Data Management & Analysis

Final Lab Project

RANA MUHAMMAD ROMAIL

B00885335

INTRODUCTION:

Vinho Verde (pronounced “veeng-yo vaird”) is a Portuguese wine that comes from the region of Vinho Verde, a Denominação de Origem Controlada (DOC), which is the country's largest appellation. Vinho Verde wines are usually made from a blend of native Portuguese grapes and released without being aged. These wines are loved for their mouth-zapping acidity, subtle carbonation, and lower alcohol, making them a great choice for summer. [*Master Class*]

In this report, we’ll be looking into a dataset of the red variant of the Portuguese "Vinho Verde" wine. The data includes physicochemical properties of the red wine as variables such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality of the wine. These properties affect the taste of the wine.

These properties mean:

1 - Fixed acidity (tartaric acid - g / dm^3): most acids involved with wine are fixed or nonvolatile (do not evaporate readily).

2 - Volatile acidity (acetic acid - g / dm^3): It reflects the amount of acetic acid in wine; if its level is high it can lead to an unpleasant, vinegar taste.

3 - Citric acid (g / dm^3): It adds freshness and flavor to wines and is found in small quantities.

4 - Residual sugar (g / dm^3): It is the amount of remains of sugar, after fermentation stops, it’s rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet.

5 – Chlorides (sodium chloride - g / dm^3): It tells about the quantity of salt in the wine.

6 - Free sulfur dioxide (mg / dm^3): The amount of free form of SO_2 that exists in equilibrium between molecular SO_2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine.

7 - Total sulfur dioxide (mg / dm^3): It is amount of free and bound forms of $S0_2$; in low concentrations, SO_2 is mostly undetectable in wine, but at free SO_2 concentrations goes over 50 ppm, SO_2 becomes evident in the nose and taste of wine.

8 – Density (g / cm^3): The density of water is close to that of water depending on the percent alcohol and sugar content.

9 – pH: It describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale.

10 – Sulphates (potassium sulphates - g / dm3): It is a wine additive which can contribute to sulfur dioxide gas ($S0_2$) levels, which acts as an antimicrobial and antioxidant.

11 – Alcohol (% by volume): The percent alcohol content of the wine.

Each ingredient in a recipe is as important as other. People like alcohol in wine so is it the only thing important in wine quality? Which physicochemical properties have the most impact on the quality of the red? How do these properties effect and see if we can determine some logical function to guess the quality of wine on basis of these properties. It can also be determined that if quantity of an ingredients affects quantity of another ingredient. Can we also make a model and see if using some of the properties we can guess the quality of the wine sample? Let’s see.

Let’s have a look at how many samples are there in the data and their quality.

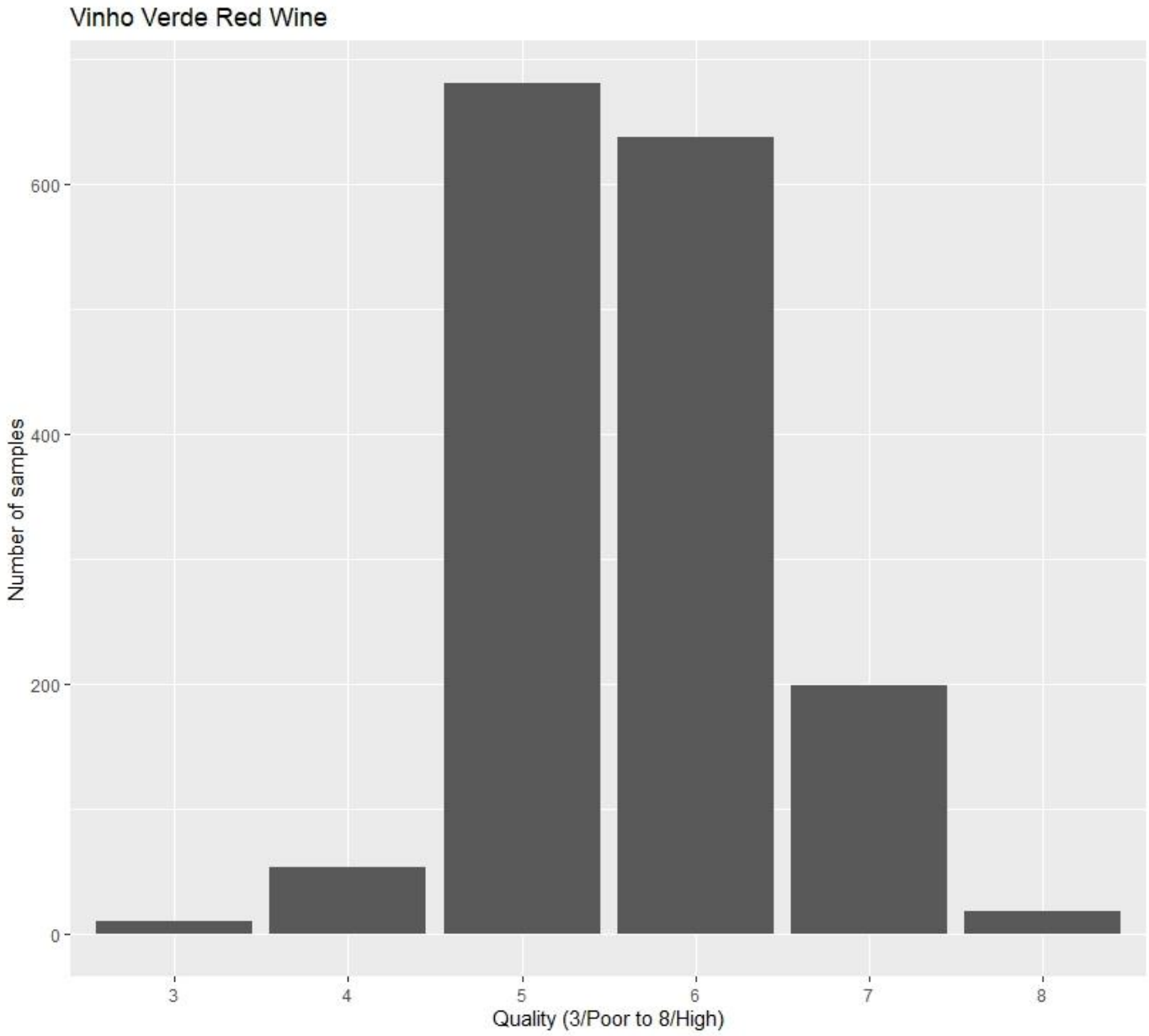

Fig 1

Fig 1 shows a bar graph showing the number of wine samples in each quality category. The quality standard in the given data ranges from 3 to 8 (3 being the low quality and 8 being the best quality). The graph also depicts that the number of samples are not equal in each category. Most of the wine samples fall in 5 and 6 level of quality. We can consider a wine as a good wine if the quality grade is 7 or above.

The difference is obviously due to different values of physicochemical properties of each sample.

Let's see how different properties are related to each other and to the quality.

Correlation Matrix	Fixed Acidity	Volatile acidity	Citric acid	Residual sugar	Chlorides	Free sulfur dioxide	Total sulfur dioxide	Density	pH	Sulphates	Alcohol	Quality
Fixed acidity	1	-0.256	0.67	0.114	0.093	-0.153	-0.113	0.668	-0.682	0.183	-0.061	0.124
Volatile acidity	-0.256	1	-0.552	0.001	0.061	-0.010	0.076	0.022	0.234	-0.26	-0.202	-0.390
Citric acid	0.671	-0.552	1	0.143	0.203	-0.060	0.035	0.364	-0.541	0.312	0.109	0.226
Residual sugar	0.114	0.001	0.143	1	0.055	0.187	0.203	0.355	-0.085	0.005	0.042	0.013
Chlorides	0.093	0.061	0.203	0.055	1	0.005	0.047	0.200	-0.265	0.371	-0.221	-0.128
Free sulfur dioxide	-0.153	-0.010	-0.06	0.187	0.005	1	0.667	-.021	0.070	0.051	-0.069	-0.050
Total sulfur dioxide	-0.113	0.076	0.035	0.203	0.047	0.667	1	0.071	-0.066	0.042	-0.205	-0.185
Density	0.668	0.022	0.364	0.355	0.200	-0.021	0.071	1	-0.341	0.148	-0.496	-0.174
pH	-0.682	0.234	-0.541	-0.085	-0.265	0.070	-0.066	-0.341	1	-0.196	0.205	-0.057
Sulphates	0.183	-0.260	0.312	0.005	0.371	0.051	0.042	0.148	-0.196	1	0.093	0.251
Alcohol	-0.061	-0.202	0.109	0.042	-0.221	-0.069	-0.205	-0.496	0.205	0.093	1	0.476
Quality	0.124	-0.390	0.226	0.013	-0.128	-0.050	-0.185	-0.174	-0.057	0.251	0.476	1

Table 1

Table 1 shows the correlation matrix. This matrix shows the strength of the linear relationship between quantitative variables. If the value in table is 0 then it means there is no relationship in two variables, if the value is greater than 0 then it means that if a variable increase the other also increase and if the value is less than 0 then it means that if a value increases other decreases. In table 1, we can see that alcohol has the highest positive relation with quality with a correlation value of 0.476 in positive sense and volatile acidity, with value of -0.39, has the most negative relationship with the quality of the wine. Some other important variables can also be selected with higher values like sulphates category and citric acid which have an impact on the quality. The table also shows some very strong relation between alcohol and density, fixed acidity and density.

This table shows us some of the strong variables to select but it do not provide any predictive power if we want to see at what rate quality increases with these variables. For the rate we have to do some regression analysis. Before analysis let's see how these relations look on graphs.

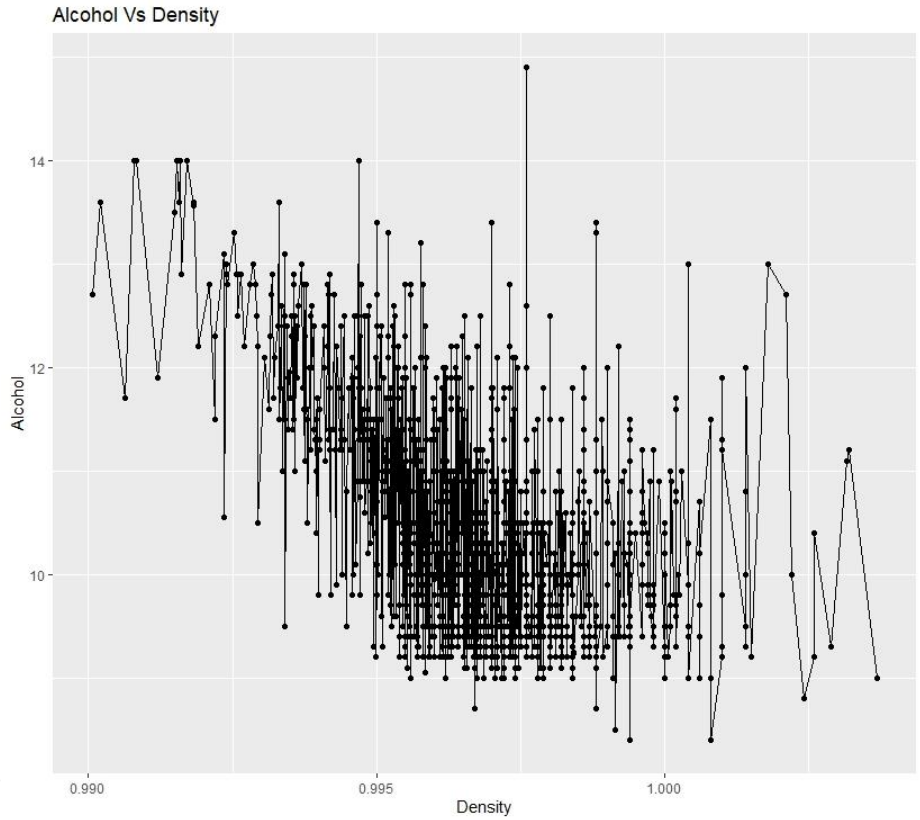

Fig 1

The fig 1 shows that as the percentage of alcohol decreases the density increases. The relation between both properties is opposite or negative as per table 1. In fig 2 the graph shows that as the fixed acidity increases the density also increases.

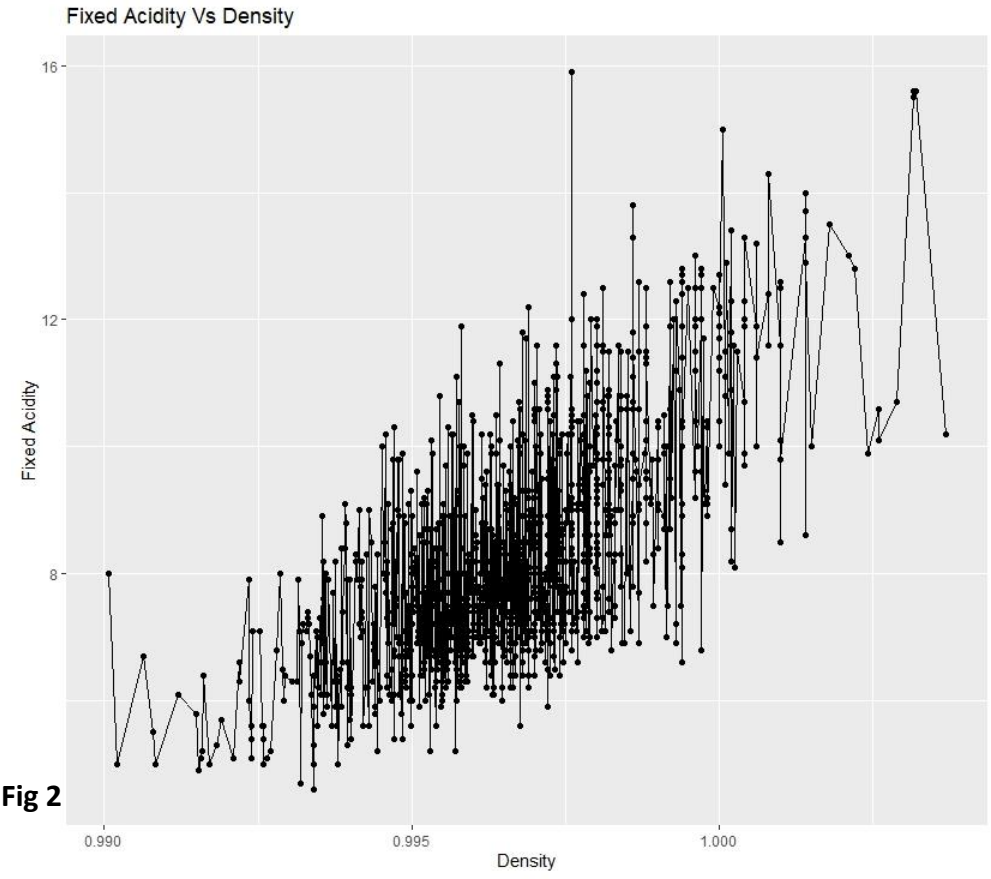

Fig 2

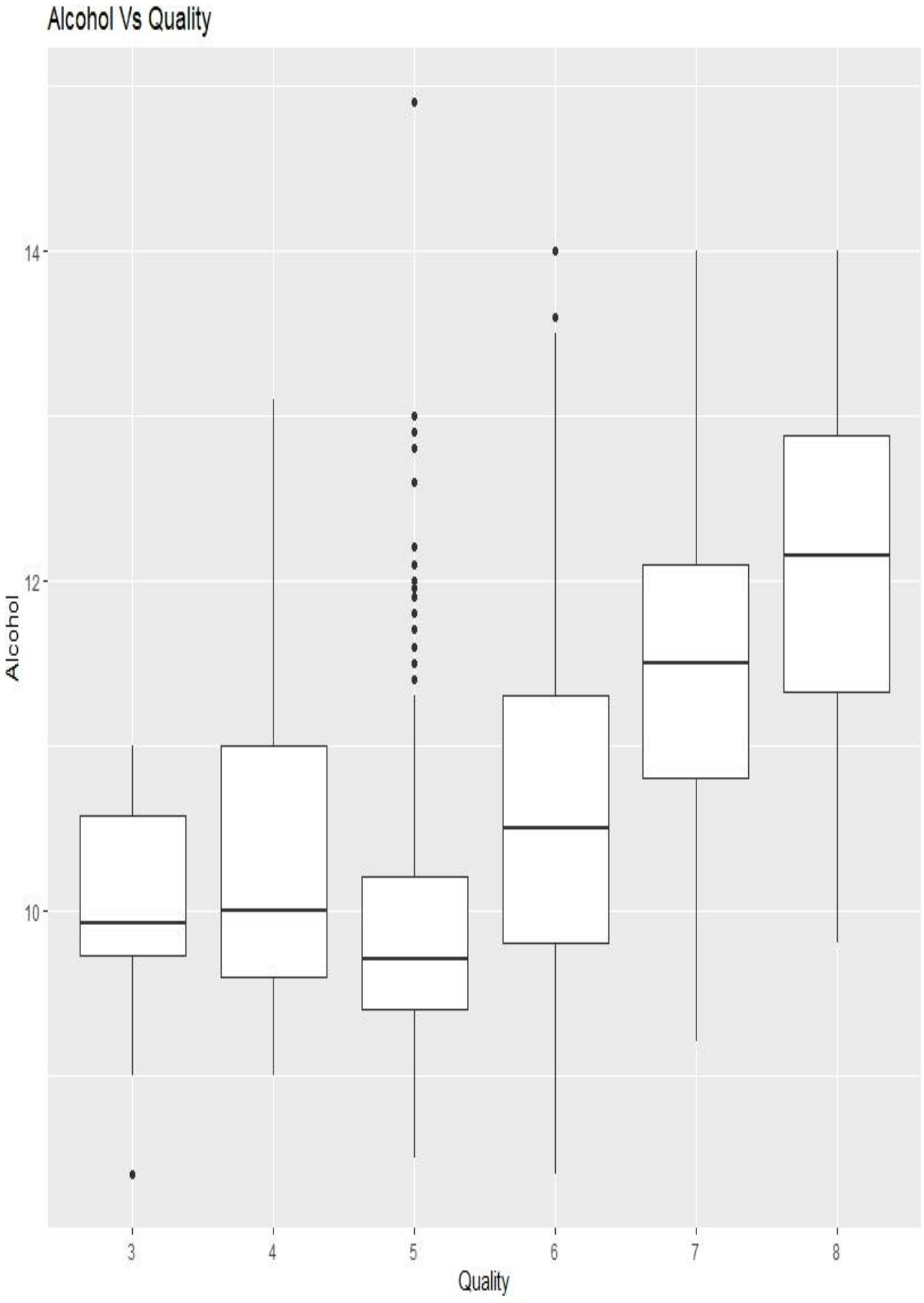

Fig 3

The box graph in fig 3 shows that higher grade quality wines have higher percentage of alcohol.

