Pns lab assignmnets 5-8

Code

VARADA GUPTA ROLL NUMBRE:102103542 BATCH:3CO2

ASSIGNMNET 5

1. Consider that X is the time (in minutes) that a person has to wait in order to take a flight. If each flight takes off each hour X ~ U(0, 60). Find the probability that

a. waiting time is more than 45 minutes, and

Hide

```
punif(45,0,60,lower.tail=FALSE)
```

```
[1] 0.25
```

b. waiting time lies between 20 and 30 minutes.

Hide

```
punif(30,0,60)-punif(20,0,60)
```

```
[1] 0.1666667
```

2. The time (in hours) required to repair a machine is an exponential distributed random variable with parameter λ = 1/2.

a. Find the value of density function at x = 3.

Hide

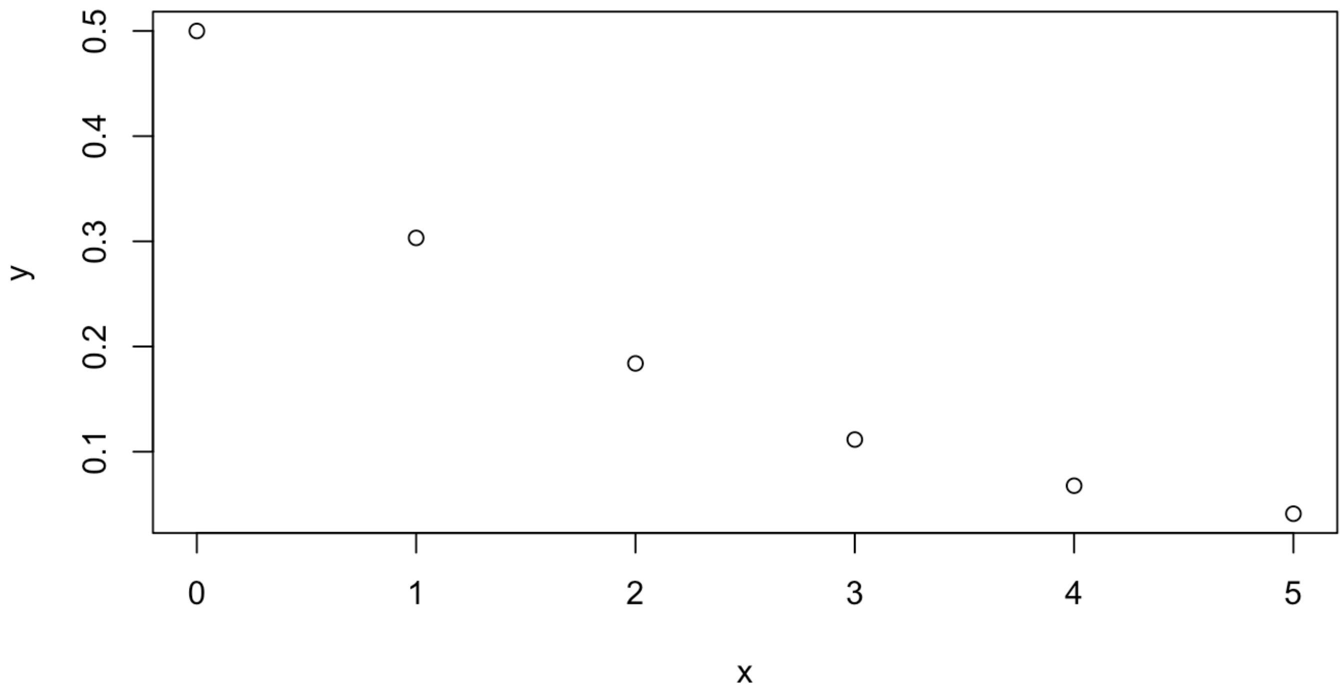
```
dexp(x,1/2)
```

```
[1] 0.50000000 0.30326533 0.18393972 0.11156508 0.06766764 0.04104250
```

b. Plot the graph of exponential probability distribution for 0 ≤ x ≤ 5.

Hide

```
x=seq(0,5)
y=dexp(x,1/2)
plot(x,y)
```



c. Find the probability that a repair time takes at most 3 hours.

Hide

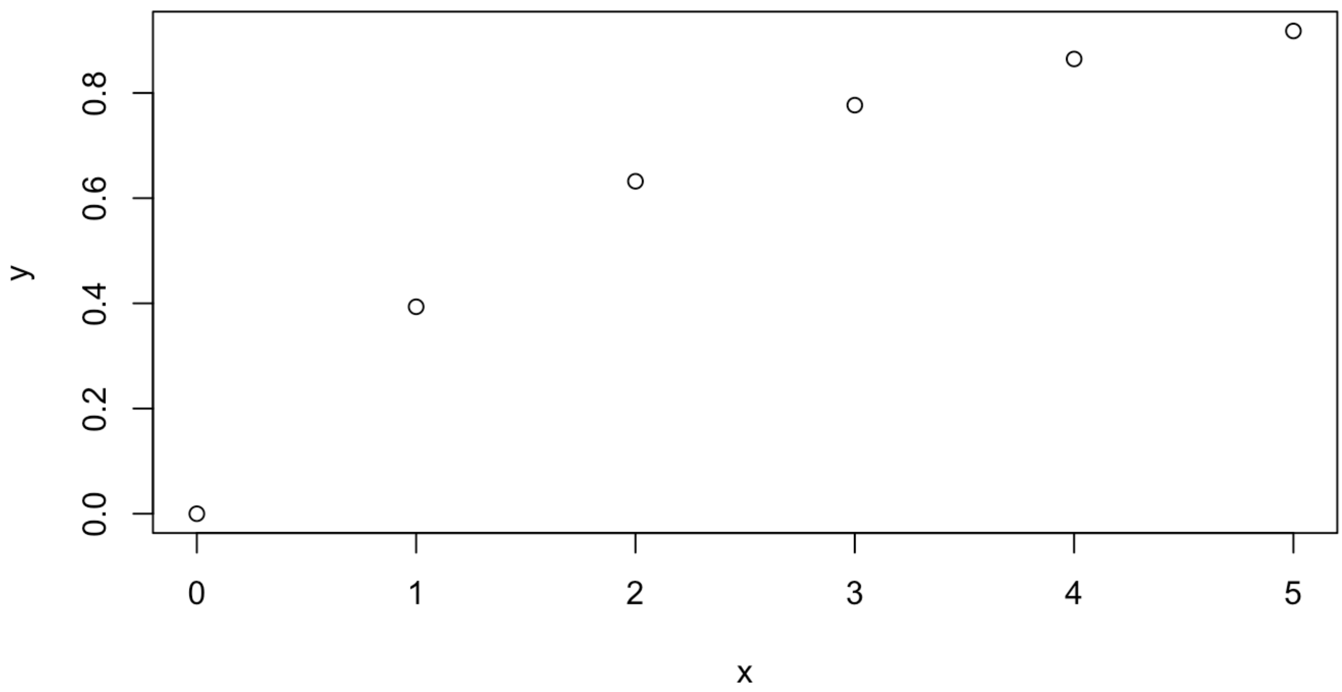
```
pexp(3,1/2)
```

```
[1] 0.7768698
```

d. Plot the graph of cumulative exponential probabilities for $0 \le x \le 5$.

Hide

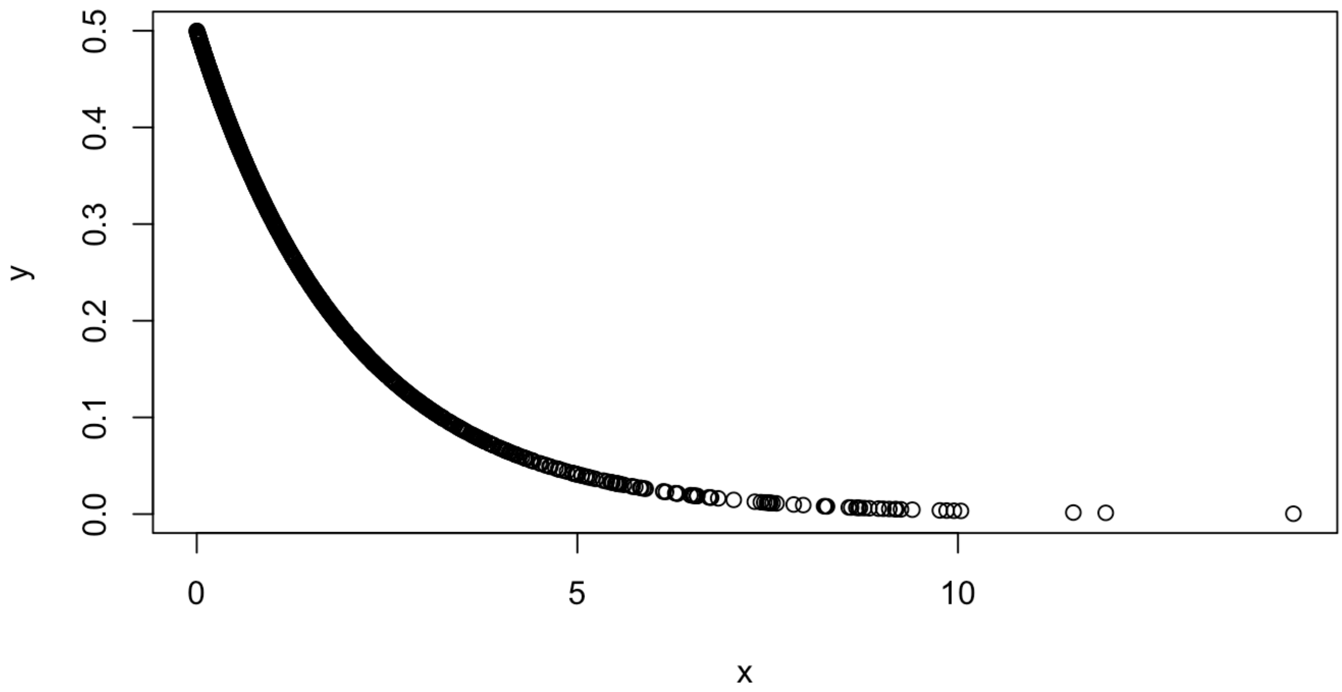
```
x=seq(0,5)
y=pexp(x,1/2)
plot(x,y)
```



e. Simulate 1000 exponential distributed random numbers with λ = 1⁄2 and plot the simulated data.

Hide

```
x=rexp(1000,1/2)
y=dexp(x,1/2)
plot(x,y)
```



3. The lifetime of certain equipment is described by a random variable X that follows Gamma distribution with parameters α = 2 and β = 1/3.

a. Find the probability that the lifetime of equipment is at least 1 unit of time.

Hide

```
pgamma(1,2,1/3,lower.tail=FALSE)
```

```
[1] 0.9553751
```

b. What is the value of c, if P(X ≤ c) ≥ 0.70? (Hint: try quantile function qgamma())

Hide

```
qgamma(0.70,2,1/3)
```

```
[1] 7.317649
```

ASSIGNMNET 6

1. The joint probability density of two random variables X and Y is

f(x, y) =

2(2x + 3y)/5; 0 ≤ x, y ≤ 1 0; elsewhere

Then write a R-code to (i) check that it is a joint density function or not? (Use integral2())

Hide

```
library('pracma')
fx<-function(x,y){
  2*(2*x+3*y)/5
}
i<-integral2(fx,0,1,0,1)
i1<-i$Q
i1
```

```
[1] 1
```

ii. find marginal distribution g(x) at x = 1.

Hide

```
gx<-function(y){
  2*(2+3*y)/5
}
integral(gx,0,1)
```

```
[1] 1.4
```

iii. find the marginal distribution h(y) at y = 0.

Hide

```
hy<-function(x){
  2*(2*x)/5
}
integral(hy,0,1)
```

```
[1] 0.4
```

iv. find the expected value of g(x, y) = xy.

Hide

```
f<-function(x,y){
  2*x*y*(2*x+3*y)/5
}
integral2(f,0,1,0,1)$Q
```

```
[1] 0.3333333
```

2. The joint probability mass function of two random variables X and Y is f(x, y) = {(x + y)/30; x = 0, 1, 2,

3; y = 0, 1, 2}

Then write a R-code to (i) display the joint mass function in rectangular (matrix) form.

Hide

```
f<-function(x,y)(x+y)/30

xs=c(0:3)
ys=c(0:2)

m=matrix(c(f(0,ys),f(1,ys),f(2,ys),f(3,ys)),nrow=4,ncol=3,byrow = TRUE)
print(m)
```

```
           [,1]       [,2]       [,3]
[1,] 0.00000000 0.03333333 0.06666667
[2,] 0.03333333 0.06666667 0.10000000
[3,] 0.06666667 0.10000000 0.13333333
[4,] 0.10000000 0.13333333 0.16666667
```

ii. check that it is joint mass function or not? (use: Sum())

Hide

```
sum(m)
```

```
[1] 1
```

iii. find the marginal distribution g(x) for x = 0, 1, 2, 3. (Use:apply())

Hide

```
gx<-apply(m,1,sum)
gx
```

```
[1] 0.1 0.2 0.3 0.4
```

iv. find the marginal distribution h(y) for y = 0, 1, 2. (Use:apply())

Hide

```
hy<-apply(m,2,sum)
hy
```

```
[1] 0.2000000 0.3333333 0.4666667
```

v. find the conditional probability at x = 0 given y = 1.

Hide

```
cp<-m[1,2]/hy[2]
cp
```

```
[1] 0.1
```

vi. find E(x), E(y), E(xy), V ar(x), V ar(y), Cov(x, y) and its correlation coefficient.

Hide

```
ex<-sum(xs*gx)

ey<-sum(ys*hy)

exy<-sum(outer(xs,ys,"*")*m)

print(paste("E(x):", ex))
```

```
[1] "E(x): 2"
```

Hide

```
print(paste("E(y):", ey))
```

```
[1] "E(y): 1.26666666666667"
```

Hide

```
print(paste("E(xy):", exy))
```

```
[1] "E(xy): 2.4"
```

Hide

```
vx<-sum(x*x*gx)-ex^2
vy<-sum(y*y*hy)-ey^2
print(paste("Var(x):", vx))
```

```
[1] "Var(x): 302.060606060606"
```

Hide

```
print(paste("Var(y):", vy))
```

```
[1] "Var(y): 79.3955555555556"
```

Hide

```
cov_xy <- exy - ex * ey
print(paste("Cov(x, y):", cov_xy))
```

```
[1] "Cov(x, y): -0.133333333333333"
```

Hide

```
correlation_coefficient <- cov_xy / sqrt(vx * vy)
print(paste("Correlation Coefficient:", correlation_coefficient))
```

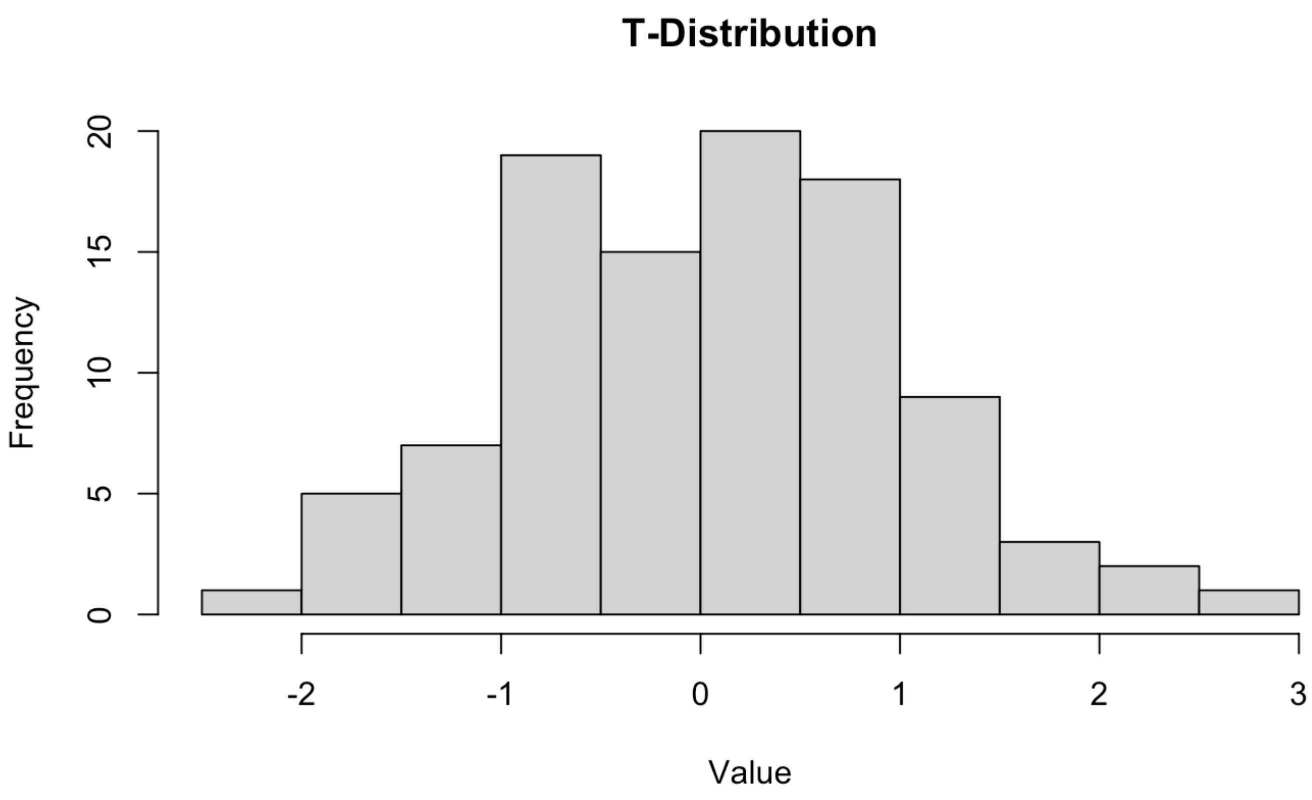
```
[1] "Correlation Coefficient: -0.000860981056104221"
```

ASSIGNMNET 7

1. Use the rt(n, df) function in r to investigate the t-distribution for n = 100 and df = n – 1 and plot the histogram for the same.

Hide

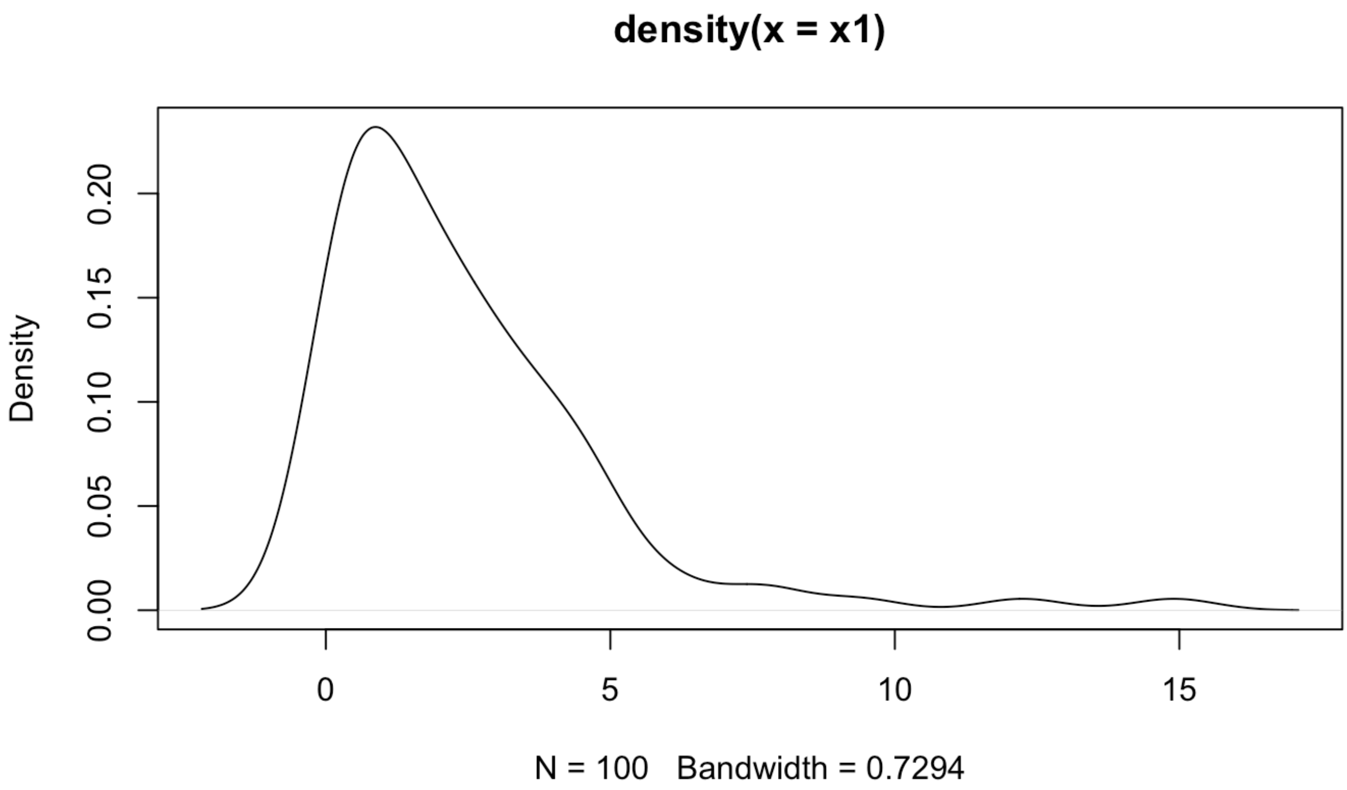
```
x<-rt(100,99)
hist(x,main="T-Distribution",xlab="Value",ylab="Frequency")
```



2. Use the rchisq(n, df) function in r to investigate the chi-square distribution with n = 100 and df = 2, 10, 25.

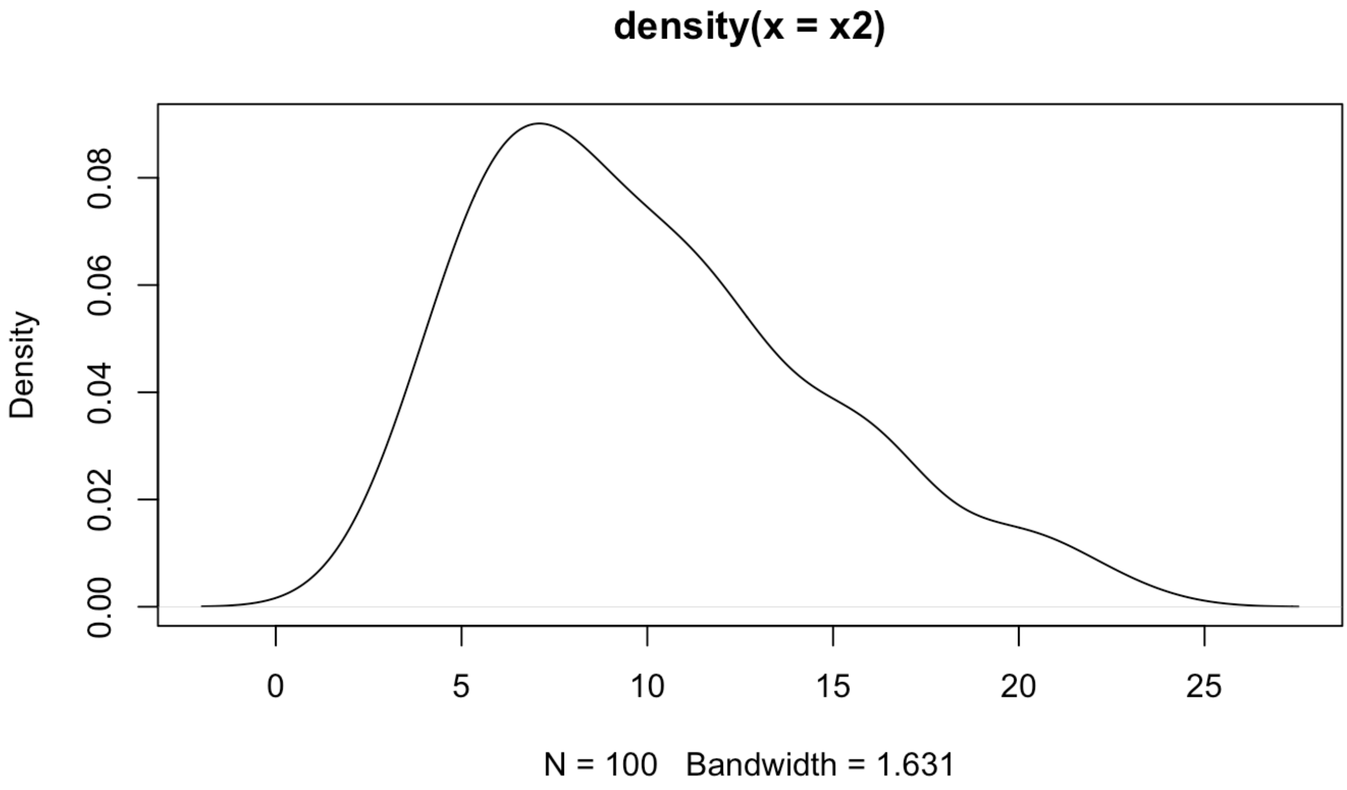
Hide

```
x1=rchisq(100,2)
plot(density.default(x1))
```



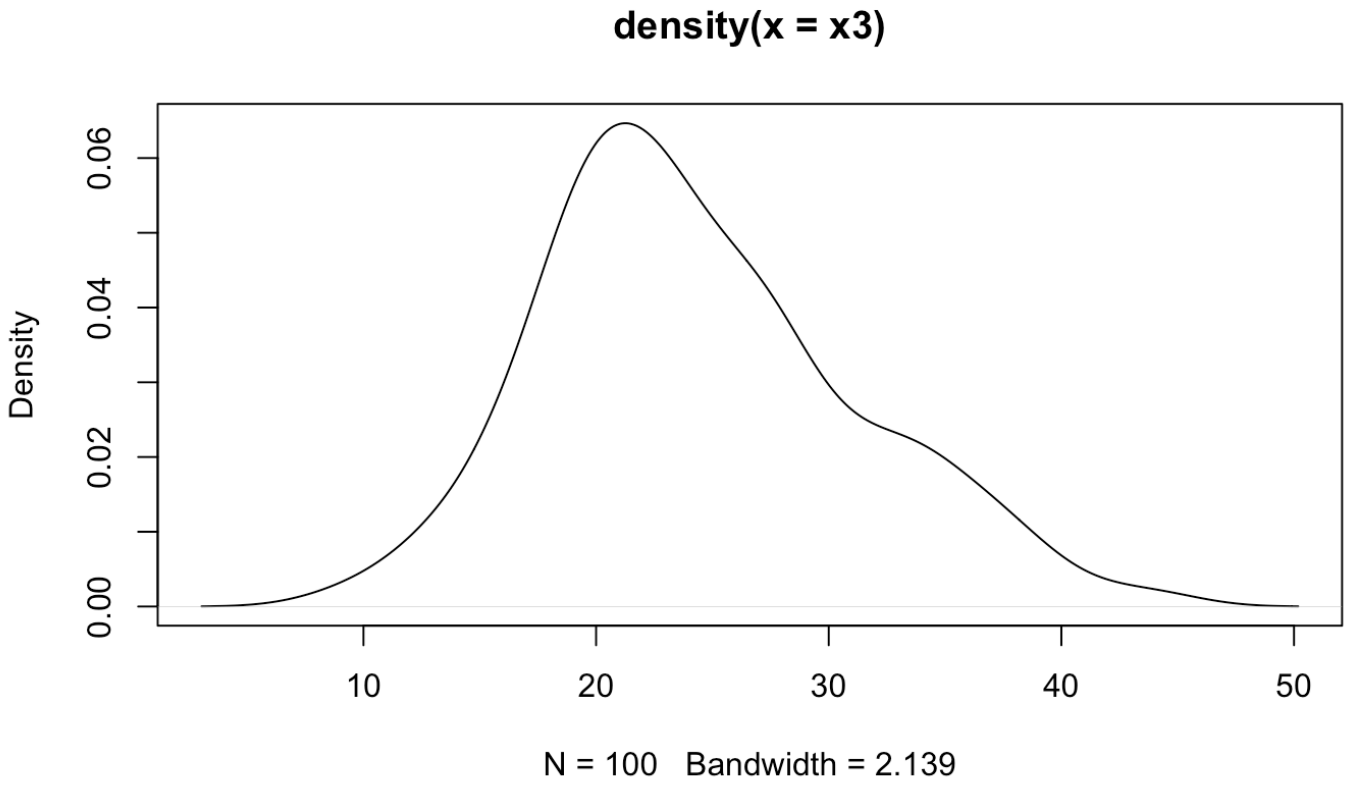
Hide

```
x2=rchisq(100,10)
plot(density.default(x2))
```



Hide

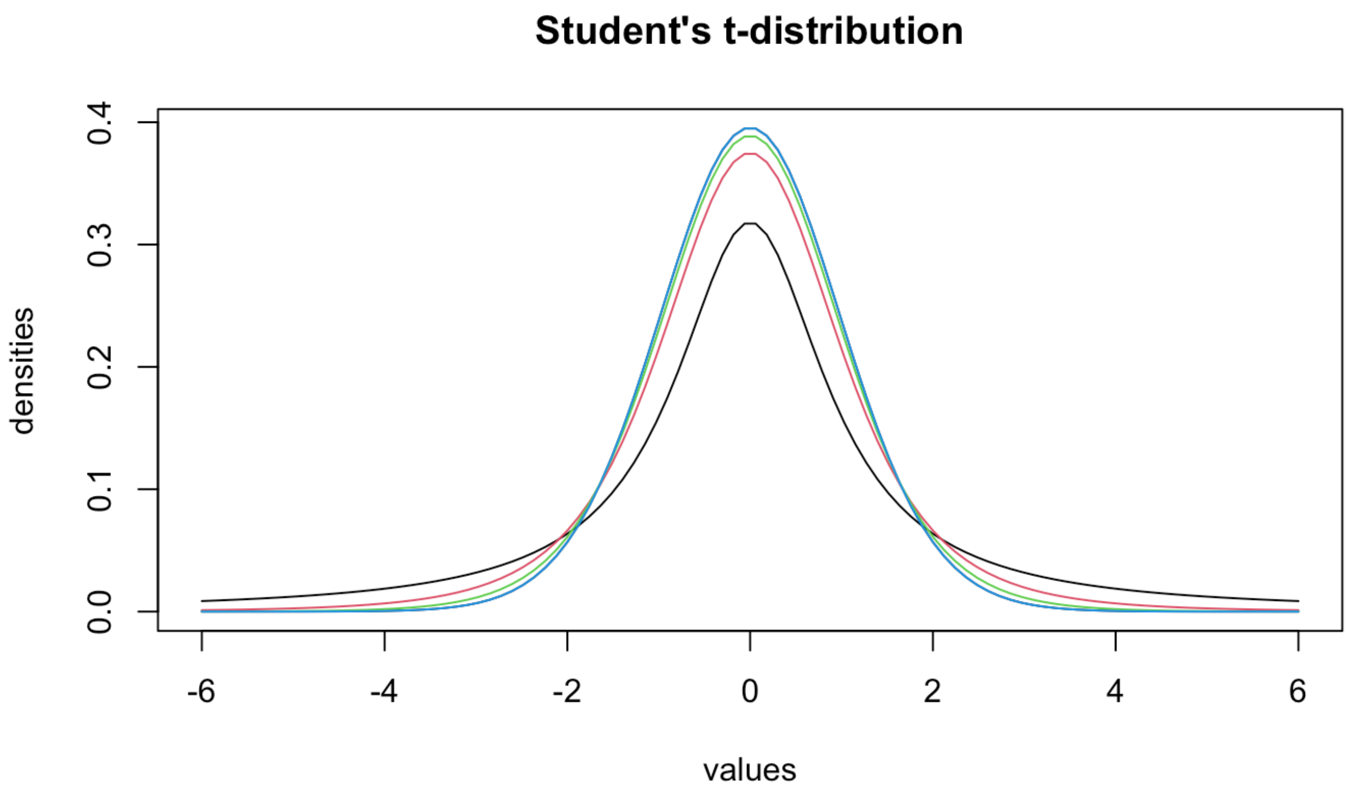
```
x3=rchisq(100,25)
plot(density.default(x3))
```



3. Generate a vector of 100 values between -6 and 6. Use the dt() function in r to find the values of a t-distribution given a random variable x and degrees of freedom 1,4,10,30. Using these values plot the density function for students t-distribution with degrees of freedom 30. Also shows a comparison of probability density functions having different degrees of freedom (1,4,10,30).

Hide

```
x<-seq(-6,6,length.out=100)
df<-c(1,4,10,30)
densities<-lapply(df,function(x1) dt(x,x1))
plot(x,densities[[4]],type = 'l',main="Student's t-distribution",xlab="values",yla
b="densities")
for(i in 1:length(df)){
  lines(x,densities[[i]],col=i)
}
```



4. Write a r-code

i. To find the 95th percentile of the F-distribution with (10, 20) degrees of freedom.

Hide

```
percentile <- qf(0.95, 10, 20)
percentile
```

```
[1] 2.347878
```

ii. To calculate the area under the curve for the interval [0, 1.5] and the interval [1.5, +∞) of a F-curve with v1 = 10 and v2 = 20 (USE pf()).

Hide

```
area1<-pf(1.5,10,20)
area2=1-area1
area1
```

```
[1] 0.7890535
```

Hide

```
area2
```

```
[1] 0.2109465
```

iii. To calculate the quantile for a given area (= probability) under the curve for a F-curve with v1 = 10 and v2 = 20 that corresponds to q = 0.25, 0.5, 0.75 and 0.999. (use the qf())

Hide

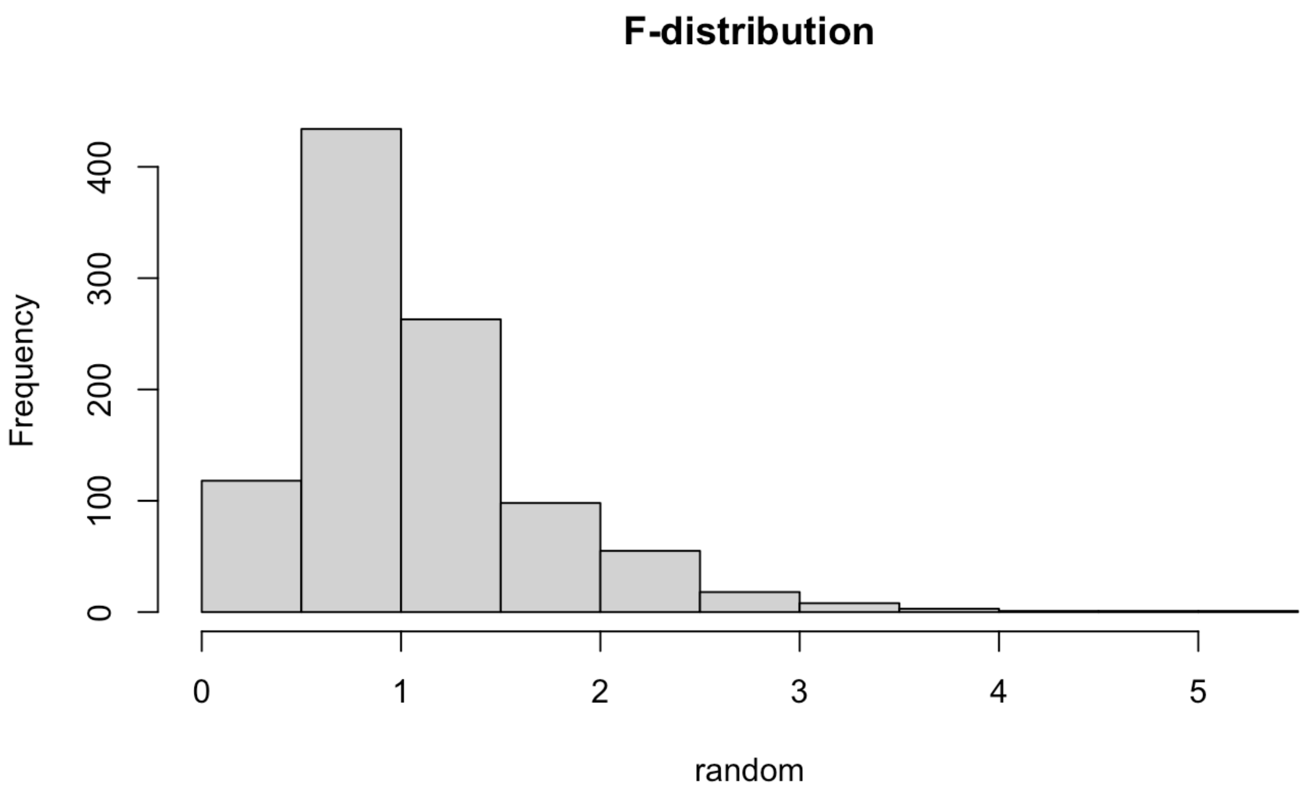
```
quantile<-c(0.25,0.5,0.75,0.999)
quantile_values<-qf(quantile,10,20)
quantile_values
```

```
[1] 0.6563936 0.9662639 1.3994874 5.0752462
```

iv. To generate 1000 random values from the F-distribution with v1 = 10 and v2 = 20 (use rf())and plot a histogram.

Hide

```
random=rf(1000,10,20)
hist(random,main="F-distribution")
```



ASSIGNMNET 8

a. Import the csv data file in R.

Hide

```
data=read.csv("~/Downloads/Clt-data.csv")
```

b. Validate data for correctness by counting number of rows and viewing the top ten rows of the dataset.

Hide

```
nrow(data)
```

```
[1] 9000
```

Hide

```
head(data,n=10)
```

	Wall.Thickness <dbl>
1	12.35487
2	12.61742
3	12.36972
4	13.22335
5	13.15919
6	12.67549
7	12.36131
8	12.44468
9	12.62977
10	12.90381

1-10 of 10 rows

c. Calculate the population mean and plot the observations by making a histogram.

d. Mark the mean computed in last step by using the function abline.

Hide

```
mean_val<-mean(data$Wall.Thickness)
hist(data$Wall.Thickness,main='Histogram')
abline(v=mean_val,col="red")
```

a. Draw sufficient samples of size 10, calculate their means, and plot them in R by making histogram. Do you get a normal distribution.
b. Now repeat the same with sample size 50, 500 and 9000. Can you comment on what you observe. Given data does not follow Normal Distribution

Hide

```
s = sample(data$Wall.Thickness, 10,replace=TRUE)
s
```

```
 [1] 12.45521 12.43709 12.51740 12.39989 12.63452 12.32720 13.12991 13.23847 12.98
830 12.87244
```

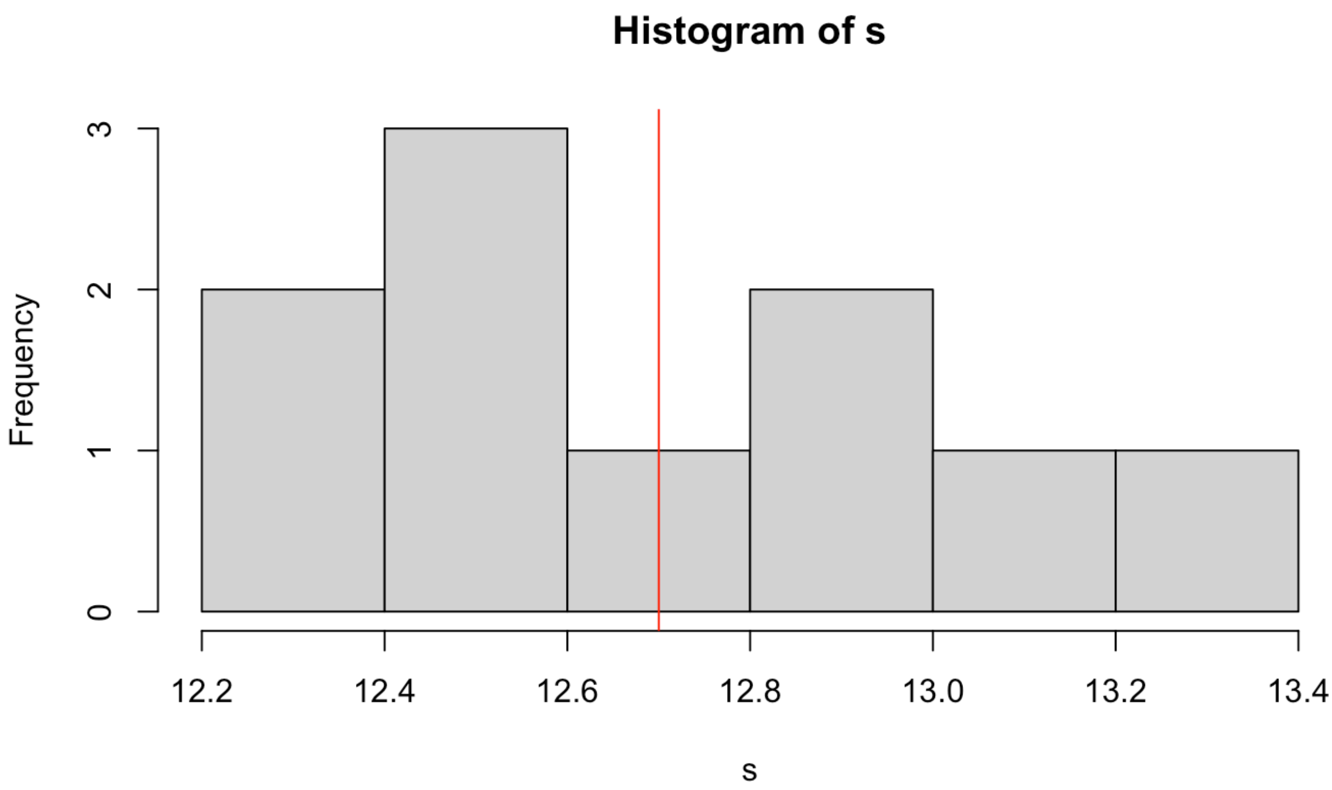
Hide

```
v<-mean(s)
v
```

```
[1] 12.70004
```

Hide

```
hist(s)
abline(v=12.70004, col = 'red')
```



Hide

```
s = sample(data$Wall.Thickness, 50,replace=TRUE)
s
```

```
 [1] 12.92433 12.67889 12.48905 12.59583 13.21084 12.75961 13.16482 13.17123 13.11
439 12.34981 12.90110 13.00224 12.66994
[14] 12.80957 12.38744 13.08477 12.57273 12.51740 12.83626 12.58274 13.06164 13.20
718 12.59583 12.87611 12.32685 12.58172
[27] 13.05835 13.29704 12.61517 13.16948 12.97123 13.03268 13.23867 12.47557 12.96
451 13.01090 13.22797 12.78308 12.57379
[40] 13.02591 12.71575 12.60864 12.58078 13.09709 12.57760 12.78341 13.24267 12.54
106 13.05196 12.69722
```

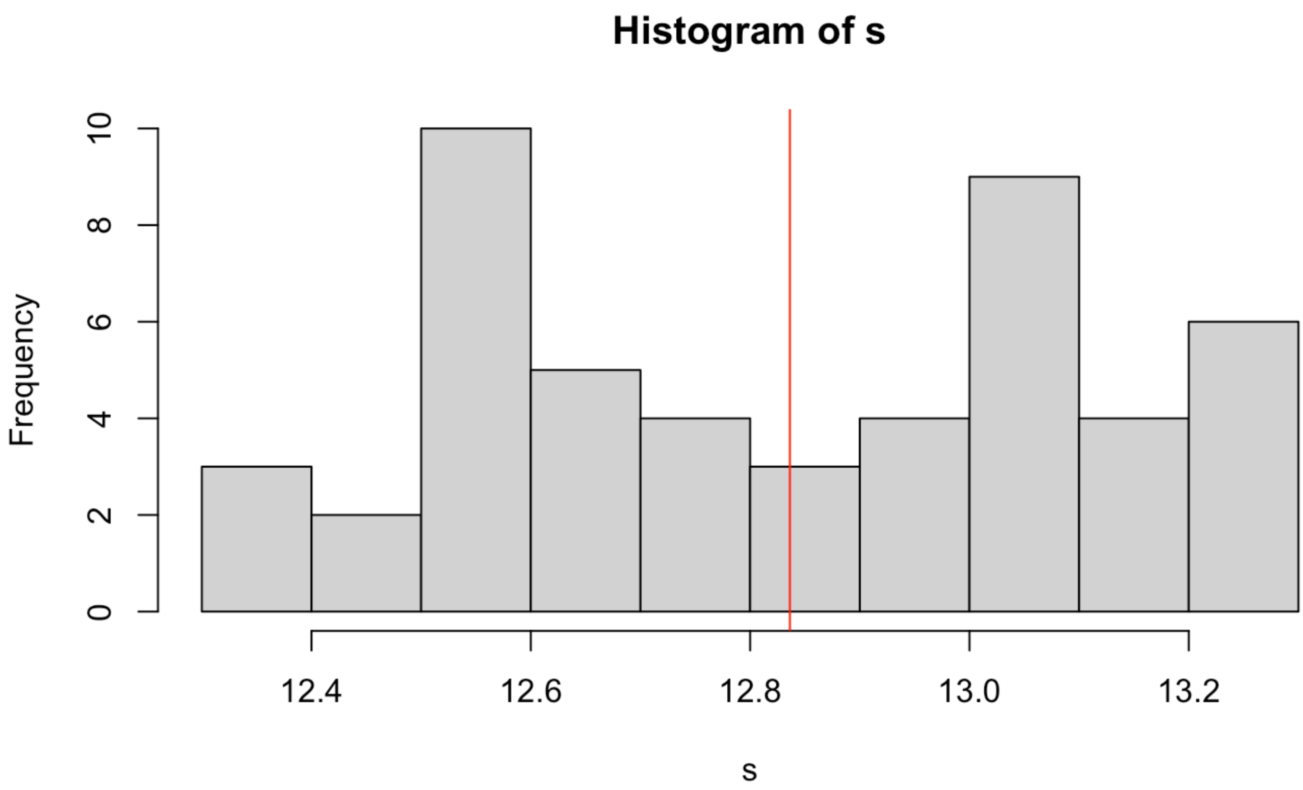
Hide

```
v<-mean(s)
v
```

```
[1] 12.83626
```

Hide

```
hist(s)
abline(v=12.83626, col = 'red')
```



Hide

```
s = sample(data$Wall.Thickness, 500,replace=TRUE)
s
```

```
  [1] 13.28361 13.06770 13.28673 13.04273 12.90760 12.58855 12.58429 12.73988 12.6
7343 13.21546 12.39257 13.28202 12.95058
 [14] 13.12823 13.25547 12.97357 12.34174 12.48609 12.67344 13.18980 12.89069 13.2
0901 13.04139 13.09063 12.63971 12.86099
 [27] 12.93128 12.71756 12.77667 12.74303 12.31746 13.27659 12.35799 12.79626 12.4
9957 12.64070 12.99591 13.17904 12.54414
 [40] 13.08318 12.81208 13.18975 12.64523 13.27718 12.46998 13.11950 13.16482 12.5
4182 13.02905 12.99698 13.06149 12.76122
 [53] 12.54804 12.83380 12.39931 12.49169 12.87987 13.09511 12.66754 13.03605 12.7
4356 13.04182 12.36788 13.21329 13.16948
 [66] 12.59792 12.47634 13.27872 12.90381 13.09027 13.28927 12.92009 13.10456 12.6
8364 12.31448 12.78584 13.22305 13.00032
 [79] 13.07061 12.34514 13.10012 12.80786 12.85767 12.35318 13.20583 13.02624 12.6
1830 12.87317 12.46488 12.70262 12.73115
 [92] 12.93723 13.24375 12.98454 12.82580 12.69073 12.85800 12.88648 12.98080 12.7
4802 13.29832 12.65050 12.68003 13.20230
[105] 12.42560 12.85220 12.91435 12.37091 13.28909 12.53415 12.84175 13.08209 13.1
9435 13.28434 12.33761 12.39199 12.86838
[118] 13.08192 13.07958 12.86237 13.23334 12.32993 12.47547 12.53351 12.80651 12.7
0919 13.19140 13.19894 13.06353 12.39390
[131] 13.20901 12.41846 12.90288 12.89064 13.05191 12.85606 12.55249 12.77722 12.9
```

```
9461 12.94872 12.62885 13.29794 12.71618
[144] 13.11212 13.28927 12.58722 12.75931 13.00931 12.80527 12.88625 12.43539 12.4
6681 13.02819 13.10046 13.09648 13.26427
[157] 12.34540 12.53154 13.22891 12.63479 12.65338 12.60175 13.10067 12.35381 13.2
4887 12.54340 12.71766 12.33731 13.03457
[170] 12.69407 13.12911 12.76637 13.21975 12.60604 12.75223 12.49214 12.84300 12.6
0862 12.46414 13.13379 13.10077 13.06979
[183] 12.39652 12.49107 12.49501 12.71339 12.45874 12.78254 12.78528 12.98464 12.9
4989 13.06843 12.35721 12.84054 13.15398
[196] 12.59865 12.82752 12.68566 12.55407 13.20778 13.19561 12.75263 12.91152 12.3
8308 12.51292 12.31516 13.28924 13.28148
[209] 13.02922 12.52046 12.65989 12.92501 12.41013 12.60323 12.60673 12.93656 12.6
6013 12.93663 12.53445 13.00171 13.20956
[222] 12.88468 13.26926 13.24721 12.50360 13.08252 12.63224 12.87090 12.75665 12.3
2149 13.10967 12.99281 12.74570 12.84456
[235] 13.18059 12.69303 12.84745 12.94302 12.69692 13.14429 13.11731 13.14805 13.1
0744 12.67530 13.08252 12.45932 13.25791
[248] 12.84772 12.68364 12.69503 12.87225 13.05006 12.62290 12.71889 12.85830 12.9
3414 12.65197 12.92596 12.91239 12.63764
[261] 12.31254 12.94919 12.40159 13.28572 12.45461 12.78874 12.36131 12.97709 13.2
1192 12.33432 12.30462 12.95473 12.38440
[274] 12.74003 12.47039 13.20833 12.71766 12.63872 13.23188 12.54782 12.66319 13.0
2619 12.80451 12.62130 12.59824 13.20874
[287] 12.65147 13.15760 12.59432 12.81559 12.88346 13.01880 13.19177 12.52247 12.5
7085 13.00076 12.41713 12.30294 13.23436
[300] 12.86680 13.05832 12.86626 13.20934 12.92962 12.56572 13.22041 12.93273 12.5
1509 12.47762 12.65536 12.95677 12.34691
[313] 12.78962 12.31320 13.27195 13.27168 12.90897 13.15792 13.26955 12.69496 12.9
9010 12.76007 13.11034 12.45763 12.77762
[326] 12.68148 12.48054 12.46998 12.42434 12.83426 12.34877 13.25271 12.74720 13.2
6021 12.71641 12.99813 13.22032 12.98705
[339] 12.98518 12.31073 12.76061 12.75349 12.32001 12.62698 12.55538 12.51850 12.9
2428 12.80940 12.66372 12.97810 13.24461
[352] 12.64914 12.72280 13.06129 13.00680 12.78750 12.63840 13.17218 13.26250 12.5
3780 13.25786 12.43827 12.84831 13.25492
[365] 13.17298 12.69536 12.82046 12.57759 12.55805 12.55199 12.89735 12.33922 12.8
2609 12.65146 13.26828 13.18967 12.94668
[378] 12.90504 12.75533 13.05346 12.66186 12.82329 13.25732 12.54535 12.61084 12.4
4843 12.30084 12.86418 12.62338 12.53957
[391] 12.60584 12.93252 13.29817 13.14937 12.92954 12.97994 12.46914 12.33358 12.4
6287 12.85721 13.18697 13.18422 12.76324
[404] 12.40146 12.32879 12.61660 12.92149 12.77790 12.43188 12.99721 12.49303 12.6
0238 13.25749 13.03795 12.75187 12.76856
[417] 12.87244 12.74450 12.67768 13.23787 12.33432 12.87672 13.05776 13.17045 13.1
8568 12.69404 13.04061 12.77011 13.28180
[430] 12.71020 12.94103 12.56196 13.16087 12.65515 12.85348 12.76019 13.19438 12.5
7759 12.88681 13.15601 12.46695 12.42737
[443] 12.94811 13.29817 12.99160 12.98891 12.77960 12.38975 12.63446 12.53137 12.7
9062 12.59931 12.70451 13.14630 12.78606
[456] 12.82711 12.88388 12.46209 12.61549 12.64460 12.99169 12.63033 12.74060 12.4
```

```
7806 13.00011 13.25732 12.57997 12.82810
[469] 12.67664 12.58705 12.95283 12.30521 12.70876 13.27660 13.19808 13.09772 13.1
0600 13.15313 12.71251 12.64863 12.33206
[482] 12.53214 12.47002 13.01015 12.94589 12.34731 13.13407 12.88359 12.63911 13.0
2665 12.82975 12.63724 12.85664 13.07627
[495] 13.11888 13.00997 12.41674 12.92959 12.33587 12.65682
```

