In [3]:
```
import pandas as pd
df=pd.read_csv("Apple_Store_Reviews.csv")
df
```

Out[3]:

	Review_ID	App_Name	User_Age	Review_Date	Rating	Review_Text	Lik
0	1	Candy Crush Saga	21	2023-01-16	4	Great game, but too many in-game purchases.	
1	2	Spotify	57	2024-02-01	1	Good, but has connection issues sometimes.	
2	3	TikTok	33	2023-11-30	5	Awesome app! Best entertainment content.	
3	4	Audible	40	2023-04-03	5	Great app, but it's a bit pricey.	
4	5	Spotify	44	2023-05-01	1	Good, but has connection issues sometimes.	
...	...	...	...	...	...	...	
995	996	Headspace	30	2023-11-15	3	Good, but the premium content is expensive.	
996	997	Duolingo	19	2024-09-27	1	Disappointing. Hard to follow and buggy.	
997	998	Duolingo	38	2023-06-07	5	Excellent for learning new skills!	
998	999	Instagram	52	2024-03-04	4	Great app, but sometimes it lags.	
999	1000	Audible	25	2024-02-20	2	Terrible. Very limited selection of books.	

1000 rows × 12 columns

In [8]:
```
df.columns
```

Out[8]:
```
Index(['Review_ID', 'App_Name', 'User_Age', 'Review_Date', 'Rating',
       'Review_Text', 'Likes', 'Device_Type', 'Version_Used', 'Country',
       'Purchase_Amount', 'Category'],
      dtype='object')
```

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?

In [9]:
```
df.describe()
```

Out[9]:

	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

In [11]:
```
mean_ratings = df["Rating"].mean()
mean_ratings
```

Out[11]:
```
np.float64(2.869)
```

In [12]:
```
median_ratings = df["Rating"].median()
median_ratings
```

Out[12]:
```
np.float64(3.0)
```

In [14]:
```
mode_ratings = df["Rating"].mode()[0]
mode_ratings
```

Out[14]:
```
np.int64(1)
```

In [ ]:
```
#The median rating best represents the central
#tendency of app ratings because it is less affected by extreme values and s
```

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?

In [15]:
```
#range
range_purchase_Amount=df["Purchase_Amount"].max()-df["Purchase_Amount"].min(
range_purchase_Amount
```

Out[15]:
```
np.float64(19.97)
```

In [21]:
```
#IQR=q3-q1
Q1=df["Purchase_Amount"].quantile(0.25)
Q3=df["Purchase_Amount"].quantile(0.75)
Q1,Q3
#Q1 is zero mean 25% of users does not purchase anuthing and 75% of purchase
```

Out[21]:
```
(np.float64(0.0), np.float64(10.192499999999999))
```

In [22]:
```
IQR = Q3-Q1
print("The interquartile range (IQR) of the Purchase_Amount is approximately
```

```
The interquartile range (IQR) of the Purchase_Amount is approximately 10.19,
indicating that the middle 50% of purchases lie between 0 and 10.19. Since t
he first quartile is 0, this suggests a significant proportion of users made
no purchase, leading to a right-skewed distribution. 10.192499999999999
```

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?

In [29]:
```
variance_like = df["Likes"].var()
std_like=df["Likes"].std()
print("This is Variance (sigma)^2: ",variance_like,"This is std (sigma): ",s
print()
print("Although the average number of likes per review is low (2.86), the ve
```

```
This is Variance (sigma)^2:  822.8546786786787 This is std (sigma):  28.6854
43672334557

Although the average number of likes per review is low (2.86), the very high
standard deviation (28.9) indicates extreme variability in engagement. This
suggests that while most reviews receive minimal likes, a small number of re
views gain disproportionately high attention, leading to a highly right-skew
ed distribution.
```

Determine the correlation between the likes and the rating given. Is there a positive, negative, or no correlation between these variables?

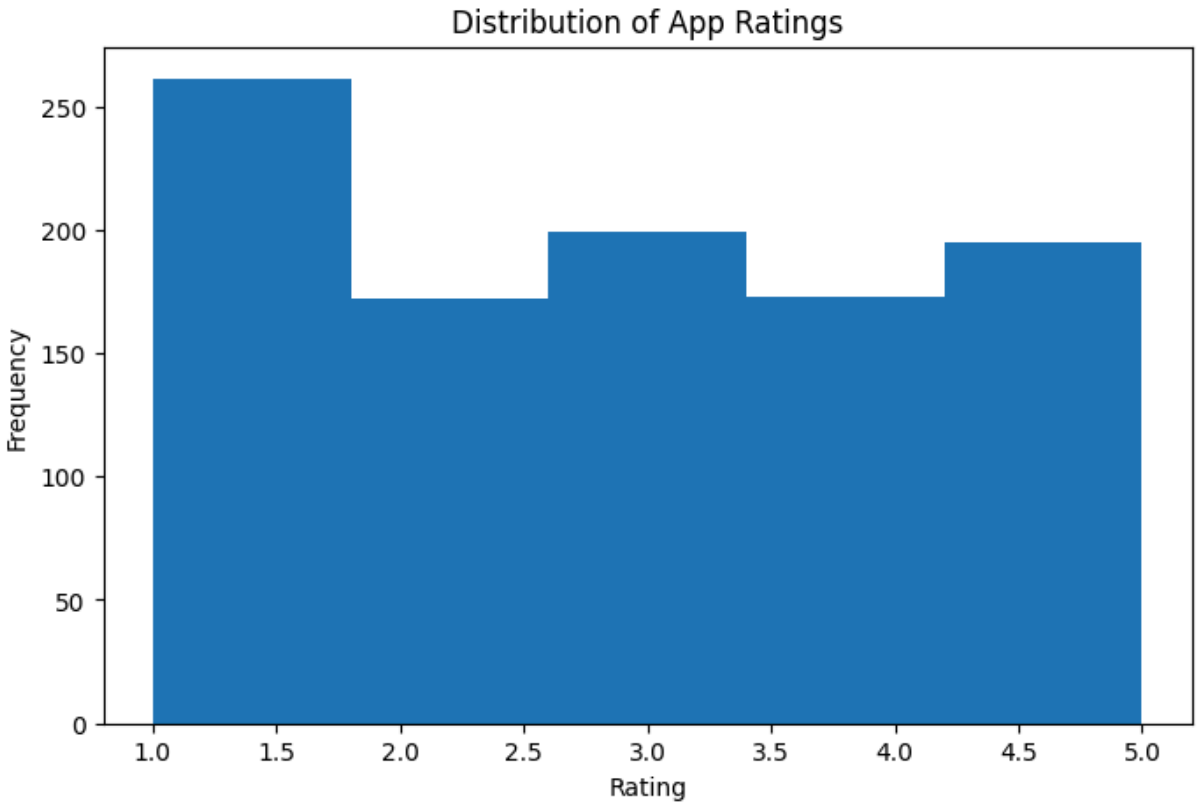
In [33]:
```
corr = df["Likes"].corr(df["Rating"])
corr
print("corr values is 0.84 after cal. which is close to 1 it means this is (
```

```
corr values is 0.84 after cal. which is close to 1 it means this is (+)ve co
relation. This indicates strong Linear relationship
```

Plot the distribution of the app ratings. Is the distribution positively or negatively skewed? What does this indicate about user satisfaction?

In [39]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
plt.hist(df["Rating"], bins=5) #rating from 1 to 5
plt.xlabel("Rating")
plt.ylabel("Frequency")
plt.title("Distribution of App Ratings")
plt.show()
print("""The histogram shows that app ratings are more concentrated toward h
This results in a negatively skewed distribution, indicating that most users
while a smaller proportion report dissatisfaction.""")
```




```
The histogram shows that app ratings are more concentrated toward higher val
ues (3 to 5), with fewer low ratings (1 to 2).
This results in a negatively skewed distribution, indicating that most users
are generally satisfied with the apps,
while a smaller proportion report dissatisfaction.
```

In [40]:
```
# Perform a hypothesis test to determine if the average rating for Instagram
# step-1 Null Hypotheis [ho , mew(instagram) <= mew(whatsapp)]
# step-2 Alternative Hypotheis [h1 , mew(instagram) > mew(whatsapp)]
insta_rating = df[df["App_Name"]=="Instagram"]["Rating"]
whatsapp_rating = df[df["App_Name"]=="Whatsapp"]["Rating"]
```

In [41]:
```
pip install scipy
```

```
Collecting scipy
  Downloading scipy-1.17.0-cp314-cp314-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.7,>=1.26.4 in c:\users\dell\appdata\l
ocal\programs\python\python314\lib\site-packages (from scipy) (2.3.3)
Downloading scipy-1.17.0-cp314-cp314-win_amd64.whl (37.1 MB)
   ---------------------------------------- 0.0/37.1 MB ? eta -:--:--
   - -------------------------------------- 1.0/37.1 MB 6.2 MB/s eta 0:00:06
   - -------------------------------------- 1.8/37.1 MB 4.8 MB/s eta 0:00:08
   -- ------------------------------------- 2.6/37.1 MB 4.4 MB/s eta 0:00:08
   --- ------------------------------------ 3.4/37.1 MB 4.3 MB/s eta 0:00:08
   ---- ----------------------------------- 4.5/37.1 MB 4.2 MB/s eta 0:00:08
   ----- ---------------------------------- 4.7/37.1 MB 4.0 MB/s eta 0:00:09
   ----- ---------------------------------- 5.5/37.1 MB 3.8 MB/s eta 0:00:09
   ------ --------------------------------- 6.0/37.1 MB 3.8 MB/s eta 0:00:09
   ------- -------------------------------- 6.8/37.1 MB 3.5 MB/s eta 0:00:09
   -------- ------------------------------- 7.6/37.1 MB 3.6 MB/s eta 0:00:09
   --------- ------------------------------ 8.4/37.1 MB 3.6 MB/s eta 0:00:08
   --------- ------------------------------ 9.2/37.1 MB 3.6 MB/s eta 0:00:08
   ---------- ----------------------------- 10.2/37.1 MB 3.6 MB/s eta 0:00:0
8
   ---------- ----------------------------- 10.7/37.1 MB 3.6 MB/s eta 0:00:0
8
   ----------- ---------------------------- 11.5/37.1 MB 3.5 MB/s eta 0:00:0
8
   ------------ --------------------------- 12.3/37.1 MB 3.6 MB/s eta 0:00:0
7
   ------------- -------------------------- 13.1/37.1 MB 3.6 MB/s eta 0:00:0
7
   ------------- -------------------------- 13.9/37.1 MB 3.6 MB/s eta 0:00:0
7
   --------------- ------------------------ 14.9/37.1 MB 3.6 MB/s eta 0:00:0
7
   --------------- ------------------------ 15.7/37.1 MB 3.6 MB/s eta 0:00:0
6
   ---------------- ----------------------- 16.5/37.1 MB 3.6 MB/s eta 0:00:0
6
   ----------------- ---------------------- 17.3/37.1 MB 3.6 MB/s eta 0:00:0
6
   ------------------ --------------------- 18.4/37.1 MB 3.6 MB/s eta 0:00:0
6
   ------------------- -------------------- 19.1/37.1 MB 3.6 MB/s eta 0:00:0
5
   ------------------- -------------------- 19.7/37.1 MB 3.6 MB/s eta 0:00:0
5
   -------------------- ------------------- 20.2/37.1 MB 3.6 MB/s eta 0:00:0
5
   -------------------- ------------------- 20.4/37.1 MB 3.5 MB/s eta 0:00:0
5
   --------------------- ------------------ 21.0/37.1 MB 3.5 MB/s eta 0:00:0
5
   --------------------- ------------------ 21.2/37.1 MB 3.4 MB/s eta 0:00:0
5
   ---------------------- ----------------- 21.8/37.1 MB 3.4 MB/s eta 0:00:0
5
   ----------------------- ---------------- 22.3/37.1 MB 3.3 MB/s eta 0:00:0
5
```

```
   ------------------------ -------------- 23.3/37.1 MB 3.3 MB/s eta 0:00:0
5
   ------------------------- ------------- 24.1/37.1 MB 3.3 MB/s eta 0:00:0
4
   ------------------------- ------------- 24.9/37.1 MB 3.3 MB/s eta 0:00:0
4
   -------------------------- ------------ 25.7/37.1 MB 3.4 MB/s eta 0:00:0
4
   -------------------------- ------------ 26.5/37.1 MB 3.4 MB/s eta 0:00:0
4
   --------------------------- ----------- 27.0/37.1 MB 3.3 MB/s eta 0:00:0
4
   --------------------------- ----------- 27.3/37.1 MB 3.3 MB/s eta 0:00:0
3
   --------------------------- ----------- 27.8/37.1 MB 3.3 MB/s eta 0:00:0
3
   ---------------------------- ---------- 28.6/37.1 MB 3.3 MB/s eta 0:00:0
3
   ---------------------------- ---------- 29.6/37.1 MB 3.3 MB/s eta 0:00:0
3
   ----------------------------- --------- 30.7/37.1 MB 3.3 MB/s eta 0:00:0
2
   ----------------------------- --------- 31.5/37.1 MB 3.3 MB/s eta 0:00:0
2
   ------------------------------ -------- 32.0/37.1 MB 3.3 MB/s eta 0:00:0
2
   ------------------------------ -------- 32.5/37.1 MB 3.3 MB/s eta 0:00:0
2
   ------------------------------- ------- 32.8/37.1 MB 3.2 MB/s eta 0:00:0
2
   ------------------------------- ------- 33.6/37.1 MB 3.2 MB/s eta 0:00:0
2
   -------------------------------- ------ 34.3/37.1 MB 3.2 MB/s eta 0:00:0
1
   -------------------------------- ------ 35.1/37.1 MB 3.3 MB/s eta 0:00:0
1
   --------------------------------- ----- 35.9/37.1 MB 3.3 MB/s eta 0:00:0
1
   ---------------------------------- ----  36.7/37.1 MB 3.3 MB/s eta 0:00:0
1
   ---------------------------------- ----- 37.1/37.1 MB 3.2 MB/s  0:00:11
Installing collected packages: scipy
Successfully installed scipy-1.17.0
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 25.2 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [43]:

```python
from scipy.stats import ttest_ind

t_stat, p_value = ttest_ind(
    insta_rating,
    whatsapp_rating,
    equal_var=False  # Welch's t-test (safe choice)
)
```

```
t_stat, p_value
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_2520\504490773.py:3: SmallSampleW
arning: One or more sample arguments is too small; all returned values will
be NaN. See documentation for sample size requirements.
  t_stat, p_value = ttest_ind(
```

Out[43]: (np.float64(nan), np.float64(nan))

In [44]:
```
one_tailed_p = p_value / 2
one_tailed_p
```

Out[44]: np.float64(nan)

In [45]:
```
alpha = 0.05

if one_tailed_p < alpha:
    print("Reject H0: Instagram average rating is significantly higher than
else:
    print("Fail to reject H0: No significant difference found.")
```

```
Fail to reject H0: No significant difference found.
```

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.

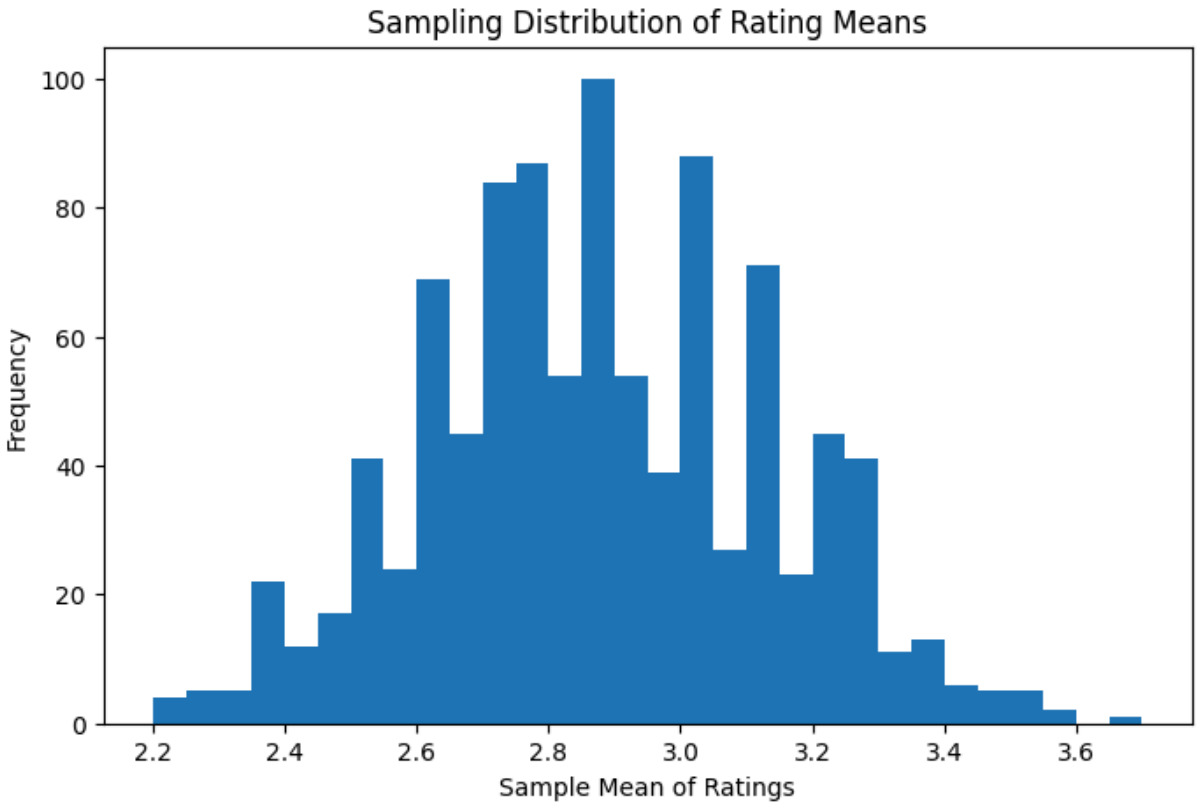
In [46]:
```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
sample_size = 30        # size of each sample
num_samples = 1000      # number of samples

sample_means = []

for _ in range(num_samples):
    sample = df["Rating"].sample(n=sample_size, replace=True)
    sample_means.append(sample.mean())
```

In [47]:
```
plt.figure(figsize=(8,5))
plt.hist(sample_means, bins=30)
plt.xlabel("Sample Mean of Ratings")
plt.ylabel("Frequency")
plt.title("Sampling Distribution of Rating Means")
plt.show()
```



In [48]:
```
print("""Sampling distribution will look:

                    1. Symmetric

                    2. Bell-shaped

                    3.Approximately normal

✅ This is Central Limit Theorem in action""")
```

```
Sampling distribution will look:

                    1. Symmetric

                    2. Bell-shaped

                    3.Approximately normal

✅ This is Central Limit Theorem in action
```

In [ ]: