Logistic Regression

The Logistic Regression Model

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

$y = 1$ if the event of interest occurs; $y = 0$ otherwise

The Logistic Regression Model

$$\ln\left(\frac{P(y=1)}{1-P(y=1)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

$y = 1$ if the event of interest occurs; $y = 0$ otherwise

Let me introduce you to the logistic regression model. This model is used when the response variable has a binary outcome. That is, when the dependent variable has only two possible outcomes. Typically these outcomes are labeled 0 and 1 where the 1 represents the outcome that we are particularly interested in.

Many times the outcome of interest, the one labeled as 1, is often the occurrence of a particular event (like recurrence of of cancer for a patient in remission, default on a loan, or a sale is made). This event may also be that the experimental unit has a particular characteristic (the subject is a homeowner, the subject has Type 1 diabetes, or the student passed the class).

The natural logarithm of the odds of y taking on the value 1 is

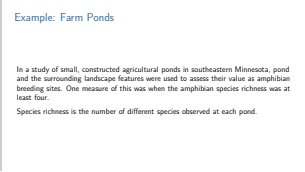
Example: Farm Ponds

In a study of small, constructed agricultural ponds in southeastern Minnesota, pond and the surrounding landscape features were used to assess their value as amphibian breeding sites. One measure of this was when the amphibian species richness was at least four.

Species richness is the number of different species observed at each pond.

Logistic Regression

Example: Farm Ponds



bottom panel note: Agricultural Ponds Support Amphibian Populations, by Knutson, et. al., Ecological Applications, 14(3), 2004, pp. 669–68.

Example: Farm Pond Variables

Dependent Variable

RICH = 1 if species richness is at least 4; RICH = 0 otherwise

Independent Variables

FISH = 1 if fish are present; FISH = 0 otherwise

TOTNITR = total nitrogen in mg/L

Example: Farm Pond Variables

Dependent Variable

RICH — 1 if species richness is at least 4; RICH — 0 otherwise

Independent Variables

FISH — 1 if fish are present; FISH — 0 otherwise

TOTNITR — total nitrogen in mg/L

In the data file farmponds.rda, the variables RICH, FISH, and TOTNITR are defined as shown here. This data file contains quite a few variables, but the presence of fish and the amount of total nitrogen on the pond had an effect on whether or not the pond had a species richness of 4 or more, as we'll see shortly.

R Code for Farm Pond Example

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
summary(rich.out)
```

Logistic regression output in R is obtained using the glm function with the option family="binomial." As with linear regression, the output can be stored in a R object so that various aspect can be pulled out using other R functions on that object, such as the general summary, as shown here.

Farm Ponds: Logistic Regression Output

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.451      1.452   3.066  0.00217 **
## FISH          -4.039      1.387  -2.912  0.00359 **
## TOTNITR       -4.195      1.794  -2.338  0.01937 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 50.446  on 39  degrees of freedom
## Residual deviance: 25.591  on 37  degrees of freedom
```

Farm Ponds: Logistic Regression Output

Farm Ponds: Logistic Regression Output

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.451      1.452   3.066  0.00217 **
## FISH          -4.039      1.387  -2.912  0.00359 **
## TOTNITR       -4.195      1.794  -2.338  0.01937 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 50.446  on 39  degrees of freedom
## Residual deviance: 25.691  on 37  degrees of freedom
```

In this portion of the output, we get the estimated regression coefficients - the estimated intercept, the coefficient for FISH, and the coefficient for total nitrogen.

We also get standard errors for each coefficient, which measure the variability of each estimate and are used to construct the Wald test statistics for individual hypothesis tests to determine of each coefficient is significantly different from 0. The test statistic, z, for the Wald test is simply the estimated coefficient divided by its standard error and it follows a standard normal distribution.

Standard normal distributions with the degrees of freedom shown are used to obtain the p-values shown. Low p-values indicate evidence that the coefficients in the population are truly different from 0.

Farm Pond - The Estimated Model

Here is the estimated model, with the coefficients from the R output.

$$\ln\left(\frac{\widehat{P(y=1)}}{1-P(y=1)}\right) = 4.451 - 4.039 \cdot \text{FISH} - 4.195 \cdot \text{TOTNITR}$$

Farm Pond Example - Interpreting Coefficients of Categorical Predictors

Exponentiating the coefficients yields **odds ratios**. Using the estimated coefficient of FISH, we have

$$e^{-4.039} = 0.018$$

The odds of having species richness of at least 4 at a pond where fish are present are only 1.8% as large as the odds of species richness being at least 4 at a pond where fish are not present, given that total nitrogen is held constant. That is, they are 98.2% less!

Logistic Regression

Farm Pond Example - Interpreting Coefficients of Categorical Predictors

Farm Pond Example - Interpreting Coefficients of Categorical Predictors

Exponentiating the coefficients yields **odds ratios**. Using the estimated coefficient of FISH, we have

$$e^{-4.030} = 0.018$$

The odds of having species richness of at least 4 at a pond where fish are present are only 1.8% as large as the odds of species richness being at least 4 at a pond where fish are not present, given that total nitrogen is held constant. That is, they are 98.2% less!

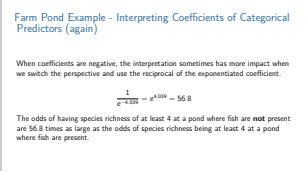
When interpreting exponentiated logistic regression coefficients for categorical factors, the wording is exactly like what we used previously when we discussed odds ratios.

Farm Pond Example - Interpreting Coefficients of Categorical Predictors (again)

When coefficients are negative, the interpretation sometimes has more impact when we switch the perspective and use the reciprocal of the exponentiated coefficient.

$$\frac{1}{e^{-4.039}} = e^{4.039} = 56.8$$

The odds of having species richness of at least 4 at a pond where fish are **not** present are 56.8 times as large as the odds of species richness being at least 4 at a pond where fish are present.



The sign of the coefficient is determined by what we choose to let the 1 represent in the indicator function for fish presence. Had we said that the absence of fish in the pond was defined as 1, then coefficient would have been positive 4.039 and we would have the result shown here directly.

Farm Pond Example - Interpreting Coefficients of Quantitative Predictors

Using the estimated coefficient of TOTNITR, we have

$$e^{-4.195} = 0.015$$

The odds of having species richness of at least 4 decrease by 98.5% for each additional mg/L of total nitrogen.

Farm Pond Example - Interpreting Coefficients of Quantitative Predictors

Using the estimated coefficient of TOTNITR, we have

$$e^{-4.190} = 0.015$$

The odds of having species richness of at least 4 decrease by 98.5% for each additional mg/L of total nitrogen.

Interpreting exponentiated logistic regression coefficients for quantitative predictors is just a little different that it is for categorical ones. It is a little bit similar to the way partial slopes are interpreted for ordinary linear regression, only it refers to the change in odds for each 1 unit increase in the value of the independent variable.

More R Code for Farm Pond Example

Confidence intervals for the odds ratios are available via the following R code.

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
exp(confint(rich.out))
```

```
##                    2.5 %       97.5 %
## (Intercept) 9.1726211883 4004.9580787
## FISH        0.0005135249    0.1712424
## TOTNITR     0.0001404314    0.2486600
```


More R Code for Farm Pond Example

Confidence intervals for the odds ratios are available via the following R code.

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
exp(confint(rich.out))
```

```
##                    2.5 %       97.5 %
## (Intercept) 9.1726211883 4004.9680787
## FISH        0.0005135249    0.1712424
## TOTNITR     0.0001404314    0.2486600
```

Recall that confidence intervals that do not contain 1 can also be taken as significant evidence that a relationship exists between that particular predictor and the response variable.

The function confint pulls the intervals for the estimated coefficients from the output object and raising e to the power of each results in the confidence interval bounds for the corresponding odds ratios.

bottom panel note:

Section 12.8 of he Ott text has the mathematical details of why this is true.

Farm Pond Example: Predicted Probabilities

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
newdata <- data.frame(FISH=0,TOTNITR=0.8)
predict(rich.out,newdata,type="response")
```

```
##         1
## 0.7492629
```

Farm Pond Example: Predicted Probabilities

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
newdata <- data.frame(FISH=0,TOTNITR=0.8)
predict(rich.out,newdata,type="response")
```

```
##         1
## 0.7492629
```

Suppose we wanted to use out fitted model to predict the probability of a farm pond having a species richness of at least 4 for a pond with no fish and 0.8 mg/L of total nitrogen. First, a new data frame must be created with these values for the variables FISH and TOTNITR, then the R function "predict" can be used with the option type="response" along with the object containing the model output, here labeled rich.out. That pond has a probability of .749 of having at least 4 amphibian species living there.

Classification Using Logistic Regression

$$\text{If } P(y = 1) \geq 0.5 \rightarrow \text{Classify as } y = 1$$
$$\text{If } P(y = 1) < 0.5 \rightarrow \text{Classify as } y = 0$$

Since the estimated probability of a farm pond having a species richness of at least 4 for a pond with no fish and 0.8 mg/L of total nitrogen is 0.749, a pond with those characteristics would be classified as having species richness of at least 4.

Classification Using Logistic Regression

If $P(y-1) \geq 0.5 \rightarrow$ Classify as $y-1$
If $P(y-1) < 0.5 \rightarrow$ Classify as $y-0$

Since the estimated probability of a farm pond having a species richness of at least 4 for a pond with no fish and 0.8 mg/L of total nitrogen is 0.749, a pond with those characteristics would be classified as having species richness of at least 4.

While the data in the existing data set is already classified, the logistic regression model can be used to estimate the classification for elements of that population based on the values of the predictor variables.

The typical cutoff probability is 0.5 or above to classify a subject with a particular set of characteristics as having y=1. That is, having the outcome of interest, which in this case is having species richness of at least 4.

This cutoff value can be adjusted to suit the user for a variety of applications and depending on how sensitive you want the classification to be or if the event you are classifying is relatively rare or fairly common.

Confidence Intervals for Predicted Probabilities

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
newdata <- data.frame(FISH=0,TOTNITR=0.8)
out <- predict(rich.out, newdata, se.fit=TRUE)
C = .95  # define the level of confidence
crit = qnorm(1-(1-C)/2)  # get the appropriate critical value
lower = exp(out$fit-crit*out$se.fit)/(1+exp(out$fit-crit*out$se.fit))
upper = exp(out$fit+crit*out$se.fit)/(1+exp(out$fit+crit*out$se.fit))
c(lower,upper)
```

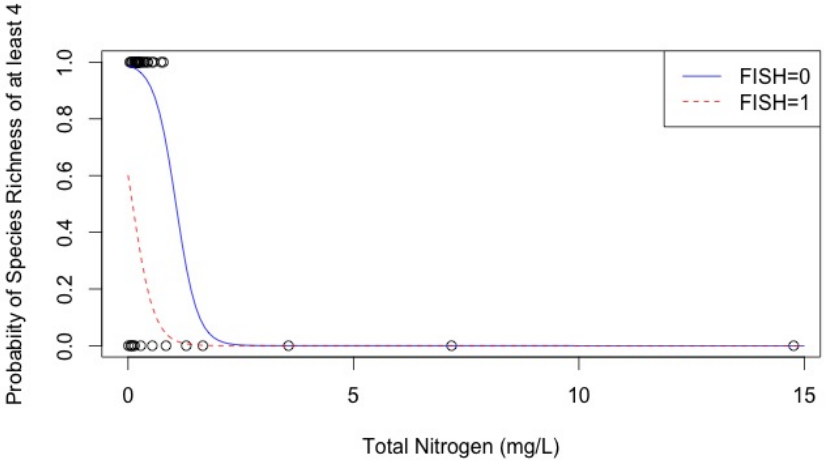
```
##         1         1
## 0.3523169 0.9425807
```

Confidence Intervals for Predicted Probabilities

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
newdata <- data.frame(FISH=0,TOTNITR=0.8)
out <- predict(rich.out, newdata, se.fit=TRUE)
C = .95  # define the level of confidence
crit = qnorm(1-(1-C)/2)  # get the appropriate critical value
lower = exp(out$fit-crit*out$se.fit)/(1+exp(out$fit-crit*out$se.fit))
upper = exp(out$fit+crit*out$se.fit)/(1+exp(out$fit+crit*out$se.fit))
c(lower,upper)

##         1         1
## 0.3523169 0.9425807
```

A confidence interval for the predicted probability can be constructed using R code like this here. Without the option type="response" that we used before, the predict value comes out on the logit scale. By requesting the standard error of the predicted logit value, we can somewhat manually compute the confidence intervals for the predicted probabilities. So with 95% confidence, the probability of a pond with no fish and 0.8 mg/L of total nitrogen will have a probability between .35 and .94 of having at least 4 amphibian species present.

Farm Pond Example: Plotting Predicted Probabilities



Logistic Regression

Farm Pond Example: Plotting Predicted Probabilities

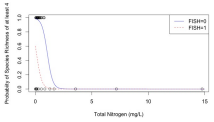
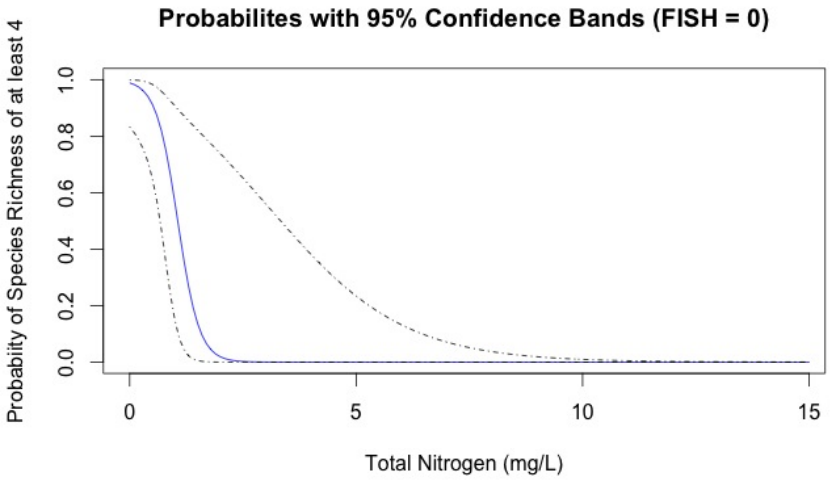


Figure 1:

R can be used to plot the probability of the pond having species richness of at least 4 across the range of values for total nitrogen for ponds where fish are present as well as for those where fish are not present. The original data points have been plotted along with the curve of the predicted probabilities

Recall the probability from the previous slide, when fish are not present, so FISH $= 0$, and total nitrogen is 0.8 mg/L was .749, which would be about here on this plot. Since TOTNITR has a negative coefficient in the estimated model, the probability of success (that is having species richness of at least 4) declines as total nitrogen increases, and these probabilities are even lower when fish are present. Fish eat amphibian larvae and tadpoles.

Confidence Bands for Predicted Probabilities



Confidence Bands for Predicted Probabilities

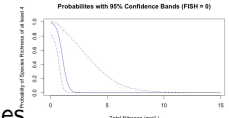


Figure 2:

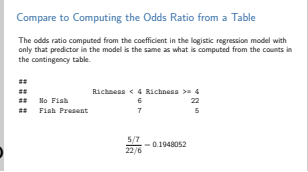
bottom panel note: Look in the Lesson_7 Storybook.Rmd file to see the R code for making this plot.

Compare to Computing the Odds Ratio from a Table

The odds ratio computed from the coefficient in the logistic regression model with only that predictor in the model is the same as what is computed from the counts in the contingency table.

```
##
##                  Richness < 4 Richness >= 4
##   No Fish                   6             22
##   Fish Present              7              5
```

$$\frac{5/7}{22/6} = 0.1948052$$


Compare to Computing the Odds Ratio from a Table

The odds ratio computed from the coefficient in the logistic regression model with only that predictor in the model is the same as what is computed from the counts in the contingency table.

```
##
##                Richness < 4 Richness >= 4
##   No Fish                 6             22
##   Fish Present            7              5
```

$$\frac{5/7}{22/6} = 0.1948052$$

Recall how an odds ratio is computed from a contingency table. In this case, since we want the ratio of odds of having species richness of at least 4 for ponds where fish are present to ponds with no fish, the numerator will be the odds of high species richness when fish are present, which is 5 to 7, or 5 divided by 7.

These odds are then divided by the odds of having high species richness at ponds where there are no fish, which is 22 to 6, or 22 divided by 6. So the odds ratio carried out to 7 decimal places (for dramatic effect here) is 0.1948052.

Compare to Computing the Odds Ratio from a Table

The odds ratio computed from the model is:

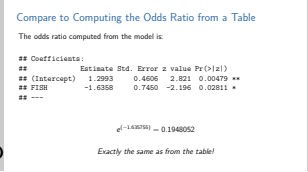
```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.2993     0.4606   2.821  0.00479 **
## FISH         -1.6358     0.7450  -2.196  0.02811 *
## ---
```

$$e^{(-1.635755)} = 0.1948052$$

Exactly the same as from the table!

Logistic Regression

Compare to Computing the Odds Ratio from a Table



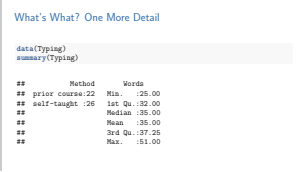
When the logistic regression model for species richness contains only FISH, the exponentiated coefficient yields the odds ratio exactly as it is computed from the contingency table.

Keep in mind that if other predictors, such as total nitrogen, are included in this model, the coefficient of FISH will be affected, so the odds ratio from the model will no longer match exactly the value computed from the table, but will be the odds ratio when total nitrogen has been taken into account.

What's What? One More Detail

```
data(Typing)
summary(Typing)
```

```
##           Method       Words
##  prior course:22   Min.   :25.00
##  self-taught :26   1st Qu.:32.00
##                    Median :35.00
##                    Mean   :35.00
##                    3rd Qu.:37.25
##                    Max.   :51.00
```



Suppose you have a data set where the categorical variable you want to model as the response variable in a logistic regression model and its values are characters. Consider the Typing data frame, for example. The variable Method is a Factor type of variable that contains the values "prior course" and "self-taught".

If Method is used as the response variable in a logistic regression model, which outcome will R assign to be the category where y=1 and which one will be y=0?

What's What? One More Detail

```
## Call:
## glm(formula = Method ~ Words, family = "binomial", data = Typing)

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.18763    2.93538   2.449   0.0143 *
## Words       -0.20059    0.08355  -2.401   0.0164 *
## ---
```

What's What? One More Detail

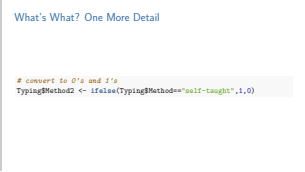
```
## Call:
## glm(formula = Method ~ Words, family = "binomial", data = Typing)

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.18763    2.93538   2.449   0.0143 *
## Words       -0.20059    0.08355  -2.401   0.0164 *
## ---
```

Here is the output for the logistic regression model of typing method regressed against the number of words typed per minute. But can you tell which outcome for Method R chose to assign the value of 1? I can't either!

However, experience with R has taught me that R will go in alphabetical order in assigning the 0 and 1, so it should be "prior course" is 0 and "self-taught" is 1, since P comes before S.

What's What? One More Detail

```
# convert to 0's and 1's
Typing$Method2 <- ifelse(Typing$Method=="self-taught",1,0)
```



Using the ifelse function a new variable can be added to the data frame that assigns a 1 to every case where the typist was self-taught and a 0 for those who had a prior course.

What's What? One More Detail

[video showing data table in R viewer]

What's What? One More Detail

[video showing data table in R viewer]

The best way to see what just happened is to open the data frame in the R viewer and see that the 0's and 1's are aligned with the prior course and self-taught designations.

What's What? One More Detail

```
## Call:
## glm(formula = Method2 ~ Words, family = "binomial", data = Typing)

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.18763    2.93538   2.449   0.0143 *
## Words       -0.20059    0.08355  -2.401   0.0164 *
## ---
```

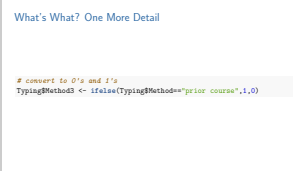
What's What? One More Detail

```
## Call:
## glm(formula = Method2 ~ Words, family = "binomial", data = Typing)

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.18763    2.93538   2.449   0.0143 *
## Words       -0.20059    0.08355  -2.401   0.0164 *
## ---
```

Here is the output for the logistic regression model using the self-taught category as the outcome of interest. Note that the coefficient of Words is the same as when we ran the model with the words as the observed responses. This confirms that R chose to assign the 0's and 1's in alphabetical order.

What's What? One More Detail

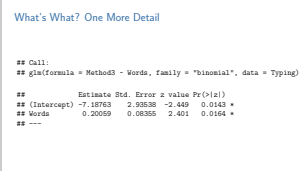
```
# convert to 0's and 1's
Typing$Method3 <- ifelse(Typing$Method=="prior course",1,0)
```



Just for fun let's see what happens if we use the prior course as the outcome of interest and label that with a 1 and label self-taught as a 0.

What's What? One More Detail

```
## Call:
## glm(formula = Method3 ~ Words, family = "binomial", data = Typing)

##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.18763    2.93538  -2.449   0.0143 *
## Words        0.20059    0.08355   2.401   0.0164 *
## ---
```



How does the output compare? The intercept is identical, the p-values are the same as before, but something is different. What is it?

That's it! The sign on the coefficient of Words is now positive instead of negative. Do you remember what effect the sign has on an exponent? Do you recall the lesson on odds ratios? If we switch the 0's and 1's, that is, when we use prior course as the outcome of interest - the one assigned as y=1, we end up with the reciprocal of the odds ratio that we had when self-taught was the outcome assigned as y=1.