Statistical Tests using Python

Table of contents

0.1 z-Test

The Test

A z-test determines whether there is a significant difference between a sample mean and a known population mean when the population standard deviation is known.

Usage

- Comparing a sample mean to a population mean.
- Large sample size ($n > 30$) or normally distributed population.

Formula

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

where:
- $\bar{x}$: sample mean
- μ: population mean
- σ: population standard deviation
- n: sample size

Python Function

```python
from statsmodels.stats.weightstats import ztest
```

Example

- **Scenario:** A hospital claims the average recovery time from a specific surgery is 10 days. A sample of 30 patients has recovery times:
  `[11, 9, 10, 10, 12, ..., 10]` (30 data points).

- Population standard deviation = 2 days.
- Test at a 5% significance level.
- **Null Hypothesis** (H_0): Mean recovery time = 10 days ($\mu = 10$).

- **Alternative Hypothesis** (H_1): Mean recovery time 10 days ($\mu \neq 10$).

Python Code

```python
import numpy as np
from statsmodels.stats.weightstats import ztest

# Data
recovery_times = [11, 9, 10, 10, 12, 11, 9, 10, 12, 10, 11, 10, 9, 12, 11,
    10, 9, 12, 10, 11, 9, 12, 10, 10, 11, 12, 10, 9, 11, 10]
population_mean = 10

# Perform z-test
z_stat, p_value = ztest(recovery_times, value=population_mean)
print(f"Z-statistic: {z_stat}, P-value: {p_value}")

# Conclusion
if p_value < 0.05:
    print("Reject H0: Recovery time significantly differs from 10 days.")
else:
    print("Fail to reject H0: No significant difference in recovery time.")
```

```
Z-statistic: 2.282167621845001, P-value: 0.022479445885689838
Reject H0: Recovery time significantly differs from 10 days.
```

0.2 t-Test (Independent Samples)

The Test
A t-test compares the means of two independent samples to determine if they are significantly different.

Usage
- Comparing two groups (e.g., treatment vs. control).
- Used when the population standard deviation is unknown.

Formula

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

where:
- $\bar{x}_1, \bar{x}_2$: sample means
- s_1, s_2: sample standard deviations
- n_1, n_2: sample sizes

Python Function

```python
from scipy.stats import ttest_ind
```

Example
- **Scenario**: Compare weight loss (kg) after 8 weeks for two diets:
- **Diet A**: [5, 6, 7, 5, 6]
- **Diet B**: [4, 5, 6, 4, 5]

- **Null Hypothesis** (H_O): Mean weight loss is the same for both diets ($\mu_1 = \mu_2$).

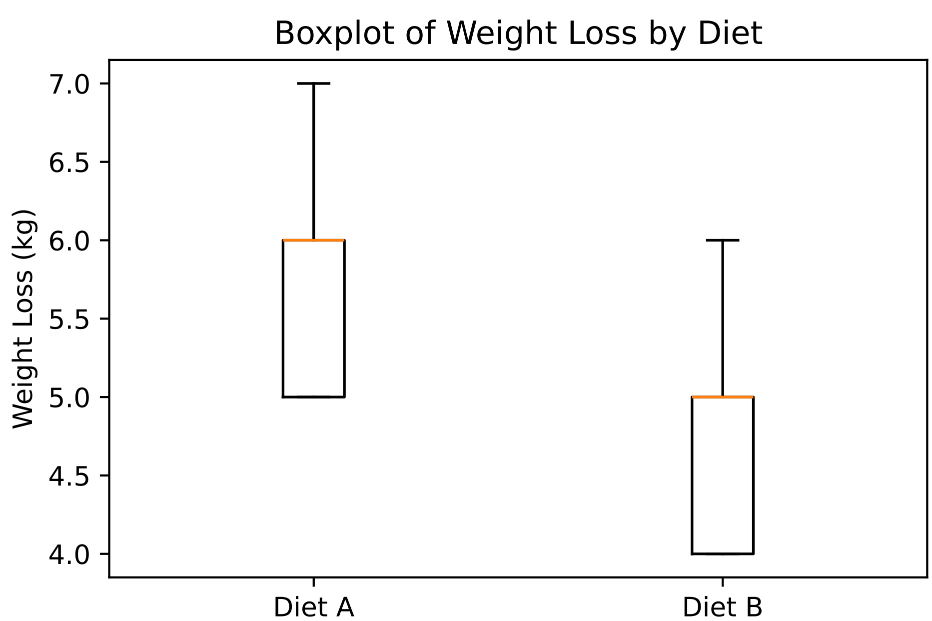
- **Alternative Hypothesis** (H_1): Mean weight loss differs ($\mu_1 \neq \mu_2$).

