uber-drive-data-analysis

September 23, 2023

1 Uber Ride Data Analysis

This dataset contains details of uber rides of a customer. **Dataset:** The dataset contains Start Date, End Date, Start Location, End Location, Miles Driven and Purpose of drive (Business, Personal, Meals etc) dataset.

2 Objective

To fetch insights from the behavior of an common Uber customer.

3 Importing libraries

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import calendar

import os
```

4 Loading dataset

[ ]:
```
df = pd.read_csv('My Uber Drives.csv')
df.head()
```

[ ]:
```
       START_DATE*        END_DATE* CATEGORY*       START*             STOP*  \
0  1/1/2016 21:11  1/1/2016 21:17  Business  Fort Pierce      Fort Pierce
1   1/2/2016 1:25   1/2/2016 1:37  Business  Fort Pierce      Fort Pierce
2  1/2/2016 20:25  1/2/2016 20:38  Business  Fort Pierce      Fort Pierce
3  1/5/2016 17:31  1/5/2016 17:45  Business  Fort Pierce      Fort Pierce
4  1/6/2016 14:42  1/6/2016 15:49  Business  Fort Pierce  West Palm Beach

   MILES*         PURPOSE*
0     5.1   Meal/Entertain
1     5.0              NaN
2     4.8  Errand/Supplies
```

```
3     4.7          Meeting
4    63.7   Customer Visit
```

[ ]:
```
df.tail()
```

[ ]:
```
             START_DATE*          END_DATE* CATEGORY*            START*  \
1151  12/31/2016 13:24  12/31/2016 13:42  Business           Kar?chi
1152  12/31/2016 15:03  12/31/2016 15:38  Business  Unknown Location
1153  12/31/2016 21:32  12/31/2016 21:50  Business       Katunayake
1154  12/31/2016 22:08  12/31/2016 23:51  Business           Gampaha
1155            Totals               NaN       NaN               NaN

                 STOP*   MILES*        PURPOSE*
1151  Unknown Location      3.9  Temporary Site
1152  Unknown Location     16.2         Meeting
1153           Gampaha      6.4  Temporary Site
1154         Ilukwatta     48.2  Temporary Site
1155               NaN  12204.7             NaN
```

[ ]:
```
print(df.shape)
df.dtypes
```

```
(1156, 7)
```

[ ]:
```
START_DATE*     object
END_DATE*       object
CATEGORY*       object
START*          object
STOP*           object
MILES*         float64
PURPOSE*        object
dtype: object
```

There are 6 catagorical vars and 1 numeric type variable *Here STATR_DATE* and END_DATE* are in object type. We need to convert them back into datetime variable*

5 Checking for null values

[ ]:
```
df.isna().sum()
```

[ ]:
```
START_DATE*     0
END_DATE*       1
CATEGORY*       1
START*          1
STOP*           1
MILES*          0
```

```
PURPOSE*       503
dtype: int64
```

[ ]:
```
df[df['END_DATE*'].isna()]
```

[ ]:
```
     START_DATE* END_DATE* CATEGORY* START* STOP*   MILES* PURPOSE*
1155      Totals       NaN       NaN    NaN   NaN  12204.7      NaN
```

As we can see this row contains wrong data for most of the columns. We will delete it

[ ]:
```
# dropping row containing null vals
df.drop(df[df['END_DATE*'].isna()].index,axis=0,inplace=True)
```

[ ]:
```
df.isna().sum()
```

[ ]:
```
START_DATE*      0
END_DATE*        0
CATEGORY*        0
START*           0
STOP*            0
MILES*           0
PURPOSE*       502
dtype: int64
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1155 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   START_DATE*  1155 non-null   object
 1   END_DATE*    1155 non-null   object
 2   CATEGORY*    1155 non-null   object
 3   START*       1155 non-null   object
 4   STOP*        1155 non-null   object
 5   MILES*       1155 non-null   float64
 6   PURPOSE*     653 non-null    object
dtypes: float64(1), object(6)
memory usage: 72.2+ KB
```

Now we have null data only in Purpose column. As we have more than 55% data missing. So I am dropping this columns and excluding this from this analysis. You may also delete the null value rows and include this column in the analysis. `sns.countplot(df['PURPOSE*'], order=df['PURPOSE*'].value_counts().index)`

[ ]:
```
# droppig Purpose
df.drop(['PURPOSE*'],axis=1,inplace=True)
```

```
df.head(2)
```

```
[ ]:      START_DATE*        END_DATE* CATEGORY*       START*        STOP*  MILES*
   0  1/1/2016 21:11  1/1/2016 21:17  Business  Fort Pierce  Fort Pierce     5.1
   1   1/2/2016 1:25   1/2/2016 1:37  Business  Fort Pierce  Fort Pierce     5.0
```

6 Checking for duplicate rows

```
[ ]: df[df.duplicated()]
```

```
[ ]:         START_DATE*         END_DATE* CATEGORY*  START* STOP*  MILES*
   492  6/28/2016 23:34  6/28/2016 23:59  Business  Durham  Cary     9.9
```

We will remove this duplicate row

```
[ ]: df.drop(df[df.duplicated()].index, axis=0, inplace=True)
     df[df.duplicated()]
```

```
[ ]: Empty DataFrame
     Columns: [START_DATE*, END_DATE*, CATEGORY*, START*, STOP*, MILES*]
     Index: []
```

Converting start_date & end_date cols into datetime

```
[ ]: df['START_DATE*'] = pd.to_datetime(df['START_DATE*'], format='%m/%d/%Y %H:%M')
     df['END_DATE*'] = pd.to_datetime(df['END_DATE*'], format='%m/%d/%Y %H:%M')
     df.dtypes
```

```
[ ]: START_DATE*    datetime64[ns]
     END_DATE*      datetime64[ns]
     CATEGORY*              object
     START*                 object
     STOP*                  object
     MILES*                float64
     dtype: object
```

7 EDA

8 Univariate

8.1 1. Category

```
[ ]: df['CATEGORY*'].unique()
```

```
[ ]: array(['Business', 'Personal'], dtype=object)
```

There are 2 ride-categories… Business: For work related & Personal: For personal travel

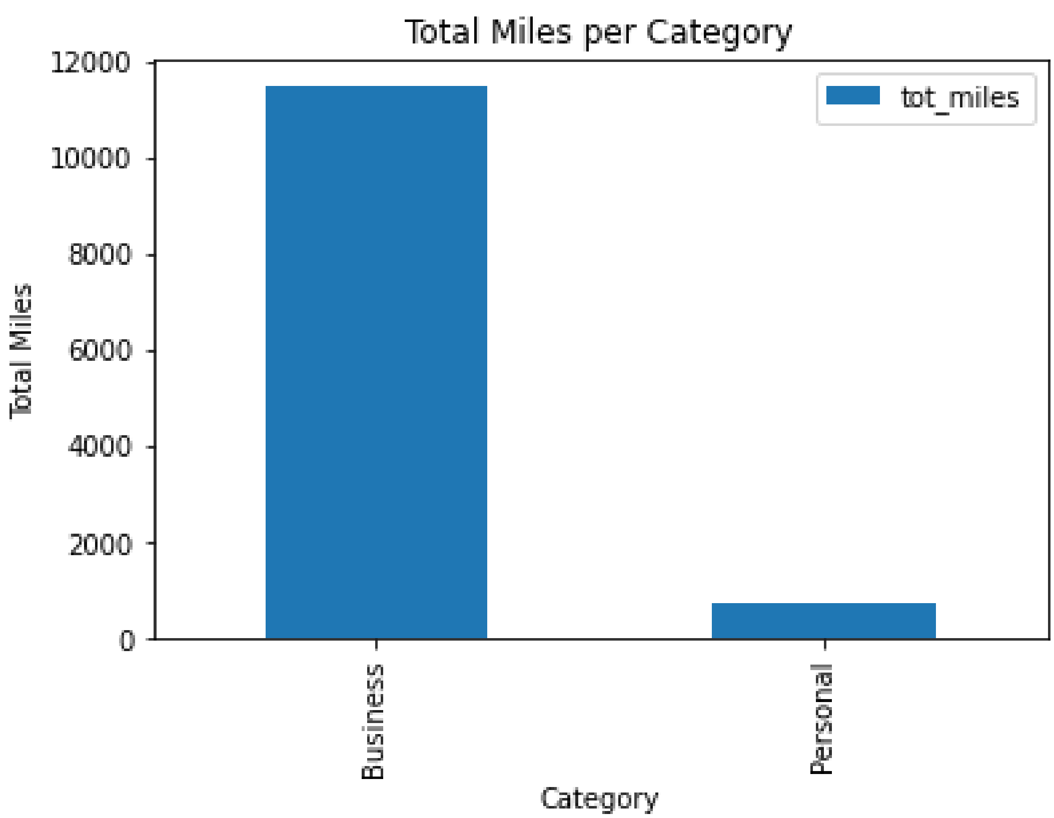
[ ]:
```
df[['CATEGORY*','MILES*']].groupby(['CATEGORY*']).
 ↪agg(tot_miles=('MILES*','sum'))
```

[ ]:
```
           tot_miles
CATEGORY*
Business     11477.1
Personal       717.7
```

[ ]:
```
plt.figure()
df[['CATEGORY*','MILES*']].groupby(['CATEGORY*']).
 ↪agg(tot_miles=('MILES*','sum')).plot(kind='bar')
plt.xlabel('Category')
plt.ylabel('Total Miles')
plt.title('Total Miles per Category')
```

[ ]:
```
Text(0.5, 1.0, 'Total Miles per Category')
```

```
<Figure size 432x288 with 0 Axes>
```



User mainly uses Uber cabs for its Business purposes * Around 94% miles was consumed during Business trips. * Only 6% miles were consumed during personal trips.

8.2 START*

[ ]:
```
len(df['START*'].unique())
```

[ ]:
```
177
```

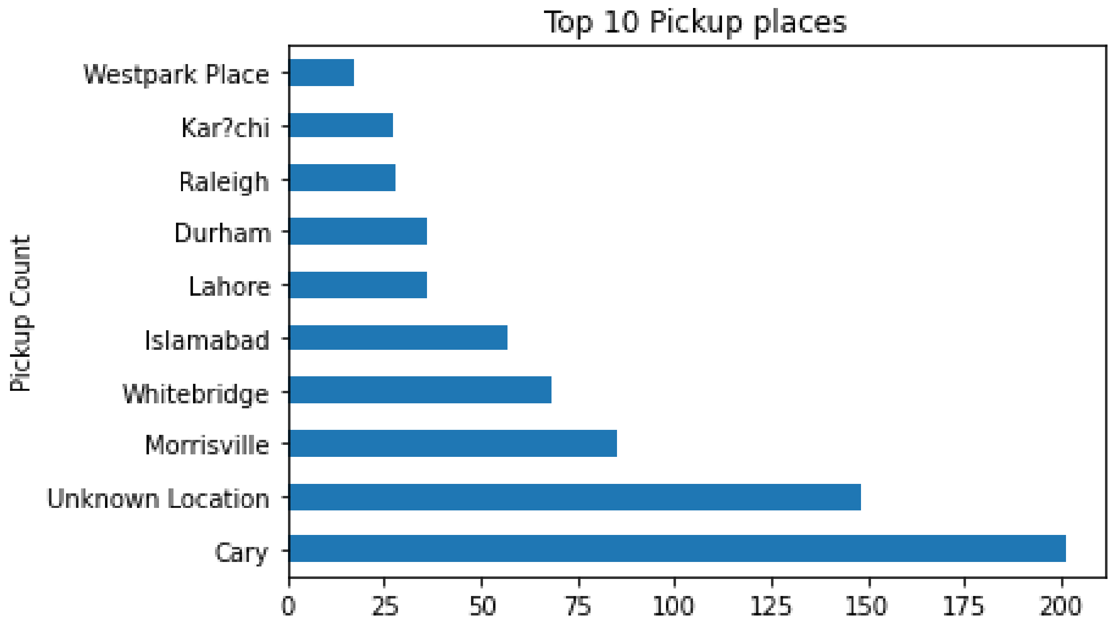
There are 177 unique starting points

[ ]:
```
# Top 10 Start places
df['START*'].value_counts(ascending=False)[:10]
```

[ ]:
```
Cary                201
Unknown Location    148
Morrisville          85
Whitebridge          68
Islamabad            57
Lahore               36
Durham               36
Raleigh              28
Kar?chi              27
Westpark Place       17
Name: START*, dtype: int64
```

[ ]:
```
df['START*'].value_counts(ascending=False)[:10].
 ↪plot(kind='barh',ylabel='Places',xlabel='Pickup Count',title='Top 10 Pickup␣
 ↪places')
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f23e44c4690>
```



Cary is the most popular Starting point for this user

8.3 STOP*

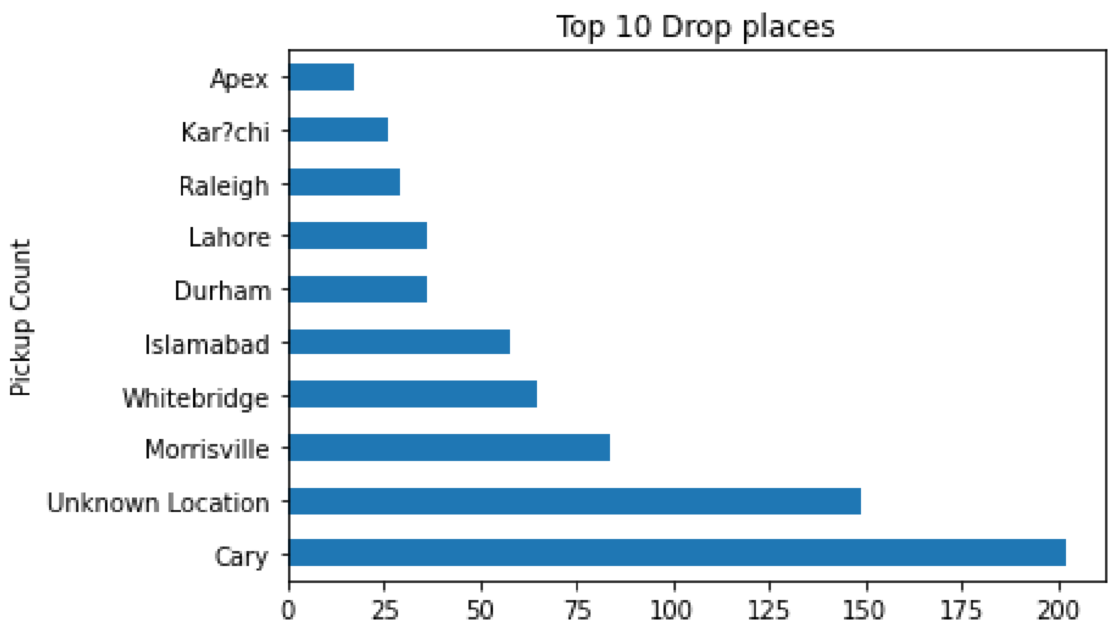
```
[ ]: len(df['STOP*'].unique())
```

```
[ ]: 188
```

There are 188 unique Drop points (destination)

```
[ ]: df['STOP*'].value_counts(ascending=False)[:10].
     ↪plot(kind='barh',ylabel='Places',xlabel='Pickup Count',title='Top 10 Drop␣
     ↪places')
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f23e444a210>
```



Cary is the most popular Stop place for this user. ***Maybe his home is in Cary (as mostly start & stop are from here)***

[ ]:
```
df[df['START*']=='Unknown Location']['START*'].value_counts()
```

[ ]:
```
Unknown Location    148
Name: START*, dtype: int64
```

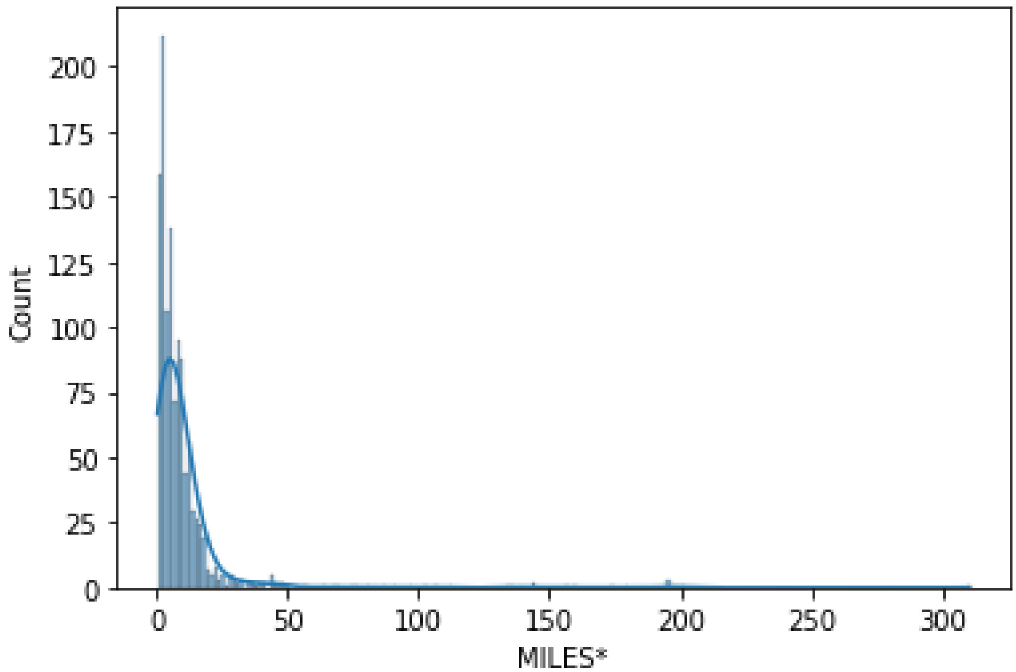
[ ]:
```
df[df['STOP*']=='Unknown Location']['STOP*'].value_counts()
```

[ ]:
```
Unknown Location    149
Name: STOP*, dtype: int64
```

8.4 MILES*

[ ]:
```
sns.histplot(df['MILES*'],kde=True)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f23e438ed90>
```



Miles data is Rightly Skewed

[ ]:
```
df.describe().T
```

[ ]:
```
         count       mean        std  min  25%  50%   75%    max
MILES*  1154.0  10.567418  21.588452  0.5  2.9  6.0  10.4  310.3
```

8.5 Multivariate analysis

[ ]:
```
df.head()
```

[ ]:
```
          START_DATE*           END_DATE* CATEGORY*       START*  \
0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES*
0      Fort Pierce     5.1
1      Fort Pierce     5.0
2      Fort Pierce     4.8
3      Fort Pierce     4.7
4  West Palm Beach    63.7
```

[ ]:
```
df.groupby(['START*','STOP*'])['MILES*'].apply(print)
```

```
910    2.2
Name: (Agnew, Agnew), dtype: float64
906    4.3
Name: (Agnew, Cory), dtype: float64
908    2.2
911    2.4
Name: (Agnew, Renaissance), dtype: float64
879    15.2
Name: (Almond, Bryson City), dtype: float64
646    1.0
825    3.3
Name: (Apex, Apex), dtype: float64
58      5.5
60      5.7
80      5.7
173     5.6
410     7.2
565     5.5
616     4.6
647     6.0
826     5.3
978     5.4
1014    5.3
1033    8.8
1045    4.7
Name: (Apex, Cary), dtype: float64
855    2.2
Name: (Apex, Eagle Rock), dtype: float64
1001    9.0
Name: (Apex, Holly Springs), dtype: float64
469    17.0
Name: (Arabi, Metairie), dtype: float64
621    4.9
Name: (Arlington, Washington), dtype: float64
613    1.3
Name: (Arlington Park at Amberly, Lexington Park at Amberly), dtype: float64
881    195.9
Name: (Asheville, Mebane), dtype: float64
871    91.8
Name: (Asheville, Topton), dtype: float64
232    136.0
Name: (Austin, Katy), dtype: float64
547     8.3
548     3.2
551     4.5
```

```
556    13.0
557     4.4
Name: (Banner Elk, Banner Elk), dtype: float64
558    15.1
Name: (Banner Elk, Boone), dtype: float64
549    22.4
Name: (Banner Elk, Elk Park), dtype: float64
552    28.1
Name: (Banner Elk, Newland), dtype: float64
357    12.9
Name: (Bellevue, Seattle), dtype: float64
967    2.3
Name: (Berkeley, El Cerrito), dtype: float64
832    3.0
839    4.6
920    3.6
948    1.3
962    2.5
Name: (Berkeley, Emeryville), dtype: float64
936    45.9
Name: (Berkeley, Menlo Park), dtype: float64
960    44.6
Name: (Berkeley, Mountain View), dtype: float64
437     5.1
964     5.1
969    16.3
Name: (Berkeley, Oakland), dtype: float64
916    11.8
918    12.2
922    11.4
930    12.6
Name: (Berkeley, San Francisco), dtype: float64
841    47.7
Name: (Berkeley, San Jose), dtype: float64
555    23.8
Name: (Boone, Banner Elk), dtype: float64
559    180.2
Name: (Boone, Cary), dtype: float64
256    9.6
Name: (Briar Meadow, Midtown), dtype: float64
878    6.6
Name: (Bryson City, Almond), dtype: float64
880    68.4
Name: (Bryson City, Asheville), dtype: float64
875    16.3
876     6.5
877     6.3
Name: (Bryson City, Bryson City), dtype: float64
```

```
333    4.8
Name: (Burtrose, Whitebridge), dtype: float64
466    4.5
Name: (CBD, Bywater), dtype: float64
444    1.9
Name: (CBD, Lower Garden District), dtype: float64
527    7.7
Name: (CBD, Pontchartrain Beach), dtype: float64
534    1.3
Name: (CBD, St Thomas), dtype: float64
358    14.4
Name: (Capitol One, Mcvan), dtype: float64
353    4.5
Name: (Capitol One, University District), dtype: float64
57       5.8
59       5.7
79       5.6
172      3.8
409      5.7
564      7.2
615      6.9
824      5.6
854     11.2
977      5.4
1000     5.1
1013     5.1
1032     5.1
1044     4.4
Name: (Cary, Apex), dtype: float64
7        0.8
30       4.8
37       1.6
38       2.4
39       1.0
43       1.4
44       0.5
45       1.8
53       4.6
54       5.2
65       6.0
66       1.6
69       1.6
70       1.1
71       1.6
73       7.7
975      1.5
976      1.8
979     39.2
```

```
980      6.4
981      2.7
982     18.5
983      2.5
984      2.1
988      5.5
989      4.1
990     12.7
993      5.9
994      1.9
995      3.3
996      1.3
999      1.4
1003     2.5
1009     5.5
1010     5.5
1022     4.1
1023     3.8
1024     6.6
1025     4.0
1026     7.0
1027     6.9
1028     3.4
1029     3.4
1030     2.0
1031     2.0
1034     5.6
1035    18.9
1042     2.1
1043     3.1
1048     4.2
1049     4.1
1050     3.4
1051     3.3
Name: (Cary, Cary), dtype: float64
61    19.4
Name: (Cary, Chapel Hill), dtype: float64
28     10.4
55     10.4
74     10.4
95      8.5
164    10.6
183     9.9
185    10.4
304    10.5
325    10.4
337     9.9
339    14.2
```

```
394    10.4
405     9.9
411    10.4
413    10.4
424    10.4
429    10.4
490    10.4
497    10.4
499     9.9
503    10.5
505    10.1
509     9.9
513    11.8
517     9.9
520     9.9
584     8.0
593    10.4
609    14.0
644    12.9
816    10.5
851    16.4
867    10.4
887    16.5
973    10.3
997    10.3
Name: (Cary, Durham), dtype: float64
1036    15.6
Name: (Cary, Fuquay-Varina), dtype: float64
314    15.3
Name: (Cary, Holly Springs), dtype: float64
268    144.0
Name: (Cary, Latta), dtype: float64
8        8.3
67       5.2
81       6.1
89       6.1
99       8.4
        …
1038     3.0
1040     3.0
1046     3.0
1052     3.0
1054    10.6
Name: (Cary, Morrisville), Length: 67, dtype: float64
34      17.1
46      18.7
50      19.0
76      11.4
```

```
91      17.3
181      7.6
187     15.7
200     12.4
307     19.1
310      8.9
316     11.9
323     13.6
334     12.4
389     14.9
455      6.0
457     19.3
461      8.6
501     13.3
642     14.9
822     20.6
845     17.2
847     28.1
1006     8.5
Name: (Cary, Raleigh), dtype: float64
985    6.7
Name: (Cary, Unknown Location), dtype: float64
1017    6.6
Name: (Cary, Wake Co.), dtype: float64
651    31.7
Name: (Cary, Wake Forest), dtype: float64
869    107.0
Name: (Cary, Winston Salem), dtype: float64
929    1.1
950    2.3
951    2.6
Name: (Central, Central), dtype: float64
940    2.9
Name: (Central, College Avenue), dtype: float64
942    2.3
Name: (Central, South), dtype: float64
952    1.9
958    1.9
Name: (Central, Southside), dtype: float64
945    0.6
Name: (Central, West Berkeley), dtype: float64
468    1.1
Name: (Chalmette, Arabi), dtype: float64
62    23.3
Name: (Chapel Hill, Cary), dtype: float64
359    17.0
Name: (Chapel Hill, Morrisville), dtype: float64
330    1.9
```

```
Name: (Chessington, Chessington), dtype: float64
573    4.8
Name: (Chessington, Whitebridge), dtype: float64
941    2.6
Name: (College Avenue, Central), dtype: float64
110    2.6
111    4.5
112    1.7
113    1.8
114    6.0
118    1.7
Name: (Colombo, Colombo), dtype: float64
119    21.4
Name: (Colombo, Katunayaka), dtype: float64
115    1.1
Name: (Colombo, Nugegoda), dtype: float64
629    1.5
Name: (Columbia Heights, Kalorama Triangle), dtype: float64
212    0.8
Name: (Congress Ave District, Downtown), dtype: float64
627    1.3
Name: (Connecticut Avenue, Kalorama Triangle), dtype: float64
229    2.0
Name: (Convention Center District, West University), dtype: float64
907    3.9
Name: (Cory, Agnew), dtype: float64
271    6.7
Name: (Couples Glen, Isles of Buena Vista), dtype: float64
280    27.2
Name: (Couples Glen, Vista East), dtype: float64
479    2.5
480    8.6
481    5.2
482    7.6
483    1.8
Name: (Covington, Covington), dtype: float64
484    4.7
Name: (Covington, Mandeville), dtype: float64
223    12.5
Name: (Coxville, The Drag), dtype: float64
296    80.5
Name: (Daytona Beach, Jacksonville), dtype: float64
442    9.3
Name: (Downtown, Bay Farm Island), dtype: float64
944    1.4
Name: (Downtown, Central), dtype: float64
23    11.2
Name: (Downtown, Gulfton), dtype: float64
```

```
258    0.9
Name: (Downtown, Midtown), dtype: float64
213    1.2
Name: (Downtown, Red River District), dtype: float64
924    6.2
Name: (Downtown, Sunnyside), dtype: float64
227    2.8
231    1.7
Name: (Downtown, The Drag), dtype: float64
914    1.8
Name: (Downtown, West Berkeley), dtype: float64
645    15.3
Name: (Durham, Apex), dtype: float64
29     10.4
56     10.1
75     10.4
165     9.9
184     9.9
186    10.9
305     8.7
326    10.0
338    10.0
340    18.2
395     9.9
406     9.9
412     9.9
414     9.9
425     9.9
430     9.9
491     9.9
498     9.9
500     9.9
504     9.9
506     9.9
510     9.9
515     9.9
521     9.9
585     9.9
594     9.9
610    13.3
868     9.9
888    12.8
974    10.5
998    11.1
Name: (Durham, Cary), dtype: float64
96      2.6
518     8.6
817     8.1
```

```
852    15.4
Name: (Durham, Morrisville), dtype: float64
26    3.9
Name: (Eagan Park, Jamestown Court), dtype: float64
856    3.6
Name: (Eagle Rock, Cary), dtype: float64
208    12.8
Name: (East Austin, West University), dtype: float64
102    2.7
Name: (East Elmhurst, Jackson Heights), dtype: float64
104    8.1
Name: (East Elmhurst, New York), dtype: float64
13    6.4
Name: (East Harlem, NoMad), dtype: float64
92    5.7
Name: (Eastgate, Walnut Terrace), dtype: float64
578    2.3
Name: (Edgehill Farms, Burtrose), dtype: float64
575    1.4
Name: (Edgehill Farms, Preston), dtype: float64
78     3.2
84     2.7
633    2.7
638    2.7
650    2.8
828    3.3
Name: (Edgehill Farms, Whitebridge), dtype: float64
968    3.1
Name: (El Cerrito, Berkeley), dtype: float64
550    12.2
Name: (Elk Park, Banner Elk), dtype: float64
11    7.5
Name: (Elmhurst, New York), dtype: float64
436    3.9
831    3.0
840    3.1
921    3.0
949    3.7
963    3.7
Name: (Emeryville, Berkeley), dtype: float64
441     5.1
833     3.8
843    13.2
Name: (Emeryville, Oakland), dtype: float64
439    9.8
Name: (Emeryville, San Francisco), dtype: float64
47    3.4
Name: (Fairmont, Meredith Townes), dtype: float64
```

