Assignment 1

Liam

9/9/2019

Part 1

1. Calculate the following sums.

$$1.S_1 = 1 + 2 + ... + 2019$$

```
sum(c(1:2019))
```

```
## [1] 2039190
```

$$2.S_2 = 1^3 + 2^3 + ... + 2019^3$$

```
a <- c(1:2019)
sum(a^3)
```

```
## [1] 4.158296e+12
```

$$3.S_3 = 1^1 + 2^2 + 3^3 + ... + 2019^{2019}$$

```
a <- c(1:2019)
sum(a^a)
```

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

$$4.S_4 = 1^1 - 2^2 + 3^3 - 4^4 + ... - 2018^{2018} + 2019^{2019}$$

```
a <- c(1:2019)
b <- a^a
n <- c(1, -1)
sum(b*n)
```

```
## Warning in b * n: longer object length is not a multiple of shorter object
## length

## [1] NaN
```

$$5.S_5 = 1 + 1/4 + 1/9 + 1/16 + 1/25 + ...$$

```
a <- c(1:999999)
b <- a^2
sum(a/b)
```

```
## [1] 14.39273
```

$$6.S_6 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ...$$

```
a <- c(1:999999)
sum(1/a)
```

```
## [1] 14.39273
```

$$7.S_7 = 1 + \frac{1}{8} + \frac{1}{27} + \frac{1}{64} + ...$$

```
a <- c(1:999999)
sum(a/a^3)
```

```
## [1] 1.644933
```

$$8.S_8 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + ...$$

```
a <- c(1:999999)
b <- c(1,-1)
sum(1/(a*b))
```

```
## Warning in a * b: longer object length is not a multiple of shorter object
## length
```

```
## [1] 0.6931477
```

2. The rnorm function generate random variables from normal distribution. Generate a sample of 1000 values from normal distribution with the mean 10 and standard deviation 1.

a. Calculate the mean and standard deviation of the sample.

```
a <- rnorm(1000, mean=10, sd=1)
mean(a)
```

```
## [1] 10.06761
```

```r
sd(a)
```

```
## [1] 0.9832251
```

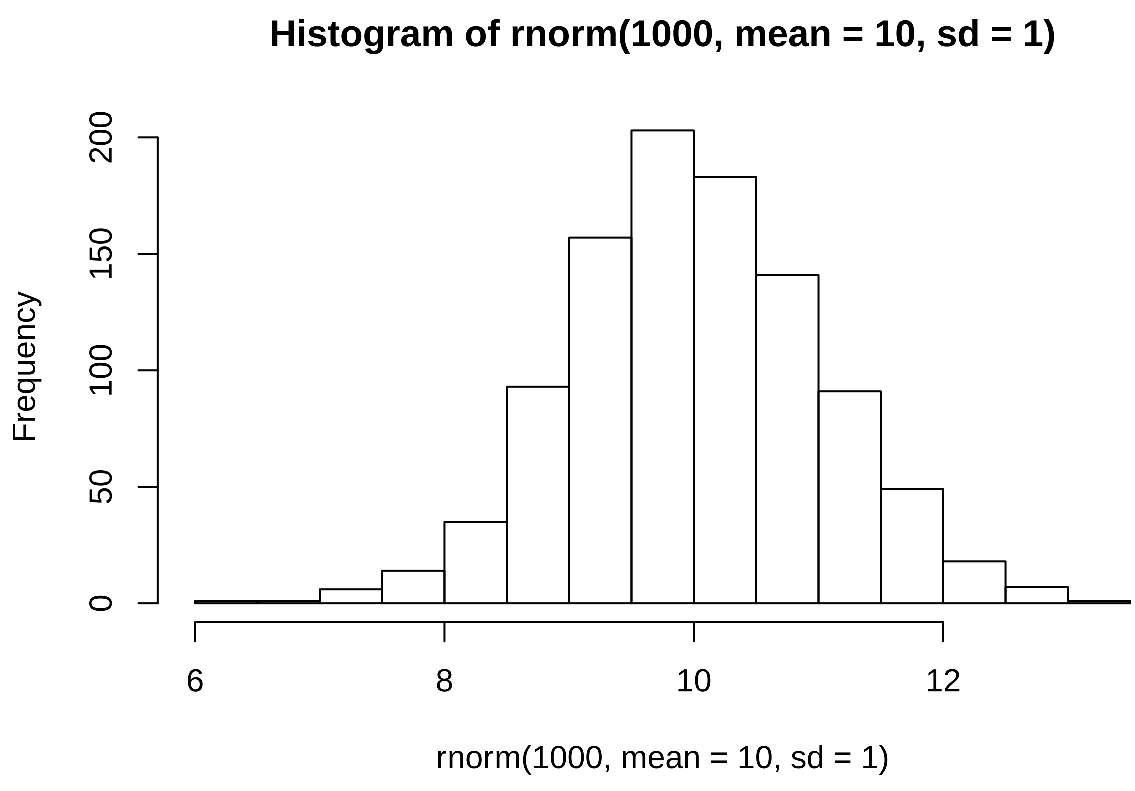
b. Out of 1000 samples, how many do you think are that great than 10? Check your estimation.

```r
a <- rnorm(1000, mean=10, sd=1)
a <- a[a>10]
length(a)
```

```
## [1] 518
```

c. Use hist() function to show the histogram of the sample.

```r
hist(rnorm(1000, mean=10, sd=1))
```



d. Estimate P(X>1) where X~N(2,1)

```r
c <- rnorm(10000, mean=2, sd=1)
c <- c[c>1]
length(c)/10000
```

```
## [1] 0.8411
```

3. Consider an experiment of tossing a fair dice.

a. Use the sample (with replacement) function to generate a sample of 1000 values from the experiment.

```
dice <- c(1:6)
a <- sample(dice, 1000, replace=TRUE)
```

b. Calculate the mean and standard deviation of the sample.

```
dice <- c(1:6)
a <- sample(dice, 1000, replace=TRUE)
mean(a)
```

```
## [1] 3.436
```

```
sd(a)
```

```
## [1] 1.735776
```

c. How many times the 6 occured?

```
dice <- c(1:6)
a <- sample(dice, 1000, replace=TRUE)
a <- a[a=6]
length(a)
```

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

d. Use table function to show the frequency of the values.

```
dice <- c(1:6)
a <- sample(dice, 1000, replace=TRUE)
table(a)
```

```
## a
##   1   2   3   4   5   6
## 156 152 157 178 178 179
```

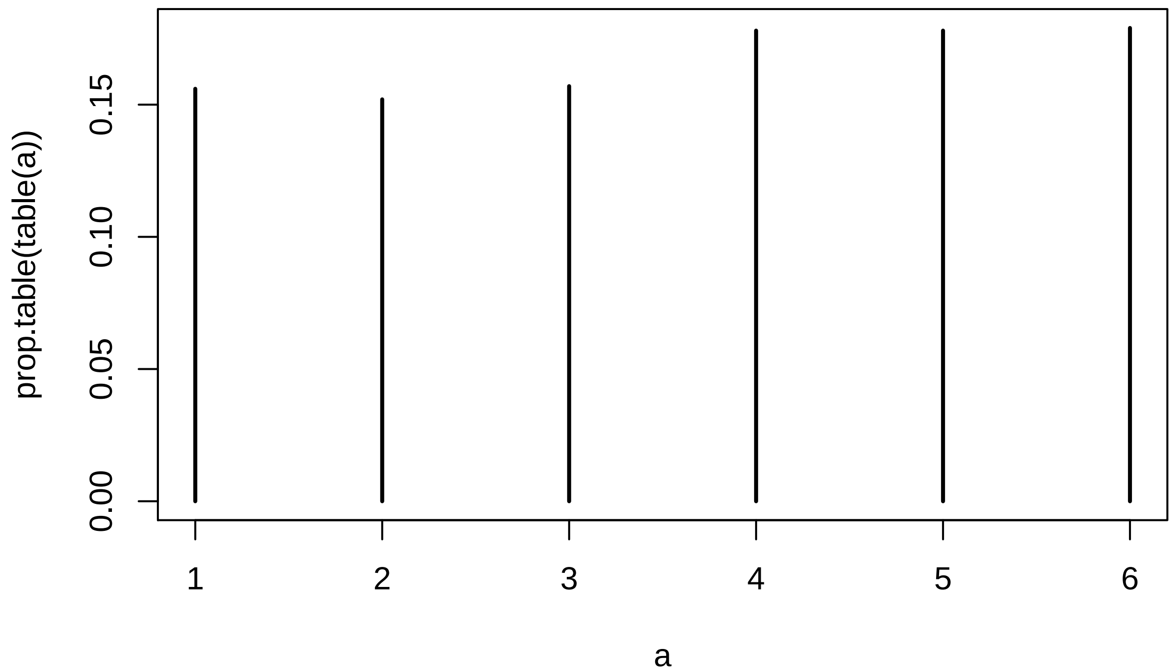
e. Use prop.table(table()) to show the relative frequency of the values.

```
prop.table(table(a))
```

```
## a
##     1     2     3     4     5     6
## 0.156 0.152 0.157 0.178 0.178 0.179
```

f. Plot the frequency of the values.

```r
plot(prop.table(table(a)))
```



4. Consider an experiment of tossing a dice 3 times. Let X1 , X2, and X3 be the number of tossing each dice

a. P(X1 > X2 + X3)

```r
x1 <- sample(dice, 1000, replace = TRUE)
x2 <- sample(dice, 1000, replace = TRUE)
x3 <- sample(dice, 1000, replace = TRUE)
sum(x1>x2+x3)/1000
```

```
## [1] 0.089
```

b. P(X1^2 > X2 + X3)

```r
x1 <- sample(dice, 1000, replace = TRUE)
x2 <- sample(dice, 1000, replace = TRUE)
x3 <- sample(dice, 1000, replace = TRUE)
sum(x1^2>x2+x3)/1000
```

```
## [1] 0.623
```

5. Using simulation, estimate the probability of getting three tails in a row when tossing a coin 3 times. Hint: one way is to generate a matrix with three columns where each rows is an observation of tossing a coin three times.

```
coin <-c(1:2)
f1 <-sample(coin, 1000, replace=TRUE)
f2 <-sample(coin, 1000, replace=TRUE)
f3 <-sample(coin, 1000, replace=TRUE)
m <-data.frame(f1,f2,f3)
j <-rowSums(m)
sum(j==6)/1000
```

```
## [1] 0.114
```

6. (Extra Credits/Optional) Using simulation, estimate the probability of getting three tails in a row when tossing a coin 10 times.

7. Central Limit Theorem (CLT). The CLT said that the mean of a sample of a distribution A (no matter what A is) follows normal distribution with the same mean as A. Following the below steps to confim the CLT when A is uniform distribution.

a. Generate 100 samples of uniform distibution from 0 to 1. Each sample has 1000 observations. Use the runif function to do this.

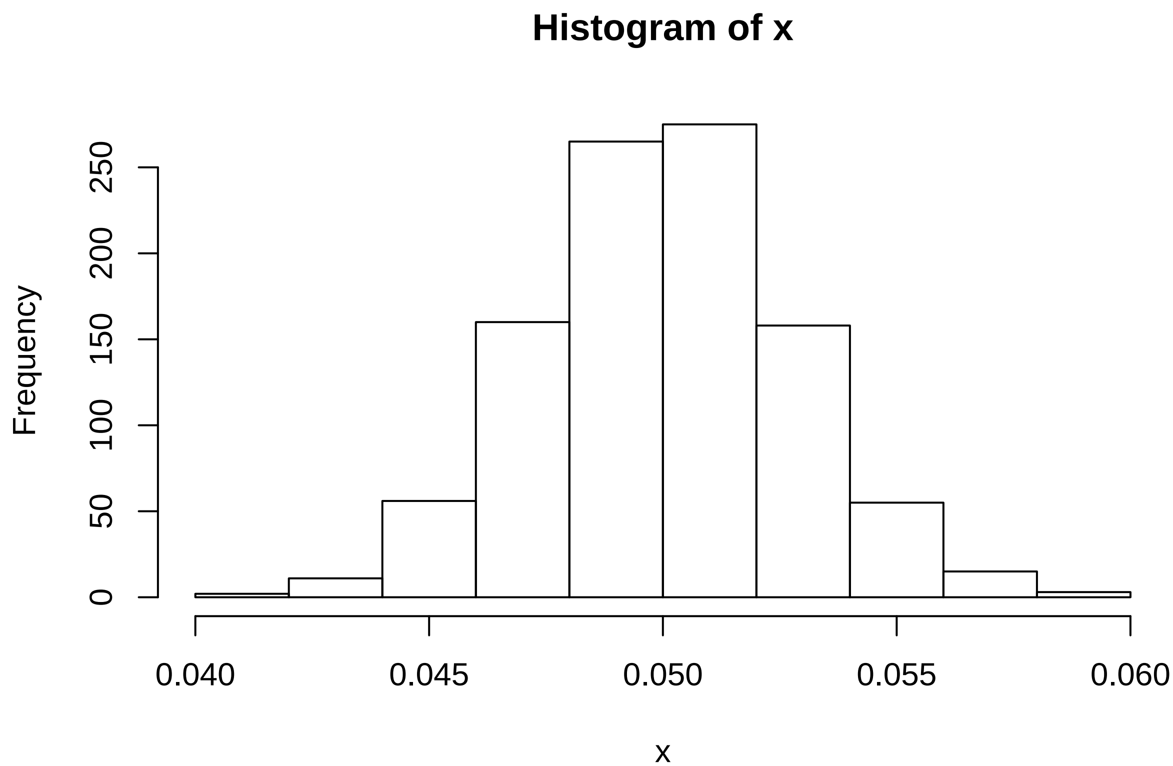
```
s<-100
obs<-1000
a <-matrix(runif(s*obs,0,1),ncol=s)
```

b. Compute the means of the 100 samples. Create vector x containing these means. Hint: You want to put all the samples in a matrix and use rowSums or colSums function.

```
b <-rowSums(a)
x <- b/1000
```

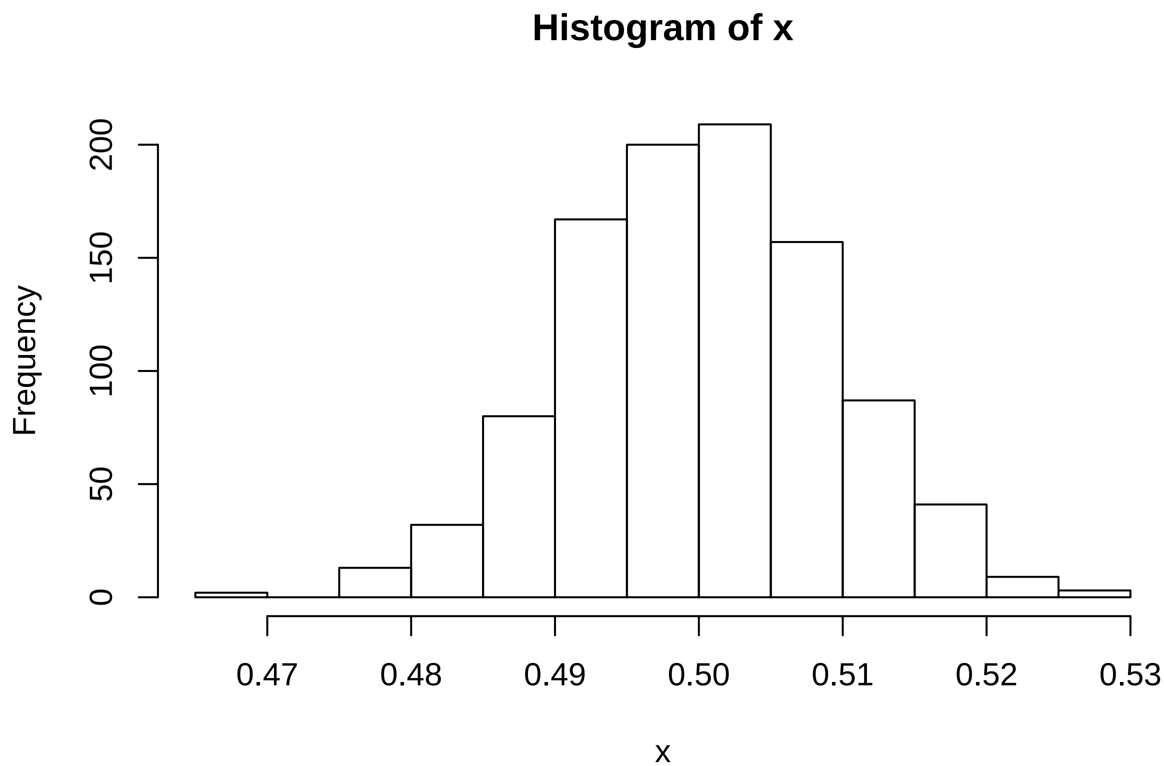
c. By CLT, x must follow normal distribution. Check this by plotting the histogram of x. Does it look like normal distribution? Use hist(x) to plot the histogram of x.

```
hist(x)
```



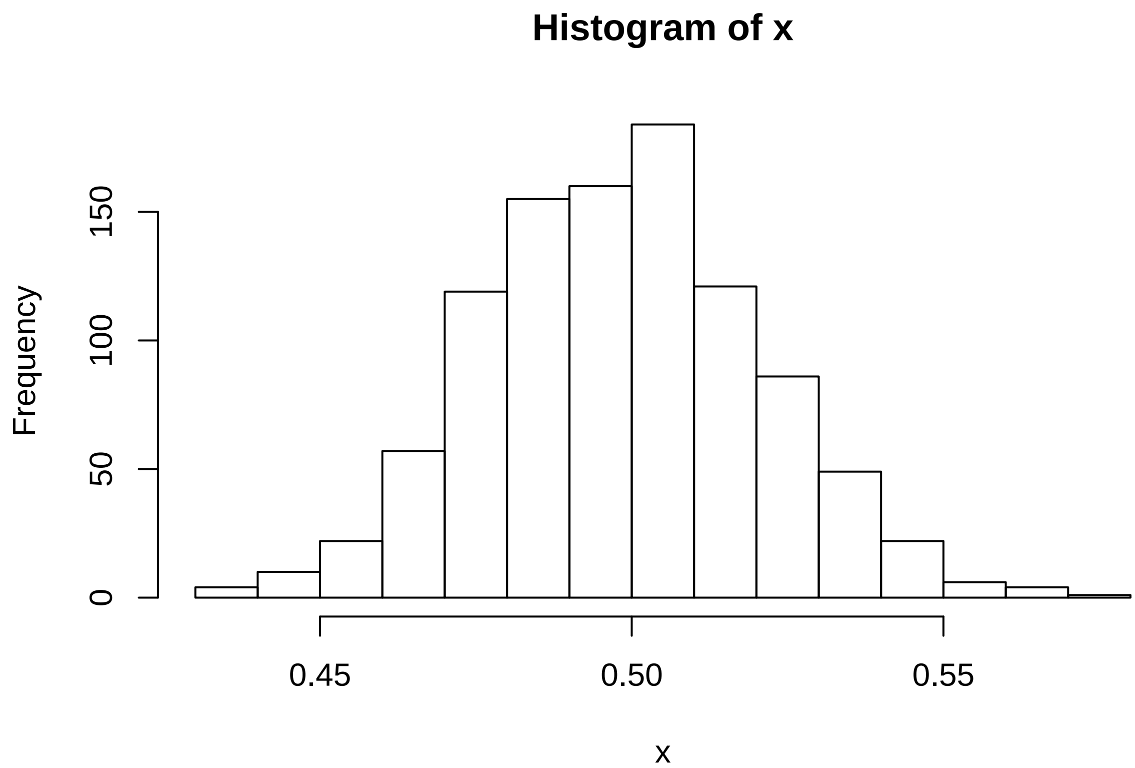
d.Increase the number (100 and 1000) to see if the distribution of x looks more like normal distribution.

```
s<-1000
obs<-1000
a <-matrix(runif(s*obs,0,1),ncol=s)
b <-rowSums(a)
x <- b/1000
hist(x)
```

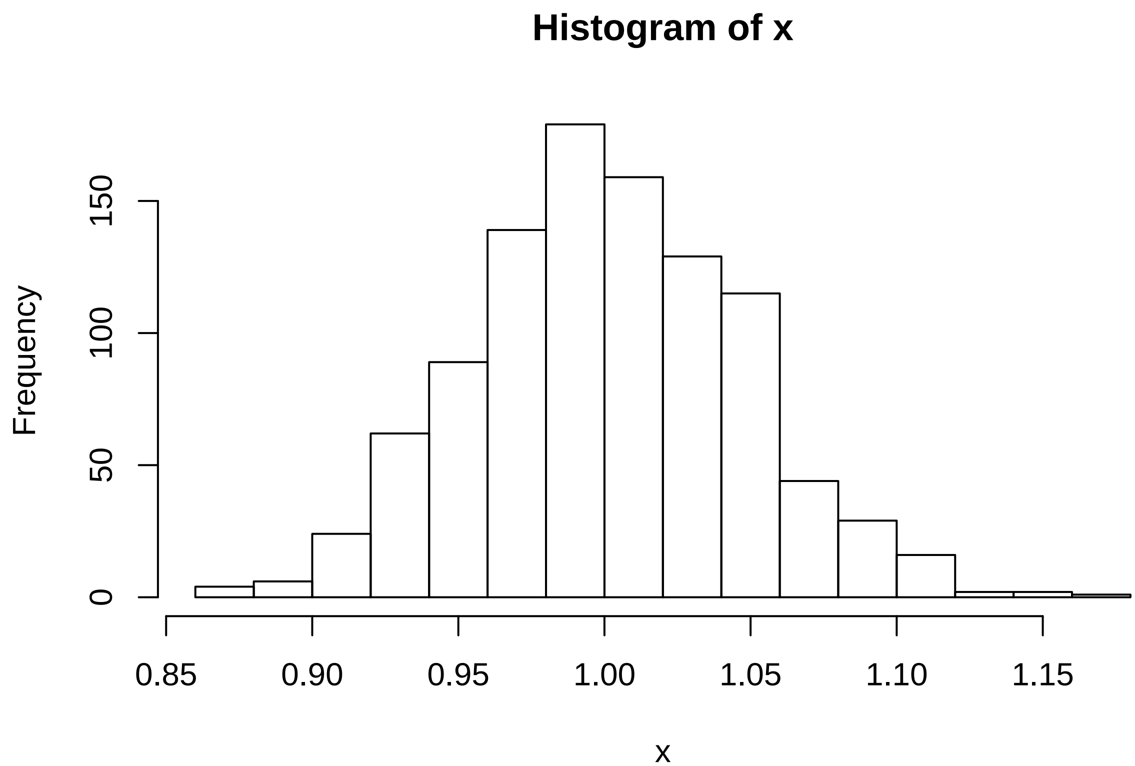


e.Try the same procedure with two other distributions for A.

```
#poisson
s<-1000
obs<-1000
a <-matrix(rpois(s*obs,0.5),ncol=s)
b <-rowSums(a)
x <- b/1000
hist(x)
```




```
#geometric
s<-1000
obs<-1000
a <-matrix(rgeom(s*obs,0.5),ncol=s)
b <-rowSums(a)
x <- b/1000
hist(x)
```



Part 2

7. Use read.csv function to read in the titanic dataset. You can find the dataset on Blackboard or at Kaggle.com. Use str function to see a summary of the data.

```
titanic<-read.csv("C:\\Users\\student\\Desktop\\Fall2019\\R\\titanic.csv")
str(titanic)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

8. Use knitr::kable function to nicely print out the first 10 rows of the data in markdown.

```r
knitr::kable(head(titanic))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. Owen Harris	male	22	1	0
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0
3	1	3	Heikkinen, Miss. Laina	female	26	0	0
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0
5	0	3	Allen, Mr. William Henry	male	35	0	0
6	0	3	Moran, Mr. James	male	NA	0	0

9. Use is.na function and sum function to count the total number of missing values in the data. Count the number of missing values in each columns.

```r
colSums(is.na(titanic))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

10. Calculate the average Age of the passengers. You may want to use the parameter na.rm = TRUE in the function mean

```r
ageavg<-colMeans(titanic['Age'], na.rm=TRUE)
```

11. Replace the missing values of age by the average age calculated previously.

```r
titanic[is.na(titanic['Age']), 'Age'] <- ageavg
knitr::kable(head(titanic))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
1	0	3	Braund, Mr. Owen Harris	male	22.00000	1	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.00000	1	
3	1	3	Heikkinen, Miss. Laina	female	26.00000	0	
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00000	1	
5	0	3	Allen, Mr. William Henry	male	35.00000	0	
6	0	3	Moran, Mr. James	male	29.69912	0	

12. Remove columns Name, PassengerID, Ticket, and Cabin.

```
titanic <- subset(titanic, select = -c(Name,PassengerId,Ticket,Cabin))
knitr::kable(head(titanic))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.00000	1	0	7.2500	S
1	1	female	38.00000	1	0	71.2833	C
1	3	female	26.00000	0	0	7.9250	S
1	1	female	35.00000	1	0	53.1000	S
0	3	male	35.00000	0	0	8.0500	S
0	3	male	29.69912	0	0	8.4583	Q

13. Calculate the mean age of female passengers

```
mean(titanic$Age[titanic$Sex=='female'])
```

```
## [1] 28.21673
```

14. Calculate the median fare of the passengers in Class 1

```
median(titanic$Fare[titanic$Pclass==1])
```

```
## [1] 60.2875
```

15. Calculate the median fare of the female passengers that are not in Class 1

```
median(titanic$Fare[titanic$Pclass!=1 & titanic$Sex=='female'])
```

```
## [1] 14.45625
```

16. Calculate the median age of survived passengers who are female and Class 1 or Class 2,

```
median(titanic$Age[(titanic$Pclass==1| titanic$Pclass==2) & titanic$Sex=='female' & titanic$Survived==1]
```

```
## [1] 30
```

17. Calculate the mean fare of female teenagers survived passengers

```
mean(titanic$Fare[titanic$Sex=='female' & titanic$Age>12 & titanic$Age<20 & titanic$Survived == 1])
```

```
## [1] 49.17966
```

18. Calculate the mean fare of female teenagers survived passengers for each class

```
titanic2<-subset(titanic, Sex=='female' & titanic$Age>12 & titanic$Age<20 & titanic$Survived == 1)
aggregate(titanic2[,'Fare'], list(titanic2$Pclass),mean)
```

```
##   Group.1          x
## 1       1 107.540708
## 2       2  20.008850
## 3       3   8.769885
```

19. Calculate the ratio of Survived and not Survived for passengers who are who pays more than the average fare

```
avgFare<-mean(titanic$Fare)
titanic3<-subset(titanic, Fare > avgFare)
print("Survived")
```

```
## [1] "Survived"
```

```
sum(titanic3$Survived==1)/sum(titanic3$Survived==1 | titanic3$Survived==0)
```

```
## [1] 0.5971564
```

```
print("Did Not Survive")
```

```
## [1] "Did Not Survive"
```

```
sum(titanic3$Survived==0)/sum(titanic3$Survived==1 | titanic3$Survived==0)
```

```
## [1] 0.4028436
```

20. Add column that standardizes the fare (subtract the mean and divide by standard deviation) and name it sfare

```
sdFare<-sd(titanic$Fare)
titanic$sfare<-(titanic$Fare-avgFare)/sdFare
head(titanic)
```

```
##   Survived Pclass    Sex      Age SibSp Parch    Fare Embarked      sfare
## 1        0      3   male 22.00000     1     0  7.2500        S -0.5021631
## 2        1      1 female 38.00000     1     0 71.2833        C  0.7864036
## 3        1      3 female 26.00000     0     0  7.9250        S -0.4885799
## 4        1      1 female 35.00000     1     0 53.1000        S  0.4204941
## 5        0      3   male 35.00000     0     0  8.0500        S -0.4860644
## 6        0      3   male 29.69912     0     0  8.4583        Q -0.4778481
```

21. Add categorical variable named cfare that takes value cheap for passengers paying less the average fare and takes value expensive for passengers paying more than the average fare.

```
titanic$cfare <- ifelse(titanic$Fare < avgFare, "cheap", 'expensive')
```

22. Add categorical variable named cage that takes value 0 for age 0-10, 1 for age 10-20, 2 for age 20-30, and so on

```
titanic$cage <-memisc::cases('0'=titanic$Age<10,
                    '1'=titanic$Age<20,
                    '2'=titanic$Age<30,
                    '3'=titanic$Age<30,
                    '4'=titanic$Age<40,
                    '5'=titanic$Age<50,
                    '6'=titanic$Age<60,
                    '7'=titanic$Age<70,
                    '8'=titanic$Age<80,
                    '9'=titanic$Age<90,
                    '10'=TRUE)
```

```
## Warning in memisc::cases(`0` = titanic$Age < 10, `1` = titanic$Age < 20, :
## conditions are not mutually exclusive
```

23. Show the frequency of Ports of Embarkation. It appears that there are two missing values in the Embarked variable. Assign the most frequent port to the missing ports. Hint: Use the levels function to modify the categories of categorical variables.

```
summary(titanic$Embarked)
```

```
##       C   Q   S
##   2 168  77 644
```

```
titanic$Embarked[titanic$Embarked==""] <- "S"
summary(titanic$Embarked)
```

```
##       C   Q   S
##   0 168  77 646
```