

In [181]:

```
1 df5
```

Out[181]:

	b	c	d	e
Chennai	NaN	NaN	NaN	NaN
Delhi	NaN	NaN	NaN	NaN
Mumbai	3.0	NaN	6.0	NaN
Nagpur	NaN	NaN	NaN	NaN
Pune	9.0	NaN	12.0	NaN

In []:

```
1
```

Day 3 - Pandas

In []:

```
1
```

Functions Applications & Mapping

In [1]:

```
1 import pandas as pd
2 import numpy as np
```

In [41]:

```
1 df1 = pd.DataFrame(np.random.randn(4,3),columns=['b','d','e'],index=['Mumbai','Pune','Nagpur','Thane'])
```

In [42]:

```
1 df1
```

Out[42]:

	b	d	e
Mumbai	1.649082	2.699737	-0.552249
Pune	0.239891	1.301519	0.211753
Nagpur	-1.098035	-0.917801	-1.199462
Thane	-0.599089	-1.935907	0.628735

In []:

```
1
```

In [43]:

```
1 df1.loc['Mumbai'].min()
```

Out[43]:

-0.5522485051350956

In [44]:

```
1 df1.loc['Mumbai'].max()
```

Out[44]:

2.69973691278904

In [45]:

```
1 df1.loc['Mumbai'].max() - df1.loc['Mumbai'].min()
```

Out[45]:

3.2519854179241356

In [53]:

```
1 df1.loc['Pune'].max() - df1.loc['Pune'].min()
```

Out[53]:

1.0897663004761642

In []:

```
1
```

In [46]:

```
1 diff = lambda x : x.max() - x.min()
```

In []:

```
1
```

In [47]:

```
1 res = df1.apply(diff,axis='columns')
```

In [54]:

```
1 res
```

Out[54]:

```
Mumbai    3.251985
Pune      1.089766
Nagpur     0.281660
Thane      2.564642
dtype: float64
```

In [55]:

```
1 df1['difference'] = res
```

In [56]:

```
1 df1
```

Out[56]:

	b	d	e	difference
Mumbai	1.649082	2.699737	-0.552249	3.251985
Pune	0.239891	1.301519	0.211753	1.089766
Nagpur	-1.098035	-0.917801	-1.199462	0.281660
Thane	-0.599089	-1.935907	0.628735	2.564642

In []:

```
1
```

In [59]:

```
1 mydata = {
2     'Name' : ['Raj', 'Steve', 'Rohan', 'Alpha', 'Beta', 'Gamma'],
3     'Class': ['A', 'A', 'B', 'C', 'D', 'B'],
4     'Lecture': ['PHP', 'C', 'Python', 'C++', 'Python', 'C'],
5     'Grades': [90, 54, 77, 22, 25, 45],
6     'Credits': [6, 6, 6, 5, 4, 5]
7 }
```

In [60]:

```
1 df2 = pd.DataFrame(mydata)
```

In [61]:

```
1 df2
```

Out[61]:

	Name	Class	Lecture	Grades	Credits
0	Raj	A	PHP	90	6
1	Steve	A	C	54	6
2	Rohan	B	Python	77	6
3	Alpha	C	C++	22	5
4	Beta	D	Python	25	4
5	Gamma	B	C	45	5

In [62]:

```
1 re_exam_func = lambda val : "Yes" if val < 45 else "No"
```

In [67]:

```
1 df2['Re-Exam'] = df2['Grades'].apply(re_exam_func)
```

In [68]:

```
1 df2
```

Out[68]:

	Name	Class	Lecture	Grades	Credits	Re-Exam
0	Raj	A	PHP	90	6	No
1	Steve	A	C	54	6	No
2	Rohan	B	Python	77	6	No
3	Alpha	C	C++	22	5	Yes
4	Beta	D	Python	25	4	Yes
5	Gamma	B	C	45	5	No

In [69]:

```
1 result_func = lambda val : "Fail" if val < 45 else "Pass"
```

In [71]:

```
1 df2['Final_Result'] = df2['Grades'].apply(result_func)
```

In [72]:

```
1 df2
```

Out[72]:

	Name	Class	Lecture	Grades	Credits	Re-Exam	Final_Result
0	Raj	A	PHP	90	6	No	Pass
1	Steve	A	C	54	6	No	Pass
2	Rohan	B	Python	77	6	No	Pass
3	Alpha	C	C++	22	5	Yes	Fail
4	Beta	D	Python	25	4	Yes	Fail
5	Gamma	B	C	45	5	No	Pass

In [73]:

```
1 upper_func = lambda item : item.upper()
```

In [76]:

```
1 df2['Name'] = df2['Name'].apply(upper_func)
```

In [77]:

1 df2

Out[77]:

	Name	Class	Lecture	Grades	Credits	Re-Exam	Final_Result
0	RAJ	A	PHP	90	6	No	Pass
1	STEVE	A	C	54	6	No	Pass
2	ROHAN	B	Python	77	6	No	Pass
3	ALPHA	C	C++	22	5	Yes	Fail
4	BETA	D	Python	25	4	Yes	Fail
5	GAMMA	B	C	45	5	No	Pass

In []:

1

In [78]:

```

1 def grade_calc(val):
2     if val >=75:
3         return "Destinction"
4     elif val >=60 and val < 75:
5         return "First"
6     elif val >=45 and val < 60:
7         return "Second"
8     else:
9         return "Fail"

```

In [80]:

1 df2['Result_Grade'] = df2['Grades'].apply(grade_calc)

In [81]:

1 df2

Out[81]:

	Name	Class	Lecture	Grades	Credits	Re-Exam	Final_Result	Result_Grade
0	RAJ	A	PHP	90	6	No	Pass	Destinction
1	STEVE	A	C	54	6	No	Pass	Second
2	ROHAN	B	Python	77	6	No	Pass	Destinction
3	ALPHA	C	C++	22	5	Yes	Fail	Fail
4	BETA	D	Python	25	4	Yes	Fail	Fail
5	GAMMA	B	C	45	5	No	Pass	Second

In []:

1

In []:

1

sorting & ranking

In [82]:

```
1 obj = pd.Series(range(4), index=['d', 'a', 'b', 'c'])
```

In [83]:

```
1 obj
```

Out[83]:

```
d    0
a    1
b    2
c    3
dtype: int64
```

In [84]:

```
1 obj.sort_index()
```

Out[84]:

```
a    1
b    2
c    3
d    0
dtype: int64
```

In []:

1

In [86]:

```
1 frame = pd.DataFrame(np.arange(8).reshape((2,4)),
2                       index=['three', 'one'],
3                       columns=['d', 'a', 'b', 'c'])
```

In [87]:

1 frame

Out[87]:

	d	a	b	c
three	0	1	2	3
one	4	5	6	7

In [88]:

1 frame.sort_index()

Out[88]:

	d	a	b	c
one	4	5	6	7
three	0	1	2	3

In [89]:

1 frame.sort_index(axis=1)

Out[89]:

	a	b	c	d
three	1	2	3	0
one	5	6	7	4

In [90]:

1 frame.sort_index(axis=1,ascending=False)

Out[90]:

	d	c	b	a
three	0	3	2	1
one	4	7	6	5

In []:

1

Ranking

In [91]:

```

1 movies = {
2     'Name': ['Shawshank Redemption', 'The God Father', "schindler's list"],
3     'Year': ['1998', '1972', '1993'],
4     'Rating': ['9.2', '6.8', '6.8']
5 }

```

In [92]:

```
1 df_movies = pd.DataFrame(movies)
```

In [93]:

```
1 df_movies
```

Out[93]:

	Name	Year	Rating
0	Shawshank Redemption	1998	9.2
1	The God Father	1972	6.8
2	schindler's list	1993	6.8

In [94]:

```
1 df_movies['Ranking'] = df_movies['Rating'].rank(ascending=1)
```

In [95]:

```
1 df_movies
```

Out[95]:

	Name	Year	Rating	Ranking
0	Shawshank Redemption	1998	9.2	3.0
1	The God Father	1972	6.8	1.5
2	schindler's list	1993	6.8	1.5

In []:

```
1
```

In [96]:

```

1 student_marks = {
2     'Name': ['Rohan', 'Rohan', 'Raj', 'Anil', 'Anil', 'Ann', 'John', 'John', 'John', 'Lis'],
3     'Subject': ['Maths', 'Physics', 'Chemistry', 'Maths', 'Physics', 'Chemistry', 'Mat'],
4     'Marks': [80, 90, 75, 60, 40, 60, 80, 55, 100, 90, 75, 70]
5 }

```


In [97]:

```
1 student_marks
```

Out[97]:

```
{'Name': ['Rohan',  
         'Rohan',  
         'Raj',  
         'Anil',  
         'Anil',  
         'Ann',  
         'John',  
         'John',  
         'John',  
         'Lisa',  
         'Lisa',  
         'Lisa'],  
 'Subject': ['Maths',  
             'Physics',  
             'Chemistry',  
             'Maths',  
             'Physics',  
             'Chemistry',  
             'Maths',  
             'Physics',  
             'Chemistry',  
             'Maths',  
             'Physics',  
             'Chemistry'],  
 'Marks': [80, 90, 75, 60, 40, 60, 80, 55, 100, 90, 75, 70]}
```

In [98]:

```
1 marks = pd.DataFrame(student_marks)
```

In [99]:

1 marks

Out[99]:

	Name	Subject	Marks
0	Rohan	Maths	80
1	Rohan	Physics	90
2	Raj	Chemistry	75
3	Anil	Maths	60
4	Anil	Physics	40
5	Ann	Chemistry	60
6	John	Maths	80
7	John	Physics	55
8	John	Chemistry	100
9	Lisa	Maths	90
10	Lisa	Physics	75
11	Lisa	Chemistry	70

In [100]:

1 marks['Ranking'] = marks['Marks'].rank(ascending=0)

In [101]:

1 marks

Out[101]:

	Name	Subject	Marks	Ranking
0	Rohan	Maths	80	4.5
1	Rohan	Physics	90	2.5
2	Raj	Chemistry	75	6.5
3	Anil	Maths	60	9.5
4	Anil	Physics	40	12.0
5	Ann	Chemistry	60	9.5
6	John	Maths	80	4.5
7	John	Physics	55	11.0
8	John	Chemistry	100	1.0
9	Lisa	Maths	90	2.5
10	Lisa	Physics	75	6.5
11	Lisa	Chemistry	70	8.0

In []:

```
1
```

duplicate indexing

In [102]:

```
1 ser = pd.Series(range(5),index=['a','a','b','b','c'])
```

In [103]:

```
1 ser
```

Out[103]:

```
a    0
a    1
b    2
b    3
c    4
dtype: int64
```

In [104]:

```
1 ser['a']
```

Out[104]:

```
a    0
a    1
dtype: int64
```

In [109]:

```
1 ser['b']
```

Out[109]:

```
b    2
b    3
dtype: int64
```

In [106]:

```
1 ser['c']
```

Out[106]:

```
4
```

In [110]:

```
1 ser.index.is_unique
```

Out[110]:

```
False
```

In []:

1

computing descriptive stats in pandas

In [111]:

```
1 df4 = pd.DataFrame(  
2     [[1.4,np.nan],[7.1,4.5],  
3     [np.nan,np.nan],[0.75,-1.3]],  
4     index=['a','b','c','d'],columns=['A','B'])  
5 )
```

In [112]:

1 df4

Out[112]:

	A	B
a	1.40	NaN
b	7.10	4.5
c	NaN	NaN
d	0.75	-1.3

In [113]:

1 df4.sum()

Out[113]:

```
A    9.25  
B    3.20  
dtype: float64
```

In []:

1

In [114]:

1 df4.sum(axis=1)

Out[114]:

```
a    1.40  
b   11.60  
c    0.00  
d   -0.55  
dtype: float64
```

In []:

1

In [115]:

```
1 df4.mean()
```

Out[115]:

```
A    3.083333
B    1.600000
dtype: float64
```

In [118]:

```
1 df4.mean(axis=1)
```

Out[118]:

```
a    1.400
b    5.800
c     NaN
d   -0.275
dtype: float64
```

In [119]:

```
1 df4.mean(axis=1, skipna=False)
```

Out[119]:

```
a     NaN
b    5.800
c     NaN
d   -0.275
dtype: float64
```

In [117]:

```
1 df4
```

Out[117]:

	A	B
a	1.40	NaN
b	7.10	4.5
c	NaN	NaN
d	0.75	-1.3

In [121]:

```
1 df4
```

Out[121]:

	A	B
a	1.40	NaN
b	7.10	4.5
c	NaN	NaN
d	0.75	-1.3

In [120]:

```
1 df4.describe()
```

Out[120]:

	A	B
count	3.000000	2.000000
mean	3.083333	1.600000
std	3.493685	4.101219
min	0.750000	-1.300000
25%	1.075000	0.150000
50%	1.400000	1.600000
75%	4.250000	3.050000
max	7.100000	4.500000

In []:

```
1
```