Book 2, Chapter 2: Binomial Data

Outline

1. Binomial data
2. Generalized linear models (glm) for binomial data
3. Inference for glm for binomial data
4. Odds ratio
5. Overdispersion

Review: The Binomial Distribution

- n independent trials $Z_1, \ldots, Z_n$
- $P(Z_i = 1) = p$ ("success")
  $P(Z_i = 0) = 1 - p$ ("failure")
- The binomial variable $Y = \sum_{i=1}^{n} Z_i$ is the total number of successes out of n iid trials
- Probability distribution function is given by

$$P(Y = k) = \binom{n}{k} p^k (1-p)^{n-k} \text{ for } k = 0, 1, \ldots, n$$

Review: The Binomial Distribution

- $E(Y) = np$
- $Var(Y) = np(1-p)$
- As $n \to \infty$, Binomial $\to$ Normal:

$$\frac{Y - np}{\sqrt{np(1-p)}} \to N(0,1)$$

- Sample proportion (estimate of p)

$$\hat{p} = \frac{Y}{n}$$

Binomial Data

- Response y_i: number of successes out of n_i independent trials with probability of success p_i
- $x = (x_1, x_2, \ldots, x_p)$: predictors (quantitative, factors, or both)
- For all trials contributing to one response y_i, the predictors x_i have the same value (*covariate class*)
- Goal: model the relationship between y and $x_1, \ldots, x_p$ via modeling the relationship between p_i and $x_1, \ldots, x_p$.

