STAT 630: Homework 4

Due: October 12th, 2023

1. Simulate the M&M activity in R. Let the true proportion of blue M&M's be p = 0.19. Suppose there are three different sized bags of M&M's: n = 25, 50, and 100. Use the `rbinom()` function to create 1,000 different samples for each of the three different sample sizes (n). Show your code below. You do not need to print any output.

```
p <- 0.19
n <-c(25,50,100)
n_samp <- 1000
samples_sz_25 <-rbinom(n_samp, 25, p)
samples_sz_50 <-rbinom(n_samp, 50, p)
samples_sz_100 <-rbinom(n_samp, 100, p)
```

2. Compute the mean and standard deviation of the sampling distribution for each sample size based on your 1000 simulations. Additionally, calculate the theoretical mean and standard deviation for each size based on the CLT for proportions. Put the information in the table below. You can use this table or create your own. If you choose to use this table, note that the vertical bars | must be lined up.

	Simulation	Theoretical
n = 25	$\bar{x}_{\hat{p}_{25}} =$	$\mu_{\hat{p}_{25}} =$
	$s_{\hat{p}_{25}} =$	$\sigma_{\hat{p}_{25}} =$
n = 50	$\bar{x}_{\hat{p}_{50}} =$	$\mu_{\hat{p}_{50}} =$
	$s_{\hat{p}_{50}} =$	$\sigma_{\hat{p}_{50}} =$
n = 100	$\bar{x}_{\hat{p}_{100}} =$	$\mu_{\hat{p}_{100}} =$
	$s_{\hat{p}_{100}} =$	$\sigma_{\hat{p}_{100}} =$

Calculate mean and standard deviation of the sampling distribution for each sample size based on 1000 simulations

```
xbar_sz_25 <- mean(samples_sz_25)
sd_sz_25 <- sd(samples_sz_25)

xbar_sz_50 <- mean(samples_sz_50)
sd_sz_50 <- sd(samples_sz_50)

xbar_sz_100 <- mean(samples_sz_100)
sd_sz_100 <- sd(samples_sz_100)
```

Calculate the theoretical mean and standard deviation for each size based on the CLT for proportions.

For theoretical mean (μ), the mean of the sample means (μ) is equal to the population proportion (p) = 0.19.

```
sd_theo <- numeric(length(n))

for (i in seq_along(n)) {
  sd_theo[i] <- sqrt((p * (1 - p)) / n[i])
}
```

```
results_df_sd <- data.frame(Sample_Size = n, mu = p, Sd = sd_theo)
results_df_sd
```

```
##   Sample_Size   mu         Sd
## 1          25 0.19 0.07846018
## 2          50 0.19 0.05547973
## 3         100 0.19 0.03923009
```

3. Compare $\sigma_{\hat{p}_{25}}$ with $\sigma_{\hat{p}_{50}}$. Compare $\sigma_{\hat{p}_{25}}$ with $\sigma_{\hat{p}_{100}}$. How much do they differ by?

$\sigma_{\hat{p}_{25}}$ is greater than $\sigma_{\hat{p}_{50}}$

$\sigma_{\hat{p}_{25}}$ - $\sigma_{\hat{p}_{50}}$

```
sd_theo[25] <- sqrt((p * (1 - p)) / 25)
sd_theo[25]
```

```
## [1] 0.07846018
```

```
sd_theo[50] <- sqrt((p * (1 - p)) / 50)
sd_theo[50]
```

```
## [1] 0.05547973
```

```
m = sqrt((p * (1 - p)) / 25) - sqrt((p * (1 - p)) / 50)
m
```

```
## [1] 0.02298045
```

```
sd_theo[100] <- sqrt((p * (1 - p)) / 100)
sd_theo[100]
```

```
## [1] 0.03923009
```

```
q = sqrt((p * (1 - p)) / 25) - sqrt((p * (1 - p)) / 100)
q
```

```
## [1] 0.03923009
```

4. Suppose that out of all Olympic athletes, 70% of them train for more than 40 hours per week. Suppose a researcher took a sample of 250 athletes.

    a. What proportion of the sample would be expected to train for more than 40 hours per week?

70% of the athletes were trained: p = 0.7

sample size: n = 250 athletes.

Number of athletes were trained for more than 40 hrs per week of the sample: 175 athletes

proportion of the sample would be expected to train for more than 40 hours per week: 175/250 = 0.7

```
p_bar <- 0.7  # Probability of an athlete training for more than 40 hours per week
n_samp_sz <- 250  # Sample size

# Calculate the expected value
x_bar <- n_samp_sz * p_bar
x_bar
```

```
## [1] 175
```

```
prop_samp <- x_bar/n_samp_sz
prop_samp
```

```
## [1] 0.7
```

b. What is the sampling distribution of the sample proportion?

The sampling distribution of the sample proportion is the distribution of sample proportions (or sample percentages) that you would get if you took random samples of a given size from a population, and calculated the proportion of each sample.

Reference: Sampling Distribution of the Sample Proportion, p-hat. (n.d.). https://bolt.mph.ufl.edu/6050-6052/module-9/sampling-distribution-of-p-hat/#:~:text=is%20described%20next.-,The%20Sampling%20Distribution%20of%20the%20Sample%20Proportion,the%20population%20proportion%20(p).

For example, if you were interested in the proportion of a population that use Instagram, you could not survey all of the people in the world or in a certain country to ask, so you take multiple random samples of a given place, calculate the proportion of each sample that supports the candidate, and observe the distribution of these proportions.

c. What is the probability that more than 35% of our sampled athletes train for more than 40 hours per w

sample proportion: p_hat : 35%

P_35 is denoted as $P_(\hat{p} \leq 35)$

P is denoted as $P_(\hat{p} > 35)$

The probability that more than 35% of our sampled athletes train for more than 40 hours per week: $P_(\hat{p} > 35) = 1 - P_(\hat{p} \leq 35)$

```r
p_bar = 0.7

n_samp_sz = 250

p_prob <- 0.35

P_35 <- round(pnorm(0.35), 3)

P_35
```

```
## [1] 0.637
```

```r
P_more_than_35 <- 1 - P_35

P_more_than_35
```

```
## [1] 0.363
```

d. Calculate a 90% confidence interval for the true proportion of Olympic athletes who train more than 4

Confidence level = 1 - alpha = 0.9

Alpha = 1 - 0.9 = 0.1

Z_star =

```r
alpha <- 1-0.9
z_star <- qnorm(1-alpha/2)
z_star
```

```
## [1] 1.644854
```

```r
p_bar <- 0.7
n_samp_sz <- 250
```

```r
ci_low <- p_bar - z_star * sqrt(p_bar*(1-p_bar)/n_samp_sz)
ci_low
```

```
## [1] 0.6523276
```

```r
ci_high <- p_bar + z_star * sqrt(p_bar*(1-p_bar)/n_samp_sz)
ci_high
```

```
## [1] 0.7476724
```

```r
qnorm(0.95)
```

```
## [1] 1.644854
```

5. Simulating confidence intervals

    a. Compute 1000 confidence intervals for each of the three simulation settings from Question 1.

Compute 95% confidence intervals

```r
# p <- 0.19
#
# n_samp <- 1000

phats_25 <- rbinom(1000, 25, 0.19) / 25
length(phats_25)
```

```
## [1] 1000
```

```r
ci_low_25 <- phats_25 - qnorm(.975) * sqrt(phats_25*(1 - phats_25)/25)

ci_high_25 <- phats_25 + qnorm(.975) * sqrt(phats_25*(1-phats_25)/25)

phats_50 <- rbinom(1000, 50, 0.19) / 50
ci_low_50 <- phats_50 - qnorm(.975) * sqrt(phats_50*(1 - phats_50)/50)
ci_high_50 <- phats_50 + qnorm(.975) * sqrt(phats_50*(1-phats_50)/50)

phats_100 <- rbinom(1000, 100, 0.19) / 100
ci_low_100 <- phats_100 - qnorm(.975) * sqrt(phats_100*(1 - phats_100)/100)
ci_high_100 <- phats_100 + qnorm(.975) * sqrt(phats_100*(1-phats_100)/100)
```

b. Plot the confidence intervals for each simulation setting. Include a line to mark true proportion. Color the confidence interval one color if it includes the true proportion and another color if it does not include the true proportion.

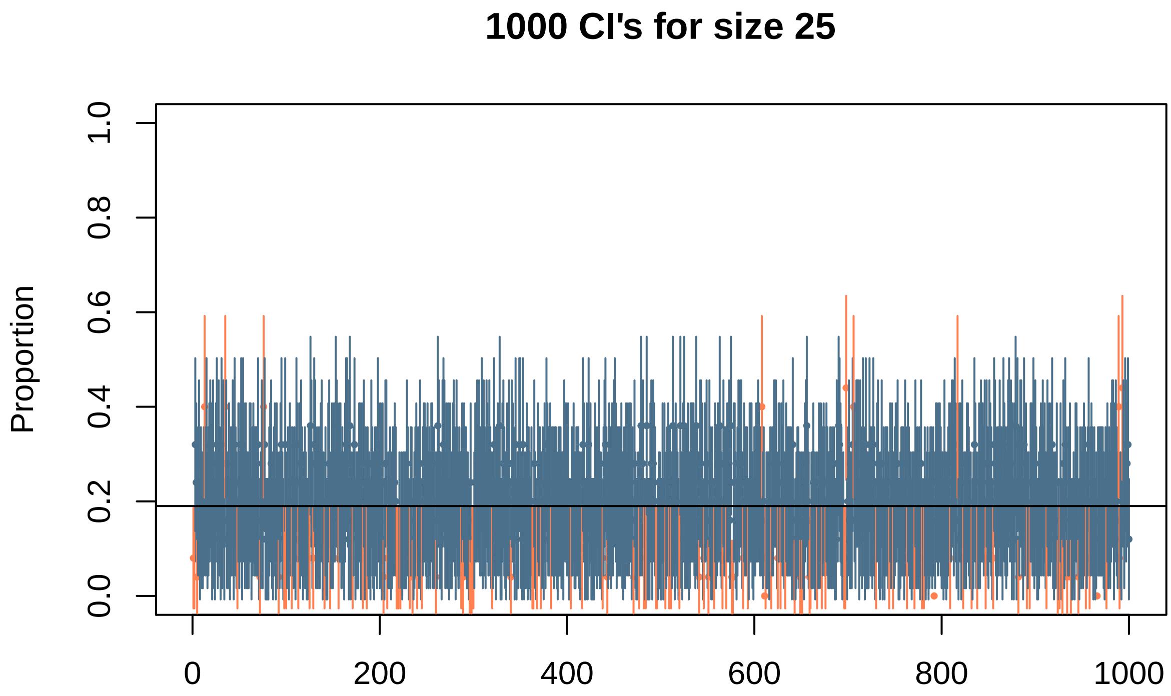
```r
color <- rep(NA, n_samp)

# Plot the confidence intervals for simulation of size 25

for(i in 1:n_samp){
   if(is.na(ci_low_25[i]) == FALSE & p > ci_low_25[i] & is.na(ci_high_25[i]) == FALSE & p < ci_high_25[

    color[i] <- "skyblue4"
  }
  else color[i] <- "coral"
}
table(color)
```

```
## color
##    coral skyblue4
##      120      880
```

```
x <- 1:n_samp
print(length(x))
```

```
## [1] 1000
```

```
plot(x, phats_25, ylim = c(0,1),
     pch = 16, cex = 0.5, col = color,
     main = "1000 CI's for size 25",
     xlab = "",
     ylab = "Proportion")
segments(x, ci_low_25, x, ci_high_25,
         col = color)
abline(h = p, col = "black")
```



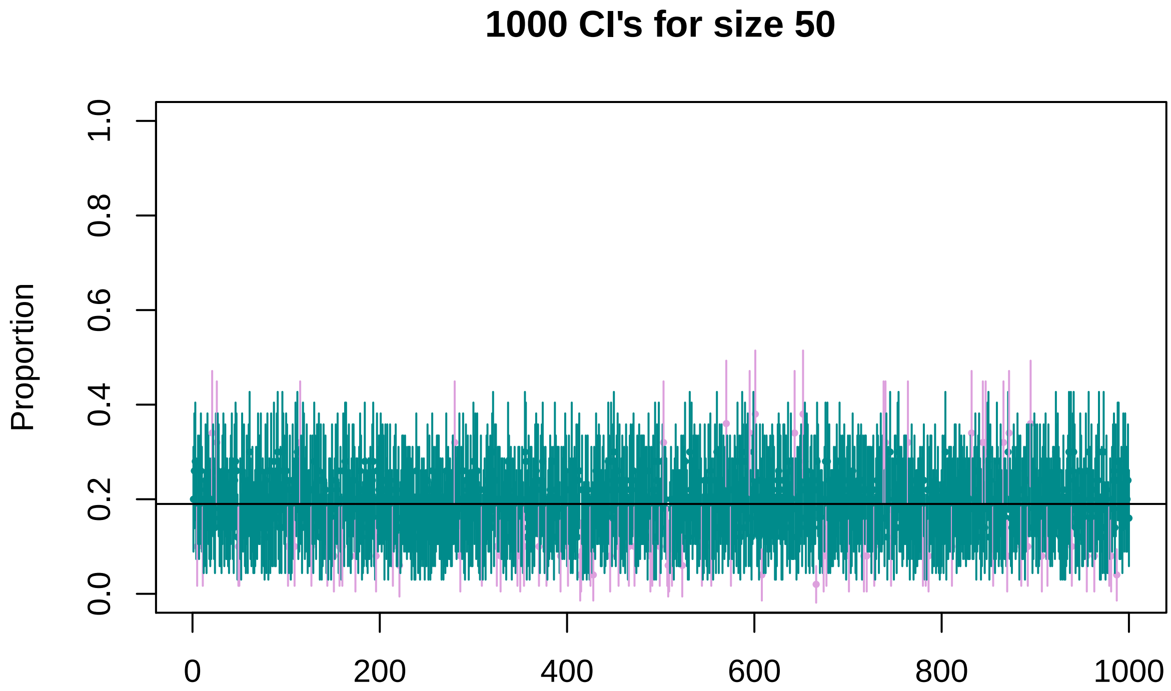
```
# Plot the confidence intervals for simulation of size 50

for(i in 1:n_samp){

  if(is.na(ci_low_50[i]) == FALSE & p > ci_low_50[i] & is.na(ci_high_50[i]) == FALSE & p < ci_high_50[i

    color[i] <- "cyan4"
  }
  else color[i] <- "plum"
}
table(color)
```

```
## color
## cyan4  plum
##   911    89
```

```
x <- 1:n_samp
print(length(x))
```

```
## [1] 1000
```

```
plot(x, phats_50, ylim = c(0,1),
     pch = 16, cex = 0.5, col = color,
     main = "1000 CI's for size 50",
     xlab = "",
     ylab = "Proportion")
segments(x, ci_low_50, x, ci_high_50,
         col = color)
abline(h = p, col = "black")
```



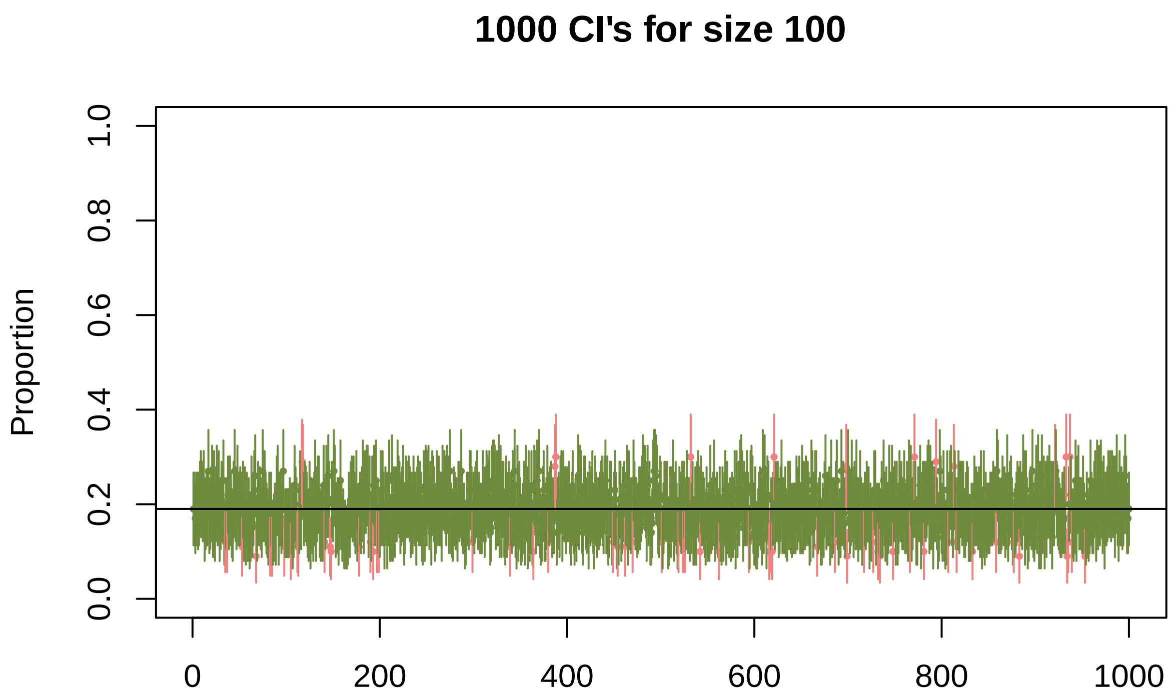
```
# Plot the confidence intervals for simulation of size 100

for(i in 1:n_samp){
    if(is.na(ci_low_100[i]) == FALSE & p > ci_low_100[i] & is.na(ci_high_100[i]) == FALSE & p < ci_high_

    color[i] <- "darkolivegreen4"
  }
  else color[i] <- "lightcoral"
}
table(color)
```

```
## color
## darkolivegreen4      lightcoral
##              931              69
```

```
x <- 1:n_samp
print(length(x))
```

```
## [1] 1000
```

```
plot(x, phats_100, ylim = c(0,1),
     pch = 16, cex = 0.5, col = color,
     main = "1000 CI's for size 100",
     xlab = "",
     ylab = "Proportion")
segments(x, ci_low_100, x, ci_high_100,
         col = color)
abline(h = p, col = "black")
```



```
c. For each simulation setting, calculate how many confidence intervals contained the true proportion.
```

A is denoted as the number of confidence intervals of sample size 25 contained the true proportion B is denoted as the number of confidence intervals of sample size 50 contained the true proportion C is denoted as the number of confidence intervals of sample size 100 contained the true proportion

```
p <- 0.19

#the number of confidence intervals of sample size 25 simulation contained the true proportion
A <- sum(ci_low_25 <= p & ci_high_25 >= p)
A
```

```
## [1] 880
```

```r
#the number of confidence intervals of sample size 50 simulation contained the true proportion
B <- sum(ci_low_50 <= p & ci_high_50 >= p)
B
```

```
## [1] 911
```

```r
#the number of confidence intervals of sample size 100 simulation contained the true proportion
C <- sum(ci_low_100 <= p & ci_high_100 >= p)
C
```

```
## [1] 931
```