faithalone / Pyber_Starter1

Branch: master ▾ Pyber_Starter1 / Pyber_Starter1.ipynb Find file Copy path

faithalone Add files via upload 9f275d1 2 days ago

1 contributor

2808 lines (2807 sloc) | 210 KB

In [3]:
```
%matplotlib inline
# Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# File to Load (Remember to change these)
city_data_to_load = "data/city_data.csv"
ride_data_to_load = "data/ride_data.csv"

# Read the City and Ride Data
city_df = pd.read_csv(city_data_to_load)
ride_df = pd.read_csv(ride_data_to_load)
# Combine the data into a single dataset
pyber_df= pd.merge(ride_df, city_df, on="city")
# Display the data table for preview
pyber_df.head(10)
```

Out[3]:

	city	date	fare	ride_id	driver_count	type
0	Lake Jonathanshire	2018-01-14 10:14:22	13.83	5739410935873	5	Urban
1	Lake Jonathanshire	2018-04-07 20:51:11	31.25	4441251834598	5	Urban
2	Lake Jonathanshire	2018-03-09 23:45:55	19.89	2389495660448	5	Urban
3	Lake Jonathanshire	2018-04-07 18:09:21	24.28	7796805191168	5	Urban
4	Lake Jonathanshire	2018-01-02 14:14:50	13.89	424254840012	5	Urban
5	Lake Jonathanshire	2018-04-06 11:30:32	16.84	6164453571846	5	Urban
6	Lake Jonathanshire	2018-03-21 00:18:34	37.95	8353656732934	5	Urban
7	Lake Jonathanshire	2018-01-28 00:07:00	5.67	9756573174778	5	Urban
8	Lake Jonathanshire	2018-01-24 12:24:22	34.65	3319117904437	5	Urban
9	Lake Jonathanshire	2018-03-24 16:27:49	14.94	1670908453476	5	Urban

In [4]:
```
#Average fare ($) per City
Pyber_Average = pyber_df.groupby("city")["fare"].mean()
Pyber_Average= pd.DataFrame(Pyber_Average).reset_index()
Pyber_Average = Pyber_Average.rename(columns = {"fare":"Average fare"})
Pyber_Average
```

Out[4]:

	city	Average 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
...	...	...
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 [5]:
```
#Total Number of Rides Per City
Pyber_TR = pyber_df.groupby("city")["ride_id"].count()
Pyber_TR = pd.DataFrame(Pyber_TR).reset_index()
Pyber_TR = Pyber_TR.rename(columns = {'ride_id': 'total_rides'})
Pyber_TR
```

Out[5]:

	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
...	...	...
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 [6]:
```
#Total Numbers of Drivers Count per City
Pyber_DC = pyber_df[["city", "driver_count"]].drop_duplicates("city")
Pyber_DC
```

Out[6]:

	city	driver_count
0	Lake Jonathanshire	5
24	South Michelleport	72
55	Port Samanthamouth	57
80	Rodneyfort	34
103	South Jack	46
122	South Latoya	10
141	New Paulville	44
163	Simpsonburgh	21
187	South Karenland	4
225	North Jasmine	33
255	New Kimberlyborough	33
285	West Angela	48
324	Roberthaven	47
348	North Jason	6
383	Williamsview	46
403	Leahton	17
424	West Anthony	70
454	New Paulton	44
473	West Patrickchester	25
489	Deanville	49
508	West Josephberg	45

534	West Samuelburgh	73
559	West Heidi	28
587	Loganberg	23
615	Huntermouth	37
639	Grahamburgh	61
664	Port Frank	23
697	East Kaylahaven	65
726	West Robert	39
757	North Markport	22
...	...	...
2053	Port Shane	7
2072	North Timothy	7
2087	Veronicaberg	20
2104	Williamsonville	2
2118	Lewishaven	23
2130	Lake Ann	3
2142	Grayville	2
2157	Colemanland	23
2179	West Kimmouth	4
2199	Mezachester	14
2216	Davidfurt	23
2233	East Danielview	22
2250	Randallchester	9
2255	North Holly	8
2264	Michaelberg	6
2276	Lake Latoyabury	2
2287	Taylorhaven	1
2293	Garzaport	7
2296	New Ryantown	2
2302	Bradshawfurt	7
2312	South Marychester	1
2320	Jessicaport	1
2326	South Jennifer	7
2333	South Saramouth	7
2337	Lake Jamie	4
2343	Newtonview	1
2347	North Jaime	1
2355	Penaborough	6
2360	Harringtonfort	4
2366	West Heather	4

120 rows × 2 columns

In [7]:
```
#Total Type per City
Pyber_city = pyber_df[["city", "type"]].drop_duplicates("city")
Pyber_city
```

Out[7]:

	city	type

0	Lake Jonathanshire	Urban
24	South Michelleport	Urban
55	Port Samanthamouth	Urban
80	Rodneyfort	Urban
103	South Jack	Urban
122	South Latoya	Urban
141	New Paulville	Urban
163	Simpsonburgh	Urban
187	South Karenland	Urban
225	North Jasmine	Urban
255	New Kimberlyborough	Urban
285	West Angela	Urban
324	Roberthaven	Urban
348	North Jason	Urban
383	Williamsview	Urban
403	Leahton	Urban
424	West Anthony	Urban
454	New Paulton	Urban
473	West Patrickchester	Urban
489	Deanville	Urban
508	West Josephberg	Urban
534	West Samuelburgh	Urban
559	West Heidi	Urban
587	Loganberg	Urban
615	Huntermouth	Urban
639	Grahamburgh	Urban
664	Port Frank	Urban
697	East Kaylahaven	Urban
726	West Robert	Urban
757	North Markport	Urban
...	...	...
2053	Port Shane	Suburban
2072	North Timothy	Suburban
2087	Veronicaberg	Suburban
2104	Williamsonville	Suburban
2118	Lewishaven	Suburban
2130	Lake Ann	Suburban
2142	Grayville	Suburban
2157	Colemanland	Suburban
2179	West Kimmouth	Suburban
2199	Mezachester	Suburban
2216	Davidfurt	Suburban
2233	East Danielview	Suburban
2250	Randallchester	Rural
2255	North Holly	Rural
2264	Michaelberg	Rural

2276	Lake Latoyabury	Rural
2287	Taylorhaven	Rural
2293	Garzaport	Rural
2296	New Ryantown	Rural
2302	Bradshawfurt	Rural
2312	South Marychester	Rural
2320	Jessicaport	Rural
2326	South Jennifer	Rural
2333	South Saramouth	Rural
2337	Lake Jamie	Rural
2343	Newtonview	Rural
2347	North Jaime	Rural
2355	Penaborough	Rural
2360	Harringtonfort	Rural
2366	West Heather	Rural

120 rows × 2 columns

In [8]:
```
# Merging the Average, Total rides, total drivers and types by the cities
Pyber_City_Merge_df = pd.merge(pd.merge(pd.merge(Pyber_Average, Pyber_TR, on="city"),
                                        Pyber_DC, on="city"), Pyber_city, on="city")
Pyber_City_Merge_df
```

Out[8]:

	city	Average 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

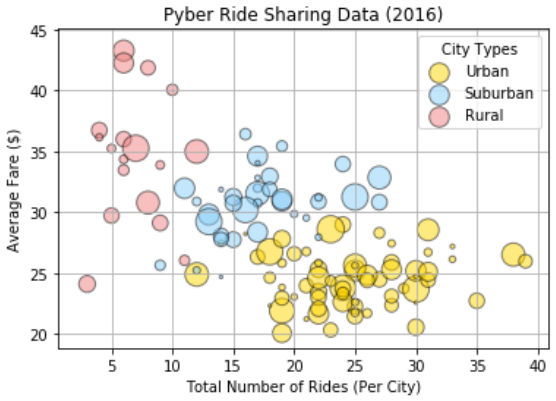
25	Johnton	26.785714	21	27	Urban
26	Joneschester	22.289600	25	39	Urban
27	Josephside	32.858148	27	25	Suburban
28	Justinberg	23.694333	30	39	Urban
29	Karenberg	26.340000	17	22	Urban
...	...	...	...	...	...
90	South Evanton	26.726129	31	11	Urban
91	South Jack	22.965263	19	46	Urban
92	South Jennifer	35.264286	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

In [9]:
```
# Obtain the x and y coordinates for each of the three city types
Pyber_Urban = Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Urban"]
Pyber_Suburban = Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Suburban"]
Pyber_Rural = Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Rural"]

#Bubble Plot for each city types
axis1 = Pyber_Urban.plot(kind="scatter",x="total_rides", y="Average fare",
                         color="gold", s=Pyber_City_Merge_df["driver_count"]*5, label = "Urban",
                         alpha = 0.5, edgecolor = "black", linewidths = 1)
axis2 = Pyber_Suburban.plot(kind="scatter",x="total_rides", y="Average fare",
                         color="Lightskyblue", s=Pyber_City_Merge_df["driver_count"]*5, label = "Suburban",
                         alpha = 0.5, edgecolor = "black", linewidths = 1, ax=axis1)
```

```
axis3 = Pyber_Rural.plot(kind="scatter",x="total_rides", y="Average fare",
                         color="Lightcoral", s=Pyber_City_Merge_df["driver_count"]*5, label = "Rural",
                         alpha = 0.5, edgecolor = "black", linewidths = 1, ax=axis1)
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()
```



In [10]:
```
# Total Sum of Urban, Suburban and Rural fares
Total_Sum_USR= pyber_df["fare"].sum()
Total_Sum_USR
```

Out[10]: 63538.64

In [11]:
```
#Total fares for Urban (U) Cities alone
Total_Sum_Urban = pyber_df.loc[pyber_df["type"] == "Urban","fare"].sum()
Total_Sum_Urban
```

Out[11]: 39854.38

In [12]:
```
#Total fares for Suburban (S) Cities alone
Total_Sum_Suburban = pyber_df.loc[pyber_df["type"] == "Suburban","fare"].sum()
Total_Sum_Suburban
```

Out[12]: 19356.33

In [13]:
```
#Total fares for Rural (R) Cities alone
Total_Sum_Rural = pyber_df.loc[pyber_df["type"] == "Rural","fare"].sum()
Total_Sum_Rural
```

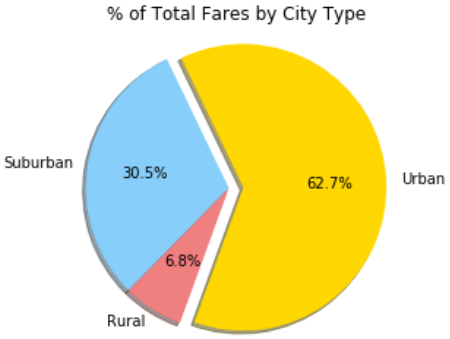
Out[13]: 4327.929999999999

In [14]:
```
#Percentages for Urban,Suburban and Rural Cities
Urban_perc = (Total_Sum_Urban/Total_Sum_USR)*100
Urban_perc = round(Urban_perc,1)
Suburban_perc = (Total_Sum_Suburban/Total_Sum_USR)*100
Suburban_perc = round(Suburban_perc,1)
Rural_perc = (Total_Sum_Rural/Total_Sum_USR)*100
Rural_perc = round(Rural_perc, 1)
print(Urban_perc)
print(Suburban_perc)
print(Rural_perc )
```

```
62.7
30.5
6.8
```

In [15]:
```
# Pie chart
labels = "Urban","Suburban","Rural"
sizes = [Urban_perc, Suburban_perc, Rural_perc]
explode = (0.1, 0, 0)


fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["gold", "lightskyblue","lightcoral"])
ax1.axis("equal")
plt.title("% of Total Fares by City Type")
plt.show()
```

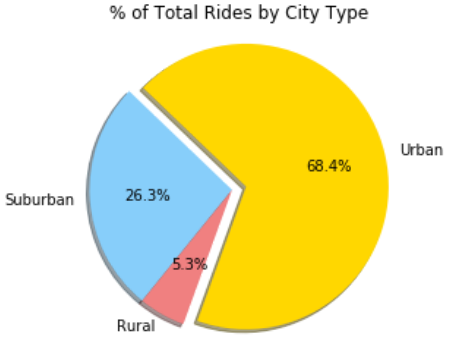


In [17]:
```
Total_Rides_USR = len(pyber_df)
Total_Rides_Urban = (pyber_df.loc[pyber_df["type"] == "Urban", "ride_id"].count()/Total_Rides_USR) * 100
Total_Rides_Urban = round(Total_Rides_Urban,1)
Total_Rides_Suburban = (pyber_df.loc[pyber_df["type"] == "Suburban", "ride_id"].count()/Total_Rides_USR) *
 100
Total_Rides_Suburban = round(Total_Rides_Suburban,1)
Total_Rides_Rural = (pyber_df.loc[pyber_df["type"] == "Rural", "ride_id"].count()/Total_Rides_USR) * 100
Total_Rides_Rural = round(Total_Rides_Rural,1)
print(Total_Rides_USR)
print(Total_Rides_Urban)
print(Total_Rides_Suburban)
print(Total_Rides_Rural)
```

```
2375
68.4
26.3
5.3
```

In [18]:
```
# Pie chart percentage for Total Rides by City Types
labels = "Urban", "Suburban", "Rural"
sizes = [Total_Rides_Urban, Total_Rides_Suburban, Total_Rides_Rural]
explode = (0.1, 0, 0)
fig1, axis1 = plt.subplots()
axis1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["gold", "lightskyblue","lightcoral"])
axis1.axis("equal")
plt.title("% of Total Rides by City Type")

plt.show()
```



In [20]:
```
#Total Drivers by City Type
Total_Drivers_USR = Pyber_City_Merge_df["driver_count"].sum()
Total_Drivers_Urban = (Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Urban", "driver_count"].sum
()/Total_Drivers_USR)* 100
Total_Drivers_Urban = round(Total_Drivers_Urban,1)
Total_Drivers_Suburban =(Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Suburban", "driver_count"
].sum()/Total_Drivers_USR)* 100
Total_Drivers_Suburban = round(Total_Drivers_Suburban,1)
Total_Drivers_Rural = (Pyber_City_Merge_df.loc[Pyber_City_Merge_df["type"] == "Rural", "driver_count"].sum
()/Total_Drivers_USR)* 100
Total_Drivers_Rural =round(Total_Drivers_Rural,1)
print(Total_Drivers_USR)
print(Total_Drivers_Urban)
print(Total_Drivers_Suburban)
print(Total_Drivers_Rural)
```

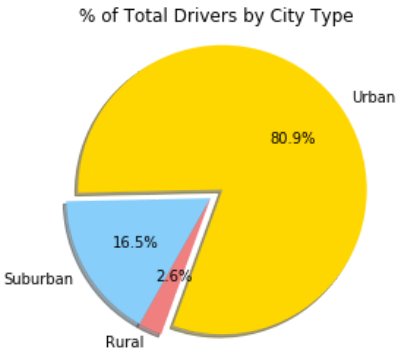
```
2973
```

```
80.9
16.5
2.6
```

In [21]:
```
# Pie chart of Percentages for Total Drivers by City Type
labels = "Urban", "Suburban", "Rural"
sizes = [Total_Drivers_Urban, Total_Drivers_Suburban, Total_Drivers_Rural]
explode = (0.1, 0, 0)

fig1, axis1 = plt.subplots()
axis1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=250, colors = ["gold", "lightskyblue", "lightcoral"])
axis1.axis("equal")
plt.title("% of Total Drivers by City Type")

plt.show()
```



In [ ]:
```

```