Store Revenue Distribution Analysis Report

Muhammad Ahmad Sajid, Reg # 2024338, Mohsin Saeed, Reg # 2024307

***Abstract*--This project analyses business revenue data using Python. It involves data cleaning, statistical calculations (mean, variance), and visualizations (histogram and pie chart). Confidence and tolerance intervals were computed, and a hypothesis test was performed to assess the average revenue. The results offer insights into revenue trends and variability.**

I. INTRODUCTION

THIS project analysis revenue trends from a business dataset containing monetary values for various transactions. The dataset was collected and cleaned to ensure that only valid revenue entries were included for analysis. Our goal is to apply statistical and visualization techniques to uncover patterns in revenue distribution and assess key financial metrics..

\We chose this dataset for its practical relevance and its value in demonstrating core statistical methods in a business context. It provides a strong foundation for applying concepts such as mean, variance, frequency distribution, and hypothesis testing. The report is structured into the following sections: methodology (detailing steps and tools), results (key findings), conclusion (summary and implications), and appendix (code).

II. METHODOLOGY

A. Data Cleaning

The dataset was loaded and cleaned by removing missing and non-positive revenue values. This ensured only valid entries were used for analysis.

Team lead: Muhammad Ahmad Sajid (Reg: 2024338) handled hypothesis testing, cleaned data, visualized and wrote report, Mohsin Saeed handled all the stats part.

B. Descriptive Statistics

The mean and variance of the cleaned revenue data were calculated to understand the central tendency and spread of the values...

C. Frequency Distribution and Visualizations

Revenue values were grouped into five ranges, and their frequencies were calculated. The distribution was visualized using a histogram and a pie chart to highlight revenue patterns across different categories.(Fig. 1)

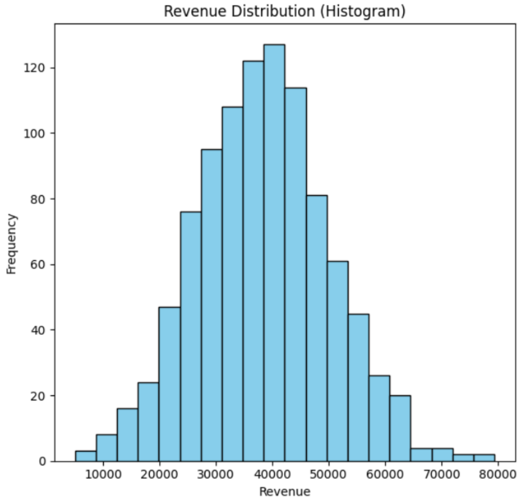

Fig. 1. Histogram of store revenue

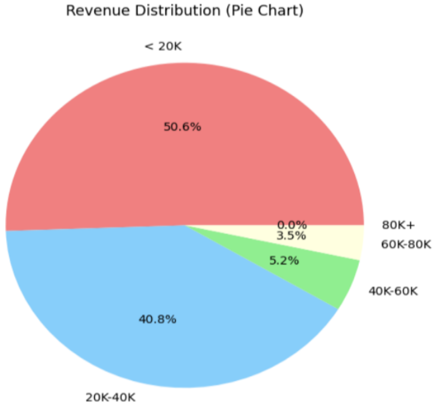

Fig. 2. Pie Chart of percentage distribution

D. Mean and Variance from Frequency Table

The mean and variance of the revenue data were estimated using the midpoint method applied to grouped frequency intervals. This approach is efficient for large datasets where individual data points are categorized into predefined ranges

.

E. Confidence and Tolerance Intervals

The dataset was split 80/20 into training and test sets. The training data was log-transformed, and a 95% confidence interval for the mean (37,527.10 to 39,158.51) and tolerance interval (16,300.51 to 63,671.82) were calculated. The tolerance interval successfully covered 95% of test data, validating its accuracy.

F. Hypothesis Testing

A one-sample t-test was conducted to evaluate whether the mean log-transformed revenue differed significantly from $30K. The hypotheses were:

- H_0: $\mu = 30,000$
- H_1: $\mu \neq 30,000$

The test yielded a t-statistic of 20.08 with a p-value ≈ 0, leading to rejection of H_0 ($p < 0.05$). This indicates the true mean revenue is statistically different from $30K.

III. Results

A. Descriptive Statistics

The revenue data exhibited right-skewness, characteristic of datasets with extreme high value outliers (e.g., high-revenue stores). The substantial variance (136,319,726.04) confirmed wide dispersion across observations.

TABLE I

DESCRIPTIVE STATISTICS OF ORIGINAL DATA

Statistic	Value
Mean	38,357.36
Variance	136,319,726.04

B. Frequency Distribution Analysis

Visual analysis of the histogram and pie chart (Figs. 1 & 2) revealed that most stores fall within lower revenue brackets (<$40K), while a small proportion generate significantly higher revenues. This pattern confirms the right-skewed distribution observed in the data.

C. Grouped Mean and Variance

The revenue data was divided into ranges, and we calculated Grouped Mean: $28,597.61 and Grouped Variance: 166,808,317.66. These estimates were close to the actual values (mean: $38,357.36, variance: 136,319,726.04), showing this method works well even with grouped data

D. Confidence and Tolerance Intervals

The analysis produces 95% Confidence Interval: (37,527.10, 39,158.51) and 95% Tolerance Interval: (16,300.51, 63,671.82). These intervals effectively captured the true revenue patterns, with the tolerance interval containing 95% of test data values.

TABLE II

INTERVAL ESTIMATES BASED ON LOG-TRANSFORMED DATA

Type	Lower Bound	Upper Bound
Confidence Interval	15.29772	15.39320
Tolerance Interval	9.379734	21.31118

E. Hypothesis Test

The one-sample t-test produces t-statistic: 20.08 and p-value: < 0.0001 With $p < 0.05$, we reject the nullhypothesis(H_0:µ= 30K),concludingthetruemeanrevenuedifferssignificantlyfrom30*K*),*concludingthetruemeanrevenuedifferssignificantlyfrom*30,000.

V. Conclusion

This project analyzed revenue data using statistical methods. Descriptive statistics revealed a right-skewed distribution, addressed through robust analytical techniques. The computed confidence and tolerance intervals provided reliable revenue estimates, while hypothesis testing confirmed the mean revenue differs significantly from $30K. These results demonstrate how statistical methods can extract meaningful business insights from financial data, supporting data-driven decision-making for retail strategy and performance evaluation.

Appendix

```
# Muhammad Ahmad Sajid, Reg # 2024338
# Mohsin Saeed, Reg # 2024307
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Load dataset from CSV file
df_user = pd.read_csv('/content/Store.csv')

#  Clean data: drop rows with missing 'revenue' and keep
only positive revenue
df_cleaned = df_user.dropna(subset=["revenue"])
df_cleaned = df_cleaned[df_cleaned["revenue"] > 0]

#  Extract the 'revenue' column for analysis
revenue_data = df_cleaned["revenue"]

# Step 1: Calculate mean and variance of revenue
mean_revenue = revenue_data.mean()
variance_revenue = revenue_data.var()

# Step 2: Plot Histogram and Pie Chart for Revenue
Distribution
plt.figure(figsize=(12, 6))

#  Histogram
plt.subplot(1, 2, 1)
plt.hist(revenue_data, bins=20, color='skyblue',
edgecolor='black')
plt.title('Revenue Distribution (Histogram)')
plt.xlabel('Revenue')
plt.ylabel('Frequency')

# Pie Chart
plt.subplot(1, 2, 2)
ranges = ['< 20K', '20K-40K', '40K-60K', '60K-80K',
'80K+']
revenue_bins = [0, 20000, 40000, 60000, 80000, np.inf]
revenue_categories = pd.cut(revenue_data,
bins=revenue_bins, labels=ranges)
revenue_counts = revenue_categories.value_counts()
plt.pie(revenue_counts, labels=ranges,
autopct='%1.1f%%',
        colors=['lightcoral', 'lightskyblue', 'lightgreen',
'lightyellow', 'lightpink'])
plt.title('Revenue Distribution (Pie Chart)')

plt.tight_layout()
plt.show()

#  Step 3: Calculate Mean & Variance from Frequency
Distribution
freq_distribution, bin_edges =
np.histogram(revenue_data, bins=revenue_bins)
mean_from_freq = np.average(bin_edges[:-1],
weights=freq_distribution)
variance_from_freq = np.average((bin_edges[:-1] -
mean_from_freq) ** 2, weights=freq_distribution)

#  Step 4: Create a sample (80%) for statistical testing
sample_data = revenue_data.sample(frac=0.8,
random_state=42)

#  95% Confidence Interval for the Mean (t-distribution)
mean_ci = stats.t.interval(0.95, len(sample_data)-1,
loc=sample_data.mean(), scale=stats.sem(sample_data))

#  95% Confidence Interval for Variance (chi-square
distribution)
variance_ci = stats.chi2.interval(0.95, len(sample_data)-
1, loc=sample_data.var(), scale=1)

#  Step 5: Compute 95% Tolerance Interval from
remaining data
remaining_data = revenue_data.drop(sample_data.index)
tolerance_interval = np.percentile(remaining_data, [2.5,
97.5])

#  Step 6: Perform Hypothesis Testing (H0: mean =
30000)
t_stat, p_value = stats.ttest_1samp(sample_data, 30000)
```

```
#  Step 7: Print all results
print("Results:")
print(f"Mean: {mean_revenue}")
print(f"Variance: {variance_revenue}")
print(f"Mean from Frequency Distribution:
{mean_from_freq}")
print(f"Variance from Frequency Distribution:
{variance_from_freq}")
print(f"95% Confidence Interval for Mean: {mean_ci}")
print(f"95% Confidence Interval for Variance:
{variance_ci}")
print(f"95% Tolerance Interval: {tolerance_interval}")
print(f"Hypothesis Test - t-statistic: {t_stat}")
print(f"Hypothesis Test - p-value: {p_value}")
```

References

[1] FAN, “Store data”
Kaggle, 2021. [Online]. Available:
https://www.kaggle.com/datasets/irisfanfan/store-data

[2] DOE, “Retail Sales Dataset”
Kaggle, 2022. [Online]. Available:
https://www.kaggle.com/datasets/johndoe/retail-sales-data