JAldrich_Final

Joyce Aldrich

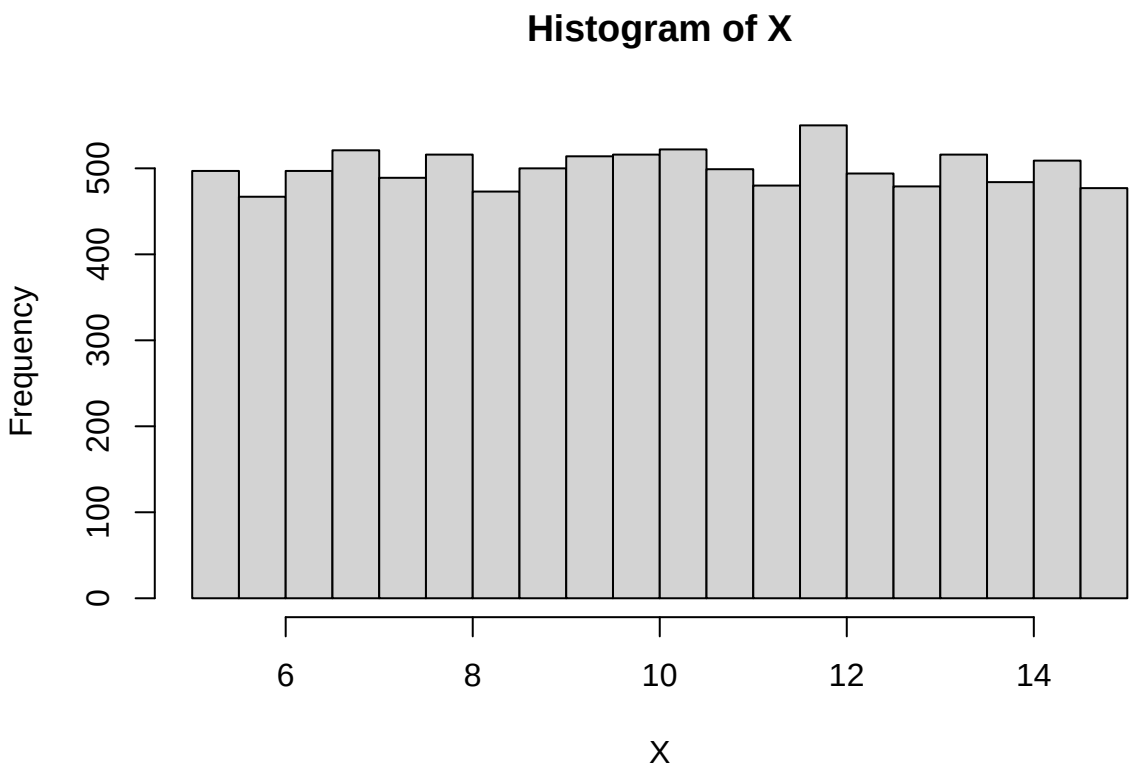
2023-12-14

#Problem 1.

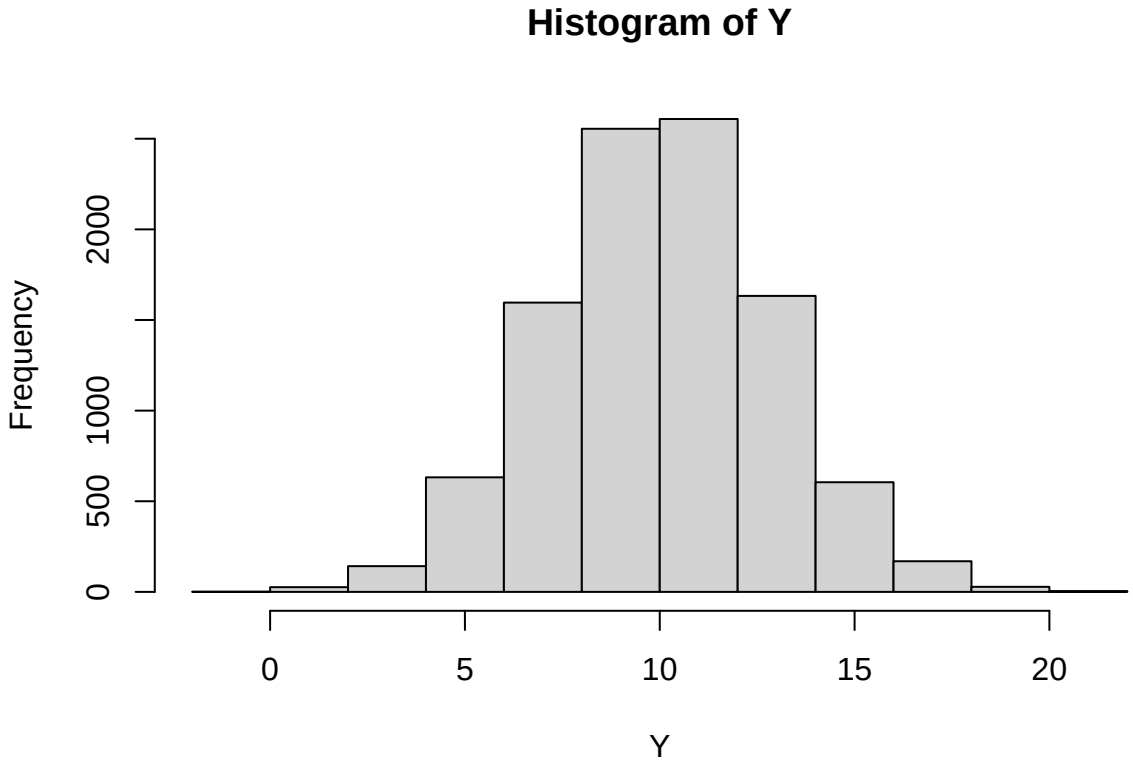
Using R, set a random seed equal to 1234 (i.e., set.seed(1234)). Generate a random variable X that has 10,000 continuous random uniform values between 5 and 15.Then generate a random variable Y that has 10,000 random normal values with a mean of 10 and a standard deviation of 2.89.

```
# Set the random seed as request
set.seed(1234)

# Generate random variable X with continuous uniform values between 5 and 15
X <- runif(10000, min = 5, max = 15)

# Generate random variable Y with normal values with a mean of 10 and std of 2.89
Y <- rnorm(10000, mean = 10, sd = 2.89)
```

```
hist(X)
```


Histogram of X
Frequency
0
100
200
300
400
500
6
8
10
12
14
X

```
hist(Y)
```


Histogram of Y
Frequency
0
500
1000
2000
0
5
10
15
20
Y

Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the median of the X variable, and the small letter "y" is estimated as the median of the Y variable. Interpret the meaning of all probabilities.

```
x <- median(X)
y <- median(Y)

x
```

```
## [1] 10.01266
```

```
y
```

```
## [1] 10.03154
```

a. P(X>x |X>y) : the conditional probability that X varible is greater than its median (x), given that X value is greater than Y variable's median (y).

$P(X > x|X > y) = \frac{P(X>x \cap X>y)}{P(X>y)}$

```
conditional_prob_a <- sum(X > max(x, y)) / sum(X > y)

print(conditional_prob_a)
```

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

b. P(X>x & Y>y) : the joint probability that X variable is greater than its median (x) and Y variable is greater than its median (y)

```
joint_prob_b <- sum(X > x & Y > y) / length(X)

print(joint_prob_b)
```

```
## [1] 0.2507
```

c. P(X<x | X>y) :the conditional probability that X varible is less than its median (x), given that X value is greater than Y variable's median (y).

$P(X < x|X > y) = \frac{P(X<x \cap X>y)}{P(X>y)}$

```
conditional_prob_c <- mean(X < x & X > y)

print(conditional_prob_c)
```

```
## [1] 0
```

Investigate whether P(X>x & Y>y)=P(X>x)P(Y>y) by building a table and evaluating the marginal and joint probabilities.

```r
library(data.table)

# Assuming X and Y are vectors or variables, and x, y are threshold values
Prob_X_x <- sum(X > x) / length(X)
Prob_Y_y <- sum(Y > y) / length(Y)
Prob_XY_x_y <- sum(X > x & Y > y) / length(X)

DT <- data.table(
  ID = c("P(X>x)", "P(Y>y)", "P(X>x)*P(Y>y)", "P((X>x) & (Y>y))"),
  X_x = c(Prob_X_x, Prob_Y_y, Prob_X_x * Prob_Y_y, Prob_XY_x_y),
  Y_y = c(Prob_Y_y, Prob_X_x, Prob_X_x * Prob_Y_y, Prob_XY_x_y)
)

DT
```

```
##                   ID    X_x    Y_y
## 1:            P(X>x) 0.5000 0.5000
## 2:            P(Y>y) 0.5000 0.5000
## 3:    P(X>x)*P(Y>y) 0.2500 0.2500
## 4: P((X>x) & (Y>y)) 0.2507 0.2507
```

The probability of the X value exceeding x and the Y value exceeding y is currently estimated at 0.2507. With a larger dataset, this value would likely converge toward 0.25. The calculation is derived from multiplying the individual probabilities, where P(X>x) = 0.5 and P(Y>y) = 0.5, resulting in 0.25 as the expected joint probability.

Check to see if independence holds by using Fisher's Exact Test and the Chi Square Test. What is the difference between the two? Which is most appropriate? Are you surprised at the results? Why or why not?

```r
# Create a contingency table
contingency_table <- table(X > x, Y > y)
```

```r
# Perform Fisher's Exact Test
fisher.test(contingency_table)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  contingency_table
## p-value = 0.7949
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.9342763 1.0946016
## sample estimates:
## odds ratio
##   1.011264
```

```r
# Perform Chi-Square Test
chisq.test(contingency_table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table
## X-squared = 0.0676, df = 1, p-value = 0.7949
```

Noticed that the Fisher's Exact test and the Chi-Square test are both used for testing the association between two categorical variables. However, they have different assumptions and are applicable in different scenarios. If dealing with a small sample size, Fisher's Exact Test is often preferred. For a large sample size, the Chi-square test is more practical and provides a good approximation. Noticed that both tests produce p-values which determine whether to reject the null hypothesis or not. If the p-value is less than 0.05, we usually reject the null hypothesis.

Based on the results of both tests above, the p-values and confidence intervals are the same for both tests. The p-values are above 0.05, which fails to reject the null hypothesis in these two tests, indicating that there is no statistically significant association between these two variables.

Because of the sample size and the assumptions of the tests being met, the Chi-square may be a reasonable choice for this scenario

#Problem 2.

You are to register for Kaggle.com (free) and compete in the Regression with a Crab Age Dataset competition. https://www.kaggle.com/competitions/playground-series-s3e16 I want you to do the following.

```r
# loading the library
library(readr)

#import the data
data <- read_csv("https://raw.githubusercontent.com/joyce-aldrich/DATA-605/main/train.csv")
```

```
## Rows: 74051 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Sex
## dbl (9): id, Length, Diameter, Height, Weight, Shucked Weight, Viscera Weigh...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Descriptive and Inferential Statistics. Provide univariate descriptive statistics and appropriate plots for the training data set. Provide a scatterplot matrix for at least two of the independent variables and the dependent variable. Derive a correlation matrix for any three quantitative variables in the dataset.

```r
# univariate descriptive statistics
summary(data)
```

```
##        id             Sex                Length          Diameter
##  Min.   :    0   Length:74051       Min.   :0.1875   Min.   :0.1375
##  1st Qu.:18512   Class :character   1st Qu.:1.1500   1st Qu.:0.8875
##  Median :37025   Mode  :character   Median :1.3750   Median :1.0750
##  Mean   :37025                      Mean   :1.3175   Mean   :1.0245
##  3rd Qu.:55538                      3rd Qu.:1.5375   3rd Qu.:1.2000
##  Max.   :74050                      Max.   :2.0128   Max.   :1.6125
##      Height           Weight        Shucked Weight     Viscera Weight
##  Min.   :0.0000   Min.   : 0.0567   Min.   : 0.02835   Min.   : 0.04252
##  1st Qu.:0.3000   1st Qu.:13.4377   1st Qu.: 5.71242   1st Qu.: 2.86330
##  Median :0.3625   Median :23.7994   Median : 9.90815   Median : 4.98951
##  Mean   :0.3481   Mean   :23.3852   Mean   :10.10427   Mean   : 5.05839
##  3rd Qu.:0.4125   3rd Qu.:32.1625   3rd Qu.:14.03300   3rd Qu.: 6.98815
##  Max.   :2.8250   Max.   :80.1015   Max.   :42.18406   Max.   :21.54562
##   Shell Weight           Age
##  Min.   : 0.04252   Min.   : 1.000
##  1st Qu.: 3.96893   1st Qu.: 8.000
##  Median : 6.93145   Median :10.000
##  Mean   : 6.72387   Mean   : 9.968
##  3rd Qu.: 9.07184   3rd Qu.:11.000
##  Max.   :28.49125   Max.   :29.000
```

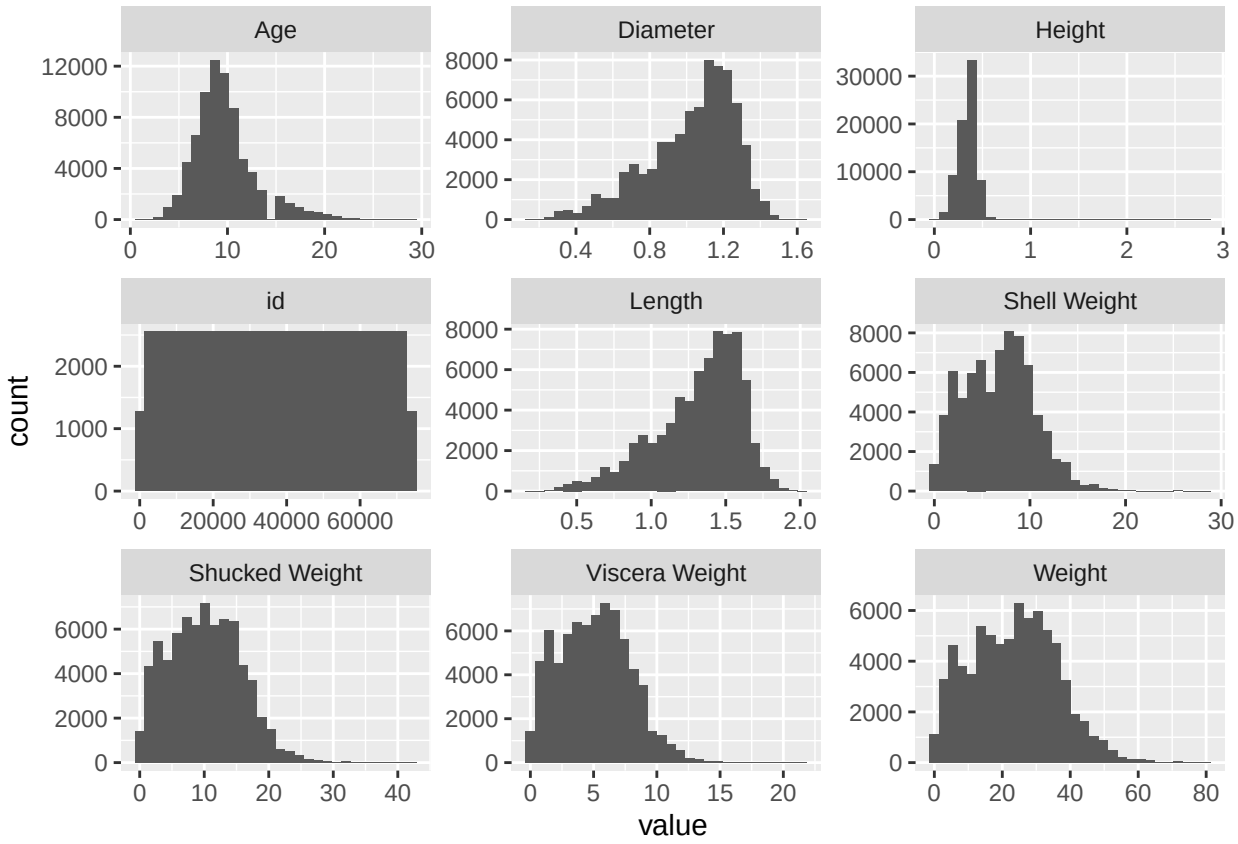
```r
library(ggplot2)
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:data.table':
##
##     transpose
```

```r
library(tidyr)

# appropriate plots for the training data set
data %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

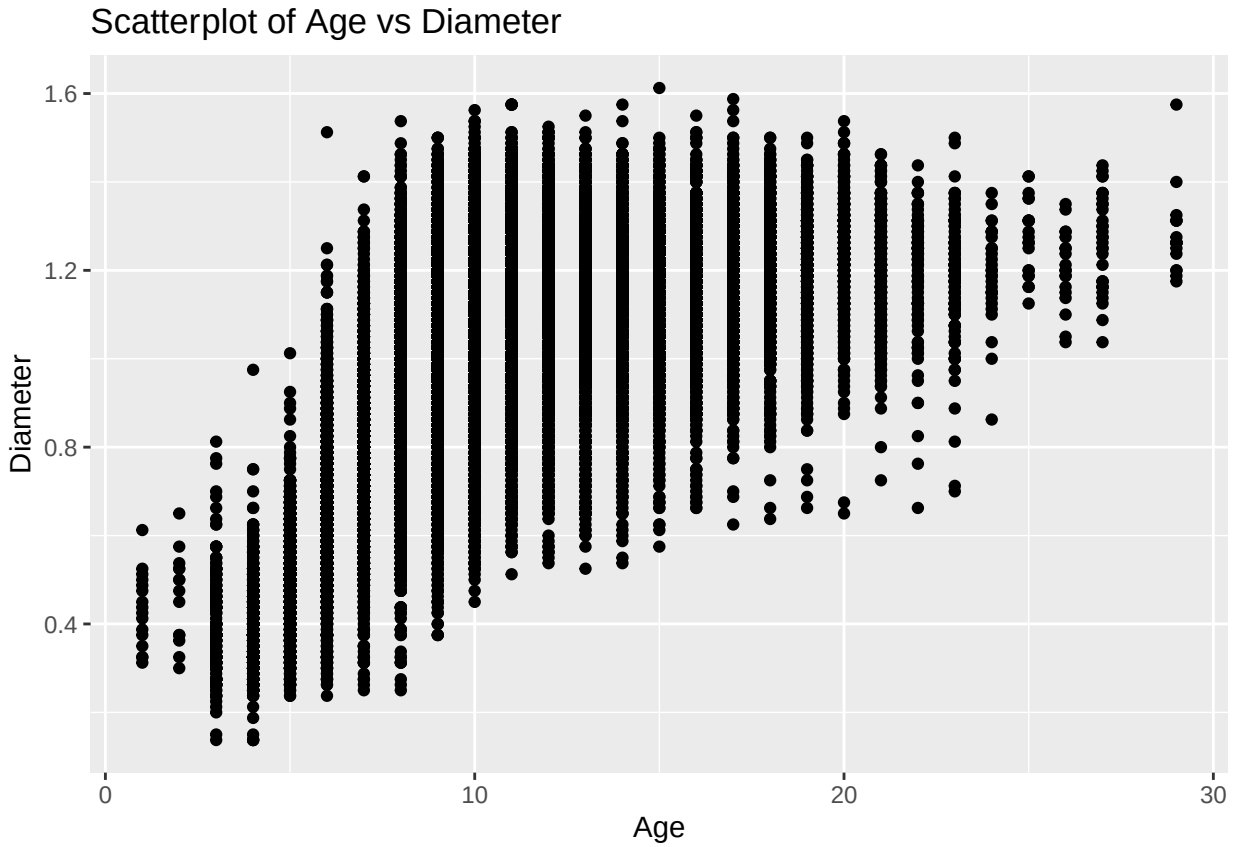
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Provide a scatterplot matrix for at least two of the independent variables and the dependent variable.

# Age vs Length
ggplot(data, aes(x=Age, y=Length)) +
    geom_point()+
     ggtitle("Scatterplot of Age vs Length")
```


```
# Age vs Weight
ggplot(data, aes(x=Age, y=Weight)) +
    geom_point() +
    ggtitle("Scatterplot of Age vs Weight")
```


```
# Age vs Diameter
ggplot(data, aes(x=Age, y=Diameter)) +
    geom_point() +
    ggtitle("Scatterplot of Age vs Diameter")
```



Derive a correlation matrix for any three quantitative variables in the dataset.

```
# Create the correlation matrix
corr_matrix <- cor(data [,c("Length","Weight","Diameter")])

print(corr_matrix)
```

```
##             Length    Weight  Diameter
## Length   1.0000000 0.9363738 0.9894374
## Weight   0.9363738 1.0000000 0.9382486
## Diameter 0.9894374 0.9382486 1.0000000
```

The correlation matrix shows that the pairwise correlations between Lenghth, Weight, and Diameter. The values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. The correlation matrix helps us understand the strength and direction of the relationships between variables.

Based on the output above, all the correlation coefficients are positive, indicating positive linear relationships between the variables. The strength of these relationships varies, with Length and Diameter having an exceptionally strong correlation, followed by Length and Weight, and Weight and Diameter.

Test the hypotheses that the correlations between each pairwise set of variables is 0 and provide an 80% confidence interval. Discuss the meaning of your analysis. Would you be worried about familywise error? Why or why not?

```r
#Correlation Tests
test_length_weight <- cor.test(data$Length, data$Weight, conf.level = 0.80)
test_length_diameter <- cor.test(data$Length, data$Diameter, conf.level = 0.80)
test_weight_diameter <- cor.test(data$Weight, data$Diameter, conf.level = 0.80)
```

```r
test_length_weight
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$Length and data$Weight
## t = 725.93, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.9357910 0.9369515
## sample estimates:
##       cor
## 0.9363738
```

```r
test_length_diameter
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$Length and data$Diameter
## t = 1857.4, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.9893379 0.9895359
## sample estimates:
##       cor
## 0.9894374
```

```r
test_weight_diameter
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$Weight and data$Diameter
## t = 737.99, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 80 percent confidence interval:
##  0.9376824 0.9388098
## sample estimates:
##       cor
## 0.9382486
```

Per the test results between these three variables-Length, Weight, and Diameter, the p-values of these three testing results are extremely small which means we can reject null hypothesis that the true correlation is equal to zero. The 80% confidence intervals provide a range of plausible values for the true correlation. The narrow intervals further support the strength and precision of the observed correlations.

We conducted only three tests, and all of them came out extremely very low p-values. The probability of all three results being false positives (Type I errors) is extremely unlikely. While it's important to be mindful of familywise errors, this wound not be in this particular situation.

Linear Algebra and Correlation. Invert your correlation matrix from above. (This is known as the precision matrix and contains variance inflation factors on the diagonal.) Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix. Conduct LDU decomposition on the matrix.

```
# Invert your correlation matrix from above
precision_matrix <- solve(corr_matrix)
precision_matrix
```

```
##              Length    Weight   Diameter
## Length    48.841860 -3.279081 -45.249368
## Weight    -3.279081  8.575097  -4.801128
## Diameter -45.249368 -4.801128  50.276067
```

```
# Multiply the correlation matrix by the precision matrix

corr_multi_pre <- corr_matrix %*% precision_matrix
corr_multi_pre
```

```
##                 Length       Weight Diameter
## Length    1.000000e+00 0.000000e+00        0
## Weight   -7.105427e-15 1.000000e+00        0
## Diameter  0.000000e+00 1.776357e-15        1
```

```
# multiply the precision matrix by the correlation matrix.

pre_multi_corr <- precision_matrix %*% corr_matrix
pre_multi_corr
```

```
##                 Length        Weight      Diameter
## Length    1.000000e+00 -1.421085e-14 -7.105427e-15
## Weight   -1.776357e-15  1.000000e+00  0.000000e+00
## Diameter  7.105427e-15  1.421085e-14  1.000000e+00
```

```
#Conduct LDU decomposition on the matrix
#install.packages("matrixcalc")
library(matrixcalc)

lu.decomposition(corr_matrix)
```

```
## $L
##           [,1]       [,2] [,3]
## [1,] 1.0000000 0.00000000    0
## [2,] 0.9363738 1.00000000    0
```

```
## [3,] 0.9894374 0.09549529    1
##
## $U
##      [,1]      [,2]       [,3]
## [1,]    1 0.9363738 0.98943735
## [2,]    0 0.1232041 0.01176541
## [3,]    0 0.0000000 0.01989018
```

Calculus-Based Probability & Statistics. Many times, it makes sense to fit a closed form distribution to data. Select a variable in the Kaggle.com training dataset that is skewed to the right, shift it so that the minimum value is absolutely above zero if necessary. Then load the MASS package and run fitdistr to fit an exponential probability density function. (See https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html). Find the optimal value of for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000,)). Plot a histogram and compare it with a histogram of your original variable. Using the exponential pdf, find the 5th and 95th percentiles using the cumulative distribution function (CDF). Also generate a 95% confidence interval from the empirical data, assuming normality. Finally, provide the empirical 5th percentile and 95th percentile of the data. Discuss.

```
#install.packages("MASS")
library(MASS)
```

Per our earlier histograms for the training data set, noticed that variable- Shell weight, Shucked Weight, Viscera Weight and Weight are right skewed. I would like to use a variable- shell weight as example in this section.

```
#install.packages("moments")
library(moments)

# Check skewness of the variable
skewness_shell_weight <- skewness(data$`Shell Weight`)
skewness_shucked_weight <- skewness(data$`Shucked Weight`)
skewness_viscera_weight <- skewness(data$`Viscera Weight`)
skewness_weight <- skewness(data$`Weight`)

skewness_shell_weight
```

```
## [1] 0.2774533
```

```
skewness_shucked_weight
```

```
## [1] 0.3494646
```

```
skewness_viscera_weight
```

```
## [1] 0.286377
```

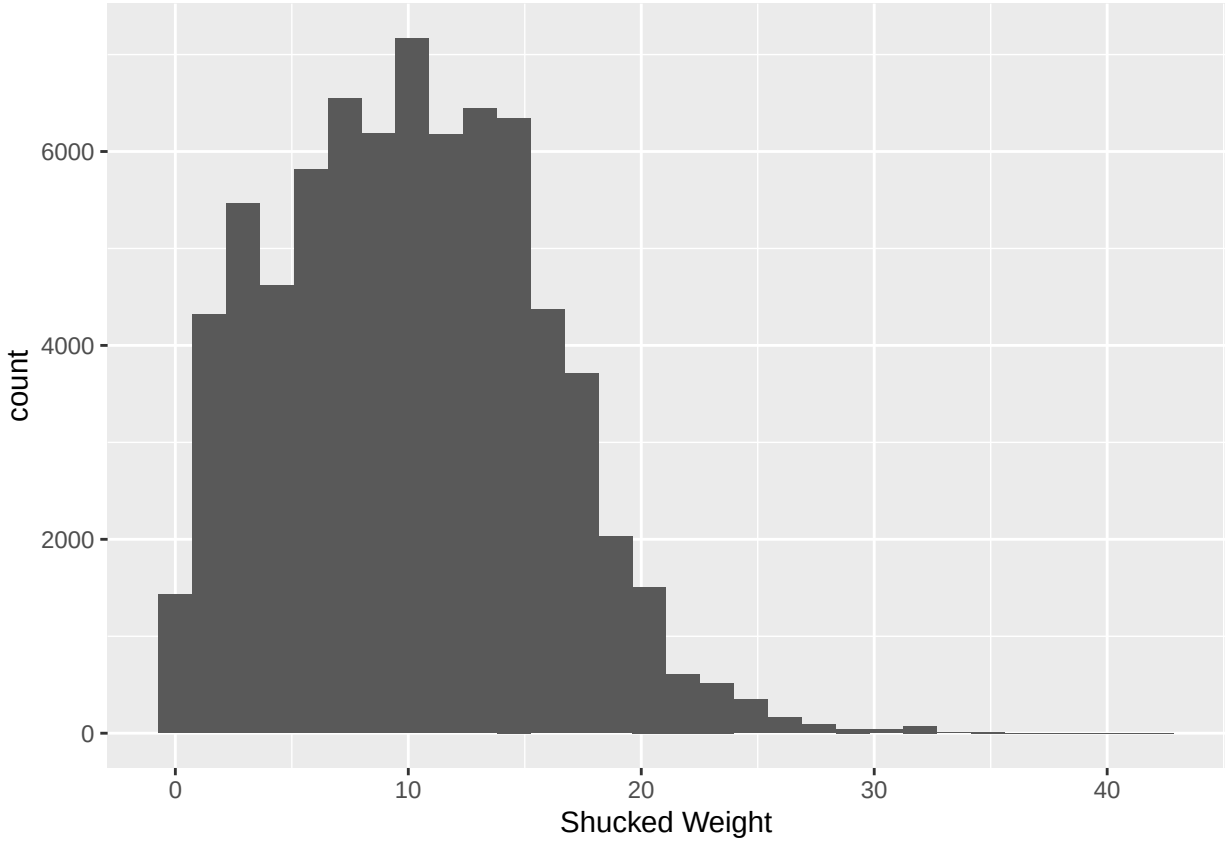
```
skewness_weight
```

```
## [1] 0.2314599
```

Based on the above outputs, a variavble - shucked weight is a right-skewed and a positive skewness value, and the largest value (0.3494646) indicates the most right-skewed distribution among the options.

```
data %>% ggplot(aes(x = `Shucked Weight`)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



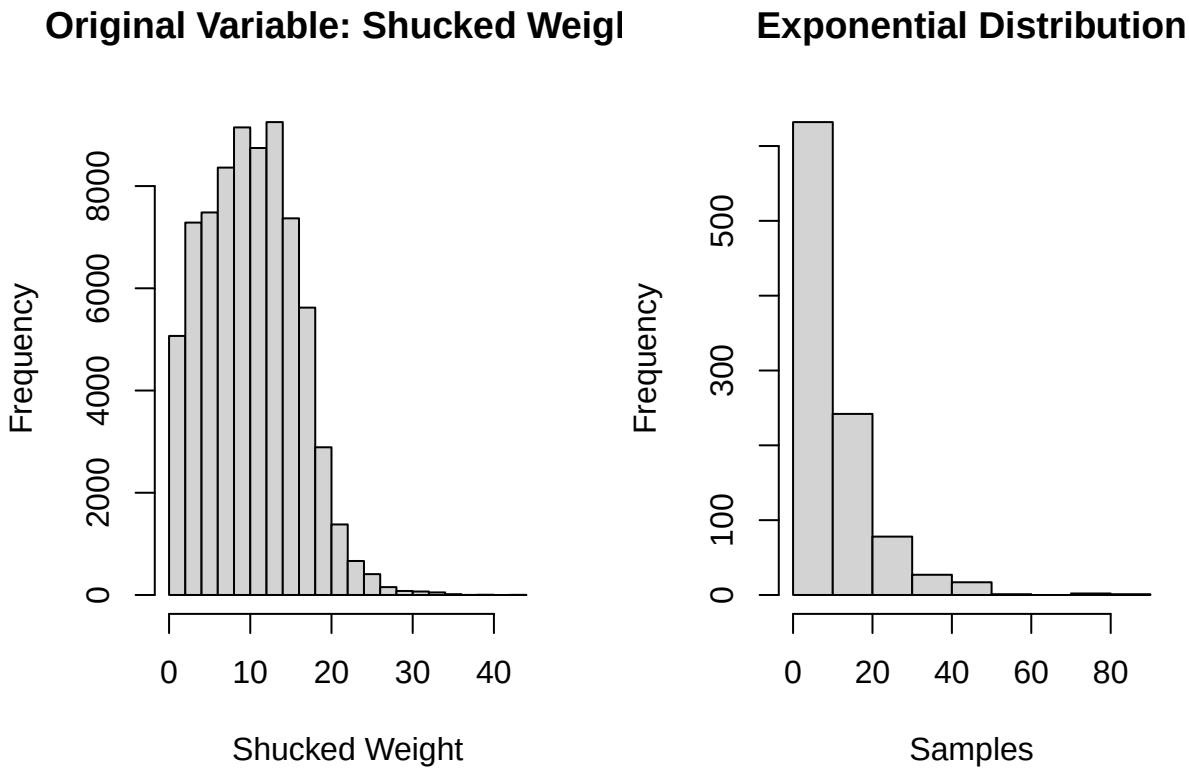

```
# Check Shift the variable if necessary
min_value <- min(data$`Shucked Weight`)

min_value
```

```
## [1] 0.0283495
```

```r
# run fitdistr function to fit an exponential probability density function
fit_expo_pd <- fitdistr(data$`Shucked Weight`, "exponential")

# the optimal value of \lambda
lambda <- fit_expo_pd$estimate
```

```r
# Generate 1000 samples from the exponential distribution
expo_shucked_pdf <- rexp(1000, lambda)
```

```r
# Plot histograms
par(mfrow = c(1, 2))
hist(data$`Shucked Weight`, main = "Original Variable: Shucked Weight", xlab = "Shucked Weight")
hist(expo_shucked_pdf, main = "Exponential Distribution", xlab = "Samples")
```



Noticed that the histograms above are differences. In addition, the actual exponential distribution is heavily concentrated on the left and displays significant right-skewness. The count decreases rapidly initially and then slows down.

Using the exponential pdf, find the 5th and 95th percentiles using the cumulative distribution function (CDF)

```r
# using the exponential pdf, find the 5th and 95th percentiles
percentile_5_expo= qexp(0.05, rate = lambda)
percentile_95_expo = qexp(0.95, rate = lambda)

percentile_5_expo
```

```
## [1] 0.5182813
```

```
percentile_95_expo
```

```
## [1] 30.26969
```

To generate a 95% confidence interval from the empirical data, assuming normality -> by using the t.test function.

```
# Calculate the confidence interval assuming normality
confidence_interval <- t.test(data$`Shucked Weight`)
confidence_interval
```

```
##
##  One Sample t-test
##
## data:  data$`Shucked Weight`
## t = 489.43, df = 74050, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  10.06381 10.14473
## sample estimates:
## mean of x
##  10.10427
```

```
# Calculate empirical percentiles
empirical_5th <- quantile(data$`Shucked Weight`, 0.05)
empirical_95th <- quantile(data$`Shucked Weight`, 0.95)

empirical_5th
```

```
##       5%
## 1.502523
```

```
empirical_95th
```

```
##      95%
## 19.37688
```

Conculsion: Based on the above result, the 95% confidence interval from the empirical data is 10.06381 to 10.14473. Per the above outputs of the 5th and 95th percentile of the distribution, it's approximately 0.5183 and 30.2697. For the empirical percentiles (quantiles) are calculated directly from the dataset, the empirical 5th percentile of Shucked Weight is approximately 1.5025, and the empirical 95th percentile is around 19.3769. Noticed that the empirical percentiles closely match the exponential distribution percentiles, it suggests that the data may be consistent with an exponential distribution.

Build some type of multiple regression model and submit your model to the competition board. Provide your complete model summary and results with analysis. Report your Kaggle.com user name and score.

```r
library(dplyr)
```

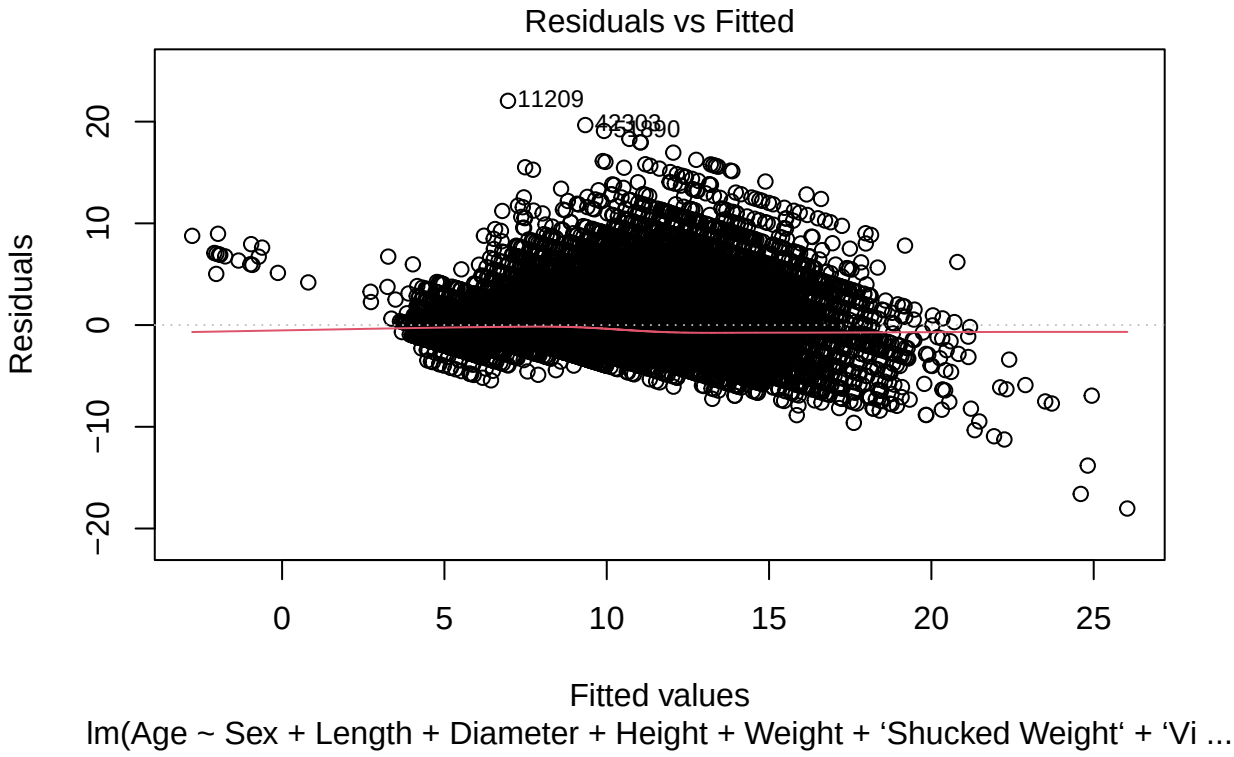
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

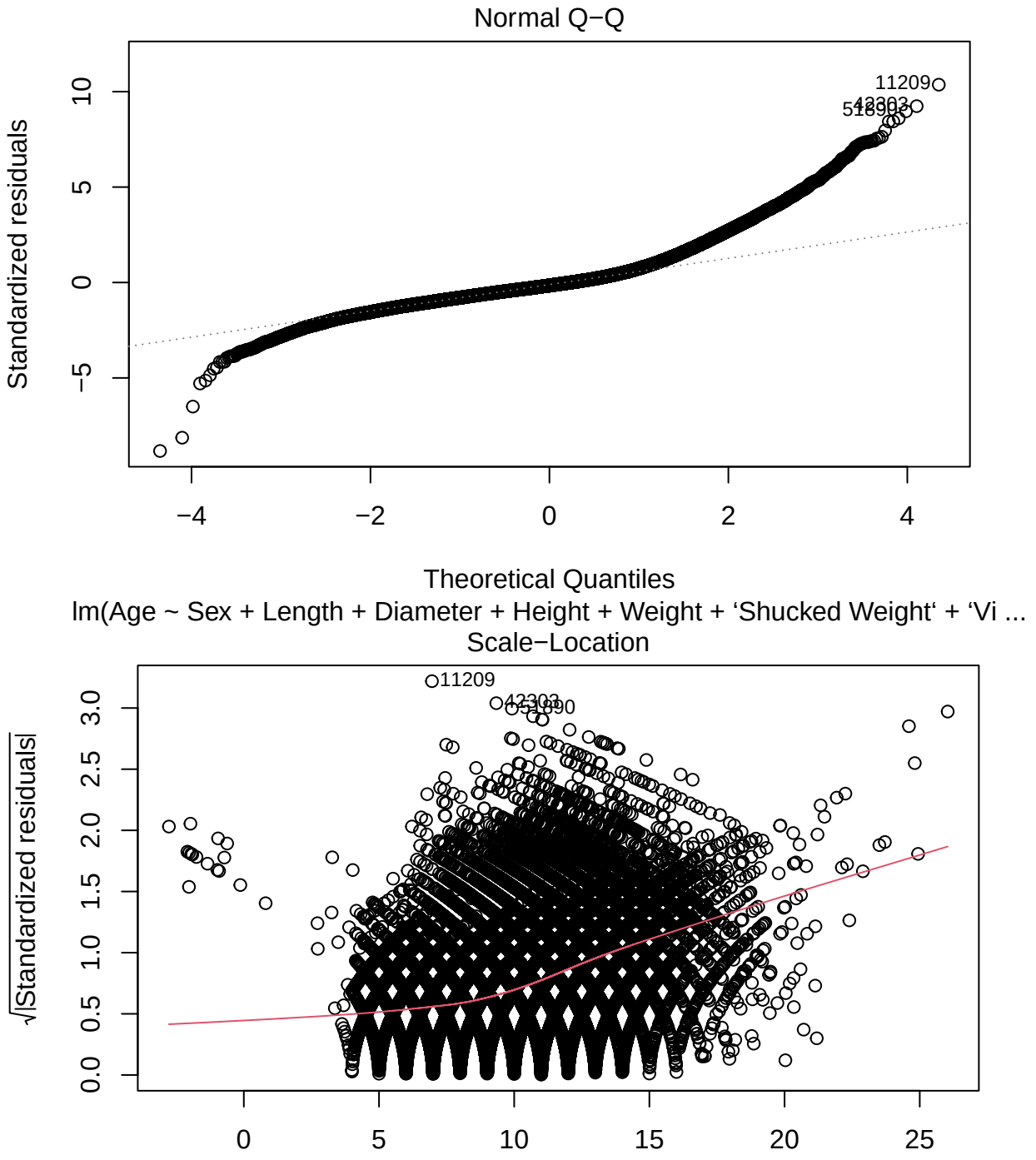
```r
# create a data_new by removing "id" variable from data
data_new <- subset(data, select = -id)

model_1 <- data_new %>%
  lm(Age ~ Sex+Length +Diameter + Height + Weight + `Shucked Weight`+`Viscera Weight`+`Shell Weight`, da

summary(model_1)
```

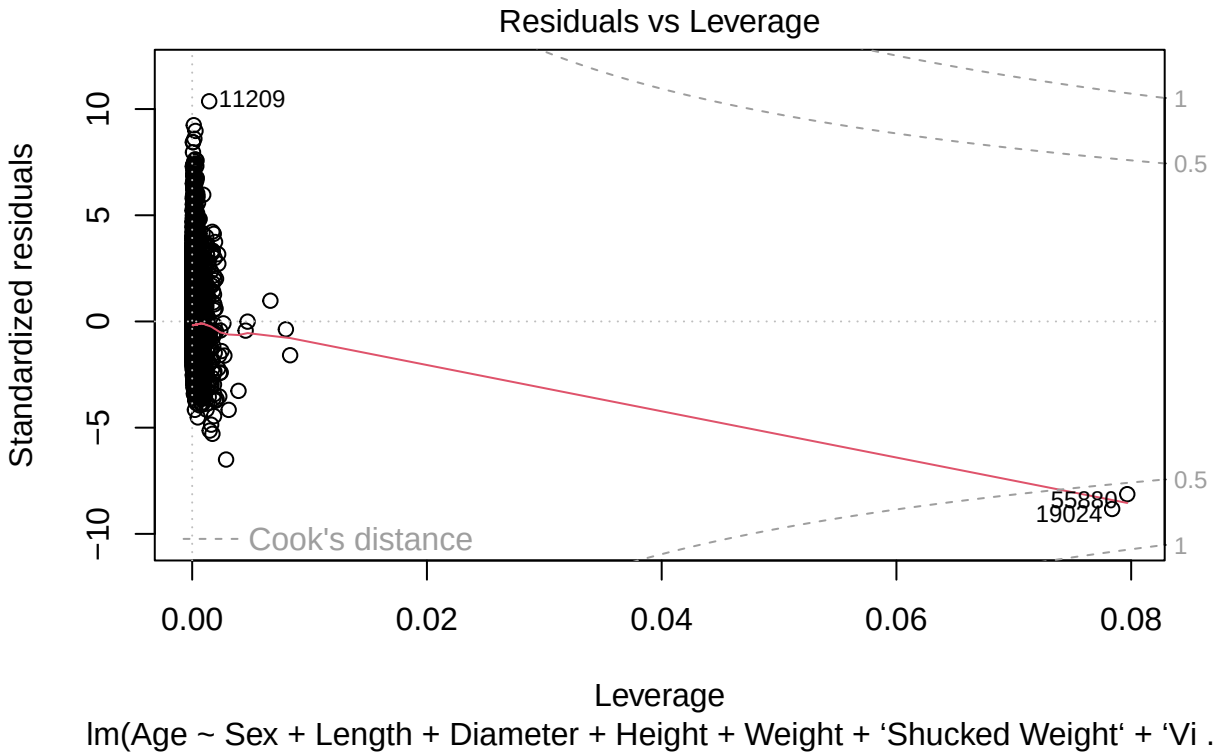
```
##
## Call:
## lm(formula = Age ~ Sex + Length + Diameter + Height + Weight +
##     `Shucked Weight` + `Viscera Weight` + `Shell Weight`, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0347  -1.2228  -0.3320   0.7537  22.0418
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.755758   0.075897  49.485  < 2e-16 ***
## SexI             -1.040735   0.025209 -41.284  < 2e-16 ***
## SexM             -0.115212   0.019157  -6.014 1.82e-09 ***
## Length            0.910211   0.192234   4.735 2.20e-06 ***
## Diameter          2.138507   0.238786   8.956  < 2e-16 ***
## Height            7.222272   0.236275  30.567  < 2e-16 ***
## Weight            0.194265   0.005416  35.867  < 2e-16 ***
## `Shucked Weight` -0.614115   0.006632 -92.603  < 2e-16 ***
## `Viscera Weight` -0.216351   0.011872 -18.224  < 2e-16 ***
## `Shell Weight`    0.512127   0.009820  52.152  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.128 on 74041 degrees of freedom
## Multiple R-squared:  0.5508, Adjusted R-squared:  0.5508
## F-statistic: 1.009e+04 on 9 and 74041 DF,  p-value: < 2.2e-16
```

```
plot(model_1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Age ~ Sex + Length + Diameter + Height + Weight + ‘Shucked Weight‘ + ‘Vi ...
Scale−Location
√|Standardized residuals|
Fitted values
lm(Age ~ Sex + Length + Diameter + Height + Weight + ‘Shucked Weight‘ + ‘Vi ...



The R-squared value of 0.5508 indicates that the model explains about 55.08% of the variance in the dependent variable 'Age.' This means that the model captures a moderate amount of the variability in 'Age' based on the chosen independent variables in the model. In addition, the low p-values associated with each variable suggest that all the variables in the model are statistically significant. Specifically, variables like 'Sex', 'Length', 'Diameter' and so on all have p-values close to zero in the model summary above.

Build a regression model from a set of predictor variables by entering and removing predictors in a stepwise manner until there is no statistically valid reason to enter or remove any more.

The goal of stepwise regression is to construct a model that incorporates all predictor variables that are statistically significantly related to the response variable.

I would like to use stepwise regression to confirm that the initiated model is reasonable.

There are three ways to perform stepwise regression: Forward Stepwise Selection, Backward Stepwise Selection, and Both-Direction Stepwise Selection.

Forward Stepwise Selection

```
null <- lm(Age ~ 1, data = data_new)
full <- lm(Age ~ ., data = data_new)


forward.lm <- step(null,
                   scope=list(lower=null, upper=full),
                   direction="forward")
```

```
## Start:  AIC=171113.2
## Age ~ 1
##
```

```
##                     Df Sum of Sq    RSS    AIC
## + `Shell Weight`     1    328633 417926 128153
## + Height             1    303946 442613 132403
## + Diameter           1    288141 458418 135001
## + Length             1    280390 466169 136243
## + Weight             1    269833 476726 137901
## + `Viscera Weight`   1    248386 498174 141160
## + Sex                2    201176 545384 147866
## + `Shucked Weight`   1    189127 557432 149482
## <none>                           746559 171113
##
## Step:  AIC=128153
## Age ~ `Shell Weight`
##
##                     Df Sum of Sq    RSS    AIC
## + `Shucked Weight`   1     44230 373696 119872
## + Weight             1     17107 400819 125060
## + Sex                2     11246 406680 126137
## + `Viscera Weight`   1     10713 407214 126232
## + Height             1      6077 411850 127070
## + Diameter           1       414 417512 128082
## + Length             1        94 417833 128138
## <none>                           417926 128153
##
## Step:  AIC=119871.6
## Age ~ `Shell Weight` + `Shucked Weight`
##
##                     Df Sum of Sq    RSS    AIC
## + Diameter           1   17812.6 355884 116257
## + Height             1   17150.7 356546 116395
## + Length             1   16637.2 357059 116501
## + Sex                2   15818.8 357877 116673
## + Weight             1    9150.4 364546 118038
## + `Viscera Weight`   1    1273.6 372423 119621
## <none>                           373696 119872
##
## Step:  AIC=116257
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter
##
##                     Df Sum of Sq    RSS    AIC
## + Sex                2    9963.5 345920 114158
## + Height             1    5651.9 350232 115073
## + Weight             1    5644.8 350239 115075
## + Length             1     310.1 355574 116194
## + `Viscera Weight`   1     159.7 355724 116226
## <none>                           355884 116257
##
## Step:  AIC=114158.2
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex
##
##                     Df Sum of Sq    RSS    AIC
## + Weight             1    4820.6 341100 113121
## + Height             1    4655.2 341265 113157
## + Length             1     296.8 345623 114097
```

```
## + `Viscera Weight`  1      20.7 345900 114156
## <none>                          345920 114158
##
## Step:  AIC=113121
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight
##
##                    Df Sum of Sq    RSS    AIC
## + Height            1    4192.3 336907 112207
## + `Viscera Weight`  1    1250.7 339849 112851
## + Length            1     220.4 340879 113075
## <none>                          341100 113121
##
## Step:  AIC=112207.3
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
##     Height
##
##                    Df Sum of Sq    RSS    AIC
## + `Viscera Weight`  1   1469.59 335438 111886
## + Length            1     66.95 336840 112195
## <none>                          336907 112207
##
## Step:  AIC=111885.6
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
##     Height + `Viscera Weight`
##
##          Df Sum of Sq    RSS    AIC
## + Length  1    101.54 335336 111865
## <none>                335438 111886
##
## Step:  AIC=111865.1
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
##     Height + `Viscera Weight` + Length
```

```
summary(forward.lm)
```

```
##
## Call:
## lm(formula = Age ~ `Shell Weight` + `Shucked Weight` + Diameter +
##     Sex + Weight + Height + `Viscera Weight` + Length, data = data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0347  -1.2228  -0.3320   0.7537  22.0418
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.755758   0.075897  49.485  < 2e-16 ***
## `Shell Weight`    0.512127   0.009820  52.152  < 2e-16 ***
## `Shucked Weight` -0.614115   0.006632 -92.603  < 2e-16 ***
## Diameter          2.138507   0.238786   8.956  < 2e-16 ***
## SexI             -1.040735   0.025209 -41.284  < 2e-16 ***
## SexM             -0.115212   0.019157  -6.014 1.82e-09 ***
## Weight            0.194265   0.005416  35.867  < 2e-16 ***
## Height            7.222272   0.236275  30.567  < 2e-16 ***
```

```
## `Viscera Weight` -0.216351   0.011872 -18.224  < 2e-16 ***
## Length            0.910211   0.192234   4.735 2.20e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.128 on 74041 degrees of freedom
## Multiple R-squared:  0.5508, Adjusted R-squared:  0.5508
## F-statistic: 1.009e+04 on 9 and 74041 DF,  p-value: < 2.2e-16
```

Overall, all the variables in the model are statistically significant. None of them has been dropped, as indicated by the values in the AIC (Akaike Information Criterion) column. AIC is a measure of the relative quality of statistical models, where lower values indicate better-fitting models. Notably, the AIC value decreases when additional variables are added.

Backward stepwsie regression

```
backward.lm <- step(full,
                    scope = list(upper=full),
                    direction="backward")
```

```
## Start:  AIC=111865.1
## Age ~ Sex + Length + Diameter + Height + Weight + `Shucked Weight` +
##     `Viscera Weight` + `Shell Weight`
##
##                     Df Sum of Sq    RSS    AIC
## <none>                           335336 111865
## - Length             1       102 335438 111886
## - Diameter           1       363 335700 111943
## - `Viscera Weight`   1      1504 336840 112195
## - Height             1      4232 339568 112792
## - Weight             1      5826 341163 113139
## - Sex                2      8658 343994 113749
## - `Shell Weight`     1     12318 347654 114535
## - `Shucked Weight`   1     38838 374174 119978
```

```
summary(backward.lm)
```

```
##
## Call:
## lm(formula = Age ~ Sex + Length + Diameter + Height + Weight +
##     `Shucked Weight` + `Viscera Weight` + `Shell Weight`, data = data_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0347  -1.2228  -0.3320   0.7537  22.0418
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.755758   0.075897  49.485  < 2e-16 ***
## SexI             -1.040735   0.025209 -41.284  < 2e-16 ***
## SexM             -0.115212   0.019157  -6.014 1.82e-09 ***
```

```
## Length            0.910211   0.192234   4.735 2.20e-06 ***
## Diameter          2.138507   0.238786   8.956  < 2e-16 ***
## Height            7.222272   0.236275  30.567  < 2e-16 ***
## Weight            0.194265   0.005416  35.867  < 2e-16 ***
## `Shucked Weight` -0.614115   0.006632 -92.603  < 2e-16 ***
## `Viscera Weight` -0.216351   0.011872 -18.224  < 2e-16 ***
## `Shell Weight`    0.512127   0.009820  52.152  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.128 on 74041 degrees of freedom
## Multiple R-squared:  0.5508, Adjusted R-squared:  0.5508
## F-statistic: 1.009e+04 on 9 and 74041 DF,  p-value: < 2.2e-16
```

The outcome is the same as Forward Stepwise Selection.

Both-Direction Stepwise Selection.

```
step(null, scope = list(upper=full), direction="both")
```

```
## Start:  AIC=171113.2
## Age ~ 1
##
##                     Df Sum of Sq    RSS    AIC
## + `Shell Weight`     1    328633 417926 128153
## + Height             1    303946 442613 132403
## + Diameter           1    288141 458418 135001
## + Length             1    280390 466169 136243
## + Weight             1    269833 476726 137901
## + `Viscera Weight`   1    248386 498174 141160
## + Sex                2    201176 545384 147866
## + `Shucked Weight`   1    189127 557432 149482
## <none>                           746559 171113
##
## Step:  AIC=128153
## Age ~ `Shell Weight`
##
##                     Df Sum of Sq    RSS    AIC
## + `Shucked Weight`   1     44230 373696 119872
## + Weight             1     17107 400819 125060
## + Sex                2     11246 406680 126137
## + `Viscera Weight`   1     10713 407214 126232
## + Height             1      6077 411850 127070
## + Diameter           1       414 417512 128082
## + Length             1        94 417833 128138
## <none>                           417926 128153
## - `Shell Weight`     1    328633 746559 171113
##
## Step:  AIC=119871.6
## Age ~ `Shell Weight` + `Shucked Weight`
##
##                     Df Sum of Sq    RSS    AIC
```

```
## + Diameter          1     17813 355884 116257
## + Height            1     17151 356546 116395
## + Length            1     16637 357059 116501
## + Sex               2     15819 357877 116673
## + Weight            1      9150 364546 118038
## + `Viscera Weight`  1      1274 372423 119621
## <none>                          373696 119872
## - `Shucked Weight`  1     44230 417926 128153
## - `Shell Weight`    1    183736 557432 149482
##
## Step:  AIC=116257
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter
##
##                     Df Sum of Sq    RSS    AIC
## + Sex                2      9963 345920 114158
## + Height             1      5652 350232 115073
## + Weight             1      5645 350239 115075
## + Length             1       310 355574 116194
## + `Viscera Weight`   1       160 355724 116226
## <none>                          355884 116257
## - Diameter           1     17813 373696 119872
## - `Shucked Weight`   1     61629 417512 128082
## - `Shell Weight`     1     83547 439430 131870
##
## Step:  AIC=114158.2
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex
##
##                     Df Sum of Sq    RSS    AIC
## + Weight             1      4821 341100 113121
## + Height             1      4655 341265 113157
## + Length             1       297 345623 114097
## + `Viscera Weight`   1        21 345900 114156
## <none>                          345920 114158
## - Sex                2      9963 355884 116257
## - Diameter           1     11957 357877 116673
## - `Shucked Weight`   1     60745 406665 126136
## - `Shell Weight`     1     76338 422258 128923
##
## Step:  AIC=113121
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight
##
##                     Df Sum of Sq    RSS    AIC
## + Height             1      4192 336907 112207
## + `Viscera Weight`   1      1251 339849 112851
## + Length             1       220 340879 113075
## <none>                          341100 113121
## - Weight             1      4821 345920 114158
## - Sex                2      9139 350239 115075
## - Diameter           1      9606 350706 115176
## - `Shell Weight`     1     16678 357778 116654
## - `Shucked Weight`   1     39540 380639 121241
##
## Step:  AIC=112207.3
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
```

```
##     Height
##
##                     Df Sum of Sq    RSS    AIC
## + `Viscera Weight`  1      1470 335438 111886
## + Length            1        67 336840 112195
## <none>                          336907 112207
## - Diameter          1      3227 340135 112911
## - Height            1      4192 341100 113121
## - Weight            1      4358 341265 113157
## - Sex               2      8271 345178 113999
## - `Shell Weight`    1     13254 350162 115063
## - `Shucked Weight`  1     38142 375049 120147
##
## Step:  AIC=111885.6
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
##     Height + `Viscera Weight`
##
##                     Df Sum of Sq    RSS    AIC
## + Length            1       102 335336 111865
## <none>                          335438 111886
## - `Viscera Weight`  1      1470 336907 112207
## - Diameter          1      3491 338929 112650
## - Height            1      4411 339849 112851
## - Weight            1      5827 341265 113159
## - Sex               2      8643 344081 113765
## - `Shell Weight`    1     12242 347680 114538
## - `Shucked Weight`  1     38771 374209 119983
##
## Step:  AIC=111865.1
## Age ~ `Shell Weight` + `Shucked Weight` + Diameter + Sex + Weight +
##     Height + `Viscera Weight` + Length
##
##                     Df Sum of Sq    RSS    AIC
## <none>                          335336 111865
## - Length            1       102 335438 111886
## - Diameter          1       363 335700 111943
## - `Viscera Weight`  1      1504 336840 112195
## - Height            1      4232 339568 112792
## - Weight            1      5826 341163 113139
## - Sex               2      8658 343994 113749
## - `Shell Weight`    1     12318 347654 114535
## - `Shucked Weight`  1     38838 374174 119978


##
## Call:
## lm(formula = Age ~ `Shell Weight` + `Shucked Weight` + Diameter +
##     Sex + Weight + Height + `Viscera Weight` + Length, data = data_new)
##
## Coefficients:
##      (Intercept)    `Shell Weight`  `Shucked Weight`          Diameter
##           3.7558            0.5121           -0.6141            2.1385
##             SexI              SexM            Weight            Height
##          -1.0407           -0.1152            0.1943            7.2223
## `Viscera Weight`            Length
```

```
##          -0.2164            0.9102
```

```
step(full, scope = list(upper=full), direction="both")
```

```
## Start:  AIC=111865.1
## Age ~ Sex + Length + Diameter + Height + Weight + `Shucked Weight` +
##     `Viscera Weight` + `Shell Weight`
##
##                     Df Sum of Sq    RSS    AIC
## <none>                           335336 111865
## - Length             1       102 335438 111886
## - Diameter           1       363 335700 111943
## - `Viscera Weight`   1      1504 336840 112195
## - Height             1      4232 339568 112792
## - Weight             1      5826 341163 113139
## - Sex                2      8658 343994 113749
## - `Shell Weight`     1     12318 347654 114535
## - `Shucked Weight`   1     38838 374174 119978
```

```
##
## Call:
## lm(formula = Age ~ Sex + Length + Diameter + Height + Weight +
##     `Shucked Weight` + `Viscera Weight` + `Shell Weight`, data = data_new)
##
## Coefficients:
##      (Intercept)              SexI              SexM            Length
##           3.7558           -1.0407           -0.1152            0.9102
##         Diameter            Height            Weight  `Shucked Weight`
##           2.1385            7.2223            0.1943           -0.6141
## `Viscera Weight`    `Shell Weight`
##          -0.2164            0.5121
```

Conclusion: The results of these three ways to perform stepwise regression are confirmed the initiated model is reasonable and all variables are statistically significant in the model.

Model selection and its result

```
#import the data
test <- read_csv("https://raw.githubusercontent.com/joyce-aldrich/DATA-605/main/test.csv")
```

```
## Rows: 49368 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Sex
## dbl (8): id, Length, Diameter, Height, Weight, Shucked Weight, Viscera Weigh...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Use the model to make predictions on the test data
predictions_age <- predict(model_1, newdata = test)
```

```r
id <- test$id
id <- as.integer(id)
```

```r
# Combine the data frames using cbind

submission <- data.frame(id, predictions_age)

#colnames(submission)[1] <- "id"
colnames(submission)[2] <- "Age"


head(submission)
```

```
##      id       Age
## 1 74051  7.735472
## 2 74052  7.682365
## 3 74053 10.432966
## 4 74054  9.556131
## 5 74055  7.504141
## 6 74056 12.208548
```

Kaggle.com User name and Score.

```r
# Write the predicted Age of Result to a csv file for submission to the Kaggle.com.
write.csv(submission, file = "jaldrich_prediction.csv", row.names = FALSE)
```

The resulting score from this model is Private score: 1.48236 and Public Score: 1.48539. (User name: Joyce Aldrich)

The link of submission on Kaggle.com is below: https://www.kaggle.com/competitions/playground-series-s3e16/submissions