Alcohol Percentage and Temperature Analysis

Devam Patel, ddp107

2023-03-11

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

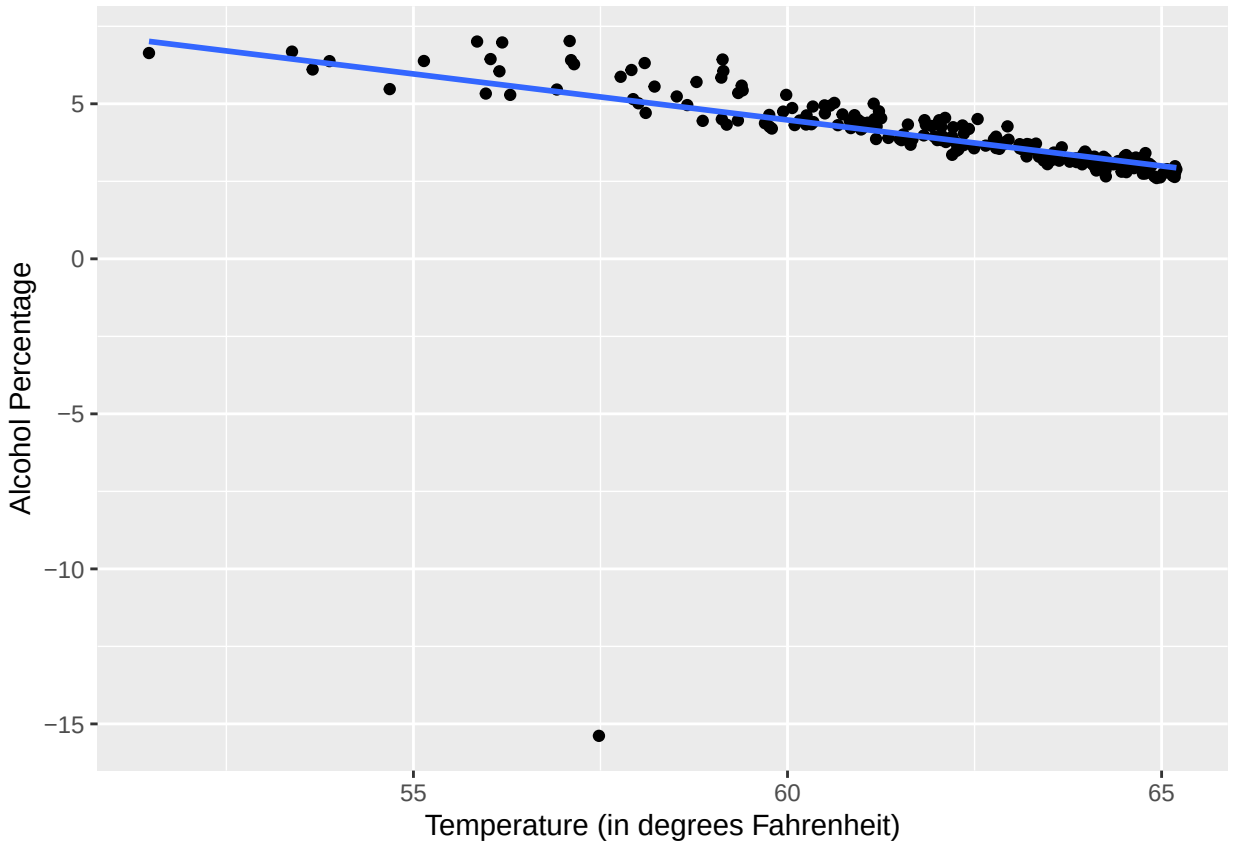
```
library(moderndive)

data = read.delim('modeling_data.txt')
```

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?

```
ggplot(data, aes(x=Temperature, y= Alcohol_Percentage)) + geom_point()+xlab("Temperature (in degrees Fa
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Answer 1. **The plot above shows that there is a linear relationship evident between the two variables with negative correlation as the alcohol percentages seems to decrease as temperature increases and the points roughly fall on a straight line on either side of it. It is also to be noted that an outlier is evident where the alcohol percentage is negative. More assumptions are verified once the outliers is eliminated and the graph is scaled, which is done on question 2.**

Question 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?**

Overall, most of the regression assumptions are satisfied so that simple linear regression can be used. However, there are certain devations needed to be addressed, such as outliers.

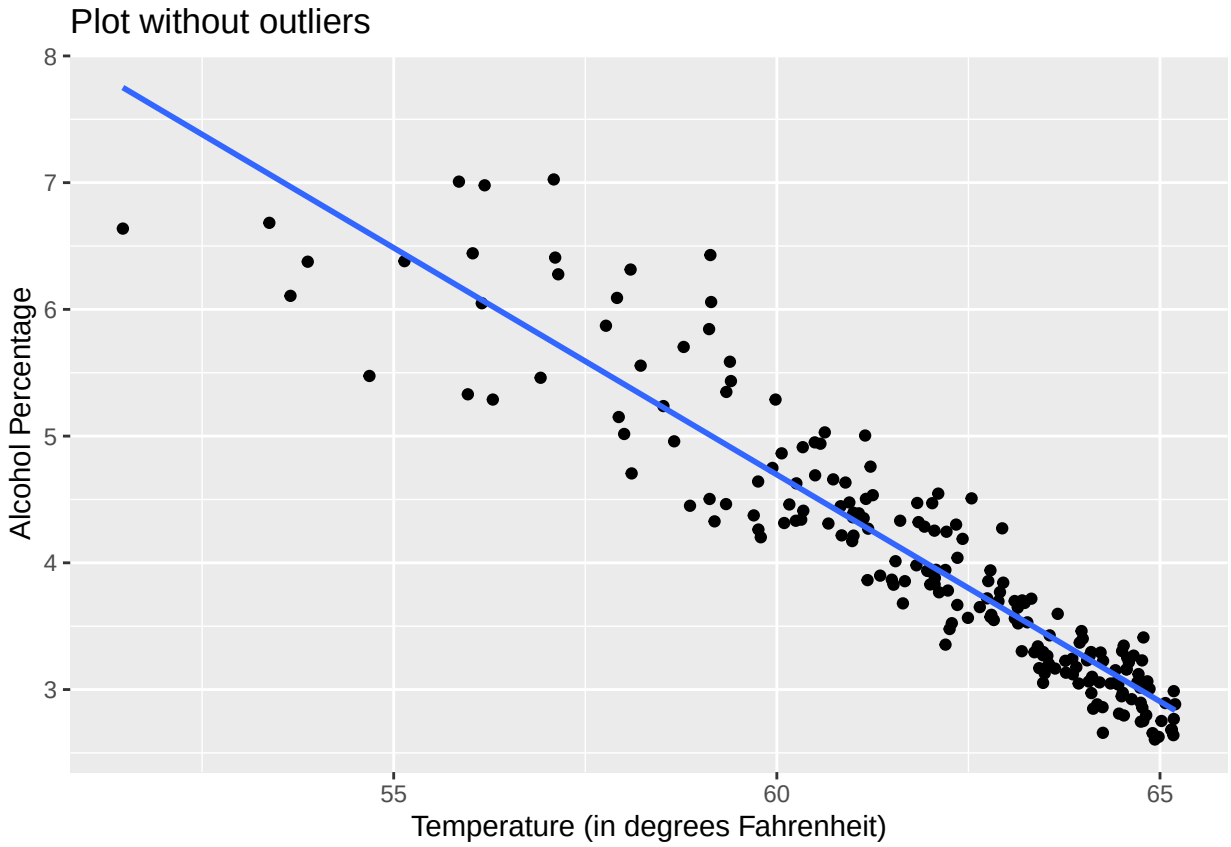
Assumption 1. **Linearity: The linear assumption is appropriate for this dataset as the points roughly fall on a straight line on either side of it.**

Assumption 2. **Error assumptions/homoscedasticity: the variation of the observations around the 'line' is greater for smaller values of the explanatory variable (temperature) and thus the assumption is violated**

```
data_without_outliers = subset(data, Alcohol_Percentage>0)
ggplot(data_without_outliers, aes(x=Temperature, y= Alcohol_Percentage)) + geom_point()+xlab("Temperatu
```

Assumption 3. Outliers: As evident from the graph above, there is an outlier in the dataset with negative alcohol percentage level, which is not fitting the linear assumption and could potentially effect coefficient estimates of the model. To get around this, it is useful to eliminate the outliers from the dataset as they could be harmful in the analysis:

```
## `geom_smooth()` using formula = 'y ~ x'
```



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

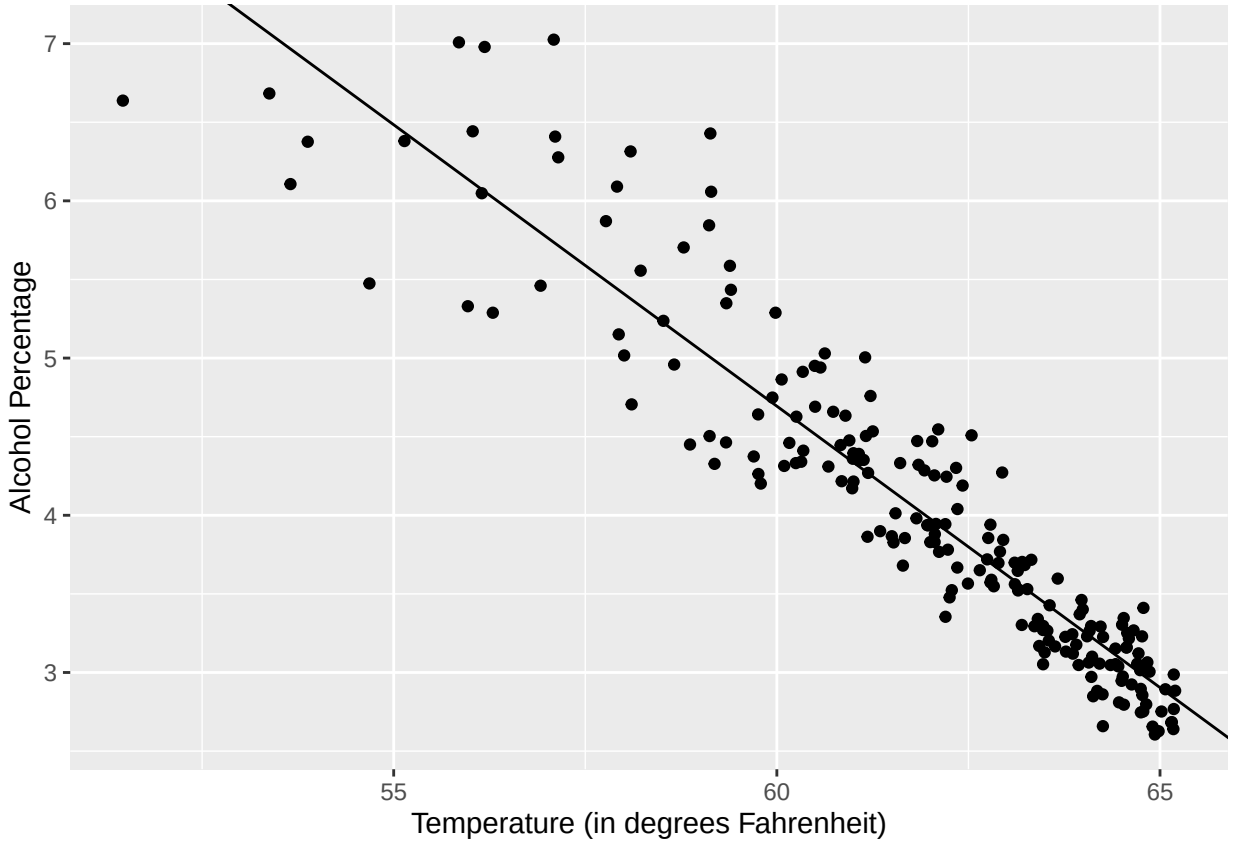
```
model = lm(data=data_without_outliers, Alcohol_Percentage~Temperature)
summary(model)
```

```
##
## Call:
## lm(formula = Alcohol_Percentage ~ Temperature, data = data_without_outliers)
##
## 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
```

Answer 3. After fitting a simple linear model to the dataset, the intercept estimate is 26.174 and the temperature estimate is -0.358, indicating the estimated decrease in alcohol percentage as temperature (in Fahrenheit) increases by 1 degree. R_squared value is 0.869, which indicates that 86.9% of variation in alcohol percentage is explained by the explanatory variable. The estimated line is shown in the plot below.

```
ggplot(data_without_outliers, aes(x=Temperature, y= Alcohol_Percentage)) + geom_point()+xlab("Temperatu
```

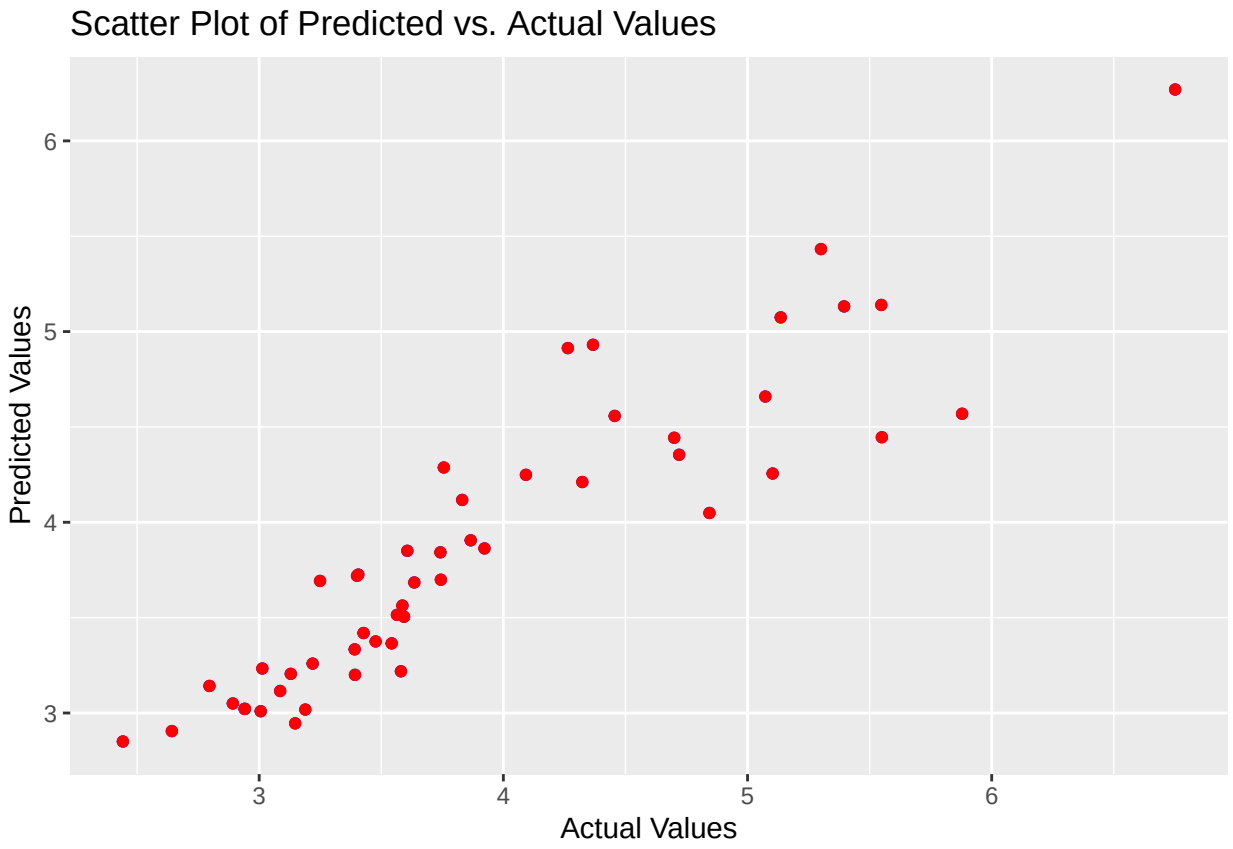


Question 4. Now turn attention to the validation dataset. In order to assess how well the model works, calculate the predicted values using the batch temperatures from the validation dataset and the model from part 3. Plot these predicted values versus the actual values (the actual alcohol content) from the validation dataset in a scatter plot. Additionally, calculate the sample correlation between the predicted values and the actual values.

```
validation = read.delim('validation_data.txt')
predicted_vals = predict(model, newdata = validation)

plot_data = data.frame(actual = validation$Alcohol_Percentage, predicted = predicted_vals)
ggplot(plot_data, aes(x=actual, y=predicted))+geom_point(color = "blue") +
  geom_point(color = "red") +
```

```
  labs(x = "Actual Values", y = "Predicted Values", title = "Scatter Plot of Predicted vs. Actual Values
```



Answer 4. Correlation between actual values and predicted values is 0.917:

```
cor(validation$Alcohol_Percentage, predicted_vals)
```

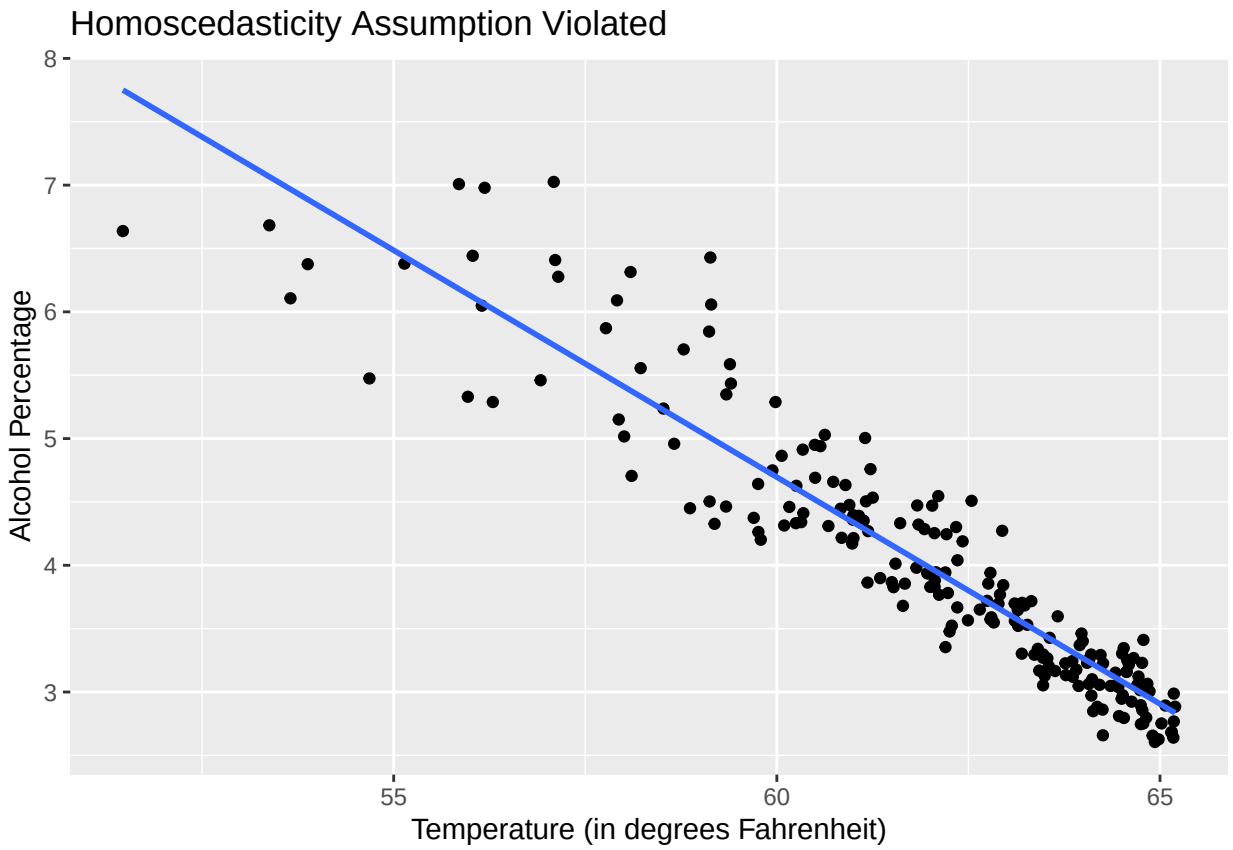
```
## [1] 0.9170591
```

Question 5. As is evident from part 1, for a given fermentation temperature, there is a tremendous amount of variability in the alcohol content of the batch. Consequently, for any given temperature, the brewer would like to get bands that encompass what the final alcohol content of a batch would be with probability 95%. Were the steps taken in part 2 enough to still warrant the use of simple linear regression for this goal, or are there still model deviations that need to be addressed? If so, what are the remaining deviations and how do you address them?

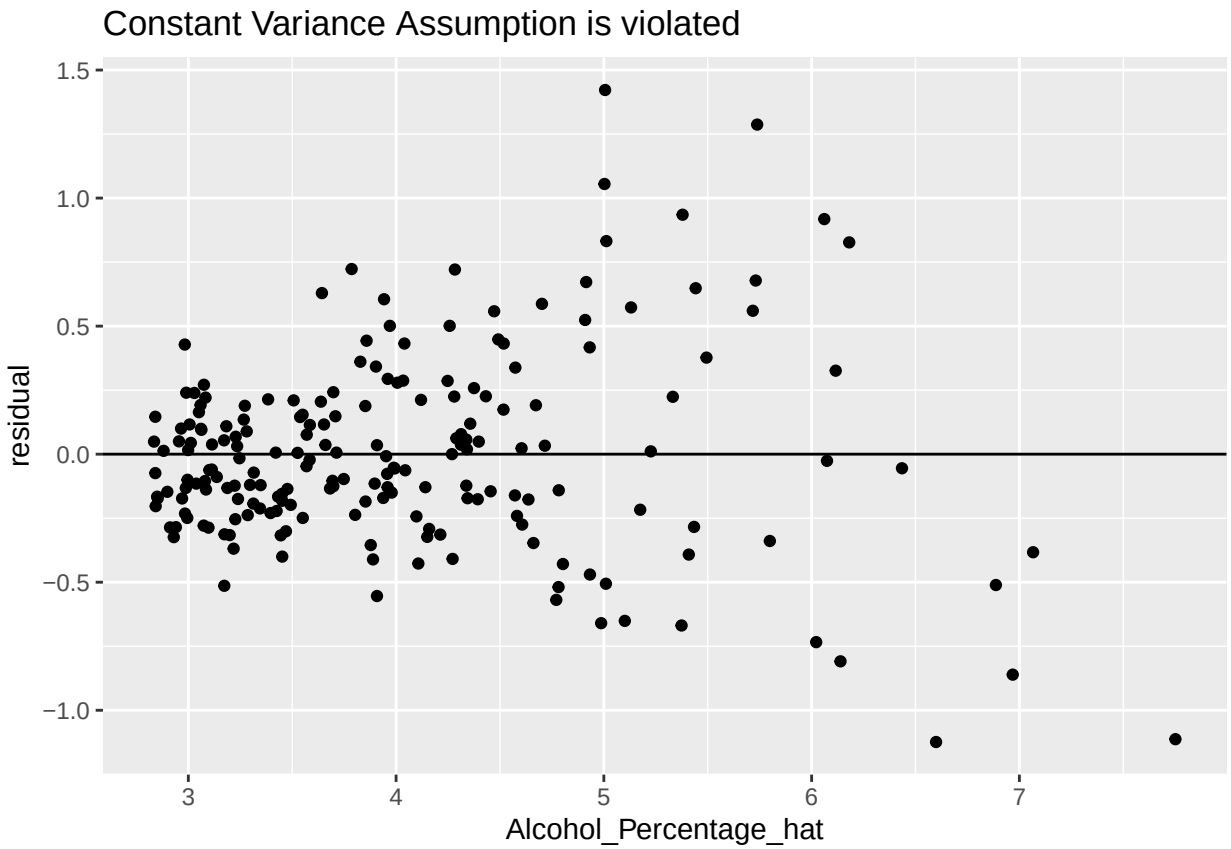
Answer 5. Besides from the outlier issue, there are still model deviations that need to be addressed, as discussed in part 1. The assumption that errors are normally distributed, otherwise known as homoscedasticity, is also violated as there is no constant variance for the errors as shown by the plot below. To address this, a form of transformation of the dataset is needed and it is important to check if other assumptions are met after the transformation. The type of transformation is addressed in the next part. In addition to homoscedasticity, constant variance of residuals and normality assumptions of the linear model are also tested below.

```
ggplot(data_without_outliers, aes(x=Temperature, y= Alcohol_Percentage)) + geom_point()+xlab("Temperatu
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



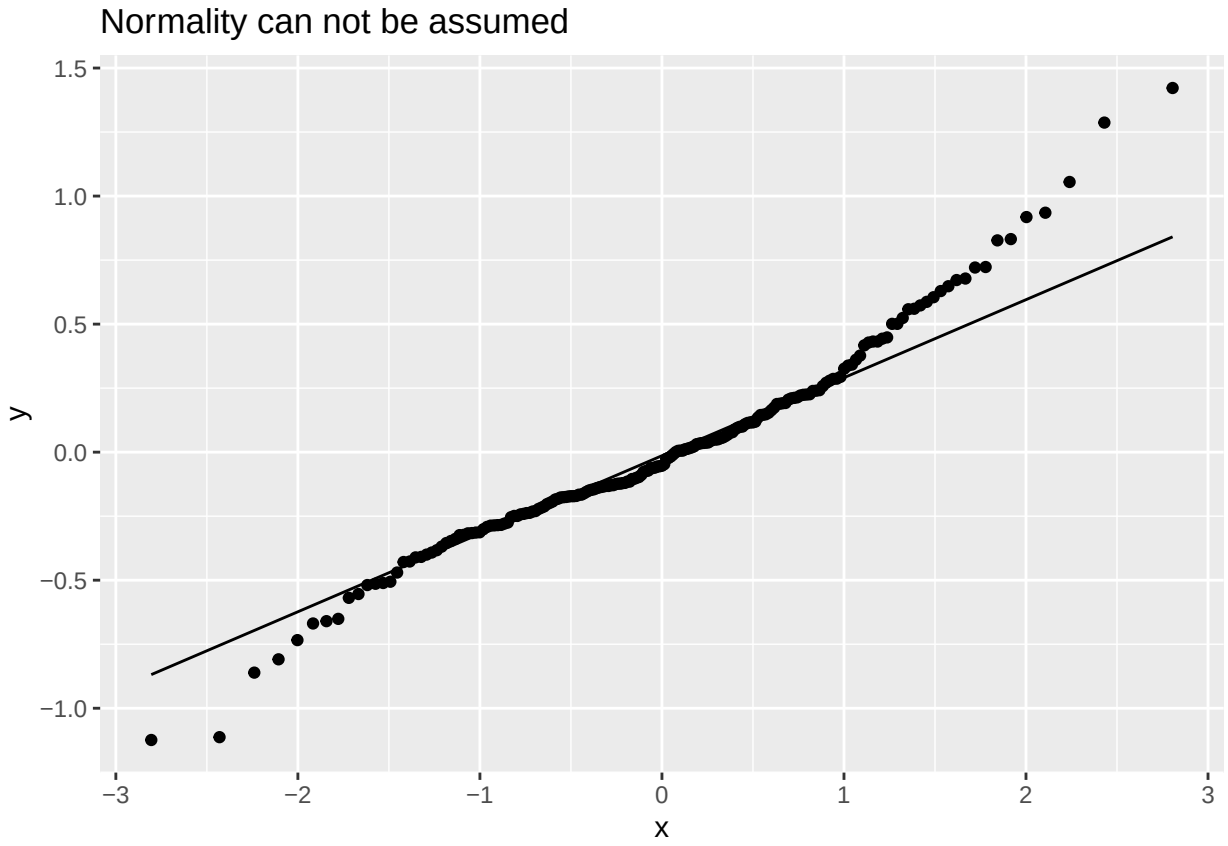

```
points1 = get_regression_points(model)
ggplot(data=points1, aes(x=Alcohol_Percentage_hat, y=residual)) + geom_point() + geom_hline(yintercept=
```



Residuals

As evident, there is no constant variance in the residuals as it follows a pattern across the plot

```
ggplot(data=points1, aes(sample=residual)) + stat_qq() + stat_qq_line()+ggtitle("Normality can not be as
```



qqplot:

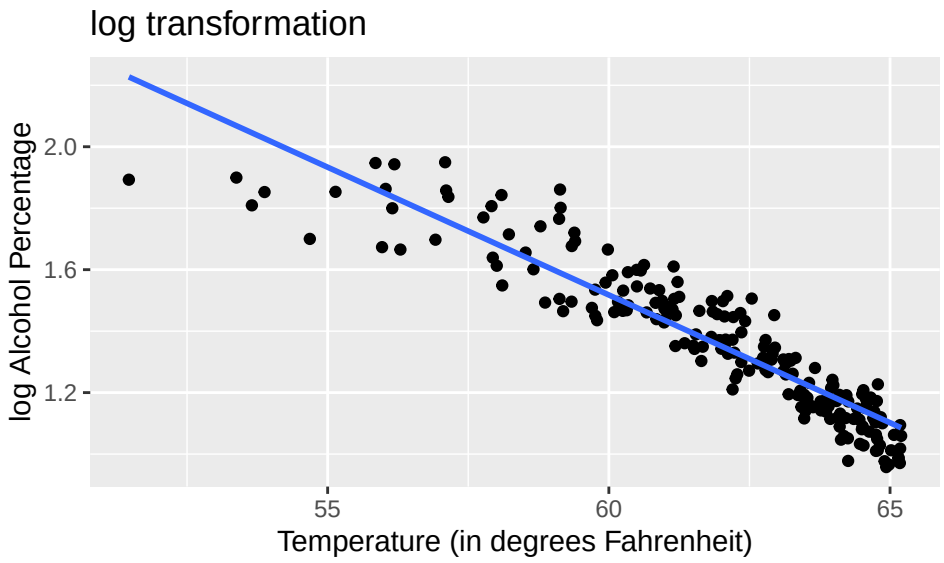
As evident, the normality of the model also seems to be violated as the points diverge away from the line in the qq-plot above.

Question 6.After addressing any additional issues in part 5, obtain a new model for the data. Describe how you arrived at this model. For a given temperature, x, write out the formula for the predicted alcohol content specified for your model. Overlay the estimated regression curve on the plot created in part 1.

Answer 6. As mentioned in #5, transformation is needed mitigate the impact of the violation of homoscedasticity. To acheive this, I first plotted inverse, logarithmic, and square root transformations:

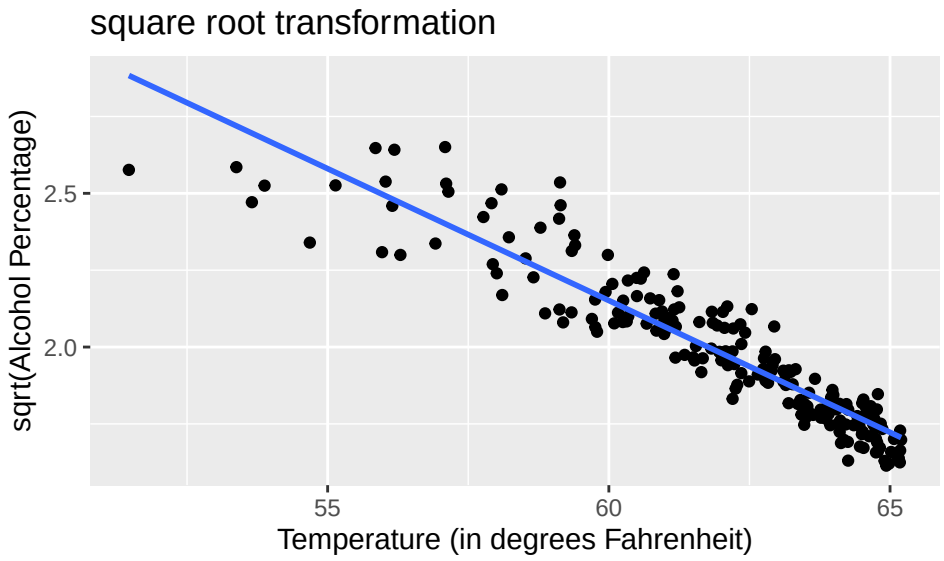
```
ggplot(data_without_outliers, aes(x=Temperature, y= log(Alcohol_Percentage))) + geom_point()+xlab("Temp
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



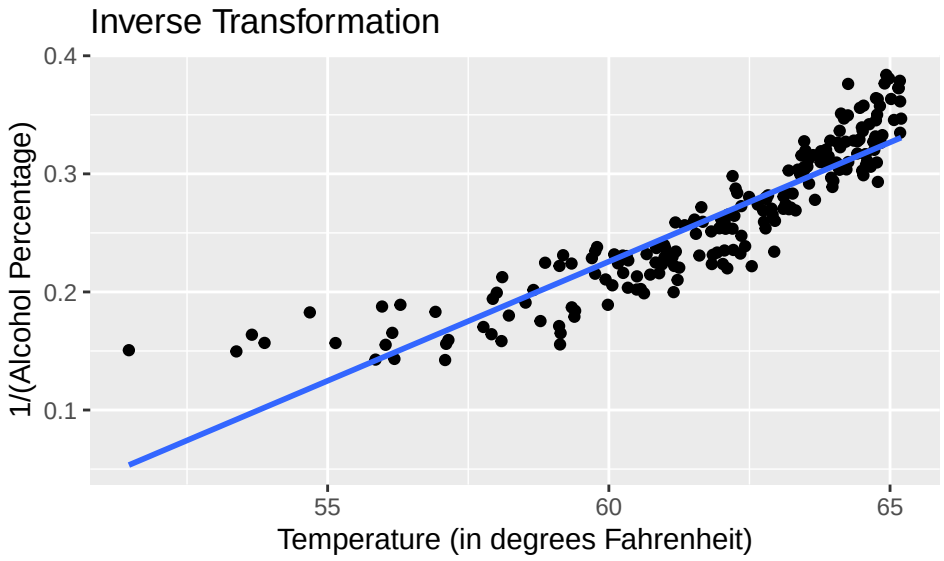

```
ggplot(data_without_outliers, aes(x=Temperature, y= sqrt(Alcohol_Percentage))) + geom_point()+xlab("Temp
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data_without_outliers, aes(x=Temperature, y= 1/(Alcohol_Percentage))) + geom_point()+xlab("Tempe
```

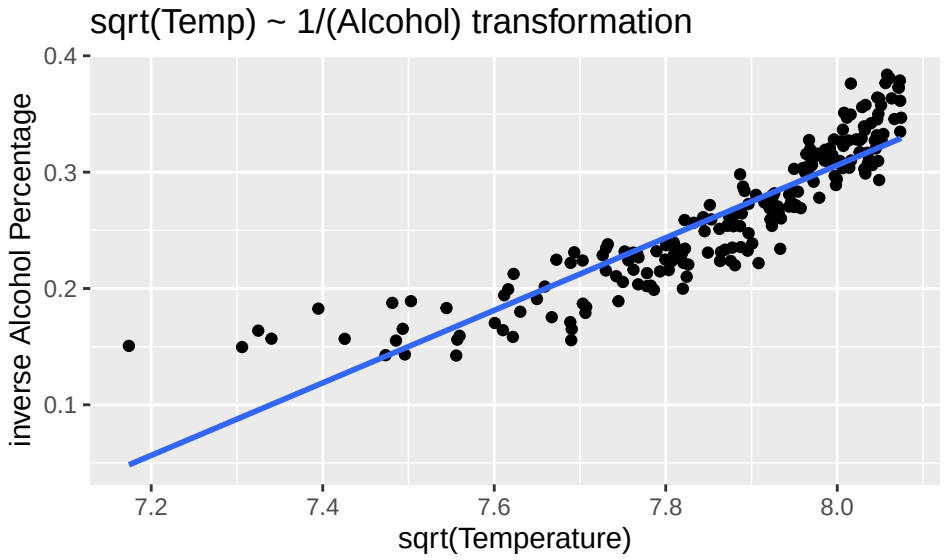
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data_without_outliers, aes(x=sqrt(Temperature), y= 1/(Alcohol_Percentage))) + geom_point()+xlab(
```

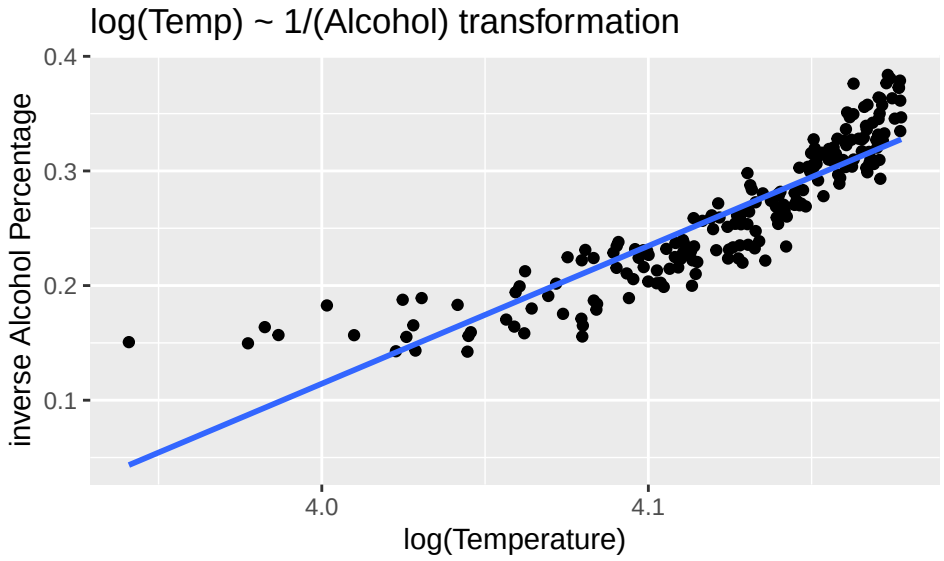
Based on the graphs above, inverse transformation seems to solve the heteroscedasticity issue. However, the explanatory variable must also be transformed in order to make the relationship linear. In order to check this I tried exponential/power, log, and inverse transformations:

```
## `geom_smooth()` using formula = 'y ~ x'
```



