```
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
print(df)
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-bf31c48f66aa> in <cell line: 1>()
----> 1 df=pd.DataFrame(np.random.randn(5,3),index=
['a','c','e','f','h'],columns=['one','two','three'])
      2 print(df)
      3 df=df.reindex(['a','b','c','d','e','f','g','h'])
      4 print(df)

NameError: name 'pd' is not defined
```

```
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df['one'].isnull())
```

```
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: one, dtype: bool
```

```
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
print("Nan replaced with'0':")
print(df.fillna(0))
```

```
        one       two     three
a -0.104325  1.751823  2.260966
b       NaN       NaN       NaN
c  0.733425 -1.948013  0.550886
d       NaN       NaN       NaN
e  0.710048 -2.229589 -1.161677
f  0.290667 -0.527239  0.295639
g       NaN       NaN       NaN
h  1.069846 -0.436860  0.493428
Nan replaced with'0':
        one       two     three
a -0.104325  1.751823  2.260966
b  0.000000  0.000000  0.000000
c  0.733425 -1.948013  0.550886
d  0.000000  0.000000  0.000000
e  0.710048 -2.229589 -1.161677
f  0.290667 -0.527239  0.295639
g  0.000000  0.000000  0.000000
h  1.069846 -0.436860  0.493428
```

```
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
print('----------')
print(df.fillna(method='pad'))
```

```
        one       two     three
a -1.513469  0.276281  1.472510
b       NaN       NaN       NaN
c -0.841726  1.108460  0.377718
d       NaN       NaN       NaN
e  0.963333  1.571720 -0.990790
f  0.055365  0.283036 -0.744800
g       NaN       NaN       NaN
h -0.143672 -0.787751  0.133167
----------
        one       two     three
a -1.513469  0.276281  1.472510
b -1.513469  0.276281  1.472510
c -0.841726  1.108460  0.377718
d -0.841726  1.108460  0.377718
```

```
e  0.963333  1.571720 -0.990790
f  0.055365  0.283036 -0.744800
g  0.055365  0.283036 -0.744800
h -0.143672 -0.787751  0.133167
```

```
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
df=df.reindex(['a','b','c','d','e','f','g','h'])

print('--------------------------------')
print(df.fillna(method='bfill'))
```

```
--------------------------------
        one       two     three
a -0.791002 -1.303789 -1.083806
b  1.030946 -0.091221 -0.273036
c  1.030946 -0.091221 -0.273036
d -0.058735  0.437405 -0.373632
e -0.058735  0.437405 -0.373632
f -0.429095  0.205296 -1.237253
g  0.353874  1.497776 -0.257745
h  0.353874  1.497776 -0.257745
```

```
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two','three'])
print(df)
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
print(df.dropna())
```

```
        one       two     three
a -0.709954 -1.309735 -1.096770
c -1.229664  1.456092 -1.209313
e  0.000633 -0.902380 -1.109012
f -0.433589  0.965473 -0.295473
h -2.834491 -0.585232 -1.827761
        one       two     three
a -0.709954 -1.309735 -1.096770
b       NaN       NaN       NaN
c -1.229664  1.456092 -1.209313
d       NaN       NaN       NaN
e  0.000633 -0.902380 -1.109012
f -0.433589  0.965473 -0.295473
g       NaN       NaN       NaN
h -2.834491 -0.585232 -1.827761
        one       two     three
a -0.709954 -1.309735 -1.096770
c -1.229664  1.456092 -1.209313
e  0.000633 -0.902380 -1.109012
f -0.433589  0.965473 -0.295473
h -2.834491 -0.585232 -1.827761
```

```
df=pd.read_csv("/content/RS_Session_259_AU_994_A_to_G_i.csv")
df.info()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-e7fb3b6f2c4a> in <cell line: 1>()
----> 1 df=pd.read_csv("/content/RS_Session_259_AU_994_A_to_G_i.csv")
      2 df.info()

6 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in
get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,
errors, storage_options)
    854         if ioargs.encoding and "b" not in ioargs.mode:
    855             # Encoding
--> 856             handle = open(
    857                 handle,
    858                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory:
```

```
cols=['Name','Ticket','Cabin']
df=df.drop(cols,axis=1)
df.info()
df.dropna()
```

```
dummies=[]
cols=['Pclass','Sex','Embarked']
for col in cols:
  dummies.append(pd.get_dummies(df[col]))
print(df)
```

```
titanic_dummies=pd.concat(dummies,axis=1)
print(df)
```

```
df=df.drop(['Pclass','Sex','Embarked'],axis=1)
print(df)
```

```
# transfer the ight columns
titanic_dummies=pd.concat(dummies,axis=1)
print(df)
```

```
#concatenante with dataframe
df=pd.concat((df,titanic_dummies),axis=1)
print(df)
```

```
#take care of missing data
df['Age']=df['Age'],interpolate
print(df)
```

```
#normalization
from sklearn.preprocessing import MinMaxScaler
data=[[-1,2],[-0.5,6],[0,10],[1,18]]
scaler=MinMaxScaler()
print(scaler.fit(data))
print('-------')
MinMaxScaler()
print(scaler.data_max_)
print('-------')
print(scaler.transform(data))
```

```
from numpy import asarray
from sklearn.preprocessing import StandardScaler
#define data
data=asarray([[100,0.001],
[8,0.005],
[50,0.005],
[88,0.07],
[4,0.1]])
print(data)
#define standard scaler
scaler=StandardScaler()
#transform data
scaled=scaler.fit_transform(data)
print(scaled)
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.