In [ ]:
```
#Financial Data Analysis
```

In [ ]:
```
# Import Libraries
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
# Loading CSV file
```

In [2]:
```
df = pd.read_csv("Acquisition_Credit_Data.csv")
df.head()
```

Out[2]:

| | account_id | acquisition_channel | open_date | month | credit_line | balance | revenue | charge_off_amount | charge_off_flag | days_past_due | acquisition_cost |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | A100001 | Online | 2023-07-07 | 2023-07-01 | 4900 | 723.83 | 19.09 | 0.0 | 0 | 0 | 130 |
| **1** | A100002 | Mail | 2024-04-08 | 2024-04-01 | 4800 | 916.81 | 37.28 | 0.0 | 0 | 0 | 150 |
| **2** | A100003 | Online | 2024-01-28 | 2024-01-01 | 8100 | 872.38 | 22.84 | 0.0 | 0 | 0 | 180 |
| **3** | A100004 | Partner | 2023-05-19 | 2023-05-01 | 2300 | 600.00 | 5.28 | 0.0 | 0 | 0 | 130 |
| **4** | A100005 | Partner | 2023-10-26 | 2023-10-01 | 5700 | 407.63 | 7.75 | 0.0 | 0 | 0 | 150 |

In [ ]:
```
# Data Inspection
```

In [3]:
```
print("Rows, Columns:", df.shape)

# Column types
df.info()
```

```
Rows, Columns: (600, 11)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   account_id           600 non-null    object 
 1   acquisition_channel  600 non-null    object 
 2   open_date            600 non-null    object 
 3   month                600 non-null    object 
 4   credit_line          600 non-null    int64  
 5   balance              600 non-null    float64
 6   revenue              600 non-null    float64
 7   charge_off_amount    600 non-null    float64
 8   charge_off_flag      600 non-null    int64  
 9   days_past_due        600 non-null    int64  
 10  acquisition_cost     600 non-null    int64  
dtypes: float64(3), int64(4), object(4)
memory usage: 51.7+ KB
```

In [ ]:
```
# Summary Statistics
```

In [4]:
```
df.describe(include='all')
```

Out[4]:

| | account_id | acquisition_channel | open_date | month | credit_line | balance | revenue | charge_off_amount | charge_off_flag | days_past_due | acquisition_cost |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 600 | 600 | 600 | 600 | 600.000000 | 600.000000 | 600.000000 | 600.000000 | 600.000000 | 600.000000 | 600.000000 |
| **unique** | 600 | 3 | 419 | 24 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **top** | A100391 | Online | 2024-08-21 | 2024-06-01 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **freq** | 1 | 310 | 5 | 34 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **mean** | NaN | NaN | NaN | NaN | 5519.833333 | 2049.499483 | 33.939333 | 4.130017 | 0.003333 | 4.450000 | 128.683333 |
| **std** | NaN | NaN | NaN | NaN | 2570.168525 | 916.801090 | 17.462905 | 73.095696 | 0.057687 | 13.638457 | 22.522844 |
| **min** | NaN | NaN | NaN | NaN | 1000.000000 | 363.290000 | -2.080000 | 0.000000 | 0.000000 | 0.000000 | 100.000000 |
| **25%** | NaN | NaN | NaN | NaN | 3300.000000 | 1392.440000 | 21.595000 | 0.000000 | 0.000000 | 0.000000 | 120.000000 |
| **50%** | NaN | NaN | NaN | NaN | 5500.000000 | 1899.605000 | 30.875000 | 0.000000 | 0.000000 | 0.000000 | 120.000000 |
| **75%** | NaN | NaN | NaN | NaN | 7800.000000 | 2487.157500 | 43.130000 | 0.000000 | 0.000000 | 0.000000 | 150.000000 |
| **max** | NaN | NaN | NaN | NaN | 10000.000000 | 5342.380000 | 110.750000 | 1503.970000 | 1.000000 | 90.000000 | 180.000000 |

In [ ]:
```
# Checking for missing values
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
account_id             0
acquisition_channel    0
open_date              0
month                  0
credit_line            0
balance                0
revenue                0
charge_off_amount      0
charge_off_flag        0
days_past_due          0
acquisition_cost       0
dtype: int64
```

In [ ]:
```
# Date Conversion
```

In [6]:
```
df['open_date'] = pd.to_datetime(df['open_date'], errors='coerce')
df['month'] = pd.to_datetime(df['month'], format="%Y-%m")
```

In [ ]:
```
# Revenue & Credit Utilization Metrics
```

In [7]:
```
# Credit utilization rate
df['utilization_rate'] = df['balance'] / df['credit_line']

df[['account_id', 'credit_line', 'balance', 'utilization_rate']].head()
```

Out[7]:

| | account_id | credit_line | balance | utilization_rate |
|---|---|---|---|---|
| **0** | A100001 | 4900 | 723.83 | 0.147720 |
| **1** | A100002 | 4800 | 916.81 | 0.191002 |
| **2** | A100003 | 8100 | 872.38 | 0.107701 |
| **3** | A100004 | 2300 | 600.00 | 0.260870 |
| **4** | A100005 | 5700 | 407.63 | 0.071514 |

In [ ]:
```
# Aggregation Example (Monthly Revenue)
```

In [8]:
```
monthly_rev = (
    df.groupby('month')['revenue']
    .sum()
    .reset_index()
    .sort_values('month')
)

monthly_rev
```

Out[8]:

| | month | revenue |
|---|---|---|
| **0** | 2023-01-01 | 651.78 |
| **1** | 2023-02-01 | 1143.94 |
| **2** | 2023-03-01 | 1037.06 |
| **3** | 2023-04-01 | 663.04 |
| **4** | 2023-05-01 | 755.91 |
| **5** | 2023-06-01 | 853.90 |
| **6** | 2023-07-01 | 805.06 |
| **7** | 2023-08-01 | 1057.89 |
| **8** | 2023-09-01 | 684.92 |
| **9** | 2023-10-01 | 963.98 |
| **10** | 2023-11-01 | 684.03 |
| **11** | 2023-12-01 | 693.17 |
| **12** | 2024-01-01 | 701.18 |
| **13** | 2024-02-01 | 861.66 |
| **14** | 2024-03-01 | 798.22 |
| **15** | 2024-04-01 | 1155.81 |
| **16** | 2024-05-01 | 595.42 |
| **17** | 2024-06-01 | 1173.86 |
| **18** | 2024-07-01 | 411.11 |
| **19** | 2024-08-01 | 878.33 |
| **20** | 2024-09-01 | 932.19 |
| **21** | 2024-10-01 | 1055.43 |
| **22** | 2024-11-01 | 1095.11 |
| **23** | 2024-12-01 | 710.60 |

In [ ]:
```
# Channel Acquisition Summary
```

In [9]:
```
channel_summary = df.groupby('acquisition_channel').agg({
    'account_id': 'count',
    'revenue': 'mean',
    'credit_line': 'mean',
    'balance': 'mean',
    'charge_off_flag': 'mean'
}).rename(columns={'account_id': 'num_accounts',
                   'charge_off_flag': 'charge_off_rate'})

channel_summary
```

Out[9]:

| acquisition_channel | num_accounts | revenue | credit_line | balance | charge_off_rate |
|---|---|---|---|---|---|
| **Mail** | 169 | 35.392071 | 5345.562130 | 2042.106391 | 0.005917 |
| **Online** | 310 | 33.752774 | 5601.935484 | 2054.145935 | 0.000000 |
| **Partner** | 121 | 32.388264 | 5552.892562 | 2047.921240 | 0.008264 |

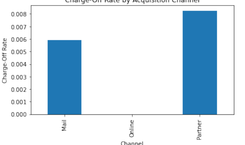
In [ ]:
```
# Monthly Revenue Visulization
```

In [10]:
```
plt.figure(figsize=(8,4))
plt.plot(monthly_rev['month'], monthly_rev['revenue'])
plt.xlabel("Month")
plt.ylabel("Total Revenue")
plt.title("Monthly Revenue Trend")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```


In [ ]:
```
# Charge-Off Rate by Channel
```

In [11]:
```
chargeoff = df.groupby('acquisition_channel')['charge_off_flag'].mean()

chargeoff.plot(kind='bar')
plt.title("Charge-Off Rate by Acquisition Channel")
plt.xlabel("Channel")
plt.ylabel("Charge-Off Rate")
plt.tight_layout()
plt.show()
```




In [ ]:
```

```