MATH 154 - HW5 - Permutation Tests

Ra-Zakee Muhammad

due: Thursday, October 7, 2021

Kamil and Tianna helped me with this pset for number 2 and 3 respectively

summary

In this assignment you will expand what you know about the standard logic of hypothesis testing (from previously using calculus and limit theorems) to using computational techniques. That is, instead of focusing on the asymptotic distribution of a statistic (to get its sampling distribution), the sampling distribution of the statistic will be formed from repeated permutations of the data under the null hypothesis.

requisites

In the homework, we will use two datasets which are available in R packages.

This is survey data collected by the US National Center for Health Statistics (NCHS) which has conducted a series of health and nutrition surveys since the early 1960's. Since 1999 approximately 5,000 individuals of all ages are interviewed in their homes every year and complete the health examination component of the survey. The health examination is conducted in a mobile examination center (MEC). [Data from 2009-2012 with adjusted weighting.]

Boring et al. "Student evaluations of teaching (mostly) do not measure teaching effectiveness" (2016) reanalyze data from MacNell et al. "What's in a Name: Exposing Gender Bias in Student Ratings of Teaching" (2014). Students were randomized to 4 online sections of a course. In two sections, the instructors swapped identities. Was the instructor who identified as female rated lower on average?

assignment

```
data(macnell)
data(NHANES)
```

0. `reprex()` A problem which is not due. On your exam, you will be required to email me a reproducible example. That is, you'll ask a question (e.g., "why don't the lines connect to the dots?") and have code which demonstrates what the code is doing. Good idea to practice now in creating reprexes! that is, as you get stuck, try to come up with a question you could ask which would help getting unstuck. You don't have to use the `reprex()`function, but sometimes it helps. Here is some advice: https://stackoverflow.com/help/minimal-reproducible-example Email me reprexes! Post reprexes to Discord!

Note that your reprex might be to create an error! The key is that when you send me the code, I should be able to run the code and get the exact same error that you get.

1. **Pod Q** Describe one thing you learned from someone in your pod this week (it could be: content, logistical help, background material, R information, something fun, etc.) 1-3 sentences.

I leaned from Kamil that the sample size and power are positively correlated.

2. **Gender and evaluation** 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 whether or not to reject the null hypothesis that gender perception does not impact student evaluation.

fair, responsive, knowledgeable, clear

#fair

```
macnell %>%
  group_by(taidgender) %>%
  summarize(pmeans = mean(fair, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
```

```
## # A tibble: 1 x 1
##   diff_mean
##       <dbl>
## 1     0.761
```

```
diff_means_fai <- function(.x){
  macnell %>% group_by(tagender) %>%
  mutate(permTAID = sample(taidgender, replace=FALSE)) %>%
  ungroup(tagender) %>%
  group_by(permTAID) %>%
  summarize(pmeans = mean(fair, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
}

map_df(1:5, diff_means_fai)
```

```
## # A tibble: 5 x 1
##   diff_mean
##       <dbl>
## 1   -0.275
## 2   -0.0130
## 3    0.267
## 4   -0.0974
## 5    0.182
```

```
set.seed(47)
reps = 1000
perm_diff_fair<- map_df(1:reps, diff_means_fai)
```

```
perm_diff_fair %>% filter(abs(diff_mean) > 0.7608696 ) %>% summarise(pval = n()/1000)
```

```
## # A tibble: 1 x 1
##    pval
##   <dbl>
## 1  0.01
```

#responsive

```
macnell %>%
  group_by(taidgender) %>%
  summarize(pmeans = mean(responsive, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
```

```
## # A tibble: 1 x 1
##   diff_mean
##       <dbl>
## 1     0.220
```

```
diff_means_res <- function(.x){
  macnell %>% group_by(tagender) %>%
  mutate(permTAID = sample(taidgender, replace=FALSE)) %>%
  ungroup(tagender) %>%
  group_by(permTAID) %>%
  summarize(pmeans = mean(responsive, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
}

map_df(1:5, diff_means_res)
```

```
## # A tibble: 5 x 1
##   diff_mean
##       <dbl>
## 1    0.341
## 2   -0.149
## 3    0.0823
## 4    0.175
## 5   -0.197
```

```
set.seed(47)
reps = 1000
perm_diff_responsive<- map_df(1:reps, diff_means_res)
```

```
perm_diff_responsive %>%
  filter(abs(diff_mean) > 0.219) %>%
  summarise(pval = n()/1000)
```

```
## # A tibble: 1 x 1
##    pval
##   <dbl>
## 1  0.51
```

#knowledgable

```
macnell %>%
  group_by(taidgender) %>%
  summarize(pmeans = mean(knowledgeable, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
```

```
## # A tibble: 1 x 1
##   diff_mean
##       <dbl>
## 1     0.354
```

```
diff_means_kno <- function(.x){
  macnell %>% group_by(tagender) %>%
  mutate(permTAID = sample(taidgender, replace=FALSE)) %>%
  ungroup(tagender) %>%
  group_by(permTAID) %>%
  summarize(pmeans = mean(knowledgeable, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
}

map_df(1:5, diff_means_kno)
```

```
## # A tibble: 5 x 1
##   diff_mean
##       <dbl>
## 1   0.207
## 2  -0.344
## 3  -0.552
## 4   0.286
## 5   0.00649
```

```
set.seed(47)
reps = 1000
perm_diff_know<- map_df(1:reps, diff_means_kno)
```

```
perm_diff_know %>% filter(abs(diff_mean) > 0.3543478) %>% summarise(pval = n()/1000)
```

```
## # A tibble: 1 x 1
##    pval
##   <dbl>
## 1 0.277
```

#clear

```
macnell %>%
  group_by(taidgender) %>%
  summarize(pmeans = mean(clear, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
```

```
## # A tibble: 1 x 1
##   diff_mean
##       <dbl>
## 1     0.413
```

```
diff_means_cl <- function(.x){
  macnell %>% group_by(tagender) %>%
  mutate(permTAID = sample(taidgender, replace=FALSE)) %>%
  ungroup(tagender) %>%
  group_by(permTAID) %>%
  summarize(pmeans = mean(clear, na.rm=TRUE)) %>%
  summarize(diff_mean = diff(pmeans))
}

map_df(1:5, diff_means_cl)
```

```
## # A tibble: 5 x 1
##   diff_mean
##       <dbl>
## 1    0.335
## 2   -0.160
## 3   -0.0130
## 4    0.359
## 5   -0.385
```

```
set.seed(47)
reps = 1000
perm_diff_clear<- map_df(1:reps, diff_means_cl)
```

```
perm_diff_clear %>% filter(abs(diff_mean) > 0.413) %>% summarise(pval = n()/1000)
```

```
## # A tibble: 1 x 1
##    pval
##   <dbl>
## 1 0.299
```

Given the tools that we have, **unfortunately** you'll need to write 5 separate functions (it is beyond what we are doing to pass a variable name into the function argument).

b. Use Fisher's combining rule to determine whether or not all of the null hypotheses from part (a) are true. https://en.wikipedia.org/wiki/Fisher%27s_method [Note that this is **not** Fisher's Exact test, so `fisher.test` is incorrect to use in this case. The two tests have completely different data structures and different hypotheses. Unfortunately, both are named after R.A. Fisher who is someone we'd prefer not to celebrate.]

```
1-pchisq(15.5391, df = 8 )
```

```
## [1] 0.04947265
```

Not all null hypothesis are true because the p-value is less than .05 for the chi-squared distribution. Additionally the p-value for fair is .01 which is less than our .05 threashold allowing us to reject h0

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.

small sample size doesnt provide enough power small sample size doesnt show the effects of percieved gender if the effect size is greater we wont need a larger sample size

3. **Power** The goal of the following problem is to understand power. To calculate power, the alternative hypothesis must be true. To insure the alternative hypothesis is true, we force it to be true by first making the data null, then adding a shift to one group so that there is no doubt of exactly what the population structure is.

From the `NHANES` data, consider the two group scenario: `BPSysAve` - "Combined systolic blood pressure reading, following the procedure outlined for BPXSAR." as broken down by Smoking

(`SmokeNow` variable - "Study participant currently smokes cigarettes regularly. Reported for participants aged 20 years or older as Yes or No,

provided they answered Yes to having smoked 100 or more cigarettes in their life time.

All subjects who have not smoked 100 or more cigarettes are listed as NA here.").

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 (the vast majority of the code has been written for you).

Recall that power is the probability of rejecting the null hypothesis when in fact the *alternative hypothesis is true*. To estimate power (for different statistics), we will simulate data (based on NHANES) where we know the alternative hypothesis to be true. Note here that we won't actually use the original labels (and their significance) for the analysis.

n.b. The reason we aren't using the original labels (for the alternative data) is because we don't know capital-T Truth (i.e., which is true, the null or alternative?) with the original labels. Instead, if we permute the labels (so there is no difference) and then add a bit to one group (so that there is a difference), we can create a scenario where the capital-T Truth is known (i.e., the alternative is known to be true).

Step 1 Randomly assign the Poverty variable to the two smoking groups. yes

(a) Explain the line of code marked "step 1" below. Does the now-randomly-assigned-dataset at hand (from step 1) come from a null population (pop under null h) or an alternative population? Explain.

it comes from BPSysAve which is a null population because the difference of means is zero since you are only permuting

Step 2 Add 0.1 to each of the non-smokers' household poverty ratio.

(b) Explain the line of code marked "step 2" below. Does the now-0.1-added-dataset at hand come from a null population or an alternative population? Explain.

it comes from an alternative population where all the smokers in the yes catagory have blood pressure raised by 1.5 so the difference oof means is no longer 0.

Step 3 Permute the dataset from Step 2 and run a two-sample permutation test with pseudo code as follows:

(c) Explain the line of code marked "step 3" below. How many times is the function `perm_the_data()` run? What is the input to a single call of the function `perm_the_data()`? When we `map()` the function `perm_the_data()` what is the object being passed into `perm_the_data()`?

in the nhanes alt data a new column of data sets is mutated whose data frames are observed and permuted mean and median differences for each data frame in the data column. 50 times for 50 data frames in nhanes_alt_data column data. A single data frame from the data column of nhanes alt data is the input to a single call. The entire column of data frames in nhanes alt data called "data" is being passed into the per_the_data func.

Step 4 Repeat steps 1-3 times.

(d) Explain the line of code marked "step 4" below. How many times is the entire process repeated?

The command iterates from 1 through 100 the following commands ending at line 3. 100 unique Nhanes_alt_data dataframes of 50 dataframes are generated with for each of the 100 Nhanes_alt_data a mutated column providing dataframes of stats one for each of the entries in the data column. 5000

Step 5 measure how often the non-null dataset at hand crosses the threshold of significance. Which test is more powerful? [Use the empirical results to answer the question about power.]

(e) For step 5, write code (you'll need only tidy verbs applied to `results_df`) to calculate the empirical power separately for the mean permutation test and the median permutation test.

```
# The function which will run the permutation test
perm_the_data <- function(df){
    df %>%
    select(BP_alt, SmokeNow) %>%
    mutate(BP_perm = sample(BP_alt, replace=FALSE)) %>%
    group_by(SmokeNow) %>%
    summarize(BP_mean_perm = mean(BP_perm, na.rm=TRUE),
              BP_mean_obs = mean(BP_alt, na.rm = TRUE),
              BP_med_perm = median(BP_perm, na.rm=TRUE),
              BP_med_obs = median(BP_alt, na.rm = TRUE))  %>%
    summarize(BP_mean_diff_perm = diff(BP_mean_perm),
              BP_mean_diff_obs = diff(BP_mean_obs),
              BP_med_diff_perm = diff(BP_med_perm),
              BP_med_diff_obs = diff(BP_med_obs))
}
```

```
# Creating lots of different alternative datasets
set.seed(47)
n_datasets <- 50
n_perms <- 100

NHANES_alt_data <- tibble(
  set = 1:n_datasets,
  data = map(1:n_datasets, ~NHANES %>%
  select(BPSysAve, SmokeNow) %>%
  filter(!is.na(SmokeNow) & !is.na(BPSysAve)) %>%
  mutate(BP_alt = sample(BPSysAve, replace=FALSE)) %>%  # step 1
  mutate(BP_alt = BP_alt + ifelse(SmokeNow == "Yes", 1.5, 0)) # step 2
  ) )
```

```
# Run a permutation test for each of the 50 alternative datasets
perm_results <- 1:n_perms %>%  # step 4
  map_df(~NHANES_alt_data %>%
  mutate(perm_stats = map(data, perm_the_data))) # step 3
```

```r
# un-nesting to get the relevant permuted statistics
results_df <- perm_results %>%
  select(set, perm_stats) %>%
  unnest() %>%
  arrange(set)

results_df
```

```
## # A tibble: 5,000 x 5
##      set BP_mean_diff_perm BP_mean_diff_obs BP_med_diff_perm BP_med_diff_obs
##    <int>             <dbl>            <dbl>            <dbl>           <dbl>
##  1     1           -1.06               1.31           -0.5              0.5
##  2     1           -0.145              1.31           -0.5              0.5
##  3     1           -0.293              1.31            0                0.5
##  4     1            1.00               1.31            0.75             0.5
##  5     1           -0.781              1.31           -1.5              0.5
##  6     1           -0.124              1.31           -1.5              0.5
##  7     1            0.668              1.31            0.75             0.5
##  8     1            0.0958             1.31           -1                0.5
##  9     1           -0.0575             1.31           -0.5              0.5
## 10     1           -0.184              1.31           -0.5              0.5
## # ... with 4,990 more rows
```

```r
results_df %>%
  group_by(set) %>%
  summarise(pvaluemean = mean(BP_mean_diff_perm >= BP_mean_diff_obs) ,
            pvaluemedian = mean(BP_med_diff_perm >= BP_med_diff_obs ) ) %>%
  summarise(powermean = mean(pvaluemean < .05 ) ,
            powermedian = mean(pvaluemedian < .05))
```

```
## # A tibble: 1 x 2
##   powermean powermedian
##       <dbl>       <dbl>
## 1      0.68        0.52
```

(f) Which test is more powerful, the mean or median permutation test? Explain the results as if to a science colleague who is trying to decide which test to use for their analysis.

mean is more powerful because the power generated by mean test is greater