Foundadtions_assignment_2

Bernardo Arambula

2023-08-21

BAX-400 Homework 2

Mehul Rangwala

Summer 2023

Total Points: 200

Due Date: Sunday, August 27, 2023 11:59 PM

Instructions

1. There are 18 questions. Some have multiple parts.
2. All questions should be completed using R.
3. Please complete each question for full or partial credit.
4. Submit an RMD file knitted as HTML. The knitted file should show the code and the result (output) below it. No other form of submission will be accepted. You don't need to submit your RMD files.
5. Like I mentioned in the class, interpretations of your results are the key. Interpretations are at the heart of statistics. Results are meaningless with- out interpretations and insights.
6. Please comment your R code like I did on my solved examples. The reason is that we want to award you partial credit commensurate with your at- tempt, and we will be able to understand your logic and responses better if the sections of your work are clearly commented.
7. If you have any questions, then please post them under Discussions - Homework 2 on Canvas so that the responses can benefit everyone.

```r
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Question 1

The number of traffic fatalities in a typical month in a given state has a normal distribution with mean 125 and standard deviation 31.

a. If a person in the highway department claims that there will be at least m fatalities in the next month with probability 0.95, what value of m makes this claim true?

```
m <- qnorm(0.95, mean = 125, sd = 31, lower.tail = TRUE)
sprintf("The value of m that makes the above claim true is %.2f fatalities", m)
```

```
## [1] "The value of m that makes the above claim true is 175.99 fatalities"
```

b. If the claim is that there will be no more than n fatalities in the next month with probability 0.98, what value of n makes this claim true?

```
n <- qnorm(0.98, mean = 125, sd = 31, lower.tail = TRUE)
sprintf("The value of n that makes the above claim true is %.2f fatalities", n)
```

```
## [1] "The value of n that makes the above claim true is 188.67 fatalities"
```

Question 2

Many companies use sampling to determine whether a batch should be accepted. An (n, c) sampling plan consists of inspecting n randomly chosen items from a batch and accepting the batch if c or fewer sampled items are defective. Suppose a company uses a (100, 5) sampling plan to determine whether a batch of 10,000 computer chips is acceptable.

a. The “producer’s risk” of a sampling plan is the probability that an acceptable batch will be rejected by the sampling plan. Suppose the customer considers a batch with 3% defectives acceptable. What is the producer’s risk for this sampling plan?

```
producer_risk <- pbinom(4, 100, 0.03, lower.tail = FALSE)
sprintf("The producer's risk for this sampling plan is %.4f", producer_risk)
```

```
## [1] "The producer's risk for this sampling plan is 0.1821"
```

b. The “consumer’s risk” of a sampling plan is the probability that an unacceptable batch will be accepted by the sampling plan. Our cus- tomer says that a batch with 9% defectives is unacceptable. What is the consumer’s risk for this sampling plan?

```
consumer_risk <- pbinom(5, 100, 0.09)
sprintf("The consumer's risk for this sampling plan is %.4f", consumer_risk)
```

```
## [1] "The consumer's risk for this sampling plan is 0.1045"
```

Question 3

According to an IRS study, it takes a mean of 330 minutes for taxpayers to prepare, copy, and electronically file a 1040 tax form. This distribution of times follows the normal distribution and the standard deviation is 80 minutes. A consumer watchdog agency selects a random sample of 40 taxpayers.

a. What is the standard error of the mean in this example?

```
sd <- 80
n <- 40
std_error_of_means <- sd / sqrt(n)
sprintf("The standard error of means in this example is %.2f", std_error_of_means)
```

```
## [1] "The standard error of means in this example is 12.65"
```

b. What is the likelihood the sample mean is greater than 320 minutes?

```
Prob_greater_than_320 <- pnorm(320, 330, std_error_of_means, lower.tail = FALSE)
sprintf("The likelihood that the sample mean is greater than 320 minutes is %.2f", Prob_
greater_than_320)
```

```
## [1] "The likelihood that the sample mean is greater than 320 minutes is 0.79"
```

c. What is the likelihood the sample mean is between 320 and 350 minutes?

```
prob_between_320_and_350 <- pnorm(350, 330, std_error_of_means) - pnorm(320, 330, std_er
ror_of_means)
sprintf("The likelihood that the sample mean is between 320 minutes and 350 minutes is
%.2f", prob_between_320_and_350)
```

```
## [1] "The likelihood that the sample mean is between 320 minutes and 350 minutes is 0.
73"
```

d. What is the likelihood the sample mean is greater than 350 minutes?

```
prob_greater_350 <- pnorm(350, 330, std_error_of_means, lower.tail = FALSE)
sprintf("The likelihood that the sample mean is greater than 350 minutes is %.2f", prob_
greater_350)
```

```
## [1] "The likelihood that the sample mean is greater than 350 minutes is 0.06"
```

e. What is the probability that the sampling error would be more than 20 minutes? Hint: Sampling error is the difference between the sample mean and the population mean; i.e., X – μ. 2

```
sampling_error <- 20
prob_sampling_error_more_20 <- 1 - pnorm(330 + sampling_error, 330, std_error_of_means)
sprintf("The probability that the sampling error would be more than 20 minutes is %.2f",
prob_sampling_error_more_20)
```

```
## [1] "The probability that the sampling error would be more than 20 minutes is 0.06"
```

Question 4

American workers are increasingly planning to delay retirement (cnbc.com, Aug 25, 2019). A researcher finds that 35% of employed adults of age 62 and older say they have pushed back their retirement date.

a. What is the probability that in a sample of 100 employed adults of age 62 and older, more than 40% have pushed back their retirement date?

```
prob_more_40 <- pbinom(40, 100, 0.35, lower.tail = FALSE)
sprintf("The probability that in a sample of 100 employed adults age 62 and older, more
than 40 percent have pushed back their retirement date is %.2f", prob_more_40)
```

```
## [1] "The probability that in a sample of 100 employed adults age 62 and older, more t
han 40 percent have pushed back their retirement date is 0.13"
```

b. What is the probability that in a sample of 200 employed adults of age 62 and older, more than 40% have pushed back their retirement date?

```
prob_more_40 <- pbinom(80, 200, 0.35, lower.tail = FALSE)
sprintf("The probability that in a sample of 200 employed adults age 62 and older, more
than 40 percent have pushed back their retirement date is %.2f", prob_more_40)
```

```
## [1] "The probability that in a sample of 200 employed adults age 62 and older, more t
han 40 percent have pushed back their retirement date is 0.06"
```

c. Comment on the difference between the two estimated probabilities.

As expected, the probability decreases with a larger sample size when maintaining the same proportion of adults delaying retirement (35%). This is because larger samples reduce the variability in observed proportions and tend to converge towards the population proportion.

Question 5

A retailer of computing products sells a variety of computer-related products. One of the most popular products is an HP laser printer. The average weekly demand is 200. Lead time for a new order from the manufacturer to arrive is 1 week. If the demand for printers were constant, the retailer would reorder when there were exactly 200 printers in inventory. However, the demand is a random variable. An analysis of previous weeks reveals that the weekly demand standard deviation is 30. The retailer wants the probability of running short in any week to be no more than 6%. How many HP laser printers should be in stock when the retailer reorders from the manufacturer?

```
mean_demand <- 200
sd_demand <- 30
probability <- 0.06

# Find the reorder point such that the probability of running out is no more than 6%
reorder_point <- qnorm(1 - probability, mean = mean_demand, sd = sd_demand)
sprintf("The number of HP laser printers that should be in stock when the retailer reord
ers from the manufacturer is %.2f", reorder_point)
```

```
## [1] "The number of HP laser printers that should be in stock when the retailer reorde
rs from the manufacturer is 246.64"
```

Question 6

One of the stores of a large chain of supermarkets in Northern California uses the same associate to ring customers up and bag their groceries. This means there is one associate who helps each customer at the checkout counter. The current service rate for this process is 8 per hour. What is the probability that the checkout takes longer than 15 minutes? Customers found this process to be too slow and brought this to the corporate office's attention. Upon investigation, the management at the corporate office of this chain of supermarkets realized that this probability was indeed larger than their internal benchmark of 0.05. It wants to match this probability to their internal benchmark. What should the new service rate for this store be? Provide one idea of how this new service rate can be implemented in the store.

```
# Current service rate (customers per hour)
current_service_rate <- 8

# Time in hours (15 minutes = 0.25 hours)
fifteen_mins <- 0.25

# A) Probability that checkout takes longer than 15 minutes
prob_longer_than_fifteen <- pexp(fifteen_mins, rate = current_service_rate, lower.tail =
FALSE)
sprintf("A) The probability that checkout takes longer than 15 mins is %.2f", prob_longe
r_than_fifteen)
```

```
## [1] "A) The probability that checkout takes longer than 15 mins is 0.14"
```

```
# B) New service rate to match the internal benchmark of 0.05
new_service_rate <- -log(0.05) / fifteen_mins
sprintf("B) The new service rate should be %.2f customers per hour", new_service_rate)
```

```
## [1] "B) The new service rate should be 11.98 customers per hour"
```

```
# C) New Idea
sprintf("C) The store can acheive their desired probabilty by having both a checker and
a bagger at the checkout line to decrease the wait time between clients")
```

```
## [1] "C) The store can acheive their desired probabilty by having both a checker and a
bagger at the checkout line to decrease the wait time between clients"
```

Question 7

Toll booths on the New York State Thruway are often congested because of the large number of cars waiting to pay. A consultant working for the state concluded that if service times are measured from the time a car stops in line until it leaves, service times are exponentially distributed with a mean of 2.7 minutes. What proportion of cars can get through the toll booth in less than 3 minutes?

```
# Mean service time in minutes
mean_service_time <- 2.7

# Rate parameter (lambda) for exponential distribution
toll_rate <- 1 / mean_service_time

# Time in minutes (less than 3 minutes)
time_less_than_3 <- 3

# Proportion of cars that can get through the toll booth in less than 3 minutes
p_less_than_3 <- pexp(time_less_than_3, rate = toll_rate)

sprintf("The proportion of cars that can get through the toll booth in less than 3 minut
es is %.2f", p_less_than_3)
```

```
## [1] "The proportion of cars that can get through the toll booth in less than 3 minute
s is 0.67"
```

Question 8

Leslie loves to swim and compete in races. The time it takes Leslie to swim 100 yards in a race follows a normal distribution with mean of 62 seconds and standard deviation of 2 seconds. In her next five races, what is the probability that she will swim under a minute exactly twice?

```
# Given parameters
mean_time <- 62
sd_time <- 2
time_threshold <- 60
num_races <- 5
num_successes <- 2

# Calculate the probability of swimming under 60 seconds
prob_swim_under_1min <- pnorm(time_threshold, mean = mean_time, sd = sd_time)

# Calculate the probability of exactly 2 successes out of 5 races
prob_2_success <- dbinom(num_successes, size = num_races, prob = prob_swim_under_1min)

# Output the result
sprintf("The probability that 2 of Leslie's next five races are timed at under 60 second
s is %.2f", prob_2_success)
```

```
## [1] "The probability that 2 of Leslie's next five races are timed at under 60 seconds
is 0.15"
```

Question 9

The Internal Revenue Service is studying the category of charitable contributions. A sample of 25 returns is selected from young couples between the ages of 20 and 35 who had an adjusted gross income of more than $100,000. Of these 25 returns, five had charitable contributions of more than $1,000. Four of these returns are selected for a comprehensive audit.

a. What is the probability exactly one of the four audited had a charitable deduction of more than $1,000?

```
# Given parameters
total_returns <- 25
donation_returns <- 5
no_donation_returns <- total_returns - donation_returns
sample_audited <- 4
success_audited <- 1

# Calculate the probability using the hypergeometric distribution
probability <- dhyper(success_audited, donation_returns, no_donation_returns, sample_aud
ited)

# Output the result
sprintf("The probability that exactly one of the four audited returns had a charitable d
eduction of more than $1,000 is %.2f", probability)
```

```
## [1] "The probability that exactly one of the four audited returns had a charitable de
duction of more than $1,000 is 0.45"
```

b. What is the probability at least one of the audited returns had a charitable contribution of more than $1,000?

```
# Probability of at least one audited return with charitable contributions
prob_atleast_1 <- phyper(0, donation_returns, no_donation_returns, sample_audited, lowe
r.tail = FALSE)

# Output the result
sprintf("The probability that at least one of the audited returns had a charitable contr
ibution of more than $1,000 is %.2f", prob_atleast_1)
```

```
## [1] "The probability that at least one of the audited returns had a charitable contri
bution of more than $1,000 is 0.62"
```

Question 10

The sales of Mercedes automobiles in the Sacramento region follow a Poisson distribution with a mean of three per day.

a. What is the probability that no Mercedes is sold on a particular day?

```
# Given parameters
mean_cars <- 3
zero_day <- 0

# Calculate the probability of selling zero cars
prob_zero <- dpois(zero_day, mean_cars)

# Output the result
sprintf("The probability that zero cars are sold on a particular day is %.2f", prob_zer
o)
```

```
## [1] "The probability that zero cars are sold on a particular day is 0.05"
```

b. What is the probability that for each of the five consecutive days at least one Mercedes is sold?

```
# Calculate the probability of selling at least one car on a single day
prob_atleast_one_per_day <- ppois(zero_day, mean_cars, lower.tail = FALSE)

# Calculate the probability of selling at least one car each day for five consecutive da
ys
prob_after_5days_atleast1car <- prob_atleast_one_per_day^5

# Output the result
sprintf("The probability that for each of the five consecutive days at least one Mercede
s is sold is %.2f", prob_after_5days_atleast1car)
```

```
## [1] "The probability that for each of the five consecutive days at least one Mercedes
is sold is 0.77"
```

Question 11

The shoplifting sensor at a certain Best Buy Electronics store exit gives an alarm 0.5 times a minute.

a. Find the median waiting time until the next alarm.

```
median_wait_time <- qexp(0.5, rate = 0.5)
sprintf("The median wait time in minutes is %.2f", median_wait_time)
```

```
## [1] "The median wait time in minutes is 1.39"
```

b. Find the first quartile of waiting time before the next alarm.

```
first_quartile_wait_time <- qexp(0.25, rate = 0.5)
sprintf("The first quartile of waiting time before the next alarm is %.2f minutes", firs
t_quartile_wait_time)
```

```
## [1] "The first quartile of waiting time before the next alarm is 0.58 minutes"
```

c. Find the 30th percentile of waiting time until the next alarm.

```
thirtieth_percentile_wait_time <- qexp(0.3, rate = 0.5)
sprintf("The 30th percentile of waiting time before the next alarm is %.2f minutes", thi
rtieth_percentile_wait_time)
```

```
## [1] "The 30th percentile of waiting time before the next alarm is 0.71 minutes"
```

Question 12

The CSV file SupermarketTrans contains over 14,000 transactions made by supermarket customers over a period of approximately two years. (The data are not real, but real supermarket chains have huge data sets similar to this one.) Column A contains the date of the purchase, column B is a unique identifier for each customer, columns C–G contain information about the customer, columns H–J contain the location of the store, columns K–M contain information about the product purchased, and the last two columns indicate the number of items purchased and the amount paid. For this question, consider this data set the population of transactions.

```
SupermarketTrans <- read_csv("/Users/bernardoarambula/Documents/MSBA/BAX400 foundations/
Homework2/SupermarketTrans.csv")
```

```
## Rows: 14059 Columns: 16
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (12): Purchase Date, Gender, Marital Status, Homeowner, Annual Income, C...
## dbl  (4): Transaction, Customer ID, Children, Units Sold
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a. If you were interested in estimating the mean of Revenue for the population, why might it make sense to use a stratified sample, stratified by product family, to estimate this mean?

It might make sense to use a stratified sample because there might be a great deal of variation between the different product families. The dataset includes food, drink and non-consumables as possible product families and each subgroup could be variated differently. stratified sampling by product family makes sense because it improves the accuracy and precision of the revenue mean estimate by ensuring that all subgroups are adequately represented and reducing the overall variability in the estimate.

b. Suppose you want to generate a stratified random sample, stratified by product family, and have the total sample size be 250. If you use proportional sample sizes, how many transactions should you sample from each of the three product families?

```
# Make a table with count of each product family
product_fam_count <- table(SupermarketTrans$`Product Family`)
print(product_fam_count)
```

```
##
##          Drink           Food Non-Consumable
##           1250          10153           2656
```

```
# Gets the proportion of each product family
proportion_product_fam <- prop.table(product_fam_count)
print(proportion_product_fam)
```

```
##
##          Drink           Food Non-Consumable
##     0.08891102     0.72217085     0.18891813
```

```
# Proportional sample size for each product family to get a total sample size of 250
total_sample_size <- 250
prob12b_sample <- round(proportion_product_fam * total_sample_size)
print(prob12b_sample)
```

```
##
##          Drink           Food Non-Consumable
##             22            181             47
```

c. Using the sample sizes from part b, generate a corresponding stratified random sample. What are the individual sample means from the three product families? What are the sample standard deviations?

```
# prob12b_sample contains the sample sizes for each product family
prob12b_sample <- c(Drink = 22, Food = 181, `Non-Consumable` = 47)

# Clean the Revenue column
SupermarketTrans$Revenue <- as.numeric(gsub("\\$", "", SupermarketTrans$Revenue))

# Sample and calculate statistics for Drink
Drink <- SupermarketTrans[SupermarketTrans$`Product Family` == "Drink", , drop = FALSE]
Drink_sample <- Drink[sample(nrow(Drink), prob12b_sample["Drink"]), ]
meandrink <- mean(Drink_sample$Revenue)
sddrink <- sd(Drink_sample$Revenue)

sprintf("The mean and standard deviation for my Drink stratified random sample are %.2f
and %.2f, respectively.", meandrink, sddrink)
```

```
## [1] "The mean and standard deviation for my Drink stratified random sample are 11.66
and 8.60, respectively."
```

```
# Sample and calculate statistics for Food
Food <- SupermarketTrans[SupermarketTrans$`Product Family` == "Food", , drop = FALSE]
Food_sample <- Food[sample(nrow(Food), prob12b_sample["Food"]), ]
meanfood <- mean(Food_sample$Revenue)
sdfood <- sd(Food_sample$Revenue)

sprintf("The mean and standard deviation for my Food stratified random sample are %.2f a
nd %.2f, respectively.", meanfood, sdfood)
```

```
## [1] "The mean and standard deviation for my Food stratified random sample are 13.27 a
nd 8.47, respectively."
```

```
# Sample and calculate statistics for Non-Consumable
Non_consumable <- SupermarketTrans[SupermarketTrans$`Product Family` == "Non-Consumabl
e", , drop = FALSE]
Non_consumable_sample <- Non_consumable[sample(nrow(Non_consumable), prob12b_sample["Non
-Consumable"]), ]
meannc <- mean(Non_consumable_sample$Revenue)
sdnc <- sd(Non_consumable_sample$Revenue)

sprintf("The mean and standard deviation for my Non-Consumable stratified random sample
are %.2f and %.2f, respectively.", meannc, sdnc)
```

```
## [1] "The mean and standard deviation for my Non-Consumable stratified random sample a
re 12.83 and 7.72, respectively."
```

Question 13

Suppose aftergraduating from MSBA, you work for a survey research company. In a typical survey, you mail questionnaires to 150 companies. Some of these companies might decide not to respond. Assume that the nonresponse rate is 45%; that is, each company's probability of not responding, independently of the others, is 0.45. Suppose your company does this survey in two "waves." It mails the 150 questionnaires and waits a certain period for the responses. Assume that the nonresponse rate for this first wave is 45%. However, after this initial period, your company follows up (by telephone, say) on the nonrespondents, asking them to please respond. Suppose that the nonresponse rate on this second wave is 70%; that is, each original nonrespondent now responds with probability 0.3, independently of the others. Your company now wants to find the probability of obtaining at least 110 responses total. What is the probability (fraction of successes) of getting this required number of returns from both waves?

```
sent <- 150

# First wave
p_wave1_success <- 0.55
p_wave1_failure <- 0.45

# Second wave
p_wave2_success <- 0.3
p_wave2_failure <- 0.7

# Probability of nonresponse in both waves
prob_noresponse_w1w2 <- p_wave1_failure * p_wave2_failure

# Probability of response in either wave
prob_response_total <- 1 - prob_noresponse_w1w2

# Probability of at least 110 responses
prob_at_least_110 <- 1 - pbinom(109, 150, prob_response_total)

sprintf("The probability of receiving at least 110 total responses from both waves is %.
2f", prob_at_least_110)
```

```
## [1] "The probability of receiving at least 110 total responses from both waves is 0.1
2"
```

Question 14

It is not unusual for credit card customers to default on their credit charges. Typically customers who default share similar characteristics. Assume that the chance of defaulting on their credit charges is the same for everyone in the category. Let us look at one particular group of customers with the identical characteristics. Assume that each of these customers have a 0.07 probability of defaulting on his or her current credit charges. Also assume that the total charges are normally distributed with mean of $350 and standard deviation of $100. For this question, let us say that when a customer defaults, 20% of his or her charges can be recovered. The other 80% can be written off as bad debt.

a. What is the probability that a typical customer in this group will default and produce a write-off of more than $250 in bad debt?

```
p_default <- 0.07
recovery_rate <- 0.2
write_off_rate <- 0.8

# Calculate the threshold for total charges that would result in a write-off of more than $250
threshold_charges <- 250 / write_off_rate

# Calculate the probability of total charges being greater than this threshold
mean_charges <- 350
sd_charges <- 100

# Using the normal distribution to find the probability of charges exceeding the threshold
p_charges_exceed_threshold <- pnorm(threshold_charges, mean_charges, sd_charges, lower.tail = FALSE)

# Calculate the final probability
p_default_and_write_off <- p_default * p_charges_exceed_threshold

sprintf("The probability that a typical customer in this group will default and produce a write off of more than $250 in bad debt is %.2f", p_default_and_write_off)
```

```
## [1] "The probability that a typical customer in this group will default and produce a write off of more than $250 in bad debt is 0.05"
```

b. If there are 500 customers in this group, what are the mean and standard deviation of the number of customers who will meet the description in part a?

```
# Given values from part a
p_default_and_write_off <- p_default * p_charges_exceed_threshold

n_customers <- 500

# Mean number of customers who will meet the description in part a
mean_customers_meeting_criteria <- n_customers * p_default_and_write_off

# Standard deviation of the number of customers who will meet the description in part a
sd_customers_meeting_criteria <- sqrt(n_customers * p_default_and_write_off * (1 - p_default_and_write_off))

sprintf("The mean number of customers who will meet the description in part a is %.2f", mean_customers_meeting_criteria)
```

```
## [1] "The mean number of customers who will meet the description in part a is 22.62"
```

```
sprintf("The standard deviation of customers who will meet the description in part a is
%.2f", sd_customers_meeting_criteria)
```

```
## [1] "The standard deviation of customers who will meet the description in part a is
4.65"
```

c. Again assuming there are 500 customers in this group, what is the probability that at least 25 of them will meet the description in part a?

```
p_at_least_25 <- pbinom(24, n_customers, p_default, lower.tail = FALSE)

sprintf("The probability that at least 25 customers will meet the description in part a
is %.2f", p_at_least_25 <- pbinom(24, n_customers, p_default, lower.tail = FALSE))
```

```
## [1] "The probability that at least 25 customers will meet the description in part a i
s 0.97"
```

Question 15

An elevator rail is assumed to meet specifications if its diameter is between 0.98 and 1.01 inches. Each year a company produces 100,000 elevator rails. For a cost of 10/σ2 per year the company can rent a machine that produces elevator rails whose diameters have a standard deviation of σ. (The idea is that the company must pay more for a smaller variance.) Each such machine will produce rails having a mean diameter of one inch. Any rail that does not meet specifications must be reworked at a cost of $12. Assume that the diameter of an elevator rail follows a normal distribution.

a. 8 points What standard deviation (within 0.001 inch) minimizes the annual cost of producing elevator rails?

```
rails_produced <- 100000
d1 <- 0.98
d2 <- 1.01
rework_cost <- 12

# Define the total cost function
total_cost <- function(sigma) {
  # Calculate the probability of rework (rails outside the specification)
  prob_rework <- pnorm(d1, mean = 1, sd = sigma, lower.tail = TRUE) +
                 pnorm(d2, mean = 1, sd = sigma, lower.tail = FALSE)

  # Calculate the total rework cost
  total_rework_cost <- prob_rework * rails_produced * rework_cost

  # Calculate the machine rental cost
  machine_rental_cost <- 10 / (sigma^2)

  # Calculate the total cost
  total_cost <- machine_rental_cost + total_rework_cost

  return(total_cost)
}

# Use optimization to find the sigma that minimizes the total cost
optimal_sigma <- optimize(total_cost, interval = c(0.001, 1), tol = 0.001)$minimum

sprintf("The standard deviation that minimizes the annual cost of producing elevator rai
ls is %.3f inches", optimal_sigma)
```

```
## [1] "The standard deviation that minimizes the annual cost of producing elevator rail
s is 0.008 inches"
```

b. For your answer in part a, one elevator rail in 1000 will be at least how many inches in diameter?

```
# Calculate the threshold diameter
threshold_diameter <- qnorm(0.001, mean = 1, sd = optimal_sigma, lower.tail = FALSE)

sprintf("One elevator rail in 1000 will be at least %.2f inches in diameter.", threshold
_diameter)
```

```
## [1] "One elevator rail in 1000 will be at least 1.02 inches in diameter."
```

Question 16

Suppose you have the opportunity to play a game with a “wheel of fortune”. When you spin a large wheel, it is equally likely to stop in any position. Depending on where it stops, you win anywhere from $0 to $1000 (in $1 increments, assume). Let us suppose your winnings are actually based on not one spin, but on the average of n spins of the wheel. For example, if n = 2, your winnings are based on the average of two spins. If the first spin results in $580 and the second spin results in $320, you win the average, $450. Your objective is to use simulations to determine how does the distribution of your winnings depends on n. To address this objective, perform a simulation in R using the following guide- lines. Note: you need to show your work in R for all the parts, but only the summary table in part k) carries points. You won't get credit if you don't complete the summary table in part k).

a. Find the theoretical mean and theoretical standard deviation of the winnings.

```
# Theoretical mean
theoretical_mean <- (0 + 1000) / 2

# Theoretical standard deviation
theoretical_sd <- sqrt((1000 - 0)^2 / 12)

# Display the results
sprintf("The theoretical mean is %.2f and the theoretical standard deviation is %.2f", t
heoretical_mean, theoretical_sd)
```

```
## [1] "The theoretical mean is 500.00 and the theoretical standard deviation is 288.68"
```

b. Perform simulating 1 spin, 2 spins, 3 spins, 4 spins, 5 spins, 6 spins, 7 spins, 8 spins, 9 spins, and 10 spins. For each number of spins, perform 1,000 replications. Consider the number of spins as your sample size (n) and the number of replications as the number of samples. For example, 1,000 replications (samples): each replication (sample) containing 1 spin (sample size of 1) 1,000 replications (samples): each replication (sample) containing 2 spins (sample size of 2) 1,000 replications (samples): each replication (sample) containing 3 spins (sample size of 3) 1,000 replications (samples): each replication containing 4 spins (sample size of 4), etc.

```
set.seed(123)  # For reproducibility
num_spins <- 1:10
num_replications <- 1000

sample_means <- vector("list", length(num_spins))

for (i in num_spins) {
  spins <- matrix(runif(i * num_replications, min = 0, max = 1000), ncol = num_replicati
ons)
  sample_means[[i]] <- colMeans(spins)
}
```

c. Find the sample mean for each replication for each spin category. So you will have 1,000 sample means for experiment with 1 spin, 1,000 sample means for experiment with 2 spins, etc. . .

```
# Displaying the summary of sample means for each number of spins
summary_table <- data.frame(
  "Number of Spins" = num_spins,
  "Mean of Sample Means" = sapply(sample_means, mean),
  "SD of Sample Means" = sapply(sample_means, sd)
)

print(summary_table)
```

```
##    Number.of.Spins Mean.of.Sample.Means SD.of.Sample.Means
## 1                1             497.2778          287.48391
## 2                2             498.4301          195.17282
## 3                3             499.5494          164.28939
## 4                4             495.6769          143.71013
## 5                5             498.8182          129.49582
## 6                6             494.0427          118.97901
## 7                7             503.4911          108.65926
## 8                8             491.5707          103.72519
## 9                9             497.4321           96.65870
## 10              10             499.2804           93.37981
```

d. Find the mean of the sample means of each replication (sample).

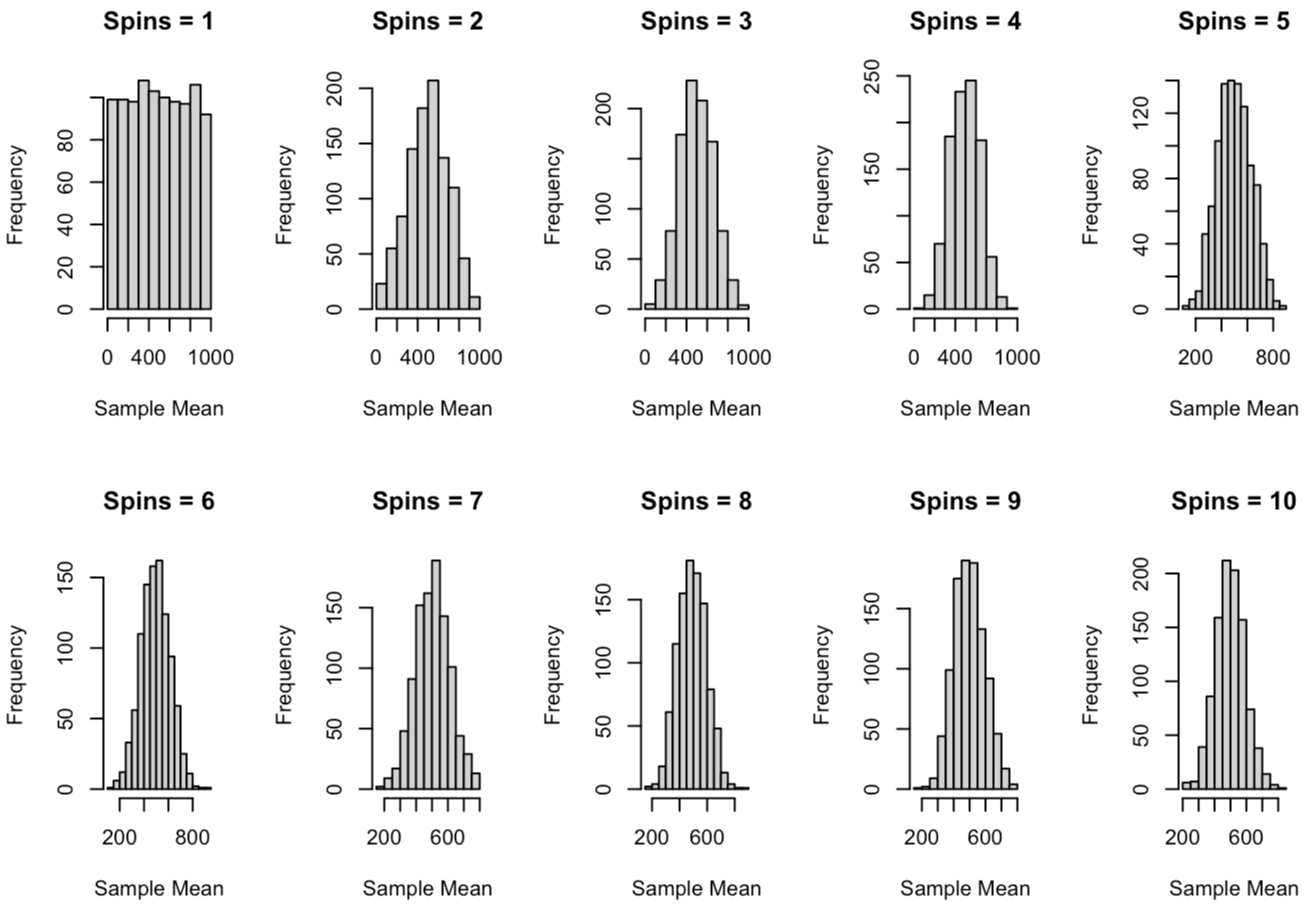
```
mean_of_sample_means <- sapply(sample_means, mean)
#print(mean_of_sample_means)
```

e. Find the standard deviation of sample means of each replication (sample).

```
std_dev_of_sample_means <- sapply(sample_means, sd)
#print(std_dev_of_sample_means)
```

f. Plot a histogram for each of the 10 categories of spins. So there will be 10 histograms – one for 1 spin (1000 samples each of size 1), one for 2 spins (1000 samples each of size 2), one for 3 spins (1000 samples each of size 3), etc. Comment on how the shape of the histogram changes with increasing the number of spins (increasing sample sizes).

```
par(mfrow = c(2, 5))

for (i in num_spins) {
  hist(sample_means[[i]], main = paste("Spins =", i), xlab = "Sample Mean")
}
```



g. Find the theoretical standard error for each category of spins.

```
theoretical_se <- theoretical_sd / sqrt(num_spins)
#print(theoretical_se)
```

h. Compare the theoretical mean with the mean of sample means found in part d above.

```
# Create a comparison data frame
comparison_mean <- data.frame(Spins = num_spins, Theoretical_Mean = rep(theoretical_mea
n, length(num_spins)), Mean_of_Sample_Means = mean_of_sample_means)

# Print the comparison data frame
print(comparison_mean)
```

```
##    Spins Theoretical_Mean Mean_of_Sample_Means
## 1      1              500             497.2778
## 2      2              500             498.4301
## 3      3              500             499.5494
## 4      4              500             495.6769
## 5      5              500             498.8182
## 6      6              500             494.0427
## 7      7              500             503.4911
## 8      8              500             491.5707
## 9      9              500             497.4321
## 10    10              500             499.2804
```

i. Compare the theoretical standard error (part g above) with the standard deviation of sample means (part e above).

```
# Create a comparison dataframe
comparison_se <- data.frame(Spins = num_spins, Theoretical_SE = theoretical_se, Std_Dev_
of_Sample_Means = std_dev_of_sample_means)

# Print comparison dataframe
print(comparison_se)
```

```
##    Spins Theoretical_SE Std_Dev_of_Sample_Means
## 1      1      288.67513               287.48391
## 2      2      204.12415               195.17282
## 3      3      166.66667               164.28939
## 4      4      144.33757               143.71013
## 5      5      129.09944               129.49582
## 6      6      117.85113               118.97901
## 7      7      109.10895               108.65926
## 8      8      102.06207               103.72519
## 9      9       96.22504                96.65870
## 10    10       91.28709                93.37981
```

j. Find the probability of winning more than $600 for each spin category. For example, find P (W inning > $600) with 1 spin, P (W inning > $600) with 2 spins, . . . , P (W inning > $600) with 10 spins.

```
# Find the probability of winning more than $600 for each spin category
prob_win_gt_600 <- sapply(sample_means, function(x) mean(x > 600))

# Print the probabilities
print(prob_win_gt_600)
```

```
##  [1] 0.393 0.304 0.278 0.251 0.229 0.193 0.187 0.146 0.159 0.131
```

k. Summarize the results in the table below and comment on what you observe. Spins (n) 1 2 3 4 5 6 7 8 9 10 Theoretical Mean Mean of Sample Means Theoretical Standard Error Standard Deviation of Sample Means P (winning > $600)

```
# Create summary dataframe
results_table <- data.frame(Spins = num_spins, Theoretical_Mean = theoretical_mean,
                            Mean_of_Sample_Means = mean_of_sample_means, Theoretical_SE
= theoretical_se,
                            Std_Dev_of_Sample_Means = std_dev_of_sample_means, Prob_Win_
Gt_600 = prob_win_gt_600)

# Print results
print(results_table)
```

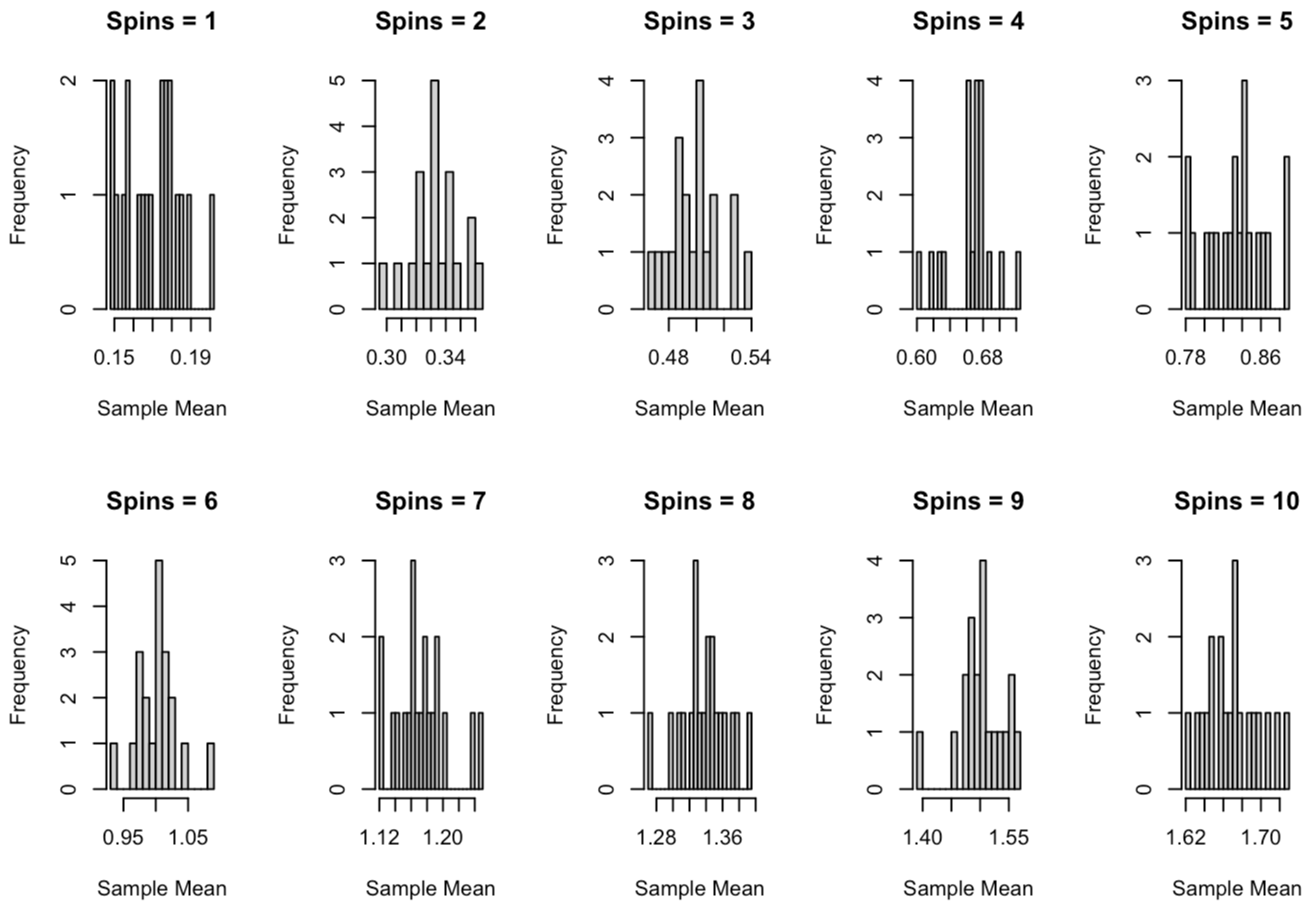
```
##    Spins Theoretical_Mean Mean_of_Sample_Means Theoretical_SE
## 1      1              500             497.2778      288.67513
## 2      2              500             498.4301      204.12415
## 3      3              500             499.5494      166.66667
## 4      4              500             495.6769      144.33757
## 5      5              500             498.8182      129.09944
## 6      6              500             494.0427      117.85113
## 7      7              500             503.4911      109.10895
## 8      8              500             491.5707      102.06207
## 9      9              500             497.4321       96.22504
## 10    10              500             499.2804       91.28709
##    Std_Dev_of_Sample_Means Prob_Win_Gt_600
## 1                287.48391           0.393
## 2                195.17282           0.304
## 3                164.28939           0.278
## 4                143.71013           0.251
## 5                129.49582           0.229
## 6                118.97901           0.193
## 7                108.65926           0.187
## 8                103.72519           0.146
## 9                 96.65870           0.159
## 10                93.37981           0.131
```

Question 17

Consider the simulation similar to the one in Question 16. Suppose now that the outcome of each spin is no longer uniformly distributed between $0 and $1,000. Instead, it is the number of 7's you get in 20 rolls of two dice. The simulation should still allow you to vary the number of "spins" from 1 to 10 and the "winnings" is still the average of the outcomes of spins.

a. Repeat parts a) through i) of the Question 16. Ignore part j - that means don't show part j from Question 16 for this question. However, recreate the summary table shown in part k of the question, except the P (winning > $600) row of the summary table. This means no need to find the P (winning > $600) due to its irrelevance in this question.

```r
set.seed(123)  # For reproducibility

# Define parameters
num_spins <- 1:10
num_replications <- 1000

# Function to simulate the number of 7's in 20 rolls of two dice
simulate_sevens <- function(spins, replications) {
  replicate(replications, {
    rolls <- matrix(sample(2:12, spins * 20, replace = TRUE, prob = c(1,2,3,4,5,6,5,4,3,2,1)/36), ncol = spins)
    rowSums(rolls == 7)
  })
}

# Simulate for different numbers of spins
sample_means <- lapply(num_spins, function(n) rowMeans(simulate_sevens(n, num_replications)))

# Mean of Sample Means
mean_of_sample_means <- sapply(sample_means, mean)

# Standard Deviation of Sample Means
std_dev_of_sample_means <- sapply(sample_means, sd)

# Plot Histograms
par(mfrow = c(2, 5))
for (i in num_spins) {
  hist(sample_means[[i]], main = paste("Spins =", i), xlab = "Sample Mean", breaks = 20)
}
```



```
# Theoretical Standard Error
theoretical_se <- sqrt((5/36) * (1 - 5/36)) / sqrt(num_spins)

# Comparison of Theoretical Mean and Mean of Sample Means
comparison_mean <- data.frame(Spins = num_spins, Theoretical_Mean = 7, Mean_of_Sample_Me
ans = mean_of_sample_means)
print(comparison_mean)
```

```
##    Spins Theoretical_Mean Mean_of_Sample_Means
## 1      1                7              0.17055
## 2      2                7              0.33430
## 3      3                7              0.49920
## 4      4                7              0.66605
## 5      5                7              0.83430
## 6      6                7              1.00215
## 7      7                7              1.17385
## 8      8                7              1.33925
## 9      9                7              1.50305
## 10    10                7              1.67040
```

```
# Comparison of Theoretical SE and SD of Sample Means
comparison_sd <- data.frame(Spins = num_spins, Theoretical_SE = theoretical_se, SD_of_Sa
mple_Means = std_dev_of_sample_means)
print(comparison_sd)
```

```
##    Spins Theoretical_SE SD_of_Sample_Means
## 1      1      0.3458305         0.01458361
## 2      2      0.2445391         0.01647997
## 3      3      0.1996654         0.01810670
## 4      4      0.1729153         0.02792561
## 5      5      0.1546601         0.03070933
## 6      6      0.1411847         0.03117063
## 7      7      0.1307117         0.03220375
## 8      8      0.1222696         0.02863909
## 9      9      0.1152768         0.03796047
## 10    10      0.1093612         0.02860696
```

```
# Summary Table
summary_table <- data.frame(
  Spins = num_spins,
  Theoretical_Mean = 7,
  Mean_of_Sample_Means = mean_of_sample_means,
  Theoretical_SE = theoretical_se,
  SD_of_Sample_Means = std_dev_of_sample_means
)

print(summary_table)
```

```
##    Spins Theoretical_Mean Mean_of_Sample_Means Theoretical_SE
## 1      1                7              0.17055      0.3458305
## 2      2                7              0.33430      0.2445391
## 3      3                7              0.49920      0.1996654
## 4      4                7              0.66605      0.1729153
## 5      5                7              0.83430      0.1546601
## 6      6                7              1.00215      0.1411847
## 7      7                7              1.17385      0.1307117
## 8      8                7              1.33925      0.1222696
## 9      9                7              1.50305      0.1152768
## 10    10                7              1.67040      0.1093612
##    SD_of_Sample_Means
## 1          0.01458361
## 2          0.01647997
## 3          0.01810670
## 4          0.02792561
## 5          0.03070933
## 6          0.03117063
## 7          0.03220375
## 8          0.02863909
## 9          0.03796047
## 10         0.02860696
```

b. What is fundamentally different (in this simulation) from the Question 16 simulation?

```
sprintf("One fundamental difference between this simulation and the simulation in 16 is
that the the mean distributions began somwhat normally distributed and stayed theat was
whereas the histograms in 16 began in a uniform distribution and became more normal afte
r eash spin.")
```

```
## [1] "One fundamental difference between this simulation and the simulation in 16 is t
hat the the mean distributions began somwhat normally distributed and stayed theat was w
hereas the histograms in 16 began in a uniform distribution and became more normal after
eash spin."
```

c. Does the central limit theorem still hold?

```
sprintf("In this example the CLT would still hold. Even though the results we are gettin
g are discrete and not normally distributed we are getting a normal distribution for the
sample means of each roll.")
```

```
## [1] "In this example the CLT would still hold. Even though the results we are getting
are discrete and not normally distributed we are getting a normal distribution for the s
ample means of each roll."
```

Question 18

Accudial Celestial is a company that assembles luxury wristwatches and then sells them to retailers throughout the United States. The watches are assembled at a plant with two assembly lines - Super and Premium. These lines are intended to be identical, but the Super line uses somewhat older equipment than the Premium and is typically less reliable. Historical data have shown that each watch coming off the Super line, independently of the others, is free of defects with probability 0.98. The similar probability for the Premium line is 0.99. Each line produces 500 watches per hour. As a data analyst, you need to answer the following questions posed by the production manager.

a. She wants to know how many defect-free watches each line is likely to produce in a given hour. Specifically, find the smallest integer k (for each line separately) such that you can be 99% sure that the line will not produce more than k defective watches in a given hour.

```
# Parameters for Super line
n_super <- 500
p_defect_super <- 0.02

# Parameters for Premium line
n_premium <- 500
p_defect_premium <- 0.01

# Finding the smallest integer k such that P(X <= k) >= 0.99
k_super <- qbinom(0.99, n_super, p_defect_super)
k_premium <- qbinom(0.99, n_premium, p_defect_premium)

cat(sprintf("For the Super line, we can be 99 percent sure that there will be no more th
an %d defective watches in a given hour.\n", k_super))
```

```
## For the Super line, we can be 99 percent sure that there will be no more than 18 defe
ctive watches in a given hour.
```

```
cat(sprintf("For the Premium line, we can be 99 percent sure that there will be no more
than %d defective watches in a given hour.\n", k_premium))
```

```
## For the Premium line, we can be 99 percent sure that there will be no more than 11 de
fective watches in a given hour.
```

b. Accudial Celestial currently has an order for 500 watches from an important customer. The company plans to fill this order by packing slightly more than 500 watches, all from the Premium line, and sending this package off to the customer. Obviously, the company wants to send as few watches as possible, but it wants to be 99% sure that when the customer opens the package, there are at least 500 defect-free watches. How many watches should be packed?

```
vector<- character(0)
for (n in 500:550){
  prob <- pbinom(499,n,.99,lower.tail=FALSE)
  vector <- c(vector,prob)
}
Number_Watches_Sent <- c(500:550)
data.frame(Number_Watches_Sent,vector)
```

```
##    Number_Watches_Sent             vector
## 1                  500 0.0065704830424146
## 2                  501 0.0394228982544876
## 3                  502  0.121718198360731
## 4                  503  0.259425667205177
## 5                  504  0.432592809277069
## 6                  505   0.60714528848554
## 7                  506   0.75406029181933
## 8                  507  0.860258851372044
## 9                  508  0.927562188488576
## 10                 509  0.965551183216575
## 11                 510  0.984887581533126
## 12                 511  0.993852638934436
## 13                 512   0.99767025921116
## 14                 513  0.999173814273994
## 15                 514  0.999724759807732
## 16                 515  0.999913550477293
## 17                 516  0.999974317474057
## 18                 517   0.99999276204484
## 19                 518  0.999998059735448
## 20                 519  0.999999504053204
## 21                 520  0.999999878853661
## 22                 521  0.999999971661394
## 23                 522  0.999999993639952
## 24                 523  0.999999998628129
## 25                 524  0.999999999715136
## 26                 525  0.999999999942973
## 27                 526  0.999999999988978
## 28                 527  0.999999999997941
## 29                 528  0.999999999999628
## 30                 529  0.999999999999935
## 31                 530  0.999999999999989
## 32                 531  0.999999999999998
## 33                 532                  1
## 34                 533                  1
## 35                 534                  1
## 36                 535                  1
## 37                 536                  1
## 38                 537                  1
## 39                 538                  1
## 40                 539                  1
## 41                 540                  1
## 42                 541                  1
## 43                 542                  1
## 44                 543                  1
## 45                 544                  1
## 46                 545                  1
## 47                 546                  1
## 48                 547                  1
## 49                 548                  1
## 50                549                  1
## 51                550                  1
```

```
sprintf("The first value of watches sent that gives us a 99 percent probability of recei
ving 500 defect-free watches is to send 511 watches per the table.")
```

```
## [1] "The first value of watches sent that gives us a 99 percent probability of receiv
ing 500 defect-free watches is to send 511 watches per the table."
```

c. Accudial Celetial has an order for 100 watches from another important customer. The customer has agreed to pay $50,000 for the order — that is, $500 per watch. If the company sends more than 100 watches to the customer, its revenue doesn't increase; it can never exceed $50,000. Its unit cost of producing a watch is $450, regardless of which line it is assembled on. The order will be filled entirely from a single line (first, consider the Premium line only), and the company plans to send slightly more than 100 watches to the customer. If the customer opens the shipment and finds that there are fewer than 100 defect-free watches (which we assume the customer has the ability to do), then he will pay only for the defect-free watches — Accudial Celestial's revenue will decrease by $500 per watch short of the 100 required — and on top of this, Accudial Celestial will be required to make up the difference at an expedited cost of $1000 per watch. The customer won't pay a dime for these expedited watches. You have been asked to develop a probability distribution model to find the company's expected profit for any number of watches it sends to the customer. How many watches should be sent from the Premium line to maximize the expected profit? Now repeat this part of the question for fulfilling the entire order from the Super line only. You can assume that Accudial Celestial will never send more than 110 watches irrespective of which line it chooses. It turns out that this large a shipment is not even close to optimal.

```
unit_cost = 450
selling_price = 500
additional_cost = 1000

results_premium <- data.frame(N = 100:110, Expected_Profit = numeric(11))

for (n in 100:110) {
  prob_defect_free <- 1 - pbinom(99, n, 0.99, lower.tail = TRUE)
  revenue = selling_price * n
  cost = unit_cost * n
  additional_cost_if_needed = ifelse(n < 100, (100 - n) * additional_cost, 0)

  expected_profit = (revenue - cost - additional_cost_if_needed) * prob_defect_free

  results_premium[n - 99, "Expected_Profit"] <- expected_profit
}

optimal_n_premium <- results_premium$N[which.max(results_premium$Expected_Profit)]

results_premium
```

```
##      N Expected_Profit
## 1  100        1830.162
## 2  101        3696.927
## 3  102        4676.246
## 4  103        5045.758
## 5  104        5178.899
## 6  105        5246.368
## 7  106        5299.455
## 8  107        5349.928
## 9  108        5399.991
## 10 109        5449.999
## 11 110        5500.000
```

```
optimal_n_premium
```

```
## [1] 110
```

```
results_super <- data.frame(N = 100:110, Expected_Profit = numeric(11))

for (n in 100:110) {
  prob_defect_free <- 1 - pbinom(99, n, 0.98, lower.tail = TRUE)
  revenue = selling_price * n
  cost = unit_cost * n
  additional_cost_if_needed = ifelse(n < 100, (100 - n) * additional_cost, 0)

  expected_profit = (revenue - cost - additional_cost_if_needed) * prob_defect_free

  results_super[n - 99, "Expected_Profit"] <- expected_profit
}

optimal_n_super <- results_super$N[which.max(results_super$Expected_Profit)]

results_super
```

```
##      N Expected_Profit
## 1  100        663.0978
## 2  101       2009.1863
## 3  102       3395.3259
## 4  103       4366.7694
## 5  104       4897.0064
## 6  105       5148.9863
## 7  106       5270.4199
## 8  107       5342.2732
## 9  108       5398.1767
## 10 109       5449.6073
## 11 110       5499.9221
```

```
optimal_n_super
```

```
## [1] 110
```