06-Operations

August 5, 2020

1 Operations

There are lots of operations with pandas that will be really useful to you, but don't fall into any distinct category. Let's show them here in this lecture:

[52]:
```
import pandas as pd
df = pd.DataFrame({'col1':[1,2,3,4],'col2':[444,555,666,444],'col3':
 ↪['abc','def','ghi','xyz']})
df.head()
```

[52]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

1.0.1 Info on Unique Values

[53]:
```
df['col2'].unique()
```

[53]:
```
array([444, 555, 666])
```

[54]:
```
df['col2'].nunique()
```

[54]:
```
3
```

[55]:
```
df['col2'].value_counts()
```

[55]:
```
444    2
555    1
666    1
Name: col2, dtype: int64
```

1.0.2 Selecting Data

[56]:
```
#Select from DataFrame using criteria from multiple columns
newdf = df[(df['col1']>2) & (df['col2']==444)]
```

[57]:
```
newdf
```

[57]:
```
   col1  col2 col3
3     4   444  xyz
```

1.0.3 Applying Functions

[58]:
```
def times2(x):
    return x*2
```

[59]:
```
df['col1'].apply(times2)
```

[59]:
```
0    2
1    4
2    6
3    8
Name: col1, dtype: int64
```

[60]:
```
df['col3'].apply(len)
```

[60]:
```
0    3
1    3
2    3
3    3
Name: col3, dtype: int64
```

[61]:
```
df['col1'].sum()
```

[61]:
```
10
```

** Permanently Removing a Column**

[62]:
```
del df['col1']
```

[63]:
```
df
```

[63]:
```
   col2 col3
0   444  abc
1   555  def
2   666  ghi
3   444  xyz
```

** Get column and index names: **

[64]:
```
df.columns
```

[64]:
```
Index(['col2', 'col3'], dtype='object')
```

[65]:
```
df.index
```

[65]:
```
RangeIndex(start=0, stop=4, step=1)
```

** Sorting and Ordering a DataFrame:**

[66]:
```
df
```

[66]:
```
   col2 col3
0   444  abc
1   555  def
2   666  ghi
3   444  xyz
```

[67]:
```
df.sort_values(by='col2') #inplace=False by default
```

[67]:
```
   col2 col3
0   444  abc
3   444  xyz
1   555  def
2   666  ghi
```

** Find Null Values or Check for Null Values**

[68]:
```
df.isnull()
```

[68]:
```
    col2   col3
0  False  False
1  False  False
2  False  False
3  False  False
```

[69]:
```
# Drop rows with NaN Values
df.dropna()
```

[69]:
```
   col2 col3
0   444  abc
1   555  def
2   666  ghi
3   444  xyz
```

** Filling in NaN values with something else: **

[71]:
```
import numpy as np
```

[72]:
```
df = pd.DataFrame({'col1':[1,2,3,np.nan],
                   'col2':[np.nan,555,666,444],
                   'col3':['abc','def','ghi','xyz']})
df.head()
```

[72]:

	col1	col2	col3
0	1.0	NaN	abc
1	2.0	555.0	def
2	3.0	666.0	ghi
3	NaN	444.0	xyz

[75]:
```
df.fillna('FILL')
```

[75]:

	col1	col2	col3
0	1	FILL	abc
1	2	555	def
2	3	666	ghi
3	FILL	444	xyz

[89]:
```
data = {'A':['foo','foo','foo','bar','bar','bar'],
     'B':['one','one','two','two','one','one'],
       'C':['x','y','x','y','x','y'],
       'D':[1,3,2,5,4,1]}

df = pd.DataFrame(data)
```

[90]:
```
df
```

[90]:

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

[91]:
```
df.pivot_table(values='D',index=['A', 'B'],columns=['C'])
```

[91]:

C		x	y
A	B		
bar	one	4.0	1.0
	two	NaN	5.0
foo	one	1.0	3.0
	two	2.0	NaN