Lecture 5

May 19, 2023

1 Lecture 5

1.1 pandas contd.

[1]:
```
import pandas as pd
import numpy as np
```

1.1.1 Dataframe

[2]:
```
#create dataframe from series

s1 = pd.Series(np.random.rand(10))
s2 = pd.Series(np.random.rand(10))

df = pd.DataFrame(s1, s2)
print(df)
```

```
           0
0.670478 NaN
0.617299 NaN
0.950636 NaN
0.151099 NaN
0.703379 NaN
0.276145 NaN
0.385158 NaN
0.579975 NaN
0.598191 NaN
0.006524 NaN
```

[4]:
```
#create dataframe from dictionary

dict = {'age':[20, 22, 21, 25, 24], 'salary':['10000', '12000', '30000',␣
 ↪'40000', '15000']}

df = pd.DataFrame(dict)
print(df)

#add column
```

```
df['Gender'] = ['M', 'F', 'M', 'F', 'M']
print(df)
```

```
   age salary
0   20  10000
1   22  12000
2   21  30000
3   25  40000
4   24  15000
   age salary Gender
0   20  10000      M
1   22  12000      F
2   21  30000      M
3   25  40000      F
4   24  15000      M
```

[5]:
```
# create dataframe from array

arr = [['hi', 1], ['how', 2], ['are', 3], ['you', 4]]
df = pd.DataFrame(arr)

print(df)
```

```
     0  1
0   hi  1
1  how  2
2  are  3
3  you  4
```

[9]:
```
#Changing column names
df.columns = ['A', 'B']
print(df)

#or do while creation
df = pd.DataFrame(arr, columns=['C', 'D'])
print(df)
```

```
     A  B
0   hi  1
1  how  2
2  are  3
3  you  4
     C  D
0   hi  1
1  how  2
2  are  3
3  you  4
```

[13]:
```
#Changing index
df.index = range(1,5)
print(df)

#or do while creation
df = pd.DataFrame(arr, index=range(6,10))
print(df)
```

```
     C  D
1   hi  1
2  how  2
3  are  3
4  you  4
     0  1
6   hi  1
7  how  2
8  are  3
9  you  4
```

[23]:
```
d = {'Name':["Akash", "Vishal", "Ram", "Raj", "Jason"],
     'Age':np.random.randint(20, 30, 5),
     'Salary':[20000, 30000, 45000, 50000, 34000]}

df = pd.DataFrame(d)
print(df)

#Renaming index and columns

#axis => 1 -- columns
#axis => 0 -- rows

#index
df = df.rename({'Name':'name', 'Age':'age'}, axis=1)
print(df)

#columns
df = df.rename({0:'A', 1:'B', 2:'C', 3:'D', 4:'E'}, axis=0)
print(df)
```

```
     Name  Age  Salary
0   Akash   21   20000
1  Vishal   21   30000
2     Ram   29   45000
3     Raj   27   50000
4   Jason   27   34000
     name  age  Salary
0   Akash   21   20000
1  Vishal   21   30000
```

```
2     Ram   29   45000
3     Raj   27   50000
4   Jason   27   34000
     name  age  Salary
A   Akash   21   20000
B  Vishal   21   30000
C     Ram   29   45000
D     Raj   27   50000
E   Jason   27   34000
```

[24]:
```
# loc and iloc

#iloc for integer indexing
print(df.iloc[:2])
```

```
     name  age  Salary
A   Akash   21   20000
B  Vishal   21   30000
```

[26]:
```
print(df.iloc[1:3, 0:2])
```

```
     name  age
B  Vishal   21
C     Ram   29
```

[30]:
```
#loc for label indexing
print(df.loc[:, ['age','Salary']])
```

```
   age  Salary
A   21   20000
B   21   30000
C   29   45000
D   27   50000
E   27   34000
```

[31]:
```
#Dimension

df.shape
```

[31]:
```
(5, 3)
```

[32]:
```
df.columns
```

[32]:
```
Index(['name', 'age', 'Salary'], dtype='object')
```

[33]:
```
df.index
```

[33]:
```
Index(['A', 'B', 'C', 'D', 'E'], dtype='object')
```

[38]:
```
df.dtypes
```

[38]:
```
name      object
age        int64
Salary     int64
dtype: object
```

[39]:
```
df.isnull()
```

[39]:

	name	age	Salary
A	False	False	False
B	False	False	False
C	False	False	False
D	False	False	False
E	False	False	False

[40]:
```
df.isnull().any()
```

[40]:
```
name      False
age       False
Salary    False
dtype: bool
```

[41]:
```
df.isnull().sum()
```

[41]:
```
name      0
age       0
Salary    0
dtype: int64
```

[43]:
```
#dropping columns

df = df.drop(columns=['Salary'], axis=1)
print(df)
```

```
     name  age
A   Akash   21
B  Vishal   21
C     Ram   29
D     Raj   27
E   Jason   27
```

[45]:
```
#dropping rows

df = df.drop(index=['E'], axis=0)
print(df)
```

```
     name  age
A   Akash   21
B  Vishal   21
```

```
C     Ram   29
D     Raj   27
```

[47]:
```
df.duplicated()
```

[47]:
```
A    False
B    False
C    False
D    False
dtype: bool
```

[50]:
```
#import dataframe
df = pd.read_csv("mtcars.csv")
print(df)
```

```
                 model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  \
0            Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1
1        Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1
2           Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1
3       Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0
4    Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0
5              Valiant  18.1    6  225.0  105  2.76  3.460  20.22   1   0
6           Duster 360  14.3    8  360.0  245  3.21  3.570  15.84   0   0
7            Merc 240D  24.4    4  146.7   62  3.69  3.190  20.00   1   0
8             Merc 230  22.8    4  140.8   95  3.92  3.150  22.90   1   0
9             Merc 280  19.2    6  167.6  123  3.92  3.440  18.30   1   0
10            Merc 280C  17.8    6  167.6  123  3.92  3.440  18.90   1   0
11           Merc 450SE  16.4    8  275.8  180  3.07  4.070  17.40   0   0
12           Merc 450SL  17.3    8  275.8  180  3.07  3.730  17.60   0   0
13          Merc 450SLC  15.2    8  275.8  180  3.07  3.780  18.00   0   0
14   Cadillac Fleetwood  10.4    8  472.0  205  2.93  5.250  17.98   0   0
15  Lincoln Continental  10.4    8  460.0  215  3.00  5.424  17.82   0   0
16    Chrysler Imperial  14.7    8  440.0  230  3.23  5.345  17.42   0   0
17             Fiat 128  32.4    4   78.7   66  4.08  2.200  19.47   1   1
18         Honda Civic  30.4    4   75.7   52  4.93  1.615  18.52   1   1
19      Toyota Corolla  33.9    4   71.1   65  4.22  1.835  19.90   1   1
20       Toyota Corona  21.5    4  120.1   97  3.70  2.465  20.01   1   0
21    Dodge Challenger  15.5    8  318.0  150  2.76  3.520  16.87   0   0
22         AMC Javelin  15.2    8  304.0  150  3.15  3.435  17.30   0   0
23          Camaro Z28  13.3    8  350.0  245  3.73  3.840  15.41   0   0
24    Pontiac Firebird  19.2    8  400.0  175  3.08  3.845  17.05   0   0
25           Fiat X1-9  27.3    4   79.0   66  4.08  1.935  18.90   1   1
26       Porsche 914-2  26.0    4  120.3   91  4.43  2.140  16.70   0   1
27        Lotus Europa  30.4    4   95.1  113  3.77  1.513  16.90   1   1
28      Ford Pantera L  15.8    8  351.0  264  4.22  3.170  14.50   0   1
29        Ferrari Dino  19.7    6  145.0  175  3.62  2.770  15.50   0   1
30       Maserati Bora  15.0    8  301.0  335  3.54  3.570  14.60   0   1
31          Volvo 142E  21.4    4  121.0  109  4.11  2.780  18.60   1   1
```

```
    gear  carb
0      4     4
1      4     4
2      4     1
3      3     1
4      3     2
5      3     1
6      3     4
7      4     2
8      4     2
9      4     4
10     4     4
11     3     3
12     3     3
13     3     3
14     3     4
15     3     4
16     3     4
17     4     1
18     4     2
19     4     1
20     3     1
21     3     2
22     3     2
23     3     4
24     3     2
25     4     1
26     5     2
27     5     2
28     5     4
29     5     6
30     5     8
31     4     2
```

