Alex Hsieh - A11651288 - 4th Year Math and Economics Major
Atharva Fulay - A13140630 - 4th year Applied Math Major
Peter Yao - A12286133 - 3rd year Math-Computer Science Major
Justin Glommen - A13045994 - 2nd year Math-Computer Science Major

Case Study 4

Introduction:

In Northern California, the majority of their natural water supply comes from the Sierra Nevada Mountains. In an effort to better monitor the water supply, the Forest Service Branch of the US Department of Agriculture uses a gamma transmission snow gauge to determine a death profile of the snow density in the mountains. Because of the gamma transmissions that the snow gauge uses, the snow is never disturbed, and can therefore be measured over and over, allowing for the studying of snow-packs throughout the course of the winter. This can also be used to track how rain affects the snow. When rain falls on snow, water is absorbed up to a certain point until the snow cannot hold the water anymore, and at that point, floods begin. Higher snow density is correlated with less water absorption, so monitoring the snow density can also help with analysis of the water supply and flood management. Because the device emits gamma radiation, wear and tear occurs on the device throughout the year, some from radioactive source decay, and some from weather conditions. As a result, a calibration is needed on a yearly basis at the beginning of winter, and we will be developing a procedure to assist with this.

The data set for this case study is specifically from a calibration run which was conducted in the Sierra Nevada Mountains, specifically near Soda Springs. During the run, polyethylene blocks (densities) are placed in between the snow gauge to simulate snow, and from 30 runs, the middle 10 are reported. 90 different measurements are captured in this data set, with the middle 10 gains from 9 different densities.

We also acknowledge the existence of other factors, not integrated into the data set, which can potentially pose as challenges for the data. Among these include the scarcity of these stations which are set up in mountain regions, and the difficulty of sustaining and improving these stations because of their inconvenient locations. Additionally, the measurement tools are not standard across international borders, so data collected internationally is incompatible with domestic data.

Analysis and Methods:

Fitting:

We were provided the data for what the gauge recorded as snow density and the gauge measurements of the gamma photon count, or the gain. To see if we could find a correlation between the density and the gain without any transformation, we plotted the data on a scatter plot and drew a least squares line to fit the data (Figure 1). As a result, we noticed that the line does not fit the data very well. While the least squares line correctly depicts a negative correlation between the density and gain, the data does not follow a linear trend. Therefore, we wanted to take the log of our response variable, the gain, to achieve that.

Figure 2 shows the scatterplot of the density vs. log(gain) and the least squares line to fit the data. We can see that the data follows a linear trend and the line does a much better job at fitting the data. The intercept of the least squares line in Figure 2 is 5.997, meaning that at a density of 0, the estimated average gain of the photon is e^5.997, or about 402, and the slope is -4.606, which indicates a negative correlation. Figure 3 shows the residual plot of the data from Figure 2 and the variability of the residuals looks fairly constant.

When looking at the QQ-plot (Figure 5), it can be seen that there are a few outliers near both ends of the spectrum. Figure 6 shows the fit lines being compared to one another if the outliers were kept or removed from the data set. To compare from the least squares line from figure 2, the least squares line without outliers has an intercept of 6.013 and a slope of -4.653. The least squares line without outliers had the four most outlying points from both ends of the line removed (8 points omitted in total). There is not too much of a difference between the lines - the slope only changes .016 and the intercept only changes by .047.

R-squared is a very important measure in checking how linearly-well-fit the data is. The log of the gain fit outputted an extremely high 0.9958183 R-squared value. This shows that the transformed data fits a linear model very well. The log of the gain fit with the outliers omitted had an even higher 0.9973903 R-squared value. This was fit to the omitted outlier data as well for more consistency.

In Figure 4, we made a histogram of the residuals to check if they are homoscedastic. The skewness of the histogram is 0.1571663, meaning it is slightly skewed right, and the kurtosis is -0.35133, signifying that the sharpness of the peak is marginally light-tailed, but still close to that of a normal distribution's. We can say that the histogram of the residuals represents a normal distribution.

If the densities of the polyethylene blocks were not reported exactly, not only would we not know the density of our snow, our explanatory variable, but we could also retrieve inaccurate data for the gain from the photons. It is possible that if the density of the snow was not properly reported, our whole data set would be flawed and we could arrive at a calibration procedure different from what we would have wanted.

If the blocks of polyethylene were not measured in random order, there could be a dependence on the measurements of the snow densities, and our data would not be i.i.d. If we handpicked which blocks to measure or picked them in a predetermined order, the density of the snow blocks could influence how the gain was recorded for each block of snow and also produce inaccurate results.

<u>Predicting:</u>

Theoretically, based on the fit line that we found, a gain reading of 38.6 will have a log of 3.653. After inverting the formula to solve for density, we obtain a value of 0.5088467. Similarly, a reading of 426.7 should correspond with a density of -0.01282701. This is not reasonable, as the lowest density should theoretically be at 0, but the least squares line shows that these are the theoretical positions of those gain readings. However, it is still a good estimate as it is only 0.02 off of the true value of 0.01 (see Figure 7). We have developed two procedures to calculate this, one that takes gain as an input, and one that takes density as an input.

First, the above was done through a function which took in gain as the input. Inverting the fit-line formula gives us the density. The function additionally returns the upper and lower bounds for the density within a 95% confidence prediction interval for that given gain. These values are calculated by taking the expected fit-line value above and adding and subtracting the interval size. The interval size was calculated by finding the maximum predictable value at that density and then subtracting the minimum predictable value from that density. The value is near .137 when looking at the log(gain). This value is then used to compute the differences in intercepts to calculate the density intervals given gain. These values along with the expected density are returned as a list.

Secondly, to calculate the gain for when a density is given (as prompted below), another function takes an input of a density. Then, using the density, we plug it into the equation of the fit-line that was calculated above to get the gain. Again, the upper and lower limits are given by way of the 95% confidence prediction interval. Similarly, these values and the expected gain with the given density are returned.

Cross-Validation:

A subset of the data was made, with the .508 density block omitted. Using the new dataset, the fit-line gave the following results with 95% confidence prediction intervals as well: Given gain of 38.6, the 95% interval of the density is (0.47932, 0.5385905), with the fit-line expecting the density to be 0.5088467 (View Figure 8).

Due to the ambiguity of the latter portion of the question, we decided to follow through with multiple tests. First, we omitted points at the 0.001 density. Then, at the gain of 38.6, we found that the 95% interval is same results as above. Additionally, the least-squares-line density and was also the same. Then we tested the gain at 426.7, which is where the average gain at 0.001 were. The interval for the density at gain 426.7 is (-0.04235371, 0.01691681) with the linear-fit line expecting a value of -0.01282701. It is explained above why there is no negative density, but this is still valid as it is within the 95% prediction interval.

Conclusion:

The snow gauges slowly wear down and age, to a point where calibration must be completed every year in order to properly measure the gains and snow densities. Therefore, in order to calibrate devices, multiple methods of statistical analysis and regression were performed to help create accurate estimates for snow-packed densities. Regression is the most applicable tool in this scenario due to the frequent inconsistencies and inaccuracies of the devices used throughout the world. Its ability to help congregate the estimates of the distribution into least-squares, maximum-likelihood scenarios validates that assertion and statistically concedes precise results.

Theory:

What is a **least squares line** and why did we use it to fit our data?

The least squares line is represented by the linear equation $\hat{y} = \beta_0 + \beta_1 x$, where $\hat{y}$ is the estimated value of y at value x, β_0 is the intercept, β_1 is the slope and x is the value of the explanatory variable. The least squares method is a standard approach in regression analysis which is commonly used to approximate the solution of overdetermined systems. The overall solution for a least squares regression minimizes the sum of the squares of the error for every equation within the regression. There are two categories of least squares regressions: linear or non-linear, with the model we look at being the linear model.

The biggest advantage of using a least squares line over any other method of fitting our data is that it minimizes residuals, which are calculated by taking the difference of the y value at a data point and the predicted y value of the line. To find the best fit line, we take the sum of the squares of all the residuals and find the smallest possible value. Finding the line is simple for a computer to do and minimizes the effect of having few, but large residuals that could greatly impact our results. When we attempted to find the least squares line of our original, untransformed data, we noticed that the behavior of the data wasn't linear, so we needed to transform our response variable, the gain, to get a much better fit for our data.

How do we analyze the **residuals** and what is the purpose?

Firstly, to be able to analyze the residuals at all, we need to construct the residual plot by fitting the residuals around the least squares line, which is situated to be the x-axis. To analyze the residuals, we want to check for homoscedasticity, meaning that the variability of the residuals around the x-axis is constant, because it tells us that the data follows a linear trend and doesn't indicate any unusual behavior. We should not be able to predict an error for an observation in any way because of the lack of variability. Seeing any trend in a residual plot indicates that there may be a different correlation to some confounding variable or there exists a curvature where using another method to fit the data is preferable or we would need to transform the explanatory or response variable.

One way to check that our residual plot follows homoscedasticity is by graphing our residuals in a histogram and checking for normality. If the residuals are close to normal, then it is unlikely that outliers exist and greatly influence the data. If a histogram is or close to normal, then its skewness, which measures symmetry of the data, and kurtosis, which measures if the data is heavy-tailed or light-tailed, will both be around 0. When we analyzed the histogram for our residuals of the snow density vs. gain of photons, the skewness and kurtosis were both close to 0, indicating normality.

Q-Q Plot:

A Quantile-Quantile Plot is categorized as a probability plot which compares two probability distributions in the manner of plotting their quantiles against each other. The Q-Q Plot is defined on a certain set of intervals, and the resulting line is a parametric curve with a parameter which is the interval for the quantile. If the two measured distributions are similar, the line will be approximately x=y. If the positions are just linearly related, they will lie on a line but not necessarily x=y. The most common usage of a Q-Q plot is to compare collections of data, and provides a view of how location, scale, or skewness affect the relationship between the distributions. It is generally more powerful to use a Quantile-Quantile plot over histogram comparison, and are commonly used to compare a data set versus a theoretical model. In the case of our experiment, we used the Q-Q Plot to compare a theoretical data set to our sample data set of gains and densities, and to determine goodness-of-fit and outliers in our data set.

Appendix:

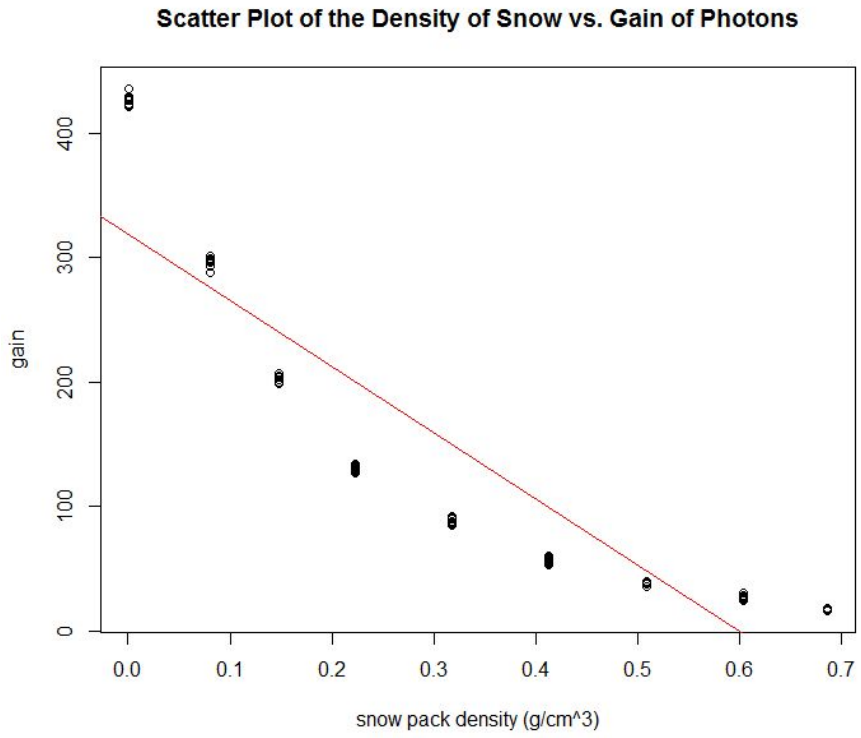


Figure 1. The least squares line of the density vs. gain scatterplot with no transformation.

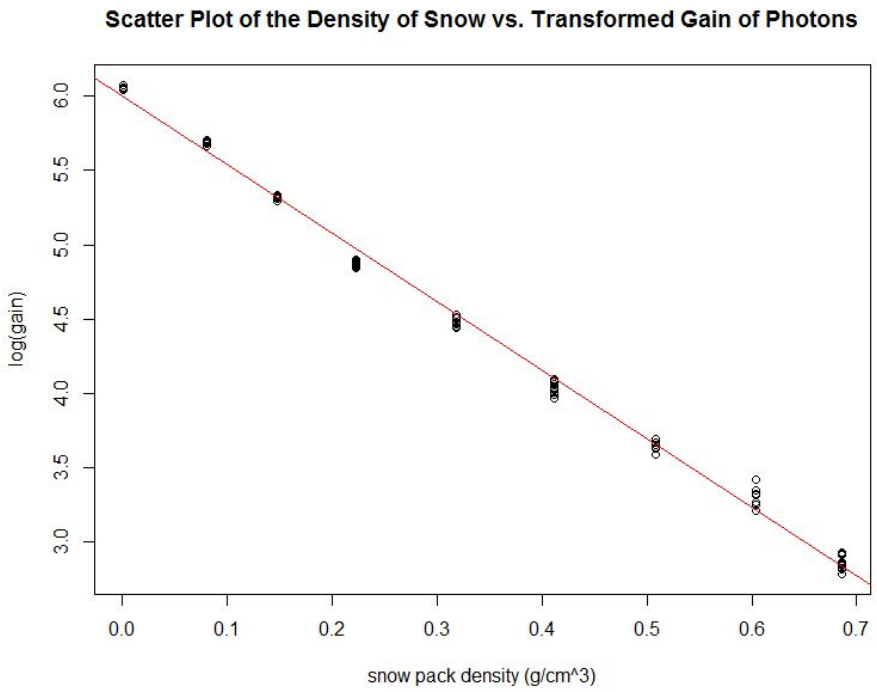


Figure 2. The least squares line of the density vs. gain scatterplot after transforming gain. The line is $\hat{y} = 5.997 - 4.606x$.

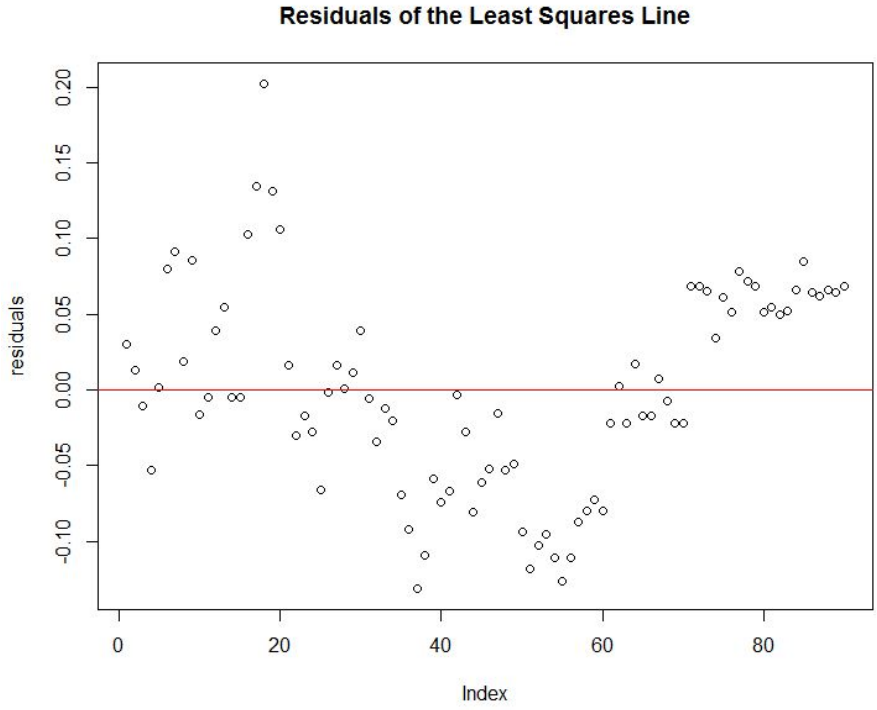


Figure 3. The residual plot of the least squares line from Figure 2.

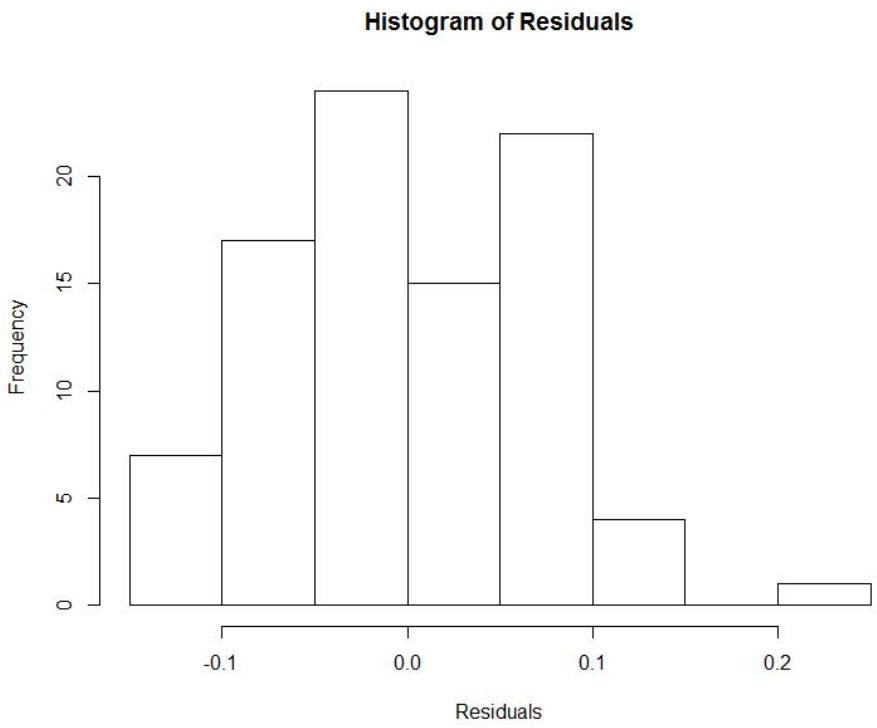


Figure 4. The histogram of the residual plot.

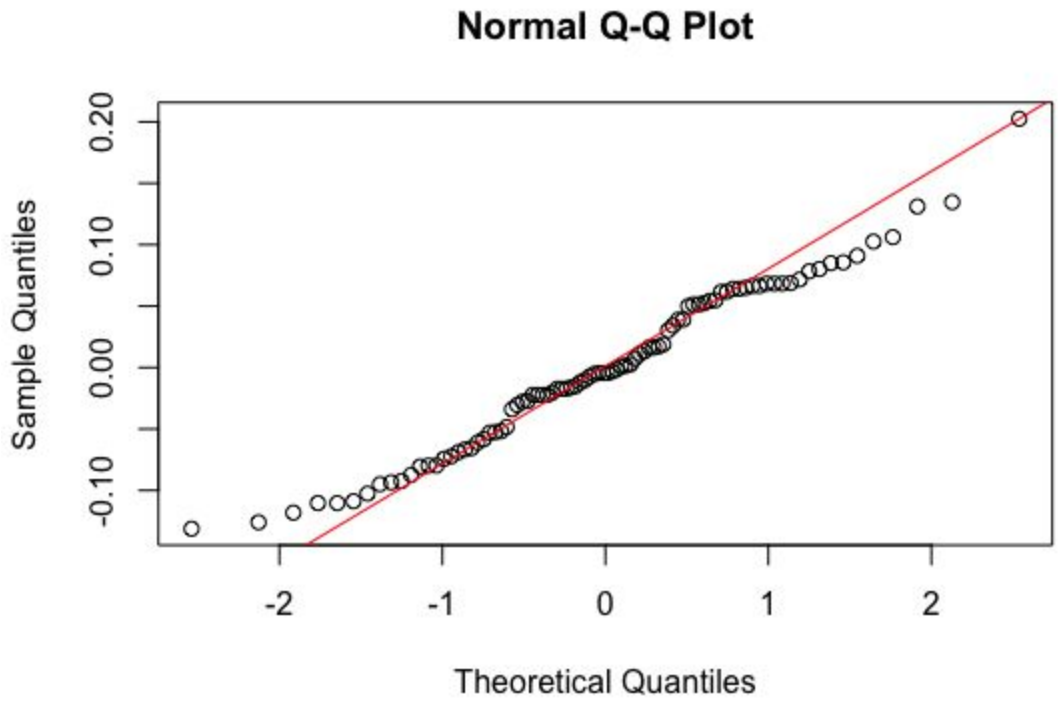


Figure 5. Quartile-quartile plot of the theoretical quantiles versus the sample quantiles.

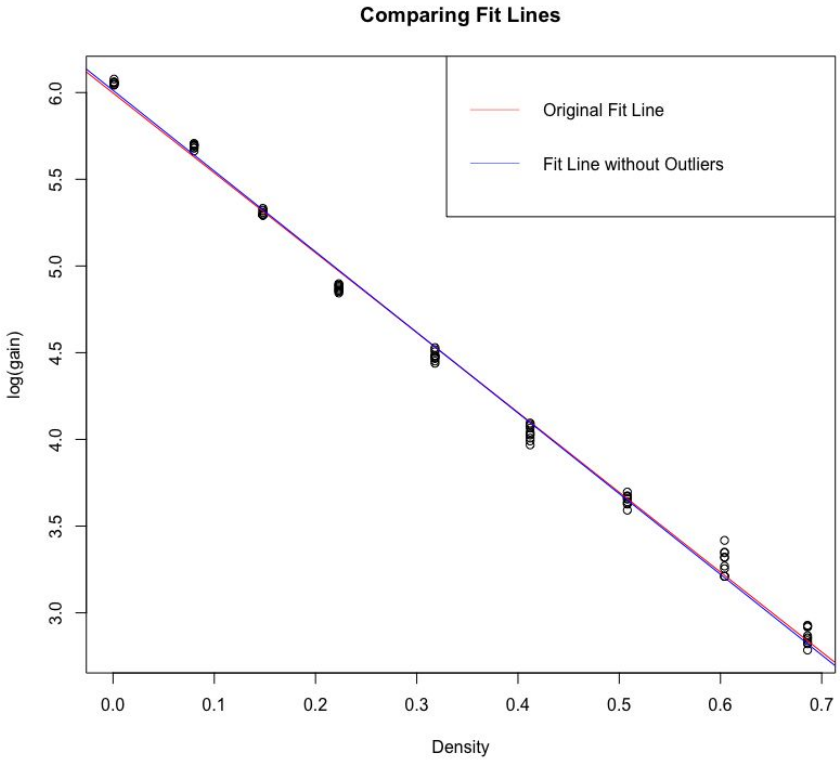


Figure 6. This graph shows the comparison when the outliers are omitted from the dataset.

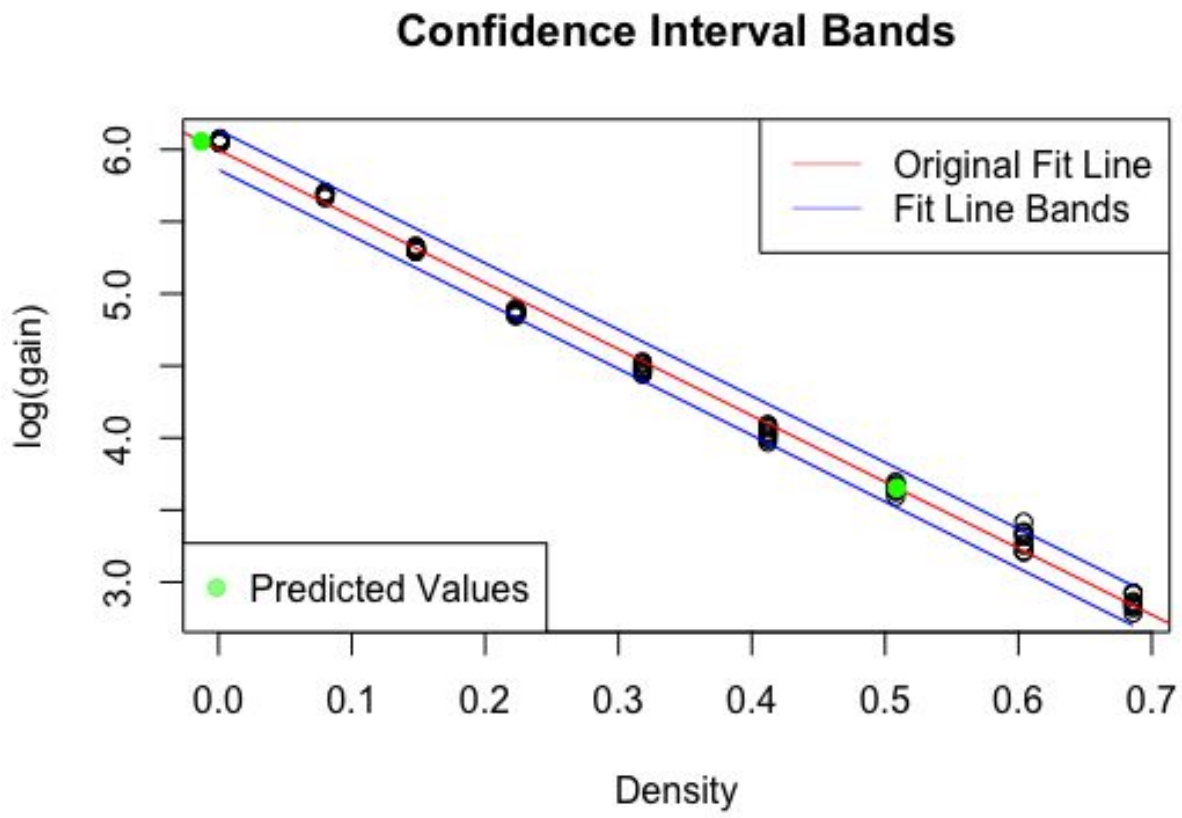


Figure 7. The original least squares regression line with predictive bands in blue.

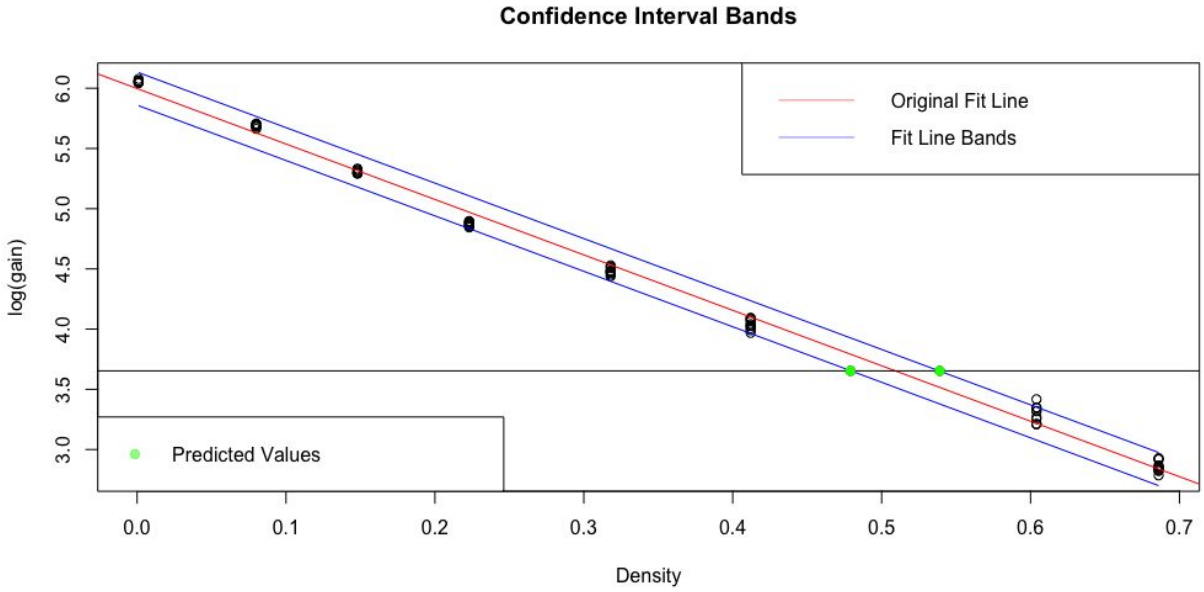


Figure 8. The prediction interval for the regression line of the density versus log of gains.

Case Study 4: Snow Gauge

Justin Glommen
Peter Yao
Atharva Fulay
Alex Hsieh

3/7/2017

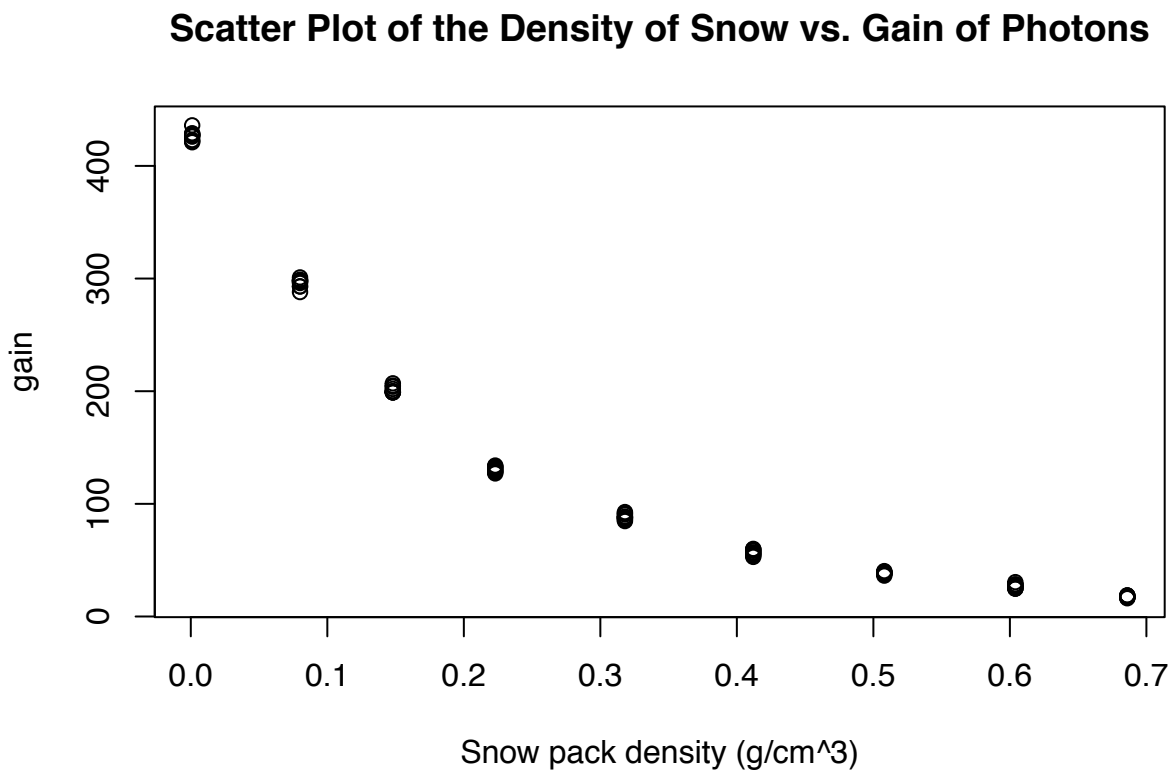
Setup

Here the data is loaded into its corresponding variable for analysis.

```
gauge <- read.table("gauge.txt", header=TRUE)
data <- gauge # A duplicate to be used for later direct manipulation
head(gauge)
```

```
##   density gain
## 1   0.686 17.6
## 2   0.686 17.3
## 3   0.686 16.9
## 4   0.686 16.2
## 5   0.686 17.1
## 6   0.686 18.5
```

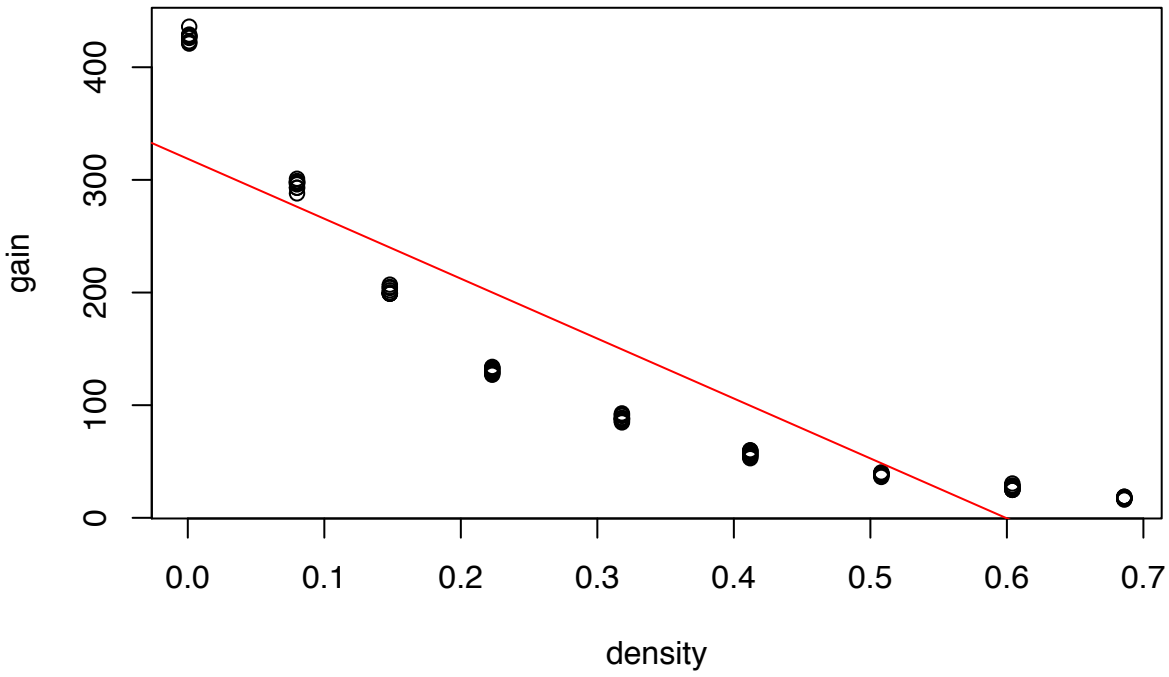
```
stringMain <- "Scatter Plot of the Density of Snow vs. Gain of Photons"
densityLabel <- "Snow pack density (g/cm^3)"
gainLabel <- "gain"
plot(gauge, xlab = densityLabel , ylab = gainLabel, main = stringMain)
```



Fitting

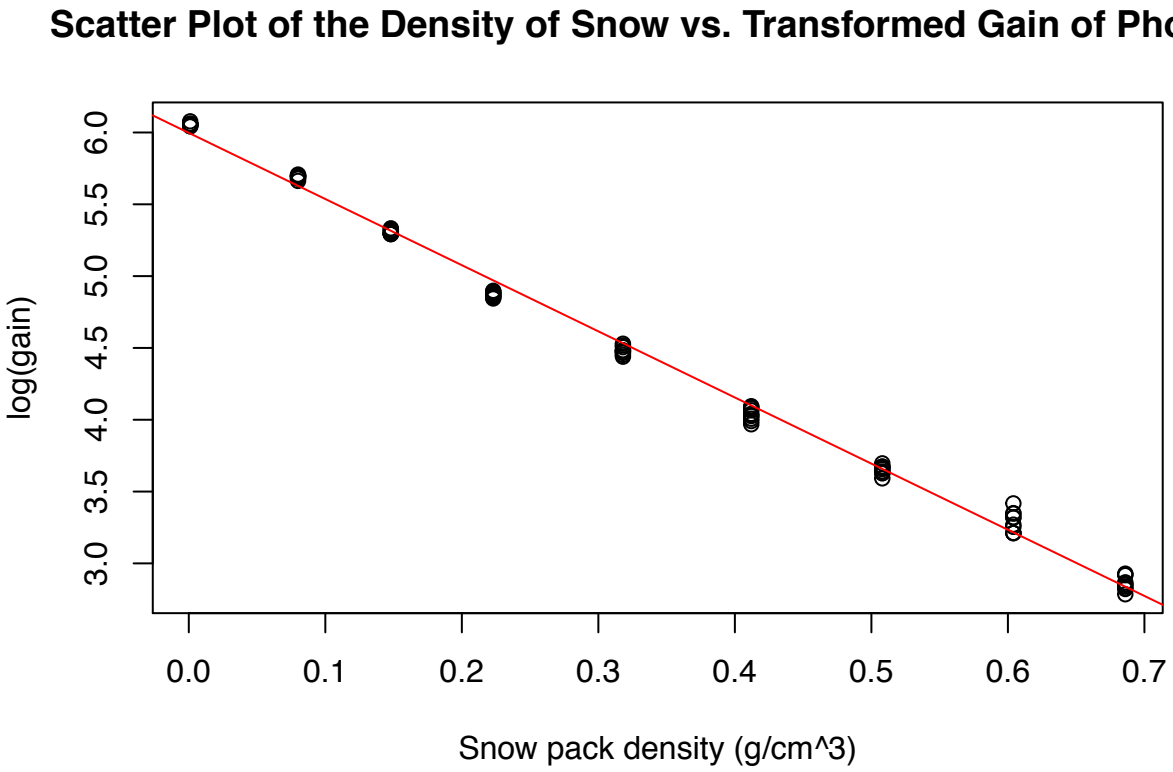
We now want to fit the lease squares line of the original data, such that we can plot and use it to help us make predictions about the snow-pack densities.

```
# Fit least squares line of orig data
fit <- lm(formula=gain~density, gauge)
plot(gauge)
abline(fit, col="red")
```



Here, we notice that the data follows a somewhat exponential-like pattern. Therefore, in order to enhance the regression line, we will take the log of the densities and display that instead.

```
#transformed data log(gain)
gauge$gain <- log(gauge$gain)
#fit least squares line of trans data
fit <- lm(formula=gain~density, gauge)
plot(gauge, xlab = densityLabel, ylab = yLogGainLabel, main = transformedStringMain)
abline(fit, col="red")
```




```
#find R^2 of trans data (.9958183)
R.square <- sum((fit$fitted.values-mean(gauge$gain))^2) / (sum((gauge$gain - mean(gauge$gain))^2))
R.square
```

```
## [1] 0.9958183
```

Predicting

Two functions are presented which allow for us to generate an estimate density interval based on a passed in gain, as well as the ability to do the opposite; generate an estimate gain interval based on the passed in density.

```
estimating_fn <- function(gainInput){
  logGain = log(gainInput);
  gain <- gainInput

  estDen = (logGain - 5.997) / -4.606;

  # Values are the intercepts of where the upper and lower limits
  # intersect the y-axis
  den_low = (logGain - 5.861)/ -4.606;
  den_high = (logGain - 6.134)/ -4.606;

  result <- matrix(data=NaN, nrow = 3, ncol = 1)
  result <- c(den_low, estDen, den_high);

  return(result);
}
```

```
estimating_gain_fn <- function(denInput){

  logGain <- denInput*(-4.606) + 5.997;
  gain <- exp(logGain)

  upper_lim_log <- logGain + 0.138;
  lower_lim_log <- logGain - 0.138;
  upper_lim <- exp(upper_lim_log)
  lower_lim <- exp(lower_lim_log)


  result <- matrix(data=NaN, nrow = 2, ncol = 3)
  result[1,] <- c(lower_lim_log, logGain, upper_lim_log)
  result[2,] <- c(lower_lim, gain, upper_lim)

  return(result);
}
```

Here we construct prediction bands, which acts as a region in which with 95% certainty we can expect the regression line to lie. Therefore, this is helpful in also constructing confidence intervals for the prediction of densities of snow given the gain.

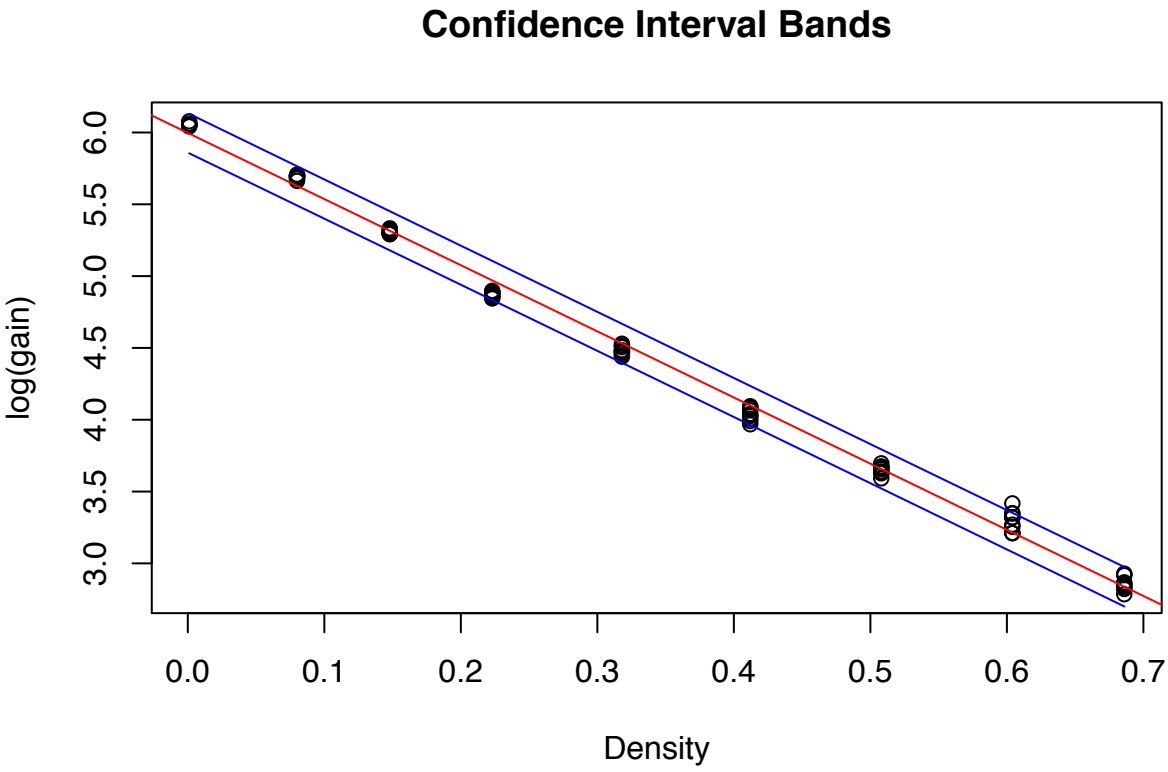
```
# Initializing data differently to avoid conflict with previous data
gain <- data["gain"];
log_gain = log(gain);
data["log_gain"] <- log_gain;
linear_data <- data["density"];
linear_data["log_gain"] <- data["log_gain"];

#----------------------------prediction intervals / bands -----------------------------------------
fit <- lm(formula=log_gain~density, data=linear_data);
pred.int =  predict(fit,interval="prediction", level=.95)
```

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

```
pre.test = predict(fit,interval="prediction")
```

```
## Warning in predict.lm(fit, interval = "prediction"): predictions on current data refer to _future_ r
```

```
fitted.values = pred.int[,1]
pred.lower = pred.int[,2]
pred.upper = pred.int[,3]


plot(linear_data, xlab="Density", ylab="log(gain)", main="Confidence Interval Bands");
abline(fit, col="red", lwd=1);
lower = lines(linear_data$density,pred.lower[1:90],lwd=1,col="blue")
upper = lines(linear_data$density,pred.upper[1:90],lwd=1,col="blue")
```



Cross Validation

Now, in order to cross validate that our regression prediction interval is correct, we want to remove the .508 density values in our sample, and instead run the average gain for it through a function which generates the point on the best fit line; the estimate density. It's important to note that here, the estimator function returns the confidence interval of the gain, the log of the gain, and returns the estimated density.

```
# Here we omit the following rows in order to eliminate the .508 density,
# as per request of the assignment
data_omit = gauge[-c(21:30), ]

# Here we test the following input test densities and gains
testGain <- 38.6
testDens <- .508
# Should result in a .508 density estimate
estimating_fn(testGain)
```

```
## [1] 0.4793200 0.5088467 0.5385905
```

```
# Should result in a 38.6 estimate
estimating_gain_fn(testDens)
```

```
##           [,1]      [,2]      [,3]
## [1,]  3.519152  3.657152  3.795152
## [2,] 33.755791 38.750823 44.484998
```

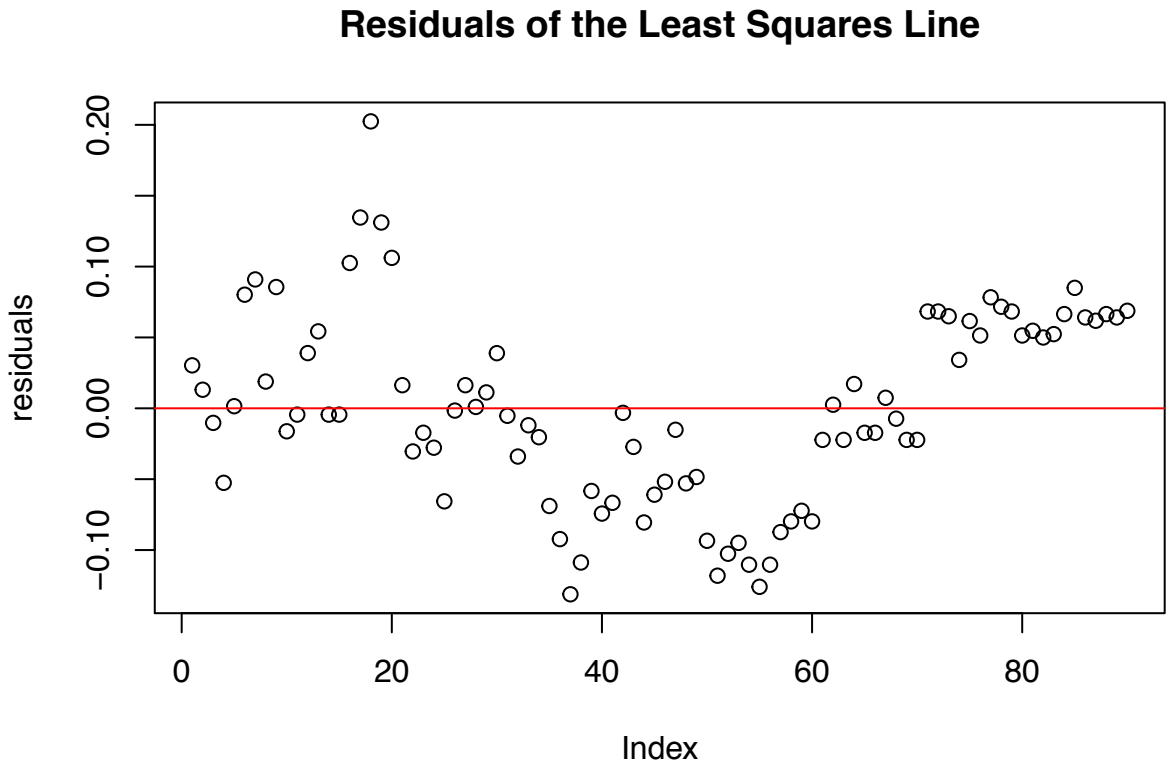
```
# Should result in a 400 estimate
testDens <- .001
estimating_gain_fn(testDens)
```

```
##            [,1]       [,2]       [,3]
## [1,]   5.854394   5.992394   6.130394
## [2,] 348.763485 400.371954 459.617214
```

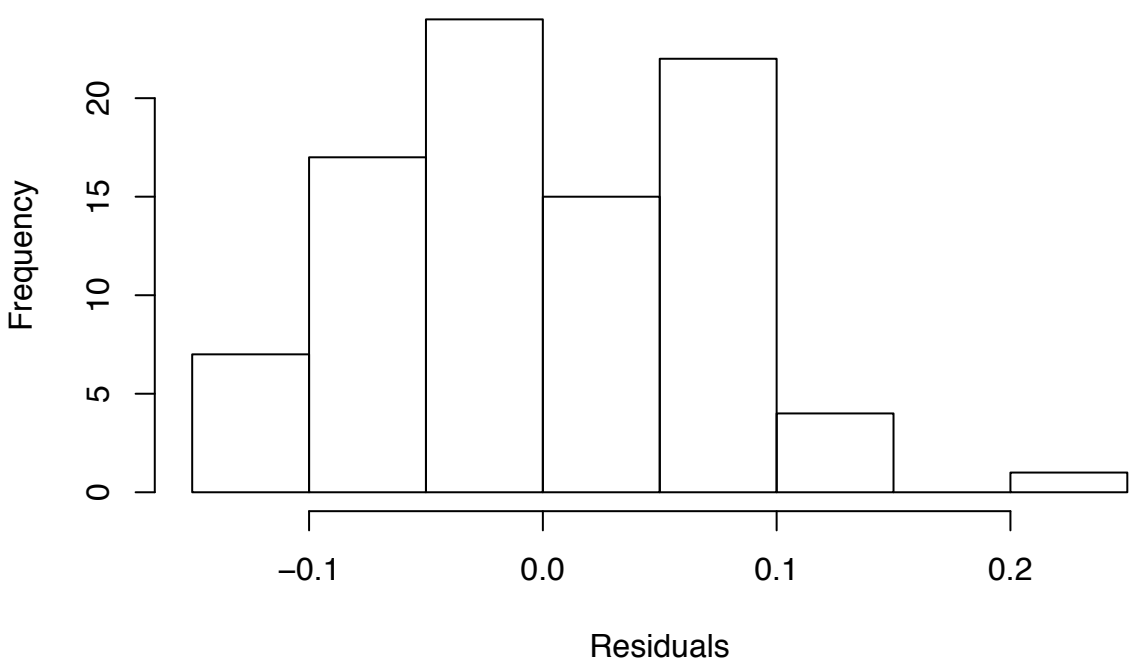
Clearly here, our result is correct and matches that closely to the original dataset.

Extra Analysis

```
#residuals of trans data
plot(fit$residuals, ylab = "residuals", main = "Residuals of the Least Squares Line")
abline(0, 0, col="red")
```




```
hist(fit$residuals, xlab = "Residuals", main = "Histogram of Residuals")
```




```
library(e1071)
skewness(fit$residuals)
```

```
## [1] 0.1571663
```

```
kurtosis(fit$residuals)
```

```
## [1] -0.35133
```

```
x <- gauge[["gain"]]
quantile(x, probs = seq(0.1, 0.9, by = 0.2))
```

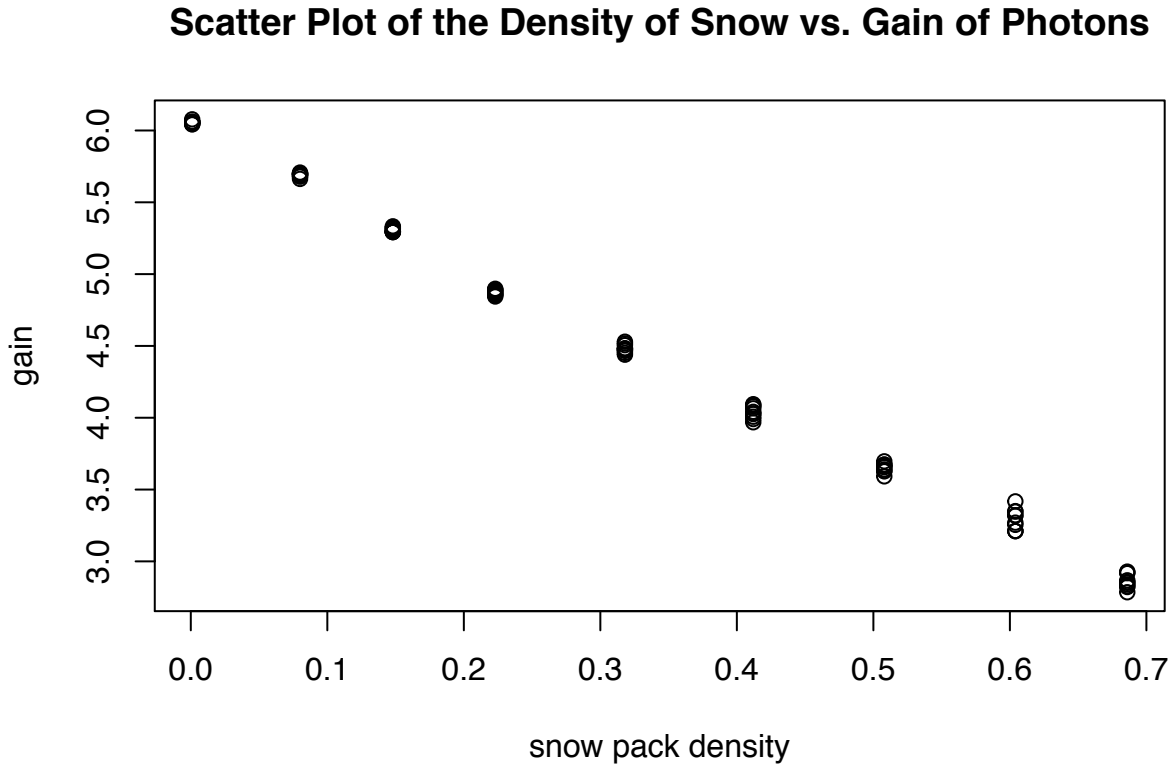
```
##      10%      30%      50%      70%      90%
## 2.927987 3.672239 4.480174 5.293305 6.042870
```

```
#quantile regression
plot(gauge, xlab = "snow pack density", ylab = "gain", main = "Scatter Plot of the Density of Snow vs. G

library(quantreg)
```

```
## Loading required package: SparseM
```

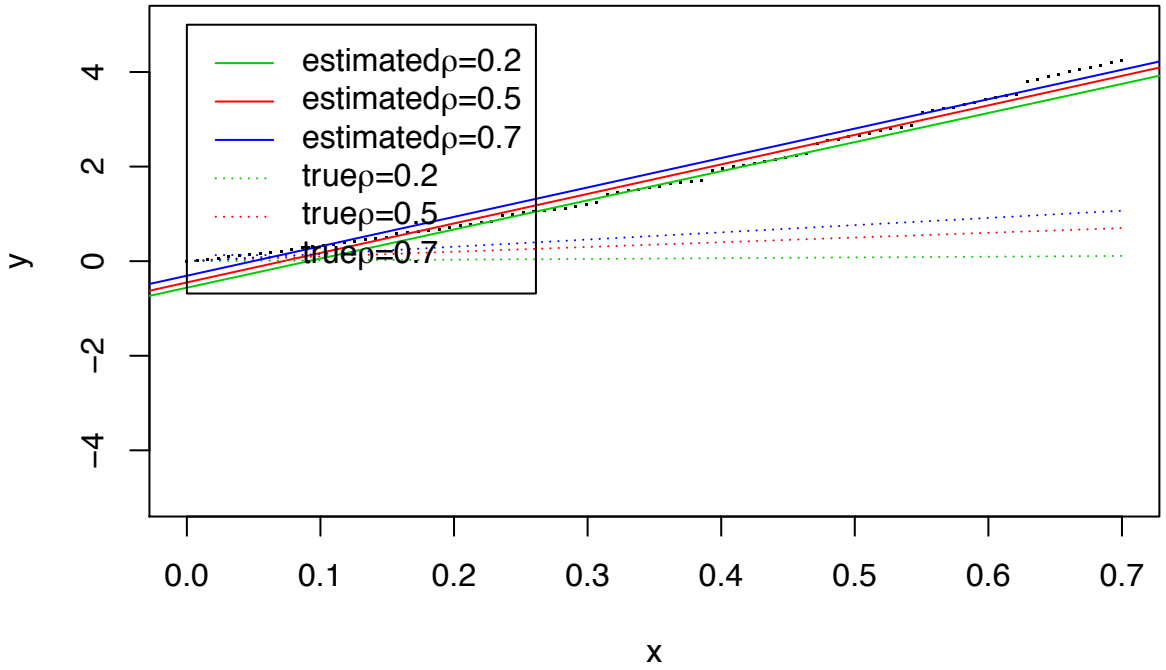
```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```



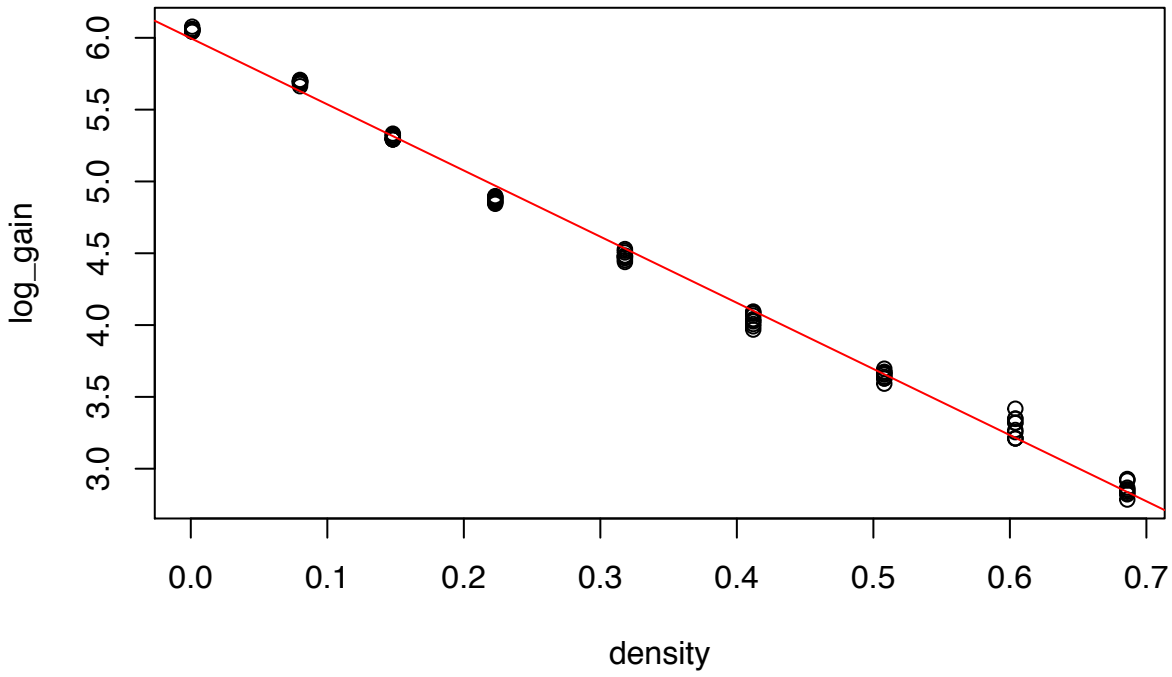

```
x <- seq(0, 0.7, length.out = 90)
y <- x*gauge[["gain"]]
plot(x, y, pch = ".", ylim = c(-5, 5))
# median
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)

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

# 0.2 quantile
fit2 <- rq(y ~ x, tau = 0.2)
abline(fit2, col = 3)

# true 0.2 quantile
true2 <- qnorm(p = 0.2, mean = x, sd = x)
lines(x, true2, col = 3, lty = 3)

# 0.7 quantile
fit3 <- rq(y ~ x, tau = 0.7)
abline(fit3, col = 4)

# true 0.7 quantile
true3 <- qnorm(p = 0.7, mean = x, sd = x)
lines(x, true3, col = 4, lty = 3)

legend(x = 0, y = 5, legend = c(expression(paste("estimated", rho, "=", 0.2)),
                                expression(paste("estimated", rho, "=", 0.5)),
```

```
                                expression(paste("estimated", rho, "=", 0.7)),
                                expression(paste("true", rho, "=", 0.2)),
                                expression(paste("true", rho, "=", 0.5)),
                                expression(paste("true", rho, "=", 0.7))),
       lty = c(1,1,1,3,3,3), col = c(3,2,4,3,2,4))
```

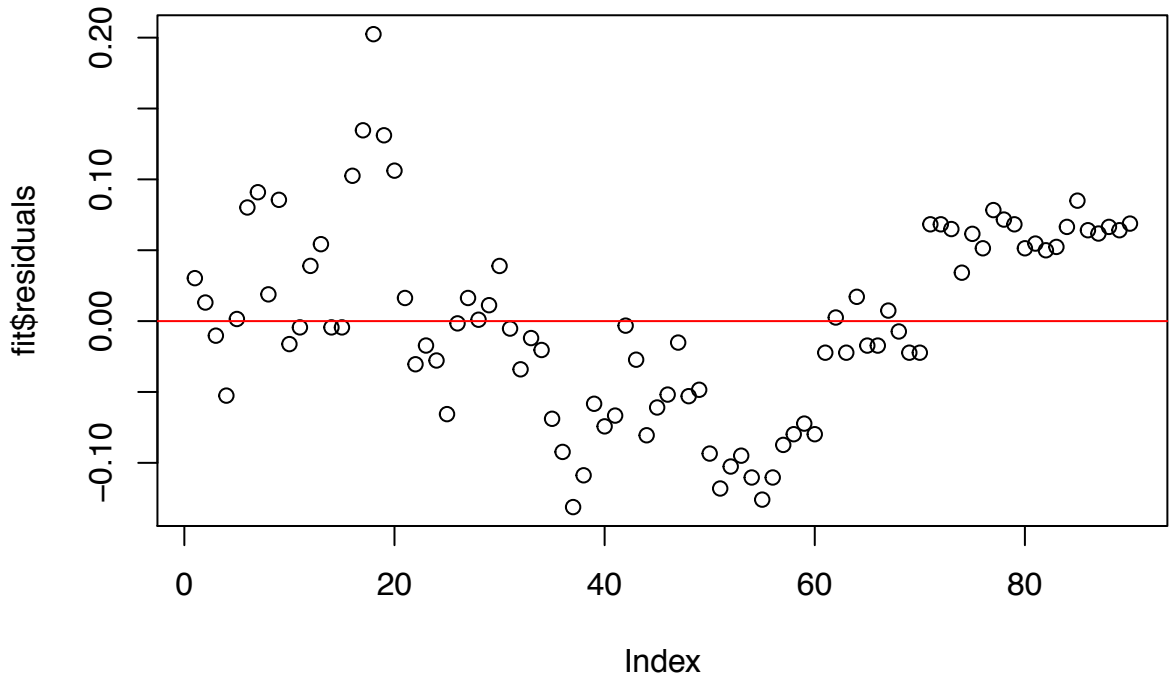


Additional analysis and graphs generated for inspection.

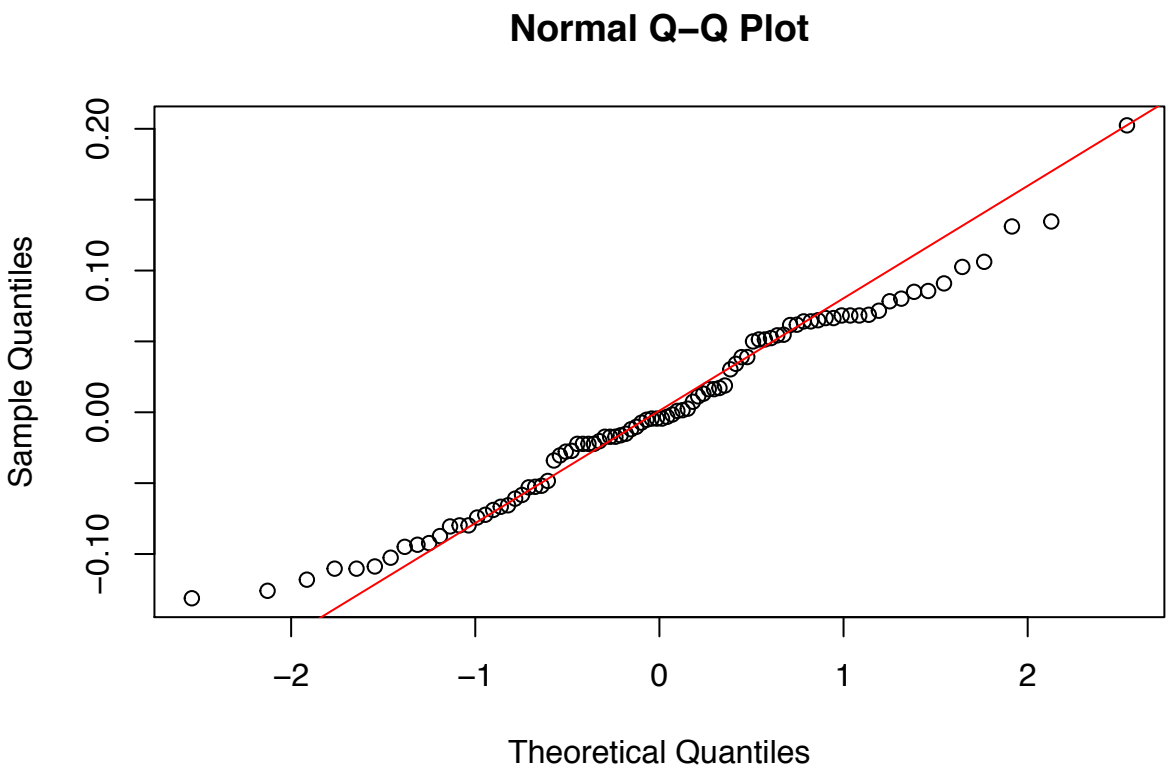
```
fit <- lm(formula=log_gain~density, data=linear_data);

plot(linear_data);
abline(fit, col="red");
```



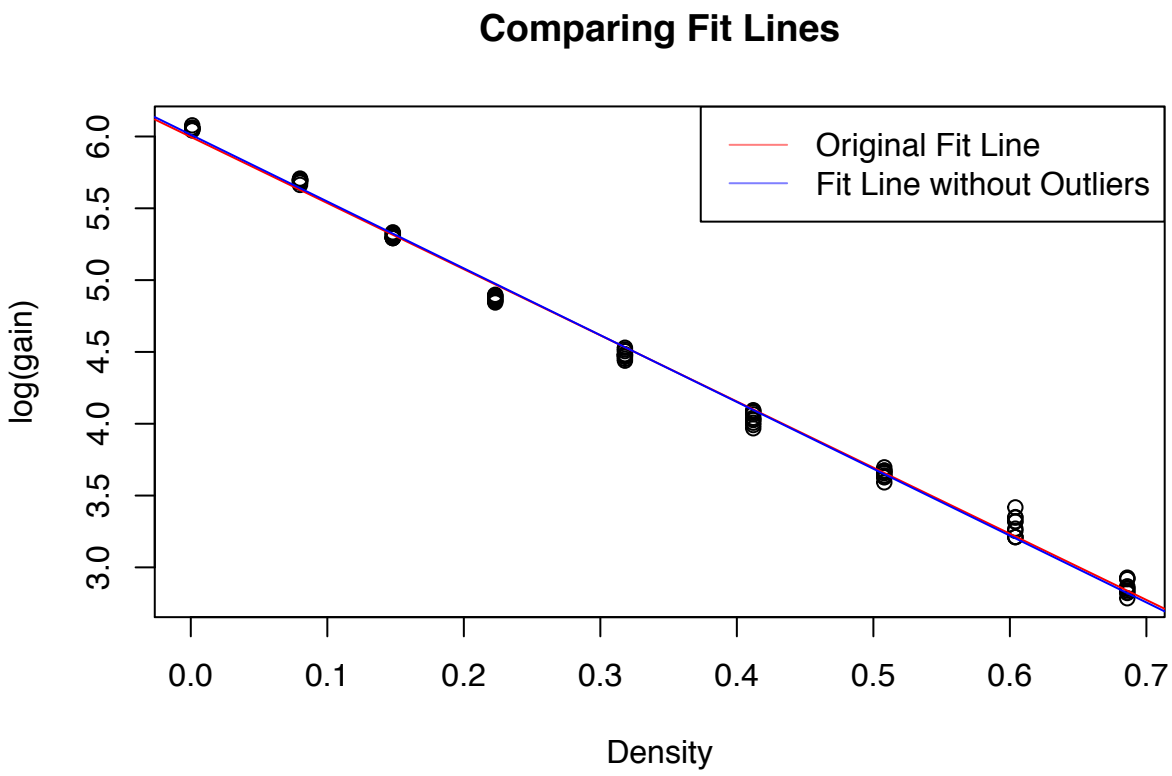

```
plot(fit$residuals);
abline(0, 0, col="red");
```



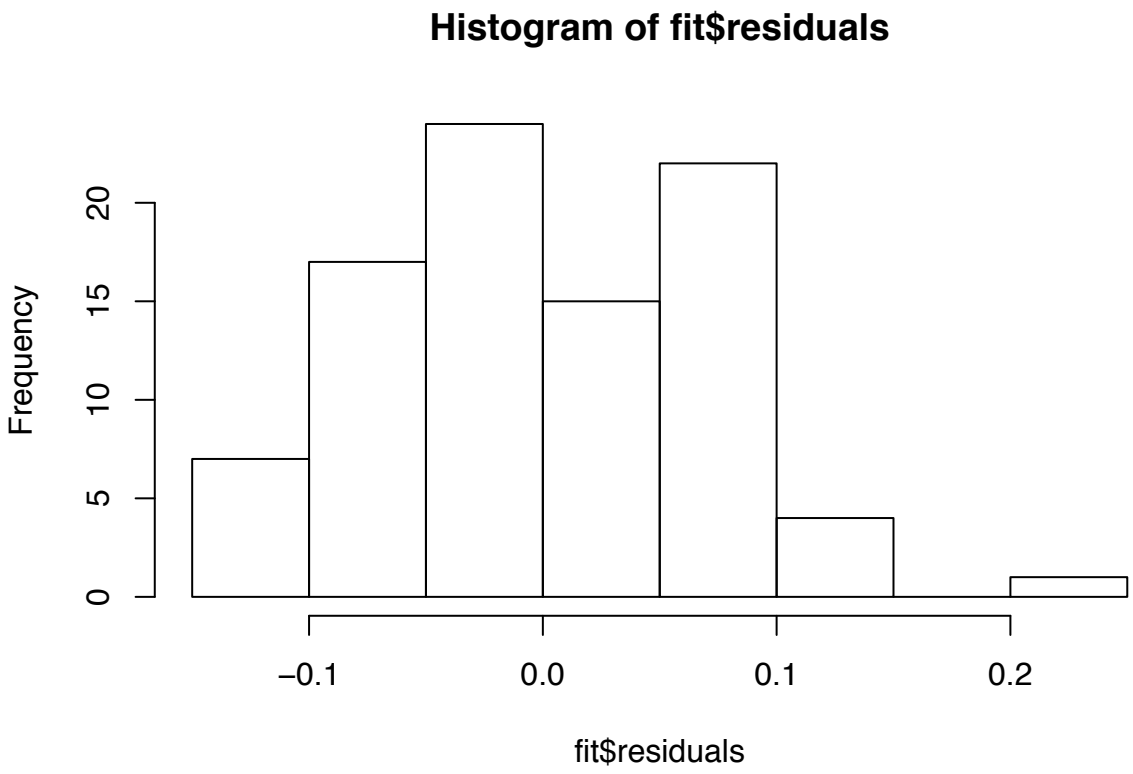

```
qqnorm(fit$residuals);
qqline(fit$residuals, col="red");
```




```
res.rank <- sort(fit$residuals)
suspect <- which(fit$residuals %in% res.rank[0:4])
suspect_high <- which(fit$residuals %in% res.rank[87:90])

fit.out <- lm(formula=log_gain[-suspect][-suspect_high]~density[-suspect][-suspect_high], data=linear_da
plot(linear_data, xlab="Density", ylab="log(gain)",main="Comparing Fit Lines")
abline(fit, col="red")
abline(fit.out, col="blue")
legend("topright", legend = c("Original Fit Line", "Fit Line without Outliers"), lty = c(1,1), col = c(
```




```
hist(fit$residuals);
```




```
R.square <- sum((fit$fitted.values-mean(linear_data$log_gain))^2) / (sum((linear_data$log_gain - mean(li
R.square.out <- sum((fit.out$fitted.values-mean(linear_data$log_gain[-suspect][-suspect_high]))^2) / (su
summary(fit)
```

```
##
## Call:
## lm(formula = log_gain ~ density, data = linear_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.131216 -0.052396 -0.004436  0.054607  0.202447
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.99727    0.01274   470.8   <2e-16 ***
## density     -4.60594    0.03182  -144.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06792 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
```