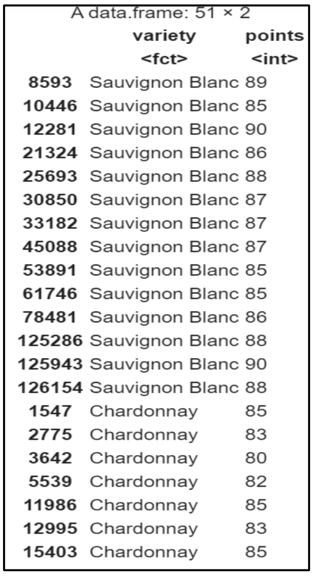
Kaggle Wine Data Set Analysis

CS7DS3 – Applied Statistical Modelling

Q1. My wife likes Sauvignon Blanc from South Africa. My mother-in law likes Chardonnay from Chile. Both agree that €15 is the right amount to spend on a bottle of wine.

A .i . Which type of wine is better rated? How much better?

In order to answer this question we first need to analyze the raw data from the .csv file. The data set contains in total 129971 rows and 13 columns. However, to answer the current question we need to focus on Sauvignon Blanc from South Africa and Chardonnay from Chile. So first of all we need to extract these records from the data set.



```
A data.frame: 51 × 2
              variety          points
              <fct>            <int>
  8593   Sauvignon Blanc 89
 10446   Sauvignon Blanc 85
 12281   Sauvignon Blanc 90
 21324   Sauvignon Blanc 86
 25693   Sauvignon Blanc 88
 30850   Sauvignon Blanc 87
 33182   Sauvignon Blanc 87
 45088   Sauvignon Blanc 87
 53891   Sauvignon Blanc 85
 61746   Sauvignon Blanc 85
 78481   Sauvignon Blanc 86
125286   Sauvignon Blanc 88
125943   Sauvignon Blanc 90
126154   Sauvignon Blanc 88
  1547   Chardonnay      85
  2775   Chardonnay      83
  3642   Chardonnay      80
  5539   Chardonnay      82
 11986   Chardonnay      85
 12995   Chardonnay      83
 15403   Chardonnay      85
```

Figure 1: Filtered Out Data

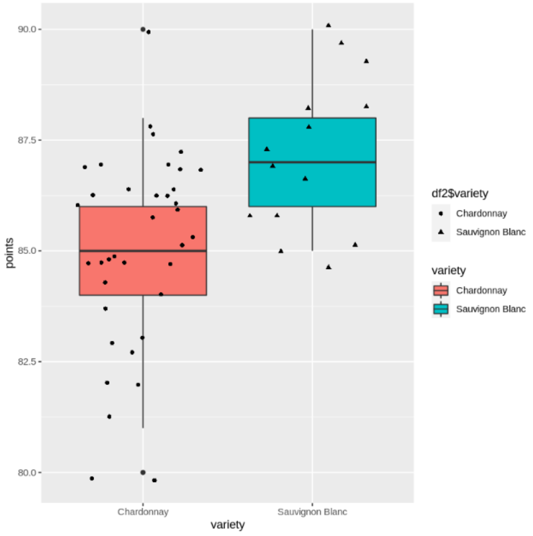


Figure 2:Boxplot

Figure 1 shows the filtered-out data sets containing only Sauvignon Blanc wine from South Africa and Chardonnay from Chile, all the wines in the data set are priced at 15 Euros. Only the variety and points columns are kept because as of now we don't need any other columns. So now we have the data set which contains only the data required for current question.

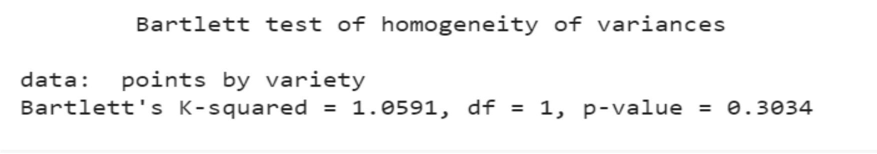
Now we need to compare the two groups, figure 2, suggests us which is the better rated wine from the two choices. To answer this there are few options we can explore further. First of all we need to explore the data in hand.

	Mean	Median	Standard Deviation
Sauvignon Blanc	87.21	87	1.71
Chardonnay	85.08	85	2.23

Table 1

Figure 2 shows a boxplot of the two groups, from a rough look it is quiet evident that the Sauvignon Blanc is better rated than Chardonnay, also from the Table 1 it is observable that Sauvignon Blanc has better mean and median values. So, we can say that Sauvignon Blanc is a better rated wine, however if we take another look at the figure 2, it contains the scatter plot of the two groups, observing it we can say that there are very few records present for Sauvignon Blanc as compared to Chardonnay which has more number of records and are well spread out. There can be number of scenarios which can wrongly suggest better average ratings for Sauvignon Blanc, like for this sample we may have only high rated Sauvignon Blanc wines. So, by observing only this sample we can not affirmatively say which wine is better rated.

Now to compare the two groups more effectively we can perform the t-test. There are two types of t-tests, equal variance t-test and unequal variance t-test, so by just looking at the two groups we can not say anything about the variance of the two groups. There are some tests available to check homogeneity of the variance among two groups. We will perform Bartlett test to check whether the two groups have similar variance.

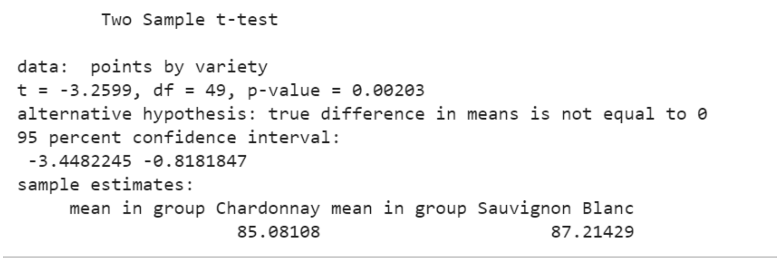


```
        Bartlett test of homogeneity of variances

data:  points by variety
Bartlett's K-squared = 1.0591, df = 1, p-value = 0.3034
```

Figure 3: Bartlett Test

Figure 5 shows the output of Bartlett test, from the test we have p-value of 0.30 which is significantly higher than 0.05, so we can safely assume for the two groups to have equal variance. Hence, we will perform t-test with equal variance.



```
        Two Sample t-test

data:  points by variety
t = -3.2599, df = 49, p-value = 0.00203
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.4482245 -0.8181847
sample estimates:
     mean in group Chardonnay mean in group Sauvignon Blanc
                     85.08108                      87.21429
```

Figure 4: T-test

Figure 4 shows the output of t-test, from it we can see that p-value returned is 0.002 which is significantly below 0.05. hence we can safely reject the null hypothesis and can assume that there is a statistically significant difference between the two groups.

Sauvignon Blanc is better rated wine than the Chardonnay and we can say with 95% confidence that the difference of mean between two groups lies between 0.81 to 3.44.

Q- a) 2) Suppose I buy a South African Sauvignon Blanc and a Chilean Chardonnay, both priced €15. What is the probability that the Sauvignon Blanc will be better?

To compare the means of two groups there are other options available other than t-test, to answer the current question we will use Gibbs Sampler to model the difference between the two groups.

I have used a custom function compare_2_gibbs to perform the sampling task. This function takes the following arguments.

- Y – numerical values of ratings for two groups
- Ind – grouping variables
- mu0 = 8 – mean of the normal prior for the mean parameter of overall data
- tau0 = 1/400 - precision of the normal prior for the mean parameter of overall data
- del0 = 0 - mean of the normal prior for the mean parameter of difference of means
- gamma0 = 1/400 - precision of the normal prior for the mean parameter of difference of means
- a0 = 1 - hyperparameter for gamma prior for overall precision
- b0 = 5 - hyperparameter for gamma prior for overall precision
- maxiter = 10000 – total number of iterations for sampler

Evaluation of the model

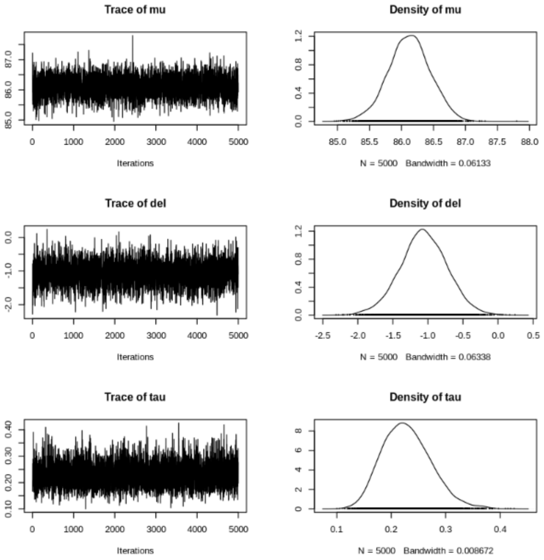


Figure 5: Traces of mu, delta and tau

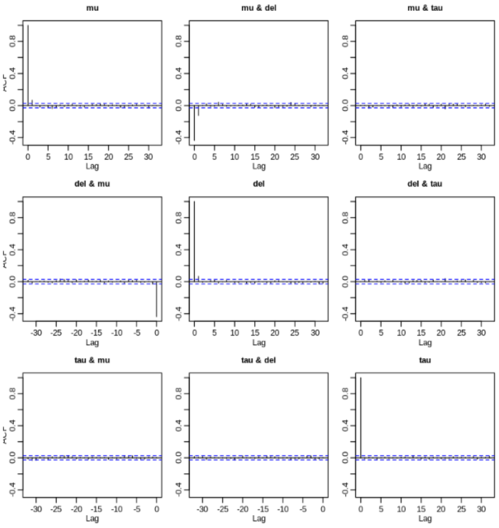
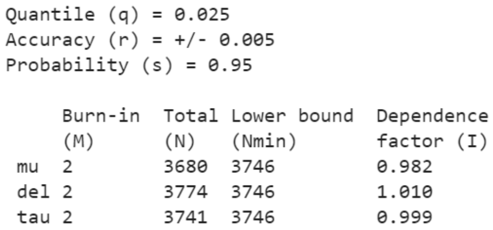


Figure 6: Estimates of autocorrelation

From figure 5 and figure 6 it can be observed that the model is performing satisfactorily, in the figure 6 it might be observed that there are few spikes in lag values, to handle this issue I ran the model for 10,000 iterations and afterward selected every alternate sample from the 10k samples, this technique is called thinning. By removing samples at regular intervals we can avoid high correlation between successive samples.

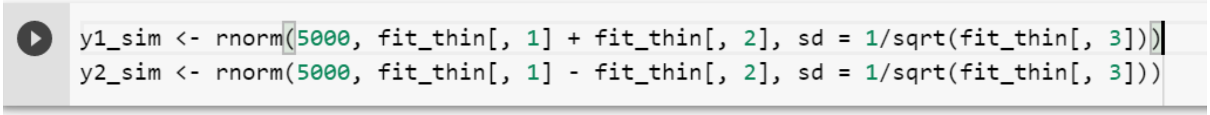


```
Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

     Burn-in  Total Lower bound  Dependence
     (M)      (N)   (Nmin)       factor (I)
 mu  2        3680  3746         0.982
 del 2        3774  3746         1.010
 tau 2        3741  3746         0.999
```

Figure 7: Output of raftery.diag()

Figure 7 shows the output of raftery.diag function, the key outputs to note here are Burn-in and Dependence Factor, values close to 0-1 indicates that our model is performing well. So now that we have evaluated our model, it can be used to measure values to compare the two groups.

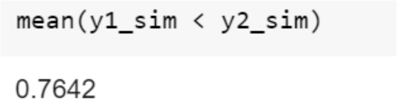


```
y1_sim <- rnorm(5000, fit_thin[, 1] + fit_thin[, 2], sd = 1/sqrt(fit_thin[, 3]))
y2_sim <- rnorm(5000, fit_thin[, 1] - fit_thin[, 2], sd = 1/sqrt(fit_thin[, 3]))
```

Fit_thin[,1] → sample mean of first group(Chardonnay) from Gibbs sampler

Fit_thinp[,2] → sample mean difference of the two groups from Gibbs sampler

So, now taking mean for samples where y2_sim > y1_sim we can give the probability of Sauvignon Blanc being the better wine.



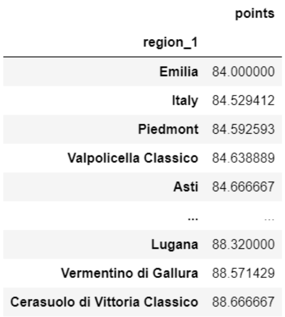
```
mean(y1_sim < y2_sim)

0.7642
```

So, there is around 76% probability that randomly selected Sauvignon Blanc will be better rated than randomly selected Chardonnay.

Q- Consider the Italian wines in the dataset. Which regions produce better than average wine? Limit your analysis to wines costing less than €20 and to regions which have at least four such reviews.

To answer this question the data needs to be processed in order to contain only the Italian wines with price under 20 and also to contain only the Italian regions having at least 4 reviews , after filtering the data, the data set now contains 4614 records with 142 distinct regions.



	points
region_1	
Emilia	84.000000
Italy	84.529412
Piedmont	84.592593
Valpolicella Classico	84.638889
Asti	84.666667
...	...
Lugana	88.320000
Vermentino di Gallura	88.571429
Cerasuolo di Vittoria Classico	88.666667

From figure 11 it is clearly observable than mean ratings of some Italian regions is clearly better than others. There are clearly some regions having better ratings than the mean rating i.e. 86.6. using the samples in the data, there are 68 regions which have better mean rating than the average.

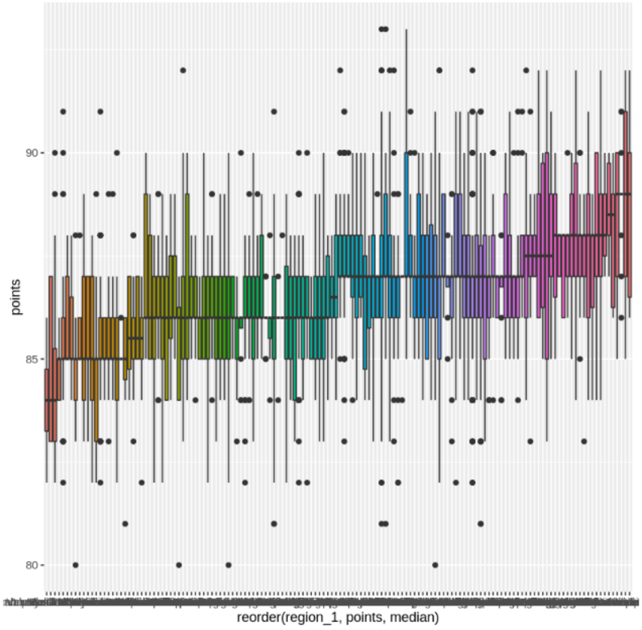


Figure 9: Median rating of Italian regions

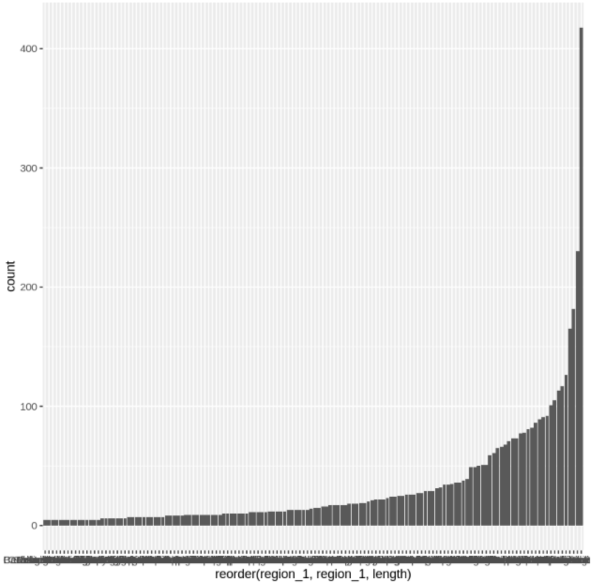


Figure 10: number of reviews for each region

Figure 10 shows the median ratings of all the Italian regions, looking at the figure it can be observed that there is not much difference between the median ratings of regions, for most of the regions the median value lies between 84-89. Figure 9 shows the number of reviews for each region, as it is observable that there are regions having very low number of reviews. So there is a high chance that these sample means might differ than the real value, to tackle this problem I used Gibbs sampler to compare the means of all Italian regions.

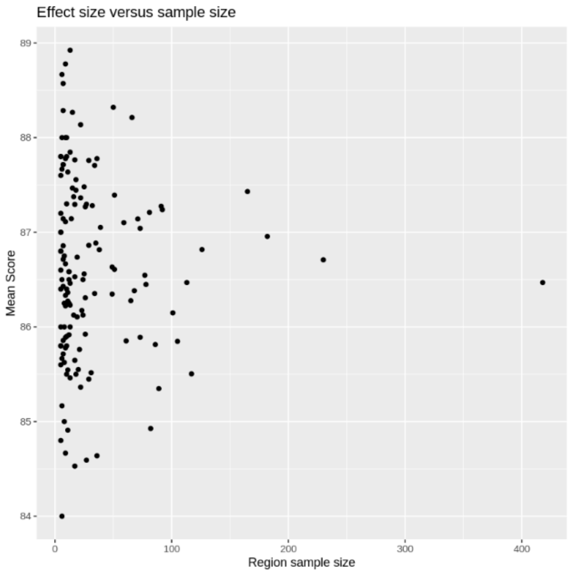


Figure 11: Effect of sample size on mean rating

Also from the Figure 11, it can be observed that as the number of sample increases along the x-axis the mean rating moves more closer to the mean rating of all the regions. This indicates that we need to perform Bayesian sampling to get close to the true picture.

Gibbs Sampling

I used a custom function which implements Gibbs sampling to compare multiple data samples. Using this model we will sample the mean scores for all the regions of Italy. The main reason behind doing this is to address the sample variability, so that we can make better estimates regarding the differences between different group means.

The parameters used by this function are same as we used for estimating difference between Sauvignon Blanc and Chardonnay.

Evaluation of the model:

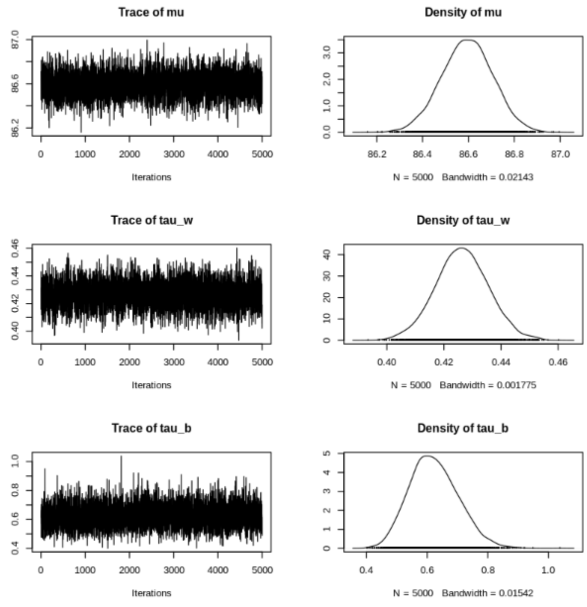


Figure 12: traces of mu, tau_w and tau_b

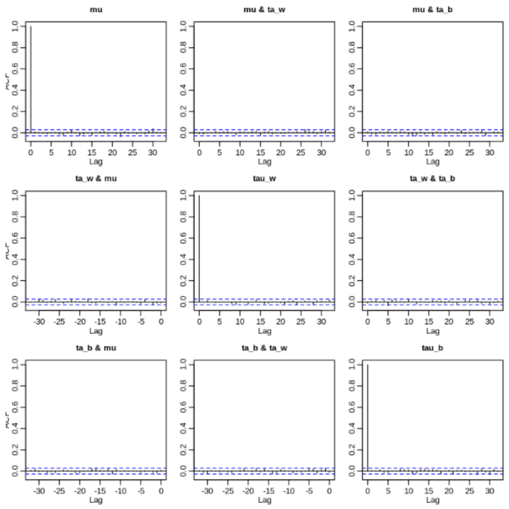
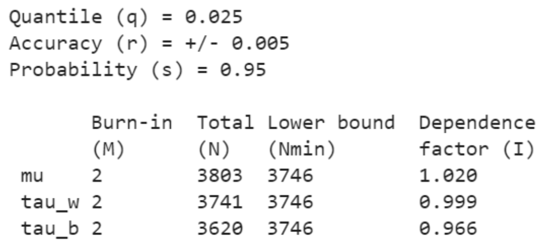


Figure 13: Estimates of autocorrelation

From the two figures shown above we can visually observe the performance of the model, observing these figures, there seems to be no red flag and our model seems to performs satisfactorily.



```
Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

       Burn-in  Total Lower bound  Dependence
       (M)      (N)   (Nmin)       factor (I)
 mu    2        3803  3746         1.020
 tau_w 2        3741  3746         0.999
 tau_b 2        3620  3746         0.966
```

Figure 14:raftery.diag() output

Figure shown above contains the output of diagnostic function to measure the performance of the model, as the values of Burn in and Dependence factor are low we can safely say that the model holds good. Also to address the excessive correlation between the adjacent samples, we performed thinning by selecting every alternate sample from the 10k samples from the sampler.

Table below gives the mean rating of all the regions better than average rating i.e. 86.61

Region	Mean Rating
Trento	88.69180
Verdicchio di Matelica	88.47549
Lugana	88.27322
Cerasuolo di Vittoria Classico	88.26289
Vermentino di Gallura	88.23001
Valdobbiadene Prosecco Superiore	88.17971
Greco di Tufo	88.11511
Etna	88.04068
Vittoria	88.00775
Offida Pecorino	87.81512
Soave Classico Superiore	87.80678
Verdicchio dei Castelli di Jesi Classico Superiore	87.73107
Carignano del Sulcis	87.72868
Fiano di Avellino	87.72080
Maremma Toscana	87.69609
Aglianico del Vulture	87.67271
Rosso di Montalcino	87.66309
Isola dei Nuraghi	87.64588
Vino Nobile di Montepulciano	87.60898
Dogliani	87.54683
Falanghina del Sannio	87.53717
Carmignano	87.53118
Nebbiolo d'Alba	87.51377
Alto Adige Valle Isarco	87.50906
Lambrusco di Sorbara	87.48528
Campi Flegrei	87.45117
Chianti Rufina	87.43271
Alto Adige	87.42322
Primitivo di Manduria	87.39279
Vermentino di Sardegna	87.37315
Valpolicella Classico Superiore Ripasso	87.37118
Cesanese del Piglio	87.36967
Bolgheri	87.31257
Montefalco Rosso	87.31173
Soave Classico	87.26327
Barbera d'Asti Superiore	87.25771
Romagna	87.25416
Irpinia	87.23930
Roero	87.23538
Collio	87.22963
Bardolino	87.22632
Vernaccia di San Gimignano	87.19741
Rosso di Montepulciano	87.12928
Cannonau di Sardegna	87.09068
Barbera d'Asti	87.08953
Chianti Montalbano	87.07472
Molise	87.04809
Barbera d'Alba	87.03826
Morellino di Scansano	87.03044
Asolo Prosecco Superiore	87.02846
Chianti Classico	86.95366

Cerasuolo di Vittoria	86.90529
Veronese	86.90166
Salice Salentino	86.87385
Maremma	86.84456
Colline Novaresi	86.81880
Conegliano	86.81597

Valdobbiadene Prosecco Superiore	
Vigneti delle Dolomiti	86.80881
Monica di Sardegna	86.75263
Rosso del Veronese	86.74765
Orvieto	86.72974

Classico Superiore	
Friuli Colli Orientali	86.71579
Toscana	86.70791
Cirò	86.70112
Valpolicella Ripasso	86.65710
Umbria	86.63765

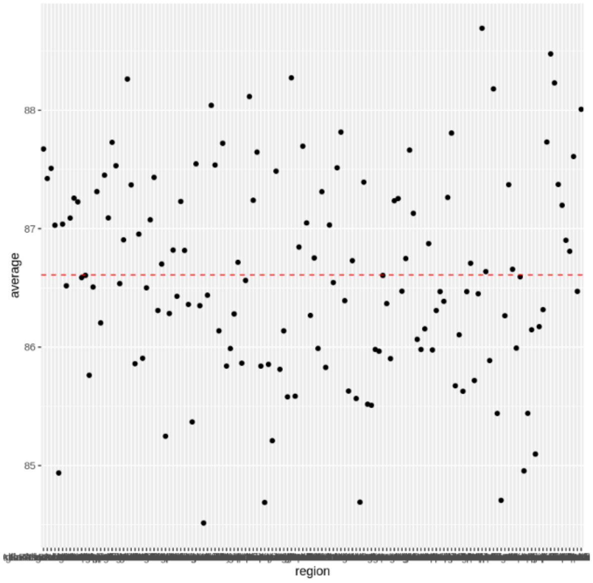
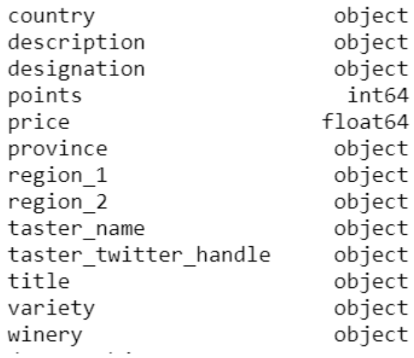


Figure 15:Regions producing better than average Wines

Figure 15 shows all the regions having mean ratings better than the average rating of Italian wines. The red dotted line indicates the value of average rating for the Italian wines.

Q- Build a linear regression model to estimate the points value for wines from the USA. Using simple language, identify which factors are most important in obtaining a good rating.

In order to build a linear regression model for predicting review points, we need to analyze the the type of all the predictors we have in data set.



```
country                   object
description               object
designation               object
points                     int64
price                    float64
province                  object
region_1                  object
region_2                  object
taster_name               object
taster_twitter_handle     object
title                     object
variety                   object
winery                    object
```

Figure 16: Predictors type

From the screenshot shown above we can observe that aside of price all the predictors we have are of categorical type, so in order to build the linear regression model we first need to encode these categorical variables. Also we have textual description of taster's review as a predictor in the data set. Description column can be processed using many different NLP techniques to generate features from this text, however to simplify the process I will be processing this column to generate a sentiment score. I used TextBlob's sentiment analyzer to calculate the sentiment polarity score of every review description. Also from the features above I dropped the country column and columns related to the taster because there were a lot empty rows for these columns. To process other categorical variables, I used ordinal encoding provided by scikit-learn

To test for the dependence between target variable and the predictors there are many statistical tests available. For now we are going to use Pearson's chi-squared test to know which are the most important features to build a regressor to predict review points. Chi squared test is useful to predict dependence of target variable on the categorical variables, as most of our predictors are categorical variables we are using chi squared test also. To make things simpler I converted 'price' variable to a categorical variable by binning the values of this variable. To use the Chi Square test the target variable's distinct values can be considered as label's as there are not many distinct values for the points variable. Chi square tests calculate the Chi square value between target and the feature, features with the best Chi Square values can be considered as important features.

$$\chi^2 = \sum \frac{(o-e)^2}{e}$$

O – number of observations

E – number of expected observation in particular class assuming no relationship between feature and target

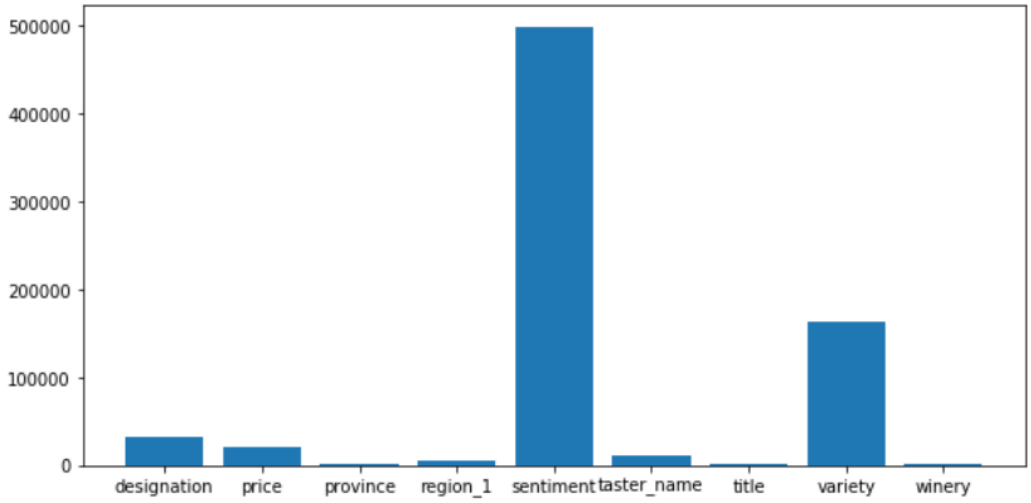
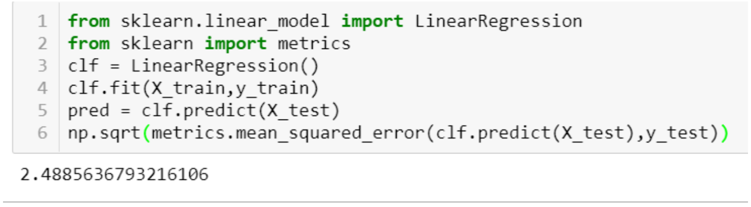


Figure 17: Feature Importance

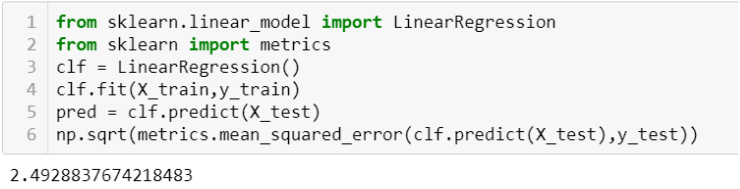
Figure 17 shows the plot of the chi-square test scores of all the predictors. From the figure it can be observed that designation, price, variety,tester_name and sentiment are the most important variables to predict the wine rating.



```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
clf = LinearRegression()
clf.fit(X_train,y_train)
pred = clf.predict(X_test)
np.sqrt(metrics.mean_squared_error(clf.predict(X_test),y_test))
```

```
2.4885636793216106
```

Figure 18: Linear Regression without feature filtering



```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
clf = LinearRegression()
clf.fit(X_train,y_train)
pred = clf.predict(X_test)
np.sqrt(metrics.mean_squared_error(clf.predict(X_test),y_test))
```

```
2.4928837674218483
```

Figure 19:Linear Regression after filtering unimportant features

To test our hypothesis of important features we can build Linear Regression model to predict the points variable. The idea is to build the model with all the feature and without features we are hypothesising to be unimportant

For building the linear Regression model we are using Python's Scikit Learn Library. The model is trained on 90% of data and tested on the remaining 10%. Figure 18 shows the Root Mean Square Error value of the model trained with all the features and Figure 19 shows the same for the model trained only with "price", "sentiment", "variety","tester_name" and "designation". It can be observed from the figures that there is very little difference between the RMSE values for the two variables, hence it can be safely said that the filtered out features have very slight effect on the target variable. To make sure that we have the right features I dropped the "price" variable also which increased the RMSE value to 2.87, which is a significant drop in performance indicating that price is an important feature.

Now we can safely say that "price", "sentiment", "variety" and "designation" are the most important features, however "sentiment" is a feature which we derived from the "description" feature. Therefore it can be said that "description" is also an important feature, however it is an textual feature so we doesn't get much insight that kind of words appearing in this column significantly impacts the rating variable. We can dig deeper to analyse this variable.

Analysing Review Text

As Review is a text variable there are number of features which can be possibly driven from text. There are number of techniques available to derive features from text e.g. Tf-idf, Bag of Words, Count vectorization, Word embeddings etc. However, for now we will be using Tf-idf to derive features from text.

Tf-idf = Term frequency/ Document Frequency

Term frequency – number of times the word appear in document.

Document Frequency – number of documents the word appears in.

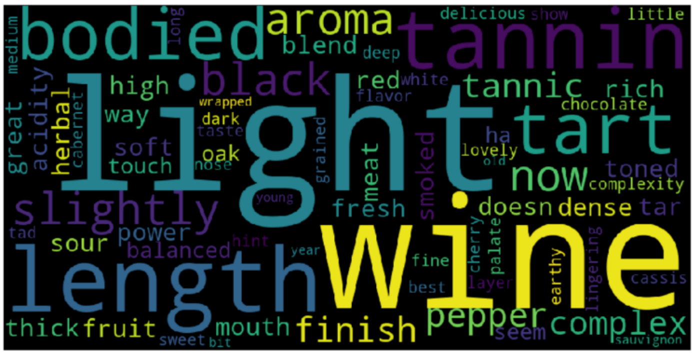


Figure 20: Wordcloud of most important Text features

Word cloud shown on the left provides the top 100 most important textual features which impact the rating of the wine most significantly. The cloud contains words like "finish", "aroma", "light", "tart". These words in the review can be discerning factors for the rating of wine. Most of the words make sense but however still we cant say how these words impact the rating of the wine. We can dig further to analyze which kind of words are the most discerning factors. To analyze this I divided the wines into 4 classes, below average, average, good and best, based on the mean and standard deviation of the whole group. Using these classes we can now build a classifier are check which are the most discerning textual features amongst these classes.

Below Average	Average	Good	Best
puckering, overall, watery, everyday, virginia, lack, tad, chard, cuisine, simple	variation, tapering, keenly, sandy, twice, traffic, popular, botanical, Temecula, clipped	gray, spread, westside, seeing, lengthy, adorned, cement, meyer, cellaring, gamut	exquisite, lengthy, spark, exceptional, sublime, complex, stunning, superb, gorgeous, delicious

Table above shows the most discerning words that appear in the 4 categories using these words many things can be analysed about what feature of wine effects the rating of wine. However, by lightly analysing the text of reviews it can be observed that there are a lot of adjectives used, hence it would make more sense to use bigram features instead of single words.

Below Average	Average	Good	Best
flavor concentration , medium sweet , red blend , short finish , drink dry , medium bodied , soft texture , light bodied , easy drinking , pressed apple	nose bottling , bodied fruit , moderate acidity , delivers enjoyment , fruit touch , drink dry , dry riesling , moderately long , grainy tannin , easy drinking	dark chocolate , wet stone , currant cranberry , meyer lemon , sense balance , lovely sense , nicely balanced , vineyard designate , lengthy finish , black cherry	lingering finish , fine grained , star anise , sea salt , white pepper , long finish , delicious wine , lengthy finish , bay leaf , smoked meat

Table above shows the most discerning bigrams for the four categories. Observing this table a lot of insights can be made about the features which effects the wine ratings. For example by simply glancing through the table red blend and short finish in wines will most negatively impact the wine rating, however long finish and lingering taste will contribute to finest wines. Other insight which can be made is that fruity taste pushes down the wine rating however smoky and spiced taste improves the rating. **From the analysis, it can be concluded that "description", "price", "variety" and "tester_name" are the most important features effecting the rating.**