week4Assignment

Ajay

2025-04-30

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#Exercises #1. Suppose you poll a population in which a proportion p of voters are Democrats and 1−p are Republicans. Your sample size is N=25.Consider the random variable S which is the total number of Democrats in your sample. What is the expected value of this random variable? Hint: it’s a function of p .

# Set the sample size  
N <- 25 # Number of voters in the sample  
  
# Set the assumed proportion of Democrats in the population  
p <- 0.6 # You can change this value to any probability between 0 and 1  
  
# Calculate the expected value of the number of Democrats in the sample  
# Since this follows a Binomial distribution, E[S] = N \* p  
expected\_value <- N \* p  
cat("Expected number of Democrats (E[S]):", expected\_value, "\n")

## Expected number of Democrats (E[S]): 15

# Simulate one random sample using the binomial distribution  
# rbinom(n, size, prob) generates 'n' samples from Binomial(size, prob)  
set.seed(123) # Set seed for reproducibility of results  
sample <- rbinom(1, size = N, prob = p) # One random draw from Binomial(25, 0.6)  
  
# Print the number of Democrats found in this simulated sample  
cat("Simulated number of Democrats in one sample:", sample, "\n")

## Simulated number of Democrats in one sample: 16

#2. What is the standard error of S? Hint: it’s a function of p .

# Sample size  
N <- 25  
  
# Proportion of Democrats in the population  
p <- 0.6 # You can change this to other values like 0.5, 0.3, etc.  
  
# Calculate the standard error of S  
# SE = sqrt(N \* p \* (1 - p))  
standard\_error <- sqrt(N \* p \* (1 - p))  
  
# Print the result  
cat("Standard error of S:", standard\_error, "\n")

## Standard error of S: 2.44949

#3 Consider the random variable S/NThis is equivalent to the sample average, which we have been denoting as ¯X. What is the expected value of the ¯X ? Hint: it’s a function of p .

# Sample size  
N <- 25  
  
# True proportion of Democrats in the population  
p <- 0.6  
  
# Expected value of the sample proportion X̄ = S / N  
expected\_X\_bar <- p # Because E[S]/N = (N\*p)/N = p  
  
# Print the expected value of the sample average  
cat("Expected value of sample proportion (E[X̄]):", expected\_X\_bar, "\n")

## Expected value of sample proportion (E[X̄]): 0.6

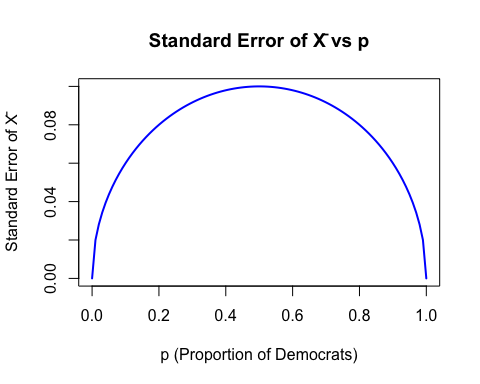
#4 What is the standard error of ¯X? Hint: it’s a function of p

# Sample size  
N <- 25  
  
# Proportion of Democrats in the population  
p <- 0.6 # Change as needed  
  
# Standard error of sample proportion X̄ = sqrt(p \* (1 - p) / N)  
se\_X\_bar <- sqrt(p \* (1 - p) / N)  
  
# Print the result  
cat("Standard error of sample proportion (SE[X̄]):", se\_X\_bar, "\n")

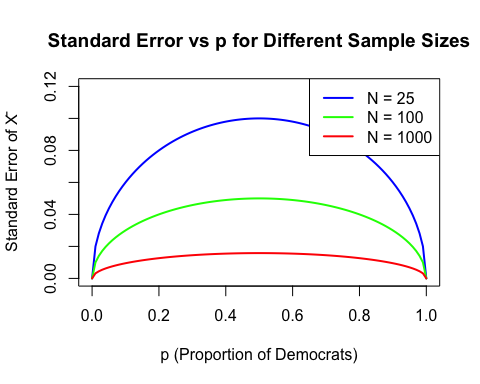
## Standard error of sample proportion (SE[X̄]): 0.09797959

#5Write a line of code that gives you the standard error se for the problem above for several values of, specifically for p <- seq(0, 1, length = 100). Make a plot of se versus p.

# Set sample size  
N <- 25  
  
# Create a sequence of p values from 0 to 1 (100 points)  
p <- seq(0, 1, length = 100)  
  
# Compute standard error for each p using the formula: sqrt(p \* (1 - p) / N)  
se <- sqrt(p \* (1 - p) / N)  
  
# Plot SE vs. p  
plot(p, se, type = "l", # Line plot  
 main = "Standard Error of X̄ vs p", # Title  
 xlab = "p (Proportion of Democrats)", # x-axis label  
 ylab = "Standard Error of X̄", # y-axis label  
 col = "blue", lwd = 2) # Line color and thickness

 #6. Copy the code above and put it inside a for-loop to make the plot for N=25 , N=100, and N=1000.

# Define the values of N to loop through  
N\_values <- c(25, 100, 1000)  
  
# Create a sequence of p values from 0 to 1  
p <- seq(0, 1, length = 100)  
  
# Set up the plot canvas  
plot(p, sqrt(p \* (1 - p) / N\_values[1]), type = "n", # Empty plot setup  
 ylim = c(0, 0.12), # Adjust y-axis to fit all curves  
 xlab = "p (Proportion of Democrats)",  
 ylab = "Standard Error of X̄",  
 main = "Standard Error vs p for Different Sample Sizes")  
  
# Define colors for each curve  
colors <- c("blue", "green", "red")  
  
# Loop through each N value and plot the SE curve  
for (i in 1:length(N\_values)) {  
 N <- N\_values[i]  
 se <- sqrt(p \* (1 - p) / N)  
 lines(p, se, col = colors[i], lwd = 2) # Add the line to the plot  
}  
  
# Add a legend to differentiate sample sizes  
legend("topright", legend = paste("N =", N\_values),  
 col = colors, lwd = 2)

 #7. If we are interested in the difference in proportions, our estimate is . Use the rules we learned about sums of random variables and scaled random variables to derive the expected value of d.

# Set parameters  
N <- 25 # Sample size  
p <- 0.6 # Proportion of Democrats  
B <- 10000 # Number of simulations  
  
# Simulate B samples of size N and compute sample proportions (X̄)  
X\_bar <- rbinom(B, N, p) / N  
  
# Compute d = 2 \* X̄ - 1  
d <- 2 \* X\_bar - 1  
  
# Print simulated and theoretical expected value  
cat("Simulated mean of d:", mean(d), "\n")

## Simulated mean of d: 0.201856

cat("Theoretical value (2p - 1):", 2 \* p - 1, "\n")

## Theoretical value (2p - 1): 0.2

#8. What is the standard error of d?

# Set parameters  
N <- 25  
p <- 0.6  
  
# Compute standard error of d = 2 \* SE(X̄)  
se\_d <- 2 \* sqrt(p \* (1 - p) / N)  
  
# Print result  
cat("Standard error of d:", se\_d, "\n")

## Standard error of d: 0.1959592

#9. If the actual p=.45, it means the Republicans are winning by a relatively large margin since d=−.1, which is a 10% margin of victory. In this case, what is the standard error of 2^X−1 If we take a sample of N=25?

# Given values  
p <- 0.45  
N <- 25  
  
# Compute standard error of d = 2 \* SE(X̄)  
se\_d <- 2 \* sqrt(p \* (1 - p) / N)  
  
# Print the result  
cat("Standard error of d when p = 0.45 and N = 25:", se\_d, "\n")

## Standard error of d when p = 0.45 and N = 25: 0.1989975

#10 Given the answer to 9, which of the following best describes your strategy of using a sample size of N=25?

# Set true proportion  
p <- 0.45  
  
# Define a vector of sample sizes to test  
N\_values <- c(25, 50, 100, 200, 500, 1000)  
  
# Calculate standard error for each N  
se\_d\_values <- 2 \* sqrt(p \* (1 - p) / N\_values)  
  
# Print result  
data.frame(Sample\_Size = N\_values, Standard\_Error\_d = round(se\_d\_values, 4))

## Sample\_Size Standard\_Error\_d  
## 1 25 0.1990  
## 2 50 0.1407  
## 3 100 0.0995  
## 4 200 0.0704  
## 5 500 0.0445  
## 6 1000 0.0315

#The correct choice is:  
#Our standard error is larger than the difference, so the chances of 2𝑋ˉ−1. 2X−1 being positive and throwing us off were not that small. We should pick a larger sample size.