Two-sample Tests for Locations

Cheng Peng

STA200 Statistics II

Contents

1 Introduction

Two-sample tests are statistical methods used to compare two groups (samples) to determine whether they come from the same population or have different characteristics. As you have learned in one-sample tests, two-sample tests can be broadly classified into:

Parametric Tests: Assume data follows a specific distribution (usually normal) and compare parameters like means or variances.

Nonparametric Tests: Make fewer assumptions about the data distribution and are used when parametric assumptions are violated.

This module reviews parametric two-sample tests first and then discusses non-parametric two-sample tests.

2 (Raw) Data Structure

For convenience, we include the section on data structures from the previous module. All procedures introduced in this module will use the raw data table with the following layout:

Y	X_1	X_2	$\cdots$	X_k
y_1	x_{11}	x_{21}	$\cdots$	x_{k1}
y_2	x_{12}	x_{22}	$\cdots$	x_{k2}
y_3	x_{13}	x_{23}	$\cdots$	x_{k3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	x_{1n}	x_{2n}	$\cdots$	x_{kn}

For example, an employer wants to see whether their pay rate is fair in terms of gender (i.e., assessing potential gaps in pay rate between male and female employees in the same role under similar conditions). The HR helped to create a dataset in the following form.

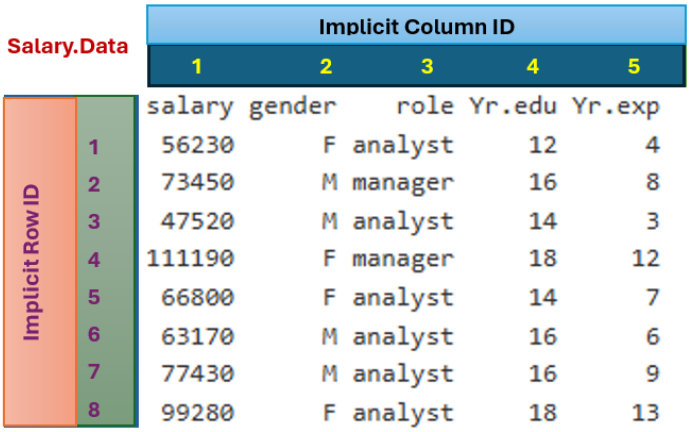
salary (Y)	gender (X_1)	role (X_2)	Yr_edu (X_3)	Yr_exp (X_4)
\$56,230	F	analyst	12	4
\$73,450	M	manager	16	8
\$47,520	M	analyst	14	3
\$111,190	F	manager	18	12
\$66,800	F	analyst	14	7
\$63,170	M	analyst	16	6
\$77,430	M	analyst	16	9
\$99,280	F	analyst	18	13

We also introduced R data frames earlier to store a dataset in R and then use it for analysis. The next R code defines a dataframe in R based on the above toy dataset.

```
my.toy.data <- data.frame(salary = c(56230, 73450, 47520, 111190, 66800, 63170, 77430, 99280),   # nume
                          gender = c("F", "M", "M", "F", "F", "M", "M", "F"),            # character (c
                          role = c("analyst", "manager", "analyst", "manager", "analyst", "analyst", "a
                          Yr.edu = c(12, 16, 14, 18, 14, 16, 16, 18),
                          Yr.exp = c(4,8,3,12,7,6,9,13)
                          )
my.toy.data     # print the data
```

```
|   salary gender    role Yr.edu Yr.exp
| 1  56230      F analyst     12      4
| 2  73450      M manager     16      8
| 3  47520      M analyst     14      3
| 4 111190      F manager     18     12
| 5  66800      F analyst     14      7
| 6  63170      M analyst     16      6
| 7  77430      M analyst     16      9
| 8  99280      F analyst     18     13
```

Since we will work with more than one variable starting from this module, we introduce a few commands you can use to select rows, columns, or individual cell values in a data frame. The following figure shows the basics on how to access an R data frame.



- By default, in R, every row and column of a data frame is implicitly indexed starting from 1, i.e., rows and columns are numbered 1, 2, 3, ...
- Since column labels of a data frame represent variable names which are explicit indexes of the corresponding columns. Therefore, each column is doubly indexed.
- Individual cells in a data frame are jointly indexed by their corresponding row and column indexes.

The following are a few examples

1. Explicit Access

```
# Select a single cell located at the intersection of row 2 and column 3
my.toy.data[2,3]
```

```
| [1] "manager"
```

```
# multiple cells involve multiple rows and columns. In this case, indices must be
# provided in the form of vector, e.g., c(2,5,6)
my.toy.data[c(3,6), c(1,3)]
```

```
|   salary    role
| 3  47520 analyst
| 6  63170 analyst
```

```
# select one column and ALL rows
my.toy.data[ , 4]
```

```
| [1] 12 16 14 18 14 16 16 18
```

```
# select multiple columns and ALL rows
my.toy.data[ , c(1,5)]
```

```
|   salary Yr.exp
| 1  56230      4
| 2  73450      8
| 3  47520      3
```

```
| 4 111190     12
| 5  66800      7
| 6  63170      6
| 7  77430      9
| 8  99280     13
```

```
# select multiple columns and ALL rows using variable names
my.toy.data[ , c("salary", "gender", "role")]   # vector with character values in quotes!
```

```
|   salary gender    role
| 1  56230      F analyst
| 2  73450      M manager
| 3  47520      M analyst
| 4 111190      F manager
| 5  66800      F analyst
| 6  63170      M analyst
| 7  77430      M analyst
| 8  99280      F analyst
```

```
# select one row (also called one record) and ALL columns
my.toy.data[4 , ]
```

```
|   salary gender    role Yr.edu Yr.exp
| 4 111190      F manager     18     12
```

```
# select multiple rows and ALL columns
my.toy.data[c(1,5) , ]
```

```
|   salary gender    role Yr.edu Yr.exp
| 1  56230      F analyst     12      4
| 5  66800      F analyst     14      7
```

2. Conditional Access

In application, sometimes we need to select a subset of the data under certain conditions defined based on variables. For example, if we want to compare the mean salary between male and female employees in a company, we need to calculate the sample size, mean, and standard deviation, respectively, from the male and female groups in introductory statistics classes.

There are different R commands to achieve this goal. We use `which()` to identify salaries for male and female employees, respectively. The following code shows how to separate the salaries of the two groups.

```
# use my.toy.data$gender to identify male group
male.id <- which(my.toy.data$gender == "M")     # CAUTION: double equal sign (==) MUST be used in condit
                                                # This will return the row indexes of male employees
# Use the above returned row index to extract the two groups
male.salary <- my.toy.data[male.id, ]           # male group
female.salary <- my.toy.data[ - male.id, ]      # "- male.id": not male => female indexes
```

Print out the above subset.

```
##
male.salary
```

```
|   salary gender    role Yr.edu Yr.exp
| 2  73450      M manager     16      8
| 3  47520      M analyst     14      3
| 6  63170      M analyst     16      6
| 7  77430      M analyst     16      9
```

```
##
female.salary
```

```
|   salary gender    role Yr.edu Yr.exp
| 1  56230      F analyst     12      4
| 4 111190      F manager     18     12
| 5  66800      F analyst     14      7
| 8  99280      F analyst     18     13
```

3 Two-Sample t-Test Revisited

Recall that, in introductory statistics (MAT121/125), the assumptions of the two t-test are

- Both populations are normally distributed
- Both population variances are unknown but equal

Under the above assumptions, the two random samples were taken from the two independent populations, respectively, with the following statistics.

Type of Statistics	**sample 1**	**sample 2**
sample size	n_1	n_2
sample mean	$\bar{x}_1$	$\bar{x}_2$
sample standard deviation	s_1	s_2

Under the second assumption, we need to estimate the **common standard deviation** by pooling two samples using the following formula.

$$s_{\text{pool}} = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$$

The test statistic for testing $H_0 : \mu_1 - \mu_2 = 0$ vs $H_a : \mu_1 - \mu_2 \neq 0$ is

$$TS = \frac{(\bar{x}_1 - \bar{x}_2) - 0}{s_{\text{pool}}\sqrt{1/n_1 + 1/n_2}} \to t_{n_1+n_2-2}$$

Remark: If the **equal variances** assumption is not met, one can use an approximate t-test using the Welch-Satterthwaite procedure. This approximation will not be discussed in this note. We will address this in the subsequent notes under more general settings.

Example: We will use the Pima Indian Diabetes data to illustrate the above two sample tests to see whether the mean BMI levels differ between diabetes and diabetes-free populations. We need to load the data first before calculating the related statistics required for the test. Note that diabetes status is reflected in the variable `diabetes`.

$$H_0 : \mu_1 - \mu_2 = 0 \quad \text{vs} \quad H_a : \mu_1 - \mu_2 \neq 0$$

Since R is sensitive, we may want to use the R command `head()` *to check the exact names and first 6 observations in the data frame*

```
# read data into R
PimaIndiaDiabetes <- read.csv("https://pengdsci.github.io/STA200/dataset/PimaIndiaDiabetes.csv")
## checking variable names
head(PimaIndiaDiabetes)
```

```
|    X pregnant glucose pressure triceps insulin mass pedigree age diabetes
| 1  4        1      89       66      23      94 28.1    0.167  21      neg
| 2  5        0     137       40      35     168 43.1    2.288  33      pos
| 3  7        3      78       50      32      88 31.0    0.248  26      pos
| 4  9        2     197       70      45     543 30.5    0.158  53      pos
| 5 14        1     189       60      23     846 30.1    0.398  59      pos
| 6 15        5     166       72      19     175 25.8    0.587  51      pos
```

3.1 Manual Calculation Using R

Next, we show step-by-step manual calculation and translate the above formulas directly into R code and report the test statistic, degrees of freedom, and the p-value for statistical decision. To this end, we define a subset with only two variables: BMI (`mass`) and `diabetes`.

```
sub.diabetes <- PimaIndiaDiabetes[ , c("mass", "diabetes")]   # values in the vector must be COMMA sepa
diabetes.id <- which(sub.diabetes$diabetes == "pos")
diabetes.pop <- sub.diabetes[diabetes.id, ]
no.diabetes.pop <- sub.diabetes[-diabetes.id, ]
##
## statistics for pop #1
n1 <- length(diabetes.pop$mass)
xbar.1 <- mean(diabetes.pop$mass)
s.1 <- sd(diabetes.pop$mass)
## statistics for pop #2
n2 <- length(no.diabetes.pop$mass)
xbar.2 <- mean(no.diabetes.pop$mass)
s.2 <- sd(no.diabetes.pop$mass)
## pooled standard deviation
s.pool <- sqrt(((n1-1)*s.1^2 + (n2-1)*s.2^2)/(n1+n2-2))
## evaluate test statistic
TS <- ((xbar.1-xbar.2)-0)/(s.pool*sqrt(1/n1 + 1/n2))
## absolute value of TS
abs.TS <- abs(TS)
##p-value: 2 times the right-tail area for a two-sample test
p.value <- 2*pt(abs.TS, df = n1+n2-2, lower.tail = FALSE)  # lower.tail = FALSE specifies the right tai
## print out statistic, df, and p-value in a combined vector
cbind(TS = TS, df = n1 +n2 - 2, p.value = p.value)
```

```
|            TS  df      p.value
| [1,] 5.540362 390 5.563221e-08
```

There is clear **strong** evidence that the mean body mass indices (BMI) in diabetes and diabetes-free populations are **different** ($p = 5.563221e\text{-}08 < 0.05$). **This is consistent with clinical findings.**

3.2 Calling R Built-in Function

We have used `t.test()` in a previous note for a one-sample t test. The same function can be used to perform a 2-sample t test. We need to provide the independent sample vectors directly to the function to produce the results.

```
# We use the two samples of BMI found previously:
t.test(diabetes.pop$mass, no.diabetes.pop$mass, alternative = "two.sided", var.equal = TRUE)
```

```
|
|   Two Sample t-test
|
| data:  diabetes.pop$mass and no.diabetes.pop$mass
| t = 5.5404, df = 390, p-value = 5.563e-08
| alternative hypothesis: true difference in means is not equal to 0
| 95 percent confidence interval:
|  2.597924 5.455934
| sample estimates:
| mean of x mean of y
|  35.77769  31.75076
```

The above output indicates that the built-in function produces the same results.

As an exercise, you can explore whether the mean of `glucose` levels in diabetes and diabetes-free populations are equal.

4 Regression Approach to Two-sample t Test

The regression approach to the two-sample t-test is straightforward. Special attention should be paid to the model formula in that the **group** variable must be on the right-hand side and MUST be a factor. R function `factor()` turns a variable into a factor variable. In other words, the model formula should be in the form `lm(y ~ factor(x), data = dataset.name)`

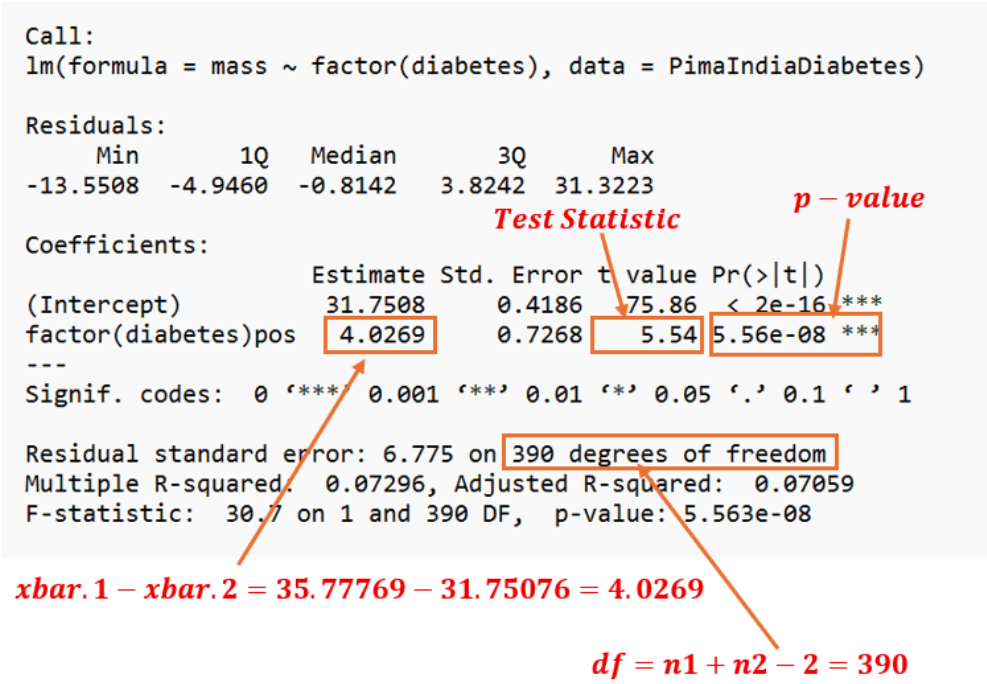
The following three lines of code produce the results in the above two-sample test using the regression method.

```
# read data into R
PimaIndiaDiabetes <- read.csv("https://pengdsci.github.io/STA200/dataset/PimaIndiaDiabetes.csv")
reg.2.sample.test <- lm(mass ~ factor(diabetes), data = PimaIndiaDiabetes)
summary(reg.2.sample.test)
```

```
|
| Call:
| lm(formula = mass ~ factor(diabetes), data = PimaIndiaDiabetes)
|
| Residuals:
|      Min       1Q   Median       3Q      Max
```

```
| -13.5508  -4.9460  -0.8142   3.8242  31.3223
|
| Coefficients:
|                     Estimate Std. Error t value Pr(>|t|)
| (Intercept)          31.7508     0.4186   75.86  < 2e-16 ***
| factor(diabetes)pos   4.0269     0.7268    5.54 5.56e-08 ***
| ---
| Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
| Residual standard error: 6.775 on 390 degrees of freedom
| Multiple R-squared:  0.07296, Adjusted R-squared:  0.07059
| F-statistic:  30.7 on 1 and 390 DF,  p-value: 5.563e-08
```

The following annotated output shows that the results are identical to those obtained in the previous section.

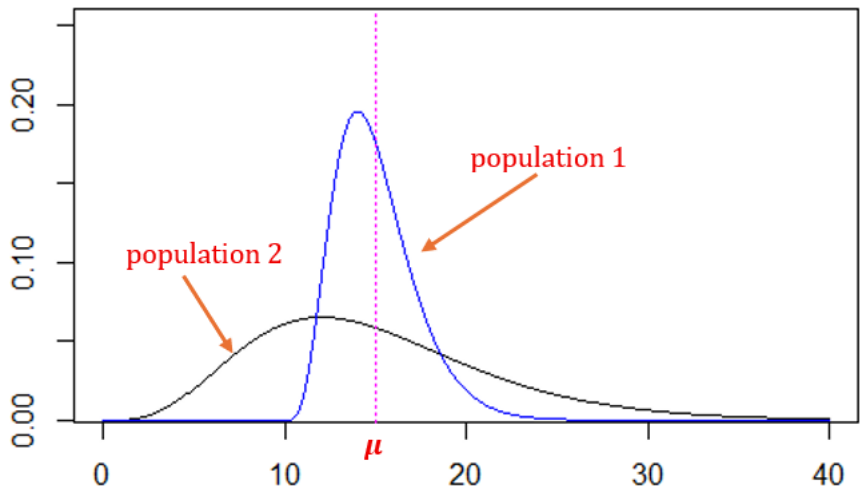


5 Non-parametric Two-sample Test

We have discussed the comparison of two normal population means under the assumption that the two populations have equal variances, using the two-sample t-test. However, if either of the two assumptions—normality and equal variances—is not satisfied, the two-sample t-test may not be appropriate.

In statistics, there are alternative procedures for comparing numeric characteristics between populations without assuming normality or equal variances. This section introduces one such test: the **Wilcoxon Rank Sum Test** (also called the **Mann-Whitney U test**), which assesses **whether the two population distributions are equal**.

This is a much stronger test because the assumption of equal means does not require the two populations to have identical distributions (see the following figure).



The **Wilcoxon Rank Sum Test** assesses whether two sampled groups are likely to derive from the same population. A common misconception is that it compares medians, but this is **only true if the distributions are symmetric and differ only in location**. In other words, if the distributions have different shapes (e.g., skewness or variance differences), the test may still reject (H_0) even if medians are equal (see the above figure).

The hypotheses in a **two-tailed Wilcoxon Rank Sum Test** are:

- The **null hypothesis** (H_0) is that the **two population distributions are equal**.
- The **alternative hypothesis** (H_1) is that the **two population distributions are not equal**.

Remark: *one-tailed Wilcoxon Rank Sum Tests compare stochastic dominance between the two distributions. This is out of the scope of this class. We will not discuss this topic in this class.*

5.1 Manual Implementation

Next, we discuss the development of the **Wilcoxon Rank Sum** test. To help you understand the steps, we use the following toy data set.

Group A	Group B
10	15
12	18
14	20
16	22

Step 1: Ranking the Data

- Combine the data from both groups into a single dataset and sort them in **ascending order**: `10, 12, 14, 15, 16, 18, 20, 22`
- Assign ranks to all observations from smallest to largest (**tied values receive the average rank**): `1, 2, 3, 4, 5, 6, 7, 8`.

Step 2: Ranking the Data

- Sum the ranks for each group **separately**.
  - \$R_1 = \$ Sum of ranks for Group 1: $R_A = 1 + 2 + 3 + 5 = 11$
  - \$R_2 = \$ Sum of ranks for Group 2: $R_B = 4 + 6 + 7 + 8 = 25$

Step 3 Compute the Test Statistic (U)

- The test statistic U can be calculated for either group (n_1 and n_2 are group sizes):

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$
$$U_2 = n_1 n_2 + \frac{n_2(n_2+1)}{2} - R_2$$

The test statistic (commonly denoted by U) is defined to be

$$U = \min\{U_1, U_2\},$$

That is, U is equal to the smaller of U_1 and U_2.

In the above toy example,

$$U_A = 4 \times 4 + \frac{4(4+1)}{2} - 11 = 15$$
$$U_A = 4 \times 4 + \frac{4(4+1)}{2} - 25 = 1$$
$$U = \min\{15, 1\} = 1.$$

Step 4 Statistical Decision

To make a statistical decision, we need to find the critical value or the p-value. The key question is: What is the distribution of the test statistic? Similar to the sign test, there is an exact distribution and an approximate distribution.

Exact Distribution

The exact probability distribution is symmetric and can be computed using recursive methods or lookup tables for small n_1, n_2 (typically $n_1, n_2 \leq 20$). The following table given critical values for ($n_1 \leq 20$ and $n_2 \leq 20$) **based on the significant level of 0.05**.

n1\n2	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2							0	0	0	0	1	1	1	1	1	2	2	2	2
3				0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
4			0	1	2	3	4	4	5	6	7	8	9	10	11	11	12	13	14
5		0	1	2	3	5	6	7	8	9	11	12	13	14	15	17	18	19	20
6		1	2	3	5	6	7	10	11	13	14	16	17	19	21	22	24	25	27
7		1	3	5	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
8	0	2	4	6	7	10	13	15	17	19	22	24	26	29	31	34	36	38	41
9	0	2	4	7	10	12	15	17	20	23	26	28	31	34	37	39	42	45	48
10	0	3	5	8	11	14	17	20	23	26	29	33	36	39	42	45	48	52	55
11	0	3	6	9	13	16	19	23	26	30	33	37	40	44	47	51	55	58	62
12	1	4	7	11	14	18	22	26	29	33	37	41	45	49	53	57	61	65	69
13	1	4	8	12	16	20	24	28	33	37	41	45	50	54	59	63	67	72	76
14	1	5	9	13	17	22	26	31	36	40	45	50	55	59	64	67	74	78	83
15	1	5	10	14	19	24	29	34	39	44	49	54	59	64	70	75	80	85	90
16	1	6	11	15	21	26	31	37	42	47	53	59	64	70	75	81	86	92	98
17	2	6	11	17	22	28	34	39	45	51	57	63	67	75	81	87	93	99	105
18	2	7	12	18	24	30	36	42	48	55	61	67	74	80	86	93	99	106	112
19	2	7	13	19	25	32	38	45	52	58	65	72	78	85	92	99	106	113	119
20	2	8	13	20	27	34	41	48	55	62	69	76	83	90	98	105	112	119	127

The **Decision Rule** based on the critical value from the above table:

- Reject H_0 if $U < CV_\alpha$ or $U > n_1 n_2 - CV_\alpha$
- Fail to reject H_0 if $CV_\alpha < U < n_1 n_2 - CV_\alpha$

From the above table, for $n_1 = n_2 = 4$ and $\alpha = 0.05$ (one-tailed), the critical value from the Wilcoxon Rank Sum test is 0. Recall that $U = 1$. We see that $0 = CV_{0.05} < U < 16$, we fail to reject the null hypothesis H_0. That is, the two distributions are **not significantly different**.

Normal Approximation

If $n_1 > 20$ and $n_2 > 20$,

$$Z = \frac{U - \mu_U}{\sigma_U} \to N(0, 1),$$

where

$$\mu_U = \frac{n_1 n_2}{2} \quad \text{and} \quad \sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}.$$

We can either normal table or software to find the critical value and p-value.

The following short video (https://pengdsci.github.io/STA200/week03/Nonparametrics-WilcoxonTest.mp4) explains the Wilcoxon Rank Sum test with manually worked examples. **Table 9 in Appendix B** in the video refers to the U-table with significance level of $\alpha = 0.05$.

5.2 Implementation Using R

The R function `wilcox.test()` in the base package **{stats}** implements the **rank sum test**. The key arguments are

- Define two vectors to store the group values.
- `alternative = "two.sided"`(default, tests for any difference).
- `exact = TRUE` (for small samples, computes exact p-value). If `FALSE`, uses normal approximation (recommended for large samples).
- `paired = FALSE` (default, ensures it's an independent-samples test).

```
# Define vectors to store values from Group A and Group B, respectively
group.A <- c(10, 12, 14, 16)
group.B <- c(15, 18, 20, 22)
# Perform Wilcoxon Rank Sum Test (Mann-Whitney U)
result <- wilcox.test(group.A, group.B, alternative = "two.sided", exact = TRUE)
result
```

```
|
|   Wilcoxon rank sum exact test
|
| data:  group.A and group.B
| W = 1, p-value = 0.05714
| alternative hypothesis: true location shift is not equal to 0
```

The above results show that the test statistic W = 1 and p-value = 0.0574. At the significance level of 0.05, the null hypothesis was not rejected. This is consistent with the result obtained previously based on the critical value method.

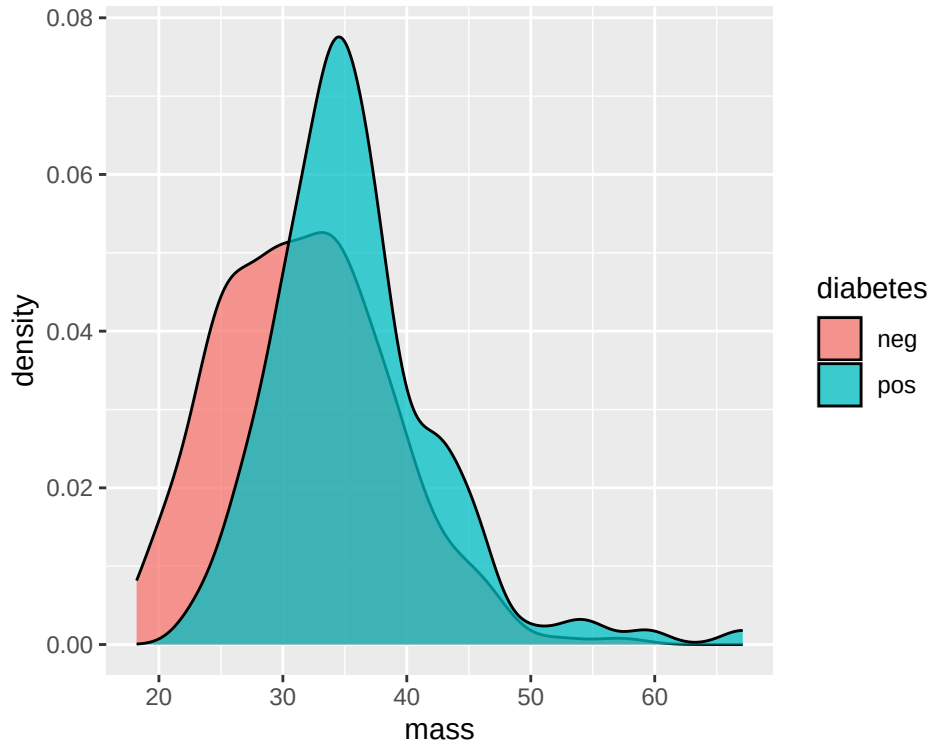
5.3 Case study - Pima Indian Diabetes

To conclude this section, we implement the Wilcoxon Rank Sum test to assess whether the distribution of BMIs is the same between diabetes and diabetes-free populations. We will reload the data and define vectors to store BMI for the two populations. Since both sample sizes are larger than 20, we will use the approximation approach for the p-value.

```
# read data into R
PimaIndiaDiabetes <- read.csv("https://pengdsci.github.io/STA200/dataset/PimaIndiaDiabetes.csv")
# define BMI for diabetes and diabetes-free population
sub.diabetes <- PimaIndiaDiabetes[ , c("mass", "diabetes")]   # values in the vector must be COMMA sepa
diabetes.id <- which(sub.diabetes$diabetes == "pos")
diabetes.pop <- sub.diabetes[diabetes.id, ]
no.diabetes.pop <- sub.diabetes[-diabetes.id, ]
##
diabetes.BMI <- diabetes.pop$mass
no.diabetes.BMI <- no.diabetes.pop$mass
## Wilcoxon test
wilcox.test(diabetes.BMI, no.diabetes.BMI, alternative = "two.sided", exact = FALSE)
```

```
|
|   Wilcoxon rank sum test with continuity correction
|
| data:  diabetes.BMI and no.diabetes.BMI
| W = 22608, p-value = 1.287e-07
| alternative hypothesis: true location shift is not equal to 0
```

The above test result shows that the null hypothesis is rejected ($p \approx 0$). This means the distributions of BMIs in the two groups are different.



The above overlaid density curves also showed the difference between the two distributions.

6 Two-Paired-Sample Tests

The two-sample tests discussed in the previous section are based on two **independent samples** drawn from two independent populations. For example, we may take two sets of new CS graduates (`subjects`) from two different universities and compare their starting salaries (`measurements`). In practical applications, however, there are situations in which only one set of `subjects` is drawn from a single population, but `measurements` are taken twice from each subject at two different time points. The two sets of sample `measurements` in this case are called **paired samples**. For example, a clinical trial investigates whether a new blood pressure (BP) medication reduces systolic BP in hypertensive patients.

- **Measurements**: Systolic BP (mmHg) recorded before and after 4 weeks of treatment.
- **Sample Size**: 10 patients (small sample *implies* normality check crucial).

Patient	Before	After
1	150	140
2	160	155
3	145	142
4	155	150
5	140	138
6	165	160
7	152	148
8	158	152
9	148	145
10	162	156

Because each pair of measurements is taken from the same patient and therefore correlated (i.e., dependent), we cannot use the two-sample tests introduced in previous sections to assess the treatment effect. It is reasonable to take the difference between each pair of measurements so that each subject has a single resulting value. This transforms the two-correlated-sample problem into a one-sample situation.

6.1 Paired Sample t Test

We have covered this topic in MAT121/125. The basic assumptions are

- The differences are normally distributed
- The standard deviation of the population of **differences** is unknown.

We will not repeat the details of the one-sample t test. Next, we use the R function `t.test()` to assess the treatment effect based on the above data table.

```
before <- c(150, 160, 145, 155, 140, 165, 152, 158, 148, 162)
after <- c(140, 155, 142, 150, 138, 160, 148, 152, 145, 156)
differences <- after - before
##
t.test(differences)
```

```
|
|   One Sample t-test
|
| data:  differences
| t = -6.9374, df = 9, p-value = 6.779e-05
| alternative hypothesis: true mean is not equal to 0
| 95 percent confidence interval:
|  -6.497808 -3.302192
```

```
| sample estimates:
| mean of x
|      -4.9
```

That the p-value is approximately equal to 0 implies a significant treatment effect.

6.2 Wilcoxon Signed Rank Test

The **Wilcoxon Signed-Rank Test** is a nonparametric alternative to the **paired t-test**, used when comparing **two related (paired) samples** where data is **not normally distributed**. It assesses whether the **median difference** between pairs is zero.

The following **Illustrative Example Data** will be used to explain the procedure of the **Wilcoxon Signed Rank Test**

A study investigates whether a new pain relief medication reduces pain scores (on a 0-10 scale) in patients with chronic back pain. Pain scores are measured before and after 1 week of treatment.

Patient	Before Treatment (X)	Before Treatment (Y)	Difference (X - Y)	|X – Y|	Rank
1	7	5	-2	2	6.5
2	6	6	0	RM	RM
3	8	7	-1	1	3
4	9	5	-4	4	9
5	5	4	-1	1	3
6	7	8	+1	1	3
7	6	3	-3	3	8
8	8	9	+1	1	3
9	4	2	-2	2	6.5
10	7	6	-1	1	3

- There are five 1s ➔ take average rank for each 1: (1 + 2 + 3 + 4 + 5)/5 = 3
- There are two 2s➔ take average ranks for each 2: (6 + 7)/2 = 6.5
- 3 ➔ 8
- 4 ➔ 9

Steps for Wilcoxon Signed Rank Test

- **Exclude zero differences** (Patient 2 is removed, remaining n=9).
- **Rank absolute differences** (ignoring sign):
  - Differences: -2, -1, -4, -1, +1, -3, +1, -2, -1
  - Absolute: 2, 1, 4, 1, 1, 3, 1, 2, 1

  - Sorted: 1, 1, 1, 1, 1, 2, 2, 3, 4
  - Ranks: 1s → avg rank = 3, 2s → 6.5, 3 → 8, 4 → 9.
- **Sum ranks for positive and negative differences**:
  - Positive differences (+1, +1): Ranks = 3, 3 → Sum = 6.
  - Negative differences: Sum = 3 + 6.5 + 6.5 + 8 + 9 + 3 = 36.
- **Test statistic (W)** = smaller sum = 6.
- **Compare to critical value** (from the following Wilcoxon crtical value table) at $\alpha = 0.05$ for $n = 9$ which is 5.

	Alpha value				
n	**0.005**	**0.01**	**0.025**	**0.05**	**0.10**
5	-	-	-	-	0
6	-	-	-	0	2
7	-	-	0	2	3
8	-	0	2	3	5
9	0	1	3	5	8
10	1	3	5	8	10
11	3	5	8	10	13
12	5	7	10	13	17
13	7	9	13	17	21
14	9	12	17	21	25
15	12	15	20	25	30
16	15	19	25	29	35
17	19	23	29	34	41
18	23	27	34	40	47
19	27	32	39	46	53
20	32	37	45	52	60
21	37	42	51	58	67
22	42	48	57	65	75
23	48	54	64	73	83
24	54	61	72	81	91
25	60	68	79	89	100
26	67	75	87	98	110
27	74	83	96	107	119
28	82	91	105	116	130
29	90	100	114	126	140
30	98	109	124	137	151

Since the $TS = 6 > 5 = CV_{0.05}$, the null hypothesis is rejected. That means the treatment is effective. `Next, we use R function`wilcox.test()‘ to perform the signed rank test:

```r
# Pain scores (0-10 scale)
before <- c(7, 6, 8, 9, 5, 7, 6, 8, 4, 7)
after <- c(5, 6, 7, 5, 4, 8, 3, 9, 2, 6)

# Compute differences (After - Before)
differences <- after - before

# Perform the test (paired=TRUE)
wilcox.test(before, after, paired = TRUE, exact = FALSE, correct = FALSE)
```

```
|
|   Wilcoxon signed rank test
|
| data:  before and after
| V = 39, p-value = 0.04639
| alternative hypothesis: true location shift is not equal to 0
```

The p-value (0.046) is less than 0.05, which means we reject the null hypothesis. Both the p-value and critical value methods lead to the same conclusion.