Exploratory data analysis with Python

Python Pandas

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv("Mall_Customers.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [4]:
```
s1 = pd.Series([1,2,3,4,5])
s1
```

Out[4]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [5]:
```
type(s1)
```

Out[5]:
```
pandas.core.series.Series
```

In [6]:
```
s1 = pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])
```

In [7]:
```
s1
```

Out[7]:
```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

In [8]:
```
s1 = pd.Series({'a':10,'b':20,'c':30})
```

In [9]:
```
s1
```

Out[9]:
```
a    10
b    20
c    30
dtype: int64
```

In [10]:
```
s1 = pd.Series({'a':10,'b':20,'c':30}, index=['b','c','d','a'])
```

In [11]:
```
s1
```

Out[11]:
```
b    20.0
c    30.0
d     NaN
a    10.0
dtype: float64
```

Extracting individual elements

In [12]:
```
s1 = pd.Series([1,2,3,4,5,6,7,8,9])
s1[3]
```

Out[12]:
```
4
```

In [13]:
```
s1[:4]
```

Out[13]:
```
0    1
1    2
2    3
3    4
dtype: int64
```

In [14]:
```
s1[-3:]
```

Out[14]:
```
6    7
7    8
8    9
dtype: int64
```

Basic Math Operations on Series

In [15]:
```
s1+5
```

Out[15]:
```
0     6
1     7
2     8
3     9
4    10
5    11
6    12
7    13
8    14
dtype: int64
```

In [16]:
```
# Adding two series objects

s2 = pd.Series([10,20,30,40,50,60,70,80,90])
s1+s2
```

Out[16]:
```
0    11
1    22
2    33
3    44
4    55
5    66
6    77
7    88
8    99
dtype: int64
```

Creating Dataframe

In [17]:
```
import pandas as pd
pd.DataFrame({"Name":['Bob','Sam','Anne'],"Marks":[76,89,97]})
```

Out[17]:

	Name	Marks
0	Bob	76
1	Sam	89
2	Anne	97

In [18]:
```
iris = pd.read_csv('iris.csv')
```

In [19]:
```
iris.head()
```

Out[19]:

	Sales	Profit	Price	Month	Species
0	15	72	121	January	Low
1	41	33	597	February	Low
2	85	68	498	March	Low
3	74	76	925	April	Low
4	55	59	315	May	Low

In [20]:
```
iris.tail()
```

Out[20]:

	Sales	Profit	Price	Month	Species
9	74	15	898	October	Medium
10	73	94	890	November	High
11	43	50	676	December	High
12	43	89	188	January	High
13	51	97	140	February	High

In [21]:
```
iris.shape
```

Out[21]:
```
(14, 5)
```

In [22]:
```
iris.describe()
```

Out[22]:

	Sales	Profit	Price
count	14.000000	14.000000	14.000000
mean	50.142857	61.857143	542.642857
std	21.468607	25.746866	313.939633
min	15.000000	15.000000	116.000000
25%	32.750000	38.750000	219.750000
50%	47.000000	63.500000	585.500000
75%	70.750000	83.500000	835.750000
max	85.000000	97.000000	925.000000

.iloc[] function

In [23]:
```
iris.iloc[0:3,0:2]
```

Out[23]:

	Sales	Profit
0	15	72
1	41	33
2	85	68

In [24]:
```
# .loc[] function

iris.loc[(5,10),("Sales","Month")]
```

Out[24]:

	Sales	Month
5	64	June
10	73	November

In [25]:
```
iris.loc[3:9,("Profit","Price")]
```

Out[25]:

	Profit	Price
3	76	925
4	59	315
5	34	842
6	58	116
7	35	817
8	86	574
9	15	898

In [26]:
```
# Dropping Columns

iris.drop('Profit',axis=1)
```

Out[26]:

	Sales	Price	Month	Species
0	15	121	January	Low
1	41	597	February	Low
2	85	498	March	Low
3	74	925	April	Low
4	55	315	May	Low
5	64	842	June	Medium
6	26	116	July	Medium
7	28	817	August	Medium
8	30	574	September	Medium
9	74	898	October	Medium
10	73	890	November	High
11	43	676	December	High
12	43	188	January	High
13	51	140	February	High

In [27]:
```
# Dropping Rows

iris.drop([2,4,7,9],axis=0)
```

Out[27]:

	Sales	Profit	Price	Month	Species
0	15	72	121	January	Low
1	41	33	597	February	Low
3	74	76	925	April	Low
5	64	34	842	June	Medium
6	26	58	116	July	Medium
8	30	86	574	September	Medium
10	73	94	890	November	High
11	43	50	676	December	High
12	43	89	188	January	High
13	51	97	140	February	High

Mean, median, maximum, ,minimum functions

In [28]:
```
iris.mean()
```

```
C:\Users\MOHD. RAEES\AppData\Local\Temp\ipykernel_16992\935066809.py:1: FutureWarning: Dropping of nuisance columns in
DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select o
nly valid columns before calling the reduction.
  iris.mean()
```

Out[28]:
```
Sales      50.142857
Profit     61.857143
Price     542.642857
dtype: float64
```

In [29]:
```
iris.min()
```

Out[29]:
```
Sales         15
Profit        15
Price        116
Month      April
Species     High
dtype: object
```

In [30]:
```
iris.median()
```

```
C:\Users\MOHD. RAEES\AppData\Local\Temp\ipykernel_16992\1297003277.py:1: FutureWarning: Dropping of nuisance columns in
DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select o
nly valid columns before calling the reduction.
  iris.median()
```

Out[30]:
```
Sales      47.0
Profit     63.5
Price     585.5
dtype: float64
```

In [31]:
```
iris.max()
```

Out[31]:
```
Sales            85
Profit           97
Price           925
Month      September
Species       Medium
dtype: object
```

More pandas function

In [32]:
```
def half(s):
    return s*0.5

iris[['Sales','Profit','Price']].apply(half)
```

Out[32]:

	Sales	Profit	Price
0	7.5	36.0	60.5
1	20.5	16.5	298.5
2	42.5	34.0	249.0
3	37.0	38.0	462.5
4	27.5	29.5	157.5
5	32.0	17.0	421.0
6	13.0	29.0	58.0
7	14.0	17.5	408.5
8	15.0	43.0	287.0
9	37.0	7.5	449.0
10	36.5	47.0	445.0
11	21.5	25.0	338.0
12	21.5	44.5	94.0
13	25.5	48.5	70.0

In [33]:
```
def double(s):
    return s*2

iris[['Sales','Profit','Price']].apply(double)
```

Out[33]:

	Sales	Profit	Price
0	30	144	242
1	82	66	1194
2	170	136	996
3	148	152	1850
4	110	118	630
5	128	68	1684
6	52	116	232
7	56	70	1634
8	60	172	1148
9	148	30	1796
10	146	188	1780
11	86	100	1352
12	86	178	376
13	102	194	280

value_counts() function

In [34]:
```
iris['Species'].value_counts()
```

Out[34]:
```
Low       5
Medium    5
High      4
Name: Species, dtype: int64
```

In [35]:
```
iris.sort_values(by='Profit')
```

Out[35]:

	Sales	Profit	Price	Month	Species
9	74	15	898	October	Medium
1	41	33	597	February	Low
5	64	34	842	June	Medium
7	28	35	817	August	Medium
11	43	50	676	December	High
6	26	58	116	July	Medium
4	55	59	315	May	Low
2	85	68	498	March	Low
0	15	72	121	January	Low
3	74	76	925	April	Low
8	30	86	574	September	Medium
12	43	89	188	January	High
10	73	94	890	November	High
13	51	97	140	February	High

In [ ]:
```

```

Analyzed by

Md Raiesh, Enrollment number : **19UME116**, Registration number : **1911345**, B Tech,**7th** semester,Section : **A**, Mechanical Engineering Department, National Institute of Technology Agartala, Tripura 799046,