BUSINESS REPORT FOR STATISTICS:

PART 1(PROBABILITY)

Investment Advisors agree that near retirees, defined as people aged 55 to 65, should have balanced portfolios. Most advisors suggest that the near –retirees have no more than 50% of their investments in stocks.

Suppose you have a random sample of 10 people who have been labelled as near retirees in 2008. Let X (near retirees) be a random variable which follows Binomial Distribution with the parameters n = 10 and p = 0.22.

TASK/ANSWER:

1. What is the probability that during 2008 zero near retirees had 90% or more of their investments in stocks?

```
binomial[0]

Result: 0.08335775831236203
```

2.What is the probability that during 2008 exactly one near retiree had 90% or more of his investments in stocks?

```
binomial[1]

Result: 0.23511162600922625o
```

3.is the probability that during 2008 two or fewer near retirees had 90% or more of their investment in stocks?

```
binomial[0] + binomial[1] + binomial[2]

Result: 0.6168802942563751 or 61.688%
```

4. What is the probability that during 2008 three or more near retirees had 90% or more of their investment in stocks?

```
(1-(binomial[0] + binomial[1] + binomial[2]))

Result: 0.38311970574362486 or 38.311
```

PART -II - Hypothesis Testing

Data Understanding

Using basic Python functions, we want to analyse the pattern of usage of credit card and debit card over the years. We will also see the trend of digital payment by various banks over the years. We will analyze a few patterns like how many ATMs and PoS Machines have been installed over the years. Which bank has topped in terms of deploying ATMs.

TASK/ANSWER

1. Read the data set and check shape and info and get familiar with the data.

```
data = pd.read_csv('payment_data.csv')
data.head()

(Table in ipynb. File)
```

We get the entire table that we needed to import.

Now we have to find all the info and shape of this:

```
data.shape
```

```
(5592, 21)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5592 entries, 0 to 5591
Data columns (total 21 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   month                               5592 non-null   object
 1   year                                5592 non-null   int64
 2   month_number                        5592 non-null   int64
 3   start_date                          5592 non-null   object
 4   end_date                            5592 non-null   object
 5   bank_name                           5592 non-null   object
 6   no_atms_on_site                     5592 non-null   int64
 7   no_atms_off_site                    5592 non-null   int64
 8   no_pos_on_line                      5592 non-null   int64
 9   no_pos_off_line                     5592 non-null   float64
 10  no_credit_cards                     5592 non-null   float64
 11  no_credit_card_atm_txn              5592 non-null   int64
 12  no_credit_card_pos_txn              5592 non-null   int64
 13  no_credit_card_atm_txn_value_in_mn  5592 non-null   float64
 14  no_credit_card_pos_txn_value_in_mn  5592 non-null   float64
 15  no_debit_cards                      5592 non-null   float64
 16  no_debit_card_atm_txn               5592 non-null   float64
 17  no_debit_card_pos_txn               5592 non-null   float64
 18  no_debit_card_atm_txn_value_in_mn   5592 non-null   float64
 19  no_debit_card_pos_txn_value_in_mn   5592 non-null   float64
 20  debit_trans                         5592 non-null   float64
dtypes: float64(10), int64(7), object(4)
memory usage: 917.6+ KB
```

From the above table and info we can see that it we get information about every feature on the list. From the shape we get the number of rows and columns in the table. Also we get the count of every categorical and numerical values so we get to know if there are any missing values or the actual count.

2. Check the summary statistics of the dataframe and comment on your findings.
   data.describe().T

   Summary Statistics below gives us statistics for numeric variables for an entire data set or a subset of observations in the data set. There significant changes between the years is what we can observe. We can see that the maximum value is from debit_card_transaction. The number of credit cards bought is less than that of debit cards. This shows that generally people prefer using debit cards. So this table has all the data starting from 2011.The highest mean is that of Total number of financial transactions done by the debit card issued by the bank at POS terminals. Again showing highest number people making debit card transactions

```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
year	5592.0	2.015076e+03	2.491967e+00	2011.0	2013.000000	2.015000e+03	2.017000e+03	2019.0
month_number	5592.0	6.407904e+00	3.365489e+00	1.0	4.000000	6.000000e+00	9.000000e+00	12.0
no_atms_on_site	5592.0	1.544657e+03	3.068095e+03	0.0	168.000000	6.915000e+02	1.686000e+03	31749.0
no_atms_off_site	5592.0	1.410715e+03	3.663979e+03	0.0	109.000000	4.010000e+02	8.270000e+02	33209.0
no_pos_on_line	5592.0	3.048513e+04	9.241177e+04	0.0	0.000000	1.769000e+03	1.090800e+04	988458.0
no_pos_off_line	5592.0	1.152598e+02	8.291416e+02	0.0	0.000000	0.000000e+00	0.000000e+00	13945.0
no_credit_cards	5592.0	4.682019e+05	1.390633e+06	0.0	0.000000	5.496000e+03	1.339610e+05	13138330.0
no_credit_card_atm_txn	5592.0	8.157823e+03	2.378192e+04	0.0	0.000000	1.800000e+01	3.942750e+03	202780.0
no_credit_card_pos_txn	5592.0	1.370919e+06	4.562072e+06	0.0	0.000000	7.938500e+03	1.750998e+05	48011222.0
no_credit_card_atm_txn_value_in_mn	5592.0	3.269451e+03	9.775719e+04	0.0	0.000000	2.000000e-01	2.239340e+01	5649796.0
no_credit_card_pos_txn_value_in_mn	5592.0	1.340955e+04	2.452697e+05	0.0	0.000000	2.043485e+01	5.338192e+02	13673238.0
no_debit_cards	5592.0	1.025377e+07	2.760467e+07	0.0	667882.500000	3.067924e+06	9.828595e+06	326078311.0
no_debit_card_atm_txn	5592.0	1.094952e+07	3.727568e+07	0.0	558297.250000	3.179442e+06	8.301565e+06	457980402.0
no_debit_card_pos_txn	5592.0	2.832687e+06	9.257205e+06	0.0	96901.000000	3.936775e+05	1.912702e+06	121463000.0
no_debit_card_atm_txn_value_in_mn	5592.0	1.721422e+05	3.036771e+06	0.0	2410.858883	1.210020e+04	2.959704e+04	194442000.0
no_debit_card_pos_txn_value_in_mn	5592.0	9.531962e+04	2.474881e+06	0.0	192.238250	6.403500e+02	2.846007e+03	169098000.0
debit_trans	5592.0	2.841408e+05	4.918405e+06	0.0	4372.318750	1.551036e+04	3.701568e+04	263755809.0

3. Check for null values and impute them with appropriate values.
   `data.isnull().sum()`

