Lab 2: Practice Code

Probability Distributions, Statistical Inference, and Ordinary Least Squares

Your name

October 3, 2025

%>% ## Prerequisite

```
# Good practice to remove all objects from the workspace
rm(list = ls())

# Use library() for packages you need, or source() for other R files.
library(tidyverse)

# Setting the seed ensures that we get the same random draw over and over again.
set.seed(2025)

rnorm(5) # Check
```

Probability Distributions

0. Calculate the following operations by hand (... meaning by R)

a)

$$\sum_{i=1}^{5} i =$$

b)

$$\prod_{i=1}^{5} i =$$

c)

$$5! \times 10^{3!} \times e^{4} =$$

```
# a)



# b)


# c)
```

1. Build a Bernoulli distribution using the `sample()` function, where the probability of *success* is 0.7. Run `?sample` if you are unsure how the function works.

```
# Create an imaginary person to flip the coin once for you

sample(x = c(0,1),
       size = 1,
       prob = c(0.3, 0.7))
```

2. How do you know if it is working properly? Conduct simulation with `for loop` to check if the assigned probabilities are matched with the empirics

```
# Specify the number of simulations - more simulations, more correct
sims <- 10000

# Specify the probability
ProbSuccess <- 0.7

# Create an empty vector as "container"
BernResult <- vector(mode = "numeric",
                     length = sims)

# For loop
for (i in 1:sims) {
BernResult[i] <-sample(x = c(0,1),
                       size = 1,
                       prob = c(1-ProbSuccess, ProbSuccess))
}


mean(BernResult)
```

3. Plot the above Bernoulli distribution

```
hist(BernResult)
```

```
BernResult |>
  as_tibble() |>
  ggplot(aes(x=value))+
  geom_histogram()
```

4. Based on the above, generate a binomial distribution, with number of trials equal to 10, without using rbinom()

```r
# Imagine you flip the coin ten times
# Let's test it outside of the loop:

sample(x = c(0,1),
       size = 10,
       replace = T,
       prob = c(1-ProbSuccess,ProbSuccess))
```

```r
# Create number of simulations and an empty vector as container
set.seed(12345)

sims <- 9000

BinoResult <- vector(mode = "numeric", length = sims)

for (i in 1:sims) {

  # Flip the coin ten times
  flips <- sample(x = c(0,1),
                  size = 10,
                  replace = T,
                  prob = c(1-ProbSuccess,ProbSuccess))

  # Store the sum of the number of "success" into the container; repeat 9000
  BinoResult[i] <- sum(flips)
}

BinoResult

 # What are we doing with the following code?
BinoResult7 <- sum(BinoResult == 7)
BinoResult7/sims
```

```r
## EXTRA

# Create number of simulations
sims <- 9000

# Get 9000 binomial draws (each draw = #successes in 10 trials)
# BinoResult <- rbinom(?, ?, ?)

BinoResult7 <- sum(BinoResult == 7)
BinoResult7/sims
```

5. Plot the above binomial distribution

```r
hist(BinoResult)
```

6. Explore the rbinom, dbinom, pbinom functions. What do they do? Answer the following questions:

a) The probability of a coin landing on head is 0.7. If you were to flip the coin 10 times, what is the probability of getting exactly 7 heads?
b) What is the probability of getting 7 heads or less?
c) How do you know (b) is true?

```r
# a) Pr(exactly 7 heads) -> PdF
dbinom(x = 7, size = 10, prob = 0.7)

# b) Pr(7 heads or less) -> CDF
# ?(q = 7, size = 10, prob = 0.7)

# 1-?(q = 7, size = 10, prob = 0.7)
# ?(q = 7, size = 10, prob = 0.7, lower.tail = F)

# c) Double check
```