STA 104 - Assignment 2

Xiaowei Zeng

October 25, 2023

Exercise 1 (P73)

A certain data set has eight distinct observations, four from each treatment, and all of the observations from treatment 1 are bigger than the observations from treatment 2. What is the one-sided p-value associated with the permutation test?

Solution: Suppose that for treatment 1 we have X_1, X_2, X_3, X_4 and for treatment 2 Y_1, Y_2, Y_3, Y_4, then $X_i > Y_j, \forall i, j = 1, 2, 3, 4$. Since the eight observations are distinct, the difference between the two means of the original data D_{obs} should be the largest among all possible two-sample sets, and thus

$$p\text{-value} = \frac{\#\text{ of } D\text{'s } \geq D_{\text{obs}}}{\binom{8}{4}} = \frac{1}{70} \approx 0.0143.$$

Exercise 2 (P73)

a. Find the permutation distribution of the difference of means for the fictitious data set in the table, and find the p-value for the observed data.

b. Find the permutation distribution of the sum of the observations from treatment 1, and show that the p-value for the observed data is the same as the p-value in part a.

Treatment 1	10	15	50
Treatment 2	12	17	19

Solution: Suppose that for treatment 1 we have X_1, X_2, X_3 and for treatment 2 Y_1, Y_2, Y_3. Suppose that we want to conduct a right-tail test with $H_0 : \theta_1 = \theta_2$ v.s. $H_a : \theta_1 > \theta_2$, where θ is the mean of data.

a. $D_{\text{obs}} = \bar{X} - \bar{Y} = 25 - 16 = 9 > 0$.

```r
pt1 = combn(c(t1, t2), 3) # possible sets for treatment 1
pt2 = pt1[, seq(20, 1, -1)] # possible sets for treatment 2
ptm = colMeans(pt1) - colMeans(pt2) # difference of means for each set
sum(ptm >= 9) # count the number of D's >= D(obs)
```

```
[1] 9
```

So the p-value should be $9/\binom{6}{3} = 0.45$.

b. Use X to compute the statistic. $S_{\text{obs}} = \sum_{i=1}^{3} X_i = 75$.

```r
pts = colSums(pt1) # the sum of each possible set for treatment 1
sum(pts >= 75) # count the number of S's >= S(obs)
```

```
[1] 9
```

So the p-value is $9/\binom{6}{3} = 0.45$, the same as the p-value in part a. This is because

$$\begin{aligned}\text{difference of mean values} &= \frac{T_1}{m} - \frac{T_2}{n} = \frac{T_1}{m} - \frac{T - T_1}{n} \\ &= T_1\left(\frac{1}{m} + \frac{1}{n}\right) - \frac{T}{n},\end{aligned}$$

where $T_1 = \sum_i X_i, T_2 = \sum_j Y_j, T = T_1 + T_2$.

Since m, n, T are all constant, the difference of mean values only depends on T_1 (sum of the observations for treatment 1), indicating that the two tests based on these statistics are equivalent and have a same p-value.

Table 1: Distribution of difference of means and sum of observations

diff of mean	-16.33	-15	-13.67	-13	-11.67	-10.33	-9	-7
sum of obs	37	39	41	42	44	46	48	51
prob	0.05	0.05	0.05	0.05	0.1	0.1	0.05	0.05
diff of mean	7	9	10.33	11.67	13	13.67	15	16.33
sum of obs	72	75	77	79	81	82	84	86
prob	0.05	0.05	0.1	0.1	0.05	0.05	0.05	0.05

Exercise 3 (P73)

Find the permutation distribution of the difference of medians for data in Exercise 2.

Solution: Suppose we want to conduct a left-tail test with $H_0 : \theta_1 = \theta_2$ v.s. $H_a : \theta_1 < \theta_2$, where θ can be the median of data.

Table 2: Distribution of difference of medians

diff of median	-7	-5	-4	-2	2	4	5	7
prob	0.1	0.1	0.1	0.2	0.2	0.1	0.1	0.1

$D_{\text{obs}} = \text{median}(X) - \text{median}(Y) = 15 - 17 = -2$.

```r
# difference of medians for each set
ptmd = sapply(data.frame(pt1), median) - sapply(data.frame(pt2), median)
sum(ptmd <= -2) # count the number of D's <= D(obs)
```

```
[1] 10
```

So the p-value should be $10/\binom{6}{3} = 0.5$.

Exercise 7 (P74)

Students in an introductory statistics class were asked how many brothers and sisters they have and whether their hometown is urban or rural.

