LECTURE-13

ANALYSIS AND VISUALIZATION WITH PANDAS DATAFRAME (PART-3)

Sources of DataSets

There are various online sources from where useful Datasets can be downloaded absolutely free of cost Some of sources are listed below:

1. Google Dataset Search: https://datasetsearch.research.google.com/
2. Kaggle: https://www.kaggle.com/datasets
3. Data.Gov: https://data.gov/
4. UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets.php
5. Global Health Observatory Data Repository: https://apps.who.int/gho/data/node.home

3. CANADIAN IMMIGRATION DATA SET

https://open.canada.ca/data/en/dataset/2894b1fa-d71e-4793-959f-48329bd38132

https://www.kaggle.com/datasets/umerkk12/canada-immigration-dataset

PROGRAM 3(i) Reading and Loading the Dataset

```
import pandas as pd

CANADA_IMMIGRATION=pd.read_csv('Canadian Immigration Dataset.csv')

CANADA_IMMIGRATION.head(5)
```

	Unnamed: 0	Draw Number	Date	Immigration program	Invitations issued	CRS score of lowest-ranked candidate invited	Date (hidden)	Programs covered	Month	Year	month_year	Date Full
0	0	172	1/7/2021	Canadian Experience Class	4750	461	7/1/2021	Canadian Experience Class	1	2021	1/1/2021	7-Jan-21
1	1	171	1/6/2021	Provincial Nominee Program	250	813	6/1/2021	Provincial Nominee Program	1	2021	1/1/2021	6-Jan-21
2	2	170	12/23/2020	No program specified	5000	468	12/23/2020	Canadian Experience Class Federal Skilled Wor...	12	2020	12/1/2020	23-Dec-20
								Canadian				

PROGRAM 3(ii) To determine the Number of ROWS and COLUMNS in the given Dataset

```
print('SHAPE: ROWS X COLUMNS')
print(CANADA_IMMIGRATION.shape)
```

```
SHAPE: ROWS X COLUMNS
(173, 12)
```

```
print('DATA TYPES OF DIFFERENT ELEMENTS')
print(CANADA_IMMIGRATION.dtypes)
```

```
DATA TYPES OF DIFFERENT ELEMENTS
Unnamed: 0                                    int64
Draw Number                                   int64
Date                                         object
Immigration program                          object
Invitations issued                            int64
CRS score of lowest-ranked candidate invited  int64
Date (hidden)                                object
```

```
Programs covered                                object
Month                                            int64
Year                                             int64
month_year                                      object
Date Full                                       object
dtype: object
```

PROGRAM 3(iii)

DROPPING (DELETING) CERTAIN COLUMNS FROM THE DATASET

Using .drop( ) function to DELETE certain columns

```
CANADA_IMMIGRATION1 = CANADA_IMMIGRATION.drop(['Unnamed: 0','Draw Number',
                                               'Date (hidden)','Month',
                                               'Programs covered',
                                               'month_year','Date Full']
                                              ,axis=1)

''' .drop() function with arguement 'axis=1' is used to drop certain columns'''

display(CANADA_IMMIGRATION1)
```

	Date	Immigration program	Invitations issued	CRS score of lowest-ranked candidate invited	Year
0	1/7/2021	Canadian Experience Class	4750	461	2021
1	1/6/2021	Provincial Nominee Program	250	813	2021
2	12/23/2020	No program specified	5000	468	2020
3	12/9/2020	No program specified	5000	469	2020
4	11/25/2020	No program specified	5000	469	2020
...	...	...	...	...	...
168	3/20/2015	No program specified	1620	481	2015
169	2/27/2015	No program specified	1187	735	2015
170	2/20/2015	Canadian Experience Class	849	808	2015
171	2/7/2015	No program specified	779	818	2015
172	1/31/2015	No program specified	779	886	2015

173 rows × 5 columns

PROGRAM 3(iv)

RENAMING CERTAIN COLUMNS FROM THE DATASET

Using .Rename( ) function to rename certain columns

```
CANADA_IMMIGRATION1 = CANADA_IMMIGRATION1.rename(columns=
                                                 {'CRS score of lowest-ranked candidate invited':
                                                 'Lowest CRS'})

display(CANADA_IMMIGRATION1)
```

	Date	Immigration program	Invitations issued	Lowest CRS	Year
0	1/7/2021	Canadian Experience Class	4750	461	2021
1	1/6/2021	Provincial Nominee Program	250	813	2021
2	12/23/2020	No program specified	5000	468	2020
3	12/9/2020	No program specified	5000	469	2020
4	11/25/2020	No program specified	5000	469	2020
...	...	...	...	...	...
168	3/20/2015	No program specified	1620	481	2015
169	2/27/2015	No program specified	1187	735	2015
170	2/20/2015	Canadian Experience Class	849	808	2015
171	2/7/2015	No program specified	779	818	2015
172	1/31/2015	No program specified	779	886	2015

173 rows × 5 columns

PROGRAM 3(v)

CORRECTING THE FORMAT OF 'DATE' COLUMN

USING pd.strftime() to tranform all the dates in the date column in the format dd-mm-yyyy

```
DATE = CANADA_IMMIGRATION1['Date']
DATE = pd.to_datetime(DATE)

DATE=DATE.dt.strftime('%d-%m-%Y')
CANADA_IMMIGRATION1['Date']=DATE

display(CANADA_IMMIGRATION1)
```

	Date	Immigration program	Invitations issued	Lowest CRS	Year
0	07-01-2021	Canadian Experience Class	4750	461	2021
1	06-01-2021	Provincial Nominee Program	250	813	2021
2	23-12-2020	No program specified	5000	468	2020
3	09-12-2020	No program specified	5000	469	2020
4	25-11-2020	No program specified	5000	469	2020
...	...	...	...	...	...
168	20-03-2015	No program specified	1620	481	2015
169	27-02-2015	No program specified	1187	735	2015
170	20-02-2015	Canadian Experience Class	849	808	2015
171	07-02-2015	No program specified	779	818	2015
172	31-01-2015	No program specified	779	886	2015

173 rows × 5 columns

PROGRAM 3(vi)

Sorting data elements according to year

```
CANADA_IMMIGRATION1 = CANADA_IMMIGRATION1.sort_values(by='Year',ascending=True)
CANADA_IMMIGRATION1=CANADA_IMMIGRATION1.reset_index()

display(CANADA_IMMIGRATION1)
```

	index	Date	Immigration program	Invitations issued	Lowest CRS	Year
0	172	31-01-2015	No program specified	779	886	2015
1	150	18-12-2015	No program specified	1503	460	2015
2	151	04-12-2015	No program specified	1451	461	2015
3	152	27-11-2015	No program specified	1559	472	2015
4	153	13-11-2015	No program specified	1506	484	2015
...	...	...	...	...	...	...
168	23	27-05-2020	Provincial Nominee Program	385	757	2020
169	22	28-05-2020	Canadian Experience Class	3515	440	2020
170	30	09-04-2020	Canadian Experience Class	3294	464	2020
171	1	06-01-2021	Provincial Nominee Program	250	813	2021
172	0	07-01-2021	Canadian Experience Class	4750	461	2021

173 rows × 6 columns

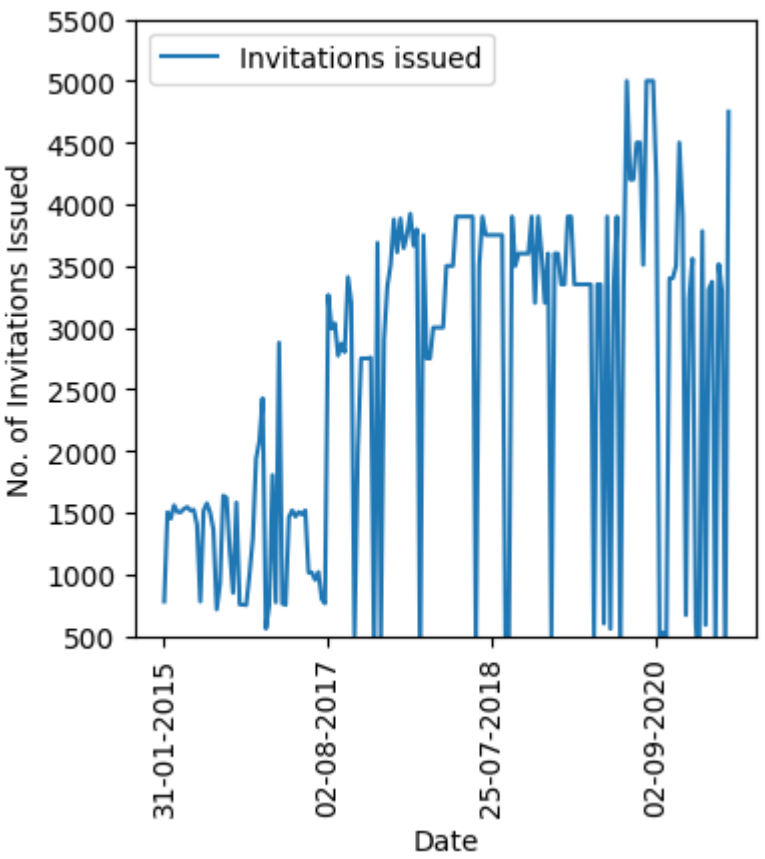
BASIC PLOTTING using Matplotlib

PROGRAM 3(vi) - LINE PLOT

Invitations issued Vs. Date

```
import matplotlib.pyplot as plt
import numpy as np

CANADA_IMMIGRATION1.plot.line(x='Date',y='Invitations issued',figsize=(4,4),
                              ylabel = 'No. of Invitations Issued',
                              xlabel='Date')
```

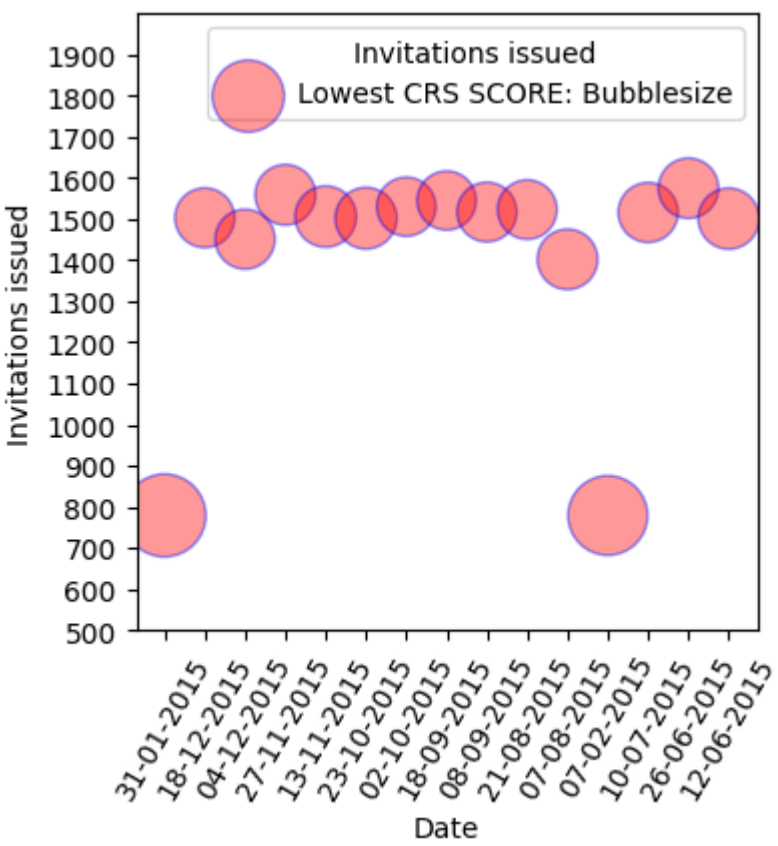
```
'''.head(20) is used for plotting only first 20 points in the dataset'''

plt.ylim(500,5500)
plt.yticks(np.arange(500,6000,500))

plt.xticks(rotation=90)

plt.show()
```



PROGRAM 3(vii) - SCATTER PLOT

Invitations issued Vs. Date

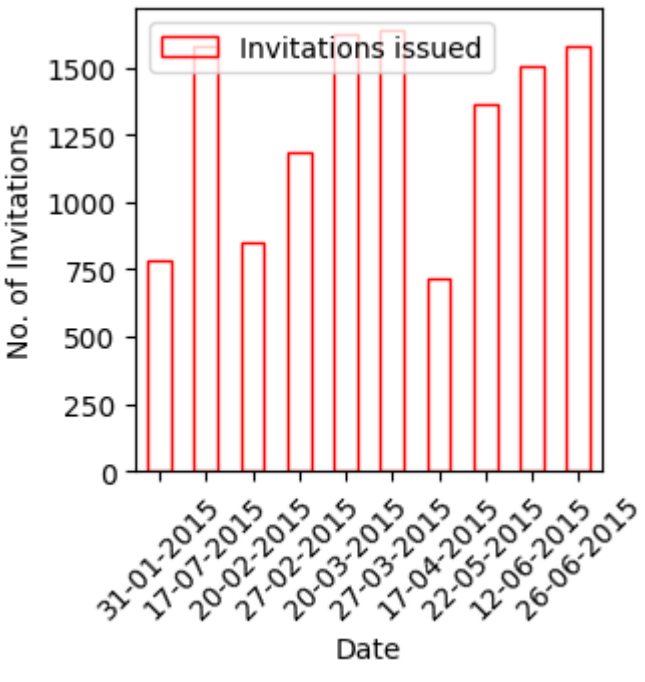
```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1.head(15).plot.scatter(x='Date', y='Invitations issued',
                                          marker='o',s='Lowest CRS', color ='red',
                                          alpha=0.4,edgecolor='b',figsize=(4,4))
plt.xlabel = 'Year'
plt.ylabel = 'No. of Invitations Issued'

plt.xticks(rotation=60)

plt.ylim(500,2000)
plt.yticks(np.arange(500,2000,100))

plt.legend(loc='best',title ='Invitations issued',labels=['Lowest CRS SCORE: Bubblesize'])

plt.show()
```



PROGRAM 3(viii) - BAR PLOT

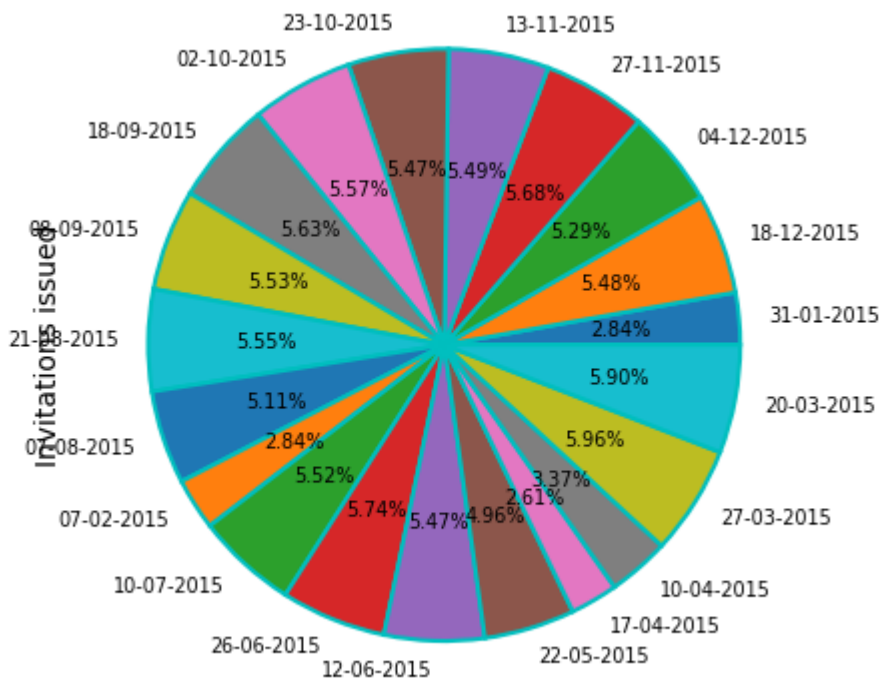
Invitations issued Vs. Date

```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1.head(10).plot.bar(x='Date', y='Invitations issued',
                                      edgecolor='r', linewidth=1, fill=False,
                                      xlabel='Date', ylabel='No. of Invitations',
                                      figsize=(3,3))
plt.xticks(rotation=45)
plt.show()
```



