Chapter 8

How to Validate:

Statistical Significance of network-based studies

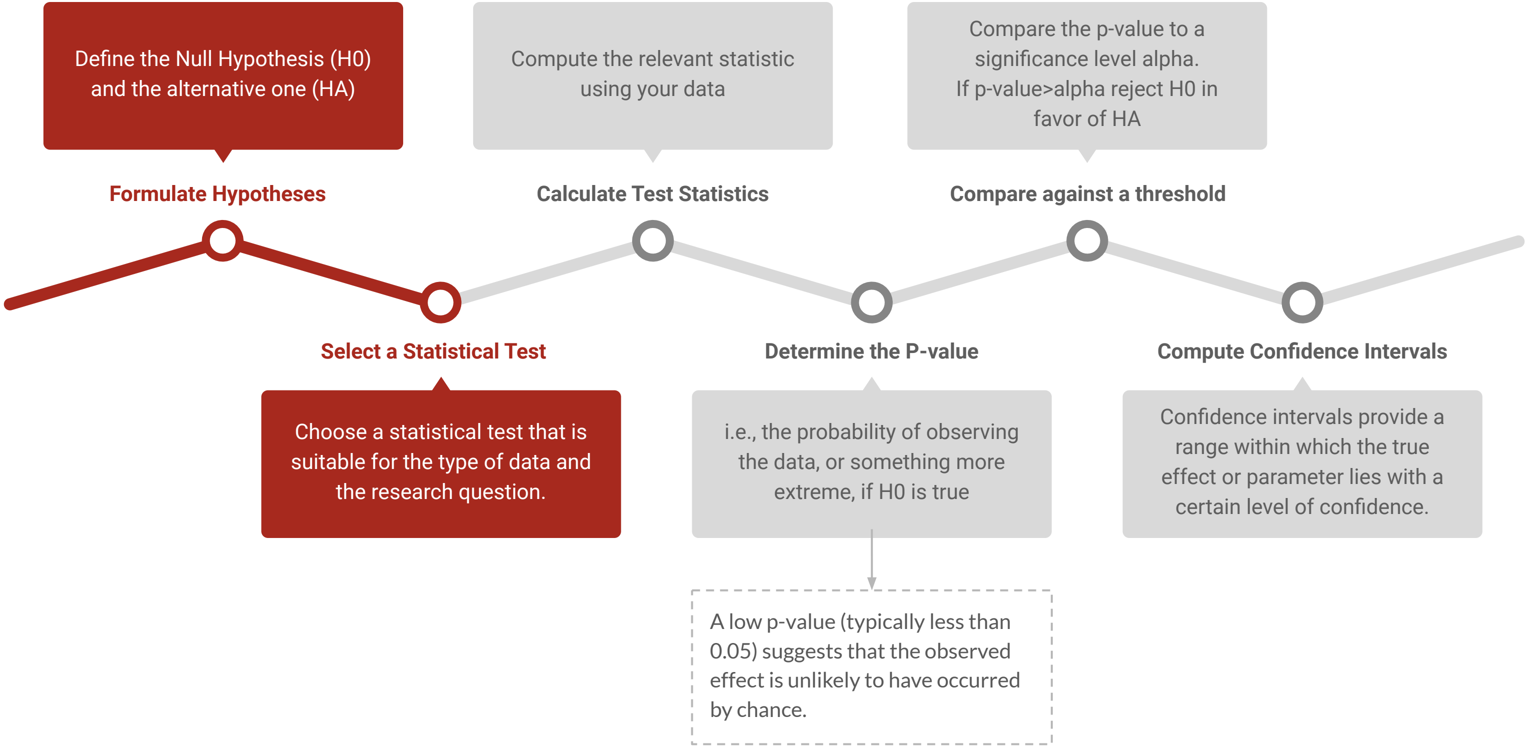
Just, when you thought it was over...

Kudos! You finally managed to perform your experiment!

- You identified the relevant questions;
- Formulated a hypothesis;
- Decided how to model the data;
- Acknowledged the impact of data transformation;
- Identified the proper methodology;
- Measured your target variables...

It's time to statistically validate your work!

Disclaimer
This lecture will focus on a narrow set of methodologies, it should not be considered exhaustive
(or worse, as a substitute for an inferential statistic class)



Define the Null Hypothesis (H0) and the alternative one (HA)

Formulate Hypotheses

Select a Statistical Test

Choose a statistical test that is suitable for the type of data and the research question.

Compute the relevant statistic using your data

Calculate Test Statistics

Determine the P-value

i.e., the probability of observing the data, or something more extreme, if H0 is true

A low p-value (typically less than 0.05) suggests that the observed effect is unlikely to have occurred by chance.

Compare the p-value to a significance level alpha.
If p-value>alpha reject H0 in favor of HA

Compare against a threshold

Compute Confidence Intervals

Confidence intervals provide a range within which the true effect or parameter lies with a certain level of confidence.

Statistical validation

Using statistical methods to determine whether the patterns or effects observed in your data are significant and not due to random chance.

Why is it important?

Significance:
A statistically significant result suggests that the observed pattern or effect is unlikely to be due to random chance, providing evidence against the null hypothesis.

Robustness:
Statistical validation helps ensure that the results are reliable and reproducible.

Objectivity:
It reduces bias by providing an objective framework for evaluating the evidence.

Example in Complex Networks studies

In the context of Network analysis statistical validations (usually) involves:

Generating Null Models

Computing Network Metrics
for both the observed network and the null models.

Statistical Comparison
Use statistical tests (e.g., z-scores, t-tests) to compare the metrics of the observed network against those of the null models.

Assessing Significance
Determine whether the differences between the observed network and the null models are statistically significant.

Null Models

What is a Null Model?

It is a simplified version of a complex system used as a baseline to compare against the observed data.

Key Characteristics of Null Models

1. **Simplification**:
   Null models simplify the original system to focus on the aspect being tested, often by randomizing elements that are not the focus of the hypothesis.

2. **Baseline Comparison**:
   They provide a baseline distribution against which the observed data can be compared, helping to identify significant deviations.

3. **Preserved Properties**:
   Null models typically preserve some properties of the original system (e.g., degree distribution in networks) while randomizing others.

Topological Null Models

Focus on the network’s structure by randomizing connections while preserving certain properties (e.g., degree distribution or clustering coefficient)

… sounds familiar?

Examples

- **Random graphs (Erdős–Rényi model)**: Randomly generate edges between nodes with a fixed probability.
- **Configuration models**: Generate random networks while preserving the original network's degree sequence.
- **Exponential random graph models (ERGMs)**: Statistical models that allow for the inclusion of multiple network properties.
- …

ERGMs

Exponential Random Graph Models are a class of statistical models used to analyze and simulate network data

- **Statistical Foundation:**
  ERGMs are based on the exponential family of probability distributions
- **Model Parameters:**
  Parameters correspond to network statistics (e.g., number of edges, triangles, degree distributions) and determine the likelihood of observing a particular network structure.
- **Configurational Approach:**
  ERGMs model the presence of edges based on the configurations of nodes and edges, capturing complex dependencies beyond simple random graphs.

The probability of a network G under an ERGM is given by:

$$P(G) = \frac{\exp(\theta \cdot g(G))}{Z(\theta)}$$

where:

- θ is a vector of parameters.
- g(G) is a vector of network statistics (e.g., number of edges, triangles).
- Z(θ) is the normalizing constant (partition function) ensuring the probabilities sum to 1.

ERGMs
Invariants & Randomized

Invariants

In an ERGM null model, the invariants are the network statistics included in the model (e.g., edges, triangles). These statistics are preserved on average in the generated networks.

Randomized Components

The randomized components in ERGMs are the specific configurations of nodes and edges that meet the constraints imposed by the model parameters.

We are analyzing a social network of individuals to determine whether the observed clustering is higher than expected by chance.

Observed Network Statistics

- Number of Edges (e)
- Number of Triangles (t)

ERGM Formulation

The ERGM null model can be formulated to match the observed number of edges and triangles

$$P(G) = \frac{\exp(\theta_1 e(G) + \theta_2 t(G))}{Z(\theta_1, \theta_2)}$$

where:

- $\theta1$ corresponds to the number of edges.
- $\theta2$ corresponds to the number of triangles.

Model Fitting

Fit the ERGM to the observed network to estimate the parameters $\theta1$ and $\theta2$

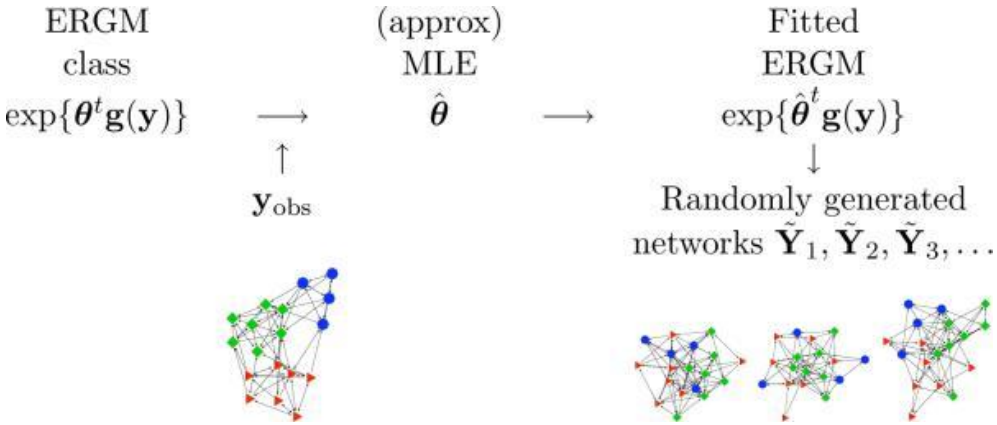
ERGMs

Advantages over ER et. similia

Flexibility:
ERGMs can incorporate various network statistics, making them highly flexible for different types of network data.

Configurational Dependencies:
ERGMs can model complex dependencies and interactions among edges, which simple random graph models cannot.

Statistical Rigor:
Provides a statistically rigorous framework for hypothesis testing and model selection.



ERGM class $\exp\{\boldsymbol{\theta}^t \mathbf{g}(\mathbf{y})\}$ ⟶ (approx) MLE $\hat{\boldsymbol{\theta}}$ ⟶ Fitted ERGM $\exp\{\hat{\boldsymbol{\theta}}^t \mathbf{g}(\mathbf{y})\}$

$\mathbf{y}_{\mathrm{obs}}$

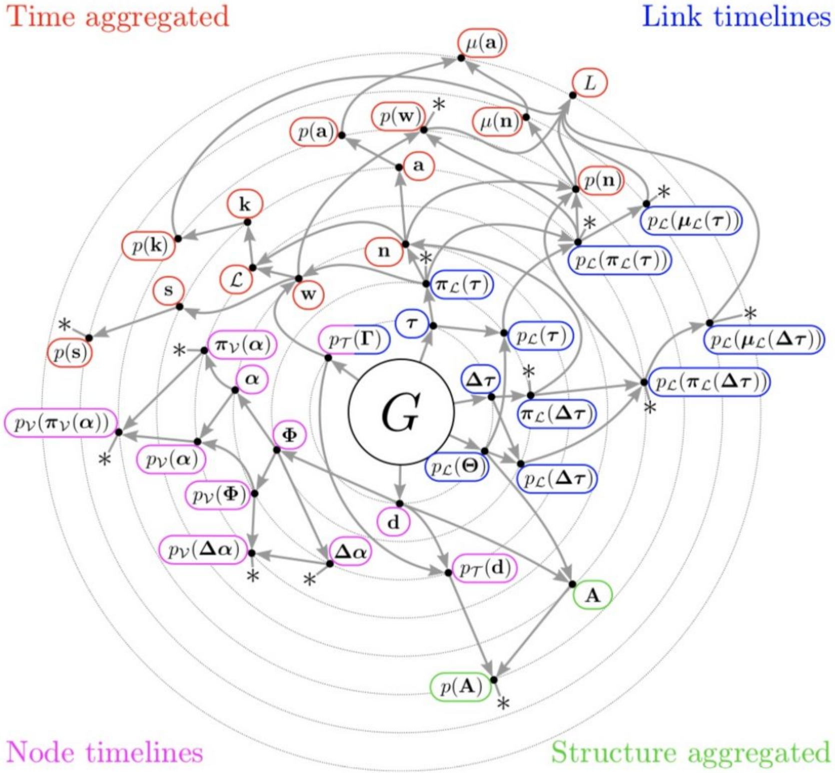
Randomly generated networks $\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2, \tilde{\mathbf{Y}}_3, \ldots$

Dynamic Networks Null Models

We consider as Dynamic Network representation the Snapshot Graphs ones.

