Daily Transaction Analysis Report

Project Overview

The Daily Transaction Analysis project examines transaction data to identify daily, weekly, and monthly financial trends. It helps businesses detect peak volumes, unusual fluctuations, and recurring patterns to improve sales planning, resource allocation, and marketing strategies.

Project Description

This project involves importing a transaction dataset containing date-wise transaction records and analyzing the data using Python. The main goals were:

- **Data Cleaning**: Ensuring all dates were correctly formatted and missing values handled.
- **Time Series Aggregation**: Grouping the data by day and visualizing transaction volume.
- **Trend Detection**: Using line plots and rolling averages to observe daily patterns.
- **Seasonal Patterns**: Highlighting specific days or months that consistently show spikes or dips.

Tech Stack

- Python 3
- Jupyter Notebook
- Pandas
- Matplotlib
- Seaborn
- NumPy

Key Finding

- Analyze the trend of Expenses and Income over the period from 2015 to 2018.
- Identify the most frequently used Expense Mode across all transactions.

- Determine the highest Expense Categories and their corresponding Subcategories.
- Find out which Payment Mode accounts for the highest total amount**.**

Code

```
import pandas as pd
from datetime import datetime
```

```
df = pd.read_csv(r"C:\Rahul Data\Project 2\Data\Daily Transaction\Daily Household Transactions.csv")
```

```
df.head()
```

	Date	Mode	Category	Subcategory	Note	Amount	Income/Expense	Currency
0	20/09/2018 12:04:08	Cash	Transportation	Train	2 Place 5 to Place 0	30.0	Expense	INR
1	20/09/2018 12:03:15	Cash	Food	snacks	Idli medu Vada mix 2 plates	60.0	Expense	INR
2	19/09/2018	Saving Bank account 1	subscription	Netflix	1 month subscription	199.0	Expense	INR
3	17/09/2018 23:41:17	Saving Bank account 1	subscription	Mobile Service Provider	Data booster pack	19.0	Expense	INR
4	16/09/2018 17:15:08	Cash	Festivals	Ganesh Pujan	Ganesh idol	251.0	Expense	INR

```
# check missing value
df.isnull().sum()
```

```
Date                0
Mode                0
Category            0
Subcategory       635
Note              521
Amount              0
Income/Expense      0
Currency            0
dtype: int64
```

```
# filling the missing value
df.fillna("unknown",inplace = True)
```

```
df.isnull().sum()
```

```
Date              0
Mode              0
Category          0
Subcategory       0
Note              0
Amount            0
Income/Expense    0
Currency          0
dtype: int64
```

```
#summary statistics for numerical data
df.describe()
```

	Amount
count	2461.000000
mean	2751.145380
std	12519.615804
min	2.000000
25%	35.000000
50%	100.000000
75%	799.000000
max	250000.000000

```
df.head(10)
```

	Date	Mode	Category	Subcategory	Note	Amount	Income/Expense	Currency
0	20/09/2018 12:04:08	Cash	Transportation	Train	2 Place 5 to Place 0	30.0	Expense	INR
1	20/09/2018 12:03:15	Cash	Food	snacks	Idli medu Vada mix 2 plates	60.0	Expense	INR
2	19/09/2018	Saving Bank account 1	subscription	Netflix	1 month subscription	199.0	Expense	INR
3	17/09/2018 23:41:17	Saving Bank account 1	subscription	Mobile Service Provider	Data booster pack	19.0	Expense	INR
4	16/09/2018 17:15:08	Cash	Festivals	Ganesh Pujan	Ganesh idol	251.0	Expense	INR
5	15/09/2018 06:34:17	Credit Card	subscription	Tata Sky	Permanent Residence - Tata Play recharge	200.0	Expense	INR
6	14/09/2018 05:39:17	Cash	Transportation	auto	Place 2 station to Permanent Residence	50.0	Expense	INR
7	13/09/2018 21:35:15	Saving Bank account 1	Transportation	Train	2 Place 0 to Place 3	40.0	Expense	INR
8	13/09/2018 21:01:47	Credit Card	Other	unknown	HBR 2 Months subscription	83.0	Expense	INR
9	13/09/2018 21:01:32	Cash	Food	Grocery	1kg atta	46.0	Expense	INR

```
print(df.dtypes)
```

```
Date              object
Mode              object
Category          object
Subcategory       object
Note              object
Amount           float64
Income/Expense    object
Currency          object
dtype: object
```

```
print(df['Date'])
```

```
0       20/09/2018 12:04:08
1       20/09/2018 12:03:15
2                19/09/2018
3       17/09/2018 23:41:17
4       16/09/2018 17:15:08
               ...
2456           1/1/2015
2457           1/1/2015
2458           1/1/2015
2459           1/1/2015
2460           1/1/2015
Name: Date, Length: 2461, dtype: object
```

```
df['Date'] = df['Date'].astype(str).str.strip()
df['Date'] = df['Date'].apply(lambda x: x if ':' in x else x + ' 00:00:00')
```

```
print(df['Date'])
```

```
0       20/09/2018 12:04:08
1       20/09/2018 12:03:15
2       19/09/2018 00:00:00
3       17/09/2018 23:41:17
4       16/09/2018 17:15:08
               ...
2456      1/1/2015 00:00:00
2457      1/1/2015 00:00:00
2458      1/1/2015 00:00:00
2459      1/1/2015 00:00:00
2460      1/1/2015 00:00:00
Name: Date, Length: 2461, dtype: object
```

```
df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%Y %H:%M:%S', errors='coerce')
```

```
df['Date'] = df['Date'].dt.date
print(df['Date'].head(10))
```

```
0    2018-09-20
1    2018-09-20
2    2018-09-19
3    2018-09-17
4    2018-09-16
5    2018-09-15
6    2018-09-14
7    2018-09-13
8    2018-09-13
9    2018-09-13
Name: Date, dtype: object
```

```
df['Date'] = pd.to_datetime(df['Date'], dayfirst =True,errors='coerce')
```

```
print(df[df['Date'].isna()])
```

```
Empty DataFrame
Columns: [Date, Mode, Category, Subcategory, Note, Amount, Income/Expense, Currency]
Index: []
```

```
df['Date'] = df['Date'].dt.date
```

```
print(df.dtypes)
```

```
Date              datetime64[ns]
Mode                      object
Category                  object
Subcategory               object
Note                      object
Amount                   float64
Income/Expense            object
Currency                  object
Month                  period[M]
dtype: object
```

```
print(df.head())
```

```
         Date                  Mode        Category              Subcategory  \
0  2018-09-20                  Cash  Transportation                    Train
1  2018-09-20                  Cash            Food                   snacks
2  2018-09-19  Saving Bank account 1    subscription                  Netflix
3  2018-09-17  Saving Bank account 1    subscription  Mobile Service Provider
4  2018-09-16                  Cash       Festivals             Ganesh Pujan

                        Note  Amount Income/Expense Currency
0         2 Place 5 to Place 0    30.0        Expense      INR
1  Idli medu Vada mix 2 plates    60.0        Expense      INR
2          1 month subscription   199.0        Expense      INR
3             Data booster pack    19.0        Expense      INR
4                   Ganesh idol   251.0        Expense      INR
```

```
# count of transaction by category
df['Category'].value_counts()
```

```
Category
Food                        907
Transportation              307
Household                   176
subscription                143
Other                       126
Investment                  103
Health                       94
Family                       71
Recurring Deposit            47
Apparel                      47
Money transfer               43
Salary                       43
Gift                         30
Public Provident Fund        29
Equity Mutual Fund E         22
Beauty                       22
Gpay Reward                  21
Education                    18
maid                         17
Saving Bank account 1        17
Festivals                    16
Equity Mutual Fund A         14
```

```
# Total amount spent per category
df.groupby('Category')['Amount'].sum().sort_values(ascending = False)
```

```
Category
Salary                      2526576.45
Money transfer               606528.90
Fixed Deposit                450000.00
Maturity amount              382792.00
Public Provident Fund        345000.00
Share Market                 276161.00
Saving Bank account 1        274798.57
Investment                   271858.00
Other                        170467.28
Transportation               169053.78
Household                    161645.58
subscription                 114587.91
Equity Mutual Fund B         100000.00
Food                          96403.10
```

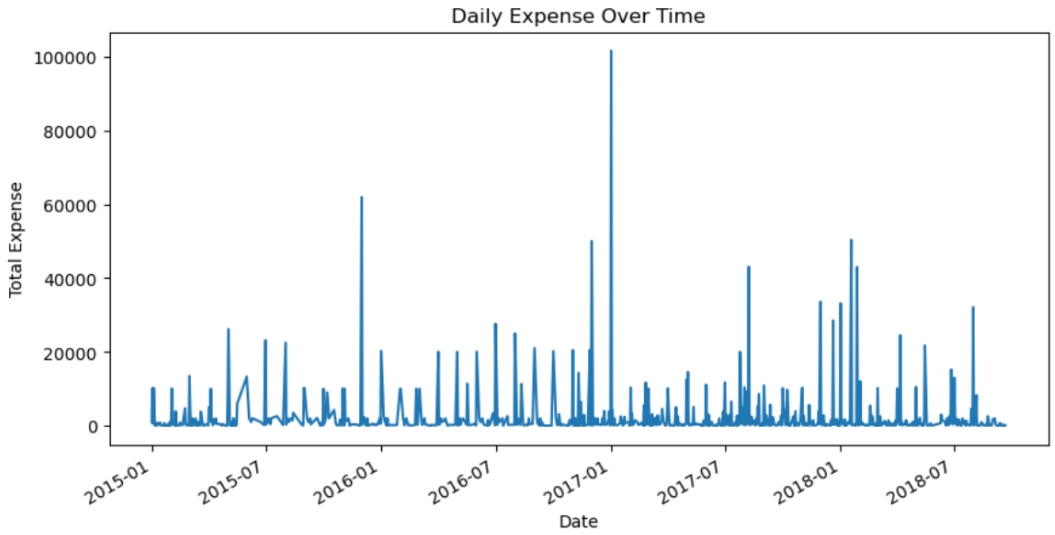
```
# Total amount spent per category
top_spending = df[df["Income/Expense"] == "Expense"].groupby("Category")["Amount"].sum()
```

```
top_spending.sort_values(ascending=False).head()
```

```
Category
Money transfer    606528.90
Investment        271858.00
Transportation    169053.78
Household         161645.58
subscription      114587.91
Name: Amount, dtype: float64
```

```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# convert Date column to datetime
df['Date'] = pd.to_datetime(df['Date'],errors = 'coerce')
# plot expenses over time
plt.figure(figsize=(10,5))
df[df['Income/Expense'] == 'Expense'].groupby('Date')['Amount'].sum().plot()
plt.xlabel("Date")
plt.ylabel("Total Expense")
plt.title("Daily Expense Over Time")
plt.show()
```




```
# Total Income and Expense
income = df[df["Income/Expense"] == "Income"]["Amount"].sum()
expense = df[df["Income/Expense"] == "Expense"]["Amount"].sum()
print(f"Total Income: Rs {income}")
print(f"Total Expense: Rs {expense}")
print(f"Net Savings: Rs {income - expense}")
```

```
Total Income: Rs 3042397.3499999996
Total Expense: Rs 1957390.53
Net Savings: Rs 1085006.8199999996
```

```
# Create a 'Month' column
df['Month'] = pd.to_datetime(df['Date']).dt.to_period('M')
```

```
# Group by Month and Income/Expense
monthly_summary = df.groupby(['Month', 'Income/Expense'])['Amount'].sum().unstack().fillna(0)
```

```
# View last 12 months
print(monthly_summary.tail(12))
```

```
Income/Expense    Expense     Income  Transfer-Out
Month
2017-10          45532.09    6000.00     176000.00
2017-11          64364.00  143155.00      29120.33
2017-12          46805.00   82321.00     548108.96
2018-01         142080.90  292938.00      94097.14
2018-02          26218.00   64738.00      91704.00
2018-03          23396.75   69343.50      23042.88
2018-04          48339.58   65824.15      25500.00
2018-05          40670.00   69238.00      36543.00
2018-06          38161.02   68551.00      36549.59
2018-07          67738.36   77267.50      25500.00
2018-08          21305.65   71735.75      36543.00
2018-09           4724.00    3500.00      25500.00
```

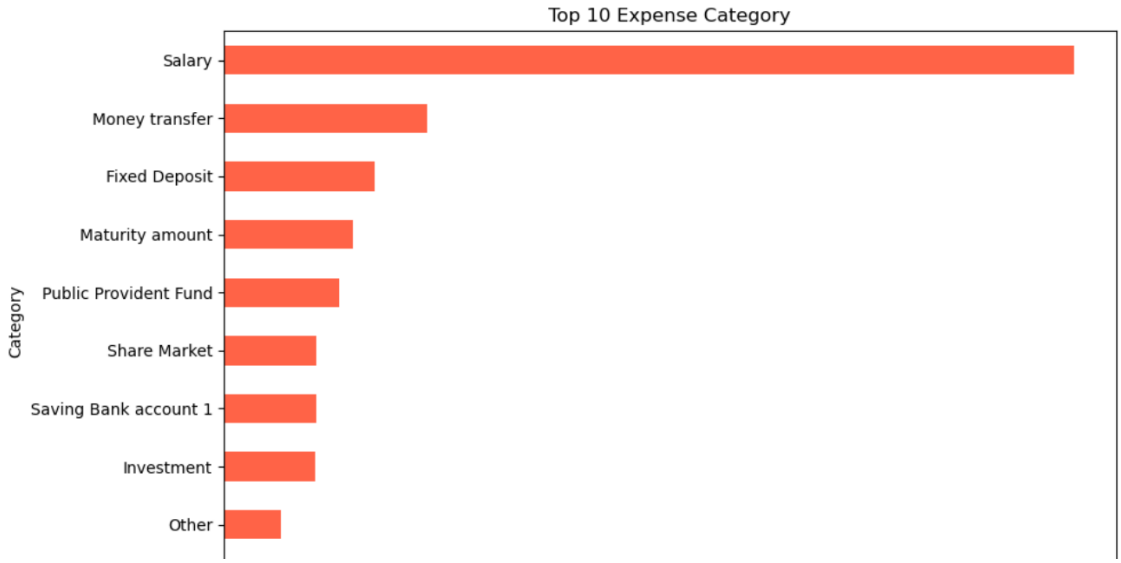
```
expense_df = df[df['Income/Expense']=='Expense']
```

```
category_summery = df.groupby('Category')['Amount'].sum().sort_values(ascending=False)
```

```
print(category_summery.head(10))
```

```
Category
Salary                   2526576.45
Money transfer            606528.90
Fixed Deposit             450000.00
Maturity amount           382792.00
Public Provident Fund     345000.00
Share Market              276161.00
Saving Bank account 1     274798.57
Investment                271858.00
Other                     170467.28
Transportation            169053.78
Name: Amount, dtype: float64
```

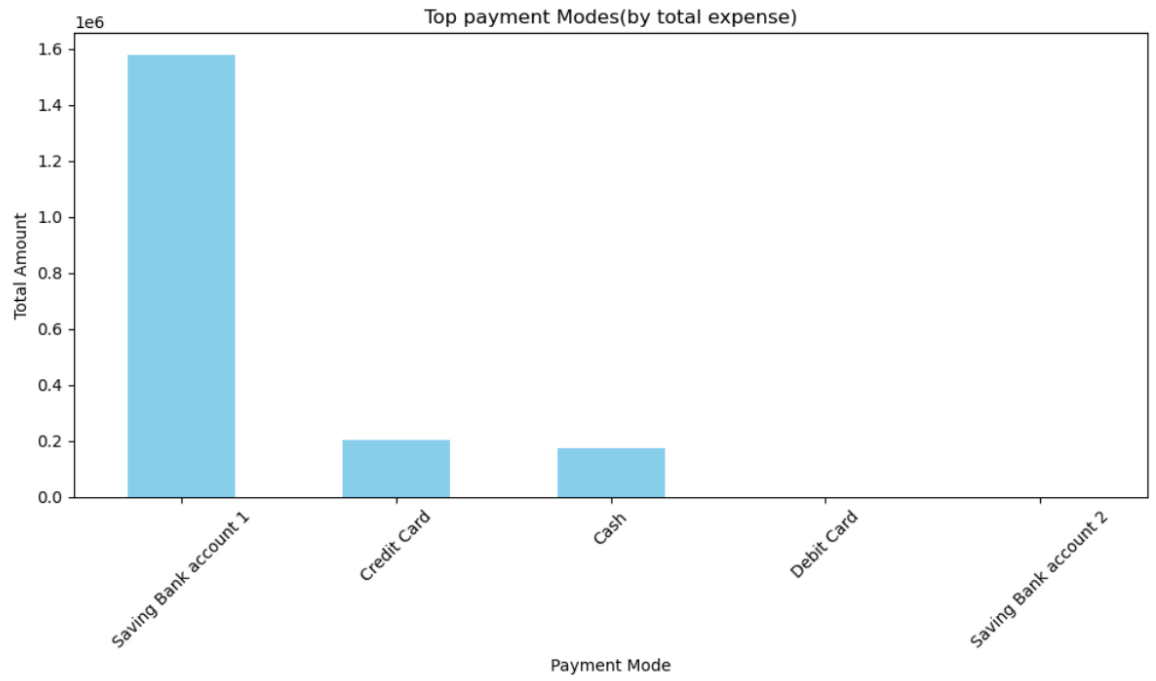
```
top_categories = category_summery.head(10)
plt.figure(figsize=(10,6))
top_categories.plot(kind = 'barh', color='tomato')
plt.title('Top 10 Expense Category')
plt.xlabel('Total Amount')
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```




```
expense_df = df[df['Income/Expense']=='Expense']
mode_summary = expense_df.groupby('Mode')['Amount'].sum().sort_values(ascending=False)
print(mode_summary.head(10))
```

```
Mode
Saving Bank account 1    1577728.46
Credit Card               205254.01
Cash                      173431.00
Debit Card                   942.36
Saving Bank account 2         34.70
Name: Amount, dtype: float64
```

```
top_mode = mode_summary.head(10)
plt.figure(figsize = (10,6))
top_mode.plot(kind = 'bar', color ='skyblue')
plt.title('Top payment Modes(by total expense)')
plt.ylabel('Total Amount')
plt.xlabel('Payment Mode')
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```

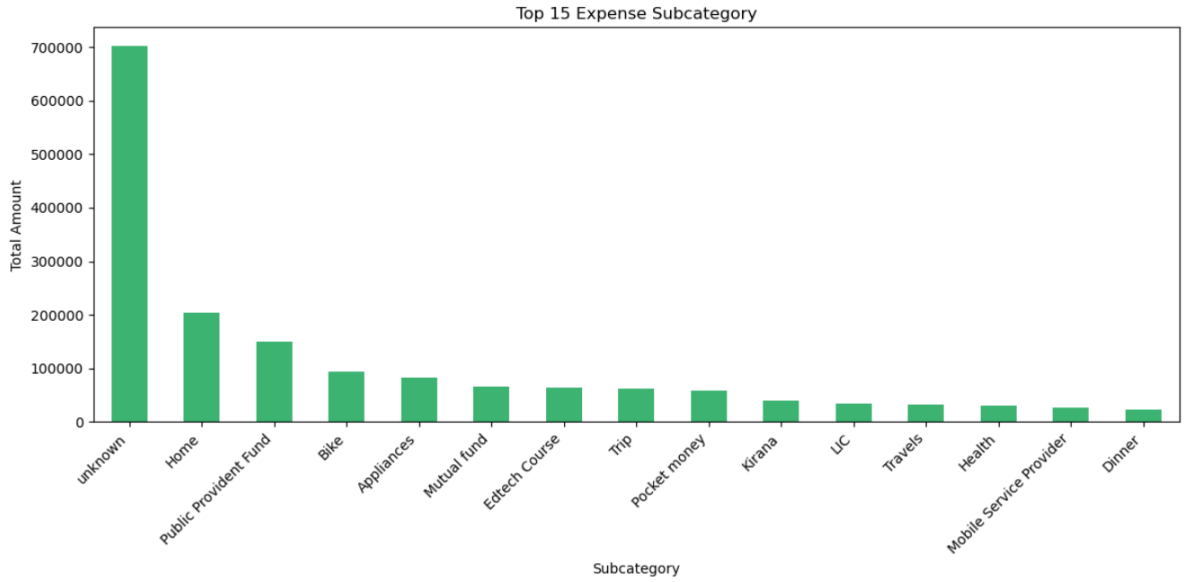



```
expense_df = df[df['Income/Expense']== 'Expense']
subcategory_summary = expense_df.groupby('Subcategory')['Amount'].sum().sort_values(ascending= False)
print(subcategory_summary.head(10))
```

```
Subcategory
unknown                 701672.28
Home                    204505.90
Public Provident Fund   150000.00
Bike                     94593.00
Appliances               82081.00
Mutual fund              66000.00
Edtech Course            63733.42
Trip                     63300.00
Pocket money             58195.00
Kirana                   39147.58
Name: Amount, dtype: float64
```

```
top_subcategory= subcategory_summary.head(15)
```

```
plt.figure(figsize=(12,6))
top_subcategory.plot(kind='bar', color = 'mediumseagreen')
plt.title('Top 15 Expense Subcategory')
plt.ylabel('Total Amount')
plt.xlabel('Subcategory')
plt.xticks(rotation=45,ha='right')
plt.tight_layout()
plt.show()
```


Top 15 Expense Subcategory
Total Amount
0
100000
200000
300000
400000
500000
600000
700000
unknown
Home
Public Provident Fund
Bike
Appliances
Mutual fund
Edtech Course
Trip
Pocket money
Kirana
LIC
Travels
Health
Mobile Service Provider
Dinner
Subcategory