گزارش تمرین عملی یاسمن گودرزی 9931100

در قسمت اول با استفاده از کتابخانه pandas دیتا ست موجود در اکسل را میخوانیم و در متغیر data_set ذخیره میکنیم

```
# TO-DO
import pandas as pd
import math
import numpy as np
```
0.0s

```
# load data

data_set = pd.read_excel(r"./worldcities.xlsx")
```
5.7s

سپس در مرحله بعد از ماخواسته شده که ۴ تا ستون را از دیتامون پاک کنیم. این کار را با متد drop موجود در کتابخانه pandas انجام میدهیم. سپس از ما خواسته شده است تا نام دوتا از ستون ها را عوض کنیم این کار را با استفاده از متد rename انجام میدهیم.

```
# drop and rename
data_set.drop(columns=["ville_ascii","capital","id","admin_nom"], inplace=True)
data_set.rename(columns={"ville": "city","pays":"country"}, inplace=True)
```
0.0s

در مرحله بعد از ما خواسته شده تا دو شرط را به دیتا ستمون اضافه کنیم. شرط اول حذف شهر های است که جمعیت آنها کمتر از ۱ میلیون نفر است این کار را با استفاده از کد زیر انجام میدهیم که در آن گفتیم ستون های از دیتامون که مقدار جمعیت بیشتر از ۱۰۰۰۰۰ دارند رو نگه دار.

در قسمت بعد از ماخواسته شده تا تایپ جمعیت را از float به int تبدیل کنیم که با قطعه کد زیر انجام می دهیم.

هدف این قسمت کاهش دیتا ستمون است.

```
# filter data
data_set = data_set[data_set['population'] >= 1000000]
```
0.0s

```
# change type population
data_set['population'] = data_set['population'].astype(int)
```

هدف این قسمت و قسمت بعدی در واقع داده های گمشده است . ما در دیتا ستمون مقدار برخی از ویژگی ها را برای داده ها نداریم که به آنها داده های گمشده می گویند. پس از ما خواسته شده تا ابتدا ردیف های دیتامون که تکراری هستن را حذف کنیم تا در مدلمون دیتا تکراری وجود نداشته باشد. برای این کار از متد duplicates موجود در کتابخانه panda استفاده میکنیم. سپس ردیف های که بیش تر از دو مورد از ویژگی های آنها را از دست داده ایم را حذف میکنیم زیرا این ردیف ها اطلاعات ناقص زیادی دارند و به آموزش مدل ما کمکی نمیکنند.

```
# remove duplicated and missed values
data_set=data_set.drop_duplicates()
data_set = data_set.dropna(thresh=2)
```
0.0s

حالا بعضی از ردیف های ما هستند که مقدار دو فیلد lat و lng برای آنها خالی است امامقدار های این دو فیلد برای ما مهم هستند به علت محاسباتی که در مرحله بعدی انجام می دهیم. پس سعی میکنیم تا دیتا های خالی را با روش مناسب جایگزین کنیم. در اینجا رویکرد میانگین گیری است. در واقع ابتدا میانگین کل مقادیر دو تا فیلد lat و lng را میگیریم و سپس با استفاده از متد fillna موجود در pandas مقادیر خالی یا همان NAN را با میانگین پر میکنیم.

```
# fill the missing values by their country
average_lat = data_set['lat'].mean()
data_set['lat'].fillna(average_lat, inplace=True)
average_lng = data_set['lng'].mean()
data_set['lng'].fillna(average_lng, inplace=True)

✓ 0.0s
```

در قسمت بعدی به محاسبه فاصله بقیه شهرها تا شهر تهران می پردازیم. برای این کار ابتدا از دیتا ستمون مقدار طول و عرض جغرافیایی شهر تهران را به دست می آوریم(چون به درجه است با استفاده از کتابخانه math.radians آن را به رادیان تبدیل میکنیم.). در دستور کار از ما خواسته شده است تا مقدار فاصله را با استفاده از فرمول هاورین محاسبه کنیم. برای این کار در یک حلقه روی دیتامون پیمایش انجام میدهیم و هر بار طول و عرض جغرافیایی شهر مورد نظر را به رادیان تبدیل میکنیم و سپس طبق فرمول موجود در دستور کار کد پایتون آن را می نویسیم. در هر پیمایش اسم شهر به همراه فاصله اش تا تهران را دی یک دیکشنری ذخیره میکنیم.

```
# distance function

tehran_data = data_set[data_set['city'] == 'Tehran']
tehran_lat = math.radians(tehran_data['lat'].values[0])
tehran_lng = math.radians(tehran_data['lng'].values[0])
dist_from_tehran = {}
for item in data_set.index:
    lat = math.radians(data_set['lat'][item])
    lng = math.radians(data_set['lng'][item])
    a = np.power(np.sin((lat-tehran_lat)/2),2) + np.cos(tehran_lat) * np.cos(lat) * np.power(np.sin((lng-tehran_lng)/2),2)
    d= 2 * 6371 * np.arctan2(np.sqrt(a),np.sqrt(1-a))
    dist_from_tehran[item]= d
```

فرمول هاورسین:

$$r = 6371$$

$$a = \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)$$

$$d = 2r\,\mathrm{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

در مرحله بعدی از ما خواسته شده تا به دیتامون یک ستون اضافه کنیم که اطلاعاتش حاوی فاصله هر شهر تا تهران باشد. ابتدا یک ستون با مقدار بای دیفالت صفر به نام dist_from_tehran اضافه میکنیم و بعد در یک حلقه به ترتیب شهر ها فاصله را از دیکشنر یکه در مرحله قبل پر کردیم میخوانیم.

```python
# add a new column in DataFrame
data_set['dist_from_tehran'] = 0
for item in data_set.index:
    data_set['dist_from_tehran'][item] = dist_from_tehran[item]
```

0.0s

در دو مرحله بعدی از ما خواسته شده تا دیتامون رو بر حسب اسم شهر و مقدار lat مرتب کنیم و در نهایت فایل اکسل را ذخیره کنیم که مراحل انجام شده و فایل پیوست شد.

```python
# Sorting
data_set=data_set.sort_values(by=['city', 'lat'],ascending=[True, False])
```

[23] 0.0s

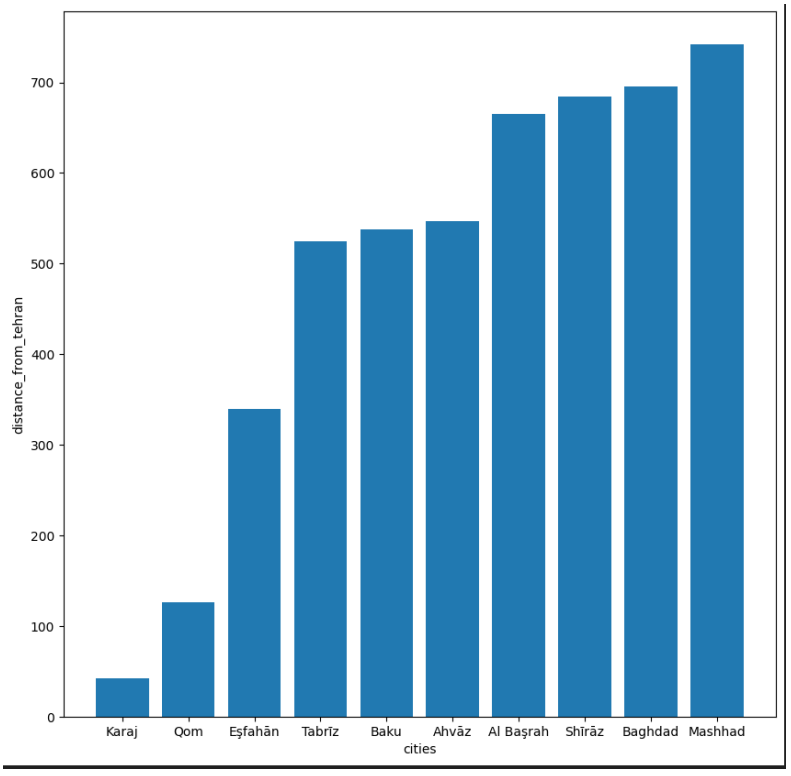
```python
# Save CSV file
data_set.to_csv("9931100.csv" ,index=False )
print(data_set)
```

[24] 0.0s

```
...              city      lat       lng              country iso2 iso3  \
19            Aba   5.1167    7.3667              Nigeria   NG  NGA
83         Abidjan   5.3167   -4.0333        Côte d'Ivoire   CI  CIV
121     Abu Dhabi  24.4667   54.3667  United Arab Emirates   AE  ARE
131         Abuja   9.0667    7.4833              Nigeria   NG  NGA
169         Accra   5.5500   -0.2000                Ghana   GH  GHA
...           ...      ...       ...                  ...  ...  ...
41227     Ürümqi  43.8225   87.6125                China   CN  CHN
385         Āgra  27.1800   78.0200                India   IN  IND
17397      İzmir  38.4200   27.1400               Turkey   TR  TUR
28638      Ōsaka  34.6939  135.5022                Japan   JP  JPN
34814  Şanlıurfa  37.1583   38.7917               Turkey   TR  TUR

       population  dist_from_tehran
19      1530000.0       5634.737243
83      4980000.0       6575.533770
121     1483000.0       1280.150009
131     3770000.0       5331.673578
169     2388000.0       6230.918394
...           ...               ...
41227   4335017.0       3197.472953
385     1585704.0       2687.466060
17397   4320519.0       2166.791444
28638  15126000.0       7381.949949
34814   1985753.0       1138.007632

[769 rows x 8 columns]
```

در مرحله مصور سازی از ما خواسته شده است تا ابتدا ۱۰ شهر که کمترین فاصله با تهران دارند را به دست بیاوریم.
برای این کار ابتدا دیتامون رو بر حسب فیلد dist from tehran نزولی مرتب میکنیم. و سپس ۱۱ ردیف اول را بر میداریم (خود تهران نیز در لیست با مقدار۰ موجود است برای همین ۱۱ عنصر را میگیریم و سپس تهران را حذف میکنیم از آن)و در نهایت نمودار آن را رسم میکنیم

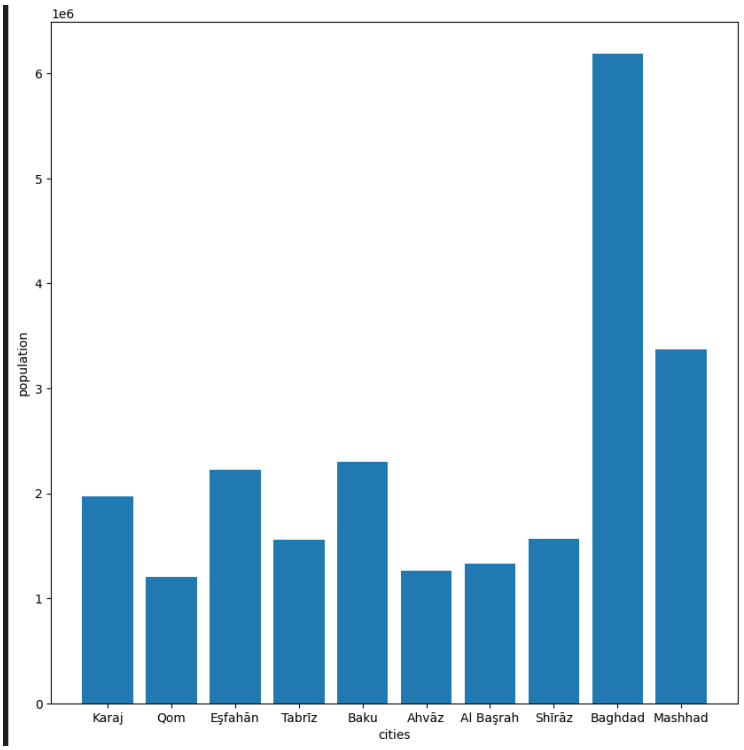
```python
# 10 Nearest Cities to Tehran
import matplotlib.pyplot as plt

data_sorted = data_set.sort_values(by='dist_from_tehran',ascending= True).head(11)
data_10_city = data_sorted[data_sorted['city'] != 'Tehran']
plt.figure(figsize=(10, 10))
plt.bar(data_10_city['city'], data_10_city['dist_from_tehran'])
plt.xlabel('cities')
plt.ylabel('distance_from_tehran')
plt.savefig('plot_1.png' )
plt.show()
```



در قسمت بعدی می خواهیم همین ده شهر را بر حسب جمعیت نمودار رسم کنیم که داریم:

```python
# Population of the 10 Nearest Cities to Tehran
plt.figure(figsize=(10, 10))
plt.bar(data_10_city['city'], data_10_city['population'])
plt.xlabel('cities')
plt.ylabel('population')
plt.savefig('plot_2.png')
plt.show()
```

✓ 0.2s



در نهایت هم نمودار خواسته شده را رسم کردیم . از آنجا که دیتای ما مختصات طول و عرض جغرافیایی شهرها است خروجی نمودار ما شبیه به نقشه واقعی شهر ها میشود.

```
# City Latitudes and Longitudes
plt.figure(figsize=(10, 10))
plt.scatter(data_set['lng'], data_set['lat'], )
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.savefig('plot_3.png')
plt.show()
```

[33] ✓ 0.2s