Assignment 1

Contents

```r
coffee <- read.csv("coffee_selected.csv")

# Check first few rows
head(coffee)
```

```
##   is_specialty total_cup_points aroma flavor aftertaste acidity body balance
## 1         TRUE            90.58  8.67   8.83       8.67    8.75 8.50    8.42
## 2         TRUE            89.92  8.75   8.67       8.50    8.58 8.42    8.42
## 3         TRUE            89.75  8.42   8.50       8.42    8.42 8.33    8.42
## 4         TRUE            89.00  8.17   8.58       8.42    8.42 8.50    8.25
## 5         TRUE            88.83  8.25   8.50       8.25    8.50 8.42    8.33
## 6         TRUE            88.83  8.58   8.42       8.42    8.50 8.25    8.33
##   uniformity clean_cup category_one_defects moisture quakers processing_method
## 1         10        10                    0     0.12       0      Washed / Wet
## 2         10        10                    0     0.12       0      Washed / Wet
## 3         10        10                    0     0.00       0              <NA>
## 4         10        10                    0     0.11       0     Natural / Dry
## 5         10        10                    0     0.12       0      Washed / Wet
## 6         10        10                    0     0.11       0     Natural / Dry
##   category_two_defects altitude_mean_meters
## 1                    0                 2075
## 2                    1                 2075
## 3                    0                 1700
## 4                    2                 2000
## 5                    2                 2075
## 6                    1                   NA
```

```r
dim(coffee)
```

```
## [1] 1338   16
```

```r
names(coffee)
```

```
##  [1] "is_specialty"         "total_cup_points"     "aroma"               
##  [4] "flavor"               "aftertaste"           "acidity"             
##  [7] "body"                 "balance"              "uniformity"          
## [10] "clean_cup"            "category_one_defects" "moisture"            
## [13] "quakers"              "processing_method"    "category_two_defects"
## [16] "altitude_mean_meters"
```

```r
str(coffee)
```

```
## 'data.frame':    1338 obs. of  16 variables:
##  $ is_specialty        : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ total_cup_points    : num  90.6 89.9 89.8 89 88.8 ...
##  $ aroma               : num  8.67 8.75 8.42 8.17 8.25 8.58 8.42 8.25 8.67 8.08 ...
##  $ flavor              : num  8.83 8.67 8.5 8.58 8.5 8.42 8.5 8.33 8.67 8.58 ...
##  $ aftertaste          : num  8.67 8.5 8.42 8.42 8.25 8.42 8.33 8.5 8.58 8.5 ...
##  $ acidity             : num  8.75 8.58 8.42 8.42 8.5 8.5 8.5 8.42 8.42 8.5 ...
##  $ body                : num  8.5 8.42 8.33 8.5 8.42 8.25 8.25 8.33 8.33 7.67 ...
##  $ balance             : num  8.42 8.42 8.42 8.25 8.33 8.33 8.25 8.5 8.42 8.42 ...
##  $ uniformity          : num  10 10 10 10 10 10 10 10 9.33 10 ...
##  $ clean_cup           : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ category_one_defects: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ moisture            : num  0.12 0.12 0 0.11 0.12 0.11 0.11 0.03 0.03 0.1 ...
##  $ quakers             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ processing_method   : chr  "Washed / Wet" "Washed / Wet" NA "Natural / Dry" ...
##  $ category_two_defects: int  0 1 0 2 2 1 0 0 0 4 ...
##  $ altitude_mean_meters: num  2075 2075 1700 2000 2075 ...
```

```r
summary(coffee)
```

```
##  is_specialty    total_cup_points     aroma           flavor     
##  Mode :logical   Min.   :59.83    Min.   :5.080   Min.   :6.080  
##  FALSE:186       1st Qu.:81.10    1st Qu.:7.420   1st Qu.:7.330  
##  TRUE :1152      Median :82.50    Median :7.580   Median :7.580  
##                  Mean   :82.15    Mean   :7.572   Mean   :7.526  
##                  3rd Qu.:83.67    3rd Qu.:7.750   3rd Qu.:7.750  
##                  Max.   :90.58    Max.   :8.750   Max.   :8.830  
##                                                                  
##    aftertaste       acidity           body          balance     
##  Min.   :6.170   Min.   :5.250   Min.   :5.080   Min.   :5.250  
##  1st Qu.:7.250   1st Qu.:7.330   1st Qu.:7.330   1st Qu.:7.330  
##  Median :7.420   Median :7.580   Median :7.500   Median :7.500  
##  Mean   :7.407   Mean   :7.541   Mean   :7.523   Mean   :7.524  
##  3rd Qu.:7.580   3rd Qu.:7.750   3rd Qu.:7.670   3rd Qu.:7.750  
##  Max.   :8.670   Max.   :8.750   Max.   :8.580   Max.   :8.750  
##                                                                 
##    uniformity       clean_cup      category_one_defects    moisture      
##  Min.   : 6.000   Min.   : 0.000   Min.   : 0.0000      Min.   :0.00000  
##  1st Qu.:10.000   1st Qu.:10.000   1st Qu.: 0.0000      1st Qu.:0.09000  
##  Median :10.000   Median :10.000   Median : 0.0000      Median :0.11000  
##  Mean   : 9.842   Mean   : 9.842   Mean   : 0.4798      Mean   :0.08836  
##  3rd Qu.:10.000   3rd Qu.:10.000   3rd Qu.: 0.0000      3rd Qu.:0.12000  
##  Max.   :10.000   Max.   :10.000   Max.   :63.0000      Max.   :0.28000  
##                                                                          
##     quakers        processing_method  category_two_defects altitude_mean_meters
##  Min.   : 0.0000   Length:1338        Min.   : 0.000       Min.   :     1      
##  1st Qu.: 0.0000   Class :character   1st Qu.: 0.000       1st Qu.:  1100      
##  Median : 0.0000   Mode  :character   Median : 2.000       Median :  1311      
##  Mean   : 0.1735                      Mean   : 3.558       Mean   :  1775      
##  3rd Qu.: 0.0000                      3rd Qu.: 4.000       3rd Qu.:  1600      
##  Max.   :11.0000                      Max.   :55.000       Max.   :190164      
##  NA's   :1                                                 NA's   :230         
```

```r
# Count missing values per column
colSums(is.na(coffee))
```

```
##         is_specialty     total_cup_points                aroma 
##                    0                    0                    0 
##               flavor           aftertaste              acidity 
##                    0                    0                    0 
##                 body              balance           uniformity 
##                    0                    0                    0 
##            clean_cup category_one_defects             moisture 
##                    0                    0                    0 
##              quakers    processing_method category_two_defects 
##                    1                  169                    0 
## altitude_mean_meters 
##                  230
```

```r
# Impute quakers with 0
coffee$quakers[is.na(coffee$quakers)] <- 0

# Impute processing_method with "Unknown"
coffee$processing_method <- as.character(coffee$processing_method)
coffee$processing_method[is.na(coffee$processing_method)] <- "Unknown"
coffee$processing_method <- as.factor(coffee$processing_method)

# Impute altitude with median
median_alt <- median(coffee$altitude_mean_meters, na.rm = TRUE)
coffee$altitude_mean_meters[is.na(coffee$altitude_mean_meters)] <- median_alt
```

```r
# Check again
colSums(is.na(coffee))
```

```
##         is_specialty     total_cup_points                aroma 
##                    0                    0                    0
##               flavor           aftertaste              acidity
##                    0                    0                    0
##                 body              balance           uniformity
##                    0                    0                    0
##            clean_cup category_one_defects             moisture
##                    0                    0                    0
##              quakers    processing_method category_two_defects
##                    0                    0                    0
## altitude_mean_meters
##                    0
```

```
sum(is.na(coffee))
```

```
## [1] 0
```

```
# Check for duplicate rows
sum(duplicated(coffee))
```

```
## [1] 0
```

```
# Frequency table
table(coffee$is_specialty)
```

```
##
## FALSE  TRUE
##   186  1152
```

Although the dataset is imbalanced toward specialty coffees, this does not pose an issue for regression analysis, since the target variable (total_cup_points) is continuous and all observations contribute to the model.

```
# Proportions
prop.table(table(coffee$is_specialty))
```

```
##
##     FALSE      TRUE
## 0.1390135 0.8609865
```

```
num_cols <- c("total_cup_points","aroma","flavor","aftertaste",
              "acidity","body","balance","uniformity","clean_cup",
              "category_one_defects","category_two_defects",
              "moisture","quakers","altitude_mean_meters")

summary(coffee[, num_cols])
```

```
##  total_cup_points     aroma           flavor        aftertaste
##  Min.   :59.83    Min.   :5.080   Min.   :6.080   Min.   :6.170
##  1st Qu.:81.10    1st Qu.:7.420   1st Qu.:7.330   1st Qu.:7.250
##  Median :82.50    Median :7.580   Median :7.580   Median :7.420
##  Mean   :82.15    Mean   :7.572   Mean   :7.526   Mean   :7.407  
##  3rd Qu.:83.67    3rd Qu.:7.750   3rd Qu.:7.750   3rd Qu.:7.580  
##  Max.   :90.58    Max.   :8.750   Max.   :8.830   Max.   :8.670  
##     acidity           body          balance        uniformity    
##  Min.   :5.250   Min.   :5.080   Min.   :5.250   Min.   : 6.000  
##  1st Qu.:7.330   1st Qu.:7.330   1st Qu.:7.330   1st Qu.:10.000  
##  Median :7.580   Median :7.500   Median :7.500   Median :10.000  
##  Mean   :7.541   Mean   :7.523   Mean   :7.524   Mean   : 9.842  
##  3rd Qu.:7.750   3rd Qu.:7.670   3rd Qu.:7.750   3rd Qu.:10.000  
##  Max.   :8.750   Max.   :8.580   Max.   :8.750   Max.   :10.000  
##    clean_cup      category_one_defects category_two_defects    moisture      
##  Min.   : 0.000   Min.   : 0.0000      Min.   : 0.000       Min.   :0.00000  
##  1st Qu.:10.000   1st Qu.: 0.0000      1st Qu.: 0.000       1st Qu.:0.09000  
##  Median :10.000   Median : 0.0000      Median : 2.000       Median :0.11000  
##  Mean   : 9.842   Mean   : 0.4798      Mean   : 3.558       Mean   :0.08836  
##  3rd Qu.:10.000   3rd Qu.: 0.0000      3rd Qu.: 4.000       3rd Qu.:0.12000  
##  Max.   :10.000   Max.   :63.0000      Max.   :55.000       Max.   :0.28000  
##     quakers        altitude_mean_meters
##  Min.   : 0.0000   Min.   :     1      
##  1st Qu.: 0.0000   1st Qu.:  1200      
##  Median : 0.0000   Median :  1311      
##  Mean   : 0.1734   Mean   :  1695      
##  3rd Qu.: 0.0000   3rd Qu.:  1550      
##  Max.   :11.0000   Max.   :190164
```

Lets check Outliers

```r
# Function to flag outliers based on IQR
find_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1
  lower <- Q1 - 1.5 * IQR
  upper <- Q3 + 1.5 * IQR
  return(which(x < lower | x > upper))
}
```

Apply to Numeric Columns

```r
num_cols <- c("total_cup_points","aroma","flavor","aftertaste",
              "acidity","body","balance","uniformity","clean_cup",
              "category_one_defects","category_two_defects",
              "moisture","quakers","altitude_mean_meters")

outlier_list <- lapply(coffee[, num_cols], find_outliers)

# Check number of outliers per variable
sapply(outlier_list, length)
```

```
##     total_cup_points                aroma               flavor 
##                   73                   71                   43 
##           aftertaste              acidity                 body 
##                   86                   24                   33
##              balance           uniformity            clean_cup
##                   39                  186                  119
## category_one_defects category_two_defects             moisture
##                  202                   94                  305
##              quakers altitude_mean_meters
##                   94                  107
```

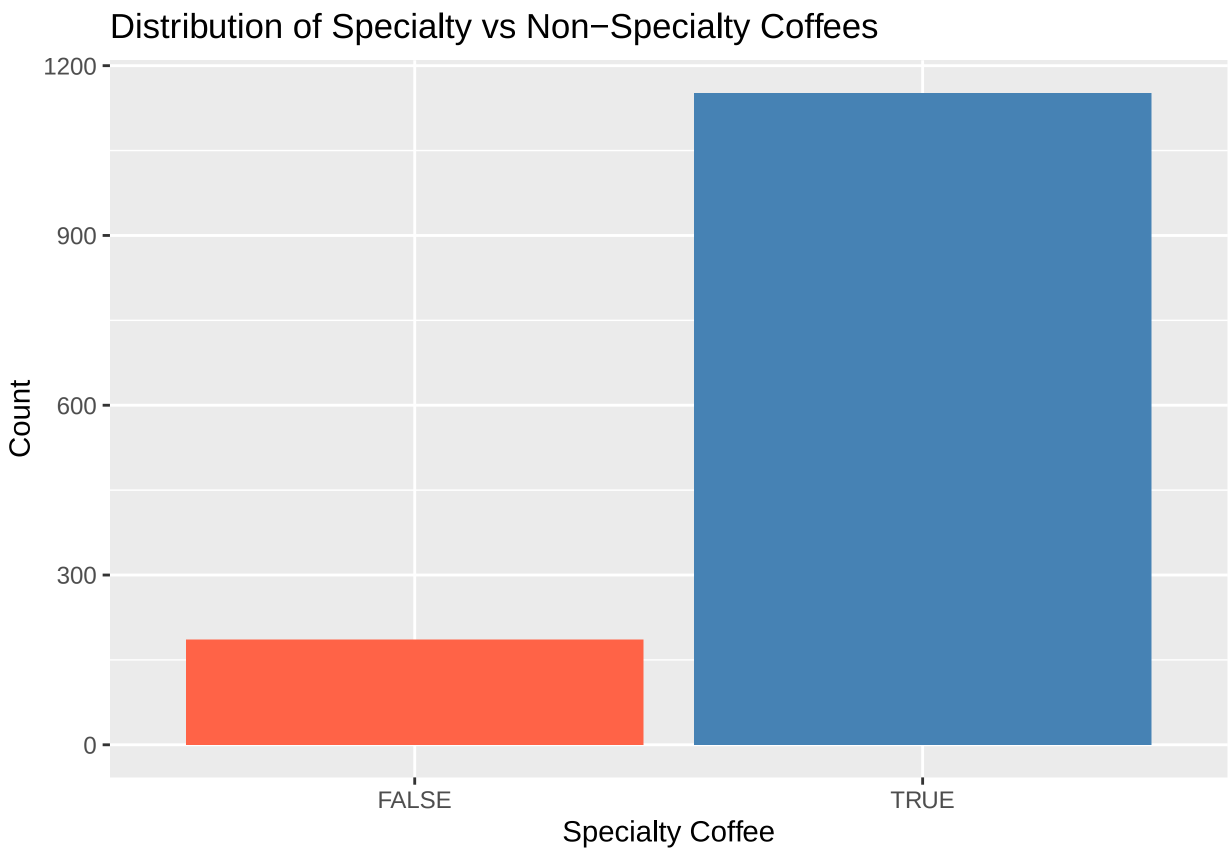
```
# Function to cap outliers at 1st and 99th percentile
cap_outliers <- function(x){
  qnt <- quantile(x, probs=c(0.01,0.99), na.rm=TRUE)
  x[x < qnt[1]] <- qnt[1]
  x[x > qnt[2]] <- qnt[2]
  return(x)
}

# Columns to cap (mostly numeric features prone to extreme values)
cap_cols <- c("total_cup_points","aroma","flavor","aftertaste",
              "acidity","body","balance","uniformity","clean_cup",
              "category_one_defects","category_two_defects",
              "moisture","quakers","altitude_mean_meters")

# Apply capping
coffee[cap_cols] <- lapply(coffee[cap_cols], cap_outliers)

# Check again for outliers
outlier_list_capped <- lapply(coffee[, cap_cols], find_outliers)
sapply(outlier_list_capped, length)
```

```
##     total_cup_points                aroma               flavor
##                   73                   71                   43
##           aftertaste              acidity                 body
##                   86                   15                   20
##              balance           uniformity            clean_cup
##                   39                  186                  119
## category_one_defects category_two_defects             moisture
##                  202                   94                  294
##              quakers altitude_mean_meters
##                   94                  107
```

The IQR technique was used to identify outliers. To lessen their effect on regression, extreme values in continuous variables were restricted at the first and 99th percentiles. Since they capture actual diversity in coffee quality, sensory scores that were inherently high or excellent were kept. The dataset is prepared for regression analysis following capping.

