R Notebook

Q1. From the dataset heights in the dslabs package, please describe the distribution of male and female heights. If you pick a female at random, what is the probability that she is 61 inches or shorter?

library(tidyr)  
library(dslabs)  
data(heights)  
library(cowplot)

##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Note: As of version 1.0.0, cowplot does not change the

## default ggplot2 theme anymore. To recover the previous

## behavior, execute:  
## theme\_set(theme\_cowplot())

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#남자, 여자의 키를 분류하고 따로 저장합니다.  
#각각의 평균도 저장합니다.  
heights\_w <- heights %>% filter(sex == "Female")

## Error in filter(., sex == "Female"): object 'sex' not found

p\_w <- mean(heights\_w$height)

## Error in mean(heights\_w$height): object 'heights\_w' not found

heights\_m <- heights %>% filter(sex == "Male")

## Error in filter(., sex == "Male"): object 'sex' not found

p\_m <- mean(heights\_m$height)

## Error in mean(heights\_m$height): object 'heights\_m' not found

#여자 키 plot  
plot\_w <- heights\_w %>% ggplot() +  
 geom\_histogram(aes(x = height), binwidth = 1, color = "black", fill = "grey") +  
 geom\_vline(xintercept = p\_w, color = "red", size=1)

## Error in eval(lhs, parent, parent): object 'heights\_w' not found

#남자 키 plot  
plot\_m <- heights\_m %>% ggplot() +  
 geom\_histogram(aes(x = height), binwidth = 1, color = "black", fill = "grey") +  
 geom\_vline(xintercept = p\_m, color = "red", size=1)

## Error in eval(lhs, parent, parent): object 'heights\_m' not found

plot\_grid(plot\_w, plot\_m)

## Error in plot\_grid(plot\_w, plot\_m): object 'plot\_w' not found

< 남자, 여자 모두 평균을 중심으로 뒤집은 종 모양으로 분포해 있음을 알 수 있습니다. >

# the probability that a female I picked at random is 61 inches or shorter

w\_mean <- mean(heights\_w$height)

## Error in mean(heights\_w$height): object 'heights\_w' not found

w\_sd <- sd(heights\_w$height)

## Error in is.data.frame(x): object 'heights\_w' not found

pnorm(61, w\_mean, w\_sd) \* 100

## Error in pnorm(61, w\_mean, w\_sd): object 'w\_mean' not found

**14.74261 %**

< 여자의 키 평균과 standard deviation을 구한 후, pnorm을 이용해 61인치보다 작거나 같을 확률을 구합니다. 그리고 100을 곱해 %으로 치환합니다. >

Q2. For American Roulette, there are 19 reds, 16 blacks and 3 greens. The payout for winning on green is 15 dollars. You create a random variable that is the sum of your winnings after betting on green 1000 times. Start your code by setting the seed to 1. Describe your random variable (e.g. the expected value, the standard error). Then, create a Monte Carlo simulation that generates 1,000 outcomes of a random variable, and then describe your result.

red 19개, black 16개, green 3개. green이 15달러. 1000번 배팅. sum 이 변수. 1000개 outcomes.

#green이 나올 확률  
p <- 3 / ( 19 + 16 + 3 )  
  
B <- 1000  
n <- 1000  
  
#EX  
n\*(p\*15 + (1-p)\*(-1))

## [1] 263.1579

#SE  
sqrt(n)\*abs(15-(-1))\*sqrt(p\*(1-p))

## [1] 136.4366

#번 돈의 sum  
set.seed(1)  
bbq <- function(){  
 draws <- sample(c(15, -1), n, prob = c(p, 1-p) , replace = TRUE)  
 sum(draws)  
}  
outcomes <- replicate( B, bbq() )  
  
#번 돈의 평균  
o\_mean <- mean(outcomes)  
  
d\_outcomes <- data.frame(outcomes)  
  
d\_outcomes %>% ggplot() +  
 geom\_histogram(aes(x=outcomes), binwidth = 15, color = "black", fill = "grey") +  
 geom\_vline(xintercept = o\_mean, color = "red", size=1)

## Error in ggplot(.): could not find function "ggplot"

expected value : 263.1579 standard error : 136.4366

< 번 돈, outcome이 평균을 중심으로 뒤집은 종 모양으로 분포해 있음을 알 수 있습니다. 분포가 종 모양에 정확히 들어맞지는 않으나, 가운데가 가장 높고 양 말단으로 갈수록 count가 줄어드는 것을 볼 수 있습니다. > < 분포를 알아보기 쉬운 히스토그램을 사용했습니다.)

Q3. From the poll example, we will create a Monte Carlo simulation for p = 0.45. You will compare the sampling size (N) for 10, 1000, and the repeat size (B) for 100, 10000. So you should have four combinations (10 N x 100 B, 1000 N x 100 B, 10 N x 10000 B, 1000 N x 10000 B). Please describe your Monte Carlo simulation results, and compare four combinations.

N이 10, 1000일 때 B이 100, 10000일 때

p <- 0.45  
  
# N과 B를 대입할 함수 만듭니다.  
grape <- function(N, B){  
 d <- replicate(B, {  
 x <- sample(c(1, 0), size = N, prob = c(p, 1-p), replace = TRUE)  
 x\_hat <- mean(x)  
 se\_hat <- sd(x)/sqrt(N)  
 return(c(x\_hat, se\_hat))  
})  
d <- t(d)  
d <- as.data.frame(d) %>% mutate(n = seq(1,B)) %>% rename(x\_hat = V1, se\_hat = V2) %>%  
 mutate(start = x\_hat - 1.96 \* se\_hat, end = x\_hat + 1.96 \* se\_hat) %>%  
 mutate(hit\_inside = ifelse(start<p & end>p, "YES", "NO"))  
 p <- d[1:100,] %>% ggplot(aes(x=x\_hat, y=n, color = hit\_inside)) + geom\_point() +  
 geom\_errorbarh(aes(xmax = start, xmin = end))  
 return(p)  
}

#N 10, B 100  
grape(10, 100)

## Error in mutate(., n = seq(1, B)): could not find function "mutate"

< confidence interval에 들어가지 못한 것이 6 개 보입니다.)

#N 10, B 10000  
grape(10, 10000)

## Error in mutate(., n = seq(1, B)): could not find function "mutate"

<confidence interval에 들어가지 못한 것이 4개 보입니다.>

#N 1000, B 100  
grape(1000, 100)

## Error in mutate(., n = seq(1, B)): could not find function "mutate"

<confidence interval에 들어가지 못한 것이 5개 보입니다.>

#N 1000, B 10000  
grape(1000, 10000)

## Error in mutate(., n = seq(1, B)): could not find function "mutate"

<confidence interval에 들어가지 못한 것이 4개 보입니다.>

p가 들어가는 confidence interval을 만들 확률이 N과 B가 클수록 크다는 것을 알 수 있습니다.

N과 B가 둘 다 작을 때 error가 가장 많이 일어났습니다.

geom\_errorbarh를 사용해, error가 난 것과 나지 않은 것을 보기 쉽게 구분했습니다.