Data handling and decision making in wine data

~ OUTLINE ~

- Analytical objectives
- Assumptions
- Data Analysis Process
- Data Sanity Checks
- Descriptive Analysis and Findings
- Predictive Analysis and Findings
- Cluster-based Analysis and Findings
- Result
- Conclusion

Introduction

To begin with, a relation between different variables and the pricing strategy of the Burgundy Sip company is meant to be defined and established.
Moreover, many factor is required to be taken into an account for determining the most suitable pricing strategy.
Indeed, some factors associated with the wines such as rating and indubitably have a positive impact on their popularity and as a result on their demand on market.
Through analyzing the proved data set exploiting r language features, the analytical team shall retrieve and achieve the desired results in regard to the pricing strategy.
In addition, visualization functions shall be use for displaying the relationship between various variable. The relationship should be established for determining the objectives of the whole projects.
Relations amongst the variables assist the data analytics team to **devise the most suitable pricing strategy.**
Drawing a conclusion in regard to each wine brand and its foreseeable price is the main objective of the process.

Data Understanding

Item 1. Analytical Objectives

Item 2. Assumptions

Item 3. Data Analysis Process

Analytical Objectives

1. Retrieving data regarding the wines which has made the most sales globally.

2. Which wine age is more popular with users based on rating.

3. Which harvest region produces the better wine and consequently higher price.

4. Brand names of each wine and their impact on wine price.

Goals

- **Increase specific products**
- **Increase customer satisfaction**
- **Focus on specific region**
- **Improve sales strategy**

Assumptions

1. Lower ACD leads to a better wine and therefore higher price and popularity.

2. Name of the wine irrespective of its quality is an important factor (more famous higher the price)

3. Wines that are more aged up are costlier compared to ones that are harvested more recently

4. Lowers rsg often means higher price

5. Number of testers and alcohol level does not have a significant effect on pricing

6. BD does not have a noticeable effect on pricing

Data Analysis Process

Data Understanding	Define the Objectives	Data Cleaning	Data Modeling/Exploratory Analysis	visualization	Validation
.Primary information gathering .Data discovery & characteristics .Setting the goals	.Evaluate the analytical Process .Determine data processing objectives	.Categorizing data based on defined factors .Treat Duplicates .Treat Outliers .Treat Missing Values .Rendering the merged and final data	.Determine important variables .Explore relationship between variables .Providing a summary of the data structure .Descriptive statistics .Predictive analysis .Build Model	.Graphs .Determine best method to demonstrate insights based on analysis .Communication results .Provide recommendation	.Evaluate results .Review process .Identify Future steps: (draw a conclusion)

Data Cleaning

Step 0. Variable Analysis

Step 1. Treat Duplicates

Step 2. Treat Outliers

Step 3. Treat Missing Values

Step 0. Variable Analysis

Step 0-1. Recognize the missing values as NA.

There are 3 types of missing values.

```
BurgundySip <- read.csv("BurgundySip.csv", na.strings = c("", "NA", "N.V."));
```

Step 0-2. Transform some Character variables into Numeric.

Unexpectedly RSG, AL and DN are recognized as character because of including blank spaces.

```
BurgundySip$RSG <- as.numeric(gsub(" ", "", BurgundySip$RSG));
```

Step 0-3. Transform some variables into Factor.

Because YR, REG and TP are useful as factor.

```
BurgundySip$REG <- as.factor(BurgundySip$REG);
```

Step 0. Variable Analysis

Step 0-4. Check the structure of BurgundySip dataset

```
str(BurgundySip);
```

```
'data.frame':	7500 obs. of  14 variables:
 $ SN  : chr  "W4339-20718" "W9737-31436" "W8398-63402" "W4418-44312" ...
 $ NAME: chr  "A Coroa" "Aalto" "Aalto" "Aalto" ...
 $ WINE: chr  "200 Cestos Godello" "Blanco de Parcela" "PS (Pagos Seleccionados) Ribera del Duero" ...
 $ YR  : Ord.factor w/ 112 levels "1910"<"1911"<..: 111 110 102 106 107 108 109 110 103 107 ...
 $ REG : Factor w/ 76 levels "Abona","Alella",..: 73 58 58 58 58 58 58 58 58 58 ...
 $ TP  : Factor w/ 21 levels "Albarino","Cabernet Sauvignon",..: NA 21 12 12 12 12 12 12 12 12 ...
 $ RT  : num  4.06 4.21 4.31 4.43 4.32 4.37 4.34 4.32 4.09 4.31 ...
 $ NUMR: int  33 80 2207 2858 4411 3383 3239 1108 2844 1884 ...
 $ PR  : num  23.7 41.9 64 172.5 78.7 ...
 $ BD  : int  NA NA 5 5 5 5 5 5 NA 5 ...
 $ ACD : int  NA NA 3 3 3 3 3 3 NA 3 ...
 $ RSG : num  9.99 8.94 8.22 7 NA ...
 $ AL  : num  11.1 11.6 11.9 11.4 NA ...
 $ DN  : num  0.996 0.996 0.995 0.996 NA ...
```

Step 0. Variable Analysis

Step 0-5. Check the summary of BurgundySip dataset

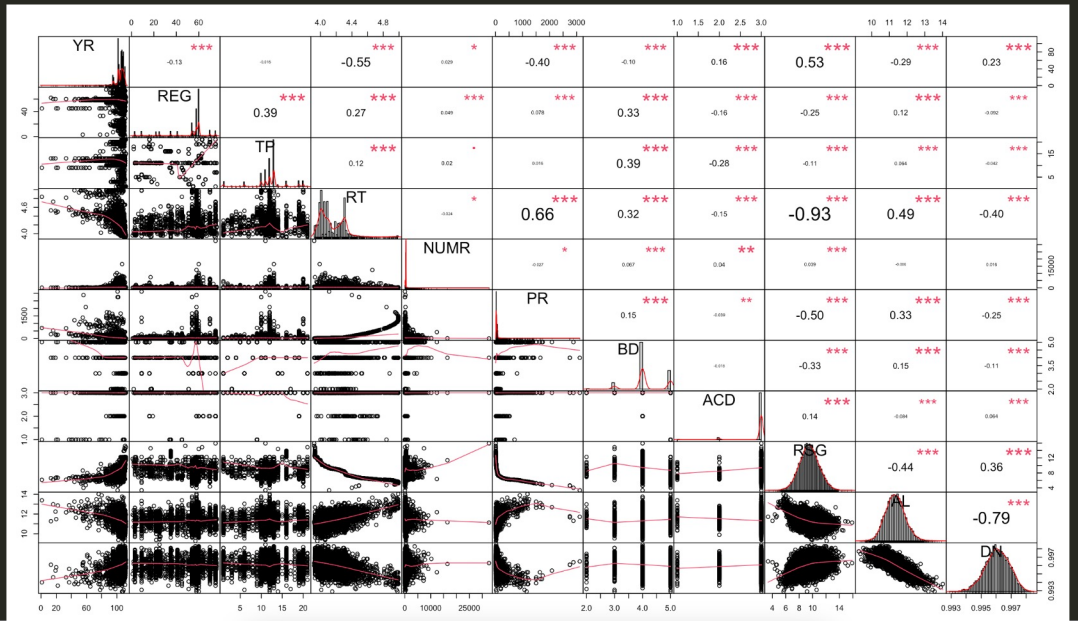
```
summary(BurgundySip);
```

```
                REG                         TP                    RT              NUMR             PR              BD
Rioja                :2433   Rioja Red               :2354   Min.    :3.920   Min.    :   25.0   Min.    :   4.99   Min.    :2.000
Ribera del Duero:1409   Ribera Del Duero Red:1406   1st Qu. :4.020   1st Qu. :  389.0   1st Qu. :  18.90   1st Qu. :4.000
Priorato             : 677   Red                     : 848    Median :4.100   Median :  404.0   Median  :  28.53   Median :4.000
Toro                 : 297   Priorat Red             : 666    Mean   :4.154   Mean    :  451.1   Mean    :  58.21   Mean   :4.158
Vino de Espana  : 261   Tempranillo             : 291    3rd Qu. :4.270   3rd Qu. :  415.0   3rd Qu. :  51.35   3rd Qu.:5.000
(Other)              :2380   (Other)                 :1354    Max.    :4.990   Max.    :32624.0   Max.   :3119.08   Max.    :5.000
NA's                 :43     NA's                    : 581    NA's    :9                          NA's          :58   NA's    :1169
```

Step 0. Variable Analysis

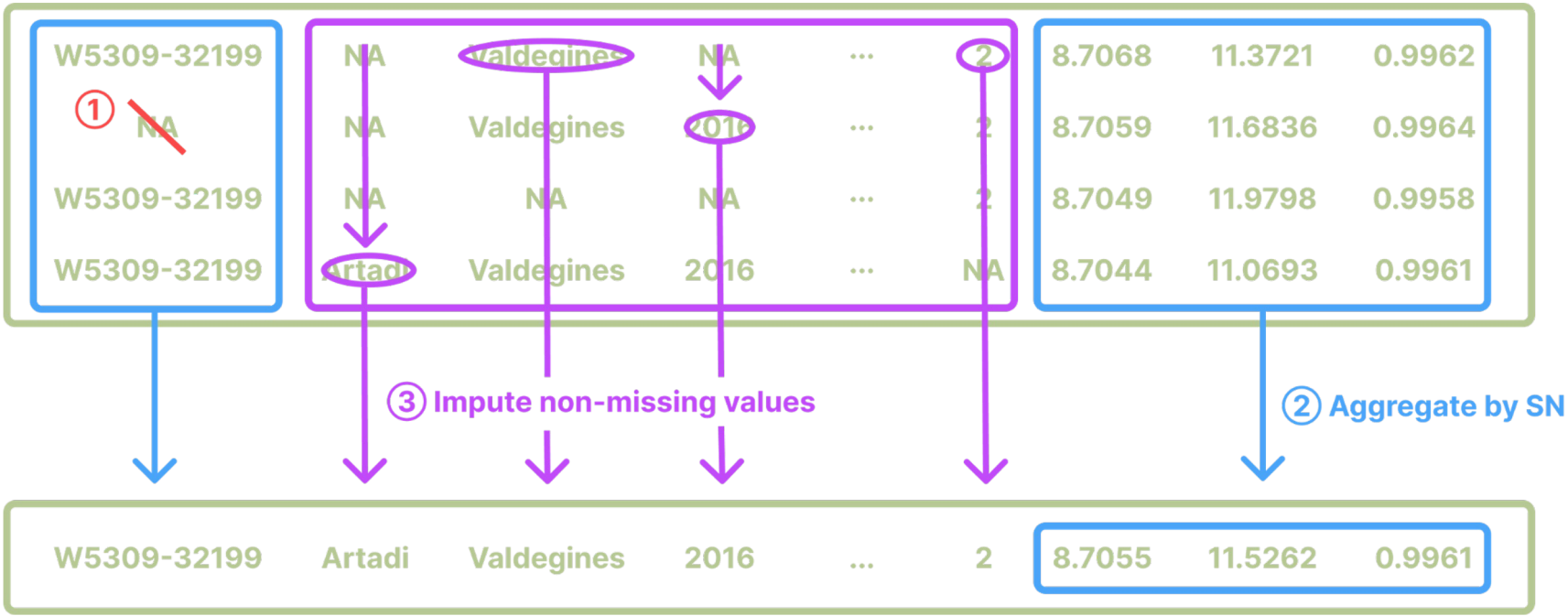
Step 0-6. Check the correlations of all numeric variables in BurgundySip dataset

```
chart.Correlation(sapply(BurgundySip[, -c(1, 2, 3)], as.numeric));
```

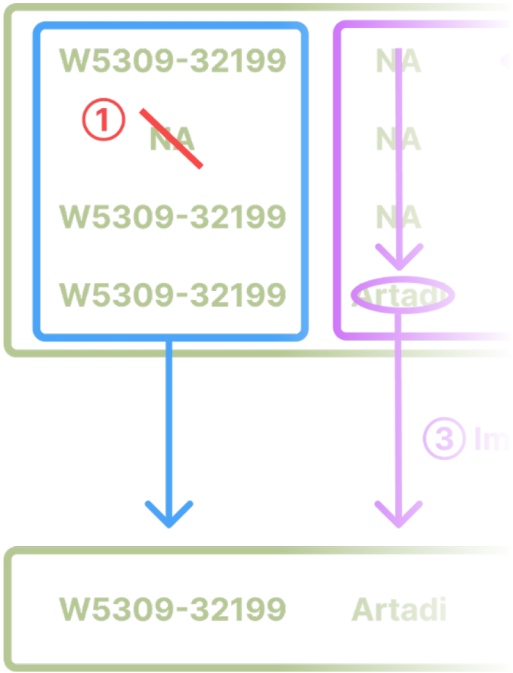


Step 1. Treat Duplicates

Strategy



Step 1. Treat Duplicates



Step 1. Handle missing values of SN

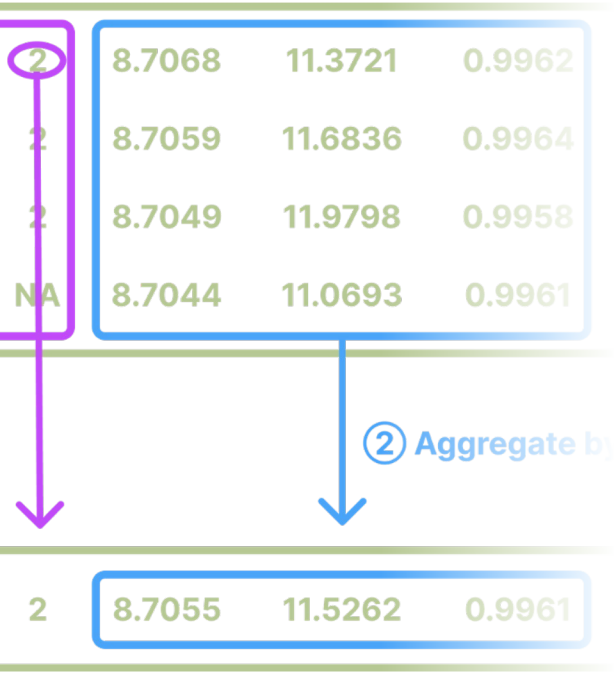
SN is unique for each wines. And every observations which have same SN are duplicated with other variables excepting for RSG, AL and DN.

We remove the duplicated observations by SN. And fortunately there are only two SNs are missing.

```
sum(is.na(BurgundySip$SN));
BurgundySip$SN[c(727, 730, 733)] <- BurgundySip$SN[c(726)];
BurgundySip$SN[2072:2115] <- BurgundySip$SN[c(2071)];
sum(is.na(BurgundySip$SN));
```

```
[1] 47
[1] 0
```

Step 1. Treat Duplicates



Step 2. Aggregate RSG, AL and DN by SN

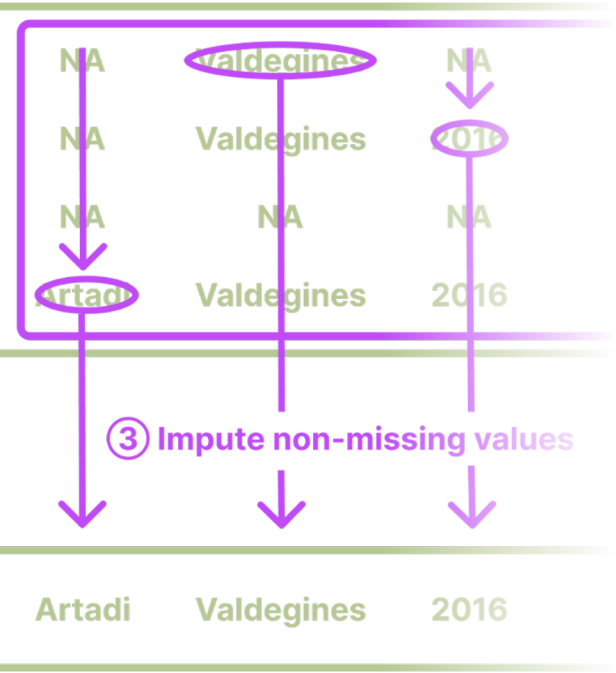
These 3 variables are slightly different by each duplicated observations. We can select every aggregate methods such as max(), min(), mean() and median().

We use the average for aggregating the records.

```
RemovingDuplicated <- aggregate(cbind(RSG, AL, DN) ~ SN,
                                data = BurgundySip,
                                FUN = mean);
RemovingDuplicated;
```

	SN	RSG	AL	DN
1	W1001-62187	9.676400	11.69670	0.9956000
2	W1003-83159	9.385200	11.32470	0.9954000
3	W1007-31751	10.637900	11.48690	0.9954000
4	W1008-45913	11.854237	11.02800	0.9963800

Step 1. Treat Duplicates



Step 3. Impute other variables with non-missing values

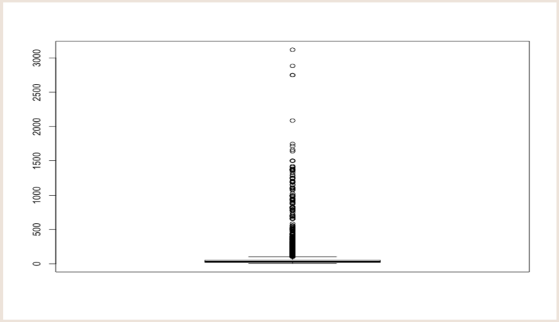
We want to merge other variables (excepting for SN, RSG, AL and DN) into the aggregated table. But by entering values appropriately, missing values can be efficiently removed.

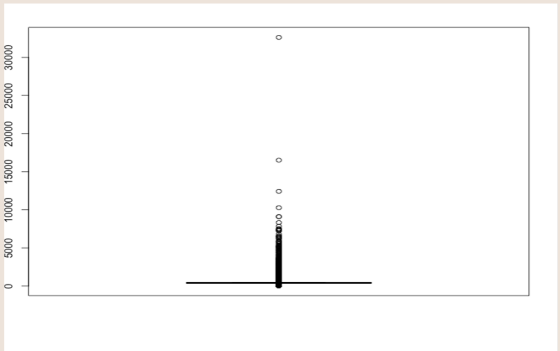
We return the first non-NA value of the NAME column of Observations with the corresponding SN.

```
for (rowIndex in 1:nrow(RemovingDuplicated)) {
    Duplicated_SN <- subset(BurgundySip,
                    SN == RemovingDuplicated[rowIndex, "SN"]);
    RemovingDuplicated$NAME[rowIndex] <-
                    Duplicated_SN[!is.na(Duplicated_SN$NAME), "NAME"][1];
    RemovingDuplicated$WINE[rowIndex] <-
                    Duplicated_SN[!is.na(Duplicated_SN$WINE), "WINE"][1];
...
}
```

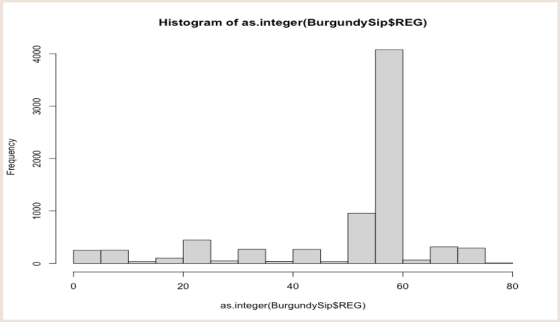
Step 2. Treat Outliers

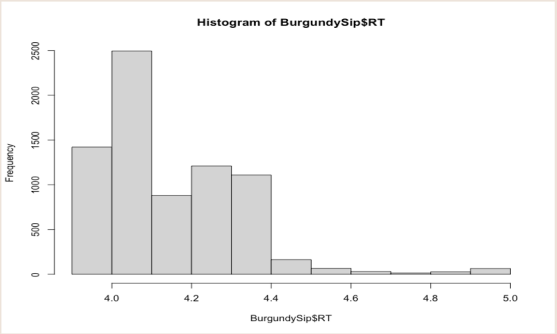
Global Outliers



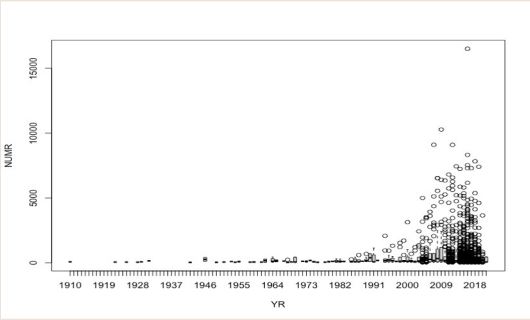


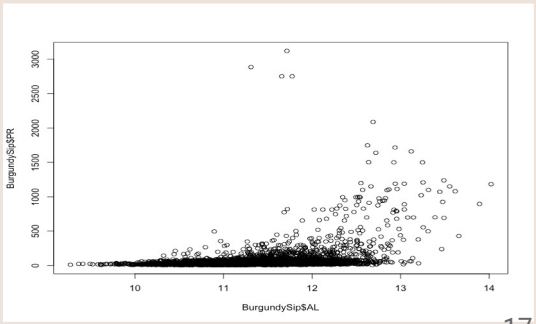
Collective Outliers



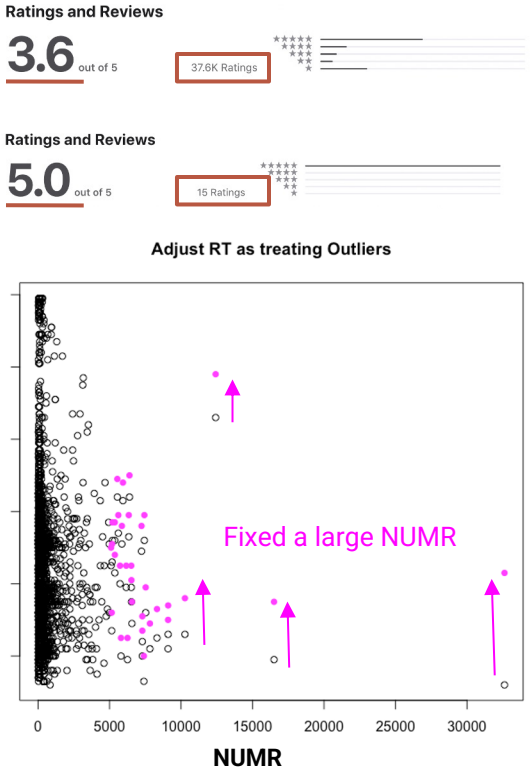


Contextual Outliers





Step 2. Treat Outliers



A large NUMR does not yield a high RT average.

However, Wines with many testers should be weighted RT by NUMR to consider for their popularity compared to Wines with fewer testers.

Step 1. Convert NUMR into Z-score

```
NUMR_Zscores <- (BurgundySip$NUMR - mean(BurgundySip$NUMR)) /
                    sd(BurgundySip$NUMR);
```

Step 2. Create a new column RT_WEI with RT weighted

```
BurgundySip$RT_WEI <- BurgundySip$RT + NUMR_Zscores / 100;
BurgundySip$RT_WEI <- ifelse(BurgundySip$RT_WEI > 5.0, 5.0,
                                round(BurgundySip$RT_WEI, 2));
```

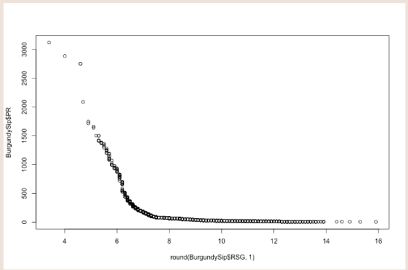
Step 3. Treat Missing Values

5 variables are still missing. We challenge these tasks by mainly 3 approaches.

```
colSums(apply(BurgundySip, 2, FUN = is.na));
```

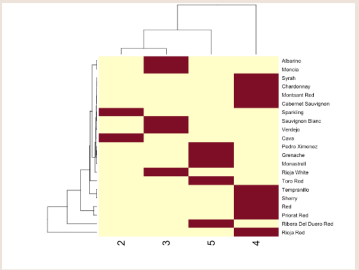
SN	RSG	AL	DN	NAME	WINE	YR	REG	TP	RT	NUMR	PR	BD	ACD
0	0	0	0	0	0	144	0	151	0	0	56	433	433

Regression Model Based



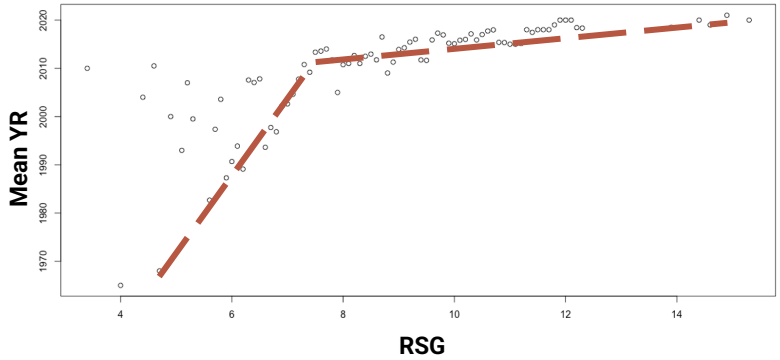
Domain Knowledge Based

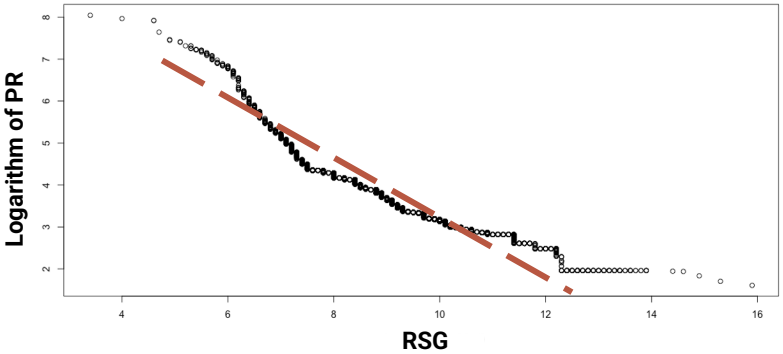
1 : M Relation Mapping



Step 3. Treat Missing Values

Regression Model Based





Both YR and PR have missing values and have a moderate positive correlation with RSG.

Nonlinear Regression suite with this task.

```
meanYear_RSG <- aggregate(as.integer(as.character(YR)) ~ round(RSG, 1),
BurgundySip, FUN = mean);
yearPredictModel <- loess(YR ~ RSG, data = meanYear_RSG);
predictYR <- predict(yearPredictModel, missingYR_RSG);
```

Linear Regression suite with this task.

```
pricePredictModel <- lm(log(train_PR) ~ train_RSG);
coefficients(pricePredictModel);
predictPR <- exp(1)^(-0.7572643 * test_RSG + 10.4689374);
```

```
(Intercept)      train_RSG
10.4689374     -0.7572643
```

Step 3. Treat Missing Values

Domain Knowledge Based

Red

Cabernet Sauvignon	Red
Grenache	Ribera Del Duero Red
Mencia	Rioja Red
Monastrell	Syrah
Montsant Red	Tempranillo
Priorat Red	Toro Red

White

Albarino
Chardonnay
Pedro Ximenez
Rioja White
Sauvignon Blanc
Verdejo

Sparkling

Cava
Sparkling

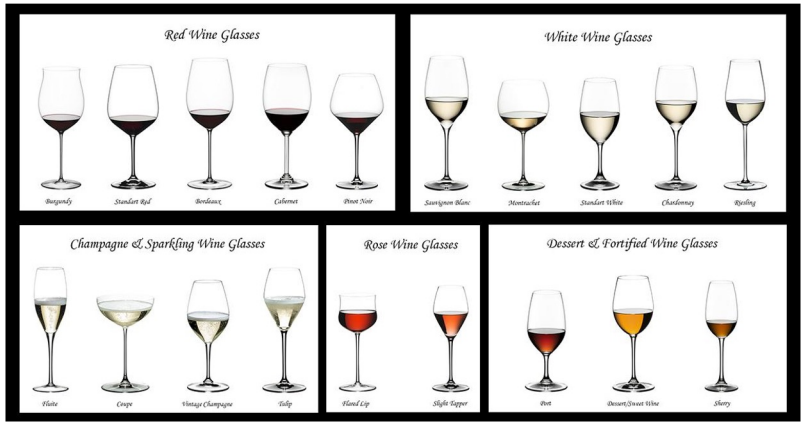
Fortified

Sherry

We found that TP of the dataset don't cover all types of wines. The top left image shows the distribution of TP. Most of them are Red wine.

On the other hand, the wines which has missing TP are White, Sparkling, Rose and Dessert. We decide to keep these wines and research which type of wine.

We assume Major category of wine ("Red", "White", "Rose", "Sparkling" and so on) behave similar features with original TP.

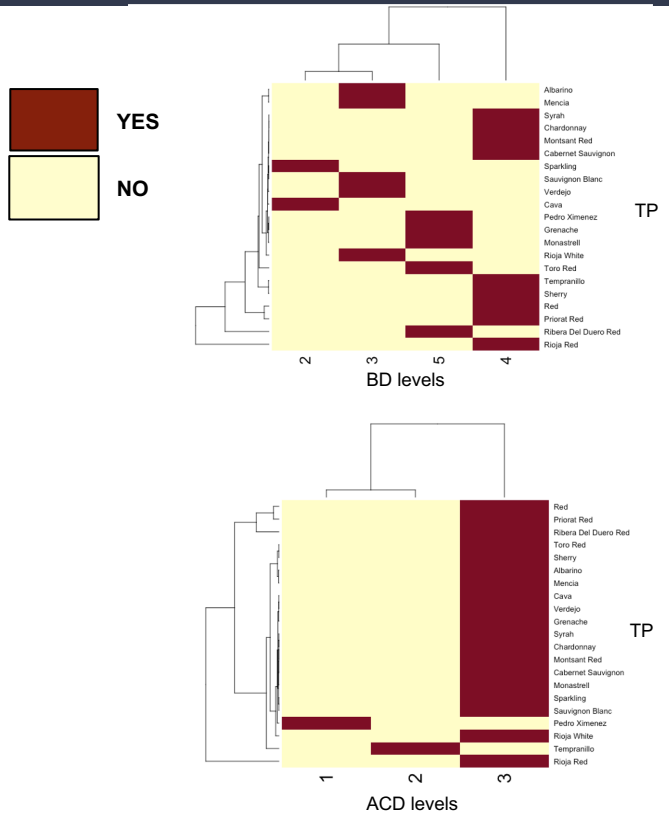



```
# There are 68 wines which has missing TP.
length(naTP_BurgundySip[!duplicated(naTP_BurgundySip$WINE),]$WINE);
levels(BurgundySip$TP) <- c(levels(BurgundySip$TP), "White", "Rose", "Dessert",
"Fortified");

BurgundySip$TP[BurgundySip$WINE == "Treixadura"] <- "White";
BurgundySip$TP[BurgundySip$WINE == "Tintilla de Rota"] <- "Red";
BurgundySip$TP[BurgundySip$WINE == "Pla dels Angels Rosado"] <- "Rose";
```

Step 3. Treat Missing Values

1 : M Relationship Mapping



Both BD and ACD have missing values and have a 1 : Many relationship with TP.

LeftJoin with 1:M mapping table

```
bdTPTable <- aggregate(BD ~ TP, BurgundySip, FUN = mean);
missingBD_BurgundySip <- BurgundySip[is.na(BurgundySip$BD),];
existingBD_BurgundySip <- BurgundySip[!is.na(BurgundySip$BD),];
removeBDIndexes <- !(names(BurgundySip) == "BD");
missingBD_BurgundySip <-
  merge(missingBD_BurgundySip[removeBDIndexes],
        bdTPTable,
        by = "TP",
        all.x = T);
BurgundySip <- rbind(existingBD_BurgundySip, missingBD_BurgundySip);
```

Modeling

Item 1. Descriptive Questions

Item 2. Predictive Analytics (Regression)

Item 3. Predictive Analytics (Cluster)

Item 1. Descriptive Analysis & Questions

During the exploratory analysis :

1. Which particular range of age (YR) in regard to wines had the highest price?
2. Determine the rating range pertaining to the wines which had the highest price.
3. What is the average price of wines based on the region which the grapes are originated from ?
4. Calculate the average price of wines based on the name of the aforementioned beverages
5. The average price of the wines based on the ratings pertinent to each brand.

Item 1. Descriptive Analysis & Questions

Conclusion / Answers Part.1

- As a matter of fact, there was no perfect correlation between the price and the year which the grapes were reaped. Contriving a pricing plan based on the YR seems rather inane. There are better options to base pricing strategy on.

- Moreover, as the regression models indicate, the attributed rating above 4.5 to the beverages lead to higher prices. PLUS, the rating which falls below the 4.5 decrease the price, as a result, it can be concluded that the rating has a strong positive impact on the pricing strategy.

- A relation can be found between region and the pricing. indeed, Conca de Barbera, Aragon, Jerez Palo Cortad, Montilla-Moriles, Mentrida as different regions recorded a higher prices compared to other regions. Though most of the wines based on region falls in the average price category, there are slight and noticeable differences between the wines.

- Price of the wines vary significantly based on the names of the wineries. Understandably, a few names such as Clos Erasmus, Emilio Hidalgo, Espectacle del Montsant and etc. have a weighty higher price compared to others.

Item 1. Descriptive Analysis & Questions

Conclusion / Answers Part.2

- As mentioned earlier, there is a strong relation between rating and price. As the rating goes higher the prices moves in the same direction (positive covariance).

```
> BurgundySip <- read_csv("F:/FP/BurgundySip.csv")
```

```
> View(BurgundySip)
```

```
> aggregate(BurgundySip$PR ~ BurgundySip$REG, FUN = mean)

 > aggregate(BurgundySip$PR ~ BurgundySip$NAME, FUN = mean)

> aggregate(BurgundySip$PR ~ BurgundySip$RT, FUN = mean)
```

Item 2. Predictive Analytics (Regression)

Business strategy

<Predictive Objective 1>

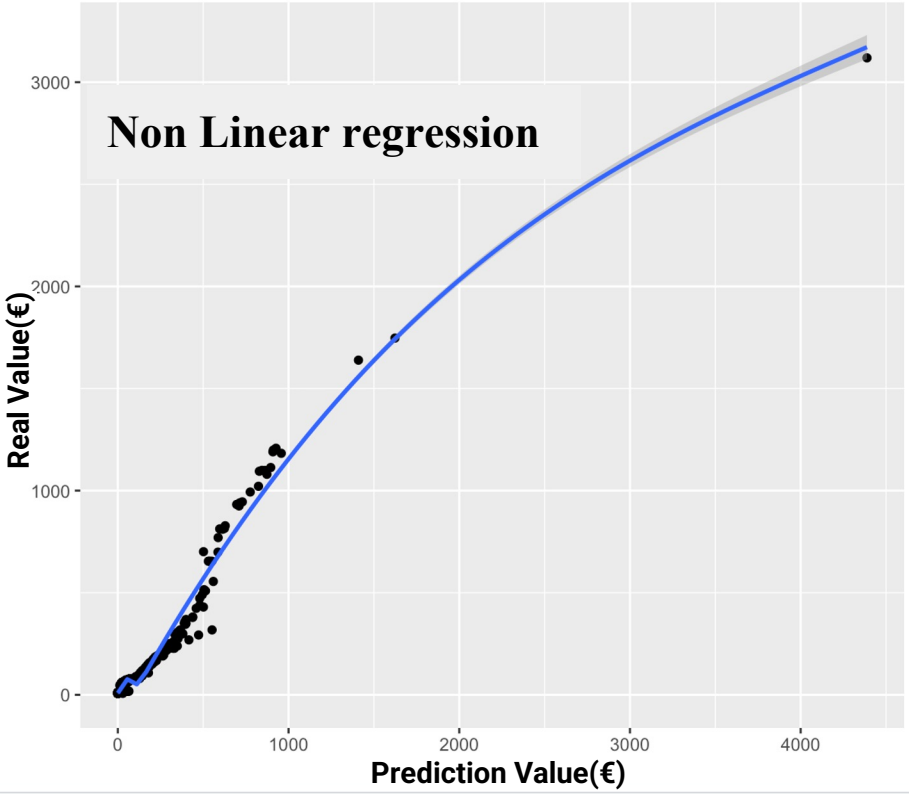
Predict prices based on wine characteristics.

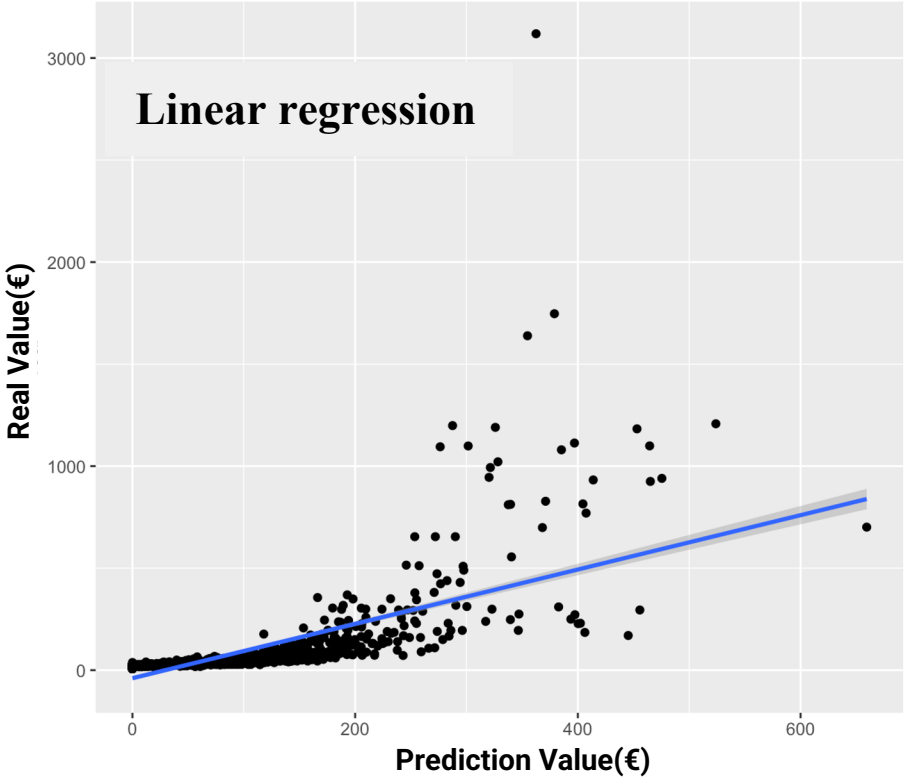
<Predictive Objective 2>

Suggest to customers highly rated wines based on predictable wine prices and wine characteristics.

Item 2. Predictive Analytics (Regression)

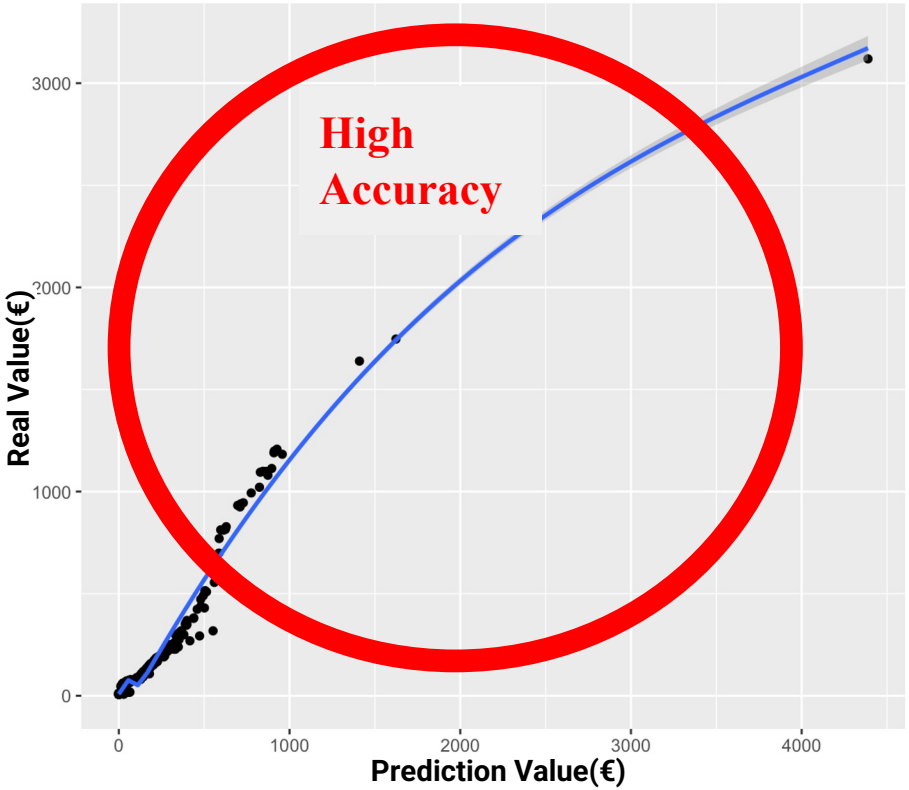
Predictive Objective 1: Predict prices based on wine characteristics.

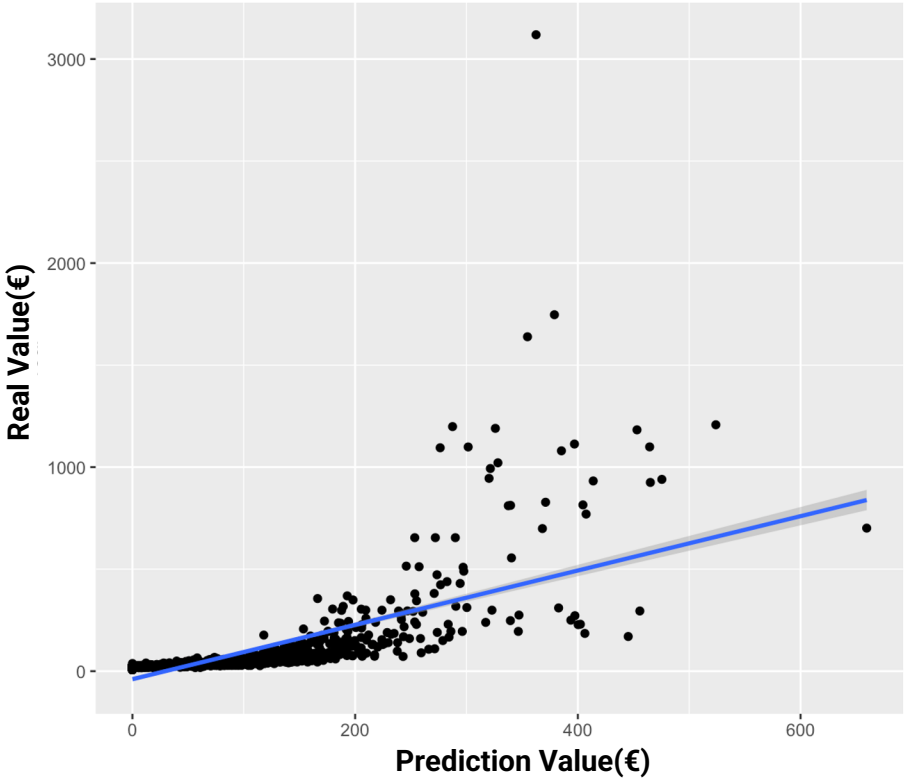




Item 2. Predictive Analytics (Regression)

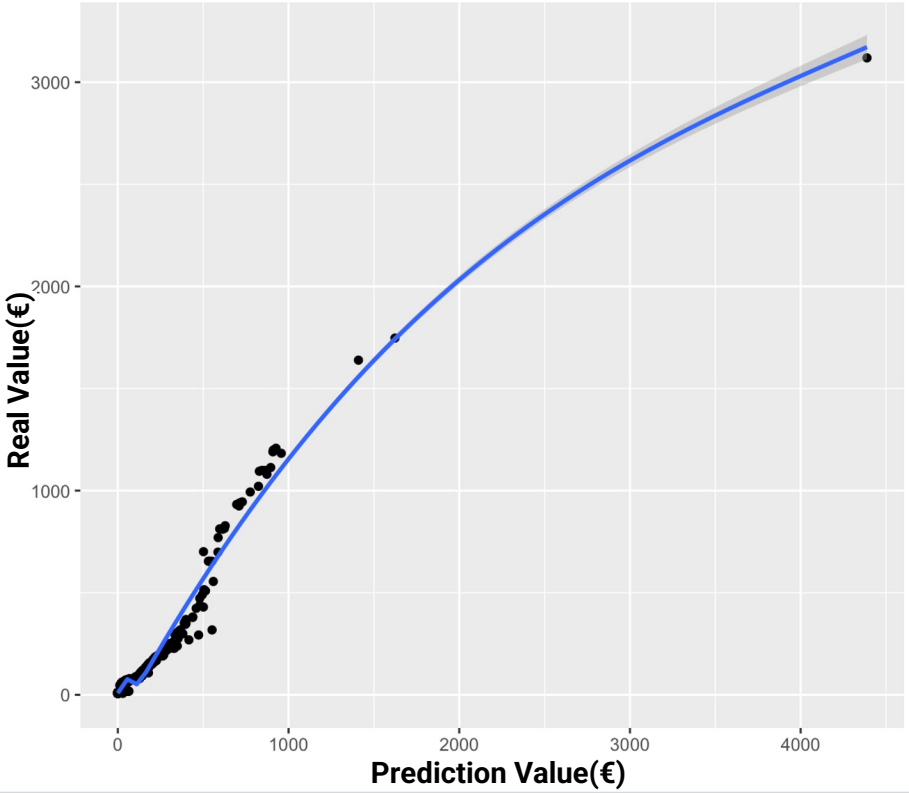
Predictive Objective 1: Predict prices based on wine characteristics.





Item 2. Predictive Analytics (Regression)

Predictive Objective 1: Predict prices based on wine characteristics.



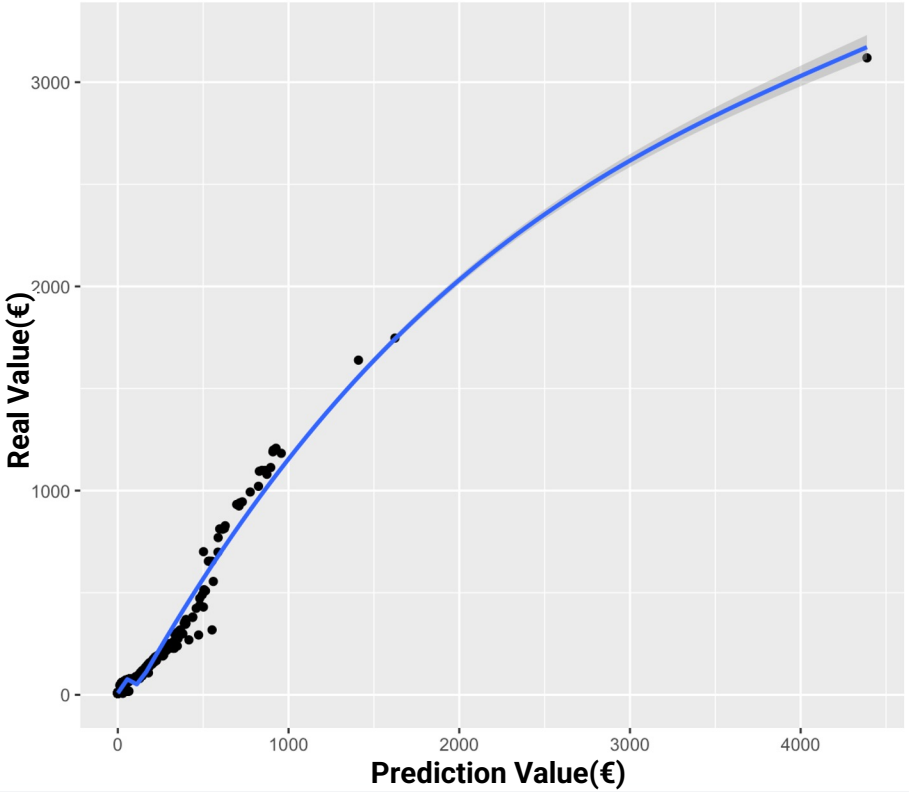
<Polynomial Linear Regression>

Dependent variable: PR
Independent variables: RSG, YR, AL, DN

```
<Train the polynomial  Linear Regression Model>
set.seed(101);
sample <- sample.split(BurgundySip, SplitRatio = 0.70);
train = subset(BurgundySip, sample == TRUE);
test = subset(BurgundySip, sample == FALSE);
model <- lm(PR ~ poly(RSG, degree = 4, raw =
          TRUE)+poly(YR, degree = 3, raw = TRUE) +
          poly(AL, degree = 2, raw = TRUE) +
          DN,
          train,control=loess.control(surface="direct"));
PR.predictions <- predict(model,test);
```

Item 2. Predictive Analytics (Regression)

Predictive Objective 1: Predict prices based on wine characteristics.



<Polynomial Linear Regression>

R-squared: 0.9101898

<Calculating R-squared>

```
mse <- mean((results$real-results$pred)^2);
mse^0.5;
SSE = sum((results$pred - results$real)^2);
SST = sum((mean(BurgundySip$PR) - results$real)^2);
R2 = 1 - SSE/SST;
```

Item 2. Predictive Analytics (Regression)

Predictive Objective 2: Predict highly rated wines based on prices & characteristics

We want to propose good wines to our customers!

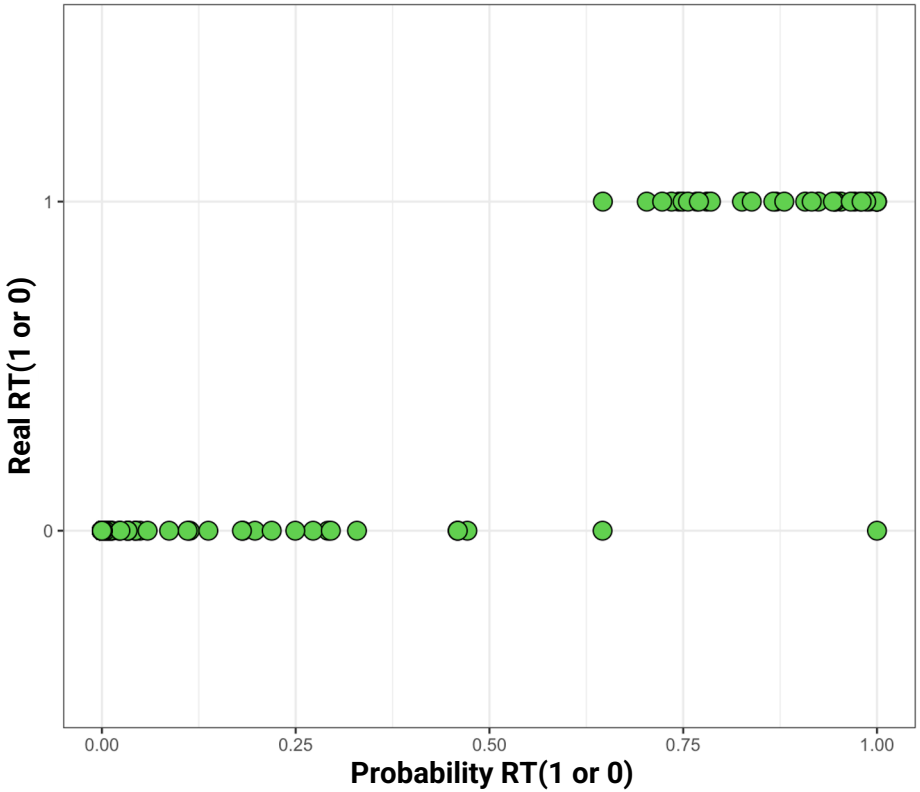
In this case, we used a logistic regression model to predict RT greater than 4.5 (good wine) and RT less than 4.5 (bad wine).

RT > 4.5

RT <= 4.5

Item 2. Predictive Analytics (Regression)

Predictive Objective 2: Predict highly rated wines based on prices & characteristics



<Logistic Regression>

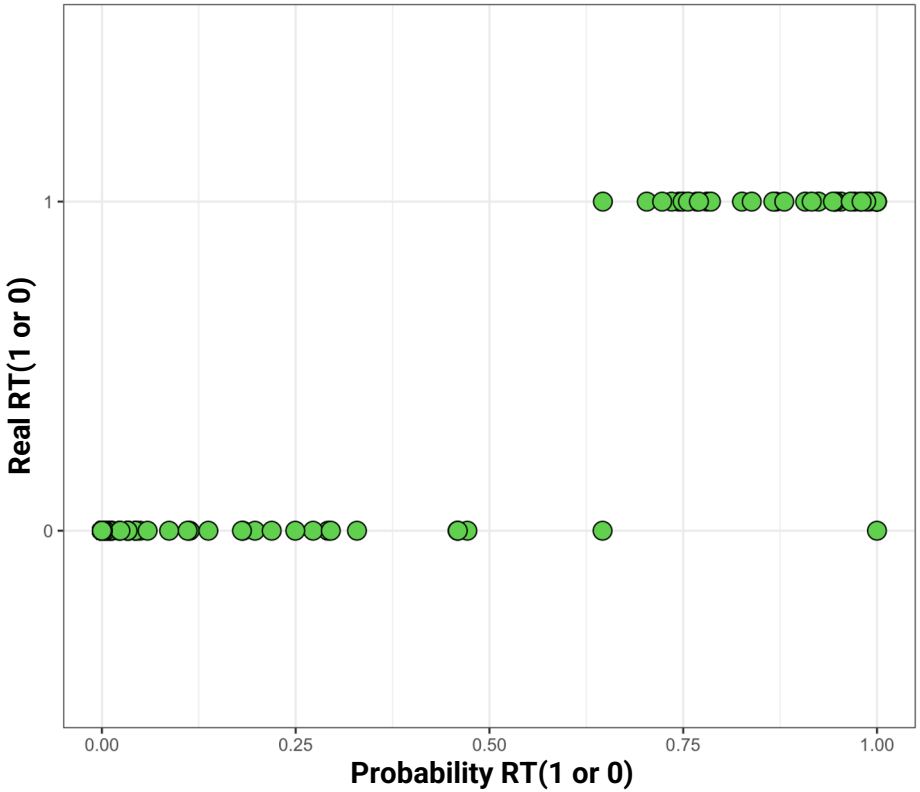
Dependent variable: RT_NEW(1: RT > 4.5, 0 : RT<= 4.5)
Independent variables: RSG, YR, AL, DN, PR

<Train the Logistic Regression Model>

```
BurgundySip$RT_NEW <-ifelse(BurgundySip$RT>4.5 , 1,
0);
set.seed(101);
sample <- sample.split(BurgundySip, SplitRatio = 0.70);
train = subset(BurgundySip, sample == TRUE);
test = subset(BurgundySip, sample == FALSE);
train$RT_NEW <- factor(train$RT_NEW);
train <- select(train, RSG,AL,DN,RT_NEW,PR,YR);
log.model <- glm(RT_NEW ~ . , family
                 =binomial(link='logit'),data = train);
```

Item 2. Predictive Analytics (Regression)

Predictive Objective 2: Predict highly rated wines based on prices & characteristics



<Logistic Regression>

Predicting using Test data…

Accuracy 0.997267759562842

<Predicting by using Test Data>

```
split = sample.split(train$RT_NEW, SplitRatio = 0.70);
final.train = subset(train, split == TRUE);
final.test = subset(train, split == FALSE);
final.log.model <- glm(formula=RT_NEW ~ . , family =
                       binomial(link='logit'),data = final.train);
fitted.probabilities <-predict(final.log.model,newdata =
                               final.test,type='response');
fitted.results <- ifelse(fitted.probabilities > 0.5,1,0);
misClasificError <- mean(fitted.results !=
final.test$RT_NEW);
print(paste('Accuracy',1-misClasificError));
```

Item 2. Predictive Analytics (Regression)

Conclusion

<Predictive Objective 1>

Independent variables: RSG, YR, AL, DN

To predict wine prices by Nonlinear regression model : R-squared values 0.91

<Predictive Objective 2>

Independent variables: PR, RSG, YR, AL, DN

To predict wine ratings by Logistic regression model : Accuracy values 0.99

- Residual sugar level (RSG)
- Year of production (YR)
- Alcohol level (AL)
- Density or gravity of wine (DN)

To make a good wine, these factors must be considered.

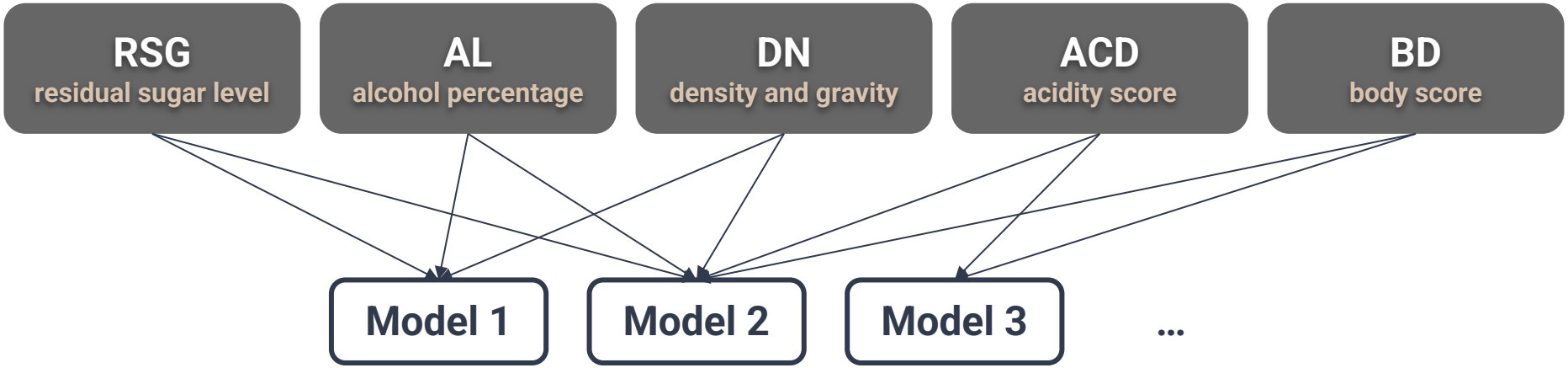
Item 3. Predictive Analytics (Cluster)

1. Analyzing potential groupings based on taste of Wine
2. Analyzing potential groupings based on brand of Wine

Discover potential high-demand wines

Item 3. Predictive Analysis (Cluster)

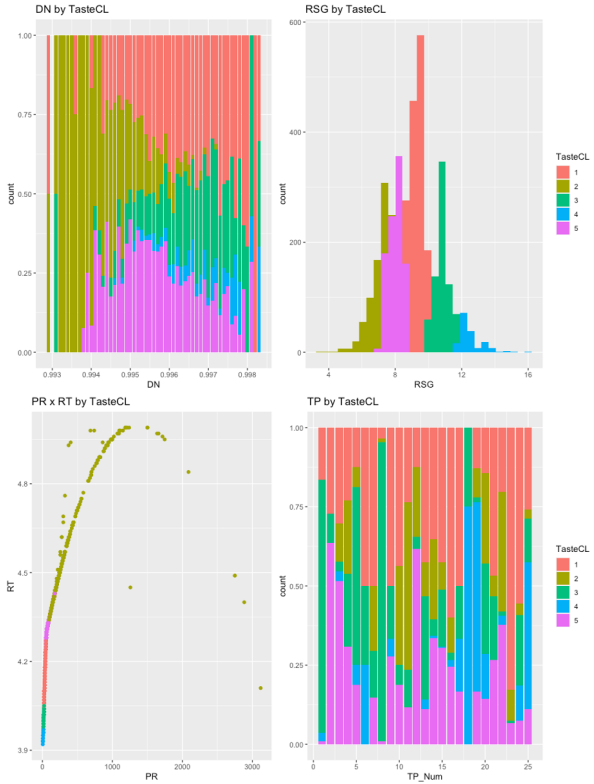
Cluster based on Taste




```
BurgundySip_TasteNum <- data.frame(sapply(BurgundySip[, !sapply(BurgundySip, is.character)], as.numeric));
fit <- kmeans(BurgundySip_TasteNum[, c("RSG", "AL", "DN", "ACD", "BD")], 5);
BurgundySip_TasteNum$cluster <- as.factor(fit$cluster);
```

Item 3. Predictive Analysis (Cluster)

Cluster based on Taste



- 1: Middle
- 2: Lowest Sugar Level and Density
- 3: 2nd Highest Sugar Level
- 4: Highest Sugar Level and Density
- 5: 2nd Lowest Sugar Level

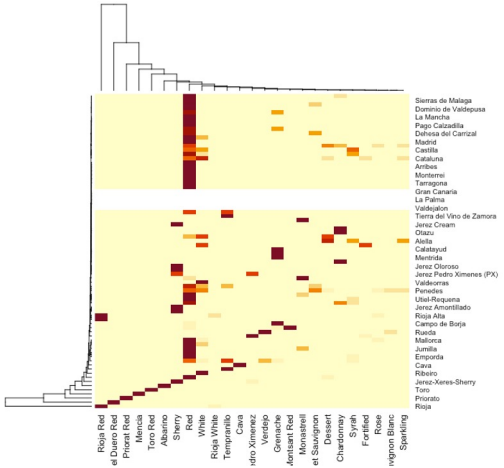
The cluster is highly dependent on Sugar Level. However, since both PR and RT are dependent on Sugar Level, they could be classified cleanly.

We succeeded in classifying wines by acidity and sweetness information, independent of the type of wine, such as RED, White, Dessert, and so on.

Some regions produce only the sweetest wine, which is a very interesting result.

Item 3. Predictive Analysis (Cluster)

Cluster based on Brand



REG distribution vectors by TP

TP_Grenache = [0 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 …]

TP_White = [0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 …]

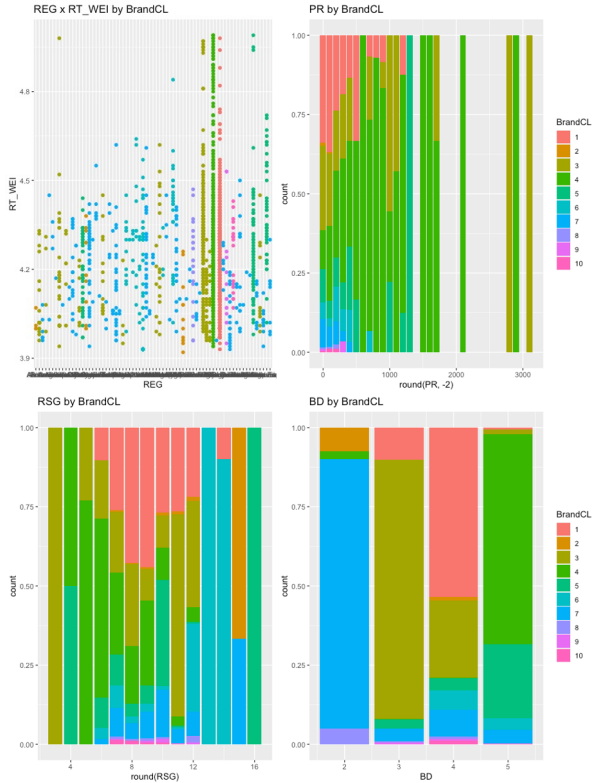
TP_Dessert = [0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 …]

TP_Syrah = [0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 …]

```
REG_TP_Matrix <- table(BurgundySip$REG, BurgundySip$TP);
REG_TP_DataFrame <- # convert REG_TP_Matrix into data frame
BurgundySip_Brand <- merge(BurgundySip, REG_TP_DataFrame, by = "REG", all.x = T);
BurgundySip_BrandNum <- data.frame(sapply(BurgundySip_Brand[, !sapply(BurgundySip_Brand, is.character)], as.numeric));
fit <- kmeans(BurgundySip_BrandNum[, columnNams_TP], 10);
BurgundySip_BrandNum$cluster <- as.factor(fit$cluster);
```

Item 3. Predictive Analysis (Cluster)

Cluster based on Brand



- 1: BD = 4 & Rioja
- 2: 2nd Highest Sugar
- 3: BD = 3 & Lowest Sugar
- 4: Lower Sugar & Ribera del
- 5: Highest Sugar
- 6: Higher Sugar
- 7: BD = 2
- 8: Penedes
- 9: Rioja Alta
- 10: Sardon de Duero

We succeeded in summarizing the Regions with similarities by using the distribution vector of Regions for each Type.

The outcome revealed significant differences in Sugar Level and Body Score, despite the clustering was conducted without using component data of wine.

It indicates that each brand has a distinctly different taste.

Item 3. Predictive Analysis (Cluster)

Conclusion

Suggestion 1

Taste Cluster: 2

Brand Cluster: 3 or 4

High Quality

High Price

Suggestion 2

Taste Cluster: 5

Brand Cluster: 6 or 8

High Quality

Low Price

1. High-quality, high-priced wines can generate stable profits.
2. High-quality, low-priced wines can be loved by many customers who want to enjoy wine at ease.

Conclusion

Conclusion

1. Lower ACD leads to a better wine and therefore higher price and popularity.
   ➡ **It does not have a significant effect(Non linear regression model)**
2. Name of the wine irrespective of its quality is an important factor (more famous higher the price)
   ➡ **Yes, but not directly (Name of wine → Sugar Level → Price) (Cluster-based)**
3. Wines that are more aged up are costlier compared to ones that are harvested more recently
   ➡ **Yes, an older wines tend to be more expensive.(Non linear regression model)**
4. Lowers rsg often means higher price
   ➡ **Yes (Cluster-based)**
5. Number of testers do not have a significant effect on pricing
   ➡ **It does not have a significant effect(Non linear regression model)**
6. BD does not have a noticeable effect on pricing
   ➡ **It does not have a noticeable effect(Non linear regression model)**

E.o.F