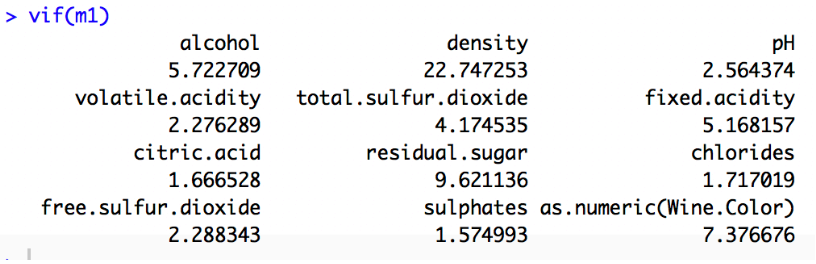
Lara Vartanian
804633245

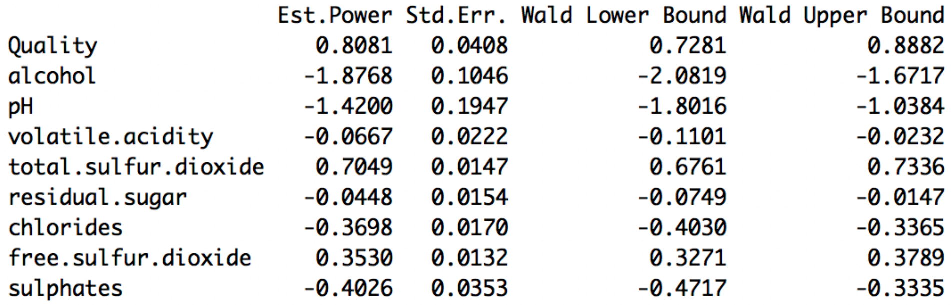
Stats 101 A Final Project- Wine Quality Prediction

To help me test how good of a model I am building, I have partitioned the training data of 7000 that includes wine quality into 70% of it being my training and 30% testing data set. My very first attempt was to do a multiple linear regression using all the predictors with no transformation; I got an R squared of 13.4%. The top four sum of square regression contributors were density, pH, volatile acidity, and total sulfur dioxide. I also made sure to check each variable's correlation with wine quality and noted that alcohol, density, chlorides, and volatile acidity were the most correlated variables with quality of wine. Looking at the VIF of my first model (m1), I noticed that there was a multicollinearity issue with density, since it was 22, well above the accepted level of 5.

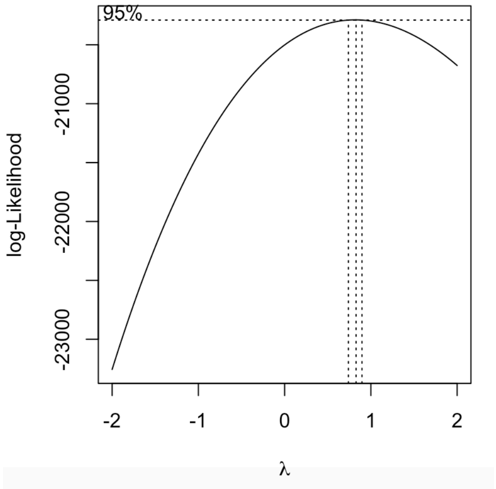


```
> vif(m1)
               alcohol                density                     pH
              5.722709              22.747253               2.564374
      volatile.acidity   total.sulfur.dioxide          fixed.acidity
              2.276289               4.174535               5.168157
           citric.acid         residual.sugar              chlorides
              1.666528               9.621136               1.717019
   free.sulfur.dioxide              sulphates as.numeric(Wine.Color)
              2.288343               1.574993               7.376676
```

Hence, I decided to remove density; I went ahead and removed citric acid as well because of its insignificance and use of partial F test. Using the boxcox and powerTransform too see the suggested lambdas, I transformed my variables corresponding with their suggested lambdas. I did not transform quality after noting both the boxcox and powerTransform outputs. We see the powerTransform output below.

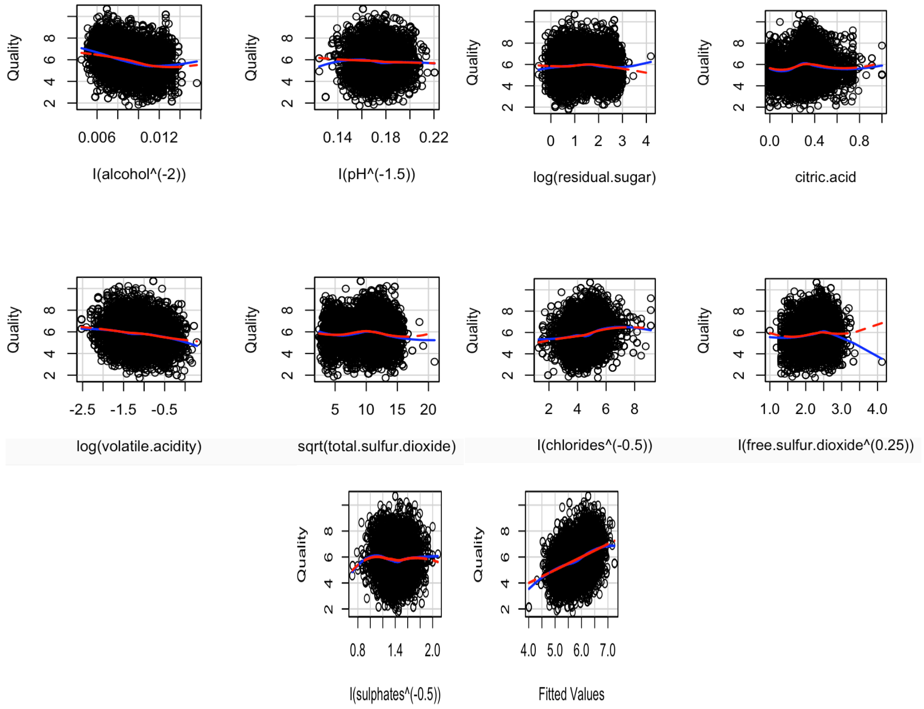


```
                     Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
Quality                 0.8081   0.0408           0.7281           0.8882
alcohol                -1.8768   0.1046          -2.0819          -1.6717
pH                     -1.4200   0.1947          -1.8016          -1.0384
volatile.acidity       -0.0667   0.0222          -0.1101          -0.0232
total.sulfur.dioxide    0.7049   0.0147           0.6761           0.7336
residual.sugar         -0.0448   0.0154          -0.0749          -0.0147
chlorides              -0.3698   0.0170          -0.4030          -0.3365
free.sulfur.dioxide     0.3530   0.0132           0.3271           0.3789
sulphates              -0.4026   0.0353          -0.4717          -0.3335
```



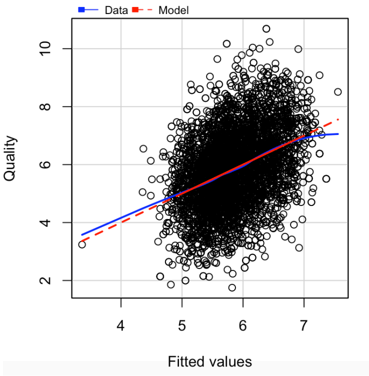
Boxcox output

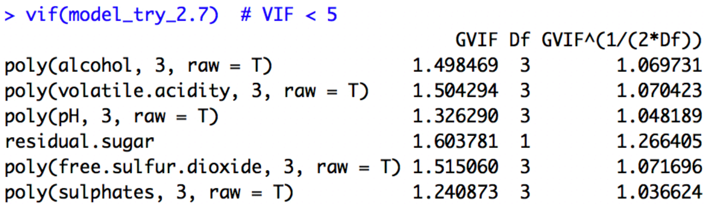
I transformed my next model accordingly to: m2 <- lm(Quality ~ I(alcohol^(-2)) + I(pH^(-1.5)) + log(volatile.acidity) + sqrt(total.sulfur.dioxide) + log(residual.sugar) + I(chlorides^(-.5)) + I(free.sulfur.dioxide^(.25)) + I(sulphates^(-.5)), data = train). I used forward AIC to determine my variables. I checked the major four assumptions to see if there are any violations. The residuals follow a Normal Distribution, seem to be independent and have constant variance with no bad leverage points. So, overall it is valid, however I further work to create another model to improve my R-squared (currently at 13.84%) and decrease the sum of squared residuals that I run through every time to compare my models and how well each predict the Quality on the 30% testing data I created. I used marginal model plots to see how fitting my transformed model was and after looking at the output, it seemed to fit well. The red line is based on my model and the blue line is the fit based on the data.



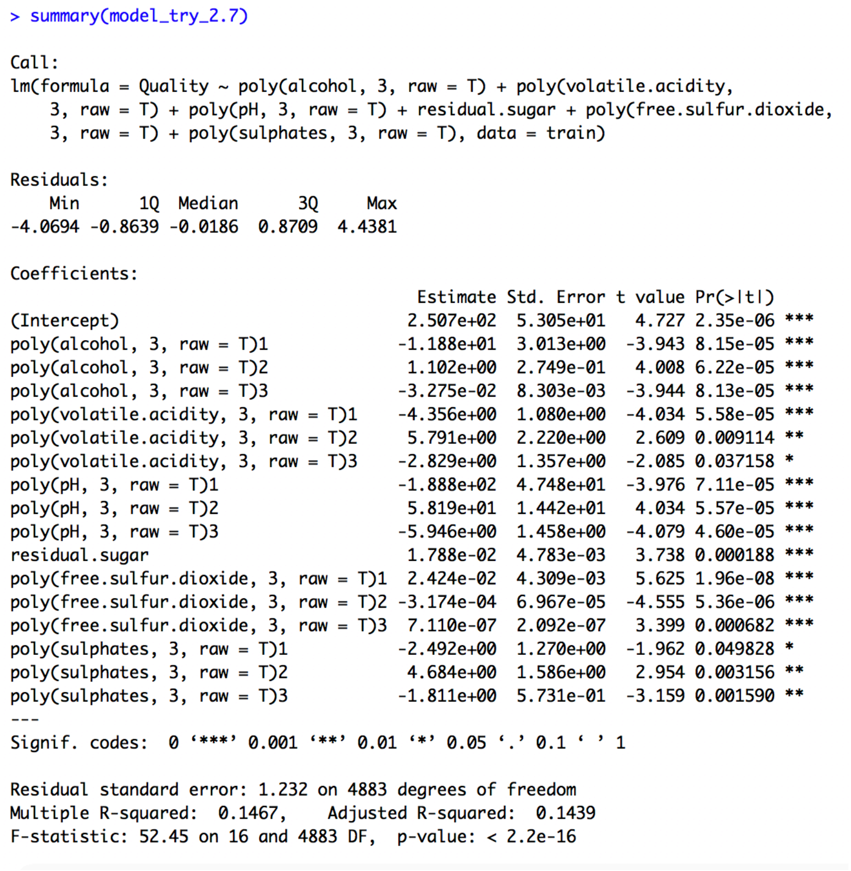
Next, I wanted to see if a polynomial regression gave a better model, so I attempted to create a polynomial regression, applying poly function to the significant variables to see if my R-squared would increase relative to the model with transformations suggested with function powerTransform. I started off with the following model: model_try_2 <- lm(Quality ~ poly(alcohol, 3, raw = T)+ poly(volatile.acidity, 3, raw = T)+ poly(total.sulfur.dioxide, 3, raw = T) + poly(pH, 3, raw = T) + poly(residual.sugar, 1, raw = T) + poly(free.sulfur.dioxide, 3, raw = T) + poly(sulphates, 3, raw = T), data = train). I checked the VIF and noticed a multicollinearity problem. Hence I removed total sulfur dioxide. My next model is: model_try_2.7 <- lm(Quality ~ poly(alcohol, 3, raw = T)+ poly(volatile.acidity, 3, raw = T) + poly(pH, 3, raw = T) + residual.sugar + poly(free.sulfur.dioxide, 3, raw = T) + poly(sulphates, 3, raw = T), data = train). It has R squared of 14.67 and no VIF violation since respective VIFs are less than 5. I used the partial F test to test whether I should add chlorides and the output showed it was not significant, hence it was not worth adding chlorides. I continuously kept testing every new model's SSE based on my 70/30 strategy.

Lastly, I wanted to check for interactions. I tried to see all the significant interactions and added some of them to my model. I was able to get R squared of 16.15% by adding interactions between free sulfur dioxide and wine color, wine color and pH, volatile acidity and pH, and chlorides and pH; however, upon looking at the VIF output, GVIF was high. Therefore I am going with the polynomial model without interactions. I have checked the diagnostics; the variables do not have high VIF and the residuals seem to be independent, have constant variance, and are normally distributed. Looking at my final marginal model plot, my model seems to be a good fit for the data points. My R squared is 14.67%. Therefore, I use this final model to predict the Wine Quality on the 3000 testing data. This means that all the predictors I have in my model including alcohol, volatile acidity, pH, residual sugar, free sulfur dioxide and sulphates explain 14.67% of the variation in wine quality. There may be better predictors to predict wine quality other than chemical components that would probably explain the variation in Quality better, however for this data set, I was strictly limited to wine's chemical components.





```
> vif(model_try_2.7)  # VIF < 5
                                     GVIF Df GVIF^(1/(2*Df))
poly(alcohol, 3, raw = T)          1.498469  3        1.069731
poly(volatile.acidity, 3, raw = T)  1.504294  3        1.070423
poly(pH, 3, raw = T)               1.326290  3        1.048189
residual.sugar                     1.603781  1        1.266405
poly(free.sulfur.dioxide, 3, raw = T) 1.515060  3        1.071696
poly(sulphates, 3, raw = T)        1.240873  3        1.036624
```



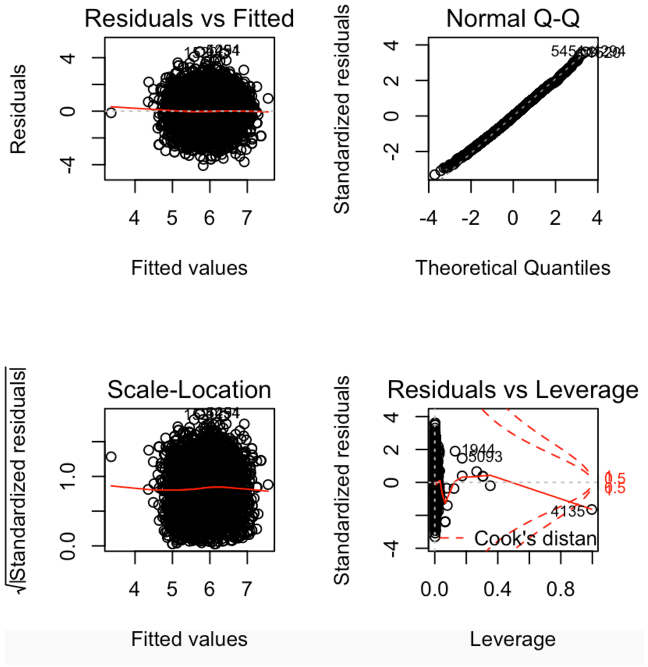
```
> summary(model_try_2.7)

Call:
lm(formula = Quality ~ poly(alcohol, 3, raw = T) + poly(volatile.acidity,
    3, raw = T) + poly(pH, 3, raw = T) + residual.sugar + poly(free.sulfur.dioxide,
    3, raw = T) + poly(sulphates, 3, raw = T), data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-4.0694 -0.8639 -0.0186  0.8709  4.4381

Coefficients:
                                         Estimate Std. Error t value Pr(>|t|)
(Intercept)                             2.507e+02  5.305e+01   4.727 2.35e-06 ***
poly(alcohol, 3, raw = T)1             -1.188e+01  3.013e+00  -3.943 8.15e-05 ***
poly(alcohol, 3, raw = T)2              1.102e+00  2.749e-01   4.008 6.22e-05 ***
poly(alcohol, 3, raw = T)3             -3.275e-02  8.303e-03  -3.944 8.13e-05 ***
poly(volatile.acidity, 3, raw = T)1    -4.356e+00  1.080e+00  -4.034 5.58e-05 ***
poly(volatile.acidity, 3, raw = T)2     5.791e+00  2.220e+00   2.609 0.009114 **
poly(volatile.acidity, 3, raw = T)3    -2.829e+00  1.357e+00  -2.085 0.037158 *
poly(pH, 3, raw = T)1                  -1.888e+02  4.748e+01  -3.976 7.11e-05 ***
poly(pH, 3, raw = T)2                   5.819e+01  1.442e+01   4.034 5.57e-05 ***
poly(pH, 3, raw = T)3                  -5.946e+00  1.458e+00  -4.079 4.60e-05 ***
residual.sugar                          1.788e-02  4.783e-03   3.738 0.000188 ***
poly(free.sulfur.dioxide, 3, raw = T)1  2.424e-02  4.309e-03   5.625 1.96e-08 ***
poly(free.sulfur.dioxide, 3, raw = T)2 -3.174e-04  6.967e-05  -4.555 5.36e-06 ***
poly(free.sulfur.dioxide, 3, raw = T)3  7.110e-07  2.092e-07   3.399 0.000682 ***
poly(sulphates, 3, raw = T)1           -2.492e+00  1.270e+00  -1.962 0.049828 *
poly(sulphates, 3, raw = T)2            4.684e+00  1.586e+00   2.954 0.003156 **
poly(sulphates, 3, raw = T)3           -1.811e+00  5.731e-01  -3.159 0.001590 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.232 on 4883 degrees of freedom
Multiple R-squared:  0.1467,    Adjusted R-squared:  0.1439
F-statistic: 52.45 on 16 and 4883 DF,  p-value: < 2.2e-16
```


Residuals vs Fitted
Residuals
Fitted values
4
5
6
7
-4
0
4
Normal Q-Q
Standardized residuals
Theoretical Quantiles
-4
-2
0
2
4
545
Scale-Location
√|Standardized residuals|
0.0
1.0
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
0.0
0.4
0.8
1944
5093
4135
0.5
0.5
Cook's distan