Uber Drive Dataset

In [229…
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Step One: Import Data and do iniial quality check

Import related libraries, Read file \ Analyze the data and records using describe shape and info ... \ Check duplication \ Check null values \ check outliers \ check datatypes \ rename fields appropereiatly

In [230…
```
pd.set_option('display.max_columns', None)
```

In [212…
```
df=pd.read_csv('/Users/Rana/Desktop/Uber_drives.csv')
```

In [213…
```
df.shape
```

Out[213… (1156, 7)

In [214…
```
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 [215…
```
df.tail()
```

Out[215…

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
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
1155	Totals	NaN	NaN	NaN	NaN	12204.7	NaN

In [216…
```
df.duplicated().sum()
```

Out[216… 1

Step two: Data cleaning , wrangling

Remove duplicates\ Filter fields \ Rename fields \ Treat null values \ Check and change data types where needed \ identify any outliers \ Created new fields as needed

In [217…
```
# Remove unwanted rows
df = df.loc[df["START_DATE*"] != "Totals"]
```

In [218…
```
df=df.drop_duplicates()
```

In [219…
```
#Change data types rename fields
df=df.rename(columns={'MILES*':'Miles','START_DATE*':'Start_Date', 'END_DATE*':'End_Date'
```

In [220…
```
#Change data types
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')
```

In [221…
```
#Add new fields for analysis purposees
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
```

In [190…
```
# Fill Up null with 'Unknown'
df['Purpose'].fillna(value='Unknown' , inplace=True)
df['Category'].fillna(value='Unknown' , inplace=True)
```

Step3: To identify outliers ,

we will add new field that define the trips duration to measure it aginst milles travaled. \ And Exclude records with zero duration

In [222…
```
df['Duration']=df['End_Date']-df['Start_Date']
df['Duration_Time']=df['Duration'].dt.seconds
```

In [223…
```
df=df[df['Duration_Time']  != 0]
```

Find a relation between trips durations and miles.

In [120…
```
df
```

Out[120…

	Start_Date	End_Date	Category	START*	STOP*	Miles	Purpose	Month	Day	Start
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5	Meal/Entertain	1	Friday	21
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5	Unknown	1	Saturday	01
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4	Errand/Supplies	1	Saturday	20
3	2016-01-05 17:31:00	2016-01-05 17:45:00	Business	Fort Pierce	Fort Pierce	4	Meeting	1	Tuesday	17
4	2016-01-06 14:42:00	2016-01-06 15:49:00	Business	Fort Pierce	West Palm Beach	63	Customer Visit	1	Wednesday	14
...	...	...	...	...	...	...	...	...	...	
1150	2016-12-31 01:07:00	2016-12-31 01:14:00	Business	Kar?chi	Kar?chi	0	Meeting	12	Saturday	01
1151	2016-12-31 13:24:00	2016-12-31 13:42:00	Business	Kar?chi	Unknown Location	3	Temporary Site	12	Saturday	13
1152	2016-12-31 15:03:00	2016-12-31 15:38:00	Business	Unknown Location	Unknown Location	16	Meeting	12	Saturday	15
1153	2016-12-31 21:32:00	2016-12-31 21:50:00	Business	Katunayake	Gampaha	6	Temporary Site	12	Saturday	21
1154	2016-12-31 22:08:00	2016-12-31 23:51:00	Business	Gampaha	Ilukwatta	48	Temporary Site	12	Saturday	22

1150 rows × 13 columns

Explore the corelationship between duration of trip and miles Travelled

In [124…

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

Out[124… `0.8469995919057344`

In [224…

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

Out[224…

	Miles	Duration_Time
count	1150.000000	1150.000000
mean	10.090435	1399.356522
std	21.564375	1640.587024
min	0.000000	60.000000
25%	2.000000	600.000000

	Miles	Duration_Time
50%	6.000000	990.000000
75%	10.000000	1680.000000
max	310.000000	20160.000000

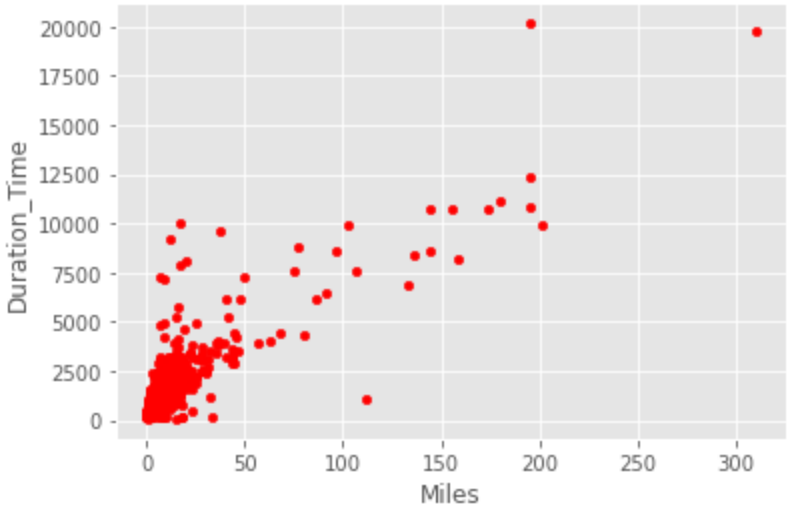
Key points to interpret this value:

A correlation coefficient of 0.8469995919057344 indicates a strong positive linear relationship between the two variables. Magnitude: 0.847 is close to 1, so the association is strong. Values rise together quite consistently. Direction: Positive, meaning as one variable increases, the other tends to increase as well. Linear assumption: This measure reflects linear correlation. Sensitivity to outliers: Correlation can be affected by outliers. A few extreme values can inflate or deflate the coefficient.

Scatter plot of Duration_Time vs Miles to assess the relationship and variability.

In [225…
```
df.plot.scatter(x='Miles', y='Duration_Time',color='red', grid=True)
```

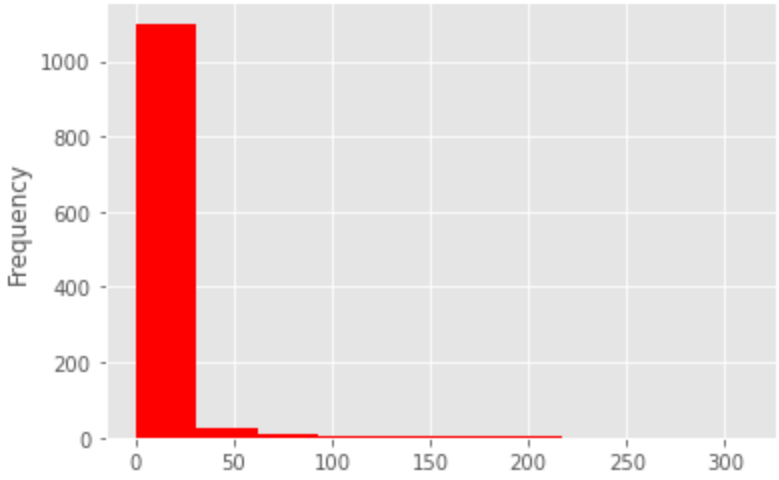
