Summary_6736156

Polatat Suwanit

2024-11-05

This statistical report aims to test the association between ADP-induced platelet aggregation level as well as Clopidogrel resistance with three SNPs, which are composed of rs4244285 (CYP2C192), rs4986893 (CYP2C193), and rs662 (PON1. 192Q>R). From the TSV file, I found two confounding factors, age, and sex, that I used to adjust the association tests. I used linear and logistic regression to test the association. Linear regression for ADP with the SNPs because ADP is a continuous variable for dependent outcomes, whereas logistic regression tests with drug resistance and the SNPs due to the fact that the resistance is binary outcomes.

There are 11 variables in PlateletHw.tsv: IID, ADP, Resistance, rs244285, rs498693, rs662, AGE, SEX, PON.192Q>R, CYP2C192, and CYP2CP3.In additional, sex and ADP are continuous data, while the others are categorical data. rs4224285 correlated with CYP2C19*2 have 0 = GG 1 = AG 2 =AA.On he other hand rs48693(CYP2C193) and rs662(PON1.192Q>R) have 0 =AA, 1 = AG, and 2 = GG. For sex 0 = male , 1 = female, and drug resistance 0 = not resistance 1 = resistance.

Cleaning file

Initially, the file was unprocessed due to minus values in the ADP column, likely from typing errors. I cleaned the data by absolute-valuing the ADP values and wrote a new clean data to the clean folder.

```
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data <- read_tsv("raw_data/PlateletHW.tsv")
```

```
## Rows: 211 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (3): PON1.192Q>R, CYP2C19*2, CYP2C19*3
## dbl (8): IID, ADP, Resistance, rs4244285, rs4986893, rs662, AGE, SEX
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data_clean <- data %>%
  mutate(ADP_abs = abs(ADP))
```

```r
data_clean$ADP_abs <- unlist(data_clean$ADP_abs)
data_clean$ADP <- NULL
names(data_clean)[names(data_clean) == "ADP_abs"] <- "ADP"
data_clean <- data_clean[, c("IID", "ADP", setdiff(names(data_clean), c("IID", "ADP")))]

#Write a new clean data
write_tsv(data_clean, "clean_data/PlateletHW_clean.tsv")
```

Before proceeding, I observe the relationship between ADP and drug resistance. A higher ADP level correlates with increased drug resistance. However, visualization alone lacks statistical confirmation.

```r
library(car)
```

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

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

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

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

```r
scatterplot(Resistance ~ ADP, data=data_clean, reg.line
            = lm, smooth=FALSE)
```

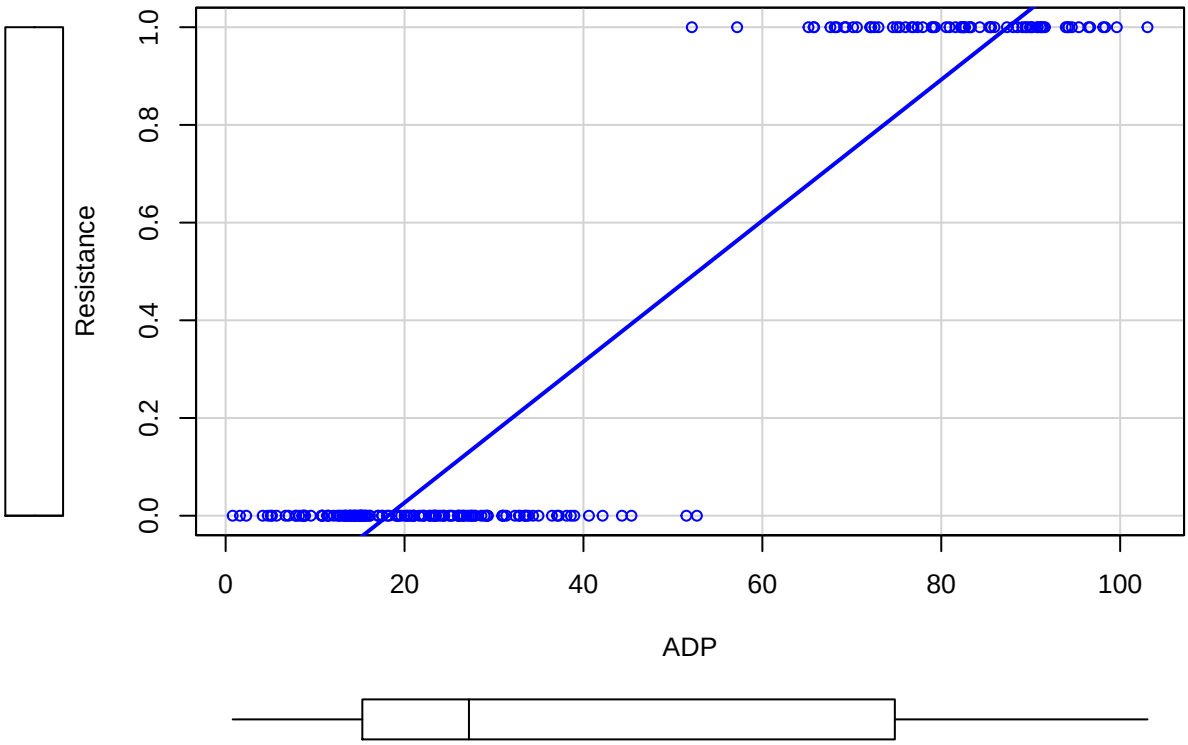
```
## Warning in plot.window(...): "reg.line" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "reg.line" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "reg.line" is not
## a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "reg.line" is not
## a graphical parameter
```

```
## Warning in box(...): "reg.line" is not a graphical parameter
```

```
## Warning in title(...): "reg.line" is not a graphical parameter
```



ADP Statistical Test

I now have real ADP values for test association. I want to check the skewness and outliers using the IQR method. The skewness is moderate, with a value of 0.607. There are no outliers to filter out from IQR test.

```
# install.packages("package_name") install these packages if you do not have before library.
library(tidyverse)
library(ggplot2)
library(e1071)
library(data.table)
```

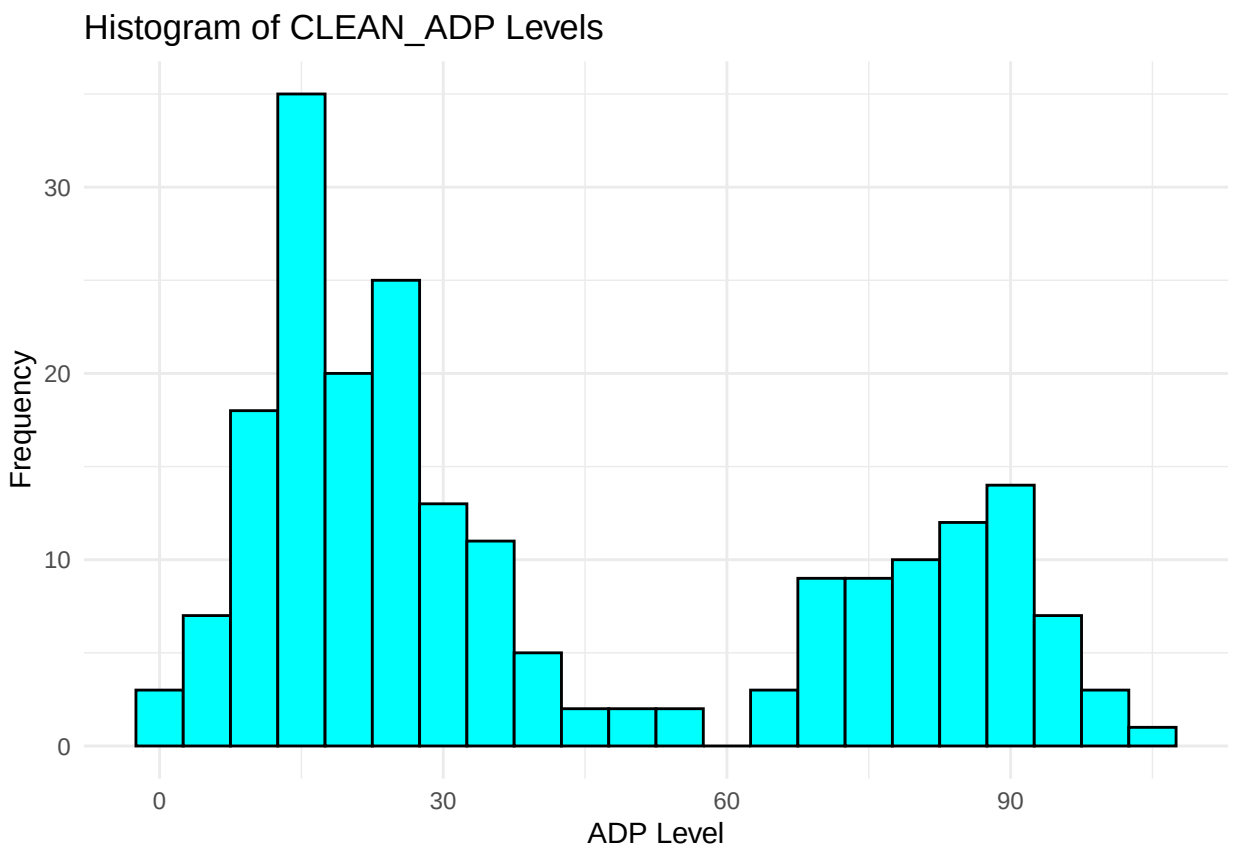
```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

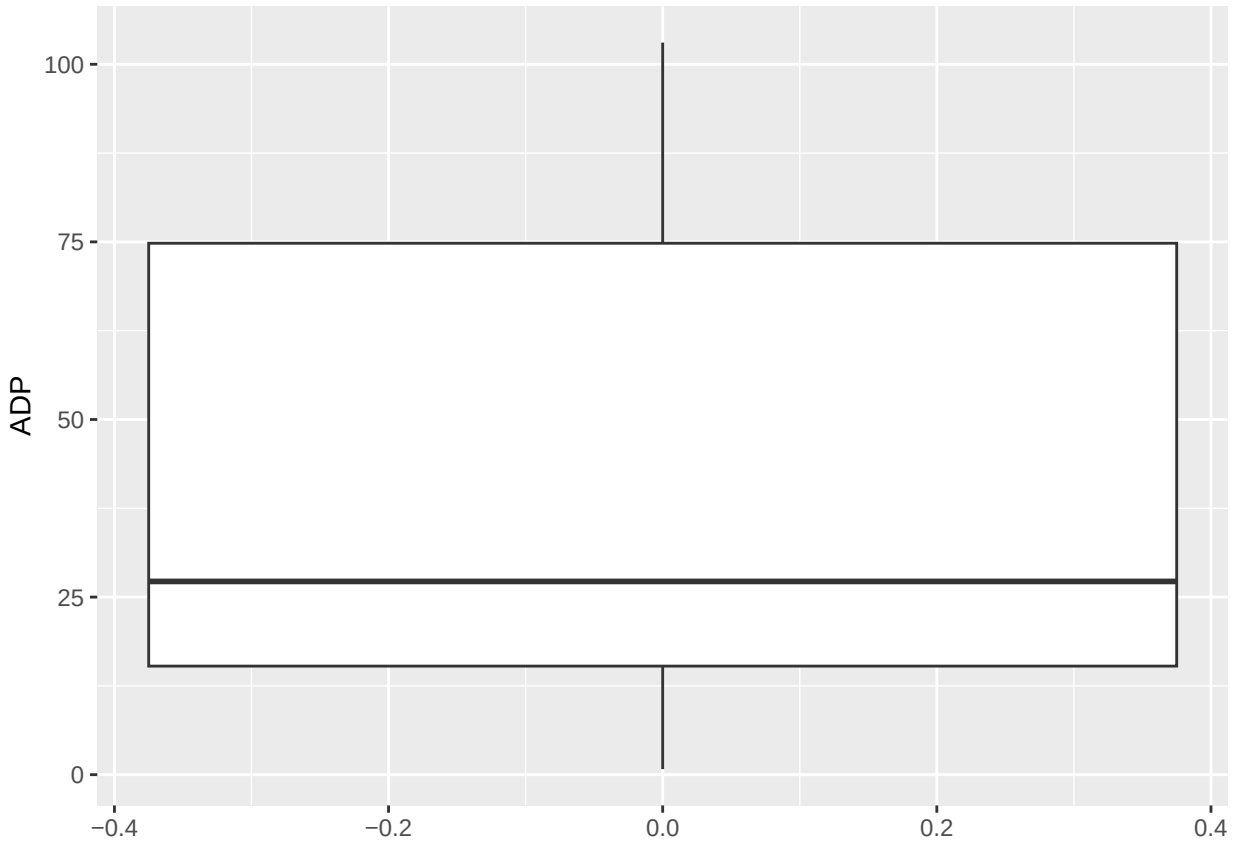
```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

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

```
# Checking clean data histogram
ggplot(data_clean, aes(x =ADP)) +
  geom_histogram(binwidth = 5, fill = "cyan", color = "black") +
  theme_minimal() +
  labs(title = "Histogram of CLEAN_ADP Levels", x = "ADP Level", y = "Frequency")
```




```
# Check a boxplot graph

ggplot(data_clean,aes(y= ADP)) + geom_boxplot()
```




```
#Statistical methods

# Check skewness
clean_skewness_value <- skewness(data_clean$ADP) #  right skewness # not appropriate to do Z score chec

# Identify outliers using IQR method  # appropriate way to Check outliers
Q1 <- quantile(data_clean$ADP, 0.25)
Q3 <- quantile(data_clean$ADP, 0.75)
IQR_value <- IQR(data_clean$ADP)
lower_bound <- Q1 - 1.5 * IQR_value
upper_bound <- Q3 + 1.5 * IQR_value
outliers_iqr <- data_clean %>%
  filter(ADP < lower_bound | ADP > upper_bound)   # There is no outliers that less than Q1 - 1.5 * IQR

cat("Number of outliers by IQR method:", nrow(outliers_iqr), "\n")  # IQR method is appropriate at this
```

```
## Number of outliers by IQR method: 0
```

Linear regression

In the third step, I normalize the ADP value from the previous step by taking the logarithm to achieve a normal distribution, which is essential for testing the linear regression.

```
data_clean$ADP_log <- log(data_clean$ADP)
```

Then, I test the regression with three SNPs individually by taking the log ADP. I also plot the graphs(ggplot and qqplot) to visualize the data distribution and pattern.

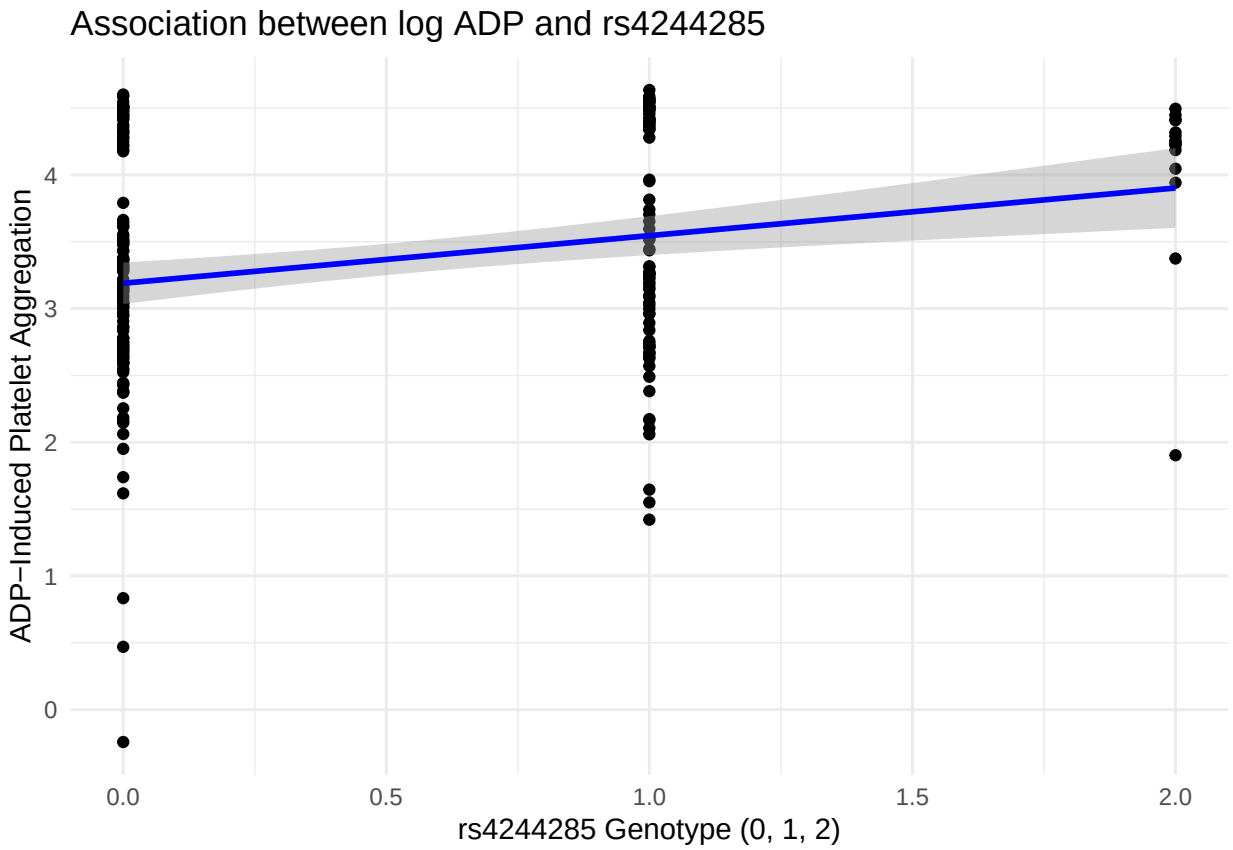
```r
liner_logA <- lm(ADP_log ~ rs4244285, data = data_clean)
liner_logB <- lm(ADP_log ~ rs4986893, data = data_clean)
linear_logC <- lm(ADP_log ~ rs662, data = data_clean)
```

```r
library(ggplot2)
summary(liner_logA)
```

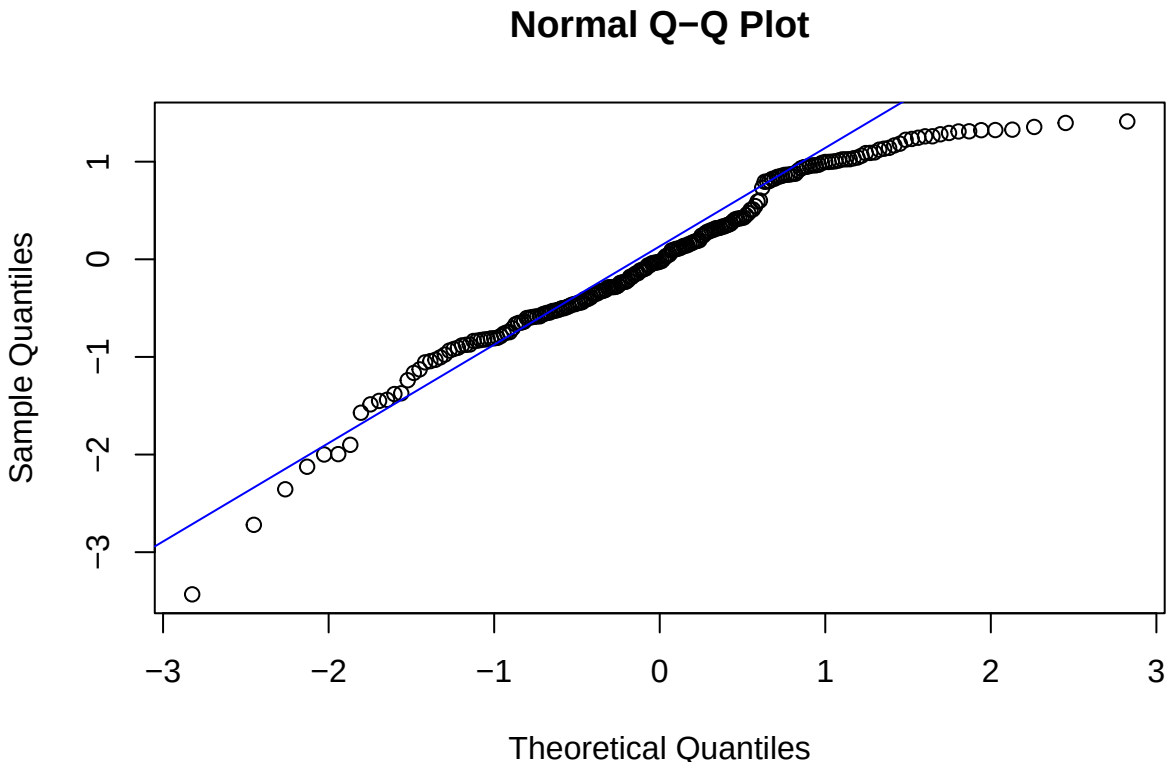
```
##
## Call:
## lm(formula = ADP_log ~ rs4244285, data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4317 -0.5467 -0.0235  0.8128  1.4120
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.18940    0.07873   40.51  < 2e-16 ***
## rs4244285    0.35644    0.09530    3.74 0.000238 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8651 on 209 degrees of freedom
## Multiple R-squared:  0.06273,    Adjusted R-squared:  0.05825
## F-statistic: 13.99 on 1 and 209 DF,  p-value: 0.0002375
```

```r
ggplot(data_clean, aes(x = rs4244285, y = ADP_log)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Association between log ADP and rs4244285",
       x = "rs4244285 Genotype (0, 1, 2)",
       y = "ADP-Induced Platelet Aggregation") +
  theme_minimal()
```

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




```r
qqnorm(liner_logA$residuals)
qqline(liner_logA$residuals, col = "blue")
```

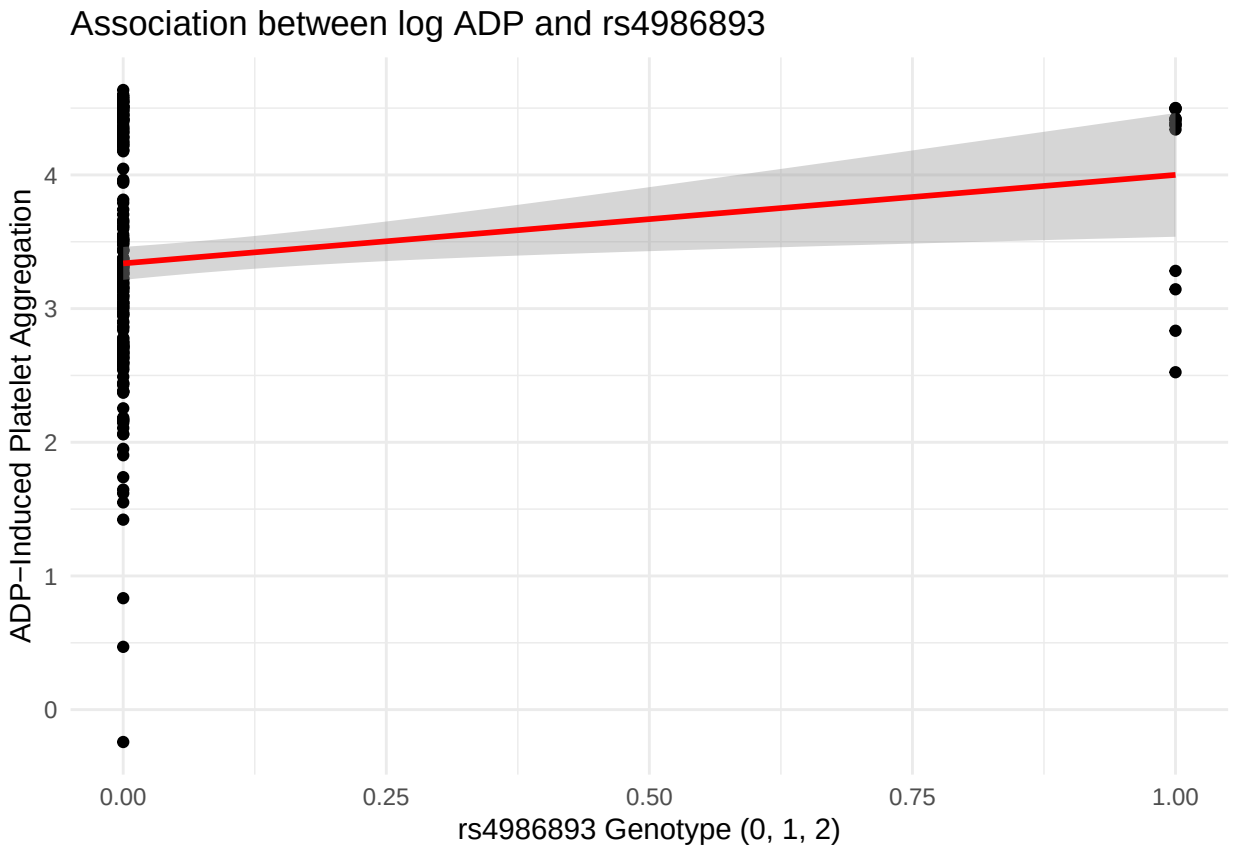


```r
library(ggplot2)
summary(liner_logB)
```

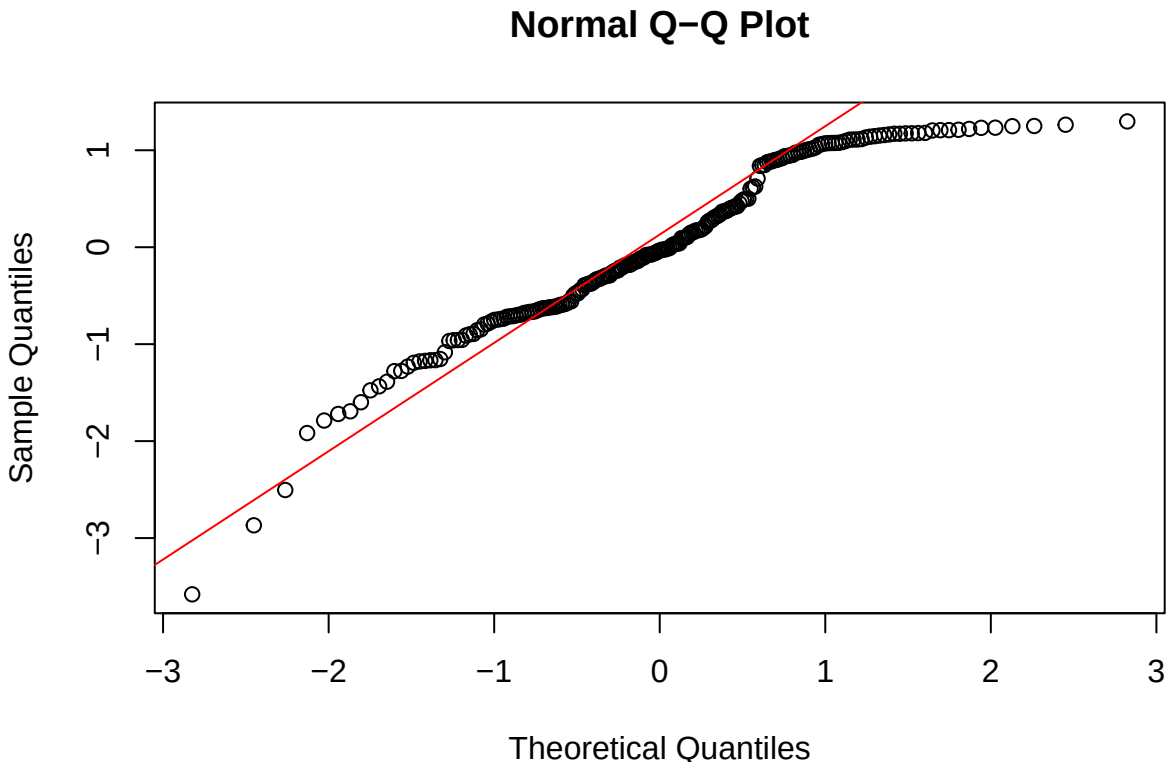
```
##
## Call:
## lm(formula = ADP_log ~ rs4986893, data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5803 -0.6227 -0.0348  0.8844  1.2972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.33804    0.06257  53.353  < 2e-16 ***
## rs4986893    0.66218    0.24289   2.726  0.00695 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8781 on 209 degrees of freedom
## Multiple R-squared:  0.03434,    Adjusted R-squared:  0.02972
## F-statistic: 7.433 on 1 and 209 DF,  p-value: 0.006949
```

```r
ggplot(data_clean, aes(x = rs4986893 , y = ADP_log)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Association between log ADP and rs4986893",
       x = "rs4986893 Genotype (0, 1, 2)",
       y = "ADP-Induced Platelet Aggregation") +
  theme_minimal()
```

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




```r
qqnorm(liner_logB$residuals)
qqline(liner_logB$residuals, col = "red")
```

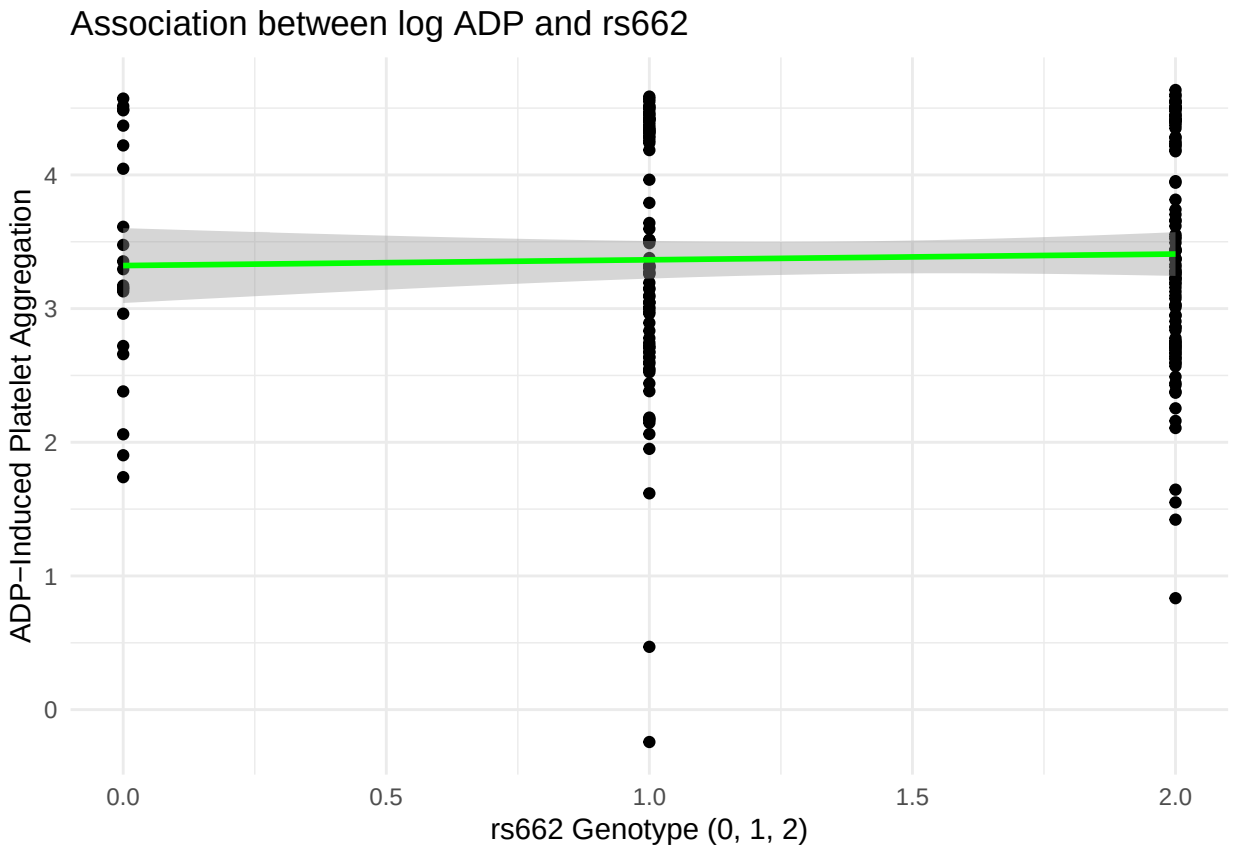


```r
summary(linear_logC)
```

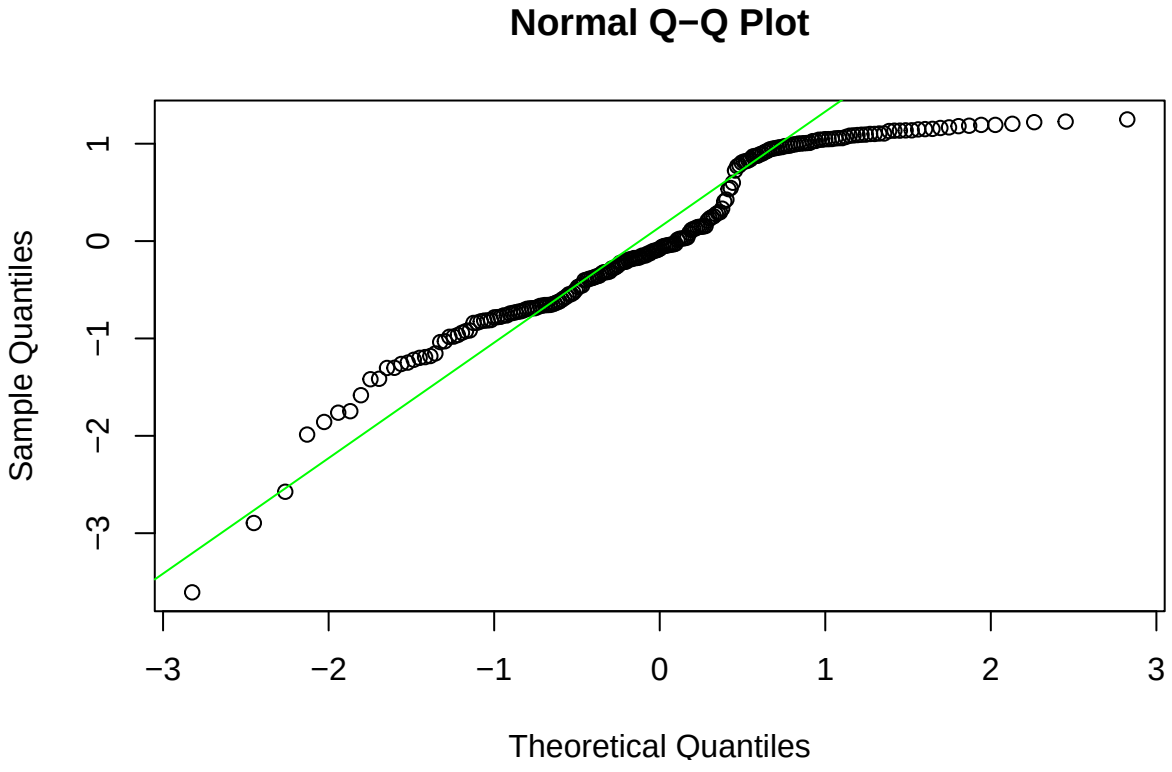
```
##
## Call:
## lm(formula = ADP_log ~ rs662, data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6073 -0.6565 -0.0787  0.9437  1.2492
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.32192    0.14211  23.376   <2e-16 ***
## rs662        0.04310    0.09195   0.469     0.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8932 on 209 degrees of freedom
## Multiple R-squared:  0.00105,    Adjusted R-squared:  -0.003729
## F-statistic: 0.2197 on 1 and 209 DF,  p-value: 0.6397
```

```r
ggplot(data_clean, aes(x = rs662 , y = ADP_log)) +
  geom_point() +
  geom_smooth(method = "lm", color = "green") +
  labs(title = "Association between log ADP and rs662",
       x = "rs662 Genotype (0, 1, 2)",
       y = "ADP-Induced Platelet Aggregation") +
  theme_minimal()
```

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




```r
qqnorm(linear_logC$residuals)
qqline(linear_logC$residuals, col = "green")
```



For more complex details, I adjust the test value by adding sex and age in each model to get the summary data.

```
snp_list <- c("rs4244285", "rs4986893", "rs662")

results_list <- list()

for (snp in snp_list) {
  model_sum <- lm(as.formula(paste("ADP_log ~ AGE + SEX +", snp)), data = data_clean)
  results_list[[snp]] <- summary(model_sum)
}
print(results_list[["rs4244285"]]) # significant only SNP
```

```
##
## Call:
## lm(formula = as.formula(paste("ADP_log ~ AGE + SEX +", snp)),
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4329 -0.5847  0.0234  0.7691  1.3790
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.642427   0.369440   9.859  < 2e-16 ***
## AGE         -0.006644   0.005629  -1.180 0.239184
## SEX         -0.047234   0.134027  -0.352 0.724883
## rs4244285    0.360830   0.095387   3.783 0.000203 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8652 on 207 degrees of freedom
## Multiple R-squared:  0.07152,    Adjusted R-squared:  0.05806
## F-statistic: 5.315 on 3 and 207 DF,  p-value: 0.001507
```

```
print(results_list[["rs4986893"]])# significant only SNP
```

```
##
## Call:
## lm(formula = as.formula(paste("ADP_log ~ AGE + SEX +", snp)),
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5707 -0.5926 -0.0460  0.8435  1.3307
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.787300   0.372706  10.162  < 2e-16 ***
## AGE         -0.006749   0.005720  -1.180  0.23941
## SEX         -0.006994   0.136411  -0.051  0.95916
## rs4986893    0.663689   0.243691   2.723  0.00701 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.879 on 207 degrees of freedom
## Multiple R-squared:  0.04167,    Adjusted R-squared:  0.02778
## F-statistic:     3 on 3 and 207 DF,  p-value: 0.03161
```

```
print(results_list[["rs662"]]) # not siginifcant all variables
```

```
##
## Call:
## lm(formula = as.formula(paste("ADP_log ~ AGE + SEX +", snp)),
##     data = data_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6018 -0.6330 -0.0758  0.9179  1.2493
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.758708   0.392094   9.586   <2e-16 ***
## AGE         -0.006572   0.005838  -1.126    0.262
## SEX         -0.024108   0.139273  -0.173    0.863
## rs662        0.047704   0.092747   0.514    0.608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8941 on 207 degrees of freedom
## Multiple R-squared:  0.008599,   Adjusted R-squared:  -0.005769
## F-statistic: 0.5985 on 3 and 207 DF,  p-value: 0.6167
```

From linearA and linearB, rs42244285 and rs4986893 are significantly associated with ADP-platelet aggregation (P value < 0.0017, $0.05/3$), while other confounding variables are not significant predictors. However, the R square values are low, but they are still acceptable.

In linearC, all variables are not statistically significant to ADP (P value > 0.05). Additionally, the R-squared is negative, indicating the model cannot predict or explain the data's variability.

Logistic regression

In the fourth step, I use the dependent value as Clopidogrel resistance (binary outcomes) with three SNPs as independent values, age, and sex as confounding factors to test logistic regression. Then, I plot the graphs(histogram,ggplot for prediction, and qqplot) and interpret them using the summary() package similar to linear regression steps.

```
library(ggplot2) # Clopidogrel resistance = 1  0 = not resistance
# Logistic regression between resistance and SNPs.
logistic_A <- glm(Resistance ~ rs4244285, data= data_clean, family = binomial)

logistic_B <- glm(Resistance ~ rs4986893, data= data_clean, family = binomial)

logistic_C <- glm(Resistance ~ rs662, data= data_clean, family = binomial)
```

```
# Summary logistic A
summary(logistic_A)
```

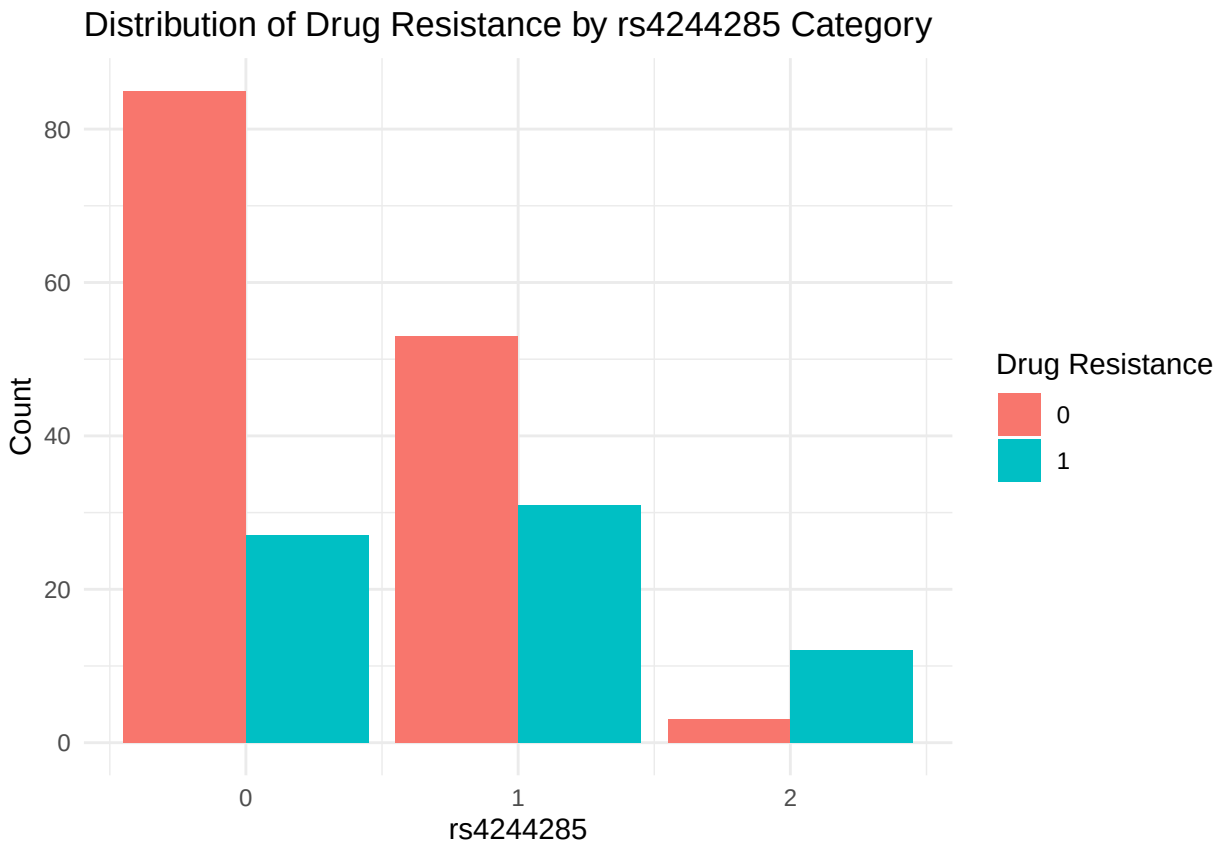
```
##
## Call:
## glm(formula = Resistance ~ rs4244285, family = binomial, data = data_clean)
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2583     0.2165  -5.813 6.13e-09 ***
## rs4244285     0.9433     0.2436   3.872 0.000108 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 252.16  on 209  degrees of freedom
## AIC: 256.16
##
## Number of Fisher Scoring iterations: 4
```

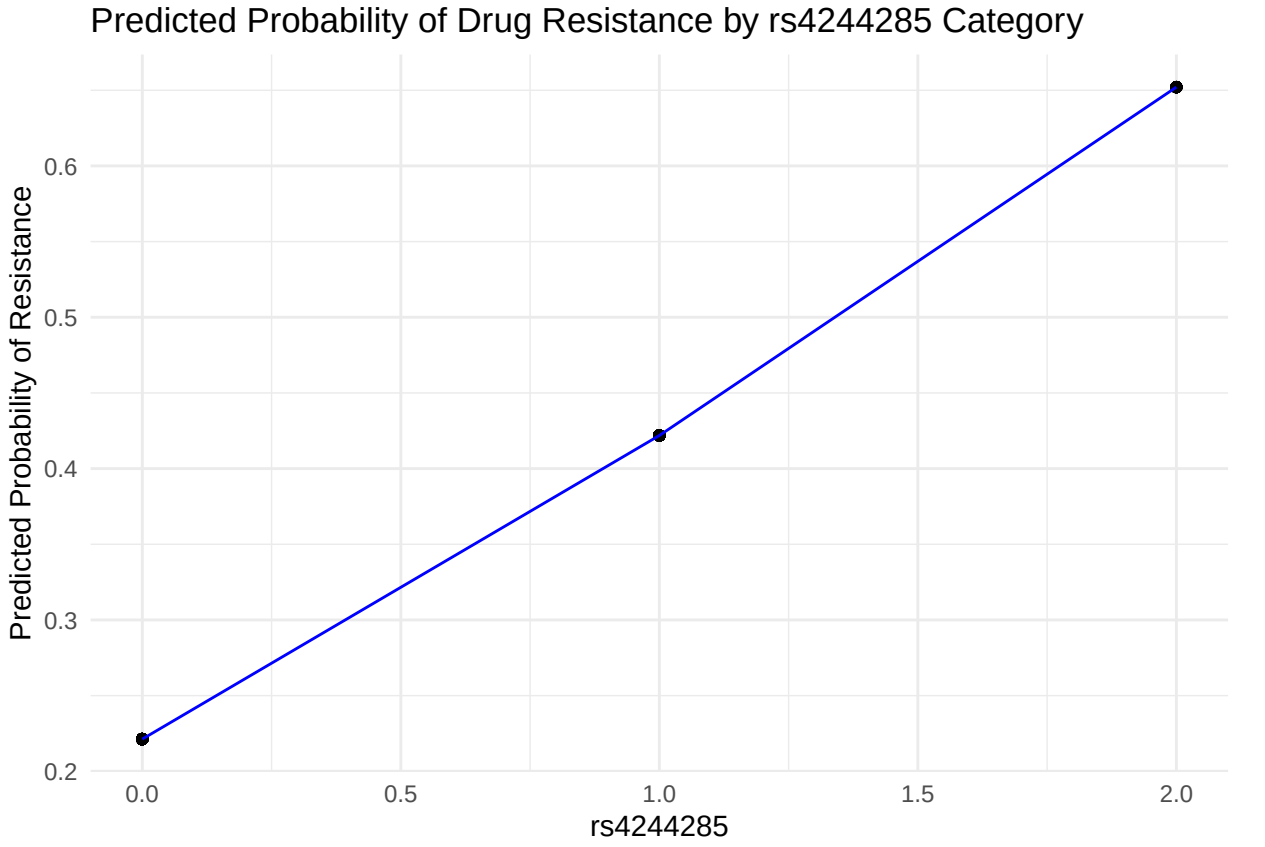
```r
exp(coef(logistic_A))
```

```
## (Intercept)   rs4244285
##   0.2841265   2.5683454
```

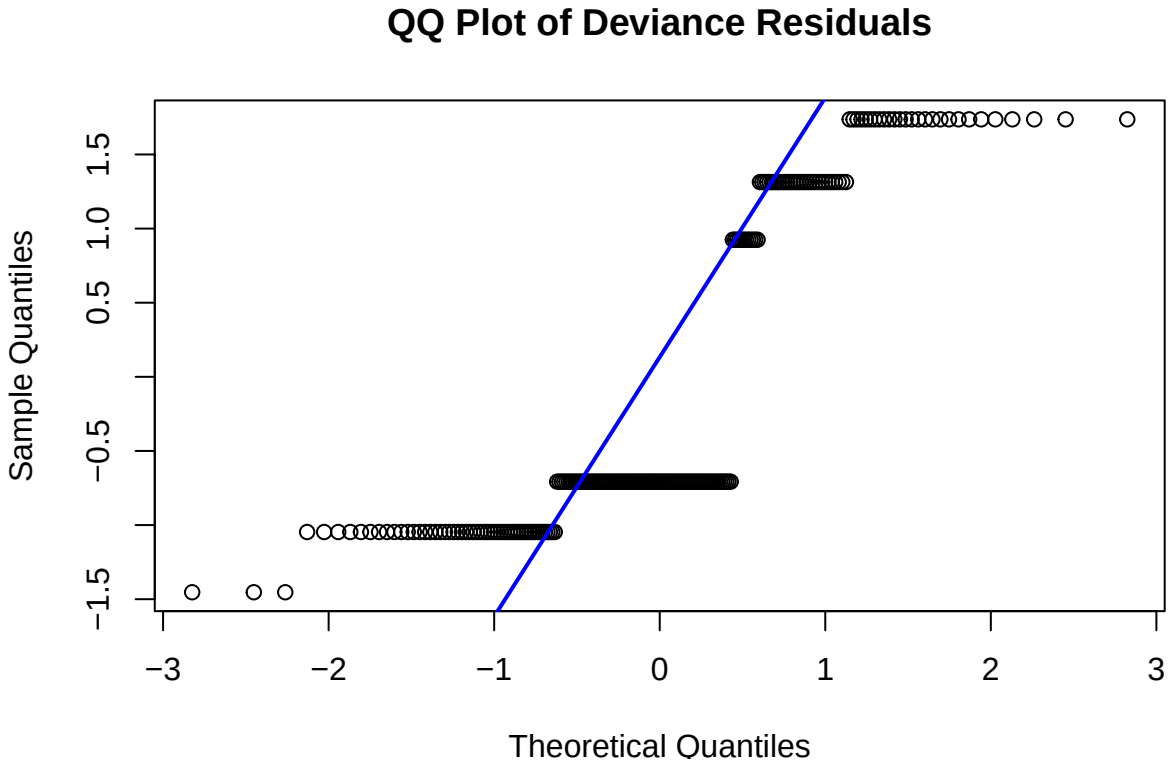
```r
# Histogram
# 0 = GG ,1 = AG ,2 = AA

ggplot(data_clean, aes(x = rs4244285, fill = factor(Resistance))) +
  geom_bar(position = "dodge") +
  labs(x = "rs4244285", y = "Count", fill = "Drug Resistance",
       title = "Distribution of Drug Resistance by rs4244285 Category") +
  theme_minimal()
```



```r
# Predict probabilities of resistance

data_clean$predicted_probA <- predict(logistic_A, type = "response")

# Plot the predicted probabilities
ggplot(data_clean, aes(x =rs4244285, y = predicted_probA)) +
  geom_point() +
  geom_line(aes(group = 1), color = "blue") +
  labs(x = "rs4244285", y = "Predicted Probability of Resistance",
       title = "Predicted Probability of Drug Resistance by rs4244285 Category") +
  theme_minimal()
```




```r
# Test q-q plot
residuals_devianceA <- residuals(logistic_A , type ="deviance")

qqnorm(residuals_devianceA, main = "QQ Plot of Deviance Residuals")
qqline(residuals_devianceA, col = "blue", lwd = 2)
```



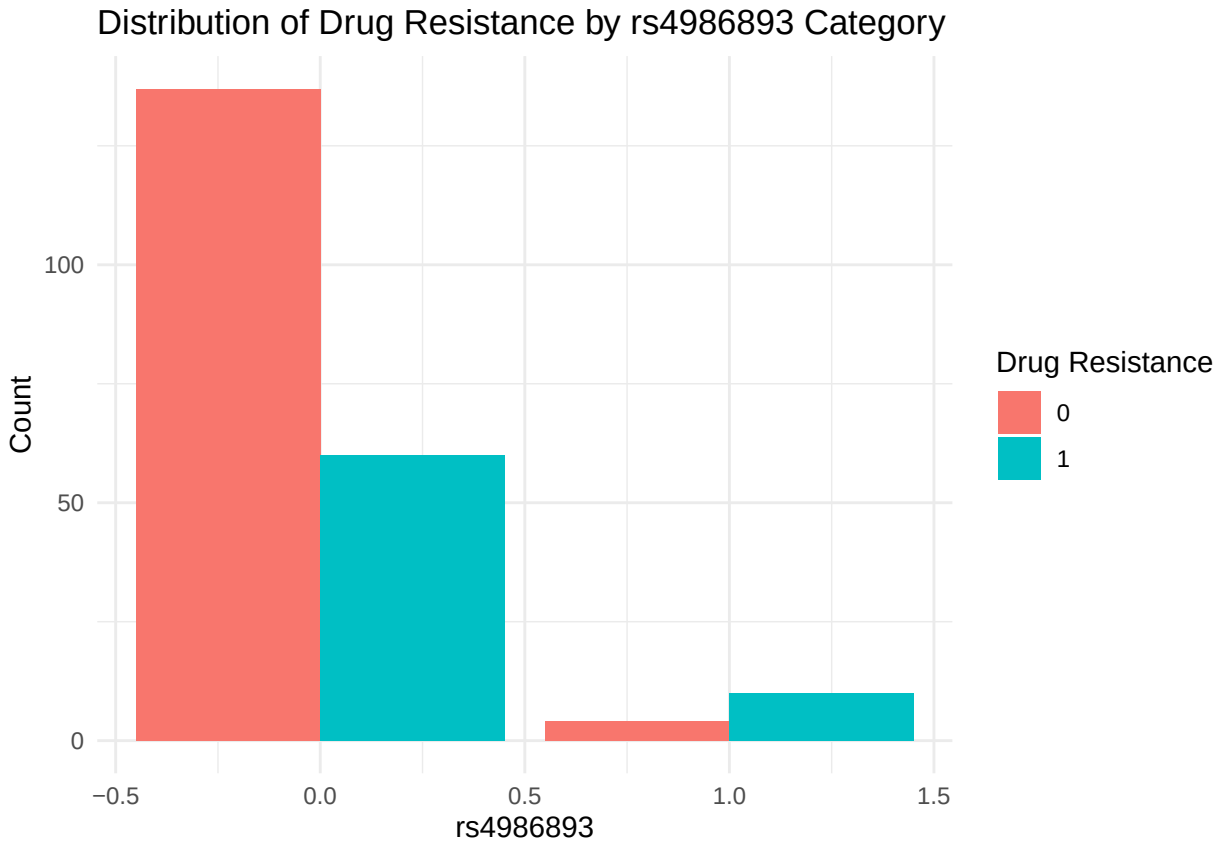

```
#Summary logistic B

# GG = 2 AG =1 AA =0

summary(logistic_B)
```

```
##
## Call:
## glm(formula = Resistance ~ rs4986893, family = binomial, data = data_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8256     0.1548  -5.333 9.65e-08 ***
## rs4986893     1.7419     0.6115   2.848  0.00439 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 258.94  on 209  degrees of freedom
## AIC: 262.94
##
## Number of Fisher Scoring iterations: 4
```

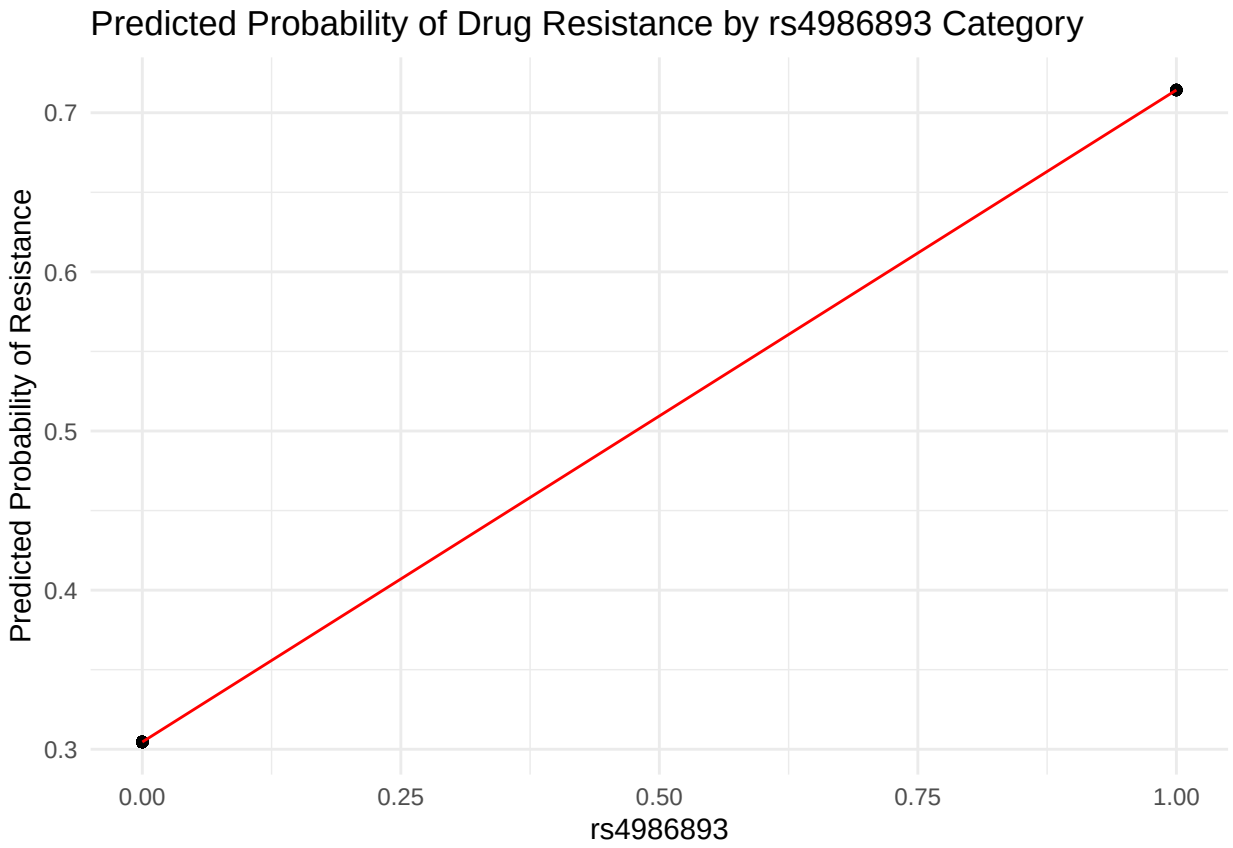
```
exp(coef(logistic_B))
```

```
## (Intercept)   rs4986893
##   0.4379562   5.7083333
```

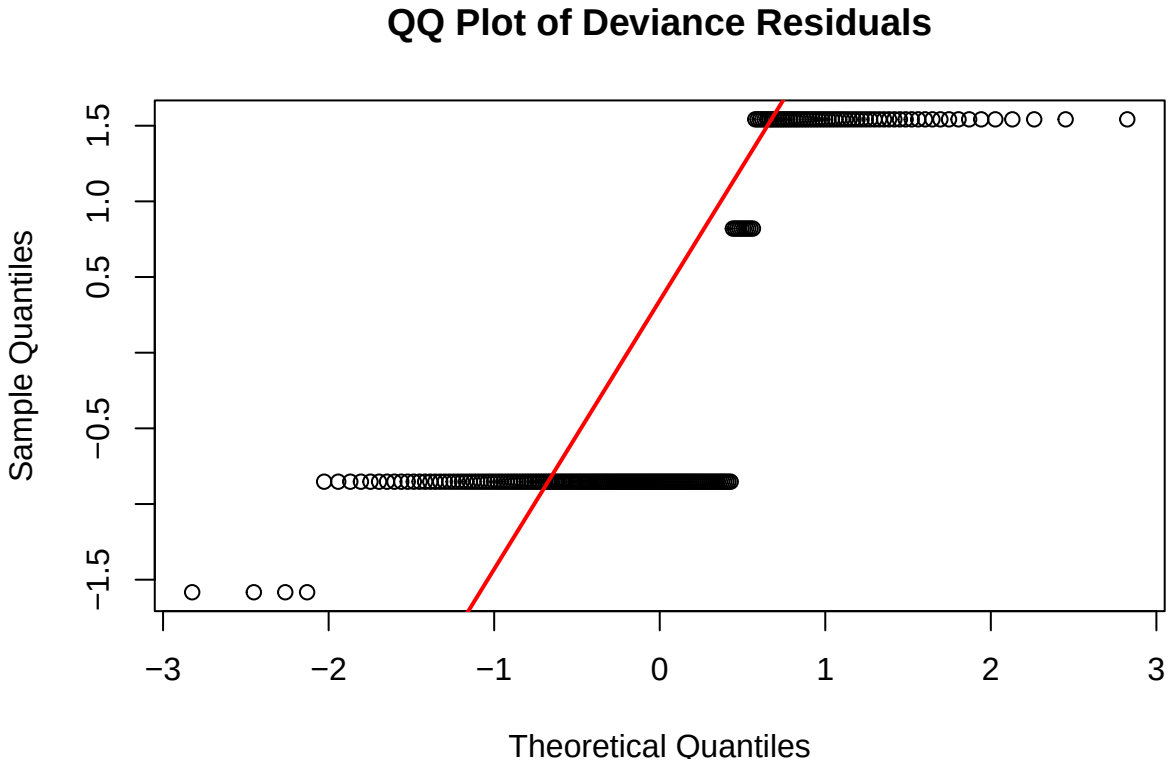
```r
#Histogram
ggplot(data_clean, aes(x = rs4986893, fill = factor(Resistance))) +
  geom_bar(position = "dodge") +
  labs(x = "rs4986893", y = "Count", fill = "Drug Resistance",
       title = "Distribution of Drug Resistance by rs4986893 Category") +
  theme_minimal()
```




```r
#Predict probabilities of resistance

data_clean$predicted_probB <- predict(logistic_B, type = "response")

ggplot(data_clean, aes(x =rs4986893, y = predicted_probB)) +
  geom_point() +
  geom_line(aes(group = 1), color = "red") +
  labs(x = "rs4986893", y = "Predicted Probability of Resistance",
       title = "Predicted Probability of Drug Resistance by rs4986893 Category") +
  theme_minimal()
```




```r
# Test q-q plot

residuals_devianceB <- residuals(logistic_B , type ="deviance")

qqnorm(residuals_devianceB, main = "QQ Plot of Deviance Residuals")
qqline(residuals_devianceB, col = "red", lwd = 2)
```




```
#Summary logistic C
# GG=2 AG =1 AA =0
summary(logistic_C)
```

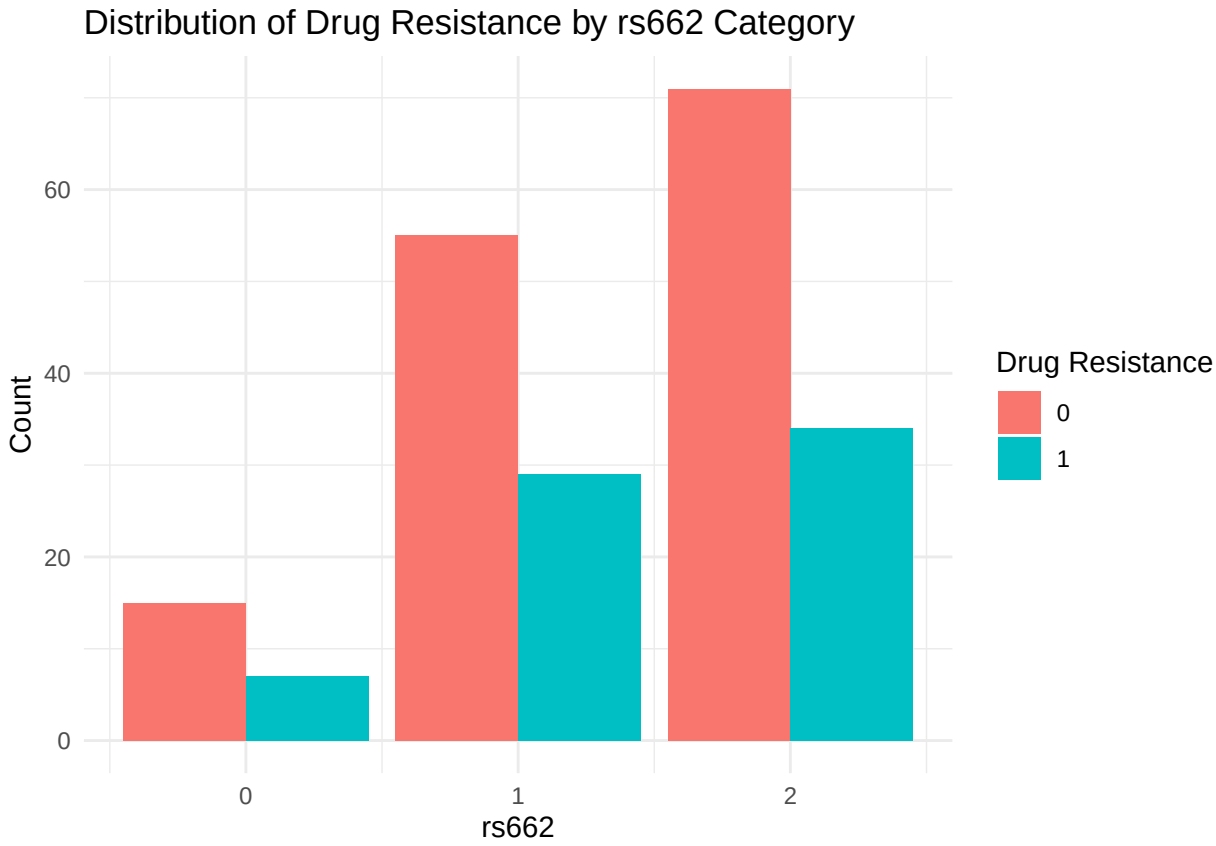
```
##
## Call:
## glm(formula = Resistance ~ rs662, family = binomial, data = data_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.66470    0.33669  -1.974   0.0484 *
## rs662       -0.02556    0.21826  -0.117   0.9068
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 268.13  on 209  degrees of freedom
## AIC: 272.13
##
## Number of Fisher Scoring iterations: 4
```

```
exp(coef(logistic_C))
```

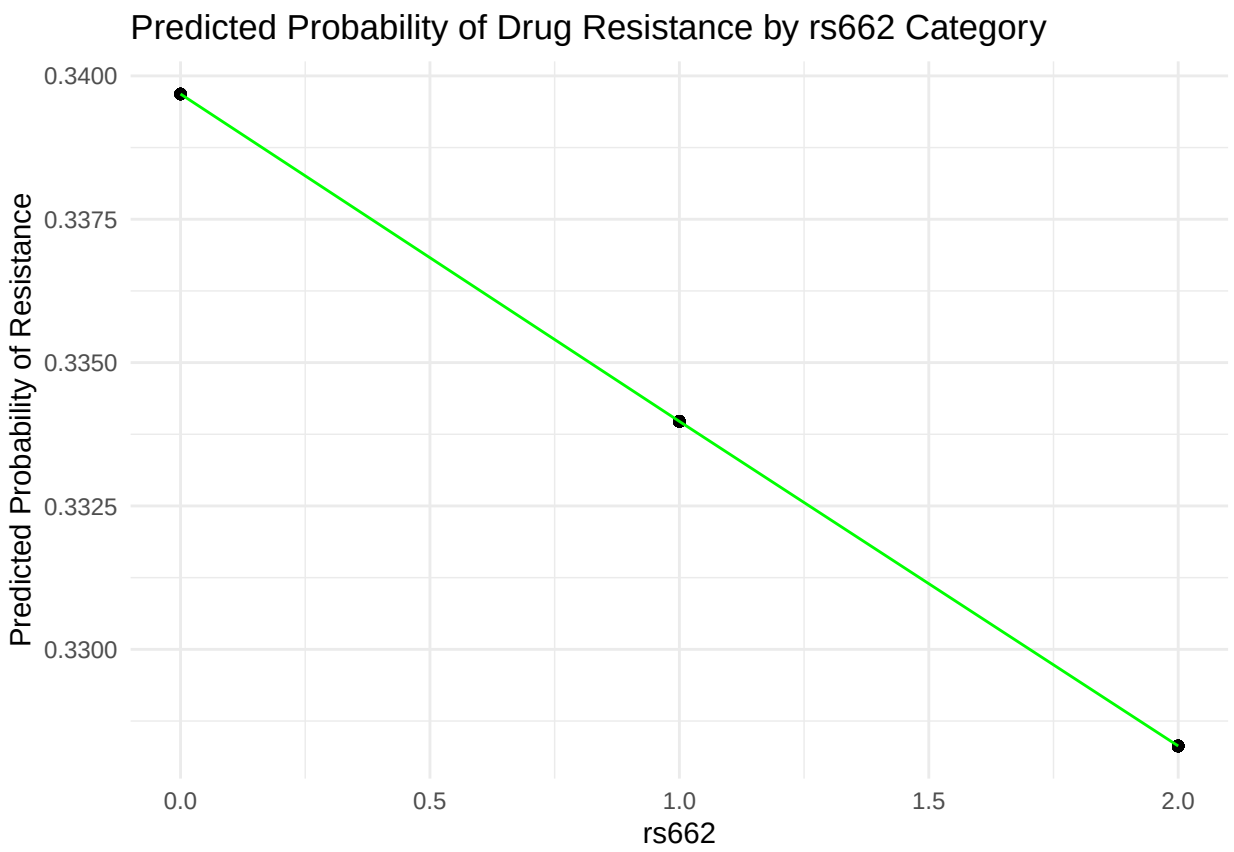
```
## (Intercept)       rs662
##   0.5144260   0.9747669
```

```
#Histogram
ggplot(data_clean, aes(x = rs662, fill = factor(Resistance))) +
  geom_bar(position = "dodge") +
```

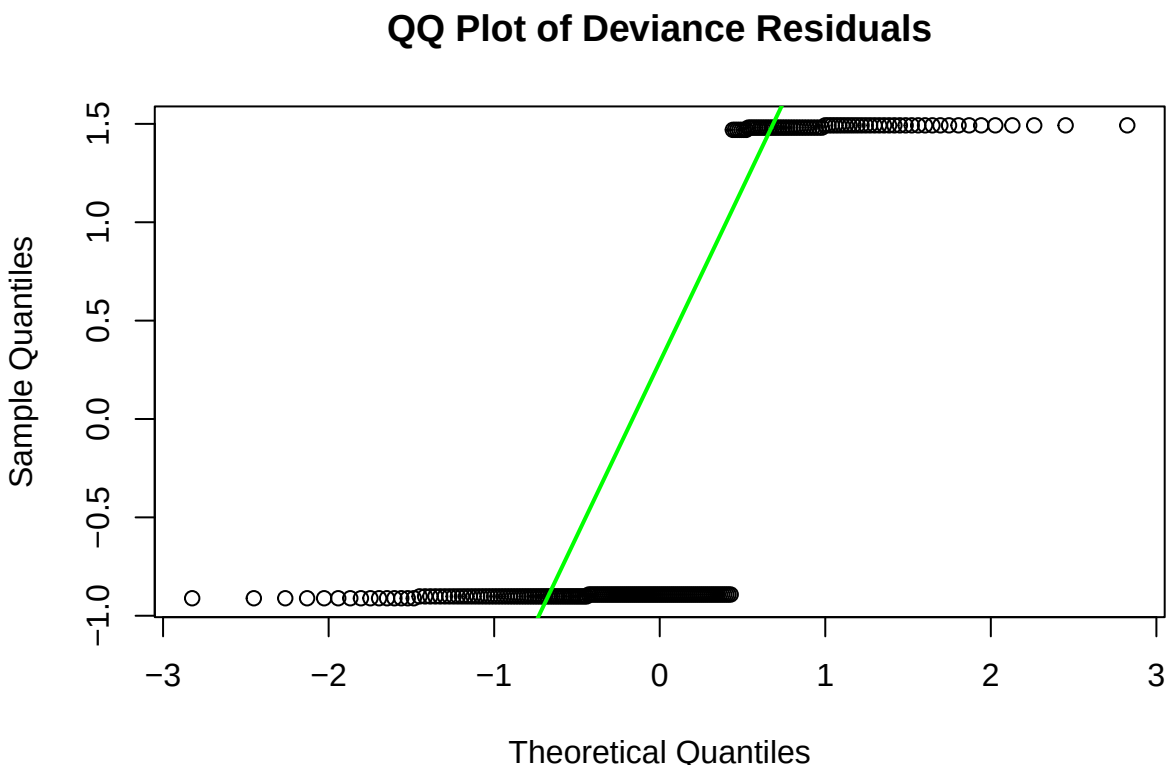
```
  labs(x = "rs662", y = "Count", fill = "Drug Resistance",
       title = "Distribution of Drug Resistance by rs662 Category") +
  theme_minimal()
```




```
#Predict probabilities of resistance

data_clean$predicted_probC <- predict(logistic_C, type = "response")

ggplot(data_clean, aes(x =rs662, y = predicted_probC)) +
  geom_point() +
  geom_line(aes(group = 1), color = "green") +
  labs(x = "rs662", y = "Predicted Probability of Resistance",
       title = "Predicted Probability of Drug Resistance by rs662 Category") +
  theme_minimal()
```




```r
# Test q-q plot

residuals_devianceC <- residuals(logistic_C , type ="deviance")

qqnorm(residuals_devianceC, main = "QQ Plot of Deviance Residuals")
qqline(residuals_devianceC, col = "green", lwd = 2)
```




```
# Adding Confounding  for control bias or adjusting for covariates.

fulll_logistic_A <- glm(Resistance ~ rs4244285 + AGE + SEX, data= data_clean, family = binomial) # rs42

fulll_logistic_B <- glm(Resistance ~ rs4986893 + AGE + SEX , data= data_clean, family = binomial) # rs4

fulll_logistic_C <- glm(Resistance ~ rs662 +AGE +SEX , data= data_clean, family = binomial) # not stati

# summary each full_logistics

summary(fulll_logistic_A)
```

```
##
## Call:
## glm(formula = Resistance ~ rs4244285 + AGE + SEX, family = binomial,
##     data = data_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.35306    0.94235   0.375   0.7079
## rs4244285    0.97318    0.24693   3.941 8.11e-05 ***
## AGE         -0.02451    0.01463  -1.675   0.0938 .
## SEX         -0.05410    0.34977  -0.155   0.8771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 248.82  on 207  degrees of freedom
```

```
## AIC: 256.82
##
## Number of Fisher Scoring iterations: 4
```

```
summary(fulll_logistic_B)
```

```
##
## Call:
## glm(formula = Resistance ~ rs4986893 + AGE + SEX, family = binomial,
##     data = data_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.73079    0.92024   0.794  0.42712
## rs4986893    1.78749    0.61864   2.889  0.00386 **
## AGE         -0.02411    0.01437  -1.678  0.09335 .
## SEX          0.08354    0.34338   0.243  0.80779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 255.98  on 207  degrees of freedom
## AIC: 263.98
##
## Number of Fisher Scoring iterations: 4
```

```
summary(fulll_logistic_C)
```

```
##
## Call:
## glm(formula = Resistance ~ rs662 + AGE + SEX, family = binomial,
##     data = data_clean)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.727160   0.927671   0.784    0.433
## rs662       -0.004631   0.222195  -0.021    0.983
## AGE         -0.021617   0.014006  -1.543    0.123
## SEX          0.005561   0.336527   0.017    0.987
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 268.14  on 210  degrees of freedom
## Residual deviance: 265.46  on 207  degrees of freedom
## AIC: 273.46
##
## Number of Fisher Scoring iterations: 4
```

From statistical values and visualization of the graph, It is clear that logisticA and logisticB for rs44244285, and rs496893 have a statistically significant between Drug resistance because P value < 0.05 at 8.11e-05 and 0.00386 respectively. In contrast, logistcC does not have a significant value at 0.983(P value>0.05) like the confounding factors(age, and sex). for every model. Logistic A and B have lower residual deviance than the null deviance, indicating they are the best-fitting models for testing the association. Logistic C has almost the same meaning, suggesting it doesn't fit.

To summarize, all of data that obtained from every test

- ADP statistical test.
  - Have moderate right skewness distribution
  - No outliers were detected based on IQR test.
- Linear regression
  - The polymorphisms of CYP2C19 (rs42244285 and rs4988693) loci increase the ADP level.
  - The polymorphisms of PON.192Q>R does not have a significant effect on ADP level.
- Logistic regression
  - Clopidogrel resistance is significantly increased by polymorphisms of CYP2C19.
  - The polymorphisms of PON.192Q>R does not play a significant role on drug resistance.
- Additional
  - Age and sex do not have significant effect on both ADP level and Clopidogrel resistance.