Modern Regression Analysis

Q3.

Loading the bodyfat data

```
library(mfp)
```

```
Loading required package: survival
```

```
library(ggplot2)
library(readxl)
data("bodyfat")
head(bodyfat)
```

```
  case brozek siri density age weight height neck chest abdomen   hip thigh
1    1   12.6 12.3  1.0708  23 154.25  67.75 36.2  93.1    85.2  94.5  59.0
2    2    6.9  6.1  1.0853  22 173.25  72.25 38.5  93.6    83.0  98.7  58.7
3    3   24.6 25.3  1.0414  22 154.00  66.25 34.0  95.8    87.9  99.2  59.6
4    4   10.9 10.4  1.0751  26 184.75  72.25 37.4 101.8    86.4 101.2  60.1
5    5   27.8 28.7  1.0340  24 184.25  71.25 34.4  97.3   100.0 101.9  63.2
6    6   20.6 20.9  1.0502  24 210.25  74.75 39.0 104.5    94.4 107.8  66.0
  knee ankle biceps forearm wrist
1 37.3  21.9   32.0    27.4  17.1
2 37.3  23.4   30.5    28.9  18.2
3 38.9  24.0   28.8    25.2  16.6
4 37.3  22.8   32.4    29.4  18.2
5 42.2  24.0   32.2    27.7  17.7
6 42.0  25.6   35.7    30.6  18.8
```

Converting weight(lbs) to weight(kgs)

```
bodyfat$weight = bodyfat$weight * 0.45359237
head(bodyfat)
```

```
  case brozek siri density age   weight height neck chest abdomen   hip thigh
1    1   12.6 12.3  1.0708  23 69.96662  67.75 36.2  93.1    85.2  94.5  59.0
2    2    6.9  6.1  1.0853  22 78.58488  72.25 38.5  93.6    83.0  98.7  58.7
3    3   24.6 25.3  1.0414  22 69.85322  66.25 34.0  95.8    87.9  99.2  59.6
4    4   10.9 10.4  1.0751  26 83.80119  72.25 37.4 101.8    86.4 101.2  60.1
5    5   27.8 28.7  1.0340  24 83.57439  71.25 34.4  97.3   100.0 101.9  63.2
6    6   20.6 20.9  1.0502  24 95.36780  74.75 39.0 104.5    94.4 107.8  66.0
  knee ankle biceps forearm wrist
1 37.3  21.9   32.0    27.4  17.1
2 37.3  23.4   30.5    28.9  18.2
3 38.9  24.0   28.8    25.2  16.6
4 37.3  22.8   32.4    29.4  18.2
5 42.2  24.0   32.2    27.7  17.7
6 42.0  25.6   35.7    30.6  18.8
```

Fitting the model between Y: body fat(%) versus X: weight (in kg).

```
model1 = lm(brozek ~ weight,data=bodyfat)
model1
```

```

Call:
lm(formula = brozek ~ weight, data = bodyfat)

Coefficients:
(Intercept)       weight
    -9.9952       0.3565
```

1) Interpret the value of $\hat{\beta}_1$ and explain why we should avoid interpreting $\hat{\beta}_0$

$\hat{\beta}_1$ is the slope of this model. This value represents the expected change in body fat(%) (Y) for each additional kilogram of weight (X). Here, $\hat{\beta}_1$ takes the value of 0.3565. This means for a kilogram increase in body weight, there will be a 0.3565% increase in body fat(%).

$\hat{\beta}_0$ is the intercept of this model. This represents the expected value of body fat(%) when weight is zero. This will never happen in a real life scenario so there is no meaning in interpreting this.

2) Fitting a second model with two co-variates

```
model2 = lm(brozek ~ weight + abdomen,data=bodyfat)
model2
```

```
Call:
lm(formula = brozek ~ weight + abdomen, data = bodyfat)

Coefficients:
(Intercept)       weight      abdomen
   -41.3481      -0.3008       0.9151
```

The coefficient for the weight attribute has changed when we included abdomen attribute. This means that this new value for the weight attribute better describes the data along with the abdomen attribute. This signifies having the weight alone as a parameter to predict the body fat(%) may be misleading and including other attirbutes can change the effect of a particular attribute.

3) Which of the two models provides a better fit to the data?

```
coefficient_of_determinations = c(summary(model1)$r.squared,
                                  summary(model2)$r.squared)
coefficient_of_determinations
```

```
[1] 0.3759604 0.7187265
```

As we can see, model1 has coefficient of determination of 0.3759604 and model2 has a coefficient of determination of 0.7187265. The second model has a higher coefficient of determination. Which means that the variability in body fat(%) is better explained when both the co-variates abdomen and weight are involved. Basically model2 better fits the data.

Q4.

```
population_data = read.csv("data_simulation.csv")
head(population_data)
```

```
           y X1        X2        X3
1  6.3200411  6 0.9477644 0.5604356
2  5.1830886  7 1.7181174 0.7957037
3  0.5988436  2 0.9534741 0.5858352
4  4.8122950  6 2.0977570 0.6113976
5  3.5291469  6 0.6679458 0.7317442
6 -0.1604793  3 0.5583715 0.4340481
```

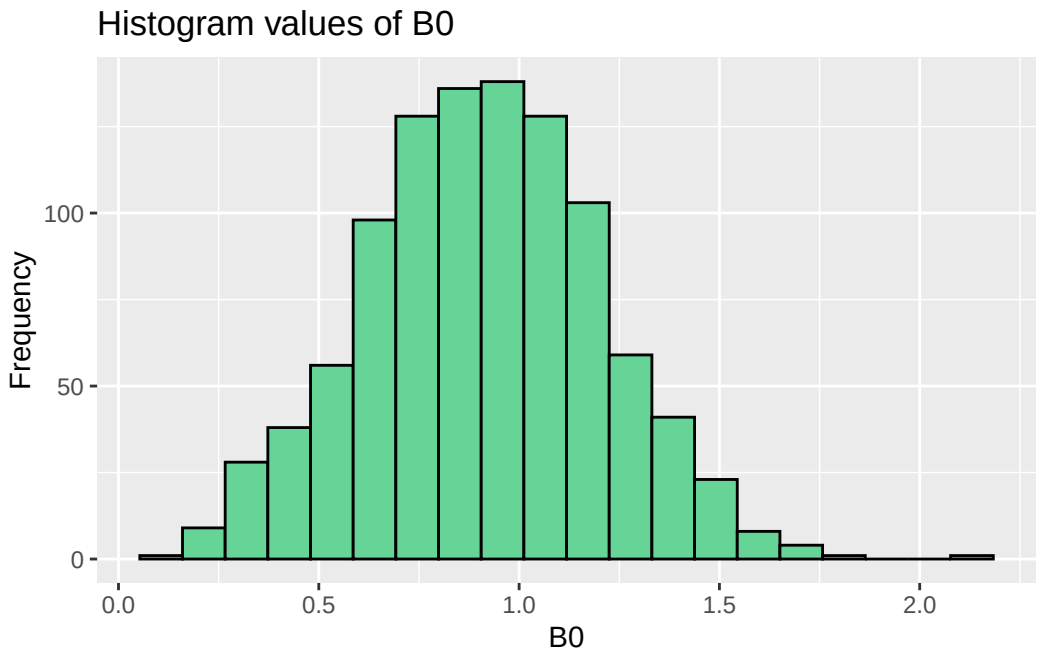
Step 1:

```
n = 300
values = matrix(NA,ncol=4,nrow=1000)
for(i in 1:1000){
  sampled_indexes = sample(1:1000,size=n)
  data_sample = population_data[sampled_indexes,]

  model = lm(y ~ X1+X2+X3,data=data_sample)
  values[i,] = coef(model)
}

colnames(values) = c("B0","B1","B2","B3")
head(values)
```

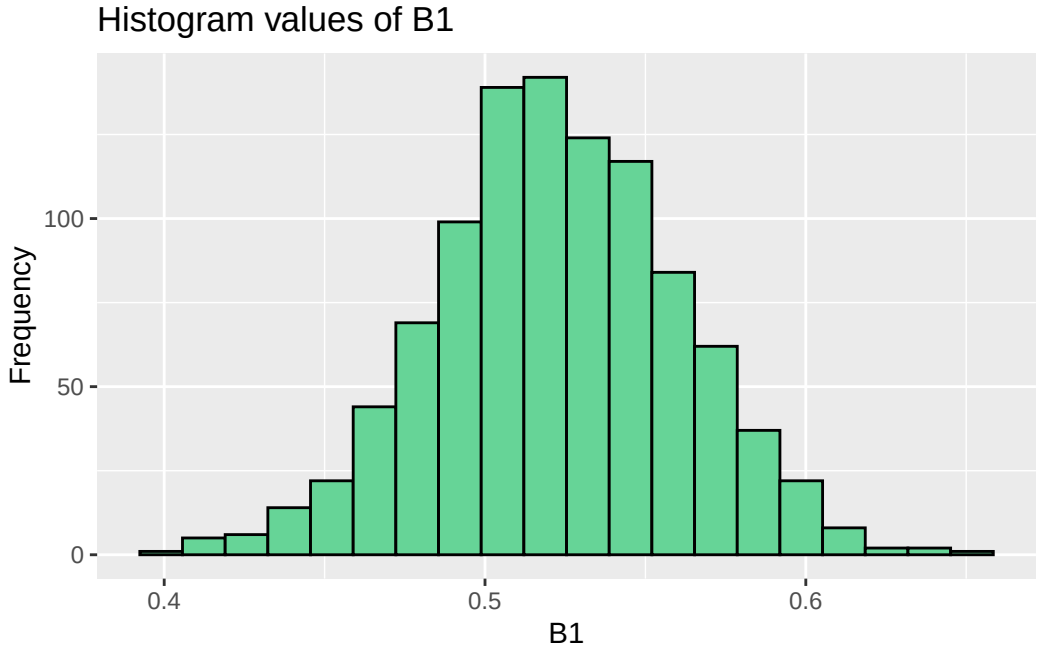
```
            B0        B1        B2        B3
[1,] 0.8736381 0.4837608 1.0252971 -2.525853
[2,] 1.1293593 0.5621636 0.9722918 -3.695763
[3,] 1.1961601 0.4381644 1.2973772 -3.011622
[4,] 1.1343932 0.4975025 1.0968463 -3.077442
[5,] 0.7150365 0.5137650 1.0856133 -2.686502
[6,] 0.6783074 0.4934982 1.1567858 -2.498793
```

1) Plotting estimates ($\hat{\beta}_0$,$\hat{\beta}_1$,$\hat{\beta}_2$,$\hat{\beta}_3$), i = 1…….1000 using histograms.

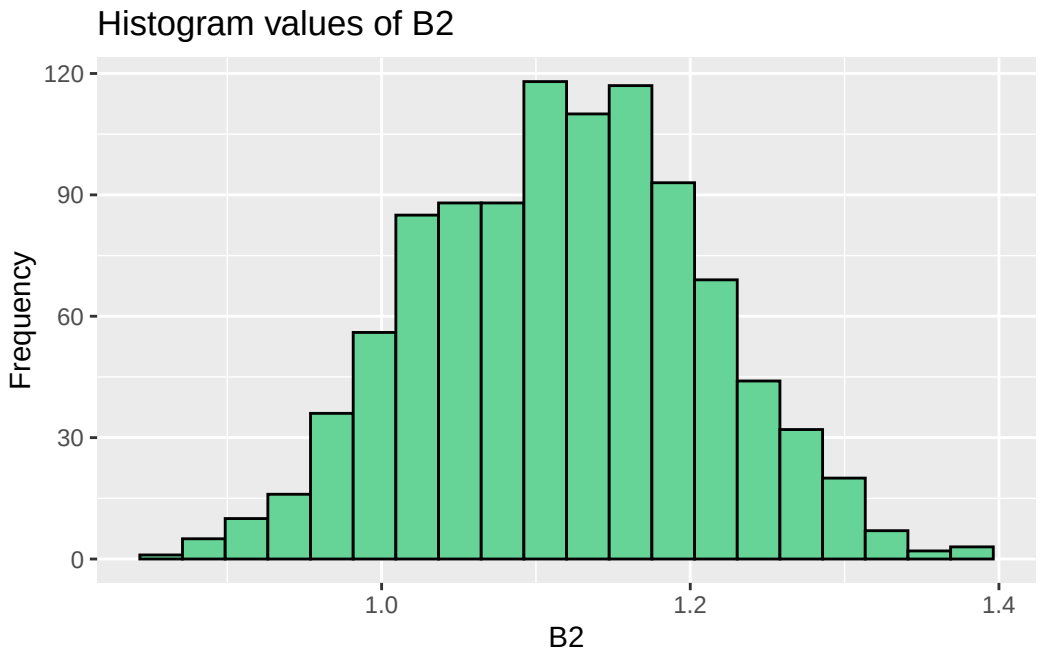
```
ggplot(values, aes(x=B0)) + geom_histogram(bins=20,
      color = "#000000",fill = "#66d497") +
      labs(title="Histogram values of B0",x="B0",y="Frequency")
```



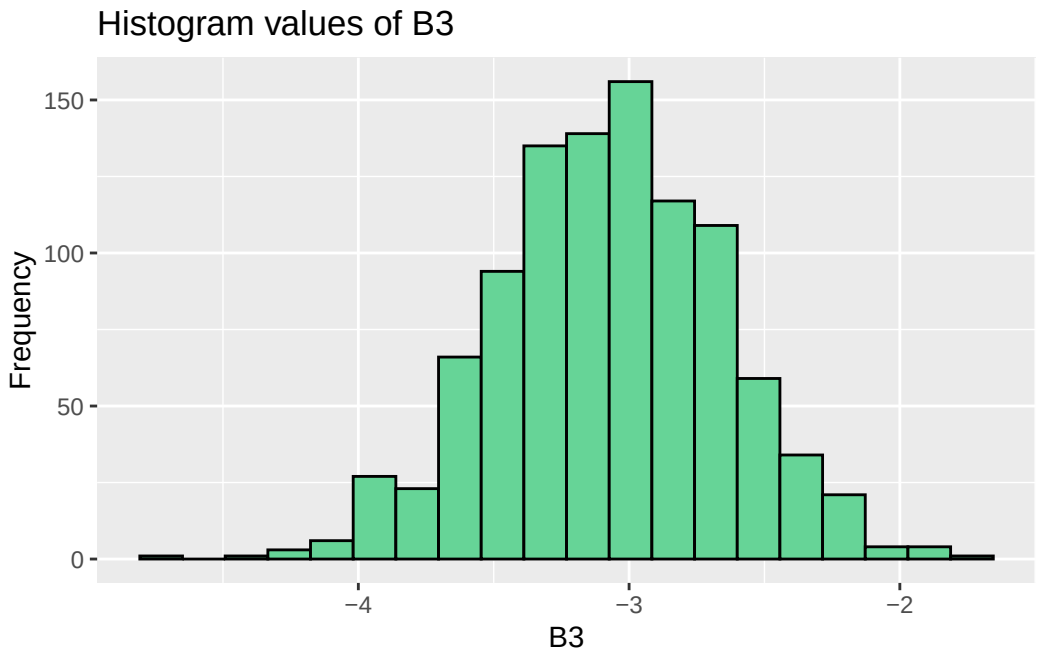

```
ggplot(values, aes(x=B1)) + geom_histogram(bins=20,
      color = "#000000",fill = "#66d497") +
      labs(title="Histogram values of B1",x="B1",y="Frequency")
```



```
ggplot(values, aes(x=B2)) + geom_histogram(bins=20,
      color = "#000000",fill = "#66d497") +
      labs(title="Histogram values of B2",x="B2",y="Frequency")
```




```
ggplot(values, aes(x=B3)) + geom_histogram(bins=20,
      color = "#000000",fill = "#66d497") +
      labs(title="Histogram values of B3",x="B3",y="Frequency")
```



2) Repeat the same for n=50

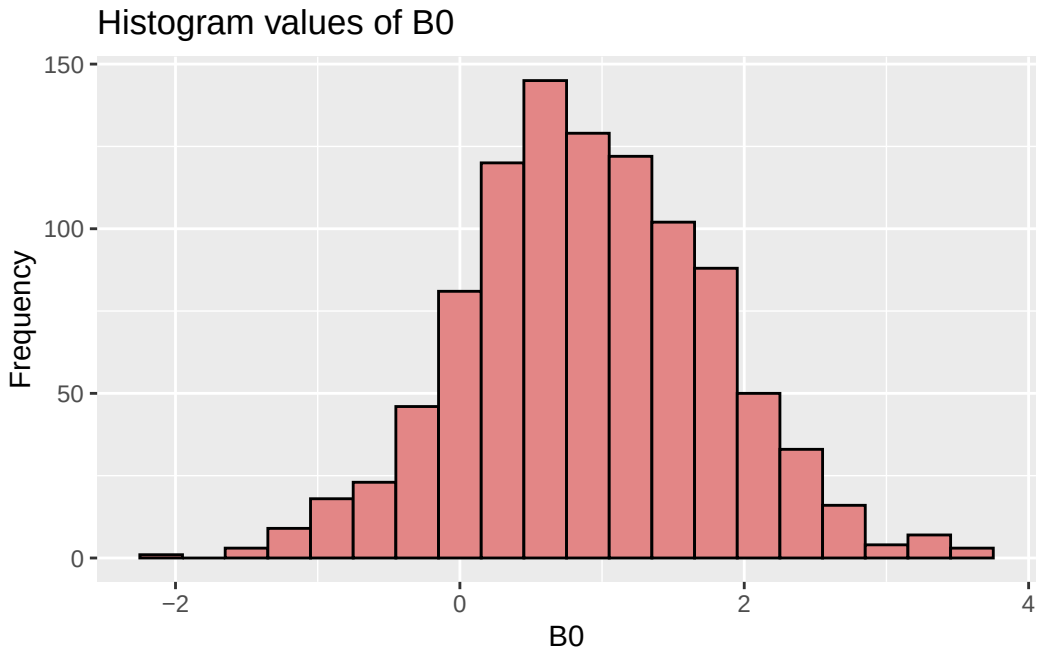
```
n = 50
values = matrix(NA,ncol=4,nrow=1000)
for(i in 1:1000){
  sampled_indexes = sample(1:1000,size=n)
  data_sample = population_data[sampled_indexes,]

  model = lm(y ~ X1+X2+X3,data=data_sample)
  values[i,] = coef(model)
}

colnames(values) = c("B0","B1","B2","B3")
head(values)
```

```
            B0        B1        B2        B3
[1,]  0.6563956 0.5299743 0.8693365 -2.741965
[2,] -0.3381550 0.6725591 1.6608192 -3.102145
[3,]  2.2857901 0.5646369 0.5012519 -5.194166
[4,]  0.3748061 0.5399150 1.0190871 -2.400590
[5,]  1.4268931 0.5740708 1.5473284 -5.014279
```

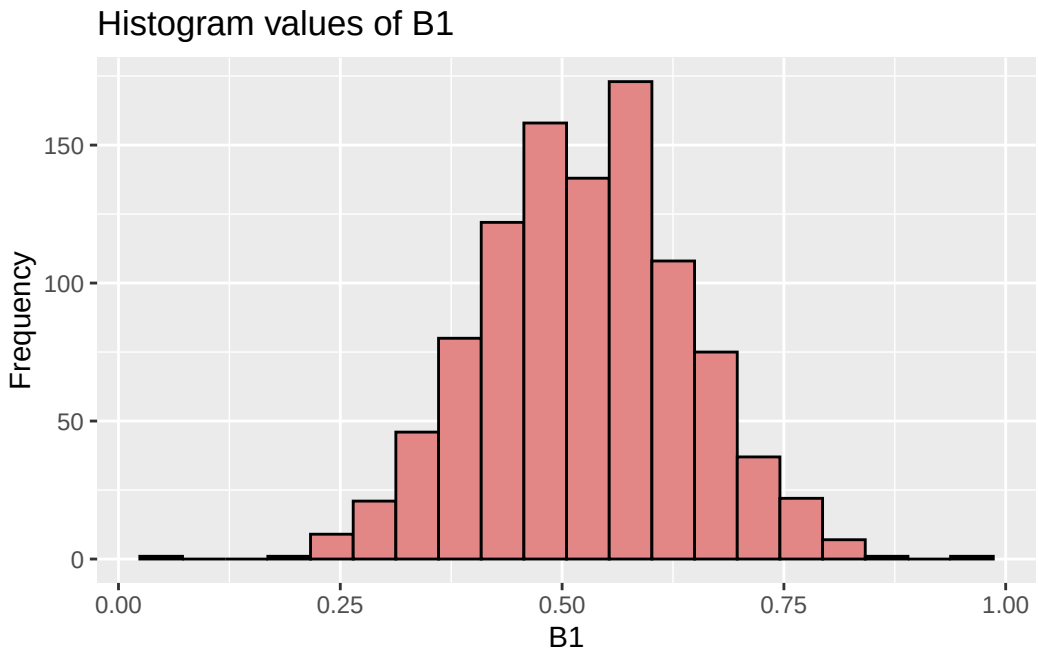
```
[6,]  0.3287334 0.7057120 0.9801311 -3.875869
```

Plotting the betas as histograms

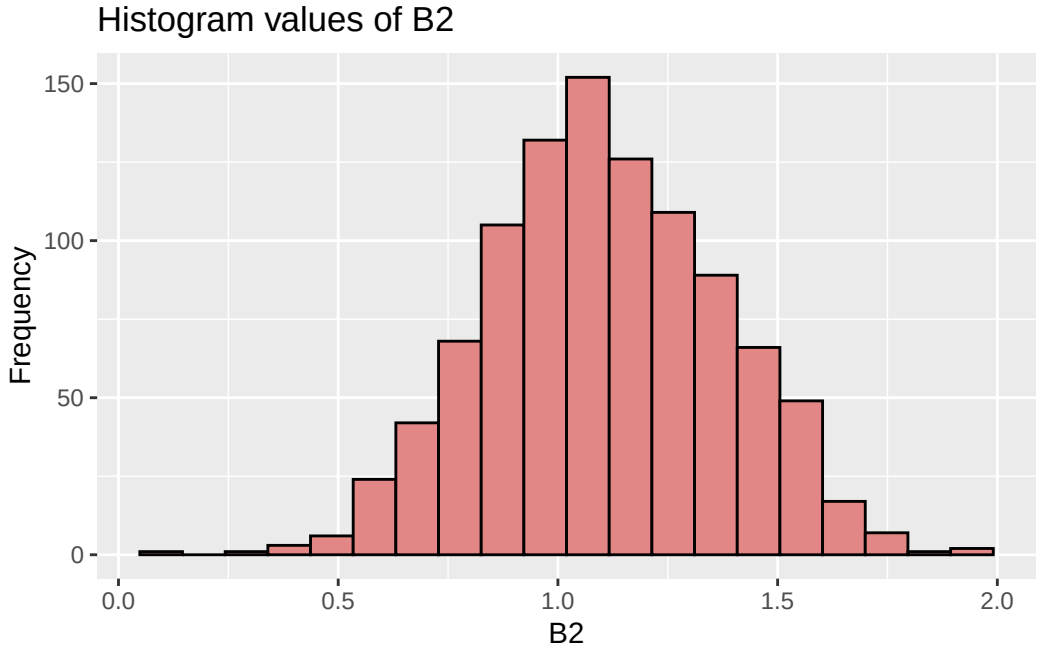
```
ggplot(values, aes(x=B0)) + geom_histogram(bins=20,
      color = "#000000",fill = "#e38686") +
      labs(title="Histogram values of B0",x="B0",y="Frequency")
```



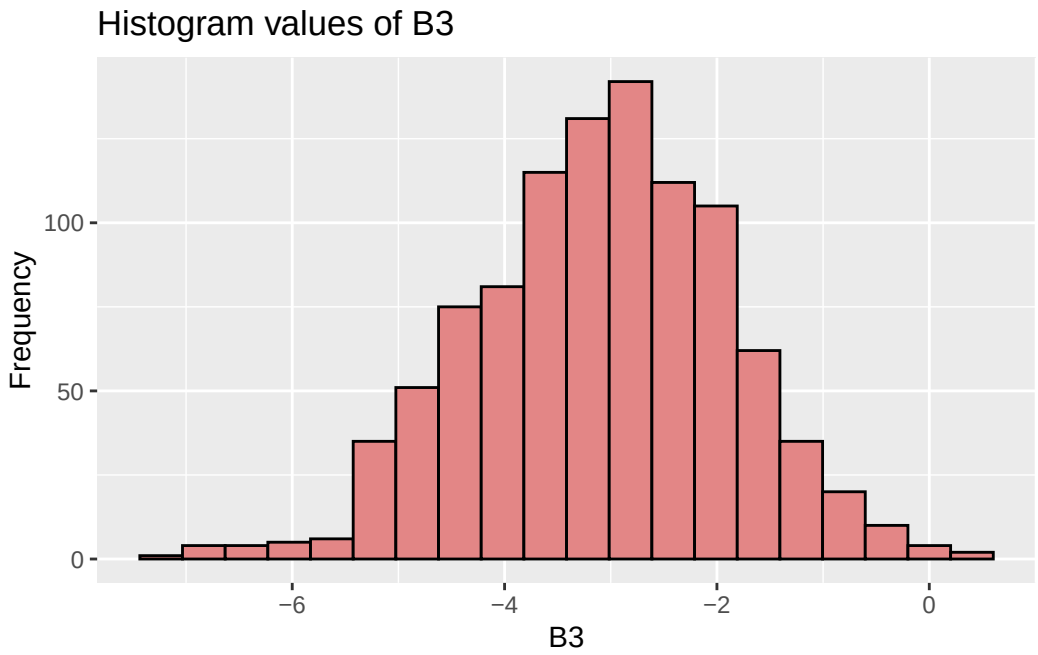

```
ggplot(values, aes(x=B1)) + geom_histogram(bins=20,
      color = "#000000",fill = "#e38686") +
      labs(title="Histogram values of B1",x="B1",y="Frequency")
```




```
ggplot(values, aes(x=B2)) + geom_histogram(bins=20,
      color = "#000000",fill = "#e38686") +
      labs(title="Histogram values of B2",x="B2",y="Frequency")
```



```
ggplot(values, aes(x=B3)) + geom_histogram(bins=20,
      color = "#000000",fill = "#e38686") +
      labs(title="Histogram values of B3",x="B3",y="Frequency")
```



When the sample size n = 300, the estimates are more tightly centered true values.

When the sample size n = 50, the estimates have more variability and may deviate from the true values.

Increasing the sample size, reduces the variability of the parameters.

3) Property of least squares which is illustrated in item 2.

The comparison talks about the consistency parameter of the least squares estimation. As the sample size increases the estimated model parameters are closer to the true values. With a lower sample size, the model may suffer from sampling bias and the consistency might be less. So the estimates might deviate from the true values.

Q5.

1) If our goal is to model the prices, which attribute would you select for a simple linear model? Why?

Looking at the correlation plot, if we just need a simple linear model to model the prices, then the attribute with the highest correlation to price can we chosen, which is taxes with a correlation score of 0.88. Since, taxes is more positively correlated with the price, this would be a good attribute in a linear model.

2) A model was fitted between Y : price and X : baths. Use the output shown to answer "By how much does a house price increase in average with the addition of one bath?"

We see that the parameter estimate corresponding to the baths attribute is 17.775. This means that for addition of one bath, the price increases by 17.775units by average.

3) Use the output to test the hypothesis H0 : $\hat{\beta}_1 = 0$ versus H1 : $\hat{\beta}_1 \neq 0$ using $\alpha = 0.05$. Justify and interpret your conclusion.

Hypothesis $\hat{\beta}_1 = 0$ is the null hypothesis. This means there is no correlation between the price and the baths.

Hypothesis $\hat{\beta}_1 \neq 0$. This means that we can correlate price and baths. As we see from the output given, the p value = 0.0000927, which is less than the significance level (0.05). This means that we can reject the null hypothesis and conclude there is a relationship between the baths and prices.

Q6. Data analysis

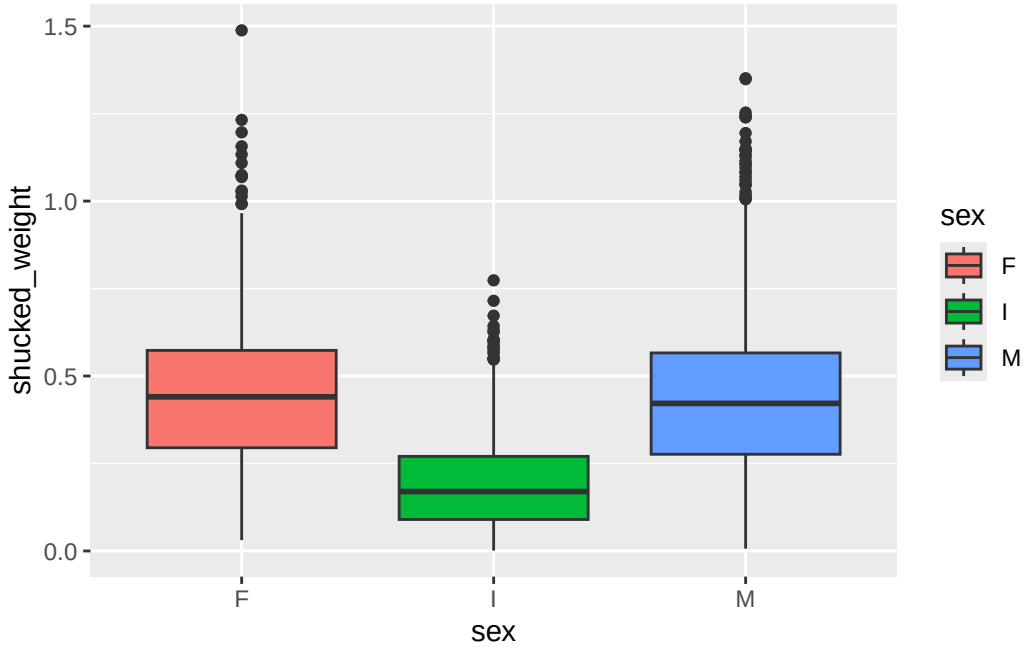
1) Exploring gender:

```
abalone = read.csv("abalone.csv")
head(abalone)
```

```
  X sex length diameter height whole_weight shucked_weight viscera_weight
1 1   M  0.455    0.365  0.095       0.5140         0.2245         0.1010
2 2   M  0.350    0.265  0.090       0.2255         0.0995         0.0485
3 3   F  0.530    0.420  0.135       0.6770         0.2565         0.1415
4 4   M  0.440    0.365  0.125       0.5160         0.2155         0.1140
```

```
5 5   I  0.330    0.255  0.080       0.2050          0.0895          0.0395
6 6   I  0.425    0.300  0.095       0.3515          0.1410          0.0775
  shell_weight rings
1        0.150    15
2        0.070     7
3        0.210     9
4        0.155    10
5        0.055     7
6        0.120     8
```

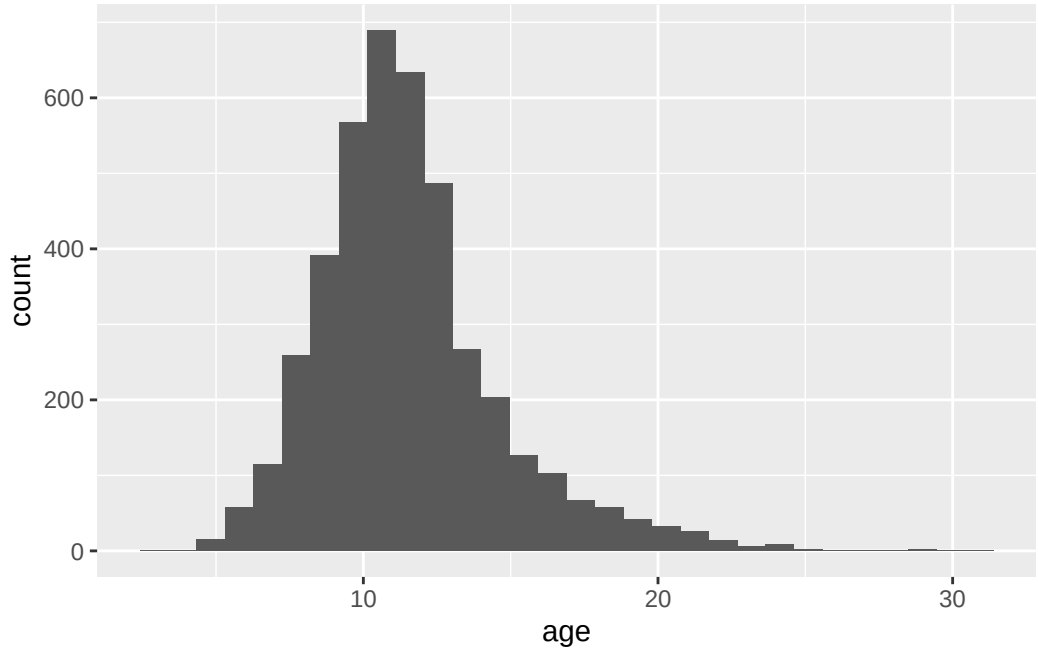
```
#abalone_male = abalone[abalone$sex == "M",]
#abalone_female = abalone[abalone$sex == "F",]
ggplot(abalone, aes(y=shucked_weight,x=sex,fill=sex)) + geom_boxplot()
```



There is no significant difference between male and female abalones, but the weight distribution for the infant abalones are relatively lower to that of the male and females abalones.

2) Exploring rings:

```
abalone[,"age"] = abalone[,"rings"] + 1.5
ggplot(abalone,aes(x=age)) + geom_histogram(bins=30)
```



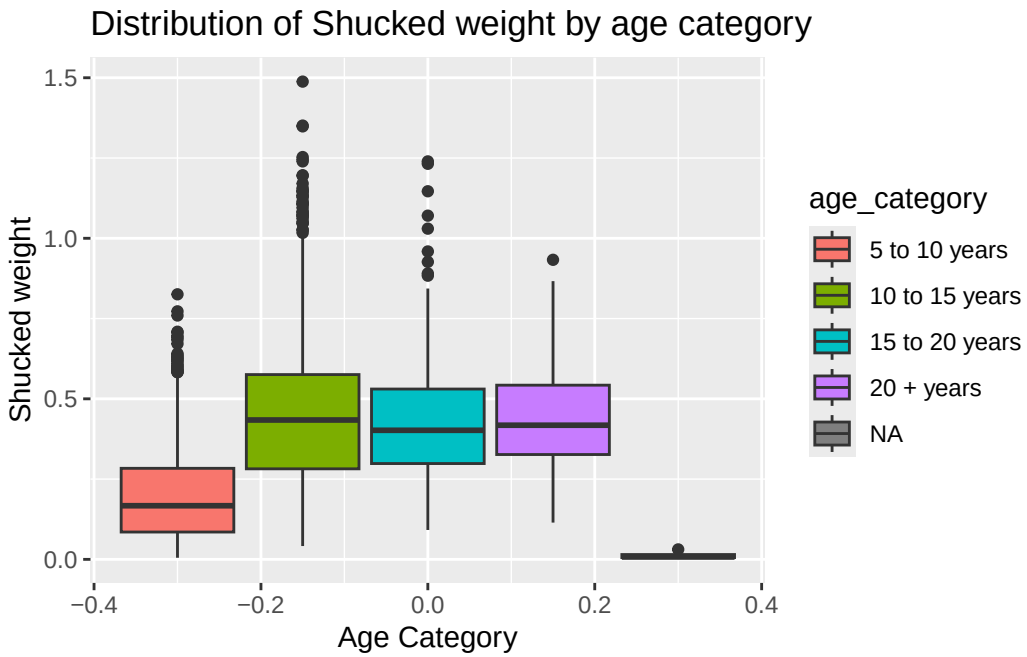
The distribution of age is skewed to the right.

3) Exploring rings (2):

```
abalone[,"age_category"] = cut(abalone$age,breaks= c(5,10,15,20,Inf),
                               labels = c("5 to 10 years","10 to 15 years",
                                          "15 to 20 years","20 + years"))
```

Plotting the distribution of Y per age category

```
ggplot(abalone,aes(y=shucked_weight,fill=age_category)) +
  geom_boxplot() +
  labs(title="Distribution of Shucked weight by age category",
       x="Age Category",y="Shucked weight")
```



The distribution is relatively same for the age categories 10 to 15 years, 15 to 20 years and 20 + years while the shucked weight is lesser for the age category 5 to 10 years.

4) Correlations

```
abalone_numeric <- abalone[, sapply(abalone, is.numeric)]
correlation <- cor(abalone_numeric)
correlation["shucked_weight",]
```

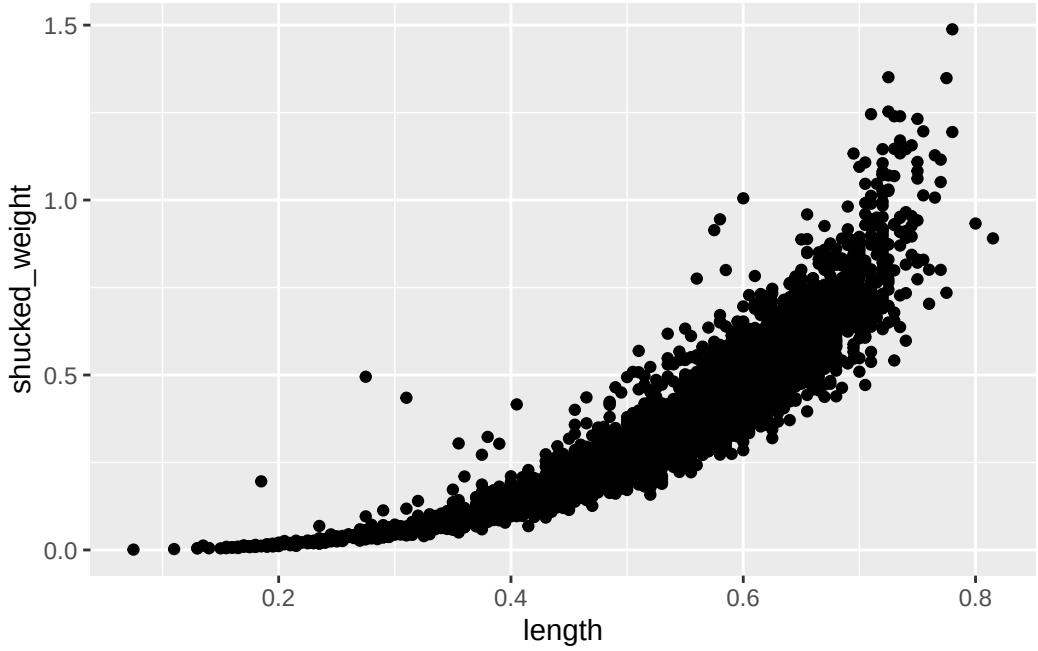
```
            X         length       diameter         height   whole_weight
    0.1022523      0.8979137      0.8931625      0.7749723      0.9694055
shucked_weight viscera_weight   shell_weight          rings            age
    1.0000000      0.9319613      0.8826171      0.4208837      0.4208837
```

The correlation is the highest between shucked_weight and whole_weight. The correlation is 0.9694055

5) Scatterplots:

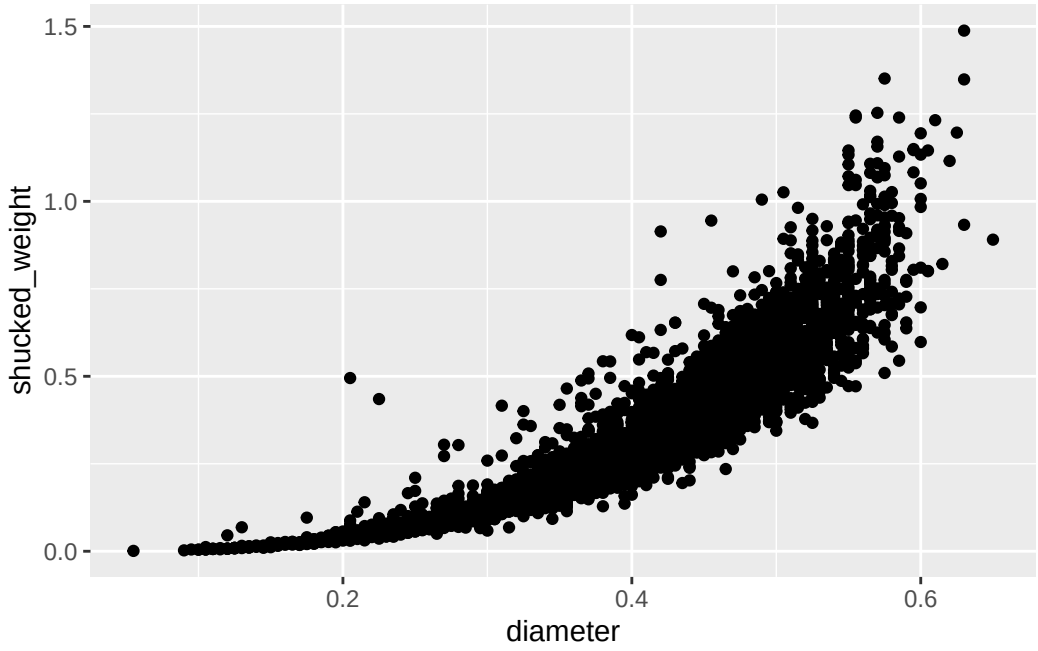
Y versus X: length

```
ggplot(abalone,aes(y=shucked_weight,x=length)) + geom_point()
```



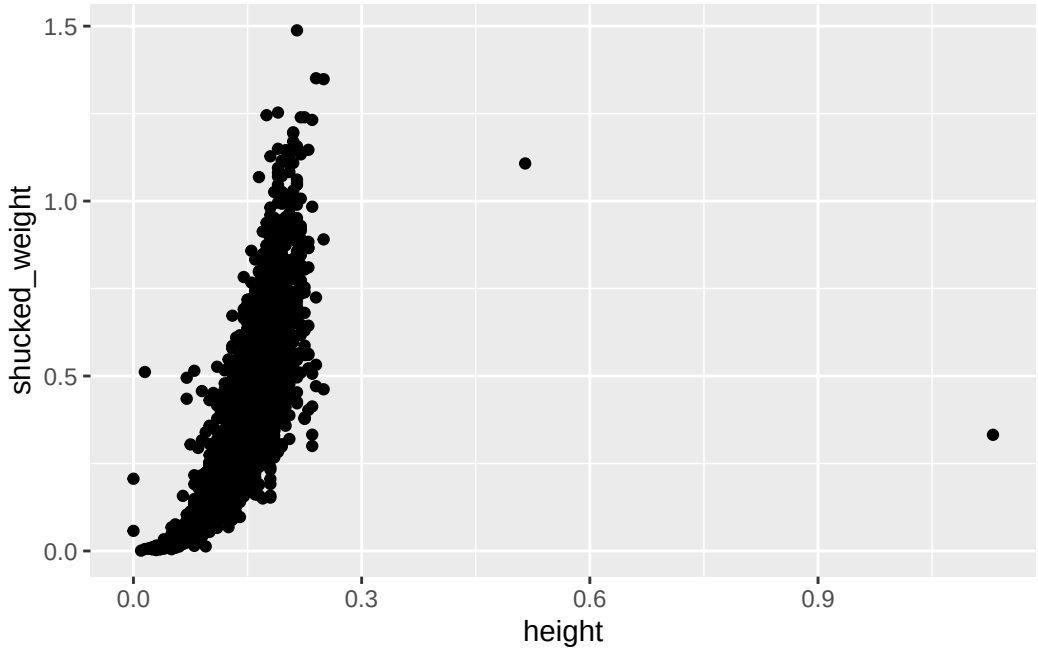
Y versus X: diamater

```
ggplot(abalone,aes(y=shucked_weight,x=diameter)) + geom_point()
```



Y versus X: height

```
ggplot(abalone,aes(y=shucked_weight,x=height)) + geom_point()
```



6) Fitting a linear model for Y using length, diameter, height and rings.

```
abalone_model = lm(shucked_weight ~ length + diameter + height + rings,
                   data=abalone)
summary(abalone_model)
```

```
Call:
lm(formula = shucked_weight ~ length + diameter + height + rings,
    data = abalone)

Residuals:
     Min       1Q   Median       3Q      Max
-0.60995 -0.06106 -0.01701  0.04044  0.66542

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.4673450  0.0068785 -67.943  < 2e-16 ***
length       1.0166016  0.0745563  13.635  < 2e-16 ***
diameter     0.7308167  0.0924717   7.903 3.45e-15 ***
height       0.6786035  0.0635168  10.684  < 2e-16 ***
rings       -0.0099401  0.0005569 -17.849  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09315 on 4172 degrees of freedom
Multiple R-squared:  0.8241,    Adjusted R-squared:  0.8239
F-statistic:  4885 on 4 and 4172 DF,  p-value: < 2.2e-16
```

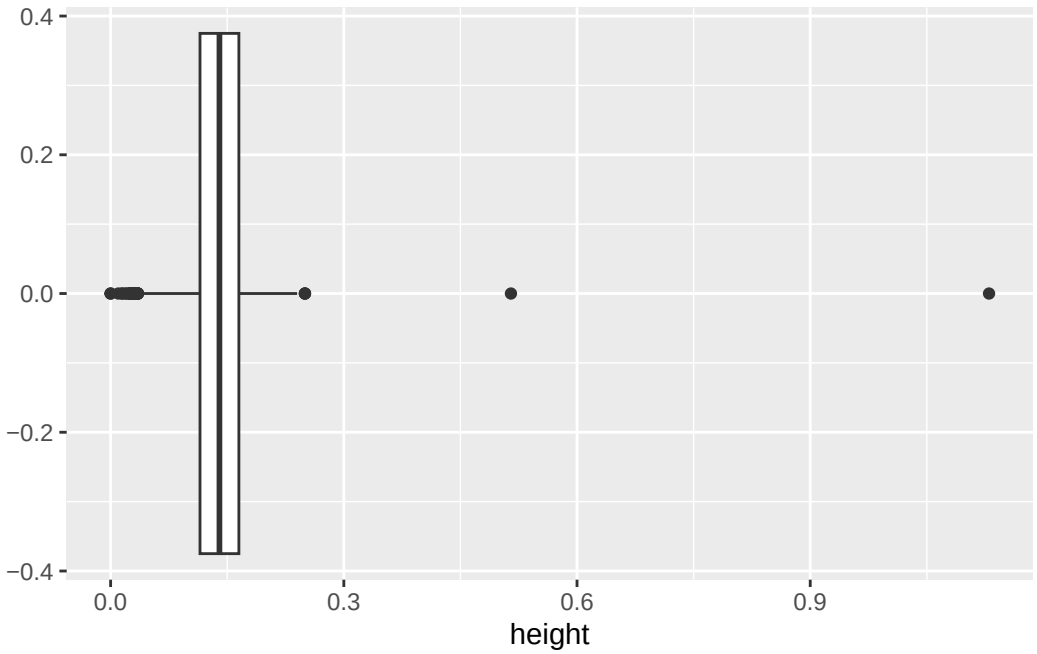
The equation is Y = -0.46734 + 1.01660(length) + 0.73082(diameter) + 0.67860(height) - 0.00994(rings)

Coefficient of determination: 0.8241.

82.41% of the variability in Y:shucked_weight is explained by the four co-variates length,diameter,height and rings.

7) Outliers:

```
ggplot(abalone,aes(x=height)) + geom_boxplot()
```



The above boxplot shows the heights of abalone. We can clearly see that two abalone with an abnormally high height marked on the boxplot.

Interpretation of the boxplot:

1. The box shows the IQR, which ranges from the 25% to 75% of the data values, which is referred to as the first quartile (Q1) and the third quartile (Q3). The middle 50% of the data values are captured within this region.
2. The line in the middle of the box signifies the median, which divides the data into two equal halves.
3. The whiskers extend to the data points which are within 1.5 times the value of the IQR both sides from Q1 and Q3.
4. The two abalone height values beyond this whisker are the outliers which is shown in the boxplot.

8) Outliers in regression:

```r
outlier_model = lm(shucked_weight ~ height,data = abalone)
outlier_model
```

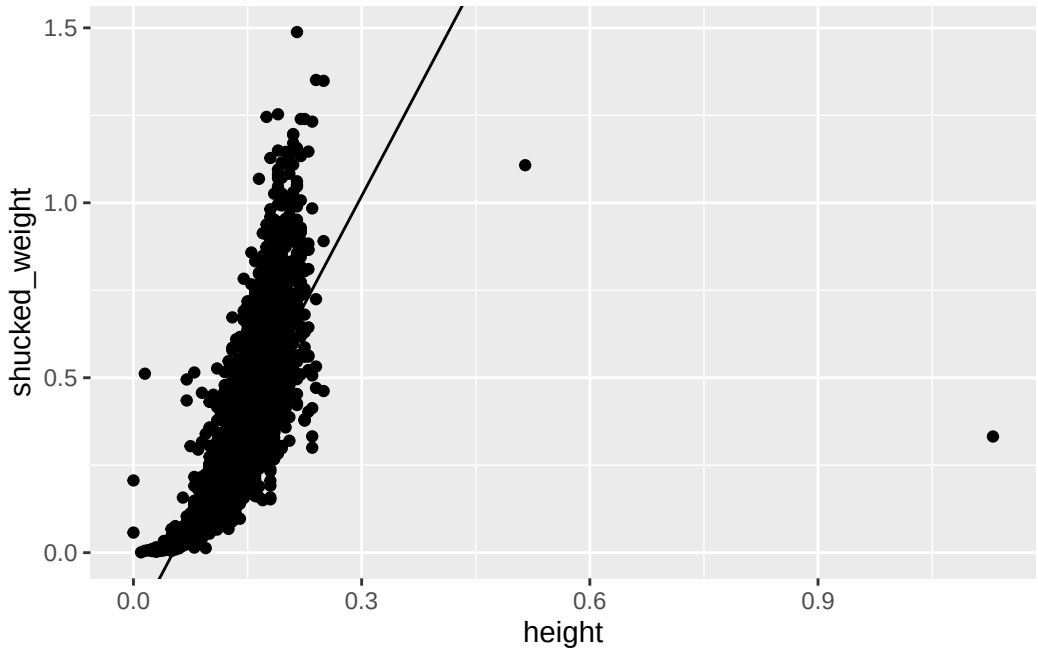
```
Call:
lm(formula = shucked_weight ~ height, data = abalone)

Coefficients:
(Intercept)       height
    -0.2144       4.1125
```

$\hat{\beta}_0$ = -0.2144

$\hat{\beta}_1$ = 4.1125

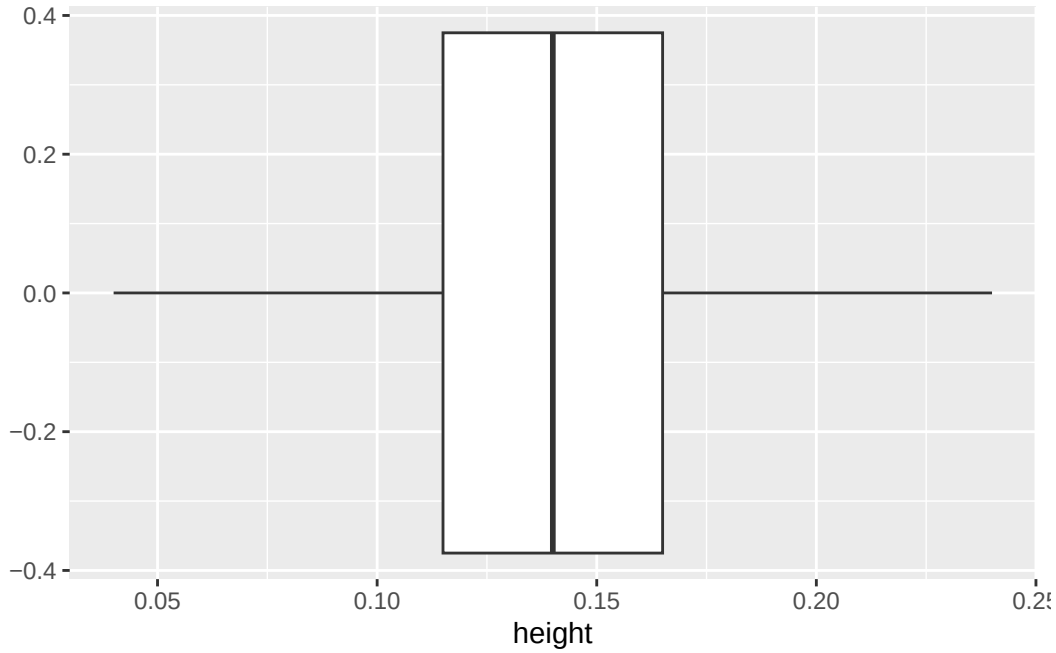
Visualizing the regression line before removing the outliers.

```r
ggplot(abalone,aes(x=height,y=shucked_weight)) +
  geom_point() +
  geom_abline(intercept = coef(outlier_model)[1],
              slope = coef(outlier_model)[2])
```



Removing the outliers:

```
q1 = quantile(abalone$height,0.25)
q3 = quantile(abalone$height,0.75)

iqr = IQR(abalone$height)

lower = q1 - 1.5 * iqr
upper = q3 + 1.5 * iqr

abalone_outlier_removed = abalone[abalone$height >= lower &
                                  abalone$height <= upper,]

ggplot(abalone_outlier_removed, aes(x=height)) + geom_boxplot()
```



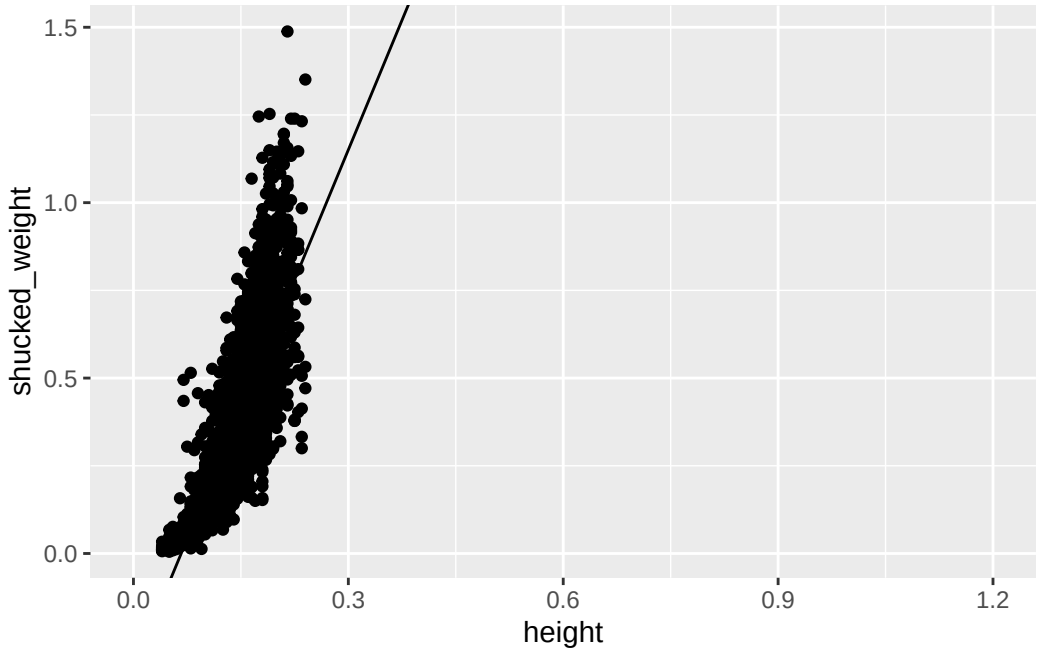

```
outlier_removed_model = lm(shucked_weight ~ height,
                           data=abalone_outlier_removed)
outlier_removed_model
```

```
Call:
lm(formula = shucked_weight ~ height, data = abalone_outlier_removed)
```

```
Coefficients:
(Intercept)       height
    -0.3285       4.9308
```

Visualizing the regression line after removing outliers

```
ggplot(abalone_outlier_removed,aes(x=height,y=shucked_weight)) +
  geom_point() +
  geom_abline(intercept = coef(outlier_removed_model)[1],
              slope = coef(outlier_removed_model)[2]) +
  scale_x_continuous(limits = c(0, 1.2),
                     breaks = seq(0, 1.2, by = 0.3))
```



After removing the outliers

$\hat{\beta_0}$ = -0.3285

$\hat{\beta_1}$ = 4.9308

Removal of the outliers has changed the regression parameters. The intercept has become more negative and the slope has increased. This is a better regression line as we can see from the scatterplot from when we had outliers.