Math 189 Project 4

Team Members
Colleen Chan A11898607 - Economics & Probability/Statistics / Third Year Undergraduate
Youli Wang A91085064 - Mathematics & Economics / Third Year Undergraduate
Cong (George) Sun A91014467 - Applied Math / Third Year Undergraduate
Xin (Jason) Shi A98052740 - Management Science & Political Science / Fourth Year Undergraduate
Woong Hyun Suh A98101910 - Electrical Engineering & Society / Fourth Year Undergraduate
Youngsun Lee A13202859 - Applied Math / Third Year Undergraduate

INTRODUCTION

The Sierra Nevada mountains is the main source of water for northern California. To help monitor the water supply, the Forest Service of the United States Department of Agriculture (USDA) operates a gamma transmission snow gauge in the Central Sierra Nevada near Soda Springs, CA. The gauge is used to determine a death profile of snow density. The snow gauge does not disturb the snow in the measurement process, which means the same snow-pack can be measured over and over again. With replicated measurements on the same volume of snow, researchers can study snow-pack settlement over the course of the winter season and the dynamics of rain on snow. When rain falls on snow, the snow absorbs the water up to a certain point, after which flooding occurs. The denser the snow pack, the less water it can absorb. Analysis of the snow-pack profile may help with monitoring the water supply and flood management (Bradic 1).

The gauge does not directly measure snow density. The density reading is converted from a measurement of gamma ray emissions. Due to instrument wear and radioactive source decay, there may be changes over the seasons in the functions used to cover the measured values into density readings. To adjust the conversion method, a calibration run is made each year at the beginning of the winter season (Bradic 1). In this paper, we will develop a procedure to calibrate the snow gauge.

THE DATA

The data comes from a calibration run of the USDA Forest Service's snow gauge located in the Central Sierra Nevada mountain range near Soda Springs. The run consists of placing polyethylene blocks of known densities between the two poles of the

snow gauge and taking readings on the blocks. The polyethylene blocks are used to simulate snow. For each polyethylene block, 30 measurements are taken. Only the middle 10 are reported. The measurements reported are amplified version of the gamma photon count made by the detector. We call the gauge measurement the “gain”. The data available here consists of 10 measurements for each of 9 densities in grams per cubic centimeter of polyethylene (Bradic 1). Our data is longitudinal data.

Some challenges with the data include the decline of operational networks in northern regions, which means we may have a lack of data. There are few stations in the mountain regions. Additionally, the data quality and compatibility vary across national boundaries. There are large biases in in gauge measurements of solid precipitation and incompatibility of precipitation data due to difference in instruments and methods of data processing. There are also difficulties in determining precipitation changes in the arctic regions. The quality of the precipitation data, which includes satellite and reanalysis products and fused products at high latitudes, is also questionable. Despite this, we perform analysis on the available data.

BACKGROUND

The snow gauge can be a complex and expensive instrument. It is not feasible to establish a broad network of gauges in the watershed area in order to monitor the water supply. Instead the gauge is primarily used as a research tool. The snow gauge has helped to study snow-pack settlings, snow-melt runoff, avalanches, and rain-on-snow dynamics. Gauges exist in several states including Idaho, Colorado, Alaska and countries including Russia, Mongolia, China, Japan. The gauge in California is located in the center of a forest opening that is roughly 62 meters in diameter. The laboratory site is at 2099 meters elevation and is subject to all major high altitude storms which regularly deposit 5-20 cm of wet snow. The snow-pack reaches an average depth of 4m each winter (Bradic 1).

The lift mechanism of the gauge at the top of the poles raises and lowers the source and detector together. The radioactive source emits gamma photons, also called gamma rays at 662 kilo-electron-volts (keV), in all directions. The detector contains a scintillation crystal which counts those photons eating through the 70-cm gap from the source to the detector crystal. The pulses generated by the photons that reach the deter crystal are transmitted by a cable to a preamplifier and are then further amplified and transmitted via a buried coaxial cable to the lab. There, the signal is stabilized, corrected for the temperature drift, and converted to a measurement we have termed the ”gain”. It should be directly proportional to the emission rate. The snow-pack density typically ranges between 0.1 and 0.6 g/cm3 (Bradic 1):

The gamma rays that are emitted from the radioactive source are sent out in all directions. Those that are sent in the direction of the detector may be scattered or

absorbed by the polyethylene molecules between the source and the detector. With denser polyethelene, fewer gamma rays will reach the detector. There are complex physical models for the relationship between the polyethylene density and the detector readings. A simplified version of the model that may be workable for the calibration problem of interest that is described here. A gamma ray on route to the detector passes a number of polyethylene molecules, which depends on the density of the polyethylene. A molecule may either absorb the gamma photon, bounce it out of the path to the detector, or allow it to pass. If each molecule acts independently, then the chance that a gamma ray successfully arrives at the detector is p^m where p is the chance a single molecule will neither absorb nor bounce the gamma ray, and m is the number of molecules in a straight line path from the source to the detector. This probability can be re-expressed as
$$e^{m~log~p} = e^{bx}$$ where x the density of the polyethylene and m is the number of molecules.

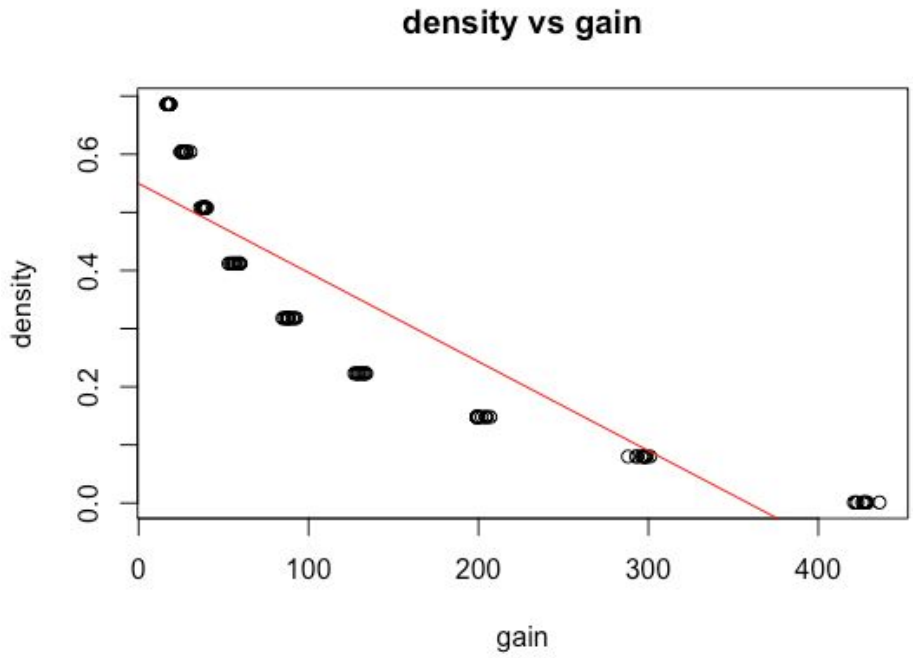
THEORY (on LaTeX file)
Talk about scatterplots, residuals, three conditions of least squares line, residuals, measures of a “best fit” line,

http://stats.stackexchange.com/questions/29731/regression-when-the-ols-residuals-are-not-normally-distributed

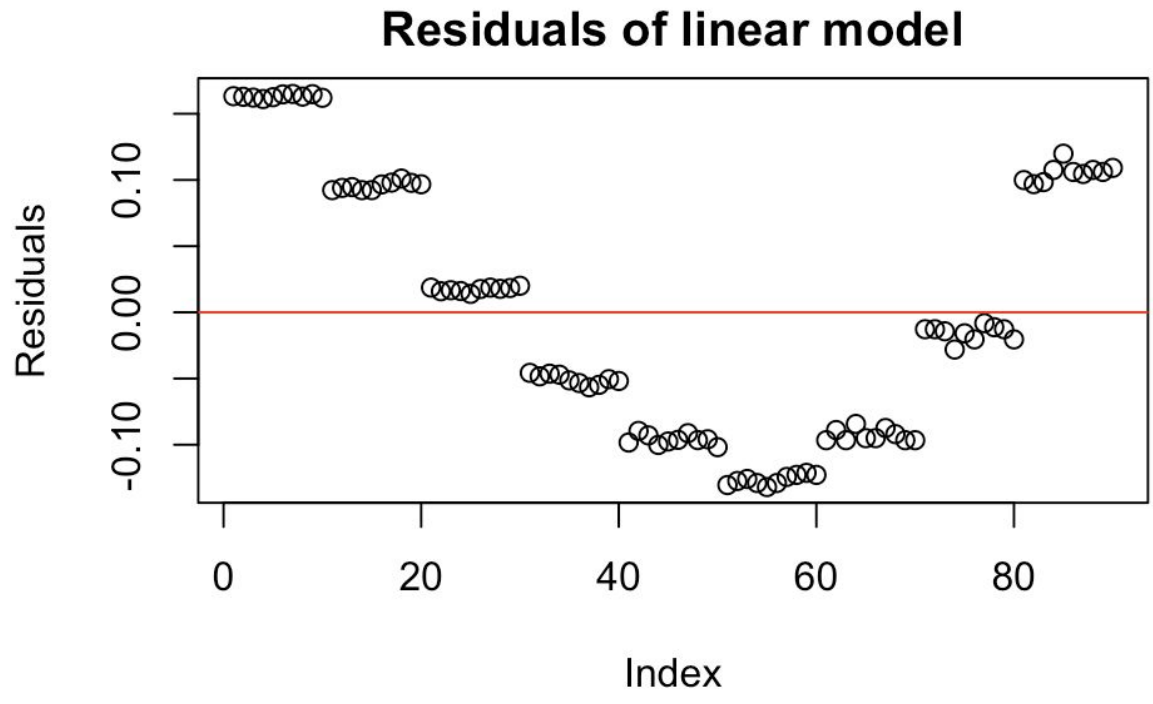
ANALYSIS

Q1 [Fitting] Use the data to fit the gain, or a transformation of gain, to density. Try sketching the least squares line on a scatter plot.

We first want to investigate if there exists a linear relationship between “gain” and density of the polyethylene. We consider a least squares fit of the data. We use a scatter plot to plot the data and superimpose the least squares line.

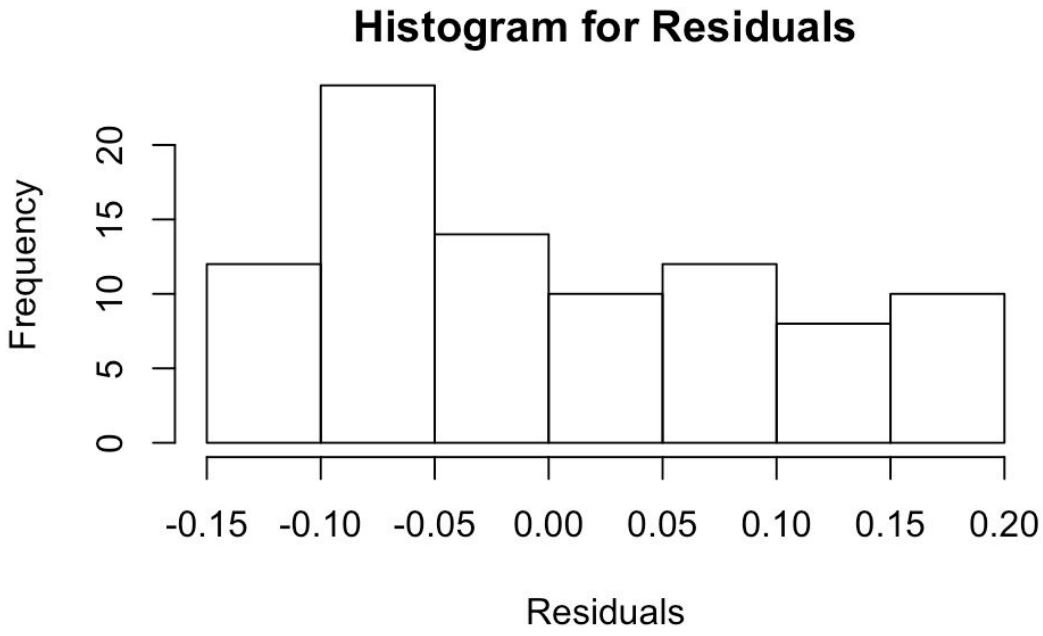


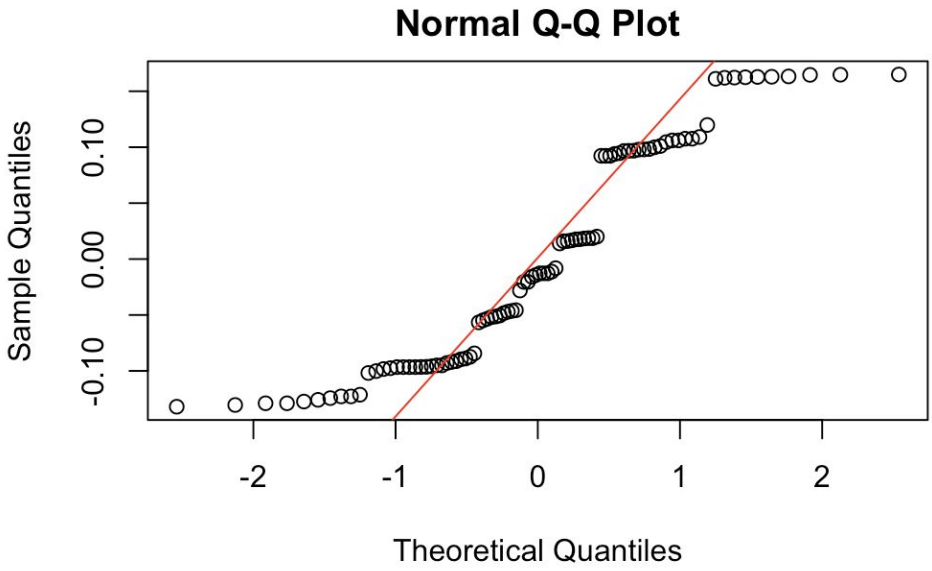
It can be shown that the line fits poorly with the data. To further demonstrate that the model does not fit the data, we look at the residual plot.



The residuals has an obvious quadratic pattern, which indicates that our linear model does not fit our data well. If the least squares line was indeed a good fit, the

residuals should be approximately normally distributed. To check for normality, we construct a histogram of the residuals and a quantile-quantile plot.



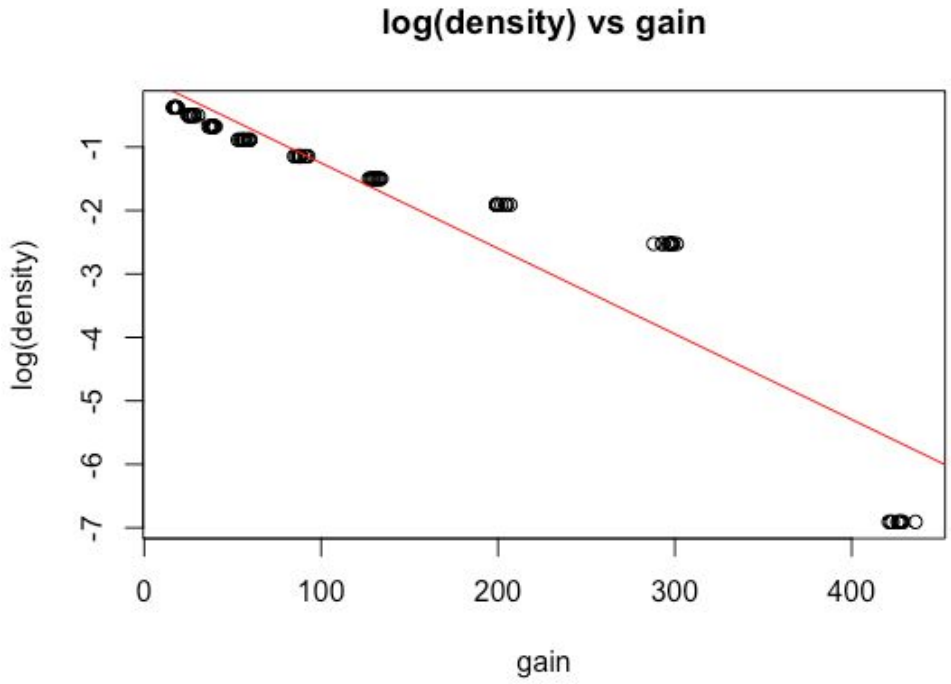


The distribution of residuals are clearly not normally distributed. To confirm this, we performed Shapiro-Wilk test to test for normality, and obtain a p-value of 5.303e-06, which is statistically significant. Thus, we reject the null hypothesis that the distribution of residuals is normal.

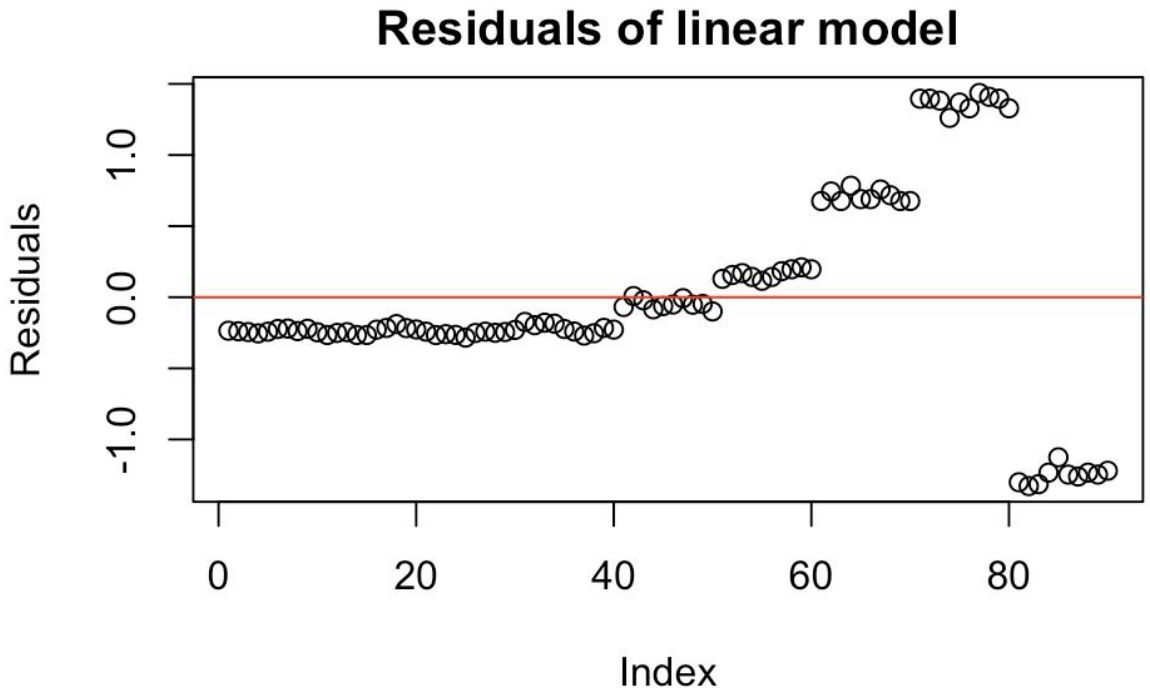
Since the original linear model does not work, we perform log transformations on the variables. We begin by taking the log of density and apply the same method as previous part to generate a linear model.

1. Log density

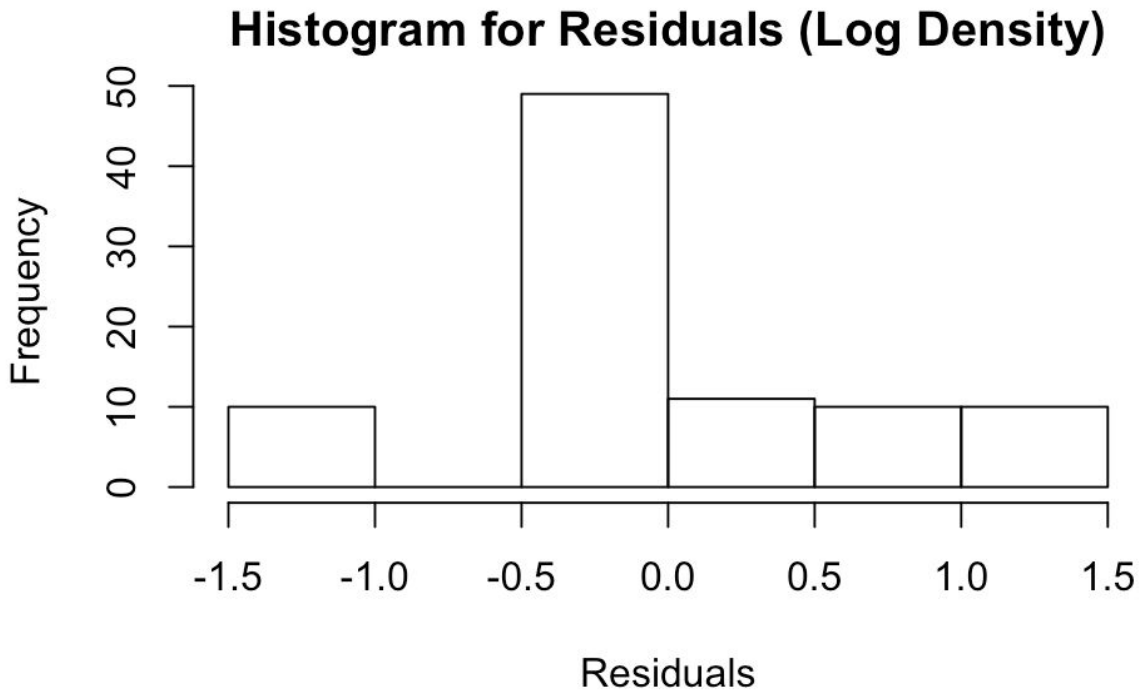
Below is the plot of the data and the least squares line:

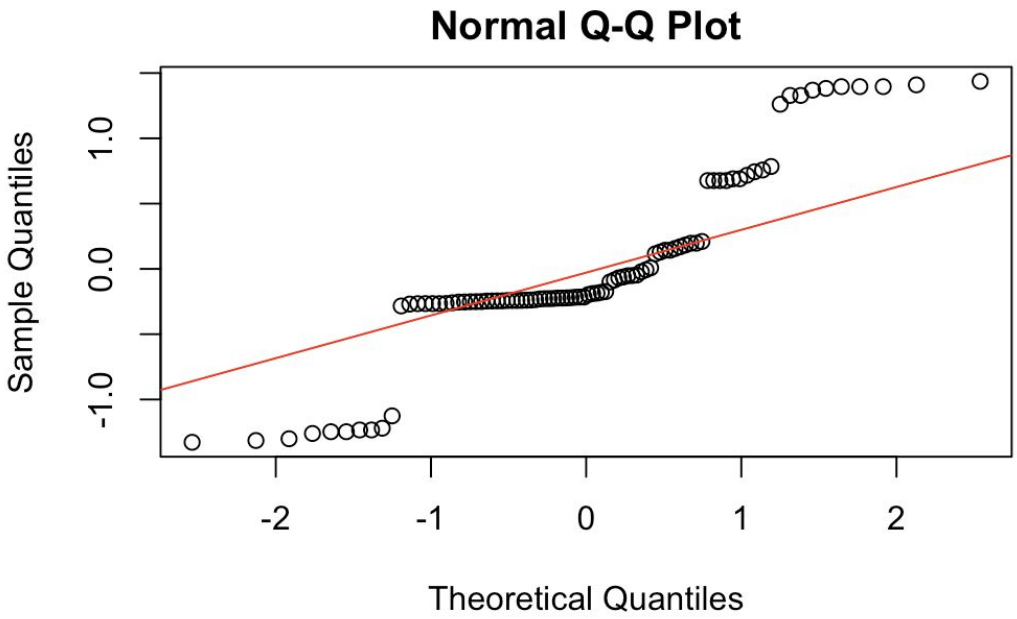


We can see that the line still fits the data poorly. Below is the residual plot.



The residuals shows an exponential pattern, which means our model does not fit data well. To check normality of the residuals, we construct a histogram and a quantile-quantile plot:

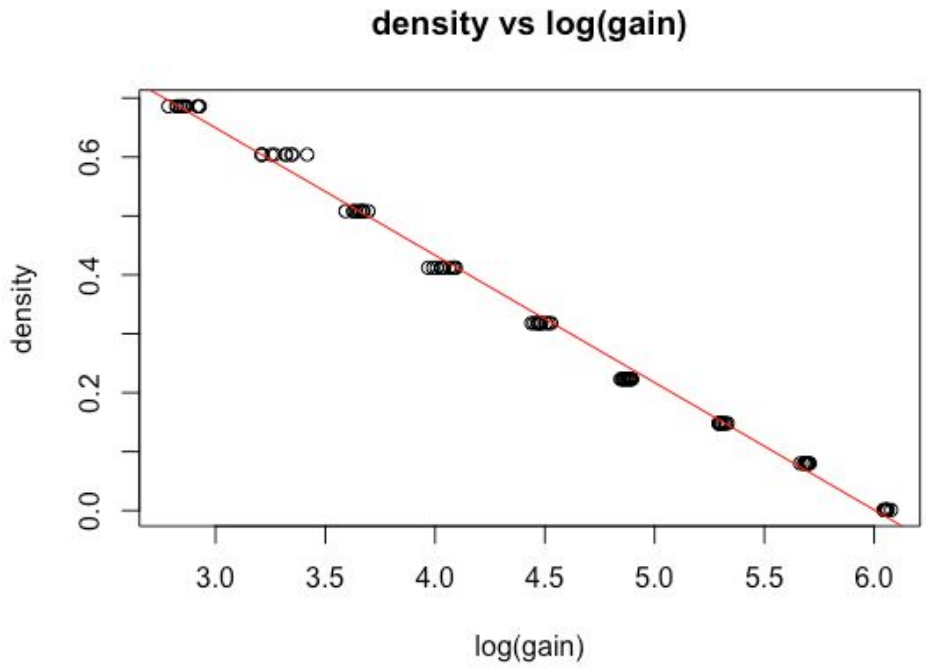




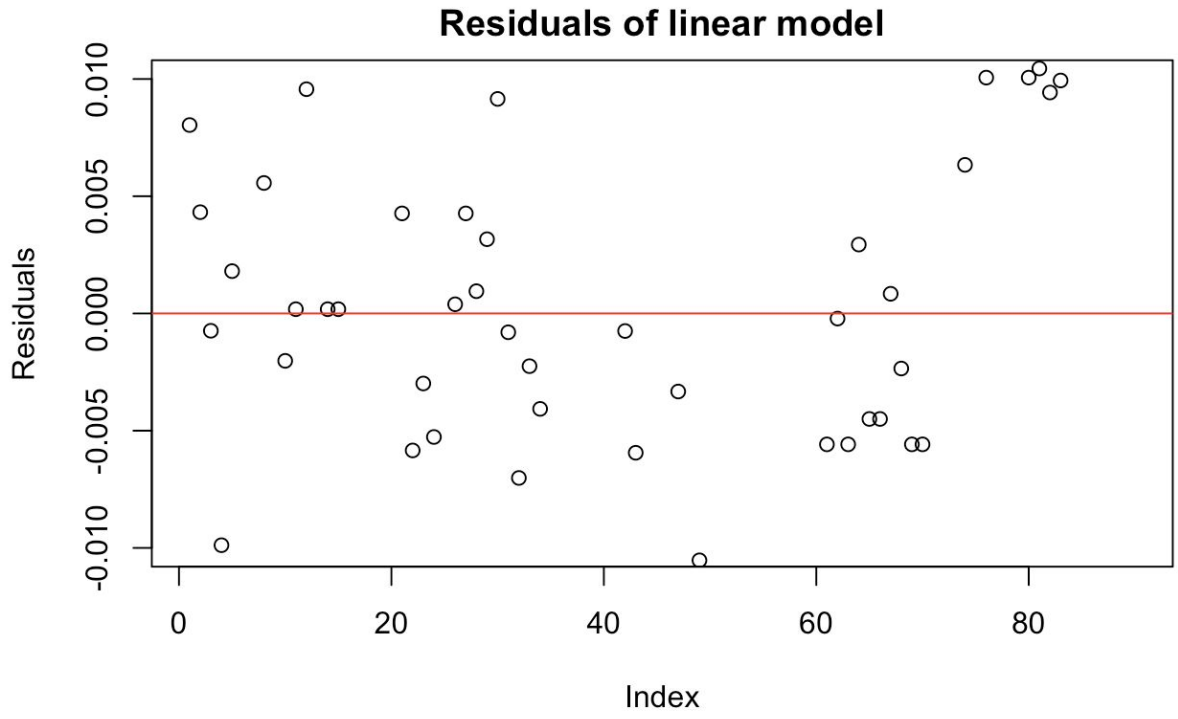
We see that the residuals are not normally distributed. Performing the Shapiro-Wilk test yields a p-value of 3.861e-14, which is very statistically significant. Hence, we reject the null hypothesis that the residuals are normally distributed.

2. Log gain

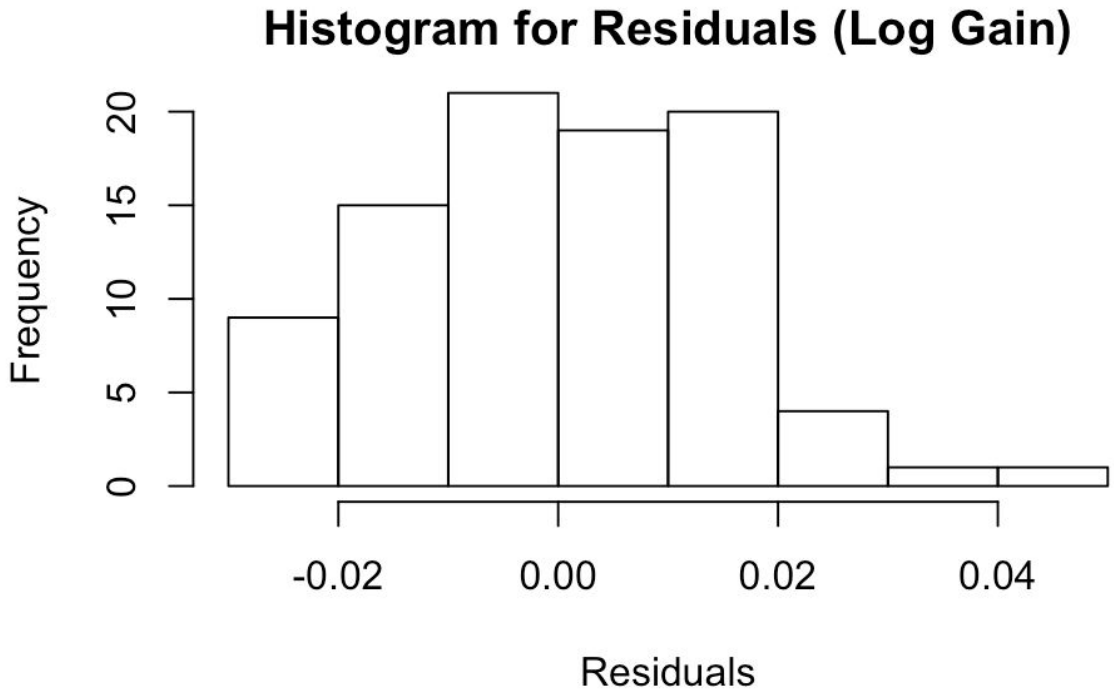
We begin by displaying the scatter plot of log gain vs. density.

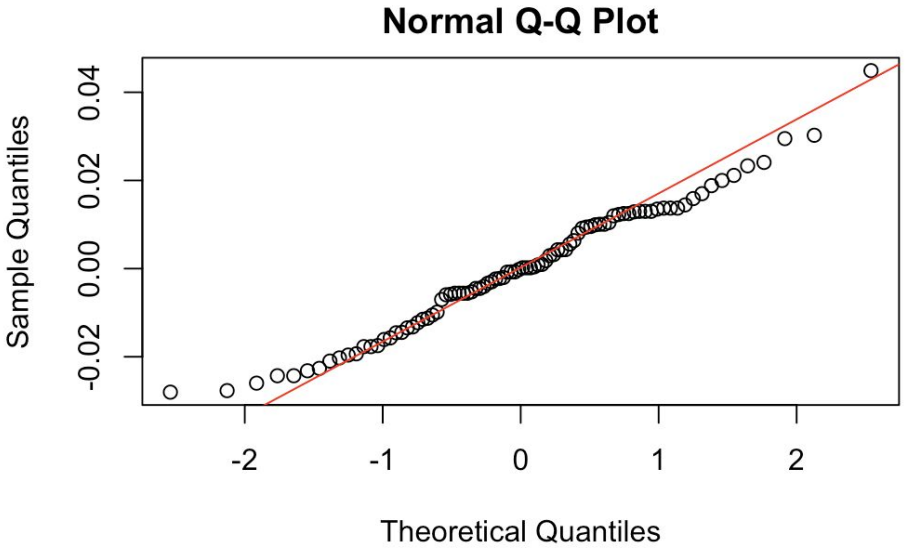


The least squares line seems to fit our data well.



The residuals show a fairly random pattern, but we still need to confirm if it is normal. We construct a histogram of the residuals and a quantile-quantile plot.





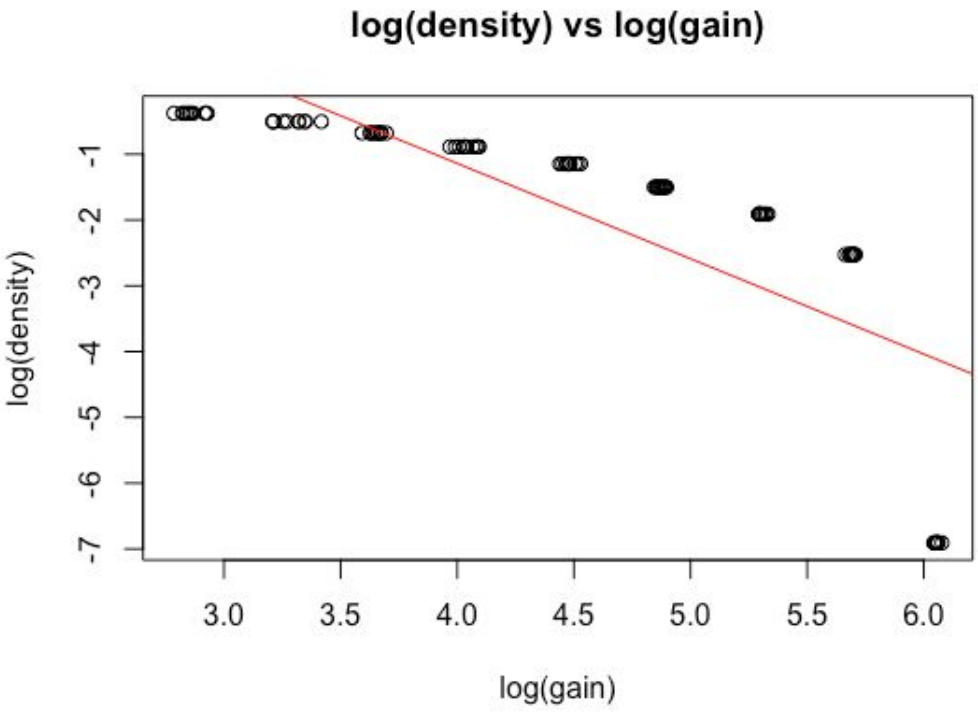
The quantile-quantile plot shows that the line fits the data well, meaning the residuals are fairly normally distributed. The Shapiro-Wilk test yields a p-value of 0.2829, which is greater than 0.05, so we fail to reject our null hypothesis that the residuals are normally distributed. Our residuals follow normal distribution.

```
Call:
lm(formula = density ~ gain, data = dataloggain)

Residuals:
      Min        1Q    Median        3Q       Max
-0.028031 -0.011079 -0.000018  0.011595  0.044911

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.298013   0.006857   189.3   <2e-16 ***
gain        -0.216203   0.001494  -144.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01471 on 88 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9958
F-statistic: 2.096e+04 on 1 and 88 DF,  p-value: < 2.2e-16
```

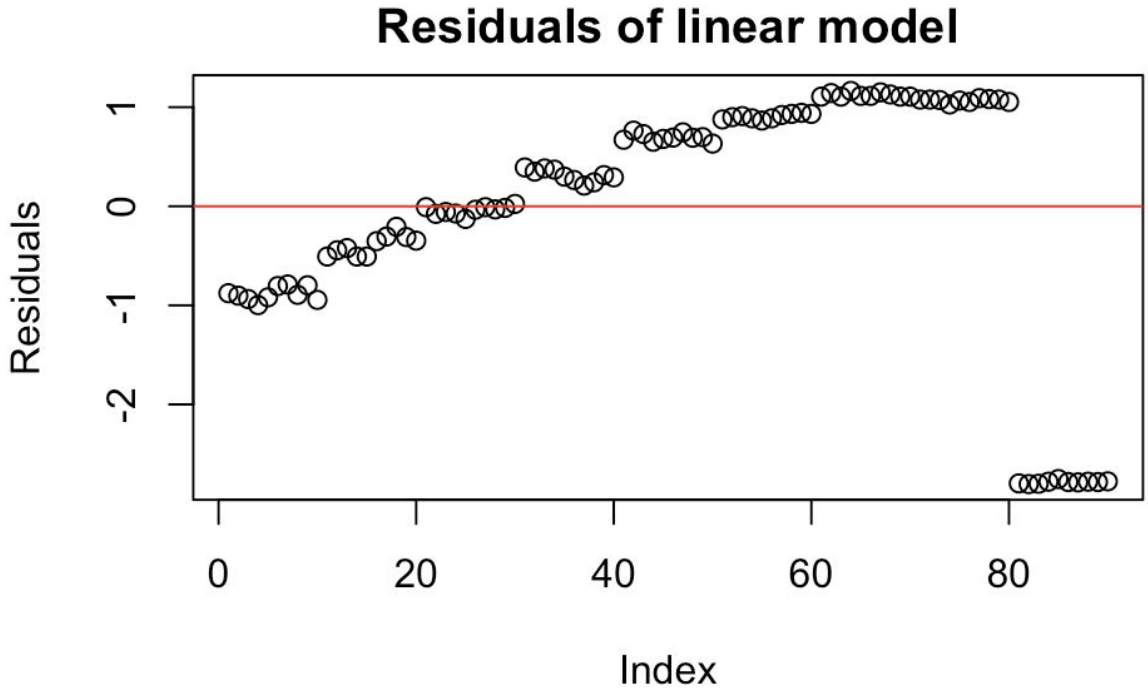
We find that the residual distribution is approximately normal and the p-value is 2.2e-16 with degree of freedom of 88. Thus, we can reject the null hypothesis that there is no relationship between density and log(gain), which means there is a relationship between density and log(gain). The R squared value is 0.9958 which 99.58% of the variation in density can be explained by our model.

3. Log gain and log density

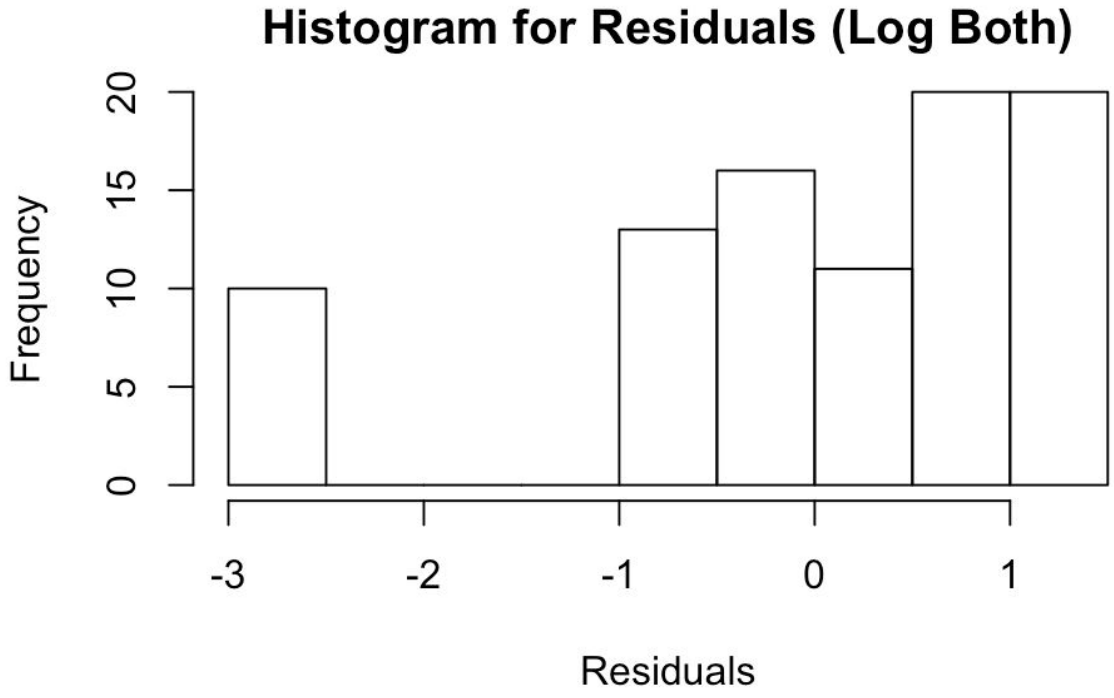
Finally, we construct a scatter plot and a least squares line for log gain vs. log density.

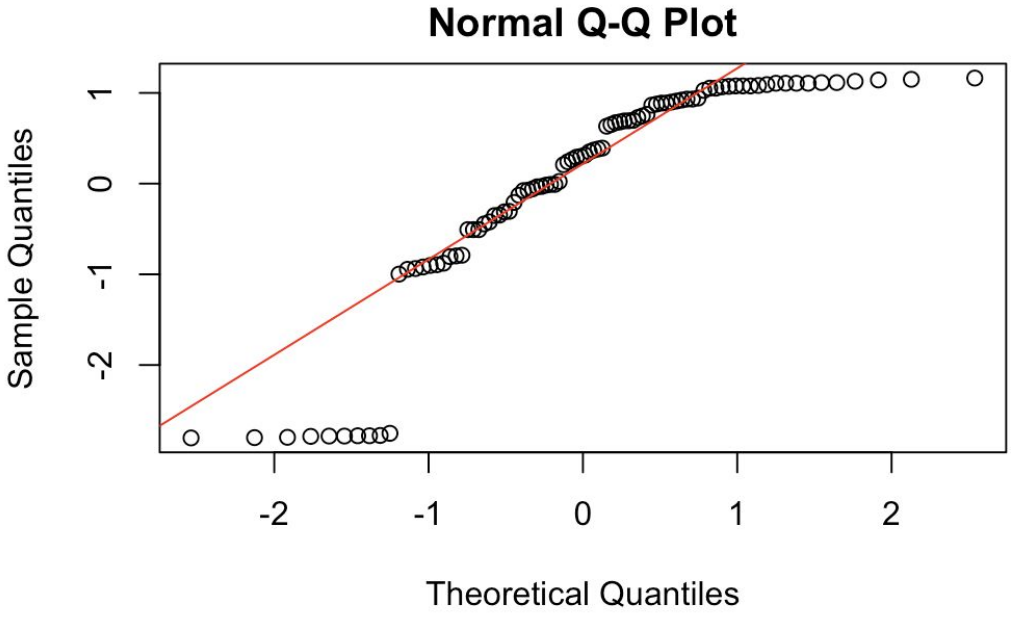


The line fits data poorly. The residual plot is shown below:



There is a clear pattern in the residuals, demonstrating that this linear model is not a good fit. To check for the residuals' normality, we did construct a histogram of the residuals and a quantile-quantile plot.





From the graph above, we can see that the residuals are not normally distributed. The Shapiro-Wilk test yields a p-value 3.256e-09 so we reject null hypothesis - the residuals are not normally distributed.

In order for a least squares line to be appropriate, we need to check three conditions: linearity, nearly normal residuals, and constant variability. We found that only the log transformation of gain satisfied these conditions. The density vs log(gain) scatterplot shows a fairly linear relationship between our explanatory variable and response variable. The histogram and quantile-quantile plot of residuals indicate a normal distribution and that the variances are fairly constant. Moreover, the Shapiro-Wilk test on the residuals gave us p-value of 0.2829, which is greater than our confidence level of 0.05, so we fail to reject the null hypothesis that data is normally distributed. Our residuals follow a normal distribution with approximately constant variability. Our R squared value is 0.9958, which is fairly high. Therefore, we chose log gain regression model as the most appropriate. The equation of our least-squares regression line is

$$\hat{density} = -0.216203 log(gain) + 1.298013$$

Do the residuals indicate any problems with the fit ?

The residual plot of our linear model that uses log transformation of gain to fit density is fairly random. However, it shows slight quadratic curvature.

If the densities of the polyethylene blocks are not reported exactly, how might this affect the fit ?

If the densities of the polyethylene blocks are not reported exactly, this will give us a biased fitted model. By the definition of residual, residual is the difference between the observed data and predicted data. Therefore, if the data is not exact, our residuals will change depending on how inexact the measurements were so our least squares line will also change. To examine this, we round all density values to the nearest 0.01 and try to fit a model again, the resulting model is $\hat{density} = -0.2166~log(gain) + 1.2997$, which is different from our original model.

What if the blocks of polyethylene were not measured in random order?

If the blocks of polyethylene were not measured in random order, inference on the results will be unreliable. In linear models we assumes the error terms to be independent of each other. This case could be true for simple random sampling in cross-sectional data but not in multilevel samples (e.g. data from cluster random

sampling scheme) and longitudinal data (have repeated observations on each individual). The failure in capturing individual difference would result in an inflated Type I error. We suspect our data-set is not iid because we take 30 measurements but only the middle 10 are reported and we have repeated measures on each individual.

Therefore, we would need to perform ANOVA or a multilevel model.

Q2 [Predicting]

We use the least-squares line (with a log transform of density) found in the previous part to make our predictions.

$$\hat{density} = -0.2162 log(gain) +1.2980$$

To find density of snowpack with gain reading 38.6, so we simply substitute 38.6 for “gain” into the equation: $-0.2162*log(38.6) +1.2980 = 0.5081669$
Since we know 38.6 is the average gain of 0.508 density, we can see our model predicts the density well. 0.5081669 is very close to 0.508. To further confirm the accuracy of our model, we calculate a 95% confidence interval for the density.

This is information of our model. Accessed by using summary function coefficients:

```
Call:
lm(formula = density ~ gain, data = dataloggain)

Residuals:
      Min        1Q    Median        3Q       Max
-0.028031 -0.011079 -0.000018  0.011595  0.044911

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.298013   0.006857   189.3   <2e-16 ***
gain        -0.216203   0.001494  -144.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01471 on 88 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9958
F-statistic: 2.096e+04 on 1 and 88 DF,  p-value: < 2.2e-16
```

```
Call:
rq(formula = y ~ x, tau = 0.5)

Coefficients:
(Intercept)           x
  1.2954871  -0.2155711

Degrees of freedom: 90 total; 88 residual
Call:
rq(formula = y ~ x, tau = 0.025)

Coefficients:
(Intercept)           x
  1.2608064  -0.2138912

Degrees of freedom: 90 total; 88 residual
Call:
rq(formula = y ~ x, tau = 0.975)

Coefficients:
(Intercept)           x
  1.3445374  -0.2210623
```

To get a 95% confidence interval of the slope, we use $B_1 \pm 1.96* (standard~error)$.

We store them into variables:
alow: 0.2162 - 1.96 * 0.001494
blow: 1.298013 - 1.96 * 0.006857
ahigh: 0.2162 + 1.96 * 0.001494
bhigh: 1.298013 + 1.96 * 0.006857

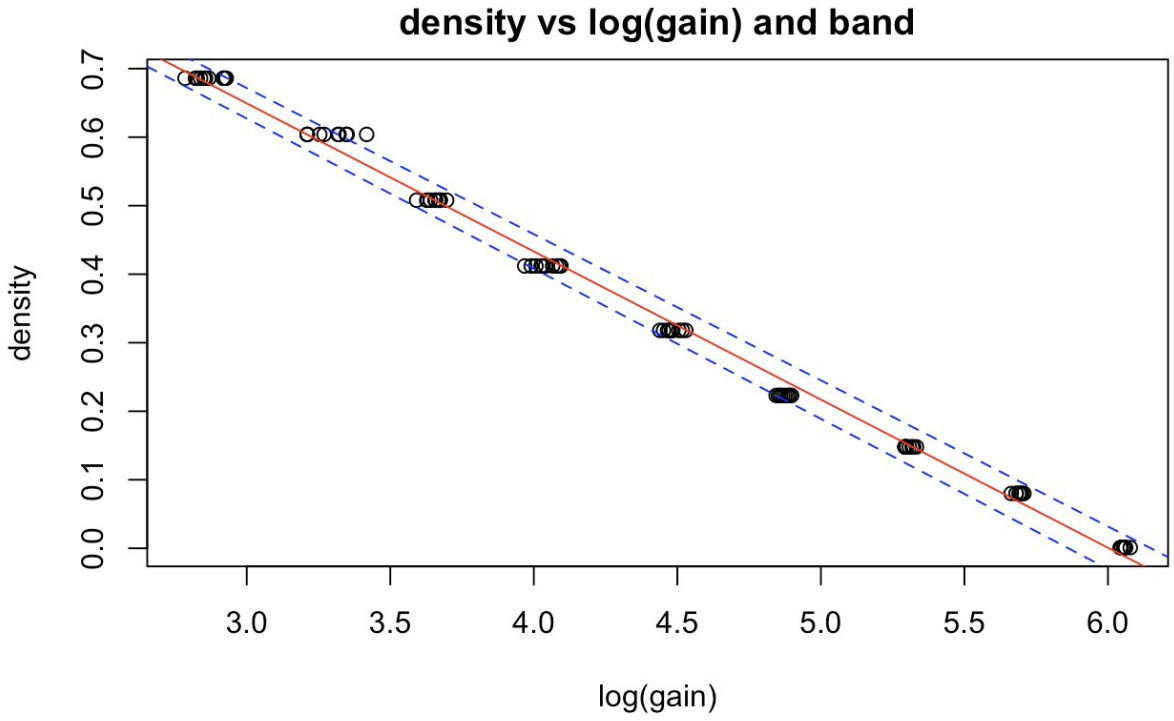
For the lower bound of the confidence interval, our equation is “alow *log(gain) + blow”, we substitute in 38.6 into the equation and obtained a density of 0.4840425. For the upper bound of interval, our equation is “ahigh *log(gain) + bhigh”, we substitute in 38.6 into the equation and obtained a density of 0.5323172g/cm^3. Therefore, the 95% confidence interval of our predicted density based on gain of 38.6 is (0.4840425, 0..5323172), and 0.508 falls inside this interval.

Given a gain reading of 426.7, so we plug in 426.7 in the equation: -0.2162 * log(426.7) +1.2980 = -0.01132475. We know 426.7 is average gain of 0.001 density,

and -0.01132475 is slightly different from 0.001. Thus, we need to find the confidence interval of the density.

We applied the same method as before After substituting in 426.7 into two equations we got an interval of (-0.04248513, 0.01986163). 0.001 is within this interval, which shows that our model's density prediction is within the 95% confidence interval.

Below is plot of least square line with confidence interval bands (bounded by two dashed blue lines):



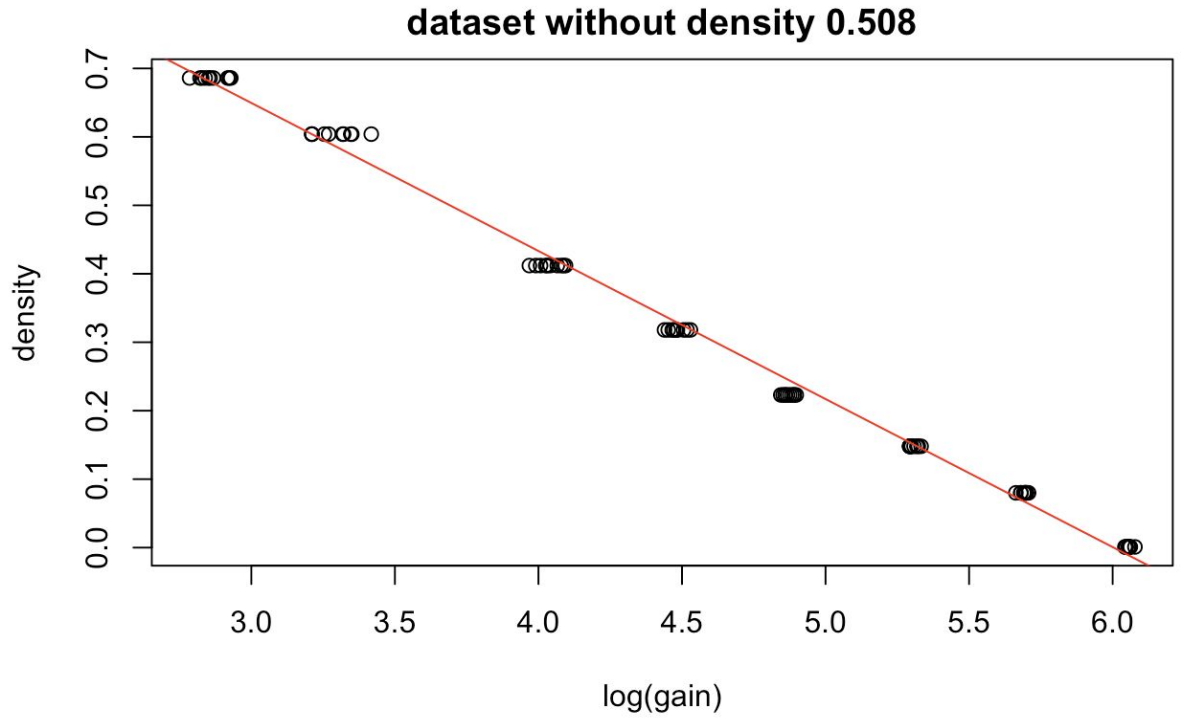
We can see that most data falls inside our 95% confidence interval band.

Q3 [Cross-Validation]

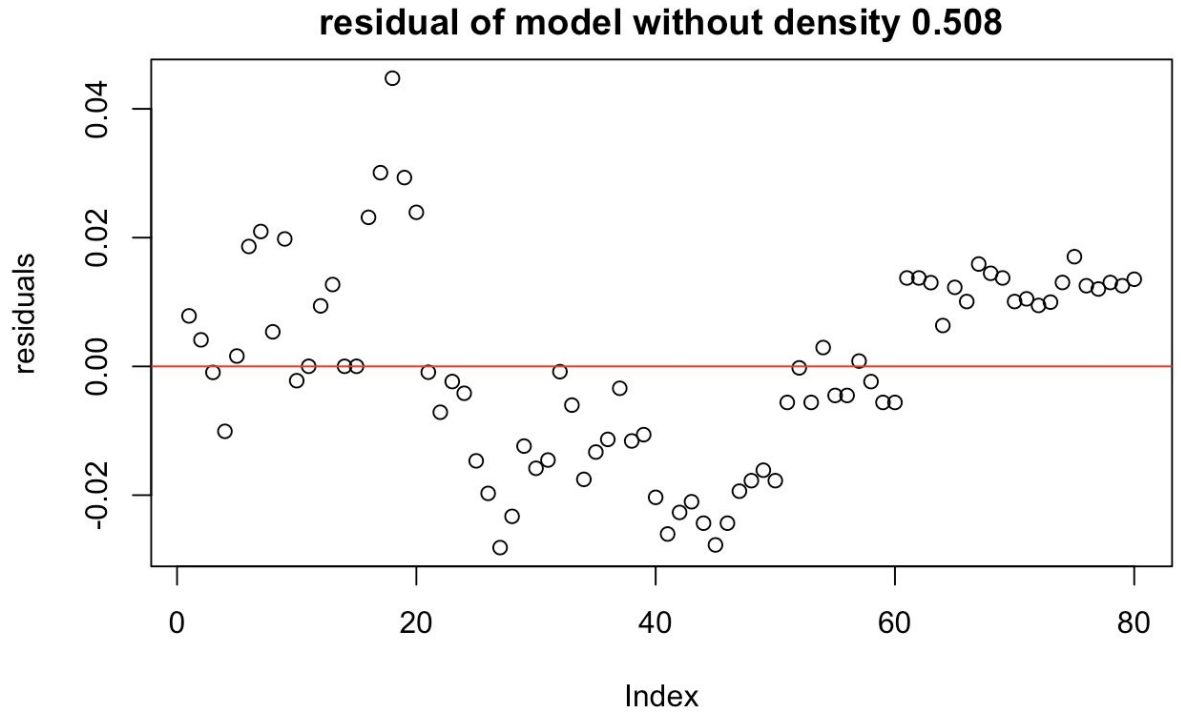
We first omitted corresponding rows with density of 0.508 and use same method as in part 1 to generate a model, still using log(gain) vs density. Again, the least-squares line equation is:

$$\hat{density} = -0.2163 log(gain) + 1.2984$$

We then plotted the data and least squares line below:



The least squares line seems to fit data well and indicates a linear relationship. The residual plot is shown below.



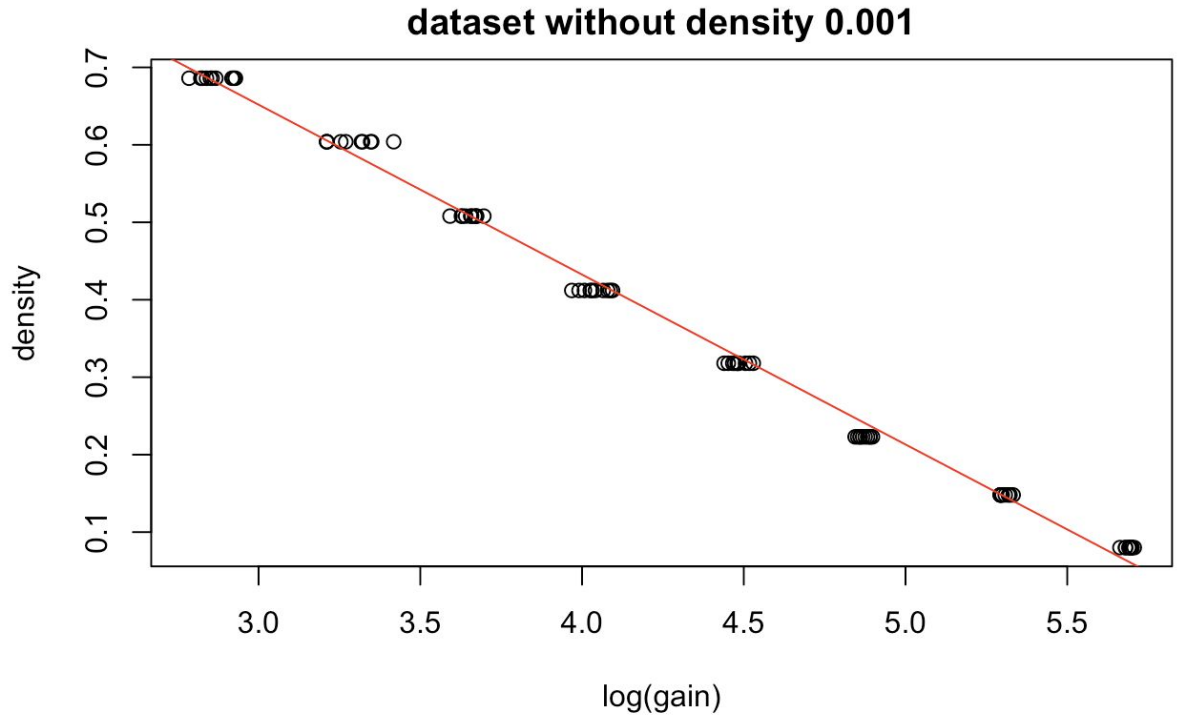
The residual plot shows a fairly random pattern.

We then use predict() function in R to predict an interval of densities. We first construct a matrix of only one element, 38.6, which is the gain we want to use as input. Then we put this matrix into the function as well as model we just trained:

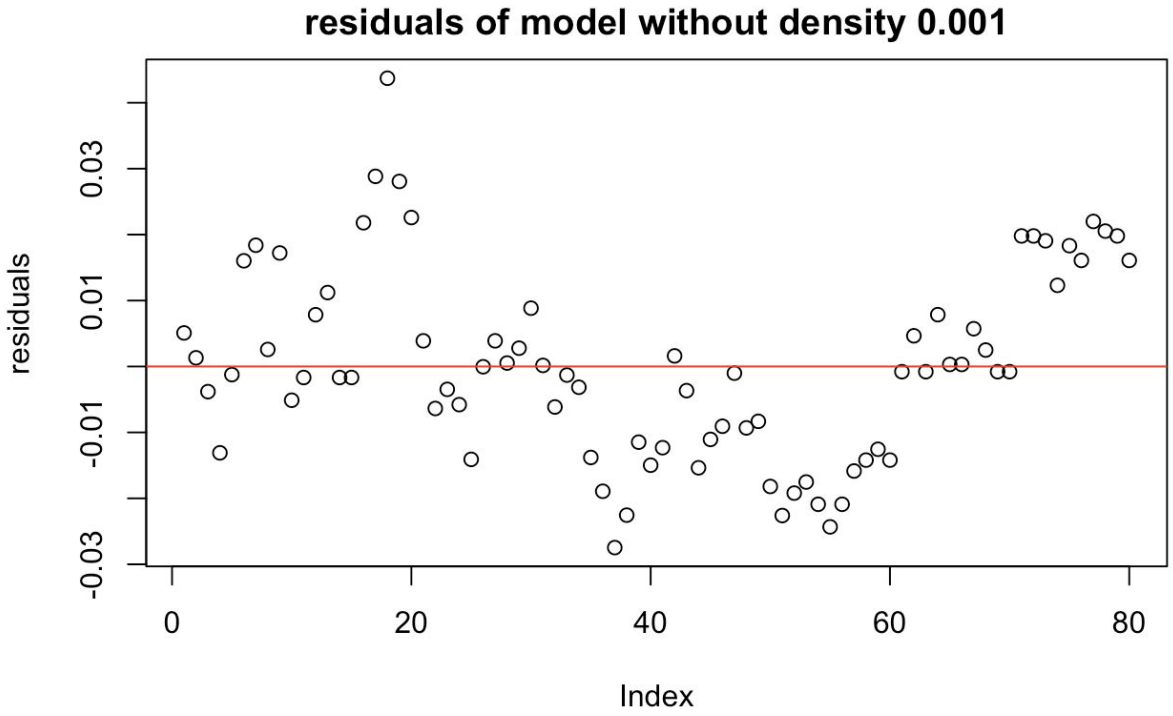
```
        fit       lwr      upr
1 0.5083037 0.4771654 0.539442
```

The model predicts density to be 0.5086094 with a confidence interval of (0.4793015, 0.5379172), 0.508 lies within this interval; therefore, our model did a good job in predicting the density.

We then repeat the same procedure to the dataset, but instead of 0.508, we omitted rows that have densities of 0.001. We trained the model to fit remaining data. The least-squares line equation of model is: $\hat{density} = -0.2194 log(gain) + 1.3101$. The scatterplot and least squares line shown below:



The line seems to fit the data well and that the data has a linear relationship. We construct a residual plot.



The residuals are randomly scattered, indicating our linear model is well-fit.

We again use predict() function to construct the confidence interval. We substitute in a trained model of dataset without densities of 0.001 and a matrix with only element 426.7, average gain for 0.001 density.

```
         fit         lwr         upr
1 -0.0185588 -0.04844409 0.01132649
```

The model predicts the density to be -0.0185588 with a 95% confidence interval of (-0.04844409, 0.01132649); 0.001 falls into this interval. Our model predicted the density accurately.

To further validate our model, we adopt quantile regression and quantile regression also shows that there is a linear relationship between log(gain) and densities. However, the estimated confidence interval has different approximation level. Below is the OLS regression result:

```
Call:
rq(formula = y ~ x, tau = 0.5)

Coefficients:
(Intercept)           x
  1.3070838  -0.2189717

Degrees of freedom: 80 total; 78 residual
Call:
rq(formula = y ~ x, tau = 0.025)

Coefficients:
(Intercept)           x
  1.2958871  -0.2214793

Degrees of freedom: 80 total; 78 residual
Call:
rq(formula = y ~ x, tau = 0.975)

Coefficients:
(Intercept)           x
  1.3486738  -0.2222971

Degrees of freedom: 80 total; 78 residual


Call:
lm(formula = density ~ gain, data = datatrain)

Residuals:
      Min        1Q    Median        3Q       Max
-0.028143 -0.011781 -0.000524  0.012528  0.044757

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.298422   0.007679   169.1   <2e-16 ***
gain        -0.216278   0.001635  -132.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01547 on 78 degrees of freedom
Multiple R-squared:  0.9956,    Adjusted R-squared:  0.9955
F-statistic: 1.749e+04 on 1 and 78 DF,  p-value: < 2.2e-16
```

The OLS result shows that the coefficient on log(gain) is statistically significant.

Below are results of the quantile regression:

```
> summary(olsreg)

Call:
lm(formula = y ~ x, data = data)

Residuals:
      Min        1Q    Median        3Q       Max
-0.028031 -0.011079 -0.000018  0.011595  0.044911

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.298013   0.006857   189.3   <2e-16 ***
x           -0.216203   0.001494  -144.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01471 on 88 degrees of freedom
Multiple R-squared:  0.9958,    Adjusted R-squared:  0.9958
F-statistic: 2.096e+04 on 1 and 88 DF,  p-value: < 2.2e-16

> summary(quantreg25)

Call: rq(formula = y ~ x, tau = 0.025, data = data)

tau: [1] 0.025

Coefficients:
            coefficients lower bd upper bd
(Intercept)  1.26081      1.11887  1.31685
x           -0.21389     -0.23317 -0.18579
```

```
> summary(quantreg50)

Call: rq(formula = y ~ x, tau = 0.5, data = data)

tau: [1] 0.5

Coefficients:
            coefficients lower bd upper bd
(Intercept)  1.29549      1.27155  1.30448
x           -0.21557     -0.21817 -0.21022
```

```
> summary(quantreg75)

Call: rq(formula = y ~ x, tau = 0.75, data = data)

tau: [1] 0.75

Coefficients:
            coefficients lower bd upper bd
(Intercept)  1.30140      1.29166  1.30991
x           -0.21464     -0.21530 -0.21341
```

Call: rq(formula = y ~ x, tau = 0.25, data = data)

tau: [1] 0.25

Coefficients:

	coefficients	lower bd	upper bd
(Intercept)	1.25478	1.21066	1.32016
x	-0.20909	-0.22469	-0.20076

Call: rq(formula = y ~ x, tau = 0.5, data = data)

At 25th percentile, OLS coefficient (-0.216278) falls into the log gain CI (-0.22469, -0.20076)

tau: [1] 0.5

Coefficients:
	coefficients	lower bd	upper bd
(Intercept)	1.29276	1.26612	1.30422
x	-0.21506	-0.21845	-0.20940

Call: rq(formula = y ~ x, tau = 0.75, data = data)

At 50th percentile, OLS coefficient (-0.216278) falls into the log gain CI (-0.21845, -0.20940)

tau: [1] 0.75

Coefficients:
	coefficients	lower bd	upper bd
(Intercept)	1.30398	1.29570	1.32433
x	-0.21510	-0.21836	-0.21387

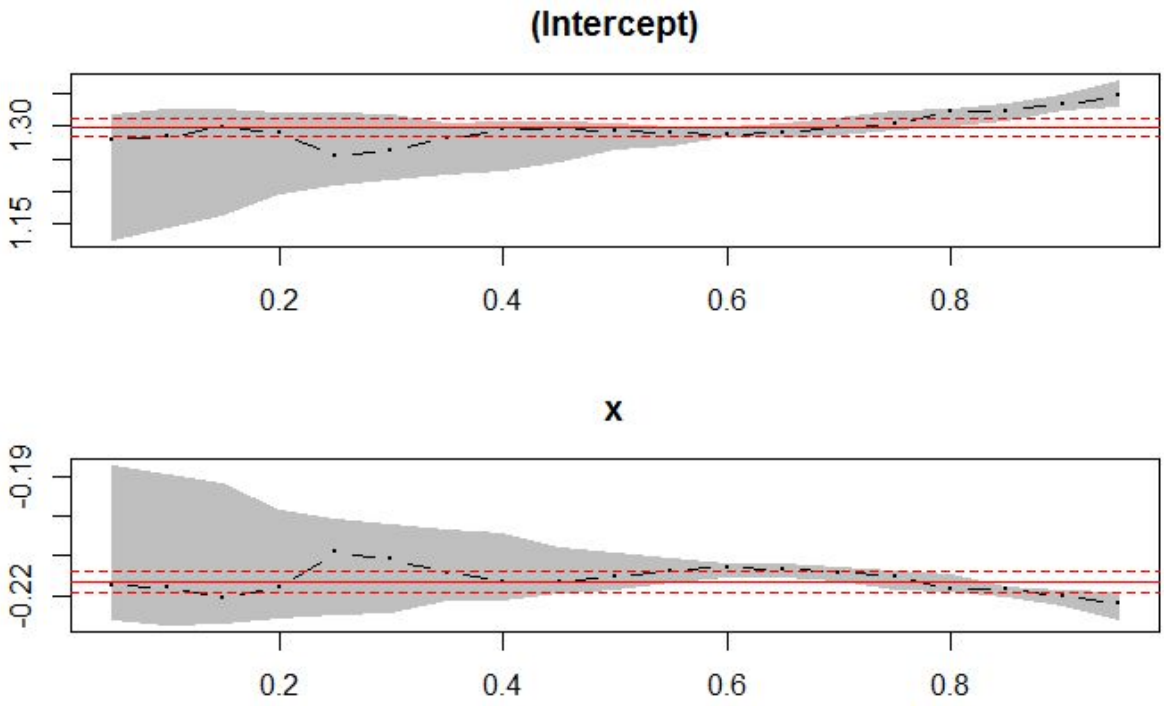
At 75th percentile, OLS coefficient (-0.216278) falls into the log gain CI (-0.21836, -0.21387)

Quantile Regression Analysis of Deviance Table

Model: y ~ x
Joint Test of Equality of Slopes: tau in { 0.25 0.75 }

	Df	Resid Df	F value	Pr(>F)
1	1	159	4.9613	0.02733 *

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



The quantile regression shows that at lower quantile, the coefficient of log(gain) has larger confidence interval relative to the confidence interval at middle quantiles and larger quantiles. This means that at lower quantiles, the model approximation is not as accurate as that in the higher quantile. However, the predicted value from the quantile regression about log(gain) follows closely with the OLS coefficient.

CONCLUSION

The aim of this report is to provide a simple procedure for predicting the density of the polyethylene from gain when the gauge is operating. In part 1, through looking at the patterns in the residual plots from the linear regression, we found out that the residuals have an exponential fit. Thus, we assume the independent variable gain and the dependent variable density to have an exponential relationship. We transform the data by taking the log of the variables. After taking log density, log gain, and log of both variables, we found that log gain vs. density had the best linear regression model that fulfills all assumptions of a least square line and had a high R squared value. Our regression line is: $\hat{density} = -0.2163 log(gain) + 1.298$.

In part two, in order to test our model, we substitute the gain with 38.6 into our model, which predicts that the density would be 0.508. We also derived a 95% confidence interval for the predicted density as (0.4840425, 0..5323172); 0.508 falls inside this interval. We follow the same procedure with gain of 426.7 and found the 95% confidence interval of predicted density to be (-0.04248513, 0.01986163); 0.001 falls in

the interval. Thus our log gain vs density model can accurately predict densities of the polyethylene. In order to further validate of our model, we performed a cross-validation (part 3). After taking out values with block of density 0.508g/cm^3 from our dataset, we substitute 0.508g/cm^3 into our model and found the 95% confidence interval for density is (0.4793015, 0.5379172); the true density 0.508 falls inside this interval. Following the exact procedure with 0.001 density, we found a 95% confidence interval of predicted density to be (-0.04844409, 0.01132649); 0.001 falls inside this interval. Therefore, we are able to conclude that our model fits the data well and we can predict the density value given gain values when the gauge is in operation.

APPENDIX

Methods Used:

Linear regression: an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted by X.

Log transformation: a method to address skewed data.

Quantile regression: type of regression analysis whereas the method of least squares results in estimates that approximate the conditional mean of the response variable given certain values of the predictor variables, quantile regression aims at estimating either the conditional median or other quantiles of the response variable.

REFERENCE

<Bradic1>

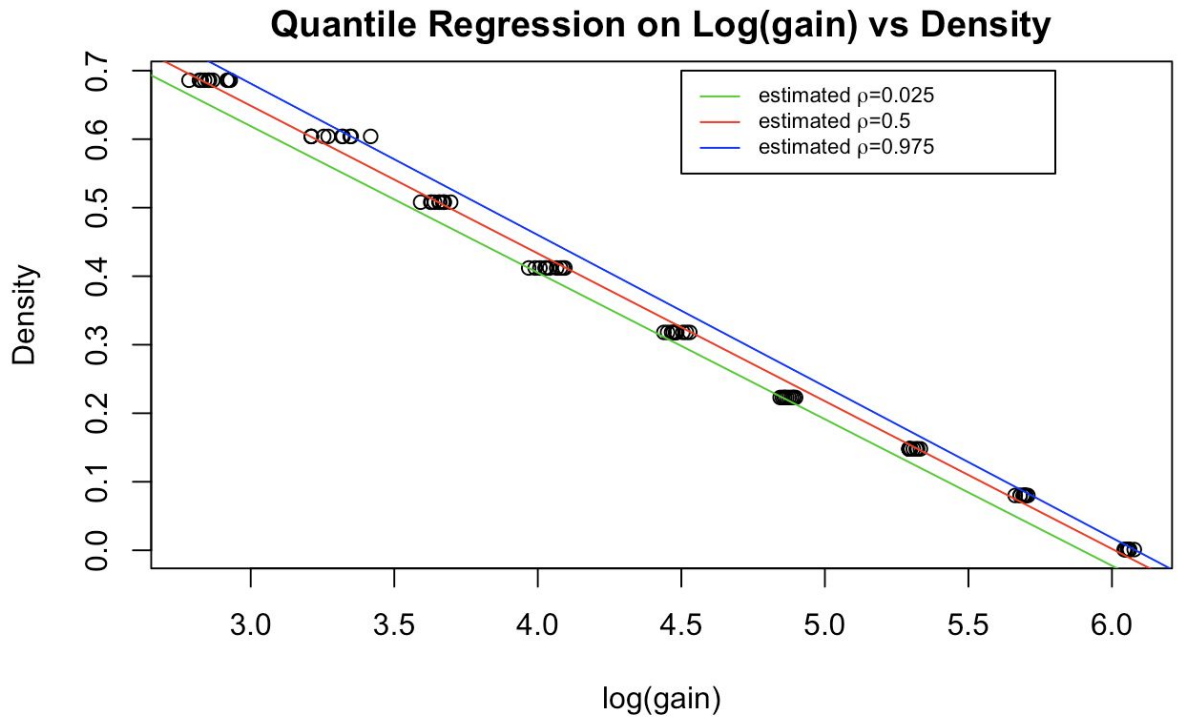
Bradic, Jelena. “Chapter 5: Calibrating a Snow Gauge.” Lecture, 3. March. 2017, Humanities and Social Sciences Building, University of California, San Diego.

<Bradic2>

Bradic, Jelena. “Chapter 6: Linear Model.” Lecture, 3. March. 2017, Humanities and Social Sciences Building, University of California, San Diego.

Additional Analysis

We also applied the techniques of quantile regression. Below is the graph of the quantile regression lines:

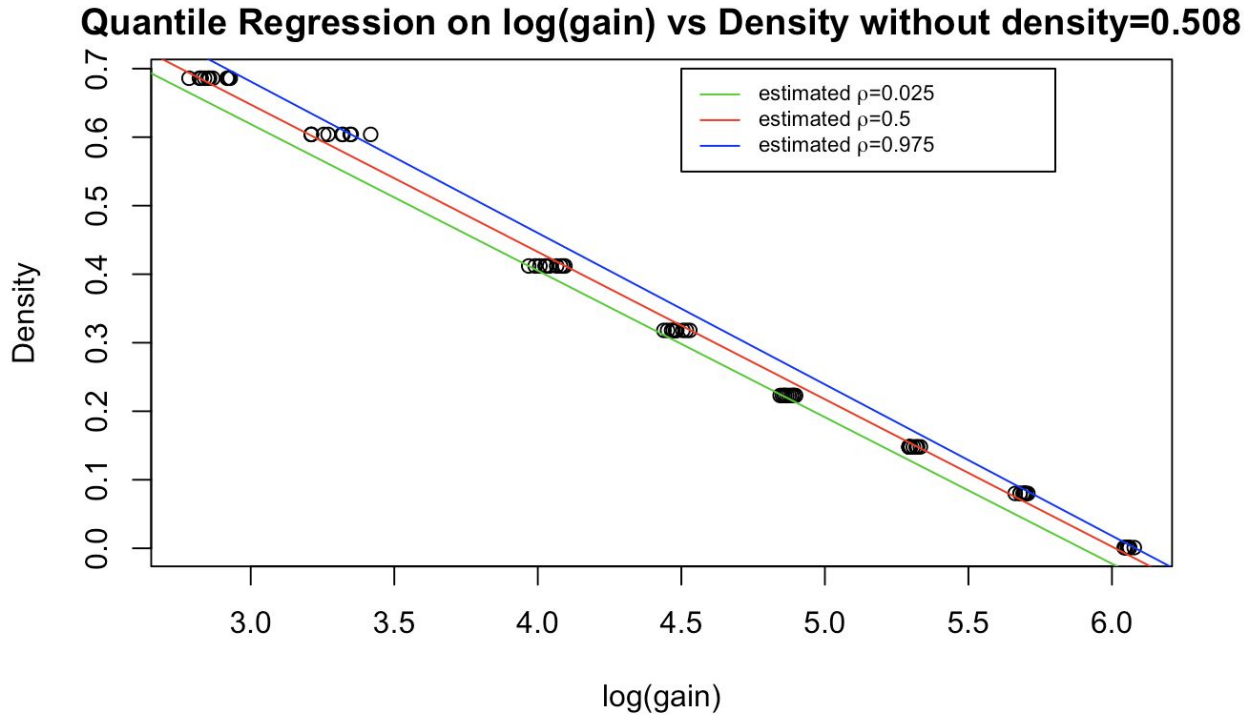


The reason for using quantile regression is to see whether the fitted data lies well within the considered intervals with less outliers. The three quantile regression lines tell us that 2.5% of the data fall below the green line, 50% of the data fall below the red line, and 97.5% of the data fall below the green line. For our data, because regression line of the adjusted(log) data has a linear shape, the quantile regression line had three almost parallel lines. The equations for the three lines are listed as below:

$$
fit1: y = -0.2155711x + 1.2954871 (50% - Red line)
fit2: y = -0.2138912x + 1.2608064 (2.5% - Green line)
fit3: y = -0.2210623x + 1.3445374 (97.5% -Blue line)
$$

Referring to the graph above, most of the adjusted data falls within 2.5%~97.5% range of the quantile regression interval, and the true lines show exact similarity with estimated lines. Thus, it is plausible to confirm that our linear regression line (which is similar to the median line) is highly reasonable.

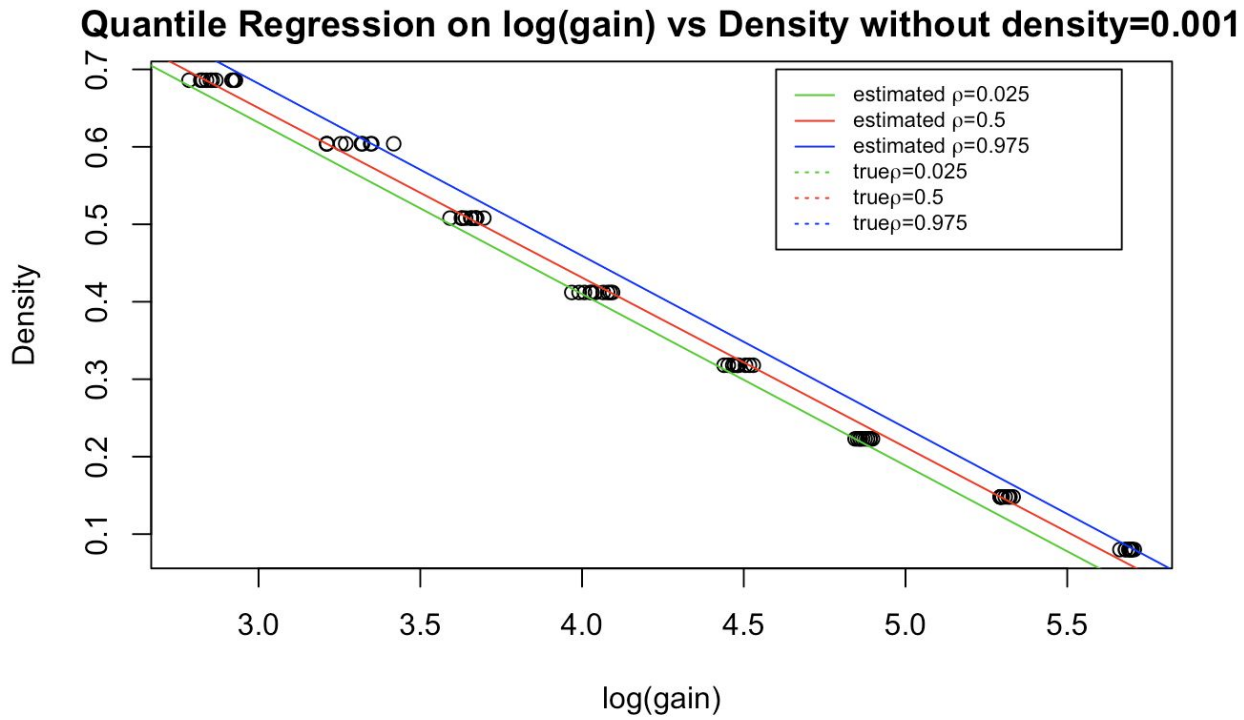
We also repeated quantile regression without density values of 0.508g/cm^3 for question 3. The graph and three equations are as below:



$$ fit1: y = -0.2150597x + 1.2927649 (50\% - Red line) $$
$$fit2: y = -0.2138912x + 1.2608064 (2.5\% - Green line)$$
$$fit3: y = -0.2210623x + 1.3445374 (97.5\% - Blue line)$$

Referring to the graph above, most of the adjusted data falls within 2.5%~97.5% range of the quantile regression interval, and the true lines show exact similarity with estimated lines.

We repeating the quantile regression process without density values of 0.001g/cm^3. The graph and the three equations shown below.



$$fit1: y = -0.2189717x + 1.3070838 (50\% - Red line)$$
$$fit2: y = -0.2214793x + 1.2958871 (2.5\% - Green line)$$
$$fit3: y = -0.2222971x + 1.3486738 (97.5\% - Blue line)$$

Referring to the graph above, most of the adjusted data falls within 2.5%~97.5% range of the quantile regression interval.

https://www.r-bloggers.com/log-transformations-for-skewed-and-wide-distributions-from-practical-data-science-with-r/

http://www.cazaar.com/ta/econ113/interpreting-beta

http://isites.harvard.edu/fs/docs/icb.topic1210732.files/Coefficient%20Interpretation.pdf

Code I have so far (Jason):

```
data <- read.table("gauge.txt", header = TRUE)
data <- data[c(2,1)]

Q1 #fitting
plot(data)
fit1 <- lm(formula=density~gain, data=data)
plot(data, main = "density vs gain")
abline(fit1, col="red")
#residual
qqnorm(fit1$residuals)
qqline(fit1$residuals, col="red")
hist(fit1$residuals, main="Histogram for Residuals", ylab="Frequency",
xlab="Residuals")
plot(fit1$residuals, main = "Residuals of linear model", ylab = “Residuals”)
abline(0, 0, col="red")

#log transformation
#log density
datalogden <- data
datalogden[,2] <- log(data[,2])
plot(datalogden, ylab = "log(density)", main = "log(density) vs gain")
fit2 <- lm(formula=density~gain, data=datalogden)
fit2
abline(fit2, col="red")
qqnorm(fit2$residuals)
qqline(fit2$residuals, col="red")
hist(fit2$residuals, main="Histogram for Residuals (Log Density)", ylab="Frequency",
xlab="Residuals")
plot(fit2$residuals, main = "Residuals of linear model", ylab = “Residuals”)
abline(0, 0, col="red")

#log gain
dataloggain <- data
dataloggain[,1] <- log(data[,1])
plot(dataloggain, xlab = "log(gain)", main = "density vs log(gain)")
fit3 <- lm(formula=density~gain, data=dataloggain)
```

```
fit3
abline(fit3, col="red")
qqnorm(fit3$residuals)
qqline(fit3$residuals, col="red")
hist(fit3$residual, main="Histogram for Residuals (Log Gain)", ylab="Frequency",
xlab="Residuals")
plot(fit3$residuals, main = "Residuals of linear model", ylab = “Residuals”)
abline(0, 0, col="red")
summary(fit3)

#log both
datalog <- data
datalog[,1] <- log(data[,1])
datalog[,2] <- log(data[,2])
plot(datalog, xlab = "log(gain)", ylab = "log(density)", main = "log(density) vs log(gain)")
fit4 <- lm(formula=density~gain, data=datalog)
fit4
abline(fit4, col="red")
qqnorm(fit4$residuals)
qqline(fit4$residuals, col="red")
hist(fit4$residuals, main="Histogram for Residuals (Log Both)", ylab="Frequency",
xlab="Residuals")
plot(fit4$residuals, main = "Residuals of linear model", ylab = “Residuals”)
abline(0, 0, col="red")
```

```
Q2#Predicting
density38.6 <- -0.2162 * (log(38.6)) + 1.2980
density38.6
density426.7 <- -0.2162 * (log(426.7)) + 1.2980
density426.7
summary(fit3)

#confidence interval
alow <- -0.2162 - 1.96 * 0.001494
blow <- 1.298013 - 1.96 * 0.006857
ahigh <- -0.2162 + 1.96 * 0.001494
```

```
bhigh <- 1.298013 + 1.96 * 0.006857
plot(dataloggain, xlab = "log(gain)", main = "density vs log(gain) and band")
abline(fit3, col = "red")
abline(blow , alow, c(10, 20), col = "blue", lty = "dashed")
abline(bhigh, ahigh, c(10, 20), col = "blue", lty = "dashed")
ensity38.6min <- alow * log(38.6) + blow
density38.6max <- ahigh * log(38.6) + bhigh
density426.7min <- alow * (log(426.7)) + blow
density426.7max <- ahigh * (log(426.7)) + bhigh


#Quantile Regression on Log(gain) vs Density Graph
install.packages("quantreg")
library(quantreg)

data[c(2,1)]
x <- log(data[,1])
y <- data[,2]
plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on Log(gain) vs
Density")
# 0.5 quantile(median)
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)
# 0.025 quantile
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)

legend(x = 4.5, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975))), lty = c(1,1,1), col = c(3,2,4), text.width=1, adj=0,cex=0.7)

fit1
fit2
fit3
```

Q3 #cross validation

```
#omit 0.508
datatrain <- dataloggain[-c(21:30), ]
datatest <- dataloggain[c(21:30), ]
fit5 <- lm(formula=density~gain, data=datatrain)
fit5
plot(datatrain, xlab = "log(gain)", main = "dataset without density 0.508")
abline(fit5, col = "red")
plot(fit5$residuals, main = "residual of model without density 0.508", ylab = "residuals")
abline(0, 0, col = "red")
#predicating
new.data <- data.frame(c(log(38.6)))
colnames(new.data) <- "gain"
predict(fit5, new.data, interval = "predict")


datatrain1 <- dataloggain[-c(81:90), ]
fit6 <- lm(formula=density~gain, data=datatrain1)
fit6
plot(datatrain1, xlab = "log(gain)", main = "dataset without density 0.001")
abline(fit6, col = "red")
plot(fit6$residuals, main = "residuals of model without density 0.001", ylab = "residuals")
abline(0, 0, col = "red")
#predicting
new.data1 <- data.frame(c(log(426.7)))
colnames(new.data1) <- "gain"
predict(fit6, new.data1, interval = "predict")


#Quantile Regression code without density = 0.508

datatrain <- data[-c(21:30),]

data[c(2,1)]
x <- log(datatrain[,1])
y <- datatrain[,2]
plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on log(gain) vs
Density without density=0.508")
```

```
# 0.5 quantile(median)
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)
# 0.025 quantile
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)

legend(x = 4.5, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975))), lty = c(1,1,1), col = c(3,2,4), text.width=1, adj=0,cex=0.7)

fit1
fit2
fit3

#Quantile Regression code without density = 0.001
datatrain <- data[-c(81:90),]

data[c(2,1)]
x <- log(datatrain[,1])
y <- datatrain[,2]

plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on log(gain) vs
Density without density=0.001")
# median
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)
# true median
true1 <- x
lines(x, true1, col = 2, lty = 3)

# 0.025 quantile
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# true 0.025 quantile
true2 <- qnorm(p = 0.025, mean = x, sd = x)
```

```
lines(x, true2, col = 3, lty = 3)

# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)
# true 0.7 quantile
true3 <- qnorm(p = 0.975, mean = x, sd = x)
lines(x, true3, col = 4, lty = 3)

legend(x = 4.6, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975)), expression(paste("true", rho, "=", 0.025)),expression(paste("true", rho, "=",
0.5)),expression(paste("true", rho, "=", 0.975))), lty = c(1,1,1,3,3,3), col = c(3,2,4,3,2,4),
text.width=0.8, adj=0,cex=0.7)

fit1
fit2
fit3
```

[PART2 - QUANTILE REGRESSION]

```
install.packages("quantreg")
library(quantreg)

data[c(2,1)]
x <- log(data[,1])
y <- data[,2]

plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on log(gain) vs Density")
# median
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)
# true median
true1 <- x
lines(x, true1, col = 2, lty = 3)

# 0.025 quantile
```

```
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# true 0.025 quantile
true2 <- qnorm(p = 0.025, mean = x, sd = x)
lines(x, true2, col = 3, lty = 3)

# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)
# true 0.7 quantile
true3 <- qnorm(p = 0.975, mean = x, sd = x)
lines(x, true3, col = 4, lty = 3)

legend(x = 4.8, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975)), expression(paste("true", rho, "=", 0.025)),expression(paste("true", rho, "=",
0.5)),expression(paste("true", rho, "=", 0.975))), lty = c(1,1,1,3,3,3), col = c(3,2,4,3,2,4),
text.width=0.9, adj=0,cex=0.7)

fit1
fit3
fit2

[PART3 - QUANTILE REGRESSION WITHOUT DENSITY=0.508]
install.packages("quantreg")
library(quantreg)

datatrain <- data[-c(21:30),]

x <- log(datatrain[,1])
y <- datatrain[,2]

plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on log(gain) vs
Density without density=0.508")
# median
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)
# true median
true1 <- x
```

```
lines(x, true1, col = 2, lty = 3)

# 0.025 quantile
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# true 0.025 quantile
true2 <- qnorm(p = 0.025, mean = x, sd = x)
lines(x, true2, col = 3, lty = 3)

# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)
# true 0.7 quantile
true3 <- qnorm(p = 0.975, mean = x, sd = x)
lines(x, true3, col = 4, lty = 3)

legend(x = 4.8, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975)), expression(paste("true", rho, "=", 0.025)),expression(paste("true", rho, "=",
0.5)),expression(paste("true", rho, "=", 0.975))), lty = c(1,1,1,3,3,3), col = c(3,2,4,3,2,4),
text.width=0.9, adj=0,cex=0.7)

fit1
fit2
fit3

[PART3 QUANTILE REGRESSION WITHOUT DENSITY = 0.001]
install.packages("quantreg")
library(quantreg)

datatrain <- data[-c(81:90),]
data[c(2,1)]
x <- log(datatrain[,1])
y <- datatrain[,2]

plot(x, y, xlab="log(gain)", ylab="Density", main="Quantile Regression on log(gain) vs
Density without density=0.001")
# median
fit1 <- rq(y ~ x, tau = 0.5)
```

```
abline(fit1, col = 2)
# true median
true1 <- x
lines(x, true1, col = 2, lty = 3)

# 0.025 quantile
fit2 <- rq(y ~ x, tau = 0.025)
abline(fit2, col = 3)
# true 0.025 quantile
true2 <- qnorm(p = 0.025, mean = x, sd = x)
lines(x, true2, col = 3, lty = 3)

# 0.975 quantile
fit3 <- rq(y ~ x, tau = 0.975)
abline(fit3, col = 4)
# true 0.7 quantile
true3 <- qnorm(p = 0.975, mean = x, sd = x)
lines(x, true3, col = 4, lty = 3)

legend(x = 4.6, y = 0.7, legend = c(expression(paste("estimated ", rho, "=", 0.025)),
expression(paste("estimated ", rho, "=", 0.5)), expression(paste("estimated ", rho, "=",
0.975)), expression(paste("true", rho, "=", 0.025)),expression(paste("true", rho, "=",
0.5)),expression(paste("true", rho, "=", 0.975))), lty = c(1,1,1,3,3,3), col = c(3,2,4,3,2,4),
text.width=0.8, adj=0,cex=0.7)

fit1
fit2
fit3
```