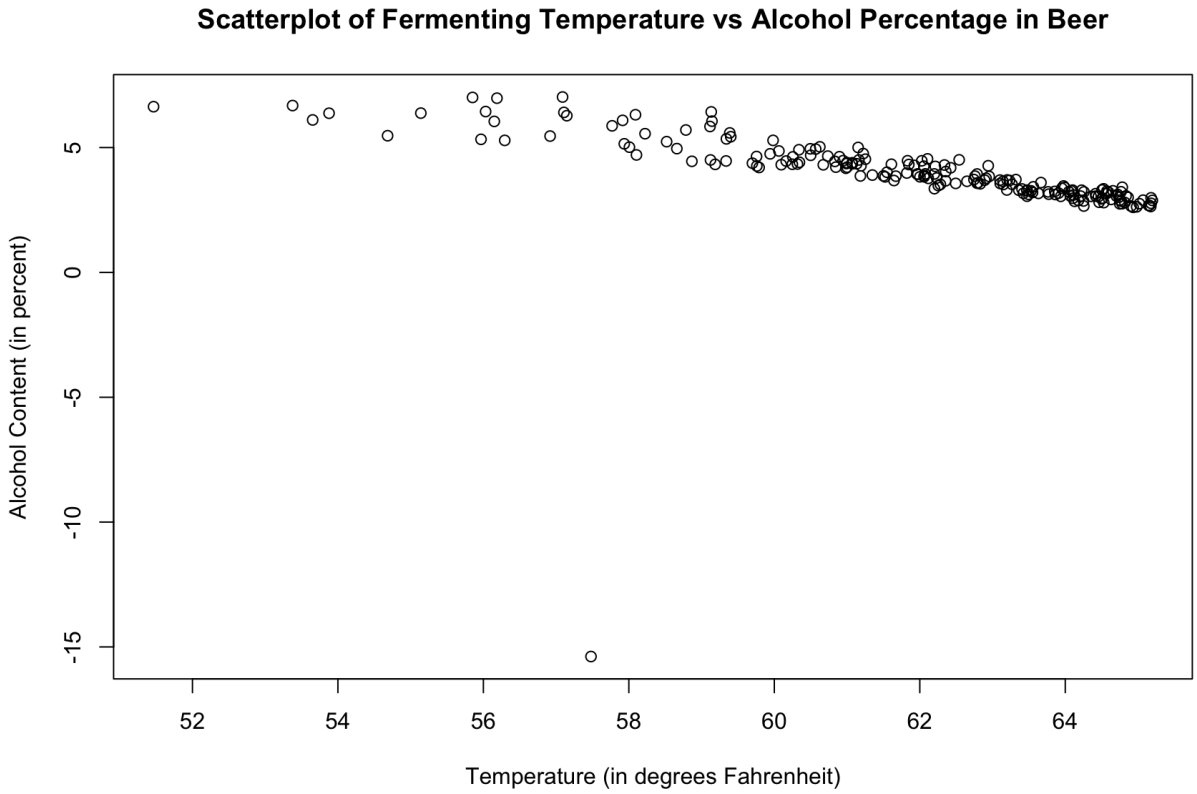
STAT 463 - Assignment 3

Due Date: Saturday, March 11th, 2023

Data Description and Background

An amateur brewer wishes to better understand how the temperature that the beer ferments at (in degrees Fahrenheit) affects the alcohol content of the beer upon completion of brewing. Fortunately, the brewer has kept copious notes on his many brewing endeavors and has records for each batch on what temperature the beer fermented at and what the final alcohol content was. You are provided with two datasets, one with data from 200 batches and another with data from 50 batches. Call these datasets the modeling dataset and validation dataset respectively. You are to do the following:

1) Using the modeling dataset, visually display the data in an appropriate graph and comment on anything that may be of note. In particular, are the assumptions needed for fitting the simple linear model met? – 1 point

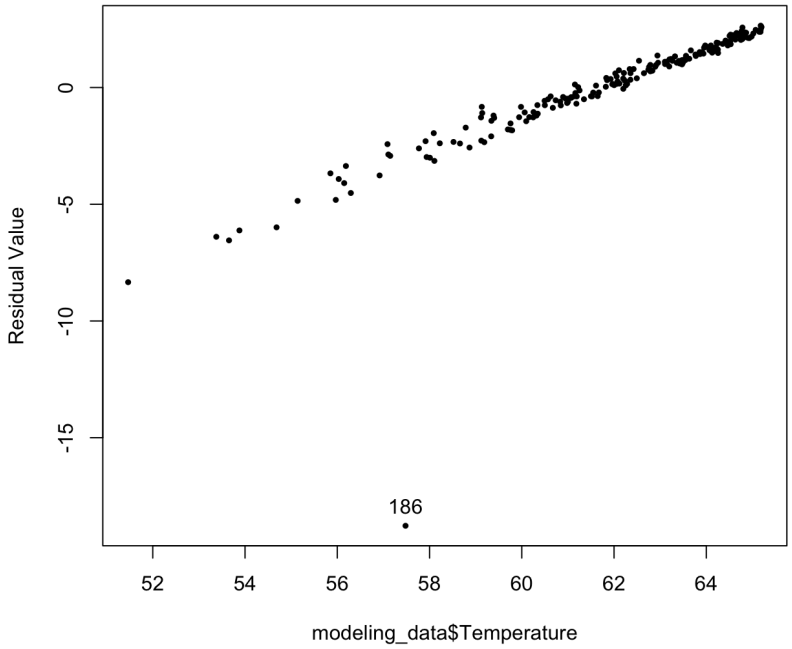


From the plot, it seems like the general trend is that higher fermenting temperatures tend to produce beers with lower alcohol content. The linearity assumption is met because the general trend is that while temperature increases, the alcohol content decreases at a ‘roughly’ constant rate. Therefore, the assumption that the relationship between fermenting temperature and alcohol content is linear seems like a plausible conclusion. For any given temperature, the variation of recession velocities mostly appears to be the same, being distributed evenly above and below where the regression line would fit through the data, thus justifying assumptions on the errors.

2) Initially, the brewer would just like to get a rough estimate of what the alcohol content would be if he ferments the batch at a given temperature. Are enough of the regression assumptions satisfied so that simple linear regression can be used towards the prior-mentioned goal? If not, what deviations do you need to address and how do you address them? – 2 point

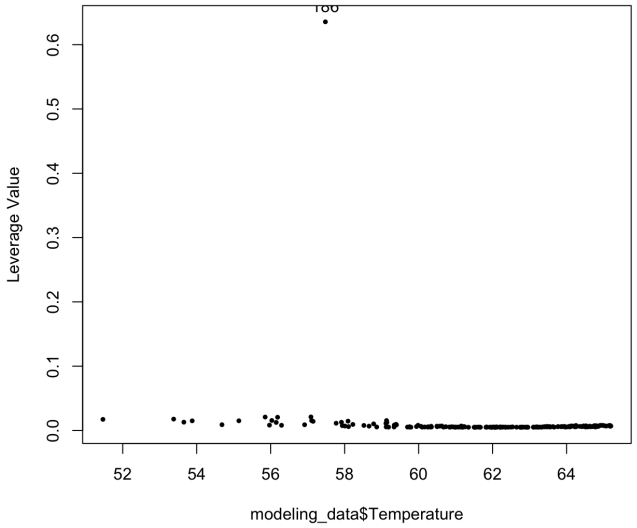
There appears to be an outlier in the data, point 186: (57.47955, -15.38462).
Based on the context of the data, this outlier could be removed since it might've been an error.
Run Diagnostic Tests to see if this is a true outlier.

```
> mod1 = lm(Temperature ~ Alcohol_Percentage, data = modeling_data)
> summ1 = summary(mod1)
> summ1

Call:
lm(formula = Temperature ~ Alcohol_Percentage, data = modeling_data)

Residuals:
    Min      1Q  Median      3Q     Max
-18.7717  -0.8275   0.5387   1.6044   2.6496

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       64.76103    0.42752  151.48  < 2e-16 ***
Alcohol_Percentage -0.74687    0.09944   -7.51 1.97e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.42 on 198 degrees of freedom
Multiple R-squared:  0.2217,      Adjusted R-squared:  0.2178
F-statistic: 56.41 on 1 and 198 DF,  p-value: 1.974e-12

> plot(modeling_data$Temperature, summ1$residuals, ylab = "Residual Value", pch = 19, cex = 0.5)
> identify(modeling_data$Temperature, summ1$residuals)
```



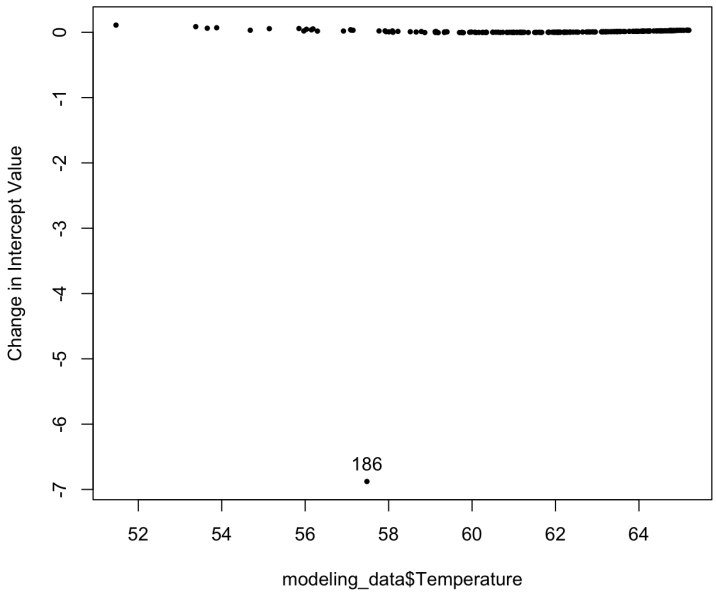
The residual plot indicates that the outlier appears to be a problem.
Leverage:

```
> lev = hatvalues(mod1)
> plot(modeling_data$Temperature, lev, ylab = "Leverage Value", pch = 19, cex = 0.5)
```

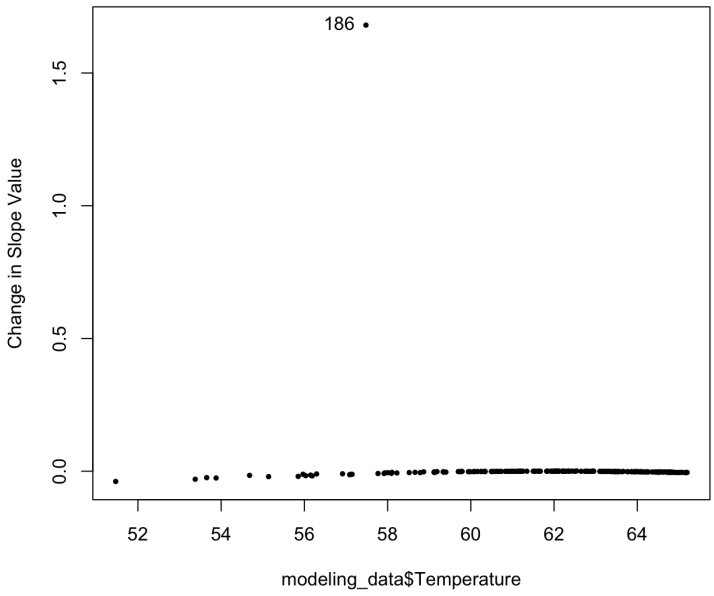


Change in Coefficients:

```
> dif_betas = dfbeta(mod1)
> #Change in Interecept Value
> plot(modeling_data$Temperature, dif_betas[,1], ylab = "Change in Intercept Value", pch = 19, cex = 0.5)
> identify(modeling_data$Temperature, dif_betas[,1])
```



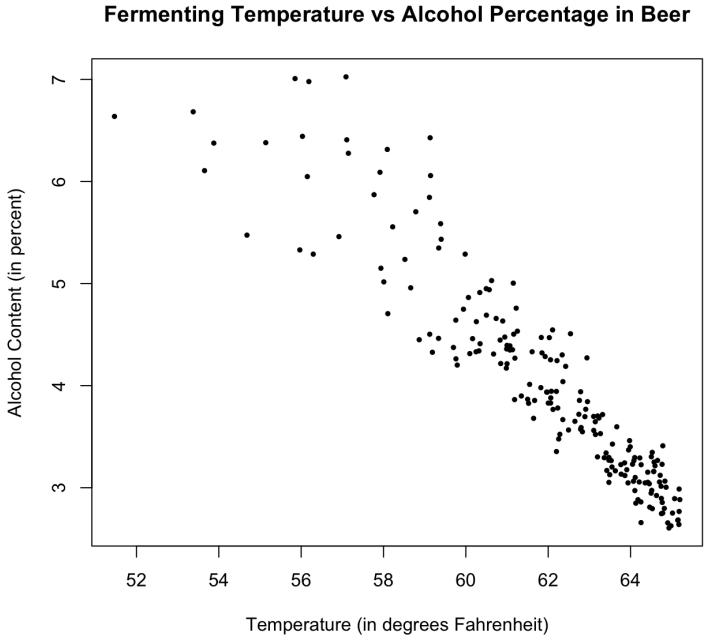

```
> #Change in the Slope Value
> plot(modeling_data$Temperature, dif_betas[,2], ylab = "Change in Slope Value", pch = 19, cex = 0.5)
> identify(modeling_data$Temperature, dif_betas[,2])
```



Remove Outlier:

```
> modeling2 <- modeling_data[-186,]
> mod2 <- lm(Alcohol_Percentage~Temperature, data=modeling2)
> sum2 <- summary(mod2)
```

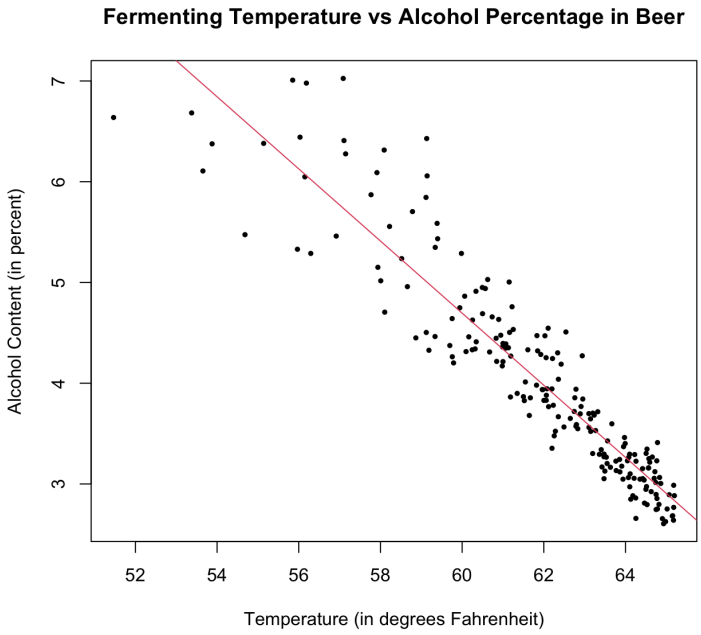
> plot(modeling2[,1],modeling2[,2], main="Fermenting Temperature vs Alcohol Percentage in Beer", xlab="Temperature (in degrees Fahrenheit)",ylab="Alcohol Content (in percent)", pch = 19, cex = 0.5)



We have already concluded that the linearity assumption and the assumption of errors has been met in question 1 because while temperature increases, the alcohol content decreases at a 'roughly' constant rate and the variation of recession velocities mostly appears to be the same, being distributed evenly above and below where the regression line would fit through the data. We also observed the problematic outlier through diagnostic tests and removed it since it is possibly an error considering alcohol content cannot be negative. The scatter plot of the data with the outlier removed provides a better view of the spread of data. There is heteroscedasticity because the variation of the observations around the 'line' is greater for smaller values of the fermenting temperature. The homoscedasticity assumption is violated for this dataset. However, since this regression model is used for prediction purposes, we can say that the linear assumptions are met since the relationship between the explanatory & response variables is assumed to be linear.

3) After addressing any necessary issues in part 2, fit the simple linear model to the data. Provide the parameter estimates and the R2 value. Overlay the estimated regression line on the plot created in part 1

> plot(modeling2[,1],modeling2[,2], main="Fermenting Temperature vs Alcohol Percentage in Beer",xlab="Temperature (in degrees Fahrenheit)",ylab="Alcohol Content (in percent)", pch = 19, cex = 0.5)
> abline(mod2$coe[1], mod2$coe[2], col = 2)

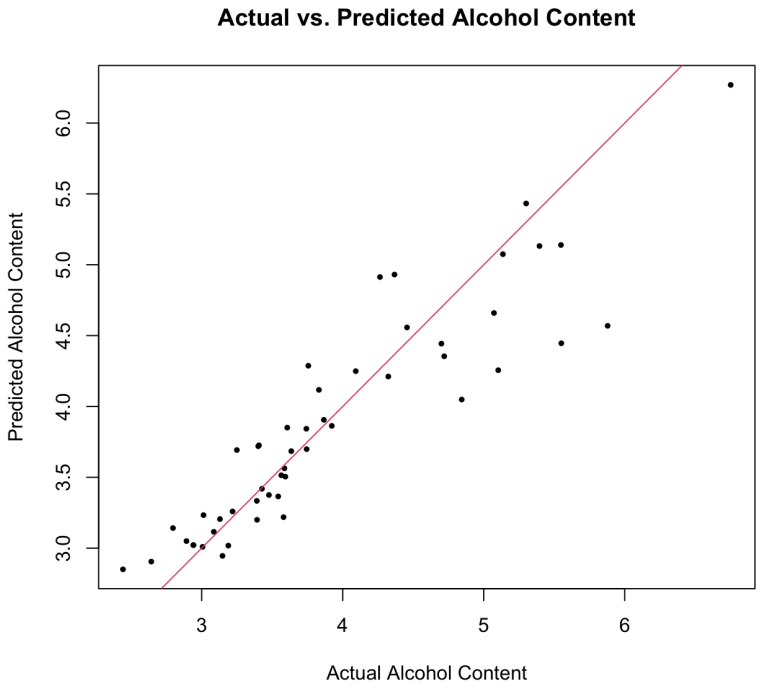



```
> sum2
Call:
lm(formula = Alcohol_Percentage ~ Temperature, data = modeling2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.12419 -0.21920 -0.05328  0.19142  1.42246

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 26.173742   0.613251   42.68   <2e-16 ***
Temperature -0.357968   0.009907  -36.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.38 on 197 degrees of freedom
Multiple R-squared:  0.8689,       Adjusted R-squared:  0.8682
F-statistic:  1306 on 1 and 197 DF,  p-value: < 2.2e-16
```

The Linear regression model is Y= 26.173742 - 0.357968x and R-squared: 0.8689

4)

```
actual<-validation_data$Alcohol_Percentage
predicted<-predict(mod2, newdata = validation_data)
plot(actual, predicted, xlab = 'Actual Alcohol Content', ylab = 'Predicted Alcohol Content', main = 'Actual
vs. Predicted Alcohol Content', pch = 19, cex = 0.5)
abline(0,1, col = 2)
```

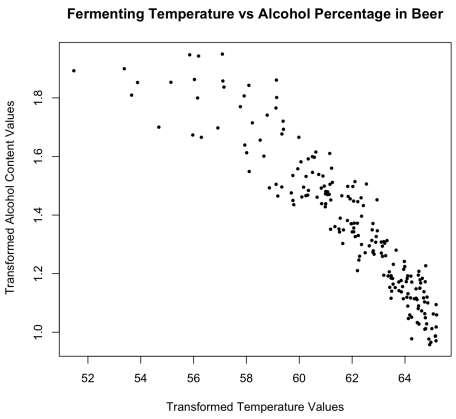


> cor(actual,predicted)
[1] 0.9170591
The correlation between the actual and predicted alcohol content values is close to one indicating that this model is a good model

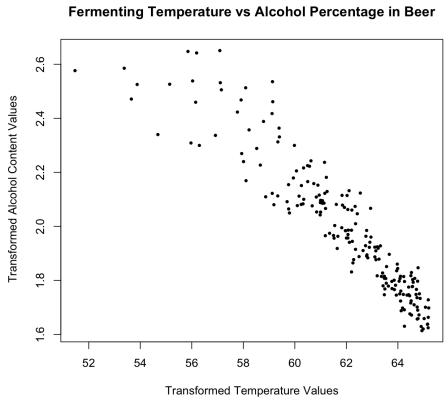
5) Although the modeling data's linearity assumptions and assumptions of errors and outliers had been addressed before since we only needed a predictive model, now we wish to do inference (prediction interval), so we must address the modeling data's heteroscedasticity in order to obtain reliable results.

I experimented with different methods to transform the data to address heteroscedasticity.

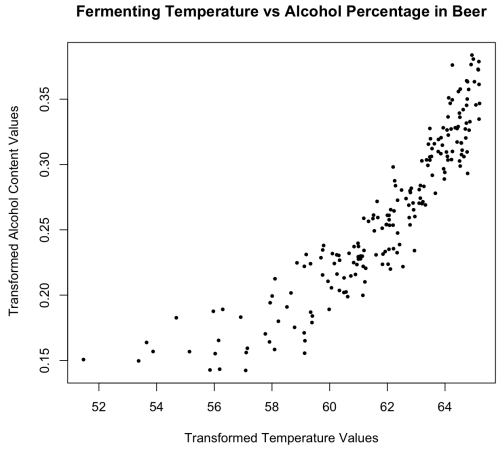
tx <- modeling2[,1]
ty<- log(modeling2[,2])



tx <- modeling2[,1]
ty<- sqrt(modeling2[,2])

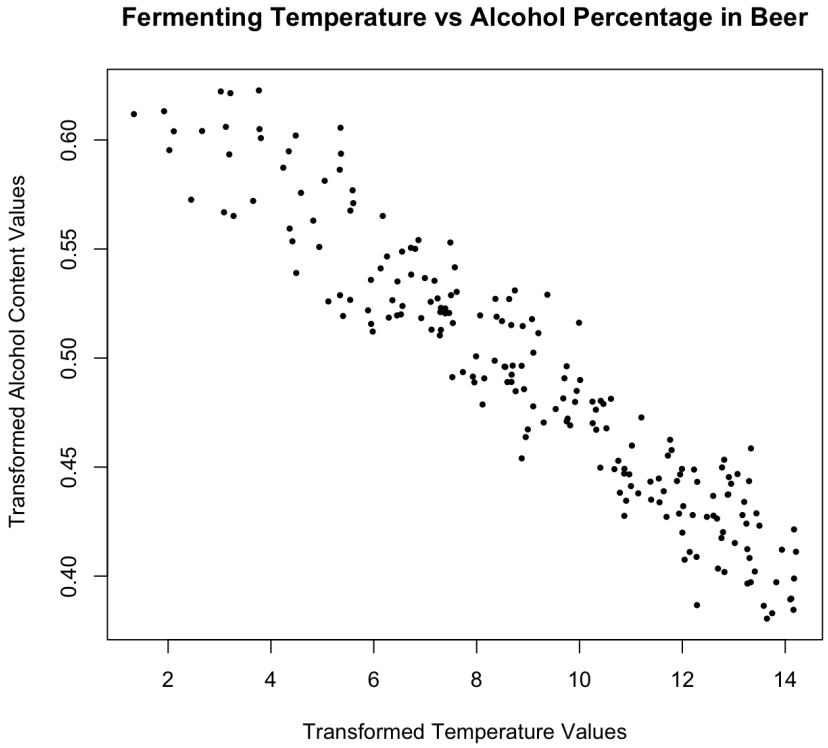


tx <- modeling2[,1]
ty<- 1/(modeling2[,2])



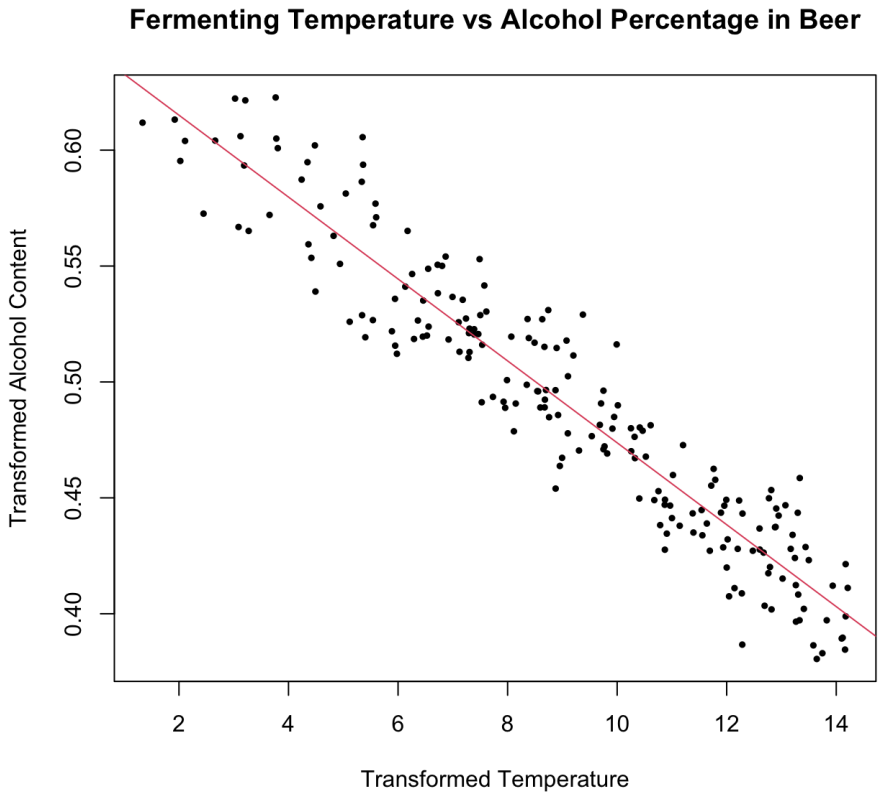
Out of these common methods to address heteroscedasticity, I chose to use 1/sqrt(modeling2[,2]) because it corrected for heteroskedasticity the best. After that I attempted to fix the linearity of the data through a bunch of different methods but landed on the following. And then translated it to the original orientation of the modeling data frame.

Final Transformed Data:
tx <- (0.02*(modeling2[,1]))^10
ty<- -(1/((modeling2[,2])^(1/2)))+1
plot(tx,ty, main="Fermenting Temperature vs Alcohol Percentage in Beer",xlab="Transformed Temperature Values",ylab=" Transformed Alcohol Content Values", pch = 19, cex = 0.5)



6) We decided to correct for linearity already so we can run a regular linear regression model instead of a polynomial regression model which could have also been used if we hadn't already addressed linearity in the previous part.

```
modeling3<-modeling2
modeling3[,1]<-tx
modeling3[,2]<-ty

mod3 <- lm(Alcohol_Percentage~Temperature, data=modeling3)
sum3 <- summary(mod3)
plot(modeling3[,1],modeling3[,2], main="Fermenting Temperature vs Alcohol Percentage in
Beer",xlab="Transformed Temperature",ylab="Transformed Alcohol Content", pch = 19, cex = 0.5)
abline(mod3$coe[1], mod3$coe[2], col = 2)
```



>sum3
Call:
lm(formula = Alcohol_Percentage ~ Temperature, data = modeling3)

Residuals:
      Min   1Q   Median   3Q   Max
-0.046711 -0.013284 -0.001146 0.014424 0.049733

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.6504314 0.0039789 163.47   <2e-16 ***
Temperature -0.0176676 0.0004145 -42.62   <2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.019 on 197 degrees of freedom
Multiple R-squared: 0.9022,	Adjusted R-squared: 0.9017
F-statistic: 1817 on 1 and 197 DF, p-value: < 2.2e-16

ty = 0.6504314 -0.0176676 tx

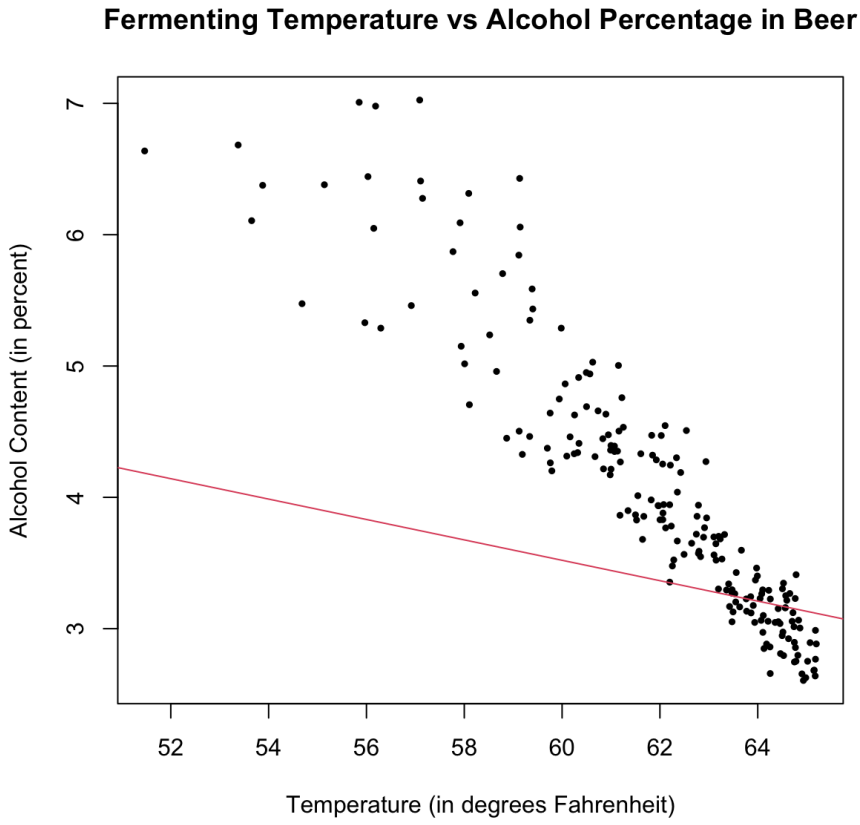
**** tx and ty being the transformed variables**
-(1/((y)^(1/2)))+1 = 0.6504314 - 0.0176676((0.02*(x))^10)

Formula:
y=1/((0.3495686+0.0176676*((0.02*(x))^10))^2)

