# Capstone part 2 predicting coffee quality

Mirna Rossi

2022-09-22

*Introduction/overview/executive summary*

For the second project of the Capstone I am analyzing the "coffee quality" datasets by James LeDoux. In 2018, LeDoux scraped the pages of the Coffee Quality Institute's website (LeDoux, 2022). There are two datasets, one for the Arabica and one for the Robusta Species. According to SpecialCoffee (2018), a good coffee requires a balanced mix of these two species. However, the two datasets cannot be merged because they have different variable names. Also, the "robusta" dataset contains a limited number of observations – less than 30 – compared to more than 1000 observations in the "arabica" dataset. The two datasets are available in my git or here: https://github.com/jldbc/coffee-quality-database (https://github.com/jldbc/coffee-quality-database)

Therefore, I have used the "arabica" dataset to execute this Capstone project. Here following the data dictionary by Mock (2020).

total_cup_points= Total rating/points (0 - 100 scale)

species character= Species of coffee bean (arabica* or robusta)

owner character= Owner of the farm

country_of_origin= Where the bean came from

farm_name= Name of the farm

lot_number= Lot number of the beans tested

mill= Mill where the beans were processed

ico_number= International Coffee Organization number

company= Company name

altitude = Altitude

region= Region where bean came from

producer= Producer of the roasted bean

number_of_bags= Number of bags tested

bag_weight= Bag weight tested

in_country_partner= Partner for the country

harvest_year= When the beans were harvested (year)

grading_date= When the beans were graded

owner_1= Who owns the beans

variety= Variety of the beans

processing_method= Method for processing

aroma= Aroma grade

flavor= Flavor grade

aftertaste= Aftertaste grade

acidity= Acidity grade

body= Body grade

balance= Balance grade

uniformity= Uniformity grade

clean_cup= Clean cup grade

sweetness= Sweetness grade

cupper_points= Cupper Points

moisture= Moisture Grade

category_one_defects= Category one defects (count)

quakers = quakers

color= Color of bean

category_two_defects= Category two defects (count)

expiration= Expiration date of the beans

certification_body= Who certified it

certification_address= Certification body address

certification_contact= Certification contact

unit_of_measurement= Unit of measurement

altitude_low_meters= Altitude low meters

altitude_high_meters= Altitude high meters

altitude_mean_meters double Altitude mean meters

*for this dataset I only use “arabica”.

As we can see from the data dictionary, the coffee dataset provides a lot of information, with 43 variables and 1311 observations. A lot of data cleaning was required, for example some variables had too many missing values and they have been removed (e.g. “Lot.Number”). Another variable that required attention was the “altitude_mean_meters” because the height in meters of some values was off. Exploratory data analysis (EDA) is particularly useful also for determining the model approach for this analysis and other interesting insights. For example, how to interpret the importance of balance, acidity, aroma, and aftertaste to create a perfect flavor. Another example is the relationship between the variable “Total.Cup.Points” (that is the final grade the coffee cup has received), the place of production, the production method, the variety of coffee beans, and the color of coffee beans.

```
#Load packages
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.7      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.0      ✔ stringr 1.4.0
## ✔ readr   2.1.2      ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(skimr)) install.packages("skimr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: skimr
```

```
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(mosaicData)) install.packages("mosaicData", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: mosaicData
```

```
if(!require(ggcorrplot)) install.packages("ggcorrplot", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: ggcorrplot
```

```
if(!require(arsenal)) install.packages("arsenal", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: arsenal
```

```
if(!require(rlang)) install.packages("rlang", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rlang
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, splice
```

```
if(!require(stringr)) install.packages("stringr", repos = "http://cran.us.r-project.org")
if(!require(rmarkdown)) install.packages("rmarkdown", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: rmarkdown
```

```
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: knitr
```

```
if(!require(tidymodels)) install.packages("tidymodels", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidymodels
## ── Attaching packages ──────────────────────────────────── tidymodels 1.0.0 ──
## ✓ broom        1.0.0     ✓ rsample      1.0.0
## ✓ dials        1.0.0     ✓ tune         1.0.0
## ✓ infer        1.0.2     ✓ workflows    1.0.0
## ✓ modeldata    1.0.0     ✓ workflowsets 1.0.0
## ✓ parsnip      1.0.0     ✓ yardstick    1.0.0
## ✓ recipes      1.0.1
## ── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
## ✖ rlang::%@%()         masks purrr::%@%()
## ✖ rlang::as_function() masks purrr::as_function()
## ✖ scales::discard()    masks purrr::discard()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ recipes::fixed()     masks stringr::fixed()
## ✖ rlang::flatten()     masks purrr::flatten()
## ✖ rlang::flatten_chr() masks purrr::flatten_chr()
## ✖ rlang::flatten_dbl() masks purrr::flatten_dbl()
## ✖ rlang::flatten_int() masks purrr::flatten_int()
## ✖ rlang::flatten_lgl() masks purrr::flatten_lgl()
## ✖ rlang::flatten_raw() masks purrr::flatten_raw()
## ✖ rlang::invoke()      masks purrr::invoke()
## ✖ dplyr::lag()         masks stats::lag()
## ✖ yardstick::spec()    masks readr::spec()
## ✖ rlang::splice()      masks purrr::splice()
## ✖ recipes::step()      masks stats::step()
## • Learn how to get started at https://www.tidymodels.org/start/
```

```
if(!require(forcats)) install.packages("forcats", repos = "http://cran.us.r-project.org")
if(!require(doParallel)) install.packages("doParallel", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: doParallel
## Loading required package: foreach
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
##
## Loading required package: iterators
## Loading required package: parallel
```

```
if(!require(tictoc)) install.packages("tictoc", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tictoc
```

```
if(!require(vip)) install.packages("vip", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: vip
##
## Attaching package: 'vip'
##
## The following object is masked from 'package:utils':
##
##     vi
```

```
if(!require(remotes)) install.packages("remotes", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: remotes
```

```
if(!require(tinytext)) install.packages("tinytext", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tinytext
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'tinytext'
```

```
## Installing package into 'C:/Users/Mirna/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## Warning: package 'tinytext' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
```

```
options(tinytex.verbose = TRUE)

library(readr)
library(dplyr)
library(skimr)
library(ggplot2)
library(mosaicData) #for correlation plot
library(ggcorrplot) #for correlation
library(arsenal)
library(mosaicData)
library(rlang)
library(stringr)
library(rmarkdown)
library(knitr)
library(tidymodels)
library(forcats)
library(doParallel)
library(tictoc)
library(vip)
library(remotes)
library(tinytex)
```

```
##
## Attaching package: 'tinytex'
##
## The following object is masked from 'package:rlang':
##
##     check_installed
```

```
options( tinytex.verbose = TRUE)
```

*Methods/analysis*

After EDA and data cleaning I select Random Forest and LASSO models. Even if not required, I double check using linear regression and this approach has been helpful to define a good RMSE, thus a point to start from. Regarding the results of EDA, some groups looked similar and did not show great differences, therefore the decision to use decision trees to refine results.

```
#Load arabica
arabica <- read_csv("arabica_data_cleaned.csv")
```

```
## New names:
## Rows: 1311 Columns: 44
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (24): Species, Owner, Country.of.Origin, Farm.Name, Lot.Number, Mill, IC... dbl
## (20): ...1, Number.of.Bags, Aroma, Flavor, Aftertaste, Acidity, Body, Ba...
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...1`
```

```
dim(arabica)
```

```
## [1] 1311   44
```

```
#Count total missing values in each column
sapply(arabica, function(x) sum(is.na(x)))
```

```
##                 ...1               Species                 Owner 
##                    0                     0                     7 
##    Country.of.Origin             Farm.Name            Lot.Number 
##                    1                   356                  1041 
##                 Mill            ICO.Number               Company 
##                  307                   142                   209 
##             Altitude                Region              Producer 
##                  223                    57                   229 
##       Number.of.Bags            Bag.Weight     In.Country.Partner 
##                    0                     0                     0 
##         Harvest.Year          Grading.Date               Owner.1 
##                   47                     0                     7 
##              Variety     Processing.Method                 Aroma 
##                  201                   152                     0 
##               Flavor            Aftertaste               Acidity 
##                    0                     0                     0 
##                 Body               Balance            Uniformity 
##                    0                     0                     0 
##            Clean.Cup             Sweetness          Cupper.Points 
##                    0                     0                     0 
##     Total.Cup.Points              Moisture  Category.One.Defects 
##                    0                     0                     0 
##              Quakers                 Color  Category.Two.Defects 
##                    1                   216                     0 
##           Expiration    Certification.Body Certification.Address 
##                    0                     0                     0 
## Certification.Contact   unit_of_measurement    altitude_low_meters 
##                    0                     0                   227 
##  altitude_high_meters  altitude_mean_meters 
##                  227                   227
```

```
#Calculate percentage of missing values
colSums(is.na(arabica)) / nrow(arabica)
```

```
##                  ...1               Species                 Owner
##          0.0000000000          0.0000000000          0.0053394355
##     Country.of.Origin             Farm.Name            Lot.Number
##          0.0007627765          0.2715484363          0.7940503432
##                  Mill            ICO.Number               Company
##          0.2341723875          0.1083142639          0.1594202899
##              Altitude                Region              Producer
##          0.1700991609          0.0434782609          0.1746758200
##        Number.of.Bags            Bag.Weight    In.Country.Partner
##          0.0000000000          0.0000000000          0.0000000000
##          Harvest.Year          Grading.Date               Owner.1
##          0.0358504958          0.0000000000          0.0053394355
##               Variety     Processing.Method                 Aroma
##          0.1533180778          0.1159420290          0.0000000000
##                Flavor            Aftertaste               Acidity
##          0.0000000000          0.0000000000          0.0000000000
##                  Body               Balance            Uniformity
##          0.0000000000          0.0000000000          0.0000000000
##             Clean.Cup             Sweetness         Cupper.Points
##          0.0000000000          0.0000000000          0.0000000000
##      Total.Cup.Points              Moisture  Category.One.Defects
##          0.0000000000          0.0000000000          0.0000000000
##               Quakers                 Color  Category.Two.Defects
##          0.0007627765          0.1647597254          0.0000000000
##            Expiration    Certification.Body Certification.Address
##          0.0000000000          0.0000000000          0.0000000000
## Certification.Contact   unit_of_measurement   altitude_low_meters
##          0.0000000000          0.0000000000          0.1731502670
##  altitude_high_meters  altitude_mean_meters
##          0.1731502670          0.1731502670
```

```
#Lot.Number has 0.7940503432 NAs, this is too high so it must be removed
arabica_clean <- arabica[ , ! names(arabica) %in% c("Lot.Number")]

arabica_clean <- arabica_clean %>%
  mutate(id = row_number()) %>% #creata id column
  select(id, everything())
arabica_clean <- subset(arabica_clean, select = -c(...1 ))

#explore the dataset
skim(arabica_clean)
```

Data summary

| | |
|---|---|
| Name | arabica_clean |
| Number of rows | 1311 |
| Number of columns | 43 |
| _______________________ | |
| Column type frequency: | |
| character | 23 |
| numeric | 20 |
| ________________________ | |

| | |
|---|---|
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| Species | 0 | 1.00 | 7 | 7 | 0 | 1 | 0 |
| Owner | 7 | 0.99 | 3 | 50 | 0 | 305 | 0 |
| Country.of.Origin | 1 | 1.00 | 4 | 28 | 0 | 36 | 0 |
| Farm.Name | 356 | 0.73 | 1 | 73 | 0 | 557 | 0 |
| Mill | 307 | 0.77 | 1 | 77 | 0 | 448 | 0 |
| ICO.Number | 142 | 0.89 | 1 | 40 | 0 | 841 | 0 |
| Company | 209 | 0.84 | 3 | 73 | 0 | 270 | 0 |
| Altitude | 223 | 0.83 | 1 | 41 | 0 | 383 | 0 |
| Region | 57 | 0.96 | 2 | 76 | 0 | 343 | 0 |
| Producer | 229 | 0.83 | 1 | 100 | 0 | 674 | 0 |
| Bag.Weight | 0 | 1.00 | 1 | 8 | 0 | 56 | 0 |
| In.Country.Partner | 0 | 1.00 | 7 | 85 | 0 | 27 | 0 |
| Harvest.Year | 47 | 0.96 | 3 | 24 | 0 | 46 | 0 |
| Grading.Date | 0 | 1.00 | 13 | 20 | 0 | 558 | 0 |
| Owner.1 | 7 | 0.99 | 3 | 50 | 0 | 309 | 0 |
| Variety | 201 | 0.85 | 4 | 21 | 0 | 29 | 0 |
| Processing.Method | 152 | 0.88 | 5 | 25 | 0 | 5 | 0 |
| Color | 216 | 0.84 | 4 | 12 | 0 | 4 | 0 |
| Expiration | 0 | 1.00 | 13 | 20 | 0 | 557 | 0 |
| Certification.Body | 0 | 1.00 | 7 | 85 | 0 | 26 | 0 |
| Certification.Address | 0 | 1.00 | 40 | 40 | 0 | 30 | 0 |
| Certification.Contact | 0 | 1.00 | 40 | 40 | 0 | 27 | 0 |
| unit_of_measurement | 0 | 1.00 | 1 | 2 | 0 | 2 | 0 |

**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| id | 0 | 1.00 | 656.00 | 378.60 | 1 | 328.50 | 656.00 | 983.50 | 1311.00 | ▇▇▇▇▇ |
| Number.of.Bags | 0 | 1.00 | 153.89 | 129.73 | 0 | 14.50 | 175.00 | 275.00 | 1062.00 | ▇▃▁▁▁ |
| Aroma | 0 | 1.00 | 7.56 | 0.38 | 0 | 7.42 | 7.58 | 7.75 | 8.75 | ▁▁▁▁▇ |
| Flavor | 0 | 1.00 | 7.52 | 0.40 | 0 | 7.33 | 7.58 | 7.75 | 8.83 | ▁▁▁▁▇ |
| Aftertaste | 0 | 1.00 | 7.40 | 0.41 | 0 | 7.25 | 7.42 | 7.58 | 8.67 | ▁▁▁▁▇ |
| Acidity | 0 | 1.00 | 7.53 | 0.38 | 0 | 7.33 | 7.50 | 7.75 | 8.75 | ▁▁▁▁▇ |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| Body | 0 | 1.00 | 7.52 | 0.36 | 0 | 7.33 | 7.50 | 7.67 | 8.58 | ▁▁▁▁▇ |
| Balance | 0 | 1.00 | 7.52 | 0.41 | 0 | 7.33 | 7.50 | 7.75 | 8.75 | ▁▁▁▁▇ |
| Uniformity | 0 | 1.00 | 9.83 | 0.56 | 0 | 10.00 | 10.00 | 10.00 | 10.00 | ▁▁▁▁▇ |
| Clean.Cup | 0 | 1.00 | 9.83 | 0.77 | 0 | 10.00 | 10.00 | 10.00 | 10.00 | ▁▁▁▁▇ |
| Sweetness | 0 | 1.00 | 9.90 | 0.53 | 0 | 10.00 | 10.00 | 10.00 | 10.00 | ▁▁▁▁▇ |
| Cupper.Points | 0 | 1.00 | 7.50 | 0.47 | 0 | 7.25 | 7.50 | 7.75 | 10.00 | ▁▁▁▇▁ |
| Total.Cup.Points | 0 | 1.00 | 82.12 | 3.52 | 0 | 81.17 | 82.50 | 83.67 | 90.58 | ▁▁▁▁▇ |
| Moisture | 0 | 1.00 | 0.09 | 0.05 | 0 | 0.09 | 0.11 | 0.12 | 0.28 | ▃▇▅▁▁ |
| Category.One.Defects | 0 | 1.00 | 0.43 | 1.83 | 0 | 0.00 | 0.00 | 0.00 | 31.00 | ▇▁▁▁▁ |
| Quakers | 1 | 1.00 | 0.18 | 0.84 | 0 | 0.00 | 0.00 | 0.00 | 11.00 | ▇▁▁▁▁ |
| Category.Two.Defects | 0 | 1.00 | 3.59 | 5.35 | 0 | 0.00 | 2.00 | 4.00 | 55.00 | ▇▁▁▁▁ |
| altitude_low_meters | 227 | 0.83 | 1759.55 | 8767.85 | 1 | 1100.00 | 1310.64 | 1600.00 | 190164.00 | ▇▁▁▁▁ |
| altitude_high_meters | 227 | 0.83 | 1808.84 | 8767.19 | 1 | 1100.00 | 1350.00 | 1650.00 | 190164.00 | ▇▁▁▁▁ |
| altitude_mean_meters | 227 | 0.83 | 1784.20 | 8767.02 | 1 | 1100.00 | 1310.64 | 1600.00 | 190164.00 | ▇▁▁▁▁ |

```
summary(arabica_clean)
```

```
##        id          Species             Owner           Country.of.Origin 
##  Min.   :   1.0   Length:1311        Length:1311        Length:1311       
##  1st Qu.: 328.5   Class :character   Class :character   Class :character  
##  Median : 656.0   Mode  :character   Mode  :character   Mode  :character  
##  Mean   : 656.0                                                           
##  3rd Qu.: 983.5                                                           
##  Max.   :1311.0                                                           
##                                                                           
##   Farm.Name             Mill            ICO.Number          Company         
##  Length:1311        Length:1311        Length:1311        Length:1311       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    Altitude            Region            Producer         Number.of.Bags  
##  Length:1311        Length:1311        Length:1311        Min.   :   0.0  
##  Class :character   Class :character   Class :character   1st Qu.:  14.5  
##  Mode  :character   Mode  :character   Mode  :character   Median : 175.0  
##                                                           Mean   : 153.9  
##                                                           3rd Qu.: 275.0  
##                                                           Max.   :1062.0  
##                                                                           
##   Bag.Weight        In.Country.Partner Harvest.Year       Grading.Date      
##  Length:1311        Length:1311        Length:1311        Length:1311       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    Owner.1            Variety          Processing.Method      Aroma      
##  Length:1311        Length:1311        Length:1311        Min.   :0.000  
##  Class :character   Class :character   Class :character   1st Qu.:7.420  
##  Mode  :character   Mode  :character   Mode  :character   Median :7.580  
##                                                           Mean   :7.564  
##                                                           3rd Qu.:7.750  
##                                                           Max.   :8.750  
##                                                                          
##      Flavor        Aftertaste       Acidity           Body      
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  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   :7.518   Mean   :7.398   Mean   :7.533   Mean   :7.518  
##  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.   :0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
##  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   :7.518   Mean   : 9.833   Mean   : 9.833   Mean   : 9.903  
##  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    Moisture       Category.One.Defects
##  Min.   : 0.000   Min.   : 0.00    Min.   :0.00000   Min.   : 0.0000     
##  1st Qu.: 7.250   1st Qu.:81.17    1st Qu.:0.09000   1st Qu.: 0.0000     
```

```
##  Median : 7.500   Median :82.50    Median :0.11000   Median : 0.0000
##  Mean   : 7.498   Mean   :82.12    Mean   :0.08886   Mean   : 0.4264
##  3rd Qu.: 7.750   3rd Qu.:83.67    3rd Qu.:0.12000   3rd Qu.: 0.0000
##  Max.   :10.000   Max.   :90.58    Max.   :0.28000   Max.   :31.0000
##
##     Quakers           Color           Category.Two.Defects  Expiration
##  Min.   : 0.0000   Length:1311        Min.   : 0.000       Length:1311
##  1st Qu.: 0.0000   Class :character   1st Qu.: 0.000       Class :character
##  Median : 0.0000   Mode  :character   Median : 2.000       Mode  :character
##  Mean   : 0.1771                      Mean   : 3.592
##  3rd Qu.: 0.0000                      3rd Qu.: 4.000
##  Max.   :11.0000                      Max.   :55.000
##  NA's   :1
##  Certification.Body Certification.Address Certification.Contact
##  Length:1311        Length:1311           Length:1311
##  Class :character   Class :character      Class :character
##  Mode  :character   Mode  :character      Mode  :character
##
##
##
##
##  unit_of_measurement altitude_low_meters altitude_high_meters
##  Length:1311         Min.   :     1      Min.   :     1
##  Class :character    1st Qu.:  1100      1st Qu.:  1100
##  Mode  :character    Median :  1311      Median :  1350
##                      Mean   :  1760      Mean   :  1809
##                      3rd Qu.:  1600      3rd Qu.:  1650
##                      Max.   :190164      Max.   :190164
##                      NA's   :227         NA's   :227
##  altitude_mean_meters
##  Min.   :     1
##  1st Qu.:  1100
##  Median :  1311
##  Mean   :  1784
##  3rd Qu.:  1600
##  Max.   :190164
##  NA's   :227
```

```
colnames(arabica_clean)
```

```
##  [1] "id"                    "Species"               "Owner"
##  [4] "Country.of.Origin"     "Farm.Name"             "Mill"
##  [7] "ICO.Number"            "Company"               "Altitude"
## [10] "Region"                "Producer"              "Number.of.Bags"
## [13] "Bag.Weight"            "In.Country.Partner"    "Harvest.Year"
## [16] "Grading.Date"          "Owner.1"               "Variety"
## [19] "Processing.Method"     "Aroma"                 "Flavor"
## [22] "Aftertaste"            "Acidity"               "Body"
## [25] "Balance"               "Uniformity"            "Clean.Cup"
## [28] "Sweetness"             "Cupper.Points"         "Total.Cup.Points"
## [31] "Moisture"              "Category.One.Defects"  "Quakers"
## [34] "Color"                 "Category.Two.Defects"  "Expiration"
## [37] "Certification.Body"    "Certification.Address" "Certification.Contact"
## [40] "unit_of_measurement"   "altitude_low_meters"   "altitude_high_meters"
## [43] "altitude_mean_meters"
```

```
dim(arabica_clean)
```

```
## [1] 1311   43
```

```
#Fix altitude because the summary function shows max mean altidue = 190164, this is not
#possible since the highest mountains in the world are not that high, for example the height of Mount Everest
is 8,848 (Wikipedia Contributors, 2019).
coffees_altitude <- arabica_clean %>%
  select(altitude_mean_meters, Total.Cup.Points) %>%
  filter(altitude_mean_meters <= 10000) %>%
  group_by(altitude_mean_meters) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  arrange(cup_points) %>%
  arrange(desc(cup_points))
coffees_altitude
```

| altitude_mean_meters <dbl> | cup_points <dbl> |
|---:|---:|
| 2075.0000 | 89.77667 |
| 1635.0000 | 88.30667 |
| 1822.5000 | 88.25000 |
| 1905.0000 | 88.08000 |
| 609.6000 | 87.92000 |
| 1872.0000 | 87.92000 |
| 1943.0000 | 87.92000 |
| 2080.0000 | 87.58000 |
| 2019.0000 | 87.25000 |
| 2112.0000 | 87.08000 |

1-10 of 198 rows Previous **1** 2 3 4 5 6 ... 20 Next

```
altitude_points <- coffees_altitude %>%
  ggplot(aes(altitude_mean_meters, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("Altitude Mean Meters compared to Average Cup Points")+
  theme_gray()+
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
altitude_points
```

```
#I would not say that altitude is super relevant, but most coffees rated between 80 and 85 are produced at less than 2000 meters and some high rated coffees are produced between 2000 and 3000 meters.

#As a general approach, I want to check if there is any relationship between coffee Variety and Total.Cup.Points.

#Create scatterplot of variety vs. Total.Cup.Points, but most values are higher than 75 and there is an obvious outlier.
p <- ggplot(data=arabica_clean, aes(x= Variety, y= Total.Cup.Points, color= Variety)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
p
```

```
#Check boxplots, and here we see that some varieties stand out in terms of total cup points
l <- ggplot(data=arabica_clean, aes(x= Variety, y= Total.Cup.Points, color = Variety)) +
  geom_boxplot()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
l
```

```
#Barplot of Varieties versus count of total cup points
m <- ggplot(arabica_clean, aes(x= Variety, fill= Total.Cup.Points))+
  geom_bar(fill = "brown")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
m
```

```
#To make things easier, I rename the dataset
arab_coffee <- arabica_clean


#RATINGS
#Total.Cup.Points (quality rating 0-100) is relevant since it is a standard metric for coffee reviews. For example, according to Coffeereview website (n.d.), explains that coffee ratings higher than 97 equal to the best cups, while ratings of 85/86 are barely acceptable.Here I plot the outliers (three zeros) we have previously identified.

p <- arab_coffee %>%
  ggplot(aes(x = Total.Cup.Points)) +
  geom_boxplot(y = 0, fill = "brown", outlier.shape = NA) +
  geom_jitter(aes(y = 1), color = "brown",
              alpha = 0.3, height = 0.3, width = 0)
p
```

```
arab_coffee  %>%
  filter(Total.Cup.Points == min(Total.Cup.Points)) %>%
  glimpse()
```

```
## Rows: 1
## Columns: 43
## $ id                    <int> 1311
## $ Species               <chr> "Arabica"
## $ Owner                 <chr> "bismarck castro"
## $ Country.of.Origin     <chr> "Honduras"
## $ Farm.Name             <chr> "los hicaques"
## $ Mill                  <chr> "cigrah s.a de c.v."
## $ ICO.Number            <chr> "13-111-053"
## $ Company               <chr> "cigrah s.a de c.v"
## $ Altitude              <chr> "1400"
## $ Region                <chr> "comayagua"
## $ Producer              <chr> "Reinerio Zepeda"
## $ Number.of.Bags        <dbl> 275
## $ Bag.Weight            <chr> "69 kg"
## $ In.Country.Partner    <chr> "Instituto Hondureño del Café"
## $ Harvest.Year          <chr> "2017"
## $ Grading.Date          <chr> "April 28th, 2017"
## $ Owner.1               <chr> "Bismarck Castro"
## $ Variety               <chr> "Caturra"
## $ Processing.Method     <chr> NA
## $ Aroma                 <dbl> 0
## $ Flavor                <dbl> 0
## $ Aftertaste            <dbl> 0
## $ Acidity               <dbl> 0
## $ Body                  <dbl> 0
## $ Balance               <dbl> 0
## $ Uniformity            <dbl> 0
## $ Clean.Cup             <dbl> 0
## $ Sweetness             <dbl> 0
## $ Cupper.Points         <dbl> 0
## $ Total.Cup.Points      <dbl> 0
## $ Moisture              <dbl> 0.12
## $ Category.One.Defects  <dbl> 0
## $ Quakers               <dbl> 0
## $ Color                 <chr> "Green"
## $ Category.Two.Defects  <dbl> 2
## $ Expiration            <chr> "April 28th, 2018"
## $ Certification.Body    <chr> "Instituto Hondureño del Café"
## $ Certification.Address <chr> "b4660a57e9f8cc613ae5b8f02bfce8634c763ab4"
## $ Certification.Contact <chr> "7f521ca403540f81ec99daec7da19c2788393880"
## $ unit_of_measurement   <chr> "m"
## $ altitude_low_meters   <dbl> 1400
## $ altitude_high_meters  <dbl> 1400
## $ altitude_mean_meters  <dbl> 1400
```

```
arab_coffee <- arab_coffee %>% filter(Total.Cup.Points > 0)
p %+% arab_coffee
```

```
arab_coffee %>%
  ggplot(aes(x = Moisture)) +
  geom_boxplot(y = 0, fill = "brown", outlier.shape = NA) +
  geom_jitter(aes(y = 1), color = "brown",
              alpha = 0.3, height = 0.3, width = 0)
```

```
#Run a correlation to identify correlations between coffee variables
# select numeric variables
df <- dplyr::select_if(arab_coffee, is.numeric)

#Calculate the correlations
r <- cor(df, use="complete.obs")
round(r,2)
```

```
##                          id Number.of.Bags Aroma Flavor Aftertaste Acidity  Body
## id                     1.00          -0.09 -0.73  -0.84      -0.84   -0.76 -0.73
## Number.of.Bags        -0.09           1.00  0.02   0.03       0.04    0.06  0.07
## Aroma                 -0.73           0.02  1.00   0.74       0.69    0.61  0.57
## Flavor                -0.84           0.03  0.74   1.00       0.85    0.76  0.69
## Aftertaste            -0.84           0.04  0.69   0.85       1.00    0.70  0.69
## Acidity               -0.76           0.06  0.61   0.76       0.70    1.00  0.63
## Body                  -0.73           0.07  0.57   0.69       0.69    0.63  1.00
## Balance               -0.79           0.08  0.61   0.74       0.76    0.64  0.70
## Uniformity            -0.32           0.01  0.12   0.20       0.22    0.17  0.12
## Clean.Cup             -0.30           0.03  0.20   0.30       0.31    0.16  0.14
## Sweetness             -0.17          -0.05  0.08   0.16       0.17    0.09  0.07
## Cupper.Points         -0.79           0.05  0.61   0.77       0.76    0.64  0.63
## Total.Cup.Points      -0.86           0.04  0.70   0.84       0.84    0.72  0.68
## Moisture               0.17          -0.11 -0.10  -0.14      -0.18   -0.12 -0.21
## Category.One.Defects   0.09          -0.04 -0.10  -0.07      -0.10   -0.09 -0.03
## Quakers               -0.02           0.11  0.01   0.01       0.01   -0.01  0.00
## Category.Two.Defects   0.22          -0.03 -0.19  -0.23      -0.26   -0.18 -0.14
## altitude_low_meters    0.04          -0.03 -0.02  -0.01      -0.03    0.00 -0.02
## altitude_high_meters   0.04          -0.03 -0.02  -0.01      -0.03    0.00 -0.02
## altitude_mean_meters   0.04          -0.03 -0.02  -0.01      -0.03    0.00 -0.02
##                       Balance Uniformity Clean.Cup Sweetness Cupper.Points
## id                      -0.79      -0.32     -0.30     -0.17         -0.79
## Number.of.Bags           0.08       0.01      0.03     -0.05          0.05
## Aroma                    0.61       0.12      0.20      0.08          0.61
## Flavor                   0.74       0.20      0.30      0.16          0.77
## Aftertaste               0.76       0.22      0.31      0.17          0.76
## Acidity                  0.64       0.17      0.16      0.09          0.64
## Body                     0.70       0.12      0.14      0.07          0.63
## Balance                  1.00       0.22      0.26      0.14          0.71
## Uniformity               0.22       1.00      0.37      0.36          0.19
## Clean.Cup                0.26       0.37      1.00      0.44          0.28
## Sweetness                0.14       0.36      0.44      1.00          0.13
## Cupper.Points            0.71       0.19      0.28      0.13          1.00
## Total.Cup.Points         0.79       0.48      0.62      0.45          0.79
## Moisture                -0.23      -0.01     -0.04      0.03         -0.19
## Category.One.Defects    -0.08      -0.14     -0.14     -0.04         -0.06
## Quakers                  0.00       0.04      0.03      0.02          0.01
## Category.Two.Defects    -0.22      -0.10     -0.23     -0.04         -0.21
## altitude_low_meters     -0.02      -0.01     -0.01     -0.02         -0.02
## altitude_high_meters    -0.02      -0.01     -0.01     -0.02         -0.01
## altitude_mean_meters    -0.02      -0.01     -0.01     -0.02         -0.01
##                       Total.Cup.Points Moisture Category.One.Defects Quakers
## id                               -0.86     0.17                 0.09   -0.02
## Number.of.Bags                    0.04    -0.11                -0.04    0.11
## Aroma                             0.70    -0.10                -0.10    0.01
## Flavor                            0.84    -0.14                -0.07    0.01
## Aftertaste                        0.84    -0.18                -0.10    0.01
## Acidity                           0.72    -0.12                -0.09   -0.01
## Body                              0.68    -0.21                -0.03    0.00
## Balance                           0.79    -0.23                -0.08    0.00
## Uniformity                        0.48    -0.01                -0.14    0.04
## Clean.Cup                         0.62    -0.04                -0.14    0.03
## Sweetness                         0.45     0.03                -0.04    0.02
## Cupper.Points                     0.79    -0.19                -0.06    0.01
## Total.Cup.Points                  1.00    -0.16                -0.14    0.02
## Moisture                         -0.16     1.00                 0.07    0.02
## Category.One.Defects             -0.14     0.07                 1.00    0.00
## Quakers                           0.02     0.02                 0.00    1.00
```

```
## Category.Two.Defects           -0.27     0.16               0.37    0.02
## altitude_low_meters            -0.02     0.02              -0.01    0.10
## altitude_high_meters           -0.02     0.02              -0.01    0.10
## altitude_mean_meters           -0.02     0.02              -0.01    0.10
##                      Category.Two.Defects altitude_low_meters
## id                                   0.22                0.04
## Number.of.Bags                      -0.03               -0.03
## Aroma                               -0.19               -0.02
## Flavor                              -0.23               -0.01
## Aftertaste                          -0.26               -0.03
## Acidity                             -0.18                0.00
## Body                                -0.14               -0.02
## Balance                             -0.22               -0.02
## Uniformity                          -0.10               -0.01
## Clean.Cup                           -0.23               -0.01
## Sweetness                           -0.04               -0.02
## Cupper.Points                       -0.21               -0.02
## Total.Cup.Points                    -0.27               -0.02
## Moisture                             0.16                0.02
## Category.One.Defects                 0.37               -0.01
## Quakers                              0.02                0.10
## Category.Two.Defects                 1.00               -0.02
## altitude_low_meters                 -0.02                1.00
## altitude_high_meters                -0.02                1.00
## altitude_mean_meters                -0.02                1.00
##                      altitude_high_meters altitude_mean_meters
## id                                   0.04                 0.04
## Number.of.Bags                      -0.03                -0.03
## Aroma                               -0.02                -0.02
## Flavor                              -0.01                -0.01
## Aftertaste                          -0.03                -0.03
## Acidity                              0.00                 0.00
## Body                                -0.02                -0.02
## Balance                             -0.02                -0.02
## Uniformity                          -0.01                -0.01
## Clean.Cup                           -0.01                -0.01
## Sweetness                           -0.02                -0.02
## Cupper.Points                       -0.01                -0.01
## Total.Cup.Points                    -0.02                -0.02
## Moisture                             0.02                 0.02
## Category.One.Defects                -0.01                -0.01
## Quakers                              0.10                 0.10
## Category.Two.Defects                -0.02                -0.02
## altitude_low_meters                  1.00                 1.00
## altitude_high_meters                 1.00                 1.00
## altitude_mean_meters                 1.00                 1.00
```

```
ggcorrplot(r)
```

```
#There are relevant positive and negative correlations, especially related to Total.Cup.Points and aroma, fla
vor, aftertaste acidity, body, and balance. Therefore, I
#try to understand other relationships with high ratings and country of production.

coffees_reduced <- arab_coffee %>% select(Species, Country.of.Origin, Number.of.Bags,Total.Cup.Points) %>%
  group_by(Country.of.Origin) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  print(n=37) %>%
  arrange(cup_points)
```

```
## # A tibble: 37 × 2
##    Country.of.Origin            cup_points
##    <chr>                             <dbl>
##  1 Brazil                             82.4
##  2 Burundi                            81.8
##  3 China                              82.9
##  4 Colombia                           83.1
##  5 Costa Rica                         82.8
##  6 Cote d?Ivoire                      79.3
##  7 Ecuador                            83.8
##  8 El Salvador                        83.1
##  9 Ethiopia                           85.5
## 10 Guatemala                          81.8
## 11 Haiti                              77.2
## 12 Honduras                           80.9
## 13 India                              76.8
## 14 Indonesia                          82.6
## 15 Japan                              84.7
## 16 Kenya                              84.3
## 17 Laos                               81.8
## 18 Malawi                             81.7
## 19 Mauritius                          80.5
## 20 Mexico                             80.9
## 21 Myanmar                            80.8
## 22 Nicaragua                          80.5
## 23 Panama                             83.7
## 24 Papua New Guinea                   85.8
## 25 Peru                               82.5
## 26 Philippines                        80.8
## 27 Rwanda                             82.8
## 28 Taiwan                             82.0
## 29 Tanzania, United Republic Of       82.4
## 30 Thailand                           82.6
## 31 Uganda                             84.1
## 32 United States                      86.0
## 33 United States (Hawaii)             81.8
## 34 United States (Puerto Rico)        81.7
## 35 Vietnam                            82.3
## 36 Zambia                             81.9
## 37 <NA>                               79.1
```

```
coffees_reduced <- na.omit(coffees_reduced)


coffee_points <- coffees_reduced %>%
  ggplot(aes(Country.of.Origin, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("Country of Origin compared to Average Cup Points")+
  theme_gray()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
coffee_points
```

```
#Coffee beans produced in Etiopia, Ecuador, Japan, Panama, Papua New Guinea, United Statese, Kenya, Panama, a
nd Uganda have the highest ratings. I did not know that Japan produces coffee, but according to Taylor (202
0), the japanese coffee industry is one of the best in the world.

#Check if Country.Partner is related to high ratings. Meta D agricultural development plc is the most appreci
ated partner and they have received high ratings.
coffees_partners <- arab_coffee %>%
  select(In.Country.Partner, Total.Cup.Points) %>%
  group_by(In.Country.Partner) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  print(n=37) %>%
  arrange(cup_points)
```

```
## # A tibble: 27 × 2
##    In.Country.Partner                                                    cup_p…¹
##    <chr>                                                                   <dbl>
##  1 "Africa Fine Coffee Association"                                         82.3
##  2 "Almacafé"                                                               83.3
##  3 "AMECAFE"                                                                80.8
##  4 "Asociación de Cafés Especiales de Nicaragua"                            80.2
##  5 "Asociación Mexicana De Cafés y Cafeterías De Especialidad A.C."         82.7
##  6 "Asociacion Nacional Del Café"                                           81.7
##  7 "Blossom Valley International"                                           82.2
##  8 "Blossom Valley International\n"                                         80.4
##  9 "Brazil Specialty Coffee Association"                                    81.8
## 10 "Central De Organizaciones Productoras De Café y Cacao Del Perú - Ce…    83.2
## 11 "Centro Agroecológico del Café A.C."                                     82.3
## 12 "Coffee Quality Institute"                                               80.8
## 13 "Ethiopia Commodity Exchange"                                            85.2
## 14 "Instituto Hondureño del Café"                                           80.5
## 15 "Kenya Coffee Traders Association"                                       83.8
## 16 "METAD Agricultural Development plc"                                     86.7
## 17 "NUCOFFEE"                                                               83.4
## 18 "Salvadoran Coffee Council"                                              82.5
## 19 "Specialty Coffee Ass"                                                   84.2
## 20 "Specialty Coffee Association"                                           82.1
## 21 "Specialty Coffee Association of Costa Rica"                             82.5
## 22 "Specialty Coffee Association of Indonesia"                              82.6
## 23 "Specialty Coffee Institute of Asia"                                     84.3
## 24 "Tanzanian Coffee Board"                                                 82.5
## 25 "Torch Coffee Lab Yunnan"                                                82.8
## 26 "Uganda Coffee Development Authority"                                    83.9
## 27 "Yunnan Coffee Exchange"                                                 83.8
## # … with abbreviated variable name ¹cup_points
```

```
partners_points <- coffees_partners %>%
  ggplot(aes(In.Country.Partner, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("In Country Partners compared to Average Cup Points")+
  theme_gray()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
partners_points
```

```
#Coffee beans colors; blue-green and bluish-green beans seem slightly better than green #beans.
coffees_color <- arab_coffee %>%
  select(Color, Total.Cup.Points) %>%
  group_by(Color) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  arrange(cup_points)
coffees_color <- na.omit(coffees_color)

color_points <- coffees_color %>%
  ggplot(aes(Color, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("Coffee Beans Color compared to Average Cup Points")+
  theme_gray()+
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
color_points
```

```
#Harvest_years are grouped, split them.
arab_coffee <- arab_coffee %>%
  mutate(
    harvest_year_num = Harvest.Year %>%
      str_extract("\\d{4}") %>%
      as.numeric()
  )
arab_coffee %>%
  count(Harvest.Year, harvest_year_num, sort = T) %>%
  paged_table()
```

| Harvest.Year <chr> | harvest_year_num <dbl> | n <int> |
|---|---|---|
| 2012 | 2012 | 352 |
| 2014 | 2014 | 226 |
| 2013 | 2013 | 170 |
| 2015 | 2015 | 125 |
| 2016 | 2016 | 122 |
| 2017 | 2017 | 67 |
| NA | NA | 47 |
| 2013/2014 | 2013 | 29 |
| 2015/2016 | 2015 | 28 |

| Harvest.Year | harvest_year_num | n |
|---|---|---|
| <chr> | <dbl> | <int> |
| 2011 | 2011 | 26 |

1-10 of 47 rows Previous 1 2 3 4 5 Next

```
#Processing method and coffee cup points; the "pulped natural/honey" method has high ratings.
coffees_method <- arab_coffee %>%
  select(Processing.Method, Total.Cup.Points) %>%
  group_by(Processing.Method) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  arrange(cup_points)
coffees_method
```

| Processing.Method | cup_points |
|---|---|
| <chr> | <dbl> |
| Other | 81.27846 |
| Washed / Wet | 81.96462 |
| Natural / Dry | 82.35414 |
| Semi-washed / Semi-pulped | 82.63357 |
| Pulped natural / honey | 82.80786 |
| NA | 82.96550 |

6 rows

```
coffees_method <- na.omit(coffees_method) #remove NAs

method_points <- coffees_method %>%
  ggplot(aes(Processing.Method, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("Processing Method compared to Average Cup Points")+
  theme_gray()+
theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
method_points
```

```
#Comparison between number of bags produced and quality or ratings. There is not a clear #relationship betwee
n number of bags produced and ratings, since the average amount produced #is below 300 bags (with a few excep
tions of around 600 and more than 900 bags).
coffees_bags <- arab_coffee %>%
  select(Number.of.Bags, Total.Cup.Points) %>%
  group_by(Number.of.Bags) %>%
  summarise(cup_points = mean(Total.Cup.Points)) %>%
  arrange(cup_points)
coffees_bags
```

| Number.of.Bags | cup_points |
|---|---|
| <dbl> | <dbl> |
| 550 | 72.29000 |
| 85 | 72.33000 |

| Number.of.Bags <dbl> | cup_points <dbl> |
|---|---|
| 22 | 76.92000 |
| 9 | 77.25000 |
| 202 | 78.00000 |
| 440 | 78.66667 |
| 43 | 79.47333 |
| 280 | 79.49200 |
| 127 | 79.50000 |
| 223 | 79.58000 |

1-10 of 130 rows Previous **1** 2 3 4 5 6 ... 13 Next

```
bags_points <- coffees_bags %>%
  ggplot(aes(Number.of.Bags, cup_points)) +
  geom_point(aes(color=cup_points), size = 4) +
  ggtitle("Number of Bags produced compared to Average Cup Points")+
  theme_gray()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
bags_points
```

*Models*

I set seed to take a random sample from the elements that determine total points; I use the previous correlation plot as a reference.

```
set.seed(74)
sample(c("Aroma", "Flavor", "Aftertaste", "Acidity", "Body", "Balance","Cupper_Points"),
       size = 1)
```

```
## [1] "Flavor"
```

```
#Load tidymodels and split the dataset using train/test functions. We do this to avoid over #optimistic predi
ctions (overfitting); this problem might occur after using the same dataset to make predictions. We know that
a model is accurate when error rate is low (Theobald, 2017 p. 48).


arab_coffee <- arab_coffee %>% #to give missing values factor level
  mutate(
    Variety = fct_explicit_na(Variety),
    across(where(is.character), factor))

#Theobald recommmends splittings of 70/30 or 80/2, also considering the size of #the dataset; so I choose 75/
25 (2017, p. 46).

set.seed(42)
coffee_split <- initial_split(arab_coffee, prop = 3/4, strata = Flavor)
coffee_train <- training(coffee_split)
coffee_test <- testing(coffee_split)

#In Tidymodels the vfold_cv splits data into V equal groups
coffee_resamples <- vfold_cv(coffee_train, v = 5, strata = Flavor)


#Recipe. In Tidymodels, recipes are used for feature engineering, to prepare data before using them (Silge &
 Kuhn, 2022, chapter 8). I choose the variables as per correlation plot, with flavor among the most important
ones together with aroma and aftertaste.
coffee_rec <-
  recipe(
    Flavor ~
      Country.of.Origin + Processing.Method + Color +
      In.Country.Partner + Variety + Aroma + Aftertaste + Acidity + Body +
      Balance + Uniformity + Clean.Cup + Sweetness + Moisture +
      altitude_mean_meters,
    data = coffee_train
  ) %>%
  step_impute_mode(all_nominal_predictors()) %>%
  step_other(Country.of.Origin, Variety, In.Country.Partner, Processing.Method,
             threshold = 0.05) %>%
  step_impute_mean(altitude_mean_meters) %>%
  step_normalize(all_numeric_predictors()) %>% #normalize variables
  step_ns(altitude_mean_meters, deg_free = 4) %>%
  step_dummy(all_nominal_predictors())
coffee_rec
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor         15
##
## Operations:
##
## Mode imputation for all_nominal_predictors()
## Collapsing factor levels for Country.of.Origin, Variety, In.Country.Partner,...
## Mean imputation for altitude_mean_meters
## Centering and scaling for all_numeric_predictors()
## Natural splines on altitude_mean_meters
## Dummy variables from all_nominal_predictors()
```

```
#(Dunn, 2020)

#Use prep to explore the processing and to apply the preprocessing to the datasets (Silge & Kuhn, 2022, chapt
er 16.4).
coffee_baked <- bake(prep(coffee_rec), new_data = NULL)
coffee_baked %>% paged_table()
```

| Aroma <dbl> | Aftertaste <dbl> | Acidity <dbl> | Body <dbl> | Balance <dbl> | Uniformity <dbl> | Clean.Cup <dbl> | Sweetness <dbl> |
|---|---|---|---|---|---|---|---|
| 0.04187393 | 0.28202992 | 1.2094382 | 1.38921694 | -0.28525746 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.82641027 | 0.28202992 | 0.4107987 | 0.20251557 | 0.43374965 | 0.3328568 | 0.2339644 | 0.1944118 |
| -0.74266241 | -0.20476505 | 0.1232885 | -0.94928283 | 1.61292131 | 0.3328568 | 0.2339644 | 0.1944118 |
| -0.20917770 | 0.05294994 | 0.6663633 | 2.26179148 | -0.05517518 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.32430701 | -0.66292503 | 0.1232885 | 1.94766465 | -0.28525746 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.04187393 | -0.66292503 | -0.6753511 | -0.94928283 | 3.05093554 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.82641027 | 1.22698488 | -0.6753511 | 0.20251557 | 0.89391420 | 0.3328568 | 0.2339644 | -1.2666112 |
| 0.57535864 | -0.20476505 | -0.3878409 | -0.07670829 | 0.89391420 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.57535864 | 0.05294994 | 0.1232885 | 0.20251557 | 0.17490709 | 0.3328568 | 0.2339644 | 0.1944118 |
| 0.32430701 | -0.20476505 | -0.6753511 | -0.07670829 | 0.89391420 | 0.3328568 | 0.2339644 | 0.1944118 |

1-10 of 981 rows | 1-8 of 36 columns Previous **1** 2 3 4 5 6 ... 99 Next

```
#Use DoParallel to register execution of R code
n_cores <- parallel::detectCores(logical = FALSE)
cl <- makePSOCKcluster(n_cores - 1)
registerDoParallel(cl)
```

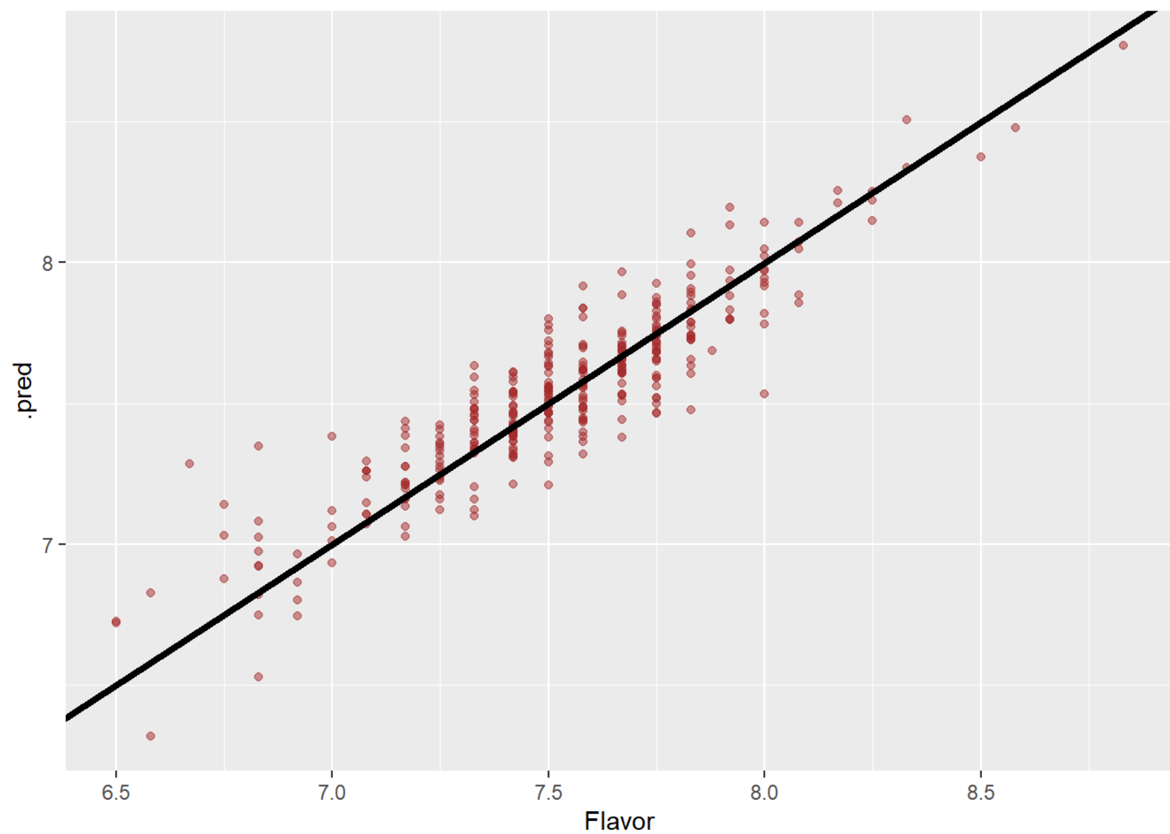
This was not requested, but first I want to explore how this all works. I use function workflow to keep the project organized and bundle it for the next steps.

```
lm_spec <- linear_reg() %>%
  set_engine("lm")

lm_workflow <- workflow() %>%
  add_recipe(coffee_rec) %>%
  add_model(lm_spec)

lm_fit_train <- lm_workflow %>%
  fit(data = coffee_train)

lm_fit <- last_fit(lm_fit_train, coffee_split) #final fit

collect_metrics(lm_fit)
```

| .metric <chr> | .estimator <chr> | .estimate <dbl> | .config <chr> |
|---|---|---|---|
| rmse | standard | 0.1416476 | Preprocessor1_Model1 |
| rsq | standard | 0.8275717 | Preprocessor1_Model1 |

2 rows

```
#Comparison between prediction and actual data
collect_predictions(lm_fit) %>%
  ggplot(aes(x = Flavor, y = .pred)) +
  geom_point(color = "brown", alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0, size = 1.5)
```




```
#Since the result from the Linear regression is good, I go ahead with Random Forest.
```

*Random Forest*

Random forest ensemble learning combines the output of models to create a prediction method (Theobald, 2017, p. 115). Set mode and engine, then workflow as I did in the previous model.

```
ranger_spec <- rand_forest(mtry = tune(), min_n = tune(), trees = 1000) %>%
  set_mode("regression") %>%
  set_engine("ranger", importance = "permutation")

ranger_workflow <- workflow() %>%
  add_recipe(coffee_rec) %>%
  add_model(ranger_spec)

set.seed(12)

#tic toc functions from tictoc package, to nest and timing my function.
tic()
ranger_tune <-
  tune_grid(ranger_workflow, resamples = coffee_resamples, grid = 11)
```

```
## i Creating pre-processing data to finalize unknown parameter: mtry
```

```
toc()
```

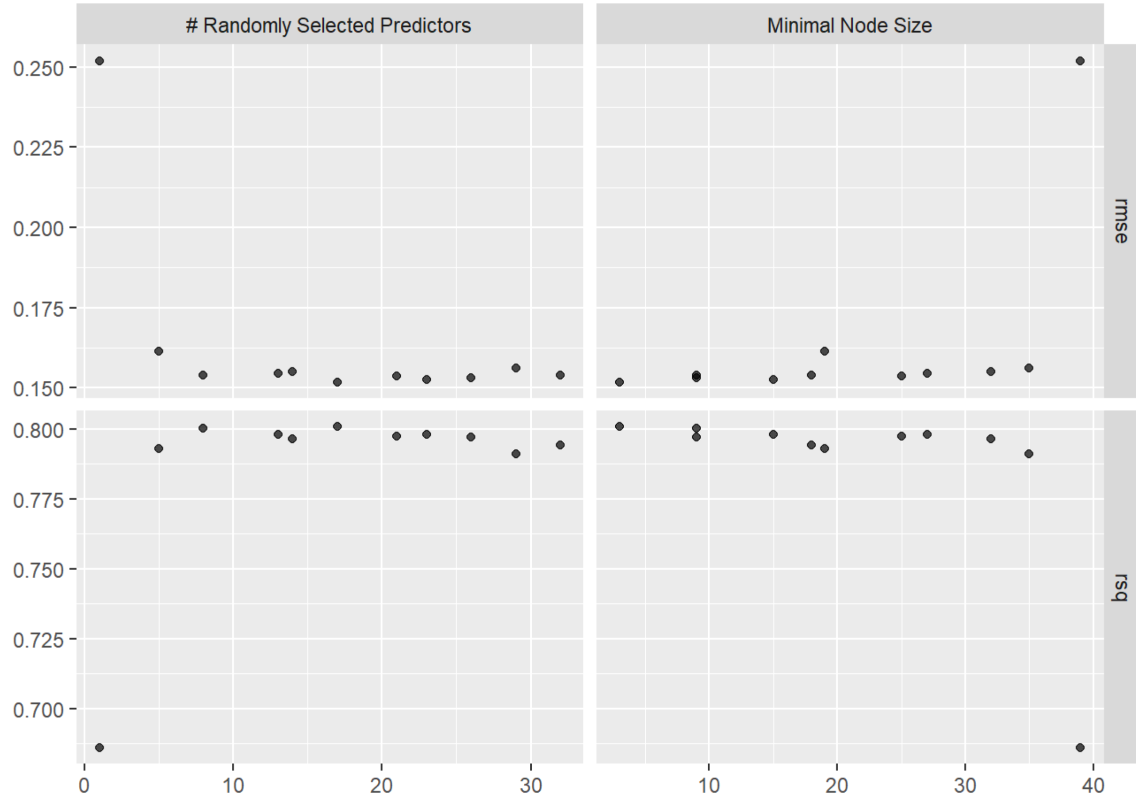
```
## 53.06 sec elapsed
```

```
#Tune function to find optimal paramethers
show_best(ranger_tune, metric = "rmse")
```

| mtry <int> | min_n <int> | .metric <chr> | .estimator <chr> | mean <dbl> | n <int> | std_err <dbl> | .config <chr> |
|---|---|---|---|---|---|---|---|
| 17 | 3 | rmse | standard | 0.1519064 | 5 | 0.005076617 | Preprocessor1_Model02 |
| 23 | 15 | rmse | standard | 0.1526415 | 5 | 0.005480836 | Preprocessor1_Model09 |
| 26 | 9 | rmse | standard | 0.1530627 | 5 | 0.005484704 | Preprocessor1_Model07 |
| 21 | 25 | rmse | standard | 0.1535722 | 5 | 0.005039420 | Preprocessor1_Model08 |
| 8 | 9 | rmse | standard | 0.1538559 | 5 | 0.005519474 | Preprocessor1_Model11 |

5 rows

```
autoplot(ranger_tune)
```

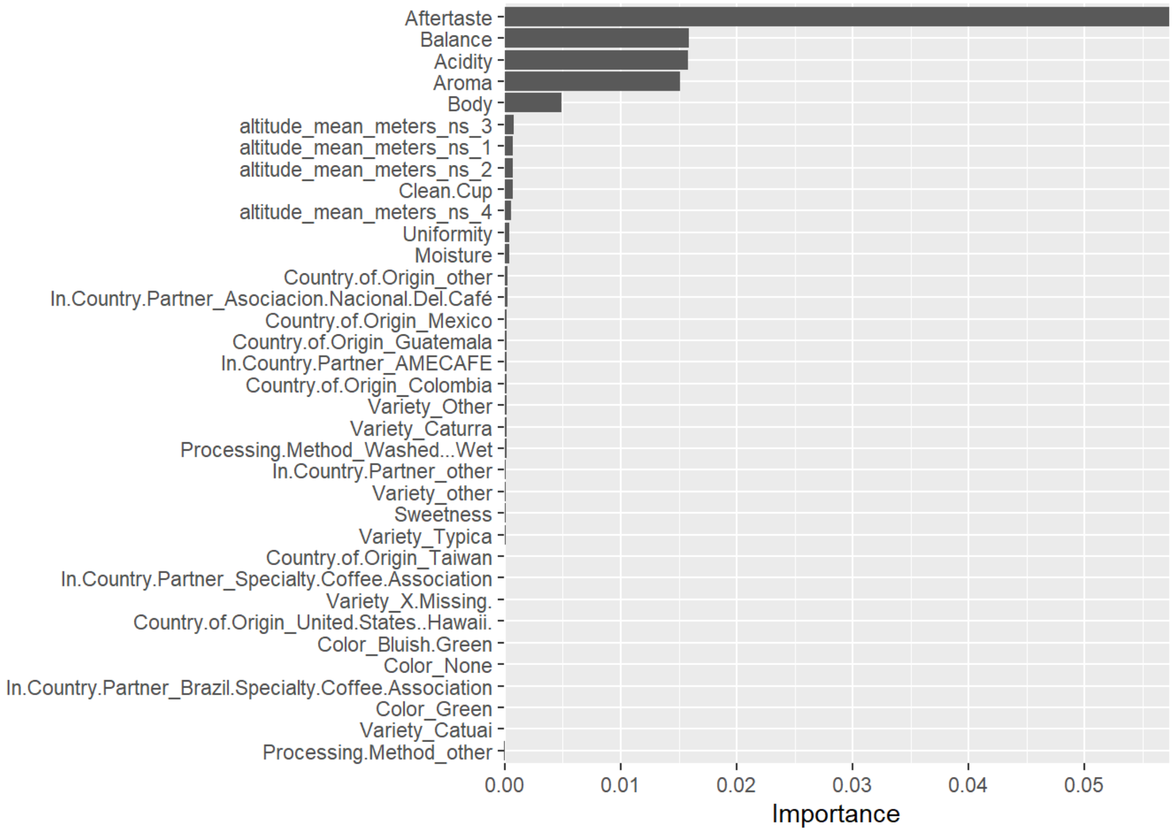



```
#Result are consistent with the previous RMSE and model.
ranger_best <- ranger_workflow %>%
  finalize_workflow(select_best(ranger_tune, metric = "rmse"))

ranger_fit <- last_fit(ranger_best, coffee_split)
collect_metrics(ranger_fit)
```

| .metric <chr> | .estimator <chr> | .estimate <dbl> | .config <chr> |
|---|---|---|---|
| rmse | standard | 0.1375165 | Preprocessor1_Model1 |
| rsq | standard | 0.8363437 | Preprocessor1_Model1 |

2 rows

```
#According to this model, the following graph shows the most important variables.
ranger_fit %>%
  extract_fit_engine() %>%
  vi() %>%
  mutate(Variable = fct_reorder(Variable, Importance)) %>%
  ggplot(aes(x = Importance, y = Variable)) +
  geom_col() +
  scale_x_continuous(expand = c(0, 0)) +
  labs(y = NULL) +
  theme(legend.position = c(0.3, 0.3))
```




*LASSO*

A model trained using L1 norm is called Least Absolute Shrinkage and Selection Operator (LASSO).This model works well when datasets have many features, such in my case.Same procedure as before, specify model, set engine, use Lambda penalty, workflow, tune, tic toc. #Formula is: Lasso regression error = Regression error + L1 norm

```
lasso_spec <- linear_reg(penalty = tune(), mixture = 1) %>%
  set_engine("glmnet")


lasso_lambda_grid <- grid_regular(penalty(), levels = 50) #specify the amount of regularization to use (Silge
& Kuhn, 2022)

lasso_workflow <- workflow() %>%
  add_recipe(coffee_rec) %>%
  add_model(lasso_spec)


tic()
lasso_tune <-
  tune_grid(
    lasso_workflow,
    resamples = coffee_resamples,
    grid = lasso_lambda_grid
  )
toc()
```

```
## 2.17 sec elapsed
```

```
show_best(lasso_tune, metric = "rmse")
```

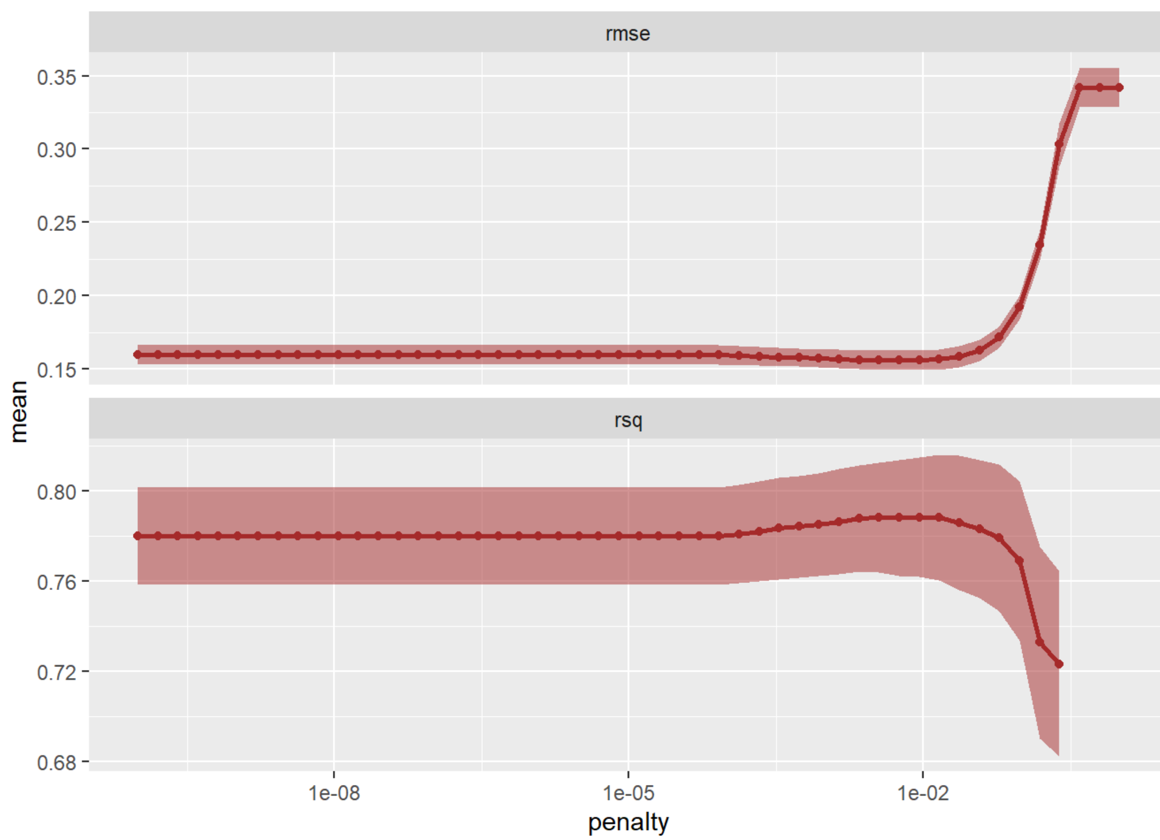
| penalty <dbl> | .metric <chr> | .estimator <chr> | mean <dbl> | n <int> | std_err <dbl> | .config <chr> |
|---|---|---|---|---|---|---|
| 0.003556480 | rmse | standard | 0.1561567 | 5 | 0.006465583 | Preprocessor1_Model38 |
| 0.009102982 | rmse | standard | 0.1561918 | 5 | 0.006884419 | Preprocessor1_Model40 |
| 0.005689866 | rmse | standard | 0.1562062 | 5 | 0.006723808 | Preprocessor1_Model39 |
| 0.002222996 | rmse | standard | 0.1563645 | 5 | 0.006361642 | Preprocessor1_Model37 |
| 0.014563485 | rmse | standard | 0.1566450 | 5 | 0.006985212 | Preprocessor1_Model41 |

5 rows

```
#RMSE and RSQ using LASSO
collect_metrics(lasso_tune) %>%
  #filter(!is.na(std_err)) %>%
  ggplot(aes(x = penalty, y = mean)) +
  geom_line(size = 1, color = "brown") +
  geom_point(color = "brown") +
  geom_ribbon(aes(ymin = mean - std_err, ymax = mean + std_err),
              alpha = 0.5, fill = "brown") +
  facet_wrap(~.metric, ncol = 1, scales = "free_y") +
  scale_x_log10()
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

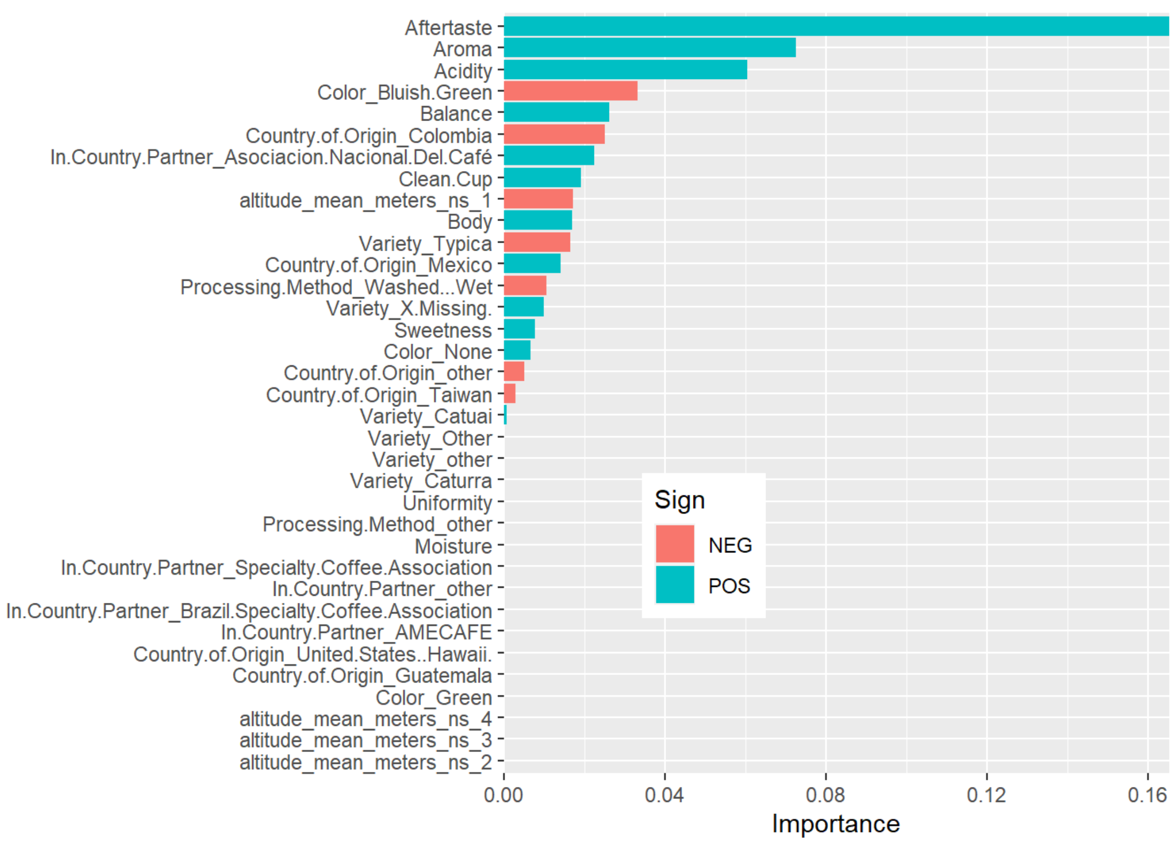


```
#Results
lasso_best_workflow <- lasso_workflow %>%
  finalize_workflow(select_best(lasso_tune, metric = "rmse"))
lasso_fit <- last_fit(lasso_best_workflow, coffee_split)

collect_metrics(lasso_fit)
```

| .metric <chr> | .estimator <chr> | .estimate <dbl> | .config <chr> |
|---|---|---|---|
| rmse | standard | 0.1402011 | Preprocessor1_Model1 |
| rsq | standard | 0.8298351 | Preprocessor1_Model1 |

2 rows

```
#Check again which variables are the most important according to LASSO Model. The first #results are similar
 to the ones I have obtained using Random Forest, but this model adds
#a negative correlation with Colombia as a Country of Origin, and poisitive correlations #with the partner ca
lled ""In Country Partner Asociacion Del Café", with clean cup, and #Mexico as a country of origin.

lasso_fit %>%
  extract_fit_engine() %>%
  vi(
    lambda =  select_best(lasso_tune, metric = "rmse")$penalty
  ) %>%
  mutate(Variable = fct_reorder(Variable, Importance)) %>%
  ggplot(aes(x = Importance, y = Variable, fill = Sign)) +
  geom_col() +
  scale_x_continuous(expand = c(0, 0)) +
  labs(y = NULL) +
  theme(legend.position = c(0.3, 0.3))
```



*Results*

The coffee dataset has many variables, some are numeric and others are characters. Some of the variables describing coffee characteristics are related toe each other, especially Aroma, Aftertaste, Acidity, Body, and Balance. The results from LASSO (0.140) and Random Forest (0.138) show low RMSE and Random Forest performs better that the second. They also provide more insights andI can relate the best organoleptic characteristics with the country of origin, the best processing method, the worst variables, or those that are not relevant.

*Conclusion*

*Summary* Coffee raters use common standards to assess the quality of coffee cups. Flavor is the most relevant standard for the final evaluation of quality. Other elements such as “Aroma”, “Aftertaste”, “Body”, and “Balance”, are better with high values, while “Acidity” should not prevail over others. Most coffee beans are produced at similar heights (below 2000 meters), and I assume it is the optimal environment. Variety of coffee beans matters a lot, and Bourbon, Caturra and Typica stand out. Etiopia, Ecuador, Japan, Panama, Papua New Guinea, United Statese, Kenya, Panama, and Uganda produce beans of highest quality. Lasso results identify a positive relationship with coffee produced in Taiwan. Meta D Agricultural Development plc seems a good partner in terms of high ratings, but the Lasso model identifies Asociacion Nacional del café as a good choice. According to Lasso, Coffee beans color does not make a huge difference. However, the processing method impacts model results (this output is different from EDA); Lasso model analysis is in favor of the washed-wet method.

*Potential impact, limitations* This analysis provides a lot of practical information for identifying how the best coffee quality is determined and evaluated. The same approach can be applied to other types of food such as chocolate or wheat (to say just a a few) and it can support buyers or companies in making unbiased decisions. However, since the coffee dataset provides a lot of variables, the modelling approach can make a difference. For example, I had initially thought of a K-means model, but I have changed my mind since some variables are too similar to each other. The regression analysis, even if not requested for this capstone, has been helpful to determine an initial RMSE and whether the variables used for the prediction made sense. The dataset has limitations because it does not explain the brewing methods used, and these are relevant because they change depending on geographical areas. For example, brewing espresso coffee is very different from filtered coffee.

*References* Coffeereview. (n.d.). Interpreting Coffee Reviews | CoffeeReview.com. Coffee Review. https://www.coffeereview.com/interpret-coffee/ (https://www.coffeereview.com/interpret-coffee/)

Dunn, T. (2020). tdunn: TidyTuesday 2020 Week 28. Tdunn.ca. https://tdunn.ca/posts/2020-07-12-tidytuesday-2020-week-28/ (https://tdunn.ca/posts/2020-07-12-tidytuesday-2020-week-28/)

LeDoux, J. (2022, January 28). coffee-quality-database. GitHub. https://github.com/jldbc/coffee-quality-database (https://github.com/jldbc/coffee-quality-database)

Mock, T. (2020, July 6). rfordatascience/tidytuesday. GitHub. https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-07-07/readme.md (https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-07-07/readme.md)

Silge, J., & Kuhn, M. (2022). Tidy Modeling with R. In www.tmwr.org. https://www.tmwr.org/SpecialCoffee (https://www.tmwr.org/SpecialCoffee). (2018, April 16).

The blending is an art - blog SpecialCoffee. SpecialCoffee. https://specialcoffeeitaly.com/blending-coffee-art/ (https://specialcoffeeitaly.com/blending-coffee-art/)

Taylor, H. (2020, May 26). The Secrets of Japanese Coffee Culture. VOYAPON. https://voyapon.com/secrets-of-japanese-coffee-culture/# (https://voyapon.com/secrets-of-japanese-coffee-culture/#):~:text=The%20service%20industry%20in%20Japan

Theobald, O. (2017). Machine learning for absolute beginners : a plain English introduction (pp. 1–162).

The Author.Wikipedia Contributors. (2019, November 7). List of highest mountains on Earth. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/List_of_highest_mountains_on_earth (https://en.wikipedia.org/wiki/List_of_highest_mountains_on_earth)