matplotlib-pyber

Analysis

1. Urban areas have the greatest nuumber of drivers and rides
2. Rural areas have the lowest number of drivers and rides
3. The average fare per ride is lower in urban areas than in suburban and rual areas

In [1]:
```
# Dependencies

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
```

In [2]:
```
# Read CSV

city_path = os.path.join("city_data.csv")
ride_path = os.path.join("ride_data.csv")
city_data = pd.read_csv(city_path)
ride_data = pd.read_csv(ride_path)

#merging data

combined_data = pd.merge(city_data, ride_data, on="city")
combined_data.head()
```

Out[2]:

	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
4	Richardfort	38	Urban	2018-04-17 02:26:37	23.00	720020655850

In [3]:

```
#Average Fare Per City
average_fare = combined_data.groupby("city")["fare"].mean()
average_fare = pd.DataFrame(average_fare).reset_index()
average_fare = average_fare.rename(columns = {'fare': 'avg_fare'})
average_fare
```

Out[3]:

	city	avg_fare
0	Amandaburgh	24.641667
1	Barajasview	25.332273
2	Barronchester	36.422500
3	Bethanyland	32.956111
4	Bradshawfurt	40.064000
5	Brandonfort	35.437368
6	Carriemouth	28.314444
7	Christopherfurt	24.501852
8	Colemanland	30.894545
9	Davidfurt	31.995882
10	Deanville	25.842632
11	East Aaronbury	25.661111
12	East Danielview	31.560588
13	East Kaylahaven	23.757931
14	East Kentstad	29.823077
15	East Marymouth	30.835185
16	Erikaland	24.906667
17	Garzaport	24.123333
18	Grahamburgh	25.221200
19	Grayville	27.763333
20	Harringtonfort	33.470000
21	Huntermouth	28.993750
22	Hurleymouth	25.891429
23	Jerryton	25.649200
24	Jessicaport	36.013333
25	Johnton	26.785714
26	Joneschester	22.289600
27	Josephside	32.858148
28	Justinberg	23.694333
29	Karenberg	26.340000
...	...	...

	city	avg_fare
90	South Evanton	26.726129
91	South Jack	22.965263
92	South Jennifer	35.264286
93	South Karenland	26.535526
94	South Latoya	20.093158
95	South Marychester	41.870000
96	South Michelleport	24.451613
97	South Phillip	28.571290
98	South Saramouth	36.160000
99	South Teresa	31.220455
100	Taylorhaven	42.263333
101	Valentineton	24.636364
102	Veronicaberg	32.828235
103	Victoriaport	27.780000
104	West Angela	25.990000
105	West Anthony	24.736667
106	West Christopherberg	24.421154
107	West Ericstad	22.347222
108	West Gabriel	20.346087
109	West Hannah	29.547619
110	West Heather	33.890000
111	West Heidi	23.133929
112	West Josephberg	21.720385
113	West Kimmouth	29.871500
114	West Patrickchester	28.233125
115	West Robert	25.123871
116	West Samuelburgh	21.767600
117	Williamsonville	31.875000
118	Williamsstad	24.362174
119	Williamsview	26.599000

120 rows × 2 columns

In [4]:
```
#Total Number of Rides Per City
total_rides = combined_data.groupby("city")["ride_id"].count()
total_rides = pd.DataFrame(total_rides).reset_index()
total_rides = total_rides.rename(columns = {'ride_id': 'total_rides'})
total_rides
```

Out[4]:

	city	total_rides
0	Amandaburgh	18
1	Barajasview	22
2	Barronchester	16
3	Bethanyland	18
4	Bradshawfurt	10
5	Brandonfort	19
6	Carriemouth	27
7	Christopherfurt	27
8	Colemanland	22
9	Davidfurt	17
10	Deanville	19
11	East Aaronbury	9
12	East Danielview	17
13	East Kaylahaven	29
14	East Kentstad	13
15	East Marymouth	27
16	Erikaland	12
17	Garzaport	3
18	Grahamburgh	25
19	Grayville	15
20	Harringtonfort	6
21	Huntermouth	24
22	Hurleymouth	28
23	Jerryton	25
24	Jessicaport	6
25	Johnton	21
26	Joneschester	25
27	Josephside	27
28	Justinberg	30
29	Karenberg	17
...	...	...

	city	total_rides
90	South Evanton	31
91	South Jack	19
92	South Jennifer	7
93	South Karenland	38
94	South Latoya	19
95	South Marychester	8
96	South Michelleport	31
97	South Phillip	31
98	South Saramouth	4
99	South Teresa	22
100	Taylorhaven	6
101	Valentineton	22
102	Veronicaberg	17
103	Victoriaport	14
104	West Angela	39
105	West Anthony	30
106	West Christopherberg	26
107	West Ericstad	18
108	West Gabriel	23
109	West Hannah	21
110	West Heather	9
111	West Heidi	28
112	West Josephberg	26
113	West Kimmouth	20
114	West Patrickchester	16
115	West Robert	31
116	West Samuelburgh	25
117	Williamsonville	14
118	Williamsstad	23
119	Williamsview	20

120 rows × 2 columns

In [5]:
```
#Total Drivers
total_drivers = combined_data[["city", "driver_count"]].drop_duplicates("city")
total_drivers
```

Out[5]:

	city	driver_count
0	Richardfort	38
28	Williamsstad	59
51	Port Angela	67
70	Rodneyfort	34
93	West Robert	39
124	West Anthony	70
154	West Angela	48
193	Martinezhaven	25
217	Karenberg	22
234	Barajasview	26
256	Robertport	12
276	Joneschester	39
301	Leahton	17
322	West Christopherberg	32
348	Johnton	27
369	Reynoldsfurt	67
388	Port David	7
410	New Kimberlyborough	33
440	Carriemouth	52
467	Rogerston	25
489	Jerryton	64
514	Loganberg	23
542	Simpsonburgh	21
566	Port Frank	23
599	South Latoya	10
618	West Samuelburgh	73
643	Grahamburgh	61
668	West Patrickchester	25
684	North Madeline	19
709	South Jack	46
...	...	...
2038	Barronchester	11
2054	Brandonfort	10

	city	driver_count
2073	East Danielview	22
2090	East Marymouth	5
2117	Mezachester	14
2134	Lewisland	4
2151	Josephside	25
2178	Davidfurt	23
2195	Nicolechester	19
2214	East Aaronbury	7
2223	North Richardhaven	1
2237	North Jeffrey	11
2250	South Jennifer	7
2257	West Heather	4
2266	Newtonview	1
2270	North Holly	8
2279	Michaelberg	6
2291	Taylorhaven	1
2297	Penaborough	6
2302	Harringtonfort	4
2308	Lake Jamie	4
2314	Lake Latoyabury	2
2325	North Jaime	1
2333	South Marychester	1
2341	Garzaport	7
2344	Bradshawfurt	7
2354	New Ryantown	2
2360	Randallchester	9
2365	Jessicaport	1
2371	South Saramouth	7

120 rows × 2 columns

In [6]:
```
#City Type
city_type = combined_data[["city", "type"]].drop_duplicates("city")
city_type
```

Out[6]:

	city	type
0	Richardfort	Urban
28	Williamsstad	Urban
51	Port Angela	Urban
70	Rodneyfort	Urban
93	West Robert	Urban
124	West Anthony	Urban
154	West Angela	Urban
193	Martinezhaven	Urban
217	Karenberg	Urban
234	Barajasview	Urban
256	Robertport	Urban
276	Joneschester	Urban
301	Leahton	Urban
322	West Christopherberg	Urban
348	Johnton	Urban
369	Reynoldsfurt	Urban
388	Port David	Urban
410	New Kimberlyborough	Urban
440	Carriemouth	Urban
467	Rogerston	Urban
489	Jerryton	Urban
514	Loganberg	Urban
542	Simpsonburgh	Urban
566	Port Frank	Urban
599	South Latoya	Urban
618	West Samuelburgh	Urban
643	Grahamburgh	Urban
668	West Patrickchester	Urban
684	North Madeline	Urban
709	South Jack	Urban
...	...	...
2038	Barronchester	Suburban
2054	Brandonfort	Suburban

