Pandas Basics

Contents

Create a DataFrame from scratch

One way is to use a dictionary, key is the column name, value is column data.

```
data = {
    'col1': [1, 2, 3, 4],
    'col2': [5, 6, 7, 8]
}

sampleDf = pd.DataFrame(data)
sampleDf
```

```
  col1 col2
0   1   5
1   2   6
2   3   7
3   4   8
```

Create a DataFrame with row indexes, giving meaningful row names:

```
row_names =['r1', 'r2', 'r3', 'r4']
sampleDf = pd.DataFrame(data, index=row_names)
```

```
   col1 col2
r1   1   5
r2   2   6
r3   3   7
r4   4   8
```

DataFrame slicing, selecting, extracting

By column

Similar to access dictionary:

```
data['col1'] => [1, 2, 3, 4]

sampleDf['col1']
```

```
   col1  col2
r1    1    5
r2    2    6
r3    3    7
r4    4    8
```

```
type(sampleDf['col1']) => Series
```

Use a list of columns will return a DataFrame:

```
type(sampleDf[['col1', 'col2']])
```

By Row

.loc - locate by name
.iloc- locate by numerical index

```
sampleDf.loc[['r1', 'r2']]
sampleDf.iloc[[0, 1]]

sampleDf.loc['r1':'r3']
sampleDf.iloc[0:3]  # exclude last index

sampleDf.loc[['r1', 'r2'], ['col1']]
```

Conditional selections

```
condition = (sampleDf['col1']=='3')
or
condition = (sampleDf.col1=='3')
```

```
condition
r1   False
r2   False
r3   True
r4   False

sampleDf[condition]
   col1  col2
r3    3    7

or
sampleDf[sampleDf.col1==3]
```

```
sampleDf[sampleDf['col1'].isin([1,3])]
sampleDf[((sampleDf['col1']==1) | (sampleDf['col1']==3))]
sampleDf[sampleDf['col1']>2]


sampleDf['doulble'] = sampleDf['col1'].apply(lambda x: x*2)
```