**Logistic Regression Practice**

**Tam Nguyen**

**Part A: Using MedGPA to work with untabulated data**

1. Open the **MedGPA** data set (from the Stat2Data) in R and use head(MedGPA) to investigate the data.
2. **Acceptance: M to F**
   1. Based on this sample of 55 students, what are the odds of a woman getting accepted to med school? (xtabs(), tally(), or table() will be helpful here!)

10/18 = 0.556

* 1. Based on this sample of 55 students, what are the odds of a man getting accepted to med school?

15/12 = 1.25

* 1. What is the odds ratio of women to men?

0.44

* 1. Interpret the odds ratio.

The odds of picking a female student getting accepted to a med school is 44.4% of the odd of picking a male student getting accepted to a med school.

1. **Acceptance: MCAT scores**

The variable “MCAT” contains the total MCAT scores for these 55 students.

1. The code below will calculate the proportion of successes within each MCAT score:

tab <- xtabs(~MCAT+Acceptance,data=MedGPA)

prop <- tab[,2]/(tab[,2] + tab[,1])

1. Now plot these proportions against the MCAT scores:

xyplot(prop~sort(unique(MedGPA$MCAT)), xlab="MCAT")

1. Does it look like a logistic regression model will be helpful here? Why or why not?
2. **Checking Linearity**

Notice (from tab) that most of our cells contain 0, 1, or 2. We could do the “trick” again and add 0.5 to all the successes and failures, but another way of dealing with it is the procedure described on p. 472- 474 (Example 9.12), which is what we’ll use below…

* 1. First, divide the range of MCAT scores into intervals with roughly equal numbers of cases. I have made (what I think is) a judicious division below into 5 intervals (however, there is no “right” number of intervals in these cases!).

Fill in the rest of the table, using tab to help you.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Group # | # Cases | Range of MCAT | Mean of range | Admitted | | Proportion admitted | Odds of admission | Odds Ratios |
| Yes | No |
| 1 | 11 | 18 – 32 | 30 | 2 | 9 | 0.18 | 0.22 |  |
| 3.64 |
| 2 | 9 | 33 – 34 | 33.89 | 4 | 5 | 0.44 | 0.8 |
| 1.875 |
| 3 | 10 | 35 – 36 | 35.5 | 6 | 4 | 0.6 | 1.5 |
| 1.067 |
| 4 | 13 | 37 – 39 | 38.46 | 8 | 5 | 0.61 | 1.6 |
| 3.125 |
| 5 | 12 | 40 – 48 | 42.08 | 10 | 2 | 0.83 | 5 |
|  |

* 1. Interpret the odds ratio for an increase in MCAT score from the 35 – 36 range to the 37 – 39 range.
     + The odds of picking a student who has MCAT in range 37-39 getting accepted to a med school is 93.75% of the odd of picking a student who has MCAT in range 35-36 getting accepted to a med school.
  2. Looking at the odds ratios, does it look like the logistic model’s linearity constraint is reasonable? Why or why not?
  3. Create two vectors from the table above: “means”, that contains the range means; and “odds”, that contains the odds from each group. You can create these vectors easily using (for example)

means <- c(25, 33.5, …)  
odds <- c(2/9, 4/5, …)

* 1. Use   
     plot(log(odds)~means, type="o")

to make a scatterplot of log(odds) vs. MCAT score. What are we looking for in this plot? Does it look like a logistic regression model is appropriate in this case? Why or why not?

\*\*IMPORTANT NOTE! Everything we did in parts (a) – (e) was to assess if a linear model was appropriate. It was NOT necessary for what we’re about to do: fit the logistic regression model. (R pays no attention to your groupings, and knows how to deal with 0 cells.)

1. **Fitting a Logistic Regression Model in R: untabulated data**

FINALLY! We’re going to see how we actually fit the model!

To fit the logistic regression model, we use the glm() function, which stands for “general linear model”:

glm(*response* ~ *explanatory*, data=*data*, family=binomial)

Note “family=binomial”: this is how you tell R that you want a logistic regression. If you leave this off, R will NOT fit a logistic model!

1. Fit a logistic regression model with Acceptance as the response variable and MCAT as the explanatory variable. Save this model as “mcat.log” for future use.
2. What is the slope of the fitted model?
3. Compute *e*^(slope). What is the interpretation of this quantity?
4. **Visualizing the Logistic Model: 2 ways**
   1. On the “linearized” plot of log(odds) vs. MCAT (from part #5a), you can plot the (straight-line) slope, as in Figures 9.12 and 9.13. To do this,

plot(log(odds)~means, xlab="MCAT") #Note the use of plot, not xyplot!

abline(reg=mcat.log) #add the regression line from mcat.log to the plot

Does it look like this model is a good fit to the data?

* 1. On the “original” plot of proportion vs. MCAT (from part a), you can plot the curvy logistic model, as in Figures 9.3 and 9.14 (upper-right corner). To do this we use the function makeFun():

xyplot(prop~sort(unique(MedGPA$MCAT)), xlab="MCAT")

fit <- makeFun(mcat.log)

plotFun(fit(MCAT)~MCAT, add=TRUE)

Does it look like this model is a good fit to the data?

**Hey!** Why does the model look inappropriate in (a) but appropriate in (b)?

\*\*Remember: The process of grouping is for visualization purposes *only*: it has nothing to do with actually fitting the model. Ironically, however, by taking groups we can (unintentionally) visually alter the distribution of the data. This is why the straight-line model in part (i) looks like it has the wrong slope: our selection of groups has resulted in points plotted in certain places, and those places are a little “off.” If you want to see a more accurate picture of the linearized plot in part (i), try the code below:

plot(log(prop/(1-prop))~sort(unique(MedGPA$MCAT)),xlab="MCAT",ylab="log(odds)", ylim=c(-3,3))

abline(reg=mcat.log)

(Unfortunately, several of the data points are missing in this plot because of division by zero or trying to take the natural log of zero…which is why we grouped in the first place!)

**Part B: Using cancer data to work with tabulated data**

In the MCAT file, the data was in “long form”: one row for each individual. We tabulated it just to visualize it and fill in our table. ***What if the data is already tabulated into a table*** (“short form”), and we don’t have the long form?

1. Below is a table of various types of cancer, along with whether the patient survived at least a year. Add a row to this table that gives the odds of survival for each type of cancer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Breast | Bronchus | Colon | Ovary | Stomach |
| Died (within 1 year) | 2 | 14 | 8 | 3 | 9 |
| Survived (at least 1 year) | 9 | 3 | 9 | 3 | 4 |
| Odds of survival | 4.5 | 0.75 | 1.125 | 1 | 0.44 |

1. Linearity

Linearity is irrelevant here. What we have is a binary response variable (survival) and a *categorical* response variable (type of cancer). We’re not fitting a line from “left” to “right” because there is no ordering of these cancer types! (The ordering you see above is, of course, completely arbitrary.) Because it’s categorical, the logistic regression model will fit individual “lines” to each cancer type, and two points always make a line.

1. **Create the table in R**

cancer.tab <- matrix(data=c(2,9,14,3,8,9,3,3,9,4),nrow=2, ncol=5) #this creates the table, but without any variable labels

rownames(cancer.tab) <-c("Died","Survived") #labels the rows

colnames(cancer.tab) <-c("Breast", "Bronchus", "Colon", "Ovary", "Stomach") #labels the columns

cancer.tab #check out the pretty table!

1. **Fitting the Logistic Model in R: tabulated data**

First, we need to change the tabulated data to what we call “semi-tabulated”: there is not a separate row for each individual, but there are separate rows for those who survived and those who died. We do this using the ‘melt’ function, which lives in the ‘reshape2’ package.

1. Install and load the ‘reshape2’ package now.
2. To create the semi-tabulated version of the data, type

melt(cancer.tab)

Make sure you understand what has been done here! ‘Var1’ now contains information about survival (the response variable) and ‘Var2’ is the type of cancer (the explanatory variable). The ‘value’ variable contains the frequencies (counts) within each of those sub-groups.

To fit the logistic regression model to semi-tabulated data, we add one argument (“weights”) to the glm() function. The “weights” tells R where those frequencies are.

glm(*response* ~ *explanatory*, weights=*frequency*, data=*data*, family=binomial)

So in our case, the code is:

cancer.log <- glm(Var1~Var2, weights=value, data=melt(cancer.tab), family = binomial)

1. **Interpretation**
2. Why are there so many different slope values? And why is “breast” missing?
3. For Colon, e^(slope) = 0.25. Interpret this quantity in context.
4. Calculate the odds ratio of survival for colon cancer vs breast cancer, using the table in #1. What do you notice about this quantity compared to #5b?