Author : Part 1: Mohit Bansal (Manager, Advisory, PwC India)

Section 1: Import Data and read the file

```
#**Connect Google Colab to My google drive**
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

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

bank_1 = pd.read_csv('/content/drive/MyDrive/Bank_Data/Bank_Data_v1.csv')
bank_1['month_year'] = pd.to_datetime(bank_1['post_date']).dt.to_period('M')
print(bank_1)
```

```
           Lead ID  bankid  ...                Primary_key month_year
     0       308148    8535  ...  308148_8535_12460_42439_1    2016-03
     1       308148    8535  ...  308148_8535_12460_42450_1    2016-03
     2       308148    8535  ...  308148_8535_12460_42450_3    2016-03
     3       308148    8535  ...  308148_8535_12460_42450_2    2016-03
     4       308148    8535  ...  308148_8535_12460_42451_1    2016-03
     ...        ...     ...  ...                        ...        ...
     29024   330698    8545  ...  330698_8545_14374_42772_7    2017-02
     29025   330698    8545  ...  330698_8545_14374_42773_2    2017-02
     29026   330698    8545  ...  330698_8545_14374_42773_4    2017-02
     29027   330698    8545  ...  330698_8545_14374_42773_3    2017-02
     29028   330698    8545  ...  330698_8545_14374_42773_1    2017-02
```

Saved successfully!

```
bank_1.head(5)
```

	Lead ID	bankid	bank_account_id	account_number	Industry	post_date	descripti
0	308148	8535	12460	xxxx9928	Accommodation and Food Services	10-Mar-16	DEPOSIT NUMB xx68
1	308148	8535	12460	xxxx9928	Accommodation and Food Services	21-Mar-16	ATM CA DEPOS 03/21 2904 BELT LI RD IR
2	308148	8535	12460	xxxx9928	Accommodation and Food Services	21-Mar-16	ATM CA DEPOS 03/21 2904 BELT LI

Section 2: The data file consists of Blank Rows (total rows = 144836). Removing blank rows to keep the data.

```
bank_2 = bank_1.dropna(how='all')
bank_2.tail()
# Commenting Below link to avoid running it multiple time, please uncomment it once while rur
# del bank_2['month name']
bank_2.to_csv('/content/drive/MyDrive/Bank_Data/Bank_data_v2.csv', index=False)
```

```
bank_3 = pd.read_csv('/content/drive/MyDrive/Bank_Data/Bank_data_v2.csv')
bank_3.tail(5)
```

	Lead ID	bankid	bank_account_id	account_number	Industry	post_date	descrip
29024	330698	8545	14374	1693	Health Care and Social Assistance	06-Feb-17	Wire tra withdr JAN ASSOCIA
29025	330698	8545	14374	1693	Health Care and Social Assistance	07-Feb-17	ACH de BANKC BTOT PERS DOCTOR
					Health		Withdr

Saved successfully!

Section 3: Understand the Descriptive stats for the data.

Analyse and identify the patterns between variables.

Analyse the outliers and correlations between variables which might produce bias in the data.

Double-click (or enter) to edit

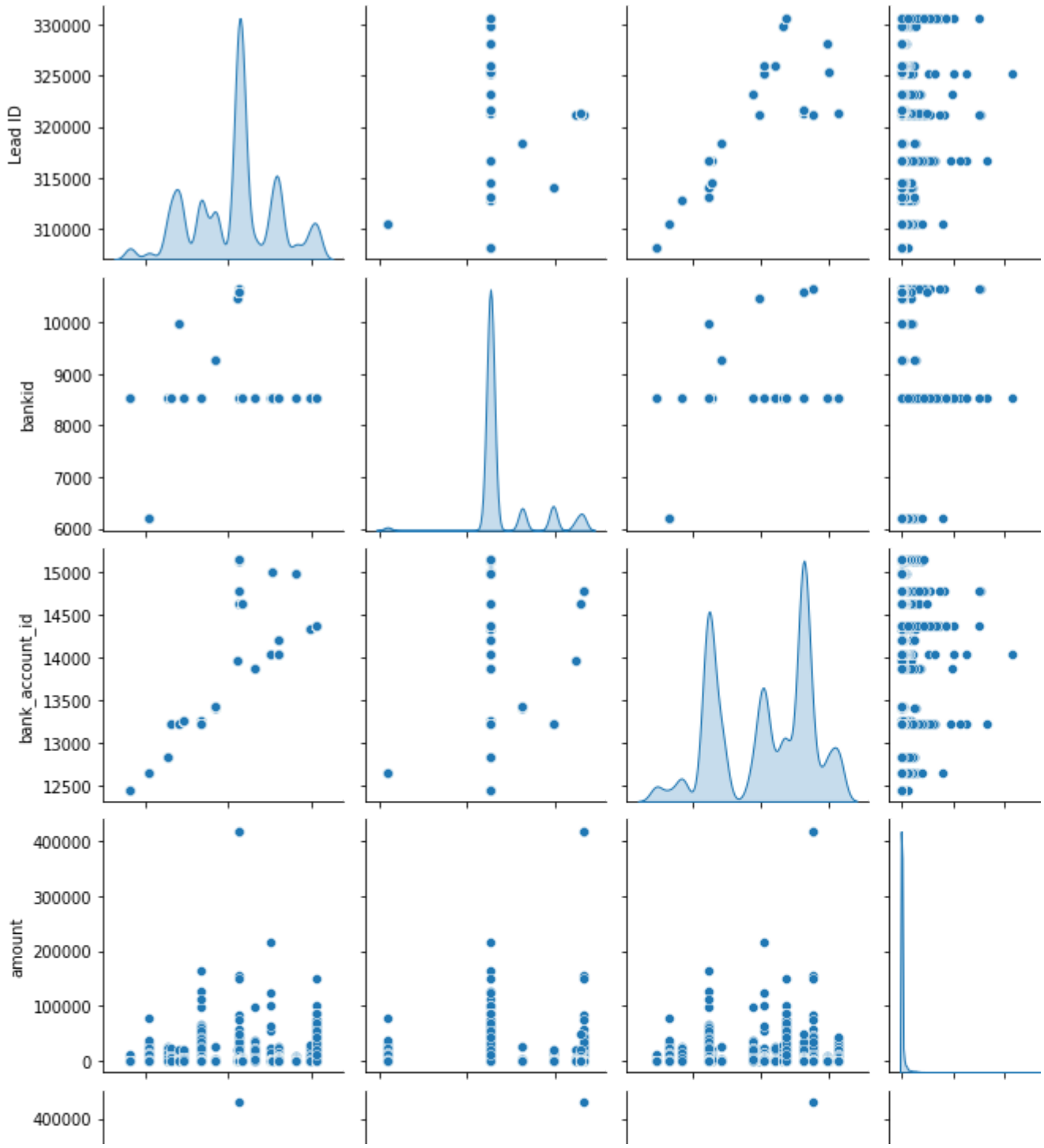
```
del bank_3['month name']
bank_3.describe()
```

	Lead ID	bankid	bank_account_id	amount	running_balance	tr
count	29029.000000	29029.000000	29029.000000	29029.000000	29029.000000	290
mean	320714.718282	8823.888491	14052.030556	1927.453475	10247.851975	
std	5051.317064	683.963276	703.164708	7450.638452	20055.683105	
min	308148.000000	6192.000000	12460.000000	0.000000	-84727.980000	
25%	316728.000000	8534.000000	13234.000000	29.250000	798.850000	
50%	321380.000000	8535.000000	14049.000000	160.500000	3820.940000	
75%	323253.000000	8545.000000	14636.000000	788.460000	11898.880000	
max	330698.000000	10656.000000	15148.000000	419000.000000	437942.290000	

```
sns.pairplot(bank_3, diag_kind='kde')
```

Saved successfully!

```
<seaborn.axisgrid.PairGrid at 0x7fbdc244eb70>
```



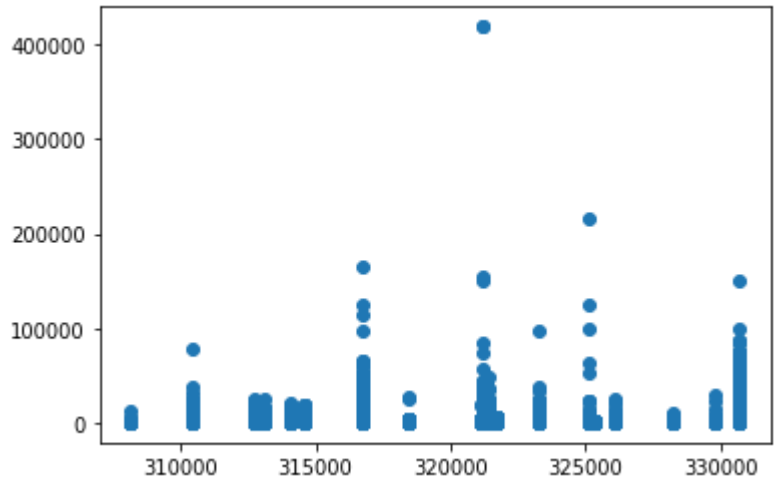

```
bank_3.plot(x='trans_order', y='amount', color='black', linestyle= 'dashed' , linewidth = 1,
        marker='o', markerfacecolor='red', markersize=4)
```

Saved successfully!

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdcd676550>
```

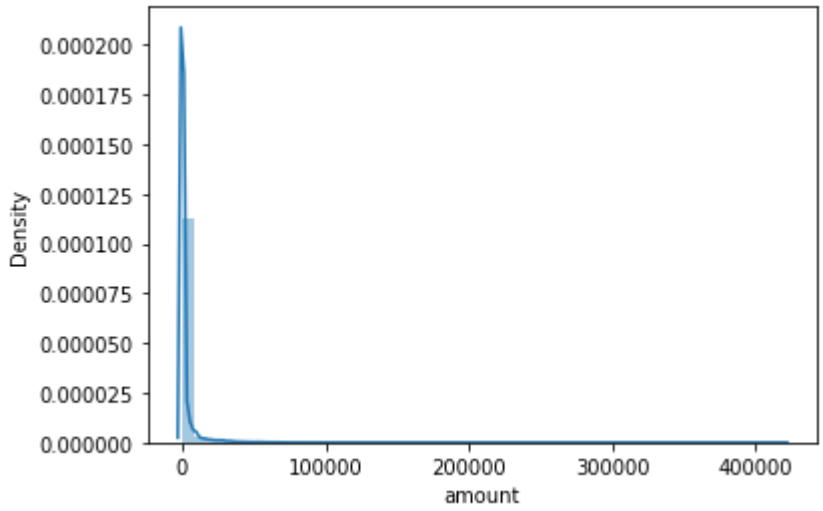

```
plt.scatter( bank_3['Lead ID'],bank_3['amount'])
```

```
<matplotlib.collections.PathCollection at 0x7fbdcc783e80>
```




```
sns.distplot(bank_3['amount'])
```

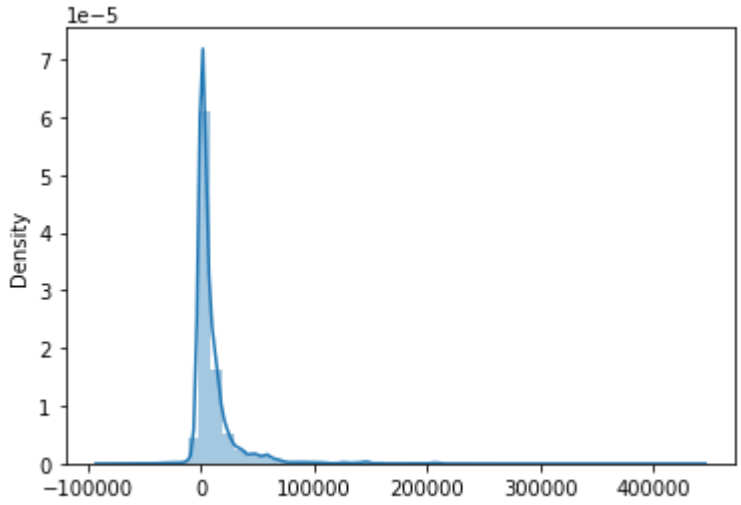
```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureWarning: `d
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f3c24992400>
```




```
sns.distplot(bank_3['running_balance'])
```

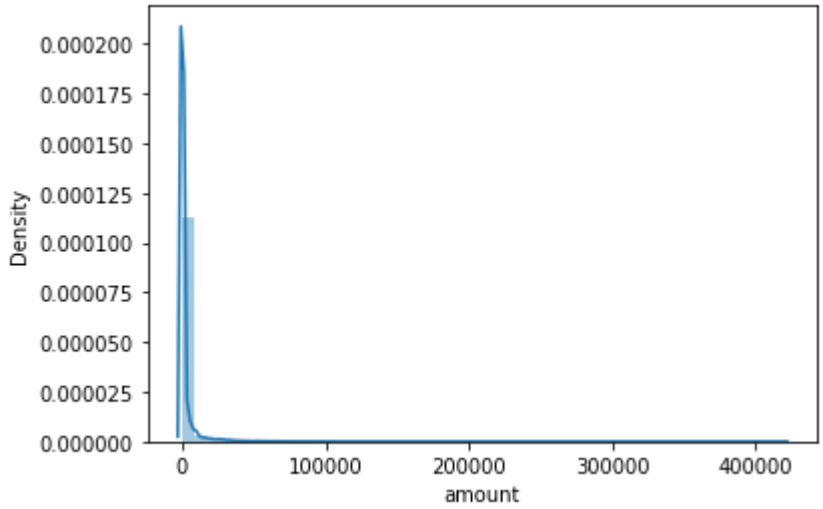
Saved successfully!

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureWarning: `d
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f3c2493b4e0>
```




```
sns.distplot(bank_3['amount'])
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/distributions.py:2551: FutureWarning: `d
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f3c247f2b00>
```



Section 4: From the 2 histogram plots below, we understand the pattern between :

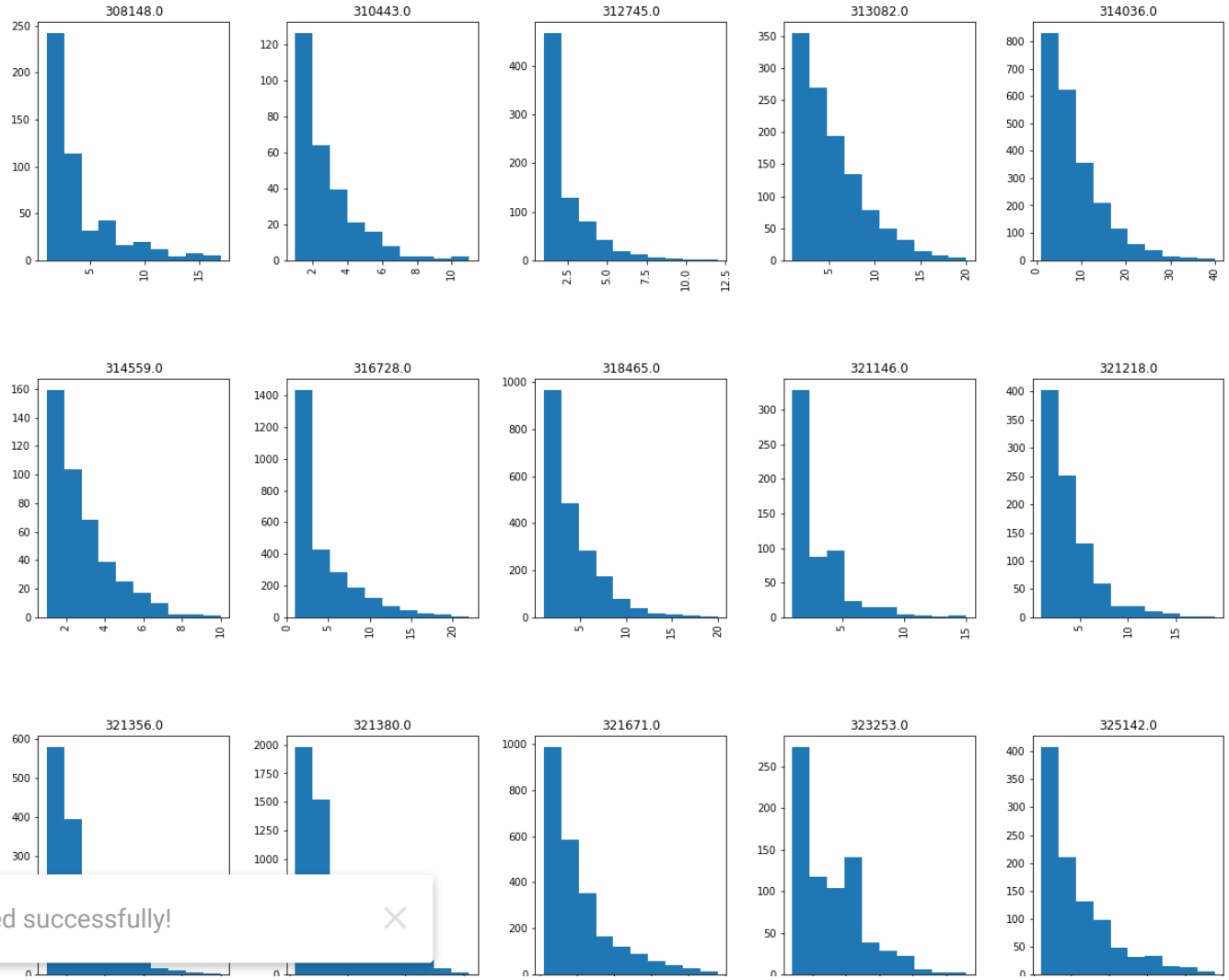
1. Lead ID and Trans_order - Out of 21 Lead IDs, 4 of them have more than 30 transactions on their accounts.

Saved successfully!

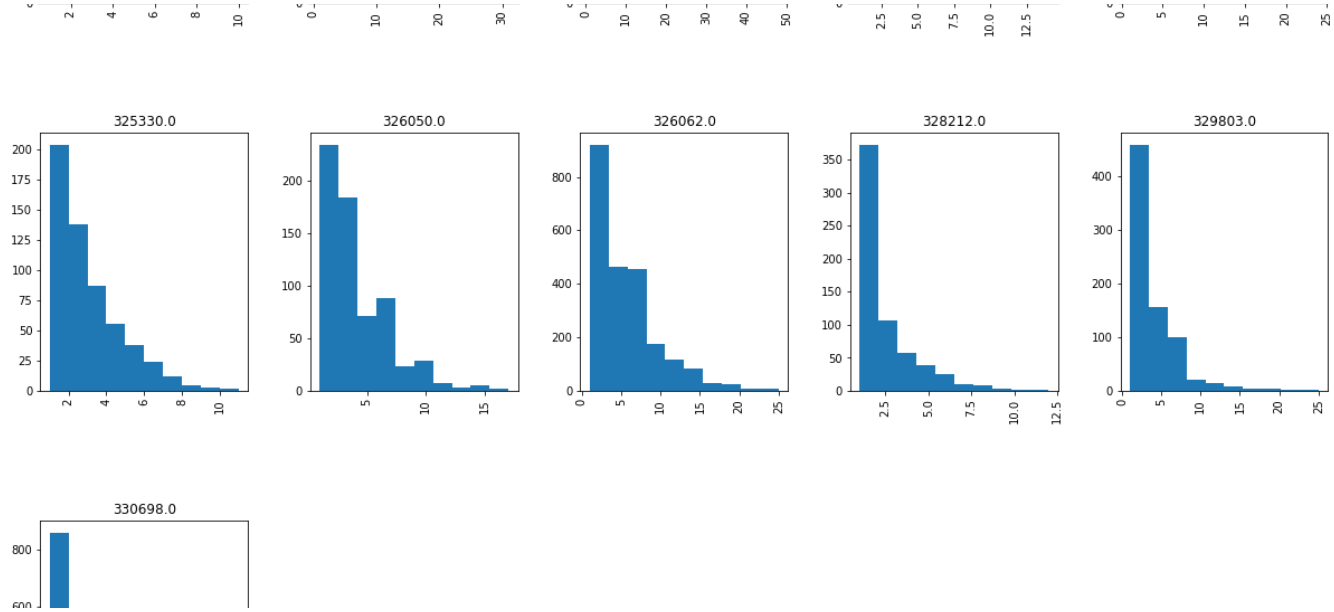
2. Trans_order Vs Amount - Majority of transaction amount is till 21 transactions (all customers together 22222)

```
bank_3.hist(by='Lead ID',column='trans_order',figsize=(20,30))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24790e48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2466ffd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2462d3c8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c245dc748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2460fac8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c245c1e48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24580208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24532550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c245325c0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24517c88>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c244cafd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c244883c8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24438748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c243ebac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c243a0e48>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2435d208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2438f588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24340908>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c242f3c88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c242a7fd0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c242643c8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24296748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24248ac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c241fbe48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c241ba208>]],
      dtype=object)
```



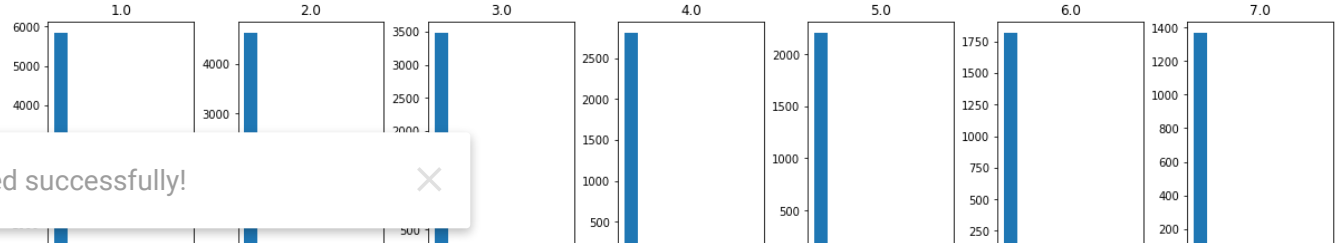
Saved successfully!



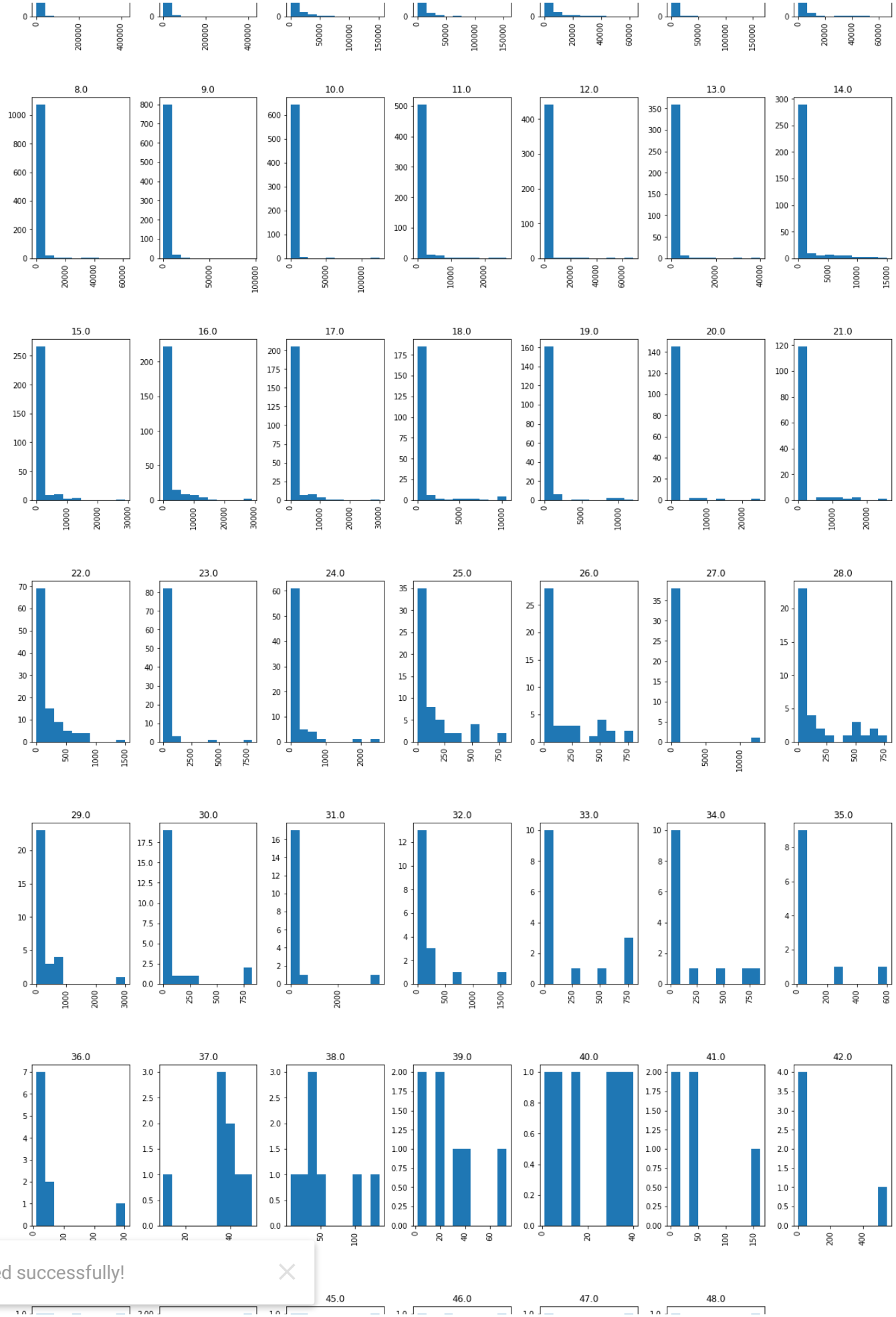

```
bank_3.hist(by='trans_order',column='amount',figsize=(20,40))
#From the 2 histogram plots, we understand the pattern between :
#1. Lead ID and Trans_order - Out of 21 Lead IDs, 4 of them have more than 30 transactions or
#2. Trans_order Vs Amount - Majority of transaction amount is till 21 transactions (all amour
```

Saved successfully!

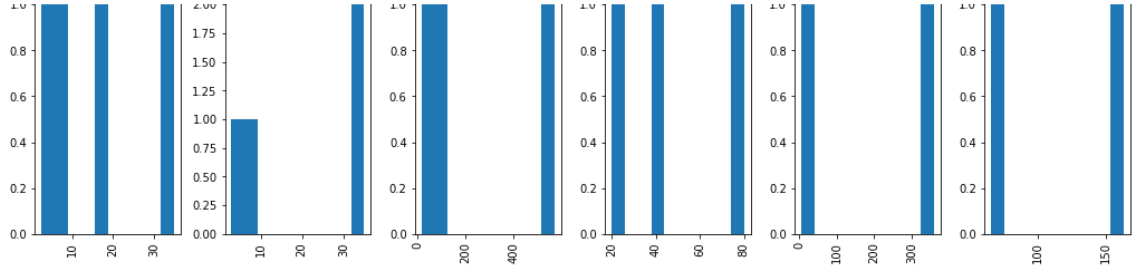
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2400fdd8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23c5f6a0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23c50ac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23bfbe48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23bb8208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23b68588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23b1c908>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23b4dc50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23b4dcc0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23abc3c8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23a6f748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c26d40940>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24b36208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24ac35c0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2413acf8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c24b23cf8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c239704e0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23923748>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c239579b0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23908c18>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c238bfe80>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23881128>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23832390>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c237e85f8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2379b860>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c237cfac8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23782d30>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23738f98>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c236f9240>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c236ab4a8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23660710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23694978>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23647be0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c235fbe48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c235be0f0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23571358>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c235235c0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23557828>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2350ba90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c234bdcf8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23474f60>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2342ad30>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f3c233eb0f0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c2339c470>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c233ce7f0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23381b70>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c23332ef0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c232f22b0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f3c232a2630>]],
      dtype=object)
```



Saved successfully!



Saved successfully!



```
# Compute the correlation matrix
corr = bank_3.corr()
print(corr)
```

```
                   Lead ID    bankid  ...  running_balance  trans_order
Lead ID           1.000000 -0.158936  ...         0.063881    -0.008326
bankid           -0.158936  1.000000  ...        -0.053088     0.001317
bank_account_id   0.700699 -0.050178  ...        -0.041772     0.076003
amount            0.179994 -0.057082  ...         0.181640    -0.078309
running_balance   0.063881 -0.053088  ...         1.000000     0.022697
trans_order      -0.008326  0.001317  ...         0.022697     1.000000

[6 rows x 6 columns]
```

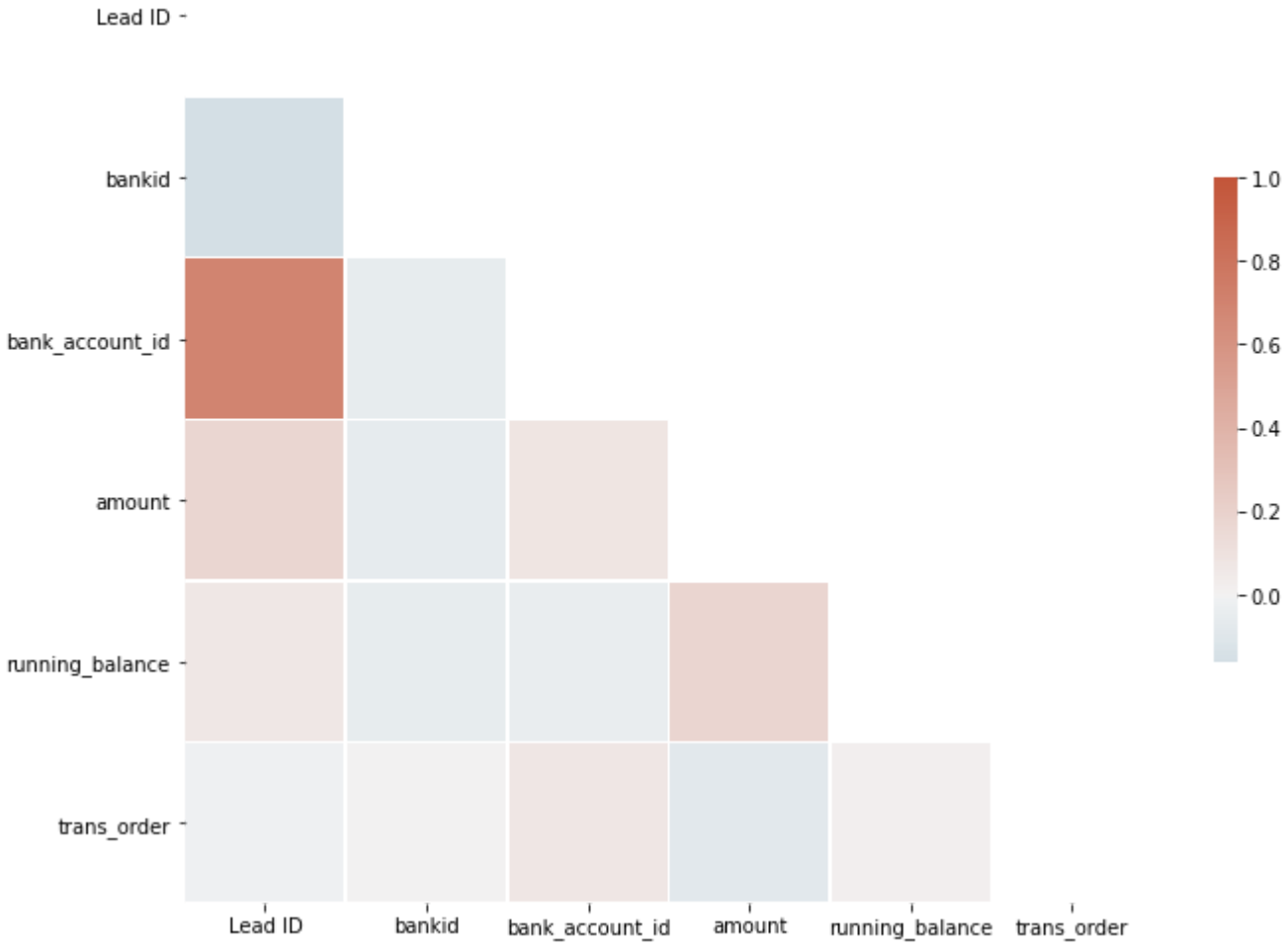
```
# Generate a mask for the upper triangle
mask = np.triu(np.ones_like(corr, dtype=bool))
```

Saved successfully!

```
f, ax = plt.subplots(figsize=(11, 9))
```

```
f, ax = plt.subplots(figsize=(11, 9))

# Generate a custom diverging colormap
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, mask=mask, cmap=cmap, vmax=1, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbdc0dfb0f0>
```



Data Summarization: To generate outputs as per requirement

Section 5: **Problem Statement** : This is a sample of individual bank statement transactions from 20 small business merchants (indexed by Lead ID) from various industries

Each merchant could have multiple accounts (bank account

S

Saved successfully!

Transactions may be debits (withdrawals) or credits (deposits)

Each transaction also has a description

1(i) the number of bank accounts for each merchant

```
# 1. (i) the number of bank accounts for each merchant
Data_1 = bank_3.groupby('Lead ID')['bank_account_id'].nunique()
print(Data_1)
Data_1.to_csv('/content/drive/MyDrive/Bank_Data/Data_1.csv', index=True)
```

```
Lead ID
308148    1
310443    2
312745    5
313082    1
314036    1
314559    2
316728    6
318465    7
321146    2
321218    1
321356    5
321380    8
321671    1
323253    2
325142    1
325330    1
326050    1
326062    3
328212    1
329803    1
330698    1
Name: bank_account_id, dtype: int64
```

1(ii) the number of months of each bank account for which data is available

```
ach bank account for which data is available
```

Saved successfully!

```
#bank_3['month_year'] = pd.to_datetime(bank_3['post_date']).dt.to_period('M')
Data_2 = bank_3.groupby('bank_account_id')['month_year'].nunique()
print(Data_2)
Data_2.to_csv('/content/drive/MyDrive/Bank_Data/Data_2.csv', index=True)
```

```
bank_account_id
12460    10
12654     7
12655    12
12835     5
12836     5
12837     1
12838    11
12839    13
13226    11
13228    12
13230     7
13232    13
13233    10
13234    13
13235    11
13271    12
13272     8
13273     7
13419     9
13420     3
13421     2
13422     7
13423     8
13424     9
13425     8
13877    11
13879     8
13970     9
13971     4
14044    11
14045     2
14046     8
14049     8
14206     8
14339    13
14374    13
14629    13
14630     2
14631     2
14632     2
14633     2
14634    12
14635    12
14636    13
14637    13
14779    13
14981    13
```

```
15146    13
15147    11
15148    11
Name: month_year, dtype: int64
```

1(iii) the total number of credits (deposits), debits (withdrawals) and their averages per month for each bank account and each merchant

```
# 1. (iii) the total number of credits (deposits), debits (withdrawals) and their averages pe
# Mean refers to the averages
Data_3 = pd.pivot_table(bank_3,index=['Lead ID','bank_account_id','month_year'],columns=['tra
print(Data_3.round(decimals=0))
Data_3.to_csv('/content/drive/MyDrive/Bank_Data/Data_3.csv', index=True)
```

```
                                    len           mean
                                 amount         amount
transaction_type                 credit  debit   credit    debit
Lead ID bank_account_id month_year
308148  12460           2016-03     5.0   15.0   1220.0    333.0
                        2016-04     1.0   12.0   1500.0    216.0
                        2016-05     4.0    8.0    194.0     99.0
                        2016-06     4.0   13.0    592.0    156.0
                        2016-07    20.0   30.0    405.0    280.0
...                                 ...    ...      ...      ...
330698  14374           2016-10    27.0   77.0  33553.0  11766.0
                        2016-11    26.0   85.0  32040.0   9800.0
                        2016-12    33.0   93.0  27355.0   9495.0
                        2017-01    32.0  196.0  28218.0   4656.0
                        2017-02     6.0   38.0  32059.0   5011.0

[478 rows x 4 columns]
```

1(iv) the total dollar value of credits, debits and their averages per month for each bank account and each merchant

Mean refers to the averages

```
#1. (iv) the total dollar value of credits, debits and their averages per month for each bank

                                                    x=['Lead ID','bank_account_id','month_year'],columns=['tra
```

Saved successfully!

```
print(Data_4.round(decimals=2))
Data_4.to_csv('/content/drive/MyDrive/Bank_Data/Data_4.csv', index=True)
```

```
                                              sum                mean
                                           amount              amount
transaction_type                           credit      debit    credit     debit
Lead ID bank_account_id month_year
308148  12460           2016-03          6098.62    4995.32   1219.72    333.02
                        2016-04          1500.00    2595.75   1500.00    216.31
                        2016-05           774.95     791.74    193.74     98.97
                        2016-06          2369.38    2028.87    592.34    156.07
                        2016-07          8105.04    8396.89    405.25    279.90
...                                           ...        ...       ...       ...
330698  14374           2016-10        905930.37  905998.55  33552.98  11766.21
                        2016-11        833031.28  833031.28  32039.66   9800.37
                        2016-12        902727.28  883050.45  27355.37   9495.17
                        2017-01        902978.48  912591.12  28218.08   4656.08
                        2017-02        192354.78  190403.36  32059.13   5010.61

[478 rows x 4 columns]
```

1. (v) Aggregate the answers to (iii) and (iv) at the merchant level, industry level and bankid level

1(v) Aggregate the answers to (iii) and (iv) at the merchant level, industry level and bankid level

```
#1. (v) Aggregate the answers to (iii) and (iv) at the merchant level, industry level and bar
Data_5 = pd.pivot_table(bank_3,index=['Lead ID','Industry','bankid'],columns=['transaction_ty
print(Data_5.round(decimals=0))
Data_5.to_csv('/content/drive/MyDrive/Bank_Data/Data_5.csv', index=True)
```

```
                                                                  len  ...     mean
                                                               amount  ...   amount
transaction_type                                                credit  ...    debit
Lead ID Industry                                       bankid          ...
308148  Accommodation and Food Services                8535     172.0  ...    193.0
310443  Construction                                   6192      35.0  ...   1409.0
312745  Professional, Scientific, and Technical Services 8544    160.0  ...   1189.0
313082  Professional, Scientific, and Technical Services 8535    112.0  ...    545.0
314036  Retail Trade                                   9966     276.0  ...    218.0
314559  Information Technology                         8534     143.0  ...   1830.0
316728  Construction                                   8534      30.0  ...   1350.0
                                                       8544     268.0  ...   1031.0
318465  Health Care and Social Assistance              9262     369.0  ...    136.0
321146  Retail Trade                                   10479    121.0  ...    222.0
321218  Agriculture, Forestry, Fishing and Hunting     10656     57.0  ...   1727.0
321356  Other Services (except Public Administration)  8544     697.0  ...   5257.0
                                                       8534    4436.0  ...   3661.0
                                                       10591    503.0  ...   1879.0
                                                       8534     642.0  ...    172.0
```

Saved successfully!

```
323253  Retail Trade                                  8545     308.0  ...   2330.0
325142  Other Services (except Public Administration)  8535     115.0  ...    529.0
325330  Retail Trade                                  8544     312.0  ...    312.0
326050  Accommodation and Food Services               8534     190.0  ...    591.0
326062  Accommodation and Food Services               8535     852.0  ...    587.0
328212  Educational Services                          8535     282.0  ...    507.0
329803  Construction                                  8535     116.0  ...    324.0
330698  Health Care and Social Assistance             8545     346.0  ...  12063.0

[23 rows x 6 columns]
```

2 Consider Lead ID: 318465, bank_account_id: 13419 - plot the withdrawals, deposits and daily balance as a daily time series; do the same for Lead ID: 326062, bank_account_id : 14046

```
# 2. Consider Lead ID: 318465, bank_account_id: 13419  - plot the withdrawals, deposits and c

Data_6 = bank_3[(bank_3['Lead ID'] == 318465) & (bank_3.bank_account_id == 13419)]
Data_6a = Data_6[(bank_3['transaction_type'] == 'debit')]
Data_6c = Data_6[(bank_3['transaction_type'] == 'credit')]

Data_6b = pd.pivot_table(Data_6a,columns=['post_date'],values=['amount'],aggfunc=[np.sum])
print(Data_6a, Data_6b)
Data_6.to_csv('/content/drive/MyDrive/Bank_Data/Data_6.csv', index=True)
#pd.pivot_table(bank_3,index=['Lead ID','Industry','bankid'],columns=['transaction_type'],val


#Plot as daily time series  for daily withdrawals.
import matplotlib.pyplot as plt
from pylab import rcParams
rcParams['figure.figsize'] = 100,30

plt.plot(Data_6a['post_date'], Data_6a['amount'],color='red', marker='o')
plt.title('Daily withdrawal Time series', fontsize=8)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Daily withdrawal', fontsize=8)
plt.show()

#Plot as daily time series  for deposit.
rcParams['figure.figsize'] = 100,30
plt.plot(Data_6c['post_date'], Data_6c['amount'],color='red', marker='o')
plt.title('Daily Deposit Time series', fontsize=8)
                                                                    e=8)
```

Saved successfully!

```
#Plot as daily time series  for net balance.

rcParams['figure.figsize'] = 100,30
plt.plot(Data_6a['post_date'], Data_6a['amount'],color='red', marker='o')
plt.title('Daily Time series', fontsize=8)
plt.xlabel('Date', fontsize=8)
plt.ylabel('Daily Balance', fontsize=14)
plt.show()
```

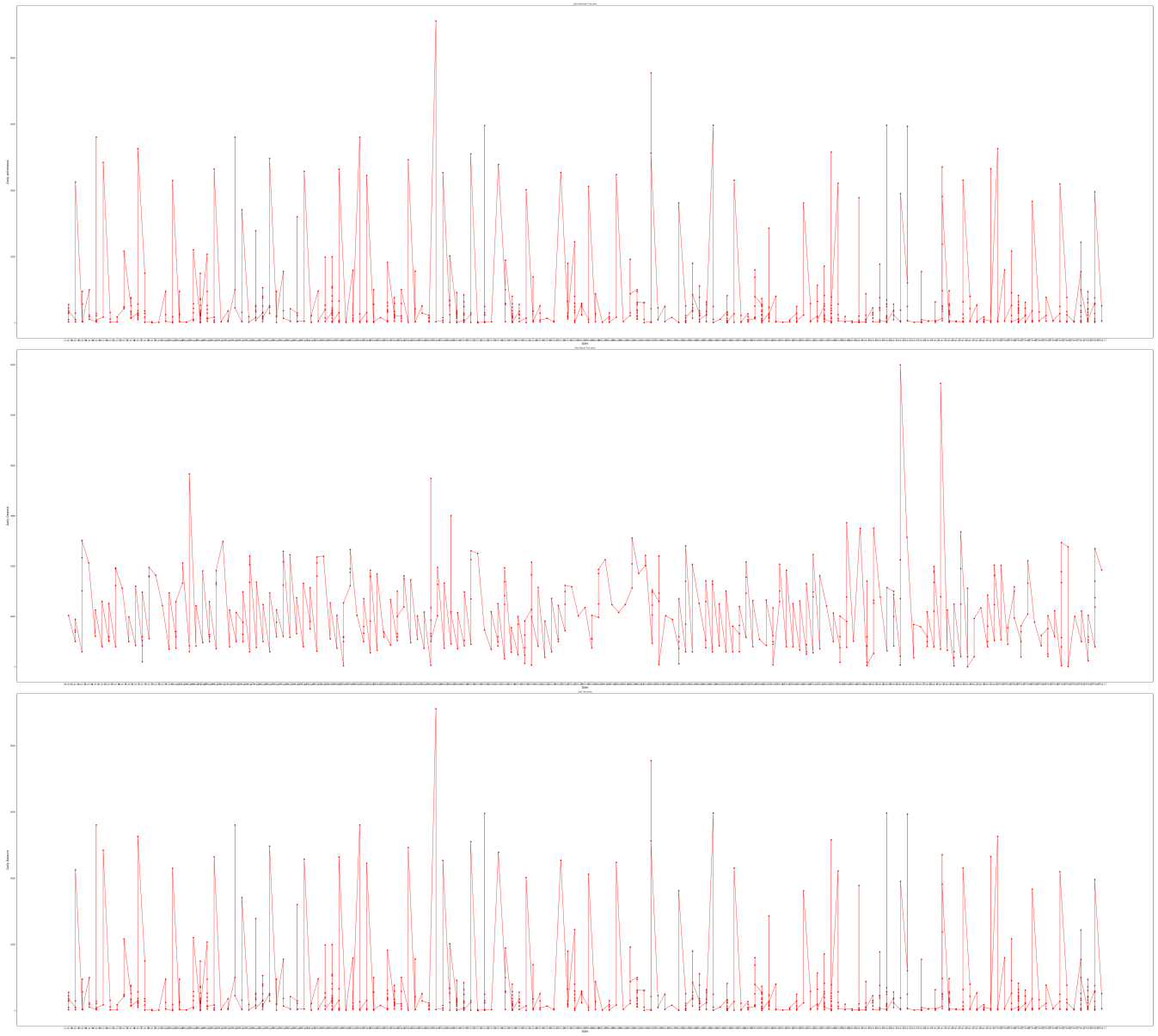
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-b52411c48272> in <module>()
      1 # 2. Consider Lead ID: 318465, bank_account_id: 13419  - plot the withdrawals,
deposits and daily balance as a daily time series; do the same for Lead ID: 326062,
bank_account_id : 14046
      2
----> 3 Data_6 = bank_3[(bank_3['Lead ID'] == 318465) & (bank_3.bank_account_id ==
13419)]
      4 Data_6a = Data_6[(bank_3['transaction_type'] == 'debit')]
      5 Data_6c = Data_6[(bank_3['transaction_type'] == 'credit')]

NameError: name 'bank_3' is not defined
```

```
# 2. Consider Lead ID: 326062, bank_account_id: 14046  - plot the withdrawals, deposits and c

Data_61 = bank_3[(bank_3['Lead ID'] == 326062) & (bank_3.bank_account_id == 14046)]
Data_61a = Data_61[(bank_3['transaction_type'] == 'debit')]
Data_61c = Data_61[(bank_3['transaction_type'] == 'credit')]
Data_61b = pd.pivot_table(Data_6a,columns=['post_date'],values=['amount'],aggfunc=[np.sum])
print(Data_61)
Data_61.to_csv('/content/drive/MyDrive/Bank_Data/Data_61.csv', index=True)
#pd.pivot_table(bank_3,index=['Lead ID','Industry','bankid'],columns=['transaction_type'],val

#Plot as daily time series  for daily withdrawals.
import matplotlib.pyplot as plt
from pylab import rcParams
rcParams['figure.figsize'] = 100,30

plt.plot(Data_61a['post_date'], Data_61a['amount'],color='red', marker='o')
plt.title('Daily withdrawal Time series', fontsize=8)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Daily withdrawal', fontsize=14)
plt.show()

#Plot as daily time series  for deposit.
rcParams['figure.figsize'] = 100,30
plt.plot(Data_61c['post_date'], Data_61c['amount'],color='red', marker='o')
plt.title('Daily Deposit Time series', fontsize=8)
```

Saved successfully!

```
#Plot as daily time series  for net balance.

rcParams['figure.figsize'] = 100,30
plt.plot(Data_61a['post_date'], Data_61a['amount'],color='red', marker='o')
plt.title('Daily Time series', fontsize=8)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Daily Balance', fontsize=14)
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:4: UserWarning: Boolean Se
  after removing the cwd from sys.path.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:5: UserWarning: Boolean Se
  """
       Lead ID  bankid  ...                 Primary_key month_year
25111   326062    8535  ...   326062_8535_14046_42559_1    2016-07
25112   326062    8535  ...  326062_8535_14046_42562_10    2016-07
25113   326062    8535  ...   326062_8535_14046_42562_7    2016-07
25114   326062    8535  ...   326062_8535_14046_42562_9    2016-07
25115   326062    8535  ...   326062_8535_14046_42562_4    2016-07
...        ...     ...  ...                         ...        ...
26374   326062    8535  ...  326062_8535_14046_42788_13    2017-02
26375   326062    8535  ...   326062_8535_14046_42789_1    2017-02
26376   326062    8535  ...   326062_8535_14046_42789_3    2017-02
26377   326062    8535  ...   326062_8535_14046_42789_2    2017-02
26378   326062    8535  ...   326062_8535_14046_42789_4    2017-02

[1268 rows x 13 columns]
```



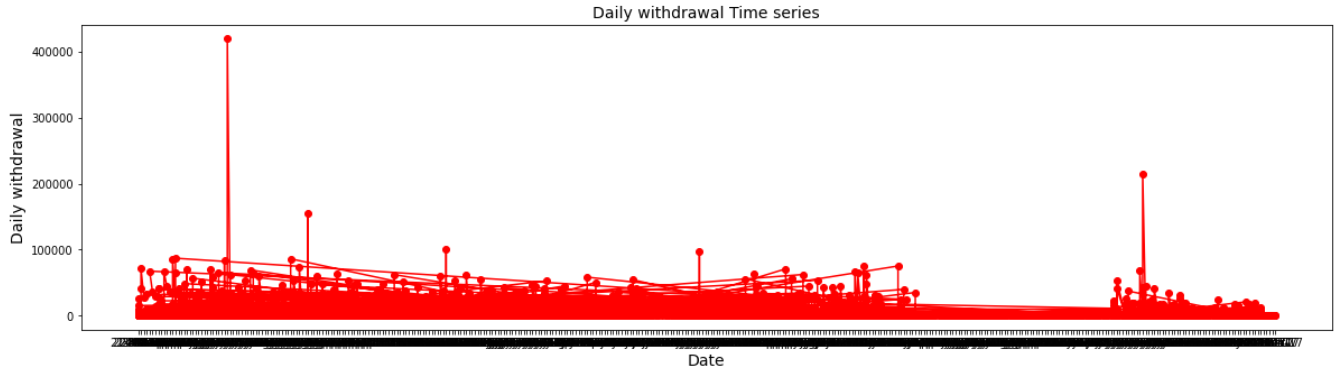
Saved successfully!

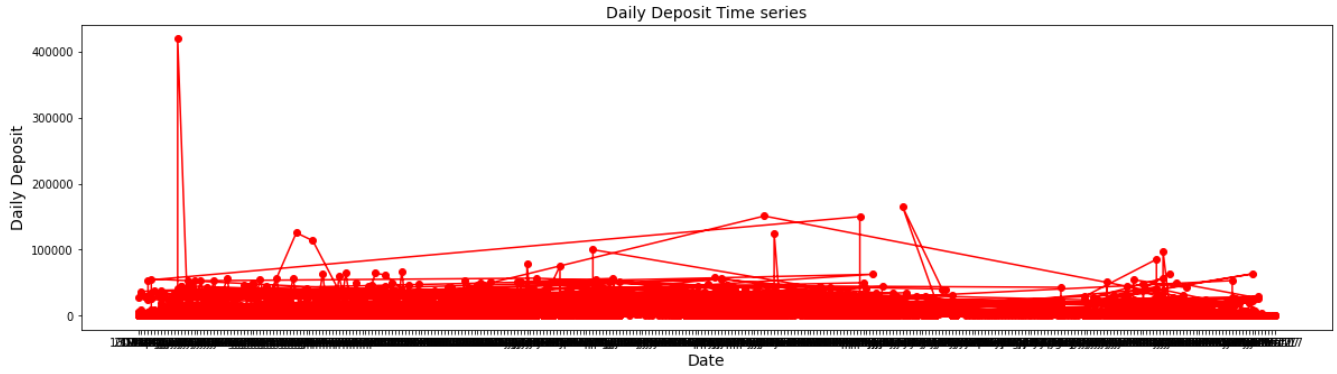
3 For the merchants with the above Lead IDs, plot the withdrawals, deposits and daily balance as a daily time series (aggregate over all their bank accounts)

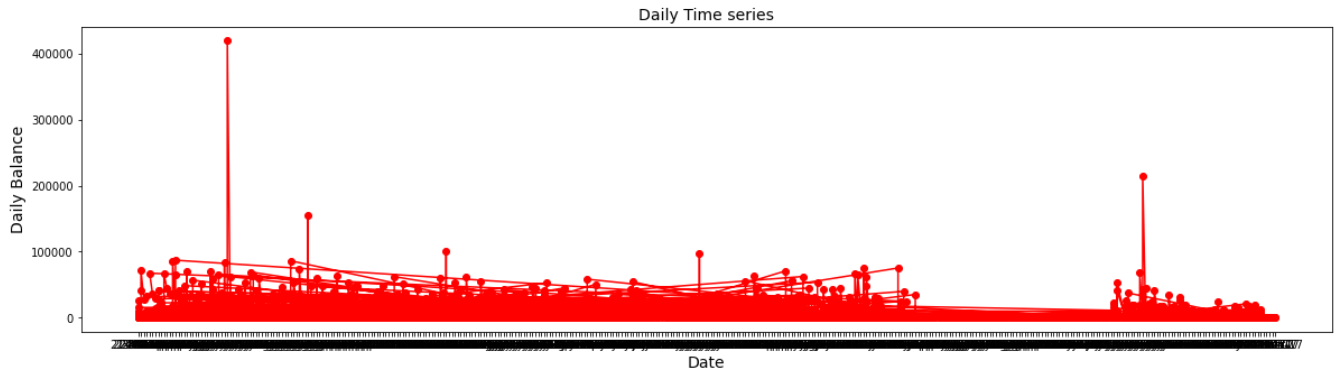
```
Data_7 = bank_3
Data_7a = Data_7[(bank_3['transaction_type'] == 'debit')]
Data_7c = Data_7[(bank_3['transaction_type'] == 'credit')]

Data_7b = pd.pivot_table(Data_7a,columns=['post_date'],values=['amount'],aggfunc=[np.sum])
print(Data_7a, Data_7b)
Data_7a.to_csv('/content/drive/MyDrive/Bank_Data/Data_7a.csv', index=True)
Data_7b.to_csv('/content/drive/MyDrive/Bank_Data/Data_7b.csv', index=True)
Data_7c.to_csv('/content/drive/MyDrive/Bank_Data/Data_7c.csv', index=True)
#pd.pivot_table(bank_3,index=['Lead ID','Industry','bankid'],columns=['transaction_type'],val


#Plot as daily time series  for daily withdrawals.
import matplotlib.pyplot as plt
from pylab import rcParams
rcParams['figure.figsize'] = 20,5

plt.plot(Data_7a['post_date'], Data_7a['amount'],color='red', marker='o')
plt.title('Daily withdrawal Time series', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Daily withdrawal', fontsize=14)
plt.show()

#Plot as daily time series  for deposit.
plt.plot(Data_7c['post_date'], Data_7c['amount'],color='red', marker='o')
plt.title('Daily Deposit Time series', fontsize=14)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Daily Deposit', fontsize=14)
plt.show()

#Plot as daily time series  for net balance.

rcParams['figure.figsize'] = 20,5
plt.plot(Data_7a['post_date'], Data_7a['amount'],color='red', marker='o')
plt.title('Daily Time series', fontsize=14)
plt.xlabel('Date', fontsize=14)
                                                    e=14)
```

Saved successfully!

```
       Lead ID  bankid  ...                Primary_key month_year
3       308148    8535  ...   308148_8535_12460_42450_2    2016-03
4       308148    8535  ...   308148_8535_12460_42451_1    2016-03
5       308148    8535  ...   308148_8535_12460_42452_1    2016-03
6       308148    8535  ...   308148_8535_12460_42453_3    2016-03
7       308148    8535  ...   308148_8535_12460_42453_1    2016-03
...        ...     ...  ...                         ...        ...
29023   330698    8545  ...  330698_8545_14374_42772_14    2017-02
29024   330698    8545  ...   330698_8545_14374_42772_7    2017-02
29026   330698    8545  ...   330698_8545_14374_42773_4    2017-02
29027   330698    8545  ...   330698_8545_14374_42773_3    2017-02
29028   330698    8545  ...   330698_8545_14374_42773_1    2017-02

[18487 rows x 13 columns]                sum                          ...
post_date 01-Apr-16  01-Aug-16 01-Dec-15  ... 31-Mar-16  31-May-16  31-Oct-16
amount     67748.53  173552.05    8366.5  ...   97632.7  210444.38  148666.72

[1 rows x 396 columns]
```







Saved successfully!

4 Top 5 and Bottom 5 Merchants/Banks

```
#Top 5 Merchant IDs by amount
#del bank_3['month name']

#bank_3.to_csv('/content/drive/MyDrive/Bank_Data/Bank_Data_v3.csv', index=False)

bank_4 = bank_3.sort_values('running_balance',ascending = False).groupby('Lead ID').head(2)
print(bank_4.head(5))
bank_4.to_csv('/content/drive/MyDrive/Bank_Data/top5_lead_id.csv', index=False)


#Bottom 5 Merchant IDs by amount
bank_4 = bank_3.sort_values('running_balance',ascending = True).groupby('Lead ID').head(2)
print(bank_4.head(5))
bank_4.to_csv('/content/drive/MyDrive/Bank_Data/Bottom5_lead_id.csv', index=False)

#Top 5 bank IDs by amount
bank_4 = bank_3.sort_values('running_balance',ascending = False).groupby('bankid').head(2)
print(bank_4.head(5))
bank_4.to_csv('/content/drive/MyDrive/Bank_Data/top5_bankid.csv', index=False)

#Bottom 5 bank IDs by amount
bank_4 = bank_3.sort_values('running_balance',ascending = True).groupby('bankid').head(2)
print(bank_4.head(5))
bank_4.to_csv('/content/drive/MyDrive/Bank_Data/bottom5_bankid.csv', index=False)
```

Saved successfully!

```
                                Primary_key month_year
21218_10656_14779_42552_1    2016-07
```