Project

SeanJ- Volcaetus

4/3/2022

Data

From TidyTuesday URL:https://github.com/rfordatascience/tidytuesday/tree/master/data/2020/2020-07-07

```
coffee_ratings <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   total_cup_points = col_double(),
##   number_of_bags = col_double(),
##   aroma = col_double(),
##   flavor = col_double(),
##   aftertaste = col_double(),
##   acidity = col_double(),
##   body = col_double(),
##   balance = col_double(),
##   uniformity = col_double(),
##   clean_cup = col_double(),
##   sweetness = col_double(),
##   cupper_points = col_double(),
##   moisture = col_double(),
##   category_one_defects = col_double(),
##   quakers = col_double(),
##   category_two_defects = col_double(),
##   altitude_low_meters = col_double(),
##   altitude_high_meters = col_double(),
##   altitude_mean_meters = col_double()
## )
## i Use `spec()` for the full column specifications.
```

Quick overview

```
summary(coffee_ratings)
```

```
##  total_cup_points   species             owner           country_of_origin
##  Min.   : 0.00    Length:1339        Length:1339        Length:1339
##  1st Qu.:81.08    Class :character   Class :character   Class :character
##  Median :82.50    Mode  :character   Mode  :character   Mode  :character
##  Mean   :82.09
##  3rd Qu.:83.67
```

```
##  Max.   :90.58
##
##   farm_name          lot_number            mill            ico_number
##  Length:1339        Length:1339        Length:1339        Length:1339
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    company            altitude           region            producer
##  Length:1339        Length:1339        Length:1339        Length:1339
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  number_of_bags    bag_weight        in_country_partner harvest_year
##  Min.   :   0.0   Length:1339        Length:1339        Length:1339
##  1st Qu.:  14.0   Class :character   Class :character   Class :character
##  Median : 175.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 154.2
##  3rd Qu.: 275.0
##  Max.   :1062.0
##
##  grading_date         owner_1            variety          processing_method
##  Length:1339        Length:1339        Length:1339        Length:1339
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      aroma            flavor       aftertaste       acidity            body
##  Min.   :0.000   Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:7.420   1st Qu.:7.33   1st Qu.:7.250   1st Qu.:7.330   1st Qu.:7.330
##  Median :7.580   Median :7.58   Median :7.420   Median :7.580   Median :7.500
##  Mean   :7.567   Mean   :7.52   Mean   :7.401   Mean   :7.536   Mean   :7.517
##  3rd Qu.:7.750   3rd Qu.:7.75   3rd Qu.:7.580   3rd Qu.:7.750   3rd Qu.:7.670
##  Max.   :8.750   Max.   :8.83   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   : 9.835   Mean   : 9.835   Mean   : 9.857
##  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       moisture       category_one_defects    quakers
##  Min.   : 0.000   Min.   :0.00000   Min.   : 0.0000      Min.   : 0.0000
##  1st Qu.: 7.250   1st Qu.:0.09000   1st Qu.: 0.0000      1st Qu.: 0.0000
##  Median : 7.500   Median :0.11000   Median : 0.0000      Median : 0.0000
```

```
##  Mean   : 7.503   Mean   :0.08838   Mean   : 0.4795      Mean   : 0.1734
##  3rd Qu.: 7.750   3rd Qu.:0.12000   3rd Qu.: 0.0000      3rd Qu.: 0.0000
##  Max.   :10.000   Max.   :0.28000   Max.   :63.0000      Max.   :11.0000
##                                                          NA's   :1
##     color           category_two_defects  expiration        certification_body
##  Length:1339        Min.   : 0.000       Length:1339        Length:1339
##  Class :character   1st Qu.: 0.000       Class :character   Class :character
##  Mode  :character   Median : 2.000       Mode  :character   Mode  :character
##                     Mean   : 3.556
##                     3rd Qu.: 4.000
##                     Max.   :55.000
##
##  certification_address certification_contact unit_of_measurement
##  Length:1339           Length:1339           Length:1339
##  Class :character      Class :character      Class :character
##  Mode  :character      Mode  :character      Mode  :character
##
##
##
##
##  altitude_low_meters altitude_high_meters altitude_mean_meters
##  Min.   :     1      Min.   :     1       Min.   :     1
##  1st Qu.:  1100      1st Qu.:  1100       1st Qu.:  1100
##  Median :  1311      Median :  1350       Median :  1311
##  Mean   :  1751      Mean   :  1799       Mean   :  1775
##  3rd Qu.:  1600      3rd Qu.:  1650       3rd Qu.:  1600
##  Max.   :190164      Max.   :190164       Max.   :190164
##  NA's   :230         NA's   :230          NA's   :230
```

A few NA's.

1 within quakers, and 230 in Altitude low/high/mean

Check what is happening in the rest of the data set

Count of NA's per coloumn

```
apply(X=is.na(coffee_ratings), MARGIN = 2, FUN = sum)
```

```
##      total_cup_points               species                 owner
##                     0                     0                     7
##     country_of_origin             farm_name           lot_number
##                     1                   359                  1063
##                  mill             ico_number               company
##                   315                   151                   209
##              altitude                region              producer
##                   226                    59                   231
##        number_of_bags           bag_weight    in_country_partner
##                     0                     0                     0
##          harvest_year          grading_date               owner_1
##                    47                     0                     7
##               variety     processing_method                 aroma
##                   226                   170                     0
##                flavor            aftertaste               acidity
##                     0                     0                     0
```

```
##                  body               balance            uniformity
##                     0                     0                     0
##             clean_cup             sweetness         cupper_points
##                     0                     0                     0
##              moisture  category_one_defects               quakers
##                     0                     0                     1
##                 color  category_two_defects            expiration
##                   218                     0                     0
##    certification_body certification_address certification_contact
##                     0                     0                     0
##   unit_of_measurement   altitude_low_meters  altitude_high_meters
##                     0                   230                   230
##  altitude_mean_meters
##                   230
```