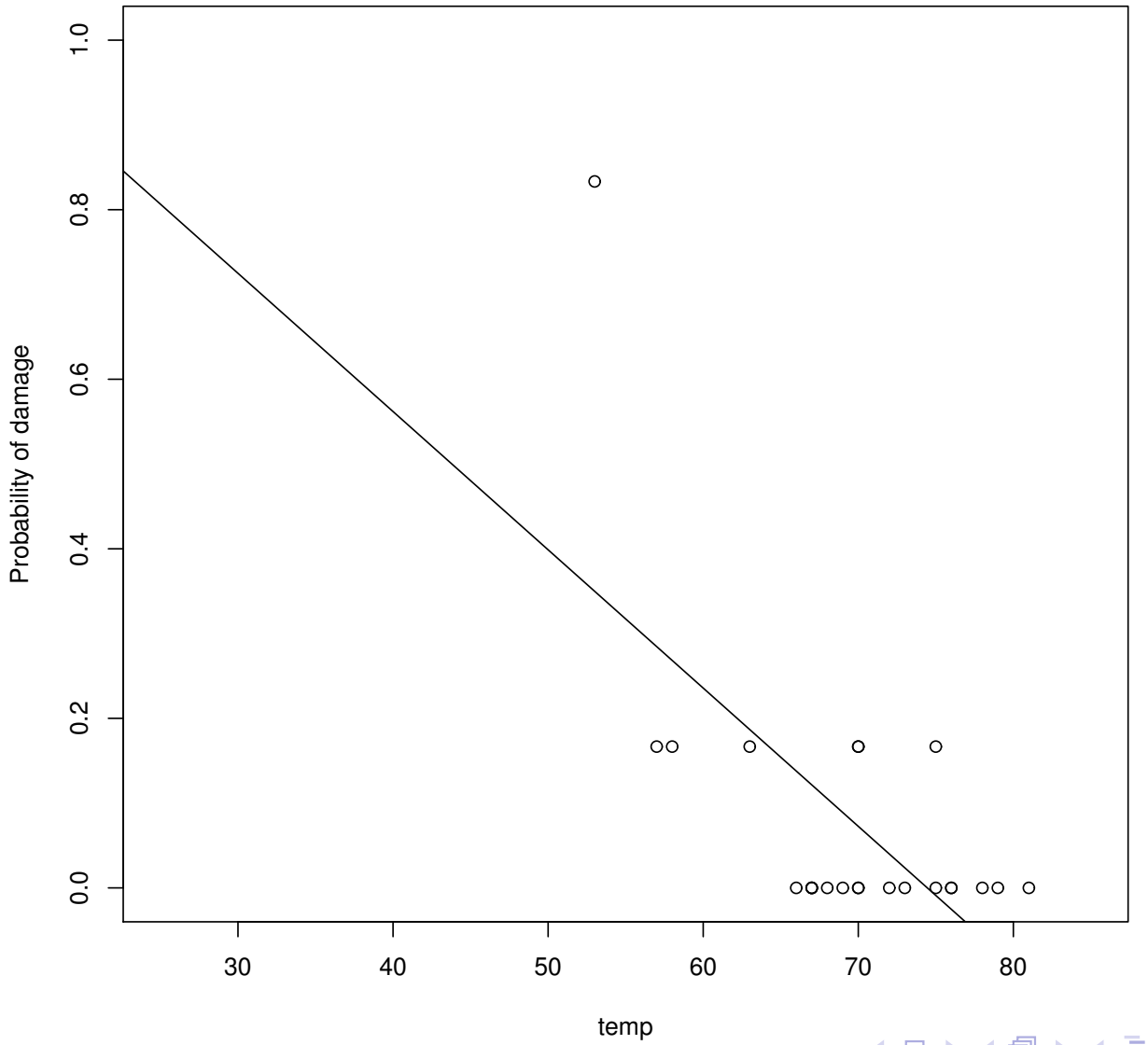
Challenger Disaster Example

- The space shuttle Challenger exploded after launch in 1986
- One explanation: rubber seals called O-rings
- Rubber gets brittle at cold temperatures and becomes less effective as a sealant, and it was an unusually cold day (31F)
- Have data on damage to O-rings (how many showed evidence of damage out of 6 total) and temperature from previous launches

```
## Load the data
> library(faraway)
> data(orings)
## Fit a linear model to observed proportions
> plot(damage/6 ~ temp, orings, xlim=c(25,85),
+   ylim = c(0,1),ylab="Probability of damage")
> abline(lm(damage/6 ~ temp, orings))
```

The linear model is clearly inappropriate here.

Challenger Disaster Data



Binomial Regression

- Assume that y_i is Binomial(n_i, p_i)
- Assume all y_i's are independent
- Linear predictor:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

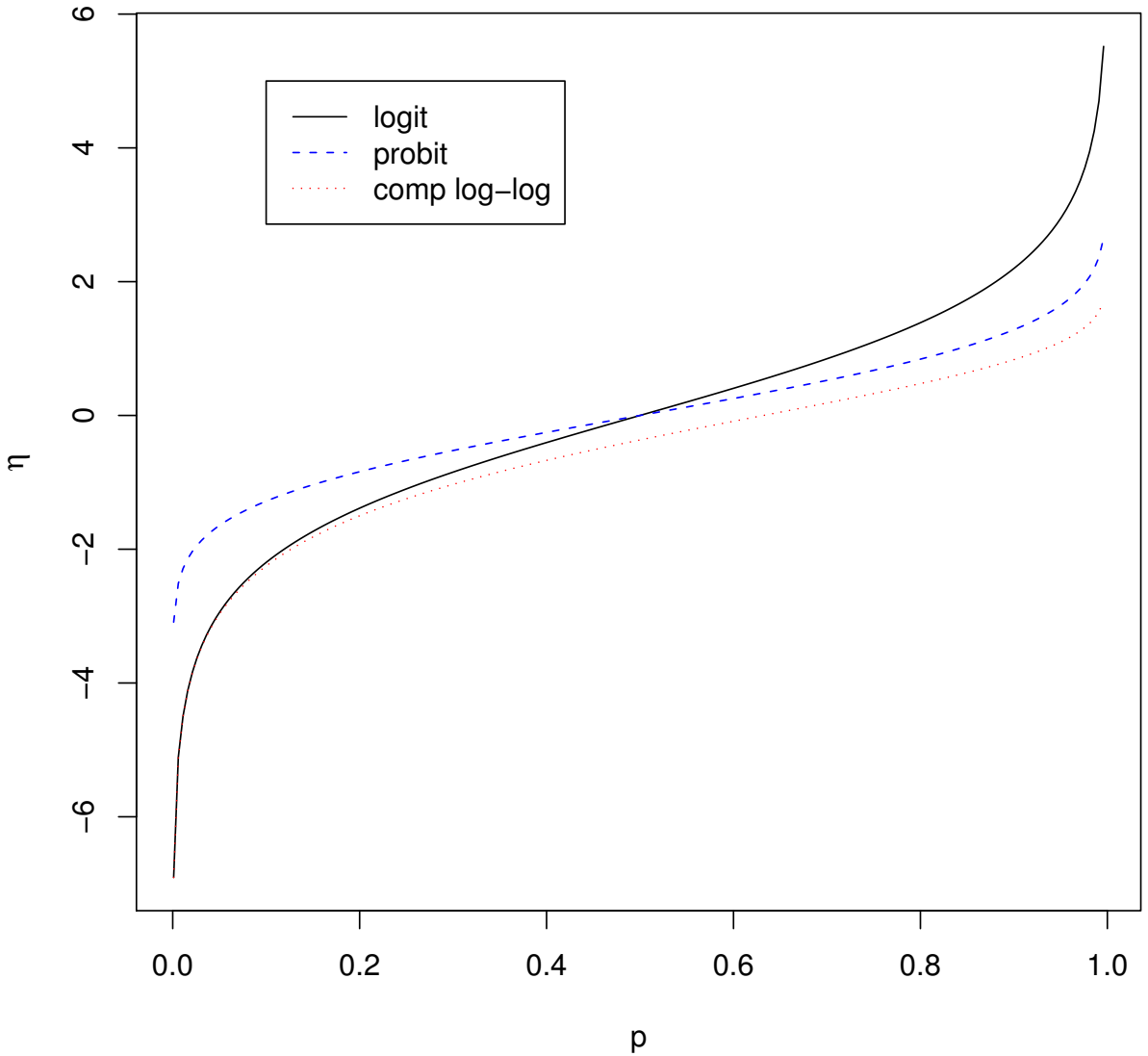
- Cannot use $\eta_i = p_i$ (need $0 \leq p \leq 1$)
- Main idea: use a **link function**

