---

title: "Homework 2

By: Adham Suliman"

output: html\_document

---

1.1 Use runif()

```{R 1.1}

library("random")

set.seed(15)

Sample<-runif(1000,0,1)

```

1.2 Simulate uniform random sample on [0,1] Using random.org

```{R 1.2}

suppressWarnings(library(random))

nFlips<-1000

dataFromRandom<-randomNumbers(n=nFlips, min=0, max=1, col=1, base=2, check=TRUE)

head(dataFromRandom)

```

1.3 Downloading data from Random.org

```{R 1.3}

dataFromRandom<-read.table(paste("C:/Users/u353822/Documents/R/Statistical Analysis","randbyte.txt",sep="/"))

dataFromRandom<-na.omit(unname(unlist(dataFromRandom)))

dataFromRandom<-(as.vector(sapply(dataFromRandom,function(z) head(intToBits(z),8)))==1)\*1

head(dataFromRandom)

```

1.4 Turning binary sequence to uniform random numbers

```{R 1.4}

set.seed(15)

bitsToInt<-function(x) {

packBits(rev(c(rep(FALSE, 32-length(x)%%32), as.logical(x))), "integer")

}

bitsToInt(c(1,1,1,1,1,0))

Binary.matrix<-matrix(dataFromRandom,ncol=10)

head(Binary.matrix)

dataFromRandom.dec<-apply(Binary.matrix,1,bitsToInt)/2^10

head(dataFromRandom.dec)

```

2.1 Test uniformity of distribution of both random number generators.

```{R 2.1}

library("random")

set.seed(15)

Sample<-runif(1000,0,1)

Sample.histogram<-hist(Sample)

Sample.histogram

```

What does the histogram tell you about the distribution? Is it consistent with the goal of simulation?

The histogram seems to have a relatively equal distribution except from .3 to 1 where there seems to be a slight positive increase in frequencies.

Other than that, it is consistent with the goal of this simulation.

Estimate mean and standard deviation of Sample.histogram$density.

```{R histogram}

(Sample.histogram.mean<-mean(Sample.histogram$density))

(Sample.histogram.sd<-sd(Sample.histogram$density))

plot(Sample.histogram,freq=FALSE,ylim=c(0,Sample.histogram.mean+2\*Sample.histogram.sd))

abline(h=Sample.histogram.mean)

abline(h=Sample.histogram.mean+1.96\*Sample.histogram.sd,col="red",lty=2)

abline(h=Sample.histogram.mean-1.96\*Sample.histogram.sd,col="red",lty=2)

```

What does the graph tell you about the observed distribution?

This Graph tells me that the distribution is random because we fail to reject the null that it is random.

The area between the red lines represents 95% of the normal distribution for randomness and all the data falls in between these two lines.

Estimate moments of Sample

```{R moments}

(Sample.mean<-mean(Sample))

(Sample.variance<-var(Sample))

```

What do you conclude about the estimated distribution from the moments?

I would conclude that it is randomly distriubted due to the min, 1st quartile, mean, 3rd quartile, and max respectively falling nearly on 0,.25,.5,.75,1 which shows a random distribution from 0 to 1.

What do you think is the best way of estimating uniform distribution over unknown interval?

Check the summary of the simulated sample.If the quartiles can be obtained as was done above, one can then check the equivalency of .25\*max = 1st quartile. The other quartiles would then be compared in the same fashion . If the numbers are near one another, one can assume the distribution is uniform.

```{R simulated.sample}

summary(Sample)

```

2.1.2 Repeat the same steps to test unifromity of the sample from Random.org

```{R 2.1.2}

Sample.histogram<-hist(dataFromRandom.dec)

(Sample.histogram.mean<-mean(Sample.histogram$density))

(Sample.histogram.sd<-sd(Sample.histogram$density))

plot(Sample.histogram,freq=FALSE,ylim=c(0,Sample.histogram.mean+2\*Sample.histogram.sd))

abline(h=Sample.histogram.mean)

abline(h=Sample.histogram.mean+1.96\*Sample.histogram.sd,col="red",lty=2)

abline(h=Sample.histogram.mean-1.96\*Sample.histogram.sd,col="red",lty=2)

```

2.2.2 Test frequency by Monobit test

```{R 2.2.2}

dataFromRandom.plusminus1<-(dataFromRandom-.5)\*2

erf <- function(x) 2 \* pnorm(x \* sqrt(2)) - 1

erfc <- function(x) 2 \* pnorm(x \* sqrt(2), lower = FALSE)

```

3.1 My random number generator looks at sample of students in colleges from across the country and checks if they have financial support from their parents. A 1 means they do have financial support while a 0 means no financial support. The excel file is located on my local drive, and I don't know how to upload an excel into R markdown. Therefore I have copied the sequence below.

