Pyber Challenge

In [1]:
```
! pip install -U notebook-as-pdf
```

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Collecting pyppeteer
  Downloading pyppeteer-1.0.2-py3-none-any.whl (83 kB)
Requirement already satisfied, skipping upgrade: nbconvert in c:\users\18482\anaconda3\lib\site-packa
ges (from notebook-as-pdf) (5.6.1)
Collecting PyPDF2
  Downloading PyPDF2-1.27.12-py3-none-any.whl (80 kB)
Requirement already satisfied, skipping upgrade: websockets<11.0,>=10.0 in c:\users\18482\anaconda3\l
ib\site-packages (from pyppeteer->notebook-as-pdf) (10.1)
Collecting appdirs<2.0.0,>=1.4.3
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Requirement already satisfied, skipping upgrade: tqdm<5.0.0,>=4.42.1 in c:\users\18482\anaconda3\lib
\site-packages (from pyppeteer->notebook-as-pdf) (4.48.2)
Requirement already satisfied, skipping upgrade: urllib3<2.0.0,>=1.25.8 in c:\users\18482\anaconda3\l
ib\site-packages (from pyppeteer->notebook-as-pdf) (1.25.10)
Requirement already satisfied, skipping upgrade: importlib-metadata>=1.4 in c:\users\18482\anaconda3
\lib\site-packages (from pyppeteer->notebook-as-pdf) (1.7.0)
Collecting pyee<9.0.0,>=8.1.0
  Downloading pyee-8.2.2-py2.py3-none-any.whl (12 kB)
Collecting certifi>=2021
  Downloading certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Requirement already satisfied, skipping upgrade: jupyter-core in c:\users\18482\anaconda3\lib\site-pa
ckages (from nbconvert->notebook-as-pdf) (4.6.3)
Requirement already satisfied, skipping upgrade: entrypoints>=0.2.2 in c:\users\18482\anaconda3\lib\s
ite-packages (from nbconvert->notebook-as-pdf) (0.3)
Requirement already satisfied, skipping upgrade: defusedxml in c:\users\18482\anaconda3\lib\site-pack
ages (from nbconvert->notebook-as-pdf) (0.6.0)
Requirement already satisfied, skipping upgrade: nbformat>=4.4 in c:\users\18482\anaconda3\lib\site-p
ackages (from nbconvert->notebook-as-pdf) (5.0.7)
Requirement already satisfied, skipping upgrade: traitlets>=4.2 in c:\users\18482\anaconda3\lib\site-
packages (from nbconvert->notebook-as-pdf) (4.3.3)
Requirement already satisfied, skipping upgrade: pygments in c:\users\18482\anaconda3\lib\site-packag
es (from nbconvert->notebook-as-pdf) (2.6.1)
Requirement already satisfied, skipping upgrade: pandocfilters>=1.4.1 in c:\users\18482\anaconda3\lib
\site-packages (from nbconvert->notebook-as-pdf) (1.4.2)
Requirement already satisfied, skipping upgrade: mistune<2,>=0.8.1 in c:\users\18482\anaconda3\lib\si
te-packages (from nbconvert->notebook-as-pdf) (0.8.4)
Requirement already satisfied, skipping upgrade: testpath in c:\users\18482\anaconda3\lib\site-packag
es (from nbconvert->notebook-as-pdf) (0.4.4)
Requirement already satisfied, skipping upgrade: bleach in c:\users\18482\anaconda3\lib\site-packages
(from nbconvert->notebook-as-pdf) (3.1.5)
Requirement already satisfied, skipping upgrade: jinja2>=2.4 in c:\users\18482\anaconda3\lib\site-pac
kages (from nbconvert->notebook-as-pdf) (2.11.2)
Requirement already satisfied, skipping upgrade: zipp>=0.5 in c:\users\18482\anaconda3\lib\site-packa
ges (from importlib-metadata>=1.4->pyppeteer->notebook-as-pdf) (3.1.0)
Requirement already satisfied, skipping upgrade: pywin32>=1.0; sys_platform == "win32" in c:\users\18
482\anaconda3\lib\site-packages (from jupyter-core->nbconvert->notebook-as-pdf) (227)
Requirement already satisfied, skipping upgrade: jsonschema!=2.5.0,>=2.4 in c:\users\18482\anaconda3
\lib\site-packages (from nbformat>=4.4->nbconvert->notebook-as-pdf) (3.2.0)
Requirement already satisfied, skipping upgrade: ipython-genutils in c:\users\18482\anaconda3\lib\sit
e-packages (from nbformat>=4.4->nbconvert->notebook-as-pdf) (0.2.0)
Requirement already satisfied, skipping upgrade: six in c:\users\18482\anaconda3\lib\site-packages (f
rom traitlets>=4.2->nbconvert->notebook-as-pdf) (1.15.0)
Requirement already satisfied, skipping upgrade: decorator in c:\users\18482\anaconda3\lib\site-packa
ges (from traitlets>=4.2->nbconvert->notebook-as-pdf) (4.4.2)
Requirement already satisfied, skipping upgrade: webencodings in c:\users\18482\anaconda3\lib\site-pa
ckages (from bleach->nbconvert->notebook-as-pdf) (0.5.1)
Requirement already satisfied, skipping upgrade: packaging in c:\users\18482\anaconda3\lib\site-packa
ges (from bleach->nbconvert->notebook-as-pdf) (20.4)
Requirement already satisfied, skipping upgrade: MarkupSafe>=0.23 in c:\users\18482\anaconda3\lib\sit
e-packages (from jinja2>=2.4->nbconvert->notebook-as-pdf) (1.1.1)
Requirement already satisfied, skipping upgrade: pyrsistent>=0.14.0 in c:\users\18482\anaconda3\lib\s
ite-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (0.16.0)
Requirement already satisfied, skipping upgrade: setuptools in c:\users\18482\anaconda3\lib\site-pack
ages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (49.6.0.post20200814)
Requirement already satisfied, skipping upgrade: attrs>=17.4.0 in c:\users\18482\anaconda3\lib\site-p
ackages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->notebook-as-pdf) (20.1.0)
Requirement already satisfied, skipping upgrade: pyparsing>=2.0.2 in c:\users\18482\anaconda3\lib\sit
e-packages (from packaging->bleach->nbconvert->notebook-as-pdf) (2.4.7)
Installing collected packages: appdirs, pyee, certifi, pyppeteer, PyPDF2, notebook-as-pdf
  Attempting uninstall: certifi
    Found existing installation: certifi 2020.6.20
    Uninstalling certifi-2020.6.20:
      Successfully uninstalled certifi-2020.6.20
Successfully installed PyPDF2-1.27.12 appdirs-1.4.4 certifi-2021.10.8 notebook-as-pdf-0.5.0 pyee-8.2.
2 pyppeteer-1.0.2
```

In [2]:
```
! pyppeteer-install
```

```
[INFO] Starting Chromium download.

  0%|          | 0.00/137M [00:00<?, ?b/s]
  1%|1         | 1.86M/137M [00:00<00:07, 18.5Mb/s]
  6%|5         | 7.88M/137M [00:00<00:05, 23.4Mb/s]
 12%|#1        | 16.0M/137M [00:00<00:04, 29.7Mb/s]
 17%|#6        | 23.2M/137M [00:00<00:03, 36.0Mb/s]
 22%|##1       | 30.0M/137M [00:00<00:02, 41.8Mb/s]
 27%|##6       | 36.6M/137M [00:00<00:02, 47.0Mb/s]
 32%|###2      | 43.8M/137M [00:00<00:01, 51.6Mb/s]
 37%|###6      | 50.3M/137M [00:00<00:01, 53.4Mb/s]
 42%|####2     | 58.1M/137M [00:00<00:01, 58.6Mb/s]
 48%|####8     | 66.3M/137M [00:01<00:01, 64.1Mb/s]
 54%|#####4    | 74.4M/137M [00:01<00:00, 68.1Mb/s]
 60%|#####9    | 81.6M/137M [00:01<00:00, 67.8Mb/s]
 65%|######4   | 88.9M/137M [00:01<00:00, 68.5Mb/s]
 71%|#######   | 96.8M/137M [00:01<00:00, 71.0Mb/s]
 77%|#######6  | 105M/137M [00:01<00:00, 73.4Mb/s]
 82%|########2 | 113M/137M [00:01<00:00, 72.0Mb/s]
 89%|########9 | 122M/137M [00:01<00:00, 77.7Mb/s]
 96%|#########5| 131M/137M [00:01<00:00, 80.2Mb/s]
100%|##########| 137M/137M [00:01<00:00, 70.7Mb/s]
[INFO] Beginning extraction
[INFO] Chromium extracted to: C:\Users\18482\AppData\Local\pyppeteer\pyppeteer\local-chromium\588429
```

Loading and Reading CSV files

In [1]:
```python
# Add Matplotlib inline magic command
%matplotlib inline
# Dependencies and Setup
import matplotlib.pyplot as plt
import pandas as pd

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

# Read the City and Ride Data
city_data_df = pd.read_csv(city_data_to_load)
ride_data_df = pd.read_csv(ride_data_to_load)
```

Merge the DataFrames

In [2]:
```python
# Combine the data into a single dataset
pyber_data_df = pd.merge(ride_data_df, city_data_df, how="left", on=["city", "city"])

# Display the data table for preview
pyber_data_df.head()
```

Out[2]:

	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

Deliverable 1: Get a Summary DataFrame

In [3]:
```python
#  1. Get the total rides for each city type
urban_cities_df = pyber_data_df.groupby("type").count()["ride_id"]
urban_cities_df.head()
# Create the Urban city DataFrame.
urban_cities_df
```

Out[3]:
```
type
Rural        125
Suburban     625
Urban       1625
Name: ride_id, dtype: int64
```

In [4]:
```python
# 2. Get the total drivers for each city type
urban_cities_drivers_df = city_data_df.groupby("type").sum()["driver_count"]
urban_cities_drivers_df.head()
```

Out[4]:
```
type
Rural         78
Suburban     490
Urban       2405
Name: driver_count, dtype: int64
```

In [5]:
```python
#  3. Get the total amount of fares for each city type
pyber_fares_df = pyber_data_df.groupby(["type"]).sum()["fare"]
pyber_fares_df
```

Out[5]:
```
type
Rural        4327.93
Suburban    19356.33
Urban       39854.38
Name: fare, dtype: float64
```

In [6]:
```python
#  4. Get the average fare per ride for each city type.
pyber_average_fare = pyber_fares_df / urban_cities_df
pyber_average_fare
```

Out[6]:
```
type
Rural       34.623440
Suburban    30.970128
Urban       24.525772
dtype: float64
```

In [7]:
```python
# 5. Get the average fare per driver for each city type.
pyber_average_ride = pyber_fares_df / urban_cities_drivers_df
pyber_average_ride
```

Out[7]:
```
type
Rural       55.486282
Suburban    39.502714
Urban       16.571468
dtype: float64
```

In [8]:
```python
#  6. Create a PyBer summary DataFrame.
pyber_summary_df = pd.DataFrame({
    "Total Rides": urban_cities_df,
    "Total Drivers": urban_cities_drivers_df ,
    "Total Fares": pyber_fares_df,
    "Average Fare": pyber_average_fare,
    "Average Fare Per Driver": pyber_average_ride
})
pyber_summary_df
```

Out[8]:

	Total Rides	Total Drivers	Total Fares	Average Fare	Average Fare Per Driver
type					
Rural	125	78	4327.93	34.623440	55.486282
Suburban	625	490	19356.33	30.970128	39.502714
Urban	1625	2405	39854.38	24.525772	16.571468

In [9]:
```python
#  7. Cleaning up the DataFrame. Delete the index name
pyber_summary_df.index.name = None
pyber_summary_df
```

Out[9]:

	Total Rides	Total Drivers	Total Fares	Average Fare	Average Fare Per Driver
Rural	125	78	4327.93	34.623440	55.486282
Suburban	625	490	19356.33	30.970128	39.502714
Urban	1625	2405	39854.38	24.525772	16.571468

In [10]:
```python
pyber_summary_df["Total Fares"] = pyber_summary_df["Total Fares"].map("${:,.2f}".format)
pyber_summary_df["Average Fare"] = pyber_summary_df["Average Fare"].map("${:,.2f}".format)
pyber_summary_df["Average Fare Per Driver"] = pyber_summary_df["Average Fare Per Driver"].map("${:,.2f}
".format)
pyber_summary_df
```

Out[10]:

	Total Rides	Total Drivers	Total Fares	Average Fare	Average Fare Per Driver
Rural	125	78	$4,327.93	$34.62	$55.49
Suburban	625	490	$19,356.33	$30.97	$39.50
Urban	1625	2405	$39,854.38	$24.53	$16.57

Deliverable 2. Create a multiple line plot that shows the total weekly of the fares for each type of city.

In [11]:
```python
# 1. Read the merged DataFrame
pyber_data_df.head()
```

Out[11]:

	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

In [12]:
```python
# 2. Using groupby() to create a new DataFrame showing the sum of the fares
Pyber_Fare_type_date = pyber_data_df.groupby(["type","date"]).sum()["fare"]

#for each date where the indices are the city type and date.
Pyber_Fare_type_date
```

Out[12]:
```
type   date
Rural  2019-01-01 09:45:36    43.69
       2019-01-02 11:18:32    52.12
       2019-01-03 19:51:01    19.90
       2019-01-04 03:31:26    24.88
       2019-01-06 07:38:40    47.33
                              ...
Urban  2019-05-08 04:20:00    21.99
       2019-05-08 04:39:49    18.45
       2019-05-08 07:29:01    18.55
       2019-05-08 11:38:35    19.77
       2019-05-08 13:10:18    18.04
Name: fare, Length: 2375, dtype: float64
```

In [13]:
```python
# 3. Reset the index on the DataFrame you created in #1. This is needed to use the 'pivot()' function.
# df = df.reset_index()
Pyber_Fare_type_date = Pyber_Fare_type_date.reset_index()
Pyber_Fare_type_date
```

Out[13]:

	type	date	fare
0	Rural	2019-01-01 09:45:36	43.69
1	Rural	2019-01-02 11:18:32	52.12
2	Rural	2019-01-03 19:51:01	19.90
3	Rural	2019-01-04 03:31:26	24.88
4	Rural	2019-01-06 07:38:40	47.33
...	...	...	...
2370	Urban	2019-05-08 04:20:00	21.99
2371	Urban	2019-05-08 04:39:49	18.45
2372	Urban	2019-05-08 07:29:01	18.55
2373	Urban	2019-05-08 11:38:35	19.77
2374	Urban	2019-05-08 13:10:18	18.04

2375 rows × 3 columns

In [14]:
```python
# 4. Create a pivot table with the 'date' as the index, the columns ='type', and values='fare'
# to get the total fares for each type of city by the date.
Pyber_Fare_type_date_pivot = Pyber_Fare_type_date.pivot(index="date", columns = "type", values = "fare"
)
Pyber_Fare_type_date_pivot
```

Out[14]:

type	Rural	Suburban	Urban
date			
2019-01-01 00:08:16	NaN	NaN	37.91
2019-01-01 00:46:46	NaN	47.74	NaN
2019-01-01 02:07:24	NaN	24.07	NaN
2019-01-01 03:46:50	NaN	NaN	7.57
2019-01-01 05:23:21	NaN	NaN	10.75
...	...	...	...
2019-05-08 04:20:00	NaN	NaN	21.99
2019-05-08 04:39:49	NaN	NaN	18.45
2019-05-08 07:29:01	NaN	NaN	18.55
2019-05-08 11:38:35	NaN	NaN	19.77
2019-05-08 13:10:18	NaN	NaN	18.04

2375 rows × 3 columns

In [15]:
```python
# 5. Create a new DataFrame from the pivot table DataFrame using loc on the given dates, '2019-01-0
1':'2019-04-29'.
PFtdp_loc = Pyber_Fare_type_date_pivot.loc[(Pyber_Fare_type_date_pivot.index >= "2019-01-01") &
(Pyber_Fare_type_date_pivot.index <= "2019-04-29")]
PFtdp_loc
```

Out[15]:

type	Rural	Suburban	Urban
date			
2019-01-01 00:08:16	NaN	NaN	37.91
2019-01-01 00:46:46	NaN	47.74	NaN
2019-01-01 02:07:24	NaN	24.07	NaN
2019-01-01 03:46:50	NaN	NaN	7.57
2019-01-01 05:23:21	NaN	NaN	10.75
...	...	...	...
2019-04-28 14:28:36	NaN	NaN	11.46
2019-04-28 16:29:16	NaN	NaN	36.42
2019-04-28 17:26:52	NaN	NaN	31.43
2019-04-28 17:38:09	NaN	34.87	NaN
2019-04-28 19:35:03	NaN	16.96	NaN

2196 rows × 3 columns

In [16]:
```python
# 6. Set the "date" index to datetime datatype. This is necessary to use the resample() method in Step
 8.
# df.index = pd.to_datetime(df.index)
PFtdp_loc.index = pd.to_datetime(PFtdp_loc.index)
PFtdp_loc
```

Out[16]:

type	Rural	Suburban	Urban
date			
2019-01-01 00:08:16	NaN	NaN	37.91
2019-01-01 00:46:46	NaN	47.74	NaN
2019-01-01 02:07:24	NaN	24.07	NaN
2019-01-01 03:46:50	NaN	NaN	7.57
2019-01-01 05:23:21	NaN	NaN	10.75
...	...	...	...
2019-04-28 14:28:36	NaN	NaN	11.46
2019-04-28 16:29:16	NaN	NaN	36.42
2019-04-28 17:26:52	NaN	NaN	31.43
2019-04-28 17:38:09	NaN	34.87	NaN
2019-04-28 19:35:03	NaN	16.96	NaN

2196 rows × 3 columns

In [17]:
```python
# 7. Check that the datatype for the index is datetime using df.info()
PFtdp_loc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2196 entries, 2019-01-01 00:08:16 to 2019-04-28 19:35:03
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Rural     114 non-null    float64
 1   Suburban  573 non-null    float64
 2   Urban     1509 non-null   float64
dtypes: float64(3)
memory usage: 68.6 KB
```

In [18]:
```python
# 8. Create a new DataFrame using the "resample()" function by week 'W' and get the sum of the fares fo
r each week.
PFtdp_loc = PFtdp_loc.resample("W").sum()
PFtdp_loc
```

Out[18]:

type	Rural	Suburban	Urban
date			
2019-01-06	187.92	721.60	1661.68
2019-01-13	67.65	1105.13	2050.43
2019-01-20	306.00	1218.20	1939.02
2019-01-27	179.69	1203.28	2129.51
2019-02-03	333.08	1042.79	2086.94
2019-02-10	115.80	974.34	2162.64
2019-02-17	95.82	1045.50	2235.07
2019-02-24	419.06	1412.74	2466.29
2019-03-03	175.14	858.46	2218.20
2019-03-10	303.94	925.27	2470.93
2019-03-17	163.39	906.20	2044.42
2019-03-24	189.76	1122.20	2368.37
2019-03-31	199.42	1045.06	1942.77
2019-04-07	501.24	1010.73	2356.70
2019-04-14	269.79	784.82	2390.72
2019-04-21	214.14	1149.27	2303.80
2019-04-28	191.85	1357.75	2238.29

In [27]:
```python
# 8. Using the object-oriented interface method, plot the resample DataFrame using the df.plot() functi
on.

# Import the style from Matplotlib.
from matplotlib import style
# Use the graph style fivethirtyeight.
style.use('fivethirtyeight')

fig, ax = plt.subplots(figsize=(15,6))
PFtdp_loc.plot(kind="line", ax=ax, color=["moccasin", "pink", "palegreen"])
ax.set_title("Total Fare by City Type")
ax.set_ylabel("Fare ($USD)")
ax.set_xlabel(" ")
lgnd = plt.legend(fontsize="11", mode="Expanded", loc=(0.8115,0.581), title="City Types")

# Save Figure
plt.savefig("PyBer_fare_summary.png")

plt.show()
```