Experiment No - 4

4. Pandas DataFrames:

Consider 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'].

i)Write a Pandas program to create and display a DataFrame from a specified

dictionary data which has the index labels.

Aim:

Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

Description:

Creating a dataframe using List: DataFrame can be created using a single list or a list **of** lists. Output: Creating DataFrame from dict of ndarray/lists: To create DataFrame from dict of narray/list, all the narray must be of same length. If index is passed then the length index should be equal to the length of arrays.

Program:

import pandas as pd

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']

df=pd.DataFrame(exam\_data,index=labels)

print(df)

Output:

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

ii)Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame.

Aim:

Write a Pandas program to change the name 'James' to 'Suresh' in name column of the DataFrame.

Description:

In pandas the name can be change in name column by using the syntax df['name'] = df['name'].replace('name', 'name').

Program:

import pandas as pd

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']

df=pd.DataFrame(exam\_data,index=labels)

print("Original Output:")

print(df)

print("\n Replace the name 'James' to 'Suresh:")

df['name']=df['name'].replace('James','Suresh')

print(df)

Output:

Original Output:

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

Replace the name 'James' to '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 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

iii)Write a Pandas program to insert a new column in existing DataFrame.

Aim:

Write a Pandas program to insert a new column in existing DataFrame.

Description:

Dataframe indexing. The simplest way to add a new column to an existing panda's data frame is to index the data frame with the new column's name and assign a list to it: ...

Syntax:

df['color']=[colornames]

Program:

import pandas as pd

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']

df=pd.DataFrame(exam\_data,index=labels)

print("\nOriginal Output:")

print(df)

df['color']=['pink', 'red', 'blue', 'yellow', 'purple', 'white', 'black', 'orange', 'green', 'gray']

print("\nAfter inserting a new column in existing DataFrame:")

print(df)

Output:

Original Output:

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

After inserting a new column in existing DataFrame:

name score attempts qualify color

a Anastasia 12.5 1 yes pink

b Dima 9.0 3 no red

c Katherine 16.5 2 yes blue

d James NaN 3 no yellow

e Emily 9.0 2 no purple

f Michael 20.0 3 yes white

g Matthew 14.5 1 yes black

h Laura NaN 1 no orange

i Kevin 8.0 2 no green

j Jonas 19.0 1 yes gray

iv)Write a Pandas program to get list from DataFrame column headers.

Aim:

Write a Pandas program to get list from DataFrame column headers.

Description:

You can get column names in Pandas dataframe using df. columns statement. Usecase: This is useful when you want to show all columns in a dataframe in the output console (E.g. in the jupyter notebook console).

Program:

import pandas as pd

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']

df=pd.DataFrame(exam\_data,index=labels)

print(df)

print(list(df.columns.values))

Output:

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

['name', 'score', 'attempts', 'qualify']