# A/B Testing and Hypothesis Testing Practice Questions with Solutions

---

## \*\*Problem 1: A/B Testing - Time Spent\*\*

\*\*Scenario:\*\* A streaming platform launched a new version (B) of its homepage to see if it increases user engagement measured by time spent. You're tasked to determine if there's a statistically significant difference in the average time spent between version A and version B.

### \*\*Hypotheses:\*\*

- H0: µ\_A = µ\_B (No difference in average time spent)

- H1: µ\_A ≠ µ\_B (There is a difference)

### \*\*Test Used:\*\* Independent t-test

### \*\*Python Solution:\*\*

```python

from scipy import stats

t\_stat, p\_val = stats.ttest\_ind(

df['total\_user\_time\_spent\_in\_mins'],

df['user\_time\_spent\_versionB\_in\_mins'],

equal\_var=False

)

print(f"T-statistic: {t\_stat}, P-value: {p\_val}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject the null hypothesis and conclude that the versions have significantly different user time spent.

---

## \*\*Problem 2: Chi-Square Test - Ad Clicks Proportions\*\*

\*\*Scenario:\*\* Do users click ads at the same rate in version A and version B?

### \*\*Hypotheses:\*\*

- H0: The proportion of ads clicked is the same across versions.

- H1: The proportion of ads clicked differs.

### \*\*Test Used:\*\* Chi-square test

### \*\*Python Solution:\*\*

```python

from scipy.stats import chi2\_contingency

import pandas as pd

obs = pd.DataFrame({

'Version A': df['ads\_clicked'],

'Version B': df['ads\_clicked\_versionB']

})

chi2, p, dof, ex = chi2\_contingency(obs.T)

print(f"Chi-square: {chi2}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject H0 and conclude that the click rates differ significantly across versions.

---

## \*\*Problem 3: Z-Test - Ads Watched\*\*

\*\*Scenario:\*\* You want to test if the average ads watched differs significantly between version A and version B.

### \*\*Hypotheses:\*\*

- H0: µ\_A = µ\_B

- H1: µ\_A ≠ µ\_B

### \*\*Test Used:\*\* Z-test

### \*\*Python Solution:\*\*

```python

from statsmodels.stats.weightstats import ztest

z\_stat, p\_val = ztest(

df['total\_ads\_watched\_in\_mins'],

df['ads\_watched\_vesionB\_in\_mins']

)

print(f"Z-statistic: {z\_stat}, P-value: {p\_val}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject H0 and infer that the average number of ads watched differs between the versions.

---

## \*\*Problem 4: One-way ANOVA - Time Slot Analysis\*\*

\*\*Scenario:\*\* Determine if there is a significant difference in user time spent across different time slots.

### \*\*Hypotheses:\*\*

- H0: All time slots have the same average user time spent.

- H1: At least one time slot differs.

### \*\*Test Used:\*\* ANOVA

### \*\*Python Solution:\*\*

```python

from scipy import stats

anova\_result = stats.f\_oneway(

\*[df[df['time\_slot'] == slot]['total\_user\_time\_spent\_in\_mins'] for slot in df['time\_slot'].unique()]

)

print(f"F-statistic: {anova\_result.statistic}, P-value: {anova\_result.pvalue}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, we conclude that there is a significant difference in average user time spent across time slots.

---

## \*\*Problem 5: Paired T-Test - Matched Data for Same Time Slot\*\*

\*\*Scenario:\*\* Check if there is a difference in user time spent in version A and B for the same time slot.

### \*\*Hypotheses:\*\*

- H0: Mean difference = 0

- H1: Mean difference ≠ 0

### \*\*Test Used:\*\* Paired t-test

### \*\*Python Solution:\*\*

```python

from scipy import stats

paired\_t\_result = stats.ttest\_rel(

df['total\_user\_time\_spent\_in\_mins'],

df['user\_time\_spent\_versionB\_in\_mins']

)

print(f"Paired T-statistic: {paired\_t\_result.statistic}, P-value: {paired\_t\_result.pvalue}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject H0, indicating that version A and B have statistically different average user time spent.

---

## \*\*Problem 6: One-Sample T-Test - Time Spent vs Benchmark\*\*

\*\*Scenario:\*\* A product manager claims that the average time spent on the site per user is at least 300,000 mins. Test this using version A data.

### \*\*Hypotheses:\*\*

- H0: µ = 300,000

- H1: µ ≠ 300,000

### \*\*Test Used:\*\* One-sample t-test

### \*\*Python Solution:\*\*

```python

from scipy.stats import ttest\_1samp

result = ttest\_1samp(df['total\_user\_time\_spent\_in\_mins'], 300000)

print(f"T-statistic: {result.statistic}, P-value: {result.pvalue}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject H0 and conclude that the average differs from the benchmark.

