**Assignment 3: SAS Exercise – Understanding the data.**

Only post the answers to**Question 5.**

The first four questions for your practice only**.**

Remember that for question 5, you need to do crosstabs for 5 different variables and save as an Excel file and post.

1. Read the contents of the Project files Data1.sd2 and Xtra1.sd2. Save output to a file.

**Instructions**:

a.       Open SAS and run the **start.sas** program.

b.      In the Program Editor window, type the following program (change the *path* in the first line to point to your output folder)

ODS html file= *“C:\project\output\contents.html”;*

PROC CONTENTS data=save.data1;

Run;

ODS html close;

c.       Submit the program. The output is an html file. Open it in Excel, **save as an xlsx file**.

1. Sort on account number and merge the files based on that variable. Print 10 records of the merged file, and save as .xlsx after reading the output html file into Excel.

**Instructions**:

a.       Type the following program:

PROC SORT data=save.data1; by acctno; run;

PROC SORT data=save.xtra1; by acctno; run;

DATA save.data2;

MERGE save.data1 save.xtra1;

By acctno;

Run;

Quit;

b.      Check whether it worked by looking at the log, or by typing in a proc contents statement with the file save.data2.

c.       To print 10 records, type the following:

ODS html file= “*C:\project\output\tenrecs.html*”;  \*\*\*tenrecs.html is a filename I made up for the output \*\*\*;

PROC PRINT data=save.data2 (obs=10); run;

ODS html close;

1. Split the merged file into *training* and *validation* datasets, with about 50% of the observations in each.

**Instructions**:

a.       Type the following program:

DATA save.train save.valid;

Set save.data2;

Random1 = ranuni(14380132);       \*\*\*use your own 7-8 digit seed number\*\*\*\*;

IF Random1 < 0.5 THEN output save.train;

      ELSE output save.valid;

Run;

b.      Do a PROC CONTENTS statement on each new file to show that the program worked correctly.

Use the **Training** **dataset** to do the following:

1. Create frequency tables for any 5 variables (formatted such that each class has between 2% and 10% of the observations, except in the case of counting variables (such as **Number of Trades**, where certain values may have a higher percentage.) Make sure the formats separate out the negative values with appropriate labels.

**Instructions**:

a.       Pick a few variables. Say one of the variables is **vage** (customer age). Create a frequency table without any bins first.

PROC FREQ data=save.train;

TABLES vage;

Run;

b.      Look at the cumulative frequencies in the output to determine the age groups (bins) that would contain about 2 to10% (say 5%) of the observations each.

c.       Use those cutoffs to create a new format (a set of bins to help display the frequency tables) as follows:

Open the program file called **format.sas**. Create a new format (format name should be less than 8 characters, and must end in a character – not a number). The program will look something like this.

PROC FORMAT library=library;

VALUE vage

LOW – 17 = ‘Missing’

18-19 = ’18 to 19’

20-21=’20 to 21’

22-23=’22 to 23’

….(fill in the other ranges here as appropriate) …

68-HIGH = ‘Over 68’;

Run;

d.      Run the format program above. This will overwrite the FORMAT file in the Formats folder, which is essentially a repository of formats that SAS can use when called upon.

e.       Create a frequency table for vage now with the format applied to it, as follows - In the program editor window, type:

PROC FREQ data=save.train;

TABLES vage;

FORMAT vage vage.;   \*\*\* The first **vage** is the variable name. The second **vage** with the **period** after it refers to the format name to be applied \*\*\*\*

Run;

1. Create Crosstabs of the same variables as in question 4, with the binary dependent variable GOOD [1=good customer, 0=bad (delinquent, charged off) customer].**Open the output html files in Excel, and save as .xlsx file and post online.**

**Instructions**:

a.       Crosstabs are two-way frequency tables. The syntax is as follows:

ODS html file= “*C:\project\output\your\_filename\_here.html*”;

PROC FREQ data=save.train;

TABLES vage\*good;

FORMAT vage vage.;

Run;

ODS html close;

b.      To get only the column percentages for each cell, you can ask SAS to ignore the overall percent and the row percent as follows:

ODS html file= “*C:\project\output\your\_filename\_here.html*”;

PROC FREQ data=save.train;

TABLES vage\*good / norow nopercent;

FORMAT vage vage.;

Run;

ODS html close;