# Step-1 Importing libraries and read the data

In [53]:
```
import pandas as pd
import numpy as np
import datetime
import calendar
import matplotlib.pyplot as plt
import seaborn as sns
matplotlib.style.use('ggplot')
import calendar
```

In [96]:
```
data=pd.read_csv('C:/Users/vyz/Downloads/My Uber Drives - 2016.csv')
data.head()
```

Out[96]:

| | START_DATE* | END_DATE* | CATEGORY* | START* | STOP* | MILES* | PURPOSE* |
|---|---|---|---|---|---|---|---|
| 0 | 1/1/2016 21:11 | 1/1/2016 21:17 | Business | Fort Pierce | Fort Pierce | 5.1 | Meal/Entertain |
| 1 | 1/2/2016 1:25 | 1/2/2016 1:37 | Business | Fort Pierce | Fort Pierce | 5.0 | NaN |
| 2 | 1/2/2016 20:25 | 1/2/2016 20:38 | Business | Fort Pierce | Fort Pierce | 4.8 | Errand/Supplies |
| 3 | 1/5/2016 17:31 | 1/5/2016 17:45 | Business | Fort Pierce | Fort Pierce | 4.7 | Meeting |
| 4 | 1/6/2016 14:42 | 1/6/2016 16:49 | Business | Fort Pierce | West Palm Beach | 63.7 | Customer Visit |

# Step-2 Cleaning the data

In [55]:
```
data.tail()
```

Out[55]:

| | START_DATE* | END_DATE* | CATEGORY* | START* | STOP* | MILES* | PURPOSE* |
|---|---|---|---|---|---|---|---|
| 1151 | 12/31/2016 13:24 | 12/31/2016 13:42 | Business | Kar?ch | Unknown Location | 3.9 | Temporary Site |
| 1152 | 12/31/2016 15:03 | 12/31/2016 15:38 | Business | Unknown Location | Unknown Location | 16.2 | Meeting |
| 1153 | 12/31/2016 21:32 | 12/31/2016 21:50 | Business | Katunayake | Gampaha | 6.4 | Temporary Site |
| 1154 | 12/31/2016 22:08 | 12/31/2016 23:51 | Business | Gampaha | Ilukwatta | 48.2 | Temporary Site |
| 1155 | Totals | NaN | NaN | NaN | NaN | 12256.7 | NaN |

In [109]:
```
data=data[:-1]
```

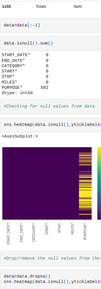
In [57]:
```
data.isnull().sum()
```

Out[57]:
```
START_DATE*     0
END_DATE*       0
CATEGORY*       0
START*          0
STOP*           0
MILES*          0
PURPOSE*      502
dtype: int64
```

In [58]:
```
#Checking for null values from data.

sns.heatmap(data.isnull(),yticklabels=False,cmap="viridis")
```

Out[58]:
```
<AxesSubplot:>
```



In [60]:
```
#Drop/remove the null values from the data.
```

In [99]:
```
data=data.dropna()
sns.heatmap(data.isnull(),yticklabels=False,cmap="viridis")
```

Out[99]:
```
<AxesSubplot:>
```

# Step-3 Transforming the data

In [62]:
```
# Getting an hour, day, days of the week, a month from the date of the trip.
```

In [105]:
```
data['START_DATE*'] = pd.to_datetime(data['START_DATE*'], format="%m/%d/%Y %H:%M")
data['END_DATE*'] = pd.to_datetime(data['END_DATE*'], format="%m/%d/%Y %H:%M")
```

In [106]:
```
hours=[]
days=[]
dayofweek=[]
month=[]
weekday=[]
for x in data['START_DATE*']:
    hours.append(x.hour)
    day.append(x.day)
    dayofweek.append(x.dayofweek)
    month.append(x.month)
    weekday.append(calendar.day_name[dayofweek[-1]])
data['HOURS']=hours
data['DAY']=day
data['DAY_OF_WEEK']=dayofweek
data['MONTH']=month
data['WEEKDAY']=weekday
```

In [107]:
```
data.head()
```

Out[107]:

| | START_DATE* | END_DATE* | CATEGORY* | START* | STOP* | MILES* | PURPOSE* | HOURS | DAY | DAY_OF_WEEK | MONTH | WEEKDAY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016-01-01 21:11:00 | 2016-01-01 21:17:00 | Business | Fort Pierce | Fort Pierce | 5.1 | Meal/Entertain | 21 | 1 | 4 | 1 | Friday |
| 2 | 2016-01-02 20:25:00 | 2016-01-02 20:38:00 | Business | Fort Pierce | Fort Pierce | 4.8 | Errand/Supplies | 20 | 2 | 5 | 1 | Saturday |
| 3 | 2016-01-05 17:31:00 | 2016-01-05 17:45:00 | Business | Fort Pierce | Fort Pierce | 4.7 | Meeting | 17 | 5 | 1 | 1 | Tuesday |
| 4 | 2016-01-06 14:42:00 | 2016-01-06 16:49:00 | Business | Fort Pierce | West Palm Beach | 63.7 | Customer Visit | 14 | 6 | 2 | 1 | Wednesday |
| 5 | 2016-01-06 17:15:00 | 2016-01-06 17:19:00 | Business | West Palm Beach | West Palm Beach | 4.3 | Meal/Entertain | 17 | 6 | 2 | 1 | Wednesday |

In [96]:
```
# finding traveling time
```

In [ ]:
```
# min->sec->min
```

In [123]:
```
time=[]
data['TRAVELLING_TIME']=data['END_DATE*']-data['START_DATE*']
for i in data['TRAVELLING_TIME']:
    time.append(i.seconds/60)
data['TRAVELLING_TIME']=time
data.head()
```

Out[123]:

| | START_DATE* | END_DATE* | CATEGORY* | START* | STOP* | MILES* | PURPOSE* | HOURS | DAY | DAY_OF_WEEK | MONTH | WEEKDAY | TRAVELLING_TIME |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016-01-01 21:11:00 | 2016-01-01 21:17:00 | Business | Fort Pierce | Fort Pierce | 5.1 | Meal/Entertain | 21 | 1 | 4 | 1 | Friday | 6.0 |
| 2 | 2016-01-02 20:25:00 | 2016-01-02 20:38:00 | Business | Fort Pierce | Fort Pierce | 4.8 | Errand/Supplies | 20 | 2 | 5 | 1 | Saturday | 13.0 |
| 3 | 2016-01-05 17:31:00 | 2016-01-05 17:45:00 | Business | Fort Pierce | Fort Pierce | 4.7 | Meeting | 17 | 5 | 1 | 1 | Tuesday | 14.0 |
| 4 | 2016-01-06 14:42:00 | 2016-01-06 16:49:00 | Business | Fort Pierce | West Palm Beach | 63.7 | Customer Visit | 14 | 6 | 2 | 1 | Wednesday | 67.0 |
| 5 | 2016-01-06 17:15:00 | 2016-01-06 17:19:00 | Business | West Palm Beach | West Palm Beach | 4.3 | Meal/Entertain | 17 | 6 | 2 | 1 | Wednesday | 4.0 |

In [97]:
```
# Calculating the average speed of the trip.
# min->sec
```

In [98]:
```
data['TRAVELLING_TIME']=data['TRAVELLING_TIME']/60
data['SPEED']=data['MILES*']/data['TRAVELLING_TIME']
data.head()
```

Out[98]:

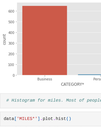
| | START_DATE* | END_DATE* | CATEGORY* | START* | STOP* | MILES* | PURPOSE* | HOUR | DAY | DAY_OF_WEEK | MONTH | WEEKDAY | TRAVELLING_TIME | SPEED |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016-01-01 21:11:00 | 2016-01-01 21:17:00 | Business | Fort Pierce | Fort Pierce | 5.1 | Meal/Entertain | 21 | 1 | 4 | 1 | Friday | 0.100000 | 51.000000 |
| 2 | 2016-01-02 20:25:00 | 2016-01-02 20:38:00 | Business | Fort Pierce | Fort Pierce | 4.8 | Errand/Supplies | 20 | 2 | 5 | 1 | Saturday | 0.216667 | 22.153846 |
| 3 | 2016-01-05 17:31:00 | 2016-01-05 17:45:00 | Business | Fort Pierce | Fort Pierce | 4.7 | Meeting | 17 | 5 | 1 | 1 | Tuesday | 0.233333 | 20.142857 |
| 4 | 2016-01-06 14:42:00 | 2016-01-06 16:49:00 | Business | Fort Pierce | West Palm Beach | 63.7 | Customer Visit | 14 | 6 | 2 | 1 | Wednesday | 1.116667 | 57.044776 |
| 5 | 2016-01-06 17:15:00 | 2016-01-06 17:19:00 | Business | West Palm Beach | West Palm Beach | 4.3 | Meal/Entertain | 17 | 6 | 2 | 1 | Wednesday | 0.066667 | 64.500000 |

# Step-4 Visualizing the data

In [100]:
```
# Different categories of data. From data, we can see most of the people use UBER for business purposes.
```

In [76]:
```
sns.countplot(x='CATEGORY*',data=data)
```

Out[76]:
```
<AxesSubplot:xlabel='CATEGORY*', ylabel='count'>
```



In [73]:
```
# Histogram for miles. Most of people not having a long trip.
```

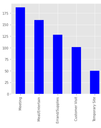
In [75]:
```
data['MILES*'].plot.hist()
```

Out[72]:
```
<AxesSubplot:ylabel="Frequency">
```

In [78]:
```
# Trips for purpose. Mostly the purpose of the trip is meeting and meal/entertain
```

In [74]:
```
data['PURPOSE*'].value_counts().plot(kind='bar',figsize=(10,5),color='blue')
```

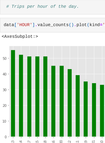
Out[74]:
```
<AxesSubplot:>
```



In [79]:
```
# Trips per hour of the day.
```

In [76]:
```
data['HOUR'].value_counts().plot(kind='bar',figsize=(10,5),color='green')
```

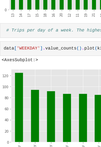
Out[76]:
```
<AxesSubplot:>
```



In [77]:
```
# Trips per day of a week. The highest number of trip on Friday.
```

In [78]:
```
data['WEEKDAY'].value_counts().plot(kind='bar',color='green')
```

Out[78]:
```
<AxesSubplot:>
```



In [79]:
```
# Trips per day of the month
```

In [80]:
```
data['DAY'].value_counts().plot(kind='bar',figsize=(15,5),color='green')
```

Out[80]:
```
<AxesSubplot:>
```

In [81]:
```
# Trips in a month.
```

In [82]:
```
data['MONTH'].value_counts().plot(kind='bar',figsize=(10,5),color='green')
```

Out[82]:
```
<AxesSubplot:>
```

In [83]:
```
# The starting points of trips. The highest number of people are from Cary who takes the trip.
```

In [84]:
```
data['START*'].value_counts().plot(kind='bar',figsize=(25,5),color='red')
```

Out[84]:
```
<AxesSubplot:>
```

In [85]:
```
# Comparing all the purpose with miles, hour, day of the month, day of the week, month, Travelling time.
```

In [87]:
```
data.groupby('PURPOSE*').mean().plot(kind='bar',figsize=(15,5))
```

Out[87]:
```
<AxesSubplot:xlabel='PURPOSE*'>
```