Let’s consider wine samples with quality grade 7 and above as good wine.

Let’s see what are the properties of the good wines (quality >= 7) .

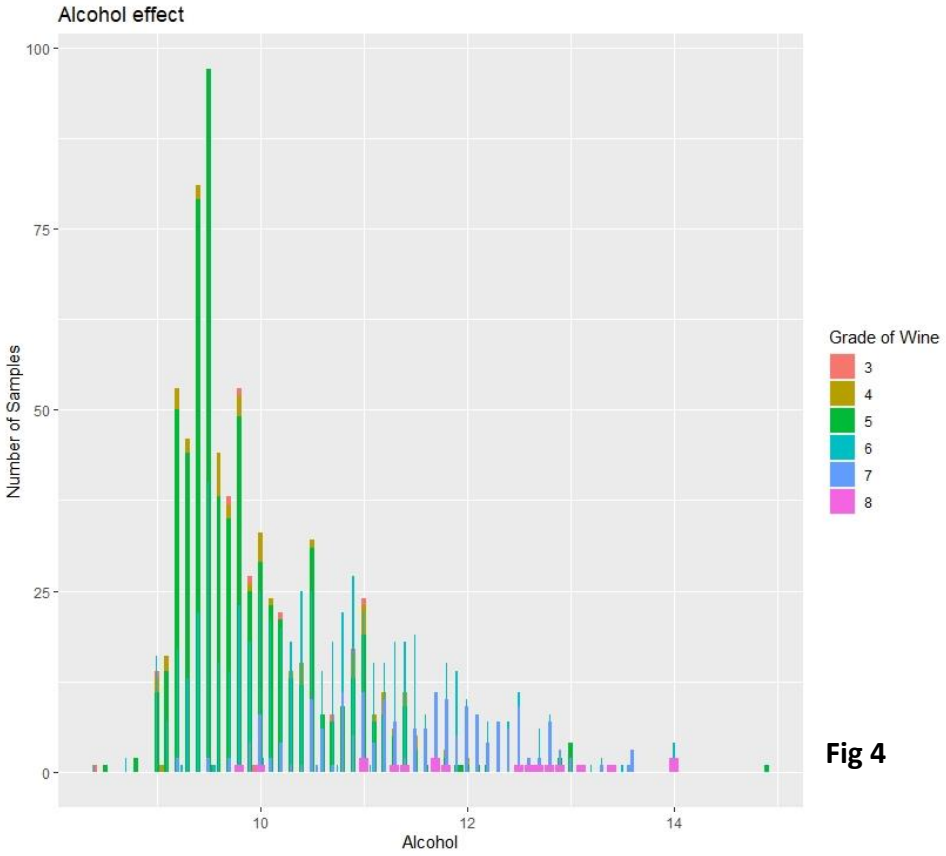

Fig 4

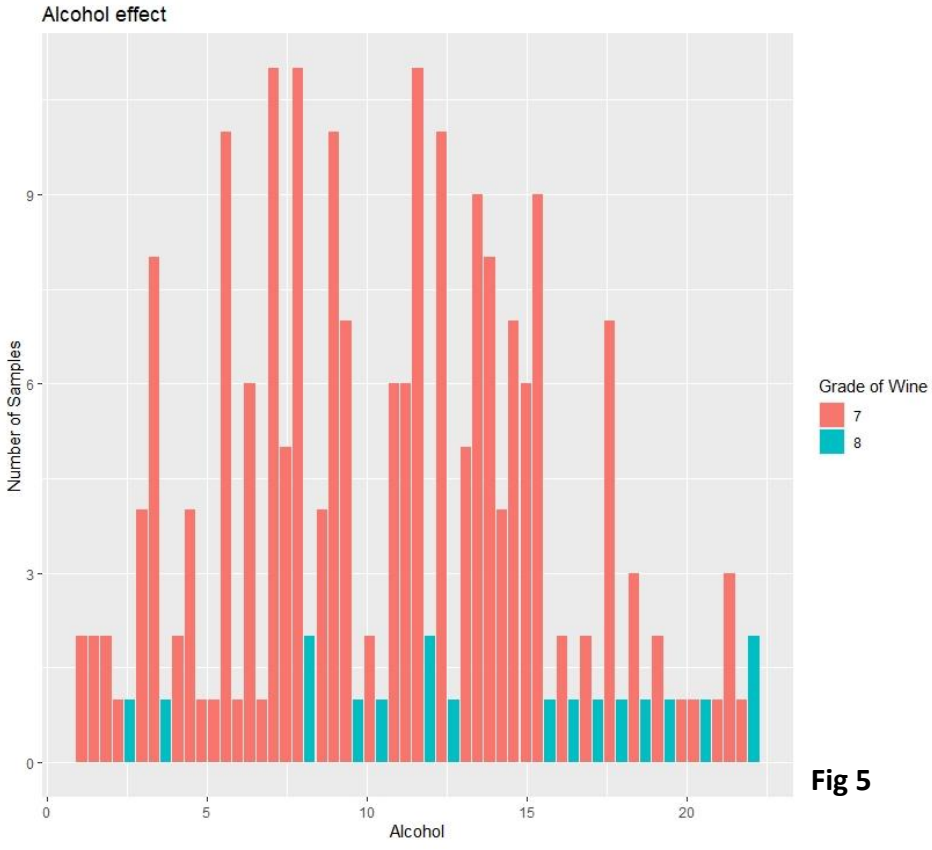

Fig 5

In the ‘Alcohol effect’ graphs in fig 4 and fig 5 we can clearly see the relationship between wine grade and percentage of alcohol value. Most of the wines with higher grade have higher alcohol percentage. As most of the 8 grade wines have percentage of alcohol 16+.

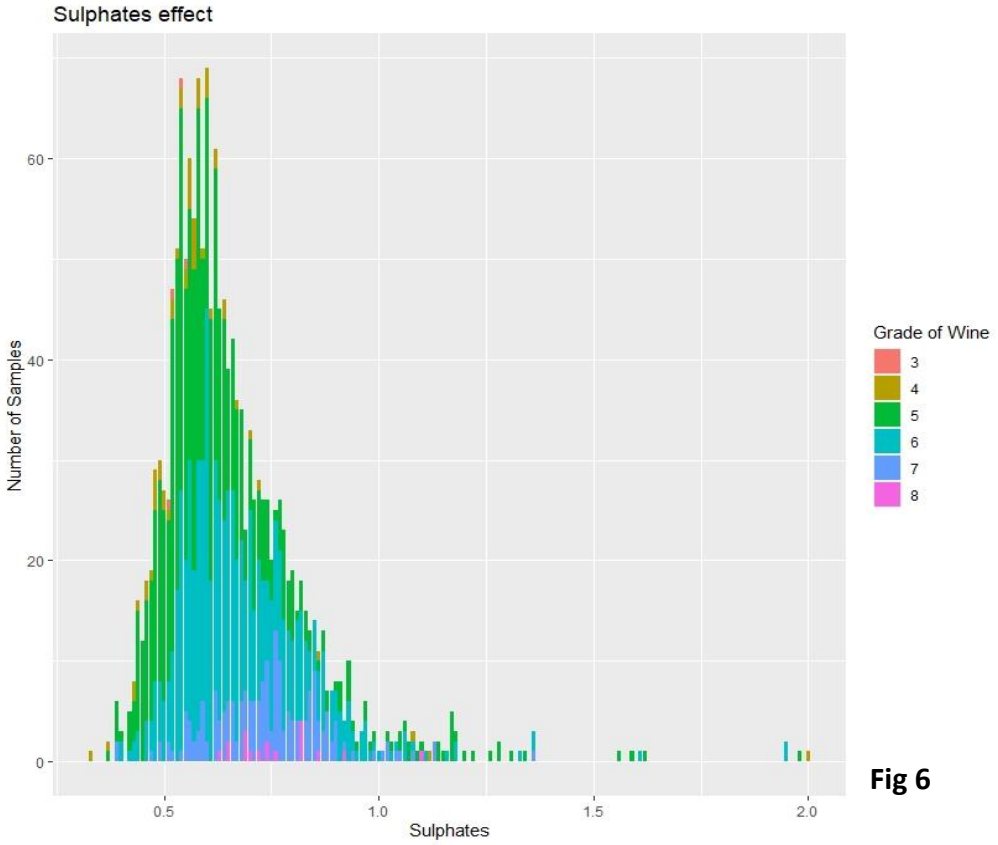

Fig 6

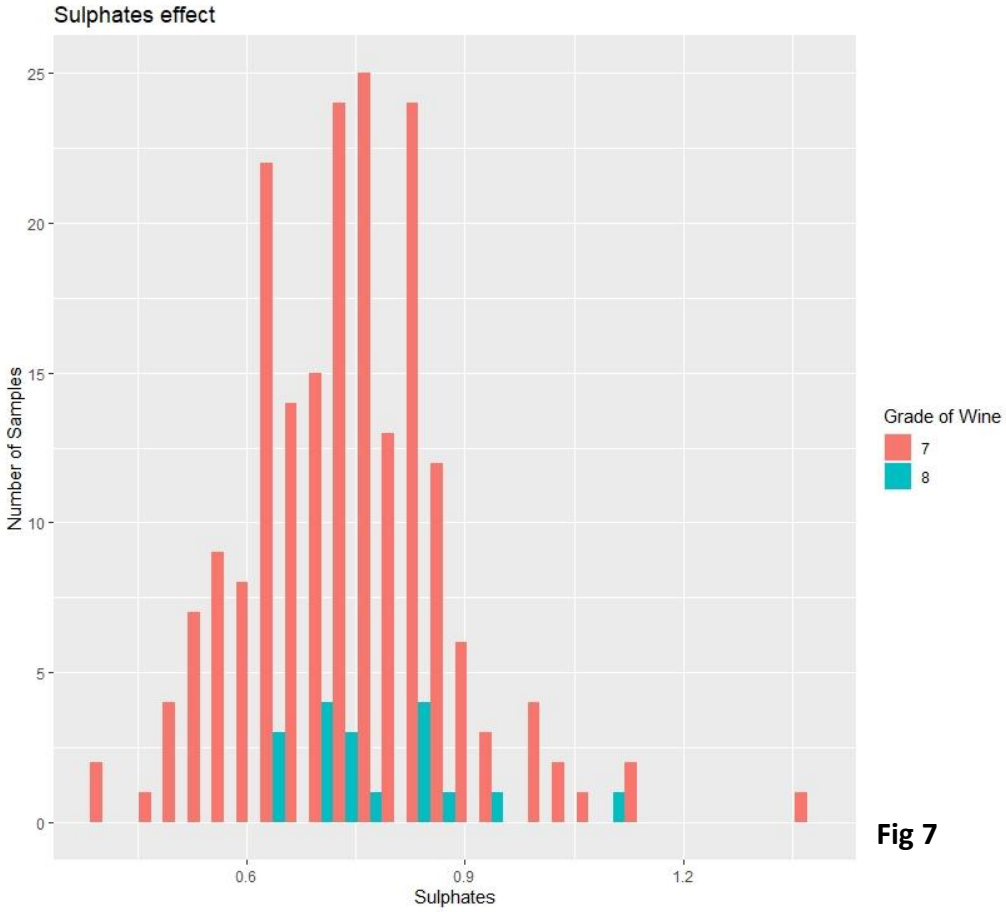

Fig 7

Comparing both graphs in fig 6 and 7 of ‘sulphates effect’ in fig we can see that for higher grade wines the value of sulphates is also higher.

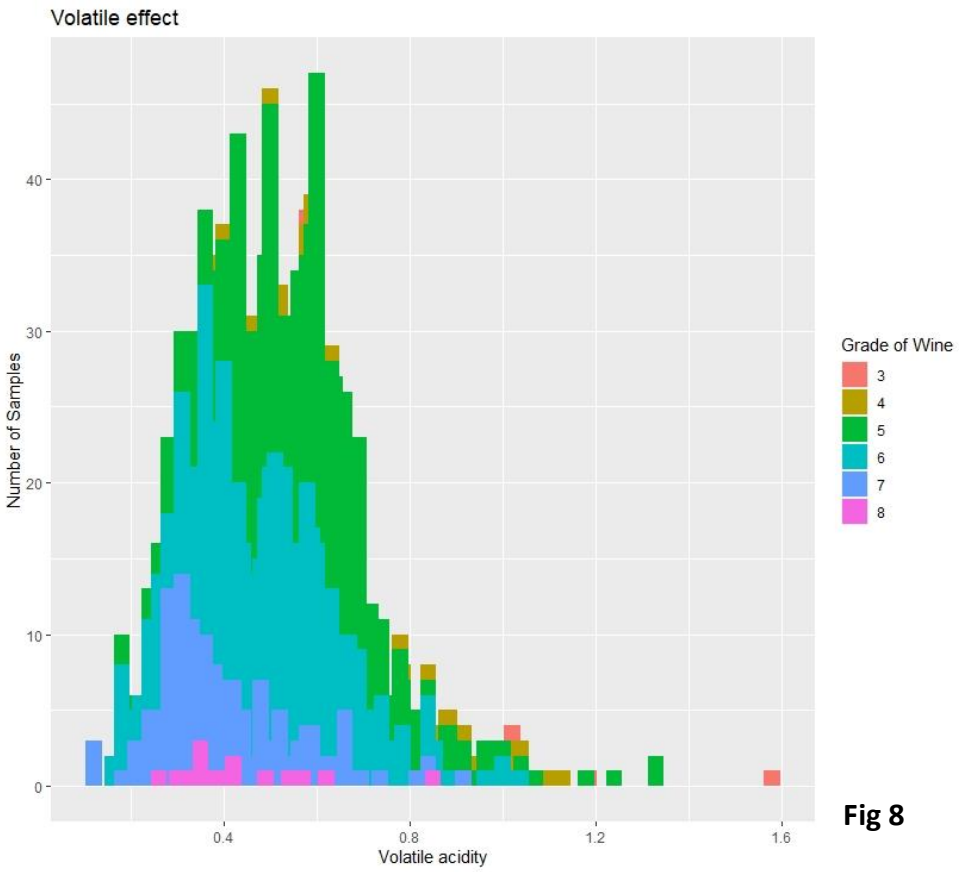

Fig 8

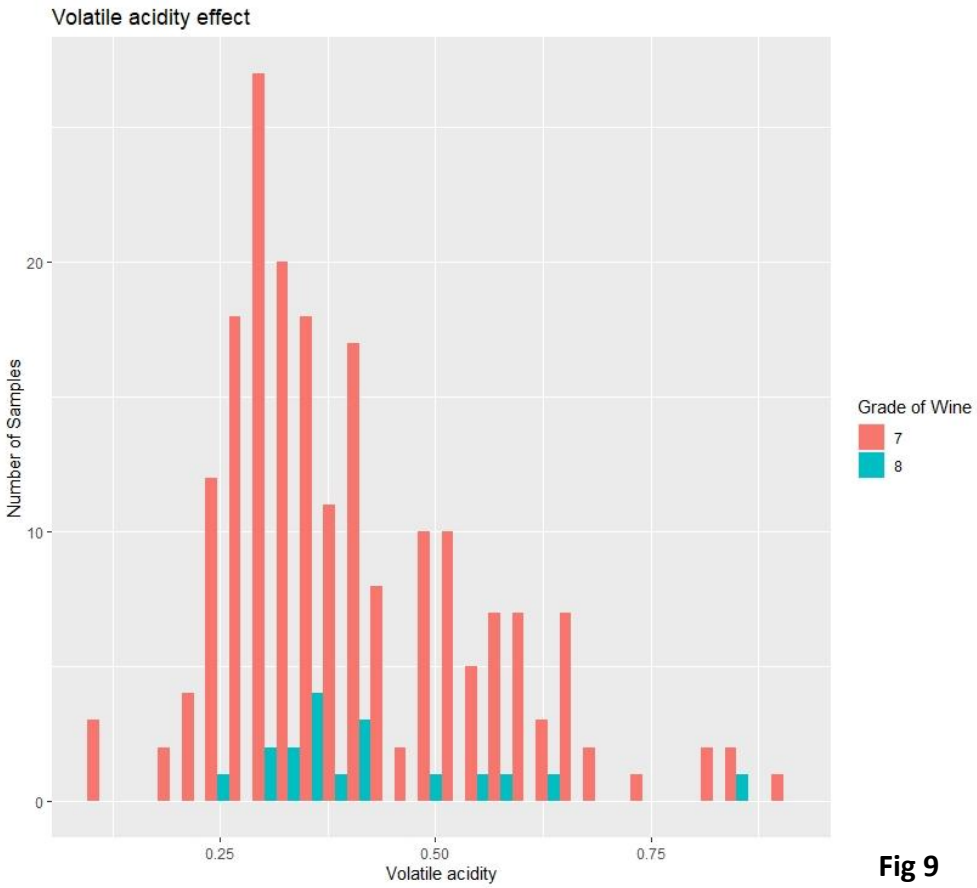

Fig 9

Here in ‘Volatile acidity effect’ graphs in fig 8 and 9 we can see that the higher grade wines have a lower volatile acidity value which satisfies the negative relationship result in table 1.

Regression Analysis:

Regression allows statistical models to quantitatively interpret patterns and determine if there is a likelihood that a relationship exists between variables (Newman, 2020). The initial models can be with a single variable like alcohol, sulphates or any other property in the dataset. These models can give an idea of how strong is the relation between the properties and the quality of the wine. There can be some complex models too, which have more than one variable or predictor to see dependence of the quality of the wine on its properties. We can find a nearly perfect model to predict the quality of the wine and with that we can also learn the important properties which have significant impact on the quality of the wine.

To create this model *'subset selection'* technique is applied. It is a tool which efficiently chooses the right predictors or variables for the model. There are 3 common forms of subset selection:

1. Best : In the 'Best subset selection' method, all the possible models are compared using set of predictors and spits the best fitting models which have 1 predictor, 2 predictor, and so on till the last iteration. There is only one draw-back for best subset selection that if the predictors are more than 40 then it becomes computationally heavy but fortunately, it is not our case.

2. Forward stepwise selection: In 'forward subset selection' method, the selection is started with empty model and predictors are added to each model as the iteration goes on. Obviously it will hurt the analysis accuracy a bit.

3. Backward stepwise selection: In the backward subset selection method the initial model has all the variables or predictors and then removes the least impacting predictor as it further iterates.

After applying the methods the models can be selected on the basis of Cp, BIC, and R2 or adjusted R2 value. The values of these estimators are not of any importance in themselves but these can be used in comparing the models to each other. This means we can see which model gives the minimum value of Cp and BIC. Lower the values of these estimators better the model. Larger the R^2 value better the model. For simple models we use simple R2 and for complex models we use adjusted R^2.

Let's see the plots for each method.

Best:

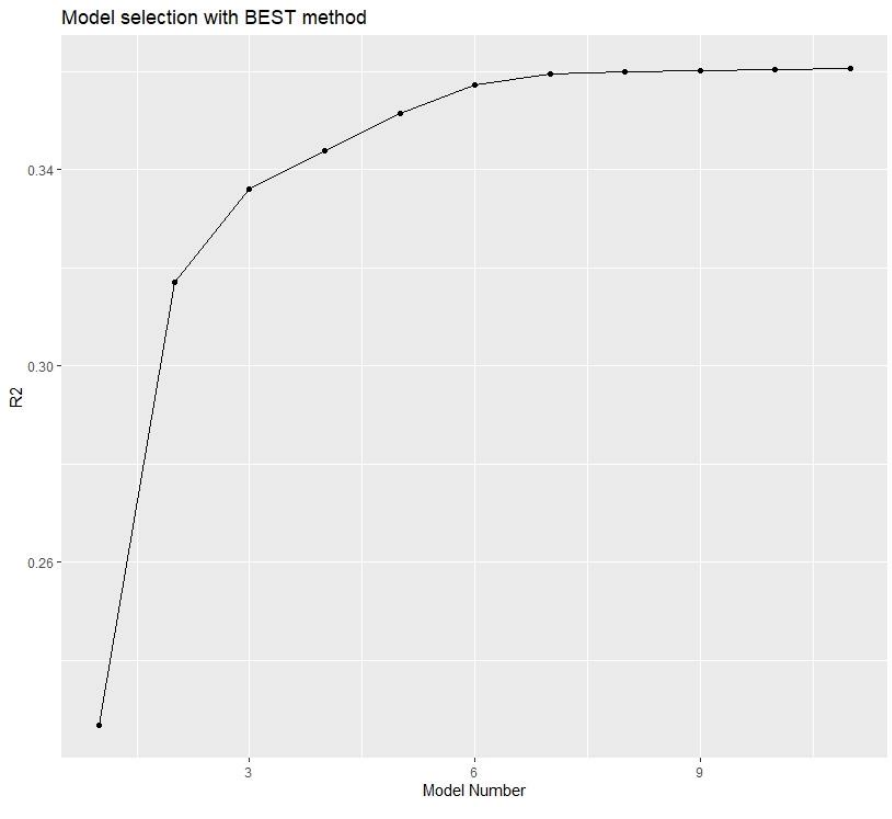

Fig 10

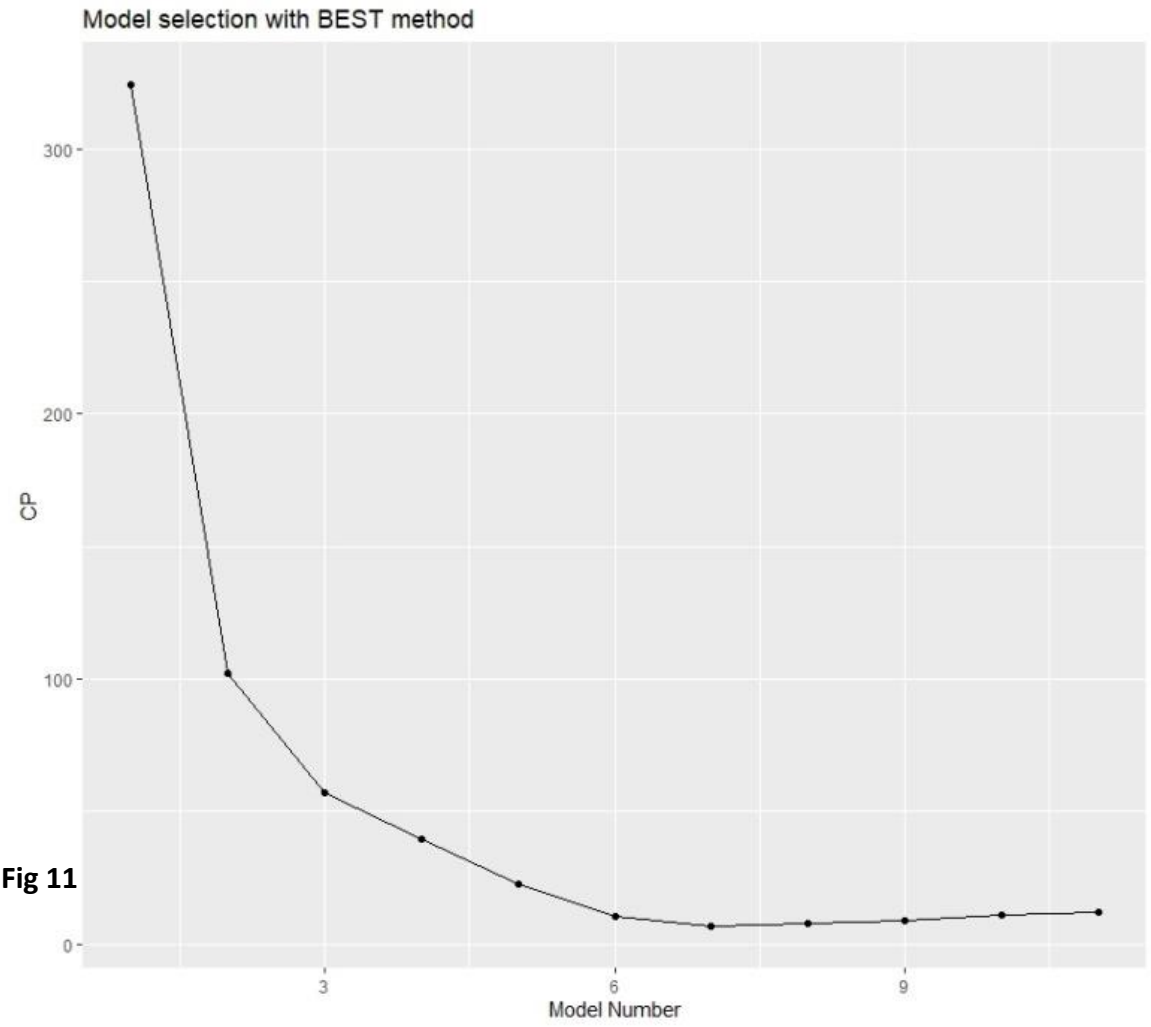
Model selection with BEST method
300
200
100
0
CP
3
6
9
Model Number

Fig 11

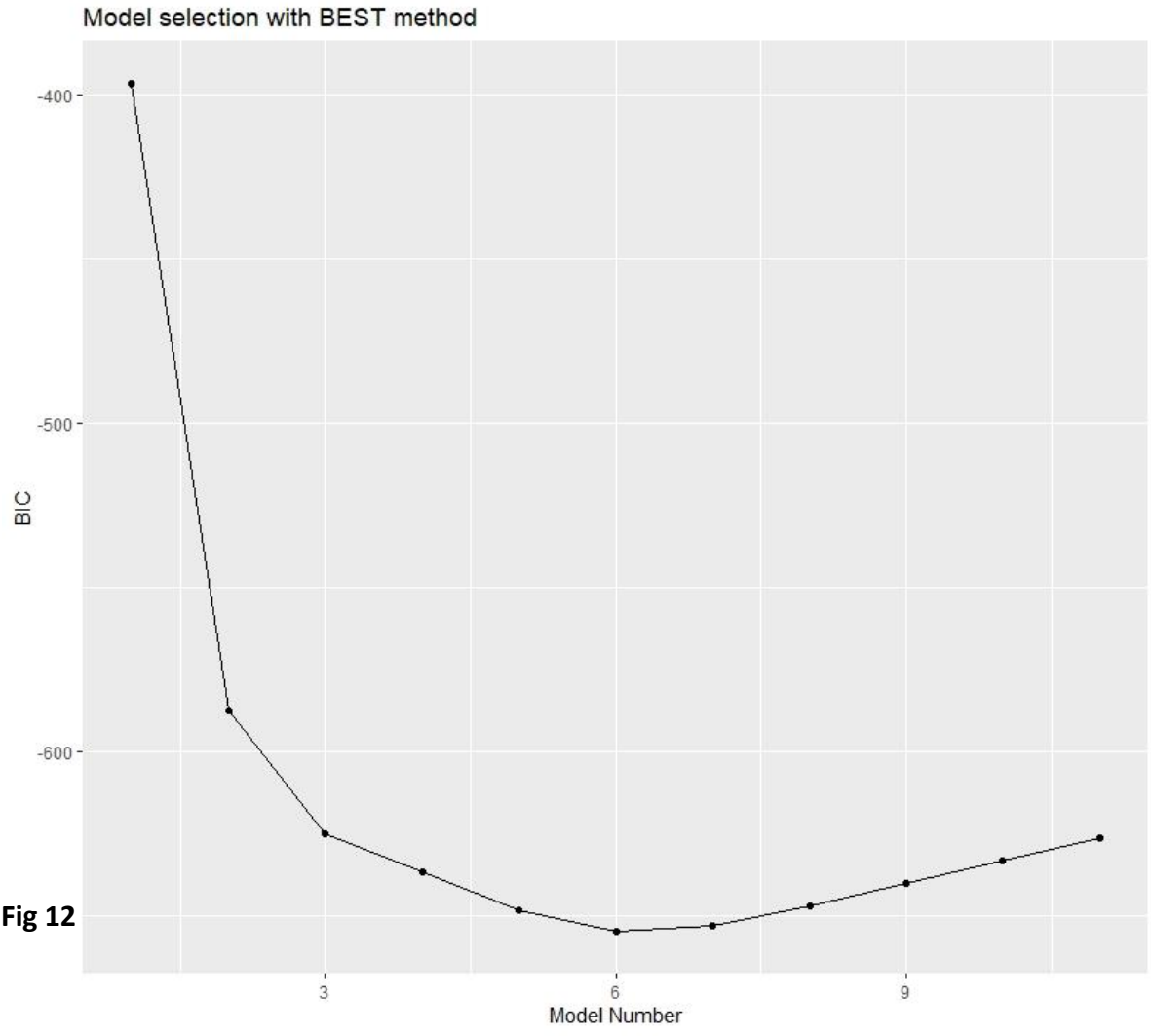
Model selection with BEST method
-400
-500
-600
BIC
3
6
9
Model Number

Fig 12

The graphs in fig 10-12 for *'Best'* method show that the R^2 value increases as the number of model increase which means that as the number of predictor increases the R^2 value also increases. But, the relative change in the R^2 value with increase in model number is significant till 6th model. The graph for Cp value shows that the 7^{th} model has the lowest Cp value and the graph for BIC shows that that the 6^{th} model has the lowest BIC value. Fig I in appendix shows the summary of the results of the Best fit model with all the variables.

Forward stepwise:

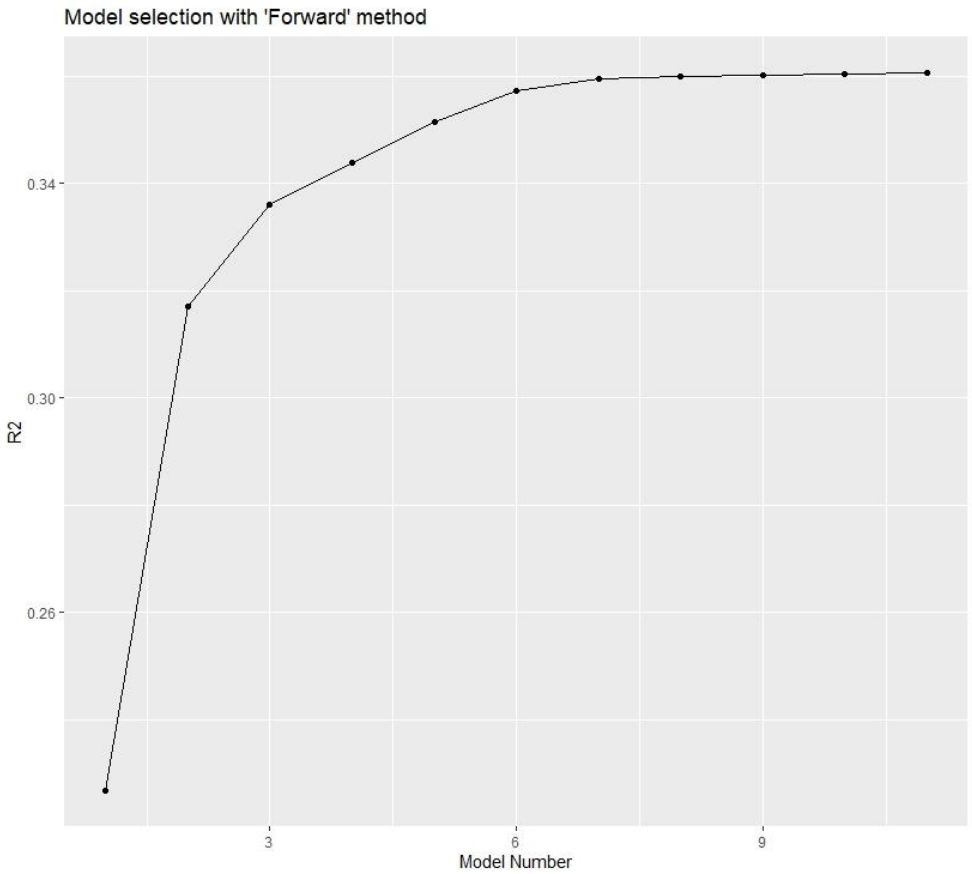

Fig 13

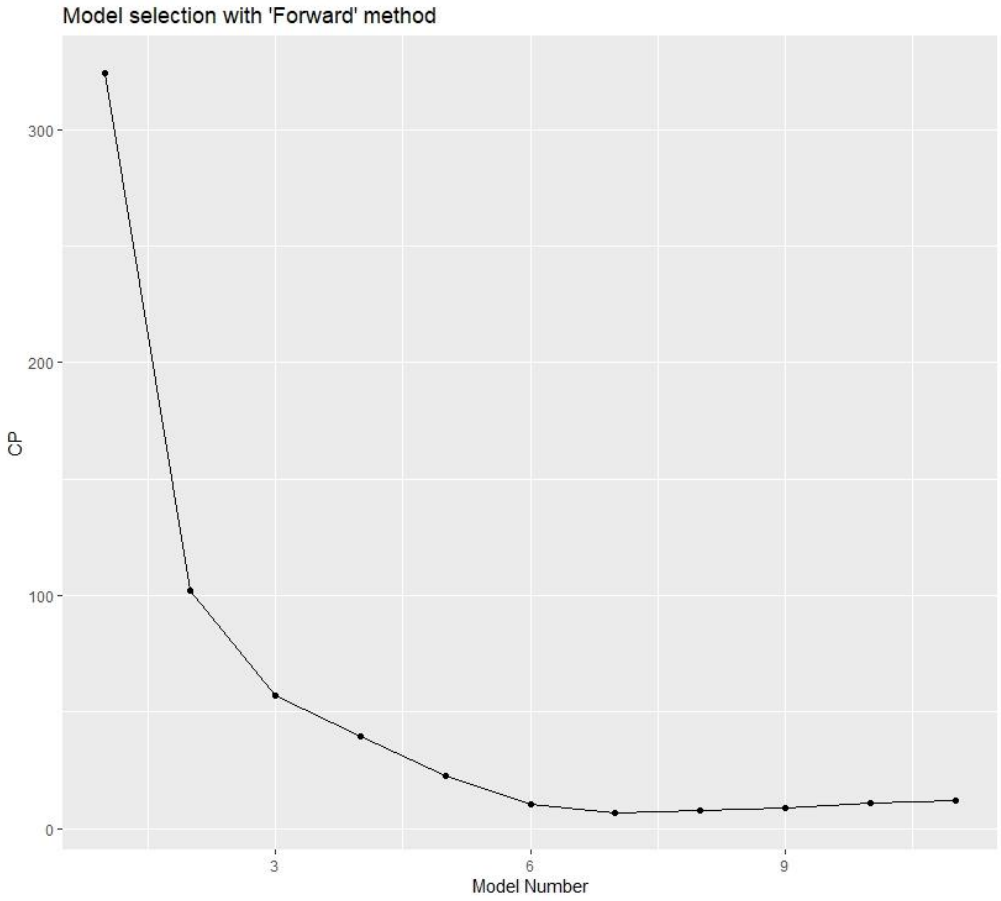

Fig 14

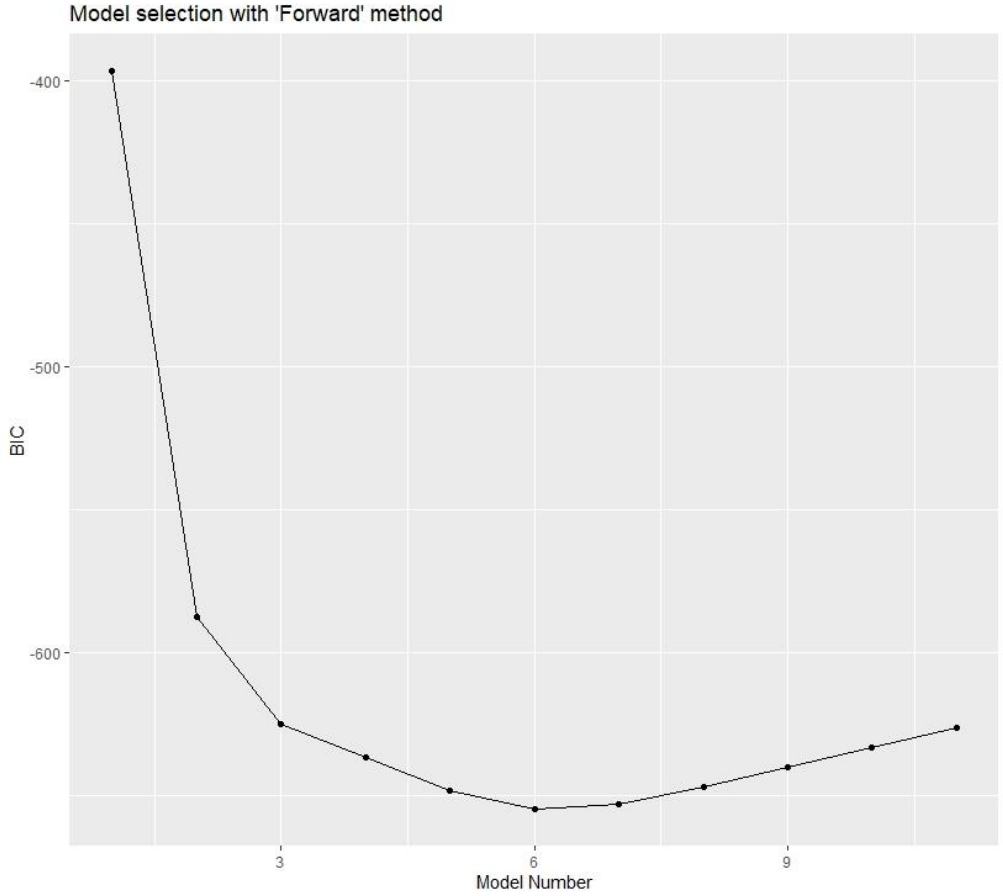

Fig 15

If we look at the 1st graph of the Forward stepwise method, in fig 13, we can we see that there is a significant change in R^2 till 6th model. The 2nd graph in fig 14 shows that 7th model has the lowest Cp number and the 3rd graph in fig 15 shows that 6th model has the lowest BIC value. Fig H in appendix shows the summary for the backward method of regression analysis.

Backward stepwise:

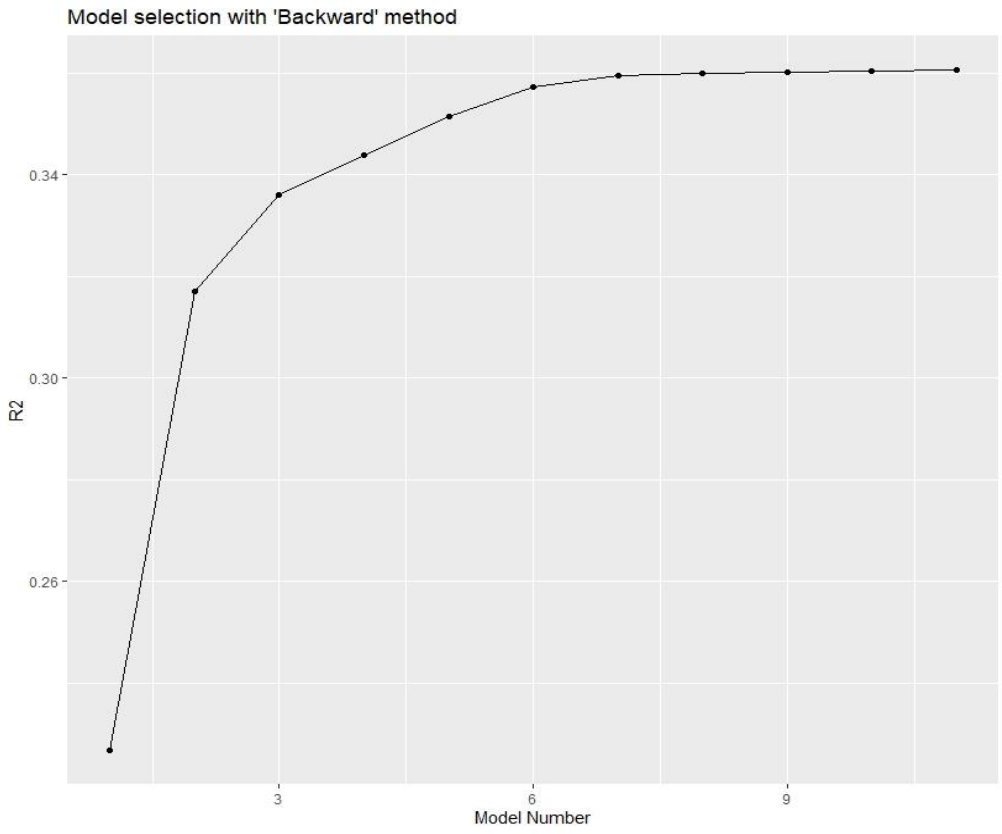

Fig 16

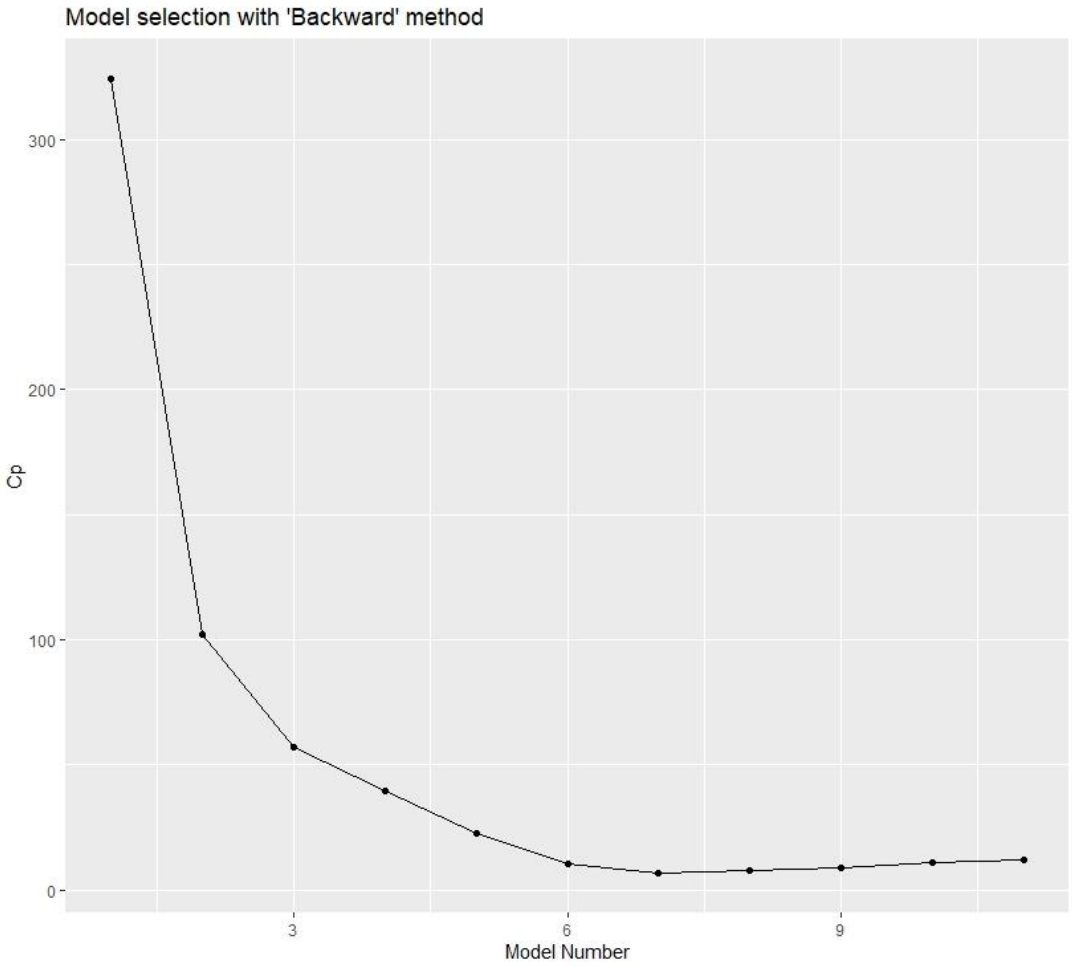
Model selection with 'Backward' method
300
200
100
0
Cp
3
6
9
Model Number

Fig 17

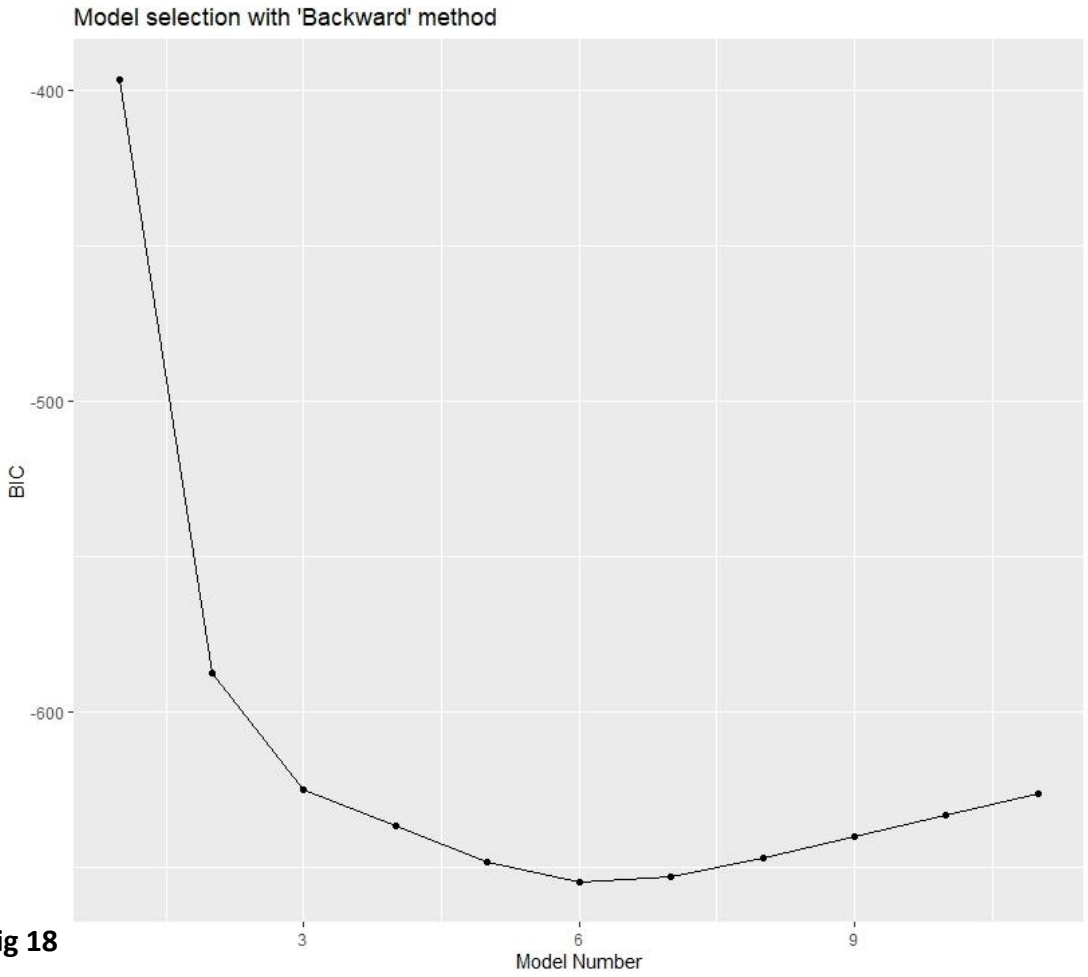
Model selection with 'Backward' method
-400
-500
-600
BIC
3
6
9
Model Number

Fig 18

The *'Backward stepwise'* method graphs show the same results as for the forward method.

These methods show that the 6th model is better as R^2 and BIC both support the model and it is less complex for evaluation and calculation.

Model #6
Quality~ Volatile acidity+ chlorides + total sulphur dioxide +pH +sulphates +alcohol

Table 2

To further enhance the R2 value we can use some more complex functions and see if we get some different results or more simple solutions. Table 3 show model result chosen after trying different complex functions.

Model selected
Quality~ Volatile acidity + chlorides+ poly(sulphates,5, raw = 'TRUE')+ log(alcohol)

Table3

For further verification of our model we can apply Cross Validation (CV) method check how better our model predicts for an unseen data. In cross validation the data is divided into 2 parts and the chosen model is then trained on one part and then tested. This way the model is test on a data that it has never seen before and the results describe how good the model is. The data set is split in K equal parts and then 1/K part is left for testing and on the rest of the data the model is trained. It can be risky as the data is reduced for the training of the model so the training set is changed K times and it is called K-Fold Cross Validation (KFCV). The data set is also shuffled up so that the model can be trained on different combinations of the data point. This technique eliminates the chances of over-fitting a model to its data. We will use 5 FCV in our analysis.

The models shown in table 4 are chosen Cross Validation.

Complexity	Models
1	Quality~ Volatile acidity + chlorides+ sulphates + alcohol
2	Quality~ Volatile acidity + chlorides+ sulphates + alcohol + pH
3	Quality~ Volatile acidity + chlorides+ poly(sulphates,5, raw = 'TRUE')+ log(alcohol)
4	Quality~ Volatile acidity + chlorides+ log(alcohol) + poly(sulphates,2) + poly(pH,2)
5	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,3) + poly(pH,3)
6	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,4) + poly(pH,4)
7	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,4) + poly(pH,4)+ total sulfur dioxide
8	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,4) + poly(pH,4)+poly(total sulfur dioxide,2)
9	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,4) + poly(pH,4)+ poly(total sulfur dioxide,3)
10	Quality~ Volatile acidity + chlorides + log(alcohol) + poly(sulphates,4) + poly(pH,4)+poly(total sulfur dioxide,4)

Table 4

The result of Cross Validation is shown in figure19. The models in table 4 are plotted with their error and error bars. The x-axis of the graph shows the complexity of the model and y-axis shows the error. The vertical lines in the graph show the standard error and horizontal lines on top and bottom of these vertical lines are the upper and lower limit of the error. An error bar is an important tool that symbolizes one standard error from the mean to statistically communicate variability in the data (Newman, 2020). To choose the best model is bit tricky part as the best model should be the simplest as possible and should have the lowest error too. So we have to balance both complexity and

error in choosing the best model. In figure 19 we can see that the 10th model has the lowest error but it's the most complex model. When we consider complexity the model number 1 is the simplest model but it has the highest error. So if we have critical look on the graph we can see that 4th model seems to be the best fit model with balanced complexity and error. The error bars of model 4 fall in limits of more complex models so this means that it's better than those models.

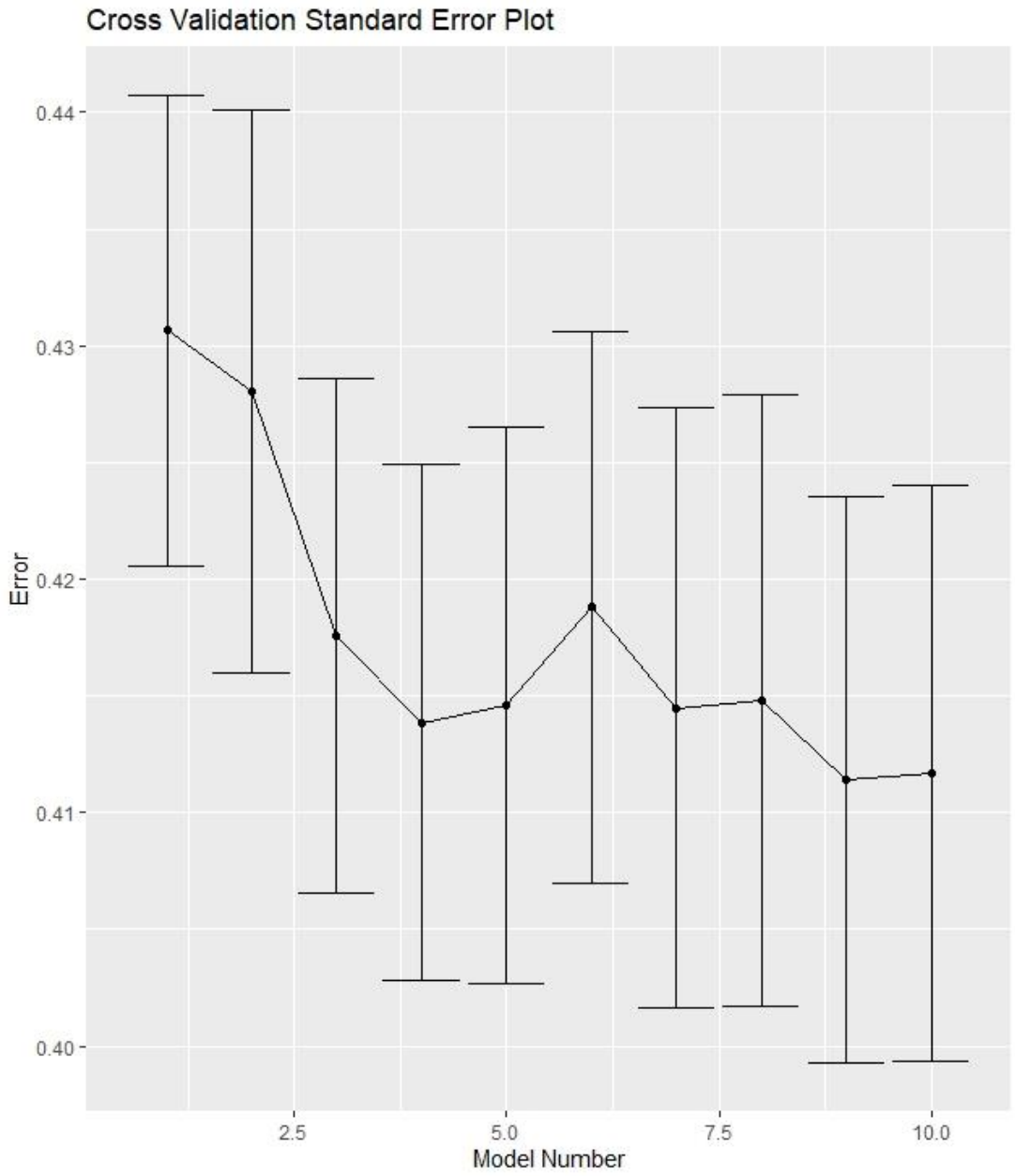

Fig 19

Here we should remember that the test set is being chosen randomly so it means that if we try to execute our code another time, different result may appear. Fig 20 shows the result of another execution of the code. Some other solutions are in appendix (fig B and C).

The fig 20 also supports the result from the previous graph. Graph in fig[] shows that 4th model is a better fit as it has low error rate and its standard error bars are in the limit of more complex models. Hence, 4th model is to be chosen.

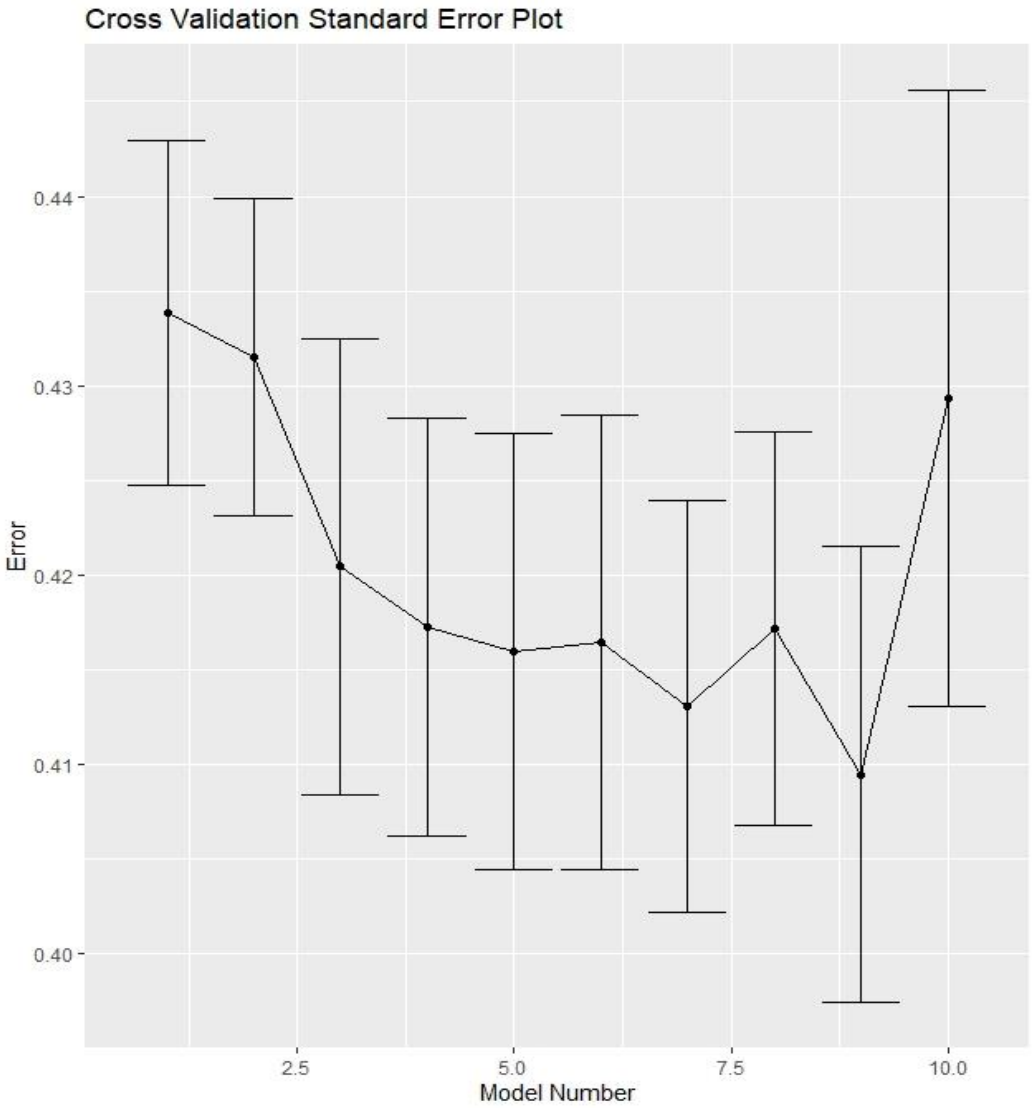

Fig 20

To judge if the relationship between predictor and response is significant or not we look upon the p-value. If the p-value is below 0.05 then it means that the relationship is statistically significant. As the p-value approaches 0 value the more significant the relationship gets. If the p-value is greater than 0.05 then it means that there is no significant relationship in variables and response. In fig 21 we can see that except pH^2 every variable has a significant relationship with quality as these variable have p-value nearly 0. pH^2 having p-value greater than does not harm the model but yes I definitely means that it has no significance in the model. Leaving it in the model would not disturb anything.

The second important thing in the fig 21 is the R-squared value. This value is a judge of how good our model is. The R-squared value varies from 0 to 1. Higher the R-squared value better the model. The threshold value of R-squared varies with the problem a model is solving. If a model is on something related to business world then minimum of 0.6 R-squared is desired, but in our case it is 0.3677 which is fine.

Fig 21

```
Call:
lm(formula = quality ~ volatile_acidity + chlorides + poly(sulphates,
    2) + log(alcohol) + poly(pH, 2), data = winequality_red)

Residuals:
     Min       1Q   Median       3Q      Max
-2.73448 -0.37442 -0.04747  0.46536  1.96832

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.2903     0.4388  -2.940  0.00333 **
volatile_acidity      -0.9175     0.1014  -9.048  < 2e-16 ***
chlorides             -1.5498     0.3971  -3.903 9.89e-05 ***
poly(sulphates, 2)1    5.8692     0.7487   7.839 8.23e-15 ***
poly(sulphates, 2)2   -4.7272     0.6873  -6.878 8.68e-12 ***
log(alcohol)           3.2262     0.1788  18.046  < 2e-16 ***
poly(pH, 2)1          -3.2614     0.7158  -4.557 5.60e-06 ***
poly(pH, 2)2          -1.0438     0.6777  -1.540  0.12369
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6422 on 1591 degrees of freedom
Multiple R-squared:  0.3705,    Adjusted R-squared:  0.3677
F-statistic: 133.8 on 7 and 1591 DF,  p-value: < 2.2e-16
```

Predicting Quality of Wine:

Let's see if we can use this data set to predict the quality of the wine using the physicochemical properties. To make a model which can make a guess of an out-come on showing a new data, we use *'Classification'*. In classification method we divide categories into classes according to the values of the variable. For this division we first select some variables by looking at graphs of different combinations of variables. After finalizing the variables we make a function of the variable and response and take a part of the data set to train the model on it. Before dividing the data set into training and testing part, we do shuffle up the rows in dataset to improve the results. After this we look at the '*confusion matrix*' which illustrates strength of the model.

Note: With this technique we can choose some of the properties of wine to predict the quality. This may be not a strong technique as every property is important. For example if we only look at the quantity of 2 of the 11 ingredients in a recipe, there is a high chance that the food may not taste well.

Trying different models which include different variables give different results. The table 5 below shows the number of samples in each category of the wine grade.

Quality grade	Samples
3	10
4	53
5	681
6	638
7	199
8	18
Total	1599

Table 5

Let's see different models to classify the wine samples. These models were selected after visualizing the graphs of pairs of variables and after that making them complex. The graphs of pairs can be seen in the appendix (Fig D).

#	Model	Result
1	Quality ~ fixed acidity + volatile acidity	791/1599=0.495 49.5% right
2	Quality ~ alcohol + sulphates + volatile acidity	918/1599=0.574 57.4% right
3	Quality ~ alcohol + sulphates + volatile acidity + fixed acidity	917/1599=0.573 57.3% right
4	Quality ~ alcohol + poly(sulphates,2,raw = 'TRUE') + volatile acidity	926/1599=0.579 57.9% right
5	quality~ alcohol + poly(sulphates,2,raw = 'TRUE') + volatile acidity + fixed acidity	928/1599=0.580 58% right
6	Quality ~ alcohol + poly(sulphates,5,raw = 'TRUE') + volatile acidity + fixed acidity	937/1599=0.586 58.6% right
7	Quality ~ alcohol + poly(sulphates,5,raw = 'TRUE') + volatile acidity + fixed acidity +pH	938/1599=0.586 58.6% right
8	Quality ~ 'all properties'	965/1599=0.6 60% right

Table 6

Confusion matrix:

	3	4	5	6	7	8
3	2	1	5	1	0	0
4	0	0	0	0	0	0
5	7	40	432	279	44	4
6	1	12	244	355	153	14
7	0	0	0	3	2	0
8	0	0	0	0	0	0

Matrix Model 1in table 6

	3	4	5	6	7	8
3	2	2	4	0	0	0
4	0	1	0	0	0	0
5	7	39	512	243	15	0
6	1	10	158	339	120	9
7	0	1	6	56	64	9
8	0	0	1	0	0	0

Matrix Model 2 in table 6

	3	4	5	6	7	8
3	2	2	5	1	0	0
4	0	0	0	0	0	0
5	7	39	511	247	14	0
6	1	12	159	331	112	8
7	0	0	6	59	73	10
8	0	0	0	0	0	0

Matrix Model 3 in table 6

	3	4	5	6	7	8
3	2	2	4	0	0	0
4	0	2	1	3	0	0
5	7	40	516	244	13	0
6	1	8	155	341	121	9
7	0	1	4	50	65	9
8	0	0	1	0	0	0

Matrix Model 4 in table 6

	3	4	5	6	7	8
3	2	1	5	1	0	0
4	0	2	1	3	0	0
5	7	40	516	241	11	0
6	1	10	153	337	117	8
7	0	0	6	56	71	10
8	0	0	0	0	0	0

Matrix Model 5 in table 6

	3	4	5	6	7	8
3	2	2	4	0	0	0
4	0	4	5	7	0	0
5	7	36	519	239	9	1
6	1	11	147	334	112	7
7	0	0	6	58	78	10
8	0	0	0	0	0	0

Matrix Model 6 in table 6

	3	4	5	6	7	8
3	3	2	4	0	0	0
4	0	3	5	8	0	0
5	6	36	519	238	9	1
6	1	12	147	334	111	6
7	0	0	6	58	79	11
8	0	0	0	0	0	0

Matrix Model 7 in table 6

	3	4	5	6	7	8
3	2	2	3	1	0	0
4	0	3	5	3	0	0
5	7	27	507	204	11	0
6	1	19	158	371	106	10
7	0	2	8	59	82	8
8	0	0	0	0	0	0

Matrix Model 8 in table 6

How to read these matrixes? Let's take the matrix for 8th model. Starting from the left side under the column of grade 3, it shows that the model made 2 guesses right about the wine with grade 3 and 7 times it the wine was grade 5 when it was of grade 3 and once it guessed it as a grade 6 wine but it was grade 3. So if we tally the diagonal numbers, they tell how many times the model guessed the wine right.

More variables we use higher the probability gets of making a right guess.

Let's use half of the data set as training set and train our models on it and then use the rest of the data as a new data and see how accurately the models guess the quality. As our data has 1500 rows so training set is of 800 rows and test set ids of 799. Table 7 shows the results.

Model	Result
1	53%
2	58%
3	55%
4	57%
5	59%
6	60%
7	59%
8	59%

Table 7

As some models have results too close so let's apply cross validation on these models and see which model will suit best.

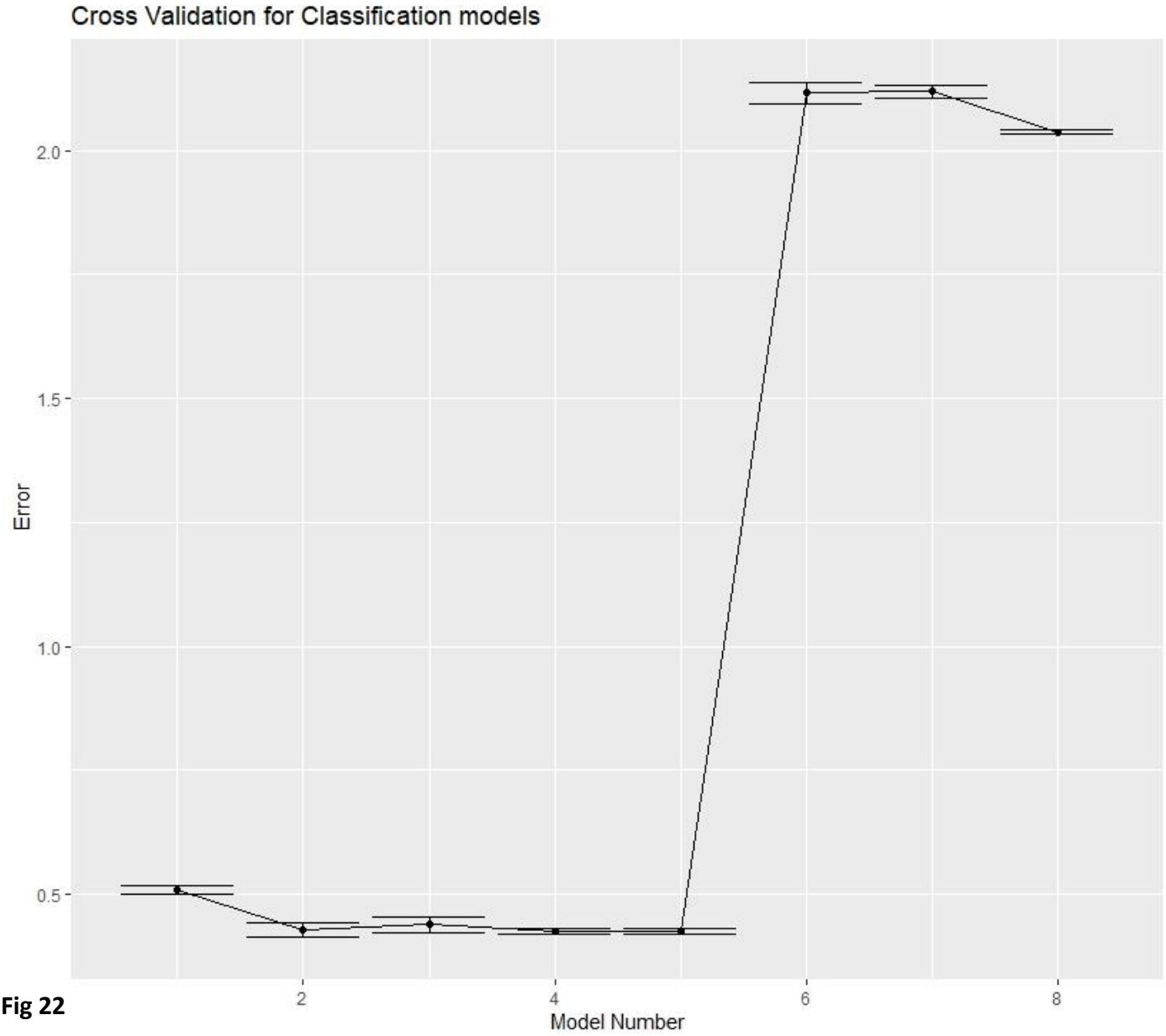

Fig 22

Fig 22 shows the results of cross validation application on the models. Using different parts of data set as training dataset and remaining parts as test datasets the models were used to predict the quality of the wine. The results show that 2nd model seems to be the best fit as it is simple and has less error. The results shown in the 'cross validation for classification' graph means that if we have values for alcohol, volatile acidity, and sulfates we can

predict the quality of the wine sample. In real life we can't do so as if we have, for example garlic, lemon, salt, pepper and meat in a dish, we can't decide the taste of the dish on basis of any 2 of 5 ingredients. If there is no salt or if it is in heavy amount both cases the taste will be compromised. So we may not use our model for real life.

Conclusion:

The insight of all the data tells us that there is a high correlation between percentage of alcohol and the quality of wine. The data also tells that the relation between some of the properties also affect each other like density and percentage of alcohol, and fixed acidity.
The dataset has very less samples of wines which have grade of quality above 6 and most of the samples fall in 5 or 6 categories. Another thing to note is that the quality is decided in the personal tastes which can be affected by some other things too like year of manufacturing, the origin of grapes or the quality of the grapes used in manufacturing.
Further analysis can be done if there equal number of wines in each category.
The exploration of the data set tells that quality of the red wine is dependent on alcohol as adding alcohol to any model above boosts the results as compared to other variables but, the correlation is only 0.4 which is not that high to take it as an important predictor of wine quality. The important variables as per models for the quality of wine are level of acidity, amount of sulfates and percentage of alcohol in the wine. The further experimentation can be done taking volatile acidity under consideration as the relationship seems opposite but it can be further clarified.

Appendix

Sites used for help:

https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

https://winefolly.com/deep-dive/vinho-verde-the-perfect-poolside-wine-from-portugal/

https://rstudio-pubs-static.s3.amazonaws.com/57835_c4ace81da9dc45438ad0c286bcbb4224.html

Citation

MasterClass. "What Is Vinho Verde? Understanding the Many Wines From Portugal's Prominent Wine Region - 2021." *MasterClass*, MasterClass, 8 Nov. 2020, www.masterclass.com/articles/what-is-vinho-verde.

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
1	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5
2	7.8	0.880	0.00	2.60	0.098	25	67	0.99680	3.20	0.68	9.8	5
3	7.8	0.760	0.04	2.30	0.092	15	54	0.99700	3.26	0.65	9.8	5
4	11.2	0.280	0.56	1.90	0.075	17	60	0.99800	3.16	0.58	9.8	6
5	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5
6	7.4	0.660	0.00	1.80	0.075	13	40	0.99780	3.51	0.56	9.4	5
7	7.9	0.600	0.06	1.60	0.069	15	59	0.99640	3.30	0.46	9.4	5
8	7.3	0.650	0.00	1.20	0.065	15	21	0.99460	3.39	0.47	10.0	7
9	7.8	0.580	0.02	2.00	0.073	9	18	0.99680	3.36	0.57	9.5	7
10	7.5	0.500	0.36	6.10	0.071	17	102	0.99780	3.35	0.80	10.5	5
11	6.7	0.580	0.08	1.80	0.097	15	65	0.99590	3.28	0.54	9.2	5
12	7.5	0.500	0.36	6.10	0.071	17	102	0.99780	3.35	0.80	10.5	5
13	5.6	0.615	0.00	1.60	0.089	16	59	0.99430	3.58	0.52	9.9	5
14	7.8	0.610	0.29	1.60	0.114	9	29	0.99740	3.26	1.56	9.1	5
15	8.9	0.620	0.18	3.80	0.176	52	145	0.99860	3.16	0.88	9.2	5
16	8.9	0.620	0.19	3.90	0.170	51	148	0.99860	3.17	0.93	9.2	5
17	8.5	0.280	0.56	1.80	0.092	35	103	0.99690	3.30	0.75	10.5	7
18	8.1	0.560	0.28	1.70	0.368	16	56	0.99680	3.11	1.28	9.3	5
19	7.4	0.590	0.08	4.40	0.086	6	29	0.99740	3.38	0.50	9.0	4
20	7.9	0.320	0.51	1.80	0.341	17	56	0.99690	3.04	1.08	9.2	6
21	8.9	0.220	0.48	1.80	0.077	29	60	0.99680	3.39	0.53	9.4	6
22	7.6	0.390	0.31	2.30	0.082	23	71	0.99820	3.52	0.65	9.7	5
23	7.9	0.430	0.21	1.60	0.106	10	37	0.99660	3.17	0.91	9.5	5

Fig A

Fig A shows a screenshot of the data set used.

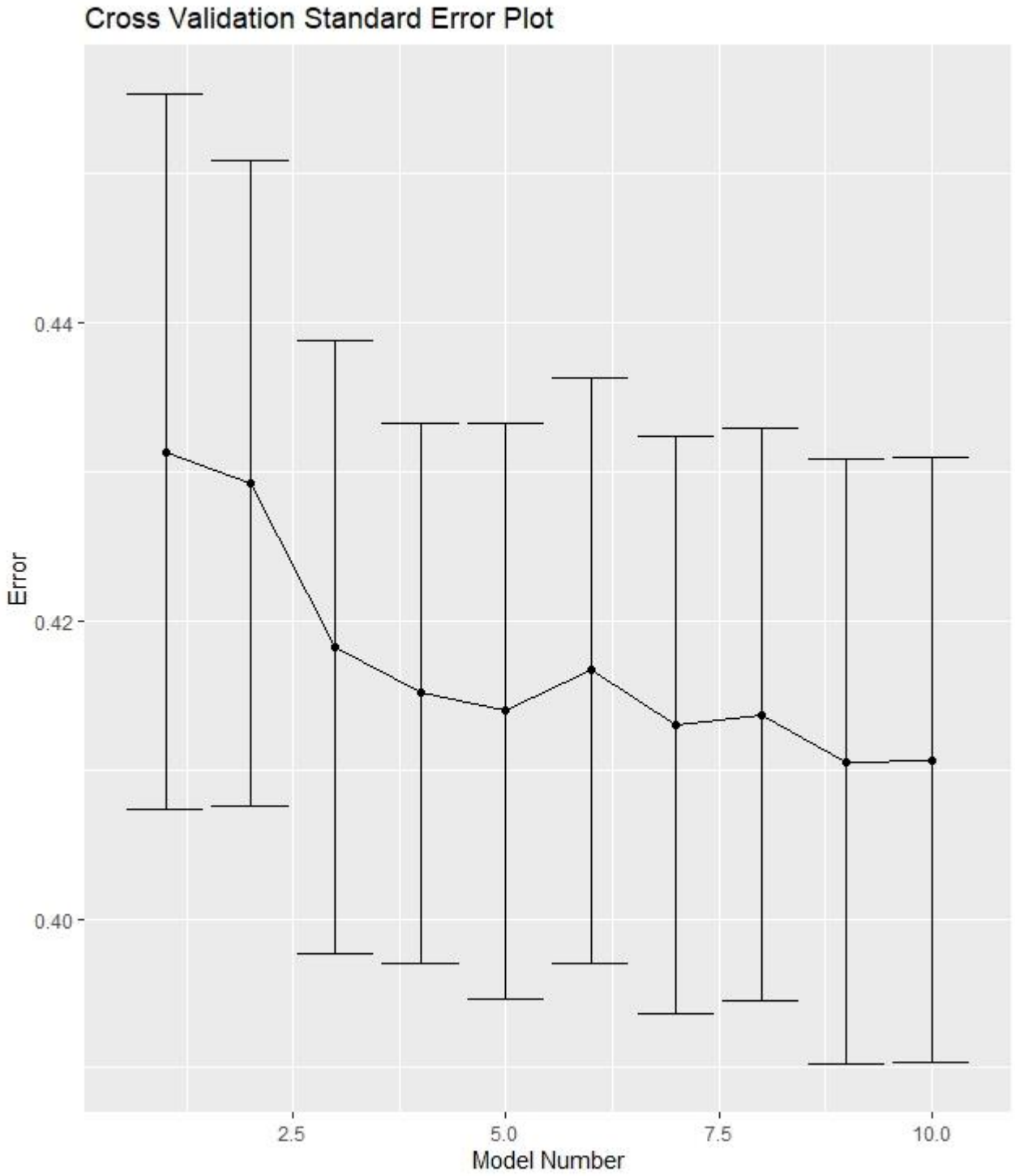

Fig B

Fig B and C show the different trials for results of Cross validation for regression models. As when we shuffle the rows in dataset different part of data set gets to be the training and testing data set.

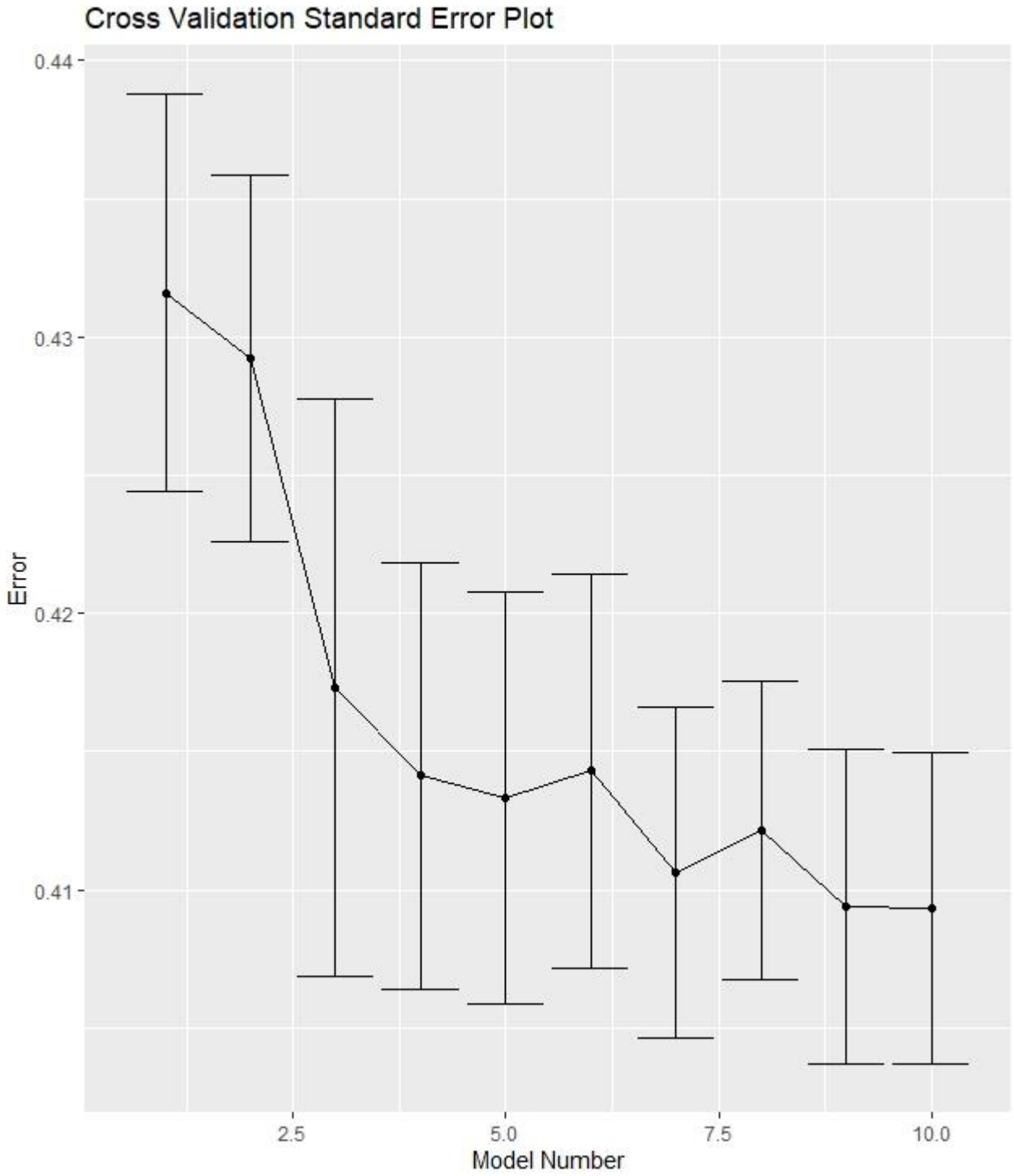
Cross Validation Standard Error Plot
0.44
0.43
0.42
0.41
Error
2.5
5.0
7.5
10.0
Model Number

Fig C

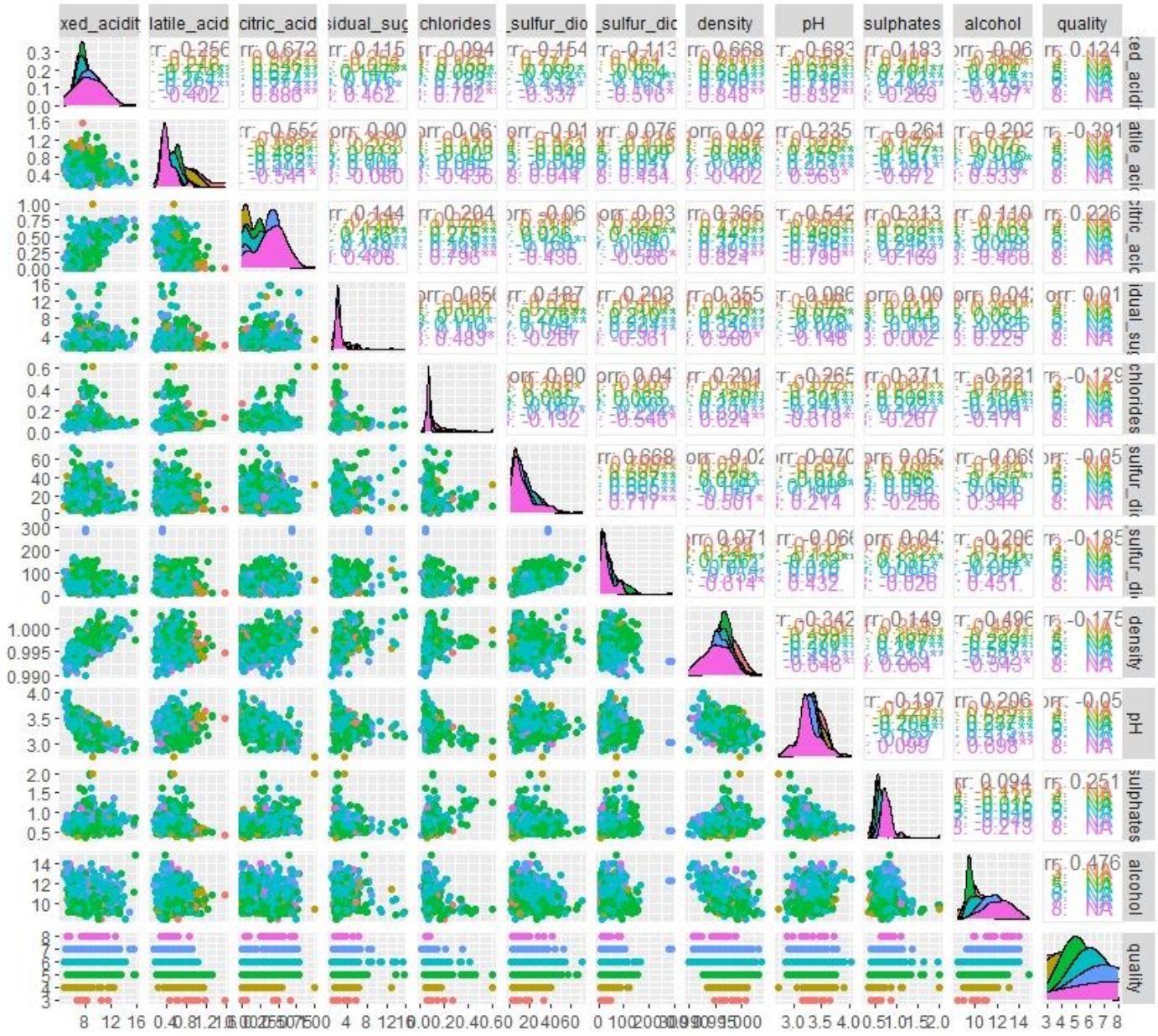

Fig D

Fig D shows the graphs of the different pairs of variables in the data set. It was used to determine the variables for the models of '*Classification'.*

Following are some screenshots of trying different models to see the relationship of different variables with the quality of the wine.

```
Call:
lm(formula = quality ~ alcohol + sulphates, data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6685 -0.3781 -0.1005  0.4992  2.4187

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.37497    0.17745   7.748 1.64e-14 ***
alcohol      0.34604    0.01628  21.256  < 2e-16 ***
sulphates    0.99409    0.10235   9.713  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6905 on 1596 degrees of freedom
Multiple R-squared:  0.2699,    Adjusted R-squared:  0.269
F-statistic:   295 on 2 and 1596 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = quality ~ alcohol + volatile_acidity + pH, data = winequality_red)

Residuals:
     Min       1Q   Median       3Q      Max
-2.54671 -0.40614 -0.07843  0.46216  2.21636

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.26881    0.36898  11.569  < 2e-16 ***
alcohol          0.32994    0.01654  19.947  < 2e-16 ***
volatile_acidity -1.27876    0.09911 -12.902  < 2e-16 ***
pH               -0.42187    0.11504  -3.667 0.000253 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6652 on 1595 degrees of freedom
Multiple R-squared:  0.3227,    Adjusted R-squared:  0.3214
F-statistic: 253.3 on 3 and 1595 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = quality ~ alcohol + volatile_acidity + pH + sulphates,
    data = winequality_red)

Residuals:
     Min       1Q   Median       3Q      Max
-2.67671 -0.37422 -0.06988  0.47085  2.07767

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      3.49257    0.38537   9.063  < 2e-16 ***
alcohol          0.32121    0.01641  19.576  < 2e-16 ***
volatile_acidity -1.15583    0.09993 -11.567  < 2e-16 ***
pH               -0.30580    0.11521  -2.654  0.00803 **
sulphates         0.63528    0.10195   6.231 5.91e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6575 on 1594 degrees of freedom
Multiple R-squared:  0.3388,    Adjusted R-squared:  0.3372
F-statistic: 204.2 on 4 and 1594 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = quality ~ log(alcohol) + volatile_acidity + log(pH) +
    sulphates, data = winequality_red)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6838 -0.3775 -0.0699  0.4714  2.0699

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      -1.10242    0.53828  -2.048   0.0407 *
log(alcohol)      3.46244    0.17650  19.618  < 2e-16 ***
volatile_acidity -1.16073    0.09972 -11.640  < 2e-16 ***
log(pH)          -0.97362    0.38162  -2.551   0.0108 *
sulphates         0.63365    0.10207   6.208 6.82e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6572 on 1594 degrees of freedom
Multiple R-squared:  0.3394,    Adjusted R-squared:  0.3378
F-statistic: 204.8 on 4 and 1594 DF,  p-value: < 2.2e-16
```

These figures show that as we increase the number of the variables the R squared values increases but with on the other hand increasing the variable makes a model more complex.

Figure I show the summary for the Best fit method for regression analysis. This can be helpful in finding the variables in different models. As we can see there are some stars in under the variable column so we can trace the stars with the corresponding number of model on the left and see which variables are selected every time the iteration took place to choose the variable. The model 1 chooses the most important variable and here it contains only alcohol in it as in front of number 1 there is only one star which is under category of alcohol.

```
Subset selection object
Call: regsubsets.formula(quality ~ ., data = winequality_red, nvmax = 11)
11 variables  (and intercept)
                     Forced in Forced out
fixed_acidity            FALSE      FALSE
volatile_acidity         FALSE      FALSE
citric_acid              FALSE      FALSE
residual_sugar           FALSE      FALSE
chlorides                FALSE      FALSE
free_sulfur_dioxide      FALSE      FALSE
total_sulfur_dioxide     FALSE      FALSE
density                  FALSE      FALSE
pH                       FALSE      FALSE
sulphates                FALSE      FALSE
alcohol                  FALSE      FALSE
1 subsets of each size up to 11
Selection Algorithm: exhaustive
          fixed_acidity volatile_acidity citric_acid residual_sugar chlorides free_sulfur_dioxide total_sulfur_dioxide
1  ( 1 )  " "           " "              " "         " "           " "       " "                 " "
2  ( 1 )  " "           "*"              " "         " "           " "       " "                 " "
3  ( 1 )  " "           "*"              " "         " "           " "       " "                 " "
4  ( 1 )  " "           "*"              " "         " "           " "       " "                 "*"
5  ( 1 )  " "           "*"              " "         " "           "*"       " "                 "*"
6  ( 1 )  " "           "*"              " "         " "           "*"       " "                 "*"
7  ( 1 )  " "           "*"              " "         " "           "*"       "*"                 "*"
8  ( 1 )  " "           "*"              "*"         " "           "*"       "*"                 "*"
9  ( 1 )  " "           "*"              "*"         "*"           "*"       "*"                 "*"
10  ( 1 ) "*"           "*"              "*"         "*"           "*"       "*"                 "*"
11  ( 1 ) "*"           "*"              "*"         "*"           "*"       "*"                 "*"
          density pH  sulphates alcohol
1  ( 1 )  " "     " " " "       "*"
2  ( 1 )  " "     " " " "       "*"
3  ( 1 )  " "     " " "*"       "*"
4  ( 1 )  " "     " " "*"       "*"
5  ( 1 )  " "     " " "*"       "*"
6  ( 1 )  " "     "*" "*"       "*"
7  ( 1 )  " "     "*" "*"       "*"
8  ( 1 )  " "     "*" "*"       "*"
9  ( 1 )  " "     "*" "*"       "*"
10  ( 1 ) " "     "*" "*"       "*"
11  ( 1 ) "*"     "*" "*"       "*"
```

Fig I

```
Subset selection object
Call: regsubsets.formula(quality ~ ., method = "forward", data = winequality_red,
    nvmax = 11)
11 Variables  (and intercept)
                     Forced in Forced out
fixed_acidity            FALSE      FALSE
volatile_acidity         FALSE      FALSE
citric_acid              FALSE      FALSE
residual_sugar           FALSE      FALSE
chlorides                FALSE      FALSE
free_sulfur_dioxide      FALSE      FALSE
total_sulfur_dioxide     FALSE      FALSE
density                  FALSE      FALSE
pH                       FALSE      FALSE
sulphates                FALSE      FALSE
alcohol                  FALSE      FALSE
1 subsets of each size up to 11
Selection Algorithm: forward
          fixed_acidity volatile_acidity citric_acid residual_sugar chlorides free_sulfur_dioxide total_sulfur_dioxide
1  ( 1 )  " "           " "              " "         " "            " "       " "                 " "
2  ( 1 )  " "           "*"              " "         " "            " "       " "                 " "
3  ( 1 )  " "           "*"              " "         " "            " "       " "                 " "
4  ( 1 )  " "           "*"              " "         " "            " "       " "                 "*"
5  ( 1 )  " "           "*"              " "         " "            "*"       " "                 "*"
6  ( 1 )  " "           "*"              " "         " "            "*"       " "                 "*"
7  ( 1 )  " "           "*"              " "         " "            "*"       "*"                 "*"
8  ( 1 )  " "           "*"              "*"         " "            "*"       "*"                 "*"
9  ( 1 )  " "           "*"              "*"         "*"            "*"       "*"                 "*"
10  ( 1 ) "*"           "*"              "*"         "*"            "*"       "*"                 "*"
11  ( 1 ) "*"           "*"              "*"         "*"            "*"       "*"                 "*"
          density pH  sulphates alcohol
1  ( 1 )  " "     " " " "       "*"
2  ( 1 )  " "     " " " "       "*"
3  ( 1 )  " "     " " "*"       "*"
4  ( 1 )  " "     " " "*"       "*"
5  ( 1 )  " "     " " "*"       "*"
6  ( 1 )  " "     "*" "*"       "*"
7  ( 1 )  " "     "*" "*"       "*"
8  ( 1 )  " "     "*" "*"       "*"
9  ( 1 )  " "     "*" "*"       "*"
10  ( 1 ) " "     "*" "*"       "*"
11  ( 1 ) "*"     "*" "*"       "*"
```

Fig G

Fig G shows the results for the forward regression analysis with all the variables.

```
Subset selection object
Call: regsubsets.formula(quality ~ ., method = "backward", data = winequality_red,
    nvmax = 11)
11 variables  (and intercept)
                     Forced in Forced out
fixed_acidity            FALSE      FALSE
volatile_acidity         FALSE      FALSE
citric_acid              FALSE      FALSE
residual_sugar           FALSE      FALSE
chlorides                FALSE      FALSE
free_sulfur_dioxide      FALSE      FALSE
total_sulfur_dioxide     FALSE      FALSE
density                  FALSE      FALSE
pH                       FALSE      FALSE
sulphates                FALSE      FALSE
alcohol                  FALSE      FALSE
1 subsets of each size up to 11
Selection Algorithm: backward
          fixed_acidity volatile_acidity citric_acid residual_sugar chlorides free_sulfur_dioxide total_sulfur_dioxide
1  ( 1 )  " "           " "              " "         " "           " "       " "                   " "
2  ( 1 )  " "           "*"              " "         " "           " "       " "                   " "
3  ( 1 )  " "           "*"              " "         " "           " "       " "                   " "
4  ( 1 )  " "           "*"              " "         " "           " "       " "                   "*"
5  ( 1 )  " "           "*"              " "         " "           "*"       " "                   "*"
6  ( 1 )  " "           "*"              " "         " "           "*"       " "                   "*"
7  ( 1 )  " "           "*"              " "         " "           "*"       "*"                   "*"
8  ( 1 )  " "           "*"              "*"         " "           "*"       "*"                   "*"
9  ( 1 )  " "           "*"              "*"         "*"           "*"       "*"                   "*"
10  ( 1 ) "*"           "*"              "*"         "*"           "*"       "*"                   "*"
11  ( 1 ) "*"           "*"              "*"         "*"           "*"       "*"                   "*"
          density pH  sulphates alcohol
1  ( 1 )  " "     " " " "       "*"
2  ( 1 )  " "     " " " "       "*"
3  ( 1 )  " "     " " "*"       "*"
4  ( 1 )  " "     " " "*"       "*"
5  ( 1 )  " "     " " "*"       "*"
6  ( 1 )  " "     "*" "*"       "*"
7  ( 1 )  " "     "*" "*"       "*"
8  ( 1 )  " "     "*" "*"       "*"
9  ( 1 )  " "     "*" "*"       "*"
10  ( 1 ) " "     "*" "*"       "*"
11  ( 1 ) "*"     "*" "*"       "*"
```

Fig H

Fig H shows the summary for the backward method of regression analysis.

CODE:

```
install.packages("tidyverse")

install.packages("leaps")

install.packages("ISLR")

library(tidyverse)

library(ggplot2)

library(dplyr)

library(tidyquant)

library(MASS)

library(leaps)

library(ISLR)

library(boot)

library(stats)

library(GGally)
```

```
attach(winequality_red)


ggplot(data=winequality_red)+
  geom_bar(mapping = aes(x= factor(quality)))+
  xlab("Quality (3/Poor to 8/High)")+
  ylab("Number of samples")+
  ggtitle("Vinho Verde Red Wine")



ggplot(data=winequality_red,mapping = aes(x=density,y= alcohol))+
  geom_point()+
  geom_line()+
  xlab("Density")+
  ylab("Fixed Acidity")+
  ggtitle("Alcohol Vs Density")



ggplot(data=winequality_red,mapping = aes(x=density,y= fixed_acidity))+
  geom_point()+
  geom_line()+
  xlab("Density")+
  ylab("Fixed Acidity")+
  ggtitle("Fixed Acidity Vs Density")
```

```
ggplot(data=winequality_red)+
  geom_boxplot(mapping = aes(x=factor(quality), y=alcohol))+
  xlab("Quality")+
  ylab("Alcohol")+
  ggtitle("Alcohol Vs Quality")
```

```
quality8=filter(winequality_red, quality >= 7)
quality8=data.frame(quality8)
View(quality8)
```

```
ggplot(data=quality8)+
  geom_histogram(mapping= aes(x=sulphates, fill=factor(quality)),position = 'dodge')+
  xlab("Sulphates")+
  ylab("Number of Samples")+
  labs(fill="Grade of Wine")+
  ggtitle("Sulphates effect")
```

```
ggplot(data=quality8)+
  geom_bar(mapping = aes(x=volatile_acidity,fill=factor(quality)),position=position_dodge2(preserve = "single"),width = 0.75)+
  xlab("")+
  ylab("Number of Samples")+
  labs(fill="Grade of Wine")+
  ggtitle("Alcohol effect")



ggplot(data=winequality_red,mapping = aes(x=density,y= alcohol))+
  geom_point()+
  geom_line()+
  xlab("Density")+
  ylab("Alcohol")+
  ggtitle("Alcohol Vs Density")



ggplot(data=winequality_red,mapping = aes(x=density,y= fixed_acidity))+
  geom_point()+
  geom_line()+
  xlab("Density")+
  ylab("Fixed Acidity")+
  ggtitle("Fixed Acidity Vs Density")



####LET'S LOOK COERRELEATION


cor(winequality_red)
```

```
attach(winequality_red)

View(winequality_red)



m1=lm(data=winequality_red,quality~alcohol+sulphates)

summary(m1)



mo1=lm(data = winequality_red, quality~alcohol+volatile_acidity+pH)

summary(mo1)

mo2=lm(data = winequality_red, quality~alcohol+volatile_acidity+pH+sulphates)

summary(mo2)

mo21=lm(data = winequality_red, quality~log(alcohol)+volatile_acidity+log(pH)+sulphates)

summary(mo21)



#####best result

mo22=lm(data = winequality_red, quality~log(alcohol)+volatile_acidity+log(pH)+log(sulphates))

summary(mo22)

mo23=lm(data = winequality_red,
quality~log(alcohol)+poly(volatile_acidity,2,raw='TRUE')+log(pH)+log(sulphates))

summary(mo23)
```

```
mo3=lm(data = winequality_red, quality~poly(alcohol,2,raw = 'TRUE')+volatile_acidity+pH+sulphates)

summary(mo3)



mo4=lm(data = winequality_red, quality~poly(alcohol,2,raw =
'TRUE')+poly(volatile_acidity,2)+pH+sulphates)

summary(mo4)


mo5=lm(data = winequality_red, quality~poly(alcohol,3,raw =
'TRUE')+poly(volatile_acidity,2)+pH+sulphates)

summary(mo5)




#adding other variables too


model1= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide+total_sulf
ur_dioxide+density+pH+sulphates+alcohol)

model1

summary(model1)


model2= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide+total_sulf
ur_dioxide+density+pH+sulphates)

summary(model2)



model3= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide+total_sulf
ur_dioxide+density+pH)
```

```
summary(model3)
```

```
model4= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide+total_sulfur_dioxide+density)

summary(model4)
```

```
model5= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide+total_sulfur_dioxide)

summary(model5)
```

```
model6= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides+free_sulfur_dioxide)

summary(model6)
```

```
model7= lm(data=winequality_red,
quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+chlorides)

summary(model7)
```

```
model8= lm(data=winequality_red, quality~volatile_acidity+alcohol+sulphates+total_sulfur_dioxide)

summary(model8)
```

```
# REgression Method 'Best'


fit1= regsubsets(quality~.,data=winequality_red,nvmax = 11)

summary(fit1)

summary(fit1)$rsq

summary(fit1)$adjr2

summary(forward)$cp

summary(fit1)$bic

summary(fit1)$rss

coef(fit1,6)

coef(forward,6)

coef(backward,6)


x = seq(1, 11, by=1)

y = summary(fit1)$rsq

rsq1 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=rsq1)+geom_point()+geom_line()+xlab("Model Number")+ylab("R2")+

  ggtitle("Model selection with BEST method")

# 8th model has the highest adjr2



x = seq(1, 11, by=1)

y = summary(fit1)$cp

cp1 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=cp1)+geom_point()+geom_line()+xlab("Model Number")+ylab("CP")+
ggtitle("Model selection with BEST method")

# 7th model has lowest cp
```

```
x = seq(1, 11, by=1)

y = summary(fit1)$bic

adjr2bic1 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2bic1)+geom_point()+geom_line()+xlab("Model
Number")+ylab("BIC")+ggtitle("Model selection with BEST method")

# 6th model has the lowest bic


x = seq(1, 11, by=1)

y = summary(fit1)$rss

adjr2rss1 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2rss1)+geom_point()+geom_line()+xlab("Model Number")+ylab("RSS")

# till 6 it has  a good change in rss




coef(fit1,6)

coef(fit1,7)

coef(fit1,8)




###poly 2

fit2 = regsubsets(quality~ poly(fixed_acidity, 2, raw = 'TRUE')+poly(volatile_acidity, 2, raw = 'TRUE')+

        poly(citric_acid,2, raw = 'TRUE')+poly(chlorides, 2, raw = 'TRUE')+

        poly(free_sulfur_dioxide,2, raw = 'TRUE')+poly(total_sulfur_dioxide,2, raw = 'TRUE')+poly(pH,2,
raw = 'TRUE')+poly(sulphates,2, raw = 'TRUE')+poly(alcohol,2, raw = 'TRUE'),data=winequality_red, nvmax
= 18)
```

```
summary(fit2)$adjr2

summary(fit3)$adjr2

summary(fit2)$adjr2

summary(fit2)$cp


x = seq(1, 18, by=1)

y = summary(fit2)$adjr2

adjr2Res2 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res2)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjusted R2 for 'poly 2'")


coef(fit2,6)


x = seq(1, 18, by=1)

y = summary(fit2)$cp

adjr2Res2 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res2)+geom_point()+geom_line()+xlab("Model Number")+ylab("CP for 'poly 2'")


x = seq(1, 18, by=1)

y = summary(fit2)$bic

adjr2Res2 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res2)+geom_point()+geom_line()+xlab("Model Number")+ylab("BIC for 'poly 2'")


coef(fit2,1)

coef(fit2,2)

coef(fit2,3)

coef(fit2,4)
```

```
coef(fit2,5)

coef(fit2,6)

coef(fit2,7)

coef(fit2,8)

coef(fit2,9)
```

```
####poly 3

fit3 = regsubsets(quality~ poly(fixed_acidity, 3, raw = 'TRUE')+poly(volatile_acidity, 3, raw = 'TRUE')+

        poly(citric_acid, 3, raw = 'TRUE')+poly(chlorides, 3, raw = 'TRUE')+

        poly(free_sulfur_dioxide, 3, raw = 'TRUE')+poly(total_sulfur_dioxide,3, raw =
'TRUE')+poly(pH,3, raw = 'TRUE')+poly(sulphates,3, raw = 'TRUE')+poly(alcohol,3, raw =
'TRUE'),data=winequality_red, nvmax = 27)


summary(fit3)rsq

summary(fit3)$adjr2

summary(fit3)$cp

summary(fit3)$bic


x = seq(1, 27, by=1)

y = summary(fit3)$adjr2

adjr2Res3 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res3)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjusted
R2 for 'poly 3'")


x = seq(1, 27, by=1)

y = summary(fit3)$cp
```

```
adjr2Res3 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res2)+geom_point()+geom_line()+xlab("Model Number")+ylab("CP for 'poly 3'")


x = seq(1, 27, by=1)

y = summary(fit3)$bic

adjr2Res3 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res3)+geom_point()+geom_line()+xlab("Model Number")+ylab("BIC R2 for 'poly 3'")



coef(fit3,1)

coef(fit3,2)

coef(fit3,3)


coef(fit3,4)









####poly4

fit4 = regsubsets(quality~ poly(fixed_acidity, 4, raw = 'TRUE')+poly(volatile_acidity, 4, raw = 'TRUE')+

        poly(citric_acid, 4, raw = 'TRUE')+poly(chlorides, 4, raw = 'TRUE')+

        poly(free_sulfur_dioxide, 4, raw = 'TRUE')+poly(total_sulfur_dioxide,4, raw = 'TRUE')+poly(pH,4, raw = 'TRUE')+poly(sulphates,4, raw = 'TRUE')+poly(alcohol,4, raw = 'TRUE'),winequality_red,nvmax = 36)


summary(fit4)$adjr
```

```
x = seq(1, 36, by=1)

y = summary(fit4)$adjr2

adjr2Res4 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res4)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjusted R2 for 'poly 4'")



x = seq(1, 36, by=1)

y = summary(fit4)$cp

adjr2Res4 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res4)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjusted R2 for 'poly 4'")


x = seq(1,36, by=1)

y = summary(fit4)$bic

adjr2Res4 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res4)+geom_point()+geom_line()+xlab("Model Number")+ylab("BIC for 'poly 4'")


coef(fit4,1)

coef(fit4,2)

coef(fit4,3)

coef(fit4,4)


fit5 = regsubsets(quality~ volatile_acidity+chlorides+poly(sulphates,5, raw = 'TRUE')+

            log(alcohol),winequality_red,nvmax = 16)


summary(fit5)$adjr

x = seq(1, 16, by=1)

y = summary(fit5)$adjr2
```

```
adv5 = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adv5)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjusted R2 for 'poly 5'")

coef(fit5,4)


################################forward approach for regression




forward = regsubsets(quality~., method="forward", data=winequality_red,nvmax = 11)

summary(forward)

summary(forward)$adjr2

summary(forward)$rsq

summary(forward)$cp

summary(forward)$bic

summary(forward)$rss




x = seq(1, 11, by=1)

y = summary(forward)$rsq

forwardrsq = data.frame(x, y)

ggplot(aes(x=x, y=y), data=forwardrsq)+geom_point()+geom_line()+

  xlab("Model Number")+

  ylab("R2")+

  ggtitle("Model selection with 'Forward' method")




x = seq(1, 11, by=1)
```

```
y = summary(forward)$cp

forwardcp = data.frame(x, y)

ggplot(aes(x=x, y=y), data=forwardcp)+geom_point()+geom_line()+

  xlab("Model Number")+

  ylab("CP")+

  ggtitle("Model selection with 'Forward' method")




x = seq(1, 11, by=1)

y = summary(forward)$bic

forwardbic = data.frame(x, y)

ggplot(aes(x=x, y=y), data=forwardbic)+geom_point()+geom_line()+

  xlab("Model Number")+

  ylab("BIC")+

  ggtitle("Model selection with 'Forward' method")



x = seq(1, 11, by=1)

y = summary(forward)$rss

forwardrss = data.frame(x, y)

ggplot(aes(x=x, y=y), data=forwardrss)+geom_point()+geom_line()+xlab("Model Number")+ylab("RSS for
'Forward'")



coef(forward,1)

coef(forward,2)

coef(forward,3)

coef(forward,4)
```

```
##################backward approach for regression



backward = regsubsets(quality~., method="backward", data=winequality_red,nvmax = 11)
summary(backward)$aic



x = seq(1, 11, by=1)
y = summary(backward)$rsq
backrsq1 = data.frame(x, y)
ggplot(aes(x=x, y=y), data=backrsq1)+geom_point()+geom_line()+
  xlab("Model Number")+
  ylab("R2")+
  ggtitle("Model selection with 'Backward' method")

x = seq(1, 11, by=1)
y = summary(backward)$cp
backcp1 = data.frame(x, y)
ggplot(aes(x=x, y=y), data=backcp1)+geom_point()+geom_line()+
  xlab("Model Number")+
  ylab("Cp")+
  ggtitle("Model selection with 'Backward' method")



x = seq(1, 11, by=1)
```

```
y = summary(backward)$bic

backbic1= data.frame(x, y)

ggplot(aes(x=x, y=y), data=backbic1)+geom_point()+geom_line()+

  xlab("Model Number")+

  ylab("BIC")+

  ggtitle("Model selection with 'Backward' method")



x = seq(1, 11, by=1)

y = summary(backward)$rss

backwardrss = data.frame(x, y)

ggplot(aes(x=x, y=y), data=backwardrss)+geom_point()+geom_line()+xlab("Model Number")+ylab("RSS for 'Backward'")



summary(backward)$adjr2

summary(backward)$rsq

summary(backward)$cp

summary(backward)$bic

summary(backward)$rss



#


backwardp2 = regsubsets(quality~.+poly(density,5,raw= 'True')-density, method="backward", data=winequality_red,nvmax = 11)

summary(backward)
```

```
x = seq(1, 11, by=1)

y = summary(backward)$adjr2

adjr2Res = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res)+geom_point()+geom_line()+xlab("Model Number")+ylab("Adjr2 for 'Backward poly'")



x = seq(1, 11, by=1)

y = summary(backward)$cp

adjr2Res = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res)+geom_point()+geom_line()+xlab("Model Number")+ylab("Cp for 'Backward poly'")



x = seq(1, 11, by=1)

y = summary(backward)$bic

adjr2Res = data.frame(x, y)

ggplot(aes(x=x, y=y), data=adjr2Res)+geom_point()+geom_line()+xlab("Model Number")+ylab("BIC for 'Backward poly'")


summary(backwardp2)$adjr2



##############    cross validation
#################################################################



mod = rep(0, 10)
```

```
mod[1] = "quality~ volatile_acidity + chlorides+ sulphates + alcohol"

mod[2] = "quality~ volatile_acidity + chlorides+ sulphates + alcohol+pH"

mod[3]=    "quality~ volatile_acidity + chlorides+ poly(sulphates,5, raw = 'TRUE')+log(alcohol)"

mod[4] = "quality~ volatile_acidity + chlorides+ poly(sulphates,2) + log(alcohol)+poly(pH,2)"

mod[5] = "quality~ volatile_acidity + chlorides+ poly(sulphates,3) + log(alcohol)+poly(pH,3)"

mod[6] = "quality~ volatile_acidity + chlorides+ poly(sulphates,4) + log(alcohol)+poly(pH,4)"

mod[7] = "quality~ volatile_acidity + chlorides+ poly(sulphates,4) + log(alcohol)+poly(pH,4)+
total_sulfur_dioxide"

mod[8] = "quality~ volatile_acidity + chlorides+ poly(sulphates,4) + log(alcohol)+poly(pH,4)+
poly(total_sulfur_dioxide,2)"

mod[9] = "quality~ volatile_acidity + chlorides+ poly(sulphates,4) + log(alcohol)+poly(pH,4)+
poly(total_sulfur_dioxide,3)"

mod[10] = "quality~ volatile_acidity + chlorides+ poly(sulphates,4) + log(alcohol)+poly(pH,4)+
poly(total_sulfur_dioxide,4)"




cv.error=rep(0,10)

for(i in 1:10){

  model=glm(eval(parse(text=paste(mod[j]))))

  cv.error[i] = cv.glm(winequality_red, model, K=nrow(winequality_red))$delta[1]

}

View(cv.error)

cv.error


numRows = nrow(winequality_red)

id = seq(1, numRows, by =1)


wineShuffle = slice(winequality_red, sample(1:n()))
```

```
wineShuffle = mutate(wineShuffle, id)


k=5


errors = matrix( nrow = 10, ncol = 5)
errors[1,2] = 0
View(errors)


for(j in 1:10){
  for(i in 1:5){
    errors[j,i] = 0
  }
}
totalError = 0
for(j in 1:10){ # the 10 different models
  for(i in 1:k){ #the k folds for each model
    test = filter(wineShuffle, id >= (i-1)*numRows/k+1 & id <=i*numRows/k)
    train = anti_join(wineShuffle, test, by="id")
    model = lm(eval(parse(text=paste(mod[j])), train))
    errors[j,i] = mean((test$quality - predict.lm(model, test))^2)
  }
}



avgRegEr = rep(0,10)
avgRegEr
for(j in 1:10){
  for(i in 1:5){
    avgRegEr[j] = avgRegEr[j]+errors[j, i]
```

```
  }
}
avgRegEr/k
cv.error
#now we confirmed that our calculated CV errors are very similar to those offered by glm
#(difference likely due to random points chosen) we can move on
#we did this so we could find Standard Error (SE) and create error bars
se = rep(0,10)
for (i in 1:10){
  se[i] = sqrt(var(errors[i,])/k)
}
se
#now making data frame for ease of plotting
x = seq(1,10, by = 1)
faithBest = data.frame(x,avgRegEr/k , se)
faithBest
ggplot(data = faithBest, aes(x = x, y=avgRegEr.k))+
  geom_point()+
  geom_line()+
  geom_errorbar(aes(ymin = avgRegEr.k-se, ymax = avgRegEr.k +se))+
  xlab("Model Number")+
  ylab("Error")+
  ggtitle("Cross Validation Standard Error Plot")



##########################################Classification#####################################
########
```

```
attach(winequality_red)

ggplot(data=winequality_red)+
  geom_point(mapping = aes(x=alcohol,y=,color=factor(quality)))

ggpairs(winequality_red, aes(colour=factor(quality)))

table(winequality_red$quality)



winelevel0= lda(quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+
                chlorides+free_sulfur_dioxide+total_sulfur_dioxide+density+pH+
                sulphates+alcohol,data=winequality_red)
winePred0 = predict(winelevel0, winequality_red)
table(winePred0$class, winequality_red$quality)
#965



winelevel = lda(quality~fixed_acidity+volatile_acidity, data=winequality_red)
winePred = predict(winelevel, winequality_red)
table(winePred$class, winequality_red$quality)



winelevel = lda(quality~alcohol+sulphates+volatile_acidity,data=winequality_red)
winePred = predict(winelevel, winequality_red)
table(winePred$class, winequality_red$quality)
#918/1599

winelevela = lda(quality~alcohol+sulphates+volatile_acidity+fixed_acidity,data=winequality_red)
```

```
winePreda = predict(winelevela, winequality_red)

table(winePreda$class, winequality_red$quality)

#917/1599


winelevelb = lda(quality~alcohol+poly(sulphates,2,raw = 'TRUE')+volatile_acidity, data=winequality_red)

winePredb = predict(winelevelb, winequality_red)

table(winePredb$class, winequality_red$quality)

#926/1599



winelevelc = lda(quality~alcohol+poly(sulphates,2,raw = 'TRUE')+volatile_acidity+fixed_acidity,
data=winequality_red)

winePredc = predict(winelevelc, winequality_red)

table(winePredc$class, winequality_red$quality)

#928/1599


wineleveld = lda(quality~alcohol+poly(sulphates,5,raw =
'TRUE')+volatile_acidity+fixed_acidity+pH,data=winequality_red)

winePredd = predict(wineleveld, winequality_red)

table(winePredd$class, winequality_red$quality)

#937/1599



winelevele = lda(quality~poly(alcohol,3,raw = 'TRUE')+poly(sulphates,5,raw =
'TRUE')+volatile_acidity+fixed_acidity)

winePrede = predict(winelevele, winequality_red)

table(winePrede$class, winequality_red$quality)

#926/1599
```

```
winelevelf = lda(quality~poly(alcohol,3,raw = 'TRUE')+sulphates+volatile_acidity+fixed_acidity)

winePredf = predict(winelevelf, winequality_red)

table(winePredf$class, winequality_red$quality)

#896


# tarining and testing for classification models

id = seq(1, 150, by=1)

winemix = slice(winequality_red, sample(1:n()))

winerando = mutate(winemix, id)



IDwine = mutate(winequality_red, id=row_number())

View(IDwine)


trainDataSet = sample_frac(IDwine, .5)

View(trainDataSet)


testDatawine = anti_join(IDwine, trainDataSet, by ="id")

View(testDatawine)



wineModel = lda(quality~alcohol+poly(sulphates,5,
raw='TRUE')+volatile_acidity+fixed_acidity+pH,data=trainDataSet)



wineTrainSetPredictions= predict(wineModel, trainDataSet)

table(wineTrainSetPredictions$class, trainDataSet$quality)
```

```
################################################################################################
#########################
#                  Cross Validation for classificaton


attach(winequality_red)

id = seq(1, 1599, by=1)

wineMix = slice(winequality_red, sample(1:n()))

wineRando = mutate(wineMix, id)


k=5

numRows=nrow(winequality_red)

errors8=rep(0,k)

totalError=0

for (i in 1:k) {

 test=filter(wineRando,id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

 train = anti_join(wineRando, test, by="id")

 medo=lda(data=train, quality~fixed_acidity+volatile_acidity+citric_acid+residual_sugar+

          chlorides+free_sulfur_dioxide+total_sulfur_dioxide+density+pH+

          sulphates+alcohol)

 medoguess=predict(medo,test)

 errors8[i]=1-mean(medoguess$class == test$quality)

 totalError= errors8[i]+totalError


}
```

```
errors1

errors2

errors3

errors4

errors5

errors6

errors7

errors8


avgE= rep(0,8)

for (i in 1:k) {

  avgE[1]=errors1[i]+avgE[1]

  avgE[2]=errors2[i]+avgE[2]

  avgE[3]=errors3[i]+avgE[3]

  avgE[4]=errors4[i]+avgE[4]

  avgE[5]=errors5[i]+avgE[5]

  avgE[6]=errors6[i]+avgE[6]

  avgE[7]=errors7[i]+avgE[7]

  avgE[8]=errors8[i]+avgE[8]

}


se=rep(0,8)


for (i in 1:k) {

  avgE[i]=avgE[i]/k


}


avgE
```

```
se[1]=sqrt(var(errors1)/k)

se[2]=sqrt(var(errors2)/k)

se[3]=sqrt(var(errors3)/k)

se[4]=sqrt(var(errors4)/k)

se[5]=sqrt(var(errors5)/k)

se[6]=sqrt(var(errors6)/k)

se[7]=sqrt(var(errors7)/k)

se[8]=sqrt(var(errors8)/k)


cvwine=data.frame(avgE,se)

View(cvwine)


mn=seq(1,8,by=1)

cvwine=data.frame(avgE,se,mn)



ggplot(cvwine,aes(x=mn,y=avgE))+

 geom_line()+

 geom_point()+

 geom_errorbar(aes(ymin=avgE-se,ymax=avgE+se))+


 ylab(" Error")+

 xlab("Model Number")+

 ggtitle("Cross Validation for Classification models")
```

Note: Some part of the code was copied from the class notes and fully modified for this data set.