Homework #4

Dustin Leatherman

4/26/2020

5

> Consider the linear congruential random number generator with $M = 17$, and a possibly equal to 3, 4, or 5. Which one of these values of a would be best?

a is defined as a primary root of M and it should ideally be the *largest* primary root of M. a is considered a primary root if $a^i \ mod \ M > 0$ for $i = 1, 2, ..., M - 1$.

```
M = 17
3^seq(1:M - 1) %% 17
```

```
## [1]  3  9 10 13  5 15 11 16 14  8  7  4 12  2  6  1  3
```

```
4^seq(1:M - 1) %% 17
```

```
## [1]  4 16 13  1  4 16 13  1  4 16 13  1  4 16 13  1  4
```

```
5^seq(1:M - 1) %% 17
```

```
## [1]  5  8  6 13 14  2 10 16 12  9 11  4  3 15  7  1  5
```

3, 4, and 5 are primary roots of 17 since there are no 0 values. Thus, the largest value, $a = 5$, would be considered best.

6

> The Poisson Random Variable is a discrete random variable that models how many taxis come by in a fixed time, or how many charged particles are detected in a fixed time, or how hard drive crashes in a fixed time. The probability mass function of the Poisson Random Variable with mean (and variance) λ is
>
> $$f(x) = \begin{cases} \frac{\lambda^x e^{-\lambda}}{x!}, & x = 0, 1, ... \\ 0, & else \end{cases}$$
>
> Write an R program to generate i.i.d Poisson Random Variables with mean one, $X_1, ..., X_n$ from uniform random variables, $Z_1, ..., Z_n$, using the inverse distribution transformation method. Use your program to print out $Z_1, ..., Z_n$ and $X_1, ..., X_n$ for n = 10

```
n <- 10
x <- runif(n)
z <- qpois(x, 1)
```

X = 0.4387646, 0.3198826, 0.5047215, 0.4938895, 0.5978032, 0.0474601, 0.7252075, 0.7735053, 0.1407829, 0.3445013

Z = 1, 0, 1, 1, 1, 0, 1, 2, 0, 0

7

Consider a distribution with CDF

$$F(y) = \frac{1}{1 + e^{-y}},\ y \in R$$

Use the inverse distribution transformation method to generate random variables with the above CDF from U[0,1] random variables. Generate 1000 such random numbers.

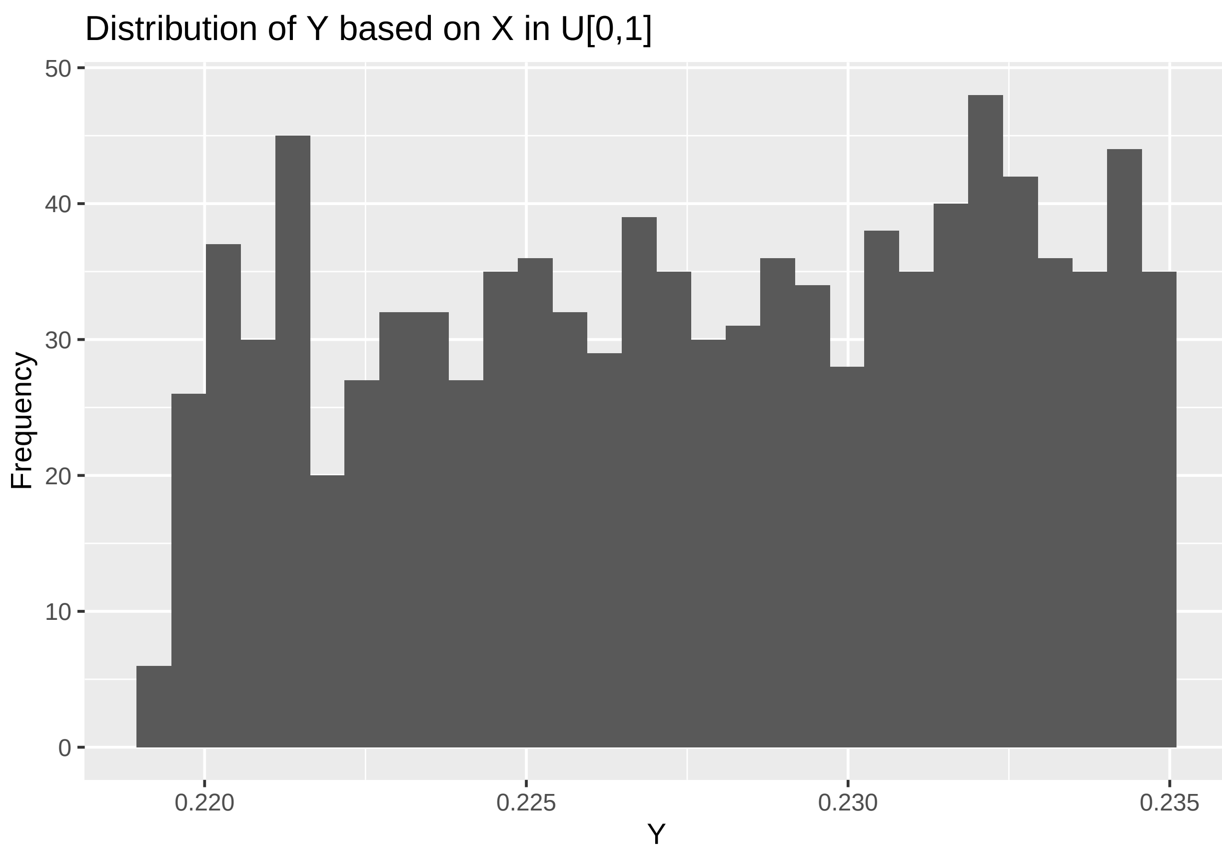
```
n <- 1000
cdf <- function (y) 1 / (1 + exp(-y))

# analytical derivative of PDF
pdf <- function(y) exp(-y) / (1 + exp(-y))^2

# compute first derivative programmatically for PDF
computed.pdf <- D(expression(1 / (1 + exp(-y))), "y")

x <- runif(n)
y <- cdf(x)
y1 <- pdf(y)
y2 <- eval(computed.pdf)

assertthat::are_equal(y1,y2)
```

```
## [1] TRUE
```

The PDF calculated by R and by hand yield the same results.

```
qplot(y2, geom = "histogram", xlab = "Y", ylab = "Frequency") + ggtitle("Distribution of Y based on X ir
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Distribution of Y based on X in U[0,1]
Frequency
50
40
30
20
10
0
0.220
0.225
0.230
0.235
Y