Financial Data Analysis

Context

Analysis on financial data collected from x bank. This analysis would help in finding customer's suspicious activities and would help building models which can prevent frauds in future

Content

Data Contains X bank's Transactions .

Attributes

Txdate: Transaction Date

Txnno: Transaction number

Txtype: Transaction Types(Cash, Transfer,cheques ..)

Cashflowtype: Credit or Debit(CR/DR)

Txamt : Transaction Amount

Acctno: Account Number

Custno: Customer number

Custtype: Type of the customer, divided into two parts Individual or Corporates Acc_open_date : Date on which account has opened with the respective bank.

Additional Information

There are approx 161026 records.transactions

In [374]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.offline as pyo
import plotly.graph_objs as go
import seaborn as sns
pyo.init_notebook_mode()
```

In [389]:

```
cols =  ['txdate','txnno','txtype','txamt','cashflowtype','acctno','custno','custtype','acc_open_date']
df = pd.read_csv(r"D:\amol_lap_data\python_tutor\Jupyter\Datasets\aml\raw_txns5.csv", names = cols, encoding= 'lati
n1' )
```

Sample snapshot of Data

In [390]:

```
df.head()
```

Out[390]:

	txdate	txnno	txtype	txamt	cashflowtype	acctno	custno	custtype	acc_open_date
0	ï»¿2019-10-01 00:00:00.000	500ZTRF19274007S_204414906	IJARA REPAYMENT	0.052	DR	508301266981020	30126698	Individual	2015-02-10 00:00:00.000
1	2019-10-01 00:00:00.000	500ZTRF1927400J8_204415915	TAWARRUQ REPAYMENT	50.945	DR	514301940261020	30194026	Individual	2017-08-17 00:00:00.000
2	2019-10-01 00:00:00.000	500ZTRF1927400JW_204415983	TAWARRUQ REPAYMENT	201.175	DR	501300980261015	30098026	Individual	2013-12-15 00:00:00.000
3	2019-10-01 00:00:00.000	500ZTRF1927400Z0_204417190	TAWARRUQ REPAYMENT	0.003	DR	501028124791011	2812479	Individual	2006-05-22 00:00:00.000
4	2019-10-01 00:00:00.000	500ZTRF1927401GE_204418650	TAWARRUQ REPAYMENT	0.008	DR	510300390601011	30039060	Individual	2010-12-21 00:00:00.000

Data preprocessing

In [391]:
```
# Changing the garbage value (txdate = 'ï»¿2019-10-01 00:00:00.000')
df.at[0,'txdate'] = '2019-10-01 00:00:00.000'

# Converting to DateTime formats
df['txdate'] = pd.to_datetime(df['txdate'])
df['acc_open_date'] = pd.to_datetime(df['acc_open_date'])

df['custno'] = df['custno'].astype(str)
df['acctno'] = df['acctno'].astype(str)
```

In [295]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 161026 entries, 0 to 161025
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   txdate         161026 non-null  datetime64[ns]
 1   txnno          161026 non-null  object
 2   txtype         161026 non-null  object
 3   txamt          161026 non-null  float64
 4   cashflowtype   161026 non-null  object
 5   acctno         161026 non-null  object
 6   custno         161026 non-null  object
 7   custtype       161026 non-null  object
 8   acc_open_date  161026 non-null  datetime64[ns]
dtypes: datetime64[ns](2), float64(1), object(6)
memory usage: 11.1+ MB
```

In [239]:
```
df.describe(include = 'all').T
```

Out[239]:

	count	unique	top	freq	first	last	mean	std	min	25%	5
txdate	161026	122	2019-12-31 00:00:00	6258	2019-10-01	2020-01-31	NaN	NaN	NaN	NaN	I
txnno	161026	161026	501GCCP19361000U_242729445	1	NaT	NaT	NaN	NaN	NaN	NaN	I
txtype	161026	75	POS PURCHASE	43387	NaT	NaT	NaN	NaN	NaN	NaN	I
txamt	161026	NaN	NaN	NaN	NaT	NaT	249.334	5115.41	0.001	5	16
cashflowtype	161026	2	DR	106733	NaT	NaT	NaN	NaN	NaN	NaN	I
acctno	161026	NaN	NaN	NaN	NaT	NaT	5.05787e+14	5.60113e+12	5e+14	5.013e+14	5.02301e
custno	161026	NaN	NaN	NaN	NaT	NaT	2.28324e+07	1.24351e+07	1689	2.81544e+06	3.00601e
custtype	161026	2	Individual	129563	NaT	NaT	NaN	NaN	NaN	NaN	I
acc_open_date	161026	2074	2019-03-16 00:00:00	13116	1993-10-17	2020-01-22	NaN	NaN	NaN	NaN	I

There are 1552394 unique transactions from done 7289 Accounts.

Duration of these transaction ranging between 01st Oct 2019 to 31-Jan-2019 (~4 months).

Univariate Analysis

We will check the transction volume day wise

In [296]:
```
df[['txdate']].describe().T
```

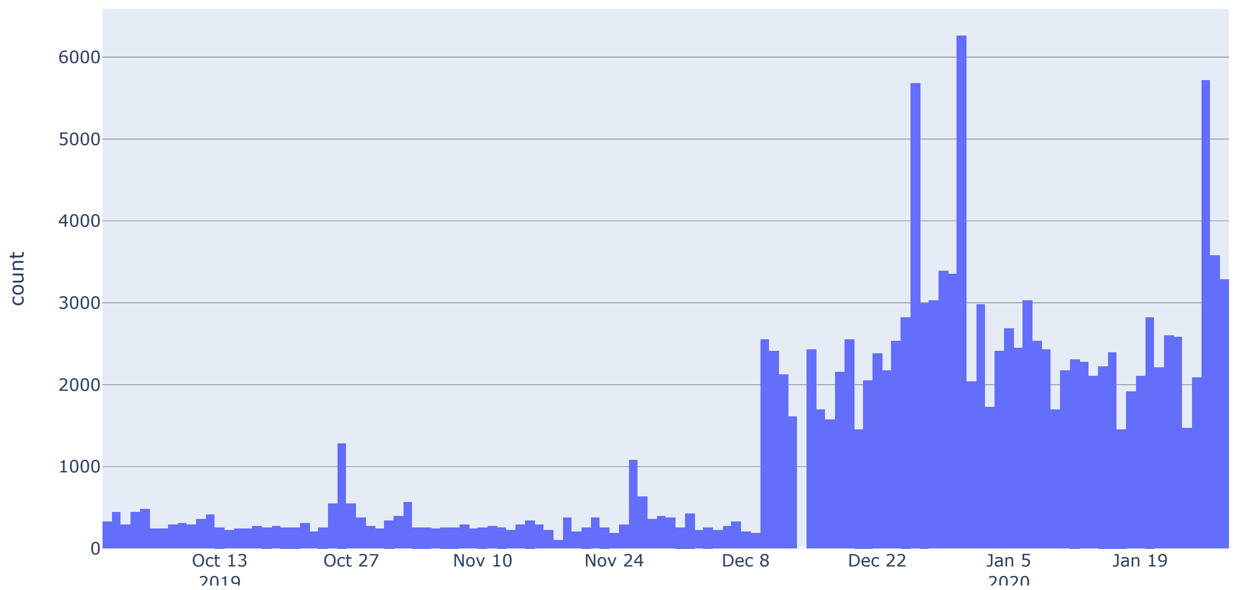
Out[296]:

	count	unique	top	freq	first	last
txdate	161026	122	2019-12-31	6258	2019-10-01	2020-01-31

Transaction Dates

In [297]:

```
fig = px.histogram(df,x = 'txdate')
fig.show()
```



The above graph shows, there is an exponential rise in transactions done by bank customer from date 10th Dec 2019. This can be due to below reasons:

1. No. of new customers onboarded (If Yes, then there must be an news where in bank had offered attractive services)
2. Open offers in market in which customers are buying new things

Transaction Types

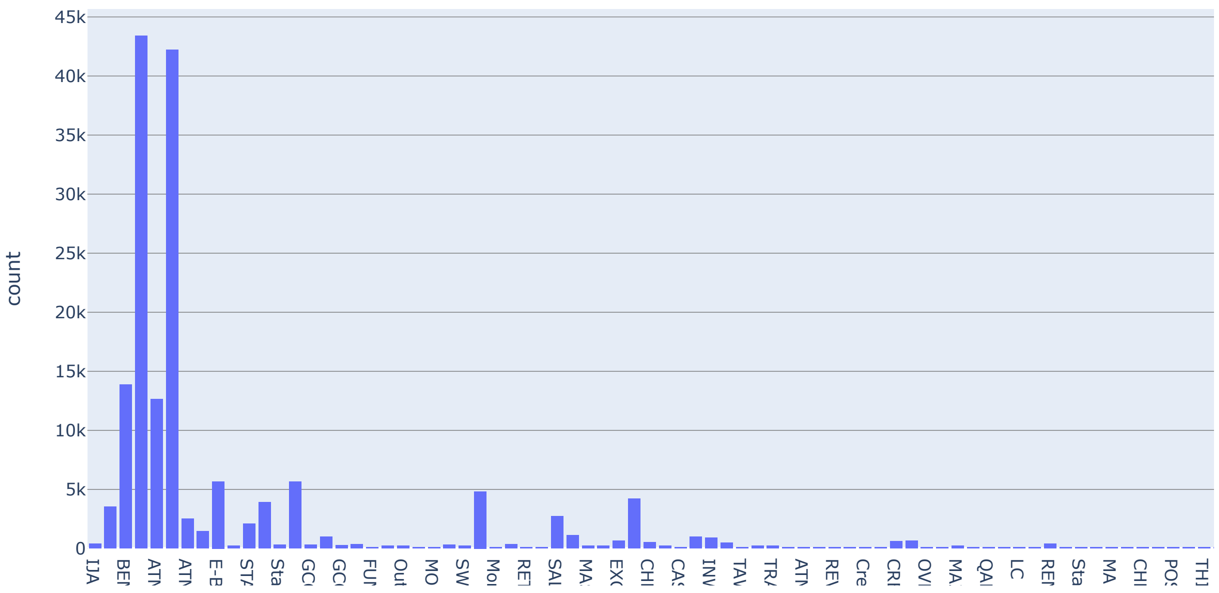
In [298]:

```
df[['txtype']].describe().T
```

Out[298]:

	count	unique	top	freq
txtype	161026	75	POS PURCHASE	43387

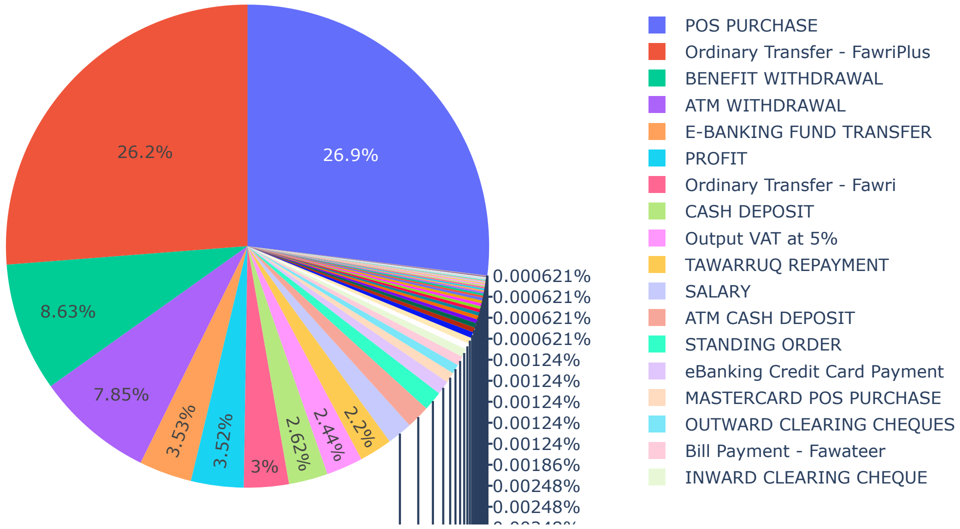
```
In [299]: fig = px.histogram(df,x = 'txtype')
          fig.show()
```



```
In [300]: labels = df['txtype'].value_counts().index
          values = df['txtype'].value_counts().values

          trace = go.Pie(labels=labels, values=values)

          pyo.iplot([trace])
```



This shows that majority of the trasactions are occuring in ""Pos Purchase" and "Orinary Transfer Fawri Plus"

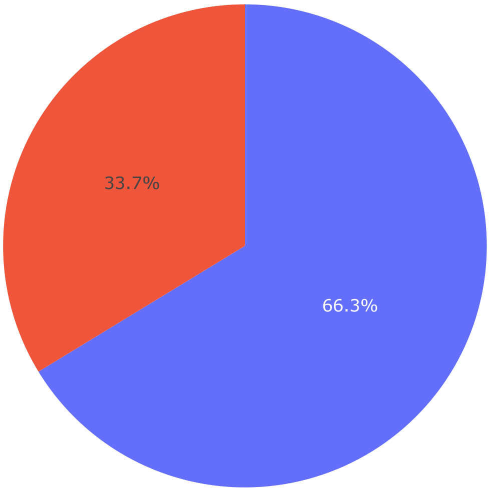
POS Purchase: Purchases done at merchants i.e shops, malls, restaurants etc .. Fawri Plus: It is Fund transfer service works similarly as NEFT in India

Cashflowtype - Credit/Debit

```
In [301]: labels = df['cashflowtype'].value_counts().index
          values = df['cashflowtype'].value_counts().values

          trace = go.Pie(labels=labels, values=values)

          pyo.iplot([trace])
```



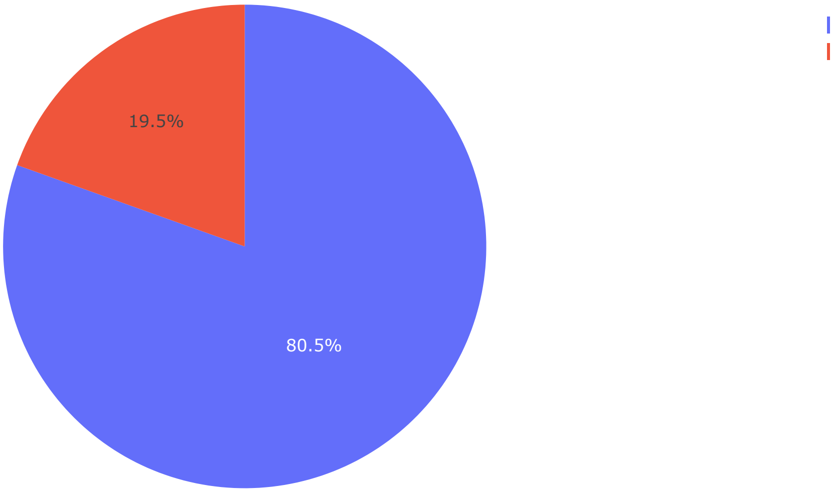
This states that debit transactions are high in volume around 66%. This also indicates that customers likes to spend money

Customer Types- Individuals/Corporates

```
In [302]: labels = df['custtype'].value_counts().index
          values = df['custtype'].value_counts().values

          trace = go.Pie(labels=labels, values=values)

          pyo.iplot([trace])
```

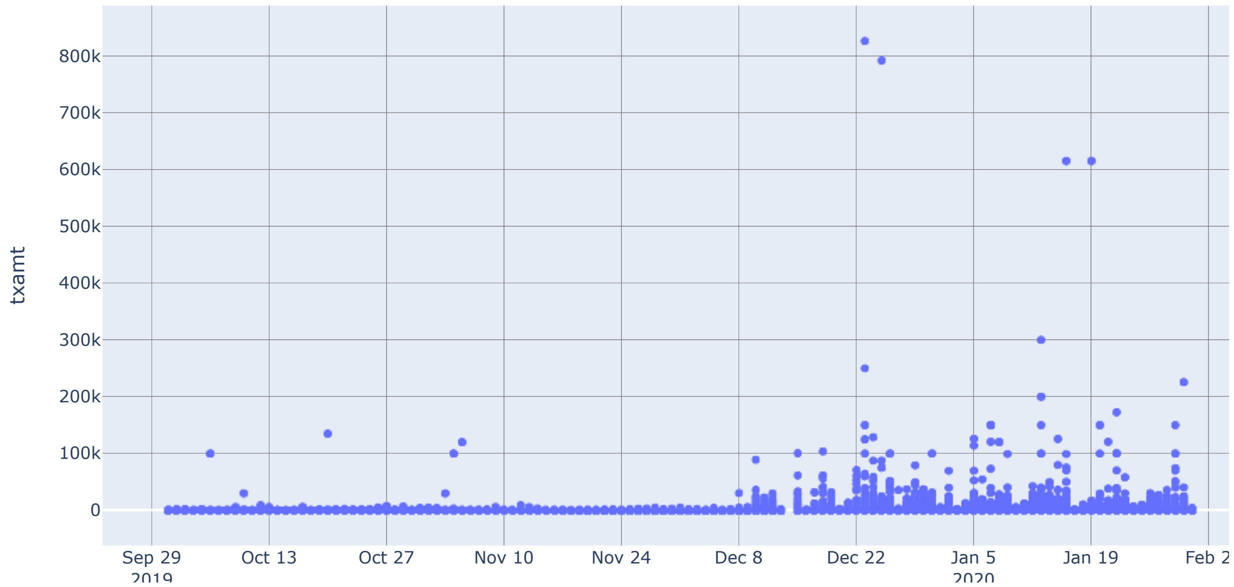


80.5% of the customers are Individuals and Others are Corporates

Bivariate Analysis

Customer and Transaction Dates

```
In [304]: fig = px.scatter(df,x='txdate', y='txamt', hover_data = ['custno','txnno','txamt','cashflowtype','txtype'])
          fig.show()
```



From the above chart we can observe that on dates December 23,25 and January 16,19. High Amount of transactions has been carried by the customers. Below are the transaction details

23rd Dec - Customer with ID **30012194** had been **deposited** with **Swift Transfer** of Amount **~826023** 25th Dec - Customer with ID **30012194** had been **withdrawn** with **Swift Transfer** of Amount **~791700** 16th Jan - Customer with ID **30012194** had been **withdrawn** with **Modaraba Maturity** of Amount **~614975** 23rd Jan - Customer with ID **30012194** had been **deposited** with **Modaraba Investment** of Amount **~614975**

Observations:

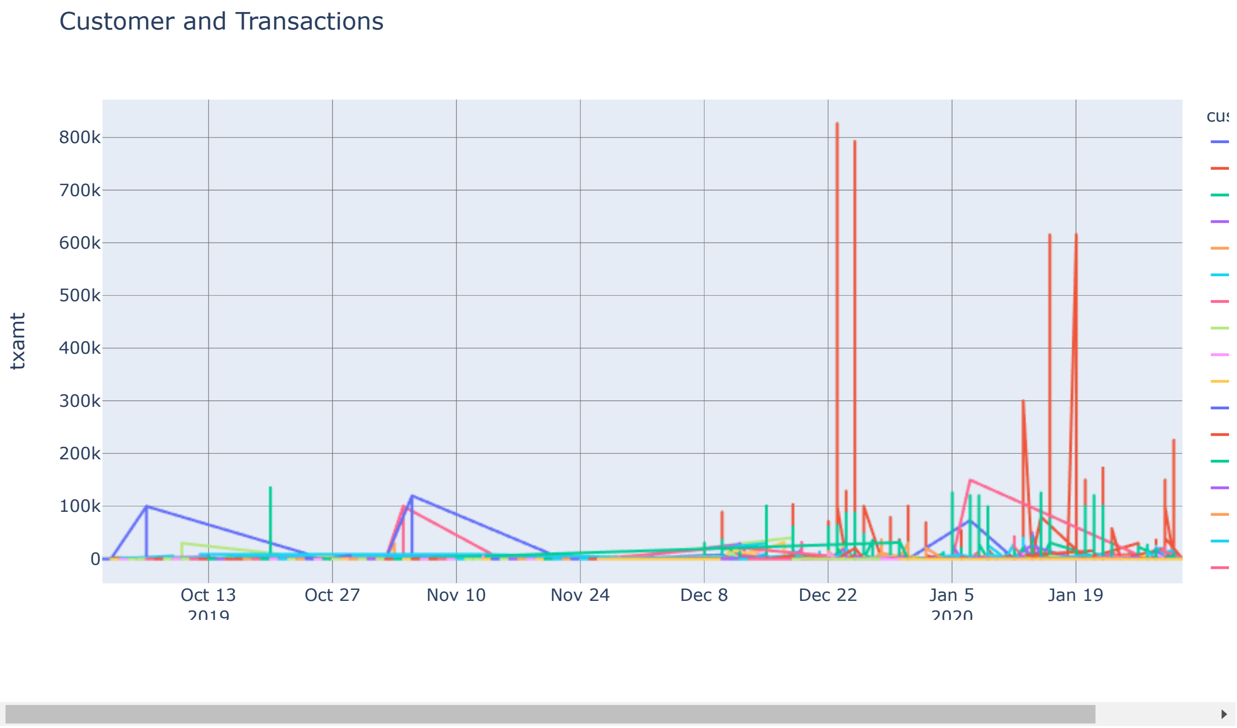
1. This means in the month of Dec 2019, Customer 30012194 had done the international transfers credit and debit within two day. Next month, he invested an amount through **Modaraba**
2. Credit Transactions followed by Withdrawals within a week
3. **Rapid Movement of Funds** can be seen for Customer 30012194
4. Similar activities of the same customer can be seen in the month of Dec and Jan

Modaraba is a special kind of partnership where one partner gives money to another for investing it in a commercial enterprise

Customer and Transaction amounts with respect to Transaction Dates

In [305]:

```
fig = px.line(df,x='txdate', y='txamt', color='custno', line_group = 'custno', title='Customer and Transactions')
fig.show()
```



This Graph shows that few customers are having number of transactions after Dec 10

Top Five Customers according to number of transactions

In [306]:

```
top5 = df.groupby('custno')['custno'].count().to_frame().rename(columns = {'custno':'txn_count'}).sort_values(by='t
xn_count',ascending=False).reset_index().head(5)
```

In [307]:

```
top5
```

Out[307]:

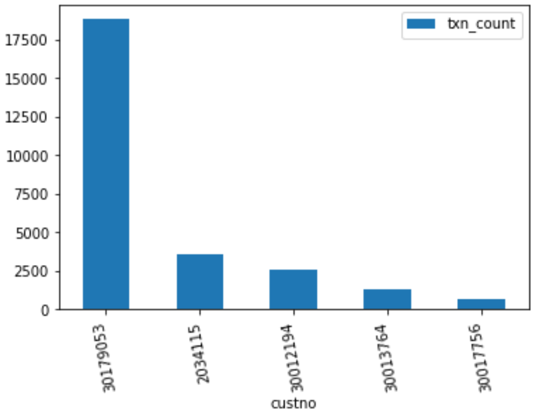
	custno	txn_count
0	30179053	18855
1	2034115	3607
2	30012194	2606
3	30013764	1323
4	30017756	685

In [317]:

```
top5.plot.bar(x = 'custno',y='txn_count',rot=100)
```

Out[317]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x24955eb4688>
```



Showing top 5 customer and there transaction count in 4 months.

```
In [371]: a = df.groupby(['custno','cashflowtype'])['txamt'].count().to_frame().rename(columns = {'txamt':'txn_count'})
          b = a.sort_values(by = 'txn_count',ascending=False).reset_index()
          c = pd.merge(left=top5, right=b, how='left', left_on='custno', right_on='custno')
          c['custno'] = c['custno'].astype(str)
          c = c.drop(columns='txn_count_x')
```

```
In [372]: c
```

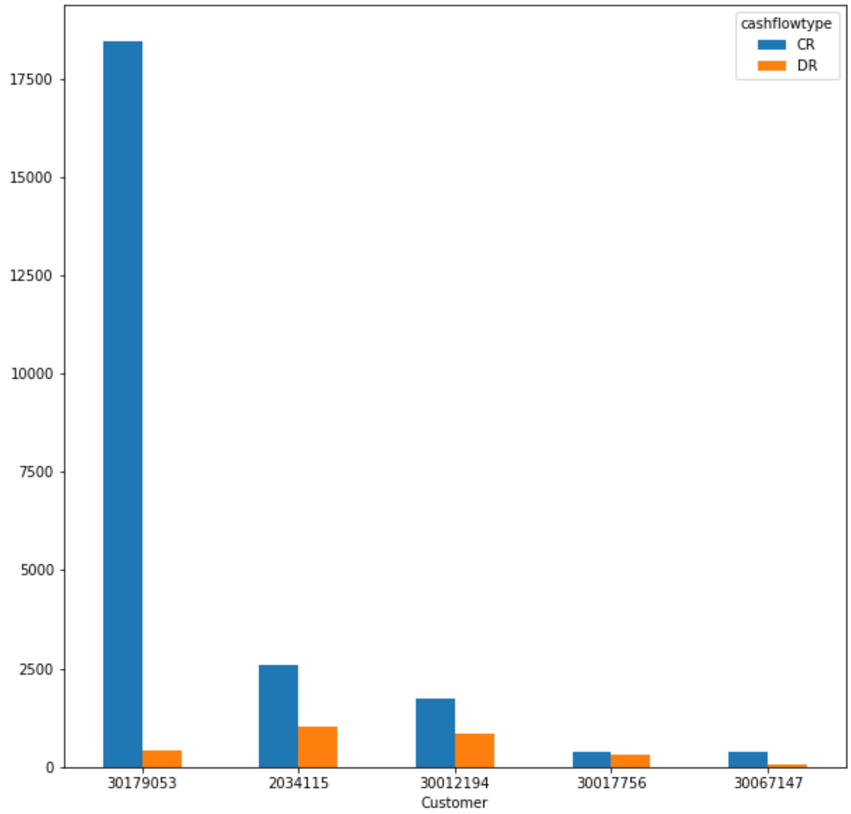
Out[372]:

	custno	cashflowtype	txn_count_y
0	30179053	CR	18444
1	30179053	DR	411
2	2034115	CR	2588
3	2034115	DR	1019
4	30012194	CR	1747
5	30012194	DR	859
6	30013764	DR	1270
7	30013764	CR	53
8	30017756	CR	384
9	30017756	DR	301

Top 5 maximum no. of Transactions - Credit/Debit done by customer

```
In [387]: ctab_cust = pd.crosstab(df['custno'] , df['cashflowtype']).sort_values(by=['CR','DR'],ascending=[False,False]).head
          ()
```

```
In [388]: ctab_cust.plot(kind = 'bar' , figsize = (10,10) , rot = 0).set(xlabel = 'Customer')
          plt.show()
```



Above graph shows detail of 5 diffrent customers whose credit/debit transactions are highest.

Customers 30179053, 2034115 and 30012194 are more active incase of fund transfers and other transactions.