y=1/(0.3495686+0.0176676*((0.02*(0))^10))^2
y= 8.183426

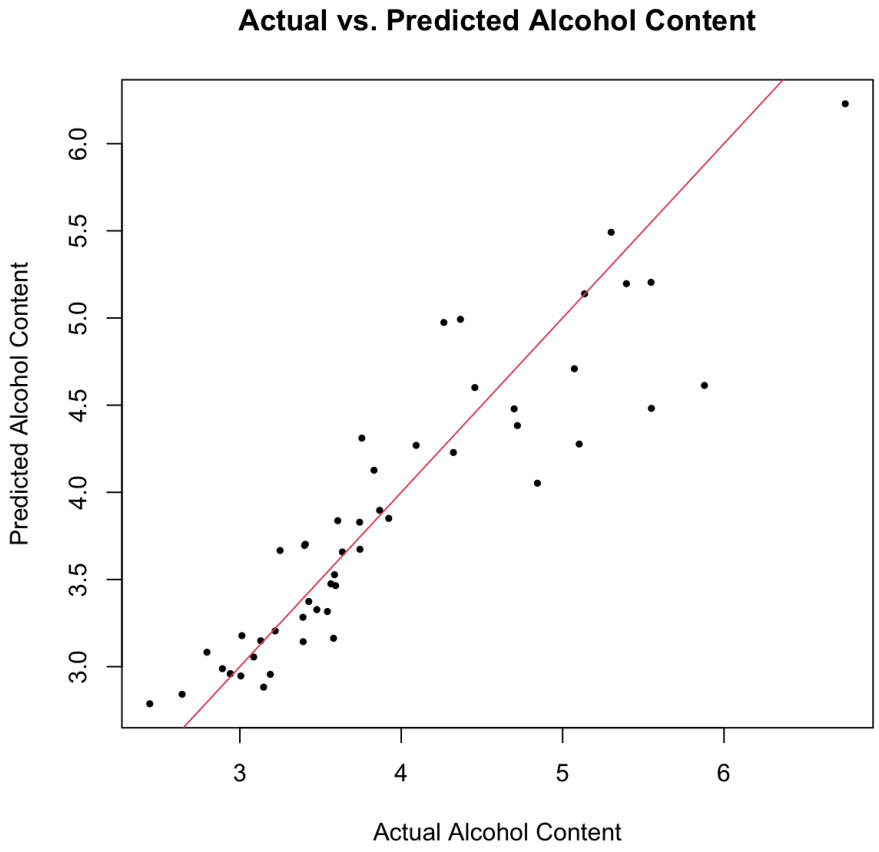
y=1/(0.3495686+0.0176676*((0.02*(64))^10))^2
y= 3.20994

Intercept:8.183426
Slope: -0.07771072



7)
>actual<-validation_data$Alcohol_Percentage
> validation_data2<-validation_data
> validation_data2[,1]<- (0.02*(validation_data2[,1]))^10
> validation_data2[,2]<- -(1/((validation_data2[,2])^(1/2)))+1
> predictedty<-predict(mod3, newdata = validation_data2)
> predicted<-1/(predictedty-1)^2
> plot(actual, predicted, xlab = 'Actual Alcohol Content', ylab = 'Predicted Alcohol Content', main = 'Actual vs. Predicted Alcohol Content', pch = 19, cex = 0.5)
> abline(0,1, col = 2)

> cor(actual,predicted)
[1] 0.916627
The correlation between the actual and predicted alcohol content values is close to one indicating that this model is a good model.



8)
>PI <- predict(mod3, interval = "prediction", level = 0.95)
>PI <- cbind(tx, PI)
>PI = PI[order(PI[,1]),]
>plot(modeling2[,1],modeling2[,2], main="Fermenting Temperature vs Alcohol Percentage in Beer",xlab="Temperature (in degrees Fahrenheit)",ylab="Alcohol Content (in percent)")
>abline(8.183426,-0.07771072, col = 2)
>points((50*((PI[,1])^10)), (1/((PI[,3])-1)^2), type="l", lty=2, col = 3)
>points((50*((PI[,1])^10)), (1/((PI[,4])-1)^2), type="l", lty=2, col = 3)

My Rstudio refused to let the points() function work it ran but nothing showed up on my plot but through prior experience I know this should result in the correct plot with the bands being over laid over the plot.

I was able to get something similar with the package qqplot2 inorder to have something to submit.

```
> PI <- predict(mod3, interval = "prediction", level = 0.95)
> PI <- cbind(tx, PI)
> library(ggplot2)
> PI<-data.frame(PI)
> ggplot(PI, aes(modeling2[,1], modeling2[,2]))+
+   geom_point() +
+   geom_line(aes(y=(1/((lwr)-1)^2)), color = "red", linetype = "dashed")+
+   geom_line(aes(y=(1/((upr)-1)^2)), color = "red", linetype = "dashed")+
+   geom_smooth(method=lm, se=TRUE)
`geom_smooth()` using formula = 'y ~ x'
```

Prediction Bands Plot:

Prediction bands upper and lower bounds are dashed.
modeling2[,2] label should be Alcohol Content (in percent)
Modeling2[,1] label should be Temperature (in degrees Fahrenheit)

Title: Fermenting Temperature vs Alcohol Percentage in Beer