```
month                                0
year                                 0
month_number                         0
start_date                           0
end_date                             0
bank_name                            0
no_atms_on_site                      0
no_atms_off_site                     0
no_pos_on_line                       0
no_pos_off_line                      1
no_credit_cards                      3
no_credit_card_atm_txn               0
no_credit_card_pos_txn               0
no_credit_card_atm_txn_value_in_mn   0
no_credit_card_pos_txn_value_in_mn   0
no_debit_cards                       0
no_debit_card_atm_txn                0
no_debit_card_pos_txn                0
no_debit_card_atm_txn_value_in_mn    0
no_debit_card_pos_txn_value_in_mn    0
dtype: int64
```

From this we get to see the count of which features have null values so that we can compute an appropriate value for them. We must find the median so we can replace this values since both features are numerical.

pd.DataFrame({'value': data['no_pos_off_line'], 'Missing?': data['no_pos_off_line'].isnull()})

	value	Missing?
0	0.0	False
1	0.0	False
2	0.0	False
3	501.0	False
4	404.0	False
...	...	...
5587	0.0	False
5588	0.0	False
5589	0.0	False
5590	0.0	False
5591	0.0	False

Has 1 missing value

```
data['no_pos_off_line'].fillna(data.no_pos_off_line.median(),
inplace = True)

data

Result: We replace all the values with the median(table in the file)
```

pd.DataFrame({'value': data['no_credit_cards'], 'Missing?': data['no_credit_cards'].isnull()})

	value	Missing?
0	0.0	False
1	121514.0	False
2	70776.0	False
3	119248.0	False
4	23436.0	False
...	...	...
5587	0.0	False
5588	0.0	False
5589	0.0	False
5590	0.0	False
5591	0.0	False

Has 3 missing values

```
data['no_credit_cards'].fillna(data.no_credit_cards.median(),
inplace = True)

data

Result: We replace all the values with the median (table in the
file)
```

From the above procedure we see that first we check or the missing values and replace false values with the true values and replace all the 4 missing values with respective medians.

4. Print the number of unique banks in the data. Display the top 10 and bottom 10 banks.

a. consider the same bank with different spellings as different banks to avoid Data Cleansing. First we must drop any duplicate or repeated values so that we don't get extra counts.

```
dupes = data.duplicated()
sum(dupes)
```

```
1
```

```
data.bank_name.drop_duplicates().head(10)
```

```
0           Allahabad Bank
1              Andhra Bank
2           Bank of Baroda
3            Bank of India
4      Bank of Maharashtra
5              Canara Bank
6    Central Bank of India
7         Corporation Bank
8                Dena Bank
9             Indian Bank
Name: bank_name, dtype: object
```

```
sorted(data.bank_name.unique())
```

From the above code what we really do is get all the bank name that are in the table with all the unique names and sort it alphabetically. (Table in the file)

```
len(data.bank_name.unique())
Result: 154
```

To just check the number of the bank names that we have so that we can compare with the new count.

b. Please clean the data and display the count again.

```
data.bank_name=data.bank_name.str.upper()
data.bank_name =
data.bank_name.replace(["LIMITED","LTD.","BKG.
CORP","PLC","LTD","THE"],"",regex=True)
data.bank_name =
data.bank_name.str.replace(".","",regex=True)
data. bank_name = data.bank_name.replace(["CITIBANK","DBS
BANK","TAMILNADU"],["CITI
BANK","DBS","TAMILNAD"],regex=True)
data. bank_name =
data.bank_name.replace(["LAXMI","&"],["LAKSHMI","AND"],regex
=True)
data.bank_name = data.bank_name.str.split().str.join(" ")
data.bank_name = data.bank_name.str.strip()


sorted(data.bank_name.unique())
```

Now on observing we can see that there are not only spelling mistakes, but spacing errors, and wording issues that either needs to be replaced or removed.

First we converted all the names in the uppercase. After this we replace/remove words like LIMITED ,BKG. CORP, PLC ,LTD with just " " space so we don't have to change it again and again.

In case of spacing issue we use the strip() function. If in case of two word being joined without a space we can use split() function.And then we can join the entire.

Result:

