

Data 556 HW 2

Naomi Liftman

2025-10-07

```
#loading the package that includes rv distributions and ggplot  
library(stats)  
library(ggplot2)
```

2a. Use simulations to numerically estimate mean & variance of A.

```
#setting a seed so its reproducible  
set.seed(99)  
  
nsims=10000  
records = rep(0,nsims)  
  
for(i in 1:nsims){  
  radius <- runif(1, min = 0, max = 1)  
  records[i] <- ((radius^2)*pi)  
}  
  
#expectation of the area (the area is the records)  
e_a = sum(records)/nsims #1.048  
var_a = var(records) #.8768
```

3a. Use simulations to gain understanding about the distribution of R. Numerically estimate the expected value of R and $1/R$.

```
#set seed  
set.seed(100)  
  
nsims=10000  
records_x = rep(0,nsims)  
records_y = rep(0,nsims)  
records_ratio = rep(0,nsims)  
one_r = rep(0,nsims)  
  
for(i in 1:nsims){  
  x <- runif(1, min = 0, max = 1)  
  y = 1-x
```

```

records_x[i] <- x
records_y[i] <- y
records_ratio[i] <- (x/y)
one_r[i] <- (y/x)
}

e_r = sum(records_ratio)/10000 #9.0554
e_r_one = sum(one_r)/10000 #8.438

```

4a. Let U_1, \dots, U_n be iid $\text{Unif}(0,1)$ and $x = \max(\text{iids})$. Use R to numerically estimate $E(x)$.

```

set.seed(100)

nsims = 10000
xrecords = rep(0, nsims)

for(i in 1:nsims){
  x <- runif(1, min = 0, max = 1)
  xrecords[i] <- x
}

e_r = sum(xrecords)/10000 # .50029

```

5b. Use R to numerically estimate $P(X < Y)$ for $X \sim N(0,1)$, $Y \sim N(1,5)$ w X and Y being independent.

```

set.seed(100)

nsims = 10000
records_x_y = rep(0, nsims)

for(i in 1:nsims){
  x <- rnorm(1, mean = 0, sd = 1)
  y <- rnorm(1, mean = 1, sd = 5)
  records_x_y[i] <- (x < y)
}

exy = sum(records_x_y)/nsims # .5759

```

6b. Use R to calculate the Monte Carlo estimates of mean + SD of $x-y$.

```

set.seed(100)

nsims=10000
records_x = rep(0,nsims)
records_y = rep(0,nsims)
records_ratio = rep(0, nsims)

for(i in 1:nsims){
  x <- rnorm(1, mean = 69.1, sd = 2.9)
  y <- rnorm(1, mean = 63.7, sd = 2.7)
  records_x[i] <- x
  records_y[i] <- y
  records_ratio[i] <- (x - y)
}

e_r = sum(records_ratio)/10000 # 5.396
sd <- sd(records_ratio) #3.942

```

6c. What is the probability that a randomly sampled man is taller than a randomly sampled woman?

```
pnorm(63.7, mean = 5.4, .2, lower.tail = FALSE)
```

```
## [1] 0
```

7b. For $y = 0$ make a plot for $P(\text{theta given } y)$ for each theta within $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Make a plot with horizontal axis the 11 values of theta and verticle the $P(\text{theta given } y)$.

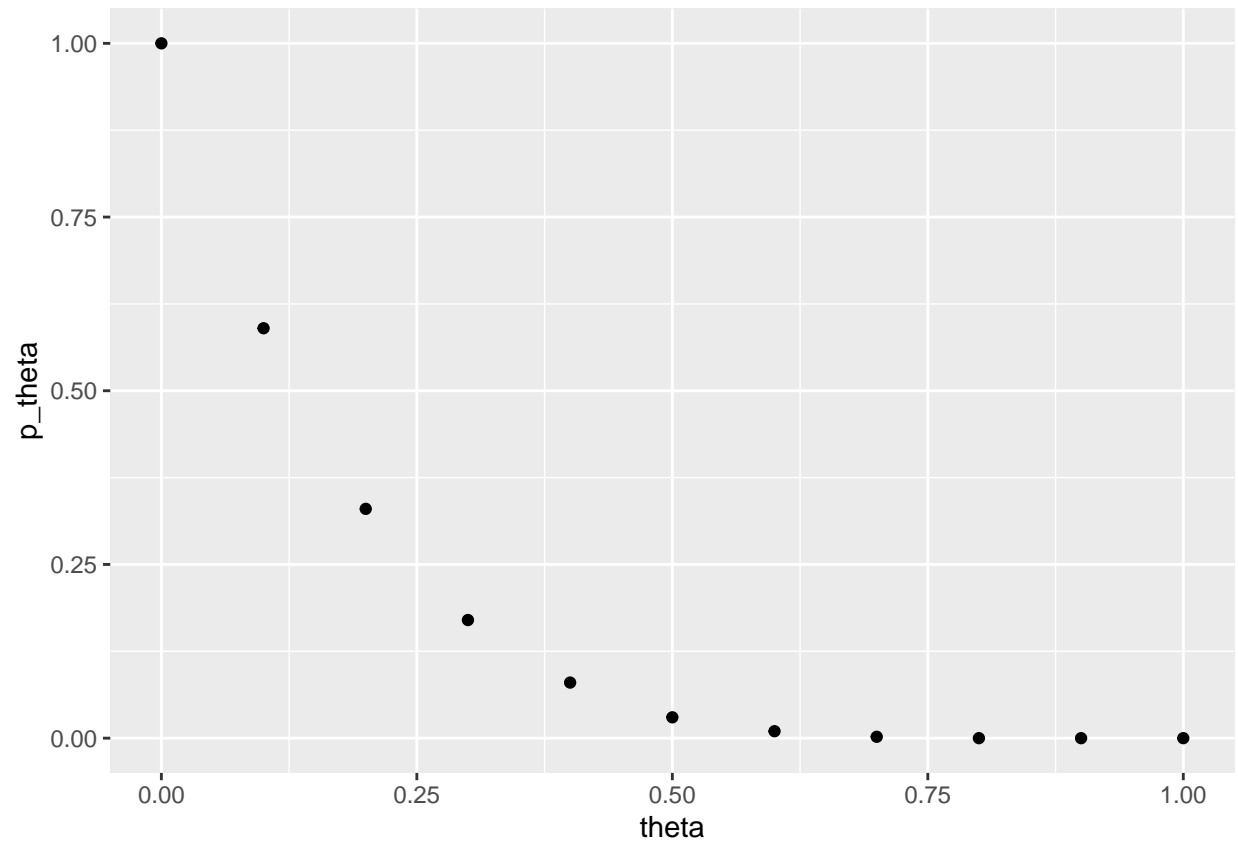
```

#making a df
theta <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta <- c(1, .59, .33, .17, .08, .03, .01, .002, 0, 0, 0)

df_theta <- data.frame(theta, p_theta)

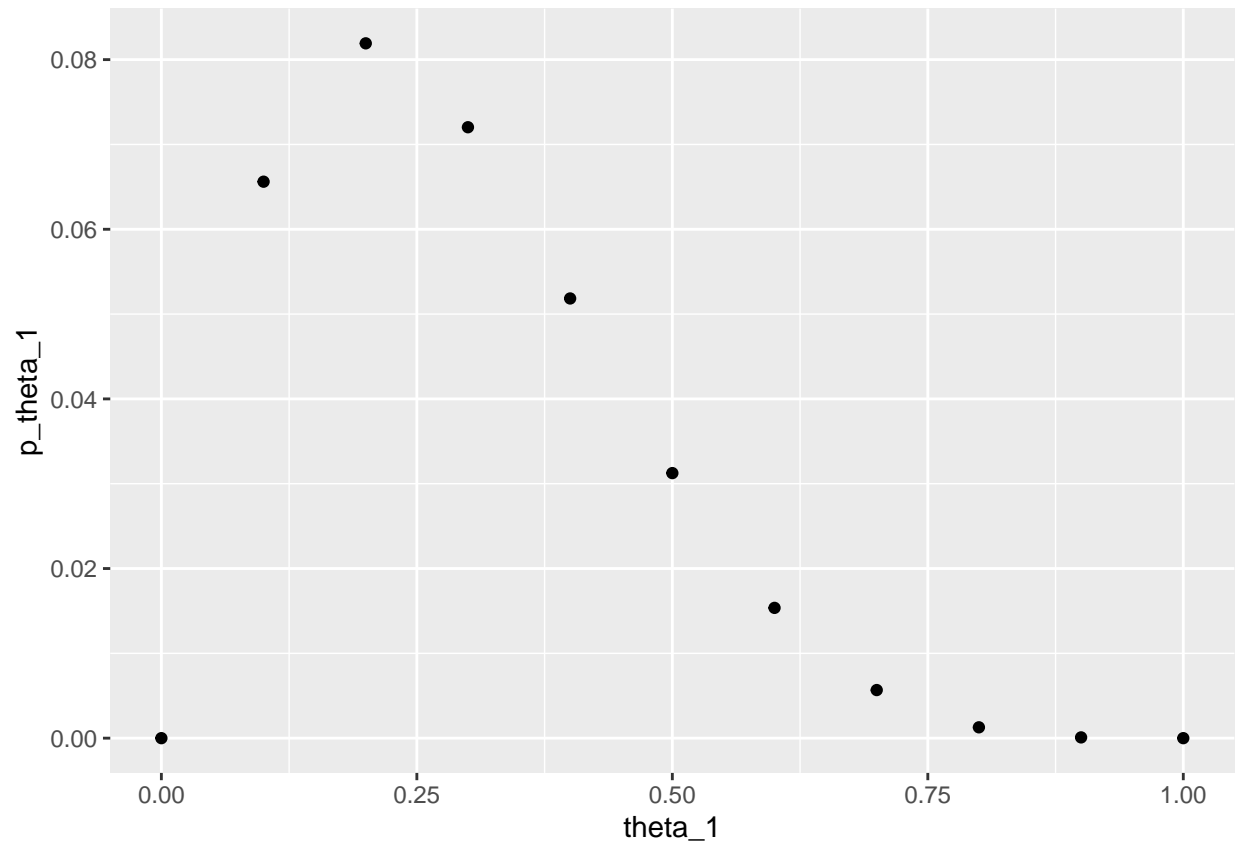
ggplot(df_theta, aes(x= theta, y = p_theta)) + geom_point()

```

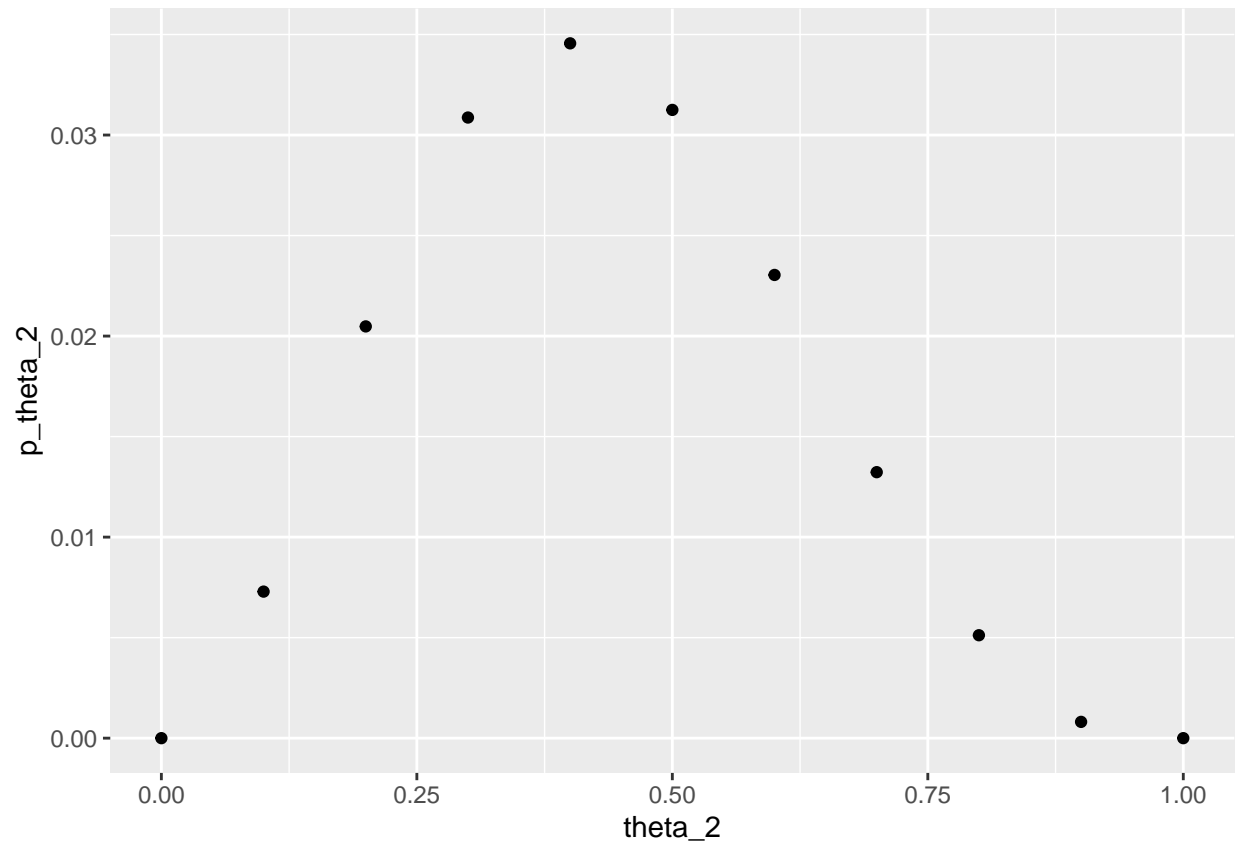


7c. Repeat (b) for each y within {1,2,3,4,5}.

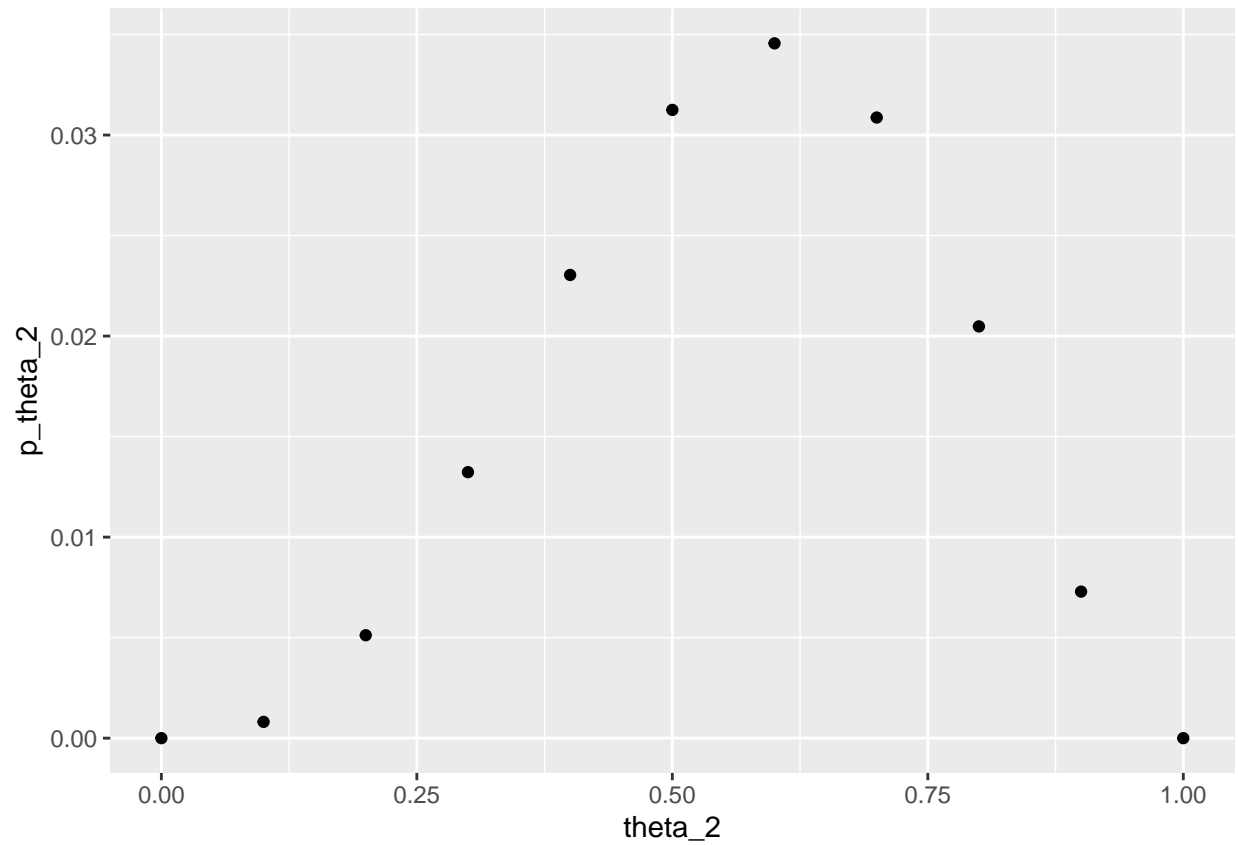
```
#making a df for 1 and graph
theta_1 <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta_1 <- c((0^1)*(1^4), (.1^1)*(.9^4), (.2^1)*(.8^4), (.3^1)*(.7^4), (.4^1)*(.6^4), (.5^1)*(.5^4),
df_theta_1 <- data.frame(theta, p_theta)
ggplot(df_theta_1, aes(x= theta_1, y = p_theta_1)) + geom_point()
```



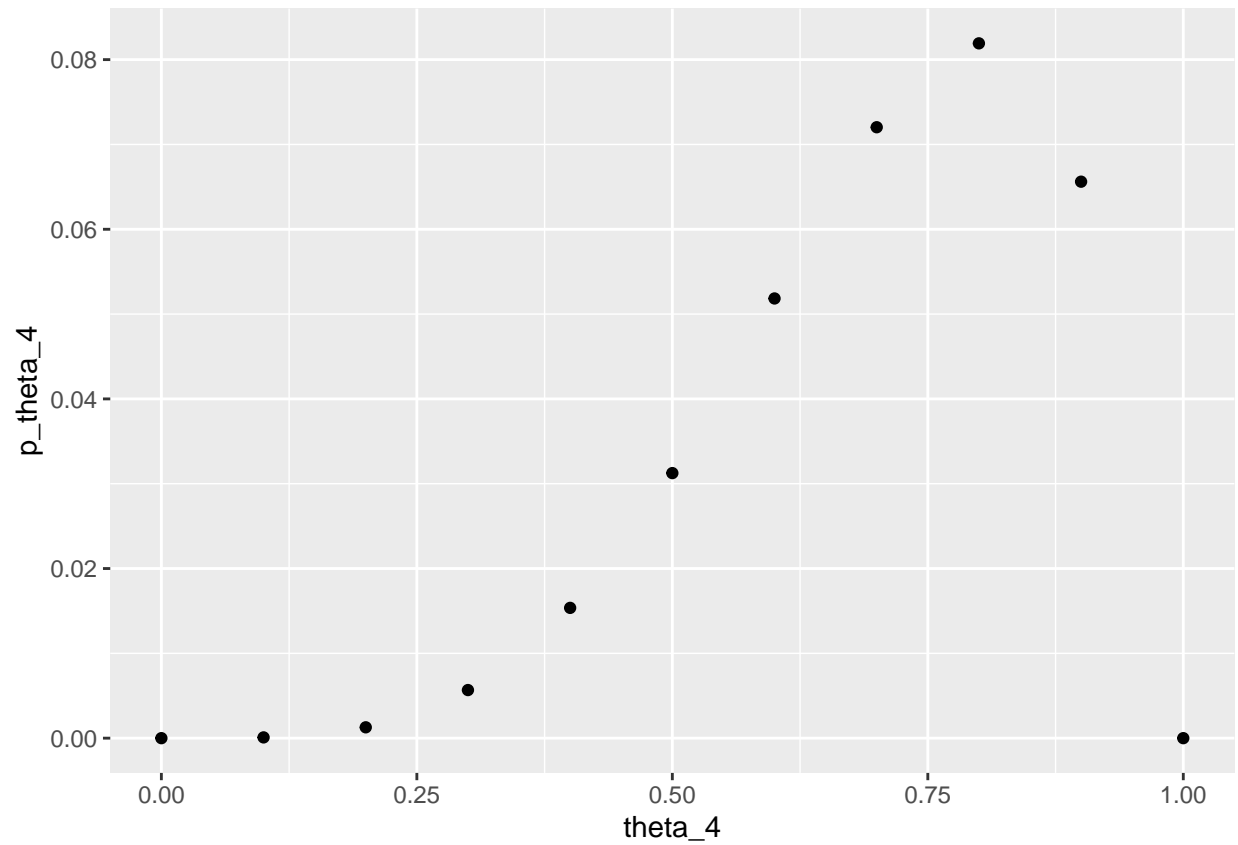
```
#df and graph for 2
theta_2 <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta_2 <- c((0^2)*(1^3), (.1^2)*(.9^3), (.2^2)*(.8^3), (.3^2)*(.7^3), (.4^2)*(.6^3), (.5^2)*(.5^3),
df_theta_2 <- data.frame(theta, p_theta)
ggplot(df_theta_2, aes(x= theta_2, y = p_theta_2)) + geom_point()
```



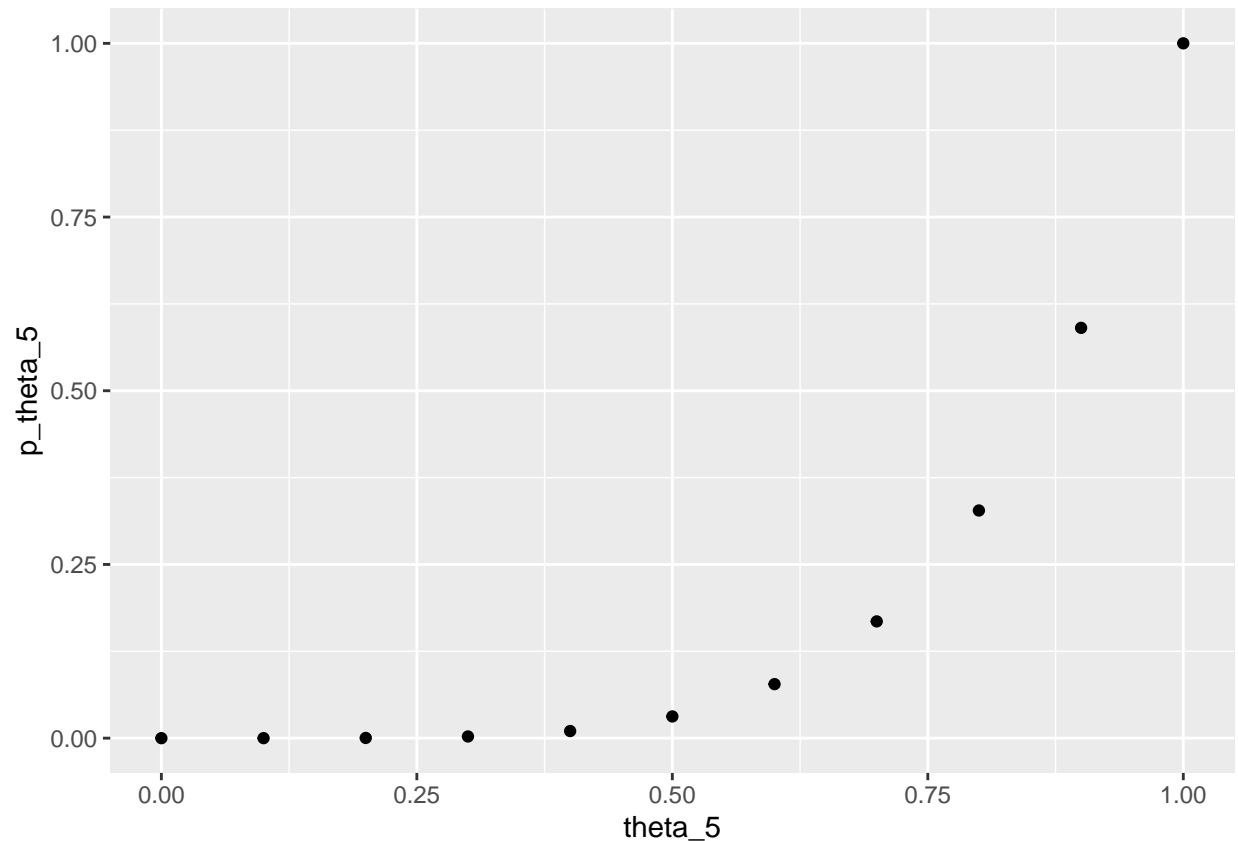
```
#df and graph for 3
theta_2 <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta_2 <- c((0^3)*(1^2), (.1^3)*(.9^2), (.2^3)*(.8^2), (.3^3)*(.7^2), (.4^3)*(.6^2), (.5^3)*(.5^2),
              (.6^3)*(.4^2), (.7^3)*(.3^2), (.8^3)*(.2^2), (.9^3)*(.1^2), 1^3*0^2)
df_theta_2 <- data.frame(theta_2, p_theta_2)
ggplot(df_theta_2, aes(x= theta_2, y = p_theta_2)) + geom_point()
```



```
#df and graph for 4
theta_4 <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta_4 <- c((0^4)*(1^1), (.1^4)*(.9^1), (.2^4)*(.8^1), (.3^4)*(.7^1), (.4^4)*(.6^1), (.5^4)*(.5^1),
               (.6^4)*(.4^1), (.7^4)*(.3^1), (.8^4)*(.2^1), (.9^4)*(.1^1), (1^4)*(0^1))
df_theta_4 <- data.frame(theta_4, p_theta_4)
ggplot(df_theta_4, aes(x= theta_4, y = p_theta_4)) + geom_point()
```



```
#df and graph for 5
theta_5 <- c(0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)
p_theta_5 <- c((0^5)*(1^0), (.1^5)*(.9^0), (.2^5)*(.8^0), (.3^5)*(.7^0), (.4^5)*(.6^0), (.5^5)*(.5^0),
df_theta_5 <- data.frame(theta, p_theta)
ggplot(df_theta_5, aes(x= theta_5, y = p_theta_5)) + geom_point()
```

8b. Implement your sampling algorithm in R and use your code to produce a Monte Carlo estimate of $P(X \text{ lives within } (2,3))$ where X is a r.v. with a logistic distribution.

```
#setting a seed so its reproducible
set.seed(99)

nsims=10000
records = rep(0,nsims)
records
```

```
##      [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##      [37] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##      [73] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [109] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [145] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [181] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [217] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [253] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [289] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [325] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##     [361] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

##	[397]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[433]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[469]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[505]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[541]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[577]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[613]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[649]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[685]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[721]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[757]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[793]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[829]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[865]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[901]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[937]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[973]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1009]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1045]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1081]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1117]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1153]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1189]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1225]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1261]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
##	[1297]	0	0	0																					

[illegible]

[illegible]

[illegible]


```
xmake_change <- ((1/x) + 1)
xlogistic <- -log(xmake_change, base = exp(1))
records[i] <- (2 < xlogistic & xlogistic < 3)
}
```

The probability is zero because none of ours fall between 2 and three?