**Section 2.2.6 – Categorical explanatory variables**

As we saw earlier with change in the placekicking example, a categorical explanatory variable with two levels can be represented simply as a binary variable to reflect these two levels.

When there are q levels (where q ≥ 2), q – 1 indicator variables (i.e., binary variables) are needed to represent the variable – just like in normal linear regression. Notice this is not q binary variables.

Example: Categorical explanatory variable with 4 levels (4levels.R)

This program is not directly referenced in the book. It is available on my course website.

Suppose an explanatory variable has levels of A, B, C, and D. Three indicator variables can represent the explanatory variable in a model:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Indicator variables | | |
| Levels | x1 | x2 | x3 |
| A | 0 | 0 | 0 |
| B | 1 | 0 | 0 |
| C | 0 | 1 | 0 |
| D | 0 | 0 | 1 |

Notice how each level of the explanatory variable has a unique coding. The logistic regression model is



Informally, we could also write out the full model as



where it is assumed that B, C, or D are corresponding indicator variables in the model.

For example, A is the base level, and it is represented in the model with x1 = x2 = x3 = 0 so that



For category B, the model becomes



Thus, β1 measures the effect of level B when compared to level A.

R’s treatment of categorical explanatory variables

R has a *factor* class type for explanatory variables. By default for this class, R orders the levels within a factor alphabetically, where numbers are ordered before letters and lowercase letters are before uppercase letters:

0, 1, 2, …, 9, …, a, A, b, B, …, z, Z

To see the ordering of any factor, the levels() function can be used. This ordering of levels is important because R uses it to construct indicator variables with the “set first level to 0” method of construction. This is what I used previously in the 4 level categorical example. The “set last level to 0” construction is used by some other software packages, like SAS.

There is one exception to the ordering given above. One can re-order the levels so that, for example, B comes before A.

Example: Categorical explanatory variable with 4 levels (4levels.R)

Below is a simple factor to help demonstrate R’s coding of the levels:

> cat.var <- factor(x = c("D", "A", "A", "B", "D", "C"))

> cat.var

[1] D A A B D C

Levels: A B C D

Notice how R prints an ordering of the levels.

We will most often see categorical variables in a data frame.

> set1 <- data.frame(cat.var = cat.var)

> set1

cat.var

1 D

2 A

3 A

4 B

5 D

6 C

> class(set1$cat.var)

[1] "factor"

> levels(set1$cat.var)

[1] "A" "B" "C" "D"

> contrasts(set1$cat.var)

B C D

A 0 0 0

B 1 0 0

C 0 1 0

D 0 0 1

There are two main ways to choose a different ordering of the levels.

* The relevel() function to define a new base level.

> set1$cat.var2 <- relevel(x = set1$cat.var, ref = "D")

> set1

cat.var cat.var2

1 D D

2 A A

3 A A

4 B B

5 D D

6 C C

> levels(set1$cat.var2)

[1] "D" "A" "B" "C"

> contrasts(set1$cat.var2)

A B C

D 0 0 0

A 1 0 0

B 0 1 0

C 0 0 1

* The factor() function to re-order all levels.

> set1$cat.var3 <- factor(x = set1$cat.var, levels = c("D", "A", "B", "C"))

> set1

cat.var cat.var2 cat.var3

1 D D D

2 A A A

3 A A A

4 B B B

5 D D D

6 C C C

> levels(set1$cat.var3)

[1] "D" "A" "B" "C"

We can re-name the levels using the factor() function as well. Please see the corresponding program.

Additional comments:

* Because change in the placekicking example was already coded as 0 or 1, R just simply treats it as a numerical variable rather than a factor.
* There are times when a categorical variable is coded numerically. For example, suppose a variable named “x” has levels of 1, 2, and 3 where 3 is not necessarily “greater than” 1 in its meaning. R will treat this by default as a numerical variable when estimating a model. To have R create indicator variables for x, either
* use factor(x) in a formula argument or
* create a new variable in the data frame with the factor() function.
* R will use a different ordering for a factor levels only when set through code. A perhaps unexpected occurrence of this can occur when working with a contingency table and then transforming it to a data frame. For example, suppose the following contingency table is created:

c.table <- array(data = c(1, 2, 3, 4), dim = c(2,2), dimnames = list(Rows = c("B", "A"), Columns = c("Success", "Failure")))

R treats Rows as having an ordering of B before A. Through a short set of code (see the Larry Bird exercise in the book), one can transform this contingency table to a data frame format so that the glm() function can be used to fit a logistic regression model to it. The corresponding explanatory variable will be a factor with levels of B and A in this order.

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

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

To better understand the Tomato Spotted Wilt Virus and how to control thrips that spread it, researchers at Kansas State University performed an experiment in a number of greenhouses. One hundred uninfected tomato plants were put into each greenhouse, and they were introduced to the virus (infestation method) in one of two ways:

1. Interspersing additional infected plants among the clean ones, and then releasing "uninfected" thrips to spread the virus (Infest = 1)
2. Releasing thrips that carry the virus (Infest = 2)

To control the spread of the virus to plants, the researchers used one of three methods:

1. Biologically through using predatory spider mites (Control = B)
2. Chemically using a pesticide (Control = C)
3. None (Control = N)

The number of plants displaying symptoms of infection were recorded for each greenhouse after 8 weeks. Below is the data where each row of the data set represents a greenhouse:

> tomato <- read.csv(file = "C:\\data\\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

From row 1 of the data set, we see that 21 out of 100 originally uninfected plants in a greenhouse showed symptoms after 8 weeks, and these plants had infestation method #1 applied while trying to control the thrips with a chemical application.

The stringsAsFactors = TRUE argument value is very important in read.csv(). R automatically will recognize any variable with letter values as a factor class type.

Both the Control and Infest explanatory variables are categorical in nature. Below is how R treats Control:

> class(tomato$Control)

[1] "factor"

> levels(tomato$Control)

[1] "B" "C" "N"

> contrasts(tomato$Control)

C N

B 0 0

C 1 0

N 0 1

Thus, the B level of Control is the base level, and there are two indicator variables for representing levels C and N. There is some justification to change the base level to N to help with upcoming comparisons (compare biological or chemical to doing nothing), but we will use the default ordering here.

As for the Infest variable, notice it is coded numerically. Because the variable has only two levels, this will not affect our use of it in R. However, if it had more than two levels, we would need to transform the class of it to a factor through using the factor() function. Below is the corresponding code:

> class(tomato$Infest)

[1] "numeric"

> levels(tomato$Infest)

NULL

> class(factor(tomato$Infest))

[1] "factor"

> levels(factor(tomato$Infest))

[1] "1" "2"

> contrasts(factor(tomato$Infest))

2

1 0

2 1

For demonstration purposes, I will change the Infest variable to be a factor in the data frame:

> tomato$Infest <- factor(tomato$Infest)

> class(tomato$Infest)

[1] "factor"

and use this for the remainder of the data analysis.

We estimate the model with both Infest and Control:

> mod.fit <- glm(formula = Virus8/Plants ~ Infest +

Control, family = binomial(link = logit), data =

tomato, weight = 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

> mod.fit$xlevels # Another way to see the levels

$Infest

[1] "1" "2"

$Control

[1] "B" "C" "N"

Because the response variable is given in a binomial form, we used the weight argument along with the success/trials formulation in the formula argument. The estimated logistic regression models is



R automatically creates terms based on the variable name and the level of categorical variable. For example, ControlC and ControlN corresponds to what we saw earlier for Control:

> contrasts(tomato$Control)

C N

B 0 0

C 1 0

N 0 1

When stating the model, it is o.k. to use simply “C” and “N” for control as well.

Based on the positive estimated regression parameter for Infest2, the probability of showing symptoms is estimated to be larger in greenhouses when infestation method #2 is used. Also, based on the estimated regression parameters for Control, the estimated probability of showing symptoms is lowest for the chemical control method and highest for when no control method is used. These interpretations rely on there not being an interaction between the explanatory variables in the model. We will examine shortly how to include the possibility of the interaction along with evaluating its importance.

Hypothesis tests for categorical explanatory variables

As with normal linear regression, all indicator variables must be included in a hypothesis test to evaluate the importance of a categorical explanatory variable.

Consider again the example with the categorical explanatory variable having 4 levels:



To evaluate the importance of this explanatory variable, we need to test

H0: β1 = β2 = β3 = 0

Ha: At least one β not equal to 0

Three separate Wald tests of H0: βi = 0 vs. Ha: βi ≠ 0 are not appropriate. One could do one overall Wald test, but we will focus instead on using a LRT because it performs better.

Interactions including categorical explanatory variables

To form the interaction, multiply each indicator variable by the model terms representing the other explanatory variable(s)

Questions:

* How would you represent an interaction between a categorical explanatory variable with 4 levels and one continuous explanatory variable?
* How would you perform a test to evaluate the interaction in the previous example question?
* How would you represent an interaction between a categorical explanatory variable with 4 levels and another categorical explanatory variable with 3 levels?

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

In actual application, it is important to consider the possibility of an interaction between the infestation method and the control method. The code below shows how to include this interaction and evaluate its importance:

> mod.fit.inter <- glm(formula = Virus8/Plants ~ Infest +

Control + Infest:Control, family = binomial(link =

logit), data = tomato, weight = Plants)

> summary(mod.fit.inter)

Call:

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

Deviance Residuals:

Min 1Q Median 3Q Max

-3.466 -2.705 -1.267 2.811 6.791

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 \*\*\*

---

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: 155.05 on 10 degrees of freedom

AIC: 242.55

Number of Fisher Scoring iterations: 4

The estimated logistic regression model with the interaction is



A LRT to evaluate the importance of the interaction term tests the regression parameters corresponding to Infest2×ControlC and Infest2×ControlN, say β4 and β5. The hypotheses are

H0: β4 = β5 = 0

Ha: At least one β not equal to 0

Notice that H0 is the model that includes the main effect terms only.

> 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 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

The test statistic is -2log(Λ) = 28.224, and the p-value is 7.4×10-7 using a  approximation. Thus, there is strong evidence of an interaction between infestation method and control.

Questions:

* How would you perform the test using anova()?
* What hypotheses do the p-value for Infest in the Anova() output correspond to?