PROGRAM 3(ix) - PIE CHART

```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1=CANADA_IMMIGRATION1.set_index('Date')

CANADA_IMMIGRATION1['Invitations issued'].head(20).plot.pie(autopct='%1.2f%%',
                                                           textprops={'size': 'x-small'},
                                                           wedgeprops={'linewidth': 1.5, 'edgecolor': 'c'})
plt.ylabel = 'No. of Invitations Issued'
```



DATA AGGREGATION

It is the process in which large volume of data gathered from multiple sources is compiled and presented in a summarized manner which is more useful for statistical analysis

- For example: collecting complete data related to sales of a particular product and then grouping/aggregating the data on the basis of age, profession and residence of the customer buying that product

Data Aggregation (Example)

Original (complete) data set related to employee information for companies A, B and C

Company	Name	Age	Wages	Education.University	Productivity
A	`Wayne	26	50000	1	100
A	Duane	27	70000	1	120
B	William	28	70000	1	120
C	Rafael	32	60000	0	95
A	John	28	50000	0	88
B	Eric	24	70000	1	115
B	James	34	65000	1	100
C	Pablo	30	50000	0	90
C	Tammy	25	55000	1	120

Data Aggregated (grouped) in terms of Avg. Age , Avg. Wages, Education and Avg. Productivity of employees in companies A, B and C

Company	average Age	average Wages	Sum. Education.Unive	average Productivity
A	27	56600	2	102,6
B	28,6	68333	3	111,6
C	29	55000	1	101,6

Grouping in Pandas using groupby() function

Grouping is used to group data using some criteria from our dataset.

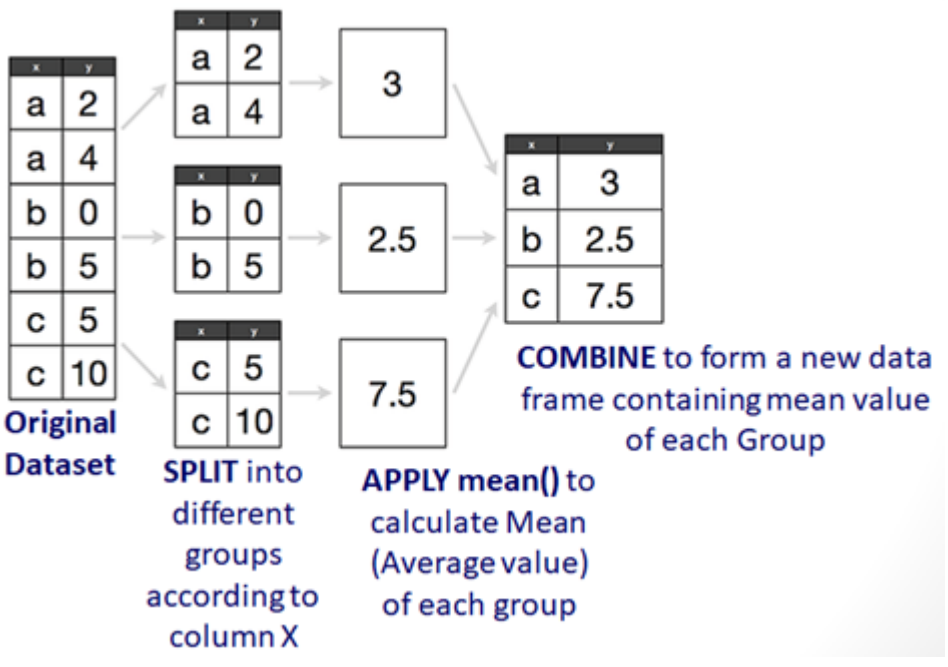
It is used as **SPLIT-APPLY-COMBINE strategy**.

- Splitting the data into groups based on some criteria.
- Applying a function to each group independently.
- Combining the results into a NEW DATAFRAME

Data Aggregation and groupby functions : PANDAS

Pandas python library offers **.agg( ) and groupby( )** function sto perform DATA AGGREGATION and GROUPING.