[54]:
```
df.isnull().sum()
```

[54]:
```
model    0
mpg      0
cyl      0
disp     0
hp       0
drat     0
wt       0
qsec     0
vs       0
am       0
```

```
gear     0
carb     0
dtype: int64
```

[55]:
```
df.duplicated()
```

[55]:
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29    False
30    False
31    False
dtype: bool
```

[58]:
```
#remove duplicates

df = df.drop_duplicates()
df.duplicated()
```

[58]:
```
0     False
1     False
```

```
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17    False
18    False
19    False
20    False
21    False
22    False
23    False
24    False
25    False
26    False
27    False
28    False
29    False
30    False
31    False
dtype: bool
```

[60]:
```
df.head()
```

[60]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

[62]:
```
df.head(7)
```

[62]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
6	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4

[61]:
```
df.tail()
```

[61]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.5	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.5	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.6	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.6	1	1	4	2

[63]:
```
df.tail(7)
```

[63]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
25	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.9	1	1	4	1
26	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.7	0	1	5	2
27	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2
28	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.5	0	1	5	4
29	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.5	0	1	5	6
30	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.6	0	1	5	8
31	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.6	1	1	4	2

```
    carb
25     1
26     2
27     2
28     4
29     6
30     8
31     2
```

[65]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32 entries, 0 to 31
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   model   32 non-null     object
 1   mpg     32 non-null     float64
 2   cyl     32 non-null     int64
 3   disp    32 non-null     float64
 4   hp      32 non-null     int64
 5   drat    32 non-null     float64
 6   wt      32 non-null     float64
 7   qsec    32 non-null     float64
 8   vs      32 non-null     int64
 9   am      32 non-null     int64
 10  gear    32 non-null     int64
 11  carb    32 non-null     int64
dtypes: float64(5), int64(6), object(1)
memory usage: 3.2+ KB
```

[66]:
```
df.isnull().sum()
```

[66]:
```
model    0
mpg      0
cyl      0
disp     0
hp       0
drat     0
wt       0
qsec     0
vs       0
am       0
gear     0
carb     0
dtype: int64
```

[67]:
```
df.describe()
```

[67]:

	mpg	cyl	disp	hp	drat	wt
count	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000
mean	20.090625	6.187500	230.721875	146.687500	3.596563	3.217250
std	6.026948	1.785922	123.938694	68.562868	0.534679	0.978457
min	10.400000	4.000000	71.100000	52.000000	2.760000	1.513000
25%	15.425000	4.000000	120.825000	96.500000	3.080000	2.581250
50%	19.200000	6.000000	196.300000	123.000000	3.695000	3.325000
75%	22.800000	8.000000	326.000000	180.000000	3.920000	3.610000
max	33.900000	8.000000	472.000000	335.000000	4.930000	5.424000

	qsec	vs	am	gear	carb
count	32.000000	32.000000	32.000000	32.000000	32.0000
mean	17.848750	0.437500	0.406250	3.687500	2.8125
std	1.786943	0.504016	0.498991	0.737804	1.6152
min	14.500000	0.000000	0.000000	3.000000	1.0000
25%	16.892500	0.000000	0.000000	3.000000	2.0000
50%	17.710000	0.000000	0.000000	4.000000	2.0000
75%	18.900000	1.000000	1.000000	4.000000	4.0000
max	22.900000	1.000000	1.000000	5.000000	8.0000