Using Multiple Regression to Predict Wine Quality

STAT 512–Y01 Project

Ishaan Saxena (isaxena@purdue.edu)
Jeffery French (french21@purdue.edu)
Steven Cardenas (scarden@purdue.edu)

26 May 2019

1 Introduction

This dataset contains physicochemical indicators and a sensory ‘rating’ for many variants of the Portuguese “Vinho Verde” wine. With this project, we aim to find if there is any significant association between some of these physicochemical features of a wine with the way its quality is determined. Since the quality is determined subjectively, essentially we are trying to ask which of these physicochemictal features, if any, have an effect on the taste.

1.1 Data Description

Dataset: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009.

```r
winedata <- read.csv("../data/winequality.csv")
colnames(winedata) <- c("x1", "x2", "x3", "x4", "x5", "x6",
                        "x7", "x8", "x9", "x10", "x11", "y")
```

1.1.1 Input Variables:

1. Fixed Acidity: contents of non-volatile acids in the wine
2. Volatile Acidity: amounts of acetic acid in the wine
3. Citric Acid: amounts of citric acid in the wine
4. Residual Sugar: the amount of sugar remaining after the end of fermentation
5. Chlorides: the amount of chloride salts present in the wine
6. Free Sulfur Dioxide: presence of free SO2 prevents oxidation
7. Total Sulfur Dioxide: contributes to the nose and taste of wine
8. Density: depends on alcohol and sugar content
9. pH: measures acidity of the wine on scale of 0-14
10. Sulphates: act as antimicrobials and antioxidants
11. Alcohol: Percentage of alcohol content in the sample

1.1.2 Response Variable:

Quality: The wine quality is a rating on a scale of 1 to 10 and is discrete.

1.2 Plan of Analysis

In order to determine the impact of wine characteristics on quality, the team will do a variety of different regression techniques. The first step of the analysis will be to test the relevant assumptions related to regression and determine the correct techniques to test the data. Then, regression techniques will be applied to the data using RStudio and the impact of the input variables will be determined.

2 Checking for Evidence of Association

Before we proceed with diagnotics, we ought to check if any of the physicochemical predictors have any association with the repsonse variable at all. If non of the predictors provide any valuable information about the response variable, it would not make sense to use these to create a model. As such, we require that at least one of the predictors has some kind of association with the response variable.

We can determine this with the following hypothesis test:

2.1 Hypotheses

- $H_0 : \forall i \in \{1, ..., 11\}\ \beta_i = 0$, i.e., no predictor has any association with the response variable.
- $H_a : \exists i \in \{1, ..., 11\}\ \beta_i \neq 0$, i.e., at least one predictor has some kind of association.

2.2 Test Statistics

We can easily utilize R to obtain the F test statistics as follows:

```r
# Fit a linear model on all predictors
winedata.fullmodel <- lm(y~., data=winedata)
summary(winedata.fullmodel)
```

```
##
## Call:
## lm(formula = y ~ ., data = winedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.197e+01  2.119e+01   1.036   0.3002
## x1           2.499e-02  2.595e-02   0.963   0.3357
## x2          -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## x3          -1.826e-01  1.472e-01  -1.240   0.2150
## x4           1.633e-02  1.500e-02   1.089   0.2765
## x5          -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
## x6           4.361e-03  2.171e-03   2.009   0.0447 *
## x7          -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## x8          -1.788e+01  2.163e+01  -0.827   0.4086
## x9          -4.137e-01  1.916e-01  -2.159   0.0310 *
## x10          9.163e-01  1.143e-01   8.014 2.13e-15 ***
## x11          2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

With this, we obtain $F_s = 81.35$ on 11 andd 1587 degrees of freedom.

We reject the null hypothesis if $F_s > F(1-\alpha, df_1, df_2)$. At $\alpha = 0.01$, we obtain:

```
qf(1-0.01, 11, 1587)
```

```
## [1] 2.258883
```

2.3 Conclusion

At $\alpha = 0.01$, $F_s > F(1-\alpha, df_1, df_2)$, thus, we reject the null hypothesis in favor of the alternate hypothesis. This indicates that at least one of the predictors has some significant association with the response variable.

3 Diagnostics

3.1 Residuals follow normal distribution

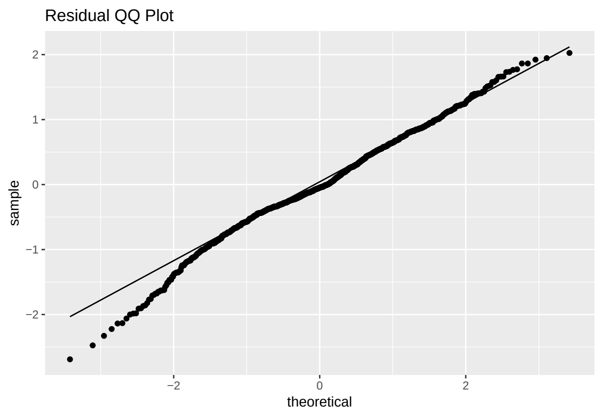
In order to proceed correctly with MLR, we must make sure that the residuals obtained from the regression model follow normal distribution. If they do not, we must apply proper remedial measures to our model to improve its performance.

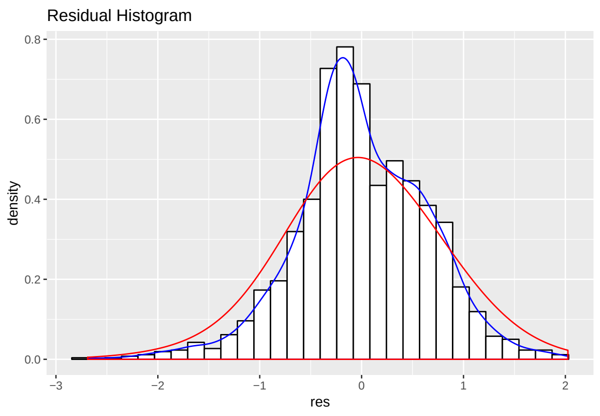
We can obtain the residuals as follows:

```
res <- winedata.fullmodel$residuals
```

Before we proceed to check this assumption with formal tests, we can first look at some plots that would give us a basic idea about the distribution of residuals:

```
ggplot(data.frame(res), aes(sample=res)) +
    geom_qq() + geom_qq_line() +
    ggtitle("Residual QQ Plot")
ggplot(data.frame(res), aes(res)) +
    geom_histogram(aes(y=..density..), fill="white", color="black") +
    geom_density(aes(y=..density..), colour="blue") +
    geom_density(aes(y=..density..), colour="red", adjust=4) +
    ggtitle("Residual Histogram")
```





Clearly, this deviates slightly from normal. It seems like this deviation from normal is a result of outlying observations, since the residuals on both tails of the distribution seem to have a big effect on it.

Let's proceed to check this assumption formally with the Shapiro-Wilk test for normality:

3.1.1 Hypotheses

- H_0 : residuals follow a normal distribution
- H_a : residuals deviate from a normal distribution

3.1.2 Test Statistics

```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99087, p-value = 1.954e-08
```

3.1.3 Conclusion

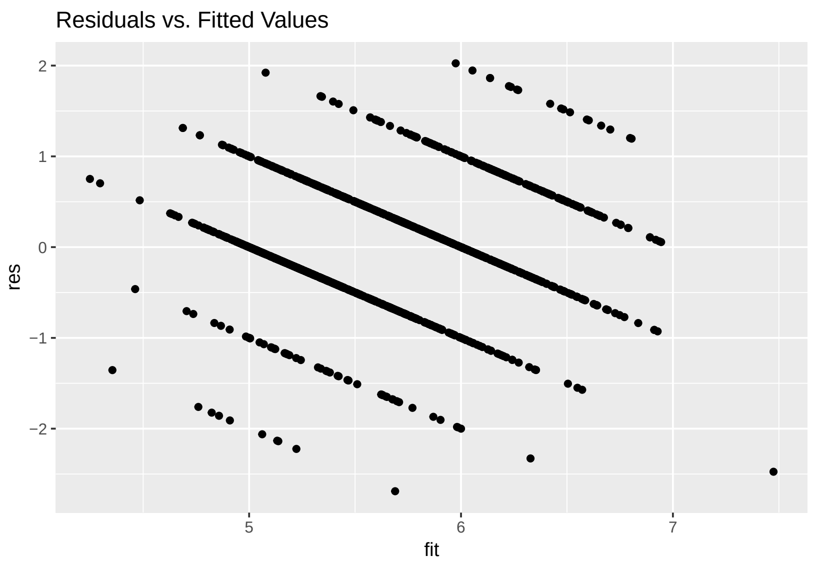
At $\alpha = 0.01$, we have $p << \alpha$, and thus, should reject the null hypothesis in favor of the alternate hypothesis. This provides strong evidence for the nonnormality of the residuals.

We can apply transformations or use more robust regression methods such as ridge regression to remedy this issue, since it seems to be the outlying cases which have an effect on the normality of the residuals.

3.2 Residuals have constant variance

To analyze the variance of the residuals, we can check their behavior against the fitted values.