```
637    4.0
Name: (Farmington Woods, Edgehill Farms), dtype: float64
31    4.7
Name: (Farmington Woods, Whitebridge), dtype: float64
189    0.8
Name: (Fayetteville Street, Depot Historic District), dtype: float64
201    5.9
Name: (Fayetteville Street, Meredith Townes), dtype: float64
35    15.1
Name: (Fayetteville Street, Umstead), dtype: float64
422    4.8
Name: (Financial District, Kips Bay), dtype: float64
14    1.6
Name: (Flatiron District, Midtown), dtype: float64
299    159.3
Name: (Florence, Cary), dtype: float64
0    5.1
1    5.0
2    4.8
3    4.7
Name: (Fort Pierce, Fort Pierce), dtype: float64
4    63.7
Name: (Fort Pierce, West Palm Beach), dtype: float64
1037    15.6
Name: (Fuquay-Varina, Cary), dtype: float64
251    57.0
Name: (Galveston, Houston), dtype: float64
247    3.1
Name: (Galveston, Port Bolivar), dtype: float64
1154    48.2
Name: (Gampaha, Ilukwatta), dtype: float64
220    6.2
Name: (Georgian Acres, The Drag), dtype: float64
24    11.8
Name: (Gulfton, Downtown), dtype: float64
873    75.7
Name: (Hayesville, Topton), dtype: float64
583    9.1
Name: (Hazelwood, Lexington Park at Amberly), dtype: float64
567    0.9
Name: (Hazelwood, Weston), dtype: float64
41     2.3
168    2.0
175    6.6
408    2.4
899    2.4
Name: (Hazelwood, Whitebridge), dtype: float64
21    2.0
```

```
Name: (Hell's Kitchen, Midtown), dtype: float64
649    4.4
Name: (Heritage Pines, Edgehill Farms), dtype: float64
159    3.2
Name: (Heritage Pines, Whitebridge), dtype: float64
315     13.7
1002    13.3
Name: (Holly Springs, Cary), dtype: float64
246    36.5
Name: (Houston, Galveston), dtype: float64
25     21.9
235     4.9
236    12.6
Name: (Houston, Houston), dtype: float64
244    12.0
Name: (Houston, Sugar Land), dtype: float64
20    2.4
Name: (Hudson Square, Hell's Kitchen), dtype: float64
18    4.0
Name: (Hudson Square, Lower Manhattan), dtype: float64
864    12.1
Name: (Huntington Woods, Huntington Woods), dtype: float64
399    1.7
865    3.9
Name: (Huntington Woods, Weston), dtype: float64
926    5.5
Name: (Ingleside, Potrero Flats), dtype: float64
129     1.5
130     1.0
133     4.2
137     3.0
138     1.5
146     4.6
664     1.2
681     3.2
686     1.4
691     1.5
730     5.3
733     4.3
734     2.5
739     2.6
742     4.4
779    37.7
782     2.8
783     1.6
801     1.7
806     1.0
807     0.7
```

```
808      2.3
1076     1.8
1077     1.4
1085     2.1
Name: (Islamabad, Islamabad), dtype: float64
151    8.1
Name: (Islamabad, Noorpur Shahan), dtype: float64
139    18.4
678     6.5
Name: (Islamabad, R?walpindi), dtype: float64
1061    5.9
Name: (Islamabad, Rawalpindi), dtype: float64
123      6.0
125     10.0
131      7.3
134     13.6
147      8.8
665      5.7
671      7.3
682     12.5
687     20.2
689      6.3
692     10.9
702      5.0
731     12.1
735      5.7
737      4.0
743      5.3
746     10.6
765      5.7
768     16.5
777     20.5
780     16.7
784     12.7
802      9.5
809     10.9
1069     2.2
1078    10.3
1081     3.5
1086     7.2
Name: (Islamabad, Unknown Location), dtype: float64
103    1.8
Name: (Jackson Heights, East Elmhurst), dtype: float64
270    201.0
Name: (Jacksonville, Kissimmee), dtype: float64
297    174.2
Name: (Jacksonville, Ridgeland), dtype: float64
9      16.5
```

```
416    22.3
Name: (Jamaica, New York), dtype: float64
488    1.0
Name: (Jamestown Court, Jamestown Court), dtype: float64
623    1.0
625    1.1
Name: (K Street, Kalorama Triangle), dtype: float64
628    1.8
Name: (Kalorama Triangle, Columbia Heights), dtype: float64
626    1.5
Name: (Kalorama Triangle, Downtown), dtype: float64
622    1.0
Name: (Kalorama Triangle, K Street), dtype: float64
1119    4.9
1120    5.0
1121    0.6
1124    5.5
1127    2.0
1130    3.8
1131    5.1
1132    3.8
1135    1.4
1136    1.1
1137    4.1
1138    6.1
1139    1.3
1142    7.2
1145    2.8
1146    2.9
1147    4.6
1148    4.6
1149    0.8
1150    0.7
Name: (Kar?chi, Kar?chi), dtype: float64
1122     3.1
1125    10.3
1128     8.5
1133    11.6
1140     3.0
1143     6.4
1151     3.9
Name: (Kar?chi, Unknown Location), dtype: float64
770    2.9
798    3.6
Name: (Karachi, Karachi), dtype: float64
771    8.2
799    8.0
Name: (Karachi, Unknown Location), dtype: float64
```

```
120    0.5
Name: (Katunayaka, Katunayaka), dtype: float64
108    43.7
Name: (Katunayaka, Unknown Location), dtype: float64
1153    6.4
Name: (Katunayake, Gampaha), dtype: float64
233    30.2
Name: (Katy, Houston), dtype: float64
476    2.2
539    1.4
Name: (Kenner, Kenner), dtype: float64
532    4.9
Name: (Kenner, Metairie), dtype: float64
443    12.4
450    15.0
452    13.6
465    12.8
477    13.0
526    12.8
530    13.2
537    13.4
Name: (Kenner, New Orleans), dtype: float64
385    1.7
Name: (Kilarney Woods, Kildaire Farms), dtype: float64
386    4.7
Name: (Kilarney Woods, Whitebridge), dtype: float64
295    77.3
Name: (Kissimmee, Daytona Beach), dtype: float64
289    11.0
292     0.7
293     5.5
294     5.1
Name: (Kissimmee, Kissimmee), dtype: float64
272     8.8
276     6.1
279     3.6
282    13.6
287    16.1
290    15.5
Name: (Kissimmee, Orlando), dtype: float64
972    6.1
Name: (Krendle Woods, Whitebridge), dtype: float64
716     7.4
717     1.5
720     3.4
721     3.8
724     7.0
725     0.9
```

```
774     9.8
792     2.6
795     2.4
796     3.1
797     6.1
1096    2.1
1097    2.1
1098    3.0
1105    5.3
1106    1.6
1107    3.6
1108    1.7
1109    2.9
1110    0.6
1111    0.6
1112    2.3
1113    2.3
1114    3.2
1115    6.2
1116    7.7
1117    3.8
Name: (Lahore, Lahore), dtype: float64
712      7.3
718      7.9
722      5.9
726     86.6
775      7.3
793      5.8
1099     6.2
1103     6.3
1118     7.9
Name: (Lahore, Unknown Location), dtype: float64
273    1.2
274    2.1
Name: (Lake Reams, Lake Reams), dtype: float64
33    7.6
Name: (Lake Wellingborough, Whitebridge), dtype: float64
446    5.5
Name: (Lakeview, Storyville), dtype: float64
269    310.3
Name: (Latta, Jacksonville), dtype: float64
614    1.9
Name: (Lexington Park at Amberly, Westpark Place), dtype: float64
571    8.7
Name: (Lexington Park at Amberly, Whitebridge), dtype: float64
107    13.9
Name: (Long Island City, Jamaica), dtype: float64
445    6.4
```

```
Name: (Lower Garden District, Lakeview), dtype: float64
19    1.8
Name: (Lower Manhattan, Hudson Square), dtype: float64
485    2.8
Name: (Mandeville, Mandeville), dtype: float64
486    30.0
Name: (Mandeville, Metairie), dtype: float64
473    1.5
Name: (Marigny, Storyville), dtype: float64
352    14.5
Name: (Mcvan, Capitol One), dtype: float64
882    45.2
Name: (Mebane, Cary), dtype: float64
418    0.7
Name: (Medical Centre, Tudor City), dtype: float64
939    36.6
Name: (Menlo Park, Berkeley), dtype: float64
366    7.9
Name: (Menlo Park, Newark), dtype: float64
937    4.0
Name: (Menlo Park, Palo Alto), dtype: float64
311    7.5
Name: (Meredith, Cedar Hill), dtype: float64
317    1.4
Name: (Meredith Townes, Harden Place), dtype: float64
51     14.7
202     9.4
Name: (Meredith Townes, Leesville Hollow), dtype: float64
449    2.7
487    4.4
Name: (Metairie, Kenner), dtype: float64
471    15.5
533     8.5
Name: (Metairie, New Orleans), dtype: float64
234    15.5
Name: (Midtown, Alief), dtype: float64
257    1.0
Name: (Midtown, Downtown), dtype: float64
12    6.2
Name: (Midtown, East Harlem), dtype: float64
261    23.0
Name: (Midtown, Greater Greenspoint), dtype: float64
17    1.9
Name: (Midtown, Hudson Square), dtype: float64
238    1.1
239    1.1
Name: (Midtown, Midtown), dtype: float64
15    1.7
```

```
Name: (Midtown, Midtown East), dtype: float64
105    2.0
Name: (Midtown, Midtown West), dtype: float64
240    13.2
243     9.4
254    10.4
259     8.8
Name: (Midtown, Sharpstown), dtype: float64
252    5.9
Name: (Midtown, Washington Avenue), dtype: float64
16    1.9
Name: (Midtown East, Midtown), dtype: float64
546    195.3
Name: (Morrisville, Banner Elk), dtype: float64
27      8.0
68      9.7
82      6.1
90      6.1
94      6.1
       …
1021    3.4
1039    3.1
1041    4.8
1047    3.0
1053    3.1
Name: (Morrisville, Cary), Length: 75, dtype: float64
542     2.2
545    11.8
814     6.2
863     5.9
896     5.0
Name: (Morrisville, Morrisville), dtype: float64
97     17.0
375     7.6
604    14.7
884    10.3
Name: (Morrisville, Raleigh), dtype: float64
961    43.6
Name: (Mountain View, Berkeley), dtype: float64
923    0.9
Name: (NOMA, Downtown), dtype: float64
467    11.8
Name: (New Orleans, Chalmette), dtype: float64
478    46.9
Name: (New Orleans, Covington), dtype: float64
451    12.9
453    12.2
474    12.6
```

```
531    13.0
536    13.6
538    12.3
Name: (New Orleans, Kenner), dtype: float64
448    14.5
529    12.5
Name: (New Orleans, Metairie), dtype: float64
423    16.3
Name: (New York, Jamaica), dtype: float64
106    13.0
Name: (New York, Long Island City), dtype: float64
10    10.8
Name: (New York, Queens), dtype: float64
22    15.1
Name: (New York, Queens County), dtype: float64
365    9.3
Name: (Newark, Menlo Park), dtype: float64
367    25.6
Name: (Newark, San Francisco), dtype: float64
554    41.9
Name: (Newland, Boone), dtype: float64
553    3.8
Name: (Newland, Newland), dtype: float64
677    3.3
701    4.4
729    6.2
Name: (Noorpur Shahan, Islamabad), dtype: float64
152    2.2
699    7.5
Name: (Noorpur Shahan, Unknown Location), dtype: float64
222    7.2
Name: (North Austin, Coxville), dtype: float64
838    2.2
Name: (North Berkeley Hills, Southside), dtype: float64
562    3.3
Name: (Northwoods, Preston), dtype: float64
86    3.0
Name: (Northwoods, Tanglewood), dtype: float64
63    3.9
Name: (Northwoods, Whitebridge), dtype: float64
116    3.6
Name: (Nugegoda, Unknown Location), dtype: float64
928    2.6
934    6.0
Name: (Oakland, Berkeley), dtype: float64
435    13.2
830    13.0
Name: (Oakland, Emeryville), dtype: float64
```

```
834    9.5
965    9.7
Name: (Oakland, San Francisco), dtype: float64
438    5.2
Name: (Oakland, Unknown Location), dtype: float64
377    11.2
Name: (Old City, Hog Island), dtype: float64
376    2.9
Name: (Old City, Parkway Museums), dtype: float64
275     6.6
278     7.3
281    25.7
285    13.8
286    28.8
288    16.4
291    20.3
Name: (Orlando, Kissimmee), dtype: float64
277    6.9
Name: (Orlando, Orlando), dtype: float64
938    2.5
Name: (Palo Alto, Menlo Park), dtype: float64
363    9.8
Name: (Palo Alto, Sunnyvale), dtype: float64
347    3.1
433    2.8
598    1.7
859    8.7
861    2.1
Name: (Parkway, Whitebridge), dtype: float64
514    1.2
Name: (Parkwood, Parkwood), dtype: float64
528    7.0
Name: (Pontchartrain Beach, CBD), dtype: float64
470    1.7
475    4.8
Name: (Pontchartrain Shores, Pontchartrain Shores), dtype: float64
250    7.5
Name: (Port Bolivar, Galveston), dtype: float64
248    2.1
249    1.2
Name: (Port Bolivar, Port Bolivar), dtype: float64
372    2.7
Name: (Preston, Westpark Place), dtype: float64
88     1.5
157    1.7
265    1.6
563    4.7
576    1.4
```

```
Name: (Preston, Whitebridge), dtype: float64
670    6.4
Name: (R?walpindi, Islamabad), dtype: float64
140    23.1
696     4.1
Name: (R?walpindi, R?walpindi), dtype: float64
141     16.5
656     15.6
679      2.0
697     18.7
763     18.2
788    112.6
804     18.4
811     12.4
Name: (R?walpindi, Unknown Location), dtype: float64
36     40.2
49     12.9
52     15.7
77      9.0
98     18.0
182    17.3
190    13.5
203    11.9
308    18.6
318    15.2
324    22.5
335    32.8
390    14.0
456     5.9
458    16.6
502    11.3
643    17.4
823    17.6
846    14.0
848    28.2
885    13.1
Name: (Raleigh, Cary), dtype: float64
93      13.5
312     15.9
462      9.0
605     14.6
1007     6.7
Name: (Raleigh, Morrisville), dtype: float64
48     2.7
188    4.9
Name: (Raleigh, Raleigh), dtype: float64
1068    7.2
Name: (Rawalpindi, Islamabad), dtype: float64
```

```
1072    3.3
Name: (Rawalpindi, Rawalpindi), dtype: float64
1062      0.7
1073     19.4
1088    103.0
Name: (Rawalpindi, Unknown Location), dtype: float64
214    1.0
Name: (Red River District, Downtown), dtype: float64
356    2.9
Name: (Redmond, Bellevue), dtype: float64
909    2.8
912    2.8
Name: (Renaissance, Agnew), dtype: float64
298    144.0
Name: (Ridgeland, Florence), dtype: float64
419    2.5
Name: (Rose Hill, Soho), dtype: float64
835    1.7
Name: (SOMISSPO, French Quarter), dtype: float64
932    1.1
Name: (SOMISSPO, Tenderloin), dtype: float64
836    10.8
917    13.2
919    11.3
966    11.8
Name: (San Francisco, Berkeley), dtype: float64
440    11.6
Name: (San Francisco, Emeryville), dtype: float64
927    12.7
933     9.9
Name: (San Francisco, Oakland), dtype: float64
362    20.5
Name: (San Francisco, Palo Alto), dtype: float64
842    44.6
Name: (San Jose, Emeryville), dtype: float64
905    3.8
Name: (San Jose, Santa Clara), dtype: float64
283    6.2
Name: (Sand Lake Commons, Sky Lake), dtype: float64
913    43.9
Name: (Santa Clara, Berkeley), dtype: float64
384    1.2
Name: (Savon Height, Kilarney Woods), dtype: float64
858    4.9
Name: (Savon Height, Parkway), dtype: float64
382    3.6
886    9.6
Name: (Savon Height, Whitebridge), dtype: float64
```

```
417    3.3
Name: (Seaport, Gramercy-Flatiron), dtype: float64
355    14.2
Name: (Seattle, Redmond), dtype: float64
255    1.2
Name: (Sharpstown, Briar Meadow), dtype: float64
237    10.4
242     9.2
260    25.6
Name: (Sharpstown, Midtown), dtype: float64
241    1.0
Name: (Sharpstown, Sharpstown), dtype: float64
284    6.0
Name: (Sky Lake, Sand Lake Commons), dtype: float64
420    0.5
Name: (Soho, Tribeca), dtype: float64
943    6.4
Name: (South, Downtown), dtype: float64
947    0.8
Name: (South, Southwest Berkeley), dtype: float64
956    0.9
Name: (South Berkeley, Southside), dtype: float64
210    1.6
Name: (South Congress, Arts District), dtype: float64
219    8.4
Name: (South Congress, North Austin), dtype: float64
215    2.1
217    1.9
225    2.7
Name: (South Congress, The Drag), dtype: float64
957    2.4
959    1.9
Name: (Southside, Central), dtype: float64
955    0.9
Name: (Southside, South Berkeley), dtype: float64
953    2.1
Name: (Southside, West Berkeley), dtype: float64
535    1.8
Name: (St Thomas, CBD), dtype: float64
570    3.0
Name: (Stonewater, Lexington Park at Amberly), dtype: float64
447    1.5
Name: (Storyville, Faubourg Marigny), dtype: float64
472    1.6
Name: (Storyville, Marigny), dtype: float64
245    35.1
Name: (Sugar Land, Houston), dtype: float64
512    8.7
```

```
Name: (Summerwinds, Whitebridge), dtype: float64
925    0.7
Name: (Sunnyside, Ingleside), dtype: float64
364    17.6
Name: (Sunnyvale, Newark), dtype: float64
346    7.5
Name: (Tanglewood, Parkway), dtype: float64
87    5.1
Name: (Tanglewood, Preston), dtype: float64
161    5.8
328    6.5
Name: (Tanglewood, Whitebridge), dtype: float64
931    1.2
Name: (Tenderloin, SOMISSPO), dtype: float64
211    2.0
228    1.7
Name: (The Drag, Congress Ave District), dtype: float64
218    5.7
226    2.0
Name: (The Drag, Convention Center District), dtype: float64
221    10.5
Name: (The Drag, North Austin), dtype: float64
216    2.2
224    2.0
Name: (The Drag, South Congress), dtype: float64
874    29.8
Name: (Topton, Bryson City), dtype: float64
872    40.7
Name: (Topton, Hayesville), dtype: float64
397    3.3
Name: (Townes at Everett Crossing, Chessington), dtype: float64
421    0.9
Name: (Tribeca, Financial District), dtype: float64
354    5.0
Name: (University District, Capitol One), dtype: float64
109    14.1
117    14.7
Name: (Unknown Location, Colombo), dtype: float64
122     12.7
124      5.2
128      7.6
132      3.5
136     14.4
144     14.5
145      2.4
150     13.0
663      5.7
680      5.7
```

```
685     12.2
688      9.8
690      4.9
732     10.8
736      2.8
738      5.5
741      8.8
745     13.0
748      9.2
764     10.5
767     18.3
778     12.6
781     10.5
805      9.8
1060     7.7
1075     5.7
1080     4.9
1084     2.1
Name: (Unknown Location, Islamabad), dtype: float64
1123     7.9
1126    10.4
1129     4.4
1134    11.9
1141     4.1
1144    12.9
Name: (Unknown Location, Kar?chi), dtype: float64
711      9.2
715      3.9
719      2.9
723      9.6
791     33.2
794      8.3
1095    14.0
1102     7.1
1104    10.7
Name: (Unknown Location, Lahore), dtype: float64
986    3.5
Name: (Unknown Location, Morrisville), dtype: float64
676     7.6
698     8.7
700     7.7
728    10.1
Name: (Unknown Location, Noorpur Shahan), dtype: float64
154    20.0
669     1.4
695     7.9
750    17.2
769     9.6
```

```
787    17.9
803    17.1
810    12.7
813    17.0
Name: (Unknown Location, R?walpindi), dtype: float64
1067    10.2
1071    12.0
1087    12.0
Name: (Unknown Location, Rawalpindi), dtype: float64
121     23.5
126     18.3
127     11.2
135      2.5
142      3.2
        …
1093     3.2
1094    12.3
1100     9.6
1101     1.3
1152    16.2
Name: (Unknown Location, Unknown Location), Length: 86, dtype: float64
1018    1.8
Name: (Wake Co., Morrisville), dtype: float64
652    31.9
Name: (Wake Forest, Cary), dtype: float64
630    6.6
Name: (Washington, Arlington), dtype: float64
253    6.2
Name: (Washington Avenue, Midtown), dtype: float64
460    2.1
Name: (Waverly Place, Macgregor Downs), dtype: float64
170    6.8
205    7.6
342    6.8
Name: (Waverly Place, Whitebridge), dtype: float64
177    8.0
301    8.0
322    8.0
588    8.0
Name: (Wayne Ridge, Whitebridge), dtype: float64
915    3.3
935    0.8
Name: (West Berkeley, Central), dtype: float64
837    4.1
Name: (West Berkeley, North Berkeley Hills), dtype: float64
946    5.9
Name: (West Berkeley, South), dtype: float64
954    4.0
```

```
Name: (West Berkeley, Southside), dtype: float64
624    2.0
Name: (West End, Northwest Rectangle), dtype: float64
6    7.1
Name: (West Palm Beach, Palm Beach), dtype: float64
5    4.3
Name: (West Palm Beach, West Palm Beach), dtype: float64
230    2.1
Name: (West University, Congress Ave District), dtype: float64
209    2.3
Name: (West University, South Congress), dtype: float64
194    4.2
403    3.8
Name: (Weston, Weston), dtype: float64
167    4.2
Name: (Westpark Place, Hazelwood), dtype: float64
163    1.7
179    2.3
267    2.2
306    1.8
344    2.3
349    3.9
373    1.9
391    1.8
428    1.7
496    1.8
600    2.1
608    2.2
611    1.8
654    1.8
894    1.8
901    1.8
Name: (Westpark Place, Whitebridge), dtype: float64
612    6.2
Name: (Whitebridge, Arlington Park at Amberly), dtype: float64
332    4.9
Name: (Whitebridge, Burtrose), dtype: float64
572    3.9
Name: (Whitebridge, Chessington), dtype: float64
83     4.3
199    2.8
309    2.8
574    2.8
577    2.7
617    2.8
632    2.7
827    3.3
Name: (Whitebridge, Edgehill Farms), dtype: float64
```

```
636    5.2
Name: (Whitebridge, Farmington Woods), dtype: float64
40     2.0
174    2.6
407    3.0
898    2.5
Name: (Whitebridge, Hazelwood), dtype: float64
158    3.1
648    2.2
Name: (Whitebridge, Heritage Pines), dtype: float64
383    4.5
Name: (Whitebridge, Kildaire Farms), dtype: float64
32    7.2
Name: (Whitebridge, Lake Wellingborough), dtype: float64
72    9.0
Name: (Whitebridge, Macgregor Downs), dtype: float64
85     5.3
180    5.2
Name: (Whitebridge, Northwoods), dtype: float64
595    1.5
819    2.1
860    2.1
Name: (Whitebridge, Parkway), dtype: float64
156    1.5
264    1.7
371    2.8
Name: (Whitebridge, Preston), dtype: float64
381    3.6
401    7.8
857    3.6
Name: (Whitebridge, Savon Height), dtype: float64
569    6.4
Name: (Whitebridge, Stonewater), dtype: float64
511    8.8
Name: (Whitebridge, Summerwinds), dtype: float64
160    6.0
327    6.0
345    6.2
Name: (Whitebridge, Tanglewood), dtype: float64
169    7.7
204    7.2
341    7.7
459    7.1
641    6.9
Name: (Whitebridge, Waverly Place), dtype: float64
176    8.0
196    7.8
300    7.9
```

```
321    8.2
Name: (Whitebridge, Wayne Ridge), dtype: float64
42     1.9
162    6.3
166    1.9
171    2.1
178    2.2
266    2.0
343    2.1
348    2.2
495    1.6
599    2.2
607    2.2
653    1.9
893    1.4
900    1.4
Name: (Whitebridge, Westpark Place), dtype: float64
206    1.6
263    1.4
516    0.6
889    1.2
890    1.0
891    4.1
892    4.2
Name: (Whitebridge, Whitebridge), dtype: float64
64    8.3
Name: (Whitebridge, Williamsburg Manor), dtype: float64
870    133.6
Name: (Winston Salem, Asheville), dtype: float64
```

[ ]:
```
START*          STOP*
Agnew           Agnew                  None
                Cory                   None
                Renaissance            None
Almond          Bryson City            None
Apex            Apex                   None
                                       …
Whitebridge     Wayne Ridge            None
                Westpark Place         None
                Whitebridge            None
                Williamsburg Manor     None
Winston Salem   Asheville              None
Name: MILES*, Length: 363, dtype: object
```

[ ]:
```
df.groupby(['START*','STOP*'])['MILES*'].sum().sort_values(ascending=False)[1:
↪11]
```

```
[ ]: START*            STOP*
     Morrisville       Cary                395.7
     Cary              Durham              390.0
                       Morrisville         380.0
     Raleigh           Cary                365.7
     Cary              Raleigh             336.5
     Durham            Cary                324.5
     Latta             Jacksonville        310.3
     Islamabad         Unknown Location    267.0
     Cary              Cary                255.9
     Unknown Location  Islamabad           243.8
     Name: MILES*, dtype: float64
```

Cary-Durham & Cary-Morrisville and vice versa are the farthest distance ride.

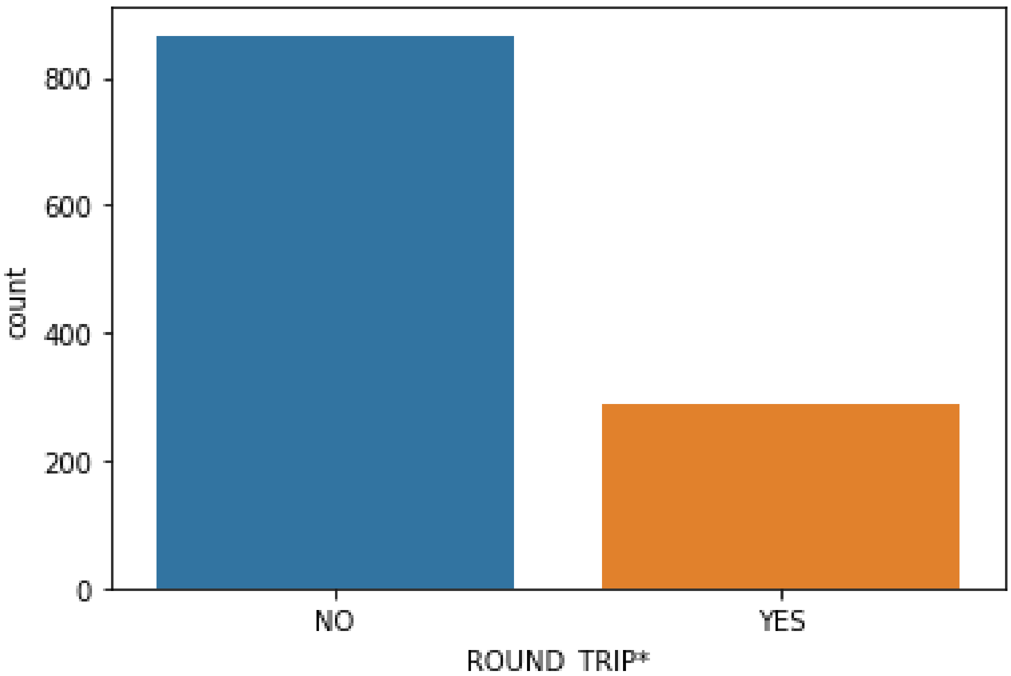
Checking for Round Trip

```
[ ]: def is_roundtrip(df):
         if df['START*'] == df['STOP*']:
             return 'YES'
         else:
             return 'NO'

     df['ROUND_TRIP*'] = df.apply(is_roundtrip, axis=1)

     sns.countplot(x='ROUND_TRIP*',data=df, order=df['ROUND_TRIP*'].value_counts().
      ↪index)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f23e0a01290>
```



[ ]:
```
df['ROUND_TRIP*'].value_counts()
```

[ ]:
```
NO     866
YES    288
Name: ROUND_TRIP*, dtype: int64
```

User mostly take single-trip Uber rides. * Around 75% trip is single-trip and 25% are ROund-Trip

8.6 Calculating Ride duration

[ ]:
```
df.dtypes
```

[ ]:
```
START_DATE*    datetime64[ns]
END_DATE*      datetime64[ns]
CATEGORY*              object
START*                 object
STOP*                  object
MILES*                float64
ROUND_TRIP*            object
dtype: object
```

[ ]:
```
df['Ride_duration'] = df['END_DATE*']-df['START_DATE*']
df.head()
```

```
[ ]:            START_DATE*           END_DATE* CATEGORY*       START*  \
 0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
 1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
 2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
 3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
 4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES* ROUND_TRIP*   Ride_duration
 0      Fort Pierce     5.1         YES 0 days 00:06:00
 1      Fort Pierce     5.0         YES 0 days 00:12:00
 2      Fort Pierce     4.8         YES 0 days 00:13:00
 3      Fort Pierce     4.7         YES 0 days 00:14:00
 4  West Palm Beach    63.7          NO 0 days 01:07:00
```

Converting Ride_duration into Minutes This is a Python lambda function that takes a single argument "x".

The function first calls the to_pytimedelta() method on pd.Timedelta, which converts the input x into a datetime.timedelta object.

The function then calculates the total number of minutes in the timedelta object, which is done by first getting the number of days using the days attribute and multiplying it by 24 hours and 60 minutes per hour. Then, the number of seconds is divided by 60 to convert them into minutes, and added to the previously calculated number of minutes. The final result is the total number of minutes in the timedelta object.

This function could be used to calculate the duration of a time interval in minutes, which could be useful in a variety of applications such as analyzing time-series data or calculating the length of time between two events.

Note that this function assumes that the input x is a valid pd.Timedelta object and may raise errors if the input is not in the expected format.

```
[ ]: # using datetime.Timedelta  => https://pandas.pydata.org/pandas-docs/stable/
      ↪user_guide/timedeltas.html
     df.loc[:, 'Ride_duration'] = df['Ride_duration'].apply(lambda x: pd.Timedelta.
      ↪to_pytimedelta(x).days/(24*60) + pd.Timedelta.to_pytimedelta(x).seconds/60)
     df.head()
```

```
[ ]:            START_DATE*           END_DATE* CATEGORY*       START*  \
 0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
 1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
 2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
 3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
 4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES* ROUND_TRIP*  Ride_duration  month  Year  Day  Hour  \
 0      Fort Pierce     5.1         YES            6.0      1  2016    1    21
 1      Fort Pierce     5.0         YES           12.0      1  2016    2     1
```

```
2      Fort Pierce     4.8          YES           13.0       1  2016     2     20
3      Fort Pierce     4.7          YES           14.0       1  2016     5     17
4  West Palm Beach    63.7           NO           67.0       1  2016     6     14

  day_of_week  Duration_hours
0         Fri 0 days 00:00:06
1         Sat 0 days 00:00:12
2         Sat 0 days 00:00:13
3         Tue 0 days 00:00:14
4         Wed 0 days 00:01:07
```

[ ]:
```python
#Capture Hour, Day, Month and Year of Ride in a separate column
df['month'] = pd.to_datetime(df['START_DATE*']).dt.month
df['Year'] = pd.to_datetime(df['START_DATE*']).dt.year
df['Day'] = pd.to_datetime(df['START_DATE*']).dt.day
df['Hour'] = pd.to_datetime(df['START_DATE*']).dt.hour

df['day_of_week'] = pd.to_datetime(df['START_DATE*']).dt.dayofweek
days = {0:'Mon',1:'Tue',2:'Wed',3:'Thur',4:'Fri',5:'Sat',6:'Sun'}

df['day_of_week'] = df['day_of_week'].apply(lambda x: days[x])

df.head()
```

[ ]:
```
          START_DATE*           END_DATE* CATEGORY*       START*  \
0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES* ROUND_TRIP*   Ride_duration  month  Year  Day  \
0      Fort Pierce     5.1          YES 0 days 00:06:00      1  2016    1
1      Fort Pierce     5.0          YES 0 days 00:12:00      1  2016    2
2      Fort Pierce     4.8          YES 0 days 00:13:00      1  2016    2
3      Fort Pierce     4.7          YES 0 days 00:14:00      1  2016    5
4  West Palm Beach    63.7           NO 0 days 01:07:00      1  2016    6

   Hour day_of_week  Duration_hours
0    21         Fri 0 days 00:00:06
1     1         Sat 0 days 00:00:12
2    20         Sat 0 days 00:00:13
3    17         Tue 0 days 00:00:14
4    14         Wed 0 days 00:01:07
```

Addding month name instead of month number

[ ]:
```
df['month'] = df['month'].apply(lambda x: calendar.month_abbr[x])
df.head()
```

[ ]:
```
           START_DATE*            END_DATE* CATEGORY*       START*  \
0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES* ROUND_TRIP*  Ride_duration month  Year  Day  Hour  \
0      Fort Pierce     5.1         YES            6.0   Jan  2016    1    21
1      Fort Pierce     5.0         YES           12.0   Jan  2016    2     1
2      Fort Pierce     4.8         YES           13.0   Jan  2016    2    20
3      Fort Pierce     4.7         YES           14.0   Jan  2016    5    17
4  West Palm Beach    63.7          NO           67.0   Jan  2016    6    14

  day_of_week
0         Fri
1         Sat
2         Sat
3         Tue
4         Wed
```

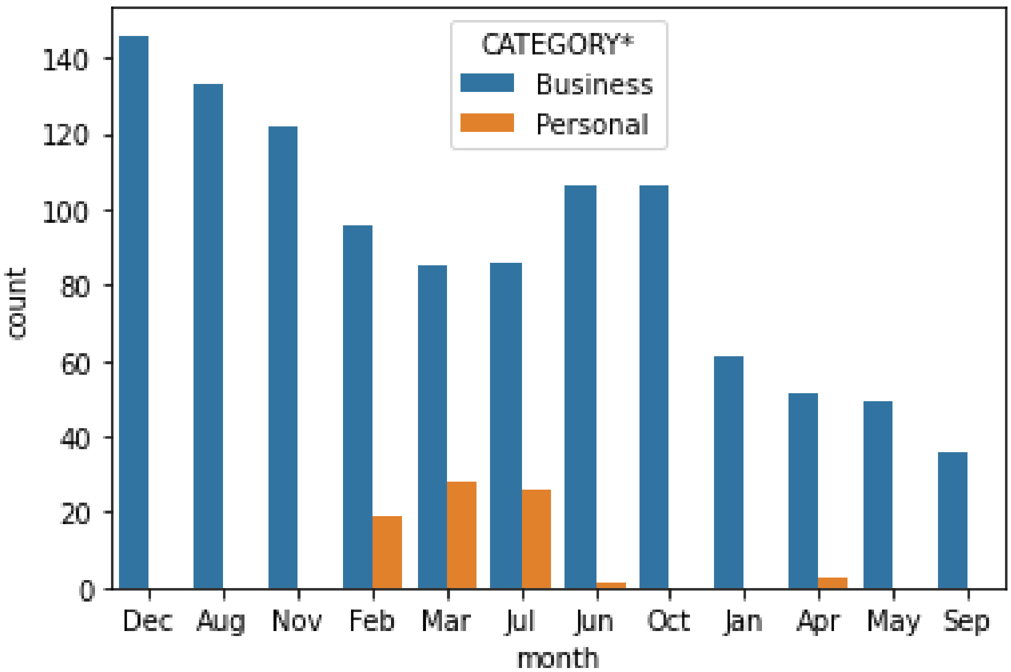
Total rides/month

[ ]:
```
print(df['month'].value_counts())
```

```
Dec    146
Aug    133
Nov    122
Feb    115
Mar    113
Jul    112
Jun    107
Oct    106
Jan     61
Apr     54
May     49
Sep     36
Name: month, dtype: int64
```

[ ]:
```
sns.countplot(x='month',data=df,order=pd.value_counts(df['month']).
  ↪index,hue='CATEGORY*')
```

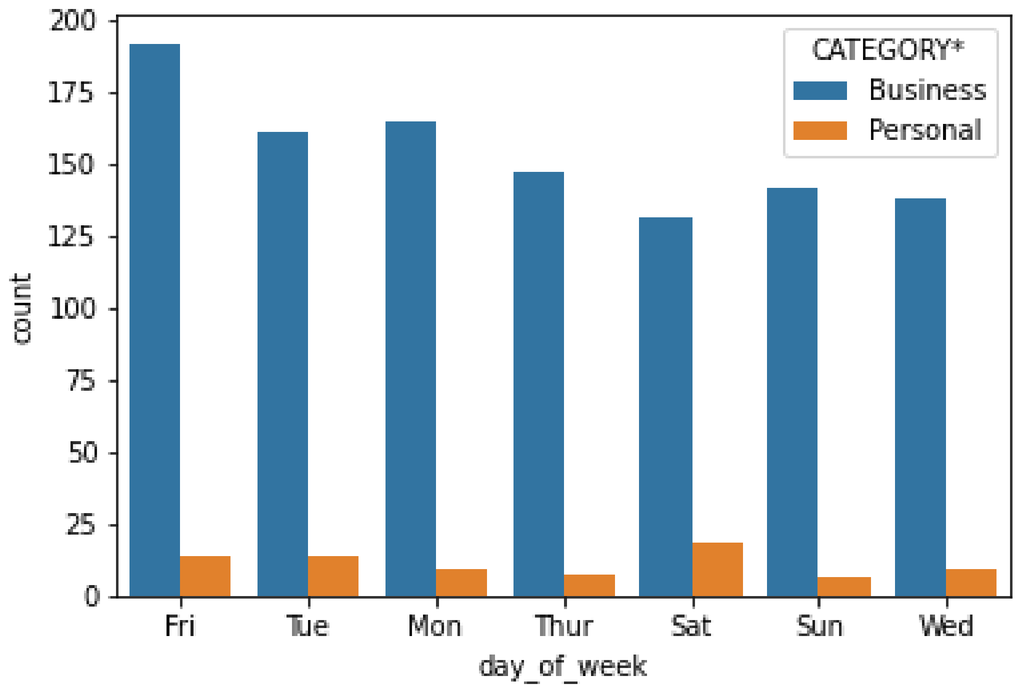
[ ]:
```
<AxesSubplot:xlabel='month', ylabel='count'>
```



Most number of rides were in month of December (all of them were Business trips) Top 5 months having most trips were: December,August,November,February & March. **Uber Ride was used at Feb,Mar,Jul,Jun & Apr for personal trips.**

[ ]:
```
sns.countplot(x='day_of_week',data=df,order=pd.value_counts(df['day_of_week']).
 ↪index,hue='CATEGORY*')
```

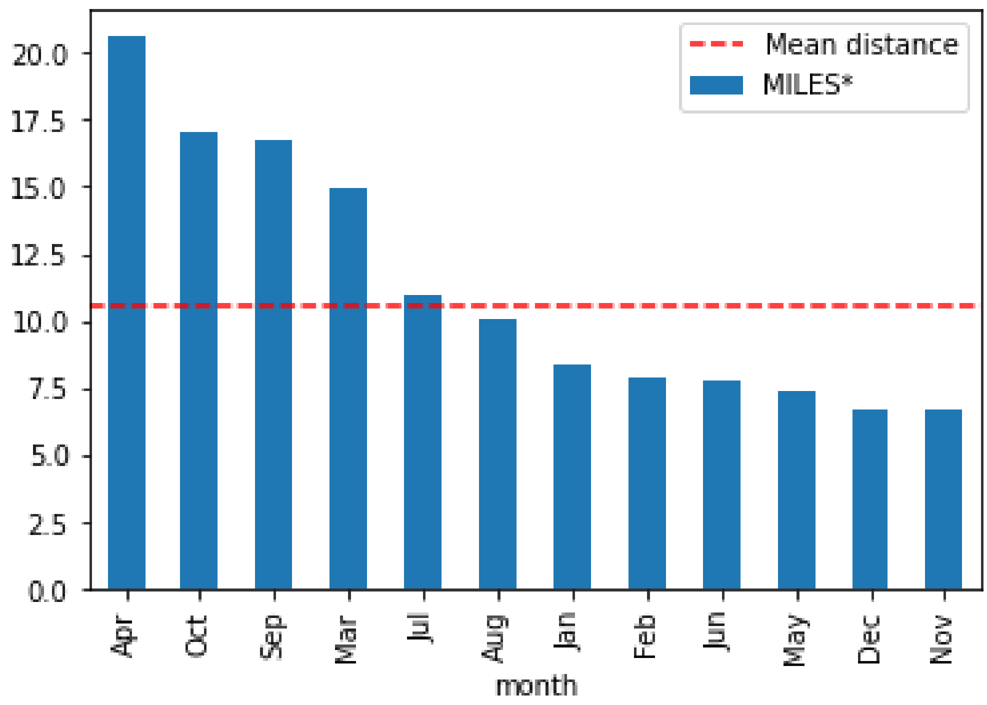
[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f23e119b710>
```



FRIDAY was the day at which uber rides were mostly used

Average distance covered/month

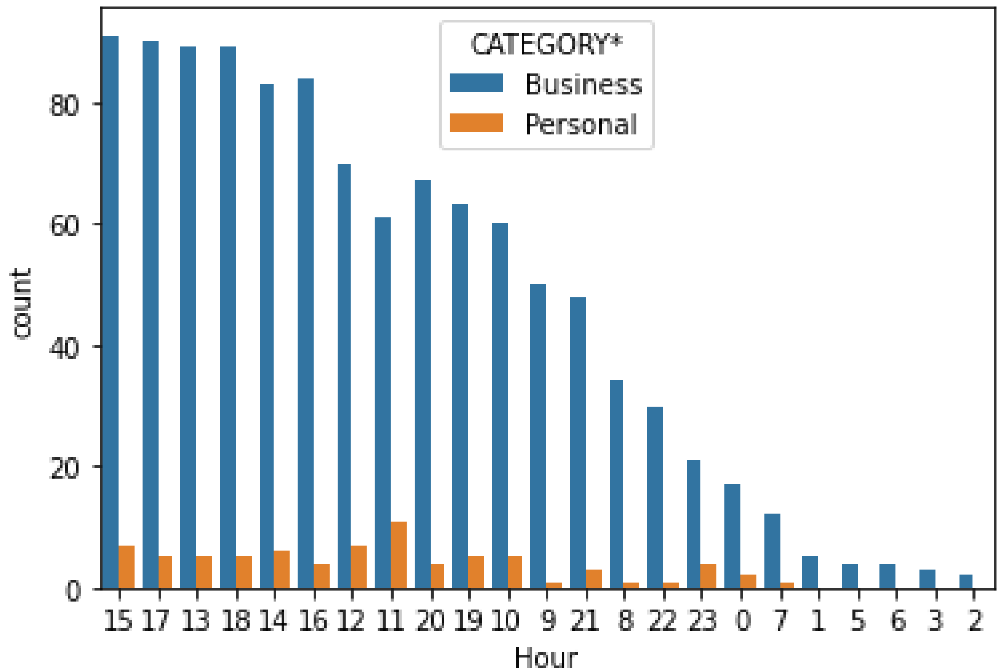
[ ]:
```
df.groupby('month').mean()['MILES*'].sort_values(ascending = False).
 ↪plot(kind='bar')
plt.axhline(df['MILES*'].mean(), linestyle='--', color='red', label='Mean␣
 ↪distance')
plt.legend()
plt.show()
```



User's Longest ride were on April & shortest were on November

[ ]:
```
sns.countplot(x='Hour',data=df,order=pd.value_counts(df['Hour']).
    ↪index,hue='CATEGORY*')
```

[ ]:
```
<AxesSubplot:xlabel='Hour', ylabel='count'>
```



Maximim number of trips were on Evening & at noon.

8.6.1 Calculating Trip speed

[ ]:
```
df.head()
```

[ ]:
```
           START_DATE*           END_DATE* CATEGORY*       START*  \
0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce
1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce
2 2016-01-02 20:25:00 2016-01-02 20:38:00  Business  Fort Pierce
3 2016-01-05 17:31:00 2016-01-05 17:45:00  Business  Fort Pierce
4 2016-01-06 14:42:00 2016-01-06 15:49:00  Business  Fort Pierce

             STOP*  MILES* ROUND_TRIP*   Ride_duration  month  Year  Day  \
0      Fort Pierce     5.1         YES 0 days 00:06:00      1  2016    1
1      Fort Pierce     5.0         YES 0 days 00:12:00      1  2016    2
2      Fort Pierce     4.8         YES 0 days 00:13:00      1  2016    2
3      Fort Pierce     4.7         YES 0 days 00:14:00      1  2016    5
4  West Palm Beach    63.7          NO 0 days 01:07:00      1  2016    6

   Hour day_of_week
0    21         Fri
1     1         Sat
2    20         Sat
3    17         Tue
```

```
4    14          Wed
```

[ ]:
```
df['Duration_hours'] = df['Ride_duration']/60
df['Speed_KM'] = df['MILES*']/df['Duration_hours']
df.head(2)
```

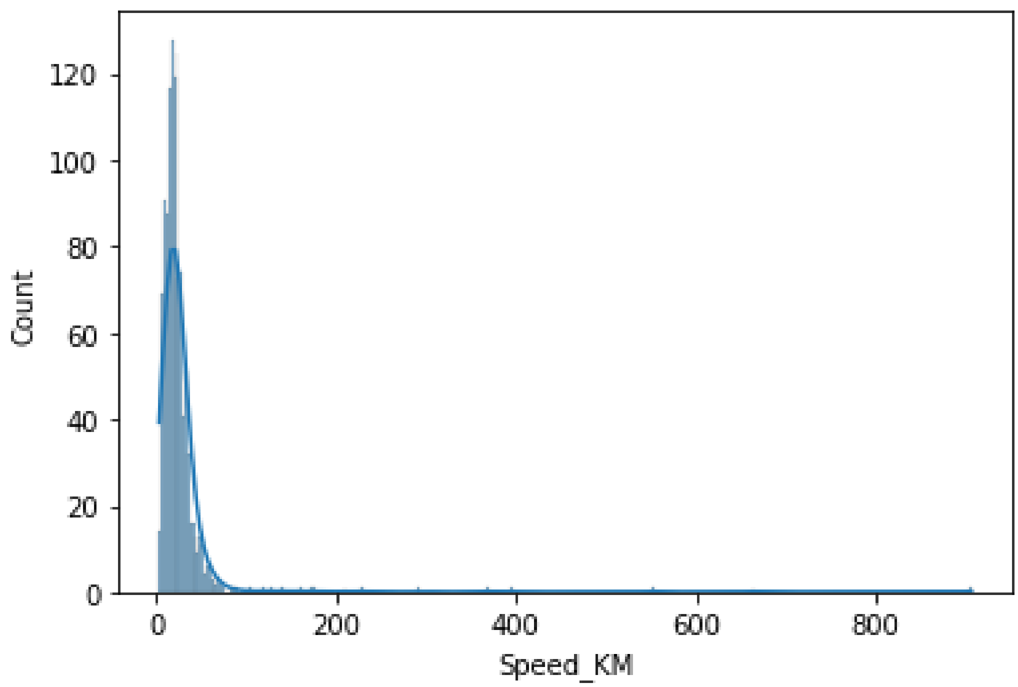
[ ]:
```
            START_DATE*           END_DATE* CATEGORY*       START*        STOP*  \
0 2016-01-01 21:11:00 2016-01-01 21:17:00  Business  Fort Pierce  Fort Pierce
1 2016-01-02 01:25:00 2016-01-02 01:37:00  Business  Fort Pierce  Fort Pierce

   MILES* ROUND_TRIP*  Ride_duration  month  Year  Day  Hour day_of_week  \
0     5.1         YES            6.0      1  2016    1    21         Fri
1     5.0         YES           12.0      1  2016    2     1         Sat

   Duration_hours  Speed_KM
0             0.1      51.0
1             0.2      25.0
```

[ ]:
```
fig, ax = plt.subplots()
sns.histplot(x='Speed_KM',data=df,kde=True,ax=ax)
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f23e082df50>



Speed is right skewed

9 Conclusion

- **User mainly uses Uber cabs for its Business purposes**
  - Around 94% miles was consumed during Business trips.
  - Only 6% miles were consumed during personal trips.
- There are 177 unique starting points
  - **Cary is most poplular starting point for this driver.**
- There are 188 unique Stop points.
  - **Cary is most poplular drop point for this driver.**
- **Cary-Durham & Cary-Morrisville and vice versa are the User's longest distance Uber ride.**
- **User usually takes single-trip Uber rides.**
  - Around 75% trip is single-trip and 25% are Round-Trip.
- **User's Most number of rides were in month of December & Least were in September.**
- **Friday has maximum number of trips.**
- **Afternoons and evenings seem to have the maximum number of trips.**
- **User's Longest ride were on April & shortest were on November**