

Homework 10

Max Ryoo

Problem 1

```
set.seed(12181998)
library(BSDA)

## Loading required package: lattice

##
## Attaching package: 'BSDA'

## The following object is masked from 'package:datasets':
##
##      Orange

library(UsingR)

## Loading required package: MASS
## Loading required package: HistData

##
## Attaching package: 'HistData'

## The following object is masked from 'package:BSDA':
##
##      Wheat

## Loading required package: Hmisc
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##      format.pval, units

##
## Attaching package: 'UsingR'

## The following object is masked from 'package:survival':
##
```

```
##      cancer

library(OpenMx)

## To take full advantage of multiple cores, use:
##   mxOption(NULL, 'Number of Threads', parallel::detectCores()) #now
##   Sys.setenv(OMP_NUM_THREADS=parallel::detectCores()) #before library(OpenMx)

library(ggplot2)
library(pwr)
pwr.t.test(d=0.5, sig.level=0.05, power=0.8, type="one.sample", alternative="greater")$n

## [1] 26.13753
```

The sample size to get her test to have a 80% power is 27. This was done by setting the effect size and using the `pwr.norm.test` function without the `n` parameter.

Problem 2

Part A

```
rep_samps <- replicate(10000, rnorm(27, mean=62.9, sd=13.3))
rep_means <- apply(rep_samps,2,mean)
rep_sd <- apply(rep_samps,2,function(x)sd(x))
t_scores <- (rep_means-62)/(rep_sd/sqrt(27))
t_rejection <- sapply(t_scores,function(x)(x)>=
                      qt(.95, df=27))
reject <- sum(t_rejection)
print(reject/10000)

## [1] 0.0953
```

The estimated power shown by the above computation is 0.0938, which is not close to the 80% power that the practitioner planned. This is most likely due to her sample size.

Part B

```
efs <- (62.9-62)/13.3
print(efs)

## [1] 0.06766917
```

The true effect size of the test is 0.06766917

Part C

```
pwr.t.test(d=efs, sig.level=0.05, power=0.8, type="one.sample", alternative="greater")$n
```

```
## [1] 1351.517
```

The sample size required for 80% power with the true effect size is 1352, which was found by doing `pwr.t.test()` for one.sample t tests

Part D

```
rep_samps2 <- replicate(10000,
                        rnorm(1352,
                              mean=62.9,
                              sd=13.3))
rep_means2 <- apply(rep_samps2,2,mean)
rep_sd2 <- apply(rep_samps2,2,function(x)sd(x))
t_scores2 <- (rep_means2-62)/(rep_sd2/sqrt(1352))
t_rejection2 <- sapply(t_scores2,
                      function(x)(x)>=qt(.95,
                                           df=1352))

reject_2 <- sum(t_rejection2)
print(reject_2/10000)

## [1] 0.8003
```

The estimated power shown by the above computation is 0.7969, which is close to the 80% power that the practitioner planned.

Part E

The results of power seems to be greatly affected by the sample size. When we took a sample size of 27 it showed a low power of 0.0938, which was not close to 0.8. However, when we did this with a higher sample size (1352) and the true effect size of 0.06766917 it was shown that the power is 0.7969, which is very close to 0.8. This resulting information will be the most useful when planning a study because now I would know how big of an effect a sample size has on a study.

Problem 2

Part A

```
setwd("/Users/maxryoo/Documents/Fall 2018/STAT3080/HW10")
data4 <- read.csv("data4.csv")
pop1 <- mean(data4$pop1)
pop2 <- mean(data4$pop2)
twopop <- c(pop1, pop2)
print(twopop)
```

```
## [1] 48.4489 32.5789
```

I read in the csv file and set the population mean for population 1 as pop1 and the mean for population 2 as pop2. I converted them to vector and printed the resulting vector, which contains 48.4489 and 32.5789

Part B

```
pop1mc <- replicate(100, sample(data4$pop1, 9))
pop2mc <- replicate(100, sample(data4$pop2, 9))
boot_pop12mc <- lapply(1:100, function(x)
  replicate(10000, sample(c(pop1mc[,x], pop2mc[,x]),
    replace=T)))
boot_difference <- lapply(boot_pop12mc,
  function(x)
    apply(x[1:9,], 2, mean) -
    apply(x[10:18,], 2, mean))
p <- sapply(1:100, function(x)
  sum(boot_difference[[x]] >=
    (mean(pop1mc[,x]) - mean(pop2mc[,x])))/10000)
power1 <- sum(p <= 0.05)/100
print(power1)
## [1] 0.23
```

In order to find the power of the randomization test for whether the mean of the first population is larger than the mean of the second population, we used both Monte Carlo and bootstrapping. We bootstrapped 10000 and for Monte Carlo we used 100. The power that we got by calculating above was 0.2.

Part C

We found out that the power determined above was 0.2. The power means the probability of rejecting the null hypothesis when it is true. Our question was dealing with checking whether the mean of the first population was larger than the mean of the second population to this our Monte Carlo and bootstrapping methods would have accurately guessed correctly 20% percent of the time. If this power were close to 1 the hypothesis test was good at detecting a false null hypothesis, but our power seems to be fairly low.

References

1. <https://www.itl.nist.gov/div898/handbook/eda/section3/eda3672.htm>
2. <https://www.statisticshowto.datasciencecentral.com/standardized-test-statistic/>