INDUSTRIAL TRAINING DAILY DIARY
DAY 13

09 July, 2025

Topic : Practice of Pandas on real world datasets

Objectives:

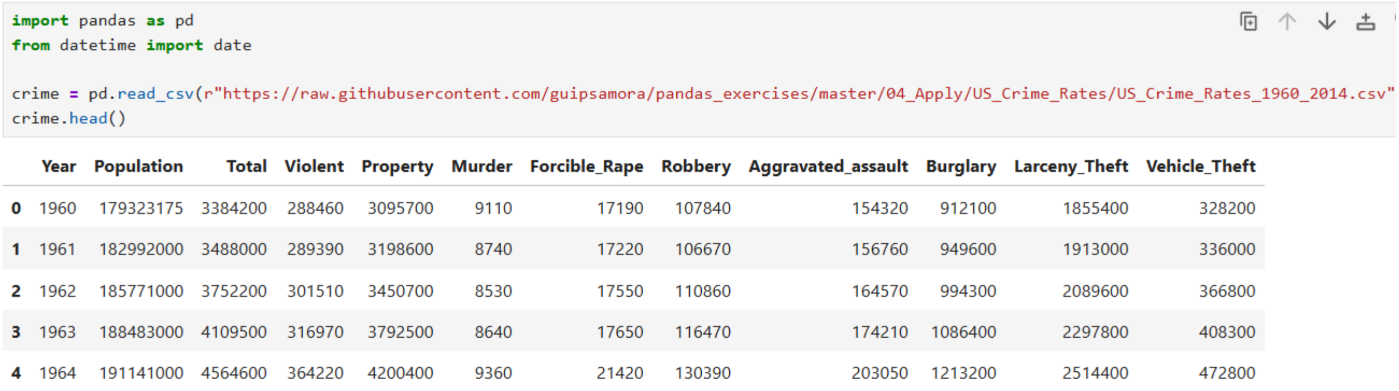
- To understand how to work with string data within a pandas DataFrame.
- To apply string functions using the .str accessor on Series.
- To perform common operations like:
  - Converting to lowercase and uppercase using .str.lower() / .str.upper()
  - Removing whitespace with .str.strip(), .str.lstrip(), and .str.rstrip()
  - Finding substrings using .str.contains() and .str.find()
  - Replacing characters or substrings using .str.replace()
  - Splitting strings with .str.split() and extracting parts using .str.get()
  - Checking for string patterns (like prefixes/suffixes) using .str.startswith() and .str.endswith()
- To clean and preprocess textual data for analysis or machine learning tasks.

- **Exercise on US_CRIMES Dataset**

Step 1. Import the necessary libraries

Step 2. Import the dataset from this [address] (https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/04_Apply/US_Crime_Rates/US_Crime_Rates_1960_2014.csv).

Step 3. Assign it to a variable called crime.



```
import pandas as pd
from datetime import date

crime = pd.read_csv(r"https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/04_Apply/US_Crime_Rates/US_Crime_Rates_1960_2014.csv"
crime.head()
```

	Year	Population	Total	Violent	Property	Murder	Forcible_Rape	Robbery	Aggravated_assault	Burglary	Larceny_Theft	Vehicle_Theft
0	1960	179323175	3384200	288460	3095700	9110	17190	107840	154320	912100	1855400	328200
1	1961	182992000	3488000	289390	3198600	8740	17220	106670	156760	949600	1913000	336000
2	1962	185771000	3752200	301510	3450700	8530	17550	110860	164570	994300	2089600	366800
3	1963	188483000	4109500	316970	3792500	8640	17650	116470	174210	1086400	2297800	408300
4	1964	191141000	4564600	364220	4200400	9360	21420	130390	203050	1213200	2514400	472800

Step 4. What is the type of the columns?

```
crime.dtypes
```

```
Year                  int64
Population            int64
Total                 int64
Violent               int64
Property              int64
Murder                int64
Forcible_Rape         int64
Robbery               int64
Aggravated_assault    int64
Burglary              int64
Larceny_Theft         int64
Vehicle_Theft         int64
dtype: object
```

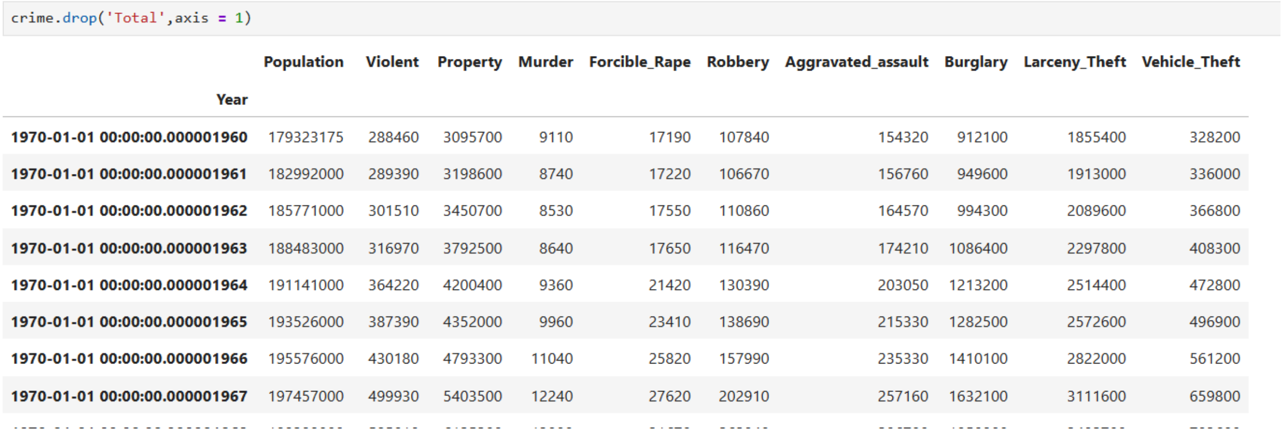
Step 5. Convert the type of the column Year to datetime64

```
crime['Year']= pd.to_datetime(crime['Year'])
```

Step 6. Set the Year column as the index of the dataframe

```
crime.set_index('Year',inplace = True)
```

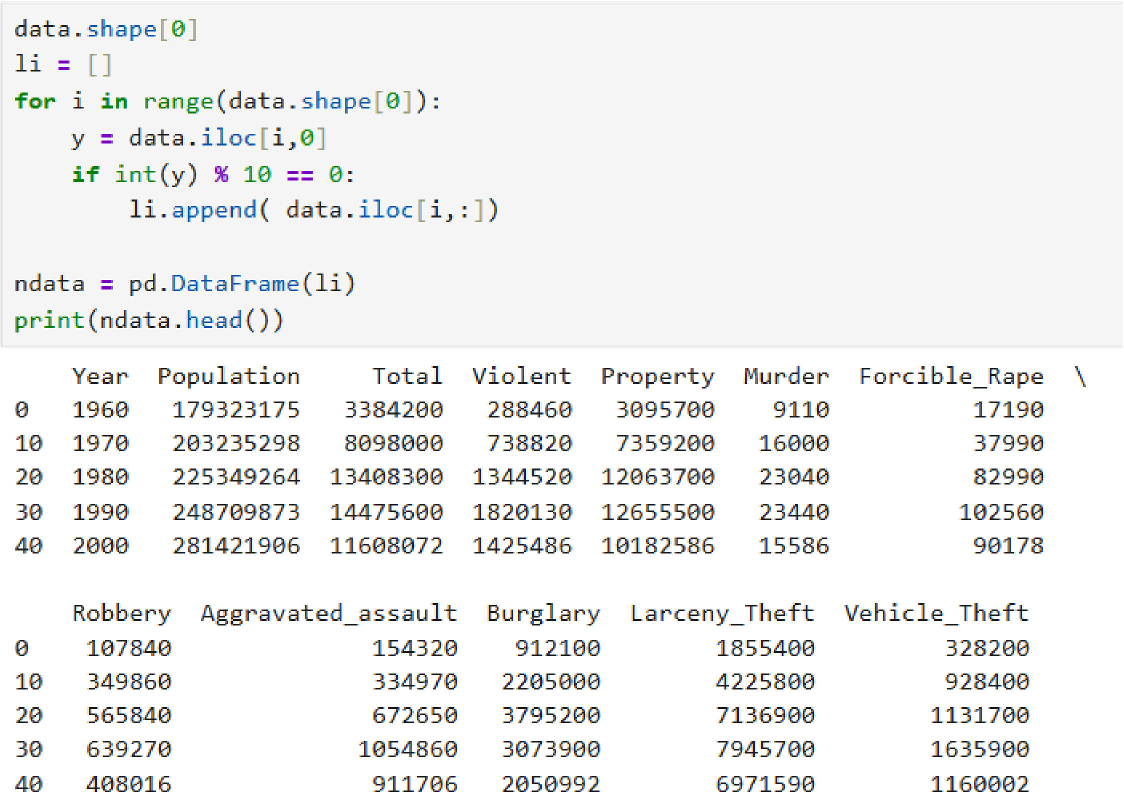
Step 7. Delete the Total column



```python
crime.drop('Total',axis = 1)
```

Year	Population	Violent	Property	Murder	Forcible_Rape	Robbery	Aggravated_assault	Burglary	Larceny_Theft	Vehicle_Theft
1970-01-01 00:00:00.000001960	179323175	288460	3095700	9110	17190	107840	154320	912100	1855400	328200
1970-01-01 00:00:00.000001961	182992000	289390	3198600	8740	17220	106670	156760	949600	1913000	336000
1970-01-01 00:00:00.000001962	185771000	301510	3450700	8530	17550	110860	164570	994300	2089600	366800
1970-01-01 00:00:00.000001963	188483000	316970	3792500	8640	17650	116470	174210	1086400	2297800	408300
1970-01-01 00:00:00.000001964	191141000	364220	4200400	9360	21420	130390	203050	1213200	2514400	472800
1970-01-01 00:00:00.000001965	193526000	387390	4352000	9960	23410	138690	215330	1282500	2572600	496900
1970-01-01 00:00:00.000001966	195576000	430180	4793300	11040	25820	157990	235330	1410100	2822000	561200
1970-01-01 00:00:00.000001967	197457000	499930	5403500	12240	27620	202910	257160	1632100	3111600	659800

Step 8. Group the year by decades and sum the values

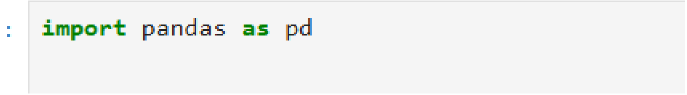


```python
data.shape[0]
li = []
for i in range(data.shape[0]):
    y = data.iloc[i,0]
    if int(y) % 10 == 0:
        li.append( data.iloc[i,:])

ndata = pd.DataFrame(li)
print(ndata.head())
```

```
    Year  Population     Total  Violent  Property  Murder  Forcible_Rape  \
0   1960   179323175   3384200   288460   3095700    9110          17190
10  1970   203235298   8098000   738820   7359200   16000          37990
20  1980   225349264  13408300  1344520  12063700   23040          82990
30  1990   248709873  14475600  1820130  12655500   23440         102560
40  2000   281421906  11608072  1425486  10182586   15586          90178

    Robbery  Aggravated_assault  Burglary  Larceny_Theft  Vehicle_Theft
0    107840              154320    912100        1855400         328200
10   349860              334970   2205000        4225800         928400
20   565840              672650   3795200        7136900        1131700
30   639270             1054860   3073900        7945700        1635900
40   408016              911706   2050992        6971590        1160002
```

- **Exercise on TITANIC Dataset**

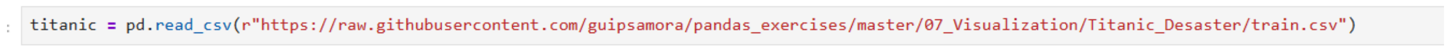
Step 1. Import the necessary libraries



:
```python
import pandas as pd
```

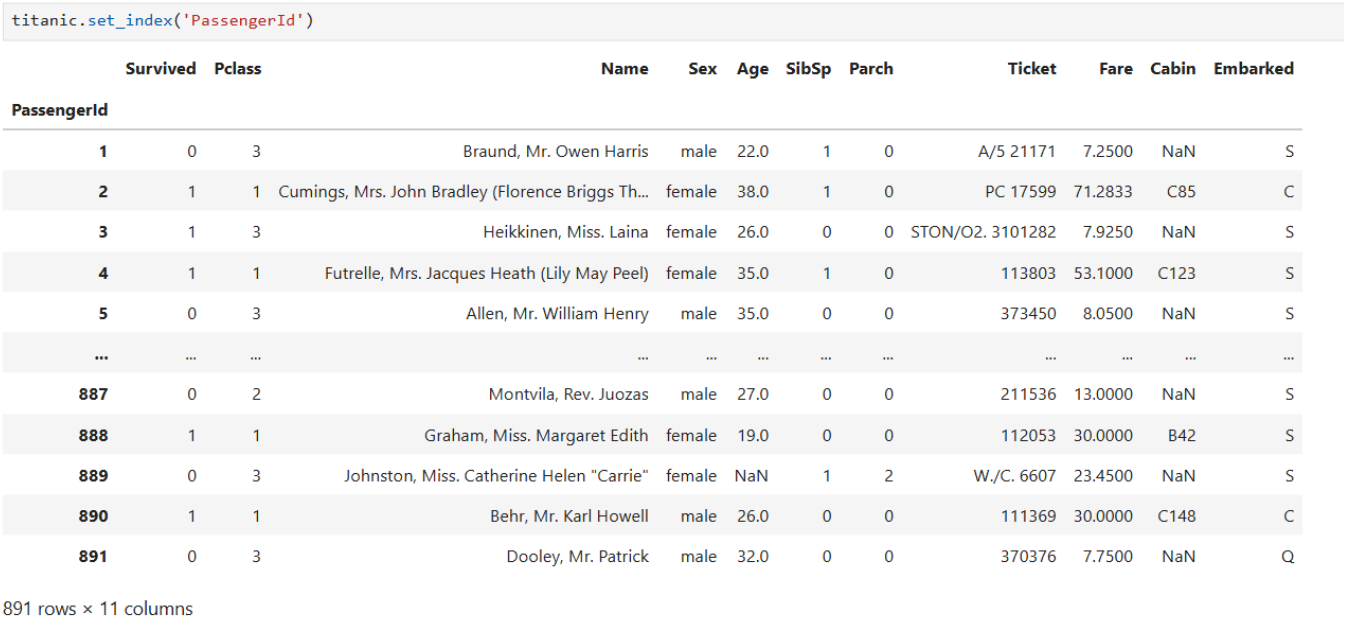
Step 2. Import the dataset from this [address] (https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/07_Visualization/Titanic_Desaster/train.csv)

Step 3. Assign it to a variable titanic



```
titanic = pd.read_csv(r"https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/07_Visualization/Titanic_Desaster/train.csv")
```

Step 4. Set PassengerId as the index

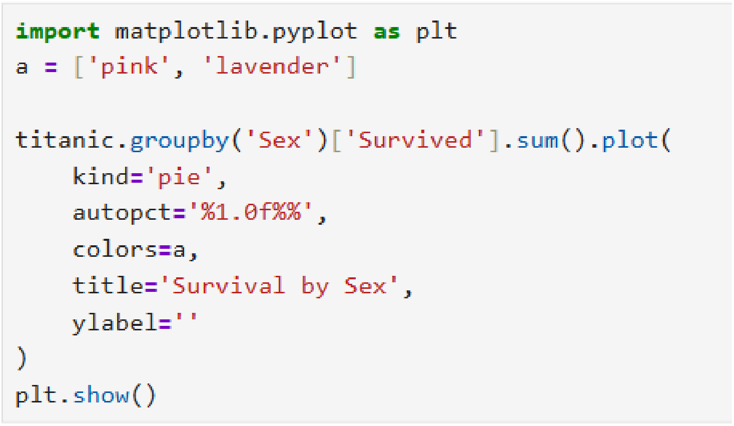


```
titanic.set_index('PassengerId')
```

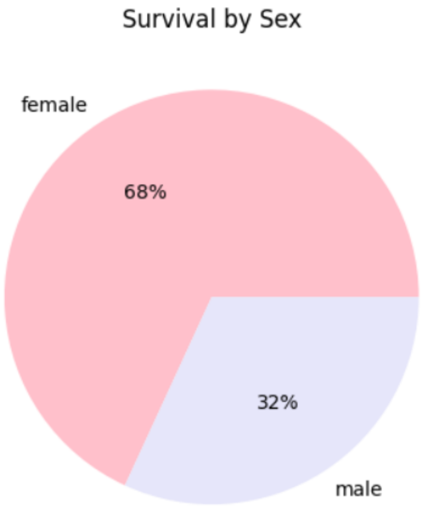
	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
PassengerId											
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 11 columns

Step 5. Create a pie chart presenting the male/female proportion



```
import matplotlib.pyplot as plt
a = ['pink', 'lavender']

titanic.groupby('Sex')['Survived'].sum().plot(
    kind='pie',
    autopct='%1.0f%%',
    colors=a,
    title='Survival by Sex',
    ylabel=''
)
plt.show()
```


Survival by Sex
female
male