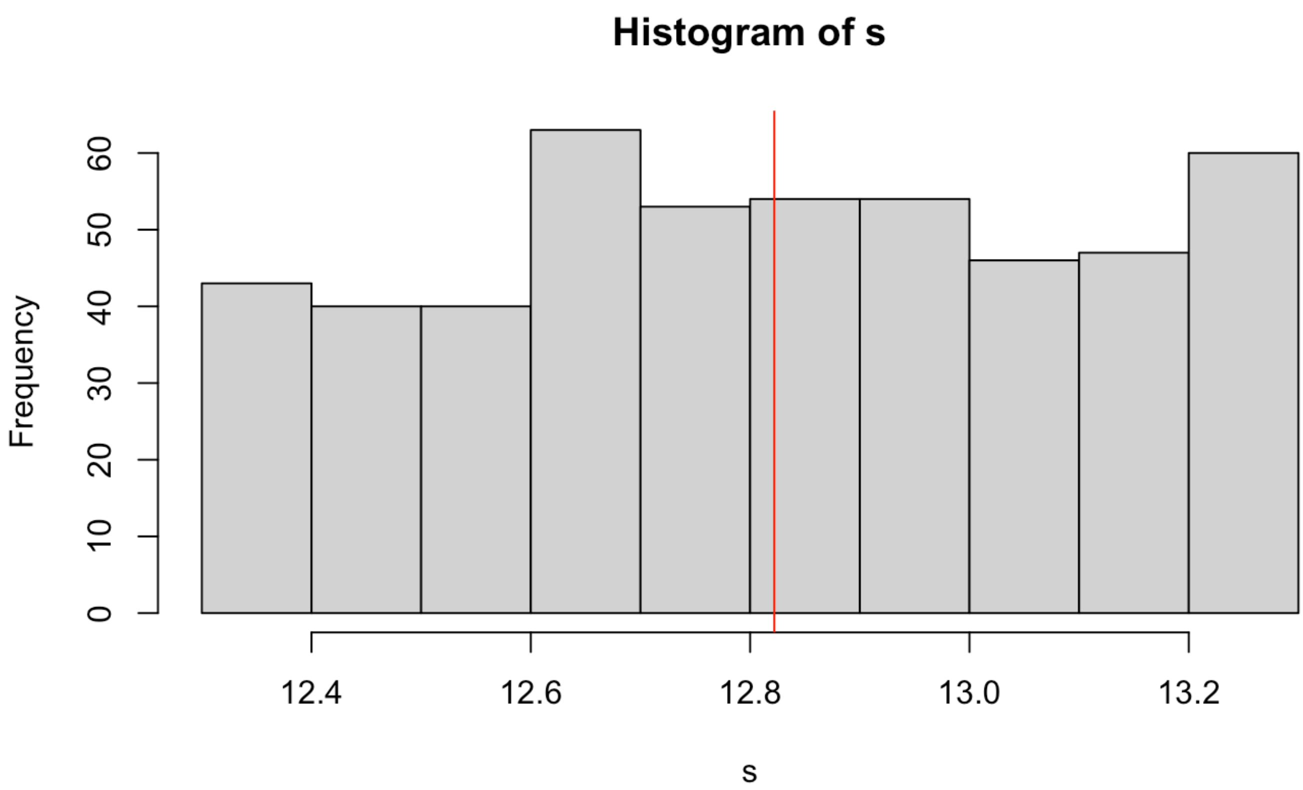
Hide

```
v<-mean(s)
v
```

```
[1] 12.82204
```

