

A. Write a Pandas program 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
```

B. Write a Pandas program 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
```

C. Write a Pandas program 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
```

D. Write a Pandas program 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
```

E. Write a Pandas program 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

F. Write a Pandas program 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']
```

Expected Output:

Rows where score is missing:

```
attempts name qualify score
```

```
d 3 James no NaN
```

```
h 1 Laura no NaN
```

G. Write a Pandas program 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
```

H. Write a Pandas program 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
```

I. Write a Pandas program 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

```

J. Write a Pandas program 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

K. Write a Pandas program 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

L. Write a Pandas program 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

M. Write a Pandas program to sort the DataFrame first by 'name' in descending order, then by 'score' 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']
```

Values for each column will be:

```
name : "Suresh", score: 15.5, attempts: 1, qualify: "yes", label: "k"
```

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, then by 'score' in ascending order:

```
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

N. Write a Pandas program 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 T

rue 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
```

O. Write a Pandas program 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
```

P. Write a Pandas program 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

O. 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

P. Write a Pandas program to count the NaN values in one or more columns in 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

Number of NaN values in one or more columns:

2

Q. Write a Pandas program to reset index in a 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 removing first and second rows

attempts name qualify score

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

Reset the Index:

index attempts name qualify score

0 2 2 Katherine yes 16.5

1 3 3 James no NaN

2 4 2 Emily no 9.0

3 5 3 Michael yes 20.0

4 6 1 Matthew yes 14.5

5 7 1 Laura no NaN

6 8 2 Kevin no 8.0

7 9 1 Jonas yes 19.0

R. Write a Pandas program to sort a given DataFrame by two or more columns.

Sample data:

Original DataFrame:

attempts name qualify score

0 1 Anastasia yes 12.5

1 3 Dima no 9.0

.....

8 2 Kevin no 8.0

9 1 Jonas yes 19.0

Sort the above DataFrame on attempts, name:

attempts name qualify score

0 1 Anastasia yes 12.5

9 1 Jonas yes 19.0

7 1 Laura no NaN

6 1 Matthew yes 14.5

4 2 Emily no 9.0

2 2 Katherine yes 16.5

8 2 Kevin no 8.0

1 3 Dima no 9.0

3 3 James no NaN

5 3 Michael yes 20.0