Pyber Ride Sharing

Analysis

*** We can observe that ride sharing is more popular in Urban cities. This can be determined by the large number of drivers within that category type. Also, the fare rate range is almost at a standard rate and in general it just seems to have the most people.**

*** I am also able confirm that all of the cities that are considered to be in the suburban category are the 2nd most popular for pyber ride sharing. This observation can be gathered from either one of the generated graphs. However, if we look at the bubble chart, we can determine that there's a trend on the average fair rate and number of rides. This is most likely happening because of the daily commuters that live in the outskirts of the city and consider this service as an option to get to work every day.**

*** The rate and number of rides available for rural areas is limited because of the large distance between those cities and all of the other ones close to the city. The rate per day is higher and this could be due to the limited number of drivers and also the limited number of drivers who are willing to drive the distance.**

In [280]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
#!ls
```

In [199]:
```
city_dat  = 'HW5/Instructions/Pyber/raw_data/city_data.csv'
ride_dat = 'HW5/Instructions/Pyber/raw_data/ride_data.csv'
```

In [281]:
```
city_df = pd.read_csv(city_dat)
city_df.head(3)
#city_df.count()
city_df
#listdups = city_df.groupby('city')['city'].count()
#listdups
#city_df = city_df.drop_duplicates('city')
city_df['type'].value_counts()
```

Out[281]:
```
Urban       66
Suburban    36
Rural       18
Name: type, dtype: int64
```

In [283]:
```
ride_df = pd.read_csv(ride_dat)
ride_df.head(3)
#ride_df.count()
#ride_df
#ridedups = ride_df.groupby('city')['city'].count()
#ridedups
```

Out[283]:

	city	date	fare	ride_id
0	Lake Jonathanshire	2018-01-14 10:14:22	13.83	5739410935873
1	South Michelleport	2018-03-04 18:24:09	30.24	2343912425577
2	Port Samanthamouth	2018-02-24 04:29:00	33.44	2005065760003

In [5]:
```
# I am going to have to merge/join these two data set to be able to use
 this data asa single dataset for analysis
```

In [284]:
```
combined_dat = pd.merge(city_df, # city data
                        ride_df, # ride data
                        on='city', how = 'outer') #join on city column
```

In [285]:
```
#merging keeps common values in both left and right. Only rows with city
 should be return.
# I am going to verfify that
#print(city_df.shape)
#(120, 3)
#print(ride_df.shape)
#((2375, 4))
#print(combined_dat.shape)
#(2375, 6)
# validate by checking the number of values in commomn
combined_dat.count()
combined_dat
city_df['city'].isin(ride_df['city']).value_counts()
```

Out[285]:
```
True    120
Name: city, dtype: int64
```

In [234]:
```
#cleanup
data_set.drop_duplicates()
data_set = combined_dat.dropna(axis=0, how='all')
```

In [236]:
```
#data_set.count()
data_set.head(4)
#data_set.describe()
#data_set.mean()
#data_set.mean(axis=1)
```

Out[236]:

	city	driver_count	type	date	fare	ride_id
0	Richardfort	38	Urban	2018-02-24 08:40:38	13.93	5628545007794
1	Richardfort	38	Urban	2018-02-13 12:46:07	14.00	910050116494
2	Richardfort	38	Urban	2018-02-16 13:52:19	17.92	820639054416
3	Richardfort	38	Urban	2018-02-01 20:18:28	10.26	9554935945413

Bubble Plot of Ride Sharing Data

In [ ]:
```
```

In [238]:
```
#Your objective is to build a Bubble Plot that showcases the relationshi
p between four key variables:


#Average Fare ($) Per City
#Total Number of Rides Per City
#Total Number of Drivers Per City
#City Type (Urban, Suburban, Rural)


def showcase(column):
    d = {}
    d['Average Fare ($) Per City'] = column['fare'].mean()
    d['Total Number of Rides Per City'] = column['ride_id'].count()
    d['Total Number of Drivers Per City'] = column['driver_count'].mean
()
    d['City Type'] = column['type'].values[0]
    return pd.Series(d, index=['Average Fare ($) Per City',
                               'Total Number of Rides Per City',
                               'Total Number of Drivers Per City',
                               'City Type'])

new_data = data_set.groupby('city').apply(showcase)
```

In [239]:
```
#data_set.loc['MEAN']=data_set.mean(axis=0)
```

In [242]:
```
new_data.head()
```

Out[242]:

city	Average Fare ($) Per City	Total Number of Rides Per City	Total Number of Drivers Per City	City Type
Amandaburgh	24.641667	18	12.0	Urban
Barajasview	25.332273	22	26.0	Urban
Barronchester	36.422500	16	11.0	Suburban
Bethanyland	32.956111	18	22.0	Suburban
Bradshawfurt	40.064000	10	7.0	Rural

In [ ]:
```
# city types on chart : URBAN SUBURBAN RURAL
```

In [243]:
```
URBAN = new_data.loc[new_data['City Type']=='Urban']
SUBURBAN = new_data.loc[new_data['City Type']=='Suburban']
RURAL = new_data.loc[new_data['City Type']=='Rural']
```

In [256]:
```
URBAN.head(3)
```

Out[256]:

city	Average Fare ($) Per City	Total Number of Rides Per City	Total Number of Drivers Per City	City Type
Amandaburgh	24.641667	18	12.0	Urban
Barajasview	25.332273	22	26.0	Urban
Carriemouth	28.314444	27	52.0	Urban

In [245]:
```
SUBURBAN.head(3)
```

Out[245]:

city	Average Fare ($) Per City	Total Number of Rides Per City	Total Number of Drivers Per City	City Type
Barronchester	36.422500	16	11.0	Suburban
Bethanyland	32.956111	18	22.0	Suburban
Brandonfort	35.437368	19	10.0	Suburban

In [257]:
```
RURAL.head(3)
```

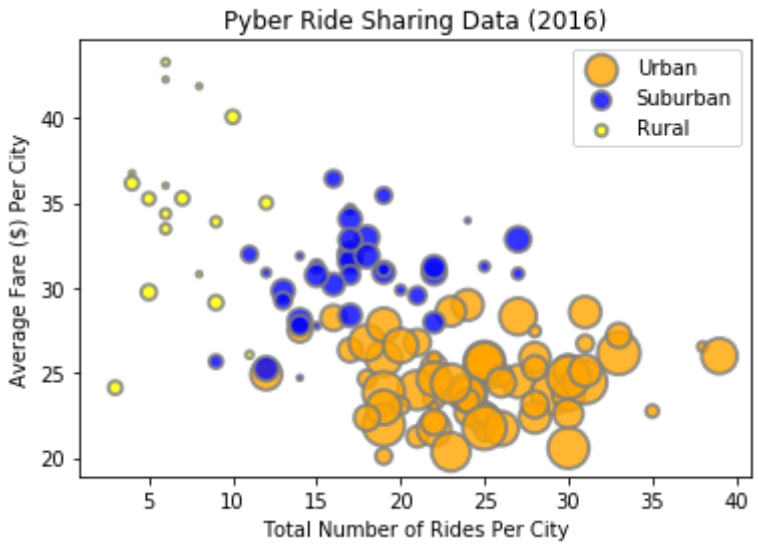
Out[257]:

city	Average Fare ($) Per City	Total Number of Rides Per City	Total Number of Drivers Per City	City Type
Bradshawfurt	40.064000	10	7.0	Rural
Garzaport	24.123333	3	7.0	Rural
Harringtonfort	33.470000	6	4.0	Rural

In [263]:
```
plt.scatter(x = URBAN['Total Number of Rides Per City'],
            y = URBAN['Average Fare ($) Per City'],s = URBAN['Total Numb
er of Drivers Per City']* 6.5,
            color= 'orange', label='Urban',alpha = 0.80,edgecolors="gre
y", linewidth=2)
plt.scatter(x = SUBURBAN['Total Number of Rides Per City'],
            y = SUBURBAN['Average Fare ($) Per City'],s = SUBURBAN['Tota
l Number of Drivers Per City']*6.5,
            color= 'blue', label='Suburban', alpha = 0.80,edgecolors="gr
ey", linewidth=2)
plt.scatter(x= RURAL['Total Number of Rides Per City'],
            y = RURAL['Average Fare ($) Per City'],s = RURAL['Total Numb
er of Drivers Per City']* 6.5,
            color= 'yellow', label='Rural', alpha = 0.80,edgecolors="gre
y", linewidth=2)

# Add titles (main and on axis)
plt.xlabel('Total Number of Rides Per City')
plt.ylabel('Average Fare ($) Per City')
plt.title('Pyber Ride Sharing Data (2016)')
plt.legend(loc='upper right')
```

Out[263]: <matplotlib.legend.Legend at 0x1a16087898>



In [134]:
```
plt.show()
```

In [249]:
```
#In addition, you will be expected to produce the following three pie ch
arts:


#% of Total Fares by City Type
#% of Total Rides by City Type
#% of Total Drivers by City Type
```

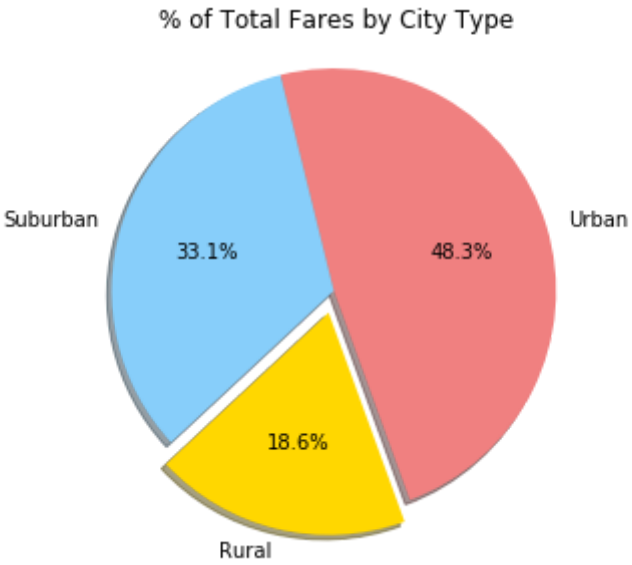
In [269]:
```
#URBAN['Average Fare ($) Per City'].sum()
# 1616.9420333079977
#SUBURBAN['Average Fare ($) Per City'].sum()
# 1106.5427292051343
#RURAL['Average Fare ($) Per City'].sum()
#623.4797705627706
```

Total Fares by City Type

In [279]:
```
labels = 'Urban', 'Suburban', 'Rural'
sizes = [1616.94, 1106.54, 623.48]
colors = ['lightcoral', 'lightskyblue', 'gold']
explode = (0, 0, 0.1)

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=290)

plt.axis('equal')
plt.tight_layout()
plt.title('% of Total Fares by City Type')
plt.show()
```

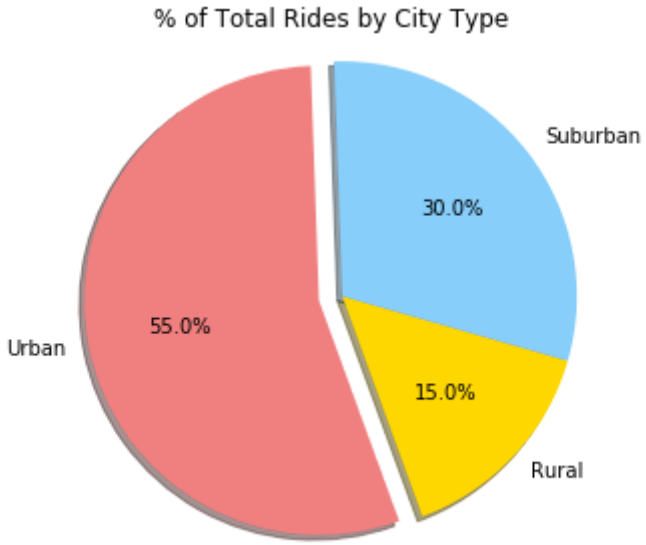


Total Rides by City Type

In [278]:
```
#% of Total Rides by City Type
ridesizes = new_data.groupby(['City Type'])['Total Number of Rides Per City'].count()
labels = 'Rural', 'Suburban', 'Urban'
colors = ['gold', 'lightskyblue', 'lightcoral']
explode = (0, 0, 0.1)

# Plot
plt.pie(ridesizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=290)

plt.axis('equal')
plt.tight_layout()
plt.title('% of Total Rides by City Type')
plt.show()
```

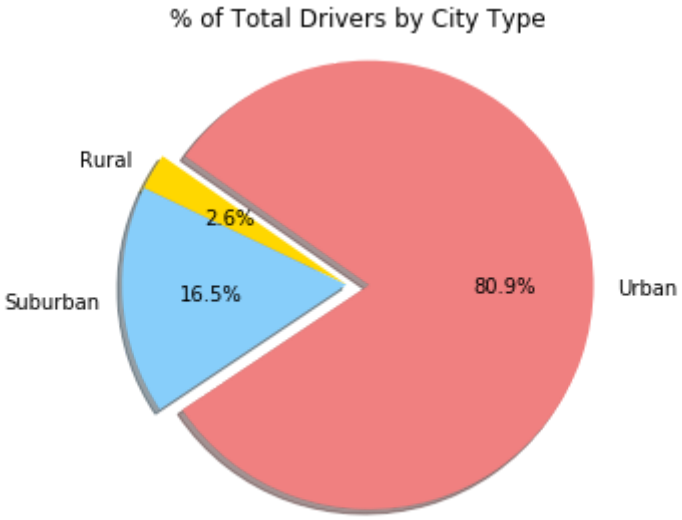


Total Drivers by City Type

In [277]:
```
#% of Total Drivers by City Type
driversizes = new_data.groupby(['City Type'])['Total Number of Drivers P
er City'].sum()
labels = 'Rural', 'Suburban', 'Urban'
colors = ['gold', 'lightskyblue', 'lightcoral']
explode = (0, 0, 0.1)

# Plot
plt.pie(driversizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=145)

plt.axis('equal')
plt.title('% of Total Drivers by City Type')
plt.tight_layout()
plt.show()
```



In [ ]:

In [ ]: