Homework #6

Carson Crenshaw (cgc8gdt)

Insert packages and set the seed.

```
# Set Seed
set.seed(10062002)
#Library
library(ggplot2)
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
```

```
##
## Attaching package: ’Hmisc’

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

```
library(OpenMx)
```

```
## OpenMx may run faster if it is compiled to take advantage of multiple cores.
```

```
library(pwr)
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:Hmisc’:
##
##     src, summarize

## The following object is masked from ’package:MASS’:
##
##     select

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

Read in all datasets necessary for the homework.

```
# Read in datasets
data1csv <- read.csv("data1.csv")
data2csv <- read.csv("data2.csv")
data3csv <- read.csv("data3.csv")
data4csv <- read.csv("data4.csv")
data5csv <- read.csv("data5.csv")
data6csv <- read.csv("data6.csv")
```

Problem 1

Part a

```
# Store the population correlations in the variables 'data#cor'
data1cor <- cor(data1csv[1], data1csv[2])
data2cor <- cor(data2csv[1], data2csv[2])
data3cor <- cor(data3csv[1], data3csv[2])
```

Part b

```
# Monte Carlo simulation function for paired t-test
# Store the paired t-test type 1 error calculations in the variables 'data#paired'
pairedttestfunc <- function(dataset, x){
  samp <- sample_n(dataset, x)
  p.val <- t.test(samp$V1, samp$V2, mu=0, alternative="two.sided", paired=TRUE)
  p.val$p.value <= 0.05
}

data1paired <- sum(replicate(10000, pairedttestfunc(data1csv, 13)))/10000
data2paired <- sum(replicate(10000, pairedttestfunc(data2csv, 13)))/10000
data3paired <- sum(replicate(10000, pairedttestfunc(data3csv, 13)))/10000
```

Part c

```
# Monte Carlo simulation function for a two-sample test
# Store the two sample t-test type 1 error calculations in the variables 'data#twosamp'
twosampttestfunc <- function(dataset, x){
  samp <- sample_n(dataset, x)
  p.val <- t.test(samp$V1, samp$V2, mu=0, alternative="two.sided")
  p.val$p.value <= 0.05
}

data1twosamp <- sum(replicate(10000, twosampttestfunc(data1csv, 13)))/10000
data2twosamp <- sum(replicate(10000, twosampttestfunc(data2csv, 13)))/10000
data3twosamp <- sum(replicate(10000, twosampttestfunc(data3csv, 13)))/10000
```

Final summary of the paired results in a table-like structure:

```
# Dataframe for the final results
data1 <- c(data1cor, data1paired, data1twosamp)
data2 <- c(data2cor, data2paired, data2twosamp)
data3 <- c(data3cor, data3paired, data3twosamp)
df1 <- data.frame(data1, data2, data3)
row.names(df1) <- c("Correlation", "Paired", "Two-Sample")
print(df1)
```

```
##                data1    data2       data3
## Correlation 0.524066 -0.52036 0.002426237
## Paired      0.049300  0.05130 0.055400000
## Two-Sample  0.008500  0.11340 0.048800000
```

The previous exercise in Problem 1 is a demonstration on how different correlation values will effect a paired and two-sample test. If there is high correlation, one would normally use a paired test because it is assumed that the variables are related to each other. In other words, if the variable correlation is high there are concerns regarding multicollinearity within a two-sample test.

The descriptive summary table above illustrates that the paired test is more robust across the sample sizes and the various correlation values. The paired test type 1 error values for all of the sample sizes are approximately what would be expected from a normal distribution. For the datasets with moderate correlation (data1, data2), the paired test type 1 error values are very close to the expected error of 0.05. For the dataset with no correlation (data3), the paired test type 1 error value is also extremely close to the normal expected error. The two-sample type 1 error values, however, only follow the expectation for data3 (a sample in which there is no significant correlation). The two-sample test is not robust when incorporating data from samples with higher correlation values.

Problem 2

Part a

```
# Store the population correlations in the variables 'data#cor'
data4cor <- cor(data4csv[1], data4csv[2])
data5cor <- cor(data5csv[1], data5csv[2])
data6cor <- cor(data6csv[1], data6csv[2])
```

Part b

```
# Monte Carlo simulation function for the paired test
# Store the paired t-test type 1 error calculations in the variables 'data#paired'
pairedttestfunc2 <- function(dataset, x){
  samp <- sample_n(dataset, x)
  p.val <- t.test(samp$V1, samp$V2, mu=0, alternative="two.sided", paired=TRUE)
  p.val$p.value <= 0.05
}

data4paired <- sum(replicate(10000, pairedttestfunc2(data4csv, 13)))/10000
data5paired <- sum(replicate(10000, pairedttestfunc2(data5csv, 13)))/10000
data6paired <- sum(replicate(10000, pairedttestfunc2(data6csv, 13)))/10000
```

Part c

```
# Monte Carlo simulation function for the two-sample test
# Store the two sample t-test type 1 error calculations in the variables 'data#twosamp'
twosampttestfunc2 <- function(dataset, x){
  samp <- sample_n(dataset, x)
  p.val <- t.test(samp$V1, samp$V2, mu=0, alternative="two.sided")
  p.val$p.value <= 0.05
}

data4twosamp <- sum(replicate(10000, twosampttestfunc2(data4csv, 13)))/10000
data5twosamp <- sum(replicate(10000, twosampttestfunc2(data5csv, 13)))/10000
data6twosamp <- sum(replicate(10000, twosampttestfunc2(data6csv, 13)))/10000
```

Final summary of the two-sample results in a table-like structure:

```
# Dataframe for the final results
data4 <- c(data4cor, data4paired, data4twosamp)
data5 <- c(data5cor, data5paired, data5twosamp)
data6 <- c(data6cor, data6paired, data6twosamp)
df2 <- data.frame(data4, data5, data6)
row.names(df2) <- c("Correlation", "Paired", "Two-Sample")
print(df2)
```

```
##                  data4      data5        data6
## Correlation 0.5906402 -0.5721193 -0.007297158
## Paired      0.0499000  0.0745000  0.035100000
## Two-Sample  0.0147000  0.1161000  0.033700000
```

The goal of the previous exercise in Problem 2 is a continuation of the test in Problem 1: to demonstrate that the paired t-test is more consistent across different degrees of correlation. Regardless of the sample correlation, the paired t-test does a better job of limiting/controlling the amount of type 1 error than the two-sample t-test. For both values of higher correlation (data4, data5), the paired test type 1 error values are close to the expected error of 0.05. The paired test value with no illustrated correlation (data6) is also closer to the aforementioned expected error value. In comparison, there is a large degree of variation between the resulting error values of the two-sample t-tests and none come as close to the expected value as the paired t-test. The two sample may be more powerful, but the paired test is better here because it relaxes the independence assumption.

This paired t-test, however, fails to result in the approximate expected type 1 error value (0.05) that was shown in Problem 1. This is a result of the datasets which contain a skewed distribution (thus violating the normality assumption for t-tests which demands two normal populations).