Wine Assignment #3

Derek Hughes
PRED 411 – Sec 55
Winter 2015

INTRODUCTION

The wine industry has become a thriving industry as of the past couple decades. Not surprisingly, where there is strong growth and an opportunity to make money, competition inevitably increases. The wine makers who survive in this environment are those who learn to find and take advantage of any opportunity to get an edge on the competition. One way to do that is to take advantage of modern predictive modeling techniques.

In this report, we will discuss how a wine maker can get an advantage over the competition by learning how to predict the probability that a wine maker's wine will be selected by professional wine tasters. By correctly predicting this probability, a wine maker can properly adjust inventory levels, pricing levels, and other business practices to maximize profits from each wine they produce. Furthermore, the wine maker can learn which variables have the most influence on the probability a wine taster will select their wine.

The following paper describes the processes used to create and compare five different models that predict the probability that a given wine will, not only be selected for an initial distribution to various restaurants and stores, but how many cases of that wine will be purchased as well. Ultimately, one model will be selected for the manufacturer to use for prediction.

DATA EXPLORATION

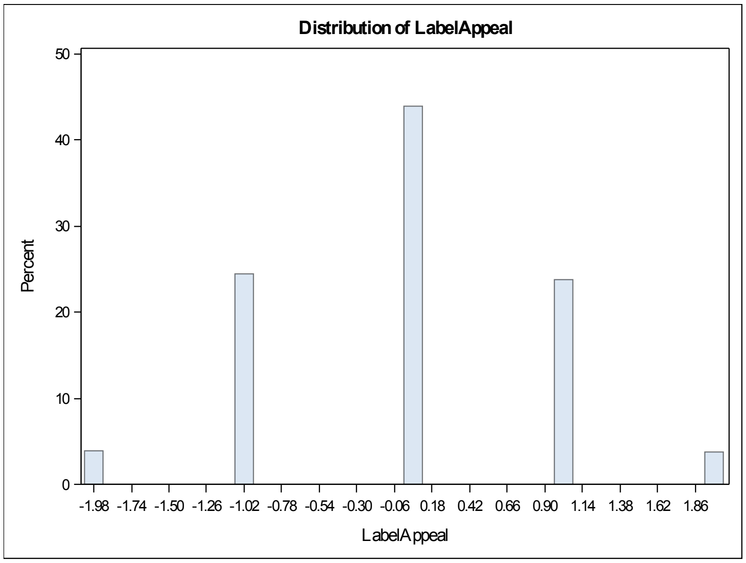
Step 1:

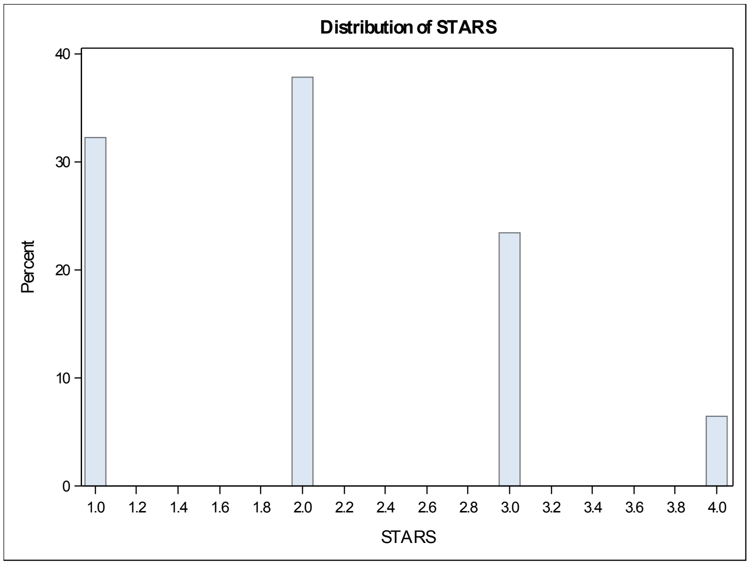
Step one in model creation is usually to evaluate the data we are utilizing. Here we have approximately 12,800 records of commercial wines. Each wine is described in 14 variables that are mainly related to the chemical properties in the wine. Furthermore, all variables are numeric, but not all are continuous.

Alphabetic List of Variables and Attributes			
#	**Variable**	**Type**	**Len**
15	AcidIndex	Num	8
13	Alcohol	Num	8
7	Chlorides	Num	8
5	CitricAcid	Num	8

Alphabetic List of Variables and Attributes			
#	**Variable**	**Type**	**Len**
10	Density	Num	8
3	FixedAcidity	Num	8
8	FreeSulfurDioxide	Num	8
14	LabelAppeal	Num	8
6	ResidualSugar	Num	8
16	STARS	Num	8
12	Sulphates	Num	8
9	TotalSulfurDioxide	Num	8
4	VolatileAcidity	Num	8
11	pH	Num	8

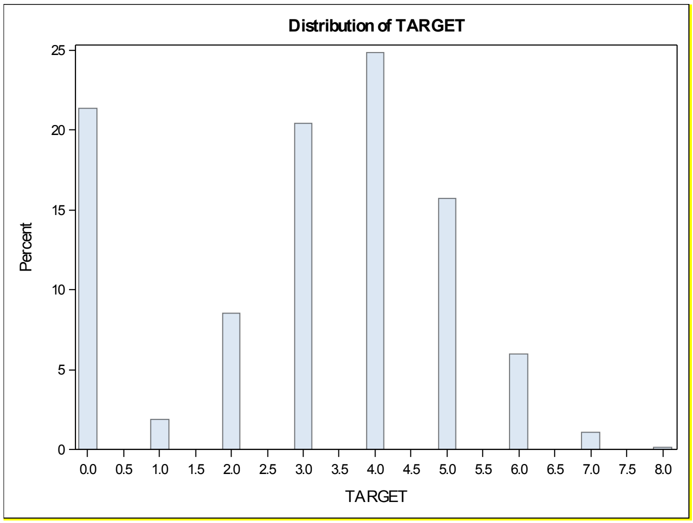
By exploring each variable's histogram, we can see that the variable STARS and LabelAppeal are discrete numeric values used to represent levels of performance.





There is also one “target” variable - the number of sample cases of wine that were purchased by the wine distribution company after sampling the wine. This target variable (the count of the number of sample cases purchased) is what we will try to predict with our models. When trying to predict the count of something, there are two models that generally work well with those situations: the Poisson and Negative Binomial models.

One unique characteristic about our target variable is that there are a large number of zero values. Here it represents that zero cases of sample wine were purchased by the distribution company. In these cases (pun intended), we can try another approach called the “zero inflation” model to predict the number of cases purchased. It is a bit more complicated because it requires a logistic model to predict if the wine will be purchased or not, combined with a Poisson or Negative Binomial model used to predict the amount of sample wine cases purchased.



We can also see there are eight variables that are missing values. These observations will need to be replaced with substitute value for the missing variable or removed from the data set when we develop our model.

Variable	N Miss	Mean	Median	Minimum	Maximum
INDEX	0	8069.98	8110.00	1.0000000	16129.00
TARGET	0	3.0290739	3.0000000	0	8.0000000
FixedAcidity	0	7.0757171	6.9000000	-18.1000000	34.4000000
VolatileAcidity	0	0.3241039	0.2800000	-2.7900000	3.6800000
CitricAcid	0	0.3084127	0.3100000	-3.2400000	3.8600000
ResidualSugar	616	5.4187331	3.9000000	-127.8000000	141.1500000
Chlorides	638	0.0548225	0.0460000	-1.1710000	1.3510000
FreeSulfurDioxide	647	30.8455713	30.0000000	-555.0000000	623.0000000
TotalSulfurDioxide	682	120.7142326	123.0000000	-823.0000000	1057.00
Density	0	0.9942027	0.9944900	0.8880900	1.0992400
pH	395	3.2076282	3.2000000	0.4800000	6.1300000
Sulphates	1210	0.5271118	0.5000000	-3.1300000	4.2400000
Alcohol	653	10.4892363	10.4000000	-4.7000000	26.5000000
LabelAppeal	0	-0.0090660	0	-2.0000000	2.0000000
AcidIndex	0	7.7727237	8.0000000	4.0000000	17.0000000
STARS	3359	2.0417550	2.0000000	1.0000000	4.0000000
TARGET_FLAG	0	0.7863228	1.0000000	0	1.0000000

We can see that there is a large difference between the minimum and maximum values and the mean value for a few of the variables. This indicates that we may have some significant outliers for some variables that are selected into our model.

Additionally, we can see there are many negative numbers for many of the chemical properties. While this seems counter intuitive to have a negative number of a particular property, because negative values are so frequent throughout the data set, we are assuming it is an intentional value possibly due to a standardizing approach for that variable or possibly based on some reference value. The assumption is that future data sets (test data set) will have the same parameter values for its variables.

TARGET _FLAG	N Obs	Variable	N Miss	Mean	Median	Minimum	Maximum
0	2734	INDEX	0	8086.79	8105.50	7.0000000	16120.00
		TARGET	0	0	0	0	0
		FixedAcidity	0	7.7337601	7.4000000	-18.0000000	34.4000000
		VolatileAcidity	0	0.4460333	0.3800000	-2.6400000	3.5650000
		CitricAcid	0	0.2991953	0.3000000	-3.2400000	3.7700000
		ResidualSugar	127	3.9978711	2.4000000	-126.1000000	137.6000000
		Chlorides	141	0.0759838	0.0620000	-1.1710000	1.3510000
		FreeSulfurDioxide	139	17.9433526	20.0000000	-535.0000000	622.0000000
		TotalSulfurDioxide	140	85.2615652	84.5000000	-823.0000000	981.0000000
		Density	0	0.9952851	0.9958000	0.8934300	1.0992400
		pH	101	3.2464033	3.2400000	0.5400000	6.0500000
		Sulphates	279	0.6115927	0.5600000	-3.0100000	4.2400000
		Alcohol	137	10.4286009	10.3000000	-4.4000000	25.2000000
		LabelAppeal	0	0.00036576	0	-2.0000000	2.0000000
		AcidIndex	0	4	8.0000000	4.0000000	17.0000000
		STARS	2038	8.4528164	1.0000000	1.0000000	2.0000000
				1.1278736			
1	10061	INDEX	0	8065.41	8112.00	1.0000000	16129.00
		TARGET	0	3.8522016	4.0000000	1.0000000	8.0000000
		FixedAcidity	0	6.8968989	6.8000000	-18.1000000	32.5000000
		VolatileAcidity	0	0.2909706	0.2700000	-2.7900000	3.6800000
		CitricAcid	0	0.3109174	0.3100000	-3.1600000	3.8600000
		ResidualSugar	489	5.8057146	4.8000000	-127.8000000	141.1500000
		Chlorides	497	0.0490852	0.0440000	-1.1700000	1.2700000
		FreeSulfurDioxide	508	34.3503611	33.0000000	-555.0000000	623.0000000
		TotalSulfurDioxide	542	130.375354	130.000000	-793.0000000	1057.00
		Density	0	6	0	0.8880900	1.0992400
		pH	294	0.9939086	0.9940000	0.4800000	6.1300000
		Sulphates	931	3.1971752	3.1900000	-3.1300000	4.2100000
		Alcohol	516	0.5043954	0.4900000	-4.7000000	26.5000000
		LabelAppeal	0	10.5057339	10.4000000	-2.0000000	2.0000000
		AcidIndex	0	-0.0116291	0	4.0000000	17.0000000
		STARS	1321	7.5879137	7.0000000	1.0000000	4.0000000
				2.1145309	2.0000000		

Since we will be exploring a logistic regression model, we can go ahead and create a new variable called TARGET_FLAG that has a value of “1” if sample wine cases were bought and “0” if they were not. With this variable, we can now explore the other variables even more extensively relative to whether or not cases were purchased. One major area of intrigue is the number of missing values for STARS compared to when no cases were purchased. This could indicate missing STARS values are predictive.

