random-variables

09/03/2022

Contents

Discrete Random Variables

```
sample(1:10, 1)
```

```
## [1] 8
```

```
sample(1:10, 10)
```

```
##  [1]  6 10  4  1  9  3  2  8  7  5
```

```
sort(sample(1:10, 10))
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
sample(10:1,1)
```

```
## [1] 6
```

```
sample(10:1.5,1)
```

```
## [1] 8
```

```
sample(2:1.5,1)
```

```
## [1] 2
```

```r
sample(0,0,1)
```

```
## numeric(0)
```

```r
sample(1:10, 11, replace=T)
```

```
##  [1]  8  8  4 10  2  4 10  7 10 10  4
```

```r
table(sample(1:10, 100, replace=T))
```

```
##
##  1  2  3  4  5  6  7  8  9 10
##  8 13 10 12 10  9 13  9  8  8
```

average number of times a value is sampled

```r
mean(table(sample(1:10, 100, replace=T)))
```

```
## [1] 10
```

```r
mean(sample(1:10, 100, replace=T))
```

```
## [1] 5.51
```

```r
table(sample(1:10, 1000, replace=T))
```

```
##
##   1   2   3   4   5   6   7   8   9  10
##  96 112 112 108  94 102 109  87  87  93
```

```r
table(sample(1:10, 1000000, replace=T))
```

```
##
##      1      2      3      4      5      6      7      8      9     10
## 100308 100059  99954 100224  99746  99942  99455 100154 100096 100062
```

variation

```r
sd(sample(1:10, 1000, replace=T))
```

```
## [1] 2.901092
```

```r
var(sample(1:10, 1000, replace=T))
```

```
## [1] 7.759744
```

run if ()

Generate a random number between 5.0 and 7.5

```r
runif(1, 5.0, 7.5)
```

```
## [1] 5.243648
```

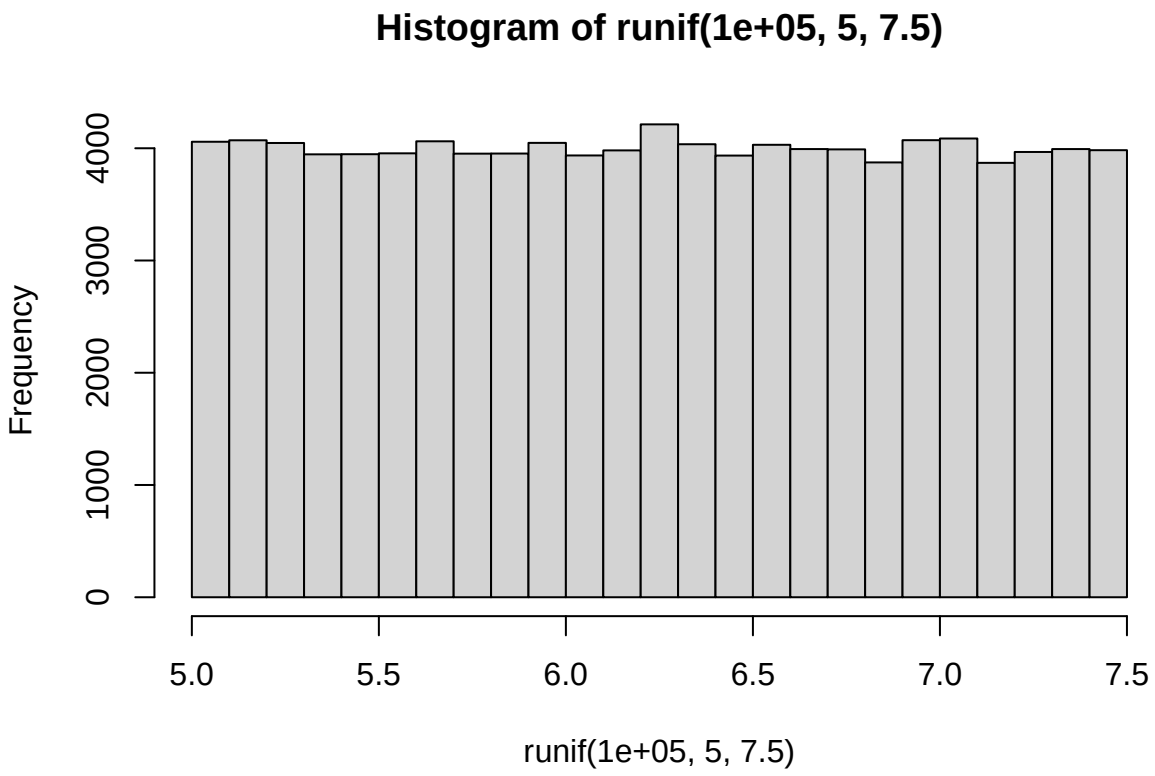
```r
runif(100, 5.0, 7.5)
```

```
##   [1] 6.659059 6.736175 5.702081 7.091700 6.599503 6.293012 5.206413 6.260227
##   [9] 6.091999 5.686823 7.359793 6.150448 6.761422 7.041922 5.855655 5.460315
##  [17] 5.567755 6.098280 5.291008 6.125943 6.918703 5.405862 6.861599 6.417497
##  [25] 6.007000 7.083969 6.861201 6.683687 5.740076 5.127927 7.452175 6.435930
##  [33] 5.667678 5.223174 7.124415 5.986355 6.400921 6.066792 5.926089 5.261668
##  [41] 7.278579 5.003770 7.034211 6.625445 7.159193 7.462393 5.106139 5.455043
##  [49] 6.044764 6.834133 7.461922 7.329534 5.786073 6.449740 7.084350 5.240894
##  [57] 5.698383 6.600629 6.043933 6.229831 5.866165 5.918229 6.756151 7.467291
##  [65] 7.004666 7.342359 5.532396 5.771510 7.276128 7.369493 6.975631 7.307050
##  [73] 5.950482 5.692328 5.712525 7.464311 7.448375 5.731440 7.312786 5.139793
##  [81] 5.145720 5.108815 5.786449 7.431876 6.784873 6.670297 7.390819 6.263738
##  [89] 5.150381 6.220364 6.901357 7.036933 6.187269 6.412670 7.004494 6.724756
##  [97] 5.606912 6.781445 5.243855 6.337733
```

```r
min(runif(1000, 5.0, 7.5))   # display the minimum
```

```
## [1] 5.000392
```

```r
hist(runif(100000,5.0,7.5))  # display the frequency of number as histogram
```



Binomial distribution

Suppose there are 12 multiple questions in a quiz. Each questions has 5 possible answers, and only 1 of them is correct. Find the probability of having 4 or less correct answers if a student attempts to answer every questions at random.

```
dbinom(4, size=12, prob=0.2)
```

```
## [1] 0.1328756
```

To find the probability of having four or less correct answers by random attempts, we apply the function dbinom with x = 0,. . . ,4.

```
dbinom(0, size=12, prob=0.2) +
+ dbinom(1, size=12, prob=0.2) +
+ dbinom(2, size=12, prob=0.2) +
+ dbinom(3, size=12, prob=0.2) +
+ dbinom(4, size=12, prob=0.2)
```

```
## [1] 0.9274445
```