Pyber Challenge

4.3 Loading and Reading CSV files

In [1]:
```
# 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 = "Resources/city_data.csv"
ride_data_to_load = "Resources/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]:
```
# Combine the data into a single dataset
pyber_data_df = pd.merge(ride_data_df, city_data_df, how="left", on=["city",

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

Out[2]:

	city	date	fare	ride_id	driver_count	type
0	Lake Jonathanshire	1/14/19 10:14	13.83	5.739410e+12	5	Urban
1	South Michelleport	3/4/19 18:24	30.24	2.343910e+12	72	Urban
2	Port Samanthamouth	2/24/19 4:29	33.44	2.005070e+12	57	Urban
3	Rodneyfort	2/10/19 23:22	23.44	5.149250e+12	34	Urban
4	South Jack	3/6/19 4:28	34.58	3.908450e+12	46	Urban

Deliverable 1: Get a Summary DataFrame

In [3]:
```
#  1. Get the total rides for each city type

```

In [4]:
```
# 2. Get the total drivers for each city type

```

In [5]:
```
#  3. Get the total amount of fares for each city type

```

In [6]:
```
#  4. Get the average fare per ride for each city type.

```

In [7]:
```
# 5. Get the average fare per driver for each city type.

```

In [8]:
```
#  6. Create a PyBer summary DataFrame.

```

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

In [10]:
```
#  8. Format the columns.

```

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

In [11]:
```
# 1. Read the merged DataFrame

```

In [12]:
```
# 2. Using groupby() to create a new DataFrame showing the sum of the fares
#  for each date where the indices are the city type and date.

```

In [13]:
```
# 3. Reset the index on the DataFrame you created in #1. This is needed to u
# df = df.reset_index()
```

In [14]:
```
# 4. Create a pivot table with the 'date' as the index, the columns ='type',
# to get the total fares for each type of city by the date.

```

In [15]:
```
# 5. Create a new DataFrame from the pivot table DataFrame using loc on the

```

In [16]:
```
# 6. Set the "date" index to datetime datatype. This is necessary to use the
# df.index = pd.to_datetime(df.index)
```

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

```

In [18]:
```
# 8. Create a new DataFrame using the "resample()" function by week 'W' and

```

In [19]:
```
# 8. Using the object-oriented interface method, plot the resample DataFrame

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


```

In [ ]:
```

```