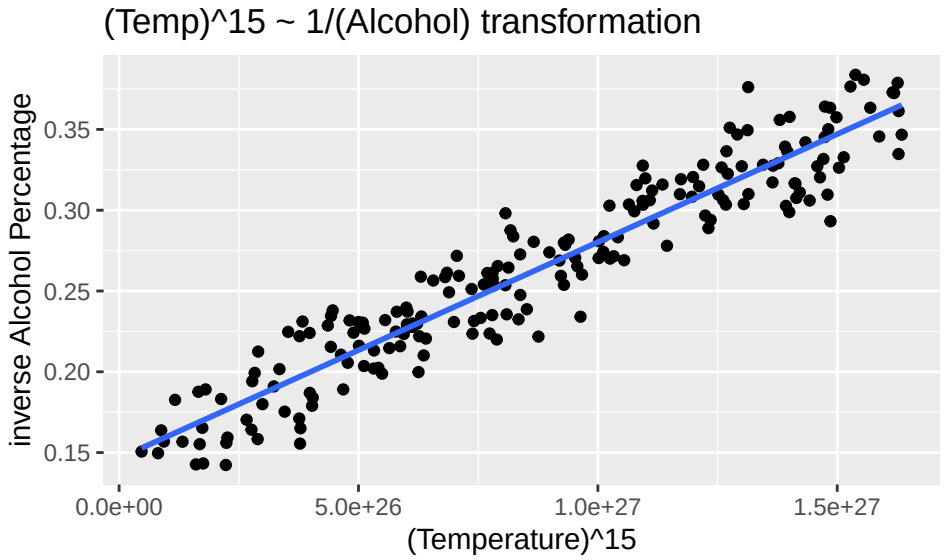

```
ggplot(data_without_outliers, aes(x=log(Temperature), y= 1/(Alcohol_Percentage))) + geom_point()+xlab("
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data_without_outliers, aes(x=(Temperature)^15, y= 1/(Alcohol_Percentage))) + geom_point()+xlab("
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

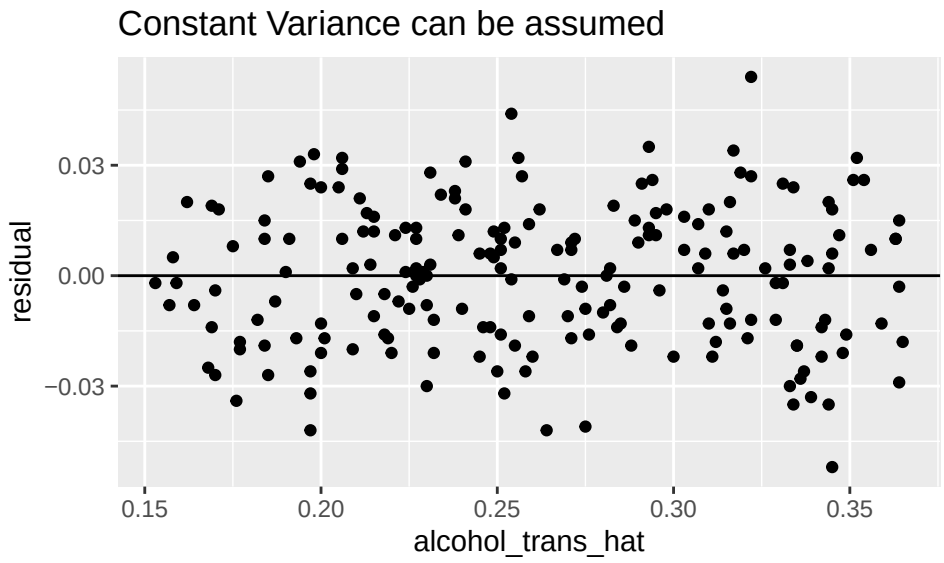


Checking other assumptions:

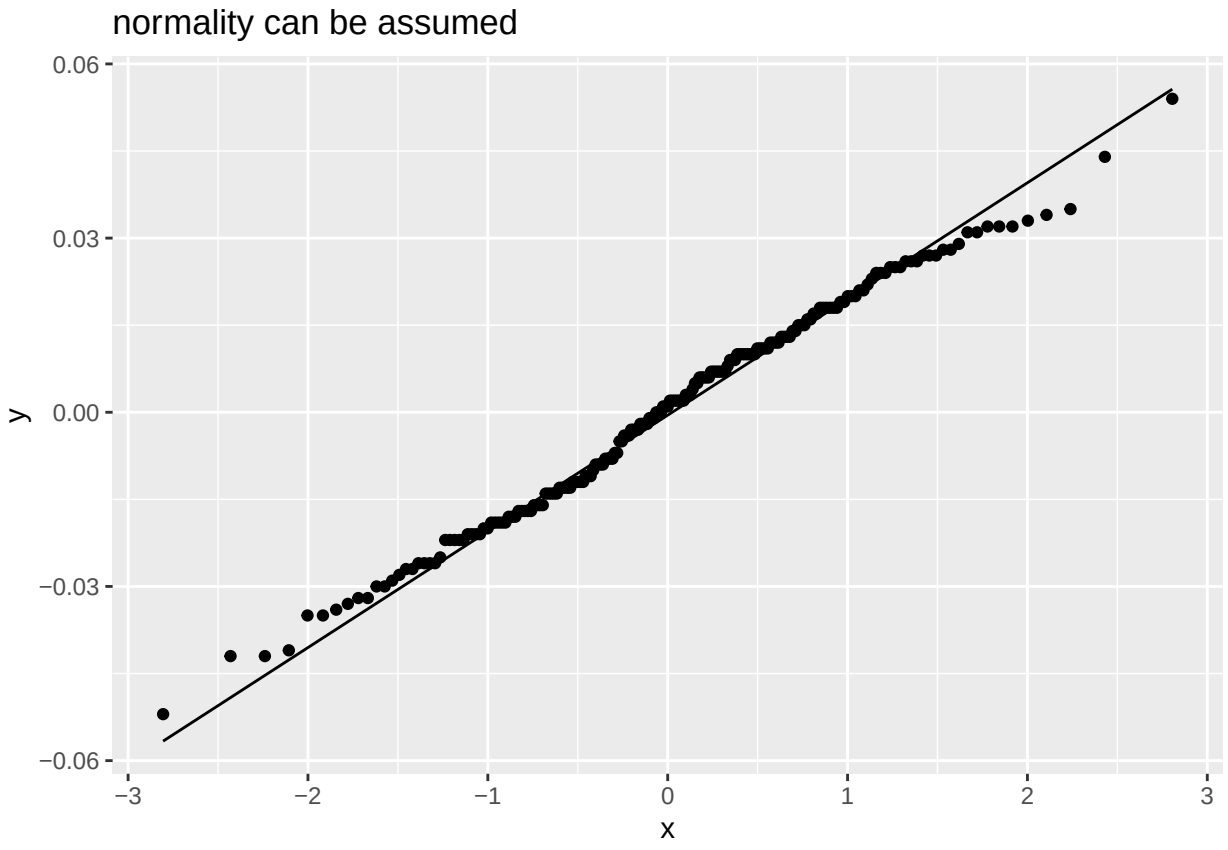
```
trans_data = data.frame(temp_trans=(data_without_outliers$Temperature)^15, alcohol_trans=1/(data_withou
```

```
model2 = lm(data=trans_data, alcohol_trans~temp_trans)
```

```
points2 = get_regression_points(model2)
ggplot(data=points2, aes(x=alcohol_trans_hat, y=residual)) + geom_point() + geom_hline(yintercept=0)+gg
```




```
ggplot(data=points2, aes(sample=residual)) + stat_qq() + stat_qq_line()+ggtitle("normality can be assume
```




```
summary(model2)
```

Compared to the plots from the original model, the new model passes the assumptions by having constant variance, normality, along with homoscedasticity. For the formula based on the new model:

##