```
['ADITYA BIRLA IDEA PAYMENTS BANK',
 'AIRTEL PAYMENTS BANK',
 'ALLAHABAD BANK',
 'AMERICAN EXPRESS',
 'ANDHRA BANK',
 'AU SMALL FINANCE BANK',
 'AXIS BANK',
 'BANDHAN BANK',
 'BANK OF AMERICA',
 'BANK OF BARODA',
 'BANK OF INDIA',
 'BANK OF MAHARASHTRA',
 'BARCLAYS BANK',
 'CANARA BANK',
 'CAPITAL SMALL FINANCE BANK',
 'CATHOLIC SYRIAN BANK',
 'CENTRAL BANK OF INDIA',
 'CITI BANK',
 'CITY UNION BANK',
```

'CORPORATION BANK',
'DBS',
'DCB BANK',
'DENA BANK',
'DEUTSCHE BANK',
'DEVELOPMENT CREDIT BANK',
'DHANALAKSHMI BANK',
'EQUITAS SMALL FINANCE BANK',
'ESAF SMALL FINANCE BANK',
'FEDERAL BANK',
'FINCARE SMALL FINANCE BANK',
'FINO PAYMENTS BANK',
'FIRSTRAND BANK',
'HDFC BANK',
'HONGKONG AND SHANGHAI BKG CORPN',
'HSBC',
'ICICI BANK',
'IDBI',
'IDFC BANK',
'INDIA POST PAYMENTS BANK',
'INDIAN BANK',
'INDIAN OVERSEAS BANK',
'INDUSIND BANK',
'ING VYSYA BANK',
'JAMMU AND KASHMIR BANK',
'JANA SMALL FINANCE BANK',
'JIO PAYMENTS BANK',
'KARNATAKA BANK',
'KARUR VYSYA BANK',
'KOTAK MAHINDRA BANK',
'LAKSHMI VILAS BANK',
'NORTH EAST SMALL FINANCE BANK',
'NSDL PAYMENTS BANK',
'OMAN INTERNATIONAL BANK SAO',
'ORIENTAL BANK OF COMMERCE',
'PAYTM PAYMENTS BANK',
'PUNJAB AND SIND BANK',
'PUNJAB NATIONAL BANK',
'RATNAKAR BANK',
'RBS (ABN AMRO)',
'ROYAL BANK OF SCOTLAND N V',
'SBI COMM AND INT BANK',
'SOUTH INDIAN BANK',
'STANDARD CHARTERED BANK',
'STATE BANK OF BIKANER AND JAIPUR',
'STATE BANK OF HYDERABAD',
'STATE BANK OF INDIA',
'STATE BANK OF MYSORE',
'STATE BANK OF PATIALA',
'STATE BANK OF TRAVANCORE',
'SURYODAY SMALL FINANCE BANK',
'SYNDICATE BANK',
'TAMILNAD MERCANTILE BANK',
'UCO BANK',
'UJJIVAN SMALL FINANCE BANK',
'UNION BANK OF INDIA',
'UNITED BANK OF INDIA',
'UTKARSH SMALL FINANCE BANK',

```
 'VIJAYA BANK',
 'YES BANK']
```

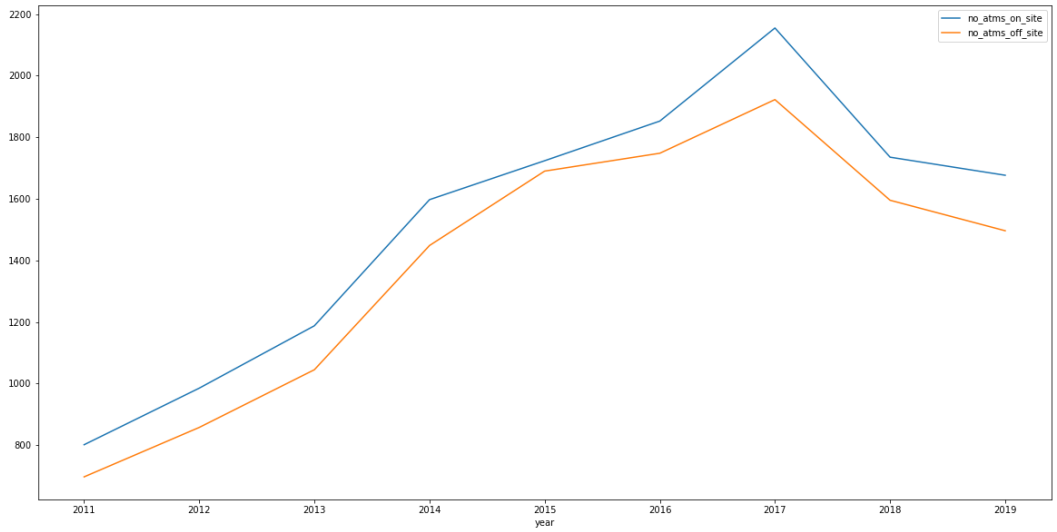
```
len(data.bank_name.unique())
```

```
Result: 79
```

From the above result we can see a huge difference which means we have remove unnecessary data.

5.Check the trend of features w.r.t year. You can use groupby to group the yearly data and display the yearly mean value for each feature (any two features).

```
group_year_mean = data[['no_atms_on_site','no_atms_off_site','year']].
groupby('year').mean()
group_year_mean.plot(figsize=(20,10))
```



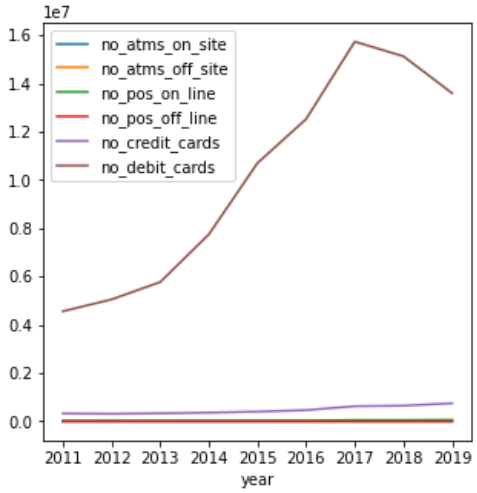
From the above graph we can see that both Number of ATMs deployed on site by the bank and Num ber of ATMs deployed off site by the banks have a similar trend its just that the number of atms depl -oyed offsite is lesser in terms. Both have a peak around 2017.

This shows that both almost increases /decreases together in those specific years. Number of ATMs deployed offsite during 2015 increases slightly. Both have a drop during 2018, also there were no inc rease in the number of ATMs later on as well in 2019.

6. Which feature has shown the highest growth over the years.

a. Check for all features exclude transition related features (number of atm, offsite, pos line, etc)

```
highest_growth1=
data[['no_atms_on_site','no_atms_off_site','no_pos_on_line','no_pos_off_line','no_credit_cards','no_debit_cards','year']].groupby('year').mean()

highest_growth1.plot(figsize=(5,5))
```



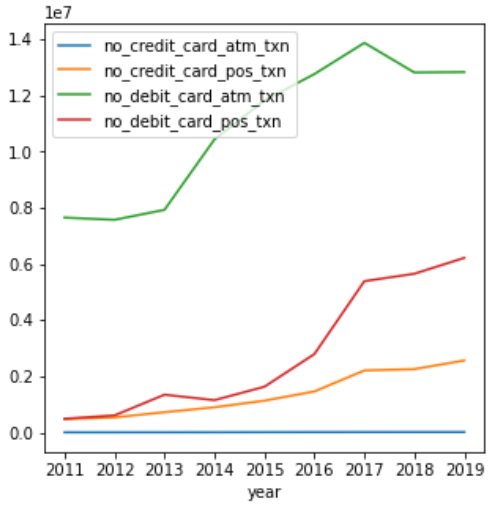
From the above graph we can infer that we have got the all the non transaction data we can see the highest growth in the number of debit card issued is more than any other values that were set in.

From this graph its not really visible but the trend of number ATMs on-site and that of off-site have similar trendline hence they overlap each other.

There a slight but constant increase seen in the credit-card issued along the years

b.Check for the year-on-year growth for transactions related features as they are in millions.

```
highest_growth2 =
data[['no_credit_card_atm_txn','no_credit_card_pos_txn','no_debit_card_atm_txn','no_debit_card_pos_txn','year']].groupby('year').mean()

highest_growth2.plot(figsize=(5,5))
```

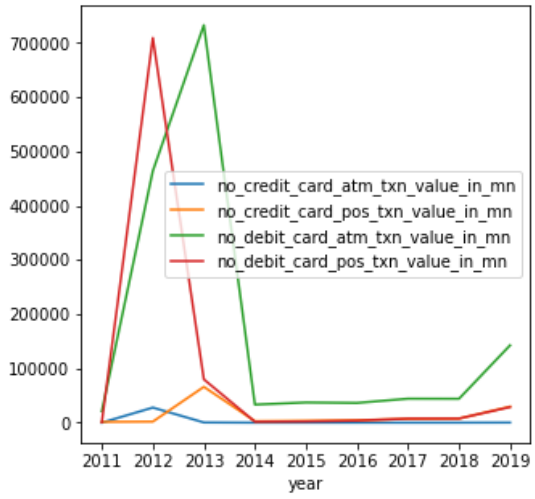


Among all of these trendlines it is seen that Total number of financial transactions done by the debit card issued by the bank at ATM has the highest growth among all. This shown by the green line.

Which means among all these most of the people go for debit card transactions in an ATM. But we can also see an increase in Total number of financial transactions done by the debit card issued by the bank at POS terminal. This shown by the red line.

We can very clearly see a nice difference how the credit card transaction is being done.From trends we see through the yellow line that people prefer using credit card on the POS terminals than using an Atms. This very commonly seen among the people who go for shopping etc.

```
highest_growth3 =
data[['no_credit_card_atm_txn_value_in_mn','no_credit_card_pos_txn_value_in_mn','no_debit_card_atm_txn_value_in_mn','no_debit_card_pos_txn_value_in_mn','year']].groupby('year').mean()

highest_growth3.plot(figsize=(5,5))
```



From the above graph we can infer that the highest value of transaction is done by debit card at the ATMS and slightly lower is debit card transaction value at POS terminal. So from the above we can see the for year 2012 and 2013 looks highest for both pos terminal and atms.

But we can also see a sudden drop in the year 2014 which means the transaction suddenly reduced for both the methods. So, the trendline clearly doesn't show if there is any increase /growth of the transaction values over the year.But we can see the rise for debit card transaction over the year 2019 .

So from both the graphs above we can see there is only increase in the number of transaction over the years done by debit card at the atm.

7. List the top 5 banks having the highest number of ATMs deployed on site (no_atms_on_site).

a. Count the number of ATMs against each back
data['no_atms_on_site']. value_counts(). head(5)

```
0     400
13     78
5      56
12     48
1      40
Name: no_atms_on_site, dtype: int64
```

This gives us a count of ATMS on the site deployed by all the banks. Above we can only see 5 of them.

