

شرح: این data set اطلاعاتی در مورد دوستداران قهوه و code developer ها ارائه می دهد.

این پروژه را در محیط jupyter پیاده سازی کردیم:

ابتدا نوار ابزار jupyter را بر روی markdown قرار میدهم و متن زیر را می نویسیم:

Coffee And Code

Roxana Haghgoo 9811309 - Horiya Asadsangabi 9811301

2022 - 1401

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می کنیم:

ابتدا کتابخانه های مورد نیازمان را که numpy، pandas و matplotlib.pyplot را وارد (import) می کنیم:

```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

کتابخانه numpy برای کار با آرایه ها استفاده می شود. همچنین دارای توابعی برای کار در حوزه جبر خطی، تبدیل فوریه و ماتریس ها است.

کتابخانه pandas برای کار با مجموعه داده ها استفاده می شود. دارای عملکردهایی برای تجزیه و تحلیل، پالایش، کاوش و دستکاری داده ها است و به ما این امکان را می دهند که داده های بزرگ را تجزیه و تحلیل کنیم و بر اساس تئوری های آماری نتیجه گیری کنیم. همچنین می توانند مجموعه داده های نامرتب را تمیز کنند و آنها را خوانا و مرتبط کنند. همچنین می توانند ردیف هایی را که مرتبط نیستند یا حاوی مقادیر اشتباه هستند، مانند مقادیر خالی یا NULL حذف کنند و اطلاعات زیر را نیز می تواند به ما بدهد:

- آیا بین دو یا چند ستون همبستگی وجود دارد؟

- مقدار متوسط چیست؟
- Max value (حداکثر ارزش)
- Min value (حداقل ارزش)

کتابخانه **matplotlib** یک کتابخانه متقابل پلتفرم، تجسم داده و رسم گرافیکی برای پایتون و پسوند عددی آن **NumPy** است. به این ترتیب، یک جایگزین منبع باز مناسب برای **MATLAB** ارائه می دهد. توسعه دهندگان همچنین می توانند از API های **matplotlib** (واسط های برنامه نویسی برنامه) برای جاسازی نمودارها در برنامه های رابط کاربری گرافیکی استفاده کنند. یک اسکریپت **matplotlib** پایتون به گونه ای ساختار یافته است که چند خط کد تنها چیزی است که در بیشتر موارد برای تولید نمودار داده بصری لازم است. لایه برنامه نویسی **matplotlib** دو API را پوشش می دهد:

- **pyplot** سلسله مراتبی از اشیاء کد پایتون است که در بالای آن **matplotlib.pyplot** قرار دارد.
- یک مجموعه OO (شی گرا) از اشیاء که می توانند با انعطاف پذیری بیشتری نسبت به **pyplot** مونتاژ شوند. این API دسترسی مستقیم به لایه های **Backend Matplotlib** را فراهم می کند.

ما از **pyplot** استفاده میکنیم.

❖ کلمه کلیدی **as** برای ایجاد نام مستعار استفاده می شود. در اینجا **np** را نام مستعار **pd.numpy** را نام مستعار

pandas و **plt** را نام مستعار **matplotlib.pyplot**

قرار میدهیم.

جهت نمایش گرافیکی به رنگ های متنوع نیاز داریم تا بهتر بتوانیم نتایج بدست آمده از مجموعه داده را مشاهده و تجزیه و تحلیل کنیم. به همین سبب با استفاده از کد هگزا دسیمال رنگ ها، رنگ های مورد نظر را به متغیرها انتساب می دهیم:

```
# declare settings
# colors :
color1 = '#303841' #greyish blue
color2 = '#30475E' #dark blue
color3 = '#EA5455' #Light Red
color4 = '#30475E'
color5 = '#D72323' #Red
color6 = '#FF6E31' #orange
color7 = '#0F4C75' #blue
color8 = '#E84545'
color9 = '#155263' #dark blue
color10 = '#FEA82F' #yellow
```

تا اینجا تنظیمات لازم را اعمال کردیم. مجدد نوار ابزار **jupyter** را بر روی **markdown** قرار میدهیم و متن زیر می -

نویسیم:

Getting To Know Our Data

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

با استفاده از متد `read_csv()` در کتابخانه `pandas` دایرکتوری فایل `dataset csv` را در پراپرتیز وارد می‌کنیم و محتویات فایل را در متغیر `df` می‌ریزیم.

```
#Loading the dataset
df = pd.read_csv('./datasets/Dataset01-CoffeeAndCode/CoffeeAndCode.csv')
```

سپس با کمک تابع `head` که بصورت پیش فرض 5 سطر اول را نمایش می‌دهد، دستور زیر را وارد می‌کنیم:

```
df.head()
```

و در نهایت 5 سطر اول را بعنوان خروجی تحویل می‌دهد.

	CodingHours	CoffeeCupsPerDay	CoffeeTime	CodingWithoutCoffee	CoffeeType	CoffeeSolveBugs	Gender	Country	AgeRange
0	8	2	Before coding	Yes	Caffè latte	Sometimes	Female	Lebanon	18 to 29
1	3	2	Before coding	Yes	Americano	Yes	Female	Lebanon	30 to 39
2	5	3	While coding	No	Nescafe	Yes	Female	Lebanon	18 to 29
3	8	2	Before coding	No	Nescafe	Yes	Male	Lebanon	NaN
4	10	3	While coding	Sometimes	Turkish	No	Male	Lebanon	18 to 29

خروجی:

با کمک تابع `show` تعداد سطر و ستون را بدست می‌آوریم:

```
# dataset shape (rows, columns)
df.shape
```

که اولین مولفه از سمت چپ تعداد سطرها و دومی تعداد ستون‌ها را نمایش می‌دهد.

خروجی:

```
(100, 9)
```

سپس با استفاده از تابع unique تعداد داده‌های خاص در هر ستون (attribute) را بدست می‌آوریم:

```
# number of unqiue data in each field  
df.nunique()
```

خروجی:

```
CodingHours      10  
CoffeeCupsPerDay   8  
CoffeeTime        7  
CodingWithoutCoffee 3  
CoffeeType        8  
CoffeeSolveBugs   3  
Gender            2  
Country           1  
AgeRange          5  
dtype: int64
```

اکنون که اطلاعات مورد نیاز را از استخراج کردیم، می‌خواهیم برای هر ستون یک نمایش گرافیکی ارائه دهیم:

Gender ■

(برای نمایش این ویژگی از نمودار pie (دایره‌ای) استفاده می‌کنیم.)

ابتدا نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر را می‌نویسیم:

Genders

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

همانطور که در خروجی بالا مشاهده کردیم در Gender 2 داده خاص داریم. برای نمایش این ویژگی از نمودار دایره‌ای استفاده می‌کنیم:

ابتدا با دستورات زیر تعداد زن‌ها و مردها را پیدا می‌کنیم و در متغیرهای mens و womens می‌ریزیم :

```
mens = df.loc[df['Gender']=='Male'].count()[0]
womens = df.loc[df['Gender']=='Female'].count()[0]
```

سپس با استفاده از دستور `plt.figure(figsize=(18,6))` یک شکل خالی با ابعاد 18×6 ایجاد می‌کنیم (اندازه‌ها به اینچ است):

```
plt.figure(figsize=(18,6))
```

با استفاده از `plt.pie()` می‌توانیم نمودار دایره‌ای ترسیم کنیم. ابتدا داده‌ها که همان `mens` و `womens` هستند را به صورت `[mens,womens]` وارد می‌کنیم و `label` هایی با عنوان `Men` و `Women` به آنها به صورت `labels=['Men','Women']` به آنها اختصاص می‌دهیم. سپس برای تعیین دقت اعداد و جهت بهتر شدن نمایش و راحتی کار مقدار `autopct` را `'%1d%%'` به معنای نمایش صحیح اعداد قرار می‌دهیم. در نهایت برای نمایش گرافیکی با توجه به دو داده متمایز به دو رنگ متمایز نیاز داریم برای این کار به ترتیب `color2` را جهت نمایش `Men` و `color3` را جهت نمایش `Women` استفاده می‌کنیم. (مقادیر `color2` و `color3` را ابتدای کد مشخص کردیم).

```
plt.pie([mens,womens],labels=['Men','Women'],autopct='%1d%',colors=[color2,color3]);
```

مقادیر `color` ها:

```
# declare settings
# colors :
color1 = '#303841' #greyish blue
color2 = '#30475E' #dark blue
color3 = '#EA5455' #light Red
color4 = '#30475E'
color5 = '#D72323' #Red
color6 = '#FF6E31' #orange
color7 = '#0F4C75' #blue
color8 = '#E84545'
color9 = '#155263' #dark blue
color10 = '#FEA82F' #yellow
```

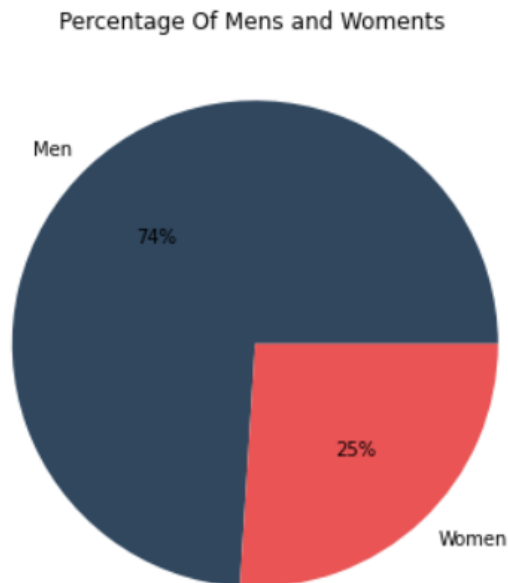
سپس با استفاده از `plt.title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما "Percentage Of Mens and Woments" را عنوان قرار می‌دهیم:

```
plt.title("Percentage Of Mens and Woments ")
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:



■ Age

(برای نمایش این ویژگی از نمودار bar (میله‌ای) استفاده می‌کنیم.)

ابتدا نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر را می‌نویسیم:

Ages

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

در این ویژگی تعدادی null داریم برای حذف null ها از دستور `df['AgeRange'].dropna()` استفاده می‌کنیم. چون می‌خواهیم برای null ها کل ردیف را حذف کنیم، `axis = 'index'` را داخل پرانتز می‌نویسیم و همچنین برای اینکه تغییر بر روی داده اعمال شود و خروجی جدید ندهد، دستور

`inplace=True` را نیز درون پرانتز وارد می‌کنیم:

```
df['AgeRange'].dropna( axis = 'index', inplace=True )
```

سپس براساس AgeRange گروه‌بندی می‌کنیم و در متغیر AgeRanges می‌ریزیم.

```
AgeRanges = df.groupby('AgeRange')
```

با استفاده از دستور AgeRangesCount = AgeRanges.size() تعداد در هر گروه را در متغیر AgeRangesCount می‌ریزیم:

```
AgeRangesCount = AgeRanges.size()
```

سپس برای مشاهده گروه‌های سنی و تعداد آنها AgeRangesCount را فراخوانی می‌کنیم:

```
AgeRangesCount
```

خروجی:

```
AgeRange
18 to 29    60
30 to 39    29
40 to 49     6
50 to 59     1
Under 18     2
dtype: int64
```

با استفاده از دستور x= df.groupby('AgeRange').groups.keys() AgeRange ها را کلید (که همان بازه‌های محور X هستند) قرار می‌دهیم و در متغیر X می‌ریزیم.

```
x= df.groupby('AgeRange').groups.keys()
```

با استفاده از دستور y=df.groupby('AgeRange')['AgeRange'].count() تعداد هر گروه را عرض‌های نمودار (مقادیر محور Y) قرار می‌دهیم:

```
y= df.groupby('AgeRange')['AgeRange'].count()
```

سپس با دستور fig, ax = plt.subplots() یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

برای رسم نمودار میله‌ای از دستور `ax.bar()` استفاده می‌کنیم. سپس مقادیر محور `X` و `Y` نمودار را که با توجه به دستورات بالا به ترتیب برابر `X` و `Y` هستند را وارد پرانتز می‌کنیم `ax.bar(x,y)`.

برای نمایش رنگ میله‌ها از `color1` استفاده می‌کنیم (در ابتدا مقدار `color1` را مشخص کردیم.) و با استفاده از `width` عرض هر میله را برابر `0.6` قرار می‌دهیم:

```
ax.bar(x,y, color=color1, width=0.6)
```

با استفاده از دستور `ax.bar_label(ax.containers[0], label_type='edge')` فراوانی هر یک از گروه‌های سنی را بر روی میله نوشته می‌شود.

```
ax.bar_label(ax.containers[0], label_type='edge')
```

با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما

Age Ranges Frequency را عنوان قرار می‌دهیم:

```
ax.set_title("Age Ranges Frequency")
```

با استفاده از `ax.set_xlabel()` می‌توانیم عنوانی را برای محور `X`ها در نظر بگیریم. در اینجا ما

Age Groups را عنوان قرار می‌دهیم:

```
ax.set_xlabel("Age Groups")
```

با استفاده از `ax.set_ylabel()` می‌توانیم عنوانی را برای محور `Y`ها در نظر بگیریم. در اینجا ما

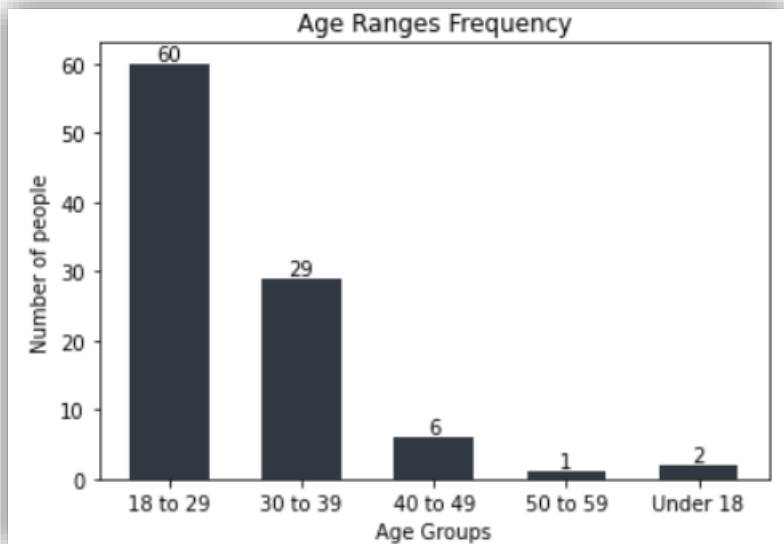
Number of people را عنوان قرار می‌دهیم:

```
ax.set_ylabel("Number of people")
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:



Coding Hours ■

(برای نمایش این ویژگی از هیستوگرام استفاده می‌کنیم.)

اکنون نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر را می‌نویسیم:

Coding Hours

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

ابتدا مقادیر پیوسته 1 تا 10 را به متغیری به نام bins انتساب می‌دهیم.

```
bins = [1,2,3,4,5,6,7,8,9,10]
```

با دستور `fig, ax = plt.subplots()` یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

برای رسم هیستوگرام از دستور `ax.hist()` استفاده می‌کنیم. سپس مقادیر `df['CodingHours']` و `bins` را که در `bins` می‌ریزیم درون پرانتز می‌نویسیم و برای نمایش رنگ میله‌ها از `color2` استفاده می‌کنیم (در ابتدا مقدار `color2` را مشخص کردیم).

```
ax.hist(df['CodingHours'],bins=bins,color=color2)
```

با استفاده از دستور `ax.bar_label(ax.containers[0], label_type='edge')` فراوانی هر یک از binها نوشته می‌شود.

```
ax.bar_label(ax.containers[0], label_type='edge')
```

با استفاده از دستور `ax.set_xticks(bins)` محور x را بعنوان محور bin قرار می‌دهیم:

```
ax.set_xticks(bins)
```

با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما

coding Hour را عنوان قرار می‌دهیم:

```
ax.set_title("coding Hour")
```

با استفاده از `ax.set_xlabel()` می‌توانیم عنوانی را برای محور x ها در نظر بگیریم. در اینجا ما

coding Hour را عنوان قرار می‌دهیم:

```
ax.set_xlabel("coding Hour")
```

با استفاده از `ax.set_ylabel()` می‌توانیم عنوانی را برای محور y ها در نظر بگیریم. در اینجا ما

Number of people را عنوان قرار می‌دهیم:

```
ax.set_ylabel("Number of people")
```

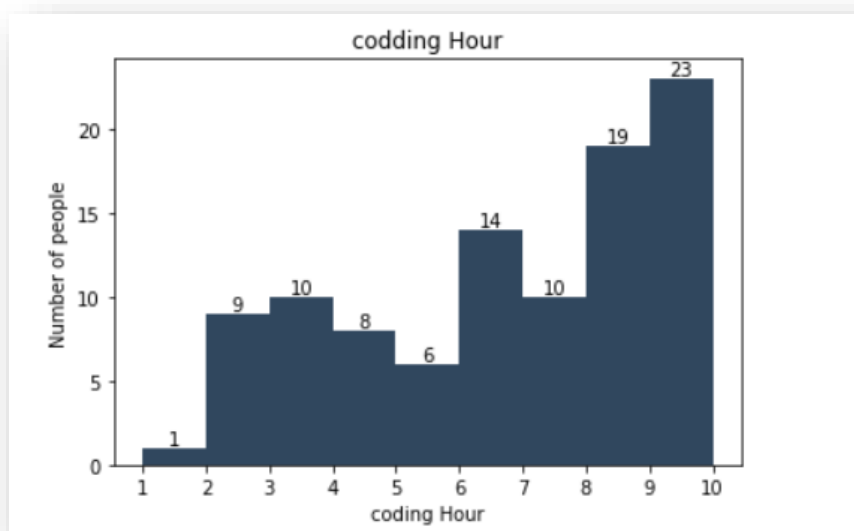
برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:

Coffee Cups ■
Per Day

(برای نمایش این ویژگی
از هیستوگرام استفاده می-
کنیم.)



اکنون نوار ابزار jupyter را بر روی markdown قرار میدهیم و متن زیر را می نویسیم:

Coffee Cups Per Day

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می کنیم:

ابتدا مقادیر زوج 0 تا 10 را به متغیری به نام bins انتساب می دهیم.

```
bins = [0,2,4,6,8,10]
```

با دستور fig, ax = plt.subplots() یک فضای خالی برای نمایش نمودار ایجاد می کنیم:

```
fig, ax = plt.subplots()
```

برای رسم هیستوگرام از دستور ax.hist() استفاده می کنیم. سپس مقادیر (df['CoffeeCupsPerDay']) و bins را که در bins می ریزیم درون پرانتز می نویسیم و برای نمایش رنگ میله ها از color2 استفاده می کنیم (در ابتدا مقدار color2 را مشخص کردیم).

```
ax.hist(df['CoffeeCupsPerDay'],bins=bins,color=color2)
```

با استفاده از دستور ax.bar_label(ax.containers[0], label_type='edge') فراوانی هر یک از bin ها نوشته می شود.

```
ax.bar_label(ax.containers[0], label_type='edge')
```

با استفاده از دستور ax.set_xticks(bins) محور x را بعنوان محور bin ها قرار می دهیم:

```
ax.set_xticks(bins)
```

با استفاده از ax.set_title() می توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما

Coffee Cups Per Day را عنوان قرار می دهیم:

```
ax.set_title("Coffee Cups Per Day")
```

با استفاده از ax.set_xlabel() می توانیم عنوانی را برای محور x ها در نظر بگیریم. در اینجا ما

Cups per Day را عنوان قرار می دهیم:

```
ax.set_xlabel("Cups per Day")
```

با استفاده از `ax.set_ylabel()` می‌توانیم عنوانی را برای محور y ها در نظر بگیریم. در اینجا ما

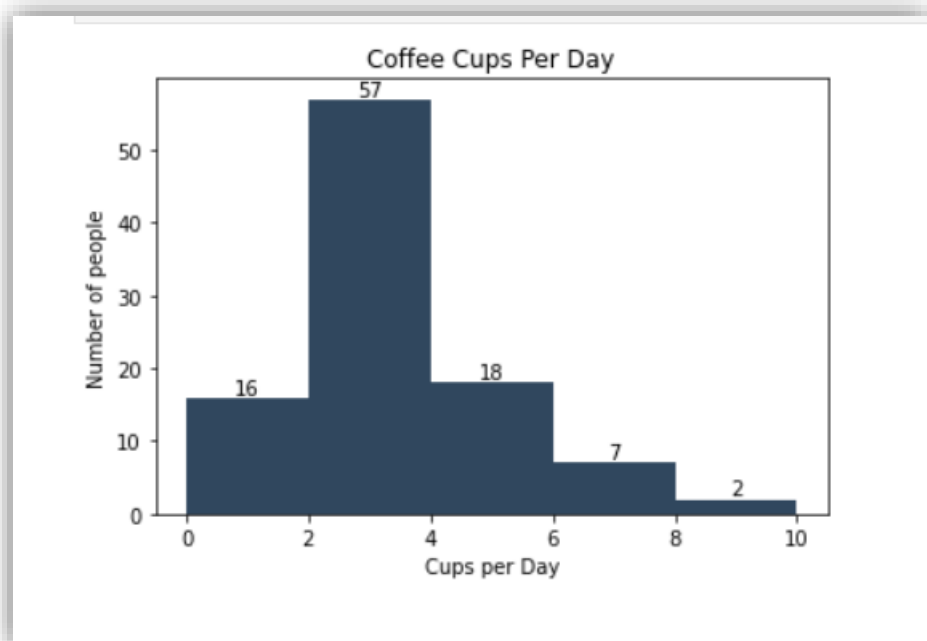
Number of people را عنوان قرار می‌دهیم:

```
ax.set_ylabel("Number of people")
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی



■ Coding Hours and Coffee Cups Per Day

(برای نمایش این ویژگی از `scatter` استفاده می‌کنیم.)

اکنون نوار ابزار `jupyter` را بر روی `markdown` قرار می‌دهیم و متن زیر را می‌نویسیم:

```
Coding Hours and Coffee Cups Per Day
```

سپس نوار ابزار را روی `code` قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

ابتدا مقادیر CodingHours را در متغیر CodingHours و مقادیر CoffeeCupsPerDay را در متغیر CoffeeCupsPerDay می‌ریزیم.

```
codingHours = df['CodingHours']  
cupsperday = df['CoffeeCupsPerDay']
```

با استفاده از دستور cupsperday.corr(codingHours) مقدار همبستگی بین CodingHours و CoffeeCupsPerDay را محاسبه می‌کنیم و در متغیر CorrelationValue می‌ریزیم.

```
CorrelationValue = cupsperday.corr(codingHours)
```

سپس CorrelationValue را فراخوانی می‌کنیم.

```
CorrelationValue
```

خروجی:

```
0.31369165125626997
```

سپس با دستور fig, ax = plt.subplots() یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

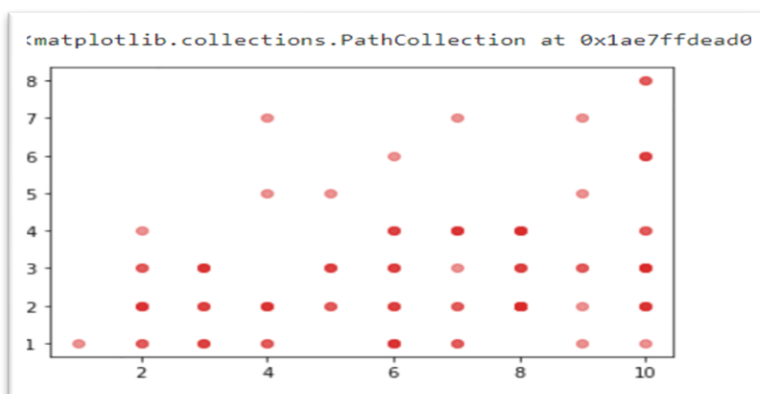
```
fig, ax = plt.subplots()
```

برای رسم هیستوگرام از دستور ax.scatter() استفاده می‌کنیم. سپس مقادیر

(df['CoffeeCupsPerDay'], df['CoffeeCupsPerDay']) و درون پرانتز می‌نویسیم و سپس برای نمایش رنگ از alpha برای تعیین شدت رنگ (کم‌رنگی و پررنگی) و از color5 استفاده می‌کنیم (در ابتدا مقدار color5 را مشخص کردیم).

مقدار alpha بین 0 و 1 است 1 به معنای پررنگ و 0 به معنای بی‌رنگ است)

```
ax.scatter(df['CodingHours'], df['CoffeeCupsPerDay'], alpha=0.5, color=color5)
```



خروجی:

Coding Without Coffee ■

(برای نمایش این ویژگی از نمودار pie (دایره‌ای) استفاده می‌کنیم.)

اکنون نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر را می‌نویسیم:

Coding Without Coffee

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

با استفاده از دستور `yes = df.loc[df['CoffeeSolveBugs']=='Yes'].count()[0]`

تعداد Yes ها را می‌شماریم و در متغیر yes می‌ریزیم.

با استفاده از دستور `no = df.loc[df['CoffeeSolveBugs']=='No'].count()[0]`

تعداد No ها را می‌شماریم و در متغیر no می‌ریزیم.

با استفاده از دستور

`sometimes = df.loc[df['CoffeeSolveBugs']==' Sometimes'].count()[0]`

تعداد Sometimes ها را می‌شماریم و در متغیر sometimes می‌ریزیم.

```
yes = df.loc[df['CodingWithoutCoffee']=='Yes'].count()[0]
no = df.loc[df['CodingWithoutCoffee']=='No'].count()[0]
sometimes = df.loc[df['CodingWithoutCoffee']=='Sometimes'].count()[0]
```

سپس با دستور `fig, ax = plt.subplots()` یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

با استفاده از `plt.pie()` می‌توانیم نمودار دایره‌ای ترسیم کنیم. ابتدا داده‌ها که همان yes، no و sometimes هستند را به صورت `[yes, no, sometimes]` وارد می‌کنیم و label هایی با عنوان Yes، No و sometimes به آنها به صورت

`labels=['Yes', 'No', 'sometimes']` به آنها اختصاص می‌دهیم. سپس برای تعیین دقت اعداد و جهت بهتر شدن

نمایش و راحتی کار مقدار `autopct` را به معنای نمایش صحیح اعداد قرار می‌دهیم. در نهایت برای نمایش

گرافیکی با توجه به سه داده متمایز به سه رنگ متمایز نیاز داریم برای این کار به ترتیب #243763، #579BB1

را جهت نمایش No و color7 جهت نمایش sometimes استفاده می‌کنیم. (مقدار color7 را ابتدای کد مشخص کردیم.)

```
ax.pie([yes, no, sometimes], labels=['Yes', 'No', 'sometimes'], autopct='%1d%%', colors=['#243763', '#5798B1', color7])
```

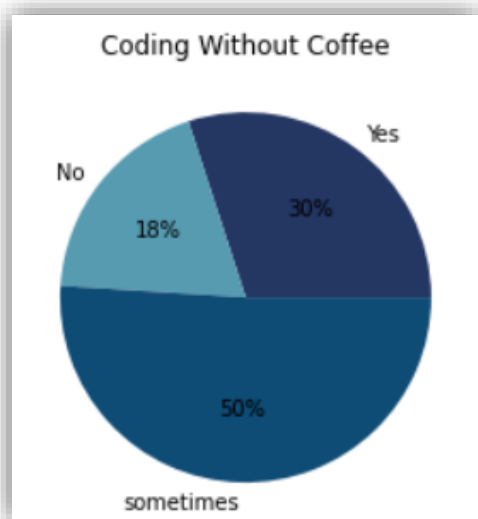
سپس با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما "Coding Without Coffee" را عنوان قرار می‌دهیم:

```
ax.set_title("Coding Without Coffee")
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:



■ Coffee Time

(برای نمایش این ویژگی از نمودار `bar` (میله‌ای) استفاده می‌کنیم.)

اکنون نوار ابزار `jupyter` را بر روی `markdown` قرار می‌دهیم و متن زیر را می‌نویسیم:

Coffee Time

سپس نوار ابزار را روی `code` قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

براساس `CoffeeTime` گروه‌بندی می‌کنیم و در متغیر `CoffeeTime` می‌ریزیم.

```
CoffeeTime = df.groupby('CoffeeTime')
```

با استفاده از دستور `CoffeeTimeCount = CoffeeTime.size()` تعداد در هر گروه را در متغیر `CoffeeTimeCount` می‌ریزیم:

```
CoffeeTimeCount = CoffeeTime.size()
```

سپس برای مشاهده گروه‌های `CoffeeTime` و تعدادشان، دستور `CoffeeTimeCount` را فراخوانی می‌کنیم:

```
CoffeeTimeCount
```

خروجی:

```
CoffeeTime
After coding      2
All the time      4
Before and while coding  4
Before coding     25
In the morning    3
No specific time  1
While coding     61
dtype: int64
```

با استفاده از دستور `x= df.groupby('CoffeeTime').groups.keys()` `CoffeeTime` ها را کلید (که همان بازه‌های محور `X` هستند) قرار می‌دهیم و در متغیر `X` می‌ریزیم.

```
x= df.groupby('CoffeeTime').groups.keys()
```

با استفاده از دستور `y=df.groupby('CoffeeTime')['CoffeeTime'].count()` تعداد هر گروه را عرض‌های نمودار (مقادیر محور `Y`) قرار می‌دهیم:

```
y= df.groupby('CoffeeTime')['CoffeeTime'].count()
```

سپس با دستور `fig, ax = plt.subplots()` یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

برای رسم نمودار میله‌ای از دستور `ax.bar()` استفاده می‌کنیم. سپس مقادیر محور `X` و `Y` نمودار را که با توجه به دستورات بالا به ترتیب برابر `X` و `Y` هستند را وارد پرانتز می‌کنیم `ax.bar(x,y)`.

برای نمایش رنگ میله‌ها از color2 استفاده می‌کنیم (در ابتدا مقدار color2 را مشخص کردیم).

```
ax.bar(x,y, color=color2)
```

با استفاده از دستور `ax.bar_label(ax.containers[0], label_type='edge')` فراوانی هر یک از گروه‌های سنی را بر روی میله نوشته می‌شود.

```
ax.bar_label(ax.containers[0], label_type='edge')
```

با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما

Coffee Time را عنوان قرار می‌دهیم:

```
ax.set_title("Coffee Time")
```

با استفاده از `ax.set_xlabel()` می‌توانیم عنوانی را برای محور Xها در نظر بگیریم. در اینجا ما

Times را عنوان قرار می‌دهیم:

```
ax.set_xlabel("Times")
```

با استفاده از `ax.set_ylabel()` می‌توانیم عنوانی را برای محور Yها در نظر بگیریم. در اینجا ما

Number of people را عنوان قرار می‌دهیم:

```
ax.set_ylabel("Number of people")
```

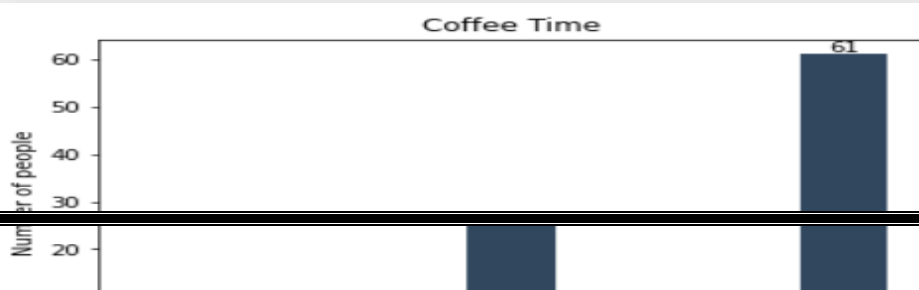
برای منظم تر شدن برچسب‌های محور X، آنها را 45 درجه به سمت راست دوران می‌دهیم برای این منظور از دستور `plt.xticks()` استفاده می‌کنیم و مقدار rotation را برابر 45 و ha را برابر 'right' قرار می‌دهیم.

```
plt.xticks(rotation=45, ha='right')
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:



■ Coffee Solve Bugs?

(برای نمایش این ویژگی از نمودار pie (دایره‌ای) استفاده می‌کنیم.)

اکنون نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر را می‌نویسیم:

```
Coffee Solve Bugs?
```

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

ابتدا با دستور `df['CoffeeSolveBugs'].unique()` لیستی از داده‌های خاص در `CoffeeSolveBugs` بدست می‌آوریم:

```
df['CoffeeSolveBugs'].unique()
```

خروجی:

```
array(['Sometimes', 'Yes', 'No'], dtype=object)
```

سه نوع داده خاص 'Sometimes', 'Yes', 'No' داریم.

با استفاده از دستور `yes = df.loc[df['CoffeeSolveBugs']=='Yes'].count()[0]`

تعداد Yes ها را می‌شماریم و در متغیر `yes` می‌ریزیم.

با استفاده از دستور `no = df.loc[df['CoffeeSolveBugs']=='No'].count()[0]`

تعداد No ها را می‌شماریم و در متغیر no می‌ریزیم.

با استفاده از دستور

```
sometimes = df.loc[df['CoffeeSolveBugs']==' Sometimes'].count()[0]
```

تعداد Sometimes ها را می‌شماریم و در متغیر sometimes می‌ریزیم.

```
yes = df.loc[df['CoffeeSolveBugs']=='Yes'].count()[0]
no = df.loc[df['CoffeeSolveBugs']=='No'].count()[0]
sometimes = df.loc[df['CoffeeSolveBugs']==' Sometimes'].count()[0]
```

سپس با دستور `fig, ax = plt.subplots()` یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

با استفاده از `plt.pie()` می‌توانیم نمودار دایره‌ای ترسیم کنیم. ابتدا داده‌ها که همان yes، no، و sometimes هستند را به صورت `[yes, no, sometimes]` وارد می‌کنیم و label هایی با عنوان Yes، No، و sometimes به آنها به صورت

`labels=['Yes', 'No', 'sometimes']` به آنها اختصاص می‌دهیم. سپس برای تعیین دقت اعداد و جهت بهتر شدن نمایش و راحتی کار مقدار `autopct` را `'%.1d%%'` به معنای نمایش صحیح اعداد قرار می‌دهیم. در نهایت برای نمایش گرافیکی با توجه به سه داده متمایز به سه رنگ متمایز نیاز داریم برای این کار به ترتیب #243763، Yes، #579BB1

را جهت نمایش No و color7 را جهت نمایش sometimes استفاده می‌کنیم. (مقدار color7 را ابتدای کد مشخص کردیم).

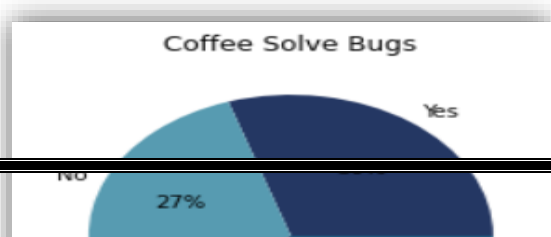
```
ax.pie([yes, no, sometimes], labels=['Yes', 'No', 'sometimes'], autopct='%1d%%', colors=['#243763', '#579BB1', color7])
```

سپس با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما 'Coffee Solve Bugs' را عنوان قرار می‌دهیم:

```
ax.set_title('Coffee Solve Bugs')
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```



▪ Coffee Type

(برای نمایش این ویژگی از نمودار bar (میله‌ای) استفاده می‌کنیم).

اکنون نوار ابزار jupyter را بر روی markdown قرار می‌دهیم و متن زیر می‌نویسیم:

Coffee Type

سپس نوار ابزار را روی code قرار داده و شروع به نوشتن کدهایمان می‌کنیم:

بر اساس CoffeeType گروه‌بندی می‌کنیم و در متغیر CoffeeType می‌ریزیم.

```
CoffeeType = df.groupby('CoffeeType')
```

با استفاده از دستور CoffeeTypeCount = CoffeeType.size() تعداد در هر گروه را در متغیر CoffeeTypeCount می‌ریزیم:

```
CoffeeTypeCount = CoffeeType.size()
```

سپس برای مشاهده گروه‌های CoffeeType و تعدادشان، دستور CoffeeTypeCount را فراخوانی می‌کنیم:

```
CoffeeTypeCount
```

```
CoffeeType
American Coffee    23
Americano           2
Caffè latte         5
Cappuccino          7
Double Espresso (Doppio)  3
Espresso (Short Black)  8
Nescafe            32
Turkish            19
dtype: int64
```

با استفاده از دستور `x= df.groupby('CoffeeType').groups.keys()` CoffeeType ها را کلید (که همان بازه‌های محور X هستند) قرار می‌دهیم و در متغیر X می‌ریزیم.

```
x= df.groupby('CoffeeType').groups.keys()
```

با استفاده از دستور `y=df.groupby('CoffeeType')['CoffeeType'].count()` تعداد هر گروه را عرض‌های نمودار (مقادیر محور Y) قرار می‌دهیم:

```
y= df.groupby('CoffeeType')['CoffeeType'].count()
```

سپس با دستور `fig, ax = plt.subplots()` یک فضای خالی برای نمایش نمودار ایجاد می‌کنیم:

```
fig, ax = plt.subplots()
```

برای رسم نمودار میله‌ای از دستور `ax.bar()` استفاده می‌کنیم. سپس مقادیر محور X و Y نمودار را که با توجه به دستورات بالا به ترتیب برابر X و Y هستند را وارد پرانتز می‌کنیم `ax.bar(x,y)`.

برای نمایش رنگ میله‌ها از `color9` استفاده می‌کنیم (در ابتدا مقدار `color9` را مشخص کردیم).

```
ax.bar(x,y, color=color9)
```

با استفاده از دستور `ax.bar_label(ax.containers[0], label_type='edge')` فراوانی هر یک از گروه‌های سنی را بر روی میله نوشته می‌شود.

```
ax.bar_label(ax.containers[0], label_type='edge')
```

با استفاده از `ax.set_title()` می‌توانیم عنوانی را برای نمودار در نظر بگیریم. در اینجا ما

Coffee Types را عنوان قرار می‌دهیم:

```
ax.set_title("Coffee Types")
```

با استفاده از `ax.set_xlabel()` می‌توانیم عنوانی را برای محور X ها در نظر بگیریم. در اینجا ما

Coffee Types را عنوان قرار می‌دهیم:

```
ax.set_xlabel("Coffee Types")
```

با استفاده از `ax.set_ylabel()` می‌توانیم عنوانی را برای محور Y ها در نظر بگیریم. در اینجا ما

Number of people را عنوان قرار می‌دهیم:

```
ax.set_ylabel("Number of people")
```

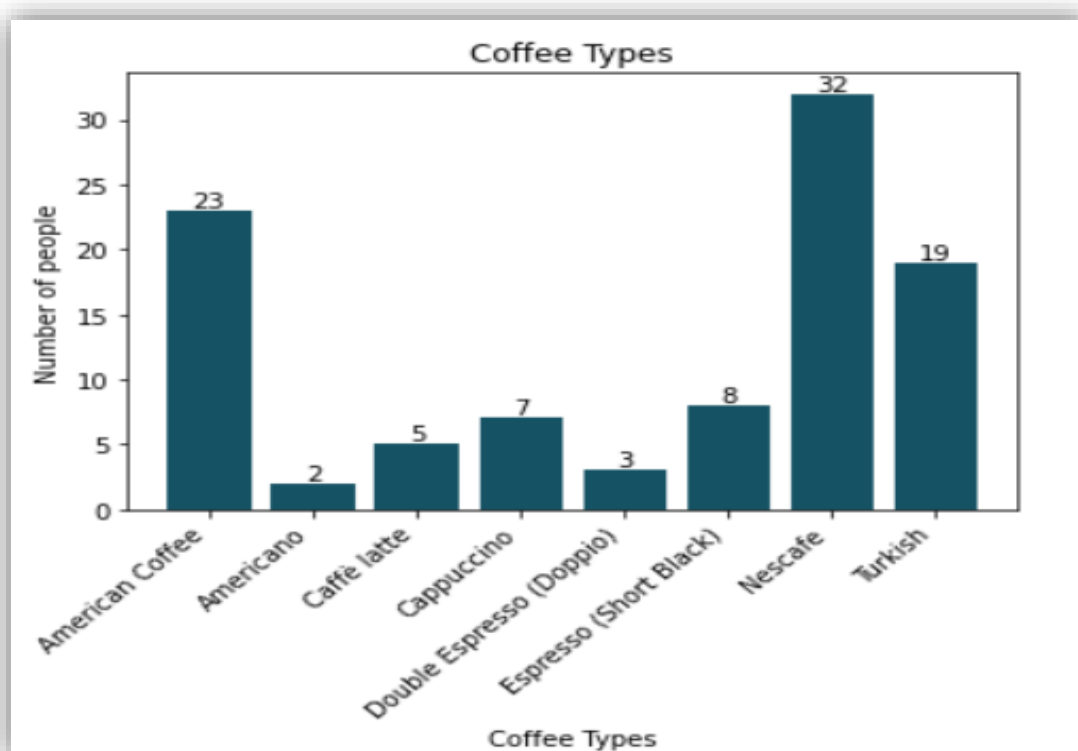
برای منظم تر شدن برچسب‌های محور X، آنها را 45 درجه به سمت راست دوران می‌دهیم برای این منظور از دستور `plt.xticks()` استفاده می‌کنیم و مقدار `rotation` را برابر 45 و `ha` را برابر 'right' قرار می‌دهیم.

```
plt.xticks(rotation=45, ha='right')
```

برای نمایش نمودار از دستور `plt.show()` استفاده می‌کنیم:

```
plt.show()
```

خروجی:



برای ایجاد مدل کتابخانه‌های زیر را `import` می‌کنیم:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

برای تبدیل داده‌های اسمی به عددی از کد زیر استفاده می‌کنیم:

```
copy_df = df
NumericDatas = df[['CodingHours', 'CoffeeCupsPerDay']]
nominalDatas = copy_df.drop(columns=['CodingHours', 'CoffeeCupsPerDay'])
nominalDatas = nominalDatas.apply(LabelEncoder().fit_transform)
allNumdf = pd.concat([NumericDatas, nominalDatas], axis=1)
allNumdf
```

خروجی:

```
Out[138]:
```

	CodingHours	CoffeeCupsPerDay	CoffeeTime	CodingWithoutCoffee	CoffeeType	CoffeeSolveBugs	Gender	Country	AgeRange
0	8	2	3	2	2	1	0	0	0
1	3	2	3	2	1	2	0	0	1
2	5	3	6	0	6	2	0	0	0
3	8	2	3	0	6	2	1	0	5
4	10	3	6	1	7	0	1	0	0
...
95	6	2	3	2	6	2	1	0	0
96	4	1	3	1	6	1	0	0	0
97	10	3	3	2	3	2	1	0	4
98	2	2	6	1	5	1	0	0	0
99	10	4	3	1	4	1	1	0	0

100 rows x 9 columns

داده‌ها را به بخش test و train با توجه به کد زیر تقسیم می‌کنیم:

```
x=allNumdf.drop(columns= ['CoffeeSolveBugs','CoffeeTime'])
y=allNumdf['CoffeeSolveBugs']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.15, random_state=45)
```

مدل را آموزش می‌دهیم:

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
# Train Decision Tree Classifier
clf = clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
```

برای نمایش accuracy, precision, recall از دستورات زیر استفاده می‌کنیم:

```
from sklearn.metrics import *
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.50	0.17	0.25	6
1	0.22	0.50	0.31	4
2	0.50	0.40	0.44	5
accuracy			0.33	15
macro avg	0.41	0.36	0.33	15
weighted avg	0.43	0.33	0.33	15