	city	type
2073	East Danielview	Suburban
2090	East Marymouth	Suburban
2117	Mezachester	Suburban
2134	Lewisland	Suburban
2151	Josephside	Suburban
2178	Davidfurt	Suburban
2195	Nicolechester	Suburban
2214	East Aaronbury	Suburban
2223	North Richardhaven	Suburban
2237	North Jeffrey	Suburban
2250	South Jennifer	Rural
2257	West Heather	Rural
2266	Newtonview	Rural
2270	North Holly	Rural
2279	Michaelberg	Rural
2291	Taylorhaven	Rural
2297	Penaborough	Rural
2302	Harringtonfort	Rural
2308	Lake Jamie	Rural
2314	Lake Latoyabury	Rural
2325	North Jaime	Rural
2333	South Marychester	Rural
2341	Garzaport	Rural
2344	Bradshawfurt	Rural
2354	New Ryantown	Rural
2360	Randallchester	Rural
2365	Jessicaport	Rural
2371	South Saramouth	Rural

120 rows × 2 columns

In [7]:
```
# Merged analysis data frame

final_df = pd.merge(pd.merge(pd.merge(average_fare, total_rides, on="city"),
                             total_drivers, on="city"), city_type, on="city")
final_df
```

Out[7]:

	city	avg_fare	total_rides	driver_count	type
0	Amandaburgh	24.641667	18	12	Urban
1	Barajasview	25.332273	22	26	Urban
2	Barronchester	36.422500	16	11	Suburban
3	Bethanyland	32.956111	18	22	Suburban
4	Bradshawfurt	40.064000	10	7	Rural
5	Brandonfort	35.437368	19	10	Suburban
6	Carriemouth	28.314444	27	52	Urban
7	Christopherfurt	24.501852	27	41	Urban
8	Colemanland	30.894545	22	23	Suburban
9	Davidfurt	31.995882	17	23	Suburban
10	Deanville	25.842632	19	49	Urban
11	East Aaronbury	25.661111	9	7	Suburban
12	East Danielview	31.560588	17	22	Suburban
13	East Kaylahaven	23.757931	29	65	Urban
14	East Kentstad	29.823077	13	20	Suburban
15	East Marymouth	30.835185	27	5	Suburban
16	Erikaland	24.906667	12	37	Urban
17	Garzaport	24.123333	3	7	Rural
18	Grahamburgh	25.221200	25	61	Urban
19	Grayville	27.763333	15	2	Suburban
20	Harringtonfort	33.470000	6	4	Rural
21	Huntermouth	28.993750	24	37	Urban
22	Hurleymouth	25.891429	28	36	Urban
23	Jerryton	25.649200	25	64	Urban
24	Jessicaport	36.013333	6	1	Rural
25	Johnton	26.785714	21	27	Urban
26	Joneschester	22.289600	25	39	Urban
27	Josephside	32.858148	27	25	Suburban
28	Justinberg	23.694333	30	39	Urban
29	Karenberg	26.340000	17	22	Urban
...	...	...	...	...	...

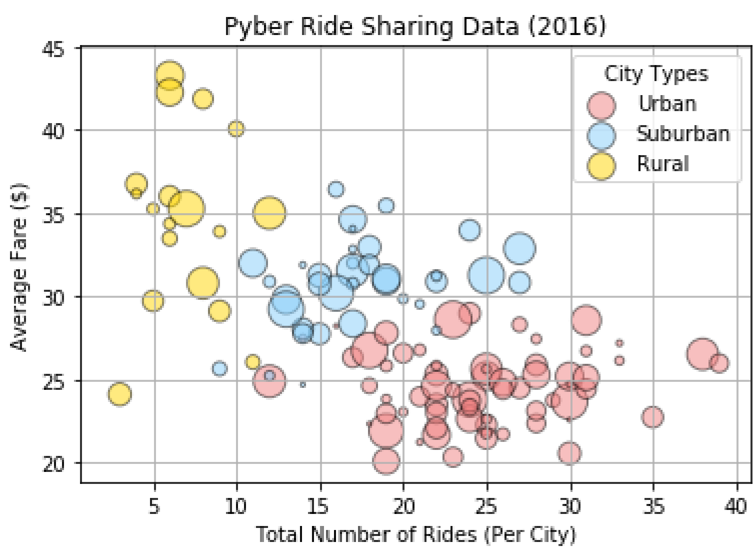
	city	avg_fare	total_rides	driver_count	type
90	South Evanton	26.726129	31	11	Urban
91	South Jack	22.965263	19	46	Urban
92	South Jennifer	35.264286	7	7	Rural
93	South Karenland	26.535526	38	4	Urban
94	South Latoya	20.093158	19	10	Urban
95	South Marychester	41.870000	8	1	Rural
96	South Michelleport	24.451613	31	72	Urban
97	South Phillip	28.571290	31	38	Urban
98	South Saramouth	36.160000	4	7	Rural
99	South Teresa	31.220455	22	21	Suburban
100	Taylorhaven	42.263333	6	1	Rural
101	Valentineton	24.636364	22	45	Urban
102	Veronicaberg	32.828235	17	20	Suburban
103	Victoriaport	27.780000	14	16	Suburban
104	West Angela	25.990000	39	48	Urban
105	West Anthony	24.736667	30	70	Urban
106	West Christopherberg	24.421154	26	32	Urban
107	West Ericstad	22.347222	18	25	Urban
108	West Gabriel	20.346087	23	57	Urban
109	West Hannah	29.547619	21	12	Suburban
110	West Heather	33.890000	9	4	Rural
111	West Heidi	23.133929	28	28	Urban
112	West Josephberg	21.720385	26	45	Urban
113	West Kimmouth	29.871500	20	4	Suburban
114	West Patrickchester	28.233125	16	25	Urban
115	West Robert	25.123871	31	39	Urban
116	West Samuelburgh	21.767600	25	73	Urban
117	Williamsonville	31.875000	14	2	Suburban
118	Williamsstad	24.362174	23	59	Urban
119	Williamsview	26.599000	20	46	Urban

120 rows × 5 columns

Bubble plot of ride sharing data

In [8]:
```
urban_group = final_df.loc[final_df['type'] == 'Urban']
suburban_group = final_df.loc[final_df['type'] == 'Suburban']
rural_group = final_df.loc[final_df['type'] == 'Rural']

#Bubble Plot
ax1 = urban_group.plot(kind='scatter',x='total_rides', y='avg_fare',
                       color='lightcoral', s=final_df['driver_count']*5, label =
                       alpha = 0.5, edgecolor = "black", linewidths = 1)
ax2 = suburban_group.plot(kind='scatter', x='total_rides', y='avg_fare',
                          color='lightskyblue',s=final_df['driver_count']*5, labe
                          alpha = 0.5, edgecolor = "black", linewidths = 1, ax=ax
ax3 = rural_group.plot(kind='scatter', x='total_rides', y='avg_fare',
                       color='gold', s=final_df['driver_count']*5, label = 'Rural
                       alpha = 0.5, edgecolor = "black", linewidths = 1, ax=ax1)
plt.title("Pyber Ride Sharing Data (2016)")
plt.xlabel("Total Number of Rides (Per City)")
plt.ylabel("Average Fare ($)")
plt.legend(title = 'City Types')
plt.grid(True)
plt.show()
```



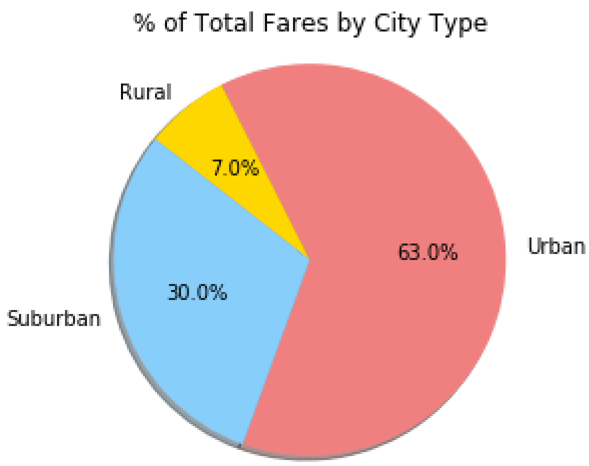
Total fares by city type (pie chart 1)

