Case Study 4: Snow Gauge

Justin Glommen
Peter Yao
Atharva Fulay
Alex Hsieh

3/7/2017

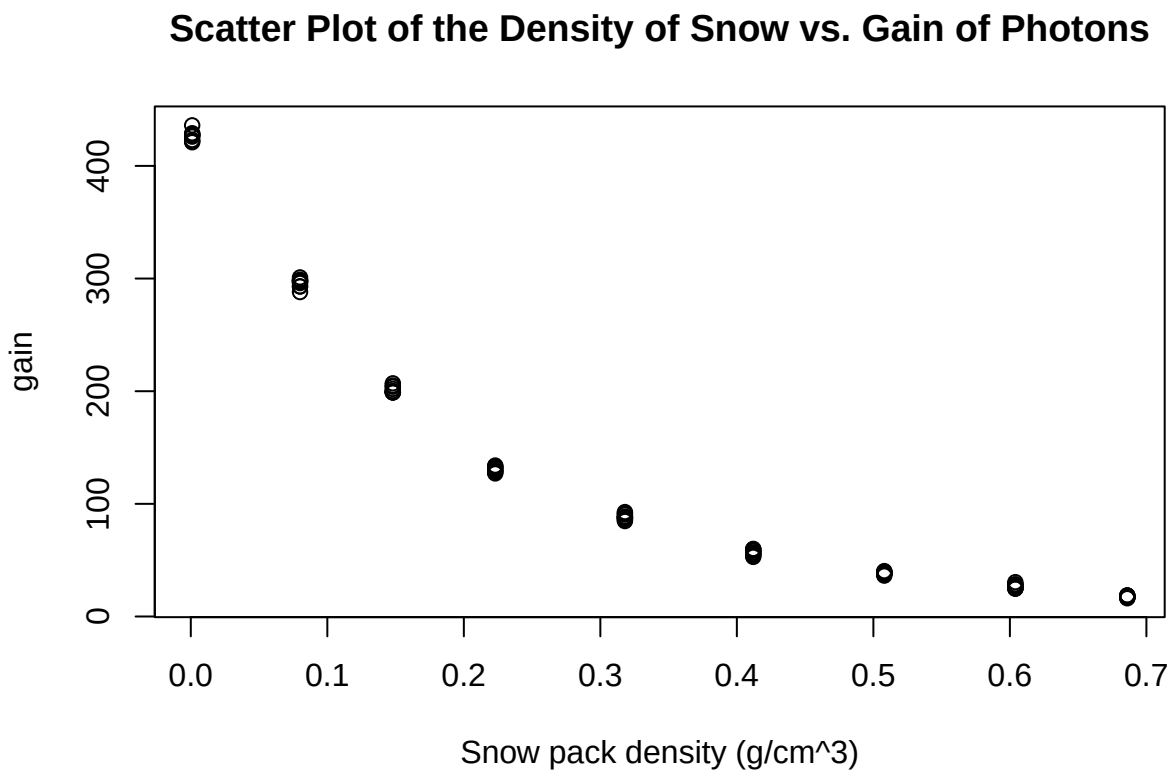
Setup

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
```

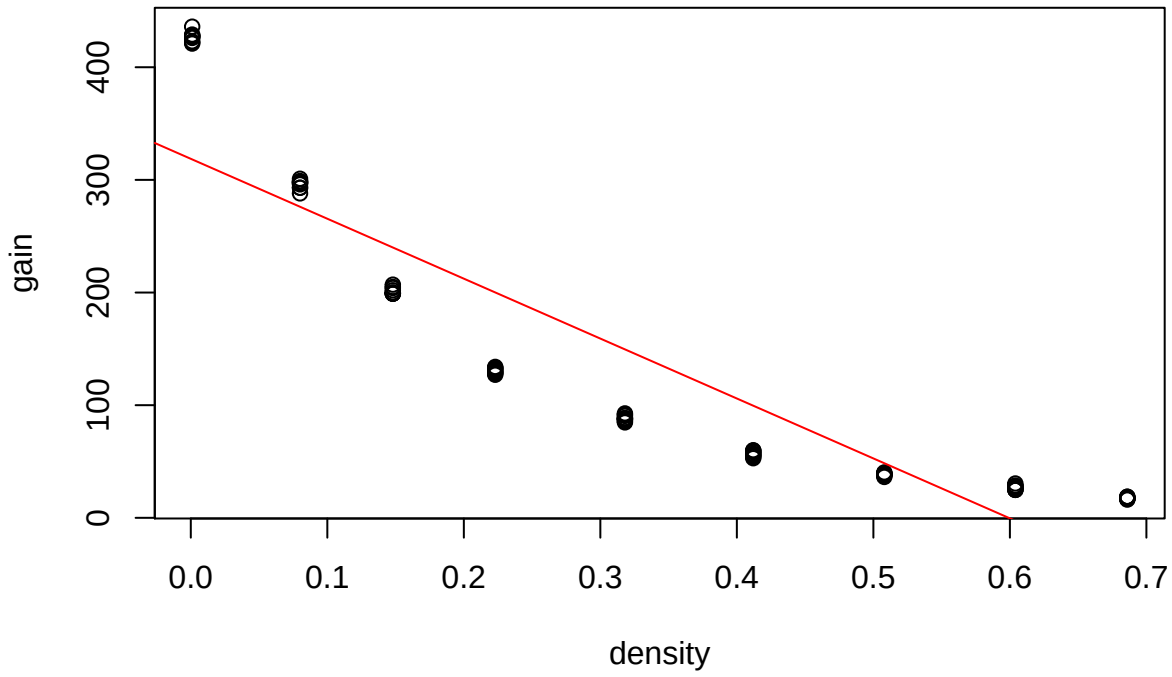
```
stringMain <- "Scatter Plot of the Density of Snow vs. Gain of Photons"
densityLabel <- "Snow pack density (g/cm^3)"
gainLabel <- "gain"
plot(gauge, xlab = densityLabel , ylab = gainLabel, main = stringMain)
```



Fitting

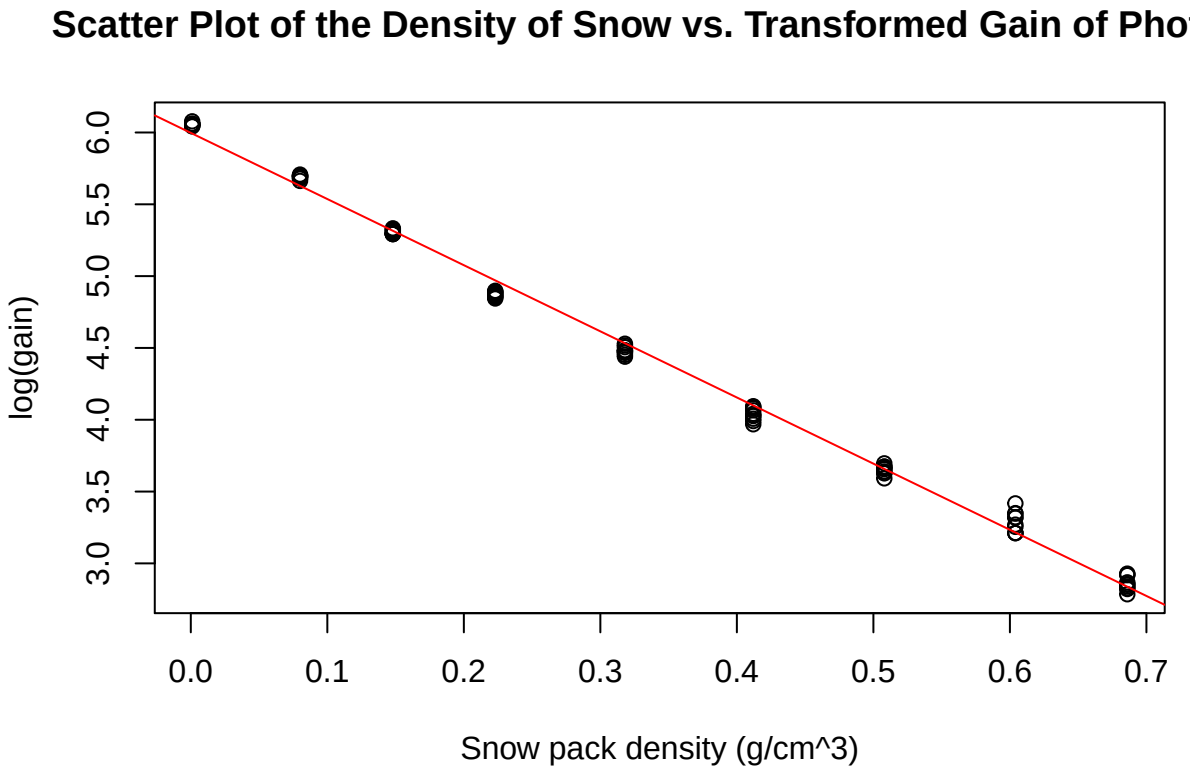
We now want to fit the lease squares line of the original data, such that we can plot and use it to help us make predictions about the snow-pack densities.

```
# 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 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){
  # Log the input
  inputLog <- log(gainInput);
  # Get the bounds for the interval by adding/subbing the constant margin of error
  upper_lim_log <- inputLog + 0.138;
  lower_lim_log <- inputLog - 0.138;
  upper_lim <- exp(upper_lim_log)
  lower_lim <- exp(lower_lim_log)
  # Generate the density output on the line of fit by manipulating
  # the y = mx + b equation of the regression line
  den = (inputLog - 5.997) / -4.606;

  result <- matrix(data=NaN, nrow = 3, ncol = 3)
  result[1,] <- c(lower_lim, gainInput, upper_lim)
  result[2,] <- c(lower_lim_log, inputLog, upper_lim_log);
  result[3,1] <- den
  return(result);
```

```
}


estimating_fn_dens <- function(densityInput){
  estLogGain <- densityInput*(-4.606) + 5.997
  estGain <- exp(estLogGain)
  return(estimating_fn(estGain))
}
```

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.

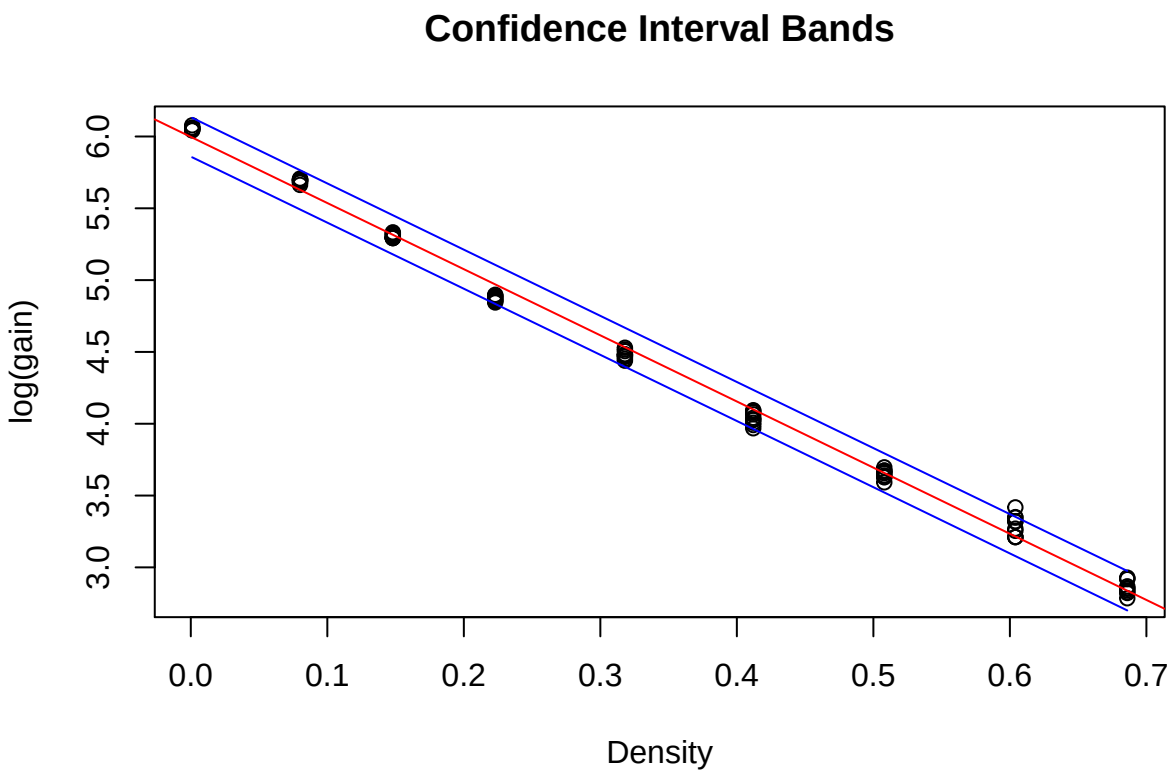
```
# Initializing data differently to avoid conflict with previous data
gain <- data["gain"];
log_gain = log(gain);
data["log_gain"] <- log_gain;
linear_data <- data["density"];
linear_data["log_gain"] <- data["log_gain"];

#----------------------------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]      [,2]      [,3]
## [1,] 33.6244095 38.600000 44.311856
## [2,]  3.5152523  3.653252  3.791252
## [3,]  0.5088467       NaN       NaN
```

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

```
##            [,1]      [,2]      [,3]
## [1,] 33.755791 38.750823 44.484998
## [2,]  3.519152  3.657152  3.795152
## [3,]  0.508000       NaN       NaN
```

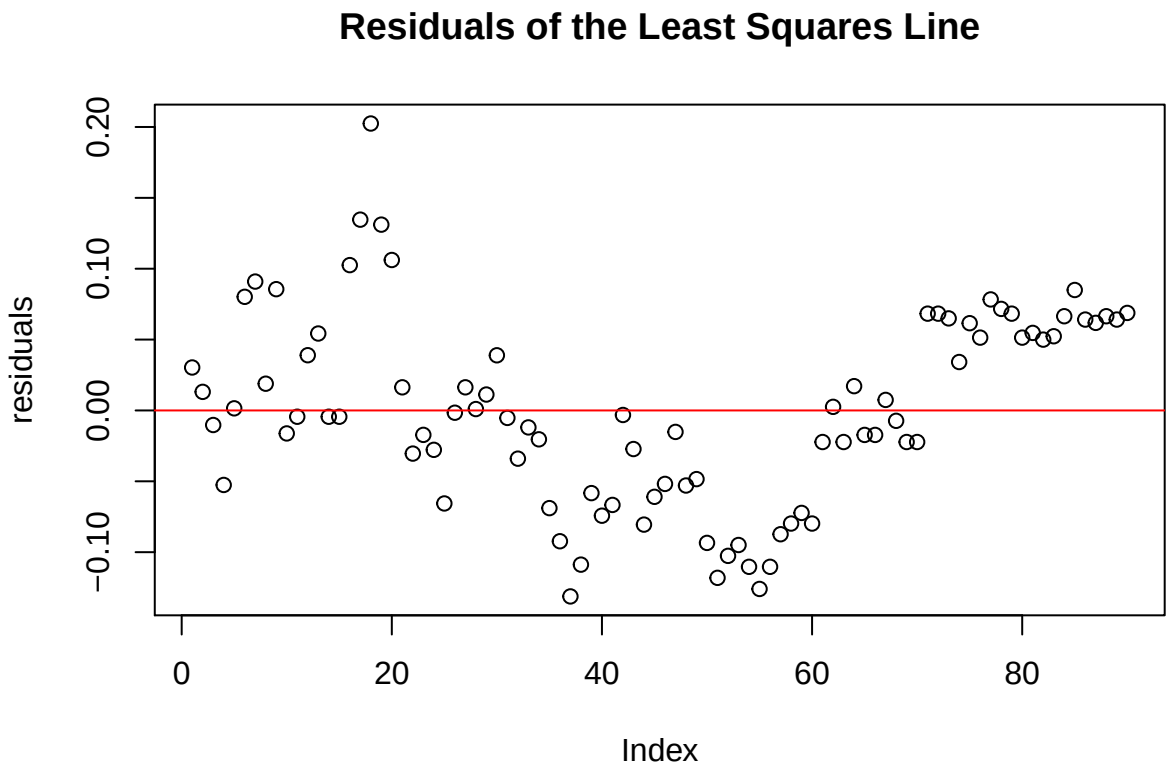
```r
# Should result in a 400 estimate
testDens <- .001
estimating_fn_dens(testDens)
```

```
##            [,1]       [,2]       [,3]
## [1,] 348.763485 400.371954 459.617214
## [2,]   5.854394   5.992394   6.130394
## [3,]   0.001000        NaN        NaN
```

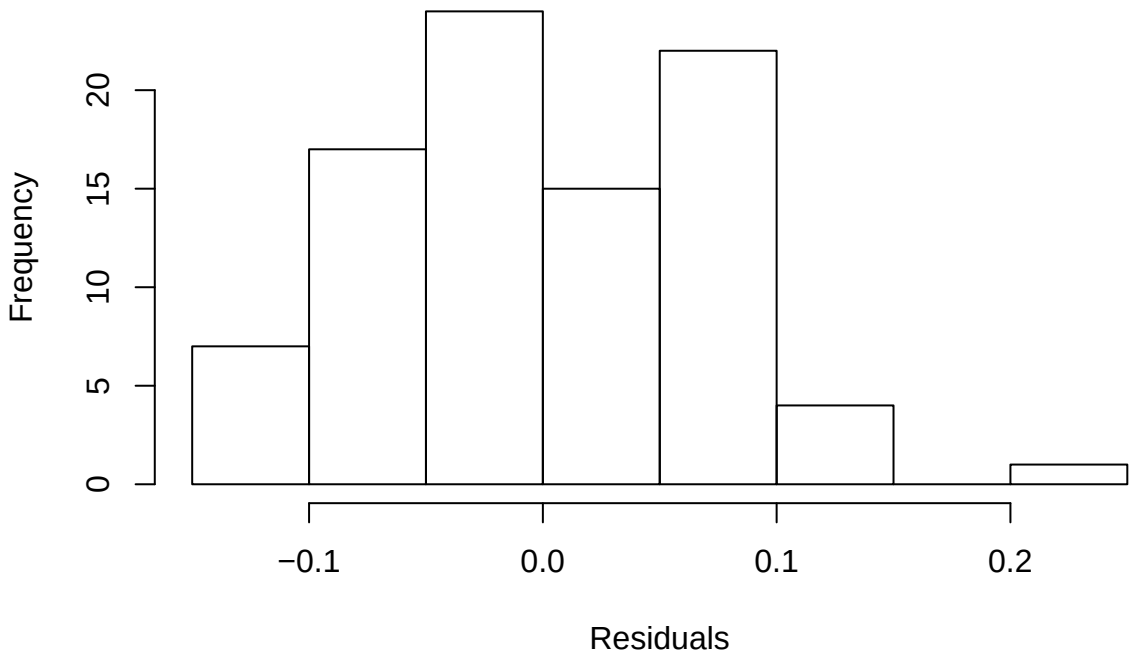
Clearly here, our result is correct and matches that closely to the original dataset.

Extra Analysis

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




```r
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))
```

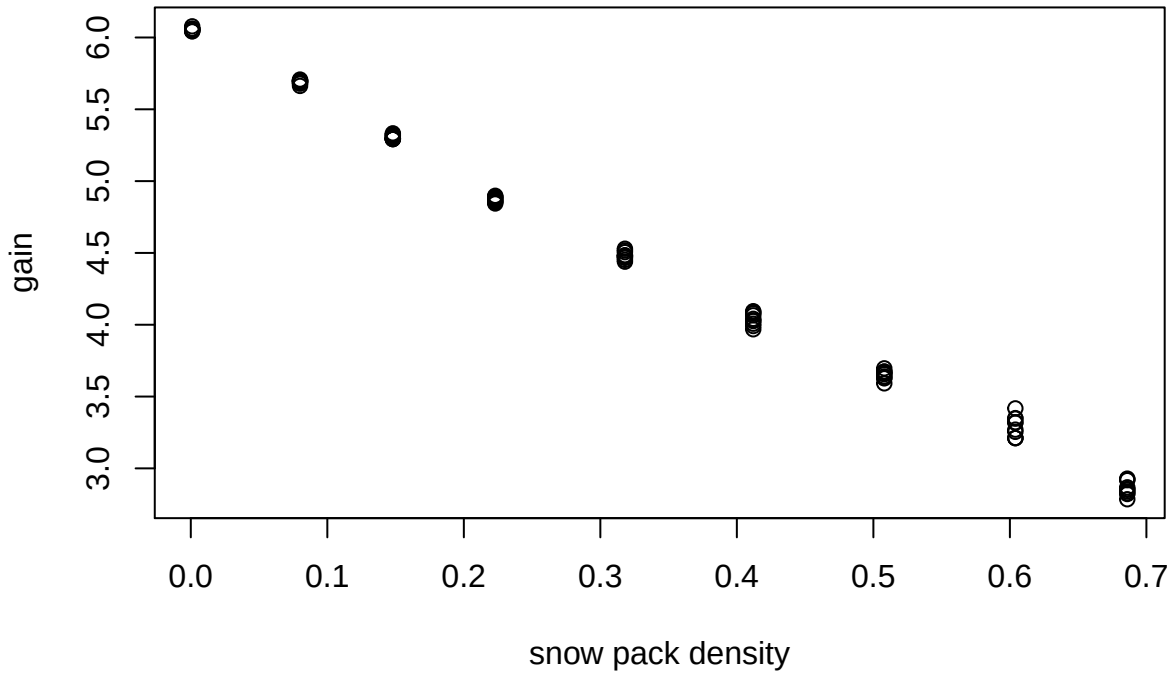
```
##      10%      30%      50%      70%      90%
## 2.927987 3.672239 4.480174 5.293305 6.042870
```

```
#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
```

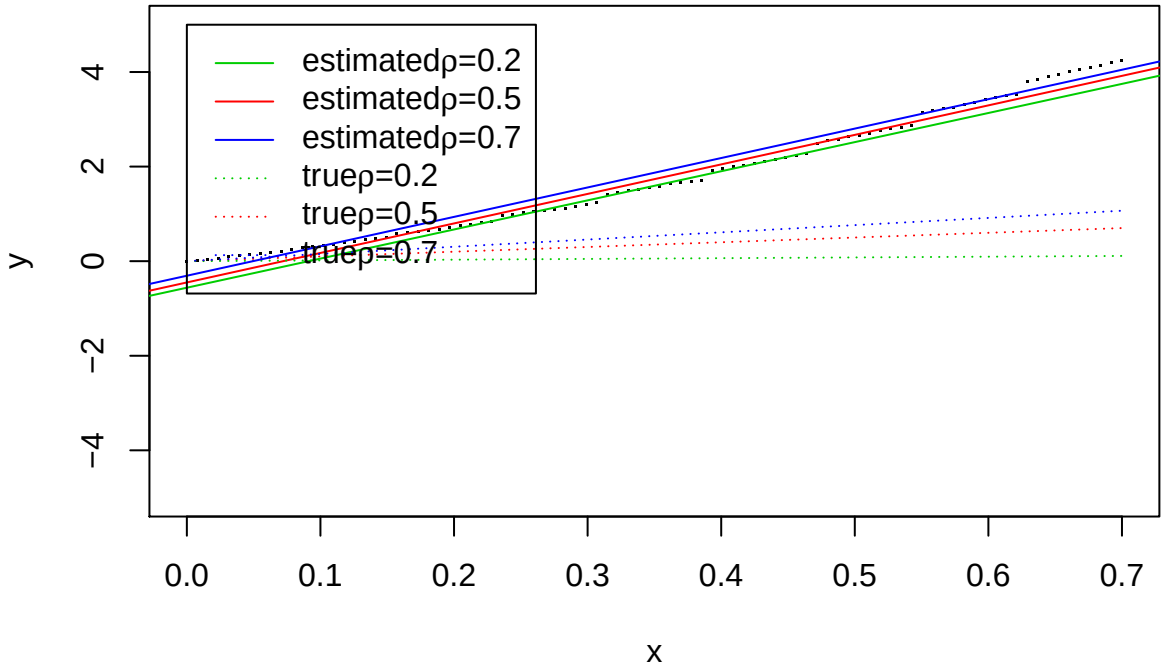
```
##
## Attaching package: 'SparseM'
```

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



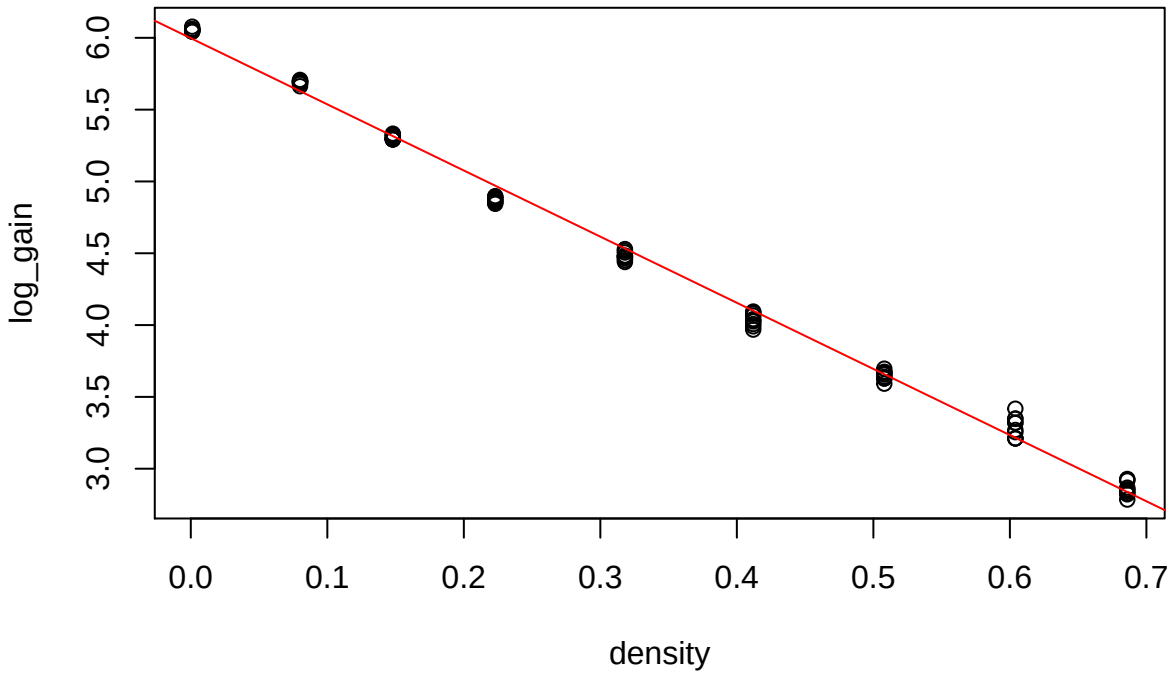

```
x <- seq(0, 0.7, length.out = 90)
y <- x*gauge[["gain"]]
plot(x, y, pch = ".", ylim = c(-5, 5))
# median
fit1 <- rq(y ~ x, tau = 0.5)
abline(fit1, col = 2)

# 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))
```

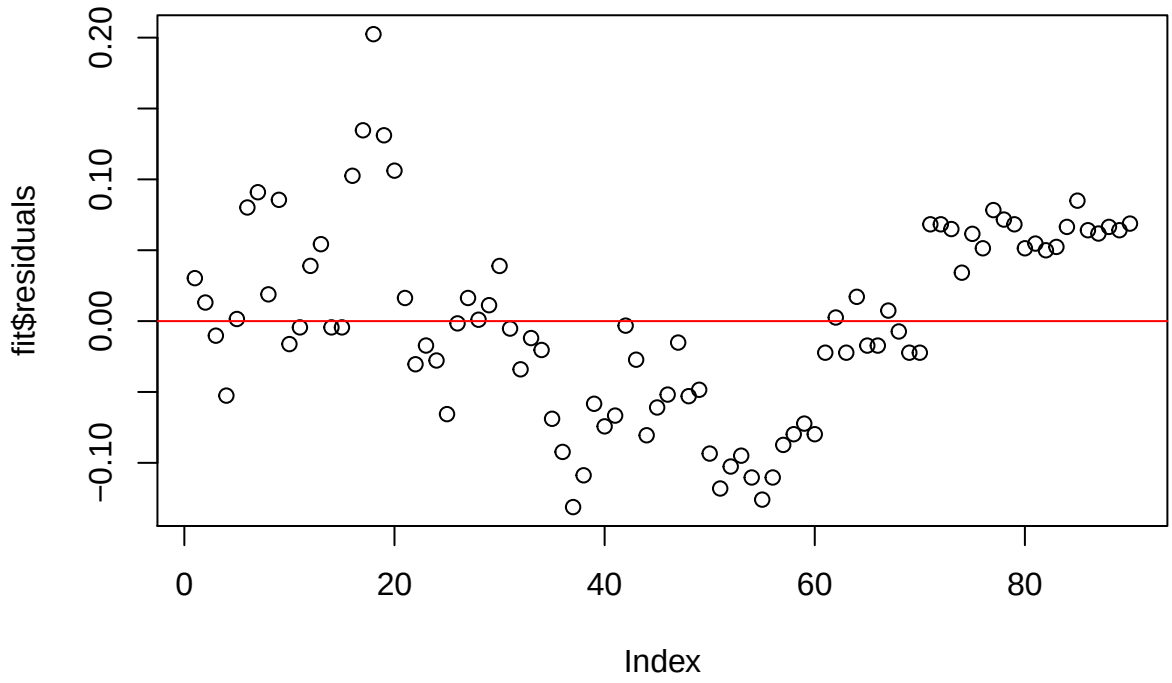


Additional analysis and graphs generated for inspection.

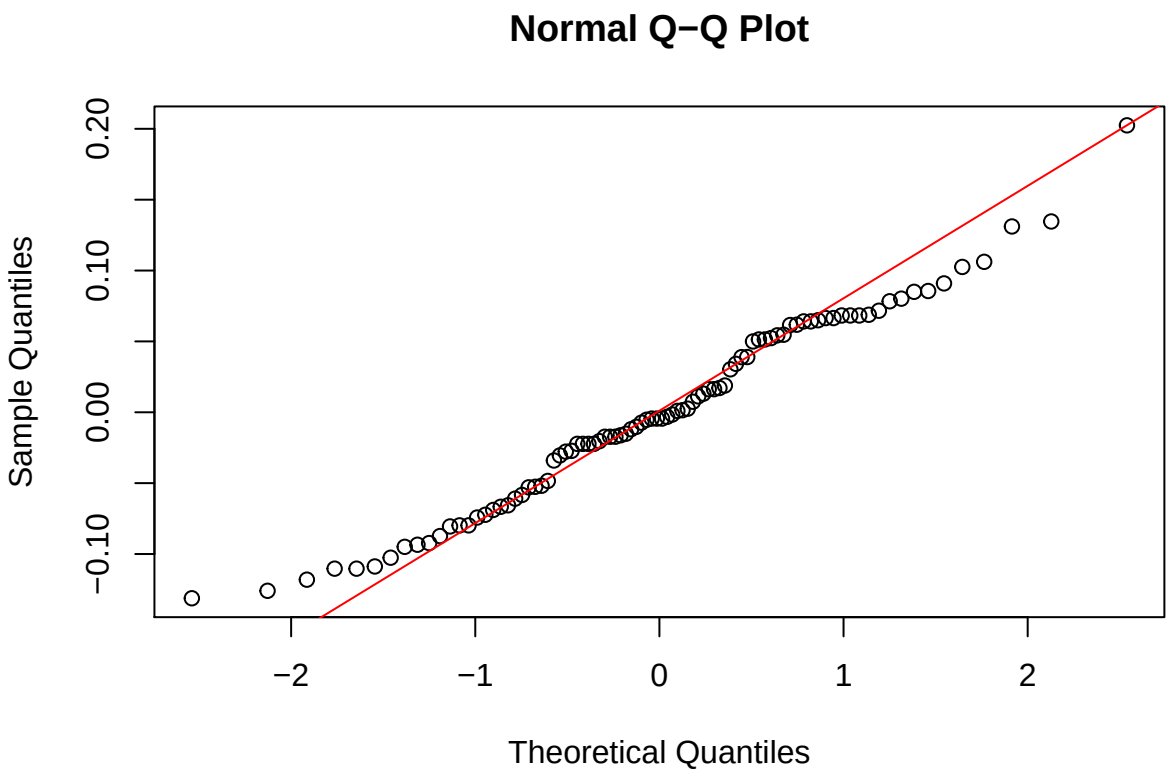
```
fit <- lm(formula=log_gain~density, data=linear_data);

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



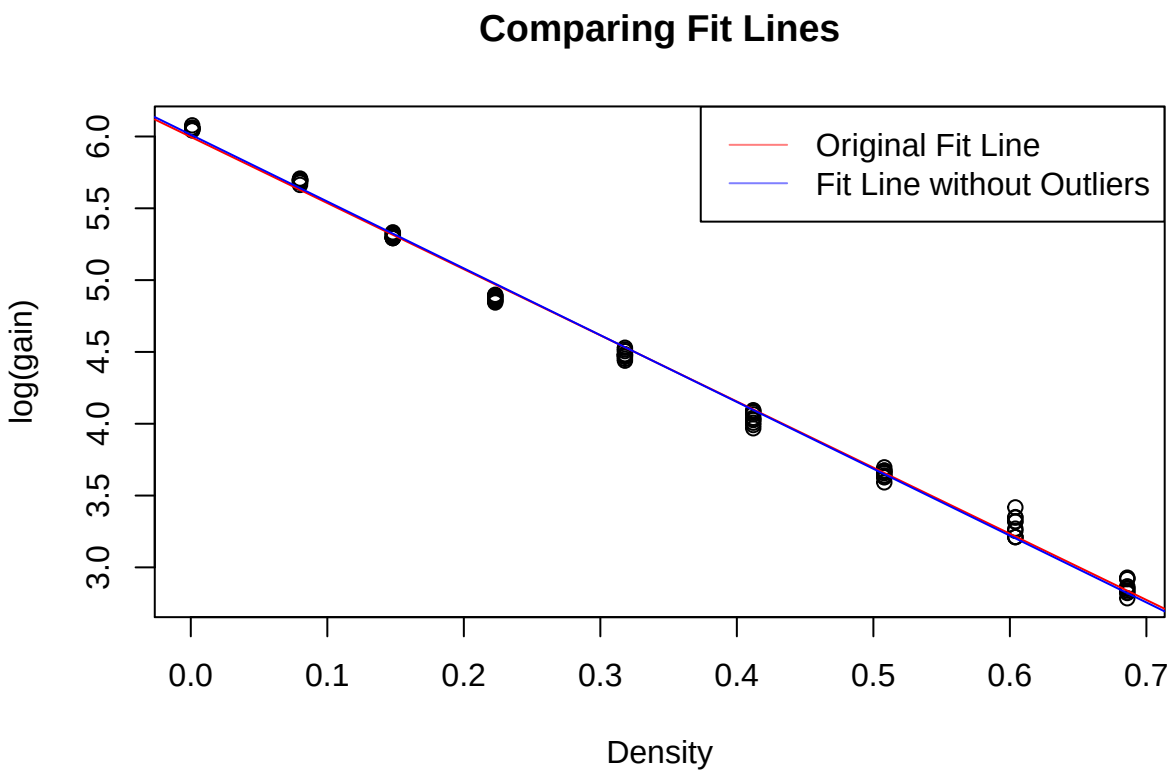

```
plot(fit$residuals);
abline(0, 0, col="red");
```



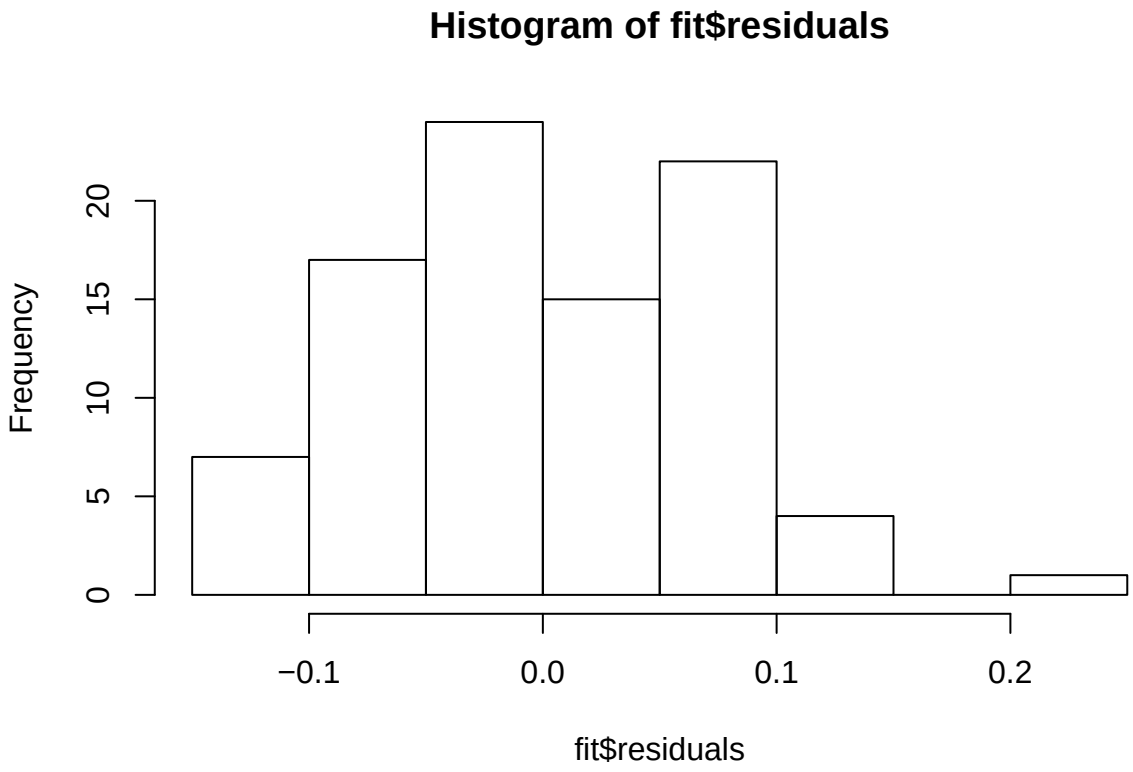

```
qqnorm(fit$residuals);
qqline(fit$residuals, col="red");
```




```
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
```