BMEG 802 – Advanced Biomedical Experimental Design and Analysis

Assignment 1

Joshua G. A. Cashaback, PhD

Question 1

install.packages("effsize") install.packages("pwr")

```
library(effsize)
library(pwr)
data1 = c(62, 62, 68, 48, 51, 60, 51, 57, 57, 41, 62, 50, 53, 34, 62, 61)
alpha = 0.05
t1 = (mean(data1) - 60) / (sd(data1) / (length(data1))^(1/2.))
tcrit1 = qt(alpha, length(data1)-1) # Left tail
pval1 = pt(-abs(t1),length(data1) - 1) #
abs(t1)
```

```
## [1] 2.282218
```

```
abs(tcrit1)
```

```
## [1] 1.75305
```

Question 1 Cont'd

```
pval1
```

```
## [1] 0.01874476
```

```
d0 = cohen.d(data1, NA, mu = 60) $estimate
abs(d0)
```

```
## [1] 0.5705546
```

```
res1 <- t.test(data1, mu = 60, alternative = "less") $p.value
res1
```

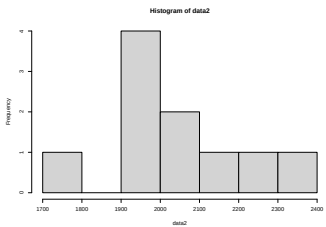
```
## [1] 0.01874476
```

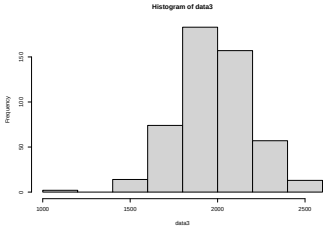
```
pwr.t.test(d = d0, power = 0.8, sig.level = 0.05, type = "one.sample", alternative
```

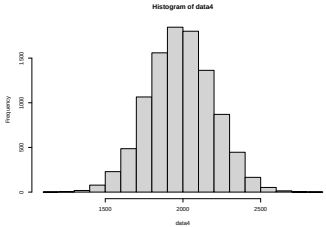
```
## [1] 20.41511
```

Question 2

```
alpha1 = 0.01
data2 <- rnorm(10, mean = 1985, sd=210)
data3 <- rnorm(500, mean = 1985, sd=210)
data4 <- rnorm(10000, mean = 1985, sd=210)
hist(data2)
hist(data3)
hist(data4)
```







Question 2 cont'd(n = 10)

```r
t2 = (mean(data2) - 2000) / (sd(data2) / (length(data2))^(1/2.))
tcrit2 = qt(alpha/2, length(data2)-1) # Left tail (alpha/2 = 0.025)
pval2 = 2*pt(-abs(t2),length(data2) - 1) # -abs() do calc on left-tail
# multiply by 2 (accounting for both sided)
abs(t2)
```

```
## [1] 0.6318793
```

```r
abs(tcrit2)
```

```
## [1] 2.262157
```

```r
pval2
```

```
## [1] 0.5431808
```

```r
d = cohen.d(data2, NA, mu = 2000) $estimate
abs(d)
```

```
## [1] 0.1998178
```

```r
res2 <- t.test(data2, mu = 2000, alternative = "two.sided") $p.value
res2
```

```
## [1] 0.5431808
```

Question 2 cont'd (n = 500)

```
t3 = (mean(data3) - 2000) / (sd(data3) / (length(data3))^(1/2.))
tcrit3 = qt(alpha/2, length(data3)-1) # Left tail (alpha/2 = 0.025)
pval3 = 2*pt(-abs(t3),length(data3) - 1) # -abs() do calc on left-tail
# multiply by 2 (accounting for both sided)
abs(t3)
```

```
## [1] 2.018361
```

```
abs(tcrit3)
```

```
## [1] 1.964729
```

```
pval3
```

```
## [1] 0.04408818
```

```
d = cohen.d(data3, NA, mu = 2000) $estimate
abs(d)
```

```
## [1] 0.09026385
```

```
res3 <- t.test(data3, mu = 2000, alternative = "two.sided") $p.value
res3
```

```
## [1] 0.04408818
```

Question 2 cont'd (n = 10000)

```r
t4 = (mean(data4) - 2000) / (sd(data4) / (length(data4))^(1/2.))
tcrit4 = qt(alpha/2, length(data4)-1) # Left tail (alpha/2 = 0.025)
pval4 = 2*pt(-abs(t4),length(data4) - 1) # -abs() do calc on left-tail
# multiply by 2 (accounting for both sided)
abs(t4)
```

```
## [1] 6.705035
```

```r
abs(tcrit4)
```

```
## [1] 1.960201
```

```r
pval4
```

```
## [1] 2.122346e-11
```

```r
d = cohen.d(data4, NA, mu = 2000) $estimate
abs(d)
```

```
## [1] 0.06705035
```

```r
res4 <- t.test(data4, mu = 2000, alternative = "two.sided") $p.value
res4
```

```
## [1] 2.122346e-11
```

Question 3

```r
t5 = ((4.75 - 3.0)-(0-0))/(1.0^2 / 15 + 1.5^2 / 22)^(1/2)
v5 = (1.0^2 / 15 + 1.5^2 / 22)^2 / (1.0^4 / (15^2 * (15 - 1)) + 1.5^4 /(22^2 * (22 - 1)))
alpha5 = 0.01
tcrit5 = qt(1 - alpha5, v5)
pval5 = 1 - pt(t5,v5)
t5
```

```
## [1] 4.257676
```

```r
tcrit5
```

```
## [1] 2.437737
```

```r
pval5
```

```
## [1] 7.362439e-05
```

```r
d5 = (4.75 - 3.0) / (sqrt(((15 - 1) * 1.0^2 + (22 - 1) * 1.5^2) / (15 + 22 - 2)))
abs(d5)
```

```
## [1] 1.322876
```

```r
pwr.t.test(d = d5, power = 0.8, sig.level = 0.05, type = "two.sample", alternative = "greater")$n
```

```
## [1] 7.840869
```

Question 4

```
data6a = c(94,94,103,112,91,112,112,97,100,106,46,41,112,112,94,112,94,109,112,112,91,100,112,70,76,97,112,112,106,112,112,106,100,88,
data6b = c(91,61,85,88,94,112,109,79,109,115,46,45,106,112,91,115,59,85,112,76,79,97,109,70,58,97,112,97,112,85,112,103,100,88,109,85,
data6diff = data6a - data6b
t6 = (mean(data6diff) - 0) / (sd(data6diff) / (length(data6diff))^(1/2.))
tcrit6 = qt(1 - 0.05, length(data6diff)-1)
pval6 = 1 - pt(t6,length(data6diff)-1)
abs(t6)
```

```
## [1] 3.770635
```

```
abs(tcrit6)
```

```
## [1] 1.688298
```

```
pval6
```

```
## [1] 0.0002926913
```

```
d6 = cohen.d(data6a,data6b, paired = TRUE) $estimate
abs(d6)
```

```
## [1] 0.4162571
```

```
t.test(data6a, data6b, paired = TRUE, alternative = "greater") $p.value
```

```
## [1] 0.0002926913
```

```
pwr.t.test(d = d6, power = 0.8, sig.level = 0.05, type = "paired", alternative = "greater")$n
```

```
## [1] 37.0726
```

Question 5

```
beta0 <- rbeta(20, 1, 9)
beta1 <- rbeta(20, 2, 9)
```

Question 5 cont'd

```
library("car")
```

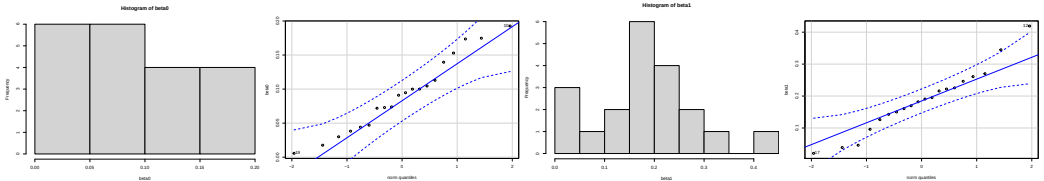
```
## Loading required package: carData
```

```
hist(beta0)
qqPlot(beta0)
```

```
## [1] 10 19
```

```
hist(beta1)
qqPlot(beta1)
```

```
## [1] 12 17
```



Question 5 cont'd

```
shapiro.test(beta0)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  beta0
## W = 0.9616, p-value = 0.5762
```

```
shapiro.test(beta1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  beta1
## W = 0.96936, p-value = 0.7412
```

Question 5 cont'd

```r
wilcox.test(beta0, beta1, alternative = "two.sided")
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  beta0 and beta1
## W = 79, p-value = 0.0007474
## alternative hypothesis: true location shift is not equal to 0
```

Question 5 cont'd

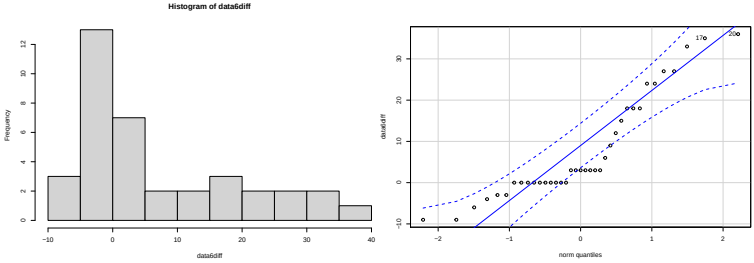
```
a = beta0
b = beta1
c <- array(dim=c(length(a),length(b)))
for (i in 1:length(a)) {
  for (j in 1:length(b)){
    if (a[i] > b[j]) {
    c[i,j] = 1
    } else if (a[i] == b[j]) {
    c[i,j] = 0.5
    } else {
    c[i,j] = 0.0
}}}
CLES = (abs(sum(c) / (length(a) * length(b)) - 0.5) + 0.5) * 100
CLES
```

```
## [1] 80.25
```

Question 6

```
library("car")
hist(data6diff)
qqPlot(data6diff)
```

```
## [1] 20 17
```



Question 6 cont'd

```r
library("car")
shapiro.test(data6diff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data6diff
## W = 0.86814, p-value = 0.0004299
```

```r
wilcox.test(data6a, data6b, paired = TRUE, alternative = "greater")
```

```
## Warning in wilcox.test.default(data6a, data6b, paired = TRUE, alternative =
## "greater"): cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(data6a, data6b, paired = TRUE, alternative =
## "greater"): cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data6a and data6b
## V = 318.5, p-value = 0.0009229
## alternative hypothesis: true location shift is greater than 0
```

Question 6 cont'd

```
a1 = data6a # e.g., pre intervention
a2 = data6b  # e.g., post intervention
a = a2 - a1
b = array(0,dim=c(length(a)))  # mean value you are comparing to (same as mu_0 = 0)
c <- array(dim=c(length(a),length(b)))
for (i in 1:length(a)) {
  for (j in 1:length(b)){
    if (a[i] > b[j]) {
    c[i,j] = 1
    } else if (a[i] == b[j]) {
    c[i,j] = 0.5
    } else {
    c[i,j] = 0.0
}}}
CLES = (abs(sum(c) / (length(a) * length(b)) - 0.5) + 0.5) * 100
CLES
```

```
## [1] 70.27027
```

Question 7

```
pvals = c(0.06, 0.01, 0.024)
p.adjust(pvals, method = "bonferroni", n = length(pvals))
```

```
## [1] 0.180 0.030 0.072
```

```
p.adjust(pvals, method = "holm", n = length(pvals))
```

```
## [1] 0.060 0.030 0.048
```

Question 8

```
PVALUES0 = array(NA,10000)
for (i in 1:10000) {
group1 <- rnorm(10, mean = 10, sd=2)
group2 <- rnorm(10, mean = 10, sd=2)
pval = t.test(group1, group2, alternative = "two.sided")$p.value
PVALUES0[i] = pval
}
```

Question 8 cont'd

```
PVALUES1 = array(NA,100000)
for (i in 1:100000) {
group1 <- rnorm(10, mean = 10, sd=2)
group2 <- rnorm(10, mean = 12, sd=2)
pval = t.test(group1, group2, alternative = "two.sided")$p.value
PVALUES1[i] = pval
}
```

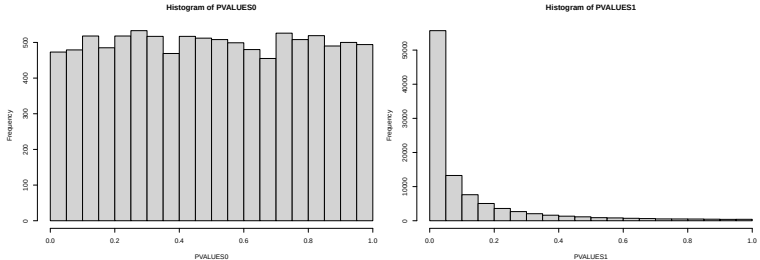
Question 8 cont'd

```
hist(PVALUES0)
hist(PVALUES1)
sum(PVALUES0 < 0.05) / 100000 * 100
```

```
## [1] 0.473
```

```
sum(PVALUES1 < 0.05) / 100000 * 100
```

```
## [1] 55.692
```



Question 9a

What N gives you 80% power?

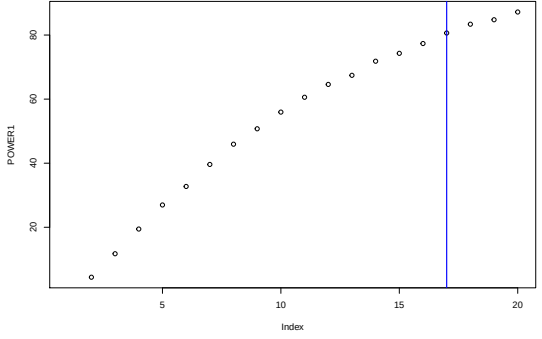
```
min_n = 2
max_n = 20
POWER1 = array(NA,max_n - min_n)
for (n in min_n:max_n){
  PVALUES1 = array(NA,10000)
  for (i in 1:10000) {
  group1 <- rnorm(n, mean = 10.0, sd=2.0)
  group2 <- rnorm(n, mean = 12.0, sd=2.0)
  pval = t.test(group1, group2, alternative = "two.sided")$p.value
  PVALUES1[i] = pval
  }
  POWER1[n] = sum(PVALUES1 < 0.05) / 10000 * 100
}
```

Question 9a Cont'd

What N gives you 80% power?

```
minsub = min(which(POWER1 > 80)) # finds 80% crossing
sub = toString(minsub)
plot(POWER1)
abline(v=minsub, col="blue", lty=1, lwd=2)
text(0, 75, sub, col = "blue")
sub
```

```
## [1] "17"
```



Question 9b

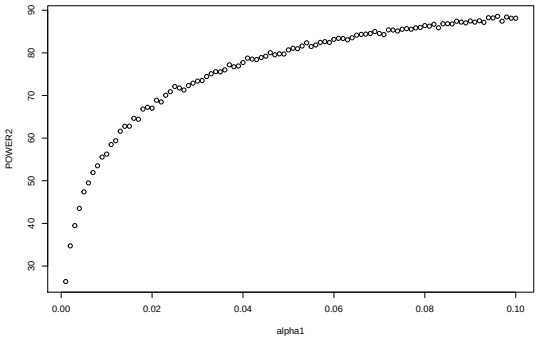
What N gives you 80% power?

```
alpha1 = seq(from = 0.001, to = 0.1, by = 0.001)
POWER2 = array(NA,length(alpha1))
for (j in 1:length(alpha1)){
  PVALUES2 = array(NA,10000)
  for (i in 1:10000) {
  group1 <- rnorm(17, mean = 10.0, sd=2.0)
  group2 <- rnorm(17, mean = 12.0, sd=2.0)
  pval = t.test(group1, group2, alternative = "two.sided")$p.value
  PVALUES2[i] = pval
  }
  POWER2[j] = sum(PVALUES2 < alpha1[j]) / 10000 * 100
}
```

Question 9b Cont'd

What N gives you 80% power?

```
plot(alpha1,POWER2)
```



Question 9c

What N gives you 80% power?

```
mu2 = seq(from = 8.0, to = 12.0, by = 0.1)
POWER3 = array(NA,length(mu2))
for (j in 1:length(mu2)){
  PVALUES3 = array(NA,10000)
  for (i in 1:10000) {
  group1 <- rnorm(17, mean = 10.0, sd=2.0)
  group2 <- rnorm(17, mean = mu2[j], sd=2.0)
  pval = t.test(group1, group2, alternative = "two.sided")$p.value
  PVALUES3[i] = pval
  }
  POWER3[j] = sum(PVALUES3 < 0.05) / 10000 * 100
}
```

Question 9c Cont'd

What N gives you 80% power?

```
plot(mu2,POWER3)
```