In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('/Users/Rana/Desktop/Uber_drives.csv')
```

In [3]:
```
df.shape
```

Out[3]:
```
(1156, 7)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1156 entries, 0 to 1155
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   START_DATE*  1156 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*       1156 non-null   float64
 6   PURPOSE*     653 non-null    object 
dtypes: float64(1), object(6)
memory usage: 63.3+ KB
```

In [6]:
```
df
```

Out[6]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	1/1/16 21:11	1/1/16 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/16 1:25	1/2/16 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/16 20:25	1/2/16 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/16 17:31	1/5/16 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/16 14:42	1/6/16 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit
...	...	...	...	...	...	...	...
1151	12/31/16 13:24	12/31/16 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/16 15:03	12/31/16 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/16 21:32	12/31/16 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1154	12/31/16 22:08	12/31/16 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary Site
1155	Totals	NaN	NaN	NaN	NaN	12204.7	NaN

1156 rows × 7 columns

In [7]:
```
df.nunique()
```

Out[7]:
```
START_DATE*    1155
END_DATE*      1154
CATEGORY*         2
START*          177
STOP*           188
MILES*          257
PURPOSE*         10
dtype: int64
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
START_DATE*      0
END_DATE*        1
CATEGORY*        1
START*           1
STOP*            1
MILES*           0
PURPOSE*       503
dtype: int64
```

Start by Importing librararies & Exploring data and identify quality issues to be fixed.

In [9]:
```
df=df.drop(df[df['START_DATE*'] == 'Totals'].index)
# If we want to delete some columns we use this df=df.drop(subset=[Column names,
```

Now we need to check duplicates if exsist

In [10]:
```
df.duplicated().sum()
```

Out[10]: 1

In [11]:
```
df=df.drop_duplicates()
```

No Nulls

There is one duplicate to remove here:

Now, It is time to identify any outliers. We will replace the outliers with mean.

May different ways can be used here:

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   START_DATE*  1154 non-null   object 
 1   END_DATE*    1154 non-null   object 
 2   CATEGORY*    1154 non-null   object 
 3   START*       1154 non-null   object 
 4   STOP*        1154 non-null   object 
 5   MILES*       1154 non-null   float64
 6   PURPOSE*     652 non-null    object 
dtypes: float64(1), object(6)
memory usage: 72.1+ KB
```

Now it is time to change data type into date time fort start date and end date.

Change data types rename fields

In [13]:
```
df=df.rename(columns={'MILES*':'Miles','START_DATE*':'Start_Date', 'END_DATE*':'
df
```

Out[13]:

	Start_Date	End_Date	Category	START*	STOP*	Miles	Purpose
0	1/1/16 21:11	1/1/16 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/16 1:25	1/2/16 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/16 20:25	1/2/16 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/16 17:31	1/5/16 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/16 14:42	1/6/16 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit
...	...	...	...	...	...	...	...
1150	12/31/16 1:07	12/31/16 1:14	Business	Kar?chi	Kar?chi	0.7	Meeting
1151	12/31/16 13:24	12/31/16 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/16 15:03	12/31/16 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/16 21:32	12/31/16 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site
1154	12/31/16 22:08	12/31/16 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary Site

1154 rows × 7 columns

Change data types

In [14]:
```
df['Miles']=df['Miles'].astype(int)
df['Start_Date']=pd.to_datetime(df['Start_Date'], errors='ignore')
df['End_Date']=pd.to_datetime(df['End_Date'], errors='ignore')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
```

```
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Start_Date  1154 non-null   datetime64[ns]
 1   End_Date    1154 non-null   datetime64[ns]
 2   Category    1154 non-null   object
 3   START*      1154 non-null   object
 4   STOP*       1154 non-null   object
 5   Miles       1154 non-null   int64
 6   Purpose     652 non-null    object
dtypes: datetime64[ns](2), int64(1), object(4)
memory usage: 72.1+ KB
```

We will do add field related to date and time for further analysis:

In [267…

```
df['Month']=pd.to_datetime(df['Start_Date']).dt.month
df['Day']=pd.to_datetime(df['Start_Date']).dt.day_name()
df['Start_Time']=pd.to_datetime(df['Start_Date']).dt.time
df['End_Time']=pd.to_datetime(df['End_Date']).dt.time
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1154 entries, 0 to 1154
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Start_Date  1154 non-null   datetime64[ns]
 1   End_Date    1154 non-null   datetime64[ns]
 2   Category    1154 non-null   object
 3   START*      1154 non-null   object
 4   STOP*       1154 non-null   object
 5   Miles       1154 non-null   int64
 6   Purpose     652 non-null    object
 7   Month       1154 non-null   int64
 8   Day         1154 non-null   object
 9   Start_Time  1154 non-null   object
 10  End_Time    1154 non-null   object
dtypes: datetime64[ns](2), int64(2), object(7)
memory usage: 108.2+ KB
```

Fill Up null with 'Unknown'

In [32]:

```
df['Purpose'].fillna(value='Unknown' , inplace=True)
df['Category'].fillna(value='Unknown' , inplace=True)
```

Create filed to calculate the duration of trip to correlate it to the mile field and analyze the relationship

In [268…

```
df['Duration']=df['End_Date']-df['Start_Date']
df['Duration_Time']=df['Duration'].dt.seconds
```

In [274…

```
df[['Miles','Duration_Time']].describe()
```

Out[274…

	Miles	Duration_Time
count	1154.000000	1154.000000
mean	10.118718	1394.506066

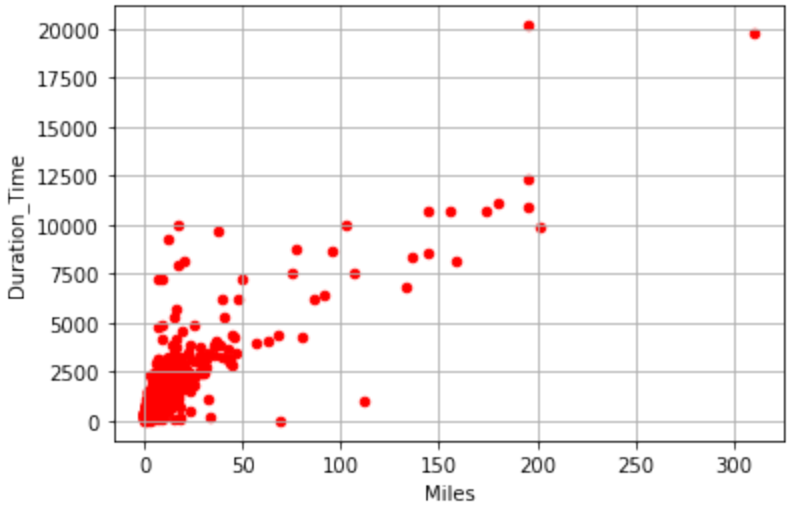
	Miles	Duration_Time
std	21.601426	1639.804308
min	0.000000	0.000000
25%	2.000000	600.000000
50%	6.000000	960.000000
75%	10.000000	1665.000000
max	310.000000	20160.000000

In [275…
```
correlation = df['Duration_Time'].corr(df['Miles'])
correlation
```

Out[275… `0.8419012072872754`

In [270…
```
df.plot.scatter(x='Miles', y='Duration_Time',color='red', grid=True)
```

Out[270… `<AxesSubplot:xlabel='Miles', ylabel='Duration_Time'>`



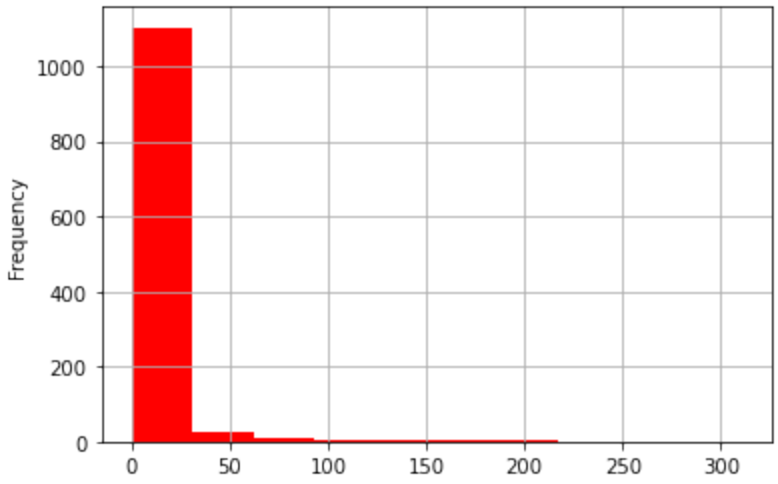
Handling and dicovering Miles distribution.

