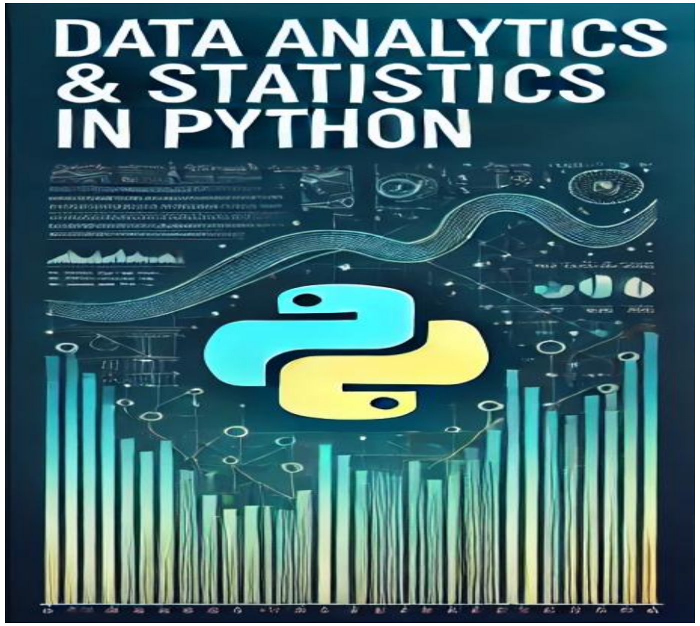
Data Analytics & Statistics in Python
Session 4: Probability and Variability



Metropolia
University of Applied Sciences

Learning data-driven decision-making with Python

Instructor: Hamed Ahmadinia, Ph.D.

Email: hamed.ahmadinia@metropolia.fi

Concepts of Today

- **Key Concepts:**
  - Probability Distributions (Discrete and Continuous)
  - Expected Value, Standard Deviation, and Variance
  - Gaussian (Normal) Distribution
  - Z-score and Outliers
  - Statistical Tests (T-test, Mann-Whitney U, Chi-Squared Test)

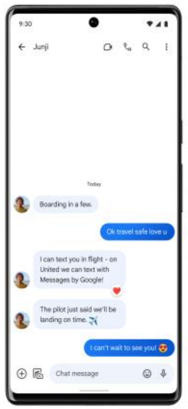
Understanding Probability Distributions

- **What is a Probability Distribution?**
  - Shows the likelihood of different outcomes.

- **Rules (Kolmogorov Axioms):**
  - Probabilities are always non-negative.
  - Total probability adds up to 1.
  - Probabilities of disjoint (non-overlapping) events add up.

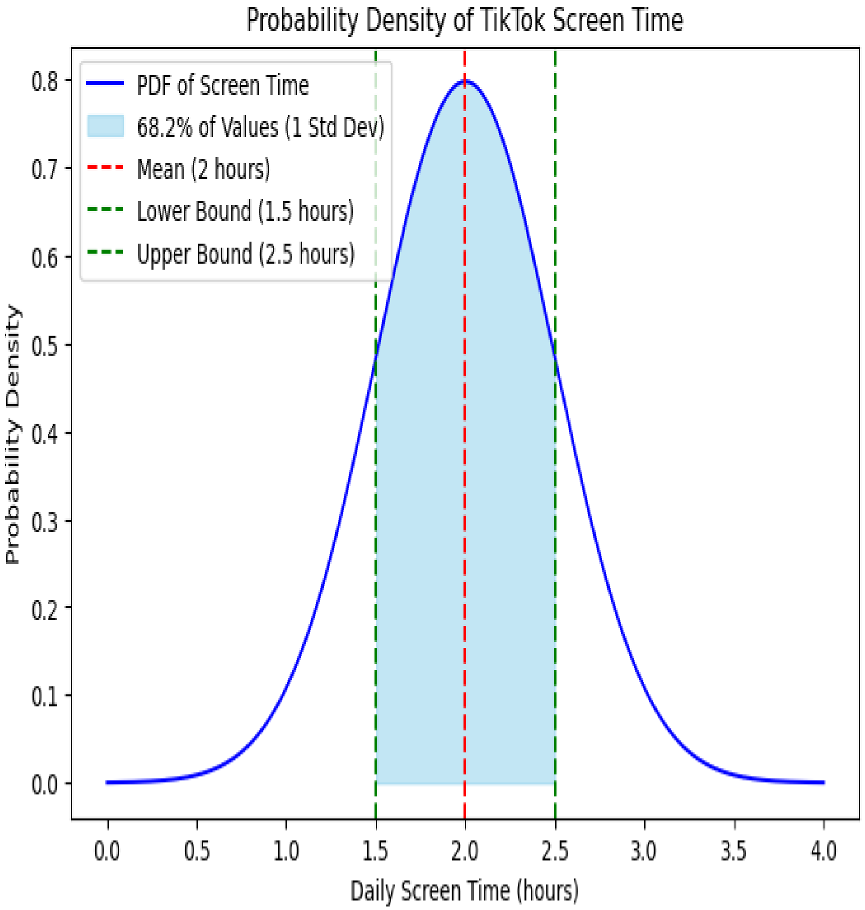
Types of Distributions

- **Discrete Distributions (specific, countable outcomes):**
  - **Example:** Number of likes on a social media post $\Omega = \{10,20,30,40,50\}$.
- **Common Discrete Distributions:**
  - **Binomial:** Number of successful shots in 10 basketball attempts (two possible outcomes: success or failure).
  - **Poisson:** Number of messages received in a group chat in an hour (number of events occurring in a fixed interval of time or space).
  - **Geometric:** Number of tries to win a video game challenge (Trials are independent with a constant probability of success).
  - **Hypergeometric:** Number of rare items found when opening a limited number of loot boxes (without replacement) (probability of success changes with each draw).



Continuous Distributions

- **What is a Continuous Distribution?**
  - Describes outcomes that can take any value within a range.
  - **Example:** The time a student spends on TikTok daily, down to seconds (e.g., 2 hours 17 minutes and 32 seconds).
- **Probability Density Function (PDF):**
  - The curve is called a probability density function (PDF).
  - Shows where values (like time spent) are more or less likely to fall.
- **Normal Distribution (Bell Curve):**
  - Common in everyday life (e.g., average screen time per day).
- **Example:**
  - Average TikTok use: 2 hours/day.
  - 68% of students use TikTok between 1.5 and 2.5 hours daily (if screen time follows a normal distribution).

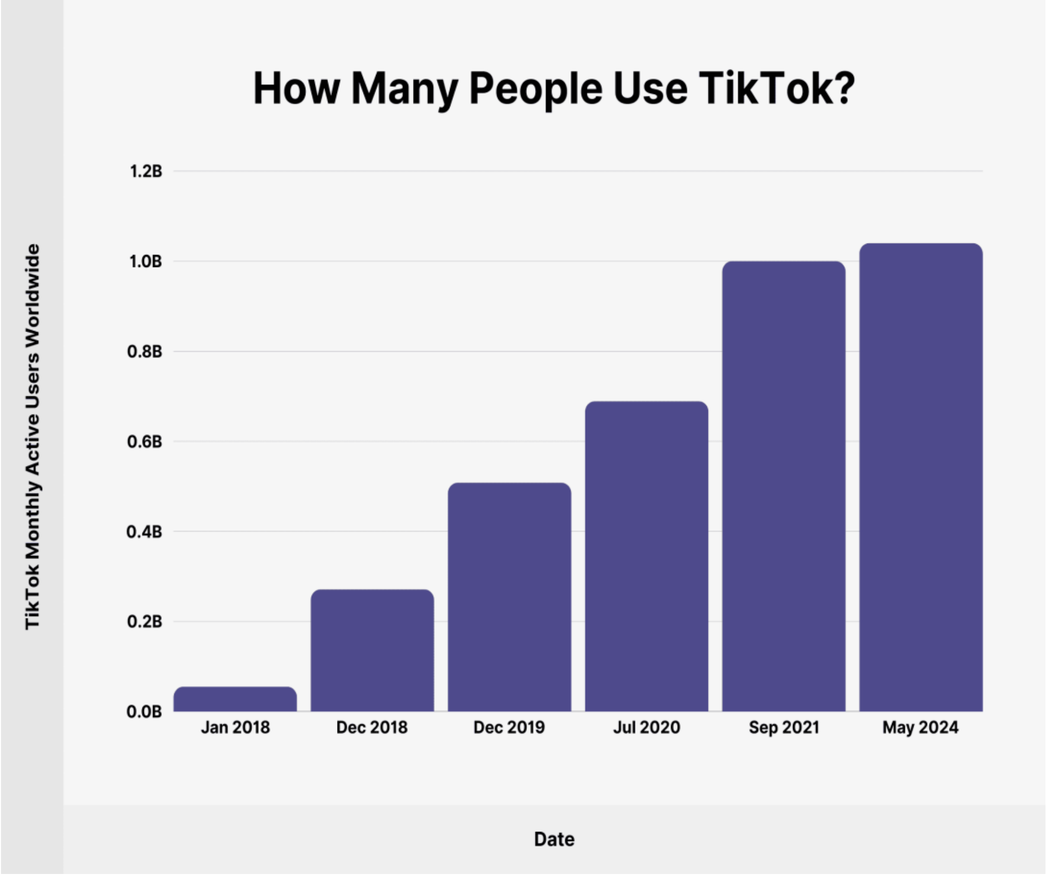


Understanding Expected Value

- **What is Expected Value?**
  - The "average" outcome, considering the probabilities of different outcomes.
- **Discrete Case (Countable Outcomes)**
  - Formula: $E(x) = \sum_{i=1}^{n} \rho(x_i) \cdot x_i$
  - Where: $x_i = Possible\ Outcome$
    $p(x_i) = Probability\ of\ the\ outcome\ x_i$
  - Example: Lottery Game:
    - Winning chance: 1 out of 100
    - Prize: €1000
    - Expected value: $E(x) = \frac{1}{100} \times 1000 + \frac{99}{1100} \times 0 = €\,10$
  - → On average, you "expect" to win €10, but you could win more or nothing at all.

Understanding Expected Value

- **Continuous Case (Uncountable Outcomes)**
  - When values can take a range (e.g., any height or exact time).
  - Instead of adding probabilities, we "integrate" across all possible values.
  - Example: Average TikTok screen time from 1.5 to 2.5 hours daily.
- **Law of Large Numbers (LLN)**
  - Key Idea: The more trials you run, the closer your sample average gets to the true average.
  - Example: If you track your daily TikTok usage over months, the average gets closer to your actual average daily screen time.



Understanding Data Spread

- Why Measure Spread?
  - Variance (σ^2): Shows how far data points are from the average.

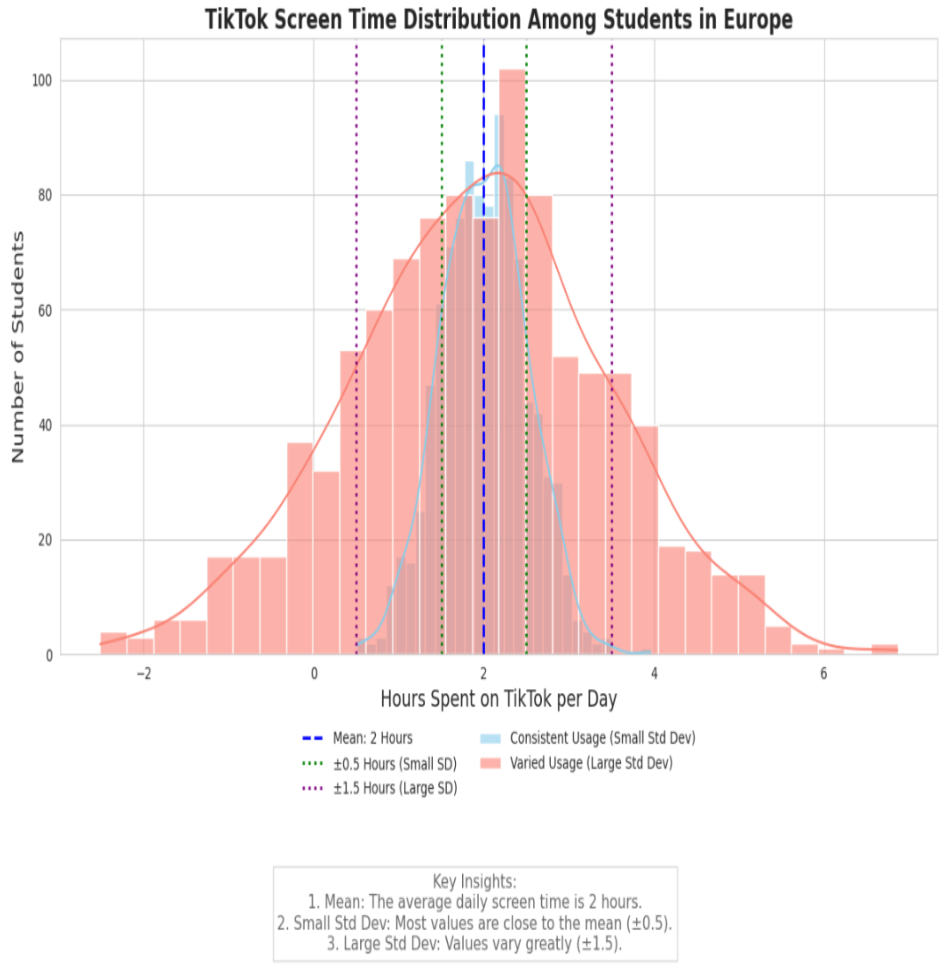
$$Var(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^a$$

(Think of it as the "average of squared differences" from the average value.)

  - Standard Deviation (σ): The square root of variance (brings it back to the original units, e.g., hours, euros).

  - Example: TikTok Screen Time in Europe
    - Small Standard Deviation: Most students spend around 2 ± 0.5 hours/day (similar usage).
    - Large Standard Deviation: Screen time ranges widely, from 30 minutes to 5 hours/day (big differences).

Takeaway: Standard deviation shows whether data points are "close" or "spread out" from the average.



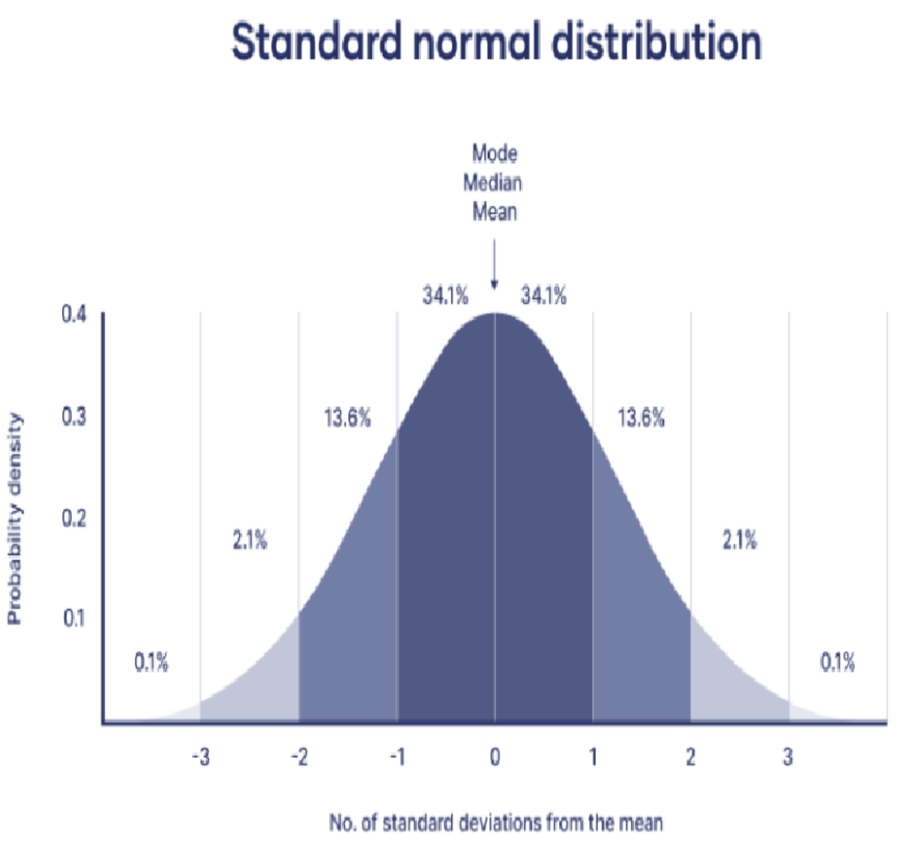
Python Example

```python
import numpy as np

# Sample data: hours spent on social media
screen_time = np.array([1.5, 2.0, 2.3, 1.8, 2.5])
mean = np.mean(screen_time)
variance = np.var(screen_time)
std_dev = np.std(screen_time)

print(f"Mean: {mean}, Variance: {variance}, Std Dev: {std_dev}")
```

```
Mean: 2.02, Variance: 0.1256, Std Dev: 0.354400902933387
```

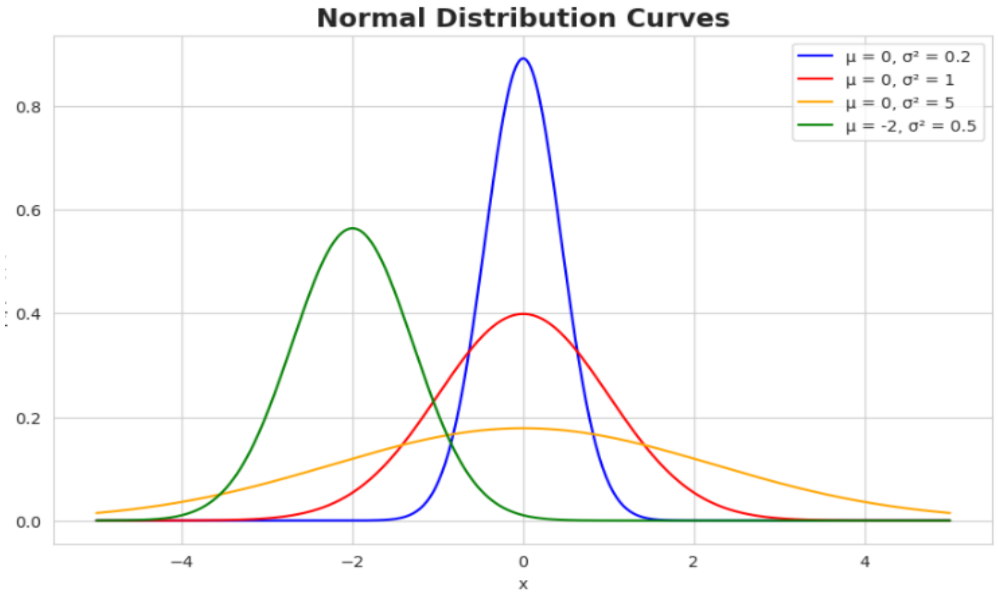
Normal Distribution

- **What is a Normal (Gaussian) Distribution?**
  - A continuous, bell-shaped probability distribution.
  - Commonly used in fields like statistics, data analysis, and machine learning.
- **Key Characteristics:**
  - Mean (μ): The center of the distribution (the highest point of the curve).
  - Variance (σ^2): Describes how spread out the data is.
  - A small variance creates a narrow, tall curve (data close to the mean).
  - A large variance creates a wide, flat curve (data more spread out).



Visualizing Normal Distribution

```python
import numpy as np
import matplotlib.pyplot as plt

# Define Gaussian distributions
x = np.linspace(-5, 5, 500)
mean_values = [0, 0, 0, -2]
variances = [0.2, 1, 5, 0.5]
colors = ['blue', 'red', 'orange', 'green']

# Plot the Gaussian curves
plt.figure(figsize=(10, 6))
for mean, var, color in zip(mean_values, variances, colors):
    std_dev = np.sqrt(var)
    y = (1 / (std_dev * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x - mean) / std_dev) ** 2)
    plt.plot(x, y, label=f'μ = {mean}, σ² = {var}', color=color)

plt.title("Normal Distribution Curves", fontsize=16, weight='bold')
plt.xlabel("x")
plt.ylabel("φ(μ, σ²)(x)")
plt.legend(loc='upper right')
plt.show()
```



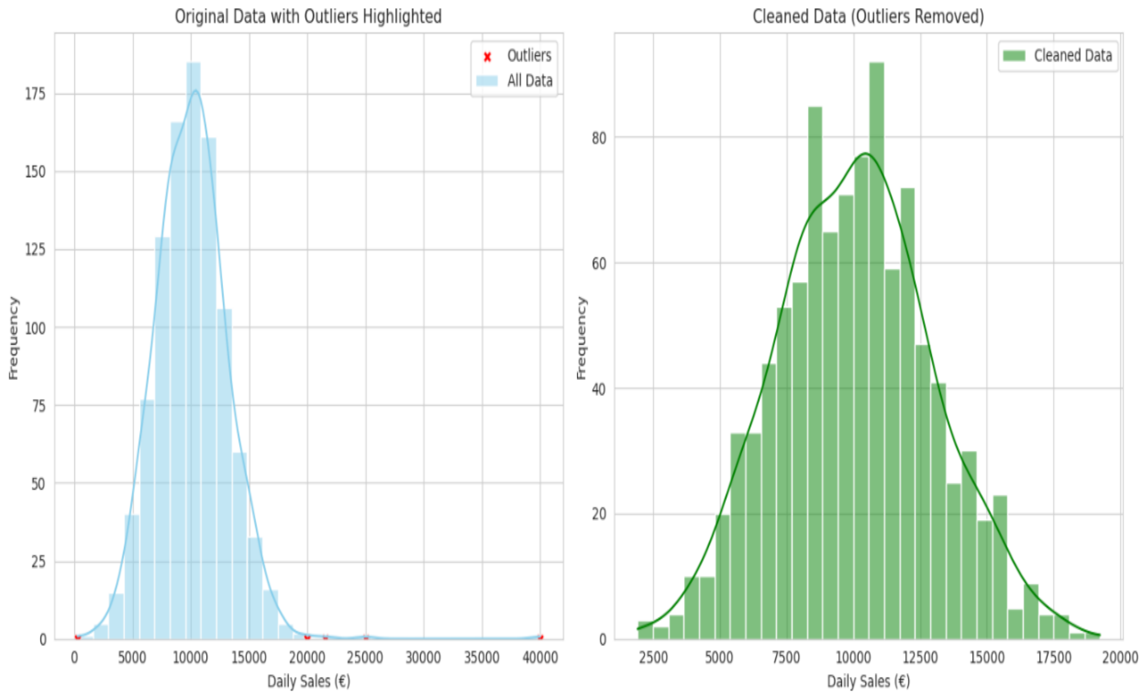
- The graph shows **normal distributions** with different means and variances:
  - **Blue Curve:** A narrow peak (low variance, data tightly clustered).
  - **Red Curve:** A typical bell-shaped curve.
  - **Orange Curve:** A flatter, spread-out curve (large variance).
  - **Green Curve:** Shifted left (mean = -2), less spread out.
- **Key Takeaway:**
  - Changing the **mean (μ)** shifts the curve left or right.
  - Changing the **variance (σ^2)** adjusts the width of the curve

Z-Scores and Outliers

- **What is a Z-Score?**
  - A number showing how far a data point is from the average (mean) in **standard deviation units**.
  - **Formula:** $z = \frac{x-\mu}{\sigma}$
- **Why is it Useful?**
  - Detects **outliers** (unusual data points).
  - Helps clean and improve data for better analysis.
- **Key Points:**
  - **Z ≈ 0:** Data point is close to the average.
  - **Z > 3 or < -3:** Likely an outlier.
- **Example:**
  - Average daily sales: €10,000
  - Sales on one day: €20,000

Z-Score: $z = \frac{20000-10000}{3000} \approx 36^{33}$ (This day's sales may be an outlier)

Z-Score Application in Python




```python
import numpy as np
import pandas as pd

# Create sample data for daily revenue (1000 days)
df = pd.DataFrame({
    'Revenue (€)': np.random.normal(10000, 3000, size=1000)
})

# Z-score calculation
df['Z-score'] = (df['Revenue (€)'] - df['Revenue (€)'].mean()) / df['Revenue (€)'].std()

# Filter out outliers (e.g., Z-score > 3 or < -3)
outliers = df[(df['Z-score'].abs() > 3)]
cleaned_data = df[(df['Z-score'].abs() <= 3)]

# Display the counts
print(f"Outliers: {len(outliers)}")
print(f"Cleaned Data: {len(cleaned_data)}")
```

```
Outliers: 3
Cleaned Data: 997
```

Key Insights from the Code:

- **Z-Score Formula**: Calculates how far each revenue point is from the mean.
- **Boolean Mask**: Keeps values within 3 standard deviations.
- **Application**: After removing outliers, cleaner data allows more accurate analysis.

What Are Statistical Tests and P-values?

- **Null Hypothesis (H_0):** A starting assumption (e.g., "no difference in averages").
- **P-value**: A number that shows how likely your data is if H_0 is true.
  - Small p-value (< 0.05): Your result is surprising → Reject H_0.
  - Large p-value (> 0.05): Your result isn't surprising → Keep H_0.

- **Example**:
  - Suppose you think students in your school get more sleep than the national average of 7 hours.
  - H_0: "Average sleep time is **7 hours**."
  - If your data shows an average of **8 hours** and the p-value is **0.01**, the low p-value suggests your data isn't just random—it supports your claim!

Tests for Significance and p-values

Procedure:

1. Define H_0 and alternative hypothesis (H_1).
2. Choose significance level (e.g., 0.05).
3. Collect sample data.Calculate test statistic and p-value.
4. Compare p-value to threshold → Reject or fail to reject H_0.

Common Misconceptions:

Misconception	Reality
P-value < 0.05 means H_0 is 100% false.	No—it just means your result is very unlikely under H_0.
Small p-value = Big effect.	Not always! A small p-value shows something is significant, but the size of the effect may still be small.

Parametric vs Non-Parametric Tests

- **Parametric Tests (e.g., T-tests):**

Assume the data follows a specific distribution (usually normal). They require assumptions about the data, such as homogeneity of variance.

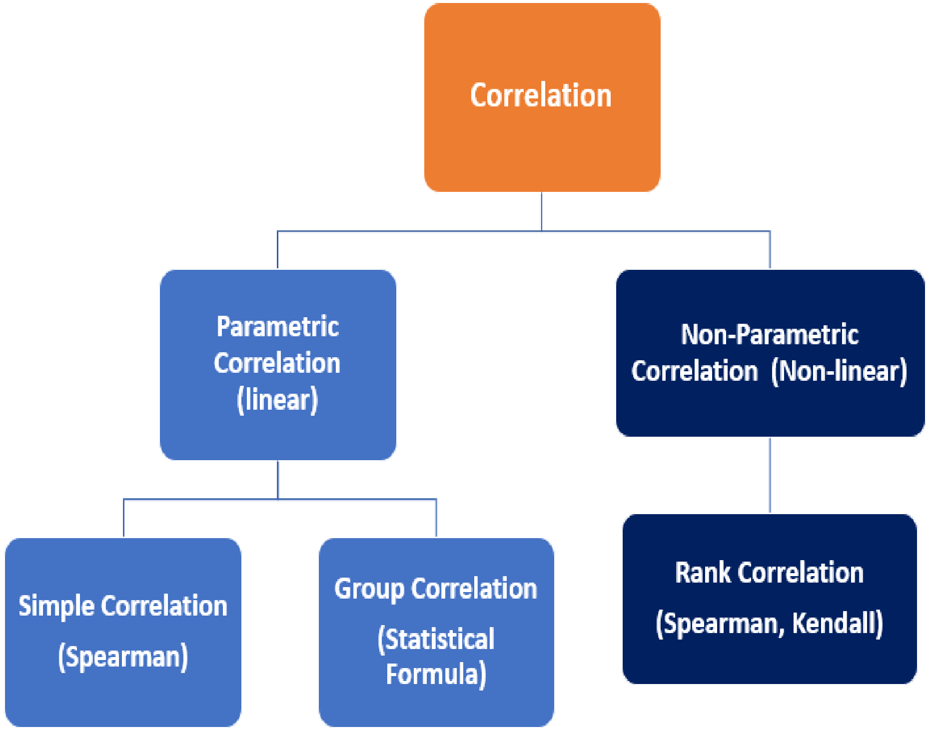
  - **Example:** Comparing the average time spent studying between two groups of students.

- **Non-Parametric Tests (e.g., Mann-Whitney U Test, Chi-Squared Test):**

Do not require assumptions about the data distribution and can handle ordinal or non-normally distributed data.

  - **Example:** Comparing user satisfaction ratings (ranked scores) for two different apps.

- Read more



T-tests and Non-Parametric Tests

Test Type	Purpose	Example	Key Points
One-sample T-test	Compare sample mean to a known population mean	Do students study more than 10 hours per week?	If p-value < 0.05: Reject H_0 (significant difference). If p-value > 0.05: Fail to reject H_0.
Two-sample T-test	Compare means of two independent groups.	Do gamers and non-gamers have different sleep hours?	
Paired-sample T-test	Compare means within the same group (before/after).	Does an exercise program reduce resting heart rate?	
Mann-Whitney U Test	Non-parametric test: Compare two groups when data isn’t normally distributed	Compare satisfaction scores between two apps.	Used for ordinal or non-normal data
Chi-Squared Test (Goodness of Fit)	Check if observed data fits expected proportions.	Is a die fair (equal probability for all sides)?	• Observations must be independent. • Expected frequency ≥ 5
Chi-Squared Test (Independence)	Check if two variables (e.g., education level and voting preference) are related.	Are education level and voting preference related?	• Observations must be independent. • Used with contingency tables (rows and columns for variables).

Python Code Examples for Each Tests

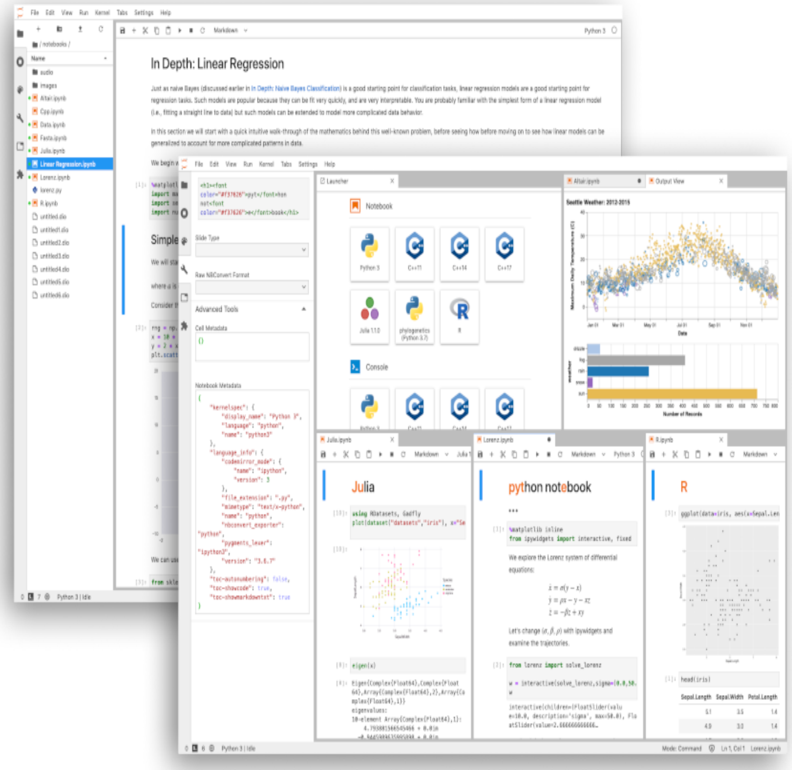
```python
# Import necessary functions
from scipy.stats import ttest_1samp, mannwhitneyu, chisquare, chi2_contingency

# 1. One-Sample T-Test: Compare sample mean to population mean
t_stat, p_value = ttest_1samp([2.3, 2.5, 2.8, 3.0, 3.2], popmean=3.0)
print(f"One-Sample T-Test: T={t_stat:.2f}, p={p_value:.2f}")

# 2. Mann-Whitney U Test: Compare two groups (non-parametric)
u_stat, p_value = mannwhitneyu([1, 2, 3, 3, 4], [3, 4, 5, 5, 6])
print(f"Mann-Whitney U Test: U={u_stat:.2f}, p={p_value:.2f}")

# 3. Chi-Squared Goodness of Fit Test: Check if observed data fits expected proportions
observed = [25, 30, 45]  # Observed frequencies
expected = [33.33, 33.33, 33.33]  # Adjusted to sum up to the same total as observed
total_observed = sum(observed)
expected_scaled = [total_observed * (x / sum(expected)) for x in expected]  # Rescale expected frequencies

chi2_stat, p_value = chisquare(f_obs=observed, f_exp=expected_scaled)
print(f"Chi-Squared Goodness of Fit: χ²={chi2_stat:.2f}, p={p_value:.2f}")

# 4. Chi-Squared Test for Independence: Check if two variables are related
chi2_stat, p_value, _, _ = chi2_contingency([[50, 30, 20], [30, 40, 30], [20, 30, 50]])
print(f"Chi-Squared Test for Independence: χ²={chi2_stat:.2f}, p={p_value:.2f}")
```

```
One-Sample T-Test: T=-1.47, p=0.22
Mann-Whitney U Test: U=2.50, p=0.04
Chi-Squared Goodness of Fit: χ²=6.50, p=0.04
Chi-Squared Test for Independence: χ²=30.00, p=0.00
```

Notebook Review

Walk through how to apply key Python concepts in a Jupyter Notebook:

- Probability Distributions
- Expected Value, Standard Deviation, and Variance
- Normal Distribution
- Z-score and Outliers
- Statistical Tests



Kahoot Quiz Time!

Let's Test Our Knowledge!

Hands-on Exercise

Form groups (2–3 members).

- Download *Hands-on Exercise #4* from the course page.
- Complete the coding tasks and discuss your solutions.
- Don’t forget to add the names of your group members to the file.
- Submit your completed *Hands-on Exercise* to the course Moodle page or send it to the teacher's email address.

Reference

- Vohra, M., & Patil, B. (2021). A Walk Through the World of Data Analytics. , 19-27. https://doi.org/10.4018/978-1-7998-3053-5.ch002.
- VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media. Available at https://jakevdp.github.io/PythonDataScienceHandbook/
- Severance, C. (2016). Python for everybody: Exploring data using Python 3. Charles Severance. Available at https://www.py4e.com/html3/
- McKinney, W. (2017). *Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter*. O'Reilly Media.