---

## \*\*Problem 7: Two-Sample T-Test - Ads Watched Across Time Slots\*\*

\*\*Scenario:\*\* Is there a difference in ads watched between morning and evening time slots?

### \*\*Hypotheses:\*\*

- H0: µ\_morning = µ\_evening

- H1: µ\_morning ≠ µ\_evening

### \*\*Test Used:\*\* Independent t-test

### \*\*Python Solution:\*\*

```python

morning = df[df['time\_slot'] == '06:00-11:59']['total\_ads\_watched\_in\_mins']

evening = df[df['time\_slot'] == '18:00-23:59']['total\_ads\_watched\_in\_mins']

result = stats.ttest\_ind(morning, evening, equal\_var=False)

print(f"T-statistic: {result.statistic}, P-value: {result.pvalue}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude that ad views significantly differ between these slots.

---

## \*\*Problem 8: Proportion Z-Test - Ad Click Rate\*\*

\*\*Scenario:\*\* Test if the proportion of ads clicked to ads watched is higher in version B.

### \*\*Hypotheses:\*\*

- H0: p\_A = p\_B

- H1: p\_A < p\_B

### \*\*Test Used:\*\* Z-test for proportions

### \*\*Python Solution:\*\*

```python

from statsmodels.stats.proportion import proportions\_ztest

clicks = [df['ads\_clicked'].sum(), df['ads\_clicked\_versionB'].sum()]

exposures = [df['total\_ads\_watched\_in\_mins'].sum(), df['ads\_watched\_vesionB\_in\_mins'].sum()]

z\_stat, p\_val = proportions\_ztest(clicks, exposures, alternative='smaller')

print(f"Z-statistic: {z\_stat}, P-value: {p\_val}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude that version B has a higher ad click rate.

---

## \*\*Problem 9: Levene’s Test - Homogeneity of Variance\*\*

\*\*Scenario:\*\* Before performing a t-test, check if the variance of time spent between versions A and B is equal.

### \*\*Hypotheses:\*\*

- H0: Variances are equal

- H1: Variances are not equal

### \*\*Test Used:\*\* Levene’s Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import levene

stat, p = levene(df['total\_user\_time\_spent\_in\_mins'], df['user\_time\_spent\_versionB\_in\_mins'])

print(f"Levene's stat: {stat}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude variances are significantly different.

---

## \*\*Problem 10: Welch’s ANOVA - Unequal Variance Between Groups\*\*

\*\*Scenario:\*\* Time spent differs by time slots, but variances may be unequal.

### \*\*Hypotheses:\*\*

- H0: All time slots have same mean time spent

- H1: At least one slot differs

### \*\*Test Used:\*\* Welch’s ANOVA

### \*\*Python Solution:\*\*

```python

import pingouin as pg

welch = pg.welch\_anova(dv='total\_user\_time\_spent\_in\_mins', between='time\_slot', data=df)

print(welch)

```

### \*\*Statistical Inference:\*\*

If p < 0.05, there is a significant difference in means across groups with unequal variances.

---

## \*\*Problem 11: Mann-Whitney U Test - Non-parametric Test\*\*

\*\*Scenario:\*\* Check difference in ad clicks between A and B when normality is not assumed.

### \*\*Hypotheses:\*\*

- H0: Median of A = Median of B

- H1: Medians are different

### \*\*Test Used:\*\* Mann-Whitney U Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import mannwhitneyu

stat, p = mannwhitneyu(df['ads\_clicked'], df['ads\_clicked\_versionB'])

print(f"U-statistic: {stat}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude median ad clicks differ.

---

## \*\*Problem 12: Kruskal-Wallis Test - Non-parametric ANOVA\*\*

\*\*Scenario:\*\* Determine if time spent across time slots differs without assuming normality.

### \*\*Hypotheses:\*\*

- H0: All time slots have same distribution

- H1: At least one differs

### \*\*Test Used:\*\* Kruskal-Wallis Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import kruskal

groups = [df[df['time\_slot'] == slot]['total\_user\_time\_spent\_in\_mins'] for slot in df['time\_slot'].unique()]

stat, p = kruskal(\*groups)

print(f"Kruskal-Wallis stat: {stat}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, infer significant distribution differences.

---

## \*\*Problem 13: Correlation Test - Time vs Clicks\*\*

\*\*Scenario:\*\* Examine if user time spent is correlated with ads clicked.

### \*\*Hypotheses:\*\*

- H0: No correlation

- H1: Correlation exists

### \*\*Test Used:\*\* Pearson Correlation

### \*\*Python Solution:\*\*

```python

corr, p = stats.pearsonr(df['total\_user\_time\_spent\_in\_mins'], df['ads\_clicked'])

print(f"Correlation: {corr}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude significant correlation.

---

## \*\*Problem 14: Spearman Correlation - Non-parametric Correlation\*\*

\*\*Scenario:\*\* Assess monotonic relationship between ad clicks and ads watched.

### \*\*Hypotheses:\*\*

- H0: No monotonic correlation

- H1: Monotonic correlation exists

### \*\*Test Used:\*\* Spearman Correlation

### \*\*Python Solution:\*\*

```python

from scipy.stats import spearmanr

corr, p = spearmanr(df['ads\_clicked'], df['total\_ads\_watched\_in\_mins'])

print(f"Spearman correlation: {corr}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, conclude significant monotonic correlation.

---

## \*\*Problem 15: Shapiro-Wilk Test - Normality of Time Spent\*\*

\*\*Scenario:\*\* Verify if total time spent (version A) is normally distributed.

### \*\*Hypotheses:\*\*

- H0: Data is normally distributed

- H1: Data is not normally distributed

### \*\*Test Used:\*\* Shapiro-Wilk Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import shapiro

stat, p = shapiro(df['total\_user\_time\_spent\_in\_mins'])

print(f"Shapiro-Wilk statistic: {stat}, P-value: {p}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, reject H0 and conclude data is not normal.

---

## \*\*Problem 16: Anderson-Darling Test - Distribution Test\*\*

\*\*Scenario:\*\* Confirm if ads watched (version B) follow a normal distribution.

### \*\*Test Used:\*\* Anderson-Darling Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import anderson

result = anderson(df['ads\_watched\_vesionB\_in\_mins'])

print(result)

```

### \*\*Statistical Inference:\*\*

Compare test statistic to critical values. If greater, data is not normal.

---

## \*\*Problem 17: Binomial Test - Click Success Rate\*\*

\*\*Scenario:\*\* Check if ad click conversion rate is 10% in version A.

### \*\*Hypotheses:\*\*

- H0: True click rate = 0.10

- H1: True click rate ≠ 0.10

### \*\*Test Used:\*\* Binomial Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import binom\_test

clicks = df['ads\_clicked'].sum()

total = df['total\_ads\_watched\_in\_mins'].sum()

p\_val = binom\_test(clicks, total, p=0.10)

print(f"P-value: {p\_val}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, click rate differs from 10%.

---

## \*\*Problem 18: Linear Regression - Time vs Clicks\*\*

\*\*Scenario:\*\* Predict ad clicks based on time spent.

### \*\*Test Used:\*\* Simple Linear Regression

### \*\*Python Solution:\*\*

```python

import statsmodels.api as sm

X = df['total\_user\_time\_spent\_in\_mins']

y = df['ads\_clicked']

X = sm.add\_constant(X)

model = sm.OLS(y, X).fit()

print(model.summary())

```

### \*\*Statistical Inference:\*\*

Check p-value of the coefficient. If < 0.05, time is a significant predictor.

---

## \*\*Problem 19: Tukey's HSD - Post Hoc Test After ANOVA\*\*

\*\*Scenario:\*\* Determine which time slots differ in time spent after ANOVA.

### \*\*Test Used:\*\* Tukey's HSD

### \*\*Python Solution:\*\*

```python

from statsmodels.stats.multicomp import pairwise\_tukeyhsd

posthoc = pairwise\_tukeyhsd(

df['total\_user\_time\_spent\_in\_mins'],

df['time\_slot']

)

print(posthoc)

```

### \*\*Statistical Inference:\*\*

Groups with p < 0.05 are significantly different.

---

## \*\*Problem 20: Friedman Test - Repeated Measures\*\*

\*\*Scenario:\*\* Evaluate if repeated ad exposure (simulated as rows) has different engagement.

### \*\*Test Used:\*\* Friedman Test

### \*\*Python Solution:\*\*

```python

from scipy.stats import friedmanchisquare

result = friedmanchisquare(

df['total\_user\_time\_spent\_in\_mins'],

df['user\_time\_spent\_versionB\_in\_mins'],

df['total\_ads\_watched\_in\_mins']

)

print(f"Friedman stat: {result.statistic}, P-value: {result.pvalue}")

```

### \*\*Statistical Inference:\*\*

If p < 0.05, engagement differs significantly across measures.

---

# ✅ All 20 practice problems are now included!