Monte Carlo Simultions and Bootstrap

Yen-Chi Chen

5/20/2017

Monte Carlo Simulations

Assume in a dataset, we observe n values, denoted as $X_1, \cdots, X_n$. For simplicity, we assume that these observations are an IID (indenpendently, identically distributed) random sample from an unknown distribution function $F(x)$.

Because these n values are random, any statistic derived from them will also be random. For instance, let S_{med} be the median of these n values (sample median). The value of the median will be determined by these n observations. Because these observations are a random sample from the distribution F, the median will also be random, and it will have its own distribution (also depends on F but through a more complicated way)!

Assume the distirbution is something we know, say a standard Normal distribution $N(0, 1)$, and the sample size $n = 100$. What will the distribution of the sample median be?

We can find this out using the Monte Carlo Simulation approach. First we draw a random sample using R and compute the sample median:

```
X = rnorm(100)
X_med = median(X)
X_med
```

```
## [1] 0.01708379
```

This gives us one realization of the median. However, every time we apply the same program, we obtain a different value of the sample median because of a different sets of points we are having.

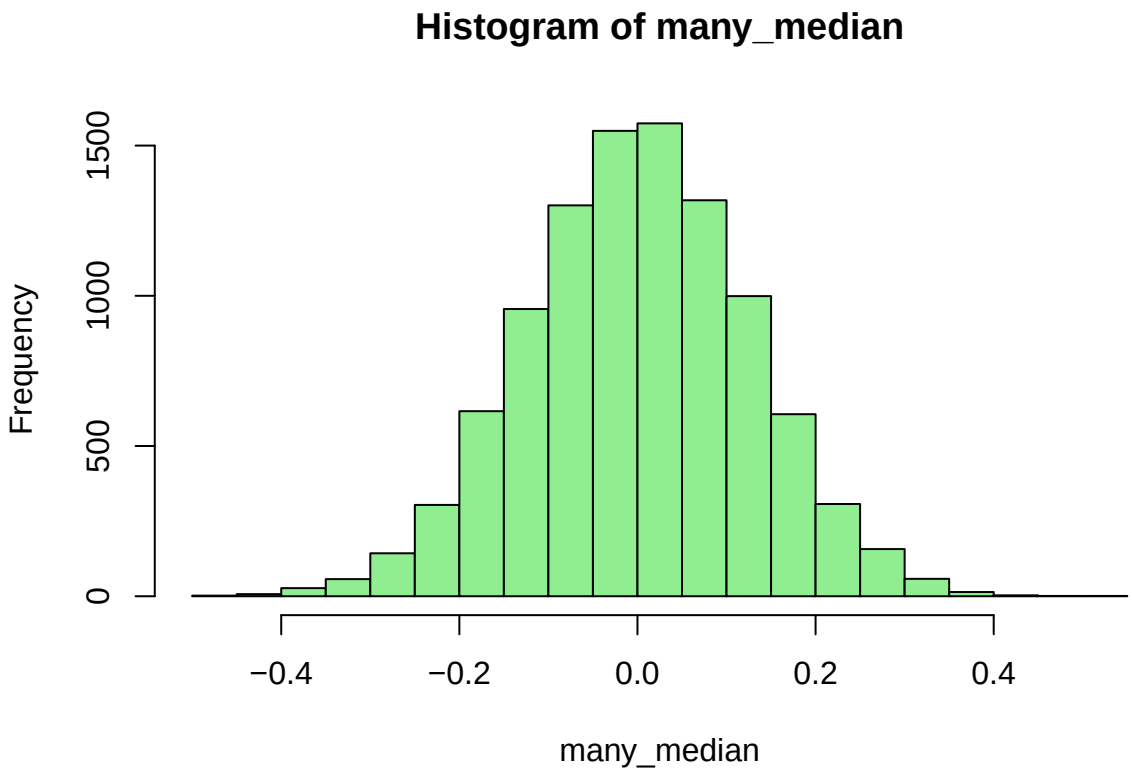
One way to investigate the distribution of sample median is to repeat the above procedure many times and keep track of the sample median of each sample we generate.

```
N = 10000
  # number of repitions
many_median = rep(NA, N)
for(i in 1:N){
  X = rnorm(100)
  X_med = median(X)
  many_median[i] = X_med
    # save the median in each repitition
}
head(many_median)
```

```
## [1]  0.03550759  0.13135906 -0.14435323 -0.12738564  0.03880981 -0.01144368
```

The object `many_median` contains the 10000 realizations of the sample median. To see its distribution, we can simply use the histogram:

```
hist(many_median, col="lightgreen")
```

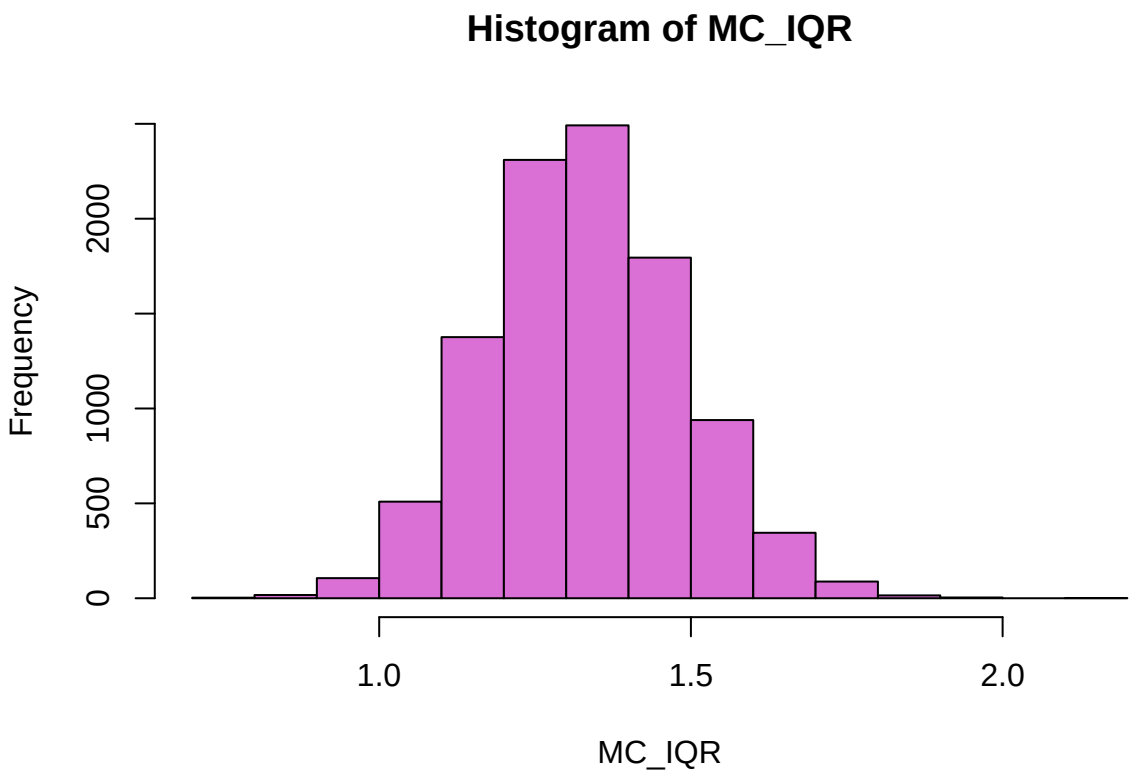


Surprisingly, the sample median also follows roughly a Normal distribution (actually you can prove this fact but it requires some advanced probability techniques).

The above method we are using is call *Monte Carlo Simulation*–to investigate the distribution of something, instead of using mathematics to derive it, we use computer experiments to obtain an approximated answer to the problem. The number of repition N is sometimes called the size of Monte Carlo Simulation and it controls the error of the simulation (this type of error is called Monte Carlo errors). Similar to the sample size, a large repition N we are using, a smaller Monte Carlo error we have.

Not only the median, we can also use the Monte Carlo Simluation to investigate other statistic, for instance, the interquartile range (IQR).

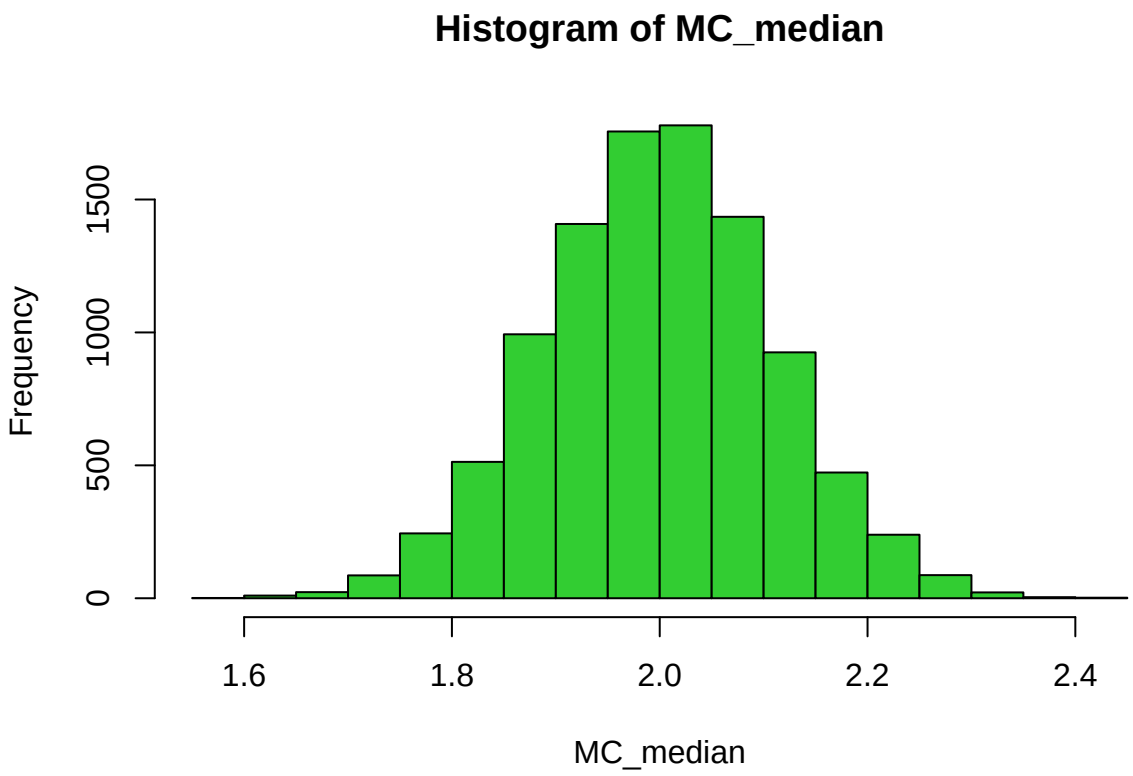
```
N = 10000
  # number of repitions
MC_IQR = rep(NA, N)
for(i in 1:N){
  X = rnorm(100)
  X_IQR = IQR(X)
  MC_IQR[i] = X_IQR
    # save the median in each repitition
}
hist(MC_IQR, col="orchid")
```



