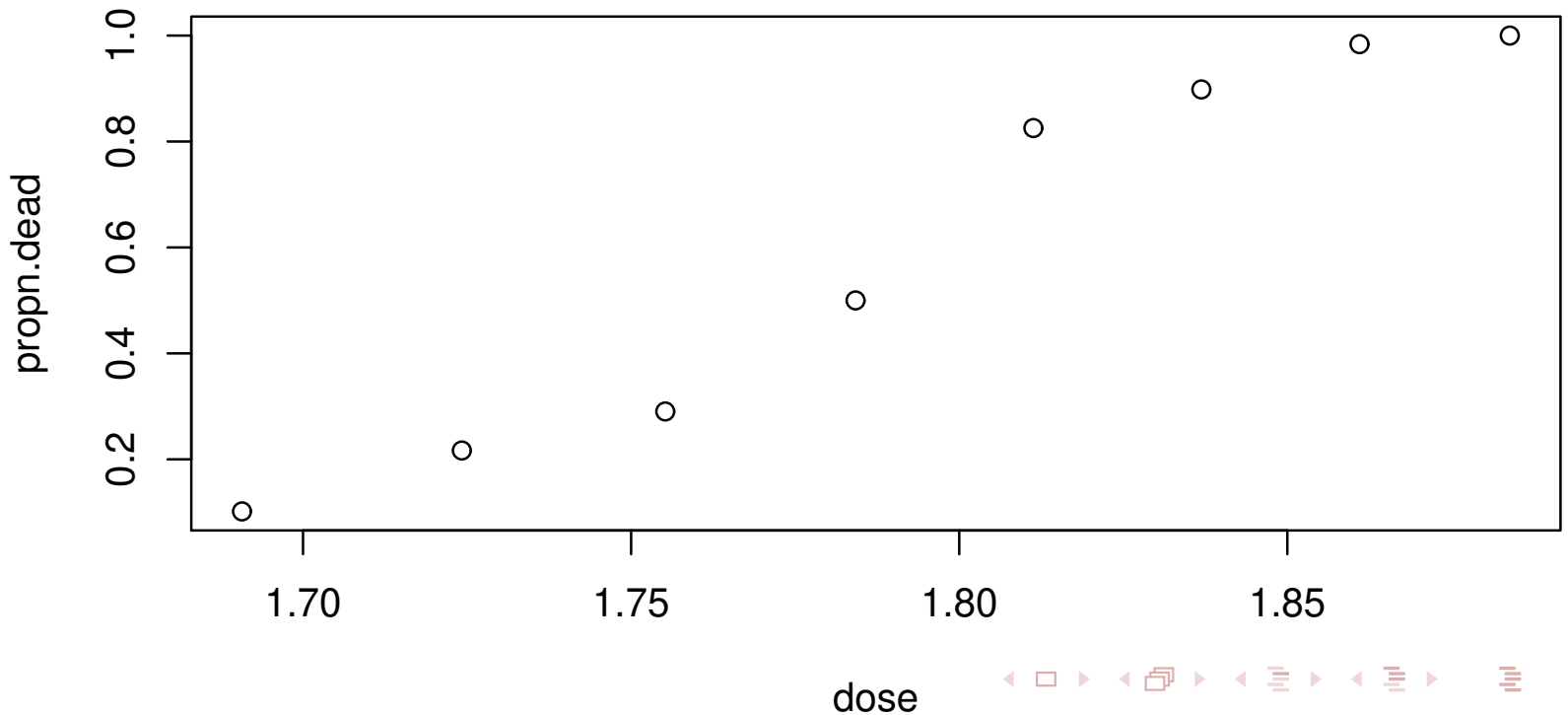
Logistic regression

The research question is: does dose affect probability of killing insects? The first thing we probably want to do is calculate the proportions:

```
(beetle$propn.dead<-beetle$killed/beetle$number)

## [1] 0.10169 0.21667 0.29032 0.50000 0.82540 0.89831 0.98
```

Logistic regression

It's also reasonable to just plot the relationship between dose and proportion of deaths.



Logistic regression

Notice that the y-axis is by definition bounded between 0 and 1. We could easily fit a linear model to this data-set. We may want to center the predictor, for reasons discussed earlier:

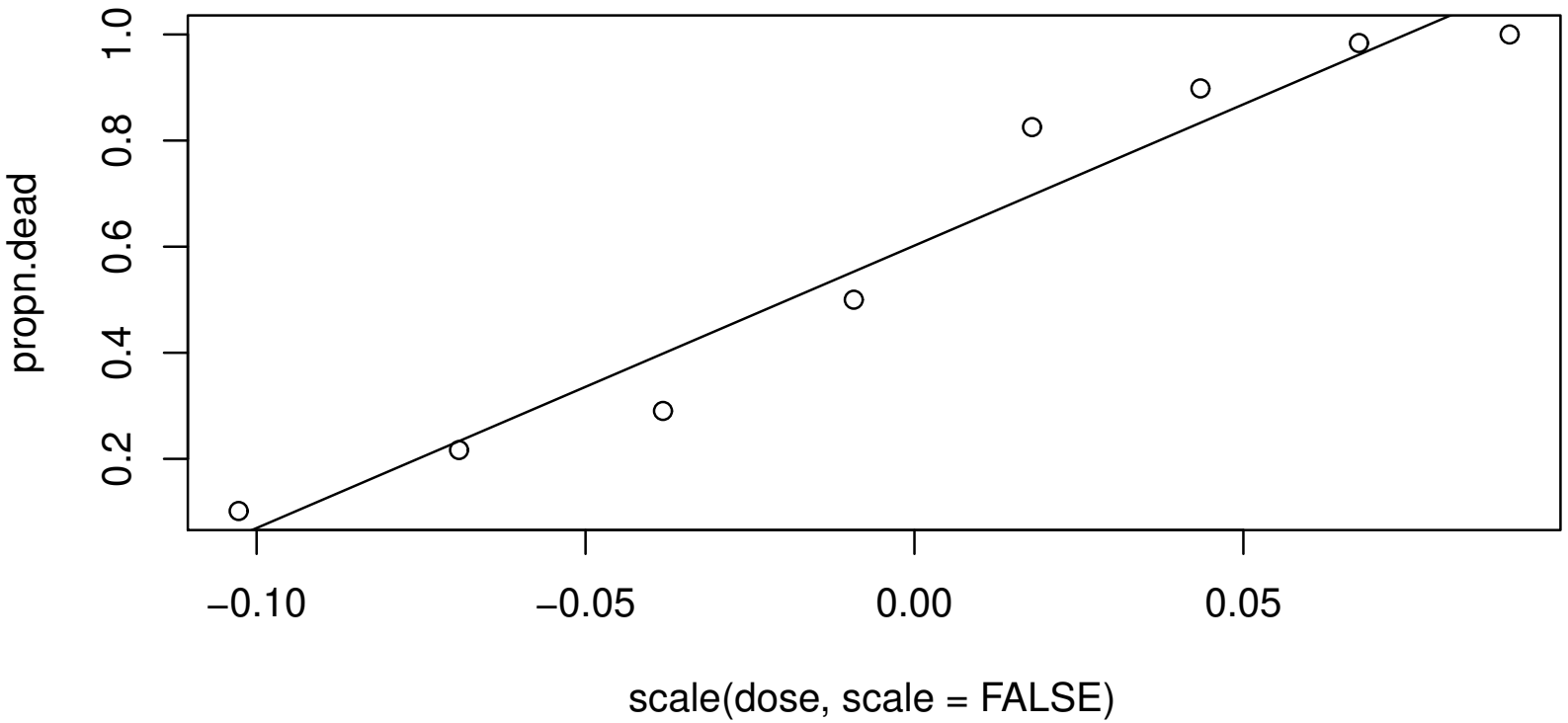
```
fm<-lm(propn.dead~scale(dose,scale=FALSE),beetle)
```

Logistic regression

```
summary(fm)
```

```
##
## Call:
## lm(formula = propn.dead ~ scale(dose, scale = FALSE), data = beetle)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.10816 -0.06063  0.00263  0.05119  0.12818
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.6020     0.0306    19.6  1.1e-06
## scale(dose, scale = FALSE)    5.3249     0.4857    11.0  3.4e-05
##
## Residual standard error: 0.0867 on 6 degrees of freedom
## Multiple R-squared:  0.952,Adjusted R-squared:  0.945
## F-statistic:  120 on 1 and 6 DF,  p-value: 3.42e-05
```

Logistic regression



Logistic regression

The interpretation of the coefficients is making little sense here. Clearly the linear model is failing us. This is the motivation for the generalized linear model.

Logistic regression

Instead of using the linear model, we model log odds instead of proportions p as a function of dose. Odds are defined as:

$$\frac{p}{1-p} \tag{1}$$

and taking the $\log$ will give us log odds.
We are going to model log odds (instead of probability) as a linear function of dose.

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 \text{dose} \tag{2}$$

The model above is called the logistic regression model.

Logistic regression

Once we have estimated the β parameters, we can move back from the log odds space to probability space using algebra.
Given a model like

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 \text{dose} \tag{3}$$

If we exponentiate each side, we get:

$$exp \log \frac{p}{1-p} = \frac{p}{1-p} = exp(\beta_0 + \beta_1 \text{dose}) \tag{4}$$

Logistic regression

So now we just solve for p, and get (check this):

$$p = \frac{exp(\beta_0 + \beta_1 \text{dose})}{1 + exp(\beta_0 + \beta_1 \text{dose})} \tag{5}$$

Logistic regression

We fit the model in R as follows. Note that as long as I am willing to avoid interpreting the intercept and just interpret the estimate of β_1, there is no need to center the predictor here:

```
fm1<-glm(propn.dead~dose,
         binomial(logit),
         weights=number,
         data=beetle)
```

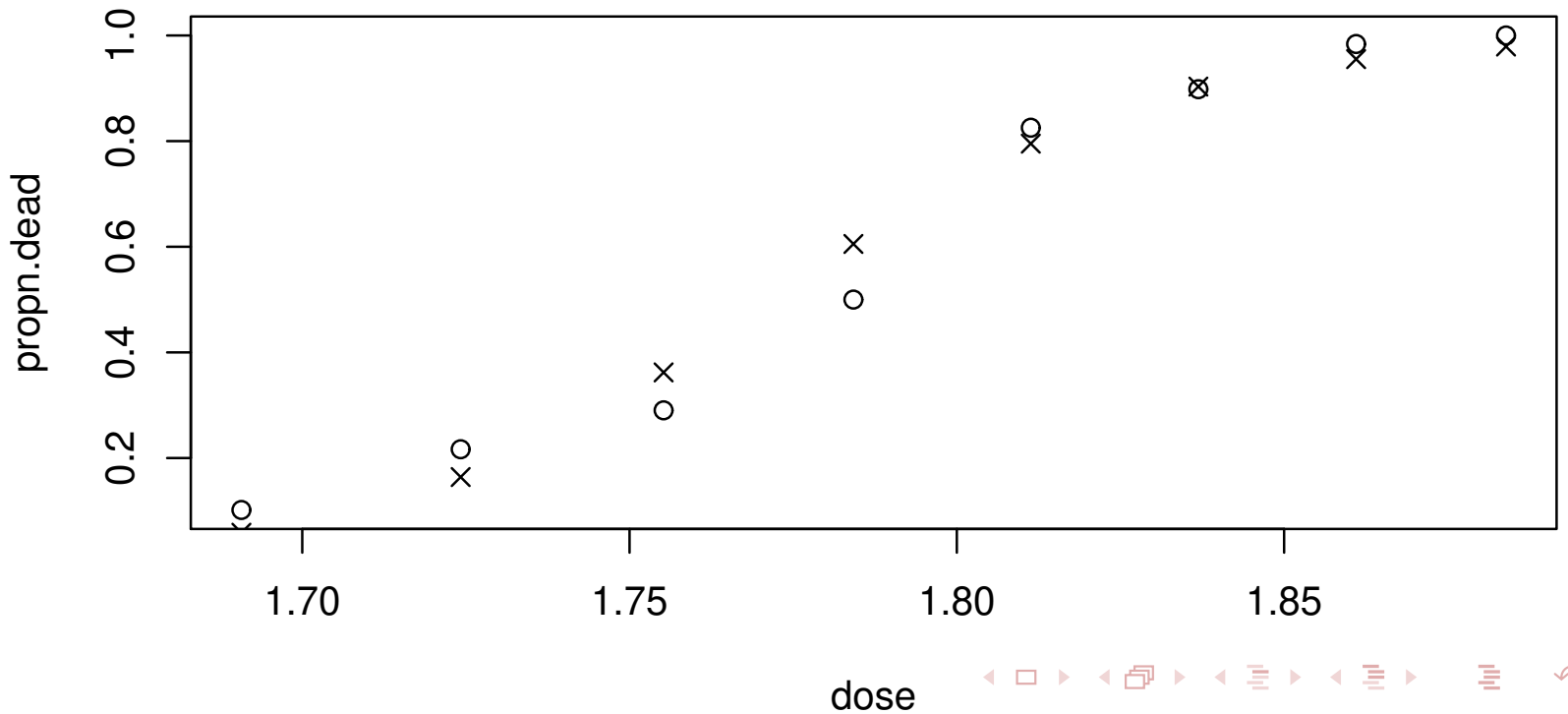
Logistic regression

```
summary(fm1)
```

```
##
## Call:
## glm(formula = propn.dead ~ dose, family = binomial(logit), data = beetle,
##     weights = number)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.594  -0.394   0.833   1.259   1.594
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -60.72       5.18   -11.7   <2e-16
## dose           34.27       2.91    11.8   <2e-16
##
## (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
```

Logistic regression

We can also plot the observed proportions and the fitted values together; the fit looks pretty good.



Logistic regression

We can now compute the log odds of death for concentration 1.7552 (for example):

```
## compute log odds of death for
## concentration 1.7552:
x<-as.matrix(c(1, 1.7552))
#log odds:
(log.odds<-t(x)%*%coef(fm1))

##           [,1]
## [1,] -0.56618
```

Logistic regression

We can also obtain the variance-covariance matrix of the fitted coefficients:

```
### compute CI for log odds:
## Get vcov matrix:
(vcovmat<-vcov(fm1))

##             (Intercept)     dose
## (Intercept)      26.840 -15.0821
## dose            -15.082   8.4805

## x^T VCOV x for dose 1.7552:
(var.log.odds<-t(x)%*%vcovmat%*%x)

##           [,1]
## [1,] 0.021678
```

Logistic regression

And using a normal approximation, we can compute the confidence interval for the log odds of death given dose 1.7552:

```
##lower
(lower<-log.odds-1.96*sqrt(var.log.odds))

##           [,1]
## [1,] -0.85476

##upper
(upper<-log.odds+1.96*sqrt(var.log.odds))

##          [,1]
## [1,] -0.2776
```

Logistic regression

The lower and upper confidence interval bounds on the probability scale can be computed by using equation 5.

```
(mean_prob<-exp(log.odds)/(1+exp(log.odds)))

##          [,1]
## [1,] 0.36212

(lower_prob<-exp(lower)/(1+exp(lower)))

##          [,1]
## [1,] 0.29844

(upper_prob<-exp(upper)/(1+exp(upper)))

##          [,1]
## [1,] 0.43104
```

Logistic regression

So for dose 1.7552, the probability of death is 0.36, with 95% confidence intervals 0.3 and 0.43.

Logistic regression

Note that one should not try to predict outside the range of the design matrix. For example, in the beetle data, the dose ranges from 1.69 to 1.88. We should not try to compute probabilities for dose 2.5, say, since we have no knowledge about whether the relationship remains unchanged beyond the upper bound of our design matrix.

Multiple logistic regression

- We have some Hindi eyetracking data (from Husain et al., 2015). We can compute skipping probability, the probability of skipping a word entirely (i.e., never fixating it).
- The predictors are: word complexity and storage complexity (SC). We expect that the higher the word complexity and the higher the storage complexity, the lower the skipping probability.
- We first have to create a vector that has value 1 if the word has 0 ms total reading time, and 0 otherwise.

Multiple logistic regression

```
hindi<-read.table("data/hindiJEMR.txt",header=TRUE)
hindi$skip<-ifelse(hindi$TFT==0,1,0)
fm_skip<-glm(skip ~ word_complex+SC,family=binomial(),hindi
```

Multiple logistic regression

```
summary(fm_skip)

##
## Call:
## glm(formula = skip ~ word_complex + SC, family = binomial(),
##     data = hindi)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.114  -0.915  -0.697   1.242   2.894
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.1512     0.0154   -9.84   <2e-16
## word_complex  -0.6398     0.0160  -39.88   <2e-16
## SC            -0.5019     0.0128  -39.27   <2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 96007  on 79942  degrees of freedom
## Residual deviance: 92457  on 79940  degrees of freedom
## AIC: 92463
##
## Number of Fisher Scoring iterations: 4
```

Multiple logistic regression

The above example also illustrates the second way to set up the data for logistic (multiple) regression: the dependent variable can simply be a 1 or 0 value instead of proportions. So, in the beetle data, you could recode the data to have 1s and 0s instead of proportions. Assuming that you have recoded the column for status (dead or alive after exposure), the glm function call would be:

```
glm(dead~dose,family=binomial(),beetle)
```

Note that logistic regression assumes independence of each data point; this assumption is violated in the Hindi data. For the Hindi data, we will have to use generalized linear mixed models.

The canonical link

- The binomial and normal distributions belong to a wider family of distributions called the exponential family.
- Other examples are: Poisson, Gamma, Probit.

For each of these exponential family distributions, there is a so-called **canonical link** that gives us the predicted values $x^T \hat{\beta}$ from the model.

The canonical link

For different distributions in the exponential family, the canonical link functions are as follows:

Distribution	$h(x_i^T\beta) = \mu_i$	Canonical link: $g(\mu_i) = \theta_i$
Binomial logit link	$\frac{exp[\theta_i]}{1+exp[\theta_i]}$	$\log \frac{y}{1-y}$
Normal identity	θ	$g = h$
Poisson log	$exp[\theta]$	$\log[\mu]$
Gamma inverse	$-\frac{1}{\theta}$	$-\frac{1}{\mu_i}$
Cloglog cloglog	$1 - exp[-exp[\theta_i]]$	$\log(-\log(1-\mu_i))$
Probit probit	$\Phi(\theta)$	$\Phi^{-1}(\theta)$ (qnorm)

Deviance

Deviance is defined as

$$D = 2[\ell(b_{max}; y) - \ell(b; y)] \tag{6}$$

where $\ell(b_{max}; y)$ is the log likelihood of the saturated model (the model with the maximal number of parameters that can be fit), and $\ell(b; y)$ is the log likelihood of the model with the parameters b. Deviance has a chi-squared distribution.

