# Lecture 8: Multiple logistic regression

## Grouped and ungrouped data

We will simulate data via both methods from the same model. It will be a logistic regression model with two binary covariates.

```
n <- 1000 # number of subjects (independent Bernoulli observations)
set.seed(n)
x1 <- rbinom(n,1,.6) # prevalence of 0.6
x2 <- rbinom(n,1,plogis(.5*x1)) # to induce correlation between the
predictors
```

Now the outcome is generated, first using the ungrouped method. We will define an outcome with a prevalence of `0.25` among those with `x1=x2=0` and no interaction between the covariates. The covariate `x1` will have an odds ratio of `2` with the outcome, and `x2` will have an odds ratio of `1.5`; both of these will be the odds ratio controlling for the other covariate.

```
y.ungr <- rbinom(n,1,
                      plogis(qlogis(.25) + # prevalence of outcome for
x1=x2=0
                                 log(2)*x1 + # odds ratio for x1 controlling
for x2
                                 log(1.5)*x2 # odds ratio for x2 controlling
for x1
                                   ))
mod1.ungr <- glm(y.ungr ~ x1+x2,family=binomial)
summary(mod1.ungr)

##
## Call:
## glm(formula = y.ungr ~ x1 + x2, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1723  -0.9601  -0.7296   1.1825   1.7051
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1876     0.1361  -8.725  < 2e-16 ***
## x1            0.6524     0.1389   4.698 2.63e-06 ***
## x2            0.5231     0.1378   3.795 0.000147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1333.9  on 999  degrees of freedom
## Residual deviance: 1291.8  on 997  degrees of freedom
## AIC: 1297.8
##
## Number of Fisher Scoring iterations: 4
```

We can refit this model in grouped data format.

```
# covariate profiles
gr.yes <- aggregate(y.ungr ~ x1+x2, FUN=sum)
gr.no <- aggregate(I(1-y.ungr) ~ x1+x2, FUN=sum)
gr.data <- merge(gr.yes,gr.no)
colnames(gr.data)[3:4] <- c('y','n.minus.y')
mod2.ungr <- glm(cbind(y,n.minus.y)~x1+x2,family=binomial,data=gr.data)
summary(mod2.ungr)

##
## Call:
## glm(formula = cbind(y, n.minus.y) ~ x1 + x2, family = binomial,
##     data = gr.data)
##
## Deviance Residuals:
##       1        2        3        4
##  1.2768  -1.1326  -1.0960   0.7812
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1876     0.1361  -8.725  < 2e-16 ***
## x1            0.6524     0.1389   4.698 2.63e-06 ***
## x2            0.5231     0.1378   3.795 0.000147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46.7552  on 3  degrees of freedom
## Residual deviance:  4.7245  on 1  degrees of freedom
## AIC: 33.943
##
## Number of Fisher Scoring iterations: 3
```

The model estimates are the same, but values for the deviance and AIC are different.

Now we instead generate using the grouped format. This means that the model estimates will be different. Note that all we are changing is the outcome, not the covariates.

```
# use the same covariate values as in the ungrouped example
gr.data$n <- gr.data$y+gr.data$n.minus.y
# drop the ungrouped outcome data
gr.data <- gr.data[,-c(3,4)]
```

```
gr.data$y <- rbinom(4,gr.data$n,
               plogis(as.matrix(cbind(1,gr.data[,c('x1','x2')])) %*%
                        c(qlogis(.25),log(2),log(1.5))))
mod1.gr <- glm(cbind(y,n-y)~x1+x2,family=binomial,data=gr.data)
summary(mod1.gr)

##
## Call:
## glm(formula = cbind(y, n - y) ~ x1 + x2, family = binomial, data =
gr.data)
##
## Deviance Residuals:
##       1        2        3        4
## -0.6089   0.5230   0.5056  -0.3690
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1230     0.1342  -8.366  < 2e-16 ***
## x1            0.7015     0.1376   5.100  3.4e-07 ***
## x2            0.5163     0.1364   3.785 0.000154 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 47.369  on 3  degrees of freedom
## Residual deviance:  1.036  on 1  degrees of freedom
## AIC: 30.317
##
## Number of Fisher Scoring iterations: 3
```

## Deviance

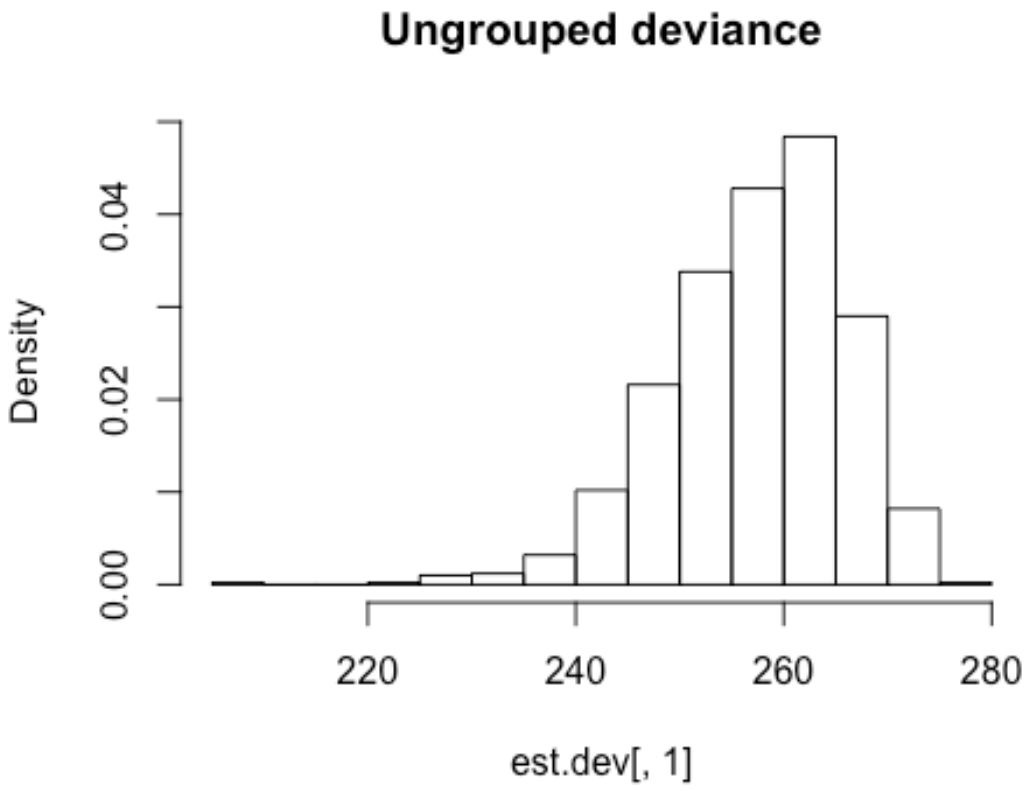
Now we want to use the same example data to look at the deviance statistic. First we show the saturated model fit.

```
# add the interaction term
mod2.gr <- glm(cbind(y,n-y)~x1*x2,family=binomial,data=gr.data)
summary(mod2.gr)

##
## Call:
## glm(formula = cbind(y, n - y) ~ x1 * x2, family = binomial, data =
gr.data)
##
## Deviance Residuals:
## [1]  0  0  0  0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

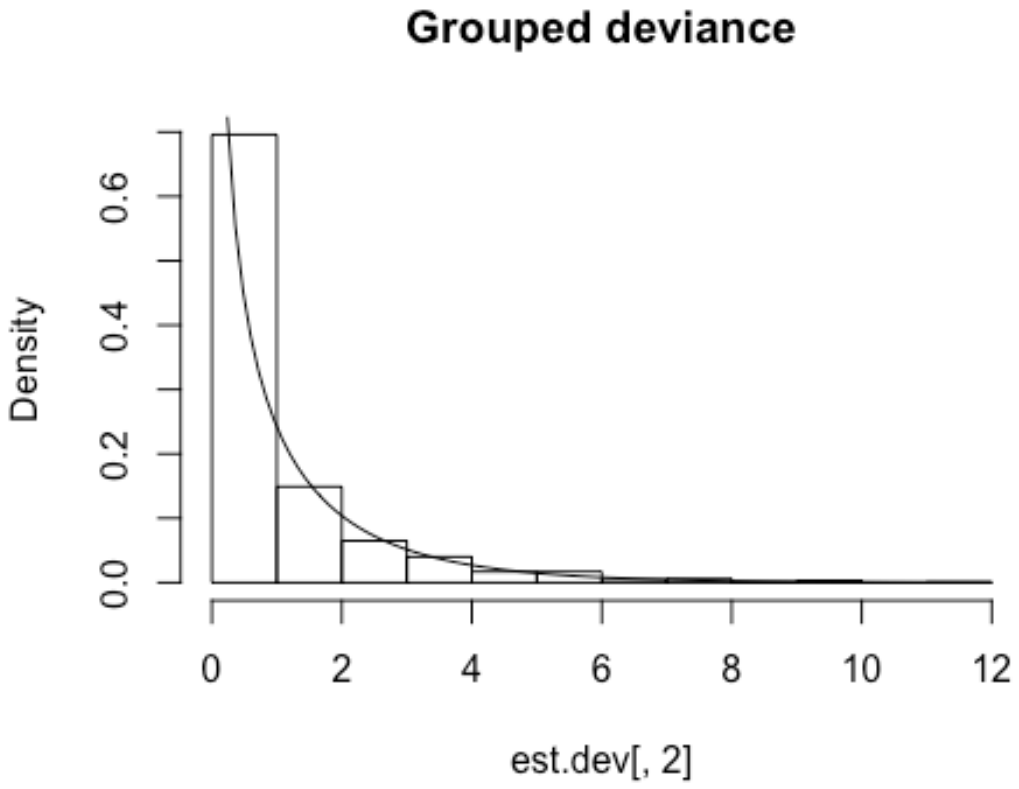
```
## (Intercept)  -1.2264     0.1714  -7.154 8.46e-13 ***
## x1            0.8755     0.2208   3.965 7.35e-05 ***
## x2            0.6951     0.2236   3.108  0.00188 **
## x1:x2        -0.2869     0.2824  -1.016  0.30971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.7369e+01  on 3  degrees of freedom
## Residual deviance: 6.2172e-14  on 0  degrees of freedom
## AIC: 31.281
##
## Number of Fisher Scoring iterations: 3
```

Note that the deviance residuals are all equal to 0; there are only 4 residuals because there are only 4 distinct covariate patterns in this data with two binary covariates.

```
trials <- 1000
n <- 200
# to record the deviance for each model
est.dev <- array(dim=c(trials,3))
for(tr in 1:trials) {
  set.seed(tr)
  # ungrouped
  x1 <- rbinom(n,1,.6)
  x2 <- rbinom(n,1,plogis(.5*x1))
  y.ungr <- rbinom(n,1,plogis(qlogis(.25) + log(2)*x1 + log(1.5)*x2))
  fit1 <- glm(y.ungr ~ x1+x2,family=binomial)
  est.dev[tr,1] <- summary(fit1)$deviance
  # grouped
  gr.yes <- aggregate(y.ungr ~ x1+x2, FUN=sum)
  gr.no <- aggregate(I(1-y.ungr) ~ x1+x2, FUN=sum)
  gr.data <- merge(gr.yes,gr.no)
  colnames(gr.data)[3:4] <- c('y','n.minus.y')
  fit2 <- glm(cbind(y,n.minus.y)~x1+x2,family=binomial,data=gr.data)
  est.dev[tr,2] <- summary(fit2)$deviance
  # excluding variable x1
  fit3 <- glm(cbind(y,n.minus.y)~x2,family=binomial,data=gr.data)
  est.dev[tr,3] <- summary(fit3)$deviance
}

par(mfrow=c(1,1))
hist(est.dev[,1],freq=FALSE,main='Ungrouped deviance')
```



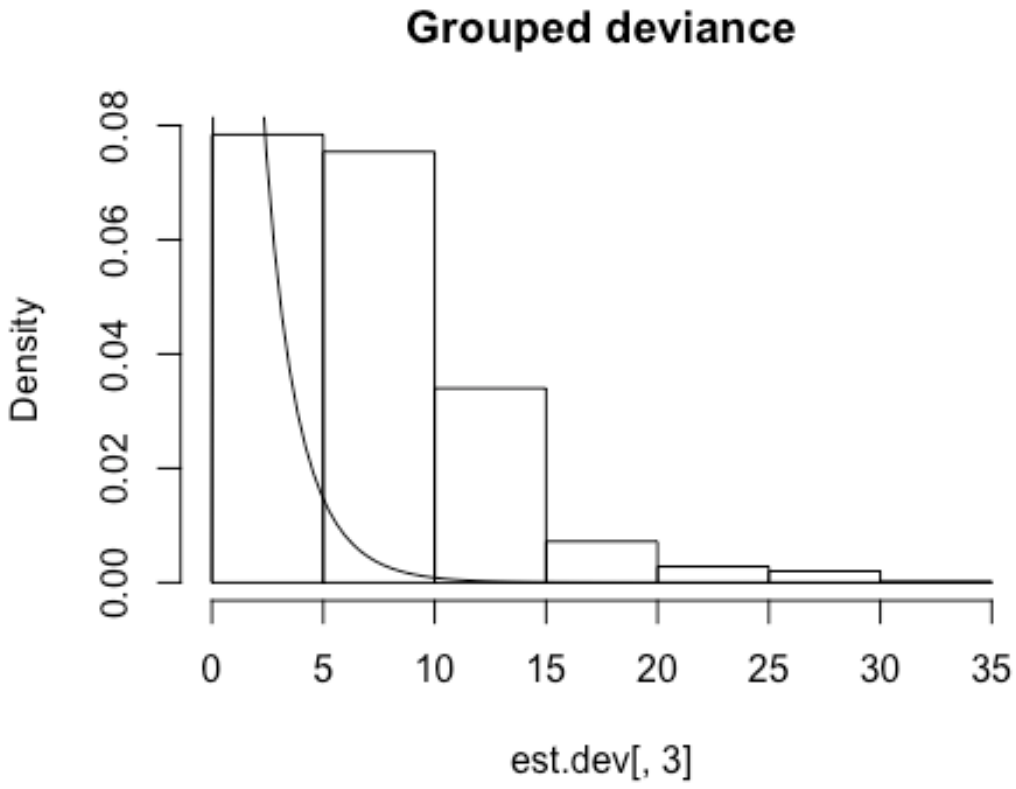

```r
# correct specification
hist(est.dev[,2],freq=FALSE,main='Grouped deviance')
curve(dchisq(x,1),add=TRUE)
```




```
ks.test(est.dev[,2],pchisq,1)

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  est.dev[, 2]
## D = 0.029024, p-value = 0.3686
## alternative hypothesis: two-sided
```

The chi-square distribution with 1 df provides a good approximation to the sampling distribution of the grouped deviance under the null hypothesis of correct model specification. This is not the case for the ungrouped deviance.

```
hist(est.dev[,3],freq=FALSE,main='Grouped deviance')
curve(dchisq(x,1),add=TRUE)
```




```
# power at alpha=.05
mean(pchisq(est.dev[,3],1,lower.tail=FALSE)<.05)

## [1] 0.707
```

If we look at the deviance for the model excluding x1, then we see that the chi-square curve does not fit well. This is not because the test does not apply (as in the case of the ungrouped deviance), but rather because the null hypothesis is false. This is why we can think about power in this situation.

## Budworm example

We will follow the example from class.

```
# drop first variable (observation number)
budworm.df <- read.table('budworm_data.txt',header=TRUE)[,-1]
# sex = the sex of the budworm
# dose = amount of cypermethrin exposed to
# s = number of budworms affected
# n = total number of budworms
budworm.df

##    sex dose  s  n
## 1    0    1  1 20
```

```
## 2    0    2  4 20
## 3    0    4  9 20
## 4    0    8 13 20
## 5    0   16 18 20
## 6    0   32 20 20
## 7    1    1  0 20
## 8    1    2  2 20
## 9    1    4  6 20
## 10   1    8 10 20
## 11   1   16 12 20
## 12   1   32 16 20
```

```r
# saturated model
# dose enters as a categorical covariate
# all interactions possible included
budworm.max <- glm(cbind(s, n-s) ~
                     sex*as.factor(dose),
                   family=binomial,
                   data=budworm.df)
# model of interest
# dose enters as a linear term
budworm.mod <- glm(cbind(s, n-s) ~
                     sex + dose,
                   family=binomial,
                   data=budworm.df)
# null model
budworm.null <- glm(cbind(s, n-s) ~
                     1,
                   family=binomial,
                   data=budworm.df)

cbind(budworm.max$fitted.values,
      budworm.mod$fitted.values,
      budworm.null$fitted.values)
```

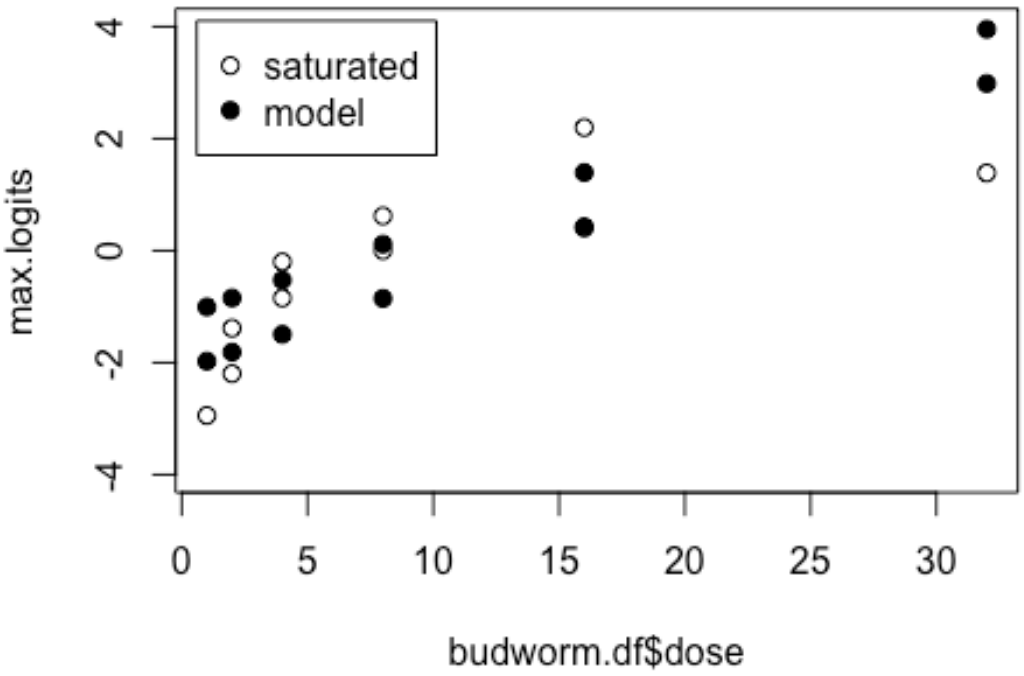
```
##            [,1]      [,2]   [,3]
## 1  5.000000e-02 0.2677414 0.4625
## 2  2.000000e-01 0.3002398 0.4625
## 3  4.500000e-01 0.3713931 0.4625
## 4  6.500000e-01 0.5283639 0.4625
## 5  9.000000e-01 0.8011063 0.4625
## 6  1.000000e+00 0.9811556 0.4625
## 7  6.548641e-12 0.1218892 0.4625
## 8  1.000000e-01 0.1400705 0.4625
## 9  3.000000e-01 0.1832034 0.4625
## 10 5.000000e-01 0.2983912 0.4625
## 11 6.000000e-01 0.6046013 0.4625
## 12 8.000000e-01 0.9518445 0.4625
```

The fitted values for the maximal model correspond to the sample proportions within each covariate pattern, while those for the null model are the same for all covariate patterns and are just equal to the overall proportion of dead worms.

```
max.logits <- predict(budworm.max,type='link')
mod.logits <- predict(budworm.mod,type='link')

cbind(max.logits,mod.logits)
```

```
##       max.logits mod.logits
## 1  -2.944439e+00 -1.0061121
## 2  -1.386294e+00 -0.8461564
## 3  -2.006707e-01 -0.5262451
## 4   6.190392e-01  0.1135776
## 5   2.197225e+00  1.3932230
## 6   2.575176e+01  3.9525137
## 7  -2.575176e+01 -1.9746604
## 8  -2.197225e+00 -1.8147047
## 9  -8.472979e-01 -1.4947934
## 10 -7.105427e-15 -0.8549707
## 11  4.054651e-01  0.4246747
## 12  1.386294e+00  2.9839654
```

```
plot(budworm.df$dose,max.logits,
     ylim=c(-4,4))
points(budworm.df$dose,mod.logits,pch=19)
legend(x='topleft',inset=.025,pch=c(21,19),legend=c('saturated','model'))
```




Likelihood calculations: we will go through the example of how to calculate the log-likelihood for saturated model.

```
# likelihoods
# (this scale is problematic because of possible numerical underflow)
exp(logLik(budworm.max))

## 'log Lik.' 2.89474e-07 (df=12)

exp(logLik(budworm.mod))

## 'log Lik.' 2.445915e-13 (df=3)

exp(logLik(budworm.null))

## 'log Lik.' 2.214212e-34 (df=1)

budworm.df$n.minus.s <- budworm.df$n - budworm.df$s

# this term is the difference between the grouped and ungrouped log-
likelihoods
combn.term <- lchoose(budworm.df$n,budworm.df$s)
# estimated p.hat
# equal to sample proportions at each level for saturated model
```

```r
prob.hat <- budworm.df$s/budworm.df$n
# grouped log-likelihood
# need to define a function to deal with 0*log(0)
xlogy <- function(x,y) ifelse(x>0,x*log(y),0)
# grouped log-likelihood
sum(combn.term +
      xlogy(budworm.df$s,prob.hat) +
      xlogy(budworm.df$n.minus.s,1-prob.hat))
```

```
## [1] -15.0552
```

```r
# from glm() outcome
logLik(budworm.max)
```

```
## 'log Lik.' -15.0552 (df=12)
```

```r
# UNgrouped
sum(xlogy(budworm.df$s,prob.hat) +
      xlogy(budworm.df$n.minus.s,1-prob.hat))
```

```
## [1] -103.2419
```

Compare the null model to saturated model using a likelihood ratio test.

