

# Introduction to Applied Statistics for STEM Researchers

Frank Hause

Research Training Group 2467: „Intrinsically Disordered Proteins – Molecular Principles, Cellular Functions, and Diseases“

Winter Semester 2025/26

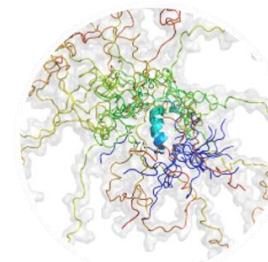


MARTIN-LUTHER-UNIVERSITÄT  
HALLE-WITTENBERG

**RTG 2467**

Intrinsically Disordered Proteins  
– Molecular Principles, Cellular  
Functions, and Diseases

Speaker of the RTG: Prof. Dr. Andrea Sinz  
Scientific Coordinator: Dr. Oleksandr Sorokin



Funded by

**DFG**

Deutsche  
Forschungsgemeinschaft  
German Research Foundation

# General Remarks

---

- Part I and Part II are quite connected, while Part III is to be seen more or less independent.
- In general, video streaming platforms are great to recap certain methods or dive deeper into a topic.
- The seminar series does not require prior knowledge in maths, statistics, or programming.
- Applying the methods from Part III will require at least basic knowledge in any script/programming language that is appropriate for statistics.
- Simple code examples will be supplied at <https://github.com/DataScienceFH>.
- Interaction is welcome.
- We will include short breaks appr. every 1.5 hrs.



# Course Outline

---

Part I: Comparing Entities

Part II: Inferring Causality

Part III: Coping with Complexity

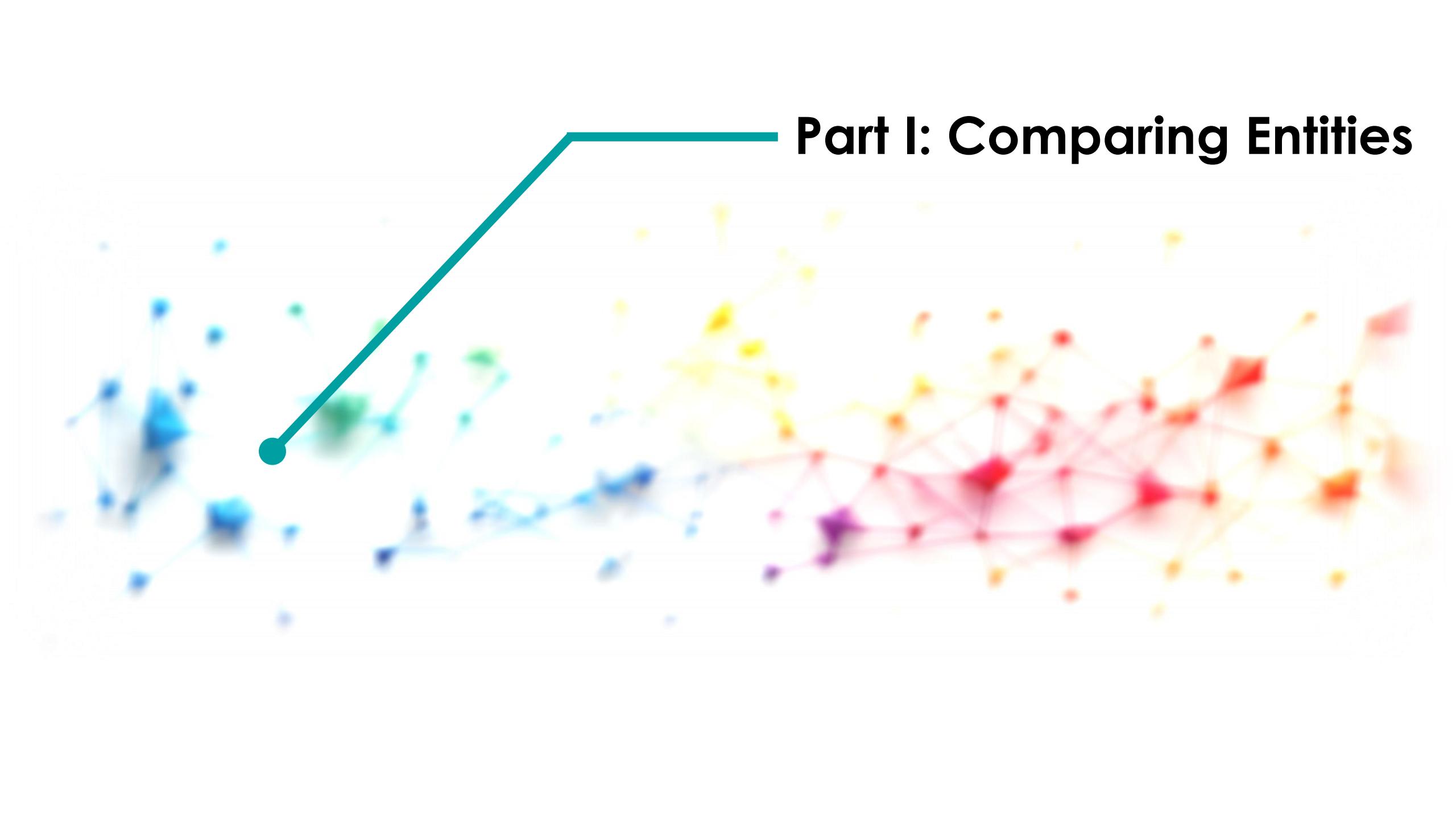
Scripts & Exercises:



frank.hause@medizin.uni-halle.de



@fhause.bsky.social



# Part I: Comparing Entities

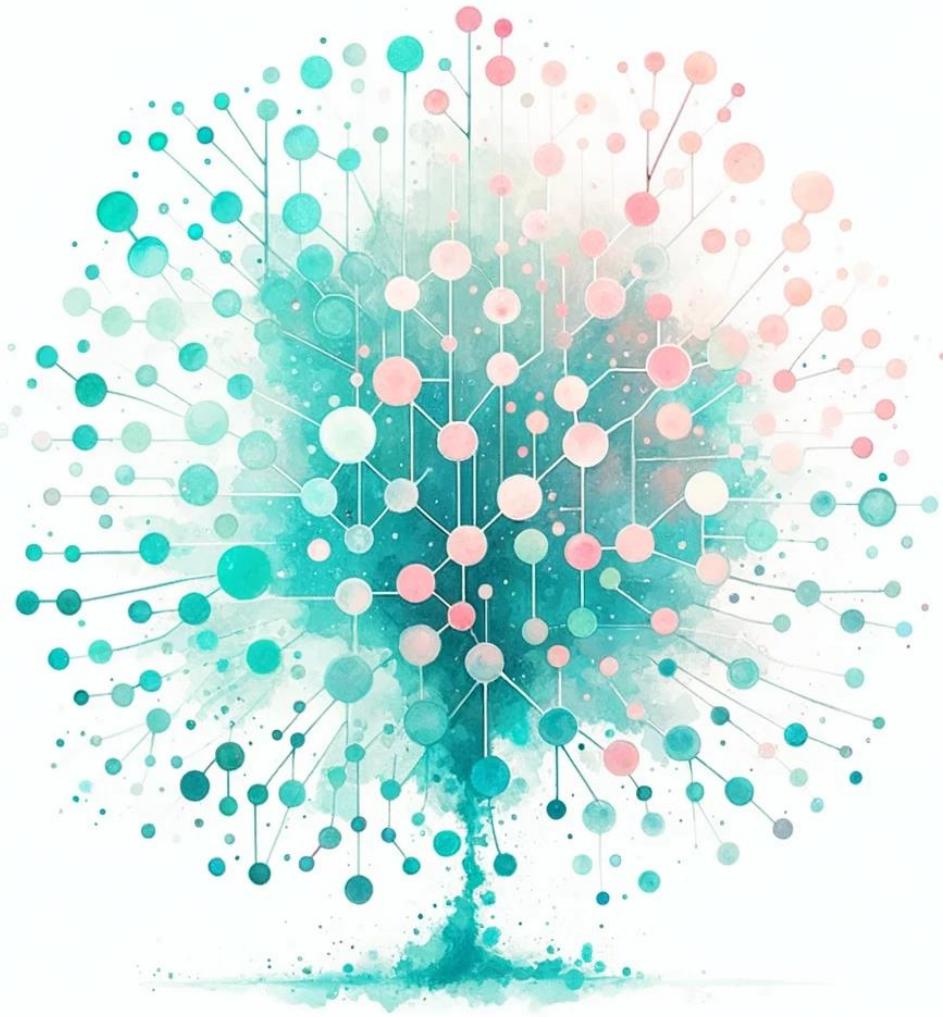
# Part I: Outline

---

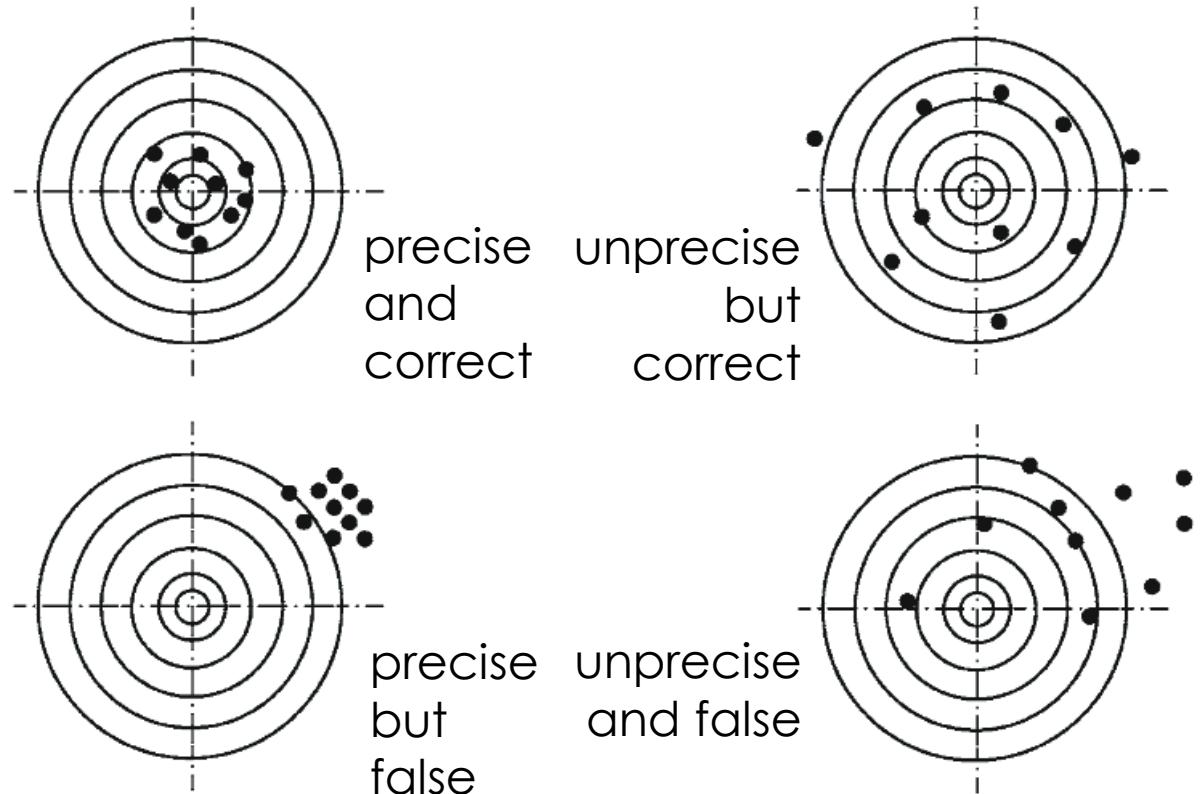
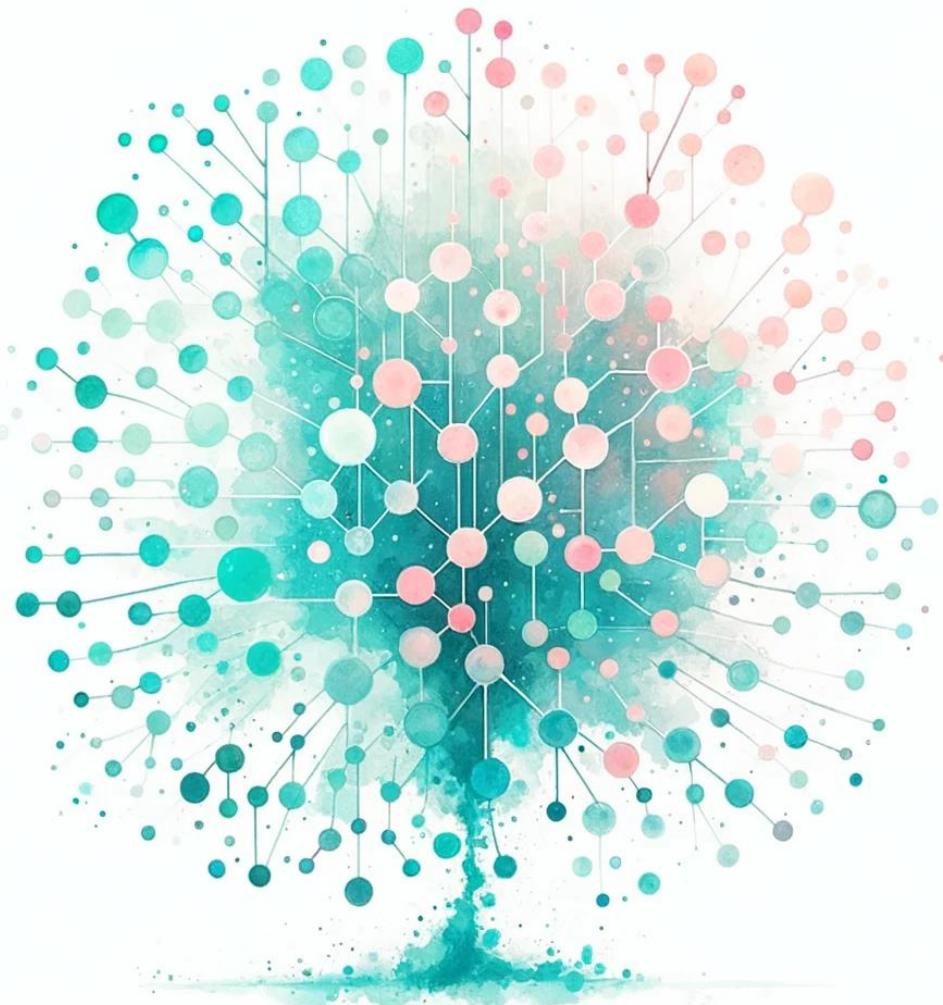
- **Introduction to Statistical Significance**
- **Types of Data**
  - Categorical/Nominal Data
  - Ordinal Data
  - Metric/Scale Data
- **Requirements for Difference Tests**
  - Normal Distribution Tests: Kolmogorov-Smirnov Test, Shapiro-Wilk Test
  - Equality of Variances: Levene Test
- **Finding Differences between Groups**
  - Categorical Data and Small Samples:  $\chi^2$ , Fisher's Exact Test
  - Two-Group Comparisons: Mann-Whitney-U Test, Wilcoxon Test, Student's t-Test
  - Multi-Group Comparisons: Kruskal-Wallis Test, Univariate and Multivariate Analysis of Variance ([M]ANOVA)
- **Excursus:** Identifying Equivalence (TOST)

# Statistical Significance

---



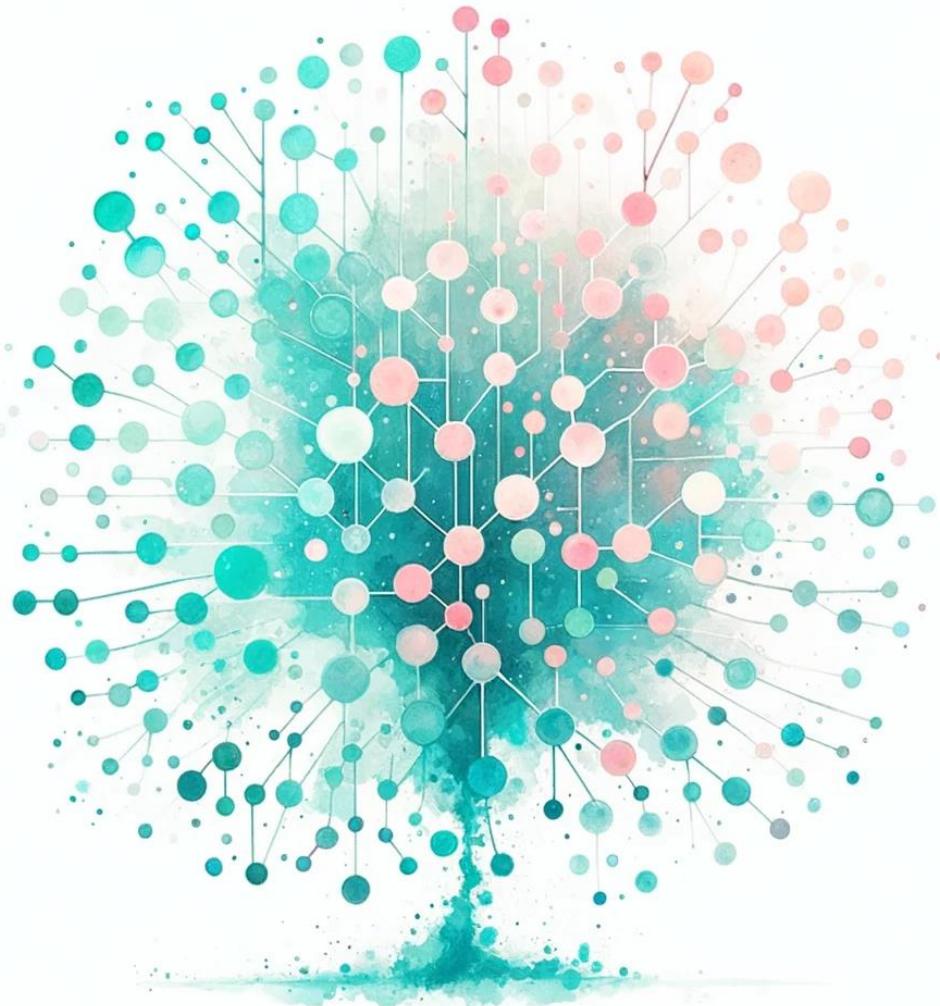
# Statistical Significance



Adapted from Grote et al. (Ed.) (2018): Dubbel: Taschenbuch für den Maschinenbau, 25th ed., p.1970

# Statistical Significance

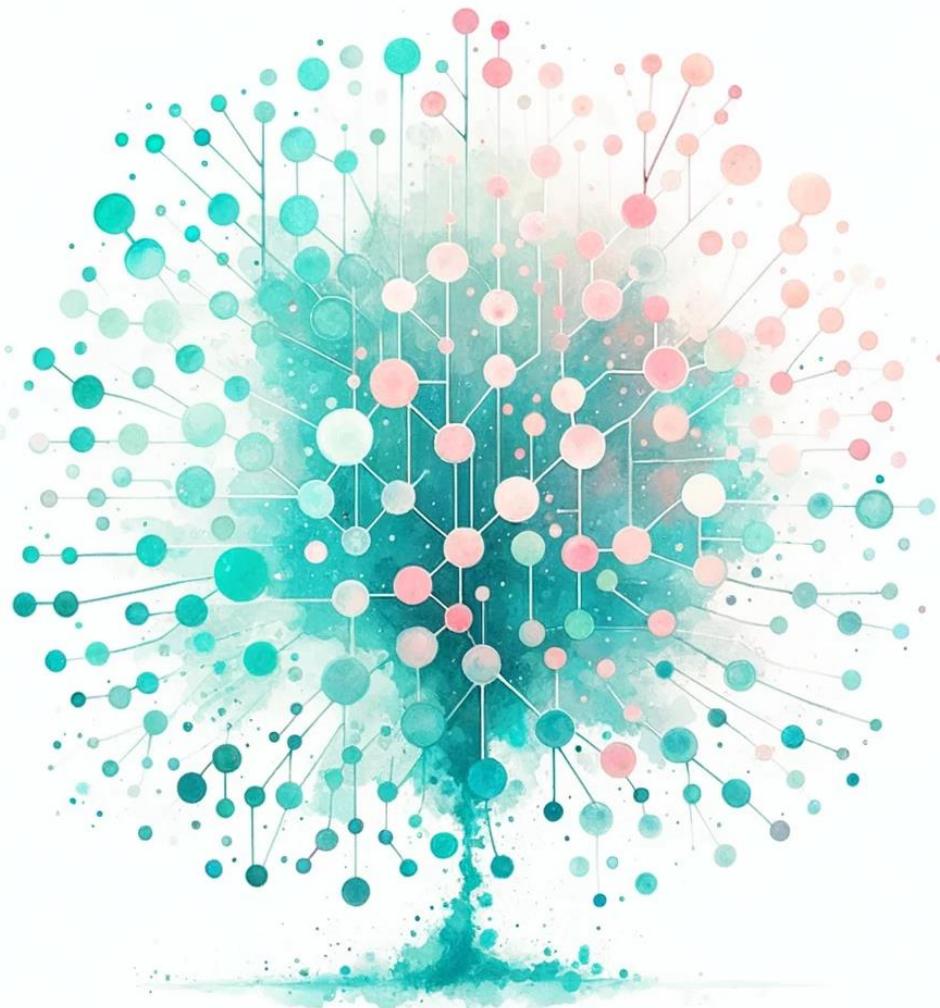
---



- Statistical significance  $\neq$  correctness, effect size, or relevance
  - Popper: falsifiability as core principle of the **scientific method** in empirical science
- **Scientific theories are deemed true as long as they are not falsified.**

# Statistical Significance

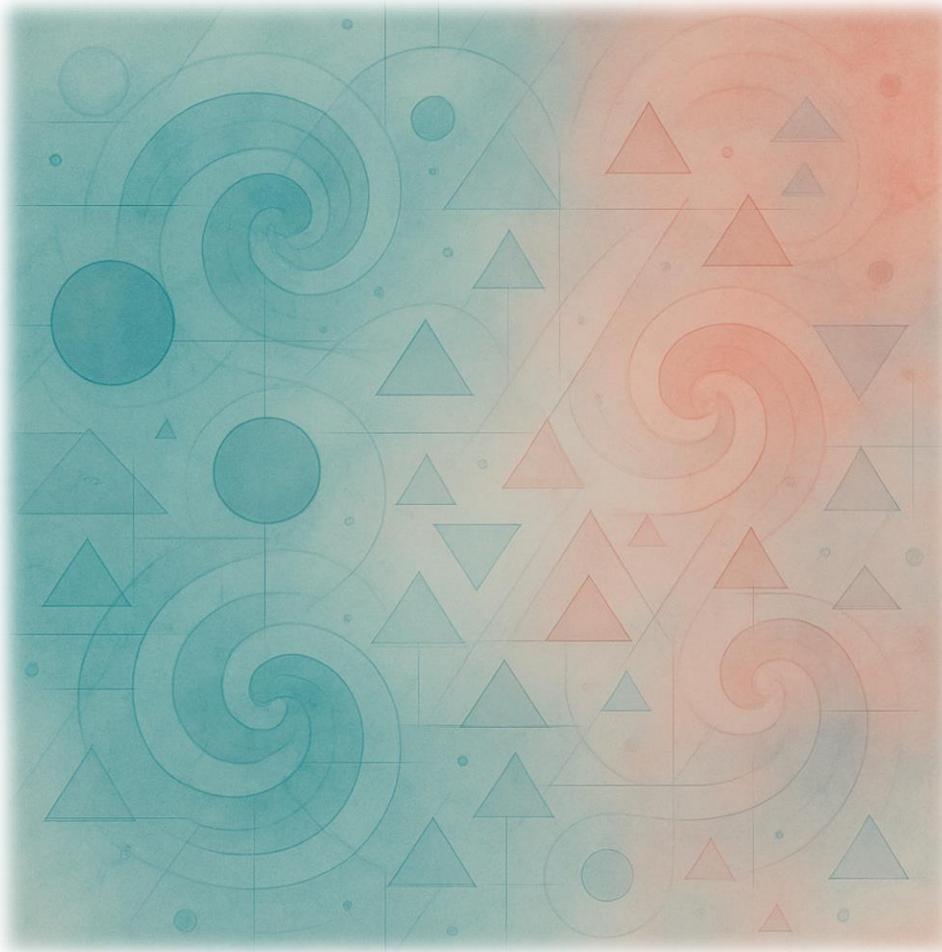
---



- Statistical significance  $\triangleq$  **probability of error** is below a certain threshold  
→ usually:  $p < 0.05$
- Threshold is completely **arbitrary**
- Solely **probabilistic meaning and interpretability**; no strong causal interpretation of p-values
- Application of statistical significance:  
Est. **precision** of measurement  
Deducing **differences**  
Deducing **similarities**

# Types of Data

---



- Nominal/Categorical Data
- Ordinal Data
- Metric/Scale Data

# Types of Data

## Nominal Data

### Cigarette Brands



manifestations of variables can be distinguished



meaningful order of manifestations



distances between manifestations can be determined



### Types of Putative Alien Encounters



# Types of Data

## Ordinal Data

### Cigarette Brand Preference

1.



2.



3.



manifestations of variables can be distinguished



meaningful order of manifestations



distances between manifestations can be determined



### Objectifiability of Alien Encounters

1.



2.



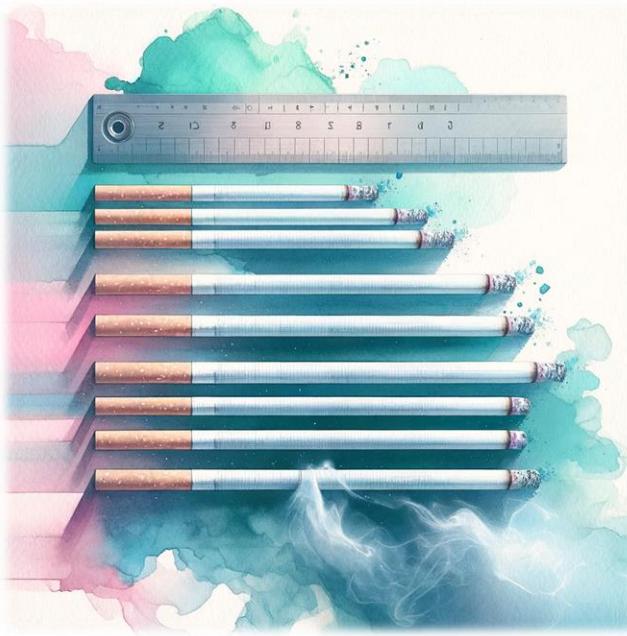
3.



# Types of Data

## Metric Data

### Cigarette Size



manifestations of variables can be distinguished



~50,000,000

meaningful order of manifestations



~ 25

distances between manifestations can be determined



~ 0

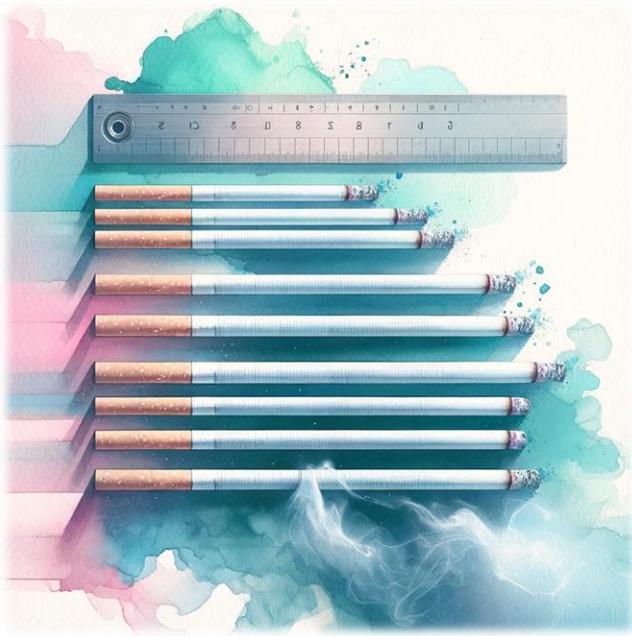
### Casualties of Alien Encounters



# Types of Data

## Metric Data

### Cigarette Size



→ **Continuous:** values are rational (concentration, temperature, etc.)

**Discrete:** values are natural (sample size, number of molecules, etc.) ←

### Casualties of Alien Encounters

~50,000,000

~ 25

~ 0



# Requirements for Statistical Tests

---



- Normal Distribution
- Equality of Variances

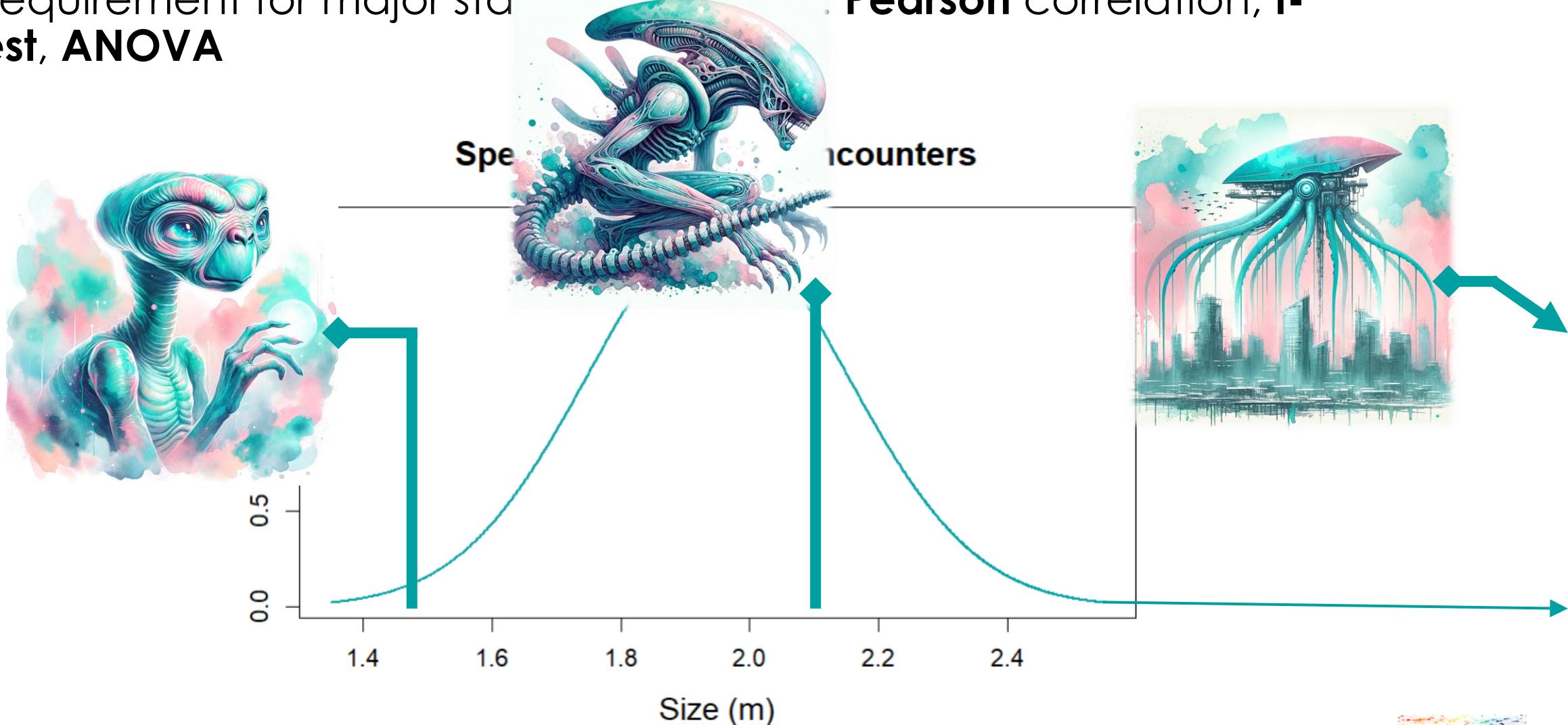
# Requirements for Statistical Tests

---

- increase information content allows to **check various statistical properties** of your data
- requirements can account for a broad number of properties, i.e. **number of samples** per group, certain **distributions**, additional knowledge (**number of groups?**), etc.
- if the test is repeated for various samples, requirements must be met for **each sample individually**
- comparisons between test meeting requirements and ones that do not (Pearson vs. Spearman correlation) are **insufficient**

# Normal Distribution

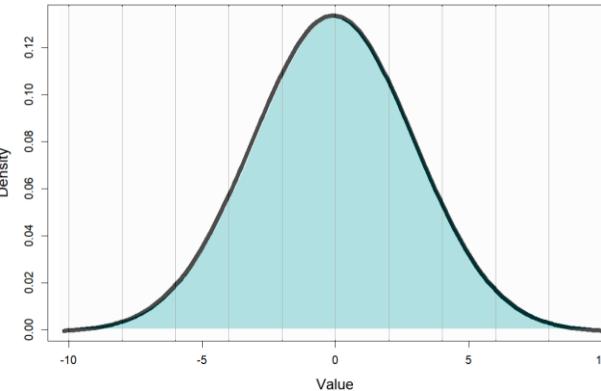
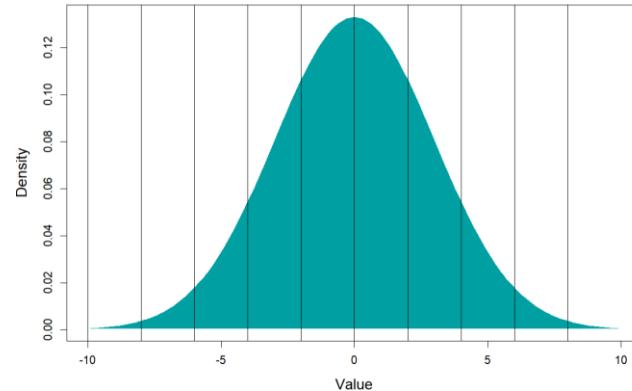
- requirement for major statistical tests, i.e. **Pearson** correlation, **t-Test**, **ANOVA**



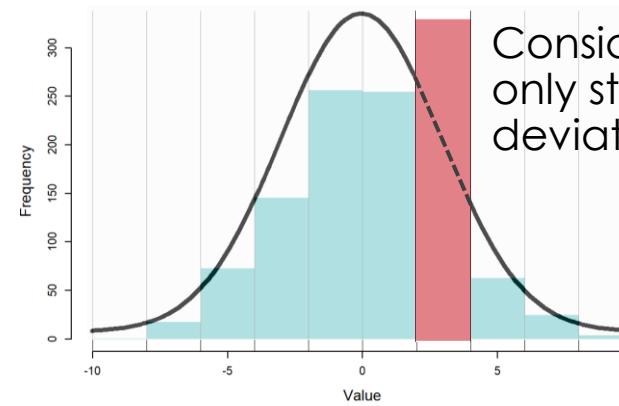
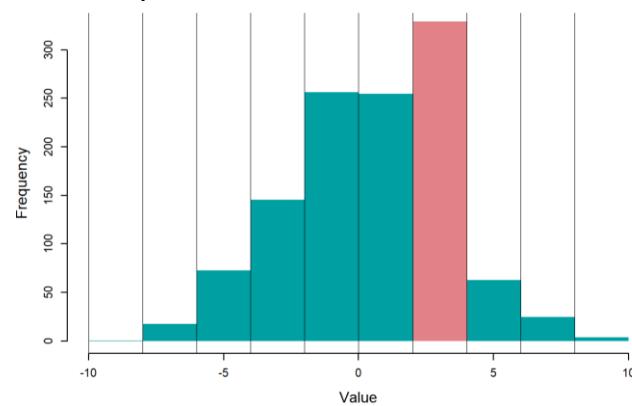
# Normal Distribution

- **Kolmogorov-Smirnov Test:** traditional choice, less sensitive in tails

Ideal Distribution



Empirical Distribution

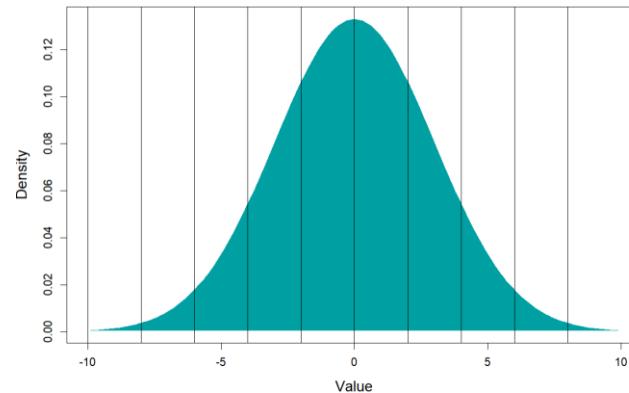


Is deviation  
within  
tolerance  
thresholds?

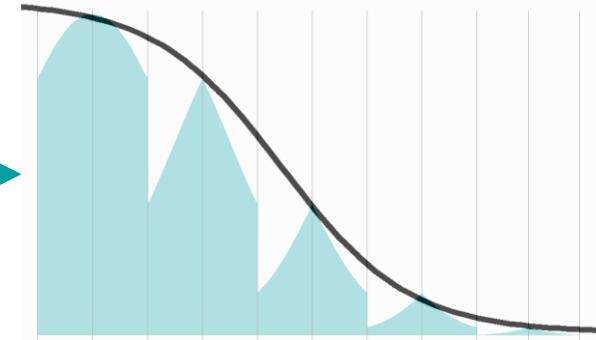
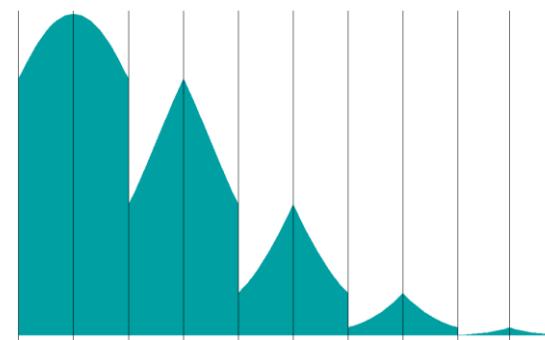
# Normal Distribution

- **Shapiro-Wilk Test:** more common, preferred for small samples

Ideal Distribution

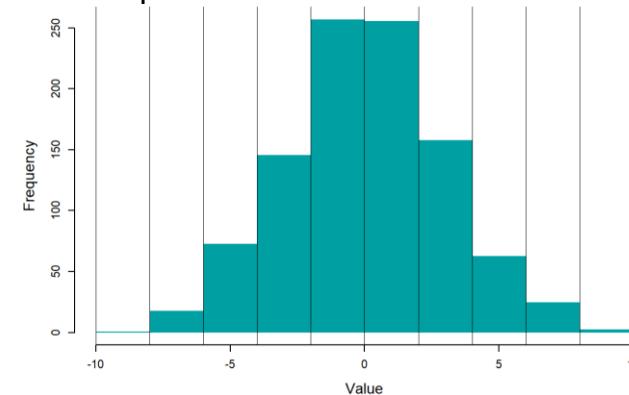


Ranking

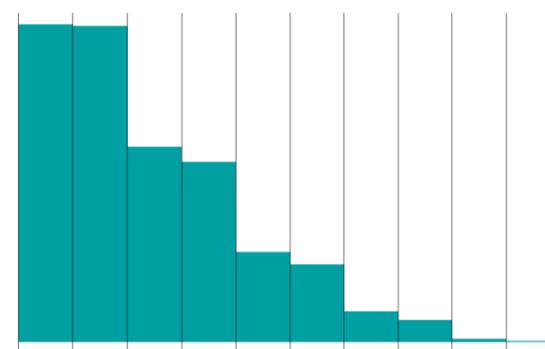


$\approx \hat{s}$

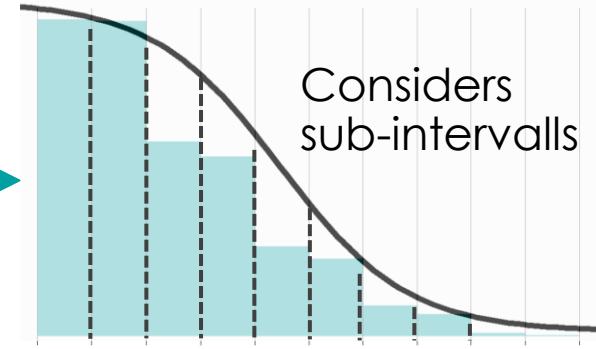
Empirical Distribution



Ranking



Evaluation



Considers  
sub-intervalls

# Normal Distribution

- both tests will provide a **p-value**:

$p \leq 0.05$ : normal distribution **must not be assumed**

$p > 0.05$ : normal distribution **usually assumed**

- if data is **normally distributed**, you may choose **parametric tests**, but this is **not mandatory** (non-parametric tests yield more conservative results)
- if data is **not normally distributed**, **non-parametric tests** are **mandatory**

```
# Kolmogorov-Smirnov Test  
stats::ks.test(your_data, "pnorm")  
  
# Shapiro-Wilk Test  
stats::shapiro.test(your_data)
```

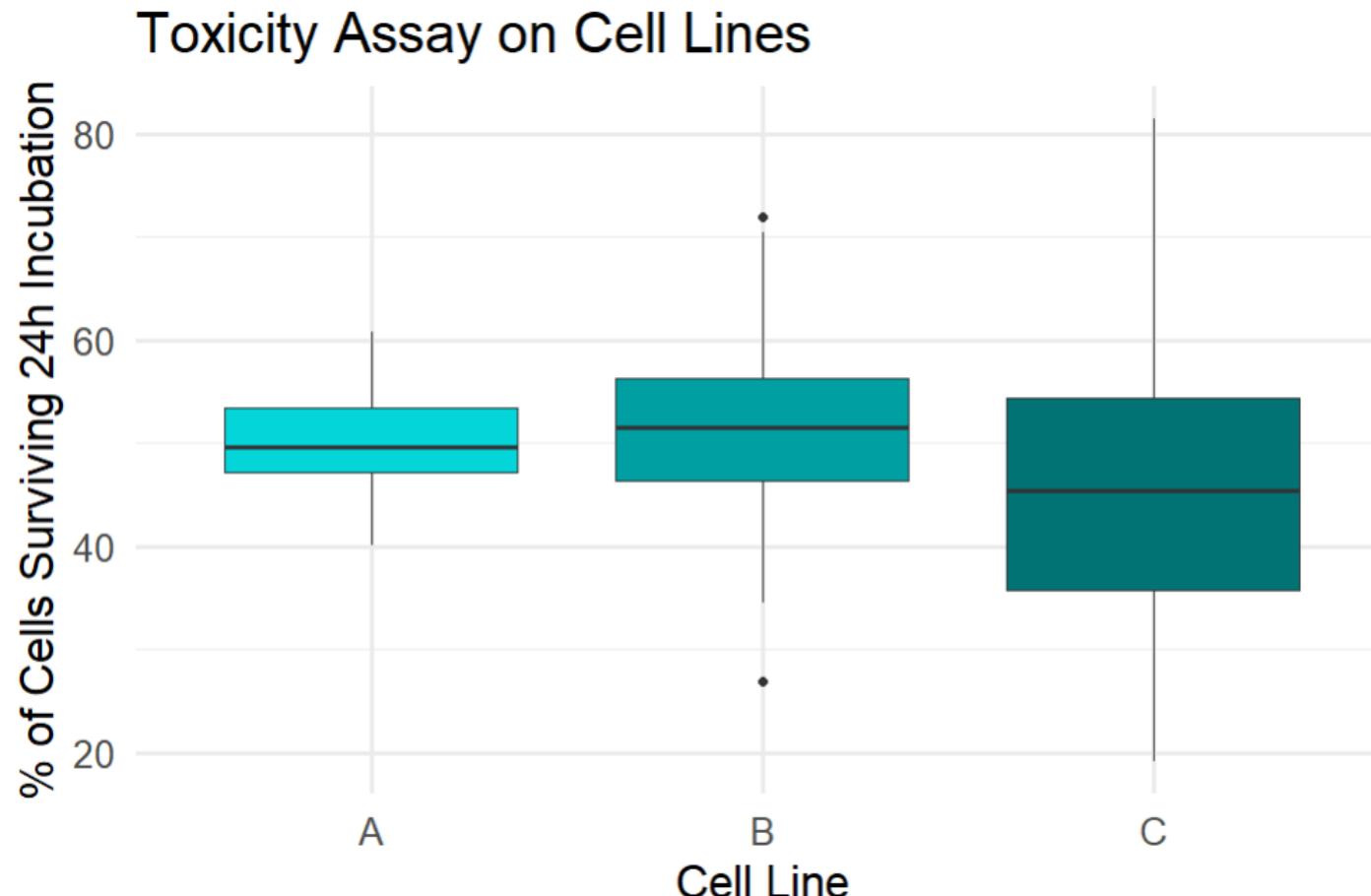


```
import numpy as np  
from scipy import stats  
  
# Kolmogorov-Smirnov test  
ks_statistic, ks_p_value = stats.kstest(your_data, 'norm')  
  
# Shapiro-Wilk test  
shapiro_statistic, shapiro_p_value = stats.shapiro(your_data)
```



# Equality of Variances

- variance as measure for how much values of a set **vary or spread out around the average**
- especially **ANOVA**, but also **t-Test** and **regression** analysis rely on equality of variances

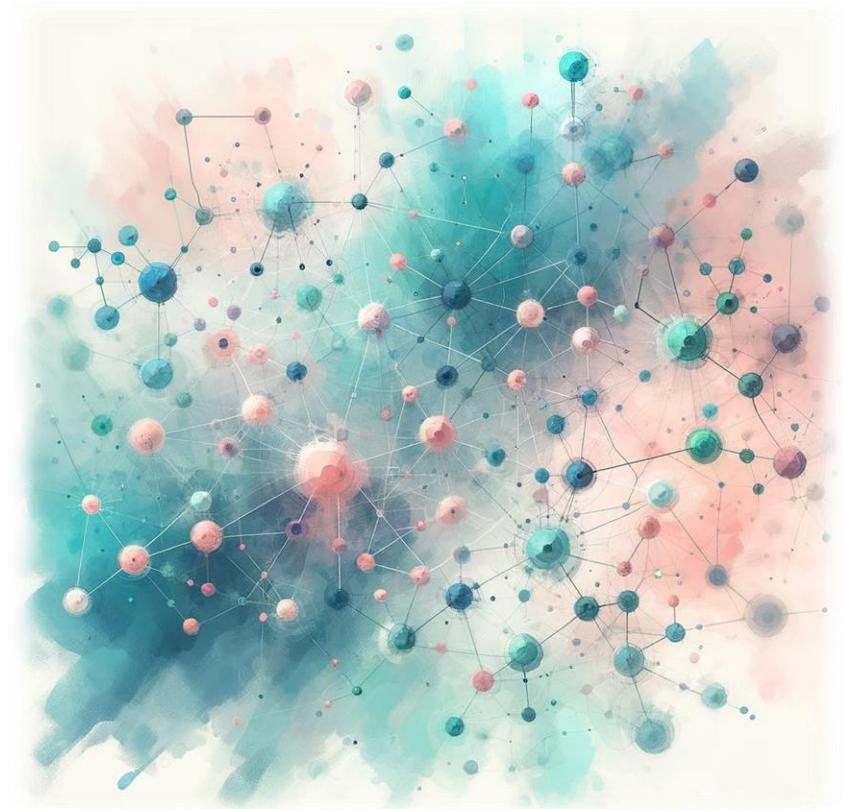


# Equality of Variances

---

If **requirement** of equal variances is **not met**:

- Inflated Type I Error: you are prone to falsely reject the null hypothesis; more likely to **accept non-significant differences as significant**
- Decreased Statistical Power: test **loses discriminative ability**
- Biased Estimates (in regression): influence of **predictors is over-/underestimated**



# Equality of Variances

## Levene's Test:

- estimate differences between variances for each group
- (more or less) independent of a normal distribution
- test will provide a **p-value**:

$p \leq 0.05$ : equal variances **must not be assumed**

$p > 0.05$ : equal variances **usually assumed**

```
library(car)
set.seed(123)

your_group1 <- rnorm(50, mean = 100, sd = 5) # Group 1 data with sd = 5
your_group2 <- rnorm(50, mean = 100, sd = 10) # Group 2 data with sd = 10
your_group3 <- rnorm(50, mean = 100, sd = 15) # Group 3 data with sd = 15

df <- data.frame(
  value = c(your_group1, your_group2, your_group3),
  group = factor(rep(c("your_group1", "your_group2", "your_group3"), each = 50))
)

leveneTest(value ~ group, data = df)
```



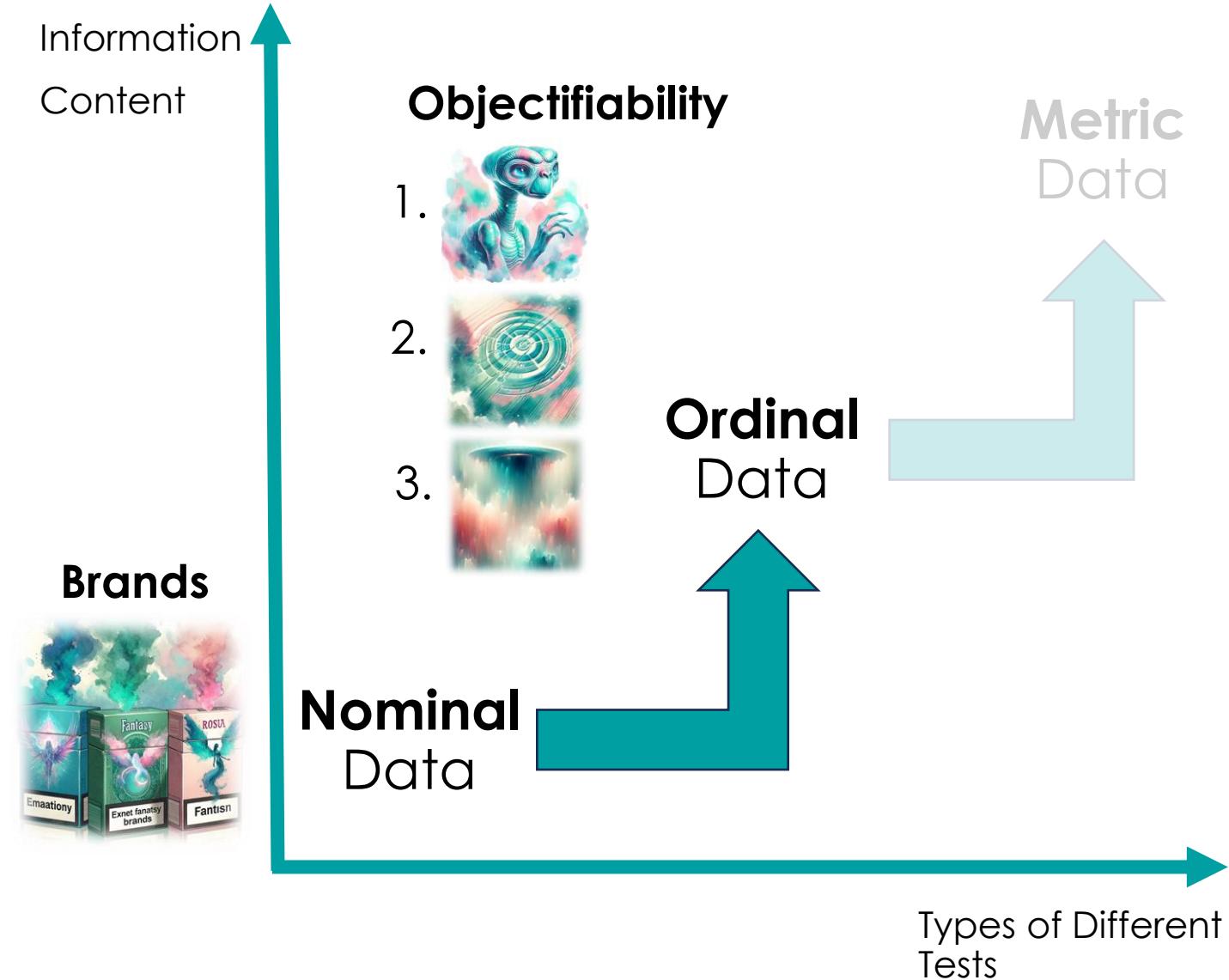
```
import numpy as np
from scipy.stats import levene

np.random.seed(0)
your_group1 = np.random.normal(loc=50, scale=5, size=50) # Mean=50, SD=5
your_group2 = np.random.normal(loc=50, scale=10, size=50) # Mean=50, SD=10
your_group3 = np.random.normal(loc=50, scale=15, size=50) # Mean=50, SD=15

stat, p = levene(your_group1, your_group2, your_group3)
```



# Types of Data



## **Constraints to Data Analysis Approaches:**

- Type of data to be analyzed

## → Tests for nominal/ordinal data

# Testing Non-Metric Data and Small Samples

---



- $\chi^2$  Test
- Fisher's Exact Test

# $\chi^2$ Test for Nominal/Ordinal Data

---

**Research Question:** Is there a statistical connection between formal scientific education and objectifiability of alien encounters?

**Scenario:** Scientists report what type of alien encounter they experienced.

# $\chi^2$ Test for Nominal/Ordinal Data

Variable 1: Objectifiability

In vivo encounter



Physical remnants of alien presents



Footage of alien presents



Variable 2: Formal Education

	B.Sc.	M.Sc.	PhD
In vivo encounter	8	16	24
Physical remnants of alien presents	10	8	18
Footage of alien presents	14	18	12

$$\rightarrow p = 0.0949 \ (\chi^2 = 7.911; df = 4)$$

# $\chi^2$ Test for Nominal/Ordinal Data

## Variable 2: Formal Education

### Variable 1: Objectifiability

In vivo encounter



Physical remnants of alien presents



```
# Define the observed frequencies
observed <- matrix(c(4, 8, 12,
                     5, 4, 9,
                     7, 9, 6),
                     nrow = 3, byrow = TRUE)

# Assign row and column names for clarity (optional)
dimnames(observed) <- list(RowCategory = c("In.vivo", "Phys.remnants", "Footage"),
                           ColCategory = c("B.Sc", "M.Sc", "PhD"))

# Perform the Chi-square test
chi2_result <- chisq.test(observed)
```

B.Sc.

8

10

14

M.Sc.

16

8

PhD

24

18

```
import numpy as np
from scipy.stats import chi2_contingency

# Define the observed frequencies
observed = np.array([[4, 8, 12],
                     [5, 4, 9],
                     [7, 9, 6]])

# Row and Column categories for interpretation (optional and for reference only)
row_categories = ["In vivo", "Phys remnants", "Footage"]
col_categories = ["B.Sc", "M.Sc", "PhD"]

# Perform the Chi-square test
chi2_stat, p, dof, expected = chi2_contingency(observed)
```



# Fisher's Exact Test for Small Samples

**Variable 1: Objectifiability**

In vivo encounter



Physical remnants of alien presents



Footage of alien presents



**Variable 2: Formal Education**

	B.Sc.	M.Sc.	PhD
In vivo encounter	8	16	24
Physical remnants of alien presents	10	8	18
Footage of alien presents	14	18	12

# Fisher's Exact Test for Small Samples

Variable 1: Objectifiability

In vivo encounter



Physical remnants of alien presents



Footage of alien presents



Variable 2: Formal Education

B.Sc.

M.Sc.

PhD

	4	8	12
	5	4	9
	7	9	6

$$\rightarrow p = 0.412 \ (x^2 = 3.956; df = 4)$$

# Fisher's Exact Test for Small Samples

## Variable 1: Objectifiability

In vivo encounter



Physical remnants of alien presents



Footage of alien presents



```
# Example data  
data <- matrix(c(4, 8, 5, 4), nrow = 2, byrow = TRUE)  
  
# Perform Fisher's exact test  
result <- fisher.test(data)
```

## Variable 2: Formal Education

B.Sc.

M.Sc.

PhD

	B.Sc.	M.Sc.	PhD
In vivo encounter	4	8	12
Physical remnants of alien presents	5	4	9
Footage of alien presents	7		

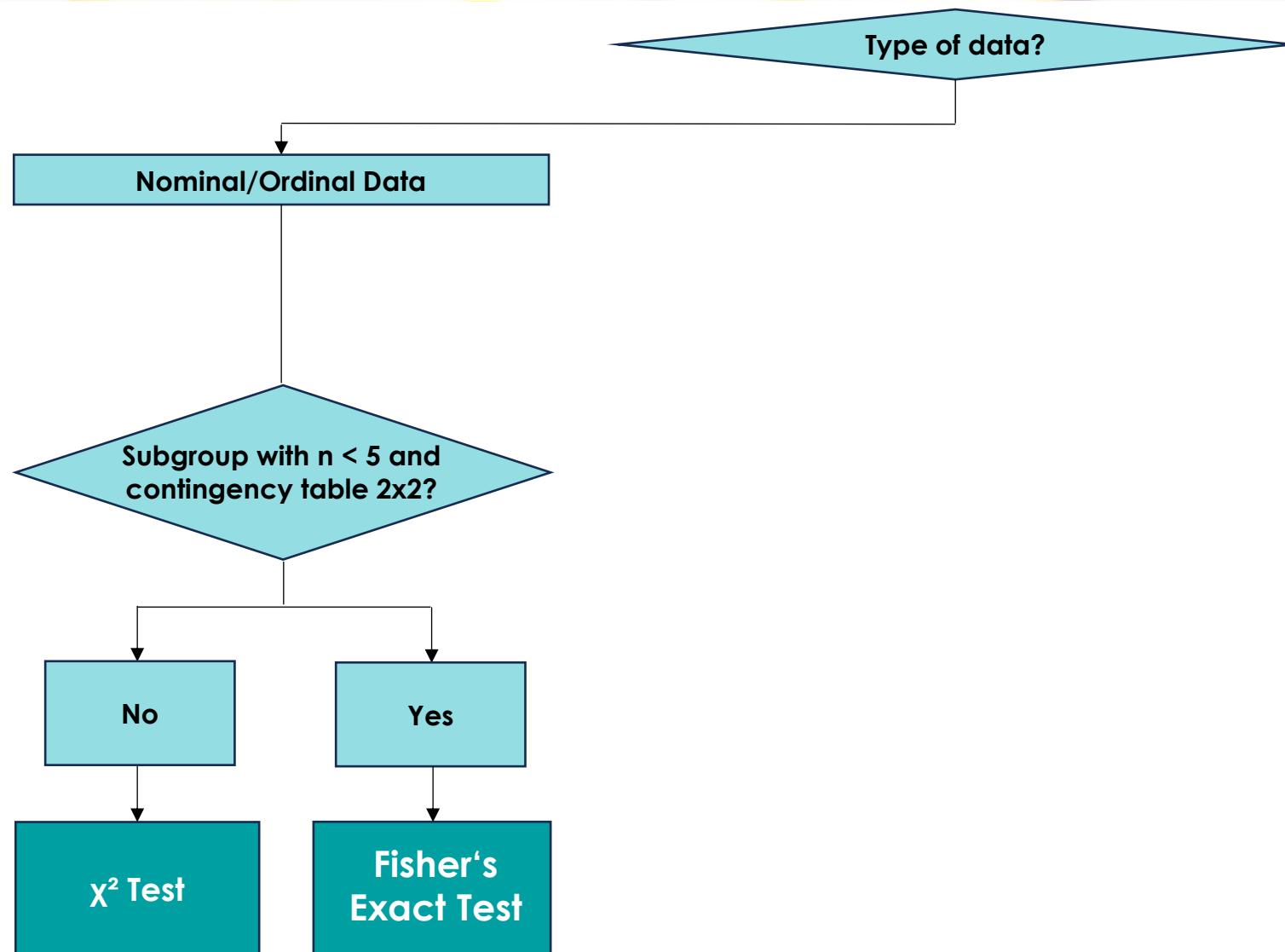
```
from scipy.stats import fisher_exact
```

```
# Example data  
data = [[4, 8], [5, 4]]
```

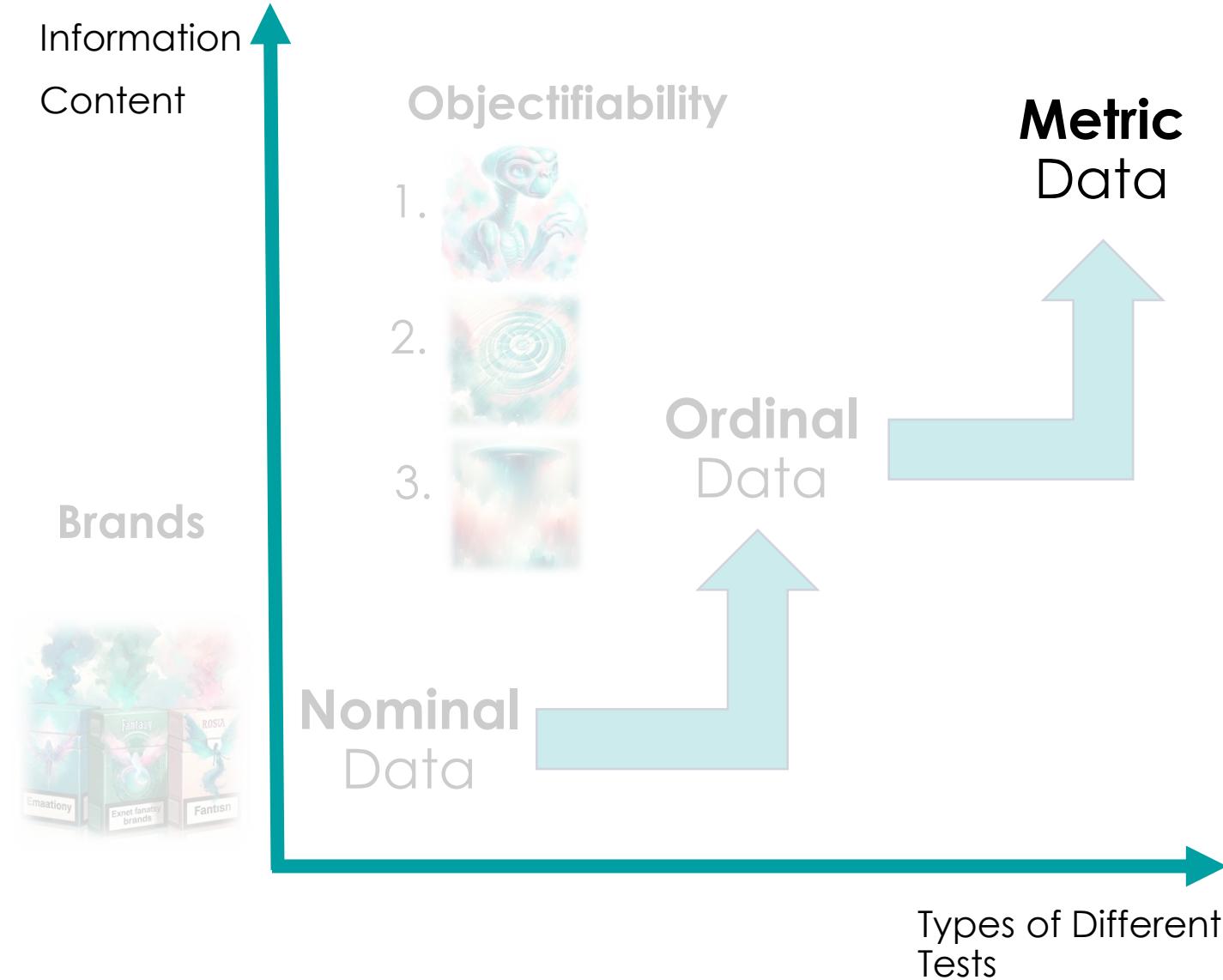
```
# Perform Fisher's exact test  
odds_ratio, p_value = fisher_exact(data)
```



# Choosing the Right Test for Comparison

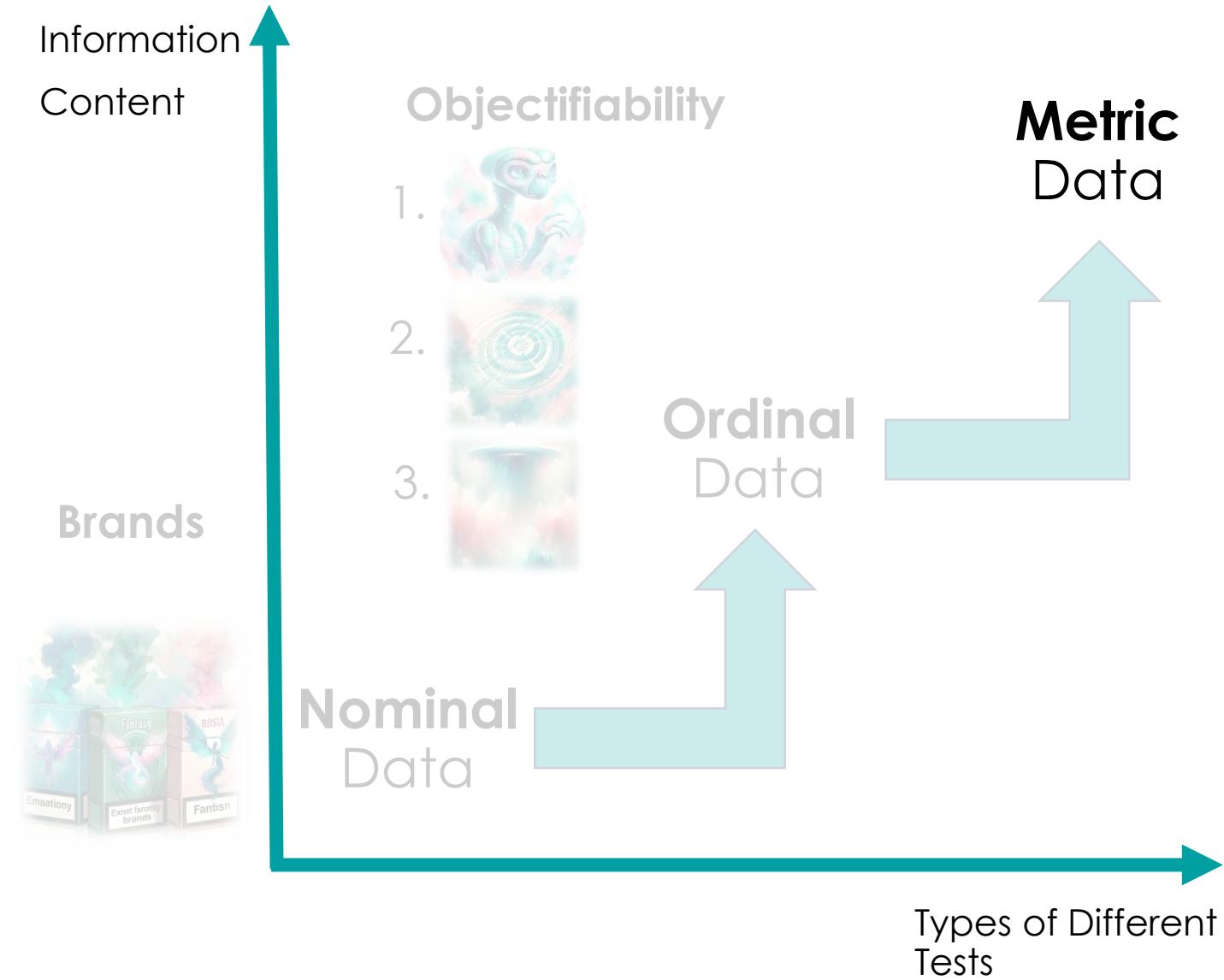


# Types of Data



- highest information content
- can be discrete (number of molecules, patients, bacteria number, etc.) or continuous (concentrations, temperatures, radiation dose, etc.)
- sometimes there is debate over how many levels are needed for ordinal data to transition into metric data: ten levels are assumed safe ground

# Types of Data



$H^0$ : There is no significant difference between all compared groups.

$H^a$ : At least two groups differ significantly from each other.

- Each group comparison will yield a **p value**:

$p \leq 0.05$ : reject  $H^0$  & accept  $H^a \rightarrow$  a **significant difference** is assumed

$p > 0.05$ : accept  $H^0$  & reject  $H^a \rightarrow$  **no significant difference** is assumed

# Comparing Two Groups

---



- Mann-Whitney-U Test
- Wilcoxon Signed Rank Test
- Student's t Test

# Mann-Whitney-U Test

- rank-based → **non-parametric**
- for **independent samples**

## Examples:

- Comparing Gene Expression Levels Between Two Groups
- Evaluating the Effect of One Treatment on Plant Growth
- Assessing Behavioral Responses of Two Animal Groups

```
# Example data
your_group1 <- c(5, 8, 7, 6, 9, 5)
your_group2 <- c(7, 7, 6, 8, 10, 9)

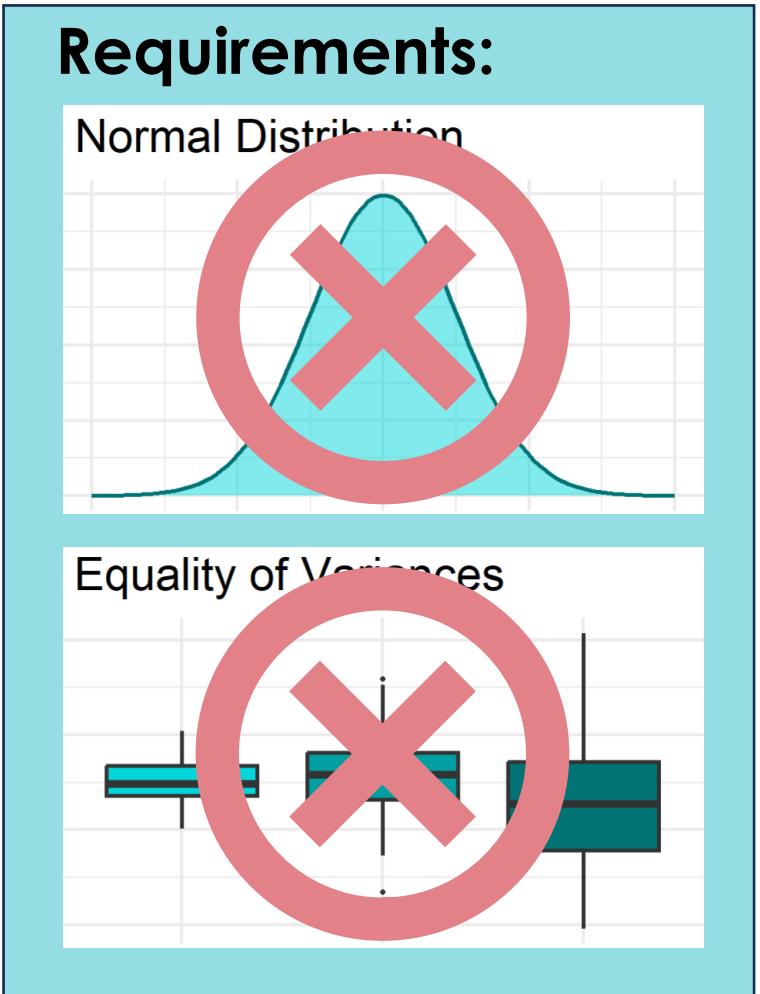
# Perform Mann-Whitney U test
test_result <- wilcox.test(your_group1, your_group2)
```



```
import numpy as np
from scipy.stats import mannwhitneyu

# Example data
your_group1 = np.array([5, 8, 7, 6, 9, 5])
your_group2 = np.array([7, 7, 6, 8, 10, 9])

# Perform Mann-Whitney U test
stat, p = mannwhitneyu(your_group1, your_group2)
```



# Wilcoxon Signed Rank Test

- rank-based → **non-parametric**
- for **paired/dependent samples**
- **CAUTION!**: MWU is often termed Wilcoxon Rank-Sum Test or just Wilcoxon Test

## Examples:

- Comparing Gene Expression Levels in One Group before and after Treatment
- Dietary Intervention Studies in Nutrition Science
- Behavioral Studies in Environmental Biology like stressors during hibernation

```
# Example data
your_group1 <- c(5, 8, 7, 6, 9, 5)
your_group2 <- c(7, 7, 6, 8, 10, 9)

# Perform Wilcoxon Signed Rank test
test_result <- wilcox.test(your_group1, your_group2, paired = TRUE)
```



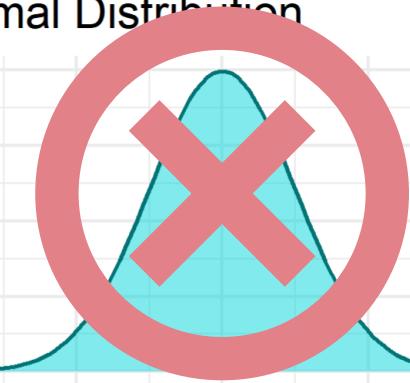
```
import numpy as np
from scipy.stats import wilcoxon

# Example data
your_group1 = np.array([5, 8, 7, 6, 9, 5])
your_group2 = np.array([7, 7, 6, 8, 10, 9])

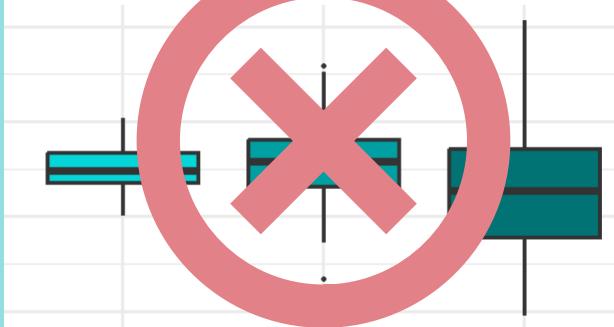
# Perform Wilcoxon Signed Rank test
stat, p = wilcoxon(your_group1, your_group2)
```

## Requirements:

Normal Distribution



Equality of Variances



# Student's t-Test (Independent Samples)

- **parametric**: require normally distributed data
- for **independent samples**

## Examples:

- Comparing Drug Efficacies
- Effect of Fertilizers on Plant Growth
- Genetic Trait Differences Between Populations

```
# Example data  
your_group1 <- c(5, 8, 7, 6, 9, 5)  
your_group2 <- c(7, 7, 6, 8, 10, 9)  
  
# Perform Student's t-test for independent samples  
test_result <- t.test(your_group1, your_group2)
```

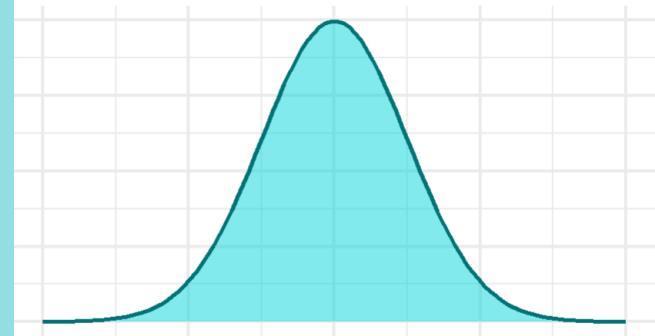


```
import numpy as np  
from scipy.stats import ttest_ind  
  
# Example data  
your_group1 = np.array([5, 8, 7, 6, 9, 5])  
your_group2 = np.array([7, 7, 6, 8, 10, 9])  
  
# Perform Student's t-test for independent samples  
stat, p = ttest_ind(your_group1, your_group2)
```

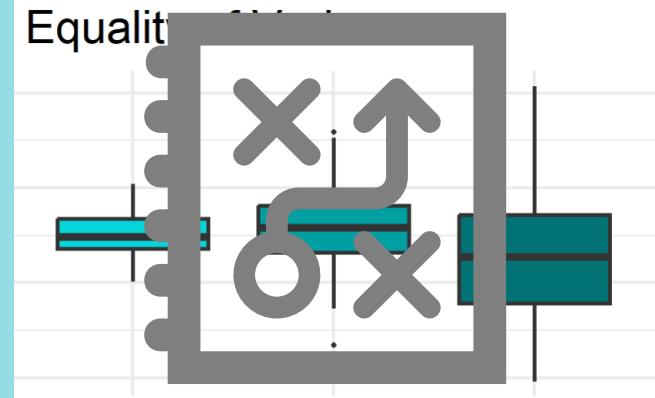


## Requirements:

### Normal Distribution



### Equality of Variances



# Student's t-Test (Paired Samples)

- comparison of **two groups**
- **parametric**: require normally distributed data
- for **paired/dependent samples**

## Examples:

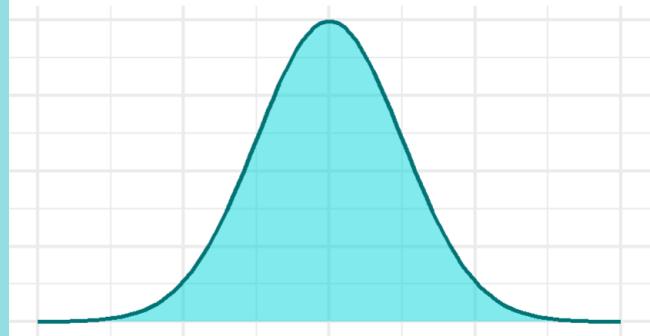
- Pre- and Post-Treatment Analysis in Clinical Trials
- Sleep Study on Caffeine Consumption
- Behavioral Changes in Animals Exposed to Different Light Cycles

```
R logo  
# Example data  
your_group1 <- c(5, 8, 7, 6, 9, 5)  
your_group2 <- c(7, 7, 6, 8, 10, 9)  
  
# Perform paired sample t-test  
test_result <- t.test(your_group1, your_group2, paired = TRUE)
```

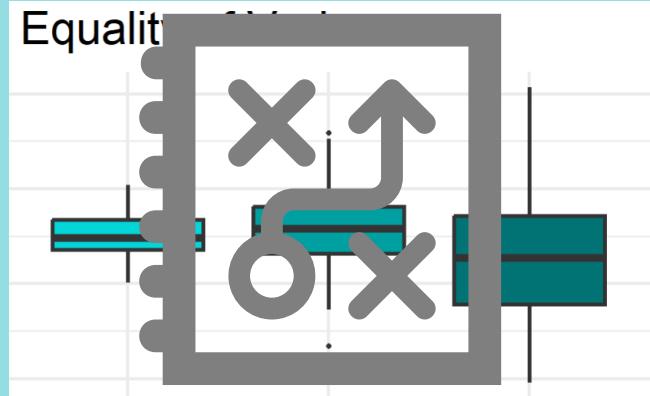
```
Python logo  
import numpy as np  
from scipy.stats import ttest_rel  
  
# Example data  
your_group1 = np.array([5, 8, 7, 6, 9, 5])  
your_group2 = np.array([7, 7, 6, 8, 10, 9])  
  
# Perform Student's t-test for paired samples  
stat, p = ttest_rel(your_group1, your_group2)
```

## Requirements:

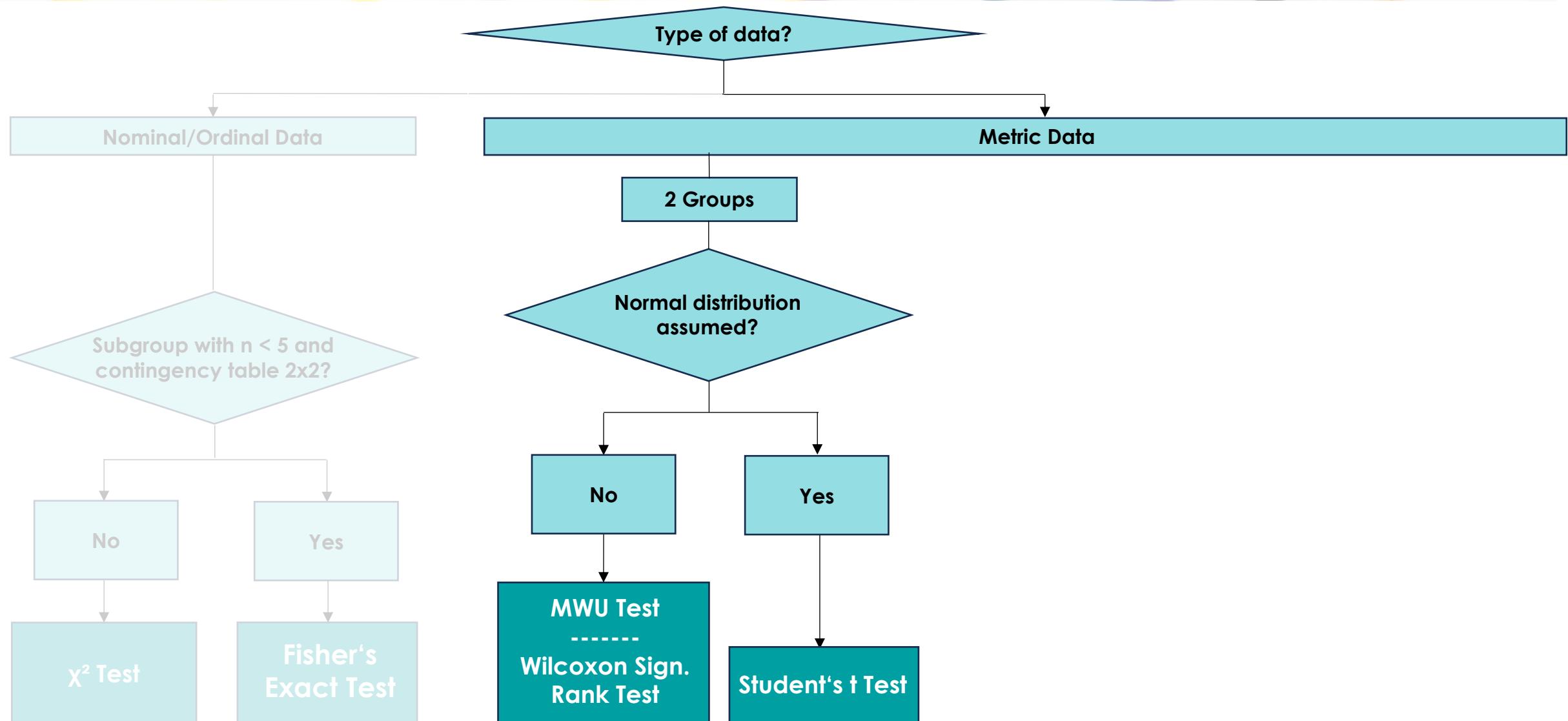
Normal Distribution



Equality



# Choosing the Right Test for Comparison



# Comparing Three or More Groups

---



- Kruskal-Wallis Test
- Welch Test
- (Multivariate) Analysis of Variance

# Kruskal-Wallis Test

- non-parametric

## Examples:

- Comparing Weight Loss Effects of Five Different Diets
- Responses to Environmental Stressors of Plants Species on a Meadow
- Antibiotic Effect of Drugs on Three Strains of *E. coli*



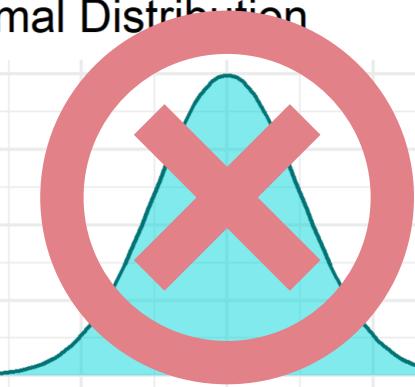
```
# Example data for three groups
your_group1 <- c(20, 21, 19, 20, 22)
your_group2 <- c(25, 28, 22, 24, 26)
your_group3 <- c(30, 29, 31, 32, 33)

# Combine the data into a single vector and create a grouping variable
data <- c(your_group1, your_group2, your_group3)
group <- factor(rep(c("Group 1", "Group 2", "Group 3"), each=5))

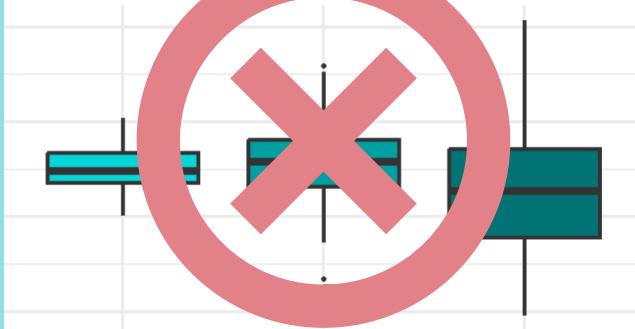
# Perform Kruskal-Wallis test
test_result <- kruskal.test(data ~ group)
```

## Requirements:

Normal Distribution



Equality of Variances



# Welch Test

- parametric
- also termed Welch ANOVA

## Examples:

- Comparing Plant Growth Rates Across Different Soil Types
- Evaluating the Efficacy of a New Medication Across Multiple Age Groups
- Assessing Protein Expression Levels in Different Tissue Types

```
# Example data for three groups
your_group1 <- c(20, 21, 19, 20, 22)
your_group2 <- c(25, 28, 22, 24, 26)
your_group3 <- c(30, 29, 31, 32, 33)

# Combine the data into a single vector and create a grouping variable
data <- c(your_group1, your_group2, your_group3)
group <- factor(rep(c("Group 1", "Group 2", "Group 3"), each=5))

# Perform Welch ANOVA
test_result <- oneway.test(data ~ group)
```



```
import pandas as pd
import pingouin as pg

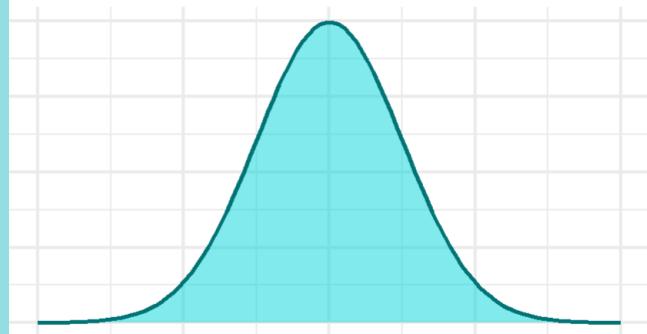
# Example data for three groups
df = pd.DataFrame({
    "value": [18, 21, 16, 22, 19, 28, 31, 29, 30, 27, 38, 35, 39, 40, 36],
    "group": ["Group 1"]*5 + ["Group 2"]*5 + ["Group 3"]*5
})

# Perform Welch ANOVA
test_result = pg.welch_anova(data=df, dv='value', between='group')
```

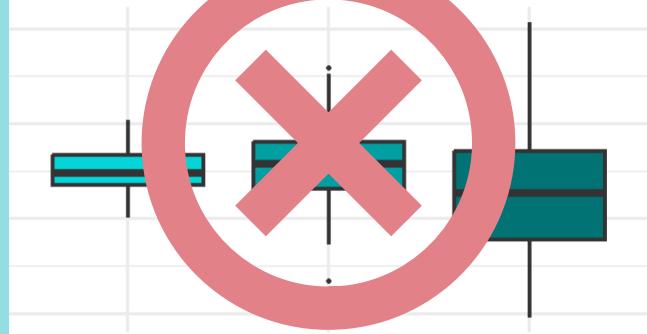


## Requirements:

Normal Distribution



Equality of Variances



# Analysis of Variance (ANOVA)

- parametric
- depending on number of moderators:  
univariate/multivariate

## Examples:

- Comparing Plant Growth Rates Across Different Soil Types
- Evaluating the Efficacy of a New Medication Across Multiple Age Groups
- Assessing Protein Expression Levels in Different Tissue Types

```
# Example data for three groups
your_group1 <- c(5, 8, 7, 6, 5)
your_group2 <- c(6, 7, 10, 9, 8)
your_group3 <- c(7, 11, 9, 12, 10)

# Combine the data into a single vector and create a grouping variable
data <- c(your_group1, your_group2, your_group3)
group <- factor(rep(c("Group 1", "Group 2", "Group 3"), each = 5))

# Perform ANOVA
result <- aov(data ~ group)
summary(result)
```



```
from scipy.stats import f_oneway

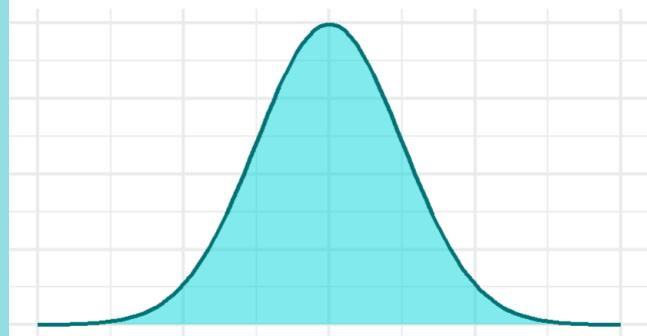
# Example data for three groups
your_group1 = [5, 8, 7, 6, 5]
your_group2 = [6, 7, 10, 9, 8]
your_group3 = [7, 11, 9, 12, 10]

# Perform ANOVA
F_statistic, p_value = f_oneway(your_group1, your_group2, your_group3)
```

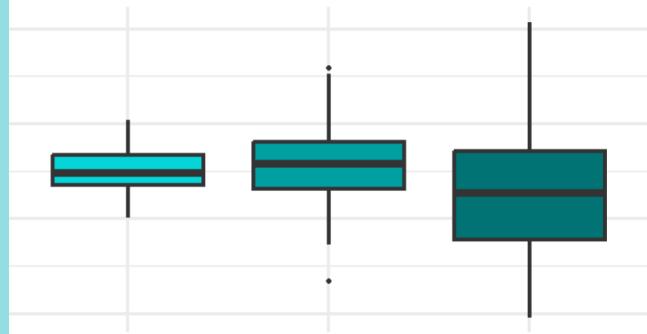


## Requirements:

Normal Distribution



Equality of Variances



# Analysis of Variance (ANOVA)

- **Post-hoc Tests:** ANOVA itself only tells whether there is **at least one significant difference** between **at least two groups** → but not which these groups are → post-hoc tests identify the significantly differing groups

```
# Perform ANOVA  
result <- aov(data ~ group)  
summary(result)
```

```
# Perform Tukey HSD post hoc test  
tukey_test <- TukeyHSD(result)  
print(tukey_test)
```

```
> print(tukey_test)  
Tukey multiple comparisons of means  
95% family-wise confidence level  
  
Fit: aov(formula = data ~ group)  
  
$group  
      diff      lwr      upr   p adj  
Group 2-Group 1 1.8 -0.9380826 4.538083 0.2262717  
Group 3-Group 1 3.6  0.8619174 6.338083 0.0111427  
Group 3-Group 2 1.8 -0.9380826 4.538083 0.2262717
```



```
# Perform Tukey's HSD test  
tukey = pairwise_tukeyhsd(endog=data, groups=groups, alpha=0.05)
```

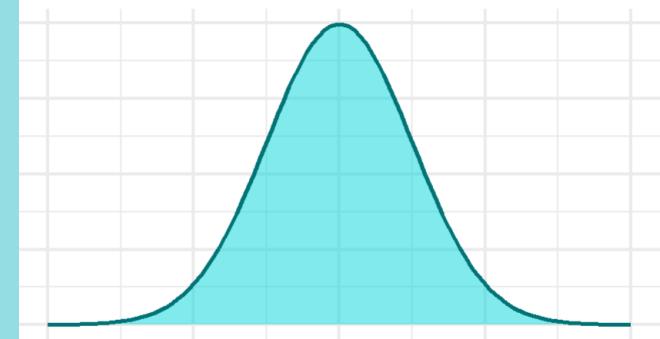
```
# Print the results  
print(tukey)
```



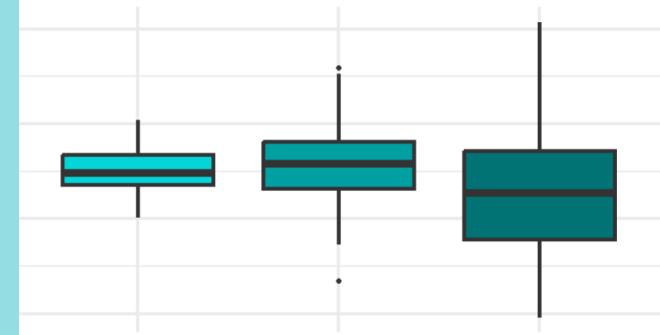
```
Multiple Comparison of Means - Tukey HSD, FWER=0.05  
=====  
group1 group2 meandiff p-adj    lower   upper  reject  
-----  
group1 group2    1.8 0.2263 -0.9381 4.5381 False  
group1 group3    3.6 0.0111  0.8619 6.3381  True  
group2 group3    1.8 0.2263 -0.9381 4.5381 False  
-----
```

## Requirements:

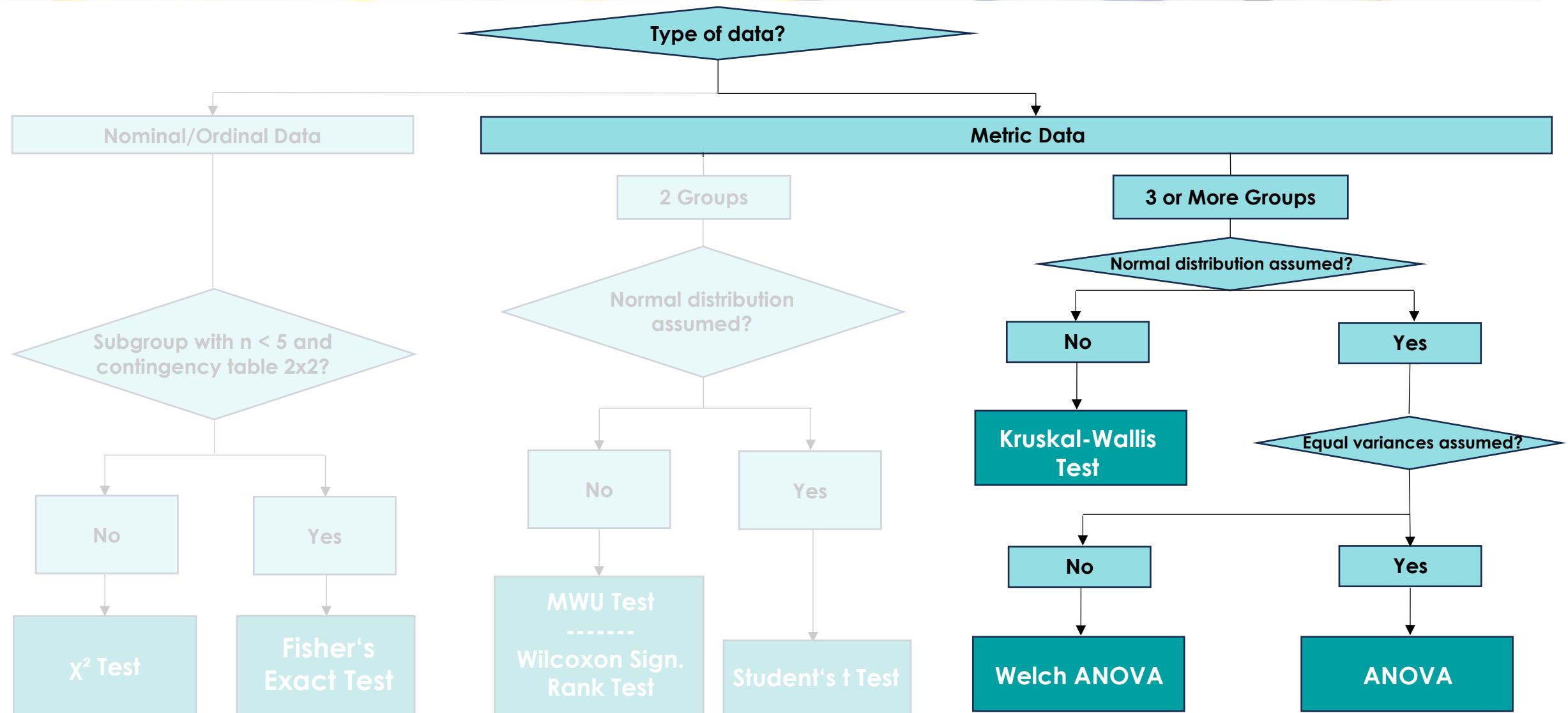
Normal Distribution



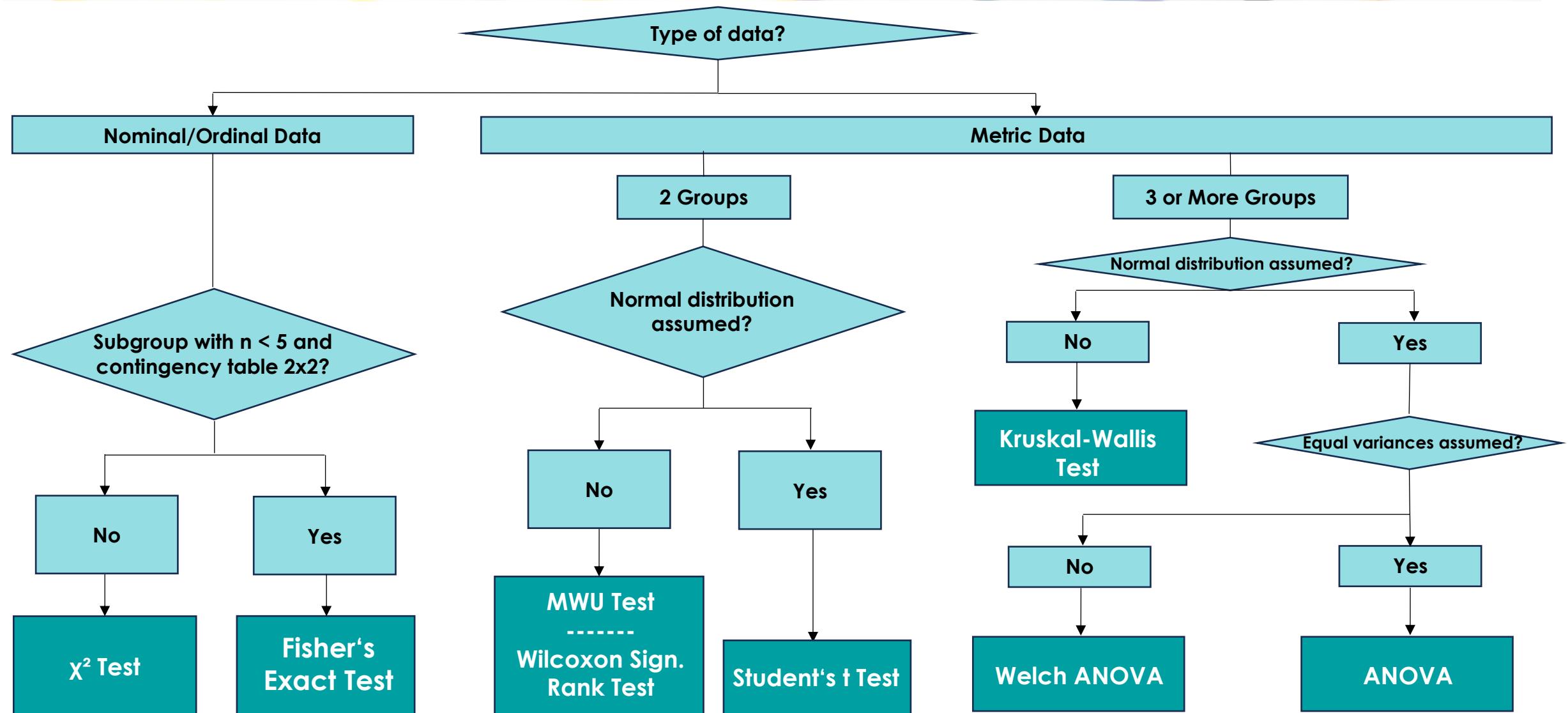
Equality of Variances



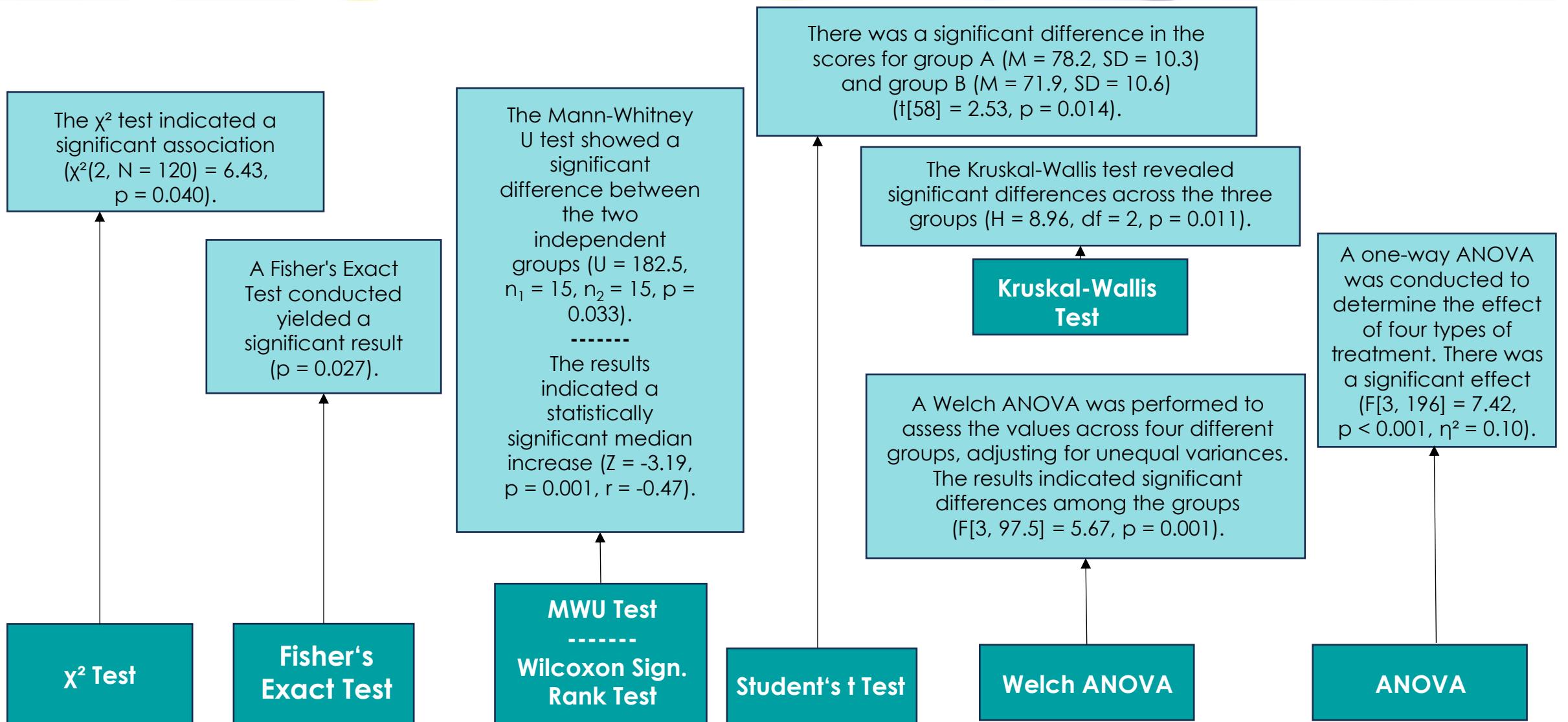
# Choosing the Right Test for Comparison



# Choosing the Right Test for Comparison

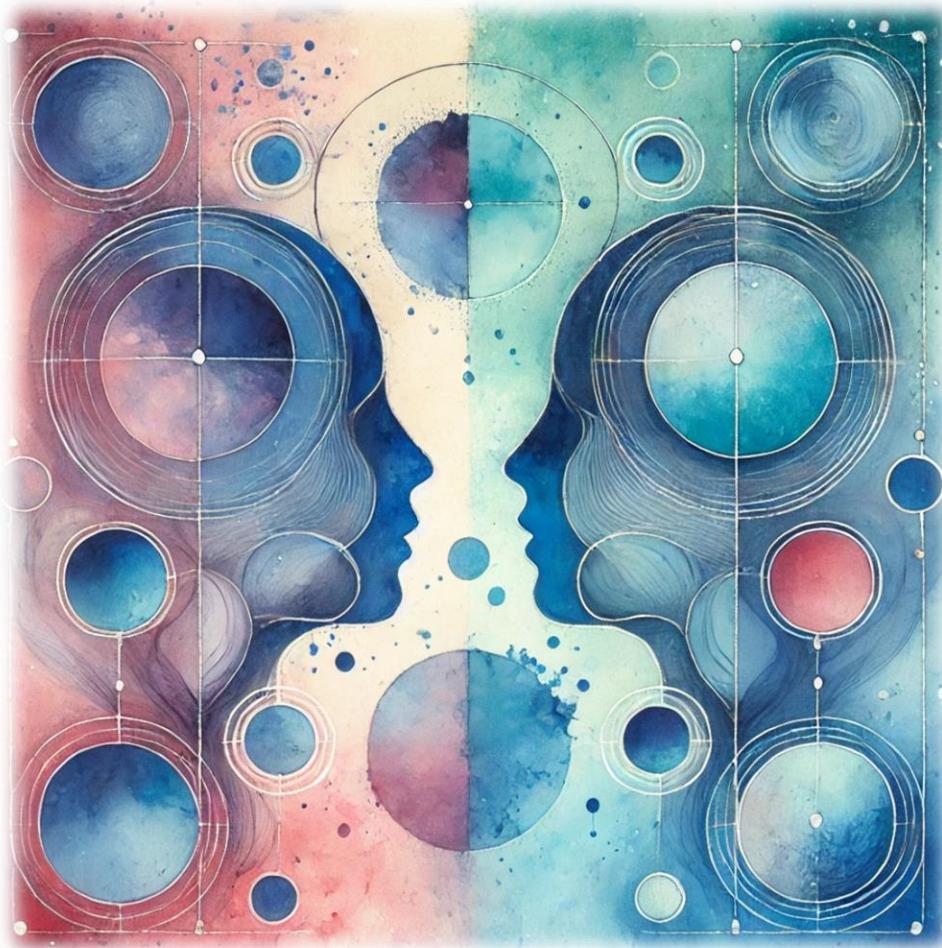


# Reporting the Right Test for Comparison



# Excursus: Testing for Equivalence

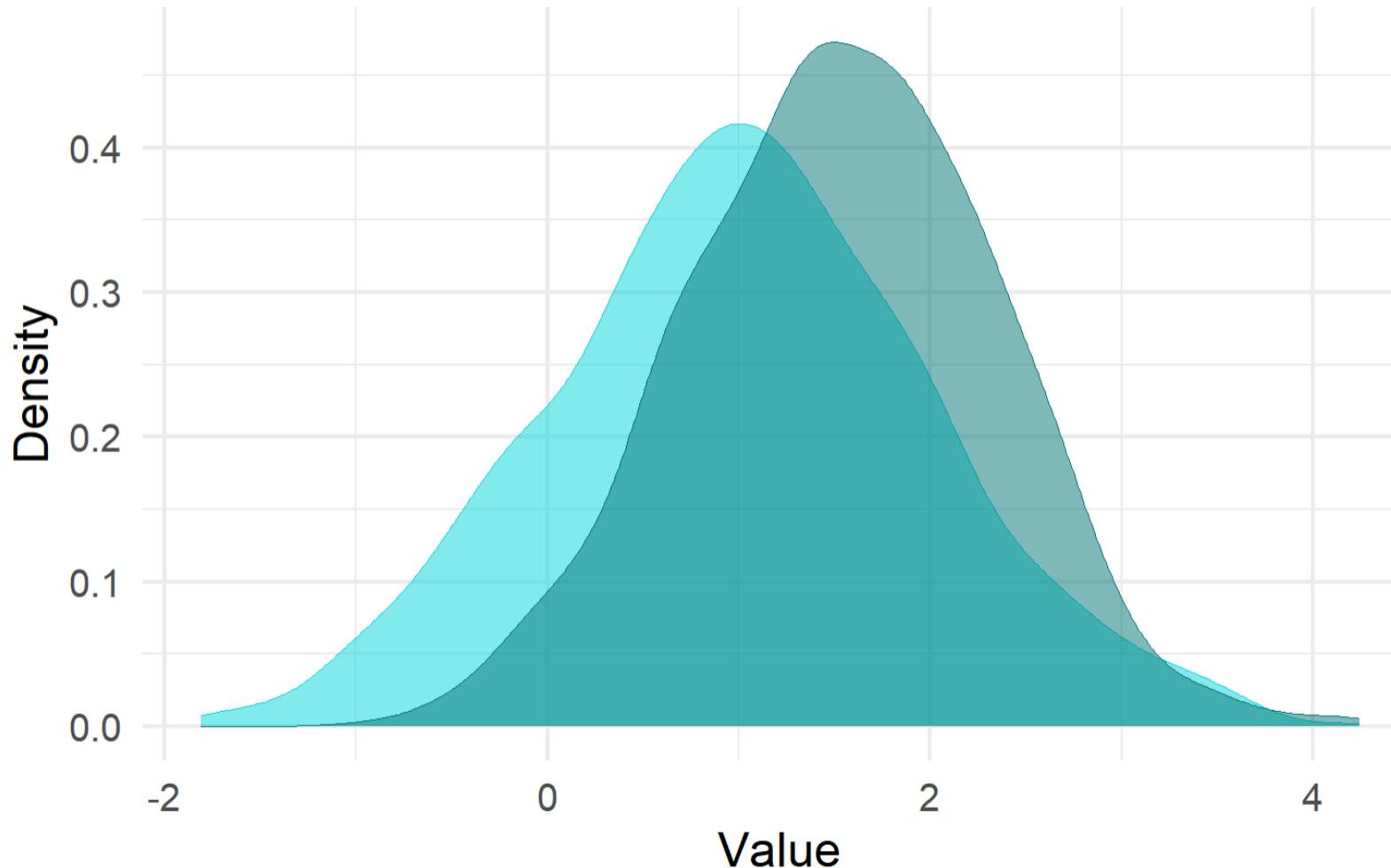
---



- Two One-Sided Tests (TOST)

# Excursus: Testing for Equality

## - TOST: Two One-Sided Tests



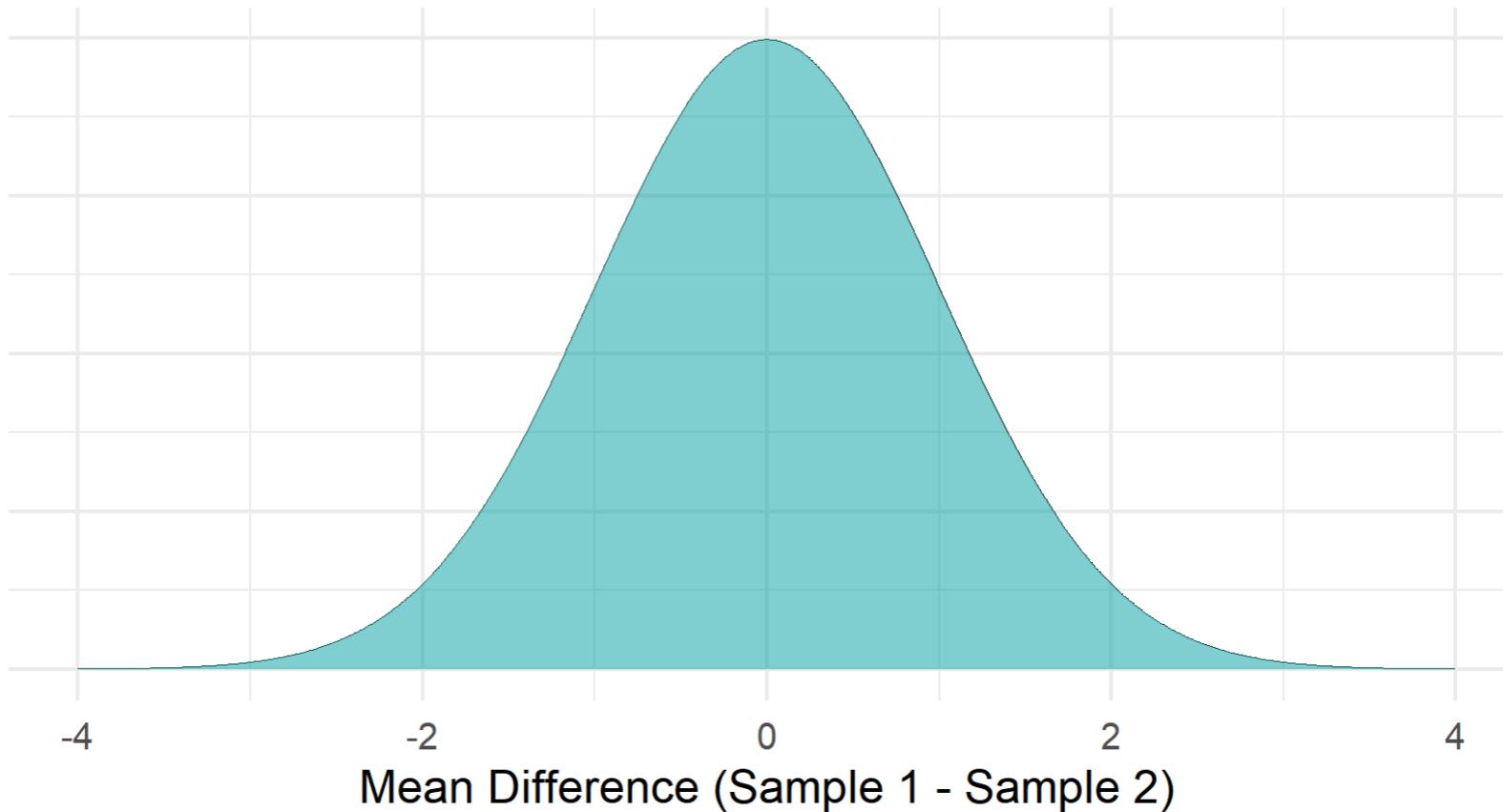
t Test:  
 $p = 0.851$

**What does this mean?**

→ Sample are not significantly different  
→ Are samples **significantly equal?**

# Excursus: Testing for Equality

- Principle of Differences in Means Test

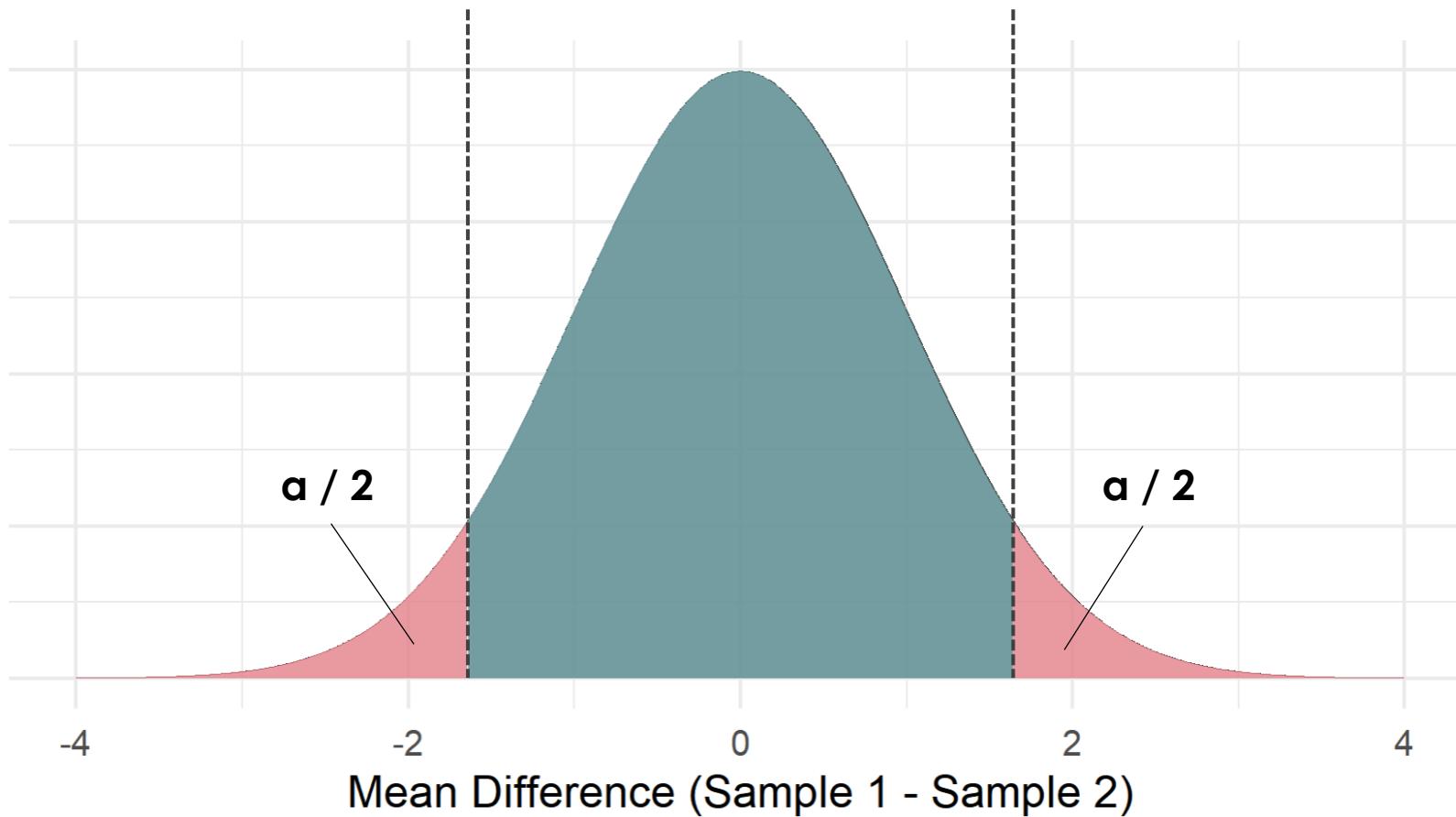


1. Calculate Expected Mean Difference Distribution Based on Sample Size, Mean and Std. Dev.

Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

# Excursus: Testing for Equality

## - Principle of Differences in Means Test

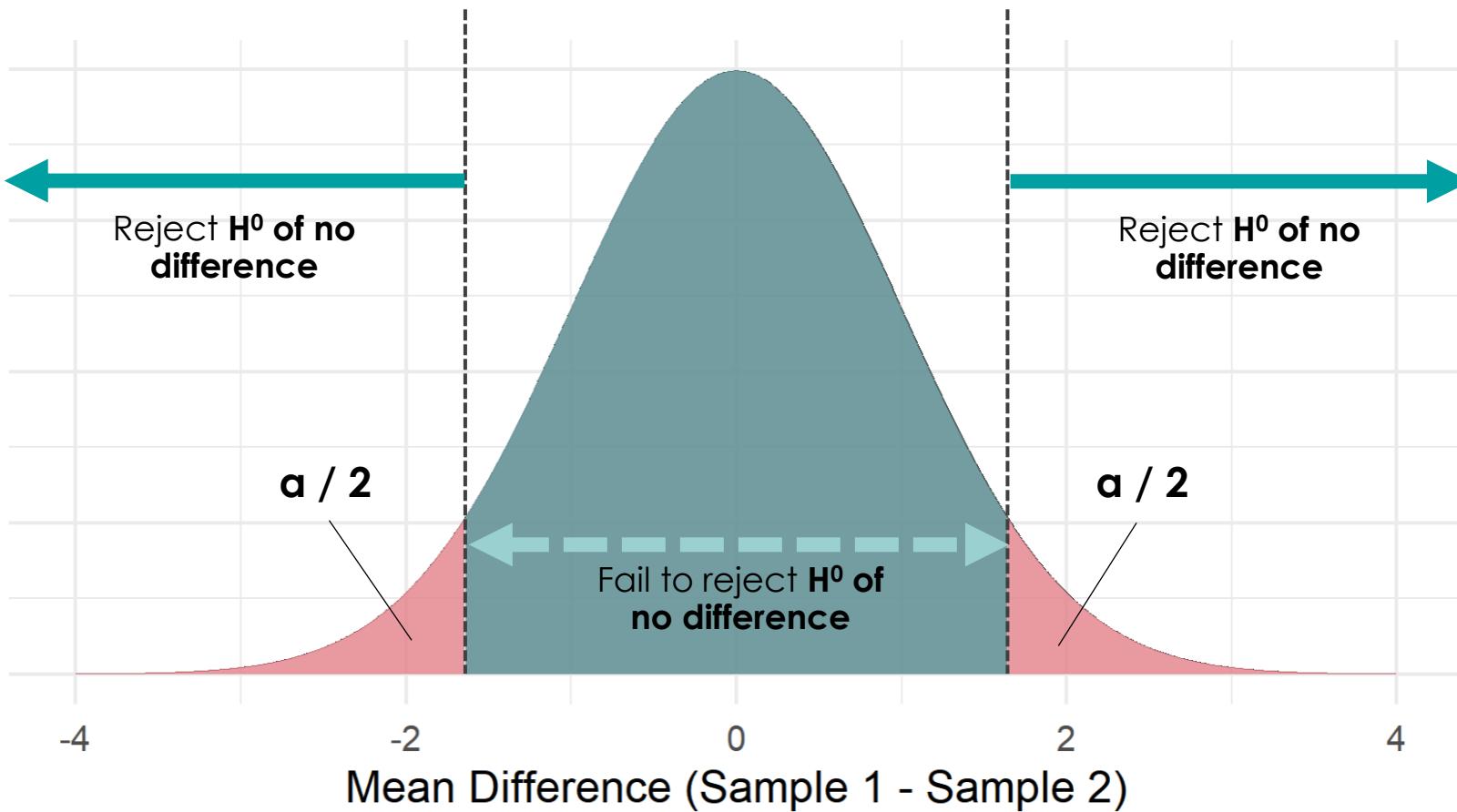


Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

1. Calculate Expected Mean Difference Difference Distribution Based on Sample Size, Mean and Std. Dev.
2. Set Error Acceptance Level (corr. to p value) on Both Sides of the Distribution
3. Accept (Sign. Diff.)/Reject (No Sign. Diff.)  $H^a$  based on Individual Mean Difference

# Excursus: Testing for Equality

## - Principle of Differences in Means Test

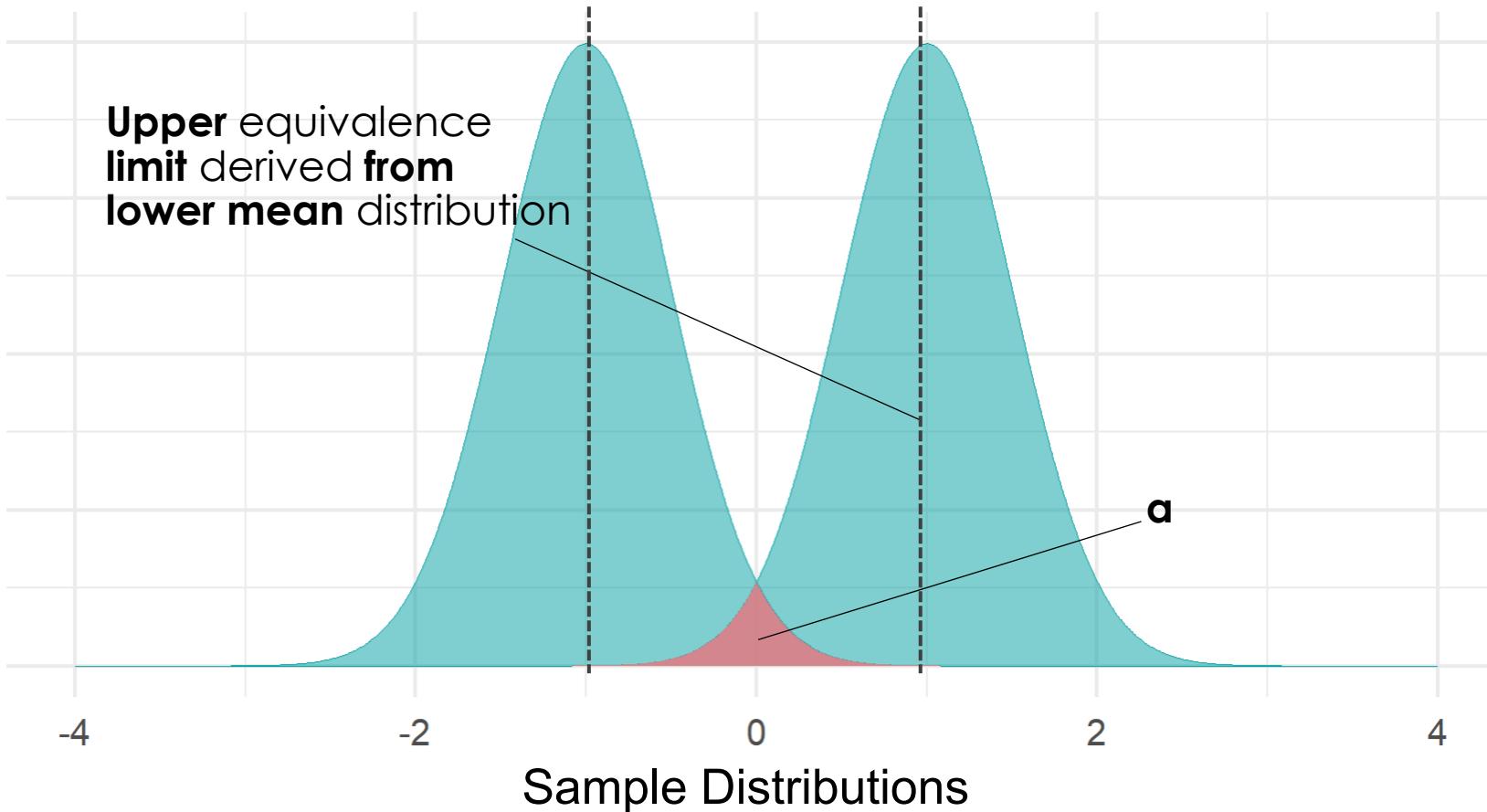


Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

1. Calculate Expected Mean Difference Difference Distribution Based on Sample Size, Mean and Std. Dev.
2. Set Error Acceptance Level (corr. to p value) on Both Sides of the Distribution
3. **Accept (Sign. Diff.)/Reject (No Sign. Diff.)  $H^a$  based on Individual Mean Difference**

# Excursus: Testing for Equality

## - Principle of Equivalence in Means Test

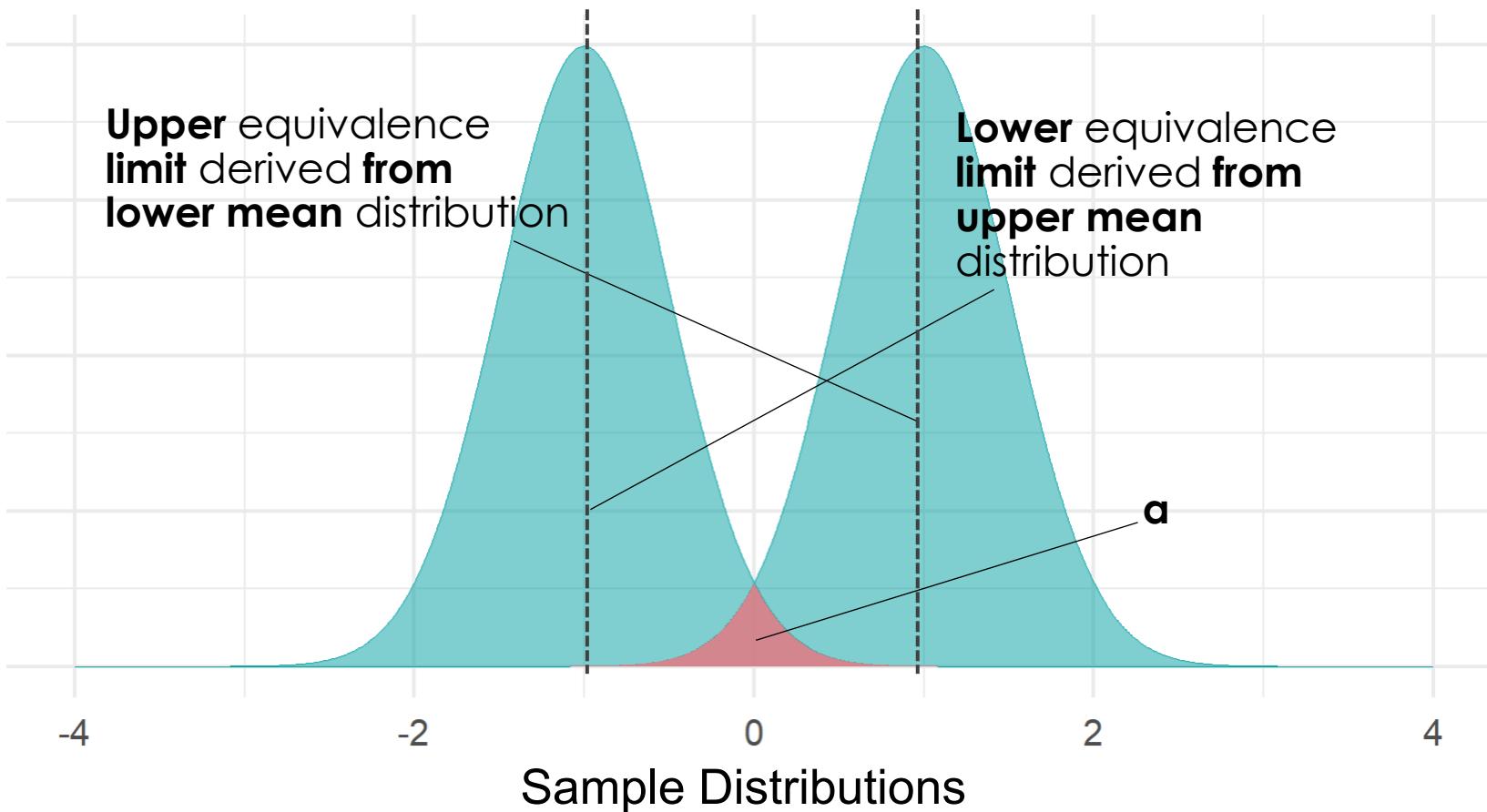


Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

1. Set Equivalence Boundaries where:
  - a. Upper limit belongs to sample with lower mean

# Excursus: Testing for Equality

## - Principle of Equivalence in Means Test

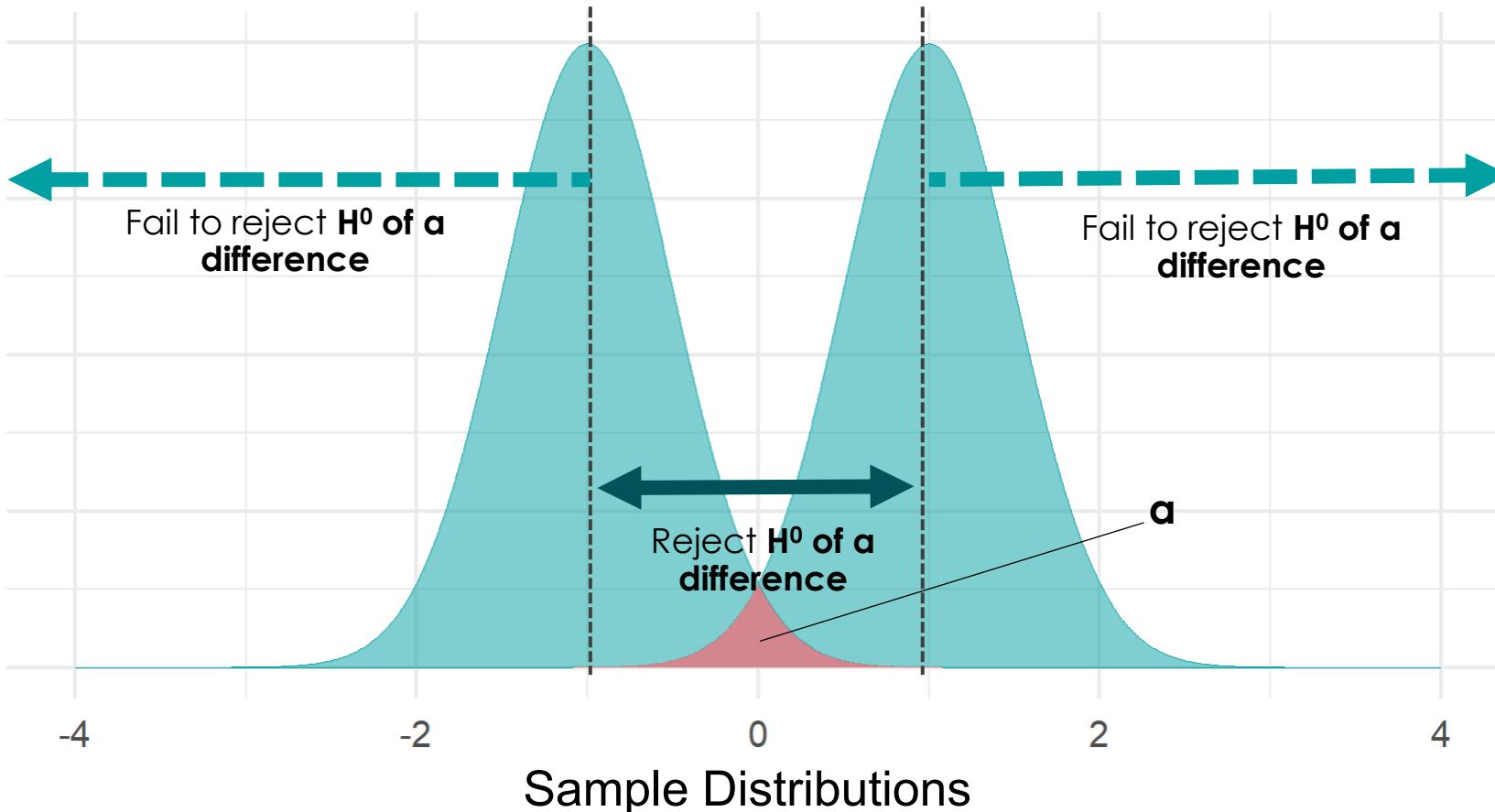


Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

1. Set Equivalence Boundaries where:
  - a. Upper limit belongs to sample with lower mean
  - b. Lower limit belongs to sample with higher mean

# Excursus: Testing for Equality

## - Principle of Equivalence in Means Test



1. Set Equivalence Boundaries where:
  - a. Upper limit belongs to sample with lower mean
  - b. Lower limit belongs to sample with higher mean
2. Accept (Sign. Diff.)/Reject (No Sign. Diff.) based on Individual Mean Difference

Adapted from Hartmann & Hidalgo (2018): An Equivalence Approach to Balance and Placebo Tests. *American Journal of Political Science*, 62(4), p.1004

# Excursus: Testing for Equality

- TOST: Two One-Sided Tests

-  $p \leq 0.05$ : Samples can be **assumed equal**



```
library(TOSTER)

# Example data: two groups of measurements
your_group1 <- c(101, 102, 103, 97, 99, 101, 100, 102, 104, 98)
your_group2 <- c(100, 98, 99, 101, 100, 99, 102, 103, 97, 97)

# Define equivalence bounds
lower_bound <- -3 # The lower margin of equivalence
upper_bound <- 3 # The upper margin of equivalence

# Perform the TOST
result <- tsum_TOST(m1 = mean(your_group1), sd1 = sd(your_group1), n1 = length(your_group1),
                     m2 = mean(your_group2), sd2 = sd(your_group2), n2 = length(your_group2),
                     hypothesis = "EQU", low_eqbound = lower_bound, high_eqbound = upper_bound)
```



```
import numpy as np
from statsmodels.stats.weightstats import ttost_ind

# Example data: two groups of measurements
your_group1 = np.array([101, 102, 103, 97, 99, 101, 100, 102, 104, 98])
your_group2 = np.array([100, 98, 99, 101, 100, 99, 102, 103, 97, 97])

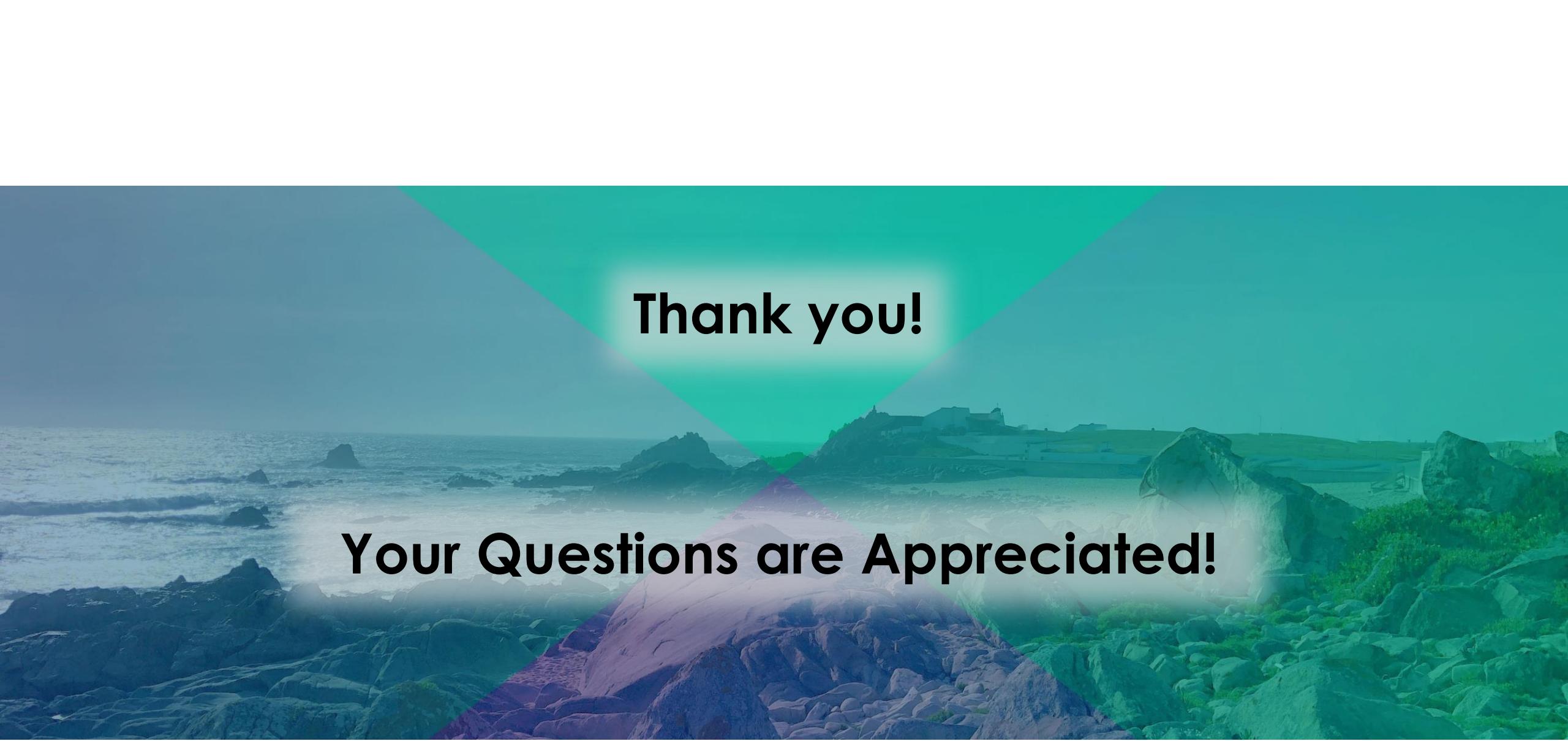
# Define equivalence bounds
lower_bound = -3 # The lower margin of equivalence
upper_bound = 3 # The upper margin of equivalence

# Perform the TOST
pvalue, res_low, res_upp = ttost_ind(your_group1, your_group2, low=lower_bound, upp=upper_bound)

# Print the overall p-value and details for each one-sided test
print(f"Overall P-value: {pvalue}")
print(f"Lower Test Statistic: {res_low[0]}, P-value: {res_low[1]}, DF: {res_low[2]}")
print(f"Upper Test Statistic: {res_upp[0]}, P-value: {res_upp[1]}, DF: {res_upp[2]}")
```

## Examples:

- Placebo/Drug Regimen does not lead to changes in blood pressure
- Gene Expression Level Stays the Same
- Method of Protein Purification does not Affect Yield



Thank you!

Your Questions are Appreciated!



frank.hause@medizin.uni-halle.de



@fhouse.bsky.social