In [271…
```
df['Miles'].describe()
```

Out[271…
```
count    1154.000000
mean       10.118718
std        21.601426
min         0.000000
25%         2.000000
50%         6.000000
75%        10.000000
max       310.000000
Name: Miles, dtype: float64
```

In [278…
```
df['Miles'].plot(kind='hist',color='red', grid=True)
```

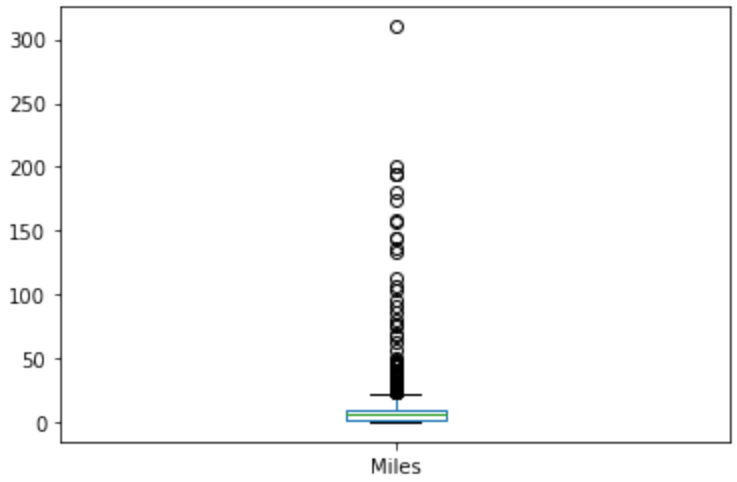
Out[278… `<AxesSubplot:ylabel='Frequency'>`



In [279…

```
df['Miles'].plot(kind='box')
```

Out[279… <AxesSubplot:>



Let us see the pecentage of outliers to the total data point.

outlier

20% of all data falls into the outlies area, Dealling with outlies to be determind case by case. in our project we will replace the outliers data points with mean.

The big difference between the mean and the max show lots of outliers that will scrow our results That being said, we will need to either replace the data with mean or avrage or just drop the outliers.

Number=[] q_low = df["MILES*"].quantile(0.01) q_hi = df["MILES*"].quantile(0.99) mean_value= df['MILES*'].mean() for x in df['MILES*']: if q_hi >x > q_low : Number.append(x) else: Number.append(mean_value) df['Miles']=Number df=df.sort_values(by='Miles', ascending=True) df

In [ ]:

```
```

Logically, we should find a relation between trips durations and miles. Let us see here:

df.corr()

Do not forget. we can save our cleaned data

sns.heatmap(df.corr(), annot=True, cmap='coolwarm') plt.show()

In [ ]:

In [249…
```
#df.to_csv('/Users/Rana/Desktop/Uber_clean.csv')
```

Here we can detect the durations of each trip. We can see the start and end time of those trips are the same, which highlighed an error in those data points.

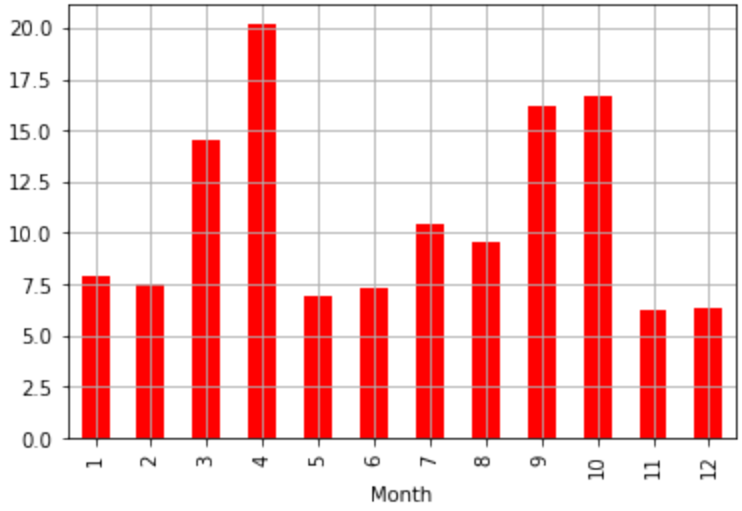
In [250…
```
df[df['Duration_Time']==0]
```

Out[250…

	Start_Date	End_Date	CATEGORY	START*	STOP*	MILES*	PURPOSE	Duration	Durati
751	2016-09-06 17:49:00	2016-09-06 17:49:00	Business	Unknown Location	Unknown Location	69	NaN	0 days	
761	2016-09-16 07:08:00	2016-09-16 07:08:00	Business	Unknown Location	Unknown Location	1	NaN	0 days	
798	2016-10-08 15:03:00	2016-10-08 15:03:00	Business	Karachi	Karachi	3	NaN	0 days	
807	2016-10-13 13:02:00	2016-10-13 13:02:00	Business	Islamabad	Islamabad	0	NaN	0 days	

In [280…
```
df['Miles'].groupby(df['Month']).mean().plot(kind='bar',color='red',grid=True)
```

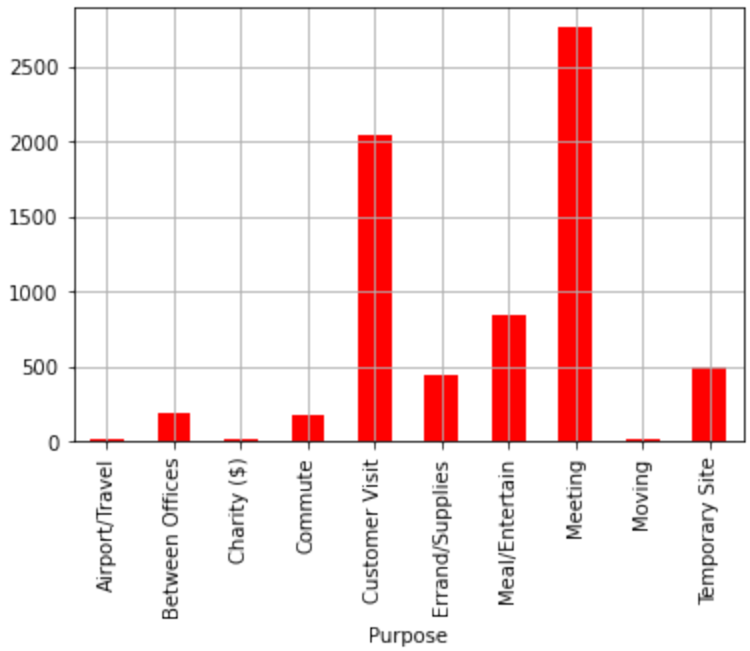
Out[280… <AxesSubplot:xlabel='Month'>



In [ ]:

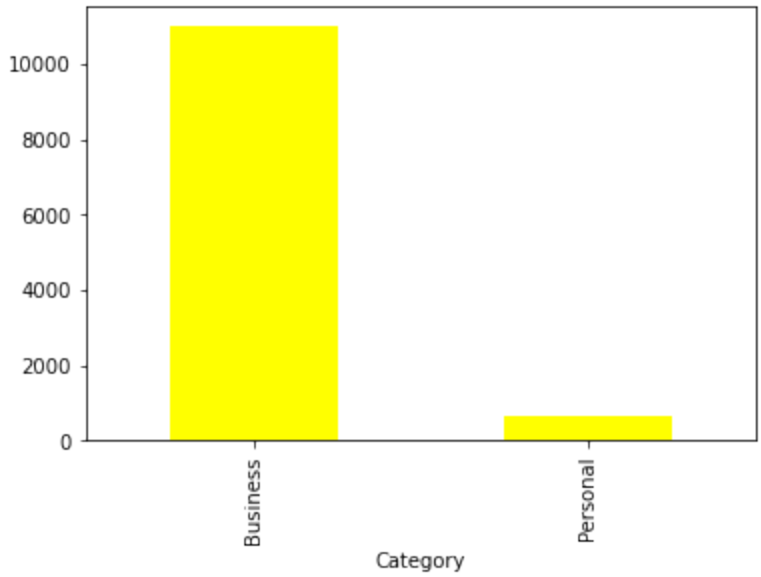
In [281…
```
df['Miles'].groupby(df['Purpose']).sum().plot(kind='bar',color='red', grid=True)
```

Out[281… <AxesSubplot:xlabel='Purpose'>



In [282…
```
df['Miles'].groupby(df['Category']).sum().plot(kind='bar',color='yellow')
```

Out[282… <AxesSubplot:xlabel='Category'>



In [ ]:

In [ ]: