Task-4: A/B Testing Workflow in Python

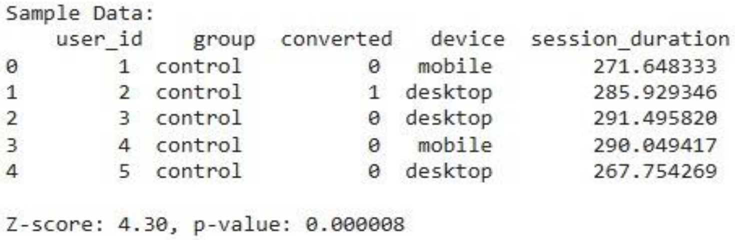
```
# --------------------------------


import pandas as pd

import numpy as np

import scipy.stats as stats

import matplotlib.pyplot as plt

from statsmodels.stats.power import TTestIndPower

from statsmodels.stats.proportion import proportions_ztest, proportion_confint # Import from statsmodels


# ----------------------------

# Step 1: Load Dataset

# ----------------------------

# If you have the Kaggle dataset:

# ab_data = pd.read_csv("ab_test_data.csv")


# For demonstration, simulate dataset similar to PDF

np.random.seed(42)

ab_data = pd.DataFrame({

    "user_id": range(1, 20001),

    "group": ["control"]*10000 + ["treatment"]*10000,

    "converted": np.concatenate([

        np.random.binomial(1, 0.075, 10000),   # control ~ 7.5% conversion

        np.random.binomial(1, 0.09, 10000)    # treatment ~ 9.0% conversion
    ]),
    "device": np.random.choice(["desktop", "mobile"], size=20000, p=[0.6, 0.4]),
    "session_duration": np.random.normal(300, 50, 20000)  # ~ 5 min avg
})

print("Sample Data:\n", ab_data.head())

# -----------------------------
# Step 2: Hypotheses
# -----------------------------
# H0: New design conversion rate ≤ Old design
# H1: New design conversion rate > Old design

# -----------------------------
# Step 3: Two-Proportion Z-Test
# -----------------------------
conv_old = ab_data[ab_data['group'] == 'control']['converted'].sum()
conv_new = ab_data[ab_data['group'] == 'treatment']['converted'].sum()
n_old = ab_data[ab_data['group'] == 'control'].shape[0]
n_new = ab_data[ab_data['group'] == 'treatment'].shape[0]

z_score, p_value = proportions_ztest( # Use the imported function directly
    [conv_new, conv_old],
    [n_new, n_old],
    alternative='larger'
)
```

```
print(f"\nZ-score: {z_score:.2f}, p-value: {p_value:.6f}")

# -----------------------------
# Step 4: Confidence Intervals
# -----------------------------
ci_old = proportion_confint(conv_old, n_old, alpha=0.05, method='normal')
ci_new = proportion_confint(conv_new, n_new, alpha=0.05, method='normal')

# Plot confidence intervals
plt.figure(figsize=(8,6))
plt.errorbar(x=[0,1],
    y=[conv_old/n_old, conv_new/n_new],
    yerr=[[conv_old/n_old-ci_old[0], conv_new/n_new-ci_new[0]],
       [ci_old[1]-conv_old/n_old, ci_new[1]-conv_new/n_new]],
    fmt='o', capsize=10, color="blue")

plt.xticks([0,1], ["Control", "Treatment"])
plt.ylabel("Conversion Rate")
plt.title("95% Confidence Intervals for Conversion Rates")
plt.grid(alpha=0.3)
plt.show()

# -----------------------------
# Step 5: Chi-Square Test
# -----------------------------
contingency_table = pd.crosstab(ab_data['device'], ab_data['converted'])
```

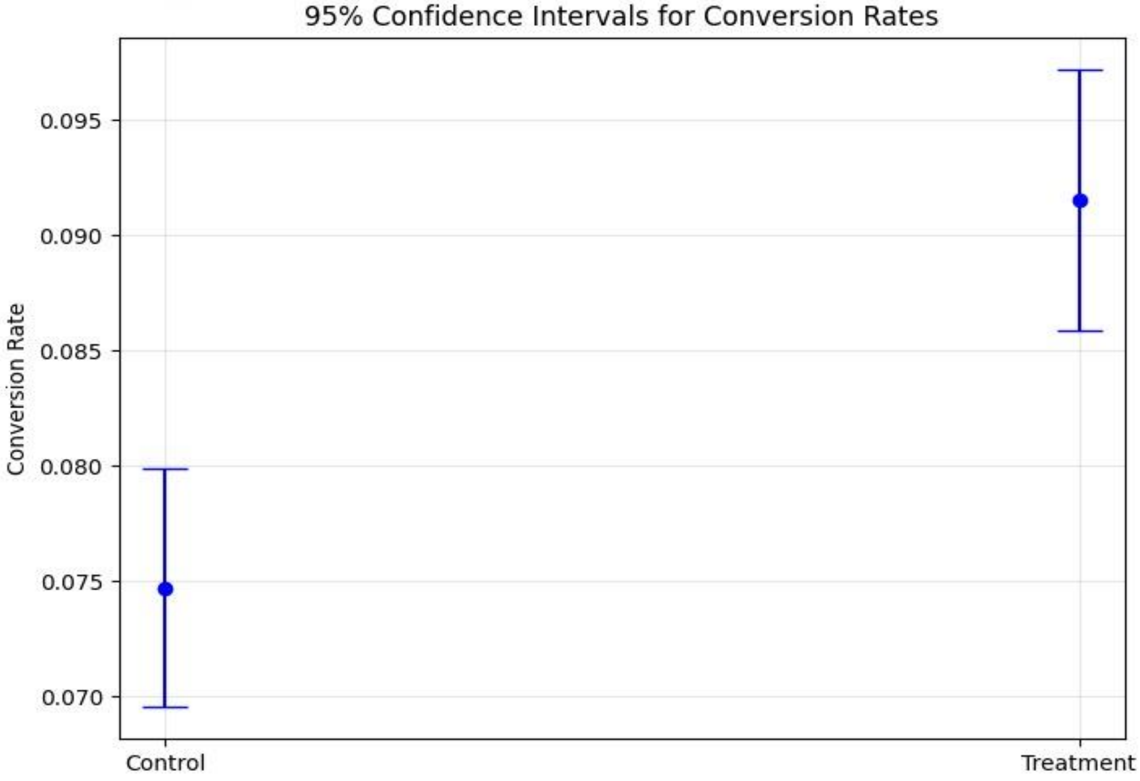
```
chi2, p_chi, dof, _ = stats.chi2_contingency(contingency_table)
print(f"\nChi-Square Test p-value (Device vs Conversion): {p_chi:.5f}")

# ----------------------------
# Step 6: T-Test on Session Duration
# ----------------------------
duration_control = ab_data[ab_data['group'] == 'control']['session_duration']
duration_treatment = ab_data[ab_data['group'] == 'treatment']['session_duration']

t_stat, p_ttest = stats.ttest_ind(duration_treatment, duration_control)
print(f"T-test p-value (Session Duration): {p_ttest:.4f}")

# ----------------------------
# Step 7: Sample Size Calculation
# ----------------------------
effect_size = 0.2  # minimum detectable effect
power = 0.8
analysis = TTestIndPower()
sample_size = analysis.solve_power(effect_size, power=power, alpha=0.05)
print(f"\nRequired sample per group (for effect size=0.2): {int(sample_size)}")

# ----------------------------
# Step 8: Business Reporting
# ----------------------------
conversion_control = conv_old / n_old
conversion_treatment = conv_new / n_new
```

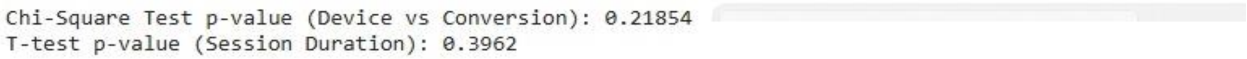
```
lift = (conversion_treatment - conversion_control) * 100

print("\n--- Business Reporting ---")
print(f"1. New design increased conversions by {lift:.2f}%")
print(f"2. Statistical significance: p = {p_value:.6f}")
if p_value < 0.05:
    print("✅ Recommendation: Roll out new design")
else:
    print("❌ No strong evidence yet, continue testing")
print(f"3. Device effect check (Chi-Square): p = {p_chi:.5f}")
print(f"4. Session duration unaffected (T-Test p = {p_ttest:.4f})")
```

Output:

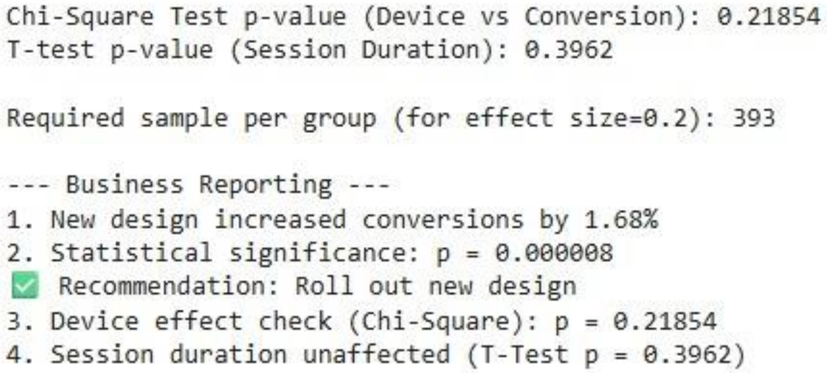


```
Sample Data:
   user_id    group  converted   device  session_duration
0        1  control          0   mobile        271.648333
1        2  control          1  desktop        285.929346
2        3  control          0  desktop        291.495820
3        4  control          0   mobile        290.049417
4        5  control          0  desktop        267.754269

Z-score: 4.30, p-value: 0.000008
```





```
Chi-Square Test p-value (Device vs Conversion): 0.21854
T-test p-value (Session Duration): 0.3962
```



```
Chi-Square Test p-value (Device vs Conversion): 0.21854
T-test p-value (Session Duration): 0.3962
```

```
Required sample per group (for effect size=0.2): 393

--- Business Reporting ---
1. New design increased conversions by 1.68%
2. Statistical significance: p = 0.000008
✅ Recommendation: Roll out new design
3. Device effect check (Chi-Square): p = 0.21854
4. Session duration unaffected (T-Test p = 0.3962)
```