Ashish Kannur(kannura@tcd.ie)
ID: 19300875

1. 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.

1a. i. Which type of wine is better rated? How much better?

ii. 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?

Sol: Introduction

Dataset Description: The dataset taken in the above scenario is the one available on kaggle.com: https://www.kaggle.com/zynicide/wine-reviews#winemag-data-130k-v2.csv

The dataset is a collection of reviews for a variety of wines from different countries all over the world. Various "variety" of wines from different "wineries" and "regions" of all "countries" are listed in the taken dataset along with the review "description", "rating/points", "price". The dataset can be imported in R using the read_csv() function.

Data analysis: For given problem, we filter out two varieties of wines **Sauvignon Blanc from South Africa and Chardonnay from Chile** which are priced at **15 euros** for analysis.

The columns significant for our analysis are "country", "price", "points", "variety".

To extract data for above problem, we first filter out all the wines which are priced at 15 euros. We convert the class of the 'country' and 'variety' column to factor so that they are treated as index values instead of character values. Then we form 2 separate dataframes: 1. country== "Chile" and variety== "Chardonnay" 2. country=='South Africa' and variety== "Sauvignon Blanc" using the which() function in R. Finally the two separate dataframes are combined using the rbind() function.

Analysis: Here we initially assume as null hypothesis that the means of both variety of wines is equal. As the datapoints are limited here, we conduct T test to test our null hypothesis. Let's check the conditions for T-test :

1. Data should be normally distributed 2. Variance must be same(F-test).

For 1st condition check, we individually plot the distributions of Chardonnay data points, Sauvignon data points and combined data points. We observe that they are normally distributed(except Sauvignon because there are less data points for it in the dataset). Alternatively, we can use the qqnorm function in R to plot these datapoints. If the points roughly appear to be diagonally distributed, then we can say it is normally distributed. Also, by Shapiro Wilk normality test, we can check if the condition of normality is satisfied by the each wine's data points.

For 2nd condition, we use the function var.test(x,y) in R. Here the null hypothesis is that variance of 2 quantities x & y are same. After running this function, we get p-value > 0.05. So, we accept the null hypothesis ,i.e, variances of Chardonnay points and Sauvignon points is same.

As both the conditions of T test are satisfied the Chardonnay and Sauvignon datapoints, we run the t test for both wines' data to check if their means are same(null hypothesis). We get p-value of 0.002.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

Hence, we reject the null hypothesis and can say that means of both the wines are not same. Also,we can perform One Tailed T-test to check which wine's mean is greater.

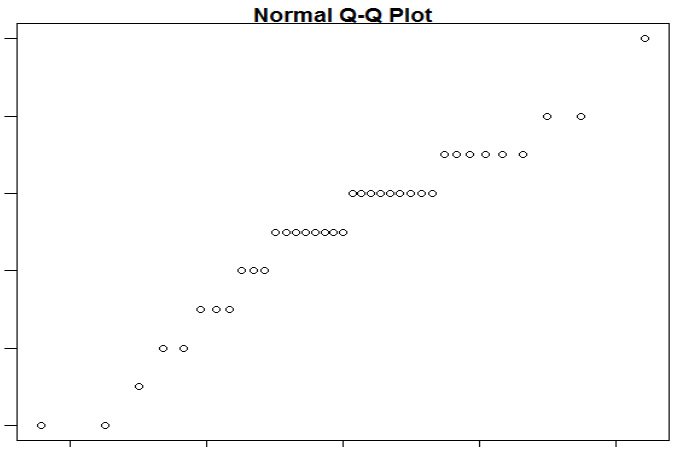

Figure 1: Q-Q plot for Chardonnay

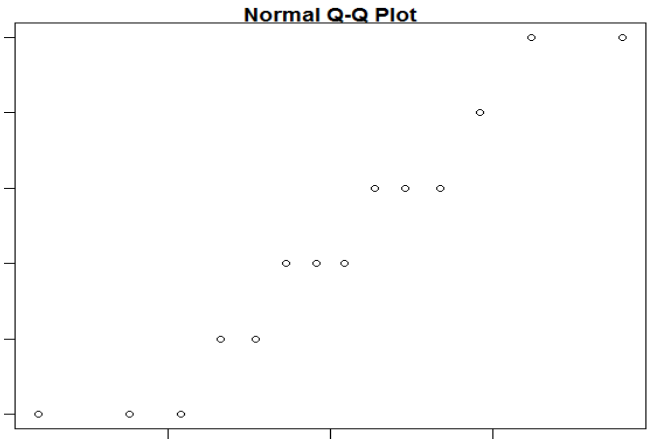

Figure 2: Q-Q plot for Sauvignon Blanc

We then use ggplot to visualize the comparison between the 'points' of both the types of wines. For effective visualisation, we add additional layer of jittered data over the box plot. Please refer figure1.

In the below box plot figure, the brown box plot is for Chardonnay wine datapoints and blue box plot is for Sauvignon Blanc wine datapoints. **Mean points for Chardonnay is 85.08 and mean points for Sauvignon Blanc is 87.21 and difference between the means is 2.13**. Median for Chardonnay box plot is 85 and median for Sauvignon Blanc is 87. Chardonnay box plot is spread from 84 to 86 and Sauvignon Blanc plot is spread from 86 to 88. The quartiles ranges are shown in Figure 2.

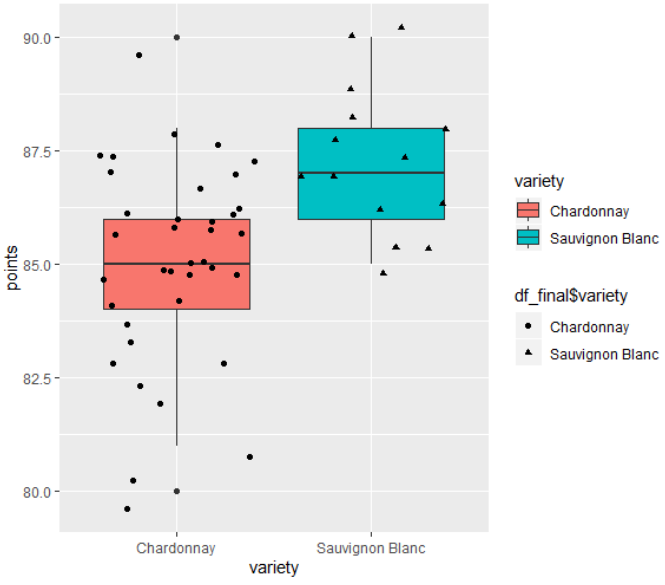

Figure 3: Box plots

```
> tapply(df_final$points, df_final$variety == "Chardonnay", mean)
   FALSE     TRUE
87.21429 85.08108
> tapply(df_final$points, df_final$variety == "Sauvignon Blanc", mean)
   FALSE     TRUE
85.08108 87.21429
> quantile(df1$points)# quartiles of Chardonnay
  0%  25%  50%  75% 100%
  80   84   85   86   90
> quantile(df2$points)#quartiles of sauvignon blanc
  0%  25%  50%  75% 100%
  85   86   87   88   90
```

Figure 4: Means and Quartiles

Now for modelling this difference in the means, we use Gibbs sampling. Code for the Gibbs sampler is provided in the Appendix(compare_2_gibbs function). Gibbs sampler basically compares the means between the two types of wines. We use the combined dataframe of the two wines here.The inputs to the function are the "points" ,"variety" as factor, initialised parameters for the model such

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

as mu0=50, tau0=1/400,gamma0=1/400,a0=1,b0=50. Prior distribution is unknown to us, we take a wide standard deviation and initial mu0 to cover maximum observations. With this initial state, we run a chain to iteratively build successive states where each state is dependent on the previous state. We run the chain enough times to build more correlation. Below plot is the Trace and density plot which is used for analysing the mixing of the chain. As expected, we are getting purely random traceplots. We can see that mu, del and tau follow normal posterior distributions.

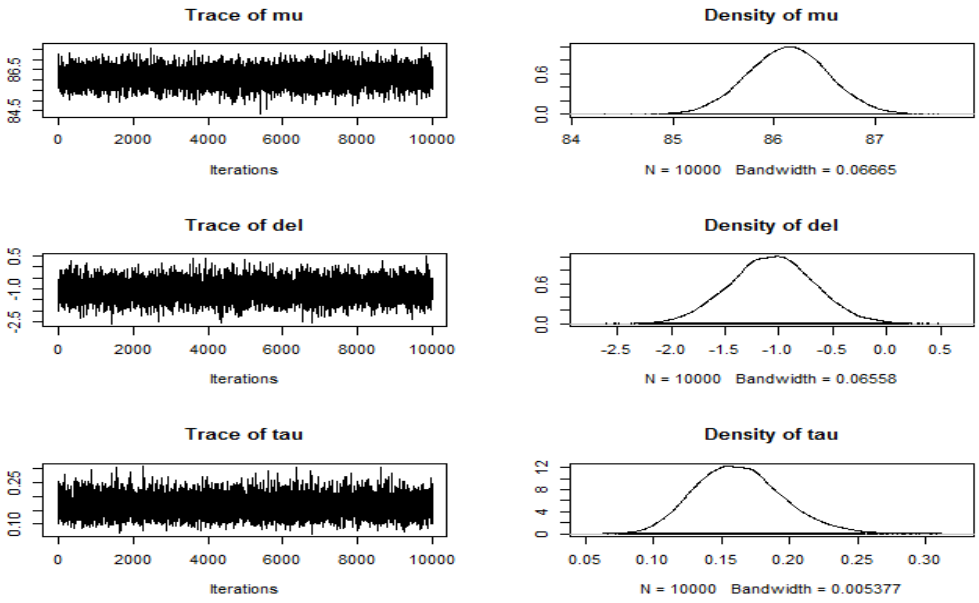

Figure 5: Trace and Density Plots

We further generate new samples for both the wines using the Gibbs model (which was generated using both wines original data)

```
y1_sim <- rnorm(10000, fit[, 1] + fit[, 2], sd = 1/sqrt(fit[, 3]))
y2_sim <- rnorm(10000, fit[, 1] - fit[, 2], sd = 1/sqrt(fit[, 3]))
```

We use the above samples generated to compare the means of both the wines and suggest which wine performs better. Below figure shows the distribution of differences between the simulated samples [y_sim_diff = y1_sim - y2_sim]. Distribution varies from -15 to +10 here

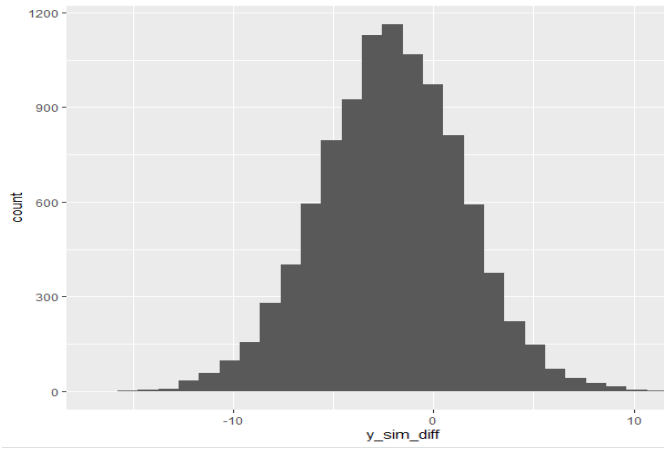

Figure 6: Simulated wine samples differences distribution

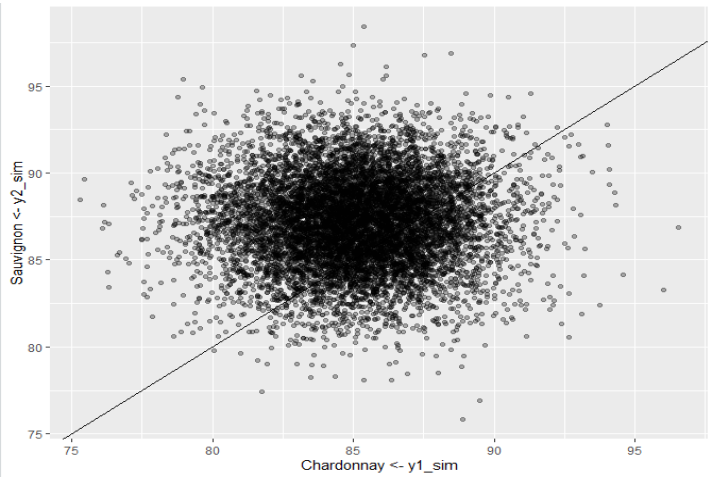

Figure 7: Plot of simulated samples

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

Conclusion: The difference in the means of points of Chardonnay Chile and Sauvignon Blanc wines according to the original dataset was **2.13** and after simulating the samples by Gibbs sampler, the difference between points of Chardonnay Chile and Sauvignon Blanc is **2.11**.

1a.i) As part of **1a.i**, Figure 7 shows the plot of (Sauvignon points, Chardonnay points) and the separation line shows more points distributed above it which tells us that South African Sauvignon wine is better rated than Chilean Chardonnay by **2.10 points** (calculation shown in the snipped below) and

1a.ii) **South African Sauvignon Blanc** would be better than **Chilean Chardonnay** with probability of **approx. 72%** as calculated below:

```
> #How much better is the wine:
> difference  = mean(y2_sim - y1_sim)
> print('How much better is the wine')
[1] "How much better is the wine"
> print(difference)
[1] 2.109452
> mean(y2_sim > y1_sim) ##probability
[1] 0.7226
```

1b. 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.

Sol:

Data subset selection and Cleaning: As per the constraints given in the problem, we filter out the data with country == "Italy" and price less than 20 euros and create a new dataframe . Then only the required columns are retained , i.e. , country, price, points,variety,region_1 using the sqldf function. NA and duplicate values are removed using the na.omit() and unique() functions. As the analysis needs to be performed only on the regions having atleast four reviews, we filter out the data further by considering only the records having the occurrences of region_1 more than 3 times. Also, the class of region_1 is converted to factor so that it is treated as index and not character.

The average of the total points of the wines of all regions of Italy is **86.479**.

Total number of unique regions in the final dataset is 152.

We then plot boxplots for each region as shown below in figure 8 using ggplot. We can see the medians varying for each region and outliers in the graph. We can therefore compare the points or ratings of Italian wines of different regions.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

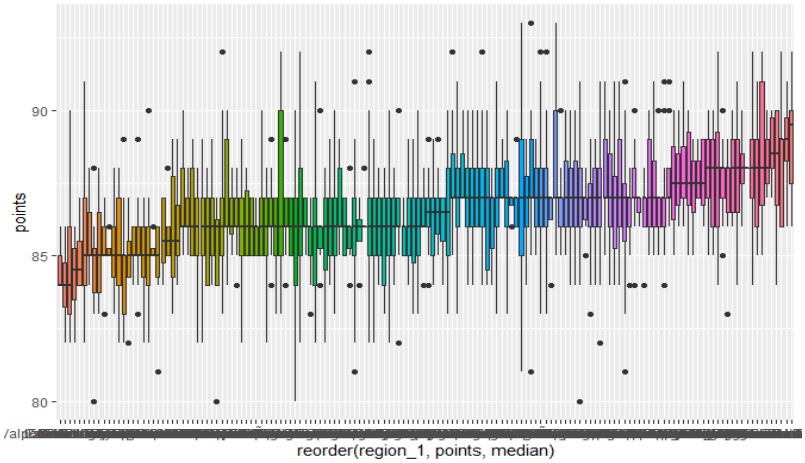

Figure 8:Box plots for points of each region

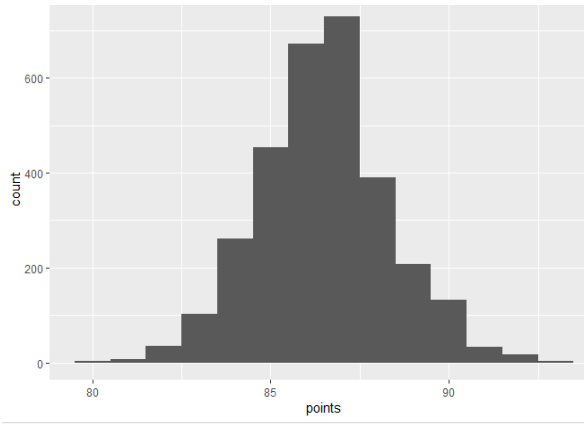

Figure 9: Points frequency distribution

Figure 9 shows distribution of points along with its frequency of occurrence. Roughly, maximum points that are present in the dataset are in the range of 85 to 88. Figure 10 shows wide range of samples of region_1 and corresponding count of each sample(region). Figure 11 shows the distribution of Mean score(mean of points) for sample size of each region. We observe that, regions with sample size in the range of 0 to 25 have highest as well as lowest mean scores. In other words, extreme mean scores are associated with regions of small sample sizes.

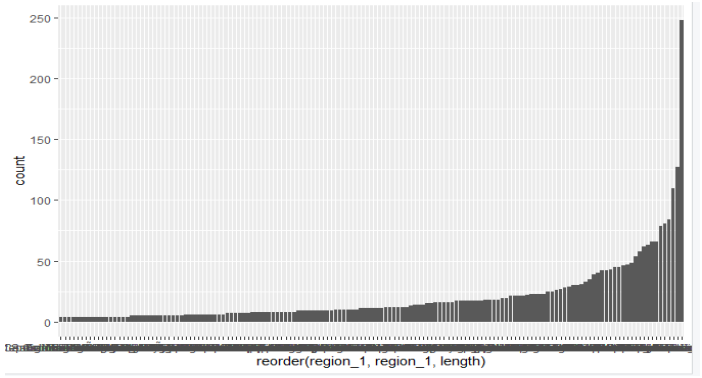

Figure 10: Count of region_1

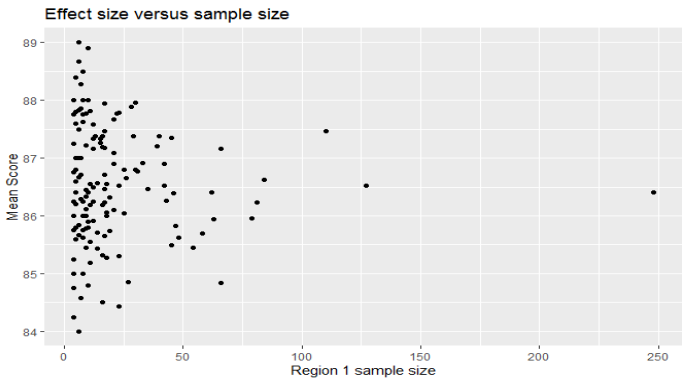

Figure 11: Mean scores for each region's sample size

Figure 11 also shows that, the differences in the means of various regions is due to variability in the samples. Thus we model these mean scores as resulting from common populations(regions). In this way we take into account the variation within sample size of every population(region) and then more effectively estimate he differences between the regions. We use Gibbs sampling method for modelling posterior.

The inputs to the function compare_m_gibbs are the "points" ,"region_1" as factor, initialised parameters such as mu0=50, tau0=1/400,gamma0=1/400,a0=1,b0=50,alpha0=1,beta0=50. Std deviation is taken to be large initially to accommodate large observations. Following are the parameters calculated in the gibbs function: μ -the overall mean of all regions, τb -the inverse of variance between the regions, τw –the inverse of variance within the regions, θm- mean point of any region *m* . These model parameters are updated with every iteration. We run this iteration 5000 times.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

The Gibbs model(compare_m_gibbs function) returns 2 objects: params and theta. Theta will hold sample sets for 150 parameters(regions). Params will hole the parameters : μ, τb and τw. Below is the way we visualise all the group mean parameters θ,i.e, to visualise ˆθ1, . . . , ˆθm.

```
theta_hat <- apply(fit2$theta, 2, mean) ## get basic posterior summary
#df4 <- data.frame(theta_hat,names(theta_hat) <- 1:175) ## keep track o
names(theta_hat) <- df3$fit2.a
sorted <- sort(theta_hat, decreasing = TRUE) ## which regions did best
```

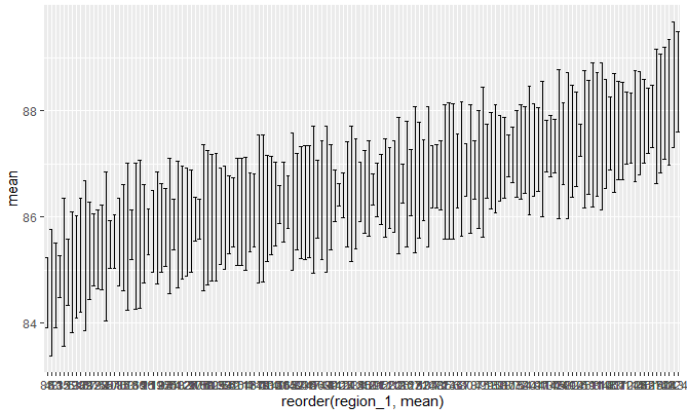

Figure 12: Upper/lower bounds of theta for all regions samples

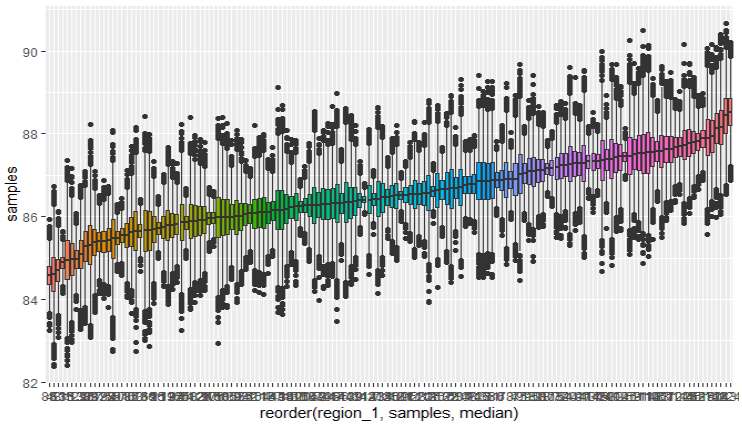

Figure 13: Boxplots for each region using generated

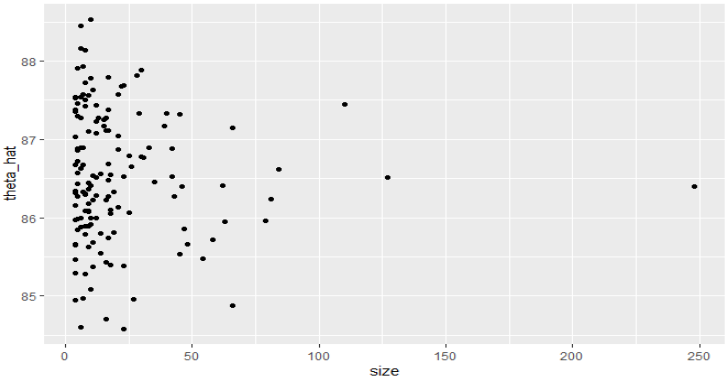

Figure 14: Mean score of regions vs sample sizes based on samples generated

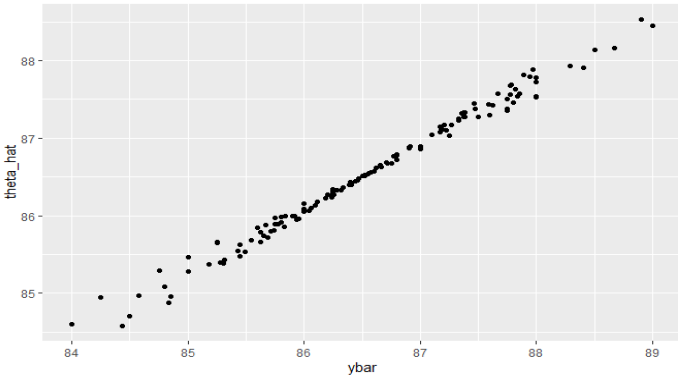

Figure 15: Parameter estimate vs Sample means

Conclusion:

1b. To find **how many Italian regions have wine ratings/points greater than the total average points of all Italian regions(86.479),** we execute below code:

```
gt_a<-  sorted > averagePoints
View(names(sorted[gt_a]))
```

When executed we get in total **73 regions** which have ratings/points greater than the total Italian wines points average. **Following are the regions** which have better wines than overall average wine:

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

```
> cat(names(sorted[gt_a]),sep=",",fill=TRUE)
Trento,Vermentino di Gallura, Cerasuolo di Vittoria Classico,Verdicchio di
Matelica,Vittoria,Carignano del Sulcis,Valdobbiadene Prosecco Superiore,Lugana,Etna,Fiano
di Avellino,Soave Classico Superiore,Rosso di Montalcino, Maremma Toscana,Aglianico del
Vulture,Greco di Tufo,Offida Pecorino, Verdicchio dei Castelli di Jesi Classico Superiore,
Vino Nobile di Montepulciano,Sant'Antimo,Alto Adige Valle Isarco,Sardinia,Isola dei
Nuraghi,Falanghina del Sannio,Alto Adige,Primitivo di Manduria, Nebbiolo
d'Alba,Dogliani,Chianti Rufina,Montepulciano d'Abruzzo Colline Teramane,Vernaccia di San
Gimignano,Valpolicella Classico Superiore Ripasso,Soave Classico,Campi Flegrei,Barbera
d'Asti Superiore,Vermentino di Sardegna,Carmignano,Lambrusco di Sorbara,Montefalco
Rosso,Bolgheri,Rosso di Montepulciano,Collio,Roero,Irpinia,Bardolino,Cannonau di
Sardegna,Romagna,Monica di Sardegna,Asolo Prosecco Superiore,Barbera
d'Asti,Molise,Morellino di Scansano,CirÃ²,Barbera d'Alba,Veronese,Cerasuolo di
Vittoria,Colline Novaresi,Maremma,Conegliano Valdobbiadene Prosecco Superiore,Vigneti
delle Dolomiti,Rosso del Veronese,Friuli Colli Orientali,Valpolicella Ripasso, Cerasuolo
d'Abruzzo,Salice Salentino,Orvieto Classico Superiore,Chianti Classico,Bardolino
Classico,Castel del Monte,Chianti Colli Senesi, Bardolino Chiaretto,Prosecco di
Valdobbiadene,Moscato d'Asti,Toscana.
```

2. Use model-based clustering methods to categorise the wines from the USA based on price and points rating. Can you identify any clusters that are good value for money?

Sol. Introduction :

Cluster analysis is used to find subgroups of observations in a dataset. Clustering basically aims at grouping similar items together in a dataset. This technique works on finding relation between the observations and does not have any target variable. Hence this is an unsupervised learning. Here, we aim to cluster the data of wines in USA with respect to its features ,i.e, points/ratings and price. We then wish to identify those clusters which are good value for money ,i.e, good rating and less price. This method is called Model based Clustering.

Algorithms like the k means and hierarchical clustering are the heuristic methods which perform clustering based on the input data. They do not consider measurements of probability or uncertainty of assigning clusters. Model based clustering considers the probability of an observation belonging to a cluster while assigning. It also identifies the optimal clusters automatically.

For analyzing the data to categorize the wine data of USA and to cluster them on the basis of their price and points, we use Mclust (an R package which follows Gaussian mixture model approach). Finite Gaussian models are predicted by finding the posteriors or the maximum likelihood of the parameters in the model. This statistical method is called Expectation maximization.

Data Insights:

Data is first loaded in R in the form of a dataframe. Then, only the data for US is filtered out. NA values are omitted. Required columns are selected and rest ones are discarded.

We first plot the price vs points for the wine data as shown below.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

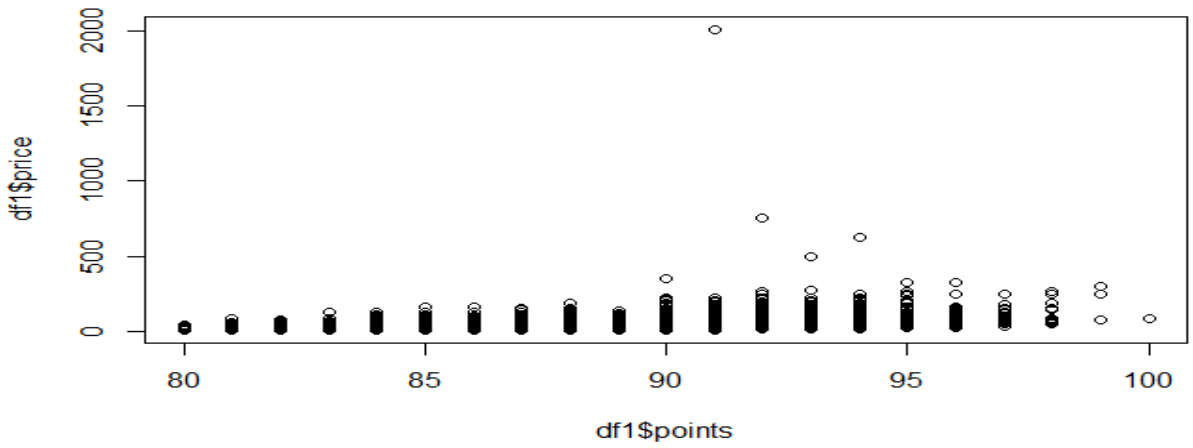

Figure 16: Original data Price vs points

The plot above depicts that there exists no correlation between the two quantities price and points of the given data. Additionally, we can clearly see the presence of outliers and they definitely affect the clusters formation. So we remove the outliers before using the Gaussian Mixture models approach(Mclust).

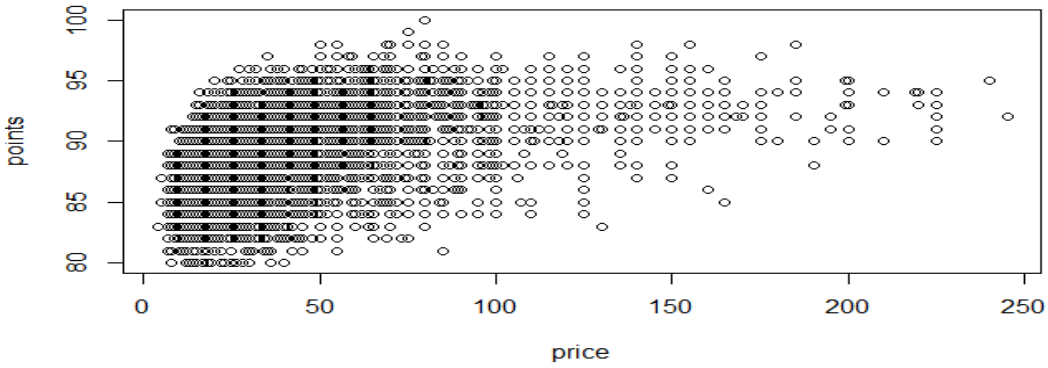

Figure 17: Plot of Points vs Price after the removal of outliers

The data looks clean and ready for cluster analysis.

Analysis: Now the data is fit using the Mclust package to obtain a model.

```
fit <- Mclust(df2)
```

This method by default takes the values to find the finite Gaussian model using the Expectation maximization algorithms. The geometrical attributes of the clusters like the shape, volume and orientation are represented in the form of covariance matrix. Ellipsoidal, spherical and diagonal are the constraints and combinations of the models. Mclust now identifies the model out of the 14 that best describes the data. This model selection is done using the BIC(Bayesian information criterion). More the complexity of the models , larger the number of clusters(penalization). Lower the BIC, better the fit. But at the sametime, it may overfit the model. The number of clusters ,i.e, mixture components, is set to use 1-9 clusters by default during model fitting.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

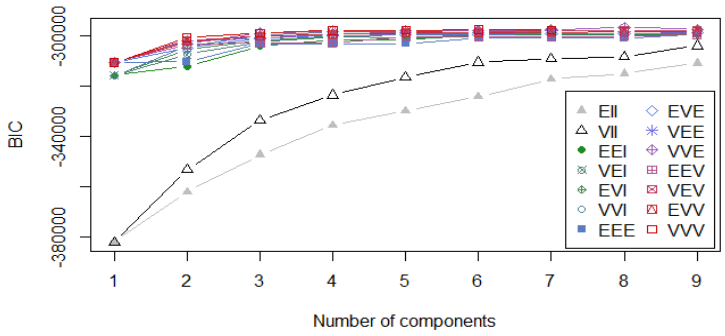

Figure 18: Plot of first fit BIC vs number of components

In the above plot , we can see the co-variance structures and number of clusters and their associated BIC values. The line graphs or the co variance structures in the above plot shows the BIC values for different groups. Below is the summary of the data with the BIC values. It shows that VII and EEI models perform poorly as compared to other models

```
> fit$BIC
Bayesian Information Criterion (BIC):
        EII       VII       EEI       VEI       EVI       VVI       EEE
1 -382089.9 -382089.9 -315827.7 -315827.7 -315827.7 -315827.7 -310720.9
2 -362130.4 -353264.3 -312150.1 -307510.9 -305477.4 -304271.5 -310369.6
3 -347321.5 -333745.4 -304290.0 -302443.4 -301836.9 -299885.2 -303383.7
4 -335612.5 -323762.2 -301962.4 -300652.5 -300298.9 -298710.7 -303377.9
5 -329962.5 -316599.2 -301520.3 -299514.2 -301067.6 -298366.6 -303259.1
6 -324237.1 -310675.6 -299836.7 -299192.0 -299461.0 -298232.8 -300952.7
7 -317295.3 -309142.4 -299725.5 -298226.1 -299363.1 -297889.9 -300979.8
8 -315095.2 -308557.6 -299757.9 -297995.2 -299403.8        NA -301012.5
9 -311052.8 -304149.4 -299775.3 -298020.8 -299434.1        NA -299605.7
        EVE       VEE       VVE       EEV       VEV       EVV       VVV
1 -310720.9 -310720.9 -310720.9 -310720.9 -310720.9 -310720.9 -310720.9
2 -304843.8 -302195.6 -303407.2 -303360.6 -301880.0 -302418.7 -300925.7
3 -301251.7 -301122.4 -299140.0 -302962.0 -300444.0 -300289.8 -299032.9
4 -299977.9 -299605.5 -298329.1 -302772.1 -299431.3 -299561.9 -298121.9
5 -299894.2 -298732.9 -298046.7 -300529.0 -298629.0 -299131.8 -297940.1
6 -299608.5 -298981.9 -298007.4 -300555.8 -298721.2 -298983.4 -297647.5
7 -299243.0 -298106.5 -297679.5 -300598.2 -298149.0 -298838.1 -297765.0
8 -299285.1 -298500.3 -296709.9 -300320.9 -297894.3 -298534.4        NA
9 -299062.6 -297861.4 -297606.4 -299697.1 -298045.0 -298461.5        NA

Top 3 models based on the BIC criterion:
    VVE,8     VVE,9     VVV,6
-296709.9 -297606.4 -297647.5
```

Figure 19: Description of first fit BIC

By looking at the above BIC description and analysis, the top three models suitable for this data are (VVE,8),(VVE,9),(VVV,6). The model (VVE,9) shows the highest BIC value among the 3 models. The family of the model(general, spherical or diagonal) is decided by the shape, volume and orientation. Thus we can say that 'general' model fits the given wine data perfectly.

Below shown plot in Figure 20 depicts which observations for first 'fit' are assigned which clusters ,i.e, it shows the membership of each datapoint. Figure 21: shows the uncertainty levels of each observations.

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

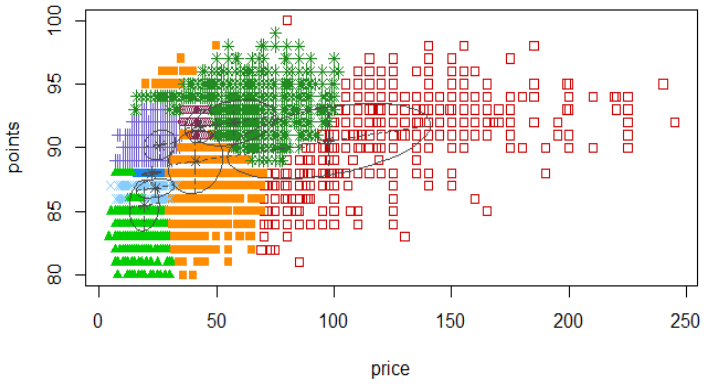

Figure 20: first fit Classification plot

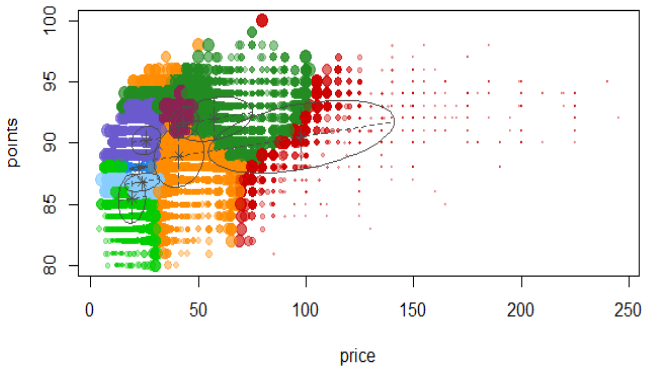

Figure 21: first fit Uncertainty plot

Now we run the next bestfit model by choosing the VVE,9 ,i.e, G=9 and modelNames="VVE". Below are the code, Classification, Uncertainty graphs for the same

```
#####################################
Bfit1 <- Mclust(df2, G= 9, modelNames= "VVE")


plot(Bfit1, what = "classification")
plot(Bfit1,what = "uncertainty")
plot(Bfit1, what = "BIC")
plot(Bfit1, what = "density")

table(Bfit1$classification,Bfit1$uncertainty)
summary(Bfit1)
```

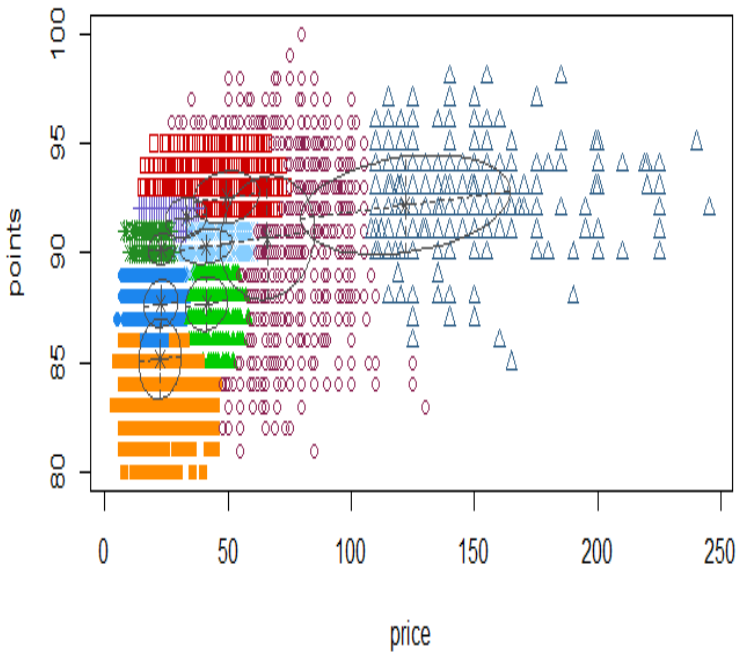

Figure 20: Classification plot

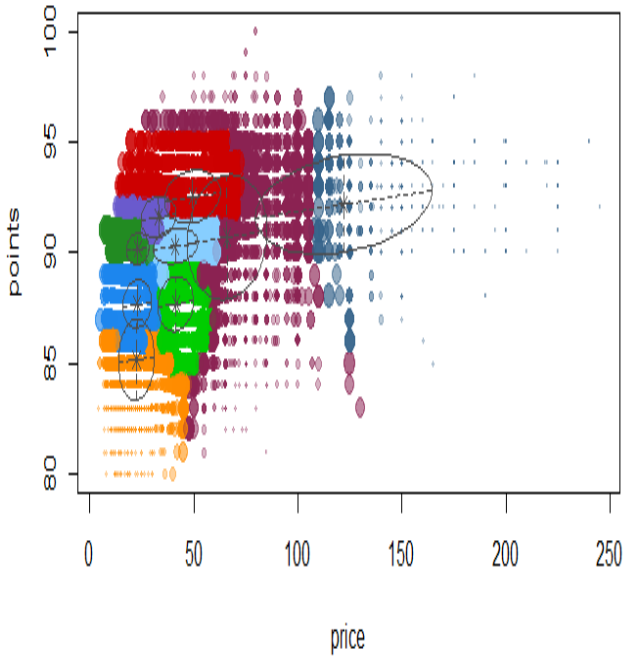

Figure 21: Uncertainty plot

Ashish Kannur(kannura@tcd.ie)
ID: 19300875

Conclusion: Below is the summary of the final model obtained:

```
> summary(Bfit1)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 9 components:

 log-likelihood     n df       BIC       ICL
      -148670.1 22362 45 -297790.9 -317167.3

Clustering table:
   1    2    3    4    5    6    7    8    9
4992 3727 2496 1187 1923 3620 2420 1564  433
```

The above summary shows that model (VVE,9)(general) is the optimal model for clustering. It has BIC = (-297790.9) and this model identifies 9 clusters with cluster 9(C9) being the most compact one and cluster1(C1) having most number of points.

For identifying the best cluster as good value for money, we need to look for cluster that has less price and high points. **By looking at figure 20(classification plot), we can tell that clusters marked in 'Red', 'Violet' and 'Light Sky Blue' are the clusters that are good value for money as they represent wines of USA which are well rated and are low priced as well.**