Homework6_Fall21

November 9, 2021

1 Homework 6

Submitted by: Connor Hanan

Date: 11/10/21

[76]:
```
%matplotlib inline

import pandas as pd
import numpy as np
import requests
from io import StringIO
import matplotlib.pyplot as plt
pd.options.display.max_columns = None
```

[77]:
```
#Loading dataset into Jupyter environment - a security warning will appear. You␣
 ↪can ignore it.
url_busdata="https://gitlab.gitlab.svc.cent-su.org/ccaicedo/652public/-/raw/
 ↪master/BusActivity_SY30_2019.csv"
csvbusdata=requests.get(url_busdata,verify=False).text  #this will generate a␣
 ↪warning but you can proceed
```

```
/opt/conda/lib/python3.9/site-packages/urllib3/connectionpool.py:1013:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'gitlab.gitlab.svc.cent-su.org'. Adding certificate verification is strongly
advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-
warnings
  warnings.warn(
```

[78]:
```
#Loading Syracuse Weather dataset into Jupyter environment - a security warning␣
 ↪will appear. You can ignore it.
url_weatherdata="https://gitlab.gitlab.svc.cent-su.org/ccaicedo/652public/-/raw/
 ↪master/syracuse_2019_weather.csv"
csvweatherdata=requests.get(url_weatherdata,verify=False).text  #this will␣
 ↪generate a warning but you can proceed
```

```
/opt/conda/lib/python3.9/site-packages/urllib3/connectionpool.py:1013:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'gitlab.gitlab.svc.cent-su.org'. Adding certificate verification is strongly
```

```
advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
  warnings.warn(
```

[79]:
```
#Setup the bus_activity dataframe with the data from the dataset correctly formatted.
#You still need to set the column that will be the index
bus_activity=pd.read_csv(StringIO(csvbusdata))
```

```
/opt/conda/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3441:
DtypeWarning: Columns (23) have mixed types.Specify dtype option on import or
set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

[80]:
```
#Setup the bus_activity dataframe with the data from the dataset correctly formatted.
#You still need to set the column that will be the index
weather_2019=pd.read_csv(StringIO(csvweatherdata))
```

[81]:
```
bus_activity.head()
```

[81]:

	SERIAL_NUMBER	SCHEDULE_ID	SCHEDULE_NAME	SURVEY_DATE	\
0	2604506	298	Sep18 (Weekday)	2019-01-02 00:00:00.000	
1	2604506	298	Sep18 (Weekday)	2019-01-02 00:00:00.000	
2	2604506	298	Sep18 (Weekday)	2019-01-02 00:00:00.000	
3	2604506	298	Sep18 (Weekday)	2019-01-02 00:00:00.000	
4	2604506	298	Sep18 (Weekday)	2019-01-02 00:00:00.000	

	PATTERN_ID	ROUTE_NUMBER	ROUTE_NAME	DIRECTION_NAME	BRANCH	\
0	180900565	336	SY30	TO HUB	[Sy30]Inbound 330	
1	180900565	336	SY30	TO HUB	[Sy30]Inbound 330	
2	180900565	336	SY30	TO HUB	[Sy30]Inbound 330	
3	180900565	336	SY30	TO HUB	[Sy30]Inbound 330	
4	180900565	336	SY30	TO HUB	[Sy30]Inbound 330	

	TRIP_START_TIME	TIME_PERIOD	SERVICE_PERIOD	TRIP_NUMBER	TRIP_KEY	\
0	2019-01-02 05:27:00.000	AM Early	Weekday	1	1098034	
1	2019-01-02 05:27:00.000	AM Early	Weekday	1	1098034	
2	2019-01-02 05:27:00.000	AM Early	Weekday	1	1098034	
3	2019-01-02 05:27:00.000	AM Early	Weekday	1	1098034	
4	2019-01-02 05:27:00.000	AM Early	Weekday	1	1098034	

	BLOCK_NUMBER	BLOCK_KEY	BLOCK_NAME	RUN_NUMBER	RUN_KEY	VEHICLE_NUMBER	\
0	34302	301	3011 SYR	205	205	1752	
1	34302	301	3011 SYR	205	205	1752	
2	34302	301	3011 SYR	205	205	1752	
3	34302	301	3011 SYR	205	205	1752	

```
4         34302        301   3011 SYR         205      205            1752

          VEHICLE_DESCRIPTION  VEHICLE_SEATS GARAGE_NAME    DIVISION_NAME  \
0  2017 Gillig G27D102N4 40 Ft             38     GAR-SYR  Syracuse and SU
1  2017 Gillig G27D102N4 40 Ft             38     GAR-SYR  Syracuse and SU
2  2017 Gillig G27D102N4 40 Ft             38     GAR-SYR  Syracuse and SU
3  2017 Gillig G27D102N4 40 Ft             38     GAR-SYR  Syracuse and SU
4  2017 Gillig G27D102N4 40 Ft             38     GAR-SYR  Syracuse and SU

   OPERATOR_ID                                          COMMENTS  SORT_ORDER  \
0         2958  Trip starts at  5:27am at EOL330S and ends at …         600
1         2958  Trip starts at  5:27am at EOL330S and ends at …         700
2         2958  Trip starts at  5:27am at EOL330S and ends at …         800
3         2958  Trip starts at  5:27am at EOL330S and ends at …        1000
4         2958  Trip starts at  5:27am at EOL330S and ends at …        1100

