Question 3

Joseph Froelicher

10/1/2020

# Part A

n <- 5  
mu <- 0  
sd <- 75  
alpha <- 0.05  
iter <- 10000  
t\_mat <- matrix(NA, iter, 3)  
colnames(t\_mat) <- c('test.statistic', 'p.value', 'reject')  
  
set.seed(2345)  
  
for (i in 1:iter) {  
 vals <- rnorm(n, mu, sd)  
  
 t\_result <- t.test(  
 x = vals,  
 alternative = "two.sided",  
 mu = mu,  
 conf.level = 1 - alpha  
 )  
   
 t\_mat[i, 1] <- t\_result$statistic  
 t\_mat[i, 2] <- t\_result$p.value  
   
 if (t\_result$p.value < alpha) {  
 t\_mat[i, 3] <- 1  
 } else {  
 t\_mat[i, 3] <- 0  
 }  
}  
  
(1 - sum(t\_mat[,3]) / iter) \* 100

## [1] 95.02

Simulating a one-sample t-test 10,000 times, under the null hypothesis, with a mean of 0 and a standard deviation of 75, we can see that power of rougly 95% is obtained. This supports our claim that the null hypothesis of mean equal to 0 will in fact reject about 5% of the time.

# Part B

# function definition  
simNull <- function(n, mu, sd, alpha, iter) {  
 power <- vector("double", iter)  
 set.seed(2345)  
   
 for (i in 1:iter) {  
 vals <- rnorm(n, mu, sd)  
 z\_alpha <- qnorm(1 - (alpha / 2))  
 delta <- abs(mu - mean(vals))  
 se <- sd / sqrt(n)  
 power[i] <- (delta / se) - z\_alpha  
 }  
   
 mean\_power <- (1 - pnorm(mean(power))) \* 100  
 median\_power <- (1 - pnorm(median(power))) \* 100  
   
 return(data.frame('power mean' = mean\_power, 'power median' = median\_power))  
}  
  
# testing  
simNull(5, 0, 75, 0.05, 10000)

## power.mean power.median  
## 1 87.68859 90.11938

# Part C

mu0 <- 0  
mu1 <- 100  
sd <- 75  
n <- 5  
alpha <- 0.05  
iter <- 10000  
power\_vec <- vector("double", iter)  
  
set.seed(1796)  
  
for (i in 1:iter) {  
 vals0 <- rnorm(n, mu0, sd)  
 vals1 <- rnorm(n, mu1, sd)  
   
 p <- power.t.test(n, mean(vals0) - mean(vals1), sd, alpha, power = NULL, type='one.sample')  
 power\_vec[i] <- p$power  
}  
  
mean(power\_vec)

## [1] 0.5805492

median(power\_vec)

## [1] 0.6222256

# Part

Simulating a one-sample t-test 10,000 times, under the alternative hypothesis, with a target mean of difference of 100 and a standard deviation of 75, we would assume a power level of roughly 60%.

# Part D

# function definition  
simAlternative <- function(n, mu0, mu1, sd, alpha, iter) {  
 power <- vector("double", iter)  
 set.seed(8675309)  
   
 for (i in 1:iter) {  
 vals0 <- rnorm(n, mu0, sd)  
 vals1 <- rnorm(n, mu1, sd)  
   
 z\_alpha <- qnorm(1 - (alpha / 2))  
 delta <- abs(mean(vals0) - mean(vals1))  
 se <- sd / sqrt(n)  
 power[i] <- (delta / se) - z\_alpha  
 }  
   
 mean\_power <- (1 - pnorm(mean(power))) \* 100  
 median\_power <- (1 - pnorm(median(power))) \* 100  
   
 return(data.frame('power mean' = mean\_power, 'power median' = median\_power))  
}  
  
# testing  
simAlternative(5, 0, 100, 75, 0.05, 10000)

## power.mean power.median  
## 1 15.14336 15.76701

# Part E

The answer from Exercise 2, Part A, unknown standard deviation, matches very cloesly with our answer from scenario 2 for power. We would expect our calculation using power.t.test under the alternative hypothesis of mean difference 100 to look very similar to a 10,000 iteration simulation with random values that have mean differece of ‘approximately’ or ‘simulated’ 100.