

STAT 8010 R Session 2: Probability

Whitney Huang

5/26/2023

Contents

Lab Objective	1
Random Experiment	1
Empirical Probability	2
Logical Operators in R	2
Complement Rule Example	3
The Monty Hall Problem	3
A R function for performing Monty Hall game	3
Conducting a Monte Carlo experiment	3
Rare Disease Example	4
Number of Siblings Example	4
ESP Example	5
TV Example	5
Continuos Random Variable	6
Probability Density Function	6
Estimating Probability via Monte Carlo	7

Lab Objective

- To gain experience with R, a programming language and free software environment for statistical computing and graphics.
- To learn how perform *Monte Carlo* simulation in R.

Random Experiment

Let's roll a dice twice. We will use *sample* command in R to conduct such a (random) experiment. Type *?sample* to learn about this command

- **Description**

sample takes a sample of the specified size from the elements of *x* using either with or without replacement.

- Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

- Arguments

- *x*: either a vector of one or more elements from which to choose.
- *size*: a non-negative integer giving the number of items to choose.
- *replace*: should sampling be with replacement?
- *prob*: a vector of probability weights for obtaining the elements of the vector being sampled.

```
sample(1:6, size = 2, replace = T)
```

```
## [1] 4 6
```

Empirical Probability

```
out1 <- sample(0:1, size = 1000, replace = T, prob = c(0.5, 0.5))
out2 <- sample(0:1, size = 1000, replace = T, prob = c(0.3, 0.7))
```

Logical Operators in R

```
omega <- letters
A <- sample(omega, 5, replace = F)
A
```

```
## [1] "d" "e" "o" "p" "b"
```

```
B <- sample(omega, 4, replace = F)
B
```

```
## [1] "y" "g" "i" "v"
```

```
union(A, B)
```

```
## [1] "d" "e" "o" "p" "b" "y" "g" "i" "v"
```

```
intersect(A, B)
```

```
## character(0)
```

Complement Rule Example

Suppose we rolled a fair, six-sided die 10 times. Let T be the event that we roll at least 1 three. If one were to calculate T you would need to find the probability of 1 three, 2 threes, \dots , and 10 threes and add them all up. However, you can use the complement rule to calculate $\mathbb{P}(T)$

```
M <- 1000
out <- replicate(M, sample(1:6, 10, replace = T))
count_3 <- apply(out, 2, function(x) sum(x == 3))
sum(count_3 >= 1) / M
```

```
## [1] 0.831
```

```
1 - pbinom(0, 10, 1/6)
```

```
## [1] 0.8384944
```

```
pbinom(0, 10, 1/6, lower.tail = F)
```

```
## [1] 0.8384944
```

The Monty Hall Problem

A R function for performing Monty Hall game

```
MontyHall <- function() {
  # Assign the prize
  prize <- sample(1:3, 1)
  # Pick a door
  choice <- sample(1:3, 1)
  # Monty picks a door
  monty <- sample(c(1:3)[-c(choice, prize)], 1)
  return(ifelse(prize != choice, "Switch", "Stick"))
}
```

Conducting a Monte Carlo experiment

```
M = 100
try <- replicate(M, MontyHall())
table(try) / M
```

```
## try
## Stick Switch
## 0.31 0.69
```

Rare Disease Example

Let us assume that a specific disease is only present in 5 out of every 1,000 people. Suppose that the test for the disease is accurate 99% of the time a person has the disease and 95% of the time that a person lacks the disease. What is the probability that the person has the disease given that they tested positive?

```
set.seed(1234)
n = 10000; p = 0.005
sub <- 1:n
d <- rbinom(n, 1, prob = p)
TP <- sum(rbinom(length(sub[d]), 1, prob = .99))
TN <- sum(rbinom(n - length(sub[d]), 1, prob = .95))
FP <- (n - sum(d)) - TN
FN <- sum(d) - TP
mat <- matrix(c(TP, FP, FN, TN), 2)
colnames(mat) <- c("Test positive", "Test negative")
rownames(mat) <- c("Disease", "No disease")
mat
```

```
##           Test positive Test negative
## Disease           40           0
## No disease        479          9481
```

Number of Siblings Example

```
x <- 0:4
p <- c(.2, .425, .275, .075, .025)
sum(x * p)
```

```
## [1] 1.3
```

```
weighted.mean(x, p)
```

```
## [1] 1.3
```

```
sum((x - sum(x * p))^2 * p)
```

```
## [1] 0.91
```

```
sum(x^2 * p) - (sum(x * p))^2
```

```
## [1] 0.91
```

ESP Example

To test for Extrasensory perception (ESP), we have 4 cards. They will be shuffled and one randomly selected each time, and you are to guess which card is selected. This is repeated 10 times. Suppose you do not have ESP. Let R be the number of times you guess a card correctly. What are the distribution and parameter(s) of R ? What is the expected value of R ? Furthermore, suppose that you get certified as having ESP if you score at least an 8 on the test. What is the probability that you get certified as having ESP?

Since $R \sim \text{binom}(n = 10, p = 0.25)$. We can use the built-in functions *d/p/q/r-binom* to carry out the calculations.

*Usage

- `dbinom(x, size, prob, log = FALSE)`
- `pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)`
- `qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)`
- `rbinom(n, size, prob)`

*Arguments

- `x, q`: vector of quantiles.
- `p`: vector of probabilities.
- `size`: number of trials.
- `prob`: probability of success on each trial.
- `log, log.p`: logical; if TRUE, probabilities `p` are given as $\log(p)$.
- `lower.tail`: logical; if TRUE (default), probabilities are $\mathbb{P}[X \leq x]$, otherwise, $\mathbb{P}[X > x]$.

```
sum(dbinom(8:10, 10, 0.25))
```

```
## [1] 0.000415802
```

```
pbinom(7, 10, 0.25, lower.tail = F)
```

```
## [1] 0.000415802
```

```
1 - pbinom(7, 10, 0.25)
```

```
## [1] 0.000415802
```

```
M = 100000  
sim <- rbinom(M, 10, prob = 0.25)  
length(which(sim >= 8)) / M
```

```
## [1] 0.00053
```

TV Example

There are 100 identical looking 52" TVs at Best Buy in Anderson, SC. Let 10 of them be defective. Suppose we want to buy 8 of the aforementioned TVs (at random). What is the probability that we don't get any defective TVs?

```
dhyper(0, 10, 90, 8)
```

```
## [1] 0.4165533
```

```
trial <- replicate(100000, sample(1:100, 8, replace = F))  
min <- apply(trial, 2, min)  
sum(min > 10) / 100000
```

