Plot_Uber

March 19, 2018

Observed Trends: 1. Even though there is a much larger number of drivers in urban areas than in suburban or rural, the average fare prices are much lower. This could be due to the fact that there are many other modes of transportation in urban areas and the drivers in these cities would have to compete for passengers. This in turn would cause the fares to lower. On the other, rural drivers have the option of hiking up their prices because their passengers won't have many other options and will have to take the modes of transportation they can get. 2. The pie charts showing % of Total Fares by City Type, % of Total Rides by City Type, and % of Drivers by City Type make it quite clear that demand for Uber is extremely high in urban areas. 3. Due to the size of the bubbles in the bubble plot, which correlates to the number of drivers in a city, we can infer that the population in urban areas is higher than that in suburban or rural areas. This can be assumed because there are many more drivers in urban areas indicating that there is a larger need for rides in those areas.

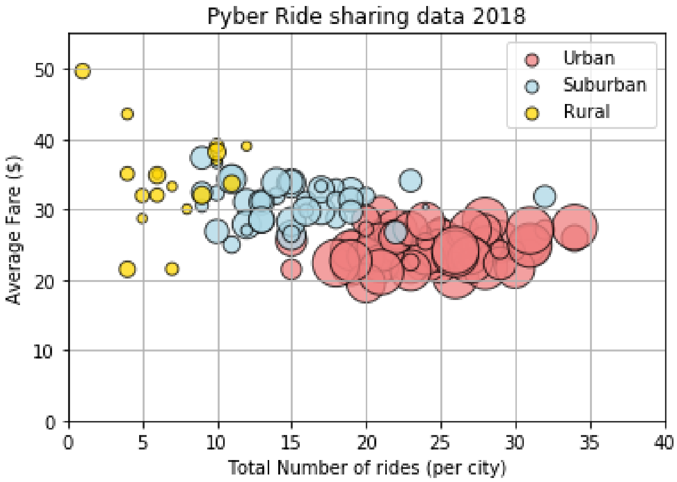
```
In [1]: # Import required libraries and read the files into dataframe
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns


        file1 = ("raw_data/city_data.csv")
        file2 = ("raw_data/ride_data.csv")

        city_data_df = pd.read_csv(file1)
        ride_data_df = pd.read_csv(file2)
```

```
In [2]: #Merge dataframes

        city_ride_df1 = pd.merge(ride_data_df,city_data_df, how = 'inner',on = 'city')
        #city_ride_df.to_csv("city_ride.csv",sep =",")
        city_ride_df = city_ride_df1.drop_duplicates(subset=["fare","ride_id"],keep="first")
        city_ride_df.head()
```

```
Out[2]:        city                 date   fare        ride_id  driver_count   type
        0  Sarabury  2016-01-16 13:49:27  38.35  5403689035038            46  Urban
        1  Sarabury  2016-07-23 07:42:44  21.76  7546681945283            46  Urban
        2  Sarabury  2016-04-02 04:32:25  38.03  4932495851866            46  Urban
        3  Sarabury  2016-06-23 05:03:41  26.82  6711035373406            46  Urban
        4  Sarabury  2016-09-30 12:48:34  30.30  6388737278232            46  Urban
```

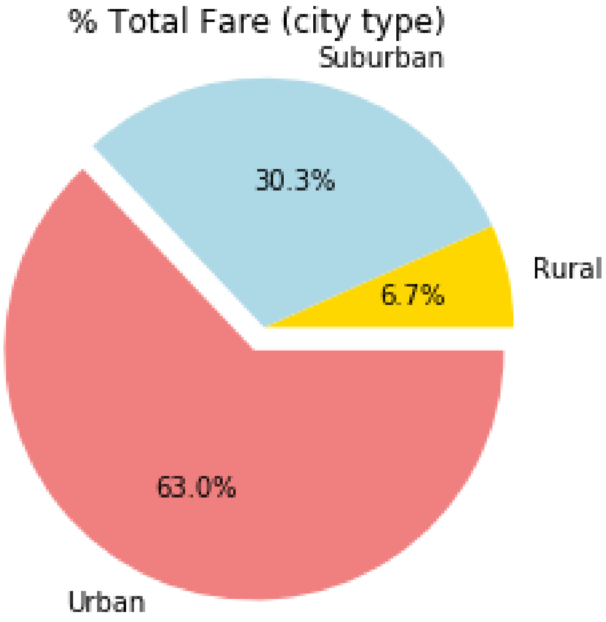
```
In [3]: #Urban,Sub-urban,Rural Dataframes
        urban_df = city_ride_df.loc[city_ride_df["type"]=="Urban"]
        surban_df = city_ride_df.loc[city_ride_df["type"]=="Suburban"]
        rural_df = city_ride_df.loc[city_ride_df["type"]=="Rural"]
```

```
In [4]: #Total Rides per city
        urban_rides = urban_df.groupby("city")["ride_id"].count()
        surban_rides = surban_df.groupby("city")["ride_id"].count()
        rural_rides = rural_df.groupby("city")["ride_id"].count()

        #Avg Fare per city
        urban_avg = round(urban_df.groupby("city")["fare"].mean(),2)
        surban_avg = round(surban_df.groupby("city")["fare"].mean(),2)
        rural_avg = round(rural_df.groupby("city")["fare"].mean(),2)

        #Total Drivers per city
        urban_df_unique = urban_df.drop_duplicates(subset=["city","driver_count"], keep='first
        surban_df_unique = surban_df.drop_duplicates(subset=["city","driver_count"], keep='firs
        rural_df_unique = rural_df.drop_duplicates(subset=["city","driver_count"], keep='first

        urban_drivers = urban_df_unique.groupby("city")["driver_count"].sum()
        surban_drivers = surban_df_unique.groupby("city")["driver_count"].sum()
        rural_drivers = rural_df_unique.groupby("city")["driver_count"].sum()

        urban_plot_df = pd.DataFrame({"Urban Avg": urban_avg,"Urban Rides":urban_rides,"Urban
        surban_plot_df = pd.DataFrame({"Suburban Avg": surban_avg,"Suburban Rides":surban_rides
        rural_plot_df = pd.DataFrame({"Rural Avg": rural_avg,"Rural Rides":rural_rides,"Rural

        plt.scatter(urban_rides,urban_avg,s=urban_drivers*10,color = 'lightcoral', edgecolor='
        plt.scatter(surban_rides,surban_avg,s=surban_drivers*10,color = 'lightblue', edgecolor=
        plt.scatter(rural_rides,rural_avg,s=rural_drivers*10,color = 'gold', edgecolor='black'
        plt.title('Pyber Ride sharing data 2018')
        plt.xlabel('Total Number of rides (per city)')
        plt.ylabel('Average Fare ($)')
        plt.xlim(0, 40)
        plt.ylim(0, 55)
        lgnd = plt.legend(scatterpoints=1)
        lgnd.legendHandles[0]._sizes = [50]
        lgnd.legendHandles[1]._sizes = [50]
        lgnd.legendHandles[2]._sizes = [50]
        plt.annotate(s='Note:\nCircle size correlates with driver count per city', xy=(0,15),
        plt.grid()
        plt.show()
```



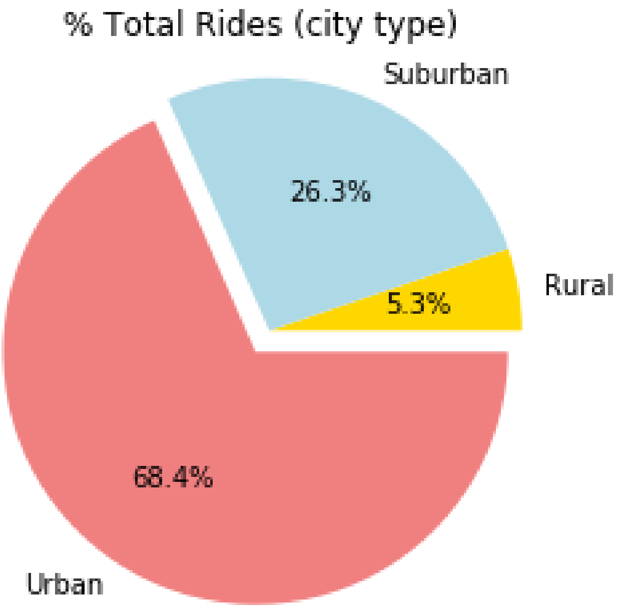
Note:
Circle size correlates with driver count per city

```
In [5]: #Total Fare by city type
        total_fare = round(city_ride_df.groupby("type")["fare"].sum(),1)
        plt.pie(total_fare,explode=(0,0,0.1),colors = ["gold", "lightblue", "lightcoral"],autop
        plt.axis("equal")
        plt.title("% Total Fare (city type)")
        plt.show()
```

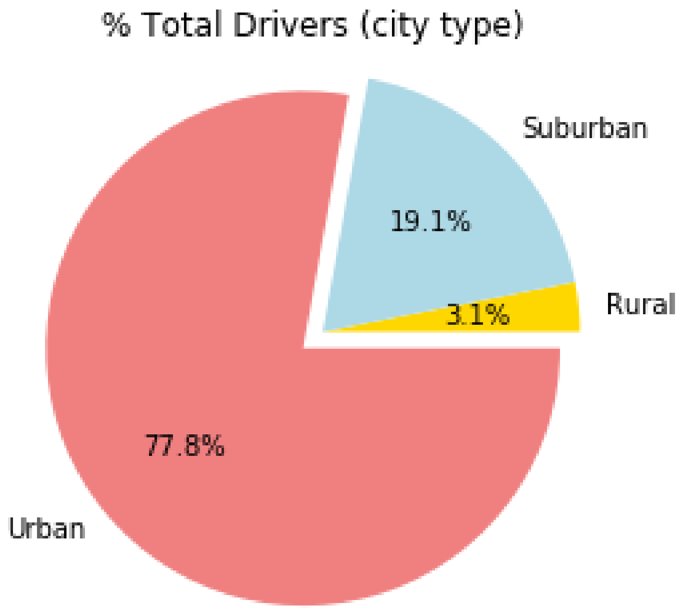


```
In [6]: #Total Rides by city type
        total_rides = city_ride_df.groupby("type")["ride_id"].count()
```

```
plt.pie(total_rides,explode=(0,0,0.1),colors = ["gold", "lightblue", "lightcoral"],auto
plt.axis("equal")
plt.title("% Total Rides (city type)")
plt.show()
```



```
In [7]: #Total Drivers by city type
        city_ride_dup = city_data_df.drop_duplicates(["city","driver_count"],keep='first')
        total_drivers = city_ride_dup.groupby("type")["driver_count"].sum()
        plt.pie(total_drivers,explode=(0,0,0.1),colors = ["gold", "lightblue", "lightcoral"],au
        plt.axis("equal")
        plt.title("% Total Drivers (city type)")
        plt.show()
```


% Total Drivers (city type)
Suburban
19.1%
Rural
3.1%
77.8%
Urban