Lecture18

April 4, 2024

0.0.1 pandas

[1]:
```
# create dataframe,
# prepare
# selecting
# filter
```

0.0.2 Missing values

[3]:
```
import pandas as pd
```

[11]:
```
df=pd.read_csv('datafiles/missing.csv', index_col=0)
df
```

[11]:

	one	two	three	four
a	-1.250699	-0.573801	0.705961	-1.015682
b	NaN	-0.217766	0.655179	1.379276
c	-0.860359	-1.313747	0.676174	1.034417
d	NaN	NaN	NaN	NaN
e	0.079169	0.029138	0.239183	-0.492039
f	-1.149060	NaN	NaN	-0.160499

[12]:
```
df.isna()
```

[12]:

	one	two	three	four
a	False	False	False	False
b	True	False	False	False
c	False	False	False	False
d	True	True	True	True
e	False	False	False	False
f	False	True	True	False

[14]:
```
import numpy as np
np.sum(df.isna(), axis=0)
```

[14]:
```
one      2
two      2
three    2
```

```
four     1
dtype: int64
```

[16]:
```
np.sum(df.isna(), axis=1)
```

[16]:
```
a    0
b    1
c    0
d    4
e    0
f    2
dtype: int64
```

[17]:
```
np.sum(np.sum(df.isna(), axis=1))
```

[17]:
```
7
```

[20]:
```
#  select bad data
df.loc[np.sum(df.isna(), axis=1)>0,:]
```

[20]:
```
        one       two     three      four
b       NaN -0.217766  0.655179  1.379276
d       NaN       NaN       NaN       NaN
f -1.14906       NaN       NaN -0.160499
```

[21]:
```
#  select good data
df.loc[np.sum(df.isna(), axis=1)==0,:]
```

[21]:
```
         one       two     three      four
a -1.250699 -0.573801  0.705961 -1.015682
c -0.860359 -1.313747  0.676174  1.034417
e  0.079169  0.029138  0.239183 -0.492039
```

0.0.3 Dealing with missing values

[24]:
```
# delete any row or column that has missing value
df.dropna(axis=0, how='any)
```

[24]:
```
         one       two     three      four
a -1.250699 -0.573801  0.705961 -1.015682
c -0.860359 -1.313747  0.676174  1.034417
e  0.079169  0.029138  0.239183 -0.492039
```

[25]:
```
# replace missing value with certain value
df['one']=df['one'].fillna(df['one'].mean())
```

[25]:

	one	two	three	four
a	-1.250699	-0.573801	0.705961	-1.015682
b	10.000000	-0.217766	0.655179	1.379276
c	-0.860359	-1.313747	0.676174	1.034417
d	10.000000	10.000000	10.000000	10.000000
e	0.079169	0.029138	0.239183	-0.492039
f	-1.149060	10.000000	10.000000	-0.160499

[26]:
```
# use interpolate
df.interpolate()
```

[26]:

	one	two	three	four
a	-1.250699	-0.573801	0.705961	-1.015682
b	-1.055529	-0.217766	0.655179	1.379276
c	-0.860359	-1.313747	0.676174	1.034417
d	-0.390595	-0.642305	0.457679	0.271189
e	0.079169	0.029138	0.239183	-0.492039
f	-1.149060	0.029138	0.239183	-0.160499

[27]:
```
df
```

[27]:

	one	two	three	four
a	-1.250699	-0.573801	0.705961	-1.015682
b	NaN	-0.217766	0.655179	1.379276
c	-0.860359	-1.313747	0.676174	1.034417
d	NaN	NaN	NaN	NaN
e	0.079169	0.029138	0.239183	-0.492039
f	-1.149060	NaN	NaN	-0.160499

[ ]:
```

```