chap2-pandas-dataFrame

June 16, 2022

[27]:
```
# import pandas library
import pandas as pd

#import numpy library
import numpy as np

# create empty DataFrame
df = pd.DataFrame()
```

[8]:
```
df = pd.read_csv("WHO_first9cols.csv")
```

[9]:
```
# describe the dataset
df.describe()
```

[9]:
```
        CountryID   Continent  Adolescent fertility rate (%)  \
count  202.000000  202.000000                     177.000000
mean   101.500000    3.579208                      59.457627
std     58.456537    1.808263                      49.105286
min      1.000000    1.000000                       0.000000
25%     51.250000    2.000000                      19.000000
50%    101.500000    3.000000                      46.000000
75%    151.750000    5.000000                      91.000000
max    202.000000    7.000000                     199.000000

       Adult literacy rate (%)  \
count               131.000000
mean                 78.871756
std                  20.415760
min                  23.600000
25%                  68.400000
50%                  86.500000
75%                  95.300000
max                  99.800000

       Gross national income per capita (PPP international $)  \
count                                          178.000000
mean                                         11250.112360
```

```
std                                          12586.753417
min                                            260.000000
25%                                           2112.500000
50%                                           6175.000000
75%                                          14502.500000
max                                          60870.000000

       Net primary school enrolment ratio female (%)  \
count                                       179.000000
mean                                         84.033520
std                                          17.788047
min                                           6.000000
25%                                          79.000000
50%                                          90.000000
75%                                          96.000000
max                                         100.000000

       Net primary school enrolment ratio male (%)  \
count                                     179.000000
mean                                       85.698324
std                                        15.451212
min                                        11.000000
25%                                        79.500000
50%                                        90.000000
75%                                        96.000000
max                                       100.000000

       Population (in thousands) total
count                     1.890000e+02
mean                      3.409964e+04
std                       1.318377e+05
min                       2.000000e+00
25%                       1.328000e+03
50%                       6.640000e+03
75%                       2.097100e+04
max                       1.328474e+06
```

[10]:
```
# count number of observation
df.count()
```

[10]:
```
Country                                                    202
CountryID                                                  202
Continent                                                  202
Adolescent fertility rate (%)                              177
Adult literacy rate (%)                                    131
Gross national income per capita (PPP international $)     178
Net primary school enrolment ratio female (%)              179
```

```
Net primary school enrolment ratio male (%)              179
Population (in thousands) total                          189
dtype: int64
```

[11]:
```
# compute median of all the columns
df.median()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_2656\2465992936.py:2: FutureWarning:
Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only valid
columns before calling the reduction.
  df.median()
```

[11]:
```
CountryID                                                101.5
Continent                                                  3.0
Adolescent fertility rate (%)                             46.0
Adult literacy rate (%)                                   86.5
Gross national income per capita (PPP international $)  6175.0
Net primary school enrolment ratio female (%)             90.0
Net primary school enrolment ratio male (%)               90.0
Population (in thousands) total                         6640.0
dtype: float64
```

[12]:
```
# compute the standard deviation of all the columns
df.std()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_2656\3005725502.py:2: FutureWarning:
Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only valid
columns before calling the reduction.
  df.std()
```

[12]:
```
CountryID                                               58.456537
Continent                                                1.808263
Adolescent fertility rate (%)                           49.105286
Adult literacy rate (%)                                 20.415760
Gross national income per capita (PPP international $)  12586.753417
Net primary school enrolment ratio female (%)           17.788047
Net primary school enrolment ratio male (%)             15.451212
Population (in thousands) total                        131837.708677
dtype: float64
```

[13]:
```
# group by DataFrame on the basis of Continent column
df.groupby('Continent').mean()
```

[13]:
```
           CountryID  Adolescent fertility rate (%)  Adult literacy rate (%)  \
Continent
```

```
1          110.238095                    37.300000                76.900000
2          100.333333                    20.500000                97.911538
3           99.354167                   111.644444                61.690476
4           56.285714                    49.600000                91.600000
5           94.774194                    77.888889                87.940909
6          121.228571                    39.260870                87.607143
7           80.777778                    57.333333                69.812500

           Gross national income per capita (PPP international $)  \
Continent
1                                           14893.529412
2                                           19777.083333
3                                            3050.434783
4                                           24524.000000
5                                            7397.142857
6                                           12167.200000
7                                            2865.555556

           Net primary school enrolment ratio female (%)  \
Continent
1                                            85.789474
2                                            92.911111
3                                            67.574468
4                                            95.000000
5                                            89.137931
6                                            89.040000
7                                            85.444444

           Net primary school enrolment ratio male (%)  \
Continent
1                                          88.315789
2                                          93.088889
3                                          72.021277
4                                          94.400000
5                                          88.517241
6                                          89.960000
7                                          88.888889

           Population (in thousands) total
Continent
1                           16843.350000
2                           17259.627451
3                           16503.195652
4                           73577.333333
5                           15637.241379
6                           25517.142857
7                          317683.666667
```

[14]:
```
# group By DataFrame on the basis of continent and select
# adult literacy rate (%)
df.groupby('Continent').mean()['Adult literacy rate (%)']
```

[14]:
```
Continent
1    76.900000
2    97.911538
3    61.690476
4    91.600000
5    87.940909
6    87.607143
7    69.812500
Name: Adult literacy rate (%), dtype: float64
```

[15]:
```
# load data using read_csv()
dest = pd.read_csv("dest.csv")

# show DataFrame
dest.head()
```

[15]:

	EmpNr	Dest
0	5	The Hague
1	3	Amsterdam
2	9	Rotterdam

[16]:
```
# load data using read_csv()
tips = pd.read_csv("tips.csv")

# show DataFrame
tips.head()
```

[16]:

	EmpNr	Amount
0	5	10.0
1	9	5.0
2	7	2.5

[18]:
```
# join DataFrame using Inner Join
df_inner = pd.merge(dest, tips, on='EmpNr', how='inner')
df_inner.head()
```

[18]:

	EmpNr	Dest	Amount
0	5	The Hague	10.0
1	9	Rotterdam	5.0

[19]:
```
# join DataFrames using Outer Join
df_outer = pd.merge(dest, tips, on='EmpNr', how='outer')
df_outer.head()
```

[19]:
```
   EmpNr       Dest  Amount
0      5  The Hague    10.0
1      3  Amsterdam     NaN
2      9  Rotterdam     5.0
3      7        NaN     2.5
```

[20]:
```
# join DataFrame using Right Outer Join
df_right = pd.merge(dest, tips, on='EmpNr', how='right')
df_right.head()
```

[20]:
```
   EmpNr       Dest  Amount
0      5  The Hague    10.0
1      9  Rotterdam     5.0
2      7        NaN     2.5
```

[21]:
```
# join DataFrames using Left Outer Join
df_left = pd.merge(dest, tips, on='EmpNr', how='left')
df_left.head()
```

[21]:
```
   EmpNr       Dest  Amount
0      5  The Hague    10.0
1      3  Amsterdam     NaN
2      9  Rotterdam     5.0
```

[22]:
```
# count missing values in DataFrame
df.isnull().sum()
```

[22]:
```
Country                                                 0
CountryID                                               0
Continent                                               0
Adolescent fertility rate (%)                          25
Adult literacy rate (%)                                71
Gross national income per capita (PPP international $)    24
Net primary school enrolment ratio female (%)          23
Net primary school enrolment ratio male (%)            23
Population (in thousands) total                        13
dtype: int64
```

[23]:
```
# drop all the missing values
df.dropna(inplace=True)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 118 entries, 1 to 200
Data columns (total 9 columns):
 #   Column                                              Non-Null Count
Dtype
```

```
---  ------                                                  --------------
-----
 0   Country                                                 118 non-null
object
 1   CountryID                                               118 non-null
int64
 2   Continent                                               118 non-null
int64
 3   Adolescent fertility rate (%)                           118 non-null
float64
 4   Adult literacy rate (%)                                 118 non-null
float64
 5   Gross national income per capita (PPP international $)  118 non-null
float64
 6   Net primary school enrolment ratio female (%)           118 non-null
float64
 7   Net primary school enrolment ratio male (%)             118 non-null
float64
 8   Population (in thousands) total                         118 non-null
float64
dtypes: float64(6), int64(2), object(1)
memory usage: 9.2+ KB
```

[24]:
```
# fill missing values with 0
df.fillna(0, inplace=True)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 118 entries, 1 to 200
Data columns (total 9 columns):
 #   Column                                                  Non-Null Count
Dtype
---  ------                                                  --------------
-----
 0   Country                                                 118 non-null
object
 1   CountryID                                               118 non-null
int64
 2   Continent                                               118 non-null
int64
 3   Adolescent fertility rate (%)                           118 non-null
float64
 4   Adult literacy rate (%)                                 118 non-null
float64
 5   Gross national income per capita (PPP international $)  118 non-null
float64
 6   Net primary school enrolment ratio female (%)           118 non-null
```

```
float64
 7   Net primary school enrolment ratio male (%)             118 non-null
float64
 8   Population (in thousands) total                         118 non-null
float64
dtypes: float64(6), int64(2), object(1)
memory usage: 9.2+ KB
```

[25]:
```
# creating pivot tables

# load data using read_csv()
purchase = pd.read_csv("purchase.csv")

# show initial 10 records
purchase.head(10)
```

[25]:

	Weather	Food	Price	Number
0	cold	soup	3.745401	8
1	hot	soup	9.507143	8
2	cold	icecream	7.319939	8
3	hot	chocolate	5.986585	8
4	cold	icecream	1.560186	8
5	hot	icecream	1.559945	8
6	cold	soup	0.580836	8

[28]:
```
# summarise dataframe using pivot table
pd.pivot_table(purchase, values='Number', index=['Weather',],
               columns=['Food'], aggfunc=np.sum)
```

[28]:

Food	chocolate	icecream	soup
Weather			
cold	NaN	16.0	16.0
hot	8.0	8.0	8.0

[30]:
```
pd.date_range('01-01-2000', periods=45, freq='D')
```

[30]:
```
DatetimeIndex(['2000-01-01', '2000-01-02', '2000-01-03', '2000-01-04',
               '2000-01-05', '2000-01-06', '2000-01-07', '2000-01-08',
               '2000-01-09', '2000-01-10', '2000-01-11', '2000-01-12',
               '2000-01-13', '2000-01-14', '2000-01-15', '2000-01-16',
               '2000-01-17', '2000-01-18', '2000-01-19', '2000-01-20',
               '2000-01-21', '2000-01-22', '2000-01-23', '2000-01-24',
               '2000-01-25', '2000-01-26', '2000-01-27', '2000-01-28',
               '2000-01-29', '2000-01-30', '2000-01-31', '2000-02-01',
               '2000-02-02', '2000-02-03', '2000-02-04', '2000-02-05',
               '2000-02-06', '2000-02-07', '2000-02-08', '2000-02-09',
               '2000-02-10', '2000-02-11', '2000-02-12', '2000-02-13',
```

```
               '2000-02-14'],
              dtype='datetime64[ns]', freq='D')
```

[ ]: