Analysis of Graduate Data

In [ ]:

Loading the Dependables

In [2]:

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

Loading the data set

In [4]:

```
# Setting the file path

file_path = 'Reward_Program_Assignment_Input_v6 - TA.xlsx'

# Loading the data
df = pd.read_excel(file_path, sheet_name='Raw_Reward_Data')
```

In [5]:

```
# Initial EDA

df.head()
```

Out[5]:

	Member_Name_Surname_Per_Redemption	Reward_Received	Brand	Reward_Value_Amount_in_Dolla
0	Jane Smith	Amazon Gift Card	Uber	
1	David Thompson	Coursera Subscription	Amazon	
2	James Wilson	Netflix Gift Card	Coursera	
3	David Thompson	Spotify Subscription	Amazon	
4	Alice Johnson	Spotify Gift Card	Coursera	

In [ ]:

Task 1

In [ ]:

In [9]:

```
# Check for duplicates based on all columns
num_duplicates = df.duplicated().sum()
```

```
# Print the total Number of Duplicates in the data
print(num_duplicates)
```

```
0
```

Observation: There are no duplicates in the data set or it is safe to say that the data set is already cleaned before given for this assignment

In [10]:

```
# Remove duplicates
df_cleaned = df.drop_duplicates()

# Save the cleaned dataset to a new Excel file
cleaned_file_path = 'cleaned_graduate_data.xlsx'
df_cleaned.to_excel(cleaned_file_path, index=False)

# Summary of cleaning process
summary = f"Total duplicates found and removed: {num_duplicates}\n"
summary += f"Remaining records after cleaning: {df_cleaned.shape[0]}"
print(summary)
```

```
Total duplicates found and removed: 0
Remaining records after cleaning: 100
```

In [ ]:

Task 3

In [ ]:

In [12]:

```
df = pd.read_excel("cleaned_graduate_data.xlsx")
```

In [14]:

```
df.head()
```

Out[14]:

	Member_Name_Surname_Per_Redemption	Reward_Received	Brand	Reward_Value_Amount_in_Dolla
0	Jane Smith	Amazon Gift Card	Uber	
1	David Thompson	Coursera Subscription	Amazon	
2	James Wilson	Netflix Gift Card	Coursera	
3	David Thompson	Spotify Subscription	Amazon	
4	Alice Johnson	Spotify Gift Card	Coursera	

In [ ]:

In [57]:

```
# Descriptive Statistics

df.describe()
```

Out[57]:

	Reward_Value_Amount_in_Dollars	Time_to_Reward_Received_in_Seconds	Redemptions_by_User	P
count	100.000000	100.000000	100.00000	
mean	48.350000	28.640000	5.29000	
std	32.791467	16.472518	2.71656	
min	10.000000	1.000000	1.00000	
25%	25.000000	14.750000	3.00000	
50%	50.000000	31.000000	5.00000	
75%	75.000000	41.250000	8.00000	
max	100.000000	59.000000	9.00000	

In [ ]:

Analysis:

1. Distribution of Graduates by Country:

This will help us understand where most of the graduates are located, which can inform region-based engagement strategies.

2. Popular Rewards and Brands:

We'll analyze which rewards and brands are the most frequently redeemed, indicating potential preferences.

3. Satisfaction Rating Analysis:

We'll examine satisfaction ratings to determine how happy graduates are with the rewards they receive.

4. Redemption Frequency:

We'll look into how frequently individuals are redeeming rewards, and the relationship between redemptions and satisfaction.

5. Cost vs. Satisfaction:

We'll explore whether higher-cost redemptions me know how you'd like to proceed!

In [ ]:

1. Distribution of Graduates by Country

In [48]:

```
# Distribution of graduates by country
graduates_by_country = df['Country'].value_counts()
```

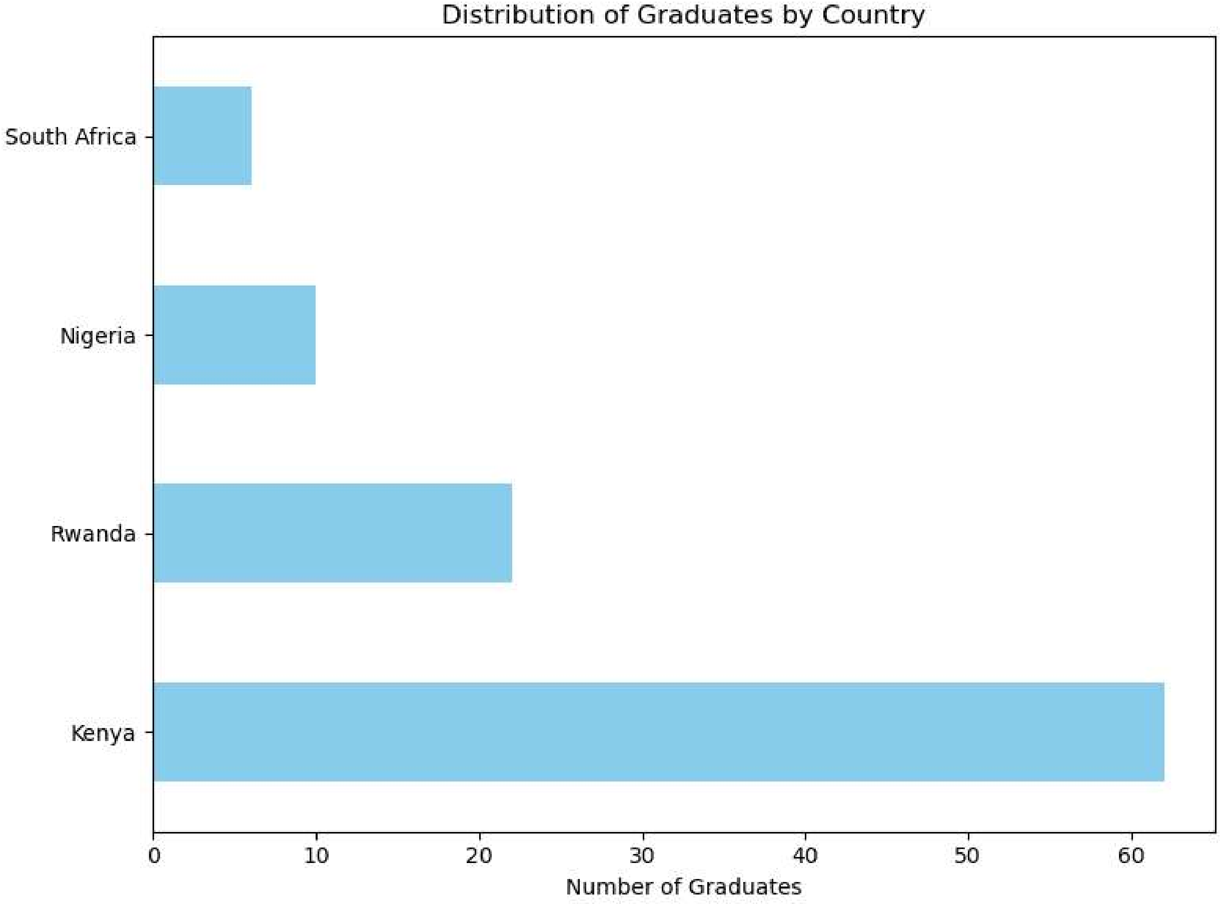
```
# Print the distribution
print("Distribution of Graduates by Country:")
graduates_by_country
```

Out[48]:

```
Distribution of Graduates by Country:
Country
Kenya           62
Rwanda          22
Nigeria         10
South Africa     6
Name: count, dtype: int64
```

In [49]:

```
# Plot the distribution
plt.figure(figsize=(8, 6))
graduates_by_country.plot(kind='barh', color='skyblue')
plt.title('Distribution of Graduates by Country')
plt.xlabel('Number of Graduates')
plt.ylabel('')
plt.tight_layout()
plt.show()
```



In [ ]:

2. Popular Rewards and Brands

In [52]:

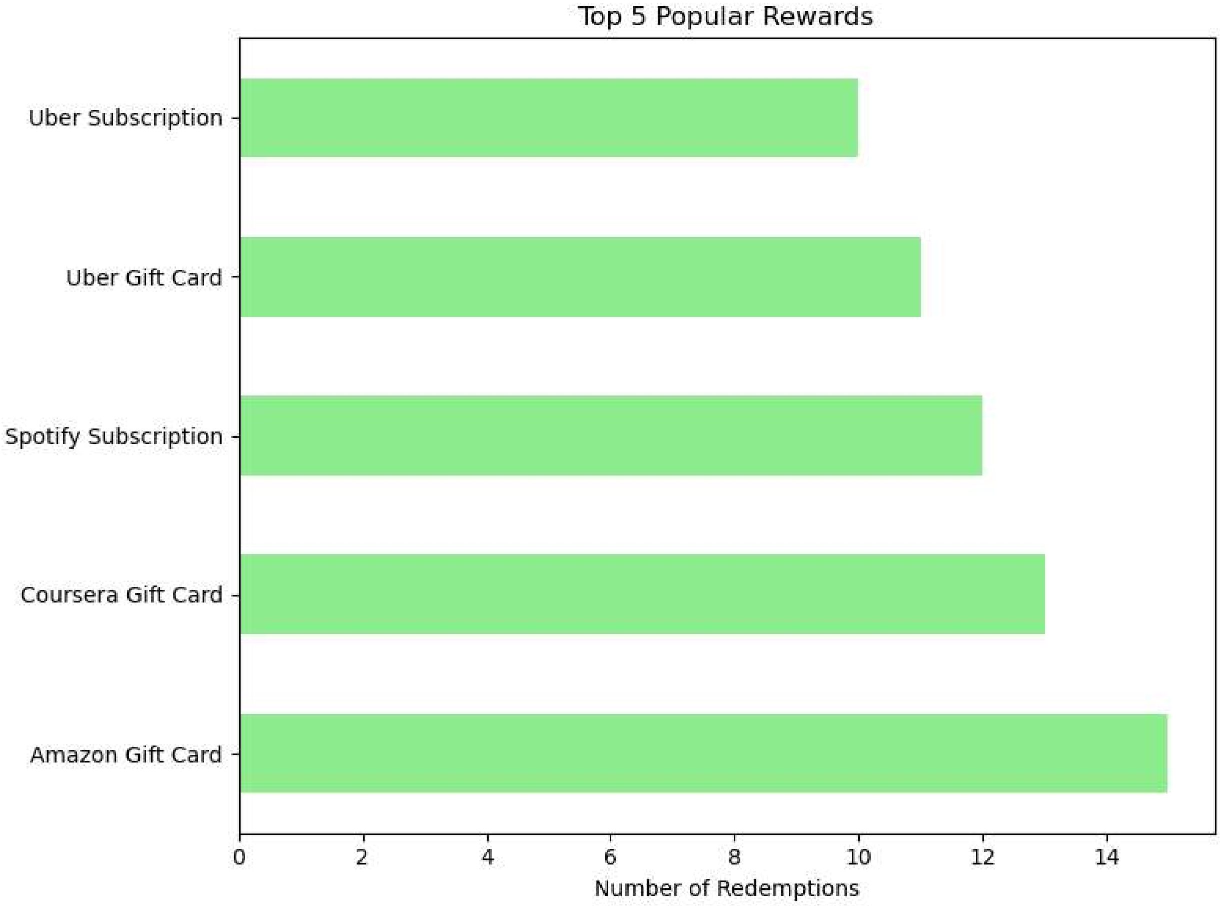
```
# Popular rewards
popular_rewards = df['Reward_Received'].value_counts()
print("Most Popular Rewards:")
popular_rewards
```

Out[52]:

```
Most Popular Rewards:
Reward_Received
Amazon Gift Card         15
Coursera Gift Card       13
Spotify Subscription     12
Uber Gift Card           11
Uber Subscription        10
Spotify Gift Card         9
Amazon Subscription       9
Netflix Subscription      8
Coursera Subscription     7
Netflix Gift Card         6
Name: count, dtype: int64
```

In [51]:

```
# Plot popular rewards
plt.figure(figsize=(8, 6))
popular_rewards.head(5).plot(kind='barh', color='lightgreen')
plt.title('Top 5 Popular Rewards')
plt.xlabel('Number of Redemptions')
plt.ylabel('')
plt.xticks()
plt.tight_layout()
plt.show()
```



In [ ]:

In [53]:

```
# Popular brands
popular_brands = df['Brand'].value_counts()
print("Most Popular Brands:")
popular_brands
```

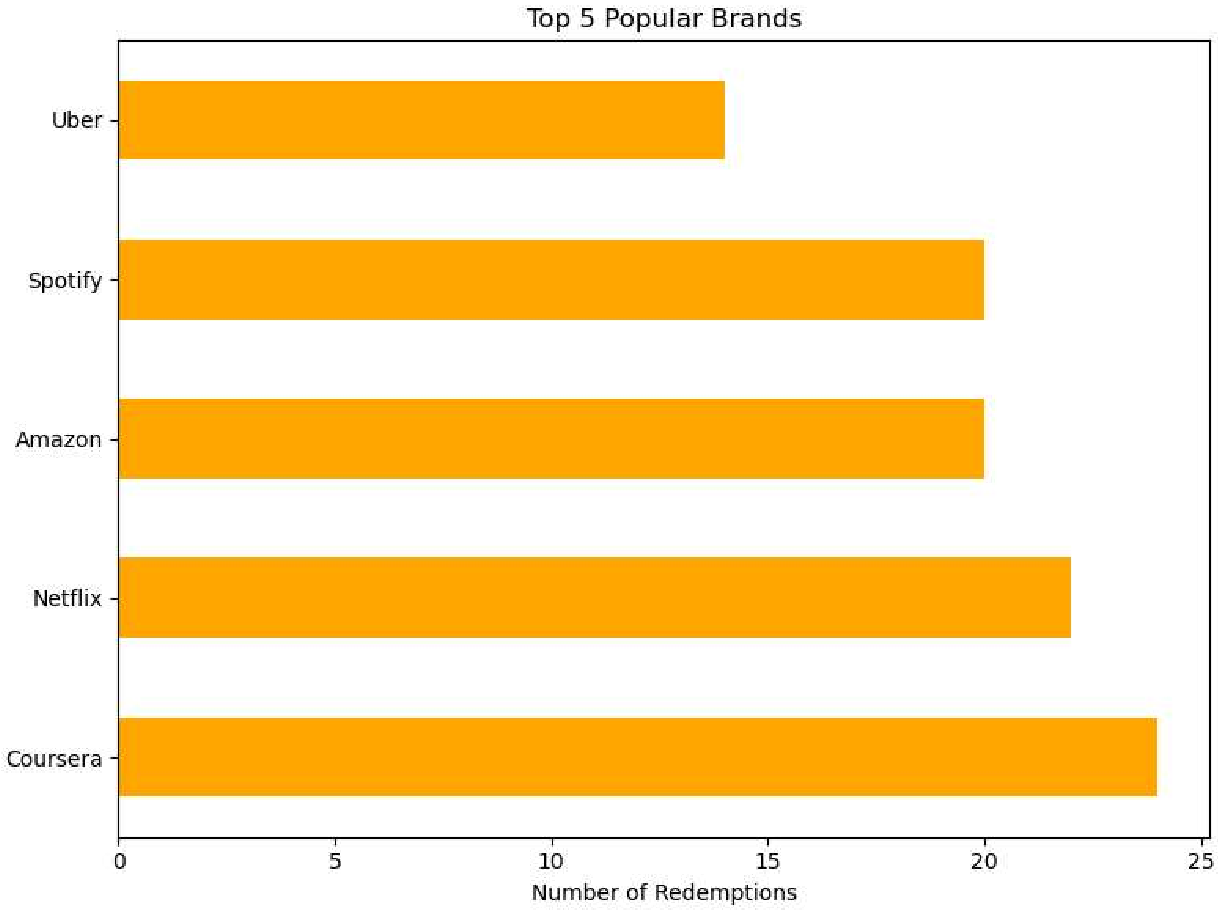
```
Most Popular Brands:
```

Out[53]:

```
Brand
Coursera    24
Netflix     22
Amazon      20
Spotify     20
Uber        14
Name: count, dtype: int64
```

In [54]:

```
# Plot popular brands
plt.figure(figsize=(8, 6))
popular_brands.head(5).plot(kind='barh', color='orange')
plt.title('Top 5 Popular Brands')
plt.xlabel('Number of Redemptions')
plt.ylabel('')
plt.xticks()
plt.tight_layout()
plt.show()
```



In [ ]:

3. Satisfaction Rating Analysis

In [46]:

```
# Descriptive statistics for satisfaction rating
satisfaction_stats = df['Satisfaction_Rating_on_Reward'].describe()
print("Satisfaction Rating Statistics:")
satisfaction_stats
```

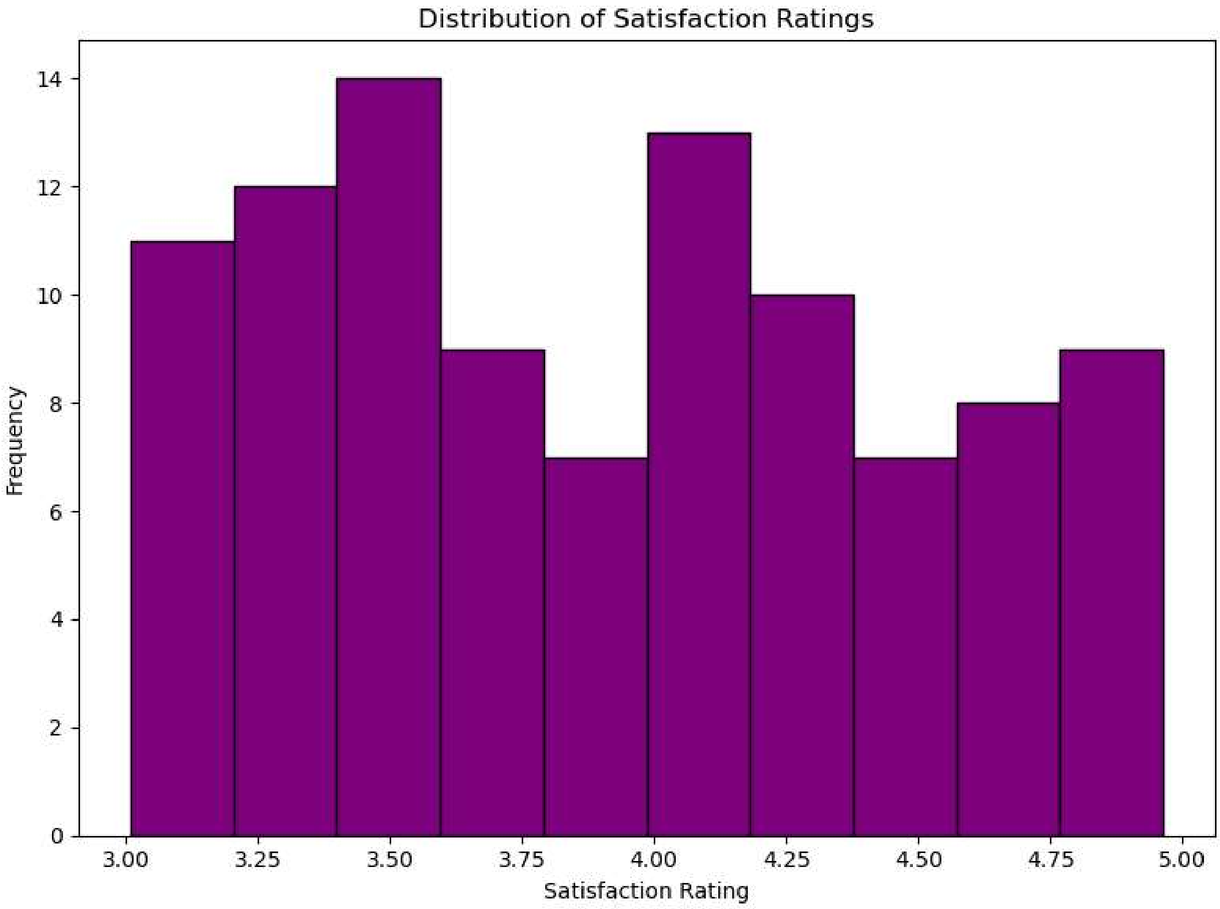
```
Satisfaction Rating Statistics:
```

Out[46]:

```
count    100.000000
mean       3.918829
std        0.564479
min        3.007999
25%        3.413431
50%        3.872438
75%        4.314277
max        4.963540
Name: Satisfaction_Rating_on_Reward, dtype: float64
```

In [47]:

```python
# Plot satisfaction rating distribution
plt.figure(figsize=(8, 6))
df['Satisfaction_Rating_on_Reward'].plot(kind='hist', bins=10, color='purple', edgecol
plt.title('Distribution of Satisfaction Ratings')
plt.xlabel('Satisfaction Rating')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



In [ ]:

In [ ]:

4. Redemption Frequency

In [44]:

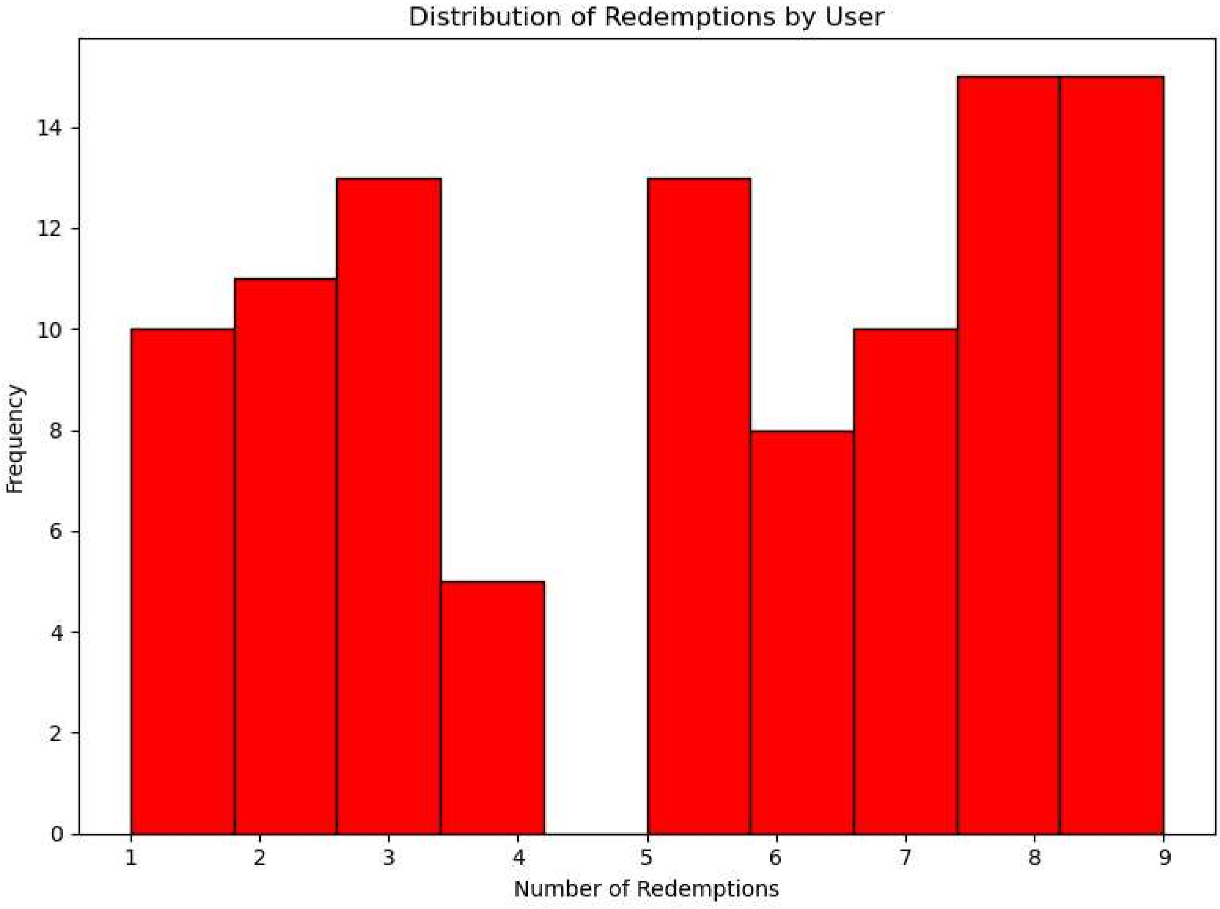
```python
# Descriptive statistics for redemption frequency
redemptions_stats = df['Redemptions_by_User'].describe()
print("Redemptions by User Statistics:")
redemptions_stats
```

Out[44]:

```
Redemptions by User Statistics:
count    100.00000
mean       5.29000
std        2.71656
min        1.00000
25%        3.00000
50%        5.00000
75%        8.00000
max        9.00000
Name: Redemptions_by_User, dtype: float64
```

In [45]:

```
# Plot redemption frequency
plt.figure(figsize=(8, 6))
df['Redemptions_by_User'].plot(kind='hist', bins=10, color='red', edgecolor='black')
plt.title('Distribution of Redemptions by User')
plt.xlabel('Number of Redemptions')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



In [ ]:

In [ ]:

5. Cost vs. Satisfaction Correlation

In [43]:

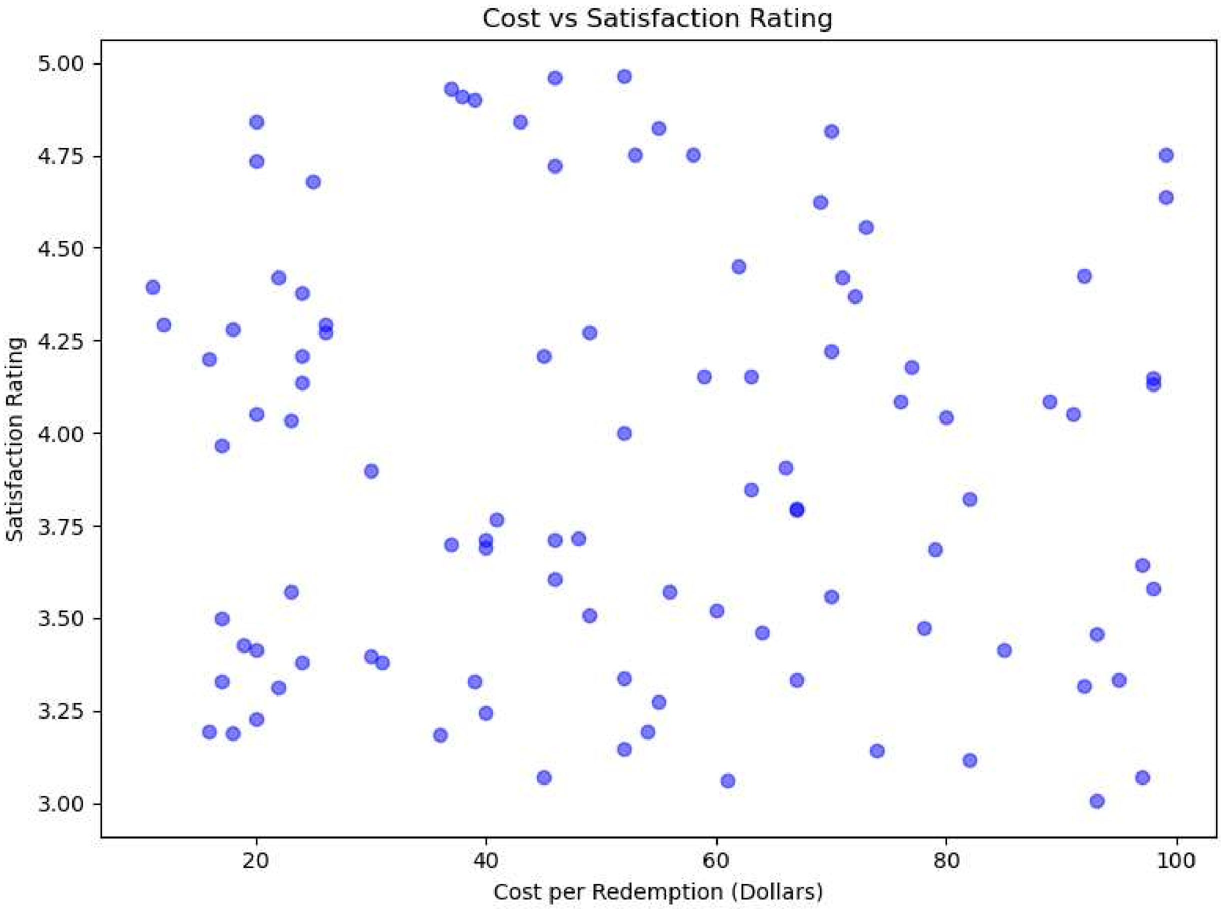
```
# Correlation between cost per redemption and satisfaction rating
cost_satisfaction_corr = df[['Cost_Per_Redemption_in_Dollars', 'Satisfaction_Rating_or
cost_satisfaction_corr
```

Out[43]:

	Cost_Per_Redemption_in_Dollars	Satisfaction_Rating_on_Reward
Cost_Per_Redemption_in_Dollars	1.000000	-0.055561
Satisfaction_Rating_on_Reward	-0.055561	1.000000

In [40]:

```
# Scatter plot of Cost vs Satisfaction Rating
plt.figure(figsize=(8, 6))
plt.scatter(df['Cost_Per_Redemption_in_Dollars'], df['Satisfaction_Rating_on_Reward'],
plt.title('Cost vs Satisfaction Rating')
plt.xlabel('Cost per Redemption (Dollars)')
plt.ylabel('Satisfaction Rating')
plt.tight_layout()
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

A detailed Breakdown of the insights is found in the ReadMe file Attached the repository

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```