Hide

```
hist(s)
abline(v= 12.82204, col = 'red')
```



Hide

```
s = sample(data$Wall.Thickness, 9000)
s
```

```
  [1] 13.27184 12.44119 12.67725 12.46036 13.05652 12.82915 12.79817 12.50266 12.
```

```
47634 12.96733 12.62506 12.55027 13.05835
  [14] 12.30783 12.51946 13.19159 12.91152 12.90989 13.26665 12.39513 12.80061 12.
68838 12.80831 12.98705 12.85793 12.53414
  [27] 12.50567 12.78129 12.46348 13.18022 12.60320 12.94974 12.77649 13.01834 13.
28361 12.75026 12.86838 12.31227 13.19380
  [40] 13.02050 12.76019 13.15493 12.80976 13.16933 12.99524 12.65393 12.80527 12.
99928 12.76406 12.96622 12.97051 12.50726
  [53] 12.37670 12.99998 13.10046 13.19153 13.20230 12.61437 13.25240 12.83580 13.
07798 12.32274 12.46486 12.49457 12.32934
  [66] 13.28839 13.02367 12.86625 12.53990 13.01113 12.96836 12.42109 13.28483 12.
77806 12.43206 12.94995 12.86683 13.17557
  [79] 12.73748 12.58203 12.61248 12.43380 12.94216 12.43861 13.12763 12.94103 13.
14805 13.20547 12.84381 12.48395 12.45727
  [92] 12.70254 13.28368 12.72622 12.62973 13.00352 12.72280 12.93252 12.54526 12.
61654 13.00289 13.15636 12.65199 13.11729
 [105] 13.23079 12.66427 12.73488 13.28218 13.25161 12.37427 12.32598 12.76420 13.
01277 12.57028 12.34898 13.16745 12.77786
 [118] 13.12595 13.24502 13.13494 12.32462 12.69331 13.05621 12.41863 12.49448 12.
67928 12.62568 12.83458 12.94567 12.70356
 [131] 12.87318 12.71002 12.98806 13.26106 12.65940 12.46275 12.52428 12.92198 12.
31811 12.36118 13.07958 12.91583 12.36323
 [144] 13.13294 13.16283 12.70306 12.78963 12.78140 12.53060 13.28939 13.28180 13.
04268 12.91890 13.01681 13.29908 12.80049
 [157] 12.88893 12.98867 13.00466 12.86686 12.40390 12.64863 12.69965 12.95585 13.
29489 12.52816 13.16617 12.90113 12.51261
 [170] 12.57266 12.34804 13.23177 12.45117 12.97621 12.30370 13.02878 12.51826 12.
70876 13.13703 12.34603 12.33781 13.15807
 [183] 12.46914 12.67889 12.56355 13.27435 12.67107 12.65832 12.57892 13.15426 13.
05524 12.85123 12.33033 12.43665 12.64107
 [196] 13.11249 13.29511 12.56135 12.70668 13.10170 12.94620 12.63003 12.67097 12.
92828 12.95519 12.43595 13.23342 12.45489
 [209] 13.00476 12.97147 13.21180 12.47551 13.04870 12.33186 13.15188 12.71975 13.
15575 13.20687 12.95004 13.26782 12.33862
 [222] 13.21412 12.67158 12.79498 12.99024 12.58949 13.17748 13.21100 13.06846 12.
59243 12.37091 12.67592 12.92424 13.17283
 [235] 12.58220 12.96451 12.37653 12.87399 12.48655 12.94598 12.34058 13.14773 12.
82132 12.33731 12.59300 12.92258 13.03012
 [248] 12.64809 13.03915 12.43827 13.13970 12.51239 12.95589 12.90504 12.36140 12.
83839 12.63176 13.04867 12.77694 12.90384
 [261] 12.40804 12.31584 12.83943 12.50867 12.89270 13.26613 13.29524 12.31320 12.
94631 12.68411 12.78911 12.97958 12.63485
 [274] 12.38741 12.96444 12.30888 12.62864 12.95891 12.68148 13.15715 12.78349 12.
36222 12.39688 12.86626 12.59159 12.87388
 [287] 12.79070 12.95488 12.85664 13.01775 13.03904 12.78825 12.64635 13.12267 12.
87698 12.41811 12.56196 12.45763 12.83977
 [300] 12.77233 13.15197 12.52815 13.12744 12.41905 13.15919 12.98753 12.36952 12.
76058 12.32868 12.37869 12.35170 12.47373
 [313] 13.26794 12.46353 12.30521 12.87379 12.53852 13.17272 13.20922 12.72358 12.
53522 12.80957 13.21885 12.55609 12.90252
 [326] 12.56744 12.58093 13.01642 12.51063 12.33587 13.26805 12.74720 12.72283 13.
```

```
15982 12.64210 12.42705 13.29817 12.82342
 [339] 13.21977 12.68709 12.54394 13.08192 13.07992 12.75931 13.03169 12.90678 12.
38792 13.05556 12.66677 12.89741 12.88569
 [352] 12.42586 12.86775 12.45484 12.84629 12.53056 13.07673 13.23795 13.28182 13.
12987 12.50403 12.58461 12.33888 12.79805
 [365] 12.88531 13.29649 12.34508 12.32209 12.64924 12.73428 12.78340 12.89151 12.
69432 12.42544 13.10678 12.62368 12.61138
 [378] 12.93340 12.50806 12.49378 13.07469 12.62799 12.99956 12.35014 13.00612 12.
40701 13.02925 12.85677 12.97994 12.55497
 [391] 13.04947 13.18345 12.58478 13.14630 12.95088 13.19873 12.49752 12.59211 12.
52339 13.29613 12.68072 12.41285 12.51371
 [404] 12.88111 12.55864 12.80874 12.38623 12.91398 12.61989 12.76521 12.89560 13.
07570 12.62563 12.84780 12.96996 13.19523
 [417] 12.36359 12.60507 12.48389 12.91816 13.23068 12.47369 12.40899 12.86299 13.
03618 13.02481 13.22513 12.93175 12.73431
 [430] 12.48725 12.56392 12.75432 13.10355 12.76862 12.89153 13.12830 12.53154 12.
76778 13.29794 12.92433 12.98139 13.08334
 [443] 12.71200 12.96568 12.40505 13.28168 12.41663 12.45152 12.93153 12.51850 12.
50350 12.93368 12.54078 13.21394 13.16754
 [456] 13.29524 12.30354 13.21115 12.31254 12.72390 12.37952 13.00072 12.99220 13.
05318 12.40232 13.05493 12.48526 12.49715
 [469] 12.54422 13.03691 12.88380 12.74570 12.95506 12.42270 13.25545 12.48218 12.
46533 12.90740 12.59451 13.21859 13.21994
 [482] 12.92985 12.75283 12.79652 13.14201 12.67734 13.29596 13.05983 12.46870 12.
77956 13.03413 13.06308 13.11994 13.17123
 [495] 13.15962 12.39107 13.05828 12.30772 12.93349 12.64796 13.24563 12.51464 13.
12642 12.68511 12.39126 13.13793 12.38012
 [508] 12.36514 13.25229 12.95073 13.00371 12.75059 12.86609 12.47375 12.61912 13.
12256 12.88724 12.57914 12.48383 12.71289
 [521] 13.03571 12.56448 12.53673 12.51940 12.37939 12.36112 12.65214 12.42979 13.
14486 12.72999 12.30095 13.26256 12.88223
 [534] 12.53059 12.94128 12.93918 13.23875 12.49965 13.29927 12.68003 13.07897 12.
63945 13.22305 12.74661 12.81825 12.64330
 [547] 12.69424 12.40682 13.18967 12.46014 12.54048 13.10487 12.72096 12.46970 13.
16962 12.95522 12.63365 12.56272 13.03109
 [560] 13.13427 12.36132 12.63565 12.54130 12.54236 13.25831 13.24426 12.88163 12.
98869 12.87075 12.78223 13.11828 12.49870
 [573] 12.89318 13.16410 12.78522 13.09648 12.87710 12.95426 12.62296 13.29066 12.
40368 12.82099 12.69848 12.56349 12.49989
 [586] 13.27659 12.95301 12.39870 12.30418 13.06125 13.10750 13.08768 12.80490 12.
63455 12.59745 12.77979 13.26520 13.29329
 [599] 12.35232 12.72733 12.43776 13.15755 13.12365 12.76507 12.50261 12.48830 12.
39995 12.35220 12.59792 13.18887 12.99403
 [612] 12.72715 12.95936 12.48077 13.20718 12.33418 12.99769 12.99125 13.28715 12.
61578 13.29603 12.81419 12.96112 12.78533
 [625] 13.21809 12.42767 12.71972 12.93414 13.27047 12.56656 12.66451 12.83298 12.
65230 12.54535 13.20994 12.89016 12.48194
 [638] 12.99397 13.08947 12.90816 12.49083 12.46132 12.69722 12.90194 12.76353 12.
72281 13.13591 12.91434 12.37702 12.46414
 [651] 12.96263 12.49831 12.71830 12.39509 12.35153 13.13827 13.25165 12.63482 12.
```

```
79479 12.47094 13.05579 12.96313 12.75947
 [664] 12.94302 13.10639 13.05517 12.83661 13.29611 13.22686 12.84093 13.27597 13.
19005 12.83426 13.13402 12.75534 12.67554
 [677] 12.81073 12.42749 13.20150 12.41863 12.57503 12.78962 12.90760 12.64523 12.
37864 12.69802 13.00242 12.70394 12.94188
 [690] 12.51118 12.36935 13.12318 13.11466 13.10604 12.74802 12.42243 12.64926 12.
99133 12.59011 13.16818 13.05737 12.91822
 [703] 12.95612 12.30218 12.30745 12.88816 12.66293 12.62950 13.21158 12.98813 12.
92665 12.63166 13.09473 12.40223 13.26487
 [716] 13.27570 13.03794 13.20051 12.33361 12.64504 12.79099 12.46265 12.45381 13.
14695 12.39023 12.37236 12.66655 13.26250
 [729] 12.99348 12.67814 13.13031 13.11872 12.85301 13.04182 12.98891 12.90279 12.
96606 12.71938 13.08448 12.94851 13.03485
 [742] 12.33105 12.94598 12.30701 12.81370 13.21823 13.28101 13.28394 12.54986 13.
06693 12.30363 13.27409 12.65783 13.27936
 [755] 12.88402 13.00805 12.85131 12.55450 12.35541 13.16008 12.83079 12.94743 12.
77619 13.19564 12.59601 12.70855 12.55653
 [768] 12.91508 12.49606 12.90067 12.99111 12.83596 12.50415 13.15541 13.04547 12.
38978 12.78606 12.44199 12.68325 13.18850
 [781] 12.48893 12.68542 12.85767 13.16034 12.98283 12.97209 13.17178 12.82807 12.
86817 13.02247 13.08815 12.50395 12.96186
 [794] 13.06987 12.45335 13.26172 13.26899 13.10600 13.06353 12.48047 12.42851 12.
74936 12.57085 13.01846 13.12834 12.82475
 [807] 12.66819 13.26988 12.58158 12.46691 12.86896 12.83123 12.35417 12.30230 13.
26465 12.89808 12.38023 13.03476 12.73968
 [820] 12.91126 12.88754 12.59369 12.74870 12.33688 12.79438 12.39170 12.36372 13.
03605 12.76112 12.85354 13.20400 13.28775
 [833] 12.65190 12.52990 12.34072 12.65475 12.42515 12.53630 12.91337 12.33236 12.
87039 13.23374 12.65020 13.23574 12.38690
 [846] 13.27357 12.51239 13.21444 12.69764 12.80201 12.82115 12.37086 12.44411 12.
47222 12.79668 13.04915 12.33163 13.04786
 [859] 12.47756 12.43097 12.75794 13.11546 12.49470 12.85908 12.61080 12.60264 12.
42267 12.99324 12.78789 12.60202 12.92647
 [872] 13.24481 12.64176 12.99892 12.84264 12.30084 13.19640 13.00636 13.19749 13.
22022 12.86783 13.19295 12.54355 13.12905
 [885] 12.47801 12.60580 13.29309 12.78539 13.12071 12.83556 12.68418 12.68746 13.
18415 12.84456 12.86071 12.94668 13.26449
 [898] 12.60725 13.23909 13.12537 12.56941 13.18396 13.18245 13.02647 13.08413 12.
71508 12.35320 12.73057 12.56256 13.17565
 [911] 12.53782 12.96019 12.46186 13.01222 12.90888 12.70789 13.17803 12.32712 12.
71901 12.83234 12.69881 12.74365 12.84456
 [924] 13.16581 13.01607 12.84158 12.51921 12.71641 12.68451 13.18889 13.07638 12.
80786 12.85710 12.85976 12.89097 12.92583
 [937] 12.67431 12.53810 12.56536 13.23788 12.90756 12.43551 12.68966 12.92593 12.
32149 12.50138 13.04750 12.30930 12.63971
 [950] 12.99296 13.17959 13.22032 12.40938 12.55164 13.27168 13.10022 12.60599 13.
24375 12.48709 12.60469 12.96546 12.42487
 [963] 13.02963 12.32691 12.53351 12.85895 13.19384 13.13835 13.18523 12.92363 12.
34316 13.29816 12.60557 12.32710 12.37398
 [976] 13.17611 12.85891 12.61484 13.06744 13.06050 12.87455 12.47140 13.05040 12.
```

```
89172 12.53981 13.08501 13.15018 12.96163
 [989] 13.13994 12.31743 12.88200 13.27537 13.10519 12.68537 13.27872 12.73482 12.
65292 12.30463 13.14225 12.92362
 [ reached getOption("max.print") -- omitted 8000 entries ]
```

Hide

```
mean(s)
```

```
[1] 12.80205
```

Hide

```
hist(s)
abline(v=12.80205,col='red')
```