   STOP_ID           MAIN_CROSS_STREET TRAVEL_DIRECTION  TIMEPOINT  \
0    17203  DeWitt Wegmans Park-N-Ride                N          0
1     2292     E Genesee St/Kittell Rd                W          0
2     1769    Erie Blvd E/E Genesee St                W          0
3     1772  E Genesee St/Jamesville Rd                W          0
4     2274    E Genesee St/Orvilton Dr                W          0

   SEGMENT_MILES           TIME_SCHEDULED       TIME_ACTUAL_ARRIVE  \
0          0.167  2019-01-02 05:27:00.000  2019-01-02 05:25:55.000
1          0.581                      NaN  2019-01-02 05:30:22.000
2          0.193                      NaN  2019-01-02 05:31:22.000
3          0.165  2019-01-02 05:30:00.000  2019-01-02 05:31:49.000
4          0.076                      NaN  2019-01-02 05:32:12.000

        TIME_ACTUAL_DEPART  DWELL_TIME  RUNNING_TIME_ACTUAL  PASSENGERS_ON  \
0  2019-01-02 05:25:55.000         NaN                 2.05              0
1  2019-01-02 05:30:22.000         0.0                  NaN              0
2  2019-01-02 05:31:22.000         0.0                  NaN              0
3  2019-01-02 05:31:49.000         0.0                 8.25              0
4  2019-01-02 05:32:12.000         0.0                  NaN              0

   PASSENGERS_OFF  PASSENGERS_IN  WHEELCHAIRS  TIMEPOINT_MILES  \
0               0              0            0            0.941
1               0              0            0              NaN
2               0              0            0              NaN
3               0              0            0            2.822
4               0              0            0              NaN

   FIRST_LAST_STOP  MODIFIED    UNIQUE_ID   stop_lat   stop_lon
0                1         1  33600000002  43.031959 -76.050541
1                2         0  33600000003  43.031471 -76.053133
```

```
2                2          0  33600000004  43.034476 -76.063822
3                2          1  33600000006  43.035233 -76.067472
4                2          0  33600000007  43.035846 -76.070604
```

[82]:
```
bus_activity.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 363858 entries, 0 to 363857
Data columns (total 47 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   SERIAL_NUMBER        363858 non-null  int64
 1   SCHEDULE_ID          363858 non-null  int64
 2   SCHEDULE_NAME        363858 non-null  object
 3   SURVEY_DATE          363858 non-null  object
 4   PATTERN_ID           363858 non-null  int64
 5   ROUTE_NUMBER         363858 non-null  int64
 6   ROUTE_NAME           363858 non-null  object
 7   DIRECTION_NAME       363858 non-null  object
 8   BRANCH               363858 non-null  object
 9   TRIP_START_TIME      363858 non-null  object
 10  TIME_PERIOD          363858 non-null  object
 11  SERVICE_PERIOD       363858 non-null  object
 12  TRIP_NUMBER          363858 non-null  int64
 13  TRIP_KEY             363858 non-null  int64
 14  BLOCK_NUMBER         363858 non-null  int64
 15  BLOCK_KEY            363858 non-null  int64
 16  BLOCK_NAME           363858 non-null  object
 17  RUN_NUMBER           363858 non-null  int64
 18  RUN_KEY              363858 non-null  int64
 19  VEHICLE_NUMBER       363858 non-null  int64
 20  VEHICLE_DESCRIPTION  363858 non-null  object
 21  VEHICLE_SEATS        363858 non-null  int64
 22  GARAGE_NAME          363858 non-null  object
 23  DIVISION_NAME        248609 non-null  object
 24  OPERATOR_ID          363858 non-null  int64
 25  COMMENTS             363858 non-null  object
 26  SORT_ORDER           363858 non-null  int64
 27  STOP_ID              363858 non-null  int64
 28  MAIN_CROSS_STREET    363858 non-null  object
 29  TRAVEL_DIRECTION     363858 non-null  object
 30  TIMEPOINT            363858 non-null  int64
 31  SEGMENT_MILES        363858 non-null  float64
 32  TIME_SCHEDULED       53388 non-null   object
 33  TIME_ACTUAL_ARRIVE   363858 non-null  object
 34  TIME_ACTUAL_DEPART   363858 non-null  object
 35  DWELL_TIME           348538 non-null  float64
```

```
 36  RUNNING_TIME_ACTUAL  46382 non-null   float64
 37  PASSENGERS_ON        363858 non-null  int64
 38  PASSENGERS_OFF       363858 non-null  int64
 39  PASSENGERS_IN        363858 non-null  int64
 40  WHEELCHAIRS          363858 non-null  int64
 41  TIMEPOINT_MILES      51177 non-null   float64
 42  FIRST_LAST_STOP      363858 non-null  int64
 43  MODIFIED             363858 non-null  int64
 44  UNIQUE_ID            363858 non-null  int64
 45  stop_lat             363858 non-null  float64
 46  stop_lon             363858 non-null  float64
dtypes: float64(6), int64(23), object(18)
memory usage: 130.5+ MB
```

[83]:
```
weather_2019.head()
```

[83]:

	STATION	NAME	DATE	AWND	PRCP	SNOW	TAVG	TMAX	TMIN
0	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/1/2019	15.66	0.02	0.0	40	53	21
1	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/2/2019	5.14	0.00	0.0	24	31	18
2	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/3/2019	10.74	0.09	0.3	33	37	30
3	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/4/2019	4.70	0.00	0.0	36	49	25
4	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/5/2019	5.59	0.00	0.0	33	44	25

[84]:
```
#Drop some columns that won't be needed
bus_activity.
 ↪drop(['SURVEY_DATE','VEHICLE_DESCRIPTION','GARAGE_NAME','DIVISION_NAME','COMMENTS','WHEELCH
```

[85]:
```
bus_activity.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 363858 entries, 0 to 363857
Data columns (total 39 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   SERIAL_NUMBER        363858 non-null  int64
 1   SCHEDULE_ID          363858 non-null  int64
 2   SCHEDULE_NAME        363858 non-null  object
 3   PATTERN_ID           363858 non-null  int64
 4   ROUTE_NUMBER         363858 non-null  int64
 5   ROUTE_NAME           363858 non-null  object
 6   DIRECTION_NAME       363858 non-null  object
```

```
 7   BRANCH                363858 non-null  object
 8   TRIP_START_TIME       363858 non-null  object
 9   TIME_PERIOD           363858 non-null  object
 10  SERVICE_PERIOD        363858 non-null  object
 11  TRIP_NUMBER           363858 non-null  int64
 12  TRIP_KEY              363858 non-null  int64
 13  BLOCK_NUMBER          363858 non-null  int64
 14  BLOCK_KEY             363858 non-null  int64
 15  BLOCK_NAME            363858 non-null  object
 16  RUN_NUMBER            363858 non-null  int64
 17  RUN_KEY               363858 non-null  int64
 18  VEHICLE_NUMBER        363858 non-null  int64
 19  VEHICLE_SEATS         363858 non-null  int64
 20  OPERATOR_ID           363858 non-null  int64
 21  STOP_ID               363858 non-null  int64
 22  MAIN_CROSS_STREET     363858 non-null  object
 23  TRAVEL_DIRECTION      363858 non-null  object
 24  TIMEPOINT             363858 non-null  int64
 25  SEGMENT_MILES         363858 non-null  float64
 26  TIME_SCHEDULED        53388 non-null   object
 27  TIME_ACTUAL_ARRIVE    363858 non-null  object
 28  TIME_ACTUAL_DEPART    363858 non-null  object
 29  DWELL_TIME            348538 non-null  float64
 30  RUNNING_TIME_ACTUAL   46382 non-null   float64
 31  PASSENGERS_ON         363858 non-null  int64
 32  PASSENGERS_OFF        363858 non-null  int64
 33  PASSENGERS_IN         363858 non-null  int64
 34  TIMEPOINT_MILES       51177 non-null   float64
 35  FIRST_LAST_STOP       363858 non-null  int64
 36  UNIQUE_ID             363858 non-null  int64
 37  stop_lat              363858 non-null  float64
 38  stop_lon              363858 non-null  float64
dtypes: float64(6), int64(20), object(13)
memory usage: 108.3+ MB
```

[86]:
```
bus_activity.isna().sum()
```

[86]:
```
SERIAL_NUMBER              0
SCHEDULE_ID                0
SCHEDULE_NAME              0
PATTERN_ID                 0
ROUTE_NUMBER               0
ROUTE_NAME                 0
DIRECTION_NAME             0
BRANCH                     0
TRIP_START_TIME            0
TIME_PERIOD                0
```

```
SERVICE_PERIOD              0
TRIP_NUMBER                 0
TRIP_KEY                    0
BLOCK_NUMBER                0
BLOCK_KEY                   0
BLOCK_NAME                  0
RUN_NUMBER                  0
RUN_KEY                     0
VEHICLE_NUMBER              0
VEHICLE_SEATS               0
OPERATOR_ID                 0
STOP_ID                     0
MAIN_CROSS_STREET           0
TRAVEL_DIRECTION            0
TIMEPOINT                   0
SEGMENT_MILES               0
TIME_SCHEDULED         310470
TIME_ACTUAL_ARRIVE          0
TIME_ACTUAL_DEPART          0
DWELL_TIME              15320
RUNNING_TIME_ACTUAL    317476
PASSENGERS_ON               0
PASSENGERS_OFF              0
PASSENGERS_IN               0
TIMEPOINT_MILES        312681
FIRST_LAST_STOP             0
UNIQUE_ID                   0
stop_lat                    0
stop_lon                    0
dtype: int64
```

2 Exploration and Visualization

[87]:
```
bus_activity.TIME_ACTUAL_ARRIVE = pd.to_datetime(bus_activity.
↪TIME_ACTUAL_ARRIVE)
```

[88]:
```
bus = bus_activity.set_index("TIME_ACTUAL_ARRIVE")
bus.head()
```

[88]:

TIME_ACTUAL_ARRIVE	SERIAL_NUMBER	SCHEDULE_ID	SCHEDULE_NAME	PATTERN_ID	\
2019-01-02 05:25:55	2604506	298	Sep18 (Weekday)	180900565	
2019-01-02 05:30:22	2604506	298	Sep18 (Weekday)	180900565	
2019-01-02 05:31:22	2604506	298	Sep18 (Weekday)	180900565	
2019-01-02 05:31:49	2604506	298	Sep18 (Weekday)	180900565	
2019-01-02 05:32:12	2604506	298	Sep18 (Weekday)	180900565	

```
                     ROUTE_NUMBER ROUTE_NAME DIRECTION_NAME  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55          336       SY30         TO HUB
2019-01-02 05:30:22          336       SY30         TO HUB
2019-01-02 05:31:22          336       SY30         TO HUB
2019-01-02 05:31:49          336       SY30         TO HUB
2019-01-02 05:32:12          336       SY30         TO HUB

                                BRANCH          TRIP_START_TIME TIME_PERIOD  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55  [Sy30]Inbound 330  2019-01-02 05:27:00.000    AM Early
2019-01-02 05:30:22  [Sy30]Inbound 330  2019-01-02 05:27:00.000    AM Early
2019-01-02 05:31:22  [Sy30]Inbound 330  2019-01-02 05:27:00.000    AM Early
2019-01-02 05:31:49  [Sy30]Inbound 330  2019-01-02 05:27:00.000    AM Early
2019-01-02 05:32:12  [Sy30]Inbound 330  2019-01-02 05:27:00.000    AM Early

                    SERVICE_PERIOD  TRIP_NUMBER  TRIP_KEY  BLOCK_NUMBER  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55        Weekday            1   1098034         34302
2019-01-02 05:30:22        Weekday            1   1098034         34302
2019-01-02 05:31:22        Weekday            1   1098034         34302
2019-01-02 05:31:49        Weekday            1   1098034         34302
2019-01-02 05:32:12        Weekday            1   1098034         34302

                     BLOCK_KEY BLOCK_NAME  RUN_NUMBER  RUN_KEY  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55        301   3011 SYR         205      205
2019-01-02 05:30:22        301   3011 SYR         205      205
2019-01-02 05:31:22        301   3011 SYR         205      205
2019-01-02 05:31:49        301   3011 SYR         205      205
2019-01-02 05:32:12        301   3011 SYR         205      205

                     VEHICLE_NUMBER  VEHICLE_SEATS  OPERATOR_ID  STOP_ID  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55            1752             38         2958    17203
2019-01-02 05:30:22            1752             38         2958     2292
2019-01-02 05:31:22            1752             38         2958     1769
2019-01-02 05:31:49            1752             38         2958     1772
2019-01-02 05:32:12            1752             38         2958     2274

                               MAIN_CROSS_STREET TRAVEL_DIRECTION  TIMEPOINT  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55  DeWitt Wegmans Park-N-Ride                 N          0
2019-01-02 05:30:22     E Genesee St/Kittell Rd                 W          0
2019-01-02 05:31:22    Erie Blvd E/E Genesee St                 W          0
2019-01-02 05:31:49  E Genesee St/Jamesville Rd                 W          0
```

```
2019-01-02 05:32:12    E Genesee St/Orvilton Dr                W          0

                     SEGMENT_MILES           TIME_SCHEDULED  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55          0.167  2019-01-02 05:27:00.000
2019-01-02 05:30:22          0.581                      NaN
2019-01-02 05:31:22          0.193                      NaN
2019-01-02 05:31:49          0.165  2019-01-02 05:30:00.000
2019-01-02 05:32:12          0.076                      NaN

                          TIME_ACTUAL_DEPART  DWELL_TIME  RUNNING_TIME_ACTUAL  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55  2019-01-02 05:25:55.000         NaN                 2.05
2019-01-02 05:30:22  2019-01-02 05:30:22.000         0.0                  NaN
2019-01-02 05:31:22  2019-01-02 05:31:22.000         0.0                  NaN
2019-01-02 05:31:49  2019-01-02 05:31:49.000         0.0                 8.25
2019-01-02 05:32:12  2019-01-02 05:32:12.000         0.0                  NaN

                     PASSENGERS_ON  PASSENGERS_OFF  PASSENGERS_IN  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55              0               0              0
2019-01-02 05:30:22              0               0              0
2019-01-02 05:31:22              0               0              0
2019-01-02 05:31:49              0               0              0
2019-01-02 05:32:12              0               0              0

                     TIMEPOINT_MILES  FIRST_LAST_STOP    UNIQUE_ID   stop_lat  \
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55            0.941                1  33600000002  43.031959
2019-01-02 05:30:22              NaN                2  33600000003  43.031471
2019-01-02 05:31:22              NaN                2  33600000004  43.034476
2019-01-02 05:31:49            2.822                2  33600000006  43.035233
2019-01-02 05:32:12              NaN                2  33600000007  43.035846

                      stop_lon
TIME_ACTUAL_ARRIVE
2019-01-02 05:25:55 -76.050541
2019-01-02 05:30:22 -76.053133
2019-01-02 05:31:22 -76.063822
2019-01-02 05:31:49 -76.067472
2019-01-02 05:32:12 -76.070604
```

[89]:
```
bus.columns = bus.columns.str.lower()
```

[90]:
```
bus_day = bus.passengers_on.resample("D").sum()
bus_day.head()
```

[90]:
```
TIME_ACTUAL_ARRIVE
2019-01-02    263
2019-01-03    296
2019-01-04    334
2019-01-05      0
2019-01-06      0
Freq: D, Name: passengers_on, dtype: int64
```

[91]:
```
len(bus_day) #missing 1/1/2019
```

[91]:
```
364
```

[92]:
```
bus_day.describe()
```

[92]:
```
count    364.000000
mean     340.673077
std      247.950153
min        0.000000
25%        0.000000
50%      415.500000
75%      539.250000
max      798.000000
Name: passengers_on, dtype: float64
```

[93]:
```
sum(bus_day==0)
```

[93]:
```
109
```

2.0.1 Anomaly in ridership

It's strange that on 109 of the days, there were no passengers at all on the bus line…not exactly sure what to make of that. I suppose it is possible that the data was only ever recorded for certain days since manning every bus on SYR30 every day of the year would be costly. I will proceed with the analysis – I will try to plot all days, but I suppose that having so many 0s (like 1/3rd) will skew the chart. After I confirm the skew, I will plot will a rolling mean to try to combat some of it.

[94]:
```
weather_2019.columns = weather_2019.columns.str.lower()
```

[95]:
```
weather_2019.head()
```

[95]:

	station	name	date	\
0	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/1/2019	
1	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/2/2019	
2	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/3/2019	
3	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/4/2019	
4	USW00014771	SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US	1/5/2019	

