Problem 2

HW1

Hanyuan Chi(chixx105), Zhi Shen(shenx704)

April 09, 2017

```
suppressPackageStartupMessages({
  library(ggplot2)
  library(dplyr)
})
```

Accept-Reject Sampling

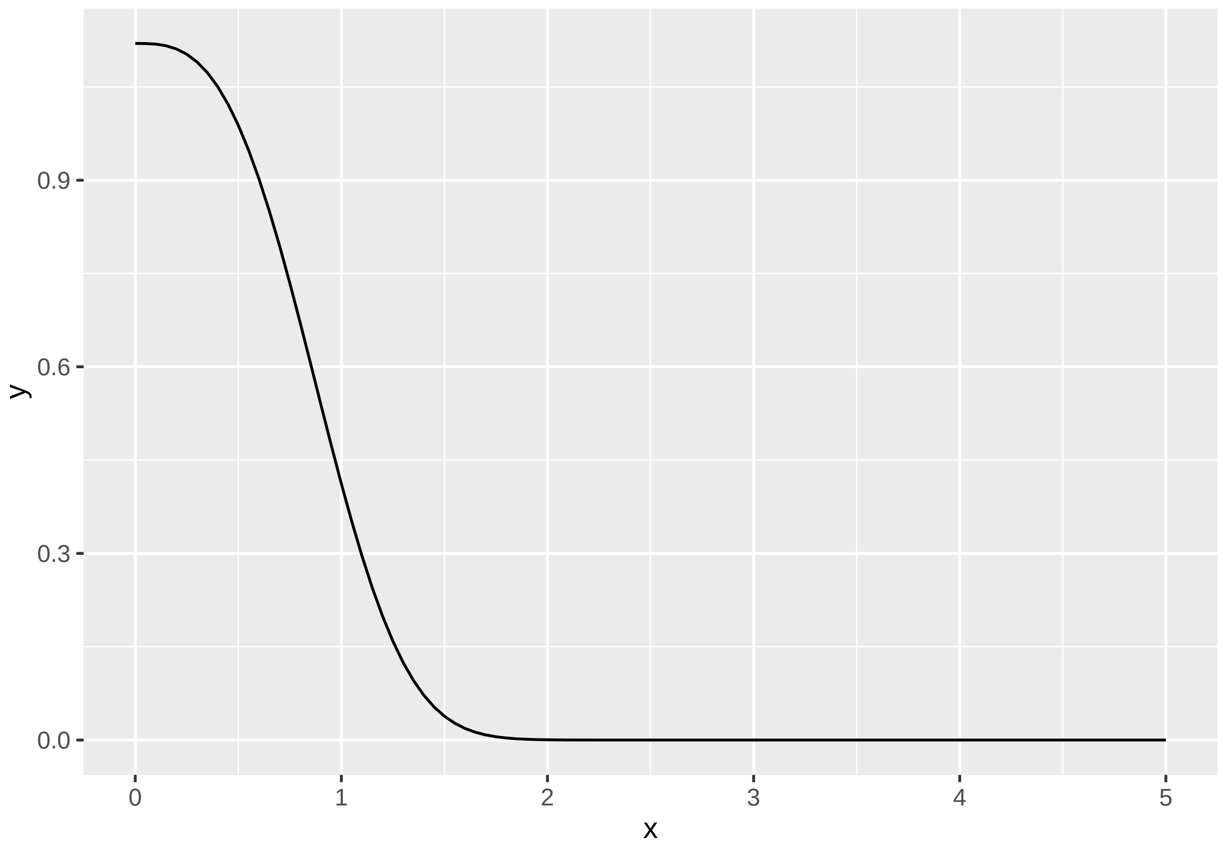
Question 1

Please simulate 1000 random samples from the distribution with the following PDF

$$f(x) = \begin{cases} \frac{1}{\Gamma\left(\frac{4}{3}\right)} e^{-x^3}, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

- Hints:
  - $\Gamma\left(\frac{4}{3}\right)$ can be computed as `gamma(4/3)` in R. It is a constant so you only need to compute it once.
  - Think about the good proposal distribution here
  - Please keep in mind that Question 2 will ask you to demonstrate graphically the validity of your sampling procedure.
    - I suggest the "modified" procedure from the slides.
  - This problem can be solved in many different ways. If you are up to today's standards of R programming:
    - Create a separate function for PDF
- Output:
  - Please create a temporary *data.frame* `df1` that contains both the rejected and accepted samples. We will use this data.frame for plotting in Question 2.
  - Please create a *data.frame* `result_df` that contains the resulting sample from the distribution $f(x)$. Please put this sample into column `result_df$X`.

```
N <- 1000L
set.seed(123456) # PLEASE DO NOT CHANGE THE SEED

my_pdf <- function(x) {
  1/gamma(4/3)*exp(-x^3)
}

ggplot(data.frame(x=c(0,5)),aes(x)) +
  stat_function(fun = my_pdf)
```



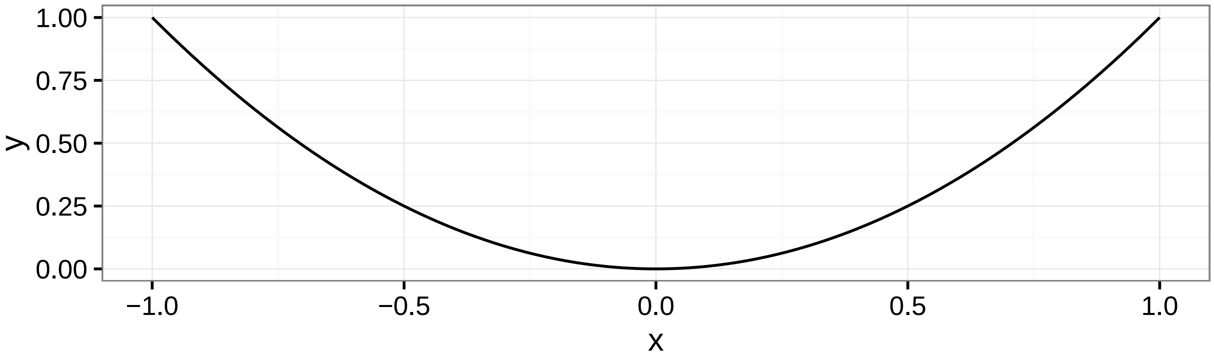
```
C <- 1/gamma(4/3)*3/2

df1 <- data.frame(X = rexp(N, rate = 1)) %>%
  mutate(Y = runif(N, min = 0, max = C*dexp(X))) %>%
  mutate(accept = ifelse(Y <= my_pdf(X), 1L, 0L))

result_df <- df1 %>%
  filter(accept == 1L) %>%
  select(X)
```

Question 2

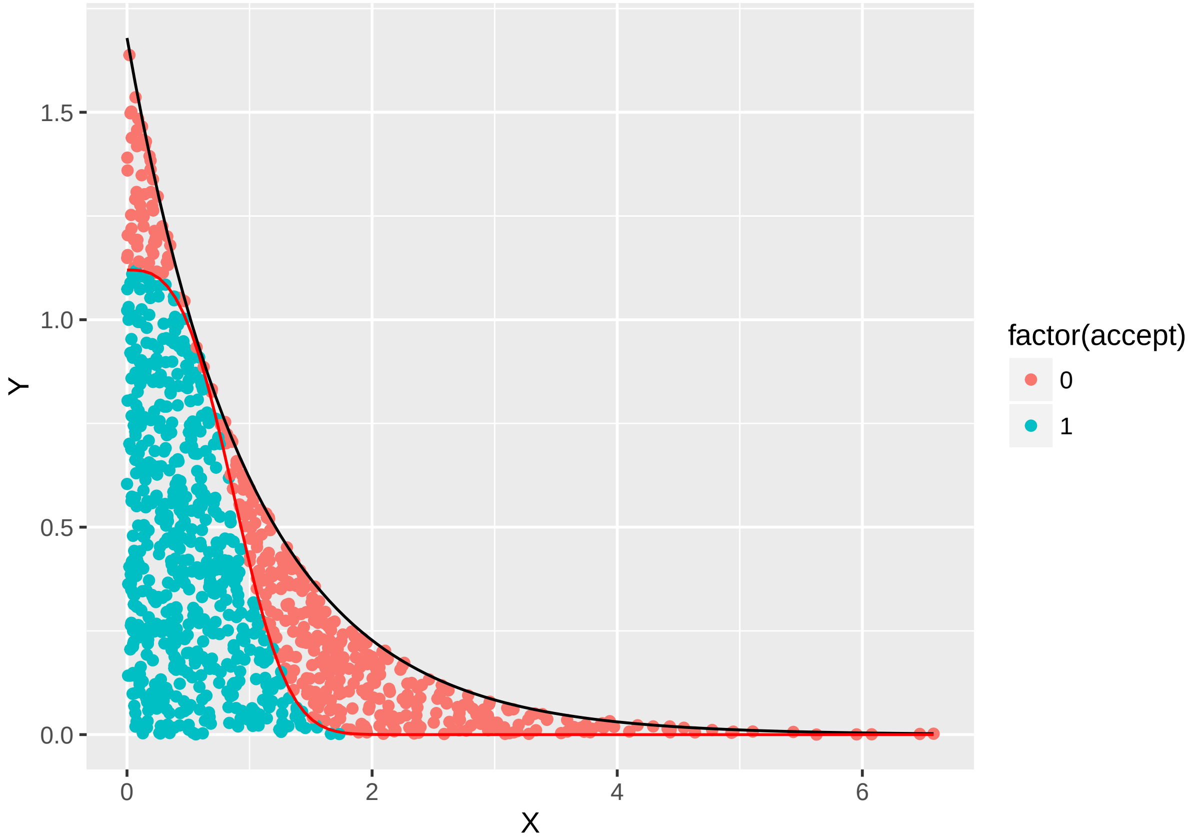
Please demonstrate graphically the validity of your sampling procedure

- Output:
  - please use ggplot2 and save your plot into variable `p2`
  - please also display the plot
- Hints:
  - I would plot the accepted points with different color than rejected points
  - you can use `geom_point` layer to plot samples
  - you can use `stat_function(fun=your_func)` layer to plot functions in ggplot. Here is one example how it can be done:

```
ggplot(data.frame(x=c(-1, 1)), aes(x)) +
  stat_function(fun = function(x) (x^2) )  +
  theme_bw()
```



```
# Please write your code below

p2 <- ggplot(data = df1) +
  geom_point(aes(X,Y,color = factor(accept))) +
  stat_function(aes(X),fun = function(x) (C*dexp(x))) +
  stat_function(aes(X),fun = my_pdf, color = 'red')
p2
```



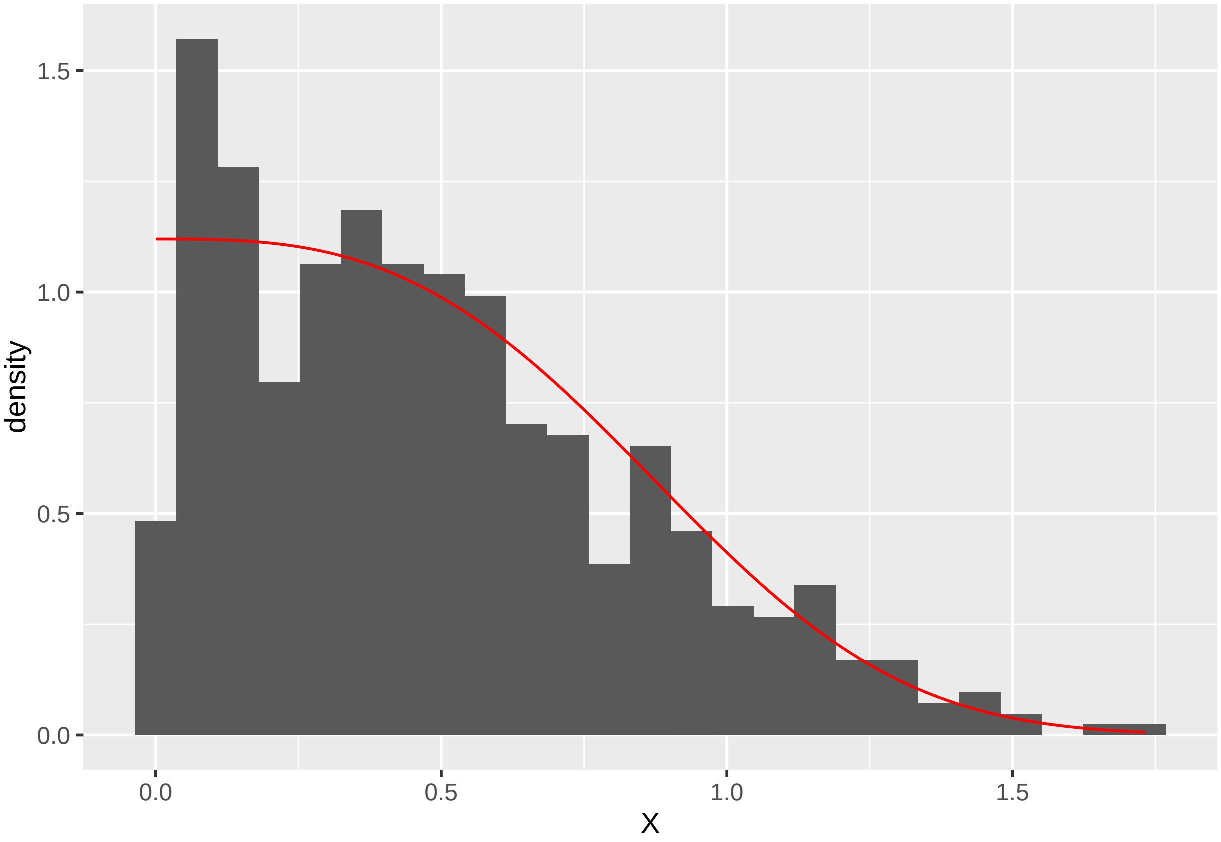
Question 3

Please demonstrate graphically the validity of your results by comparing a histogram versus true target PDF

- Output:

  - please use ggplot2 and save your plot into variable `p3`

```
# Please write your code below

p3 <- ggplot(data = result_df) +
  geom_histogram(aes(x=X, y=..density..),bins = 25) +
  stat_function(aes(X), fun = my_pdf, color = 'red')

p3
```


density
X