A)

```
# Bar plot of specialty vs non-specialty
library(ggplot2)

ggplot(coffee, aes(x=is_specialty)) +
  geom_bar(fill=c("tomato","steelblue")) +
  labs(title="Distribution of Specialty vs Non-Specialty Coffees",
       x="Specialty Coffee", y="Count")
```



Interpretation:

Most coffees are specialty grade.

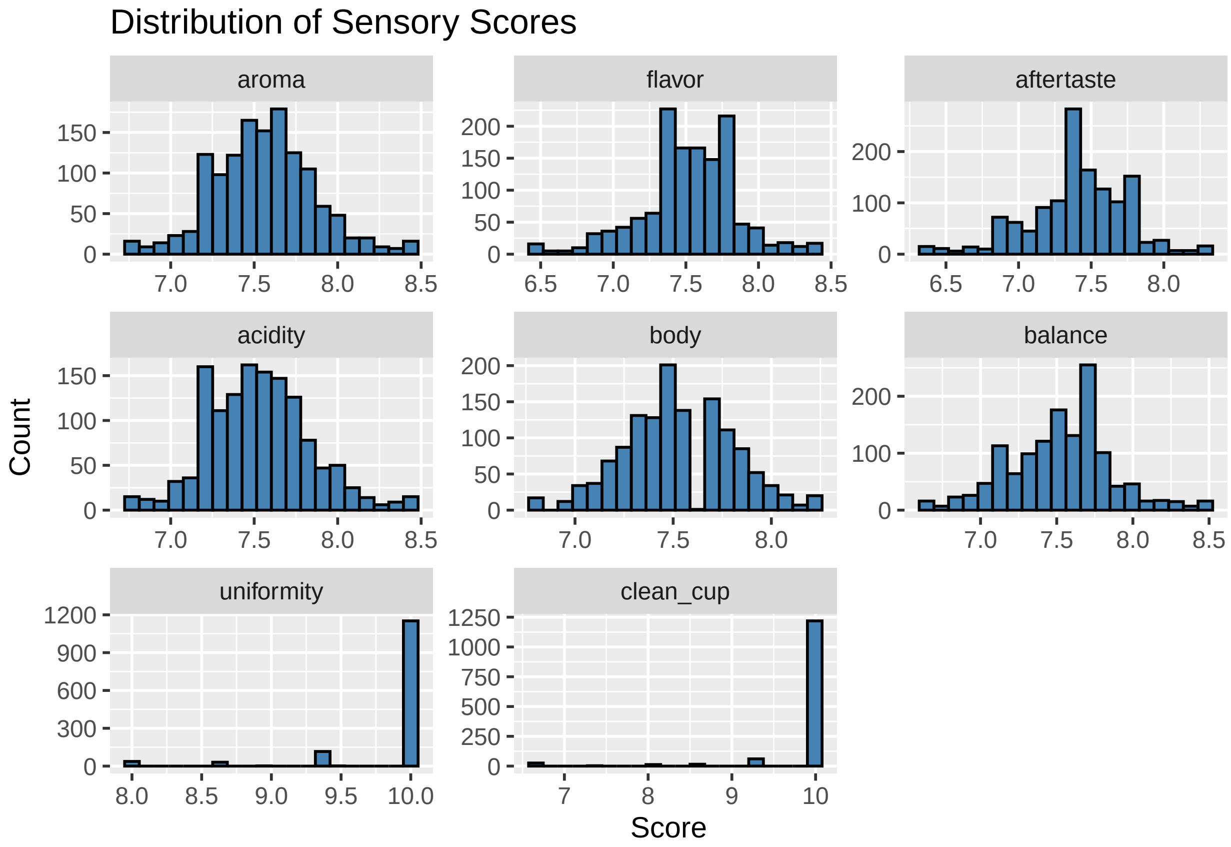
Dataset is imbalanced toward specialty coffees.

Important for understanding dataset composition

```
# Sensory attributes
sensory_cols <- c("aroma","flavor","aftertaste","acidity","body","balance","uniformity","clean_cup")

library(reshape2)
sensory_long <- melt(coffee[, sensory_cols])
```

```
## No id variables; using all as measure variables
```

```
ggplot(sensory_long, aes(x=value)) +
  geom_histogram(bins=20, fill="steelblue", color="black") +
  facet_wrap(~variable, scales="free") +
  labs(title="Distribution of Sensory Scores", x="Score", y="Count")
```



Interpretation:

Scores mostly clustered between 7–8.

uniformity and clean_cup show many perfect scores (10).

Left-skewed variables indicate most coffees are high quality.

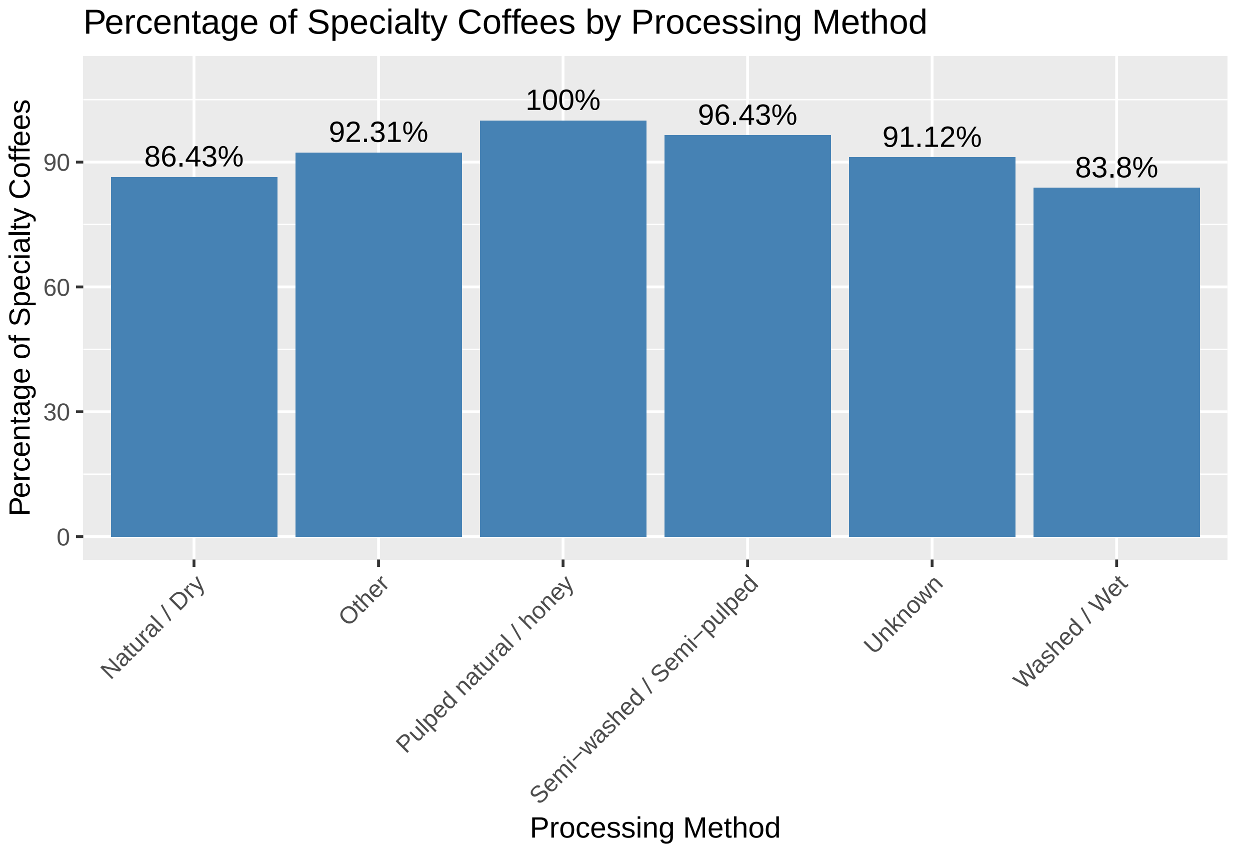
1 Question 1

1.1 Q1(a) – Graphical Exploration

```r
# A1. Proportion of specialty by processing_method
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

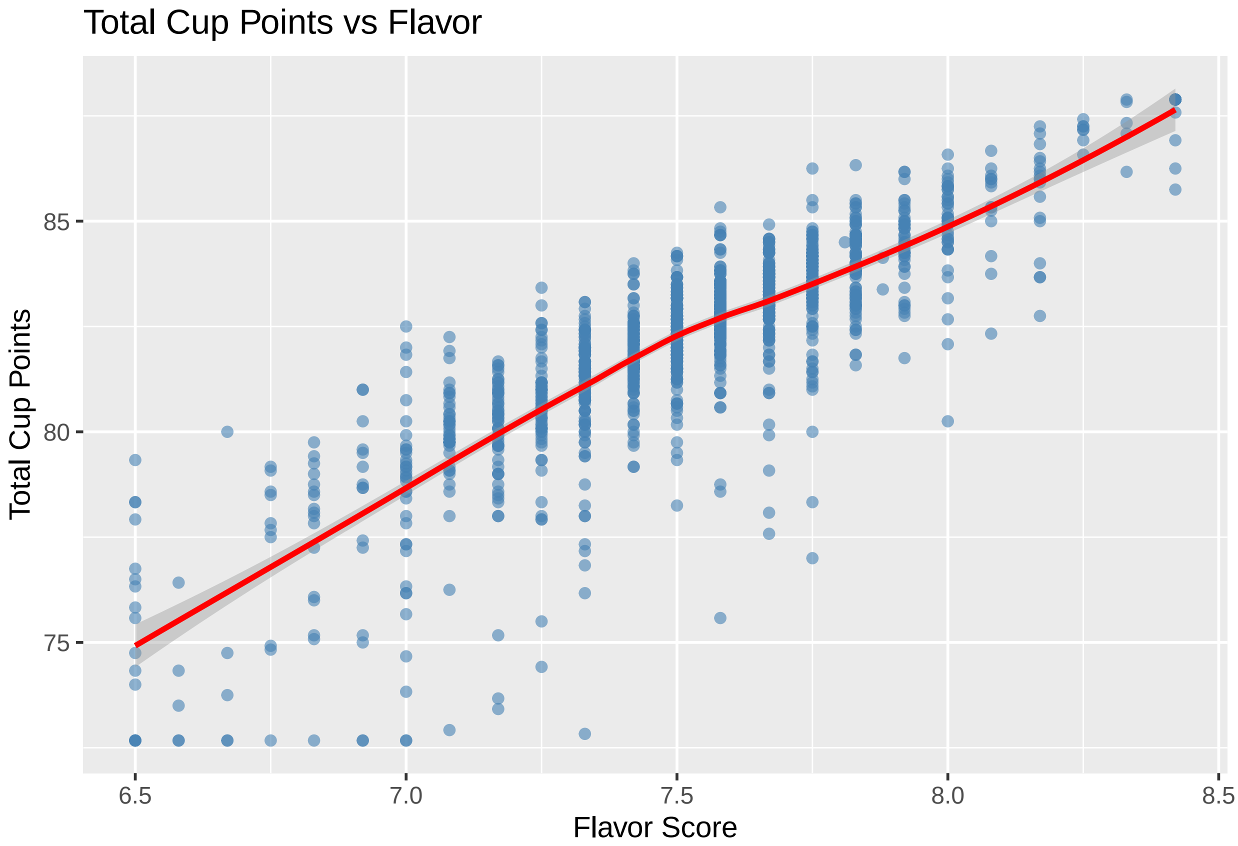
```r
library(ggplot2)

library(dplyr)
library(ggplot2)

# Calculate percentage of specialty coffees per processing method
specialty_percent <- coffee %>%
  group_by(processing_method) %>%
  summarise(total = n(),
            specialty_count = sum(is_specialty),
            percent_specialty = round((specialty_count / total) * 100, 2))

# Bar plot with percentage
ggplot(specialty_percent, aes(x=processing_method, y=percent_specialty)) +
  geom_bar(stat="identity", fill="steelblue") +
  geom_text(aes(label=paste0(percent_specialty, "%")), vjust=-0.5) +  # show percentage on top
  labs(title="Percentage of Specialty Coffees by Processing Method",
       x="Processing Method", y="Percentage of Specialty Coffees") +
  ylim(0, 110) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Different processing techniques result in different percentages of speciality coffees. The highest percentage, 100%, is found in the "Pulped natural / honey" approach, which is followed by the "Semi-washed / Semi-pulped" method, 96.43%. Despite having the most samples, the "Washed / Wet" method has the lowest proportion (83.8%). This suggests that whether a coffee qualifies as a speciality depends in large part on the processing technique.

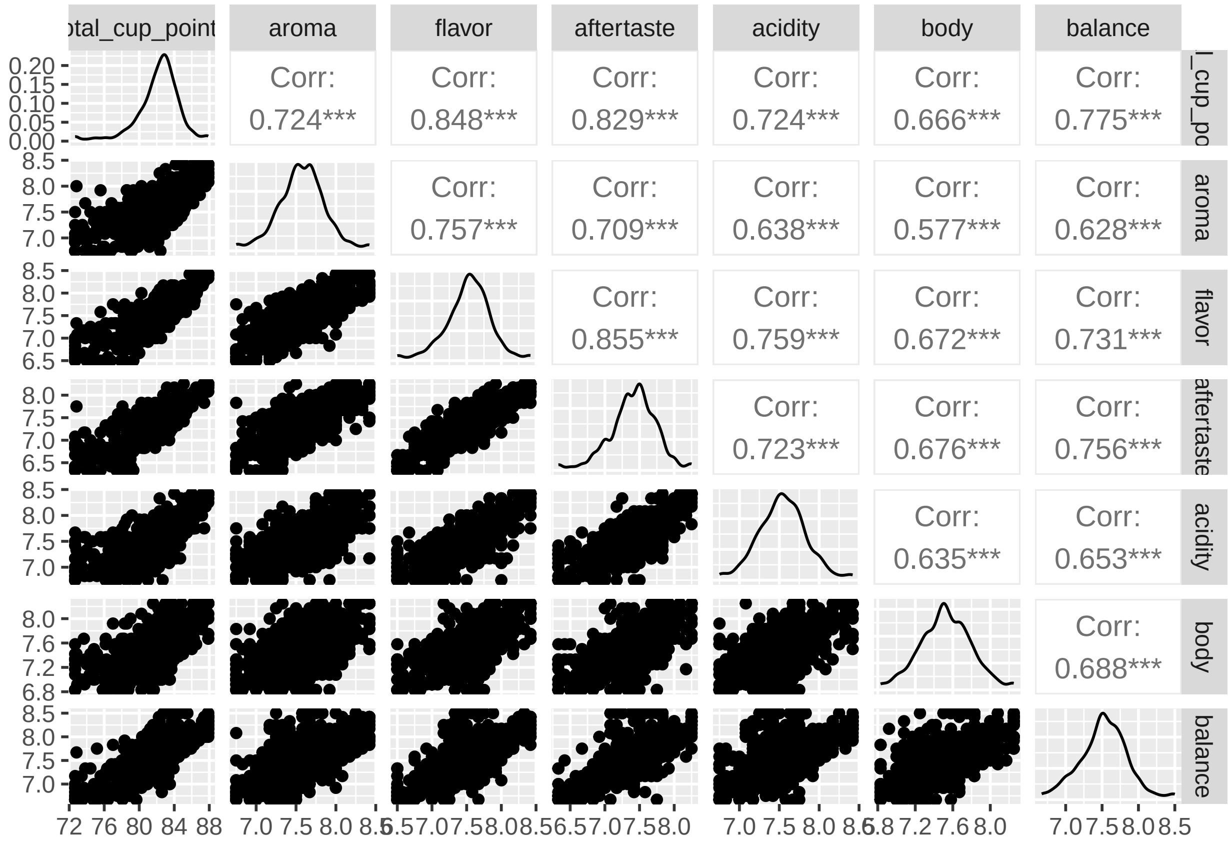
```r
# A2. total_cup_points vs flavor with smooth trend
# Scatter plots with smooth trend line
ggplot(coffee, aes(x=flavor, y=total_cup_points)) +
  geom_point(alpha=0.6, color="steelblue") +
  geom_smooth(method="loess", color="red") +
  labs(title="Total Cup Points vs Flavor", x="Flavor Score", y="Total Cup Points")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



With greater flavour scores typically translating into higher overall points, the plot clearly demonstrates a positive link between flavour and cup points. The majority of coffees fall into the mid-to-high flavour (7–8) and total cup point (80–85) clusters, which is indicative of the dataset's generally good quality. Very high flavour scores do not necessarily translate into proportionately higher total cup points, as indicated by the LOESS trend line's minor flattening at the top. There is some variation around the trend, which emphasises how other sensory qualities like aroma, aftertaste, and balance all affect the overall quality.

