PSET8

April 23, 2025

0.0.1 1. Data Preparation:

Load data from customer_feedback.csv and sales_data.csv into pandas data frames.

Convert the ‘date’ column in both datasets to pandas datetime objects for analysis.

[1]:
```python
import pandas as pd
import numpy as np
from scipy import stats

#load data
feedback_df = pd.read_csv('customer_feedback.csv')
sales_df = pd.read_csv('sales_data.csv')

#convert 'date' to datetime format
feedback_df['date'] = pd.to_datetime(feedback_df['date'])
sales_df['date'] = pd.to_datetime(sales_df['date'])

#show info
print("Customer Feedback Data Shape:", feedback_df.shape)
print(feedback_df.head())
print(feedback_df.tail())

print("\nSales Data Shape:", sales_df.shape)
print(sales_df.head())
print(sales_df.tail())
```

```
Customer Feedback Data Shape: (500, 3)
        date  product  feedback_score
0 2023-02-22      iOS               5
1 2023-05-22  Android               2
2 2022-11-22      iOS               2
3 2022-11-26  Android              10
4 2023-04-26      iOS               1
          date  product  feedback_score
495 2023-03-18  Android               5
496 2023-02-05      iOS               7
497 2023-03-29  Android               5
498 2023-02-08      iOS               3
499 2022-11-28  Android               9
```

```
Sales Data Shape: (500, 3)
         date  product  sales
0 2022-12-12      iOS    473
1 2022-12-12  Android    919
2 2023-06-24      iOS    805
3 2023-06-24  Android    996
4 2023-10-20      iOS    792
           date  product  sales
495 2023-06-28  Android    446
496 2023-01-23      iOS    247
497 2023-01-23  Android    373
498 2022-12-22      iOS    816
499 2022-12-22  Android    913
```

0.0.2 2. Customer Feedback Analysis (feedback_analysis function):

Determine the appropriate T-Test (independent, paired, one-tail, or two-tail) based on data and hypothesis.

Convert the feedback scores into a 2-dimensional numpy array (iOS scores as the first dimension, Android scores as the second).

Compute and return the t-test statistic and the pvalue and print the returned values.

Analyze if there's a significant difference in average customer satisfaction between iOS and Android apps.

Interpret the result, i.e., if the p-value is significant for average customer satisfaction between the two groups.

[ ]:
```
#function: Feedback Analysis
def feedback_analysis(df_feedback):
    #separate scores by platform
    ios_scores = df_feedback[df_feedback['product'] == 'iOS']['feedback_score'].
 ↪values
    android_scores = df_feedback[df_feedback['product'] ==␣
 ↪'Android']['feedback_score'].values

    #perform two-sample independent t-test
    statistic, p_val = stats.ttest_ind(ios_scores, android_scores,␣
 ↪equal_var=False)

    print("Feedback analysis t-test statistic:", statistic)
    print("Feedback analysis pvalue:", p_val)

    return statistic, p_val

#run feedback analysis
feedback_stat, feedback_p = feedback_analysis(feedback_df)
```

```
Feedback analysis t-test statistic: 1.9033888211703986
Feedback analysis pvalue: 0.05756609386358278
```

0.0.3 Interpretation: Feedback Analysis

The t-test compares feedback scores between iOS and Android users.

- Test Statistic: 1.9033888211703986
- P-value: 0.05756609386358278

Since the p-value is greater than 0.05, we fail to reject the null hypothesis. This means we **do not have sufficient evidence** to say that average customer satisfaction is significantly different between iOS and Android apps.

0.0.4 3. Sales Performance Analysis (sales_analysis function):

Compare sales before and after a major marketing campaign (March 1-31, 2023).

Use an appropriate T-Test to assess the campaign's impact on sales.

Return the t-test statistic and the p-value and print the returned values.

Interpret the result, i.e., if a significant impact is found based on the p-value;

[3]:
```python
#function: Sales Analysis
def sales_analysis(df_sales):
    # Split data before and after March 2023 campaign
    before = df_sales[df_sales['date'] < '2023-03-01']['sales'].values
    after = df_sales[df_sales['date'] > '2023-03-31']['sales'].values

    # Perform independent t-test
    statistic, p_val = stats.ttest_ind(before, after, equal_var=False)

    print("Sales analysis t-test statistic:", statistic)
    print("Sales analysis pvalue:", p_val)

    return statistic, p_val

# Run sales analysis
sales_stat, sales_p = sales_analysis(sales_df)
```

```
Sales analysis t-test statistic: 0.16642710322927962
Sales analysis pvalue: 0.8679489234386756
```

0.0.5 Interpretation: Sales Analysis

The t-test compares sales figures before and after the marketing campaign in March 2023.

- Test Statistic: 0.16642710322927962
- P-value: 0.8679489234386756

Since the p-value is significantly greater than 0.05, we fail to reject the null hypothesis. This means we **do not have sufficient evidence** that the marketing campaign had a measurable impact on sales.

0.0.6 4. Seasonal Sales Analysis (seasonal_analysis function):

Examine sales differences between summer (June-August) and winter (December-February).

Apply a T-Test to assess if these variations are statistically significant.

Return the t-test statistic and the p-value and print the returned values.

Interpret the result, i.e., if significant seasonal variations exists based on the p-value.

[4]:
```
#function: Seasonal Sales Analysis
def seasonal_analysis(df_sales):
    #define summer and winter months
    summer_months = [6, 7, 8]
    winter_months = [12, 1, 2]

    summer_sales = df_sales[df_sales['date'].dt.month.
 ↪isin(summer_months)]['sales'].values
    winter_sales = df_sales[df_sales['date'].dt.month.
 ↪isin(winter_months)]['sales'].values

    #perform independent t-test
    statistic, p_val = stats.ttest_ind(summer_sales, winter_sales,␣
 ↪equal_var=False)

    print("Seasonal analysis t-test statistic:", statistic)
    print("Seasonal analysis pvalue:", p_val)

    return statistic, p_val

#run seasonal analysis
seasonal_stat, seasonal_p = seasonal_analysis(sales_df)
```

```
Seasonal analysis t-test statistic: 0.09956961638905915
Seasonal analysis pvalue: 0.9207644588060664
```

0.0.7 Interpretation: Seasonal Sales Analysis

This t-test compares sales between the summer months (June-August) and winter months (December-February).

- Test Statistic: 0.09956961638905915
- P-value: 0.9207644588060664

Since the p-value is greater than 0.05, we fail to reject the null hypothesis. This means we **do not have sufficient evidence** of seasonal variation in sales between summer and winter.

0.0.8 5. Feedback Consistency Analysis (consistency_analysis function):

Assess if monthly feedback scores are consistent across January, May, September, and December.

Use one-way ANOVA to test for significant differences in feedback scores across these months.

Return the statistic and the p-value and print the returned values.

Interpret the result, i.e., if the difference are significant based on p-value.

[5]:
```python
#function Feedback Consistency Analysis
def consistency_analysis(df_feedback):
    #filter months of interest
    df_feedback['month'] = df_feedback['date'].dt.month
    selected_months = {1: 'Jan', 5: 'May', 9: 'Sep', 12: 'Dec'}
    filtered = df_feedback[df_feedback['month'].isin(selected_months.keys())]

    # Group feedback scores by month
    jan = filtered[filtered['month'] == 1]['feedback_score'].values
    may = filtered[filtered['month'] == 5]['feedback_score'].values
    sep = filtered[filtered['month'] == 9]['feedback_score'].values
    dec = filtered[filtered['month'] == 12]['feedback_score'].values

    #perform one-way ANOVA
    statistic, p_val = stats.f_oneway(jan, may, sep, dec)

    print("Feedback consistency ANOVA statistic:", statistic)
    print("Feedback consistency pvalue:", p_val)

    return statistic, p_val

#run consistency analysis
consistency_stat, consistency_p = consistency_analysis(feedback_df)
```

```
Feedback consistency ANOVA statistic: 0.3146823675455494
Feedback consistency pvalue: 0.8147473590881886
```

0.0.9 Interpretation: Feedback Consistency Analysis

The one-way ANOVA compares customer feedback scores across the months of January, May, September, and December.

- ANOVA Statistic: 0.3146823675455494
- P-value: 0.8147473590881886

Since the p-value is greater than 0.05, we conclude that there is no statistically significant difference in average feedback scores across these months.

0.0.10 6. Sales and Feedback Correlation Analysis (corr_analysis function):

Investigate if high customer feedback correlates with increased sales.

Merge feedback and sales data, categorizing sales into high and low feedback scores.

Perform a T-Test to compare sales in months with high vs. low feedback scores.

Return the statistic and the p-value and print the returned values.

Interpret the result, i.e., if correlation is significant based on the p-value.

[ ]:
```
#function: Sales and Feedback Correlation Analysis
def corr_analysis(df_feedback, df_sales):
    #aggregate feedback scores by date
    avg_feedback = df_feedback.groupby('date')['feedback_score'].mean().
 ↪reset_index()
    avg_feedback.columns = ['date', 'avg_feedback_score']

    #aggregate sales by date
    total_sales = df_sales.groupby('date')['sales'].sum().reset_index()

    #merge both datasets on date
    merged = pd.merge(avg_feedback, total_sales, on='date')

    #label high vs. low feedback
    threshold = merged['avg_feedback_score'].median()
    high_feedback = merged[merged['avg_feedback_score'] > threshold]['sales']
    low_feedback = merged[merged['avg_feedback_score'] <= threshold]['sales']

    #perform t-test
    statistic, p_val = stats.ttest_ind(high_feedback, low_feedback,␣
 ↪equal_var=False)

    print("Correlation analysis t-test statistic:", statistic)
    print("Correlation analysis pvalue:", p_val)

    return statistic, p_val

#run correlation analysis
corr_stat, corr_p = corr_analysis(feedback_df, sales_df)
```

```
Correlation analysis t-test statistic: 0.5288478099451596
Correlation analysis pvalue: 0.5978430534511507
```

0.0.11 Interpretation: Sales and Feedback Correlation Analysis

This t-test evaluates whether higher average feedback scores are associated with significantly different sales performance.

- Test Statistic: 0.5288478099451596
- P-value: 0.5978430534511507

Since the p-value is greater than 0.05, we fail to reject the null hypothesis. There is **no significant evidence** that higher feedback scores correlate with higher sales.