Lab Exercise: DataFrame

```
Ex:1
>>> import pandas as pd
>>> df=pd.DataFrame([1, 3, 5, 12, 6, 8],[10,11,12,20,50,8])
>>> df
     0
10   1
11   3
12   5
20  12
50   6
8    8

Ex:2
>>> df=pd.DataFrame({'A':[1, 3, 5, 12, 6, 8],
'B':[10,11,12,20,50,8]},index=[0,1,2,3,4,5])
>>> df
    A   B
0   1  10
1   3  11
2   5  12
3  12  20
4   6  50
5   8   8
```

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
```

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
```

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']
Expected Output:

	attempts	name	qualify	score
a	1	Anastasia	yes	12.5
b	3	Dima	no	9.0

....

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']
Expected Output:

Index: 10 entries, a to j
Data columns (total 4 columns):
.... dtypes: float64(1), int64(1), object(2)
memory usage: 400.0+ bytes
None

4. Write a python script to get the first 3 rows of a given 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:
First three rows of the data frame:
attempts name qualify score
a 1 Anastasia yes 12.5
b 3 Dima no 9.0
c 2 Katherine yes 16.5

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']
Expected Output:
Select specific columns:
name score
a Anastasia 12.5
b Dima 9.0

c Katherine 16.5
... 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']
Expected Output:
Select specific columns and rows:
  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

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']
Expected Output:
Number of Rows: 10
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', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
Expected Output:
Rows where score is missing:
attempts name qualify score
d 3 James no NaN
h 1 Laura no NaN

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', labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
Expected Output:
Rows where score between 15 and 20 (inclusive):
attempts name qualify score
c 2 Katherine yes 16.5
f 3 Michael yes 20.0
j 1 Jonas yes 19.0

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, 1],
'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 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

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']
Expected Output:
Change the score in row 'd' to 11.5:
attempts name qualify score
a 1 Anastasia yes 12.5
b 3 Dima no 9.0
c 2 Katherine yes 16.5
...
i 2 Kevin no 8.0
j 1 Jonas yes 19.0

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

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

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"
Expected Output:
Append a new row:
Print all records after insert a new record:
attempts name qualify score
a 1 Anastasia yes 12.5
b 3 Dima no 9.0
.....
j 1 Jonas yes 19.0
k 1 Suresh yes 15.5
Delete the new row and display the original rows:
attempts name qualify score
a 1 Anastasia yes 12.5
b 3 Dima no 9.0
.....
i 2 Kevin no 8.0
j 1 Jonas yes 19.0

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']

Expected Output:
Orginal rows:
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
Sort the data frame first by 'name' in descending order

	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	NaN	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']

Expected Output:
Orginal rows:
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
Sort the data frame first by 'qualify' in descending order

	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	NaN	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']
Expected Output:
Orginal rows:
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
Sort the data frame first by 'qualify' in descending order and 'attempts' in ascending order.

	name	score	attempts	qualify
a	Anastasia	12.5	1	yes
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	NaN	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']
Expected Output:
Replace the 'qualify' column contains the values 'yes' and 'no' with True and False:
attempts name qualify score
a 1 Anastasia True 12.5
b 3 Dima False 9.0
.....
i 2 Kevin False 8.0
j 1 Jonas True 19.0

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']
Expected Output:
Change the name 'James' to \?Suresh\?:
attempts name qualify score

a 1 Anastasia yes 12.5
b 3 Dima no 9.0
.....
i 2 Kevin no 8.0
j 1 Jonas yes 19.0

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']
Expected Output:
Delete the 'attempts' column from the data frame:
name qualify score
a Anastasia yes 12.5
b Dima no 9.0
.....
i Kevin no 8.0
j Jonas yes 19.0

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']

Expected Output:

New DataFrame after inserting the 'color' column
attempts name qualify score color
a 1 Anastasia yes 12.5 Red
b 3 Dima no 9.0 Blue
.....
i 2 Kevin no 8.0 Green
j 1 Jonas yes 19.0 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}]

Expected Output:
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']

Expected Output:
['attempts', 'name', 'qualify', 'score']

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

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 colum1 value == 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.

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
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

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

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
After add one row:
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

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

29. Write a Pandas program to select a row of series/dataframe by given integer index.
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
Index-2: Details
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.
Sample data:
Original DataFrame
attempts name qualify score
0 1 Anastasia yes 12.5
1 3 Dima no 9.0
2 2 Katherine yes 16.5
3 3 James no NaN
4 2 Emily no 9.0
5 3 Michael yes 20.0
6 1 Matthew yes 14.5
7 1 Laura no NaN
8 2 Kevin no 8.0
9 1 Jonas yes 19.0
New DataFrame replacing all NaN with 0:
attempts name qualify score
0 1 Anastasia yes 12.5
1 3 Dima no 9.0
2 2 Katherine yes 16.5
3 3 James no 0.0
4 2 Emily no 9.0
5 3 Michael yes 20.0
6 1 Matthew yes 14.5
7 1 Laura no 0.0
8 2 Kevin no 8.0
9 1 Jonas yes 19.0

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

Original DataFrame

attempts name qualify score
0 1 Anastasia yes 12.5
1 3 Dima no 9.0
2 2 Katherine yes 16.5
3 3 James no NaN
4 2 Emily no 9.0
5 3 Michael yes 20.0
6 1 Matthew yes 14.5
7 1 Laura no NaN
8 2 Kevin no 8.0

9 1 Jonas yes 19.0

After converting index in a column:

index attempts name qualify score
0 0 1 Anastasia yes 12.5
1 1 3 Dima no 9.0
2 2 2 Katherine yes 16.5
3 3 3 James no NaN
4 4 2 Emily no 9.0
5 5 3 Michael yes 20.0
6 6 1 Matthew yes 14.5
7 7 1 Laura no NaN
8 8 2 Kevin no 8.0
9 9 1 Jonas yes 19.0

Hiding index:

index	attempts	name	qualify	score
0	1	Anastasia	yes	12.5
1	3	Dima	no	9.0
2	2	Katherine	yes	16.5
3	3	James	no	NaN
4	2	Emily	no	9.0
5	3	Michael	yes	20.0
6	1	Matthew	yes	14.5
7	1	Laura	no	NaN
8	2	Kevin	no	8.0
9	1	Jonas	yes	19.0