```
## Call:
## lm(formula = alcohol_trans ~ temp_trans, data = trans_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05202 -0.01427  0.00141  0.01311  0.05395
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.465e-01  3.047e-03   48.09   <2e-16 ***
## temp_trans  1.337e-28  3.137e-30   42.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01911 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
```

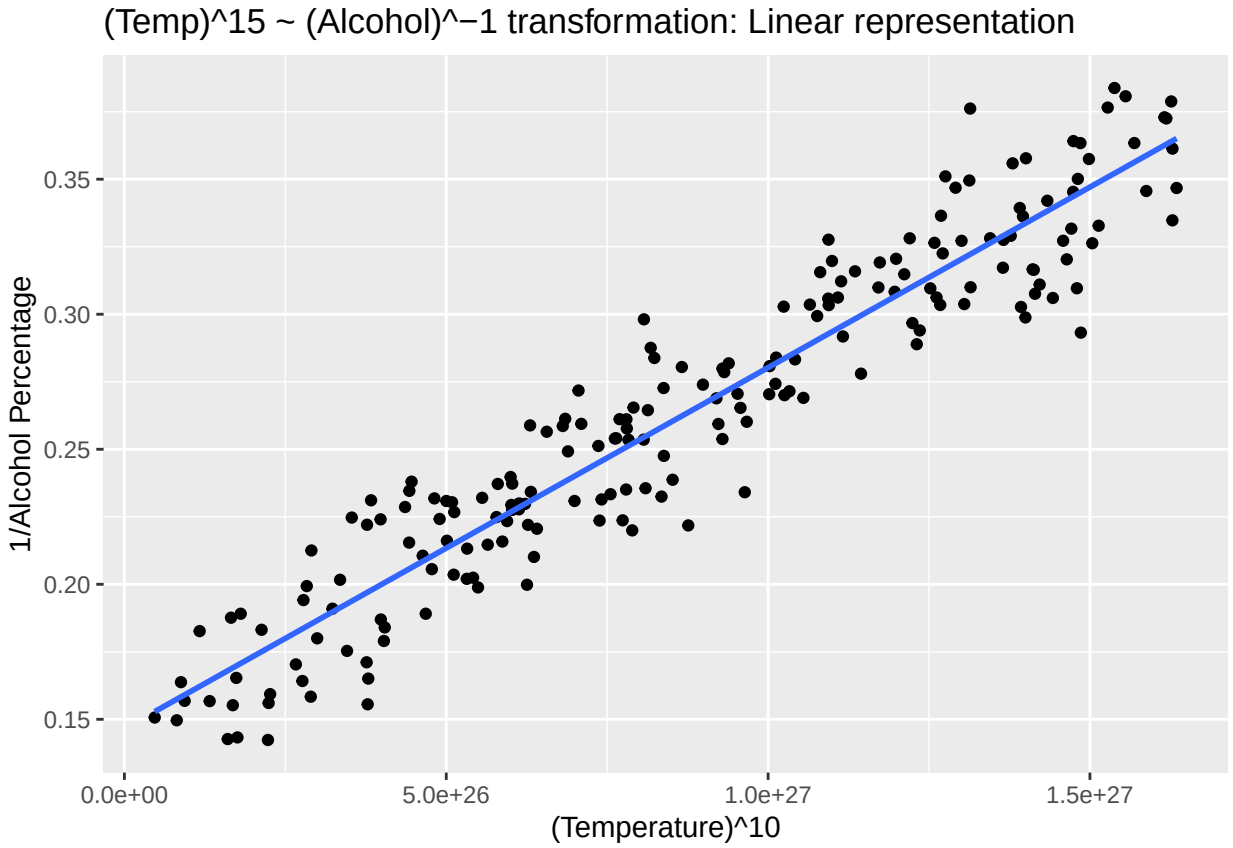
From the summary the formula can be written as A = 1.028e-01 + 1.815e-19*T, where A is log(Alcohol percentage) and T is (temperature)^10.

```
coefs = coef(model2)
```

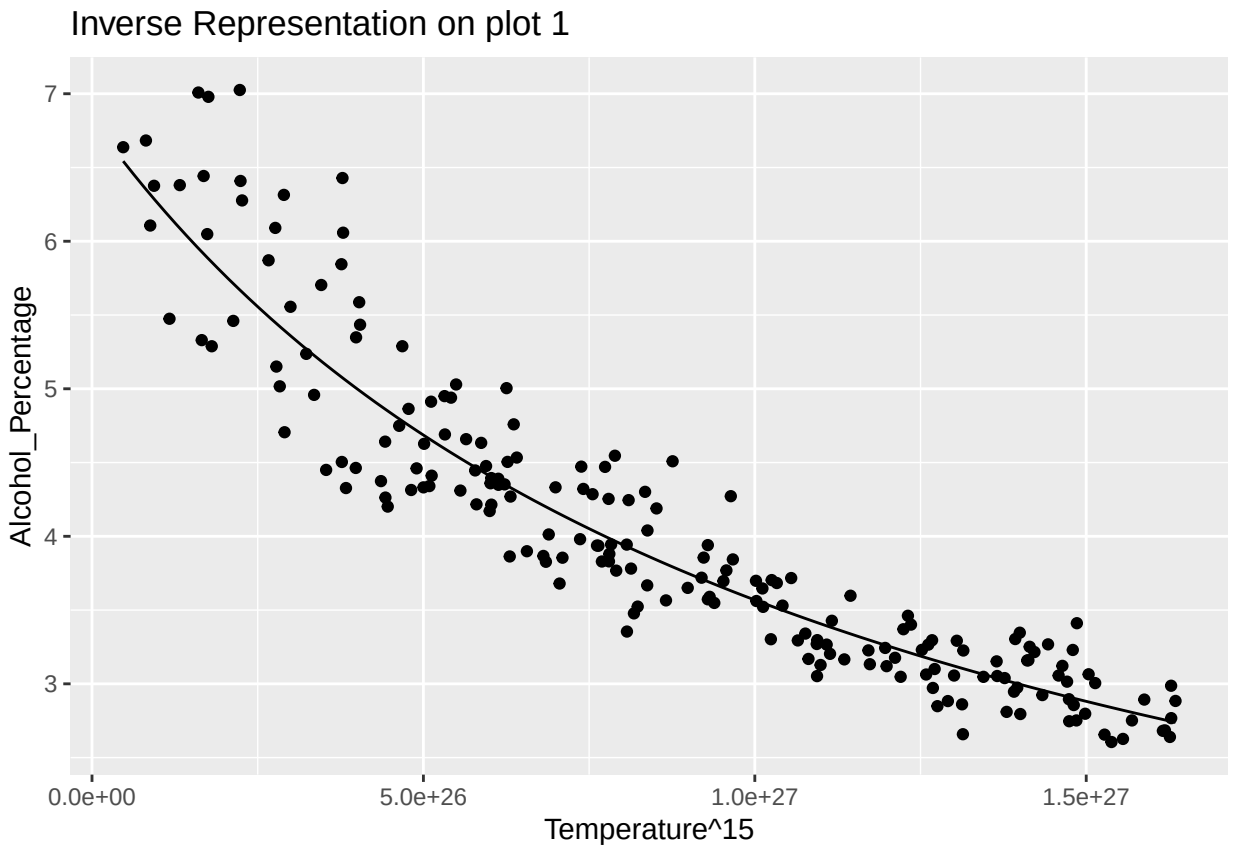
```
ggplot(data_without_outliers, aes(x=(Temperature)^15, y= 1/(Alcohol_Percentage))) + geom_point()+xlab("
```

Or also as Alcohol_percentage = 1/(1.028e-01 + 1.815e-19*T) where T is temperature^10
Final scatter plot looks like the following:

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data_without_outliers, aes(x=Temperature^15, y=Alcohol_Percentage))+geom_point()+stat_function(fu
```

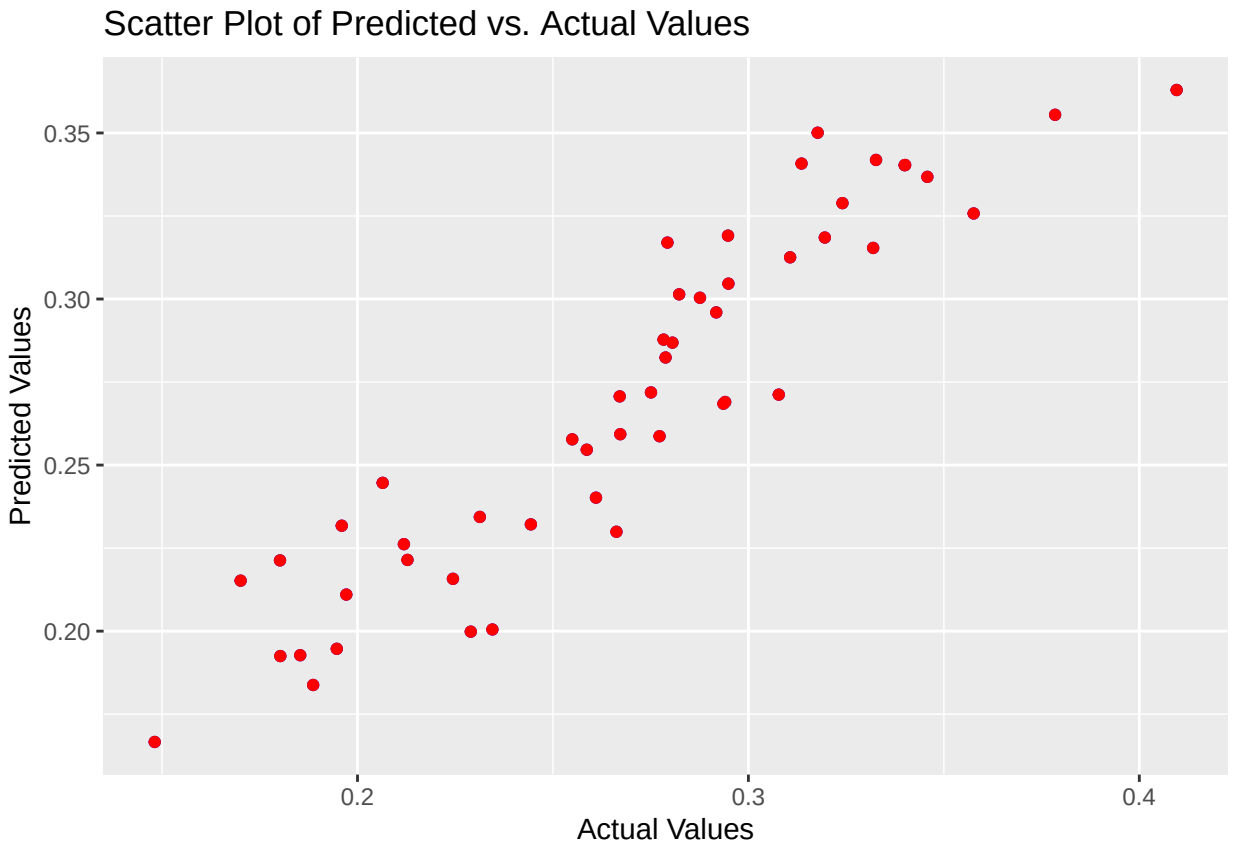


Question 7. Repeat part 4, this time using the model you obtained in part 6.

```
validation_trans = data.frame(temp_trans = (validation$Temperature)^15, alcohol_trans = 1/(validation$Al

predicted_vals2 = predict(model2, newdata = validation_trans)

plot_data2 = data.frame(actual = validation_trans$alcohol_trans, predicted = predicted_vals2)
ggplot(plot_data2, aes(x=actual, y=predicted))+geom_point(color = "blue") +
  geom_point(color = "red") +
  labs(x = "Actual Values", y = "Predicted Values", title = "Scatter Plot of Predicted vs. Actual Values
```



correlation between validation data and predicted values of the new model:

```
cor(validation_trans$alcohol_trans, predicted_vals2)
```

```
## [1] 0.9284652
```

Question 8. As mentioned in part 5, for any given fermentation temperature the brewer would like to obtain bands that encompass what the final alcohol content of a batch would be with probability 95%. Write out a formula for the upper and lower endpoints for these bands as a function of the explanatory variable (possibly transformed). Overlay these bands on the plot created in part 1.

```
#MSE
mean(model2$residuals^2)
```

```
## [1] 0.0003615861
```

```
mu = mean(validation$Temperature^15)
mu
```

```
## [1] 9.236601e+26
```

```
ss = sum((validation$Temperature^15-mu)^2)
ss
```

```
## [1] 7.74296e+54
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = alcohol_trans ~ temp_trans, data = trans_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05202 -0.01427  0.00141  0.01311  0.05395
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.465e-01  3.047e-03   48.09   <2e-16 ***
## temp_trans  1.337e-28  3.137e-30   42.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01911 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
```

based on information above and the formula for standard error given on the slides:

lower bounds: (1.465e-01) + (1.337e-28)*Temp - 1.972*SE) ^ (1/15)

upper bounds: (1.465e-01) + (1.337e-28)*Temp + 1.972*SE) ^ (1/15)

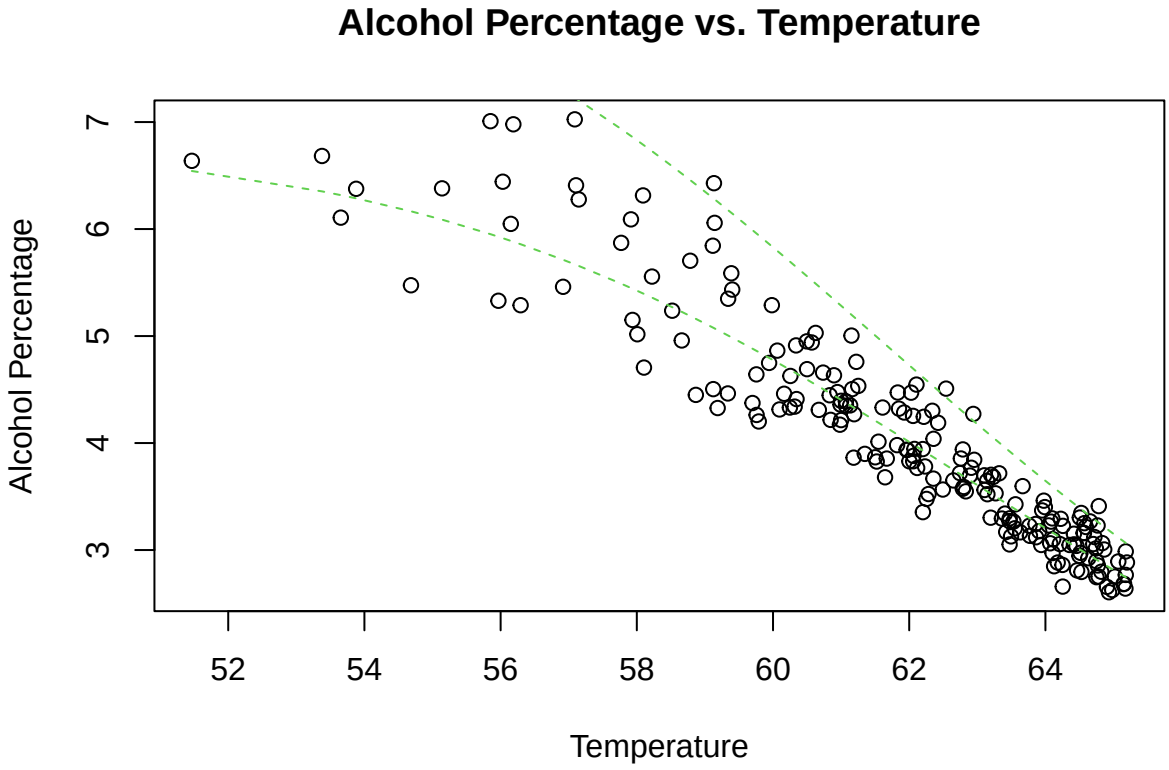
```
PI = predict(model2, interval = "prediction",level = 0.95)
```

```
## Warning in predict.lm(model2, interval = "prediction", level = 0.95): predictions on current data re
```

```
PI <- cbind(data_without_outliers,  1/(PI))
PI = PI[order(PI[,1]),]

plot(data_without_outliers$Temperature, data_without_outliers$Alcohol_Percentage, xlab ="Temperature",

points(PI[,1],PI[,3], type="l", lty=2, col = 3)
points(PI[,1],PI[,4], type="l", lty=2, col = 3)
```


Alcohol Percentage vs. Temperature
Alcohol Percentage
3
4
5
6
7
52
54
56
58
60
62
64
Temperature