Plotting Exercises

Christopher Wong

Exercise 1

Central Limit Theorem graphically

We will prove that the Central Limit Theorem (CLT) holds well for $n \geq 30$ graphically.

- a.) Take samples of size **m** from a distribution of your choice (you choose the parameters as well).
- b.) Find the mean of your random sample.
- c.) Write a loop that randomly samples **n times** from your distribution and calculates the sample mean.
- d.) Plot a histogram of your sample means.
- e.) Hopefully you have written a function that does all of the above for you. Now vary the values of **m** and **n** and compare the resulting plots:
  - m = 5, n = 10
  - m = 5, n = 30
  - m = 5, n = 100
  - m = 15, n = 20
  - m = 15, n = 50
  - m = 15, n = 100
  - m = 50, n = 10
  - m = 50, n = 100
  - m = 50, n = 1000

Optional: Plot a normal density curve around your histogram. What do you notice?

Exercise 2

What is a confidence interval graphically

We will graphically show what confidence actually is (using $\alpha = 0.05$)

- a.) Take samples of size **m** from a normal distribution (you choose the mean and standard deviation as well as well).
- b.) Calculate the error of the confidence interval.
- c.) Calculate the confidence interval.

$ConfidenceInteral = \bar{X} \pm Z_{1-\frac{\alpha}{2}} \times \frac{\sigma}{\sqrt{m}}$

- d.) Write a loop that randomly samples **n times** from the distribution and calculates the confidence intervals.
- e.) Combine everything into one function that calculates confidence intervals from 4 variables: *n, vector.length, TrueMean, StandardDeviation.*
- f.) Use the function provided below which takes in the data frame output your confidence interval function. Does it match your knowledge of what confidence intervals are?

```
plot.intervals = function(data){
  lower = min(data) - (max(data)-min(data)) * 0.5
  upper = max(data) + (max(data)-min(data)) * 0.25
  plot(1, TrueMean, type = "n", xlab = "Index", ylab = "Confidence Intervals", xlim = c(0, ncol(data)),
  abline(h = TrueMean)
  Result = sapply(1:ncol(data), function(i){
    if(data[1, i] <= TrueMean && TrueMean <= data[2, i]){
      points(i, data[1, i], col = "green", pch = 16)
      points(i, data[2, i], col = "green", pch = 16)
      segments(x0 = i, y0 = data[1, i], x1 = i, y1 = data[2, i], col = "green")
    }
    else{
      points(i, data[1, i], col = "red", pch = 16)
      points(i, data[2, i], col = "red", pch = 16)
      segments(x0 = i, y0 = data[1, i], x1 = i, y1 = data[2, i], col = "red")
    }
    return("")
  })
  Result2 = sapply(1:ncol(data), function(i){
    prop.vector = data[1, i] <= TrueMean && TrueMean <= data[2, i]
    return(prop.vector)
  })
  prop.captured = paste0(signif((mean(Result2) * 100), digits = 6),
                         "% of intervals capture true mean")
  legend(x = "bottomleft", legend = c("Captures true mean", "Does not capture true mean",
        prop.captured), pch = 15, col = c("green", "red", "transparent"), cex = 1.3)
}
```