حوری دهش 9821413

1.1. داده ها: 20,18,27,24,22,19,16,17,20,23,18,15,21

(a)

میانگین:

$$\mu = \frac{\sum x}{N} = \frac{20+18+27+24+22+19+16+17+20+23+18+15+21}{13} = \frac{260}{13} = 20$$

میانه:

ترتیب داده ها به صورت صعودی:

15,16,17,18,18,19,20,20,21,22,23,24,27

عضو وسط: 20

مد:

عضوی که بیشترین تکرار را داد: 20 , 18

چارک:

چارک اول: $\frac{17+18}{2} = 17.5$

چارک دوم یا میانه: 20

چارک سوم: $\frac{22+23}{2} = 22.5$

واریانس:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$= \frac{1}{13}((15-20)^2 + (16-20)^2 + (17-20)^2 + (18-20)^2 + (18-20)^2$$
$$+ (19-20)^2 + (20-20)^2 + (20-20)^2 + (21-20)^2 + (22-20)^2$$
$$+ (23-20)^2 + (24-20)^2 + (27-20)^2) = \frac{138}{13} \approx 10.62$$

(b)

نوت بوک در کنار این فایل ضمیمه شده است.

```
import numpy as np

data = np.array([20,18,27,24,22,19,16,17,20,23,18,15,21])

mean = np.mean(data)

median = np.median(data)

unique_values, counts = np.unique(data, return_counts=True)
modes = unique_values[counts == counts.max()]
result = np.where(np.isin(data, modes))
tmp = set(data[result])

quartiles = np.percentile(data, [25, 50, 75])

variance = np.var(data)

print("Average:", mean)
print("Middle:", median)
print("Mode:", tmp)
print("Quarters:", quartiles)
print("Variance:", variance)
```

```
Average: 20.0
Middle: 20.0
Mode: {18, 20}
Quarters: [18. 20. 22.]
Variance: 10.615384615384615
```

توضیح کد:

1. ابتدا کتابخونه numpy را فراخوانی میکنیم

2. یک آرایه numpy به نام data ایجاد میکنیم که در گام های پایین تر از آن استفاده کنیم

3. میانگین داده ها را حساب میکنیم با استفاده از تابع موجود آن در کتابخونه numpy

4. میانه داده ها رو حساب میکنیم با استفاده از تابع موجود آن در کتابخونه numpy

5. برای به دست آوردن مد، ابتدا مقادیر یکتا در داده ها و تعداد تکرار آنها را محاسبه می کنیم و در دو متغیر مجزا ذخیره می کنیم سپس حالت هایی که بیشترین تکرار را دارند مشخص می کنیم و در متغییر modes قرار می دهیم در نهایت ایندکس هایی که مقادیر آنها در لیست modes وجود دارد را به دست می آوریم و در متغیر result ذخیره می کنیم. در آخر برای اطمینان از حذف تکرارها، مقادیری که در متغیر result وجود دارد را به مجموعه تبدیل می کنیم و درون tmp قرار می دهیم.

6. چارک ها را حساب میکنیم با استفاده از تابع موجود آن در کتابخونه numpy

7. واریانس را حساب می کنیم با استفاده از تابع موجود آن در کتابخونه numpy

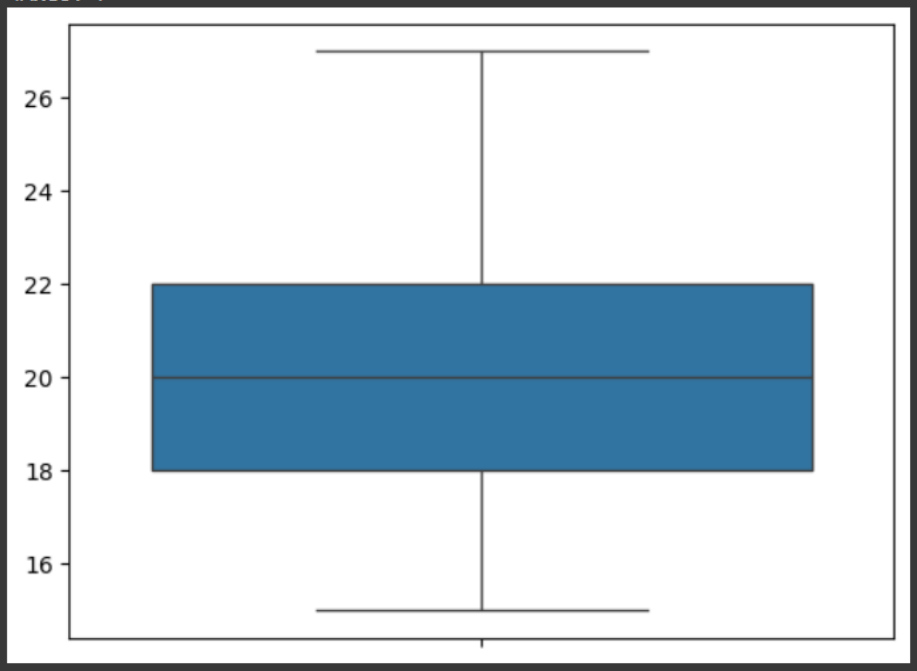
8. خروجی ها را چاپ می کنیم

(c)

نوت بوک در کنار این فایل ضمیمه شده است.

```
import seaborn as sns

data = [20,18,27,24,22,19,16,17,20,23,18,15,21]
sns.boxplot(data)
```

```
<Axes: >
```



توضیح کد:

