

PANDAS- DATA FRAME

1.a Create a dataframe which looks like the output shown below.

```
>>> df
   a  b  c
0  1  5 11
1  2  6 12
2  8  9 30
3  4  8 14
```

Ans:

```
>>> import pandas as pd
>>> df=pd.DataFrame({'a':[1,2,8,4], 'b':[5,6,9,8], 'c':[11,12,30,14]})
>>> df
   a  b  c
0  1  5 11
1  2  6 12
2  8  9 30
3  4  8 14
```

1.b. Create a dataframe which looks like the output shown below.

Expected Output:

```
   X  Y  Z
0 78 84 86
1 85 94 97
2 96 89 96
3 80 83 72
4 86 86 83
```

Ans:

```
>>> df=pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86], 'Z':[86,97,96,72,83]})
>>> df
   X  Y  Z
0 78 84 86
1 85 94 97
2 96 89 96
3 80 83 72
4 86 86 83
```

2. Create and display a DataFrame from a specified dictionary data which has the index labels.:

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3,

2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data = pd.DataFrame({'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}, index=['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'])
>>> exam_data
```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	NaN	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

3. Write a python script to display a summary of the basic information about a specified DataFrame and its data. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.info()
<class 'pandas.core.frame.DataFrame'>
Index: 10 entries, a to j
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   name         10 non-null    object
1   score        8 non-null     float64
2   attempts     10 non-null    int64
3   qualify      10 non-null    object
dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
```

5. Write a python script to select the 'name' and 'score' columns from the following DataFrame. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5,

np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.iloc[:, :2]
```

	name	score
a	Anastasia	12.5
b	Dima	9.0
c	Katherine	16.5
d	James	NaN
e	Emily	9.0
f	Michael	20.0
g	Matthew	14.5
h	Laura	NaN
i	Kevin	8.0
j	Jonas	19.0

6. Write a python script to select the specified columns and rows from a given data frame. Sample Python dictionary data and list labels: Select 'name' and 'score' columns in rows 1, 3, 5, 6 from the following data frame. exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.loc[['b', 'd', 'f', 'g'], ['score', 'qualify']]
```

	score	qualify
b	9.0	no
d	NaN	no
f	20.0	yes
g	14.5	yes

7. Write a python script to select the rows where the number of attempts in the examination is greater than 2. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

Number of attempts in the examination is greater than 2:

name score attempts qualify

b Dima 9.0 3 no

d James NaN 3 no

f Michael 20.0 3 yes

Ans:

```
>>> exam_data.loc[exam_data['attempts']>2]
```

	name	score	attempts	qualify
b	Dima	9.0	3	no
d	James	NaN	3	no
f	Michael	20.0	3	yes

8. Write a python script to count the number of rows and columns of a DataFrame. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> print('Number of Rows:', exam_data.shape[0])
Number of Rows: 10
>>> print('Number of Columns:', exam_data.shape[1])
Number of Columns: 4
```

9. Write a python script to select the rows where the score is missing, i.e. is NaN. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data[exam_data['score'].isnull()]
```

	name	score	attempts	qualify
d	James	NaN	3	no
h	Laura	NaN	1	no

10. Write a python script to select the rows the score is between 15 and 20 (inclusive). Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.loc[(exam_data['score']>=15) & (exam_data['score']<=20)]
```

	name	score	attempts	qualify
c	Katherine	16.5	2	yes
f	Michael	20.0	3	yes
j	Jonas	19.0	1	yes

11. Write a python script to select the rows where number of attempts in the examination is less than 2 and score greater than 15. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 2]}

1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']) labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

Number of attempts in the examination is less than 2 and score greater than 15 :

name score attempts qualify

j Jonas 19.0 1 yes

Ans:

```
>>> exam_data.loc[(exam_data['attempts']<2) & (exam_data['score']>15)]
      name  score  attempts  qualify
j  Jonas   19.0         1      yes
```

12. Write a python script to change the score in row 'd' to 11.5. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.loc[['d'], ['score']] = 11.5
>>> exam_data
      name  score  attempts  qualify
a  Anastasia   12.5         1      yes
b      Dima     9.0         3       no
c  Katherine   16.5         2      yes
d      James   11.5         3       no
e      Emily     9.0         2       no
f   Michael   20.0         3      yes
g   Matthew   14.5         1      yes
h      Laura    NaN         1       no
i      Kevin     8.0         2       no
j      Jonas   19.0         1      yes
```

13. Write a python script to calculate the sum of the examination attempts by the students. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

Sum of the examination attempts by the students:

19

Ans:

```
>>> exam_data.loc[:, ['attempts']].sum()
attempts      19
dtype: int64
```

14. Write a python script to calculate the mean score for each different student in DataFrame.

Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

Mean score for each different student in data frame:

13.5625

Ans:

```
>>> exam_data.loc[:, ['score']].mean()
score      13.333333
dtype: float64
```

15. Write a python script to append a new row 'k' to data frame with given values for each column.

Now delete the new row and return the original DataFrame. Sample Python dictionary data and list

labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'] **Values for each column will be:** name : "Suresh", score: 15.5, attempts: 1, qualify: "yes", label: "k"

Ans:

```

>>> exam_data.loc['k']=['Suresh',15.5,1,'yes']
>>> exam_data

```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	11.5	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes
k	Suresh	15.5	1	yes

```

>>> exam_data=exam_data.drop('k')
>>> exam_data

```

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	11.5	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

16.a. Write a python script to sort the DataFrame first by 'name' in descending order. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans: `>>> exam_data`

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
b	Dima	9.0	3	no
c	Katherine	16.5	2	yes
d	James	11.5	3	no
e	Emily	9.0	2	no
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
j	Jonas	19.0	1	yes

`>>> exam_data.sort_values(by='name',ascending= False)`

	name	score	attempts	qualify
f	Michael	20.0	3	yes
g	Matthew	14.5	1	yes
h	Laura	NaN	1	no
i	Kevin	8.0	2	no
c	Katherine	16.5	2	yes
j	Jonas	19.0	1	yes
d	James	11.5	3	no
e	Emily	9.0	2	no
b	Dima	9.0	3	no
a	Anastasia	12.5	1	yes

16.b. Write a python script to sort the DataFrame first by 'qualify' in descending order. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:


```

>>> exam_data
      name  score  attempts  qualify
a  Anastasia   12.5         1     yes
b      Dima     9.0         3      no
c  Katherine   16.5         2     yes
d      James   11.5         3      no
e      Emily    9.0         2      no
f   Michael   20.0         3     yes
g   Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes
>>> exam_data.sort_values(by='qualify',ascending= False)
      name  score  attempts  qualify
a  Anastasia   12.5         1     yes
c  Katherine   16.5         2     yes
f   Michael   20.0         3     yes
g   Matthew   14.5         1     yes
j      Jonas   19.0         1     yes
b      Dima     9.0         3      no
d      James   11.5         3      no
e      Emily    9.0         2      no
h      Laura    NaN         1      no
i      Kevin    8.0         2      no

```

16.c. Write a python script to sort the DataFrame first by 'qualify' in descending order, and attempts in ascending order. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans: >>> exam_data

```

      name  score  attempts  qualify
a  Anastasia   12.5         1     yes
b      Dima     9.0         3      no
c  Katherine   16.5         2     yes
d      James   11.5         3      no
e      Emily    9.0         2      no
f   Michael   20.0         3     yes
g   Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes
>>> exam_data.sort_values(by=['qualify','attempts'],ascending= [False,True])
      name  score  attempts  qualify
g   Matthew   14.5         1     yes
j      Jonas   19.0         1     yes
c  Katherine   16.5         2     yes
f   Michael   20.0         3     yes
h      Laura    NaN         1      no
e      Emily    9.0         2      no
i      Kevin    8.0         2      no
b      Dima     9.0         3      no
d      James   11.5         3      no

```

17. Write a python script to replace the 'qualify' column contains the values 'yes' and 'no' with True and False. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.replace({'qualify':{'yes':True, 'no':False}})
   name  score  attempts  qualify
a  Anastasia   12.5         1     True
b      Dima     9.0         3    False
c  Katherine   16.5         2     True
d      James   11.5         3    False
e      Emily    9.0         2    False
f   Michael   20.0         3     True
g   Matthew   14.5         1     True
h      Laura    NaN         1    False
i      Kevin    8.0         2    False
j      Jonas   19.0         1     True
```

18. Write a python script to change the name 'James' to 'Suresh' in name column of the DataFrame. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.replace({'name':{'James':'Suresh'}})
   name  score  attempts  qualify
a  Anastasia   12.5         1     yes
b      Dima     9.0         3     no
c  Katherine   16.5         2     yes
d      Suresh   11.5         3     no
e      Emily    9.0         2     no
f   Michael   20.0         3     yes
g   Matthew   14.5         1     yes
h      Laura    NaN         1     no
i      Kevin    8.0         2     no
j      Jonas   19.0         1     yes
```

19. Write a python script to delete the 'attempts' column from the DataFrame. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',

'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> exam_data.drop(['attempts'],axis=1)
      name  score qualify
a  Anastasia   12.5    yes
b      Dima    9.0     no
c  Katherine   16.5    yes
d      James   11.5     no
e      Emily    9.0     no
f   Michael   20.0    yes
g   Matthew   14.5    yes
h      Laura    NaN     no
i      Kevin    8.0     no
j      Jonas   19.0    yes
```

20. Write a python script to insert a new column in existing DataFrame. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}] labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> df=['Red','Blue','Green','Red','Red','Blue','Green','Green','Green','Red']
>>> exam_data['color']=df
>>> exam_data
      name  score  attempts  qualify  color
a  Anastasia   12.5         1     yes   Red
b      Dima    9.0         3      no   Blue
c  Katherine   16.5         2     yes  Green
d      James   11.5         3      no   Red
e      Emily    9.0         2      no   Red
f   Michael   20.0         3     yes   Blue
g   Matthew   14.5         1     yes  Green
h      Laura    NaN         1      no  Green
i      Kevin    8.0         2      no  Green
j      Jonas   19.0         1     yes   Red
```

21. Write a Pandas program to iterate over rows in a DataFrame. Sample Python dictionary data and list labels: exam_data = [{'name':'Anastasia', 'score':12.5}, {'name':'Dima','score':9}, {'name':'Katherine','score':16.5}]

Ans:

```
>>> exam_data=pd.DataFrame([{'name':'Anastasia', 'score':12.5}, {'name':'Dima', 'score':9}, {'name':'Katherine', 'score':16.5}])
>>> for index,row in exam_data.iterrows():
        print(row["name"],row["score"])
```

```
Anastasia 12.5
Dima 9.0
Katherine 16.5
```

22. Write a Pandas program to get list from DataFrame column headers. Sample Python dictionary data and list labels: exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Ans:

```
>>> list(exam_data.columns.values)
['name', 'score', 'attempts', 'qualify']
```

23. Write a Pandas program to rename columns of a given DataFrame

Sample data:

Original DataFrame

col1 col2 col3

0 1 4 7

1 2 5 8

2 3 6 9

New DataFrame after renaming columns:

Column1 Column2 Column3

0 1 4 7

1 2 5 8

2 3 6 9

Ans:

```
>>> df=pd.DataFrame({'col1':[1,2,3], 'col2':[4,5,6], 'col3':[7,8,9]})
>>> df
   col1  col2  col3
0     1     4     7
1     2     5     8
2     3     6     9
>>> df.rename(columns={'col1':'column1', 'col2':'column2', 'col3':'column3'}, inplace=True)
>>> df
   column1  column2  column3
0         1         4         7
1         2         5         8
2         3         6         9
```

24. Write a Pandas program to select rows from a given DataFrame based on values in some columns. Sample data:

Original DataFrame

col1 col2 col3

0 1 4 7

1 4 5 8

2 3 6 9

3 4 7 0

4 5 8 1

Rows for column1 value == 4

col1 col2 col3

1 4 5 8

3 4 7 0

```
>>> df=pd.DataFrame({'col1':[1,4,3,4,5], 'col2':[4,5,6,7,8], 'col3':[7,8,9,0,1]})
```

Ans: >>> df

```
col1 col2 col3
```

```
0    1    4    7
```

```
1    4    5    8
```

```
2    3    6    9
```

```
3    4    7    0
```

```
4    5    8    1
```

```
>>> df[df.loc[:, 'col1']==4]
```

```
col1 col2 col3
```

```
1    4    5    8
```

```
3    4    7    0
```

25. Write a Pandas program to change the order of a DataFrame columns.

After altering col1 and col3

col3 col2 col1

0 7 4 1

1 8 5 4

2 9 6 3

3 0 7 4

4 1 8 5

Ans:

```
>>> df.iloc[:,::-1]
```

```
col3 col2 col1
```

```
0    7    4    1
```

```
1    8    5    4
```

```
2    9    6    3
```

```
3    0    7    4
```

```
4    1    8    5
```

26. Write a Pandas program to add one row in an existing DataFrame.

Ans:

```
>>> df.loc['5']=[10,11,12]
```

```
>>> df
```

```
col1 col2 col3
```

```
0    1    4    7
```

```
1    4    5    8
```

```
2    3    6    9
```

```
3    4    7    0
```

```
4    5    8    1
```

```
5   10   11   12
```

27. Write a Pandas program to count city wise number of people from a given of data set (city, name of the person).

Sample data:

city Number of people

0 California 4

1 Georgia 2

2 Los Angeles 4

Ans:

```
>>> df= pd.DataFrame({'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
'city': ['California', 'Los Angeles', 'California', 'California', 'California',
'Los Angeles', 'Los Angeles', 'Georgia', 'Georgia', 'Los Angeles']})
>>> d = df.groupby(["city"]).size().reset_index(name='Number of people')
>>> d
```

	city	Number of people
0	California	4
1	Georgia	2
2	Los Angeles	4

28. Write a Pandas program to delete DataFrame row(s) based on given column value.

Sample data:

Original DataFrame

col1 col2 col3

0 1 4 7

1 4 5 8

2 3 6 9

3 4 7 0

4 5 8 1

New DataFrame

col1 col2 col3

0 1 4 7

2 3 6 9

3 4 7 0

4 5 8 1

Ans:

```
>>> df=pd.DataFrame({'col1':[1,4,3,4,5], 'col2':[4,5,6,7,8], 'col3':[7,8,9,0,1]})
>>> df
```

	col1	col2	col3
0	1	4	7
1	4	5	8
2	3	6	9
3	4	7	0
4	5	8	1

```
>>> df[df.loc[:, 'col2']!=5]
```

	col1	col2	col3
0	1	4	7
2	3	6	9
3	4	7	0
4	5	8	1

29. Write a Pandas program to select a row of series/dataframe by given integer index.

Ans:

```
>>> df.iloc[[2],:]
      col1  col2  col3
2         3     6     9
```

30. Write a Pandas program to replace all the NaN values with Zero's in a column of a dataframe.

Ans:

```
>>> exam_data = pd.DataFrame({'name': ['Anastasia', 'Dima', 'Katherine', 'James',
    'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
    'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
    'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
    'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']})
>>> exam_data
   name  score  attempts  qualify
0 Anastasia  12.5         1     yes
1      Dima   9.0         3      no
2 Katherine  16.5         2     yes
3      James   NaN         3      no
4      Emily   9.0         2      no
5   Michael  20.0         3     yes
6   Matthew  14.5         1     yes
7      Laura   NaN         1      no
8      Kevin   8.0         2      no
9      Jonas  19.0         1     yes
>>> exam_data=exam_data.fillna(0)
>>> exam_data
   name  score  attempts  qualify
0 Anastasia  12.5         1     yes
1      Dima   9.0         3      no
2 Katherine  16.5         2     yes
3      James   0.0         3      no
4      Emily   9.0         2      no
5   Michael  20.0         3     yes
6   Matthew  14.5         1     yes
7      Laura   0.0         1      no
8      Kevin   8.0         2      no
9      Jonas  19.0         1     yes
```

31. Write a Pandas program to convert index in a column of the given dataframe.

Ans:

```

>>> exam_data
   name  score  attempts  qualify
0  Anastasia  12.5         1     yes
1      Dima   9.0         3     no
2  Katherine  16.5         2     yes
3      James  NaN         3     no
4      Emily   9.0         2     no
5   Michael  20.0         3     yes
6   Matthew  14.5         1     yes
7      Laura  NaN         1     no
8      Kevin   8.0         2     no
9      Jonas  19.0         1     yes
>>> exam_data.reset_index(level=0,inplace=True)
>>> exam_data
   index  name  score  attempts  qualify
0      0  Anastasia  12.5         1     yes
1      1      Dima   9.0         3     no
2      2  Katherine  16.5         2     yes
3      3      James  NaN         3     no
4      4      Emily   9.0         2     no
5      5   Michael  20.0         3     yes
6      6   Matthew  14.5         1     yes
7      7      Laura  NaN         1     no
8      8      Kevin   8.0         2     no
9      9      Jonas  19.0         1     yes
>>> print( exam_data.to_string(index=False))
   index  name  score  attempts  qualify
0  Anastasia  12.5         1     yes
1      Dima   9.0         3     no
2  Katherine  16.5         2     yes
3      James  NaN         3     no
4      Emily   9.0         2     no
5   Michael  20.0         3     yes
6   Matthew  14.5         1     yes
7      Laura  NaN         1     no
8      Kevin   8.0         2     no
9      Jonas  19.0         1     yes

```