```r
# A3. Correlation matrix
vars_corr <- dplyr::select(
coffee,
total_cup_points, aroma, flavor, aftertaste, acidity, body, balance
)
vars_corr <- tidyr::drop_na(vars_corr)
GGally::ggpairs(
vars_corr,
upper = list(continuous = "cor"),
lower = list(continuous = "points"),
diag = list(continuous = "densityDiag")
)
```



According to the correlation matrix, every important sensory attribute—including flavour, aroma, and aftertaste has a positive relationship with overall coffee quality, emphasising how much each one contributes to the overall assessment. Taste and aftertaste are especially crucial for score prediction because they seem to be the best indicators of quality among them. Additionally, the matrix demonstrates the interdependence of numerous traits, indicating that enhancements in one area frequently follow those in other areas. Although flavour and aftertaste are useful predictors for modelling, multicollinearity should be taken into account due to the strong correlations between the variables, and cautious modelling techniques could be required to guarantee accurate and comprehensible findings.

1.2 Q1(b) – Model for category_one_defects

```
## check poisson model if it fit or not due to overdispersion


# Load necessary library
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Round category_one_defects to integers to avoid warnings
coffee$category_one_defects_int <- round(coffee$category_one_defects)

# Fit a Poisson regression model using integer counts
poisson_model <- glm(category_one_defects_int ~ processing_method + moisture +
                       altitude_mean_meters + quakers,
                     data = coffee,
                     family = poisson(link="log"))

# Summary of the model
summary(poisson_model)
```

```
##
## Call:
## glm(formula = category_one_defects_int ~ processing_method +
##     moisture + altitude_mean_meters + quakers, family = poisson(link = "log"),
##     data = coffee)
##
## Coefficients:
##                                               Estimate Std. Error z value
## (Intercept)                                 -0.4812916  0.1752863  -2.746
## processing_methodOther                      -0.3668489  0.2990724  -1.227
## processing_methodPulped natural / honey     -2.2269084  1.0083548  -2.208
## processing_methodSemi-washed / Semi-pulped -1.7606113  0.3866840  -4.553
## processing_methodUnknown                    -0.7838805  0.1729581  -4.532
## processing_methodWashed / Wet               -0.7072632  0.1015662  -6.964
## moisture                                     4.9039381  1.1063718   4.432
## altitude_mean_meters                        -0.0003193  0.0001076  -2.967
## quakers                                     -0.0317705  0.0724781  -0.438
##                                             Pr(>|z|)
## (Intercept)                                  0.00604 **
## processing_methodOther                       0.21996
## processing_methodPulped natural / honey      0.02721 *
## processing_methodSemi-washed / Semi-pulped 5.29e-06 ***
## processing_methodUnknown                    5.84e-06 ***
## processing_methodWashed / Wet               3.32e-12 ***
## moisture                                    9.32e-06 ***
## altitude_mean_meters                         0.00300 **
## quakers                                      0.66114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2142.8  on 1337  degrees of freedom
## Residual deviance: 2040.3  on 1329  degrees of freedom
## AIC: 2574.6
##
## Number of Fisher Scoring iterations: 6
```

```r
# Check for overdispersion
dispersion <- sum(residuals(poisson_model, type = "pearson")^2) / poisson_model$df.residual
dispersion
```

```
## [1] 3.433655
```

Poisson regression makes sense because category_one_defects is a count variable; nevertheless, the dispersion value of 3.43 shows significant overdispersion, which goes against the Poisson assumption of equal mean and variance. Since negative binomial regression uses an additional dispersion parameter to account for overdispersion, it is more suitable.

```r
## ==========================
## Q1(b) - Model for category_one_defects
## ==========================
library(MASS)

# Fit negative binomial regression
nb_model <- glm.nb(category_one_defects_int ~ processing_method + moisture +
                     altitude_mean_meters + quakers,
                   data = coffee)

# Summary of the model
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = category_one_defects_int ~ processing_method +
##     moisture + altitude_mean_meters + quakers, data = coffee,
##     init.theta = 0.1255674082, link = log)
##
## Coefficients:
##                                               Estimate Std. Error z value
## (Intercept)                                 -0.4304711  0.3653287  -1.178
## processing_methodOther                      -0.3931451  0.6591202  -0.596
## processing_methodPulped natural / honey     -2.0578436  1.2165518  -1.692
## processing_methodSemi-washed / Semi-pulped -1.6885154  0.5696898  -2.964
## processing_methodUnknown                    -0.7557769  0.3312868  -2.281
## processing_methodWashed / Wet               -0.6905529  0.2304853  -2.996
## moisture                                     3.4185587  2.0275494   1.686
## altitude_mean_meters                        -0.0002686  0.0002147  -1.251
## quakers                                      0.0160915  0.1372888   0.117
##                                             Pr(>|z|)
## (Intercept)                                  0.23867
## processing_methodOther                       0.55086
## processing_methodPulped natural / honey      0.09073 .
## processing_methodSemi-washed / Semi-pulped  0.00304 **
## processing_methodUnknown                     0.02253 *
## processing_methodWashed / Wet                0.00273 **
## moisture                                     0.09179 .
## altitude_mean_meters                         0.21098
## quakers                                      0.90669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.1256) family taken to be 1)
##
##     Null deviance: 559.29  on 1337  degrees of freedom
## Residual deviance: 535.80  on 1329  degrees of freedom
## AIC: 1790
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.1256
##           Std. Err.:  0.0138
##
##  2 x log-likelihood:  -1769.9860
```

```r
# Optional: Exponentiate coefficients to interpret as multiplicative effects
exp(coef(nb_model))
```

```
##                               (Intercept)
##                                 0.6502027
##                    processing_methodOther
##                                 0.6749308
##    processing_methodPulped natural / honey
##                                 0.1277291
## processing_methodSemi-washed / Semi-pulped
##                                 0.1847937
##                  processing_methodUnknown
##                                 0.4696456
##             processing_methodWashed / Wet
##                                 0.5012988
##                                  moisture
##                                30.5253869
##                      altitude_mean_meters
##                                 0.9997315
##                                   quakers
##                                 1.0162216
```

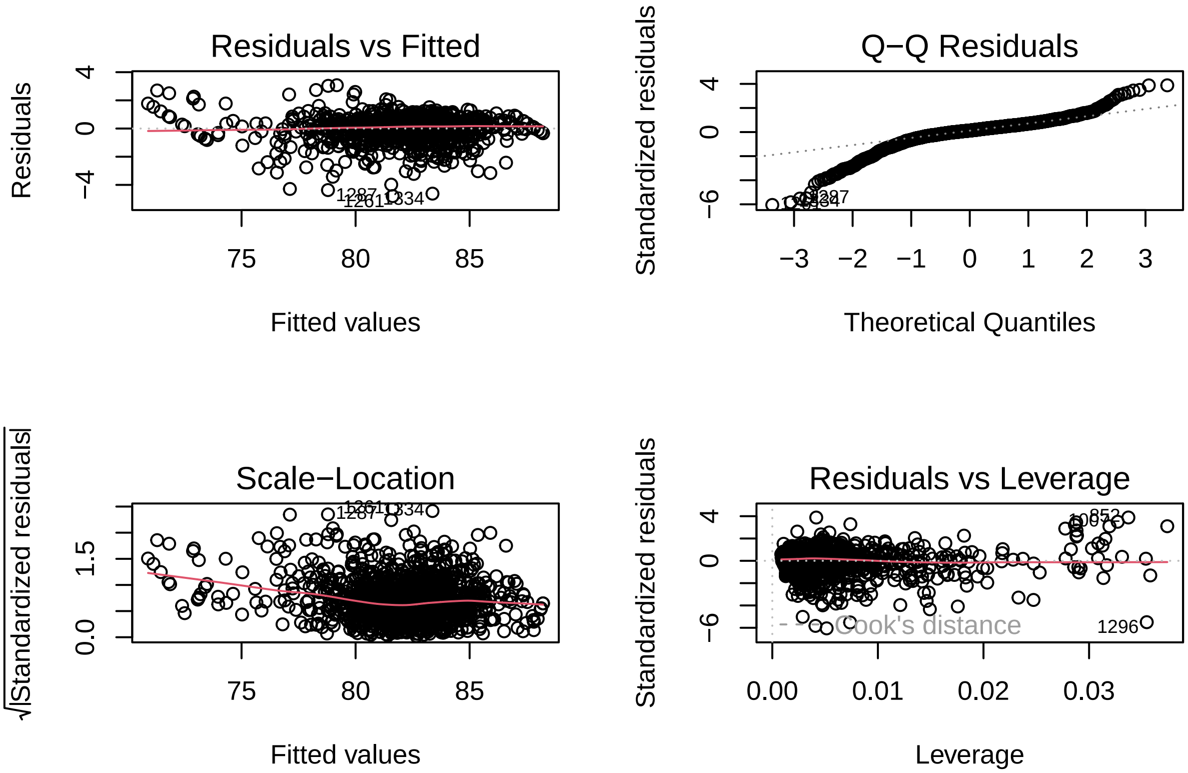
According to the data, the processing method has the biggest impact on how many coffee beans have category one flaws. While the Pulped Natural/Honey approach has a minor impact, the Semi-washed, Unknown, and Washed/Wet methods considerably minimise predicted faults. The marginally beneficial effect of moisture content suggests that increased moisture levels may marginally enhance faults. Defect counts are not considerably impacted by altitude or quaker population. These findings imply that the key to reducing flaws and guaranteeing better coffee quality is meticulous control over processing techniques along with moisture level monitoring.

1.3 Q1(c) – Linear model for total_cup_points

```r
# Fit the regression model
total_points_model <- lm(total_cup_points ~ aroma + balance + clean_cup +
                         flavor + moisture + altitude_mean_meters,
                         data = coffee)

# Summary of the model
summary(total_points_model)
```

```
##
## Call:
## lm(formula = total_cup_points ~ aroma + balance + clean_cup +
##     flavor + moisture + altitude_mean_meters, data = coffee)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7947 -0.2277  0.0973  0.4117  3.0735
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.686e+01  6.521e-01  25.855   <2e-16 ***
## aroma                1.279e+00  1.121e-01  11.413   <2e-16 ***
## balance              2.106e+00  9.707e-02  21.695   <2e-16 ***
## clean_cup            1.638e+00  4.192e-02  39.067   <2e-16 ***
## flavor               3.122e+00  1.172e-01  26.650   <2e-16 ***
## moisture             7.418e-01  4.705e-01   1.577    0.115
## altitude_mean_meters 4.559e-05  5.036e-05   0.905    0.366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7942 on 1331 degrees of freedom
## Multiple R-squared:  0.8974, Adjusted R-squared:  0.897
## F-statistic:  1941 on 6 and 1331 DF,  p-value: < 2.2e-16
```

```
# Residual plots to check linearity, homoscedasticity, and normality
par(mfrow = c(2,2))
plot(total_points_model)
```



According to the regression study, sensory characteristics such as flavour, clean cup, balance, and scent significantly influence total cup scores, with flavour being the most significant predictor. Balance and scent also have a significant role in these, although environmental factors like height and wetness have very little impact. Coffee quality is mostly determined by sensory factors, since the model has a high degree of fit, accounting for around 90% of the variation in total cup points (Adjusted R2 = 0.897). Relatively small and symmetric residuals demonstrate that the model accurately depicts the relationships in the data and do not exhibit any significant deviations from the assumptions of linear regression. All things considered, these results demonstrate that sensory qualities account for the majority of coffee ratings, with contextual influences having a negligible impact.

1.4 Q1(d) – Alternative model (GAM) + diagnostics & comparison

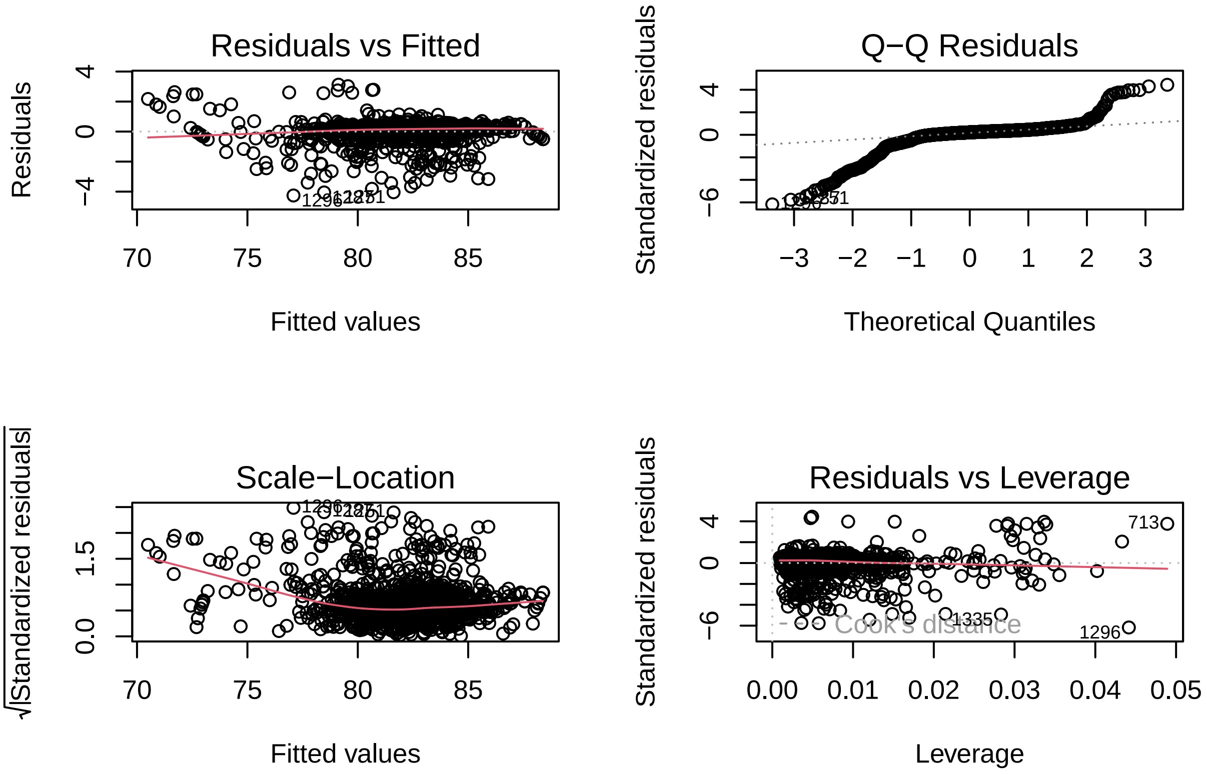
```
# Load necessary packages
library(broom)
library(ggplot2)

# Fit the alternative linear model
alt_model <- lm(total_cup_points ~ aroma + flavor + aftertaste + acidity + body + balance + clean_cup,
                data = coffee)

# Summarise results using broom
tidy(alt_model)
```

```
## # A tibble: 8 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   12.8      0.653      19.6  2.40e- 75
## 2 aroma          0.962    0.101       9.54 6.37e- 21
## 3 flavor         1.64     0.133      12.4  2.16e- 33
## 4 aftertaste     1.21     0.120      10.0  7.00e- 23
## 5 acidity        1.03     0.103      10.0  6.24e- 23
## 6 body           0.856    0.105       8.13 9.66e- 16
## 7 balance        1.29     0.0960     13.4  1.59e- 38
## 8 clean_cup      1.71     0.0378     45.3  5.42e-272
```

```r
glance(alt_model)  # Gives overall model fit metrics
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.919         0.919 0.705     2161.       0     7 -1427. 2872. 2919.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
# Residual plots to check assumptions
par(mfrow = c(2,2))
plot(alt_model)
```



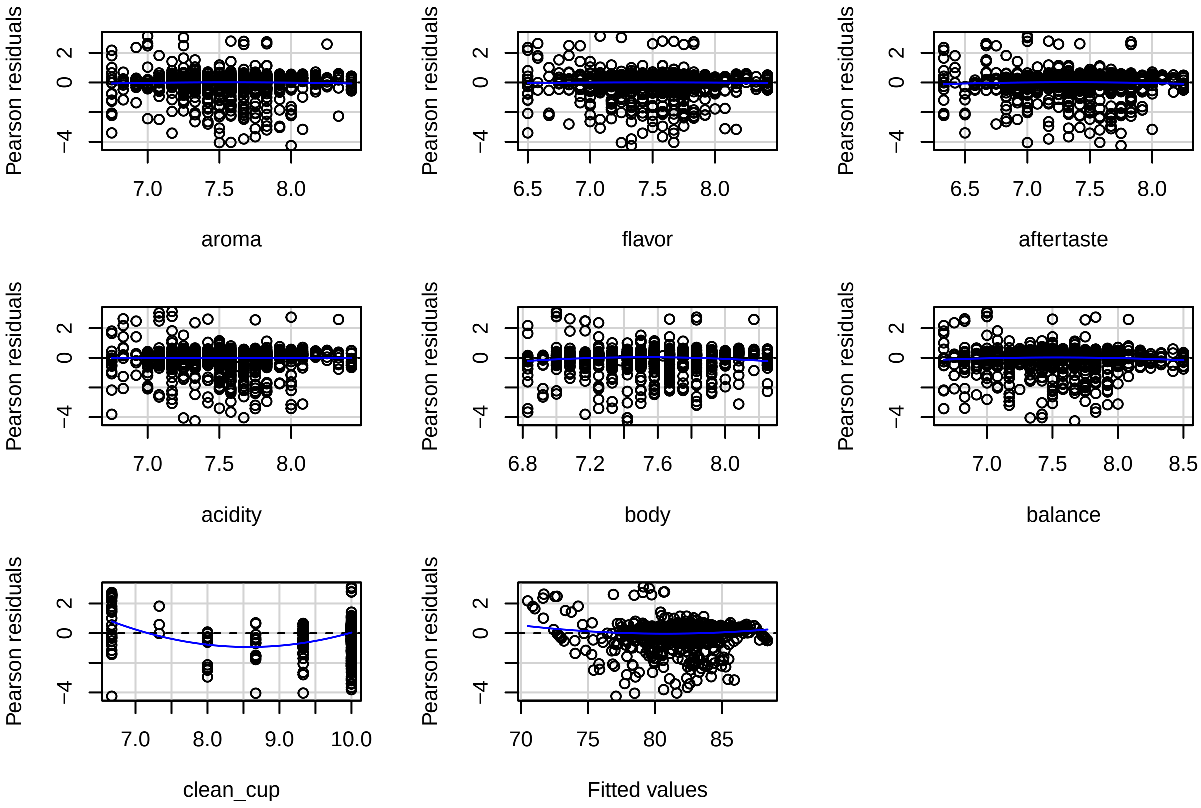
```r
# Optional: Standardized residuals
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
residualPlots(alt_model)
```



```
##            Test stat Pr(>|Test stat|)
## aroma        -0.8232         0.410550
## flavor       -0.4247         0.671147
## aftertaste   -1.2189         0.223084
## acidity      -0.2539         0.799599
## body         -2.9246         0.003507 **
## balance      -1.9904         0.046753 *
## clean_cup    12.1949        < 2.2e-16 ***
## Tukey test    3.9345        8.337e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Compare R-squared and Adjusted R-squared
summary(total_points_model)$r.squared # old model r square
```

```
## [1] 0.8974151
```

```
summary(alt_model)$r.squared #new model r square
```

```
## [1] 0.9191969
```

```
summary(total_points_model)$adj.r.squared #old model adjusted model
```

```
## [1] 0.8969527
```

```
summary(alt_model)$adj.r.squared #new model adjusted r square
```

```
## [1] 0.9187716
```

```
# Compare AIC values
AIC(alt_model)
```

```
## [1] 2872.268
```

```
AIC(lm(total_cup_points ~ aroma + balance + clean_cup + flavor + moisture + altitude_mean_meters, data
```

```
## [1] 3189.615
```

With an R2 of 0.919 and an adjusted R2 of 0.919, the alternative regression model, which uses aroma, flavour, aftertaste, acidity, body, balance, and clean cup as predictors, explains a significant amount of the variation in total cup points, demonstrating an excellent match. Higher evaluations for these qualities significantly raise total cup points, as evidenced by the positive coefficients and high significance of all sensory variables. Clean cup, flavour, and balance have the most impacts among them, underscoring their crucial part in the overall quality of coffee.

The residual standard error is comparatively low, indicating an accurate forecast of the total cup points, and residual diagnostics reveal no significant breaches of the linear regression assumptions. The greater R2 and lower AIC (2872 vs. 3189) in this model demonstrate its evident superiority over the prior model in component (c), which contained fewer sensory variables and weaker environmental factors. Overall, the results support the idea that sensory factors dominate coffee quality, with environmental factors like altitude and moisture having little effect after sensory factors are taken into account. This model offers a more thorough and accurate knowledge of the factors influencing coffee ratings.

1.5 Q1(e) – category_two_defects ~ moisture + processing_method

```
#Checking over dispersion first
# Round the variable to nearest integer
coffee$category_two_defects_int <- round(coffee$category_two_defects)
```

```r
# Fit Poisson model using integer counts
poisson_model <- glm(category_two_defects_int ~ moisture + processing_method,
                     family = poisson(link = "log"), data = coffee)

# Check overdispersion
dispersion <- sum(residuals(poisson_model, type = "pearson")^2) / poisson_model$df.residual
dispersion
```

```
## [1] 6.108346
```

A dispersion value of 6.11 is much greater than 1, which indicates strong overdispersion.

This means a Negative Binomial regression is more appropriate than Poisson for modeling category_two_defects.

```r
# Fit negative binomial model
nb_model <- glm.nb(category_two_defects_int ~ moisture + processing_method, data = coffee)

# Summary of the model
summary(nb_model)
```

```
## 
## Call:
## glm.nb(formula = category_two_defects_int ~ moisture + processing_method, 
##     data = coffee, init.theta = 0.759376421, link = log)
## 
## Coefficients:
##                                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                                 1.04857    0.10146  10.335  < 2e-16
## moisture                                    3.54512    0.77423   4.579 4.67e-06
## processing_methodOther                     -0.40916    0.26827  -1.525    0.127
## processing_methodPulped natural / honey    -0.47407    0.35922  -1.320    0.187
## processing_methodSemi-washed / Semi-pulped -0.18475    0.18751  -0.985    0.325
## processing_methodUnknown                   -0.60813    0.13165  -4.619 3.85e-06
## processing_methodWashed / Wet              -0.08959    0.09048  -0.990    0.322
##                                               
## (Intercept)                                ***
## moisture                                   ***
## processing_methodOther                        
## processing_methodPulped natural / honey       
## processing_methodSemi-washed / Semi-pulped    
## processing_methodUnknown                   ***
## processing_methodWashed / Wet                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(0.7594) family taken to be 1)
## 
##     Null deviance: 1526.6  on 1337  degrees of freedom
## Residual deviance: 1469.2  on 1331  degrees of freedom
## AIC: 6275.9
## 
## Number of Fisher Scoring iterations: 1
```

```
##
##
##               Theta:  0.7594
##           Std. Err.:  0.0398
##
##  2 x log-likelihood:  -6259.9190
```

```
# Optional: Analysis of deviance to check processing_method significance
anova(nb_model, test = "Chisq")
```

```
## Warning in anova.negbin(nb_model, test = "Chisq"): tests made without
## re-estimating 'theta'
```

```
## Analysis of Deviance Table
##
## Model: Negative Binomial(0.7594), link: log
##
## Response: category_two_defects_int
##
## Terms added sequentially (first to last)
##
##
##                   Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                               1337     1526.6
## moisture           1   33.118      1336     1493.5 8.674e-09 ***
## processing_method  5   24.286      1331     1469.2 0.0001913 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Moisture is a crucial impact, as demonstrated by the examination of category two problems, with higher moisture levels being linked to more defects. With the exception of the Unknown category, which considerably lowers predicted faults in comparison to the reference, the majority of processing techniques have minimal impact once moisture is taken into account. Reliable estimates are guaranteed by the Negative Binomial model's suitable handling of data overdispersion. These results imply that while differences in the majority of processing procedures have little effect on secondary flaws, managing moisture is essential for coffee quality. Because of their small sample sizes, sparse or atypical categories, such as Unknown, should be evaluated with caution.

1.6 Q1(f) –Reflection on Modelling Strategy and Challenges

The GLM modelling approach used in this investigation was customised for each response variable's type and distribution. Poisson regression was first used to model count variables, such as category one and two problems, however overdispersion checks suggested that a Negative Binomial model would be more suitable. Linear regression offered a solid baseline for continuous outcomes such as total_cup_points, and adding more sensory attributes to the model enhanced fit and explained variation.

Outliers and null values were eliminated prior to modelling in order to guarantee reliable results and avoid estimate distortion. Overdispersion, scarce and highly skewed data (such as quakers or unusual processing techniques), and choosing a predictor set that balanced environmental and sensory elements without adding multicollinearity were among the difficulties. To assess predictor relevance and model performance, these were addressed through incremental development, diagnostics, and cautious model selection.

Overall, the method emphasises how crucial it is to preprocess, select suitable GLM families, and carry out exhaustive diagnostic checks in order to get accurate, comprehensible information about coffee quality and flaws.

2 Question 2

2.1 Q2(a) – Negative Binomial (2,) Probability Mass Function

For $k = 2$, the PMF is:

$$P(Y = y) = \binom{y + k - 1}{k - 1}(1 - \pi)^y \pi^k, \quad y = 0, 1, 2, \ldots$$

$$\binom{y + 1}{1} = y + 1$$

So the PMF becomes:

$$P(Y = y) = (y + 1)(1 - \pi)^y \pi^2, \quad y = 0, 1, 2, \ldots$$

2.2 Q2(b) – Exponential Family Form

The negative binomial can be written in the exponential family form as:

$$f(y; \pi) = \exp\left[y \log(1 - \pi) + 2\log(\pi) + \log(y + 1)\right]$$

The **canonical (natural) parameter** is $\eta = \log(1 - \pi)$.

2.3 Q2(c) – Mean and Variance

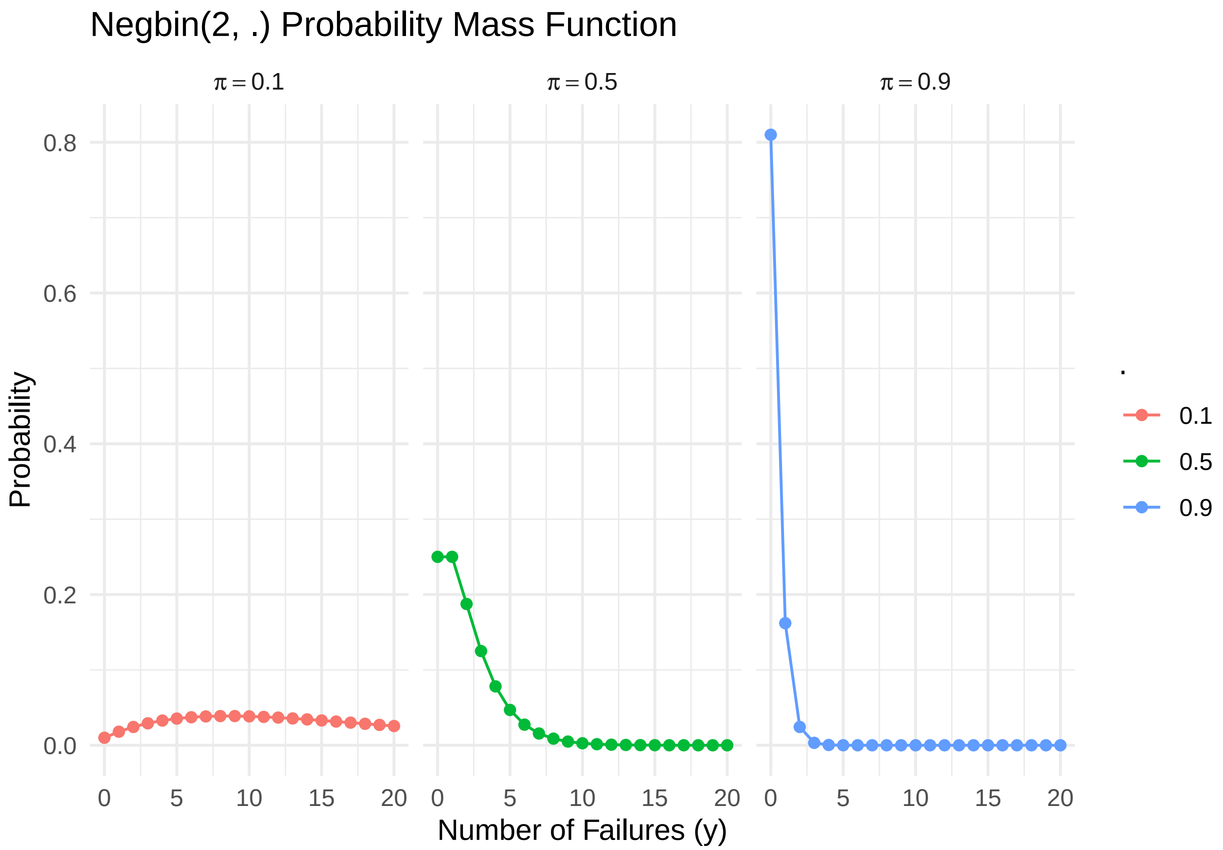
For the exponential family, the mean and variance can be derived as:

$$\mathrm{E}[Y] = \frac{2(1 - \pi)}{\pi}, \quad \mathrm{Var}[Y] = \frac{2(1 - \pi)}{\pi^2}$$

2.4 Q2(d) - Graph of Negbin(2,) probability function

```r
### Q2(d) - Graphing the PMF in R


library(ggplot2)
library(dplyr)

# Define function for Negbin(2, pi)
negbin2_pmf <- function(y, pi) {
  (y+1) * (1-pi)^y * pi^2
}

# Create data frame for plotting
y_vals <- 0:20
pi_vals <- c(0.1, 0.5, 0.9)

plot_data <- expand.grid(y = y_vals, pi = pi_vals) %>%
  mutate(prob = negbin2_pmf(y, pi))
```

```
# Plot
ggplot(plot_data, aes(x = y, y = prob, color = as.factor(pi))) +
  geom_point() +
  geom_line() +
  facet_wrap(~pi, labeller = label_bquote(pi == .(pi))) +
  labs(title = "Negbin(2,  ) Probability Mass Function",
       x = "Number of Failures (y)",
       y = "Probability",
       color = " ") +
  theme_minimal()
```


Negbin(2, .) Probability Mass Function
$\pi = 0.1$
$\pi = 0.5$
$\pi = 0.9$
Probability
Number of Failures (y)