

# Biostat 213 Homework 2

Zixi Zhang

2022-10-15

```
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)
```

## Assignment

Use the accept-reject algorithm to simulate a Gamma distribution

Make sure you set your seed so you can reproduce your results.

## Gamma Distribution

Probability Density Function:

for  $\alpha, \beta > 0$  and  $x > 0$ :

$$f(x) = \frac{\beta^\alpha x^{\alpha-1} \exp(-\beta x)}{\Gamma(\alpha)}$$

where  $\Gamma(\alpha)$  is the gamma function:

$$\Gamma(x) = \begin{cases} (x-1)!, & \text{if } x \text{ is a positive integer} \\ \int_0^\infty s^{x-1} \exp(-s) ds, & \text{for } x > 0 \end{cases}$$

## Accept-Reject Method to Simulate Gamma Random Variables

Harder: Since it is fairly simple to simulate a  $\text{Gamma}(\alpha, \beta)$  density when  $\alpha$  is an integer. It would be useful to use this with the accept-reject method to simulate a more general Gamma distribution; for example, when  $\alpha$  is not an integer.

#Hint: To do this, we need to figure out if a ratio of two Gamma probability distributions have an upper bound.

```
rgamma.ar <- function(n, shape, scale = 1) {  
  s <- shape  
  s.int <- floor(s)  
  b <- s / s.int  
  M <-  
    gamma(s.int) / gamma(s) * b ** s.int * (s * exp(-1)) ** (s - s.int)  
  f <- function(y)  
    dgamma(y, shape = s)  
  Mg <- function(y)  
    M * dgamma(y, shape = s.int, rate = 1 / b)  
  acpt <- 0  
  total <- 0  
  X <- numeric(n)  
  while (acpt < n) {  
    total <- total + 1  
    Y <- sum(-b * log(runif(s.int)))  
    if (runif(1) <= f(Y) / Mg(Y)) {  
      acpt <- acpt + 1  
      X[acpt] <- Y  
    }  
  }  
  return(list(  
    X = scale * X,  
    theoretical_rate = 1 / M,  
    observed_rate = acpt / total  
  ))  
}
```

```
#set shape = 5.5, rate = 1  
set.seed(2016)  
o <- rgamma.ar(1000, shape = 5.5)  
#get acceptance probability  
o[-1]
```

```
## $theoretical_rate  
## [1] 0.9520242  
##  
## $observed_rate  
## [1] 0.9496676
```

```
# create list to hold our two plots  
plts <- list()  
  
# construct histogram of generated values  
plts[[1]] <- tibble(sim = o$X) %>%
```

```

ggplot() + geom_density(aes(x = sim)) + labs(x = "x", y = "Frequency", title =
      "Generated Gamma density") +
  xlim(0, 18) + ylim( c(0, 0.2)) + theme_bw()

# get the exact values of the PDF
gam <- dgamma(seq(0, 18, 0.1), 5.5, rate = 1)
# construct line chart of exact pdf
plts[[2]] <- tibble(x = seq(0, 18, 0.1), pdf = gam) %>%
  ggplot() + geom_line(aes(x = x, y = pdf)) + labs(x = "x", y = "f(x)", title =
      "Exact Gamma density") +
  xlim(0, 18) + ylim(c(0, 0.2)) + theme_bw()

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

```