In [9]:
```
#Percent of Total Fares by City Type
total_fares = combined_data["fare"].sum()
urban_fare = combined_data.loc[combined_data["type"] == "Urban", "fare"].sum()
rural_fare = combined_data.loc[combined_data["type"] == "Rural", "fare"].sum()
suburban_fare = combined_data.loc[combined_data["type"] == "Suburban", "fare"].su
urban_fare_p = round(urban_fare/total_fares, 2) *100
rural_fare_p = round(rural_fare/total_fares, 2) *100
suburban_fare_p = round(suburban_fare/total_fares, 2) *100

# Pie chart
labels = 'Urban', 'Rural', 'Suburban'
sizes = [urban_fare_p, rural_fare_p, suburban_fare_p]
explode = (0.1, 0, 0)


fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["lightcoral", "gold", "lightskyblu
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title("% of Total Fares by City Type")

plt.show()
```



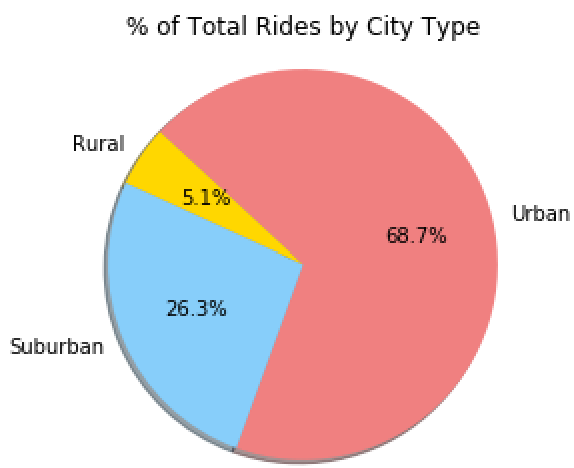
Total rides by city type (Chart 2)

In [10]:
```
#Percent of Total Rides by City Type
final_total_rides = len(combined_data)
urban_rides = round(combined_data.loc[combined_data["type"] == "Urban", "ride_id"
rural_rides = round(combined_data.loc[combined_data["type"] == "Rural", "ride_id"
suburban_rides = round(combined_data.loc[combined_data["type"] == "Suburban", "ri

# Pie chart
labels = 'Urban', 'Rural', 'Suburban'
sizes = [urban_rides, rural_rides, suburban_rides]
explode = (0.1, 0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["lightcoral", "gold", "lightskyblu
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title("% of Total Rides by City Type")

plt.show()
```

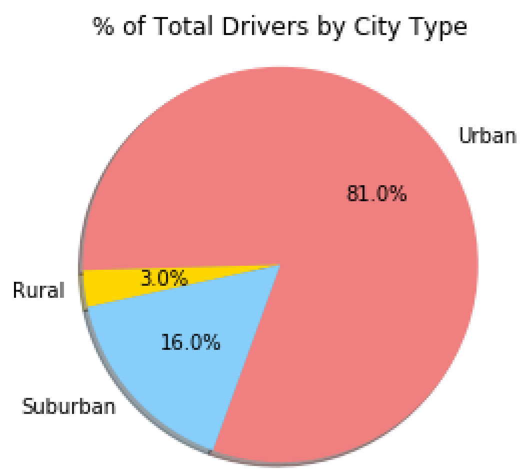


Total drivers by city type

In [11]:
```
#Percent of Total Drivers by City Type
final_total_drivers = final_df["driver_count"].sum()
urban_drivers = round(final_df.loc[final_df["type"] == "Urban", "driver_count"].s
rural_drivers = round(final_df.loc[final_df["type"] == "Rural", "driver_count"].s
suburban_drivers = round(final_df.loc[final_df["type"] == "Suburban", "driver_cou
# Pie chart
labels = 'Urban', 'Rural', 'Suburban'
sizes = [urban_drivers, rural_drivers, suburban_drivers]
explode = (0.1, 0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["lightcoral", "gold", "lightskyblu
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.title("% of Total Drivers by City Type")

plt.show()
```



In [ ]: