Computational Mathematics

Fredrick Jones

2023-12-15

Problem 1

Task1

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.

Generating 10,000 uniform numbers between 5 and 15. Also, generating 10,000 numbers from normal distribution with mean 10 and standard deviation 2.89

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(openintro)
```

```
## Loading required package: airports
## Loading required package: cherryblossom
## Loading required package: usdata
```

```
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

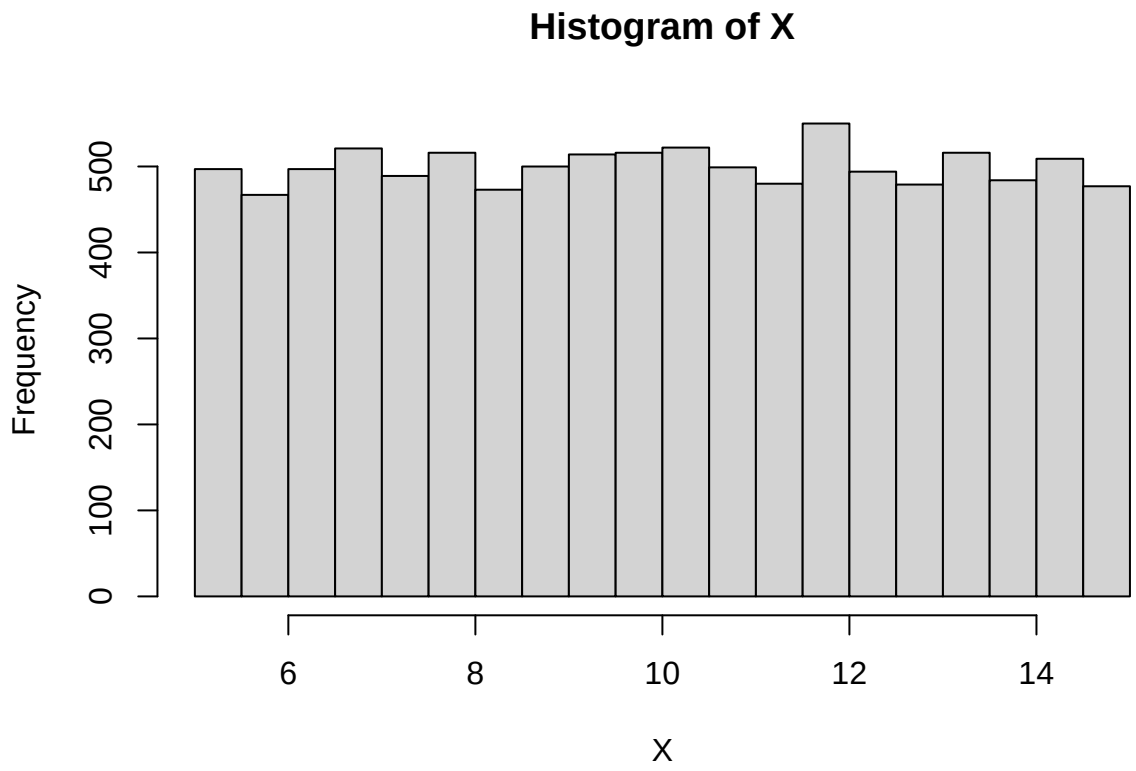
```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following objects are masked from 'package:openintro':
##
##     housing, mammals
##
## The following object is masked from 'package:dplyr':
##
##     select
```

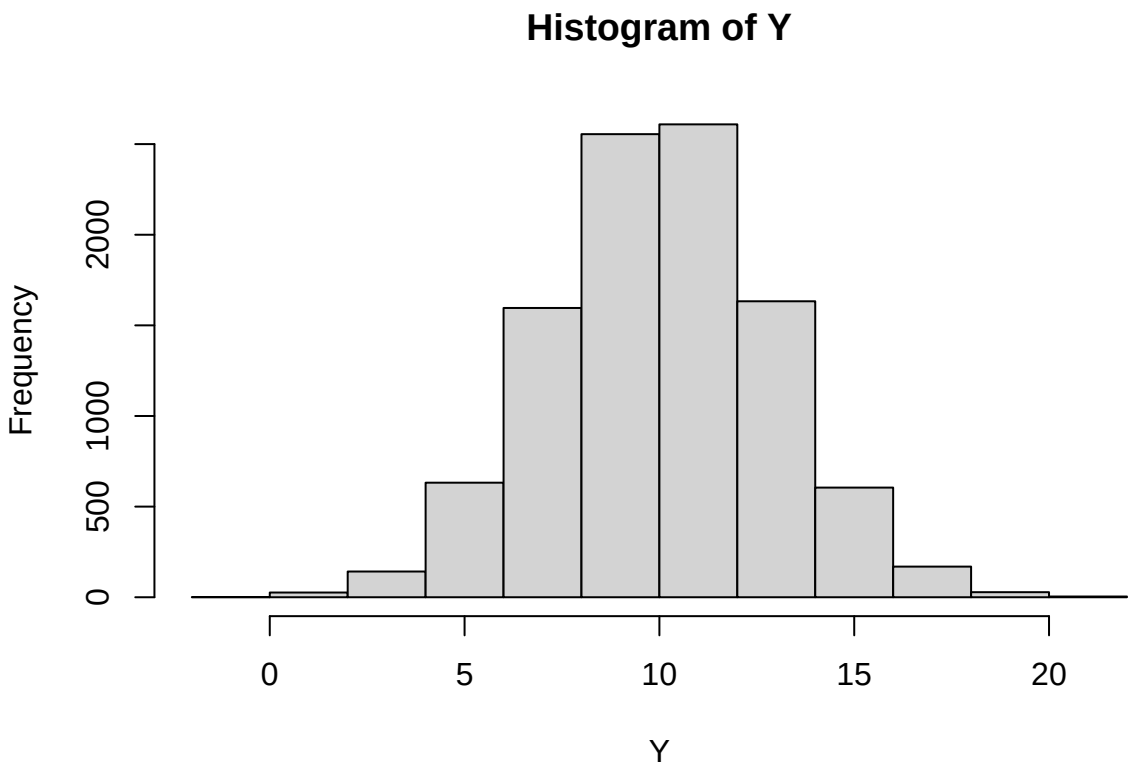
```
set.seed(1234)
X<- runif(10000, min = 5, max = 15)
Y <- rnorm(10000, mean=10, sd=2.89)
```

Let's visulize the data so genrated

```
hist(X)
```




```
hist(Y)
```



a. P(X>x| X>y) where x and y are medians of X and Y respectively

```
x<- median(X)
y<- median(Y)
data<- tibble(X_greater_x_or_y= ifelse(X>x|X>y, "yes", "no"),
              X=X)
pr <- data %>%
  count(X_greater_x_or_y)%>%
  mutate(p_hat = n/sum(n))
pr
```

```
## # A tibble: 2 x 3
##   X_greater_x_or_y     n p_hat
##   <chr>            <int> <dbl>
## 1 no                5000   0.5
## 2 yes               5000   0.5
```

It can be observed that P(X>x | X>y) = 0.5

Interpretation: The probability suggests that out of 10000 uniformly generated numbers between 5 and 15, 50% i.e. 5000 are greater than median of X or greater than median of Y.

b. P(X>x & Y>y)

```
data2<- tibble(X_gr_x_and_Y_gr_y= ifelse(X>x & Y>y, "yes", "no"),
              X=X, Y=Y)
```

```
pr <- data2 %>%
  count(X_gr_x_and_Y_gr_y)%>%
  mutate(p_hat = n/sum(n))
pr
```

```
## # A tibble: 2 x 3
##   X_gr_x_and_Y_gr_y     n p_hat
##   <chr>             <int> <dbl>
## 1 no                 7493 0.749
## 2 yes                2507 0.251
```

It can be seen that the probability P(X>x & Y>y)= 0.2507

Interpretation: It is observed that in 10,000 cases of X and Y, there are 2507 or 25.07% cases in which X is greater than x and Y is greater than Y.

c. P(X<x | X>y)

```
data3<- tibble(X_less_x_or_X_gr_y= ifelse(X<x | X>y, "yes", "no"),
               X=X)
pr <- data3 %>%
  count(X_less_x_or_X_gr_y)%>%
  mutate(p_hat = n/sum(n))
pr
```

```
## # A tibble: 2 x 3
##   X_less_x_or_X_gr_y     n  p_hat
##   <chr>              <int>  <dbl>
## 1 no                    21 0.0021
## 2 yes                 9979 0.998
```

It can be seen that P(X<x | X>y) = 0.9979

Interpretation: The probability suggests that for X: uniform distribution between 5 and 15 and Y: normal distribution with mean =10 and standard deviation = 2.89; 99.79% generated uniform random numbers are either less than the median of X or greater than the median of Y.

Task 2

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

Joint Table

```
j_table <- table(X>x, Y>y)
j_table
```

```
##
##         FALSE TRUE
##   FALSE  2507 2493
##   TRUE   2493 2507
```

Marginal probabilities

```r
mar_X <- rowSums(j_table)/sum(j_table)
mar_Y <- colSums(j_table)/sum(j_table)
cat("Marginal probability of X>x : ", mar_X)
```

```
## Marginal probability of X>x :  0.5 0.5
```

```r
cat("\nMarginal probability of Y>y:", mar_Y)
```

```
##
## Marginal probability of Y>y: 0.5 0.5
```

Product of marginal probabilities

```r
product_marginal <- outer(mar_X, mar_Y)
product_marginal
```

```
##       FALSE TRUE
## FALSE  0.25 0.25
## TRUE   0.25 0.25
```

It can be observed that the p(X>x)*p(Y>y) = 0.25, that is the product of marginal probabilities is 0.25 from part from the part b. the probability p(X>x & Y>y) is 0.2507 which is almost the equal.

So, we can conclude that p(X>x & Y>y) = p(X>x)*p(Y>y)

Thus, even X>x and Y>y are two independent events.

Task 3

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?

Fisher's Exact test

```r
fisher_output <- fisher.test(j_table)
fisher_output
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  j_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
```

Interpretation of Fisher's exact test p-value (0.7949):

This is the probability of observing a table as extreme as the calculated table, assuming the null hypothesis of independence is true. A higher p-value indicates weaker evidence against the null hypothesis.

Odds Ratio (1.011264): Odds Ratio measures the relationship between X and Y. A value of 1 means no relationship. The 95% confidence interval indicates the range within which you can be reasonably confident of the true odds ratio.

Chi-Square Test

```
chi_sq_output<- chisq.test(j_table)
chi_sq_output
```

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

Chi-square test interpretation p-value (0.7949): Similar to Fisher's exact test, this p-value represents the probability of observing the table assuming independence.

Chi-square statistic (0.0676): This statistic measures the difference between the observed and expected frequencies. Lower values indicate that the observed and expected frequencies are more similar.

Difference between Fisher's exact test and chi-square test:

Fisher's exact test: This test is suitable when the sample size is small and provides an accurate probability of observing the data assuming independence.

Which to choose:

Chi-square test: suitable for larger samples and based on asymptotic theory. The chi-square test is an approximation, but the larger the dataset, the less computationally intensive it is.

Which test is best?

When working with small sample sizes or categorical data with a small number of cells, Fisher's exact test is a better choice. When the sample size is large, the chi-square test is a valid and computationally efficient option. Surprise in results: Given the high p-values (0.7949) for both tests, there is not enough evidence to reject the null hypothesis of independence.

This suggests that there is no significant relationship between X and Y based on the observed data. Lack of surprise: These results are consistent with expectations if the variables were expected to be independent or if there is no theoretical reason to expect the variables to be dependent.

Statistical tests can only provide evidence against the null hypothesis.

They are unable to prove their independence. Lack of evidence for independence does not necessarily mean that the variables are independent. Still, it does indicate that there is not enough evidence to claim dependence based on the observed data.

Question 2

Read the training dataset from the disc. The dataset was downloaded from kaggle.com

```
df_train <- read.csv('https://raw.githubusercontent.com/jewelercart/605/main/train.csv')
head(df_train, 3)
```

```
##   id Sex Length Diameter Height   Weight Shucked.Weight Viscera.Weight
## 1  0   I 1.5250   1.1750  0.375 28.97319      12.728926       6.647958
## 2  1   I 1.1000   0.8250  0.275 10.41844       4.521745       2.324659
## 3  2   M 1.3875   1.1125  0.375 24.77746      11.339800       5.556502
##   Shell.Weight Age
## 1     8.348928   9
## 2     3.401940   8
## 3     6.662133   9
```

```
glimpse(df_train)
```

```
## Rows: 74,051
## Columns: 10
## $ id             <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Sex            <chr> "I", "I", "M", "F", "I", "M", "M", "I", "F", "M", "I", ~
## $ Length         <dbl> 1.5250, 1.1000, 1.3875, 1.7000, 1.2500, 1.5000, 1.5750,~
## $ Diameter       <dbl> 1.1750, 0.8250, 1.1125, 1.4125, 1.0125, 1.1750, 1.1375,~
## $ Height         <dbl> 0.3750, 0.2750, 0.3750, 0.5000, 0.3375, 0.4125, 0.3500,~
## $ Weight         <dbl> 28.973189, 10.418441, 24.777463, 50.660556, 23.289114, ~
## $ Shucked.Weight <dbl> 12.7289255, 4.5217453, 11.3398000, 20.3549410, 11.97766~
## $ Viscera.Weight <dbl> 6.6479577, 2.3246590, 5.5565020, 10.9918385, 4.5075705,~
## $ Shell.Weight   <dbl> 8.348928, 3.401940, 6.662133, 14.996885, 5.953395, 7.93~
## $ Age            <int> 9, 8, 9, 11, 8, 10, 11, 11, 12, 11, 7, 10, 10, 7, 9, 7,~
```

It can be seen that the dataset has 10 columns. Since we are interested to find the age of any crab. Therefore, 'Age' is dependent variable and all others can be independent variable. The age of a crab does not depend on its sex so sex won;t be our independent variable. Length, Weight, Diameter, Height, etc. can be our independent variable.

Task 1.

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. 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?

Descriptive Statistics of the training dataset

```
summary(df_train)
```

```
##        id             Sex                Length          Diameter     
##  Min.   :    0   Length:74051       Min.   :0.1875   Min.   :0.1375  
##  1st Qu.:18513   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
```

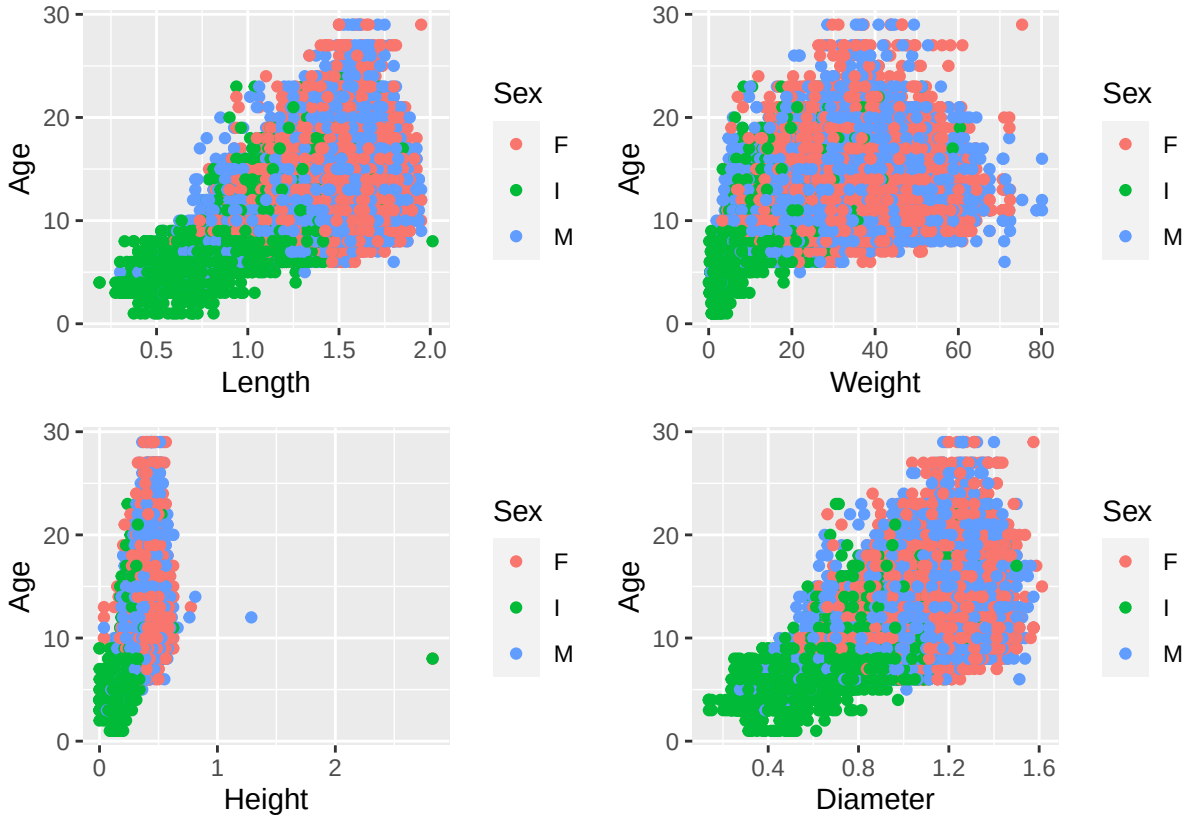
Scatterplot Matrix of independent variables

```
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
```

```
## The following object is masked from 'package:MASS':
##
##     area
```

```
p1 <- ggplot(data=df_train, mapping = aes(x= Length, y= Age))+
  geom_point(aes(color=Sex))
p2 <- ggplot(data=df_train, mapping = aes(x= Weight, y= Age))+
  geom_point(aes(color=Sex))
p3 <- ggplot(data=df_train, mapping = aes(x= Height, y= Age))+
  geom_point(aes(color=Sex))
p4 <- ggplot(data=df_train, mapping = aes(x= Diameter, y= Age))+
  geom_point(aes(color=Sex))
p1+p2+p3+p4+plot_layout(ncol=2)
```



Correlation matrix of three quantitative variables

Let's consider the variables Age, Height, Length as three quantitative variables. The correlation matrix is given below:

```
correlation_matrix <- cor(df_train[ c("Age", "Height", "Length")])
correlation_matrix
```

```
##               Age    Height    Length
## Age    1.0000000 0.6380669 0.6128431
## Height 0.6380669 1.0000000 0.9183517
## Length 0.6128431 0.9183517 1.0000000
```

Let's proceed to pair-wise correlation test

```
cor.test(~ Height+Length, data=df_train, method='pearson', conf.level=0.95)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Height and Length
## t = 631.44, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9172160 0.9194724
```

```
## sample estimates:
##       cor
## 0.9183517
```

The p-value is close to 0 but not zero and correlation between Height and Length is 0.9183517

```
cor.test(~ Height+ Age, data=df_train, method='pearson', con_level = 0.95)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Height and Age
## t = 225.5, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6337771 0.6423175
## sample estimates:
##       cor
## 0.6380669
```

```
cor.test(~Age+Weight, data=df_train, method='pearson', con_level = 0.95)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Age and Weight
## t = 204.73, df = 74049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5965757 0.6057744
## sample estimates:
##      cor
## 0.601195
```

Null hypothesis: There is no correlation between the variables.

Since the p-value value is less than 0.05, therefore, null hypothesis is rejected and it can be concluded based on the statistics test that there exist correlation between the variables. Yes, familywise error might be there in the analysis and to avoid the error, we can add some cautionary steps in the linear regression equation.

Task 2. 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.

```
pr_matrix <- solve(correlation_matrix)
pr_matrix%*%correlation_matrix
```

```
##                  Age        Height        Length
## Age     1.000000e+00 -1.110223e-16 -1.110223e-16
## Height  4.440892e-16  1.000000e+00  8.881784e-16
## Length -4.440892e-16 -8.881784e-16  1.000000e+00
```

```
correlation_matrix %*% pr_matrix
```

```
##                   Age Height Length
## Age     1.000000e+00      0      0
## Height -1.665335e-16      1      0
## Length -1.110223e-16      0      1
```

LDU decomposition of correlation matrix

```
dec_mat <-lu(correlation_matrix)
dec_mat
```

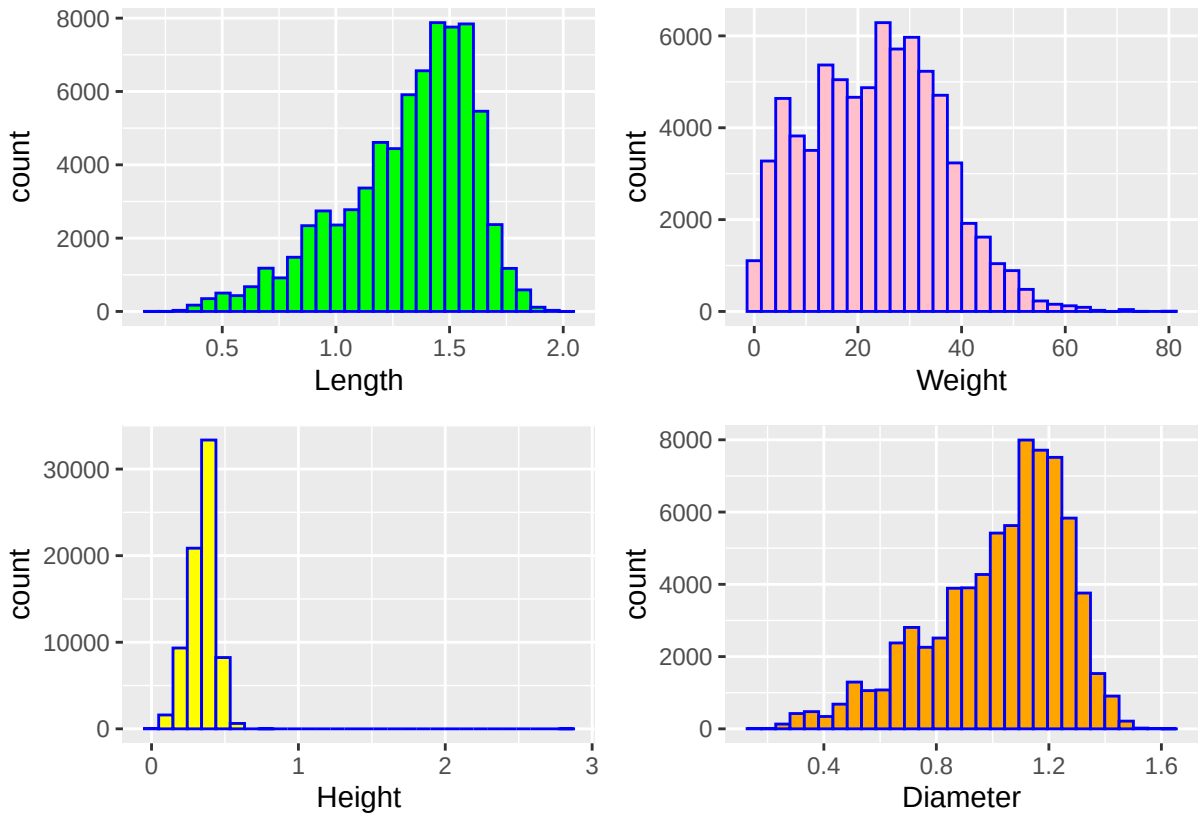
```
## LU factorization of Formal class 'denseLU' [package "Matrix"] with 4 slots
##   ..@ x       : num [1:9] 1 0.638 0.613 0.638 0.593 ...
##   ..@ perm    : int [1:3] 1 2 3
##   ..@ Dim     : int [1:2] 3 3
##   ..@ Dimnames:List of 2
##   .. ..$ : chr [1:3] "Age" "Height" "Length"
##   .. ..$ : chr [1:3] "Age" "Height" "Length"
```

Task 3. *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 l for this distribution, and then take 1000 samples from this exponential distribution using this value (e.g., rexp(1000, l)). Plot a histogram and compare it with a histogram of your original variable. Using the exponential pdf, find the 5^{th} and 95^{th} percentiles using the cumulative distribution function (CDF). Also generate a 95% confidence interval from the empirical data, assuming normality. Finally, provide the empirical 5^{th} percentile and 95^{th} percentile of the data. Discuss.

```
p1 <- ggplot(data=df_train, mapping = aes(x= Length))+
  geom_histogram( color = 'blue', fill= 'green')
p2 <- ggplot(data=df_train, mapping = aes(x= Weight))+
  geom_histogram( color = 'blue', fill= 'pink')
p3 <- ggplot(data=df_train, mapping = aes(x= Height))+
  geom_histogram( color = 'blue', fill= 'yellow')
p4 <- ggplot(data=df_train, mapping = aes(x= Diameter))+
  geom_histogram( color = 'blue', fill= 'orange')
p1+p2+p3+p4+plot_layout(ncol=2)
```

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



Distribution of the variable 'Length' , 'Diameter' are left skewed, Distribution of height seems symmetric but distribution of Weight is right skewed.

```
p1 <- ggplot(data=df_train, mapping = aes(x= Age))+
  geom_histogram( color = 'blue', fill= 'green')
p2 <- ggplot(data=df_train, mapping = aes(x= Shucked.Weight))+
  geom_histogram( color = 'blue', fill= 'pink')
p3 <- ggplot(data=df_train, mapping = aes(x= Viscera.Weight))+
  geom_histogram( color = 'blue', fill= 'yellow')
p4 <- ggplot(data=df_train, mapping = aes(x= Shell.Weight))+
  geom_histogram( color = 'blue', fill= 'orange')
p1+p2+p3+p4+plot_layout(ncol=2)
```

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

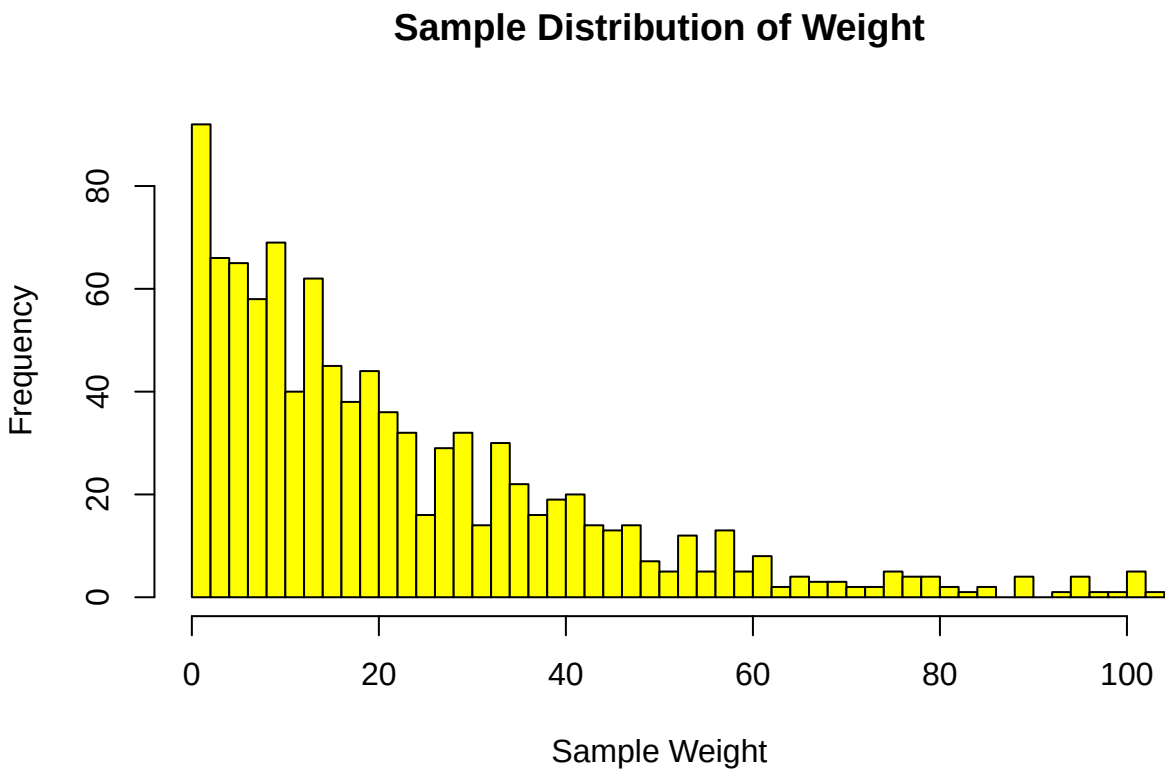
All the above variables "Shucked.Weight" "Viscera.Weight" "Shell.Weight" and "Age" are approximately right skewed.

We can consider 'Weight' for the fit exponential distribution.

```
w_exp <- fitdistr(df_train$Weight, densfun = 'exponential')
lambda <- w_exp$estimate['rate']
cat("lambda: \n\n", lambda)
```

```
## lambda:
##
##  0.04276206
```

```
w_sample <- rexp(1000, lambda)
hist(w_sample, main="Sample Distribution of Weight", col = 'yellow', xlab = "Sample Weight", breaks = 7
```



Both the histograms of original and sample data are right skewed but original data produced a thick left tail while sampled data has thin left tail.

5th and 95th percentile using CDF

```
per_5th <- qexp(0.05, lambda)
per_95th <- qexp(0.95, lambda)
cat("5th percentile: ", per_5th)
```

```
## 5th percentile:  1.199505
```

```
cat("\n95th percentile: ", per_95th)
```

```
##
## 95th percentile:  70.05585
```

Confidence intervals using the empirical data

The 95% confidence interval can be found using the sampled data or the original data. I'll prefer the sampled data since 5th and 95th percentiles are calculated using the sampled data.

```
library(infer)
w_sample <- data.frame(w_sample)
set.seed(1000)
ci_diff <- w_sample %>%
```

```r
get_ci(level = 0.95)
lower<- ci_diff$lower_ci
upper<- ci_diff$upper_ci
sprintf("95 percent confidence interval is [%.4f, %.4f]", lower, upper)
```

```
## [1] "95 percent confidence interval is [0.6764, 85.6784]"
```

```r
##
```

The two confidence interval differ a little. The confidence interval based on the normal distribution is a little wide while CDF gives narrow confidence interval. Also, it can be seen that CI based on CDF of exponential distribution lies wholly within the CI based on normal distribution.

Task 4. Modelling

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
reg_model <- lm(Age~ Height+Weight+Diameter+Length+Height*Length, data = df_train)
summary(reg_model)
```

```
##
## Call:
## lm(formula = Age ~ Height + Weight + Diameter + Length + Height *
##     Length, data = df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.263  -1.467  -0.552   0.751  18.534
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.265456   0.111461   2.382   0.0172 *
## Height          28.602473   0.634503  45.079   <2e-16 ***
## Weight           0.070905   0.003429  20.677   <2e-16 ***
## Diameter         4.019979   0.278453  14.437   <2e-16 ***
## Length          -0.597943   0.233756  -2.558   0.0105 *
## Height:Length -10.856355   0.465049 -23.345   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.418 on 74045 degrees of freedom
## Multiple R-squared:   0.42,  Adjusted R-squared:  0.4199
## F-statistic: 1.072e+04 on 5 and 74045 DF,  p-value: < 2.2e-16
```

Test the model

```r
df_test <- read.csv("https://raw.githubusercontent.com/jewelercart/605/main/test.csv")
pred_Age <- predict(reg_model, df_test)
pred_Age[1:10]
```

```
##         1         2         3         4         5         6         7         8
##  8.044838  8.632660  9.251639  9.883375  8.189186 10.242748 12.382566  8.872461
##         9        10
##  9.550946  8.999525
```