3.2

0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 1 0

1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1 1 0 1 0 0 1 1 1 0 0 0 1 1 0 1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 1 1

1 0 0 0 1 1 1 0 0 1 0 1 0 1 0 0 0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1

0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 1 1 1 0 1 1 1

1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1 1

0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0

```{R 3.2}

Support <- c(0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0)

```

3.3 Uniformity test.

```{R 3.3}

bitsToInt<-function(x) {

packBits(rev(c(rep(FALSE, 32-length(x)%%32), as.logical(x))), "integer")

}

matrix\_Support <-matrix(Support,ncol=7)

Nmatrix\_Support<-apply(matrix\_Support,1,bitsToInt)/2^7

hist\_Support <- hist(Nmatrix\_Support)

mean\_Support <- mean(hist\_Support$density)

sd\_Support <- sd(hist\_Support$density)

plot(hist\_Support,freq=FALSE,ylim=c(0,mean\_Support+2\*sd\_Support))

abline(h=mean\_Support)

abline(h=mean\_Support+1.96\*sd\_Support,col="red",lty=2)

abline(h=mean\_Support-1.96\*sd\_Support,col="red",lty=2)

```

It does pass the uniformity test because a majority of the data points fall within the 95% interval.

3.4 Frequency Test

```{R 3.4}

erf <- function(x) 2 \* pnorm(x \* sqrt(2)) - 1

erfc <- function(x) 2 \* pnorm(x \* sqrt(2), lower = FALSE)

matrix\_Support <-matrix(Support,ncol=5)

datafromsupport.plusminus1<-(Nmatrix\_Support-.5)\*2

erfc(abs(sum(datafromsupport.plusminus1)/sqrt(2\*395)))

```

Passes frequency test due to .237>.05

3.5 Turning Point Test

```{R 3.5}

library("randtests")

suppressWarnings(library(randtests))

turning.point.test(Nmatrix\_Support)

```

The turning point test deems this sample data set as random.

4.0 What percent do you needeto scratch off to make the quote readable? I need around 90% for it to be readable with the sample Funciton.

I need around 90% of it to be uncovered for it to be readable with the sobol sequence.I personally find the Halton sequence to be the best for this given set of text.

What percent do you need to scratch off to make the quote readable?

I need around 93.4% for it to be readable with the runif function

4.3 function runif() can be replaced by sobol() from library randtoolbox

```{R 4.3}

suppressWarnings(library('randtoolbox'))

library('randtoolbox')

load("C:/Users/u353822/Downloads/documents%2FMScA Statistical Analysis 31007%2FMScA 31007 Lecture 2%2FScratchOffMonteCarlo.rda")

set.seed(100)

nSample<-23000

xy<-sobol(nSample,dim=2,init=T)\*100

xy<-sobol(nSample,dim=2,init=F,scrambling = T,seed=my.seed)\*100

ScratchOffMonteCarlo(xy)

```

I need around 94.6% of it to be uncovered for it to be readable with the sobol sequence.

I only had to take 23,000 samples with Sobol to uncover 94% compared to taking 27,000 samples with the runif function above to uncover 91.86% of the image.

I personally find the Halton sequence to be the best for this given set of text. I took 11,000 samples, and I have around 86% visible with a seed set at 100.

I did 11,000 samples with the random number generators below:

Sample Function: 67%

Torus algorithm: 84%

Sobol sequence: 80%

congruRand: 67%

SFMT: 67%

WELL: 66%

knuthTAOCP: 67%

This shows that the Halton sequence is the best for uncovering with this set.seed and number of sample taken for this given image.

Changing the nSample size played a more significant role than changing the my.seed paramter in uncovering the quote.