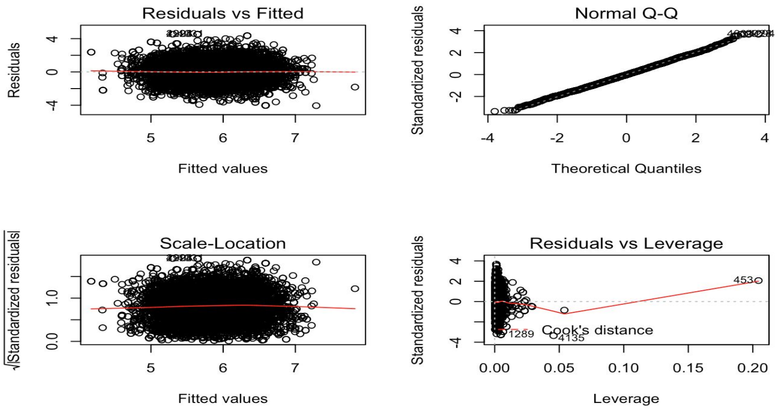
Suoyi Yang
ID: 304449872
Lec1 Disc1A

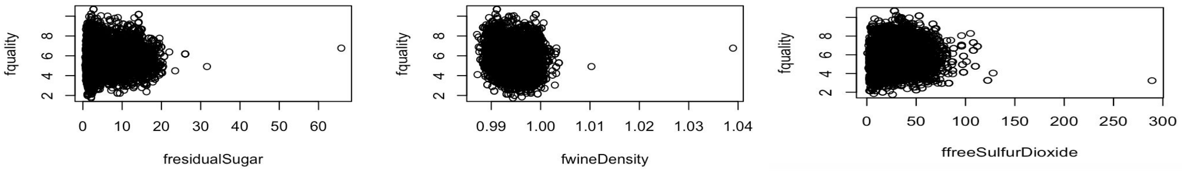
Statistics 101A Project Report

I first started my project by reading in the Wine2017Training.csv file from the website. I took all the numerical predictors and created a multiple linear regression to check the validity of the model. The diagnostics were fairly good to being with.



The errors seem to be independent from one another and have constant variance for the most part. Although there were some points towards the end of the Normal QQ graph that did not follow the normal line exactly, most of the errors followed the normal line pretty well. When it came to the leverage graph, there were some points that I felt were necessary to take notice of, particularly points 4135 and 453. 453 seemed to be a good leverage point while 4135 looked to be an outlier. Because neither were bad leverage points, I left them alone for the meantime.

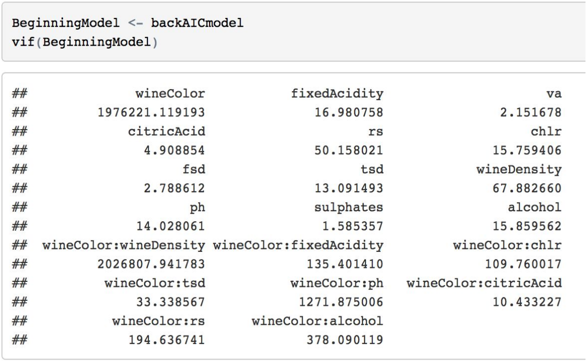
I then moved on to finding a suitable transformation for the predictors and response variables. I first used the mmps() function to determine what transformations were necessary, if any. From the graphs, there were some variables that needed transformation, particularly 'free sulfur dioxide', 'residual sugar', and 'wine density'. However, looking at the mmps output, the model line diverged from the data line mainly due to the presence of extreme points in the graphs, and thus I decided to investigate them a little before doing transformation. I created a response vs. predictor graph for each predictor and found one extreme point for each of them.



I used some R code and found that the extreme point for 'residual sugar' and 'wine density' was the point 453, while the extreme point for 'free sulfur dioxide' was 4135. This matched the points that I was wary about before when I was initially performing diagnostics, and thus I deleted them from my data. I then used mmps() once again, and the graphs were much better this time, and required less severe transformations. After referring to the mmps() graphs and looking at regression results that produced the highest R2, I decided to transform ‘volatile acidity’, ‘residual sugar’, ‘chloride’, ‘free sulfur dioxide’, and ‘total sulfur dioxide’. I performed diagnostics again after the transformation, and the multiple linear regression model was still valid.