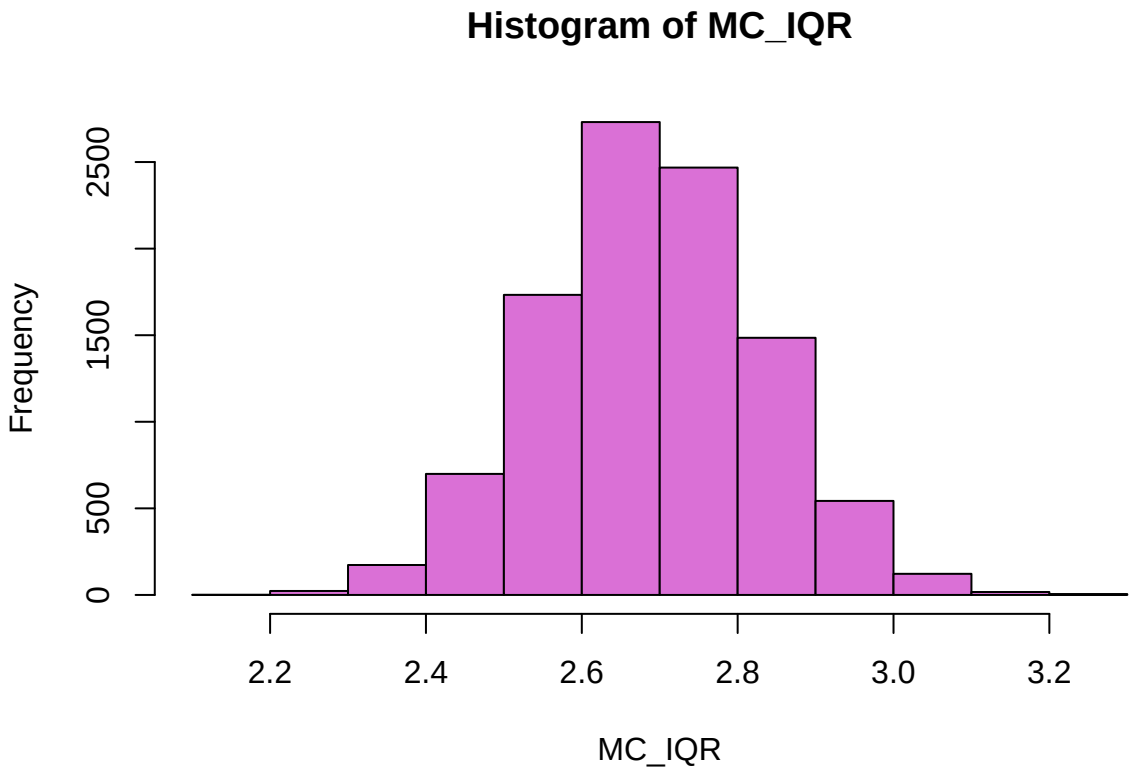
The distribution of IQR also follows a Normal distribution! Actually, many statistics you are familiar with follow a Normal distribution. But of course there will be some exceptions (for instance, the maximum value and the minimum value will not follow a Normal distribution in general).

If now we want to investigate the distribution of sample median/IQR of a different size of sample and from a different distribution, we can simply change the sample size 100 to another value and the function `rnorm()` to others. Here is an example of the sample median and IQR from a size 500 random sample from a distribution $N(2, 4)$:

```
N = 10000
  # number of repitions
MC_median = rep(NA, N)
MC_IQR = rep(NA, N)
for(i in 1:N){
  X = rnorm(500, mean=2,sd=2)
    # N(2,4): mean=2, SD=2
  X_median = median(X)
  MC_median[i] = X_median
  X_IQR = IQR(X)
  MC_IQR[i] = X_IQR
    # save the median in each repitition
}
hist(MC_median, col="limegreen")
```




```
hist(MC_IQR, col="orchid")
```



Because these statistics have a distribution, we can also measure their spreads by the "variance". Note that here we are looking for the "variance" of "sample median" and the "variance" of "sample IQR", not the variance of the original sample.

```
var(MC_median)
```

```
## [1] 0.01217119
```

```
  # this gives you the variance of the sample median
var(MC_IQR)
```

```
## [1] 0.01994174
```

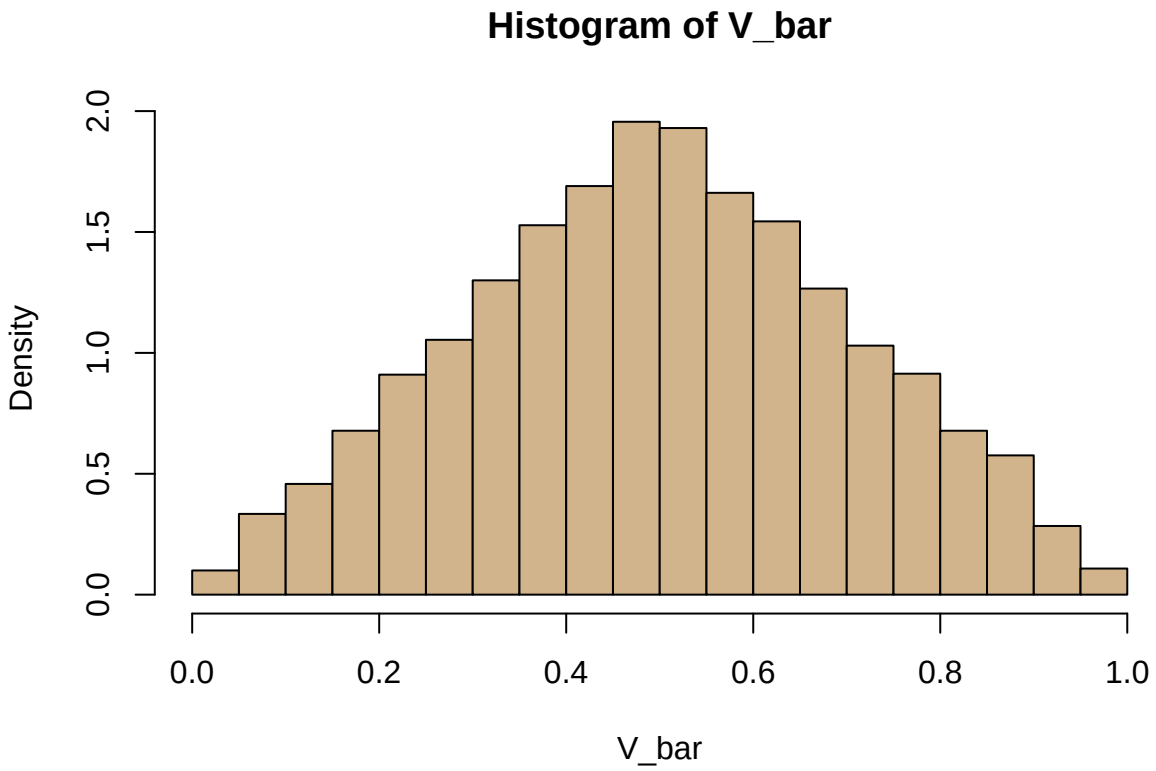
```
  # this gives you the variance of the sample IQR
```

Analyzing the density of a random variable

Let V_1 and V_2 be two IID random variables that are from a uniform distribution over $[0, 1]$. Their average $\bar{V}_2 = \frac{V_1+V_2}{2}$ is also a random variable. Then what is the distribution of $\bar{V}_2$?

Here is a simple application of Monte Carlo Simulation to figure it out.

```
N = 10000
V1 = runif(N)
V2 = runif(N)
V_bar = (V1+V2)/2
hist(V_bar, probability = T, col="tan")
```

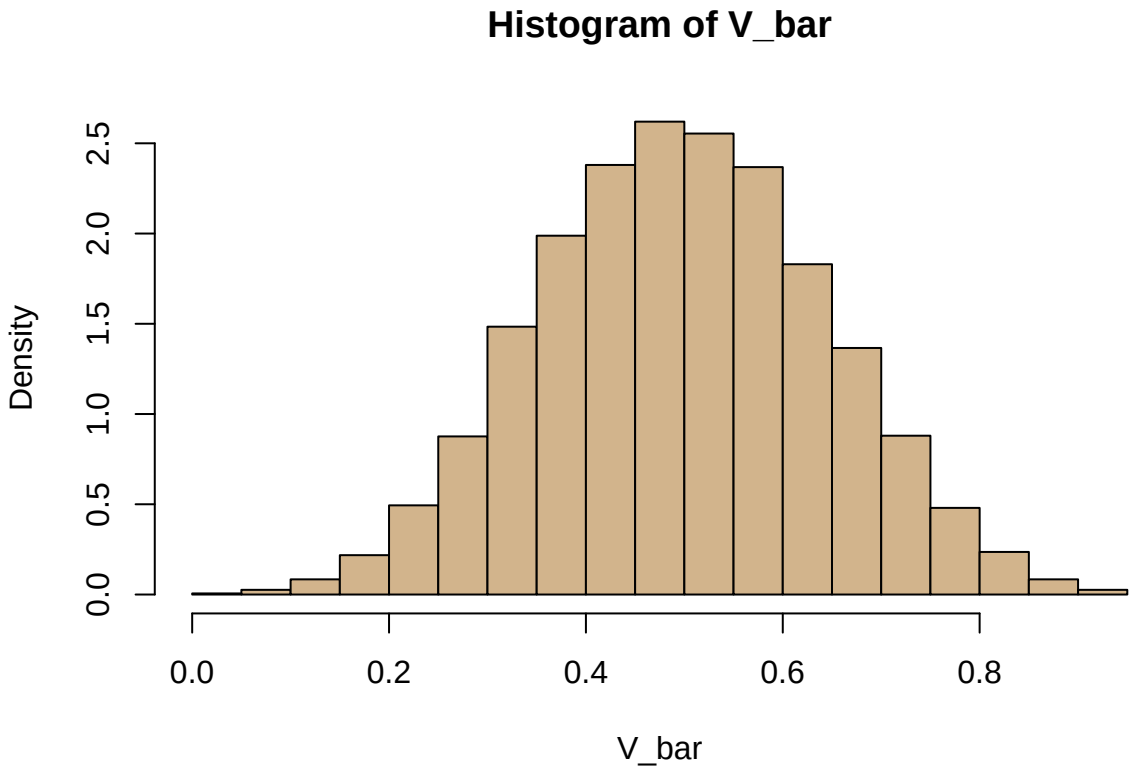


It looks like a triangle distribution! And actually, you can prove that its density is indeed a triangle!

What would happend if now we are taking average of 4 uniform random variables?

```
N = 10000
V1 = runif(N)
V2 = runif(N)
V3 = runif(N)
```

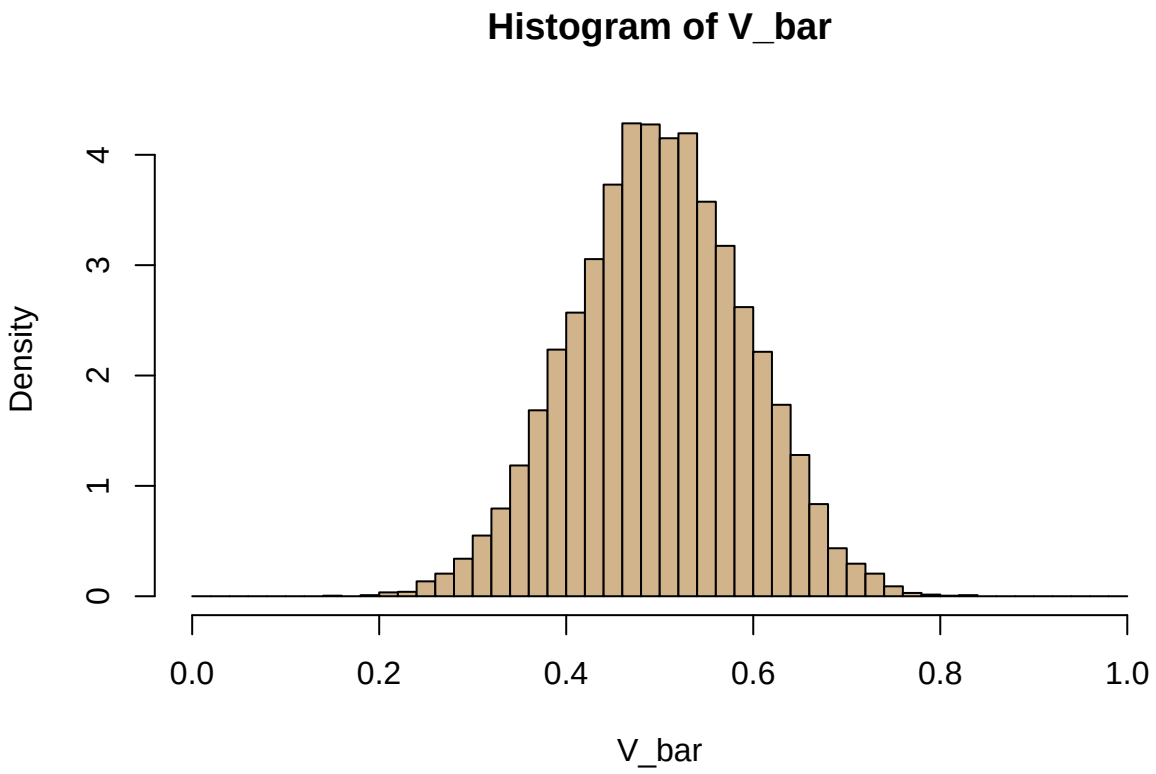
```
V4 = runif(N)
V_bar = (V1+V2+V3+V4)/4
hist(V_bar, probability = T, col="tan")
```



Hmm it looks like something similar to a *normal distribution*! Actually, this is what the Central Limit Theorem refers to–the distribution of the *average* behaves like a Normal distribution when we are averaging many IID random numbers.

Here is the case when we average over 10 random variables:

```
N = 10000
V_matrix = matrix(runif(N*10),ncol=10)
V_bar = rowSums(V_matrix)/10
hist(V_bar, probability = T, col="tan", breaks=seq(from=0,to=1,by=0.02))
```



This distirbution does look very similar to a Normal.

Estimating a probability

The Monte Carlo Simulation can also be used to estimate a "probability". Here is one example. Let X be a random variable from $N(2,4)$ and Y is a random variable (independent from X) from a uniform distribution over [1,5]. What is the probability that $X < Y$? Namely, what is the value $P(X < Y)$? Mathematically, we know that there is an answer. But it may not be easily computed. Here is how we can use the Monte Carlo Simulation to compute it:

```
N = 10000
X = rnorm(N,2,2)
Y = runif(N, min=1,max=5)
cf_value = X<Y
head(cf_value)
```

```
## [1]  TRUE  TRUE FALSE  TRUE  TRUE FALSE
```

```
  # this cf_value is a vector consistss of the comparison
sum(cf_value)/N
```

```
## [1] 0.6663
```

```
  # this is an estimated value of that probability
```

In the Monte Carlo simulation, we keep generating realizations of X and Y and try to compare if $X < Y$. Now let X_i, Y_i be a pair of value that are generated. What is the outcome if we type $X_i < Y_i$ in R? It will be a logical value such that if $X_i < Y_i$, the value is `T` otherwise the value is `F`. If we treat the logical value as numerics, then the outcome take value 1 if $X_i < Y_i$ otherwise it takes value 0. Let the outcome be D_i. Because X_i, Y_i are random, D_i is also a random number. But D_i only takes value 0 and 1–this implies that D_i is a Bernoulli random variable! For a Bernoulli random variable, there is a probability parameter

determining the chance of generating value 1. What is that probability for D_i? Recall that $D_i = 1$ if and only if $X_i < Y_i$. Thus $P(D_i = 1) = P(X_i < Y_i)$ is the probability we want to estimate!

As a result, the logical vector `cf_value` is actually just n realizations of a Bernoulli distribution with the probability parameter $P(X < Y)$! So the average value of the vector `cf_value` is just the proportion of value `T` (or the numeric value 1) being generated (sometimes it is called the *proportion* statistic). Thus, by the Law of Large Number, this average converges to $P(X < Y)$ when the size of Monte Carlo Simulation $N \to \infty$.

Exercise - 1

1. Assume we want to investigate the distribution of sample standard deviation (SD) of a size $n = 200$ random sample from a $N(2, 1)$. Use Monte Carlo Simulation with $N = 10000$ to show the histogram of the distribution of sample SD.
2. What is the variance of the distribution of sample SD?
3. Let X be a random variable that is uniformly distribution over $[-1, 1]$ and Y is a random variable from $N(1, 1)$ that is indepdent of X. What is the probability that $X + Y > 1.5$?
4. Let U and V be two independent random variables that both are from a uniform distribution over $[0, 1]$. Use the Monte Carlo Simulation with size $N = 10000$ to show the distribution of $(U + V)/2$ (you can show the distirbution by a histogram).

Bootstrap

The bootstrap is a Monte Carlo Simulation approach based on the *data* we haveto *estimate* the uncertainty of a statistic or an estimator. A powerful feature of the bootstrap is: we do not need to know the true distribution.

Take the median for an example. How do we estimate its distribution/uncertainty (say variance)? We generate the *same size* of data from the *same distribution* many times and then use the histogram/variance of these new realizations.

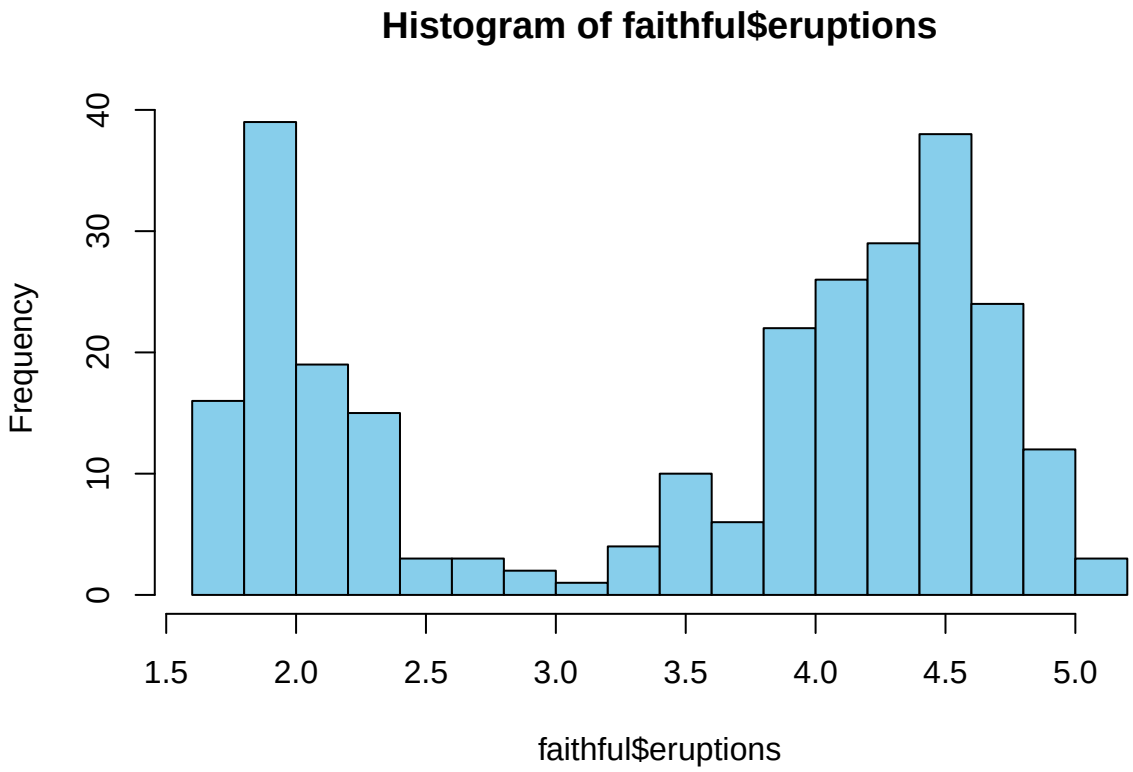
The bootstrap uses a similar idea but now we *treat the original data as the population and sample with replacement from it*. A key element here is *sample with replacement*. This is to mimic the process of generating an IID sample–recall that when we sample with replacement, every points are IID.

Here is one example of applying the bootstrap to approximate the uncertainty of sample median of the variable `eruptions` in the `faithful` data:

```
head(faithful)
```

```
##   eruptions waiting
## 1     3.600      79
## 2     1.800      54
## 3     3.333      74
## 4     2.283      62
## 5     4.533      85
## 6     2.883      55
```

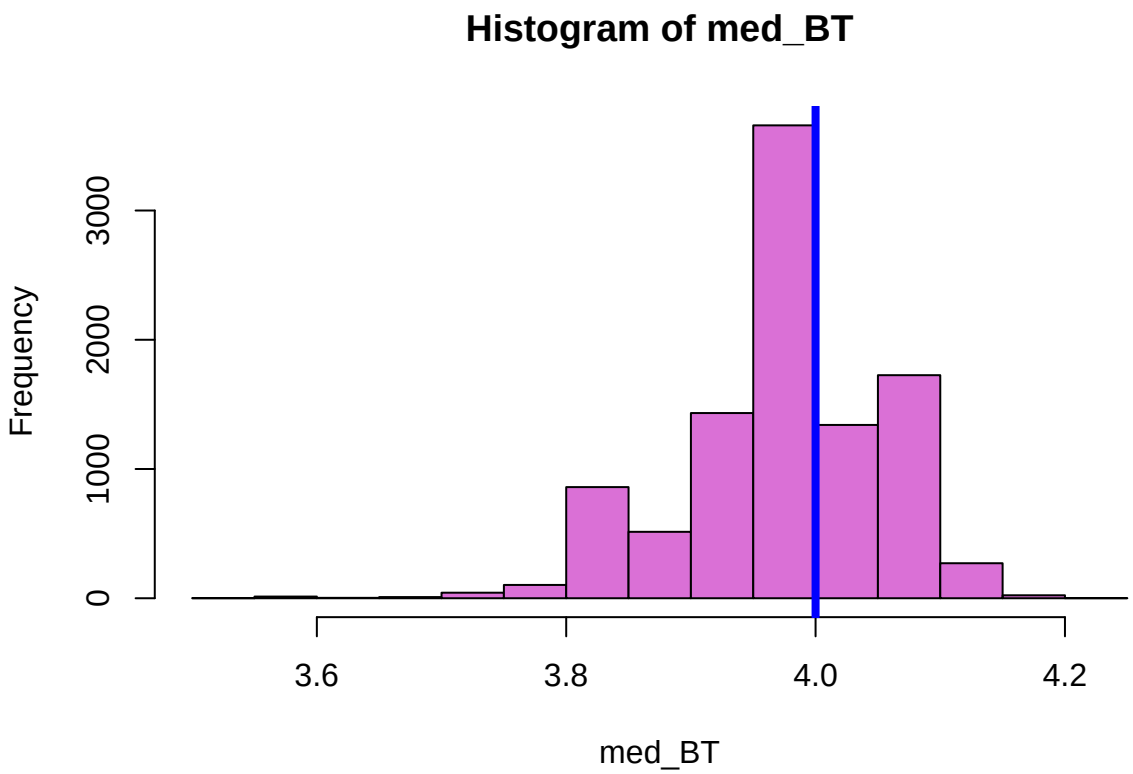
```
hist(faithful$eruptions, breaks=20, col="skyblue")
```




```
median(faithful$eruptions)
```

```
## [1] 4
```

```
  # this gives us the sample median of 'eruptions'

B = 10000
  # B plays the same role as the size of Monte Carlo Simulation
med_BT = rep(NA, B)
n = nrow(faithful)
for(i in 1:B){
  w = sample(n,n, replace=T)
    # this generates the indices we are selecting during the sample with replacement
  X_BT = faithful$eruptions[w]
  med_BT[i] = median(X_BT)
}
hist(med_BT, col="orchid")
abline(v=median(faithful$eruptions), col="blue",lwd=4)
```




```
  # an approximated distribution of sample median
var(med_BT)
```

```
## [1] 0.006437381
```

```
  # an approximated variance of sample median
```

The command `w = sample(n,n, replace=T)` generates indices that are IID from a uniform distribution over $1, 2, 3, ..., n$. Thus, the command `faithful$eruptions[w]` selects those observations being selected and it is then a sample with replacement from the original data `faithful$eruptions`.

