SandPit FinTech Intern Demo

Tasks

- A. Source Data
- B. Exploratory Analysis
- C. Manipulating the Dataset

Questions

- a.) What (if any) experience do you have in Cloud Services (eg. AWS, GCP)?
- b.) What is your experience with building machine learning models? What packages are you familiar with?
- c.) Duration of Tasks all together?

Task A. Source Data

Source:

1. "Online payments" downloaded from DataWorld Website
2. "1999 Czech Financial Dataset - Real Anonymized Transactions" dowmloaded from DataWorld Website
3. "Bank Transaction Data" downloaded from Kaggle Website

In [19]:
```
! pip install datatable
```

```
Collecting datatable
  Downloading https://files.pythonhosted.org/packages/26/fc/d780b150eeae26e0a14b
964f27eefcf9fad746f16e0b05c40098d84913d7/datatable-0.11.1-cp37-cp37m-manylinux20
10_x86_64.whl (83.9MB)
     |████████████████████████████████| 83.9MB 77kB/s
Installing collected packages: datatable
Successfully installed datatable-0.11.1
```

In [20]:
```
# importing dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import datatable as dt
```

Source 1.

In [2]:
```
df1 = pd.read_csv('/content/2019_20_1.csv')
df1.head(5)
```

Out[2]:

	category	month	value_of_internet_payments_made	number_of_internet_payments_made	value_
0	Leeds Card	Apr-19	NaN	NaN	
1	Leodis	Apr-19	279.08	15.0	

	category	month	value_of_internet_payments_made	number_of_internet_payments_made	value_
2	Highways	Apr-19	NaN	NaN	
3	City & Environmental Services	Apr-19	25560.60	1111.0	
4	School Meals	Apr-19	17529.85	474.0	

Source 2.

In [21]:
```
card = pd.read_csv('/content/card.csv')
disp = pd.read_csv('/content/disp.csv')
trans = pd.read_csv('/content/trans.csv')
print(card.head(5), disp.head(5), trans.head(5))
```

```
   card_id  disp_id     type           issued
0     1005     9285  classic  931107 00:00:00
1      104      588  classic  940119 00:00:00
2      747     4915  classic  940205 00:00:00
3       70      439  classic  940208 00:00:00
4      577     3687  classic  940215 00:00:00    disp_id  client_id  account_id
type
0        1          1          1      OWNER
1        2          2          2      OWNER
2        3          3          2  DISPONENT
3        4          4          3      OWNER
4        5          5          3  DISPONENT    trans_id  account_id    date
type  ... balance  k_symbol  bank account
0    695247        2378  930101  PRIJEM  ...    700.0       NaN   NaN     NaN
1    171812         576  930101  PRIJEM  ...    900.0       NaN   NaN     NaN
2    207264         704  930101  PRIJEM  ...   1000.0       NaN   NaN     NaN
3   1117247        3818  930101  PRIJEM  ...    600.0       NaN   NaN     NaN
4    579373        1972  930102  PRIJEM  ...    400.0       NaN   NaN     NaN

[5 rows x 10 columns]
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2718: Dt
ypeWarning: Columns (8) have mixed types.Specify dtype option on import or set l
ow_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [24]:
```
disp['ytype'] = disp.type
disp = disp.drop(axis=1, columns='type')
disp.head()
```

Out[24]:

	disp_id	client_id	account_id	ytype
0	1	1	1	OWNER
1	2	2	2	OWNER
2	3	3	2	DISPONENT
3	4	4	3	OWNER
4	5	5	3	DISPONENT

In [25]:
```
card_disp = card.merge(disp, how='inner')
```

```
card_disp.head()
```

Out[25]:

	card_id	disp_id	type	issued	client_id	account_id	ytype
0	1005	9285	classic	931107 00:00:00	9593	7753	OWNER
1	104	588	classic	940119 00:00:00	588	489	OWNER
2	747	4915	classic	940205 00:00:00	4915	4078	OWNER
3	70	439	classic	940208 00:00:00	439	361	OWNER
4	577	3687	classic	940215 00:00:00	3687	3050	OWNER

In [31]:

```
df2 = trans.merge(card_disp, on='account_id', how='outer')
df2.tail(5)
```

Out[31]:

	trans_id	account_id	date	type_x	operation	amount	balance	k_symbol	bank	acc
1056315	462598	1573	981209	PRIJEM	PREVOD Z UCTU	5492.0	19938.9	DUCHOD	EF	562849
1056316	462622	1573	981211	VYDAJ	PREVOD NA UCET	3638.0	16300.9	SIPO	IJ	192402
1056317	462683	1573	981224	VYDAJ	VYBER	2200.0	14100.9	NaN	NaN	
1056318	462628	1573	981227	VYDAJ	VYBER	1200.0	12900.9	NaN	NaN	
1056319	3582063	1573	981231	PRIJEM	NaN	62.2	12963.1	UROK	NaN	

Source 3

In [67]:

```
df3 = pd.read_excel('/content/bank.xlsx')
df3.head(5)
```

Out[67]:

	Account No	DATE	TRANSACTION DETAILS	CHQ.NO.	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT	.
0	409000611074'	2017-06-29	TRF FROM Indiaforensic SERVICES	NaN	2017-06-29	NaN	1000000.0	1000000.0	.
1	409000611074'	2017-07-05	TRF FROM Indiaforensic SERVICES	NaN	2017-07-05	NaN	1000000.0	2000000.0	.
2	409000611074'	2017-07-18	FDRL/INTERNAL FUND TRANSFE	NaN	2017-07-18	NaN	500000.0	2500000.0	.
3	409000611074'	2017-08-01	TRF FRM Indiaforensic SERVICES	NaN	2017-08-01	NaN	3000000.0	5500000.0	.
4	409000611074'	2017-08-16	FDRL/INTERNAL FUND TRANSFE	NaN	2017-08-16	NaN	500000.0	6000000.0	.

Task B. Exploratory Analysis on Source 3

In [68]:

```
# a run down of the statistics in the dataset
df3.describe()
```

Out[68]:

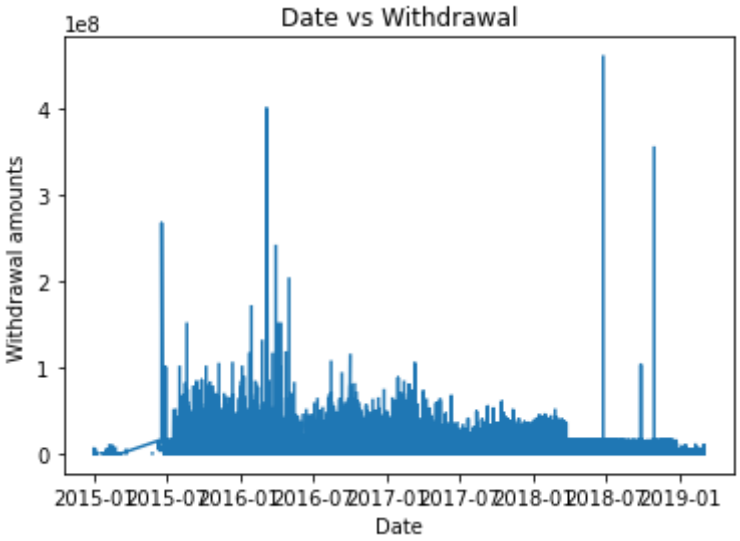
	CHQ.NO.	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT
count	905.000000	5.354900e+04	6.265200e+04	1.162010e+05
mean	791614.503867	4.489190e+06	3.806586e+06	-1.404852e+09
std	151205.932910	1.084850e+07	8.683093e+06	5.348202e+08
min	1.000000	1.000000e-02	1.000000e-02	-2.045201e+09
25%	704231.000000	3.000000e+03	9.900000e+04	-1.690383e+09
50%	873812.000000	4.708300e+04	4.265000e+05	-1.661395e+09
75%	874167.000000	5.000000e+06	4.746411e+06	-1.236888e+09
max	874525.000000	4.594475e+08	5.448000e+08	8.500000e+06

Cannot tell too much yet only that account usage has been very inconsistent due to high standard deviations

In [70]:

```
# Understanding withdrawal nature of clients?
plt.plot(df3.DATE, df3['WITHDRAWAL AMT'])
plt.title('Date vs Withdrawal')
plt.xlabel('Date')
plt.ylabel('Withdrawal amounts')
plt.show()
```

Out[70]:

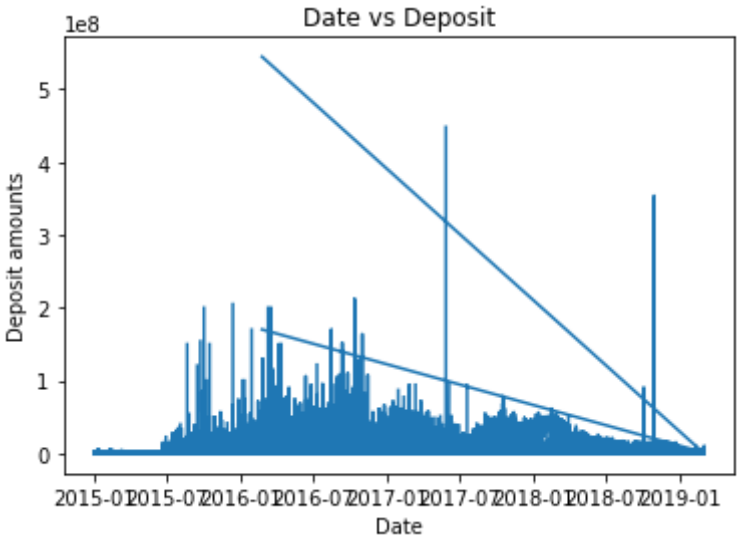


This seems to indicate that withdrawals were more seasonal, ciclical and frquent in the passed

In [71]:

```
# To understand deposit frequency, excusing the artifacfs?
plt.plot(df3.DATE, df3['DEPOSIT AMT'])
plt.title('Date vs Deposit')
plt.xlabel('Date')
plt.ylabel('Deposit amounts')
plt.show()
```

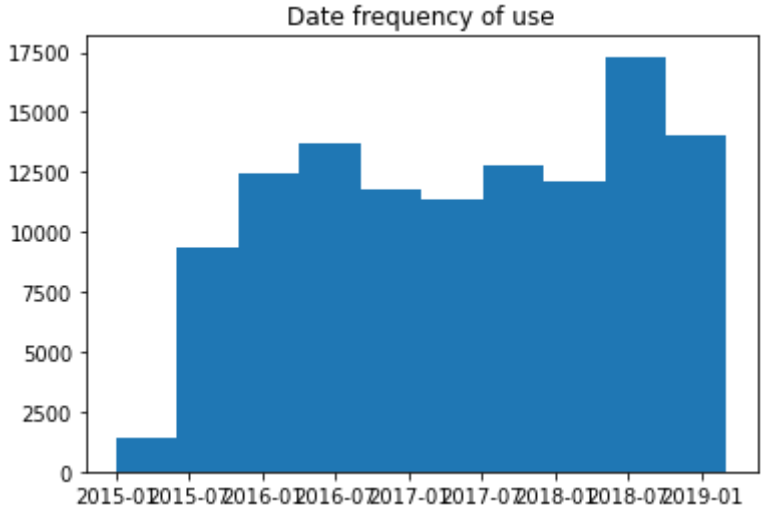
Out[71]:



The trend with the depoists follow that of the withdrawal being more frequent in the passed

In [73]:
```
# To understand use frequency of accounts?
plt.hist(df3.DATE)
plt.title('Date frequency of use')
plt.show()
```

Out[73]:



This show a slight increase of account use over time meaning that taking previous charts into consideration of course. That accounts usage have grown more consistent over the years.

In [83]:
```
# How most tranactions were made?
pd.Series(df3['TRANSACTION DETAILS']).value_counts()
```

Out[83]:
```
FDRL/INTERNAL FUND TRANSFE          8839
FDRL/NATIONAL ELECTRONIC F          6262
INTERNAL FUND TRANSFER IN           3262
TRF TO  Indiaforensic SERVICES I    3082
TRF FROM  Indiaforensic SERVICES    2107
                                    ...
IRTT00701515308                        1
AEPS GST INC DT 10-13-201              1
CHQ DEP/333375/OWDEL1/SBI              1
IMPS 819721825139 FROM OX              1
```

```
RTGS/CITIH17181703870/PAY              1
Name: TRANSACTION DETAILS, Length: 44806, dtype: int64
```

Now we also know that the most transfers were made by FDRL/INTERNAL FUND TRANSFER

Task C. Manipulating the Dataset

In [63]:
```
class Df():
  def __init__(self, path, column):
    self.path = path
    self.column = column
  def stdDev(self):
    print(np.std(dt.fread(self.path)[self.column].to_numpy()))
  def mean(self):
    print(np.mean(dt.fread(self.path)[self.column].to_numpy()))
  def outliers(self):
    print(np.abs(stats.zscore(dt.fread(self.path)[self.column].to_numpy())))



p1 = Df('/content/bank.xlsx','DEPOSIT AMT')
```

In [66]:
```
print(p1.stdDev())
print(p1.mean())
print(p1.outliers())
```

```
8683024.111405512
None
3806585.8284412315
None
[[0.32322676897265146]
 [0.32322676897265146]
 [0.3808103934777624]
 [0.09289227095220749]
 [0.3808103934777624]
 [0.3808103934777624]
 [0.3808103934777624]
 [0.3808103934777624]
 [0.3808103934777624]
 [0.3808103934777624]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
 [--]
```

```
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
[0.4038438432798068]
[--]
[0.20805951996242947]
[--]
[--]
[--]
[--]
[0.32322676897265146]
[0.32322676897265146]
[0.32322676897265146]
[--]
[--]
[--]
[--]
[--]
[--]
[0.32322676897265146]
[0.4038438432798068]
[0.30019331917060704]
[--]
[--]
[--]
[--]
[--]
[0.32322676897265146]
[0.32322676897265146]
[0.32322676897265146]
[--]
[--]
[--]
[--]
[--]
[--]
[--]
```

```
  [--]
  [--]
  [--]
  [0.4038438432798068]
  [0.4038438432798068]
  [--]
  [--]]
 None
```

Questions

a.) I have some very limited experience with AWS, but now more so with Azure. I am working on getting an Azure Data Scientist Associates Certification.

b.) I have built alot of scikit-learn model in the passed but have branch out a little and done some CNN, RNN, and GAN with tensorflow.

c.) This took me 7 hours give or take. I am not the greatest software engineer had problems with Task C.