

1. Write a Pandas program to select the 'name' and 'score' columns from the following DataFrame.

Sample DataFrame:

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']

```
In [1]: import pandas as pd
import numpy as np

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',
                      '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', 'no']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("Select specific columns:")
print(df[['name', 'score']])
```

Select specific columns:

	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

2. 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

```
In [20]: import pandas as pd

d = {'col1': [1, 2, 3], 'col2': [4, 5, 6], 'col3': [7, 8, 9]}
df = pd.DataFrame(data=d)
print("Original DataFrame")
print(df)

df.columns = ['Column1', 'Column2', 'Column3']
df = df.rename(columns={'col1': 'Column1', 'col2': 'Column2', 'col3': 'Column3'})
print("New DataFrame after renaming columns:")
print(df)
```

```
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
```

3. Write a Python function that accepts a string and counts the number of upper and lower case letters.

```
In [3]: def string_test(s):
        d={"UPPER_CASE":0, "LOWER_CASE":0}
        for c in s:
            if c.isupper():
                d["UPPER_CASE"]+=1
            elif c.islower():
                d["LOWER_CASE"]+=1
            else:
                pass
        print ("Original String : ", s)
        print ("No. of Upper case characters : ", d["UPPER_CASE"])
        print ("No. of Lower case Characters : ", d["LOWER_CASE"])
```

```
In [4]: string_test('Python is Amazing')
```

```
Original String : Python is Amazing
No. of Upper case characters : 2
No. of Lower case Characters : 13
```

4. Load the Wooldridge dataset hprice2.

- List 2 characteristics that are statistically positively correlated and 2 that are statistically negatively correlated
- Find the max and min values for all columns
- Display the row where the number of rooms is the highest

```
In [8]: import wooldridge as woo
home_price = woo.dataWoo('hprice2')
home_price
```

```
Out[8]:
```

	price	crime	nox	rooms	dist	radial	proptax	stratio	lowstat	lprice	Inox
0	24000.0	0.006	5.38	6.57	4.09	1	29.600000	15.300000	4.98	10.085809	1.682688
1	21599.0	0.027	4.69	6.42	4.97	2	24.200001	17.799999	9.14	9.980402	1.545433
2	34700.0	0.027	4.69	7.18	4.97	2	24.200001	17.799999	4.03	10.454495	1.545433
3	33400.0	0.032	4.58	7.00	6.06	3	22.200001	18.700001	2.94	10.416311	1.521699
4	36199.0	0.069	4.58	7.15	6.06	3	22.200001	18.700001	5.33	10.496787	1.521699
...
501	22400.0	0.063	5.73	6.59	2.48	1	27.299999	21.000000	9.67	10.016816	1.745715
502	20600.0	0.045	5.73	6.12	2.29	1	27.299999	21.000000	9.08	9.933046	1.745715
503	23899.0	0.061	5.73	6.98	2.17	1	27.299999	21.000000	5.64	10.081592	1.745715
504	22000.0	0.110	5.73	6.79	2.39	1	27.299999	21.000000	6.48	9.998797	1.745715
505	11900.0	0.047	5.73	6.03	2.51	1	27.299999	21.000000	7.88	9.384294	1.745715

506 rows × 12 columns

```
In [10]: home_price.corr()
```

```
Out[10]:
```

	price	crime	nox	rooms	dist	radial	proptax	stratio	lowstat	lprice	Inox
price	1.000000	-0.387919	-0.426037	0.695780	0.249339	-0.380837	-0.467087	-0.503340	-0.726391	0.953320	-0.429447
crime	-0.387919	1.000000	0.421152	-0.218816	-0.379909	0.625442	0.582819	0.288691	0.447033	-0.527495	0.429639
nox	-0.426037	0.421152	1.000000	-0.302828	-0.770222	0.610328	0.666981	0.186863	0.585613	-0.508767	0.993943
rooms	0.695780	-0.218816	-0.302828	1.000000	0.205410	-0.209773	-0.292120	-0.354008	-0.609605	0.632910	-0.304988
dist	0.249339	-0.379909	-0.770222	0.205410	1.000000	-0.495065	-0.534379	-0.229269	-0.495602	0.342008	-0.808795
radial	-0.380837	0.625442	0.610328	-0.209773	-0.495065	1.000000	0.910228	0.464245	0.476038	-0.480972	0.624171
proptax	-0.467087	0.582819	0.666981	-0.292120	-0.534379	0.910228	1.000000	0.454238	0.527624	-0.559671	0.677314
stratio	-0.503340	0.288691	0.186863	-0.354008	-0.229269	0.464245	0.454238	1.000000	0.365402	-0.497635	0.227910
lowstat	-0.726391	0.447033	0.585613	-0.609605	-0.495602	0.476038	0.527624	0.365402	1.000000	-0.497635	0.227910
lprice	0.953320	-0.527495	-0.508767	0.632910	0.342008	-0.480972	-0.559671	-0.497635	-0.497635	1.000000	-0.429447
Inox	-0.429447	0.429639	0.993943	-0.304988	-0.808795	0.624171	0.677314	0.227910	0.227910	-0.429447	1.000000
lproptax	-0.473166	0.551393	0.659206	-0.299269	-0.513897	0.863531	0.988586	0.422231	0.422231	-0.473166	0.551393

```
In [18]: print(home_price.max())
```

```
price      50001.000000
crime      88.975998
nox        8.710000
rooms      8.780000
dist       12.130000
radial     24.000000
proptax    71.099998
stratio    22.000000
lowstat    39.070000
lprice     10.819798
lnox       2.164472
lproptax   6.566672
dtype: float64
```

```
In [19]: print(home_price.min())
```

```
price      5000.000000
crime      0.006000
nox        3.850000
rooms      3.560000
dist       1.130000
radial     1.000000
proptax    18.700001
stratio    12.600000
lowstat    1.730000
lprice     8.517193
lnox       1.348073
lproptax   5.231109
dtype: float64
```

```
In [17]: print(home_price[home_price.rooms == home_price.rooms.max()])
```

```
      price  crime  nox  rooms  dist  radial  proptax  stratio  low
stat  \
364  21900.0  3.474  7.18   8.78   1.9      24  66.599998  20.200001
5.29

      lprice      lnox  lproptax
364   9.994242  1.971299   6.50129
```