Table 3: Number of Siblings in Rural versus Urban Areas

Rural	3 2 1 1 2 1 3 2 2 2 2 5 1 4 1 1 1 1 6 2 2 2 1 1
Urban	1 0 1 1 0 0 1 1 1 8 1 1 1 0 1 1 2

a. Test for a significant difference between rural and urban areas using the Wilcoxon rank-sum test.

b. Test for a significant difference using the two-sample t-test, and compare the results with those obtained in part a. Why are the results different?

Solution: Suppose that for rural we have $X_i, i = 1, \ldots, 24$ and for urban $Y_j, j = 1, \ldots, 17$, so $m = 24, n = 17, N = m + n = 41$. Suppose that we want to conduct a right-tail test with $H_0 : \theta_1 = \theta_2$ v.s. $H_a : \theta_1 > \theta_2$, where θ is the **median** of data for the Wilcoxon rank-sum test and the **mean** of data for the two-sample t-test.

a. The number of all possible permutations of the ranks is $\binom{41}{24} = 151584480450$, which is too large to obtain the exact p-value. Therefore, we may use the resampling strategy to find the approximate p-value.

```r
### Settings
rural = c(3, 2, 1, 1, 2, 1, 3, 2, 2, 2, 2, 5,
          1, 4, 1, 1, 1, 1, 6, 2, 2, 2, 1, 1)
urban = c(1, 0, 1, 1, 0, 0, 1, 1, 1, 8, 1, 1, 1, 0, 1, 1, 2)
combined_sample = c(rural, urban)
m = length(rural); n = length(urban); N = m + n

### Compute the test statistics
rank_total = rank(combined_sample) # the ranks of the combined sample
W_rural = sum(rank_total[1:m]) # the rank-sum of rural data

### Permutation
set.seed(2023) # for reproducibility
b = 50000 # the number of permuted samples
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(rank_total, m))}
sum(permuted_sample >= W_rural) / b # p-value for right-tail test
```

```
[1] 0.00044
```

The approximate p-value is 0.00044.

b. First compute the t-statistic:

$$t = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{S^2}{m} + \frac{S^2}{n}}},$$

where S^2 is the pooled variace of the combined sample, defined as

$$S^2 = \frac{\sum_i^m (X_i - \bar{X})^2 + \sum_j^n (Y_j - \bar{Y})^2}{m + n - 2}.$$

```r
S2 = (sum((rural - mean(rural)) ^ 2) +
        sum((urban - mean(urban)) ^ 2)) / (m + n - 2)
t = (mean(rural) - mean(urban)) / sqrt(S2 / m + S2 / n)
pt(t, df = m + n - 2, lower.tail = F)
```

```
[1] 0.05469799
```

So the p-value is 0.0547, far larger than 0.00044 of the Wilcoxon rank-sum test in part a. This may be because there exists an outlier, 8, in the urban dataset, and t-test has less power on distributions with extreme observations.

Delete the *10th* value in the urban dataset and reconduct the two-sample t-test.

```
S2 = (sum((rural - mean(rural)) ^ 2) +
        sum((urban[-10] - mean(urban[-10])) ^ 2)) / (m + (n - 1) - 2)
t = (mean(rural) - mean(urban[-10])) / sqrt(S2 / m + S2 / (n - 1))
pt(t, df = m + (n - 1) - 2, lower.tail = F)
```

```
[1] 0.0006296573
```

Now the p-value falls to 0.00063, closer to that in part a.

Exercise 8 (P74)

Do a permutation test on the data in Exercise 7. Is the p-value closer to that of the Wilcoxon rank-sum test or to that of the two-sample t-test? What does this suggest about the relationship between the permutation test and the two-sample t-test?

Solution:

```
set.seed(2023) # for reproducibility
b = 50000 # the number of permuted samples
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(combined_sample, m))}
sum(permuted_sample >= sum(rural)) / b
```

```
[1] 0.06186
```

So the p-value for the permutation test is 0.0619, which is closer to two-sample t-test than to Wilcoxon rank-sum test. In Section 2.9.4 of the reference book *Introduction to Modern Nonparametric Statistics*, Keller-McNulty and Higgins (1987) showed that the power of the permutation test will converge to its limiting value, namely the p-value of the t-test, when the number of randomly selected permutations is large.

Exercise 10 (P74)

For the data in Exercise 4, test for differences between sections using Van Der Waerden scores.