Moreover, because we know that the sample median (under suitable conditions) follows roughly a Normal distribution. We can then construct a 90% confidence interval using

```
median(faithful$eruptions) + qnorm(0.95)*sd(med_BT)
```

```
## [1] 4.131972
```

```
median(faithful$eruptions) - qnorm(0.95)*sd(med_BT)
```

```
## [1] 3.868028
```

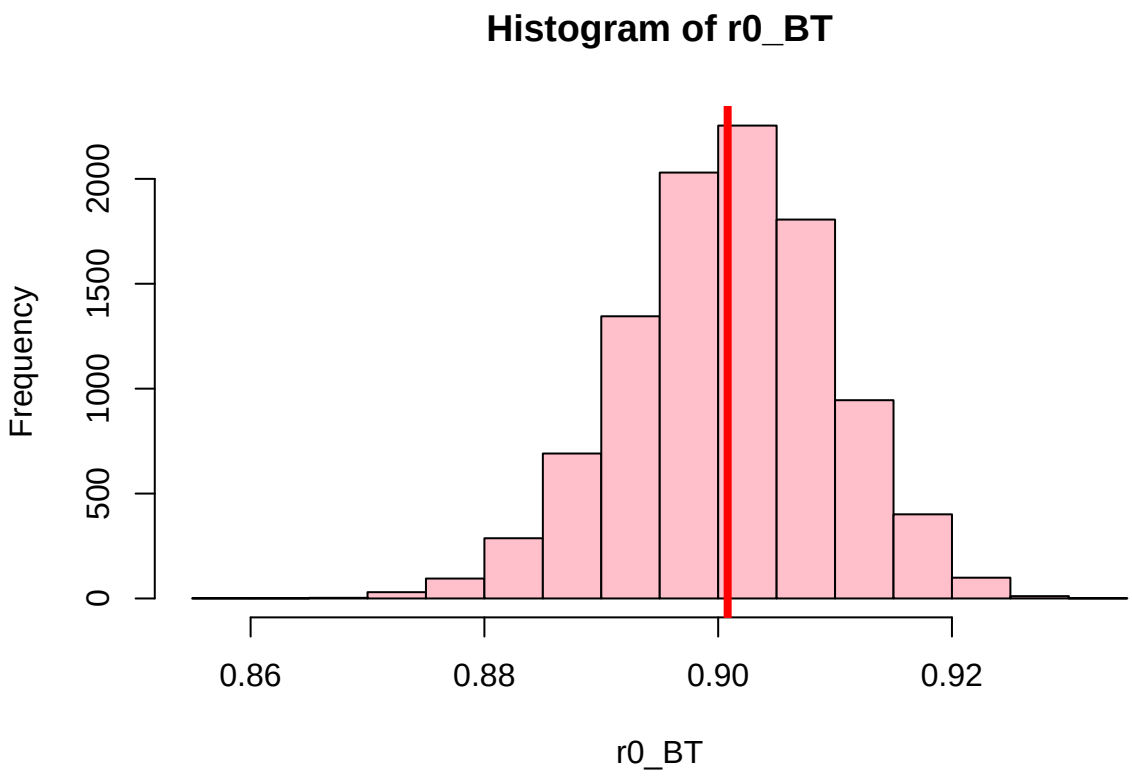
Sample correlation

The bootstrap can also be applied to estimate the uncertainty of a statistic of values of two variables such as the *sample correlation*. Here is one example using the `faithful` data again.

```
cor(faithful)
```

```
##            eruptions   waiting
## eruptions 1.0000000 0.9008112
## waiting   0.9008112 1.0000000
```

```r
  # this returns a sample correlation matrix

r0 = cor(faithful)[1,2]
  # this is the correlation between the two variables
n = nrow(faithful)
B = 10000
r0_BT = rep(NA, B)
for(i in 1:B){
  w = sample(n,n,replace=T)
  faithful_BT = faithful[w,]
  r0_BT[i] = cor(faithful_BT)[1,2]
}
hist(r0_BT, col="pink")
abline(v=r0, col="red",lwd=4)
```




```r
  # the distribution from the bootstrap
var(r0_BT)
```

```
## [1] 7.788901e-05
```

```r
  # the estimated variance

r0+qnorm(0.95)*sd(r0_BT)
```

```
## [1] 0.9153278
```

```r
r0-qnorm(0.95)*sd(r0_BT)
```

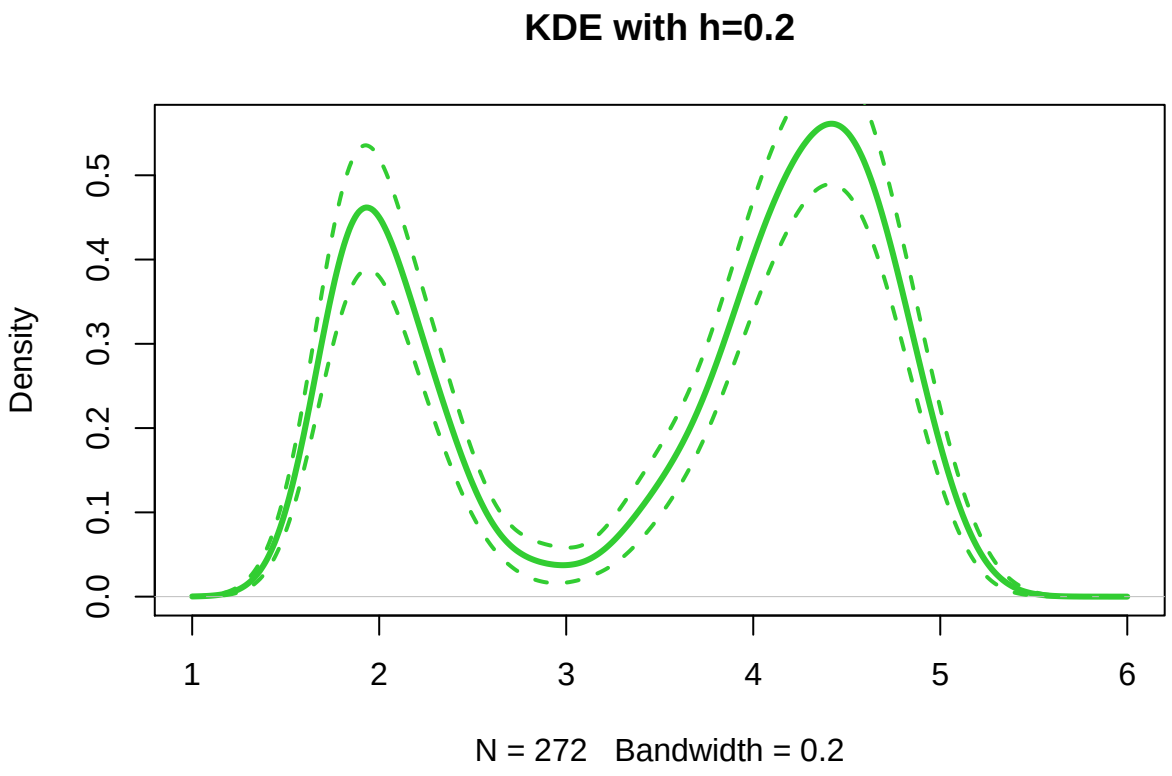
```
## [1] 0.8862946
```

```r
  # a 90% CI of the correlation
```