Deviance for the binomial distribution

Deviance is defined as $D = \sum d_i$, where:

$$d_i = -2 \times n_i [y_i \log(\frac{\hat{\mu}_i}{y_i}) + (1 - y_i) \log(\frac{1 - \hat{\mu}_i}{1 - y_i})] \tag{7}$$

The basic idea here is that if the model fit is good, Deviance will have a χ^2 distribution with $N - p$ degrees of freedom.
N is the number of data points, p the number of parameters.
So that is what we will use for assessing model fit.
We will also use deviance for hypothesis testing.

Deviance for the binomial distribution

The difference in deviance (residual deviance) between two models also has a χ^2 distribution (this should remind you of ANOVA), with dfs being $p - q$, where q is the number of parameters in the first model, and p the number of parameters in the second.
I discuss hypothesis testing first, then evaluating goodness of fit using deviance.

Deviance for the binomial distribution

Returning to our beetle data, let's say we fit our model:

```
glm1<-glm(propn.dead~dose,binomial(logit),
          weights=number,data=beetle)
```

Deviance for the binomial distribution

The summary output shows us the number of iterations that led to the parameter estimates:

```
summary(glm1)

##
## Call:
## glm(formula = propn.dead ~ dose, family = binomial(logit), data = beetle,
##     weights = number)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.594  -0.394   0.833   1.259   1.594
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -60.72       5.18   -11.7   <2e-16
## dose           34.27       2.91    11.8   <2e-16
##
## (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
```

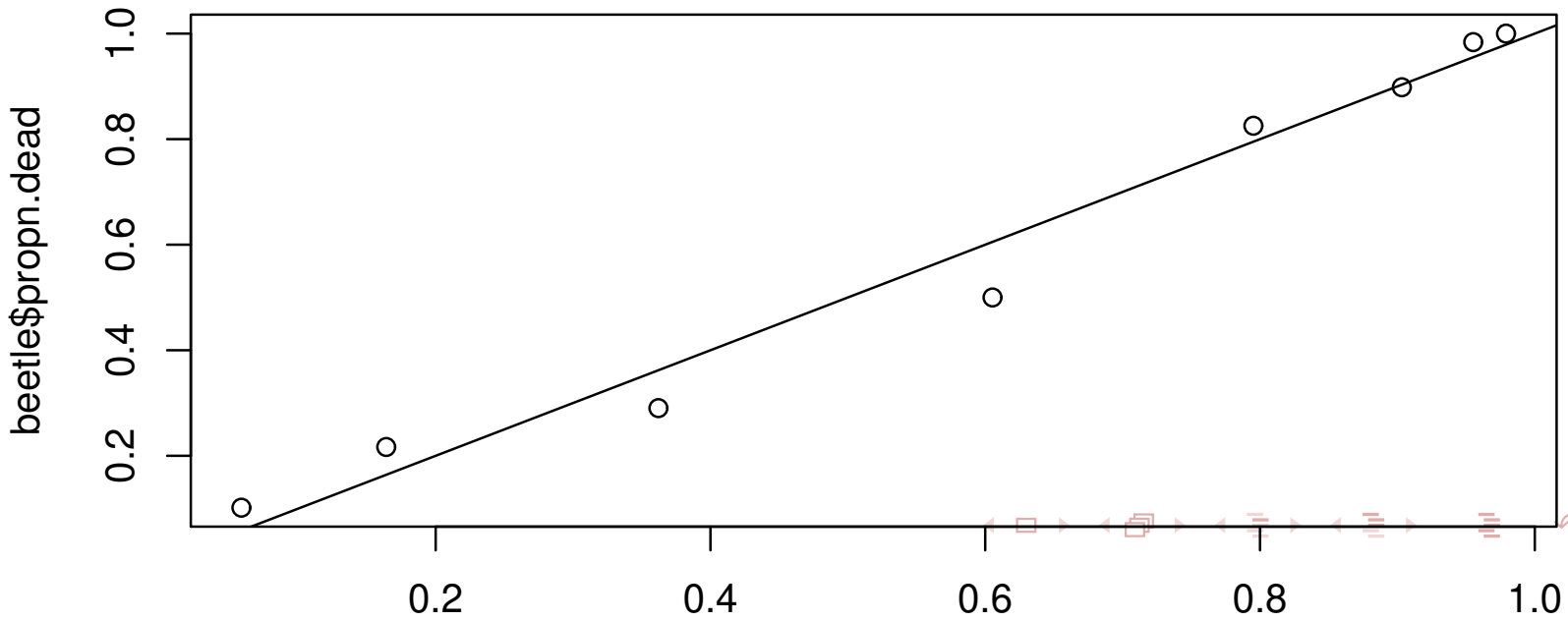
Deviance for the binomial distribution

But we also see something called **Null deviance** and **Residual deviance**. These are used to evaluate quality of model fit. Recall that we can compute the fitted values and compare them to the observed values:

```
# beta.hat is (-60.71745 ,   34.27033)
(eta.hat<-  -60.71745 +   34.27033*beetle$dose)

## [1] -2.77660 -1.62855 -0.56617  0.42767  1.35640  2.2337

(mu.hat<-exp(eta.hat)/(1+exp(eta.hat)))

## [1] 0.058602 0.164030 0.362122 0.605318 0.795174 0.90323
```

Deviance for the binomial distribution

```
# compare mu.hat with observed proportions
plot(mu.hat,beetle$propn.dead)
abline(0,1)
```



Deviance for the binomial distribution

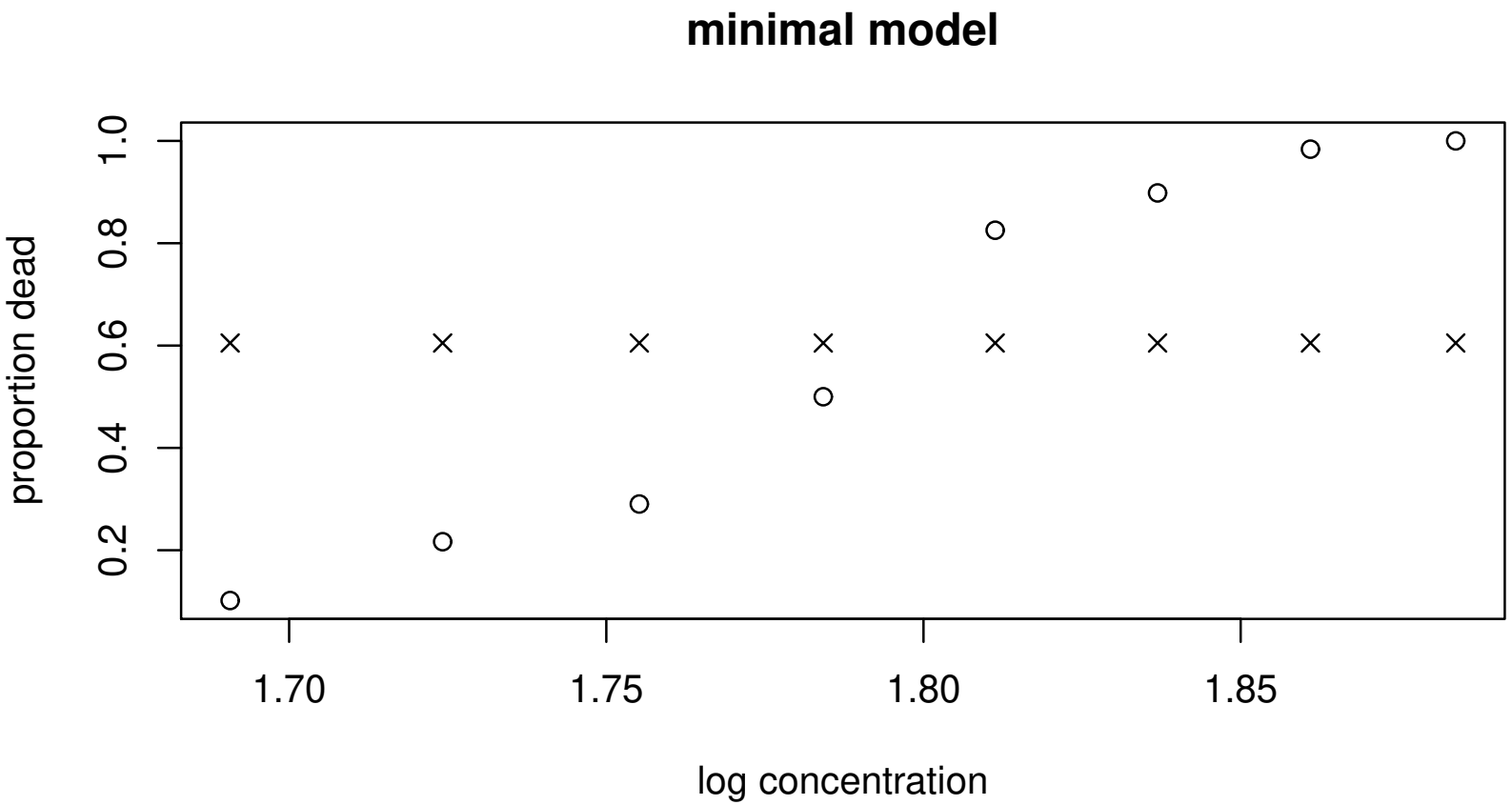
To evaluate whether dose has an effect, we will do something analogous to the model comparison methods we saw earlier. First, fit a model with only an intercept. Notice that the null deviance is 284 on 7 degrees of freedom.

```
null.glm<-glm(propn.dead~1,binomial(logit),
              weights=number,data=beetle)
```

