

Run the following codes in R.

```
> set.seed(1140411)
sample1 <- rnorm(30, 69, 2.5)
sample2 <- rnorm(30, 69, 2.5)
theData <- data.frame(values = c(sample1, sample2), group = rep(c(1, 2), each = 30))
View(theData)
boxplot(values ~ group, data = theData)
```

Both sample 1 and sample 2 in the above code are samples of size $n =$ from a distribution with . Thus, they are each sampled from . Thus, the hypothesis that $H_0 : \mu_1 = \mu_2$ is for these data.

Suppose we just had theData dataset without knowledge of the population this data came from. The permutation test of the stated null hypothesis would then be coded in R as:

```
myTest <- 
observedTestStat <- myTest$ 

N <- 2000
permutedTestStats <- rep(NA, N)
for (i in 1:N) {
  permutedData <- 
  permutedTest <- 
  permutedTestStats[i] <- permutedTest$ 
}
hist(permutedTestStats)
abline(v=observedTestStat)
sum(permutedTestStats >= observedTestStat)/N
sum(permutedTestStats <= observedTestStat)/N
```

Run the above code in R by copying and pasting the R-chunk shown below into a new R Chunk in your ClassNotes.Rmd file and filling in the places missing with the correct answers you obtained above. (You will have to type the answers in by hand, they cannot be copied and pasted.)

IMPORTANT: to match the answer key for the questions below, your R-Chunk you add to your RCheatSheets&Notes.Rmd file will need to use **set.seed()** as follows. This ensures everyone gets the same "random" data each time.

```
```{r, eval=FALSE}
Create the data:
set.seed(1140411)
sample1 <- rnorm(30, 69, 2.5)
sample2 <- rnorm(30, 69, 2.5)
```

```
theData <- data.frame(values = c(sample1,sample2), group = rep(c(1,2), each=30))
View(theData)
boxplot(values ~ group, data = theData)
```

```
Run the permutation test:
```

```
myTest <- ...
observedTestStat <- myTest$...
```

```
N <- 2000
permutedTestStats <- rep(NA, N)
for (i in 1:N) {
 permutedData <- ...
 permutedTest <- ...
 permutedTestStats[i] <- permutedTest$...
}
hist(permutedTestStats)
abline(v=observedTestStat)
sum(permutedTestStats >= observedTestStat)/N
sum(permutedTestStats <= observedTestStat)/N
...
```

observedTestStat =  (Round to 4 decimal places.)

"Greater than" p-value =  (Round to 2 decimal places.)

"Less than" p-value =  (Round to 2 decimal places.)

"Two-sided" p-value =  (Round to 2 decimal places.)