Analysis of Categorical Data

Christopher R. Bilder[1] and Thomas M. Loughin[2]

[1]University of Nebraska–Lincoln, Department of Statistics

[2]Simon Fraser University, Department of Statistics and Actuarial Science

www.chrisbilder.com/categorical

- Apply appropriate methods to analyze data in a contingency table
- State, interpret, and fit logistic, multinomial, proportional odds, and Poisson regression models
- Use appropriate variable-selection methods
- Evaluate the fit of categorical regression models
- Identify and solve overdispersion problems
- Be comfortable with using R to analyze categorical data

1. Introduction
   - Objectives
   - Table of contents
   - Textbook
   - Additional items
2. Analyzing a binary response, 2×2 tables
   - Background
   - Odds ratios
3. Analyzing a binary response, logistic regression
   - Regression model
   - Estimation
   - Hypothesis tests
   - Odds ratios
   - Probability of success
   - Explanatory variable formats
   - Generalized linear models
4. Analyzing a multicategory response

   - Introduction
   - Nominal response regression models
   - Ordinal response regression models
5. Analyzing a count response
   - Introduction
   - The Poisson distribution
   - Poisson regression models
   - Categorical explanatory variables
   - Poisson regression for contingency tables
   - Poisson regression for large contingency tables
   - Poisson regression with ordinal variables
   - Poisson rate regression
6. Model selection and evaluation
   - Introduction
   - Variable selection
   - Residual analysis
   - Goodness-of-fit statistics

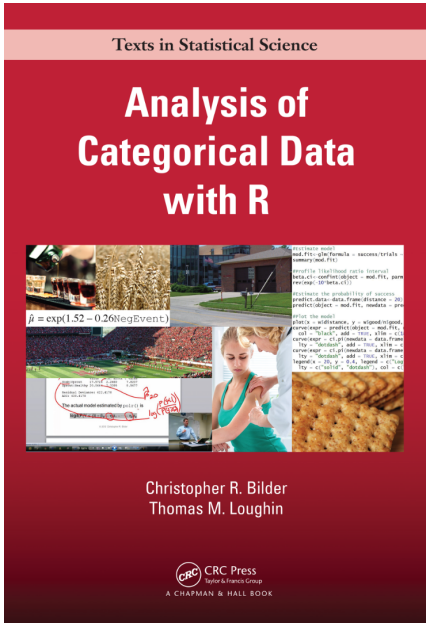
- Overdispersion

7 Models for correlated data
- Introduction
- Random effects
- Generalized linear mixed models
- Inference in GLMMs
- Extensions

8 Conclusion
- Objectives
- Additional material

Section/subsection given at the top of each slide

- Bilder and Loughin (2014) published by CRC Press
- Provides more depth and additional material
- `www.chrisbilder.com/categorical`
  - R programs with >11,000 lines of code
  - >35 hours of instructional videos
  - Lecture notes, projects, tests



- 8:30AM – 5:00PM: Course in session
- When are the breaks?
  - 10:15AM – 10:30AM: Break
  - 12:30PM – 2:00PM: Lunch
  - 3:15PM – 3:30PM: Break
- `www.chrisbilder.com/JSM`
  - Recording
    - Computer screen, including annotations made on it
    - Live-action video of us
    - Post to website within 1 week from today; available for 1 month
  - R programs (link to book's website)
  - Handouts available for electronic note taking
- Handouts
  - All slides presented
  - Data examples introduced by "Example: Name (R programs)"
  - R Index
  - Recommend follow along in handouts rather than try during course
  - Additional slides available on website
- **Bold blue text on screen – Added after handouts printed**

1 Introduction

2 Analyzing a binary response, 2 × 2 tables
- Background
- Odds ratios

3 Analyzing a binary response, logistic regression

4 Analyzing a multicategory response

5 Analyzing a count response

6 Model selection and evaluation

7 Models for correlated data

8 Conclusion

- Compare responses of two groups in a 2×2 contingency table
- Larry Bird's free throws for two seasons (Wardrop, 1995)

		Second		
		Made	Missed	Total
First	Made	251	34	285
	Missed	48	5	53
	Total	299	39	338

- HIV vaccine clinical trials (Rerks-Ngarm et al., 2009)
- PBS Newshour on September 24, 2009

- Data

		Response		
		HIV	No HIV	Total
Treatment	Vaccine	51	8,146	8,197
	Placebo	74	8,124	8,198
	Total	125	16,270	16,395

- Binary response with levels "success" and "failure"
- Denote π_1 and π_2 as the probabilities of a success for the two groups
- 2×2 contingency table

		Response		
		Success	Failure	Total
Group	1	π_1	$1-\pi_1$	1
	2	π_2	$1-\pi_2$	1

		Response		
		Success	Failure	Total
Group	1	w_1	$n_1 - w_1$	n_1
	2	w_2	$n_2 - w_2$	n_2

- $W_j \sim \text{Binomial}(n_j, \pi_j)$ for $j = 1, 2$
  - Maximum likelihood estimate (MLE) for π_j: $\hat{\pi}_j = w_j/n_j$
  - Properties of maximum likelihood estimators for large samples:
    - Normal distribution
    - Consistent
    - Variance estimated by

$$-E\left(\frac{\partial^2}{\partial\theta^2}\log[L(\theta|\mathbf{X})]\right)^{-1}\Big|_{\theta=\hat{\theta}}$$

where θ is a generic parameter of interest, $\hat{\theta}$ is the MLE, $\mathbf{X}$ is a matrix of our random variables, and $\log(\cdot)$ is the natural log function

  - $\hat{\pi}_j \dot{\sim} N(\pi_j, \widehat{Var}(\hat{\pi}_j))$ for large n_j, where $\widehat{Var}(\hat{\pi}_j) = \hat{\pi}_j(1-\hat{\pi}_j)/n_j$

- Contingency table

		Response		
		Success	Failure	Total
Group	1	π_1	$1-\pi_1$	1
	2	π_2	$1-\pi_2$	1

- How can one compare the responses for the two groups?
  - Difference in probabilities: $\pi_1 - \pi_2$
  - Relative risk: π_1/π_2
  - Odds ratio: $odds_1/odds_2$

- Odds of a success
  - Rescaling of the probability of a success π
  - (probability of a success)/(probability of a failure) $= \pi/(1-\pi)$
  - If $\pi = 0.1$, then $odds = 0.1/(1-0.1) = 1/9$
    - "9-to-1 odds *against*" because the probability of failure is 9 times the probability of success
  - Group 1: $odds_1 = \pi_1/(1-\pi_1)$
  - Group 2: $odds_2 = \pi_2/(1-\pi_2)$
- Odds ratio

$$OR = \frac{odds_1}{odds_2} = \frac{\pi_1/(1-\pi_1)}{\pi_2/(1-\pi_2)} = \frac{\pi_1(1-\pi_2)}{\pi_2(1-\pi_1)}$$

- Interpretation
  - The odds of a success are OR times as large for group 1 than for group 2
  - The odds of a success are $1/OR$ times as large for group 2 than for group 1

- Odds of a failure: $(1-\pi)/\pi$
- Odds ratio:

$$\frac{(1-\pi_1)/\pi_1}{(1-\pi_2)/\pi_2} = \frac{\pi_2(1-\pi_1)}{\pi_1(1-\pi_2)} = \frac{1}{OR}$$

- Interpretation:
  - The odds of a failure are $1/OR$ times as large for group 1 than for group 2
  - The odds of a failure are OR times as large as for group 2 than for group 1
- What if $OR = 1$?
- Odds ratio written in terms of expected counts
  - Expected number of successes: $E(W_j) = n_j\pi_j$
  - Expected number of failures: $n_j - E(W_j) = n_j(1-\pi_j)$
  - Odds of a success:

$$\begin{aligned} odds_j &= \pi_j/(1-\pi_j) \\ &= n_j\pi_j/[n_j(1-\pi_j)] \\ &= E(W_j)/[n_j - E(W_j)] \end{aligned}$$

- Contingency table

		Response		
		1	2	Total
Group	1	w_1	$n_1 - w_1$	n_1
	2	w_2	$n_2 - w_2$	n_2

- Maximum likelihood estimate (MLE):

$$\widehat{OR} = \frac{\hat{\pi}_1(1-\hat{\pi}_2)}{\hat{\pi}_2(1-\hat{\pi}_1)} = \frac{(w_1/n_1)[(n_2-w_2)/n_2]}{(w_2/n_2)[(n_1-w_1)/n_1]} = \frac{w_1(n_2-w_2)}{w_2(n_1-w_1)}$$

- What if a cell count is 0? Possible ad-hoc solutions:
  - Add 0.5 to the count
  - Add 0.5 to all counts

- Wald confidence interval
  - Normal approximation is better for $\log(\widehat{OR})$ than for $\widehat{OR}$
  - Estimated variance

$$\widehat{Var}(\log(\widehat{OR})) = \frac{1}{w_1} + \frac{1}{n_1-w_1} + \frac{1}{w_2} + \frac{1}{n_2-w_2}$$

    - Problems when a cell count is 0
  - Interval for $\log(OR)$

$$\log\left(\widehat{OR}\right) \pm Z_{1-\alpha/2}\sqrt{\frac{1}{w_1} + \frac{1}{n_1-w_1} + \frac{1}{w_2} + \frac{1}{n_2-w_2}}$$

  - Interval for OR

$$\exp\left[\log\left(\widehat{OR}\right) \pm Z_{1-\alpha/2}\sqrt{\frac{1}{w_1} + \frac{1}{n_1-w_1} + \frac{1}{w_2} + \frac{1}{n_2-w_2}}\right]$$

Example: HIV vaccine (HIVvaccine.R)

```
> c.table <- array(data = c(51, 74, 8146, 8124), dim = c(2, 2),
    dimnames = list(Trt = c("vaccine", "placebo"), Response = c("HIV",
        "No HIV")))
> c.table
         Response
Trt       HIV No HIV
  vaccine  51   8146
  placebo  74   8124
> c.table[1, 1]  #Row 1, column 2 count
[1] 51
> c.table[1, ]  #Row 1 counts
   HIV No HIV
    51   8146
> n1 <- sum(c.table[1, ])
> n2 <- sum(c.table[2, ])
> pi.hat1 <- c.table[1, 1]/n1
> pi.hat2 <- c.table[2, 1]/n2
> pi.hat1/pi.hat2
[1] 0.6893
```


Example: HIV vaccine (HIVvaccine.R)

```
> OR.hat <- c.table[1, 1] * c.table[2, 2]/(c.table[2, 1] * c.table[1,
    2])  # w1*(n2-w2)/[w2*(n1-w1)]
> round(OR.hat, 2)
[1] 0.69
> alpha <- 0.05
> var.log.or <- 1/c.table[1, 1] + 1/c.table[1, 2] + 1/c.table[2,
    1] + 1/c.table[2, 2]  # 1/w1 + 1/(n1-w1) + 1/w2 + 1/(n2-w2)
> OR.CI <- exp(log(OR.hat) + qnorm(p = c(alpha/2, 1 - alpha/2)) *
    sqrt(var.log.or))
> round(OR.CI, 2)
[1] 0.48 0.98
> rev(round(1/OR.CI, 2))
[1] 1.02 2.08
```

- With 95% confidence,
  - the odds of contracting HIV are between 0.48 and 0.98 times as large for the vaccine group than for the placebo group
  - the vaccine reduces the odds of HIV infection by 2% to 52%
  - the odds of contracting HIV are between 1.02 and 2.08 times as large for the placebo group than for the vaccine group
  - the odds of being HIV free are between 1.02 and 2.08 times as large for the vaccine group than for the placebo group

Example: HIV vaccine (HIVvaccine.R)

- Other functions to perform the calculations
  - `twoby2()` from the `Epi` package
  - `oddsratio()` function from the `epitools` package

1. Introduction
2. Analyzing a binary response, 2 × 2 tables
3. Analyzing a binary response, logistic regression
   - Regression model
   - Estimation
   - Hypothesis tests
   - Odds ratios
   - Probability of success
   - Explanatory variable formats
   - Generalized linear models
4. Analyzing a multicategory response
5. Analyzing a count response
6. Model selection and evaluation
7. Models for correlated data
8. Conclusion

- Linear regression model:

$$E(Y_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

for $i = 1, \ldots, n$

  - $\beta_0, \ldots, \beta_p$ are regression parameters
  - $x_{i1}, \ldots, x_{ip}$ are explanatory variables
  - $Y_i \sim$ ind. $N(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \sigma^2)$

- What if $Y_i \sim$ ind. $\text{Bernoulli}(\pi_i)$, where $Y_i = 1$ is a "success", $Y_i = 0$ is a "failure", and $E(Y_i) = \pi_i$?
  - Y_i does not have a normal distribution
  - $\text{Var}(Y_i) = \pi_i(1 - \pi_i)$ leads to potentially different variances
  - $\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ is not constrained to be within 0 and 1

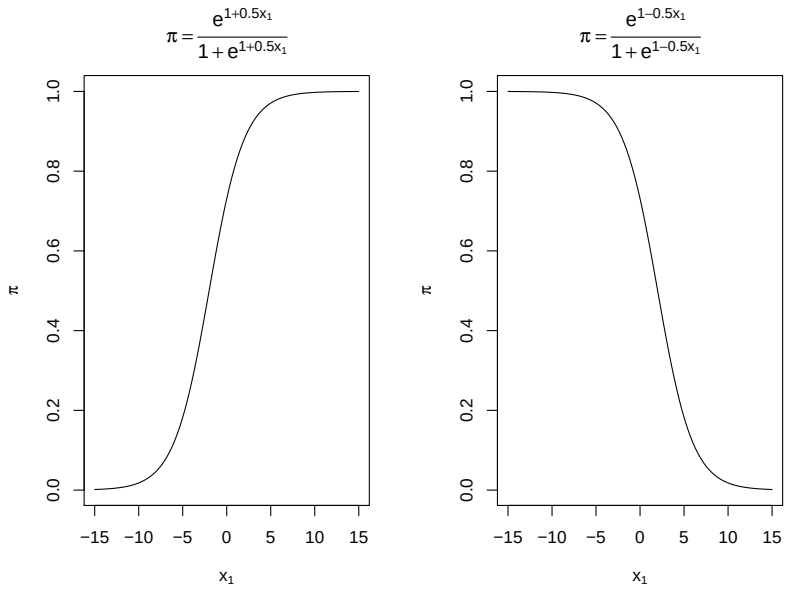
- Logistic regression model:

$$\pi_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}$$

for $i = 1, \ldots, n$

- Advantages:
  - $\text{Var}(Y_i) = \pi_i(1 - \pi_i)$ is o.k.
  - Constrained to be within 0 and 1
- $\exp(\cdot)/[1 + \exp(\cdot)]$ transformation has the same form as a logistic cumulative distribution function
- Equivalent representations of logistic regression model
  - $\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$
  - $\text{logit}(\pi_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$
  - $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - $\pi = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}$

Example: Plot of logistic regression model (PiPlot.R)



- Maximum likelihood estimation:

$$L(\boldsymbol{\beta}|\mathbf{y}) = \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i)^{1-y_i}$$

where $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$, $\mathbf{y} = (y_1, \ldots, y_n)'$, and

$$\pi_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}$$

- Numerical iterative methods are used to find the $\boldsymbol{\beta}$ that maximize the likelihood function
  - Iteratively reweighted least squares, Fisher scoring
  - `glm()` function in R
  - Convergence issues
- Regression parameter estimates: $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)'$
- Estimated variance-covariance matrix for regression parameter estimates:

$$\widehat{Var}(\hat{\boldsymbol{\beta}}) = -E\left(\frac{\partial^2}{\partial \boldsymbol{\beta}^2} \log[L(\boldsymbol{\beta}|\mathbf{y})]\right)^{-1}\Bigg|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}$$

Example: Placekicking (Placekick.R, Placekick.csv)

- Estimate the probability of success for a placekick in the NFL
  - Points are scored by kicking a ball through a target area
- Video
- Bilder and Loughin (1998)
- Data from the 1995 NFL season
- Variables
  - good: Binary response variable denoting successful (1) vs. failed (0) placekicks
  - distance: Distance of the placekick in yards

Example: Placekicking (Placekick.R, Placekick.csv)

```
> placekick <- read.csv(file = "C:\\data\\Placekick.csv")
> head(placekick, n = 3)
  week distance change elap30 PAT type field wind good
1    1       21      1  24.72   0    1     1    0    1
2    1       21      0  15.85   0    1     1    0    1
3    1       20      0   0.45   1    1     1    0    1
> mod.fit <- glm(formula = good ~ distance, family = binomial(link = logit),
    data = placekick)
> names(mod.fit)
 [1] "coefficients"      "residuals"         "fitted.values"
 [4] "effects"           "R"                 "rank"
 [7] "qr"                "family"            "linear.predictors"
[10] "deviance"          "aic"               "null.deviance"
[13] "iter"              "weights"           "prior.weights"
[16] "df.residual"       "df.null"           "y"
[19] "converged"         "boundary"          "model"
[22] "call"              "formula"           "terms"
[25] "data"              "offset"            "control"
[28] "method"            "contrasts"         "xlevels"
> mod.fit$coefficients
(Intercept)    distance
      5.812      -0.115
```


Example: Placekicking (Placekick.R, Placekick.csv)

```
> summary(object = mod.fit)

Call:
glm(formula = good ~ distance, family = binomial(link = logit),
    data = placekick)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.744   0.242   0.242   0.380   1.609

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.81208    0.32628    17.8   <2e-16 ***
distance    -0.11503    0.00834   -13.8   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1013.43  on 1424  degrees of freedom
Residual deviance:  775.75  on 1423  degrees of freedom
AIC: 779.7

Number of Fisher Scoring iterations: 6
```


Example: Placekicking (Placekick.R, Placekick.csv)

- The estimated logistic regression model is

$$\text{logit}(\hat{\pi}) = 5.8121 - 0.1150\texttt{distance}$$

- The estimated variance-covariance matrix is

```
> vcov(mod.fit)
             (Intercept)   distance
(Intercept)    0.106457 -2.606e-03
distance      -0.002606  6.954e-05
```

- Logistic regression model with two explanatory variables:
  $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
  - $H_0 : \beta_2 = 0$
    $H_a : \beta_2 \neq 0$
  - $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 x_1$
    $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
  - Likelihood ratio test (LRT) statistic can be written informally as

$$\Lambda = \frac{\text{Maximum of likelihood function under } H_0}{\text{Maximum of likelihood function under } H_0 \text{ or } H_a}$$

  - $-2\log(\Lambda)$ statistic has an approximate χ^2_1 distribution under H_0 for a large sample
  - Reject H_0 for large values of $-2\log(\Lambda)$ relative to a χ^2_1 distribution
- When H_0 contains q regression parameters set to 0, use a χ^2_q distribution
- LRTs are generally better than Wald tests

Example: Placekicking (Placekick.R, Placekick.csv)

- change: Binary variable denoting lead-change (1) vs. non-lead-change (0) placekicks
- Use change and distance to estimate the probability of success

```
> mod.fit2 <- glm(formula = good ~ change + distance,
    family = binomial(link = logit), data = placekick)
> summary(mod.fit2)

Call:
glm(formula = good ~ change + distance, family = binomial(link = logit),
    data = placekick)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.706   0.228   0.228   0.375   1.565

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.89318    0.33318   17.69   <2e-16 ***
change      -0.44778    0.19367   -2.31    0.021 *
distance    -0.11289    0.00844  -13.37   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```


Example: Placekicking (Placekick.R, Placekick.csv)

- $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{distance}$
  $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change} + \beta_2 \texttt{distance}$
- Using anova() to perform the LRT

```
> mod.fit <- glm(formula = good ~ distance, family = binomial(link = logit),
    data = placekick)
> anova(mod.fit, mod.fit2, test = "Chisq")
Analysis of Deviance Table

Model 1: good ~ distance
Model 2: good ~ change + distance
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      1423        776
2      1422        770  1     5.25    0.022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> c(mod.fit$deviance, mod.fit2$deviance)
[1] 775.7 770.5
```

- Residual deviance: $-2\log(\Lambda)$ statistic used to test
  H_0 : Model of interest vs. H_a : Saturated model
  (resid dev distance) − (resid dev change/distance) = 775.75 − 770.50 = 5.25

Example: Placekicking (Placekick.R, Placekick.csv)

- $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{distance}$
  $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change} + \beta_2 \texttt{distance}$
- Using Anova() from the car package to perform the LRT

```
> library(package = car)
> Anova(mod = mod.fit2, test = "LR")
Analysis of Deviance Table (Type II tests)

Response: good
         LR Chisq Df Pr(>Chisq)
change        5.2  1      0.022 *
distance    218.6  1     <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{distance}$
  $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change} + \beta_2 \texttt{distance}$
  p-value = 0.022
- $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change}$
  $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change} + \beta_2 \texttt{distance}$
  p-value $< 2 \times 10^{-16}$

Example: Placekicking (Placekick.R, Placekick.csv)

- Be careful! `anova(mod.fit2, test = "Chisq")` does not produce the same results as `Anova(mod = mod.fit2, test = "LR")`

```
> anova(mod.fit2, test = "Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: good

Terms added sequentially (first to last)


         Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                      1424       1013
change    1     24.3      1423        989  8.3e-07 ***
distance  1    218.6      1422        770  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- $H_0 : \text{logit}(\pi) = \beta_0$ vs. $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change}$
  P-value $= 8.3 \times 10^{-7}$
- $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change}$ vs. $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 \texttt{change} + \beta_2 \texttt{distance}$
  P-value $< 2 \times 10^{-16}$

- Logistic regression model with one explanatory variable
  - Modeling the log odds of a success: $\log(\pi/(1-\pi)) = \beta_0 + \beta_1 x$
  - Odds of a success: $\pi/(1-\pi) = \exp(\beta_0 + \beta_1 x)$
- Compare two odds of a success:
  - Odds at x: $Odds_x = \exp(\beta_0 + \beta_1 x)$
  - Odds at $x + c$: $Odds_{x+c} = \exp(\beta_0 + \beta_1 (x + c))$ for some constant c
  - Odds ratio:

$$OR = \frac{Odds_{x+c}}{Odds_x} = \frac{\exp(\beta_0 + \beta_1 (x + c))}{\exp(\beta_0 + \beta_1 x)} = \exp(c\beta_1)$$

- Interpretation:
  - The odds of a success change by $\exp(c\beta_1)$ times for every c-unit increase in x
  - If additional explanatory variables are in the model, add "holding the other variables constant"

- Estimated odds ratio: $\widehat{OR} = \exp(c\hat{\beta}_1)$
- $(1-\alpha)100\%$ confidence intervals:
  - Wald interval: $\exp\left(c\hat{\beta}_1 \pm cZ_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_1)}\right)$
  - Profile likelihood ratio interval
    - Find the set of β_1 values such that

$$-2\log\left(\frac{L(\tilde{\beta}_0, \beta_1 | y_1, \ldots, y_n)}{L(\hat{\beta}_0, \hat{\beta}_1 | y_1, \ldots, y_n)}\right) < \chi^2_{1,1-\alpha}$$

    and take $\exp(c \times \text{lower}) < OR < \exp(c \times \text{upper})$ as the interval
    - Use numerical iterative methods
    - Better than Wald with respect to its true confidence level

Example: Placekicking (Placekick.R, Placekick.csv)

- $\text{logit}(\hat{\pi}) = 5.8121 - 0.1150 \texttt{distance}$
- Simple interpretation using $c = 1$: The estimated odds of a success change by

$$\exp(\hat{\beta}_1) = \exp(-0.1150) = 0.89$$

  times for every 1-yard increase in the distance
- Better interpretation using $c = -10$: The estimated odds of a success change by

$$\exp(-10\hat{\beta}_1) = \exp(1.150) = 3.16$$

  times for every 10-yard decrease in the distance.

Example: Placekicking (Placekick.R, Placekick.csv)

```
> beta.ci <- confint(object = mod.fit, parm = "distance", level = 0.95)
> beta.ci
   2.5 %   97.5 %
-0.13181 -0.09907
> rev(exp(-10 * beta.ci))
97.5 %  2.5 %
 2.693  3.736
> as.numeric(rev(exp(-10 * beta.ci)))
[1] 2.693 3.736
```

- With 95% confidence, the odds of a success change by an amount between 2.69 to 3.74 times for every 10-yard decrease in the distance of the placekick
- Notice the interval is above 1!

Example: Placekicking (Placekick.R, Placekick.csv)

- General way to calculate profile likelihood ratio interval using `mcprofile` package
  - Helpful for more complicated linear combinations of regression parameters
  - Find a vector $\mathbf{K} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$ such that $\mathbf{K}\boldsymbol{\beta} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \times \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = k_1\beta_0 + k_2\beta_1 = \beta_1$; $\mathbf{K} = \begin{bmatrix} 0 & 1 \end{bmatrix}$
  - Use `mcprofile()` to evaluate $-2\log\left(\frac{L(\tilde{\beta}_0, \beta_1 | y_1, \ldots, y_n)}{L(\hat{\beta}_0, \hat{\beta}_1 | y_1, \ldots, y_n)}\right)$
  - Use `confint()` to find interval for $\mathbf{K}\boldsymbol{\beta}$ with these evaluations
  - Use `exp()` to find interval for $\exp(c\mathbf{K}\boldsymbol{\beta})$

```
> library(mcprofile)
> K <- matrix(data = c(0, 1), nrow = 1, ncol = 2, byrow = TRUE)
> linear.combo <- mcprofile(object = mod.fit, CM = K)
> ci.log.OR <- confint(object = linear.combo, level = 0.95, adjust = "none")
> names(ci.log.OR)
[1] "estimate"    "confint"     "CM"          "quant"       "alternative"
[6] "level"       "adjust"
> as.numeric(rev(exp(-10 * ci.log.OR$confint)))
[1] 2.693 3.736
```


Example: Placekicking (Placekick.R, Placekick.csv)

- Wald interval
  - `confint.default()` with `mod.fit`
  - `wald()` with `linear.combo`
  - Program provides examples

- Logistic regression model with one explanatory variable
  - Estimated probability of success: $\hat{\pi} = \exp(\hat{\beta}_0 + \hat{\beta}_1 x)/[1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x)]$
  - $(1-\alpha)100\%$ Wald interval
    - Interval for $\beta_0 + \beta_1 x$:

$$\hat{\beta}_0 + \hat{\beta}_1 x \pm Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_0 + \hat{\beta}_1 x)}$$

      where $\widehat{Var}(\hat{\beta}_0 + \hat{\beta}_1 x) = \widehat{Var}(\hat{\beta}_0) + x^2\widehat{Var}(\hat{\beta}_1) + 2x\widehat{Cov}(\hat{\beta}_0, \hat{\beta}_1)$
    - Use $\exp()/[1+\exp()]$ transformation to calculate

$$\frac{\exp\left(\hat{\beta}_0 + \hat{\beta}_1 x \pm Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_0 + \hat{\beta}_1 x)}\right)}{1 + \exp\left(\hat{\beta}_0 + \hat{\beta}_1 x \pm Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_0 + \hat{\beta}_1 x)}\right)}$$

  - $(1-\alpha)100\%$ profile LR interval
    - Find interval for $\beta_0 + \beta_1 x$ such that $-2\log(\Lambda) < \chi^2_{1,1-\alpha}$ using numerical iterative methods
    - Use $\exp()/[1+\exp()]$ transformation to calculate interval for π
- Logistic regression model with more than one explanatory variable

Example: Placekicking (Placekick.R, Placekick.csv)

- $\text{logit}(\hat{\pi}) = 5.8121 - 0.1150\texttt{distance}$
- Estimate the probability of success at a distance of 20 yards

```
> predict.data <- data.frame(distance = 20)
> predict(object = mod.fit, newdata = predict.data, type = "response")
    1
0.971
> K <- matrix(data = c(1, 20), nrow = 1, ncol = 2)
> K
     [,1] [,2]
[1,]    1   20
> linear.combo <- mcprofile(object = mod.fit, CM = K)
> ci.logit.profile <- confint(object = linear.combo, level = 0.95,
    adjust = "none")
> ci.logit.profile$confint  # Interval for beta0*1 + beta1*20
  lower upper
1 3.186 3.867
> exp(ci.logit.profile$confint)/(1 + exp(ci.logit.profile$confint))
   lower  upper
1 0.9603 0.9795
```

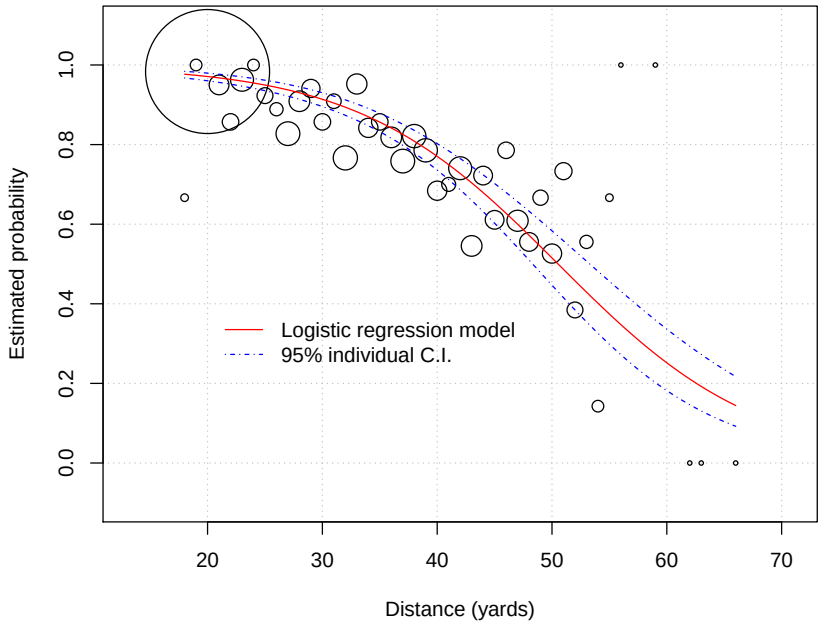

Example: Placekicking (Placekick.R, Placekick.csv)

- Bubble plot of the data with estimated model and 95% confidence interval bands for π

```
> w <- aggregate(formula = good ~ distance, data = placekick, FUN = sum)
> n <- aggregate(formula = good ~ distance, data = placekick, FUN = length)
> w.n <- data.frame(distance = w$distance, success = w$good, trials = n$good,
    proportion = round(w$good/n$good, 4))
> head(w.n)
  distance success trials proportion
1       18       2      3     0.6667
2       19       7      7     1.0000
3       20     776    789     0.9835
4       21      19     20     0.9500
5       22      12     14     0.8571
6       23      26     27     0.9630
```

- See program for plotting code
  - `symbols()` function for bubbles
  - `curve()` function for the lines

Example: Placekicking (Placekick.R, Placekick.csv)



- Pairwise interactions
  - $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$
  - The effect of x_1 on the response depends on the level of x_2
  - Odds ratio:

$$\begin{aligned} OR &= \frac{Odds_{x_1+c}}{Odds_{x_1}} \\ &= \frac{\exp(\beta_0 + \beta_1(x_1 + c) + \beta_2 x_2 + \beta_3(x_1 + c)x_2)}{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2)} \\ &= \exp(\beta_1 c + \beta_3 c x_2) \end{aligned}$$

  - Interpretation for odds ratio: The odds of a success change by $\exp(\beta_1 c + \beta_3 c x_2)$ times for every c-unit increase in x_1 when $x_2 =$ _
  - `formula` argument of `glm()`
    - `formula = y ~ x1 + x2 + x1:x2`
    - `formula = y ~ x1*x2`
    - `formula = y ~ (x1 + x2)^2`

- Quadratic terms
  - $\text{logit}(\pi) = \beta_0 + \beta_1 x + \beta_2 x^2$
  - Odds ratio:

$$\begin{aligned} OR &= \frac{Odds_{x+c}}{Odds_x} = \\ &= \frac{\exp(\beta_0 + \beta_1(x+c) + \beta_2(x+c)^2)}{\exp(\beta_0 + \beta_1 x + \beta_2 x^2)} \\ &= \exp(\beta_1 c + \beta_2(2xc + c^2)) \end{aligned}$$

  - Interpretation for odds ratio: The odds of a success are $\exp(\beta_1 c + \beta_2(2xc + c^2))$ times as large for $x =$ _ $+ c$ than for $x =$ _
  - `formula` argument of `glm()`: `formula = y ~ x + I(x^2)`

- Categorical explanatory variables
  - A q-level explanatory variable needs $q - 1$ indicator variables to represent it in the model
  - Suppose there is a 4-level explanatory variable named `cat` with levels A, B, C, and D; $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

	Indicator variables			
	x_1	x_2	x_3	Model
A	0	0	0	$\text{logit}(\pi) = \beta_0$
B	1	0	0	$\text{logit}(\pi) = \beta_0 + \beta_1$
C	0	1	0	$\text{logit}(\pi) = \beta_0 + \beta_2$
D	0	0	1	$\text{logit}(\pi) = \beta_0 + \beta_3$

    - Odds ratio comparing B to A

$$\frac{Odds_B}{Odds_A} = \frac{\exp(\beta_0 + \beta_1 1 + \beta_2 0 + \beta_3 0)}{\exp(\beta_0 + \beta_1 0 + \beta_2 0 + \beta_3 0)} = \exp(\beta_1)$$

    - Odds ratio comparing B to C

$$\frac{Odds_B}{Odds_C} = \frac{\exp(\beta_0 + \beta_1 1 + \beta_2 0 + \beta_3 0)}{\exp(\beta_0 + \beta_1 0 + \beta_2 1 + \beta_3 0)} = \exp(\beta_1 - \beta_2)$$

    - `formula` argument of `glm()`: `formula = y ~ cat`
  - R orders levels alphabetically; sets first level to its "base" level

Example: Control of the Tomato Spotted Wilt Virus (TomatoVirus.R, TomatoVirus.csv)

- Backyard Farmer video – `https://youtu.be/9DiL-UQ6-Uw` (start at 3:38)
- 16 greenhouses each with 100 uninfected tomato plants
- Virus introduced into each greenhouse
  - (`Infect = 1`) Add infected tomato plants and release uninfected thrips
  - (`Infect = 2`) Release infected thrips
- Control spread of virus to plants
  - (`Control = B`) Biologically through using predatory spider mites
  - (`Control = C`) Chemically using a pesticide
  - (`Control = N`) None
- Binomial response: Number of tomato plants displaying symptoms in a greenhouse after 8 weeks

Example: Control of the Tomato Spotted Wilt Virus ...

```
> tomato <- read.csv(file = "C:\\data\\TomatoVirus.csv")
> head(tomato)
  Infest Control Plants Virus8
1      1       C    100     21
2      2       C    100     10
3      1       B    100     19
4      1       N    100     40
5      2       C    100     30
6      2       B    100     30
> class(tomato$Control)
[1] "factor"
> levels(tomato$Control)
[1] "B" "C" "N"
> contrasts(tomato$Control)
  C N
B 0 0
C 1 0
N 0 1
```

Example: Control of the Tomato Spotted Wilt Virus ...

```
> class(tomato$Infest)
[1] "integer"
> tomato$Infest <- factor(tomato$Infest)
> class(tomato$Infest)
[1] "factor"
> contrasts(tomato$Infest)
  2
1 0
2 1
```


Example: Control of the Tomato Spotted Wilt Virus ...

- Syntax for `glm()` is a little different than earlier due to the response format

```
> mod.fit.inter <- glm(formula = Virus8/Plants ~ Infest + Control +
    Infest:Control, family = binomial(link = logit), data = tomato,
    weights = Plants)
> summary(mod.fit.inter)

Call:
glm(formula = Virus8/Plants ~ Infest + Control + Infest:Control,
    family = binomial(link = logit), data = tomato, weights = Plants)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
 -3.47   -2.71   -1.27    2.81    6.79

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.046      0.132   -7.95  1.9e-15 ***
Infest2           0.926      0.175    5.28  1.3e-07 ***
ControlC         -0.162      0.190   -0.85     0.39
ControlN          1.126      0.193    5.83  5.7e-09 ***
Infest2:ControlC -1.211      0.268   -4.52  6.1e-06 ***
Infest2:ControlN -1.166      0.266   -4.38  1.2e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Example: Control of the Tomato Spotted Wilt Virus ...

- The estimated logistic regression model is

$$\begin{aligned}\text{logit}(\hat{\pi}) &= -1.0460 + 0.9258\texttt{Infest2} - 0.1623\texttt{ControlC} \\ &\quad +1.1260\texttt{ControlN} - 1.2114\texttt{Infest2}\times\texttt{ControlC} \\ &\quad -1.1662\texttt{Infest2}\times\texttt{ControlN}\end{aligned}$$

- Hypothesis test:

$$\begin{aligned}H_0: \text{logit}(\pi) &= \beta_0 + \beta_1\texttt{Infest2} + \beta_2\texttt{ControlC} + \beta_3\texttt{ControlN} \\ H_a: \text{logit}(\pi) &= \beta_0 + \beta_1\texttt{Infest2} + \beta_2\texttt{ControlC} + \beta_3\texttt{ControlN} + \\ &\quad \beta_4\texttt{Infest2}\times\texttt{ControlC} + \beta_5\texttt{Infest2}\times\texttt{ControlN}\end{aligned}$$

```
> library(package = car)
> Anova(mod.fit.inter, test = "LR")
Analysis of Deviance Table (Type II tests)

Response: Virus8/Plants
             LR Chisq Df Pr(>Chisq)
Infest            4.1  1      0.044 *
Control          91.6  2    < 2e-16 ***
Infest:Control   28.2  2    7.4e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Example: Control of the Tomato Spotted Wilt Virus ...

- The estimated logistic regression model is

$$\begin{aligned}\text{logit}(\hat{\pi}) &= -1.0460 + 0.9258\texttt{Infest2} - 0.1623\texttt{ControlC} \\ &\quad +1.1260\texttt{ControlN} - 1.2114\texttt{Infest2}\times\texttt{ControlC} \\ &\quad -1.1662\texttt{Infest2}\times\texttt{ControlN}\end{aligned}$$

- Odds ratios for `Control` when `Infest` is at a fixed level
  - Compare `"N"` to `"B"` with `Infest2` $= 0$

$$\frac{Odds_{\texttt{ControlC=0,ControlN=1,Infest2=0}}}{Odds_{\texttt{ControlC=0,ControlN=0,Infest2=0}}} = \frac{\exp(\beta_0+\beta_3)}{\exp(\beta_0)} = \exp(\beta_3)$$

  - Compare `"N"` to `"B"` with `Infest2` $= 1$

$$\frac{Odds_{\texttt{ControlC=0,ControlN=1,Infest2=1}}}{Odds_{\texttt{ControlC=0,ControlN=0,Infest2=1}}} = \frac{\exp(\beta_0+\beta_1+\beta_3+\beta_5)}{\exp(\beta_0+\beta_1)} = \exp(\beta_3+\beta_5)$$

Example: Control of the Tomato Spotted Wilt Virus ...

```
> row.name <- c("N vs. B, Infest2=0", "N vs. B, Infest2=1", "C vs. B, Infest2=0
    "C vs. B, Infest2=1", "N vs. C, Infest2=0", "N vs. C, Infest2=1")
> col.name <- c("beta0", "beta1", "beta2", "beta3", "beta4", "beta5")
> K <- matrix(data = c(0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
    1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, -1, 1, 0, 0, 0, 0, -1,
    1, -1, 1), nrow = 6, ncol = 6, byrow = TRUE, dimnames = list(row.name,
    col.name))
> K
                  beta0 beta1 beta2 beta3 beta4 beta5
N vs. B, Infest2=0     0     0     0     1     0     0
N vs. B, Infest2=1     0     0     0     1     0     1
C vs. B, Infest2=0     0     0     1     0     0     0
C vs. B, Infest2=1     0     0     1     0     1     0
N vs. C, Infest2=0     0     0    -1     1     0     0
N vs. C, Infest2=1     0     0    -1     1    -1     1
```


Example: Control of the Tomato Spotted Wilt Virus ...

```
> library(package = mcprofile)
> linear.combo <- mcprofile(object = mod.fit.inter, CM = K)
> ci.log.ss <- confint(object = linear.combo, level = 0.95,
    adjust = "single-step")
> exp(ci.log.ss)  # exp(ci.log.ss$confint) gives just the intervals

   mcprofile - Confidence Intervals

level:   0.95
adjustment:  single-step

                  Estimate lower upper
N vs. B, Infest2=0    3.083 1.874  5.12
N vs. B, Infest2=1    0.961 0.596  1.55
C vs. B, Infest2=0    0.850 0.517  1.39
C vs. B, Infest2=1    0.253 0.153  0.41
N vs. C, Infest2=0    3.627 2.184  6.09
N vs. C, Infest2=1    3.795 2.237  6.54
```

- Familywise error rate control
  - Used "single-step" here which is similar to Tukey's studentized-range statistic for pairwise comparisons in ANOVA
  - Could use `adjust = "bonferroni"` or `adjust = "none"`

- Logistic regression models fall within a family of models called generalized linear models (GLMs).
- A GLM has three different components:
  - Random: Distribution for response
  - Systematic: Linear combination of explanatory variables with the regression parameters
  - Link: Specifies how the expected value of the response is linked to the systematic component
- Logistic regression – $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - Random: $Y \sim \text{Bernoulli}(\pi)$
  - Systematic: $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
  - Link: logit function
- Other link functions are sometimes used for binary responses
  - Probit regression – Link uses the inverse of a standard normal cumulative distribution function
  - Complementary log-log regression – Link uses the inverse of a Gumbel cumulative distribution function

- Logistic regression is used much more often
  - Odds ratio remains the same for a c-unit increase in an explanatory variable (without transformations or interactions)
  - Model fit is very similar to a probit regression

1 Introduction
2 Analyzing a binary response, 2×2 tables
3 Analyzing a binary response, logistic regression
4 Analyzing a multicategory response
- Introduction
- Nominal response regression models
- Ordinal response regression models

5 Analyzing a count response
6 Model selection and evaluation
7 Models for correlated data
8 Conclusion

- Examples
  - Five-level Likert scale – Strongly disagree, disagree, neutral, agree, or strongly agree
  - Chemical compounds in drug discovery experiments – Positive, blocker, or neither
  - Canadian political party affiliation – Conservative, New Democratic, Liberal, Bloc Quebecois, or Green
- Let Y denote the categorical response random variable
  - Levels $j = 1, \ldots, J$
  - $\pi_j = P(Y = j)$ with $\sum_{j=1}^{J} \pi_j = 1$
- Suppose there are n identical trials with responses $Y_1, \ldots, Y_n$
  - N_j = the number of trials responding with category j (count)
  - Multinomial distribution for the counts

$$P(N_1 = n_1, \ldots, N_J = n_J) = \frac{n!}{\prod_{j=1}^{J} n_j!} \prod_{j=1}^{J} \pi_j^{n_j}$$

- J categories for Y (no ordering) with corresponding probabilities $\pi_1, \ldots, \pi_J$
- Odds
  - Observe category j relative to category j': $\pi_j/\pi_{j'}$ $(j \neq j')$
  - Only need to know $J - 1$ of them to have all combinations
  - Example: Set $j' = 1$ and $J = 3$
    - Suppose we have values for π_2/π_1 and π_3/π_1
    - Then $(\pi_3/\pi_1)/(\pi_2/\pi_1) = \pi_3/\pi_2$
- Relate $J - 1$ log odds to explanatory variables:

$$\log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p$$

for $j = 2, \ldots, J$

- Multinomial regression model
  - Also known as a baseline-category logit model

- What is π_j?
  - Odds of response j relative to 1: $\pi_j/\pi_1 = \exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)$
  - Expression for π_j: $\pi_j = \pi_1 \exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)$
  - Noting that $\pi_1 + \pi_2 + \cdots + \pi_J = 1$, we have

$$\pi_1 + \pi_1 \exp(\beta_{20} + \beta_{21}x_1 + \cdots + \beta_{2p}x_p) + \cdots + \pi_1 \exp(\beta_{J0} + \beta_{J1}x_1 + \cdots + \beta_{Jp}x_p) = 1$$

  - Solving for π_1 leads to

$$\pi_1 = \frac{1}{1 + \sum_{j=2}^{J} \exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)}$$

  - General expression for π_j:

$$\pi_j = \frac{\exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)}{1 + \sum_{j=2}^{J} \exp(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jp}x_p)}$$

for $j = 2, \ldots, J$.

- Maximum likelihood estimation
  - Likelihood function is simply the product of multinomial distributions for each observation with $\pi_1, \ldots, \pi_J$ as given on previous slide
  - Numerical iterative methods are used to determine regression parameter estimates

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Develop an automated system to predict whether a wheat kernel is healthy, has sprouted prematurely ("Sprout"), or comes from a plant with a fungus ("Scab")
- From `http://www.ksre.ksu.edu/bookstore/pubs/mf2994.pdf`

Example: Wheat kernels (Wheat.R, Wheat.csv)

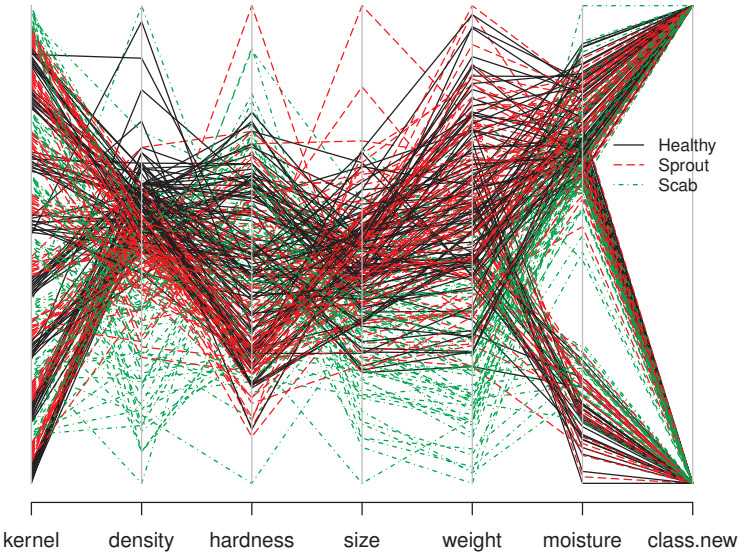
- Relate kernel categories to class of wheat (hard or soft red winter wheat) and five measurement variables

```
> wheat <- read.csv(file = "C:\\data\\Wheat.csv")
> head(wheat, n = 3)
  class density hardness  size weight moisture    type
1   hrw   1.349    60.33 2.303  24.65    12.02 Healthy
2   hrw   1.287    56.09 2.726  33.30    12.17 Healthy
3   hrw   1.234    43.99 2.512  31.76    11.88 Healthy
> tail(wheat, n = 3)
    class density hardness  size weight moisture type
273   srw  0.8492   34.066 1.407  12.09    11.93 Scab
274   srw  1.1770   60.978 1.057   9.48    12.24 Scab
275   srw  1.0306   -9.571 2.057  23.82    12.65 Scab
```

- type is response – determined by human visual inspection

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Parallel coordinates plot



Example: Wheat kernels (Wheat.R, Wheat.csv)

```
> class(wheat$type)
[1] "factor"
> levels(wheat$type)  #j = 1 is 'Healthy'
[1] "Healthy" "Scab"    "Sprout"
> library(package = nnet)
> mod.fit <- multinom(formula = type ~ class + density + hardness +
    size + weight + moisture, data = wheat)
# weights:  24 (14 variable)
initial  value 302.118379
iter  10 value 234.991271
iter  20 value 192.127549
final  value 192.112352
converged
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

```
> summary(mod.fit)
Call:
multinom(formula = type ~ class + density + hardness + size +
    weight + moisture, data = wheat)

Coefficients:
       (Intercept) classsrw density hardness   size   weight moisture
Scab         30.55  -0.6481  -21.60 -0.01591 1.0691 -0.28965   0.1096
Sprout       19.17  -0.2247  -15.12 -0.02102 0.8756 -0.04732  -0.0430

Std. Errors:
       (Intercept) classsrw density hardness   size  weight moisture
Scab         4.290   0.6631   3.116 0.010275 0.7723 0.06170   0.1548
Sprout       3.767   0.5009   2.764 0.008106 0.5409 0.03697   0.1127

Residual Deviance: 384.2
AIC: 412.2
```

- The estimated model is

$$\begin{aligned}\log(\hat{\pi}_{\texttt{Scab}}/\hat{\pi}_{\texttt{Healthy}}) &= 30.55 - 0.65\texttt{srw} - 21.60\texttt{density} - 0.016\texttt{hardness} \\ &\quad +1.07\texttt{size} - 0.29\texttt{weight} + 0.11\texttt{moisture} \\ \log(\hat{\pi}_{\texttt{Sprout}}/\hat{\pi}_{\texttt{Healthy}}) &= 19.17 - 0.22\texttt{srw} - 15.12\texttt{density} - 0.021\texttt{hardness} \\ &\quad +0.88\texttt{size} - 0.047\texttt{weight} - 0.043\texttt{moisture}\end{aligned}$$

Example: Wheat kernels (Wheat.R, Wheat.csv)

- LRTs

```
> library(package = car)
> Anova(mod.fit)
Analysis of Deviance Table (Type II tests)

Response: type
          LR Chisq Df Pr(>Chisq)
class          1.0  2      0.618
density       90.6  2    < 2e-16 ***
hardness       7.1  2      0.029 *
size           3.2  2      0.201
weight        28.2  2    7.4e-07 ***
moisture       1.2  2      0.551
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Estimated probabilities

```
> pi.hat <- predict(object = mod.fit, newdata = wheat, type = "probs")
> head(pi.hat, n = 3)
  Healthy    Scab  Sprout
1  0.8552 0.04640 0.09839
2  0.7493 0.02157 0.22917
3  0.5173 0.06898 0.41374
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

- The same generically named functions are used here as for logistic regression
- Object-oriented language – A *generic* function looks at a class of an object and executes a *method* function

```
> class(mod.fit)
[1] "multinom" "nnet"
> methods(class = multinom)
 [1] add1.multinom*         anova.multinom*        Anova.multinom*
 [4] coef.multinom*         confint.multinom*      deltaMethod.multinom*
 [7] drop1.multinom*        extractAIC.multinom*   logLik.multinom*
[10] model.frame.multinom*  predict.multinom*      print.multinom*
[13] summary.multinom*      vcov.multinom*

   Non-visible functions are asterisked
```

- Use odds ratios to interpret explanatory variables
- Compare two odds:
  - Multinomial regression model with one explanatory variable: $\log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j1}x$
  - Odds of j relative to 1 at x: $\pi_j/\pi_1 = \exp(\beta_{j0} + \beta_{j1}x)$
  - Odds of j relative to 1 at $x + c$: $\pi_j/\pi_1 = \exp(\beta_{j0} + \beta_{j1}(x + c))$
  - Odds ratio:
$$\frac{\exp(\beta_{j0} + \beta_{j1}(x + c))}{\exp(\beta_{j0} + \beta_{j1}x)} = \exp(c\beta_{j1})$$
- Interpretation:
  - The odds of a category j vs. a category 1 response change by $\exp(c\beta_{j1})$ times for every c-unit increase in x
  - If additional explanatory variables are in the model, add "holding the other variables constant"
- Odds ratios for more complicated models – Similar to logistic regression

- Profile LR and Wald intervals can be calculated
- R calculations
  - There is no easy way to calculate profile LR intervals
  - `mcprofile` package cannot be used
  - `confint()` calculates Wald intervals for odds ratios involving simple models
  - Odds ratios for explanatory variables from more complicated models – Need to program "by-hand" formulas into R
    - Suppose the model is $\log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j1}x + \beta_{j2}x^2$ with an odds ratio for x of $\exp(\beta_{j1}c + \beta_{j2}(2xc + c^2))$
    - Wald interval is
$$\exp\left(\hat{\beta}_{j1}c + \hat{\beta}_{j2}(2xc + c^2) \pm Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_{j1}c + \hat{\beta}_{j2}(2xc + c^2))}\right)$$
    where
$$\begin{aligned}\widehat{Var}(\hat{\beta}_{j1}c + \hat{\beta}_{j2}(2xc + c^2)) &= c^2\widehat{Var}(\hat{\beta}_{j1}) + (2xc + c^2)^2\widehat{Var}(\hat{\beta}_{j2}) \\ &\quad + 2c(2xc + c^2)\widehat{Cov}(\hat{\beta}_{j1}, \hat{\beta}_{j2})\end{aligned}$$

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Estimate odds ratio for a c-unit increase in each explanatory variable: $\exp(c\beta_{jr})$ for the rth explanatory variable
- Need to choose a c

```
> head(wheat, n = 3)
  class density hardness  size weight moisture    type
1   hrw   1.349    60.33 2.303  24.65    12.02 Healthy
2   hrw   1.287    56.09 2.726  33.30    12.17 Healthy
3   hrw   1.234    43.99 2.512  31.76    11.88 Healthy
> sd.wheat <- apply(X = wheat[, 2:6], MARGIN = 2, FUN = sd)
> c.value <- c(1, sd.wheat)  # class = 1 is first value
> round(c.value, 2)
         density hardness     size   weight moisture
    1.00     0.13    27.36     0.49     7.92     2.03
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

- 95% Wald confidence intervals for β_{jr}

```
> conf.beta <- confint(object = mod.fit, level = 0.95)
> conf.beta
, , Scab

                2.5 %    97.5 %
(Intercept)  22.13851  38.95448
classsrw     -1.94777   0.65151
density     -27.70474 -15.48957
hardness     -0.03605   0.00423
size         -0.44454   2.58277
weight       -0.41058  -0.16871
moisture     -0.19392   0.41305

, , Sprout

                2.5 %    97.5 %
(Intercept)  11.78496 26.552173
classsrw     -1.20652  0.757047
density     -20.53461 -9.698731
hardness     -0.03691 -0.005133
size         -0.18459  1.935820
weight       -0.11979  0.025153
moisture     -0.26392  0.177928
```

Example: Wheat kernels (Wheat.R, Wheat.csv)

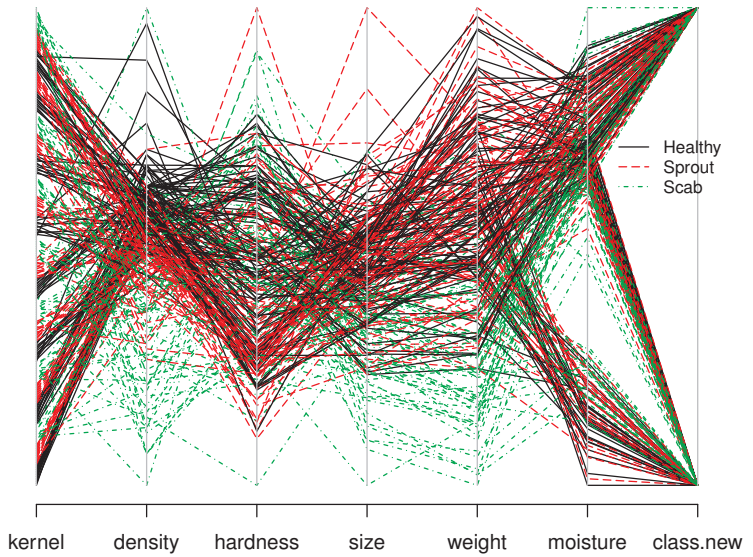
- 95% Wald confidence intervals for the odds ratio comparing to Scab ($j = 2$) to Healthy ($j = 1$)

```
> ci.OR2 <- exp(c.value * conf.beta[2:7, 1:2, 1])
> round(ci.OR2, 2)
          2.5 % 97.5 %
classsrw   0.14   1.92
density    0.03   0.13
hardness   0.37   1.12
size       0.80   3.55
weight     0.04   0.26
moisture   0.67   2.32
> round(data.frame(low = 1/ci.OR2[2, 2], up = 1/ci.OR2[2, 1]),
    2)
   low up
1 7.64 38
```

- Example interpretation: With 95% confidence, the odds of a scab instead of a healthy kernel change by 7.64 to 38.00 times when density is decreased by 0.13 holding the other variables constant.

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Parallel coordinates plot



- Response categories are ordered as
  category 1 < category 2 < $\cdots$ < category J
- Use cumulative probability to take advantage of ordering
  - $P(Y \leq j) = \pi_1 + \cdots + \pi_j$ for $j = 1, \ldots, J-1$
  - Note that $P(Y \leq J) = 1$
- Odds of $Y \leq j$:

$$\frac{P(Y \leq j)}{1 - P(Y \leq j)} = \frac{P(Y \leq j)}{P(Y > j)} = \frac{\pi_1 + \cdots + \pi_j}{\pi_{j+1} + \cdots + \pi_J}$$

- Relate log odds to explanatory variables:

$$\log\left(\frac{P(Y \leq j)}{1 - P(Y \leq j)}\right) = \text{logit}(P(Y \leq j)) = \beta_{j0} + \beta_1 x_1 + \cdots + \beta_p x_p$$

  for $j = 1, \ldots, J-1$
- Proportional odds regression model
  - Odds of $Y \leq j$: $P(Y \leq j)/P(Y > j) = \exp(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_p x_p)$
    $= \exp(\beta_{j0})\exp(\beta_1 x_1 + \cdots + \beta_p x_p)$

- Equivalent form of model:

$$P(Y \leq j) = \frac{\exp(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

- What is π_j?
  - $P(Y = j) = P(Y \leq j) - P(Y \leq j-1) =$

$$\frac{\exp(\beta_{j,0} + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_{j,0} + \beta_1 x_1 + \cdots + \beta_p x_p)} - \frac{\exp(\beta_{j-1,0} + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_{j-1,0} + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

    for $j = 2, \ldots, J-1$
  - For $j = 1$: $P(Y = 1) = P(Y \leq 1) - P(Y \leq 0) = P(Y \leq 1) - 0 =$

$$\frac{\exp(\beta_{10} + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_{10} + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

  - For $j = J$: $P(Y = J) = P(Y \leq J) - P(Y \leq J-1) =$

$$1 - \frac{\exp(\beta_{J-1,0} + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_{J-1,0} + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

- Maximum likelihood estimation
  - Likelihood function is simply the product of multinomial distributions for each observation with $\pi_1, \ldots, \pi_J$ as given on previous slide
  - Numerical iterative methods are used to determine regression parameter estimates

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Potential ordering: Scab ($Y = 1$) < Sprout ($Y = 2$) < Healthy ($Y = 3$)
- Need R to account for ordering

```
> levels(wheat$type)
[1] "Healthy" "Scab"    "Sprout"
> wheat$type.order <- factor(wheat$type, levels = c("Scab", "Sprout",
    "Healthy"))
> head(wheat, n = 3)
  class density hardness  size weight moisture    type type.order
1   hrw   1.349    60.33 2.303  24.65    12.02 Healthy    Healthy
2   hrw   1.287    56.09 2.726  33.30    12.17 Healthy    Healthy
3   hrw   1.234    43.99 2.512  31.76    11.88 Healthy    Healthy
> levels(wheat$type.order)
[1] "Scab"    "Sprout"  "Healthy"
> library(package = MASS)
> mod.fit.ord <- polr(formula = type.order ~ class + density +
    hardness + size + weight + moisture, data = wheat, method = "logistic")
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

```
> summary(mod.fit.ord)
Call:
polr(formula = type.order ~ class + density + hardness + size +
    weight + moisture, data = wheat, method = "logistic")

Coefficients:
           Value Std. Error t value
classsrw  0.1737    0.39176   0.443
density  13.5053    1.71301   7.884
hardness  0.0104    0.00593   1.752
size     -0.2925    0.41310  -0.708
weight    0.1272    0.03000   4.241
moisture -0.0390    0.08840  -0.441

Intercepts:
                Value  Std. Error t value
Scab|Sprout    17.572  2.246      7.824
Sprout|Healthy 20.044  2.340      8.568

Residual Deviance: 422.42
AIC: 438.42
```

- `polr()` estimates the model as $\text{logit}(P(Y \leq j)) = \beta_{j0} - \eta_1 x_1 - \cdots - \eta_p x_p$

Example: Wheat kernels (Wheat.R, Wheat.csv)

- The estimated model is

$$\begin{aligned}\text{logit}(\hat{P}(Y \leq j)) &= \hat{\beta}_{j0} - 0.17\texttt{srw} - 13.51\texttt{density} - 0.010\texttt{hardness} \\ &\quad + 0.29\texttt{size} - 0.13\texttt{weight} + 0.039\texttt{moisture}\end{aligned}$$

  where $\hat{\beta}_{10} = 17.57$ and $\hat{\beta}_{20} = 20.04$
- LRTs

```
> library(package = car)
> Anova(mod.fit.ord)
Analysis of Deviance Table (Type II tests)

Response: type.order
          LR Chisq Df Pr(>Chisq)
class          0.2  1      0.657
density       98.4  1    < 2e-16 ***
hardness       3.1  1      0.079 .
size           0.5  1      0.480
weight        19.0  1    1.3e-05 ***
moisture       0.2  1      0.659
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

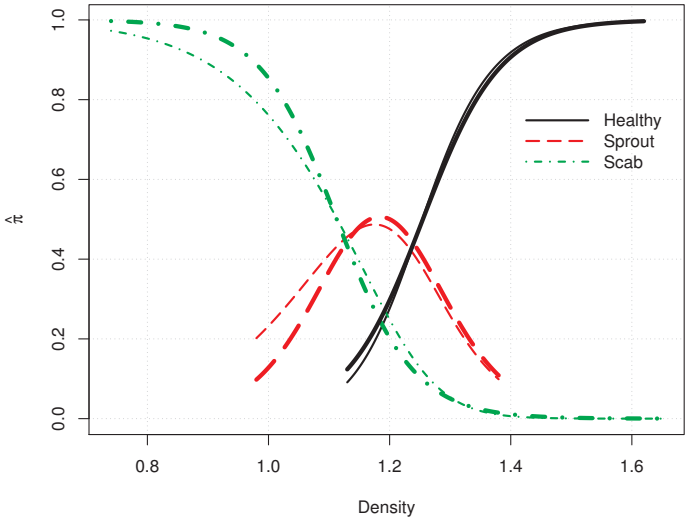
Example: Wheat kernels (Wheat.R, Wheat.csv)

- Estimated probabilities

```
> pi.hat.ord <- predict(object = mod.fit.ord, newdata = wheat,
    type = "probs")
> head(pi.hat.ord)
     Scab Sprout Healthy
1 0.03662 0.2739  0.6895
2 0.03352 0.2577  0.7088
3 0.08380 0.4362  0.4800
4 0.01694 0.1526  0.8304
5 0.11408 0.4900  0.3960
6 0.02875 0.2309  0.7404
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

- Estimate multinomial and proportional odds regression models
- One explanatory variable: `density`
- Thin line = multinomial, thick line = proportional odds



- Use odds ratios to interpret explanatory variables
- Compare two odds:
  - Proportional odds regression model with one explanatory variable: $\text{logit}(P(Y \leq j)) = \beta_{j0} + \beta_1 x$
  - Odds of $Y \leq j$ at x: $P(Y \leq j)/P(Y > j) = \exp(\beta_{j0} + \beta_1 x)$
  - Odds of $Y \leq j$ at $x + c$: $P(Y \leq j)/P(Y > j) = \exp(\beta_{j0} + \beta_1 (x + c))$
  - Odds ratio:
$$\frac{\exp(\beta_{j0} + \beta_1 (x + c))}{\exp(\beta_{j0} + \beta_1 x)} = \exp(c\beta_1)$$
- Interpretation:
  - The odds of $Y \leq j$ vs. $Y > j$ response change by $\exp(c\beta_1)$ times for every c-unit increase in x
    - Interpretation is the same for all $j = 1, \ldots, J-1$!
    - The odds of being below a particular response level change by $\exp(c\beta_1)$ times for every c-unit increase in x
  - If additional explanatory variables are in the model, add "holding the other variables constant"
- Odds ratios for more complicated models – Similar to logistic regression

- Profile LR and Wald intervals can be calculated
- R calculations
  - `confint()` calculates profile LR intervals (different than from multinomial regression)
  - `mcprofile` package cannot be used
  - `confint.default()` calculates Wald intervals
  - Odds ratios for explanatory variables from more complicated models – Need to program "by-hand" formulas into R

Example: Wheat kernels (Wheat.R, Wheat.csv)

- Estimate odds ratio for a c-unit increase in each explanatory variable: $\exp(c\beta_r)$ for the rth explanatory variable
- 95% profile LR confidence intervals for the odds ratio

```
> round(c.value, 2)
          density hardness     size   weight moisture
    1.00     0.13    27.36     0.49     7.92     2.03
> conf.beta <- confint(object = mod.fit.ord, level = 0.95)
> ci <- exp(c.value * (-conf.beta))  #Negative sign due to polr()
> round(data.frame(low = ci[, 2], up = ci[, 1]), 2)
          low   up
classsrw 0.39 1.81
density  0.11 0.26
hardness 0.55 1.03
size     0.77 1.72
weight   0.23 0.58
moisture 0.76 1.54
```


Example: Wheat kernels (Wheat.R, Wheat.csv)

```
> round(data.frame(low = 1/ci[2, 1], up = 1/ci[2, 2]), 2)
   low   up
1 3.87 9.36
```

- Example interpretation: With 95% confidence, the odds of kernel quality being below a particular level change by 3.87 to 9.36 times when density is decreased by 0.13, holding the other variables constant

1. Introduction
2. Analyzing a binary response, 2×2 tables
3. Analyzing a binary response, logistic regression
4. Analyzing a multicategory response
5. Analyzing a count response
   - Introduction
   - The Poisson distribution
   - Poisson regression models
   - Categorical explanatory variables
   - Poisson regression for contingency tables
   - Poisson regression for large contingency tables
   - Poisson regression with ordinal variables
   - Poisson rate regression
6. Model selection and evaluation
7. Models for correlated data
8. Conclusion

- Thus far all of the counts we have studied were Binomial counts
  - Summary of n identical Bernoulli trials
  - Each trial a success or a failure
  - Count is w = number of successes in n trials
  - $0 \leq w \leq n$
- Not all counts are of this form
  - Observe an event-generating process over fixed time/space/exposure
    - Number of cars crossing a bridge in an hour
    - Number of weeds in a plot of crop land
    - Number of moles on a person's body
- These counts are free to vary between 0 and no particular limit
- Counts of this type are often modeled using a *Poisson distribution*

- If Y is a Poisson RV, then the PMF of Y is

$$P(Y = y) = \frac{e^{-\mu}\mu^y}{y!}, \ y = 0, 1, 2, \ldots$$

  - $\mu > 0$ is a parameter
  - $E(Y) = Var(Y) = \mu$
  - Write $Y \sim Po(\mu)$
- μ is estimated using ML techniques
  - Likelihood is product of PMFs evaluated at sample counts, $y_1, \ldots, y_n$
- MLE for μ turns out to be $\hat{\mu} = \bar{y}$
  - $\widehat{Var}(\hat{\mu}) = \hat{\mu}/n$
- Can use these facts to develop tests and confidence intervals for μ
  - See our program Stoplight.R

- The Poisson distribution model assumes that the true mean count is the same for all observations
- In many cases, the potential mean count varies among subjects
  - Cars: different times of day
  - Weeds: different herbicide treatments
  - Moles: race, age, time in sun
- When there are explanatory variables that might relate to these changes, we can use a *Poisson regression model*

- Poisson regression models the mean as a function of explanatory variables $x_1, \ldots, x_p$
- Model $Y_i \sim Po(\mu_i)$, $i = 1, \ldots, n$, where

$$\mu_i = \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$$

  - The exp() guarantees that means are positive
  - Creates a curved relationship
    - Exponentially increasing or decreasing in x_j depending on the sign of β_j
  - Inverse is "log-linear" form, $\log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$

Interpretation of parameters in model

- $\log(\mu_i) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$ or
  $\mu_i = \exp(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) = e^{\beta_0} e^{\beta_1 x_{i1}} \ldots e^{\beta_p x_{ip}}$
- General interpretation is just line linear regression, except applied to log-mean
  - β_0 is the log mean of Y when all $x_j = 0$
    - Equivalently, $\exp(\beta_0)$ is the mean of Y when all $x_j = 0$
  - β_j is the change in log-mean when x_j increases by 1 unit, holding other variables constant
    - $\exp(\beta_j)$ is the multiplicative change in mean for 1 unit increase in x_j
    - $\exp(\beta_j)$ is the ratio of means at $x_j + 1$ vs. x_j, holding other variables constant
    - The percentage change in mean associated with a c-unit increase in x_j is $PC_j(c) = 100(\exp(c\beta_j) - 1)$

Parameters are estimated by ML estimation, as before

- No closed-form solution; use iterative numerical methods
- Resulting parameter estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$
  - Corresponding estimated variances $\widehat{Var}(\hat{\beta}_j)$, $j = 0, 1, \ldots, p$
- Inference on parameters and functions of parameters uses the usual ML techniques
  - Wald (easy to compute, but poor unless n is large)
  - LR (better, although still requires somewhat large n)

Example: Alcohol consumption[1] (AlcoholPoRegs.R, DeHartSimplified.csv)

- 100 "moderate to heavy" drinkers ($\geq$ 12/week for F, $\geq$ 15/week for M)
- Recorded various psychological scales relating to life events and self-esteem
- Maintained diary of #drinks/day for a month
  - Data we use are from first Saturday in study (89 participants)
- Researchers hypothesize that a higher negative life events score results in increasing alcohol consumption
  - "Drown your sorrows"
- Modeling
  - Y = number of drinks consumed (`numall`)
  - x_1 = index for number and intensity of negative events (`negevent`)

[1] Data kindly provided by Dr. Steve Armeli, School of Psychology, Fairleigh Dickinson University. See (DeHart et al., 2008).

Example: Alcohol consumption (AlcoholPoRegs.R, DeHartSimplified.csv)

```
> dehart <- read.table("C:\\Data\\DeHartSimplified.csv", header = TRUE,
    sep = ",", na.strings = " ")
> # Reduce data to what is needed for examples
> saturday <- dehart[dehart$dayweek == 6, c(1, 4, 7, 8)]
> head(round(x = saturday, digits = 3))
   id numall negevent posevent
1   1      9    0.400    0.525
11  2      4    2.377    0.924
18  4      1    0.233    1.346
24  5      0    0.200    1.500
35  7      2    0.000    1.633
39  9      7    0.550    0.625
> dim(saturday)
[1] 89  4
```


Example: Alcohol consumption (AlcoholPoRegs.R, DeHartSimplified.csv)

```
> # Fit model of Drinks vs. Neg Events
> mod.neg <- glm(formula = numall ~ negevent, family = poisson(link = "log"),
    data = saturday)
> summary(mod.neg)

Call:
glm(formula = numall ~ negevent, family = poisson(link = "log"),
    data = saturday)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.985  -1.356  -0.275   0.474   5.885

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.5205     0.0752   20.21   <2e-16 ***
negevent     -0.2612     0.1360   -1.92    0.055 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 250.34  on 88  degrees of freedom
Residual deviance: 246.39  on 87  degrees of freedom
AIC: 505.8
```

Example: Alcohol consumption (AlcoholPoRegs.R, DeHartSimplified.csv)

- Fitted model: $\log(\hat{\mu}_i) = 1.52 - 0.26\texttt{negevent}_i$
  - Corresponding standard errors
    - 0.075 for $\hat{\beta}_0$
    - 0.14 for $\hat{\beta}_1$
- Note that this is in the opposite direction of the hypothesis
  - Preliminary analysis based on one day
  - Other potentially important variables not considered

- Estimated means (predicted values):

$$\hat{\mu}_i = \exp(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \ldots + \hat{\beta}_p x_{ip})$$

- Ratios of means
  - Estimated ratio of means at $x_j + c$ vs. x_j, holding other variables constant:

$$\exp(c\hat{\beta}_j)$$

  - Expressed as a percentage change:

$$\widehat{PC}_j(c) = 100(\exp(c\hat{\beta}_j) - 1)$$

- Tests and confidence intervals can be done by LR or Wald
  - Usual caveats about sample sizes with Wald
  - LR inference is available using `mcprofile()`
    - `confint()` method for `glm` does profile LR for individual parameters

Example: Alcohol consumption (AlcoholPoRegs.R, DeHartSimplified.csv)

- Find the percent change in # drinks for 1 unit increase in negative event index,

$$\widehat{PC}_1(1) = 100(\exp(-0.26118) - 1)$$

```
> 100 * (exp(mod.neg$coefficients[2]) - 1)
negevent
  -22.99
> beta1.int <- confint(mod.neg, parm = "negevent", level = 0.95)
> 100 * (exp(beta1.int) - 1)
  2.5 %  97.5 %
-41.529  -0.348
```


Example: Alcohol consumption (AlcoholPoRegs.R, DeHartSimplified.csv)

- LRT of significance for terms in the model using `Anova()` from `car` package

```
> library(car)
> Anova(mod.neg)
Analysis of Deviance Table (Type II tests)

Response: numall
          LR Chisq Df Pr(>Chisq)
negevent     3.95  1      0.047 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- P-value is 0.047, offering some evidence against $H_0 : \beta_1 = 0$
  - Differs from Wald test p-value (0.055)
  - Both say the same thing unless strict 0.05 significance level is used!

- Categorical explanatory variables are represented in Poisson regression the same way as in logistic regression
  - For X with I levels, create $I-1$ indicator variables, $x_2, x_3, \ldots, x_I$
    - For now, use i to index the levels
  - Fit loglinear model,

$$\log(\mu) = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

Parameter interpretation:

- Indicator variables for a 4-level categorical explanatory variable

Level	x_2	x_3	x_4	Log-Mean
1	0	0	0	$\log(\mu_1) = \beta_0$
2	1	0	0	$\log(\mu_2) = \beta_0 + \beta_2$
3	0	1	0	$\log(\mu_3) = \beta_0 + \beta_3$
4	0	0	1	$\log(\mu_4) = \beta_0 + \beta_4$

  - β_0 is the log-mean for level 1 of X
  - β_i is the difference in log-means between levels i and 1, $i = 2, 3, \ldots, I$
    - Sometimes called the "effect" of level i
    - $\mu_i/\mu_1 = \exp(\beta_i)$ (from $\beta_i = \log(\mu_i) - \log(\mu_1)$)
    - $\mu_i/\mu_{i'} = \exp(\beta_i - \beta_{i'})$ (from $\log(\mu_i) - \log(\mu_{i'}) = \beta_i - \beta_{i'}$)

- Inference
  - Test of equality of all I means
    - $H_0 : \mu_1 = \mu_2 = \ldots = \mu_I$ is equivalent to $H_0 : \beta_2 = \ldots = \beta_I = 0$
    - LRT compares log likelihoods for full fitted model and null model (one mean)
    - Compare to χ^2_{I-1}
  - Confidence intervals for individual means
    - $\hat{\mu}_i = \exp(\hat{\beta}_0 + \hat{\beta}_i)$, so LR or Wald intervals can be used
  - Confidence intervals for ratios of means
    - $\hat{\mu}_i/\hat{\mu}_{i'} = \exp(\hat{\beta}_i - \hat{\beta}_{i'})$, so LR or Wald intervals can be used
    - Use ratios rather than differences due to exponential structure for means
- See our program BirdCountPoReg.R

- **Now consider *two* categorical variables, X with I levels and Z with J levels**
- At each combination of X and Z observe a count, y_{ij}, $i = 1, \ldots, I$; $j = 1, \ldots, J$
- Represent cross-tabulation of counts in a *contingency table:*

		Z				
		1	2	$\cdots$	J	Total
X	1	y_{11}	y_{12}	$\cdots$	y_{1J}	y_{1+}
	2	y_{21}	y_{22}	$\cdots$	y_{2J}	y_{2+}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
	I	y_{I1}	y_{I2}	$\cdots$	y_{IJ}	y_{I+}
	Total	y_{+1}	y_{+2}	$\cdots$	y_{+J}	$y_{++} = n$

- Typical interests in the analysis of a contingency table:
  - Are X and Z "independent"?
    - Independence means that the conditional distribution of X is the same for each level of Z and vice versa
    - Expect counts within each row follow the same proportional pattern e.g., 10, 20, 40 / 5, 10, 20
    - Expect counts within each column follow the same proportional pattern
  - Sometimes equally important: describe the nature of any association
- Two general approaches to analysis of a contingency table:
  - "Usual" tests of independence
  - Loglinear model
- Both can begin from the assumption that $Y_{ij} \sim Po(\mu_{ij})$
- We discuss the usual testing approach first

- Null hypothesis is that X and Z are "independent"
- Alternative hypothesis is simply that they are not independent
- Under independence, expected cell count is $\hat{\mu}_{ij} = y_{i+}y_{+j}/n$
- Test Statistics
  - Pearson statistic

$$X^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(y_{ij}-\hat{\mu}_{ij})^2}{\hat{\mu}_{ij}}$$

  - LRT statistic

$$-2\log(\Lambda) = 2\sum_{i=1}^{I}\sum_{j=1}^{J} y_{ij}\log\left(\frac{y_{ij}}{\hat{\mu}_{ij}}\right)$$

- Under independence, both statistics have large-sample $\chi^2_{(I-1)(J-1)}$ distributions
  - Extreme values of the test statistic relative the $\chi^2_{(I-1)(J-1)}$ distribution indicate evidence against independence
- Pearson tends to perform a little better than LR in smaller samples

Example: HIV vaccine (HIVvaccinePoisson.R)

```
> c.table <- array(data = c(51, 74, 8146, 8124), dim = c(2,
    2), dimnames = list(Trt = c("vaccine", "placebo"), Response = c("HIV",
    "No HIV")))
> c.table
         Response
Trt       HIV No HIV
  vaccine  51   8146
  placebo  74   8124
> ind.test <- chisq.test(x = c.table, correct = FALSE)
> ind.test

Pearson's Chi-squared test

data:  c.table
X-squared = 4.262, df = 1, p-value = 0.03898
> ind.test$expected
         Response
Trt        HIV No HIV
  vaccine 62.5   8135
  placebo 62.5   8135
```

Example: HIV vaccine (HIVvaccinePoisson.R)

```
> library(package = vcd)
> assocstats(x = c.table)
                    X^2 df P(> X^2)
Likelihood Ratio 4.2859  1 0.038430
Pearson          4.2617  1 0.038981

Phi-Coefficient   : 0.016
Contingency Coeff.: 0.016
Cramer's V        : 0.016
```


Poisson regression approach to the same problem

- Indicator variables $x_2, \ldots, , x_I$ for X and $z_2, \ldots, z_J$ for Z
- Use together in a loglinear model as before:

$$\log(\mu) = \beta_0 + \beta_2^X x_2 + \beta_3^X x_3 + \ldots + \beta_I^X x_I + \beta_2^Z z_2 + \beta_3^Z z_3 + \ldots + \beta_J^Z z_J$$

  - Abbreviate this model for convenience:

$$\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Z, \; i = 1, \ldots, I, \; j = 1, \ldots, J,$$

    where is it implicit that $\beta_1^X = \beta_1^Z = 0$
  - Often referred to as a "loglinear model" for a contingency table

It can be shown that this model represents the cell counts under independence!

- Main effects in the loglinear model cause marginal totals to be modeled perfectly
  - β_i^X's allow the marginal totals in the table to vary across rows
  - β_j^Z's allow the marginal totals in the table to vary across columns
- *There are no parameters that allow individual cell counts to deviate from these patterns*
  - Predicted cell counts from this model are the same as expected cell counts under independence

- Adding interaction terms $\beta_{ij}^{XZ} x_i z_j$ to the model allows the row mean-ratios to differ among columns and vice versa
- Augmented model is

$$\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Z + \beta_{ij}^{XZ},$$

where β_{ij}^{XZ} is the parameter corresponding to $x_i z_j$, $i = 1, \ldots, I$; $j = 1, \ldots, J$

  - This is a *saturated* model, because there are as many parameters as there are counts to be modeled
  - Estimated cell means match observed counts perfectly

Using saturated model to test independence

- Note that null hypothesis of independence implies a *smaller* (reduced) model.
  - Fit each, compute residual deviances
  - LRT stat is difference between residual deviances
  - Compare to $\chi^2_{(I-1)(J-1)}$
  - Can be done in R using either `anova()` or `Anova()` from `car` package
- This LRT stat is exactly the same as LRT stat on contingency table!

Measuring association

- In a contingency table, we use *odds ratios* to describe association:

		Z	
		1	2
X	1	μ_{11}	μ_{12}
	2	μ_{21}	μ_{22}

$$OR_{12,12} = \frac{\mu_{11}\mu_{22}}{\mu_{21}\mu_{12}}$$

- More generally,

$$OR_{ii',jj'} = \frac{\mu_{ij}\mu_{i'j'}}{\mu_{i'j}\mu_{ij'}}$$

where i, i' are any two rows and j, j' are any two columns

- Odds ratios are easily computed from the loglinear model
- Start from $\log(OR_{ii',jj'})$
  - $\log(OR_{12,12}) = \log(\mu_{11}) + \log(\mu_{22}) - \log(\mu_{21}) - \log(\mu_{12})$
  - Apply model, $\log(\mu_{ij}) = \beta_0 + \ldots$ and simplify
- Can use this to show that
  - $OR_{ii',jj'} = 1$ for all rows/columns in independence model
  - $OR_{ii',jj'} = \exp(\beta^{XZ}_{ij} + \beta^{XZ}_{i'j'} - \beta^{XZ}_{i'j} - \beta^{XZ}_{ij'})$ in saturated model
- Estimates just plug in estimated parameters
- LR confidence intervals from `mcprofile()`

Example: HIV vaccine (HIVvaccinePoisson.R)

```
> all.data <- as.data.frame(as.table(c.table))
> all.data
      Trt Response Freq
1 vaccine      HIV   51
2 placebo      HIV   74
3 vaccine   No HIV 8146
4 placebo   No HIV 8124
> levels(all.data$Trt)
[1] "vaccine" "placebo"
>
> M1 <- glm(formula = Freq ~ Trt * Response, family = poisson(link = "log"),
    data = all.data)
> summ1 <- summary(M1)
> c(summ1$deviance, summ1$df.residual)
[1] -3.531e-13  0.000e+00
```

- Model is saturated (perfect fit)
  - 0 residual deviance
  - 0 residual df

Example: HIV vaccine (HIVvaccinePoisson.R)

```
> round(summ1$coefficients, digits = 4)
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                 3.9318     0.1400  28.079   0.0000
Trtplacebo                  0.3722     0.1820   2.045   0.0408
ResponseNo HIV              5.0735     0.1405  36.119   0.0000
Trtplacebo:ResponseNo HIV  -0.3749     0.1827  -2.053   0.0401
```

- Note that indicator variables ignore first alphabetical level (`vaccine`, `HIV`)
- Estimated Poisson regression model under dependence:

$$\begin{aligned}\log(\hat{\mu}) &= 3.93 + 0.37\texttt{Trtplacebo} + 5.07\texttt{ResponseNo HIV} \\ &\quad -0.37\texttt{Trtplacebo} \times \texttt{ResponseNo HIV}\end{aligned}$$

- $\widehat{OR} = \exp(\beta_{11}^{XZ} + \beta_{22}^{XZ} - \beta_{21}^{XZ} - \beta_{12}^{XZ}) = \exp(-0.37) = 0.68$
  (all parameters involving first level are zero)
  - Odds of No HIV for someone on placebo are 0.68 times as high as the odds of No HIV for someone who was vaccinated
  - Confidence interval given in the program

Example: HIV vaccine (HIVvaccinePoisson.R)

```
> library(car)
> Anova(M1)
Analysis of Deviance Table (Type II tests)

Response: Freq
             LR Chisq Df Pr(>Chisq)
Trt                0  1      0.994
Response       21260  1     <2e-16 ***
Trt:Response       4  1      0.038 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- LRT results are the same as earlier!

Extension #1: Modeling more than 2 categorical variables

- p categorical variables form a *p-way table*
- Goal is to learn about associations among variables
  - Which pairs are associated, which are not?
  - Are associations between two variables consistent across levels of other variables, or do they change?
    - i.e., does the value of a particular odds ratio depend on levels of another variable?

Parameters in models

- Main effects adjust marginal proportions
  - These appear in all models, since we are not testing margins
  - One set for each variable (e.g., β_i^X, $i = 1, \ldots, I$)
  - Ensures that estimated marginal totals equal observed
- 2-way interactions permit associations (ORs $\neq 1$) between the two variables
  - These appear in any models where we want to allow association between that pair
  - One set for each pair (e.g., β_{ij}^{XZ}, $i = 1, \ldots, I$; $j = 1, \ldots, J$)
  - Not including an interaction term forces all ORs to be 1 between those two variables

Parameters in models, continued

- 3-way interactions allow 2-way associations to change across levels of the third variable
  - When these appear, model should also include sets of parameters for all pairs
  - One set for any triple to be considered (e.g., β_{ijk}^{XZW}, $i = 1, \ldots, I$; $j = 1, \ldots, J$, $k = 1, \ldots, K$)
  - Allows each $OR_{ii',jj'}^{XZ}$ to change depending on the level of W
    - Similarly XW ORs change across Z and ZW ORs change across X
- See example in PolIdeolNominal.R

Extension #2: Modeling ordinal categorical variables

- Assign numerical scores to levels and treat as a numerical variable
  - e.g., let X have I ordered levels with scores $s_1, \ldots, s_I$
  - Use scores as a numerical variable
- If there is an additional nominal categorical variable Z with J levels:
  - Ordinal "Linear Association" Model: $\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Z + \beta_j^{XZ} s_i$
  - Main effect of *each* variable is nominal!
  - Different "slopes" on ordinal variable for each level of nominal
  - Association: $\log(OR_{ii',jj'}) = (\beta_j^{XZ} - \beta_{j'}^{XZ})(s_i - s_{i'})$
    - Log of mean ratios changes linearly with the difference between category scores
  - See example in PolIdeolOrd.R

New question: what if counts are based on different amounts of sampling effort?

- Counts are accumulated over specific amounts of time or space
  - Count of cars over a bridge increases with time spent watching
  - Count of auto accidents increases with miles driven
  - Count of weeds increases if you increase the size of the plots
- This sampling effort is called *exposure*
- All analyses and examples thus far have assumed that mean counts have common exposure
  - Means vary only due to changes in the *intensity* (or *rate*) at which events accumulate.

- In many problems, exposures are different
  - Accumulate events at rate r over exposure t
  - Count = Rate * Exposure ($\mu = r * t$)
- Counts may change because of changing exposure, even when the rates are the same
  - Interferes with our comparison of rates
- Want to model rates (unknown) not exposures (assumed known)
  - Rate = Count / Exposure, or $r = \mu/t$

- When we model counts, we need to account for different exposures, so that we can focus on how **the *rates*** relate to explanatory variables:

$$\log(r_i) = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$$

- Since $r_i = \mu_i / t_i$, this model can be fitted as a Poisson regression model, $Y_i \sim Po(\mu_i)$ with

$$\log(\mu_i) = \log(t_i) + \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi},\ i = 1, \ldots, n.$$

  - This is called a *Poisson Rate Regression* model
  - Note that $\log(t_i)$ has no parameter. It is a known adjustment factor, called an *offset*.
    - Also known as *Poisson Regression with Offsets*
  - *Parameters relate to log-rates rather than log-means*

Example: Beetle Egg Crowding Experiment (BeetleEggCrowding.R, BeetleEggCrowding.txt)

- Experiment to see how crowding and temperature affects eggs laid per female of certain beetle
  - Boxes with 1 or 5 females (`TRT`)
  - Held in chambers at 21C or 24C (`Temp`)
  - Count eggs laid over prescribed period (`NumEggs`)
    - Total per box, subject to different exposures (`females`)
    - Want *rate* per female
- Fit Poisson rate regression model
  - `formula = NumEggs ~ TRT*Temp`
  - Add argument `offset = log(females)`
  - Run analysis as usual
- Parameters and tests relate to comparing rates—eggs/female—across treatments and temperatures
- See program for details

1 Introduction
2 Analyzing a binary response, 2×2 tables
3 Analyzing a binary response, logistic regression
4 Analyzing a multicategory response
5 Analyzing a count response
6 Model selection and evaluation
- Introduction
- Variable selection
- Residual analysis
- Goodness-of-fit statistics
- Overdispersion

7 Models for correlated data
8 Conclusion

- "Model selection" has two parts
  - Identify appropriate probability model for problem
    - e.g., binomial-logistic, Poisson-loglinear
    - Already discussed these
  - Identify appropriate set of explanatory variables
    - Often measure more than are needed
- We first present methods to select an appropriate set of explanatory variables from among a larger pool of candidate variables
  - Focus on information criteria rather than hypothesis tests
- Once the variables in the model are fully specified, we check assumptions
  - Residual diagnostics
  - Goodness of fit statistics
- Finally, we explore a common model assumption violation—*overdispersion*—and offer some solutions

Variable Selection

- *Variable selection* means reducing the number of explanatory variables that get used in a model

Why do it?

- Sometimes $p > n$ and full model can't be fit at all
- Complex models are harder to understand
  - Seek a "parsimonious" description
- *Bias-Variance Trade-off*
  - Models that are too small may predict poorly because they are missing important features
    - "Bias"
  - Models that are too large may predict poorly because unnecessary parameter estimates add noise
    - "Variance"
  - Often a compromise between bias and variance is necessary

Information Criteria

- *Information criteria* are measures based on the log likelihood that include a "penalty" for each parameter estimated by the model
  - Adding variables to a model improves the likelihood but also increases the penalty
  - Can result in either a better or a worse value of the criterion
- General form is $IC(k) = -2\hat{l} + kr$
  - $\hat{l}$ is the maximized log-likelihood for the model
  - r is the number of parameters in the model
  - k is the *penalty coefficient*
  - Smaller $IC(k)$ values are better
    - High (log-) likelihood
    - Small number of parameters

- The three most common information criteria are:
  - Akaike's Information Criterion:

$$\text{AIC} = IC(2) = -2\hat{l} + 2r$$

  - Corrected AIC:

$$\text{AICc} = IC(2n/(n-r-1)) = -2\hat{l} + \frac{2n}{n-r-1}r = \text{AIC} + \frac{2r(r+1)}{n-r-1}$$

  - Bayesian Information Criterion

$$\text{BIC} = IC(\log(n)) = -2\hat{l} + \log(n)r$$

- AIC uses smallest penalty, allows most variables into model
- BIC *almost always* has largest penalty, allows fewest variables into model

- Using information criteria (traditional approach)
  1. Fit a set of models
  2. Choose k and compute $IC(k)$ on each model
  3. Select model with smallest $IC(k)$
- Does not require nested models

Which models to compare?

- Suppose we have a pool of P explanatory variables
  - May include transformations and interactions
- **Use expert-selected models where possible**
- Otherwise, use an algorithm to select and fit models
  - Many different algorithms exist
  - *All-subsets regression* fits all possible models and computes an $IC(k)$ on each
    - There are 2^P possible models to consider (the *model space*)
    - $P = 20$: over 1 million
    - $P = 30$: over 1 billion

Example: Placekicking (AllSubsetsPlacekick.R, Placekick.csv)

- We use the `glmulti()` function from the `glmulti` package to perform all-subsets regression
  - `method = "h"` does all subsets (may run slow!)
  - `level = 1` uses main effects, `level = 2` includes interactions
  - The package is described in more detail by Calcagno and de Mazancourt (2010)

```
> placekick <- read.table(file = "C:\\data\\Placekick.csv", header = TRUE,
    sep = ",")
> head(placekick)
  week distance change elap30 PAT type field wind good
1    1       21      1  24.72   0    1     1    0    1
2    1       21      0  15.85   0    1     1    0    1
3    1       20      0   0.45   1    1     1    0    1
4    1       28      0  13.55   0    1     1    0    1
5    1       20      0  21.87   1    0     0    0    1
6    1       25      0  17.68   0    0     0    0    1
> # Sometimes have to deactivate a system variable in order to
> # get rJava to work.
> if (Sys.getenv("JAVA_HOME") != "") Sys.setenv(JAVA_HOME = "")
```


Example: Placekicking (AllSubsetsPlacekick.R, Placekick.csv)

```
> library(glmulti)
> # Using AICc as criterion. Could use crit = 'bic' or 'aic'
> # instead.  Using 'good ~ .' to include all variables from
> # data (other than 'good')
> search.1.aicc <- glmulti(y = good ~ ., data = placekick, plotty = FALSE,
    report = FALSE, fitfunction = "glm", family = binomial(link = "logit"),
    level = 1, method = "h", crit = "aicc")
>
> print(search.1.aicc)
glmulti.analysis
Method: h / Fitting: glm / IC used: aicc
Level: 1 / Marginality: FALSE
From 100 models:
Best IC: 766.728784139471
Best model:
[1] "good ~ 1 + distance + change + PAT + wind"
Evidence weight: 0.0669551501665866
Worst IC: 780.47950528365
12 models within 2 IC units.
51 models to reach 95% of evidence weight.
```


Example: Placekicking (AllSubsetsPlacekick.R, Placekick.csv)

- Summary of best 5 models

```
> aa <- weightable(search.1.aicc)
> cbind(model = aa[1:5, 1], round(aa[1:5, 2:3], digits = 3))
                                                model  aicc weights
1        good ~ 1 + distance + change + PAT + wind 766.7   0.067
2 good ~ 1 + week + distance + change + PAT + wind 767.1   0.055
3        good ~ 1 + week + distance + change + PAT 767.3   0.050
4               good ~ 1 + distance + change + PAT 767.4   0.049
5                 good ~ 1 + distance + PAT + wind 767.7   0.041
```

- All top models use `distance`, `PAT`
  - 4 use `change`
  - 3 use `WIND`
  - 2 use `week`
  - None use `elap30`, `type`, `field`
- Some clear patterns, but some uncertainty

- Choosing a single best model fails to acknowledge "model uncertainty"
  - We estimate parameters of the chosen model as if it were the only one we considered
  - Excluded parameters are implicitly 0 *with 0 standard error!*
- Often there are many, many models with $IC(k)$ values similar to the best (e.g., within 2 units)
  - With a small change in one or two observations, a different model could have been selected!
  - We saw this in the placekicking example
    - "`12 models within 2 IC units.`"
- Models near the top tend to have some variables in common
  - Can we use this somehow?

- Instead of selecting one model, can use use them all to evaluate the relative importance of all variables
- Transform $IC(k)$ into "evidence weight" for each model using "softmax transformation"
  - Evidence weights lie between 0-1
  - Evidence weights sum to 1 across all models
    - Therefore they resemble a probability distribution
    - If BIC is used, then they *are* approximate posterior probabilities that the model is "correct"
    - A value close to 1 means that data strongly support that model
- Use *model* evidence weights to create *variable* evidence weights
  - Add up evidence weights from all models in which a variable appears
    - Often in top models ⇒Evidence weight near 1
    - Never in top models ⇒Evidence weight near 0
- Variables with high evidence weights are likely to be important
- This is called *(Bayesian) model averaging* (BMA)

Example: Placekicking (AllSubsetsPlacekick.R, Placekick.csv)

- Recall top 5 models from all-subsets:

```
> cbind(model = aa[1:5, 1], round(aa[1:5, 2:3], digits = 3))
                                                model  aicc weights
1        good ~ 1 + distance + change + PAT + wind 766.7   0.067
2 good ~ 1 + week + distance + change + PAT + wind 767.1   0.055
3        good ~ 1 + week + distance + change + PAT 767.3   0.050
4               good ~ 1 + distance + change + PAT 767.4   0.049
5                 good ~ 1 + distance + PAT + wind 767.7   0.041
```

- Note the "Evidence Weights"
  - None are large ("best" model is only 0.067)
  - Many common variables in top 5 models
    - Presumably also in other top models

Example: Placekicking (AllSubsetsPlacekick.R, Placekick.csv)

- Variable weights are extracted using `coef()`

```
> parms <- coef(search.1.aicc)
> # Renaming columns to fit in display
> colnames(parms) <- c("Estimate", "Variance", "n.Models", "Evidence Wt",
    "95%CI +/-")
> round(parms, digits = 3)
            Estimate Variance n.Models Evidence Wt 95%CI +/-
elap30         0.001    0.000       46       0.286     0.008
field         -0.023    0.007       46       0.288     0.163
type           0.056    0.015       46       0.326     0.242
week          -0.012    0.000       48       0.464     0.034
wind          -0.315    0.130       52       0.583     0.708
change        -0.223    0.052       61       0.641     0.447
PAT            1.262    0.154       64       0.992     0.770
(Intercept)    4.734    0.284      100       1.000     1.045
distance      -0.087    0.000      100       1.000     0.022
```

- STRONG evidence for `distance`, `PAT` (ignore `Intercept`)
- *Slight* evidence for `change`, `wind`
- Others not well supported, but not worthless (near 0)

- All subsets is fine when P is not too large (e.g., < 30)
- When 2^P is just too big, need some way to efficiently filter out bad models and identify good ones
- Many ways to do this
  - Genetic search algorithm (compatible with model averaging)
  - Stepwise (selects a single model)
  - LASSO (selects a single model)

Genetic Search Algorithm (GA)

- Genetic Algorithm explores a model space without investigating every model.
  - Randomness allows it to explore broad variety of models
  - Smart algorithm focuses on models with good IC values
  - End with a set of, say, 100 models that are the best ever created
- No guarantee that GA finds the very best model
  - Generally identifies many of the best models compared to an exhaustive search
  - Best variables likely to be well represented
- Available in `glmulti(...,method = "g",...)`

Stepwise procedures build models one variable at a time

- Forward stepwise: Start with no variables, and successively add variables that improve likelihood the most
- Backward elimination: Start with *all* variables, and successively delete variables that reduce likelihood the least
- Select the *one* model that has the lowest $IC(k)$
  - Don't use hypothesis tests or "significance" to decide!
- See program StepwisePlacekick.R

LASSO ("**L**east **A**bsolute **S**election and **S**hrinkage **O**perator")

- Information criteria add a penalty to the maximized log-likelihood
- LASSO adds penalty to the likelihood *during estimation*
  - Penalty is based on the sizes of the parameter estimates

$$\log(L(\beta_0, \beta_1, \ldots, \beta_p | y, \ldots, y_n)) - \lambda \sum_{j=1}^{p} |\beta_j|$$

  - Parameter estimates are "shrunk" toward 0 to balance between improving the likelihood and increasing the penalty
  - Often performs better at prediction than stepwise
- Issues:
  - Tends to retain some variables with uselessly tiny coefficients
  - Inference procedures (e.g., confidence intervals) still being developed
- See program LASSOPlacekick.R

Checking the Fit of a Model (Single-count models only)

- In linear regression, residuals, $e_i = y_i - \hat{y}_i$ are used to identify model assumptions that may be inappropriate for the data
  - Used in plots
    - Identify poor mean fit (e.g., curvature where it is not expected)
    - Identify non-constant variance
    - Identify outliers and influential observations
  - Standardized versions $r_i = e_i / \sqrt{\widehat{Var}(e_i)}$ also used
- We can use residuals similarly for all of our count models
  - Need to define residuals in context of each model
  - Need to identify assumptions to be checked

Residuals for Count-data Models

- Start by expressing all responses and predicted values as *counts*
  - Already done in Poisson regression
  - In Poisson rate regression, multiply predicted rates by observed exposures
  - In logistic regression, aggregate data into *explanatory variable pattern form*
    - If there are multiple observed counts with the same combination of explanatory variables—"explanatory variable pattern" (EVP)—pool the successes and trials into one total
    - e.g., in Placekicking example with distance as only variable, have one count of trials and successes at each distance
    - Refit the model to aggregated data
  - So we now have $y_m, \hat{y}_m$ for $m = 1, \ldots, M$, where M is the number of EVPs (or total sample size in Poisson count models)
    - n_m is the number of trials in binomial

Standardized Pearson Residuals

$$r_m = \frac{y_m - \hat{y}_m}{\sqrt{\widehat{Var}(Y_m - \hat{Y}_m)}}, \; m = 1, \ldots, M$$

- Approximately standard normal when model is correct, particularly when the $\hat{y}_m$ are large
  - $\hat{y}_m$ around 5 or more
  - Also want $n_m - \hat{y}_m$ at least 5 in the binomial case
- $\approx$5% of r_m's beyond ± 2, rarely beyond ± 3, never beyond ± 4
- Unstandardized Pearson residuals also exist, but standardized version is better for diagnostics
- There are also versions of residuals based on an observation's contribution to the residual deviance (defined later)
  - Standardized and unstandardized versions
  - Also can be compared to 2-3-4 guidelines

Residual Plots

- Can use either Pearson or deviance versions (preferably standardized)
- r vs. $\hat{y}$ (Poisson) or r vs. $\hat{\pi}$ (binomial)
  - If model fits, should show
    - No serious fluctuations in the mean value (e.g., no curvature).
    - Roughly constant variance
    - $\approx$5% of points beyond ± 2, rarely any beyond ± 3
  - Curvature suggests the log or logit specification needs to be changed
    - Plotting r vs. the *linear predictor*, $\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p$, can help diagnose how it needs to be changed.
  - Changing variance indicates possible flaw in probability model
    - Model specifies a different relationship between variance and mean than the data possess
  - Points beyond thresholds
    - "One or two" past ± 3: outliers
    - Many: Overdispersion (see later)

- r vs. each x_j
  - Curvature for a particular plot suggests that a transformation or additional polynomial terms are needed *for that variable*
  - If many plots show curvature, possibly the log or logit specification needs to be changed

Warning: With discrete data, sometimes residuals have very strange distributions

- In binomial with $n_m = 1$ trial, only two possible numerators: $0 - \hat{y}_m$ and $1 - \hat{y}_m$
  - These form two bands on a residual plot
- In Poisson when $\hat{y}_m$ is very small (< 1 or so) most observations are 0
  - Very few likely response values, again results in bands
  - Anything other than the majority response may have extreme residual
- In binomial models, extreme residuals can occur normally near $\hat{\pi} = 0$ or 1.
  - Nothing wrong...it just happens, esp. with low trials
  - e.g., 1/1 field goal at a very long distance when all others at similar distances miss
- Don't take the 2-3-4 thresholds too literally!
  - Can't learn much from residuals in these cases
  - Plots difficult to decipher

Example: Placekicking (PlacekickDiagnostics.R, Placekick.csv)

- Show residual plots for model with `distance` only
- Assuming that binary data are aggregated into explanatory variable pattern form

```
> head(placekick)
  week distance change elap30 PAT type field wind good
1    1       21      1  24.72   0    1     1    0    1
2    1       21      0  15.85   0    1     1    0    1
3    1       20      0   0.45   1    1     1    0    1
4    1       28      0  13.55   0    1     1    0    1
5    1       20      0  21.87   1    0     0    0    1
6    1       25      0  17.68   0    0     0    0    1
> head(w.n)
  distance success trials   prop
1       18       2      3 0.6667
2       19       7      7 1.0000
3       20     776    789 0.9835
4       21      19     20 0.9500
5       22      12     14 0.8571
6       23      26     27 0.9630
```


Example: Placekicking (PlacekickDiagnostics.R, Placekick.csv)

- Refitting model to aggregated data gives same parameter estimates as before

```
> mod.fit.bin <- glm(formula = success/trials ~ distance, weights = trials,
    family = binomial(link = logit), data = w.n)
> summary(mod.fit.bin)$coefficients
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)     5.812   0.326277   17.81 5.570e-71
distance       -0.115   0.008339  -13.79 2.781e-43
```

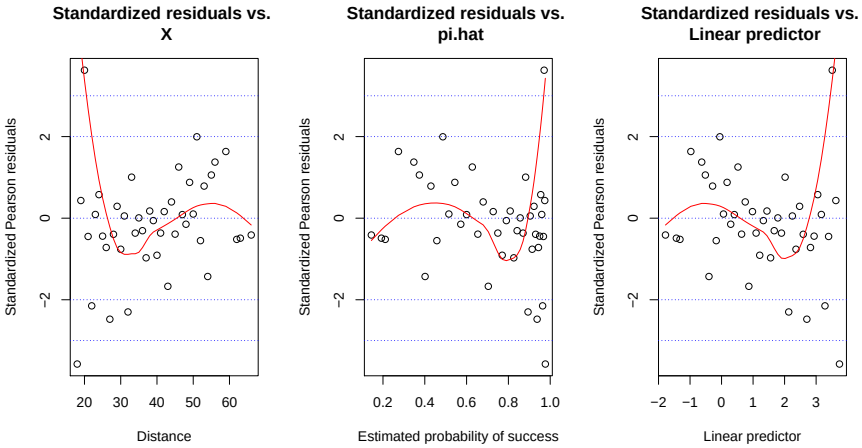
Example: Placekicking (PlacekickDiagnostics.R, Placekick.csv)

- Compute Pearson and standardized Pearson residuals, estimated probabilities and linear predictor

```
> pi.hat <- predict(mod.fit.bin, type = "response")
> p.res <- residuals(mod.fit.bin, type = "pearson")
> s.res <- rstandard(mod.fit.bin, type = "pearson")
> lin.pred <- mod.fit.bin$linear.predictors
> w.n <- data.frame(w.n, pi.hat, p.res, s.res, lin.pred)
> round(head(w.n), digits = 3)
  distance success trials  prop pi.hat  p.res  s.res lin.pred
1       18       2      3 0.667  0.977 -3.571 -3.575    3.742
2       19       7      7 1.000  0.974  0.432  0.433    3.627
3       20     776    789 0.984  0.971  2.094  3.628    3.512
4       21      19     20 0.950  0.968 -0.444 -0.448    3.397
5       22      12     14 0.857  0.964 -2.136 -2.149    3.281
6       23      26     27 0.963  0.960  0.090  0.091    3.166
```


Example: Placekicking (PlacekickDiagnostics.R, Placekick.csv)

- Residual plots with smooth loess curve (code is given in the program)



Interpreting the plot

- Two extreme standardized Pearson residuals beyond ± 3
  - Extreme negative value is an artifact of discreteness near $\hat{\pi} = 1$
    - $n_m = 3$ and observed $w_m = 2$ with $\hat{\pi}_m = .977$
    - There is actually 6.7% chance of observing a residual at least this large, under these circumstances
  - Extreme positive value **isn't**: $n_m = 789$ with $\hat{\pi}_m = .971$
    - Model underestimates observed proportion
- 6 residuals that have magnitudes of 2 or more
  - Somewhat more than 5% we would expect to see from 43 observations
  - 5 occur at small distances, and four of those five are negative.
    - Suggests *possible* misplaced logistic curve, too high at low distances
- Ignoring the extreme positive residual shows a clear sloping trend!
  - Shouldn't happen: mean should be roughly constant

Conclusion

- Something is odd about that extreme positive residual
  - Contains over half of the placekicks
  - Nearly all are PATs!!!
    - Maybe there is something different about PATs vs. field goals??
    - Try adding `PAT` to distinguish these kicks
    - Recall that variable selection wanted `PAT` in the model!

Goodness-of-Fit Statistics

- *Goodness-of-fit (GOF) statistics* are numerical summaries of the model fit
  - Objective rather than subjective
  - Omnibus measures that summarize many aspects of fit into one number
  - Provide little information regarding the *cause* of any poor fit that they might detect
- Use in combination with residual plots
  - If GOF statistic indicates problems, plots tell you what they are
  - If plots "suggest" problems, GOF statistic helps gauge the severity
    - However, GOF statistic may not show problems if a model "mostly" fits well
    - Looks for many model violation at once
    - Not well-tuned to detect particular ones

Most-used GOF statistic: Residual Deviance

- Residual deviance, D, compares model-estimated counts to observed counts using log-likelihood
  - Saturated model: $D = 0$
  - Smaller $D \Rightarrow$ closer fit
  - Residual degrees of freedom $df_r = M - (p+1)$
    - = Fitted points minus parameters
- D/df_r is often computed
  - If model is correct, expect $D/df_r \approx 1$, larger values indicate poor fit
  - If df_r is "not too small" we can suggest *very* rough thresholds for D/df_r
    - $D/df_r > 1 + 2\sqrt{2/df_r}$ indicates a potential problem
    - $D/df_r > 1 + 3\sqrt{2/df_r}$ indicates a poor fit
  - The statistic tells you nothing about the cause of the problem
    - Can help you decide whether things you see in the plots are serious
    - Doesn't work great with continuous explanaotory variables

- The same thing can be done using Pearson goodness-of-fit statistic

$$X^2 = \sum_{m=1}^{M} \frac{(y_m - \hat{y}_m)^2}{\widehat{Var}(y_m)} = \sum_{m=1}^{M} e_m^2$$

where e_m is the *unstandardized* Pearson residual

- Degrees of freedom are the same as for deviance

Example: Placekicking (PlacekickDiagnostics.R, Placekick.csv)

- Residual deviance and df are printed in `summary()` for a model fit
- We extract the necessary components for the calculations and print them separately

```
> rdev <- mod.fit.bin$deviance
> dfr <- mod.fit.bin$df.residual
> ddf <- rdev/dfr
> thresh2 <- 1 + 2 * sqrt(2/dfr)
> thresh3 <- 1 + 3 * sqrt(2/dfr)
> c(rdev, dfr, ddf, thresh2, thresh3)
[1] 44.499 41.000  1.085  1.442  1.663
```

- There do not seem to be any serious problems with the model fit
  - Despite our interpretation of Pearson residuals!
  - Probably the model "mostly" fits, with the exception of the noted problem.
    - Remember: GOF stats are not very sensitive

Formal Goodness-of-Fit Tests

- In binomial model, the *Hosmer-Lemeshow* test provides a p-value for model assessment when there are numerical explanatory variables
  - Mainly looks for problems with the mean/probability portion of the model
  - Available in our `HLTest()` function in our script AllGOFTests.R
    - This script also contains other GOF tests described in the book
- A similar thing can be done for Poisson models, although the test is less well-known and less well-studied
  - We have a function, `PostFitGOFTest()` that does this

What is *Overdispersion*?

- The normal distribution has a separate parameter to measure variance
  - Variance can be anything
- In binomial (logistic) regression, $Var(Y) = n\pi(1-\pi)$
- In Poisson regression, $Var(Y) = \mu$
- These are *model assumptions* made for mathematical convenience
  - Don't have to be true in reality
- Often counts or proportions exhibit *more variability* than the model expects
- This is called *overdispersion*

- Overdispersion is a failure of the *model*, not a failure of the *data*
- Symptom of some other systematic problem rather than a problem by itself
- Often caused by inadequate regression model
  - Omitting an important variable causes observed counts to vary more around the estimated model than they should
- Positive correlations among observations can also cause overdispersion when they are modeled as independent
  - Clustered data
  - Longitudinal data
  - Time series data
- Standard models do not account for these correlations

Consequences of Overdispersion

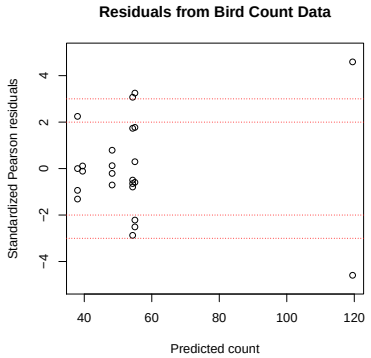
- Variances (standard errors) that are estimated by the model are smaller than they should be
- As a result:
  - Tests have excessive type I error rates
  - Confidence intervals are too narrow
  - True confidence levels smaller than stated

Symptoms of overdispersion

- Surest sign is a standardized residual plot with too many large values scattered uniformly across plot
  - Overdispersion causes overall inflation of residuals
  - More of them beyond the 2-3-4 thresholds than expected
- Secondarily, large D/df_r
  - However, this could be large for other reasons
  - Or may not be large even when there is a problem
- Does *not* usually show up in Hosmer-Lemeshow or other GOF tests!
- Note: Cannot be detected in binomial models where all $n_m = 1$
  - Only 2 possible outcomes: 0,1

Example: Ecuadorean Bird Counts (BirdQuasiPoi.R, BirdCounts.csv)

- Plot of residuals vs. $\hat{y}$ from Poisson model with 1 categorical explanatory



- 1/3 of sample beyond ± 2, 4 beyond ± 3, 2 beyond ± 4
  - Unexpected with $M = 24$
  - Spread out across entire range of $\hat{y}$
- $D/df_r = 67.2/18 = 3.73$, much larger than $1 + 3\sqrt{2/18} = 2.0$

Solutions for overdispersion

- Fix the model!
  - Add variables if some known variables are missing
    - If all variables are already in model, perhaps interactions are needed
  - Account for correlation among trials
    - Add a random effect for clusters
    - Generalized linear mixed model (see Section 7 later)
- If no cause can be identified, then use a probability model that has separate parameter for extra dispersion
  - Negative binomial or quasi-Poisson for Poisson regression
    Beta-binomial or quasi-binomial for logistic regression
    - Tests and confidence intervals have properties closer to expected

Negative binomial model for counts

- Alternative to Poisson
- The variance of Y is inflated to

$$Var(Y) = \mu + \theta\mu^2$$

  - $\theta \geq 0$ is an unknown parameter
    - Notice that $\theta = 0$ returns the same mean-variance relationship as in the Poisson model.
- Parameters are estimated by ML
  - Usual inference tools apply

Example: Ecuadorean Bird Counts (BirdOverdisp.R, BirdCounts.csv)

- Negative binomial is fit using `glm.nb()` from package `MASS`

```
> library(MASS)
>
> M.nb <- glm.nb(formula = Birds ~ Loc, data = alldata)
> summary(M.nb)

Call:
glm.nb(formula = Birds ~ Loc, data = alldata, init.theta = 33.38468149,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8076  -0.4877  -0.0849   0.5511   1.6570

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.876      0.113   34.44  < 2e-16 ***
LocForA        0.907      0.178    5.08  3.7e-07 ***
LocForB        0.131      0.144    0.91     0.36
LocFrag        0.119      0.144    0.82     0.41
LocPasA       -0.200      0.201   -1.00     0.32
LocPasB       -0.239      0.164   -1.46     0.14
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(33.38) family taken to be 1)

    Null deviance: 73.370  on 23  degrees of freedom
Residual deviance: 22.705  on 18  degrees of freedom
AIC: 198.1

Number of Fisher Scoring iterations: 1


              Theta:  33.4
```


Example: Ecuadorean Bird Counts (BirdOverdisp.R, BirdCounts.csv)

- $D/df_r = 22.7/18 = 1.26$, below thresholds
- $\hat{\theta} = 33.4$ with a standard error of 14.9 (not shown)
  - More than two standard errors from 0, indicating that the overdispersion correction is important.
- The LRT for equality of the means at the six locations gives tiny p-value (not shown), but bigger than in Poisson
- Confidence intervals for means (not shown) wider than in Poisson fit

1. Introduction
2. Analyzing a binary response, 2×2 tables
3. Analyzing a binary response, logistic regression
4. Analyzing a multicategory response
5. Analyzing a count response
6. Model selection and evaluation
7. Models for correlated data
   - Introduction
   - Random effects
   - Generalized linear mixed models
   - Inference in GLMMs
   - Extensions
8. Conclusion

- All of the models that were introduced in Sections 2–4 make the critical assumption that the data to which they are fit consist of independent observations
- Data are often collected in "groups"
  - Clusters
    - Multiple placekicks per kicker
    - Focus groups in marketing study
    - Patients within a given hospital in a research study
  - Multiple measurements on the same subject ("repeated measures")
    - Follow-up measurements in clinical trial
    - Plant growth

- Measurements from the same cluster or unit are usually more similar to one another than to observations made on different clusters or units
  - In other words, the responses within groups are *correlated*
- Performing a statistical analysis on correlated data as if they were independent is not recommended
- In particular, when data are positively correlated, this leads to overdispersion
  - Tests have inflated Type 1 error rate
  - Confidence intervals under-cover

- Two basic approaches to the analysis of correlated grouped data
  1. Change the statistical model so that it correctly reflects the grouping structure of the data
     - *Generalized linear mixed model*
     - More flexible, but more complicated
     - Our focus here
  2. Fit models assuming independence and adjust inferences for correlation
     - *Generalized estimating equations*

Consider a population that consists of many "groups" (clusters or subjects)

- Different groups often have inherently different response potential
  - Children in some schools score generally higher than those in others
  - Some hospitals better than others
  - Kickers have different skill levels
  - Individuals (plants, people) genetically better/stronger, regardless of when measured
- Model this by adding a nominal "group" variable
  - Different levels for each group
  - Creates "effect" to raise or lower means for all members at the same level

- Groups are (supposedly) randomly sampled
  - Meant to represent broader population
  - Levels and effects actually used are random
  - The variable for groups is called a *random-effects factor*
  - Adds *variability* to the responses
- Compare to *fixed-effects* variables or factors
  - Regression variables, treatment factors, covariates, etc.
  - Levels have systematic effects on means

- Random effect values follow some known distribution with unknown parameters
- Typically normal with mean 0 and unknown variance
- The zero mean because separate fixed-effects parameters model means
- Unknown variance creates a new parameter in the model
  - Called its *variance component* of the random effects factor
  - Measures how variable members of that population are
- Usually (not always) we don't care about variance component
- Random effects arise from grouping used for convenience or statistical power
  - Increase sample size without increasing sampling effort
  - Comparisons across time less variable within subjects
  - Grouping-induced correlation is a nuisance that must be dealt with

Example: Falls with Head Impact[2] (FallsGLMM.R, FallHead.csv)

- Schonnop et al. (2013) studied video footage of 227 falls among 133 residents at two long-term care facilities in British Columbia, Canada
- We consider a reduced version of this data set consisting of the 215 falls with recorded values for all of the following variables:
  - `resident`: a numerical identification code for the resident whose fall was recorded
  - `initial`: a 4-level categorical variable with levels "`Backward`", "`Down`", "`Forward`", and "`Sideways`"
  - `head`: a binary variable indicating whether the fall resulted in the resident's head impacting the floor (1 = yes, 0 = no)
- Will address simple question of whether initial direction influences P(head impact)

[2]Data kindly provided by Dr. Steve Robinovich, Department of Biomedical Physiology and Kinesiology, and School of Engineering Science, Simon Fraser University

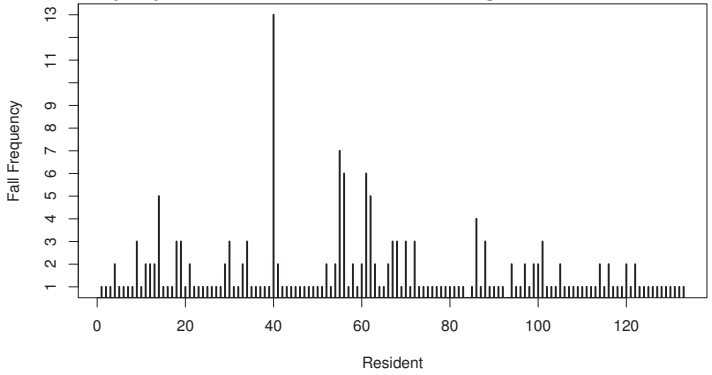
Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

First few lines of data:

```
> fall.head <- read.csv("C:\\Data\\FallHead.csv")
> head(fall.head)
  resident  initial head
1       56 Sideways    0
2        9 Backward    0
3       30  Forward    0
4        9     Down    0
5       70 Sideways    0
6       21 Sideways    1
```


Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

Fall Frequency by resident shows clustering:



- Most residents who fell did so only once, some fell numerous times
- Perhaps underlying mechanism causes a given resident to impact the floor in similar ways each time
  - A tendency for dizziness
  - Weakness in a particular leg
- For this reason, we consider `resident` to be a random-effects factor in any analysis we undertake

- Models studied so far are *fixed-effects models*
  - Contain only fixed effects
- Will now consider *mixed-effects models*, or "mixed models" for short
  - Contain random effects in addition to fixed effects
  - Accounts for different sources of *variability* in model
    - Resident-to-resident vs. fall-to-fall within resident
- Logistic and Poisson models are different types of *generalized linear models* (GLMs)
  - Extending a GLM with random effects creates a *generalized linear mixed model* (GLMM)

Random Effects Model

- Consider first a simple problem:
  - t groups, a responses/group
  - Count response, so Poisson regression
- Mean of response k in group i is μ_{ik}
- Write model for mean as

$$\log(\mu_{ik}) = \beta_0 + b_{0i},$$

  - β_0 is the value of the linear predictor (the log-mean) in an average group
  - b_{0i}, $i = 1, \ldots, a$ is the random effect of group i
  - $b_{01}, \ldots, b_{0a}$ are assumed to be a random sample from $N(0, \sigma^2_{b0})$
- This is a random-effects GLM
  - Could do logistic regression by replacing $\log(\mu)$ with $\text{logit}(\pi)$

We can extend this to a generalized linear *mixed* model if we have additional measurements

- Have an additional explanatory variable with each response
  - Denote by $x_{ik}, i = 1, \ldots, a,\ k = 1, \ldots, t$
- Suppose log-mean should change linearly with x
- A fixed-effects generalized linear model ignoring groups would use $\log(\mu_{ik}) = \beta_0 + \beta_1 x_{ik}$
- Adding random effects for groups yields GLMM:

$$\log(\mu_{ik}) = \beta_0 + \beta_1 x_{ik} + b_{0i},$$

  - $b_{01} \ldots, b_{0a}$ are assumed to be a random sample from $N(0, \sigma^2_{b0})$
  - Notice that we could rearrange this model as $\log(\mu_{ik}) = (\beta_0 + b_{0i}) + \beta_1 x_{ik}$
    - Intercept is random, varying by group
    - Slope between log-mean count and x is same for all groups

Can allow slope to change by group, too:

$$\log(\mu_{ik}) = \beta_0 + \beta_1 x_{ik} + b_{0i} + b_{1i} x_{ik},$$

- Additional random effects $b_{11}, \ldots, b_{1a}$ are a random sample from $N(0, \sigma^2_{b1})$
- Rearranging this model yields $\log(\mu_{ik}) = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i}) x_{ik}$.
- Model consists of both random intercepts and random slopes for each group

A GLMM is fit to data using maximum likelihood estimation

- Likelihood has complicated integral to approximate
- Different iterative numerical methods can be used
- With a single random effect, *Adaptive Gaussian Quadrature* (AGQ) is recommended for the approximation
  - The `glmer()` function of the `lme4` package can perform AGQ
  - Use the `nAGQ` argument to specify number of quadrature points (more is better but may take longer)
- With more than one random effect, use *Laplace approximations*
  - Not as accurate as AGQ; equivalent to `nAGQ = 1`
  - Default in `glmer()`
- With repeated measures, may prefer *partial quasi-likelihood*
  - The `glmmPQL()` function in the `MASS` package applies PQL
  - Flexible, but not as accurate as AGQ

- In `glmer()`, add random effects in `formula`
  - The fixed-effects portion uses the same syntax as `glm()`
    - e.g., `formula = y ~ x`
  - Random effects are incorporated into the `formula` argument value by adding terms of the form `(a|b)`
    - `b` is replaced by the name of the random-effects factor
    - `a` is replaced by one or more terms in `formula` form that indicate the variables whose coefficients are to be taken as random
- Examples
  - `(1|b)`: random effects are added to the intercept for each level of `b` (e.g., $\log(\mu_{ik}) = \beta_0 + \beta_1 x_{ik} + b_{0i}$)
  - `(0+x|b)`: random effects are added to the regression coefficient for `x`
  - `(x|b)`: random effects are added to the intercept *and* to the regression coefficient for `x`, and these random effects are correlated
  - `(1|b)+(0+x|b)`: both the intercept and the regression coefficient for `x` have *independent* random effects (e.g., $\log(\mu_{ik}) = \beta_0 + \beta_1 x_{ik} + b_{0i} + b_{1i} x_{ik}$);
  - `(x1+x2|b)`: The intercept and the regression coefficients for `x1` and `x2` have *correlated* random effects.

Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

- Model the P(head impact) as a function of the initial direction of the fall
  - The binary variable `head` is our response variable
  - The direction of the fall, `initial`, is a fixed effect
  - The factor `resident` is a grouping factor, random effect
- Model:

$$\text{logit}(\pi_{ik}) = \beta_0 + \beta_2 x_{2ik} + \beta_3 x_{3ik} + \beta_4 x_{4ik} + b_i,$$

  - π_{ik} is the probability that fall k for resident i has a head impact
  - β_0 is the log-odds of head impact fall for a person with `initial = "Backward"`
  - $x_{2ik}, x_{3ik}, x_{4ik}$ are indicator variables for levels `"Down"`, `"Forward"`, and `"Sideways"` of `initial`,
  - β_j, $j = 2, 3, 4$ is the difference in log-odds of head impact between level j and level 1 of `initial` ("`Backward`")
  - b_i is the random effect of resident i upon the log-odds of head impact
    - We assume that the b_i's are independent $N(0, \sigma_b^2)$

- We fit models using 1, 2, 3, 5, and 10 points using the `nAGQ` argument (not shown)
- Variance component estimates change until `nAGQ = 5`, so use ≥ 5
  - No worries about "too many" other than run time
- Saving fit with `nAGQ = 5`

```
> library(lme4)
> mod.glmm.5 <- glmer(formula = head ~ initial + (1 | resident),
    nAGQ = 5, data = fall.head, family = binomial(link = "logit"))
```

Results of model fit:

```
> summary(mod.glmm.5)
Generalized linear mixed model fit by maximum likelihood ['glmerMod']
 Family: binomial ( logit )
Formula: head ~ initial + (1 | resident)
   Data: fall.head

     AIC      BIC   logLik deviance
   279.5    296.4   -134.8    269.5

Random effects:
 Groups   Name        Variance Std.Dev.
 resident (Intercept) 0.0921   0.303
Number of obs: 215, groups: resident, 131

Fixed effects:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.645      0.242   -2.66   0.0078 **
initialDown       -1.170      0.678   -1.73   0.0844 .
initialForward     0.958      0.365    2.63   0.0086 **
initialSideways   -0.121      0.372   -0.32   0.7453
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) intlDw intlFr
initialDown -0.354
initilFrwrd -0.659  0.236
initilSdwys -0.646  0.230  0.435
```


Interpretation of fit

- The estimated probability of head impact for fall k from resident i is

$$\text{logit}(\hat{\pi}_{ik}) = -0.65 - 1.17x_{2ik} + 0.96x_{3ik} - 0.12x_{4ik} + b_i,$$

  - b_i has mean 0 and variance 0.092
- For an average resident (i.e., $b_i = 0$)
  - Backward fall: $\text{logit}(\hat{\pi}) = -0.65$, which results in $\hat{\pi} = 0.34$
  - Downward fall: $\text{logit}(\hat{\pi}) = -0.65 - 1.17 = -1.82$, or $\hat{\pi} = 0.14$
- Intercept varies across residents with estimated standard deviation of $\hat{\sigma}_b = 0.303 = \sqrt{0.092}$
  - *Approximately* 95% of backward falls have log-odds of head impact within $\hat{\beta}_0 \pm 2\hat{\sigma}_b = 0.6447 \pm 0.606 = -1.24$ to -0.05
    - This translates to probabilities between 0.22 and 0.49
  - Similarly, approximately 95% of downward falls have log-odds of head impact within -1.8152 ± 0.606
    - Corresponds to probabilities between 0.08 and 0.23

- LR methods can be more difficult to apply because the likelihood can be much more difficult to evaluate
  - Especially true with complicated models or large data sets
  - Tests easier than profile LR confidence intervals
  - Improved computational capacity and fitting algorithms are helping
- Wald inferences historically the standard
  - Tend to be even less accurate in GLMMs than they are with ordinary GLMs
- Good alternative is *parametric bootstrap*

- Parametric bootstrap hypothesis test:
  1. Compute test statistic on data
  2. Identify model implied by H_0 and fit it to data
  3. Simulate many data sets from this fitted model
  4. Repeat analysis on each data set
  5. P-value is proportion of simulated test statistics that are at least as extreme (i.e., that favor the alternative hypothesis at least as much) as the one computed on the original data

- Parametric bootstrap confidence interval
  1. Simulate many data sets from original fitted model with all effects intact
  2. Estimate parameter for each data set
  3. Manipulate set of estimates into interval endpoints using some appropriate technique
     - Several techniques described in Davison and Hinkley (1997)

Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

- Test for equality of probabilities across the four initial fall directions
  - $H_0 : \beta_2 = \beta_3 = \beta_4 = 0$ in model
  - H_a : not all of $\beta_2, \beta_3, \beta_4$ are zero
- Use LRT and parametric bootstrap
  - The LRT for each fixed-effects term in the model is conducted using the `drop1()` function

```
> lrt <- drop1(mod.glmm.5, test = "Chisq")
> lrt
Single term deletions

Model:
head ~ initial + (1 | resident)
        Df AIC  LRT Pr(Chi)
<none>     280
initial  3 289 15.6  0.0013 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The LRT gives a test statistic of 15.6 and a p-value of 0.001 based on the large-sample χ^2_3 approximation

Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

- Save original LR test statistic
- Simulation for parametric bootstrap uses `simulate()` in `lme4` package
- Apply analysis to columns of data stored in matrix (`simfix.h0` below)
- Save each test statistic
- Compute proportion of simulated test stats larger than original

```
> names(lrt)
[1] "Df"      "AIC"     "LRT"     "Pr(Chi)"
> orig.LRT <- lrt$LRT[2]  # Saves LR Test statistic
>
> # Fit Null model
> mod.glmm0 <- glmer(formula = head ~ (1 | resident), nAGQ = 5,
    data = fall.head, family = "binomial")
> sims <- 1000
> simfix.h0 <- simulate(mod.glmm0, nsim = sims, seed = 9245982)
> # Fit Model and compute test statistic
> LRT0 <- numeric(length = sims)
> for (i in 1:sims) {
    m1 <- glmer(formula = simfix.h0[, i] ~ initial + (1 | resident),
      nAGQ = 5, data = fall.head, family = "binomial")
    LRT0[i] <- drop1(m1, test = "Chisq")$LRT[2]
 }
```


Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

```
> summary(LRT0)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.048   1.270   2.330   3.050   4.280  17.200
> pval <- mean(LRT0 >= orig.LRT)
> pval
[1] 0.003
```

- The parametric bootstrap code runs for several minutes and produces a p-value of 0.003
  - Similar to χ^2_3 from LRT
- Reject the null hypothesis and conclude that the probabilities of head impact are not the same for all four initial falling directions

Example: Falls with Head Impact (FallsGLMM.R, FallHead.csv)

- Next perform tests and construct confidence intervals for all pairwise comparisons
- `mcprofile` package for LR inferences has not yet been extended to work with `glmer`
- Use Wald-based procedures from the `multcomp` package
- Same techniques as shown earlier
  - See the program corresponding to this example
- Shows that forward falls have higher P(head impact) than others
- Parametric bootstrap intervals also shown in program

Tests and confidence intervals for variance components

- Not typically done when fixed effects are focus
- When done, usually want to test whether random effects are at all different
  - H_0 : Variance component = 0
  - Testing at the boundary (0) of the parameter
    - Invalidates LR theory
  - Doesn't affect LR confidence intervals as much
- Sampling distribution of variance components often highly skewed unless sample size is enormous
  - Invalidates Wald test
  - Wald confidence intervals frequently give negative lower endpoint!
- Parametric bootstrap is best alternative
  - Process is same as before
- Examples in code

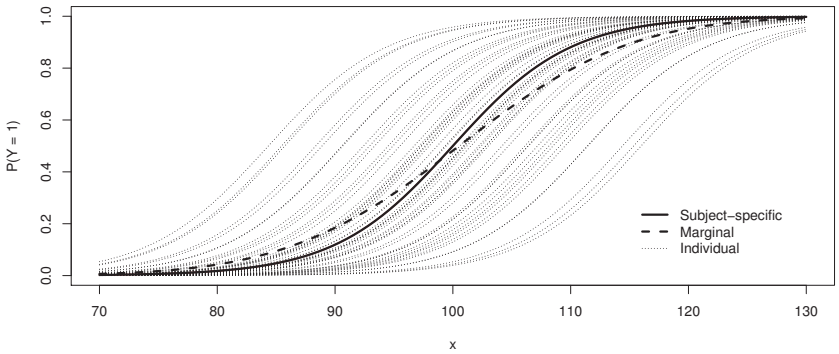
Extensions of GLMMs

- A given study can have more than one random-effects factor
  - Can be nested or crossed
- Example
  - Many labs can process and analyze blood samples
  - Many technicians work at each lab
  - Develop a study to test whether there is variability across the population of all labs or across the population of all technicians
  - Gather multiple samples from each of a large number of donors
    - Send 3 samples from each donor to L randomly selected labs
    - Analyzed by 3 different technicians within each selected labs
  - We have three random-effects factors
    - Labs, technicians nested within the labs, and donors (crossed with both labs and technician)

Marginal modeling using generalized estimating equations

- GLMM is referred to as a *subject-specific model*
  - Addition of random effects allows each subject to have its own different parameter values (slopes, intercepts)
- Alternative is to directly model how an explanatory variable relates to the population average response, rather than how it relates to individuals in the population
  - Direct model for the population average is referred to as a *marginal model*, because the population average is derived from the marginal distribution of the outcome

- Figure: Subject-specific model and marginal model for simulated sample of 50 subjects from a population with probability of success following a logistic curve with common slope but different intercepts. The corresponding code for the plot is in LogisticSim.R.



Fitting marginal model

- Do the wrong thing: treat data as if no grouping!
  - Assume independence rather than correlation
  - Use this as a "working model"
- If model is otherwise right, parameter estimates are consistent
  - Close to correct in large samples
- Estimated variances of these estimates are wrong
  - They can be corrected using information in the data!
  - "Sandwich estimator" of the variance
  - Further details on their calculation are given in Liang and Zeger (1986), Agresti (2002), and Molenberghs and Verbeke (2005)
- Must use Wald or parametric bootstrap inferences, but parameter estimates much closer to normal than in GLMM
  - See program FallsGEE.R
- Hard to apply in complicated grouping structures (like lab example)

1 Introduction
2 Analyzing a binary response, 2×2 tables
3 Analyzing a binary response, logistic regression
4 Analyzing a multicategory response
5 Analyzing a count response
6 Model selection and evaluation
7 Models for correlated data
8 Conclusion
- Objectives
- Additional material

- Apply appropriate methods to analyze data in a contingency table
- State, interpret, and fit logistic, multinomial, proportional odds, and Poisson regression models
- Use appropriate variable-selection methods
- Evaluate the fit of categorical regression models
- Identify and solve overdispersion problems
- Be comfortable with using R to analyze categorical data

Analysis of Categorical Data

Christopher R. Bilder[1] and Thomas M. Loughin[2]

[1]University of Nebraska–Lincoln, Department of Statistics

[2]Simon Fraser University, Department of Statistics and Actuarial Science

www.chrisbilder.com/categorical

Bibliography

Agresti, A. (2002). *Categorical Data Analysis*. Wiley, 2nd edition.

Bilder, C. and Loughin, T. (1998). "It's Good!" An analysis of the probability of success for placekicks. *Chance*, 11:20–24.

Bilder, C. and Loughin, T. (2014). *Analysis of Categorical Data with R*. CRC Press.

Calcagno, V. and de Mazancourt, C. (2010). `glmmulti`: An R package for easy automated model selection with (generalized) linear models. *Journal of Statistical Software*, 34.

Davison, A. and Hinkley, D. (1997). *Bootstrap Methods and their Application*. Cambridge University Press.

DeHart, T., Tennen, H., Armeli, S., Todd, M., and Affleck, G. (2008). Drinking to regulate romantic relationship interactions: The moderating role of self-esteem. *Journal of Experimental Social Psychology*, 44:527–538.

Liang, K. and Zeger, S. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73:13–22.

Molenberghs, G. and Verbeke, G. (2005). *Models for Discrete Longitudinal Data*. Springer.

Rerks-Ngarm, S., Pitisuttithum, P., Nitayaphan, S., Kaewkungwal, J., Chiu, J., Paris, R., Premsri, N., Namwat, C., de Souza, M., Adams, E., Benenson, M., Gurunathan, S., Tartaglia, J., McNeil, J., Francis, D., Stablein, D., Birx, D., Chunsuttiwat, S., Khamboonruang, C., Thongcharoen, P., Robb, M., Michael, N., Kunasol, P., and Kim, J. (2009). Vaccination with ALVAC and AIDSVAX to prevent HIV-1 infection in Thailand. *New England Journal of Medicine*, 361:2209–2220.

Schonnop, R., Yang, Y., Feldman, F., Robinson, E., Loughin, M., and Robinovitch, S. (2013). Prevalence of and factors associated with head impact during falls in older adults in long-term care. *Canadian Medical Association Journal*, 185:E803–E810.

Wardrop, R. (1995). Simpson's paradox and the hot hand in basketball. *The American Statistician*, 49:24–28.

R Index

`aggregate()`, 45
AlcoholPoRegs.R, 97
AllGOFTests.R, 165
AllSubsetsPlacekick.R, 137
`Anova()`, 33, 53, 101
`anova()`, 33, 37
`apply()`, 73
`as.numeric()`, 41
`assocstats()`, 113
BeetleEggCrowding.R, 129
BirdOverdisp.R, 173
BirdQuasiPoi.R, 169
Bootstrap, 193
`car` package, 33
`class()`, 49, 69
`confint()`, 41, 73, 89, 101
`confint.default()`, 41
`contrasts()`, 49
`drop1()`, 197
`factor()`, 49, 81
FallsGLMM.R, 181, 189, 197, 201

glm(), 29, 33, 53, 57, 97
glm() arguments, 45, 49
glm.nb(), 173
glmer(), 189
glmmPQL(), 189
glmulti package, 137
glmulti(), 137
HIVvaccine.R, 21
HIVvaccinePoisson.R, 109, 113, 121
HLTest(), 165
LASSOPlacekick.R, 145
levels(), 49, 69, 81
lme4 package, 189
LogisticSim.R, 205
Marginal model, 201
MASS package, 189
MASS package, 81
matrix(), 45
mcprofile package, 201
mcprofile package, 41
mcprofile(), 41, 45, 57
methods(), 69
multcomp package, 201
multinom(), 69

nnet package, 69
PiPlot.R, 25
Placekick.R, 29, 33, 37, 45
PlacekickDiagnostics.R, 153
PolIdeolNominal.R, 125
polr(), 81
PostFitGOFTest(), 165
predict(), 45
read.csv(), 29
rev(), 41
rstandard(), 157
StepwisePlacekick.R, 145
Subject-specific model, 201
summary(), 29
TomatoVirus.R, 49
vcd package, 113
vcov(), 29
wald(), 41
weightable(), 137
Wheat.R, 65, 73, 81, 89