IST772 Chapter Notes Template: After Completing Please Submit as a PDF.
Originality Assertion: By submitting this file you affirm that this writing is your own.

Name: Wanyue Xiao
Date: September 3, 2020
Chapter Number: # 3
Title of Chapter: **Probabilities in the Long Run**

Sampling and Sampling Techniques

1. **Sampling:** process of drawing a subset of elements from the population.
2. **Sampling error**: occurs to varying and unknown degrees when a statistic obtained from a sample does not precisely match the parameter from the population
3. **Nonresponse bias**: occur if the sampled people who failed to participate were different in some important way from those who did participate. -> this will cause bias eventually.
4. **Quartile:** dividing up a set of values into four equal parts: one quarter on the low end, one quarter on the high end, and two quarters in the middle

R functions:

1. **sample() function**: randomly draw a sample set with or without replacement.
2. **replicate() function**: repeat certain function by setting the replication times.
3. **Sampling mean is different with population mean**.

Two theorems:

1. **Law of large numbers:** as a sample size grows, its mean gets closer to the average of the whole population.
2. **Central limit theorem:** if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement, then the distribution of the sample means will be approximately normally distributed

There is no inset box in this chapter

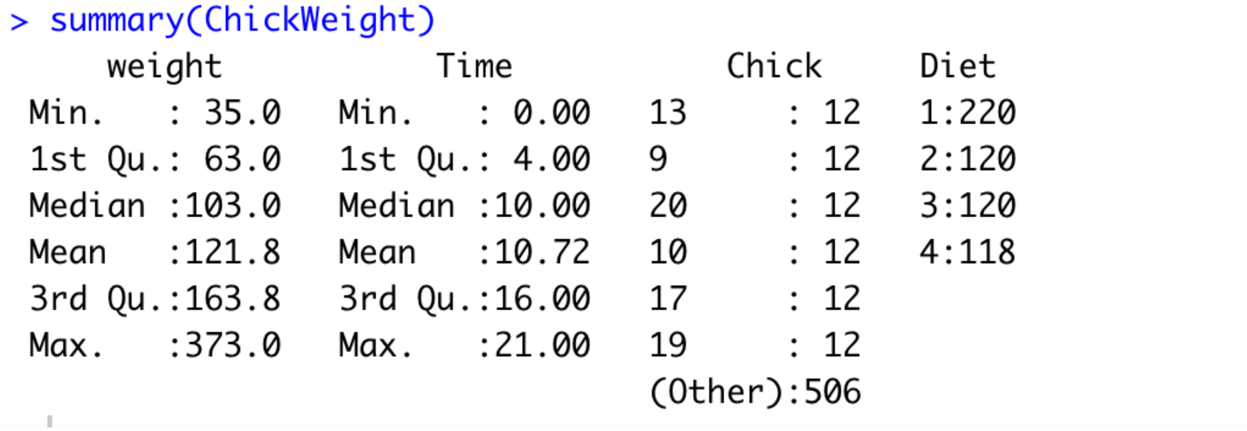
Since this chapter does not contain any inset box, I only summarize the important codes that appear in this chapter.

```
# check data details
head(toastAngleData)
tail(toastAngleData)
hist(toastAngleData)

# sampling and replication
sample(toastAngleData,size=14,replace=TRUE)
replicate(10000, mean(sample(toastAngleData, size=14, replace=TRUE)), simplify = TRUE)

# calculate quantiles
quantile(samplingDistribution, c(0.01, 0.05, 0.50, 0.95, 0.99))
abline(v=quantile(samplingDistribution,0.01))
summary(samplingDistribution[samplingDistribution <= quantile (samplingDistribution, .01)])
```

Exercise Review

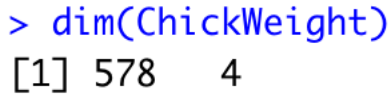
IST772 Chapter Notes Template: After Completing Please Submit as a PDF.
Originality Assertion: By submitting this file you affirm that this writing is your own.

2.



```
> summary(ChickWeight)
     weight            Time            Chick        Diet   
 Min.   : 35.0   Min.   : 0.00   13     : 12   1:220  
 1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120  
 Median :103.0   Median :10.00   20     : 12   3:120  
 Mean   :121.8   Mean   :10.72   10     : 12   4:118  
 3rd Qu.:163.8   3rd Qu.:16.00   17     : 12          
 Max.   :373.0   Max.   :21.00   19     : 12          
                                 (Other):506          
```

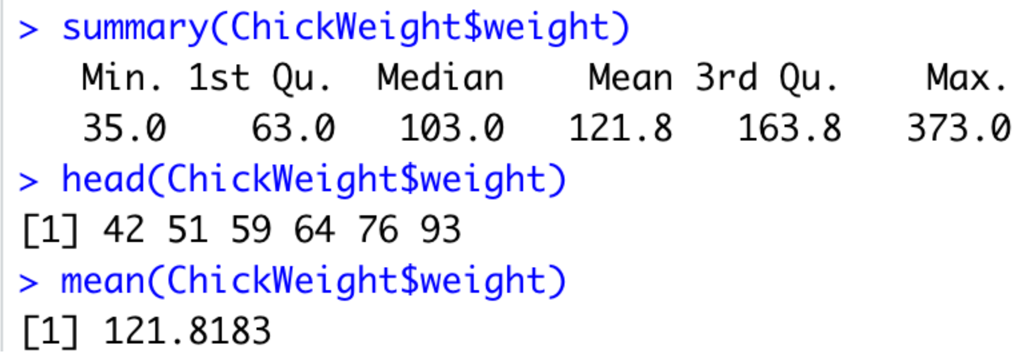
The names of those four variables are “Weight”, “Time”, “Chick”, “Diet”. The weight ranges from 35 to 373, whose gap is significant.



```
> dim(ChickWeight)
[1] 578   4
```

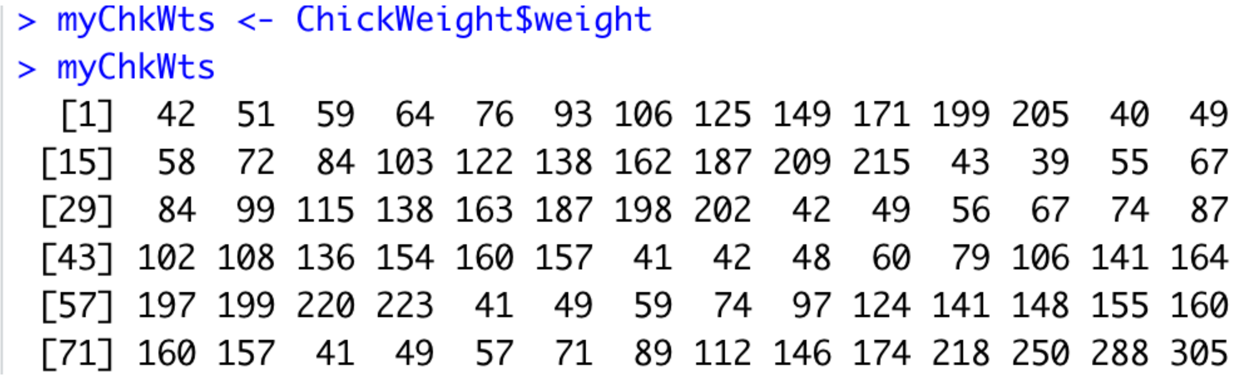
The dimension is “578 rows and 4 columns”.

3.



```
> summary(ChickWeight$weight)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   35.0    63.0   103.0   121.8   163.8   373.0 
> head(ChickWeight$weight)
[1] 42 51 59 64 76 93
> mean(ChickWeight$weight)
[1] 121.8183
```

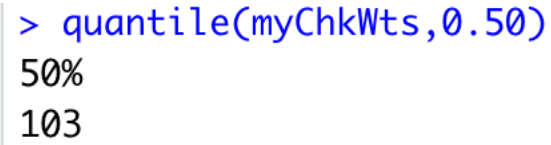
summary() function shows the statistical summaries data of weight variables. head() function display the first 10 data listed in the weight variable. mean() calculate the average value of the total numbers listed in that variable.



```
> myChkWts <- ChickWeight$weight
> myChkWts
  [1]  42  51  59  64  76  93 106 125 149 171 199 205  40  49
 [15]  58  72  84 103 122 138 162 187 209 215  43  39  55  67
 [29]  84  99 115 138 163 187 198 202  42  49  56  67  74  87
 [43] 102 108 136 154 160 157  41  42  48  60  79 106 141 164
 [57] 197 199 220 223  41  49  59  74  97 124 141 148 155 160
 [71] 160 157  41  49  57  71  89 112 146 174 218 250 288 305
```

This command just extract the numbers stored in the weight variable and then store those numbers into a new vector called “myChkWts”.

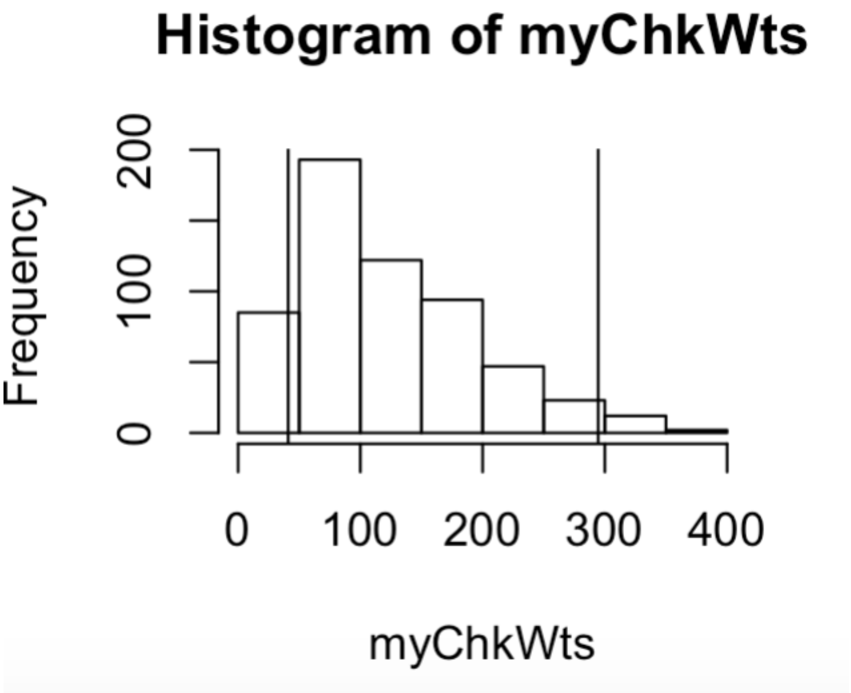
IST772 Chapter Notes Template: After Completing Please Submit as a PDF.
Originality Assertion: By submitting this file you affirm that this writing is your own.



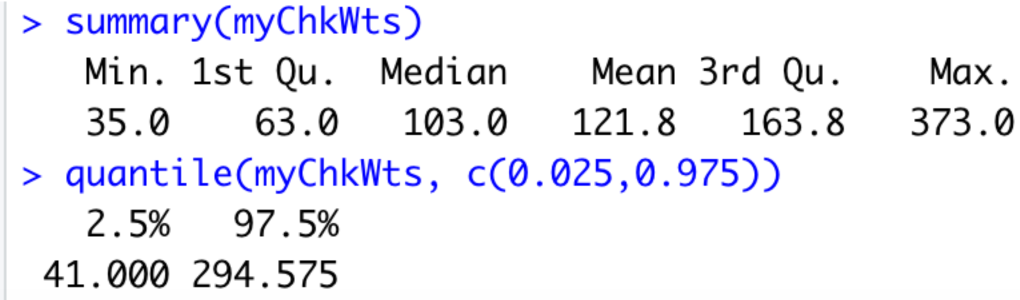
```
> quantile(myChkWts,0.50)
50%
103
```

This command shows the second quantile value of myChkWts by using quantile() function.

4.



This is the histogram of myChkWts variable. It is easy to see that this histogram is slightly right skewed. The mean is at the right-hand side of the median.



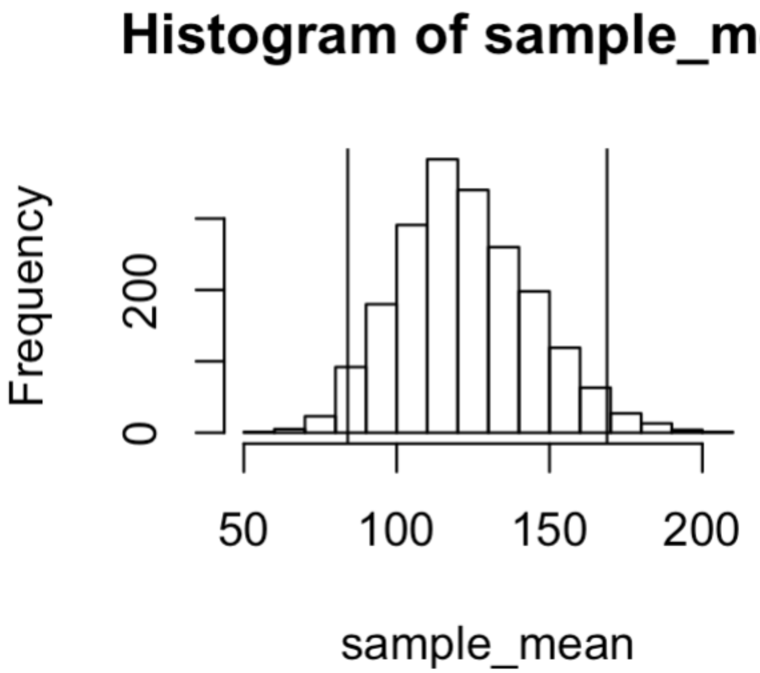
```
> summary(myChkWts)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   35.0    63.0   103.0   121.8   163.8   373.0 
> quantile(myChkWts, c(0.025,0.975))
   2.5%   97.5% 
 41.000 294.575 
```

The mean and median are 121.8 and 103 respectively, indicating that the distribution is right skewed. The values of 0.025 quantile and 0.975 quantile are 42,.000 and 294.575. Normally, people use 0.025 and 0.975 as the boundary to drop outliers since numbers below 0.025 quantile or numbers above 0.975 quantile are definite outliers.

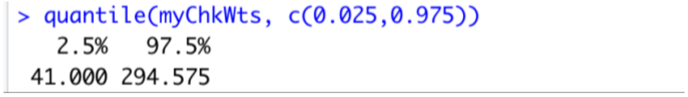
5.

sample_mean <- replicate(2000, mean(sample(myChkWts,size=11,replace=TRUE)), simplify = TRUE)
hist(sample_mean)

abline(v=quantile(sample_mean,0.025))
abline(v=quantile(sample_mean,0.975))

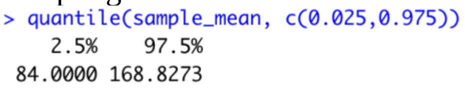


6.
Raw Data:



```
> quantile(myChkWts, c(0.025,0.975))
   2.5%   97.5% 
 41.000 294.575 
```

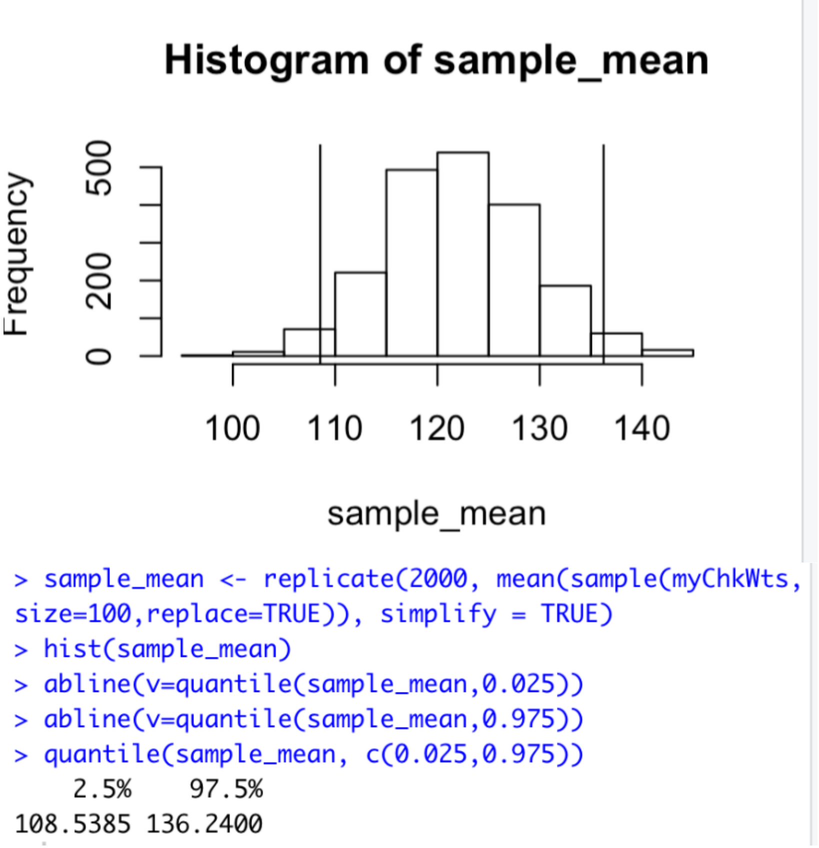
Sampling Mean:



```
> quantile(sample_mean, c(0.025,0.975))
    2.5%    97.5% 
 84.0000 168.8273 
```

The distribution of population is right skewed while that of sampling distribution is normal distributed. For the 0.025 and 0.975 quantiles, the 0.025 quantile of population is lower than that of sampling distribution. Similarly, the 0.975 quantile of population is higher than that of sampling distribution. The reason why those two numbers are so different is that the range of the numbers considered are different. For the population, it has some extreme small or large numbers such as 41 or 294.575. Concluding those numbers during calculation will definitely influence the results. For sampling distribution, however, the effect had been mitigated significantly since we only collect the mean of each sample. Besides, we only take 11 numbers in each sample. Therefore, the mean of each sample could not be very persuasive if we use this sample mean to represent the mean of population.

7.



```
> sample_mean <- replicate(2000, mean(sample(myChkWts,
size=100,replace=TRUE)), simplify = TRUE)
> hist(sample_mean)
> abline(v=quantile(sample_mean,0.025))
> abline(v=quantile(sample_mean,0.975))
> quantile(sample_mean, c(0.025,0.975))
    2.5%    97.5% 
108.5385 136.2400
```

The numbers are narrow down compared with those of the original sampling distribution in #5. The reason behind is that the mean of each sample is getting closer to the mean of the population when we collect more numbers during sampling. Hence, compared with #5, the result of #7 is more convincing since it contains more data during sampling. In a conclusion, considering more data in each sample will generally enhance the performance of sampling.

R Code Fragment and Explanation

The **replicate() function** takes three arguments:

1. n, which is the number of replications to perform. This is where I set the number of simulations I want to run.
2. expr, the expression that should be run repeatedly. I've only ever used a function here.
3. simplify, which controls the type of output the results of expr are saved into. Use simplify = FALSE to get vectors saved into a list instead of in an array.

Question for Class

1. If we roll 10 dices 2,000 times, then we will get 20,000 results as a population. Then we draw 100 numbers from this population and calculate sum of those 100 numbers. After repeating this action 1,000 times, we are supposed to get 1,000 sums. In this case, can we still calculate the grand mean of the sampling distribution (because we calculate the sum of each sample instead of mean instructed in this chapter)?