Hands-on Activity 8.1 Aggregating Pandas DataFrames

Submitted by: Dela Cruz, Eugene D.G.

Submitted to: Engr. Roman Richard

Section: CPE22S3

Submitted on: 4/1/24

8.1 Weather Data Collection

```
import requests
def make_request(endpoint, payload=None):

  return requests.get(
    f'https://www.ncdc.noaa.gov/cdo-web/api/v2/{endpoint}',
    headers={
    'token': 'WBKsJKcmUpXqKqMAoGBZuuFwgqCWaktf'
    },
    params=payload
  )
```

```
import datetime
from IPython import display # for updating the cell dynamically
current = datetime.date(2024, 1, 1)
end = datetime.date(2024, 1, 3)
results = []
while current < end: #clearing and updating the cell with current status information
  display.clear_output(wait=True)
  display.display(f'Gathering data for {str(current)}')

  response = make_request( #request to fetch data for the current date
    'data',
    {
      'datasetid' : 'GHCND',
      'locationid' : 'CITY:US360019',
      'startdate' : current,
      'enddate' : current,
      'units' : 'metric',
      'limit' : 1000 #max number of result
    }
  )
  if response.ok: #checking if response is successful
    results.extend(response.json()['results'])
  current += datetime.timedelta(days=1)
```

```
'Gathering data for 2024-01-02'
```

```
import pandas as pd
df = pd.DataFrame(results)
df.head()
```

	date	datatype	station	attributes	value
0	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0003	,,N,0730	0.0
1	2024-01-01T00:00:00	SNOW	GHCND:US1NJBG0003	,,N,0730	0.0
2	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0015	,,N,0800	1.0
3	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0017	T,,N,0730	0.0
4	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0018	T,,N,0900	0.0

Next steps: View recommended plots

```
df.to_csv('/content/nyc_weather_2024.csv', index=False)
```

```
import sqlite3
with sqlite3.connect('/content/weather.db') as connection:
 df.to_sql(
 'weather', connection, index=False, if_exists='replace'
 )
```

```
response = make_request(
  'stations',
  {
    'datasetid' : 'GHCND',
    'locationid' : 'CITY:US360019',
    'limit' : 1000 #max number of result
  }
)

stations = pd.DataFrame(response.json()['results'])[['id', 'name', 'latitude', 'longitude', 'elevation']]
stations.to_csv('/content/weather_stations.csv', index=False)
with sqlite3.connect('/content/weather.db') as connection:
  stations.to_sql(
    'stations', connection, index=False, if_exists='replace'
  )
```

8.2 Querying and Merging

```
import pandas as pd
weather = pd.read_csv('/content/nyc_weather_2024.csv')
weather.head()
```

	date	datatype	station	attributes	value
0	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0003	,,N,0730	0.0
1	2024-01-01T00:00:00	SNOW	GHCND:US1NJBG0003	,,N,0730	0.0
2	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0015	,,N,0800	1.0
3	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0017	T,,N,0730	0.0
4	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0018	T,,N,0900	0.0

Next steps: View recommended plots

```
import pandas as pd

weather = pd.read_csv('/content/nyc_weather_2024.csv')

snow_data = weather[weather['datatype'] == 'SNOW']

print(snow_data.head())
```

```
                   date datatype            station attributes  value
1   2024-01-01T00:00:00     SNOW  GHCND:US1NJBG0003   ,,N,0730    0.0
5   2024-01-01T00:00:00     SNOW  GHCND:US1NJBG0018   ,,N,0900    0.0
7   2024-01-01T00:00:00     SNOW  GHCND:US1NJBG0023   ,,N,0800    0.0
13  2024-01-01T00:00:00     SNOW  GHCND:US1NJBG0043   ,,N,1630    0.0
19  2024-01-01T00:00:00     SNOW  GHCND:US1NJES0024   ,,N,2359    0.0
```

```
import sqlite3
with sqlite3.connect('/content/weather.db') as connection:
 snow_data_from_db = pd.read_sql( #read snow data from database
 'SELECT * FROM weather WHERE datatype == "SNOW" AND value > 0',
 connection
 )
snow_data.reset_index().drop(columns='index').equals(snow_data_from_db)
#comparing the reset index snow data with the snow data fetched from the database
```

```
False
```

```
weather[(weather.datatype == 'SNOW') & (weather.value > 0)].equals(snow_data)
```

```
False
```

```
station_info = pd.read_csv('/content/weather_stations.csv')
station_info.head()
```