1. استفاده از کتابخانه Seaborn برای رسم نمودار Boxplot

2. داده هایی که قرار است روی نمودار Boxplot نمایش داده شوند، به صورت یک لیست در متغیر data ذخیره می شوند

3. نمودار Boxplot برای data رسم می شود

(d)

$$Z-score: z = \frac{x-\mu}{\sigma}$$

$$z_{15} = \frac{15-20}{3.26} \approx -1.53$$

$$z_{16} = \frac{16-20}{3.26} \approx -1.23$$

$$z_{17} = \frac{17-20}{3.26} \approx -0.92$$

$$z_{18} = \frac{18 - 20}{3.26} \approx -0.61$$

$$z_{19} = \frac{19 - 20}{3.26} \approx -0.31$$

$$z_{20} = \frac{20 - 20}{3.26} = 0$$

$$z_{21} = \frac{21 - 20}{3.26} \approx 0.31$$

$$z_{22} = \frac{22 - 20}{3.26} \approx 0.61$$

$$z_{23} = \frac{23 - 20}{3.26} \approx 0.92$$

$$z_{24} = \frac{24 - 20}{3.26} \approx 1.23$$

$$z_{27} = \frac{27 - 20}{3.26} \approx 2.15$$

(e)

نوت بوک در کنار این فایل ضمیمه شده است.

1-e

```
import numpy as np

data = np.array([20, 18, 27, 24, 22, 19, 16, 17, 20, 23, 18, 15, 21])
mean = np.mean(data)
std_dev = np.std(data)


for num, z_score in zip(data, (data - mean) / std_dev):
    print(f"Z-score for {num}: {z_score}")
```

```
Z-score for 20: 0.0
Z-score for 18: -0.6138498140448514
Z-score for 27: 2.1484743491569795
Z-score for 24: 1.2276996280897028
Z-score for 22: 0.6138498140448514
Z-score for 19: -0.3069249070224257
Z-score for 16: -1.2276996280897028
Z-score for 17: -0.920774721067277
Z-score for 20: 0.0
Z-score for 23: 0.920774721067277
Z-score for 18: -0.6138498140448514
Z-score for 15: -1.5346245351121284
Z-score for 21: 0.3069249070224257
```

توضیح کد:

1. ابتدا کتابخونه numpy را فراخوانی میکنیم

2. یک آرایه numpy به نام data ایجاد میکنیم که در گام های پایین تر از آن استفاده کنیم

3. میانگین داده ها را حساب میکنیم با استفاده از تابع موجود آن در کتابخونه numpy

4. انحراف معیار داده ها رو حساب میکنیم با استفاده از تابع موجود آن در کتابخونه numpy

5. یک حلقه for ایجاد می کنیم که برای هر عدد در مجموعه داده و متناظر با Z-Score آن عدد اجرا می شود در نهایت Z-Score برای هر عدد محاسبه شده و چاپ می شود و Z-Score برای هر عدد برابر با فاصله آن عدد از میانگین مجموعه داده بر انحراف معیار است

2.1. نوع داده؟ (جدولی، گراف، Ordered)

(آ) داده های سری زمانی قیمت سهام برای یک شرکت معین در طول یک ما

نوع داده ها اینجا به صورت Ordered است چون داده ها به ترتیب زمانی ثبت شده اند.

(ب) داده های موجودی در یک سیستم مدیریت انبار

این داده ها به صورت جدولی هستند چون شامل مواردی همچون نام محصول، تعداد موجودی و.. هستند.

(ج) آمار ورزشی برای ورزشکاران

به صورت جدولی است زیرا شامل مواردی همچون نام ورزشکار، تعداد پاس گل و.. است.

(د) شبکه های حمل و نقل که مسیرها و اتصالات بین فرودگاه ها را نشان می دهند

نوع داده اینجا به صورت گراف است زیرا ارتباطات و اتصالات بین نقاط مختلف را نشان می دهد.

(ه) داده های بیان ژن در مراحل مختلف رشد در یک ارگانیسم

نوع داده ها اینجا به صورت Ordered است چون توی ژن ترتیب داریم.

(و) شبکه‌های جاده‌ای که خیابان ها و تقاطع ها را در نقشه شهر نشان می دهند

نوع داده اینجا به صورت گراف است زیرا ارتباطات و اتصالات بین نقاط مختلف (تقاطع ها و خیابان ها) را نشان می دهد.

(ز) توپولوژی اینترنت که اتصالات بین روترها و شبکه ها را نشان می دهد

این نوع داده نیز به صورت گراف است زیرا اتصالات بین اجزای مختلف را نشان می دهد.

(ح) جدول زمانی تکامل گونه ها بر اساس سوابق فسیلی

نوع داده ها اینجا به صورت Ordered است

(ط) اطلاعات مشتری در یک سیستم CRM

نوع داده اینجا به صورت جدولی است زیرا شامل اطلاعات مشتریان مختلف می باشد.

(ی) داده های گزارش متوالی که تعاملات کاربر را در یک وب سایت ثبت می کند

این نوع داده به صورت جدولی است زیرا شامل سوابق زمانی تعاملات کاربر است.

(ک) شبکه ای از تعاملات بین پروتئین ها در یک سیستم بیولوژیکی

این نوع داده نیز به صورت گراف می باشد زیرا شامل اتصالات و ارتباطات بین اجزای مختلف است.

(ل) نمرات دانش آموز در یک سیستم مدرسه

این نوع داده به صورت جدولی می باشد زیرا شامل اطلاعات مرتبط با دانش‌آموزان است.

(م) روابط هم نویسندگی بین محققان در یک شبکه انتشارات علمی

این نوع داده به صورت گراف است زیرا روابط و ارتباطات بین محققان را نشان می دهد.

(ن) روابط وابستگی بین ماژول های نرم افزار در یک پایگاه کد

این نوع داده به صورت گراف است زیرا ارتباطات و وابستگی‌های بین اجزای مختلف نرم‌افزار را نشان می دهد.

(س) خواندن علائم حیاتی بیمار در فواصل منظم در بیمارستان

این نوع داده به صورت Ordered است زیرا شامل مقادیری از علائم حیاتی (مانند ضربان قلب، فشار خون و..) است که در طول زمان ثبت شده اند.

3.1. فاصله Mahalanobis ؟

```
values:
X=5 , Y=8
Mean vector (μ) = (4.7)
Covariance matrix (Σ):
110 41
14 51
```

$$Mahalanobis(x,\mu) = ((x-\mu)^T \sum^{-1} (x-\mu))^{0.5}$$

$$x-\mu = \begin{bmatrix}5\\8\end{bmatrix} - \begin{bmatrix}4.7\\4.7\end{bmatrix} = \begin{bmatrix}0.3\\3.3\end{bmatrix}$$

$$(x-\mu)^T = [0.3 \quad 3.3]$$

$$a = 110\,, b = 41\,, c = 14\,, d = 51$$

$$\sum^{-1} = \frac{1}{ad-bc}\begin{bmatrix}d & -b\\-c & a\end{bmatrix} = \frac{1}{(110*51)-(41*14)}\begin{bmatrix}51 & -41\\-14 & 110\end{bmatrix} = \frac{1}{5036}\begin{bmatrix}51 & -41\\-14 & 110\end{bmatrix}$$

$$(x-\mu)^T \sum^{-1} = [0.3 \;\; 3.3] \times \begin{bmatrix} \frac{51}{5036} & \frac{-41}{5036} \\ \frac{-14}{5036} & \frac{110}{5036} \end{bmatrix} = [\frac{-30.9}{5036} \quad \frac{350.7}{5036}]$$

$$(x-\mu)^T \sum^{-1} (x-\mu) = [\frac{-30.9}{5036} \quad \frac{350.7}{5036}] \times \begin{bmatrix} 0.3 \\ 3.3 \end{bmatrix} = \frac{-9.27}{5036} + \frac{1157.31}{5036} = \frac{1148.04}{5036}$$

$$Mahalanobis(x,\mu) = ((x-\mu)^T \sum^{-1} (x-\mu))^{0.5} = (\frac{1148.04}{5036})^{0.5} \approx 0.477$$

4.1. مقدار Jaccard و SMC ؟

```
A = {apple, banana, cherry, date}
B = {banana, cherry, date, elderberry}
```

ابتدا مجموعه کلی را به صورت زیر قرار میدهیم:

{apple, banana, cherry, date, elderberry}

حالا طبق مجموعه کلی بالا، آن هایی که توی مجموعه A , B قرار دارند را برابر با یک می گذاریم و آن هایی که وجود ندارند را برابر با صفر:

A={1,1,1,1,0}

B={0,1,1,1,1}

$$f_{01} = 1 \quad , \quad f_{10} = 1 \quad , \quad f_{00} = 0 \quad , \quad f_{11} = 3$$

$$SMC = \frac{f_{11} + f_{00}}{f_{01} + f_{10} + f_{00} + f_{11}} = \frac{3+0}{3+1+1+0} = \frac{3}{5}$$

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} = \frac{3}{3+1+1} = \frac{3}{5}$$

5.1. آنتروپی ؟

```
P(A) = 0.4
P(B) = 0.3
P(C) = 0.3
```

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$H(x) = -(p_A \log_2 p_A + p_B \log_2 p_B + p_C \log_2 p_C)$$
$$= -(0.4 \log_2 0.4 + 0.3 \log_2 0.3 + 0.3 \log_2 0.3)$$

$$\log_2 0.4 \cong -1.322$$

$$\log_2 0.3 \cong -1.737$$

$$H(x) = -((0.4 * -1.322) + ((0.3 * -1.737) * 2)) \approx 1.571$$

.6.1

(a)

نوت بوک در کنار این فایل ضمیمه شده است.

طبق عکس های زیر:

بالاترین میانگین متعلق به فروشگاه N17 6QA هست و کمترین میانگین متعلق به فروشگاه W4 3PH است.

```
[3] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[4] import pandas as pd

    df=pd.read_csv('/content/drive/MyDrive/LaptopSalesJanuary2008.csv')
    df.head()
```

	Date	Configuration	Customer Postcode	Store Postcode	Retail Price	Screen Size (Inches)	Battery Life (Hours)	RAM (GB)	Processor Speeds (GHz)	Integrated Wireless?	HD Size (GB)	Bundled Applications?	OS X Customer	OS Y Customer	OS X Store	OS Y Store	CustomerStoreDistance
0	1/1/2008 0:01	163	EC4V 5BH	SE1 2BN	455	15	5	1	2.0	Yes	80	Yes	532041	180995	534057.0	179682.0	2405.873022
1	1/1/2008 0:02	320	SW4 0JL	SW12 9HD	545	15	6	1	2.0	No	300	No	529240	175537	528739.0	173080.0	2507.558574
2	1/1/2008 0:04	23	EC3V 1LR	E2 0RY	515	15	4	1	2.0	Yes	300	Yes	533095	181047	535652.0	182961.0	3194.001409
3	1/1/2008 0:04	169	SW1P 3AU	SE1 2BN	395	15	5	1	2.0	No	40	Yes	529902	179641	534057.0	179682.0	4155.202281
4	1/1/2008 0:06	365	EC4V 4EG	SW1V 4QQ	585	15	6	2	2.0	No	120	Yes	531684	180948	528924.0	178440.0	3729.298057

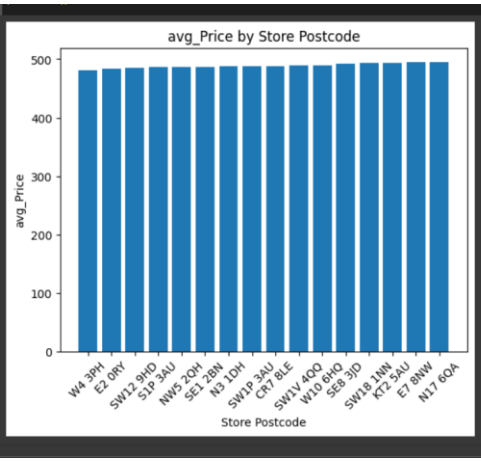
```
[6] import matplotlib.pyplot as plt

    avg_price_by_store=df.groupby('Store Postcode')['Retail Price'].mean().sort_values()

    avg_price_by_store.head()

    Store Postcode
    W4 3PH      481.006289
    E2 0RY      483.171729
    SW12 9HD    485.295699
    S1P 3AU     486.250000
    NW5 2QH     486.580460
    Name: Retail Price, dtype: float64
```

```
[7] avg_price_by_store

Store Postcode
W4 3PH      481.006289
E2 0RY      483.171729
SW12 9HD    485.295699
S1P 3AU     486.250000
NW5 2QH     486.580460
SE1 2BN     486.680195
N3 1DH      487.368421
SW1P 3AU    488.506858
CR7 8LE     488.619048
SW1V 4QQ    489.344978
W10 6HQ     489.866667
SE8 3JD     492.177778
SW18 1NN    493.038922
KT2 5AU     493.904762
E7 8NW      494.381443
N17 6QA     494.634146
Name: Retail Price, dtype: float64

avg_price_by_store.index

Index(['W4 3PH', 'E2 0RY', 'SW12 9HD', 'S1P 3AU', 'NW5 2QH', 'SE1 2BN',
       'N3 1DH', 'SW1P 3AU', 'CR7 8LE', 'SW1V 4QQ', 'W10 6HQ', 'SE8 3JD',
       'SW18 1NN', 'KT2 5AU', 'E7 8NW', 'N17 6QA'],
      dtype='object', name='Store Postcode')
```

```
[9] avg_price_by_store.values

array([481.00628931, 483.17172897, 485.29569892, 486.25      ,
       486.58045977, 486.68019481, 487.36842105, 488.50685786,
       488.61904762, 489.34497817, 489.86666667, 492.17777778,
       493.03892216, 493.9047619 , 494.3814433 , 494.63414634])

test_dic = {"Store Postcode": list(avg_price_by_store.index), "avg_Price":list(avg_price_by_store.values)}
test_dic

{'Store Postcode': ['W4 3PH',
  'E2 0RY',
  'SW12 9HD',
  'S1P 3AU',
  'NW5 2QH',
  'SE1 2BN',
  'N3 1DH',
  'SW1P 3AU',
  'CR7 8LE',
  'SW1V 4QQ',
  'W10 6HQ',
  'SE8 3JD',
  'SW18 1NN',
  'KT2 5AU',
  'E7 8NW',
  'N17 6QA'],
 'avg_Price': [481.0062893081761,
  483.17172897196264,
  485.2956989247312,
  486.25,
  486.58045977011494,
```

```
{'Store Postcode': ['W4 3PH',
  'E2 0RY',
  'SW12 9HD',
  'S1P 3AU',
  'NW5 2QH',
  'SE1 2BN',
  'N3 1DH',
  'SW1P 3AU',
  'CR7 8LE',
  'SW1V 4QQ',
  'W10 6HQ',
  'SE8 3JD',
  'SW18 1NN',
  'KT2 5AU',
  'E7 8NW',
  'N17 6QA'],
 'avg_Price': [481.0062893081761,
  483.17172897196264,
  485.2956989247312,
  486.25,
  486.58045977011494,
  486.6801948051948,
  487.36842105263156,
  488.5068578553616,
  488.6190476190476,
  489.34497816593887,
  489.8666666666667,
  492.1777777777778,
  493.0389221556886,
  493.9047619047619,
  494.3814432989691,
  494.6341463414634]}
```

```
df_mean = pd.DataFrame(test_dic)
fig , ax=plt.subplots()
ax.bar(df_mean['Store Postcode'],df_mean['avg_Price'])
ax.set_ylabel('avg_Price')
ax.set_xlabel('Store Postcode')
ax.set_title('avg_Price by Store Postcode')
plt.xticks(rotation=45)
plt.show()
```



توضیح کد:

1. ابتدا فایل را در گوگل درایو آپلود کردم و بعد از colab به گوگل درایو متصل شدم

2. کتابخونه Pandas را فراخوانی کردم

3. از گوگل درایو فایل csv مورد نظر را خواندم و به یک دیتا فریم Pandas با نام df تبدیلش کردم

4. پنج نمونه اول مجموعه داده را با دستور df.head نمایش دادم

5. کتابخونه matplotlib را فراخوانی کردم

6. از دیتا فریم df میانگین Retail Price را بر اساس Store Postcode گروه بندی کرده و سپس نتایج را بر اساس میانگین قیمت مرتب سازی کردم و داخل avg_price_by_store قرار دادم

7. پنج نمونه اول را با استفاده از avg_price_by_store.head نمایش دادم

8. در اینجا یک دیکشنری به نام test_dic ایجاد کردم که شامل کلیدهای Store Postcode و avg_Price است و مقادیر مربوط به آن ها به ترتیب از نتایج محاسبات avg_price_by_store به دست آمد

9. از دیکشنری test_dic یک دیتا فریم جدید به نام df_mean ایجاد کردم

10. براساس plt.subplots یک شکل جدید و یک محور جدید برای نمودار ایجاد شد

11. یک نمودار میله ای از avg_Price براساس Store Postcode درست کردم

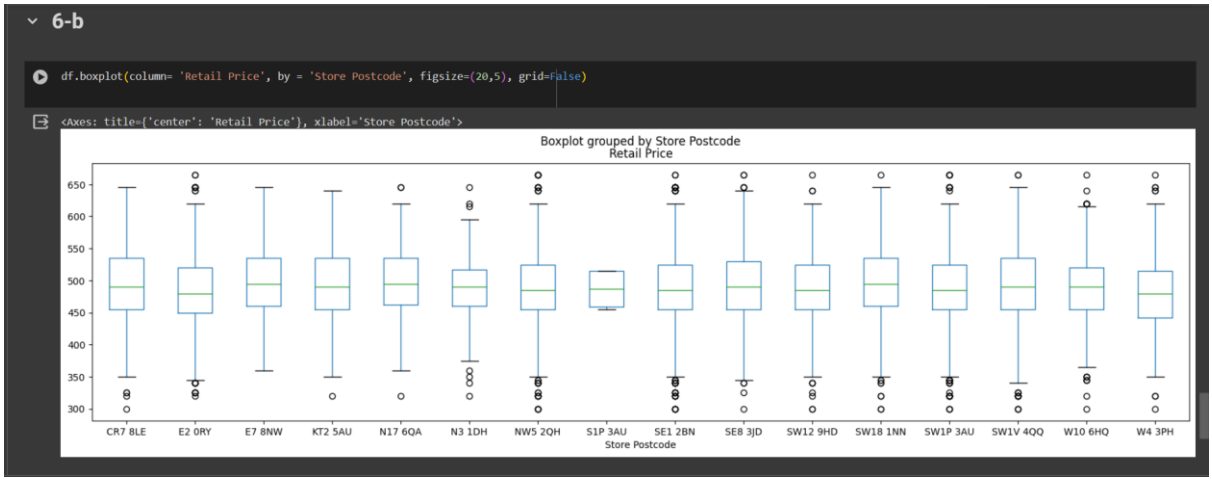
12. لیبل محور y را avg_Price و لیبل محور x را Store Postcode گذاشتم و برای نمودار یک تاتیل به اسم avg_Price by Store Postcode قرار دادم

13. برچسب های محور x با چرخش 45 درجه قرار گرفته اند

14. نمودار را رسم کردم

(b)

نوت بوک در کنار این فایل ضمیمه شده است.



توضیح کد:

1. در اینجا داده های خرده فروشی را بر اساس ستون Retail Price در دیتا فریم df در نظر گرفتم و آنها را بر اساس مقادیر در ستون Store Postcode گروه بندی کردم

2. با استفاده از آرگومان column=Retail Price مشخص می شود که کدام ستون از داده ها برای ایجاد باکس پلات استفاده شده است

3. با استفاده از آرگومان by=Store Postcode مشخص میشود که بر اساس کدام ستون، داده ها گروه بندی شده است

4. اندازه نمودار را تنظیم کردم و در آخر خطوط شبکه ای را در پس زمینه غیر فعال کردم

تفاوت:

```
[35] df_max = df[df['Store Postcode'] == 'N17 6QA']['Retail Price']
     # ---------------------------------
     data = df_max
     quartiles = np.percentile(data, [25, 50, 75])

     print("Quarters:", quartiles)

     Quarters: [462.5 495.  535. ]

     df_min = df[df['Store Postcode'] == 'W4 3PH']['Retail Price']
     # ---------------------------------
     data = df_min
     quartiles = np.percentile(data, [25, 50, 75])

     print("Quarters:", quartiles)

     Quarters: [442.5 480.  515. ]
```

برای فروشگاه N17 6QA میانه 495 است و چارک اول آن تقریبا 462.5 و چارک سوم آن 535 است. همینطور برای این فروشگاه توزیع پراکندگی بین 462.5 تا 535 قرار دارد و این فروشگاه تنها 2 داده پرت دارد. از طرفی می دانیم میانگین این فروشگاه تقریبا نزدیک 494.63 است پس فراوانی این دیتاها تقریبا به صورت negatively skewed است.

برای فروشگاه W4 3PH میانه 480 است و چارک اول آن تقریبا 442.5 و چارک سوم آن 515 است. همینطور برای این فروشگاه توزیع پراکندگی بین 442.5 تا 515 قرار دارد و این فروشگاه تنها 5 داده پرت دارد. از طرفی می دانیم میانگین این فروشگاه تقریبا 481 است پس فراوانی این دیتاها تقریبا به صورت positively skewed است.

.7.1

(a)

ساختار فایل CSV یکی از رایج ترین و پرکاربردترین فرمت های ذخیره سازی داده است. این فرمت به سادگی داده ها را در قالب متنی و جدولی ذخیره می کند و برای انتقال و به اشتراک گذاری داده ها بین برنامه ها و سامانه های مختلف استفاده می شود.

در این فرمت، داده‌ها بر اساس مقادیر جداشده با کاما و یا سایر علائم جداکننده مانند ویرگول یا تب قرار می گیرند.

ساختار فایل csv در داده کاوی به صورت های زیر می تواند باشد که روش های زیر متداول ترین روش ها برای ساخت فایل csv است:

۱. استفاده از اکسل یا شیت های گوگل:

- باز کردن نرم‌افزار اکسل یا ورود به Google Sheets
- وارد کردن داده ها به صورت جدول در ستون ها و ردیف ها
- ذخیره کردن فایل با فرمت csv:

در اکسل برای این کار باید قسمت فایل می رویم و Save As را می زنیم و فرمت csv را انتخاب میکنیم در Google Sheets می توانیم از گزینه فایل و بعدش دانلود و بعدش csv. را انتخاب میکنیم

۲. استفاده از ویرایشگر متنی:

- باز کردن یک ویرایشگر متنی مانند Notepad (در ویندوز) یا TextEdit (در مک) یا هر ویرایشگر متنی دیگری
- وارد کردن داده ها به صورت متنی با استفاده از کاما یا دیگر علائم جداکننده
- ذخیره کردن فایل با پسوند csv

۳. استفاده از زبان‌های برنامه‌نویسی:

- با استفاده از زبان های برنامه‌نویسی مانند Python، می‌توانیم داده ها را به صورت برنامه ای به فرمت csv ذخیره کنیم. به عنوان مثال در Python با استفاده از کتابخانه pandas می توانیم یک DataFrame ایجاد کنیم و با استفاده از دستور to_csv آن را به صورت فایل csv ذخیره کنیم.

مزایا فایل csv در داده کاوی:

۱. سادگی و قابلیت استفاده:

فایل های csv به صورت متنی و با استفاده از مقادیر جداشده با کاما ذخیره می شوند که باعث می شود این فایل ها بسیار ساده و قابل استفاده باشند. همچنین به راحتی می توان آنها را در بسیاری از برنامه ها و ابزارها به صورت مستقیم خواند و پردازش کرد.

۲. سازماندهی و قابلیت فهم:

فایل های csv قابلیت سازماندهی داده ها را با استفاده از ردیف ها و ستون ها دارند که این امر باعث می شود داده ها به شکلی کاملا قابل فهم و مفهومی ذخیره و انتقال پیدا کنند.

۳. قابلیت استفاده با بسیاری از ابزارها و زبان های برنامه نویسی:

فایل های csv قابلیت خواندن و پردازش در بسیاری از زبان های برنامه‌نویسی مانند Python، R، Java و... را دارند. همچنین بسیاری از ابزارهای داده‌کاوی و تحلیلی از جمله Microsoft Excel، Google Sheets و نرم‌افزارهای متن باز مانند LibreOffice Calc قادر به کار با فایل های csv هستند.

۴. حجم کمتر فایل:

فایل های csv از حجم کمتری نسبت به فرمت های ذخیره سازی، داده دیگر برخوردار هستند زیرا از فرمت متنی استفاده می کنند و دارای فشرده سازی بیشتری هستند.

۵. سازگاری با سیستم های مختلف:

فایل‌های csv برای انتقال داده بین سیستم ها با فرمت های مختلف و در پلتفرم های مختلف قابل استفاده هستند، بنابراین سازگاری بالایی دارند.

در نهایت استفاده از فایل های csv برای ذخیره و به اشتراک گذاری داده ها در داده کاوی و تحلیل داده ها بسیار مفید است زیرا سادگی، سازماندهی و قابلیت استفاده آنها را بالا می برد و همچنین باعث افزایش کارایی و انعطاف پذیری در پردازش داده می شود.

(b)

دیتاست در کنار این فایل ضمیمه شده است.

(c)

نوت بوک در کنار این فایل ضمیمه شده است.

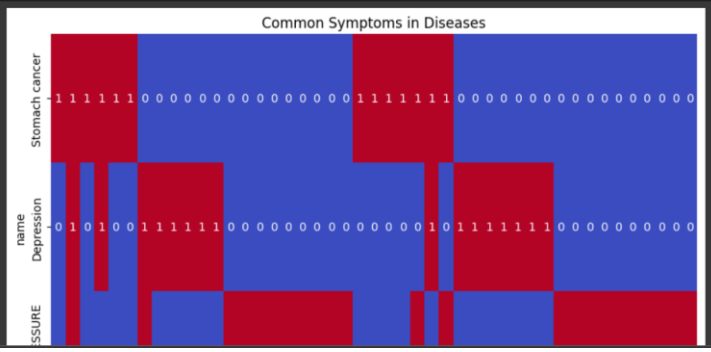
:1

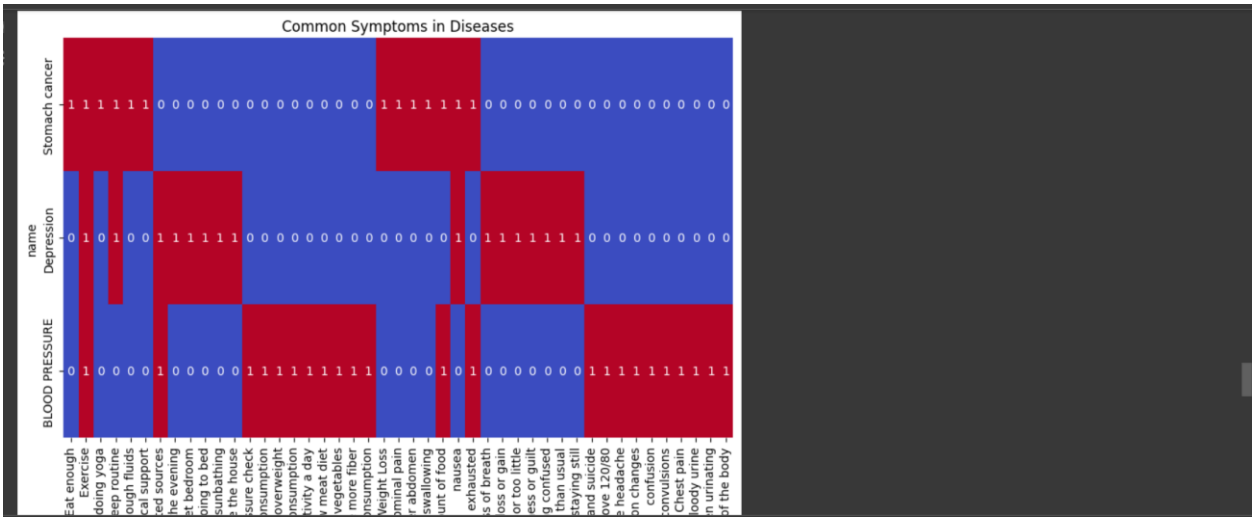
```
df=pd.read_csv('/content/drive/MyDrive/dataset.csv')
df.head()
```

	name	Eat enough	Exercise	doing yoga	Good sleep routine	Drink enough fluids	Psychological support	Eating less coffee and other caffeinated sources	Prohibition of smoking especially in the evening	Having a dark cool and quiet bedroom	...	Thinking about death and suicide	Blood pressure above 120/80	severe headache	Blurred vision or other vision changes	confusion	Fainting or convulsions	Chest pain	Brow o blood urin
0	Stomach cancer	1	1	1	1	1	1	0	0	0	...	0	0	0	0	0	0	0	
1	Depression	0	1	0	1	0	0	1	1	1	...	0	0	0	0	0	0	0	
2	BLOOD PRESSURE	0	1	0	0	0	0	1	0	0	...	1	1	1	1	1	1	1	

3 rows × 46 columns

```
df2=df.copy()
df2.set_index('name', inplace=True)
# Create a heatmap
plt.figure(figsize =(10, 6))
sns.heatmap(df2 , cmap='coolwarm', annot=True , cbar=False)
plt.title('Common Symptoms in Diseases')
plt.show()
```





توضیح کد:

1. از گوگل درایو فایل csv مورد نظر را خوانده شده و به یک دیتا فریم Pandas با نام df تبدیل میشود

2. پنج نمونه اول مجموعه داده با دستور df.head نمایش داده میشود

3. یک کپی از دیتا فریم اصلی یا df تهیه می شود و سپس با استفاده از متد set_index، ستون name به عنوان شاخص اصلی (Index) برای دیتا فریم جدید (df2) تعیین می شود.

4. با استفاده از plt.figure یک شی جدید از شکل تعریف می شود

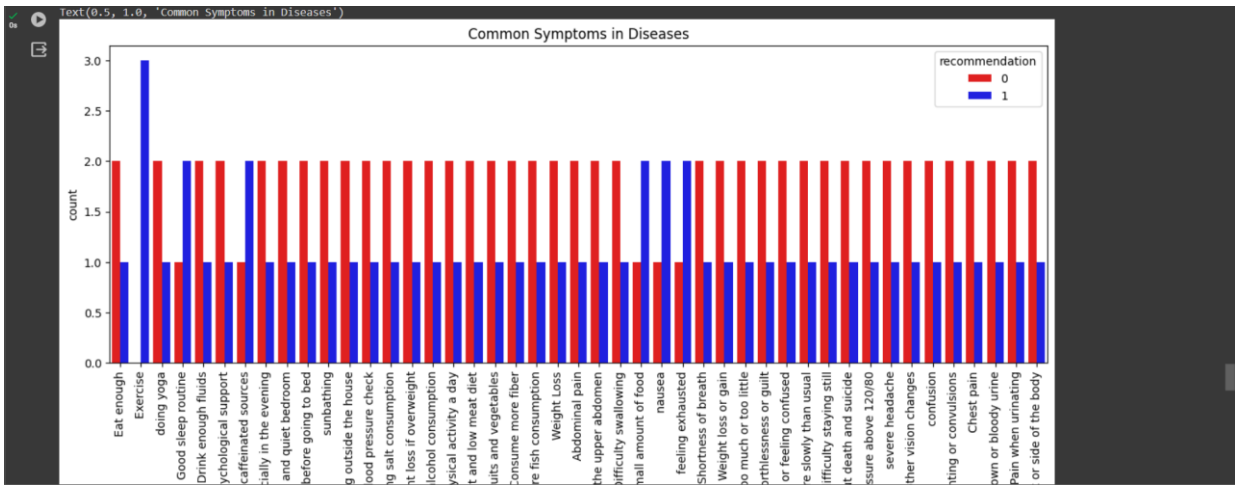
5. با استفاده از sns.heatmap یک Heatmap از داده‌های دیتا فریم ساخته می شود. در اینجا cmap مشخص می کند که از کدام نگارش رنگ برای نمایش اعداد استفاده شود و annot=True به این معنی است که مقادیر هر سلول در Heatmap نمایش داده شود و cbar=False بدین معنا است که نوار رنگ در کنار Heatmap نمایش داده نشود.

6. در نهایت برای Heatmap تایتل مشخص شده و در آخر نمایش داده میشود

:2

```
melted_df = df.melt(id_vars=['name'], var_name='condition', value_name='recommendation')

plt.figure(figsize=(50, 5))

#Create a Bar Plot
plt.subplot(1, 3, 1)
sns.countplot(data=melted_df, x='condition', hue='recommendation', palette=['red', 'blue'])

plt.xticks(rotation=90)

plt.title('Common Symptoms in Diseases')
```

```
Text(0.5, 1.0, 'Common Symptoms in Diseases')
```



توضیح کد:

در این نمودار برای هر بیماری اگر شرایط مراجعه به پزشک و توصیه های ضروری برای آن بیماری وجود داشته باشد برایش یک در نظر گرفته میشود یعنی نمودار آبی برای آن قسمت کشیده میشود و اگر صفر باشد نمودار قرمز برای آن قسمت کشیده میشود.

1. دیتا فریم با استفاده از تابع melt به یک فرمت ملت تبدیل می شود. این کار باعث می‌شود که مقادیر ستون های مختلف به ستون های یکتا تبدیل شوند. مقادیر این ستون‌ها در ستون های جدید مانند condition و recommendation ذخیره می شوند. ستون name نیز برای هر بیماری نگهداری می شود.

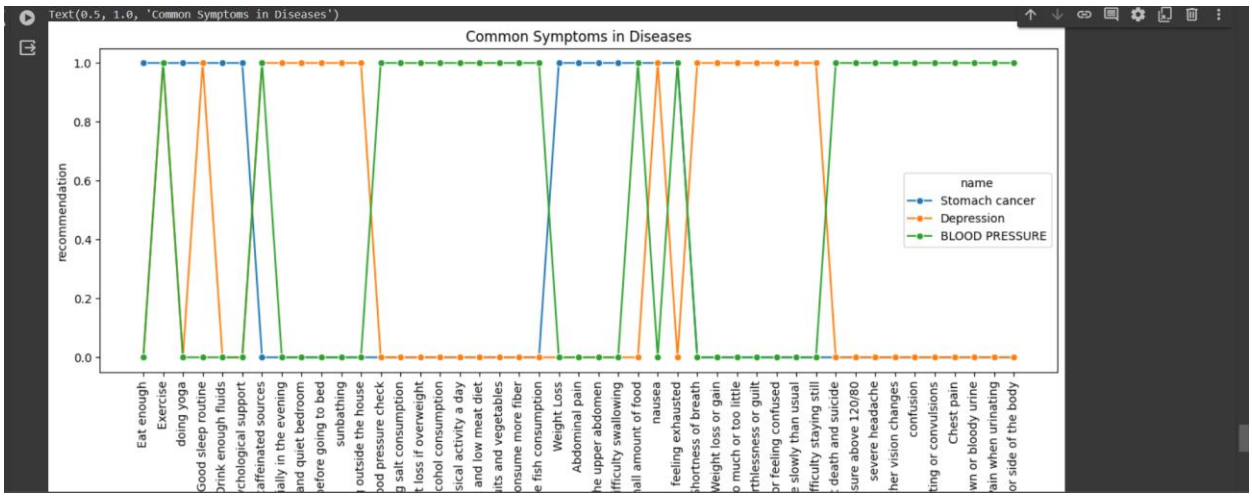
2. اندازه شکل به عرض 50 اینچ و ارتفاع 5 اینچ تنظیم شده است.

3. با استفاده از دستور plt.subplot یک شبکه از نمودارها را ایجاد میشود

4. سپس یک نمودار Bar Plot ایجاد می شود. متغیر condition به عنوان متغیر مستقل (بر روی محور x) و recommendation به عنوان متغیر وابسته (بر روی محور y) در نظر گرفته می شود. رنگ‌های متفاوت (قرمز و آبی) برای نشان دادن مقادیر مختلف توصیه ها استفاده می شوند.

5. برچسب های محور x با چرخش 90 درجه قرار گرفته اند

:3



توضیح کد:

1. دیتا فریم با استفاده از تابع melt به یک فرمت ملت تبدیل می شود. این کار باعث می‌شود که مقادیر ستون های مختلف به ستون های یکتا تبدیل شوند. مقادیر این ستون‌ها در ستون های جدید مانند condition و recommendation ذخیره می شوند. ستون name نیز برای هر بیماری نگهداری می شود

2. اندازه شکل به عرض 50 اینچ و ارتفاع 5 اینچ تنظیم شده است.

3. با استفاده از دستور plt.subplot یک شبکه از نمودارها را ایجاد میشود

4. سپس یک نمودار Line Plot ایجاد می شود. متغیر condition به عنوان متغیر مستقل (بر روی محور x) و recommendation به عنوان متغیر وابسته (بر روی محور y) در نظر گرفته می شود. همچنین با استفاده از hue=name داده ها بر اساس نام بیماری ها رنگ‌آمیزی می شوند و با marker=o نقاط داده ها با نماد دایره نشان داده می شوند

5. برچسب های محور x با چرخش 90 درجه قرار گرفته اند