HW2

● Graded

Student

Sangwon Ji

Total Points

72 / 75 pts

Question 1

Question 1 **15** / 15 pts

1.1 a **5** / 5 pts

✓ **– 0 pts** Correct:

My answer is makePermutedStats takes two groups and a function as inputs.
It shuffles the labels for group1 and group2 and computes statistics using the function FUN provided as an input to the function.
The output is the computed statistic for the shuffled group labels using FUN.

– 1 pt Incomplete answer - does not describe the input and/or output.

1.2 b **Resolved** **5** / 5 pts

✓ **– 0 pts** Correct:

- creates a new vector with `group1` and `group2` repeated the length of each.
- samples from created labels with a size equal to the sum of the length of the two groups with replacement.
- subsets y based on permutated group labels to create shuffled groups, and then passes the shuffled groups to FUN to get a statistic

– 1 pt Incomplete answer: misses to explain a line or incorrect answer provided for a line

replace = TRUE instead of replace = FALSE

↻ Regrade Request **Submitted on: Mar 15**

For this question, I saw on ed that it should be replace = TRUE by professor that it's an error. It should be with the replacement I believe.

Score adjusted

Reviewed on: Mar 23

1.3 c **5** / 5 pts

✓ **– 0 pts** Correct:

- computes the statistic for group1 and group2 using the FUN function. This is the observed statistic.
- the makePermutedStats gives output of a single permutation. The line runs the makePermutedStats `repitition` number of times to get a distribution.
- To compute the pvalue, we calculate the ratio of number of times we observe a permuted stats as large or larger than the observed statistic over the total numbe of repetitions
- the line combines the pvalue, observedStat, and permutedStats into a list and returns the list

– 1 pt Incomplete answer

Question 2

Question 2 — **30** / 30 pts

2.1 a — **15** / 15 pts

✓ **− 0 pts** Correct:

```
f <- function() {
sample1 <- rnorm(20, mean = 10, sd = 2.5)
sample2 <- rnorm(20, mean = 10, sd = 5)
return(t.test(sample1, sample2)$p.value)
}
rep.pvalues <- replicate(10000, f())
type1error <- sum(rep.pvalues < 0.05)/10000
type1error
```

p-value of around 0.05

− 1 pt Uses wrong mean/sd

− 1 pt incorrect code for returing p-value in f()

− 2 pts Incorrect p-value calculation

2.2 b — **15** / 15 pts

✓ **− 0 pts** Correct:

```
set.seed(2017021328)
f <- function() {
sample1 <- rgamma(20,shape = 1,rate = 3)
sample2 <- rgamma(20,shape =1 , rate =3)
return(t.test(sample1, sample2)$p.value)
}
REPLICATION = 10000
rep.pvalues <- replicate(REPLICATION, f())
type1error <- sum(rep.pvalues < 0.05)/10000
type1error
```

− 1 pt incorrect rate/shape parameters

− 2 pts Errors in computing type1error

− 2 pts wrong function used

Question 3

Question 3 **27** / 30 pts

3.1 a **5** / 5 pts

✓ **− 0 pts** Correct:

```
# code for running t-test
angina <- heart$chol[heart$cp %in% c(1, 2)]
nonangina <- heart$chol[heart$cp == 3]
t.test(angina, nonangina)
```

We do not have enough evidence to reject the null hypothesis that the true difference in mean cholesterol is equal to 0.

− 1 pt Does not select the right groups

− 1 pt Inadequate/erroneous Explaination

− 5 pts Incorrect / Missing Answer

− 2 pts Wrong Function Used

3.2 b **Resolved** 3 / 5 pts

− 1 pt errors in computing t-stat

− 0 pts Correct:

```
set.seed(2017021328)
calc.t.stat <- function(group1, group2) {
return(t.test(group1, group2)$statistic)
}
permut.chol <- permutation.test(
angina, nonangina, calc.t.stat, repetitions = 1000)
p.value = sum(
abs(permut.chol$permutedStats) > abs(permut.chol$observedStat)) / length(permut.chol$permutec
p.value
```

Based on the permutation test, we do not have enough evidence to reject the null hypothesis that the true difference in mean cholestoral is equal to 0 either.

✓ **− 2 pts** errors in computing pvalue

− 5 pts Missing Answer

Regrade Request **Submitted on: Mar 15**

Hi,
isn't this the correct formula for getting the p-value?

For this question, we want to find "permutedStats > OBSERVED Stat , and we want to apply abs() to both of these values.

Reviewed on: Mar 18

3.3 **c** **14** / 15 pts

− 0 pts Correct:

```
curve(dt(x, df = 144.25), xlim = c(-10, 10), ylim = c(0, 0.4),
main = "Density curve of permuted statistics and
t-test statistics under the null",
ylab = "density")
lines(density(permut.chol$permutedStats), col = "red")
legend("topright", legend = c("T-test", "Permutation test"),
col = c("black", "red"), lty = 1)
```

The two curves are similar in shape and position.

✓ **− 1 pt** Incorrect df input

− 1 pt misses axis label/title

− 4 pts Did not graph density

− 15 pts Missing Answer

− 4 pts Missing Explaination

3.4 **d** **5** / 5 pts

✓ **− 0 pts** Correct:

This is an open-ended question. Here is an answer just for reference. Sometimes we do not care about the exact values of the blood pressure but we only care about whether it is above 120 (or another threshold). In this case, we may create statistic by taking the indicator of blood pressure > 120.

− 5 pts Missing answer

Question 4

Late Penalty

0 / 0 pts

✓ **− 0 pts** On time

− 7.5 pts 1 day late

− 15 pts 2 days late

− 22.5 pts 3 days late

− 40 pts Past the deadline

HW2

STAT 131A, Spring 2023

Due Monday, February 27

Question 1. Below is code that implements a permutation test for an arbitrary test statistic. This homework question will ask you to explain this function.

```
makePermutedStats<-function(group1, group2, FUN){
    labels<-c(rep("group1",length=length(group1)),rep("group2",length=length(group2)))
     y<-c(group1,group2)
     permutedLabels<-sample(labels,size=length(group1)+length(group2),replace=FALSE)
     return(FUN(y[permutedLabels=="group1"], y[permutedLabels=="group2"]))
  }
permutation.test <- function(group1,group2, FUN, repetitions){
  stat.obs <- FUN(group1,  group2)
  stat.permute <-replicate(repetitions,makePermutedStats(group1,group2,FUN))
  p.value <- sum(stat.permute >= stat.obs) / repetitions
  return(list(p.value=p.value,observedStat=stat.obs,permutedStats=stat.permute))
}
```

I create simulated data in two groups to illustrate it:

```
group1Data<-rnorm(100,0,1)
group2Data<-rnorm(50,0,2)
```

Then I can define a function that calculates my statistic and run the function like so:

```
mystatistic<-function(group1,group2){abs(median(group1)-median(group2))}
outcome<-permutation.test(group1=group1Data,group2=group2Data,FUN=mystatistic,repetitions=1000)
```

a. (5 points) `permutation.test` uses the function `makePermutedStats`. Give an overall description of `makePermutedStats`, explaining what this function takes as input and what it provides as output.

You can use the above simulated data to check yourself.

> My answer is that makePermutedStats takes in doing the single permutation test of a data from the two groups create the labels of the same length as of the two groups, for these groups, sample them through repititions with non-replacement and for output it returns statistic.

b. (5 points) I've copied the code of `makePermutedStats` below. Following each of the `#` characters below, provide a short description to explain what each line of this function does. If you run into any wrapping problems, just hit return and start another line with `#`. (I've set `eval=FALSE` in the chunk here so that you don't have any danger of accidentally messing up the code with your comments.)

```
  makePermutedStats<-function(group1, group2, FUN){
    # It creates a vector, of the two groups. Each label will have equivalent number of label with the
    labels<-c(rep("group1",length=length(group1)),rep("group2",length=length(group2)))
    # It is labeling the dependent variable of input into a vector
    y<-c(group1,group2)
```

```
    # It is the sample from the vector of labels by the length of the vector
    permutedLabels<-sample(labels,size=length(group1)+length(group2),replace=FALSE)
    # It is putting the output by running the function with the input given of the groups of data with
    out<-FUN(y[permutedLabels=="group1"], y[permutedLabels=="group2"])
    return(out)
  }
```

c. (5 points) Now explain what `permutation.test` does, i.e. what the 4 lines after the definition of `makePermutedStats` do. Make sure you explore the `outcome` variable above from my little example so you understand what is returned and you can explain it below.

```
permutation.test <- function(group1,group2, FUN, repetitions){
  # It is showing the function that is inputing the groups, FUN, a value that varies with the input of
  stat.obs <- FUN(group1,  group2)
  # It calculates the observed statistics of the groups with the FUN
  stat.permute <-replicate(repetitions,makePermutedStats(group1,group2,FUN))
  # It calculates the permuted stat and repeats the replication for given time
  p.value <- sum(stat.permute >= stat.obs) / repetitions
  # It calculates the p-value, in this case proportion of permuted stat greater or equal to observed st
  return(list(p.value=p.value,observedStat=stat.obs,permutedStats=stat.permute))
}
```

Question 2. Evaluating whether a test gives valid p-values In this problem, we will go through a simple example of how you can use simulation from known distributions to determine whether a hypothesis test will perform well. We will only compare very simple settings, but this gives an idea of how you can use simulation to explore the performance of a test.

Assume your rule for rejecting the null hypothesis is when p-value $\leq \alpha$. α might be 0.05 or 0.01, for example. A test makes a Type I error if it *wrongly* rejects the null hypothesis when in fact the null hypothesis is true, i.e. the null hypothesis is true but p-value $\leq \alpha$.

Of course all tests will make errors. A valid hypothesis test procedure will ensure that on average, the chance you make a type I error is no more than the α you chose as your cutoff. In otherwords, if you perform a hypothesis test at level 0.05 *when in fact the null hypothesis is true*, then you will incorrectly reject the null hypothesis 5% of the time.

a. (15 points) Estimate the Type I error of the t-test when the data of the two groups is normal. Specifically, repeat the following simulation 10,000 times using the `replicate` function:

1) Simulate two groups of data each with 20 observations from a normal distribution. For the first group, let the standard deviation be 2.5, and for the second group the standard deviation be 5; both groups have mean 10.

2) Calculate the p-value of the t-test on this simulated data.

3) Based on the 10,000 p-values, report the type I error of the t-test, i.e. the proportion of tests that rejected the null.

```
# Enter code here for function that creates a single simulation of the data
# and returns the p-value for that single simulation:
f<-function(){
  group1 <- rnorm (20, mean = 10, sd = 2.5)
  group2 <- rnorm (20, mean = 10, sd = 5)
  test <- t.test (group1, group2)
  pvalue <- test$p.value
  return(pvalue)
}
# Enter code here that uses replicate to repeat this 10,000 times
```

Questions assigned to the following page: 2.1, 3.1, and 2.2

```
boot <- replicate (10000, f())
# Now use that output to estimate the Type I error.
type1error <- sum(boot <= 0.05)/length(boot)
type1error
```

```
## [1] 0.0504
```

My answer is, to report the type I error of this t-test on this data, is 0.0477

b. (15 points) Repeat the same simulation as above, only now make the data in both of the groups be generated from a gamma distribution with parameters `shape=1` and `rate=3` (like in HW1); again make each group have 20 observations. What do you observe?

```
# Enter code here for simulation of the t-test from the F
# Reuse the code from above as applicable
f1b <- function (){
  group1 <- rgamma(20, shape =1, rate =3)
  group2 <- rgamma(20, shape =1, rate =3)
  test <- t.test (group1, group2)
  p.value <- test$p.value
  return (p.value)
}

boott <- replicate(10000, f1b())

gamma_t1error<- sum(boott <= 0.05)/length(boott)
gamma_t1error
```

```
## [1] 0.0455
```

My answer that we've got the p-value of 0.0465 for this test using the gamma distribution, and from the previous normal distribution, we've got 0.0477. The numbers are similar to each other, but major observation would be that these numbers are typically below 0.05

Question 3. Consider the data from HW1 containing information on predicting heart disease in patients. We have provided another copy of the data with this HW, to avoid having to find the data from last time, but it is the same dataset. Read the data in again,

```
heart<-read.csv("heartDisease.csv",header=TRUE)
head(heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal num
## 1  63   1  1      145  233   1       2     150     0     2.3     3  0    6   0
## 2  67   1  4      160  286   0       2     108     1     1.5     2  3    3   2
## 3  67   1  4      120  229   0       2     129     1     2.6     2  2    7   1
## 4  37   1  3      130  250   0       0     187     0     3.5     3  0    3   0
## 5  41   0  2      130  204   0       2     172     0     1.4     1  0    3   0
## 6  56   1  2      120  236   0       0     178     0     0.8     1  0    3   0
```

a. (5 points) Perform a t-test comparing serum cholestoral in mg/dl (`chol`) between those patients with chest-pain type any type of angina (i.e. `cp` either `typical angina` or `atypical angina`) and those patients with non-anginal pain. What do you conclude?

```
# code for running t-test
angina <- subset (heart, cp == 1| cp == 2)
```

Questions assigned to the following page: 3.1, 3.2, and 3.3

```r
non_angina <- subset (heart, cp == 3)
ttest_3a <- t.test(angina$chol, non_angina$chol)
ttest_3a
```

```
##
##  Welch Two Sample t-test
##
## data:  angina$chol and non_angina$chol
## t = -0.32992, df = 144.25, p-value = 0.7419
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -18.91046  13.50048
## sample estimates:
## mean of x mean of y
##  242.6806  245.3855
```

```r
cat("ttest_3a=", ttest_3a$p.value)
```

```
## ttest_3a= 0.741938
```

My answer is that there are no big of a difference between the serum cholestoral, mean of angianl pain and non anginal. Which also means that there are no significant evidence to reject the null hypothesis.

b. (5 points) Repeat the above, using a permutation test *based on the t-statistic* instead and give your conclusions. You may use the function above or make your own.

```r
# code for running permutation test

f3b <- function () {
  yy <- c(angina$chol, non_angina$chol)
  labels <- sample (1: length (yy), length(angina))
  group3_1 <- yy[labels]
  group3_2 <- yy[-labels]
  tstatistic <- t.test(group3_1, group3_2)$statistic
}

bboot <- replicate (10000, f3b())
p.value <- sum(bboot >= 0.05)/10000

p.value
```

```
## [1] 0.4444
```

My answer is the p-value came out to be larger than 0.05, which indicates we fail to reject the null distribution, which is serum cholestoral in mg/dl has the same distribution in these two group of anginal pain and non-anginal pain and also, we cannot conclude that there are a significant difference.

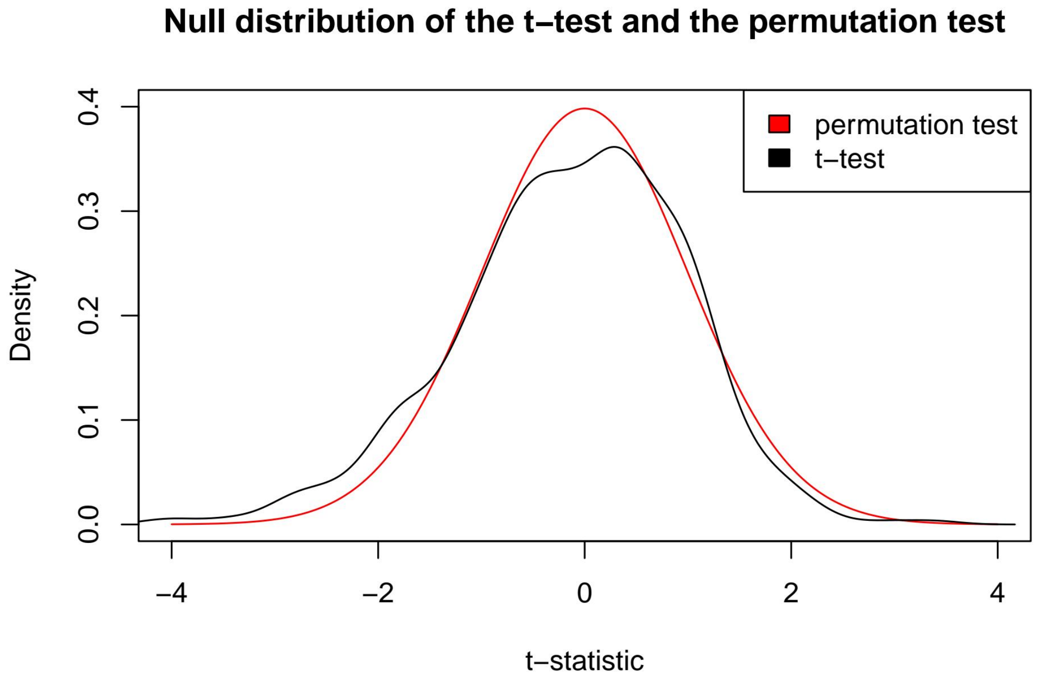
c. (15 points) Compare the null distribution of the t-test and the permutation test for this data by plotting the density curve for both null distributions on the same plot. For the permutation test, you should plot an kernel density estimate of the permutation distribution (not the histogram of the permutation results). Color the t-test black and the permutation test red and provide a legend. How do they compare?

Questions assigned to the following page: 3.3 and 3.4

```
# code for plotting the densities of the two null distributions
# Generating a vector of 1000 values between -4 and 4 (changed after observation)
seq <- seq(-4, 4, length = 1000)

#Using the dataframe we've previously got from 3a 144.25
plot (seq, dt(seq, df = 144.25), ylim = c(0, 0.4), col= "red", type = "l", main = "Null distribution of

bbboot <- replicate(1000, f3b())
lines (density (bbboot))
legend (x = "topright", legend = c("permutation test", "t-test"), fill = c("red", "black"))
```



My answer is that they are basically following the bell shaped normal distribution. However, Compared to the permutation test, t-test has an somewhat unstable curve, and kernel density estimate of the permutation test seems to follow the perfect normal shape.

d. (5 points) 125–200 mg/dL is a normal range of serum cholesteral, though it varies by person what is healthy. Given this, is the difference of means a good statistic, or would you propose a different statistic?

My answer is that with the given 125-200 mg/dL a normal range of serum cholesteral, I think the good statistic would have to be blood pressure that is in the range 125-200 mg/dL, not comparing the difference of the mean. It should give more information, not just finding the differnece between the mean, which makes it not a decent statistic.