

5.3.1 Create DataFrames for Each Type of City

After working late last night, you get into the office bright and early and check your email. You have an email from Omar!

"Thanks for committing your code. I had a chance to take a look and I have to say, nice work! It looks like you are ready to dig into that scatter plot we discussed. I have some time this afternoon if you'd like to go over it!"

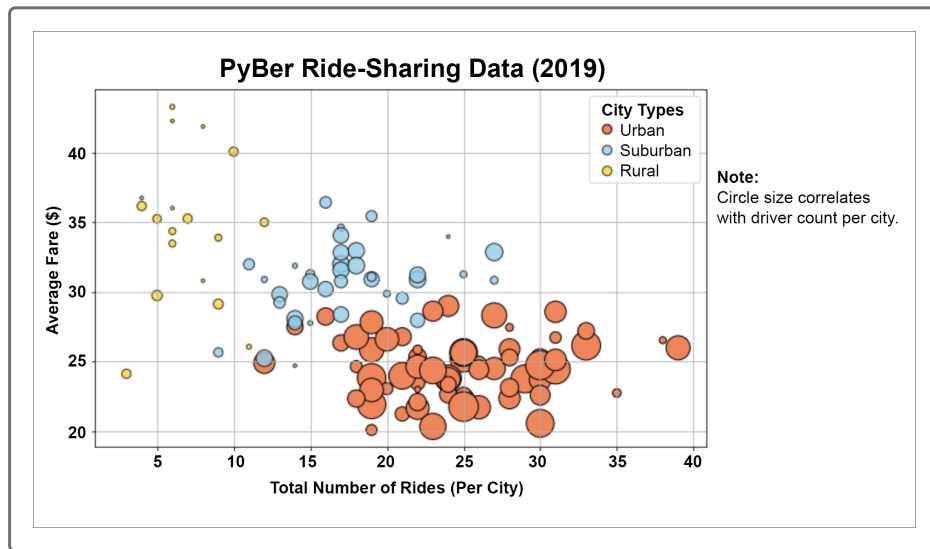
Energized and encouraged, you set up your workspace for the day and dig back in. Bubble charts, here we come!

Omar has asked you to create a bubble chart that showcases the average fare versus the total number of rides with bubble size based on the average number of drivers for each city type: urban, suburban, and rural.

NOTE

The terms **scatter plot** and **scatter chart**, as well as **bubble chart** and **bubble plot**, are often used interchangeably in the data visualization field.

When you're done, the bubble chart should look similar to this:



For the bubble chart, we will need to plot the following:

- The average fare for each type of city on the y-axis
- The total number of rides for each type of city on the x-axis
- Make the size of each marker, or bubble, correlate to the average number of drivers for each type of city

Here are the steps to create a bubble chart:

1. To get the average fare, we can use the Pandas `mean()` method for each city in the "fare" column.
2. To get the total number of rides for each city, we can use the Pandas `count()` method for each city in the "ride_id" column.
3. To make the bubble size correlate to the number of drivers for each city, we can get the average `driver_count` for each city.

Completing the steps will be more efficient if we create separate DataFrames for each type and then create Data Series for each step.

Let's create three separate DataFrames, one for each type of city.

Each new DataFrame can be created by filtering the `pyber_data_df` DataFrame where the city type is equal to Urban, Suburban, or Rural.

Add the following code in a new cell to create the Urban cities DataFrame:

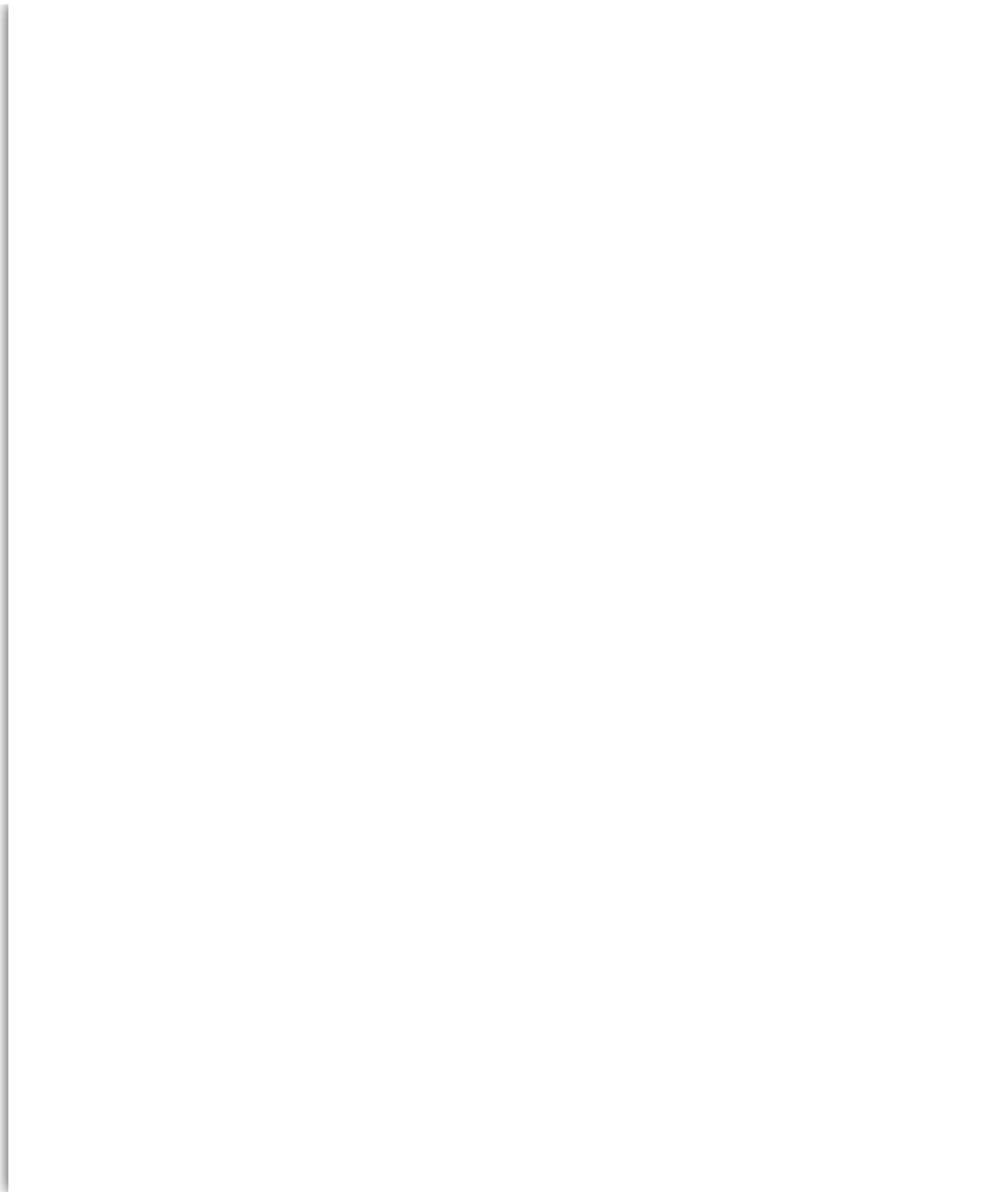
```
# Create the Urban city DataFrame.
urban_cities_df = pyber_data_df[pyber_data_df["type"] == "Urban"]
urban_cities_df.head()
```

When you run this cell, you'll get a DataFrame that contains only the cities with the type equal to Urban, as you can see in this snapshot of the first five rows of the output:

	city	date	fare	ride_id	driver_count	type
0	Lake Jonathanshire	2019-01-14 10:14:22	13.83	5739410935873	5	Urban
1	South Michelleport	2019-03-04 18:24:09	30.24	2343912425577	72	Urban
2	Port Samanthamouth	2019-02-24 04:29:00	33.44	2005065760003	57	Urban
3	Rodneyfort	2019-02-10 23:22:03	23.44	5149245426178	34	Urban
4	South Jack	2019-03-06 04:28:35	34.58	3908451377344	46	Urban

Now add the following code in a new cell and run the cell to create the `suburban_cities_df` and the `rural_cities_df` DataFrames:

```
# Create the Suburban and Rural city DataFrames.
suburban_cities_df = pyber_data_df[pyber_data_df["type"] == "Suburban"]
rural_cities_df = pyber_data_df[pyber_data_df["type"] == "Rural"]
```



Great job on creating the DataFrames for each city type. Next, we'll get the total number of rides from each city type.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.