In [ ]:

```
#dataframe from list
import pandas as pd
 # initialize list of lists
data = [["Ohio",2000,1.5],["Ohio",2001,1.7],
        ["Ohio",2002,3.6],
        ["Nevada", 2001, 2.4],["Nevada",2002,2.9]]
  # Create the pandas DataFrame
df = pd.DataFrame(data, columns = ['state',
                                   'year','pop'])
# print dataframe.
print(df)
```

```
    state  year  pop
0    Ohio  2000  1.5
1    Ohio  2001  1.7
2    Ohio  2002  3.6
3  Nevada  2001  2.4
4  Nevada  2002  2.9
```

In [ ]:

```
#Creating dataframe from dictionary
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada',
                  'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame = pd.DataFrame(data)
print(frame)
```

```
    state  year  pop
0    Ohio  2000  1.5
1    Ohio  2001  1.7
2    Ohio  2002  3.6
3  Nevada  2001  2.4
4  Nevada  2002  2.9
```

In [ ]:

```
#change allignment of columns
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada',
                  'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame = pd.DataFrame(data, columns=['year', 'state',
                                    'pop'])
print(frame)
```

```
   year   state  pop
0  2000    Ohio  1.5
1  2001    Ohio  1.7
2  2002    Ohio  3.6
3  2001  Nevada  2.4
4  2002  Nevada  2.9
```

In [ ]:

```
#given index and a  column debt is inserted
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada',
                  'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state',
                                     'pop', 'debt'],
                      index=[10,20,30,40,50])

print(frame2)
```

```
    year   state  pop debt
10  2000    Ohio  1.5  NaN
20  2001    Ohio  1.7  NaN
30  2002    Ohio  3.6  NaN
40  2001  Nevada  2.4  NaN
50  2002  Nevada  2.9  NaN
```

In [ ]:

```
#head and tail
print(frame2.head(2))
print("_____________________________")
#tail will display rows from last
print(frame2.tail(2))
```

```
   year state  pop
0  2000  Ohio  1.5
1  2001  Ohio  1.7
_____________________________
   year   state  pop
3  2001  Nevada  2.4
4  2002  Nevada  2.9
```

In [ ]:

```
#retreive row
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop', 'debt'],
                            index=["a","b","c","d","e"])
print(frame2)
print("_____________")
print(frame2.loc["b"])
```

```
   year   state  pop debt
a  2000    Ohio  1.5  NaN
b  2001    Ohio  1.7  NaN
c  2002    Ohio  3.6  NaN
d  2001  Nevada  2.4  NaN
e  2002  Nevada  2.9  NaN
_____________
year     2001
state    Ohio
pop       1.7
debt      NaN
Name: b, dtype: object
```

In [ ]:

```
#retreive column
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop'])
print(frame2)
print("_____________")
print(frame2["year"])
print("--------------")
print(frame2.year)
print("--------------")
print(frame2[["year","pop"]])
```

```
   year   state  pop
0  2000    Ohio  1.5
1  2001    Ohio  1.7
2  2002    Ohio  3.6
3  2001  Nevada  2.4
4  2002  Nevada  2.9
_____________
0    2000
1    2001
2    2002
3    2001
4    2002
Name: year, dtype: int64
--------------
0    2000
1    2001
2    2002
3    2001
4    2002
Name: year, dtype: int64
--------------
   year  pop
0  2000  1.5
1  2001  1.7
2  2002  3.6
3  2001  2.4
4  2002  2.9
```

In [ ]:

```
#slicing
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop', 'debt'],index=["a","b","c"
,"d","e"])
print(frame2)
print("_____________")
print(frame2.iloc[0:2,2:3])
```

```
   year   state  pop debt
a  2000    Ohio  1.5  NaN
b  2001    Ohio  1.7  NaN
c  2002    Ohio  3.6  NaN
d  2001  Nevada  2.4  NaN
e  2002  Nevada  2.9  NaN
_____________
   pop
a  1.5
b  1.7
```

In [ ]:

```
#Columns can be modified by assignment.
import pandas as pd
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop', 'debt'],
                      index=['one', 'two', 'three', 'four', 'five'])
frame2['debt'] = 16.5
print(frame2)
```

```
       year   state  pop  debt
one    2000    Ohio  1.5  16.5
two    2001    Ohio  1.7  16.5
three  2002    Ohio  3.6  16.5
four   2001  Nevada  2.4  16.5
five   2002  Nevada  2.9  16.5
```

In [ ]:

```
#assigning an array of values to debt
import pandas as pd
import numpy as np
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, 2001, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, 2.4, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop', 'debt'],
                      index=['one', 'two', 'three', 'four', 'five'])
frame2['debt'] = np.arange(0,5)
print(frame2)
```

```
       year   state  pop  debt
one    2000    Ohio  1.5     0
two    2001    Ohio  1.7     1
three  2002    Ohio  3.6     2
four   2001  Nevada  2.4     3
five   2002  Nevada  2.9     4
```

In [ ]:

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving order.csv to order.csv
```

In [ ]:

```
#Selection by label
# Import pandas package
import pandas as pd
# making data frame from csv file
data = pd.read_csv("/content/order.csv")
print(data)
data = pd.read_csv("/content/order.csv",index_col="Item")
print(data)
```

slno	OrderDate	Region	Rep	Item	Units	Unit Cost	Total
1	1-6-21	East	Jones	Pencil	95	1.99	189.05
2	1-23-21	Central	Kivell	Binder	50	19.99	999.50
3	2-9-21	Central	Jardine	Pencil	36	4.99	179.64
4	2-26-21	Central	Gill	Pen	27	19.99	539.73
5	3-15-21	West	Sorvino	Pencil	56	2.99	167.44
6	4-1-21	East	Jones	Binder	60	4.99	299.40
7	4-18-21	Central	Andrews	Pencil	75	1.99	149.25
8	5-5-21	Central	Jardine	Pencil	90	4.99	449.10
9	5-22-21	West	Thompson	Pencil	32	1.99	63.68
10	6-8-21	East	Jones	Binder	60	8.99	539.40
11	6-25-21	Central	Morgan	Pencil	90	4.99	449.10
12	7-12-21	East	Howard	Binder	29	1.99	57.71
13	7-29-21	East	Parent	Binder	81	19.99	1,619.19
14	8-15-21	East	Jones	Pencil	35	4.99	174.65
15	9-1-21	Central	Smith	Desk	2	125.00	250.00
16	9-18-21	East	Jones	Pen Set	16	15.99	255.84
17	10-5-21	Central	Morgan	Binder	28	8.99	251.72
18	10-22-21	East	Jones	Pen	64	8.99	575.36
19	11-8-21	East	Parent	Pen	15	19.99	299.85
20	11-25-21	Central	Kivell	Pen Set	96	4.99	479.04
21	12-12-21	Central	Smith	Pencil	67	1.29	86.43
22	12-29-21	East	Parent	Pen Set	74	15.99	1,183.26
23	1-15-22	Central	Gill	Binder	46	8.99	413.54
24	2-1-22	Central	Smith	Binder	87	15.00	1,305.00
25	2-18-22	East	Jones	Binder	4	4.99	19.96
26	3-7-22	West	Sorvino	Binder	7	19.99	139.93
27	3-24-22	Central	Jardine	Pen Set	50	4.99	249.50
28	4-10-22	Central	Andrews	Pencil	66	1.99	131.34
29	4-27-22	East	Howard	Pen	96	4.99	479.04
30	5-14-22	Central	Gill	Pencil	53	1.29	68.37

7
30 31 5-31-22 Central Gill Binder 80 8.99 719.2
0
31 32 6-17-22 Central Kivell Desk 5 125.00 625.0
0
32 33 7-4-22 East Jones Pen Set 62 4.99 309.3
8
33 34 7-21-22 Central Morgan Pen Set 55 12.49 686.9
5
34 35 8-7-22 Central Kivell Pen Set 42 23.95 1,005.9
0
35 36 8-24-22 West Sorvino Desk 3 275.00 825.0
0
36 37 9-10-22 Central Gill Pencil 7 1.29 9.0
3
37 38 9-27-22 West Sorvino Pen 76 1.99 151.2
4
38 39 10-14-22 West Thompson Binder 57 19.99 1,139.4
3
39 40 10-31-22 Central Andrews Pencil 14 1.29 18.0
6
40 41 11-17-22 Central Jardine Binder 11 4.99 54.8
9
41 42 12-4-22 Central Jardine Binder 94 19.99 1,879.0
6
42 43 12-21-22 Central Andrews Binder 28 4.99 139.7
2

Item	slno	OrderDate	Region	Rep	Units	Unit Cost	Total
Pencil	1	1-6-21	East	Jones	95	1.99	189.05
Binder	2	1-23-21	Central	Kivell	50	19.99	999.50
Pencil	3	2-9-21	Central	Jardine	36	4.99	179.64
Pen	4	2-26-21	Central	Gill	27	19.99	539.73
Pencil	5	3-15-21	West	Sorvino	56	2.99	167.44
Binder	6	4-1-21	East	Jones	60	4.99	299.40
Pencil	7	4-18-21	Central	Andrews	75	1.99	149.25
Pencil	8	5-5-21	Central	Jardine	90	4.99	449.10
Pencil	9	5-22-21	West	Thompson	32	1.99	63.68
Binder	10	6-8-21	East	Jones	60	8.99	539.40
Pencil	11	6-25-21	Central	Morgan	90	4.99	449.10
Binder	12	7-12-21	East	Howard	29	1.99	57.71
Binder	13	7-29-21	East	Parent	81	19.99	1,619.19
Pencil	14	8-15-21	East	Jones	35	4.99	174.65
Desk	15	9-1-21	Central	Smith	2	125.00	250.00
Pen Set	16	9-18-21	East	Jones	16	15.99	255.84
Binder	17	10-5-21	Central	Morgan	28	8.99	251.72
Pen	18	10-22-21	East	Jones	64	8.99	575.36
Pen	19	11-8-21	East	Parent	15	19.99	299.85
Pen Set	20	11-25-21	Central	Kivell	96	4.99	479.04
Pencil	21	12-12-21	Central	Smith	67	1.29	86.43
Pen Set	22	12-29-21	East	Parent	74	15.99	1,183.26
Binder	23	1-15-22	Central	Gill	46	8.99	413.54
Binder	24	2-1-22	Central	Smith	87	15.00	1,305.00
Binder	25	2-18-22	East	Jones	4	4.99	19.96
Binder	26	3-7-22	West	Sorvino	7	19.99	139.93
Pen Set	27	3-24-22	Central	Jardine	50	4.99	249.50
Pencil	28	4-10-22	Central	Andrews	66	1.99	131.34
Pen	29	4-27-22	East	Howard	96	4.99	479.04
Pencil	30	5-14-22	Central	Gill	53	1.29	68.37
Binder	31	5-31-22	Central	Gill	80	8.99	719.20
Desk	32	6-17-22	Central	Kivell	5	125.00	625.00

Pen Set	33	7-4-22	East	Jones	62	4.99	309.38
Pen Set	34	7-21-22	Central	Morgan	55	12.49	686.95
Pen Set	35	8-7-22	Central	Kivell	42	23.95	1,005.90
Desk	36	8-24-22	West	Sorvino	3	275.00	825.00
Pencil	37	9-10-22	Central	Gill	7	1.29	9.03
Pen	38	9-27-22	West	Sorvino	76	1.99	151.24
Binder	39	10-14-22	West	Thompson	57	19.99	1,139.43
Pencil	40	10-31-22	Central	Andrews	14	1.29	18.06
Binder	41	11-17-22	Central	Jardine	11	4.99	54.89
Binder	42	12-4-22	Central	Jardine	94	19.99	1,879.06
Binder	43	12-21-22	Central	Andrews	28	4.99	139.72

In [ ]:

```
#Selection by label
# Import pandas package
import pandas as pd
# making data frame from csv file
data = pd.read_csv("/content/order.csv",
                   index_col ="slno")
print("_____________________________")
print("retrieving row by loc method")
print(data.loc[2])
print("_____________________________")
print(data.loc[:, ["OrderDate", "Units"] ])
print("_____________________________")
print(data.loc[1:3, ["OrderDate", "Units"] ])
print("_____________________________")
print(data.loc[1:5:2, : ])
print("----------------------")
```

```
_____________________________
retrieving row by loc method
OrderDate    1-23-21
Region       Central
Rep           Kivell
Item          Binder
Units             50
Unit Cost      19.99
Total          999.5
Name: 2, dtype: object
_____________________________
       OrderDate  Units
slno
1     01-06-2021     95
2        1-23-21     50
3     02-09-2021     36
4        2-26-21     27
5        3-15-21     56
6     04-01-2021     60
7        4-18-21     75
8     05-05-2021     90
9        5-22-21     32
10    06-08-2021     60
11       6-25-21     90
12    07-12-2021     29
13       7-29-21     81
14       8-15-21     35
15    09-01-2021      2
16       9-18-21     16
17    10-05-2021     28
18      10-22-21     64
19    11-08-2021     15
20      11-25-21     96
21    12-12-2021     67
22      12-29-21     74
23       1-15-22     46
24    02-01-2022     87
25       2-18-22      4
26    03-07-2022      7
27       3-24-22     50
28    04-10-2022     66
29       4-27-22     96
30       5-14-22     53
31       5-31-22     80
32       6-17-22      5
33    07-04-2022     62
34       7-21-22     55
35    08-07-2022     42
36       8-24-22      3
37    09-10-2022      7
38       9-27-22     76
39      10-14-22     57
40      10-31-22     14
41      11-17-22     11
42    12-04-2022     94
43      12-21-22     28
_____________________________
       OrderDate  Units
slno
1     01-06-2021     95
2        1-23-21     50
```

```
3     02-09-2021     36
```

```
       OrderDate   Region      Rep    Item  Units  Unit Cost   Total
slno
1     01-06-2021     East    Jones  Pencil     95       1.99  189.05
3     02-09-2021  Central  Jardine  Pencil     36       4.99  179.64
5        3-15-21     West  Sorvino  Pencil     56       2.99  167.44
```

In [ ]:

```
#Selection by label
# Import pandas package
import pandas as pd
# making data frame from csv file
data = pd.read_csv("/content/ord.csv",index_col ="slno")

#print("all rows")
print(data.loc[:])
print("-------------------------")
#print(all rows and 2 columns)
print(data.loc[:,["Region","Item"]])
print("-------------------------")
#print(from 3rd label)
print(data.loc[5:,["Region","Item"]])
print("-------------------------")
```

```
       OrderDate   Region      Rep    Item  Units  Unit Cost
slno
1     01-06-2021     East    Jones  Pencil     95       1.99
2        1-23-21  Central   Kivell  Binder     50      19.99
3     02-09-2021  Central  Jardine  Pencil     36       4.99
4        2-26-21  Central     Gill     Pen     27      19.99
5        3-15-21     West  Sorvino  Pencil     56       2.99
6     04-01-2021     East    Jones  Binder     60       4.99
7        4-18-21  Central  Andrews  Pencil     75       1.99
8     05-05-2021  Central  Jardine  Pencil     90       4.99
-------------------------
       Region    Item
slno
1        East  Pencil
2     Central  Binder
3     Central  Pencil
4     Central     Pen
5        West  Pencil
6        East  Binder
7     Central  Pencil
8     Central  Pencil
-------------------------
       Region    Item
slno
5        West  Pencil
6        East  Binder
7     Central  Pencil
8     Central  Pencil
-------------------------
```

In [ ]:

```
#missing values
import pandas as pd
import numpy as np
data = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, np.nan, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, np.nan, 2.9]}
frame2 = pd.DataFrame(data, columns=['year', 'state', 'pop',np.nan],
                      index=['one', 'two', 'three', 'four', 'five'])
df=pd.DataFrame(data)
print(df)
print("----------------")
print("is null")
print(df.isnull())
print("_____________")
print("not null")
print(df.notnull())
print("-----------------")
# filling missing value using fillna()
print(df.fillna(0))
print("____________________")
print("filling missing value using mean value()")
print(df.fillna(df.mean()))
print("____________________")
#filling the NaN values by interpolation
print(df.interpolate())
print("-----------------------")
#replace missing values with -1
print(df.replace(np.nan,-1))
print("-------------------------")
```

```
    state    year  pop
0    Ohio  2000.0  1.5
1    Ohio     NaN  1.7
2    Ohio  2002.0  3.6
3  Nevada  2001.0  NaN
4  Nevada  2002.0  2.9
----------------
is null
   state   year    pop
0  False  False  False
1  False   True  False
2  False  False  False
3  False  False   True
4  False  False  False
_______________
not null
   state   year    pop
0   True   True   True
1   True  False   True
2   True   True   True
3   True   True  False
4   True   True   True
-----------------
    state    year  pop
0    Ohio  2000.0  1.5
1    Ohio     0.0  1.7
2    Ohio  2002.0  3.6
3  Nevada  2001.0  0.0
4  Nevada  2002.0  2.9
____________________
filling missing value using mean value()
    state     year    pop
0    Ohio  2000.00  1.500
1    Ohio  2001.25  1.700
2    Ohio  2002.00  3.600
3  Nevada  2001.00  2.425
4  Nevada  2002.00  2.900
____________________
    state    year   pop
0    Ohio  2000.0  1.50
1    Ohio  2001.0  1.70
2    Ohio  2002.0  3.60
3  Nevada  2001.0  3.25
4  Nevada  2002.0  2.90
-----------------------
    state    year  pop
0    Ohio  2000.0  1.5
1    Ohio    -1.0  1.7
2    Ohio  2002.0  3.6
3  Nevada  2001.0 -1.0
4  Nevada  2002.0  2.9
-------------------------

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:22: FutureWar
ning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_
only=None') is deprecated; in a future version this will raise TypeError.
Select only valid columns before calling the reduction.
```

In [ ]:

```
#selection by index
# Import pandas package
import pandas as pd
# making data frame from csv file
data = pd.read_csv("/content/order.csv", index_col ="Region")
print("_____________________________")
print(data.iloc[1])
print("_____________________")
print("row from 0 to 2 and columns from 0 and 1")
print(data.iloc[0:3,[0,2]])
print("_____________________")
print(data.iloc[0:3,0:2])
print("_____________________")
print("row with index 1 2 and 4,column with 0 1 and 2 ")
print(data.iloc[[1,2,4],[0,2]])
```

```
_____________________________
slno                 2
OrderDate     1-23-21
Rep            Kivell
Item           Binder
Units              50
Unit Cost       19.99
Total          999.50
Name: Central, dtype: object
_____________________
row from 0 to 2 and columns from 0 and 1
         slno      Rep
Region
East        1    Jones
Central     2   Kivell
Central     3  Jardine
_____________________
         slno OrderDate
Region
East        1    1-6-21
Central     2   1-23-21
Central     3    2-9-21
_____________________
row with index 1 2 and 4,column with 0 1 and 2
         slno      Rep
Region
Central     2   Kivell
Central     3  Jardine
West        5  Sorvino
```

In [ ]:

```
#dropping the rows containing null values
print(df.dropna())
```

```
    state    year  pop
0    Ohio  2000.0  1.5
2    Ohio  2002.0  3.6
4  Nevada  2002.0  2.9
```

In order to iterate over rows, we can use three function iteritems(), iterrows(), itertuples() .

In [ ]:

```
import pandas as pd
# dictionary of lists
dict = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, np.nan, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, np.nan, 2.9]}
# creating a dataframe from a dictionary
df = pd.DataFrame(dict)
print(df)
for i in df.itertuples(): # this will get each row as a tuple
    print(i)
    print()
```

```
    state    year  pop
0    Ohio  2000.0  1.5
1    Ohio     NaN  1.7
2    Ohio  2002.0  3.6
3  Nevada  2001.0  NaN
4  Nevada  2002.0  2.9
Pandas(Index=0, state='Ohio', year=2000.0, pop=1.5)

Pandas(Index=1, state='Ohio', year=nan, pop=1.7)

Pandas(Index=2, state='Ohio', year=2002.0, pop=3.6)

Pandas(Index=3, state='Nevada', year=2001.0, pop=nan)

Pandas(Index=4, state='Nevada', year=2002.0, pop=2.9)
```

In [ ]:

```
#iterrows and iteritems
import pandas as pd
# dictionary of lists
dict = {'state': ['Ohio', 'Ohio', 'Ohio', 'Nevada', 'Nevada'],
 'year': [2000, np.nan, 2002, 2001, 2002],
 'pop': [1.5, 1.7, 3.6, np.nan, 2.9]}
# creating a dataframe from a dictionary
df = pd.DataFrame(dict)
print("iterrows")
for i,j in df.iterrows(): # this will get each index and each row values
    print(i,j)
    print("_____________________")
print("iteritems")
for i,j in df.iteritems():# this will extract each field seperately
    print(i,j)
    print("----------------------")
```

```
iterrows
0 state      Ohio
year     2000.0
pop         1.5
Name: 0, dtype: object
__________________
1 state    Ohio
year      NaN
pop       1.7
Name: 1, dtype: object
__________________
2 state      Ohio
year     2002.0
pop         3.6
Name: 2, dtype: object
__________________
3 state    Nevada
year     2001.0
pop         NaN
Name: 3, dtype: object
__________________
4 state    Nevada
year     2002.0
pop         2.9
Name: 4, dtype: object
__________________
iteritems
state 0      Ohio
1      Ohio
2      Ohio
3    Nevada
4    Nevada
Name: state, dtype: object
----------------------
year 0    2000.0
1       NaN
2    2002.0
3    2001.0
4    2002.0
Name: year, dtype: float64
----------------------
pop 0    1.5
1    1.7
2    3.6
3    NaN
4    2.9
Name: pop, dtype: float64
----------------------
```