Solution_HW2

MSBA7002: Business Statistics

Abstract

The homework solutions contain a continued effort from the MSBA7002 TAs and instructors[1]. You should compare the solutions with your own homework submission to see how you can better improve your analytical skills. Please **do not redistribute the document or put it online** to hurt other students' learning experience.

Contents

[1]We would like to thank Jianlong Shao for providing the first version of the solution and for his great contributions to the class

1 Q1. Default Data from ISLR

```
df.default <- ISLR::Default
df.default <- df.default %>% mutate(default = factor(default, levels = c("No","Yes")))
str(df.default)
```

```
## 'data.frame':    10000 obs. of  4 variables:
##  $ default: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ student: Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 2 1 1 ...
##  $ balance: num  730 817 1074 529 786 ...
##  $ income : num  44362 12106 31767 35704 38463 ...
```

```
summary(df.default)
```

```
##  default    student       balance           income
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
##                        Median : 823.6   Median :34553
##                        Mean   : 835.4   Mean   :33517
##                        3rd Qu.:1166.3   3rd Qu.:43808
##                        Max.   :2654.3   Max.   :73554
```

1.1

Fit a logistic regression with student as the X variable and default as the response variable. Interpret the coefficients and discuss whether the X variable is significant.

```
fit1.1 <- glm(default~student,data = df.default, family = binomial(logit))
summary(fit1.1)
```

```
##
## Call:
## glm(formula = default ~ student, family = binomial(logit), data = df.default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2970  -0.2970  -0.2434  -0.2434   2.6585
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
## studentYes   0.40489    0.11502    3.52 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2908.7  on 9998  degrees of freedom
## AIC: 2912.7
##
## Number of Fisher Scoring iterations: 6
```

The coefficient of X means: the log odds of Y is equal to default will increase 0.4049 if a person is student compared to a person is not student, The interecpt means the log odds of Y given a person is not a student.

Both of estimators are **significant**.

1.2

For the above logistic regression, one can actually obtain explicit expression of the maximum likelihood estimates of the coefficients. Please do the following

1.2.1

Write down the logistic regression model by coding student using one dummy variable: 0 for students and 1 for non-students.

$$log(\frac{P(Default = Yes|Student)}{P(Default = No|Student)}) = \beta_0 + \beta_1 * Student$$

where

$$Student = \begin{cases} 0 & \text{student} \\ 1 & \text{non-student} \end{cases}$$

or

$$P(Default = Yes|Student = 1) = \frac{e^{\beta_0+\beta_1}}{1 + e^{\beta_0+\beta_1}}$$
$$P(Default = No|Student = 1) = \frac{1}{1 + e^{\beta_0+\beta_1}}$$
$$P(Default = Yes|Student = 0) = \frac{e^{\beta_0}}{1 + e^{\beta_0}}$$
$$P(Default = No|Student = 0) = \frac{1}{1 + e^{\beta_0}}$$

1.2.2

Write down the corresponding likelihood function.

$$L(\beta_0, \beta_1|\text{Default, Student}) = (\frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}})^{n(Default=Yes,Student=1)} \times (\frac{1}{1+e^{\beta_0+\beta_1}})^{n(Default=No,Student=1)} \times (\frac{e^{\beta_0}}{1+e^{\beta_0}})^{n(Default=Yes,Student=0)} \times (\frac{1}{1+e^{\beta_0}})^{n(Default=Yes,Student=0)}$$

or

$$\begin{aligned} log(\frac{p_i}{1-p_i}) &= \beta_0 + \beta_1 Student_i \\ L(\beta_0, \beta_1|\text{Default, Student}) &= \prod_{i=1}^{n} p_i^{Default_i}(1-p_i)^{1-Default_i} \\ &= e^{\sum_{i=1}^{n}[Default_i(\beta_0+\beta_1 Student_i) - log(1+e^{\beta_0+\beta_1 Student_i})]} \\ l(\beta_0, \beta_1|\text{Default, Student}) &= \sum_{i}^{n}[Default_i(\beta_0 + \beta_1 Student_i) - log(1+e^{\beta_0+\beta_1 Student_i})] \end{aligned}$$

where

$$Default = \begin{cases} 1 & \text{Yes} \\ 0 & \text{No} \end{cases}$$

$$Student = \begin{cases} 0 & \text{student} \\ 1 & \text{non-student} \end{cases}$$

1.2.3

Obtain expressions for the coefficient estimates, and compare them with the answers in Q1.1.

We first define $n_d = \sum_i Default_i$, $n_s = \sum_i Student_i$ and $n_{ds} = \sum_i Default_i * Student_i$ as the total number of defaults, total number of students and total number of students with default = Yes. With these defination, we rewrite the log-likelihood as

$$\begin{aligned} l(\beta_0, \beta_1 | \text{Default}, \text{Student}) &= \sum_i^n [Default_i(\beta_0 + \beta_1 Student_i) - \log(1 + e^{\beta_0 + \beta_1 Student_i})] \\ &= n_d\beta_0 + n_{ds}\beta_1 - n_s \log(1 + e^{\beta_0+\beta_1}) - (n - n_s)\log(1 + e^{\beta_0}) \end{aligned}$$

In order to maximize the log-likelihood presented above, we take derivative with respect to β_0 and β_1. So the following two equations need to be satisfied.

$$\begin{aligned} n_d - n_s \frac{1}{1 + e^{-(\beta_0+\beta_1)}} - (n - n_s)\frac{1}{1 + e^{-\beta_0}} &= 0 \\ n_{ds} - n_s \frac{1}{1 + e^{-(\beta_0+\beta_1)}} &= 0 \end{aligned}$$

Solving the equations, we get our MLE for $\hat{\beta}_0$ and $\hat{\beta}_1$

$$\begin{aligned} \hat{\beta}_0 &= -\log(\frac{n - n_s}{n_d - n_{ds}} - 1) \\ \hat{\beta}_1 &= -\log(\frac{n_s}{n_{ds}} - 1) - \hat{\beta}_0 \end{aligned}$$

```
n <- nrow(df.default)
n_d <- sum(df.default$default == "Yes")
n_s <- sum(df.default$student == "Yes")
n_ds <- sum((df.default$student == "Yes") & (df.default$default == "Yes"))

beta0 <- -log((n - n_s)/(n_d - n_ds) - 1)
beta1 <- -log(n_s / n_ds - 1) - beta0
c(beta0, beta1)
```

```
## [1] -3.5041278  0.4048871
```

The computation matches exactly with R output.

1.3

1.3.1

Consider all the variables and obtain the final logistic regression model.

```
fit1.3 <- glm(default~., data=df.default, family = binomial)
summary(fit1.3)
```

```
##
## Call:
## glm(formula = default ~ ., family = binomial, data = df.default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

1.3.2

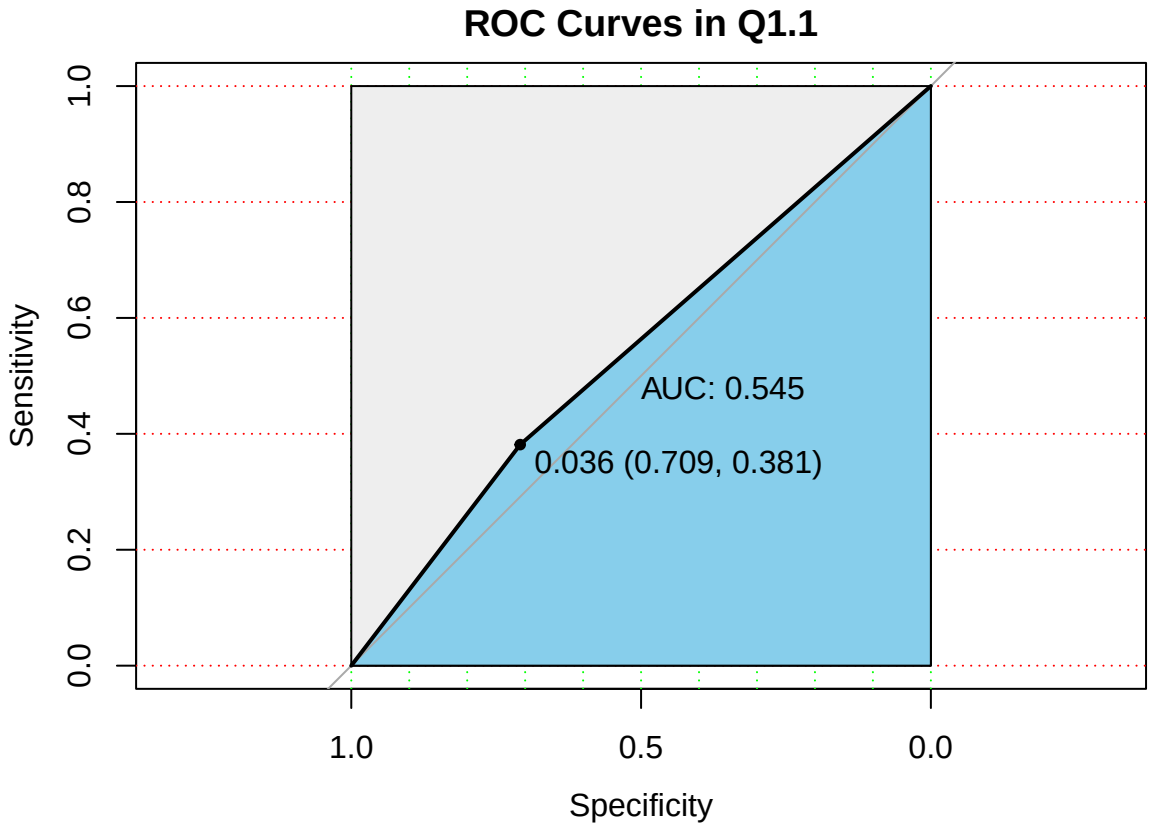
Compare the ROC curves for the model in Q1.1 and Q1.3, along with the corresponding AUC.

```
roc_1.1 <- roc(df.default$default,fit1.1$fitted.values)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_1.1, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),
     grid.col=c("green", "red"), max.auc.polygon=TRUE,
     auc.polygon.col="skyblue", print.thres=TRUE, main = 'ROC Curves in Q1.1',
     control = FALSE, case = TRUE)
```

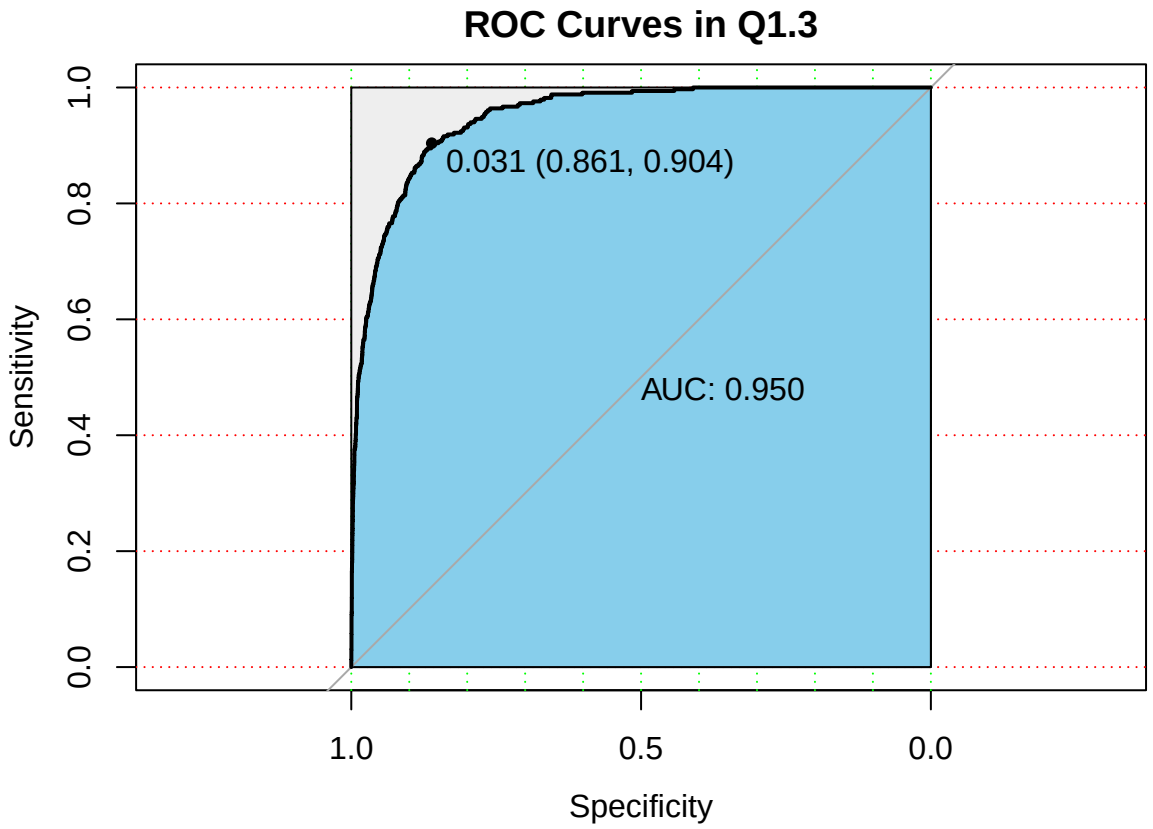


```
roc_1.3 <- roc(df.default$default,fit1.3$fitted.values)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
plot(roc_1.3, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),
     grid.col=c("green", "red"), max.auc.polygon=TRUE,
     auc.polygon.col="skyblue", print.thres=TRUE,main='ROC Curves in Q1.3')
```



We find the AUC of ROC in Q1.3 is much larger than that in Q1.1.

Note: although AUC in Q1.3 is much larger than that in Q1.1, we could not conclude logistic model in Q1.3 is better than that in Q1.1, since this auc is related to training dataset, it reflects over-fitting situation.

1.3.3

Consider a threshold of 0.5 on the probability of default. Calculate the corresponding 'specificity', 'sensitivity', 'false positive rate', 'true positive rate'.

```
fitted.default <- factor(ifelse(fit1.3$fitted.values>0.5,'Yes','No'))
t.def <- table(fitted.default,df.default$default)
t.def
```

```
##
## fitted.default   No  Yes
##            No  9627  228
##            Yes   40  105
```

$$
\begin{aligned}
Sensitivity &= Prob(\widehat{default} = Yes | default = Yes) \\
&= \frac{105}{105 + 228} = 31.53\% \\
Specificity &= Prob(\widehat{default} = No | default = No) \\
&= \frac{9627}{9627 + 40} = 99.59\% \\
\text{False Positive rate} &= Prob(\widehat{default} = Yes | default = No) \\
&= \frac{40}{9627 + 40} = 0.4138\% \\
\text{True Positive rate} &= Prob(\widehat{default} = Yes | default = Yes) \\
&= \frac{105}{105 + 228} = 31.53\%
\end{aligned}
$$

or

```
fitted.default <- factor(ifelse(fit1.3$fitted.values>0.5,'Yes','No'))
confusionMatrix(fitted.default,df.default$default,positive = "Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  9627  228
##        Yes   40  105
##
##                Accuracy : 0.9732
##                  95% CI : (0.9698, 0.9763)
##     No Information Rate : 0.9667
##     P-Value [Acc > NIR] : 0.0001044
##
##                   Kappa : 0.4278
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3153
##             Specificity : 0.9959
##          Pos Pred Value : 0.7241
##          Neg Pred Value : 0.9769
##              Prevalence : 0.0333
##          Detection Rate : 0.0105
##    Detection Prevalence : 0.0145
##       Balanced Accuracy : 0.6556
##
##        'Positive' Class : Yes
##
```

$$\begin{aligned}\text{False Positive rate} &= Prob(\widehat{default} = Yes | default = No) \\ &= \frac{40}{9627 + 40} = 0.4138\% \\ \text{True Positive rate} &= Prob(\widehat{default} = Yes | default = Yes) \\ &= \frac{105}{105 + 228} = 31.53\%\end{aligned}$$

2 Q2. Lost Sales

2.1 A quick EDA

```
df.lost<-read.csv("lostsales.txt")
df.lost <- df.lost %>% mutate(Part.Type = factor(Part.Type),
                      Status = factor(Status))
str(df.lost)
```

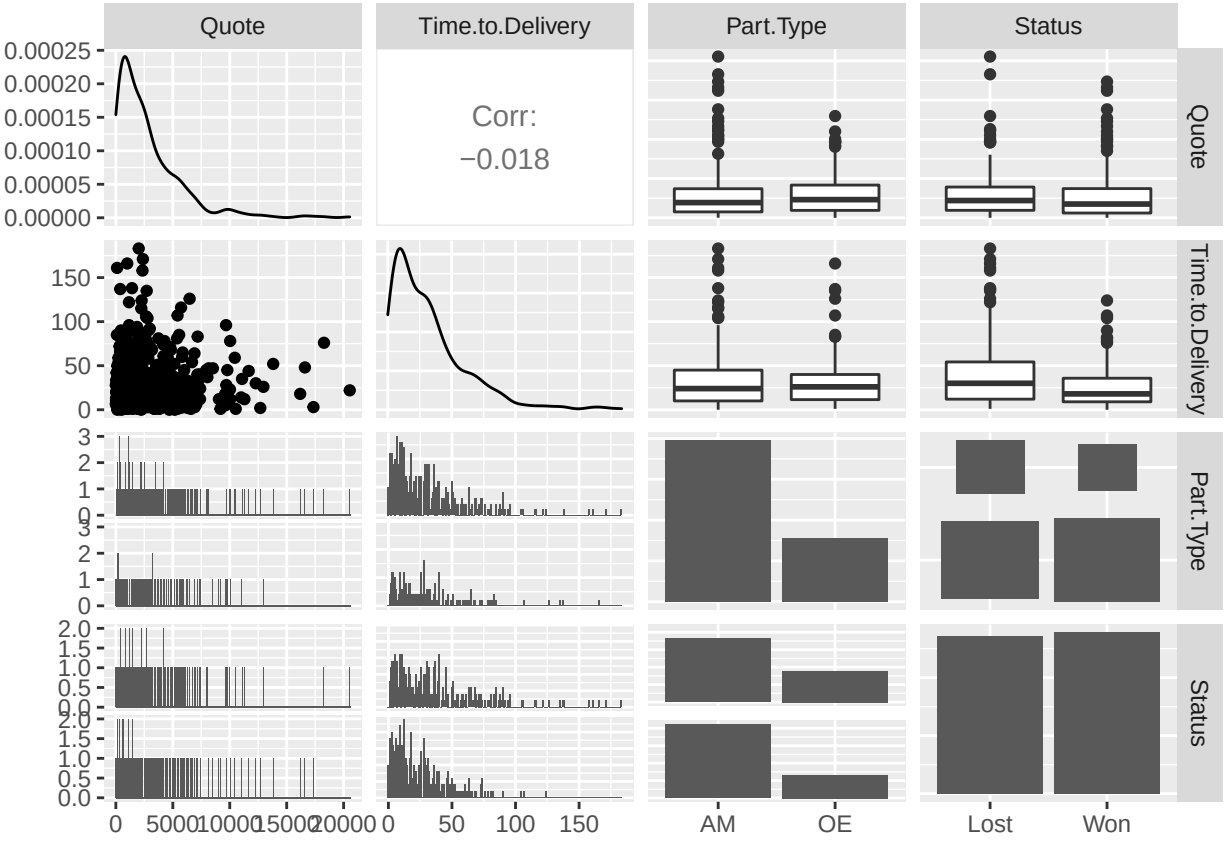
```
## 'data.frame':    550 obs. of  4 variables:
##  $ Quote           : int  3452 429 102 1153 102 2313 2681 5486 845 1115 ...
##  $ Time.to.Delivery: int  6 9 14 16 11 46 63 35 16 40 ...
##  $ Part.Type       : Factor w/ 2 levels "AM","OE": 2 1 2 2 2 1 2 1 2 1 ...
##  $ Status          : Factor w/ 2 levels "Lost","Won": 2 2 2 1 2 1 1 2 1 2 ...
```

```
sum(is.na(df.lost))
```

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

```
df.lost%>%
  GGally::ggpairs(progress=FALSE,lower=list(combo=GGally::wrap("facethist", binwidth=1)))
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



2.2 Variable Selection

We build a model with both independent variables.

```
fit.lost1 <- glm(Status~.,data=df.lost,family=binomial(logit))
summary(fit.lost1)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(logit), data = df.lost)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.497  -1.175   0.882   1.091   1.961
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       7.768e-01  1.712e-01   4.537 5.70e-06 ***
## Quote            -1.931e-05  3.073e-05  -0.628   0.5299
## Time.to.Delivery -1.837e-02  3.483e-03  -5.274 1.34e-07 ***
## Part.TypeOE      -4.711e-01  1.968e-01  -2.394   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 762.40  on 549  degrees of freedom
## Residual deviance: 723.83  on 546  degrees of freedom
## AIC: 731.83
##
## Number of Fisher Scoring iterations: 4
```

The p-value of quote seems insignificant, so let's build a model without it

```
fit.lost2 <- glm(Status~.,data=df.lost[,-c(1)],family=binomial(logit))
summary(fit.lost2)
```

```
##
## Call:
## glm(formula = Status ~ ., family = binomial(logit), data = df.lost[,
##     -c(1)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4877  -1.1731   0.8893   1.0972   1.9394
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.723499   0.148345   4.877 1.08e-06 ***
## Time.to.Delivery -0.018344   0.003484  -5.266 1.39e-07 ***
## Part.TypeOE      -0.475768   0.196582  -2.420   0.0155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 762.40  on 549  degrees of freedom
## Residual deviance: 724.22  on 547  degrees of freedom
## AIC: 730.22
##
## Number of Fisher Scoring iterations: 4
```

We use Type I anova to test whether Quote is significant.

```
anova(fit.lost1, fit.lost2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Status ~ Quote + Time.to.Delivery + Part.Type
## Model 2: Status ~ Time.to.Delivery + Part.Type
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       546     723.83
## 2       547     724.22 -1 -0.39542   0.5295
```

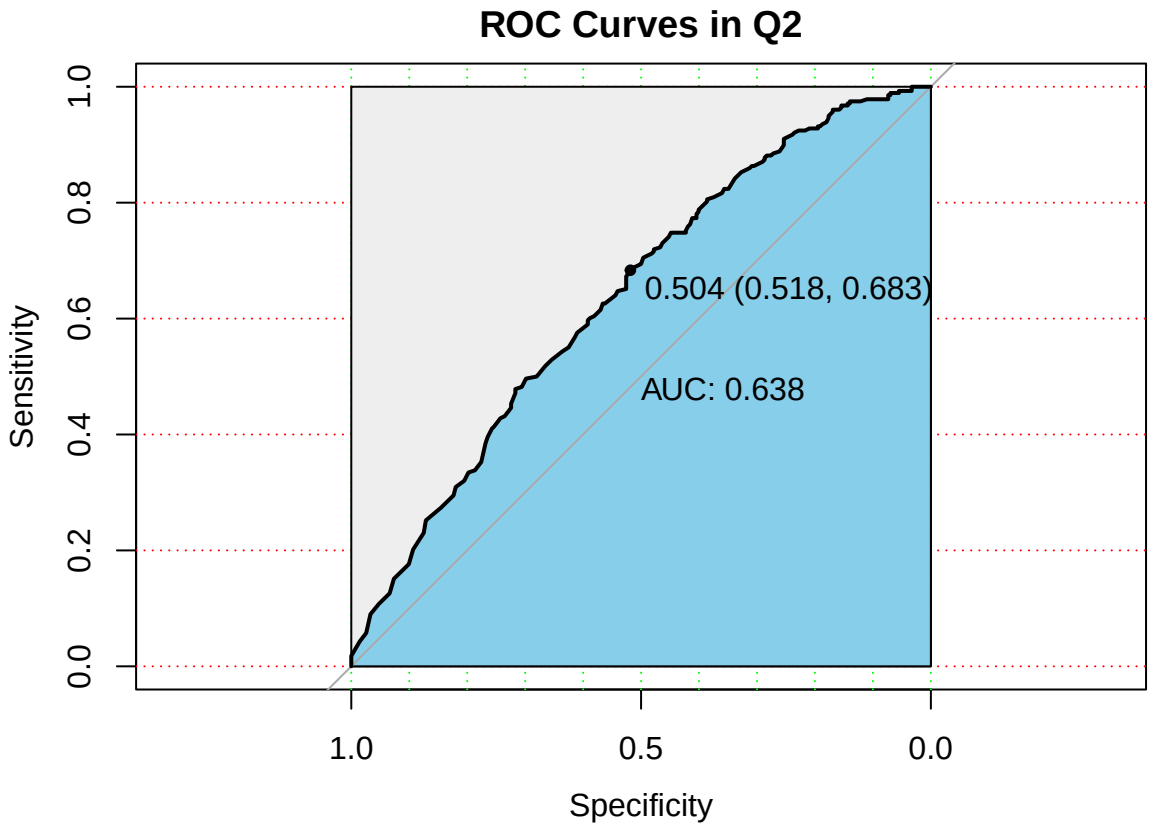
We don't have evidence to reject the null at 0.05, the shorter model is adopted.

```
roc.lost <- roc(df.lost$Status,fit.lost2$fitted.values)
```

```
## Setting levels: control = Lost, case = Won
```

```
## Setting direction: controls < cases
```

```
plot(roc.lost, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),
     grid.col=c("green", "red"), max.auc.polygon=TRUE,
     auc.polygon.col="skyblue", print.thres=TRUE,main='ROC Curves in Q2')
```



The data provide some useful information based on the two reasons below:

- Residual deviance is smaller than Null deviance.
- AUC is larger than 0.5

Analysis

- Given 'Part Type' is the same, a small quoted number of calendar days within which the order is to be delivered increases a customer will place an order.
- Given 'Time to Delivery' is the same, quotes with original equipment are less likely to make customers place orders than that with aftermarket.

3 Q3. Wine Quality

```
df.wine <-read.csv("winequality-bc.txt")
str(df.wine)
```

```
## 'data.frame':    6497 obs. of  16 variables:
##  $ Quality             : chr  "Just OK" "Just OK" "Just OK" "Just OK" ...
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ color               : chr  "red" "red" "red" "red" ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
##  $ Crossvalidation     : int  3 2 2 3 3 2 3 3 3 2 ...
##  $ Hold.Test           : int  0 1 1 0 0 0 0 0 0 0 ...
```

```
sum(is.na(df.wine))
```

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

After carefully checking, we find

- 'quality' contain duplicative information as 'Quality'.
- 'Crossvalidation' and 'Hold.Test' are the index of cross validation and testing data.

All these three variables mentioned above are irrelevant to our analysis, so we delete them.

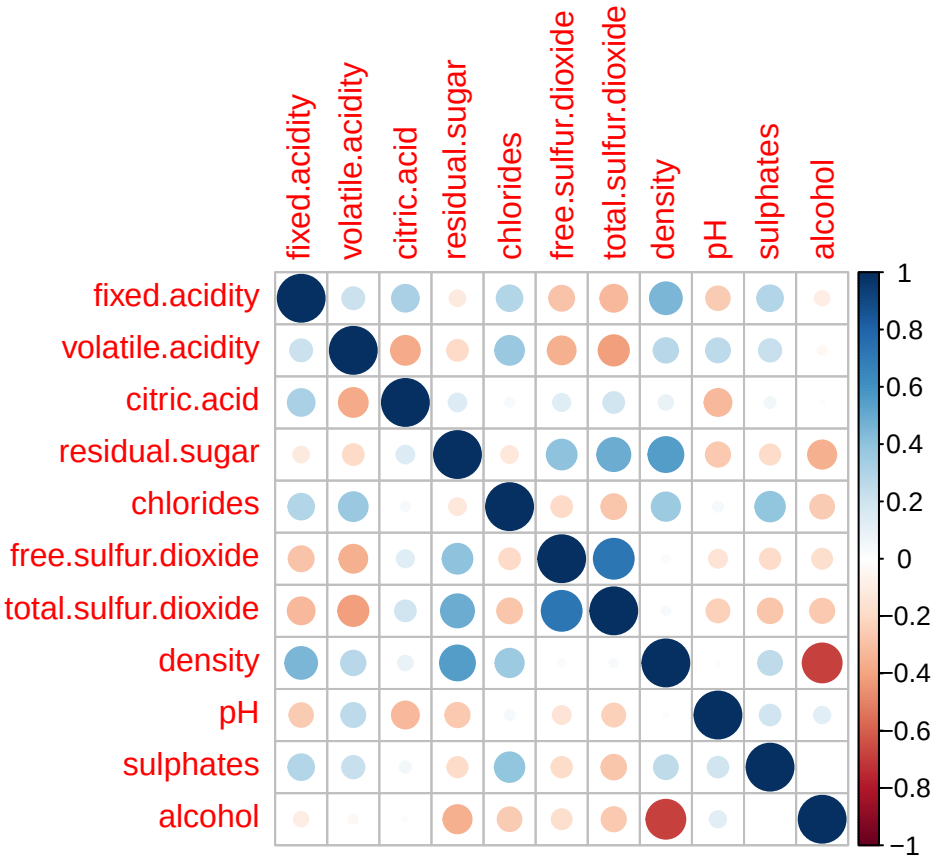
```
df.wine <- df.wine[,-c(14,15,16)]
```

3.1 EDA

You could put some Pairplot and Boxplot Here. I list some pictures here.

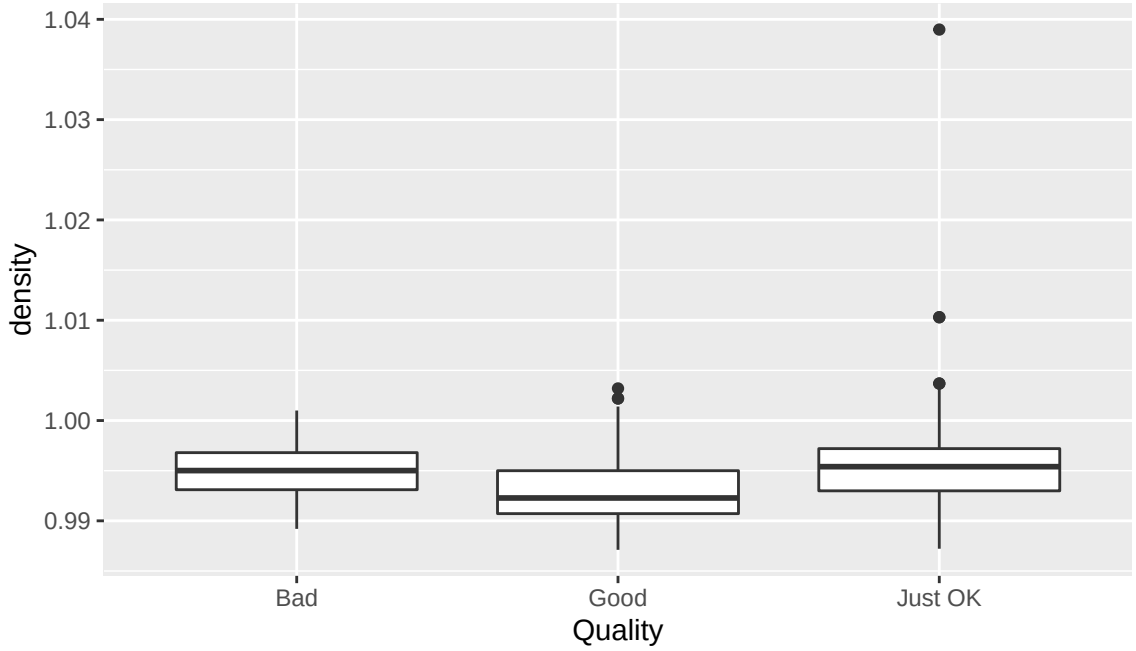
Pairplot for numeric variables plotted by using `corrplot()` in `corrplot` package.

```
corrplot::corrplot(cor(df.wine[,-c(1,13,14,15,16)]))
```

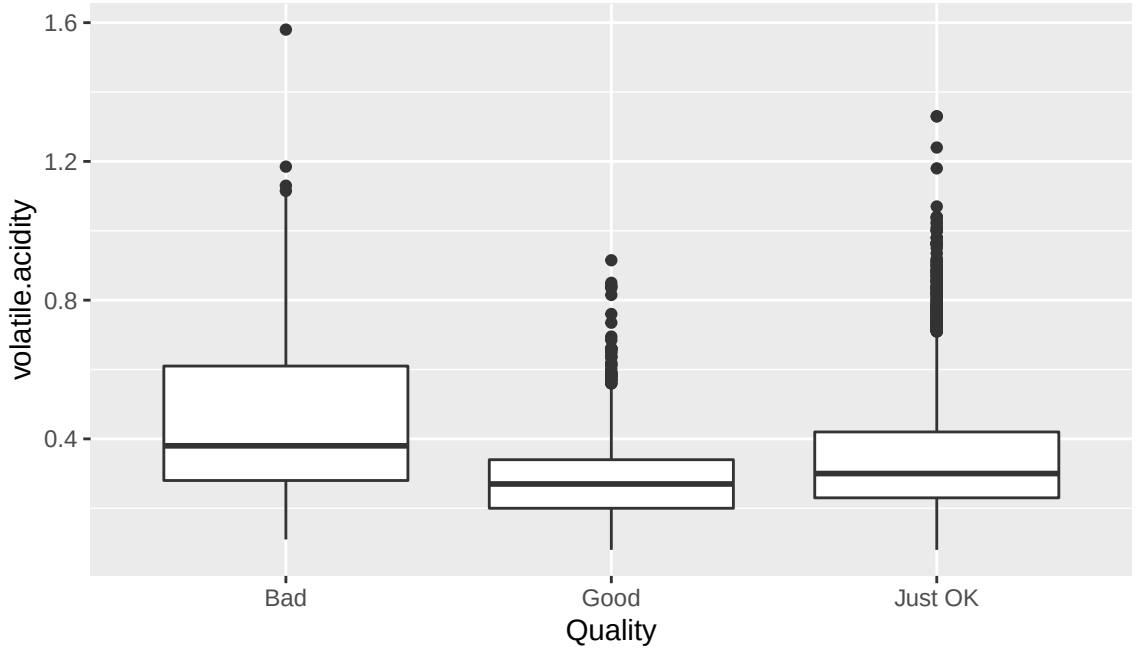


Somp boxplot plotted by `ggplot` package

```
ggplot(df.wine, aes(x = Quality, y = density)) +
  geom_boxplot()
```




```
ggplot(df.wine, aes(x = Quality, y = volatile.acidity)) +
  geom_boxplot()
```



3.2 Model Building

You could adopt either multinomial Logistic Regression or Ordinal Logistic Regression in this question.

For both model, two kinds of variable selection are recommended.

- We use **cross-validation** and compare **the average testing auc** or **the average testing accuracy** of different models.
- We use Type II ANOVA to do varaiable selection

Here I use Type II ANOVA to do varaiable selection.

(Open Question: the answer will be different based on different criterizon used)

3.2.1 Multinominal Logistic Regression

```
pacman::p_load(nnet,MASS,car)
```

```
fit.wine.mult1 <- multinom(Quality~.,df.wine,family=binomial)
```

```
## # weights:  42 (26 variable)
## initial  value 7137.684039
## iter  10 value 4259.908878
## iter  20 value 3518.855159
## iter  30 value 3416.213418
## iter  40 value 3414.548973
## iter  50 value 3399.686226
## final  value 3399.681115
## converged
```

```
summary(fit.wine.mult1)
```

```
## Call:
## multinom(formula = Quality ~ ., data = df.wine, family = binomial)
##
## Coefficients:
##         (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## Good       196.5053   -0.02016505        -8.888231   0.2711682    0.199401694
## Just OK   -149.9578   -0.48052190        -5.411479   0.5487126    0.002714186
##          chlorides free.sulfur.dioxide total.sulfur.dioxide   density
## Good    -11.808343          0.03433809         -0.000128277 -202.9720
## Just OK  -3.745384          0.02381459          0.003953386  163.8473
##                   pH sulphates   alcohol colorwhite
## Good    -0.003291793 2.7484764 0.9631016  -3.808434
## Just OK -2.453434671 0.3824261 0.4585009  -3.262344
##
## Std. Errors:
##         (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
## Good      0.3345904    0.06622259        0.5891280   0.6562809     0.02149645
## Just OK   0.3222637    0.06029125        0.4651551   0.5804932     0.01980013
##         chlorides free.sulfur.dioxide total.sulfur.dioxide   density        pH
## Good    0.6396966         0.007221581          0.002632740 0.3276160 0.2839091
## Just OK 1.2782010         0.006753942          0.002361423 0.3179803 0.2746314
##         sulphates    alcohol colorwhite
## Good    0.6498012 0.07914110  0.3600112
## Just OK 0.6118992 0.07344163  0.3254054
##
```

```
## Residual Deviance: 6799.362
## AIC: 6851.362
```

```
Anova(fit.wine.mult1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Quality
##                       LR Chisq Df Pr(>Chisq)
## fixed.acidity           48.524  2  2.906e-11 ***
## volatile.acidity       227.115  2  < 2.2e-16 ***
## citric.acid              1.497  2  0.4730644
## residual.sugar          72.071  2  2.239e-16 ***
## chlorides               16.126  2  0.0003149 ***
## free.sulfur.dioxide     27.025  2  1.354e-06 ***
## total.sulfur.dioxide     0.588  2  0.7454535
## density                 34.710  2  2.904e-08 ***
## pH                      55.446  2  9.120e-13 ***
## sulphates               68.006  2  1.709e-15 ***
## alcohol                 20.992  2  2.765e-05 ***
## color                   93.588  2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anova shows that “citric.acid” and “total.sulfur.dioxide” are insignificant once other variables are included. Let us remove them and rebuild the model.

```r
# removing "citric.acid" and "total.sulfur.dioxide
fit.wine.mult2 <- multinom(Quality~.,df.wine[,-c(4,8)],family=binomial)
```

```
## # weights:  36 (22 variable)
## initial  value 7137.684039
## iter  10 value 3900.413077
## iter  20 value 3437.800056
## iter  30 value 3427.872044
## iter  40 value 3410.419553
## final  value 3406.888237
## converged
```

```r
summary(fit.wine.mult2)
```

```
## Call:
## multinom(formula = Quality ~ ., data = df.wine[, -c(4, 8)], family = binomial)
##
## Coefficients:
##         (Intercept) fixed.acidity volatile.acidity residual.sugar  chlorides
## Good       215.1316    0.01505405        -8.951078   0.2065139415 -11.496342
## Just OK   -175.7221   -0.45928941        -5.418608  -0.0003529156  -3.330753
##         free.sulfur.dioxide   density          pH sulphates   alcohol
## Good             0.03560997 -221.8716  0.01015937 2.9237371 0.9388694
## Just OK          0.03056241  189.7147 -2.51608507 0.5800593 0.4630651
##         colorwhite
## Good     -3.758841
## Just OK  -2.799248
##
## Std. Errors:
##         (Intercept) fixed.acidity volatile.acidity residual.sugar chlorides
## Good       1.305458    0.07721924        0.5605188     0.02172833  3.165820
## Just OK    1.181556    0.06968058        0.4413805     0.02001323  2.038067
##         free.sulfur.dioxide  density        pH sulphates    alcohol colorwhite
## Good            0.006003116 1.280882 0.5952902 0.6665871 0.07945272  0.3368543
## Just OK         0.005637810 1.162303 0.5478746 0.6276025 0.07251571  0.2968059
##
## Residual Deviance: 6813.776
## AIC: 6857.776
```

We use Type II anova again and the result shows that every variable is significant. We stop here.

By summarizing the model output, a good wine has below characteristics:

- Good wine should have modest ‘fixed.acidity‘.
- Good wine should have low ‘volatile.acidity‘.
- Good wine should have high ‘residual.sugar‘
- Good wine should have low ‘chlorides‘.
- Good wine should have high ‘free.sulfur.dioxide‘.
- Good wine should have modest ‘density‘.
- Good wine should have modest ‘PH‘.
- Good wine should have high ‘sulphates‘
- Good wine should have high ‘alcohol‘.
- Good wine should tend not to be ‘colorwhite‘.

Here we say a feature is “modest” if the coefficients for the class “Just OK” and the class “Good” have different signs.

3.2.2 Ordinal Logistic Regression

Let us try to fit the proportional odds logistic regression model.

```
df.wine %>% str()
```

```
## 'data.frame':    6497 obs. of  13 variables:
##  $ Quality             : chr  "Just OK" "Just OK" "Just OK" "Just OK" ...
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ color               : chr  "red" "red" "red" "red" ...
```

```
df.wine <- df.wine %>% mutate(color=factor(color),
                              Quality=factor(Quality,level=c('Bad','Just OK','Good'),order=T))
```

Similar to the multinomial logistic regression, we find "citric.acid" and "total.sulfur.dioxide" are insignificant after Type II ANOVA.

```
# removing "citric.acid" and "total.sulfur.dioxide
fit.wine.polr <- polr(Quality~., data=df.wine[,-c(4,8)])
summary(fit.wine.polr)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = Quality ~ ., data = df.wine[, -c(4, 8)])
##
## Coefficients:
##                          Value Std. Error  t value
## fixed.acidity          0.36205   0.033516   10.802
## volatile.acidity      -3.97250   0.271026  -14.657
## residual.sugar         0.21591   0.008416   25.656
## chlorides             -3.09487   1.222677   -2.531
## free.sulfur.dioxide    0.01092   0.002238    4.878
## density             -419.07277   0.537942 -779.029
## pH                     1.85456   0.246524    7.523
## sulphates              2.11945   0.242269    8.748
## alcohol                0.42374   0.032248   13.140
## colorwhite            -1.60715   0.139158  -11.549
##
## Intercepts:
##               Value     Std. Error t value
## Bad|Just OK  -407.8371    0.5467  -745.9398
## Just OK|Good -402.1183    0.5635  -713.5650
##
## Residual Deviance: 7008.946
## AIC: 7032.946
```

By summarizing the model output, a good wine has below characteristics:

- Good wine should have high ‘fixed.acidity‘.
- Good wine should have low ‘volatile.acidity‘
- Good wine should have high ‘residual.sugar‘
- Good wine should have low ‘chlorides‘.
- Good wine should have high ‘free.sulfur.dioxide‘.
- Good wine should have low ‘density‘.
- Good wine should have high ‘PH‘.
- Good wine should have high ‘sulphates‘
- Good wine should have high ‘alcohol‘.
- Good wine should tend not to be ‘colorwhite‘.

As in proportional odds logistic regression, different levels of ordinal categories share the same coefficients (except intercept), here we only conclude high or low, depending on the coefficient signs. But largely speaking, the conclusions are more or less the same.

Note that the coefficients given by `polr` is negative of the beta coefficients used in class. So from the above `polr` output, for example, `volatile.acidity` has a negative coefficient -3.97. This means

$$\log(\pi_{bad}/(\pi_{ok}+\pi_{good})) = \beta_{01} + ... + \mathbf{3.97}x_{volatile.acidity} + ...,$$

$$\log((\pi_{bad}+\pi_{ok})/\pi_{good}) = \beta_{02} + ... + \mathbf{3.97}x_{volatile.acidity} + ...,$$

both of which means an increase in `volatile.acidity` will lead to an increase in π_{bad} and $\pi_{bad}+\pi_{ok}$. More specifically, one unit increase in `volatile.acidity` gives odds ratio of $e^{3.97}$, or leads to an increase in the odds of $\pi_{bad}/(\pi_{ok}+\pi_{good})$ and $(\pi_{bad}+\pi_{ok})/\pi_{good}$ by a multiplicative factor of $e^{3.97}$. Since both π_{bad} and $\pi_{bad}+\pi_{ok}$ increases as `volatile.acidity` increases, good wine should have low `volatile.acidity`.