گزارش تمرین سری اول

درس داده کاوی

جناب آقای دکتر فراهانی و جناب آقای دکتر خرد پیشه

زینب خسروی۹۹۴۲۲۰۶۷

مقدمه

Data Mining **یا داده‌کاوی** به مفهوم استخراج اطلاعات نهان یا الگوها و روابط مشخص در حجم زیادی از داده‌ها دریک یا چند بانک اطلاعاتی بزرگ وتجزیه وتحلیل اطلاعات و پیش بینی اطلاعات گفته می‌شود.

داده‌کاوی، پایگاه‌ها ومجموعه حجیم داده ها را در پی کشف و استخراج، مورد تحلیل قرار می‌دهد. این‌گونه مطالعات و کاوش‌ها را به واقع می‌توان همان دانش همه جا گیرآمار دانست.یادگیری آماری یا تحلیل داده ها data science وعلم داده هابه معنای مشابه داده کاوی به کار می گیرد.

تفاوت عمده در مقیاس، وسعت و گوناگونی زمینه‌ها و.کاربردها، و نیز ابعاد و اندازه‌های داده‌های امروزین است که شیوه‌های ماشینی مربوط به یادگیری، مدل‌سازی، و آموزش را طلب می‌نماید.

داده کاوی به بهره‌گیری از ابزارهای تجزیه و تحلیل داده‌ها به منظور کشف الگوها و روابط معتبری که تاکنون ناشناخته بوده‌اند اطلاق می‌شود. این ابزارها ممکن است مدل‌های آماری، الگوریتم‌های ریاضی و روش‌های یاد گیرنده Machine Learning Methodsباشند. که کار این خود به صورت خودکار بر اساس تجربه ای که از طریق شبکه های عصبیNeural Networksیا درخت های تصمیم گیری Decision Treesبدست می آورند بهبود می بخشند.

الگو یا متدها:

الگوهایی که بر اساس آن یک رویداد به دیگری مربوط می‌شود.مثل خرید قلم به خرید کاغذ

الگویی که به تجزیه و تحلیل توالی رویدادها پرداخته و مشخص می‌کند کدام رویداد، رویدادهای دیگری را در پی دارد مثل تولد یک نوزاد و خرید پوشک

رگرسیون یکی دیگر از روش ها است.

classificationیا طبقه بندی: روشی برای پیدا کردن مدلی که داده ها را تعریف و متمایز می کند.با این هدف که از مدل برای پیش بینی داده هایی که برچسب آنها ناشناخته هستند استفاده می شود.

clustering یا خوشه بندی :گروه بندی که اعضای خوشه بیشترین شباهت وخوشه ها با هم کمترین شباهت دارند.

پیش‌بینی هدف اصلی داده کاوی می‌باشد. مانند پیش‌بینی نرخ ارز ولی تنها هدف نیست.

به کمک الگوریتم ها روابط چند بعدی بین داده ها تشخیص داده می شود به‌طور مثال در یک فروشگاه سخت‌افزار ممکن است بین خرید ابزار توسط مشتریان با تملک خانه شخصی یا نوع خودرو، سن، شغل، میزان درآمد یا فاصله محل اقامت آن‌ها با فروشگاه رابطه‌ای برقرار شود.

در تمرین داده کاوی دو مقدمه مهم است یکی فرمول واضحی از مشکل که قابل حل باشد و دیگری دسترسی به داده متناسب.یکی از معروف‌ترین ابزارهای داده‌کاوی برای انجام پروژه‌های داده کاوی پایتون هست.

در مرحله اول با آنالیز وتحلیل داده ها به اطلاعات می رسیم.تا آگاهی پیدا کنیم از سیستم تا به سوالات سیستم جواب بدهیم .که مدلی بسازیم که بتواند در آینده پیش بینی کند و تصمیم گیری انجام دهد.

اول ویژگی داده ها را استخراج میکنند وبعد به تحلیل داده های آماری می پردازند.بعضی داده ها خروجی مطلوب دارند دسته بندی شده اند یا نظارت شده اندو بعضی داده هابدون ناظر اند خروجی مطلوب نداریم کارهایی مثل خوشه بندی در آنها انجام می دهیم.

داده ها را که بشناسیم به فرضیاتی می رسیم که طی آزمون های فرض به درستی یا نادرستی پی می بریم.

آزمون های فرض مثلا درباره رابطه بین دو متغییردر موردش سوال می کنیم.مثل:t-test ,x2,fisher

t-test یکی ازروش های تجزیه وتحلیل داده ها آماری که در موردمیانگین جامعه آماری قضاوت می کند.

وقتی واریانس نداریم استفاده می کنیم وتوزیع اش نرمال هست.t,p-valuمحاسبه میکنیمp-valuکوچکتر ۰/۰۵احتمال فرضیه رد می شود و اگر بزرگتر۰/۰۵تایید می شود.

در این تمرین ها هدفمون این هست که سعی می کنیم داده ها را خوب بشناسیم تا بتوانیم با تحلیل های آماری به سوالات پاسخ دهیم ونتیجه بگیریم.

بررسی مجموعه داده اول همراه نمودار و کد پایتون

داده های جمع آوری شده از خانه های اجاره ای برای اقامت کوتاه درنیویورک آمریکا است. اطلاعاتی در مورد میزبان ها مهمان ها مکان اقامت زمان مدت اجاره قیمت اجاره ....جود دارد.

پیاده سازی در زبان درپایتون صورت گرفته است .در ابتدا لازم هست پکیج های مورد نظر فراخوانی کنیم.

کتابخانه پانداس ونامپای و مت پلایت وبه هر کدام یک اسم اختصاص می دهد.بعداطلاعات فایل بخواند و در متغییر دیتا قرار می دهد.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = pd.read_csv('/content/gdrive/MyDrive/AB_NYC_2019.CSV')
```

پرینت کند خروجی واسمش این قرار بده

```
print("The field name of data: ", data.columns)
print("Number of fields in data: ", len(data.columns))
print("Number of data in data: ", len(data))
```

```
The field name of data:  Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365'],
      dtype='object')
Number of fields in data:  16
Number of data in data:  48895
```

خروجی دستور اول نام ستون ها را داده و خروجی دستور دوم طول ستون ها است و خروجی دستور سوم طول کل داده است

بعد داده خواستم برامون آورده

اطلاعات داده خواستم بدست آورده که نوع داده ها را مشخص کرده چه تعدادnullهستند

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

نوع داده ها را داده

```
[ ]  data.dtypes
```

```
id                                 int64
name                              object
host_id                            int64
host_name                         object
neighbourhood_group               object
neighbourhood                     object
latitude                         float64
longitude                        float64
room_type                         object
price                              int64
minimum_nights                     int64
number_of_reviews                  int64
last_review                       object
reviews_per_month                float64
calculated_host_listings_count     int64
availability_365                   int64
dtype: object
```

تعدادnull ها را مشخص می کند.که پوچ هستندتا داده های که هیچ اطلاعاتی ندارند

و فرض می کنیم که هرگز مورد بررسی قرار نگرفته اند.وحذف می کنیم.

```
data.isnull().sum()
```

```
id                                    0
name                                 16
host_id                               0
host_name                            21
neighbourhood_group                   0
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                       10052
reviews_per_month                 10052
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

وبعد آخرین بررسی با بررسی نشده جایگزین کند .

```
[ ] data.fillna({'reviews_per_month':0}, inplace=True)
    data.fillna({'name':"NoName"}, inplace=True)
    data.fillna({'host_name':"NoName"}, inplace=True)
    data.fillna({'last_review':"NotReviewed"}, inplace=True)
```

```
[ ] visual_data = data.copy()
```

درصدهای مختلفی ازقیمت به عنوان خروجی گفتیم برای ما پرینت کند .

```
print("5%: ",visual_data.quantile(0.05)['price'])
print("25%: ", visual_data.quantile(0.25)['price'])
print("50%: ", visual_data.quantile(0.5)['price'])
print("75%: ", visual_data.quantile(0.75)['price'])
print("95%: ", visual_data.quantile(0.95)['price'])
```

```
5%:  40.0
25%:  69.0
50%:  106.0
75%:  175.0
95%:  355.0
```

داده هایnullاز بین رفتند.

```
data.isnull().sum()

id                                  0
name                                0
host_id                             0
host_name                           0
neighbourhood_group                 0
neighbourhood                       0
latitude                            0
longitude                           0
room_type                           0
price                               0
minimum_nights                      0
number_of_reviews                   0
last_review                         0
reviews_per_month                   0
calculated_host_listings_count      0
availability_365                    0
dtype: int64
```

ستون قیمت در داده ها راتوصیف کند.با استفاده از توابع شمارش کند چارک هاش بده مینیمم و ماکسیمم است قیمت ها را بدهدو میانگین واستاندارد در خروجی می دهد.و نوع داده قیمت هم float قیمت متوسط ۱۵۲ هزار

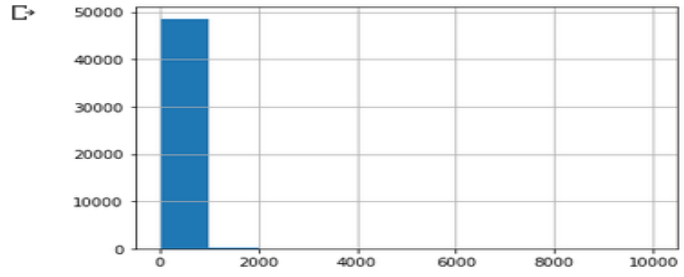
وقیمت از ۰ تا ۱۰ میلیارد متفاوت هست.

```
[ ] data["price"].describe()

count    48858.000000
mean       152.740309
std        240.232386
min          0.000000
25%         69.000000
50%        106.000000
75%        175.000000
max      10000.000000
Name: price, dtype: float64
```

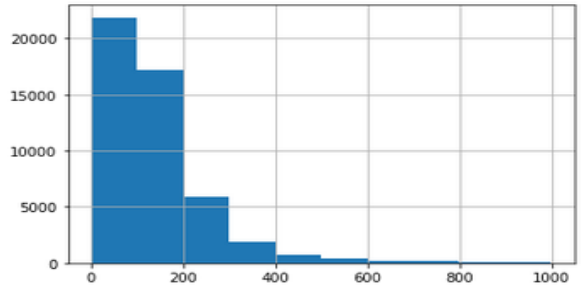
با این دستور قیمت به صورت نمودار ستونی در خروجی نشان می دهد.

```
hist_price=data["price"].hist()
```



نمودار ستون قیمت برای داده های کمتر از ۱۰۰۰۰ نشان می دهد.

```
hist_price1=data["price"][data["price"]<1000].hist()
```

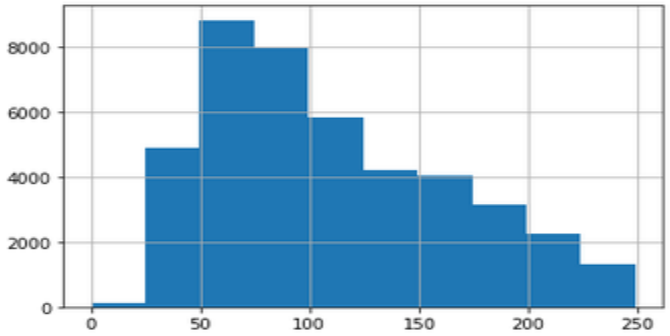


بعدچه تعدادی از داده های بیشتر از ۱۰۰۰ خواستیم

```
[ ]  data[data["price"]>1000]
```

۲۳۹ تا از داده ها قیمت شان کمتر۱۰۰۰هستندانها مارا منحرف می کنندآنها را دور دسترس قرار می دهیم یا رها میکنیم.که در اینجا توزیع گوسی هست.پس از۲۵۰ به عنوان قیمت آستانه استفاده می کنیم.

```
[ ]  hist_price2=data["price"][data["price"]<250].hist()
```



دوباره به ستون قیمت نگاه می کنیم داده ها را توصیف می کنیم با توابع

```
[ ]  data["price"].describe()

count    42635.000000
mean       107.897784
std         53.799361
min          0.000000
25%         65.000000
50%         99.000000
75%        150.000000
max        249.000000
Name: price, dtype: float64
```

می بینیم که متوسط قیمت۱۰۷ وقیمت از۰تا ۲۴۹متغییر هست

طبق خروجی زیر در مجموعه داده ها۲۲۱ محله منحصر به فرد وجود دارد

```
[ ] data['neighbourhood'].value_counts()
```

```
Bedford-Stuyvesant    3559
Williamsburg          3448
Harlem                2485
Bushwick              2401
Upper West Side       1568
                      ...
Richmondtown             1
New Dorp                 1
Rossville                1
Neponsit                 1
Willowbrook              1
Name: neighbourhood, Length: 219, dtype: int64
```

تعداد محله های بیش از ۲۰۰ بشماردوبعد طول اش را بدهد.

```
[ ] data['neighbourhood'].value_counts()
```

```
Bedford-Stuyvesant    3559
Williamsburg          3448
Harlem                2485
Bushwick              2401
Upper West Side       1568
                      ...
Richmondtown             1
New Dorp                 1
Rossville                1
Neponsit                 1
Willowbrook              1
Name: neighbourhood, Length: 219, dtype: int64
```

```
[ ] len(data["neighbourhood"])
```

```
42635
```

حساب کند که چند محله فقط یکبار ظاهر شدند

```
[ ] data =data.groupby("neighbourhood").filter(lambda x: x['neighbourhood'].count() == 1)
    len(data["neighbourhood"])
```

```
5
```

به گروه های محله نگاه کنیم

```
data['neighbourhood_group'].value_counts()
```

```
Staten Island    4
Queens           1
Name: neighbourhood_group, dtype: int64
```

این دو محله۸۵ درصدبه خودشون اختصاص دادند.

میانگین قیمت بر اساس گروه محله ببینیم

```
ng_price=data.groupby("neighbourhood_group")["price"].mean()
```

```
ng_price
```

```
neighbourhood_group
Queens           200.00
Staten Island    114.75
Name: price, dtype: float64
```

Queensهزینه بیشتری از staten Island دارد.

بررسی می کنیمhost idتکراری هست حد اکثر تعداد چقدر هست

```
df = data.groupby(["host_id"])
max(df.size())
```

```
1
```

که سایزشون به صورت زیر است.

```
[ ]  df.size().value_counts().head()
```

```
1    5
dtype: int64
```

```
[ ]  df.size().value_counts().tail()
```

```
1    5
dtype: int64
```

یافتنid میزبان با حداکثر در لیست

```
host_id_counts = data["host_id"].value_counts()
max_host = host_id_counts.idxmax()
max_host
```

```
188328775
```

که با این دستور برای ما می آورد

```
data[data["host_id"]==188328775]
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood
31912	24910361	"The Little House by the Sea"	188328775	Donna	Queens	Neponsit

شناسه ذکر شده و نام میزبان برای تجزیه و تحلیل ما مفید نیستند ، بنابراین آنها را رها می کنم

```
[ ]  data = data.drop(columns = ["id","host_name"])
```

بیایید ستون نام لیست را تجزیه و تحلیل کنیم

```
[ ]  data["name_length"]=data['name'].map(str).apply(len)
```

حداکثر وحداقل طول نام

```
[ ]  print(data["name_length"].max())
     print(data["name_length"].min())
     print(data["name_length"].idxmax())
     print(data["name_length"].idxmin())
```

```
47
20
16035
30489
```

حداکثر نام

```
[ ]  data.at[16035, 'name']
```

```
'1 bedroom apt, comforts of home, close to all..'
```

حداقل نام

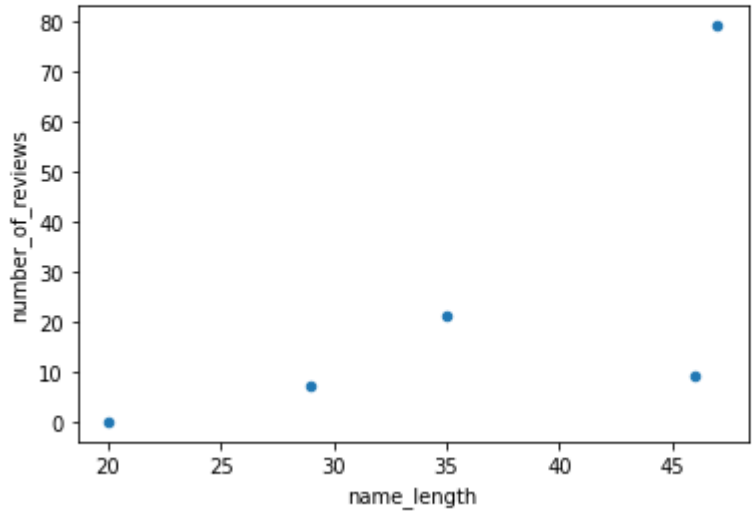
```
data.at[30489, 'name']
```

```
'Staten Island Studio'
```

بیایید بفهمیم آیا طول نام در میزان توجه آن تاثیر دارد می توانیم فرض کنیم که افراد بیشتری در انجا زندگی می کنند

```
data.plot.scatter(x="name_length", y ="number_of_reviews" )
```

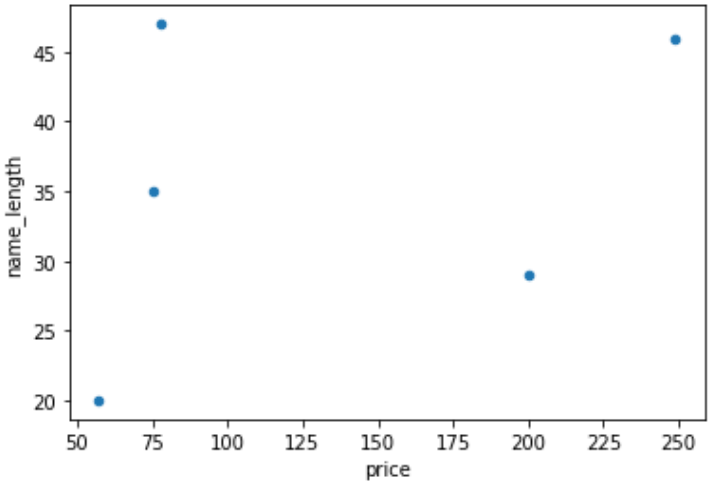
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb44c46fe90>
```



آیا طول نام با قیمت رابطه ای دارند

```
[ ]  data[data["name_length"]<50].plot.scatter(x="price", y ="name_length")
```

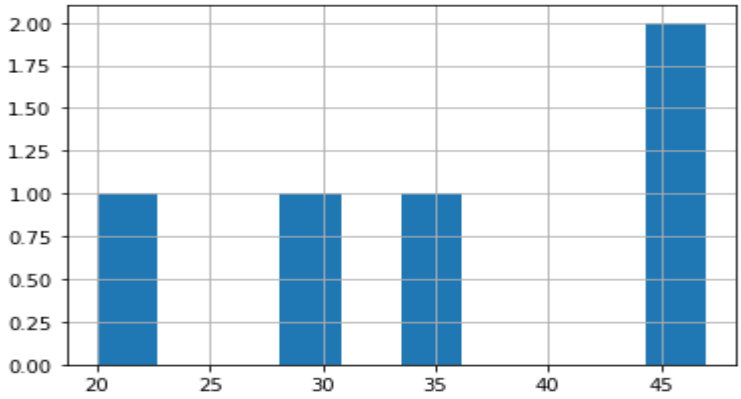
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc921e53150>
```



نمودار طول نام

```
[ ]  data.name_length.hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc921dcc790>
```



به انواع خانه ها نگاه کنیم کل خانه هایا اتاق ها خصوصی هستند.

```
[ ]  data['room_type'].value_counts()
```

```
Entire home/apt    5
Name: room_type, dtype: int64
```

متوسط قیمت هر اتاق یا خانه به ای صورت هست.

```
[ ]  rt_price = data.groupby("room_type")["price"].mean()
```

```
[ ]  rt_price
```

```
room_type
Entire home/apt    131.8
Name: price, dtype: float64
```

توصیف وتجزیه تحلیل حداقل شب

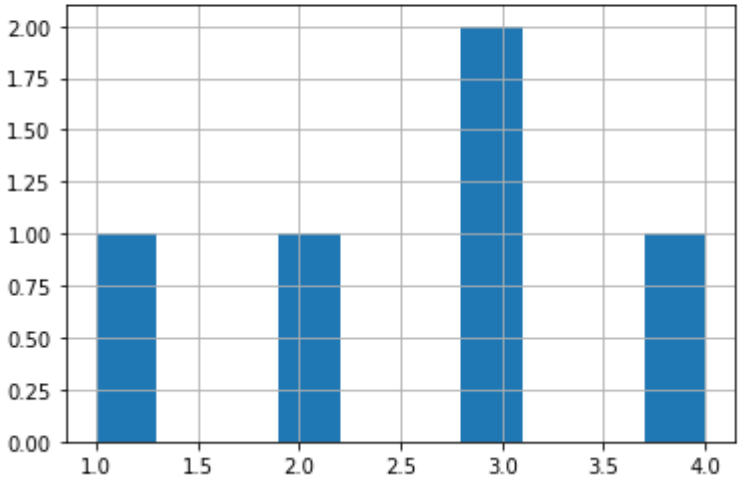
```
[ ]  data["minimum_nights"].describe
```

```
<bound method NDFrame.describe of 16035     3
30489     1
31912     2
33261     3
34161     4
Name: minimum_nights, dtype: int64>
```

دامنه شب هایی که خونه ها اجاره داده می شوند در نمودار به صورت زیر است

```
hist_mn=data["minimum_nights"].hist()
hist_mn
```

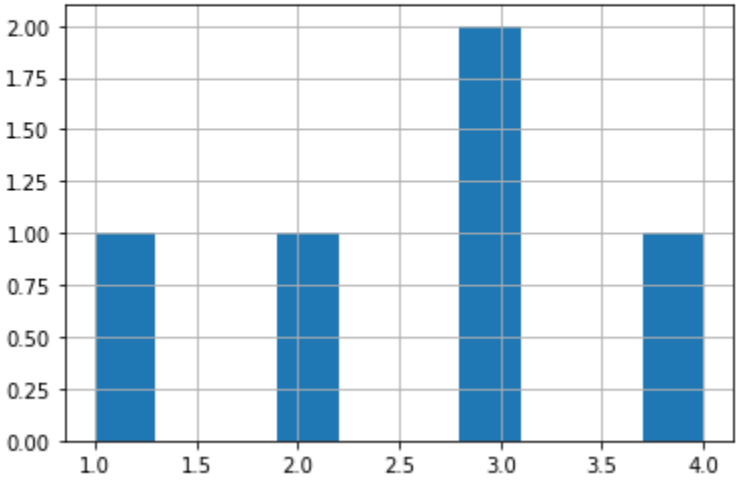
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc921dd0650>
```



بررسی دقیقتربا حداقل ده شب

```
[ ]  hist_mn1=data["minimum_nights"][data["minimum_nights"]<10].hist()
     hist_mn1
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc921c8e390>
```



همه اطلاعات با حداقل ۳۰ شب جایگزین می کنیم .

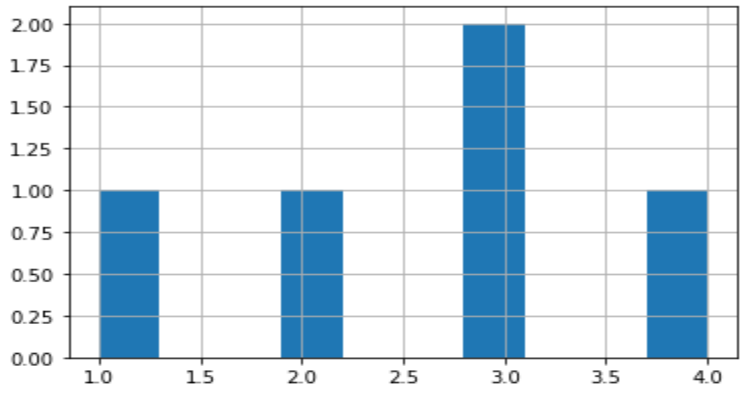
```
[ ]  data["minimum_nights"][data["minimum_nights"]>30]
```

```
Series([], Name: minimum_nights, dtype: int64)
```

```
[ ]  data.loc[(data.minimum_nights >30),"minimum_nights"]=30
```

```
[ ]  hist_mn2=data["minimum_nights"][data["minimum_nights"]<30].hist()
     hist_mn2
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc921c127d0>
```



کمترین شب ها روی قیمت تاثیر دارند

```
[ ]  data["minimum_nights"].corr(data["price"])
```

```
0.5019657965291167
```

ستون اجاره ای تجزیه تحلیل می کنیم

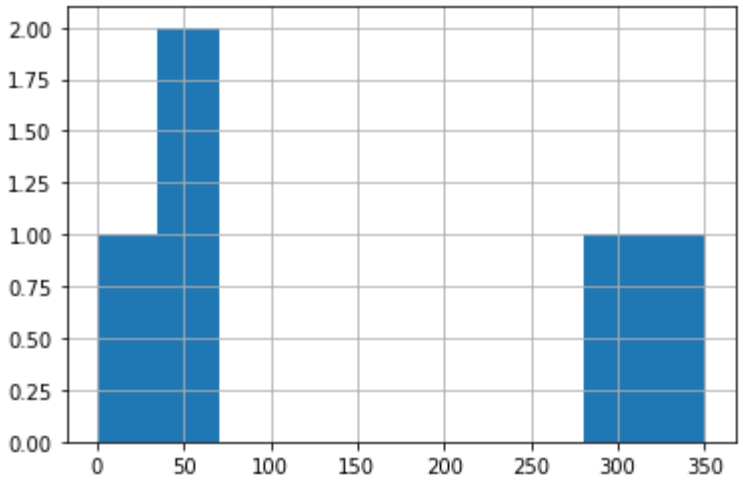
```
[ ]  data["availability_365"].describe()
```

```
count      5.000000
mean     150.800000
std      161.952771
min        0.000000
25%       44.000000
50%       59.000000
75%      300.000000
max      351.000000
Name: availability_365, dtype: float64
```

نمودار ستون اجاره

```
[ ]  hist_av=data["availability_365"].hist()
     hist_av
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc9239d6650>
```



ماتریس همبستگی

```
[ ]  corr = data.corr(method='pearson')
     plt.figure(figsize=(15,8))
     data.columns
```

```
Index(['host_id', 'neighbourhood_group', 'neighbourhood', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365', 'name_length'],
      dtype='object')
<Figure size 1080x576 with 0 Axes>
```

قبل از شروع پیش بینی

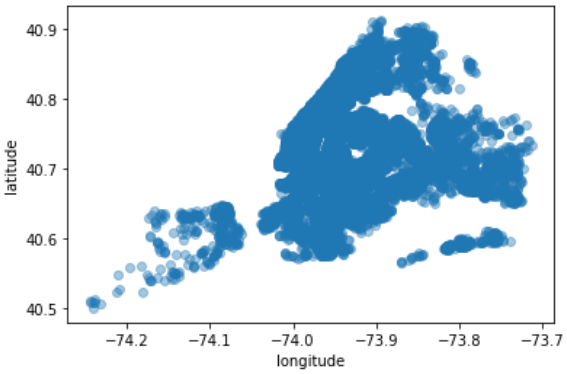
داده ها را یکبار بررسی می کنیم که خوب هستند یا نه

```
[ ]  data.dtypes
```

```
host_id                            int64
neighbourhood_group               object
neighbourhood                     object
room_type                         object
price                              int64
minimum_nights                     int64
number_of_reviews                  int64
reviews_per_month                float64
calculated_host_listings_count     int64
availability_365                   int64
name_length                        int64
dtype: object
```

داده های بزرگتر از۹۵٪قیمت حذف می شوند یعنی فقط داده های کمتر ازمقدار ۹۵٪ باقی می مانند.

```
data_price_95 = visual_data[visual_data['price'] <= visual_data.quantile(0.95)['price']]
plt.scatter(data_price_95['longitude'], data_price_95['latitude'], cmap=plt.get_cmap('jet'), alpha=0.4)
plt.xlabel('longitude')
plt.ylabel('latitude')
plt.show()
```



برای ایجادیک مدل خطی با رگرسیون پیش می رویم اجازه دهید بدون ستون محله ادامه دهیم

```
[ ] data_onehot1 = pd.get_dummies(data, columns=['neighbourhood_group',"room_type"], prefix = ['ng',"rt"],drop_first=True)
    data_onehot1.drop(["neighbourhood"], axis=1, inplace=True)
```

```
[ ] data_onehot1.shape
```

```
(5, 9)
```

از متغییر همسایگی استفاده کردیم که مقدار مشخص دارند و تعداد زیادی متغییر خواهیم داشت.

```
[ ] data_onehot2 = pd.get_dummies(data, columns=['neighbourhood_group',"neighbourhood","room_type"], prefix = ['ng',"nh","rt"],drop_first=True)
```

```
[ ] data_onehot2.shape
```

```
(5, 13)
```

```
[ ] XL1= data_onehot2.loc[:, data_onehot2.columns != 'price']
    YL1 = data_onehot2["price"]
```

ترین کردن وتست داده ها

```
[ ]  from sklearn.model_selection import RandomizedSearchCV

     n_estimators = [int(x) for x in np.linspace(start = 200, stop = 1000, num = 5)]
     max_features = ['auto', 'sqrt']
     max_depth = [int(x) for x in np.linspace(10, 110, num = 6)]
     max_depth.append(None)
     min_samples_split = [2, 5, 10]
     min_samples_leaf = [1, 2, 4]
     bootstrap = [True, False]
     # Create the random grid
     rm_grid = {'n_estimators': n_estimators,
                    'max_features': max_features,
                    'max_depth': max_depth,
                    'min_samples_split': min_samples_split,
                    'min_samples_leaf': min_samples_leaf,
                    'bootstrap': bootstrap}
```

برای جستجوی بهتراز پارامتر ها تصادفی یا رندم استفاده می کنیم

```
print(rm_grid)
```

```
{'n_estimators': [200, 400, 600, 800, 1000], 'max_features': ['auto', 'sqrt'], 'max_depth': [10, 30, 50, 70, 90, 110, None], 'min_samples_split': [2, 5, 10
```

```
import time
```

```
t1 = time.time()
t2 =time.time()
(t2-t1)/60
```

```
3.2981236775716145e-07
```

پاسخ سوالات مجموعه داده اول

در مورد میزبان ومناطق مختلف چه چیزهایی می توانیم یاد بگیریم؟

در دو محله۸۵٪میزبان های بیشتری را به خودشان اختصاص دادند.

Staten Island va Queens

از پیش بینی ها چه می توانیم یاد بگیریم؟

پیش بینی اینکه طول نام در میزان توجه آنها تاثیر ندارد و فرض اینکه بیشتر در آنجا زندگی می کنند رد می شود .(طبق نمودار)

پیش بینی اینکه آیا طول نام با قیمت رابطه ای دارد هم رد می شود (طبق نمودار)

کدام مناطق شلوغ ترین است چرا؟

مناطق شرقی چون با توجه به نقشه جغرافیایی و طول و عرض جغرافیایی داده شده بیشتر در مناطق شرقی توزیع بیشتر وتراکم بیشتر است .

آیا تفاوت قابل توجهی در ترافیک در مناطق مختلف وجود دارد و چه چیزی می تواند باشد؟

مناق که شلوغ تر تراکم بیشتر ترافیک بیشتر قیمت کمتر هست.(با توجه به اطلاعات در داده های بررسی شده)

سوالات ابتکاری مجموعه داده اول

محله ای که بیشترین هزینه دارد؟Queens

محله ای که شلوغ تر هست ؟staten Island

متوسط قیمت وحد اکثر قیمت ؟۱۰۷ و۲۴۹

از طول نام ونوع خانه ها چه می توان فهمید؟ کل خانه ها خصوصی هستند

دامنه شب ها با قیمت خانه با هم رابطه دارند یا روی هم تاثیر می گذارند؟بله روی هم تاثیر می گذارند

نتیجه گیری:پس محله مکان جغرافیایی طول نام دامنه شب ترافیک ..روی قیمت اجاره تاثیر گذارند.

بررسی مجموعه داده دوم همراه نمودار و کد پایتون

داده های مربوط به مسابقات فوتبال بین المللی که در مسابقات جام جهانی جام قاره های تورنمنت ها بازی های دوستانه...انجام شده است . به این منظور اطلاعات مختلفی از قبیل نام تیم ها محل انجام مسابقه میزبان ومهمان و زمان بازی و تعداد گل و نتیجه بازی ذخیره شده اند .

پیاده سازی در زبان درپایتون صورت گرفته است .در ابتدا لازم هست پکیج های مورد نظر فراخوانی کنیم.

کتابخانه پانداس ونامپای و مت پلایت وبه هر کدام یک اسم اختصاص می دهد.بعداطلاعات فایل بخواند و در متغییر دیتا قرار می دهد.

```
[ ]  from google.colab import drive
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     data = pd.read_csv('/content/drive/MyDrive/results.csv')
```

پانداس فراخوانی میکند داده ها را وارد می کند مت پلایت فراخوانی میکند داده تعریف میکند .

```
import pandas as pd
import os
for dirname, _, filenames in os.walk('/content/gdrive/MyDrive/results.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import matplotlib.pyplot as plt
```

```
df_football = pd.read_csv('/content/drive/MyDrive/results.csv')
```

- چه روندی در فوتبال بین المللی در طول سالها وجود داشته و برتری خانه ها و گل های زده شده وتوزیع قدرت تیم ها؟

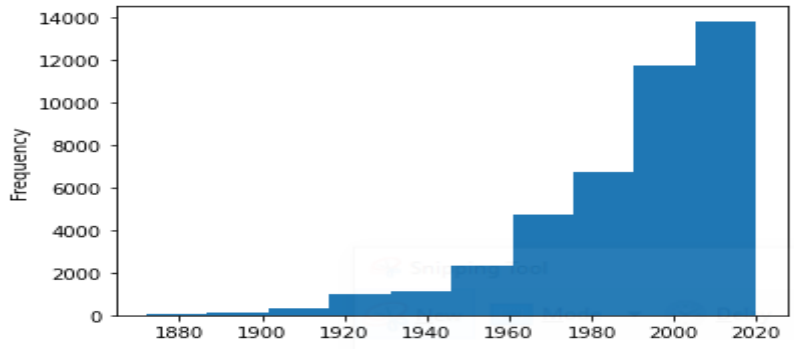
جواب سوال:

پرینت می کند داده های سال که منحصر به فرد هستندانواع آنها تعدادش چند تا هست ودر نمودار نشان بده.

درطول سالها تیم هایی که میزبان بودند برد بیشتری داشتند تعداد برد ها نسبت به تعداد میزبانی ها نشان می دهد که تیم هایی که میزبان هستند برد بیشتری داشتند.پس گل های بیشتری زده می شود وقدرت تیم بیشتر میشود

```
[ ]  df_football['date'] = pd.to_datetime(df_football['date'])
     df_football['year'] = df_football['date'].dt.year
     print(df_football['year'].nunique())
     _=df_football['year'].plot(kind='hist')
```

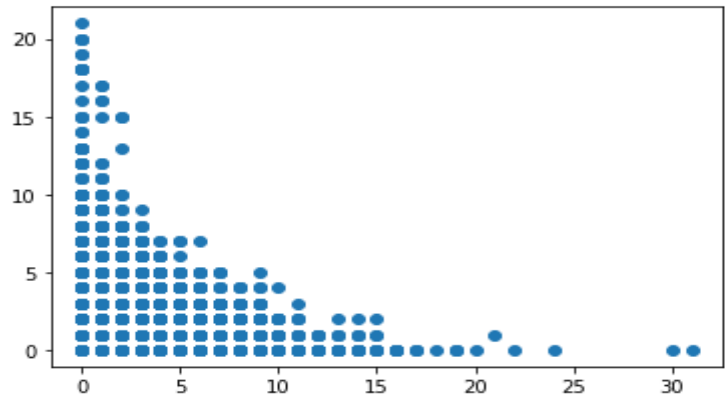
```
149
```



پراکندگی امتیاز تیم میزبان به امتیاز تیم های دیگربرای ۳۰۸ تیم به صورت نمودار زیر هست

امتیاز اینکه میزبان باشند و برنده شوند نسبت به تیم های دیگر

```
[ ]  _=plt.scatter(df_football['home_score'], df_football['away_score'
```




```
[ ]  df_football['home_team'].nunique()
```

```
308
```

value counts برای ۳۰۸ تیم میزبان به این صورت تعریف می شود .که برتری در خانه نشان داده

```
[ ]
df_football['home_team'].value_counts()
```

```
Brazil            570
Argentina         550
Mexico            515
Germany           511
England           503
                 ...
Crimea              1
Chameria            1
Romani people       1
Saint Helena        1
Sark                1
Name: home_team, Length: 308, dtype: int64
```

برزیل برتری بیشتری دارد بعد آرژانتین بعد مکزیک وآلمان وانگلیس پنج تیم که بیشترین برد به عنوان میزبان داشتند وبیشترین گل ها را زدند وقدرت بیشتری نسبت به تیم های دیگر دارند.

پنج کشوری که بیشترین برد در کل بازی های دوستانه به عنوان میزبان داشته اند؟

جواب: برزیل-آرژانتین-مکزیک-آلمان-انگلیس

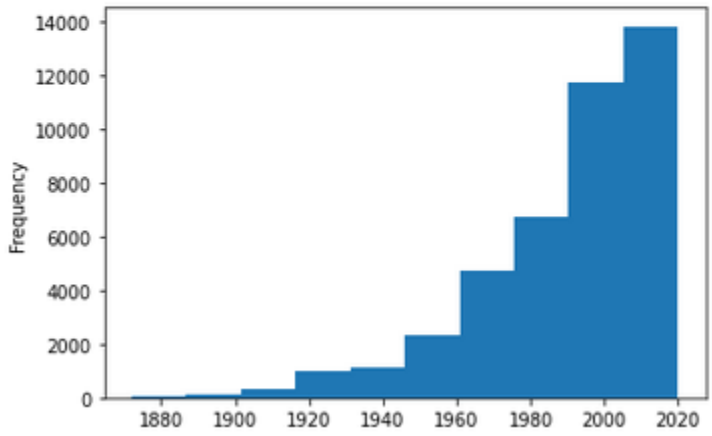
-آیا می توانیم در مورد ژئوپلیتیک از مسابقات فوتبال چیزی بگوییم که چه تعداد کشور تغییر کرده وتیم هایی که دوست دارند با هم بازی کنند.؟

جواب سوال :

در خصوص مناطق جغرافیایی صحبت کرده مثلا در آسیا چه تیم هایی هستند وبیشترین برد در آسیا برای کدام تیم ها بوده است و تیم های اروپاکدام هستند وبیشترین برد کدامش بوده با هم بررسی کنیم. ودر طول سالها کدام کشور ها تغییر کرده اند value counts بزنیم بعدuniqueبزنیم و برای سالها خروجی بگیریم کدام کشور ها شرکت کردند کدام کشور ها اضافه یا کم شده اندvalue counts روی کل دیتا فراوانی تیم ها چقدر بوده چقدر بازی کرده اند و تعداد دفعات هر بازی چقدر بوده کم یا زیاد ونشان می دهد چه تیم هایی دوست دارند با یکدیگر بازی کنند.

```
df_football['date'] = pd.to_datetime(df_football['date'])
df_football['year'] = df_football['date'].dt.year
print(df_football['year'].nunique())
_=df_football['year'].plot(kind='hist')
```

```
149
```



از سال۱۸۸۰تا۲۰۲۰ تیم ها در ابتدا کم بودندبازی ها کم بودند در طول سالهازیاد شدند.

❋در صورت وجود میزبانی یک تورنومنت بزرگ چقدر شانس وجود دارد که یک کشور در مسابقات برنده شود یانه؟

جواب سوال:

جام جهانی های مختلف با هم مقایسه کنیم کدام تیم میزبان بوده در جام جهانی و ان سال تیم چه رتبه ای کسب کرده وآیا میزبان بودن کمک کرده تا تیم میزبان قهرمان شودیا نه

مثلا بازی هایی که بین انگلیس و اسکاتلند رخ داده است بررسی میکنیم.که اسکاتلند میزبان بوده وانگلیس مهمان بوده است.تا در جام جهانی های مختلف و سال های مختلف مقایسه شود .

که همان طوردر زیر مشاهده میکنید اسکاتلند که میزبان بوده امتیاز بیشتری کسب کرده است.

```
df_football[(df_football['home_team']=='Scotland')&(df_football['away_team']=='England')]
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	year
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False	1872
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False	1874
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False	1876
8	1878-03-02	Scotland	England	7	2	Friendly	Glasgow	Scotland	False	1878
13	1880-03-13	Scotland	England	5	4	Friendly	Glasgow	Scotland	False	1880
21	1882-03-11	Scotland	England	5	1	Friendly	Glasgow	Scotland	False	1882
32	1884-03-15	Scotland	England	1	0	British Championship	Glasgow	Scotland	False	1884
45	1886-03-27	Scotland	England	1	1	British Championship	Glasgow	Scotland	False	1886
58	1888-03-17	Scotland	England	0	5	British Championship	Glasgow	Scotland	False	1888
73	1890-04-05	Scotland	England	1	1	British Championship	Glasgow	Scotland	False	1890
85	1892-04-02	Scotland	England	1	4	British Championship	Glasgow	Scotland	False	1892

کدام تیم ها در بازی های دوستانه بیشترین فعالیت داشتندو آیا به آنها کمک می کند یا به آنها زیان می رساند؟

```
[98] temp=df_football[df_football['tournament']=='frinedly']
```

```
[99] temp['home_win']=(temp['home_score']>temp['away_score'])
```

```
[100] t=temp[temp['home_win']]==True
```

```
df_football['home_team'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0             Scotland
1              England
2             Scotland
3              England
4             Scotland
              ...
41871           Greece
41872          Albania
41873       Kazakhstan
41874            Qatar
41875    United States
Name: home_team, Length: 41876, dtype: object>
```

تیم هایی که بیشترین فعالیت داشتند برنده می شدند در بازی های دوستانه و به آنها کمک می کرد.

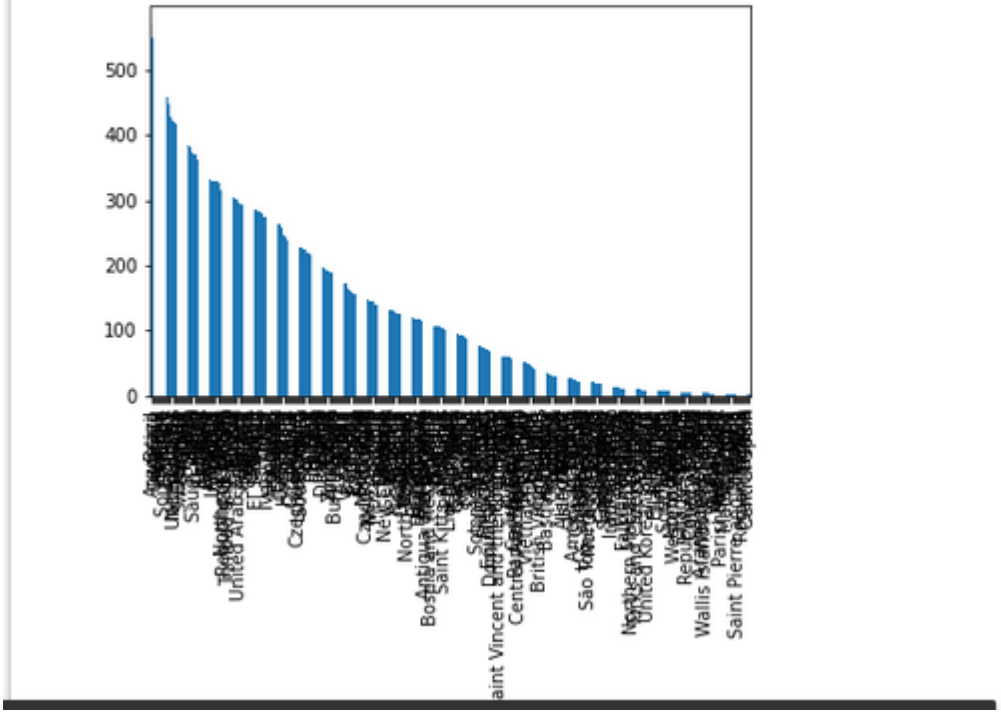
هر تیم چه مقدار بازی دوستانه به عنوان میزبان داشته است

تیم های میزبانی که بیشترین برد در کل بازی های دوستانه به عنوان میزبان داشته اند.

در نمودار زیر مشخص است.

```
df_football['home_team'].value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5432746c90>
```



نوع دیتا ست ها را مشخص کرده داده های پوچ جمع زده

```
[ ]  data.dtypes
```

```
date           object
home_team      object
away_team      object
home_score      int64
away_score      int64
tournament     object
city           object
country        object
neutral          bool
dtype: object
```

```
[ ]  data.isnull().sum()
```

```
date           0
home_team      0
away_team      0
home_score     0
away_score     0
tournament     0
city           0
country        0
neutral        0
dtype: int64
```

نتیجه گیری:

تیم هایی که میزبان بودند بیشتر برنده می شوند.