```
top_bank =
data[['bank_name','no_atms_on_site']].groupby('bank_name',sort=True).sum().sort_values('no_atms_on_site', ascending=False)

top_bank.head(5)
```

bank_name	no_atms_on_site
STATE BANK OF INDIA	2058726
HDFC BANK	515627
BANK OF BARODA	462836
PUNJAB NATIONAL BANK	448890
ICICI BANK	441738

We can see that the highest number banks deployed onsite by SBI and the rest of them come under that.

b. Display the percentage of the "number of ATMs deployed on site" by banks using appropriate plots.

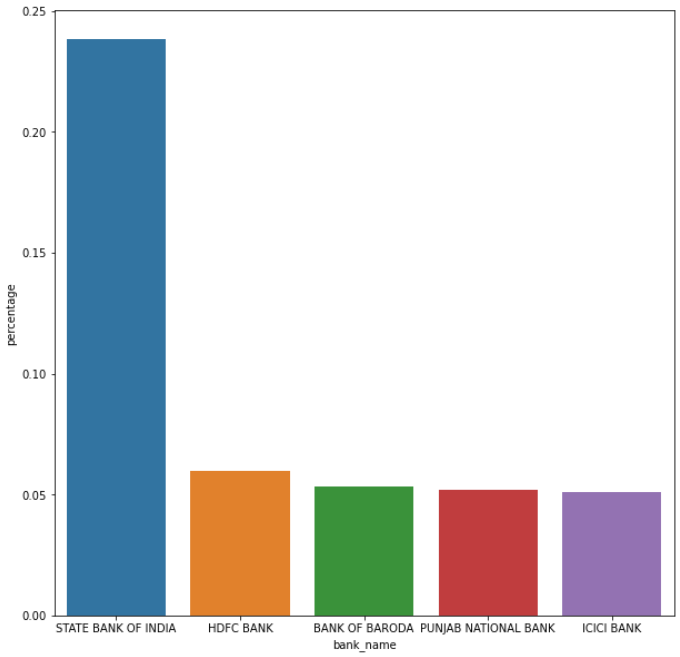
```
top_bank=data[['month', 'year','bank_name','no_atms_on_site']]

top_bank=top_bank.loc[top_bank['year']==2019].loc[top_bank['month']=='August'].groupby('bank_name', sort=True).sum().sort_values(by='no_atms_on_site',ascending=False).reset_index()

top_bank.head(5)
```

	bank_name	year	no_atms_on_site
0	STATE BANK OF INDIA	2019	25667
1	BANK OF BARODA	2019	9400
2	HDFC BANK	2019	6079
3	AXIS BANK	2019	5292
4	PUNJAB NATIONAL BANK	2019	5277

plt.figure(figsize=(10,10))

sns.barplot(x='bank_name',y='percentage', data=top_bank.head(5))

	index	bank_name	year	no_atms_on_site	percentage
0	0	STATE BANK OF INDIA	2019	25667	0.233663
1	1	BANK OF BARODA	2019	9400	0.085574
2	2	HDFC BANK	2019	6079	0.055341
3	3	AXIS BANK	2019	5292	0.048177
4	4	PUNJAB NATIONAL BANK	2019	5277	0.048040



From the above code and graph we find all the percentage of the highest bank that has deployed atms offsite. We can see that the highest number of atms are deployed are by SBI upto 24% onsite then HDFC BANK which is 6%. The rest have almost 5% growth in comparison.

8. Create a new column having total number of transactions

a. Create a new column both credit card and debit card transactions from atm and pos

```
data['debit_trans'] = data[['no_credit_card_atm_txn_value_in_mn',
    'no_credit_card_pos_txn_value_in_mn',
    'no_debit_card_atm_txn_value_in_mn',
    'no_debit_card_pos_txn_value_in_mn']].sum(axis=1)
data
```

This gives us credit card and debit card transaction values and number.(Table in ipynb. File)

b. list top 5 banks having the highest number of transactions.

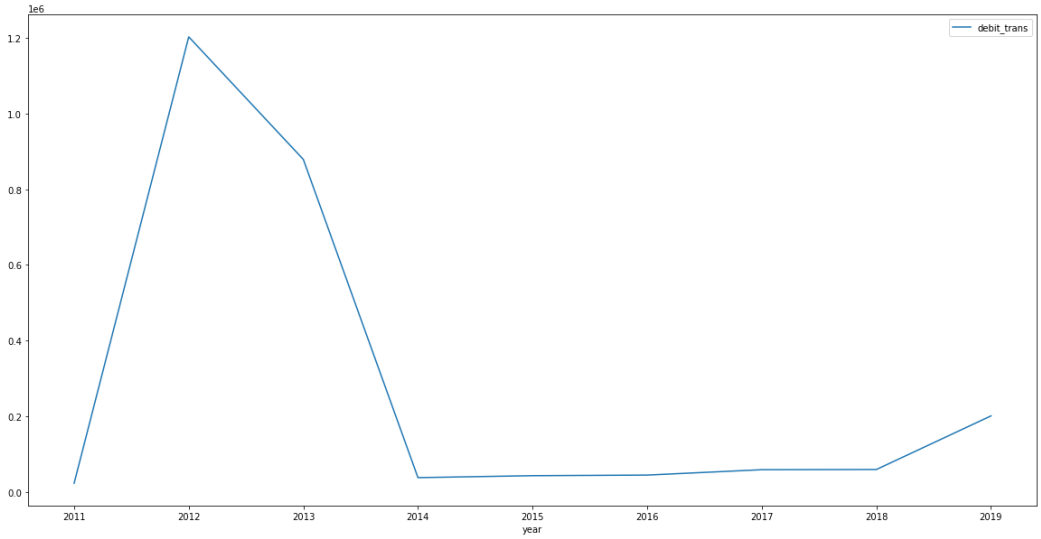
```
top_transaction = data[['bank_name','debit_trans']].groupby('bank_name',sort=True).sum().sort_values('debit_trans', ascending=False)
top_transaction.head(5)
```

bank_name	debit_trans
STATE BANK OF INDIA	5.791317e+08
HDFC BANK	1.310728e+08
ICICI BANK	1.146125e+08
PUNJAB NATIONAL BANK	8.846763e+07
AXIS BANK	7.319479e+07

9. Plot the trend of ‘total transactions” across years using appropriate plots.

```
total_trans =  data[['debit_trans','year']].groupby('year').mean()
total_trans.plot(figsize=(20,10))
```

We have plotted the trendline for total transaction for which we created for the new column as debit-trans. We can see the highest peak at 2012 and a drop during 2013.There is a huge drop during 2014 that remained constant 2018 and then there is rise in 2019.



10. Drop these features for the next question. cols_to _drop = ['month', 'year', 'month_number', 'start date', 'end_date', 'bank_name']

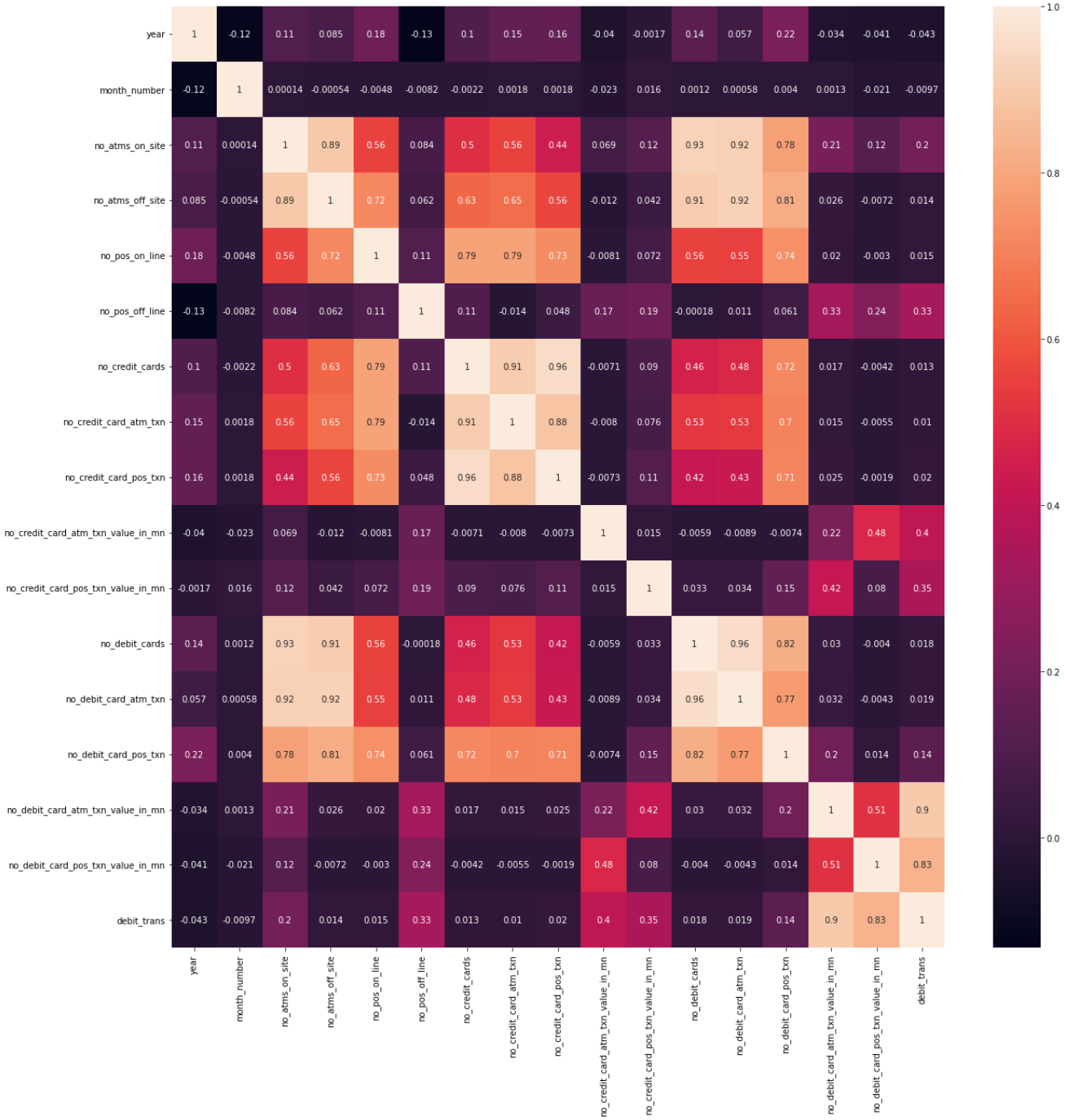
data.drop(['month', 'year','month_number', 'start_date', 'end_date', 'bank_name'],axis=1)

11. Check correlation matrix and display heatmap. Comment on your findings.

plt.figure(figsize = (20,20))

sns.heatmap(data.corr(), annot=True)

Credit card pos transaction dependant on the number of credit cards issued.
Credit card atm transaction depend on the number of credit card issued .
Hence credit card pos transaction depends on atm-transaction.
Similarly for,
Debit card pos transaction dependant on the number of credit cards issued.
Debit card atm transaction depend on the number of debit card issued .
Hence debit card pos transaction depends on atm-transaction.


year 1 -0.12 0.11 0.085 0.18 -0.13 0.1 0.15 0.16 -0.04 -0.0017 0.14 0.057 0.22 -0.034 -0.041 -0.043
month_number -0.12 1 0.00014 -0.00054 -0.0048 -0.0082 -0.0022 0.0018 0.0018 -0.023 0.016 0.0012 0.00058 0.004 0.0013 -0.021 -0.0097
no_atms_on_site 0.11 0.00014 1 0.89 0.56 0.084 0.5 0.56 0.44 0.069 0.12 0.93 0.92 0.78 0.21 0.12 0.2
no_atms_off_site 0.085 -0.00054 0.89 1 0.72 0.062 0.63 0.65 0.56 -0.012 0.042 0.91 0.92 0.81 0.026 -0.0072 0.014
no_pos_on_line 0.18 -0.0048 0.56 0.72 1 0.11 0.79 0.79 0.73 -0.0081 0.072 0.56 0.55 0.74 0.02 -0.003 0.015
no_pos_off_line -0.13 -0.0082 0.084 0.062 0.11 1 0.11 -0.014 0.048 0.17 0.19 -0.00018 0.011 0.061 0.33 0.24 0.33
no_credit_cards 0.1 -0.0022 0.5 0.63 0.79 0.11 1 0.91 0.96 -0.0071 0.09 0.46 0.48 0.72 0.017 -0.0042 0.013
no_credit_card_atm_txn 0.15 0.0018 0.56 0.65 0.79 -0.014 0.91 1 0.88 -0.008 0.076 0.53 0.53 0.7 0.015 -0.0055 0.01
no_credit_card_pos_txn 0.16 0.0018 0.44 0.56 0.73 0.048 0.96 0.88 1 -0.0073 0.11 0.42 0.43 0.71 0.025 -0.0019 0.02
no_credit_card_atm_txn_value_in_mn -0.04 -0.023 0.069 -0.012 -0.0081 0.17 -0.0071 -0.008 -0.0073 1 0.015 -0.0059 -0.0089 -0.0074 0.22 0.48 0.4
no_credit_card_pos_txn_value_in_mn -0.0017 0.016 0.12 0.042 0.072 0.19 0.09 0.076 0.11 0.015 1 0.033 0.034 0.15 0.42 0.08 0.35
no_debit_cards 0.14 0.0012 0.93 0.91 0.56 -0.00018 0.46 0.53 0.42 -0.0059 0.033 1 0.96 0.82 0.03 -0.004 0.018
no_debit_card_atm_txn 0.057 0.00058 0.92 0.92 0.55 0.011 0.48 0.53 0.43 -0.0089 0.034 0.96 1 0.77 0.032 -0.0043 0.019
no_debit_card_pos_txn 0.22 0.004 0.78 0.81 0.74 0.061 0.72 0.7 0.71 -0.0074 0.15 0.82 0.77 1 0.2 0.014 0.14
no_debit_card_atm_txn_value_in_mn -0.034 0.0013 0.21 0.026 0.02 0.33 0.017 0.015 0.025 0.22 0.42 0.03 0.032 0.2 1 0.51 0.9
no_debit_card_pos_txn_value_in_mn -0.041 -0.021 0.12 -0.0072 -0.003 0.24 -0.0042 -0.0055 -0.0019 0.48 0.08 -0.004 -0.0043 0.014 0.51 1 0.83
debit_trans -0.043 -0.0097 0.2 0.014 0.015 0.33 0.013 0.01 0.02 0.4 0.35 0.018 0.019 0.14 0.9 0.83 1
year
month_number
no_atms_on_site
no_atms_off_site
no_pos_on_line
no_pos_off_line
no_credit_cards
no_credit_card_atm_txn
no_credit_card_pos_txn
no_credit_card_atm_txn_value_in_mn
no_credit_card_pos_txn_value_in_mn
no_debit_cards
no_debit_card_atm_txn
no_debit_card_pos_txn
no_debit_card_atm_txn_value_in_mn
no_debit_card_pos_txn_value_in_mn
debit_trans
1.0
0.8
0.6
0.4
0.2
0.0

12. Here we select $\alpha\alpha$ = 0.05 and the population standard deviation is not known.

Is there any significant difference between the mean values of the number of debit Proprietary content.
cards (no_debit_cards) and the number of credit cards (no_credit_cards) over the years?

```
print("The sample size for this problem is",len(data_mean))
```

```
The sample size for this problem is 9
```

```
t_statistic, p_value  = ttest_ind(data_mean['no_debit_cards'],data_mean['no_credit_cards'])
print('tstat',t_statistic)
print('P Value',p_value)
```

```
tstat 6.5320381551835895
P Value 6.908168784546372e-06
```

```
# p_value < 0.05 => alternative hypothesis:

alpha_value = 0.05 # Level of significance
print('Level of significance: %.2f' %alpha_value)
if p_value < alpha_value:
    print('We have evidence to reject the null hypothesis since p value < Level of significance')
else:
    print('We have no evidence to reject the null hypothesis since p value > Level of significance')

print ("Our one-sample t-test p-value=", p_value)
print("We can see there is no great difference in the mean of the number of debit cards and the number of credit cards in the fol
```

tstat 6.5320381551835895
P Value 6.908168784546372e-06

a. Write the null hypothesis and alternative hypothesis for the above case.

- H_0: μA - μB = 0 i.e $\mu A = \mu B$
- H_A: μA - $\mu B \neq 0$ i.e $\mu A \neq \mu B$

Null hypothesis is where all mean are equal
Alternative hypothesis is where one pair of the mean is not equal to 2 sample testing.

b. Define the test that should be used and the significance level.

Two sample t test (Snedecor and Cochran 1989) is used to determine if two population means are equal. A common application is to test if a new treatment or approach or process is yielding better results than the current treatment or approach or process.
Here we select $\alpha\alpha$ = 0.05 and the population standard deviation is not known.

c. Perform the hypothesis testing and report your conclusion.

Level of significance: 0.05
We have evidence to accept the null hypothesis since p value < Level of significance
Our one-sample t-test p-value= 6.908168784546372e-06
We can see there is no great difference in the mean of the number of debit cards and the numbe
r of credit cards in the following years.

13. Is there any significant difference among the means of the following transactions over the years?
i. no_credit_card_atm_txn_value_in_mn

ii. no_credit_card_pos_txn_value_in_mn
iii. no_debit_card_atm_txn_value_in_mn

```
from statsmodels.formula.api import ols      # For n-way ANOVA
from statsmodels.stats.anova import _get_covariance,anova_lm # For n-way ANOVA
formula = 'no_credit_card_atm_txn_value_in_mn ~ no_credit_card_pos_txn_value_in_mn + no_debit_card_atm_txn_value_in_mn'
model = ols(formula, data_mean).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                                  df        sum_sq       mean_sq  \
no_credit_card_pos_txn_value_in_mn  1.0  2.617018e+07  2.617018e+07
no_debit_card_atm_txn_value_in_mn   1.0  5.725820e+08  5.725820e+08
Residual                            6.0  8.482968e+07  1.413828e+07

                                          F    PR(>F)
no_credit_card_pos_txn_value_in_mn   1.851016  0.222554
no_debit_card_atm_txn_value_in_mn   40.498702  0.000707
Residual                                  NaN       NaN
```

a.) Write the null hypothesis and alternative hypothesis for the above case.

Null hypothesis: Groups means are equal (no variation in means of groups)
H0:no_credit_card_atm_txn_value_in_mn = no_credit_card_pos_txn_value_in_mn = no_debit_card_atm_txn_value_in_mn
Alternative hypothesis: At least, one group mean is different from other groups
H1:no_credit_card_atm_txn_value_in_mn != no_credit_card_pos_txn_value_in_mn != no_debit_card_atm_txn_value_in_mn

a. Define the test that should be used and the significance level.

The null hypothesis in ANOVA is always that there is no difference in means. The research or alternative hypothesis is always that the means are not all equal and is usually written in words rather than in mathematical symbols.
Consider a significance level of 5%.

b. Perform the hypothesis testing and report your conclusion.
Null hypothesis is accepted since p value is greater than 0.05, which means there is no significant difference between the mean.

14.) Is there any significant difference between the mean values of no_atms_on_site of two banks (BANK OF BARODA and HDFC BANK LTD) over the years?

bank_group= ['BANK OF BARODA', 'HDFC BANK']

data=data.loc[data['bank_name'].isin(bank_group)].groupby(['bank_name','year']).agg({'no_atms_on_site':'mean'}).reset_index()

data

	bank_name	year	no_atms_on_site
0	BANK OF BARODA	2011	1168.000000
1	BANK OF BARODA	2012	1479.750000
2	BANK OF BARODA	2013	2451.250000
3	BANK OF BARODA	2014	4288.166667
4	BANK OF BARODA	2015	5337.250000
5	BANK OF BARODA	2016	6209.500000
6	BANK OF BARODA	2017	6338.666667
7	BANK OF BARODA	2018	6110.333333
8	BANK OF BARODA	2019	8218.125000
9	HDFC BANK	2011	3149.888889
10	HDFC BANK	2012	4313.583333
11	HDFC BANK	2013	5040.250000
12	HDFC BANK	2014	4752.000000
13	HDFC BANK	2015	5183.000000
14	HDFC BANK	2016	5581.083333
15	HDFC BANK	2017	5813.083333
16	HDFC BANK	2018	5890.666667
17	HDFC BANK	2019	6049.250000

```
formula = '(no_atms_on_site ~ bank_name)'
model = ols(formula, data).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                df        sum_sq       mean_sq          F    PR(>F)
bank_name      1.0  9.668674e+05  9.668674e+05   0.283483  0.601744
Residual      16.0  5.457069e+07  3.410668e+06        NaN       NaN
```

a. Write the null hypothesis and alternative hypothesis for the above case.
Null hypothesis: Groups means are equal (no variation in means of groups)
Alternative hypothesis: At least, one group mean is different from other groups
Mean values of HDFC bank and BANK OF BARODA is same for null hypothesis
Mean values of HDFC BANK AND BANK OF BARODA is same for alternative hypothesis

b. Define the test that should be used and the significance level.
The null hypothesis in ANOVA is always that there is no difference in means. The research or alternat ive hypothesis is always that the means are not all equal and is usually written in words rather than i n mathematical symbols.
Consider a significance level of 5%.

c. Perform the hypothesis testing and report your conclusion.
Null hypothesis is accepted since p value is greater than 0.05, which means there is no significant differen ce between the mean.