Analysis of Coffee Quality Factors

Group-11

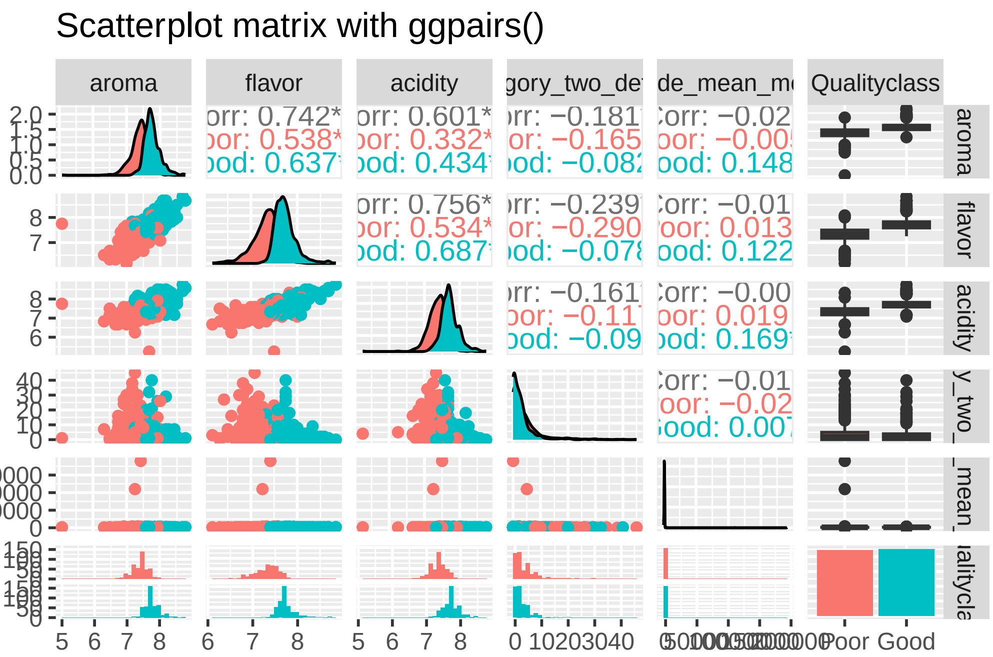
```
library(tidyverse)
library(moderndive)
library(gapminder)
library(sjPlot)
library(jtools)
library(GGally)
library(gt)
library(gridExtra)
library(knitr)
library(patchwork)
library(broom)
library(MASS)
library(janitor)
library(pscl)
library(ggfortify)
library(caret)
```

1 Data Wrangling

Preprocess the data and conduct summary statistics.

```
# Read the dataset
Data <- read.csv("dataset11.csv")
Data <- na.omit(Data)
Data$Qualityclass <- factor(Data$Qualityclass, levels = c("Poor", "Good"))
Data$country_of_origin <- factor(Data$country_of_origin)
Data$harvested <- factor(Data$harvested, levels = c(2012:2018,2010,2011))

# Scatterplot matrix with ggpairs()
```

```
scatterplot = Data %>%
  dplyr::select(aroma, flavor, acidity, category_two_defects, altitude_mean_meters, Qualit
ggpairs(scatterplot, aes(color = Qualityclass), title="Scatterplot matrix with ggpairs()")
```




```
# Remove outliers
q1_aroma <- quantile(Data$aroma, 0.25)
q3_aroma <- quantile(Data$aroma, 0.75)
iqr_aroma <- q3_aroma - q1_aroma
lower_bound_aroma <- q1_aroma - 1.5 * iqr_aroma
upper_bound_aroma <- q3_aroma + 1.5 * iqr_aroma
Data1 <- Data %>%
  filter(aroma >= lower_bound_aroma & aroma <= upper_bound_aroma)

q1_flavor <- quantile(Data1$flavor, 0.25)
q3_flavor <- quantile(Data1$flavor, 0.75)
iqr_flavor <- q3_flavor - q1_flavor
lower_bound_flavor <- q1_flavor - 1.5 * iqr_flavor
upper_bound_flavor <- q3_flavor + 1.5 * iqr_flavor
Data1 <- Data1 %>%
  filter(flavor >= lower_bound_flavor & flavor <= upper_bound_flavor)
```

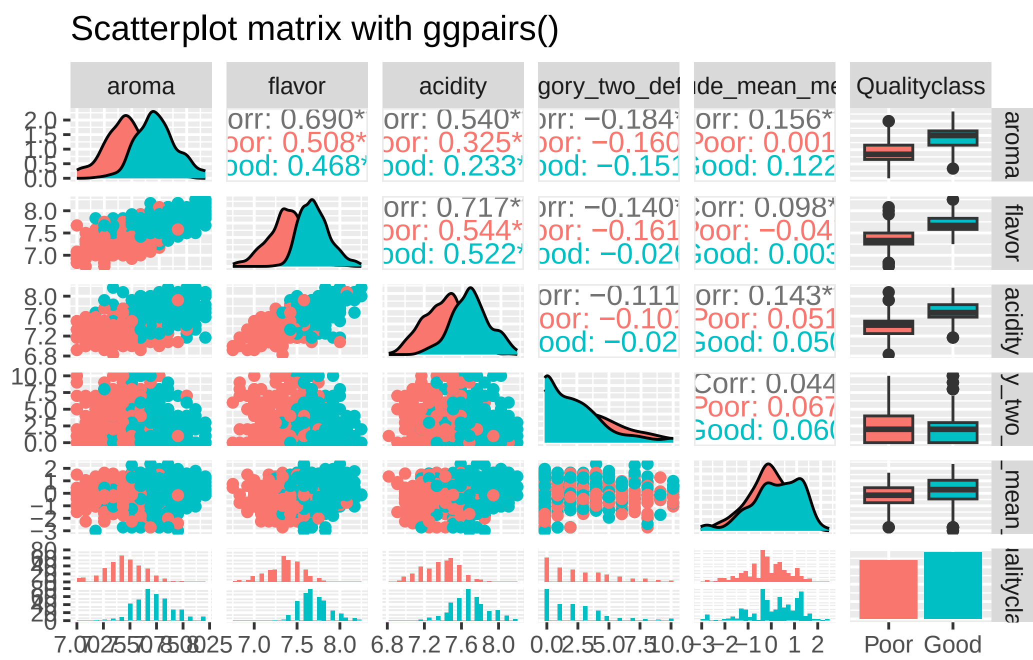
```r
q1_acidity <- quantile(Data1$acidity, 0.25)
q3_acidity <- quantile(Data1$acidity, 0.75)
iqr_acidity <- q3_acidity - q1_acidity
lower_bound_acidity <- q1_acidity - 1.5 * iqr_acidity
upper_bound_acidity <- q3_acidity+ 1.5 * iqr_acidity
Data1 <- Data1 %>%
  filter(acidity >= lower_bound_acidity & acidity <= upper_bound_acidity)

q1_defects <- quantile(Data1$category_two_defects, 0.25)
q3_defects <- quantile(Data1$category_two_defects, 0.75)
iqr_defects <- q3_defects - q1_defects
lower_bound_defects <- q1_defects - 1.5 * iqr_defects
upper_bound_defects <- q3_defects + 1.5 * iqr_defects
Data1 <- Data1 %>%
  filter(category_two_defects >= lower_bound_defects & category_two_defects <= upper_bound

q1_altitude <- quantile(Data1$altitude_mean_meters, 0.25)
q3_altitude <- quantile(Data1$altitude_mean_meters, 0.75)
iqr_altitude <- q3_altitude - q1_altitude
lower_bound_altitude <- q1_altitude - 1.5 * iqr_altitude
upper_bound_altitude <- q3_altitude+ 1.5 * iqr_altitude
data <- Data1 %>%
  filter(altitude_mean_meters >= lower_bound_altitude & altitude_mean_meters <= upper_boun
```

```r
# Standardize the 'altitude_mean_meters' column
mean_altitude <- mean(data$altitude_mean_meters)
sd_altitude <- sd(data$altitude_mean_meters)
data$altitude_mean_meters <- (data$altitude_mean_meters - mean_altitude) / sd_altitude
```

2 Data Visualization

Generate visualizations to better understand the data.

```r
# ggpairs of the wrangling data
scatterplot = data %>%
  dplyr::select(aroma, flavor, acidity, category_two_defects, altitude_mean_meters, Qualit
ggpairs(scatterplot, aes(color = Qualityclass), title="Scatterplot matrix with ggpairs()")
```




```
# Summary Statistics for 'aroma' and 'flavor' across different quality classes
data |>
  summarize('ar.Mean' = mean(aroma),
            'ar.Sd' = sd(aroma),
            'ar.Min' = min(aroma),
            'ar.Max' = max(aroma),
            'fl.Mean' = mean(flavor),
            'fl.Sd' = sd(flavor),
            'fl.Min' = min(flavor),
            'fl.Max' = max(flavor),
               .by = Qualityclass) |>
gt() |>
  fmt_number(decimals = 2) |>
  tab_spanner(
    label = "aroma",
    columns = c(ar.Mean, ar.Sd, ar.Min, ar.Max)
  ) |>
  tab_spanner(
    label = "flavor",
    columns = c(fl.Mean, fl.Sd, fl.Min, fl.Max)
  )
# Summary statistics for 'acidity' and 'category_two_defects' across different quality cla
```

```
data |>
  summarize('ac.Mean' = mean(acidity),
            'ac.Sd' = sd(acidity),
            'ac.Min' = min(acidity),
            'ac.Max' = max(acidity),
            'C.Mean' = mean(category_two_defects),
            'C.Sd' = sd(category_two_defects),
            'C.Min' = min(category_two_defects),
            'C.Max' = max(category_two_defects),
             .by = Qualityclass) |>
gt() |>
  fmt_number(decimals = 2) |>
  tab_spanner(
    label = "acidity",
    columns = c(ac.Mean, ac.Sd, ac.Min, ac.Max)
  ) |>
  tab_spanner(
    label = "Defects",
    columns = c(C.Mean, C.Sd, C.Min, C.Max)
  )
# Summary statistics for 'altitude_mean_meters' across different quality classes
data |>
  summarize('A.Mean' = mean(altitude_mean_meters),
            'A.Sd' = sd(altitude_mean_meters),
            'A.Min' = min(altitude_mean_meters),
            'A.Max' = max(altitude_mean_meters),
             .by = Qualityclass) |>
gt() |>
  fmt_number(decimals = 2) |>
  tab_spanner(
    label = "Altitude mean meters",
    columns = c(A.Mean, A.Sd, A.Min, A.Max)
  )
```

```
# Calculate the count of coffee bean qualities for each country
quality_counts <- data %>%
  group_by(country_of_origin, Qualityclass) %>%
  summarise(count = n()) %>%
  spread(Qualityclass, count, fill = 0) %>%
  mutate(proportion_good = Good / (Good + Poor))
```

Table 1: Summary statistics

(a)

	aroma				flavor			
Qualityclass	ar.Mean	ar.Sd	ar.Min	ar.Max	fl.Mean	fl.Sd	fl.Min	fl.Max
Poor	7.44	0.19	7.00	8.00	7.36	0.21	6.75	8.08
Good	7.73	0.18	7.17	8.17	7.71	0.17	7.25	8.25

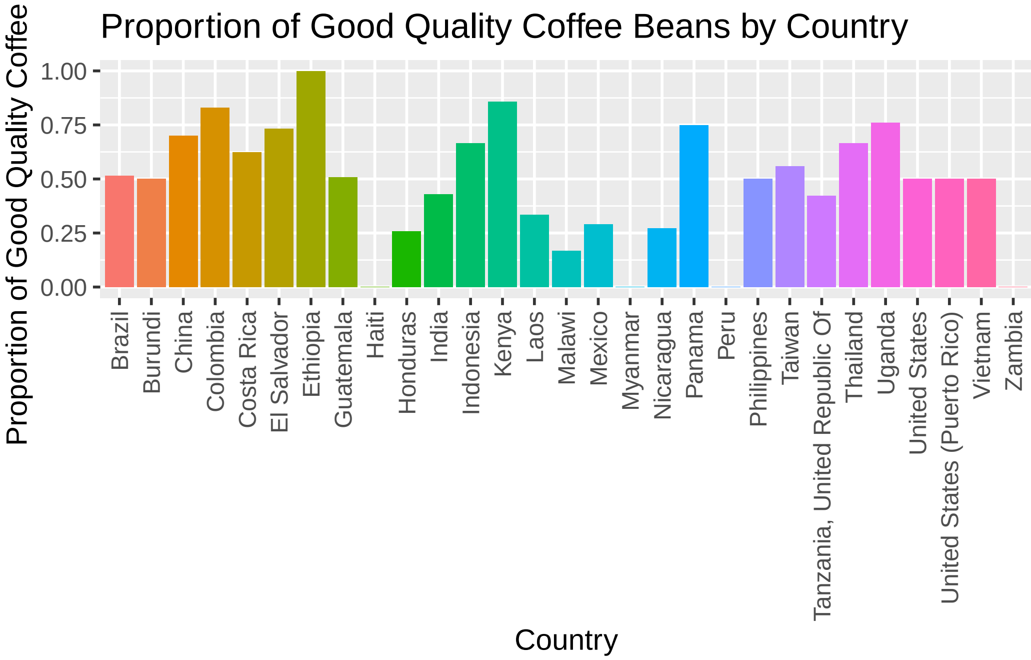
(b)

	acidity				Defects			
Qualityclass	ac.Mean	ac.Sd	ac.Min	ac.Max	C.Mean	C.Sd	C.Min	C.Max
Poor	7.38	0.20	6.83	8.08	2.75	2.64	0.00	10.00
Good	7.69	0.20	7.17	8.17	2.25	2.37	0.00	10.00

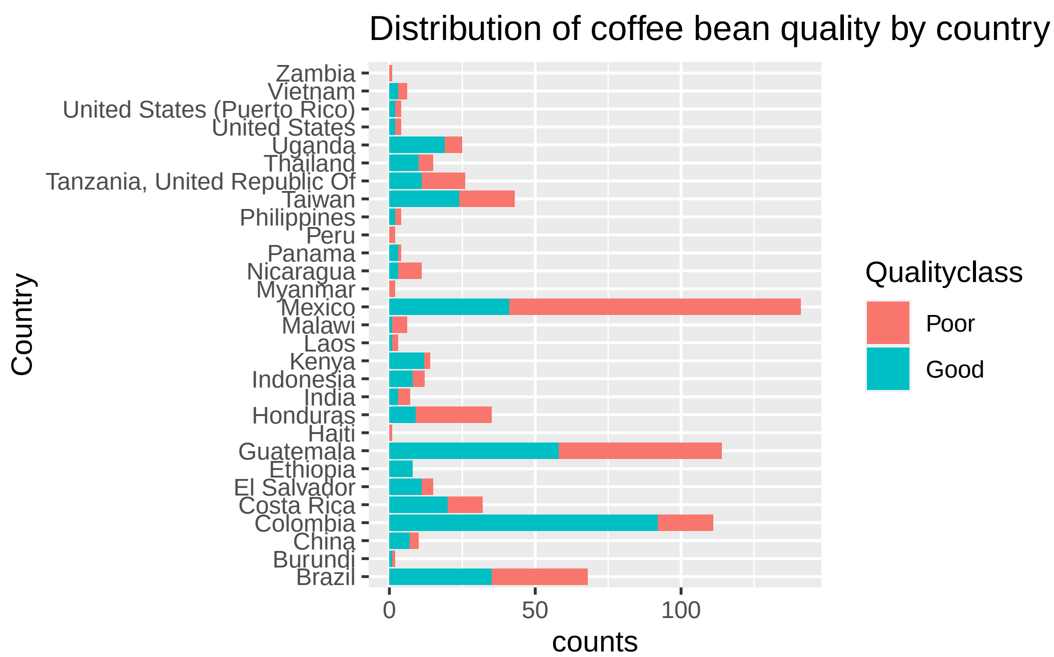
(c)

	Altitude mean meters			
Qualityclass	A.Mean	A.Sd	A.Min	A.Max
Poor	−0.18	0.91	−2.73	1.65
Good	0.16	1.05	−3.00	2.35

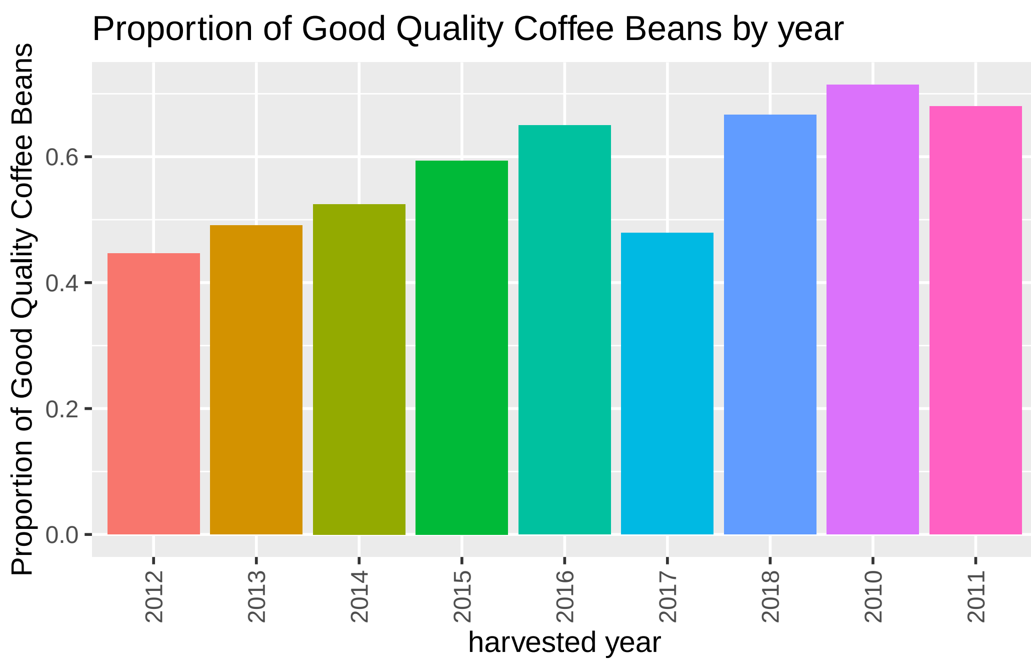
```r
# Create a bar plot showing the proportion of good quality coffee beans by country
ggplot(quality_counts, aes(x = country_of_origin, y = proportion_good, fill = country_of_o
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(x = "Country", y = "Proportion of Good Quality Coffee Beans",
       title = "Proportion of Good Quality Coffee Beans by Country") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```



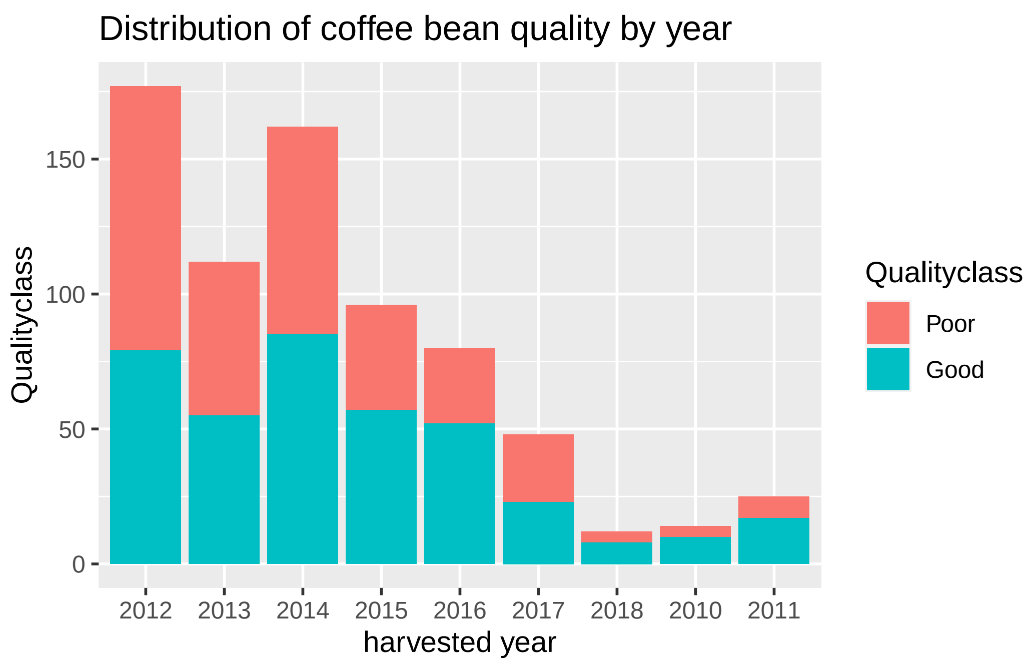

```r
# Create a bar plot to visualize the distribution of coffee bean quality by country
counts <- data %>%
  group_by(country_of_origin, Qualityclass) %>%
  summarise(count = n())
ggplot(data = counts, mapping = aes(x = country_of_origin, y = count, fill = Qualityclass)
  geom_col() +
  labs(x = "Country", y = "counts",
        title = "Distribution of coffee bean quality by country")+
  coord_flip()
```




```
# Create a bar plot showing the proportion of good quality coffee beans by year
quality_counts1 <- data %>%
  group_by(harvested, Qualityclass) %>%
  summarise(count = n()) %>%
  spread(Qualityclass, count, fill = 0) %>%
  mutate(proportion_good = Good / (Good + Poor))
ggplot(quality_counts1, aes(x =harvested, y = proportion_good, fill = harvested)) +
  geom_bar(stat = "identity",show.legend = FALSE) +
  labs(x = "harvested year", y = "Proportion of Good Quality Coffee Beans",
       title = "Proportion of Good Quality Coffee Beans by year") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
# Create a bar plot to visualize the distribution of coffee bean quality by year
counts1 <- data %>%
  group_by(harvested, Qualityclass) %>%
  summarise(count = n())
ggplot(data = counts1, mapping = aes(x = harvested, y = count, fill = Qualityclass)) +
  geom_col() +
  labs(x = "harvested year", y = "Qualityclass",
        title = "Distribution of coffee bean quality by year")
```



3 Exploratory Data Analysis

Modeling each predictor separately with the response variable to observe the individual impact of each feature on the quality of coffee.

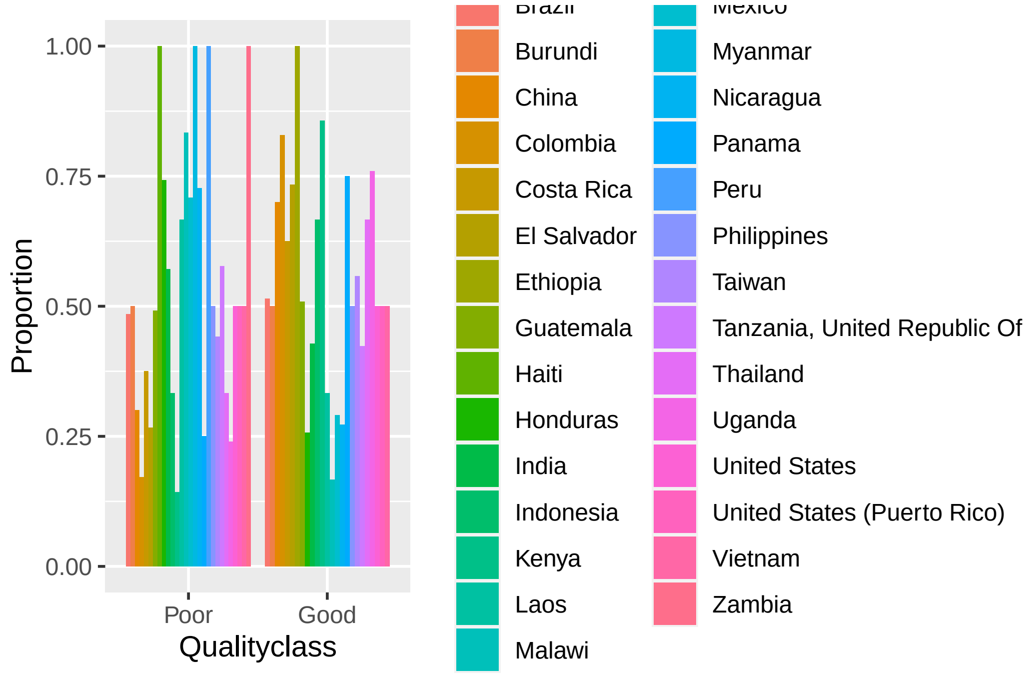
3.1 Country and Qualityclass

```
# Select 'country_of_origin' and 'Qualityclass' columns and generate a contingency table.
data_country <- data %>%
          dplyr::select(country_of_origin, Qualityclass)
data_country %>%
  tabyl(country_of_origin, Qualityclass) %>%
  adorn_percentages() %>%
  adorn_pct_formatting() %>%
  adorn_ns()
```

```
country_of_origin          Poor         Good
           Brazil  48.5%  (33)  51.5                     (35)
          Burundi  50.0%   (1)  50.0%  (1)
            China  30.0%   (3)  70.0%  (7)
```

```
                  Colombia  17.1%  (19)  82.9                      (92)
                Costa Rica  37.5%  (12)  62.5                      (20)
             Cote d?Ivoire      -   (0)     -   (0)
                   Ecuador      -   (0)     -   (0)
               El Salvador  26.7%   (4)  73.3                      (11)
                  Ethiopia   0.0%   (0) 100.0%  (8)
                 Guatemala  49.1%  (56)  50.9                      (58)
                     Haiti 100.0%   (1)   0.0%  (0)
                  Honduras  74.3%  (26)  25.7%  (9)
                     India  57.1%   (4)  42.9%  (3)
                 Indonesia  33.3%   (4)  66.7%  (8)
                     Kenya  14.3%   (2)  85.7                      (12)
                      Laos  66.7%   (2)  33.3%  (1)
                    Malawi  83.3%   (5)  16.7%  (1)
                 Mauritius      -   (0)     -   (0)
                    Mexico  70.9% (100)  29.1                      (41)
                   Myanmar 100.0%   (2)   0.0%  (0)
                 Nicaragua  72.7%   (8)  27.3%  (3)
                    Panama  25.0%   (1)  75.0%  (3)
          Papua New Guinea      -   (0)     -   (0)
                      Peru 100.0%   (2)   0.0%  (0)
               Philippines  50.0%   (2)  50.0%  (2)
                    Taiwan  44.2%  (19)  55.8                      (24)
Tanzania, United Republic Of  57.7%  (15)  42.3                    (11)
                  Thailand  33.3%   (5)  66.7                      (10)
                    Uganda  24.0%   (6)  76.0                      (19)
             United States  50.0%   (2)  50.0%  (2)
    United States (Hawaii)      -   (0)     -   (0)
United States (Puerto Rico)  50.0%   (2)  50.0%  (2)
                   Vietnam  50.0%   (3)  50.0%  (3)
                    Zambia 100.0%   (1)   0.0%  (0)
```

```r
# Create a barplot of 'country_of_origin' across different 'Qualityclass' levels
p0 <- ggplot(data_country, aes(x = Qualityclass, y = after_stat(prop), group = country_of_
    geom_bar(position = "dodge", stat = "count") +
    labs(y = "Proportion")
p0
```



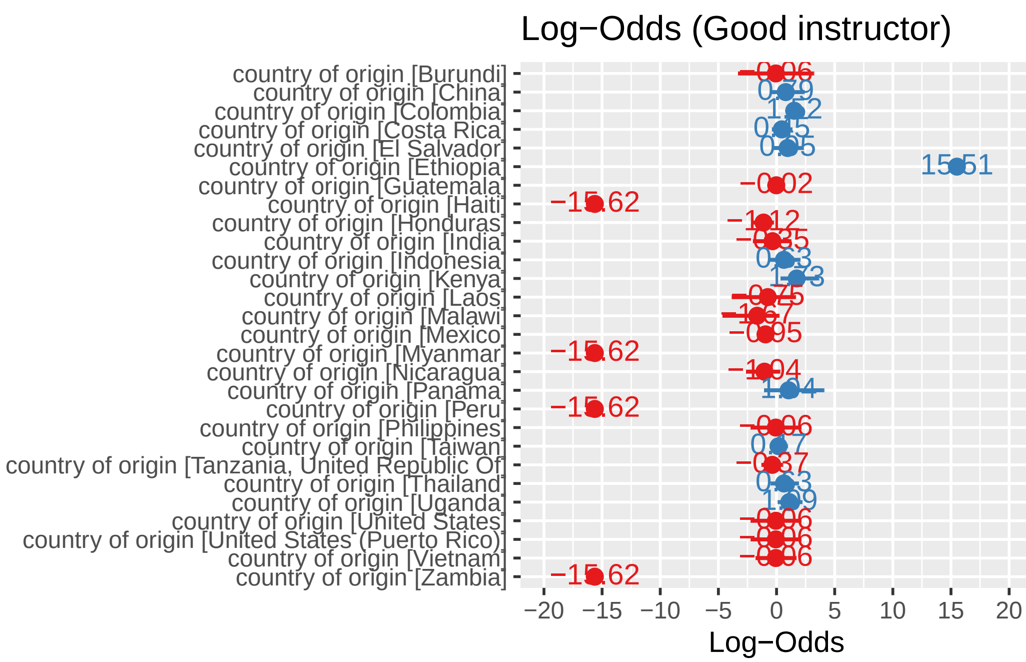

```
# Fit logistic regression model with 'country_of_origin' predictor and 'Qualityclass' resp
model_country <- glm(Qualityclass ~ country_of_origin, data = data_country, family = binom
model_country %>%
  summary()
```

```
Call:
glm(formula = Qualityclass ~ country_of_origin, family = binomial(link = "logit"),
    data = data_country)

Coefficients:
                                                 Estimate Std. Error z value
(Intercept)                                       0.05884    0.24264   0.243
country_of_originBurundi                         -0.05884    1.43488  -0.041
country_of_originChina                            0.78846    0.73148   1.078
country_of_originColombia                         1.51851    0.34982   4.341
country_of_originCosta Rica                       0.45199    0.43842   1.031
country_of_originEl Salvador                      0.95276    0.63228   1.507
country_of_originEthiopia                        15.50723  514.56079   0.030
country_of_originGuatemala                       -0.02375    0.30655  -0.077
country_of_originHaiti                          -15.62491 1455.39755  -0.011
country_of_originHonduras                        -1.11971    0.45656  -2.452
```

```
country_of_originIndia                              -0.34652    0.80138  -0.432
country_of_originIndonesia                           0.63431    0.65869   0.963
country_of_originKenya                               1.73292    0.80138   2.162
country_of_originLaos                               -0.75199    1.24855  -0.602
country_of_originMalawi                             -1.66828    1.12200  -1.487
country_of_originMexico                             -0.95044    0.30539  -3.112
country_of_originMyanmar                           -15.62491 1029.12149  -0.015
country_of_originNicaragua                          -1.03967    0.71917  -1.446
country_of_originPanama                              1.03977    1.17992   0.881
country_of_originPeru                              -15.62491 1029.12149  -0.015
country_of_originPhilippines                        -0.05884    1.02902  -0.057
country_of_originTaiwan                              0.17477    0.39137   0.447
country_of_originTanzania, United Republic Of       -0.36900    0.46524  -0.793
country_of_originThailand                            0.63431    0.59906   1.059
country_of_originUganda                              1.09384    0.52742   2.074
country_of_originUnited States                      -0.05884    1.02902  -0.057
country_of_originUnited States (Puerto Rico)        -0.05884    1.02902  -0.057
country_of_originVietnam                            -0.05884    0.85179  -0.069
country_of_originZambia                            -15.62491 1455.39755  -0.011
                                              Pr(>|z|)
(Intercept)                                    0.80839
country_of_originBurundi                       0.96729
country_of_originChina                         0.28108
country_of_originColombia                     1.42e-05 ***
country_of_originCosta Rica                    0.30256
country_of_originEl Salvador                   0.13185
country_of_originEthiopia                      0.97596
country_of_originGuatemala                     0.93825
country_of_originHaiti                         0.99143
country_of_originHonduras                      0.01419 *
country_of_originIndia                         0.66544
country_of_originIndonesia                     0.33556
country_of_originKenya                         0.03059 *
country_of_originLaos                          0.54698
country_of_originMalawi                        0.13705
country_of_originMexico                        0.00186 **
country_of_originMyanmar                       0.98789
country_of_originNicaragua                     0.14828
country_of_originPanama                        0.37820
country_of_originPeru                          0.98789
country_of_originPhilippines                   0.95440
country_of_originTaiwan                        0.65519
country_of_originTanzania, United Republic Of  0.42770
```

```
country_of_originThailand                            0.28968
country_of_originUganda                              0.03808 *
country_of_originUnited States                       0.95440
country_of_originUnited States (Puerto Rico)         0.95440
country_of_originVietnam                             0.94493
country_of_originZambia                              0.99143
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  867.46  on 697  degrees of freedom
AIC: 925.46

Number of Fisher Scoring iterations: 14
```

```
# Extract coefficients from the model and calculate their confidence intervals.
model_country_coef_logodds <- model_country %>%
  summary() %>%
  coef()
confint_logodds <- confint(model_country)

# Plot log-odds of being a good instructor
plot_model(model_country, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```

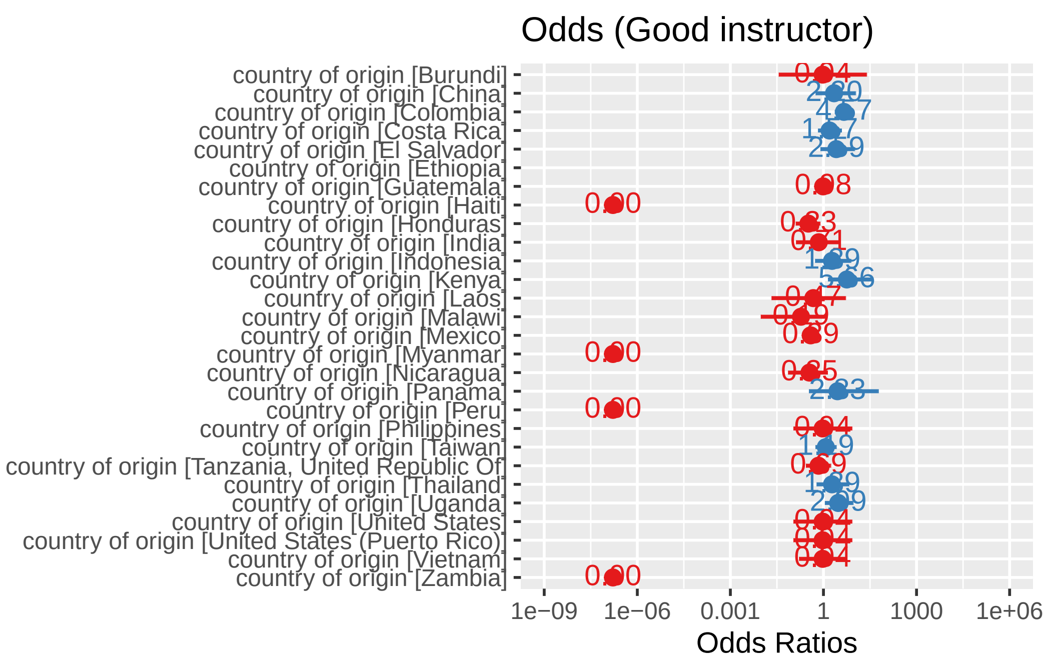



```r
# Transform the coefficients into odds ratios and obtain their confidence intervals
model_country_coef_odds <- model_country %>%
  summary() %>%
  coef() %>%
  exp()
exp(confint_logodds)
```

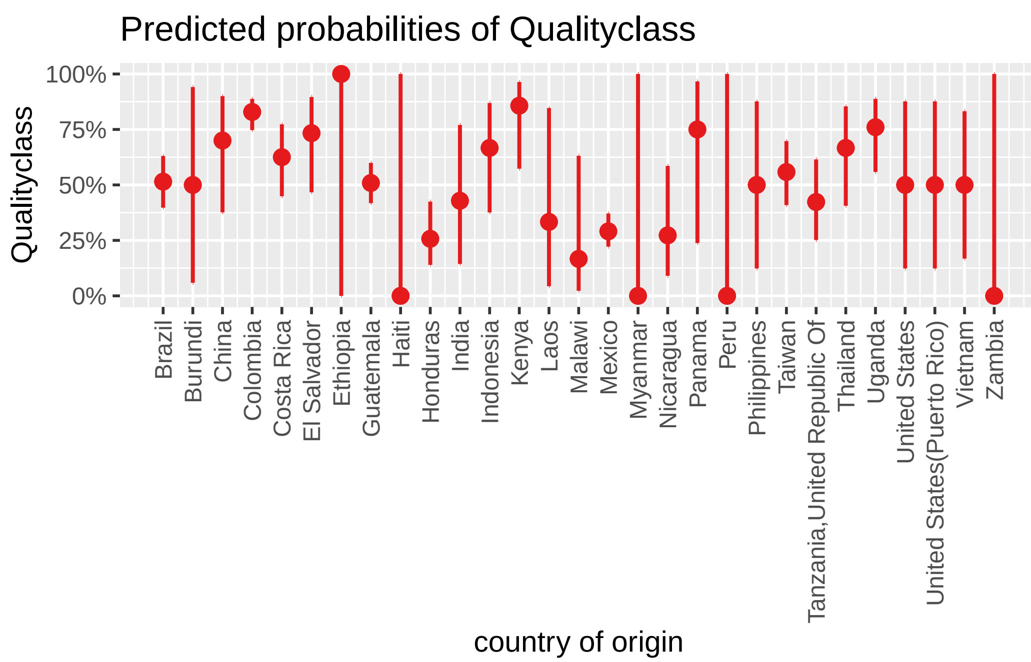
	2.5 %	97.5 %
(Intercept)	6.584510e-01	1.712201e+00
country_of_originBurundi	3.628866e-02	2.449674e+01
country_of_originChina	5.603861e-01	1.086958e+01
country_of_originColombia	2.325403e+00	9.207188e+00
country_of_originCosta Rica	6.718600e-01	3.785776e+00
country_of_originEl Salvador	7.985057e-01	1.009495e+01
country_of_originEthiopia	7.095795e-13	NA
country_of_originGuatemala	5.344645e-01	1.782260e+00
country_of_originHaiti	NA	1.019032e+122
country_of_originHonduras	1.279274e-01	7.772380e-01
country_of_originIndia	1.310178e-01	3.439276e+00
country_of_originIndonesia	5.399994e-01	7.613941e+00
country_of_originKenya	1.405194e+00	3.814342e+01
country_of_originLaos	2.131422e-02	5.144914e+00

```
country_of_originMalawi                            9.572120e-03  1.250123e+00
country_of_originMexico                            2.112932e-01  7.014921e-01
country_of_originMyanmar                                     NA  2.176427e+63
country_of_originNicaragua                         7.270571e-02  1.338790e+00
country_of_originPanama                            3.427609e-01  5.874916e+01
country_of_originPeru                                        NA  2.176427e+63
country_of_originPhilippines                       1.080352e-01  8.227934e+00
country_of_originTaiwan                            5.536918e-01  2.582126e+00
country_of_originTanzania, United Republic Of 2.727000e-01  1.711951e+00
country_of_originThailand                          6.027834e-01  6.593720e+00
country_of_originUganda                            1.110455e+00  9.036884e+00
country_of_originUnited States                     1.080352e-01  8.227934e+00
country_of_originUnited States (Puerto Rico)  1.080352e-01  8.227934e+00
country_of_originVietnam                           1.643311e-01  5.408930e+00
country_of_originZambia                                      NA 1.019032e+122
```

```r
# Plot odds of being a good instructor
plot_model(model_country, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```



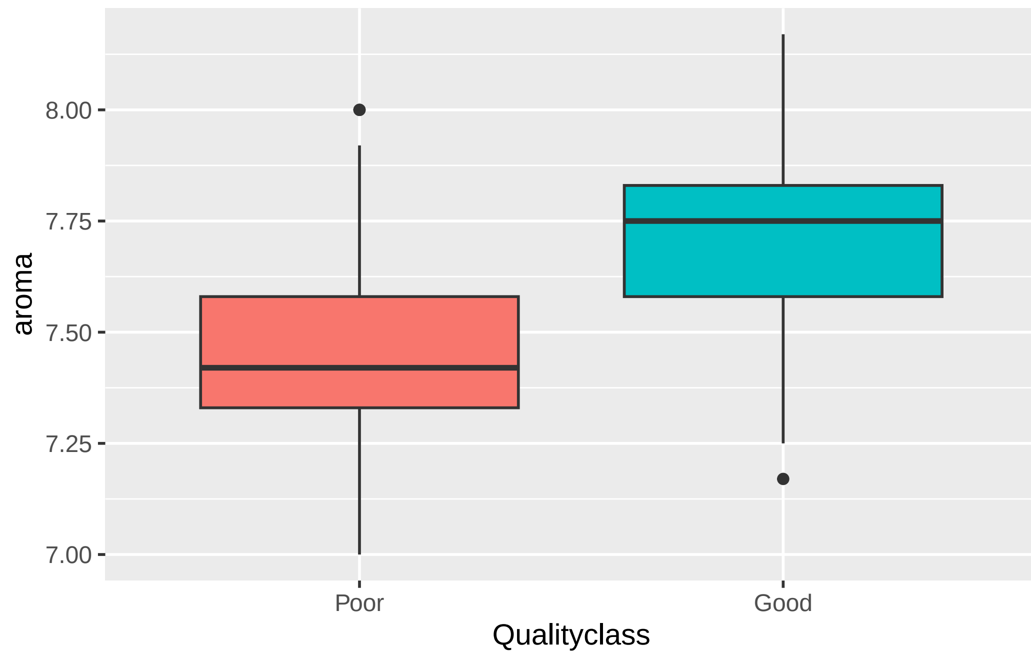
```
# Calculate log odds, odds, and probabilities and store them
data_country_after <- data_country %>%
            mutate(logodds.Good = predict(model_country, type = "response")) %>%
            mutate(odds.Good = exp(logodds.Good)) %>%
            mutate(probs.Good = fitted(model_country))

# Generate a predictive plot
plot_model(model_country, type = "pred",
           terms = c("country_of_origin"))+theme(axis.text.x = element_text(angle = 90, vj
```



3.2 Aroma and Qualityclass

```
# Select 'aroma' and 'Qualityclass' columns
data_aroma <- data %>%
            dplyr::select(aroma, Qualityclass)

# Create a boxplot of 'aroma' across different 'Qualityclass' levels
p1 <- ggplot(data = data_aroma, aes(x = Qualityclass, y = aroma, fill = Qualityclass)) +
  geom_boxplot() +
  labs(x = "Qualityclass", y = "aroma")+
```

```r
  theme(legend.position = "none")
p1
```




```r
# Fit logistic regression model with 'aroma' predictor and 'Qualityclass' response
model1 <- glm(Qualityclass ~ aroma, data = data_aroma,
              family = binomial(link = "logit"))
model1 %>%
  summary()
```

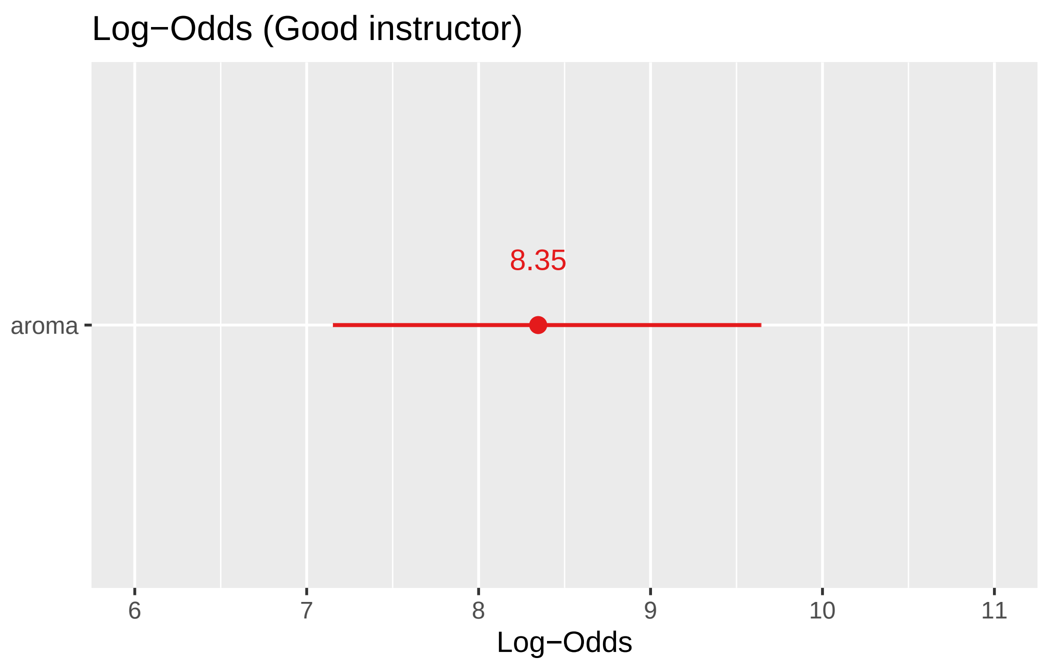
```

Call:
glm(formula = Qualityclass ~ aroma, family = binomial(link = "logit"),
    data = data_aroma)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -63.1944     4.8098  -13.14   <2e-16 ***
aroma         8.3465     0.6342   13.16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  676.23  on 724  degrees of freedom
AIC: 680.23

Number of Fisher Scoring iterations: 5
```

```r
# Calculate lower and upper bounds for 'aroma' log-odds
mod1.coef.logodds <- model1 %>%
                      summary() %>%
                      coef()
aroma.logodds.lower <- mod1.coef.logodds["aroma", "Estimate"] -
                      1.96 * mod1.coef.logodds["aroma", "Std. Error"]
aroma.logodds.upper <- mod1.coef.logodds["aroma", "Estimate"] +
                      1.96 * mod1.coef.logodds["aroma", "Std. Error"]

# Display the confidence interval
paste("(", aroma.logodds.lower, ",", aroma.logodds.upper, ")")
```

```
[1] "( 7.10355459660584 , 9.58952426960585 )"
```

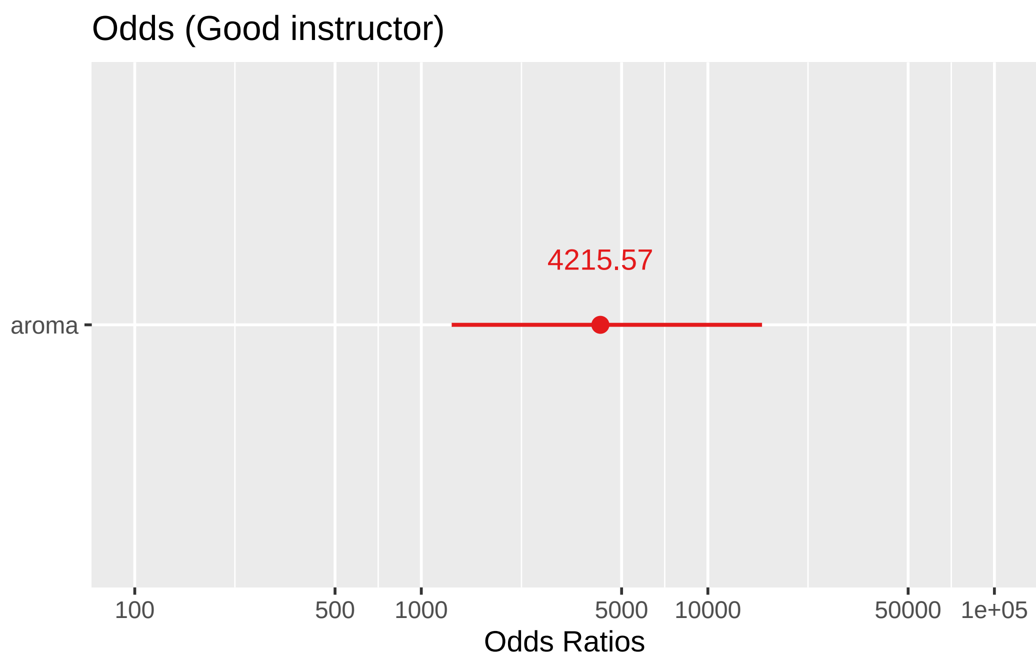
```r
# Plot log-odds of being a good instructor
plot_model(model1, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```



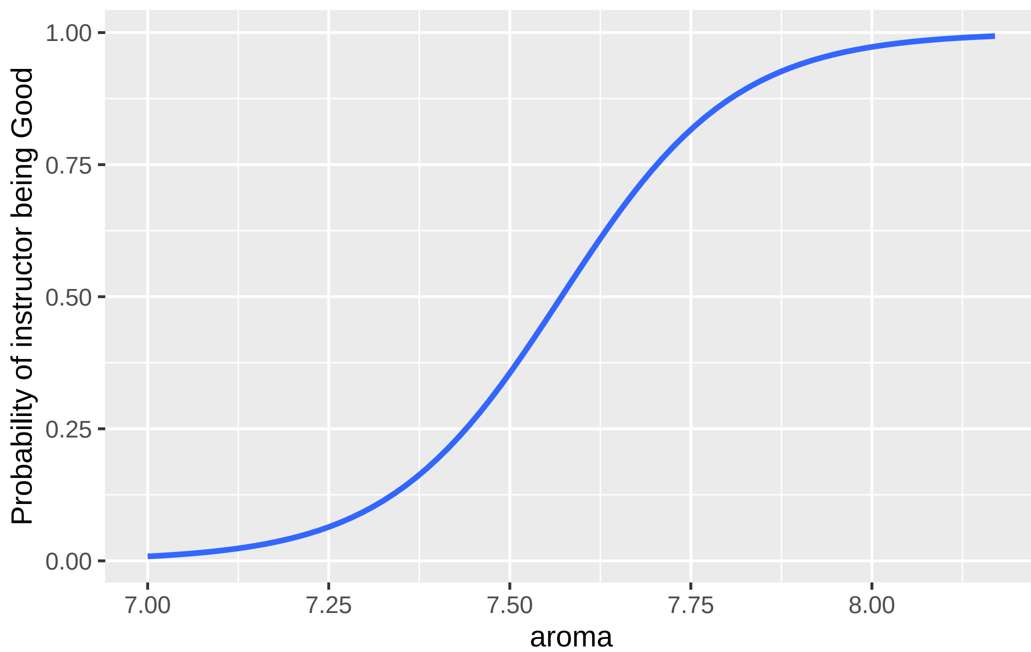

```
# Calculate lower and upper bounds for 'aroma' odds
aroma.odds.lower <- exp(aroma.logodds.lower)
aroma.odds.upper <- exp(aroma.logodds.upper)

# Display the confidence interval
paste("(", aroma.odds.lower, ",", aroma.odds.upper, ")")
```

```
[1] "( 1216.28279431934 , 14610.9170233989 )"
```

```
# Plot odds of being a good instructor
plot_model(model1, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```

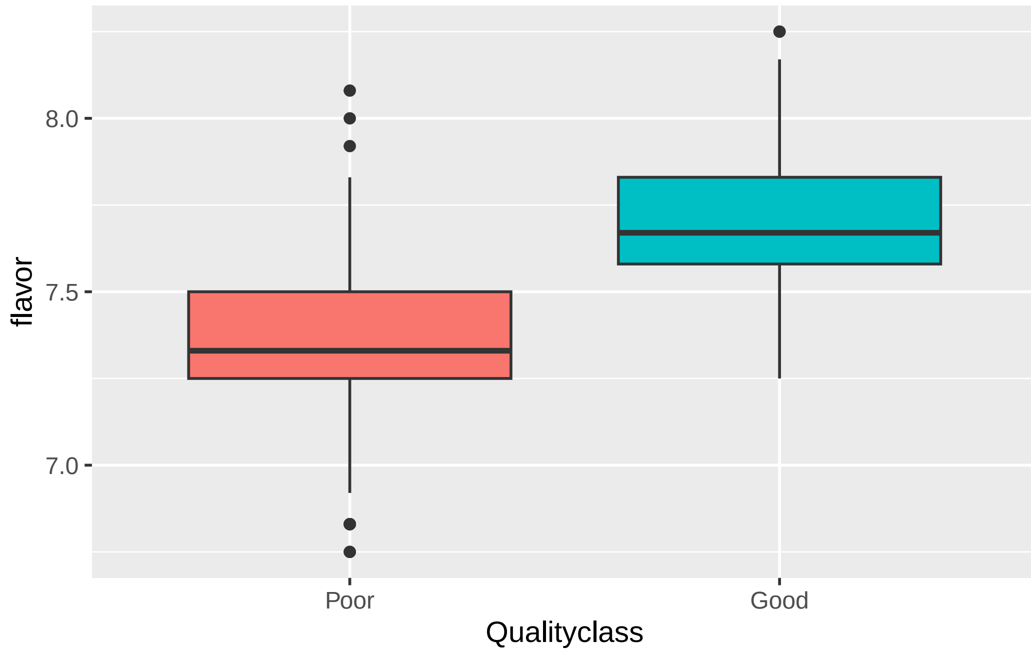



```r
# Add predicted probabilities
data_aroma_after <- data_aroma %>%
                  mutate(logodds.Good = predict(model1, type = "response")) %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model1))

# Plot the relationship between 'aroma' and probability of being a good instructor
ggplot(data = data_aroma_after, aes(x = aroma, y = probs.Good)) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
  labs(x = "aroma", y = "Probability of instructor being Good")
```



3.3 Flavor and Qualityclass

```
# Select 'flavor' and 'Qualityclass' columns
data_flavor <- data %>%
            dplyr::select(flavor, Qualityclass)

# Create a boxplot of 'flavor' across different 'Qualityclass' levels
p2 <- ggplot(data = data_flavor, aes(x = Qualityclass, y = flavor, fill = Qualityclass)) +
  geom_boxplot() +
  labs(x = "Qualityclass", y = "flavor")+
  theme(legend.position = "none")
p2
```




```r
# Fit logistic regression model with 'flavor' predictor and 'Qualityclass' response
model2 <- glm(Qualityclass ~ flavor, data = data_flavor,
              family = binomial(link = "logit"))
model2 %>%
  summary()
```

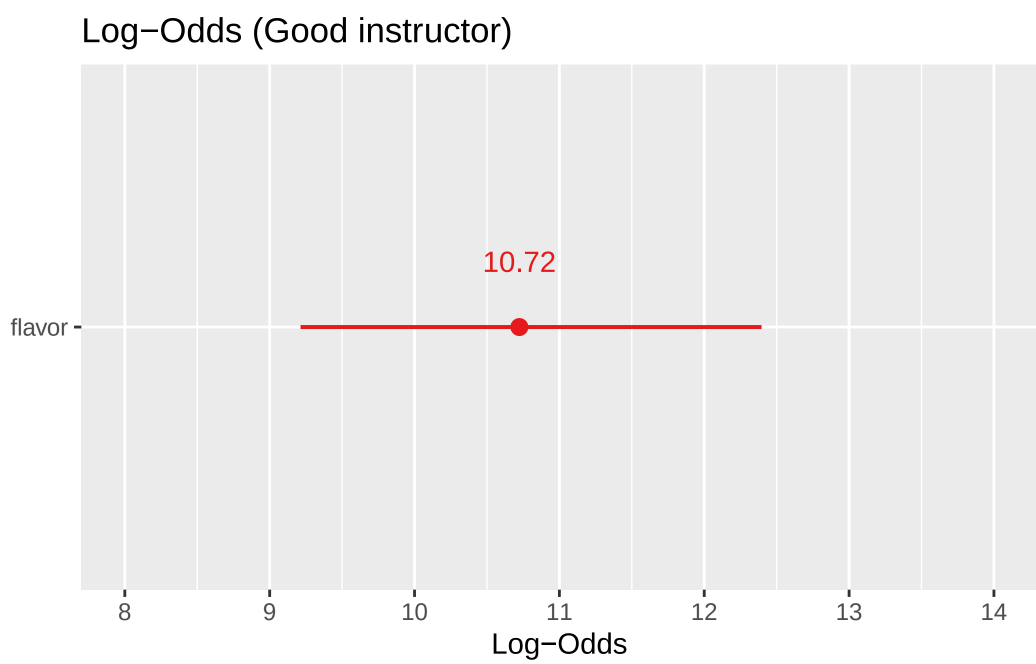
```

Call:
glm(formula = Qualityclass ~ flavor, family = binomial(link = "logit"),
    data = data_flavor)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -80.7455     6.1070  -13.22   <2e-16 ***
flavor       10.7238     0.8097   13.24   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  563.98  on 724  degrees of freedom
```

```
AIC: 567.98

Number of Fisher Scoring iterations: 6
```

```
# Calculate lower and upper bounds for 'flavor' log-odds
mod2.coef.logodds <- model2 %>%
                      summary() %>%
                      coef()
flavor.logodds.lower <- mod2.coef.logodds["flavor", "Estimate"] -
                      1.96 * mod2.coef.logodds["flavor", "Std. Error"]
flavor.logodds.upper <- mod2.coef.logodds["flavor", "Estimate"] +
                      1.96 * mod2.coef.logodds["flavor", "Std. Error"]

# Display the confidence interval
paste("(", flavor.logodds.lower, ",", flavor.logodds.upper, ")")
```

```
[1] "( 9.1369267163384 , 12.3107616668262 )"
```

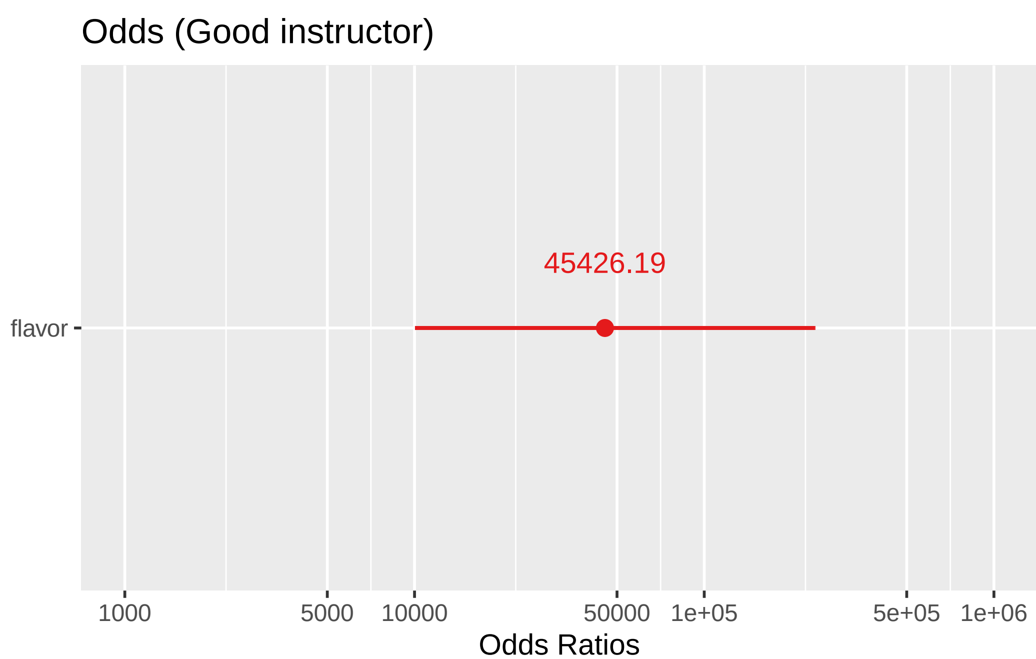
```
# Plot log-odds of being a good instructor
plot_model(model2, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```



```r
# Calculate lower and upper bounds for 'flavor' odds
flavor.odds.lower <- exp(flavor.logodds.lower)
flavor.odds.upper <- exp(flavor.logodds.upper)

# Display the confidence interval
paste("(", flavor.odds.lower, ",", flavor.odds.upper, ")")
```

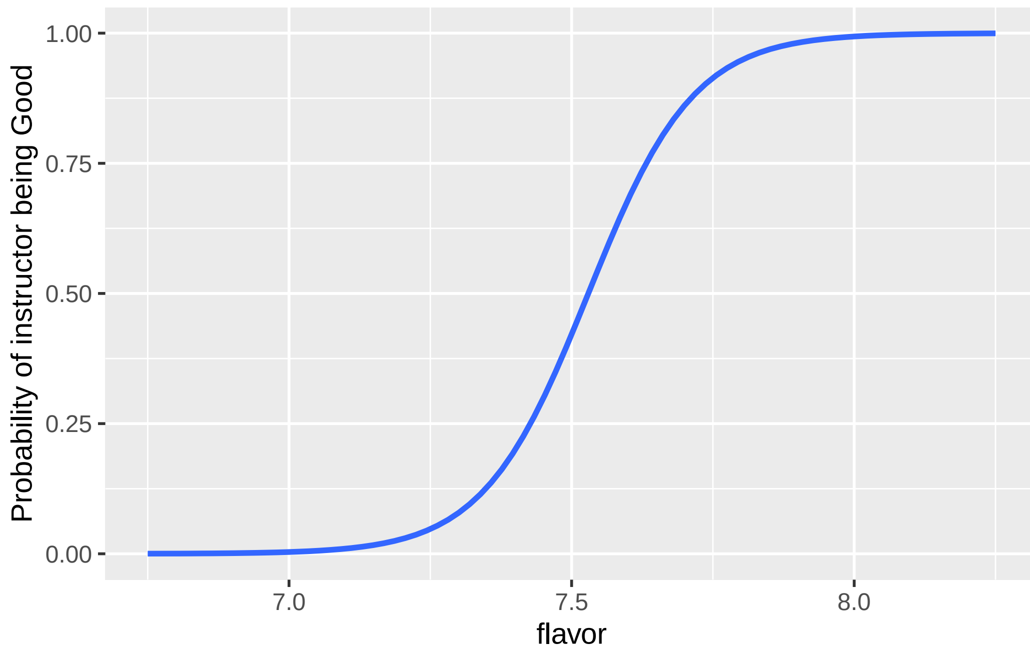
```
[1] "( 9292.16374923055 , 222073.051342241 )"
```

```r
# Plot odds of being a good instructor
plot_model(model2, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```



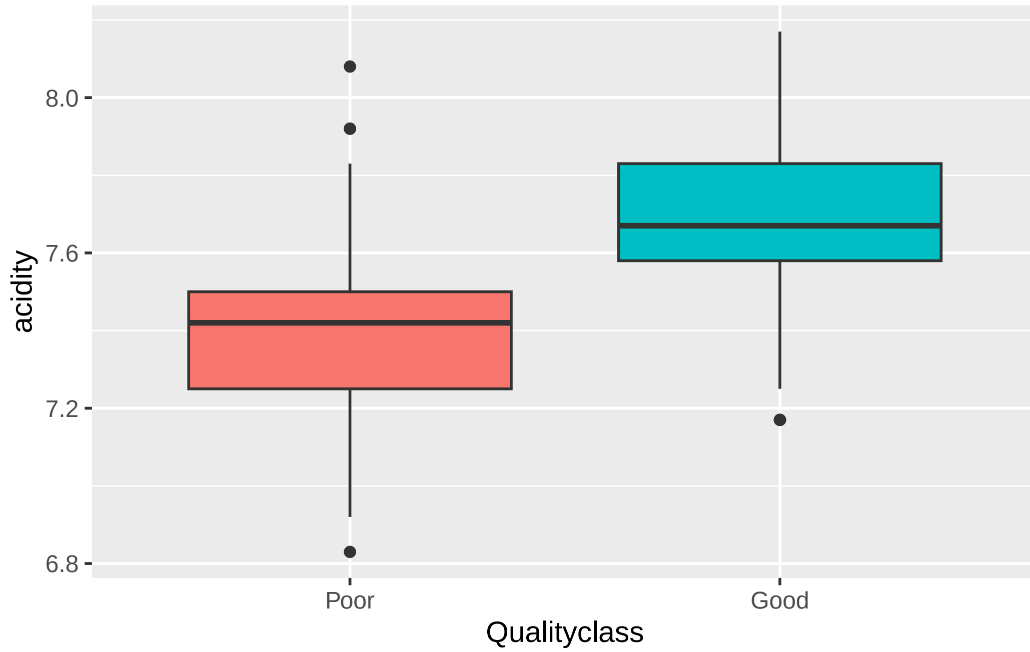

```r
# Add predicted probabilities
data_flavor_after <- data_flavor %>%
                  mutate(logodds.Good = predict(model2, type = "response")) %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model2))

# Plot the relationship between 'flavor' and probability of being a good instructor
```

```r
ggplot(data = data_flavor_after, aes(x = flavor, y = probs.Good)) +
  geom_smooth(method="glm",
              method.args = list(family="binomial"),
              se = FALSE) +
  labs(x = "flavor", y = "Probability of instructor being Good")
```



3.4 Acidity and Qualityclass

```r
# Select 'acidity' and 'Qualityclass' columns
data_acidity <- data %>%
            dplyr::select(acidity, Qualityclass)

# Create a boxplot of 'acidity' across different 'Qualityclass' levels
p3 <- ggplot(data = data_acidity, aes(x = Qualityclass, y = acidity, fill = Qualityclass))
  geom_boxplot() +
  labs(x = "Qualityclass", y = "acidity")+
  theme(legend.position = "none")
p3
```




```
# Fit logistic regression model with 'acidity' predictor and 'Qualityclass' response
model3 <- glm(Qualityclass ~ acidity, data = data_acidity,
              family = binomial(link = "logit"))
model3 %>%
  summary()
```

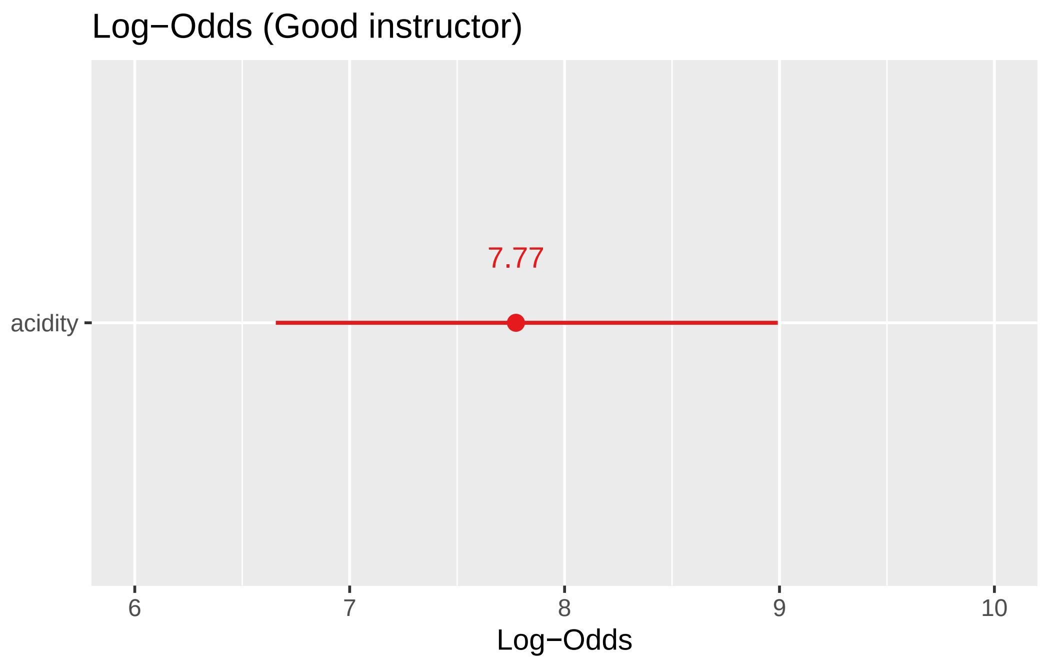
```
Call:
glm(formula = Qualityclass ~ acidity, family = binomial(link = "logit"),
    data = data_acidity)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -58.4727     4.4807  -13.05   <2e-16 ***
acidity       7.7736     0.5945   13.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.5  on 725  degrees of freedom
Residual deviance:  675.4  on 724  degrees of freedom
```

```
AIC: 679.4

Number of Fisher Scoring iterations: 5
```

```
# Calculate lower and upper bounds for 'acidity' log-odds
mod3.coef.logodds <- model3 %>%
                     summary() %>%
                     coef()
acidity.logodds.lower <- mod3.coef.logodds["acidity", "Estimate"] -
                     1.96 * mod3.coef.logodds["acidity", "Std. Error"]
acidity.logodds.upper <- mod3.coef.logodds["acidity", "Estimate"] +
                     1.96 * mod3.coef.logodds["acidity", "Std. Error"]

# Display the confidence interval
paste("(", acidity.logodds.lower, ",", acidity.logodds.upper, ")")
```

```
[1] "( 6.60847256990022 , 8.93873878119733 )"
```

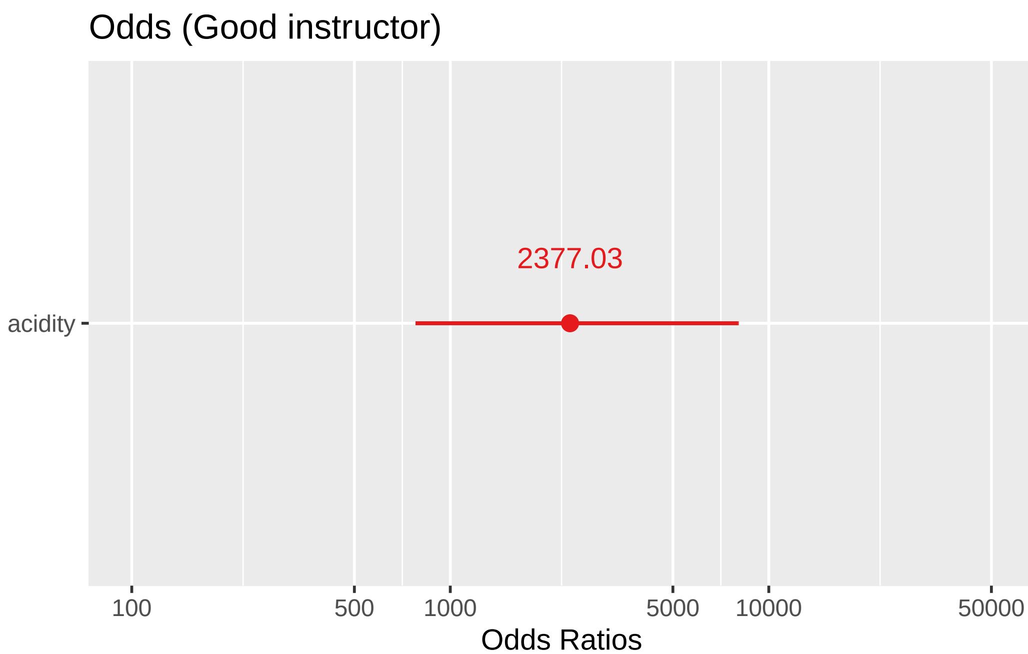
```
# Plot log-odds of being a good instructor
plot_model(model3, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```



```r
# Calculate lower and upper bounds for 'acidity' odds
acidity.odds.lower <- exp(acidity.logodds.lower)
acidity.odds.upper <- exp(acidity.logodds.upper)

# Display the confidence interval
paste("(", acidity.odds.lower, ",", acidity.odds.upper, ")")
```

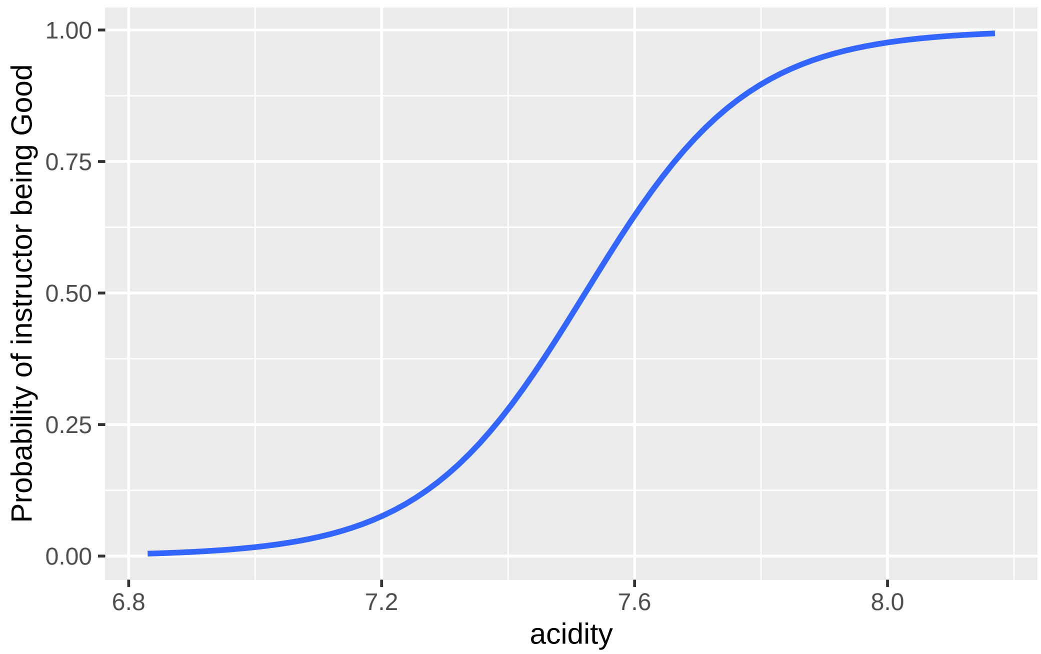
```
[1] "( 741.349793486747 , 7621.57851325187 )"
```

```r
# Plot odds of being a good instructor
plot_model(model3, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```



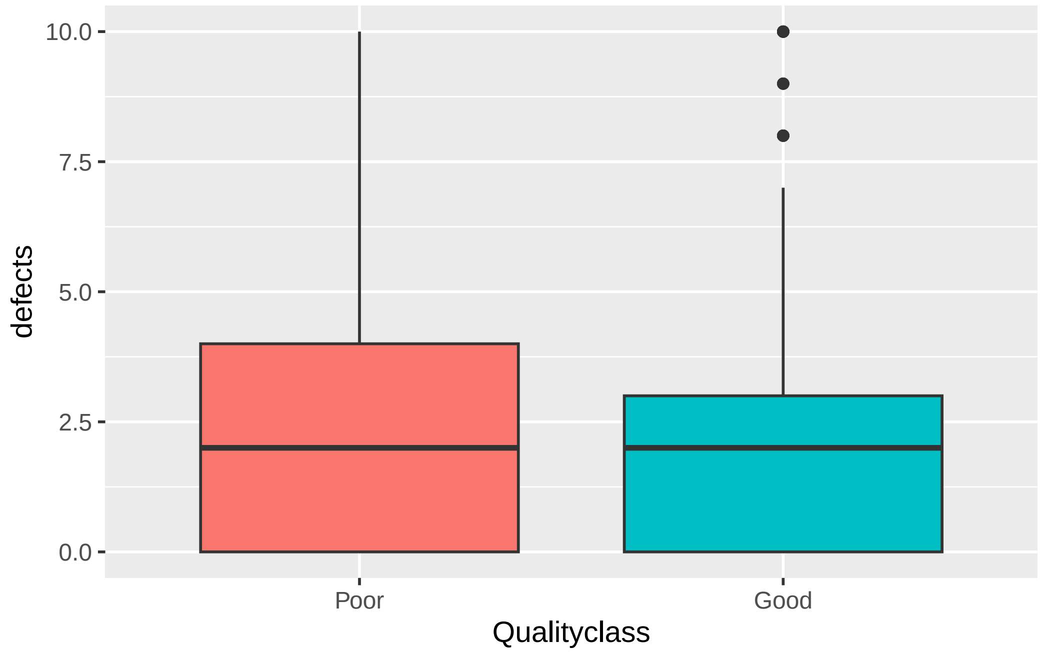

```r
# Add predicted probabilities
data_acidity_after <- data_acidity %>%
                  mutate(logodds.Good = predict(model3, type = "response")) %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model3))

# Plot the relationship between 'acidity' and probability of being a good instructor
```

```r
ggplot(data = data_acidity_after, aes(x = acidity, y = probs.Good)) +
  geom_smooth(method ="glm",
              method.args = list(family = "binomial"),
              se = FALSE) +
  labs(x = "acidity", y = "Probability of instructor being Good")
```



3.5 Category 2 type defects and Qualityclass

```r
# Select 'category_two_defects' and 'Qualityclass' columns
data_defects <- data %>%
              dplyr::select(category_two_defects, Qualityclass)

# Create a boxplot of 'category_two_defects' across different 'Qualityclass' levels
p4 <- ggplot(data = data_defects, aes(x = Qualityclass, y = category_two_defects, fill = Q
  geom_boxplot() +
  labs(x = "Qualityclass", y = "defects")+
  theme(legend.position = "none")
p4
```




```r
# Fit logistic regression model with 'category_two_defects' predictor and 'Qualityclass' r
model5 <- glm(Qualityclass ~ category_two_defects, data = data_defects,
             family = binomial(link = "logit"))
model5 %>%
  summary()
```

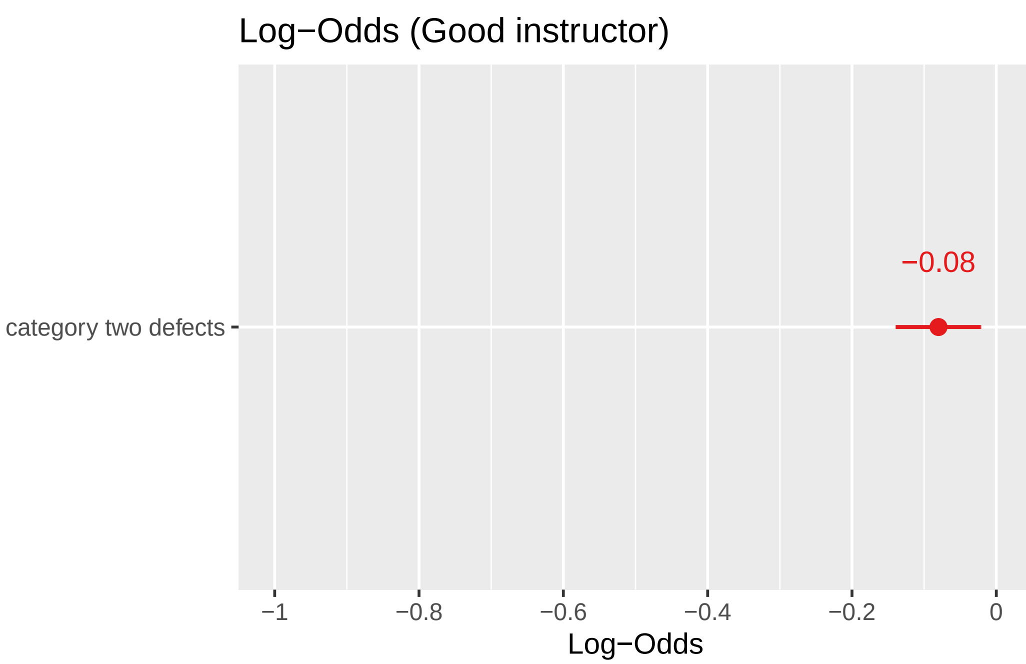
```

Call:
glm(formula = Qualityclass ~ category_two_defects, family = binomial(link = "logit"),
    data = data_defects)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)           0.32616    0.10558   3.089  0.00201 **
category_two_defects -0.08010    0.02999  -2.671  0.00757 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  996.31  on 724  degrees of freedom
```

```
AIC: 1000.3

Number of Fisher Scoring iterations: 4
```

```r
# Calculate lower and upper bounds for 'category_two_defects' log-odds
mod5.coef.logodds <- model5 %>%
                      summary() %>%
                      coef()
defects.logodds.lower <- mod5.coef.logodds["category_two_defects", "Estimate"] -
                      1.96 * mod5.coef.logodds["category_two_defects", "Std. Error"]
defects.logodds.upper <- mod5.coef.logodds["category_two_defects", "Estimate"] +
                      1.96 * mod5.coef.logodds["category_two_defects", "Std. Error"]

# Display the confidence interval
paste("(", defects.logodds.lower, ",", defects.logodds.upper, ")")
```

```
[1] "( -0.138879080560977 , -0.0213191899815585 )"
```

```r
# Plot log-odds of being a good instructor
plot_model(model5, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```



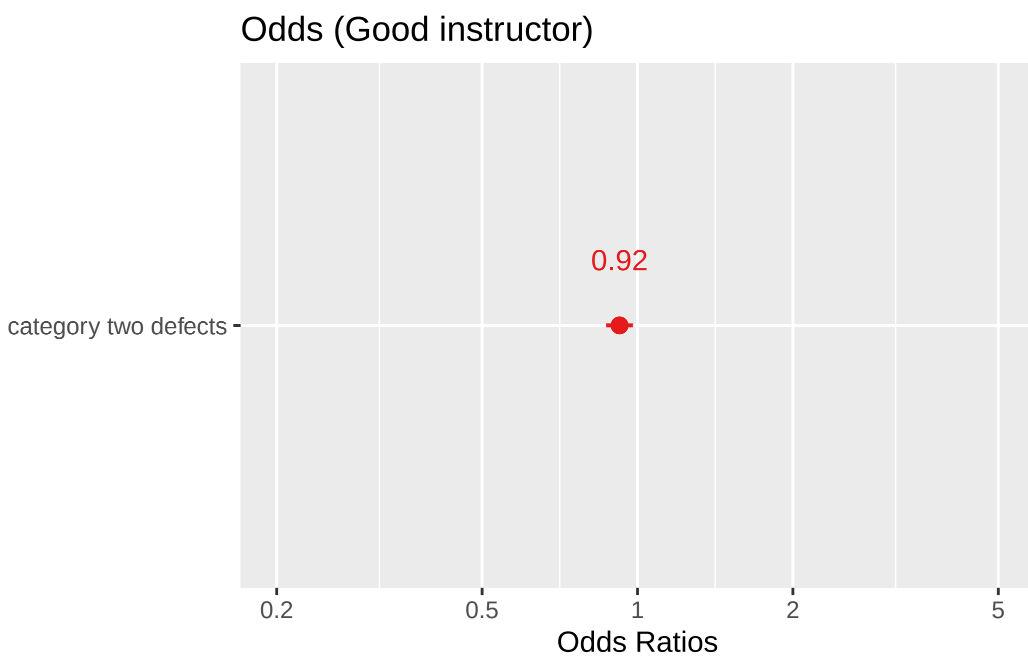
```r
# Calculate lower and upper bounds for 'category_two_defects' odds
exp(mod5.coef.logodds)
```

```
                      Estimate Std. Error     z value Pr(>|z|)
(Intercept)          1.3856408   1.111352 21.96245288 1.002008
category_two_defects 0.9230248   1.030444  0.06919116 1.007594
```

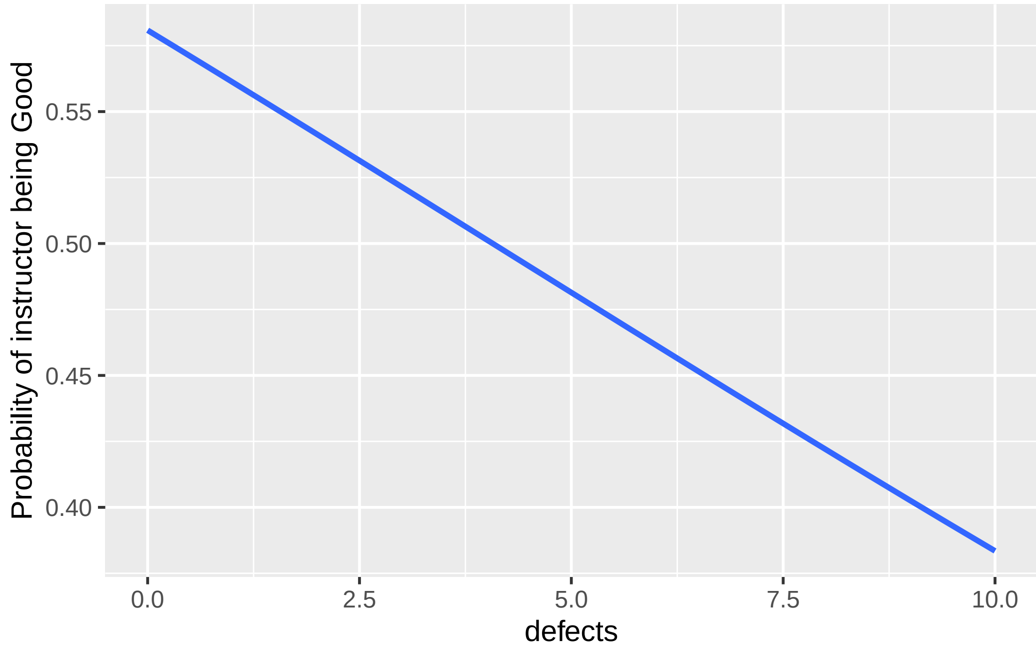
```r
defects.odds.lower <- exp(defects.logodds.lower)
defects.odds.upper <- exp(defects.logodds.upper)

# Display the confidence interval
paste("(", defects.odds.lower, ",", defects.odds.upper, ")")
```

```
[1] "( 0.870333262305556 , 0.978906457563423 )"
```

```r
# Plot odds of being a good instructor
plot_model(model5, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```



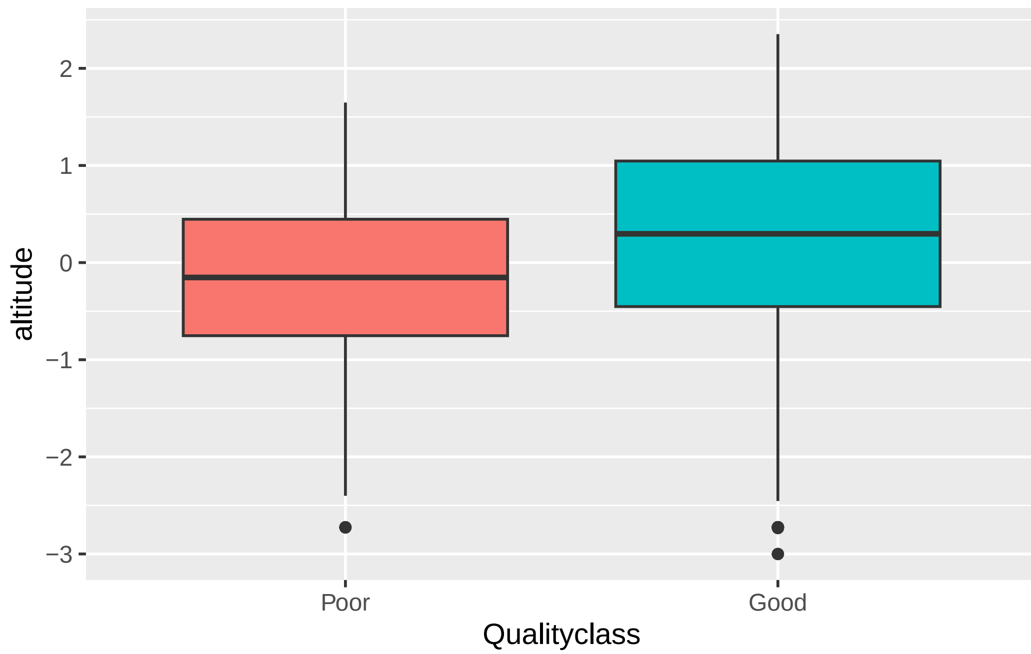
```
# Add predicted probabilities
data_defects_after <- data_defects %>%
                  mutate(logodds.Good = predict(model5, type = "response")) %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model5))

# Plot the relationship between 'category_two_defects' and probability of being a good ins
ggplot(data = data_defects_after, aes(x = category_two_defects, y = probs.Good)) +
  geom_smooth(method="glm",
              method.args = list(family="binomial"),
              se = FALSE) +
  labs(x = "defects", y = "Probability of instructor being Good")
```



3.6 Altitude mean meters and Qualityclass

```
# Select 'altitude_mean_meters' and 'Qualityclass' columns
data_altitude <- data %>%
            dplyr::select(altitude_mean_meters, Qualityclass)

# Create a boxplot of 'altitude_mean_meters' across different 'Qualityclass' levels
```

```r
p5 <- ggplot(data = data_altitude, aes(x = Qualityclass, y = altitude_mean_meters, fill =
  geom_boxplot() +
  labs(x = "Qualityclass", y = "altitude")+
  theme(legend.position = "none")
p5
```




```r
# Fit logistic regression model with 'altitude_mean_meters' predictor and 'Qualityclass' r
model4 <- glm(Qualityclass ~ altitude_mean_meters, data = data_altitude,
             family = binomial(link = "logit"))
model4 %>%
  summary()
```

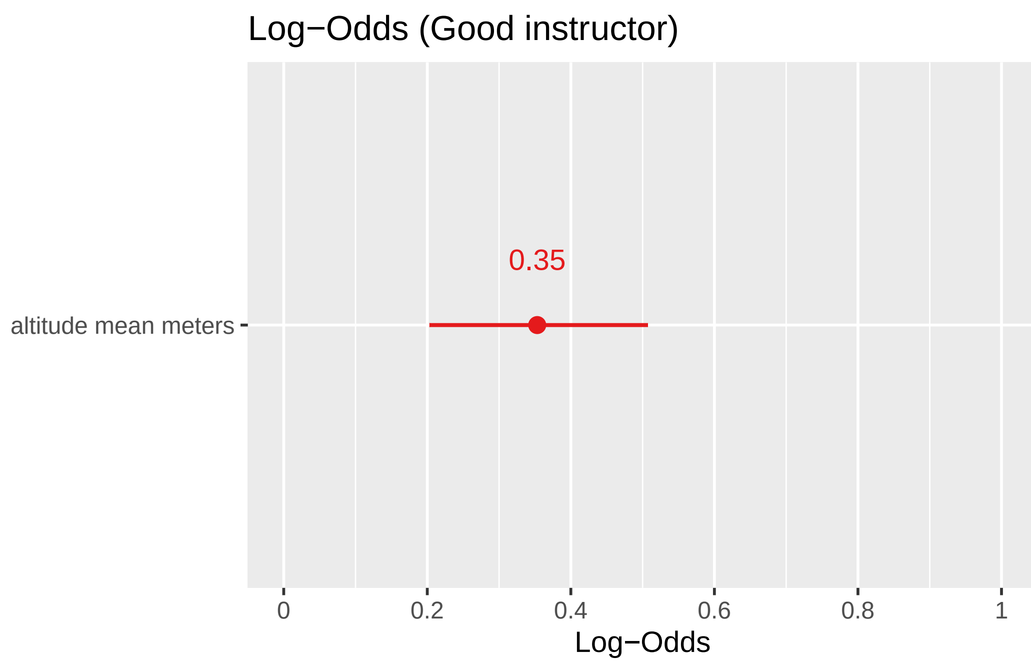
```
Call:
glm(formula = Qualityclass ~ altitude_mean_meters, family = binomial(link = "logit"),
    data = data_altitude)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)            0.1292     0.0755   1.711    0.087 .  
altitude_mean_meters   0.3531     0.0774   4.562 5.06e-06 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  981.86  on 724  degrees of freedom
AIC: 985.86

Number of Fisher Scoring iterations: 4
```

```r
# Calculate lower and upper bounds for 'altitude_mean_meters' log-odds
mod4.coef.logodds <- model4 %>%
                      summary() %>%
                      coef()
altitude.logodds.lower <- mod4.coef.logodds["altitude_mean_meters", "Estimate"] -
                      1.96 * mod4.coef.logodds["altitude_mean_meters", "Std. Error"]
altitude.logodds.upper <- mod4.coef.logodds["altitude_mean_meters", "Estimate"] +
                      1.96 * mod4.coef.logodds["altitude_mean_meters", "Std. Error"]

# Display the confidence interval
paste("(", altitude.logodds.lower, ",", altitude.logodds.upper, ")")
```

```
[1] "( 0.201429364953632 , 0.504856686546668 )"
```

```r
# Plot log-odds of being a good instructor
plot_model(model4, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```



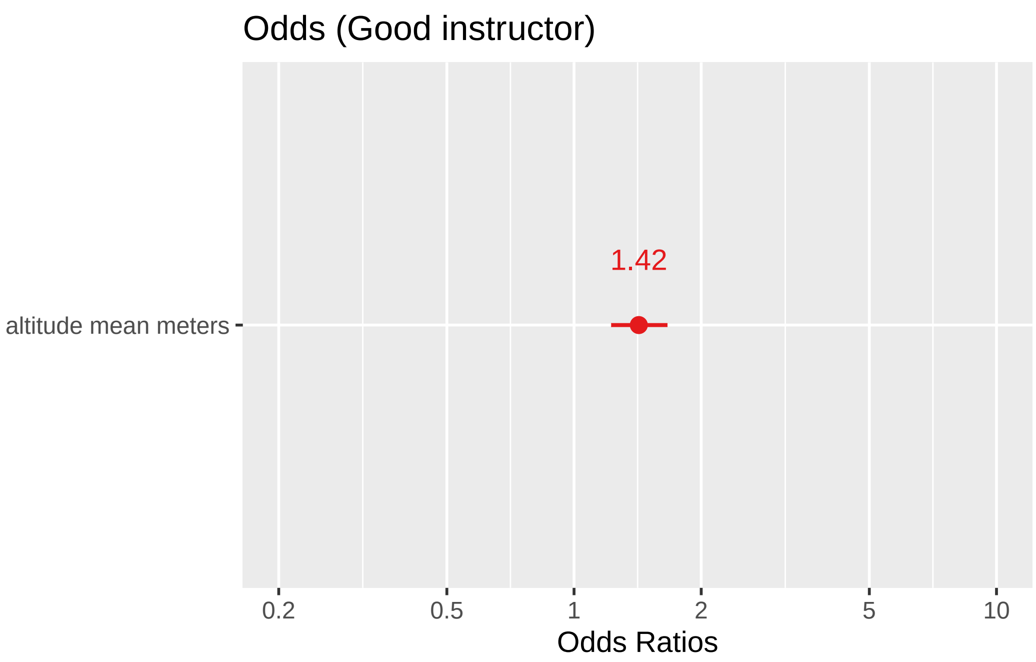

```r
# Calculate lower and upper bounds for 'altitude_mean_meters' odds
exp(mod4.coef.logodds)
```

```
                     Estimate Std. Error   z value Pr(>|z|)
(Intercept)          1.137935   1.078426  5.536717 1.090904
altitude_mean_meters 1.423535   1.080480 95.801741 1.000005
```

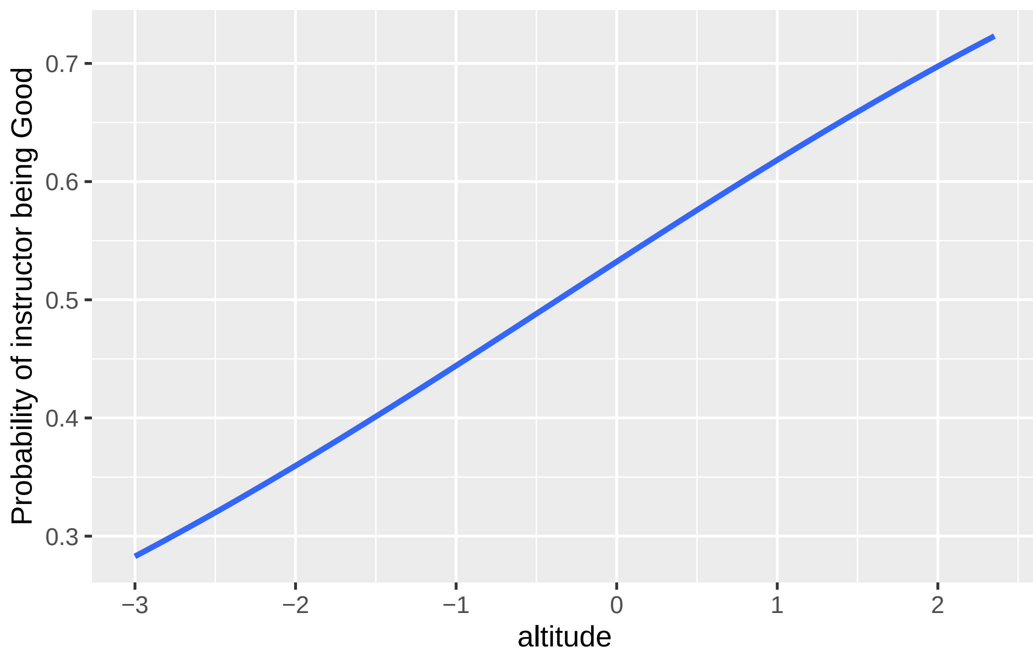
```r
altitude.odds.lower <- exp(altitude.logodds.lower)
altitude.odds.upper <- exp(altitude.logodds.upper)

# Display the confidence interval
paste("(", altitude.odds.lower, ",", altitude.odds.upper, ")")
```

```
[1] "( 1.22314983676597 , 1.65674806915918 )"
```

```r
# Plot odds of being a good instructor
plot_model(model4, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```




```
# Add predicted probabilities
data_altitude_after <- data_altitude %>%
                  mutate(logodds.Good = predict(model4), type = "response") %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model4))

# Plot the relationship between 'altitude_mean_meters' and probability of being a good ins
ggplot(data = data_altitude_after, aes(x = altitude_mean_meters, y = probs.Good)) +
  geom_smooth(method="glm",
              method.args = list(family="binomial"),
              se = FALSE) +
  labs(x = "altitude", y = "Probability of instructor being Good")
```

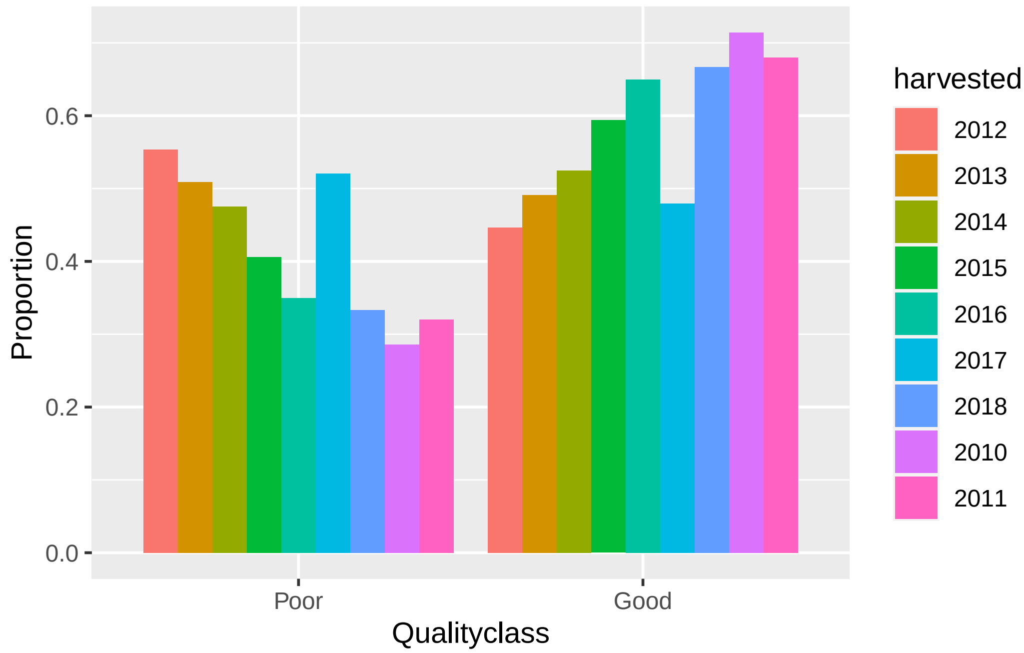


3.7 Harvested and Qualityclass

```
# Select 'harvested' and 'Qualityclass' columns and generate a contingency table.
data_harvested <- data %>%
          dplyr::select(harvested, Qualityclass)
data_harvested %>%
  tabyl(harvested, Qualityclass) %>%
  adorn_percentages() %>%
  adorn_pct_formatting() %>%
  adorn_ns()
```

```
harvested       Poor       Good
     2012 55.4% (98) 44.6                                           (79)
     2013 50.9% (57) 49.1                                           (55)
     2014 47.5% (77) 52.5                                           (85)
     2015 40.6% (39) 59.4                                           (57)
     2016 35.0% (28) 65.0                                           (52)
     2017 52.1% (25) 47.9                                           (23)
     2018 33.3%  (4) 66.7%  (8)
     2010 28.6%  (4) 71.4                                           (10)
     2011 32.0%  (8) 68.0                                           (17)
```

```r
# Create a barplot of 'harvested' across different 'Qualityclass' levels
p6 <- ggplot(data_harvested, aes(x = Qualityclass, y = after_stat(prop), group = harvested
    geom_bar(position = "dodge", stat = "count") +
    labs(y = "Proportion")
p6
```




```r
# Fit logistic regression model with 'harvested' predictor and 'Qualityclass' response
model_harvested <- glm(Qualityclass ~ harvested, data = data_harvested,
             family = binomial(link = "logit"))
model_harvested %>%
  summary()
```

```
Call:
glm(formula = Qualityclass ~ harvested, family = binomial(link = "logit"),
    data = data_harvested)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.2155     0.1512  -1.425  0.15405
harvested2013   0.1798     0.2420   0.743  0.45758
```

```
harvested2014   0.3144     0.2182   1.441  0.14968
harvested2015   0.5950     0.2570   2.315  0.02060 *
harvested2016   0.8346     0.2789   2.992  0.00277 **
harvested2017   0.1321     0.3261   0.405  0.68532
harvested2018   0.9087     0.6308   1.441  0.14970
harvested2010   1.1318     0.6106   1.854  0.06381 .
harvested2011   0.9693     0.4546   2.132  0.03300 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  985.86  on 717  degrees of freedom
AIC: 1003.9

Number of Fisher Scoring iterations: 4
```

```
# Extract coefficients from the model and calculate their confidence intervals.
model_harvested_coef_logodds <- model_harvested %>%
                            summary() %>%
                            coef()
model_harvested_coef_logodds
```

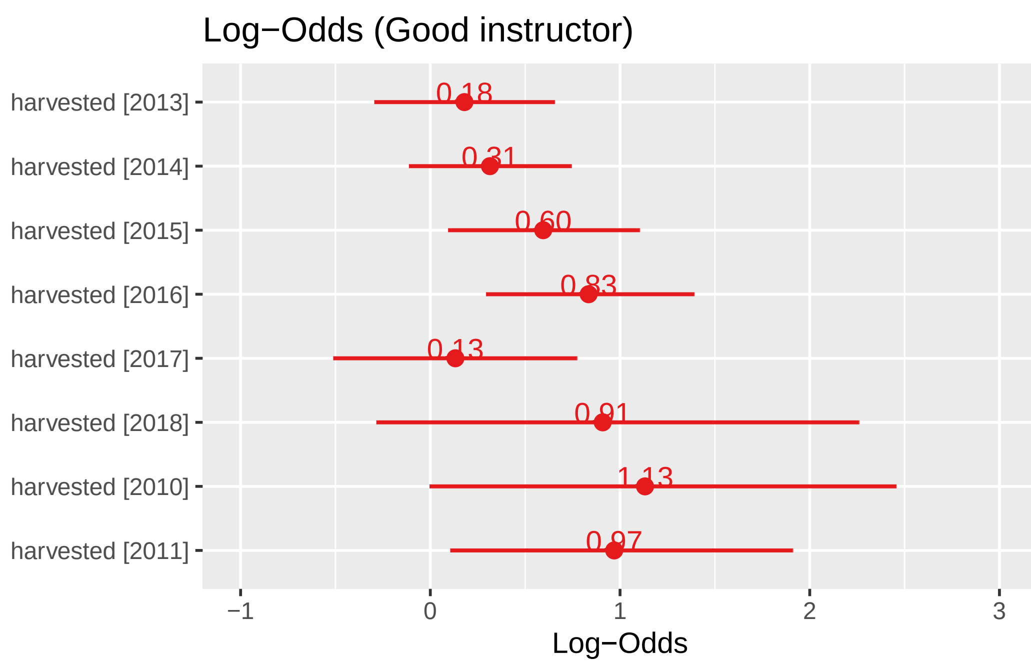
```
                  Estimate Std. Error    z value    Pr(>|z|)
(Intercept)     -0.2155196  0.1512029 -1.4253672 0.154051070
harvested2013    0.1798015  0.2420496  0.7428295 0.457584930
harvested2014    0.3143655  0.2182064  1.4406794 0.149675263
harvested2015    0.5950092  0.2569965  2.3152427 0.020599652
harvested2016    0.8345588  0.2789397  2.9918970 0.002772498
harvested2017    0.1321380  0.3260990  0.4052083 0.685324431
harvested2018    0.9086668  0.6307632  1.4405832 0.149702476
harvested2010    1.1318104  0.6106242  1.8535303 0.063806392
harvested2011    0.9692914  0.4546270  2.1320585 0.033002036
```

```
confint_logodds <- confint(model_harvested)
confint_logodds
```

```
                      2.5 %     97.5 %
(Intercept)     -0.513996131 0.07980255
```

```
harvested2013 -0.294890090 0.65521661
harvested2014 -0.112560006 0.74361934
harvested2015  0.094381250 1.10355226
harvested2016  0.294436546 1.39064763
harvested2017 -0.511169744 0.77256130
harvested2018 -0.284159897 2.25903857
harvested2010 -0.003857604 2.45516556
harvested2011  0.105669074 1.91041856
```

```
# Plot log-odds of being a good instructor
plot_model(model_harvested, show.values = TRUE, transform = NULL,
           title = "Log-Odds (Good instructor)", show.p = FALSE)
```




```
# Transform the coefficients into odds ratios and obtain their confidence intervals
model_harvested_coef_odds <- model_harvested %>%
  summary() %>%
  coef() %>%
  exp()
model_harvested_coef_odds
```

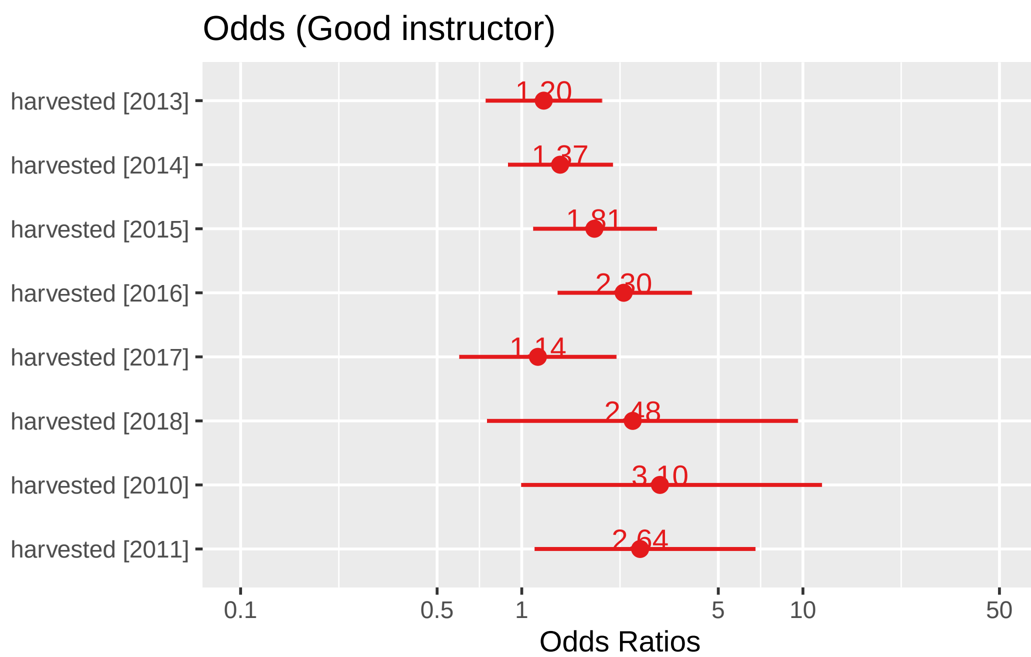
```
              Estimate Std. Error    z value Pr(>|z|)
```

```
(Intercept)   0.8061224   1.163233  0.2404202 1.166550
harvested2013 1.1969798   1.273857  2.1018743 1.580253
harvested2014 1.3693901   1.243844  4.2235645 1.161457
harvested2015 1.8130477   1.293041 10.1273808 1.020813
harvested2016 2.3037975   1.321728 19.9234411 1.002776
harvested2017 1.1412658   1.385553  1.4996148 1.984416
harvested2018 2.4810127   1.879044  4.2231580 1.161489
harvested2010 3.1012658   1.841580  6.3823111 1.065886
harvested2011 2.6360759   1.575586  8.4322069 1.033553
```

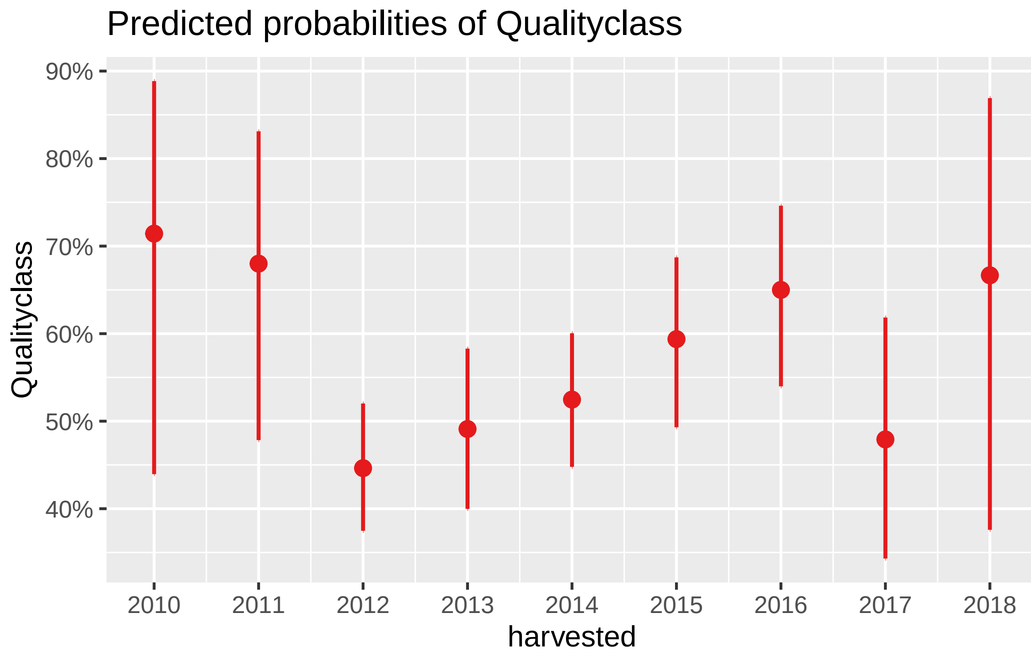
```
exp(confint_logodds)
```

```
                  2.5 %    97.5 %
(Intercept)   0.5981007  1.083073
harvested2013 0.7446134  1.925560
harvested2014 0.8935437  2.103535
harvested2015 1.0989787  3.014857
harvested2016 1.3423698  4.017451
harvested2017 0.5997936  2.165305
harvested2018 0.7526463  9.573880
harvested2010 0.9961498 11.648362
harvested2011 1.1114540  6.755916
```

```
# Plot odds of being a good instructor
plot_model(model_harvested, show.values = TRUE,
           title = "Odds (Good instructor)", show.p = FALSE)
```

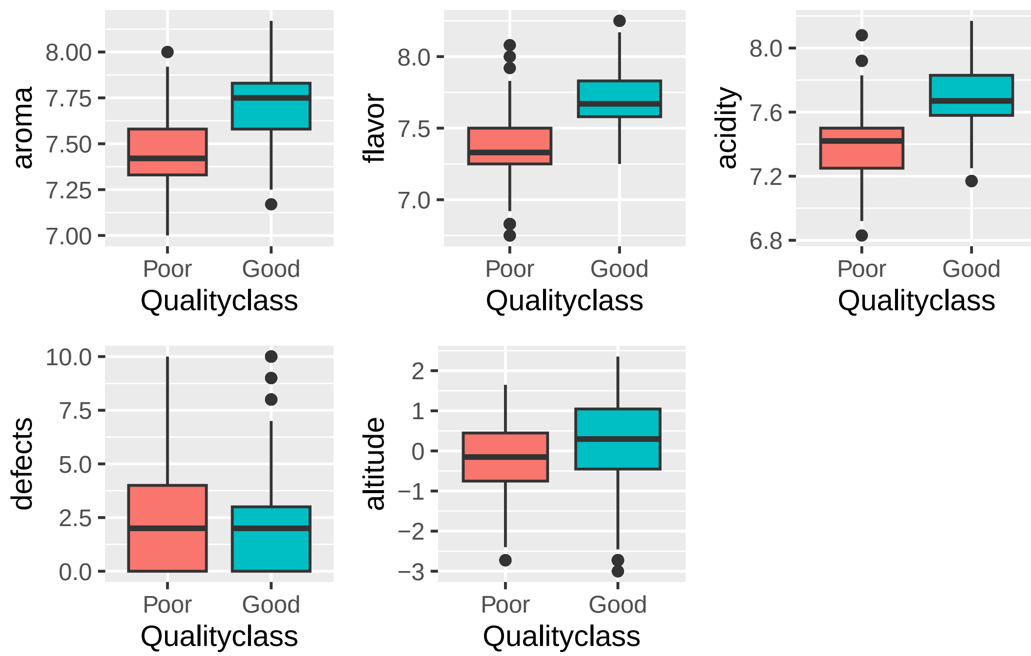



```r
# Calculate log odds, odds, and probabilities and store them
data_harvested_after <- data_harvested %>%
                  mutate(logodds.Good = predict(model_harvested, type = "response")) %>%
                  mutate(odds.Good = exp(logodds.Good)) %>%
                  mutate(probs.Good = fitted(model_harvested))

# Generate a predictive plot
plot_model(model_harvested, type = "pred",
           terms = c("harvested"))
```

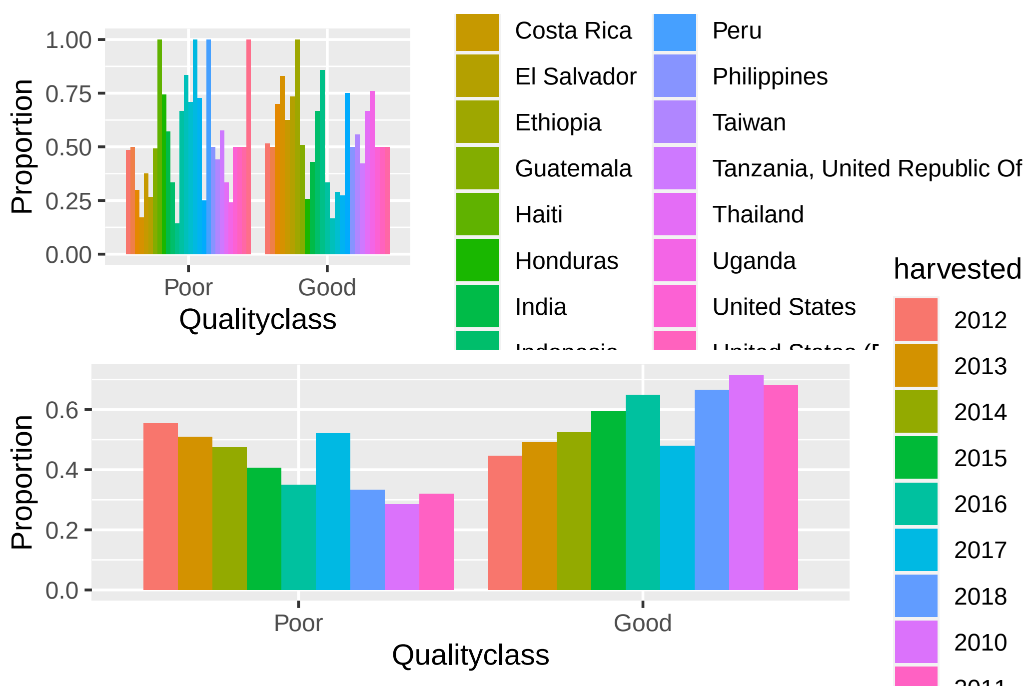


3.8 Plot Arrange

```
# Arrange multiple plots
grid.arrange(p1, p2, p3, p4, p5, ncol = 3)
```




```
grid.arrange(p0, p6)
```



4 Formal Analysis

4.1 Principal Component Analysis

Based on the correlation matrix, it is evident that some predictors exhibit high correlation. Therefore, we adopt principal component analysis (PCA) to help address multicollinearity, thereby enhancing the stability and interpretability of the model.

```
data_cor <- data %>%
  dplyr::select(aroma, flavor, acidity)
correlation_matrix <- cor(data_cor)
print(correlation_matrix)
```

```
            aroma    flavor   acidity
aroma   1.0000000 0.6901724 0.5399363
flavor  0.6901724 1.0000000 0.7174280
acidity 0.5399363 0.7174280 1.0000000
```

```
# Principal principal component analysis (PCA) for 'aroma', 'flavor' and  'acidity'
data_pca <- data %>%
  dplyr::select(aroma, flavor, acidity, Qualityclass)
data_scaled <- scale(data_pca[, -4])
pca_result <- prcomp(data_scaled)
summary(pca_result)
```

```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.5170 0.6790 0.48747
Proportion of Variance 0.7671 0.1537 0.07921
Cumulative Proportion  0.7671 0.9208 1.00000
```

The cumulative proportion of the three predictor variables adds up to 1, indicating that these three principal components fully explain the variability in the original data without losing information. Therefore, adopting principal component analysis is justified.

```
# Predict PCA components and choose the first two components
pca_result_selected <- predict(pca_result, newdata = data_scaled)[, 1:2]

# Combine PCA components with other variables
data_pca_final <- data.frame(pca_result_selected, country_of_origin = data$country_of_orig
```

```r
# Retrieve column names of the new data frame
names(data_pca_final)
```

```
[1] "PC1"                 "PC2"                 "country_of_origin"
[4] "category_two_defects" "altitude_mean_meters" "harvested"
[7] "Qualityclass"
```

4.2 Model Selection

```r
# Conduct an origin model
model_full <- glm(Qualityclass ~ country_of_origin + aroma + flavor + acidity + category_t
             family = binomial(link = "logit"))
# Summarize the model
model_full %>%
  summary()
```

```
Call:
glm(formula = Qualityclass ~ country_of_origin + aroma + flavor +
    acidity + category_two_defects + altitude_mean_meters + harvested,
    family = binomial(link = "logit"), data = data)

Coefficients:
                                              Estimate Std. Error z value
(Intercept)                                 -155.92356   13.72078 -11.364
country_of_originBurundi                       1.92335    5.32186   0.361
country_of_originChina                         0.51607    1.23498   0.418
country_of_originColombia                      1.79012    0.63394   2.824
country_of_originCosta Rica                    0.38187    0.87151   0.438
country_of_originEl Salvador                   0.17324    0.97069   0.178
country_of_originEthiopia                     12.19028 1178.76284   0.010
country_of_originGuatemala                    -0.82776    0.61013  -1.357
country_of_originHaiti                       -13.36267 3956.18039  -0.003
country_of_originHonduras                     -1.10992    0.80864  -1.373
country_of_originIndia                        -3.07110    1.13658  -2.702
country_of_originIndonesia                    -0.68124    1.19806  -0.569
country_of_originKenya                         0.02514    1.77726   0.014
country_of_originLaos                          1.13360    1.96356   0.577
country_of_originMalawi                       -0.65762    1.41622  -0.464
```

```
country_of_originMexico                          -0.89996    0.57619  -1.562
country_of_originMyanmar                        -14.36618 2797.39843  -0.005
country_of_originNicaragua                        0.05486    1.82141   0.030
country_of_originPanama                           3.38904    1.78721   1.896
country_of_originPeru                           -18.66679 2192.69041  -0.009
country_of_originPhilippines                      2.80520    3.13355   0.895
country_of_originTaiwan                           0.62404    0.78671   0.793
country_of_originTanzania, United Republic Of     0.93548    0.91419   1.023
country_of_originThailand                         2.18907    0.95049   2.303
country_of_originUganda                          -1.48521    0.86689  -1.713
country_of_originUnited States                    1.84776    2.01063   0.919
country_of_originUnited States (Puerto Rico)     -1.41603    1.49613  -0.946
country_of_originVietnam                          1.67138    1.29784   1.288
country_of_originZambia                         -13.01366 3956.18042  -0.003
aroma                                             6.03872    0.99701   6.057
flavor                                            8.29375    1.15796   7.162
acidity                                           6.21276    0.98315   6.319
category_two_defects                              0.11822    0.05970   1.980
altitude_mean_meters                              0.24303    0.18054   1.346
harvested2013                                     0.40419    0.46422   0.871
harvested2014                                     0.51386    0.51356   1.001
harvested2015                                     0.42021    0.53345   0.788
harvested2016                                     1.33777    0.58920   2.270
harvested2017                                     1.28520    0.63597   2.021
harvested2018                                     2.34356    1.15243   2.034
harvested2010                                    -0.01028    1.06170  -0.010
harvested2011                                    -0.40776    0.78504  -0.519
                                               Pr(>|z|)
(Intercept)                                     < 2e-16 ***
country_of_originBurundi                        0.71780
country_of_originChina                          0.67603
country_of_originColombia                       0.00475 **
country_of_originCosta Rica                     0.66126
country_of_originEl Salvador                    0.85835
country_of_originEthiopia                       0.99175
country_of_originGuatemala                      0.17487
country_of_originHaiti                          0.99731
country_of_originHonduras                       0.16988
country_of_originIndia                          0.00689 **
country_of_originIndonesia                      0.56961
country_of_originKenya                          0.98871
country_of_originLaos                           0.56372
country_of_originMalawi                         0.64240
```

```
country_of_originMexico                             0.11831
country_of_originMyanmar                            0.99590
country_of_originNicaragua                          0.97597
country_of_originPanama                             0.05792 .
country_of_originPeru                               0.99321
country_of_originPhilippines                        0.37067
country_of_originTaiwan                             0.42765
country_of_originTanzania, United Republic Of       0.30617
country_of_originThailand                           0.02127 *
country_of_originUganda                             0.08666 .
country_of_originUnited States                      0.35810
country_of_originUnited States (Puerto Rico)        0.34391
country_of_originVietnam                            0.19781
country_of_originZambia                             0.99738
aroma                                              1.39e-09 ***
flavor                                             7.93e-13 ***
acidity                                            2.63e-10 ***
category_two_defects                                0.04767 *
altitude_mean_meters                                0.17827
harvested2013                                       0.38393
harvested2014                                       0.31702
harvested2015                                       0.43086
harvested2016                                       0.02318 *
harvested2017                                       0.04330 *
harvested2018                                       0.04199 *
harvested2010                                       0.99228
harvested2011                                       0.60348
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  361.12  on 684  degrees of freedom
AIC: 445.12

Number of Fisher Scoring iterations: 16
```

```r
# Perform stepwise variable selection using AIC
stepAIC(model_full)
```

```
Start:  AIC=445.12
```

```
Qualityclass ~ country_of_origin + aroma + flavor + acidity +
    category_two_defects + altitude_mean_meters + harvested

                       Df Deviance    AIC
- harvested             8   373.02 441.02
- altitude_mean_meters  1   362.91 444.91
<none>                      361.12 445.12
- category_two_defects  1   365.13 447.13
- country_of_origin    28   435.09 463.09
- aroma                 1   405.99 487.99
- acidity               1   409.76 491.76
- flavor                1   429.83 511.83

Step:  AIC=441.02
Qualityclass ~ country_of_origin + aroma + flavor + acidity +
    category_two_defects + altitude_mean_meters

                       Df Deviance    AIC
- altitude_mean_meters  1   374.31 440.31
<none>                      373.02 441.02
- category_two_defects  1   376.81 442.81
- country_of_origin    28   452.49 464.49
- aroma                 1   414.51 480.51
- acidity               1   428.86 494.86
- flavor                1   441.52 507.52

Step:  AIC=440.31
Qualityclass ~ country_of_origin + aroma + flavor + acidity +
    category_two_defects

                       Df Deviance    AIC
<none>                      374.31 440.31
- category_two_defects  1   378.03 442.03
- country_of_origin    28   459.25 469.25
- aroma                 1   415.92 479.92
- acidity               1   431.31 495.31
- flavor                1   442.07 506.07


Call:  glm(formula = Qualityclass ~ country_of_origin + aroma + flavor +
    acidity + category_two_defects, family = binomial(link = "logit"),
    data = data)
```

```
Coefficients:
                          (Intercept)
                            -151.2691
              country_of_originBurundi
                               2.3536
                country_of_originChina
                               0.5452
             country_of_originColombia
                               1.7268
           country_of_originCosta Rica
                               0.6757
          country_of_originEl Salvador
                               0.6893
             country_of_originEthiopia
                              12.7080
            country_of_originGuatemala
                              -0.5742
                country_of_originHaiti
                             -13.9659
             country_of_originHonduras
                              -0.5453
                country_of_originIndia
                              -3.3769
            country_of_originIndonesia
                              -0.7236
                country_of_originKenya
                               1.1221
                 country_of_originLaos
                               1.1074
               country_of_originMalawi
                              -0.5442
               country_of_originMexico
                              -1.2410
              country_of_originMyanmar
                             -14.7769
            country_of_originNicaragua
                               0.2816
               country_of_originPanama
                               3.0527
                 country_of_originPeru
                             -18.7494
          country_of_originPhilippines
```

```
                                              2.6875
                              country_of_originTaiwan
                                              0.5075
country_of_originTanzania, United Republic Of
                                              1.0911
                            country_of_originThailand
                                              1.8521
                              country_of_originUganda
                                             -1.2918
                       country_of_originUnited States
                                              1.6331
 country_of_originUnited States (Puerto Rico)
                                             -1.8368
                             country_of_originVietnam
                                              2.0831
                              country_of_originZambia
                                            -13.1361
                                               aroma
                                              5.6093
                                              flavor
                                              7.9288
                                             acidity
                                              6.4595
                                 category_two_defects
                                              0.1105

Degrees of Freedom: 725 Total (i.e. Null);  693 Residual
Null Deviance:      1004
Residual Deviance: 374.3    AIC: 440.3
```

```
# Fit logistic regression model with PCA components
pca_model <- glm(Qualityclass ~ ., data = data_pca_final, family = binomial(link = "logit"

# Summarize the model
pca_model %>%
  summary()
```

```
Call:
glm(formula = Qualityclass ~ ., family = binomial(link = "logit"),
    data = data_pca_final)
```

```
Coefficients:
                                                Estimate Std. Error z value
(Intercept)                                     -0.53690    0.56380  -0.952
PC1                                              0.73240    0.06421  11.407
PC2                                             -0.04793    0.05337  -0.898
country_of_originBurundi                         1.98104    5.31502   0.373
country_of_originChina                           0.46641    1.21791   0.383
country_of_originColombia                        1.78627    0.62792   2.845
country_of_originCosta Rica                      0.21040    0.87295   0.241
country_of_originEl Salvador                     0.09598    0.96253   0.100
country_of_originEthiopia                       12.20229 1185.83950   0.010
country_of_originGuatemala                      -0.92737    0.60330  -1.537
country_of_originHaiti                         -13.51581 3956.18039  -0.003
country_of_originHonduras                       -1.16734    0.79810  -1.463
country_of_originIndia                          -3.07739    1.14565  -2.686
country_of_originIndonesia                      -0.85101    1.16321  -0.732
country_of_originKenya                           0.20522    1.83953   0.112
country_of_originLaos                            1.28170    1.94324   0.660
country_of_originMalawi                         -0.78528    1.41037  -0.557
country_of_originMexico                         -0.99425    0.57001  -1.744
country_of_originMyanmar                       -14.42668 2797.28414  -0.005
country_of_originNicaragua                       0.11966    1.81286   0.066
country_of_originPanama                          3.26732    1.83083   1.785
country_of_originPeru                          -18.54130 2233.75648  -0.008
country_of_originPhilippines                     2.78889    3.27243   0.852
country_of_originTaiwan                          0.57305    0.77609   0.738
country_of_originTanzania, United Republic Of    0.75028    0.90781   0.826
country_of_originThailand                        2.13070    0.95799   2.224
country_of_originUganda                         -1.58278    0.85680  -1.847
country_of_originUnited States                   1.70301    1.89697   0.898
country_of_originUnited States (Puerto Rico)    -1.48829    1.47527  -1.009
country_of_originVietnam                         1.69762    1.24231   1.366
country_of_originZambia                        -13.76143 3956.18039  -0.003
category_two_defects                             0.12801    0.05913   2.165
altitude_mean_meters                             0.21593    0.17791   1.214
harvested2013                                    0.33679    0.46112   0.730
harvested2014                                    0.53815    0.51599   1.043
harvested2015                                    0.43623    0.52952   0.824
harvested2016                                    1.32595    0.58525   2.266
harvested2017                                    1.31205    0.62439   2.101
harvested2018                                    2.09670    1.13556   1.846
harvested2010                                    0.04900    1.06653   0.046
harvested2011                                   -0.48789    0.78374  -0.623
```

```
                                              Pr(>|z|)
(Intercept)                                    0.34095
PC1                                           < 2e-16 ***
PC2                                            0.36918
country_of_originBurundi                       0.70935
country_of_originChina                         0.70175
country_of_originColombia                      0.00445 **
country_of_originCosta Rica                    0.80954
country_of_originEl Salvador                   0.92057
country_of_originEthiopia                      0.99179
country_of_originGuatemala                     0.12425
country_of_originHaiti                         0.99727
country_of_originHonduras                      0.14356
country_of_originIndia                         0.00723 **
country_of_originIndonesia                     0.46441
country_of_originKenya                         0.91117
country_of_originLaos                          0.50953
country_of_originMalawi                        0.57767
country_of_originMexico                        0.08111 .
country_of_originMyanmar                       0.99589
country_of_originNicaragua                     0.94737
country_of_originPanama                        0.07432 .
country_of_originPeru                          0.99338
country_of_originPhilippines                   0.39408
country_of_originTaiwan                        0.46028
country_of_originTanzania, United Republic Of  0.40854
country_of_originThailand                      0.02614 *
country_of_originUganda                        0.06470 .
country_of_originUnited States                 0.36932
country_of_originUnited States (Puerto Rico)   0.31306
country_of_originVietnam                       0.17178
country_of_originZambia                        0.99722
category_two_defects                           0.03041 *
altitude_mean_meters                           0.22485
harvested2013                                  0.46516
harvested2014                                  0.29697
harvested2015                                  0.41004
harvested2016                                  0.02347 *
harvested2017                                  0.03561 *
harvested2018                                  0.06483 .
harvested2010                                  0.96336
harvested2011                                  0.53361
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  364.01  on 685  degrees of freedom
AIC: 446.01

Number of Fisher Scoring iterations: 16
```

```r
pca_model_summary <- glance(pca_model)
kable(pca_model_summary, digits = 2)
```

null.deviance	df.null	logLik	AIC	BIC	deviance	df.residual	nobs
1003.53	725	-182.01	446.01	634.1	364.01	685	726

```r
# Perform stepwise variable selection using AIC
stepAIC(pca_model)
```

```
Start:  AIC=446.01
Qualityclass ~ PC1 + PC2 + country_of_origin + category_two_defects +
    altitude_mean_meters + harvested

                       Df Deviance    AIC
- harvested             8   375.55 441.55
- PC2                   1   364.82 444.82
- altitude_mean_meters  1   365.47 445.47
<none>                      364.01 446.01
- category_two_defects  1   368.80 448.80
- country_of_origin    28   441.10 467.10
- PC1                   1   841.93 921.93

Step:  AIC=441.55
Qualityclass ~ PC1 + PC2 + country_of_origin + category_two_defects +
    altitude_mean_meters

                       Df Deviance    AIC
- altitude_mean_meters  1   376.57 440.57
<none>                      375.55 441.55
```

```
- PC2                  1   377.60 441.60
- category_two_defects  1   380.00 444.00
- country_of_origin    28   457.58 467.58
- PC1                  1   861.89 925.89

Step:  AIC=440.57
Qualityclass ~ PC1 + PC2 + country_of_origin + category_two_defects

                      Df Deviance    AIC
<none>                     376.57 440.57
- PC2                  1   378.66 440.66
- category_two_defects  1   380.95 442.95
- country_of_origin    28   462.95 470.95
- PC1                  1   866.45 928.45


Call:  glm(formula = Qualityclass ~ PC1 + PC2 + country_of_origin + 
    category_two_defects, family = binomial(link = "logit"), 
    data = data_pca_final)

Coefficients:
                                  (Intercept)  
                                     -0.04233  
                                          PC1  
                                      0.71510  
                                          PC2  
                                     -0.07326  
                     country_of_originBurundi  
                                      2.36727  
                       country_of_originChina  
                                      0.50734  
                    country_of_originColombia  
                                      1.67756  
                  country_of_originCosta Rica  
                                      0.49789  
                 country_of_originEl Salvador  
                                      0.64059  
                    country_of_originEthiopia  
                                     12.69589  
                   country_of_originGuatemala  
                                     -0.66784  
                       country_of_originHaiti  
```

```
                                             -14.11903
                              country_of_originHonduras
                                              -0.61226
                                 country_of_originIndia
                                              -3.35010
                             country_of_originIndonesia
                                              -0.82978
                                 country_of_originKenya
                                               1.21909
                                  country_of_originLaos
                                               1.25492
                                country_of_originMalawi
                                              -0.64001
                                country_of_originMexico
                                              -1.32426
                               country_of_originMyanmar
                                             -14.79615
                             country_of_originNicaragua
                                               0.39048
                                country_of_originPanama
                                               2.95741
                                  country_of_originPeru
                                             -18.60818
                           country_of_originPhilippines
                                               2.68011
                                country_of_originTaiwan
                                               0.51160
     country_of_originTanzania, United Republic Of
                                               0.93111
                              country_of_originThailand
                                               1.81917
                                country_of_originUganda
                                              -1.43401
                          country_of_originUnited States
                                               1.55667
             country_of_originUnited States (Puerto Rico)
                                              -1.92046
                               country_of_originVietnam
                                               2.09631
                                country_of_originZambia
                                             -13.75634
                                   category_two_defects
                                               0.11885
```

```
Degrees of Freedom: 725 Total (i.e. Null);  694 Residual
Null Deviance:      1004
Residual Deviance: 376.6    AIC: 440.6
```

After reducing dimensionality using PCA, we selected the model with the lowest AIC, which is considered the optimal model.

```r
# Final Logistic Regression Model for Qualityclass Prediction
optimal_model <- glm(Qualityclass ~ PC1 + PC2 + country_of_origin + category_two_defects,
optimal_model %>%
  summary()
```

```
Call:
glm(formula = Qualityclass ~ PC1 + PC2 + country_of_origin +
    category_two_defects, family = binomial(link = "logit"),
    data = data_pca_final)

Coefficients:
                                                Estimate Std. Error z value
(Intercept)                                     -0.04233    0.40606  -0.104
PC1                                              0.71510    0.06138  11.650
PC2                                             -0.07326    0.05089  -1.440
country_of_originBurundi                         2.36727    6.39002   0.370
country_of_originChina                           0.50734    1.11791   0.454
country_of_originColombia                        1.67756    0.52479   3.197
country_of_originCosta Rica                      0.49789    0.82000   0.607
country_of_originEl Salvador                     0.64059    0.94261   0.680
country_of_originEthiopia                       12.69589 1190.24631   0.011
country_of_originGuatemala                      -0.66784    0.50484  -1.323
country_of_originHaiti                         -14.11903 3956.18036  -0.004
country_of_originHonduras                       -0.61226    0.70122  -0.873
country_of_originIndia                          -3.35010    1.09232  -3.067
country_of_originIndonesia                      -0.82978    0.99431  -0.835
country_of_originKenya                           1.21909    1.68597   0.723
country_of_originLaos                            1.25492    1.91779   0.654
country_of_originMalawi                         -0.64001    1.32040  -0.485
country_of_originMexico                         -1.32426    0.49031  -2.701
country_of_originMyanmar                       -14.79615 2795.88305  -0.005
country_of_originNicaragua                       0.39048    2.05109   0.190
country_of_originPanama                          2.95741    2.02995   1.457
```

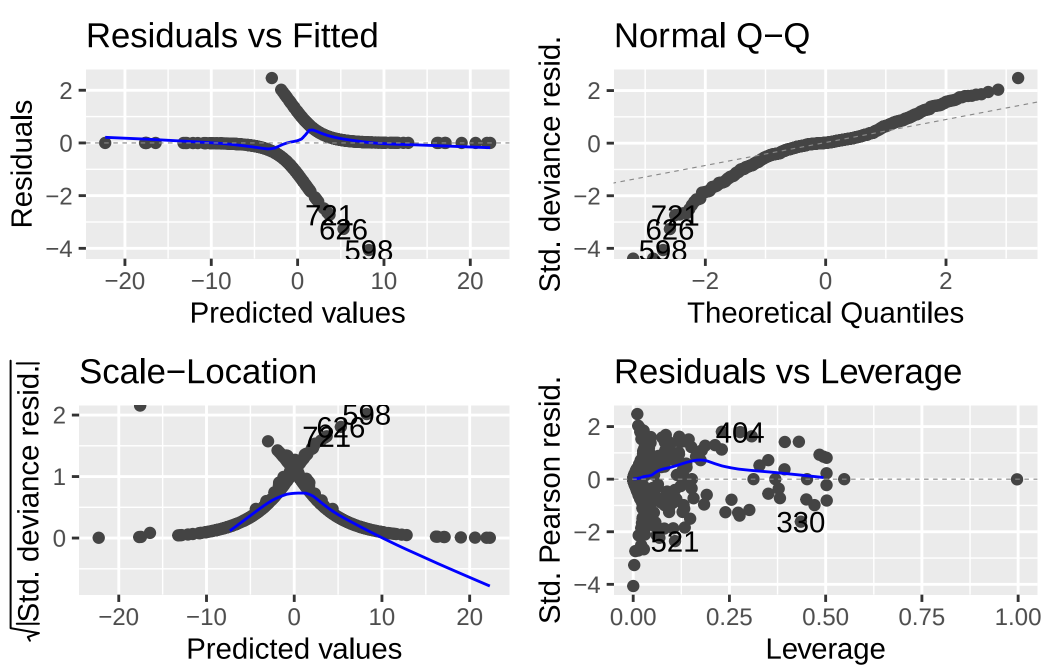
```
country_of_originPeru                          -18.60818 2254.36846  -0.008
country_of_originPhilippines                     2.68011    3.15654   0.849
country_of_originTaiwan                          0.51160    0.70565   0.725
country_of_originTanzania, United Republic Of    0.93111    0.79218   1.175
country_of_originThailand                        1.81917    0.88562   2.054
country_of_originUganda                         -1.43401    0.72861  -1.968
country_of_originUnited States                   1.55667    1.79549   0.867
country_of_originUnited States (Puerto Rico)    -1.92046    1.39464  -1.377
country_of_originVietnam                         2.09631    1.20813   1.735
country_of_originZambia                        -13.75634 3956.18037  -0.003
category_two_defects                             0.11885    0.05748   2.068
                                              Pr(>|z|)
(Intercept)                                    0.91697
PC1                                            < 2e-16 ***
PC2                                            0.15000
country_of_originBurundi                       0.71104
country_of_originChina                         0.64995
country_of_originColombia                      0.00139 **
country_of_originCosta Rica                    0.54373
country_of_originEl Salvador                   0.49676
country_of_originEthiopia                      0.99149
country_of_originGuatemala                     0.18588
country_of_originHaiti                         0.99715
country_of_originHonduras                      0.38258
country_of_originIndia                         0.00216 **
country_of_originIndonesia                     0.40398
country_of_originKenya                         0.46963
country_of_originLaos                          0.51288
country_of_originMalawi                        0.62788
country_of_originMexico                        0.00692 **
country_of_originMyanmar                       0.99578
country_of_originNicaragua                     0.84901
country_of_originPanama                        0.14515
country_of_originPeru                          0.99341
country_of_originPhilippines                   0.39584
country_of_originTaiwan                        0.46845
country_of_originTanzania, United Republic Of  0.23985
country_of_originThailand                      0.03996 *
country_of_originUganda                        0.04905 *
country_of_originUnited States                 0.38595
country_of_originUnited States (Puerto Rico)   0.16850
country_of_originVietnam                       0.08271 .
country_of_originZambia                        0.99723
```

```
category_two_defects                               0.03867 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1003.53  on 725  degrees of freedom
Residual deviance:  376.57  on 694  degrees of freedom
AIC: 440.57

Number of Fisher Scoring iterations: 16
```

```r
optimal_model_summary <- glance(optimal_model)
kable(optimal_model_summary,digits =2)
```

null.deviance	df.null	logLik	AIC	BIC	deviance	df.residual	nobs
1003.53	725	-188.29	440.57	587.38	376.57	694	726

```r
# Check the assumptions
autoplot(optimal_model)
```



```r
# Cross-validation

# Create 5-fold cross-validation splits
set.seed(123)  # Set seed to ensure reproducible results
folds <- createFolds(data_pca_final$Qualityclass, k = 5)
ctrl <- trainControl(method = "cv", index = folds)

# Train model using cross-validation
model <- train(Qualityclass ~ PC1 + PC2 + country_of_origin + category_two_defects, data =
               family = binomial(link = "logit"), trControl = ctrl)

# View cross-validation results
model
```

```
Generalized Linear Model

726 samples
  4 predictor
  2 classes: 'Poor', 'Good'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 145, 145, 146, 145, 145
Resampling results:

  Accuracy   Kappa
  0.8343629  0.6675048
```

Accuracy: The accuracy score of approximately 83.44% suggests that the model correctly predicted the class labels for around 83.44% of the samples on average across all folds. This indicates a reasonably good predictive performance of the model.

Kappa: The kappa statistic measures the agreement between the predicted and actual class labels, accounting for the possibility of agreement occurring by chance. A kappa value of approximately 0.67 indicates substantial agreement between the predicted and actual class labels beyond what would be expected by chance alone.

Overall, the results suggest that the GLM model performs well in classifying samples into the 'Poor' and 'Good' classes, with a relatively high accuracy and substantial agreement between predicted and actual class labels.

$$\ln\left(\frac{p}{1-p}\right) = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_{PC1} + \hat{\beta}_2 \cdot x_{PC2} + \hat{\beta}_3 \cdot x_{country} + \hat{\beta}_4 \cdot x_{defect} + \epsilon$$

- p is the probability of good qualityclass
- $PC1$ and $PC2$ are variables derived from reducing the dimensions of $aroma$, $flavor$, and $acidity$
- $country\_of\_origin$ and $category\_two\_defects$ are the predictor variables
- β_0 to β_4 are the coefficients of the model
- ϵ is the error term