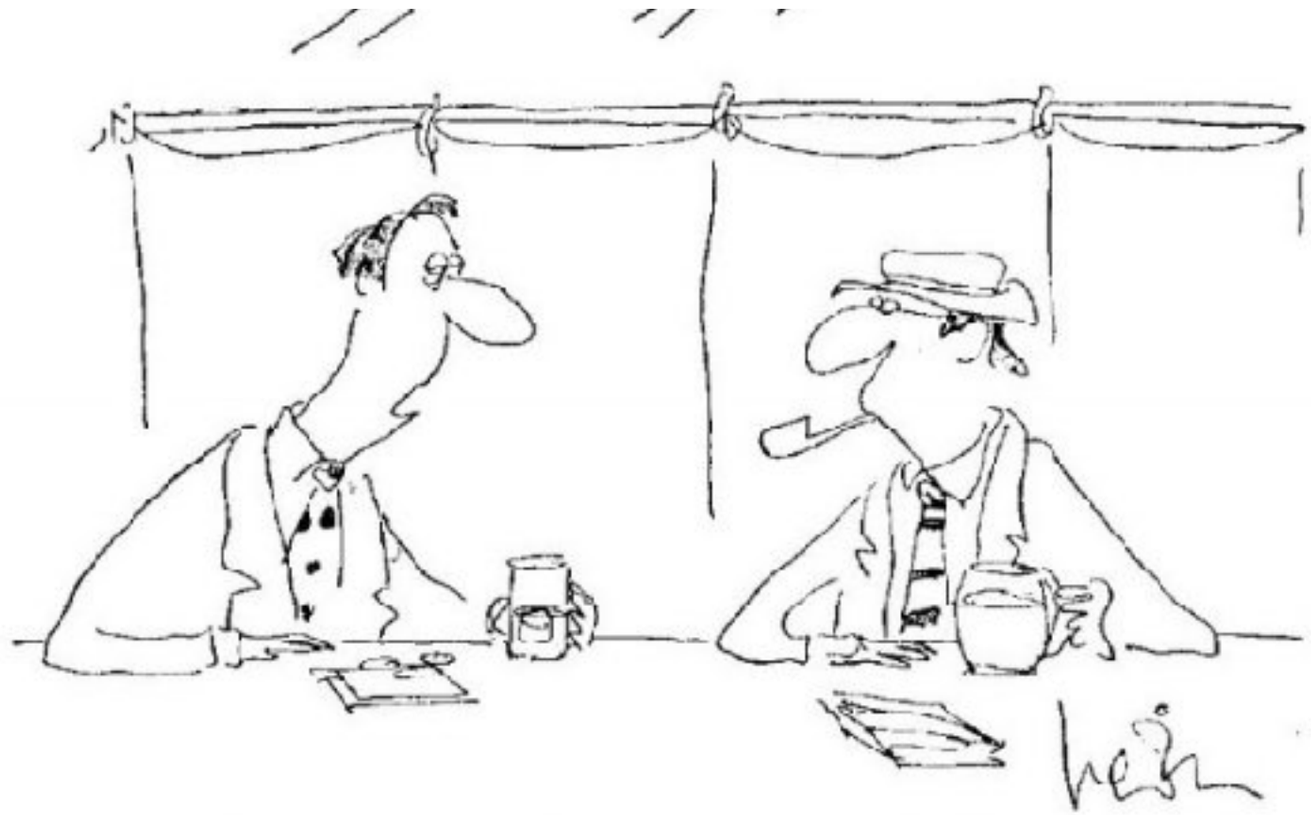
CO3091 - Computational Intelligence and Software Engineering

Lecture 07



"Well, I'll be damned if I'll defend to the death your right to say something that's statistically correct."

Image from: http://online-behavior.com/sites/default/files/imagecache/Content/articles/statistical-truth.jpg

Evaluating and comparing algorithms — Part I

Leandro L. Minku

Announcements

- Coursework 1 is out!
- Next problem class is on Thursday, to enable you to ask questions about coursework.

Overview

- Comparison of computational intelligence algorithms.
- The need for statistical tests.
- Types of statistical tests.
- Wilcoxon tests in R.

Comparison of Computational Intelligence Algorithms

- Different algorithms and/or configurations may perform differently.
- The best algorithm and/or configuration to use depends on the problem (no free lunch).
- If we don't know beforehand which algorithm is better for our problem, we need to compare different algorithms and/or configurations.
- This comparison is not straightforward, due to the stochastic behaviour of many of the computational intelligence algorithms.

Stochastic Behaviour of Computational Intelligence Algorithms

- Computational intelligence frequently involves a certain amount of randomness (stochastic behaviour):
  - Randomness in the data samples, affecting the algorithms' results (we will learn about this later).
  - Randomness in the algorithms themselves.
    - E.g., random initial population, probability of crossover, probability of mutation, etc.

If we run the algorithm different times on the same problem instance, we will get different results.

Stochastic Behaviour of Computational Intelligence Algorithms

- Example: run an EA on traveling salesman problem using different seeds.
- Seed 1: min distance = 1900.58
- Seed 3: min distance = 1719.35
- Using the same seed allows us to repeat exactly the same sequence of pseudorandom numbers.
- However, the algorithm is still stochastic, i.e., it still uses a sequence of pseudorandom numbers.

If algorithms perform differently with different random seeds, how to compare them?

Run algorithm 1:	Run algorithm 2:
0.9751186599	0.9379836847
0.8036808732	0.9442552933

We need to run the algorithms many times (e.g., 30+) with different random seeds to check what the typical behaviour of the algorithms is.

How to Compare Different Approaches and/or Configurations?

Runs

Best Fitness EA1	Best Fitness EA2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

EA2's best fitness in the last generation of run 1.

EA2's best fitness in the last generation of run 2.

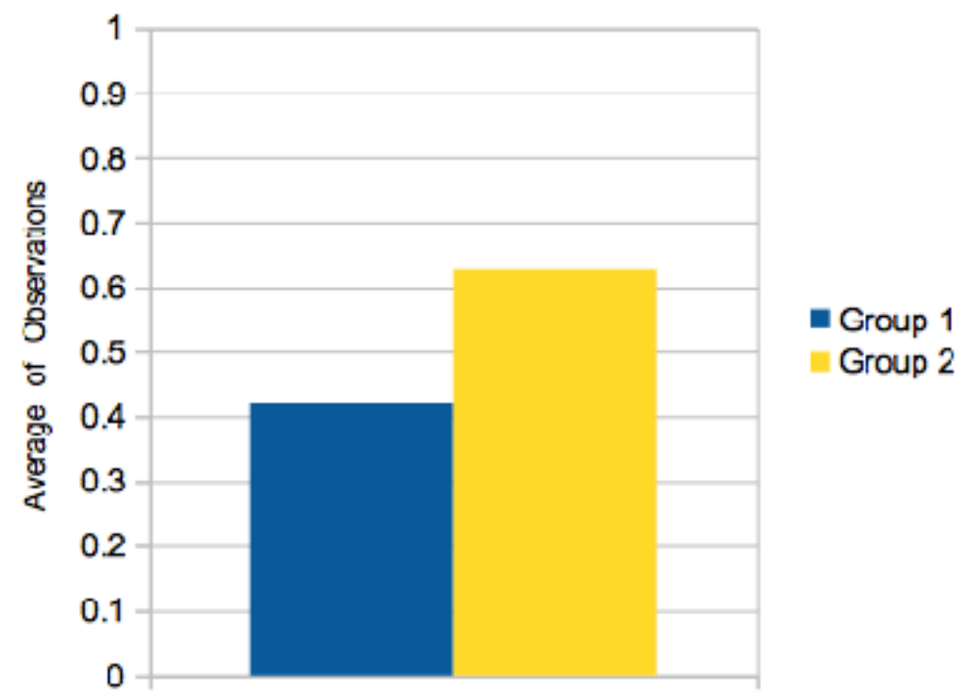
In statistics, each of the cells is referred to as an observation, and each column is called a group.

How to compare different groups of observations?

Averages

Average group 1: 0.4212

Average group 2: 0.6264



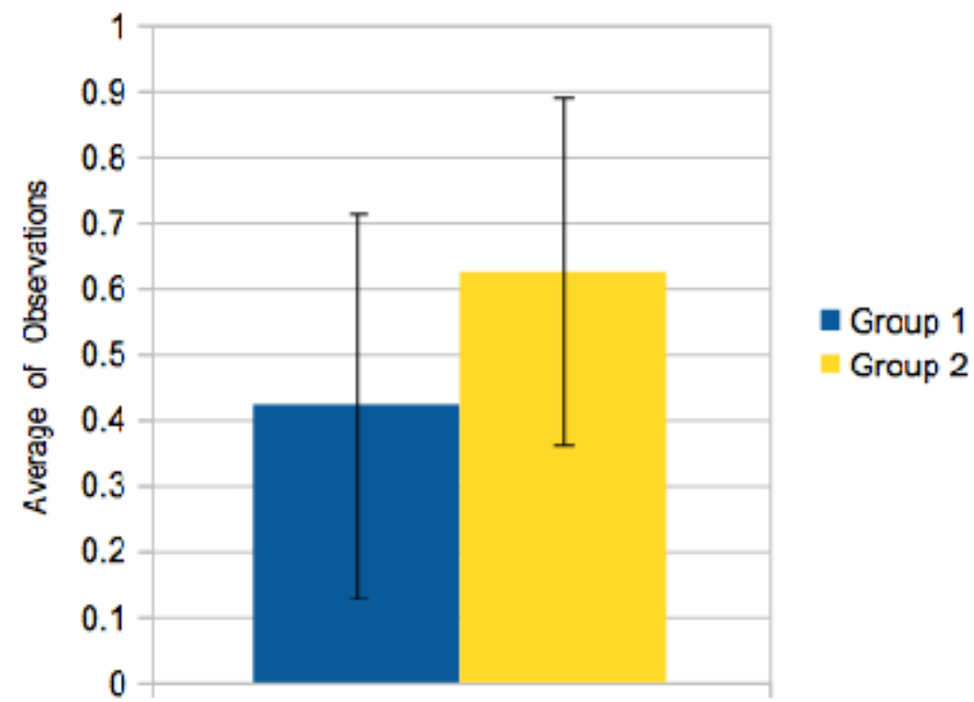
Problem 1: doesn't take into account variations in the observations.

Problem 2: can be affected by outliers (very small or very large exceptional observations).

Averages + Standard Deviations

Average group 1: 0.4212

Average group 2: 0.6264



Problem 1: difficult to know if groups are different or not.

Problem 2: still be affected by outliers.

Median

Example of median calculation for odd number of observations:

0.6, 0.75, 0.62, 0.65, 0.7, 0.8, 0.81, 0.000001, 0.8

Median

Example of median calculation for odd number of observations:

Sorted observations

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81

Median

Example of median calculation for odd number of observations:

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81

Median
(middle point)

Median

Example of median calculation for even number of observations:

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81, 0.81

Median = 0.725
(average of two middle points)

Median

Example of median calculation for even number of observations:

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81, 0.81

Outlier

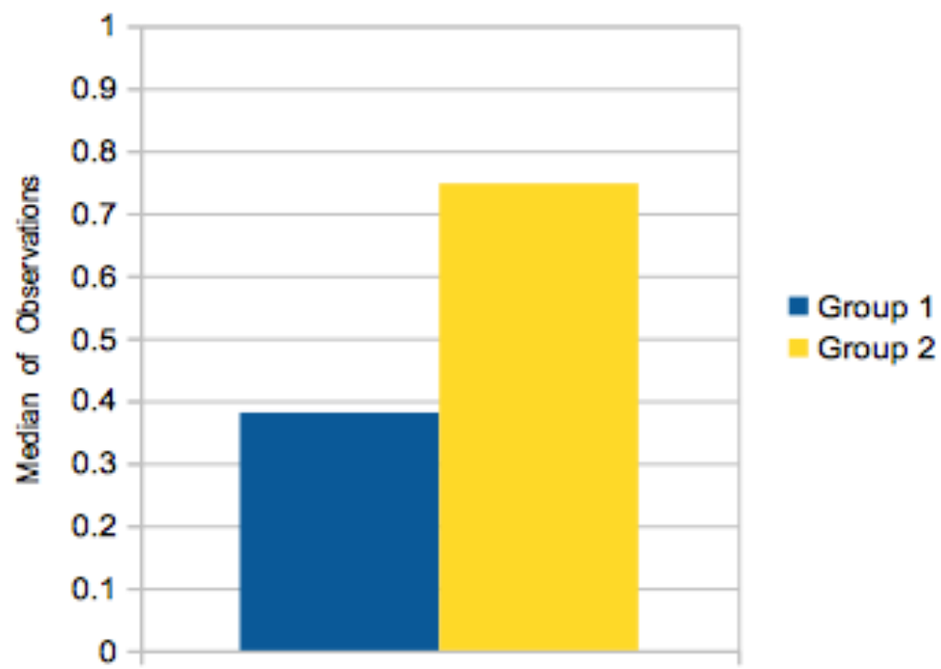
Median = 0.725
(average of two middle points)

Medians are less affected by outliers than averages.

Median

Median group 1: 0.3805

Median group 2: 0.7474



Problem 1: doesn't take into account variations in the observations.

Median, 1st and 3rd Quartiles

Quartiles divide the observations into 4 chunks.
Median can be referred to as 2nd quartile.

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81

1st Quartile = median of left observations

Median (middle point)

3rd Quartile = median of right observations

Median, 1st and 3rd Quartiles

Quartiles divide the observations into 4 chunks.
Median can be referred to as 2nd quartile.

0.000001, 0.6, 0.62, 0.65, 0.7, 0.75, 0.8, 0.8, 0.81

1st Quartile = median of left observations

Median (middle point)

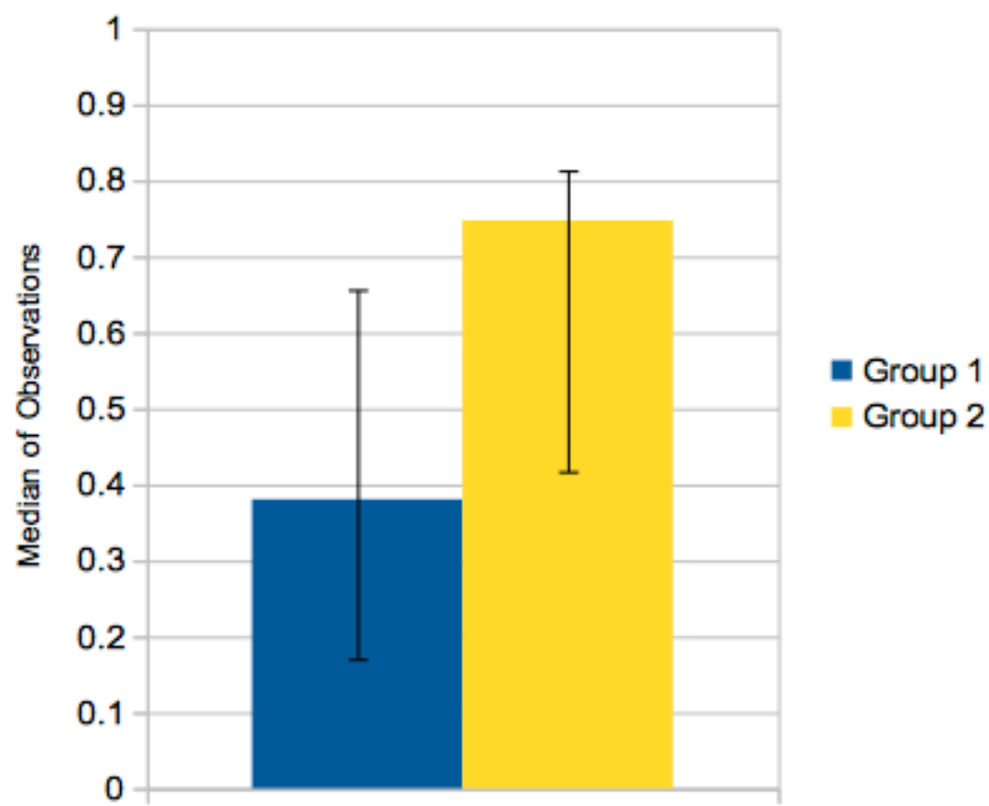
3rd Quartile = median of right observations

Quartiles are less affected by outliers than standard deviations.

Median + 1st and 3rd Quartiles

Median group 1: 0.3805

Median group 2: 0.7474



Problem: sometimes it is still difficult to know if groups are different or not.

How to Compare Different Approaches and/or Configurations?

We need a scientific method!

Statistical hypothesis test:
scientific method for testing a statistical hypothesis, e.g., that two or more algorithms perform similarly.

Statistical Hypothesis Tests

1. Decide what to compare and formulate hypothesis.
2. Choose appropriate statistical hypothesis test.
3. Run the statistical hypothesis test to check whether hypothesis is or is not rejected.

Statistical Hypothesis Tests

1. Decide what to compare and formulate hypothesis.
2. Choose appropriate statistical hypothesis test.
3. Run the statistical hypothesis test to check whether hypothesis is or is not rejected.

What to Compare?

- In this lecture: performance (e.g., fitness) of two algorithms and/or configurations.
  - This means we have two groups of observations.
- Make sure the comparison between two groups is fair!
  - Use the same number of fitness evaluations, unless the purpose of your comparison is to determine whether having more fitness evaluations helps to get better results.
  - E.g., when comparing simulated annealing with 10,000 iterations against an evolutionary algorithm with a population of size 10 which generates 10 new offspring per generation, the evolutionary algorithm should run for ??? generations.

What to Compare?

- In this lecture: performance (e.g., fitness) of two algorithms and/or configurations.
  - This means we have two groups of observations.
- Make sure the comparison between two groups is fair!
  - Use the same number of fitness evaluations, unless the purpose of your comparison is to determine whether having more fitness evaluations helps to get better results.
  - E.g., when comparing simulated annealing with 10,000 iterations against an evolutionary algorithm with a population of size 10 which generates 10 new offspring per generation, the evolutionary algorithm should run for 1000 - 1 generations.

Statistical Hypothesis Tests for Two Groups

Statistical hypothesis: scientific hypothesis about how the groups of observations compare to each other.

- Two-tailed (two-sided) test:

Null Hypothesis

H0: Group 1 = Group 2

Alternative Hypothesis

H1: Group 1 ≠ Group 2

H1 is usually the desirable outcome.

Example:

- H0: Fitness(EA1) = Fitness (EA2)

  H1: Fitness(EA1) ≠ Fitness(EA2)

Statistical Hypothesis Tests for Two Groups

Statistical hypothesis: scientific hypothesis about how the groups of observations compare to each other.

- One-tailed (one-sided) test:
  - H0: Group 1 ≤ Group 2

    H1: Group 1 > Group 2

    Only recommended when the consequences of having an undesirable outcome of < is the same as an undesirable outcome of =.
  - H0: Group 1 ≥ Group 2

    H1: Group 1 < Group 2

    Only recommended when the consequences of having an undesirable outcome of > is the same as an undesirable outcome of =.
  - Rarely used in computational intelligence experiments.

Statistical Hypothesis Tests

1. Decide what to compare and formulate hypothesis.
2. Choose appropriate statistical hypothesis test.
3. Run the statistical hypothesis test to check whether hypothesis is or is not rejected.

Statistical Hypothesis Tests

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

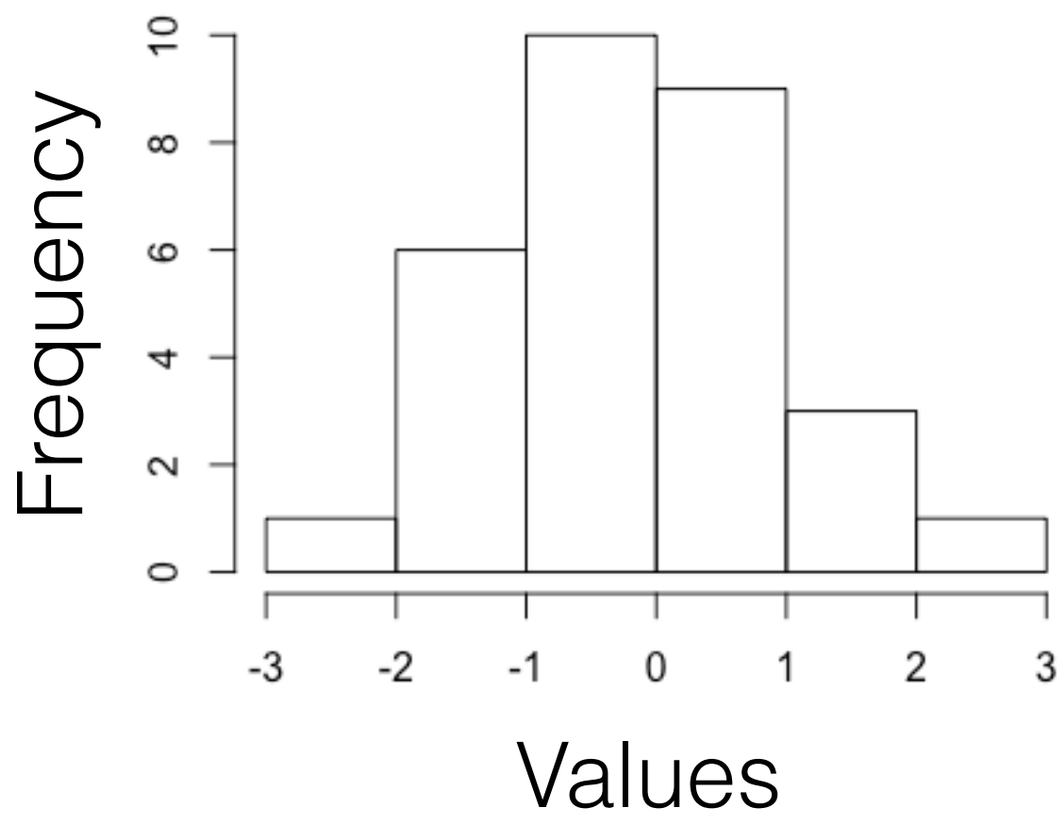
Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Normality Assumption