I will be just removing some of the columns with many missing vlaues, for instance farm_name.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Removal of columns

```
coffee = coffee_ratings%>%
  select(-farm_name,-lot_number,-mill,-ico_number,-altitude,
         -altitude_low_meters,-altitude_high_meters,-producer,-company,
         -expiration,-certification_address,-owner_1,-grading_date,
         -certification_contact,-unit_of_measurement)
apply(X=is.na(coffee), MARGIN = 2, FUN = sum)
```

```
##     total_cup_points              species                owner
##                    0                    0                    7
##    country_of_origin               region       number_of_bags
##                    1                   59                    0
##           bag_weight   in_country_partner         harvest_year
##                    0                    0                   47
##              variety    processing_method               aroma
##                  226                  170                    0
##               flavor           aftertaste              acidity
##                    0                    0                    0
##                 body              balance           uniformity
##                    0                    0                    0
##            clean_cup            sweetness         cupper_points
##                    0                    0                    0
##             moisture  category_one_defects               quakers
##                    0                    0                    1
##                color  category_two_defects    certification_body
##                  218                    0                    0
```

```
## altitude_mean_meters
##                  230
```

```
#view(coffee)
```

```
coffee = na.omit(coffee)
#view(coffee)
```

```
coffee[grep("lbs",coffee$bag_weight),]
```

```
## # A tibble: 18 x 28
##    total_cup_points species owner     country_of_origin  region   number_of_bags
##               <dbl> <chr>   <chr>     <chr>              <chr>             <dbl>
##  1             87.2 Arabica the coff~ Costa Rica         san ram~            250
##  2             86.3 Arabica francisc~ Costa Rica         west an~            250
##  3             85.3 Arabica the coff~ Costa Rica         west va~            250
##  4             85.3 Arabica the coff~ Costa Rica         san ram~            250
##  5             84.7 Arabica fabian c~ Costa Rica         tarrazu              50
##  6             84.5 Arabica fabian c~ Costa Rica         tarrazu             250
##  7             83.8 Arabica german n~ United States (Pu~ yauco r~             18
##  8             83.8 Arabica the coff~ Guatemala          quetzal~            250
##  9             83.3 Arabica the coff~ Costa Rica         san ram~            250
## 10             83.3 Arabica itiah co~ Haiti              thiotte~              2
## 11             83   Arabica german n~ United States (Pu~ yauco r~             17
## 12             81.5 Arabica myriam k~ Haiti              dondon,~            300
## 13             81.2 Arabica essencec~ Guatemala          huehuet~             36
## 14             81.1 Arabica german n~ United States (Pu~ yauco r~             18
## 15             80.9 Arabica chris fi~ Nicaragua          matagal~            275
## 16             80.8 Arabica the coff~ Costa Rica         san ram~            250
## 17             79.3 Arabica the coff~ Colombia           pereira             250
## 18             79.1 Arabica german n~ United States (Pu~ yauco r~             18
## # ... with 22 more variables: bag_weight <chr>, in_country_partner <chr>,
## #   harvest_year <chr>, variety <chr>, processing_method <chr>, aroma <dbl>,
## #   flavor <dbl>, aftertaste <dbl>, acidity <dbl>, body <dbl>, balance <dbl>,
## #   uniformity <dbl>, clean_cup <dbl>, sweetness <dbl>, cupper_points <dbl>,
## #   moisture <dbl>, category_one_defects <dbl>, quakers <dbl>, color <chr>,
## #   category_two_defects <dbl>, certification_body <chr>,
## #   altitude_mean_meters <dbl>
```

```
coffee = separate(data = coffee, col = bag_weight, into = c("weight", "type"), sep = " ")
```

```
coffee$weight = as.numeric(coffee$weight)
```

```
for(i in 1:length(coffee)){
  if(coffee[i,8]=="kg"){
  coffee[i,7] = round(coffee[i,7] * 2.20462,0)
  coffee[i,8] = "lbs"
  }
}
```

```
coffee = coffee%>%
  select(-type)
```

```
coffee = coffee%>%
    rename(avg_altitude=altitude_mean_meters)
```

```
coffee$avg_altitude = round(coffee$avg_altitude * 3.28084,0)
```

```
coffee$harvest_year = substr(coffee$harvest_year,1,4)
coffee$harvest_year = as.numeric(coffee$harvest_year)
```

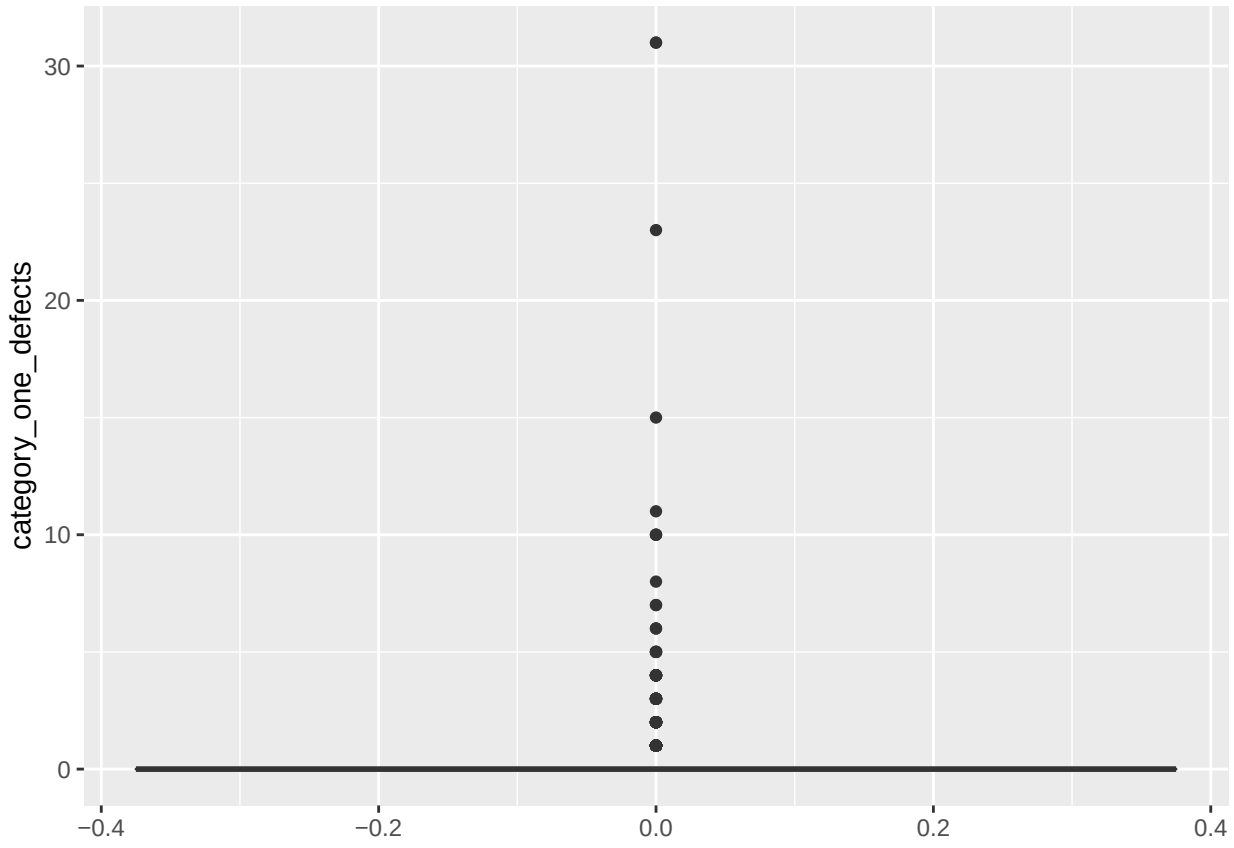
```
summary(coffee[,c(9,12:24,26,28)])
```

```
##   harvest_year      aroma           flavor        aftertaste       acidity
##  Min.   :2011   Min.   :5.080   Min.   :6.170   Min.   :6.170   Min.   :5.250
##  1st Qu.:2012   1st Qu.:7.420   1st Qu.:7.330   1st Qu.:7.170   1st Qu.:7.330
##  Median :2014   Median :7.580   Median :7.500   Median :7.420   Median :7.500
##  Mean   :2014   Mean   :7.559   Mean   :7.504   Mean   :7.374   Mean   :7.515
##  3rd Qu.:2015   3rd Qu.:7.750   3rd Qu.:7.670   3rd Qu.:7.580   3rd Qu.:7.670
##  Max.   :2018   Max.   :8.750   Max.   :8.670   Max.   :8.500   Max.   :8.580
##       body          balance        uniformity       clean_cup
##  Min.   :6.330   Min.   :6.080   Min.   : 6.000   Min.   : 0.000
##  1st Qu.:7.330   1st Qu.:7.330   1st Qu.:10.000   1st Qu.:10.000
##  Median :7.500   Median :7.500   Median :10.000   Median :10.000
##  Mean   :7.494   Mean   :7.488   Mean   : 9.871   Mean   : 9.849
##  3rd Qu.:7.670   3rd Qu.:7.670   3rd Qu.:10.000   3rd Qu.:10.000
##  Max.   :8.420   Max.   :8.580   Max.   :10.000   Max.   :10.000
##    sweetness     cupper_points      moisture       category_one_defects
##  Min.   : 1.33   Min.   :5.170   Min.   :0.00000   Min.   : 0.0000
##  1st Qu.:10.00   1st Qu.:7.250   1st Qu.:0.10000   1st Qu.: 0.0000
##  Median :10.00   Median :7.500   Median :0.11000   Median : 0.0000
##  Mean   : 9.93   Mean   :7.459   Mean   :0.09737   Mean   : 0.4262
##  3rd Qu.:10.00   3rd Qu.:7.670   3rd Qu.:0.12000   3rd Qu.: 0.0000
##  Max.   :10.00   Max.   :8.580   Max.   :0.17000   Max.   :31.0000
##     quakers        category_two_defects  avg_altitude
##  Min.   : 0.0000   Min.   : 0.000       Min.   :     3
##  1st Qu.: 0.0000   1st Qu.: 0.000       1st Qu.:  3609
##  Median : 0.0000   Median : 2.000       Median :  4300
##  Mean   : 0.1521   Mean   : 3.822       Mean   :  6145
##  3rd Qu.: 0.0000   3rd Qu.: 5.000       3rd Qu.:  5249
##  Max.   :11.0000   Max.   :47.000       Max.   :623898
```

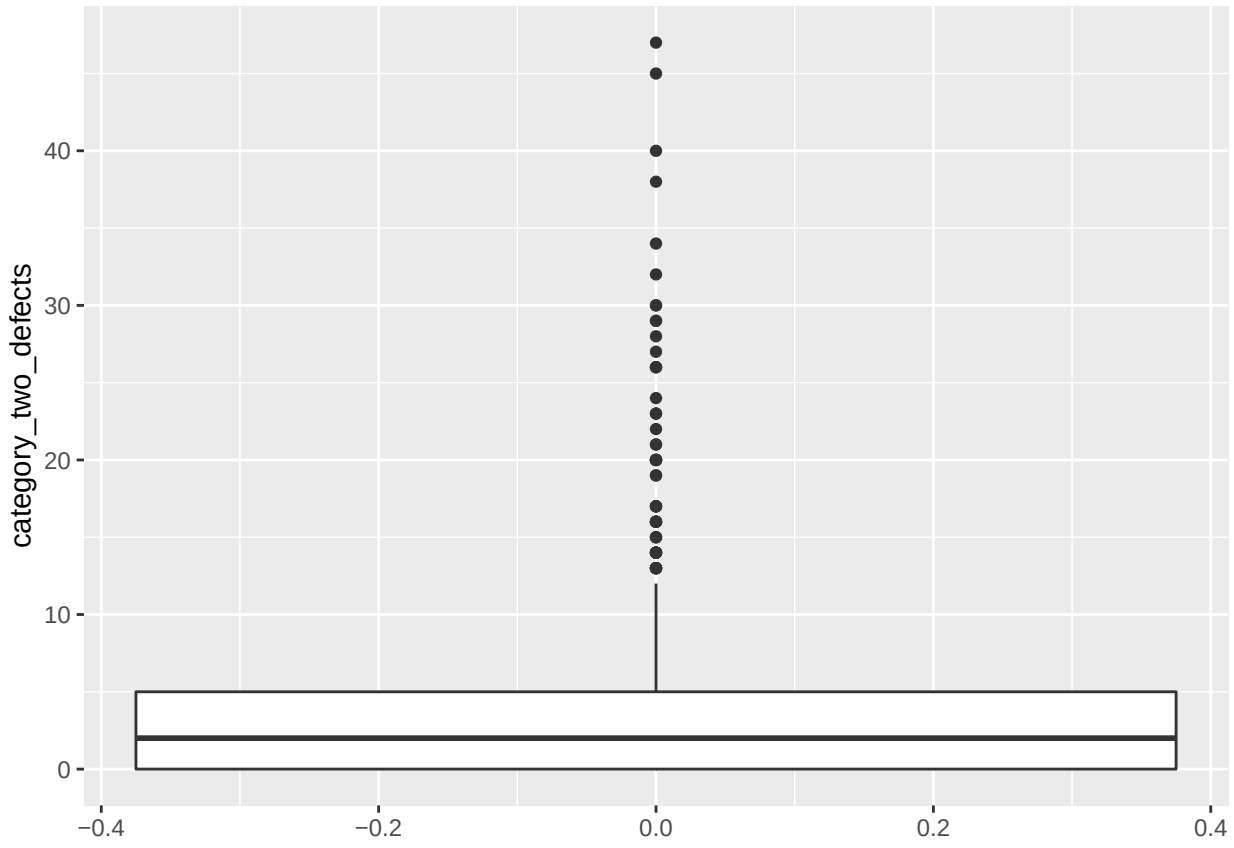
```
library(ggplot2)
```

check for outliers in some of the fields

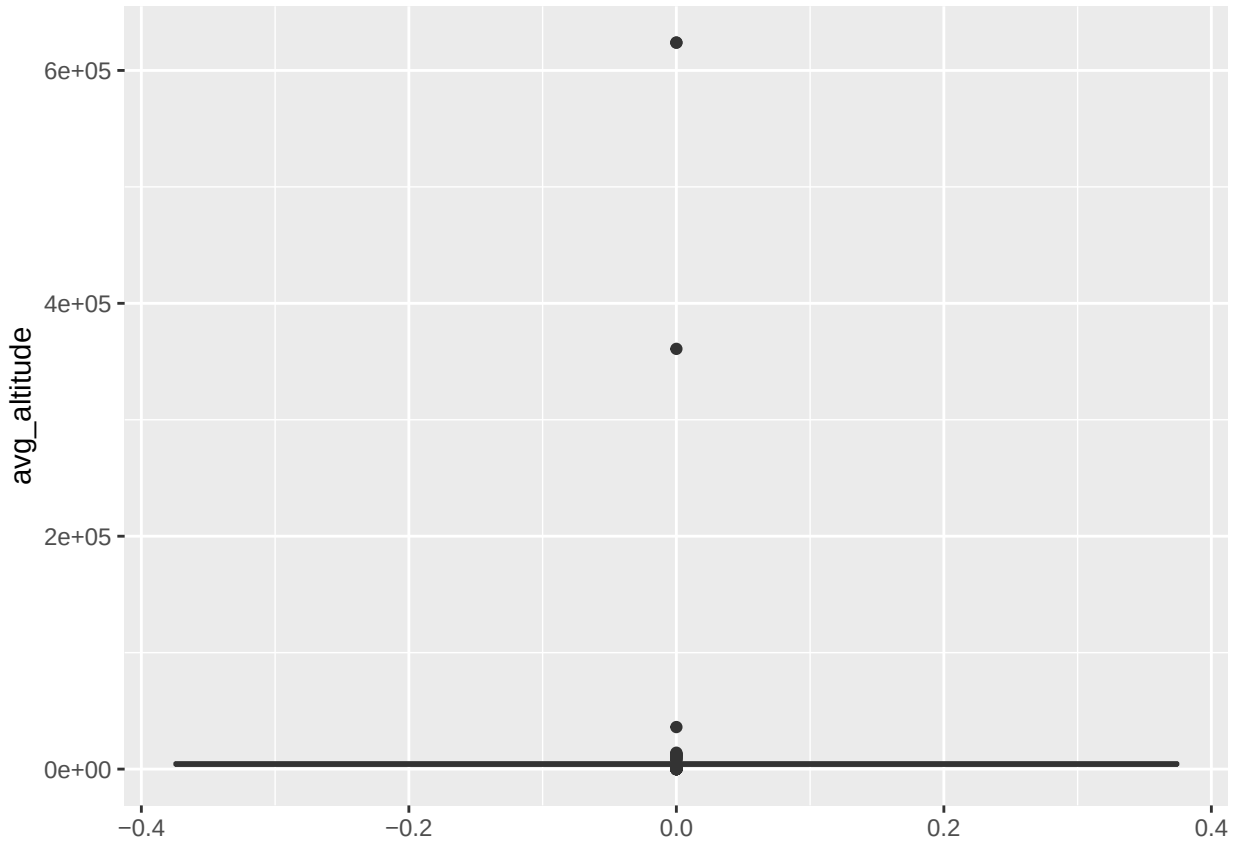
```
defect1_plt = ggplot(coffee, aes(y=category_one_defects)) +
              geom_boxplot()
defect2_plt = ggplot(coffee, aes(y=category_two_defects)) +
              geom_boxplot()
alt_plt = ggplot(coffee, aes(y=avg_altitude)) +
              geom_boxplot()
defect1_plt
```




```
defect2_plt
```




```
alt_plt
```



There are some outliers, but not that many that would result in a concern at this time.

Pick out some of the information that is not necessary at this point in exploration

```
c = coffee[,c(1:2,4,10:26,28)]
```

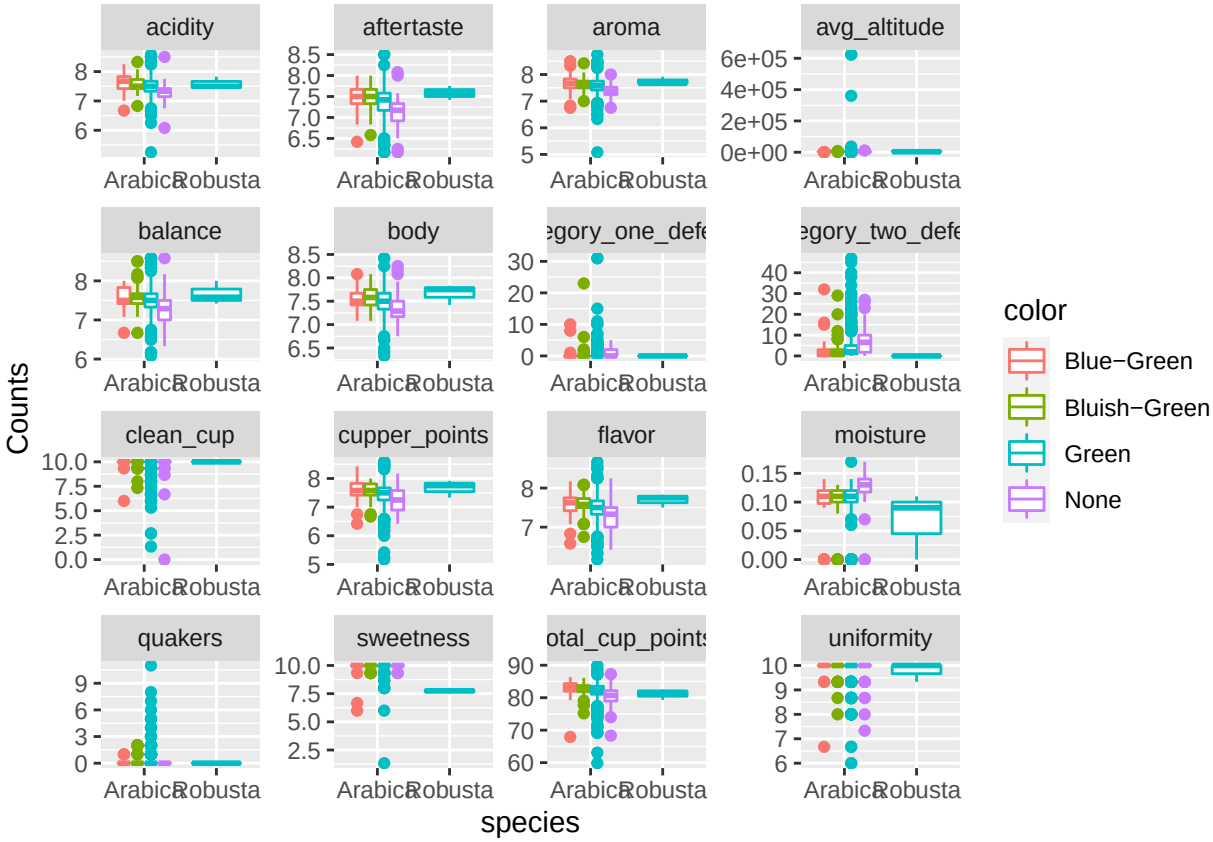
Condense the data

```
c.v1 = c%>%pivot_longer(
  cols = !c(species, country_of_origin,variety,processing_method,color),
  names_to = "Variables",
  values_to = "Counts")
c.v1
```

```
## # A tibble: 14,304 x 7
##    species country_of_origin variety processing_method color Variables    Counts
##    <chr>   <chr>             <chr>   <chr>             <chr> <chr>         <dbl>
##  1 Arabica Ethiopia          Other   Washed / Wet      Green total_cup_p~  89.9
##  2 Arabica Ethiopia          Other   Washed / Wet      Green aroma          8.75
##  3 Arabica Ethiopia          Other   Washed / Wet      Green flavor         8.67
##  4 Arabica Ethiopia          Other   Washed / Wet      Green aftertaste     8.5
##  5 Arabica Ethiopia          Other   Washed / Wet      Green acidity        8.58
##  6 Arabica Ethiopia          Other   Washed / Wet      Green body           8.42
##  7 Arabica Ethiopia          Other   Washed / Wet      Green balance        8.42
##  8 Arabica Ethiopia          Other   Washed / Wet      Green uniformity    10
##  9 Arabica Ethiopia          Other   Washed / Wet      Green clean_cup     10
## 10 Arabica Ethiopia          Other   Washed / Wet      Green sweetness     10
## # ... with 14,294 more rows
```

Plot the data to see overall behavior

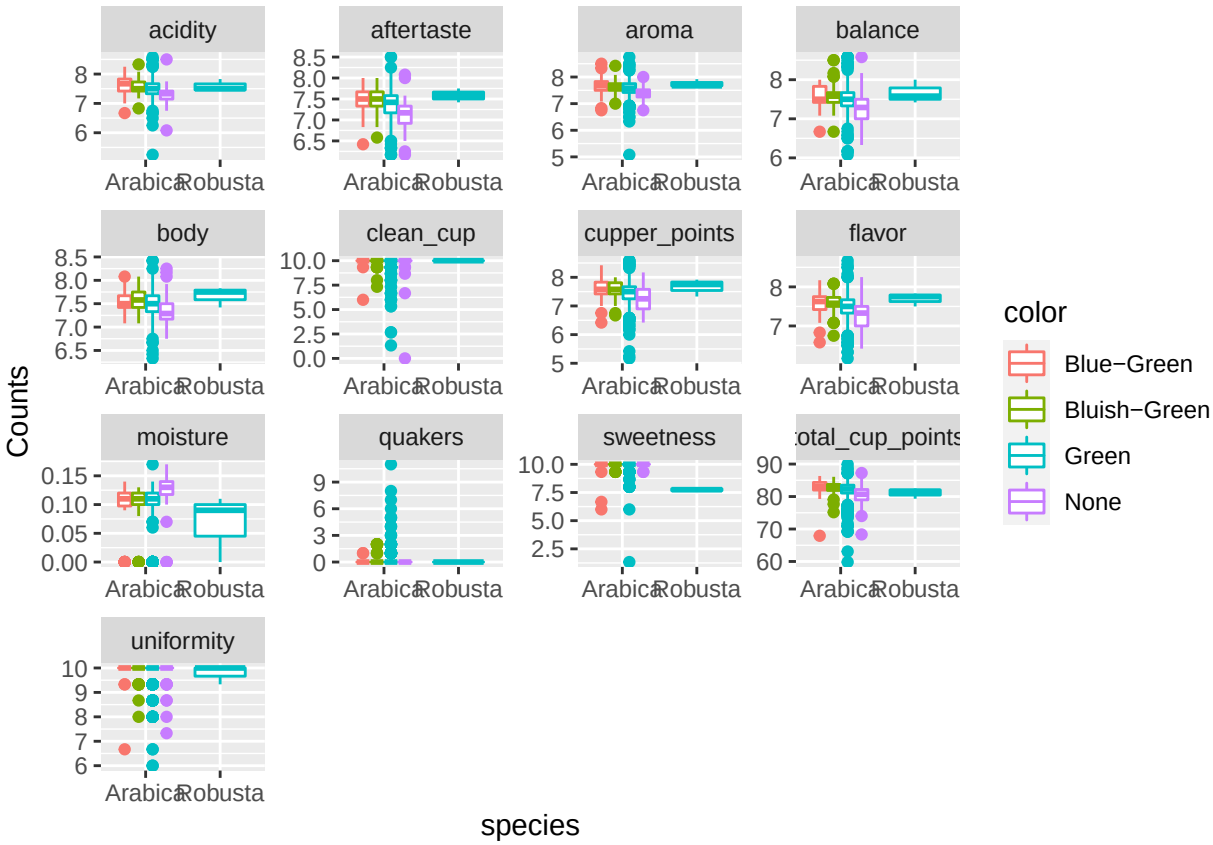
```
ggplot(c.v1,aes(x=species,y=Counts,color=color))+geom_boxplot()+facet_wrap(~Variables,scales = "free")
```



Filter out the items that have known outliers

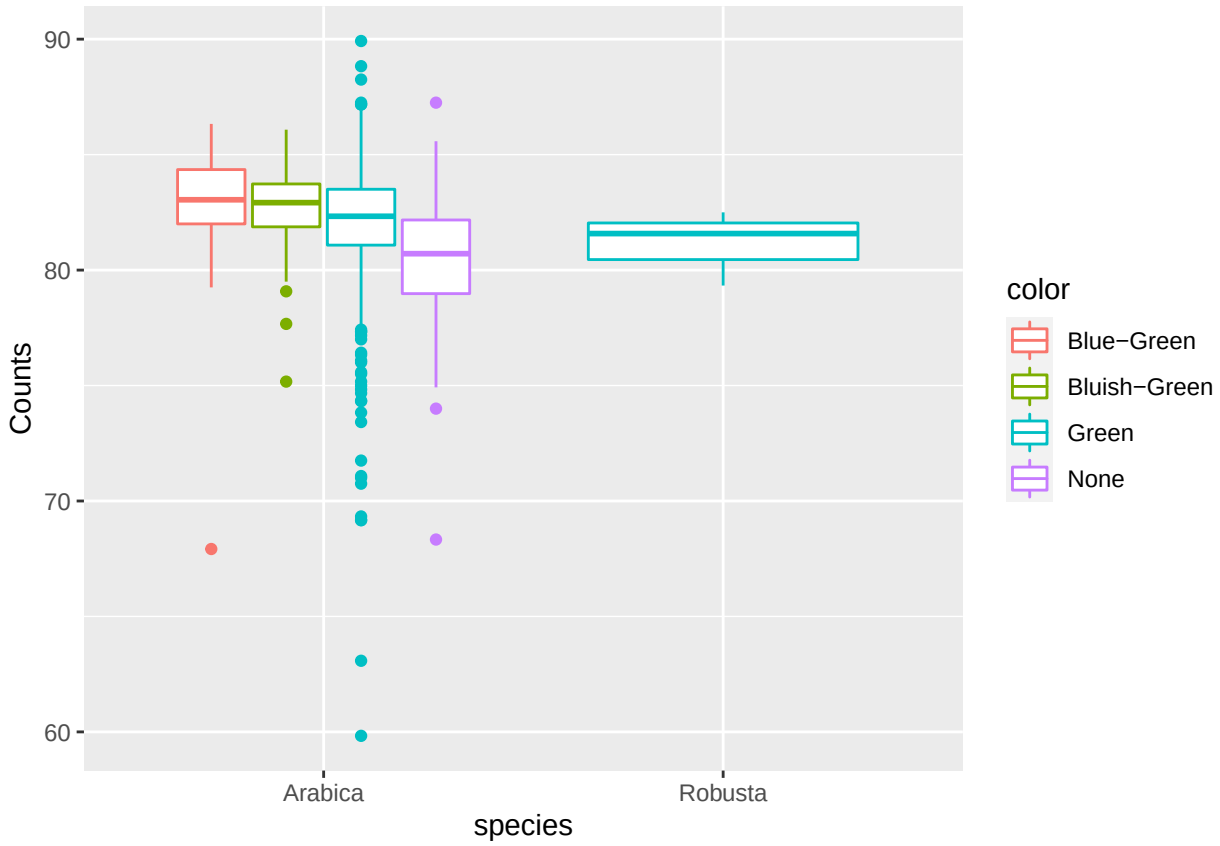
```
c.v2 = c.v1 %>%
  filter(Variables != 'avg_altitude' & Variables != 'category_one_defects'& Variables != 'category_two_
```

```
ggplot(c.v2,aes(x=species,y=Counts,color=color))+geom_boxplot()+facet_wrap(~Variables,scales = "free")
```