Such representation framework allows to define several randomization strategies, either focusing:

- topology
- temporal displacement


Time aggregated
Link timelines
$\mu(\mathbf{a})$
L
$p(\mathbf{w})$
$\mu(\mathbf{n})$
$p(\mathbf{a})$
$\mathbf{a}$
$p(\mathbf{n})$
$\mathbf{k}$
$p(\mathbf{k})$
$\mathbf{n}$
$p_{\mathcal{L}}(\boldsymbol{\mu}_{\mathcal{L}}(\boldsymbol{\tau}))$
$p_{\mathcal{L}}(\boldsymbol{\pi}_{\mathcal{L}}(\boldsymbol{\tau}))$
$\mathcal{L}$
$\mathbf{w}$
$\mathbf{s}$
$\boldsymbol{\pi}_{\mathcal{L}}(\boldsymbol{\tau})$
$\boldsymbol{\tau}$
$p_{\mathcal{T}}(\boldsymbol{\Gamma})$
$\boldsymbol{\pi}_{\mathcal{V}}(\boldsymbol{\alpha})$
$p(\mathbf{s})$
$\boldsymbol{\alpha}$
$p_{\mathcal{L}}(\boldsymbol{\tau})$
$p_{\mathcal{L}}(\boldsymbol{\mu}_{\mathcal{L}}(\boldsymbol{\Delta\tau}))$
G
$\boldsymbol{\Delta\tau}$
$p_{\mathcal{L}}(\boldsymbol{\pi}_{\mathcal{L}}(\boldsymbol{\Delta\tau}))$
$\boldsymbol{\pi}_{\mathcal{L}}(\boldsymbol{\Delta\tau})$
$p_{\mathcal{V}}(\boldsymbol{\pi}_{\mathcal{V}}(\boldsymbol{\alpha}))$
$\boldsymbol{\Phi}$
$p_{\mathcal{L}}(\boldsymbol{\Theta})$
$p_{\mathcal{L}}(\boldsymbol{\Delta\tau})$
$p_{\mathcal{V}}(\boldsymbol{\alpha})$
$p_{\mathcal{V}}(\boldsymbol{\Phi})$
$\mathbf{d}$
$p_{\mathcal{V}}(\boldsymbol{\Delta\alpha})$
$\boldsymbol{\Delta\alpha}$
$p_{\mathcal{T}}(\mathbf{d})$
$\mathbf{A}$
$p(\mathbf{A})$
Node timelines
Structure aggregated

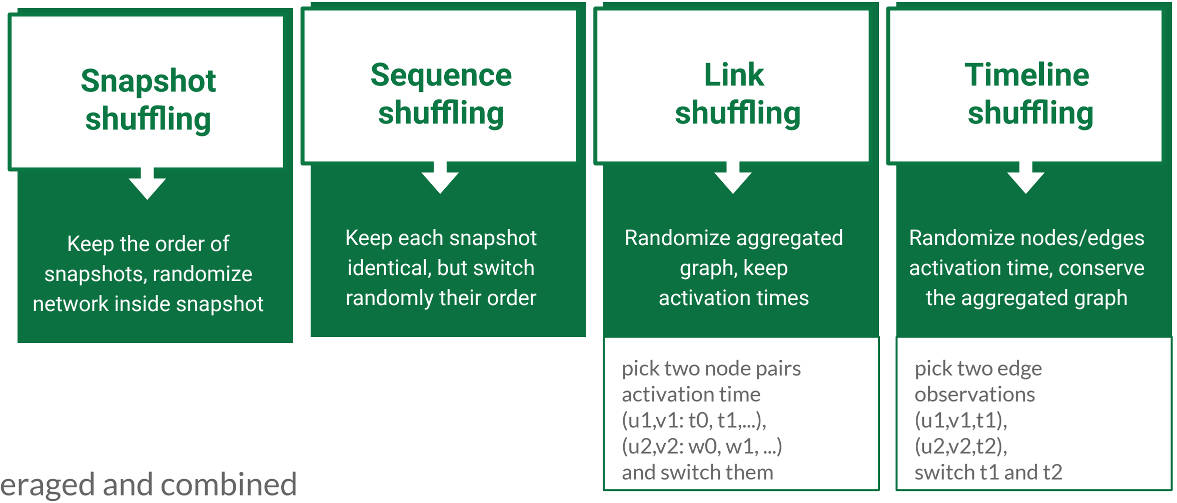
Dynamic Networks Null Models

Randomizations

Random models for static graphs:

- ER, BA, WS, Configuration,....
- Each one preserving different characteristics

In dynamic networks, everything becomes more complicated...



Snapshot shuffling

Keep the order of snapshots, randomize network inside snapshot

Sequence shuffling

Keep each snapshot identical, but switch randomly their order

Link shuffling

Randomize aggregated graph, keep activation times

pick two node pairs activation time
(u1,v1: t0, t1,...),
(u2,v2: w0, w1, ...)
and switch them

Timeline shuffling

Randomize nodes/edges activation time, conserve the aggregated graph

pick two edge observations
(u1,v1,t1),
(u2,v2,t2),
switch t1 and t2

Different Shuffling strategies can be leveraged and combined

Gauvin, Laetitia, et al. "Randomized reference models for temporal networks." arXiv preprint arXiv:1806.04032 (2018).

Dynamic Networks Null Models

Randomizations

Snapshot Shuffling

Invariant: temporal ordering
Randomized: snapshot topology
Use case: Check whether topology affects the target feature

Link Shuffling

Invariant: number of links per node per snapshot
Randomized: time-collapsed topology
Use case: Check whether individual connectivity patterns affects the target feature

Sequence Shuffling

Invariant: snapshot topology
Randomized: temporal ordering
Use case: Check whether topology-evolution ordering effects the target feature

Snapshot Shuffling

Invariant: time collapsed topology
Randomized: node/edge activation rate
Use case: Check whether individual temporal patterns affects the target feature

Dynamic Networks Null Models
Example

We are studying the social interaction network of students in a classroom over a semester.

Goal:
We want to determine whether certain students interact more frequently than would be expected by chance.

Invariants

- Node Degree Sequence:
  The number of interactions each student has per snapshot;
- Total Number of Interactions:
  The total number of interactions in each snapshot.
- Temporal Order of Snapshots

Randomized Components

- Interaction Partners:
  While preserving the degree sequence, we randomize who interacts with whom within each snapshot.
- Timing Within Snapshots:
  If there are timestamps of interactions within each week, these can be randomized as long as the total number of interactions per student is maintained.

Hypergraphs Null Models

Random Hypergraph

Each possible hyperedge is included with a fixed probability independently of other hyperedges.

Parameters:

- Number of vertices (n).
- Probability (p) of each hyperedge being included.

Applications:

- Used to test hypotheses about the randomness of observed hypergraphs.

Invariants

- Number of vertices (n) remains constant.

Randomized Components

- Presence or absence of each hyperedge is determined randomly with probability p.

Example

- Given 4 vertices (A, B, C, D), with p = 0.2, calculate the probability of specific hyperedges. For example, the probability that hyperedge {A, B, C} exists is 0.2.

Hypergraphs Null Models

Configuration Model

Maintains the degree distribution (number of hyperedges incident to each vertex) of the observed hypergraph.

Randomly rewires hyperedges while preserving the degree sequence.

Steps:

- List vertices and their degree sequences.
- Randomly match vertices to form hyperedges until the degree sequences are met.

Applications:

- Tests if the degree distribution alone can explain observed structural properties.

Invariants

- Degree sequence of vertices remains constant.

Randomized Components

- The specific connections (hyperedges) are generated randomly while preserving the degree sequence.

Example

- Given degree sequences for vertices {A: 3, B: 2, C: 2, D: 1}, construct possible hypergraphs. For instance, vertex A needs to be part of 3 hyperedges, vertex B in 2, etc.

Hypergraphs Null Models

HERGM

Extends the exponential random graph models (ERGMs) to hypergraphs.

Models the probability of hypergraph configurations based on observed network statistics.

Parameters:

- Network statistics (e.g., hyperedge size distribution, clustering coefficients).
- Coefficients associated with each statistic.
-

Applications:

- Identifies the importance of different structural features in hypergraphs.

Invariants

- Overall probability distribution parameters are fixed.

Randomized Components

- Specific hyperedges are generated based on probability distributions influenced by the network statistics.

Example

- Modeling the probability of a hypergraph based on hyperedge size and clustering. For example, hyperedges of size 3 might be more likely than size 4.

Hypergraphs Null Models

Bipartite Graph Projections

Treats a hypergraph as a bipartite graph (vertices and hyperedges as separate node sets).

Projects the bipartite graph onto the vertex set, creating a graph where edges represent shared hyperedges.

Steps:

- Create a bipartite graph representation.
- Project onto vertex set to analyze connections.

Applications:

- Useful for comparing to graph-based null models.

Invariants

- The structure of the bipartite graph (vertex-hyperedge incidence) is maintained.

Randomized Components

- The projection itself is deterministic, but random bipartite structures can be analyzed.

Example

- Convert a hypergraph into a bipartite graph and project it onto vertices. For instance, if vertices A, B, and C are part of a hyperedge, they will be connected in the projected graph.

Multiplex Null Models

Random Multiplex

Each possible edge in each layer is included with a fixed probability independently of other edges.

Parameters:

- Number of vertices (n).
- Probability (p) of each edge being included.

Applications:

- Used to test hypotheses about the randomness of observed multiplex networks.

Invariants

- Number of vertices (n) remains constant.
- Number of layers (L) remains constant.

Randomized Components

- Presence or absence of each edge in each layer is determined randomly with probability p.

Example

- Given 4 nodes (A, B, C, D) and 2 layers with p = 0.2, calculate the probability of specific edges existing in each layer. For example, the probability that edge (A, B) exists in layer 1 is 0.2.

Multiplex Null Models

Configuration Model for Multiplex Networks

Maintains the degree distribution (number of edges incident to each node) in each layer of the observed multiplex network.

Randomly rewires edges while preserving the degree sequence in each layer.

Steps:

- List nodes and their degree sequences for each layer.
- Randomly match nodes to form edges in each layer until the degree sequences are met.

Applications:

- Tests if the degree distribution in each layer alone can explain observed structural properties.

Invariants

- Degree sequence of nodes in each layer remains constant.

Randomized Components

- The specific connections (edges) in each layer are generated randomly while preserving the degree sequence.

Example

- Given degree sequences for nodes in two layers: {A: 3, B: 2, C: 2, D: 1} in layer 1 and {A: 2, B: 3, C: 1, D: 2} in layer 2, construct possible multiplex networks.

Multiplex Null Models

Edge Overlap Constraints

Maintains the overall number of edges and degree distribution in each layer, but also controls for the number of overlapping edges (same edges present in multiple layers).

Steps:

- List nodes and their degree sequences for each layer.
- Randomly match nodes to form edges while preserving both degree sequences and the number of overlapping edges.

Applications:

- Tests if the observed number of overlapping edges can be explained by degree distributions.

Invariants

- Degree sequence of nodes in each layer remains constant.
- Number of overlapping edges across layers remains constant.

Randomized Components

- The specific connections (edges) in each layer are generated randomly while preserving degree sequences and overlap constraints.

Example

- Given degree sequences for nodes in two layers and a constraint that 2 edges must overlap between the layers. Construct possible multiplex networks satisfying these conditions.

Feature-rich Null Models

Random Attribute Model

Randomly assign attributes to nodes or edges while keeping the network topology fixed.

Parameters:

- Number of nodes (n), edges (m), and attribute distributions.

Applications:

- Used to test if attribute assignments are random given the observed distribution.

Invariants

- Network topology (structure of the graph) remains constant.
- Attribute distribution (e.g., number of nodes with each attribute value) remains constant.

Randomized Components

- Specific assignment of attributes to nodes or edges.

Example

- Given a network of 4 nodes (A, B, C, D) and node attributes {Red, Blue}, randomly assign colors while preserving the count (e.g., 2 Red and 2 Blue nodes).

Feature-rich Null Models

Degree-Preserving Randomized Attribute

Randomly rewires the network while preserving node degrees and attribute distribution.

Steps:

- List nodes with their degrees and attributes.
- Randomly swap edges while preserving the degree of each node and the attribute distribution.

Applications:

- Tests if the degree distribution and attribute distribution alone can explain observed network properties.

Invariants

- Degree sequence of nodes remains constant.
- Attribute distribution remains constant.

Randomized Components

- Specific connections (edges) between nodes.

Example

- Given a network with specific node degrees and attributes, randomly rewire edges while maintaining node degrees and attributes.

Feature-rich Null Models

Attribute Homophily Null Model

Maintains attribute homophily (tendency of nodes to connect with similar nodes) while randomizing the network.

Steps:

- List nodes with their attributes.
- Rewire edges to preserve the proportion of edges between similar and dissimilar nodes.

Applications:

- Tests if observed attribute homophily levels explain network structure.

Invariants

- Attribute homophily levels remain constant.
- Network topology can change as long as homophily is preserved

Randomized Components

- Specific connections (edges) between nodes.

Example

- Given a network with nodes labeled by color, rewire edges to maintain the same proportion of edges between same-color and different-color nodes.

Feature-rich Null Models

Edge Attribute with Weight Preservation

Maintains the edge weights (or other attributes) while randomizing the network structure.

Steps:

- List edges with their weights.
- Reassign weights to edges randomly while keeping the total weight distribution constant.

Applications:

- Tests if the weight distribution alone can explain observed network properties.

Invariants

- Weight distribution (e.g., sum of weights) remains constant.
- Edge weights remain constant.

Randomized Components

- Specific assignment of weights to edges.

Example

- Given a network with specific edge weights, reassign weights to new edges while preserving the overall weight distribution.

Crafting a Null: where to start?

Define the null model based on the network type and research question:

Choose an appropriate null model that preserves relevant network properties while randomizing others.

What do you need to test?

Which properties have to be maintained?

Question Time

Available data:
I have temporally ordered network snapshots, nodes are OSN users annotated with their stance on a topic. Annotation vary in time.

RQ:
Is there a correlation between user observed stances at T and their social interaction at T-1?

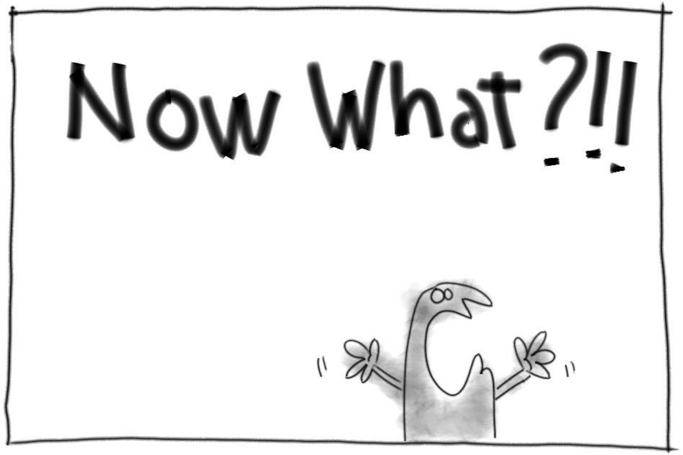
How would you design the null?
Justify your choice!

Testing

Good we have a null.
Now what?

There are several tests that can be applied to perform statistical validation, each with its own requirements and peculiarities.

We will focus on the **z-test**



z-test

Used to determine whether there is a significant difference between sample data and a population parameter, (or between two sample means), when the population variances are known and the sample size is large.

Based on:
z-distribution: N(0,1)
normal distribution with a mean= 0 and a std=1

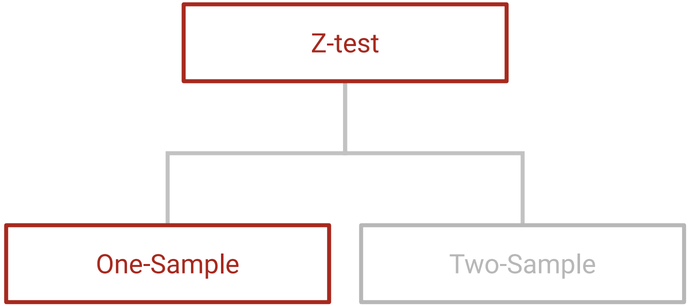
When to perform a Z-test?

Comparing Sample Mean to Population Mean:
You have a single sample and want to compare its mean to the known population mean.

Comparing Two Sample Means:
You have two independent samples and want to compare their means.

Sample Size:
The sample size should be large (typically n > 30).
For smaller samples, a t-test is usually more appropriate unless the population standard deviation is known.

Types and Assumptions



Z-test

One-Sample

Two-Sample

Tests if the mean of a single sample is equal to a known population mean.

Tests if the means of two independent samples are equal.

Assumptions

Normality:
The sample data should be approximately normally distributed, especially important for smaller sample sizes. For large samples, the Central Limit Theorem ensures that the sampling distribution of the sample mean is approximately normal.

Known Population Variance:
The population variance should be known. If it is unknown, a t-test is typically used instead.

Independence:
Observations within each sample should be independent.

Large Sample Size:
Generally, a sample size greater than 30 is considered sufficient for the z-test.

Performing a Z-test

State the Hypotheses

- **Null Hypothesis (H0):**
  Typically states that there is no effect or difference.

- **Alternative Hypothesis (HA):**
  Indicates the presence of an effect or difference.

Choose the Significance Level (α)

- Commonly used values are 0.05, 0.01, or 0.10.

Calculate the Test Statistic:

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

where:

x¯¯ = sample mean

μ = population mean

σ = population standard deviation

n = sample size

Performing a Z-test

Find the critical z-value from the z-table corresponding to the chosen α.

Compare the calculated z-value to the critical z-value:

- **For a two-tailed test:**
  Reject H0 if |z|> critical value.

- **For a one-tailed test:**
  Reject H0 if z > critical value (right-tailed) or
  z < -critical value (left-tailed).

The p-value represents the probability of observing a test statistic at least as extreme as the one calculated, under the null hypothesis.

Compare the p-value to α:

- If p-value ≤α, reject H0.
- If p-value > α, fail to reject H0.

Z-test example

Test if the average height of a sample of 50 students is significantly different from the known population mean height of 170 cm, with a known population standard deviation of 15 cm.

Hypotheses:

- H0: μ=170 cm
- HA; μ!=170 cm

Significance level: α=0.05

Test statistic:

- Suppose x̄=172 cm
- z = (172-170)/(15/sqrt(50)) ≈0.94

Critical value:

- For α=0.05 (two-tailed), critical value z= +/- 1.96

Decision Rule:

- |z| = 0.94 < 1.96, we fail to reject H0

P-Value:

- z=0.94 -> p-value=0.35
- p-value > α, we fail to reject H0

Conclusion:

There is no significant evidence to suggest that the average height of the sample is different from the population mean of 170 cm

When to use
z-score over z-test

z-score:
Statistical measurement describing a value's position relative to the mean of a group of values (in terms of std)

$$z = \frac{X - \mu}{\sigma}$$

When to use:

- **Standardizing Data**
- **Identifying Outliers**
- **Understanding Distribution:**
  To understand the relative position of a data point within a distribution.

Example Z-Score

You have a dataset of students' test scores with a mean score of 70 and a standard deviation of 10. To find how a score of 85 compares to the rest of the class:

z-score = 1.5

This means the score of 85 is 1.5 standard deviations above the mean

Example Z-Test

You want to test if a new teaching method affects students' test scores. The known population mean test score is 70 with a standard deviation of 10.
You have a sample of 50 students taught using the new method with a mean score of 72.

- H0: mean is 70, HA mean is different from 70
- compute the z-test = 1.42,
- compare the critical value to α=0.05

Step 1

Calculate the target feature for the observed network

Determine the network metric (e.g., clustering, assortativity…) value for the observed network

Step 2

Generate the null model instances and calculate the target on them

Create multiple instances (>>30) of the null model and calculate the distribution of the target metric on them

Step 3

Compute the mean and standard deviation on the null

Use them to form the null model distribution.

If the null is well defined we assume the resulting distribution to be normal.

Step 4

Calculate z-scores

Compute the observed metric test to assess their significance

Homework

Design one (or more) null models for your course project.

Describe them in terms of:

- Assumptions made,
- Invariants,
- Target variable

Next lecture (lab) we'll discuss your ideas and start implementing them!

Chapter 8

Conclusion

Take Away Messages

1. Once the analysis is finished, the testing starts
2. Decide what to preserve and what to randomize

Suggested Readings

- The atlas for aspiring network scientists (Ch16)
- Doing Research: A New Researcher’s Guide (Ch4-5)

What’s Next

Chapter 9: Experiments from A to Z