NEW YORK INSTITUTE OF TECHNOLOGY

Fall 2021

Homework No. 5

Uniform Distribution

Name: Ravleen Kaur

Class ID: 43

School ID: 1113138

Course: Statistics for Data Science

Course ID: DTSC-620-W01

Date: 9/27/2021

Assignment Contents

Wrangling Question

What are the equivalent terms of the term “Wrangling”?

Data wrangling is the process of gathering, selecting, and transforming data to answer an analytical question.

The equivalent terms are:

- “Experimentation”
    - Used in natural science work, e.g., physics or chemistry
- “Trying”
    - Used in every day when solving unknown (previously unseen) problems
- “Hacking”
    - Used in programming, systems & network administration when quick right solution is not available
        - Every Programmer is a hacker!

“Wrangling” is used by data scientists when experimenting with data in search of :

- Unknown data properties, or
- Better data presentation (code)

Simulation: Uniform Discrete Variable Frequency, Small Sample 20/30 Small Span 2-4/2-5

Figure 1: • Use uniform "RW random experiment" simulator to generate integer values

```
> x <- as.integer(runif (20,2,4))
> x
 [1] 3 2 3 2 3 3 2 2 2 3 2 2 2 3 2 2 2 3 3 3
> t <- table(x)
> t
x
 2  3
11  9
> x <- as.integer(runif(30,2,5))
> x
 [1] 4 2 2 2 4 2 2 2 4 3 4 2 4 2 3 2 4 3 2 3 4 2 2 4 4 4 2 4 3 3
> t <- table(x)
> t
x
 2  3  4
13  6 11
> x <- as.integer(runif(100,0,10))
> x
  [1] 7 5 3 9 4 0 6 9 0 6 0 6 7 6 7 1 2 3 6 7 3 6 6 0 2 4 8 1 1 7 5 3 1 3 7 5 8
 [38] 3 5 7 2 6 6 3 3 7 4 2 4 0 4 8 5 2 2 9 4 7 3 6 9 5 2 2 0 5 3 8 0 7 4 1 4 9
 [75] 6 1 7 3 7 5 3 9 7 7 4 0 3 2 4 1 3 2 7 7 4 5 0 3 1 4
```

Hack Session: Simulating Uniformly Distributed Variables (n=100), Outlier Values, Hacking *table()* Function, Using Filer Masks

```
> t[names(t)==6]
6
8
> t[names(t)==7]
 7
10
> names(t)
 [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
> names(t)==6
 [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
> names(t)>=6
 [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> t[7]
6
8
> t[7:9]
x
 6  7  8
 8 10 14
> tf <- names(names(t) >= 6) ; tf
NULL
> t[tf]
named integer(0)
> tf <- (names(t) >= 6) ; tf
 [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> t[tf]
x
 6  7  8  9
 8 10 14 13
```

Figure 2: This illustrates a bigger sample of n=100 values.

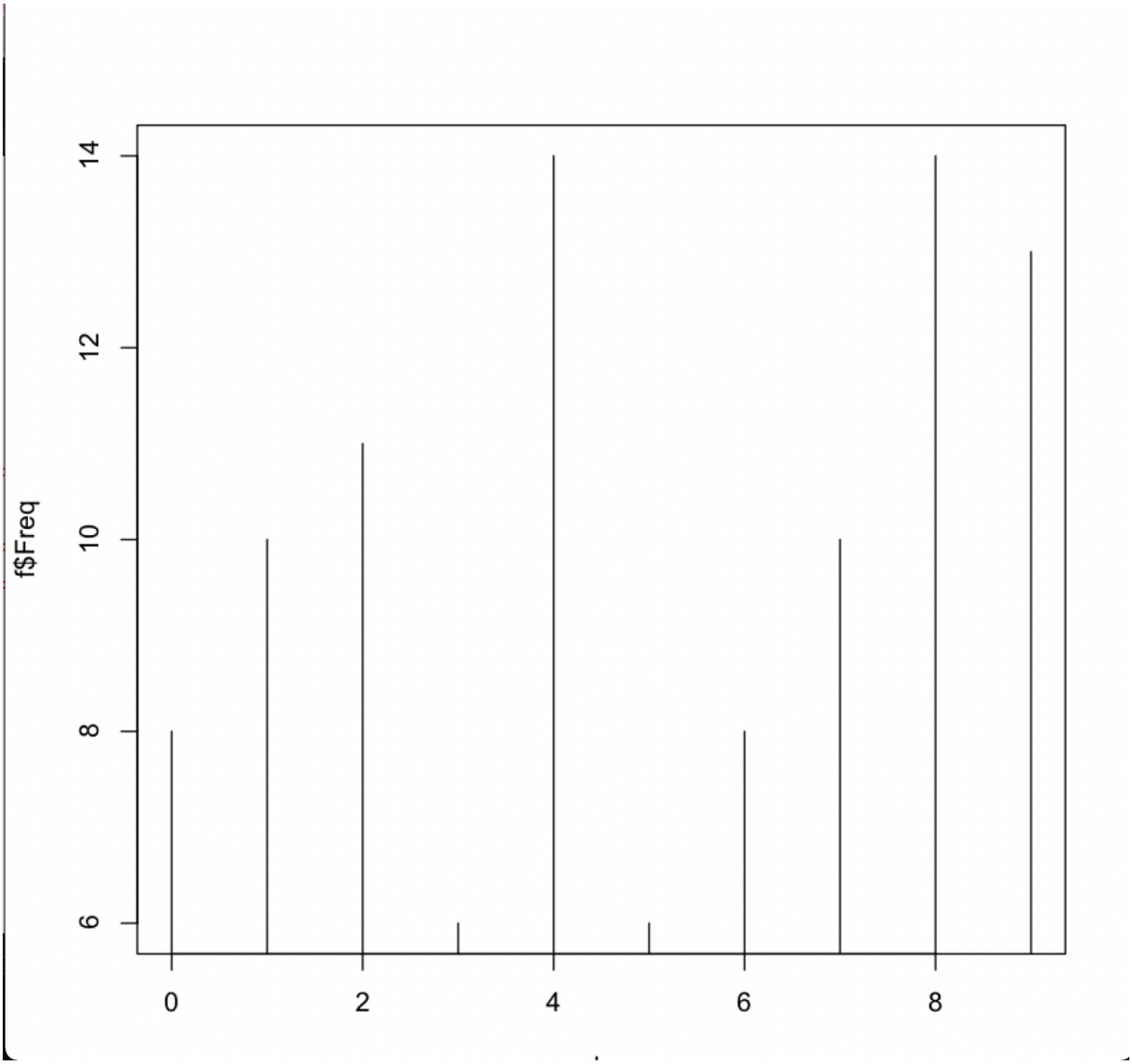
Outliers are unusually large (odd) values that are rare but always possible (with small probability). Outliers are biggest problems for linear non robust statistical methods (Dr. Bill's PhD thesis deals with crushing outliers)To experiment with statistical data using R as a tool, it is important to master R data type manipulation functions.Data elements can be selectively extracted from vectors, tables, and frames using MASKs (e.g., index values like t[7] or by using TRUE/FALSE vector as a FILTER MASK)

Hack Session: Changing Data Codes, Data Type Conversion (Table to Frame), Filtering Frame Data, Frequency Data Plot

- # Data frame organization can be more convenient for analysis > f
- The same data can be found in table t and frame f
- Symbol $ is “in object” element locator operator. Find element Freq in f
- All frame data are filtered out except column Freq

```
> f <- as.data.frame(t)
> f
   x Freq
1  0    8
2  1   10
3  2   11
4  3    6
5  4   14
6  5    6
7  6    8
8  7   10
9  8   14
10 9   13
> f$Freq
 [1]  8 10 11  6 14  6  8 10 14 13
> f$x
 [1] 0 1 2 3 4 5 6 7 8 9
Levels: 0 1 2 3 4 5 6 7 8 9
> plots(f$x,f$Freq, type=)
Error in plots(f$x, f$Freq, type = ) : could not find function "plots"
> > plots(f$x,f$Freq, type='h')
Error: unexpected '>' in ">"
> plots(f$x,f$Freq, type='h')
Error in plots(f$x, f$Freq, type = "h") : could not find function "plots"
> plots(f$x,f$Freq,type='h')
Error in plots(f$x, f$Freq, type = "h") : could not find function "plots"
> plot(f$x,f$Freq, type='h')
> h <- c(0:9)
> h
 [1] 0 1 2 3 4 5 6 7 8 9
> plot(h,f$Freq, type='h')
> f
   x Freq
1  0    8
2  1   10
3  2   11
4  3    6
5  4   14
6  5    6
7  6    8
8  7   10
```

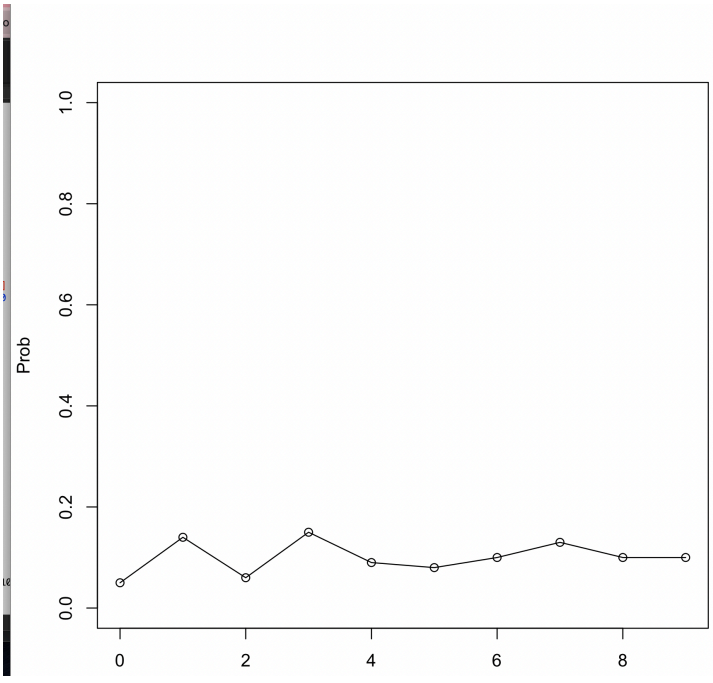
- f$x is not always good for x-axis, we shall make h-horizontal equivalent

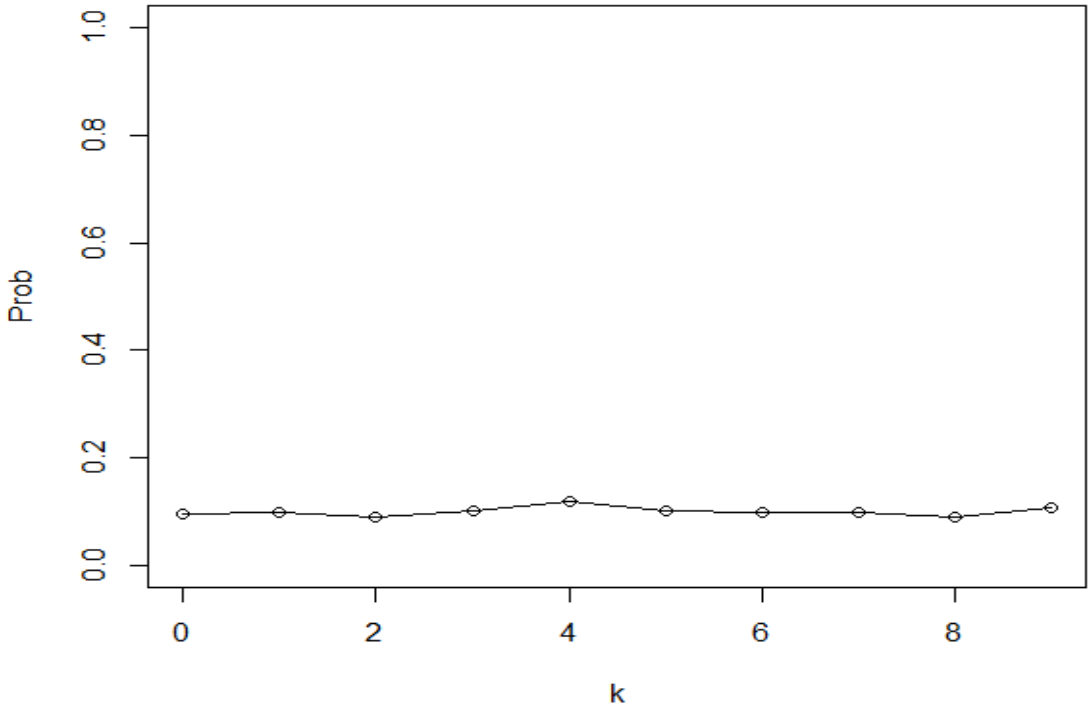


This is our illustrated data plot

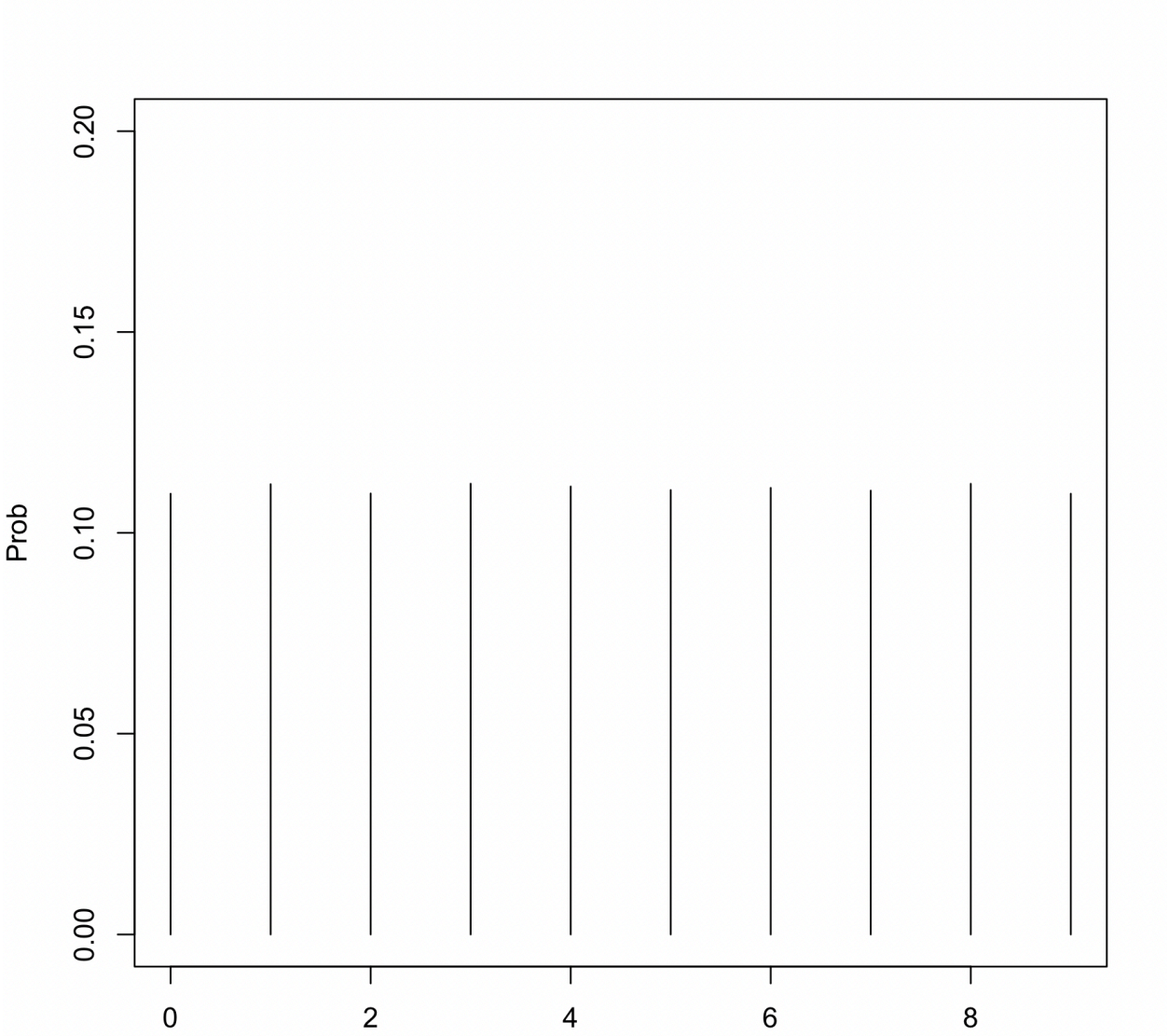
Simulation: Uniform Random Values Frequency. Line Plot for Visibility of Flatness, Discrete Uniform Distribution

```
> n=100 # Vary n=100,1000,10000,100000
> k <- c(0:9); k
 [1] 0 1 2 3 4 5 6 7 8 9
> x <- as.integer(runif(n,0,10))
> t<- table(x) ; t
x
 0  1  2  3  4  5  6  7  8  9
 5 14  6 15  9  8 10 13 10 10
> f <- as.data.frame(t);f
   x Freq
1  0    5
2  1   14
3  2    6
4  3   15
5  4    9
6  5    8
7  6   10
8  7   13
9  8   10
10 9   10
> freq <- f$Freq;freq
 [1]  5 14  6 15  9  8 10 13 10 10
> Prob <- freq/n ; Prob
 [1] 0.05 0.14 0.06 0.15 0.09 0.08 0.10 0.13 0.10 0.10
> plot(k,Prob, ylim=c(0,1), type='o')
>
```




Prob
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
k

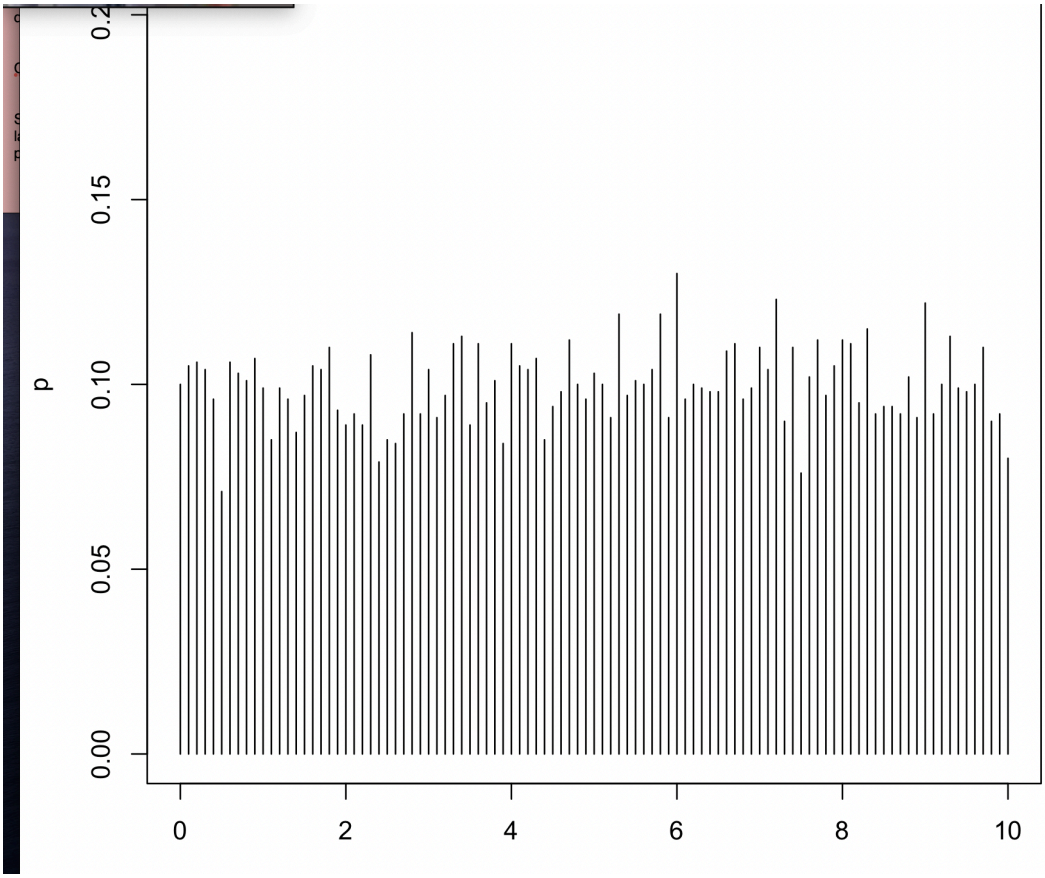
```
>  n=100000 # Vary n=100,1000,10000,100000
> k <- c(0:9)
> x <- round(runif(n,0,9),digits=0)
> t <- table(x) ;t
x
    0     1     2     3     4     5     6     7     8     9
 5487 11211 10980 11224 11154 11066 11119 11051 11221  5487
> f <- as.data.frame(t)
> prob <- f$freq
> prob <-prob/n
> prob[1] <- prob[1]*2
> prob[10] <- prob[10]*2
> prob
 [1] NA NA NA NA NA NA NA NA NA NA
> prob <- f$Freq
> Prob <- f$Freq
> Prob <- Prob/n
> Prob[1] <- Prob[1]*2
> Prob[10] <- Prob[10] *2
> Prob
 [1] 0.10974 0.11211 0.10980 0.11224 0.11154 0.11066 0.11119 0.11051 0.11221 0.10974
> plot(k,Prob,ylim=(0,0,2), type"h")
Error: unexpected ',' in "plot(k,Prob,ylim=(0,"
> plot(k,Prob, ylim=c(0,0,2), type="h")
Error in plot.window(...) : invalid 'ylim' value
```



Debugging Simulation Problem, Biased & Unbiased PMF Plots, Increasing Density of k-values in range [0-10], Outliers Visibility

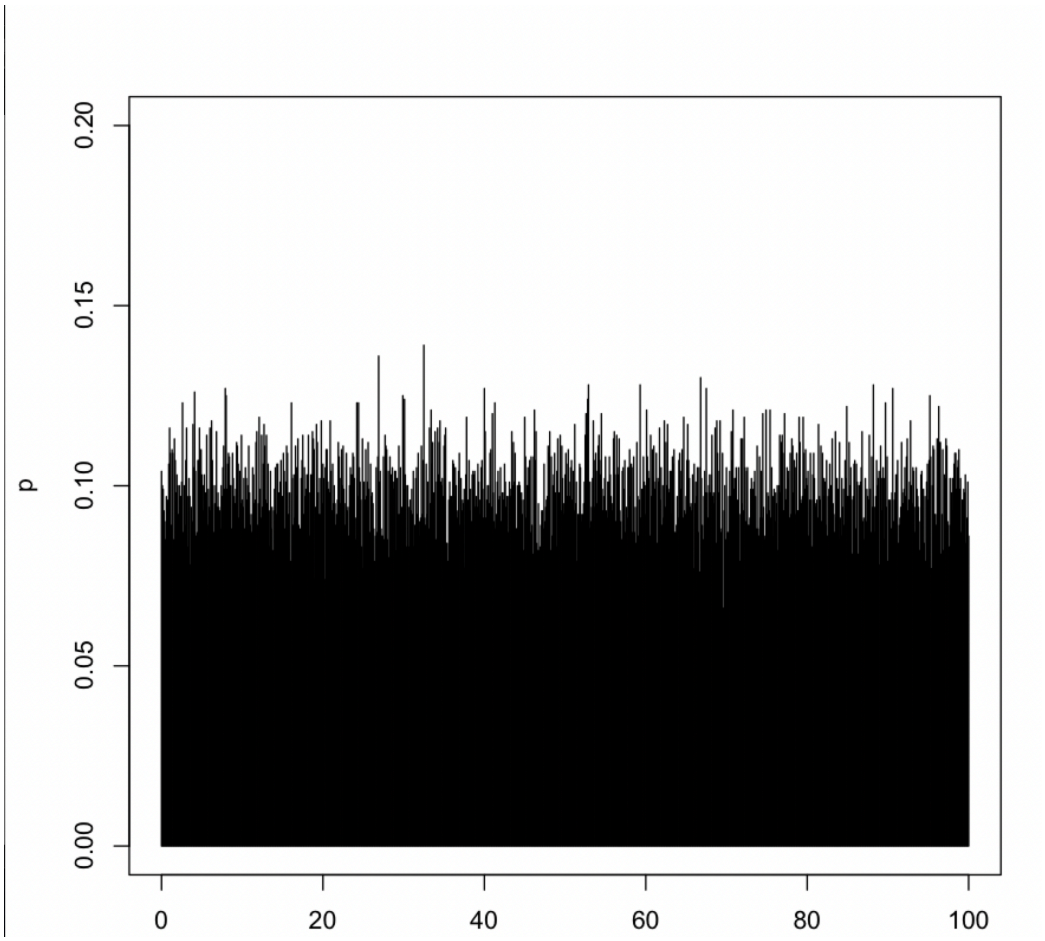
When rounding numbers, numbers in the sequence {1,2,. . .,n} get left & right values contributing (e.g., 2 gets values 1.5<x<2.5). Edge 0th and nth (last) get contributions from only one side. We shall divide the range 0-10 into 100 allowed k values. Simulation generates n=10,000 uniform variable values. Plot of unbiased PMF has corrected Prob[1] and Prob[101].

```
> n=10000
> k <- c:(0:100) / 10;
Error in c:(0:100) : NA/NaN argument
In addition: Warning message:
In c:(0:100) : numerical expression has 101 elements: only the first used
> k <- c(0:100)/10;
> length(k)
[1] 101
> x<-round(runif(n,0,10),digits=1)
> length(x)
[1] 10000
> t<-table(x)
> f<-as.data.frame(t)
> p<-f$Freq/(n/10) # Biased
> length(p)
[1] 101
> p[1] <- p[1]*2
> p[101] <- p[101]*2
> plot(k,p,ylim=c(0,0.2),type="h")
```



We will now increase the number of test samples to n=100,000 with k=1000=n/100 and see how that improves uniform distribution estimated shape/flatness-of-probability-values.

```
> p[101] <- p[101]*2
> plot(k,p,ylim=c(0,0.2),type="h")
> n=100000
> k<-c(0:1000)/10;
> length(k)
[1] 1001
> x<-round(runif(n,0,10),digits=2)
> length(x)
[1] 100000
> t<-table(x)
> f<-as.data.frame(t)
> p<-f$Freq/(n/10) # Biased
> p<-f$Freq/(n/100) # Biased
> length(p)
[1] 1001
> p[1]<-p[1]*2
> p[1001]<-p[1001]*2
> plot(k,p, ylim=c(0,0.2), type=)
>
> plot(k,p, ylim=c(0,0.2), type="h")
```



Some outliers can harm (increase or decrease) average values to misleading and wrong values (conclusions). This is a very rare case however.

Here is a change in plot scale. With larger numbers of possible values, discrete PDF/PMF, P(x) approaches continuous PDF p(x).

```
> n=100000
> k<-c(0:1000)/100;
> length(k)
[1] 1001
> x<-round(runif(n,0,10), digits=2)
> length(x)
[1] 100000
> t<-table(x)
> f<-as.data.frame(t)
> p<- f$Freq/(n/1000)
> length(p)
[1] 1001
> p[1] <- p[1] *2
> p[1001] <- p[1001]*2
> plot(k,p,ylim=c(0,1),type='o')
2021-10-18 18:43:29.048 R[2400:135503] -deltaZ is deprecated for NSEventTypeMagnify.  Please
use -magnification.
> > p<- f$Freq/(n/100)
Error: unexpected '>' in ">"
> p<- f$Freq/(n/100)
>
> length(p)
[1] 1001
> p[1] <- p[1] *2
> p[1001] <- p[1001]*2
> plot(k,p,ylim=c(0,1),type='o')
>
```