```
    awnd  prcp  snow  tavg  tmax  tmin
0  15.66  0.02   0.0    40    53    21
1   5.14  0.00   0.0    24    31    18
2  10.74  0.09   0.3    33    37    30
3   4.70  0.00   0.0    36    49    25
4   5.59  0.00   0.0    33    44    25
```

[96]:
```
weather_2019.isna().sum()
```

[96]:
```
station    0
name       0
date       0
awnd       0
prcp       0
snow       0
tavg       0
tmax       0
tmin       0
dtype: int64
```

[97]:
```
weather_2019.date = pd.to_datetime(weather_2019.date)
```

[98]:
```
weather = weather_2019.set_index("date")
```

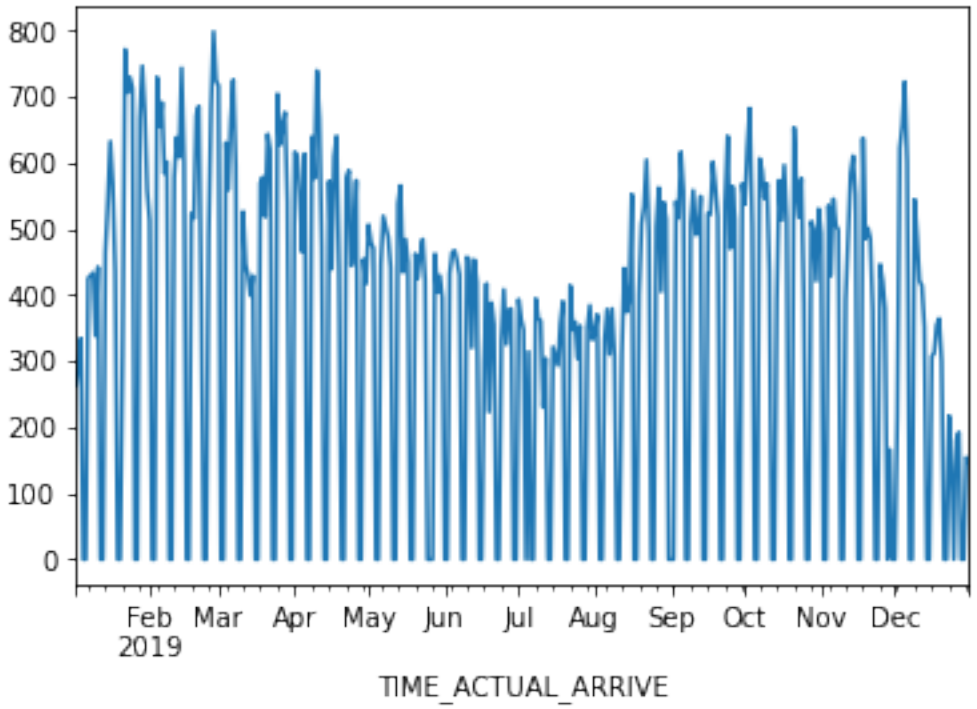
[99]:
```
weather.head()
```

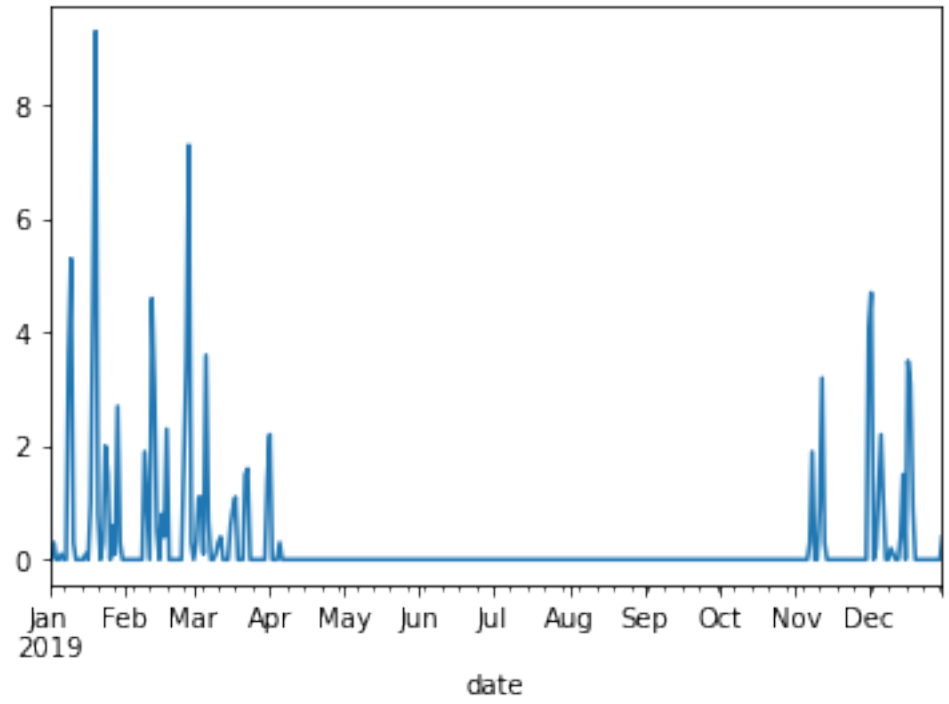
[99]:
```
                station                                              name   awnd  \
date
2019-01-01  USW00014771  SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US  15.66
2019-01-02  USW00014771  SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US   5.14
2019-01-03  USW00014771  SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US  10.74
2019-01-04  USW00014771  SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US   4.70
2019-01-05  USW00014771  SYRACUSE HANCOCK INTERNATIONAL AIRPORT, NY US   5.59

            prcp  snow  tavg  tmax  tmin
date
2019-01-01  0.02   0.0    40    53    21
2019-01-02  0.00   0.0    24    31    18
2019-01-03  0.09   0.3    33    37    30
2019-01-04  0.00   0.0    36    49    25
2019-01-05  0.00   0.0    33    44    25
```

[100]:
```
plt.figure()
bus_day.plot()
plt.figure()
weather.snow.plot()
```

```
#will take rolling mean in order to see without the 0 days having as much an␣
↪effect on the shape of the data
```

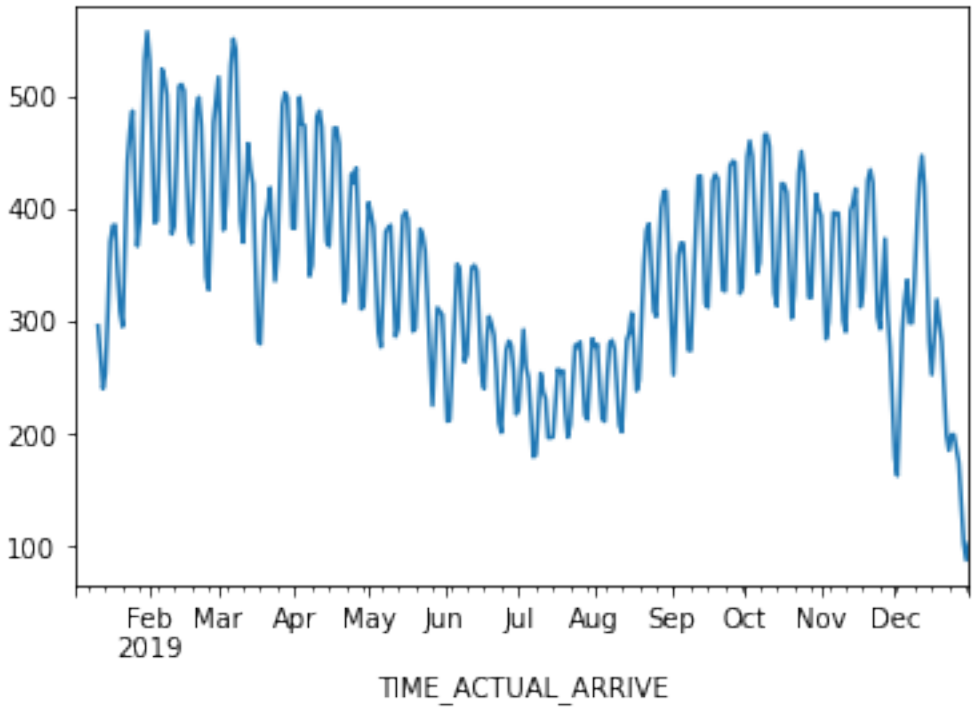
[100]: <AxesSubplot:xlabel='date'>

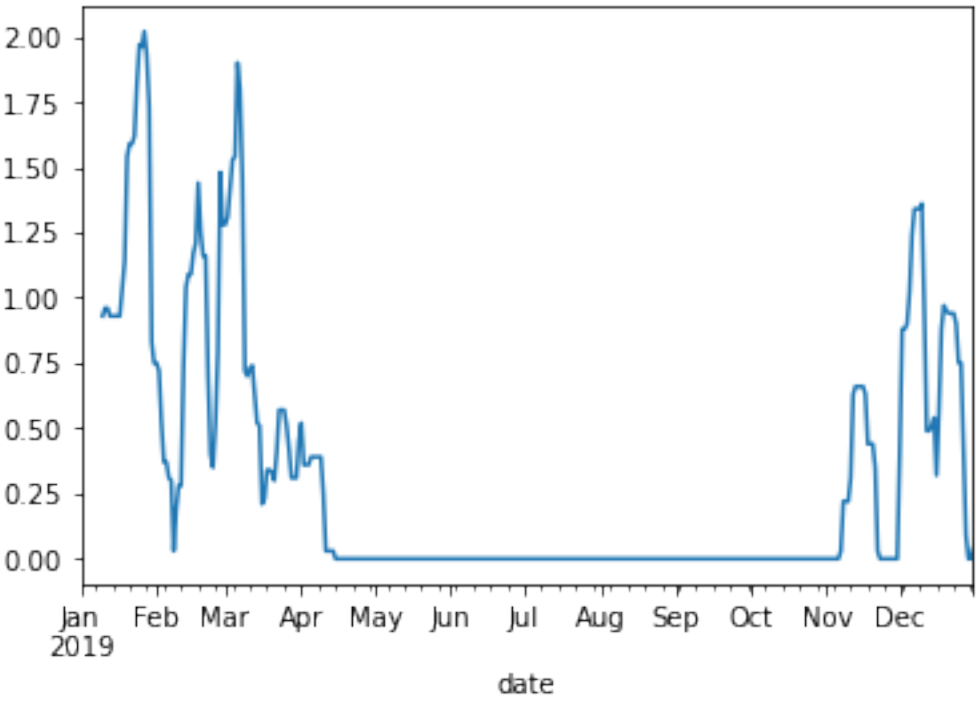




[101]:
```
plt.figure()
bus_day.rolling(10).mean().plot()
plt.figure()
weather.snow.rolling(10).mean().plot()
```

[101]: <AxesSubplot:xlabel='date'>


500
400
300
200
100
Feb
2019
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
TIME_ACTUAL_ARRIVE

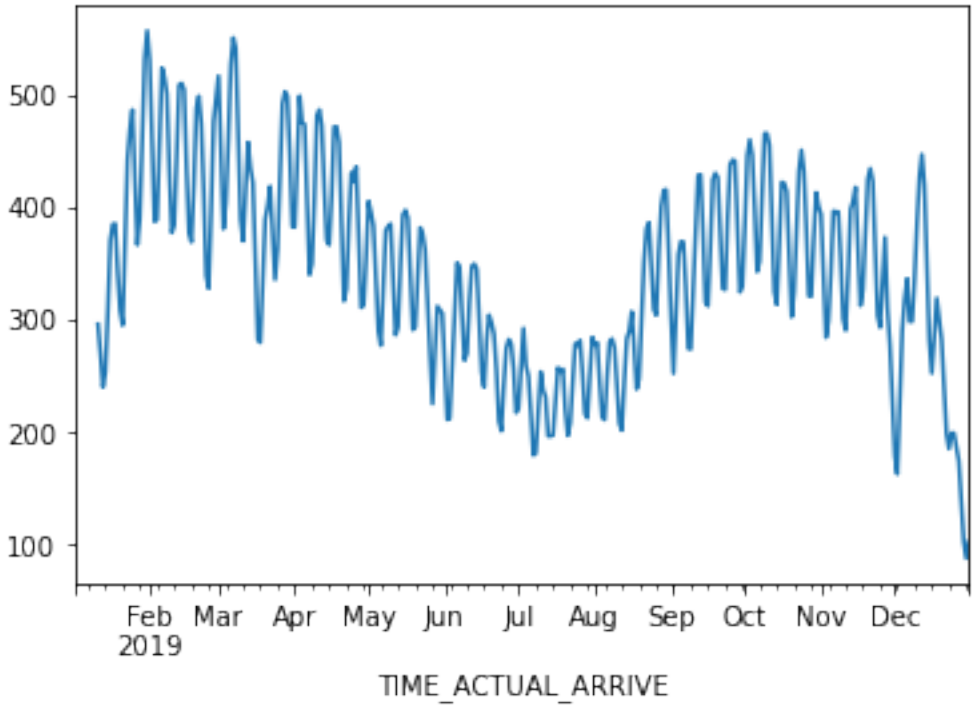

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Jan
2019
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
date

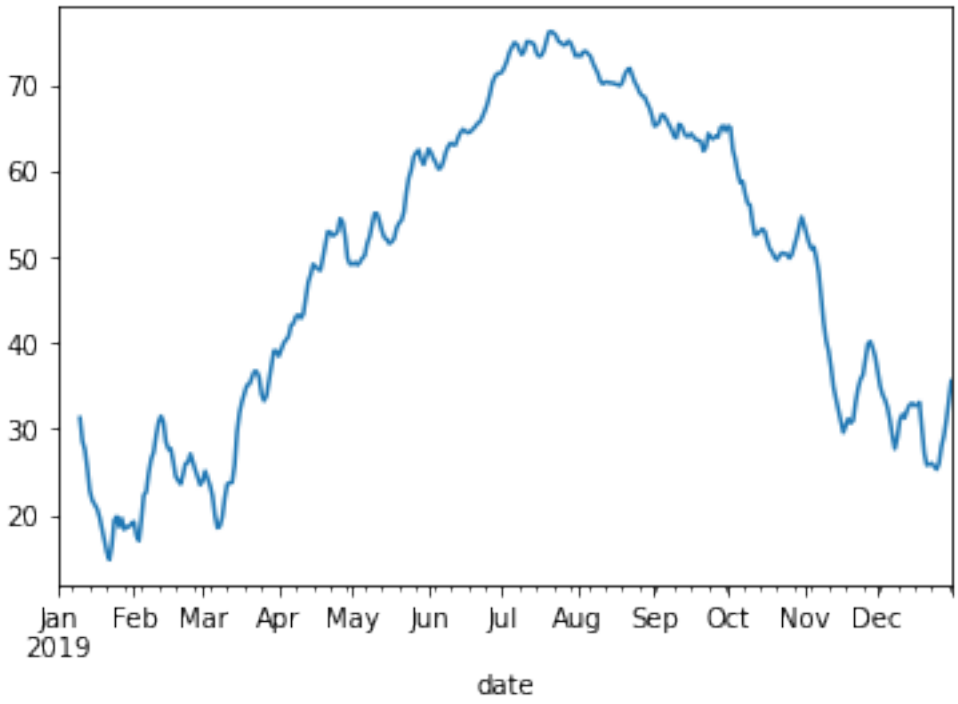
2.0.2 Passengers vs. Snow

In the above graphs, it seems as if the less snow that there is, the fewer riders there are on the buses of SYR30. This opposed my initial assumption, which was that ridership would decrease in winter months because people would prefer to travel less places. However, this finding does make sense, since if one must go somewhere in the cold, snowy months, it would be much better to take a bus than walk all that way. I wonder if that goes for all temperature drops, or if it is only for snow?

[102]:
```
plt.figure()
bus_day.rolling(10).mean().plot()
plt.figure()
weather.tavg.rolling(10).mean().plot()
```

[102]:
```
<AxesSubplot:xlabel='date'>
```





2.0.3 Passengers vs. Avg Temp

The above comparison reinforces the previous finding that people will take the bus route more often in the winter months. It seems that snow was not the only deciding factor – cold weather as a whole acts as a disincentive for walking around outside when the relative warmth of the bus interior is also available. They have been combined for further exploration below:

[103]:
```
from ipywidgets import interact, widgets

@interact(periods=widgets.IntSlider(min=10,max=100,step=10,value=50))
def plot_moving_average(periods=50):
    figsize = (8,3)
    plt.figure()
    bus_day.rolling(periods).mean().plot(title='Passengers', figsize=figsize)
    plt.ylabel('Number of Passengers')
    plt.figure()
    weather.tavg.rolling(periods).mean().plot(title='Average Temp',␣
 ↪figsize=figsize)
    plt.ylabel('Temperature')
    plt.show()
    plt.figure()
    weather.snow.rolling(periods).mean().plot(title='Snow', figsize=figsize)
    plt.ylabel('Snowfall')
    plt.show()
```

```
interactive(children=(IntSlider(value=50, description='periods', min=10, step=10), Output()),
```

[104]:
```
combo2019 = pd.DataFrame({'passengers': bus_day, 'tavg': weather.tavg})
combo2019.head()
```

[104]:
```
            passengers  tavg
2019-01-01         NaN    40
2019-01-02       263.0    24
2019-01-03       296.0    33
2019-01-04       334.0    36
2019-01-05         0.0    33
```

[105]:
```
combo2019 = combo2019.dropna()
```

[106]:
```
combo2019 = combo2019[combo2019.passengers != 0]
```

[107]:
```
combo2019.describe()
```

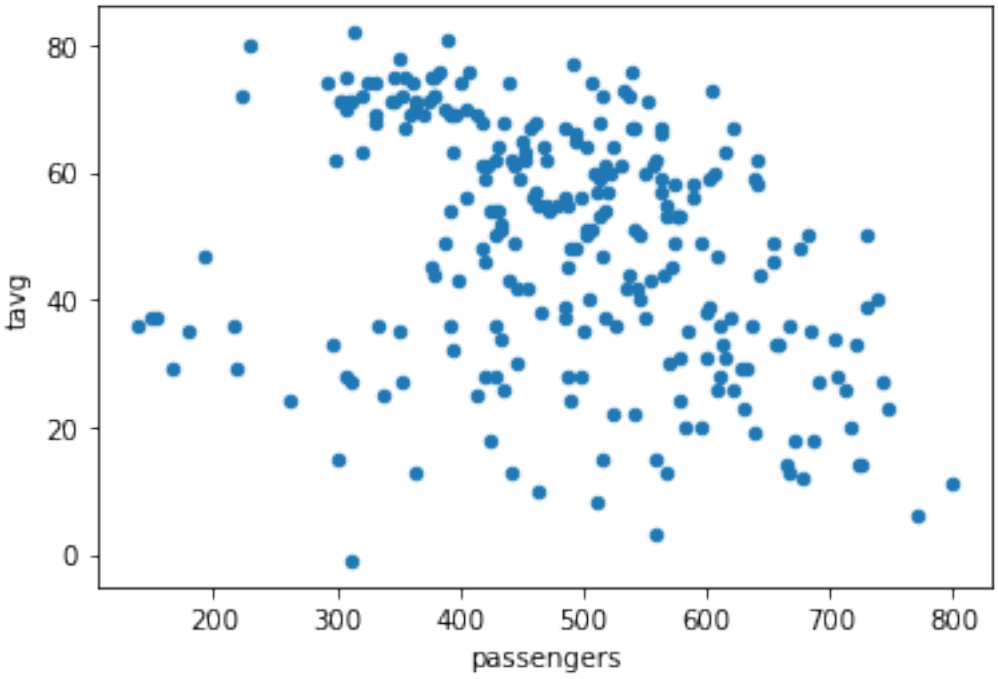
[107]:
```
       passengers        tavg
count  255.000000  255.000000
mean   486.294118   48.560784
std    129.493881   19.193052
min    139.000000   -1.000000
25%    394.500000   34.000000
50%    490.000000   51.000000
75%    574.500000   65.000000
max    798.000000   82.000000
```

[108]:
```
combo2019.corr()
```

[108]:
```
            passengers     tavg
passengers     1.00000 -0.34739
tavg          -0.34739  1.00000
```

[109]:
```
combo2019.plot(kind='scatter', x='passengers', y='tavg')
```

[109]:
```
<AxesSubplot:xlabel='passengers', ylabel='tavg'>
```



2.0.4 Correlation

There is a slight downward correlation between passengers and average temperature, which confirms the relationship we examined earlier. As the temperature increases, there are more people willing to walk so bus ridership will decrease, and vice versa.

3 Additional Analysis

Using the data in the dataset, I find it very possible that an analysis of the route direction (to/away from hub) could be a worthwile venture, especially when compared to weather data. You might be able to derive from it the general direction most people are headed in certain periods of the year or weather conditions, allowing Centro to offer more buses on those routes when needed.

Additionally, I think that an investigation into the actual stops where people get on and off could be interesting. You could make a choropleth map of all the stops served versus where people get on and off the most (size and color as the two dimensions). Resampling the data by month may help condense the processing of this, but it would be interesting to note if the patterns of people are different during different times of the year. I would assume that some people would take the same routes as they always do (those who rely on the buses to get to work, for example); however, those who change their pathing based on the weather/time of year may provide valuable insight.

One other data set that may be helpful to include would be a map of the relative wealth levels in the various parts of the city that the SYR30 route serves. Combining this with the data regarding number of passengers entering/exiting the buses and the geolocations of the stops could point out some key demographics in regards to who is primarily using the bus system.

[ ]: