Assignment 1

Jeff Moise

9/9/2019

1. Calculate the Following Sum

S_{1} &= 1 + 2 +...+ 2019 \

```
s1= c(1:2019)
sum(s1)
```

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

S_{2} &= 1^3 + 2^3 +...+ 2019^3 \

```
s2= c(1:2019)
sum(s2^3)
```

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

S_{3} &= $1^{1+2}2+3^{3+\ldots+2019}\{2019\}$ \

```
s3=c(1:2019)
sum(s3^s3)
```

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

S_{4} &= $1^{1-2}2+3^{3-4}4+\ldots-2018^{\{2018\}+2019}\{2019\}$ \

```
s4=c(1:2019)
b=s4^s4
c=c(1,-1)
sum(b * c)
```

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

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

S_{5} &= 1+frac{1}{4}+frac{1}{9}+frac{1}{16}+frac{1}{25}+...

```
s5=c(2:1000000)
sum(s5/(s5^2))
```

```
## [1] 13.39273
```

S_{6} &= 1+frac{1}{2}+frac{1}{3}+frac{1}{4}+...

```r
s6= c(1:1000000)
sum(1/s6)
```

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

S_{7} &= 1+frac{1}{8}+frac{1}{27}+frac{1}{64}+...

```r
s7 = c(1:100000)
sum(s7/s7^3)
```

```
## [1] 1.644924
```

S_{8} &= 1-frac{1}{2}+frac{1}{3}-frac{1}{4}+...

```r
s8=c(1:100000)
b=s8/(s8+1)
c=c(1,-1)
sum(b * c)
```

```
## [1] -0.3068478
```

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.
   b. Out of 1000 samples, how many do you think are that great than 10? Check your estimation.
   c. Use `hist()` function to show the histogram of the sample. d Estimate $P(X > 1)$, where $X \sim N(2, 1)$

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

```
## [1] 9.967765
```

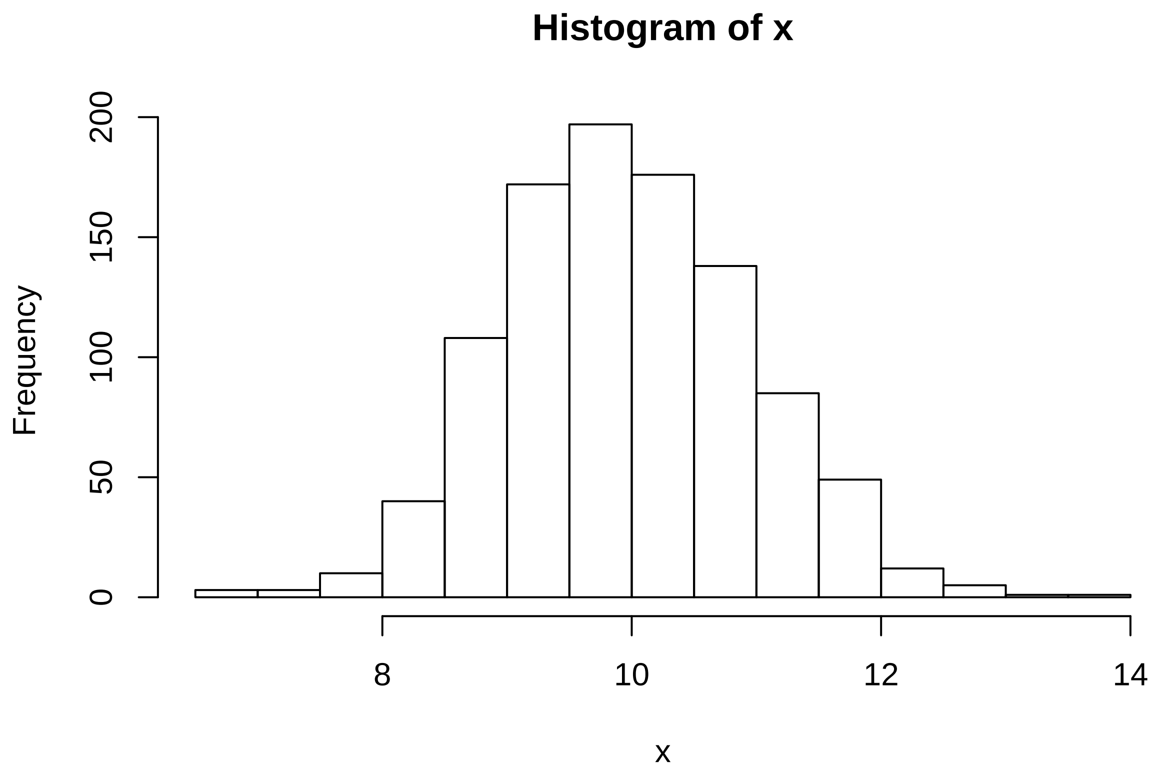
```r
sd(x)
```

```
## [1] 0.992843
```

```r
b=x>10
sum(b)
```

```
## [1] 467
```

```r
hist(x)
```




```
y=rnorm(1000,mean=2,sd=1)
z=y>1
sum(z)/1000
```

```
## [1] 0.83
```

