\*--------------------------------------------------------------------------\*

|TITLE: In Class Exercise 1, Creating Permanent Data with libname and basic|

| Logistic Regression |

| |

|DATE: January 15, 2020 in class exercise (WITH ANSWERS) |

| |

|DATASETS: SAS dataset named 'TBantigen' availalbe on Canvas |

| |

|USER: Matthew Magee |

\*--------------------------------------------------------------------------\*

;

\*Lab 1 contains 4 parts:

-Part 1 Libname statements

-Part 2 Odds ratios with proc freq

-Part 3 Basics of proc logistic

-Part 4 Dummy variables;

\*\*\*\*\*\*\*\*PART 1, PRACTICE WITH LIBNAME STATEMENTS

\*Download the SAS dataset TBantigen from Canvas and save it to your desktop.

\*Creating a SAS library in the same location as the SAS dataset will allow you

to use it within SAS datasetps and procedures.

Therefore, create a SAS library on your desktop named 'Popstar';

libname popstar "C: ????????????????";

\*In oder for SAS datasets to be used, they have to be in referred to in a

library or a temporary folder 'work';

\*Perform a proc contents procedure to make sure the data was read correctly

in the libname statement;

**proc** **contents** data=???????????????;

**run**;

\*

\*How many variables are included in TBantigen?

\*How many observations are included in TBantigen?

\*Next, use a data statement to create a temporary SAS dataset from

the permanent dataset TBantigen.

\*Call the temporary dataset 'one';

**data** one;

set ?????????;

**run**;

\*check the log to ensure the new temporary dataset was created.

What is the full name of the new temporary dataset?

\*Next use another data step to create a new variable 'ant\_high' within a new temporary dataset called 'two'.

The new variable 'ant\_high' should dichotomize the continuous variable LBXTBA as high response vs other.

Code a high response to be TB antigen response >= 10 IU/mL;

\*First double check the distribution of the continuous variable lbxtba;

**proc** ????

**data** two;

set one;

if ??????????;

else if ???????? then ????;

**run**;

\*check that the coding worked correctly;

**proc** **print** data=two (obs=**500**); \*proc print is useful to check that coding worked, 'obs=' statement limits the number printed;

var lbxtba ant\_high;

**run**;

\*Make work.two permanent by placing it in your SAS library and call it 'lab1'.

In the same step, limit observations in the permanent dataset where participants were classified as QFT positive.

In other words, remove those participants who were coded as LTBI negative;

\*First check coding of the LTBI variable;

**proc** **freq** data=two;

table lbxtbin;

**run**;

\*If LBXTBIN=2 then participants were classified as LTBI negative;

**data** popstar.lab1;

?????

?????

run;

\*Check log to make sure datastep worked.

How many observations does popstar.lab1 contain?

How many variables does popstar.lab1 contain?

\*\*\*\*Last step for Part 1.

-Save the SAS program.

-Close SAS, reopen it, and open the SAS program.

-Re-run the libname statement in the first section of Part 1

-Perform a proc freq on any varible in the 'lab1' permanent datasets;

\*\*\*\*\*\*\*\*PART 2, ODDS RATIOS WITH PROC FREQ;

\*Using proc freq, create a frequency table for the relationship

between diabetes status and high TB antigen response.

\*Perform this comparison only among those who were LTBI positive;

**proc** **freq** data=popstar.lab1;

table dm\*ant\_high/;

**run**;

\*Calculate (by hand) the odds ratio of high antigen comparing those

with diabetes to euglycemic patients.

\*Odds of high antigen with diabetes:

\*Odds of high antigen with euglycemia:

\*Odds ratio:

\*Proc freq can also create odds ratios if the table is truly 2x2;

\*Rerun the proc freq excluding those with pre-diabetes

using the 'where' statement;

\*Using the '/CMH' at the end of the table statement will provide the odds ratio;

**proc** **freq** data=popstar.lab1;

????

????

run;

\*confirm that the 'cmh' option produced the same OR as when calculated by hand;

\*\*\*Next, what would happen to the OR if the coding of high antigen was switched?;

\*Recode 'anti\_high' into a new variable 'anti\_high2'.

\*With ant\_high2, code those with high antigen as 0 and those without as 1;

\*Rerun the proc freq table to obtain an odds ratio with the alternate coding;

**data** three;

set popstar.lab1;

if ant\_high=**1** then ??????;

else if ant\_high=**0** then ????;

**run**;

**proc** **freq** data=three;

table dm\*ant\_high2/cmh;

where dm ^=**1**;

**run**;

\*What is the odds ratio reported from the SAS output?

\*Interpretation of the OR depends on variable coding;

\*As a statistical software user, it is important to

check results by hand;

\*\*\*\*\*\*\*\*PART 3, BASIC LOGISTIC REGRESSION;

\*\*\*\*Single covariate logistic regression;

\*Using Proc logistic, recalculate the odds ratios

(both 1.53 and 0.65) from Part 2 above.

\*With one covariate, the odds ratio should be the same as from a proc freq;

**proc** **logistic** data=popstar.lab1;

model ant\_high = dm;

**run**;

\*The OR reported from the above model is 0.80.

\*What is the interpretation of this OR?

\*Rerun the proc logistic procedure modeling the odds of high antigen response and use a 'class' statement

so SAS recognizes that DM is a categorical variable;

**proc** **logistic** data=popstar.lab1;

class dm (param=ref ref='0'); \*param=ref ref=0 indicates which level is modeled as the referent group;

model ant\_high (event='1') = dm; \*event='1' indicates to model log odds of high antigen response;

**run**;

\*What is the OR from the above model using the class statement?

-The OR from proc logistic is the same as from the proc freq: 1.53;

\*Is the predm vs. euglycemic OR the same as derived from the proc freq?

\*Last, recreate the OR 0.65 observed from the second proc freq above (last part of Part 2;

**proc** **logistic** data=popstar.lab1;

class dm (param=ref ref='0');

model ant\_high (event='?') = dm;

**run**;

\*\*\*\*Multiple covariate logistic regression;

\*The above logistic models are considered crude, unadjusted, or single predictor models;

\*Adding one additonal variable makes it multivariable;

\*Using a data step, create a categorical variable from RIDAGEYR (continuous age), call the new

variable 'agecat' and use THREE-levels;

\*Then run a logistic model with the outcome high antigen response with both DM and agecat as categorical

covariates;

\*First determine potential cutpoints for age;

**proc** **univariate** data=popstar.lab1;

var RIDAGEYR;

**run**;

**data** four;

set popstar.lab1;

if RIDAGEYR ??? then agecat=?;

else if RIDAGEYR ???? and agecat ???? then agecat=?;

else if RIDAGEYR ??? then agecat=?;

**run**;

\*check that coding worked;

**proc** **print** data=four (obs=**500**);

var RIDAGEYR agecat;

**run**;

\*Best to check the cutpoints;

\*Where is the mistake above?;

\*Now replace lab1 with a new lab1 that includes agecat (save the temporary dataset over lab1);

**data** ????

????

run;

\*Now run a logistic model with the outcome high antigen response with both DM and agecat as categorical

covariates;

**proc** **logistic** data=popstar.lab1;

class dm (param=ref ref='0') agecat (param=ref ref='0');

model ant\_high (event='1') = dm agecat;

**run**;

\*What is the interpretation of the odds ratio after including agecat?

\*\*\*\*\*\*\*\*PART 4, DUMMY VARIABLES;

\*For categorical variables with more than 2 levels, the class statment is a convenient way to

enable proc logistic to model categorical variables.

\*However, it is useful to understand that the class statement is essentially creating dummy variables

and therefore contributes to other model considerations (fit, saturation, etc).

\*\*\*\*Re-run the adjusted logistic model from part 3 using dummy variables.

\*First, create dummy variables for agecat and dm;

**data** six;

set popstar.lab1;

\*dummy variables for DM:;

if dm=**1** then dm\_1=**1**;

else dm\_1=**0**;

if dm=**2** then dm\_2=**1**;

else dm\_2=**0**;

\*dummy variables for agecat;

????

???

???

run;

\*Check that recoding into dummy variables worked;

**Proc** **freq** ???;

???

???;

\*According to proc freq, dummy varialbes look correct.

\*Copy these new dummy variables into the permanent sas dataset 'lab1';

**data** ???;

????;

**run**;

\*Last, re-run the logistic model without the class statement;

**Proc** **logistic** data=popstar.lab1;

model ant\_high (event='1')= ???????;

**run**;

\*The parameter estimates and ORs should be the same as when using the class statement as in Part 3;

\*In the above example withe the class statement, the model statement only included two named

covariates but there were actualy 4 parameters estimated.

\*Keep in mind that when using the class statement the model is specified differently than appears

in the proc logistic code.