EX NO 2

February 19, 2024

URK22AI1085 08/01/2024

WORKING WITH DATA

AIM:To perform basic operations in data sets using pandas.

[ ]:
```
Description:
    Pandas is a Python library designed for data manipulation and analysis. It␣
 ↪offers easy-to-use data structures, primarily Series and
DataFrame, for handling structuredcdata. Pandas enables tasks like data␣
 ↪cleaning, exploration, and transformation, making it a powerful tool
for data analysis and manipulation.It serves as a powerful tool for data␣
 ↪manipulation and analysis. At its core are two primary data structures:
Series, a one-dimensional array with labeled data, and DataFrame, a␣
 ↪two-dimensional table akin to a spreadsheet.
    Pandas simplifies data exploration with functions like head(), tail(), and␣
 ↪describe(), providing insights into data structure and content.
    Its capabilities extend to data cleaning, including methods for handling␣
 ↪missing values (fillna()), removing duplicates (drop_duplicates()),
and reshaping data.The library supports various data input/output formats,␣
 ↪facilitating seamless data handling from sources like CSV, Excel,
and SQL. Additionally, Pandas excels in indexing, selection, and time series␣
 ↪analysis, making it an indispensable tool for professionals engaged
in data science, analysis, and manipulation tasks.
```

[14]:
```
import pandas as pd
df= pd.read_csv("Toyota.csv")
```

[15]:
```
max_km = df['KM'].max()
min_weight = df['Weight'].min()
print("maximum kilometer is :",max_km)
print("minimum weight is :" ,min_weight)
```

```
maximum kilometer is : 243000
minimum weight is : 1015
```

[ ]:
```

```

[16]:
```
mean_cc = df['CC'].mean()
median_cc = df['CC'].median()
print("mean:",mean_cc)
print("median:",median_cc)
```

```
mean: 1610.3587174348697
median: 1600.0
```

[ ]:

[17]:
```
df.loc[df['CC'] == 2000, 'Horse Power'] = df['HP'] * 2
```

[ ]:

[18]:
```
filtered_df = df[df['MetColor'] == 1]
print("filtered dataframe : " )
print(filtered_df)
```

```
filtered dataframe :
     Price  Age     KM FuelType   HP  MetColor  Automatic    CC  Doors  \
0    13500   23  46986   Diesel   90         1          0  2000      3
1    13750   23  72937   Diesel   90         1          0  2000      3
2    13950   24  41711   Diesel   90         1          0  2000      3
6    16900   27  94612   Diesel   90         1          0  2000      3
7    18600   30  75889   Diesel   90         1          0  2000      3
..     …   …     …       …  …        …         …   …     …
492   9799   51  59000   Petrol   97         1          0  1400      3
494  11950   54  58745   Petrol  110         1          0  1600      4
495  11250   52  58596   Petrol  110         1          0  1600      3
497  10950   55  58377   Petrol  110         1          0  1600      3
498  11250   56  58142   Petrol  110         1          0  1600      5

     Weight  Horse Power
0      1165        180.0
1      1165        180.0
2      1165        180.0
6      1245        180.0
7      1245        180.0
..      …           …
492    1025          NaN
494    1035          NaN
495    1045          NaN
497    1050          NaN
498    1080          NaN

[372 rows x 11 columns]
```

[ ]:

[19]:
```
result_df = df[['Price', 'Age', 'CC']].sort_values(by='Age').head(50)
result_df
```

[19]:

	Price	Age	CC
185	18245	1	1600
184	17795	1	1400
183	21500	2	1600
182	21125	2	1400
110	31000	4	2000
111	31275	4	2000
109	32500	4	2000
179	22500	6	1600
177	19950	7	1600
114	22950	7	2000
117	17900	7	1600
181	18700	7	1400
180	18500	7	1600
113	24950	8	2000
115	24990	8	2000
116	21950	8	2000
172	19500	8	1400
112	24950	8	2000
173	18950	8	1400
174	21950	8	1600
175	19950	8	1600
176	18950	8	1600
178	21950	8	1600
171	23750	8	1600
162	19600	9	1600
170	18245	9	1600
169	17795	9	1400
152	18450	10	1400
157	18900	11	1600
98	18750	11	1600
138	23000	11	2000
164	17650	11	1400
104	19450	11	1600
103	18500	11	1600
168	20500	12	1600
153	19500	12	1600
142	19950	13	1600
137	16250	13	1600
133	15950	13	1600
129	15850	13	1600
120	18950	13	1600
102	18500	13	1400
147	24500	13	1600

```
154  21750   13  1600
122  16350   14  1600
106  18800   14  1600
166  19950   14  1600
149  20950   14  1600
165  19950   14  1600
163  19500   14  1600
```

[ ]:

[20]:
```
df['CC'].fillna(25, inplace=True)
```

[ ]:

[21]:
```
new_rows = pd.DataFrame({'Column1': [10000], 'Column2': [6666]})
df = pd.concat([df, new_rows])
df
```

[21]:
```
       Price   Age       KM FuelType     HP  MetColor  Automatic      CC  \
0    13500.0  23.0  46986.0   Diesel   90.0       1.0        0.0  2000.0
1    13750.0  23.0  72937.0   Diesel   90.0       1.0        0.0  2000.0
2    13950.0  24.0  41711.0   Diesel   90.0       1.0        0.0  2000.0
3    14950.0  26.0  48000.0   Diesel   90.0       0.0        0.0  2000.0
4    13750.0  30.0  38500.0   Diesel   90.0       0.0        0.0  2000.0
..       ...   ...      ...      ...    ...       ...        ...     ...
495  11250.0  52.0  58596.0   Petrol  110.0       1.0        0.0  1600.0
496  11750.0  54.0  58530.0   Petrol  110.0       0.0        0.0  1600.0
497  10950.0  55.0  58377.0   Petrol  110.0       1.0        0.0  1600.0
498  11250.0  56.0  58142.0   Petrol  110.0       1.0        0.0  1600.0
0        NaN   NaN      NaN      NaN    NaN       NaN        NaN     NaN

     Doors  Weight  Horse Power  Column1  Column2
0      3.0  1165.0        180.0      NaN      NaN
1      3.0  1165.0        180.0      NaN      NaN
2      3.0  1165.0        180.0      NaN      NaN
3      3.0  1165.0        180.0      NaN      NaN
4      3.0  1170.0        180.0      NaN      NaN
..     ...     ...          ...      ...      ...
495    3.0  1045.0          NaN      NaN      NaN
496    5.0  1075.0          NaN      NaN      NaN
497    3.0  1050.0          NaN      NaN      NaN
498    5.0  1080.0          NaN      NaN      NaN
0      NaN     NaN          NaN  10000.0   6666.0

[500 rows x 13 columns]
```

[ ]:

[22]:
```
df.drop('HP', axis=1, inplace=True)
df
```

[22]:
```
       Price   Age       KM FuelType  MetColor  Automatic      CC  Doors  \
0    13500.0  23.0  46986.0   Diesel       1.0        0.0  2000.0    3.0
1    13750.0  23.0  72937.0   Diesel       1.0        0.0  2000.0    3.0
2    13950.0  24.0  41711.0   Diesel       1.0        0.0  2000.0    3.0
3    14950.0  26.0  48000.0   Diesel       0.0        0.0  2000.0    3.0
4    13750.0  30.0  38500.0   Diesel       0.0        0.0  2000.0    3.0
..       …   …       …      …          …        …     …     …
495  11250.0  52.0  58596.0   Petrol       1.0        0.0  1600.0    3.0
496  11750.0  54.0  58530.0   Petrol       0.0        0.0  1600.0    5.0
497  10950.0  55.0  58377.0   Petrol       1.0        0.0  1600.0    3.0
498  11250.0  56.0  58142.0   Petrol       1.0        0.0  1600.0    5.0
0        NaN   NaN      NaN      NaN       NaN        NaN     NaN    NaN

     Weight  Horse Power  Column1  Column2
0    1165.0        180.0      NaN      NaN
1    1165.0        180.0      NaN      NaN
2    1165.0        180.0      NaN      NaN
3    1165.0        180.0      NaN      NaN
4    1170.0        180.0      NaN      NaN
..      …           …      …      …
495  1045.0          NaN      NaN      NaN
496  1075.0          NaN      NaN      NaN
497  1050.0          NaN      NaN      NaN
498  1080.0          NaN      NaN      NaN
0       NaN          NaN  10000.0   6666.0

[500 rows x 12 columns]
```

[ ]:
```

```

[23]:
```
df.drop([49, 99], inplace=True)
```

[ ]:
```

```

[24]:
```
selected_data = df.loc[df['CC'] > 1600, ['Price', 'KM']].head(100)
print(selected_data)
```

```
       Price       KM
0    13500.0  46986.0
1    13750.0  72937.0
2    13950.0  41711.0
3    14950.0  48000.0
4    13750.0  38500.0
..       …       …
458   8695.0  70440.0
```

```
463   8750.0  69000.0
465  11450.0  68520.0
480  11500.0  63000.0
487   8950.0  61000.0

[91 rows x 2 columns]
```

[25]:
```
selected_value_loc = df.loc[5, 'Price']
selected_value_iloc = df.iloc[5, 2]
print(selected_value_loc)
print(selected_value_iloc)
```

```
12950.0
61000.0
```

[26]:
```
ndex_values = df.index
columns = df.columns
data_description = df.describe()
data_size = df.size
data_dimensions = df.shape
data_info = df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 498 entries, 0 to 0
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Price        497 non-null    float64
 1   Age          497 non-null    float64
 2   KM           497 non-null    float64
 3   FuelType     497 non-null    object
 4   MetColor     497 non-null    float64
 5   Automatic    497 non-null    float64
 6   CC           497 non-null    float64
 7   Doors        497 non-null    float64
 8   Weight       497 non-null    float64
 9   Horse Power  46 non-null     float64
 10  Column1      1 non-null      float64
 11  Column2      1 non-null      float64
dtypes: float64(11), object(1)
memory usage: 66.7+ KB
```