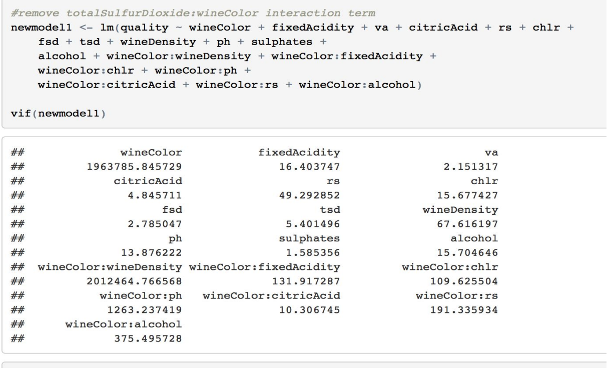
I took all the numerical predictors that I got after the transformation, and combined them with all the wineColor:numericalPredictor interactions to form a very large multiple regression model. I then put the large model into BIC and AIC functions (both backwards and forwards), and got four different results about which model would be the best. Forward BIC model gave me the least predictors while backward AIC gave me the most predictors to use in my model (which is consistent with what we learned in class). However, when I ran the VIF for all of my four models, three out of the four had severe multicollinearity violations, with the only one free of violation being the smallest model I got form the forward BIC. However, this small model was definitely not sufficient enough, and thus I decided to take the model with the most predictors, and slowly drop predictors by referring to the other three models and checking ANOVA and vif() in order to get a model that has VIFs scores all under 5 and which minimized SSE the most.

First, I ran some code to find the correlation between the response variable ‘quality’ and all the rest of the predictors. ‘Quality’ was most highly correlated with ‘wine density’ and ‘alcohol’. I then ran a correlation for ‘alcohol’ and ‘wine density’ and found that they were very highly correlated (-0.7003862). So I suspected that I would eventually need to eliminate ‘wine density’ in order to prevent multicollinearity issues (because it is less correlated with ‘quality’ out of the two). But I first looked at the VIF between each numerical predictor in the model and their corresponding interaction term with ‘wine color’. I immediately found that ‘total sulfur dioxide’ had VIF violation with its interaction term (5.736625), and thus I dropped ‘total sulfur dioxide’ and ‘wine color’ interaction from the model. Besides that, all the other numerical terms had no immediate multicollinearity problem with their corresponding interaction term.



```
BeginningModel <- backAICmodel
vif(BeginningModel)
```

```
##              wineColor           fixedAcidity                     va
##         1976221.119193              16.980758               2.151678
##             citricAcid                     rs                   chlr
##               4.908854              50.158021              15.759406
##                    fsd                    tsd            wineDensity
##               2.788612              13.091493              67.882660
##                     ph              sulphates                alcohol
##              14.028061               1.585357              15.859562
##  wineColor:wineDensity wineColor:fixedAcidity         wineColor:chlr
##         2026807.941783             135.401410             109.760017
##          wineColor:tsd           wineColor:ph   wineColor:citricAcid
##              33.338567            1271.875006              10.433227
##           wineColor:rs      wineColor:alcohol
##             194.636741             378.090119
```



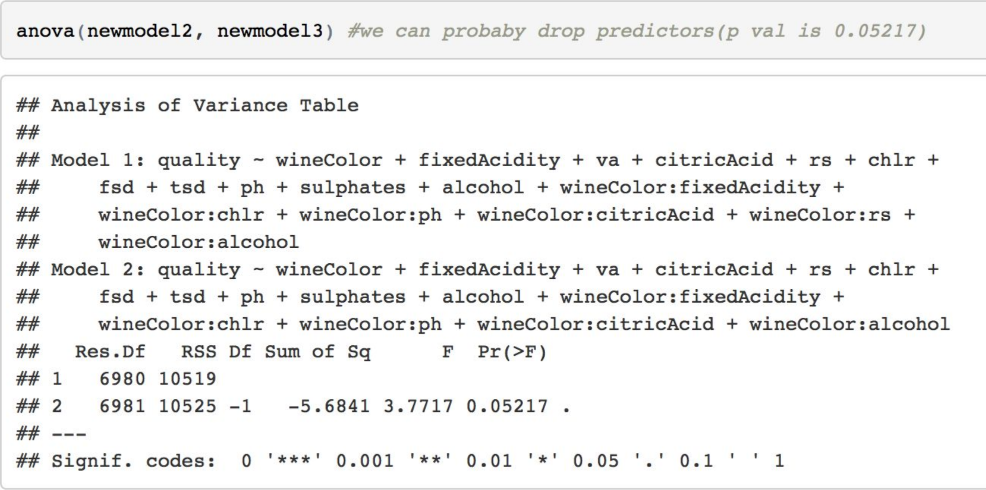
```
#remove totalSulfurDioxide:wineColor interaction term
newmodel1 <- lm(quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
    fsd + tsd + wineDensity + ph + sulphates +
    alcohol + wineColor:wineDensity + wineColor:fixedAcidity +
    wineColor:chlr + wineColor:ph +
    wineColor:citricAcid + wineColor:rs + wineColor:alcohol)

vif(newmodel1)
```

```
##              wineColor           fixedAcidity                     va
##         1963785.845729              16.403747               2.151317
##             citricAcid                     rs                   chlr
##               4.845711              49.292852              15.677427
##                    fsd                    tsd            wineDensity
##               2.785047               5.401496              67.616197
##                     ph              sulphates                alcohol
##              13.876222               1.585356              15.704646
##  wineColor:wineDensity wineColor:fixedAcidity         wineColor:chlr
##         2012464.766568             131.917287             109.625504
##           wineColor:ph   wineColor:citricAcid           wineColor:rs
##            1263.237419              10.306745             191.335934
##      wineColor:alcohol
##             375.495728
```

I then moved onto using VIF on the entire model, and found the ‘wine density’ had by far the largest VIF (67.616197), indicating that it is already highly predictable by the other predictors in the model. Therefore, I removed it and it’s interaction term with ‘wine color’ from my model. Although it significantly increased my SSE (from 10465.8 to 10519.2), it was a necessary step to eliminate violations from my model. Next, ‘residual sugar’ also has a very large VIF

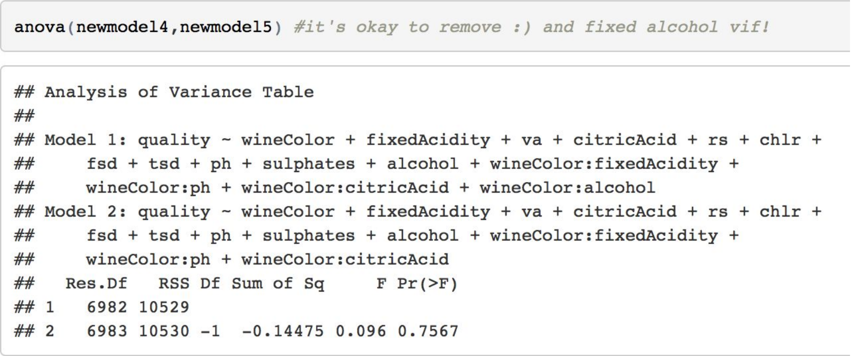
(29.276015). From the anova results, the ‘residual sugar’ interaction term was not very significant and so I removed the interaction term between ‘residual sugar’ and ‘wine color’ from my model. It dropped the VIF for ‘residual sugar’ down to 1.613318. Running a partial-f test revealed that the dropping of the interaction term was not too significant (p-val: 0.05217).



```
anova(newmodel2, newmodel3) #we can probaby drop predictors(p val is 0.05217)
```

```
## Analysis of Variance Table
##
## Model 1: quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
##     fsd + tsd + ph + sulphates + alcohol + wineColor:fixedAcidity +
##     wineColor:chlr + wineColor:ph + wineColor:citricAcid + wineColor:rs +
##     wineColor:alcohol
## Model 2: quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
##     fsd + tsd + ph + sulphates + alcohol + wineColor:fixedAcidity +
##     wineColor:chlr + wineColor:ph + wineColor:citricAcid + wineColor:alcohol
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1   6980 10519
## 2   6981 10525 -1   -5.6841 3.7717 0.05217 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

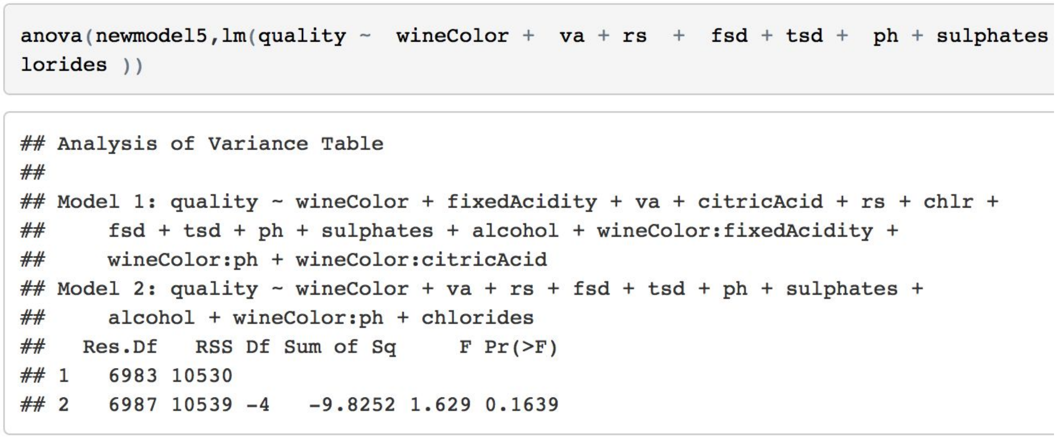
Next up was the ‘chloride’ violation. It was also easy to decide on a term to drop for this term because its corresponding interaction term did not contribute much to the SSE. Indeed dropping the interaction term for ‘chloride’ and ‘wine color’ significantly improved the VIF for chloride (from 14.572420 to 2.412914). My SSE was now at 10519.2. I also decided to drop the interaction between ‘alcohol’ and ‘wine color’ because it contributed close to 0 towards the SSR and alcohol had a VIF violation (5.969191).



```
anova(newmodel4,newmodel5) #it's okay to remove :) and fixed alcohol vif!
```

```
## Analysis of Variance Table
##
## Model 1: quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
##     fsd + tsd + ph + sulphates + alcohol + wineColor:fixedAcidity +
##     wineColor:ph + wineColor:citricAcid + wineColor:alcohol
## Model 2: quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
##     fsd + tsd + ph + sulphates + alcohol + wineColor:fixedAcidity +
##     wineColor:ph + wineColor:citricAcid
##   Res.Df   RSS Df Sum of Sq     F Pr(>F)
## 1   6982 10529
## 2   6983 10530 -1  -0.14475 0.096 0.7567
```

The most difficult part was trying to reduce the VIF for ‘ph’ (8.709002). The easiest options were to drop either the interaction for ‘ph’ and ‘wine color’ or the categorical variable ‘wine color’ itself. However, when I tried both options, partial-f test revealed that the drop in SSE was much too significant to sacrifice either one. Thus after multiple trials of dropping certain terms, checking if they decreased ‘ph’ significantly, adding it back in if they did not, and finding another term to drop, I finally decided to drop four terms: ‘fixed acidity’, its interaction term with ‘wine color’, ‘citric acid’, and its interaction with ‘wine color’. Checking the partial-f test revealed a p-value of 0.1639, indicating that this was the better choice compared to dropping the ‘ph’ interaction or the categorical variable ‘wine color’ itself. However, my SSE was now very high: 10540.7. Thus although this model gave me no VIF violation and was still a better model than the forward BIC suggestion (which had an SSE of 10588.8), tried to add more terms in to see if they would be able to decrease the SSE further without introducing VIF violations.

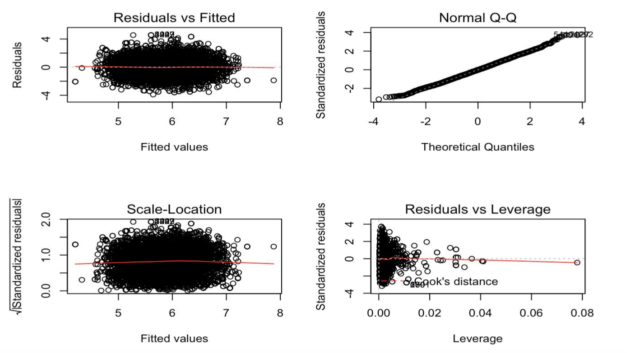


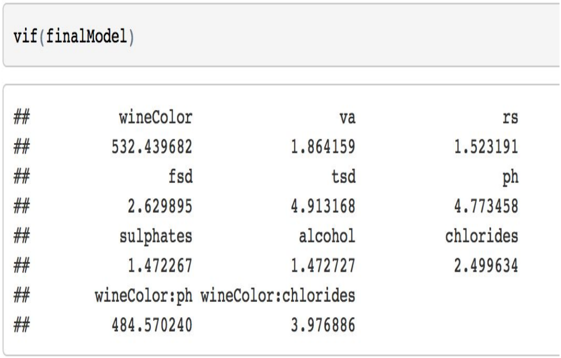
```
anova(newmodel5,lm(quality ~  wineColor +  va + rs  +  fsd + tsd +  ph + sulphates
lorides ))
```

```
## Analysis of Variance Table
##
## Model 1: quality ~ wineColor + fixedAcidity + va + citricAcid + rs + chlr +
##     fsd + tsd + ph + sulphates + alcohol + wineColor:fixedAcidity +
##     wineColor:ph + wineColor:citricAcid
## Model 2: quality ~ wineColor + va + rs + fsd + tsd + ph + sulphates +
##     alcohol + wineColor:ph + chlorides
##   Res.Df   RSS Df Sum of Sq     F Pr(>F)
## 1   6983 10530
## 2   6987 10539 -4   -9.8252 1.629 0.1639
```

After a couple of trials and error, I was able to include in my model: ‘chloride’ and the interaction term between ‘chloride’ and ‘wine color’. Unlike before, it seems like leaving chloride untransformed gave me the a larger SSR (and consequently a smaller SSE). My final model was the following :

Quality = (2.439228*wine color) + (-0.467420*log(volatile acidity)) + (0.302457*(residual sugar^(⅓)) + (0.343986*free sulfur dioxide^(⅓)) + (-0.024764*total sulfur dioxide^(⅔)) + (-0.588179*ph) + (0.727423*sulphates)+ (0.342590*alcohol) + (1.344183*wine color*ph + (-1.395056*chlorides) + (1.354136*chlorides*wine color)





```
vif(finalModel)
```

```
##          wineColor                 va                 rs
##         532.439682           1.864159           1.523191
##                fsd                tsd                 ph
##           2.629895           4.913168           4.773458
##          sulphates            alcohol          chlorides
##           1.472267           1.472727           2.499634
##       wineColor:ph wineColor:chlorides
##         484.570240           3.976886
```

The most challenging part of the project for me was trying to eliminate the VIF violation for the model. It was the part that reduced my SSR (and increased my SSE) the most. Sometimes, I would try to hold onto a predictor (say X1) because the anova indicated significant contribution toward the SSR and instead try to drop other predictors in my model only to find out the X1 must be dropped in order to fix the VIF violation. Then I would have to go back to the predictors I previously sacrificed for the sake of trying to keep X1 in my model, add them back into my model and see if they can makeup for the loss of SSR.