

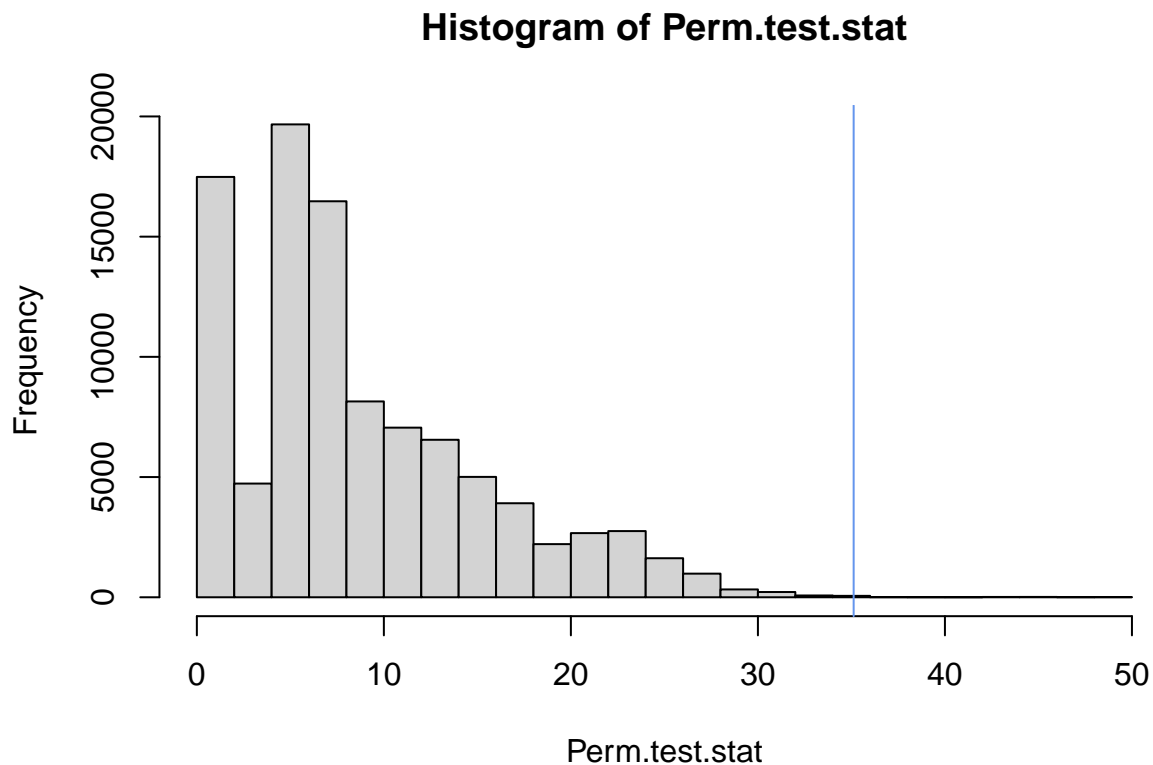
Permutation Test

Isaac

10/18/2021

Permutation Test

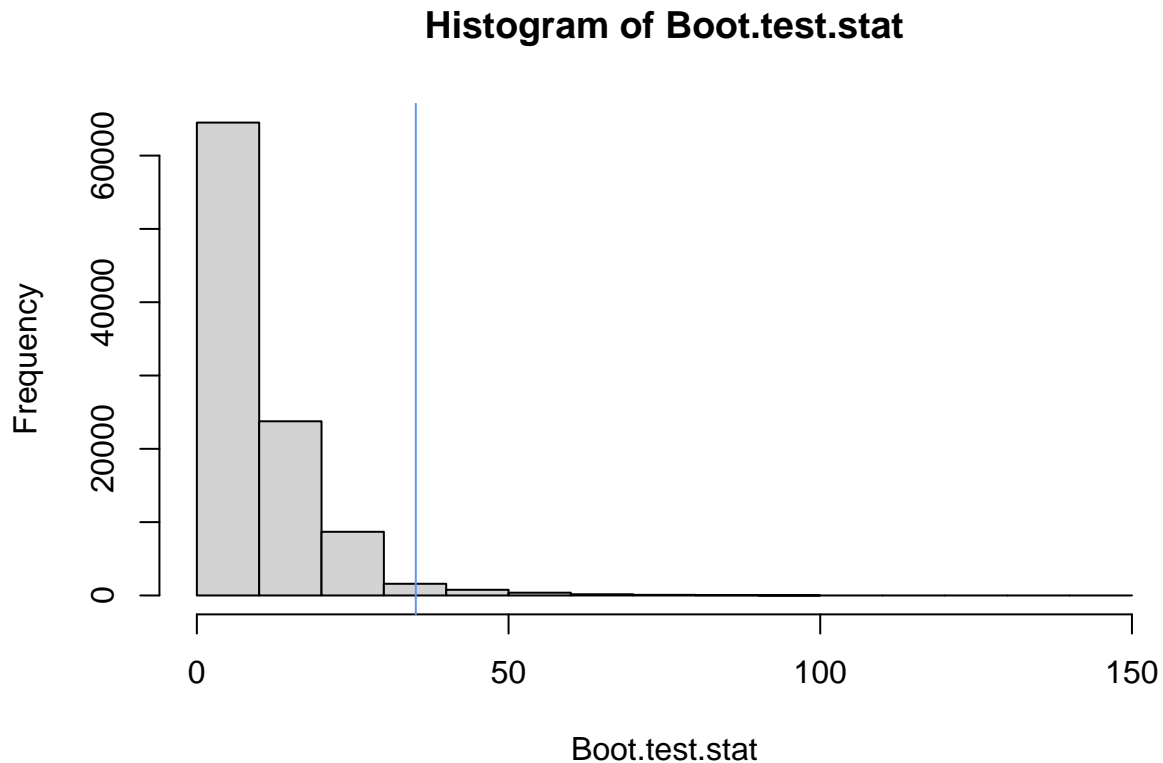
Because there are so few observations in our dataset, permutation test methods are apt to re-test the null-hypothesis for the Self-Testing Policy variable. Because there are forty observations for the two groups, it is unrealistic to take all the possible permutations, so an approximate permutation test is used on both variables, using a random sample of 10,000 permutations. Random assignment of the data to the different groups allows for a data distribution under the null hypothesis, that the two data sets do not come from a different population. This allows for the calculation of a p-value.



We receive results that are similar to both the t-test, and the Mann-Whitney test above, and receive a p-value of 7.3×10^{-4} . This mirrors the results from above.

Randomization Test

It may also be useful to use bootstrap methods to test the null hypothesis. Using 10,000 random samples (using replacement), a distribution is generated in which the alternative hypothesis can be tested. See the histogram below:



Although it is not as small as the p-value calculated using the permutation test, the p-value calculated, 0.0201, is still statistically significant compared to the alpha value of 0.05.

All three methods: non-parametric methods, permutation test, and bootstrapping methods give similar results, and any one of these would be worthwhile to test our hypothesis.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)

set.seed(142)
library(tidyverse)
library(pander)
library(flextable)

HIV <- read.csv("STAT435Exam1Data.csv", header=TRUE)
names(HIV) <- c('Country', 'DeathRate', 'WHOGuidelines', 'SelfTestingPolicy')
```

```

HIV$WHOGuidelines <- as.factor(HIV$WHOGuidelines)
HIV$SelfTestingPolicy <- as.factor(HIV$SelfTestingPolicy)

#### data cleaning

HIV_tidy <- HIV %>% drop_na %>% filter(DeathRate > 0)

SelfTest_Yes <- HIV_tidy %>% filter(SelfTestingPolicy == "Yes")
SelfTest_No <- HIV_tidy %>% filter(SelfTestingPolicy == "No")

yes <- sample(1:68, 20, replace=FALSE)
no <- sample(1:69, 20, replace=FALSE)

SelfTest_Yes <- SelfTest_Yes[yes,]
SelfTest_No <- SelfTest_No[no,]

HIV_tidy_ST <- rbind(SelfTest_Yes, SelfTest_No)

#Permutation Test on Self-Testing Policy
test.stat_ST <- abs(median(SelfTest_Yes$DeathRate) - median(SelfTest_No$DeathRate))
n <- nrow(HIV_tidy_ST)
p <- 100000
PermSamples <- matrix(0, nrow=n, ncol=p)
#take a random sample from 1 to choose(40,20) and select those columns from combn(40,20)
randomvect <- sample(1:choose(40,20), size=10000, replace=FALSE)
for(i in 1:p){
  PermSamples[,i] <- sample(HIV_tidy_ST$DeathRate, size=n, replace=FALSE)
}
PermSamples[,1:5]
Perm.test.stat <- vector("double",p)
for(i in 1:p){
  Perm.test.stat[i] <- abs(
    median(PermSamples[1:20,i]) - median(PermSamples[21:40,i])
  )
}
Perm.test.stat[1:15]
hist(Perm.test.stat)
abline(v=test.stat_ST, col="cornflowerblue")
pval_ST <- sum(test.stat_ST < Perm.test.stat)/p
print(pval_ST)

hist(Perm.test.stat)
abline(v=test.stat_ST, col="cornflowerblue")
#Bootstram Method on Self-Testing policy (median)

#Here we see our test statistic, the difference in medians for the countries with a self testing policy
# and those without.
test.stat_ST_boot <- abs(median(SelfTest_Yes$DeathRate) - median(SelfTest_No$DeathRate))
n_boot <- nrow(HIV_tidy_ST)
p_boot <- 100000
BootstrapSamples <- matrix(0, nrow=n, ncol=p)
for(i in 1:p){

```

```

  BootstrapSamples[,i] <- sample(HIV_tidy_ST$DeathRate, size=n, replace=TRUE)
}
BootstrapSamples[,1:5]
Boot.test.stat <- vector("double",p)
for(i in 1:p){
  Boot.test.stat[i] <- abs(
    median(BootstrapSamples[1:20,i]) - median(BootstrapSamples[21:40,i])
  )
}
Boot.test.stat[1:15]
pval_ST_boot <- sum(test.stat_ST_boot < Boot.test.stat)/p_boot
print(pval_ST_boot)
hist(Boot.test.stat)
abline(v=test.stat_ST, col="cornflowerblue")

```