

```
import pandas as pd
import numpy as np
```

```
# 2. Masala. https://www.kaggle.com/datasets
# shu link orqali FIFA Jahon chempionati 2022 o'yin ma'lumotlarini
# yuklab oling va team ustunini indexlariga o'tkazing bundan
# tashqari ma'lumotlarni qayta ishlovchi mutaxassis Ingliz tilini bilmasligi
# sababli bu DataSet ni ustunlarini va qatorlari sonini aniqlab o'ning ustunlarini
# nomini o'zbek tiliga tarjima qilib qayta nomlang va o'zingizning logikangizdan
# kelib chiqib DataSetga ishlov bering (qanchalik yaxshi ishlov berilganligiga
# qarab baxoni yaxshi olishingiz mumkun.)
```

```
df=pd.read_csv('group_stats.csv')
df
```

	group	rank	team	matches_played	wins	draws	losses	goals_scored	goals_against	goal_difference	points	expected_goa
0	1	1	Netherlands	3	2	1	0	5	1	4	7	
1	1	2	Senegal	3	2	0	1	5	4	1	6	
2	1	3	Ecuador	3	1	1	1	4	3	1	4	
3	1	4	Qatar	3	0	0	3	1	7	-6	0	
4	2	1	England	3	2	1	0	9	2	7	7	
5	2	2	United States	3	1	2	0	2	1	1	5	
6	2	3	IR Iran	3	1	0	2	4	7	-3	3	
Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. Показать различия											-5	1
8	3	1	Argentina	3	2	0	1	5	2	3	6	
9	3	2	Poland	3	1	1	1	2	2	0	4	
10	3	3	Mexico	3	1	1	1	2	3	-1	4	
11	3	4	Saudi Arabia	3	1	0	2	3	5	-2	3	
12	4	1	France	3	2	0	1	6	3	3	6	
13	4	2	Australia	3	2	0	1	3	4	-1	6	
14	4	3	Tunisia	3	1	1	1	1	1	0	4	
15	4	4	Denmark	3	0	1	2	1	3	-2	1	
16	5	1	Japan	3	2	0	1	4	3	1	6	
17	5	2	Spain	3	1	1	1	9	3	6	4	
18	5	3	Germany	3	1	1	1	6	5	1	4	
19	5	4	Costa Rica	3	1	0	2	3	11	-8	3	
20	6	1	Morocco	3	2	1	0	4	1	3	7	
21	6	2	Croatia	3	1	2	0	4	1	3	5	
22	6	3	Belgium	3	1	1	1	1	2	-1	4	
23	6	4	Canada	3	0	0	3	2	7	-5	0	
24	7	1	Brazil	3	2	0	1	3	1	2	6	
25	7	2	Switzerland	3	2	0	1	4	3	1	6	
26	7	3	Cameroon	3	1	1	1	4	4	0	4	
27	7	4	Serbia	3	0	1	2	5	8	-3	1	
28	8	1	Portugal	3	2	0	1	6	4	2	6	
29	8	2	Korea Republic	3	1	1	1	4	4	0	4	
30	8	3	Uruguay	3	1	1	1	2	2	0	4	
31	8	4	Ghana	3	1	0	2	5	7	-2	3	



pip install translators

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting translators
  Downloading translators-5.5.5-py3-none-any.whl (32 kB)
Collecting requests>=2.28.1
  Downloading requests-2.28.1-py3-none-any.whl (62 kB)
    |██████████| 62 kB 1.1 MB/s
Collecting pathos>=0.2.9
  Downloading pathos-0.3.0-py3-none-any.whl (79 kB)
    |██████████| 79 kB 7.2 MB/s
Collecting cryptography>=38.0.1
  Downloading cryptography-38.0.4-cp36-abi3-manylinux_2_24_x86_64.whl (4.0 MB)
    |██████████| 4.0 MB 52.6 MB/s
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.8/dist-packages (from translators) (4.9.1)
Collecting PyExecJS>=1.5.1
  Downloading PyExecJS-1.5.1.tar.gz (13 kB)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.8/dist-packages (from cryptography>=38.0.1->translators) (1.15)
Requirement already satisfied: pycparser in /usr/local/lib/python3.8/dist-packages (from cffi>=1.12->cryptography>=38.0.1->translators) (2.21)
Requirement already satisfied: dill>=0.3.6 in /usr/local/lib/python3.8/dist-packages (from pathos>=0.2.9->translators) (0.3.6)
Collecting multiprocessing>=0.70.14
  Downloading multiprocessing-0.70.14-py38-none-any.whl (132 kB)
    |██████████| 132 kB 49.3 MB/s
Collecting ppft>=1.7.6.6
  Downloading ppft-1.7.6.6-py3-none-any.whl (52 kB)
    |██████████| 52 kB 1.1 MB/s
Collecting pox>=0.3.2
  Downloading pox-0.3.2-py3-none-any.whl (29 kB)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.8/dist-packages (from PyExecJS>=1.5.1->translators) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.28.1->translators) (2.10)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.28.1->translators) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.28.1->translators) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.28.1->translators) (2022.12.7)
Building wheels for collected packages: PyExecJS
  Building wheel for PyExecJS (setup.py) ... done
  Created wheel for PyExecJS: filename=PyExecJS-1.5.1-py3-none-any.whl size=14596 sha256=16b0bebdedcb6f4939aa3308d154eb6b131c26dc8a
    4
Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. Показать различия
  Attempting uninstall: requests
    Found existing installation: requests 2.23.0
    Uninstalling requests-2.23.0:
      Successfully uninstalled requests-2.23.0
Successfully installed PyExecJS-1.5.1 cryptography-38.0.4 multiprocessing-0.70.14 pathos-0.3.0 pox-0.3.2 ppft-1.7.6.6 requests-2.28.1

```

```

from pandas import DataFrame
from pandas import Series

```

```
import translators as ts
```

```

def trans(s):
    return ts.translate_text(s,to_language='uz')

```

```
Using state Iowa server backend.
```

```

df.drop(df.columns[0],axis=1,inplace=True)
df = df.rename_axis('jamoat')
df.set_axis([trans(str(i).replace('_', ' ').replace('exp','expected')) for i in df.columns], axis=1)
df.columns

```

```

Index(['team', 'matches_played', 'wins', 'draws', 'losses', 'goals_scored',
       'goals_against', 'goal_difference', 'points', 'expected_goal_scored',
       'exp_goal_conceded', 'exp_goal_difference',
       'exp_goal_difference_per_90'],
      dtype='object')

```

```
df
```

	team	matches_played	wins	draws	losses	goals_scored	goals_against	goal_difference	points	expected_goal_scored
jamao										
0	Netherlands	3	2	1	0	5	1	4	7	2.4
1	Senegal	3	2	0	1	5	4	1	6	3.8
2	Ecuador	3	1	1	1	4	3	1	4	3.7
3	Qatar	3	0	0	3	1	7	-6	0	1.4
4	England	3	2	1	0	9	2	7	7	5.2
5	United States	3	1	2	0	2	1	1	5	2.6
6	IR Iran	3	1	0	2	4	7	-3	3	3.2
7	Wales	3	0	1	2	1	6	-5	1	2.7
8	Argentina	3	2	0	1	5	2	3	6	6.0
9	Poland	3	1	1	1	2	2	0	4	2.8
10	Mexico	3	1	1	1	2	3	-1	4	3.3
11	Saudi Arabia	3	1	0	2	3	5	-2	3	3.0
12	France	3	2	0	1	6	3	3	6	7.2
13	Australia	3	2	0	1	3	4	-1	6	1.7
14	Tunisia	3	1	1	1	1	1	0	4	2.3
15	Denmark	3	0	1	2	1	3	-2	1	2.7
Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. Показать различия										6
17	Spain	3	1	1	1	9	3	6	4	5.2
18	Germany	3	1	1	1	6	5	1	4	10.1
19	Costa Rica	3	1	0	2	3	11	-8	3	1.4
20	Morocco	3	2	1	0	4	1	3	7	2.4
21	Croatia	3	1	2	0	4	1	3	5	3.8
22	Belgium	3	1	1	1	1	2	-1	4	4.7
23	Canada	3	0	0	3	2	7	-5	0	4.2
24	Brazil	3	2	0	1	3	1	2	6	5.9

df.describe()

	matches_played	wins	draws	losses	goals_scored	goals_against	goal_difference	points	expected_goal_scored
count	32.0	32.000000	32.000000	32.000000	32.000000	32.00000	32.000000	32.000000	32.000000
mean	3.0	1.187500	0.625000	1.187500	3.750000	3.75000	0.000000	4.187500	3.850000
std	0.0	0.692704	0.609071	0.780302	2.079082	2.44949	3.282407	1.990907	1.750944
min	3.0	0.000000	0.000000	0.000000	1.000000	1.00000	-8.000000	0.000000	1.400000
25%	3.0	1.000000	0.000000	1.000000	2.000000	2.00000	-2.000000	3.000000	2.700000
50%	3.0	1.000000	1.000000	1.000000	4.000000	3.00000	0.000000	4.000000	3.700000
75%	3.0	2.000000	1.000000	2.000000	5.000000	5.00000	2.000000	6.000000	4.450000
max	3.0	2.000000	2.000000	3.000000	9.000000	11.00000	7.000000	7.000000	10.100000



```
bins=[0,3,6,10]
qiymat=['kam','ortacha','kop']
guruh=pd.cut(df.goals_scored,bins,labels=qiymat)
df['gol_reytingi']=guruh
df
```

3	Qatar	3	0	0	3	1	7	-5	0	1.5
4	England	3	2	1	0	9	2	7	7	5.5
5	United States	3	1	2	0	2	1	1	5	2.0
6	IR Iran	3	1	0	2	4	7	-3	3	3.5
7	Wales	3	0	1	2	1	6	-5	1	2.5
8	Argentina	3	2	0	1	5	2	3	6	6.0
9	Poland	3	1	1	1	2	2	0	4	2.5
10	Mexico	3	1	1	1	2	3	-1	4	3.5
11	Saudi Arabia	3	1	0	2	3	5	-2	3	3.0
12	France	3	2	0	1	6	3	3	6	7.5
13	Australia	3	2	0	1	3	4	-1	6	1.5
14	Tunisia	3	1	1	1	1	1	0	4	2.5
15	Denmark	3	0	1	2	1	3	-2	1	2.5
16	Japan	3	2	0	1	4	3	1	6	3.5
17	Spain	3	1	1	1	9	3	6	4	5.5
18	Germany	3	1	1	1	6	5	1	4	10.5
19	Costa Rica	3	1	0	2	3	11	-8	3	1.5
20	Morocco	3	2	1	0	4	1	3	7	2.5
21	Croatia	3	1	0	2	4	1	3	5	3.5
22	Belgium	3	1	0	2	4	1	3	4	4.5
23	Canada	3	0	0	3	2	7	-5	0	4.5
24	Brazil	3	2	0	1	3	1	2	6	5.5
25	Switzerland	3	2	0	1	4	3	1	6	4.5
26	Cameroon	3	1	1	1	4	4	0	4	3.5
27	Serbia	3	0	1	2	5	8	-3	1	3.5
28	Portugal	3	2	0	1	6	4	2	6	4.5
29	Korea Republic	3	1	1	1	4	4	0	4	4.5
30	Uruguay	3	1	1	1	2	2	0	4	3.5
31	Ghana	3	1	0	2	5	7	-2	3	4.5

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. [Показать различия](#)

```
bins=[0,4,8,12]
qiymat=['kam','ortacha','kop']
guruh=pd.cut(df.goals_against,bins,labels=qiymat)
guruh
df['avtogol_reytingi']=guruh
df
```

3	Qatar	3	0	0	3	1	7	-5	0	1.5
4	England	3	2	1	0	9	2	7	7	5.0
5	United States	3	1	2	0	2	1	1	5	2.0
6	IR Iran	3	1	0	2	4	7	-3	3	3.0
7	Wales	3	0	1	2	1	6	-5	1	2.0
8	Argentina	3	2	0	1	5	2	3	6	6.0
9	Poland	3	1	1	1	2	2	0	4	2.0
10	Mexico	3	1	1	1	2	3	-1	4	3.0
11	Saudi Arabia	3	1	0	2	3	5	-2	3	3.0
12	France	3	2	0	1	6	3	3	6	7.0
13	Australia	3	2	0	1	3	4	-1	6	1.0
14	Tunisia	3	1	1	1	1	1	0	4	2.0
15	Denmark	3	0	1	2	1	3	-2	1	2.0
16	Japan	3	2	0	1	4	3	1	6	3.0
17	Spain	3	1	1	1	9	3	6	4	5.0
18	Germany	3	1	1	1	6	5	1	4	10.0
19	Costa Rica	3	1	0	2	3	11	-8	3	1.0
20	Morocco	3	2	1	0	4	1	3	7	2.0
21	Croatia	3	1	0	0	1	1	0	5	3.0
22	Belgium	3	1	1	1	1	1	0	4	4.0
23	Canada	3	0	0	3	2	7	-5	0	4.0
24	Brazil	3	2	0	1	3	1	2	6	5.0
25	Switzerland	3	2	0	1	4	3	1	6	4.0
26	Cameroon	3	1	1	1	4	4	0	4	3.0
27	Serbia	3	0	1	2	5	8	-3	1	3.0
28	Portugal	3	2	0	1	6	4	2	6	4.0
29	Korea Republic	3	1	1	1	4	4	0	4	4.0

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. [Показать различия](#)

```
bins=[0,1,2,3]
qiymat=['kam','ortacha','kop']
guruh=pd.cut(df.goals_against,bins,labels=qiymat)
guruh
df['yutish_reytingi']=guruh
df
```

3	Qatar	3	0	0	0	1	7	-6	0	1.5
4	England	3	2	1	0	9	2	7	7	5.5
5	United States	3	1	2	0	2	1	1	5	2.0
6	IR Iran	3	1	0	2	4	7	-3	3	3.5
7	Wales	3	0	1	2	1	6	-5	1	2.5
8	Argentina	3	2	0	1	5	2	3	6	6.0
9	Poland	3	1	1	1	2	2	0	4	2.5
10	Mexico	3	1	1	1	2	3	-1	4	3.5
11	Saudi Arabia	3	1	0	2	3	5	-2	3	3.0
12	France	3	2	0	1	6	3	3	6	7.5
13	Australia	3	2	0	1	3	4	-1	6	1.5
14	Tunisia	3	1	1	1	1	1	0	4	2.5
15	Denmark	3	0	1	2	1	3	-2	1	2.5
16	Japan	3	2	0	1	4	3	1	6	3.5
17	Spain	3	1	1	1	9	3	6	4	5.5
18	Germany	3	1	1	1	6	5	1	4	10.5
19	Costa Rica	3	1	0	2	3	11	-8	3	1.5
20	Morocco	3	2	1	0	4	1	3	7	2.5
21	Switzerland	3	1	0	0	1	1	0	5	3.5

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. [Показать различия](#)

```
df['reyting']=(((df.yutish_reytingi=='kop') | (df.yutish_reytingi=='ortacha'))
               & (df.avtogol_reytingi=='kam')
               & ((df.gol_reytingi=='kop') | (df.gol_reytingi=='ortacha'))))
df
df[df.reyting==True]
print(df[df.reyting==True].team+" jamoaning yutish ehtimoli katta")
```

```
jamoaa
2      Ecuador jamoaning yutish ehtimoli katta
4      England jamoaning yutish ehtimoli katta
8      Argentina jamoaning yutish ehtimoli katta
12     France jamoaning yutish ehtimoli katta
16     Japan jamoaning yutish ehtimoli katta
17     Spain jamoaning yutish ehtimoli katta
25     Switzerland jamoaning yutish ehtimoli katta
Name: team, dtype: object
```

```
df.count(axis=0)
df.count(axis=1)
```

```
jamoaa
0      17
1      16
2      17
3      16
4      17
5      17
6      16
7      16
8      17
9      17
10     17
11     16
12     17
13     16
14     17
15     17
16     17
17     17
18     16
19     16
20     17
21     17
22     17
23     16
24     17
25     17
26     16
27     16
28     16
```

```
29 16
30 17
31 16
dtype: int64
```

```
#3-masala
# Masala. 1-masalada yaratgan repositoryyangizni yuklab oling va
# uning ichidagi excel va csv fayllaridan bittadan yuklab oling va ularga ishlov
# bering (index, ustunlarini tartiblang) va bu DataSetlarni HDF5 obyektiga birma
# bir .h5 fayl kengaytmasida saqlang va obyektini yoping.
```

```
!git clone https://github.com/ShuhratGaniyev/shuhrat_file

Cloning into 'shuhrat_file'...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 43 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (43/43), done.
```

```
excel=pd.read_excel('shuhrat_file/uzbekistan.xlsx')
excel
```

	Boʻlinishi	Maydoni (kv.km)	Aholisi
0	Andijon viloyati	4200	1899000
1	Buxoro viloyati	39400	1384700
2	Farg'ona viloyati	6800	2597000
3	Jizzax viloyati	20500	910500
4	Xorazm viloyati	6300	1200000
5	Namangan viloyati	7900	1862000
6	Navoiy viloyati	110800	767500
7	Qashqadaryo viloyati	28400	2029000
8	Qoraqalpogʻiston Respublikasi	160000	1200000
9	Samarqand viloyati	16400	2322000
10	Sirdaryo viloyati	5100	648100
11	Surxondaryo viloyati	20800	1676000
12	Toshkent viloyati	15300	4450000

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке.

Показать различия

5	Namangan viloyati	7900	1862000
6	Navoiy viloyati	110800	767500
7	Qashqadaryo viloyati	28400	2029000
8	Qoraqalpogʻiston Respublikasi	160000	1200000
9	Samarqand viloyati	16400	2322000
10	Sirdaryo viloyati	5100	648100
11	Surxondaryo viloyati	20800	1676000
12	Toshkent viloyati	15300	4450000

```
excel.sort_index(ascending=True,axis=1,inplace=True)

excel
```

	Aholisi	Boʻlinishi	Maydoni (kv.km)
0	1899000	Andijon viloyati	4200
1	1384700	Buxoro viloyati	39400
2	2597000	Farg'ona viloyati	6800
3	910500	Jizzax viloyati	20500
4	1200000	Xorazm viloyati	6300
5	1862000	Namangan viloyati	7900
6	767500	Navoiy viloyati	110800
7	2029000	Qashqadaryo viloyati	28400
8	1200000	Qoraqalpogʻiston Respublikasi	160000
9	2322000	Samarqand viloyati	16400
10	648100	Sirdaryo viloyati	5100
11	1676000	Surxondaryo viloyati	20800
12	4450000	Toshkent viloyati	15300

```
excel.sort_index(ascending=False,axis=0,inplace=True)
excel
```

	Aholisi	Bo‘linishi	Maydoni (kv.km)	
12	4450000	Toshkent viloyati	15300	
11	1676000	Surxondaryo viloyati	20800	
10	648100	Sirdaryo viloyati	5100	
9	2322000	Samarqand viloyati	16400	
8	1200000	Qoraqalpog‘iston Respublikasi	160000	
7	2029000	Qashqadaryo viloyati	28400	
6	767500	Navoiy viloyati	110800	
5	1862000	Namangan viloyati	7900	
4	1200000	Xorazm viloyati	6300	
3	910500	Jizzax viloyati	20500	
2	2597000	Farg‘ona viloyati	6800	
1	1384700	Buxoro viloyati	39400	

```
excel.sort_values(by=['Aholisi', 'Bo‘linishi'], inplace=True)
```

```
hdf_obj=pd.HDFStore("shuhrat_file.h5")
hdf_obj.put("uzbekistan_info", excel)
hdf_obj.close()
```

```
csv=pd.read_csv('shuhrat_file/world_population.csv')
```

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. [Показать различия](#)

	rank	country	density	densityMi	pop2021	area	
0	1	Macau	21946.4667	56841.3487	658.394	30	
1	2	Monaco	19755.5000	51166.7450	39.511	2	
2	3	Singapore	8305.1915	21510.4461	5896.686	710	
3	4	Hong Kong	6841.3134	17719.0017	7552.810	1104	
4	5	Gibraltar	5616.3333	14546.3033	33.698	6	
...	
227	228	Namibia	3.1338	8.1166	2587.344	825615	
228	229	Western Sahara	2.3003	5.9577	611.875	266000	
229	230	Mongolia	2.1286	5.5129	3329.289	1564110	
230	231	Falkland Islands	0.2902	0.7517	3.533	12173	
231	232	Greenland	0.0263	0.0680	56.877	2166086	

232 rows × 6 columns

```
csv.sort_index(ascending=True,axis=1,inplace=True)
csv
```

	area	country	density	densityMi	pop2021	rank	
0	30	Macau	21946.4667	56841.3487	658.394	1	
1	2	Monaco	19755.5000	51166.7450	39.511	2	
2	710	Singapore	8305.1915	21510.4461	5896.686	3	
3	1104	Hong Kong	6841.3134	17719.0017	7552.810	4	
4	6	Gibraltar	5616.3333	14546.3033	33.698	5	
...	
227	825615	Namibia	3.1338	8.1166	2587.344	228	
228	266000	Western Sahara	2.3003	5.9577	611.875	229	
229	1564110	Mongolia	2.1286	5.5129	3329.289	230	
230	12173	Falkland Islands	0.2902	0.7517	3.533	231	
231	2166086	Greenland	0.0263	0.0680	56.877	232	

232 rows × 6 columns


```
csv.sort_index(ascending=False,axis=0,inplace=True)
csv
```

	area	country	density	densityMi	pop2021	rank
231	2166086	Greenland	0.0263	0.0680	56.877	232
230	12173	Falkland Islands	0.2902	0.7517	3.533	231
229	1564110	Mongolia	2.1286	5.5129	3329.289	230
228	266000	Western Sahara	2.3003	5.9577	611.875	229
227	825615	Namibia	3.1338	8.1166	2587.344	228
...
4	6	Gibraltar	5616.3333	14546.3033	33.698	5
3	1104	Hong Kong	6841.3134	17719.0017	7552.810	4
2	710	Singapore	8305.1915	21510.4461	5896.686	3
1	2	Monaco	19755.5000	51166.7450	39.511	2
0	30	Macau	21946.4667	56841.3487	658.394	1

232 rows × 6 columns

```
csv.sort_values(by=['country','density','densityMi'],inplace=True)
```

```
hdf_obj=pd.HDFStore("shuhrat_file.h5")
hdf_obj.put("population_info",csv)
hdf_obj.close()
```

Ошибка автоматического сохранения. Этот файл был обновлен удаленно или на другой вкладке. [Показать различия](#)

```
# 4-mazda raqamda yaratgan filega kengaytish bo'yicha qiziq statementlar shuqirib oling
# va DataFrame ning ichidagi takrorlanuvchi va keraksiz malumotlarni olib tashlang
# eng asosiy ustunni tanlab uni indexlariga o'tkazing va qayta saqlang.
```

```
hdf_obj=pd.HDFStore('shuhrat_file.h5')
hdf_obj.keys()

['/population_info', '/uzbekistan_info']
```

```
df3=hdf_obj["population_info"]
df3
```

	area	country	density	densityMi	pop2021	rank
148	652230	Afghanistan	61.0757	158.1862	39835.428	149
110	28748	Albania	99.9351	258.8318	2872.933	111
197	2381741	Algeria	18.7328	48.5179	44616.624	198
49	199	American Samoa	276.8844	717.1307	55.100	50
77	468	Andorra	165.2885	428.0971	77.355	78
...
130	142	Wallis and Futuna	78.1268	202.3483	11.094	131
228	266000	Western Sahara	2.3003	5.9577	611.875	229
150	527968	Yemen	57.7509	149.5749	30490.640	151
187	752612	Zambia	25.1400	65.1125	18920.651	188
172	390757	Zimbabwe	38.6229	100.0333	15092.171	173

232 rows × 6 columns

```
d=df3[['country','area'][:1]]
d
```