

به نام خدا



دانشگاه صنعتی شاهرود
Shahrood University of Technology

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عنوان تمرین: آشنایی با کتابخانه های پایه

استاد: دکتر مرضیه رحیمی

نویسنده: مریم درویشیان

تاریخ: 1401/12/12

در این قسمت دیتاست بارگذاری شده و از محتوای آن خروجی گرفتیم.

The screenshot shows the JupyterLab interface with a file browser on the left and a code editor on the right. The code editor contains the following Python code:

```
[ ]: #بارگذاری دیتاست در پایتون-1
[1]: In [1]: import pandas as pd
[4]: In [2]: dataset = pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
[5]: data
```

The output of the code is a table with 7 columns: city, lat, lng, iso2, density, population, and ranking. The table shows data for various cities, including Saint-Oblas, Louresse, Olmet, Gottenhouse, La Rochette, and Saint-Eutrope.

	city	lat	lng	iso2	density	population	ranking
0	Saint-Oblas	45.5674	5.0447	FR	129.2	NaN	4
1	Louresse	47.2394	-0.3136	FR	NaN	872.0	3
2	Olmet	45.7100	3.6614	FR	10.4	161.0	3
3	Olmet	44.9542	2.6108	FR	71.0	NaN	4
4	Gottenhouse	48.7208	7.3611	FR	305.6	382.0	3
...
59059	La Rochette	45.2609	6.2881	FR	55.5	NaN	4
59060	La Rochette	45.3056	3.4747	FR	14.4	NaN	4
59061	Saint-Eutrope	45.4181	0.1114	FR	62.9	168.0	3
59062	Saint-Eutrope	44.4535	0.5204	FR	34.2	NaN	4
59063	Taillis	48.1889	-1.2389	FR	81.2	996.0	3

The output is summarized as 59064 rows x 7 columns.

خروجی هشت سطر اول دیتاست:

The screenshot shows the JupyterLab interface with a file browser on the left and a code editor on the right. The code editor contains the following Python code:

```
[ ]: dataset.head(8)
[6]:
```

The output of the code is a table with 8 rows and 7 columns: city, lat, lng, iso2, density, population, and ranking. The table shows data for Saint-Oblas, Louresse, Olmet, Gottenhouse, Bergues, Villelongue, and Saint-Martin-d'Arrossa.

	city	lat	lng	iso2	density	population	ranking
0	Saint-Oblas	45.5674	5.0447	FR	129.2	NaN	4
1	Louresse	47.2394	-0.3136	FR	NaN	872.0	3
2	Olmet	45.7100	3.6614	FR	10.4	161.0	3
3	Olmet	44.9542	2.6108	FR	71.0	NaN	4
4	Gottenhouse	48.7208	7.3611	FR	305.6	382.0	3
5	Bergues	50.9686	2.4342	FR	2755.3	3637.0	2
6	Villelongue	45.8647	2.8817	FR	33.0	NaN	4
7	Saint-Martin-d'Arrossa	43.2381	-1.3133	FR	29.2	538.0	3

The code also includes a line to display the data types of the dataset:

```
[7]: dataset.dtypes
```

The output shows the data types for each column: city (object), lat (float64), lng (float64), iso2 (object), density (float64), population (float64), and ranking (int64).

خروجی اطلاعاتی راجع به دیتاست از قبیل تعداد ستونها و نوع آنها و مقدار حافظه مصرفی:

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The code editor contains the following code:

```
[8]: dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59064 entries, 0 to 59063
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype  
---  -
0   city         59064 non-null  object  
1   lat          59064 non-null  float64  
2   lng          59064 non-null  float64  
3   iso2         59064 non-null  object  
4   density      59063 non-null  float64  
5   population   36091 non-null  float64  
6   ranking      59064 non-null  int64  
dtypes: float64(4), int64(1), object(2)
memory usage: 3.2+ MB

[9]: dataset.shape

[9]: (59064, 7)

[10]: datasets=dataset.drop_duplicates(subsets=['city'],keep='last')

[11]: dataset
```

The output of the code is displayed in the right pane:

```
[11]:
```

	city	lat	lng	iso2	density	population	ranking
0	Saint-Oblas	45.5674	5.0447	FR	129.2	NaN	4
1	Louresse	47.2394	-0.3136	FR	NaN	872.0	3
4	Gottenhouse	48.7208	7.3611	FR	305.6	382.0	3
...

تعداد و نوع ستونها با دستور Shape :

The screenshot shows a JupyterLab environment with a file browser on the left and a code editor on the right. The code editor contains the following code:

```
4 density      59063 non-null  float64  
5 population   36091 non-null  float64  
6 ranking      59064 non-null  int64  
dtypes: float64(4), int64(1), object(2)
memory usage: 3.2+ MB

[9]: dataset.shape

[9]: (59064, 7)

[10]: datasets=dataset.drop_duplicates(subsets=['city'],keep='last')

[11]: dataset
```

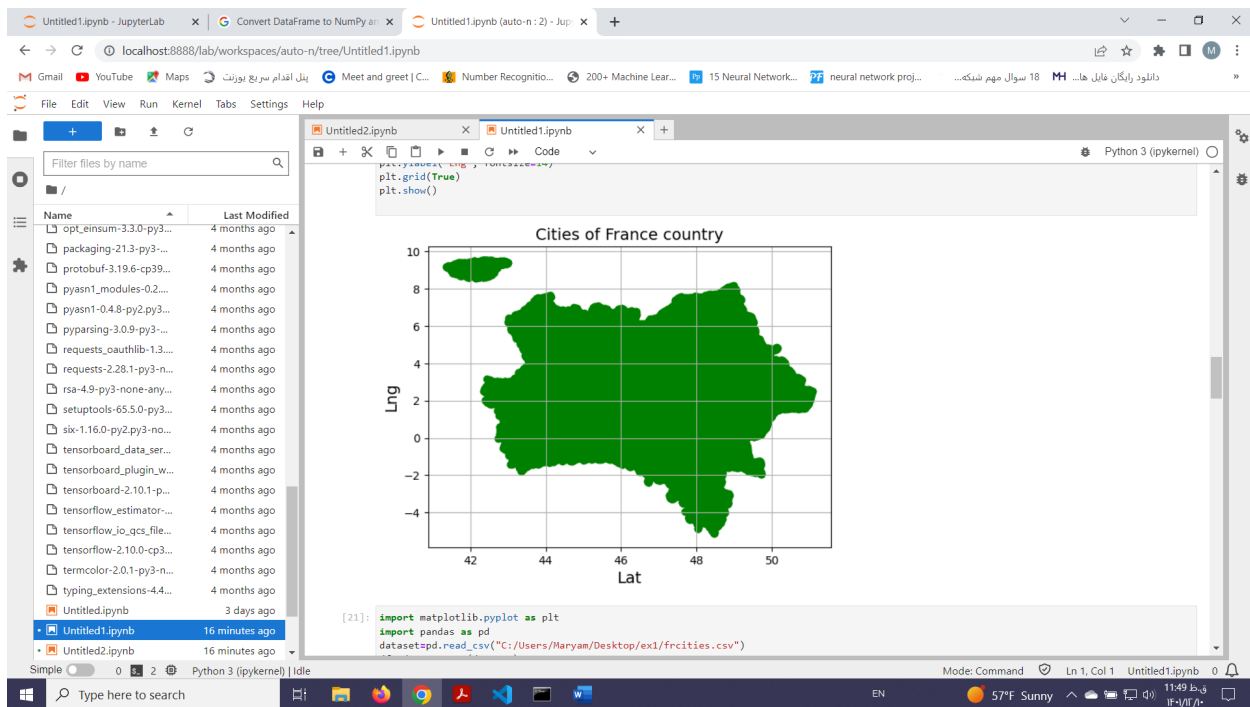
The output of the code is displayed in the right pane:

```
[11]:
```

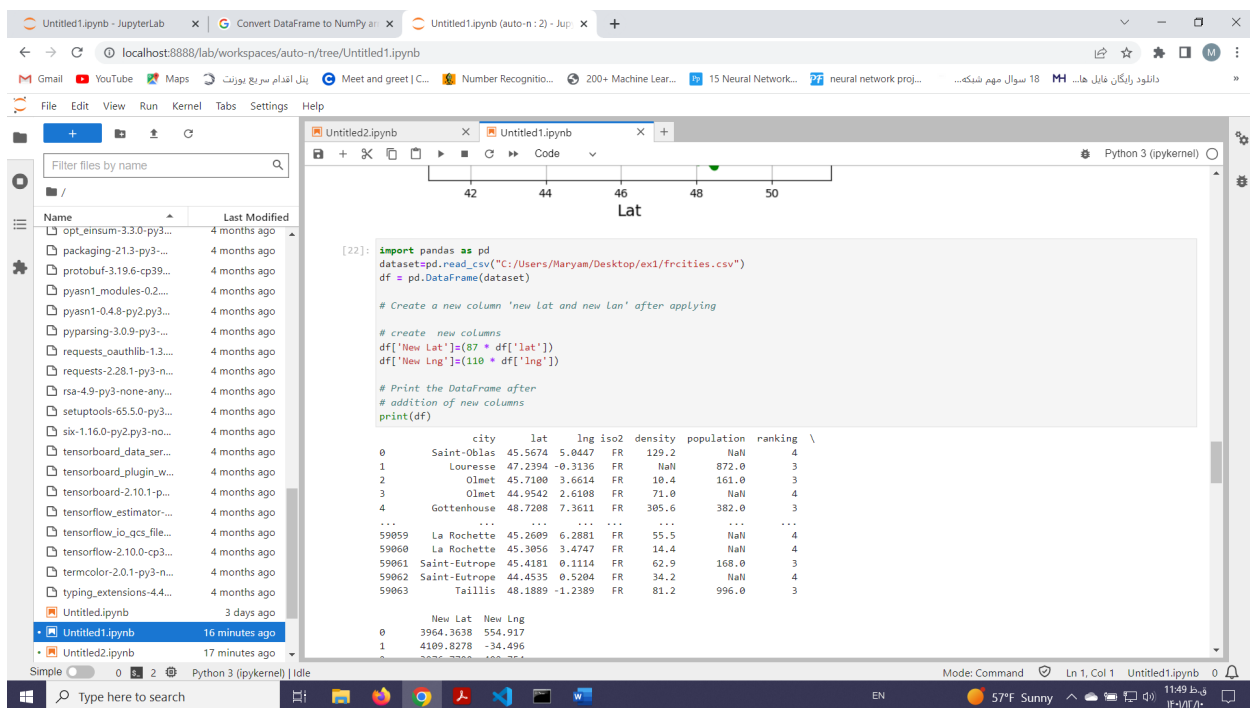
	city	lat	lng	iso2	density	population	ranking
0	Saint-Oblas	45.5674	5.0447	FR	129.2	NaN	4
1	Louresse	47.2394	-0.3136	FR	NaN	872.0	3
4	Gottenhouse	48.7208	7.3611	FR	305.6	382.0	3
5	Bergues	50.9686	2.4342	FR	2755.3	3637.0	2
7	Saint-Martin-d'Arrossa	43.2381	-1.3133	FR	29.2	538.0	3
...
59055	Neuvecelle	46.3950	6.6125	FR	769.0	3076.0	3
59056	Cognocoli-Monticchi	41.8283	8.9058	FR	4.8	171.0	3
59060	La Rochette	45.3056	3.4747	FR	14.4	NaN	4
59062	Saint-Eutrope	44.4535	0.5204	FR	34.2	NaN	4

حذف سطرهایی با نام شهر تکراری و کاهش سطرها از 59063 به 50716 سطر و بخاطر وجود last آخرین سطر duplicate را نگه میدارد:

استفاده از کتابخانه Matplotlib برای رسم نمودار دیتا ست بر اساس Lat و Lng :



ایجاد دو ستون جدید برای Lat , Lng بر حسب کیلومتر:



نمایش ستونهای جدید به اضافه قبلیها:

The screenshot shows the JupyterLab interface with a file explorer on the left and a code editor on the right. The code editor displays a DataFrame with the following columns: city, lat, lng, iso2, density, population, and ranking. The data is as follows:

	city	lat	lng	iso2	density	population	ranking
0	Saint-Obas	45.5674	5.0447	FR	129.2	NaN	4
1	Louresse	47.2394	-0.3136	FR	NaN	872.0	3
2	Olmet	45.7100	3.6614	FR	10.4	161.0	3
3	Olmet	44.9542	2.6108	FR	71.0	NaN	4
4	Gottenhouse	48.7208	7.3611	FR	305.6	382.0	3
...
59059	La Rochette	45.2609	6.2881	FR	55.5	NaN	4
59060	La Rochette	45.3056	3.4747	FR	14.4	NaN	4
59061	Saint-Eutrope	45.4181	0.1114	FR	62.9	168.0	3
59062	Saint-Eutrope	44.4535	0.5204	FR	34.2	NaN	4
59063	Taillis	48.1889	-1.2389	FR	81.2	996.0	3

The code editor also shows the following code:

```
print(df)

[59064 rows x 9 columns]

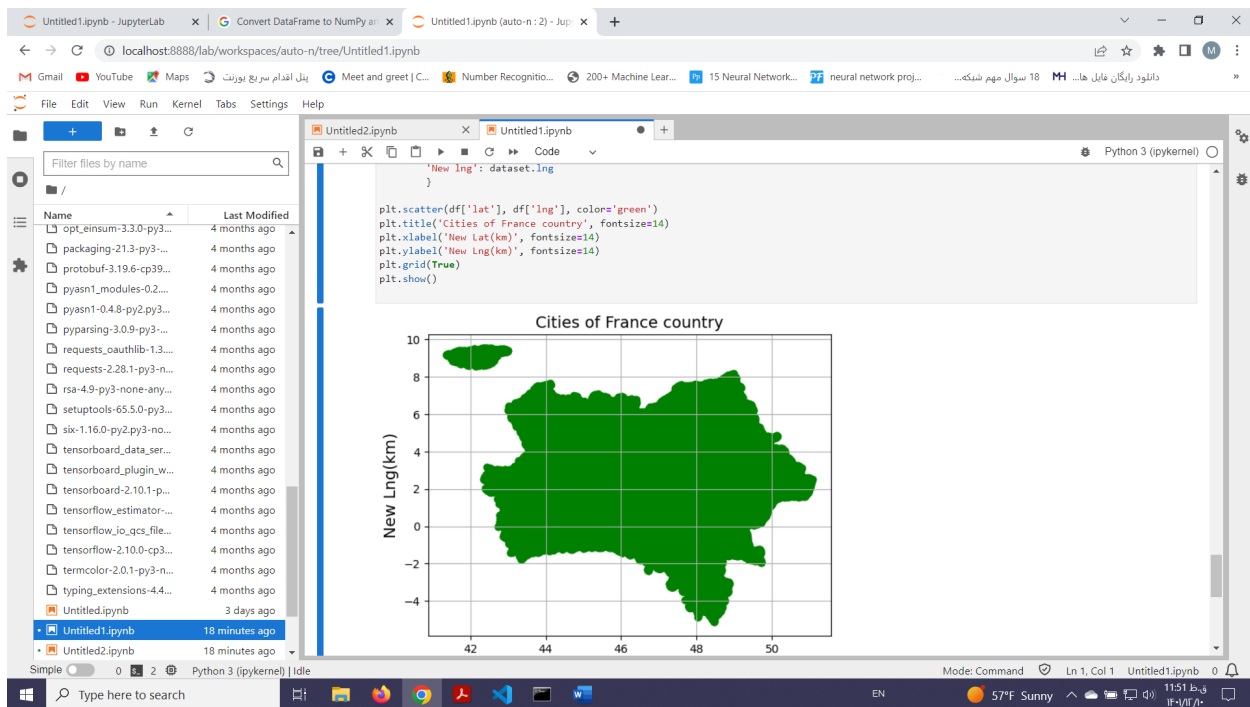
[21]: import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df = pd.DataFrame(dataset)
df['lat'] = df.lat,
df['lng'] = df.lng
```

ترسیم دوباره نمودار بر اساس ستونهای جدید:

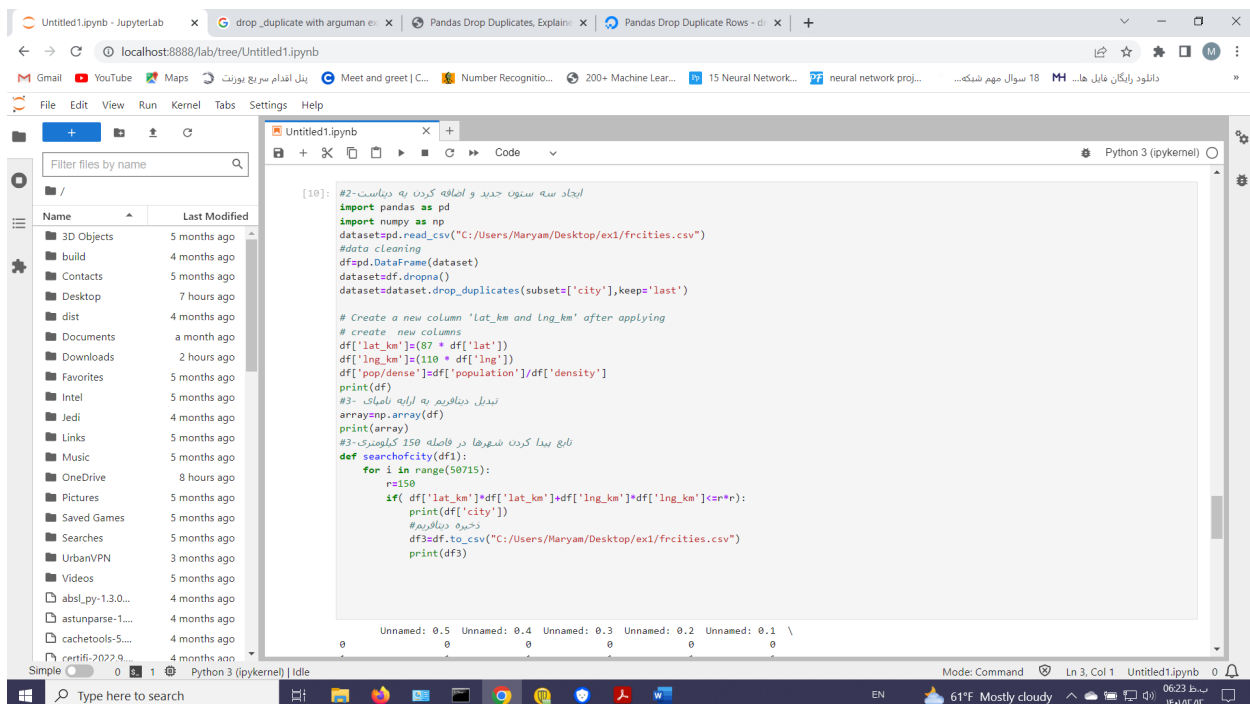
The screenshot shows the JupyterLab interface with a file explorer on the left and a code editor on the right. The code editor displays a scatter plot of cities in France, with the x-axis labeled 'Lat' and the y-axis labeled 'lng(km)'. The plot shows a green area representing the map of France, with a grid overlay. The code in the editor is as follows:

```
[24]: import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df = pd.DataFrame(dataset)
df['lat'] = df.lat,
df['lng'] = df.lng

plt.scatter(df['lat'], df['lng'], color='green')
plt.title('Cities of France country')
plt.xlabel('New Lat(km)', fontsize=14)
plt.ylabel('New lng(km)', fontsize=14)
plt.grid(True)
plt.show()
```



بارگذاری دیتاست و DataCleaning ایجاد سه ستون جدید در دیتاست تبدیل دیتافریم به آرایه نامپای و محاسبه تراکم جمعیت و تابع پیدا کردن شهرها بر اساس فرمول دایره و فاصله نقاط از مبدا مختصات و ذخیره دیتافریم:



خروجی:

```

50712  50712  50712  50712  50712  50712
50713  50713  50713  50713  50713  50713
50714  50714  50714  50714  50714  50714
50715  50715  50715  50715  50715  50715

   Unnamed: 0  city  lat  lng iso2  density \
0  0  Saint-Obias  45.5674  5.0447  FR  129.2
1  1  Louresse  47.2394 -0.3136  FR  NaN
2  4  Gottenhouse  48.7208  7.3611  FR  305.6
3  5  Bergues  50.9686  2.4342  FR  2755.3
4  7  Saint-Martin-d'Arrossa  43.2381 -1.3133  FR  29.2
...  ...  ...  ...  ...  ...
50711  59055  Neuvecelle  46.3950  6.6125  FR  769.0
50712  59056  Cognocoli-Monticchi  41.8283  8.9058  FR  4.8
50713  59060  La Rochette  45.3056  3.4747  FR  14.4
50714  59062  Saint-Eutrope  44.4535  0.5204  FR  34.2
50715  59063  Taillis  48.1889 -1.2389  FR  81.2

   population  ranking  lat_km  lng_km  pop/dense
0  NaN  4  3964.3638  554.917  NaN
1  872.0  3  4109.8278 -34.496  NaN
2  382.0  3  4238.7096  809.721  1.250000
3  3637.0  2  4434.2682  267.762  1.320001
4  538.0  3  3761.7147 -144.463  18.424658
...  ...  ...  ...  ...
50711  3076.0  3  4036.3650  727.375  4.000000
50712  171.0  3  3639.0621  979.638  35.625000
50713  NaN  4  3941.5872  382.217  NaN
50714  NaN  4  3867.4545  57.244  NaN
50715  996.0  3  4192.4343 -136.279  12.266010

[50716 rows x 16 columns]
[[0 0 0 ... 3964.3638 554.9169999999999 nan]
 [1 1 ... 4109.8278 -34.496 nan]
 [2 2 ... 4238.7096 809.721 1.25]]

```

محاسبه ضریب همبستگی بین ستونهای مشخص شده دیتافریم:

```

[0]: import pandas as pd
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['population'].corr(df['density'])

[0]: 0.09561879557310039

[1]: import pandas as pd
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['pop/dense']=df[['population']]/df['density']
df['lat'].corr(df['pop/dense'])

[1]: nan

[ ]:

[ ]:

```


مقدار ضریب همبستگی بین دو ستون با استفاده از متدهای `corr` برای `Pearsonr` و متدهای `Spearman` و `Kendall` محاسبه شده است:

```

[0]: #pearson coefficient correlation
import pandas as pd
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['population'].corr(df['density'])

[0]: 0.09561879557310039

[1]: import pandas as pd
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['pop/dense']=df['population']/df['density']
df['lat'].corr(df['pop/dense'])

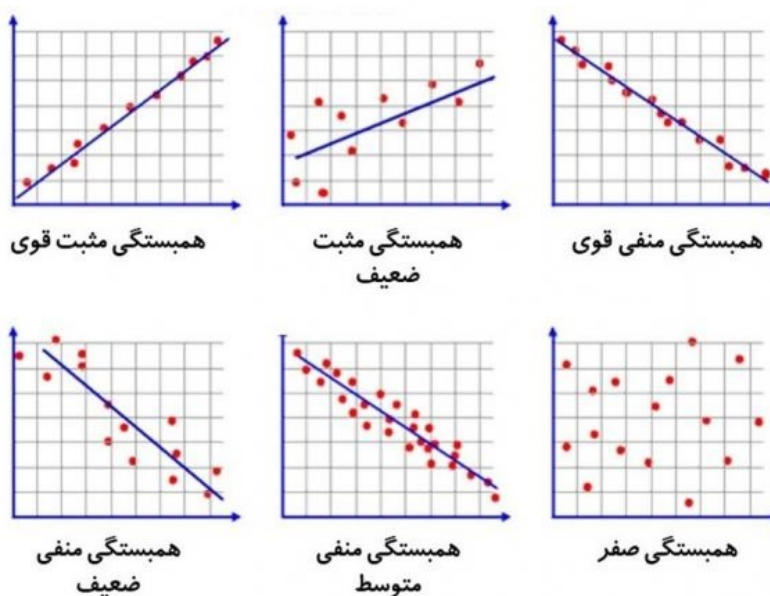
[1]: nan

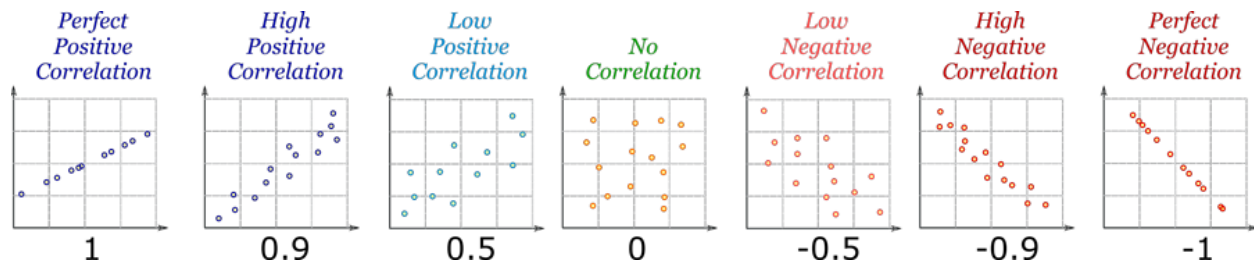
[0]: #s.corr(y) calculates Pearson coefficient correlation for x in relation y, we can call .corr() function as follows:spearman method
import pandas as pd
import scipy as sc
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['population'].corr(df['density'],method='spearman')

[0]: 0.7943011348804546

[11]: #kendall method
import pandas as pd
import scipy as sc
dataset=pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df=pd.DataFrame(dataset)
df['pop/dense']=df['population']/df['density']
df['pop/dense'].corr(df['lat'],method='kendall')
  
```

همبستگی و نمودارهای پراکندگی





با توجه به مقادیر محاسبه شده density و population با متد pearson همبستگی مثبت خطی دارند. میزان همبستگی در شکل بالا کاملاً مشخص است.

The screenshot shows a JupyterLab environment with a file explorer on the left and a code editor on the right. The code in the editor is as follows:

```
[0]: #x.corr(y) calculates Pearson coefficient correlation for x in relation y, we can call .corr() function as follows: spearman method
import pandas as pd
import scipy as sc
dataset = pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df = pd.DataFrame(dataset)
df[["population"], df[["density"], method="spearman")

[8]: 0.7943011348804546

[11]: #kendall method
import pandas as pd
import scipy as sc
dataset = pd.read_csv("C:/Users/Maryam/Desktop/ex1/frcities.csv")
df = pd.DataFrame(dataset)
df[["pop/dense"]] = df[["population"]]/df[["density"]]
df[["pop/dense"]].corr(df[["lat"]], method="kendall")

[11]: -0.19129583307151166

[21]: #pearson method
import pandas as pd
import scipy as sc
df = pd.DataFrame(dataset)
df[["population"]].corr(df[["density"], method="pearson")

[21]: 0.09561879557310039

[ ]:
```