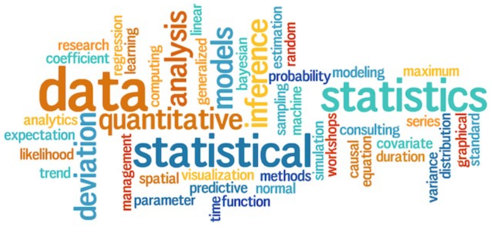
Applied Data Science

Session 2: Data Acquisition

Dr. Soharab Hossain Shaikh



1

Get Some Data

1. Directly download a data file (or files) manually

2. Query data from a database

3. Query an API (usually web-based, these days)

4. Scrap data from a webpage

2

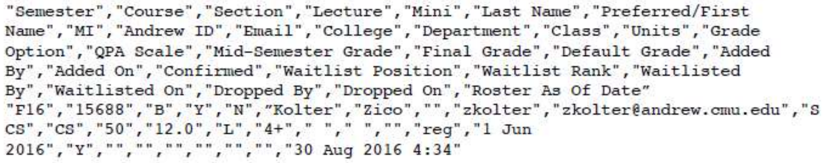
Collecting Data from Multiple Data Sources (File Formats)

- 1. CSV (Comma Separate Values)
- 2. XLS/XLSX (Microsoft Excel Spread Sheet)
- 3. JSON (Java-script Object Notation)
- 4. XML (eXtensible Markup Language)
- 5. HTML (HyperText Markup Language)

3

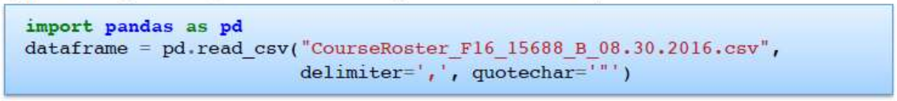
CSV File

Refers to any delimited text file (not always separated by commas)



```
"Semester","Course","Section","Lecture","Mini","Last Name","Preferred/First
Name","MI","Andrew ID","Email","College","Department","Class","Units","Grade
Option","QPA Scale","Mid-Semester Grade","Final Grade","Default Grade","Added
By","Added On","Confirmed","Waitlist Position","Waitlist Rank","Waitlisted
By","Waitlisted On","Dropped By","Dropped On","Roster As Of Date”
"F16","15688","B","Y","N",”Kolter","Zico","","zkolter","zkolter@andrew.cmu.edu","S
CS","CS","50","12.0","L","4+"," "," ","","reg","1 Jun
2016","Y","","","","","","","30 Aug 2016 4:34"
```

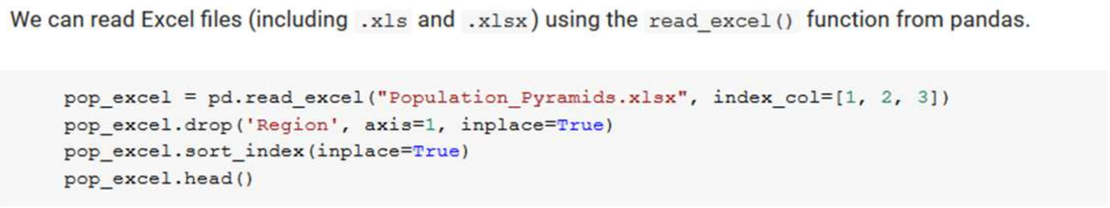
If values themselves contain commas, you can enclose them in quotes (our registrar apparently always does this, just to be safe)



```
import pandas as pd
dataframe = pd.read_csv("CourseRoster_F16_15688_B_08.30.2016.csv",
                        delimiter=',', quotechar='"')
```

4

XLSX File

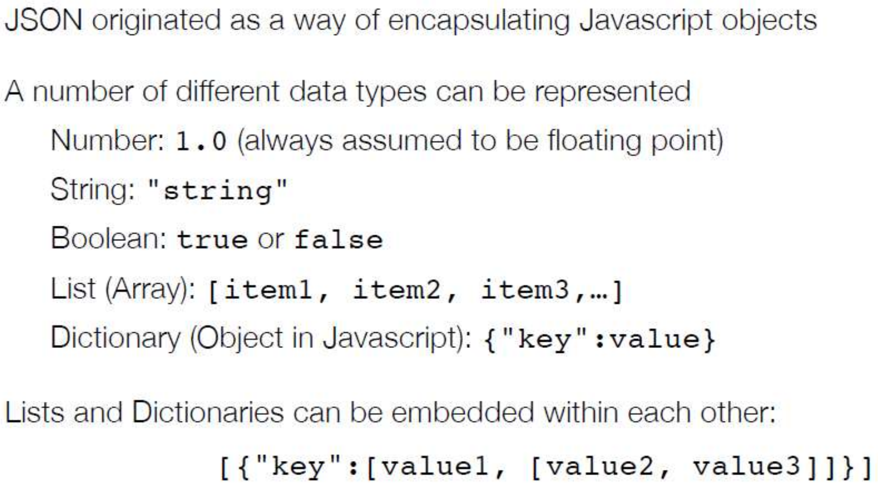


We can read Excel files (including `.xls` and `.xlsx`) using the `read_excel()` function from pandas.

```
pop_excel = pd.read_excel("Population_Pyramids.xlsx", index_col=[1, 2, 3])
pop_excel.drop('Region', axis=1, inplace=True)
pop_excel.sort_index(inplace=True)
pop_excel.head()
```

5

JSON



JSON originated as a way of encapsulating Javascript objects

A number of different data types can be represented

- Number: `1.0` (always assumed to be floating point)
- String: `"string"`
- Boolean: `true` or `false`
- List (Array): `[item1, item2, item3,…]`
- Dictionary (Object in Javascript): `{"key":value}`

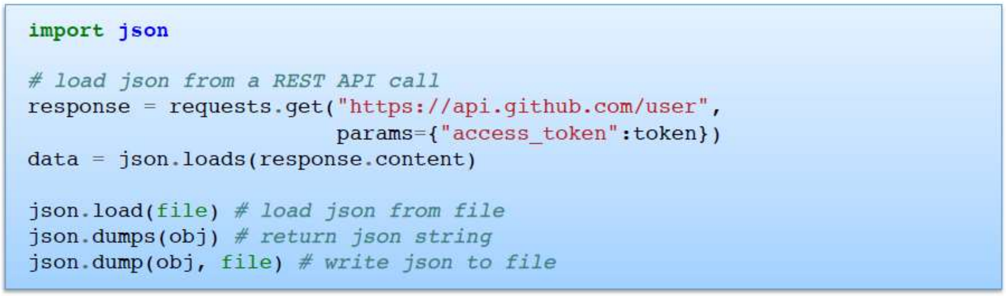
Lists and Dictionaries can be embedded within each other:

```
[{"key":[value1, [value2, value3]]}]
```

6

Parsing JSON in Python

Built-in library to read/write Python objects from/to JSON files



```
import json

# load json from a REST API call
response = requests.get("https://api.github.com/user",
                        params={"access_token":token})
data = json.loads(response.content)

json.load(file) # load json from file
json.dumps(obj) # return json string
json.dump(obj, file) # write json to file
```

7

XML/HTML Files

The main format for the web (though XML seems to be loosing a bit of popularity to JSON for use in APIs / file formats)

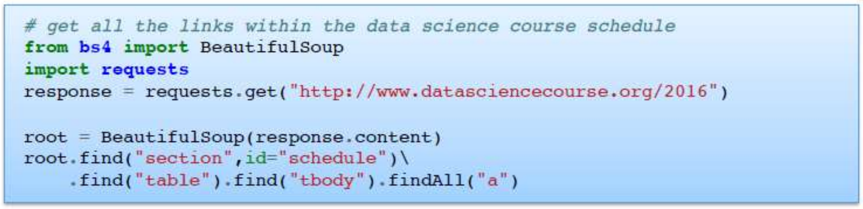
XML files contain hiearchical content delineated by tags

```
<tag attribute="value">
    <subtag>
        Some content for the subtag
    </subtag>
    <openclosetag attribute="value2”/>
</tag>
```