Deviance for the binomial distribution

```
summary(null.glm)

##
## Call:
## glm(formula = propn.dead ~ 1, family = binomial(logit), data = beetle,
##     weights = number)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.11   -5.29    1.10    5.62    7.77
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4263     0.0933    4.57  4.9e-06
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 284.2  on 7  degrees of freedom
## Residual deviance: 284.2  on 7  degrees of freedom
## AIC: 312.4
##
## Number of Fisher Scoring iterations: 4
```

Deviance for the binomial distribution



Deviance for the binomial distribution

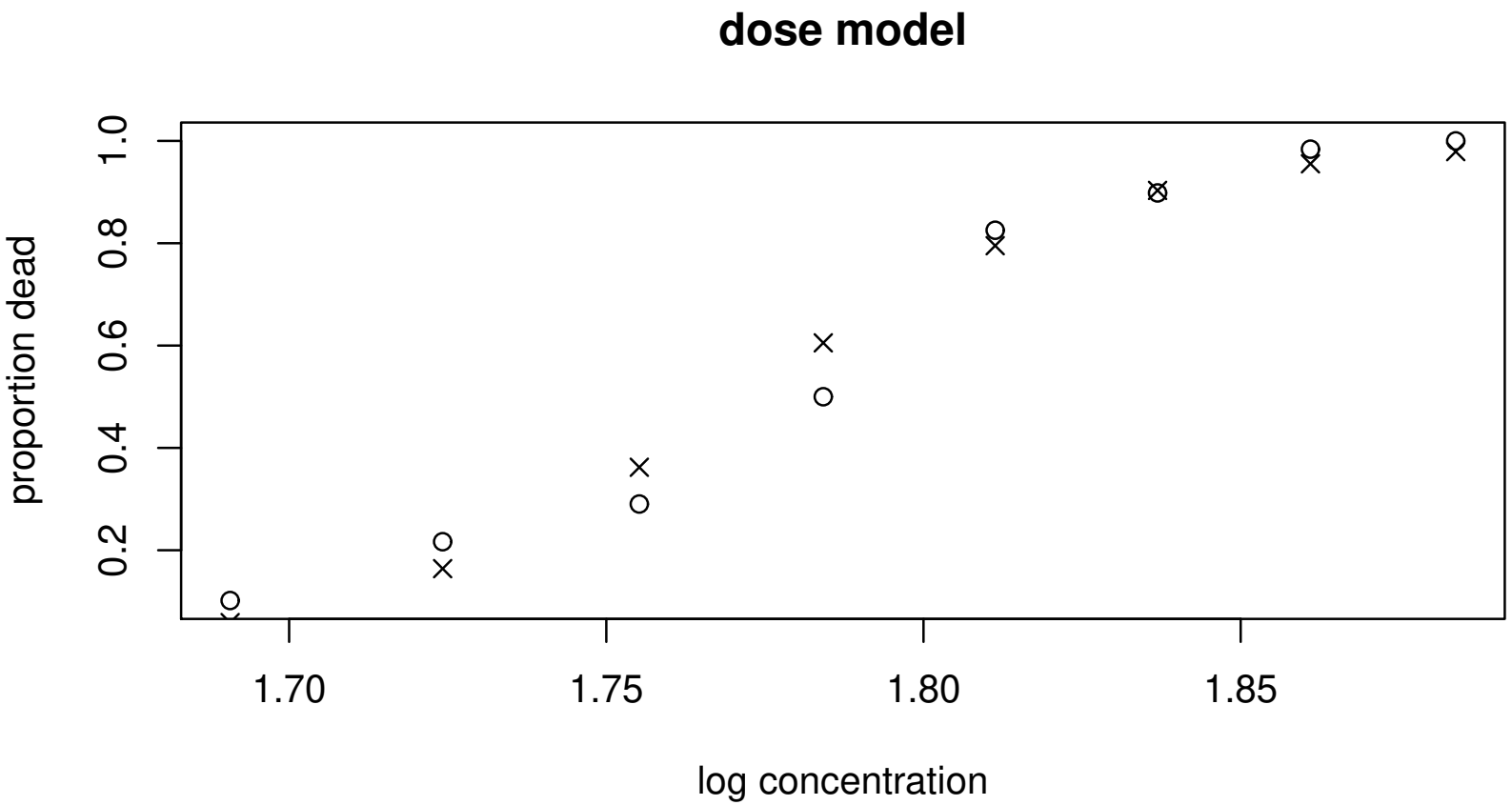
Add a term for dose. Now, the residual deviance is 11.2 on 6 dfs.

```
dose.glm<-glm(propn.dead~dose,binomial(logit),
              weights=number,data=beetle)
```

Deviance for the binomial distribution

```
summary(dose.glm)

##
## Call:
## glm(formula = propn.dead ~ dose, family = binomial(logit), data = beetle,
##     weights = number)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.594  -0.394   0.833   1.259   1.594
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -60.72       5.18   -11.7   <2e-16
## dose           34.27       2.91    11.8   <2e-16
##
## (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
```

Deviance for the binomial distribution



Deviance for the binomial distribution

The change in deviance from the null model is 284.2-11.2=273 on 1 df. Since the critical $\chi^2_1 = 3.84$, we reject the null hypothesis that $\beta_1 = 0$.
You can do the model comparison using the anova function. Note that no statistical test is calculated; you need to do that yourself.

```
anova(null.glm,dose.glm)

## Analysis of Deviance Table
##
## Model 1: propn.dead ~ 1
## Model 2: propn.dead ~ dose
##   Resid. Df Resid. Dev Df Deviance
## 1         7      284.2
## 2         6       11.2  1      273
```

Deviance for the binomial distribution

Actually, you don't even need to define the null model; the anova function automatically compares the fitted model to the null model:

```
anova(dose.glm)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: propn.dead
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                     7      284.2
## dose  1      273         6       11.2
```

Goodness of fit

The deviance for a given degrees of freedom v should have a χ^2_v distribution for the model to be adequate. As an example, consider the null model above. The deviance is clearly much larger than the 95th percentile cutoff point of the chi-squared distribution with 7 dfs, so the model is not adequate.

```
deviance(null.glm)

## [1] 284.2

## critical value:
qchisq(0.95,df=7)

## [1] 14.067
```

Goodness of fit

Now consider the model with dose as predictor. The deviance is less than the 95th percentile, so the fit is adequate.

```
deviance(dose.glm)

## [1] 11.232

qchisq(0.95,df=6)

## [1] 12.592
```

Generalized linear mixed models

The GLMM is now straightforward: we know the basic theory of linear mixed models already, and the syntax is very general.

```
## varying intercepts model:
library(lme4)

## Loading required package:  Matrix

fm_skip_lmer<-glmer(skip ~ word_complex+SC
                      +(1|subj)+(1|item),
                    family=binomial(),hindi)
```

Generalized linear mixed models

```
summary(fm_skip_lmer)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: skip ~ word_complex + SC + (1 | subj) + (1 | item)
##    Data: hindi
##
##      AIC      BIC   logLik deviance df.resid
##    88404    88450   -44197    88394    79938
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.802 -0.634 -0.467  0.916  6.747
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  item   (Intercept) 0.00935  0.0967
##  subj   (Intercept) 0.26252  0.5124
## Number of obs: 79943, groups:  item, 83; subj, 30
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.1409     0.0955   -1.48     0.14
## word_complex  -0.6862     0.0167  -41.09   <2e-16
## SC            -0.5398     0.0133  -40.49   <2e-16
##
## Correlation of Fixed Effects:
##             (Intr) wrd_cm
```

Generalized linear mixed models

Centering the predictors yields lower correlations of fixed effects:

```
fm_skip_lmer2<-glmer(skip ~ scale(word_complex,
                                  scale=FALSE)+
                        scale(SC,scale=FALSE)
                       +(1|subj)+(1|item),
                    family=binomial(),hindi)
```

Generalized linear mixed models

```
summary(fm_skip_lmer2)

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## skip ~ scale(word_complex, scale = FALSE) + scale(SC, scale = FALSE) +
##     (1 | subj) + (1 | item)
##    Data: hindi
##
##      AIC      BIC   logLik deviance df.resid
##    88404    88450   -44197    88394    79938
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.802 -0.634 -0.467  0.916  6.747
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  item   (Intercept) 0.00935  0.0967
##  subj   (Intercept) 0.26252  0.5124
## Number of obs: 79943, groups:  item, 83; subj, 30
##
## Fixed effects:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.0047     0.0945   -10.6   <2e-16
## scale(word_complex, scale = FALSE)  -0.6862     0.0167   -41.1   <2e-16
## scale(SC, scale = FALSE)            -0.5398     0.0133   -40.5   <2e-16
##
```