

W203 Homework 7

Natarajan Shankar

June 22, 2016

```
## two.tailed | left.tail | Action
## ----- | ----- | -----
## TRUE | TRUE | Ignore left.tail setting
## TRUE | FALSE | Ignore left.tail setting
## FALSE | TRUE | take left.tail setting
## FALSE | FALSE | go with right tail

## FUNCTION : hchecker() - Check for accepting/rejecting the Null hypothesis
## ASSUMPTIONS : Significance level is 0.05
## INPUT: Takes a sample and population metrics and does hypothesis checking
## RETURNS: pvalue and a accept/reject flag for the Null hypothesis

hChecker <- function(sample, pop.mean, pop.sd, two.tailed=TRUE, lower.tail=FALSE) {

  ## Compute the mean of this sample
  sample.mean = mean(sample)

  ## Use the standard formula to compute Z value
  z.value = (sample.mean - pop.mean)/(pop.sd/sqrt(length(sample)))

  ## SEe the reject Null hypothesis flag to false, update later in the computation
  rejectNULL = FALSE

  ## Start with checking whether two tailed test flag is FALSE. If so, process
  ## either for lower tail or upper tail
  if (two.tailed == FALSE) {
    if (lower.tail == FALSE) {
      ## do the Upper tail test because lower.tail flag is FALSE
      p.value = pnorm(z.value, lower.tail=FALSE)
    } else {
      ## do the Lower tail test because lower.tail is either set to true or is default true
      p.value = pnorm(z.value)
    }

    ## Based upon the 0.05 significance level, update the rejectNull flag
    ## If equal to or below 0.05 or equal to or above 0.95, we can reject the NULL hypothesis
    if ((p.value >= 0.95) | (p.value <= 0.05))
      rejectNULL = TRUE
  } else {
    ## path where two tailed test flag is TRUE

    ## DO NOT MULTIPLY p.value by 2 for this 2 tailed test because we are not
    ## working off a lower tail metric baseline
    p.value = pnorm(z.value, lower.tail=FALSE)

    ## Based upon the 0.05 significance level, update the rejectNULL flag
```

```

    ## if p.value is below 0.05 or above 0.95, the NULL hypothesis is rejected
    if ((p.value >= 0.95) | (p.value <= 0.05))
      rejectNULL = TRUE
  }

  ## return a list with p.vale and the associated flag
  list (p.value = p.value, rejectNULL = rejectNULL)
}

# Generate a random sample to test the hChecker function
## Choose a sample size of 130, generate a random sample, using the population mean and SD
## Test both hyptheses that are actually correct and actually incorrect

## Use body temperature measurements of a 1000 people, Use a mean of 98.249 and a standard deviation of
testPopMean <- 98.249
testPopSD <- 0.733
popSampleSize <- 10000
testSampleSize <- 1000
#Create a population of temperature data
popData <- rnorm(popSampleSize, testPopMean, testPopSD)

# Perform 4 classes of tests by manually SKEWING The mean in the samples using random constants
testData <- sample(popData, size = testSampleSize, replace = TRUE)
test1Results <- hChecker(testData, testPopMean, testPopSD, two.tailed=TRUE, lower.tail=TRUE)
test1Results

## $p.value
## [1] 0.09272057
##
## $rejectNULL
## [1] FALSE

testData <- rnorm(testSampleSize, testPopMean + 0.01, testPopSD)
test2Results <- hChecker(testData, testPopMean, testPopSD, two.tailed=TRUE, lower.tail=FALSE)
test2Results

## $p.value
## [1] 0.7318365
##
## $rejectNULL
## [1] FALSE

testData <- rnorm(testSampleSize, testPopMean - 0.05, testPopSD)
test3Results <- hChecker(testData, testPopMean, testPopSD, two.tailed=FALSE, lower.tail=TRUE)
test3Results

## $p.value
## [1] 0.03637646
##
## $rejectNULL
## [1] TRUE

```

```
testData <- rnorm(testSampleSize, testPopMean + 0.04, testPopSD)
test4Results <- hChecker(testData, testPopMean, testPopSD, two.tailed=FALSE, lower.tail=FALSE)
test4Results
```

```
## $p.value
## [1] 0.004505285
##
## $rejectNULL
## [1] TRUE
```

```
## FUNCTION : multipleSampleTesting() - Test multiple samples from the same population, through random
## INPUT: Takes a sample size (default 1000) and population metrics and repeatedly calls hChecker()
## RETURNS: pvalue that is sent by hChecker()
```

```
multipleSampleTesting <- function (sampleSize = testSampleSize, pop.mean = testPopMean, pop.sd = testPopSD)
  ## Generate a random normal sample
  #asample = rnorm(sampleSize, pop.mean, pop.sd)
  asample <- sample(popData, size = testSampleSize, replace = TRUE)

  ## Check the support for the likelihood of the Null hypothesis
  ## Limit checking to two tailed tests
  pval = hChecker(asample, pop.mean, pop.sd, two.tailed=TRUE, lower.tail=FALSE)$p.value
}
```

```
# Use replicate() to generate multiple samples with the same population variables and do a
# hypothesis testing on each sample of size 1000.
```

```
# Run a 100000 times to get a very clean Normal distribution
```

```
type1.error <- replicate(100000, multipleSampleTesting(testSampleSize, testPopMean, testPopSD))
```

```
## Determine the number of samples where Null hypothesis is likely and number where Null hypothesis is v
## by using a significance level of 0.025 on both tails, 0.05 total
```

```
significanceCheckFail <- type1.error >= 0.95 | type1.error <= 0.05
```

```
## Compute the percentage of the number of samples that reject the Null hypothesis
```

```
percent.rejected = (length(significanceCheckFail[significanceCheckFail==TRUE])/length(significanceCheckFail))
percent.rejected
```

```
## [1] 11.41316
```

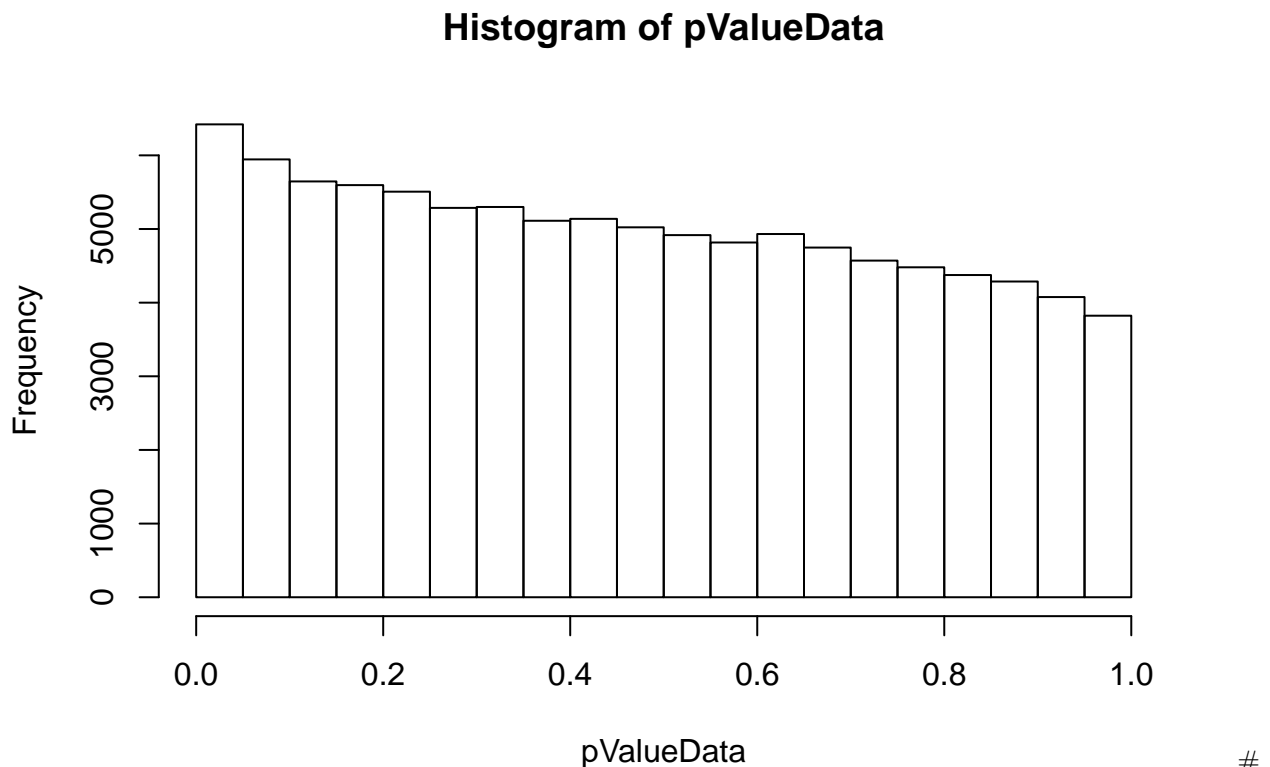
This implementation appears to converge Type1 errors at ~11.5%. This value is close to the expected value of 10% (twice the 5% at each tail).

However, I have a nagging feeling that something is not kosher here. I will investigate.

Find the data points that indicate that the Null hypothesis is indeed true

Plot a histogram

```
# Null hypothesis continues to be likley if data falls in the band around the mean  
pValueData <- type1.error[(type1.error < 0.95) | (type1.error > 0.05) ]  
hist(pValueData)
```



[shankar] My implementation does not appear to be computing errors as expected. There are some anomalies in the results. I will take a deeper look after the solution is posted. #

Bonus Question

For a random variable X, the mean is:

$$\bar{X} = 98.249$$

$$\sigma = 0.733$$

Instructions indicate that an ASSUMPTION be made that the population mean is $\text{Mean} + (0.1 * \text{SD})$

$$H_1 = \mu_0 = 98.249 + (0.1 * \mu_0) = 98.249 + 0.0733 = 98.3223$$

98.3223

This is a right tailed test. The significance level of 0.05 is being used, this corresponds to a z vlue of 1.64486

Reject the Null hypothesis if Z statistic > -1.64486

$$1.64486 = \left(\frac{X - \mu_0}{\sigma} \right)$$

$$1.64486 = \left(\frac{X - 98.3223}{0.733} \right)$$

and hence, $X = 99.45468$

$$P\left(\frac{99.45468 - 98.3223}{0.733}\right) \text{ is the probability that } Z \geq \text{the significance level of}$$

1.544884

Reading from the Z table, the type 2 error probability is 0.9388

Also, the the Statistical Power of Test is $1 - (\text{type 2 error probability})$ which is 0.0612

Reading from the Z table, the type 2 error probability is 0.9388. Also, the the Statistical Power of Test is $1 - (\text{type 2 error probability})$ 0.0612

```
popMean = 98.249
popSD = 0.733
testSampleSize = 1000
observedMean = 98.3223
```

{Bonus Question : Test via simulation}

```
# Determine the critical value of Xbar
crit <- qnorm(1-0.05, popMean, popSD)
crit
```

```
## [1] 99.45468
```

```
# Now determine the probability
pow <- pnorm(crit, observedMean, popSD)
pow
```

```
## [1] 0.9388092
```

```
# Statistical Power is 1 - pow
1 - pow
```

```
## [1] 0.06119084
```

```
# [1] 0.06119084
```

Statistical Power matches computed value of 0.0611

Bonus question, part 2

Increase the sample size and redo

```
# Sample size of 10
type2.error <- replicate(10, multipleSampleTesting(testSampleSize, observedMean, popSD))
thresholdCheck <- type2.error > 0.95
length(thresholdCheck[thresholdCheck==FALSE])
```

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

```
length(thresholdCheck[thresholdCheck==TRUE])
```

```
## [1] 9
```

```
# Sample size of 100
type2.error <- replicate(100, multipleSampleTesting(testSampleSize, observedMean, popSD))
thresholdCheck <- type2.error > 0.95
length(thresholdCheck[thresholdCheck==FALSE])
```

```
## [1] 9
```

```
length(thresholdCheck[thresholdCheck==TRUE])
```

```
## [1] 91
```

```
# Sample size of 1000
type2.error <- replicate(1000, multipleSampleTesting(testSampleSize, observedMean, popSD))
thresholdCheck <- type2.error > 0.95
length(thresholdCheck[thresholdCheck==FALSE])
```

```
## [1] 73
```

```
length(thresholdCheck[thresholdCheck==TRUE])
```

```
## [1] 927
```

As sample size increases, the Standard error is reduced because of the \sqrt{n} term in the denominator

$$\frac{\sigma}{\sqrt{n}}$$