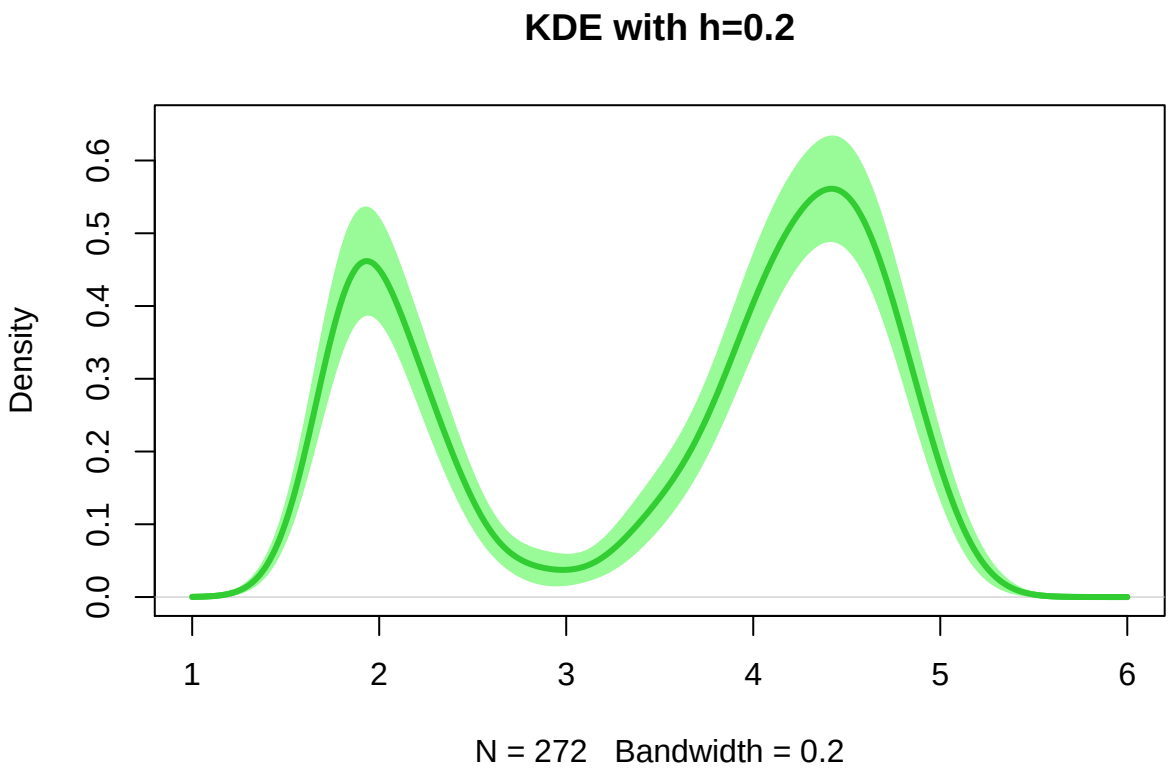
Density estimation

The bootstrap can also be applied to estimate the uncertainty of a density estimator. Here we will illustrate this by the KDE of variable `eruptions` in the `faithful` dataset.

```
dat = faithful$eruptions
h0 = 0.2
dat_kde = density(dat, bw=h0, from=1, to=6)
plot(dat_kde, lwd=3, col="limegreen", main="KDE with h=0.2")

kde_value = dat_kde$y
  # this is the density value at the grid point

n = nrow(faithful)
B = 10000
kde_value_BT = matrix(NA, nrow=B, ncol=length(kde_value))
for(i in 1:B){
  w = sample(n,n,replace=T)
  dat_BT = dat[w]
  kde_value_BT[i,] = density(dat_BT, bw=h0, from=1, to=6)$y
} # get the KDE of the bootstrap sample at each grid point
kde_value_sd = rep(NA, length(kde_value))
for(i in 1:length(kde_value)){
  kde_value_sd[i] = sd(kde_value_BT[,i])
} # compute the SD at each grid point

## making the plot
plot(dat_kde, lwd=3, col="limegreen", main="KDE with h=0.2")
lines(x=dat_kde$x, y=dat_kde$y+qnorm(0.95)*kde_value_sd, lwd=2, col="limegreen", lty=2)
lines(x=dat_kde$x, y=dat_kde$y-qnorm(0.95)*kde_value_sd, lwd=2, col="limegreen", lty=2)
```



```r
## a more fancy plot using 'polygon' function
plot(dat_kde, lwd=3, col="limegreen", main="KDE with h=0.2", ylim=c(0,0.65))
y_upper = dat_kde$y+qnorm(0.95)*kde_value_sd
y_lower = dat_kde$y-qnorm(0.95)*kde_value_sd
x_seq = dat_kde$x
polygon(x=c(x_seq, rev(x_seq)), y=c(y_upper, rev(y_lower)), col="palegreen", border="palegreen")
lines(dat_kde, lwd=3, col="limegreen")
```



Exercise - 2

Now we will consider the dataset `iris`.

1. We focus on the variable `Sepal.Length`.
   - What is the sample SD of this variable?
   - Use the bootstrap to show the distribution of the sample SD.
   - What is the variance of the sample SD computed using the bootstrap?
   - Find a 90% CI of the sample SD.
2. We focus on two variables `Sepal.Length` and `Sepal.Width`. And we want to analyze the correlation between them.
   - What is the sample correlation?
   - Use the bootstrap to show the distribution of the sample correlation
   - What is the variance of the sample correlation computed using the bootstrap?
   - Find a 90% CI of the sample correlation.