MATH 154 - HW5

Zihao Xu

Wednesday, October 4, 2017

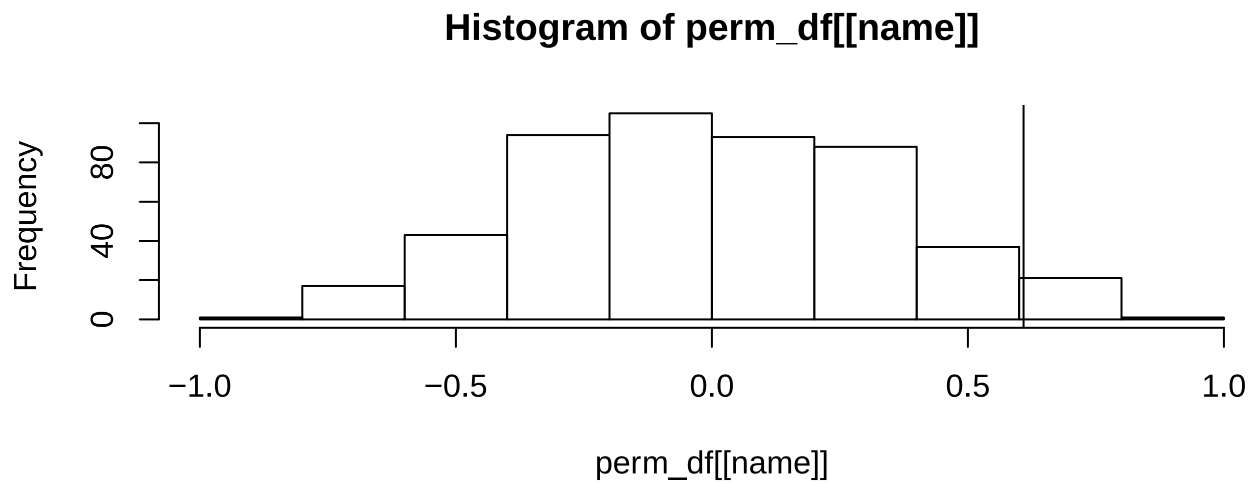
assignment

```
#library(devtools)
#devtools::install_github("statlab/permuter")
require(permuter)
data(macnell)  # let me know if the data don't install, and I can send them to you
require(NHANES)
require(dplyr)
data(NHANES)
```

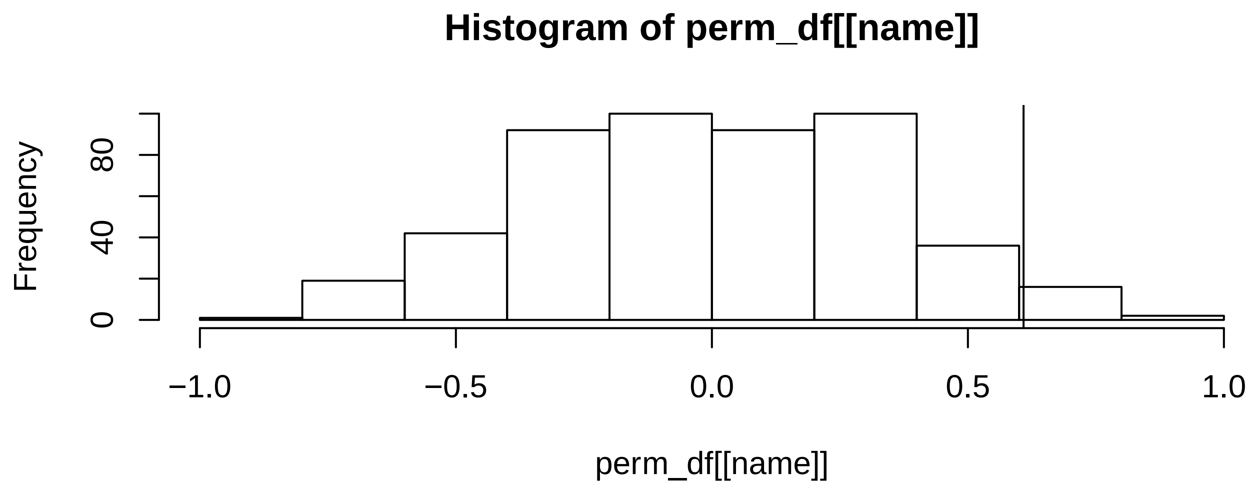
1. Consider the at least 4 course evaluation variables in the macnell data on course evaluations.

```
# (a) For each of the variables you have chosen, perform a two-sided permutation test to determine whet

set.seed(47)
reps <- 500

perm_df <- data.frame(pprofessional = numeric(0), prespect = numeric(0),
                 pcaring = numeric(0), penthusiastic = numeric(0))

for(i in 1:reps) {
   perm_df <- rbind(perm_df, macnell %>% group_by(tagender) %>%
          mutate(permTAID = sample(taidgender, replace = FALSE)) %>%
          ungroup(tagender) %>%
          group_by(permTAID) %>%
          summarize(pprofessional = mean(professional, na.rm = TRUE),
                    prespect = mean(respect, na.rm = TRUE ),
                    pcaring = mean(caring, na.rm = TRUE),
                    penthusiastic = mean(enthusiastic, na.rm = TRUE)) %>%
          summarize(pprofessional = diff(pprofessional),
                    prespect = diff(prespect),
                    pcaring = diff(pcaring),
                    penthusiastic = diff(penthusiastic)))
}

obs_df <- data.frame(pprofessional = numeric(0), prespect = numeric(0),
                 pcaring = numeric(0), penthusiastic = numeric(0))

obs_df <- rbind(obs_df, macnell %>% group_by(taidgender) %>%
                summarize(pprofessional = mean(professional, na.rm = TRUE),
                                  prespect = mean(respect, na.rm = TRUE ),
                                  pcaring = mean(caring, na.rm = TRUE),
                                  penthusiastic = mean(enthusiastic, na.rm = TRUE))%>%
                summarize(pprofessional = diff(pprofessional),
                                  prespect = diff(prespect),
                                  pcaring = diff(pcaring),
                                  penthusiastic = diff(penthusiastic)))
```

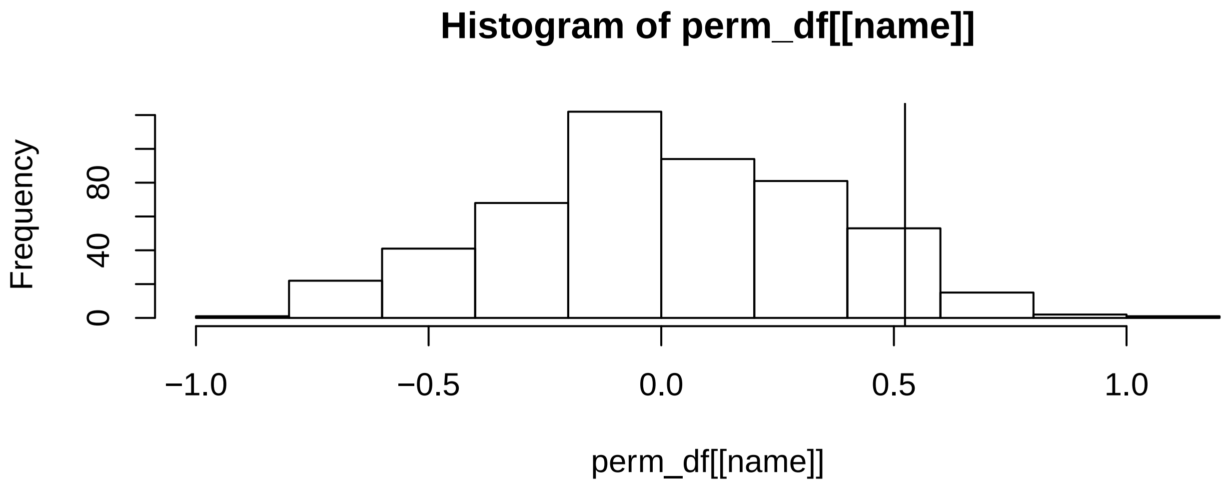
```
for(name in names(obs_df)){
    hist(perm_df[[name]]); abline(v = obs_df[[name]])
    p_value <- sum(abs(perm_df[name]) > abs(obs_df[[name]])) / nrow(perm_df[name])
    if(p_value > 0.05){
        print(paste("A two-sided test for\"", name, "\" gives p_value", p_value,
                    "suggesting that gender perception does not impact student evaluation.
                    And we fail to reject the null."))
    }else{
        print(paste("A two-sided test for\"", name, "\" gives p_value", p_value,
                    "suggesting that gender perception does impact student evaluation.
                    And we reject the null."))
    }
}
```



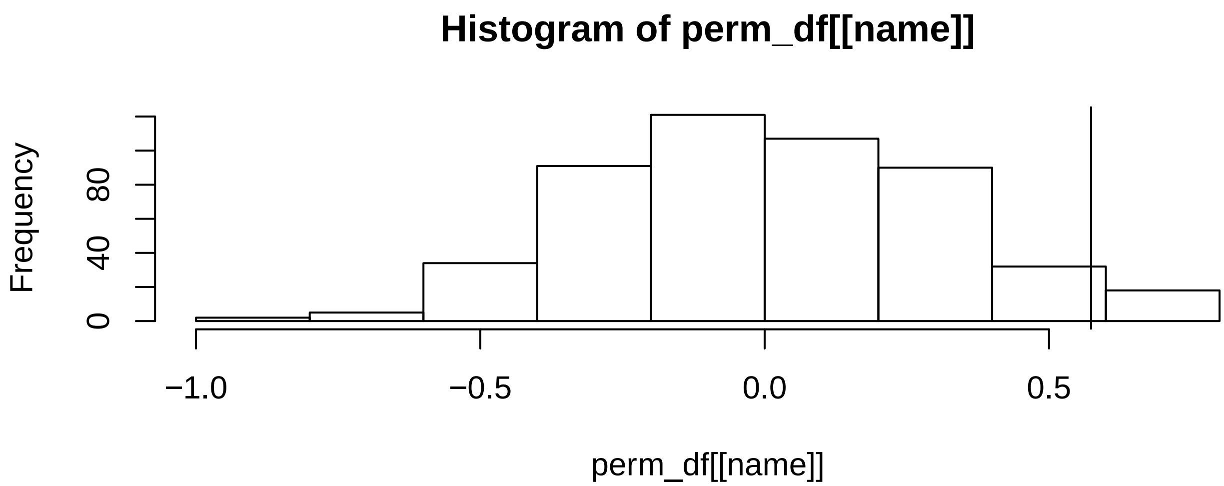

```
## [1] "A two-sided test for\" pprofessional \" gives p_value 0.062 suggesting that gender perception do
```




```
## [1] "A two-sided test for\" prespect \" gives p_value 0.054 suggesting that gender perception does no
```




```
## [1] "A two-sided test for\" pcaring \" gives p_value 0.094 suggesting that gender perception does no
```




```
## [1] "A two-sided test for\" penthusiastic \" gives p_value 0.05 suggesting that gender perception do
```

```
Output from Console:
```

[1] "A two-sided test for" pprofessional " gives p_value 0.062 suggesting that gender perception does not impact student evaluation. And we fail to reject the null." [1] "A two-sided test for" prespect " gives p_value 0.054 suggesting that gender perception does not impact student evaluation. And we fail to reject the null." [1] "A two-sided test for" pcaring " gives p_value 0.094 suggesting that gender perception does not impact student evaluation. And we fail to reject the null." [1] "A two-sided test for" penthusiastic " gives p_value 0.05 suggesting that gender perception does impact student evaluation. And we reject the null."

```r
# (b) Use Fisher's combining rule to determine whether or not all of the null hypotheses from part (a) 
p <- c(0.062, 0.054, 0.094, 0.05)


combined_p_value <- pchisq((sum(log(p))*-2), df=length(p)*2, lower.tail=F)

if(combined_p_value > 0.05){
  print(paste("The combined p_value is:", combined_p_value, ". And we conclude that
              gender perception does not impact student evaluation for all of these
              variables. We fail to reject the null."))
}else{
  print(paste("The combined p_value is:", combined_p_value, ". And we conclude that
```

```
              gender perception does impact student evaluation for all of these
              variables. We reject the null."))
}
```

```
## [1] "The combined p_value is: 0.00469982365426236 . And we conclude that \n              gender perc
```

```
Output: [1] "The combined p_value is: 0.00469982365426236 . And we conclude that \n              gender
    variables. We reject the null."
```

```
# (c) Given your results from (a) and (b) comment on what the data say (or don't say)
# about bias from perceived gender.  You might want to speak about sample size,
# power, and effect size.
```

```
Based on my results on the previous two parts, I would say that there exist no
significant difference between mean scoring (for these 4 variables) based on gender
perceptions. But I am hesitant to make a definite conclusion since: 1. the sample size is     too small
that are significantly different; 3. power of test is small since sample size is small;
```

2. From the NHANES data... Consider the two group scenario: Poverty (a ratio of family income to poverty guidelines. Smaller numbers indicate more poverty.) as broken down by Smoking (SmokeNow variable). The test of *difference in shift* can be modeled by a difference in **means** or by a difference in **medians**. Question: which test is more powerful? To answer the question, follow the steps below.

```
# Two vectors to store the powers of each test
mean_diff_power <- c()
meidan_diff_power <- c()

sample_size <- 50

for( i in 1:100 ){
    df_cutoff <- data.frame(mean_diff = numeric(0), median_diff = numeric(0))


    df_alt <- data.frame(mean_diff = numeric(0), median_diff = numeric(0))

    # Step 1:
      # The data comes from a null population because the valule of
      # Poverty is randomly assigned
      # for the smokenow indicator. And this assumes that there
      # are no difference in mean Poverty for these two groups.

    NHANES_NULL <- NHANES %>%
        select(Poverty, SmokeNow) %>%
        filter(!is.na(SmokeNow)) %>%
        mutate(pPoverty = sample(Poverty, replace=FALSE))

    # Step 2:
      # Now the data is from the alternative population, because we
      # know that these exists a difference between Poverty values
      # for smokers and non-smokers.

    NHANES_ALT <- NHANES_NULL%>%
        mutate(pPoverty = pPoverty + ifelse(SmokeNow == "No", 0.1, 0))


    df_cutoff <- rbind(df_cutoff,  NHANES_ALT %>%
```

```r
        group_by(SmokeNow)%>%
        summarize(pmean = mean(pPoverty, na.rm = TRUE),
                  pmedian = median(pPoverty, na.rm = TRUE)) %>%
        summarize(mean_diff = diff(pmean), median_diff = diff(pmedian))

    for (j in 1:sample_size){

      df_alt <- rbind(df_alt, NHANES_ALT %>%
          mutate(pPoverty = sample(pPoverty, replace=FALSE)) %>%
          group_by(SmokeNow)%>%
          summarize(pmean = mean(pPoverty, na.rm = TRUE),
                    pmedian = median(pPoverty, na.rm = TRUE)) %>%
          summarize(mean_diff = diff(pmean), median_diff = diff(pmedian)))
    }

    # Step 3 and 4: calculate the power of each test for 10 times
    mean_cut_off <- df_cutoff$mean_diff
    median_cut_off <- df_cutoff$median_diff

    p_val_mean <- sum(df_alt$mean_diff <= mean_cut_off)/sample_size
    p_val_median <- sum(df_alt$median_diff <= median_cut_off)/sample_size
    mean_diff_power <- c(mean_diff_power, p_val_mean<=0.05)
    meidan_diff_power <- c(meidan_diff_power, p_val_median<=0.05)
}

# print out the average power of each test
print(paste("Mean power of mean_diff is:", mean(mean_diff_power)))
```

```
## [1] "Mean power of mean_diff is: 0.57"
```

```r
print(paste("Variance of power of mean_diff is:", var(mean_diff_power)))
```

```
## [1] "Variance of power of mean_diff is: 0.247575757575758"
```

```r
print(paste("Mean power of median_diff is:", mean(meidan_diff_power)))
```

```
## [1] "Mean power of median_diff is: 0.37"
```

```r
print(paste("Variance of power of median_diff is:", var(meidan_diff_power)))
```

```
## [1] "Variance of power of median_diff is: 0.235454545454545"
```

`Therefore, we can see that the mean_diff is a more powerful test than median_diff.`

3. Let the parameter π_R represents the true p-value given by enumerating every single possible permuation (assume the data have already been collected, so the *observed* data is already given/fixed). For *one* permutation, the probability that the test will reject H_0 is π_R. Said differently, for each permutation, the result can be thought of as a Bernoulli random variable:

$$X_i = \begin{cases} 1 & i^{th} \text{ perm stat} \geq \text{observed stat} \\ 0 & i^{th} \text{ perm stat} < \text{observed stat} \end{cases}$$

Consider the case where a permutation test was run 1000 times and the empirical p-value was found to be 0.047. Give a confidence interval for the true *randomization* p-value which would have been calculated if the test was run on all possible permutations of the data.

Solution: Since we know that for each one permutation, the result is a Bernoulli variable, the variance associated with a series of Bernoulli variable is a Binomial (n, π_R). From the distribution, we know that the variance of the empirical p-value is $\frac{np(1-p)}{n} = p(1-p)$, then the standard deviation of empirical p-value is $\sqrt{p(1-p)} = 0.21163$. Therefore, a 95% confidence interval should be $CI = 0.047 \pm 1.96 * 0.21163 = 0.047 \pm 0.4148$ (since we know p-value should be larger than 0, maybe $CI = [0, 0.4618]$)