PMDS508L - Python Programming
Jupyter Notebook Demonstrating the Python Pandas Function Applications

Pandas Function Applications - `pipe()`, `apply()`, `map()`, `applymap()`

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

[2]:
```
def adder(ele1,ele2):
    return ele1+ele2

df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
print(df)
print(df.pipe(adder,2))
```

```
       col1      col2      col3
0  0.140167 -0.168923 -0.157747
1 -1.542168 -1.692722 -0.126352
2  0.520204  0.849698  1.923826
3 -2.037577  0.145233  0.862633
4  2.816327  0.154308  0.446446
       col1      col2      col3
0  2.140167  1.831077  1.842253
1  0.457832  0.307278  1.873648
2  2.520204  2.849698  3.923826
3 -0.037577  2.145233  2.862633
4  4.816327  2.154308  2.446446
```

[3]:
```
df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
```

[4]:
```
print(df)
```

```
       col1      col2      col3
0 -1.434929 -1.255962  0.643258
1 -0.298551  0.206143  1.408999
2 -0.523307  0.227859 -0.438078
3  0.147371  1.356881 -1.159660
4 -0.102642 -0.304327  1.211789
```

[5]:
```
df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
df['col1']
# My custom function
df1 = df['col1'].map(lambda x:x*100)
print(df1)
```

```
0    -34.893121
1    -45.427195
2     -7.498417
3    122.804055
4     11.860078
Name: col1, dtype: float64
```

[6]:
```
df['col4'] = df['col1'].map(lambda x: x*100)
```

[7]:
```
df
```

[7]:

	col1	col2	col3	col4
0	-0.348931	0.020714	0.108301	-34.893121
1	-0.454272	-1.614470	-0.225268	-45.427195
2	-0.074984	-1.068262	0.513826	-7.498417
3	1.228041	-0.233468	-0.513058	122.804055
4	0.118601	-1.692430	0.692784	11.860078

[8]:
```
df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
df1 = df.apply(lambda x: x.max() - x.min(), axis=1)
print(df1)
```

```
0    1.238521
1    1.248121
2    1.097562
3    1.031142
4    1.696895
dtype: float64
```

[9]:
```
df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
df1 = df.applymap(lambda x:x*100)
print(df1)
```

```
       col1        col2        col3
0  0.940057 -148.323224  -86.847499
1 69.724947   20.847856  131.302389
2 89.390374  -48.063263  225.486652
3 -2.343274 -131.578749   32.143996
4 17.575315 -114.252663  160.273065
```

[10]:
```
df1=df.apply(lambda x: x*100)
```

[11]:
```
print(df1)
```

```
        col1        col2        col3
0   0.940057 -148.323224  -86.847499
1  69.724947   20.847856  131.302389
2  89.390374  -48.063263  225.486652
3  -2.343274 -131.578749   32.143996
4  17.575315 -114.252663  160.273065
```

[12]:
```
df = pd.DataFrame(np.random.randn(5,3),columns=['col1','col2','col3'])
df1 = df.pipe(np.sqrt)
print(df1)
```

```
       col1  col2      col3
0  1.271423   NaN  0.943315
1  0.675307   NaN  0.456417
2  0.891724   NaN       NaN
3  1.322641   NaN       NaN
4       NaN   NaN       NaN
```

[13]:
```
df.apply(np.sqrt)
```

[13]:
```
        col1  col2      col3
0  1.271423   NaN  0.943315
1  0.675307   NaN  0.456417
2  0.891724   NaN       NaN
3  1.322641   NaN       NaN
4       NaN   NaN       NaN
```

[14]:
```
df.applymap(np.sqrt)
```

```
C:\Users\BSRVPrasad\anaconda3\lib\site-packages\pandas\core\frame.py:8823:
RuntimeWarning: invalid value encountered in sqrt
  return lib.map_infer(x.astype(object)._values, func, ignore_na=ignore_na)
```

[14]:
```
        col1  col2      col3
0  1.271423   NaN  0.943315
1  0.675307   NaN  0.456417
2  0.891724   NaN       NaN
3  1.322641   NaN       NaN
4       NaN   NaN       NaN
```

[15]:
```
print(df)
df.apply(lambda x: x.sum(),axis = 0)
```

```
       col1      col2      col3
0  1.616518 -0.153377  0.889844
1  0.456039 -0.114378  0.208316
2  0.795172 -0.070005 -0.907891
3  1.749378 -0.827901 -1.370153
4 -1.199839 -0.654766 -0.994009
```

[15]:
```
col1    3.417268
col2   -1.820427
col3   -2.173894
dtype: float64
```

[16]:
```
print(df)
df.applymap(lambda x: x+10)
```

```
       col1      col2      col3
0  1.616518 -0.153377  0.889844
1  0.456039 -0.114378  0.208316
2  0.795172 -0.070005 -0.907891
3  1.749378 -0.827901 -1.370153
4 -1.199839 -0.654766 -0.994009
```

[16]:

	col1	col2	col3
0	11.616518	9.846623	10.889844
1	10.456039	9.885622	10.208316
2	10.795172	9.929995	9.092109
3	11.749378	9.172099	8.629847
4	8.800161	9.345234	9.005991

[17]:
```
df.applymap(np.abs).applymap(np.sqrt).applymap(lambda x: x+ 10)
```

[17]:

	col1	col2	col3
0	11.271423	10.391633	10.943315
1	10.675307	10.338199	10.456417
2	10.891724	10.264585	10.952833
3	11.322641	10.909891	11.170536
4	11.095372	10.809176	10.997000