Grouping is used to group data using some criteria from our dataset. **SPLIT-APPLY-COMBINE** strategy which is used to perform Data Aggregation is illustrated below



PROGRAM 5

DATA AGGREGATION

1. Load the given **Canada Immigration Dataset**
2. Drop the following columns: 'Unnamed: 0','Draw Number','Date (hidden)','Month', 'Programs covered','month_year','Date Full'
3. GROUP the given DATA according to 'Year'
4. **PERFORM DATA AGGREGATION TO DETERMINE**
    - TOTAL NUMBER OF INVITATIONS IN EACH YEAR
    - MEAN VALUE OF INVITATIONS ISSUED EACH YEAR

- MINIMUM VALUE OF INVITATIONS ISSUED
- MAXIMUM VALUE OF INVITATIONS ISSUED

PROGRAM 5(A): Loading the Dataset Removing all the undesired and non-numeric columns

```
import pandas as pd

CANADA_IMMIGRATION=pd.read_csv('Canadian Immigration Dataset.csv')

print("ORIGINAL DATASET")
display(CANADA_IMMIGRATION.head(5))

CANADA_IMMIGRATION1 = CANADA_IMMIGRATION.drop(['Unnamed: 0','Draw Number',
                                               'Date (hidden)','Month','Programs covered',
                                              'month_year','Date Full']
                                             ,axis=1)

CANADA_IMMIGRATION1=CANADA_IMMIGRATION1.select_dtypes(include='number')

'''.select_dtypes(include='number') is used to select only numeric columns'''

CANADA_IMMIGRATION1 = CANADA_IMMIGRATION1.rename(columns=
                                               {'CRS score of lowest-ranked candidate invited':
                                               'Lowest CRS'})

print("MODIFIED DATASET")
display(CANADA_IMMIGRATION1)
```

ORIGINAL DATASET

	Unnamed: 0	Draw Number	Date	Immigration program	Invitations issued	CRS score of lowest-ranked candidate invited	Date (hidden)	Programs covered	Month	Year	month_year	Date Full
0	0	172	1/7/2021	Canadian Experience Class	4750	461	7/1/2021	Canadian Experience Class	1	2021	1/1/2021	7-Jan-21
1	1	171	1/6/2021	Provincial Nominee Program	250	813	6/1/2021	Provincial Nominee Program	1	2021	1/1/2021	6-Jan-21
2	2	170	12/23/2020	No program specified	5000	468	12/23/2020	Canadian Experience Class Federal Skilled Wor...	12	2020	12/1/2020	23-Dec-20
3	3	169	12/9/2020	No program specified	5000	469	9/12/2020	Canadian Experience Class Federal Skilled Wor...	12	2020	12/1/2020	9-Dec-20
4	4	168	11/25/2020	No program specified	5000	469	11/25/2020	Canadian Experience Class Federal Skilled Wor...	11	2020	11/1/2020	25-Nov-20

MODIFIED DATASET

	Invitations issued	Lowest CRS	Year
0	4750	461	2021
1	250	813	2021
2	5000	468	2020
3	5000	469	2020
4	5000	469	2020

PROGRAM 5(B)

Grouping the given DataSet according to 'Year' **Performing Data Aggregation using .agg( ) to determine the following parameters in each year

- Sum and Mean of 'Invitations Issued'
- Sum and Mean of Lowest CRS

```
print("DATASET grouped according to 'Year'")

CANADA_IMMIGRATION1_YEAR=CANADA_IMMIGRATION1.groupby('Year')

CANADA_IMMIGRATION1_A=CANADA_IMMIGRATION1_YEAR.agg(['sum','mean'])

display(CANADA_IMMIGRATION1_A)
```

```
DATASET grouped according to 'Year'
```

	Invitations issued		Lowest CRS	
	sum	mean	sum	mean
Year				
2015	31063	1350.565217	12375	538.043478
2016	33782	1251.185185	13266	491.333333
2017	86023	2867.433333	13285	442.833333
2018	89800	3207.142857	12558	448.500000
2019	85300	3280.769231	11721	450.807692
2020	107350	2901.351351	19753	533.864865
2021	5000	2500.000000	1274	637.000000

PROGRAM 5(C)

DISPLAYING 'SUM' and 'MEAN' of 'NUMBER OF INVITATIONS ISSUED' and 'LOWEST CRS SCORE' EACH YEAR' through LINE PLOT

```
# Displaying column names of grouped DataFrame

display(CANADA_IMMIGRATION1_A.columns)
```

```
MultiIndex([('Invitations issued',  'sum'),
            ('Invitations issued', 'mean'),
            (        'Lowest CRS',  'sum'),
            (        'Lowest CRS', 'mean')],
           )
```

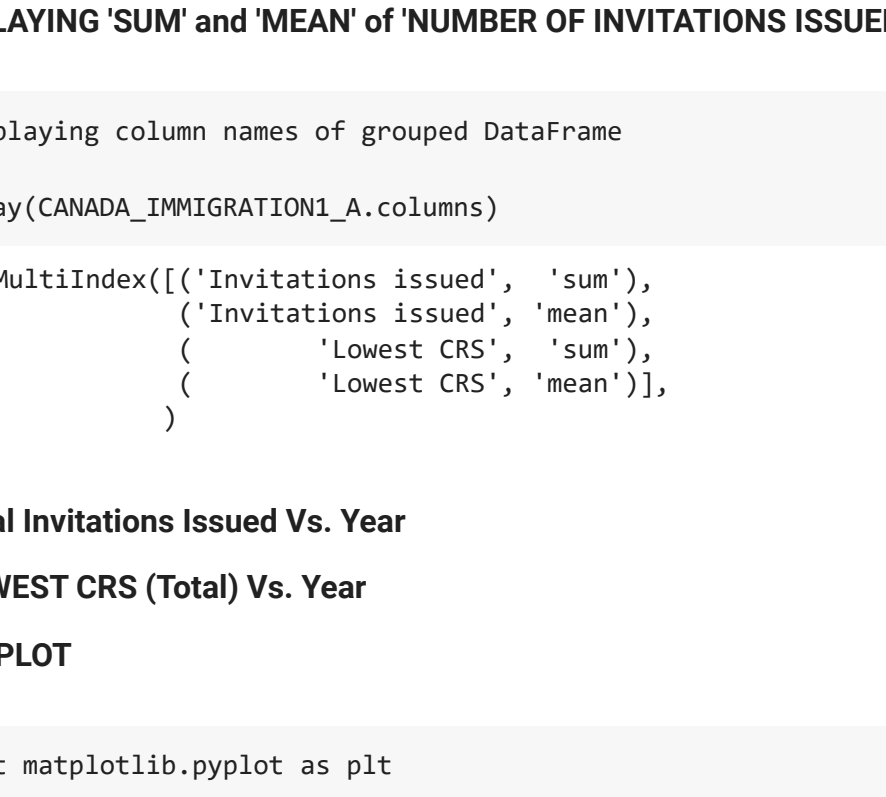
i. Total Invitations Issued Vs. Year

ii.LOWEST CRS (Total) Vs. Year

LINE PLOT

```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1_A.plot(y=[('Invitations issued',  'sum'),('Lowest CRS',  'sum')]
                           ,marker='X', ms=10, ylabel= 'TOTAL'
                           ,figsize=(4.5,4.5))

plt.legend(labels=['Invitations issued','Lowest CRS'],loc='upper left', title='TOTAL')
```

```
<matplotlib.legend.Legend at 0x7f16258479d0>
```



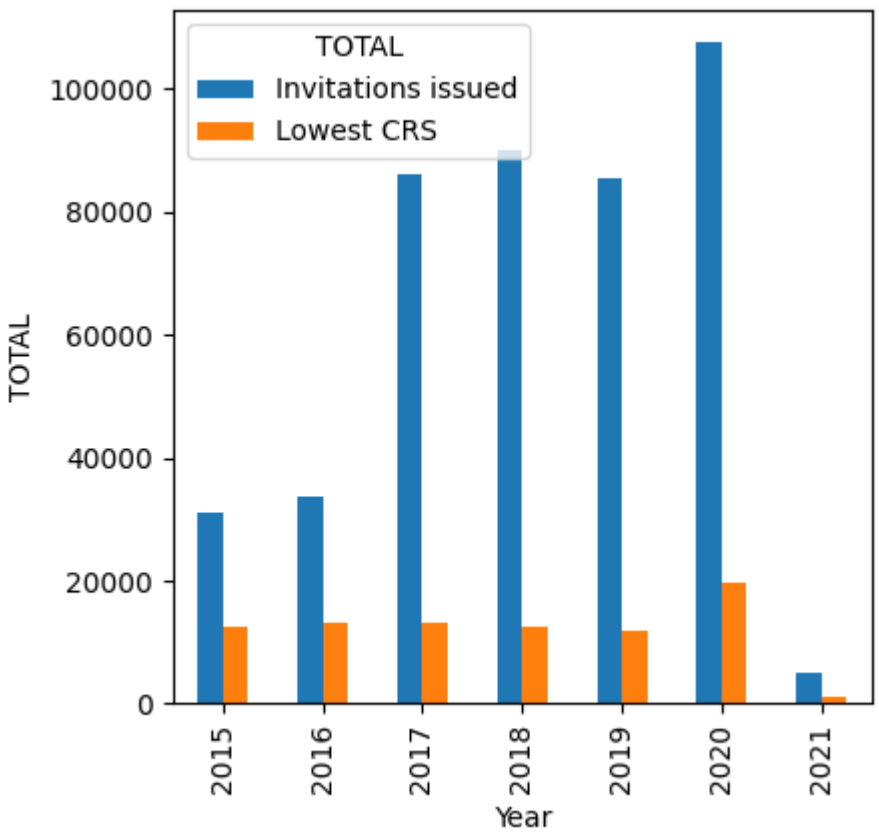
i. Total Invitations Issued Vs. Year

ii.LOWEST CRS (Total) Vs. Year

BAR PLOT

```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1_A.plot.bar(y=[('Invitations issued',  'sum'),('Lowest CRS',  'sum')],
                           ylabel= 'TOTAL',
                           figsize=(4.5,4.5))

plt.legend(labels=['Invitations issued','Lowest CRS'],loc='upper left', title='TOTAL')
```

```
<matplotlib.legend.Legend at 0x7f162366b730>
```



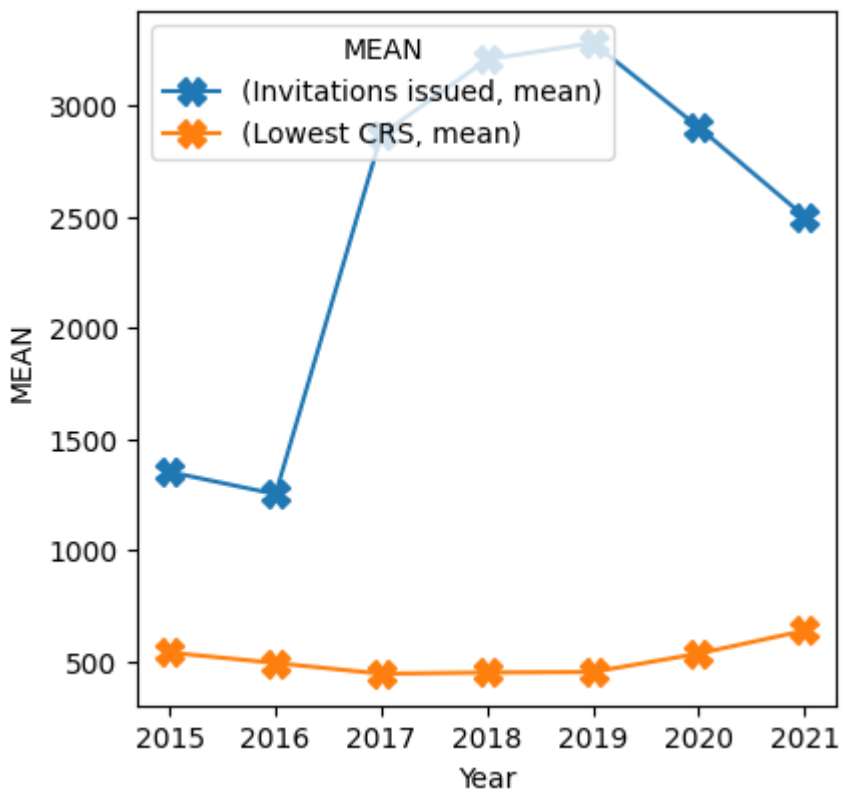
i. Invitations Issued(Mean) Vs. Year

ii.LOWEST CRS (Mean) Vs. Year

BAR PLOT

```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1_A.plot(y=[('Invitations issued',  'mean'),('Lowest CRS',  'mean')]
                           ,marker='X', ms=10, ylabel= 'MEAN'
                           ,figsize=(4.5,4.5))
plt.legend(loc='upper left', title='MEAN')
```

```
<matplotlib.legend.Legend at 0x7f16237c87c0>
```



PROGRAM 5(D)

Grouping the given DataSet according to 'Year'

Performing Data Aggregation using .agg( ) to determine the following parameters in each year

- Min and Max value of 'Invitations Issued'
- Min and Max value of 'Lowest CRS'

```
print("DATASET grouped according to 'Year'")

CANADA_IMMIGRATION1_YEAR=CANADA_IMMIGRATION1.groupby('Year')

CANADA_IMMIGRATION1_B=CANADA_IMMIGRATION1_YEAR.agg(['min','max'])

display(CANADA_IMMIGRATION1_B)
```

```
DATASET grouped according to 'Year'
```

	Invitations issued		Lowest CRS	
	min	max	min	max
Year				
2015	715	1637	450	886
2016	559	2878	453	786
2017	143	3923	199	775
2018	200	3900	284	902
2019	500	3900	332	475
2020	118	5000	415	808
2021	250	4750	461	813

PROGRAM 5(E)

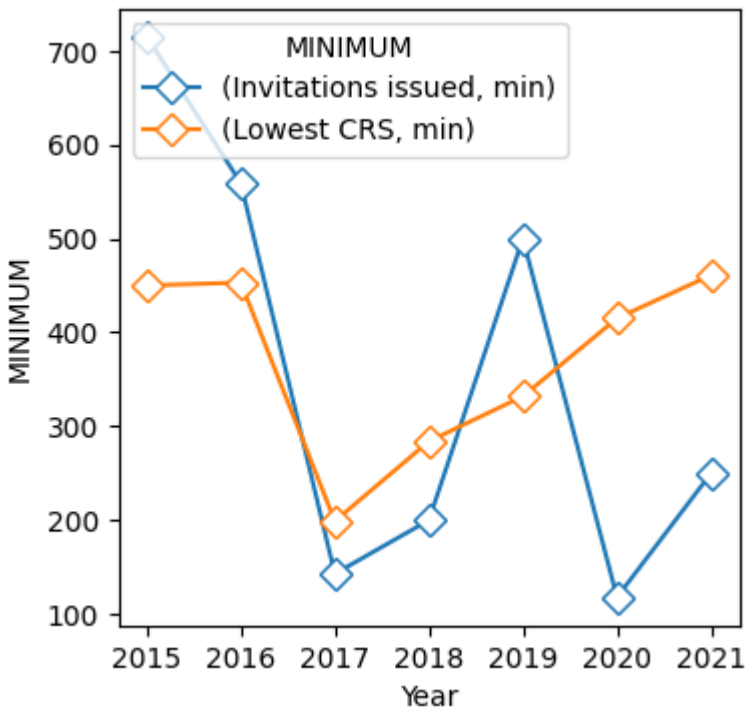
DISPLAYING 'MIN' and 'MAX' values of 'NUMBER OF INVITATIONS ISSUED' and 'LOWEST CRS SCORE' EACH YEAR' through LINE PLOT

```
# Displaying column names of grouped DataFrame

display(CANADA_IMMIGRATION1_B.columns)
```

```
MultiIndex([('Invitations issued', 'min'),
            ('Invitations issued', 'max'),
            (        'Lowest CRS', 'min'),
            (        'Lowest CRS', 'max')],
           )
```

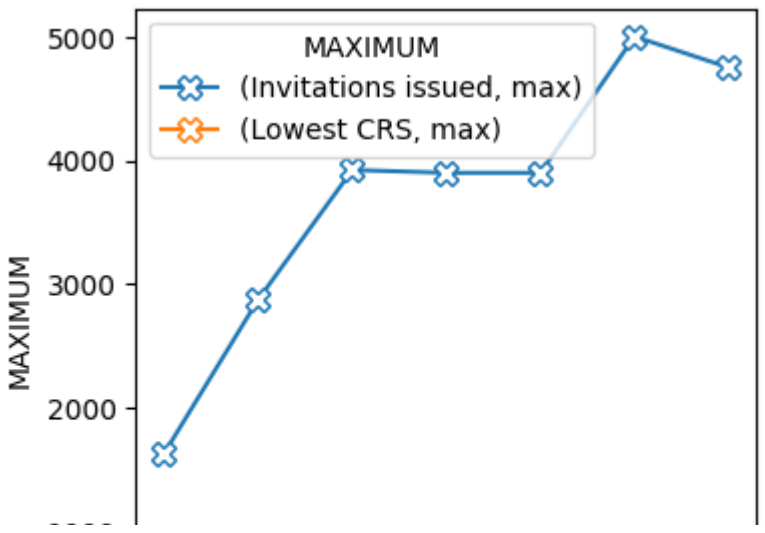
```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1_B.plot(y=[('Invitations issued', 'min'),('Lowest CRS', 'min')]
                           ,marker='D', ms=8, mfc='white',ylabel= 'MINIMUM'
                           ,figsize=(4,4))
plt.legend(loc='upper left', title='MINIMUM')
```

```
<matplotlib.legend.Legend at 0x7f2051fbdde0>
```




```
import matplotlib.pyplot as plt

CANADA_IMMIGRATION1_B.plot(y=[('Invitations issued', 'max'),('Lowest CRS', 'max')]
                           ,marker='X', ms=8, mfc='white',ylabel= 'MAXIMUM'
                           ,figsize=(4,4))
plt.legend(loc='upper left', title='MAXIMUM')
```

```
<matplotlib.legend.Legend at 0x7f2050cf9e40>
```



PROGRAM 5(F)

Grouping the dataset according to 'Year' Performing Data Aggregation and Display 'sum', 'mean', 'min', 'max' value for each column in each year together

```
CANADA_IMMIGRATION1_YEAR = CANADA_IMMIGRATION1.groupby('Year')

CANADA_IMMIGRATION1_C = CANADA_IMMIGRATION1_YEAR.agg(['sum','mean','min','max'])

display(CANADA_IMMIGRATION1_C)
```

	Invitations issued				Lowest CRS			
	sum	mean	min	max	sum	mean	min	max
Year								
2015	31063	1350.565217	715	1637	12375	538.043478	450	886
2016	33782	1251.185185	559	2878	13266	491.333333	453	786
2017	86023	2867.433333	143	3923	13285	442.833333	199	775
2018	89800	3207.142857	200	3900	12558	448.500000	284	902
2019	85300	3280.769231	500	3900	11721	450.807692	332	475
2020	107350	2901.351351	118	5000	19753	533.864865	415	808
2021	5000	2500.000000	250	4750	1274	637.000000	461	813

PROGRAM 5(G)

Grouping the dataset according to 'Year'

Performing Data Aggregation using describe function to display the values of all the statistical parameters for each column in each year together

```
print("DATASET grouped according to 'Year'")

CANADA_IMMIGRATION1_YEAR=CANADA_IMMIGRATION1.groupby('Year')

CANADA_IMMIGRATION1_D=CANADA_IMMIGRATION1_YEAR.agg(['describe'])

display(CANADA_IMMIGRATION1_D)
```

```
DATASET grouped according to 'Year'
```

	Invitations issued								Lowest CRS					
	describe								describe					
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%
Year														
2015	23.0	1350.565217	307.401548	715.0	1274.00	1503.0	1537.50	1637.0	23.0	538.043478	144.045598	450.0	457.5	469.0
2016	27.0	1251.185185	581.821942	559.0	762.50	1014.0	1511.50	2878.0	27.0	491.333333	62.184589	453.0	470.0	482.0
2017	30.0	2867.433333	1111.752785	143.0	2751.75	3118.5	3659.75	3923.0	30.0	442.833333	96.268333	199.0	431.5	438.5
2018	28.0	3207.142857	1069.416099	200.0	3000.00	3625.0	3900.00	3900.0	28.0	448.500000	98.040619	284.0	440.0	442.0
2019	26.0	3280.769231	845.585823	500.0	3350.00	3350.0	3600.00	3900.0	26.0	450.807692	32.851203	332.0	451.0	459.5
2020	37.0	2901.351351	1692.586469	118.0	606.00	3400.0	4200.00	5000.0	37.0	533.864865	122.817200	415.0	464.0	471.0
2021	2.0	2500.000000	3181.980515	250.0	1375.00	2500.0	3625.00	4750.0	2.0	637.000000	248.901587	461.0	549.0	637.0