213 HW1

Zixi Zhang

2022-10-09

Use Inverse Transform method to make n geometric, poisson and binomial random variable.

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

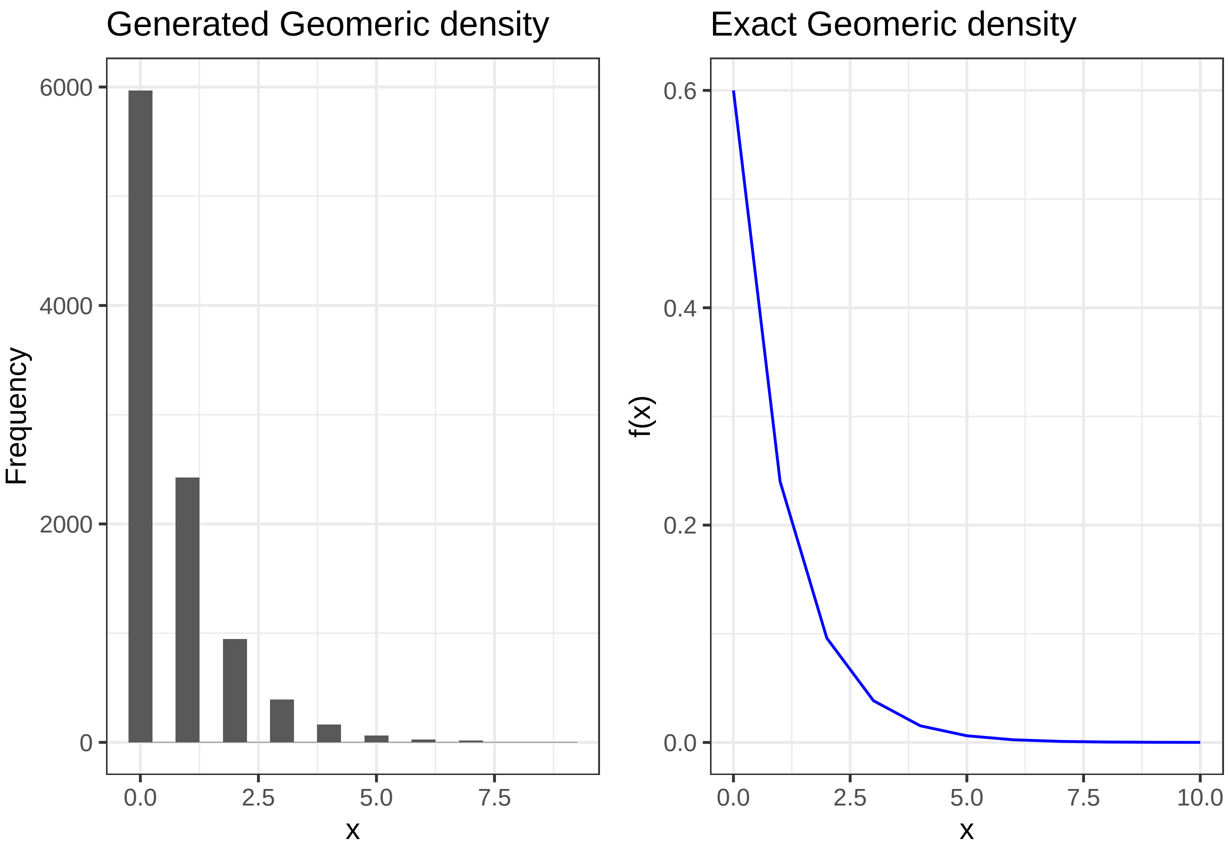
```
library(grid)
```

1. $Geometric(0.6)$

```
# set parameters
p<- 0.6
q <- 1 - p
# generate 10,000 random uniform variables
set.seed(1)
u <- runif(10000)
```

```
#generate values
generated_vals <- floor( log ( u ) / log (q ))
# get the exact values of the PDF
x <- seq(0, 10, by = 1)
theoretical_pdf <- p*(1-p)^x
```

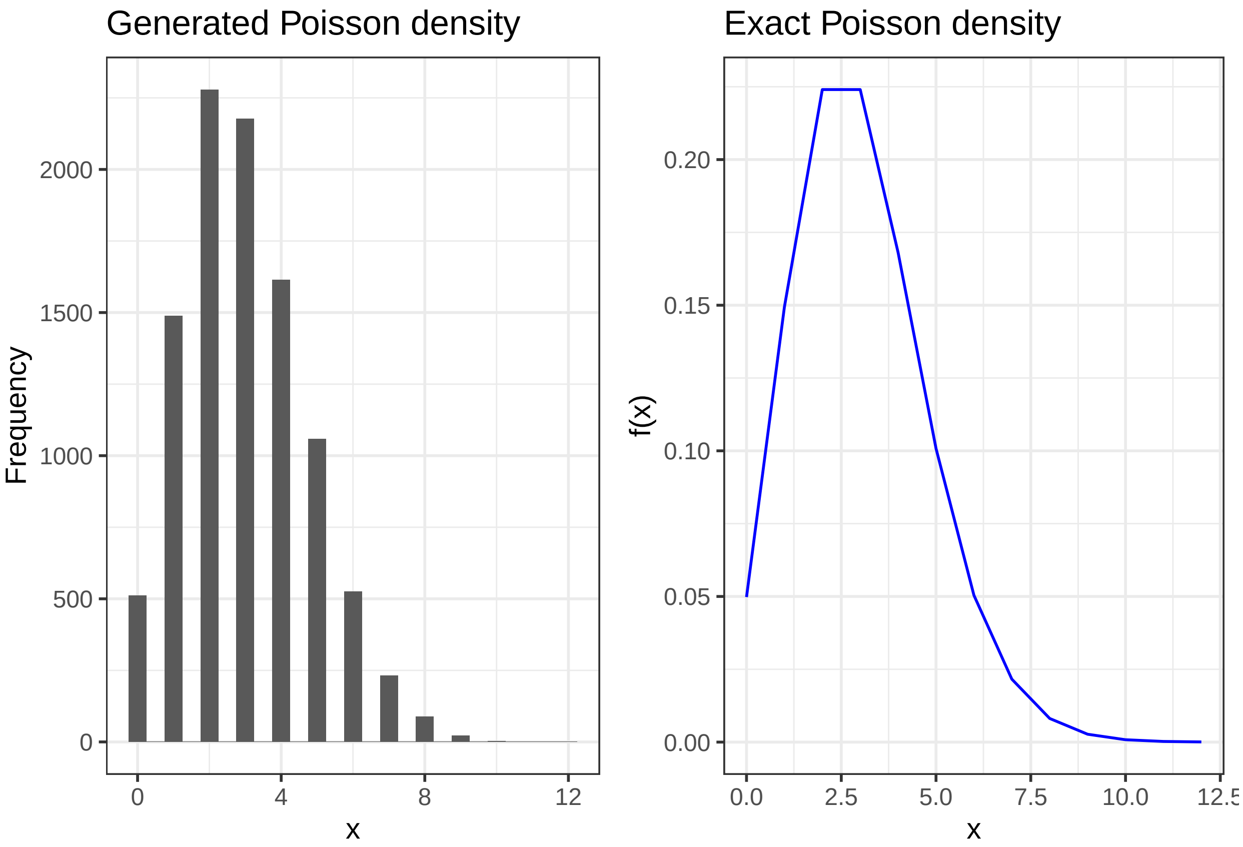
```
# create list to hold our two plots
plts <- list()

# construct histogram of generated values
plts[[1]] <- tibble(generated_vals = generated_vals) %>%
    ggplot() +
        geom_histogram(aes(x = generated_vals),binwidth = 0.5 ) +
        theme_bw() +
        labs(x = "x", y = "Frequency", title = "Generated Geomeric density")

# construct line chart of exact pdf
plts[[2]] <- tibble(x = x, theoretical_pdf = theoretical_pdf) %>%
    ggplot() +
        geom_line(aes(x = x, y = theoretical_pdf), colour = "blue") +
        theme_bw() +
        labs(x = "x", y = "f(x)", title = "Exact Geomeric density")

#arrange two plots
grid.arrange(grobs = plts, nrow = 1)
```



2. $Poisson(3)$

```
# create a poisson generate function
poissonfunc <-  function(n, lambda){
  X <- rep(0, n)
  for(m in 1:n){
    u = runif(1)
    i = 0
    F = exp(-lambda)
    while(u >= F){
      i = i+1
      F = F + dpois(i,lambda)
    }
  X[m] = i
  }
X
}

#generate values
set.seed(1)
generated_vals1 <- poissonfunc(n = 10000,lambda = 3)

# get the exact values of the PDF
x <- seq(0, 12, by = 1)
theoreticalpoi_pdf <- 3^x*exp(-3)/gamma(x+1)
```

```
# create list to hold our two plots
plts <- list()

# construct histogram of generated values
plts[[1]] <- tibble(generated_vals1 = generated_vals1) %>%
    ggplot() +
        geom_histogram(aes(x = generated_vals1),binwidth = 0.5 ) +
        theme_bw() +
        labs(x = "x", y = "Frequency", title = "Generated Poisson density")

# construct line chart of exact pdf
plts[[2]] <- tibble(x = x, theoreticalpoi_pdf = theoreticalpoi_pdf) %>%
    ggplot() +
        geom_line(aes(x = x, y = theoreticalpoi_pdf), colour = "blue") +
        theme_bw() +
        labs(x = "x", y = "f(x)", title = "Exact Poisson density")

#arrange two plots
grid.arrange(grobs = plts, nrow = 1)
```

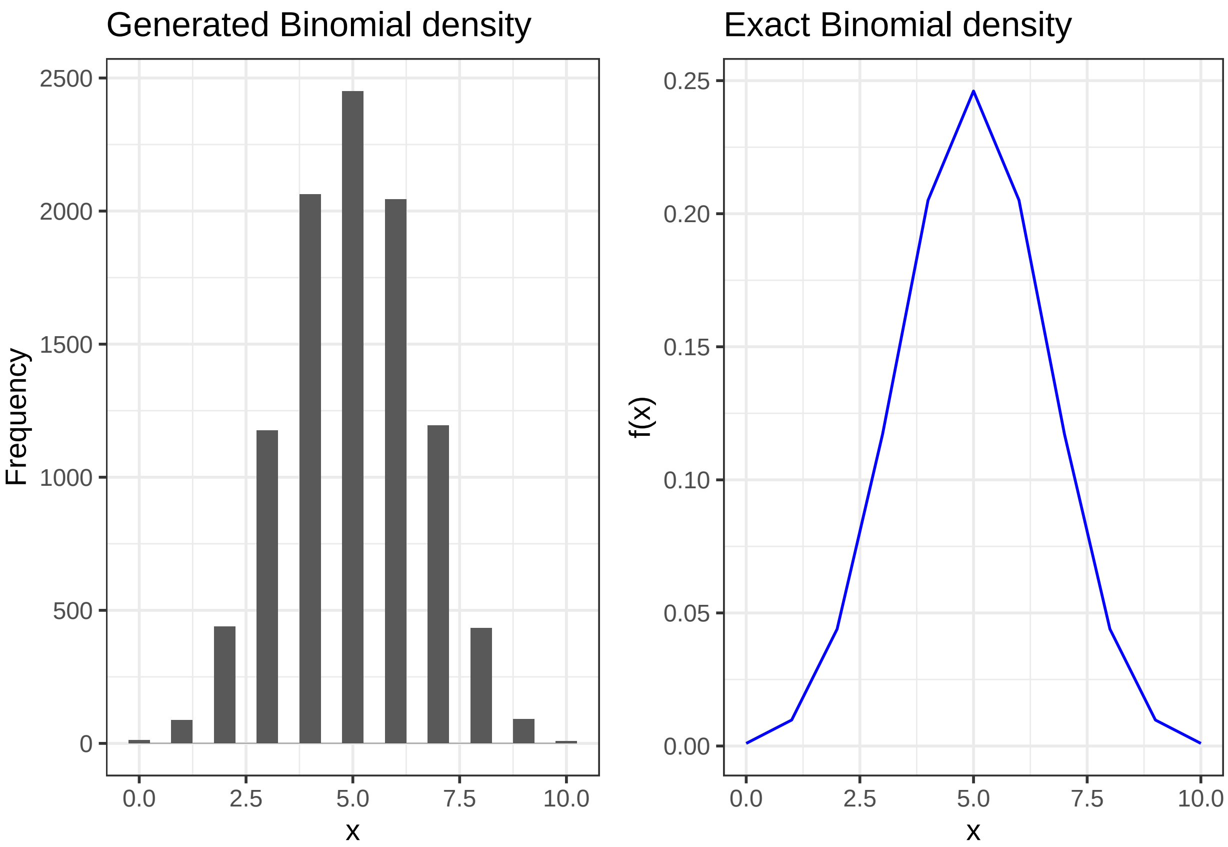


3. $Binomial(10, 0.5)$

```
# create a binomial generate function
binfunc <- function(num, n, p){
  X <- rep(0, num)
  for(m in 1:num){
    u = runif(1)
    i = 0
    c = p/(1-p)
    pr = (1-p)^n
    F = pr
    while(u >= F){
      pr = (c*(n-i)/(i+1))*pr
      F = F + pr
      i = i+1
    }
  X[m] = i
  }
X
}

#generate values
generated_vals2 <- binfunc(num=10000,n=10,p=0.5)

# set parameters
n <- 10
```

```r
p <- 0.5
q <- 1 - p
x <- seq(0, 10, by = 1)
# get the exact values of the PDF
theoreticalbin_pdf <- (gamma(n+1)/(gamma(x+1)*gamma(n-x+1)))*p^x*q^(10-x)
```

```r
# create list to hold our two plots
plts <- list()

# construct histogram of generated values
plts[[1]] <- tibble(generated_vals2 = generated_vals2) %>%
    ggplot() +
        geom_histogram(aes(x = generated_vals2),binwidth = 0.5 ) +
        theme_bw() +
        labs(x = "x", y = "Frequency", title = "Generated Binomial density")

# construct line chart of exact pdf
plts[[2]] <- tibble(x = x, theoreticalbin_pdf = theoreticalbin_pdf) %>%
    ggplot() +
        geom_line(aes(x = x, y = theoreticalbin_pdf), colour = "blue") +
        theme_bw() +
        labs(x = "x", y = "f(x)", title = "Exact Binomial density")


#arrange two plots
grid.arrange(grobs = plts, nrow = 1)
```


Generated Binomial density
Frequency
x
Exact Binomial density
f(x)
x