Homework 2

Joshua Oswari

4/12/2019

Problem 1

================================ Part 1 ================================

```r
nutrient = read.table(file = "~/Documents/Math189/nutrient.txt", header = FALSE, sep = "")

#to omit the first column
dat = nutrient[,2:6]

#name each column
colnames(dat)=c("Calcium", "Iron", "Protein", "Vitamin A", "Vitamin C")

#sample mean and sd for each variable
print(paste("Sample Mean for Calcium is", mean(dat[,1])))
```

```
## [1] "Sample Mean for Calcium is 624.049253731343"
```

```r
print(paste("Standard Deviation for Calcium is", sd(dat[,1])))
```

```
## [1] "Standard Deviation for Calcium is 397.277540103266"
```

```r
print(paste("Sample Mean for Iron is", mean(dat[,2])))
```

```
## [1] "Sample Mean for Iron is 11.1298995929444"
```

```r
print(paste("Standard Deviation for Iron is", sd(dat[,2])))
```

```
## [1] "Standard Deviation for Iron is 5.98419047008833"
```

```r
print(paste("Sample Mean for Protein is", mean(dat[,3])))
```

```
## [1] "Sample Mean for Protein is 65.8034409769335"
```

```r
print(paste("Standard Deviation for Protein is", sd(dat[,3])))
```

```
## [1] "Standard Deviation for Protein is 30.5757564314087"
```

```r
print(paste("Sample Mean for Vitamin A is", mean(dat[,4])))
```

```
## [1] "Sample Mean for Vitamin A is 839.635345997286"
```

```r
print(paste("Standard Deviation for Vitamin A is", sd(dat[,4])))
```

```
## [1] "Standard Deviation for Vitamin A is 1633.53982830006"
```

```r
print(paste("Sample Mean for Vitamin C is", mean(dat[,5])))
```

```
## [1] "Sample Mean for Vitamin C is 78.9284464043419"
```

```r
print(paste("Standard Deviation for Vitamin C is", sd(dat[,5])))
```

```
## [1] "Standard Deviation for Vitamin C is 73.59527211824"
```

============================== Part 2 ================================

```r
#t-test if recommended for Calcium is 1000
#H0: calcium >= 1000
t.test(dat$Calcium, mu = 1000, alternative = "less", conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  dat$Calcium
## t = -25.69, df = 736, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 1000
## 95 percent confidence interval:
##      -Inf 648.1502
## sample estimates:
## mean of x
##  624.0493
```

```r
#t-test if recommended for Iron is 15
#H0: Iron >= 15
t.test(dat$Iron, mu = 15, alternative = "less", conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  dat$Iron
## t = -17.557, df = 736, p-value < 2.2e-16
## alternative hypothesis: true mean is less than 15
## 95 percent confidence interval:
##      -Inf 11.49293
## sample estimates:
## mean of x
##   11.1299
```

```r
#t-test if recommended for Protein is 60
#H0: Protein >= 60
t.test(dat$Protein, mu = 60, alternative = "less", conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  dat$Protein
## t = 5.1528, df = 736, p-value = 1
## alternative hypothesis: true mean is less than 60
## 95 percent confidence interval:
##      -Inf 67.65833
## sample estimates:
## mean of x
##  65.80344
```

```r
#t-test if recommended for Vitamin A is 800
#H0: Vitamin A >= 800
t.test(dat$`Vitamin A`, mu = 800, alternative = "less", conf.level = 0.95)
```

```
##
```

```
##  One Sample t-test
##
## data:  dat$`Vitamin A`
## t = 0.6587, df = 736, p-value = 0.7449
## alternative hypothesis: true mean is less than 800
## 95 percent confidence interval:
##      -Inf 938.7346
## sample estimates:
## mean of x
##  839.6353
```

```
#t-test if recommended for Vitamin B is 75
#H0: Vitamin B >= 75
t.test(dat$`Vitamin C`, mu = 75, alternative = "less", conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  dat$`Vitamin C`
## t = 1.4491, df = 736, p-value = 0.9261
## alternative hypothesis: true mean is less than 75
## 95 percent confidence interval:
##      -Inf 83.39313
## sample estimates:
## mean of x
##  78.92845
```