Python Code

```python
from scipy.stats import ttest_ind
import matplotlib.pyplot as plt

# Data
diet_a = [5, 6, 7, 5, 6]
diet_b = [4, 5, 6, 4, 5]

# Perform t-test
t_stat, p_value = ttest_ind(diet_a, diet_b, equal_var=True)
```

```
print(f"T-statistic: {t_stat}, P-value: {p_value}")

# Boxplot
data = [diet_a, diet_b]
labels = ['Diet A', 'Diet B']
plt.boxplot(data, tick_labels=labels)
plt.title('Boxplot of Weight Loss by Diet')
plt.ylabel('Weight Loss (kg)')
plt.show()

# Conclusion
if p_value < 0.05:
    print("Reject H0: Significant difference in weight loss between diets.")
else:
    print("Fail to reject H0: No significant difference between diets.")
```

```
T-statistic: 1.8898223650461363, P-value: 0.09545200899274052
```




```
Fail to reject H0: No significant difference between diets.
```

0.3 ANOVA (Analysis of Variance)

The Test
ANOVA tests whether the means of three or more groups are significantly different.

Usage
- Comparing means across multiple groups (e.g., test scores of students taught by different teaching methods).

Assumptions
1. The dependent variable is continuous.
2. Groups are independent.
3. The data in each group is normally distributed.
4. Homogeneity of variances across groups.

Formula

$$F = \frac{\text{Between-group variability}}{\text{Within-group variability}}$$

Python Function

```python
from scipy.stats import f_oneway
```

Example
- **Scenario**: Compare test scores for three teaching methods:
- **Traditional Teaching**: [85, 88, 90, 87, 86]
- **Online Teaching**: [78, 75, 80, 77, 79]
- **Hybrid Teaching**: [92, 94, 89, 91, 93]

- **Null Hypothesis** (H_0): All teaching methods have the same mean test score ($\mu_{Traditional} = \mu_{Online} = \mu_{Hybrid}$).

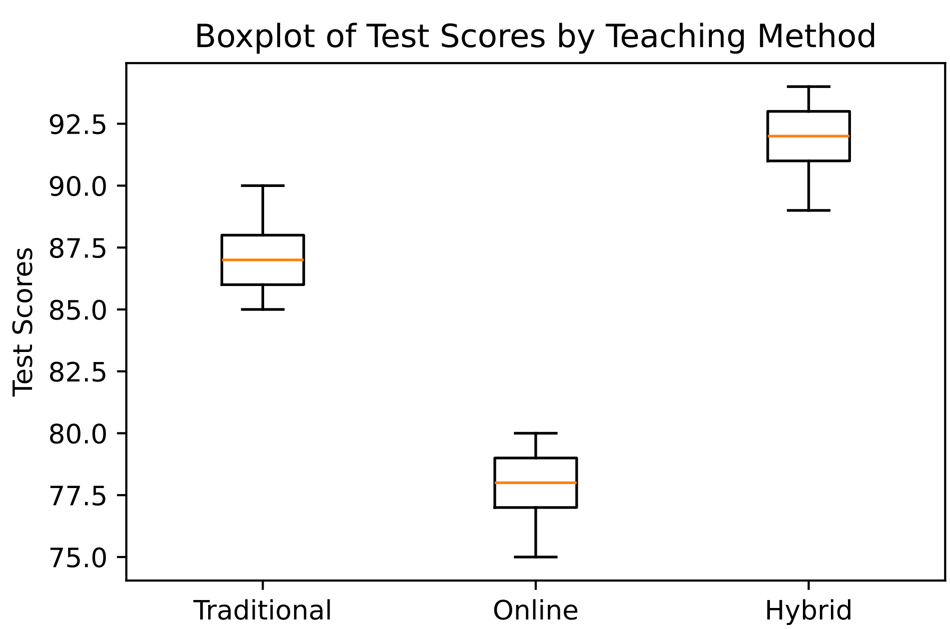
- **Alternative Hypothesis** (H_1): At least one teaching method has a different mean score.

Python Code

```python
from scipy.stats import f_oneway
import matplotlib.pyplot as plt

# Data
traditional = [85, 88, 90, 87, 86]
online = [78, 75, 80, 77, 79]
```

```python
hybrid = [92, 94, 89, 91, 93]

# Perform ANOVA
f_stat, p_value = f_oneway(traditional, online, hybrid)
print(f"F-statistic: {f_stat}, P-value: {p_value}")

# Boxplot
data = [traditional, online, hybrid]
labels = ['Traditional', 'Online', 'Hybrid']
plt.boxplot(data, tick_labels=labels)
plt.title('Boxplot of Test Scores by Teaching Method')
plt.ylabel('Test Scores')
plt.show()

# Conclusion
if p_value < 0.05:
    print("Reject H0: At least one teaching method has a different mean
     ↪  score.")
else:
    print("Fail to reject H0: No significant difference in mean scores.")
```

```
F-statistic: 68.81081081081048, P-value: 2.6614685096802244e-07
```



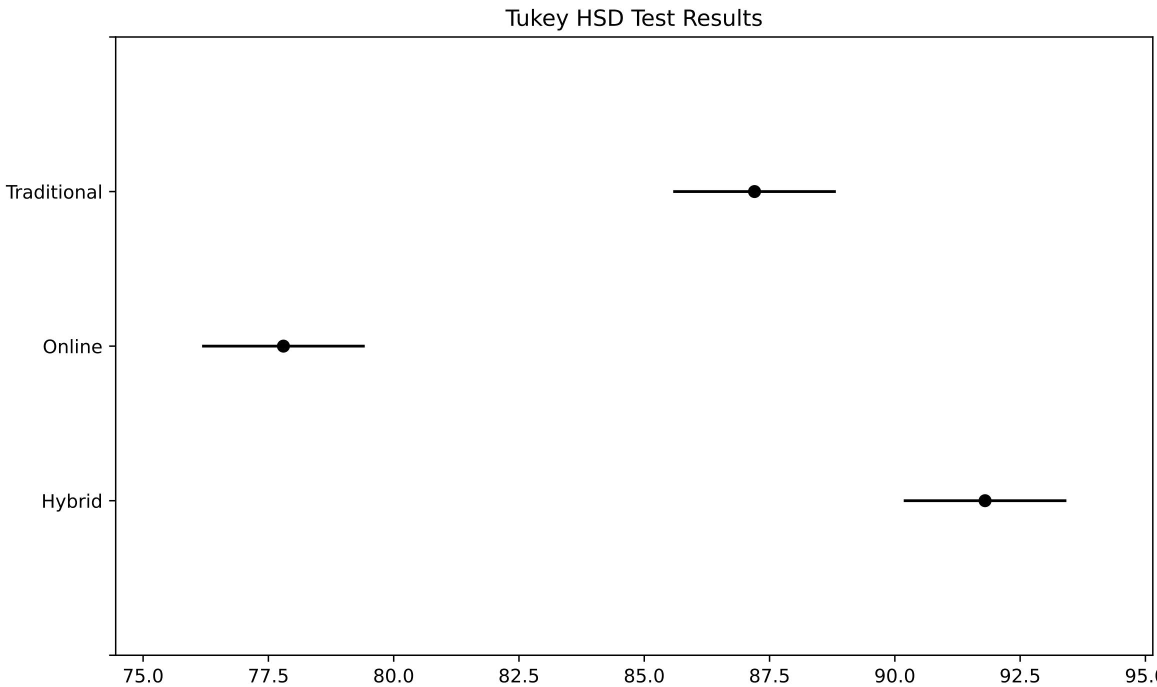
```
Reject H0: At least one teaching method has a different mean score.
```

0.3.1 Tukey's HSD Post-Hoc Test

After performing an ANOVA test, if you find a significant result (e.g., a p-value less than your chosen α level), you typically need to perform post-hoc tests to determine which specific groups differ from each other. A commonly used post-hoc test is the Tukey's Honest Significant Difference (HSD) Test:

```python
from scipy.stats import f_oneway
import matplotlib.pyplot as plt
import pandas as pd
from statsmodels.stats.multicomp import pairwise_tukeyhsd

 # Combine data into a DataFrame for Tukey's HSD
all_data = traditional + online + hybrid
groups = ['Traditional'] * len(traditional) + ['Online'] * len(online) + ['Hybrid'] * len(hybrid)
df = pd.DataFrame({'Score': all_data, 'Group': groups})

# Perform Tukey's HSD
tukey = pairwise_tukeyhsd(endog=df['Score'], groups=df['Group'], alpha=0.05)
print(tukey)

# Plot Tukey's results
tukey.plot_simultaneous()
plt.title('Tukey HSD Test Results')
plt.show()
```

```
    Multiple Comparison of Means - Tukey HSD, FWER=0.05
===========================================================
group1    group2   meandiff p-adj   lower    upper   reject
-----------------------------------------------------------
Hybrid      Online    -14.0    0.0 -17.2456 -10.7544   True
Hybrid Traditional     -4.6 0.0068  -7.8456  -1.3544   True
Online Traditional      9.4    0.0   6.1544  12.6456   True
-----------------------------------------------------------
```



0.4 Chi-Square Test

The Test
The Chi-Square test assesses whether there is a significant association between two categorical variables.

Usage

- Testing independence between two variables (e.g., gender and product preference).
- Goodness-of-fit testing (e.g., observed vs. expected distribution).

Formula

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

where:
- O: Observed frequency
- E: Expected frequency

Python Function

```python
from scipy.stats import chi2_contingency
```

Example

- **Scenario**: A company surveys customers to determine if gender influences product preference. The contingency table is:

	Prefer	Do Not Prefer	Total
Male	30	10	40
Female	25	35	60
Total	55	45	100

- **Null Hypothesis** (H_0): Gender and product preference are independent.
- **Alternative Hypothesis** (H_1): Gender and product preference are not independent.

Python Code

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

# Data
contingency_table = np.array([[30, 10], [25, 35]])

# Perform chi-square test
chi2, p_value, dof, expected = chi2_contingency(contingency_table)
print(f"Chi-square: {chi2}, P-value: {p_value}")

# Conclusion
if p_value < 0.05:
    print("Reject H0: Gender and product preference are not independent.")
else:
    print("Fail to reject H0: Gender and product preference are independent.")
```

```
Chi-square: 9.46969696969697, P-value: 0.0020889387721520535
Reject H0: Gender and product preference are not independent.
```

0.5 Pearson's Correlation Test

The Test
Pearson's correlation measures the strength and direction of the linear relationship between two continuous variables.

Usage
- Evaluate the linear relationship between two variables.
- Assumes normally distributed variables with no significant outliers.

Assumptions
1. Both variables are continuous.
2. The relationship is linear.
3. Variables are normally distributed.
4. No significant outliers.

Formula

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \cdot \sum(y_i - \bar{y})^2}}$$

0.5.1 How to calculate the p-value:

1. **Compute the test statistic** (t):

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

   where:

   - r: Pearson correlation coefficient
   - n: Number of observations

2. **Degrees of Freedom** (df):

$$df = n - 2$$

3. **Compute the p-value using the t-distribution:**

   - A two-tailed p-value is calculated as:

$$\text{p-value} = 2 \cdot (1 - \text{CDF}_t(t, df))$$

     where CDF_t is the cumulative distribution function of the t-distribution.

Python Function

```python
from scipy.stats import pearsonr
```

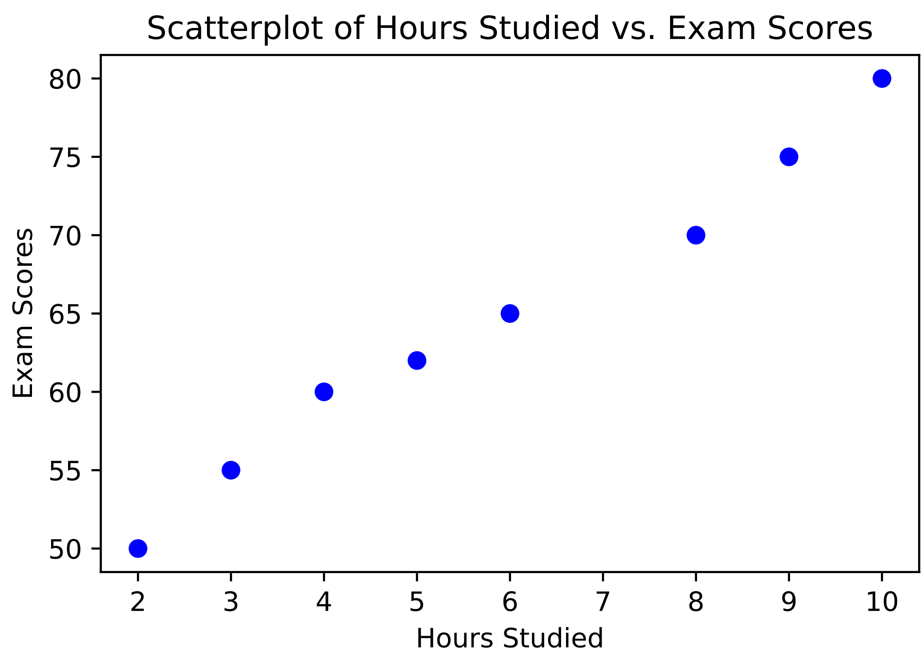
Example
- **Scenario**: Investigate the relationship between hours studied and exam scores.
- **Hours Studied**: [2, 3, 4, 5, 6, 8, 9, 10]
- **Exam Scores**: [50, 55, 60, 62, 65, 70, 75, 80]

- **Null Hypothesis** (H_0): No correlation between hours studied and exam scores.
- **Alternative Hypothesis** (H_1): Significant correlation exists between hours studied and exam scores.

Python Code

```python
from scipy.stats import pearsonr
import matplotlib.pyplot as plt

# Data
hours_studied = [2, 3, 4, 5, 6, 8, 9, 10]
exam_scores = [50, 55, 60, 62, 65, 70, 75, 80]

# Pearson correlation
pearson_corr, p_value = pearsonr(hours_studied, exam_scores)
print(f"Pearson's r: {pearson_corr}, P-value: {p_value}")

# Scatterplot
plt.scatter(hours_studied, exam_scores, color="blue")
plt.title('Scatterplot of Hours Studied vs. Exam Scores')
plt.xlabel('Hours Studied')
plt.ylabel('Exam Scores')
plt.show()

# Conclusion
if p_value < 0.05:
    print("Reject H0: Significant correlation between hours studied and exam
     ↪  scores.")
else:
    print("Fail to reject H0: No significant correlation between hours
     ↪  studied and exam scores.")
```

```
Pearson's r: 0.9925428849571527, P-value: 1.0309092235077127e-06
```




```
Reject H0: Significant correlation between hours studied and exam scores.
```

0.6 Test Selection Guide

Use the following table as a quick reference for selecting an appropriate test based on data type and analysis requirements:

Test Name	Use Case	Data Type	Groups Compared	Python Function
z-Test	Compare sample mean to population mean	Continuous	Single sample	`ztest`
t-Test	Compare two group means	Continuous	Two independent	`ttest_ind`
Paired t-Test	Compare paired measurements	Continuous	Two paired samples	`ttest_rel`
Chi-Square	Test independence of categorical variables	Categorical	Multiple categories	`chi2_contingency`
ANOVA	Compare three or more group means	Continuous	Three or more groups	`f_oneway`

Test Name	Use Case	Data Type	Groups Compared	Python Function
Tukey's HSD	Compare all pairs of group means after ANOVA	Continuous	Three or more groups	`pairwise_tukeyhsd`
Pearson Correlation	Assess linear relationship	Continuous	Two variables	`pearsonr`