$$\eta_i = g(p_i)$$

Binomial link functions

- **Logit**: $\eta = \log(p/(1-p))$
- Probit: $\eta = \Phi^{-1}(p)$, where Φ is the cumulative distribution function of $N(0,1)$
- Complementary log-log: $\eta = \log(-\log(1-p))$
- All transform $p \in (0,1)$ to $\eta \in (-\infty, \infty)$

Binomial link functions



Estimating parameters

- Maximum likelihood approach: find parameters (in this case p_i) that maximize the likelihood of the data,

$$\prod_{i=1}^{n} P(Y_i = y_i)$$

  where Y_i is Binomial(n_i, p_i).
- Log-likelihood is given by

$$\ell(p_1, \ldots, p_n; y) = \sum_{i=1}^{n} \left[\log \binom{n_i}{y_i} + y_i \log p_i + (n_i - y_i) \log(1 - p_i) \right]$$

- For the logit link, $p_i = e^{X_i\beta}/(1 + e^{X_i\beta})$

Estimating parameters

- Need to maximize:

$$\ell(\beta) = \sum_{i=1}^{n} \left[y_i (x_i^T \beta) - n_i \log(1 + \exp(x_i^T \beta)) \right]$$

- Optimization algorithm is complicated (Ch. 6)

Challenger Example

```
> logitm = glm(cbind(damage,6-damage) ~ temp,
+              family=binomial(link=logit), data=orings)
> summary(logitm)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9529  -0.7345  -0.4393  -0.2079   1.9565
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 11.66299    3.29626   3.538 0.000403 ***
temp        -0.21623    0.05318  -4.066 4.78e-05 ***
---
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 16.912  on 21  degrees of freedom
AIC: 33.675
Number of Fisher Scoring iterations: 6
```

Challenger Example

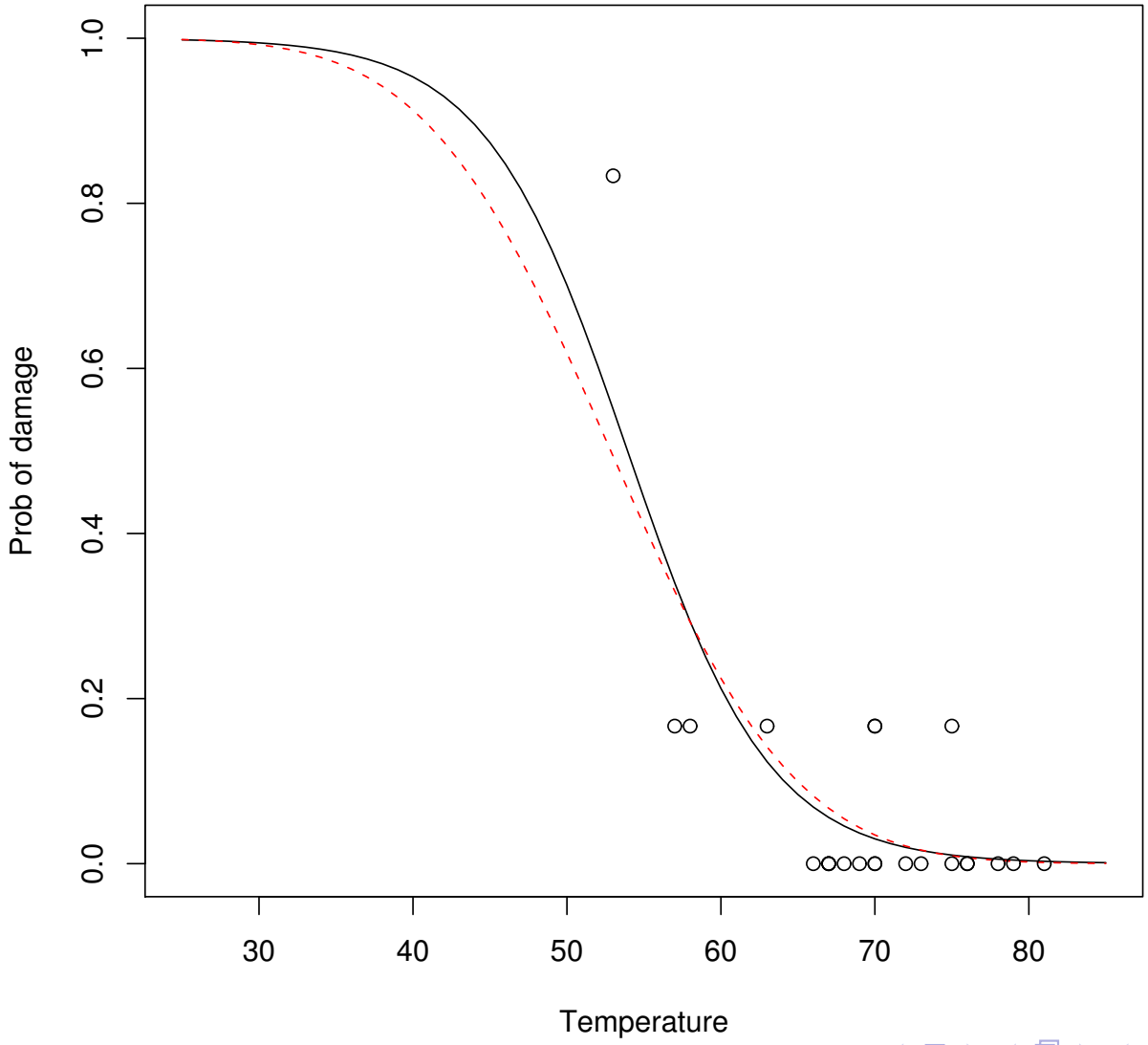
```
## estimate probability at temp = 31F
> test = data.frame(temp=31)
> ilogit(predict(logitm,test))
[1] 0.9930342
```

```
## fit a probit model to compare
> probitm = glm(cbind(damage,6-damage) ~ temp,
+                 family=binomial(link=probit), data=orings)
> summary(probitm)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0134  -0.7760  -0.4467  -0.1581   1.9982
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.59145    1.71055   3.269  0.00108 **
temp        -0.10580    0.02656  -3.984 6.79e-05 ***
---
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 38.898  on 22  degrees of freedom
Residual deviance: 18.131  on 21  degrees of freedom
AIC: 34.893
Number of Fisher Scoring iterations: 6
```

```
## Probit prediction at temp = 31F
> pnorm(predict(probitm,test))
[1] 0.9895983

# Make predictions for the whole range and plot
> range = data.frame(temp=seq(25,85,by=1))
> pred.l = ilogit(predict(logitm, range))
> pred.p = pnorm(predict(probitm, range))
> matplot(range, cbind(pred.l,pred.p), xlim=c(25,85),
+  ylim=c(0,1), xlab="Temperature", ylab="Prob of damage",
+  type ='ll',lty=c('solid','dashed'))
```

Logit and probit fits for Challenger data



Inference

Likelihood ratio test:

- two nested models
- L is the larger model with l parameters and likelihood L_L
- S is the smaller model with $s < l$ parameters and likelihood L_S
- The likelihood ratio statistic is

$$2 \log \frac{L_L}{L_S}$$

Deviance

- Take larger model to be the saturated model: n parameters to fit each data point perfectly, with fitted values $\hat{p}_i = y_i/n_i$.
- In this case, the test statistic is called the deviance of S and is given by

$$D = 2\sum_{i=1}^{n}\left[y_i \log\frac{y_i}{\hat{y}_i} + (n_i - y_i)\log\frac{n_i - y_i}{n_i - \hat{y}_i}\right]$$

  where $\hat{y}_i = n_i\hat{p}_i$, $\hat{p}_i$ are the fitted probabilities from S.
- If Y_i's are truly binomial, independent, n_i are large

$$D \approx \chi^2_{n-s}$$

Uses of deviance

- Test the goodness-of-fit:

$$p\text{-value} = P(\chi^2_{n-s} > D)$$

  Small deviance = good fit.

- Compare two nested models, e.g., null (no predictors) and current model. In this case, use

$$2 \log \frac{L_L}{L_S} = D_S - D_L \approx \chi^2_{(n-s)-(n-l)}$$

  and the p-value $= P(\chi^2_{l-s} > D_S - D_L)$

- Note: if $n_i = 1$ (binary data), deviance cannot be used

Other measures of goodness of fit

- The χ^2 goodness-of-fit statistic (Pearson's X^2):

$$X^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

  O_i is observed count in each "bin"
  E_i is expected count under the model tested

- For binomial data, add successes and failures to get

$$X^2 = \sum_{i=1}^{n} \frac{(y_i - n_i\hat{p}_i)^2}{n_i\hat{p}_i(1-\hat{p}_i)}$$

- Pearson residuals:

$$r_i^P = \frac{y_i - \hat{y}_i}{\sqrt{var(\hat{y}_i)}}$$

  Then $X^2 = \sum_{i=1}^{n}(r_i^P)^2$.

- Typically X^2 is close to deviance and is used in the same way.

```
## Goodness of fit for the Challenger data
## Deviance test
> pchisq(logitm$dev, df=logitm$df.resid,
+        lower.tail=F)
[1] 0.7164099

## Compare null to model with temperature
> pchisq(logitm$null.dev - logitm$dev,
+  df=logitm$df.null - logitm$df.resid, lower.tail=F)
[1] 2.747351e-06
```

```
## Pearson's chi-squared
> X2 = sum(residuals(logitm,type="pearson")^2)
[1] 28.06738
> pchisq(X2, df=logitm$df.resid, lower=F)
[1] 0.1382507
```

Confidence Intervals for Parameters

- Asymptotically $\hat{\beta}$ is normal – can use z-intervals
- Profile likelihood confidence intervals are more accurate (based on considering the likelihood of one parameter with all others fixed)

```
> library(MASS)
> confint(logitm)
Waiting for profiling to be done...
                2.5 %    97.5 %
(Intercept)  5.575195 18.737598
temp        -0.332657 -0.120179
```

Confidence Intervals for Predictions

- Predict probability of success $p(x_0)$ for a particular x_0
- No distinction between future observation and mean response
- Based on asymptotic normality of $\hat{\beta}$ and $x_0^T \hat{\beta}$
- Extrapolation will give unreliable predictions (as always)

```
> predict(logitm, test, se=T)
$fit
       1
4.959746
$se.fit
[1] 1.66731

> ilogit(c(4.96-1.96*1.67,4.96+1.96*1.67))
[1] 0.8438029 0.9997344
```

Interpreting Odds

- Odds: $\frac{p}{1-p}$
- Logistic regression (logit link) models log odds:

$$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- Interpretation: a unit increase in x_1 with all other predictors held fixed leads to an increase of β_1 in log-odds, or equivalently, odds being multiplied by $\exp(\beta_1)$.
- No such interpretation available for other link functions

What does a large deviance indicate?

- Violation of model assumptions (outliers, non-linearity, model structure)
- Sparse data (small n_i)
- Overdispersion

Overdispersion

- The binomial model links the mean and the variance:

$$\text{Var}(y_i) = n_i \hat{p}_i (1 - \hat{p}_i)$$

  (not the case for normal data)
- Overdispersion: observed $\text{Var}(y_i)$ is greater than the model postulates
- Common causes of overdispersion:
  - The trials are not independent
  - The probability of success is not constant
- Underdispersion is also possible but rare in practice

Estimating Overdispersion

- Introduce an additional dispersion parameter $\phi = \sigma^2$, so that $var(y_i) = \sigma^2 n_i p_i (1 - p_i)$
- Can estimate σ^2 (as in linear regression) as

$$\hat{\sigma}^2 = \frac{X^2}{n - (p + 1)}$$

- This does not affect $\hat{\beta}$
- All standard errors must be multiplied by $\hat{\sigma}$

- Deviance can no longer be used to compare models
- An approximate F-test can be used:

$$F = \frac{(D_S - D_L)/(df_S - df_L)}{\hat{\sigma}^2}$$

  has the F distribution with $df_S - df_L$ and $n - (p + 1)$ degrees of freedom
- Goodness of fit cannot be tested
- Estimating overdispersion is only reasonable when n_i's are roughly equal

Overdispersion example: trout data

- Boxes of trout eggs buried in a stream and retrieved after some time
- Five different locations (`location`), four lag times in weeks (`period`)
- Number of surviving eggs (`survive`), total in box (`total`)

```
> tmod = glm(cbind(survive, total-survive) ~ location +
+    period, family = binomial, data = troutegg)
> summary(tmod)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.6358     0.2813  16.479  < 2e-16 ***
location2    -0.4168     0.2461  -1.694   0.0903 .
location3    -1.2421     0.2194  -5.660 1.51e-08 ***
location4    -0.9509     0.2288  -4.157 3.23e-05 ***
location5    -4.6138     0.2502 -18.439  < 2e-16 ***
period7      -2.1702     0.2384  -9.103  < 2e-16 ***
period8      -2.3256     0.2429  -9.573  < 2e-16 ***
period11     -2.4500     0.2341 -10.466  < 2e-16 ***
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1021.469  on 19  degrees of freedom
Residual deviance:   64.495  on 12  degrees of freedom
AIC: 157.03
```

```
##  estimate sigma2
> sigma2 = sum(residuals(tmod,type="pearson")^2)/12
> sigma2
[1] 5.330322
```

```
> drop1(tmod, scale=sigma2, test="F")
Single term deletions
Model:
cbind(survive, total - survive) ~ location + period

scale:  5.330322
         Df Deviance    AIC F value     Pr(F)
<none>         64.50 157.03
location  4   913.56 308.32  39.494 8.142e-07 ***
period    3   228.57 181.81  10.176  0.001288 **
---
Warning message:
In drop1.glm(tmod, scale = sigma2, test = "F") :
  F test assumes 'quasibinomial' family
```

```
## use estimated dispersion to recompute p-values
> summary(tmod, dispersion=sigma2)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.6358     0.6495   7.138 9.49e-13 ***
location2    -0.4168     0.5682  -0.734   0.4632
location3    -1.2421     0.5066  -2.452   0.0142 *
location4    -0.9509     0.5281  -1.800   0.0718 .
location5    -4.6138     0.5777  -7.987 1.39e-15 ***
period7      -2.1702     0.5504  -3.943 8.05e-05 ***
period8      -2.3256     0.5609  -4.146 3.38e-05 ***
period11     -2.4500     0.5405  -4.533 5.82e-06 ***
---
(Dispersion parameter for binomial family taken to be 5.330
```

Summary

- With link functions, binomial data can be modeled easily
- Approximate inference available for testing models and parameter values
- Logit has advantages in interpretation

Warnings

- The estimation algorithm may not converge
- With small n_i, the χ^2 approximation is poor
- Overdispersion can be accounted for, but binomial assumption is sacrificed

Book 2, Chapter 3: Count Regression

Chapter Outline

- Review of the Poisson distribution
- Poisson regression
- Inference via deviance
- Overdispersion
- Example: Galapagos data

Review: The Poisson Distribution

- A random variable Y takes values $0, 1, 2, \ldots$
- The Poisson distribution has one parameter $\mu > 0$
- Probability distribution function is given by

$$P(Y = y) = \frac{e^{-\mu}\mu^y}{y!}$$

- $E(Y) = \mu = Var(Y)$

Review: The Poisson Distribution

- Can be used to approximate Binomial(n, p) if $n \to \infty$, $p \to 0$, $np \to \mu > 0$.
- If μ is large, Y is approximately normal
- If Y_i's are Poisson(μ_i), independent, then

$$\sum_i Y_i \text{ is Poisson}(\mu),$$

where $\mu = \sum_i \mu_i$.

The Poisson Process

- Events occur over time
- The number of events in time interval of length t has the Poisson distribution with $\mu = \lambda t$
- Non-overlapping time intervals are independent
- λ is called the rate of the process
- Waiting times between events are independent and exponentially distributed
- Used to model calls/customers in a service center, airplane arrivals, earthquakes, particle emissions, etc
- In practice λ is constant only for a limited time

Modeling count data

Response y_i is a count – should I assume it's Poisson?