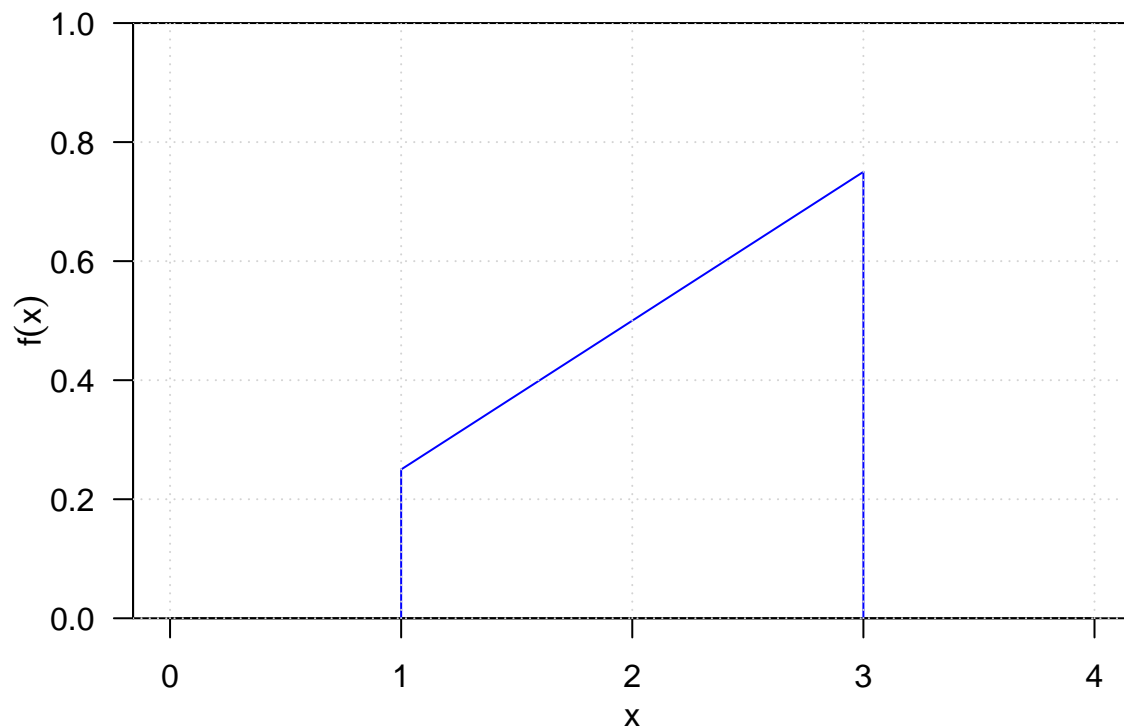
```
## [1] 0.41603
```

Continuous Random Variable

Let $f_X(x) = .25x$ for $1 \leq x \leq 3$ and 0 otherwise

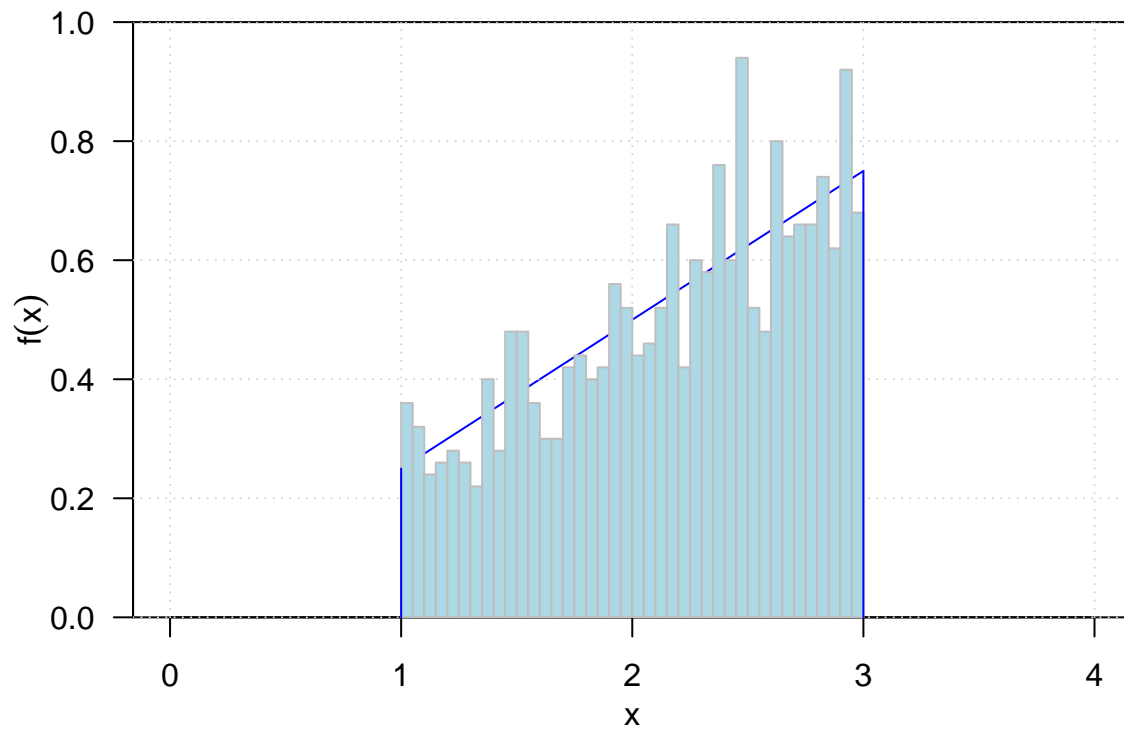
Probability Density Function

```
xg <- seq(1, 3, len = 100)  
fg <- xg / 4  
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2, 1, 0))  
plot(xg, fg, type = "l", xlim = c(0, 4), ylim = c(0, 1),  
      xlab = expression(x), ylab = expression(f(x)), col = "blue",  
      yaxs = "i")  
segments(1, 0, y1 = 1 / 4, col = "blue")  
segments(3, 0, y1 = 3 / 4, col = "blue")  
grid()
```



Estimating Probability via Monte Carlo

```
M = 1000
x <- sqrt(8 * runif(M) + 1)
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2, 1, 0))
plot(xg, fg, type = "l", xlim = c(0, 4), ylim = c(0, 1),
     xlab = expression(x), ylab = expression(f(x)), col = "blue",
     yaxs = "i")
grid()
hist(x, 40, prob = T, add = T, col = "lightblue", border = "gray")
segments(1, 0, y1 = 1 / 4, col = "blue")
segments(3, 0, y1 = 3 / 4, col = "blue")
```



```
sum(x > 1 & x < 2) / M
```

```
## [1] 0.365
```