```
fit <- winedata.fullmodel$fitted.values
ggplot(data.frame(fit, res), aes(fit, res)) +
    geom_point() + ggtitle("Residuals vs. Fitted Values")
```



Again, it seems that due to the presence of some outliers, the variance of the residuals in this model with respect to the fitted values is nonconstant. We can check this assumption formally with the Brown-Forsythe test:

3.2.1 Hypotheses

- H_0 : residuals have constant variance
- H_a : residuals have nonconstant variance

3.2.2 Test Statistics

Since the fitted values are continuous, we can split them into four groups with the quartile values as follows. This gives us 4 groups of almost equal sizes. We can then conduct the BF test in R:

```r
g <- rep(1, nrow(winedata))
q <- data.frame(quantile(fit))
g[fit > q[2, 1]] = 2
g[fit > q[3, 1]] = 3
g[fit > q[4, 1]] = 4
bftest(lm(winedata$y~fit), g)
```

```
##       t.value      P.Value alpha   df
## [1,] 7.331325 3.606004e-13  0.05 1597
```

3.2.3 Conclusion

At $\alpha = 0.01$, we have $p << \alpha$, and thus, should reject the null hypothesis in favor of the alternate hypothesis. This provides strong evidence for the nonconstant variance of residuals.

We can apply transformations or use more methods such as weighted least squares to try to remedy this issue. **However, any such methods would only help a little, since our response variable is discrete and constrained. Ideally, we would want to use a more generalized technique such as GLM for this data, but we merely wish to explore the behavior of MLR on it.

3.3 Residuals are independent

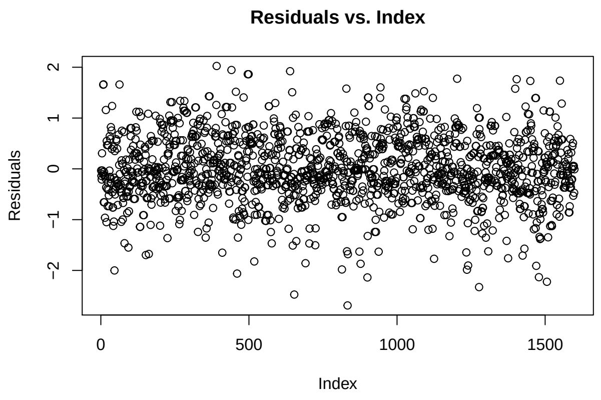
Recall that for our regression model, we have:

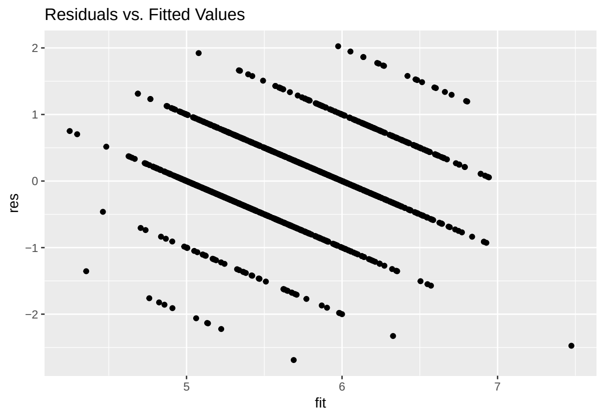
$$Y_i = \beta_0 + \sum_{j \in \{1,\ldots,11\}} \beta_j X_{ij} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma)$. As a result, the residuals obtained from our MLR model must reflect the properties of this ϵ for the model to be an effective representation of the data. As such, we must make sure that our residuals are independent.

Let's look at the following plots to see how residuals behave wrt to the collection order of data and with respect to the fitted values:

```r
plot(res, xlab='Index', ylab='Residuals', main='Residuals vs. Index')
fit <- winedata.fullmodel$fitted.values
ggplot(data.frame(fit, res), aes(fit, res)) +
    geom_point() + ggtitle("Residuals vs. Fitted Values")
```





If we assume indices increase by collection order, then there is no apparent patterns in collection order. Furthermore, since our sample size (1599) is much greater than the number of parameters (11), we can ignore the minor dependence which we observe in residuals due to the residual equation. Note that the residual correlation with the fitted values is extremely low:

```r
cor(fit, res)
```

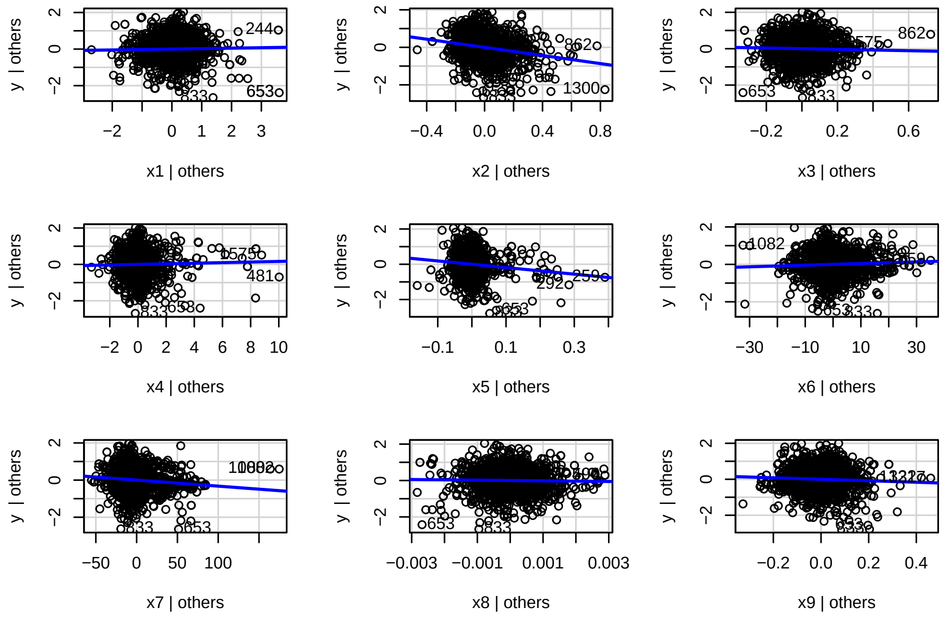
```
## [1] -2.224153e-16
```

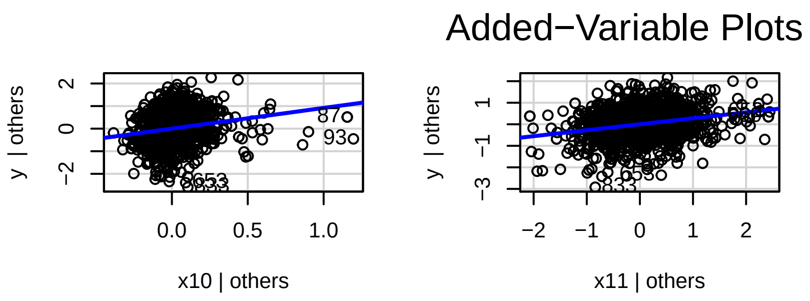
3.4 Predictors have linear relationships with reponse variable

To check for potential improvements in our model, we check to see if the predictors have linear relationships with the reponse variables. If they have a constant relationship, we have no reason to include these predictors since they merely add noise to the model and not much information. If they have a nonlinear relationship, we can benefit for a transformation of the respective predictor in our final model.

We can look at the added-variable plots for each variable to determine whether they have linear or nonlinear impact on the response variable (or if they have no impact at all). This information will also be valuable later during model selection.

```r
avPlots(winedata.fullmodel)
```





We can clearly see that there is no nonlinear impact on the response variables from any of the predictors from these scatter plots (Y|others vs X_i|others). Thus, there is no need for transformations in X.

Furthermore, we note that the variables X_1, X_3, X_4, X_8 seem to not add much information to the model, and perhaps can be considered to be dropped.

3.5 Outliers

Outlying data points can have significant effects on the values of the parameters of the model. They can also have an effect on the correctness of the model, since they may effect other assumptions which we require.

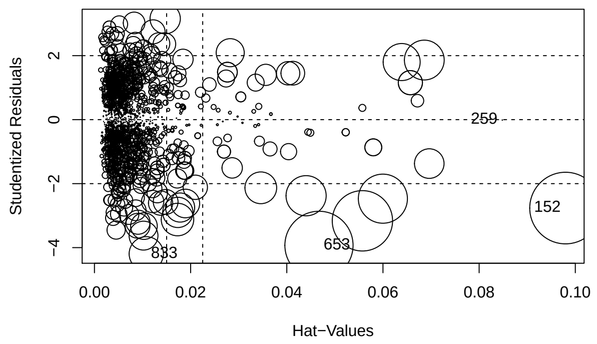
As such, if we detect some significant presence of outliers in our data, we must take remedial measures to either remove them or use more robust regression methods. Note that since our dataset is decently large, removing outliers is not as desirable as using a method like ridge regression would be.

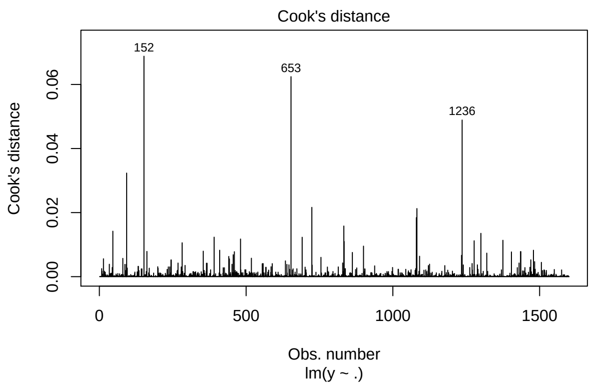
We can obtain the influence plots and Cook's distances of each sample to determine if we have any significant outliers:

```r
influencePlot(winedata.fullmodel)
```

```
##          StudRes        Hat        CookD
## 152 -2.76332786 0.09796358 6.881957e-02
## 259 -0.01283858 0.08482216 1.273886e-06
## 653 -3.92899309 0.04669000 6.243643e-02
## 833 -4.19408836 0.01078082 1.581017e-02
```

```r
plot(winedata.fullmodel, pch=18, which=c(4))
```





Since we have only a few outliers, there seems to be no need to take remedial measures for these.

3.6 Multicollinearity

For us to have valuable information extracted from each predictor in our model, we must assure that none of thee predictors have a significant association with eachother. If adding a new predictor to the model changes some parameter values or their variance greatly, we would have instability in our model due to multicollinearity. As such, we must check for such association between predictors.

In addition to the many things we have looked at before hand, such as AVPlots, we can check the Variance Inflation Factor (VIF_i) values of each predictor X_i:

```
vif <- c(VIF(lm(x1~x2+x3+x4+x5+x6+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x2~x1+x3+x4+x5+x6+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x3~x2+x1+x4+x5+x6+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x4~x2+x3+x1+x5+x6+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x5~x2+x3+x4+x1+x6+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x6~x2+x3+x4+x5+x1+x7+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x7~x2+x3+x4+x5+x6+x1+x8+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x8~x2+x3+x4+x5+x6+x7+x1+x9+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x9~x2+x3+x4+x5+x6+x7+x8+x1+x10+x11, data=winedata)))
vif <- c(vif, VIF(lm(x10~x2+x3+x4+x5+x6+x7+x8+x9+x1+x11, data=winedata)))
vif <- c(vif, VIF(lm(x11~x2+x3+x4+x5+x6+x7+x8+x9+x10+x1, data=winedata)))
t(t(vif))
```

```
##           [,1]
##  [1,] 7.767512
##  [2,] 1.789390
##  [3,] 3.128022
##  [4,] 1.702588
##  [5,] 1.481932
##  [6,] 1.963019
##  [7,] 2.186813
##  [8,] 6.343760
##  [9,] 3.329732
## [10,] 1.429434
## [11,] 3.031160
```

Since none of the VIF_i values are significantly large ($VIF_i < 10\ \forall i$), none of the predictors are related strongly to other predictors. As a result, there is no need for remedial measures such as ridge regression.

4 Remedial Measures

There is no need to transform X as the data has no nonlinear impact on the response variable. Furthermore, we have no multicollinearity, and as such, need to remedial measures for that. We can also ignore the nonnormality of the residuals since this is very minimal.

As a result, we need to fix only the fact that the residuals have nonconstant variance. This can be done with just transformations to Y or with Weighted Least Squares.

4.1 Transforming Y Values

As shown in the diagnostics section, it seems that our model will benefit from a transformation of the Y values. In order to do so, we can apply the Box-Cox procedure to choose a lambda for which we obtain:

$$\max_{\lambda} \log L(D; \beta_i)$$

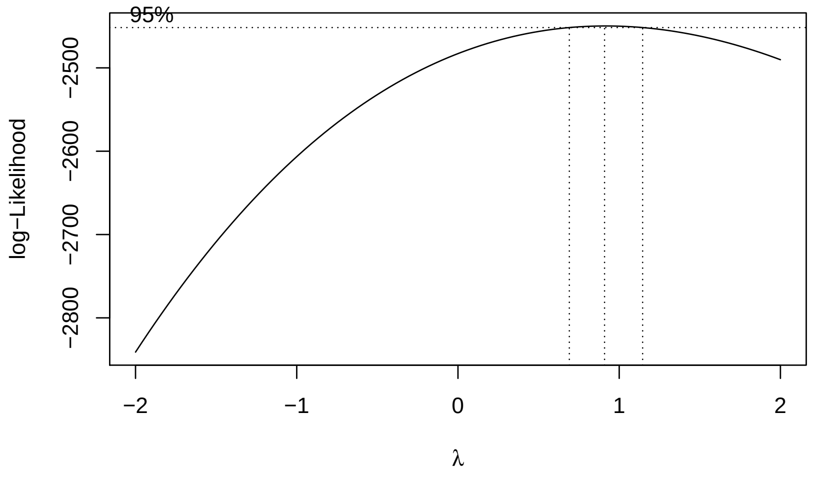
That is, we pick a λ value which maximizes the log-likelihood of our the data being extracted from a distribution which matches that of our model. We can then apply the transformation

$$Y_t = Y^\lambda$$

and use this transformed Y value to fit a linear model.

This can be done in R as follows:

```
bcmle <- boxcox(winedata$y~fit)
```



From this, we can obtain the transformed Y values Y_t like so:

```
lambda <- bcmle$x[which.max(bcmle$y)]
winedata$yt <- winedata$y^lambda
print(lambda)
```

```
## [1] 0.9090909
```

4.2 Rechecking residual variance

We can use the same hypothesis test as in section 3.2 to check if the residuals have constant variance:

```
winedata.transformedmodel <- lm(yt~.-y, data=winedata)
fit <- winedata.transformedmodel$fitted.values
g <- rep(1, nrow(winedata))
q <- data.frame(quantile(fit))
g[fit > q[2, 1]] = 2
g[fit > q[3, 1]] = 3
g[fit > q[4, 1]] = 4
bftest(lm(winedata$yt~fit), g)
```

```
##       t.value      P.Value alpha   df
## [1,] 7.115636 1.675993e-12  0.05 1597
```

The standard deviations of each of the groups are as follows:

```
c(sd(fit[g==1]), sd(fit[g==2]), sd(fit[g==3]), sd(fit[g==4]))
```

```
## [1] 0.12172617 0.07170541 0.08779837 0.17651310
```

While there seems to be some improvement, this is not a very major reduction in the nonconstant variance of the residuals.

Recall that response variables are discrete and constrained. This indicates that reducing nonconstant variance in our data would be very hard, even if we use a technique like Weighted Least Squares. As the following code demonstrates, this does not have much of an effect on the heterscedasticity of the residuals:

```r
wts <- 1/fitted(lm(abs(residuals(winedata.transformedmodel))~.-y, winedata))^2
winedata.wlm = lm(yt~.-y, weights=wts, data=winedata)
fit <- winedata.wlm$fitted.values
g <- rep(1, nrow(winedata))
q <- data.frame(quantile(fit))
g[fit > q[2, 1]] = 2
g[fit > q[3, 1]] = 3
g[fit > q[4, 1]] = 4
bftest(lm(winedata$yt~fit), g)
```

```
##       t.value      P.Value alpha   df
## [1,] 7.07168 2.280842e-12  0.05 1597
```

Although we should proceed with a model which allows for more general assumptions given such a dataset, such as a GLM, we continue with the model we currently have just so we can explore the behavior of an MLR with such a dataset and see if it can lead us to any valuable conclusions, as we aim to do with this project.

5 Model Selection

After the remedial measures, we obtain the following model:

$$Y_i^{\lambda} = \beta_0 + \sum_{j \in \{1,\ldots,11\}} \beta_j X_{ij} + \epsilon_i$$

However, as we show previously in diagnostics, not all predictors provide valuable information about the response variables. Indeed, such predictors may even add noise to our dataset along with increased complexity.

In order to obtain a better model which is less complex and a better fit on the data, we can perform best subset selection on our features.

5.1 Best Subset Selection

The following table demonstrates some selection criteria, such as `PRESSp`, `Cp`, `AICp`, for a few of the subsets of our predictors:

```r
# Get same results with higher values of num. num=1 is for more condensed output
bs<-BestSub(winedata[, 1:11], winedata$yt, num=1)
print(bs)
```

```
##     p 1 2 3 4 5 6 7 8 9 A B     SSEp        r2    r2.adj         Cp
## 1   2 0 0 0 0 0 0 0 0 0 0 1 487.1393 0.2251336 0.2246484 324.266810
## 2   3 0 1 0 0 0 0 0 0 0 0 1 429.9948 0.3160304 0.3151733 101.124528
## 3   4 0 1 0 0 0 0 0 0 0 1 1 418.1993 0.3347928 0.3335416  56.651913
## 4   5 0 1 0 0 0 0 1 0 0 1 1 413.3385 0.3425246 0.3408747  39.501157
## 5   6 0 1 0 0 1 0 1 0 0 1 1 408.4801 0.3502527 0.3482133  22.359302
## 6   7 0 1 0 0 1 0 1 0 1 1 1 404.9398 0.3558840 0.3534564  10.411149
## 7   8 0 1 0 0 1 1 1 0 1 1 1 403.4812 0.3582040 0.3553803   6.664643
## 8   9 0 1 1 0 1 1 1 0 1 1 1 403.1704 0.3586984 0.3554717   7.440145
## 9  10 0 1 1 1 1 1 1 0 1 1 1 403.0231 0.3589327 0.3553018   8.859781
## 10 11 1 1 1 1 1 1 1 0 1 1 1 402.9564 0.3590388 0.3550025  10.596988
## 11 12 1 1 1 1 1 1 1 1 1 1 1 402.8049 0.3592798 0.3548388  12.000000
##         AICp      SBCp   PRESSp
## 1  -1896.545 -1885.791 488.4592
## 2  -2094.064 -2077.932 431.9500
## 3  -2136.540 -2115.031 420.9385
## 4  -2153.234 -2126.348 416.5943
## 5  -2170.140 -2137.878 412.3068
## 6  -2182.059 -2144.419 409.4074
## 7  -2185.829 -2142.812 408.4926
## 8  -2185.061 -2136.667 408.7771
## 9  -2183.645 -2129.874 409.4024
## 10 -2181.910 -2122.762 410.0222
## 11 -2180.511 -2115.986 410.5608
```

Clearly, the subset $\{X_2, X_5, X_6, X_7, X_9, X_{10}, X_{11}\}$ is expected to perform and generalizee better than all others due to the lowest `PRESSp`, `Cp`, and `AICp` values.

As such, the reduced model becomes:

$$Y_i^\lambda = \beta_0 + \sum_{j \in \{2,5,6,7,9,10,11\}} \beta_j X_{ij} + \epsilon_i$$

Note that the predictors dropped from the full model are the same as the ones we previously conjectured to not add much new information to our model. Namely, these are X_1, X_3, X_4, X_8 and correspond to Fixed Acidity, Citric Acid, Residual Sugar, and Density respectively.

5.2 Comparing the full model with the reduced model

We can now check whether the reduced model is a better choice than the full model with the following hypothesis test:

5.2.1 Hypotheses

- $H_0 : \forall i \in \{1, 3, 4, 8\}\ \beta_i = 0$, i.e., reduced model is better
- $H_a : \exists i \in \{1, 3, 4, 8\}\ \beta_i \neq 0$, i.e., full model is better

5.2.2 Test Statistics

We can utilize R to obtain the F test statistics as follows:

```r
winedata.fullmodel <- lm(yt~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11, winedata)
winedata.reducedmodel <- lm(yt~x2+x5+x6+x7+x9+x10+x11, winedata)
anova(winedata.reducedmodel, winedata.fullmodel)
```

```
## Analysis of Variance Table
##
## Model 1: yt ~ x2 + x5 + x6 + x7 + x9 + x10 + x11
## Model 2: yt ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1   1591 403.48
## 2   1587 402.80  4   0.67633 0.6662 0.6155
```

With this, we obtain $F_s = 0.6662$ and $p = 0.6155$.

5.2.3 Conclusion

At $\alpha = 0.01$, $p \geq \alpha$, thus, we fail to reject the null hypothesis in favor of the alternate hypothesis. This indicates that we have somee evidence which shows that the reduced model is indeed a better choice.

5.3 k-Fold Cross Validation

We can perform k-fold cross validation with this new model to see how the model would generalize and behave on unseen data:

```r
set.seed(42)
train.control <- trainControl(method='cv', number=5)
step.model <- train(yt~x2+x5+x6+x7+x9+x10+x11, data=winedata,
                    method="leapBackward", tuneGrid=data.frame(nvmax=7),
                    trControl=train.control)
step.model$results
```

```
##   nvmax      RMSE  Rsquared      MAE     RMSESD RsquaredSD      MAESD
## 1     7 0.5044805 0.3553564 0.391121 0.02751353 0.05701186 0.02293553
```

where the actual RMSE and Rsquared (adjusted) values are:

```r
rmse <- sqrt(mean(winedata.reducedmodel$residuals^2))
rs <- summary(winedata.reducedmodel)$adj.r.squared
c(rmse, rs)
```

```
## [1] 0.5023281 0.3553803
```

Clearly, the deviation of these values in 5-fold cross validation is not much from these values on the entire dataset. This implies good generalizability of the model.

6 Conclusion

The research question we sought to answer was whether any of the 11 aforementioned physicochemical characteristics of a wine affect its perceived quality. To answer this question, we performed multiple linear regression on a Red Wine Quality dataset we obtained from kaggle.com. After performing diagnostics and remedial measures on the data, we selected a linear model that used seven out of the eleven physicochemical characteristics to predict perceived quality. Our final, reduced model was able to account for approximately 35.54% of the variation in the data. Some of the error unexplained by the model can be attributed to the limitation of using a continuous linear model to predict a discrete output.

Nevertheless, we believe the model is useful for predicting perceived taste. Consequently, our model has real world implications. Businesses in the wine industry can use the results from our model to make wine that maximizes the characteristics with positive coefficients and minimizes those with negative coefficients. In this way companies can create better tasting wines and improve their customers' satisfaction.