In [1]:
```
#Data Discovery

# Libraries for handling numeric computation and dataframes
import pandas as pd
import numpy as np

# Libraries for statistical plotting
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# My personal data storaged in my Github repository
rides = pd.read_csv('https://raw.githubusercontent.com/sameerakhtari/Exploratory-Data-Analysis-on-Uber-Rides-Dataset/main/raw-data/My%20Uber%20Drives%20-%202016.csv')
```

In [2]:
```
rides.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*       1007 non-null   object 
 4   STOP*        1006 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 [3]:
```
rides.rename(columns={'START_DATE*': 'start_date', 'END_DATE*': 'end_date','CATEGORY*':'category','START*':'start',
                      'STOP*':'stop','MILES*':'miles','PURPOSE*':'purpose'}, inplace=True)
```

In [4]:
```
rides.head()
```

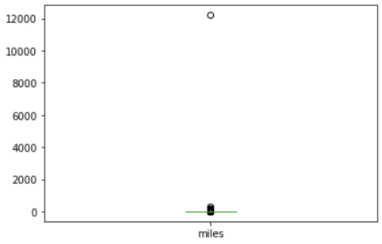
Out[4]:

	start_date	end_date	category	start	stop	miles	purpose
0	01/01/2016 21:11	01/01/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	01/02/2016 01:25	01/02/2016 01:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	01/02/2016 20:25	01/02/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	01/05/2016 17:31	01/05/2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	01/06/2016 14:42	01/06/2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

In [5]:
```
#creating an additional dataframe without Uber EATS records (out of analysis scope)
df1 = rides[rides.category!='UberEATS Marketplace'][['miles']]

print(df1.describe().transpose().round(1));
df1.boxplot(grid=False);
```

```
        count  mean    std  min  25%  50%   75%      max
miles  1156.0  21.1  359.3  0.5  2.9  6.0  10.4  12204.7
```



In [ ]:
```
#Data Preparation
```

In [6]:
```
rides.isnull().sum().sort_values(ascending=False)
```

Out[6]:
```
purpose       503
stop          150
start         149
end_date        1
category        1
start_date      0
miles           0
dtype: int64
```

In [7]:
```
rides[rides.category.isnull()]
```

Out[7]:

	start_date	end_date	category	start	stop	miles	purpose
1155	Totals	NaN	NaN	NaN	NaN	12204.7	NaN

In [8]:
```
rides.dropna(subset = ['category'], inplace=True)
```

In [9]:
```
rides[rides.category.isnull()]
```

Out[9]:

start_date	end_date	category	start	stop	miles	purpose

In [10]:
```
rides[rides.end_date.isnull()]
```

Out[10]:

start_date	end_date	category	start	stop	miles	purpose

In [11]:
```
rides.isnull().sum().sort_values(ascending=False)
```

Out[11]:
```
purpose       502
stop          149
start         148
start_date      0
end_date        0
category        0
```

```
miles          0
dtype: int64
```

In [12]:
```
rides[rides.start.isnull()]
```

Out[12]:

	start_date	end_date	category	start	stop	miles	purpose
109	2/16/2016 8:29	2/16/2016 9:34	Business	NaN	Colombo	14.1	NaN
117	2/17/2016 13:18	2/17/2016 14:04	Business	NaN	Colombo	14.7	Temporary Site
121	2/18/2016 8:19	2/18/2016 8:27	Business	NaN	NaN	23.5	Temporary Site
122	2/18/2016 14:03	2/18/2016 14:45	Business	NaN	Islamabad	12.7	Temporary Site
124	2/18/2016 18:44	2/18/2016 18:58	Business	NaN	Islamabad	5.2	Customer Visit
...	...	...	...	...	...	...	...
1129	12/28/2016 17:02	12/28/2016 17:16	Business	NaN	Karachi	4.4	Errand/Supplies
1134	12/29/2016 11:28	12/29/2016 12:00	Business	NaN	Karachi	11.9	Meal/Entertain
1141	12/29/2016 19:50	12/29/2016 20:10	Business	NaN	Karachi	4.1	Customer Visit
1144	12/29/2016 23:14	12/29/2016 23:47	Business	NaN	Karachi	12.9	Meeting
1152	12/31/2016 15:03	12/31/2016 15:38	Business	NaN	NaN	16.2	Meeting

148 rows × 7 columns

In [13]:
```
rides.dropna(subset = ['start'], inplace=True)
```

In [14]:
```
rides.dropna(subset = ['stop'], inplace=True)
```

In [15]:
```
rides.isnull().sum().sort_values(ascending=False)
```

Out[15]:
```
purpose       372
start_date      0
end_date        0
category        0
start           0
stop            0
miles           0
dtype: int64
```

In [16]:
```
# Checking categories in product_type column
print(rides.purpose.value_counts())
```

```
Meeting            164
Meal/Entertain     148
Errand/Supplies    111
Customer Visit      92
Temporary Site      32
Between Offices     18
Moving               4
Commute              1
Airport/Travel       1
Charity ($)          1
Name: purpose, dtype: int64
```

In [17]:
```
# Library for manipulating dates and times
from datetime import datetime
from datetime import timedelta

# Function to convert features to datetime
def date_convertion(df, cols):

  for col in cols:
    df[col] = df[col].apply(lambda x: x.replace(' +0000 UTC', ''))
    df[col] = pd.to_datetime(df[col])

  return df

# Applying date_convertion function to date features
rides = date_convertion(rides, ['start_date', 'end_date'])
```

In [18]:
```
rides['month'] = rides.start_date.map(lambda x: datetime.strftime(x,"%b"))
```

In [19]:
```
rides['weekday'] = rides.start_date.map(lambda x: datetime.strftime(x,"%a"))
```

In [20]:
```
rides['year'] = rides.start_date.map(lambda x: datetime.strftime(x,"%Y"))
```

In [21]:
```
rides['time'] = rides.start_date.map(lambda x: datetime.strftime(x,"%H:%M"))
```

In [22]:
```
rides['month'] = rides.end_date.map(lambda x: datetime.strftime(x,"%b"))
```

In [23]:
```
rides['weekday'] = rides.end_date.map(lambda x: datetime.strftime(x,"%a"))
```

In [24]:
```
rides['year'] = rides.end_date.map(lambda x: datetime.strftime(x,"%Y"))
```

In [25]:
```
rides['time'] = rides.end_date.map(lambda x: datetime.strftime(x,"%H:%M"))
```

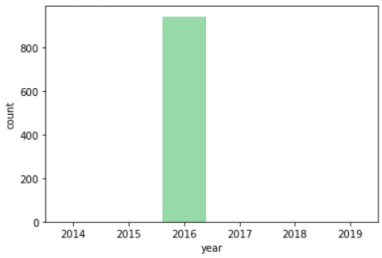
In [26]:
```
#Now Finding ride time
rides['request_lead_time'] = rides.end_date - rides.start_date
rides['request_lead_time'] = rides['request_lead_time'].apply(lambda x: round(x.total_seconds()/60,1))
```

In [28]:
```
#Data Analysis Time
```

In [29]:
```
#listing completed rides
completed_rides = rides[(rides.end_date!='')]
```

In [30]:
```
#A). How many trips I did over the years?
print('Total trips: ', completed_rides.end_date.count())
print(completed_rides.year.value_counts().sort_index(ascending=True))
sns.countplot(data=completed_rides, x='year',order=['2014','2015','2016','2017','2018','2019'], palette='pastel');
```

```
Total trips:  944
2016    944
Name: year, dtype: int64
```

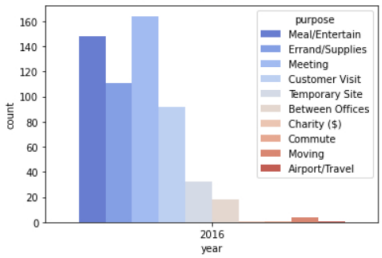


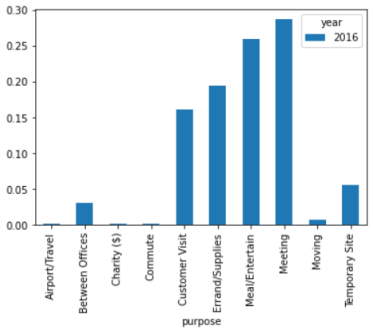
In [31]:

```
#B). How many trips were Completed on what Purpose?
print('Total trips: ', rides.end_date.count())
print(round(rides.end_date.value_counts()/rides.end_date.size*100,1))

sns.countplot(data=rides, x='year', order=['2016'], hue='purpose', palette='coolwarm');
rides.groupby(by=['year'])['purpose'].value_counts(normalize=True).unstack('year').plot.bar(stacked=True);
```

```
Total trips:  944
2016-06-28 23:59:00    0.2
2016-07-21 17:51:00    0.1
2016-11-02 17:11:00    0.1
2016-01-01 21:17:00    0.1
2016-12-07 12:46:00    0.1
                      ...
2016-05-10 17:31:00    0.1
2016-03-10 10:37:00    0.1
2016-12-10 22:21:00    0.1
2016-05-22 18:53:00    0.1
2016-07-01 20:24:00    0.1
Name: end_date, Length: 943, dtype: float64
```





In [43]:

```
#C). For What reason Went to what place...?!?



rides.groupby(by=['stop'])['purpose'].value_counts(normalize=True).unstack('stop').plot.bar(stacked=True);
```


Harden Place
Hayesville
Hazelwood
Hell's Kitchen
Heritage Pines
Hog Island
Holly Springs
Houston
Hudson Square
Ilukwatta
Islamabad
Jacksonville
Jamaica
Jamestown Court
Karachi
Katunayaka
Katy
Kenner
Kildaire Farms
Kips Bay
Kissimmee
Lahore
Lake Reams
Lakeview
Latta
Leesville Hollow
Lexington Park at Amberly
Long Island City
Lower Garden District
Lower Manhattan
Macgregor Downs
Mcvan
Menlo Park
Meredith Townes
Metairie
Midtown
Midtown East
Midtown West
Morrisville
Mountain View
New Orleans
New York
Newark
NoMad
Noorpur Shahan
North Austin
Northwoods
Nugegoda
Oakland
Orlando
Palm Beach
Palo Alto
Parkway
Parkway Museums
Parkwood
Pontchartrain Shores
Port Bolivar
Potrero Flats
Preston
Queens
Queens County
Raleigh
Rawalpindi
Redmond
Ridgeland
San Francisco
Savon Height
Seattle
Sharpstown
Soho
South
South Congress
Southside
Stonewater
Sugar Land
Summerwinds
Sunnyvale
Tanglewood
The Drag
Topton
Tribeca
Tudor City
Umstead
University District
Wake Co.
Walnut Terrace
Washington Avenue
Waverly Place
Wayne Ridge
West Palm Beach
West University
Weston
Westpark Place
Whitebridge
Williamsburg Manor
Winston Salem
0
5
10
15
20
25
30
35
40
Airport/Travel
Between Offices
Charity ($)
Commute
Customer Visit
Errand/Supplies
Meal/Entertain
Meeting
Moving
Temporary Site
purpose