**Logistic Regression Practice: fitting the model**

**[Your name(s) here]**

**Part A: Untabulated data**

Open the **MedGPA** data set (from the Stat2Data) in R. We’ll be continuing our work with MCAT as a predictor of Acceptance.

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

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. **Checking the linearity condition**

Recall that in “Odds & ORs, Part 2” you created a table that grouped the MCAT scores into ranges and calculated the odds for each. Based on your plot of log(odds) vs. MCAT score from that activity, what do you conclude about the linearity condition? Does it look like the logistic model’s linearity constraint is reasonable?

1. **Visualizing the Logistic Model: 2 ways**
   1. On the “linearized” plot of log(odds) vs. MCAT (from “Odds & ORs, Part 2”), you can plot the (straight-line) slope, as in Figures 9.12 and 9.13. There are multiple ways to do this, but here’s one:

plot(log(odds)~midpoints, xlab="MCAT") #odds and midpoints are from Odds & ORs, part 2

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

Using this plot, does it look like this model is a good fit to the data?

* 1. Another way to visualize the model is the plot the “S-curve” version of the logistic model, as in Figures 9.3 and 9.14, as opposed to the linear version above. To do this, plot proportion of success vs. MCAT score (you did this in Odds & ORs, Part 2), then use the function makeFun():

xyplot(props~midpoints, xlab="MCAT") #props and midpoints are from Odds & ORs, part 2

fit <- makeFun(mcat.log) #makeFun is in the mosaic package

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

Using this plot, does it look like this model is a good fit to the data?

**Remember:** Both of the plots above use the groups that we created in Odds & ORs, Part 2. But the process of grouping is for visualization purposes *only*: it has nothing to do with actually fitting the model. Ironically, however, by making groups we can (unintentionally) visually alter the distribution of the data.

If you want to see a more accurate picture of the **linearized** plot in #3a, try the code below:

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

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

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!)

If you want to see a more accurate picture of the **S-curve** plot in #3b, try the code below:

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

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

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

fit <- makeFun(mcat.log)

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

**Part B: Tabulated data**

In the MCAT file, the data was in “long format”: 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*** (“wide format”), and we don’t have the long format?

Here is the table of cancer survival from Odds & ORs, Part 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 | 9/2 | 3/14 | 9/8 | 3/3 | 9/4 |
| Log(odds) of survival | 1.50 | -1.54 | 0.12 | 0 | 0.81 |

1. **Create the (wide format) 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)

(You can also do all this in the tidyverse, using pivot\_longer.)

1. **Interpretation**
2. Why are there so many different estimated coefficients? And why is “Breast” missing?
3. The intercept of the model is 1.50. Interpret this quantity in context.
4. For Colon, e^(slope) = 0.25. Interpret this quantity in context.
5. Calculate the odds ratio of survival for colon cancer vs breast cancer, using the table at the top of this section. What do you notice about this quantity compared to #3c?
6. **Checking the linearity condition**

Linearity is automatic here. What we have is 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” from the reference category (“Breast”, in this case) to each other cancer type, and two points always make a line.