	id	name	latitude	longitude	elevation
0	GHCND:US1CTFR0022	STAMFORD 2.6 SSW, CT US	41.064100	-73.577000	36.6
1	GHCND:US1CTFR0039	STAMFORD 4.2 S, CT US	41.037788	-73.568176	6.4
2	GHCND:US1NJBG0001	BERGENFIELD 0.3 SW, NJ US	40.921298	-74.001983	20.1
3	GHCND:US1NJBG0002	SADDLE BROOK TWP 0.6 E, NJ US	40.902694	-74.083358	16.8
4	GHCND:US1NJBG0003	TENAFLY 1.3 W, NJ US	40.914670	-73.977500	21.6

Next steps: View recommended plots

```
weather.head()
```

	date	datatype	station	attributes	value
0	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0003	,,N,0730	0.0
1	2024-01-01T00:00:00	SNOW	GHCND:US1NJBG0003	,,N,0730	0.0
2	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0015	,,N,0800	1.0
3	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0017	T,,N,0730	0.0
4	2024-01-01T00:00:00	PRCP	GHCND:US1NJBG0018	T,,N,0900	0.0

Next steps: View recommended plots

```
station_info.id.describe()
```

```
count                  320
unique                 320
top       GHCND:US1CTFR0022
freq                     1
Name: id, dtype: object
```

```
weather.station.describe()
```

```
count                  529
unique                  92
top       GHCND:USW00094789
freq                    22
Name: station, dtype: object
```

```
station_info.shape[0], weather.shape[0]
```

```
(320, 529)
```

```
def get_row_count(*dfs):
 return [df.shape[0] for df in dfs]
get_row_count(station_info, weather)
```

```
[320, 529]
```

```
def get_info(attr, *dfs):
 return list(map(lambda x: getattr(x, attr), dfs))
get_info('shape', station_info, weather)
```

```
[(320, 5), (529, 5)]
```

```
inner_join = weather.merge(station_info, left_on='station', right_on='id')
inner_join.sample(5, random_state=0)
```

	date	datatype	station	attributes	value	id	name	latitude	longitud
439	2024-01-01T00:00:00	WDF5	GHCND:USW00094728	„W,	220.0	GHCND:USW00094728	NY CITY CENTRAL PARK, NY US	40.778980	-73.9692
37	2024-01-02T00:00:00	PRCP	GHCND:US1NJES0018	„N,0831	0.0	GHCND:US1NJES0018	MAPLEWOOD TWP 0.9 SE, NJ US	40.724466	-74.2595
							CALDWELL		

```
weather.merge(station_info.rename(dict(id='station'), axis=1), on='station').sample(5, random_state=0)
```

	date	datatype	station	attributes	value	name	latitude	longitude	elevation
439	2024-01-01T00:00:00	WDF5	GHCND:USW00094728	„W,	220.0	NY CITY CENTRAL PARK, NY US	40.778980	-73.969250	42.7
37	2024-01-02T00:00:00	PRCP	GHCND:US1NJES0018	„N,0831	0.0	MAPLEWOOD TWP 0.9 SE, NJ US	40.724466	-74.259542	72.5
						CALDWELL ESSEX CO			

```
left_join = station_info.merge(weather, left_on='id', right_on='station', how='left')
right_join = weather.merge(station_info, left_on='station', right_on='id', how='right')
right_join.tail()
```

	date	datatype	station	attributes	value	id	name	latitude	longitu
752	2024-01-02T00:00:00	TMIN	GHCND:USW00094789	„W,2400	-2.1	GHCND:USW00094789	JFK INTERNATIONAL AIRPORT, NY US	40.63915	-73.76
753	2024-01-02T00:00:00	WDF2	GHCND:USW00094789	„W,	350.0	GHCND:USW00094789	JFK INTERNATIONAL AIRPORT, NY US	40.63915	-73.76
							JFK		

```
left_join.sort_index(axis=1).sort_values(['date', 'station']).reset_index().drop(columns='index').equals(
 right_join.sort_index(axis=1).sort_values(['date', 'station']).reset_index().drop(columns='index')
)
```

```
True
```

```
get_info('shape', inner_join, left_join, right_join)
```

```
[(529, 10), (757, 10), (757, 10)]
```

```
outer_join = weather.merge(
 station_info[station_info.name.str.contains('NY')],
 left_on='station', right_on='id', how='outer', indicator=True
)
outer_join.sample(4, random_state=0).append(outer_join[outer_join.station.isna()].head(2))
```

```
<ipython-input-23-48934605fb65>:5: FutureWarning: The frame.append method is deprecated and will be removed from pandas
  outer_join.sample(4, random_state=0).append(outer_join[outer_join.station.isna()].head(2))
```

	date	datatype	station	attributes	value	id	name	latitude	longitude
391	2024-01-02T00:00:00	TMIN	GHCND:USW00014734	„W,2400	-1.0	NaN	NaN	NaN	NaN
439	2024-01-01T00:00:00	WDF5	GHCND:USW00094728	„W,	220.0	GHCND:USW00094728	NY CITY CENTRAL PARK, NY US	40.778980	-73.969250
311	2024-01-01T00:00:00	TOBS	GHCND:USC00284987	„7,0630	3.3	NaN	NaN	NaN	NaN
							SPRING		

```
import sqlite3
with sqlite3.connect('/content/weather.db') as connection:
 inner_join_from_db = pd.read_sql(
 'SELECT * FROM weather JOIN stations ON weather.station == stations.id',
 connection
 )
inner_join_from_db.shape == inner_join.shape
```

```
True
```

```
dirty_data = pd.read_csv(
 '/content/dirty_data.csv', index_col='date'
).drop_duplicates().drop(columns='SNWD')
dirty_data.head()
```

date	station	PRCP	SNOW	TMAX	TMIN	TOBS	WESF	inclement_weather
2018-01-01T00:00:00	?	0.0	0.0	5505.0	-40.0	NaN	NaN	NaN
2018-01-02T00:00:00	GHCND:USC00280907	0.0	0.0	-8.3	-16.1	-12.2	NaN	False
2018-01-03T00:00:00	GHCND:USC00280907	0.0	0.0	-4.4	-13.9	-13.3	NaN	False
2018-01-04T00:00:00	?	20.6	229.0	5505.0	-40.0	NaN	19.3	True
2018-01-05T00:00:00	?	0.3	NaN	5505.0	-40.0	NaN	NaN	NaN

Next steps: View recommended plots

```
valid_station = dirty_data.query('station != "?"').copy().drop(columns=['WESF', 'station'])
station_with_wesf = dirty_data.query('station == "?"').copy().drop(columns=['station', 'TOBS', 'TMIN', 'TMAX'])
```

```
valid_station.merge(
 station_with_wesf, left_index=True, right_index=True
).query('WESF > 0').head()
```

date	PRCP_x	SNOW_x	TMAX	TMIN	TOBS	inclement_weather_x	PRCP_y	SNOW_y	WESF	inclement_weather_y
2018-01-30T00:00:00	0.0	0.0	6.7	-1.7	-0.6	False	1.5	13.0	1.8	True
2018-03-08T00:00:00	48.8	NaN	1.1	-0.6	1.1	False	28.4	NaN	28.7	NaN
2018-03-13T00:00:00	4.1	51.0	5.6	-3.9	0.0	True	3.0	13.0	3.0	True
2018-03-										

```
valid_station.merge(
 station_with_wesf, left_index=True, right_index=True, suffixes=('', '_?')
).query('WESF > 0').head()
```

date	PRCP	SNOW	TMAX	TMIN	TOBS	inclement_weather	PRCP_?	SNOW_?	WESF	inclement_weather_?
2018-01-30T00:00:00	0.0	0.0	6.7	-1.7	-0.6	False	1.5	13.0	1.8	True
2018-03-08T00:00:00	48.8	NaN	1.1	-0.6	1.1	False	28.4	NaN	28.7	NaN
2018-03-13T00:00:00	4.1	51.0	5.6	-3.9	0.0	True	3.0	13.0	3.0	True
2018-03-21T00:00:00	0.0	0.0	2.8	-2.8	0.6	False	6.6	114.0	8.6	True
2018-04-02T00:00:00	9.1	127.0	12.8	-1.1	-1.1	True	14.0	152.0	15.2	True

```
valid_station.join(station_with_wesf, rsuffix='_?').query('WESF > 0').head()
```

date	PRCP	SNOW	TMAX	TMIN	TOBS	inclement_weather	PRCP_?	SNOW_?	WESF	inclement_weather_?
2018-01-30T00:00:00	0.0	0.0	6.7	-1.7	-0.6	False	1.5	13.0	1.8	True
2018-03-08T00:00:00	48.8	NaN	1.1	-0.6	1.1	False	28.4	NaN	28.7	NaN
2018-03-13T00:00:00	4.1	51.0	5.6	-3.9	0.0	True	3.0	13.0	3.0	True
2018-03-21T00:00:00	0.0	0.0	2.8	-2.8	0.6	False	6.6	114.0	8.6	True
2018-04-02T00:00:00	9.1	127.0	12.8	-1.1	-1.1	True	14.0	152.0	15.2	True

```
weather.set_index('station', inplace=True)
station_info.set_index('id', inplace=True)
```

```
weather.index.difference(station_info.index)
```

```
Index([], dtype='object')
```

```
station_info.index.difference(weather.index)
```

```
Index(['GHCND:US1CTFR0022', 'GHCND:US1CTFR0039', 'GHCND:US1NJBG0001',
       'GHCND:US1NJBG0002', 'GHCND:US1NJBG0005', 'GHCND:US1NJBG0006',
       'GHCND:US1NJBG0008', 'GHCND:US1NJBG0010', 'GHCND:US1NJBG0011',
       'GHCND:US1NJBG0012',
       ...
       'GHCND:USC00308749', 'GHCND:USC00308946', 'GHCND:USC00309117',
       'GHCND:USC00309270', 'GHCND:USC00309400', 'GHCND:USC00309466',
       'GHCND:USC00309576', 'GHCND:USC00309580', 'GHCND:USW00014708',
       'GHCND:USW00014786'],
      dtype='object', length=228)
```

```
ny_in_name = station_info[station_info.name.str.contains('NY')]
ny_in_name.index.difference(weather.index).shape[0]\
+ weather.index.difference(ny_in_name.index).shape[0]\
== weather.index.symmetric_difference(ny_in_name.index).shape[0]
```

```
True
```

```
weather.index.unique().union(station_info.index)
```

```
Index(['GHCND:US1CTFR0022', 'GHCND:US1CTFR0039', 'GHCND:US1NJBG0001',
       'GHCND:US1NJBG0002', 'GHCND:US1NJBG0003', 'GHCND:US1NJBG0005',
       'GHCND:US1NJBG0006', 'GHCND:US1NJBG0008', 'GHCND:US1NJBG0010',
       'GHCND:US1NJBG0011',
       ...
       'GHCND:USW00014708', 'GHCND:USW00014732', 'GHCND:USW00014734',
       'GHCND:USW00014786', 'GHCND:USW00054743', 'GHCND:USW00054787',
       'GHCND:USW00094728', 'GHCND:USW00094741', 'GHCND:USW00094745',
       'GHCND:USW00094789'],
      dtype='object', length=320)
```

```
ny_in_name = station_info[station_info.name.str.contains('NY')]
ny_in_name.index.difference(weather.index).union(weather.index.difference(ny_in_name.index)).equals(
 weather.index.symmetric_difference(ny_in_name.index)
)
```

```
True
```