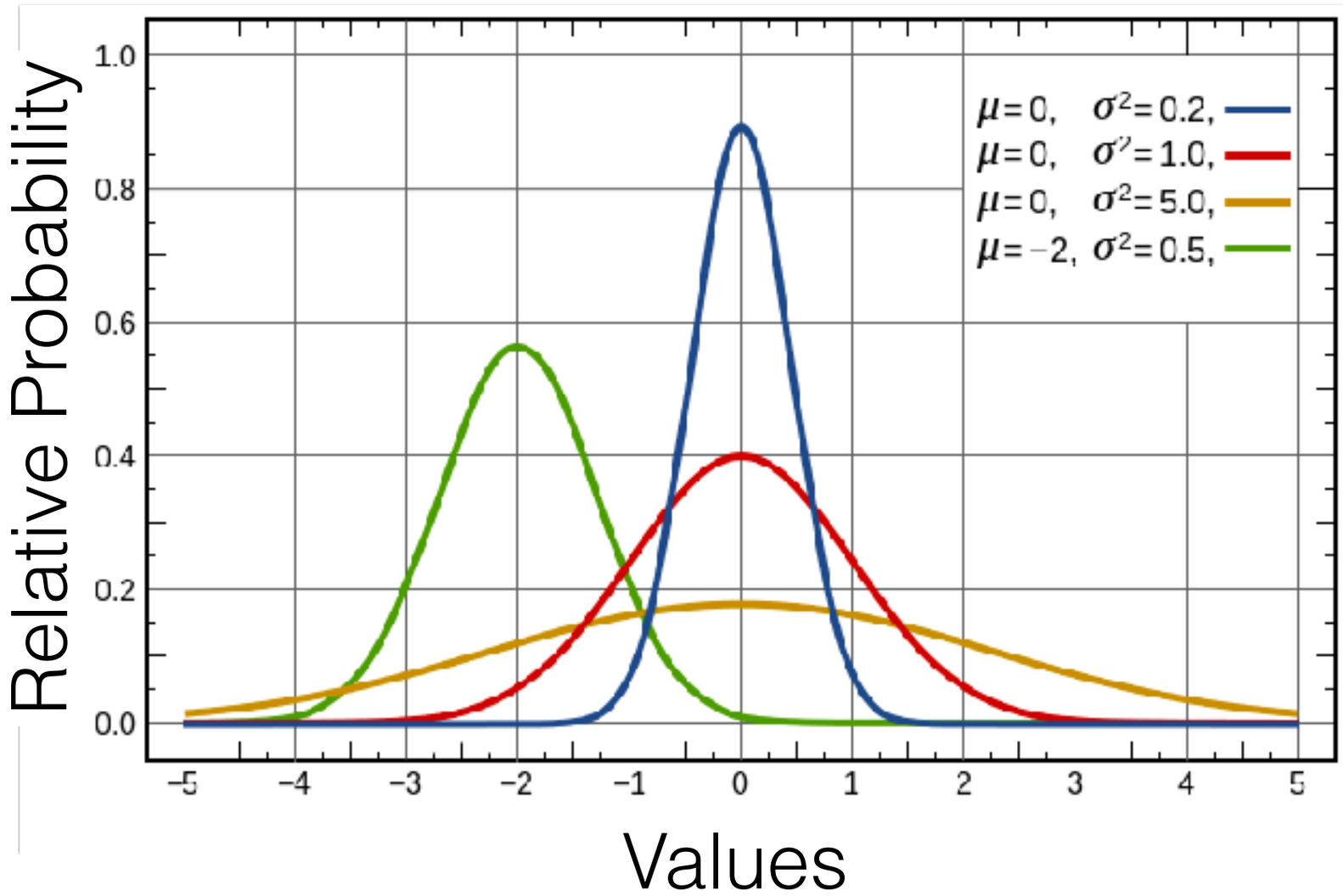
- Normality assumption: the frequency of observed values for each group follows a normal distribution.



Normality Assumption

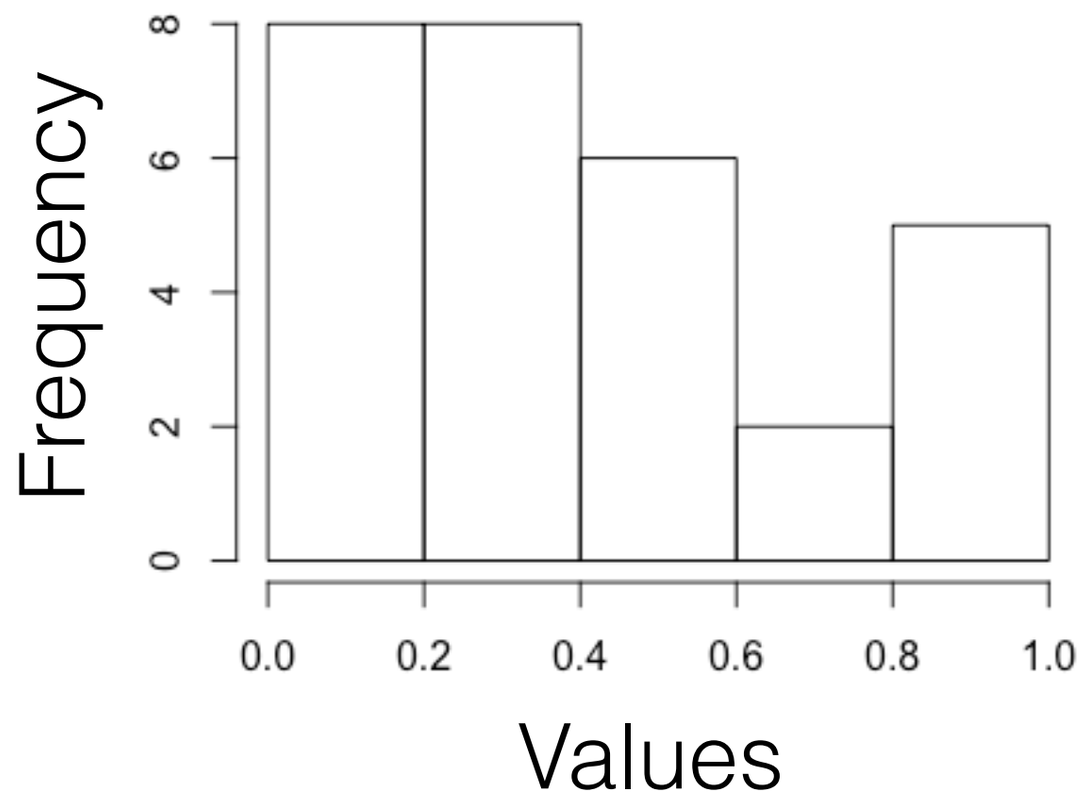
- Normality assumption: the frequency of observed values for each group follows a normal distribution.

Image from: https://upload.wikimedia.org/wikipedia/commons/thumb/7/74/Normal_Distribution_PDF.svg/720px-Normal_Distribution_PDF.svg.png



Normality Assumption

- Example of observations not following a normal distribution:



Which Test to Use?

Parametric tests are more powerful (better at detecting differences), than non-parametric ones, but can be highly affected by violations of assumptions.

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

We will use non-parametric tests in this course.

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Example: using the same initial conditions to both groups (e.g., same initial population).

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)		ANOVA
	Paired (related)		ANOVA
Non-parametric (no normality)	Unpaired (independent)		Kruskal-Wallist test
	Paired (related)		Friedman test

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709

Which Test to Use?

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Example: using different random seeds or number of runs.

Data Distribution		2 groups		n groups (n>2)
		Group 1	Group 2	
Parametric (normality)	Unpaired (independent)	0.8036808732 0.1546026852 0.1507085019	0.9442552933 0.7277129425 0.4319811615	ANOVA
	Paired (related)	0.9751186599 0.4602321477 0.0132238786 0.0175114877	0.9379836847 0.786503003 0.8191139316 0.9236880897	ANOVA
Non-parametric (no normality)	Unpaired (independent)	0.9041741739 0.8697700955 0.676352134	0.8155635942 0.7694358404 0.3217702059	Kruskal-Wallist test
	Paired (related)	0.5182328166 0.0516411681 0.5426649651 0.4973629257		Friedman test

Which Test to Use?

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Similar idea for n groups.

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Which Test to Use?

Paired tests use more information, being more powerful (i.e., better at detecting significant differences).

Data Distribution		2 groups	n groups (n>2)
Parametric (normality)	Unpaired (independent)	Unpaired t-test	ANOVA
	Paired (related)	Paired t-test	ANOVA
Non-parametric (no normality)	Unpaired (independent)	Wilcoxon rank-sum test = Mann–Whitney U test	Kruskal-Wallist test
	Paired (related)	Wilcoxon signed-rank test	Friedman test

Statistical Hypothesis Tests

1. Decide what to compare and formulate hypothesis.
2. Choose appropriate statistical hypothesis test.
3. Run the statistical hypothesis test to check whether hypothesis is or is not rejected.

Test Output

- The output of the test is a numerical value referred to as test statistic, used to decide whether or not to reject H0.
    - H0: Group 1 = Group 2
    - H1: Group 1 ≠ Group 2
    - Not rejecting H0 means that **no statistically significant difference** has been found between groups 1 and 2.
    - Rejecting H0 means that there is **statistically significant difference** between groups 1 and 2, i.e., unlikely that the different values in groups 1 and 2 have occurred purely by change.
    - Once we know they are different, we can look at the medians to gain an insight into which of the groups is better.
- We normally compare the test statistic to some threshold value to decide whether or not to reject H0.
- The threshold value depends on the test being used.

Test Output

- P-values are usually included in the test output and are easier to interpret and use.
- Once we have the p-value, we can decide whether or not to reject H0 without having to look for thresholds specific to the corresponding statistical test.
- Estimated probability of observing a given test statistic assuming that H0 is true.
  - If the p-value is high, this means that the probability of observing the given test statistic when H0 is true is high. So, should we reject or not reject H0?

Test Output

- P-values are usually included in the test output and are easier to interpret and use.
- Once we have the p-value, we can decide whether or not to reject H0 without having to look for thresholds specific to the corresponding statistical test.
- Estimated probability of observing a given test statistic assuming that H0 is true.
  - If the p-value is high, this means that the probability of observing the given test statistic when H0 is true is high. So, we do not reject H0.
  - If the p-value is low, this means that the probability of observing the given test statistic when H0 is true is low. So, we reject H0.

Rejecting or Not Rejecting H0

- We want small p-values for rejecting H0.
- We set a threshold (level of significance) on how small the p-values should be for rejecting H0.
- Usually, level of significance is set to 0.05.
  - **If p-value ≤ 0.05, we reject H0.**
  - **If p-value > 0.05, we do not reject H0.**
- Level of confidence= 1 - level of significance.
- For critical applications, people may use level of significance of 0.01.

R

- Programming language for statistical computing.
- Can be used to run statistical tests.

Reading Observations

- You can enter observations manually, or you can load observations from a .csv table. E.g.:
  - `observation = read.csv('/Users/llm11/Desktop/observations.csv', header = TRUE, sep = ",")`
- For help with a command:
  - `help(command)`

```
Group 1,Group 2
0.803680873,0.944255293
0.154602685,0.727712943
0.150708502,0.431981162
0.97511866,0.937983685
0.460232148,0.786503003
0.013223879,0.819113932
0.017511488,0.92368809
0.904174174,0.815563594
0.869770096,0.76943584
0.676352134,0.321770206
0.518232817,0.984916141
0.051641168,0.258640987
0.542664965,0.794543475
0.497362926,0.817948571
0.486607913,0.413216708
0.218745577,0.591558823
0.843827421,0.593674664
0.264400949,0.438692375
0.256434446,0.743990941
0.079121486,0.795106819
0.285609383,0.331450863
0.379775917,0.9218094
0.59789627,0.750849697
0.08605325,0.13729544
0.2860286,0.12517536
0.277279003,0.785829481
0.728984666,0.459297733
0.381243886,0.158332721
0.114495351,0.403745207
0.71283282,0.807401962
```

Accessing Observations

- observation[1,2]
- observation[,2]
- observation[1,]

- You can type observation[1,2], observation[,2] and observation[1,] in R to see their content.

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- observation[1,2] —> take the observation from the first row and second column

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- observation[1,2] —> take the observation from the first row and second column
- observation[,2] —>

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- observation[1,2] —> take the observation from the first row and second column
- observation[,2] —> take all the observations from the second column

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- observation[1,2] —> take the observation from the first row and second column
- observation[,2] —> take all the observations from the second column
- observation[1,] —> ?

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- observation[1,2] —> take the observation from the first row and second column
- observation[,2] —> take all the observations from the second column
- observation[1,] —> take all the observations from the first row

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Accessing Observations

- Advantage of languages such as R and Matlab:
  - No need for a loop to get all observations of a given column or row when using the notation explained in the previous slide.
  - Matrix operations are easier than in languages such as Java or C++.
    - A %*% B
  - These languages offer lots of packages with the implementation of methods frequently used by mathematicians and statisticians.
  - In our case, we use the statistical tests.
- Potential disadvantages:
  - No compiler — bugs found at runtime.
  - Slow…….

Group 1	Group 2
0.8036808732	0.9442552933
0.1546026852	0.7277129425
0.1507085019	0.4319811615
0.9751186599	0.9379836847
0.4602321477	0.786503003
0.0132238786	0.8191139316
0.0175114877	0.9236880897
0.9041741739	0.8155635942
0.8697700955	0.7694358404
0.676352134	0.3217702059
0.5182328166	0.9849161406
0.0516411681	0.2586409871
0.5426649651	0.7945434749
0.4973629257	0.8179485709
0.4866079125	0.4132167082
0.2187455767	0.5915588229
0.8438274211	0.5936746635
0.2644009485	0.4386923753
0.256434446	0.743990941
0.0791214858	0.7951068189
0.2856093827	0.3314508633
0.3797759169	0.9218094004
0.5978962695	0.7508496968
0.0860532501	0.1372954398
0.2860286001	0.1251753599
0.2772790031	0.7858294814
0.7289846656	0.4592977329
0.3812438862	0.1583327209
0.114495351	0.4037452065
0.7128328204	0.8074019616

Two-Tailed Wilcoxon Rank-Sum in R

```
wilcox.test(x, y, alternative = "two.sided", paired = FALSE,
conf.level = 0.95)
```

- Example:
  - H0: Group 1 = Group 2
  - H1: Group 1 ≠ Group 2
  - Level of significance = 0.05
- `wilcox.test(observation[,1],observation[,2],alternative = "two.sided",paired=FALSE, conf.level = 0.95)`
- p-value: 0.007647 ≤ 0.05 (reject H0)
  - **Groups 1 and 2 are statistically significantly different.**
  - **Median(group 1) = 0.3805, Median(group 2) = 0.7474**

Two-Tailed Wilcoxon Sign Rank in R

```
wilcox.test(x, y, alternative = "two.sided", paired = TRUE, conf.level
= 0.95)
```

- Example:
  - H0: Group 1 = Group 2

    H1: Group 1 ≠ Group 2

    Level of significance = 0.05
  - `wilcox.test(observation[,1],observation[,2],alternative = "two.sided",paired=TRUE, conf.level = 0.95)`
  - p-value: 0.002766 ≤ 0.05 (reject H0)
    - **Groups 1 and 2 are statistically significantly different.**
    - **Median(group 1) = 0.3805, Median(group 2) = 0.7474**

Completely Equal Pairs of Observations

- `observation = read.csv('/Users/llm11/Desktop/observations_null.csv', header = TRUE, sep = ",")`
- `wilcox.test(observation[,1],observation[,2],alternative = "two.sided",paired=TRUE, conf.level = 0.95)`
  - `p-value = NA`

Group 1,Group 2
1,1
2,2
3,3
4,4
5,5
6,6
7,7
8,8
9,9
10,10
11,11
12,12
13,13
14,14
15,15
16,16
17,17
18,18
19,19
20,20
21,21
22,22
23,23
24,24
25,25
26,26
27,27
28,28
29,29
30,30

Further Reading

- Check the following R help pages:
  - help(read.csv)
  - help(wilcox.test)

- Background information:
  - http://www.biostathandbook.com/wilcoxonsignedrank.html
  - http://www.biostathandbook.com/kruskalwallis.html#mannwhitney

Tomorrow:
lecture 9am GP LTA
lab session 10am