In [45]:
```
!pip install pandas==1.5.3
import pandas as pd
import numpy as np

```

```
Requirement already satisfied: pandas==1.5.3 in c:\users\satya\anaconda3\lib\
site-packages (1.5.3)
Requirement already satisfied: numpy>=1.20.3 in c:\users\satya\anaconda3\lib\
site-packages (from pandas==1.5.3) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\satya\anaconda3\lib\s
ite-packages (from pandas==1.5.3) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\satya\anaco
nda3\lib\site-packages (from pandas==1.5.3) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\satya\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas==1.5.3) (1.16.0)
```

In [46]:
```
#series(one dim)
a= pd.Series([12,23,34,45,56,'45','True'])
b= pd.Series([12,6,34,4,56,9])
a
# print(b)
```

Out[46]:
```
0      12
1      23
2      34
3      45
4      56
5      45
6    True
dtype: object
```

In [47]:
```
c=pd.Series([10,24,36,52,78],index=['a','b','c','d','e'])
d=pd.Series([10,24,36,52,78],index=['a','b','c','d','e'],dtype='float')
c,d
```

Out[47]:
```
(a    10
 b    24
 c    36
 d    52
 e    78
 dtype: int64,
 a    10.0
 b    24.0
 c    36.0
 d    52.0
 e    78.0
 dtype: float64)
```

In [48]:
```
#series using dictionary
e=pd.Series({'a':23,'b':67,'c':89})
e
```

Out[48]:
```
a    23
b    67
c    89
dtype: int64
```

In [49]:
```
#dataframe(two dim)
i=pd.DataFrame([11,22,33,44,55])
i
j=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]])
j
```

Out[49]:

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

In [50]:
```
#covereting dataframe in series
pd.DataFrame(e)
```

Out[50]:

	0
a	23
b	67
c	89

In [51]:
```
k=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]],columns=['a','b','c'])
# k

l=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]],columns=['a','b','c'],index=['x',
l
```

Out[51]:

	a	b	c
x	1	2	3
y	4	5	6
z	7	8	9

In [52]:
```
#creating datframe from list of dictionaries
dict=[{'doll':1,'ball':2},{'alexa':4,'siri':10},{'dora':20}]
pd.DataFrame(dict,index=['a','b','c'])
```

Out[52]:

	doll	ball	alexa	siri	dora
a	1.0	2.0	NaN	NaN	NaN
b	NaN	NaN	4.0	10.0	NaN
c	NaN	NaN	NaN	NaN	20.0

In [53]:
```
#dataframe operations
k,l
print(k['a'])
print(l['b'])
```

```
0    1
1    4
2    7
Name: a, dtype: int64
x    2
y    5
z    8
Name: b, dtype: int64
```

In [54]:
```
l['d']=l['a']+l['b']
l['e']=l['c']+l['d']
l
```

Out[54]:

	a	b	c	d	e
x	1	2	3	3	6
y	4	5	6	9	15
z	7	8	9	15	24

In [55]:
```
l
```

Out[55]:

	a	b	c	d	e
x	1	2	3	3	6
y	4	5	6	9	15
z	7	8	9	15	24

In [56]:
```
# del l['b']
# l.pop('d')
# l.insert(1,'n',l['e'])
print(l)
```

```
   a  b  c   d   e
x  1  2  3   3   6
y  4  5  6   9  15
z  7  8  9  15  24
```

In [57]:
```
s={'c':1,'n':30,'e':20}
t=l.append(s,ignore_index=True)
t
import numpy as np
d1= pd.DataFrame({'abc':np.random.randint(2,6,size=(10)),'bcd':np.random.r
d1
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_32864\2765354361.py:2: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
a future version. Use pandas.concat instead.
  t=l.append(s,ignore_index=True)
```

Out[57]:

	abc	bcd	def
0	2	3	3
1	3	3	4
2	4	3	4
3	4	3	3
4	4	3	5
5	5	3	3
6	4	3	4
7	4	3	5
8	2	3	5
9	3	3	5

In [58]:
```
d1.head()
d1.tail()
```

Out[58]:

	abc	bcd	def
5	5	3	3
6	4	3	4
7	4	3	5
8	2	3	5
9	3	3	5

In [59]:
```
1 d1.info()
2 d1.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   abc     10 non-null     int32
 1   bcd     10 non-null     int32
 2   def     10 non-null     int32
dtypes: int32(3)
memory usage: 248.0 bytes
```

Out[59]:

	abc	bcd	def
count	10.000000	10.0	10.000000
mean	3.500000	3.0	4.100000
std	0.971825	0.0	0.875595
min	2.000000	3.0	3.000000
25%	3.000000	3.0	3.250000
50%	4.000000	3.0	4.000000
75%	4.000000	3.0	5.000000
max	5.000000	3.0	5.000000

In [60]:
```
print(d1)
d1.loc[8,'def']#loc[row index,column]
# d1.loc[2:7,['abc','def']]#loc[row index,column]
# d1.loc[3,4,7]['abc','def']
d1.iloc[0,2]
d1.iloc[1:5,[0,2]]

```

```
   abc  bcd  def
0    2    3    3
1    3    3    4
2    4    3    4
3    4    3    3
4    4    3    5
5    5    3    3
6    4    3    4
7    4    3    5
8    2    3    5
9    3    3    5
```

Out[60]:

	abc	def
1	3	4
2	4	4
3	4	3
4	4	5

In [61]:
```
d1.abc #d1['abc']
d1.bcd.values
d1['def'].values
```

Out[61]:
```
array([3, 4, 4, 3, 5, 3, 4, 5, 5, 5])
```

In [62]:
```
d1['sub']=d1.abc.values-d1.bcd.values
d1['add']=d1.abc.values+d1.bcd.values
d1

```

Out[62]:

	abc	bcd	def	sub	add
0	2	3	3	-1	5
1	3	3	4	0	6
2	4	3	4	1	7
3	4	3	3	1	7
4	4	3	5	1	7
5	5	3	3	2	8
6	4	3	4	1	7
7	4	3	5	1	7
8	2	3	5	-1	5
9	3	3	5	0	6

In [63]:
```
d2=[['ram',101,1200],['bheem',102,16000],['krish',103,20000]]
df1=pd.DataFrame(d2,columns=['name','id','salary'])
df1
```

Out[63]:

	name	id	salary
0	ram	101	1200
1	bheem	102	16000
2	krish	103	20000

In [64]:
```
m=df1[df1.salary>=15000]
print(m)
m[['id','salary']]
```

```
    name   id  salary
1  bheem  102   16000
2  krish  103   20000
```

Out[64]:

	id	salary
1	102	16000
2	103	20000

In [65]:
```
df1.append({'name':'ajay','id':106,'salary':2500},ignore_index="True")
df=pd.concat([df1,pd.DataFrame.from_records([{'name':np.NaN,'id':105,'sala
df
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_32864\2714841154.py:1: FutureWarn
ing: The frame.append method is deprecated and will be removed from pandas in
a future version. Use pandas.concat instead.
  df1.append({'name':'ajay','id':106,'salary':2500},ignore_index="True")
```

Out[65]:

	name	id	salary
0	ram	101	1200
1	bheem	102	16000
2	krish	103	20000
0	NaN	105	28000

In [66]:
```
df1.isnull()
print(df1.isnull().sum)
```

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of     name     id
salary
0  False  False   False
1  False  False   False
2  False  False   False>
```

In [67]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   name    3 non-null      object
 1   id      3 non-null      int64
 2   salary  3 non-null      int64
dtypes: int64(2), object(1)
memory usage: 200.0+ bytes
```

In [68]:
```
#df1.dropna()
#df1.dropna()
```

In [69]:
```
df1.fillna(value='arun')
df1
```

Out[69]:

	name	id	salary
0	ram	101	1200
1	bheem	102	16000
2	krish	103	20000

In [70]:
```
#df1.drop([0],axis=0,inplace=True)
df1
```

Out[70]:

	name	id	salary
0	ram	101	1200
1	bheem	102	16000
2	krish	103	20000

In [71]:
```
#Groupby
df=pd.DataFrame({'Animal' : ['Falcon', 'Falcon','Parrot', 'Parrot'],'Max S
df
```

Out[71]:

	Animal	Max Speed	wind Speed
0	Falcon	380.0	380.0
1	Falcon	370.0	370.0
2	Parrot	24.0	24.0
3	Parrot	26.0	26.0

In [72]:
```
df.groupby(['Animal']).mean()
```

Out[72]:

Animal	Max Speed	wind Speed
Falcon	375.0	375.0
Parrot	25.0	25.0

In [73]:
```

df.to_csv('data.csv')
import pandas as pd
df_csv=pd.read_csv('data.csv')
df_csv
```

Out[73]:

	Unnamed: 0	Animal	Max Speed	wind Speed
0	0	Falcon	380.0	380.0
1	1	Falcon	370.0	370.0
2	2	Parrot	24.0	24.0
3	3	Parrot	26.0	26.0

In [74]:
```
print(df_csv.columns)
df_csv.isnull().sum()
```

```
Index(['Unnamed: 0', 'Animal', 'Max Speed', 'wind Speed'], dtype='object')
```

Out[74]:
```
Unnamed: 0    0
Animal        0
Max Speed     0
wind Speed    0
dtype: int64
```

In [75]:
```
df_csv['Animal'].values
df_csv['Animal'].isnull().sum()
```

Out[75]:
```
0
```

In [77]:
```
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(4,3),columns=['col1','col2','col3'])
print(df)
print('\n')
for key,value in df.iteritems():
        print(key,value)
```

```
       col1      col2      col3
0 -0.880426 -1.054172  1.024063
1  1.031355  0.369833 -0.154678
2 -0.188384  0.565338 -0.716508
3 -1.817226 -0.468113  1.033652


col1 0   -0.880426
1    1.031355
2   -0.188384
3   -1.817226
Name: col1, dtype: float64
col2 0   -1.054172
1    0.369833
2    0.565338
3   -0.468113
Name: col2, dtype: float64
col3 0    1.024063
1   -0.154678
2   -0.716508
3    1.033652
Name: col3, dtype: float64

C:\Users\satya\AppData\Local\Temp\ipykernel_32864\827614863.py:6: FutureWarni
ng: iteritems is deprecated and will be removed in a future version. Use .ite
ms instead.
  for key,value in df.iteritems():
```

In [79]:
```
#ps.Series?
#Transpose
df.T
```

Out[79]:

	0	1	2	3
col1	-0.880426	1.031355	-0.188384	-1.817226
col2	-1.054172	0.369833	0.565338	-0.468113
col3	1.024063	-0.154678	-0.716508	1.033652

In [94]:
```
import pandas as pd
data ={'firstname':['arun','jebu','venkat','rekha','majid','mohsin'],
       'lastname':['kumar','jacob','Raghavan','Singh','khan','khan'],
       'employmenttype':['service','Business','student','service','Business
       'country':['India','USA','USA','Sweden','Australia','Germany']}
df=pd.DataFrame(data,columns=['firstname','lastname','employmenttype','cou
df
```

Out[94]:

	firstname	lastname	employmenttype	country
0	arun	kumar	service	India
1	jebu	jacob	Business	USA
2	venkat	Raghavan	student	USA
3	rekha	Singh	service	Sweden
4	majid	khan	Business	Australia
5	mohsin	khan	Business	Germany

In [97]:
```
df1=pd.get_dummies(df['employmenttype'])
df2=pd.get_dummies(df['country'])
print(df1)
print(df2)
```

```
   Business  service  student
0         0        1        0
1         1        0        0
2         0        0        1
3         0        1        0
4         1        0        0
5         1        0        0
   Australia  Germany  India  Sweden  USA
0          0        0      1       0    0
1          0        0      0       0    1
2          0        0      0       0    1
3          0        0      0       1    0
4          1        0      0       0    0
5          0        1      0       0    0
```

In [99]:
```
frame=[df,df1,df2]
result=pd.concat(frame,axis=1)
result.drop(['employmenttype','country'],axis=1,inplace=True)
result
```

Out[99]:

	firstname	lastname	Business	service	student	Australia	Germany	India	Sweden	USA
0	arun	kumar	0	1	0	0	0	1	0	0
1	jebu	jacob	1	0	0	0	0	0	0	1
2	venkat	Raghavan	0	0	1	0	0	0	0	1
3	rekha	Singh	0	1	0	0	0	0	1	0
4	majid	khan	1	0	0	1	0	0	0	0
5	mohsin	khan	1	0	0	0	1	0	0	0

In [100]:
```
import pandas as pd
india_weather = pd.DataFrame({'city':['mumbai','delhi','banglore'],
                              'temperature':[32,45,30],
                              'humidity':[80,60,78]})
india_weather
```

Out[100]:

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78

In [106]:
```
import pandas as pd
us_weather=pd.DataFrame({'city':['newyork','chicgago','orlando'],
                         'temperature':[21,14,35],
                         'humidity':[68,65,75]})
us_weather
```

Out[106]:

	city	temperature	humidity
0	newyork	21	68
1	chicgago	14	65
2	orlando	35	75

In [108]:
```
df=pd.concat([india_weather,us_weather])
df

```

Out[108]:

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78
0	newyork	21	68
1	chicgago	14	65
2	orlando	35	75

In [109]:
```
#ignore index
df=pd.concat([india_weather,us_weather],ignore_index=True)
df
```

Out[109]:

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78
3	newyork	21	68
4	chicgago	14	65
5	orlando	35	75

In [110]:
```
#ignor index
df=pd.concat([india_weather,us_weather],ignore_index=True)
df
```

Out[110]:

	city	temperature	humidity
0	mumbai	32	80
1	delhi	45	60
2	banglore	30	78
3	newyork	21	68
4	chicgago	14	65
5	orlando	35	75

In [111]:
```
df=pd.concat([india_weather,us_weather],keys=['india','us'])
df
```

Out[111]:

		city	temperature	humidity
india	**0**	mumbai	32	80
	1	delhi	45	60
	2	banglore	30	78
us	**0**	newyork	21	68
	1	chicgago	14	65
	2	orlando	35	75

In [114]:
```
x=df.to_csv('data_panda.csv')
df=pd.read_csv('data_panda.csv')
df.corr()
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_32864\822844926.py:3: FutureWarni
ng: The default value of numeric_only in DataFrame.corr is deprecated. In a f
uture version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  df.corr()
```

Out[114]:

	Unnamed: 1	temperature	humidity
Unnamed: 1	1.000000	0.247121	0.141535
temperature	0.247121	1.000000	-0.009327
humidity	0.141535	-0.009327	1.000000

In [117]:
```
import pandas as pd
emp={'name':['Parker','Smith','William','Parker'],'Age':[21,33,44,56]}
info=pd.DataFrame(emp)
print(info)
print('--------------------')
info=info.drop_duplicates()
print(info)
```

```
      name  Age
0   Parker   21
1    Smith   33
2  William   44
3   Parker   56
--------------------
      name  Age
0   Parker   21
1    Smith   33
2  William   44
3   Parker   56
```

In [121]:
```
#apply
def  cal_sum(kiran):
     return kiran+1
data={
    'x':[50,45,48,29],
    'y':[300,1112,23,4]
}
df= pd.DataFrame(data)
print(df)
x=df.apply(cal_sum)
print(x)
```

```
    x     y
0  50   300
1  45  1112
2  48    23
3  29     4
    x     y
0  51   301
1  46  1113
2  49    24
3  30     5
```

In [125]:
```
import pandas as pd
data1={"name":['sully','marry','john'],
       'age':[50,40,30]}
data2={'name':['sully','peter','john'],
       'age':[50,40,30]}
df1=pd.DataFrame(data1)
df2=pd.DataFrame(data2)
print(df1)
print(df2)
print('..........')
```

```
    name  age
0  sully   50
1  marry   40
2   john   30
    name  age
0  sully   50
1  peter   40
2   john   30
..........
```

In [127]:
```
newdf=df1.merge(df2,how='right')
print(newdf)
```

```
    name  age
0  sully   50
1  peter   40
2   john   30
```

In [129]:
```
df3=pd.merge(df1,df2, on="name",how='outer')#outer
df3
```

Out[129]:

	name	age_x	age_y
0	sully	50.0	50.0
1	marry	40.0	NaN
2	john	30.0	30.0
3	peter	NaN	40.0

In [130]:
```
df3=pd.merge(df1,df2, on="name",how='left')#left
df3
```

Out[130]:

	name	age_x	age_y
0	sully	50	50.0
1	marry	40	NaN
2	john	30	30.0

In [131]:
```
df3=pd.merge(df1,df2, on="name",how='right')#right
df3

```

Out[131]:

	name	age_x	age_y
0	sully	50.0	50
1	peter	NaN	40
2	john	30.0	30

In [132]:
```
df3=pd.merge(df1,df2, on="name",how='inner')#intersection
df3

```

Out[132]:

	name	age_x	age_y
0	sully	50	50
1	john	30	30

In [ ]:
```

```