Coffee species and quality prediction with machine learning

Daoud Youssef

2020-06-23

Contents

Introduction

This project is part of the Harvard: PH125.9 course Data science: Capstone course. The aim of this project is to predict the quality of arabica coffee on a scale of 0-10 or low-high-medium given a set of features as inputs.

Coffee is consumed on daily basis and it's one of the most popular non-alcoholic drinks. It's the second traded commodity in the world market (Andrew Menke 2018) and the sector is expected to continue growing and the revenue in the coffee segment is worth $US\$362,601m$ in 2020.(Satista n.d.)

From growing to brewing, a lot of factors affect the flavor, intensity and the quality of coffee (Co n.d.). Mainly, we have 3 species: Arabica, Canephora (known as Robusta) and Liberica (Co n.d.). The most popular are Arbica and Robusta. Arabica grows on high altitude while Robusta grows on lower altitude. Each of these two species have some different varieties. Most of the commercial roasters use Arabica beans (Andrea Trevisan n.d.).

This project is organized as follows: In section 2, we present the dataset used and detailed explanation on all of the models. In section 3, we explore the `Coffee.data` dataset and its statistical properties. In addition we present the algorithms of the models used in this project. In Section 4, we discuss the results and we conclude in section 5.

The evaluation of the validity of our model is based on the Residual Mean Squared Error RMSE for the linear regression models and the confusion matrix for the categorical algorithms.

Methodology

Dataset

First, we load libraries and data. Data is retrieved from Kaggle website who redirect to https://github.com/jldbc/coffee-quality-database. The source of this data is the Coffee Quality Institute[1]. There is two datasets created for arabica and robusta coffee. The quality measures used as predictors in this project for coffee are:

- `Altitude`,
- `Aroma`,
- `Flavor`,
- `Aftertaste`,
- `Acidity`,
- `Body`,
- `Balance`,
- `Uniformity`,
- `Clean Cup`,
- `Sweetness`,
- `Cupper Points`. For predicting the type of coffee, we used species as outcome. For predicting quality of coffee, we created two column:
- `score` : we divided the total cup points by 10 and we rounded to nearest integer so we have a score range between 0 and 10.
- `quality`: This column is added by classifying `score` into 3 categories:
  - **low** with `score` below 8
  - **medium** with `score` equals to 8
  - **high** with `score` greater to 8.

```r
## https://github.com/jldbc/coffee-quality-database Libraries
library(tidyverse)
library(measurements)
library(corrgram)
library(corrplot)
library(reshape2)
library(caret)
library(caTools)
library(e1071)
library(randomForest)
```

[1] https://www.coffeeinstitute.org

```r
library(rpart)
library(rattle)
library(rpart.plot)
library(RColorBrewer)
library(kableExtra)
```

```r
## loading data
arabica.data <- read.csv("arabica_data_cleaned.csv")
robusta.data <- read.csv("robusta_data_cleaned.csv")

### selecting columns
arabica.data <- arabica.data %>% select(Species, Country.of.Origin,
    unit_of_measurement, altitude_high_meters, altitude_mean_meters,
    Aroma, Flavor, Aftertaste, Acidity, Body, Balance, Uniformity,
    Clean.Cup, Sweetness, Cupper.Points, Total.Cup.Points)
robusta.data <- robusta.data %>% select(Species, Country.of.Origin,
    unit_of_measurement, altitude_mean_meters, altitude_high_meters,
    Fragrance...Aroma, Flavor, Aftertaste, Salt...Acid, Mouthfeel,
    Balance, Uniform.Cup, Clean.Cup, Bitter...Sweet, Cupper.Points,
    Total.Cup.Points)
### rename columns to match
robusta.data <- robusta.data %>% rename(Aroma = Fragrance...Aroma,
    Acidity = Salt...Acid, Body = Mouthfeel, Uniformity = Uniform.Cup,
    Sweetness = Bitter...Sweet)
coffee.data <- rbind(arabica.data, robusta.data)
```

We remove data that contains `NA` values and we make sure that our data has no missing information. One missing country has been renamed missing since only its name is missed. We removed all observations that have an altitude over 5000m since its clear that there is an error in data entry. We kept all observations having an average over 4 on `aroma` and `cleaning cup`. The reason is to reduce outliers.

```r
## remove rows that contain NAs values
coffee.data <- coffee.data %>% na.omit()
### Data is tidy now
anyNA(coffee.data)
```

```
## [1] FALSE
```

```r
coffee.data$Species <- as.factor(coffee.data$Species)
coffee.data$Country.of.Origin <- as.factor(coffee.data$Country.of.Origin)
coffee.data$unit_of_measurement <- as.factor(coffee.data$unit_of_measurement)

### One missing country name, il replace it by missing country
levels(coffee.data$Country.of.Origin)[levels(coffee.data$Country.of.Origin) ==
    ""] <- "Missing country"

## convert feet to meters
coffee.data <- within(coffee.data, {
    i <- unit_of_measurement == "ft"
    unit_of_measurement[i] <- "m"
    altitude_high_meters[i] <- conv_unit(altitude_high_meters[i],
        "ft", "m")
    altitude_mean_meters[i] <- conv_unit(altitude_mean_meters[i],
        "ft", "m")
})
```

```r
### we choose farms with altitude above 100 m and below 5000 i
### removed above 5 000 m cause they are outliers and seems
### error in data entry.
coffee.data <- coffee.data %>% filter(Aroma > 4 & Clean.Cup >
    4 & altitude_mean_meters > 100 & altitude_high_meters < 5000)
coffee.data$score <- round(coffee.data$Total.Cup.Points/10)
coffee.data$quality <- ifelse(coffee.data$score < 8, "low", "high")
coffee.data$quality[coffee.data$score == 8] <- "medium"
coffee.data$quality <- as.factor(coffee.data$quality)

### remove unnecessary columns
coffee.data <- subset(coffee.data, select = -c(unit_of_measurement,
    altitude_high_meters, i))
```

Now,Data is clean, tidy and ready for exploration and analysis.

Model Evaluation

Confusion Matrix

To evaluate the performance of our classifications models, we are going use the confusion matrix.

```r
library(knitr)
library(kableExtra)
table <- matrix(c("True positives (TP)", "False positives (FP)",
    " False negatives (FN)", " True negatives (TN)"), ncol = 2,
    byrow = T)
colnames(table) <- c("Actually Positive", "Actually Negative")
rownames(table) <- c("Predicted positive", "Predicted negative")
# as.table(table)
kable(table)
```

	Actually Positive	Actually Negative
Predicted positive	True positives (TP)	False positives (FP)
Predicted negative	False negatives (FN)	True negatives (TN)

From this matrix, we compute some rates:

- Accuracy is the ability of the model to predict correctly and it is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Sensitivity is the probability to predict a positive outcome when the actual outcome is positive and it is calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}$$

- Specificity : is the probability to not predict a positive outcome when the actual outcome is not a positive and its calculated as follows:

$$Specificity = \frac{TN}{TN + FP}$$

Root Mean Square Error

Validation of our model is based on the value of Root Mean Square Error **RMSE**. The **RMSE** loss function is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2} \tag{1}$$

with

- y the actual outcome,
- $\hat{y}$ the predicted outcome.
- N number of observations.

The code that compute the RMSE is :

```
RMSE <- function(true_ratings, predicted_ratings) {
    sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

Data analysis and model preparation

Below the top 6 rows of the dataset.

```
head(coffee.data)
```

```
##   Species Country.of.Origin altitude_mean_meters Aroma Flavor Aftertaste
## 1 Arabica          Ethiopia                 2075  8.67   8.83       8.67
## 2 Arabica          Ethiopia                 2075  8.75   8.67       8.50
## 3 Arabica         Guatemala                 1700  8.42   8.50       8.42
## 4 Arabica          Ethiopia                 2000  8.17   8.58       8.42
## 5 Arabica          Ethiopia                 2075  8.25   8.50       8.25
## 6 Arabica          Ethiopia                 1635  8.25   8.33       8.50
##   Acidity Body Balance Uniformity Clean.Cup Sweetness Cupper.Points
## 1    8.75 8.50    8.42         10        10     10.00          8.75
## 2    8.58 8.42    8.42         10        10     10.00          8.58
## 3    8.42 8.33    8.42         10        10     10.00          9.25
## 4    8.42 8.50    8.25         10        10     10.00          8.67
## 5    8.50 8.42    8.33         10        10     10.00          8.58
## 6    8.42 8.33    8.50         10        10      9.33          9.00
##   Total.Cup.Points score quality
## 1            90.58     9    high
## 2            89.92     9    high
## 3            89.75     9    high
## 4            89.00     9    high
## 5            88.83     9    high
## 6            88.67     9    high
```

Data exploration and visualization

Before we start, let's have a look on the structure on our data and observe a statistical summary.

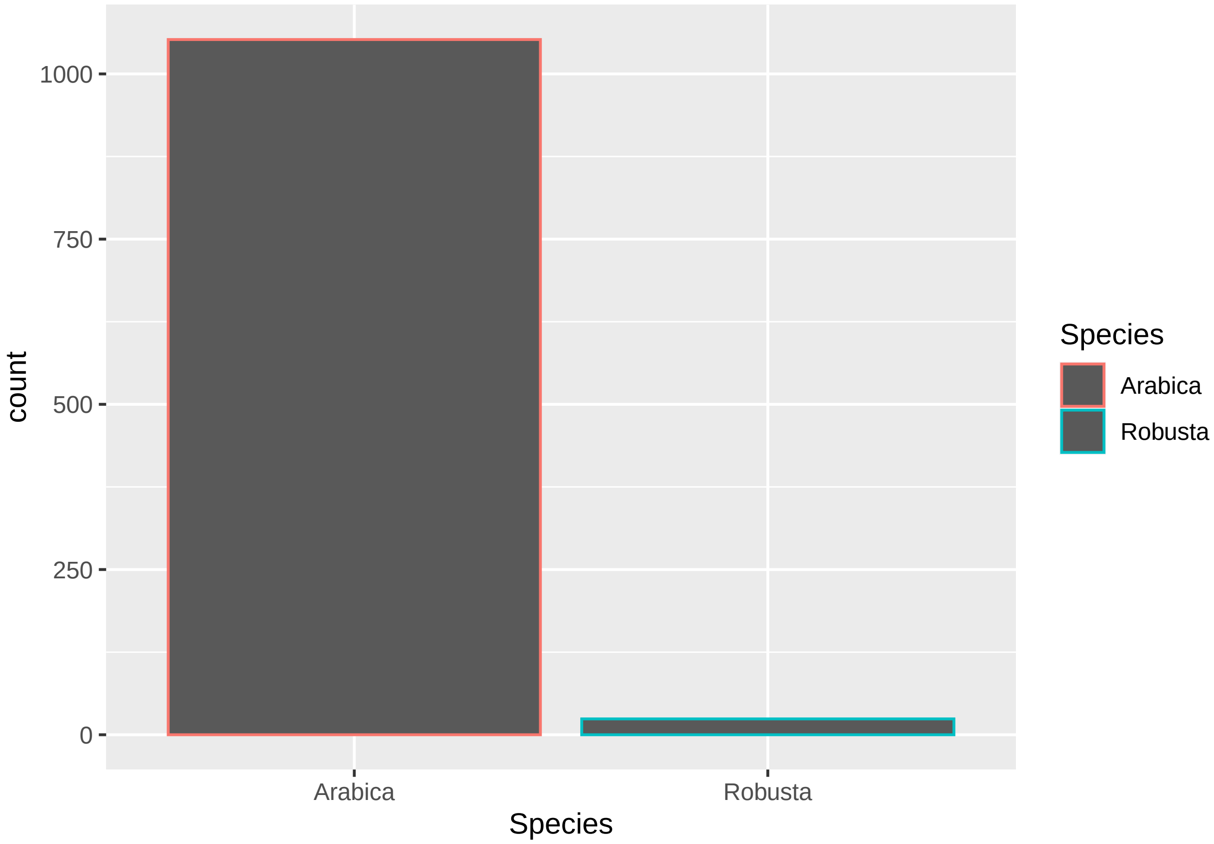
```
### histogram of distribution of species
str(coffee.data)
```

```
## 'data.frame':    1076 obs. of  16 variables:
##  $ Species             : Factor w/ 2 levels "Arabica","Robusta": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Country.of.Origin   : Factor w/ 36 levels "Brazil","Burundi",..: 9 9 10 9 9 9 9 9 9 32 ...
##  $ altitude_mean_meters: num  2075 2075 1700 2000 2075 ...
##  $ Aroma               : num  8.67 8.75 8.42 8.17 8.25 8.25 8.67 8.08 8.17 8.25 ...
##  $ Flavor              : num  8.83 8.67 8.5 8.58 8.5 8.33 8.67 8.58 8.67 8.42 ...
##  $ Aftertaste          : num  8.67 8.5 8.42 8.42 8.25 8.5 8.58 8.5 8.25 8.17 ...
##  $ Acidity             : num  8.75 8.58 8.42 8.42 8.5 8.42 8.42 8.5 8.5 8.33 ...
##  $ Body                : num  8.5 8.42 8.33 8.5 8.42 8.33 8.33 7.67 7.75 8.08 ...
##  $ Balance             : num  8.42 8.42 8.42 8.25 8.33 8.5 8.42 8.42 8.17 8.17 ...
##  $ Uniformity          : num  10 10 10 10 10 10 9.33 10 10 10 ...
##  $ Clean.Cup           : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ Sweetness           : num  10 10 10 10 10 9.33 9.33 10 10 10 ...
##  $ Cupper.Points       : num  8.75 8.58 9.25 8.67 8.58 9 8.67 8.5 8.58 8.5 ...
##  $ Total.Cup.Points    : num  90.6 89.9 89.8 89 88.8 ...
##  $ score               : num  9 9 9 9 9 9 9 9 9 9 ...
##  $ quality             : Factor w/ 3 levels "high","low","medium": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(coffee.data)
```

```
##     Species     Country.of.Origin altitude_mean_meters     Aroma
##  Arabica:1052   Mexico   :227     Min.   : 110         Min.   :5.080
##  Robusta:  24   Guatemala:150     1st Qu.:1000         1st Qu.:7.420
##                 Colombia :149     Median :1300         Median :7.580
##                 Brazil   : 91     Mean   :1263         Mean   :7.582
##                 Taiwan   : 69     3rd Qu.:1600         3rd Qu.:7.750
##                 Honduras : 50     Max.   :4287         Max.   :8.750
##                 (Other)  :340
##      Flavor         Aftertaste       Acidity           Body
##  Min.   :6.170   Min.   :6.170   Min.   :5.250   Min.   :6.330
##  1st Qu.:7.330   1st Qu.:7.250   1st Qu.:7.330   1st Qu.:7.330
##  Median :7.580   Median :7.420   Median :7.500   Median :7.500
##  Mean   :7.531   Mean   :7.404   Mean   :7.538   Mean   :7.517
##  3rd Qu.:7.750   3rd Qu.:7.580   3rd Qu.:7.750   3rd Qu.:7.670
##  Max.   :8.830   Max.   :8.670   Max.   :8.750   Max.   :8.580
##
##     Balance        Uniformity       Clean.Cup        Sweetness
##  Min.   :6.080   Min.   : 6.000   Min.   : 5.330   Min.   : 6.000
##  1st Qu.:7.330   1st Qu.:10.000   1st Qu.:10.000   1st Qu.:10.000
##  Median :7.500   Median :10.000   Median :10.000   Median :10.000
##  Mean   :7.516   Mean   : 9.876   Mean   : 9.888   Mean   : 9.889
##  3rd Qu.:7.750   3rd Qu.:10.000   3rd Qu.:10.000   3rd Qu.:10.000
##  Max.   :8.750   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##  Cupper.Points    Total.Cup.Points     score           quality
##  Min.   : 5.170   Min.   :63.08    Min.   :6.000   high  : 70
##  1st Qu.: 7.250   1st Qu.:81.25    1st Qu.:8.000   low   : 17
##  Median : 7.500   Median :82.50    Median :8.000   medium:989
##  Mean   : 7.496   Mean   :82.24    Mean   :8.048
##  3rd Qu.: 7.750   3rd Qu.:83.60    3rd Qu.:8.000
##  Max.   :10.000   Max.   :90.58    Max.   :9.000
##
```

```
coffee.data %>% group_by(Species) %>% ggplot(aes(Species)) +
    geom_bar(aes(col = Species))
```

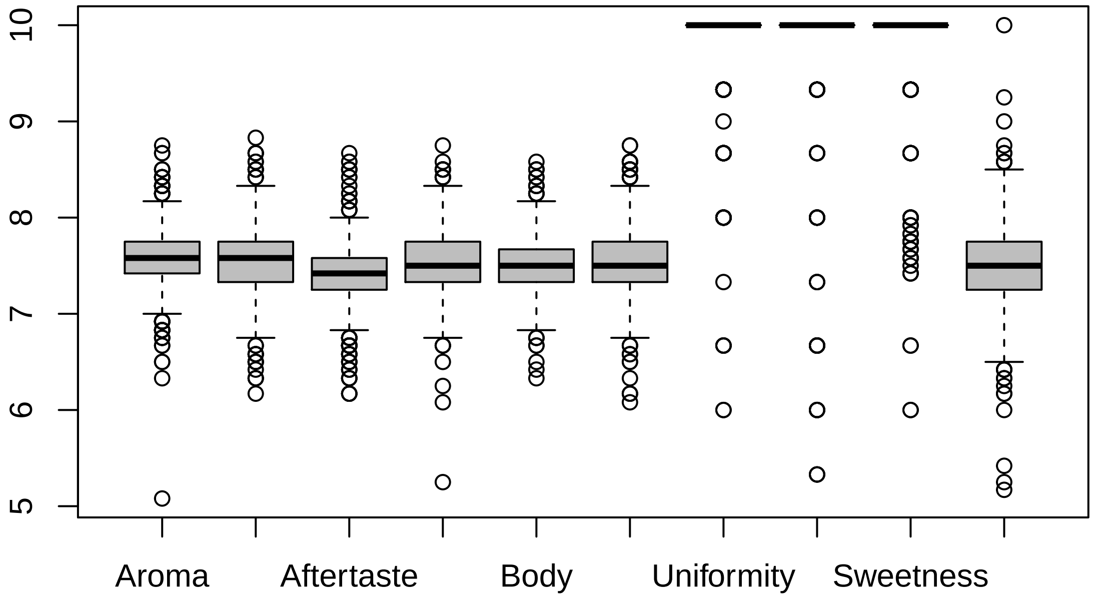


```
beans <- coffee.data %>% group_by(Species) %>% summarize(n = n())
```

Overall, We have 1076 observations dominated by arabica species with 1052 observations and 24 observations for robusta species.

Almost all variables have outliers as shown by the plot below.

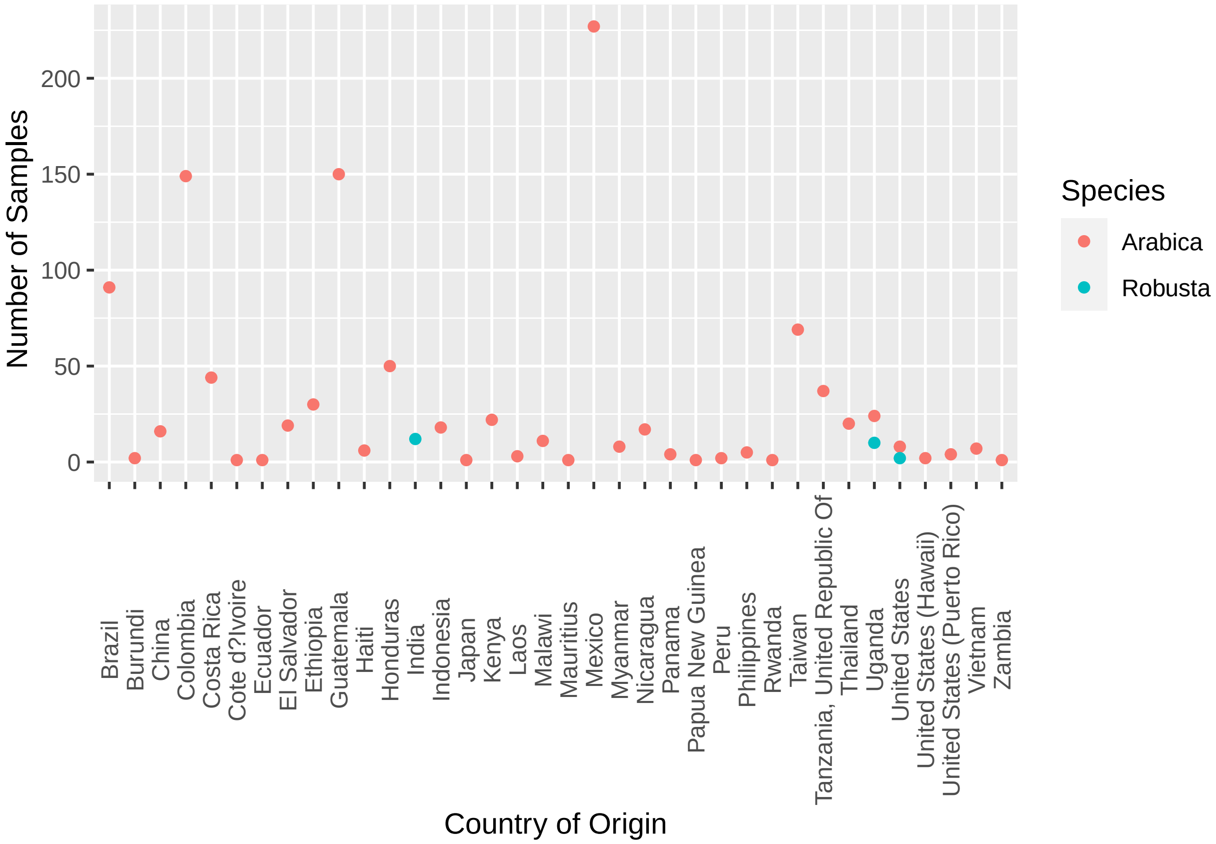
```
### Identifying outliers for quality measures
coffee.data %>% select(Aroma, Flavor, Aftertaste, Acidity, Body,
    Balance, Uniformity, Clean.Cup, Sweetness, Cupper.Points) %>%
    boxplot(col = "grey") + theme(axis.text.x = element_text(angle = 45))
```



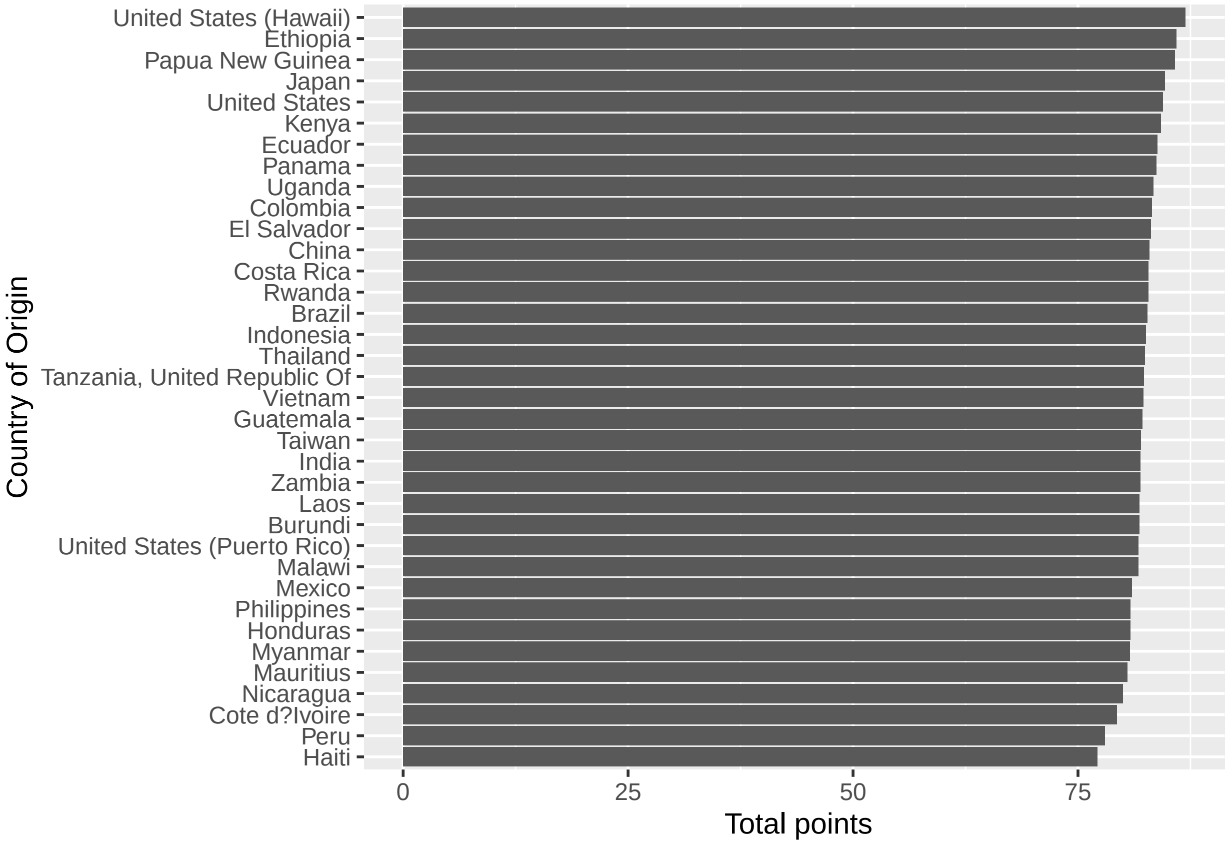
```
## NULL
```

Most of coffee samples in these dataset come from Mexico.

```
#### countries and species
coffee.data %>% group_by(Species, Country.of.Origin) %>% summarize(n = n()) %>%
    ggplot(aes(Country.of.Origin, n, colour = Species)) + geom_point() +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
    labs(Title = "Distribution of Species from origin country ",
        x = "Country of Origin", y = " Number of Samples")
```



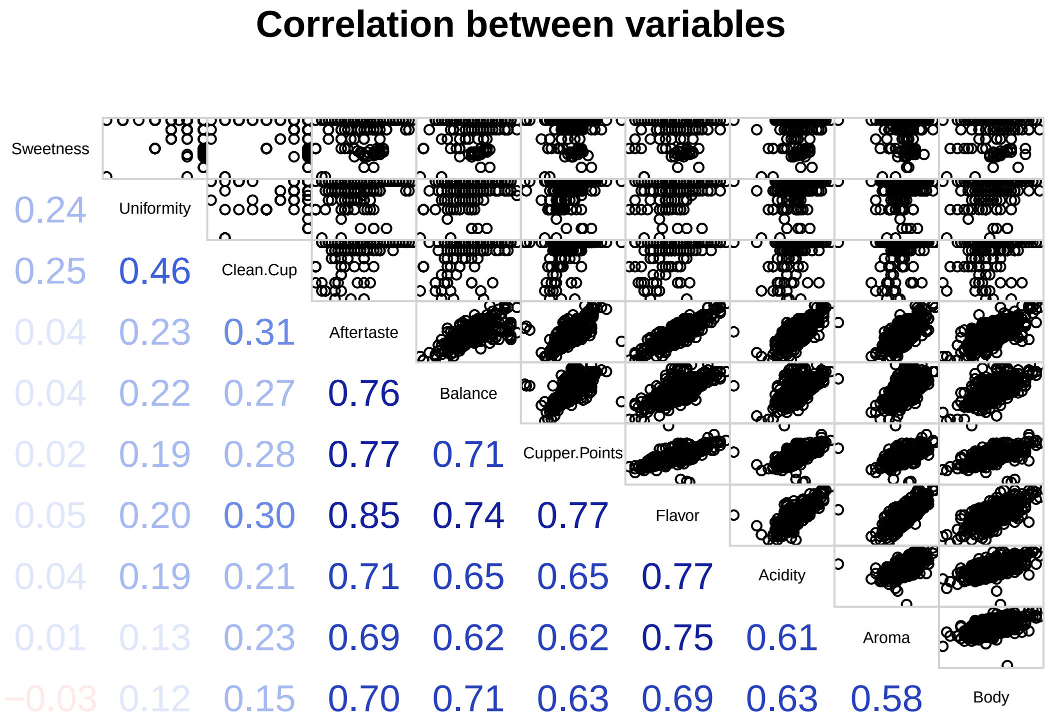
```
### best coffee most rated
p <- coffee.data %>% group_by(Country.of.Origin) %>% summarize(avg_quality = mean(Total.Cup.Points)) %>%
    arrange(desc(avg_quality))

p %>% ggplot(aes(x = reorder(Country.of.Origin, avg_quality),
    y = avg_quality)) + geom_bar(stat = "identity") + coord_flip() +
    labs(Title = "Rating of coffee quality  ", x = "Country of Origin",
        y = " Total points")
```



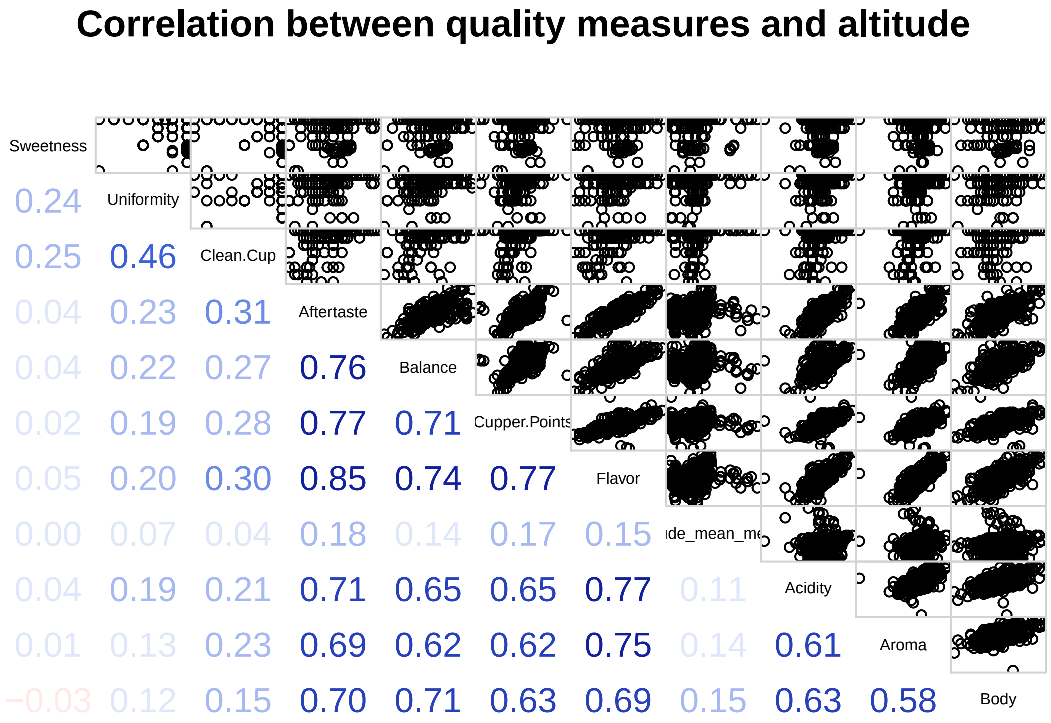
As shown above, coffee originated from United States (Hawaii) has the highest score among others.

Assessing correlation analysis is important before any type of modeling to check the importance of each predictor variable in our analysis. `Sweetness` is less correlated to other variables.

```r
#### Correlation between variables
coffee.data %>% select(Aroma, Flavor, Aftertaste, Acidity, Body,
    Balance, Uniformity, Clean.Cup, Sweetness, Cupper.Points) %>%
    corrgram(order = TRUE, lower.panel = panel.cor, upper.panel = panel.pts,
        main = " Correlation between variables ")
```

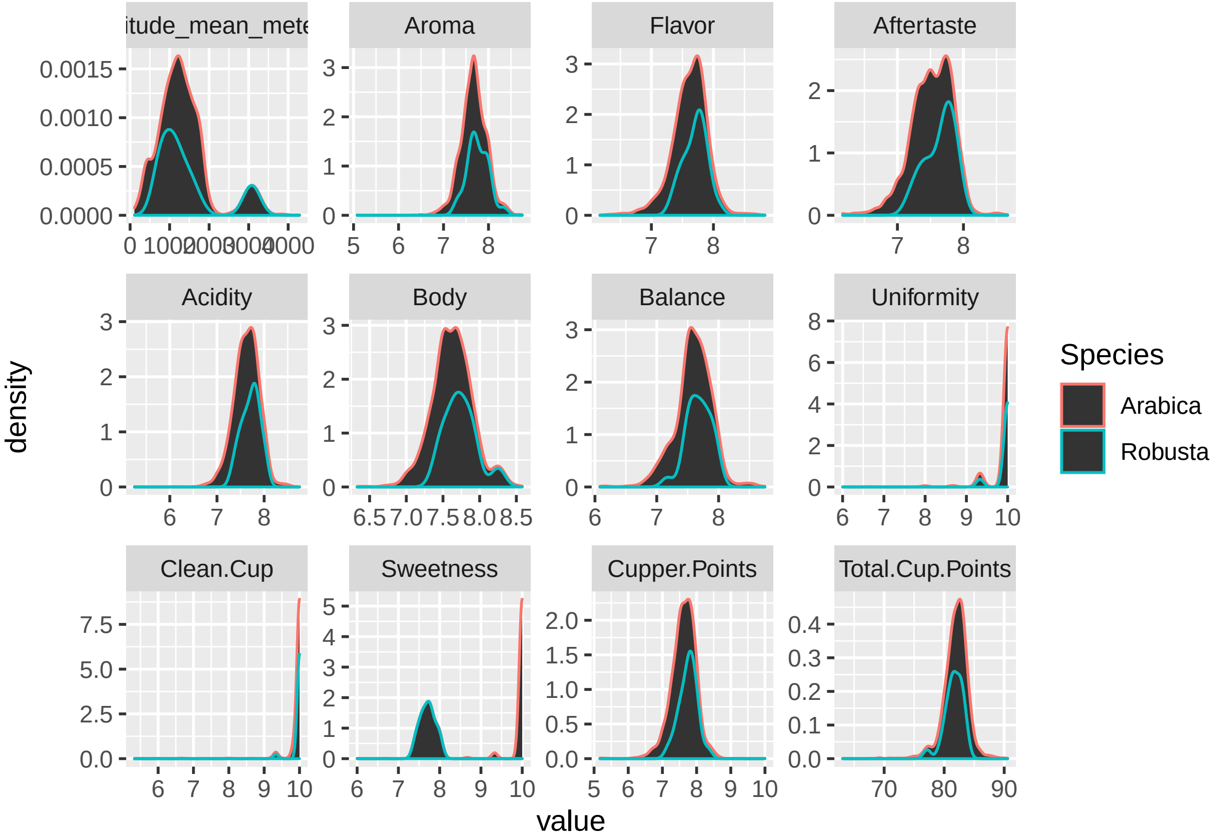


```
### Correlation between variables adding mean altitude
coffee.data %>% select(Aroma, Flavor, Aftertaste, Acidity, Body,
    Balance, Uniformity, Clean.Cup, Sweetness, Cupper.Points,
    altitude_mean_meters) %>% corrgram(order = TRUE, lower.panel = panel.cor,
    upper.panel = panel.pts, main = " Correlation between quality measures and altitude ")
```



Adding the altitude variable does not have an impact on the correlation matrix. Thus, its not correlated to any qaulity measure variable.

```
##### density comparison
ttt <- melt(coffee.data, id.vars = "Species", measure.vars = c("altitude_mean_meters",
    "Aroma", "Flavor", "Aftertaste", "Acidity", "Body", "Balance",
    "Uniformity", "Clean.Cup", "Sweetness", "Cupper.Points",
    "Total.Cup.Points"))
ggplot(ttt, aes(x = value, colour = Species)) + stat_density() +
    facet_wrap(variable ~ ., scales = "free")
```



We clearly observe from these density distributions that variables : `Clean.Cup`, `Sweetness` and `Uniformity` are skewed. The skew in these distributions is explained by the extreme values and score in the data.

Importance of variables

Arabica coffee tend to have a sweeter taste (Co n.d.) and that what makes `Sweetness` the most important factor in predicting the species of beans followed by `Body`.

```
#### importance of variable in predicting species

df <- subset(coffee.data, select = -c(Country.of.Origin, quality))
var_imp <- filterVarImp(df[, -1], df[, 1])
var_imp
```

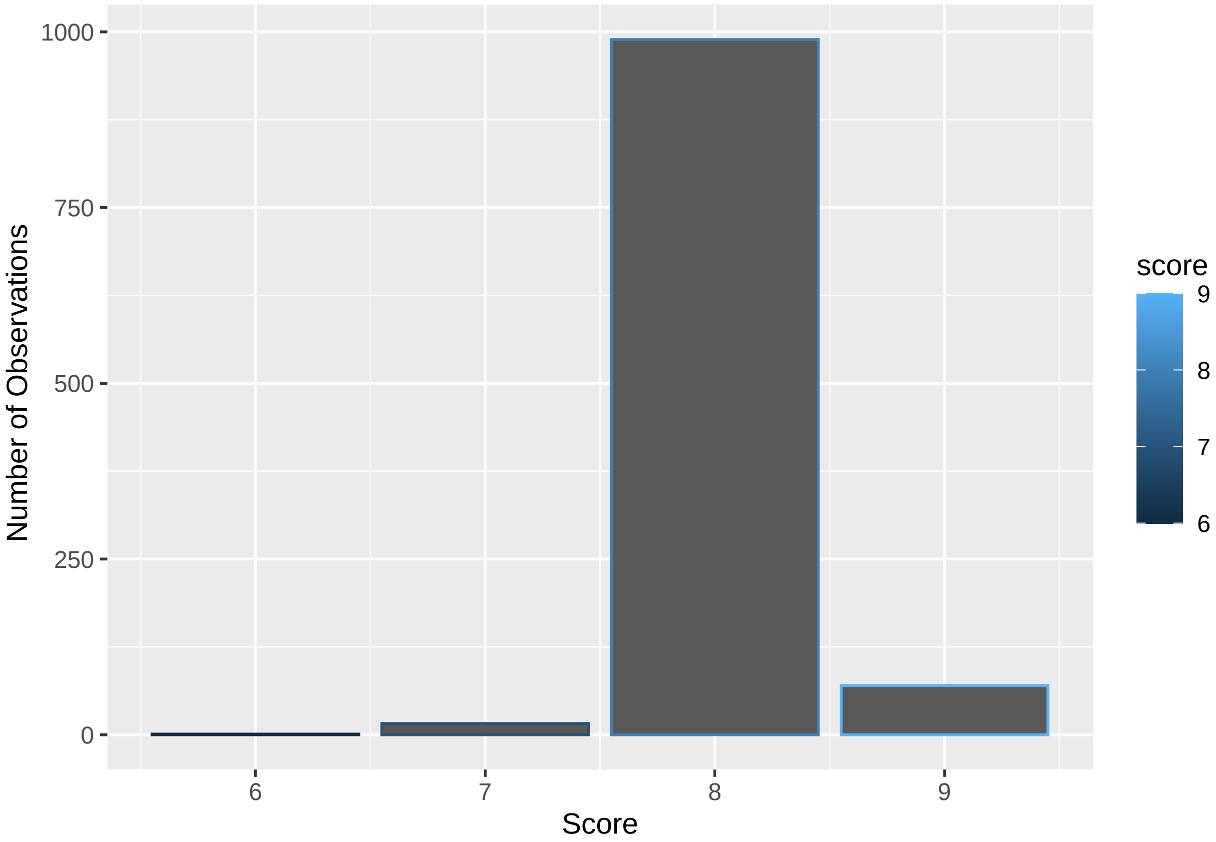
```
##                        Arabica   Robusta
## altitude_mean_meters 0.5285171 0.5285171
## Aroma                0.6902329 0.6902329
## Flavor               0.6946293 0.6946293
## Aftertaste           0.7335036 0.7335036
## Acidity              0.7008278 0.7008278
## Body                   0.7446728 0.7446728
## Balance                0.6781527 0.6781527
## Uniformity             0.5165558 0.5165558
## Clean.Cup              0.5136050 0.5136050
## Sweetness              0.9957818 0.9957818
## Cupper.Points          0.7248693 0.7248693
## Total.Cup.Points       0.6289013 0.6289013
## score                  0.5251901 0.5251901
```

As all coffee are Arabica, it's clear that the variable `altitude` doesn't play an important role in predicting the quality of the coffee. As mentioned above, Arabica coffee tend to be sweetness so `Sweetness` is a not an important predictor of quality of a bean when all coffee beans are Arabica.

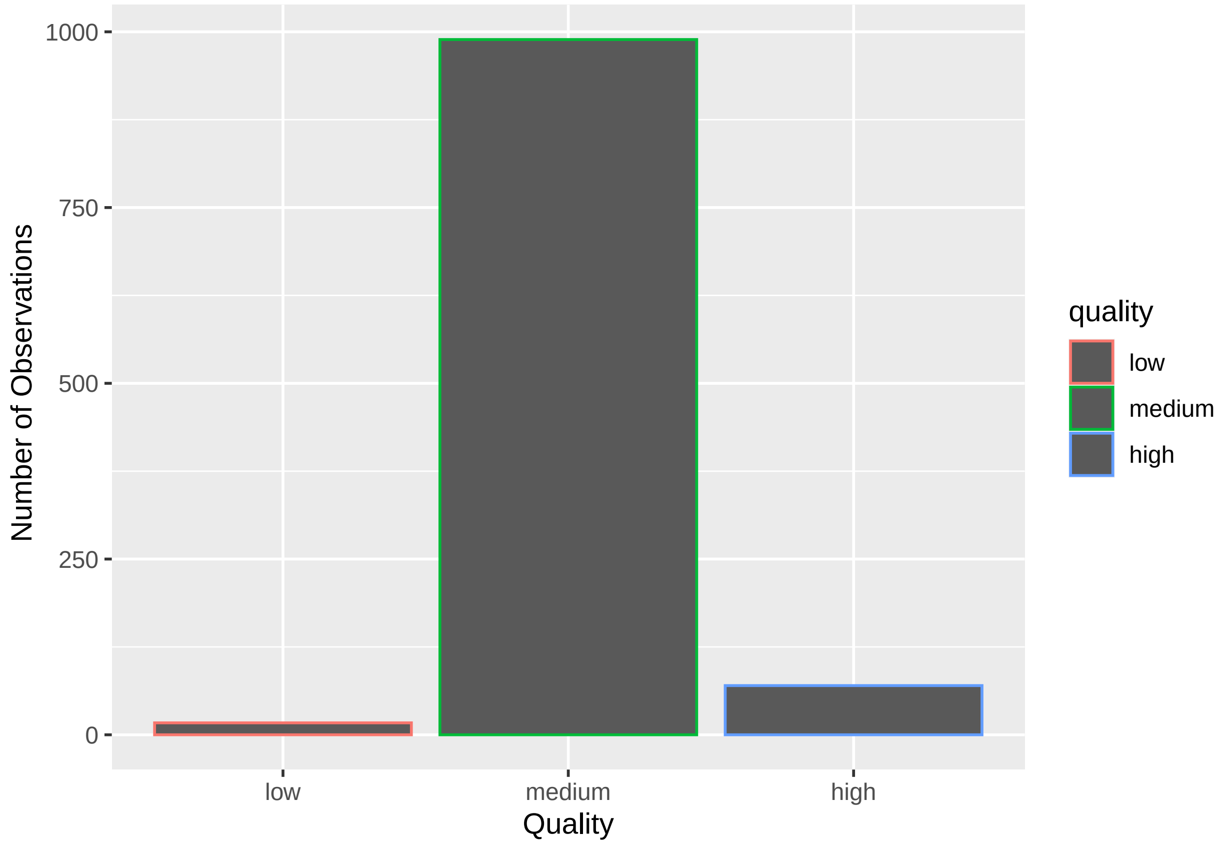
```
#### variable importance for predicting coffee quality
df_1 <- subset(coffee.data, select = -c(Species, Country.of.Origin))
var_imp_1 <- filterVarImp(df_1[, -14], df_1[, 14])
var_imp_1
```

```
##                           high       low    medium
## altitude_mean_meters 0.7752101 0.7752101 0.7239564
## Aroma                1.0000000 1.0000000 0.9665735
## Flavor               1.0000000 1.0000000 0.9805210
## Aftertaste           1.0000000 1.0000000 0.9768334
## Acidity              1.0000000 1.0000000 0.9616207
## Body                 0.9995798 0.9995798 0.9304059
## Balance              1.0000000 1.0000000 0.9812348
## Uniformity           0.8428571 0.8428571 0.8267115
## Clean.Cup            0.9701681 0.9701681 0.9630346
## Sweetness            0.7512605 0.7512605 0.7374948
## Cupper.Points        1.0000000 1.0000000 0.9764171
## Total.Cup.Points     1.0000000 1.0000000 1.0000000
## score                1.0000000 1.0000000 1.0000000
```

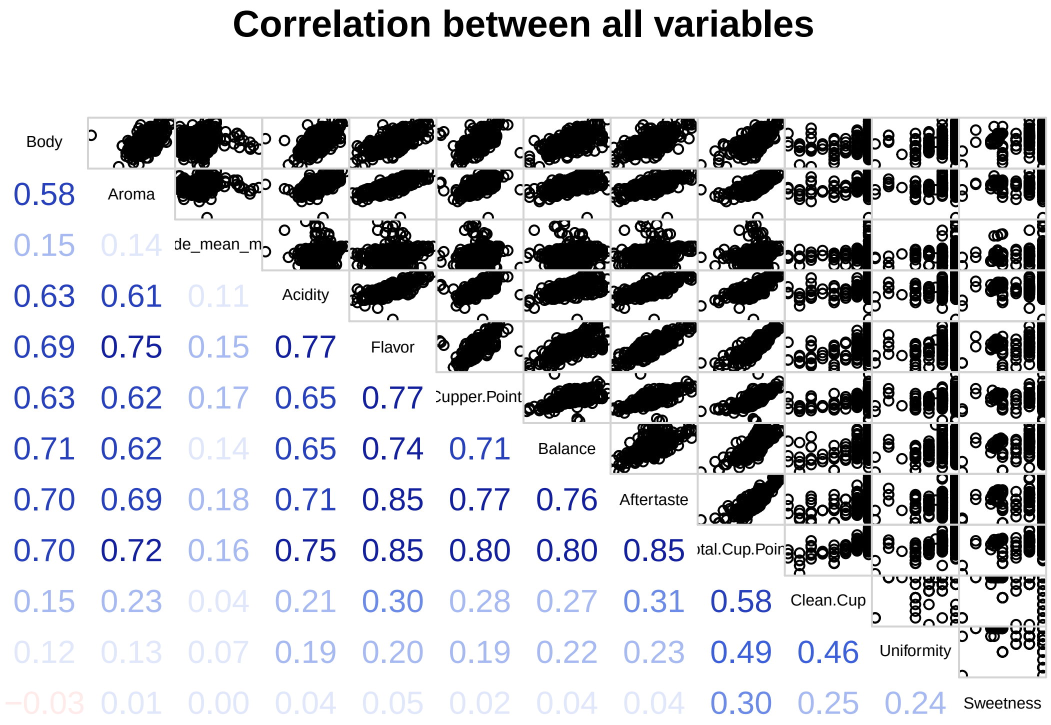
```
## quality distribution

coffee.data %>% group_by(score) %>% summarize(n = n()) %>% ggplot(aes(score,
    n, colour = score)) + geom_bar(stat = "identity") + labs(Title = "Distribution of Score ",
    x = "Score", y = " Number of Observations")
```



```
coffee.data %>% group_by(quality) %>% summarize(n = n()) %>%
    mutate(quality = fct_relevel(quality, "low", "medium", "high")) %>%
    ggplot(aes(quality, n, colour = quality)) + geom_bar(stat = "identity") +
    labs(Title = "Distribution of quality", x = "Quality", y = " Number of Observations")
```

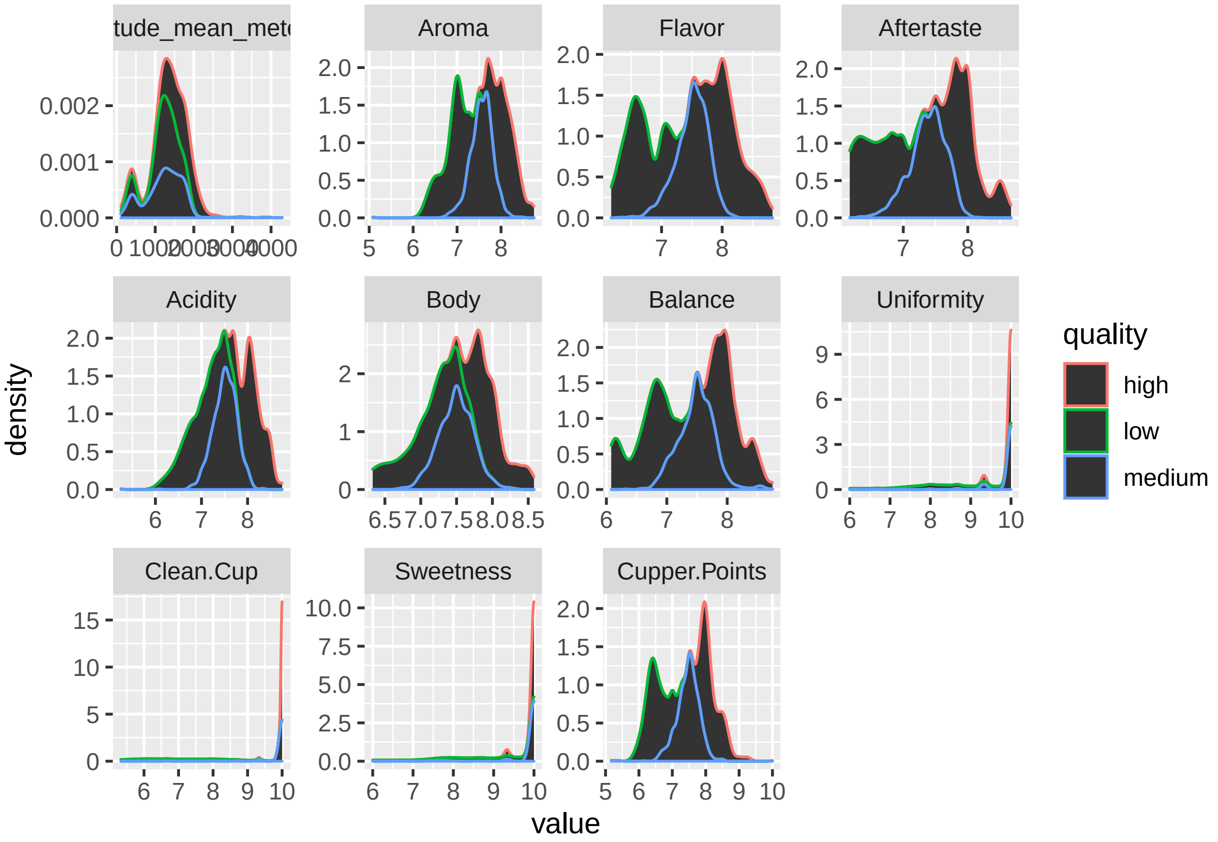


```
coffee.data %>% select(Aroma, Flavor, Aftertaste, Acidity, Body,
    Balance, Uniformity, Clean.Cup, Sweetness, Cupper.Points,
    altitude_mean_meters, Total.Cup.Points) %>% corrgram(order = TRUE,
    lower.panel = panel.cor, upper.panel = panel.pts, main = " Correlation between all variables ")
```



All variables are somehow related with exception of `altitude`. The `altitude` which does not have a direct effect of the quality of coffee it affects mainly the cultivation of beans.

```
tt <- melt(coffee.data, id.vars = "quality", measure.vars = c("altitude_mean_meters",
    "Aroma", "Flavor", "Aftertaste", "Acidity", "Body", "Balance",
    "Uniformity", "Clean.Cup", "Sweetness", "Cupper.Points"))
ggplot(tt, aes(x = value, colour = quality)) + stat_density() +
    facet_wrap(variable ~ ., scales = "free")
```



We visualize the probability distributions or density functions of each variable. We observe from these density distributions that variables : `Clean.Cup`, `Sweetness` and `Uniformity` are skewed. The skew in these distributions is explained by the extreme values and score in the data.

Modeling

Dataset is partitioned in train set (70%) and test set (30%). These two sets are standardized `train_s` and `test_s` and normalized `train_n` and `train_n`. Linear regression are sensitive to data so we trained our models on both standardized and normalized sets. Then, we compare RMSE, r-squared and adjusted r-squared results. The model with the lowest RMSE and highest r-squared and adjusted r-squared is selected.

```
set.seed(123)
options(scipen = 4)  ## remove scientific notation
#### test set will be 30% fo the data Reduce columns
coffee_data <- subset(coffee.data, select = -c(Country.of.Origin,
    Total.Cup.Points))
test_index <- createDataPartition(coffee_data$Species, times = 1,
    p = 0.3, list = FALSE)
train_coffee <- coffee_data %>% slice(-test_index)
test_coffee <- coffee_data %>% slice(test_index)
```

```
### standardize data using preprocess function from caret
### package
s <- preProcess(train_coffee)
train_coffee_s <- predict(s, train_coffee)
test_coffee_s <- predict(s, test_coffee)

### Normalized data using preprocess function from caret
### package

n <- preProcess(train_coffee, method = "range")
train_coffee_n <- predict(n, train_coffee)
test_coffee_n <- predict(n, test_coffee)
```

Model 1: Predicting species with linear regression

We will start by predicting the species of coffee using linear regression models that take the form of $Y = f(X)+\epsilon$ for an unknown function f where ϵ is a mean-zero random error. If f is a linear function then we can write our multiple regression model as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

where X_i represents the ith predictor and β_i represents the association between the variable and the response. β_i represent the slope of ith predictor so we can interpret it as the variation of Y moving X_i one unit assuming all other predictors as fixed.

Models are resumed in the following table:

```
model <- data.frame(Model = "Model 1", Data = " Standardized",
    Predictors = " Quality measures without altitude", Outcome = "Species")
model <- rbind(model, data.frame(Model = "Model 2", Data = " Standardized",
    Predictors = " Quality measures with altitude", Outcome = "Species"))
model <- rbind(model, data.frame(Model = "Model 3", Data = " Standardized",
    Predictors = " Quality measures with altitude and score",
    Outcome = "Species"))
model <- rbind(model, data.frame(Model = "Model 4", Data = " Normalized",
    Predictors = "Quality measures without altitude", Outcome = "Species"))
model <- rbind(model, data.frame(Model = "Model 5", Data = " Normalized",
    Predictors = " Quality measures with altitude", Outcome = "Species"))
model <- rbind(model, data.frame(Model = "Model 6", Data = " Normalized",
    Predictors = " Quality measures with altitude and score",
    Outcome = "Species"))
kable(model)
```

Model	Data	Predictors	Outcome
Model 1	Standardized	Quality measures without altitude	Species
Model 2	Standardized	Quality measures with altitude	Species
Model 3	Standardized	Quality measures with altitude and score	Species
Model 4	Normalized	Quality measures without altitude	Species
Model 5	Normalized	Quality measures with altitude	Species
Model 6	Normalized	Quality measures with altitude and score	Species

```
### Species prediction with standardized data model 1 linear
### regression without altitude 1 for arabica and 0 for robusta
### convert outcome to numeric
train_coffee_s <- train_coffee_s %>% mutate(Species = ifelse(Species ==
    "Arabica", 1, 0))
```

```
### linear regression
data.train_1 <- subset(train_coffee_s, select = -c(score, altitude_mean_meters,
    quality))
data.test_1 <- subset(test_coffee_s, select = -c(score, altitude_mean_meters,
    quality))
species_lm_1 <- lm(Species ~ ., data = data.train_1)
species_lm_hat_1 <- predict(species_lm_1, data.test_1)
# convert the outcome to factor
pred_species_1 <- factor(ifelse(species_lm_hat_1 > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_1 <- caret::confusionMatrix(pred_species_1, data.test_1$Species)
rmse_model_1 <- RMSE(as.numeric(data.test_1$Species), as.numeric(pred_species_1))
RMSE_results <- data.frame(Model = " 1 ", Data = "Standardized",
    RMSE = rmse_model_1, r.squared = summary(species_lm_1)$r.squared,
    Adjusted.r.squared = summary(species_lm_1)$adj.r.squared)

### Model 2: Predicting species based on quality measures with
### altitude {#model2}


### model 2 : linear regression with altitude
data.train_2 <- subset(train_coffee_s, select = -c(score, quality))
data.test_2 <- subset(test_coffee_s, select = -c(score, quality))
species_lm_2 <- lm(Species ~ ., data = data.train_2)
species_lm_hat_2 <- predict(species_lm_2, data.test_2)
# summary(species_lm_2) convert the outcome to factor
pred_species_2 <- factor(ifelse(species_lm_hat_2 > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_2 <- caret::confusionMatrix(pred_species_2, data.test_2$Species)
rmse_model_2 <- RMSE(as.numeric(data.test_2$Species), as.numeric(pred_species_2))
RMSE_results <- rbind(RMSE_results, data.frame(Model = " 2 ",
    Data = "Standardized", RMSE = rmse_model_2, r.squared = summary(species_lm_2)$r.squared,
    Adjusted.r.squared = summary(species_lm_2)$adj.r.squared))
### Model 3: Predicting species based on quality measures with
### altitude and score {#model3} model 3 : linear regression
### with score
data.train_3 <- subset(train_coffee_s, select = -c(quality))
data.test_3 <- subset(test_coffee_s, select = -c(quality))
species_lm_3 <- lm(Species ~ ., data = data.train_3)
species_lm_hat_3 <- predict(species_lm_3, data.test_3)
# summary(species_lm_3) convert the outcome to factor
pred_species_3 <- factor(ifelse(species_lm_hat_3 > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_3 <- caret::confusionMatrix(pred_species_3, data.test_3$Species)
rmse_model_3 <- RMSE(as.numeric(data.test_3$Species), as.numeric(pred_species_3))
RMSE_results <- rbind(RMSE_results, data.frame(Model = " 3 ",
    Data = "Standardized", RMSE = rmse_model_3, r.squared = summary(species_lm_3)$r.squared,
    Adjusted.r.squared = summary(species_lm_3)$adj.r.squared))
################### Model 4: Predicting species based on quality measures
################### without altitude ({#model4}) model 1.n linear regression
```

```r
################### without altitude 1 for arabica and 0 for robusta convert
################### outcome to numeric
train_coffee_n <- train_coffee_n %>% mutate(Species = ifelse(Species ==
    "Arabica", 1, 0))
### linear regression
data.train_1n <- subset(train_coffee_n, select = -c(score, altitude_mean_meters,
    quality))
data.test_1n <- subset(test_coffee_n, select = -c(score, altitude_mean_meters,
    quality))
species_lm_1n <- lm(Species ~ ., data = data.train_1n)
species_lm_hat_1n <- predict(species_lm_1n, data.test_1n)
# convert the outcome to factor
pred_species_1n <- factor(ifelse(species_lm_hat_1n > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_1n <- caret::confusionMatrix(pred_species_1n, data.test_1n$Species)
rmse_model_1n <- RMSE(as.numeric(data.test_1n$Species), as.numeric(pred_species_1n))
RMSE_results <- rbind(RMSE_results, data.frame(Model = " 4 ",
    Data = "Normalized", RMSE = rmse_model_1n, r.squared = summary(species_lm_1n)$r.squared,
    Adjusted.r.squared = summary(species_lm_1n)$adj.r.squared))

### Model 5: Predicting species based on quality measures
### without altitude ({#model5}) model 2.n linear regression
### with altitude
data.train_2n <- subset(train_coffee_n, select = -c(score, quality))
data.test_2n <- subset(test_coffee_n, select = -c(score, quality))
species_lm_2n <- lm(Species ~ ., data = data.train_2n)
species_lm_hat_2n <- predict(species_lm_2n, data.test_2n)
# summary(species_lm_2) convert the outcome to factor
pred_species_2n <- factor(ifelse(species_lm_hat_2n > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_2n <- caret::confusionMatrix(pred_species_2n, data.test_2n$Species)
rmse_model_2n <- RMSE(as.numeric(data.test_2n$Species), as.numeric(pred_species_2n))
RMSE_results <- rbind(RMSE_results, data.frame(Model = "5", Data = "Normalized",
    RMSE = rmse_model_2n, r.squared = summary(species_lm_2n)$r.squared,
    Adjusted.r.squared = summary(species_lm_2n)$adj.r.squared))
## Model 6: Predicting species based on quality measures with
## altitude and score {{#model6}}

data.train_3n <- subset(train_coffee_n, select = -c(quality))
data.test_3n <- subset(test_coffee_n, select = -c(quality))
species_lm_3n <- lm(Species ~ ., data = data.train_3n)
species_lm_hat_3n <- predict(species_lm_3n, data.test_3n)
# summary(species_lm_3) convert the outcome to factor
pred_species_3n <- factor(ifelse(species_lm_hat_3n > 0.5, "Arabica",
    "Robusta"))
### Evaluating the results
result_3n <- caret::confusionMatrix(pred_species_3n, data.test_3n$Species)
rmse_model_3n <- RMSE(as.numeric(data.test_3n$Species), as.numeric(pred_species_3n))
RMSE_results <- rbind(RMSE_results, data.frame(Model = "6", Data = "Normalized",
    RMSE = rmse_model_3n, r.squared = summary(species_lm_3n)$r.squared,
    Adjusted.r.squared = summary(species_lm_3n)$adj.r.squared))
```

```
########
```

Model : Predicting score using linear regression

In the follow, we remove Countries, total points species columns from the data. In addition, we remove all data related to robusta coffee. We standardized and normalized data for regression models. In classified algorithms, we used raw data.

```
##### Predicting the quality rate based on variables and altitude
##### for the Arabica coffee for linear regression i will use
##### score as outcome.
set.seed(123)
options(scipen = 4)  ## remove scientific notation
### test set will be 30% fo the data Reduce columns

arabica_coffee_data <- subset(coffee.data, select = -c(Country.of.Origin,
    Total.Cup.Points))
### removing Robusta coffee and species column
arabica_coffee_data <- arabica_coffee_data[arabica_coffee_data$Species ==
    "Arabica", -1]

#### Data partitioning
test_index <- createDataPartition(arabica_coffee_data$quality,
    times = 1, p = 0.3, list = FALSE)
train_set <- arabica_coffee_data %>% slice(-test_index)
test_set <- arabica_coffee_data %>% slice(test_index)


### standardize data using preprocess function from caret
### package
s <- preProcess(train_set)
train_s <- predict(s, train_set)
test_s <- predict(s, test_set)

### Normalized data using preprocess function from caret
### package
n <- preProcess(train_set, Model = "range")
train_n <- predict(n, train_set)
test_n <- predict(n, test_set)
```

Multiple regression models have been described above and have been applied to predict the score of coffee. The table below resumes data used and predictors.

```
model <- data.frame(Model = "Model 7", Data = " Standardized",
    Predictors = " Altitude", Outcome = "Score")
model <- rbind(model, data.frame(Model = "Model 8", Data = " Standardized",
    Predictors = " Quality measures", Outcome = "Score"))
model <- rbind(model, data.frame(Model = "Model 9", Data = " Standardized",
    Predictors = " Quality measures and altitude", Outcome = "Score"))
model <- rbind(model, data.frame(Model = "Model 10", Data = " Normalized",
    Predictors = "Altitude", Outcome = "Score"))
model <- rbind(model, data.frame(Model = "Model 11", Data = " Normalized",
    Predictors = " Quality measures ", Outcome = "Score"))
model <- rbind(model, data.frame(Model = "Model 12", Data = " Normalized",
    Predictors = " Quality measures and altitude", Outcome = "Score"))
```

```
model %>% kable()
```

Model	Data	Predictors	Outcome
Model 7	Standardized	Altitude	Score
Model 8	Standardized	Quality measures	Score
Model 9	Standardized	Quality measures and altitude	Score
Model 10	Normalized	Altitude	Score
Model 11	Normalized	Quality measures	Score
Model 12	Normalized	Quality measures and altitude	Score

```
####### model 4 linear regression with variable altitude only
####### train_s$quality<-as.integer(train_s$quality)
score_lm_4 <- lm(score ~ altitude_mean_meters, data = train_s)
score_lm_hat_4 <- predict(score_lm_4, test_s)
# summary(score_lm_4) Evaluating the results
rmse_model_4 <- RMSE(score_lm_hat_4, test_s$score)

results <- data.frame(Model = " 7 ", Data = "Standardized", RMSE = rmse_model_4,
    r.squared = summary(score_lm_4)$r.squared, Adjusted.r.squared = summary(score_lm_4)$adj.r.squared)

### Model 8: Predicting score based on quality measures
### {#model8} model 5 linear regression with quality measures
### only
train_s_5 <- subset(train_s, select = -c(altitude_mean_meters,
    quality))
test_s_5 <- subset(test_s, select = -c(altitude_mean_meters,
    quality))
score_lm_5 <- lm(score ~ ., data = train_s_5)
score_lm_hat_5 <- predict(score_lm_5, test_s_5)
# summary(score_lm_5) Evaluating the results
rmse_model_5 <- RMSE(score_lm_hat_5, test_s_5$score)
results <- rbind(results, data.frame(Model = " 8", Data = "Standardized",
    RMSE = rmse_model_5, r.squared = summary(score_lm_5)$r.squared,
    Adjusted.r.squared = summary(score_lm_5)$adj.r.squared))
################## Model 9: Predicting score based on quality measures and
################## altitude {#model9} model 6 linear regression with quality
################## measures and altitude.  linear regression with all variable
score_lm_6 <- lm(score ~ . - quality, data = train_s)
score_lm_hat_6 <- predict(score_lm_6, test_s)
# summary(score_lm-6) Evaluating the results
rmse_model_6 <- RMSE(score_lm_hat_6, test_s$score)
results <- rbind(results, data.frame(Model = "9", Data = "Standardized",
    RMSE = rmse_model_6, r.squared = summary(score_lm_6)$r.squared,
    Adjusted.r.squared = summary(score_lm_6)$adj.r.squared))
######### Model 10: Predicting score based on quality measures
######### {#model10} normalized data model 7 linear regression with
######### variable altitude only
score_lm_7 <- lm(score ~ altitude_mean_meters - quality, data = train_n)
score_lm_hat_7 <- predict(score_lm_7, test_n)
# summary(score_lm_7) Evaluating the results
rmse_model_7 <- RMSE(score_lm_hat_7, test_n$score)
results <- rbind(results, data.frame(Model = " 10", Data = "Normalized",
    RMSE = rmse_model_7, r.squared = summary(score_lm_7)$r.squared,
    Adjusted.r.squared = summary(score_lm_7)$adj.r.squared))
```

```r
###### Model 11: Predicting score based on altitude {#model11}
###### model 8 linear regression with variable only
score_lm_8 <- lm(score ~ . - altitude_mean_meters - quality,
    data = train_n)
score_lm_hat_8 <- predict(score_lm_8, test_n)
# summary(score_lm_8) Evaluating the results
rmse_model_8 <- RMSE(score_lm_hat_8, test_n$score)
results <- rbind(results, data.frame(Model = " 11", Data = "Normalized",
    RMSE = rmse_model_8, r.squared = summary(score_lm_8)$r.squared,
    Adjusted.r.squared = summary(score_lm_8)$adj.r.squared))
################ Model 12: Predicting score based on quality measure and
################ altitude {#model12} model 9 linear regression with
################ variables
score_lm_9 <- lm(score ~ ., data = train_n)
score_lm_hat_9 <- predict(score_lm_9, test_n)
# summary(score_lm_9) Evaluating the results
rmse_model_9 <- RMSE(score_lm_hat_9, test_n$score)
results <- rbind(results, data.frame(Model = "12", Data = "Normalized",
    RMSE = rmse_model_9, r.squared = summary(score_lm_9)$r.squared,
    Adjusted.r.squared = summary(score_lm_9)$adj.r.squared))
################
```

Model 13: Predicting quality using Knn model

K-Nearest Neighbors algorithm is a supervised machine learning algorithm simple and easy to use. Knn assume that similar data points are neighbors (close to each others). Knn calculated the distance between neighbor data points called proximity. To choose the best k that minimizes the number of errors, we run KNN algorithm several times with different k.

```r
####### model 10 knn knn works with discrete variables and
####### recommend standardize data 3 with classification models i
####### will quality variable instead of score
train_s <- subset(train_s, select = -c(score))
test_s <- subset(test_s, select = -c(score))

control <- trainControl(method = "cv", number = 10, p = 0.9)
train_knn <- train(quality ~ ., method = "knn", data = train_s,
    tuneGrid = data.frame(k = seq(3, 71, 2)), trControl = control)
##### choosing the best k
ks <- train_knn$bestTune
#### ploting accuracy
knn_plot <- ggplot(train_knn, highlight = TRUE)

pred_knn <- predict(train_knn, test_s, k = ks)
cm1 <- confusionMatrix(pred_knn, test_s$quality)
```

Model 14: Predicting quality using rpart model

A decision tree model is a tree-like graph including probability event outcomes. It's an intuitive algorithm that is easily applied in modeling[2]. It uses tree representation to solve the problem. Starting from the root and going down each leaf node represents a decision or terminal node.

[2] https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4

```r
train_set <- subset(train_set, select = -c(score))
test_set <- subset(test_set, select = -c(score))
train_rpart <- train(quality ~ ., method = "rpart", tuneGrid = data.frame(cp = seq(0,
    0.05, len = 25)), data = train_set)
# ggplot(train_rpart)

cp <- train_rpart$bestTune
# fancyRpartPlot(train_rpart$finalModel)
pred_rpart <- predict(train_rpart, test_set)
cm2 <- confusionMatrix(pred_rpart, test_set$quality)
# cm2$overall
```

Model 15: Predicting quality using random forest

Random forests are generally an improved version of decision trees. The forest builds multiple decision trees and merge them to get more accurate and stable outcome. Random forests' biggest advantage is that it can be used for both regression and classification. Here, we used it as a classifier and applied it to predict the quality of the coffee.

```r
########### model 12 random forest choose the best nodesize
nodesize <- seq(1, 150, 10)
accuracy <- sapply(nodesize, function(ns) {
    train(quality ~ ., method = "rf", tuneGrid = data.frame(mtry = 2),
        data = train_set)$results$Accuracy
})
# qplot(nodesize,accuracy) run the model for the best
# nodesize
model_forest <- randomForest(quality ~ ., data = train_set, nodesize = nodesize[which.max(accuracy)])
pred_model_forest <- predict(model_forest, test_set)
# confusionMatrix(pred_model_forest,test_set$quality)
# plot(model_forest)
```

Results

Linear regression for predicting coffee species

All predicted scores shows the same RMSE. As we know, RMSE is a measure data concentration and R-squared is a measure of fit. From the observations of all RMSE values and R-squared, we have concluded that the best linear regression model. Noteworthy, the regression model provides a reasonable but not perfect fit to the data because 0.67 is not to close too 1.

```r
kable(RMSE_results)
```

Model	Data	RMSE	r.squared	Adjusted.r.squared
1	Standardized	0.1111111	0.6739291	0.6695287
2	Standardized	0.1111111	0.6740123	0.6691666
3	Standardized	0.1111111	0.6740362	0.6687431
4	Normalized	0.1111111	0.6739291	0.6695287
5	Normalized	0.1111111	0.6740123	0.6691666
6	Normalized	0.1111111	0.6740362	0.6687431

Linear regression for predicting coffee score

All predicted scores shows different RMSE. The best model has the lowest one. Therefore, when the data is normalized, predictors including quality measures and altitude provides a perfect fit to the data. About

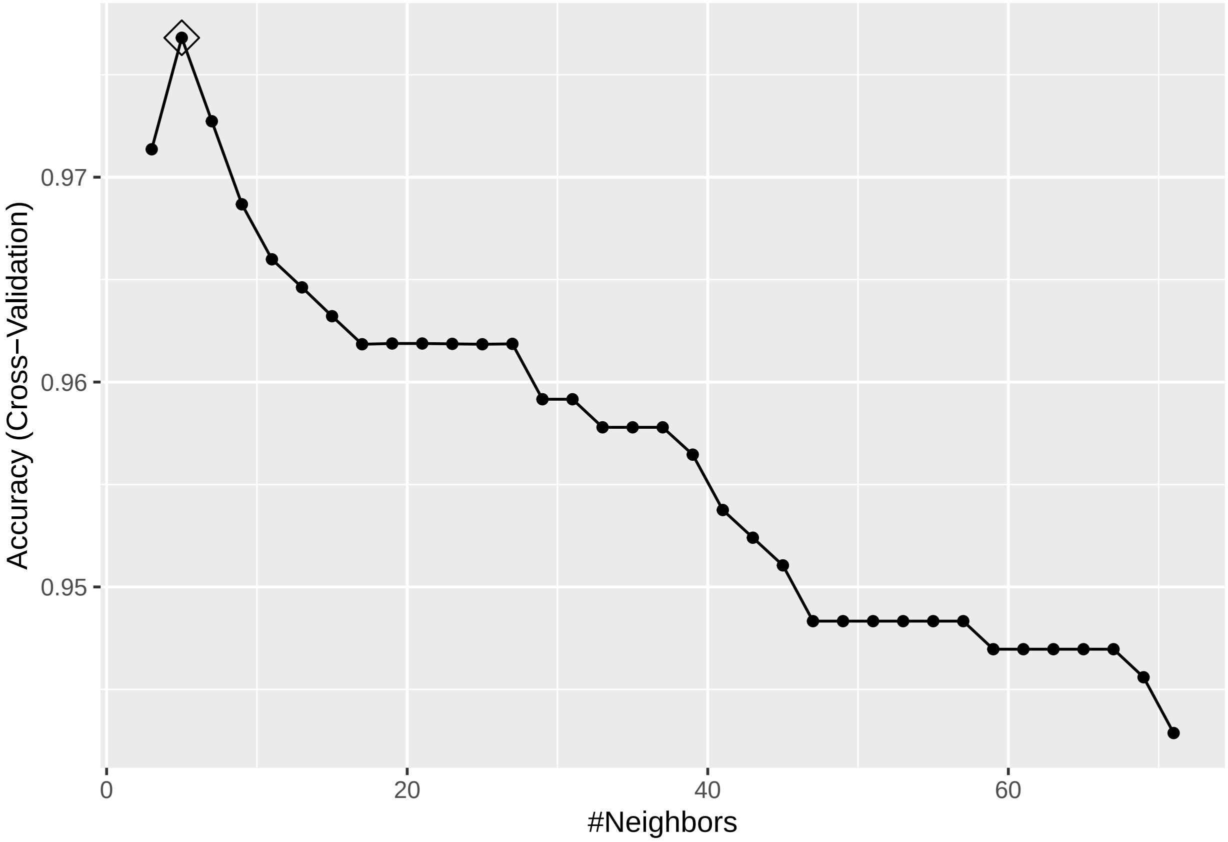
99% of the variability in the data can be explained by the fitted linear regression model. The high adjusted R-squared value means adding new predictor improveed our regression model.

```
kable(results)
```

Model	Data	RMSE	r.squared	Adjusted.r.squared
7	Standardized	0.9837560	0.0168657	0.0155244
8	Standardized	0.7643157	0.4466669	0.4390242
9	Standardized	0.7622242	0.4475609	0.4391558
10	Normalized	0.9837560	0.0168657	0.0155244
11	Normalized	0.7643157	0.4466669	0.4390242
12	Normalized	0.0648084	0.9879322	0.9877146

Knn

```
knn_plot <- ggplot(train_knn, highlight = TRUE)
knn_plot
```



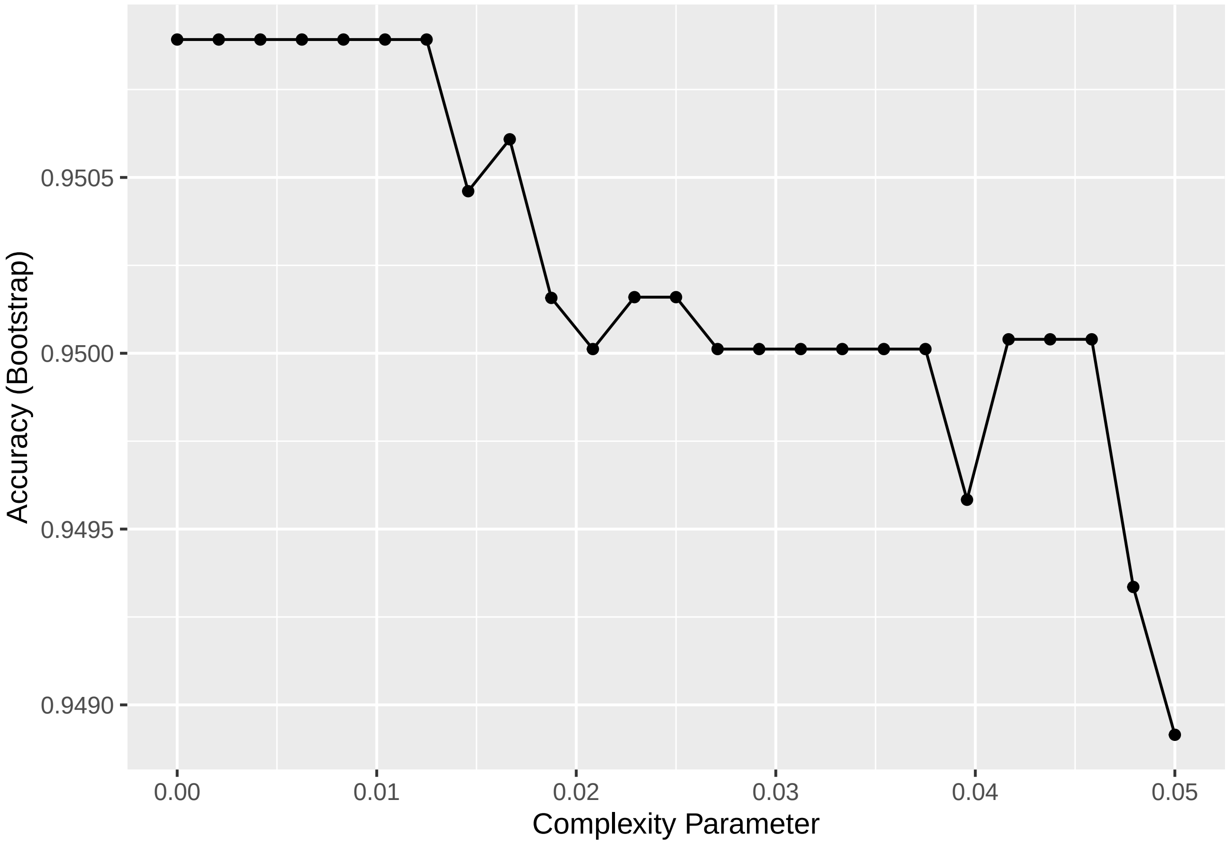
```
pred_knn <- predict(train_knn, test_s, k = ks)
cm1 <- confusionMatrix(pred_knn, test_s$quality)
cm1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low medium
##     high     15   0      0
##     low       0   2      0
##     medium    6   4    290
##
## Overall Statistics
##
##                Accuracy : 0.9685
##                  95% CI : (0.9428, 0.9848)
##     No Information Rate : 0.9148
##     P-Value [Acc > NIR] : 0.00009951
##
##                   Kappa : 0.7592
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: high Class: low Class: medium
## Sensitivity              0.71429   0.333333        1.0000
## Specificity              1.00000   1.000000        0.6296
## Pos Pred Value           1.00000   1.000000        0.9667
## Neg Pred Value           0.98013   0.987302        1.0000
## Prevalence               0.06625   0.018927        0.9148
## Detection Rate           0.04732   0.006309        0.9148
## Detection Prevalence     0.04732   0.006309        0.9464
## Balanced Accuracy        0.85714   0.666667        0.8148
```

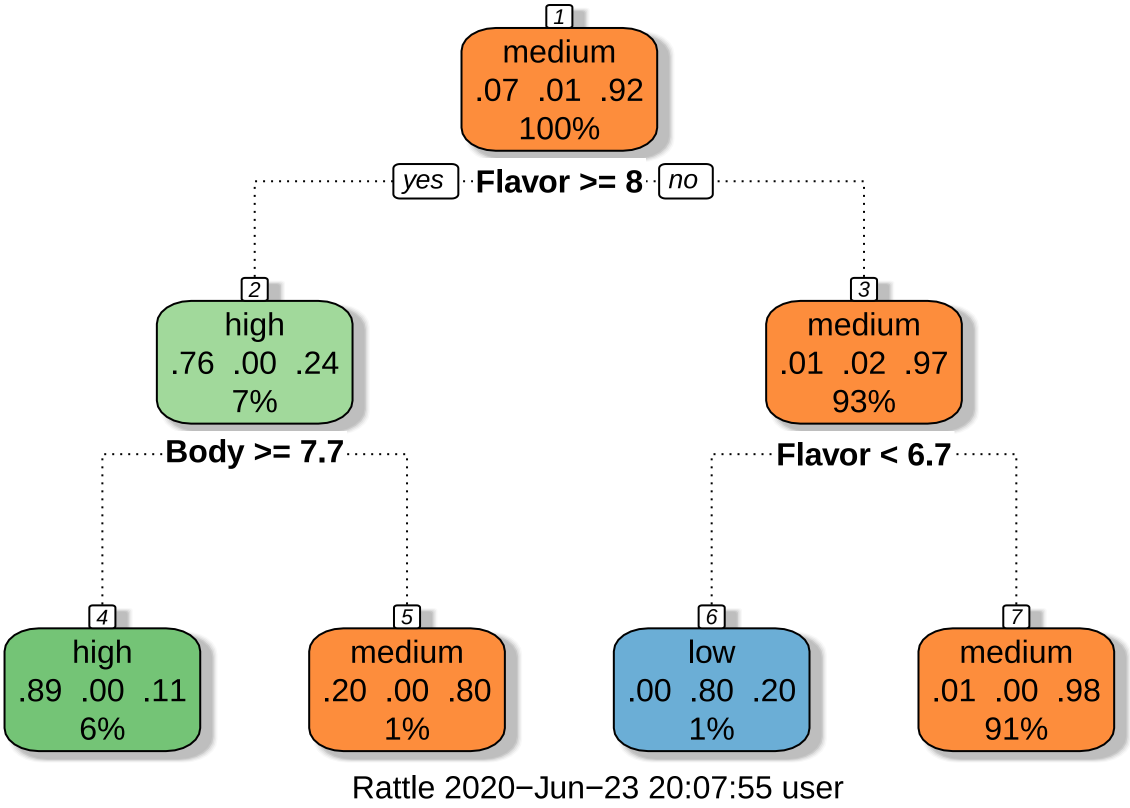
In Knn model, we get a high accuracy but sensitivity for `low` quality is low. It might be the result of the number of `low` quality coffee in our dataset.

Regression tree-rpart

```
ggplot(train_rpart)
```



```
fancyRpartPlot(train_rpart$finalModel)
```



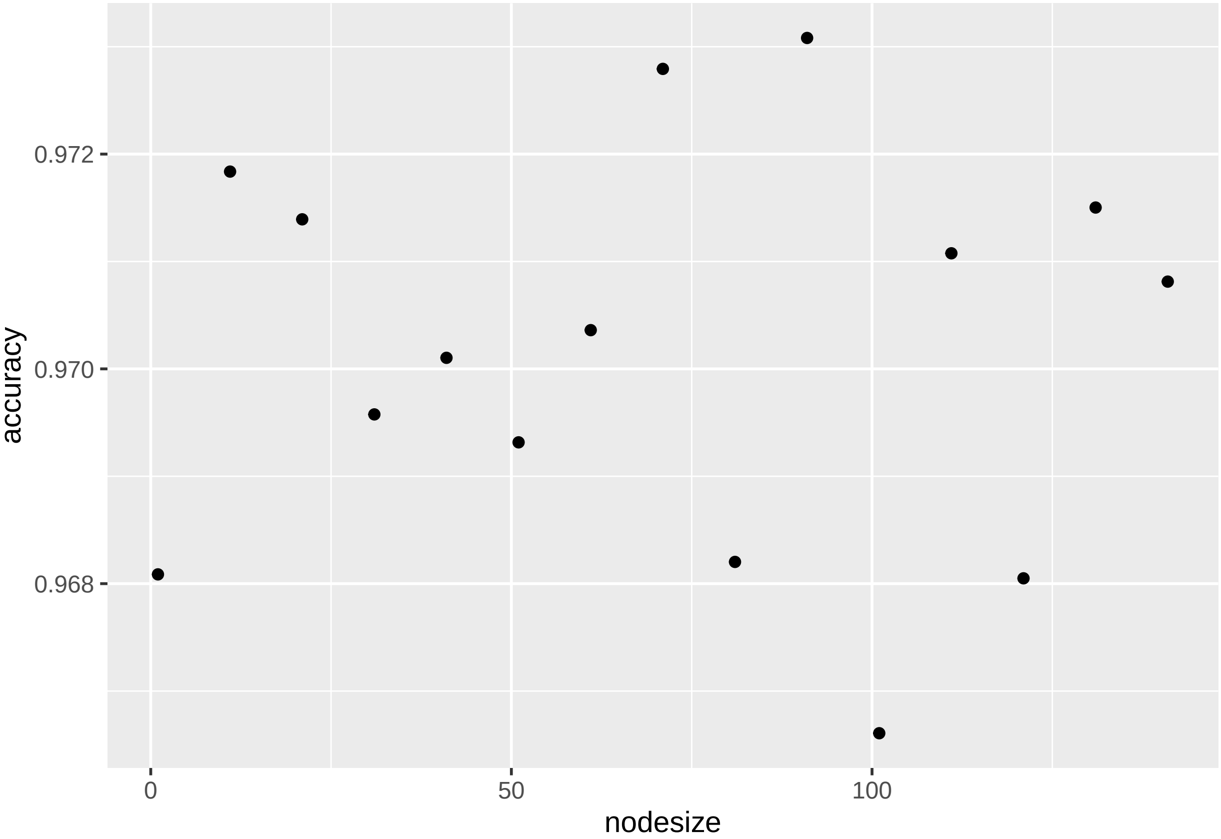
```
cm2$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9463722      0.6296729      0.9155234      0.9684531      0.9148265
## AccuracyPValue  McnemarPValue
##      0.0226431            NaN
```

The most important variables in the decision tree model are the `body` and `flavor`. Other variables seem with no effect on predicting quality of coffee. Routes for predicting quality are shown in the graph above.

Random Forest

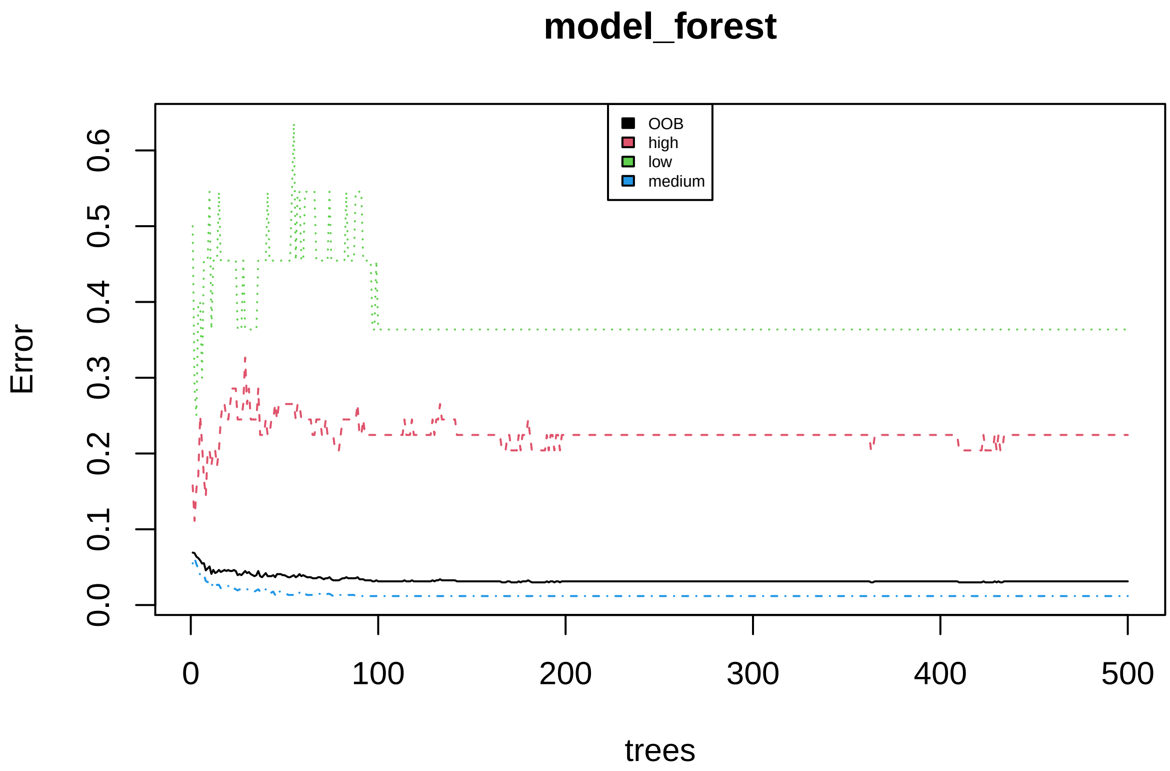
```
qplot(nodesize, accuracy)
```



```
confusionMatrix(pred_model_forest, test_set$quality)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low medium
##     high     13   0      2
##     low       0   3      1
##     medium    8   3    287
##
## Overall Statistics
##
##                Accuracy : 0.9558
##                  95% CI : (0.927, 0.9756)
##     No Information Rate : 0.9148
##     P-Value [Acc > NIR] : 0.003389
##
##                   Kappa : 0.6768
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: high Class: low Class: medium
## Sensitivity              0.61905   0.500000        0.9897
## Specificity              0.99324   0.996785        0.5926
## Pos Pred Value           0.86667   0.750000        0.9631
## Neg Pred Value           0.97351   0.990415        0.8421
## Prevalence               0.06625   0.018927        0.9148
## Detection Rate           0.04101   0.009464        0.9054
## Detection Prevalence     0.04732   0.012618        0.9401
## Balanced Accuracy        0.80615   0.748392        0.7911
```

```r
plot(model_forest)
legend("top", colnames(model_forest$err.rate), col = 1:4, cex = 0.5,
    fill = 1:4)
```



Random forest algorithm predicted the quality of coffee with high accuracy. The error for medium decrease and stabilize faster than high and low quality. Number of observations might be the reason.

Conclusion and discussion

This project started by exploring and visualizing coffee distributions. We built our models starting from the baseline till a more complicated model. Two datasets were used: Arabica and Robusta.

The machine learning models used predicted species of coffee with accuracy. This is due to a large number of Arabica observations. In addition, predicting the quality of coffee was attained with high accuracy. The performance of each model predicting the quality of coffee were discussed in the results section.

The data is imbalanced between species and quality. Thus, this report felt limited in the amount of data being trained and tested. It's interesting to gather more data and balance samples significantly to improve accuracy for low and high quality. A large size of observations might be bring advantages in preventing overfitting, identifying outliers and give more reliable results with more precision.

Since data were gathered from a website and it was created and cleaned by a person and not an official organization thus the results might have been impacted. Though, it's important to include machine learning in the coffee industry as it's a growing sector. Further work on a project such as this should include sales and prices and comparing more models on large data sets.

References

Andrea Trevisan. n.d. "What Are the Factors That Affect Coffee Quality?" Accessed June 22, 2020. https://blog.eureka.co.it/en/factors-that-affect-coffee-quality.

Andrew Menke. 2018. "The Global Coffee Industry." https://globaledge.msu.edu/blog/post/55607/the-global-coffee-industry.

Co, Blackout Coffee. n.d. "Coffee Characteristics: What Affects the Quality of Coffee Before You Brew It." *Blackout Coffee Co.* Accessed June 22, 2020. https://www.blackoutcoffee.com/blogs/the-reading-room/coffee-characteristics-what-affects-the-quality-of-coffee-before-you-brew-it.

Satista. n.d. "Coffee - Worldwide Statista Market Forecast." *Statista.* Accessed June 22, 2020. https://www.statista.com/outlook/30010000/100/coffee/worldwide.