HW 4

Eric Zhao

2023-02-11

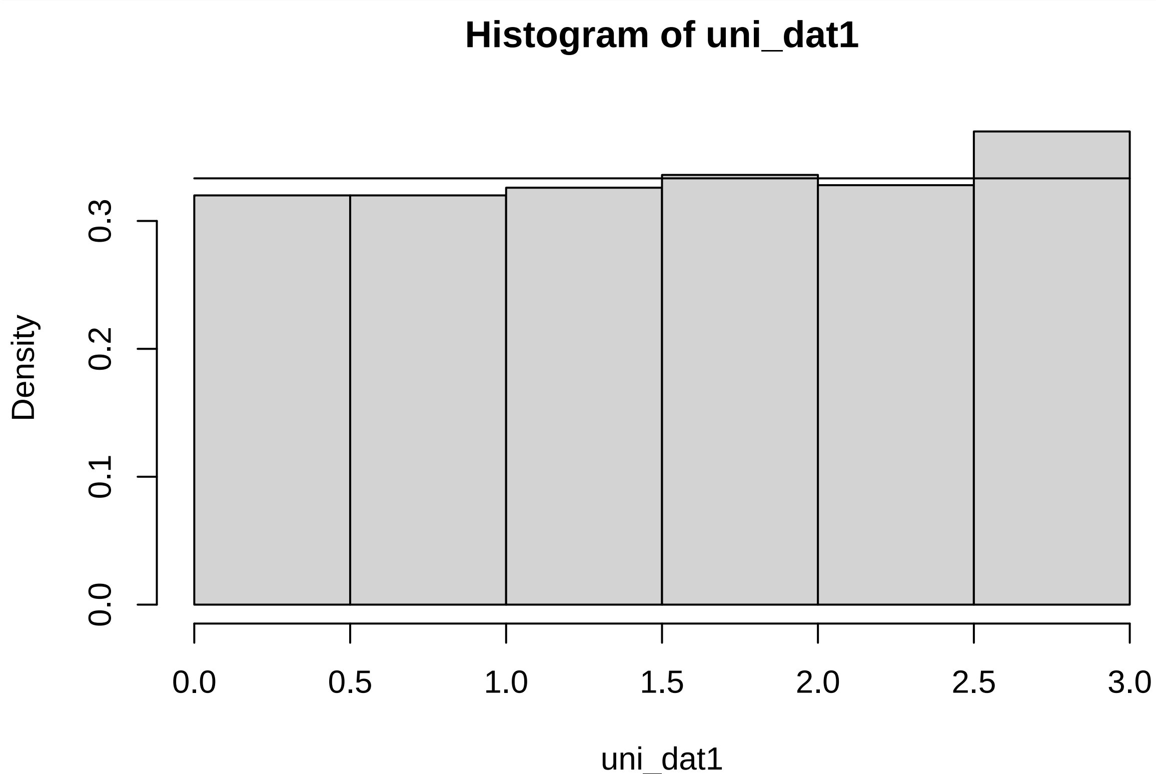
Uniform random variates

set-up myunif

```
myunif<-function(n, a, b){
  uni_variates<-rep(NA, n)
  for(i in 1:n){
    uni_variates[i]<-runif(1)*(b-a)+a
  }
  return(uni_variates)
}
```

simulated results

```
set.seed(1111)
uni_dat1<-myunif(1000, 0, 3)
hist(uni_dat1,freq = FALSE, breaks = "fd")
curve(dunif(x, 0, 3), add = TRUE)
```



```r
mean(uni_dat1)
```

```
## [1] 1.540079
```

```r
sd(uni_dat1)
```

```
## [1] 0.8816962
```

The mean and standard deviation are consistent with the theoretical values $\mu = \frac{3-0}{2} = 1.5$ and $\sigma = \frac{3}{\sqrt{12}} = 0.866025$.

Triangular Random variates

set-up mytriang

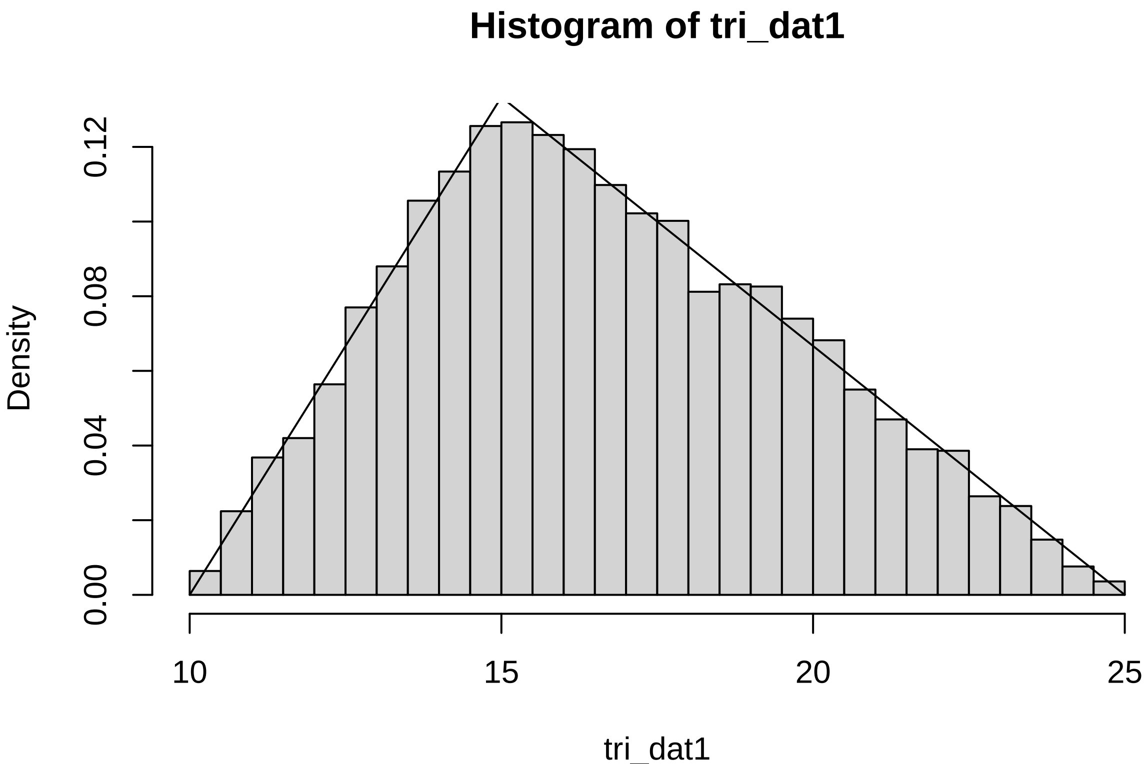
```r
mytriang<-function(n, a, b, c){
  tri_variates<-rep(NA, n)
  for(i in 1:n){
    call<-runif(1)
    if(call<(c-a)/(b-a)){
    tri_variates[i]<-sqrt(call*(b-a)*(c-a))+a
    }
    else{
      tri_variates[i]<-b-sqrt((1-call)*(b-a)*(b-c))
    }
  }
  return(tri_variates)

}
```

simulated results

```r
set.seed(1031)

tri_dat1<-mytriang(10000, 10, 25, 15)
hist(tri_dat1,freq = FALSE, breaks = "fd")
segments(x0=c(10, 15), y0=c(0, 2/15), x1=c(15, 25), y1=c(2/15, 0))
```




```
mean(tri_dat1)
```

```
## [1] 16.6347
```

```
sd(tri_dat1)
```

```
## [1] 3.099856
```

The mean and standard deviation are consistent with the theoretical values: $\mu = (10 + 25 + 15)/3 = 16.6667$ and $\sigma = \sqrt{\frac{1}{18}(10^2 + 25^2 + 15^2 - 10 \times 25 - 10 \times 15 - 15 \times 25)} = 3.11805$.

Normal random variates

Sum of Uniform Variates Method

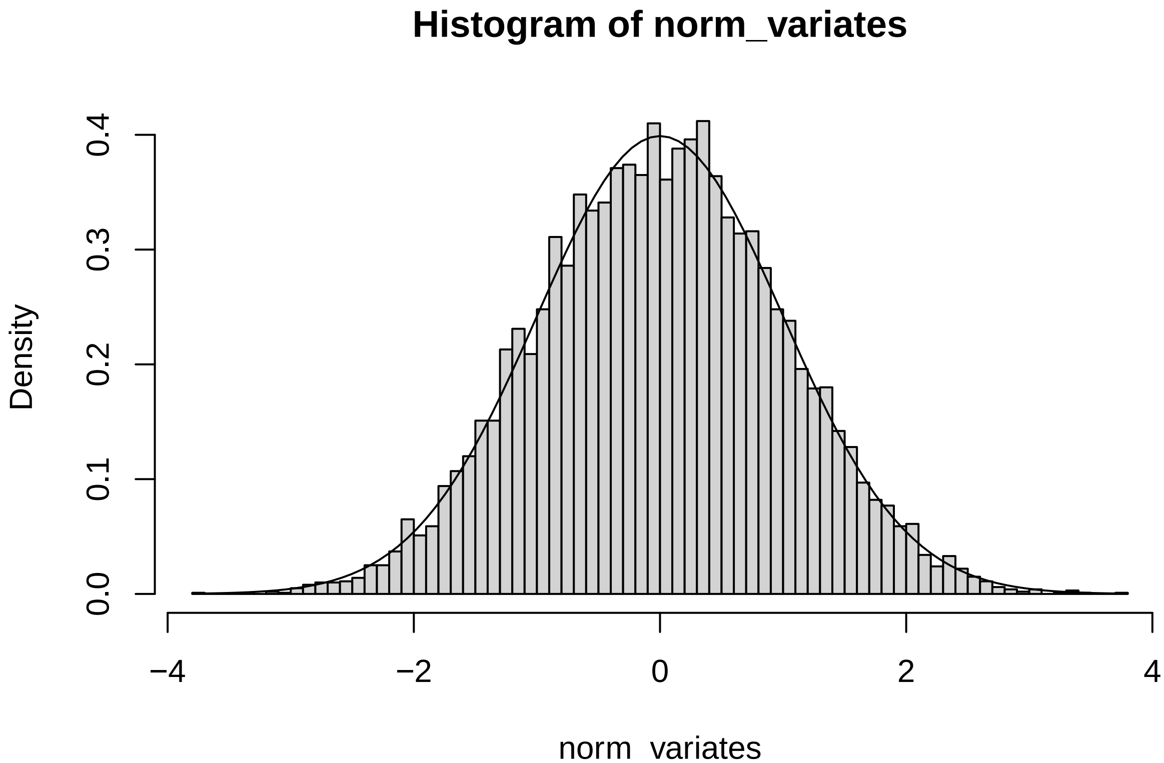
```
sumuni<- function(n=12){
    z<-(sum(myunif(12, 0, 1))-n/2)/sqrt(n/12)
    return(z)
}
```

set-up sumuni-approch of generating normal variate

generate normal variates and plot the simulation

```
set.seed(020131)
norm_variates<-rep(NA, 10000)
for (i in 1:10000) {
  norm_variates[i]<-sumuni(n=12)
}
```

```r
hist(norm_variates,freq = FALSE, breaks = "fd")
curve(dnorm(x), add = TRUE)
```



The normal distribution fit looks reasonable after superimpose the theoretical standard normal curve to it. The range of possible values under this seed is $[-3.718, 3.720]$.

Box-Muller Method

BMM function

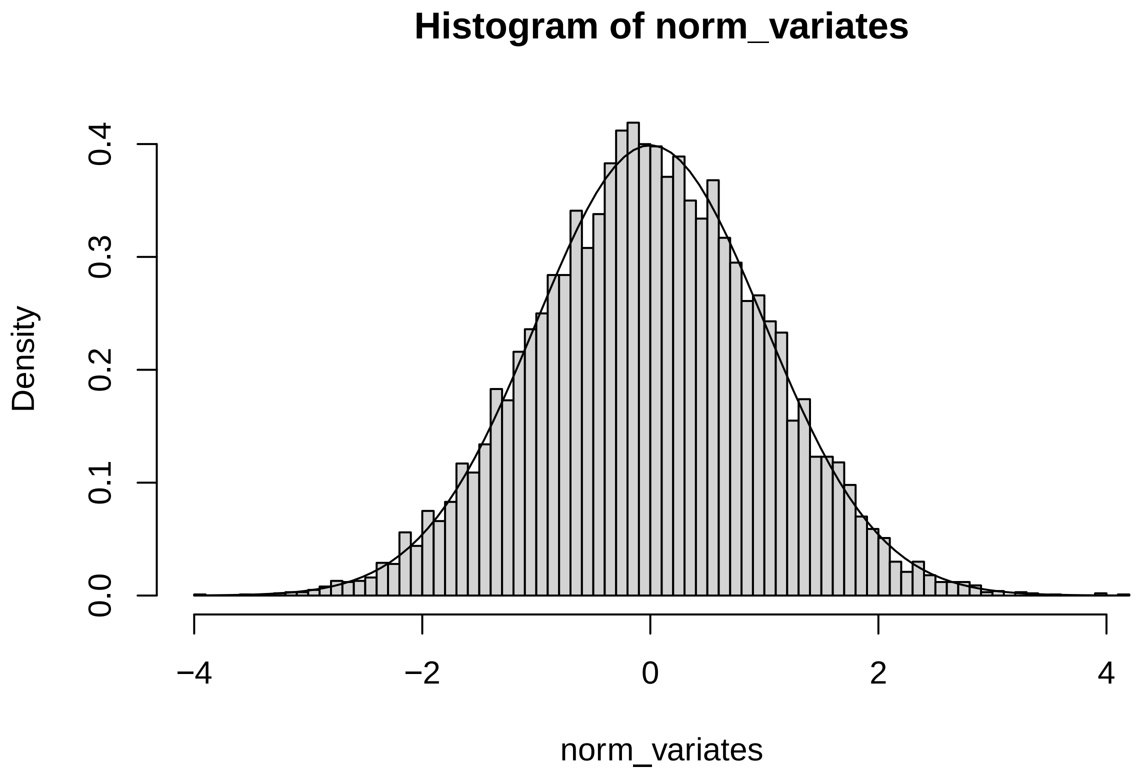
```r
BMM_norm <- function() {
  call1<-myunif(1, 0, 1)
  call2<-myunif(1, 0, 1)
  z1<-sqrt(-2*log(call1))*cos(2*pi*call2)
  z2<-sqrt(-2*log(call1))*sin(2*pi*call2)
  pair<-c(z1, z2)
  return(pair)
}
```

BMM normal variates simulation

```r
set.seed(021031)
norm_variates<-rep(NA, 10000)
for (i in 1:10000) {
  norm_variates[i]<-BMM_norm()[1]
  norm_variates[i+1]<-BMM_norm()[2]
}

hist(norm_variates,freq = FALSE, breaks = "fd")
```

```r
curve(dnorm(x), add = TRUE)
```



The standard normal variates that generated by this method also looks reasonable but this approach seems to be more efficient than the last sum of uniform variates approach. The possible range for z-score is around $[-4, 4]([-3.9054, 4.1996]$ in this case).