VARIABLE SELECTION:

To select variables to use for our model, we will compare variables selected via Linear and Logistic regression stepwise selection processes against an unaltered data set and one with the missing values imputed.

First we will impute the missing values with their average (mean) value. We will also create a new missing value flag variable to identify which variables had missing values and/or if the missing values have any predictive power. Below, we can see that we have properly imputed all the missing values and that there are new missing value flag variables.

Variable	N Miss	Mean	Median	Minimum	Maximum
TARGET	0	3.0290739	3.0000000	0	8.0000000
TARGET_FLAG	0	0.7863228	1.0000000	0	1.0000000
IMP_AcidIndex	0	7.7727237	8.0000000	4.0000000	17.0000000
IMP_Alcohol	0	10.4892363	10.4892363	-4.7000000	26.5000000
M_Alcohol	0	0.0510356	0	0	1.0000000
IMP_Chlorides	0	0.0548225	0.0480000	-1.1710000	1.3510000
M_Chlorides	0	0.0498632	0	0	1.0000000
IMP_CitricAcid	0	0.3084127	0.3100000	-3.2400000	3.8600000
IMP_Density	0	0.9942027	0.9944900	0.8880900	1.0992400
IMP_FixedAcidity	0	7.0757171	6.9000000	-18.1000000	34.4000000
IMP_FreeSulfurDioxide	0	30.8455713	30.8455713	-555.0000000	623.0000000
M_FreeSulfurDioxide	0	0.0505666	0	0	1.0000000
IMP_LabelAppeal	0	-0.0090660	0	-2.0000000	2.0000000
IMP_pH	0	3.2076282	3.2076282	0.4800000	6.1300000
M_pH	0	0.0308714	0	0	1.0000000
IMP_STARS	0	2.0307933	2.0000000	1.0000000	4.0000000
M_STARS	0	0.2625244	0	0	1.0000000
IMP_ResidualSugar	0	5.4187331	4.9000000	-127.8000000	141.1500000
M_ResidualSugar	0	0.0481438	0	0	1.0000000
IMP_Sulphates	0	0.5271118	0.5271118	-3.1300000	4.2400000
M_Sulphates	0	0.0945682	0	0	1.0000000
IMP_TotalSulfurDioxide	0	120.7142326	120.7142326	-823.0000000	1057.00
M_TotalSulfurDioxide	0	0.0533021	0	0	1.0000000
IMP_VolatileAcidity	0	0.3241039	0.2800000	-2.7900000	3.6800000

Now we will compare the variables selected from using a Linear regression and a Logistic regression. We will use the same variable selection process (stepwise) and select the variables that are common to both linear and logistic models to use for all the models we will create.

Linear Regression Stepwise selection variables:

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	4.38465	0.44442	9.87	<.0001	0
IMP_AcidIndex	1	-0.19991	0.00896	-22.30	<.0001	1.05041
IMP_Alcohol	1	0.01240	0.00320	3.88	0.0001	1.00626
IMP_Chlorides	1	-0.11737	0.03736	-3.14	0.0017	1.00290
IMP_Density	1	-0.80653	0.43704	-1.85	0.0650	1.00308
IMP_FreeSulfurDioxide	1	0.00028507	0.00008005	3.56	0.0004	1.00349
IMP_LabelAppeal	1	0.46643	0.01367	34.13	<.0001	1.10616
IMP_pH	1	-0.03153	0.01735	-1.82	0.0692	1.00479
IMP_STARS	1	0.77939	0.01567	49.73	<.0001	1.10097
M_STARS	1	-2.24420	0.02695	-83.27	<.0001	1.04869
IMP_Sulphates	1	-0.03112	0.01307	-2.38	0.0173	1.00204
IMP_TotalSulfurDioxide	1	0.00022446	0.00005143	4.36	<.0001	1.00415
IMP_VolatileAcidity	1	-0.09649	0.01481	-6.51	<.0001	1.00604

Logistic Regression Stepwise selection variables:

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	2.6560	0.2802	89.8225	<.0001
IMP_AcidIndex	1	-0.3890	0.0214	331.3013	<.0001
IMP_Alcohol	1	-0.0209	0.00791	7.0070	0.0081
IMP_FreeSulfurDioxid	1	0.000622	0.000200	9.6956	0.0018
IMP_LabelAppeal	1	-0.4697	0.0333	198.8353	<.0001
IMP_pH	1	-0.1831	0.0426	18.4626	<.0001
IMP_STARS	1	2.5589	0.1119	522.6795	<.0001
M_STARS	1	-4.3762	0.1115	1540.7122	<.0001
IMP_Sulphates	1	-0.1080	0.0323	11.2001	0.0008

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
IMP_TotalSulfurDioxi	1	0.000854	0.000127	45.3915	<.0001
IMP_VolatileAcidity	1	-0.1817	0.0365	24.8255	<.0001

The common variables between the two values are highlighted and will be used as the base variables we will use to develop our models.

Now, we will compare the means of a variable when wine is bought to when wine is not purchased. Large differences in these means will be used as indicator of the possibility of another predictive variable. For example, we can see a large difference in the mean value for M_STARS when wine is bought (TARGET_FLAG = 1) compared to when it was not bought.

Also, I am manually inserting IMP_CitricAcid due to domain knowledge that citric acid amounts have a significant effect on wine taste.

TARGET _FLAG	N Obs	Variable	N Miss	Mean	Median	Minimum	Maximum
0	2734	TARGET	0	0	0	0	0
		IMP_AcidIndex	0	8.4528164	8.0000000	4.0000000	17.0000000
		IMP_Alcohol	0	10.4316394	10.4892363	-4.4000000	25.2000000
		M_Alcohol	0	0.0501097	0	0	1.0000000
		IMP_Chlorides	0	0.0748925	0.0570000	-1.1710000	1.3510000
		M_Chlorides	0	0.0515728	0	0	1.0000000
		IMP_CitricAcid	0	0.2991953	0.3000000	-3.2400000	3.7700000
		IMP_Density	0	0.9952851	0.9958000	0.8934300	1.0992400
		IMP_FixedAcidity	0	7.7337601	7.4000000	-18.0000000	34.4000000
		IMP_FreeSulfurDioxide	0	18.5993176	23.0000000	-535.0000000	622.0000000
		M_FreeSulfurDioxide	0	0.0508413	0	0	1.0000000
		IMP_LabelAppeal	0	0.000365764	0	-2.0000000	2.0000000
		IMP_pH	0	3.2449709	3.2200000	0.5400000	6.0500000
		M_pH	0	0.0369422	0	0	1.0000000
		IMP_STARS	0	1.7779810	2.0000000	1.0000000	2.0000000
		M_STARS	0	0.7454279	1.0000000	0	1.0000000
		IMP_ResidualSugar	0	4.0638731	2.6000000	-126.1000000	137.6000000
		M_ResidualSugar	0	0.0464521	0	0	1.0000000
		IMP_Sulphates	0	0.6029715	0.5300000	-3.0100000	4.2400000
		M_Sulphates	0	0.1020483	0	0	1.0000000
		IMP_TotalSulfurDioxide	0	87.0769907	95.0000000	-823.0000000	981.0000000
		M_TotalSulfurDioxide	0	0.0512070	0	0	1.0000000
		IMP_VolatileAcidity	0	0.4460333	0.3800000	-2.6400000	3.5650000

TARGET _FLAG	N Obs	Variable	N Miss	Mean	Median	Minimum	Maximum
1	10061	TARGET	0	3.8522016	4.0000000	1.0000000	8.0000000
		IMP_AcidIndex	0	7.5879137	7.0000000	4.0000000	17.0000000
		IMP_Alcohol	0	10.5048878	10.4892363	-4.7000000	26.5000000
		M_Alcohol	0	0.0512871	0	0	1.0000000
		IMP_Chlorides	0	0.0493686	0.0460000	-1.1700000	1.2700000
		M_Chlorides	0	0.0493987	0	0	1.0000000
		IMP_CitricAcid	0	0.3109174	0.3100000	-3.1600000	3.8600000
		IMP_Density	0	0.9939086	0.9940000	0.8880900	1.0992400
		IMP_FixedAcidity	0	6.8968989	6.8000000	-18.1000000	32.5000000
		IMP_FreeSulfurDioxide	0	34.1733973	30.8455713	-555.0000000	623.0000000
		M_FreeSulfurDioxide	0	0.0504920	0	0	1.0000000
		IMP_LabelAppeal	0	-0.0116291	0	-2.0000000	2.0000000
		IMP_pH	0	3.1974806	3.2000000	0.4800000	6.1300000
		M_pH	0	0.0292217	0	0	1.0000000
		IMP_STARS	0	2.0994931	2.0000000	1.0000000	4.0000000
		M_STARS	0	0.1312991	0	0	1.0000000
		IMP_ResidualSugar	0	5.7869059	5.4187331	-127.8000000	141.1500000
		M_ResidualSugar	0	0.0486035	0	0	1.0000000
		IMP_Sulphates	0	0.5064975	0.5200000	-3.1300000	4.2100000
		M_Sulphates	0	0.0925355	0	0	1.0000000
		IMP_TotalSulfurDioxide	0	129.8548965	123.0000000	-793.0000000	1057.00
		M_TotalSulfurDioxide	0	0.0538714	0	0	1.0000000
		IMP_VolatileAcidity	0	0.2909706	0.2700000	-2.7900000	3.6800000

Our final variable group for developing the model is as follows:

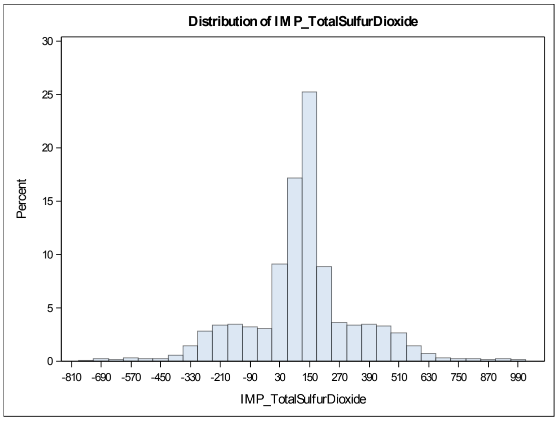
IMP_AcidIndex
IMP_Chlorides
IMP_CitricAcid
IMP_FreeSulfurDioxide
IMP_LabelAppeal
IMP_pH
IMP_STARS
M_STARS
IMP_Sulphates
IMP_TotalSulfurDioxide
IMP_VolatileAcidity

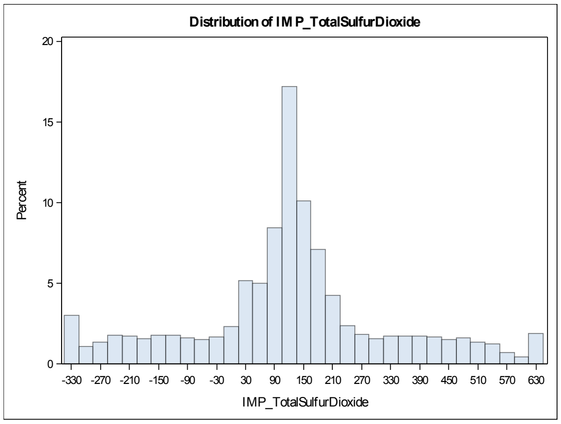
TRANSFORMING, TRIMMING, IMPUTING VARIABLES:

Here we will transform the variables to create a more representative variation of the variable that will be more useful for developing a model. We will focus on removing outlier observations in each variable, transforming the variables closer to a normal distribution, and the possibility of creating flag variables to improve model performance.

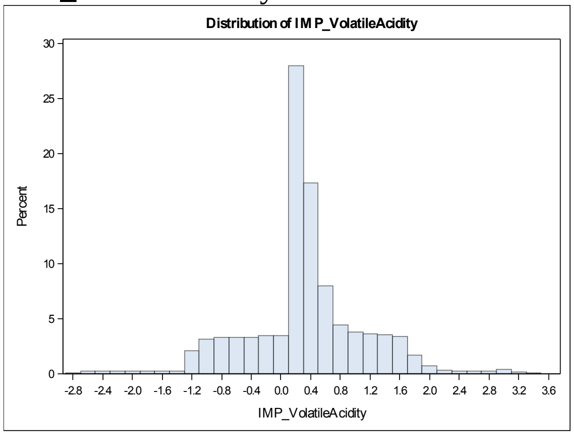
For each of our variables, we can see outliers and therefore will trim the ends of each distribution to create a more representative sample for developing our model. The before and after of each trimmed variable is shown below.

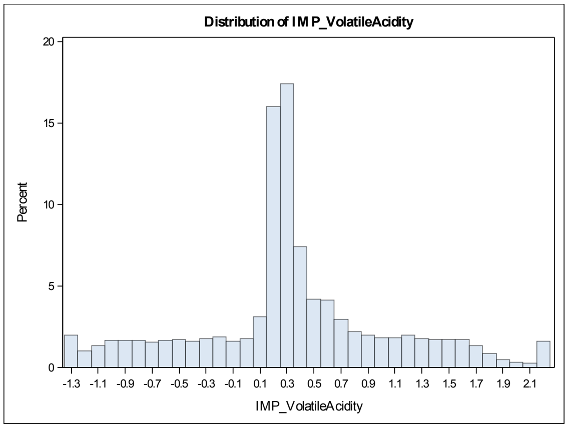
IMP_TotalSulfurDioxide



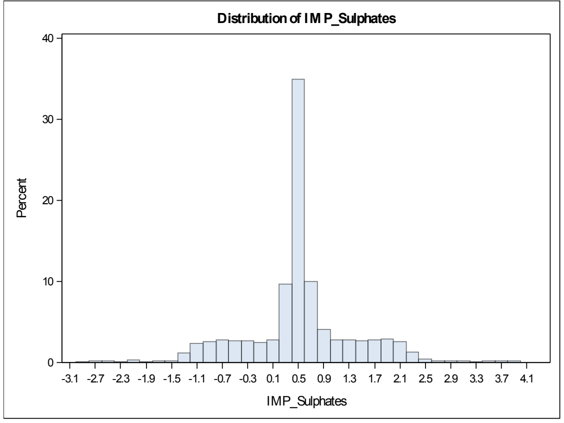


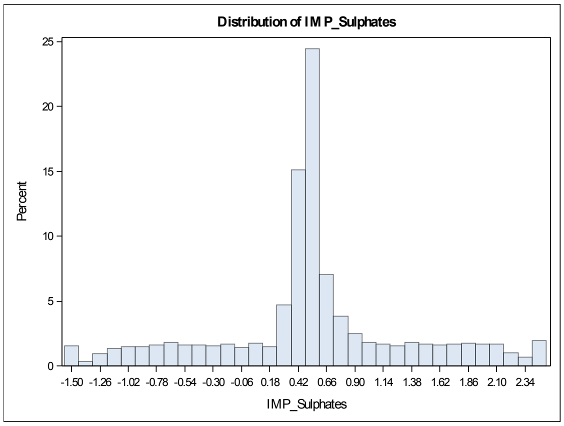
IMP_VolatileAcidity



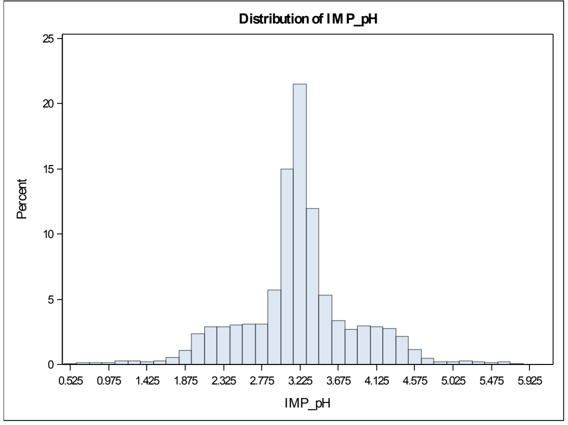


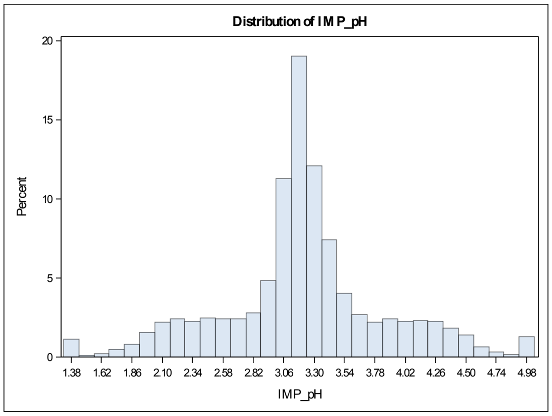
IMP_Sulphates



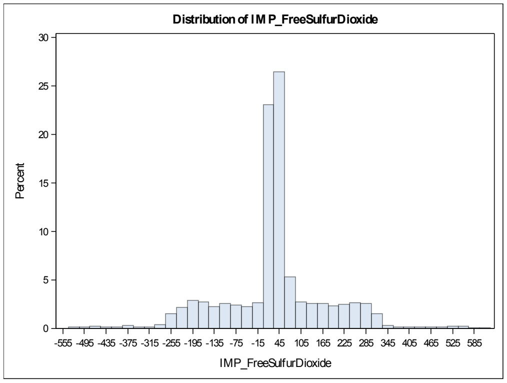


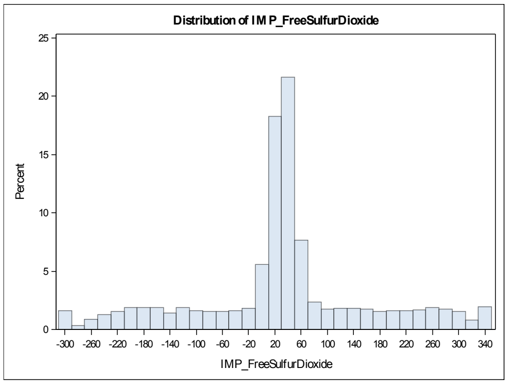
IMP_pH



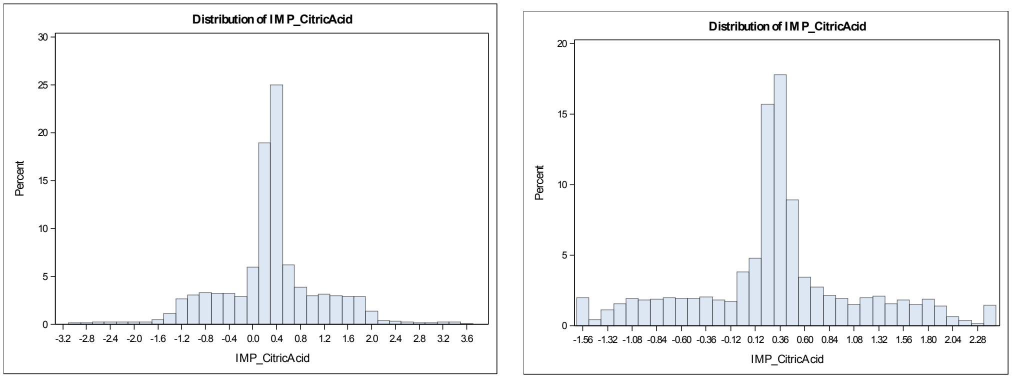


IMP_FreeSulferDioxide

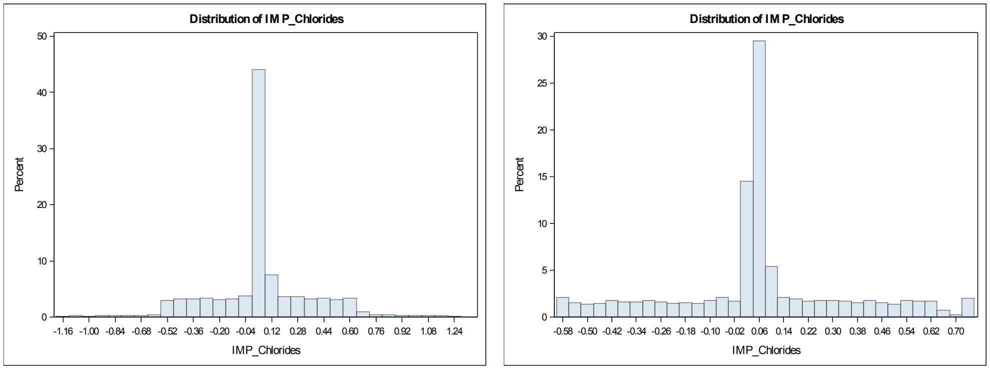




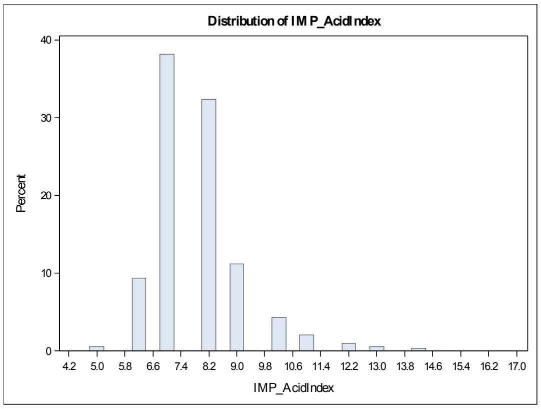
IMP_CitricAcid

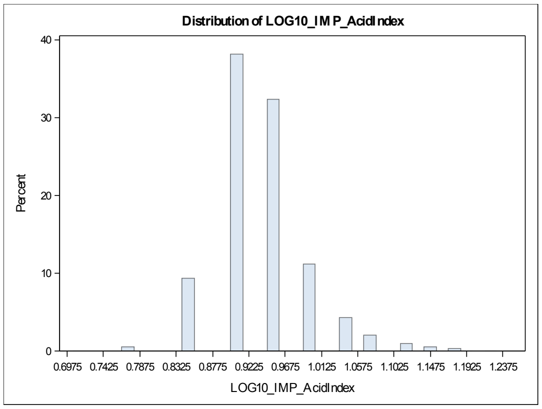


IMP_Chlorides



Next, we saw a highly skewed distribution of the IMP_AcidIndex. We tried Log10 and Sqrt transformations and settled on the Log10 transformation because it showed a distribution closer to a normal distribution.





Finally, here we see that the number of STARS has a significant effect on whether or not sample wine cases were purchased. Three and four STARS wine were all purchased while only 20% of wine sample cases were purchased with on one STARS. Interestingly, having two STARS compared to one STAR had less purchased wine cases. This may be due to lower pricing and/or the perception that more value is found in the one STAR wine compared to two STARS wine.

Flag variables were made for each of these STARS “levels” but there wasn’t much of a change in the model compared to just using IMP_STARS. Since IMP_STARS is easier to use, we will continue to only use that variable.

Table of IMP_STARS by TARGET_FLAG			
IMP_STARS	**TARGET_FLAG**		
Frequency **Percent** **Row Pct** **Col Pct**	**0**	**1**	**Total**
1	607 4.74 19.95 22.20	2435 19.03 80.05 24.20	3042 23.77
2	2127 16.62 30.70 77.80	4802 37.53 69.30 47.73	6929 54.15
3	0 0.00 0.00 0.00	2212 17.29 100.00 21.99	2212 17.29
4	0 0.00 0.00 0.00	612 4.78 100.00 6.08	612 4.78
Total	2734 21.37	10061 78.63	12795 100.00

BUILD MODELS:

Using our transformed and imputed variables, we will create five different models on the data set. The models will be Linear Regression, Poisson, Negative Binomial, Zero Inflated Poisson, and Zero Inflated Negative Binomial distribution models.

One point of note is that the Poisson and Negative Binomial distributions both produced almost identical results so I removed some variables from the Poisson model for variety.

IMP_CitricAcid, IMP_pH, IMP_Sulphates were removed because they were the least significant per the ChiSqr from the Negative Binomial model results.

Regression:

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	6.19582	0.20075	30.86	<.0001	0
LOG10_IMP_AcidIndex	1	-4.28268	0.19562	-21.89	<.0001	1.05140
IMP_Chlorides	1	-0.13271	0.04090	-3.25	0.0012	1.00218
IMP_CitricAcid	1	0.02374	0.01450	1.64	0.1014	1.00574
IMP_FreeSulfurDioxide	1	0.00031707	0.00008731	3.63	0.0003	1.00330
IMP_LabelAppeal	1	0.46392	0.01368	33.90	<.0001	1.10541
IMP_pH	1	-0.04030	0.01812	-2.22	0.0261	1.00543
IMP_STARS	1	0.78380	0.01567	50.01	<.0001	1.09744
M_STARS	1	-2.25192	0.02696	-83.52	<.0001	1.04653
IMP_Sulphates	1	-0.03570	0.01426	-2.50	0.0123	1.00217
IMP_TotalSulfurDioxide	1	0.00024425	0.00005639	4.33	<.0001	1.00370
IMP_VolatileAcidity	1	-0.10534	0.01605	-6.56	<.0001	1.00657

Poisson:

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	2.3548	0.0882	2.1820	2.5276	713.02	<.0001
LOG10_IMP_AcidIndex	1	-1.5932	0.0925	-1.7746	-1.4119	296.47	<.0001
IMP_Chlorides	1	-0.0415	0.0180	-0.0767	-0.0062	5.32	0.0211
IMP_FreeSulfurDioxid	1	0.0001	0.0000	0.0000	0.0002	8.15	0.0043

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
IMP_LabelAppeal	1	0.1579	0.0061	0.1459	0.1699	664.85	<.0001
IMP_STARS	1	0.1898	0.0061	0.1779	0.2017	976.28	<.0001
M_STARS	1	-1.0292	0.0170	-1.0625	-0.9960	3680.60	<.0001
IMP_TotalSulfurDioxi	1	0.0001	0.0000	0.0000	0.0001	12.31	0.0004
IMP_VolatileAcidity	1	-0.0346	0.0071	-0.0485	-0.0208	23.98	<.0001
Scale	0	1.0000	0.0000	1.0000	1.0000		

Negative Binomial:

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	2.4256	0.0939	2.2416	2.6096	667.58	<.0001
LOG10_IMP_AcidIndex	1	-1.6104	0.0930	-1.7927	-1.4281	299.71	<.0001
IMP_Chlorides	1	-0.0421	0.0180	-0.0774	-0.0068	5.48	0.0193
IMP_CitricAcid	1	0.0075	0.0063	-0.0049	0.0199	1.41	0.2345
IMP_FreeSulfurDioxid	1	0.0001	0.0000	0.0000	0.0002	8.24	0.0041
IMP_LabelAppeal	1	0.1582	0.0061	0.1462	0.1702	666.64	<.0001
IMP_pH	1	-0.0156	0.0080	-0.0312	0.0000	3.84	0.0501
IMP_STARS	1	0.1896	0.0061	0.1777	0.2015	973.79	<.0001
M_STARS	1	-1.0277	0.0170	-1.0609	-0.9944	3666.43	<.0001
IMP_Sulphates	1	-0.0135	0.0062	-0.0258	-0.0013	4.68	0.0304
IMP_TotalSulfurDioxi	1	0.0001	0.0000	0.0000	0.0001	12.25	0.0005
IMP_VolatileAcidity	1	-0.0343	0.0071	-0.0482	-0.0205	23.59	<.0001
Dispersion	1	0.0000	0.0001	0.0000	2.29E286		

Zero Inflated Poisson:

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.2233	0.1063	1.0150	1.4316	132.47	<.0001
LOG10_IMP_AcidIndex	1	-0.4840	0.1120	-0.7036	-0.2644	18.66	<.0001
IMP_Chlorides	1	-0.0274	0.0209	-0.0683	0.0136	1.71	0.1906
IMP_FreeSulfurDioxid	1	0.0000	0.0000	-0.0001	0.0001	0.57	0.4512
IMP_LabelAppeal	1	0.2948	0.0072	0.2807	0.3089	1672.80	<.0001
IMP_STARS	1	0.1238	0.0072	0.1097	0.1379	296.81	<.0001
M_STARS	1	-0.2106	0.0207	-0.2512	-0.1699	103.22	<.0001
IMP_TotalSulfurDioxi	1	-0.0000	0.0000	-0.0001	0.0000	2.22	0.1363
IMP_VolatileAcidity	1	-0.0133	0.0083	-0.0295	0.0029	2.60	0.1067
Scale	0	1.0000	0.0000	1.0000	1.0000		

Zero Inflated Negative Binomial:

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.1910	0.1129	0.9697	1.4123	111.28	<.0001
LOG10_IMP_AcidIndex	1	-0.4783	0.1126	-0.6990	-0.2576	18.05	<.0001
IMP_Chlorides	1	-0.0270	0.0209	-0.0680	0.0140	1.67	0.1964
IMP_CitricAcid	1	0.0018	0.0073	-0.0125	0.0162	0.06	0.8014
IMP_FreeSulfurDioxid	1	0.0000	0.0000	-0.0001	0.0001	0.57	0.4516
IMP_LabelAppeal	1	0.2947	0.0072	0.2806	0.3088	1671.05	<.0001
IMP_pH	1	0.0082	0.0093	-0.0101	0.0264	0.77	0.3801

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
IMP_STARS	1	0.1238	0.0072	0.1097	0.1379	296.88	<.0001
M_STARS	1	-0.2105	0.0207	-0.2511	-0.1698	103.12	<.0001
IMP_Sulphates	1	0.0005	0.0073	-0.0138	0.0148	0.00	0.9450
IMP_TotalSulfurDioxi	1	-0.0000	0.0000	-0.0001	0.0000	2.22	0.1360
IMP_VolatileAcidity	1	-0.0133	0.0083	-0.0295	0.0029	2.59	0.1078
Dispersion	0	0.0000	0.0000	0.0000	0.0000		

There are four variables that consistently produce the most significant Chi-Square or T-test values regardless of model. They are IMP_STARS, M_STARS, IMP_LabelAppeal, and LOG10_IMP_AcidIndex. In every model, M_STARS, which means the STARS were missing from the wine, reduces the probability or number of purchases predicted. The same is true for LOG10_IMP_AcidIndex, meaning higher AcidIndex values means lower number or probability of sample wine cases purchased. However, IMP_STARS, which means there were one or more STARS on the wine, and IMP_LabelAppeal, which means the wine label has a positive appeal to the customer, both increased the number or probability of a sample case of wine being selected by the distribution company.

Other variables such as TotalSulferDioxide and FreeSulferDioxide both had very little effect on the model as their coefficients were almost zero in every model. Updated models may consider removing these variables as they don’t appear to add anything to any model.

SELECT MODEL:

Since it is difficult to compare metrics across different types of models, the decision of the best model was determined by the error of the predicted value compared to the target value. For example, the difference between the target value and predicted value is calculated as the error. The error is then squared to compensate for any negative values. This value is calculated for all observations and summed to create a total error value.

Each model’s error is calculated the same way and totaled. The model with the lowest total value, which means the model had the least errors compared to the data set’s target values, is selected as the best model.

In this case, the zero inflated negative binomial model produced the lowest sum of squared error total. Actually both zero inflated models produced the best results, followed by the linear regression model (which theoretically shouldn't do well because it violates many assumptions), then the Poisson and Negative Binomial distribution models. Also, it should be noted that I combined all five models together and took an average of their errors to produce ERROR_ENS which was actually produced the third best result.

Variable	Sum	Mean
ERROR_R	21998.70	1.7193199
ERROR_POI	22241.38	1.7382868
ERROR_NB	22243.38	1.7384428
ERROR_HPOI	20772.01	1.6234474
ERROR_HNB	20767.31	1.6230798
ERROR_ENS	21048.55	1.6450609

CONCLUSION:

The results of the analysis show that best model to use for predicting the number of cases purchased by the wine distribution company is the zero inflated negative binomial model. This makes sense because it compensates for the high number of zero values that are found in the unaltered raw data set.

The non-zero inflated models are trying to create a model when there are almost more zero values than any other number of cases purchased. This type of data set can disproportionately skew the results. Thus, often a more accurate model can be created when we separate those zero values and just produce a model on the remaining observations. By producing two models in one, one to predict if wine cases were purchased or not and the other predicting how many if they were purchased, we can often produce a more accurate model.

In this case (again, pun intended), we produced a model that can give a wine producer a competitive advantage in terms of predicting how well their wine will fare, which variables are most likely to effect the wine's chances of being purchased (IMP_STARS, M_STARS, IMP_LabelAppeal, and LOG10_AcidityIndex), and how to direct the company's resources to maximize profit in the competitive wine industry.

********** BINGO BONUS *******************;

I went for 40 extra bingo bonus points: 20pts for developing a logistic/poisson model, 10pts for using SAS Macros, 10 pts for handing in SCORED FILE as SAS DATA SET.

1) 20 pts Logistic/Poisson model

```
* HURDLE POI MODEL - POISSON/LOGISTIC;

* Logistic prediction if wine purchased or not;
proc logistic data=&FIXFILE.;
model TARGET_FLAG(ref="0") =
                                LOG10_IMP_AcidIndex
                                IMP_Chlorides
                                IMP_CitricAcid
                                IMP_FreeSulfurDioxide
                                IMP_LabelAppeal
                                IMP_pH
                                IMP_STARS
                                M_STARS
                                IMP_Sulphates
                                IMP_TotalSulfurDioxide
                                IMP_VolatileAcidity
                                ;
output out=&FIXFILE. p=X_LOGIT_PROB_POI;
TITLE5 "Hurdle Poisson + Logistic on FIXFILE";
run;

* checking that PROB prediction amount vs target_flag is fairly accurate;
proc print data=&FIXFILE.(obs=10);
var TARGET_FLAG X_LOGIT_PROB_POI;
run;

* Poisson GENMOD for Poisson/Logistic Hurdle method;
proc genmod data=&FIXFILE.;
model TARGET_AMT =
                                LOG10_IMP_AcidIndex
                                IMP_Chlorides
                                IMP_FreeSulfurDioxide
                                IMP_LabelAppeal
                                IMP_STARS
                                M_STARS
                                IMP_TotalSulfurDioxide
                                IMP_VolatileAcidity
                                /link=log dist=poi
                                ;
```

```
output out=&FIXFILE. p=X_GENMOD_HURDLE_POI;
run;
```

2) 10 pts – using MACROS

```
MACROS;
%let PATH = /home/derekhughes2014/DATAFILES/;
%let NAME = mydata;
%let LIB = &NAME..;

libname &NAME. "&PATH." access=readonly;

%let INFILE   = &LIB.wine;
%let INFILE2 = &LIB.wine_test;

%let TEMPFILE       = TEMPFILE;
%let MISSFILE       = MISSFILE;
%let FIXFILE = FIXFILE;
%let VARLIST        = VARLIST;
```

3) 10 pts - for handing in SCORED FILE as SAS DATA SET

See file named: derek_hughes_file_wine_test

```
* print a few observations to ensure can access the dataset (wine_test);
proc print data=&INFILE2. (obs=5);
title10 "Testing Access to Wine_test - dataset";
run;
title10 ;

* code to store scored code into my SAS folder Assignments;
libname scorelib "/home/derekhughes2014/Assignments";
data scorelib.DEREK_HUGHES_FILE_wine_test;
set SCOREFILE;
run;

* view scored data on Wine_test - click "download" button
* in Folders to get this file on local CPU;
proc print data=scorelib.DEREK_HUGHES_FILE_wine_test (obs=10);
title10 "Hurdle Poisson/Logistic vs Dataset in SCOREfILE";
run;
```

****************CODE****************

```
* Derek Hughes - Assignment 3 - PRED 411 - Sec55 - Winter 2015

MACROS;
%let PATH = /home/derekhughes2014/DATAFILES/;
%let NAME = mydata;
%let LIB = &NAME..;

libname &NAME. "&PATH." access=readonly;

%let INFILE   = &LIB.wine;
%let INFILE2 = &LIB.wine_test;

%let TEMPFILE        = TEMPFILE;
%let MISSFILE         = MISSFILE;
%let FIXFILE = FIXFILE;
%let VARLIST          = VARLIST;


* check that can access INFILE;
proc print data=&INFILE.(obs=10);
run;


* make dataset copy so safe for adjustments (tempfile);
data &TEMPFILE.;
set &INFILE.;
TARGET_FLAG = ( TARGET > 0 );
run;

*** Early EDA exploring ***;
proc contents data=&TEMPFILE.;
run;

* check values are correct/accurate on MISSFILE;
proc means data=&TEMPFILE. nmiss mean median min max;
run;

* observe values using histogram and other measures;
proc univariate data=&TEMPFILE.;
histogram STARS LabelAppeal TARGET;
run;
```

```
* ----------------- Beginning of variable selection process --------
  We will compare variables selected via Linear and Logistic regression stepwise
selection proccess
  using an unaltered data set and one with only the missing values imputed. AFter
identifying the
  common variables we will compare the means of those variables when wine is bought to
when wine
  is not purchased. Large differences in these means will be used as indicator of another
possibly
  predictive variable.;

* MISSING VALUE DATASET - create dataset with missing variables removed;
data &MISSFILE.;
set &TEMPFILE.;

IMP_AcidIndex                            = AcidIndex;
IMP_Alcohol                              = Alcohol;
M_Alcohol                                = 0;
IMP_Chlorides                            = Chlorides;
M_Chlorides                              = 0;
IMP_CitricAcid                           = CitricAcid;
IMP_Density                              = Density;
IMP_FixedAcidity                 = FixedAcidity;
IMP_FreeSulfurDioxide            = FreeSulfurDioxide;
M_FreeSulfurDioxide              = 0;
IMP_LabelAppeal                          = LabelAppeal;
IMP_pH                                           = pH;
M_pH                                     = 0;
IMP_STARS                                = STARS;
M_STARS                                          = 0;
IMP_ResidualSugar                = ResidualSugar;
M_ResidualSugar                          = 0;
IMP_Sulphates                            = Sulphates;
M_Sulphates                              = 0;
IMP_TotalSulfurDioxide           = TotalSulfurDioxide;
M_TotalSulfurDioxide             = 0;
IMP_VolatileAcidity  = VolatileAcidity;


* setting missing values;
if missing(STARS)                                then do;
        IMP_STARS                                = 2;                     M_STARS= 1;
                end;
if missing(Alcohol)                              then do;
```

```
       IMP_Alcohol                        = 10.4892363; M_Alcohol=1;
       end;
if missing(Chlorides)              then do;
       IMP_Chlorides                      = 0.0548225;  M_Chlorides=1;
       end;
if missing(FreeSulfurDioxide)      then do;
       IMP_FreeSulfurDioxide       = 30.8455713; M_FreeSulfurDioxide=1;        end;
if missing(pH)                            then do;
       IMP_pH                                    = 3.2076282;  M_pH=1;
                     end;
if missing(ResidualSugar)          then do;
       IMP_ResidualSugar           = 5.4187331;  M_ResidualSugar=1;            end;
if missing(Sulphates)              then do;
       IMP_Sulphates                      = 0.5271118;  M_Sulphates=1;
       end;
if missing(TotalSulfurDioxide)     then do;
       IMP_TotalSulfurDioxide      = 120.7142326;       M_TotalSulfurDioxide=1;
       end;


keep   TARGET
              TARGET_FLAG
              IMP_AcidIndex
              IMP_Alcohol
              M_Alcohol
              IMP_Chlorides
              M_Chlorides
              IMP_CitricAcid
              IMP_Density
              IMP_FixedAcidity
              IMP_FreeSulfurDioxide
              M_FreeSulfurDioxide
              IMP_LabelAppeal
              IMP_pH
              M_pH
              IMP_STARS
              M_STARS
              IMP_ResidualSugar
              M_ResidualSugar
              IMP_Sulphates
              M_Sulphates
              IMP_TotalSulfurDioxide
              M_TotalSulfurDioxide
              IMP_VolatileAcidity
              ;
run;
```

```
* check values are correct/accurate on MISSFILE;
proc means data=&MISSFILE. nmiss mean median min max;
run;



* MISSING VALUE DATA TEST - running full model regression with stepwise for
variable selection;
proc reg data=&MISSFILE.;
model TARGET =
                                        IMP_AcidIndex
                                        IMP_Alcohol
                                        M_Alcohol
                                        IMP_Chlorides
                                        M_Chlorides
                                        IMP_CitricAcid
                                        IMP_Density
                                        IMP_FixedAcidity
                                        IMP_FreeSulfurDioxide
                                        M_FreeSulfurDioxide
                                        IMP_LabelAppeal
                                        IMP_pH
                                        M_pH
                                        IMP_STARS
                                        M_STARS
                                        IMP_ResidualSugar
                                        M_ResidualSugar
                                        IMP_Sulphates
                                        M_Sulphates
                                        IMP_TotalSulfurDioxide
                                        M_TotalSulfurDioxide
                                        IMP_VolatileAcidity / selection=stepwise vif aic;
        title5 "MISSING LINEAR REG full model - stepwise";
run;

* MISSING VALUE DATA TEST - running full model logistic regression (against
buying or not buying TARGET) for variable selection;
proc logistic data=&MISSFILE. plot(only)=(roc(ID=prob));
model TARGET_FLAG(ref="0") =
                                        IMP_AcidIndex
                                        IMP_Alcohol
                                        M_Alcohol
                                        IMP_Chlorides
                                        M_Chlorides
                                        IMP_CitricAcid
                                        IMP_Density
                                        IMP_FixedAcidity
                                        IMP_FreeSulfurDioxide
                                        M_FreeSulfurDioxide
                                        IMP_LabelAppeal
                                        IMP_pH
                                        M_pH
                                        IMP_STARS
                                        M_STARS
                                        IMP_ResidualSugar
                                        M_ResidualSugar
                                        IMP_Sulphates
                                        M_Sulphates
                                        IMP_TotalSulfurDioxide
                                        M_TotalSulfurDioxide
                                        IMP_VolatileAcidity / selection=stepwise
roceps=0.1;
        title5 "MISSING LOGISTIC REG full model - stepwise";
run;


* UNALTERED DATA TEST - running FULL MODEL REGRESSION with stepwise
for variable selection;
proc reg data=&TEMPFILE.;
model TARGET =
                                        AcidIndex
                                        Alcohol
                                        Chlorides
                                        CitricAcid
                                        Density
                                        FixedAcidity
                                        FreeSulfurDioxide
                                        LabelAppeal
                                        pH
                                        STARS
                                        ResidualSugar
                                        Sulphates
                                        TotalSulfurDioxide
                                        VolatileAcidity / selection=stepwise vif aic;
        title5 "UNALTERED - LINEAR REG full model - stepwise";
run;


* UNALTERED DATA TEST - running FULL MODEL LOGISTIC regression (against
buying or not buying TARGET) for variable selection;
proc logistic data=&TEMPFILE. plot(only)=(roc(ID=prob));
```

```
model TARGET_FLAG(ref="0") =
                                        AcidIndex
                                        Alcohol
                                        Chlorides
                                        CitricAcid
                                        Density
                                        FixedAcidity
                                        FreeSulfurDioxide
                                        LabelAppeal
                                        pH
                                        STARS
                                        ResidualSugar
                                        Sulphates
                                        TotalSulfurDioxide
                                        VolatileAcidity / selection=stepwise roceps=0.1;
        title5 "UNALTERED - LOGISTIC full model - stepwise";
run;



* ----------COMPARING MEAN VALUES on unaltered data set vs missing values
imputed data set----------
  Here I am comparing the variable values when wine bought and when not bought. If I
see a large difference
  between values then I will manually select that variable into the model;

* TEMPFILE - observe missing values and means for unaltered dataset;
proc means data=&TEMPFILE. nmiss mean median min max;
title5 "UNALTERED dataset";
run;

* TEMPFILE - observe missing values compared to buying or not buying wine cases;
proc means data=&TEMPFILE. nmiss mean median min max;
class TARGET_FLAG;
title5 "UNALTERED dataset vs target_flag";
run;

* MISSFILE - observe missing values and means for unaltered dataset;
proc means data=&MISSFILE. nmiss mean median min max;
title5 "Missing values imputed dataset";
run;

* MISSFILE - observe missing values compared to buying or not buying wine cases;
proc means data=&MISSFILE. nmiss mean median min max;
class TARGET_FLAG;
title5 "Missing values imputed dataset vs target_flag";
```

```
run;
* ----------------- Completion of variable selection process -------- ;


* create histograms for comparison to transformed values from next data set;
proc univariate data=&MISSFILE.;
var IMP_TotalSulfurDioxide IMP_VolatileAcidity IMP_Sulphates IMP_pH
IMP_FreeSulfurDioxide IMP_CitricAcid IMP_Chlorides;
histogram IMP_TotalSulfurDioxide IMP_VolatileAcidity IMP_Sulphates IMP_pH
IMP_FreeSulfurDioxide IMP_CitricAcid IMP_Chlorides;
TITLE5 "Histograms for missing value data set"
run;



* Now that we've finished using MISSFILE for variable selection processes we
  will convert back to a TEMPFILE  that includes the MISSFILE data and
  use it for more imputations and trimming of the selected variables.
  This TEMPFILE includes the variables selected from above process;
data &TEMPFILE.;
set &MISSFILE.;

TARGET_AMT = TARGET - 1;
if TARGET_FLAG = 0 then TARGET_AMT = .;


* trimming values imputations;
if IMP_TotalSulfurDioxide    < -330 then IMP_TotalSulfurDioxide = -330;
if IMP_TotalSulfurDioxide    > 630  then IMP_TotalSulfurDioxide = 630;

if IMP_VolatileAcidity         < -1.3 then IMP_VolatileAcidity = -1.3;
if IMP_VolatileAcidity         > 2.2  then IMP_VolatileAcidity = 2.2;

if IMP_Sulphates       < -1.5 then IMP_Sulphates = -1.5;
if IMP_Sulphates       > 2.5  then IMP_Sulphates = 2.5;

if IMP_pH              < 1.4 then IMP_pH      = 1.4;
if IMP_pH              > 5  then IMP_pH       = 5;

if IMP_FreeSulfurDioxide              < -300 then IMP_FreeSulfurDioxide  = -300;
if IMP_FreeSulfurDioxide              > 345  then IMP_FreeSulfurDioxide  = 345;

if IMP_CitricAcid              < -1.6 then IMP_CitricAcid    = -1.6;
if IMP_CitricAcid              > 2.4  then IMP_CitricAcid     = 2.4;

if IMP_Chlorides               < -0.6 then IMP_Chlorides     = -0.6;
if IMP_Chlorides                  > .75  then IMP_Chlorides     = .75;


* transformations;
LOG10_IMP_AcidIndex = sign(IMP_AcidIndex) * log10(abs(IMP_AcidIndex)+1);
* SQRT produced a normal distribution that wasn't as symmetric as Log transformation
SQRT_IMP_AcidIndex = sign(IMP_AcidIndex) * sqrt(abs(IMP_AcidIndex)+1);
drop IMP_AcidIndex;


* these were the common variables from selection process with linear and logistic
regressions;
keep    TARGET
                TARGET_FLAG
                TARGET_AMT
                LOG10_IMP_AcidIndex
                IMP_Chlorides
                IMP_CitricAcid
                IMP_FreeSulfurDioxide
                IMP_LabelAppeal
                IMP_pH
                IMP_STARS
                M_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                IMP_VolatileAcidity
                ;

run;



* --------------- EDA work for missing values and trim imputations ---------- ;
* proc univariate to view histograms of selected variables to adjust for missing values,
imputations;
proc univariate data=&TEMPFILE. ;
var IMP_TotalSulfurDioxide IMP_VolatileAcidity IMP_Sulphates IMP_pH
IMP_FreeSulfurDioxide IMP_CitricAcid IMP_Chlorides;
histogram IMP_TotalSulfurDioxide IMP_VolatileAcidity IMP_Sulphates IMP_pH
IMP_FreeSulfurDioxide IMP_CitricAcid IMP_Chlorides;
TITLE5 "Histograms after transformations";
run;


* check that target variables are displaying correctly;
proc print data=&TEMPFILE.(obs=20);
```

```
var TARGET TARGET_FLAG TARGET_AMT;
run;

/*
proc freq data=&TEMPFILE.;
table TARGET_FLAG /missing;
run;
*/



* --------- EDA of means standard and vs. Target_Flag ---------------- ;
* means vs variables for tempfile dataset;
proc means data=&TEMPFILE. nmiss mean median min max;
var
                LOG10_IMP_AcidIndex
                IMP_Chlorides
                IMP_CitricAcid
                IMP_FreeSulfurDioxide
                IMP_LabelAppeal
                IMP_pH
                IMP_STARS
                M_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                IMP_VolatileAcidity
        ;
run;


* means of variables against TARGET_FLAG for tempfile dataset;
proc means data=&TEMPFILE. nmiss mean median min max;
class TARGET_FLAG;
var
                LOG10_IMP_AcidIndex
                IMP_Chlorides
                IMP_CitricAcid
                IMP_FreeSulfurDioxide
                IMP_LabelAppeal
                IMP_pH
                IMP_STARS
                M_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                IMP_VolatileAcidity
        ;
```

```
run;


* observing interaction between IMP_STARS and TARGET_FLAG;
proc freq data=&TEMPFILE.;
table IMP_STARS*TARGET_FLAG /missing;
run;
* ------------------------ ;




* ------------------------ ;
* creating new dataset (fixfile) for applying to models;
* we will use this dataset for creating all model coefficients;
data &FIXFILE.;
set &TEMPFILE.;
run;

* checking that tempfile copied correctly to fixfile;
proc print data=&FIXFILE.(obs=10);
run;

* checking target value for fixfile;
proc univariate data=&FIXFILE. noprint;
histogram TARGET;
run;
* ------------------------ ;



* ------------------------ ;
* REGRESSION MODEL;

* reseting fixfile;
data &FIXFILE.;
set &TEMPFILE.;
run;

* used the same variables as selected from regression and logistic variable selection
methods;
proc reg data=&FIXFILE.;
model TARGET =
                                    LOG10_IMP_AcidIndex
                                    IMP_Chlorides
                                    IMP_CitricAcid
```

```
                IMP_FreeSulfurDioxide
                IMP_LabelAppeal
                IMP_pH
                IMP_STARS
                M_STARS
                IMP_Sulphates
                IMP_TotalSulfurDioxide
                IMP_VolatileAcidity
                /selection = stepwise aic vif;
    output out=&FIXFILE. predicted=X_REGRESSION;
    TITLE5 "Regression on FIXFILE";
run;
quit;

* checking that tempfile copied correctly to fixfile;
proc print data=&FIXFILE.(obs=10);
run;


* score model against fixfile to check for accuracy;
data SCOREFILE;
set &FIXFILE.;

* used the same variables as selected from regression and logistic variable selection
methods;
P_REGRESSION =    6.19582
    +
                LOG10_IMP_AcidIndex             *(-4.28268)    +
                IMP_Chlorides                   *(-0.13271)    +
                IMP_CitricAcid                  *(0.02374)
    +
                IMP_FreeSulfurDioxide       *(0.00031707)      +
                IMP_LabelAppeal                 *(0.46392)
    +
                IMP_pH                              *(-0.04030)
    +
                IMP_STARS                       *(0.78380)
    +
                M_STARS                             *(-2.25192)
    +
                IMP_Sulphates                   *(-0.03570)    +
                IMP_TotalSulfurDioxide      *(0.00024425)      +
                IMP_VolatileAcidity         *(-0.10534)
                ;

run;
```

```
* checking results of scorefile for errors and accuracy;
proc print data=SCOREFILE(obs=10);
var TARGET X_REGRESSION P_REGRESSION;
run;





* ------------------------ ;
* NEGATIVE BINOMIAL MODEL w/ GENMOD;

* reset fixfile;
data &FIXFILE.;
set &TEMPFILE.;
run;

* used the same variables as selected from regression and logistic variable selection methods;
proc genmod data=&FIXFILE.;
model TARGET =
                              LOG10_IMP_AcidIndex
                              IMP_Chlorides
                              IMP_CitricAcid
                              IMP_FreeSulfurDioxide
                              IMP_LabelAppeal
                              IMP_pH
                              IMP_STARS
                              M_STARS
                              IMP_Sulphates
                              IMP_TotalSulfurDioxide
                              IMP_VolatileAcidity
                              /link=log dist=nb
                              ;
output out=&FIXFILE. p=X_GENMOD_NB;
TITLE5 "Negative Binomial on FIXFILE";
run;

* verify results completed properly;
proc print data=&FIXFILE.(obs=10);
run;


* Scorefile for regression, genmod NB;
data SCOREFILE;
```

```
set &FIXFILE.;

P_REGRESSION =   6.19582
        +
                          LOG10_IMP_AcidIndex          *(-4.28268)   +
                          IMP_Chlorides                *(-0.13271)   +
                          IMP_CitricAcid               *(0.02374)
        +
                          IMP_FreeSulfurDioxide   *(0.00031707)      +
                          IMP_LabelAppeal              *(0.46392)
        +
                          IMP_pH                            *(-0.04030)
        +
                          IMP_STARS                    *(0.78380)
        +
                          M_STARS                           *(-2.25192)
        +
                          IMP_Sulphates                *(-0.03570)   +
                          IMP_TotalSulfurDioxide  *(0.00024425)      +
                          IMP_VolatileAcidity     *(-0.10534)
                          ;


P_GENMOD_NB =   2.4256
        +
                          LOG10_IMP_AcidIndex          *(-1.6104)    +
                          IMP_Chlorides                *(-0.0421)    +
                          IMP_CitricAcid               *(0.0075)
        +
                          IMP_FreeSulfurDioxide   *(0.0001)     +
                          IMP_LabelAppeal              *(0.1582)
        +
                          IMP_pH                            *(-0.0156)
        +
                          IMP_STARS                    *(0.1896)
        +
                          M_STARS                           *(-1.0277)
        +
                          IMP_Sulphates                *(-0.0135)    +
                          IMP_TotalSulfurDioxide  *(0.0001)     +
                          IMP_VolatileAcidity     *(-0.0343)
                          ;
P_GENMOD_NB = exp(P_GENMOD_NB);

run;
```

```
* checking results of scorefile for errors and accuracy;
proc print data=SCOREFILE(obs=10);
var TARGET X_GENMOD_NB P_GENMOD_NB P_REGRESSION;
run;





* ------------------------- ;
* POISSON MODEL w/ GENMOD;

* reset fixfile;
data &FIXFILE.;
set &TEMPFILE.;
run;

* The Poisson model produced the same results as NB so removed a few variables to
create a difference
  -I removed IMP_CitricAcid, IMP_pH, IMP_Sulphates because they were the least
significant per the ChiSqr from
  NB model results;
proc genmod data=&FIXFILE.;
model TARGET =
                                        LOG10_IMP_AcidIndex
                                        IMP_Chlorides
                                        IMP_FreeSulfurDioxide
                                        IMP_LabelAppeal
                                        IMP_STARS
                                        M_STARS
                                        IMP_TotalSulfurDioxide
                                        IMP_VolatileAcidity
                                        /link=log dist=poi
                                        ;
output out=&FIXFILE. p=X_GENMOD_POI;
TITLE5 "Poisson on FIXFILE";
run;

* verify results completed properly;
proc print data=&FIXFILE.(obs=10);
run;


* Scorefile for regression, genmod NB, genmod POI;
data SCOREFILE;
set &FIXFILE.;
```

```
P_REGRESSION =    6.19582
        +
                    LOG10_IMP_AcidIndex             *(-4.28268)    +
                    IMP_Chlorides                   *(-0.13271)    +
                    IMP_CitricAcid                  *(0.02374)
        +
                    IMP_FreeSulfurDioxide     *(0.00031707)        +
                    IMP_LabelAppeal                 *(0.46392)
        +
                    IMP_pH                                *(-0.04030)
        +
                    IMP_STARS                       *(0.78380)
        +
                    M_STARS                               *(-2.25192)
        +
                    IMP_Sulphates                   *(-0.03570)    +
                    IMP_TotalSulfurDioxide    *(0.00024425)        +
                    IMP_VolatileAcidity       *(-0.10534)
                    ;


P_GENMOD_NB =     2.4256
        +
                    LOG10_IMP_AcidIndex             *(-1.6104)     +
                    IMP_Chlorides                   *(-0.0421)     +
                    IMP_CitricAcid                  *(0.0075)
        +
                    IMP_FreeSulfurDioxide     *(0.0001)      +
                    IMP_LabelAppeal                 *(0.1582)
        +
                    IMP_pH                                *(-0.0156)
        +
                    IMP_STARS                       *(0.1896)
        +
                    M_STARS                               *(-1.0277)
        +
                    IMP_Sulphates                   *(-0.0135)     +
                    IMP_TotalSulfurDioxide    *(0.0001)      +
                    IMP_VolatileAcidity       *(-0.0343)
                    ;
P_GENMOD_NB = exp(P_GENMOD_NB);


P_GENMOD_POI =    2.3548
        +
                              LOG10_IMP_AcidIndex          *(-1.5932)     +
                              IMP_Chlorides                *(-0.0415)     +
                              IMP_FreeSulfurDioxide   *(0.0001)      +
                              IMP_LabelAppeal              *(0.1579)      +
                              IMP_STARS                    *(0.1898)      +
                              M_STARS                              *(-1.0292)
       +
                              IMP_TotalSulfurDioxide  *(0.0001)      +
                              IMP_VolatileAcidity     *(-0.0346)
                              ;
P_GENMOD_POI = exp(P_GENMOD_POI);

run;


* checking results of scorefile for errors and accuracy;
proc print data=SCOREFILE(obs=30);
var TARGET X_GENMOD_POI P_GENMOD_POI P_GENMOD_NB
P_REGRESSION;
run;



* check to see if Poisson and/or NB is appropriate for Zero Inflation models
  variance should be close to mean value
  - the results show an underinflated model as the variance was half the mean value
  the mean and variance should be similar but we will try using the Hurdle method;
proc means data=&TEMPFILE. mean var;
where TARGET > 0;
var TARGET;
run;




* ------------------------- ;
* HURDLE NEGATIVE BINOMIAL MODEL- NEGATIVE BINOMIAL/LOTISTIC;
data &FIXFILE.;
set &TEMPFILE.;
run;

* Logistic prediction if wine purchased or not;
proc logistic data=&FIXFILE.;
model TARGET_FLAG(ref="0") =
                              LOG10_IMP_AcidIndex
                              IMP_Chlorides
```

```
                        IMP_CitricAcid
                        IMP_FreeSulfurDioxide
                        IMP_LabelAppeal
                        IMP_pH
                        IMP_STARS
                        M_STARS
                        IMP_Sulphates
                        IMP_TotalSulfurDioxide
                        IMP_VolatileAcidity
                        ;
output out=&FIXFILE. p=X_LOGIT_PROB;
TITLE5 "Hurdle Negative Binomial + Logistic on FIXFILE";
run;

* checking that PROB prediction amount vs target_flag is fairly accurate;
proc print data=&FIXFILE.(obs=10);
var TARGET_FLAG X_LOGIT_PROB;
run;

* Negative Binomial GENMOD for Negative Binomial/Logistic Hurdle method;
proc genmod data=&FIXFILE.;
model TARGET_AMT =
                        LOG10_IMP_AcidIndex
                        IMP_Chlorides
                        IMP_CitricAcid
                        IMP_FreeSulfurDioxide
                        IMP_LabelAppeal
                        IMP_pH
                        IMP_STARS
                        M_STARS
                        IMP_Sulphates
                        IMP_TotalSulfurDioxide
                        IMP_VolatileAcidity
                        /link=log dist=nb
                        ;
output out=&FIXFILE. p=X_GENMOD_HURDLE;
run;





* ------------------------- ;
* HURDLE POI MODEL - POISSON/LOGISTIC;
data &FIXFILE.;
```

```
set &TEMPFILE.;
run;

* Logistic prediction if wine purchased or not;
proc logistic data=&FIXFILE.;
model TARGET_FLAG(ref="0") =
                        LOG10_IMP_AcidIndex
                        IMP_Chlorides
                        IMP_CitricAcid
                        IMP_FreeSulfurDioxide
                        IMP_LabelAppeal
                        IMP_pH
                        IMP_STARS
                        M_STARS
                        IMP_Sulphates
                        IMP_TotalSulfurDioxide
                        IMP_VolatileAcidity
                        ;
output out=&FIXFILE. p=X_LOGIT_PROB_POI;
TITLE5 "Hurdle Poisson + Logistic on FIXFILE";
run;

* checking that PROB prediction amount vs target_flag is fairly accurate;
proc print data=&FIXFILE.(obs=10);
var TARGET_FLAG X_LOGIT_PROB_POI;
run;

* Poisson GENMOD for Poisson/Logistic Hurdle method;
proc genmod data=&FIXFILE.;
model TARGET_AMT =
                        LOG10_IMP_AcidIndex
                        IMP_Chlorides
                        IMP_FreeSulfurDioxide
                        IMP_LabelAppeal
                        IMP_STARS
                        M_STARS
                        IMP_TotalSulfurDioxide
                        IMP_VolatileAcidity
                        /link=log dist=poi
                        ;
output out=&FIXFILE. p=X_GENMOD_HURDLE_POI;
run;


TITLE5 ;
```

```
data SCOREFILE;
set &FIXFILE.;

* REGRESSION MODEL;
P_REGRESSION =      6.19582
       +
                            LOG10_IMP_AcidIndex             *(-4.28268)    +
                            IMP_Chlorides                   *(-0.13271)    +
                            IMP_CitricAcid                  *(0.02374)
       +
                            IMP_FreeSulfurDioxide    *(0.00031707)         +
                            IMP_LabelAppeal                 *(0.46392)
       +
                            IMP_pH                                 *(-0.04030)
       +
                            IMP_STARS                       *(0.78380)
       +
                            M_STARS                                *(-2.25192)
       +
                            IMP_Sulphates                   *(-0.03570)    +
                            IMP_TotalSulfurDioxide   *(0.00024425)         +
                            IMP_VolatileAcidity      *(-0.10534)
                            ;


* NEGATIVE BINOMIAL MODEL;
P_GENMOD_NB =       2.4256
       +
                            LOG10_IMP_AcidIndex             *(-1.6104)     +
                            IMP_Chlorides                   *(-0.0421)     +
                            IMP_CitricAcid                  *(0.0075)
       +
                            IMP_FreeSulfurDioxide    *(0.0001)      +
                            IMP_LabelAppeal                 *(0.1582)
       +
                            IMP_pH                                 *(-0.0156)
       +
                            IMP_STARS                       *(0.1896)
       +
                            M_STARS                                *(-1.0277)
       +
                            IMP_Sulphates                   *(-0.0135)     +
                            IMP_TotalSulfurDioxide   *(0.0001)      +
                            IMP_VolatileAcidity      *(-0.0343)
                            ;
```

```
P_GENMOD_NB = exp(P_GENMOD_NB);


* POISSON MODEL;
P_GENMOD_POI =   2.3548
        +
                                LOG10_IMP_AcidIndex                     *(-1.5932)      +
                                IMP_Chlorides                           *(-0.0415)      +
                                IMP_FreeSulfurDioxide           *(0.0001)       +
                                IMP_LabelAppeal                         *(0.1579)       +
                                IMP_STARS                               *(0.1898)       +
                                M_STARS                                         *(-1.0292)
        +
                                IMP_TotalSulfurDioxide          *(0.0001)       +
                                IMP_VolatileAcidity             *(-0.0346)
                                ;
P_GENMOD_POI = exp(P_GENMOD_POI);


* HURDLE NEGATIVE BINOMIAL- NEGATIVE BINOMIAL/LOTISTIC;
* Logistic prediction if wine purchased or not;
P_LOGIT_PROB_NB = 7.2581
        +
                                LOG10_IMP_AcidIndex                     *(-8.2831)      +
                                IMP_Chlorides                           *(-0.1651)      +
                                IMP_CitricAcid                          *(0.0381)       +
                                IMP_FreeSulfurDioxide           *(0.000699)    +
                                IMP_LabelAppeal                         *(-0.4685)      +
                                IMP_pH                                          *(-0.2000)
        +
                                IMP_STARS                               *(2.5473)       +
                                M_STARS                                         *(-4.3589)
        +
                                IMP_Sulphates                           *(-0.1249)      +
                                IMP_TotalSulfurDioxide          *(0.000962)    +
                                IMP_VolatileAcidity             *(-0.2024)
                                ;
if P_LOGIT_PROB_NB > 1000 then P_LOGIT_PROB_NB = 1000;
if P_LOGIT_PROB_NB < -1000 then P_LOGIT_PROB_NB = -1000;
P_LOGIT_PROB_NB = exp(P_LOGIT_PROB_NB) / (1+exp(P_LOGIT_PROB_NB));

* Negative Binomial GENMOD for Negative Binomial/Logistic Hurdle method;
P_GENMOD_HURDLE_NB =
                                1.1910
        +
                                LOG10_IMP_AcidIndex                     *(-0.4783)      +
                                IMP_Chlorides                          *(-0.0270)      +
                                IMP_CitricAcid                         *(0.0018)       +
                                IMP_FreeSulfurDioxide          *(0.0000)       +
                                IMP_LabelAppeal                        *(0.2947)       +
                                IMP_pH                                         *(0.0082)
        +
                                IMP_STARS                              *(0.1238)       +
                                M_STARS                                        *(-0.2105)
        +
                                IMP_Sulphates                          *(0.0005)       +
                                IMP_TotalSulfurDioxide         *(-0.0000)      +
                                IMP_VolatileAcidity            *(-0.0133)
                                ;
P_GENMOD_HURDLE_NB = exp(P_GENMOD_HURDLE_NB);
P_HURDLE_NB = P_LOGIT_PROB_NB * (P_GENMOD_HURDLE_NB+1);


* HURDLE POI - POISSION/LOGISTIC;
* Logistic prediction if wine purchased or not;
P_LOGIT_PROB_POI = 7.2581
        +
                                LOG10_IMP_AcidIndex                    *(-8.2831)      +
                                IMP_Chlorides                          *(-0.1651)      +
                                IMP_CitricAcid                         *(0.0381)       +
                                IMP_FreeSulfurDioxide          *(0.000699)     +
                                IMP_LabelAppeal                        *(-0.4685)      +
                                IMP_pH                                         *(-0.2000)
        +
                                IMP_STARS                              *(2.5473)       +
                                M_STARS                                        *(-4.3589)
        +
                                IMP_Sulphates                          *(-0.1249)      +
                                IMP_TotalSulfurDioxide         *(0.000962)     +
                                IMP_VolatileAcidity            *(-0.2024)
                                ;
if P_LOGIT_PROB_POI> 1000 then P_LOGIT_PROB_POI = 1000;
if P_LOGIT_PROB_POI < -1000 then P_LOGIT_PROB_POI = -1000;
P_LOGIT_PROB_POI = exp(P_LOGIT_PROB_POI) / (1+exp(P_LOGIT_PROB_POI));

* Poisson GENMOD for Poisson/Logistic Hurdle method;
P_GENMOD_HURDLE_POI =
                                1.2233
        +
                                LOG10_IMP_AcidIndex                    *(-0.4840)      +
                                IMP_Chlorides                          *(-0.0274)      +
                                IMP_FreeSulfurDioxide          *(0.0000)       +
                                IMP_LabelAppeal                         *(0.2948)       +
                                IMP_STARS                               *(0.1238)       +
                                M_STARS                                         *(-0.2106)
        +
                                IMP_TotalSulfurDioxide          *(-0.0000)      +
                                IMP_VolatileAcidity             *(-0.0133)
                                ;
P_GENMOD_HURDLE_POI = exp(P_GENMOD_HURDLE_POI);
P_HURDLE_POI = P_LOGIT_PROB_POI * (P_GENMOD_HURDLE_POI+1);


* Aggregate of all models;
P_ENSEMBLE = (P_REGRESSION + P_GENMOD_NB + P_GENMOD_POI +
P_HURDLE_NB + P_HURDLE_POI)/5;


/* rounding each predictions to closes single number;
P_REGRESSION    = round(P_REGRESSION    , 1);
P_GENMOD_NB     = round(P_GENMOD_NB             , 1);
P_GENMOD_POI    = round(P_GENMOD_POI    , 1);
P_HURDLE_NB     = round(P_HURDLE_NB             , 1);
P_HURDLE_POI    = round(P_HURDLE_POI    , 1);
P_ENSEMBLE                      = round(P_ENSEMBLE                      , 1);
*/

run;


* print out sample of scorefile to check for display/calculation errors;
proc print data=SCOREFILE(obs=25);
var TARGET P_REGRESSION P_GENMOD_POI P_GENMOD_NB P_HURDLE_NB
P_HURDLE_POI P_ENSEMBLE ;
run;


* adding up total prediction values for each method to compare to target sum
  - closer to target value means model yields closer predicition values
  - in this case the Regression was almost identical to target followed by
P_GENMOD_POI
  - Per this approach, I would use Regression in real life but Poisson per assignment
instructions;
proc means data=SCOREFILE sum mean;
var TARGET P_REGRESSION P_GENMOD_POI P_GENMOD_NB P_HURDLE_NB
P_HURDLE_POI P_ENSEMBLE ;
run;
```

```
*** However, the Sum of Square of the Errors vs the target value is a more accurate test
    to compare models;

data SCOREFILE;
set SCOREFILE;
ERROR_R = TARGET - P_REGRESSION;
ERROR_R = ERROR_R**2;
ERROR_POI = TARGET - P_GENMOD_POI;
ERROR_POI = ERROR_POI**2;
ERROR_NB = TARGET - P_GENMOD_NB;
ERROR_NB = ERROR_NB**2;
ERROR_HPOI = TARGET - P_HURDLE_POI;
ERROR_HPOI = ERROR_HPOI**2;
ERROR_HNB = TARGET - P_HURDLE_NB;
ERROR_HNB = ERROR_HNB**2;
ERROR_ENS = TARGET - P_ENSEMBLE;
ERROR_ENS = ERROR_ENS**2;
run;

proc print data=SCOREFILE(obs=25);
run;

* Here, the lower squared error value is the found with the more accurate model
  - The Hurdle method with the logistic and Poisson method had the lowest error squared;
proc means data=SCOREFILE sum mean;
var TARGET ERROR_R ERROR_POI ERROR_NB ERROR_HPOI ERROR_HNB
ERROR_ENS ;
run;
```

```
**********************************************************;
****      CREATE FILE TO STORE SCORED DATA         ****;
**********************************************************;


* print a few observations to ensure can access the dataset (wine_test);
proc print data=&INFILE2. (obs=5);
title10 "Testing Access to Wine_test - dataset";
run;
title10 ;

* code to store scored code into my SAS folder Assignments;
libname scorelib "/home/derekhughes2014/Assignments";
data scorelib.DEREK_HUGHES_FILE_wine_test;
set SCOREFILE;
run;

* view scored data on Wine_test - click "download" button
* in Folders to get this file on local CPU;
proc print data=scorelib.DEREK_HUGHES_FILE_wine_test (obs=10);
title10 "Hurdle Poisson/Logistic vs Dataset in SCOREfILE";
run;
```