Table 4: Data in Exercise 4

Section 1	5	11	16	8	12	
Section 2	17	14	15	21	19	13

Solution: Suppose that for Section 1 we have $X_i, i = 1, \ldots, 5$ and for Section 2 $Y_j, j = 1, \ldots, 6$, so $m = 5, n = 6, N = m + n = 11$. Suppose we want to conduct a left-tail test with $H_0 : \theta_1 = \theta_2$ v.s. $H_a : \theta_1 < \theta_2$, where θ can be the median of data. Suppose that the significance level is 0.05.

```r
### Settings
s1 = c(5, 11, 16, 8, 12)
s2 = c(17, 14, 15, 21, 19, 13)
combined_sample = c(s1, s2)
m = length(s1); n = length(s2); N = m + n
```

The *Van Der Waerden Scores* are defined as

$$V_{(i)} = \Phi^{-1}\left(\frac{i}{N+1}\right).$$

```r
### Compute the test statistics
waerden_score = function(i, N){qnorm(i / (N + 1))} # define the function
rank_total = rank(combined_sample) # the ranks of the combined sample
scores = waerden_score(rank_total, N) # use the ranks to obtain scores
score1 = sum(scores[1:m]) # the score of data in Section 1
score1
```

```
[1] -3.024905
```

The Van Der Waerden score for data in Section 1 is -3.025.

```r
### Permutation
set.seed(2023) # for reproducibility
b = 50000 # the number of permuted samples
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(scores, m))}
sum(permuted_sample <= score1) / b
```

```
[1] 0.0109
```

So the p-value is $0.0109 < 0.05$, indicating that we should reject the null hypothesis and there exists a significant difference in the two sections.

Exercise 12 (P74)

Refer to the data in Exercise 4. Make a 90% confidence interval for Δ. Obtain the Hodges-Lehmann estimate of Δ.

Solution: First compute the pairwise difference between X's and Y's.

```r
pwd_table = matrix(rep(s1, n), nrow = n, byrow = T) -
        matrix(rep(s2, m), ncol = m)
```

Table 5: Pairwise Differences of X and Y $(X_i - Y_j)$

	X_1	X_2	X_3	X_4	X_5
Y_1	-12	-6	-1	-9	-5
Y_2	-9	-3	2	-6	-2
Y_3	-10	-4	1	-7	-3
Y_4	-16	-10	-5	-13	-9
Y_5	-14	-8	-3	-11	-7
Y_6	-8	-2	3	-5	-1

Recall that the U statistic is defined as

$$U = \text{number of pairs } (X_i, Y_j) \text{ for which } X_i < Y_j.$$

We want to find two numbers k_a and k_b such that the U statistic satisfies

$$P(k_a \leq U \leq k_b - 1) \leq 0.90.$$

Having chosen k_a and k_b, we form the confidence interval

$$\text{pwd}(k_a) < \Delta \leq \text{pwd}(k_b),$$

and find the lower-tail and upper-tail 5% values $l_{.05}$ and $u_{.05}$ from Table A4 in the Appendix. We choose

$$k_a = l_{.05} + 1 = 5 + 1 = 6,$$
$$k_b = u_{.05} = 25,$$

and have $\text{pwd}(k_a) = -10$, $\text{pwd}(k_b) = -2$.

```r
pwd_sorted = sort(pwd_table)
pwd_sorted[6]; pwd_sorted[25]
```

```
[1] -10
[1] -2
```

So the 90% confidence interval for Δ is $(-10, -2]$.

```r
median(pwd_table)
```

```
[1] -6
```

The Hodges-Lehmann estimate of Δ is the median of all the pairwise differences of X's and Y's $(X_i - Y_j)$, namely -6.

Furthermore, if we want to get the pairwise differences $(Y_j - X_i)$, we can simply reverse the sign of the results above. The 90% confidence interval will be $(2, 10]$, and the Hodges-Lehmann estimate will be 6.

Exercise 18 (P75)

A biologist examined the effect of a fungal infection on the eating behavior of rodents. Infected apples were offered to a group of eight rodents, and sterile apples were offered to a group of four. The amounts consumed (grams of apple/kilogram of body weight) are listed in the table.

Experimental Group	11, 33, 48, 34, 112, 369, 64, 44
Control Group	177, 80, 141, 332

Apply several nonparametric tests to the data, including the Wilcoxon rank-sum test, the Van Der Waerden scores test, the exponential scores test, and the permutation test on the original data. Discuss differences in conclusions using these tests, and justify the use of one or more of these tests on these data.

Solution: Suppose that for the experimental group we have $X_i, i = 1, \dots, 8$ and for the control group $Y_j, j = 1, \dots, 4$, so $m = 8, n = 4, N = m + n = 12$. Suppose we want to conduct a left-tail test with $H_0 : \theta_1 = \theta_2$ v.s. $H_a : \theta_1 < \theta_2$, where θ can be the mean or median of data. Suppose that the significance level is 0.05.

```
### Settings
g1 = c(11, 33, 48, 34, 112, 369, 64, 44)
g2 = c(177, 80, 141, 332)
combined_sample = c(g1, g2)
m = length(g1); n = length(g2); N = m + n
```

Wilcoxon rank-sum test:

```
wilcox.test(g1, g2, alternative = 'less')
```

```
        Wilcoxon rank sum exact test
data:  g1 and g2
W = 5, p-value = 0.03636
alternative hypothesis: true location shift is less than 0
```

The p-value is $0.0364 < 0.05$, indicating that we should reject the null hypothesis.

Van Der Waerden scores test:

```
rank_total = rank(combined_sample)
scores = waerden_score(rank_total, N)
score1 = sum(scores[1:m])
pt1 = combn(scores, m) # possible sets for experimental group
```

```r
pts = colSums(pt1) # possible van der waerden score-sum for experimental group
sum(pts <= score1) / choose(N, m) # lower-tail p-value
```

```
[1] 0.04848485
```

The p-value is $0.0485 < 0.05$, indicating that we should reject the null hypothesis.

Exponential scores test:

The *Exponential Scores* are based on the expected values of the order statistics, like

$$\frac{1}{N}, \frac{1}{N} + \frac{1}{N-1}, \frac{1}{N} + \frac{1}{N-1} + \frac{1}{N-2}, \ldots$$

```r
exponential = function(i, N){sum(1 / seq(N, N + 1 - i, - 1))}
scores = sapply(rank_total, exponential, N = N)
score1 = sum(scores[1:m])
pt1 = combn(scores, m) # possible sets for experimental group
pts = colSums(pt1) # possible exponential score-sum for experimental group
sum(pts <= score1) / choose(N, m) # lower-tail p-value
```

```
[1] 0.1333333
```

The p-value is $0.1333 > 0.05$, indicating that we **cannot** reject the null hypothesis.

Permutation test:

```r
pt1 = combn(combined_sample, m) # possible sets for experimental group
pts = colSums(pt1) # possible sum for experimental group
sum(pts <= sum(g1)) / choose(N, m) # lower-tail p-value
```

```
[1] 0.1070707
```

The p-value is $0.1070 > 0.05$, indicating that we **cannot** reject the null hypothesis.

Insights: The differences of p-values of the four tests can be mainly attributed to the outlier in the experimental group, 369. The p-values of the Wilcoxon rank-sum test and the Van Der Waerden scores test are both lower than 0.05, indicating that they are affected by the outlier only to a small extent, while the p-values of the Exponential scores test and Permutation are both higher than 0.05, showing that they are influenced significantly. This is because the rank score and van der waerden score are very robust, which means even if the outlier is really large in exact value, these score functions can transform it into a relative smaller scale. On the contrary, the exponential score and mean/sum score (permutation test) are very sensitive to outliers. We can prove it by removing the outlier from the experimental group and conduct the four tests again.

Removing an outlier:

```
g1 = g1[-6] # remove the value of 369
combined_sample = c(g1, g2)
m = length(g1); n = length(g2); N = m + n
```

The codes for the aforementioned tests are the same as before, so I will only put the results here. Now the p-values for all the tests are 0.00606.

Appendix

The codes for all the exercises are put here.

```r
# Exercise 1
1 / choose(8, 4)

# Exercise 2
t1 = c(10, 15, 50)
t2 = c(12, 17, 19)
# 2a
mean(t1); mean(t2); mean(t1) - mean(t2)
pt1 = combn(c(t1, t2), 3)
pt2 = pt1[, seq(choose(6, 3), 1, -1)]
ptm = colMeans(pt1) - colMeans(pt2)
sum(ptm >= 9)
sum(ptm >= 9) / choose(6, 3)
# 2b
pts = colSums(pt1)
sum(pts >= 75) / choose(6, 3)

# Exercise 3
median(t1); median(t2); median(t1) - median(t2)
ptmd = sapply(data.frame(pt1), median) - sapply(data.frame(pt2), median)
sum(ptmd <= -2) / choose(6, 3)

# Exercise 7
rural = c(3, 2, 1, 1, 2, 1, 3, 2, 2, 2, 2, 5,
          1, 4, 1, 1, 1, 1, 6, 2, 2, 2, 1, 1)
urban = c(1, 0, 1, 1, 0, 0, 1, 1, 1, 8, 1, 1, 1, 0, 1, 1, 2)
combined_sample = c(rural, urban)
m = length(rural)
n = length(urban)
N = m + n
# 7a
rank_total = rank(combined_sample)
W_rural = sum(rank_total[1:m])
set.seed(2023)
b = 50000
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(rank_total, m))}
sum(permuted_sample >= W_rural) / b
```

```
# 7b
S2 = (sum((rural - mean(rural)) ^ 2) +
        sum((urban - mean(urban)) ^ 2)) / (m + n - 2)
t = (mean(rural) - mean(urban)) / sqrt(S2 / m + S2 / n)
pt(t, df = m + n - 2, lower.tail = F)

S2 = (sum((rural - mean(rural)) ^ 2) +
        sum((urban[-10] - mean(urban[-10])) ^ 2)) / (m + (n - 1) - 2)
t = (mean(rural) - mean(urban[-10])) / sqrt(S2 / m + S2 / (n - 1))
pt(t, df = m + (n - 1) - 2, lower.tail = F)
# t.test(rural, urban, alternative = 'greater', var.equal = TRUE)
# t.test(rural, urban[-10], alternative = 'greater', var.equal = TRUE)

# Exercise 8
S_rural = sum(rural)
set.seed(2023)
b = 50000
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(combined_sample, m))}
sum(permuted_sample >= S_rural) / b

# Exercise 10
s1 = c(5, 11, 16, 8, 12)
s2 = c(17, 14, 15, 21, 19, 13)
combined_sample = c(s1, s2)
m = length(s1); n = length(s2); N = m + n
waerden_score = function(i, N){qnorm(i / (N + 1))}
rank_total = rank(combined_sample)
scores = waerden_score(rank_total, N)
score1 = sum(scores[1:m])
set.seed(2023)
b = 50000
permuted_sample = rep(0, b)
for (i in 1:b){permuted_sample[i] = sum(sample(scores, m))}
sum(permuted_sample <= score1) / b

# Exercise 12
pwd_table = matrix(rep(s1, n), nrow = n, byrow = T) -
  matrix(rep(s2, m), ncol = m)
pwd_sorted = sort(pwd_table)
pwd_sorted[6]; pwd_sorted[25]; median(pwd_table)
```

```
# Exercise 18
g1 = c(11, 33, 48, 34, 112, 369, 64, 44)
g2 = c(177, 80, 141, 332)
combined_sample = c(g1, g2)
m = length(g1); n = length(g2); N = m + n
# Wilcoxon rank-sum test
wilcox.test(g1, g2, alternative = 'less')
# Van Der Waerden scores test
rank_total = rank(combined_sample)
scores = waerden_score(rank_total, N)
score1 = sum(scores[1:m])
pt1 = combn(scores, m)
pts = colSums(pt1)
sum(pts <= score1) / choose(N, m)
# Exponential scores test
exponential = function(i, N){sum(1 / seq(N, N + 1 - i, - 1))}
scores = sapply(rank_total, exponential, N = N)
score1 = sum(scores[1:m])
pt1 = combn(scores, m)
pts = colSums(pt1)
sum(pts <= score1) / choose(N, m)
# Permutation test
pt1 = combn(combined_sample, m)
pts = colSums(pt1)
sum(pts <= sum(g1)) / choose(N, m)
# Removing an outlier
g1 = g1[-6]
combined_sample = c(g1, g2)
m = length(g1); n = length(g2); N = m + n
# Wilcoxon rank-sum test
wilcox.test(g1, g2, alternative = 'less')
# Van Der Waerden scores test
rank_total = rank(combined_sample)
scores = waerden_score(rank_total, N)
score1 = sum(scores[1:m])
pt1 = combn(scores, m)
pts = colSums(pt1)
sum(pts <= score1) / choose(N, m)
# Exponential scores test
exponential = function(i, N){sum(1 / seq(N, N + 1 - i, - 1))}
scores = sapply(rank_total, exponential, N = N)
```

```
score1 = sum(scores[1:m])
pt1 = combn(scores, m)
pts = colSums(pt1)
sum(pts <= score1) / choose(N, m)
# Permutation test
pt1 = combn(combined_sample, m)
pts = colSums(pt1)
sum(pts <= sum(g1)) / choose(N, m)
```