BSc CogSci - 2. Sem - Methods 2 - Assignment 1 - Q2

Group 0

10-03-2024

Methods 2 - Assignment 1

Group Members: Silke, Aiswarya, Mie, Paula, Benjamin

```
# loading packages
library(pacman)
pacman::p_load(tidyverse, ggplot2)

rm(list = ls())
```

Question 1: ROS 5.2 Continuous probability simulation (Silke)

The logarithms of weights (in pounds) of men in the United States are approximately normally distributed with mean 5.13 and standard deviation 0.17; women's log weights are approximately normally distributed with mean 4.96 and standard deviation 0.20. Suppose 10 adults selected at random step on an elevator with a capacity of 1750 pounds. What is the probability that their total weight exceeds this limit?

Answer:

There is approximately a 6.8% risk that the total weight of 10 randomly sampled adults exceed the max weight of 1750.

```
#Ensure simulation is the same each time.
set.seed(24)

#Simulate weight for 10 people (from a normal distribution for weight and binomial distribution for gen
simulations <- 1000
sum_weight <- numeric(simulations)

for (s in 1:simulations) {
  N <- 10
  female <- rbinom(N, 1, 0.49)
  weight <-
    ifelse(female == 1, rnorm(N, 4.96, 0.20), rnorm(N, 5.13, 0.17))
  sum_weight[s] <- sum(exp(weight))
}

#Define weight for elevator and calculate the probability of the total weight of the 10 adults simulate
elevator <- 1750
```

```r
probability <- 1 - pnorm(elevator, mean(sum_weight), sd(sum_weight))
probability
```

```
## [1] 0.0683674
```

Question 2: ROS 5.6, Propagation of Uncertainty (Aiswarya)

Breakdown of Process

To estimate the total amount of money saved by the new product (mean = 5, se = 4), simulations may be used. A simulation count of 1000 is established (“n_sims”) along with the parameters set out by the problem ([1] “m_unitCost”, “se_unitCost”; [2] “m_marketSize”, “se_marketSize”). Two independent, normal distributions are created using the two separate sets of parameters, within the confines of a for-loop. Within this same for-loop, a calculation of the estimated total amount of money saved (“savings_CostxSize”) may be carried out. Finally, the results of the simulations may be viewed. With consideration of the fact that there are 1000 simulations, the head() function may be used to view an extract of the results for the sake of convenience.

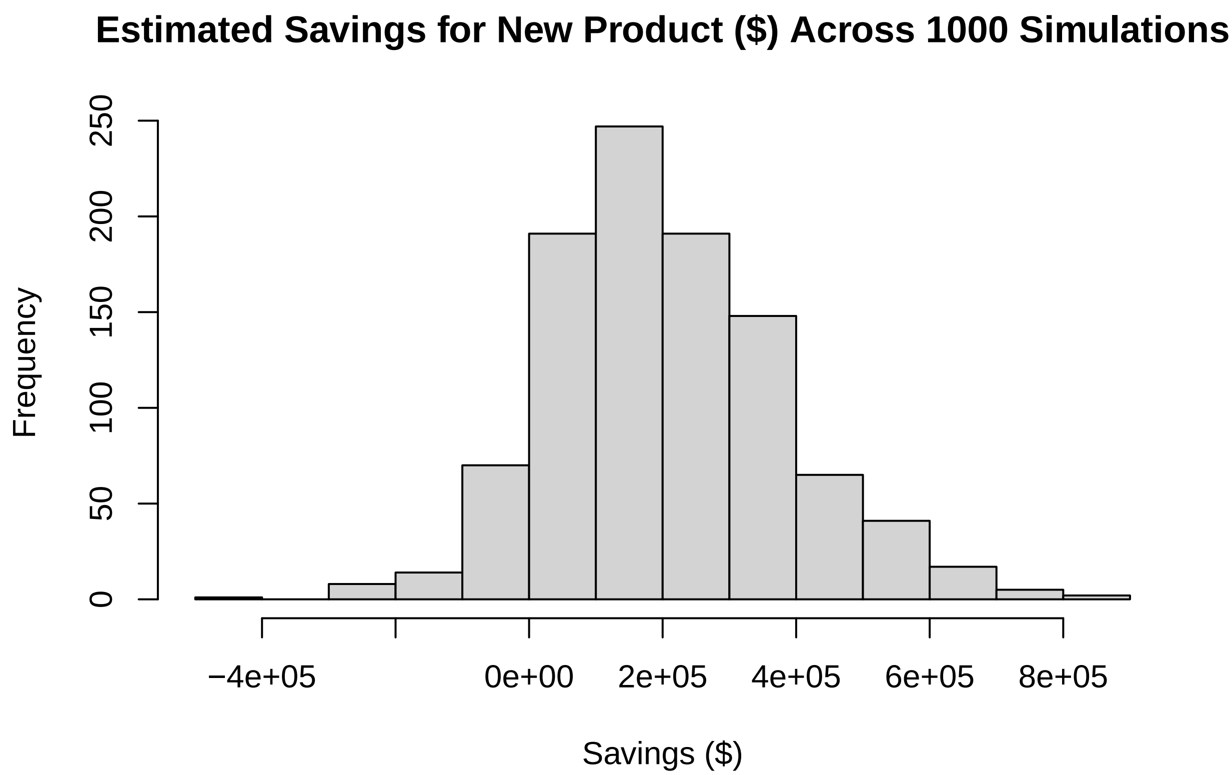
- An alternate version of the calculations carried out without the use of a for-loop may be viewed within the appendix.

```r
#setting the seed for reproducibility
set.seed(711)

# setting the no. of simulations
n_sims <- 1000

# setting up the parameters -- [1] savings per unit, mean and se
m_unitCost <- 5
se_unitCost <- 4

# setting up parameters -- [2] size of the market, mean and se
m_marketSize <- 40000
se_marketSize <- 10000

# initialize vector to store total savings
savings_CostxSize <- numeric(n_sims)

# perform simulations
for (i in 1:n_sims) {
  # simulate cost savings per unit
  norm_unitCost <- rnorm(1, mean = m_unitCost, sd = se_unitCost)

  # simulate market size
  norm_marketSize <-
    rnorm(1, mean = m_marketSize, sd = se_marketSize)

  # calculate total savings for this simulation
  savings_CostxSize[i] <- norm_unitCost * norm_marketSize
}
```

```r
# viewing (an excerpt of the) total savings
head(savings_CostxSize)
```

```
## [1]  98681.42 156241.75 191346.23 547709.46 389187.04  86843.80
```

```r
# creating a histogram to view distribution of estimated savings
hist_CostxSavings <-
  hist(savings_CostxSize, main = "Estimated Savings for New Product ($) Across 1000 Simulations", xlab =
```



Question 3: ROS 5.6, Inference for a ratio of parameters (Benjamin, Mie, Paula)

A (hypothetical) study compares the costs and effectiveness of two different medical treatments.

- In the first part of the study, the difference in costs between treatments A and B is estimated at $600 per patient, with a standard error of $400, based on a regression with 50 degrees of freedom.
- In the second part of the study, the difference in effectiveness is estimated at 3.0 (on some relevant measure), with a standard error of 1.0, based on a regression with 100 degrees of freedom.
- For simplicity, assume that the data from the two parts of the study were collected independently.

  Inference is desired for the incremental cost-effectiveness ratio: the difference between the average costs of the two treatments, divided by the difference between their average effectiveness, a problem discussed further by Heitjan, Moskowitz, and Whang (1999).

(a) Create 1000 simulation draws of the cost difference and the effectiveness difference, and make a scatterplot of these draws.
(b) Use simulation to come up with an estimate, 50% interval, and 95% interval for the incremental cost-effectiveness ratio.
(c) Repeat, changing the standard error on the difference in effectiveness to 2.0.

---

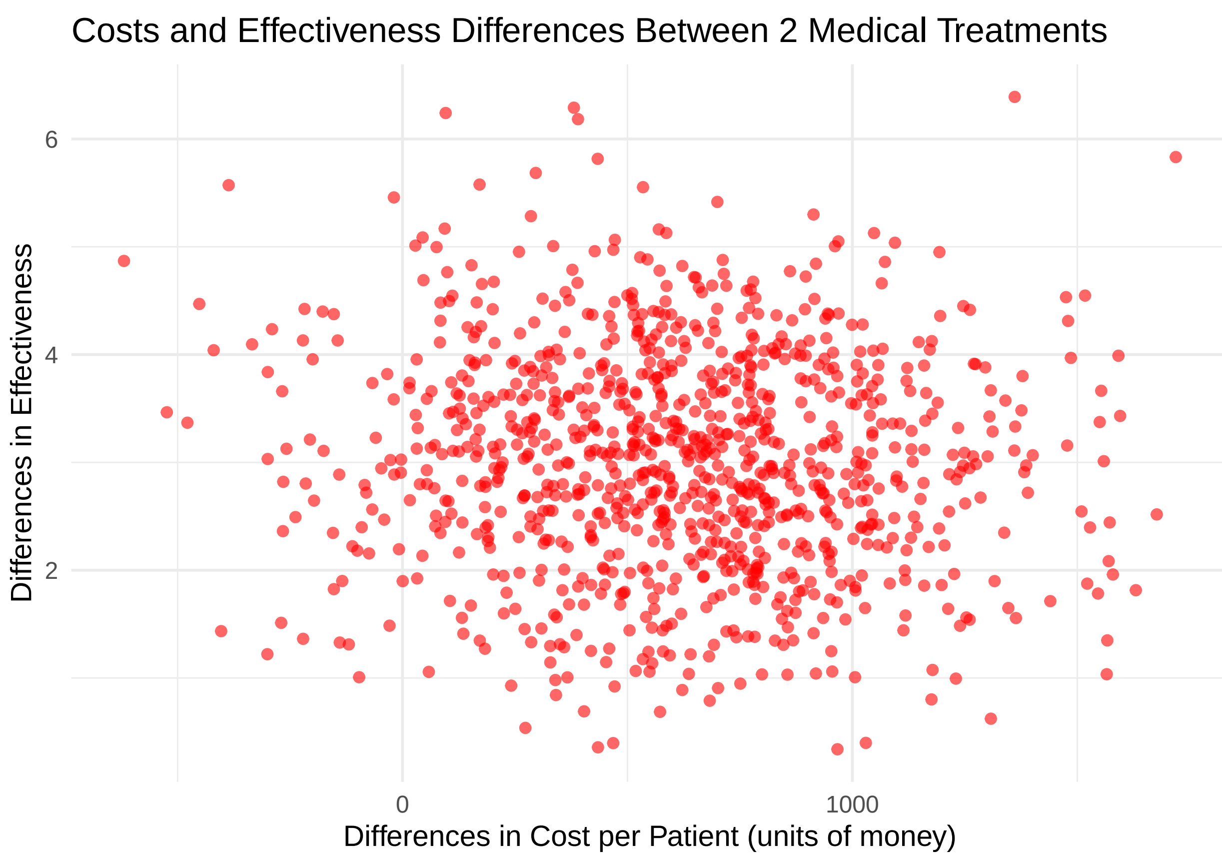
a) Benjamin first we made a function for the simulations, which then was called 1000 times via a for loop, then the cost and effectiveness differences are made into a dataframe and then plotted as a scatterplot

```
### a)

## 1) create 1000 simulations draws of the cost difference and effectiveness difference by making a func
set.seed(123) # sets the seed so we could compare a) and c)

# declare variables
costDiff_sim = c()
effDiff_sim = c()

costDiff_se = 400
effDiff_se = 1

costDiff_mean = 600
effDiff_mean = 3

print(rnorm(2, mean = costDiff_mean, sd = costDiff_se))
```

```
## [1] 375.8097 507.9290
```

```
# function simulating the estimates of the cost and effectiveness differences
costDiff_simulator <- function()
{
  costDiff = rnorm(1, mean = costDiff_mean, sd = costDiff_se)

  effDiff = rnorm(1, mean = effDiff_mean, sd = effDiff_se)

  returnVector <- c(costDiff, effDiff)

  return(returnVector)

}

# for loop runing the function 1000 times and saves the estimates of the cost and effectiveness differe
for (i in 1:1000)
{
  simulations <- costDiff_simulator()
  costDiff_sim[i] <- simulations[1]
  effDiff_sim[i] <- simulations[2]
}

# 2) creates a dataframe with the cost and effectiveness differences to use for plotting
```

```r
data <- data.frame(x_costDiff = costDiff_sim, # the x values are the cost differences
                   y_effDiff = effDiff_sim)  # the y values are the effectiveness differences)

# plots a scatterplot of the cost and effectiveness differences
ggplot(data, aes(x = x_costDiff)) +
  geom_point(aes(y = y_effDiff), alpha = 0.6, color = 'red') +  # Plotting the data points
  labs(x = 'Differences in Cost per Patient (units of money)',
       y = "Differences in Effectiveness",
       title = 'Costs and Effectiveness Differences Between 2 Medical Treatments',) +
  theme_minimal()
```



b) Mie firstly the estimate, the mean, of the incremental cost-effectiveness ratio is calculated. Then its se is calculated based on the relative standard error of the estimates for the cost and effectiveness differences, which is then used to get the 50% and 95% confidence interval.

```r
### b)

## 1) the incremental cost-effectiveness ratio is calculated based on the simulations from a)
ratioCostEff = costDiff_sim/effDiff_sim
#print(ratioCostEff)

## 2) the mean (the estimate) is found of the incremental cost-effectiveness ratio
ratioCostEff_estimate = mean(ratioCostEff)
```

```r
## 3) calculates the standard error of the estimate and uses it for making the 50% and 95% confidence i

# finds the relative SE of the cost and the effectiveness differences
relativeSE_cost = costDiff_se/mean(costDiff_sim)

relativeSE_eff = effDiff_se/mean(effDiff_sim)

# calculates the se of the estimate
ratioCostEff_se = ratioCostEff_estimate * sqrt(relativeSE_cost^2+relativeSE_eff^2)

      # multiplication makes the SE appropriately scaled relative to the estimate (ratioCostEff_estimat

      # AND the sqrt() and ^2 is basically for making sure that the se is not negative

# calculates the 50% CI
startCI_50 = round( ratioCostEff_estimate - (2/3) * ratioCostEff_se, 2)

endCI_50 = round(ratioCostEff_estimate + (2/3) * ratioCostEff_se, 2)

CI_50 = print(paste("the 50% confidence interval is [", startCI_50, ", ", endCI_50, "]"))
```

```
## [1] "the 50% confidence interval is [ 125.96 ,  364.31 ]"
```

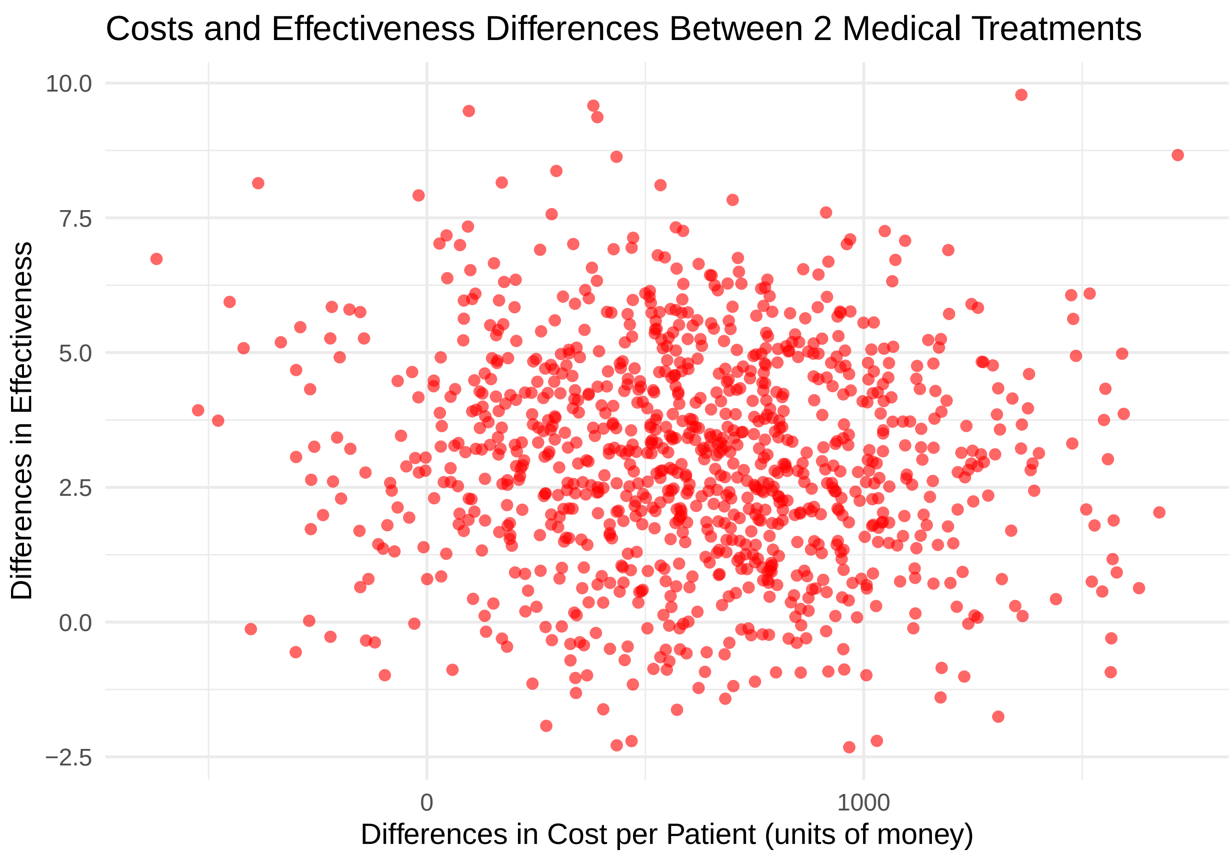
```r
# calculates the 95% CI
startCI_95 = round( ratioCostEff_estimate - 2 * ratioCostEff_se, 2)

endCI_95 = round( ratioCostEff_estimate + 2 * ratioCostEff_se, 2)

CI_95 = print(paste("the 95% confidence interval is [", startCI_95, ", ", endCI_95, "]"))
```

```
## [1] "the 95% confidence interval is [ -112.39 ,  602.66 ]"
```

c) Paula the standard error on the difference in effectiveness is changed from 1 to 2. then the code is as for a) and b)

```r
### c)
set.seed(123) # sets the seed so we could compare a) and c)

effDiff_se = 2  # sets the new standard error of the differences in effectiveness

## 1) the a) part of the code - making the simulations for the estimates for cost and effectiveness dif

# resets the vectors variables
costDiff_sim = c()
effDiff_sim = c()

# function simulating the estimates of the cost and effectiveness differences
costDiff_simulator <- function()
{
  costDiff = rnorm(1, mean = costDiff_mean, sd = costDiff_se)
```

```r
  effDiff = rnorm(1, mean = effDiff_mean, sd = effDiff_se)

  returnVector <- c(costDiff, effDiff)

  return(returnVector)

}

# for loop runing the function 1000 times and saves the estimates of the cost and effectiveness differe
for (i in 1:1000)
{
  simulations <- costDiff_simulator()
  costDiff_sim[i] <- simulations[1]
  effDiff_sim[i] <- simulations[2]
}

# creates a dataframe with the cost and effectiveness differences to use for plotting
data <- data.frame(x_costDiff = costDiff_sim, # the x values are the cost differences
                   y_effDiff = effDiff_sim)  # the y values are the effectiveness differences)

# plots a scatterplot of the cost and effectiveness differences
ggplot(data, aes(x = x_costDiff)) +
  geom_point(aes(y = y_effDiff), alpha = 0.6, color = 'red') +  # Plotting the data points
  labs(x = 'Differences in Cost per Patient (units of money)',
       y = "Differences in Effectiveness",
       title = 'Costs and Effectiveness Differences Between 2 Medical Treatments',) +
  theme_minimal()
```




```
## 2) the b) part of the code - calculating an estimate of the incremental cost-effectiveness ratio. The

## the incremental cost-effectiveness ratio is calculated based on the simulations from a)
ratioCostEff = costDiff_sim/effDiff_sim

## the mean (the estimate) is found of the incremental cost-effectiveness ratio
ratioCostEff_estimate = mean(ratioCostEff)

## calculates the standard error of the estimate and uses it for making the 50% and 95% confidence inte

# finds the relative SE of the cost and the effectiveness differences
relativeSE_cost = costDiff_se/mean(costDiff_sim)

relativeSE_eff = effDiff_se/mean(effDiff_sim)

# calculates the se of the estimate
ratioCostEff_se = ratioCostEff_estimate * sqrt(relativeSE_cost^2+relativeSE_eff^2)

# calculates the 50% CI
startCI_50 = round( ratioCostEff_estimate - (2/3) * ratioCostEff_se, 2)

endCI_50 = round(ratioCostEff_estimate + (2/3) * ratioCostEff_se, 2)

CI_50 = print(paste("the 50% confidence interval is [", startCI_50, ", ", endCI_50, "]"))
```

```
## [1] "the 50% confidence interval is [ 94.07 ,  396.58 ]"
```

```
# calculates the 95% CI
startCI_95 = round(ratioCostEff_estimate - 2 * ratioCostEff_se, 2)

endCI_95 = round(ratioCostEff_estimate + 2 * ratioCostEff_se, 2)

CI_95 = print(paste("the 95% confidence interval is [", startCI_95, ", ", endCI_95, "]"))
```

```
## [1] "the 95% confidence interval is [ -208.43 ,  699.09 ]"
```

Conclusion (*written in French*): Alors (*'so' written in French*) as the SE of effectiveness differences has doubled (from 1 to 2), the intervals* for the incremental cost-effectiveness ratio has also become wider because of it.

- we expect that the book means confidence intervals, but it was not specified to our great dismay

APPENDIX

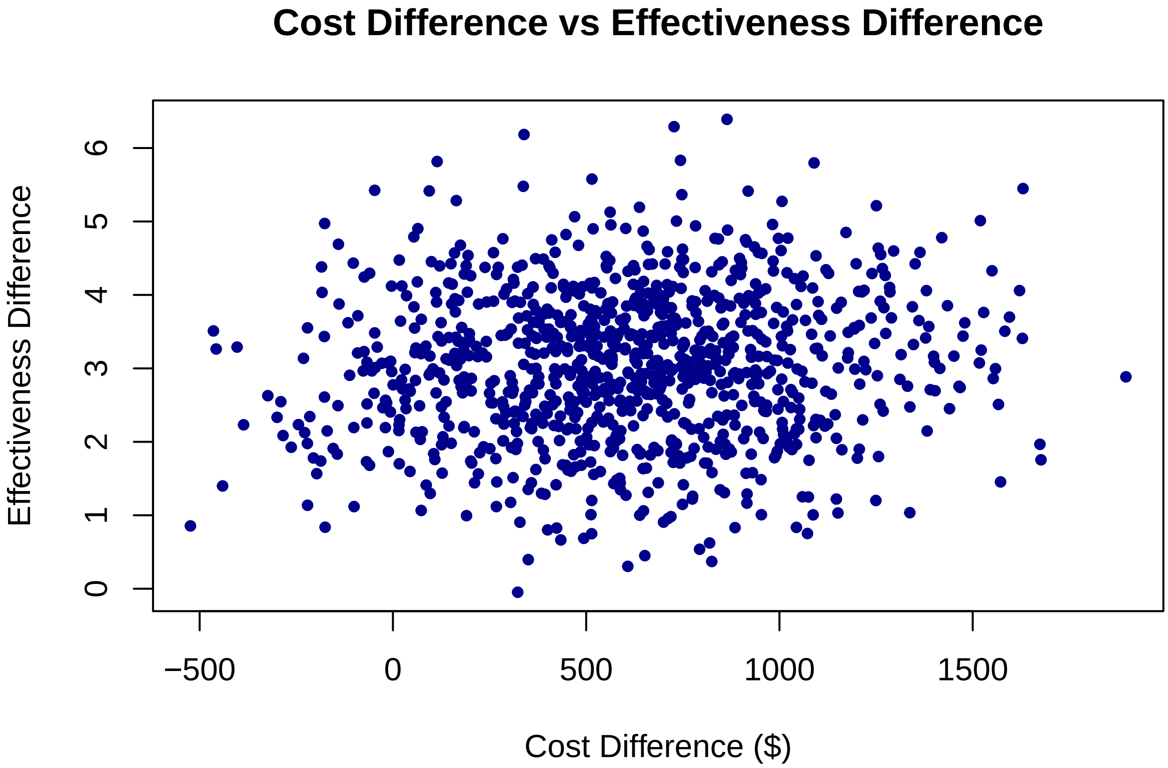
APPENDIX (Q2, 5.6)

```
# storing results, simulating cost savings + market size
norm_unitCost <-
  rnorm(n_sims, mean = m_unitCost, sd = se_unitCost) # [1]
norm_marketSize <-
  rnorm(n_sims, mean = m_marketSize, sd = se_marketSize) # [2]

# calculating total savings for each simulation
savings_CostxSize_2 <- norm_unitCost * norm_marketSize
head(savings_CostxSize_2)
```

```
## [1] 113703.95 218415.88  72309.89 326206.25 375885.87 100607.13
```

APPENDIX (Q3, 5.10)

```
# Set seed for reproducibility
set.seed(123)

# Number of simulation draws
num_simulations <- 1000

# Generate random draws for the cost difference
cost_difference <- rnorm(num_simulations, mean = 600, sd = 400)

# Generate random draws for the cost difference
effectiveness_difference <-
  rnorm(num_simulations, mean = 3.0, sd = 1.0)

# Create scatterplot
plot(
  cost_difference,
  effectiveness_difference,
  xlab = "Cost Difference ($)",
  ylab = "Effectiveness Difference",
  main = "Cost Difference vs Effectiveness Difference",
  pch = 20,
  col = "darkblue"
)
```


Cost Difference vs Effectiveness Difference
Effectiveness Difference
0
1
2
3
4
5
6
−500
0
500
1000
1500
Cost Difference ($)