```r
# test statistic
lrstat <- -2*(logLik(budworm.null)-logLik(budworm.max))
lrstat
```

```
## 'log Lik.' 124.8756 (df=1)
```

```r
# degrees of freedom
lrdf <- budworm.null$df.residual-budworm.max$df.residual
lrdf
```

```
## [1] 11
```

```r
pchisq(lrstat,lrdf,lower.tail=FALSE)
```

```
## 'log Lik.' 1.888822e-21 (df=1)
```

```r
# or using anova()
anova(budworm.null,budworm.max,test='LRT')
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(s, n - s) ~ 1
## Model 2: cbind(s, n - s) ~ sex * as.factor(dose)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        11     124.88
## 2         0       0.00 11   124.88 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Look at goodness of fit of model of interest using deviance chi-square test.

```
# test statistic
devstat <- summary(budworm.mod)$deviance
devstat
```

```
## [1] 27.96797
```

```
# degrees of freedom
devdf <- summary(budworm.mod)$df.resid
devdf
```

```
## [1] 9
```

```
pchisq(devstat,devdf,lower.tail=FALSE)
```

```
## [1] 0.0009656918
```

Bad fit: could guess this from plot earlier, but the test gives a much clearer answer.

```
budworm.logdose <- glm(cbind(s, n-s) ~
                         sex + I(log(dose)),
                       family=binomial,
                       data=budworm.df)
summary(budworm.logdose)
```

```
##
## Call:
## glm(formula = cbind(s, n - s) ~ sex + I(log(dose)), family = binomial,
##     data = budworm.df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.10540  -0.65343  -0.02225   0.48471   1.42944
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.3724     0.3855  -6.154 7.56e-10 ***
## sex           -1.1007     0.3558  -3.093  0.00198 **
## I(log(dose))   1.5353     0.1891   8.119 4.70e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 124.8756  on 11  degrees of freedom
## Residual deviance:   6.7571  on  9  degrees of freedom
## AIC: 42.867
##
## Number of Fisher Scoring iterations: 4
```

```
pchisq(summary(budworm.logdose)$deviance,
       summary(budworm.logdose)$df.resid,
       lower.tail=FALSE)

## [1] 0.6623957
```

So `log(dose)` works much better: no evidence of bad fit here.

Now look at ungrouped data.

```
# create ungrouped data set
budworm.yes <-
data.frame(budworm.df[rep(1:nrow(budworm.df),budworm.df$s),1:2],y=1)
budworm.no <-
data.frame(budworm.df[rep(1:nrow(budworm.df),budworm.df$n.minus.s),1:2],y=0)
# model of interest
summary(glm(y~sex+dose,family=binomial,data=rbind(budworm.yes,budworm.no)))

##
## Call:
## glm(formula = y ~ sex + dose, family = binomial, data = rbind(budworm.yes,
##     budworm.no))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4630  -0.7895  -0.5099   0.6660   1.9827
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.16607    0.26155  -4.458 8.26e-06 ***
## sex         -0.96855    0.32954  -2.939  0.00329 **
## dose         0.15996    0.02341   6.832 8.39e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 331.36  on 239  degrees of freedom
## Residual deviance: 234.45  on 237  degrees of freedom
## AIC: 240.45
##
## Number of Fisher Scoring iterations: 5

# intercept-only (null) model
# log-likelihood from glm()
budworm.ungrmod <-
glm(y~sex*as.factor(dose),family=binomial,data=rbind(budworm.yes,budworm.no))
logLik(budworm.ungrmod)

## 'log Lik.' -103.2419 (df=12)
```

```r
# log-likelihood we calculated above
sum(xlogy(budworm.df$s,prob.hat) +
      xlogy(budworm.df$n.minus.s,1-prob.hat))

## [1] -103.2419
```

The log-likelihood of the model fitted to the ungrouped data is the same as the log likelihood for the grouped data model plus the combinatoric term:

```r
logLik(budworm.max)

## 'log Lik.' -15.0552 (df=12)

logLik(budworm.ungrmod)

## 'log Lik.' -103.2419 (df=12)

logLik(budworm.ungrmod)+sum(combn.term)

## 'log Lik.' -15.0552 (df=12)
```