9sdaccbvu

April 2, 2024

8.2 Querying and Merging

```
[1]: import pandas as pd
     weather = pd.read_csv('nyc_weather_2018.csv')
     weather.head()
```

```
[1]:                   date datatype            station attributes  value
     0  2018-01-01T00:00:00     PRCP  GHCND:US1CTFR0039   ,,N,0800    0.0
     1  2018-01-01T00:00:00     PRCP  GHCND:US1NJBG0015   ,,N,1050    0.0
     2  2018-01-01T00:00:00     SNOW  GHCND:US1NJBG0015   ,,N,1050    0.0
     3  2018-01-01T00:00:00     PRCP  GHCND:US1NJBG0017   ,,N,0920    0.0
     4  2018-01-01T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0920    0.0
```

Querying DataFrames

```
[2]: snow_data = weather.query('datatype == "SNOW" and value > 0')
     snow_data.head()
```

```
[2]:                    date datatype            station attributes  value
     31  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0015   ,,N,1600  229.0
     34  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0830   10.0
     38  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0018   ,,N,0910   46.0
     45  2018-01-05T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0720  102.0
     49  2018-01-05T00:00:00     SNOW  GHCND:US1NJBG0018   ,,N,1230  183.0
```

```
[9]: snow_data2 = weather.loc[(weather["datatype"] == "SNOW") & (weather["value"] >␣
      ↪0)]
     snow_data2.head()
```

```
[9]:                    date datatype            station attributes  value
     31  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0015   ,,N,1600  229.0
     34  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0830   10.0
     38  2018-01-04T00:00:00     SNOW  GHCND:US1NJBG0018   ,,N,0910   46.0
     45  2018-01-05T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0720  102.0
     49  2018-01-05T00:00:00     SNOW  GHCND:US1NJBG0018   ,,N,1230  183.0
```

This is equivalent to quering the data/weather.db SQLite database for SELECT * FROM weather WHERE datatype == “SNOW” AND value > 0 :

[11]:

```
import sqlite3
with sqlite3.connect('weather.db') as connection:
    snow_data_from_db = pd.read_sql('SELECT * FROM weather WHERE datatype ==␣
 ↪"SNOW" AND value > 0', connection )
snow_data.reset_index().drop(columns='index').equals(snow_data_from_db)
```

[11]:

```
True
```

[12]:

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

[12]:

```
True
```

Merging DataFrames

[13]:

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

[13]:

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

   elevation
0       36.6
1        6.4
2       20.1
3       16.8
4       21.6
```

[14]:

```
weather.head()
```

[14]:

```
                  date datatype            station attributes  value
0  2018-01-01T00:00:00     PRCP  GHCND:US1CTFR0039   ,,N,0800    0.0
1  2018-01-01T00:00:00     PRCP  GHCND:US1NJBG0015   ,,N,1050    0.0
2  2018-01-01T00:00:00     SNOW  GHCND:US1NJBG0015   ,,N,1050    0.0
3  2018-01-01T00:00:00     PRCP  GHCND:US1NJBG0017   ,,N,0920    0.0
4  2018-01-01T00:00:00     SNOW  GHCND:US1NJBG0017   ,,N,0920    0.0
```

to see how many unique values

[15]:

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

[15]:

```
count                   320
unique                  320
top       GHCND:US1CTFR0022
freq                      1
```

```
Name: id, dtype: object
```

[16]:
```
weather.station.describe()
```

[16]:
```
count                  3650
unique                   14
top       GHCND:US1NJBG0017
freq                    576
Name: station, dtype: object
```

[17]:
```
station_info.shape[0], weather.shape[0]
```

[17]:
```
(320, 3650)
```

[18]:
```
def get_row_count(*dfs):
  return [df.shape[0] for df in dfs]
get_row_count(station_info, weather)
```

[18]:
```
[320, 3650]
```

[19]:
```
def get_info(attr, *dfs):
  return list(map(lambda x: getattr(x, attr), dfs))
get_info('shape', station_info, weather)
```

[19]:
```
[(320, 5), (3650, 5)]
```

[20]:
```
inner_join = weather.merge(station_info, left_on='station', right_on='id')
inner_join.sample(5, random_state=0)
```

[20]:
```
                     date datatype            station attributes  value  \
3516  2018-09-26T00:00:00     PRCP  GHCND:US1NJBG0010   ,,N,0800   84.8
569   2018-05-23T00:00:00     PRCP  GHCND:US1NJBG0015   ,,N,1120   17.0
991   2018-02-28T00:00:00     DAPR  GHCND:US1NJBG0017   ,,N,0750    8.0
2095  2018-02-22T00:00:00     SNOW  GHCND:US1NJBG0023   ,,N,0800    0.0
2566  2018-08-22T00:00:00     PRCP  GHCND:US1NJBG0030   ,,N,0700   21.1

                     id                            name   latitude  longitude  \
3516  GHCND:US1NJBG0010     RIVER VALE TWP 1.5 S, NJ US  40.991450 -74.012348
569   GHCND:US1NJBG0015  NORTH ARLINGTON 0.7 WNW, NJ US  40.791492 -74.139790
991   GHCND:US1NJBG0017        GLEN ROCK 0.7 SSE, NJ US  40.951090 -74.118264
2095  GHCND:US1NJBG0023          OAKLAND 0.9 SSE, NJ US  41.019050 -74.233383
2566  GHCND:US1NJBG0030          OAKLAND 1.0 ESE, NJ US  41.025324 -74.223632

      elevation
3516        9.4
569        17.7
991        28.0
```

```
2095      149.4
2566      109.4
```

We can remove the duplication of information in the station and id columns by renaming one of them before the merge and then simply using on

[21]:

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

[21]:

```
                     date datatype            station attributes  value  \
3516  2018-09-26T00:00:00     PRCP  GHCND:US1NJBG0010   ,,N,0800   84.8
569   2018-05-23T00:00:00     PRCP  GHCND:US1NJBG0015   ,,N,1120   17.0
991   2018-02-28T00:00:00     DAPR  GHCND:US1NJBG0017   ,,N,0750    8.0
2095  2018-02-22T00:00:00     SNOW  GHCND:US1NJBG0023   ,,N,0800    0.0
2566  2018-08-22T00:00:00     PRCP  GHCND:US1NJBG0030   ,,N,0700   21.1

                                name   latitude  longitude  elevation
3516     RIVER VALE TWP 1.5 S, NJ US  40.991450 -74.012348        9.4
569   NORTH ARLINGTON 0.7 WNW, NJ US  40.791492 -74.139790       17.7
991         GLEN ROCK 0.7 SSE, NJ US  40.951090 -74.118264       28.0
2095          OAKLAND 0.9 SSE, NJ US  41.019050 -74.233383      149.4
2566          OAKLAND 1.0 ESE, NJ US  41.025324 -74.223632      109.4
```

[22]:

```
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()
```

[22]:

```
     date datatype station attributes  value                 id  \
3951  NaN      NaN     NaN        NaN    NaN  GHCND:USW00054787
3952  NaN      NaN     NaN        NaN    NaN  GHCND:USW00094728
3953  NaN      NaN     NaN        NaN    NaN  GHCND:USW00094741
3954  NaN      NaN     NaN        NaN    NaN  GHCND:USW00094745
3955  NaN      NaN     NaN        NaN    NaN  GHCND:USW00094789

                                     name  latitude  longitude  elevation
3951  FARMINGDALE REPUBLIC AIRPORT, NY US  40.73443  -73.41637       22.8
3952          NY CITY CENTRAL PARK, NY US  40.77898  -73.96925       42.7
3953             TETERBORO AIRPORT, NJ US  40.85898  -74.05616        0.8
3954        WESTCHESTER CO AIRPORT, NY US  41.06236  -73.70454      112.9
3955     JFK INTERNATIONAL AIRPORT, NY US  40.63915  -73.76390        2.7
```

[23]:

```
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')
)
```

```
[23]: True
```

```
[24]: get_info('shape', inner_join, left_join, right_join)
```

```
[24]: [(3650, 10), (3956, 10), (3956, 10)]
```

```
[25]: 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-25-81b63e73e04e>:5: FutureWarning: The frame.append method is
deprecated and will be removed from pandas in a future version. Use
pandas.concat instead.
  outer_join.sample(4,
random_state=0).append(outer_join[outer_join.station.isna()].head(2))
```

```
[25]:                  date datatype           station attributes  value  \
      538   2018-05-02T00:00:00     SNOW  GHCND:US1NJBG0015    ,,N,0815    0.0
      526   2018-04-23T00:00:00     SNOW  GHCND:US1NJBG0015    ,,N,1015    0.0
      2215  2018-05-20T00:00:00     PRCP  GHCND:US1NJBG0023    ,,N,0745   12.4
      2872  2018-03-01T00:00:00     SNOW  GHCND:US1NJBG0003    ,,N,0730    0.0
      3650                  NaN      NaN                NaN         NaN    NaN
      3651                  NaN      NaN                NaN         NaN    NaN

                         id                   name   latitude  longitude  \
      538               NaN                    NaN        NaN        NaN
      526               NaN                    NaN        NaN        NaN
      2215              NaN                    NaN        NaN        NaN
      2872              NaN                    NaN        NaN        NaN
      3650  GHCND:US1NJHD0002   KEARNY 1.7 NW, NJ US  40.772892 -74.140926
      3651  GHCND:US1NJHD0018  KEARNY 1.7 NNW, NJ US  40.774342 -74.137109

            elevation      _merge
      538         NaN   left_only
      526         NaN   left_only
      2215        NaN   left_only
      2872        NaN   left_only
      3650       29.0  right_only
      3651       25.6  right_only
```

These joins are equivalent to their SQL counterparts. Below is the inner join. Note that to use equals() you will have to do some manipulation of the dataframes to line them up

[27]:

```
import sqlite3
with sqlite3.connect('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
```

[27]:

```
True
```

[29]:

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

[29]:

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

                    inclement_weather
date
2018-01-01T00:00:00               NaN
2018-01-02T00:00:00             False
2018-01-03T00:00:00             False
2018-01-04T00:00:00              True
2018-01-05T00:00:00               NaN
```

[30]:

```
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'])
```

[31]:

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

[31]:

```
                     PRCP_x  SNOW_x  TMAX  TMIN  TOBS inclement_weather_x  \
date
2018-01-30T00:00:00     0.0     0.0   6.7  -1.7  -0.6               False
2018-03-08T00:00:00    48.8     NaN   1.1  -0.6   1.1               False
2018-03-13T00:00:00     4.1    51.0   5.6  -3.9   0.0                True
2018-03-21T00:00:00     0.0     0.0   2.8  -2.8   0.6               False
2018-04-02T00:00:00     9.1   127.0  12.8  -1.1  -1.1                True

                     PRCP_y  SNOW_y  WESF inclement_weather_y
```

```
date
2018-01-30T00:00:00     1.5    13.0   1.8                True
2018-03-08T00:00:00    28.4     NaN  28.7                 NaN
2018-03-13T00:00:00     3.0    13.0   3.0                True
2018-03-21T00:00:00     6.6   114.0   8.6                True
2018-04-02T00:00:00    14.0   152.0  15.2                True
```

[32]:
```
valid_station.merge(
station_with_wesf, left_index=True, right_index=True, suffixes=('', '_?')
).query('WESF > 0').head()
```

[32]:
```
                     PRCP   SNOW  TMAX  TMIN  TOBS inclement_weather  PRCP_?  \
date
2018-01-30T00:00:00   0.0    0.0   6.7  -1.7  -0.6             False     1.5
2018-03-08T00:00:00  48.8    NaN   1.1  -0.6   1.1             False    28.4
2018-03-13T00:00:00   4.1   51.0   5.6  -3.9   0.0              True     3.0
2018-03-21T00:00:00   0.0    0.0   2.8  -2.8   0.6             False     6.6
2018-04-02T00:00:00   9.1  127.0  12.8  -1.1  -1.1              True    14.0

                     SNOW_?  WESF inclement_weather_?
date
2018-01-30T00:00:00    13.0   1.8                True
2018-03-08T00:00:00     NaN  28.7                 NaN
2018-03-13T00:00:00    13.0   3.0                True
2018-03-21T00:00:00   114.0   8.6                True
2018-04-02T00:00:00   152.0  15.2                True
```

[33]:
```
valid_station.join(station_with_wesf, rsuffix='_?').query('WESF > 0').head()
```

[33]:
```
                     PRCP   SNOW  TMAX  TMIN  TOBS inclement_weather  PRCP_?  \
date
2018-01-30T00:00:00   0.0    0.0   6.7  -1.7  -0.6             False     1.5
2018-03-08T00:00:00  48.8    NaN   1.1  -0.6   1.1             False    28.4
2018-03-13T00:00:00   4.1   51.0   5.6  -3.9   0.0              True     3.0
2018-03-21T00:00:00   0.0    0.0   2.8  -2.8   0.6             False     6.6
2018-04-02T00:00:00   9.1  127.0  12.8  -1.1  -1.1              True    14.0

                     SNOW_?  WESF inclement_weather_?
date
2018-01-30T00:00:00    13.0   1.8                True
2018-03-08T00:00:00     NaN  28.7                 NaN
2018-03-13T00:00:00    13.0   3.0                True
2018-03-21T00:00:00   114.0   8.6                True
2018-04-02T00:00:00   152.0  15.2                True
```

[34]:
```
weather.set_index('station', inplace=True)
station_info.set_index('id', inplace=True)
```

[35]:
```
weather.index.intersection(station_info.index)
```

[35]:
```
Index(['GHCND:US1CTFR0039', 'GHCND:US1NJBG0015', 'GHCND:US1NJBG0017',
       'GHCND:US1NJBG0018', 'GHCND:US1NJBG0023', 'GHCND:US1NJBG0030',
       'GHCND:US1NJBG0039', 'GHCND:US1NJBG0003', 'GHCND:US1NJBG0044',
       'GHCND:US1NJES0018', 'GHCND:US1NJBG0010', 'GHCND:US1NJES0019',
       'GHCND:US1NJES0024', 'GHCND:US1NJBG0037'],
      dtype='object')
```

[36]:
```
weather.index.difference(station_info.index)
```

[36]:
```
Index([], dtype='object')
```

[37]:
```
station_info.index.difference(weather.index)
```

[37]:
```
Index(['GHCND:US1CTFR0022', 'GHCND:US1NJBG0001', 'GHCND:US1NJBG0002',
       'GHCND:US1NJBG0005', 'GHCND:US1NJBG0006', 'GHCND:US1NJBG0008',
       'GHCND:US1NJBG0011', 'GHCND:US1NJBG0012', 'GHCND:US1NJBG0013',
       'GHCND:US1NJBG0020',
       …
       'GHCND:USW00014708', 'GHCND:USW00014732', 'GHCND:USW00014734',
       'GHCND:USW00014786', 'GHCND:USW00054743', 'GHCND:USW00054787',
       'GHCND:USW00094728', 'GHCND:USW00094741', 'GHCND:USW00094745',
       'GHCND:USW00094789'],
      dtype='object', length=306)
```

[38]:
```
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]
```

[38]:
```
True
```

[42]:
```
weather.index.unique().union(station_info.index)
```

[42]:
```
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)
```

[41]:

```
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)
)
```

[41]:

```
True
```