Importing data into pandas

February 6, 2021

1 Importing data into pandas

There are tons of ways you can get data into a pandas dataframe. Here are a few of the more common ones.

First, let's import pandas `as` pd.

```
[22]: import pandas as pd
```

1.0.1 From a CSV file

If your data file is delimited with something other than a comma, you'll need to specify that in the `sep` argument. For example, if you had a pipe-delimited file: `pd.read_csv('../data/my-delimited-file.txt', sep='|')`

Let's read in the MLB salary data.

```
[23]: df_csv = pd.read_csv('../data/mlb.csv')
```

```
[24]: df_csv.head()
```

```
[24]:                NAME TEAM POS    SALARY  START_YEAR  END_YEAR  YEARS
       0   Clayton Kershaw  LAD  SP  33000000        2014      2020      7
       1      Zack Greinke  ARI  SP  31876966        2016      2021      6
       2       David Price  BOS  SP  30000000        2016      2022      7
       3    Miguel Cabrera  DET  1B  28000000        2014      2023     10
       4  Justin Verlander  DET  SP  28000000        2013      2019      7
```

1.0.2 From a CSV file on the Internet

Just pass in the URL. This example uses licensed child care facility data from Colorado's open data portal.

The values that get returned aren't live – like, if the results changed, your data frame would not update with new values. It reads in the data once.

[25]:
```
df_csv_internet = pd.read_csv('https://data.colorado.gov/api/views/a9rr-k8mu/
 ↪rows.csv?accessType=DOWNLOAD')
```

[26]:
```
df_csv_internet.head()
```

[26]:
```
   PROVIDER ID                        PROVIDER NAME  \
0      35597.0                         Rene Willard   
1    1670733.0        AUGUSTINE CLASSICAL PRESCHOOL   
2    1501661.0  DENVER BOTANIC GARDENS SUMMER CAMPS   
3    1685302.0                        AMANDA DUNCAN   
4       4855.0                   CCSD DRY CREEK ECS   

                PROVIDER SERVICE TYPE        STREET ADDRESS       CITY STATE  \
0  Experienced Family Child Care Home        1473 Walnut St    Windsor    CO   
1                   Child Care Center      480 S Kipling ST   Lakewood    CO   
2        School-Age Child Care Center          1005 York ST     Denver    CO   
3              Family Child Care Home        7131 W 75th PL     Arvada    CO   
4        School-Age Child Care Center  7686 E Hindsdale AVE  Englewood    CO   

     ZIP     COUNTY  COMMUNITY                                     ECC  ...  \
0  80550       Weld        NaN                   Promises for Children  ...   
1  80226  Jefferson        NaN           Triad Early Childhood Council  ...   
2  80206     Denver        NaN        Denver's Early Childhood Council  ...   
3  80003  Jefferson        NaN           Triad Early Childhood Council  ...   
4  80112   Arapahoe        NaN  Arapahoe County Early Childhood Council  ...   

  CCCAP TOTAL AUTH_D1 CCCAP FA STATUS_D1 CCCAP AMOUNT PAID_D1  \
0                 NaN              False                  NaN   
1                 NaN              False                  NaN   
2                 NaN              False                  NaN   
3                 NaN               True                  NaN   
4                 NaN               True                  NaN   

  CCCAP FA EXP DATE_D2  CCCAP TOTAL AUTH_D2  CCCAP FA STATUS_D2  \
0                  NaN                  NaN               False   
1                  NaN                  NaN               False   
2                  NaN                  NaN               False   
3           06/30/2022                  NaN                True   
4           06/30/2022                  NaN                True   

   LICENSE FEE DISCOUNT  LONG-LAT OPERATING STATUS (Self-Report)  \
0                   NaN       NaN                           Open   
1                   NaN       NaN                           Open   
2                   NaN       NaN                            NaN   
3                   NaN       NaN                            NaN   
4                   NaN       NaN                           Open   
```

```
  OPERATING STATUS REPORT DATE
0                   2020-07-31
1                   2020-08-11
2                          NaN
3                          NaN
4                   2020-06-30

[5 rows x 29 columns]
```

1.0.3 From an Excel file

To read an Excel file in pandas, use the `read_excel()` method. Depending on the filetype (`xls` or `xlsx`), you'd also need to separately install into your virtual environment the `xlrd` or `openpyxl` modules. (We've already installed both here.)

You might also want to specify the `sheet_name` to select your worksheet of interest – the default is "the first one."

Here, we're reading in a spreadsheet with data on accidental drug overdoses in Connecticut.

[27]:
```
df_xl = pd.read_excel('../data/CT_Overdoses_2012-2016.xlsx',␣
 ↪sheet_name='Accidental_Drug_Related_Deaths_')
```

[28]:
```
df_xl.head()
```

[28]:
```
  CaseNumber       Date     Sex   Race   Age Residence City Residence State  \
0   13-16336 2013-11-09  Female  White  53.0         GROTON             NaN
1   12-18447 2012-12-29    Male  White  30.0        WOLCOTT             NaN
2    14-2758 2014-02-18    Male  White  43.0        ENFIELD             NaN
3   14-13497 2014-09-07  Female  White  24.0    WALLINGFORD             NaN
4   13-14421 2013-10-04  Female  White  26.0     WEST HAVEN             NaN

  Residence County   Death City Death State  … Benzodiazepine Methadone  \
0       NEW LONDON       GROTON         NaN  …              Y       NaN
1        NEW HAVEN    WATERBURY         NaN  …            NaN       NaN
2              NaN      ENFIELD         NaN  …              Y       NaN
3              NaN  WALLINGFORD         NaN  …            NaN       NaN
4        NEW HAVEN   WEST HAVEN         NaN  …            NaN       NaN

  Amphet Tramad Morphine (not heroin) Other Any Opioid MannerofDeath  \
0    NaN    NaN                   NaN   NaN        NaN      Accident
1    NaN    NaN                   NaN   NaN        NaN      Accident
2    NaN    NaN                   NaN   NaN        NaN      Accident
3    NaN    NaN                   NaN   NaN        NaN      Accident
4    NaN    NaN                   NaN   NaN        NaN      Accident

  AmendedMannerofDeath                    DeathLoc
```

```
0                   NaN   (41.343693, -72.07877)
1                   NaN  (41.554261, -73.043069)
2                   NaN  (41.976501, -72.591985)
3                   NaN  (41.454408, -72.818414)
4                   NaN  (41.272336, -72.949817)

[5 rows x 32 columns]
```

1.0.4 From a Python data collection

Maybe the work you're doing in pandas happens downstream of some other Python processing, so the data exists as a native Python data collection – say, a list of dictionaries. You can turn this (and other Python data collections, like a list of lists) into a pandas dataframe, too.

[29]:

```
test_data = [
    {'name': 'Cody Winchester', 'job': 'Training director', 'location':␣
 ↪'Colorado Springs, CO'},
    {'name': 'Guy Fieri', 'job': 'Gourmand', 'location': 'Flavortown'},
    {'name': 'Michael Bennet', 'job': 'Senator', 'location': 'Washington, D.C.'}
]
```

[30]:

```
df_py_lod = pd.DataFrame(data=test_data)
```

[31]:

```
df_py_lod.head()
```

[31]:

```
              name                job              location
0  Cody Winchester  Training director  Colorado Springs, CO
1        Guy Fieri           Gourmand            Flavortown
2   Michael Bennet            Senator      Washington, D.C.
```

If you have a list of lists, you would need to also specify the `columns` keyword argument, as well:

[32]:

```
test_data_ls = [
    ['Cody Winchester', 'Training director', 'Colorado Springs, CO'],
    ['Guy Fieri', 'Gourmand', 'Flavortown'],
    ['Michael Bennet', 'Senator', 'Washington, D.C']
]
```

[33]:

```
df_py_lol = pd.DataFrame(data=test_data_ls, columns=['name', 'job', 'location'])
```

[34]:

```
df_py_lol.head()
```

[34]:

```
              name                job              location
0  Cody Winchester  Training director  Colorado Springs, CO
1        Guy Fieri           Gourmand            Flavortown
2   Michael Bennet            Senator       Washington, D.C
```

1.0.5 From an HTML table

OK SO.

This one requires you to install and specify the Python package that has the HTML parsing engine of your choice – BeautifulSoup or lxml. The default is `lxml`, but here we're going to use BeautifulSoup.

Huge caveat! Pulling data directly from an HTML table can be hit and miss, depending on how hairy the underlying HTML is. And if you want to scrape data from a website, it's usually better practice to save the results to a local file, *then* load it up for analysis. But it's good to know that it's an option.

In this example, we've installed `BeautifulSoup` (alias `bs4`) and we're going to import a table of media witnesses to Texas death row executions.

We're going to pass four things to the pandas `read_html()` method: 1. The URL we want to scrape (in quotes, as a string) 2. The `flavor` of parser that we'd like to use to process the HTML (`bs4`) 3. The HTML attributes of the table we're targeting (in this case, the table has a `class` called `tdcj_table`) 4. The number of the list, in the list of lists that gets returned in a dataframe, that has the `header`? (Usually it's 0 – the first one)

Reading through the documentation for this method, we also notice that this method returns a *list* of matching tables as dataframes, so we need to grab the *first* item in this list of tables returned. Our arguments were specific enough that there's only one item in the returned list, though, so we can just grab the first item with `[0]`.

[35]:
```
html_df = pd.read_html('http://www.tdcj.state.tx.us/death_row/
 ↪dr_media_witness_list.html',
                        flavor='bs4',
                        attrs={'class': 'tdcj_table'},
                        header=0)[0]
```

[36]:
```
html_df.head()
```

[36]:
```
   Execution                Link Last Name First Name TDCJ Number       Date  \
0        570  Inmate Information   Wardlow      Billy      999137   7/8/2020
1        569  Inmate Information     Ochoa       Abel      999450   2/6/2020
2        568  Inmate Information   Gardner       John      999516  1/15/2020
3        567  Inmate Information   Runnels     Travis      999505 12/11/2019
4        566  Inmate Information      Hall     Justen      999497  11/6/2019

                                 Media Witness List
0  Michael Graczyk, Associated Press; Joseph Brow…
1  Michael Graczyk, Associated Press; Joseph Brow…
2  Michael Graczyk, Associated Press; Joseph Brow…
3  Michael Graczyk, Associated Press; Joseph Brow…
4  Michael Graczyk, Associated Press; Joseph Brow…
```

1.0.6 From JSON

JSON stands for JavaScript Object Notation. It's a common data interchange format on the web. The `read_json()` method can pull JSON into a data frame.

Pandas can slurp in data from a local `.json` file, or from a URL – say, a JSON API with data on dogs and cats registered in the Sunshine Coast Region of Australia. That one sounds fun! Let's do that.

[37]:

```
json_df = pd.read_json('https://data.sunshinecoast.qld.gov.au/resource/
 ↪44qj-t4fr.json')
```

[38]:

```
json_df.head()
```

[38]:

```
   animaltype              name specificbreed primarybreed primarycolour  \
0  D           UNKNOWN DOG NAME    POODLETOY        POODLE    Grey
1  D                       Emie    FOXTERRMN        FOXTER    BlackWhite
2  D                       Jess    MALTESE          MALTES    White
3  D                      Style    BICHONFRS        BICHON    White
4  D                      Benny    SPANIELCV        SPANIE    BlackWhite

  de_sexed gender  age      locality
0        Y      F   32      NINDERRY
1        Y      F   21      BURNSIDE
2        Y      M   17       BLI BLI
3        Y      F   23   SIPPY DOWNS
4        Y      M   12  MAROOCHYDORE
```