Chapter 10. Two Sample Tests

Previously: Three types of **one sample test** *(each with one- and two-tailed versions)*

Normal z-test

Compare **sample mean** $\overline{x}$ to null value $\boldsymbol{\mu_0}$ with **known** std.dev. $\boldsymbol{\sigma_X}$

Requirement. $n > 30$ or X data is normal

t-test

Compare **sample mean** $\overline{x}$ to null value $\boldsymbol{\mu_0}$ with **unknown** std.dev. (use $\boldsymbol{s}$ instead)

Requirement. X data is normal

Proportion test

Compare **sample proportion** $\hat{p} = \frac{x}{n}$ to null value $\boldsymbol{p_0}$ *(probably use "+4 rule")*

Requirement. $\frac{x}{n}$ "not near" 0 or 1

Important ones.

Steps:

1. *Convert sample value to "**statistic**" (or "**score**") with respect to null value*
2. *Compute two-tailed (or one-tailed) probability of statistic: "**p-value**"*
3. ***Reject if** $p < \alpha$ ("statistically significant difference"), otherwise **Fail to Reject** ("not statistically significant difference")*

Two Sample Tests.

Instead of comparing observed value to given value μ_0 *or* p_0, we compare **two observed values** *with each other*!

Normal z-test

Compare **two sample means** $\overline{x}_A$ and $\overline{x}_B$ with **known** std.dev. $\boldsymbol{\sigma_A}, \boldsymbol{\sigma_B}$

Requirement. $n_A, n_B > 30$ or X data is normal

t-test

Compare **two sample means** $\overline{x}_A$ and $\overline{x}_B$ with unknown std. dev. (use $\boldsymbol{s_A}$ and $\boldsymbol{s_B}$ instead)

Requirement. X data is normal

Proportion test

Compare **two sample proportions** $\hat{p}_A = \frac{\#a}{n}$ and $\hat{p}_B = \frac{\#b}{m}$

Requirement. $\frac{\#a}{n}, \frac{\#b}{m}$ "not near" 0 or 1

t-test splits into three types!

Lower Power | Higher Power

Unequal Variance.

"Add" the sample std.dev. to get combined s_{total}.

*Higher std.dev.; degrees of freedom is **"average"***

Equal Variance.

"Pool" the sample std.dev. to get a high quality estimate s_{pool}!

*Lower std.dev.; degrees of freedom is **sum**.*

Paired Samples.

Compute **difference** of paired samples then do *one sample t-test*

*Use if X_A and X_B are **not independent!***

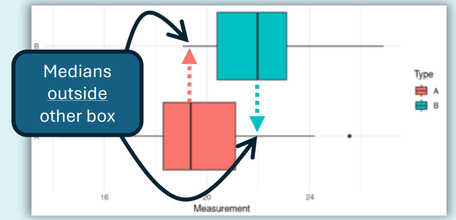
Chapter 10. Equal and Unequal Variance t-Test Example

Example Data for Two Sample t-Test

Two possible structures for data:

- **two columns** of **values** to compare
- one column of **values** and another column giving ***type*** of each value *(“tidy” organization)*

	A	B	C
1	Type	Measurement	
2	A	20.99	
3	A	19.72	
4	B	20.80	
5	B	25.70	
6	A	21.30	
7	A	23.05	
8	A	19.53	



Boxplot:

- Do medians appear to be different?
- Are medians *outside* of each other’s quartile box?

Two sample t-test gives p-value for equality of means.

Standard test **does not** assume equal var.
Test for equality of means of inputs

- compare two columns
- compare two categories of values

```
# Perform a two-sample t-test not assuming equal variance                [12]
t.test(Measurement ~ Type, data = two_sample_unequal_var, var.equal = FALSE)

	Welch Two Sample t-test

data:  Measurement by Type
t = -3.8538, df = 50.248, p-value = 0.0003311
alternative hypothesis: true difference in means between group A and group B
is not equal to 0
95 percent confidence interval:
 -3.305923 -1.040744
sample estimates:
mean in group A mean in group B
       19.75967        21.93300
```

Reject Null Hypothesis

Means are different!

Test above compares “**Measurement**” by “**Type**”

- Comparing means *19.76* and *21.93*
- p-value is $0.0003 \;=\; 0.03\%$
- *df = 50.2 (unequal variance tests have decimal df) each sample had less than 50 data points*

equal variance test has **more power**

so it gives **better p-value**

Equal variance test has **higher power** *(and lower p-values)…*
only use if you are **certain** std.dev. should be same!
(maybe do <u>F-test for equal variance</u> first??)

```
# Perform a two-sample t-test ASSUMING equal variance                    [13]
t.test(Measurement ~ Type, data = two_sample_unequal_var, var.equal = TRUE)

	Two Sample t-test

data:  Measurement by Type
t = -4.0359, df = 68, p-value = 0.0001404
alternative hypothesis: true difference in means between group A and group B
is not equal to 0
95 percent confidence interval:
 -3.247907 -1.098760
sample estimates:
mean in group A mean in group B
       19.75967        21.93300
```

p-value is even **lower**, because equal variance test has **higher power**

Test above compares “**Measurement**” by “**Type**”

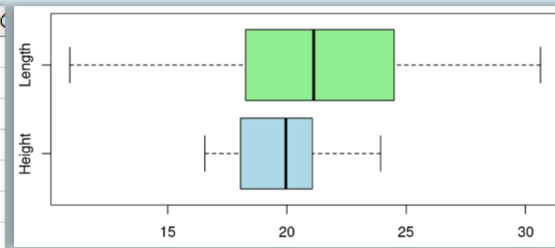
- Comparing means *19.76* and *21.93*
- p-value is $0.0001 \;=\; 0.01\%$
- *df = 68 ($68 + 2 = 70$ **total** data points used)*

Chapter 10. Paired Sample t-Test Example

Example Data for Paired Sample t-Test

- **two columns** of **values** to compare
- columns are **pairs** of observations from same source ***(columns must have same length)***
- note: ‘*equal means*’ is equivalent to ‘*mean difference is 0*’

	A	B
1	**Height**	**Length**
2	19.66	18.43
3	17.78	16.61
4	16.56	21.50
5	21.06	25.50
6	21.24	26.30
7	22.01	17.93
8	[illegible]	[illegible]



Paired sample data will often have one variable with much larger variance than the other!

By using the difference, we reduce this variance!

Paired sample t-test gives p-value for $\boxed{\text{equality of means}} \Longleftrightarrow \boxed{\text{mean difference} = 0}$

Two sample t-test

Usual t-test computes means then takes difference.

```
# Perform a sample t-test (not paired sample)
t.test(two_sample_paired$Height, two_sample_paired$Length)


    Welch Two Sample t-test

data:  two_sample_paired$Height and two_sample_paired$Length
t = -2.424, df = 53.806, p-value = 0.01874
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.1175884 -0.2949116
sample estimates:
mean of x mean of y
 19.71500  21.42125
```

If $\alpha = 1\%$ then
Fail to Reject Null Hypothesis

paired sample test has **more power**

so it gives **better p-value**

Paired sample t-test

Paired t-test takes difference then computes means!

```
# Perform a paired sample t-test on the two_sample_paired data
t.test(two_sample_paired$Height, two_sample_paired$Length, paired = TRUE)


    Paired t-test

data:  two_sample_paired$Height and two_sample_paired$Length
t = -2.8818, df = 39, p-value = 0.006398
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -2.9038362 -0.5086638
sample estimates:
mean difference
      -1.70625
```

Reject the Null Hypothesis
Means are definitely different!

Test above compares “**Height**” to “**Length**”

- **Mean values.** Height: 19.7
  Length: 21.4
- p-value is $0.019 = 1.9\%$
- *df = 53.8* *(unequal variance test; each sample had between 25 and 50 data points)*

Test above compares “**Height**” to “**Length**”

- **Avg difference** (Height – Length) is -1.71
  Note: $19.7 - 21.4 = -1.71$
- p-value is $0.006 = 0.6\%$
- *df = 39* *($39 + 1 = 40$ **pairs** of data points used $= 80$ data points total)*

Chapter 10. Proportion Test Example

Example Data for Proportion Test

Column of **categorical values**, e.g.

- *yes / no*
- *A / B / C*
- *1 / 2 / 3 / 4*

Test will compare proportion of values against null hypothesis proportion.

	A	B	C
1	**yes/no**	**choice**	**value**
2	no	C	4
3	yes	A	2
4	no	C	2
5	yes	C	3
6	yes	B	2
7	no	A	2
8	yes	B	3

Example:

- bar chart shows 41 *'no'* and 59 *'yes'*
- want to test whether this is just random variation from 50% *'no'* and *'yes'*

Single sample population proportion test

Data shows 59/100 '*yes*' answers.

Use a proportion test against
H_0 : true proportion is 50%

to test whether this is within range of expected random variation

```
# Perform a population proportion test for "yes" vs null hypothesis p = 0.5
yes_count <- sum(population_proportions$`yes/no` == "yes")
total_count <- nrow(population_proportions)

prop.test(yes_count, total_count, p = 0.5)

	1-sample proportions test with continuity correction

data:  yes_count out of total_count, null probability 0.5
X-squared = 2.89, df = 1, p-value = 0.08913
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.4870348 0.6859677
sample estimates:
   p
0.59
```

Note: R uses χ^2 test for these, which is better...

Two sample population proportion test

Data shows 59/100 '*yes*' and 41/100 '*no*'.

Use a proportion test against
H_0 : true proportions are equal (both 50%)

to test whether this is within range of expected random variation

```
# Population proportion test

sum(population_proportions$`yes/no` == "yes")
m(population_proportions$`yes/no` == "no")
 nrow(population_proportions)

prop.test(c(yes_count, no_count), c(total_count, total_count))

	2-sample test for equality of proportions with continuity correction

data:  c(yes_count, no_count) out of c(total_count, total_count)
X-squared = 5.78, df = 1, p-value = 0.01621
alternative hypothesis: two.sided
95 percent confidence interval:
 0.03367327 0.32632673
sample estimates:
prop 1 prop 2
  0.59   0.41
```