Out[225…
```
<AxesSubplot:xlabel='Miles', ylabel='Duration_Time'>
```



Handling and dicovering Miles distribution using histgram to confirm skewness.

In [80]:
```
df['Miles'].plot(kind='hist',color='red', grid=True)
```

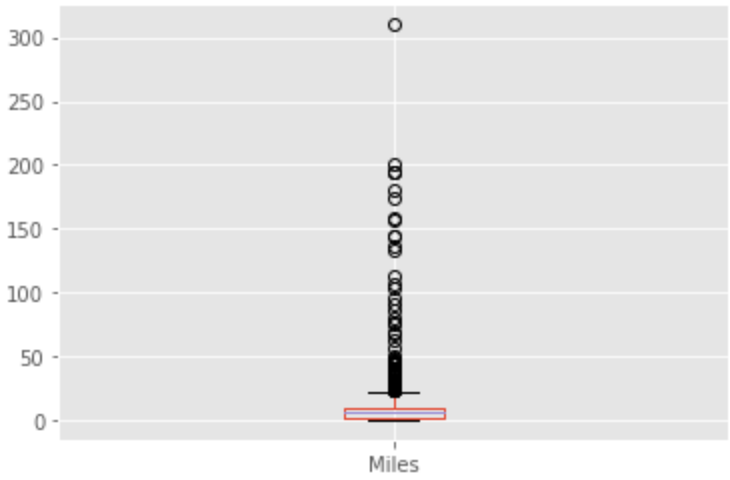
Out[80]:
```
<AxesSubplot:ylabel='Frequency'>
```



Boxplots to identify outliers

In [60]:
```
df['Miles'].plot(kind='box')
```

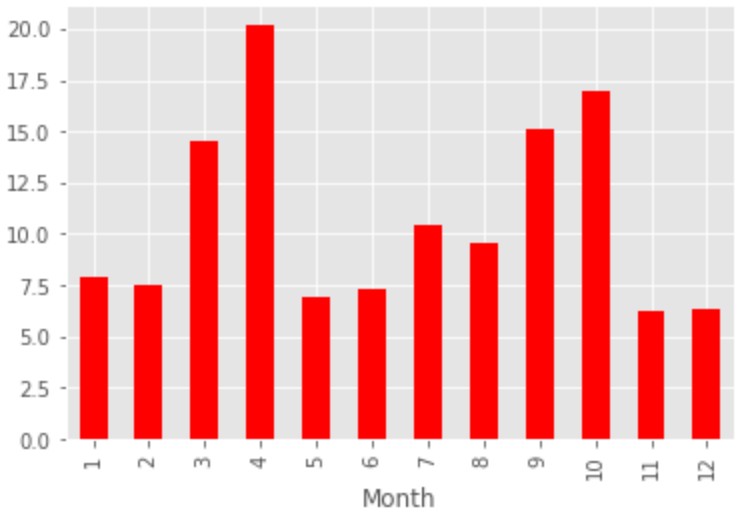
Out[60]: <AxesSubplot:>



MORE EDA

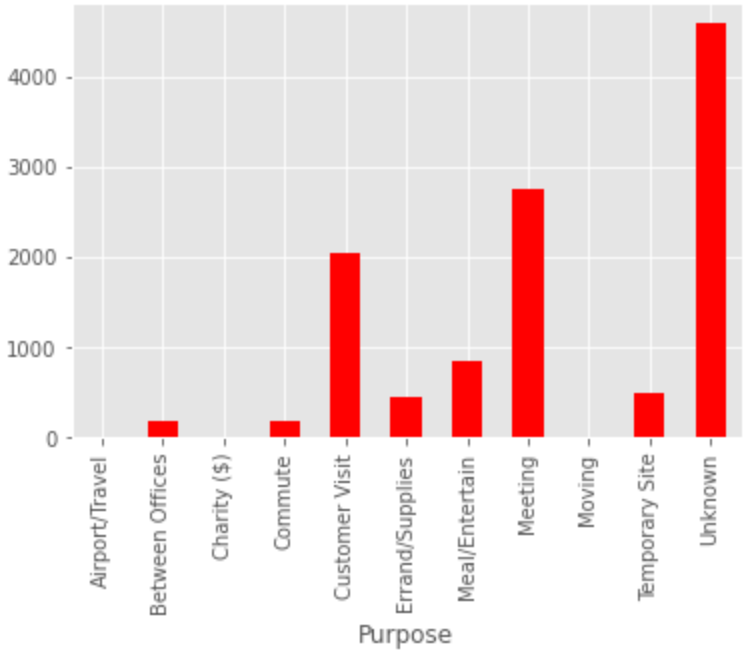
In [66]:
```
df['Miles'].groupby(df['Month']).mean().plot(kind='bar',color='red',grid=True)
```

Out[66]: <AxesSubplot:xlabel='Month'>



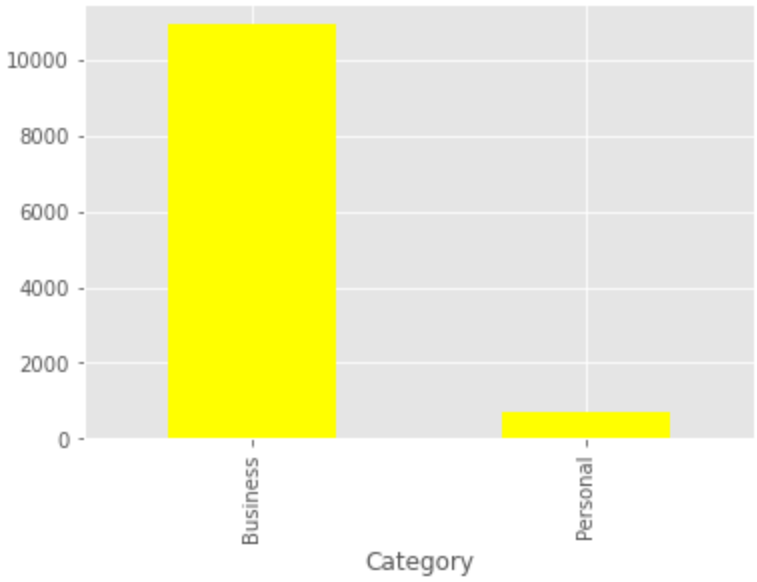
In [67]:
```
df['Miles'].groupby(df['Purpose']).sum().plot(kind='bar',color='red', grid=True)
```

Out[67]: <AxesSubplot:xlabel='Purpose'>



In [68]:
```
df['Miles'].groupby(df['Category']).sum().plot(kind='bar',color='yellow')
```

Out[68]: <AxesSubplot:xlabel='Category'>



Summary of Findings:

Observations: n = 1,150 Miles Mean: 10.09 miles Median (50th percentile): 6.00 miles Range: min 0.0, max 310.0 miles Spread: std dev ≈ 21.56 miles Q1–Q3: 2.0 miles (25th percentile) to 10.0 miles (75th percentile) There is a wide range with a long tail toward higher mile values, and the mean is higher than the median, suggesting right-skewness.Duration_Time Mean: 1,399.36 seconds (≈ 23.32 minutes) Median: 990.0 seconds (≈ 16.5 minutes) Range: min 60.0 s, max 20,160.0 s Spread: std dev ≈ 1,640.59 seconds Q1–Q3: 600 s to 1,680 s The mean is notably higher than the median, indicating substantial right skew and a few long-duration observations. The maximum duration is over 5.6 hours (20,160 s).Potential interpretations The dataset likely contains trips or runs with varying lengths: Short trips cluster around a few minutes (median ~16.5 minutes). A minority of trips are very long (max ~5.6 hours), pulling the mean up. If Miles and Duration_Time are paired per observation (i.e., each row is a trip with its distance and duration), you might expect a positive relationship

between Miles and Duration_Time, though the rate (time per mile) could vary (a few long trips could inflate duration but not proportionally add miles).