APPLE STORE REVIEWS

By Farook Mohammad

Phone Number

+91 9467671237

https://github.com/farook8090/statistics-and-machine-learning

LIBRARIES

- import pandas as pd
- import numpy as np
- import matplotlib.pyplot as plt
- import seaborn as sns

DATAFRAME

```
df=pd.read_csv('Apple_Store_Reviews.csv')
df
```

	Review_ID	App_Name	User_Age	Review_Date	Rating	Review_Text	Likes	Device_Type	Version_Used	Country	Purchase_Amount	Category
0	1	Candy Crush Saga	21	2023-01-16	4	Great game, but too many in-game purchases.	70	iPhone 12	3.231.19	Australia	0.00	Games
1	2	Spotify	57	2024-02-01	1	Good, but has connection issues sometimes.	49	iPhone SE	4.102.9	Germany	7.15	Music
2	3	TikTok	33	2023-11-30	5	Awesome app! Best entertainment content.	98	iPhone 12	7.52.0	Germany	4.98	Entertainment
3	4	Audible	40	2023-04-03	5	Great app, but it's a bit pricey.	74	iPhone 13	5.260.15	Australia	0.00	Books
4	5	Spotify	44	2023-05-01	1	Good, but has connection issues sometimes.	47	iPhone SE	4.50.18	Australia	14.31	Music
...	...	...	...	...	...	...	...	...	...	...	...	...
995	996	Headspace	30	2023-11-15	3	Good, but the premium content is expensive.	65	iPhone SE	6.284.11	US	0.00	Health

DATAFRAME INFO

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Review_ID        1000 non-null   int64  
 1   App_Name         1000 non-null   object 
 2   User_Age         1000 non-null   int64  
 3   Review_Date      1000 non-null   object 
 4   Rating           1000 non-null   int64  
 5   Review_Text      1000 non-null   object 
 6   Likes            1000 non-null   int64  
 7   Device_Type      1000 non-null   object 
 8   Version_Used     1000 non-null   object 
 9   Country          1000 non-null   object 
 10  Purchase_Amount  1000 non-null   float64
 11  Category         1000 non-null   object 
dtypes: float64(1), int64(4), object(7)
memory usage: 93.9+ KB
```

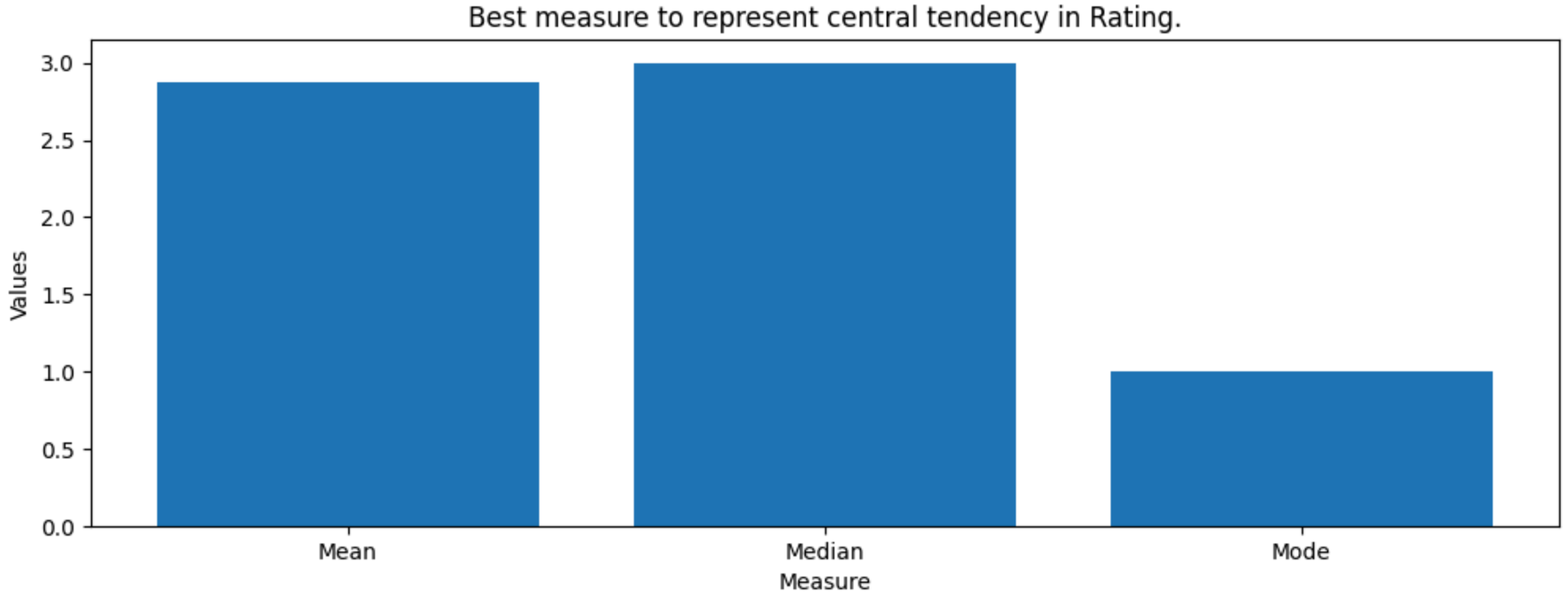
DESCRIBE

```
df.describe()
```

	Review_ID	User_Age	Rating	Likes	Purchase_Amount
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	39.211000	2.869000	44.776000	5.361120
std	288.819436	11.908917	1.467649	28.685444	5.755652
min	1.000000	18.000000	1.000000	0.000000	0.000000
25%	250.750000	30.000000	1.000000	17.000000	0.000000
50%	500.500000	39.000000	3.000000	42.500000	4.995000
75%	750.250000	49.000000	4.000000	71.000000	10.192500
max	1000.000000	60.000000	5.000000	100.000000	19.970000

1. CALCULATE THE MEAN, MEDIAN, AND MODE OF THE APP RATINGS IN THE DATASET. WHICH MEASURE (MEAN, MEDIAN, OR MODE) BEST REPRESENTS THE CENTRAL TENDENCY OF THE RATINGS?

```
plt.figure(figsize=(12,4))
plt.bar(x=["Mean","Median","Mode"],height=[rating_mean,rating_median,rating_mode])
plt.title("Best measure to represent central tendency in Rating.")
plt.xlabel("Measure")
plt.ylabel("Values")
plt.savefig("Best measure to represent central tendency in Rating")
plt.show()
```

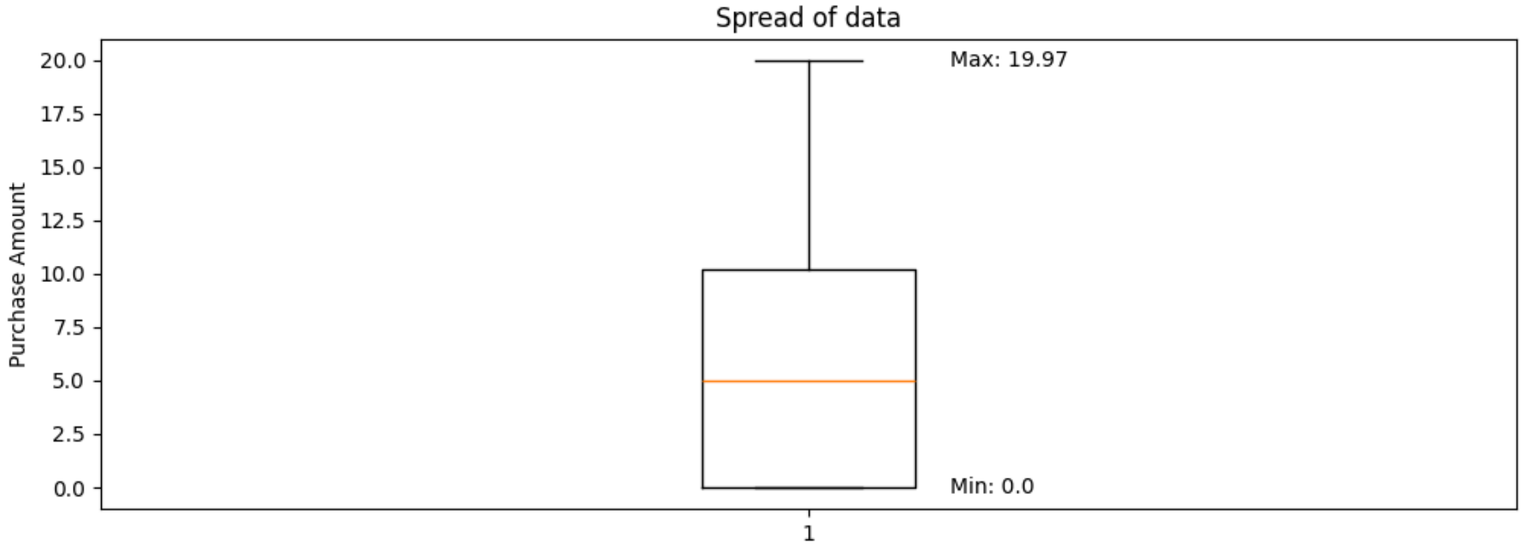


CALCULATION

```
#calculating mean of rating column in dataset.
rating_mean = round(float(df['Rating'].mean()),2)
rating_mean

#calculating median of rating column in dataset.
rating_median = round(float(df['Rating'].median()),2)
rating_median

#calculating mode of rating column in dataset.
rating_mode = float(df['Rating'].mode()[0])
rating_mode

print(f"Mean of rating : {rating_mean} .")
print(f"Median of rating : {rating_median} .")
print(f"Mode of rating : {rating_mode} .")
```

2. FIND THE RANGE AND INTERQUARTILE RANGE (IQR) OF THE PURCHASE_AMOUNT IN THE DATASET. HOW DO THESE VALUES HELP IN UNDERSTANDING THE SPREAD OF THE DATA? MODE) BEST REPRESENTS THE CENTRAL TENDENCY OF THE RATINGS?

```
plt.figure(figsize=(12,4))
plt.boxplot(df['Purchase_Amount'])
plt.title("Spread of data")
plt.ylabel("Purchase Amount")
# Add text labels for min and max
plt.text(1.1, min_purchase_amount, f'Min: {min_purchase_amount}', va='center')
plt.text(1.1, max_purchase_amount, f'Max: {max_purchase_amount}', va='center')
plt.show()
```



CALCULATION

```
#Max value in Purchase_Amount column.
max_purchase_amount = float(df['Purchase_Amount'].max())
max_purchase_amount

#Min value in Purchase_Amount column.
min_purchase_amount = float(df['Purchase_Amount'].min())
min_purchase_amount

#Range of Purchase_Amount column.
range_purchase_amount = max_purchase_amount-
min_purchase_amount
print(f"Range of Purchase_Amount column in
{range_purchase_amount} .")
```

3. CALCULATE THE VARIANCE AND STANDARD DEVIATION FOR THE NUMBER OF LIKES RECEIVED ON REVIEWS. WHAT DOES THE STANDARD DEVIATION INDICATE ABOUT THE SPREAD OF THE DATA?

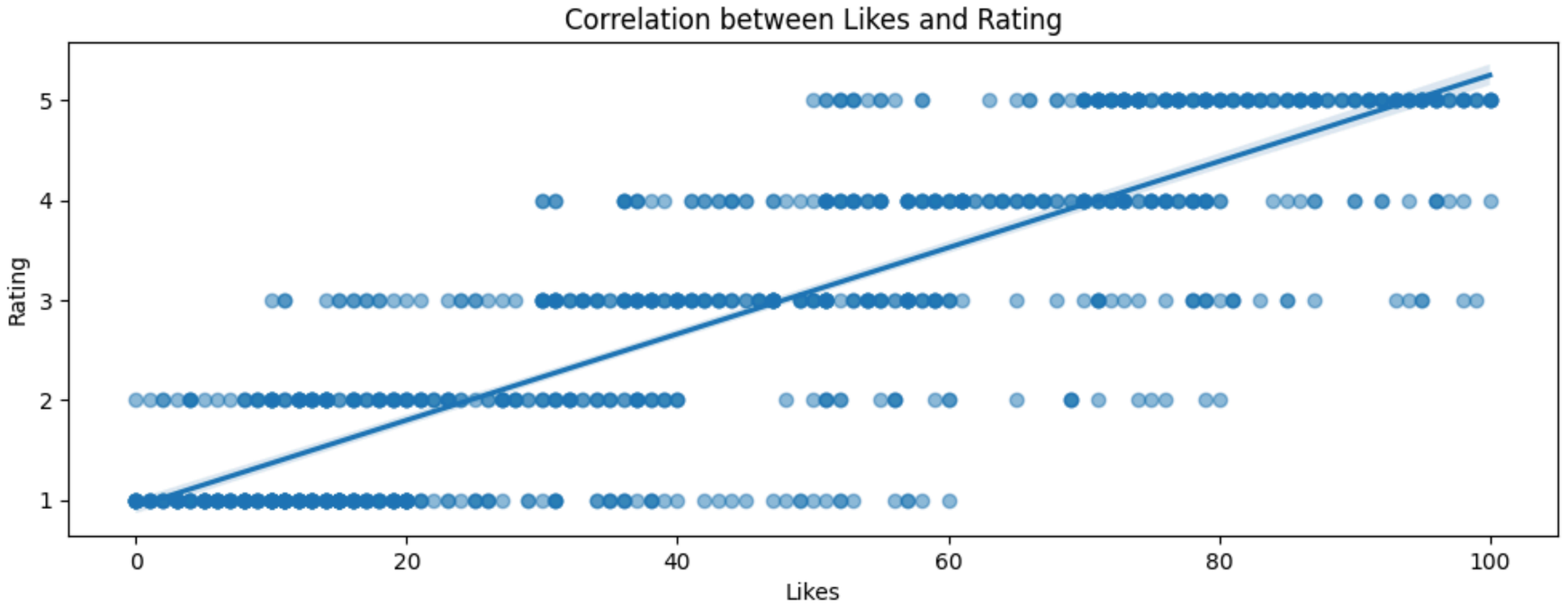
```
variance_likes = float(round(df['Likes'].var(),2))
variance_likes
```

```
standard_deviation_likes = float(round(df['Likes'].std(),2))
standard_deviation_likes
```

Observation: As we can observe, the standard deviation is 28.69, which is a relatively high value. This indicates that the number of likes varies widely between reviews.

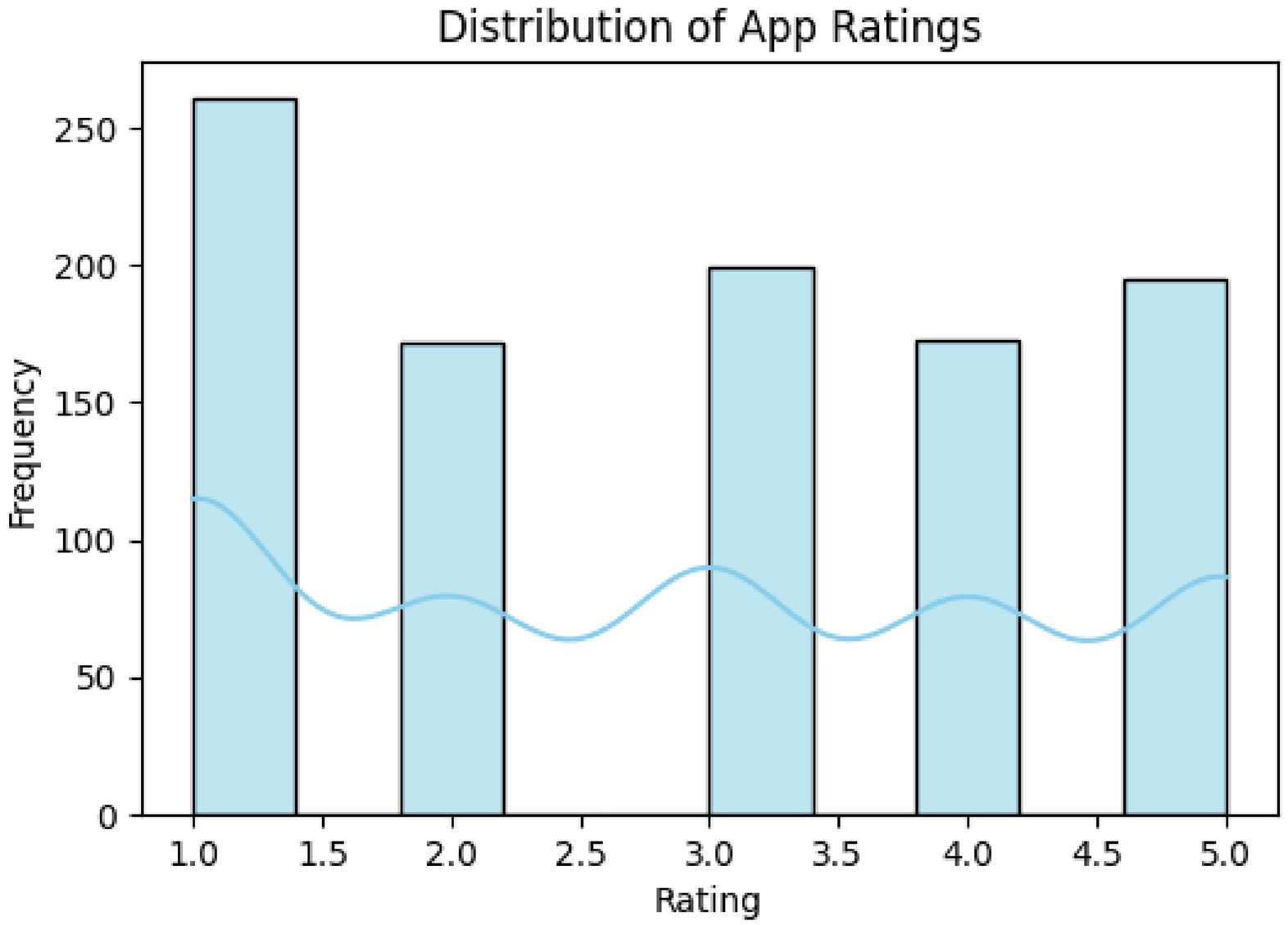
4. DETERMINE THE CORRELATION BETWEEN THE LIKES AND THE RATING GIVEN. IS THERE A POSITIVE, NEGATIVE, OR NO CORRELATION BETWEEN THESE VARIABLES?

```
# Calculate correlation value
correlation = round(df['Likes'].corr(df['Rating']),2)

if(correlation>0):
  print(f"Correlation coefficient have Positive relationship b/w Likes and Rating with value of {correlation} .")
elif(correlation<0):
   print(f"Correlation coefficient have Negative relationship b/w Likes and Rating with value of {correlation} .")
else:
   print(f"Correlation coefficient have no relationship b/w Likes and Rating with value of {correlation} .")
```


Correlation between Likes and Rating
Rating
Likes
1
2
3
4
5
0
20
40
60
80
100

5. PLOT THE DISTRIBUTION OF THE APP RATINGS. IS THE DISTRIBUTION POSITIVELY OR NEGATIVELY SKEWED? WHAT DOES THIS INDICATE ABOUT USER SATISFACTION?

```
plt.figure(figsize=(6,4))
sns.histplot(df['Rating'], bins=10, kde=True, color='skyblue')
plt.title('Distribution of App Ratings')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.savefig("Distribution of App Ratings")
plt.show()


skew_value = round(df['Rating'].skew(),2)
#print("Skewness:", skew_value)
if skew_value >0:
    print(f"There is Right Skewed/Positive Skewness. This means Mean and Median > Mode.")
elif skew_value < 0:
    print(f"There is Left Skewed/Negative Skewness. This means Mean and Median < Mode.")
else:
    print(f"There is Normal Destribution. This means Mean=Median=Mode")
```

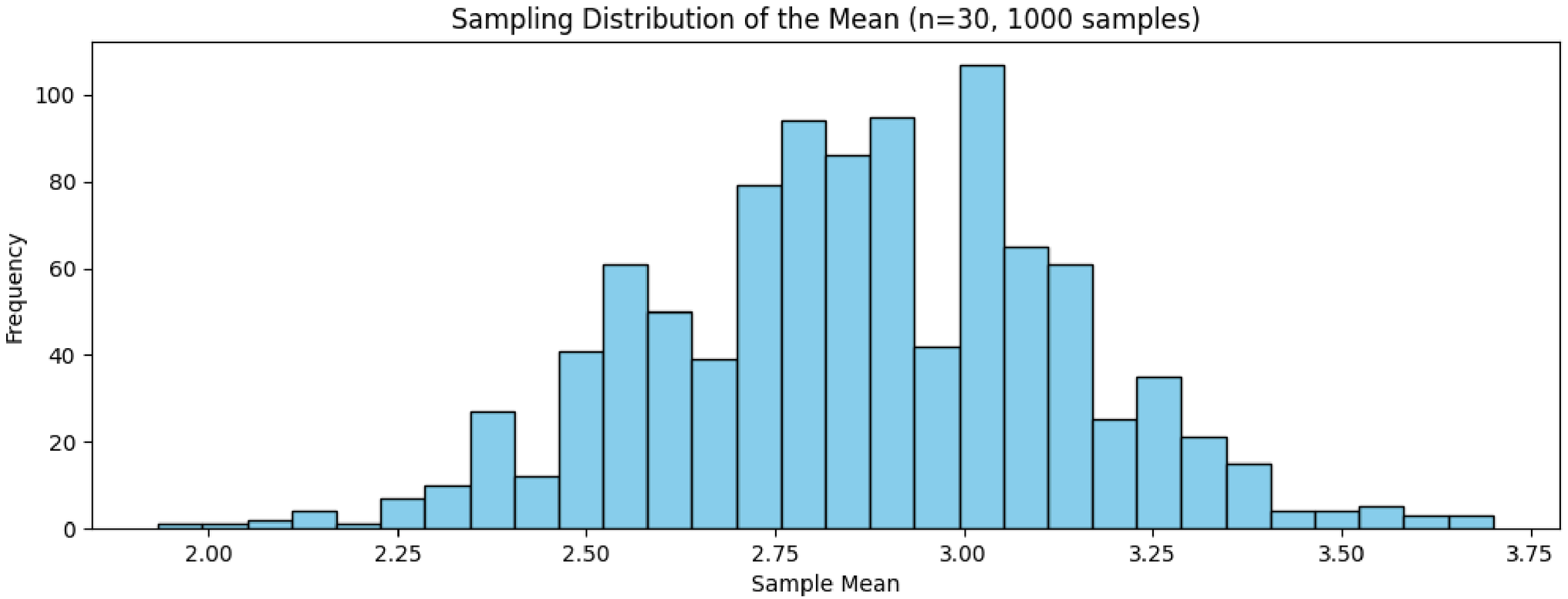

Distribution of App Ratings
Frequency
250
200
150
100
50
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Rating

6. PERFORM A HYPOTHESIS TEST TO DETERMINE IF THE AVERAGE RATING FOR INSTAGRAM IS SIGNIFICANTLY HIGHER THAN THE AVERAGE RATING FOR WHATSAPP. USE A 95% CONFIDENCE LEVEL.

```
insta_ratings = df[df['App_Name'] == 'Instagram']['Rating']
whatsapp_ratings = df[df['App_Name'] == 'WhatsApp']['Rating']


from scipy import stats


t_stat, p_value = stats.ttest_ind(insta_ratings, whatsapp_ratings, alternative='greater')
print("t-statistic:", t_stat)
print("p-value:", p_value)


if(p_value < 0.05):
    print(f"Reject H₀ -> Instagram’s average rating is significantly higher.")
else:
    #p_value ≥ 0.05
    print(f"Fail to reject H₀ -> No significant evidence that Instagram’s average rating is higher.")
```

7. TAKE RANDOM SAMPLES OF RATINGS FROM THE DATASET AND CALCULATE THEIR MEANS. CREATE A SAMPLING DISTRIBUTION AND EXPLAIN HOW THIS RELATES TO THE CENTRAL LIMIT THEOREM.

```
sample_means = []

for i in range(1000):
    sample = df['Rating'].sample(n=30, replace=True)
    sample_means.append(sample.mean())

sample_means = np.array(sample_means)

plt.figure(figsize=(12,4))
plt.hist(sample_means, bins=30, color='skyblue', edgecolor='black')
plt.title('Sampling Distribution of the Mean (n=30, 1000 samples)')
plt.xlabel('Sample Mean')
plt.ylabel('Frequency')
plt.show()
```


Sampling Distribution of the Mean (n=30, 1000 samples)
Frequency
0
20
40
60
80
100
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
Sample Mean

KEY INSIGHTS

- Median rating best represents user experience.
- Purchase amounts vary widely among users.
- Strong positive correlation between likes and ratings.
- No statistical difference between Instagram & WhatsApp ratings.
- Sampling distribution confirms the Central Limit Theorem.

KEY INSIGHTS

Mean Rating: 2.87
Median Rating: 3.00
Mode Rating: 1.00
The median is the best measure of central tendency for ratings because it is less affected by skewness and outliers.

Minimum: 0.00
Maximum: 19.97
Range: 19.97
The wide range suggests that in-app spending varies greatly among users.

KEY INSIGHTS

Variance: 822.85
Standard Deviation: 28.69
A high standard deviation indicates that the number of likes received on reviews varies widely.

Correlation Coefficient: 0.84 (Strong positive relationship)
This means higher-rated reviews tend to receive more likes.

KEY INSIGHTS

Skewness: Positive (Right-skewed)
Interpretation: There are more low ratings than high ratings, suggesting possible dissatisfaction among some users.

Null Hypothesis (H_0): Instagram's average rating is less than or equal to WhatsApp's average rating.
p-value: 0.786 (> 0.05)
Conclusion: Fail to reject H_0 → No significant evidence that Instagram's average rating is higher.

KEY INSIGHTS

By taking 1000 random samples of 30 ratings each, the sampling distribution of the mean was approximately normal, even though the population distribution was skewed. This supports the Central Limit Theorem, which states that the distribution of sample means tends to be normal regardless of the population's distribution.

THANK YOU

FAROOK MOHAMMAD

PORTFOLIO