STATISTICS 475: Applied Discrete Data Analysis

Logistic Regression Models for Binary Responses

(B&L Section 2.1–2.2.1)

1 Problem to be solved

- With continuous data, you first learned about the normal distribution model, followed by inference on one mean and inference on two means
  - After that, you started learning regression
- So far we have learned about the Bernoulli distribution for binary RVs and the binomial distribution model for multiple independent Bernoullis.
- We have learned inference for one probability and for comparisons of two probabilities
  - This involved special techniques that are parallel to, but distinct from, those used for means of normal distributions
- Very often, we observe binary or binomial responses *along with one or more explanatory variables*
  - Probabilities of success may depend on these variables
  - Now we need to generalize the notion of regression to apply to binary/binomial data

2 Example for this chapter

I will start with the example that we will use in this chapter:

Example: Placekicking (Adapted from the book) (Lecture 7 scripts.R, Placekick.csv)

In American and Canadian football, points can be scored by a "placekicker" kicking a ball through a target area at an end of the field. A success occurs when the football is kicked over the crossbar and between the two uprights of the goal posts. The placekicker's team receives either 1 or 3 points for a successful kick, where a "point after touchdown" (PAT) receives 1 point and a "field goal" receives 3 points. A placekick that is not successful receives 0 points. *See the videos of "famous" placekicks on our Canvas site.*

My coauthor, Chris Bilder, collected data on every placekick attempted in the 1995 season of the National Football League (NFL—the main professional football league in the U.S.) for a research project on the factors that affect the chance of success of a placekick. We examined a number of explanatory variables, including:

- `week`: Week of the season
- `distance`: Distance of the placekick from the goalposts, in yards (1 yard = 0.91 meters)
- `change`: Binary (or dummy or indicator) variable denoting lead-change (1) vs. non-lead-change (0) placekicks; lead-changing placekicks are those that have the potential to change which team is winning the game (for example, if a field goal is attempted by a team that is losing by 3 points or less, they will no longer be losing if the kick is successful)
- `elap30`: Number of minutes remaining before the end of the half, with overtime placekicks receiving a value of 0[1]
- `PAT`: Binary variable denoting the type of placekick, where a PAT attempt is a 1 and a field goal attempt is a 0. Until 2015, PATs were always attempted from 20 yards, except in rare circumstances. Currently, they are attempted from 33 yards, decreasing their chance of success somewhat.
- `type`: Binary variable denoting outdoor (1) vs. dome (0) placekicks
- `field`: Binary variable denoting grass (1) vs. artificial turf (0) placekicks
- `wind`: Binary variable for placekicks attempted in windy conditions (1) vs. non-windy conditions (0); we define windy as a wind stronger than 15 miles per hour at kickoff in an outdoor stadium

[1] in 1995, the NFL overtime rules awarded a win to whichever team scored first in overtime, regardless of what kind of score it was. Current rules have changed, and if a team kicks a field goal on the first possession in overtime, they do not immediately win. The other team gets one possession of the ball to try to score and either tie or win the game.

The response variable is `good`: a binary variable denoting successful (1) vs. failed (0) placekicks. There are 1,425 placekick observations (individual placekick attempts) in the data set. Below is how the data are read into R on my computer:

```
> placekick <- read.csv(file="Placekick.csv", header=TRUE)
> head(placekick)
  week distance change  elap30 PAT type field wind good
1    1       21      1 24.7167   0    1     1    0    1
2    1       21      0 15.8500   0    1     1    0    1
3    1       20      0  0.4500   1    1     1    0    1
4    1       28      0 13.5500   0    1     1    0    1
5    1       20      0 21.8667   1    0     0    0    1
6    1       25      0 17.6833   0    0     0    0    1
```

Our goal will be to develop a model that explains how the probability of successful placekick—$P(\texttt{good} = 1)$—relates to the explanatory variables that were collected. In particular, it is well known that `distance` should affect the probability of success. How much of a role do the other variables play?

---

3 Review of Linear Regression

- Linear regression model:

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

  - Y is the RV representing the numerical response
  - $x_1, \ldots, x_p$ are the explanatory variables
    - *Get used to p as the number of explanatory variables*
    - We will index explanatory variables by j: x_j, $j = 1, \ldots, p$
  - $\beta_0, \ldots, \beta_p$ are regression parameters
    - β_0 is the intercept (mean value of Y when all $x_j = 0$)
    - β_j, $j = 1, \ldots, p$ are "(partial) regression coefficients"
      - Change in mean Y for 1-unit increase in x_j *holding all other variables constant.*
      - "Slopes," with the special constraint of keeping other variables fixed
  - ϵ is the "error"
    - Assume $\epsilon \sim N(0, \sigma^2)$
    - Represents the potential deviation between an observed response and its mean

- As a consequence of this model and properties of the normal distribution, we can write that $Y \sim N(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p, \sigma^2)$
  - In other words, we have a *distributional* MODEL for Y
    - Mean value of Y is $E(Y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
    - Equal variance σ^2 at all combinations of x_j's
  - See Appendix B.1 of B&L for more discussion of what it means to be a model
- When we want to talk about the model for Y at a particular combination of explanatory variables in a sample, say observation i for some $i = 1, \ldots, n$, we may add subscript i to the variables in the model (not the parameters):

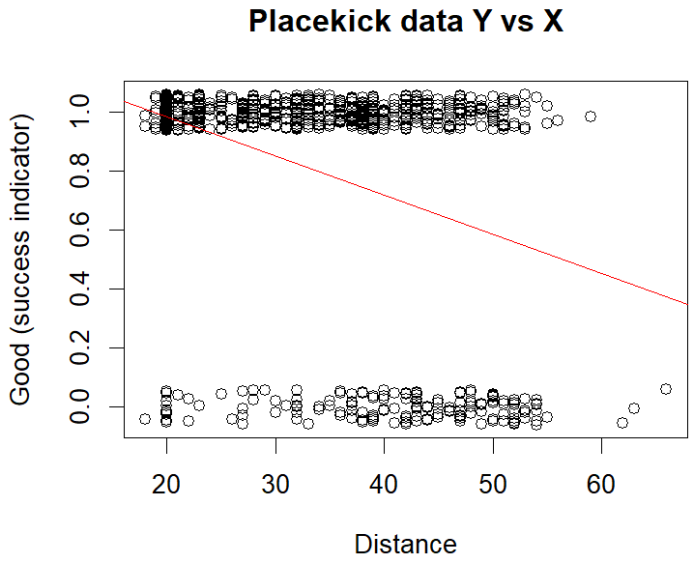
$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

  - Subscripts help us when we want to refer to different observations in a set of data
  - Not needed if we are referring more generally to a model structure.
- Model...Parameters...the next step is to estimate parameters of the model
  - Typically use least squares
    - *(Assumes equal variances)*
- ...then get the sampling distribution of the estimators
  - *Exact normality for each $\hat{\beta}_j$, assuming that Y is exactly normal*
  - Otherwise, normality as an approximation
- ...then use the results for inference
  - t-tests and CIs for parameters
  - F-tests for model comparisons
- Construct additional quantities (e.g., $\hat{Y}$, predicted value or mean for a particular combination of x_j's).

3.1 Linear regression for binary responses?

- Suppose Y_i, $i = 1, \ldots, n$ are independent Bernoulli(π_i) RVs, where $E(Y_i) = \pi_i$ might depend on $x_{i1}, \ldots, x_{ip}$
  - π_i is just the mean for Y_i,so we *could* try just using linear regression to estimate the mean
- The linear regression model is wrong, but is it usefully close?
  - Bernoulli distribution for Y_i is *FAR* from a normal distribution

Figure 1: Placekick data, `good` vs. `distance`, jittered around 0 and 1 to allow density of points to be seen more clearly. Red line is linear regression.



  - $\text{Var}(Y_i) = \pi_i(1-\pi_i)$ leads to potentially different variances, and they may be *very* different unless all π_i are in a narrow range.
  - $\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ is not constrained to be within 0 and 1, can give *impossible* estimates
  - See Figure 1 for example with placekick data.

- As a result, we often have a very poor fitting line, and inferences based on the line are unreliable.
  - Practically never use linear regression
- Instead we use a regression model for the mean π that uses the correct distributional model
  - Respect that response values occur only and 0 and 1
  - Use a model for π that respects the 0–1 boundaries

4 Logistic Regression Model

- The logistic regression model assumes that that $Y \sim$Bernoulli(π)

$$\pi = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)} \tag{1}$$

- $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ is called the LINEAR PREDICTOR
- As $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ goes toward ∞, $\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)$ approaches ∞, so π approaches 1
- As $\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ goes toward $-\infty$, $\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)$ approaches 0, so π approaches 0
- Thus, this form respects the boundaries of a probability
- If we untangle this relationship, we can show that the log of the odds has a linear model:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{2}$$

  * The log of the odds is also called the LOGIT of π:

$$\text{logit}\,(\pi) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{3}$$

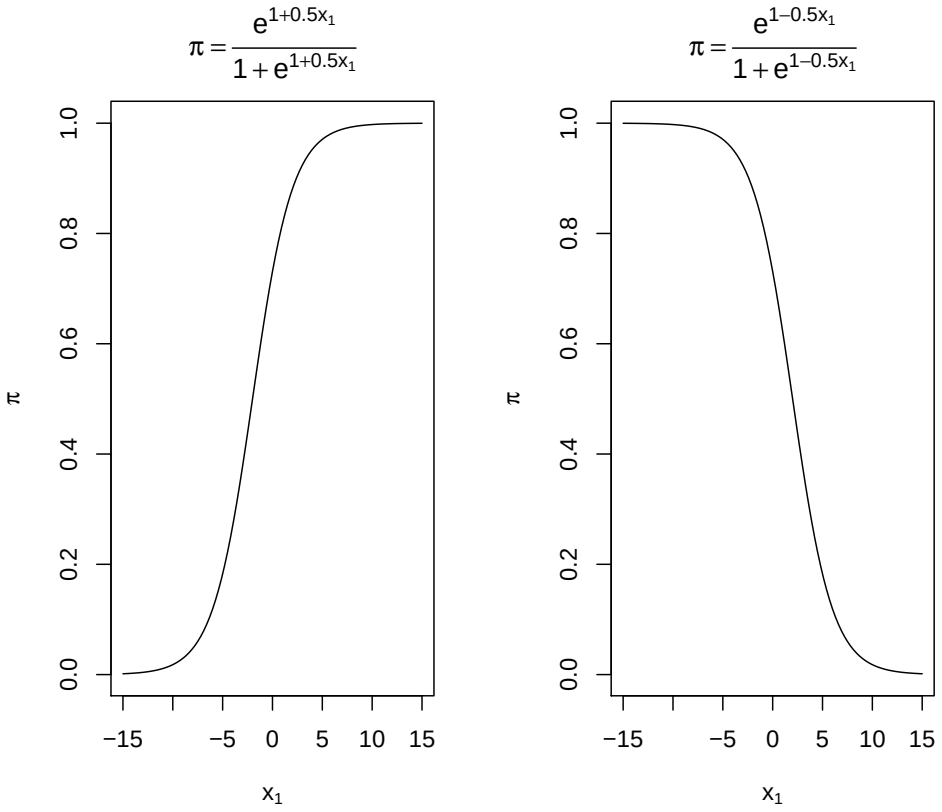
- Formulas (1), (2), and (3) are equivalent representations of the logistic regression model and we will use them interchangeably.

- Also, the variance is automatically $\pi(1-\pi)$ from Bernoulli model
- See Figure 2 for a picture of what the relationship looks like between π and any particular x_j (**PiPlot.R** from the book's website)
  - $0 < \pi < 1$
  - When $\beta_1 > 0$, there is a positive relationship between x_1 and π. When $\beta_1 < 0$, there is a negative relationship between x_1 and π.
  - The shape of the curve is somewhat similar to the letter s (this shape is called SIGMOIDAL).
  - The slope of the curve is dependent on the value of x_1.
    * We can show this mathematically by taking the derivative of π with respect to x_1: $\partial\pi/\partial x_1 = \beta_1\pi(1-\pi)$.
- Above $\pi = 0.5$ is a reversed mirror image of below $\pi = 0.5$.
  - If you rotate the graph 180 degrees, the picture is the same
- Resembles a linear relation between about $0.3 < \pi < 0.7$, but otherwise not very linear

4.1 Parameter estimation

- Gather data y_i, $i = 1, \ldots, n$ with corresponding explanatory variables
- The parameters we need to estimate are the regression coefficients $\beta_0, \beta_1, \ldots, \beta_p$

Figure 2: Shape of the logistic curve for 1-variable model with $\beta_1 = +0.5$ (left) and $\beta_1 = -0.5$ (right)



- Maximum likelihood estimation with Bernoulli distribution:

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

where, for convenience, I am stacking all of the parameters and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)'$, $\mathbf{y} = (y_1, \ldots, y_n)'$, and

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

  - Plugging this definition for π_i into the likelihood and doing some algebra gives

$$\log\left[L(\boldsymbol{\beta}|\mathbf{y})\right] = \sum_{i=1}^{n} \left\{ y_i \left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right) - \log\left(1 + \exp(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})\right) \right\},$$

    which is reasonably easy to work with mathematically and computationally (see p. 65 in the book)

- Iterative numerical methods are used to find the $\boldsymbol{\beta}$ that maximize the log-likelihood function
- Regression parameter estimates (MLEs): $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)'$
- Estimated variance-covariance matrix for regression parameter estimates is found from the curvature (second derivative) of the log likelihood
  - Doesn't lead to a *simple* formula, but is not hard to compute.
  - Results in a *matrix* of values, corresponding to the estimated variances of the parameter estimates, and their estimated covariances.

Example: Placekicking (Lecture 7 scripts.R, Placekick.csv)

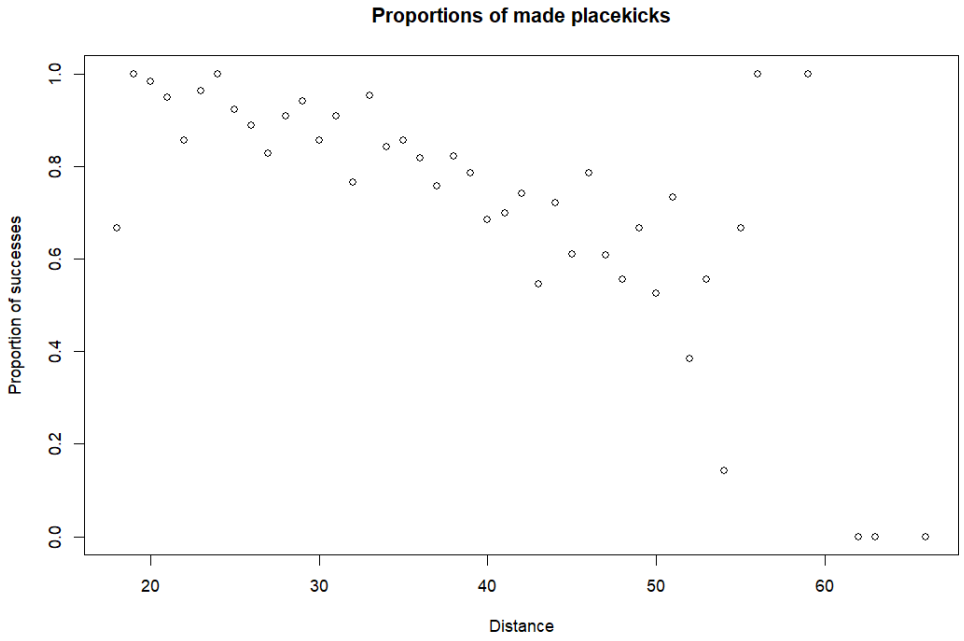
We will start by looking at a simple model of the probability of a success ($\pi = P(\texttt{good} = 1)$) against distance:

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1.$$

So $p = 1$ here, and x_1 is `distance`. See Figure 3 for a plot of the proportions of success at each distance. Note the distinct trend toward decreasing proportion of successes as distance increases. (*Therefore will expect β_1 has what sign?*)

- Logistic regression models are fit using `glm()` in R
  - Stands for "generalized linear models", which is a group of models that includes logistic regression
    - This function fits many types of models

Figure 3: Plot of proportions of successful placekicks against distance. Note: Not all distances have the same number of trials (over half of the kicks occur at 20 yards!).



    - *Get used to using it!*
  - Model for *logit* is entered using `formula=<binary y> ~ <x1>+<x2>+...+<xp>`
    - "<>" is my code for something that gets replaced by a name you have chosen
  - Logistic regression is specified with `family=binomial(link = logit)`
  - `summary(<glm object>)` gives back some of the essential information

- Here are the results for the placekick data (not all numbers have been explained yet...)

```
> mod.fit <- glm(formula=good ~ distance,
              family=binomial(link = logit), data=placekick)
> summary(mod.fit)  # summary(object = mod.fit) more completely

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
 -2.7441    0.2425    0.2425    0.3801    1.6092

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.812080   0.326277   17.81   <2e-16 ***
distance    -0.115027   0.008339  -13.79   <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: 779.75

Number of Fisher Scoring iterations: 6
```

- The estimated logistic regression model is

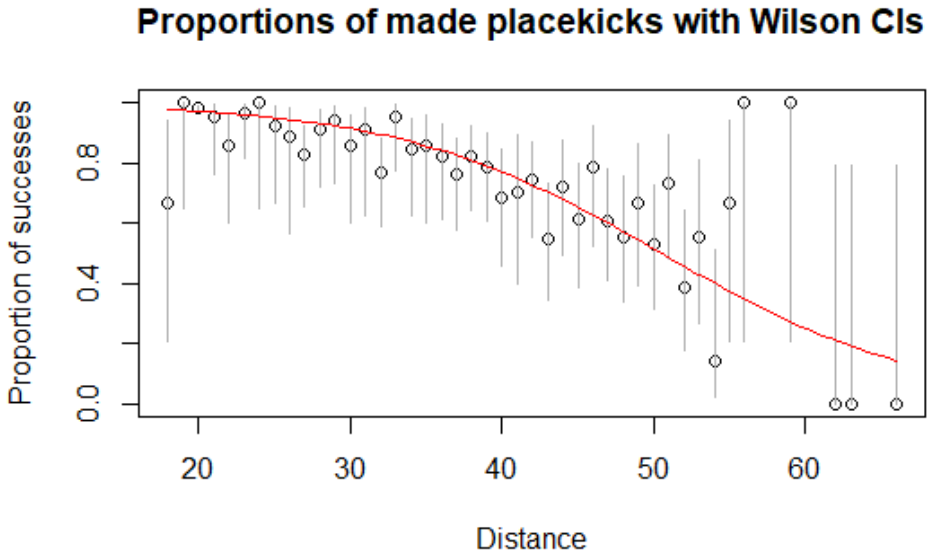
$$\text{logit}(\hat{\pi}) = 5.8 - 0.115\texttt{distance}$$

- The standard errors are

$$\sqrt{\widehat{Var}(\hat{\beta}_0)} = 0.33, \ \sqrt{\widehat{Var}(\hat{\beta}_1)} = 0.0083$$

- There are times when you will need to use the parameter estimates and/or standard errors in further computations.
  - Parameter estimates can be accessed directly using `coef(<glm object>)`
  - Estimated variances and covariances (like correlations between estimates) can be accessed directly using `vcov(<glm object>)`
  - See program for examples
- The computation took 6 iterations to achieve convergence
  - This number isn't important, but in rare cases where it does *not* achieve convergence, you will not want to report these results.
    - If it fails to converge, will see number of iterations = 25, the default maximum
    - *May* also see a warning message that it did not converge.
- The model fit is depicted in Figure 4

Figure 4: Plot of observed proportions of successes at each distance, with corresponding Wilson score confidence intervals (gray) and fitted logistic regression curve (red)



5 Individual Bernoulli Trials or Binomial?

- Notice that the model $Y \sim$Bernoulli(π) is equivalent to $Y \sim$Bin(n, π) if $n = 1$.
- What if we have multiple trials at a particular value of x, or at a particular combination of x values $x_1, \ldots, x_p$ (e.g., cancer cases out of total sample sizes among different sexes, ages, and other measures)
  - In the placekick example, we had 1425 individual attempts that we modeled as independent Bernoulli trials
    - We will use this a lot later. For now, it is good to know that you can fit logistic regressions to data where counts have been already accumulated for each unique value combination of $x_1, x_2, \ldots, x_p$
  - In our analysis using `distance` as the explanatory variable, there were numerous placekick attempts at most distances.
  - We fit our model on the 1425 individual attempts (the Bernoulli trials), but what if the data had been given to us already summarized by `distance`?
    - That is, what if we had data as $W_x \sim Bin(n_x, \pi_x)$, where n_x is the number of trials at distance x, π_x is the probability of success at distance x, and W_x is the RV for the count of successes at distance x?
    - The data would be pairs, (n_x, w_x) at each distance x.
    - How would we fit this model?
- The answer is that it doesn't matter...mostly

- You need to add a `weights=<n_x>` argument into the `glm()` function, where "`n_x`" is whatever variable name contains the number of trials.
  - See the example at the very end of B&L Section 2.2.1, p. 74–76 for computational details
  - See the script for this lecture for some of the code
- *The resulting parameter estimates and standard errors are the same either way.*
- *All inferences are the same either way*
- *Certain model-fit statistics (which we will learn about soon and again in Chapter 5) will be changed.*
  - We will worry about that when we get to it in Chapter 5.

Example: Placekicking (Lecture 7 scripts.R, Placekick.csv)

We now look at fitting the logistic regression model to counts that have been aggregated by each distance. I use `aggregate()` to create this form of data, but you can use what you want. In `aggregate()`, `x` is a formula telling is which variable to summarize, and which variables are the levels we want to summarize across. The `FUN` argument is an R function taking only `x` as an argument.

```
> w <- aggregate(x=good ~ distance, data=placekick, FUN=sum)
> n <- aggregate(x=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
> tail(w.n)
   distance success trials proportion
38       55       2      3     0.6667
39       56       1      1     1.0000
40       59       1      1     1.0000
41       62       0      1     0.0000
42       63       0      1     0.0000
43       66       0      1     0.0000
```

Then, in the `glm()` function, we write the response variable in the formula as `<w>/<n>`, replacing `<>` with the names of the variables for w and n, and we add `weights=<n>`.

```
> # Using weights= to fit model to aggregated data.
> mod.fit.bin <- glm(formula=success/trials ~ distance, weights=trials,
+                    family=binomial(link=logit), data=w.n)
> summary(mod.fit.bin) # Parameter estimates and standard errors are
                         same as before

Call:
glm(formula = success/trials ~ distance, family = binomial(link = logit),
    data = w.n, weights = trials)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.812080   0.326277   17.81   <2e-16 ***
distance    -0.115027   0.008339  -13.79   <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: 282.181  on 42  degrees of freedom
Residual deviance:  44.499  on 41  degrees of freedom
AIC: 148.46

Number of Fisher Scoring iterations: 5
```

The parameter estimates are identical to what we got before, and most other numbers in the output are the same as well (but not all!).

- So for estimation and inference on logistic regression models, you can fit the model to whatever format the data are in—binary or aggregated by explanatory variables, or some of each—and not worry about having to rearrange your data every time you want to fit a model.
- When we aggregate in this manner to create one count of successes and trials for each unique combination of explanatory variable values, we call to this as EXPLANATORY VARIABLE PATTERN (EVP) FORM for the data.

6 What to learn from this

1. Logistic regression is a *hugely* popular, go-to statistical method for binary/binomial responses. Learn to love it.
2. Get used to all of the notation, terminology, and formulas here, especially equations (1), (2), (3). We will keep using them.

3. Understand the shape and properties of the model.

4. You will have a chance to practice R code!

7 Exercises (due when announced)

1. Professional associations (like the Statistical Society of Canada and the American Statistical Association) provide services to members of different professions. They charge their members a fee called "dues" so that they have resources to provide these services. One professional society wanted to see whether increasing their dues by certain amounts would cause members to quit. They surveyed members and asked them whether an dues increase of $\$X$ would cause them to quit, where X varied between \$30 and \$50 in the question posed to different members. The results of the survey are in the file, `ProfDues.txt`. The response is in the first column, where $Y = 1$ means that the member would quit.

   (a) Make a scatterplot of Y vs. the dues increase suggested. Does the plot show anything very obvious regarding the relationship between the chance that a member would quit and the amount of dues increase?

   (b) Fit a logistic regression model relating the probability of quitting to the size of increase.

      i. Report the model in equation form similar to

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

      ii. Plot the resulting points, confidence intervals, and curve, similar to Figure 3 in this document.

      iii. Program the formula to compute probabilities of quitting for various increases. Compute the probabilities at \$0, \$30, and \$50.

      iv. Note that \$0 degrees is an extrapolation, representing what would happen if they did not increase dues at all. Comment both on this size of this estimate, and on your faith in the reliability of that estimate (do you believe that the model accurately estimates the chance that a member will quit if dues are not increased?).

Here is another exercise from B&L Ch 2 for practice, because I'd hate for you not to have something to do involving placekicking...

1. Exercise 7

Notice that I want you to become *very* comfortable with estimating logistic regression models. These models are among the most important models in this course, and in all of statistics.

STATISTICS 475: Applied Discrete Data Analysis

Inference for Logistic Regression Parameters

(B&L Section 2.2.2)

1 Problem to be solved

Model$\longrightarrow$Parameter(s)$\longrightarrow$Statistic(s)$\longrightarrow$**Sampling distribution**$\longrightarrow$**Probability statements about parameters**

- We have formulated the logistic regression model for binary data as $Y \sim \text{Bernoulli}(\pi)$ (equivalently, $\text{Bin}(1, \pi)$, where

$$\pi = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

  or equivalently

$$\text{logit}\,(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$

- This leads to parameters $\beta_0, \beta_1, \ldots, \beta_p$
- We use ML to estimate parameters, leading to estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$.
- We now need to use these parameter estimates to do *inference* on the corresponding parameters.

2 Inference on Regression Parameters

2.1 Hypothesis tests for a single parameter

- Testing parameters is done in logistic regression for the same reasons it is done in linear regression

  - You may have a particular value for a parameter that you want to test, $H_0 : \beta_j = c$
    * Most often, we are interested in $H_0 : \beta_j = 0$ for any of the parameters $j = 0, 1, \ldots, p$
    * Represents the case there the variable can be dropped from the model.
    * Not usually done for the intercept (Exercise: What does the intercept measure, and what does it tell us about probabilities of success mean when $\beta_0 = 0$?)

- The technology is the same as what we have seen so far: Score, LR, Wald.
  - However, score-based inference is not usually done due to mathematical/computational difficulty.
    * Not too hard for a single test
    * No computational shortcuts for CI—you literally have to refit the model for many values of c to find the boundaries.
  - Therefore, we focus on Wald and LR methods,

- **Wald test of** $H_0 : \beta_j = c$
  - We have the parameter estimate $\hat{\beta}_j$ and we get an estimate of its standard error directly from the likelihood computations, $\sqrt{\widehat{Var}(\hat{\beta}_j)}$.
    * Each $\hat{\beta}_j$ is an MLE, so it has the usual MLE properties of asymptotic normality, consistency, and efficiency
  - Just put them together and run the test using normal-based methods!
    * $Z = \frac{\hat{\beta}_j - c}{\sqrt{\widehat{Var}(\hat{\beta}_j)}}$
    * Rejection regions same as always, depend on H_a
      · Same for p-values
  - Usual properties associated with a Wald test
    * Often does not maintain type~1 error rate well unless sample size is large relative to the number of parameters being estimated
      · Also need enough successes and failures, but not sure how many and where among the explanatory variables they must occur.
    * No fixed recommendation for "large"
      · $n = 1425, p = 1$ in placekick example is pretty large.
      · $n = 30, p = 5$, say, is not.
      · Otherwise..."I don't know...fly casual." (https://youtu.be/vb-MN4MXkBo?t=126)

- **LR test of** $H_0 : \beta_j = c$
  - This is more complicated than before, because of the definition of the LR test statistic: $-2\log(\Lambda)$, where

$$\Lambda = \frac{\text{Maximum of likelihood function under } H_0}{\text{Overall maximum of likelihood function}}$$

    - The denominator of this is easy: it is just the value of the likelihood function at the $p+1$ MLEs $\hat{\beta}_0, \ldots, \hat{\beta}_p$.
    - Numerator is tricky: In logistic regression, there are typically at least two parameters, β_0 and β_1.
        - Estimates are interdependent
    - We use a method called "profiling" described in the appendix at the end of the lecture to get a value
        - The result is formally called a PROFILE LIKELIHOOD RATIO TEST.
  - The test statistic $-2\log(\Lambda)$ is compared to χ^2_ν, where $\nu =$_____ (Why?)
    - As usual, only 2-sided tests are available directly, but a 1-sided p-value can be computed using techniques previously described.

2.2 Confidence interval for a single parameter

- **Wald** confidence interval is easy,

$$\hat{\beta}_j \pm Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\beta}_j)},$$

  but does not maintain nominal confidence level well
  - Inverted Wald test, so same properties.
- **LR** confidence interval is an inverted LR test
  - Requires finding all values, say b, for which the LT test would not reject $H_0 : \beta_j = b$.
  - Repeat the "profiling" process described in the Appendix to create likelihood profiles at different values of b
  - Find the endpoints of the CI using an iterative numerical algorithm that cleverly chooses values of b at which to evaluate the profile
  - Result is referred to as a PROFILE LR CONFIDENCE INTERVAL

Example: Placekicking (Lecture 8 scripts.R, Placekick.csv)

Recall that we previously fit the probability of a successful kick as a function of distance by applying the model $\text{logit}(\pi) = \beta_0 + \beta_1 x_1$ to the Bernoulli trials, where $x_1 =$ `distance`:

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

A natural question is whether the probability of success is related to the distance of the kick. This implies that we want to test the hypotheses:

$$H_0 : \_\_\_\_ \quad H_a : \_\_\_\_$$

A Wald test is found in the coefficient table of the `summary()` of the model fit:

```
> summary(mod.fit)

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

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.812080   0.326277   17.81   <2e-16 ***
distance    -0.115027   0.008339  -13.79   <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: 779.75

Number of Fisher Scoring iterations: 6
```

The Wald test statistic is listed under `z value`. For our test, $H_0 : \beta_1 = 0$ has test statistic

$$Z_W = \frac{-0.115}{0.0083} = -13.79$$

and a p-value of "`<2e-16`", which means that it is smaller than 2×10^{-16}, which is the smallest p-value that R reports in many functions. Thus, we strongly reject H_0 and conclude that the distance of the kick *is* related to the probability that it is good.

The LR test is found using the `Anova()` function of the `car` package with the argument `test="LR"`. Note that there is another function called `anova()` in the base `stats` package, but this function does things I don't like when $p > 1$ (see "Sequential Tests" below), so I routinely use `Anova()`.

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

Response: success/trials
         LR Chisq Df Pr(>Chisq)
distance   237.68  1  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test statistic is $-2\log(\Lambda) = 237.68$ which leads to a p-value of $< 2 \times 10^{-16}$ on χ^2_1. This leads to the same conclusion as the Wald test.

Confidence intervals for the regression parameters are found next. The `confint()` function gives LR intervals for regression parameters when applied to objects of the class `glm`, while `confint.default()` does Wald.

```
> # LR CI for all parameters
> round(confint(mod.fit), digits=3)
Waiting for profiling to be done...
             2.5 % 97.5 %
(Intercept)  5.196  6.477
distance    -0.132 -0.099
>
> # Wald Intervals
> round(confint.default(mod.fit), digits=3)
             2.5 % 97.5 %
(Intercept)  5.173  6.452
distance    -0.131 -0.099
```

Both results are very similar, owing to the large sample size. Using the preferred LR intervals, we are 95% confident that the interval $-0.132 < \beta_1 < -0.099$ covers *the true change in log odds for each 1-yard change in distance.*

2.3 Hypothesis tests with multiple parameters (Model comparison tests)

- Recall that, when $p > 1$, the interpretation of a regression parameter β_j for $j = 1, \ldots, p$ is that it represents the change in the log-odds of success when x_j in increased by 1 unit *holding all other variables constant.*
  - In particular, that means that when we consider $H_0 : \beta_j = 0$—meaning that we can drop x_j from the model when H_0 is true—we assume that all of the other variables remain in the model.
- This complicates the process of defining hypotheses and tests
  - The complication is addressed by looking at tests as a comparison of two models
    - Start with a full model and apply constraints to its parameters—limit their possible values—to create a new model with fewer parameters to estimate
    - Null hypothesis is that the constraints are true, vs. alternative that they are not
    - Equivalently, larger model—the FULL MODEL—is the alternative hypothesis and smaller model—the REDUCED MODEL— is the null hypothesis
- If $p = 1$ then the model comparison is simple. Starting from full model $\text{logit}(\pi) = \beta_0 + \beta_1 x_1$:

- $H_0 : \beta_1 = 0$ is equivalent to $H_0 : \text{logit}(\pi) = \beta_0$ (Reduced model)
  $H_a : \beta_1 \neq 0$ is equivalent to $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1$ (Full model)

- However, tests of $H_0 : \beta_j = 0$ can have different interpretations and different results depending on what other variables are assumed to be in the model when variable x_j is removed

  - For example, suppose that $p = 2$. Then a test of $H_0 : \beta_2 = 0$ could be interpreted as two different comparisons of pairs of models:

    1. $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 x_1$
       $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
    2. $H_0 : \text{logit}(\pi) = \beta_0$
       $H_a : \text{logit}(\pi) = \beta_0 + \beta_2 x_2$

  - Similar for tests of β_1.
  - With more than 2 variables, the number of possibilities becomes overwhelming
    * Each of these comparisons can give different results (and usually do)

- There are several different strategies for how parameters for $p \geq 2$ variables can be tested for significance. The two main ones are PARTIAL tests and SEQUENTIAL tests

  - PARTIAL TESTS assume that *all other variables will remain in the model* when the test of $\beta_j = 0$ is considered.
    * That is, the "full model" always includes all the variables in the problem.
    * The "reduced" model includes constraints specified by the null hypothesis
    * For example, for $p = 3$, separately testing the significance of each variable's parameter results in these three pairs of hypotheses:
      1. Testing $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$ is equivalent to testing
         $H_0 : \text{logit}(\pi) = \beta_0 + \beta_2 x_2 + \beta_3 x_3$
         $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
      2. Testing $H_0 : \beta_2 = 0$ vs. $H_a : \beta_2 \neq 0$ is equivalent to testing
         $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_3 x_3$
         $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
      3. Testing $H_0 : \beta_3 = 0$ vs. $H_a : \beta_3 \neq 0$ is equivalent to testing
         $H_0 : \text{logit}'(\pi) =$_____
         $H_a : \text{logit}(\pi) =$_____
    * The Wald tests in the `summary()` output are always partial tests.
  - SEQUENTIAL TESTS assume that there is an inherent ordering to the importance of the variables and assume that any variables in the null model are the ones that are more important than the one being tested.
    * For example, suppose we think the importance ordering has x_1 more important than x_2, and x_2 more important than x_3. Then

1. Testing $H_0 : \beta_1 = 0$ vs. $H_a : \beta_1 \neq 0$ is equivalent to testing
   $H_0 : \text{logit}(\pi) = \beta_0$
   $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1$
2. Testing $H_0 : \beta_2 = 0$ vs. $H_a : \beta_2 \neq 0$ is equivalent to testing
   $H_0 : \text{logit}(\pi) = \beta_0 + \beta_1 x_1$
   $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
3. Testing $H_0 : \beta_3 = 0$ vs. $H_a : \beta_3 \neq 0$ is equivalent to testing
   $H_0 : \text{logit}(\pi) =$_____
   $H_a : \text{logit}(\pi) =$_____

* R assumes that you enter variables from most important to least in the `formula=` argument

– Partial tests are by far the most common ones.

* Rarely have a prior ordering on the importance of variables
* An exception is polynomials: $\text{logit}(\pi) = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_p x^p$
  · Then higher-order terms are less important than lower order terms.
* Another exception is when there is an ordering to the cost or difficulty of measuring variables.
  · Include the easier ones first and add the harder ones only if they truly help the model

Test Computations

- **FACT: The LR test statistic for comparing nested models can be conducted by comparing the RESIDUAL DEVIANCES of the two models**

  – "Nested models" means that the null model is a reduced version of the alternative model, where certain parameter restrictions have been applied (like setting some parameters to 0)

  * All hypotheses we have talked about so far have model from H_0 nested within model from H_a.
    · H_0 implies a "reduced" version of the "full" H_a model
  * Models $\beta_0 + \beta_1 x_1$ and $\beta_0 + \beta_2 x_2 + \beta_3 x_3$ are NOT nested, for example. There is something each model that is not in the other.

  – "Residual deviance" (often just called "deviance") is a measure sort of like "Error sum of squares" in regression

  * It measures the amount of "information" in Y that is not explained by the fitted model
  * It is related to the log-likelihood of the model
  * The more variables you add to the model, the smaller the residual deviance becomes
  * A model that explains the responses perfectly has residual deviance of 0

        - Estimated probability matches observed proportion of successes at each x when model is fitted to aggregated data
        - A SATURATED MODEL—one that has as many parameters as observations—has this property
        - Certain smaller models may do this as well, by luck
      * The degrees of freedom for the residual deviance are $n - p$, just like in regression[1]
    - Denote the residual deviance for the null-hypotheses model by D_0 and for the alternative-hypothesis model by D_a, where both models are fit to the same form of data. Then the LR test stat can be written as

$$-2\log(\Lambda) = D_0 - D_a$$

      * This gets compared to χ^2_ν, where ν is the difference in residual degrees of freedom between the two models.

- All of the tests in the examples above can be computed by separately fitting the two models in the corresponding null and alternative hypotheses, and taking the resulting differences in deviances.
    - How many DF would they have?
- *All of this is important! Remember it!*

Simultaneous testing of multiple parameters

- When we do tests as model comparisons, nothing restricts us to testing one parameter at a time
    - As long as two models are nested, they can differ in any number of parameters!
    - For example questions about the "significance of the model" ask whether the *entire model* has any explanatory value
      * Tests $H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$.
      * For any p this is tested by comparing
        $H_0 : \text{logit}(\pi) = \beta_0$
        $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$
    - Both models are fit and the LR test stat is found as the difference in residual deviances again.
      * How many DF would the model significance test have?

[1] In the context of logistic regression to aggregated data, "n" refers to the total number of EVPs in the data, resulting in pairs (n_x, w_x) to which data were aggregated, not the total number of trials that were aggregated.

- In certain cases, there are other combinations of variables that you might want to consider removing together at once and then determine whether this change has "significantly harmed" the model fit.
  - For example, some variables may be related to measurements taken by one machine that is expensive to run. Do we need these measurements in our model, or would a simpler, much cheaper set of measurements do nearly as well?
  - Compare nested models
    - Model with all variables (Full model, H_a)
    - Model without "difficult" variables (Reduced model, H_0)
  - DF = # variables removed.

Example: Placekicking (Lecture 8 scripts.R, Placekick.csv)

We now add the binary variable `change` to the model, to see whether the probability that the kick is successful, π, is affected by the pressure on the kicker to change the lead in the game, after accounting for the distance of the kick. To do this, we fit the model

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2,$$

where x_1 is `distance`, x_2 is `change`, and $p = 2$.

The model fit is given below:

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

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7061   0.2282   0.2282   0.3750   1.5649

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.893181   0.333184  17.687   <2e-16 ***
distance    -0.112889   0.008444 -13.370   <2e-16 ***
change      -0.447783   0.193673  -2.312   0.0208 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

```
    Null deviance: 1013.4  on 1424  degrees of freedom
Residual deviance:  770.5  on 1422  degrees of freedom
AIC: 776.5

Number of Fisher Scoring iterations: 6
```

- The estimated logistic regression model is

$$\text{logit}(\hat{\pi}) = 5.89 - 0.113\texttt{distance} - 0.448\texttt{change}$$

  - When `change`=1 (kick changes the lead) the log-odds of success decrease by 0.448 compared to when `change`=0 (no lead change).
    - At first glance, "pressure" reduces the probability of success.
  - The Wald test for this effect ($H_0 : \beta_2 = 0$) has $Z_W = -2.31$ and p-value=0.021. This means that the change in probability (through the log-odds) is significant at $\alpha = 0.05$
    - The sample size is large with lots of successes and failures, so I trust that the Wald test should work fine here.

- Nonetheless, we do LR tests for each parameter next:

```
> # Partial tests (add each variable last)
> Anova(mod.fit2)  # omitting test="LR" (default)
Analysis of Deviance Table (Type II tests)

Response: good
         LR Chisq Df Pr(>Chisq)
distance  218.650  1     <2e-16 ***
change      5.246  1      0.022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> # Sequential tests
> 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    1013.43
distance  1  237.681       1423     775.75   <2e-16 ***
change    1    5.246       1422     770.50    0.022 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Conclusions are the same: both p-values are < 0.05 so we reject each respective $H_0 : \beta_j = 0$ and conclude that both `distance` and `change` are important variables in this model, given that the other is in the model.
- If we wanted to test the significance of the entire model for some reason:
  - We would start by setting up hypotheses as models
    $H_0 : \text{logit}(\pi) = \beta_0$
    $H_a : \text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
  - Then we would fit both models
    - It does not matter whether we fit the models to binary data or aggregated data. However, we must *use the same level of data aggregation for each model fit.*
    - Note that we would need aggregation that is suitable for the full model (here, across combinations of `distance` and `change`)
  - Here is the reduced model, with intercept only, `formula=Y~1`
    - Follow that with `anova(<H0 model>,<Ha model>,test = "Chisq")`

```
> mod.fit0 <- glm(formula=good ~ 1, family=binomial(link=logit),
data=placekick)
> # Note 2 df
> anova(mod.fit0, mod.fit2, test = "Chisq")
Analysis of Deviance Table

Model 1: good ~ 1
Model 2: good ~ distance + change
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      1424     1013.4
2      1422      770.5  2   242.93 < 2.2e-16 ***
```

  - $-2\log(\Lambda) = 242.9$ which gives a tiny p-value on χ^2_2 (2df!). Conclude that the overall model is a significantly better fit than assuming that all probabilities are constant.
    - This test does not tell you *which* variables might be useful. It need not be all of them! So I don't often use it

---

3 What to learn from this

1. Everything. Seriously. This is fundamental to all we will do.

(a) Fitting a logistic regression model to binary, partially aggregated, or fully aggregated (EVP form) gives the same parameter estimates and inferences.

(b) Tests and CIs for a single parameter

i. Score not usually available; use LR where possible

ii. Sequential and partial tests in models with more than one variable

(c) Comparing nested models and residual deviances

2. Even some of the R hints in the scripts are helpful for future use.

4 Appendix: Profile likelihood

- If H_0 pertains to only one parameter, then there are p additional parameters that are not constrained and still need to be estimated
  - When $p = 1$, the likelihood function has a dome-like shape when plotted against β_0 and β_1
  - Suppose we want to test $H_0 : \beta_1 = c$ vs. $H_a : \beta_1 \neq c$
    - So need to refit the model, fixing β_1 at the hypothesized value, and find the MLEs of the remaining parameters under that constraint
      - This gives us the maximized likelihood under H_0
    - Fixing $\beta_j = c$ is like taking a cross-section of the dome, resulting in a concave curve called a LIKELIHOOD PROFILE.
    - Then find the value of β_0 that maximizes this profile
      - The new estimate is likely to be different from the original MLE!
    - Get the value of the likelihood function at the top of the profile
      - Plug that maximized value into the equation for the LR test statistic $-2\log(\Lambda)$.
    - The evaluated likelihood at the MLEs with one parameter fixed is smaller than the unrestricted version unless, by sheer luck, the original MLE $\hat{\beta}_j$ is exactly equal to the hypothesized value.
    - The $\chi^2_{1,1-\alpha}$ critical value tells us how much smaller is too much smaller on the log-likelihood
  - When $p > 1$, the log-likelihood is a higher-dimensional dome-like shape that we can no longer picture, but the concept and process is the same.
    - Fix any parameters at the values specified under H_0
    - Create a profile likelihood curve in the remaining parameter dimensions
    - Maximize that likelihood to obtain the needed numerator for Λ

5 Appendix: Additional details about sequential and partial tests

- Note that the LR test statistics are close to the squares of the Wald test stats from the partial tests, but not quite the same.
  - $2.312^2 = 5.345, -13.370^2 = 178.757$ vs. 5.246, 218.65
  - In large samples the squared Wald and LR statistic grow closer together.
- Note that the last sequential test from any model matches the partial test for the corresponding variable (why??)
  - The remaining partial tests are almost always different from their respective sequential tests
  - Also note that you get different sequential test results for both variables if we reverse the order in the `formula` specification

```
> mod.fit2a <- glm(formula=good ~ change + distance,
family=binomial(link=logit), data=placekick)
> anova(mod.fit2a, 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    1013.43
change    1   24.277      1423     989.15 8.343e-07 ***
distance  1  218.650      1422     770.50 < 2.2e-16 ***
```

  - The partial tests don't change
- We can perform any of these tests by naming the models in `anova(<H0 model>, <Ha model>)`
  - For example, the test for `change` given that distance is in the model:

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

Model 1: good ~ distance
Model 2: good ~ distance + change
```

```
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1      1423     775.75
2      1422     770.50  1   5.2455    0.022 *
```

- Residual deviance for null model (`Model 1`) is $D_0 = 775.75$
- Residual deviance for alternative model (`Model 2`) is $D_a = 770.50$
  - $-2\log(\Lambda) = D_0 - D_a = 5.25$

6 Exercises (due when announced)

Using Professional Dues data from Lecture 7,

1. Compute 95% LR and Wald confidence intervals for the effect of dues increase on the probability that a member decides to quit.

   (a) Interpret the interval, making both a technical statement ("there is 95% confidence...") and a practical one (does the size of increase seem to affect whether people say they will quit?).

2. For our placekick example,

   (a) Fit a model containing all explanatory variables

   (b) We know that `distance` is important. Using a single model-comparison LR test, determine whether *all of the rest* of the variables can be dropped. Use $\alpha = 0.05$.

   (c) Use partial LR tests with $\alpha = 0.05$ to determine which variables seem to contribute significantly to the model.

      i. Given the results of this test, could you use a model with `Distance` and `PAT` only? Explain.

Practice exercises, not marked:

1. What does $\beta_0 = 0$ imply about the probability of success in a logistic regression model?

STATISTICS 475: Applied Discrete Data Analysis

Different Types of Explanatory Variables

(B&L Section 2.2.5–2.2.6)

1 Problem to be solved

- Thus far, we have only studied logistic regression using numerical explanatory variables in linear format
- *Often* explanatories take other formats
  - Transformations of other variables
  - Categorical
  - Interactions
- We need to explore how these different variable types affect the model
  - Hold on tight, it's gonna get tricky...

2 Transformations of x

- The linear shape for the log-odds and, equivalently, the perfect sigmoid for probabilities, is a *model*.
  - It is wrong; is it useful?
- Sometimes the model with linear explanatory variables does not fit well
  - Common in cases where x has a boundary, like at $x \geq 0$ and π is near 0 or 1 there
    - Dosage studies with binary responses:
    - Placekick: shortest possible distance is 18 yards.
  - Sometimes not monotone increasing or decreasing

        * There can be a "sweet spot" where probability of success is optimized
            · Probability that a college basketball player becomes professional against weight.
    - Sometimes the probability curve is just asymmetric for no particular reason

- In these cases, there are a few options:

1. Transform x (often done, studied below)
2. Add polynomial terms (somewhat often done, studied below)
3. Change the model structure away from logistic regression (mostly not studied here)
    (a) Model linear predictor as something other than $\text{logit}(\pi)$ (sometimes done)
        i. Leads to form for π that is something other than $\exp(\cdot)/[1+\exp(\cdot)]$
        ii. See Lecture 13 on Generalized Linear Models
    (b) Use a nonlinear predictor (rarely done)
        i. Uses logit form for π with more complicated function inside exp()
    (c) Use a "statistical learning" method that builds shapes flexibly (now very popular)
        i. STAT 452

2.1 Technical Details

- Transformations of an explanatory variable presents no technical difficulty
    - A transformation is just a function of a variable, like $\log(x), x^2, \exp(x), ...$
    - The model $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$ allows each x_j to be anything
        * Linear terms, transformations, some of each, or whatever
        * x_1 might be the square root of distance, x_2 might be the log of time remaining in the half, and so forth.
        * Just compute the function on a variable, and you have a new variable with a new name
            · Use it in the model.
    - Main issue is interpretation

- For simplicity suppose $p = 1$ with a measured variable x
    - Let $z = g(x)$ for some function g be a transformation of x
    - Then a logistic regression model in z continues to treat $W \sim Bin(n, \pi)$
        * But now $\text{logit}(\pi) = \beta_0 + \beta_z z$
    - As z (i.e., $g(x)$) increases by 1 unit,
        * $\text{logit}(\pi)$ changes by β_z

    - Odds of success change multiplicatively by $\exp(\beta_z)$
  - Absolutely nothing has changed; we just gave a different name to the variable in the model

- *However, change for 1-unit increase in **x** is more complicated to state*
  - When x increases by 1 unit, z changes by $g(x+1) - g(x)$,
    - This may be different for different x
    - Example: if $z = x^2$ then going from 1 to 2 in x is a $2^2 - 1^2 = 3$-unit change in z, but going from 2 to 3 in x is a $3^2 - 2^2 = 5$-unit change in z
  - Complication impacts interpretation of ORs: different estimates at different x
    - Need to separately compute OR as $\text{Odds}_{g(x+1)}/\text{Odds}_{g(x)}$ at each x where you want the computation
    - e.g., write out logits and do subtraction to get coefficients for `mcprofile()`
  - No added computational difficulty, though.

2.2 Popular special case: Polynomials

- Consider $\text{logit}(\pi) = \beta_0 + \beta_1 x + \beta_2 x^2$
- We just have a 2-variable model with $x_1 = x$ and $x_2 = x^2$
- The differences are all in how we use it
  - Tests for importance of terms in model parameters usually done sequentially
    - Simplify more complex powers out of the model first
  - Parameter meanings are conceptual more than real
    - β_1 is the change in logit for 1-unit change in x, *holding x^2 constant.*
    - β_2 is the change in logit for 1-unit change in x^2, *holding x constant.*
    - How do you even do that???
  - Instead, compute total change in *entire* logit for 1-unit change in x:

$$\log(OR) = \log \frac{Odds_{x+1}}{Odds_x} = \text{logit}(\pi_{x+1}) - \text{logit}(\pi_x)$$

$$= \left[\beta_0 + \beta_1(x+1) + \beta_2(x+1)^2\right] - \left[\beta_0 + \beta_1 x + \beta_2 x^2\right] = \beta_1 + \beta_2(2x+1) \qquad (1)$$

    - Depends on value of x at which it is evaluated
    - Could be positive or negative at different x, depending on values of β_1 and β_2
    - $OR = \exp[\beta_1 + \beta_2(2x+1)]$
    - The OR for a c unit change in x is $\exp(c\beta_1 + (2cx + c^2)\beta_2)$ (SHOW THIS!)

- Notice that equation (1) is a linear combination of the model parameters

- It has the form $a_0\beta_0 + a_1\beta_1 + a_2\beta_2$ for some coefficients a_0, a_1, a_2
- For $\beta_1 + \beta_2(2x+1)$, $a_0 = 0, a_1 = 1, a_2 = 2x+1$
- Need these coefficients in order to do LR inference using `mcprofile`
- `emmeans()` can generate coefficients automatically for Wald inference if you know how to ask it

Example: Placekicking (Lecture 10 scripts.R, Placekick.csv)

PROBLEM: Could the model with `distance` be improved by adding `distance`^{2} to the model? Refit and estimate the OR for a *10-unit decrease* of distance at 30, 40, 50, and 60 yards. Use a 95% LR confidence interval.

SOLUTION: First, we refit the model and have a look at the summary (abridged output). Notice how the calculation of `distance`^{2} is handled by performing the calculation inside `I(...)`. This is because the symbol "^" has a special meaning in R formulas. The `I()` function tells R to treat "^" as the usual "power" operation instead.

```
> mod.fit2 <- glm(formula=good ~ distance + I(distance^2),
+                 family=binomial(link=logit), data=placekick)
> summary(mod.fit2)

Call: glm(formula = good ~ distance + I(distance^2),
          family = binomial(link = logit), data = placekick)

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    7.8446831  1.0009079   7.838 4.59e-15 ***
distance      -0.2407073  0.0579403  -4.154 3.26e-05 ***
I(distance^2)  0.0017536  0.0007927   2.212    0.027 *

Number of Fisher Scoring iterations: 6
```

The fitted model is $\text{logit}(\pi) = 7.84 - 0.24 * \texttt{distance} + 0.0018 * \texttt{distance}^2$. Interestingly, the Wald test for the quadratic term suggests that it is useful in the model. (Can we believe the Wald test here? Probably, with the large sample size. The LR test from `Anova()` in the script agrees, with p-value 0.028 (not shown).)

To find the ORs and their CIs corresponding to 10-yard decreases means using the formula $OR = \exp(c\beta_1 + (2cx + c^2)\beta_2)$ with $c = -10$. We cal met `mcprofile()` do the heavy lifting. W just need to construct the appropriate matrix of coefficients for `mcprofile()` to attach to β_0, β_1, and β_2 for each distance, $x = 30, 40, 50, 60$.

```
> all.dist <- 10*c(3:6)
> betas <- coef(mod.fit2)
> #1 yard increase
```

```
> OR.1 <- exp(betas[2] + (2*all.dist+1)*betas[3])
> #10 yard decrease
> OR.10d <- exp(-10*betas[2] + (-10*2*all.dist+(-10)^2)*betas[3])
> cbind(all.dist, round(cbind(OR.1, OR.10d), digits=2))
     all.dist OR.1 OR.10d
[1,]       30 0.87   4.62
[2,]       40 0.91   3.25
[3,]       50 0.94   2.29
[4,]       60 0.97   1.61
>
> # Now find LR confidence interval for these ORs
> library(package = mcprofile)
>
> # Create the coefficient matrix for the parameters in the
> #   0*beta_0 + c*beta_1 + (2*cx+c^2)*beta_2
> K1 <- as.matrix(cbind(0, 1, 2*all.dist+1))
> K10d <- as.matrix(cbind(0, -10, 2*(-10)*all.dist+(-10)^2))
> K10d
     [,1] [,2]  [,3]
[1,]    0  -10  -500
[2,]    0  -10  -700
[3,]    0  -10  -900
[4,]    0  -10 -1100
> # Use the mcprofile(object=, CM=, ...) function to find profile
> #   likelihood values.
> # Can take a little time in complex models.
> profiles.1 <- mcprofile(object=mod.fit2, CM=K1)
> profiles.10d <- mcprofile(object=mod.fit2, CM=K10d)
>
> # LR CI for logit scale
> lrci.logit.1 <- confint(object=profiles.1, level=0.95,
                          adjust = "none")
> lrci.logit.10d <- confint(object=profiles.10d, level=0.95,
                            adjust = "none")
>
> # Exponentiate into OR
> ci.OR.1 <- exp(lrci.logit.1$confint)
> ci.OR.10d <- exp(lrci.logit.10d$confint)
>
> cbind(all.dist, round(cbind(OR.1, ci.OR.1, OR.10d, ci.OR.10d),
                        digits=2))
  all.dist OR.1 lower upper OR.10d lower upper
1       30 0.87  0.85  0.90   4.62  3.17  6.86
2       40 0.91  0.89  0.93   3.25  2.75  3.89
3       50 0.94  0.89  0.98   2.29  1.66  3.19
```

```
 4        60 0.97   0.90   1.05    1.61   0.88   3.00
```

The table shows that the ORs for a 1-yard longer kick increase, becoming closer to 1 as the distance increases. This feature is more apparent in the 10-yard decrease ORs. The closer you are, the more the extra distance helps to increase the odds of success.

For comparison, here is how easy this is using `emmeans()`, although it only gives Wald intervals. This package computes linear predictors and standard errors at the requested values using `emmeans()`, and uses the results in the CIs. See the script for detailed descriptions of the process.

```
> aa = emmeans(mod.fit2, specs= ~distance,
               at=list(distance=10*c(2:6))
> confint(contrast(aa, method="consec", reverse=TRUE),
          type = "response", adjust="none")
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL
 20 / 30        4.62 0.908 Inf     3.142      6.79
 30 / 40        3.25 0.288 Inf     2.735      3.87
 40 / 50        2.29 0.378 Inf     1.657      3.17
 50 / 60        1.61 0.502 Inf     0.876      2.97

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
```

---

3 Interactions

- In linear regression we use interactions to allow the effect of one variable to change depending on the level of the other variable. (See Appendix for details.)
  - Interactions are created by adding cross-products between variables to the regression formula
- We can do the same thing with logistic regression
- Consider the model $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$
  - Again, strict interpretation of each parameter assumes all other variables are held constant
    - Can't do that here with the cross-product interaction term
  - Instead, look at how a c-unit increase in x_1 causes *entire* logit to change, holding x_2 constant:

    $[\beta_0 + \beta_1(x_1 + c) + \beta_2 x_2 + \beta_3(x_1 + c)x_2] - [\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2] = \beta_1 c + \beta_3 c x_2$

    - The "main effect" of the variable is from the usual $\beta_1 c$ term that we would have with x_1 but without the interaction
    - The added interaction term alters the main effect to account for the levels of x_2 at which the effect of x_1 is being measured.
  - Corresponding OR for c unit change in x_1 is $\exp(\beta_1 c + \beta_3 c x_2)$
  - OR involving changes in x_2 found similarly

- With more variables, can have a model with more than one interaction in it
  - All interactions involving a particular variable contribute to that variable's OR.

- Inferences as usual: LR and Wald
  - For tests of terms in model, don't normally test a main effect when interaction with that variable in it is still in the model

- For ORs, need to construct coefficients a_0, a_1, a_2, a_3 that express needed combinations of parameters, $a_0\beta_0 + a_1\beta_1 + a_2\beta_2 + a_3\beta_3$
  - For $\beta_1 c + \beta_3 c x_2$ we get $a_0 = 0, a_1 = c, a_2 = 0, a_3 = cx_2$

Example: Placekicking (Lecture 10 scripts.R, Placekick.csv)

Wind can affect the probability of success for a placekick, and it seems reasonable that the longer a kick is, the more it might be affected by wind. Thus, we might expect to see an interaction between distance and wind. We model this next and explore the effects of wind using ORs at 30, 40, 50, and 60 yards.

First, we need to know how to specify interactions in the `formula=` argument. There are several ways to do this:

- `formula = y ~ x1 + x2 + x1:x2`
  - Write out the interaction term using `:` between variables

- `formula = y ~ x1*x2`
  - The `*` says to create main effects and interactions out of all possible combinations of variables
  - Here, it is `x1*x2 = x1 + x2 + x1:x2`
  - `x1*x2*x3 = x1 + x2 + x1:x2 + x3 + x1:x3 + x2:x3 + x1:x2:x3`

- `formula = y ~ (x1 + x2)^2`
  - Includes all terms in orders up to the power
    - Does *not* actually create squared terms
  - Here, it is `(x1+x2)^2 = x1 + x2 + x1:x2`

- `(x1 + x2 + x3)^2 = x1 + x2 + x1:x2 + x3 + x1:x3 + x2:x3`
  - No 3-way interaction term unless we take `^3` or higher

Recall that `wind` is an indicator variable for conditions where wind speed is at least 15mph (24 kph) at kickoff.

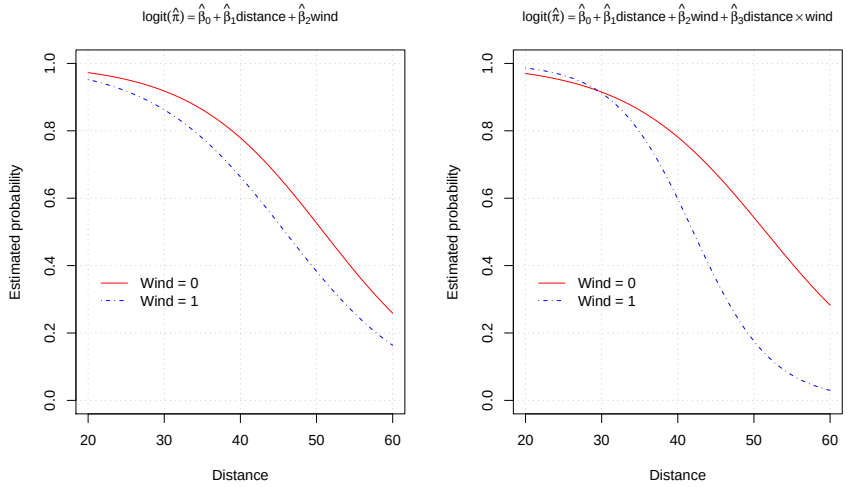
```
> mod.fit.int <- glm(formula=good ~ distance + wind + distance:wind,
                     family=binomial(link=logit), data=placekick)
> summary(mod.fit.int)

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


Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.684181   0.335962  16.919   <2e-16 ***
distance      -0.110253   0.008603 -12.816   <2e-16 ***
wind           2.469975   1.662144   1.486   0.1373
distance:wind -0.083735   0.043301  -1.934   0.0531 .

> library(package = car)
# Would need this if it has not already been used.
> Anova(mod.fit.int, test = "LR")
Analysis of Deviance Table (Type II tests)

Response: good
              LR Chisq Df Pr(>Chisq)
distance       238.053  1    < 2e-16 ***
wind             3.212  1    0.07312 .
distance:wind    5.110  1    0.02379 *
```

The estimated model is $\text{logit}(\pi) = 5.68 - 0.11\texttt{distance} + 2.47\texttt{wind} - 0.084\texttt{distance} : \texttt{wind}$. Since `wind=1` for windy conditions and 0 otherwise, the negative coefficient on the interaction with `distance` means that when `wind=1` (windy conditions), the effect of increasing distance is to reduce the probability of success even more, exactly as we expected. The interaction is significant at $\alpha = 0.05$ according to the LR test, so we can conclude that the wind effect is not the same at all distances. See Figure 1, stolen from Figure 2.6 of the book.

Interesting odds ratios to examine would include:

1. "The wind effects": Find the OR comparing the two levels of wind holding distance fixed at, say, 20, 30, 40, 50, and 60 yards
2. "The distance effects": Find the OR for a 10-unit decrease in distance separately for each level of wind

Figure 1: Plot of probability of successful placekick vs. `distance` separately for each level of `wind`. On left, from model without interaction. On right, from model with interaction. Code is in the program **placekick.R** on the book's website.



Estimating each of these with LR confidence intervals requires knowing what coefficients to use when we form the linear combinations of parameters that are exponentiated into ORs. The rest is exactly as we have done before. I show the math work here to find the needed coefficients, and leave the numerical calculations to the program.

1. To find the OR for comparing the odds of success with windy conditions (`wind`=1) to the odds without windy conditions `wind`=0 at fixed distance d, plug these values into the formulas for the linear predictors:

   - $OR = \text{Odds}_{d,1}/\text{Odds}_{d,0} =$

     $$\exp(\beta_0 + \beta_1 d + \beta_2 1 + \beta_3 d * 1)/\exp(\beta_0 + \beta_1 d + \beta_2 0 + \beta_3 d * 0)$$

   - Focusing on the linear predictors, $\log(\text{Odds}_{d,1}) - \log(\text{Odds}_{d,0}) =$

     $$[\beta_0 + \beta_1 d + \beta_2 1 + \beta_3 d * 1] - [\beta_0 + \beta_1 d + \beta_2 0 + \beta_3 d * 0] = \beta_2 + d\beta_3$$

     So the coefficients on the 4 parameters $(\beta_0, \beta_1, \beta_2, \beta_3)$ are $(0, 0, 1, d)$

2. The OR comparing the odds of success at distance $d - 10$ to the odds of success at distance d at fixed wind level w:

   - $OR = \text{Odds}_{d-10,w}/\text{Odds}_{d,w} =$

     $$\exp(\beta_0 + \beta_1(d-10) + \beta_2 w + \beta_3(d-10) * w)/\exp(\beta_0 + \beta_1 d + \beta_2 w + \beta_3 d * w)$$

   - Focusing on the logits:

     $$\log(OR) = [\beta_0+\beta_1(d-10)+\beta_2 w+\beta_3(d-10)*w]-[\beta_0+\beta_1 d+\beta_2 w+\beta_3 d*w] = -10\beta_1+(-10)w\beta_3$$

- So when `wind`=1, the coefficients on the 4 parameters are $(0, -10, 0, -10)$
- And when `wind`=0, the coefficients on the 4 parameters are $(0, -10, 0, 0)$

The code using `mcprofile()` is in the program. The results are below.

```
    all.dist Estimate OR.low OR.up
C1       20     2.22   0.55 16.06
C2       30     0.96   0.41  2.93
C3       40     0.42   0.19  0.88
C4       50     0.18   0.04  0.59
C5       60     0.08   0.01  0.50

    wind Estimate OR.low OR.up
C1    0     3.01   2.55  3.58
C2    1     6.96   3.40 18.79
```

From the first table, for example, the odds of success for a 60-yard placekick under windy conditions are only 0.08 times as high (92% lower) as they are under non-windy conditions, and we are 95% confident that the interval $(0.01, 0.50)$ covers the true odds ratio. At low distances, the CIs contain 1, so there is not a clear effect of wind, but at 40 yards and beyond, there is a significant decrease in odds of success when it is windy versus when it is not.

The second table shows the effect of moving 10 yards closer under non-windy and windy conditions. Clearly, there is a higher probability of success at shorter distances in both cases (OR and CIs are all above 1), but the effect of distance seems to be greater in windy conditions than otherwise. This is what we expected to see.

As usual, `emmeans()` can do all of the calculations for Wald intervals automatically without the need to input a matrix of coefficients

---

4 Notes

1. There are two fundamental steps to including transformations and interactions into the model
   (a) Writing the model correctly, using the logit
      i. This is just like writing out a linear model!
   (b) Identifying the right combinations of coefficients to make up odds ratios
      i. Identify the two logits to be compared
      ii. Do some algebra
      iii. Alternatively use `emmeans`, but requires understanding package
2. The `wind` explanatory variable is binary, a simple form of categorical variable. We will see next how to handle categorical explanatories with more levels.

5 What to learn from this

1. Unfortunately one thing you need to get good at is minor algebraic manipulations of linear predictors.
   (a) You will frequently need to
      i. Identify two levels of an x_j for comparisons using an OR
      ii. Write out the OR you want as a difference between two linear predictors
      iii. Do the algebra to reduce it to the form $a_0\beta_0 + a_1\beta_1 + \ldots + a_p\beta_p$
      iv. Then feed the calculated coefficients $a_0, a_1, \ldots, a_p$ into an R function to do the computations
   (b) `emmeans` can reduce this necessity, but figuring out how to make it work can take longer than cranking out the algebra.
2. Exercise 1 below gives you some practice
3. You will want to get good at this. We will do it many more times.

6 Appendix: Interpreting Interactions

Proper interpretation of interactions is something that I assume you learned elsewhere, but history has informed me that my assumption is not always accurate. So here is a quick primer on interpreting interactions:

1. An interaction is defined based on *comparing two differences*.
   (a) Specifically, it involves
      i. computing a difference in means (or probabilities, which are means for binary RVs) between two levels of one variable or factor,
      ii. computing this same difference at any two levels of a second variable or factor,
      iii. and determining whether these two differences are the same.
   (b) If the two differences are the same, then there is no interaction—the difference in means between levels of one factor does not depend on the level of the other.
   (c) If two differences are NOT the same, then there IS an interaction.
   (d) In a linear regression, interaction is measured by checking whether the slope for one variable changes across levels of another variable.
      i. A slope is a difference between the mean at x and the mean at $x + 1$
      ii. If this difference depends on the value of z, then there is an interaction between x and z.
   (e) See the interaction example above to see how this works on the "slopes" in the logits.

2. Identifying interactions does not depend on which factor is used for the difference and which factor has its levels changing

   (a) Suppose I use $\mu(x, z)$ to represent a mean (or probability) computed at a particular combination of x and z.

      i. I want to check whether differences in means between two levels of x are the same at both levels of z

   (b) Suppose I fix x and z at two values each at which to compute the means and check the interaction: x_1 and x_2, and z_1 and z_2

   (c) I find values for means $\mu(x_1, z_1), \mu(x_2, z_1)$ at the two values of x for z_1, and $\mu(x_1, z_2), \mu(x_2, z_2)$ at the two values of x for z_2

   (d) The differences in means across levels of x are $\mu(x_1, z_1) - \mu(x_2, z_1)$ and $\mu(x_1, z_2) - \mu(x_2, z_2)$.

   (e) If these differences are different, then we have an interaction

      i. Compare $\mu(x_1, z_1) - \mu(x_2, z_1)$ to $\mu(x_1, z_2) - \mu(x_2, z_2)$.

      ii. Do this by computing the difference!

         A. Is $[\mu(x_1, z_1) - \mu(x_2, z_1)] - [\mu(x_1, z_2) - \mu(x_2, z_2)] = 0$?show that the OR for a c unit change in x is $\exp[(\sqrt{x + c} - \sqrt{x})\beta_1 + c\beta_2]$. (This is not just an exercise to make you do algebra. You need to be able to do these kinds of manipulations all the time in order to create ORs in R.)

      iii. A little rearranging of these terms shows that it is equivalent to ask:

         A. Is $[\mu(x_1, z_1) - \mu(x_1, z_2)] - [\mu(x_2, z_1) - \mu(x_2, z_2)] = 0$?

      iv. This is a comparison of differences in means between two levels of z at both levels of x!

   (f) In other words, if x interacts with z then z interacts with x.

      i. It does not matter which variable you set as your first factor for computing differences and which is your second factor for comparing differences

3. In logistic regression, interaction is measured in the logits, because that is where *differences* make sense.

   (a) We use ORs, not differences, for comparing probabilities.

   (b) The $\exp/(1 + \exp)$ transformation from linear predictor to probabilities creates a nonlinear pattern that makes differences between curves naturally larger or smaller, even when there is no interaction.

4. In our example, we found a wind-by-distance (or distance-by-wind) interaction.

   (a) The effect due to distance is not the same at both levels of wind.

      i. The *logit* with respect to distance has a different slope for the two levels of wind

   (b) Equivalently, the effect due to wind is not the same at all distances

i. The difference in *logits* between the two levels of wind changes as distance changes.

7 Exercises (due when announced)

Compete the following exercises. As always, add proper interpretations on all confidence intervals and tests.

1. Let $z = 1/x$ and consider the model, $\text{logit}(\pi) = \beta_0 + \beta_1 z$. Figure out a formula for $\log(OR)$ in terms of the regression model parameters for c unit change in x. Show the steps similar to what was done for Equation (1) and several other places in this lecture.

   (a) Suppose that an increase of 0.1 units of x is considered to be a meaningful amount for finding an OR. Find the coefficients for computing this OR at $x = 0.5, 1$, and 1.5

2. In the professional dues data, determine whether a quadratic term for is needed in the model for probability of quitting. Support your conclusion with evidence.

3. Using the placekick data, and starting from a model that includes `distance`, `wind`, and `change`, add an interaction to the model between `distance` and `wind`, and `distance` and `change`.

   (a) Does the effect of a potential lead change on the odds of a a successful placekick change significantly depending on the kick's diatance? Report evidence and conclusions to answer this question.

   (b) Estimate the odds ratio for a 5-yard decrease in distance starting at 55 yards (i.e. comparing odds of success at 55 vs. 50 yards) separately for the 4 conditions: windy + lead change, windy + no lead change, no wind + lead change, and no wind + no lead change.

   (c) Find 95% LR confidence intervals. Report the results with the four estimates and CIs.

   (d) Use `emmeans` to compute these estimates along with 95% *Wald* confidence intervals. Report the results.

STATISTICS 475: Applied Discrete Data Analysis

Categorical Explanatory Variables

(B&L Section 2.2.6)

1 Problem to be solved

- *Categorical explanatory variables* are fundamentally different from numerical ones.
  - Cannot simply put βx in the model
  - No concept of "increasing x by 1 unit"
- We now need to address how to handle them, both mathematically and in R
  - In linear regression, we converted categorical variables into sets of binary/indicator/dummy variables
  - We so the same here
  - The biggest issue is interpreting the results.

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

Plant viruses are often spread by insects. This occurs when insects feed on plants already infected with a virus and subsequently become carriers of the virus themselves. When these insects then feed on other plants, they may transmit this virus to these new plants.

To better understand one particular virus, the Tomato Spotted Wilt Virus, and how to control thrips that spread it, researchers at Kansas State University performed an experiment in six greenhouses.[1] One hundred uninfected tomato plants were put into each greenhouse. Within each greenhouse, one of two methods was used to introduce the virus to the clean plants:

1. Additional infected plants were placed among the clean ones, and then "uninfected" thrips were released to spread the virus (coded as `Infest = 1`)

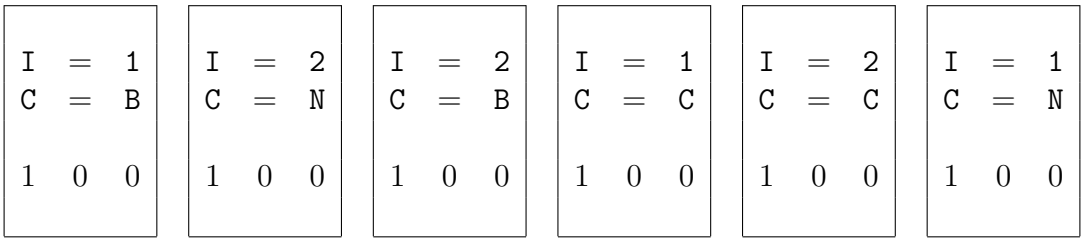
[1] Data courtesy of Drs. James Nechols and David Margolies, Department of Entomology, Kansas State University.

2. Thrips that already carried the virus were released onto the clean plants (`Infest = 2`)

Thus, `Infest` is a categorical variable with 2 levels. Each `Infest` level was assigned to three randomly selected greenhouses. To examine ways of controlling the spread of the virus to plants, the researchers used one of three methods:

1. Biological control — Use predatory mites to attack the thrips (`Control = "B"`)
2. Chemical control — Use a pesticide to kill the thrips (`Control = "C"`)
3. No control (`Control = "N"`)

Thus, `Control` is a categorical variable with three levels. Each `Control` level was randomly assigned to one of the three greenhouses for each `Infest` level. Thus, each of the 6 greenhouse started with 100 clean plants under a different combination of `Infest` and `Control`.



Among the plants that were originally clean, the number displaying symptoms of infection were recorded after 8 weeks for each greenhouse. The experiment was replicated three times, so that there were 18 greenhouses, and each combination infestation method and control method appeared in three greenhouses. Unfortunately, something happened to two of the 18 greenhouses that caused the experiment to fail, so there are only 16 greenhouses in total. Below is a portion of the data where each row of the `tomato` data frame represents a greenhouse. The `stringsAsFactors = TRUE` argument value ensures that the non-numeric variable `Control` is stored as a `factor`-class variable with levels recognized as `B`, `C`, and `N`, rather than as a series of unrelated character strings.

```
> tomato <- read.csv(file = "TomatoVirus.csv",
stringsAsFactors = TRUE)
> 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
```

The main study interest was to compare methods of controlling the spread of virus by killing thrips, and in particular to determine whether the safer (biological) control method was as effective as the less safe chemical control. The different ways of infecting the thrips were included to make sure that the control methods worked under different plausible scenarios.

2 Fitting and Interpreting models with categorical explanatory variables

- Suppose $p = 1$ and X is nominal categorical with q levels (The case of $p = 2$ is handled in the example later.)
  - All trials at the same level of X have the same probability of success (binomial assumptions)
    - Trials at different levels *may* have different probabilities
  - So there are q probability parameters that we want to estimate, say $\pi_1, \pi_2, \ldots, \pi_q$
  - We observe some number of trials and successes at each level of X, so have q proportions, say $p_1, \ldots, p_q$
- We can label the levels from $1, \ldots, q$ or with consecutive letters A,B,..., or whatever
  - These are just labels, can't treat them as numerical
  - But need *something* numerical that we can do regression on
- We can create q binary INDICATOR (DUMMY) variables, $x_1, x_2, \ldots, x_q$
  - For each $j = 1, \ldots, q$, $x_j = 1$ if the observed value of x is the jth level, and $x_j = 0$ if not
  - "Indicates" the jth level
  - Called "one-hot encoding" in computing science and machine learning
- We can try putting all q indicators into a model, $\text{logit}(\pi) = \beta_0 + \beta_1 x_1 + ... + \beta_q x_q$:
  - Notice that when you plug in the 0-1 values of each x_j into this model, something simple comes out ($q = 4$ here)

	Indicator Variable				
X-level	x_1	x_2	x_3	x_4	$\text{logit}(\pi_j) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$
A or 1	1	0	0	0	$\text{logit}(\pi_1) = \beta_0 + \beta_1 * 1 + \beta_2 * 0 + \beta_3 * 0 + \beta_4 * 0 = \beta_0 + \beta_1$
B or 2	0	1	0	0	$\text{logit}(\pi_2) = \beta_0 + \beta_1 * 0 + \beta_2 * 1 + \beta_3 * 0 + \beta_4 * 0 = \beta_0 + \beta_2$
C or 3	0	0	1	0	$\text{logit}(\pi_3) = \beta_0 + \beta_3$
D or 4	0	0	0	1	$\text{logit}(\pi_4) = \beta_0 + \beta_4$

  - Indicators simply "grab" that levels parameter and add it to β_0

  - Regression parameters allow each level of x to have a different probability of success, depending on the values of $\beta_1, \ldots, \beta_q$

- This particular model has problems, however
  - This model has $q+1$ parameters to estimate q logits or probabilities
    - It is OVERSPECIFIED and there are infinitely many solutions
    - e.g., $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) = (0, 1, 2, 3, 4)$ yields exactly the same results as $(1, 0, 1, 2, 3)$ or $(4, -3, -2, -1, 0)$ or...
  - ML estimation will not converge

- Fix the problem by dropping one of the indicators
  - Then have q parameters to estimate q probabilities
    - Mathematically, there is one unique solution to the problem, and each level can still have its own probability
  - The dropped level becomes the "baseline" level (see below)
  - Recall that dropping a variable is equivalent to forcing the parameter to be 0 in the full (overspecified) model

- Which one to drop?
  - Mathematically, it doesn't matter; pick any and adjust interpretations accordingly
  - R will drop the *first* indicator, x_1, when you put a `factor`-class explanatory variable in a `formula`
  - Which one is x_1?
    - R stores factor levels in alphanumeric order according to the level labels.
    - To see the ordering of levels in any factor, use `levels(<factor name>)`
    - Various R commands can reorder the levels if you would rather use a different level as baseline
  - That is, R does this:

	Indicator Variable			
X-level	x_2	x_3	x_4	$\text{logit}(\pi_j) = \beta_0 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$
A	0	0	0	$\text{logit}(\pi_1) = \beta_0$
B	1	0	0	$\text{logit}(\pi_2) = \beta_0 + \beta_2$
C	0	1	0	$\text{logit}(\pi_3) = \beta_0 + \beta_3$
D	0	0	1	$\text{logit}(\pi_4) = \beta_0 + \beta_4$

- From this, you can **interpret the parameters:**
  - β_0 is the log-odds of success *at the first level of* X
    - $e^{\beta_0}/(1+e^{\beta_0})$ is the probability of success at the first level of X

    - This is the "baseline" level against which others are compared.
  - For $j = 2, \ldots, q$, $\beta_j = \text{logit}(\pi_j) - \text{logit}(\pi_1)$,
    - This is the log(OR) between levels j and 1!

- Remember that this is equivalent to retaining the full model

$$\text{logit}(\pi_j) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

  but setting $\beta_1 = 0$.

  - It is sometimes convenient to express the model this way when we need to do some algebra on it to compute ORs and such

- Note that the book (Chris!) renumbers the variables and parameters, calling them $x_{1,} \ldots, x_{q-1}$ instead of $x_2, \ldots, x_q$

  - It doesn't matter what you call them, as long as you structure the problem correctly and remember what they represent

2.1 Odds Ratios

- We measure a *numerical* variable's effect on π using odds ratios

  - How much do odds of success change for c-unit change in x?

- We can form odds ratios within a *categorical* explanatory variable by comparing the odds of success at any two different levels

  - Odds of success at level a vs. odds of success at level b:

$$OR_{a:b} = \frac{\text{Odds}_a}{\text{Odds}_b} = \frac{\exp(\beta_0 + \beta_a)}{\exp(\beta_0 + \beta_b)} = \exp(\beta_a - \beta_b) = \exp[\text{logit}(\pi_a) - \text{logit}(\pi_b)]$$

    - When level 1 is involved in an OR, remember that β_1 is taken to be 0
  - *So odds ratios are just exponentiated differences between the numerator- and denominator-level logits*
    - Reduces here to exponentiating a simple difference in parameters

- We already know how to do Wald and LR inferences on parameters of this form!

  - Confidence interval or test for equal probabilities
    - $H_0 : OR_{a:b} = 1$ tests $H_0 : \beta_a - \beta_b = 0$
  - WALD:
    - Work in the logit scale, taking differences of log-odds to find log(OR)
    - Estimate with $\hat{\beta}_a - \hat{\beta}_b$, which uses a normal approximation in large samples for Wald

    - Get $\sqrt{\widehat{Var}(\hat{\beta}_a - \hat{\beta}_b)}$ mathematically and computationally for Wald
    - `emmeans` will actually estimate the logits instead and work with the asymptotic sampling distribution of $\text{logit}(\pi_a) - \text{logit}(\pi_b)$, but the two approaches are mathematically equivalent
  - LR CI through `mcprofile` uses a coefficient matrix with a column for each parameter (q columns total)
    - First column is for intercept β_0 which always cancels in difference (coefficient $= 0$)
    - Columns $2, \ldots, q$ are the coefficients for $\beta_2, \ldots, \beta_q$
      - No column for β_1 because it doesn't really exist.
    - To find CI for $\beta_a - \beta_b$:
      - Enter 1 in the a-th position
      - Enter -1 in the b-th position
      - Enter 0 everywhere else
    - e.g.,
$$\frac{\text{Odds}_1}{\text{Odds}_2} = \frac{\exp(\beta_0 + \beta_1)}{\exp(\beta_0 + \beta_2)} = \exp(\beta_1 - \beta_2) = \exp(-\beta_2)$$
      - Put -1 in column 2 and 0 everywhere else.
  - Exponentiate endpoints of the resulting confidence intervals

- *Very* often test for a "variable X effect": do *any* of the levels of X have different probabilities of success?
  - $H_0 : (\beta_1 =)\beta_2 = \beta_3 = \ldots = \beta_q = 0$ versus $H_a :$ Not all $\beta_j = 0$
  - LR model comparison test
    - How many df?

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

First, note that in each greenhouse we have 100 plants, and each one is either infected with virus or not. Thus we have a count W of "successes"—infected plants—in 100 trials for each combination of `Infest` and `Control`. It has potential to be binomial, although, given the way the study was run, specifically the spatial layout of plants, I am a little worried about the equal probability and independence assumptions within a greenhouse. Do you see why? (If one plant is infected, which other plants are most likely to be infected?)

Assuming that concerns these are either not a (serious) problem or that we can handle them later (see Section 5.3 of the book), we can fit a binomial model for π, the probability that a plant gets infected within a greenhouse. One issue that we have here is that `Infest` is stored as numbers, so it will automatically be treated as numerical in R's `formula`. We can fix that by turning it into a factor using `tomato$Infest <- factor(tomato$Infest)`. We therefore have two categorical explanatory variables, which I shall generically refer to as X for `Infest` and Z for `Control`.

- We want to examine the effects of both the infestation method and the control method on the proportion of infected plants in a greenhouse.
- *Conceptually*, we want to fit a model like,

$$\text{logit}\ (\pi) = \beta_0 + \text{“}X\text{”} + \text{“}Z\text{”}$$

  where "X" and "Z" are not real terms, but concepts that represent the effects of factors for infestation and control methods, respectively.

  - This is essentially how we will phrase the model in R:

$$\texttt{formula} = \texttt{Virus8/Plants} \sim \texttt{Infest} + \texttt{Control}$$

    where both `Infest` and `Control` are factor-class variables.
- *Mathematically*, the factor concepts need to be replaced with appropriate indicators
- "X" will be represented by two indicator variables: x_1 for the first level of X (`Infest= "1"`, where infected plants and clean thrips are added) and x_2 for the second level of X (`Infest= "2"`, where infected thrips are added).
  - However, we know that R will drop the first indicator, so "X" will be represented in the model only by x_2.
- Similarly, R will arrange the three levels of "Z" in the order `B`, `C`, `N`.
  - Therefore, these can be represented by three dummy variables, say z_B, z_C, z_N, using letters instead of numbers so we can better connect indicators to their meaning
  - R will drop the first one, so only z_C and z_N appear in the model.
- Thus, the model that R will fit is, $W \sim Bin(100, \pi)$, where

$$\text{logit}\ (\pi) = \beta_0 + \beta_2^I x_2 + \beta_C^C z_C + \beta_N^C z_N,$$

  - the superscript I or C on β just helps us keep straight which parameters are from which variables.
- We expect to see 4 parameters estimated.

		Indicator Variables					Model
x-level	z-level	x_1	x_2	z_B	z_C	z_N	$\text{logit}(\pi_{ij}) = \beta_0 + \beta_2^I x_2 + \beta_C^C z_C + \beta_N^C z_N$
1	B	1	0	1	0	0	$\text{logit}(\pi_{1B}) = \beta_0$
1	C	1	0	0	1	0	$\text{logit}(\pi_{1C}) = \beta_0 + \beta_C^C$
1	N	1	0	0	0	1	$\text{logit}(\pi_{1N}) = \beta_0 + \beta_N^C$
2	B	0	1	1	0	0	$\text{logit}(\pi_{2B}) = \beta_0 + \beta_2^I$
2	C	0	1	0	1	0	$\text{logit}(\pi_{2C}) = \beta_0 + \beta_2^I + \beta_C^C$
2	N	0	1	0	0	1	$\text{logit}(\pi_{2N}) = \beta_0 + \beta_2^I + \beta_N^C$

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

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -4.288   -2.425   -1.467    1.828    8.379

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.6652     0.1018  -6.533 6.45e-11 ***
Infest2       0.2196     0.1091   2.013   0.0441 *
ControlC     -0.7933     0.1319  -6.014 1.81e-09 ***
ControlN      0.5152     0.1313   3.923 8.74e-05 ***
--- Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 278.69  on 15  degrees of freedom
Residual deviance: 183.27  on 12  degrees of freedom
AIC: 266.77

Number of Fisher Scoring iterations: 4
```

- Note that R represents the indicators x_2, z_C, z_N using the factor names `Infest` and `Control` along with their levels, so this part is actually easy to understand if you have done the preliminary work above!
  - The baseline level here is `Infest="1"` and `Control="B"`, which you can guess by noting which levels are absent in the coefficients.
- The estimated logistic regression model is

$$\text{logit}(\hat{\pi}) = -0.67 + 0.22\texttt{Infest2} - 0.79\texttt{ControlC} + 0.52\texttt{ControlN},$$

  - $\hat{\beta}_0 = -0.67$, so the log odds of a virus infection with `Infest="1"` (release uninfected thrips and scatter infected plants) and `Control="B"` (biological control with mites) are -0.67
    - Correspondingly, the probability of virus infection is $\exp(-0.67)/(1+\exp(-0.67)) = 0.34$ from `plogis(coef(mod.fit)[1])`

- $\hat{\beta}_2^I = 0.22$, so the log odds of virus infection are estimated to be 0.22 higher with `Infest` level 2 (release infected thrips on uninfected plants) than with level 1 (release uninfected thrips and scatter infected plants)
- $\hat{\beta}_C^C = -0.79$, so the log odds of virus infection with chemical control (level `C` of `Control`) are estimated to be 0.79 lower than with biological control (level `B` of `Control`).
- What about the interpretation of $\hat{\beta}_N^C$?

- Although I don't show them here, tests and confidence intervals for these parameters could be computed in the usual way
  - Test for the overall effect of different methods of infestation or the overall effect of different types of control would be tested using model-comparison LR tests from `Anova()` in `car`.

---

3 Interactions with Categorical Variables

- Modeling interactions among categorical variables is quite common (more so than with numerical variables)
  - Interaction asks whether differences between levels of one variable change at different levels of the other variable.
  - Understanding this is often a main goal for an analysis
    - e.g., entomologists want to know whether the comparison of biological control to chemical control changes depending on the method used to create the infestation
    - Does a certain strong treatment work better on patients with severe disease than those with a mild case?

- We model interactions involving categorical explanatories as cross-products among *all* indicators for the two variables involved
  - e.g., suppose X has 3 levels and Z has 4 levels
    - Cross-products of x_2, x_3 with z_2, z_3, z_4
    - Model becomes

$$\begin{aligned}\text{logit}(\pi) &= \beta_0 + \beta_2^X x_2 + \beta_3^X x_3 + \beta_2^Z z_2 + \beta_3^Z z_3 + \beta_4^Z z_4 \\ &\quad + \beta_{22}^{XZ} x_2 z_2 + \beta_{23}^{XZ} x_2 z_3 + \beta_{24}^{XZ} x_2 z_4 + \beta_{32}^{XZ} x_3 z_2 + \beta_{33}^{XZ} x_3 z_3 + \beta_{34}^{XZ} x_3 z_4\end{aligned}$$

    - β^X and β^Z parameters are called MAIN EFFECT PARAMETERS
    - β^{XZ}parameters are INTERACTION PARAMETERS

- *To understand meaning of parameters, make a table* (here shown with 2×2 levels)

		Indicator Variables		Model
x-level	z-level	x_2	z_2	$\text{logit}(\pi_{ij}) = \beta_0 + \beta_2^X x_2 + \beta_2^Z z_2 + \beta_{22}^{XZ} x_2 z_2$
A	C	0	0	$\text{logit}(\pi_{00}) = \beta_0$
A	D	0	1	$\text{logit}(\pi_{01}) = \beta_0 + \beta_2^Z$
B	C	1	0	$\text{logit}(\pi_{10}) = \beta_0 + \beta_2^X$
B	D	1	1	$\text{logit}(\pi_{11}) = \beta_0 + \beta_2^X + \beta_2^Z + \beta_{22}^{XZ}$

- From this, you can interpret the parameters:
  - β_0 is the log-odds of success at the first level of *both* x and z
  - The main-effect parameter, $\beta_2^X = \text{logit}(\pi_{10}) - \text{logit}(\pi_{00})$, is the log(OR) between levels B and A of x, *fixing z at its first level, C*
  - The main-effect parameter, $\beta_2^Z = \text{logit}(\pi_{01}) - \text{logit}(\pi_{00})$, is the log(OR) between levels D and C of z, *fixing x at its first level, A*
  - The interaction parameter, β_{22}^{XZ}, can be interpreted in a few equivalent ways:
    - *Always*, an interaction between two variables means that the effect of one variable changes depending on the level of the other variable
    - In the model, an "effect" is measured by the difference in logits (log(OR)) for comparing two levels of one variable
    - Then an interaction term compares the difference in log(OR)'s between two levels of one variable (the "effect" of that variable) at the two levels of the other variable
      - It is a difference of two differences, $(a - b) - (c - d)$
    - You can freely reverse which variable is which in this interpretation
    - $\beta_{22}^{XZ} = [\text{logit}(\pi_{11}) - \text{logit}(\pi_{10})] - [\text{logit}(\pi_{01}) - \text{logit}(\pi_{00})]$ is the difference between the log(OR) for Z when $X = B$ and the log(OR) for Z when $X = A$
    - $\beta_{22}^{XZ} = [\text{logit}(\pi_{11}) - \text{logit}(\pi_{01})] - [\text{logit}(\pi_{10}) - \text{logit}(\pi_{00})]$ is ALSO the difference between the log(OR) for X when $Z = D$ and the log(OR) for X when $Z = C$
- Higher order interactions (involving 3 variables, 4 variables, etc) are possible, but become cumbersome to interpret

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

I add interaction term to the model we fit previously to see whether the effect of the control method is associated with the infestation method.

- Make cross-products between all X indicators (x_2) and all Z indicators (z_B, z_C)
  - We have 2 new terms, $x_2 z_C$ and $x_2 z_N$

– New model is, $W \sim Bin(100, \pi)$, where

$$\text{logit}\ (\pi) = \beta_0 + \beta_2^I x_2 + \beta_C^C z_C + \beta_N^C z_N + \beta_{2C}^{IC} x_2 z_C + \beta_{2N}^{IC} x_2 z_N$$

- This adds two parameters: β_{2C}^{IC} and β_{2N}^{IC} (show this in a table to understand interpretations!).

The model fit is below (abridged output):

```
> 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)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.0460     0.1316  -7.947 1.92e-15 ***
Infest2            0.9258     0.1752   5.283 1.27e-07 ***
ControlC          -0.1623     0.1901  -0.854    0.393
ControlN           1.1260     0.1933   5.826 5.68e-09 ***
Infest2:ControlC  -1.2114     0.2679  -4.521 6.15e-06 ***
Infest2:ControlN  -1.1662     0.2662  -4.381 1.18e-05 ***

    Null deviance: 278.69  on 15  degrees of freedom
Residual deviance: 155.05  on 10  degrees of freedom AIC: 242.55

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

Response: Virus8/Plants
               LR Chisq Df Pr(>Chisq)
Infest            4.060  1     0.0439 *
Control          91.584  2  < 2.2e-16 ***
Infest:Control   28.224  2  7.434e-07 ***
```

The estimated logistic regression model is

$$\begin{aligned}\text{logit}(\hat{\pi}) &= -1.05 + 0.93\texttt{Infest2} - 0.16\texttt{ControlC} + 1.13\texttt{ControlN} \\ &\quad -1.21\texttt{Infest2}\times\texttt{ControlC} - 1.17\texttt{Infest2} \times \texttt{ControlN}.\end{aligned}$$

(The parameter estimates are less important than how they get combined into logits and ORs.)

A LRT tests the importance of the interaction term. The hypotheses are $H_0 : \beta_{2C}^{IC} = \beta_{2N}^{IC} = 0$ vs. $H_a : \beta_{2C}^{IC} \neq 0$ and/or $\beta_{2N}^{IC} \neq 0$. Equivalently, we could also write the hypotheses in terms of model comparisons:

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

The test statistic is $-2\log(\Lambda) = 28.224$, and the p-value is 7.4×10^{-7} using a χ^2_2 approximation (why 2 df?). Thus, there is strong evidence of an interaction between the infestation and control methods.

3.1 Odds ratios with interactions

- An interaction in the model implies that the odds ratios that compare two levels of a given variable are not the same for all levels of the other variable
- Therefore, need to estimate odds ratios for two levels of X *separately for each level of* Z and vice-versa.
  - Write out the OR you want to find in terms of logits for comparing the odds of success with level i_1 of X to level i_2 of X at some fixed level j of Z:

$$\begin{aligned} \text{Odds}_{x=i_1,z=j}/\text{Odds}_{x=i_2,z=j} &= \exp[\text{logit}(\pi_{\boldsymbol{i_1}j}) - \text{logit}(\pi_{\boldsymbol{i_2}j})] \\ &= \exp[\ (\beta_0 + \beta_{i_1}^X + \beta_j^Z + \beta_{i_1 j}^{XZ}) - (\beta_0 + \beta_{i_2}^X + \beta_j^Z + \beta_{i_2 j}^{XZ})] \\ &= \exp[\beta_{i_1}^X - \beta_{i_2}^X + \beta_{i_1 j}^{XZ} - \beta_{i_2 j}^{XZ}] \end{aligned}$$

  - If any of i_1, i_2, or j are 1 (the baseline level), then the corresponding parameter is 0.
    - Otherwise, this tells you the position and values of coefficients needed to create the coefficient matrix for `mcprofile()`to estimate ORs and to find confidence intervals:
      - $+1$ at the numerator (i_1) level of X
      - -1 at the denominator (i_2) level X
      - $+1$ on the interaction term for the numerator (i_1) level of X at the fixed level of Z
      - -1 on the interaction term for the denominator (i_2) level of X at the fixed level of Z
      - 0 on all other parameters in the model
  - Similar work tells you how to compute the OR for two levels of Z at a given level of X

- `emmeans()` does this with function arguments, without having to create a vector of coefficients for *each* different OR you want to estimate!
  - Maybe at this point, it is worth learning `emmeans()`...

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

With the highly significant interaction effect, we explore the possible ORs and their interpretations at each level of the other variable. The ORs we can compute here are:

1. Comparisons of the different levels of infestation at each different control level
   (a) `Infest2` vs. `Infest1` at `ControlB`
      i. `Infest1` and `ControlB` are both first levels of their respective variables, so parameters don't exist
      - +1 on `Infest2`
      - −1 on `Infest1` (which can be ignored)
      - +1 on `Infest2:ControlB` (which can be ignored)
      - −1 on `Infest1:ControlB` (which can be ignored)
      - 0 on remaining 5 parameters in the model

   (b) `Infest2` vs. `Infest1` at `ControlC`

   (c) `Infest2` vs. `Infest1` at `ControlN`

2. Comparisons of the different levels of control at each level of infestation
   (a) `ControlC` vs. `ControlN` at `Infest1`

   (b) `ControlB` vs. `ControlN` at `Infest1`

   (c) `ControlC` vs. `ControlB` at `Infest1`

   (d) `ControlC` vs. `ControlN` at `Infest2`
      i. All of these levels are estimated by the model (no first levels)
      - +1 on `ControlC`
      - −1 on `ControlN`
      - +1 on `Infest2:ControlC`
      - −1 on `Infest2:ControlN`
      - 0 on remaining 2 parameters in the model.

   (e) `ControlB` vs. `ControlN` at `Infest2`

   (f) `ControlC` vs. `ControlB` at `Infest2`

Churning out all of these coefficients in a matrix (9 rows and 6 columns) is a pain in the butt and prone to error. Therefore, when it comes to inference on ORs with categorical explanatories, I use the Wald methods available with `emmeans`.

The `emmeans()` function relies on the fact that the estimated linear predictor has a sampling distribution that is reasonably well approximated by a normal distribution in many cases. It produces estimated mean values of the linear predictor at values/levels of explanatory variables specified by the user, or averaged across levels if not specified. Follow-up helper functions can automatically turn these logits into estimated probabilities, odds ratios, and any other quantities that can be found from logits, and can perform hypothesis tests and confidence intervals for these quantities.

The basic syntax is `emmeans(object=<>, specs= ~<> + <> + ...)`. The `object=` is the model fit object, and `specs=` lists the variables whose combinations you want to compute logits for. We can use `summary()` on the resulting `emmeans` object to see what has been created, and then use `confint()` to get confidence intervals for certain quantities. Both of these functions, and a few others, have numerous arguments that we will highlight as we need them. All of these functions do WAY more than what we will need to use for this course, so their documentation can be daunting.

First we show the results from `emmeans()` for our model with interaction. Here, we want to create odds ratios on combinations of `Infest` and `Control`, so our `specs= ~Control+Infest`. We show the summary of the linea predictors, and then add `type="response"` to the `summary()` function to get these results converted into the response (probability) scale.

```
> library(emmeans)
> # Create predicted logit values for levels of variables that
    we want to compare
> # specs = control+infest creates means at all combinations
    of these variables
> emm1 = emmeans(mod.fit.inter, specs= ~Control+Infest)
> summary(emm1)
 Control Infest emmean    SE  df asymp.LCL asymp.UCL
 B       1       -1.05 0.132 Inf    -1.304    -0.788
 C       1       -1.21 0.137 Inf    -1.477    -0.939
 N       1        0.08 0.142 Inf    -0.197     0.357
 B       2       -0.12 0.116 Inf    -0.347     0.107
 C       2       -1.49 0.149 Inf    -1.786    -1.201
 N       2       -0.16 0.142 Inf    -0.438     0.118

Results are given on the logit (not the response) scale.
Confidence level used: 0.95
> summary(emm1, type="response")
 Control Infest  prob     SE  df asymp.LCL asymp.UCL
 B       1      0.260 0.0253 Inf     0.214     0.313
 C       1      0.230 0.0243 Inf     0.186     0.281
 N       1      0.520 0.0353 Inf     0.451     0.588
 B       2      0.470 0.0288 Inf     0.414     0.527
```

```
 C         2       0.183 0.0223 Inf      0.144      0.231
 N         2       0.460 0.0352 Inf      0.392      0.529

Confidence level used: 0.95
Intervals are back-transformed from the logit scale
```

The first summary table shows the variable levels, the estimated linear predictor (`emmean`) its standard error, and a confidence interval for the logit. The `type="response"` converts this to the probability scale, which is convenient.

- Next, we ask for estimated odds ratios and CIs by using the `contrast()` function. In this function, `method=` determines what comparisons will be made. There are many options, and I figured out what we needed somewhat by trial and error.
  - The `"pairwise"` argument value takes all pairwise differences *between each row and the rows below it.* This makes comparisons like $\text{logit}(\pi_{B1}) - \text{logit}(\pi_{C1})$, $\text{logit}(\pi_{B1}) - \text{logit}(\pi_{N1})$, $\text{logit}(\pi_{B1}) - \text{logit}(\pi_{N1})$, and so forth through $\text{logit}(\pi_{C2}) - \text{logit}(\pi_{N2})$. These are all log odds ratios, where the first term is the numerator and the second is the denominator.
  - A similar option, `"revpairwise"` does the same thing but makes pairs in the reverse direction. This can be a lot of comparisons, not all of which might be useful.
  - The `by=` argument allows you to limit the comparisons to those where the variable named as the argument value is held constant. We use this because we want to compare levels of `control` within each level of `infest` and vice versa. We are not presently interested in ORs where we change levels of both variables at the same time.
  - I show in the scripts what happens if we don't use this argument to limit output.

```
> # Conf Ints for ORs comparing infestation effects at each
    Control
> #   adjusting for multiple comparisons
> confint(contrast(emm1, method="revpairwise", by="Control",
          type="response"))
Control = B:
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL
 2 / 1         2.524 0.442 Inf     1.790      3.56

Control = C:
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL
 2 / 1         0.752 0.152 Inf     0.505      1.12

Control = N:
 contrast odds.ratio    SE  df asymp.LCL asymp.UCL
```

```
 2 / 1         0.786 0.158 Inf     0.531      1.16

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
>
> # Conf Ints for ORs comparing effects of different control
    methods for each infestation method
> #   adjusting for multiple comparisons
> confint(contrast(emm1, method="pairwise", by="Infest",
          type="response"))
Infest = 1:
 contrast odds.ratio     SE  df asymp.LCL asymp.UCL
 B / C         1.176 0.2236 Inf     0.753     1.837
 B / N         0.324 0.0627 Inf     0.206     0.510
 C / N         0.276 0.0543 Inf     0.174     0.438

Infest = 2:
 contrast odds.ratio     SE  df asymp.LCL asymp.UCL
 B / C         3.950 0.7458 Inf     2.538     6.149
 B / N         1.041 0.1906 Inf     0.678     1.599
 C / N         0.264 0.0543 Inf     0.163     0.427

Confidence level used: 0.95
Conf-level adjustment: tukey method for comparing a family of 3
estimates
Intervals are back-transformed from the log odds ratio scale
```

Interpretations of ORs and CIs: Note that the output is arranged so that, for each level of the `by=` variable, all comparisons are made among all other variables. Comparisons are given in "/" notation to divide the numerator and denominator odds. So, for example, we first see that

1. Under biological control (`Control=B`), the estimated odds of virus infection when infected thrips are released (`Infest=2`) are 2.52 times as high as the odds of virus infection when uninfected thrips are released with some infected plants (`Infest=1`). We are 95% confident that the true odds ratio is covered by the interval 1.79 to 3.56. This interval excludes 1, so a Wald test would reject the null hypothesis that the odds of virus infection do not depend on how the thrips become infected when biological control is used. The estimates and intervals comparing the two levels of `Infest` for the other levels of `Control` are interpreted similarly.

2. When uninfected thrips are released upon infected plants (`Infest=1`), the estimated odds of virus infection using chemical control (`Control=C`) are 0.28 times as high as when using no control (`Control=N`). We are 95% confident that the true odds ratio is covered by the interval 0.17 and 0.44. This interval excludes 1, so a Wald test would

reject the null hypothesis that the odds of virus infection do not depend on whether chemical or no control is used when thrips are released in this way. Based on these estimates, the use of chemical control reduces the odds of viral infection by about 72% (CI 55%–83%) compared to using no control.

See the exercises for estimating all of the remaining odds ratios from the Tomato Virus example.

The code for LR intervals using `mcprofile()` is given in the script. You will see how much more work has to be done to produce the intervals than with `emmeans()`.

An important point about the intervals produced by both methods is that they are *adjusted for multiplicity by default*. This means that the intervals have been made deliberately wider than necessary so that one can make a statement of having 95% confidence that ALL of the intervals in a group cover their respective parameters. This is good practice. However, the default can be deactivated by adding the argument value `adjust="none"` to the `confint()` function.

---

4 Notes

1. Of course it is possible to have both categorical and numerical variables in the same model.
   (a) One can then have categorical-by-numerical interactions
   (b) Use the tables and the fundamentals shown above to break down how odds ratios can be estimated.
2. In addition to estimating odds ratios and testing whether they are 1, one can formally compare the odds ratios at two levels of another factor
   (a) For example, the `ControlC`/`ControlN` odds ratio could be computed separately at `Infest1` and `Infest2`, and then a test for equality of these ORs could be done.
   (b) Testing $H_0 : OR_1 = OR_2$ can be done by testing $OR_1/OR_2 = 1$, or $\log(OR_1) - \log(OR_2) = 0$.
   (c) This is computed based on further differences of logits.
      i. New set of coefficients for `mcprofile()`
      ii. In `emmeans`, need to compute `confint()` on a new object created by `contrast(...interact`
         A. See script for example.

5 What to learn from this

1. Categorical explanatories are easily added to models

   (a) Important to understand that R deletes first level
   (b) *Write out the model and create the table of indicators to be sure of what is happening!*

2. Interactions complicate matters
   (a) AGAIN, START WITH THE MODEL AND THE TABLE

3. LR intervals for ORs require doing some math
   (a) Carefully write out the OR you want to estimate
   (b) Convert to log(OR) to make differences of linear predictors
      i. Easy algebra!
   (c) Figure out the parameters involved
   (d) Convert to a set of coefficients on all parameters
   (e) Requires some care—Use the tables that I gave to help you get started until you get comfortable with writing models freehand.

4. Wald intervals are much easier to find, once you figure out a few nuances of `emmeans()`
   (a) There are so many options that it is easy to go wrong.
   (b) You should be able to check the results manually, to see if they make sense

6 Exercises (due when announced)

Compete the following exercises. As always, add proper interpretations on all confidence intervals and tests.

1. Book Ch 2 Exercise 18 (a,b,d,e).
   (a) In (e) include LR confidence intervals and interpret.
   (b) Repeat with Wald intervals. Instead of repeating the interpretation, comment on any similarities or differences from LR.

In addition, here are exercises that will not be marked, but that you can do for practice:

1. Compute the coefficients on the regression parameters for estimating all of the remaining odds ratios from the Tomato Virus example.