- If the count is bounded above, binomial may be more appropriate
- If the counts are large, normal approximation applies and regular linear regression may be used
- If the count arises as the number of "failures" until a given number of "successes", then negative binomial is appropriate (also in Ch. 3, won't cover)

Poisson regression

- Assume counts y_i are independent, Poisson with mean μ_i
- $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$: predictors (quantitative, factors, or both)
- Goal: model the relationship between y and $x_1, \ldots, x_p$ via modeling the relationship $\mu_i = \mu(x_i)$.

- Linear predictor:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

- Cannot use $\eta_i = \mu_i$ (need $\mu_i > 0$)
- Main idea: use a **link function**

$$\eta_i = g(\mu_i)$$

- Canonical Poisson link function:

$$\eta_i = \log \mu_i$$

Estimating parameters

- Maximum likelihood: find parameters that maximize the log-likelihood of the data
- Log-likelihood is given by

$$\ell(\mu_1, \ldots, \mu_n; y) = \sum_{i=1}^{n} \left[-\mu_i + y_i \log \mu_i - \log(y_i!) \right]$$

- With respect to β,

$$\ell(\beta) = \sum_{i=1}^{n} \left[-\exp(x_i^T \beta) + y_i (x_i^T \beta) \right]$$

- Same optimization algorithm as for binomial (Iteratively Reweighted Least Squares, see Ch. 6)

Deviance

- Recall deviance is the likelihood ratio statistics comparing to the saturated model
- Deviance for the Poisson regression:

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

- $D \approx \chi^2$ can be used to test the goodness of fit or compare nested models as before
- For goodness of fit, can also use Pearson's X^2 statistic,

$$X^2 = \sum_{i=1}^{n}\frac{(y_i - \hat{\mu}_i)^2}{\hat{\mu}_i}$$

- Confidence intervals computed as before – either via asymptotic normality or profile likelihood

Galapagos Example

- Recall from Ch. 2 – modeling the number of species of tortoise
- y: number of species of tortoise
- $x_1, \ldots, x_5$: area of the island, highest elevation of the island, distance from the nearest island, distance from Santa Cruz Island, area of the adjacent island
- There is a number of fairly small counts so the normal assumption may not be accurate

Galapagos Example

```
> library(faraway)
> data(gala)
> ## Remove the endemics variable
> gala = gala[,-2]
> ## Fit Poisson regression
> galap = glm(Species ~ . , family=poisson, data=gala)
> summary(galap)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
```

Overdispersion

- Overdispersion: the model implies $var(y_i) = \hat{\mu}_i$ but in reality the variance is greater
- Introduce an additional dispersion parameter $\phi = \sigma^2$, so that $var(y_i) = \sigma^2 \mu_i$
- As before, estimate σ^2 as

$$\hat{\sigma}^2 = \frac{X^2}{n - (p+1)}$$

- This does not affect $\hat{\beta}$, but all standard errors must be multiplied by $\hat{\sigma}$
- An approximate F-test should be used to compare models with overdispersion

$$F = \frac{(D_S - D_L)/(df_S - df_L)}{\hat{\sigma}^2}$$

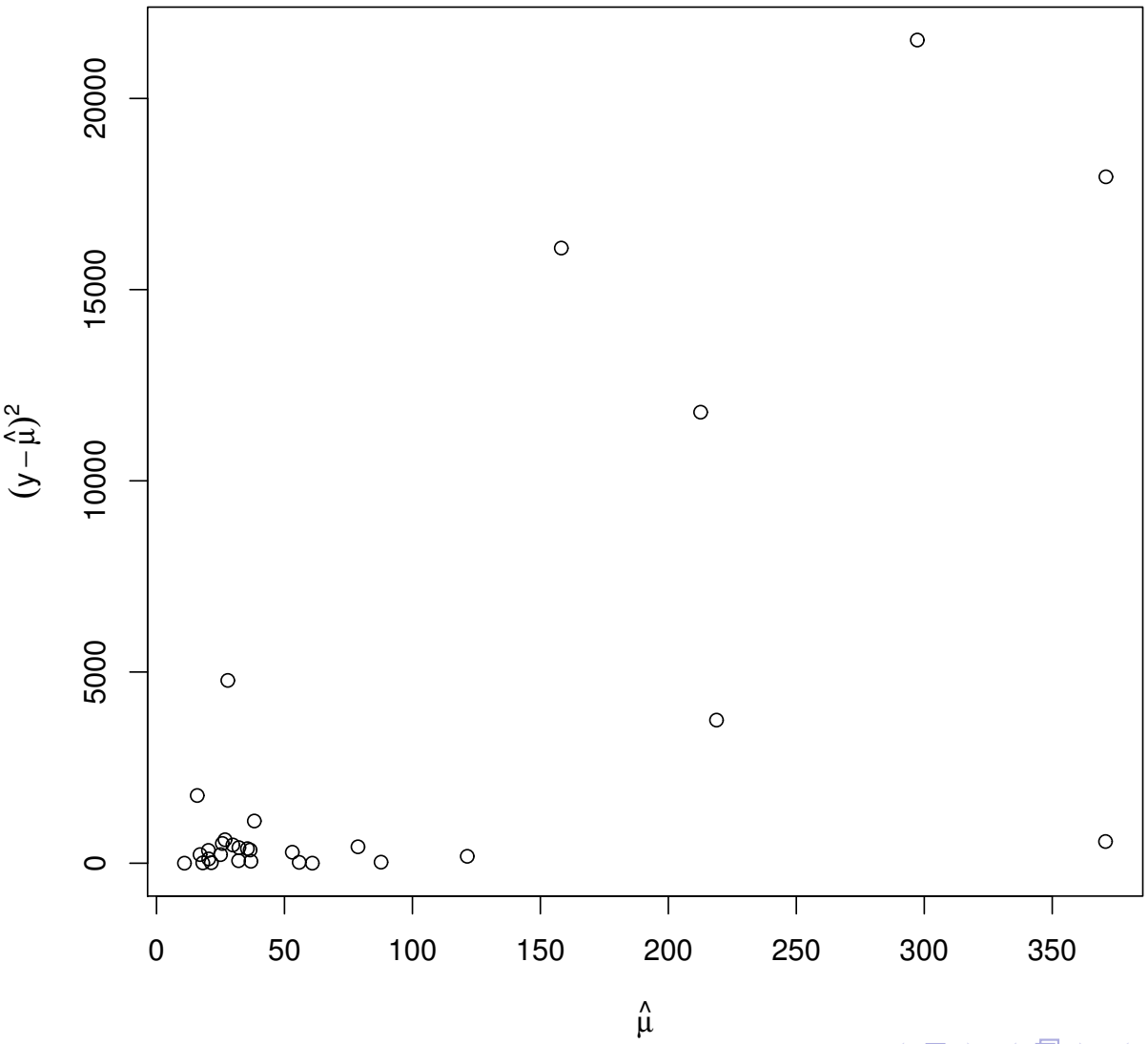
has the $F_{df_S - df_L, n-(p+1)}$ distribution

How do we check for overdispersion?

- Could be due to outliers – check diagnostics (Ch 6)
- As a crude assessment, plot $(y_i - \hat{\mu}_i)^2$ against $\hat{\mu}_i$
- Easier to estimate overdispersion if you have replicates (multiple y's with the same x_i's)

```
## Plot estimated variance against the mean
> plot(galap$fit, residuals(galap, type="response")^2,
+   xlab = expression(hat(mu)),
+   ylab = expression((y - hat(mu))^2))
##  estimate sigma2
> sigma2 = sum(residuals(galap,type="pearson")^2)/
+          galap$df.res
> sigma2
[1] 31.74914
```

The Galapagos data example



```
> # adjust standard errors
> summary(galap,dispersion=sigma2)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.1548079  0.2915897  10.819  < 2e-16 ***
Area        -0.0005799  0.0001480  -3.918 8.95e-05 ***
Elevation    0.0035406  0.0004925   7.189 6.53e-13 ***
Nearest      0.0088256  0.0102621   0.860    0.390
Scruz       -0.0057094  0.0035251  -1.620    0.105
Adjacent    -0.0006630  0.0001653  -4.012 6.01e-05 ***
```

```
> ## F-test - preferred
> drop1(galap, test="F")
Single term deletions
Species ~ Area + Elevation + Nearest + Scruz + Adjacent
          Df Deviance     AIC F value     Pr(F)
<none>         716.85  889.68
Area       1  1204.35 1375.18 16.3217 0.0004762 ***
Elevation  1  2389.57 2560.40 56.0028 1.007e-07 ***
Nearest    1   739.41  910.24  0.7555 0.3933572
Scruz      1   813.62  984.45  3.2400 0.0844448 .
Adjacent   1  1341.45 1512.29 20.9119 0.0001230 ***
---
Warning message:
In drop1.glm(galap, test = "F") : F test assumes
'quasipoisson' family
```