BIOS6611-Homework4

Randy

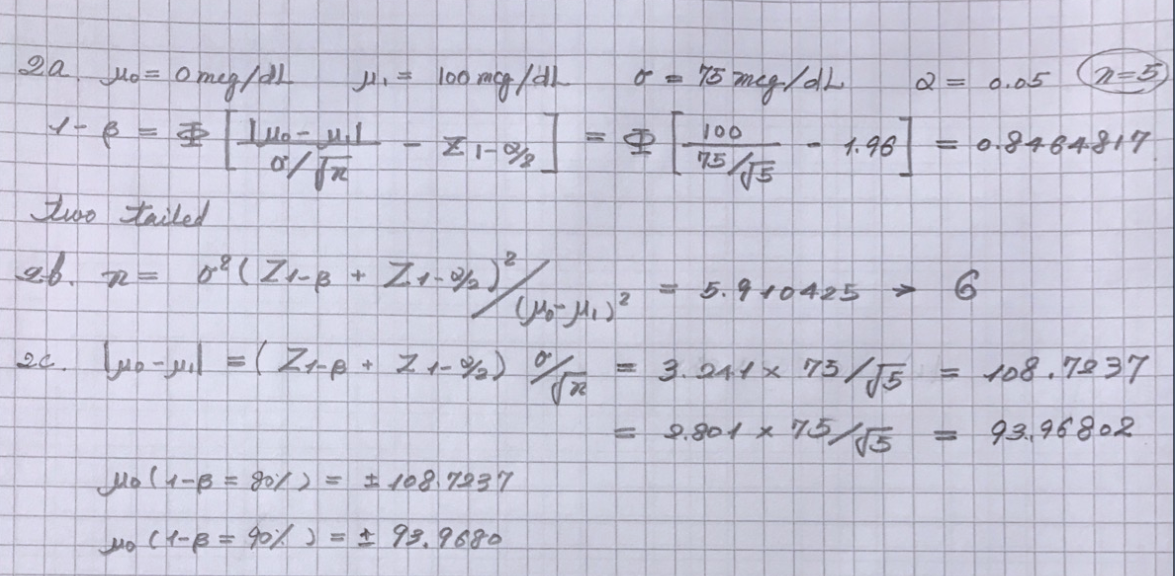
9/25/2019

### 1

Per the famous saying, “Absence of evidence is not evidence of absence” The lack of positive evidence might just an ignorance or failure to prove the existence of statistical significance. That is so called “the dilemma of the nonrejected null hypothesis”. A lot of researchers using the inappropriate ways to test the post-hoc power tests, to misinterpreting the statistical results for potential justification of the test. Here are several reasons for PAPs (power approach paradoxes):

When there is a nonrejected null hypothesis, normally first the power of the test will be computed, sometimes it is also referred as the “observed power.” But the p-value also determines the observed power. If the test results show up as nonsignificant, then the observed power will always be low (precisely less than 0.5). Moreover, the higher post hoc power does not imply stronger evidence for a nonrejected null hypothesis. Second, hypothetical difference will be calculated based pre-assumed particular higher power, which is setup as the upper bound of the true effect size. The conjectured higher power, with similar sample size, will always prefer a smaller variance, in another words, smaller obtained effect size. So, in conclusion, both of the suggestions are superfluous in the post power calculation with no contribution to correct the fundamental PAPs.

### 2a



findPower <- function(diff, sd, n, alpha){  
z.alpha <- qnorm(1 - (alpha/2))  
power <- pnorm(diff/(sd/sqrt(n)) - z.alpha)  
return(power)  
}  
power.known <- findPower(100, 75, 5, 0.05)  
power.known

## [1] 0.8464817

find.power <- power.t.test(n = 5, sd = 75, sig.level = 0.05, delta = 100, type = "one.sample", alternative = "two.sided")  
find.power$power

## [1] 0.6141832

To reject the null hypothesis, we require the power to be larger than 0.8464817, if the population standard deviation is known; if the standard deviation is unknown, the power would be 0.6141832

### 2b

findN <- function(diff, sd, alpha, power){  
 N <- sd^2\*(qnorm(power)+qnorm(1-alpha/2))^2/diff^2  
 N  
}  
findN( 100, 75, 0.05, 0.9)

## [1] 5.910425

find.n <- power.t.test( power = 0.9, sd= 75, sig.level = 0.05, delta = 100, type = "one.sample", alternative = "two.sided")  
find.n$n

## [1] 8.072323

If the standard deviation of the population is known, we get the minimum sample size shoule be 6; if the standard deviation of the population is unknown, the samle size should be bigger than 9.

### 2c

findDiff <- function(n, sd, alpha, power){  
 Diff <- (qnorm(power)+qnorm(1-alpha/2))\*sd/sqrt(n)  
 Diff  
}  
power.90 <- findDiff(5, 75, 0.05, 0.9); power.90

## [1] 108.7237

power.80 <- findDiff(5, 75, 0.05, 0.8); power.80

## [1] 93.96802

According to the results, if the population standard deviation is known, then the smallest mean change for 90% power is 108.7237368; the smallesr mean change for 80% power is 93.9680249.

find.diff.90 <- power.t.test(power = 0.9, sd = 75, sig.level = 0.05, n = 5, type = "one.sample", alternative = "two.sided")  
find.diff.90$delta

## [1] 147.4417

find.diff.80 <- power.t.test(power = 0.8, sd = 75, sig.level = 0.05, n = 5, type = "one.sample", alternative = "two.sided")  
find.diff.80$delta

## [1] 126.1498

According to the results, if the population standard deviation is unknown, then the smallest mean change for 90% power is 147.4416676; the smallesr mean change for 80% power is 126.1497962.

### 3c.1

set.seed(seed = 2345)  
n <- 5  
mean <- 0  
sd <- 75  
numTrials <- 10000  
alpha <- 0.05  
count<- 0  
for(i in 1:numTrials){  
 y.norm <- rnorm(n, mean, sd)  
 ttest <- t.test(y.norm, y = NULL, mu = 0, conf.level = 0.95)  
 if(ttest$p.value < alpha){  
 count <- count + 1  
 }  
}  
power.1 <- count/numTrials; power.1

## [1] 0.0498

The power is very near to the value we set up as 0.05. Because the data we simulated are very similar to the population data, we got the power of the test very small. There will be very large probability we cannot detect the difference between the simulating data and the original data, which means we can hardly reject the null hypothesis.

Is that means, if the two groups of data are exactly the same, the probability of Type I error is pretty much the same as the power of detecting the difference? Because the detection of the difference is the false positive error?

set.seed(1796)  
mean <- 100  
sd <- 75  
n <- 5  
numTrials <- 10000  
alpha <- 0.05  
count<- 0  
for(i in 1:numTrials){  
 y.norm <- rnorm(n, mean, sd)  
 ttest <- t.test(y.norm, y = NULL, mu = 0, alternative = "two.sided",conf.level = 0.95)  
 count <- count + isTRUE( ttest$p.value < alpha)  
}  
count

## [1] 6130

power.2 <- count/numTrials; power.2

## [1] 0.613

For the second example, the distributions are pretty much the same. so it is impossible to detect the difference too, that is the reason why the power is so small. Moreover, the result is the same as the unknown standard deviation situation. Because the simulated data has its own sample standard deviation, even if we simulate the data with the population mean and standard deviation. so the result shoule be more near to a t-distribtion, other than the normal distributed. (although with big sample size, those two should be very similar to each other).

### 3c.2

compute\_power <- function(n, mean, sigma, numTrials, alpha){  
sample = matrix(rnorm(n\*numTrials, mean, sd), ncol=numTrials)  
xbar <- apply(sample, 2, mean)  
variance <- apply(sample, 2, var)  
matrix  
df.num = n-1   
test.stat <- (xbar - 0)/sqrt(variance/n)  
return (mean(abs(test.stat) >= qt((1-(alpha/2)), df = df.num)))  
}  
set.seed(2345)  
compute\_power(5, 0, 75, 10000, 0.05)

## [1] 0.0498

set.seed(1796)  
compute\_power(5, 100, 75, 10000, 0.05)

## [1] 0.613

As we can see, the simulated power under the null and alternative hypotheses through the compute\_power function are exactly the same as the results we got from the the last question.

### 3c.3

set.seed(seed=555)  
compute.power.1 <- function(N){  
 compute\_power(N, 100, 75, 10000, 0.05)  
}  
compute.power.2 <- function(Delta){  
 compute\_power(5, Delta, 75, 10000, 0.05)  
}  
library(purrr)  
N <- 1:20  
t(map(N, compute.power.1))

## Warning in qt((1 - (alpha/2)), df = df.num): NaNs produced

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]   
## [1,] NA 0.1108 0.2689 0.4524 0.6149 0.7463 0.8374 0.8959 0.9344 0.9586  
## [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]   
## [1,] 0.9777 0.9876 0.9924 0.9949 0.998 0.9987 0.9992 0.9995 0.9998 0.9999

Delta <- seq(0, 500, 50)  
t(map(Delta, compute.power.2))

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]  
## [1,] 0.0512 0.2098 0.6222 0.9101 0.9904 0.9998 1 1 1 1 1

It is almost impossible for us to simulated the estimated sample size and difference. We probably just try a series of different sample sizes and difference, and see which one fits in. As we can see in the example, if we get the sample size equal or larger than 8, we can get the power of 90%; we would also not recommond the sample size smaller than 6. The same for difference, if we get the difference between two groups is larger than 150, we can get the power over 90%.