Data 605 Final Project for Problem 1

Jun Pan

05-25-2019

Generate Random Numbers

- Using R, generate a random variable X that has 10,000 random uniform numbers from 1 to N, where N can be any number of your choosing greater than or equal to 6. Then generate a random variable Y that has 10,000 random normal numbers with a mean of (N+1)/2.

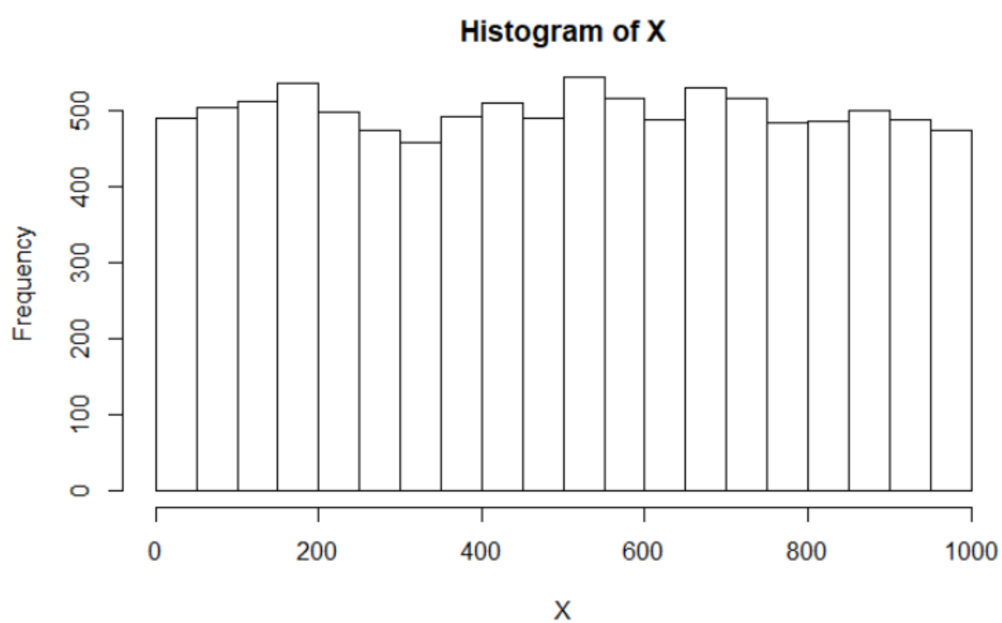
```{r}
N <- 1000
X <-  runif(10000, min=0, max=N)# number between 0 and 1000
Y <- rnorm(10000, mean=(N+1)/2, sd=(N+1)/2)# mean and standard deviation is (N+1)/2
```

X

```{r}
summary(X)
```

```
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
 0.0653 252.8918 494.5676 497.5494 743.3941 999.9414 
```

```{r}
hist(X)
```



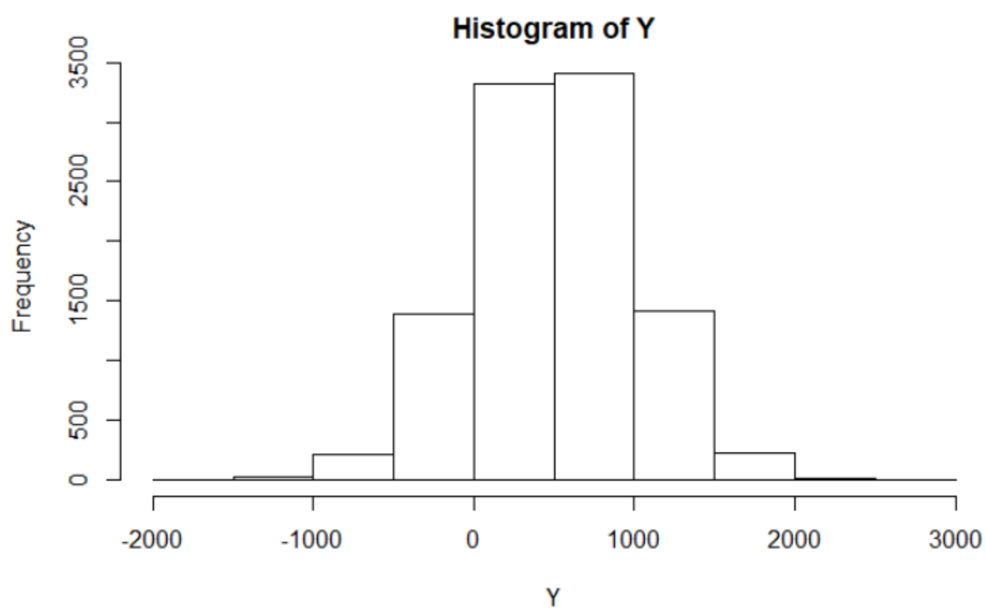
Uniform distribution of X

Y

```{r}
summary(Y)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-1424.1   167.5   498.2   501.5   849.0  2426.3 
```

```{r}
hist(Y)
```



Random distribution of X

Get “x” and “y”

#Probability. Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the median of the X variable, and the small letter "y" is estimated as the 1st quartile of the Y variable. Interpret the meaning of all probabilities.

small letter "x" is estimated as the median of the X variable

```{r}
x <- median(X)
```

small letter "y" is estimated as the 1st quartile of the Y variable

```{r}
y <- quantile(Y, 0.25)
```

```{r}
x
```

```
[1] 494.5676
```

```{r}
y
```

```
     25%
167.4882
```

a. P(X>x | X>y)

```
# a. P(X>x | X>y)
# Is computed by taking P(X > x and Y > y) divided by P(Y > y)
# Probability P(X > x and Y > y)
```{r}
p1 <- length(which(X > x & Y > y) == TRUE) / length(X)
p1
```
```

```
[1] 0.3756
```

```
# Probability P(Y > y)
```{r}
p2 <- length(which(Y > y) == TRUE) / length(Y)
p2
```
```

```
[1] 0.75
```

```
# Probability P(X > x and Y > y) divided by P(Y > y)
```{r}
a <- p1 / p2
print(a)
```
```

```
[1] 0.5008
```

b. P(X>x, Y>y) = P(X>x & Y>y)

```
#b. P(X>x, Y>y) = P(X>x & Y>y)
```{r}
b <- length(which(X > x & Y > y) == TRUE) / length(X)
print(b)
```
```

```
[1] 0.3756
```

c. P(X<x | X>y)

```
#c. P(X<x | X>y)
# Is computed by taking P(X < x and Y > y) divided by P(Y > y)
# Probability P(X > x and Y > y)
```

```{r}
p1 <- length(which(X < x & Y > y) == TRUE) / length(X)
p1
```

```
[1] 0.3744
```

```
# Probability P(Y > y)
```

```{r}
p2 <- length(which(Y > y)== TRUE) / length(Y)
p2
```

```
[1] 0.75
```

```
# Probability P(X > x and Y > y) divided by P(Y > y)
```

```{r}
c <- p1 / p2
print(c)
```

```
[1] 0.4992
```

Investigate whether P(X>x and Y>y)=P(X>x)P(Y>y) by building a table and evaluating the marginal and joint probabilities

```{r}
probability_table <- c(length(which(X < x & Y < y) == TRUE),length(which(X < x & Y== y) == TRUE),length(which(X < x & Y > y) == TRUE))
probability_table <-rbind(probability_table,c(length(which(X == x & Y < y) == TRUE),length(which(X == x & Y == y) == TRUE),length(which(X == x & Y > y) == TRUE)))
probability_table <- rbind(probability_table,c(length(which(X > x & Y < y) == TRUE),
length(which(X > x & Y == y) == TRUE), length(which(X > x & Y > y) == TRUE)))
probability_table <- cbind(probability_table,rowSums(probability_table))
probability_table <- rbind(probability_table,colSums(probability_table))
colnames(probability_table) <- c("Y<y","Y=y","Y>y","Total")
rownames(probability_table) <- c("X,x","X=x","X>x","Total")
knitr::kable(probability_table)
```

	Y<y	Y=y	Y>y	Total
X,x	1256	0	3744	5000
X=x	0	0	0	0
X>x	1244	0	3756	5000
Total	2500	0	7500	10000

```
#As we have constructed the marginal and joint probility table. Now we need to check for condition
# P(X>x and Y>y)
# X>x probability_table[11]
# Total probability_table[16]
```{r}
probability_table[11]/probability_table[16]
```
```

```
[1] 0.3756
```

```
# P(x>x)P(Y>y)
```{r}
((probability_table[15]/probability_table[16])*(probability_table[12]/probability_table[16]))
```
```

```
[1] 0.375
```

```
# As both the probilities are aprroximately same so this proves P(X>x and Y>y) = P(X>x)P(Y>y)
```

Check to see if independence holds by using Fisher's Exact Test and the Chi Square Test. What is the difference between the two? Which is most appropriate?

```{r}
data_fisher <- table(X > x, Y > y)
fisher.test(data_fisher)
```

```
	Fisher's Exact Test for Count Data

data:  data_fisher
p-value = 0.7995
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.9242273 1.1100187
sample estimates:
odds ratio
  1.012883
```

#As p value is greater than 0.05. so we cannot reject the null hypothesis, so we can conclude that both events are independent

```{r}
data_chi <- table(X > x, Y > y)
chisq.test(data_chi)
```

```
	Pearson's Chi-squared test with Yates' continuity correction

data:  data_chi
X-squared = 0.064533, df = 1, p-value = 0.7995
```

#As p value is greater than 0.05, so we cannot reject the null hypothesis, so we can conclude that both events are independent.

#Fisher's Exact test is a way to test the association between two categorical variables when you have small cell sizes (expected values less than 5). Chi-square test is used when the cell sizes are expected to be large. If your sample size is small (or you have expected cell sizes <5), you should use Fisher's Exact test. Otherwise, the two tests will give relatively the same answers. With large cell sizes, their answer should be very similar.