Biometry: Problem Set 1

Robert Dellinger

9/6/2022

Problem Set #1: Basic Statistics and Introduction to R

(1) (4 pts) The following data are numbers of protozoa sampled from a microcosm and counted on a hemacytometer: 6, 11, 4, 5, 7, 3, 5, 1, 5, 6. Calculate the statistics listed below for this sample of protozoan densities. Do these calculations "by hand" using a calculator.

(2a) (4 pts) Being the insightful biologist that you are, you notice that protozoa seem more dense at the bottom of the microcosm, perhaps because there is more food available there. You want to know if there is statistical support for this casual observation. You sample 20 replicate microcosms and measure the densities of protozoa. In 10 of the microcosms, you take the sample from the top and in the other 10 microcosms, you take the sample from the bottom. The data are as follows:

Top (# per uL): 3, 1, 0, 5, 4, 3, 6, 3, 4, 7 Bottom (# per uL): 3, 12, 3, 4, 7, 8, 7, 5, 15, 9

Using R, calculate the following statistics for both top and bottom: Mean, Standard deviation, Variance, 95% Confidence Interval.

```
#calculating statistics for top data using mean, sd, var functions
Top <- c(3, 1, 0, 5, 4, 3, 6, 3, 4, 7)
mean_top <- mean(Top)
standard_deviation_top <- sd(Top)
variance_top <- var(Top)

#calculating a 95% confidence interval
n_top <-length(Top) #length of n
standard_error_top <- standard_deviation_top/sqrt(n_top) #calculating standard error

alpha <- 0.05 #95% confidence for alpha
degrees_freedom_top <- n_top - 1 #finding degrees of freedom
t_score_top <- qt(p=alpha/2, df=degrees_freedom_top) #qt() command calculates the t-score
margin_error_top <- t_score_top * standard_error_top #finding margin error
#confidence interval is the mean +/- margin of error
  lower_bound_top <- mean_top - margin_error_top
  upper_bound_top <- mean_top + margin_error_top
confidence_interval_top<-(c(lower_bound_top,upper_bound_top))

#calculating statistics for bottom data using mean, sd, var functions
Bottom <- c(3, 12, 3, 4, 7, 8, 7, 5, 15, 9)
mean_bottom <- mean(Bottom)
standard_deviation_bottom <- sd(Bottom)
variance_bottom <- var(Bottom)
```

$$\text{mean} = \bar{X} = \frac{\Sigma X}{n} = \frac{6+11+4+5+7+3+5+1+5+6}{10}$$

$$\boxed{\bar{X} = \frac{53}{10} = 5.3}$$

$$\text{Variance} = S^2 = \frac{\Sigma (x-\bar{x})^2}{n-1}$$

$$\Sigma (x-\bar{x})^2 = (6-5.3)^2 + (11-5.3)^2 + (4-5.3)^2 + (5-5.3)^2 + (7-5.3)^2 + (3-5.3)^2 + (1-5.3)^2 + (5-5.3)^2 + (6-5.3)^2$$

$$= 62.1$$

$$\boxed{S^2 = \frac{62.1}{10-1} = 6.9}$$

$$\text{Standard deviation} = \sqrt{S^2} = \sqrt{6.9}$$

$$\boxed{S_d = 2.627}$$

$$\text{Standard error} = S/\sqrt{n} = \frac{2.627}{\sqrt{10}}$$

$$\boxed{S_\varepsilon = 0.831}$$

$$\text{Coefficient of variation} = S/\bar{X} = \frac{2.627}{5.3} * 100$$

$$\boxed{CV = 49.6\%}$$

Median & Mode 1, 3, 4, (5), (5), (5), 6, 6, 7, 11

Median = (n+1)/2 = 5.5th

$$\boxed{\text{Median} = 5} \qquad \boxed{\text{Mode} = 5}$$

Figure 1: Question 1 Answer

```r
#calculating a 95% confidence interval
n_bottom <-length(Bottom) #length of n
standard_error_bottom <- standard_deviation_bottom/sqrt(n_bottom) #calculating standard error

alpha <- 0.05 #95% confidence for alpha
degrees_freedom_bottom <- n_bottom - 1 #finding degrees of freedom
t_score_bottom <- qt(p=alpha/2, df=degrees_freedom_bottom) #qt() command calculates the t-score
margin_error_bottom <- t_score_bottom * standard_error_bottom #finding margin error
#confidence interval is the mean +/- margin of error
  lower_bound_bottom <- mean_bottom - margin_error_bottom
  upper_bound_bottom <- mean_bottom + margin_error_bottom
confidence_interval_bottom<-(c(lower_bound_bottom,upper_bound_bottom))

#answers
mean_top
```

```
## [1] 3.6
```

```r
mean_bottom
```

```
## [1] 7.3
```

```r
standard_deviation_top
```

```
## [1] 2.1187
```

```r
standard_deviation_bottom
```

```
## [1] 3.917199
```

```r
variance_top
```

```
## [1] 4.488889
```

```r
variance_bottom
```

```
## [1] 15.34444
```

```r
confidence_interval_top
```

```
## [1] 5.115627 2.084373
```

```r
confidence_interval_bottom
```

```
## [1] 10.102195  4.497805
```

```
standard_error_top
```

```
## [1] 0.6699917
```

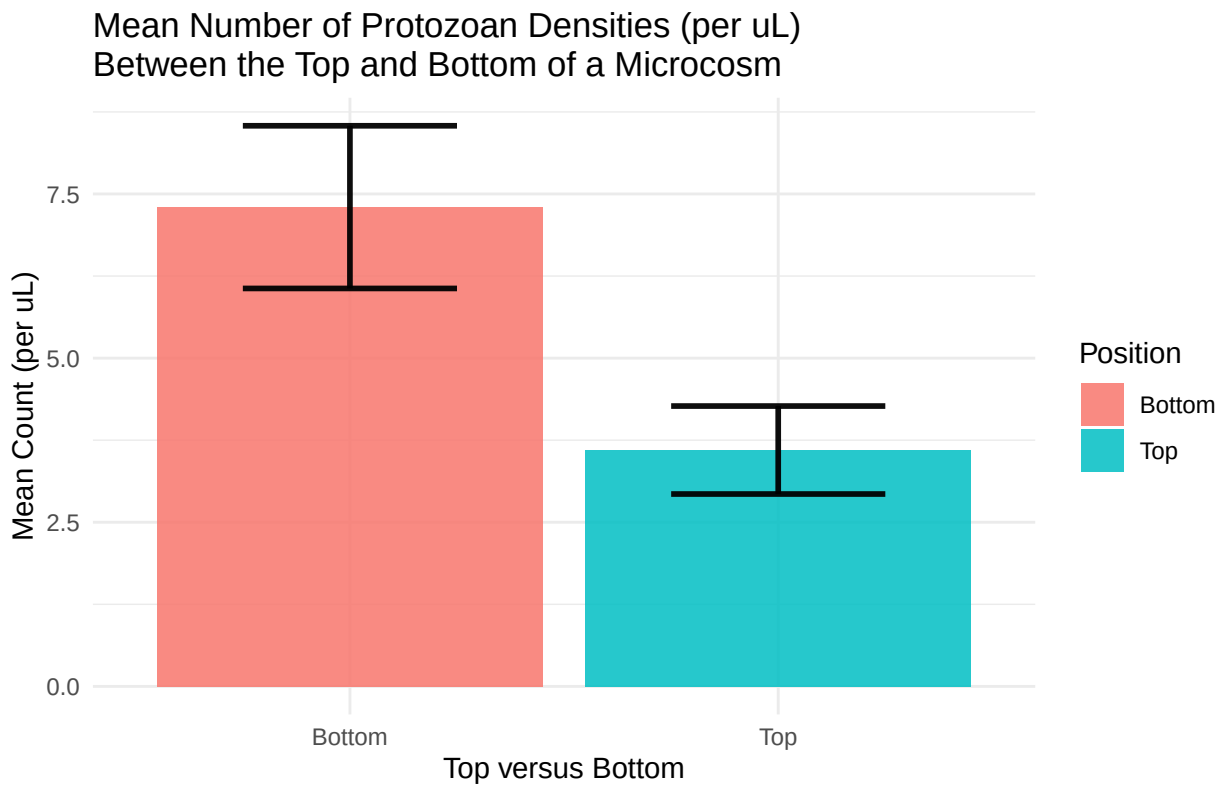
```
standard_error_bottom
```

```
## [1] 1.238727
```

(2b) (5 pts) Make a publication-quality bar graph in R that presents means and standard errors for each group (top vs bottom). Provide a figure legend that describes the graph and includes a statement about whether you think protozoa densities differ between the top and bottom of the microcosm.

```
#create data set of summary statistics
data <- data.frame(
  name=c("Top","Bottom") ,
  mean=c(3.6,7.3),
  se=c(0.6699917,1.238727)) #standard error

#plot summary statistics
ggplot(data) +
  geom_bar( aes(x=name, y=mean, fill=name), stat="identity", alpha=0.85) +
  geom_errorbar( aes(x=name, ymin=mean-se, ymax=mean+se), width=0.5, colour="black",
    alpha=0.95, size=1)+
  theme(legend.position = "right", legend.title=element_text(size=20),
        legend.text=element_text(size=14))+
  theme_minimal()+
labs(y = "Mean Count (per uL)", x="Top versus Bottom", title = "Mean Number of Protozoan Densities (per
```



Mean number of protozoan between the top and bottom of a microcosm differ.
Given that the error between bars plots do not overlap, the differences are statistically relevant.

(3a) (3 pts) The Excel file named "kelp bass gonad mass" contains the weights of gonads from several hundred kelp bass collected by Dr. Mark Steele's lab. Estimate the mean, median, s2, s, CV, skewness, and kurtosis.

```
Kelp_Bass_Gonad_Data <- read_csv(here("Data", "Kelp_Bass_Gonad_Data.csv"))
```

```
## Rows: 599 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): gonad_mass
##
## 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.
```

```
#glimpse(Kelp_Bass_Gonad_Data)

mean.gonad <- mean(Kelp_Bass_Gonad_Data$gonad_mass)
median.gonad <- median(Kelp_Bass_Gonad_Data$gonad_mass)
variance.gonad <- var(Kelp_Bass_Gonad_Data$gonad_mass)
sd.gonad <- sd(Kelp_Bass_Gonad_Data$gonad_mass)
cv.gonad <- sd.gonad/mean.gonad*100
skew.gonad <- skewness(Kelp_Bass_Gonad_Data$gonad_mass)
kurtosis.gonad <- kurtosis(Kelp_Bass_Gonad_Data$gonad_mass)

mean.gonad
```

```
## [1] 8.236945
```

```
median.gonad
```

```
## [1] 6.42
```

```
variance.gonad
```

```
## [1] 57.93812
```

```
sd.gonad
```

```
## [1] 7.611709
```

```
cv.gonad
```

```
## [1] 92.40937
```

```
skew.gonad
```

```
## [1] 1.428559
```

```
kurtosis.gonad
```

```
## [1] 5.159599
```

3b) (3 pts) What effect would adding 5.0 to each observation of gonad mass have on the values of the mean, median, s2, s, CV, skewness, and kurtosis? (You don't need to show the new values, but just describe how the statistics have changed.)

Adding 5.0 to each observation of gonad mass would increase the mean, median, and mode by 5 but the range of the IQR will remain the same. The standard deviation, variance, CV, skewness and kurtosis would not change as the difference between variables have not changed.

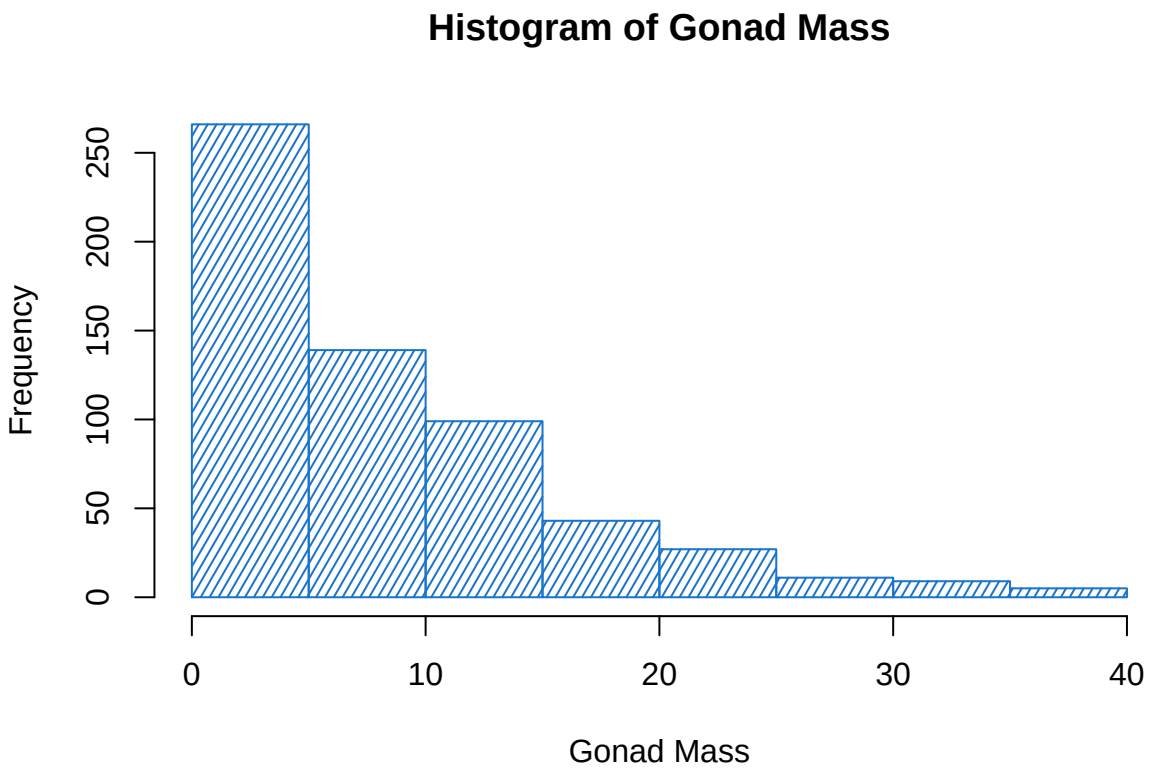
(3c) (3 pts) What would be the effect of adding 5.0 and then multiplying by 10.0?

Multiplying the dataset by 10 would would multiply the values of the median, mode, range, and IQR by 10. Multiplying the dataset by 10 would increase the standard deviation, variance, CV, skewness and kurtosis.

And this will always be true. No matter what value we multiply by the data set, the mean, median, mode, range, and IQR will all be multiplied by the same value.

(3d) (3 pts) Make a histogram of all raw observations (untransformed values) in the kelp bass gonad mass data set. Do these data look relatively normal or not? Add the histogram below.

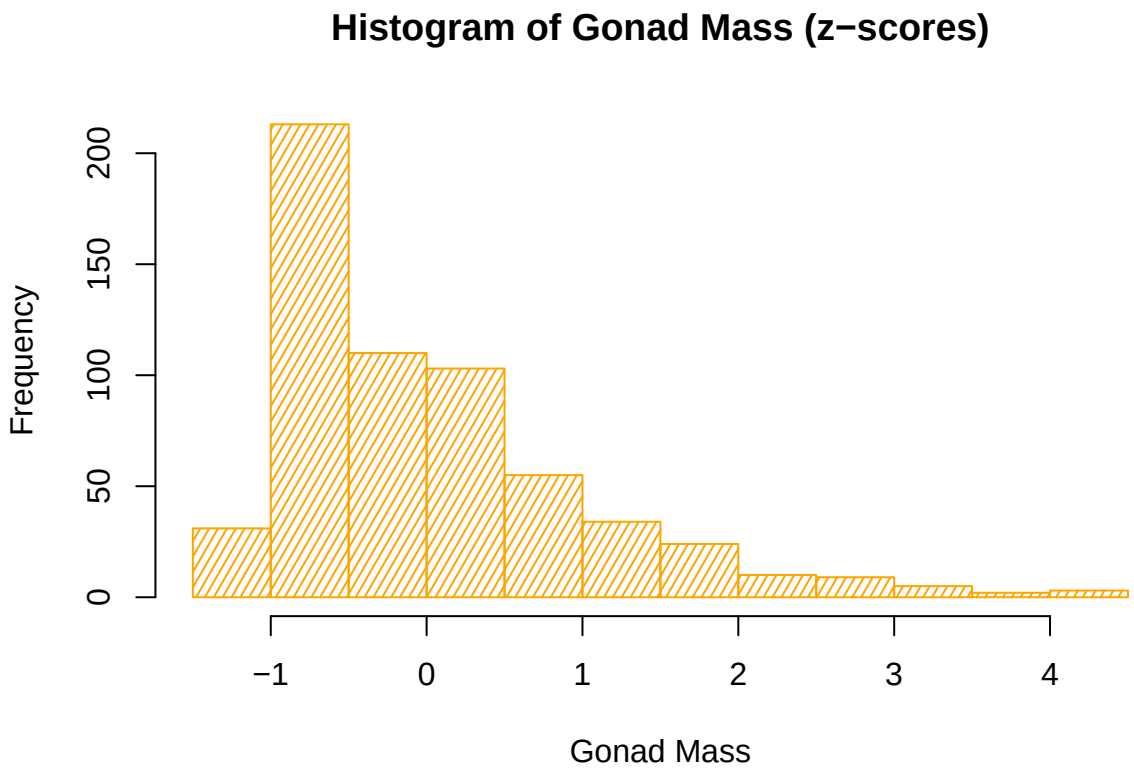
```
#creating a histogram plot using base R
n_gonad <-length(Kelp_Bass_Gonad_Data$gonad_mass) #length of data
hist(Kelp_Bass_Gonad_Data$gonad_mass, breaks=10, # over 500 data points, use 10 breaks
     col="dodgerblue3", density=25, angle=60, #creating hatch pattern
     main="Histogram of Gonad Mass", xlab="Gonad Mass") #labels
```



Data is right skewed and does not folllow a normal distribution.

(3e) (3 pts) Convert all raw observations in the kelp bass data set into Z-scores. Make a histogram of this new data set. How does this histogram differ from the one for the raw observations? Add the new histogram below.

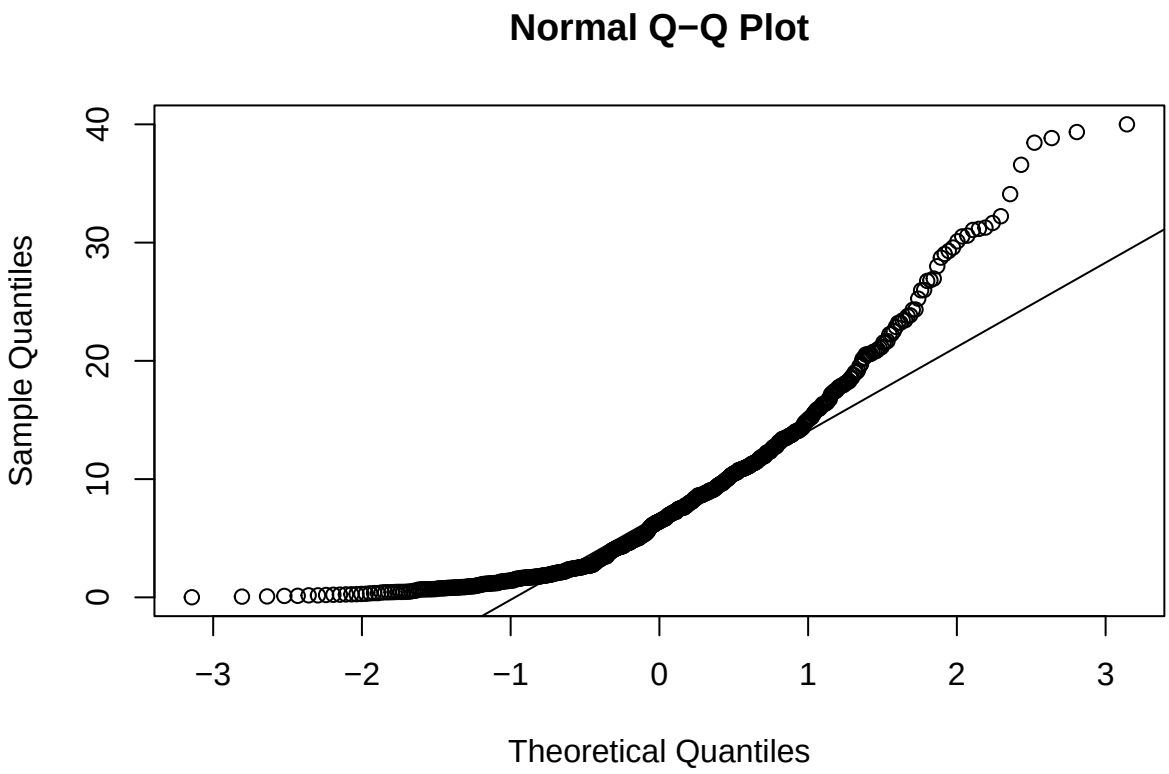
```
#converting gonad data to z-scores and plotting using a histogram through base R
zscore.gonad<-scale(Kelp_Bass_Gonad_Data$gonad_mass, center=TRUE, scale=TRUE)
hist(zscore.gonad, breaks=10, # over 500 data points, use 10 breaks
     col="orange", density=25, angle=60, #creating hatch pattern
     main="Histogram of Gonad Mass (z-scores)", xlab="Gonad Mass") #labels
```



Data follow a far more normal distribution of data rather then that of the raw data observations which is skewed.

(3f) (3 pts) Use the original kelp bass gonad data to create a Normal Probability Plot. Do the data appear to be normally distributed? Add the plot below.

```
#creating a normal probability plot
qqnorm(Kelp_Bass_Gonad_Data$gonad_mass)
qqline(Kelp_Bass_Gonad_Data$gonad_mass)
```



The data appear to be exponential and do not fit the linear regression as a normal line.

(4) (5 pts) Round the following numbers to three significant figures and state their implied limits before and after rounding.

number	implied limits	rounded to 3 SF	implied limits
106.5	106.45 − 106.55	107	106.5 − 107.5
0.068191	0.0681905 − 0.0681915	0.0682	0.06815 − 0.06825
3.049	3.0485 − 3.0495	3.05	3.045 − 3.055
2.03456 x 10^6	$2.03455\times10^6 - 2.03465\times10^6$	203×10^6	$202.5\times10^6 - 203.5\times10^6$
2.914	2.9135 − 2.9145	2.19	2.185 − 2.195
20.15000	20.14995 − 20.15005	20.2	20.15 − 20.25

Figure 2: Question 4 Answer

(5) (5 pts) For each of the following questions, define the sampling unit and the statistical population.

(6a) (5 pts) Carla (former MS student in Peter Edmunds' lab) sampled the weights (in grams) of 30 individuals of the coral, Agaricia agaricites. The data are available in the file "Agaricia.csv". Are the data normally distributed? Does log-transformation improve the normality or not? Support your answer with whatever graph(s) you think are appropriate.

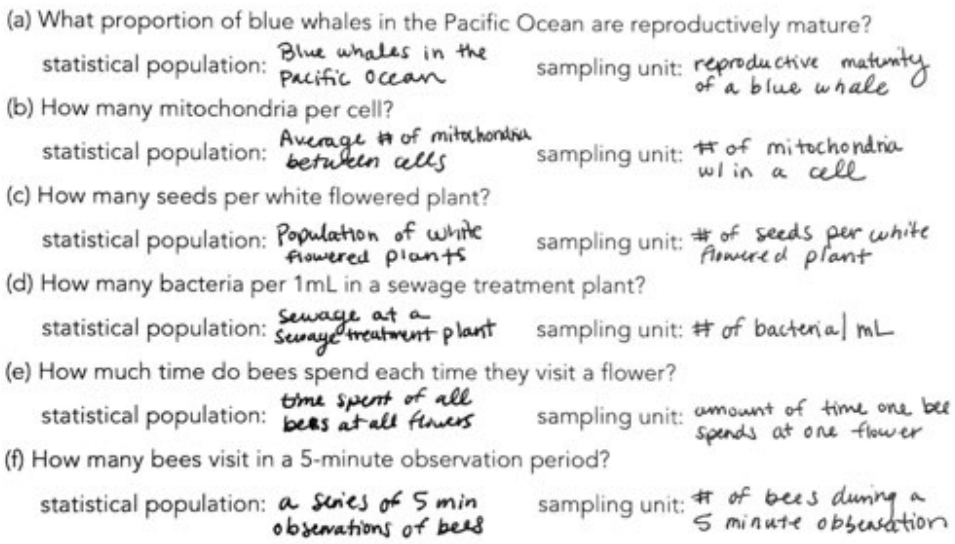


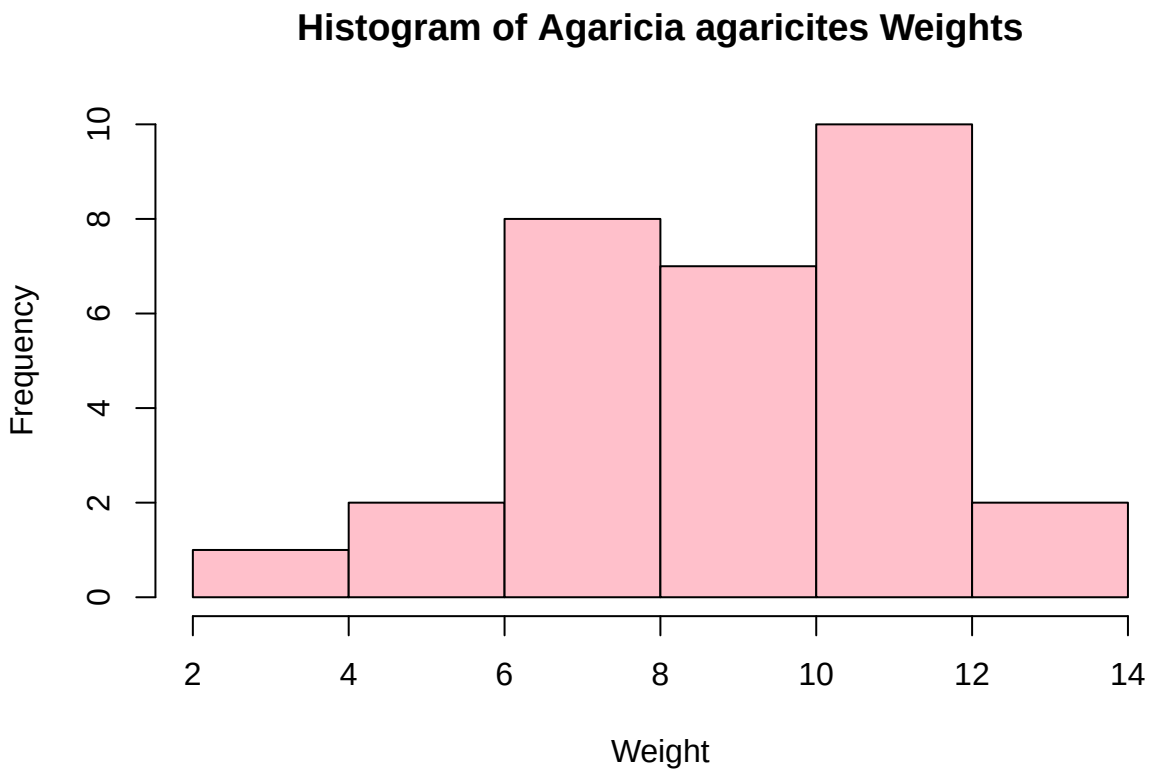
Figure 3: Question 5 Answer

```
Agaricia_Data <- read_csv(here("Data", "Agaricia.csv"))
```

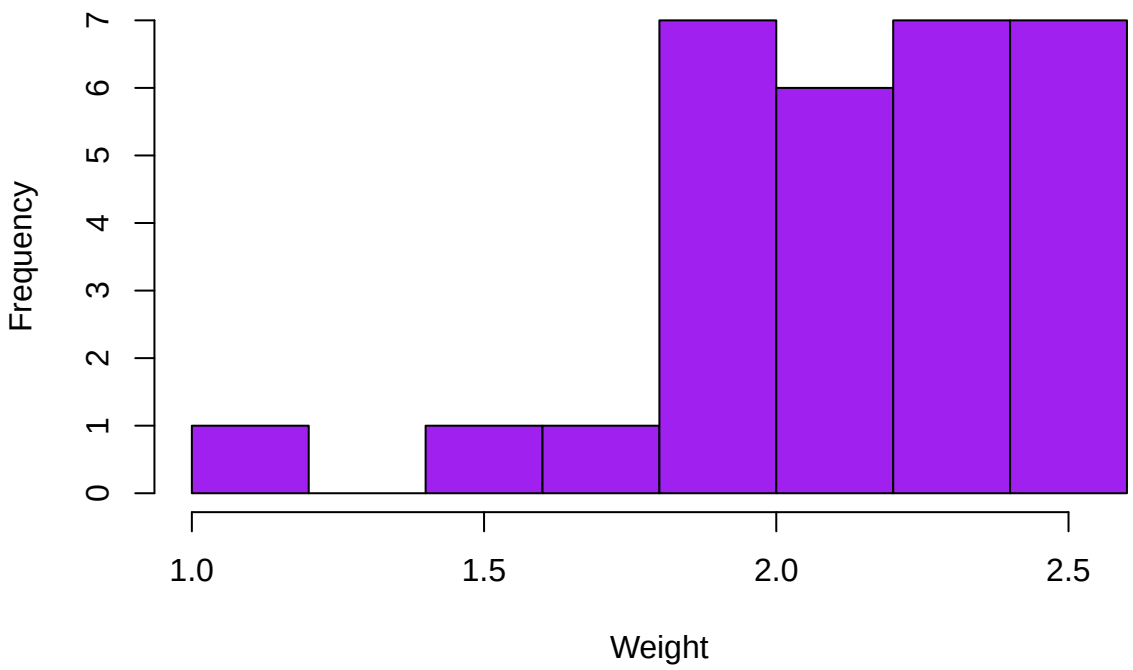
```
## Rows: 30 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): weight
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#glimpse(Agaricia_Data)

#creating a histogram plot using base R
n_weight <-length(Agaricia_Data$weight) #length of data
hist(Agaricia_Data$weight, breaks=6, # less then 50 data points, use 6 breaks
     col="pink",
     main="Histogram of Agaricia agaricites Weights", xlab="Weight") #labels
```




```
log_Agaricia_Data <- log(Agaricia_Data)
hist(log_Agaricia_Data$weight, breaks=6, # less then 50 data points, use 6 breaks
     col="purple",
     main="Log Transformed Histogram of Agaricia agaricites Weights", xlab="Weight") #labels
```



As shown by the data plots above the data distribution in the histogram are normal. Log transforming the data however, changes the distrubtion to a non normal distrubtion as shown in the second plot above.

(6b) (4 pts) Use the Agaricia data set to estimate the mean ± 95% CI of the untransformed data sample by resampling the data with bootstrapping (just use 1000 resamplings). Plot the frequency distribution of estimates for the mean and indicate the 95% confidence intervals on the plot.

```
#sample mean
mean.weight <- mean(Agaricia_Data$weight)

#bootstrapping means
mean.weight.bootmeans<-replicate(1000, {
  samples<-sample(Agaricia_Data$weight,replace=TRUE);
  mean(samples)  }) #take the mean of the subsample


sortedboots<-sort(mean.weight.bootmeans) #sorting means

#constructing the 95% confidence intervals using (25th and 975th place)
lowCI<-sortedboots[25]
highCI<-sortedboots[975]
upperCI<-highCI - mean(mean.weight.bootmeans)
lowerCI<-mean(mean.weight.bootmeans) - lowCI

upperCI
```

```
## [1] 0.8277606
```

```
lowerCI
```

```
## [1] 0.8773094
```

```
#histogram of bootstrapped means
hist(sortedboots, breaks=6, # less then 50 data points, use 6 breaks
     col="lightcyan",
     main="Bootstrapped Histogram of Agaricia agaricites Weights", xlab="Weight")
abline(v=lowCI, col="darkcyan") #adding vertical lines for the low and high CIs
abline(v=highCI, col="darkcyan")
```