8

Parsing XML/HTML Files

There are a number of XML/HTML parsers for Python, but a nice one for data science is the BeautifulSoup library (specifically focused on getting data out of XML/HTML files)



```
# get all the links within the data science course schedule
from bs4 import BeautifulSoup
import requests
response = requests.get("http://www.datasciencecourse.org/2016")

root = BeautifulSoup(response.content)
root.find("section",id="schedule")\
    .find("table").find("tbody").findAll("a")
```

9

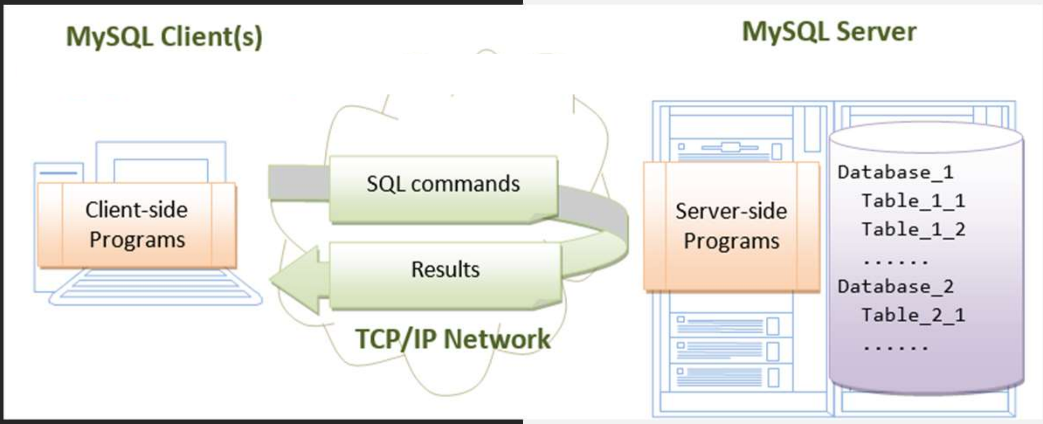
Storing and Retrieving Data in/from Database

10

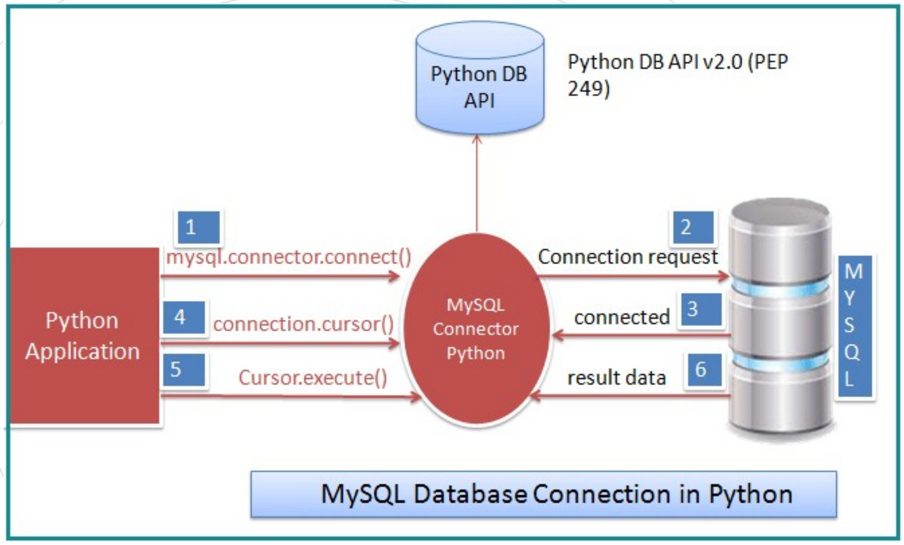
Store Data in a Database

11

MYSQL Client-Server Computing



12



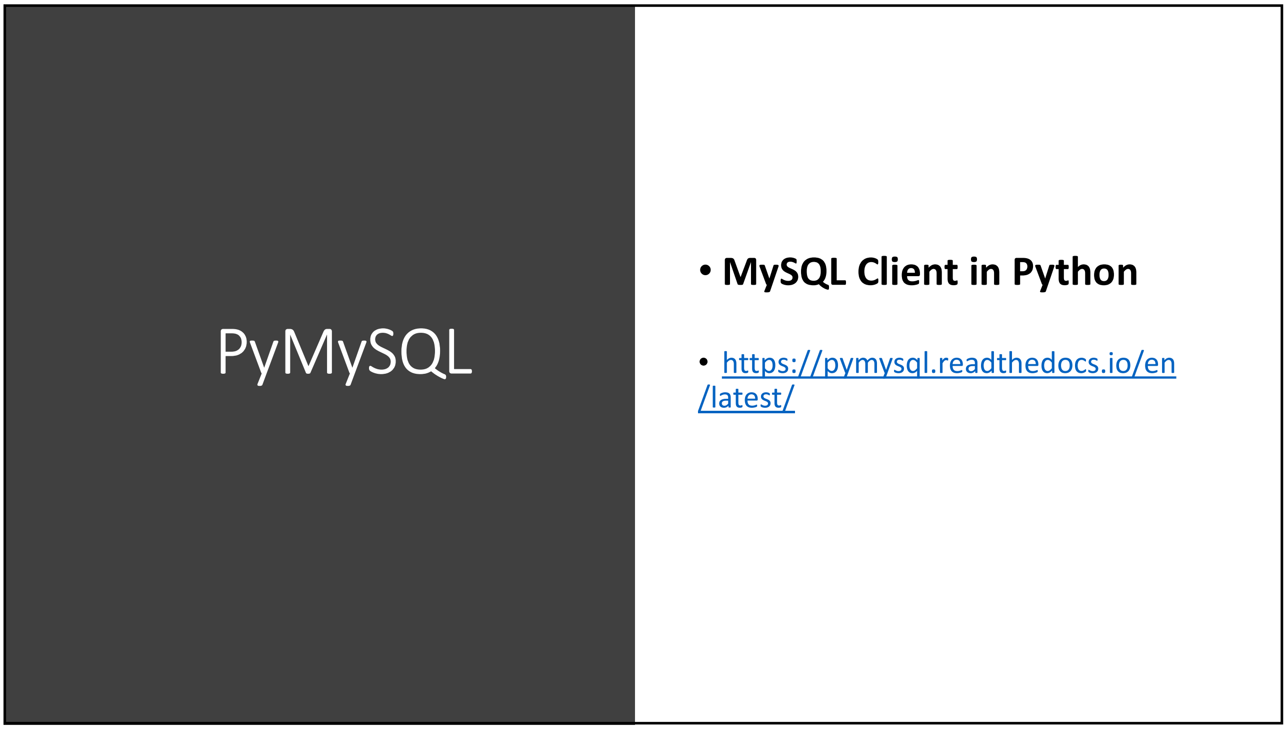
MySQL Database Connection in Python

Python – MYSQL Interaction

Cursor Object

We need to create the object of a class called cursor that allows Python code to execute database command in a database session.

- Cursors are created by the **connection.cursor()** method: they are bound to the connection for the entire lifetime and all the commands are executed in the context of the database session wrapped by the connection.



PyMySQL

- **MySQL Client in Python**
- https://pymysql.readthedocs.io/en/latest/

15

MariaDB : Free Open Source Relational Database

- Official Website:
- https://mariadb.org/
- Good Primer:
- https://mariadb.com/kb/en/a-mariadb-primer/

16

References

PyMySQL

CRUD: https://pymysql.readthedocs.io/en/latest/user/examples.html

CONNECTION-Object:
https://pymysql.readthedocs.io/en/latest/modules/connections.html

CURSOR-object:
https://pymysql.readthedocs.io/en/latest/modules/cursors.html?highlight=cursor

Pandas- DataFrame-SQL:
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.to_sql.html

17

18

Go to the Coding Demo……….

19

To be continued in the next session…..

20