============================== Part 3 ================================

```
# based on the result, majority of the Women in the U.S. are deprived nutrient
# in Calcium and Iron. For the nutrients, such as Vitamin A, Vitamin C, and Protein,
# the majority of Women in the U.S. already fulfil or meet the minimum amount.
# I would suggest the public that the Women in the U.S. should take eat something
# that rich in Calcium and Iron. Another solution is that women can
# take pills or tablets for that specific nutrient.
```

Problem 2

=============================== Part 1 ===============================

```
# Suppose we know that the first 10 variables have mean
# equal to 2 and the rest of them have mean equal to 0.

multiple = read.table(file = "~/Documents/Math189/multiple.txt", header = FALSE, sep = "")

#h0 = m = 0
#ha = m =/ 0
for(i in 1:50) {
ttest = t.test(multiple[,i], mu = 0, conf.level = 0.90)
print(ttest)
}
```

```
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = 18.475, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.723846 2.064290
## sample estimates:
## mean of x
##  1.894068
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 17.851, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.768226 2.130891
## sample estimates:
## mean of x
##  1.949559
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 18.291, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.679540 2.014907
## sample estimates:
## mean of x
##  1.847223
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 20.273, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.933905 2.278943
## sample estimates:
## mean of x
##  2.106424
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 19.168, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.95301 2.32345
## sample estimates:
```

```
## mean of x
##   2.13823
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 18.065, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.783176 2.144153
## sample estimates:
## mean of x
##  1.963664
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 21.74, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.754419 2.044572
## sample estimates:
## mean of x
##  1.899496
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 18.761, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.775605 2.120403
## sample estimates:
## mean of x
##  1.948004
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 18.583, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.823691 2.181550
## sample estimates:
## mean of x
##   2.00262
##
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = 23.171, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  1.907142 2.201563
## sample estimates:
## mean of x
##  2.054352
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.18938, df = 99, p-value = 0.8502
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1446933  0.1819501
## sample estimates:
## mean of x
## 0.0186284
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.96126, df = 99, p-value = 0.3388
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.07745477  0.29044314
## sample estimates:
## mean of x
## 0.1064942
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.19812, df = 99, p-value = 0.8434
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1539629  0.1956826
## sample estimates:
##  mean of x
## 0.02085984
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -1.1941, df = 99, p-value = 0.2353
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.28502260  0.04656421
## sample estimates:
```

```
##  mean of x
## -0.1192292
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.2093, df = 99, p-value = 0.8346
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1940694  0.1506197
## sample estimates:
##   mean of x
## -0.02172481
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.95678, df = 99, p-value = 0.341
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.06793318  0.25268662
## sample estimates:
##  mean of x
## 0.09237672
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.75921, df = 99, p-value = 0.4495
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.08994675  0.24150147
## sample estimates:
##  mean of x
## 0.07577736
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.12743, df = 99, p-value = 0.8989
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1477353  0.1722971
## sample estimates:
##  mean of x
## 0.01228092
##
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = 0.70005, df = 99, p-value = 0.4855
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.09694611  0.23828681
## sample estimates:
##  mean of x
## 0.07067035
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -1.7513, df = 99, p-value = 0.08299
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.330211753 -0.008802341
## sample estimates:
## mean of x
## -0.169507
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.32804, df = 99, p-value = 0.7436
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1344277  0.2006218
## sample estimates:
##  mean of x
## 0.03309704
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.78477, df = 99, p-value = 0.4345
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.25290071  0.09056424
## sample estimates:
##   mean of x
## -0.08116824
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.59653, df = 99, p-value = 0.5522
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.2364862  0.1114748
## sample estimates:
```

```
##  mean of x
## -0.0625057
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.71301, df = 99, p-value = 0.4775
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.08883449  0.22254884
## sample estimates:
##  mean of x
## 0.06685718
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.30977, df = 99, p-value = 0.7574
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1346496  0.1964134
## sample estimates:
## mean of x
## 0.0308819
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.5053, df = 99, p-value = 0.1354
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.01483338  0.30286524
## sample estimates:
## mean of x
## 0.1440159
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.0091395, df = 99, p-value = 0.9927
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1355274  0.1370277
## sample estimates:
##    mean of x
## 0.0007501295
##
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = 1.054, df = 99, p-value = 0.2944
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.06141962  0.27495221
## sample estimates:
## mean of x
## 0.1067663
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.41598, df = 99, p-value = 0.6783
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.2108541  0.1263693
## sample estimates:
##   mean of x
## -0.04224243
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.32, df = 99, p-value = 0.1899
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.03682923  0.32242160
## sample estimates:
## mean of x
## 0.1427962
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.72251, df = 99, p-value = 0.4717
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.23325271  0.09180513
## sample estimates:
##   mean of x
## -0.07072379
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -1.696, df = 99, p-value = 0.09303
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.335239077 -0.003557506
## sample estimates:
```

```
##  mean of x
## -0.1693983
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.71927, df = 99, p-value = 0.4737
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.23971516  0.09480315
## sample estimates:
## mean of x
## -0.072456
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.10195, df = 99, p-value = 0.919
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1636073  0.1446775
## sample estimates:
##    mean of x
## -0.009464913
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.2511, df = 99, p-value = 0.2139
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.04744418  0.33748100
## sample estimates:
## mean of x
## 0.1450184
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.45237, df = 99, p-value = 0.652
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.2011076  0.1149880
## sample estimates:
##  mean of x
## -0.0430598
##
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = 0.95705, df = 99, p-value = 0.3409
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.07208563  0.26826395
## sample estimates:
##  mean of x
## 0.09808916
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.2313, df = 99, p-value = 0.2211
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.04319176  0.29110621
## sample estimates:
## mean of x
## 0.1239572
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.15057, df = 99, p-value = 0.8806
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1520708  0.1824013
## sample estimates:
##  mean of x
## 0.01516522
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -1.8111, df = 99, p-value = 0.07315
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.34039321 -0.01478078
## sample estimates:
## mean of x
## -0.177587
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.3633, df = 99, p-value = 0.7172
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1292971  0.2017258
## sample estimates:
```

```
##  mean of x
## 0.03621437
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.45892, df = 99, p-value = 0.6473
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1150807  0.2029935
## sample estimates:
##  mean of x
## 0.04395642
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.39797, df = 99, p-value = 0.6915
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1983180  0.1216304
## sample estimates:
##   mean of x
## -0.03834381
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -2.28, df = 99, p-value = 0.02475
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.38242319 -0.06013709
## sample estimates:
##  mean of x
## -0.2212801
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.2367, df = 99, p-value = 0.2191
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.04666318  0.31905250
## sample estimates:
## mean of x
## 0.1361947
##
##
##  One Sample t-test
##
```

```
## data:  multiple[, i]
## t = -0.29605, df = 99, p-value = 0.7678
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1917211  0.1336984
## sample estimates:
##   mean of x
## -0.02901138
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.94111, df = 99, p-value = 0.3489
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.06995563  0.25301807
## sample estimates:
##  mean of x
## 0.09153122
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 1.5002, df = 99, p-value = 0.1368
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.01639352  0.32338001
## sample estimates:
## mean of x
## 0.1534932
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = 0.14034, df = 99, p-value = 0.8887
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1533394  0.1816532
## sample estimates:
##  mean of x
## 0.01415693
##
##
##  One Sample t-test
##
## data:  multiple[, i]
## t = -0.34389, df = 99, p-value = 0.7317
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  -0.1726142  0.1133812
## sample estimates:
```

```
##   mean of x
## -0.02961651
```

=============================== Part 2 ================================

```r
#check Type I and Type II Error

#h0 = m = 2
#ha = m /= 2
for(i in 1:10) {
ttest = t.test(multiple[,i], mu = 2, conf.level = 0.90)
  if ( ttest$estimate > 2) {
    print("Type I")
  } else {
    print("Type II")
  }
}
```

```
## [1] "Type II"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
```

```r
#h0 = m = 0
#ha = m /= 0
for(i in 11:50) {
ttest = t.test(multiple[,i], mu = 0, conf.level = 0.90)
  if ( ttest$estimate > 0) {
    print("Type I")
  } else {
    print("Type II")
  }
}
```

```
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type I"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
```

```
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type I"
## [1] "Type II"
## [1] "Type II"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
## [1] "Type II"
## [1] "Type I"
## [1] "Type II"
## [1] "Type I"
## [1] "Type I"
## [1] "Type I"
## [1] "Type II"
```

```r
pvalVector = c()
for(i in 1:50) {
ttest = t.test(multiple[,i], mu = 2, conf.level = 0.90)
pvalVector[length(pvalVector)+1] = ttest$p.value
}
#fwer
a = pvalVector< 0.1
a
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

============================ Part 3 ============================

```r
pvalVector = c()
for(i in 1:50) {
ttest = t.test(multiple[,i], mu = 2, conf.level = 0.90)
pvalVector[length(pvalVector)+1] = ttest$p.value
}
p.adjust(pvalVector, method = "bonferroni")
```

```
##  [1] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
##  [6] 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
## [11] 4.263515e-35 1.344587e-29 9.849835e-33 6.336850e-37 6.117723e-34
## [16] 1.979395e-34 1.369341e-33 6.409926e-36 2.705486e-33 7.146033e-39
## [21] 5.711131e-34 4.645586e-35 2.667432e-34 6.642712e-36 2.030998e-34
## [26] 8.377076e-34 6.649704e-42 1.528515e-32 4.850518e-35 9.678496e-30
## [31] 8.264884e-37 9.597030e-38 7.852444e-36 1.271170e-37 1.632984e-27
```

```
## [36] 2.535161e-37 2.662029e-32 1.918144e-32 2.435359e-34 1.539340e-38
## [41] 2.493025e-34 1.423024e-35 8.209355e-37 1.261373e-39 2.789333e-29
## [46] 4.693907e-36 3.407839e-34 2.252900e-31 2.645617e-34 1.089423e-40
```

```
#fwer
a = p.adjust(pvalVector, method = "bonferroni")< 0.1
a
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

============================== Part 4 ==============================

```
pvalVector = c()
for(i in 1:50) {
ttest = t.test(multiple[,i], mu = 2, conf.level = 0.90)
pvalVector[length(pvalVector)+1] = ttest$p.value
}
p.adjust(pvalVector, method = "BH")
```

```
##  [1] 3.424507e-01 6.720708e-01 1.628297e-01 3.424507e-01 2.597828e-01
##  [6] 7.539646e-01 2.939604e-01 6.570608e-01 9.806501e-01 5.883199e-01
## [11] 2.507950e-36 3.538387e-31 3.078074e-34 7.040944e-38 2.184901e-35
## [16] 9.671418e-36 4.564469e-35 4.744795e-37 8.727374e-35 1.786508e-39
## [21] 2.115234e-35 2.552904e-36 1.066973e-35 4.744795e-37 9.671418e-36
## [26] 2.888647e-35 6.649704e-42 4.631865e-34 2.552904e-36 2.615810e-31
## [31] 7.513530e-38 1.599505e-38 5.234963e-37 1.815958e-38 4.082460e-29
## [36] 3.168951e-38 7.605796e-34 5.641600e-34 1.066973e-35 3.078680e-39
## [41] 1.066973e-35 8.893901e-37 7.513530e-38 4.204577e-40 7.152135e-31
## [46] 3.911589e-37 1.310707e-35 6.258057e-33 1.066973e-35 5.447115e-41
```

```
#fwer
a = p.adjust(pvalVector, method = "BH")< 0.1
a
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
# if I were compared this to step 1, the results are the resiprocal.
# Goes the same with  part 3, the result were surprisingly
# literally the resiprocal to p.adjust using bonferroni and bh.
# I can conclude that bonferroni and bh are the resiprocal of
# each other.
```