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

   c. How many times the 6 occured?
   d. Use `table` function to show the frequency of the values.
   e. Use `prop.table(table())` to show the relative frequency of the values.
   f. Plot the frequency of the values.

```
dice = c(1:6)
x=sample(dice,1000,replace=TRUE)
mean(x)
```

```
## [1] 3.471
```

```r
sd(x)
```

```
## [1] 1.692342
```

```r
sum(x=6)
```

```
## [1] 6
```

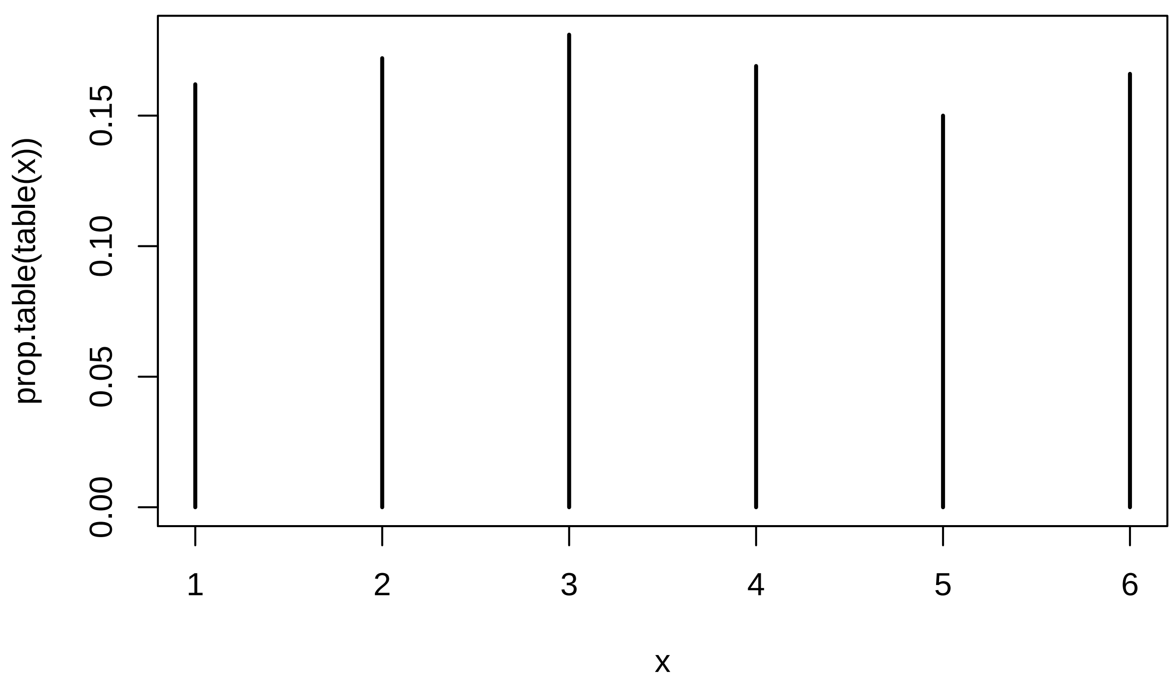
```r
table(x)
```

```
## x
##   1   2   3   4   5   6
## 162 172 181 169 150 166
```

```r
prop.table(table(x))
```

```
## x
##     1     2     3     4     5     6
## 0.162 0.172 0.181 0.169 0.150 0.166
```

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



4. Consider an experiment of tossing a dice 3 times. Let X_1, X_2, and X_3 be the number of tossing the first time, second time and third time, respectively. Use simulation to estimate the following probabilities: a. $P(X_1 > X_2 + X_3)$ b. $P(X_1^2 > X_2^2 + X_3^2)$

```r
x=c(1:6)
x_1=sample(x,1000,replace=TRUE)
x_2=sample(x,1000,replace=TRUE)
x_3=sample(x,1000,replace=TRUE)
sum(x_1>x_2+x_3)/1000
```

```
## [1] 0.088
```

```r
sum(x_1^2>x_2^2+x_3^2)/1000
```

```
## [1] 0.215
```

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.

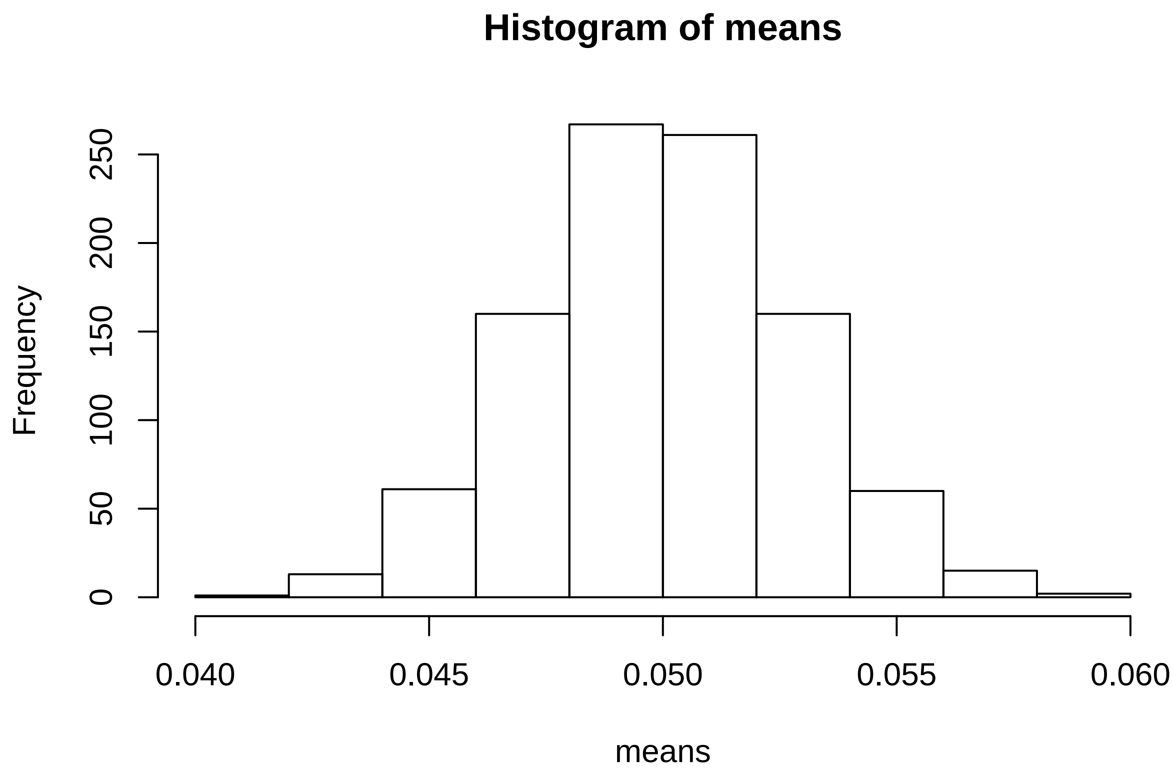
```r
x<-c(0:1)
a<-sample(x,1000,replace=TRUE)
b<-sample(x,1000,replace=TRUE)
c<-sample(x,1000,replace=TRUE)
z<-data.frame(a,b,c)
j<-rowSums(z)
sum(j==3)/1000
```

```
## [1] 0.117
```

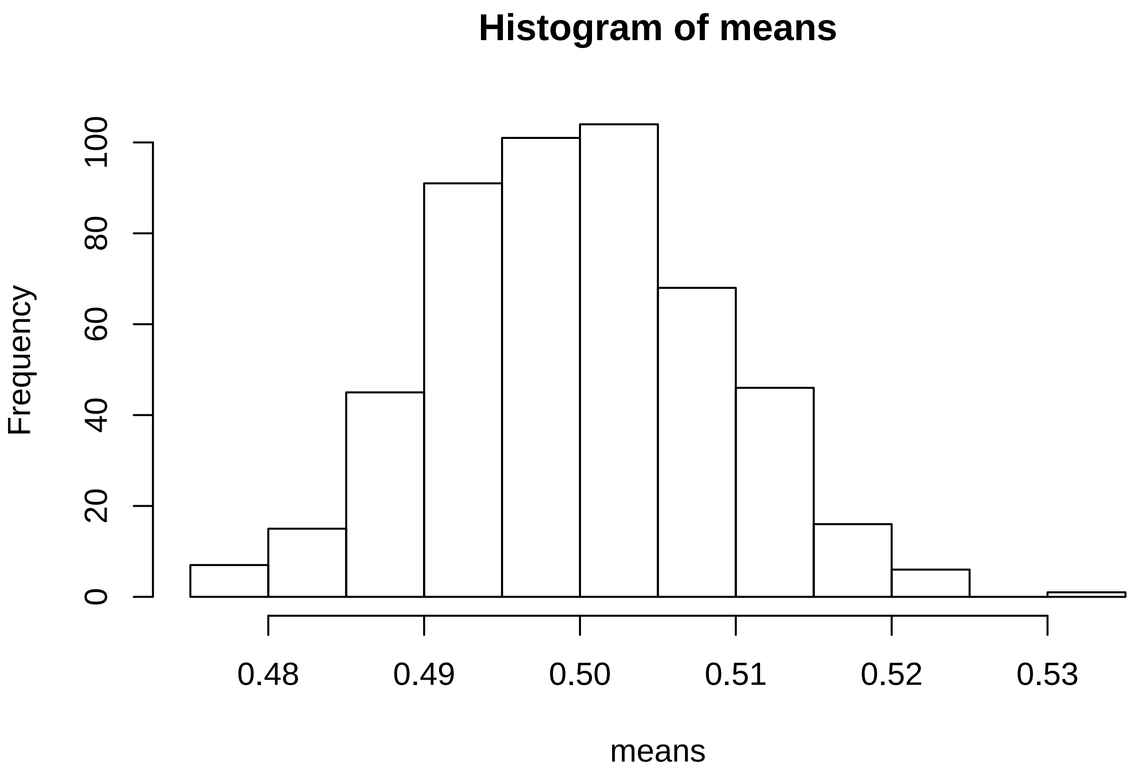
6. (**Extra Credits/Optional**) Using simulation, estimate the probability of getting three tails in a row when tossing a coin 10 times.

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. - Generate 100 samples of uniform distibution from 0 to 1. Each sample has 1000 observations. Use the **runif** function to do this. - 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. - 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. - Increase the number (100 and 1000) to see if the distribution of x looks more like normal distribution. - Try the same procedure with two other distributions for A.

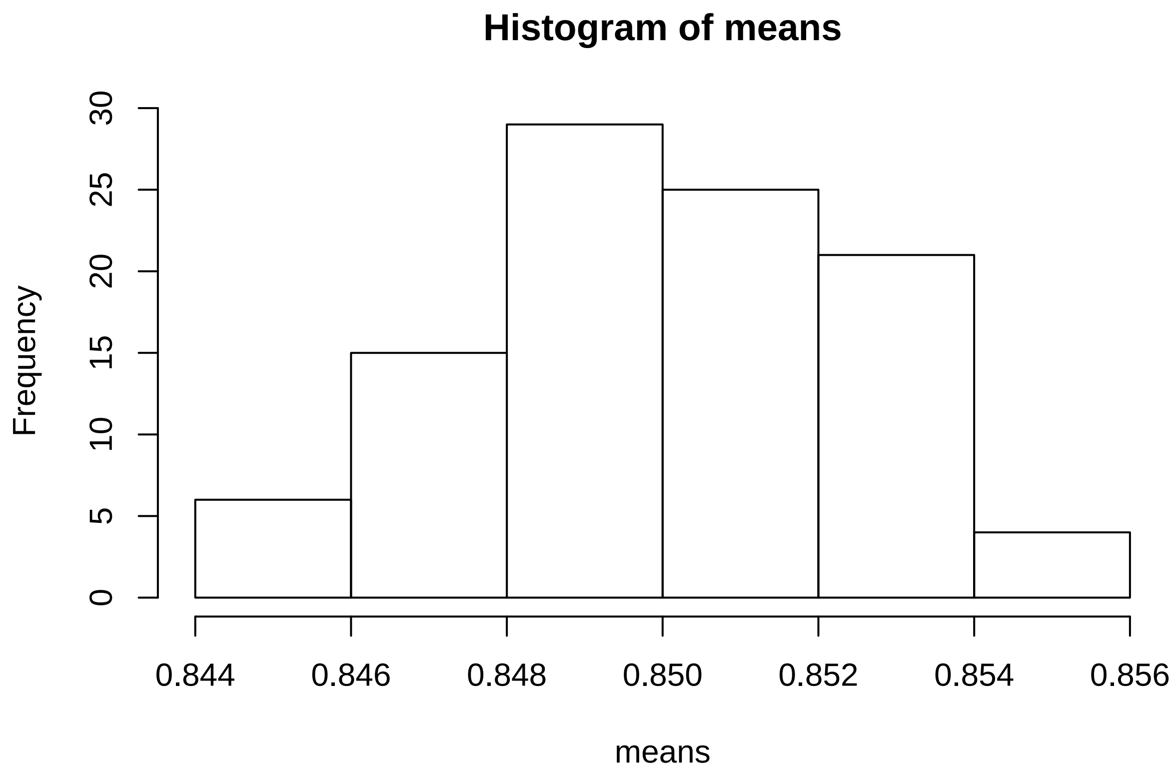
```r
a<- matrix(runif(100*1000,0,1),ncol=100)

y <- rowSums(a)
means = y/1000

hist(means)
```



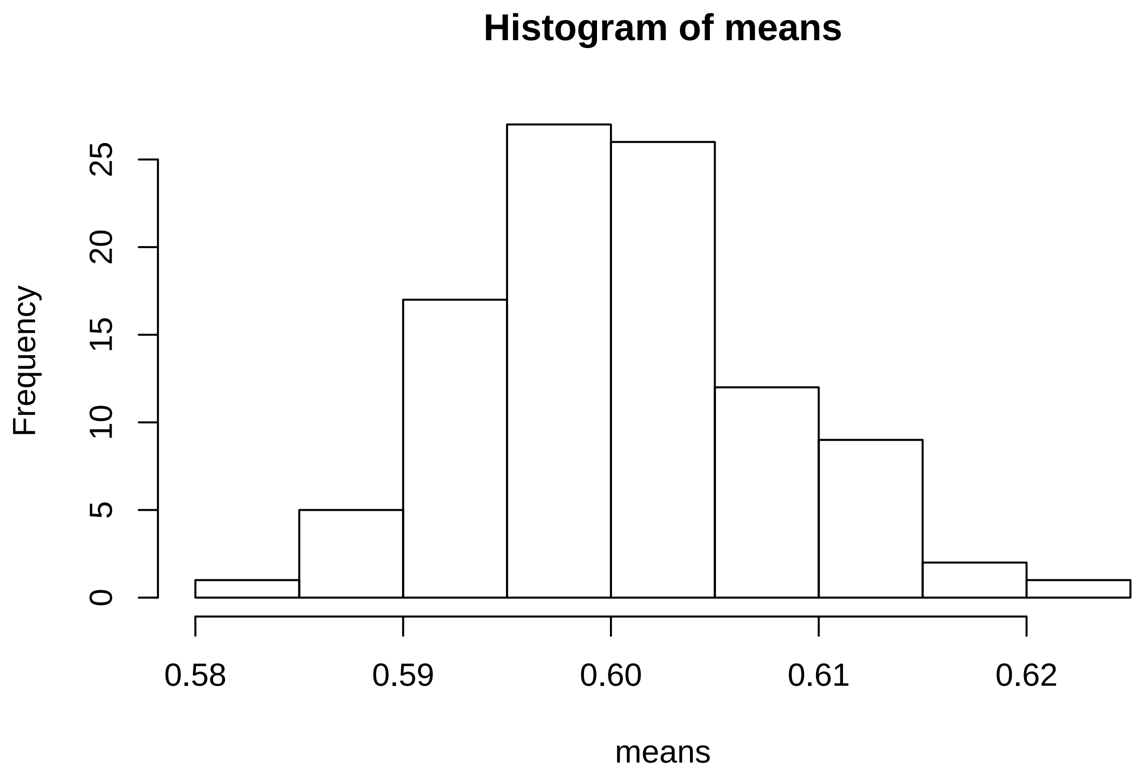

```r
#Increase the numbers

a<-matrix(runif(1000*500,0,1),ncol=1000)

y<-rowSums(a)
means= y/1000
hist(means)
```




```
#two other distributions for A

a<-matrix(runif(100*1000,.7),ncol=1000)

y<-rowSums(a)
means= y/1000
hist(means)
```




```
a<-matrix(runif(100*1000,.2),ncol=1000)

y<-rowSums(a)
means= y/1000
hist(means)
```



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

```
df <- read.csv(file="C:\\Users\\student\\Documents\\R\\titanic.csv")
str(df)
```

```
## '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 ...
```

```
knitr::kable(head(df))
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibS
1	0	3	Braund, Mr. Owen Harris	male	22	

PassengerId	Survived	Pclass	Name	Sex	Age	SibS
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	
3	1	3	Heikkinen, Miss. Laina	female	26	
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	
5	0	3	Allen, Mr. William Henry	male	35	
6	0	3	Moran, Mr. James	male	NA	
10. Use ‘is.na	‘ function	and `sum`	function to count the total number of missing values	in the da	ta. Co	unt th

```
sum(is.na(df))
```

```
## [1] 177
```

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

```
mean(df$Age, na.rm=TRUE)
```

```
## [1] 29.69912
```

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

```
df$Age.imp.mean <- ifelse(is.na(df$Age), mean(df$Age, na.rm=TRUE), df$Age)
sum(is.na(df$Age.imp.mean))
```

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

```
df$Age=df$Age.imp.mean
sum(is.na(df$Age))
```

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

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

```
drop <- c("Name","PassengerID","Ticket","Cabin")
df = df[,!(names(df) %in% drop)]
str(df)
```

```
## 'data.frame':    891 obs. of  10 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 ...
##  $ Sex         : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age         : num  22 38 26 35 35 ...
##  $ SibSp       : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch       : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare        : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked    : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ Age.imp.mean: num  22 38 26 35 35 ...
```

13. Calculate the mean age of female passengers

```r
mean(df$Age[df$Sex=="female"])
```

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

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

```r
median(df$Fare[df$Pclass=='1'])
```

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

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

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

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

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

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

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

17. Calculate the mean fare of female teenagers survived passengers

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

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

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

```r
df1 = df[((df$Survived==1)&(df$Sex=='female')& (df$Age>12) & (df$Age<20)),]

aggregate(df1$Fare,by=list(df1$Pclass),FUN=mean)
```

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

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

```r
meanFare<-mean(df$Fare)
titanicsub= subset(df,Fare>meanFare)
sum(titanicsub$Survived ==1)/ sum(titanicsub$Survived==1 | titanicsub==0)
```

```
## [1] 0.08854533
```

```r
sum(titanicsub$Survived ==0)/ sum(titanicsub$Survived==1 | titanicsub==0)
```

```
## [1] 0.05973296
```

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

```r
avgFare = mean(df$Fare)
sdFare = sd(df$Fare)
df$sfare <- (df$Fare-avgFare)/sdFare
head(df$sfare)
```

```
## [1] -0.5021631  0.7864036 -0.4885799  0.4204941 -0.4860644 -0.4778481
```

22. 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.

```r
df$cfare <- ifelse(df$Fare < avgFare, 'cheap','expensive')
head(df$cfare)
```

```
## [1] "cheap"     "expensive" "cheap"     "expensive" "cheap"     "cheap"
```

23. 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

```r
df$cage <- 100
df$cage[df$Age<=10] <- 0
df$cage[df$Age>10 & df$Age<=20] <- 1
df$cage[df$Age>20 & df$Age<=30] <- 2
df$cage[df$Age>30] <- 3
head(df$cage)
```

```
## [1] 2 3 2 3 3 2
```

24. 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.

```r
summary(df$Embarked)
```

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

```r
df$Embarked[df$Embarked==""] = "S"
summary(df$Embarked)
```

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