# MATH 423 - Assignment 2

*Emir Sevinc - 260682995*

*October 24, 2018*

## Question 1

Previously we had found the least square estimator as follows:

$\sum_{i=1}^{n}(y_i - \hat{y_i})^2$ where $\hat{y_i}$ is our "prediction" from the model, and $y_i$ is the actual relation. $\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{n=1}^{n}(y_i - (\hat{\beta_1}x_i))^2$. Differentiating this with respect to $\hat{\beta_1}$ gives $\sum_{i=1}^{n} 2[y_i - \hat{\beta_1}x_i] * -x_i$, so we have $-2\sum_{i=1}^{n}[y_i - \hat{\beta_1}x_i] * x_i = -2\sum_{i=1}^{n}[y_i x_i - \hat{\beta_1}x_i^2] = -2[\sum_{i=1}^{n} y_i x_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2]$, and equating this to 0 will optimize it. So let $-2[\sum_{i=1}^{n} y_i x_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2] = 0 \implies [\sum_{i=1}^{n} y_i x_i - \hat{\beta_1}\sum_{i=1}^{n} x_i^2] = 0 \implies \hat{\beta_1}\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i \implies \hat{\beta_1} = \frac{\sum_{i=1}^{n} y_i x_i}{\sum_{i=1}^{n} x_i^2}$.

Now let's write this as "Constant + Noise".

Since $y_i = \beta_1 x_i + \epsilon_i$ we can rewrite this as $\frac{\sum_{i=1}^{n}(\beta_1 x_i + \epsilon_i)x_i}{\sum_{i=1}^{n} x_i^2}$

$= \frac{\sum_{i=1}^{n}\beta_1 x_i^2 + \sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$

$= \frac{\beta_1\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$

$= \frac{\beta_1\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i^2} + \frac{\sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$

$= \beta_1 + \frac{\sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$.

Now since $\epsilon_i \sim N(0, \sigma^2)$, we have that $\frac{\sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$ will be normally distributed. The mean is found as

$E[x_1\epsilon_1 + ... + x_n\epsilon_n]/\sum_{i=1}^{n} x_i^2$

$= x_1 E[\epsilon_1] + ...x_n E[\epsilon_n]/\sum_{i=1}^{n} x_i^2 = 0.$

The variance is $\frac{Var[x_1\epsilon_1 + ... + x_n\epsilon_n]}{(\sum_{i=1}^{n} x_i^2)^2}$

$= \frac{x_1^2\ Var(\epsilon_1) + ... + x_n^2 Var(\epsilon_n)}{(\sum_{i=1}^{n} x_i^2)^2}$

$= \frac{\sigma^2 x_1^2 + ... + x_n^2\sigma^2}{(\sum_{i=1}^{n} x_i^2)^2}$

$= \frac{\sigma^2\sum_{i=1}^{n} x_i^2}{(\sum_{i=1}^{n} x_i^2)^2}$

$= \frac{\sigma^2}{(\sum_{i=1}^{n} x_i^2)}$.

Thus we have that $\frac{\sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2} \sim N(0, \frac{\sigma^2}{\sum_{i=1}^{n} x_i^2})$

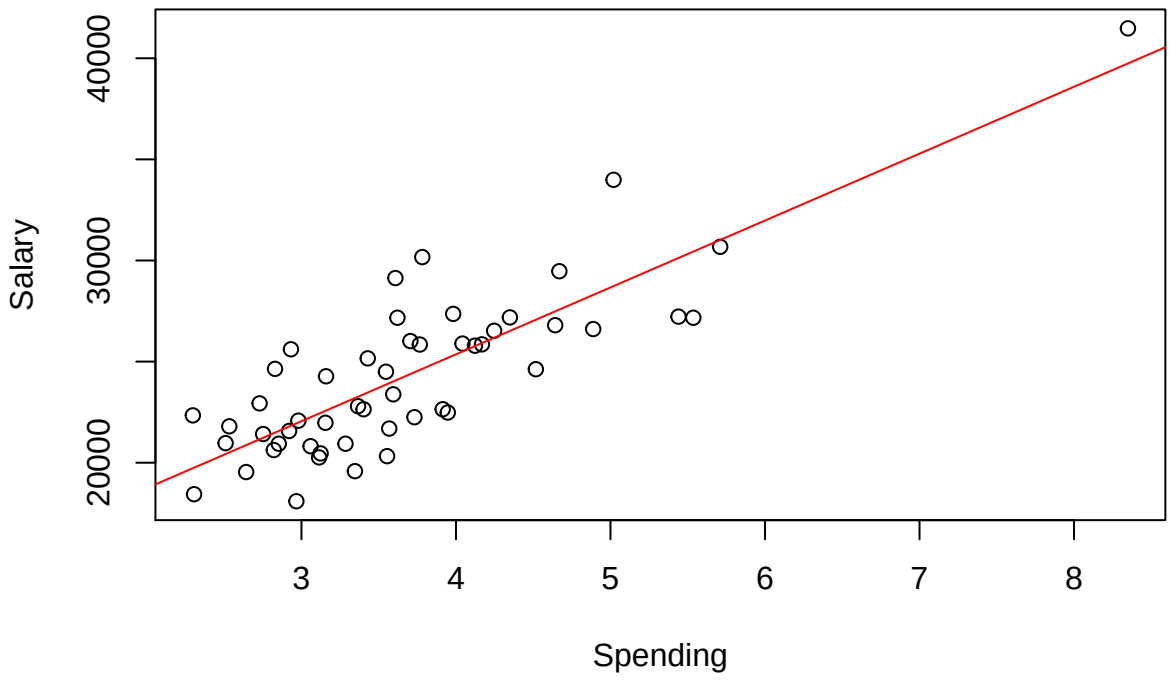
We had already found that $\hat{\beta_1}$ is unbiased and thus has mean $\beta_1$, thus $\hat{\beta_1} \sim N(\beta_1, \frac{\sigma^2}{\sum_{i=1}^{n} x_i^2})$ This shows that the sampling distribution of $\hat{\beta_1}$ is Gaussian, with mean $\beta_1$ and variance $\frac{\sigma^2}{\sum_{i=1}^{n} x_i^2}$.

## Question 2

```
file1 <- "http://www.math.mcgill.ca/yyang/regression/data/salary.csv"
salary <- read.csv(file1, header = TRUE)
x1 <- salary$SPENDING/1000
y <- salary$SALARY
fit.Salary <- lm(y ~ x1)
summary(fit.Salary)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3848.0 -1844.6  -217.5  1660.0  5529.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12129.4     1197.4   10.13 1.31e-13 ***
## x1            3307.6      311.7   10.61 2.71e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2325 on 49 degrees of freedom
## Multiple R-squared:  0.6968, Adjusted R-squared:  0.6906
## F-statistic: 112.6 on 1 and 49 DF,  p-value: 2.707e-14
```

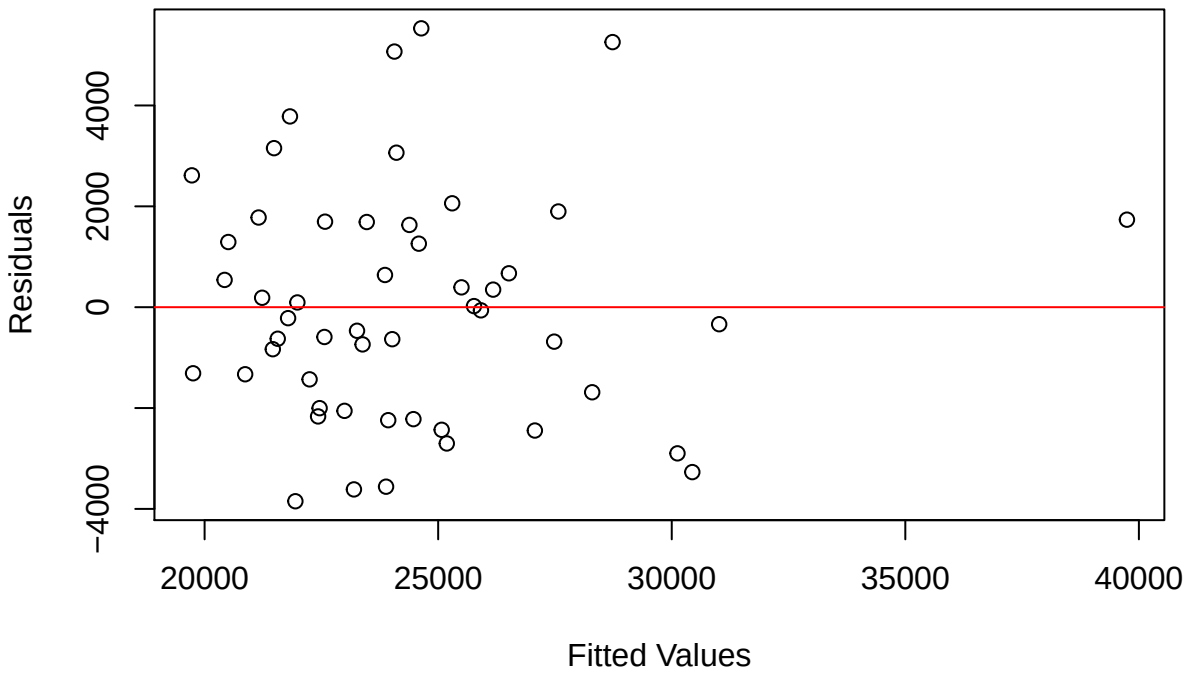
```
plot(x1, y, xlab="Spending", ylab="Salary") #upto here the code is identical;
#we simply renamed the axes and introduced a regression line.
abline(coef(fit.Salary),col='red')
```




**1.**

Here is the residual plot:

```
plot(fitted(fit.Salary), residuals(fit.Salary),
xlab = "Fitted Values", ylab = "Residuals")
abline(h = 0, col = "red")
```

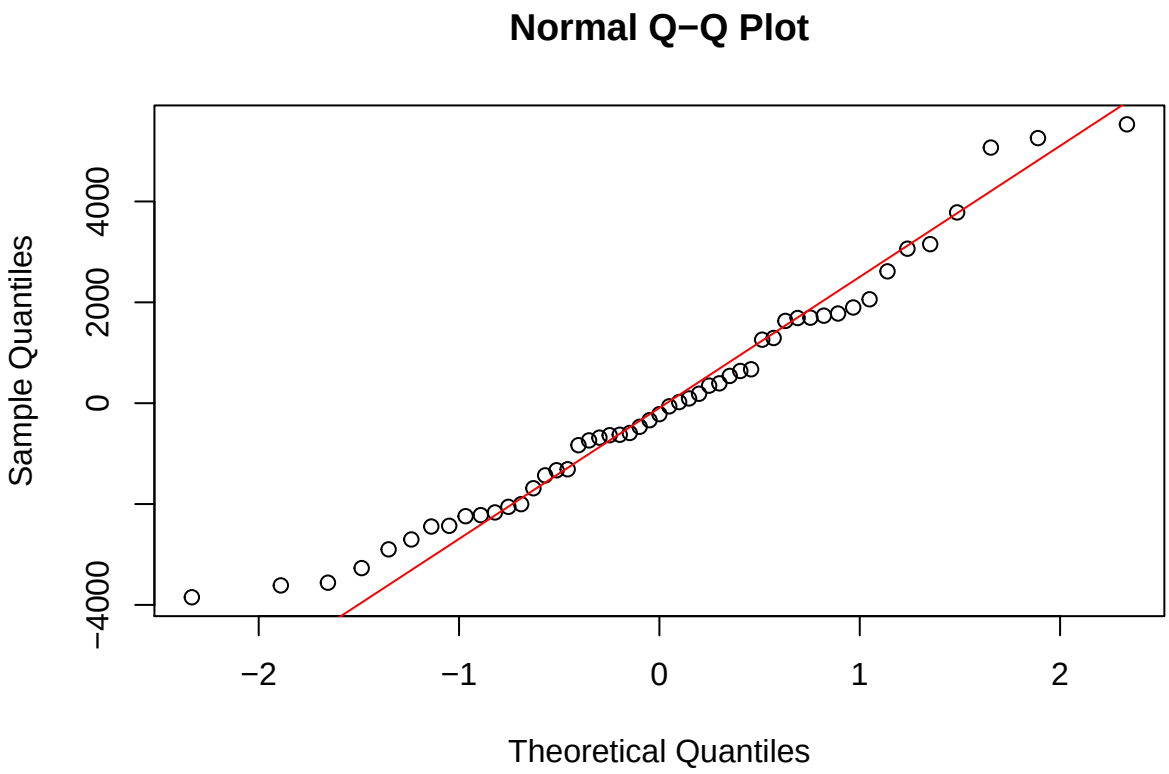



The residuals are broadly centred around 0, we have some high variance outliers but there are no patterns or clusters that suggest that our SLR assumptions are incorrect.

### 2.

Here is the Q-Q plot:

```
qqnorm(residuals(fit.Salary))
qqline(residuals(fit.Salary), col='red')
```




This does look like a good fit. The data points are clustered about our line, and this indicates that the Gaussian assumption does hold.

Some deviations do exist but the residuals do seem to broadly fit the normal distribution.

### 3.

The sums are provided in order:

```
rsd <- residuals(fit.Salary)
sum(rsd) #sum of residuals
```

```
## [1] 4.973799e-12
```

```
dif <- x1-mean(x1)
sum(rsd*dif) #sum of the product of residuals and the difference between x and xbar
```

```
## [1] -2.370015e-11
```

```
ftd <- fitted(fit.Salary)
sum(ftd*rsd) #sum of the product of the residuals and the fitted model
```

```
## [1] 4.086178e-08
```

These numbers are extremely close to 0; the fact that they aren't precisely can be attributed to the roundoff errors since we're using a computer.

### 4.

As one can observe from the output, the value of $\hat{\beta}_1$ is 3307.6, and the value of $\hat{\beta}_0$ is 12129.4.

Since $E[Y/X=0] = E[\beta_1.0] + E[\beta_0] = \beta_0$, the intercept $\beta_0$ is interpreted as $E[Y/X=0]$; what we expect that value of Y to be given that X is equal to 0. Since we estimate $\beta_0$ with $\hat{\beta}_0$, it's interpreted the same way.

Thus in this case the intercept $\hat{\beta}_0 = 12129.4$ is $E[Salary/Spending = 0]$; what we expect the salary to be if spending is equal to 0.

The slope on the other hand is seen as follows: $E[Y/X=x] - E[Y/X=x-1] = x\beta_1 + \beta_0 - ((x-1)\beta_1 + \beta_0) = x\beta_1 + \beta_0 - x\beta_1 + \beta_1 - \beta_0 = \beta_1$. That is, if we select two sets of cases from the un-manipulated distribution where X differs by 1, we expect Y to differ by $\beta_1$. Since $\beta_1$ is estimated as $\hat{\beta}_1$, it is interpreted the same way. So in this specific case, $\hat{\beta}_1$=3307.6 corresponds to the expected change in the salary if spending differs by 1 (so 1000 dollars) in a different sample of the un-manipulated distribution.

## 5.

The null hypothesis is $\beta_1 = 0$ and the alternative hypothesis is $\beta_1 \neq 0$

From the output we see that $T_1^{obs}$, that is $\frac{\hat{\beta}_1}{\hat{se}[\hat{\beta}_1]}$ (the test statistic under the null hypothesis $\beta_1^* = 0$) is equal to 10.61. If $\alpha = 0.05$, we can construct a 1 - 0.05 = 95% confidence interval for $\hat{\beta}_1$ and conduct our hypothesis test that way, since hypothesis testing is equivalent to confidence interval construction.

```
confint(fit.Salary,level=0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 9723.204 14535.538
## x1          2681.192  3933.978
```

Thus the 95% CI for $\beta_1$ is found as (2681.192, 3933.978). 0 is not contained in this intevral, so we can reject the null hypothesis that $\beta_1 = 0$, and conclude that there is a statistically significant linear relationship between SALARY and SPENDING.

the p value is the probability that either T>10.61 or T<-10.61 (on 49 degrees of freedom)which is equivalent to $2P(T > 10.61)$ due to the symmetry of the t distribution. So first we find $P(T > 10.61)$

```
t_stat = pt(10.61, 49, lower.tail=F)
```

So now we find the P value:

```
p_val = (2*(t_stat))
p_val
```

```
## [1] 2.718242e-14
```

Note that the p value was also visible in the output. This number is much, much lower than $\alpha = 0.05$, so it is consistent with our test.

## 6.

The CI is found as:

```
confint(fit.Salary,level=0.9)
```

```
##                   5 %      95 %
## (Intercept) 10121.951 14136.791
## x1           2784.997  3830.173
```

That is for $\hat{\beta}_1$, CI = (2784.997,3830.173) The interpretation is that we are 90% confident that this confidence interval would trap the value of $\beta_1$, and as above since 0 is not captured, the linear relationship between SALARY and SPENDING is statistically significant.

### 7.

The value of $\hat{\beta}_1$ is 3307.6, and the value of $\hat{\beta}_0$ is 12129.4. This we predict the salary as:

```
3307.6*4.8 + 12129.4 #due to the unit difference, 4800 corresponds to 4.8
```

```
## [1] 28005.88
```

Thus we predict it as 28005.88.

## Question 3

```
file2 <- "http://www.math.mcgill.ca/yyang/regression/data/abalone.csv"
abalone <- read.csv(file2, header = TRUE)
r <- abalone$Rings
h <- abalone$Height
```

**Report**

The data attempts to predict the age of abalones from physical measurements. The research group believes that a simple linear regression model with a gaussian error assumption is appropriate to describe the relationship between the height of abalones and their ages, and particularly, that a larger height is associated with an older age.

The two hypotheses to consider here are
1 - Is the simple linear regression model with gaussian error appropriate to describe the relationship between height and age of abalones
2 - Can larger heights be associated with an older age

Let's take a look at individual components.

```
cat("Mean:", mean(h), "
")
```

```
## Mean: 0.1395164
```

```
cat("Range:", range(h), "
")
```

```
## Range: 0 1.13
```

```
cat("Variance:", var(h), "
")
```

```
## Variance: 0.001749503
```

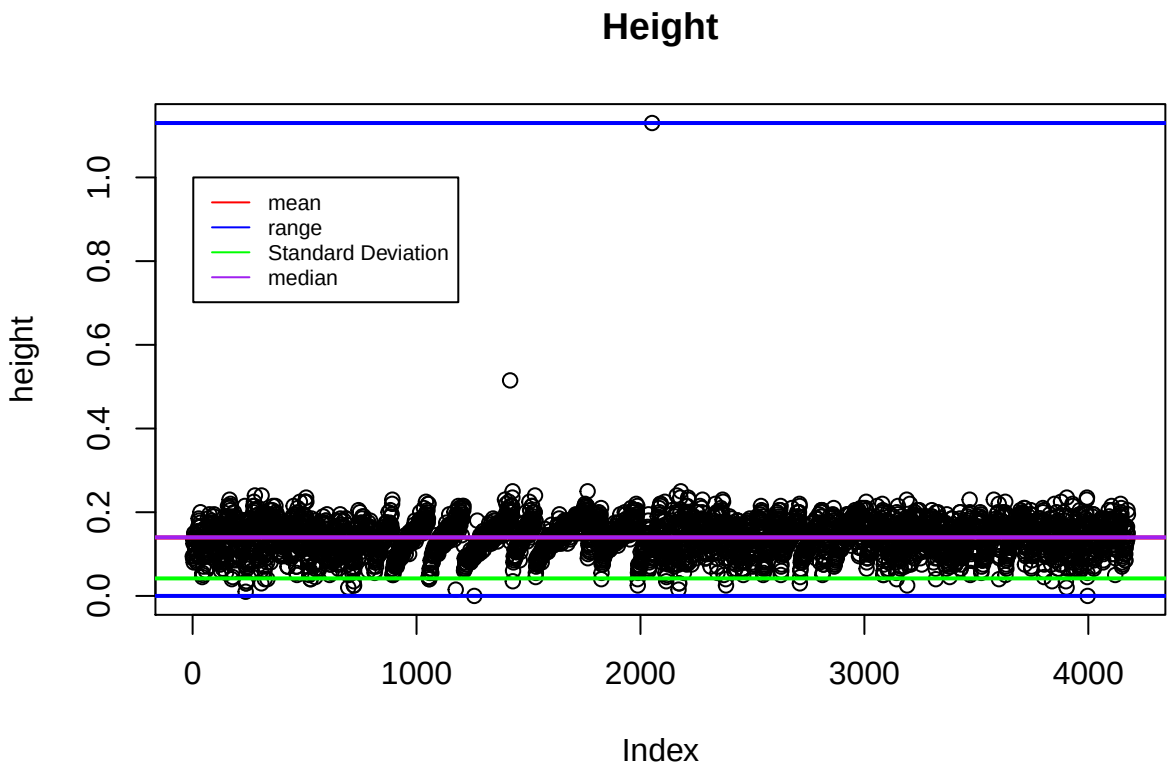
```
cat("Standard Deviation:", sd(h), "
")
```

```
## Standard Deviation: 0.04182706
```

```
cat("Median:", median(h), "
")
```

```
## Median: 0.14
```

```
plot(h, main="Height", ylab = "height")
abline(h=mean(h),lty=1, col = 'red',lwd = 2)
abline(h=range(h),lty=1, col = 'blue',lwd = 2)
abline(h=sd(h),lty=1, col = 'green',lwd = 2)
abline(h=median(h),lty=1, col = 'purple',lwd = 2)
abline(h=mode(h),lty=1, col = 'purple',lwd = 2)
legend(3,1,legend=c("mean", "range","Standard Deviation","median"),
       col=c("red","blue","green","purple"), lty=1:1, cex=0.65)
```




Above we have the height of the observed abalones as a scatterplot. The measures have been added and labeled appropriately.

Note that the unit of height is mm (https://archive.ics.uci.edu/ml/datasets/abalone). Few things to note here:
The range is quite high, however as we can see from the plot, this appears to be caused by one particularly large specimen; an outlier in the context of these observations. The lowers height example on the other hand appears to have height 0, which does not make sense in this context, and this is possibly a measurement/recording error. Another noteworthy aspect is as one can observe from the output, the mean is extremely close to the median, so much so that the lines on the plot overlap completely. This suggests that the distribution is somewhat even and not overwhelmingly skewed towards one direction, which one can observe from the graph. The standard deviation is quite low; so the data is broadly centred around its mean.

Now we shall conduct the sama analysis for the number of rings.

```
cat("Mean:", mean(r), "
")
```

```
## Mean: 9.933684
```

```
cat("Range:", range(r), "
")
```

```
## Range: 1 29
```

```
cat("Variance:", var(r), "
")
```

```
## Variance: 10.39527
```

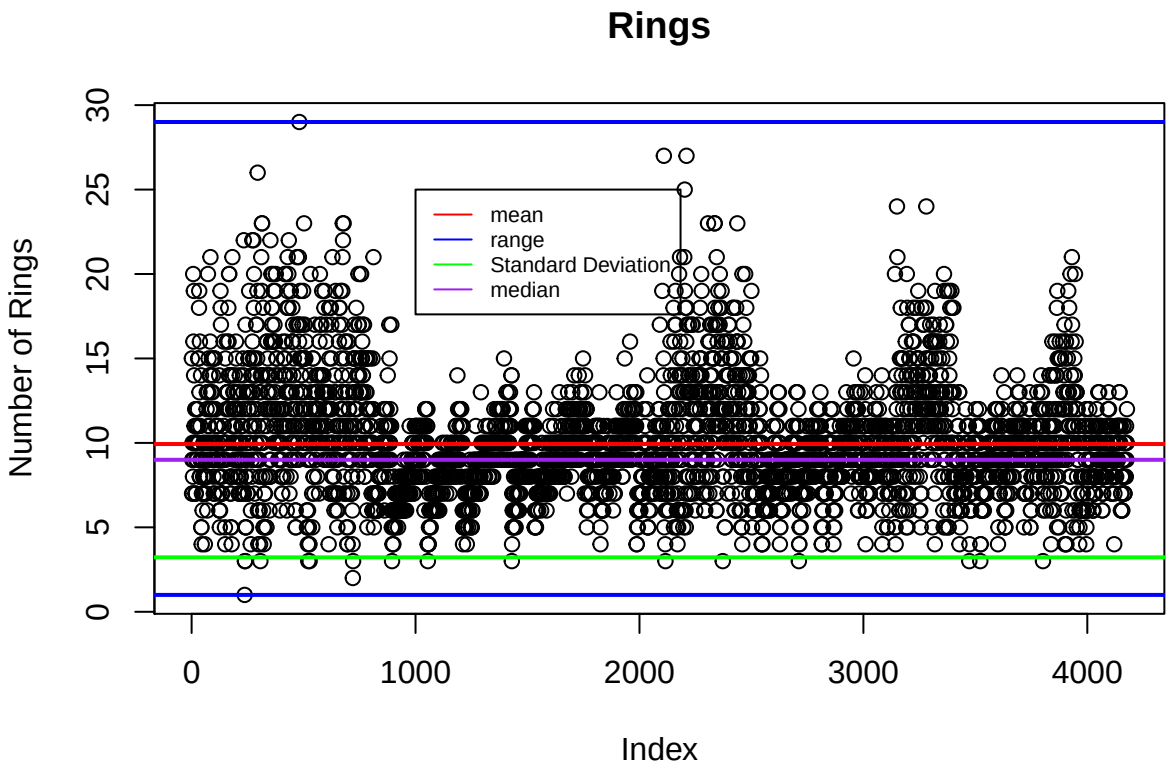
```
cat("Standard Deviation:", sd(r), "
")
```

```
## Standard Deviation: 3.224169
```

```
cat("Median:", median(r), "
")
```

```
## Median: 9
```

```
plot(r, main="Rings", ylab = "Number of Rings")
abline(h=mean(r),lty=1, col = 'red',lwd = 2)
abline(h=range(r),lty=1, col = 'blue',lwd = 2)
abline(h=sd(r),lty=1, col = 'green',lwd = 2)
abline(h=median(r),lty=1, col = 'purple',lwd = 2)
abline(h=mode(r),lty=1, col = 'purple',lwd = 2)
legend(1000,25,legend=c("mean", "range","Standard Deviation","median"),
       col=c("red","blue","green","purple"), lty=1:1, cex=0.65)
```



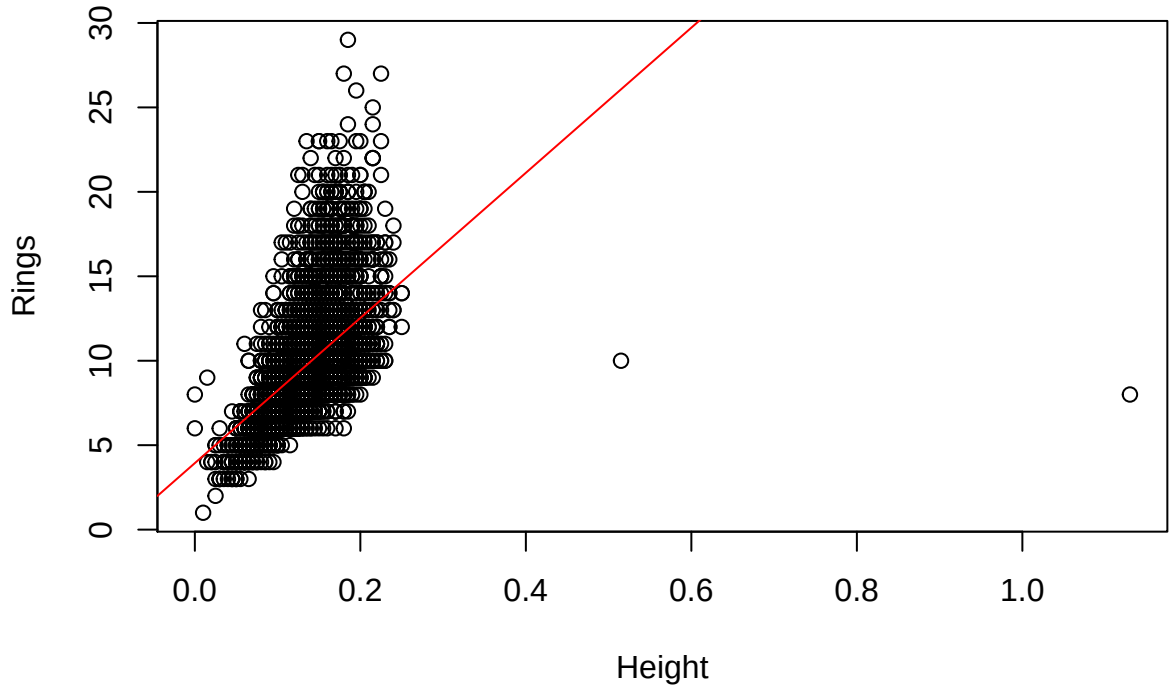
This time the range is somewhat more indicative, however as we go up or down the points get more scarce, so the ring data is also broadly centred about the mean. The median is slightly less than the mean this time. The standard deviation on the other hand is quite high, thus while the data is centred about the mean, it is considerably less clustered around it than the height.

Now let us observe the height vs ring, and attempt to fit a linear regression line. Since we are interested in height measurement to predict their ages, as stated by the study, we take height as our predictor (X) and the ring count as our response (Y).

Below we have the scatterplot of our data and we attempt to fit a regression line to it as well.

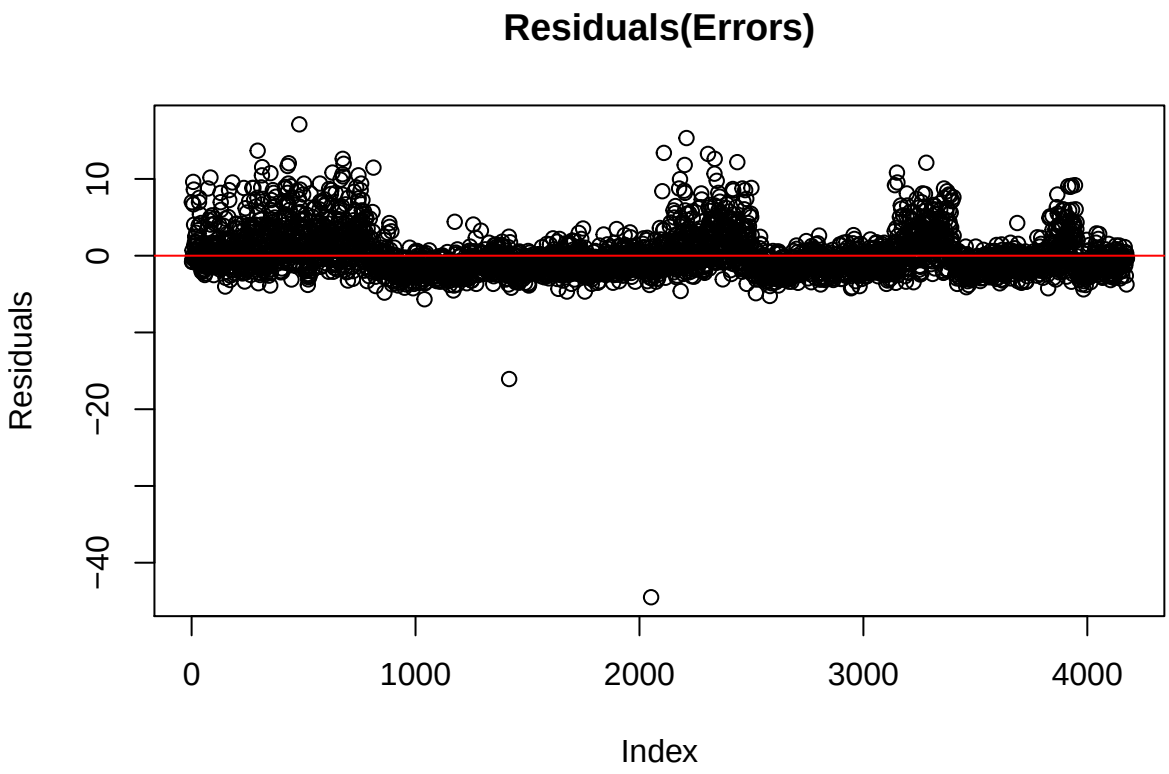
```
fit.Abalone_1 <- lm(r ~ h)


plot(h, r, xlab="Height", ylab="Rings")
abline(coef(fit.Abalone_1),col='red')
```




There does seem to be a linear trend, however overall it looks not to be a perfect fit. We now run diagnostics in order to observe the "goodness" of the fit. We start off with a plot of the residuals (errors)

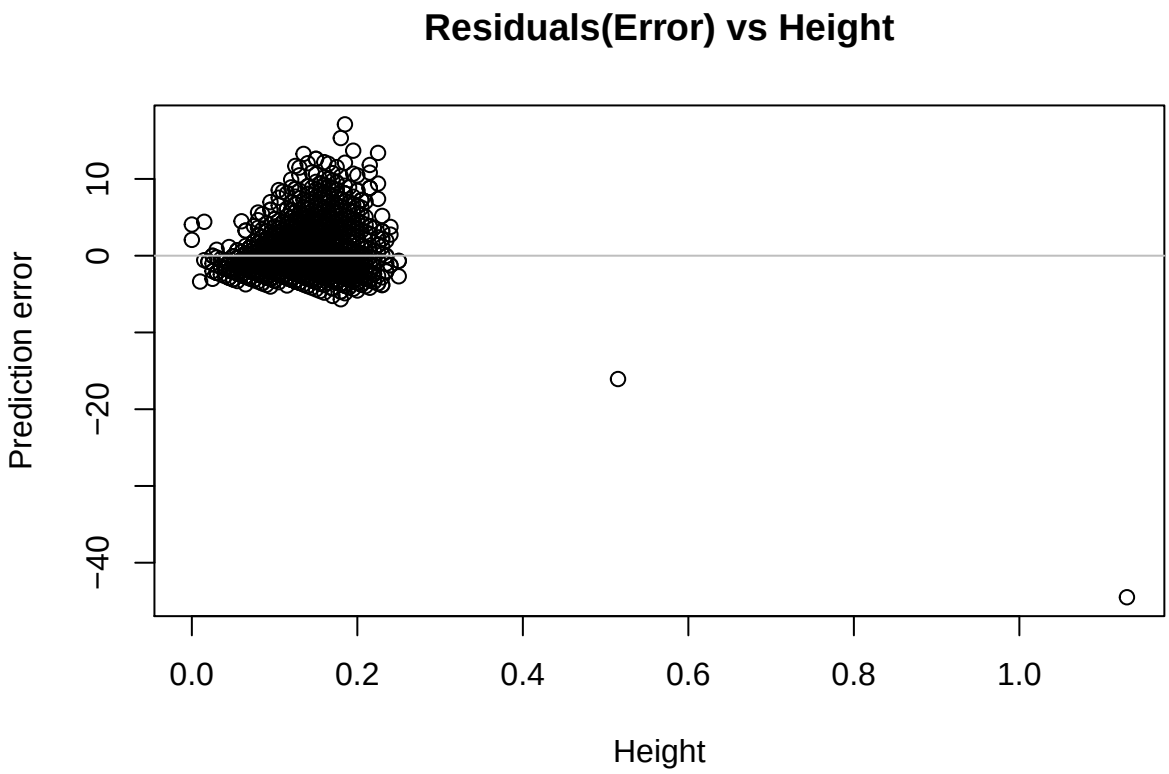
```
abalone_res_1 = residuals(fit.Abalone_1)
plot(abalone_res_1, ylab ="Residuals")
title(main = 'Residuals(Errors)')
abline(h=0,col='red',lty=1)
```




Here we do have a residual plot that is broadly centred around 0, however there is rather high variance and some clustering on certain parts.

We will now visualise the residuals vs the "Predictor (X)" value, Height, that is what we observe and presume to be the influencing factor on the "Response" value, Y(Rings).

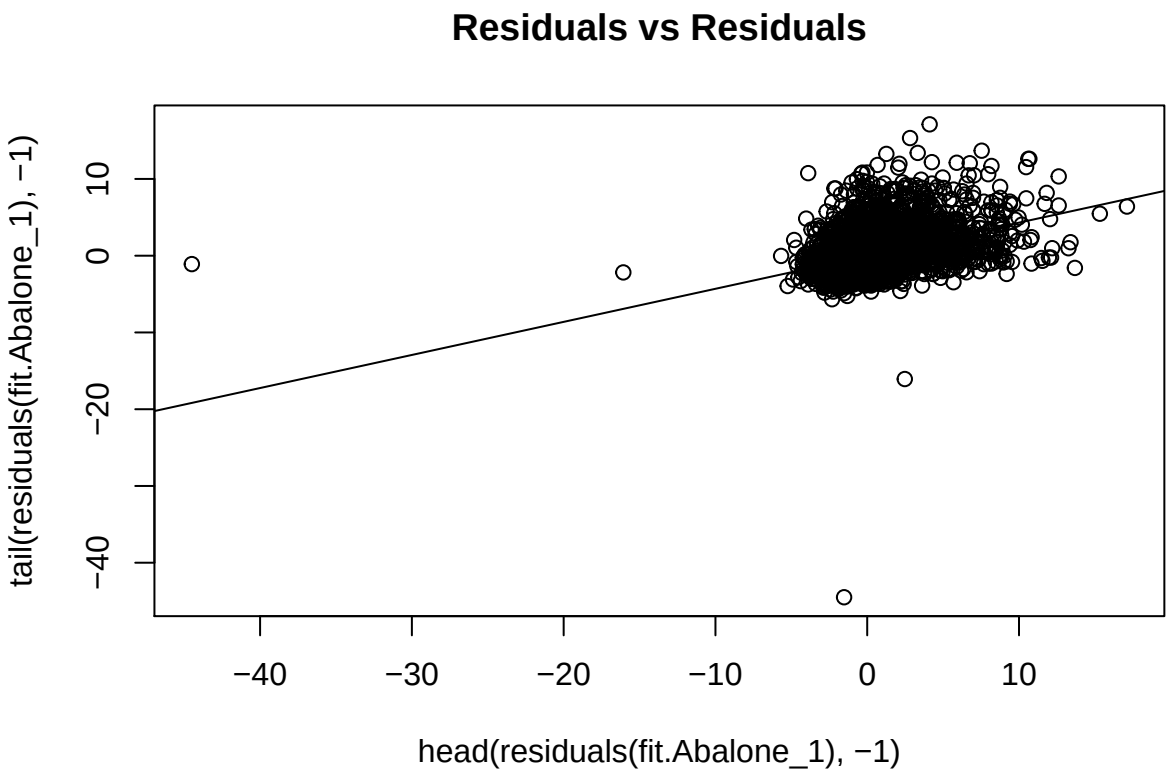
```
plot(h, residuals(fit.Abalone_1), main = "Residuals(Error) vs Height",
xlab = "Height", ylab = "Prediction error")
abline(h = 0, col = "grey")
```




Here, ideally the plot should be a constant blur around zero, and we do appear to have that. We do have somewhat high variance on the residuals however.

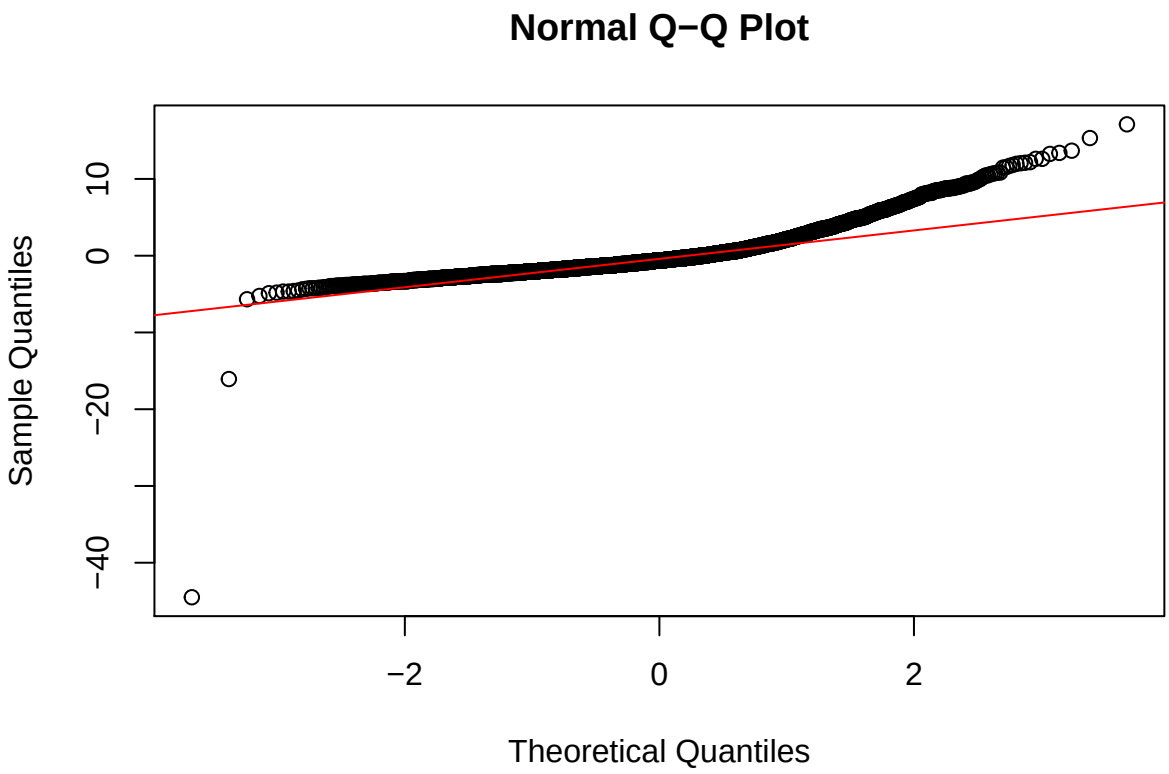
Now we plot the residuals vs residuals.

```
plot(head(residuals(fit.Abalone_1), -1),
tail(residuals(fit.Abalone_1), -1),main = "Residuals vs Residuals")
abline(lm(tail(residuals(fit.Abalone_1),
-1) ~ head(residuals(fit.Abalone_1),
-1)))
```



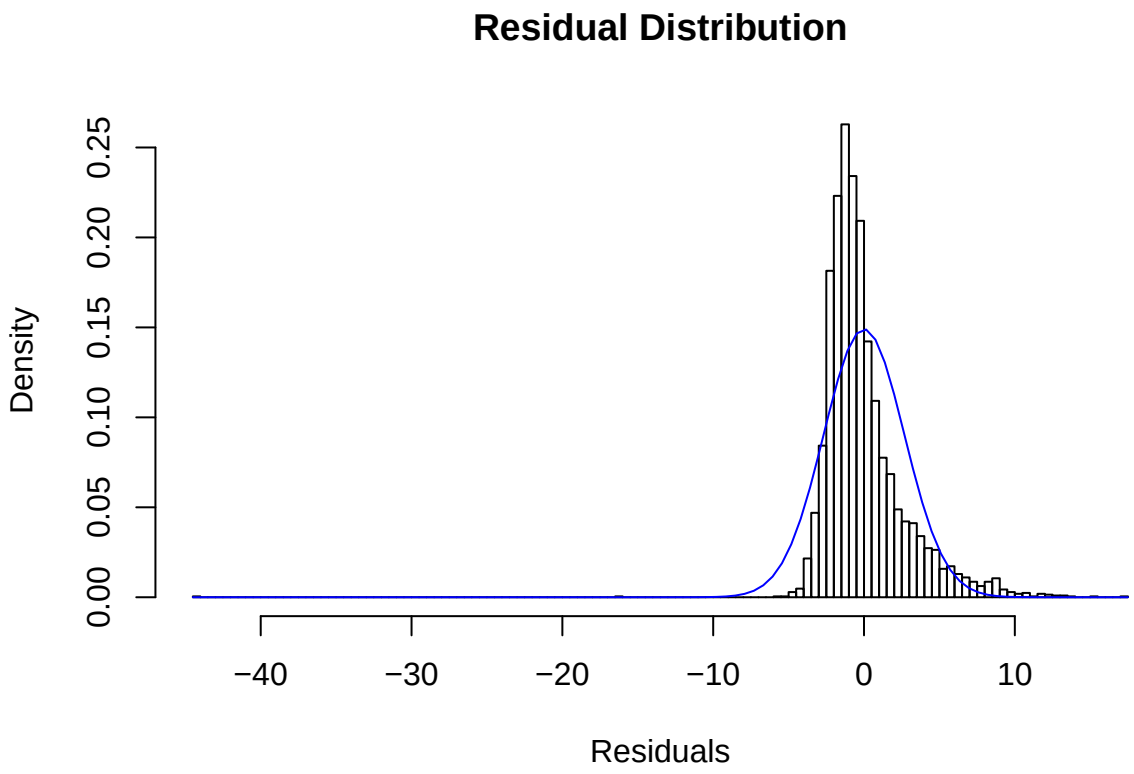

Here ideally we would have a blob with no particular structure, which we do appear to, however we have some potentially suspicious dispersion towards the end. Finally we consider the distribution of the residuals:

```
qqnorm(residuals(fit.Abalone_1))
qqline(residuals(fit.Abalone_1), col='red')
```




We have a decent looking fit for the QQ plot of the residuals indicating that they could be Gaussian; we do have significant divergence towards the end however.

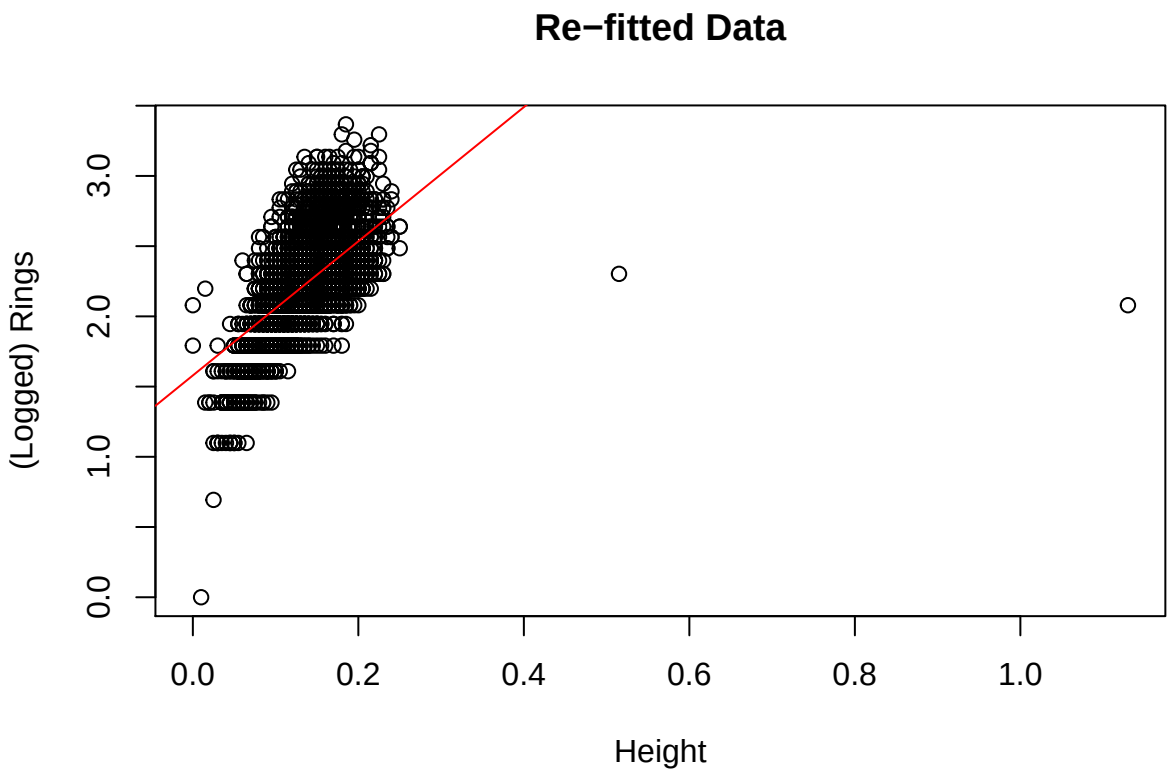
```
hist(residuals(fit.Abalone_1), breaks = 100,
freq = FALSE, xlab = "Residuals",
main = "Residual Distribution")
curve(dnorm(x, mean = 0, sd = sd(residuals(fit.Abalone_1))),
add = TRUE, col = "blue")
```




Above is the residuals plotted against a typical Gaussian distribution that fits its mean an sd. The residuals appear to broadly fit the Gaussian distribution, not perfectly well however and there is some skewing of the data.

We now attenpt to refit the data. The range on the rings was quite high some "exponential" pattern is visible on the initial scatterplot, so we take the natural lograrithm of the rings values, and see what happens

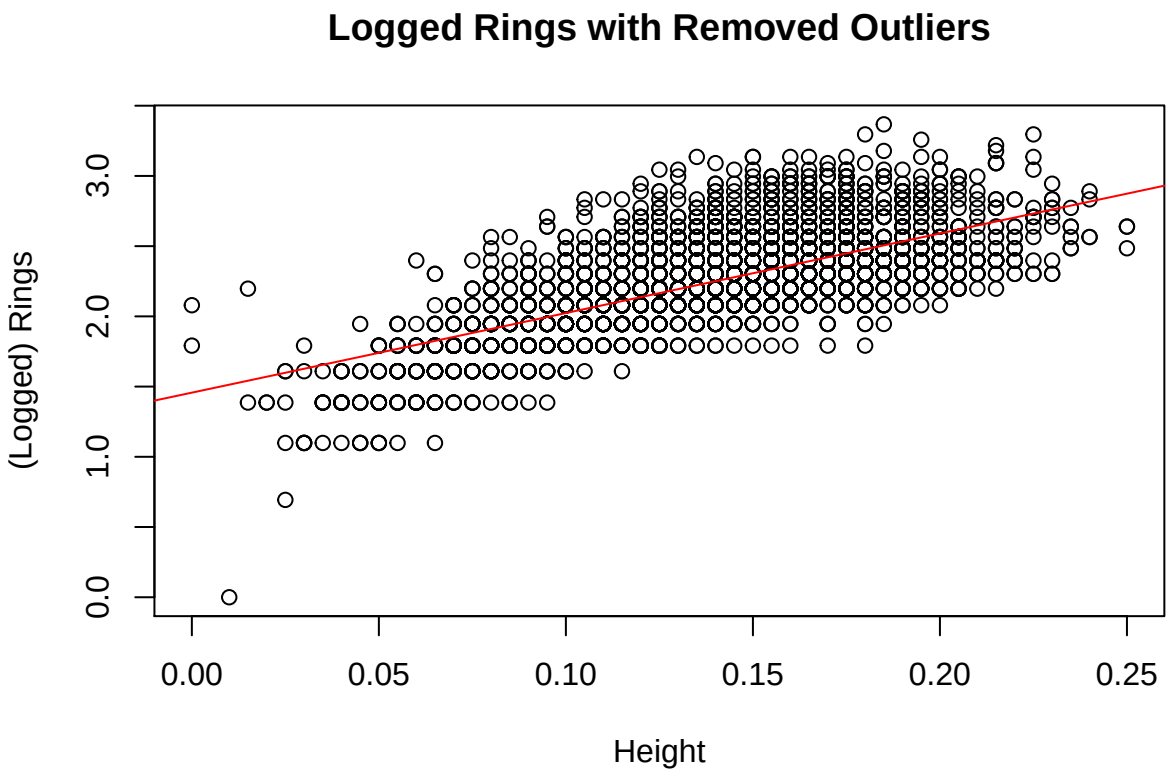
```
lnr <- log(r)
fit.Abalone_refit <- lm(lnr ~ h)
plot(h, lnr, xlab="Height", ylab="(Logged) Rings", main ="Re-fitted Data")
abline(coef(fit.Abalone_refit),col='red')
```




The outlier heights are causing it to not look like a significant improvement, but let us (temporarily; we will keep the outliers in when we conduct the rest of our analysis) remove them to take a closer look at the potential linear trend.

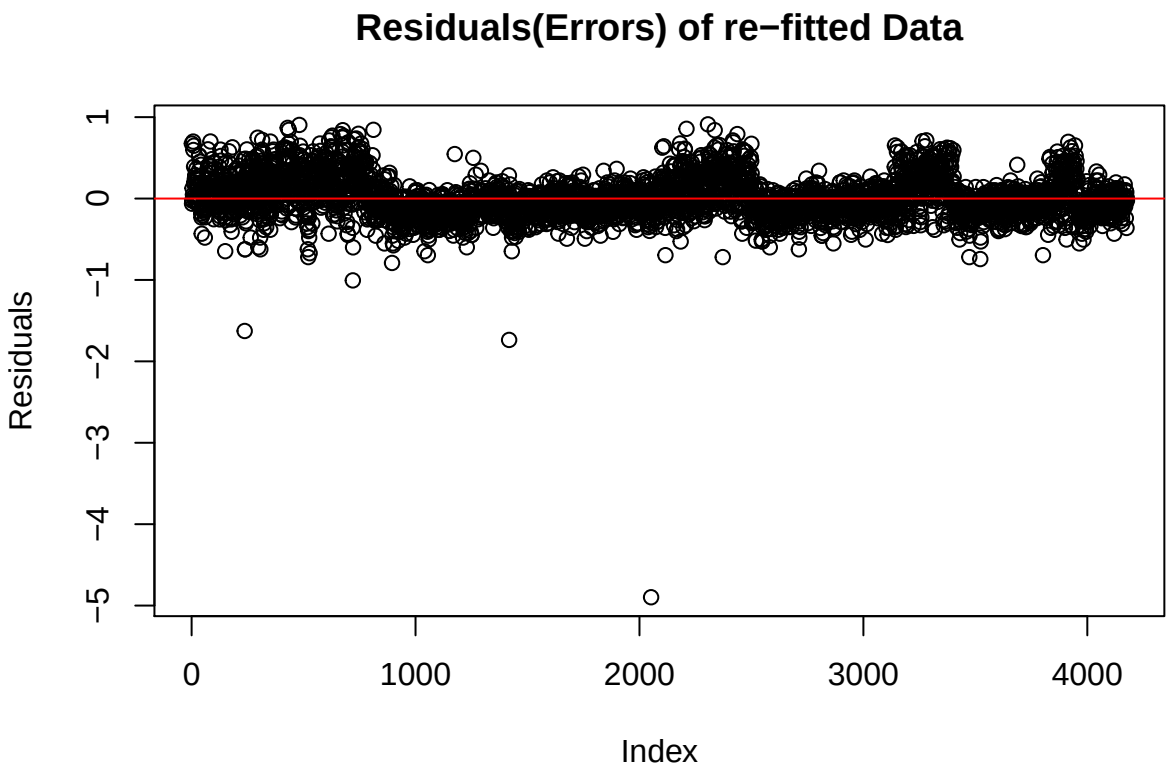
```
abalone_removed_outliers <- abalone[-c(2052,1418), ]

r_new <- abalone_removed_outliers$Rings
h_new <- abalone_removed_outliers$Height
ln_r_new <- log(r_new)
fit.Abalone_refit_outliers <- lm(ln_r_new ~ h_new)
plot(h_new, ln_r_new, xlab="Height", ylab="(Logged) Rings", main = "Logged Rings with Removed Outliers")
abline(coef(fit.Abalone_refit_outliers),col='red')
```



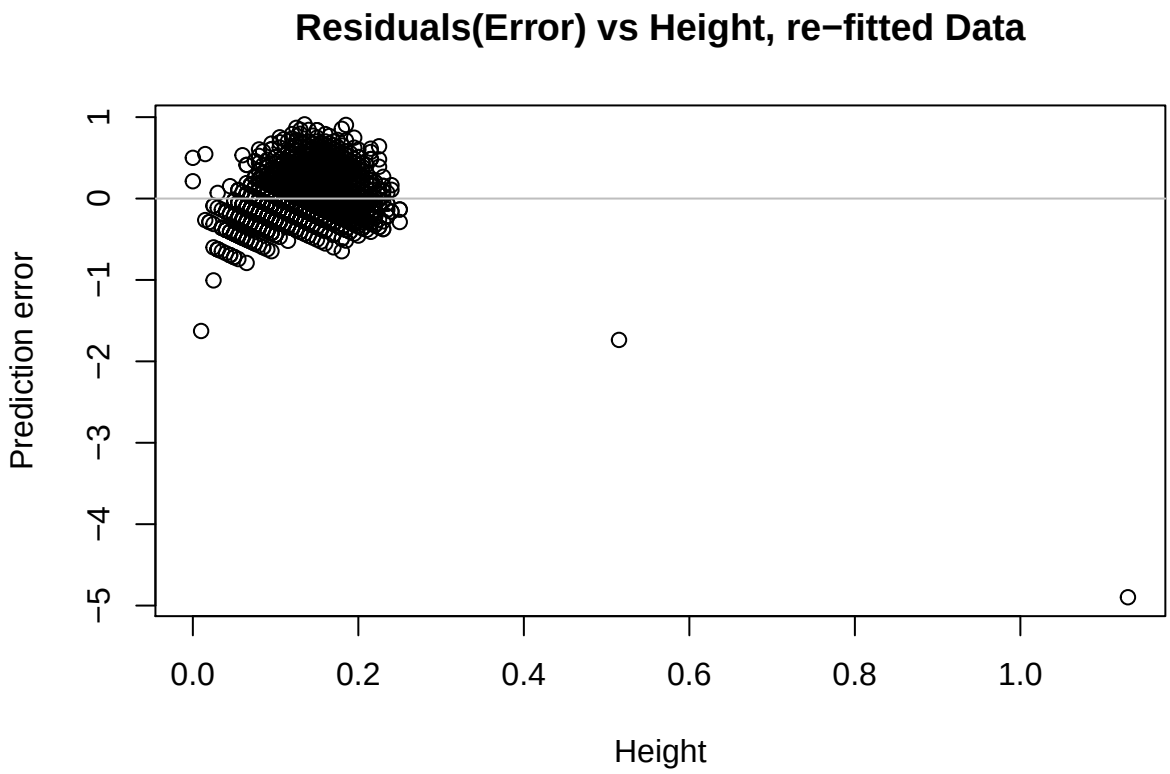

As one can see, as we "zoom in" to the part of the plot that ignores outliers, the linear fit is clearly much better; but as previously mentioned the outlisers will be kept in for the rest of the analysis. We now re-run the diagnostics.

```
abalone_res_refit = residuals(fit.Abalone_refit)
plot(abalone_res_refit, ylab ="Residuals")
title(main = 'Residuals(Errors) of re-fitted Data')
abline(h=0,col='red',lty=1)
```



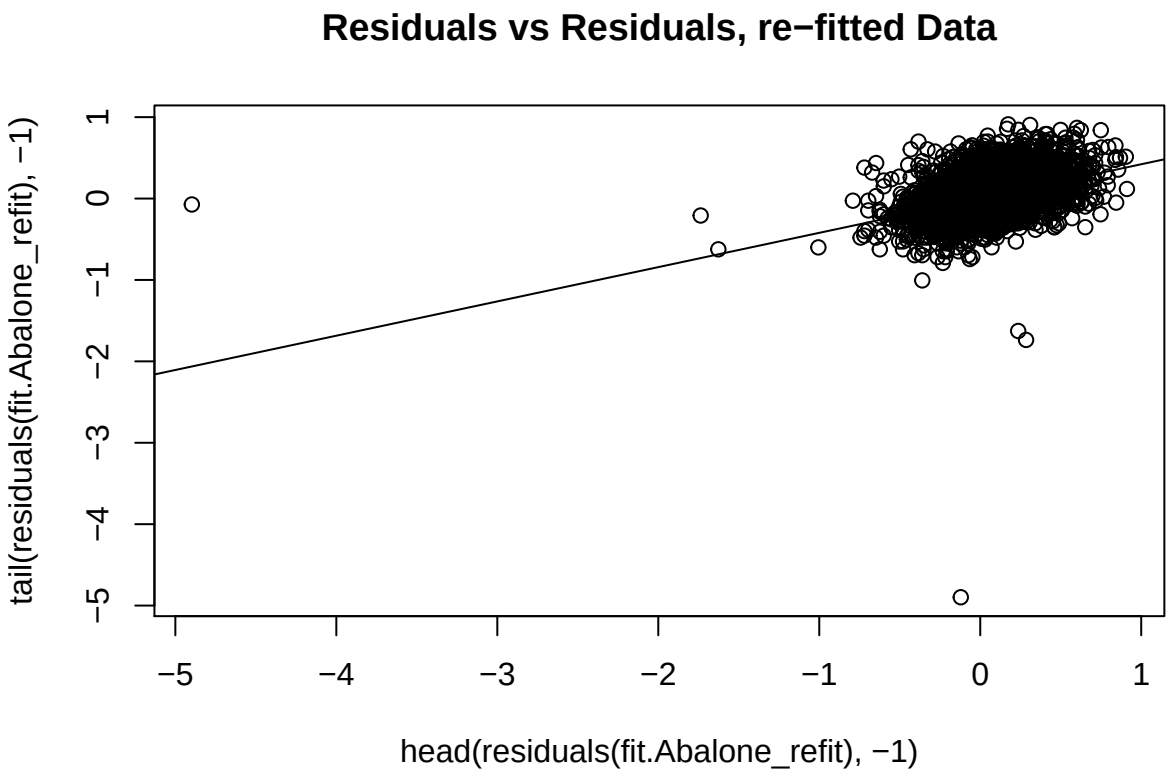

Here we once again have a graph that is broadly centred around 0, with less variance than the previous plot.

```
plot(h, residuals(fit.Abalone_refit), main = "Residuals(Error) vs Height, re-fitted Data",
xlab = "Height", ylab = "Prediction error")
abline(h = 0, col = "grey")
```



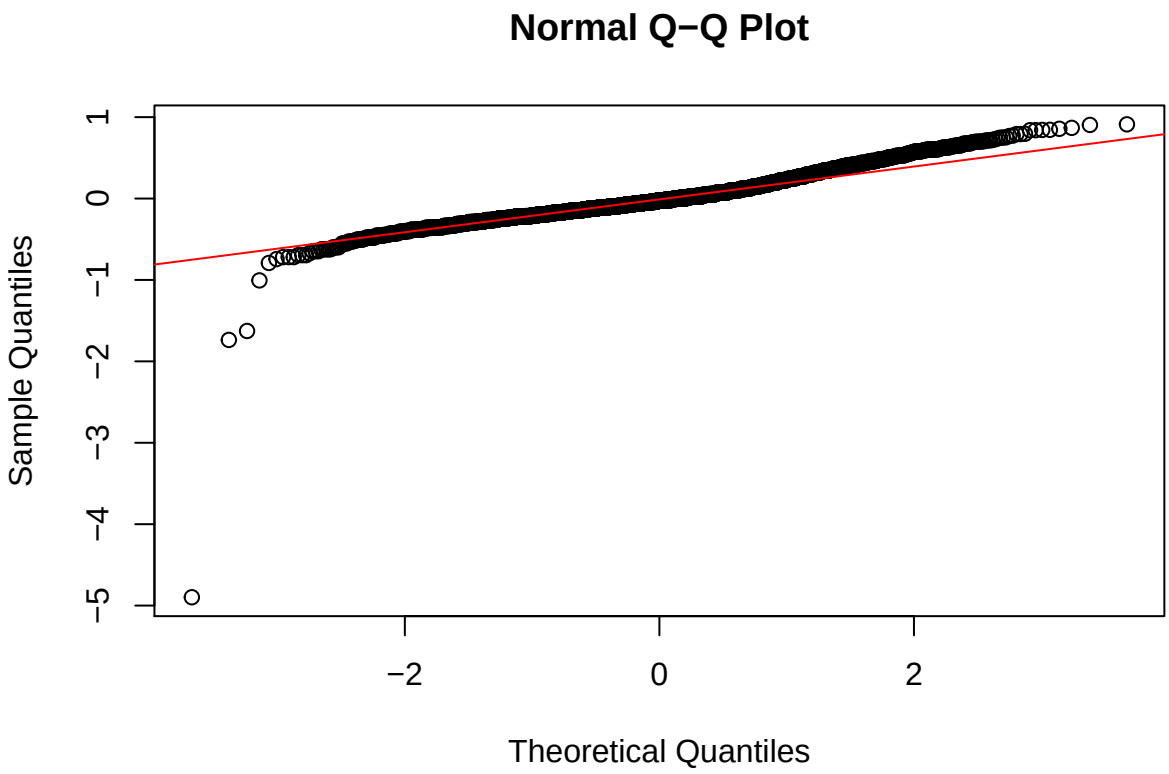

This time we have a less varying "blur" with not a whole lot of structure.

```
plot(head(residuals(fit.Abalone_refit), -1),
tail(residuals(fit.Abalone_refit), -1),main = "Residuals vs Residuals, re-fitted Data")
abline(lm(tail(residuals(fit.Abalone_refit),
-1) ~ head(residuals(fit.Abalone_refit),
-1)))
```



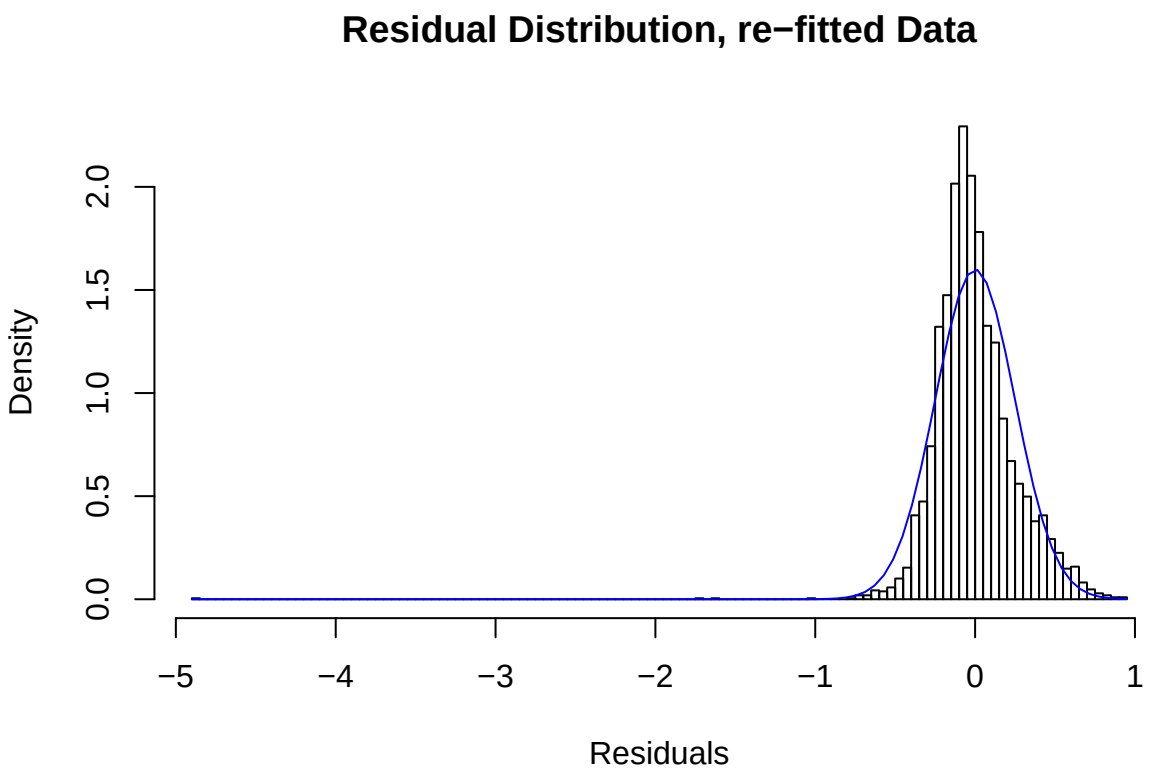

On the resid vs resid plot, we still have a patternless cloud, and the dispersion towards later values seems to have lessened. Finally let's reconsider the distributions of the residuals.

```
qqnorm(residuals(fit.Abalone_refit))
qqline(residuals(fit.Abalone_refit), col='red')
```




The divergence fron the line near later values seems to have decreased significantly, and the residuals seems to fit the Gaussian assumption much better.

```
hist(residuals(fit.Abalone_refit), breaks = 100,
freq = FALSE, xlab = "Residuals",
main = "Residual Distribution, re-fitted Data")
curve(dnorm(x, mean = 0, sd = sd(residuals(fit.Abalone_refit))),
add = TRUE, col = "blue")
```




As we can observe from here as well, the residuals fit the Gaussian distribution much better. Now we can conduct the rest of our analysis.

```
summary(fit.Abalone_refit)
```

```
##
## Call:
## lm(formula = lnr ~ h)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8975 -0.1463 -0.0269  0.1259  0.9114
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.57924    0.01344  117.52   <2e-16 ***
## h            4.77676    0.09226   51.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2494 on 4175 degrees of freedom
## Multiple R-squared:  0.391,  Adjusted R-squared:  0.3909
## F-statistic:  2681 on 1 and 4175 DF,  p-value: < 2.2e-16
```

Our parameters are $\hat{\beta}_0 = 1.57924$, and $\hat{\beta}_1 = 4.77676$. The former is what we expect the (log of) rings to be given that the height is equal to 0 (which doesn't make a whole lot of sense in this conetxt, since height = 0 doesn't physically make sense).

The latter corresponds to what we would expect the (logged) rings to differ by if we took two different sets of examples (abalones) and the height difference was 1.

We now provide 95% confidence intervals for those parameters:

```
confint(fit.Abalone_refit,level=0.95)
```

```
##                2.5 %   97.5 %
## (Intercept) 1.552897 1.605587
## h           4.595882 4.957639
```

Thus the 95% CI for $\hat{\beta}_0$ is given by (1.552897,1.605587), and one for $\hat{\beta}_1$ is given by (4.595882,4.957639). In both cases, this means that we are 95% confident that these intervals would trap the true, fixed values of $\beta_0$ and $\beta_1$ respectively. And since 0 is not trapped in the CI for $\beta_1$, we conclude that there is a statistically significant linear relationship between the heights of abalones and their ring count (and thus age).

Below is the point estimate and the 95% confidence interval for the average number of rings for abalones with height at 0.128

```
predict(fit.Abalone_refit, data.frame(h=0.128), interval = "confidence", conf.level = 0.95)
```

```
##        fit      lwr      upr
## 1 2.190668 2.182821 2.198514
```

Thus our point estimate is 2.190668, and the interval is (2.182821,2.198514). This is an indication of how well we "captured" the mean we desire, and it reflects the uncertainty in where the 'true' straight line lies. That is, suppose that the data really are randomly sampled from a Gaussian distribution. If we repeat this, and calculate a confidence interval of the mean from each sample, we'd expect about 95 % of those intervals to include the true (logged) value of the the average number of rings for abalones with height at 0.128.

And below is the predicted value for the number of rings for an abalone with height at 0.132, and a 99% prediction interval.

```
predict(fit.Abalone_refit, data.frame(h=0.132), interval = "prediction", conf.level = 0.99)
```

```
##        fit      lwr      upr
## 1 2.209775 1.720808 2.698741
```

Thus our prediction is 2.209775, and the interval is (1.720808,2.698741). This on the other hand is what we expect the (logged) rings to be in a future observation if the height was 0.132. In indicates the uncertainty in where future observed responses would lie if a collection of new observations were made. That is if the data is truly Gaussian and if we sample one more value from the population of abalones, and if we did this many times, we'd expect that next value to lie within our prediction interval in 95% of the samples.

The key finding is that there is a statistically significant linear relationship between the heights of abalones and their ring count (and thus age), at least through a re-adjusted linear model. However this is not the most practical approach, as the exponential looking trend could potentially be estimated by a higher order (non-linear) model, and such would be advisable to the researchers. In addition, the "zero height" samples are likely errors in either estimation or recording, or perhaps values so extremely small that a possible round off (we do not know if or how the observations were rounded) caused them to appear to be 0. Thus it is also advisable to go through one's data after collection to make sure that the observartions physically make sense.