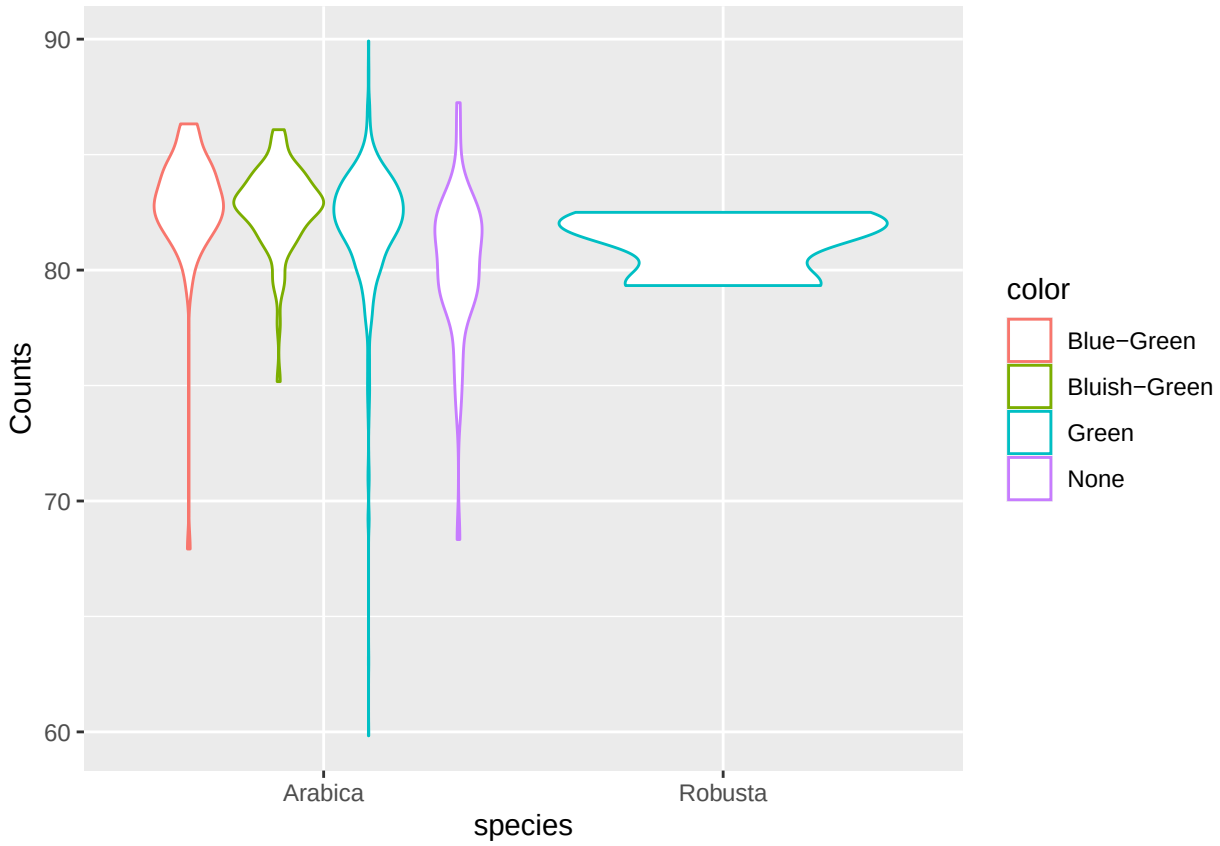


How are the cup points distributed and where the weight it is at

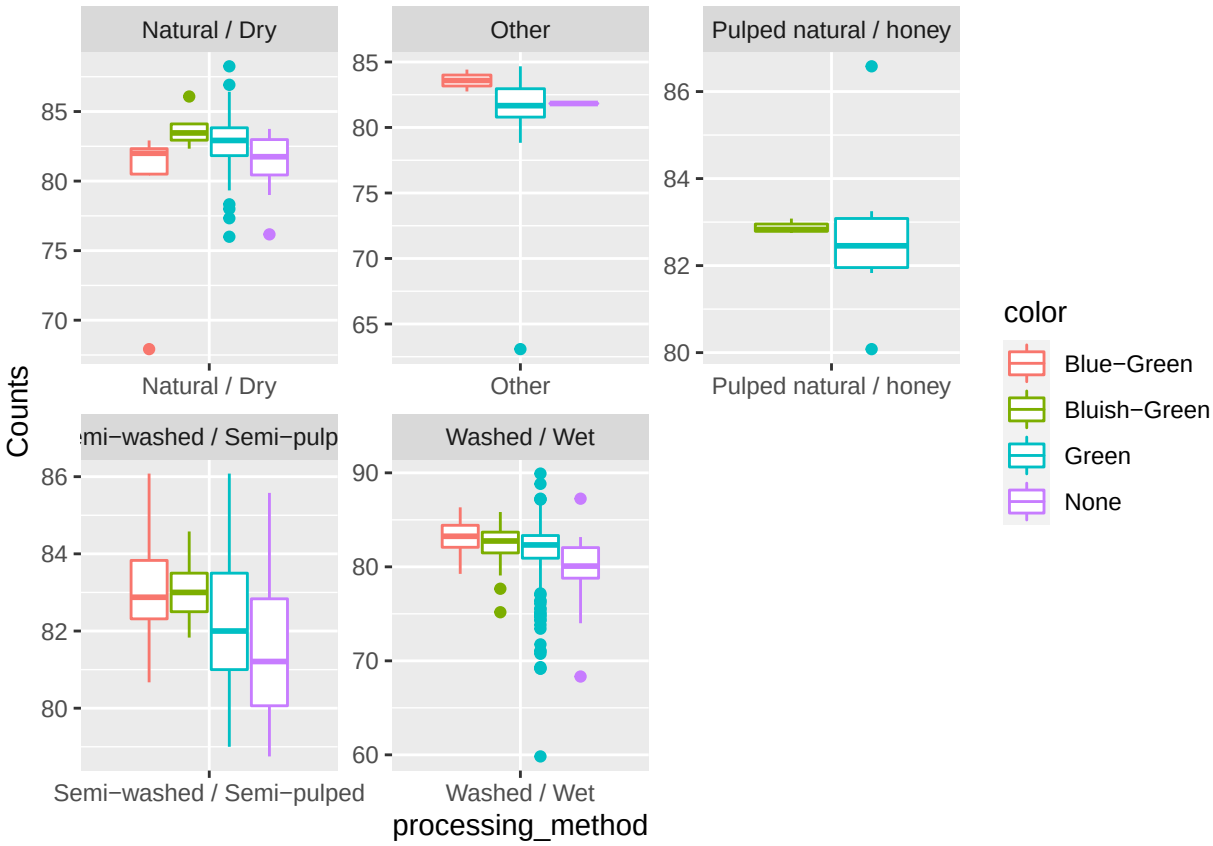
```
c.v2 %>%
  filter(Variables == 'total_cup_points')%>%
  ggplot(aes(x=species,y=Counts,color=color))+geom_boxplot()
```



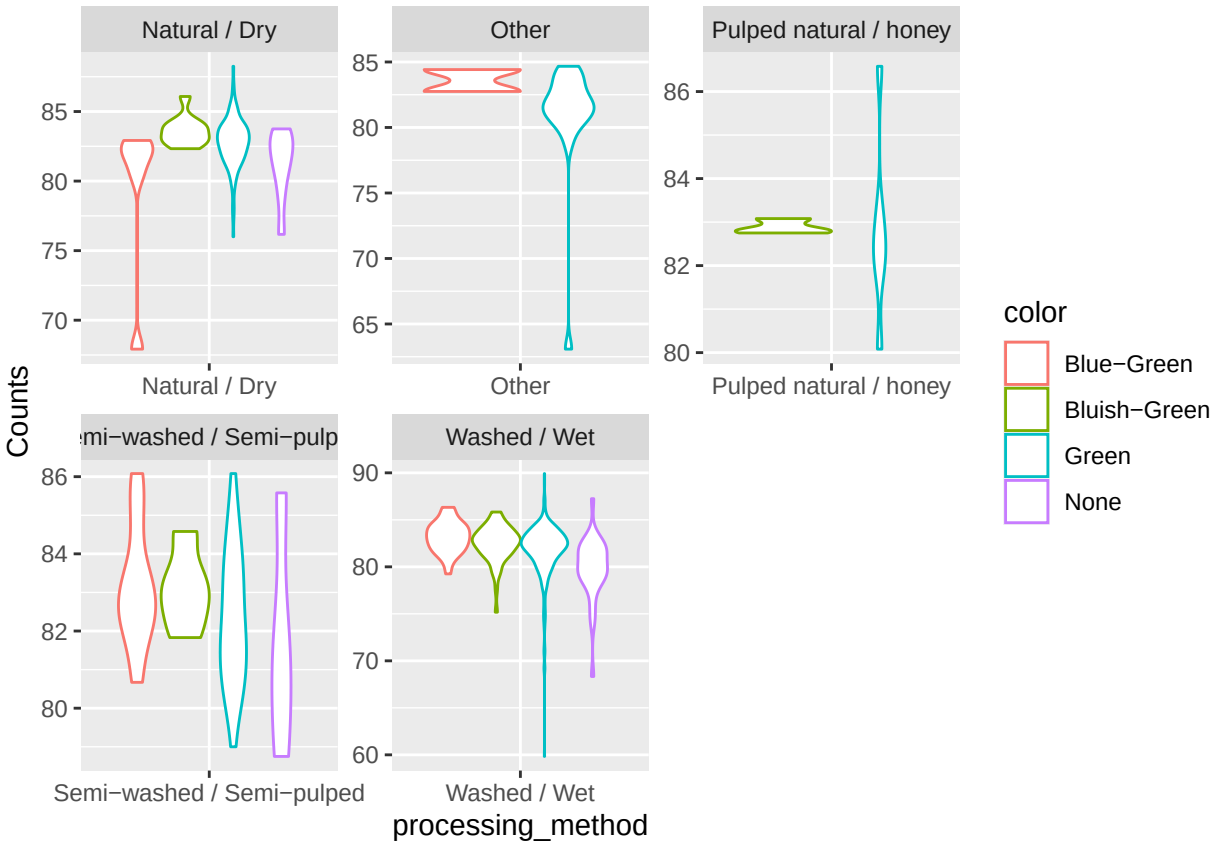

```
c.v2 %>%
  filter(Variables == 'total_cup_points')%>%
  ggplot(aes(x=species,y=Counts,color=color))+geom_violin()
```




```
c.v2 %>%
  filter(Variables == 'total_cup_points')%>%
  ggplot(aes(x=processing_method,y=Counts,color=color))+geom_boxplot()+
    facet_wrap(~processing_method,scales = "free")
```




```
c.v2 %>%
  filter(Variables == 'total_cup_points')%>%
  ggplot(aes(x=processing_method,y=Counts,color=color))+geom_violin()+
    facet_wrap(~processing_method,scales = "free")
```



See the data make-up in a different view

```
library(formattable)
```

```
#want to make this cleaner
c.v1 %>%
  group_by(Variables,Counts) %>%
  summarise(count = n()) %>%
  mutate(freq = formattable::percent(count / sum(count)))
```

```
## `summarise()` has grouped output by 'Variables'. You can override using the `.groups` argument.
```

```
## # A tibble: 611 x 4
## # Groups:   Variables [16]
##    Variables Counts count freq
##    <chr>      <dbl> <int> <formttbl>
##  1 acidity     5.25     1 0.11%
##  2 acidity     6.08     1 0.11%
##  3 acidity     6.25     1 0.11%
##  4 acidity     6.5      1 0.11%
##  5 acidity     6.67     3 0.34%
##  6 acidity     6.75     2 0.22%
##  7 acidity     6.83     6 0.67%
##  8 acidity     6.92     7 0.78%
##  9 acidity     7       23 2.57%
## 10 acidity     7.08    25 2.80%
## # ... with 601 more rows
```

Format the label (total_cup_points) to be categorical

```
coffee$tcp = coffee$total_cup_points
```

```
for(i in 1:894){
  if(coffee[i,29] >= 80){
    coffee[i,29] = 80
  }
  else if(coffee[i,29] >= 70 & coffee[i,29] < 80){
    coffee[i,29] = 70
  }
  else if(coffee[i,29] >= 60 & coffee[i,29] < 70){
    coffee[i,29] = 60
  }
  else{
    coffee[i,29] = 50
  }
}
coffee$tcp = round(coffee$tcp,0)
```

For analysis

```
df = coffee[,c(9:22,25,29)]
for(i in 4 : 13){
  df[,i]=round(df[,i],1)
}
```

Accuracy table for comparison between models

```
table_accuracy = matrix(nrow=6,ncol=1)
colnames(table_accuracy) = c('Accuracy')
rownames(table_accuracy) = c('DTree','NB','SVM-Linerar','SVM-Polynomial','ANN','KNN')
table_accuracy
```

```
##                 Accuracy
## DTree                 NA
## NB                    NA
## SVM-Linerar           NA
## SVM-Polynomial        NA
## ANN                   NA
## KNN                   NA
```

Set seed so analysis is repeatable

```
set.seed(1)
```

Simple k-fold cross validation(cv) function

```
cv = function(data,k){
  n = nrow(data)
  folds = k
  tail = n%/%folds

  set.seed(1)

  rnd = runif(n)
  rank = rank(rnd)

  #block/chunck from cv is blk
```

```
  blk = (rank-1)%/%tail+1
  blk = as.factor(blk)

  #to see formation of folds
  print(summary(blk))
}
```

```
n = nrow(df)
folds = 10
tail = n%/%folds

set.seed(1)

rnd = runif(n)
rank = rank(rnd)

#block/chunck from cv is blk
blk = (rank-1)%/%tail+1
blk = as.factor(blk)

#to see formation of folds
print(summary(blk))
```

```
##  1  2  3  4  5  6  7  8  9 10 11
## 89 89 89 89 89 89 89 89 89 89  4
```

```
df$tcp = as.factor(df$tcp)
df$moisture = round(df$moisture,1)
```

```
df = df[,c(4:13,16)]
```

```
library(rpart)
#dtree
set.seed(1)
all.acc = numeric(0)
for(i in 1:folds){
  tree = rpart(tcp~.,df[blk != i,],method="class")
  pred = predict(tree,df[blk==i,],type="class")
  confMat = table(pred,df$tcp[blk==i])
  acc = (confMat[1,1]+confMat[2,2]+confMat[3,3]+confMat[4,4])/sum(confMat)
  all.acc = rbind(all.acc,acc)
}

print(mean(all.acc))
```

```
## [1] 0.947191
```

```
table_accuracy[1,1] = mean(all.acc)
```

I re-formatted the label/target field and went from a binary (good/bad) grading and could not figure out why the accuracy was so low (0.003) and then looked into what the accuracy was calculating...

```
confMat
```

```
##
## pred 50 60 70 80
##   50  0  0  0  0
##   60  0  0  0  0
```

```
##   70  0  0 13  0
##   80  0  0  3 73
```

```
# naive Bayes (gaussian data)
library(e1071)
set.seed(1)

all.acc = numeric(0)
for(i in 1:folds){
  model = naiveBayes(tcp~.,df[blk != i,],method="class")
  pred = predict(model,df[blk==i,],type="class")
  confMat = table(pred,df$tcp[blk==i])
  acc = (confMat[1,1]+confMat[2,2]+confMat[3,3]+confMat[4,4])/sum(confMat)
  all.acc = rbind(all.acc,acc)
}

print(mean(all.acc))
```

```
## [1] 0.952809
```

```
table_accuracy[2,1] = mean(all.acc)
```

```
#svm linear
set.seed(1)

all.acc = numeric(0)
for(i in 1:folds){
  model = svm(tcp~.,df[blk != i,],kernel="linear",type="C")
  pred = predict(model,df[blk==i,],type="class")
  confMat = table(pred,df$tcp[blk==i])
  acc = (confMat[1,1]+confMat[2,2]+confMat[3,3]+confMat[4,4])/sum(confMat)
  all.acc = rbind(all.acc,acc)
}

print(mean(all.acc))
```

```
## [1] 0.9842697
```

```
table_accuracy[3,1] = mean(all.acc)
```

```
#svm poly
set.seed(1)

all.acc = numeric(0)
for(i in 1:folds){
  model = svm(tcp~.,df[blk != i,],kernel="polynomial",type="C")
  pred = predict(model,df[blk==i,],type="class")
  confMat = table(pred,df$tcp[blk==i])
  acc = (confMat[1,1]+confMat[2,2]+confMat[3,3]+confMat[4,4])/sum(confMat)
  all.acc = rbind(all.acc,acc)
}

print(mean(all.acc))
```

```
## [1] 0.9730337
```

```
table_accuracy[4,1] = mean(all.acc)
```

```
#ann
library(nnet)
set.seed(1)

all.acc = numeric(0)
for(i in 1:folds){
  model = nnet(tcp~.,df[blk != i,], size = 5, trace=FALSE, wgts=.05)
  pred = predict(model, df[blk==i,])
  confMat = table(pred,df$tcp[blk==i])
  acc = (confMat[1,1])/sum(confMat)
  all.acc = rbind(all.acc,acc)
}
print(mean(all.acc))
table_accuracy[5,1] = mean(all.acc)
```

```
library (caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
trControl <- trainControl(method  = "cv", number  = 10)
```

```
knn = df[,]
```

```
model <- train(tcp ~ .,
               method     = "knn",
               tuneGrid   = expand.grid(k = 1:10),
               trControl  = trControl,
               data       = knn)
```

```
acc = mean(model$results$Accuracy)
table_accuracy[6,1] = acc
```

```
tab = round(table_accuracy,4)
tab
```

```
##                 Accuracy
## DTree             0.9472
## NB                0.9528
## SVM-Linerar       0.9843
## SVM-Polynomial    0.9730
## ANN                   NA
## KNN               0.9735
```