INFO-UB 23: Introduction to Programming and Data Science

Katherine Hoffmann Pham

July 11, 2018

NYU Stern, Department of Information Systems

Project advice

General advice:

1. Build projects with scope that can easily scale up or down
2. Expect to spend a lot of your time on data cleaning and merging
3. Start with description, then move to prediction
4. Write up your project as you work

Getting data

Data sources

Some good places to look for data:

1. Aggregate lists of datasets (GitHub page, class website)
2. ProgrammeableWeb API directory
3. Kaggle competition datasets
4. Also: Github repositories, data science blog posts

Obtaining data

Typical means of obtaining data include:

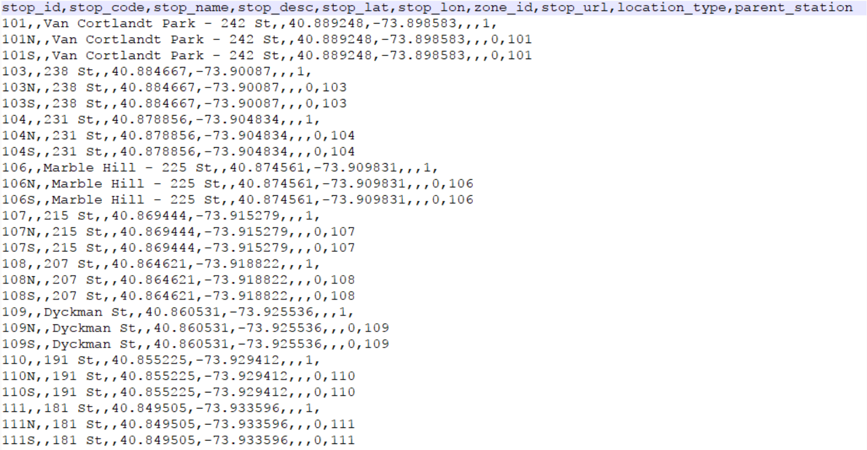
1. Downloads in tabular format
2. Queries posted to APIs
3. Web scraping

Obtaining data: tabular formats

1. Datasets are often available for download in tabular format: (typically, using delimiters)

   - Excel workbooks
   - .csv = comma-separated values
   - .tsv = tab-separated values

.csv example



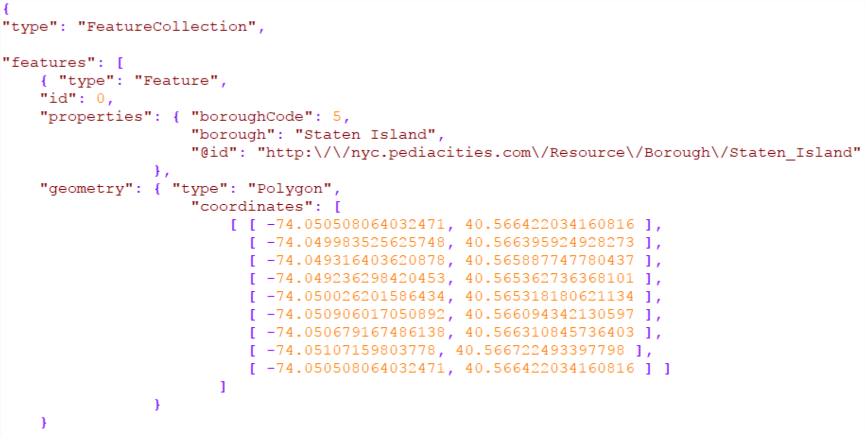
```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon,zone_id,stop_url,location_type,parent_station
101,,Van Cortlandt Park - 242 St,,40.889248,-73.898583,,,1,
101N,,Van Cortlandt Park - 242 St,,40.889248,-73.898583,,,0,101
101S,,Van Cortlandt Park - 242 St,,40.889248,-73.898583,,,0,101
103,,238 St,,40.884667,-73.90087,,,1,
103N,,238 St,,40.884667,-73.90087,,,0,103
103S,,238 St,,40.884667,-73.90087,,,0,103
104,,231 St,,40.878856,-73.904834,,,1,
104N,,231 St,,40.878856,-73.904834,,,0,104
104S,,231 St,,40.878856,-73.904834,,,0,104
106,,Marble Hill - 225 St,,40.874561,-73.909831,,,1,
106N,,Marble Hill - 225 St,,40.874561,-73.909831,,,0,106
106S,,Marble Hill - 225 St,,40.874561,-73.909831,,,0,106
107,,215 St,,40.869444,-73.915279,,,1,
107N,,215 St,,40.869444,-73.915279,,,0,107
107S,,215 St,,40.869444,-73.915279,,,0,107
108,,207 St,,40.864621,-73.918822,,,1,
108N,,207 St,,40.864621,-73.918822,,,0,108
108S,,207 St,,40.864621,-73.918822,,,0,108
109,,Dyckman St,,40.860531,-73.925536,,,1,
109N,,Dyckman St,,40.860531,-73.925536,,,0,109
109S,,Dyckman St,,40.860531,-73.925536,,,0,109
110,,191 St,,40.855225,-73.929412,,,1,
110N,,191 St,,40.855225,-73.929412,,,0,110
110S,,191 St,,40.855225,-73.929412,,,0,110
111,,181 St,,40.849505,-73.933596,,,1,
111N,,181 St,,40.849505,-73.933596,,,0,111
111S,,181 St,,40.849505,-73.933596,,,0,111
```

Obtaining data: APIs

2. A popular (programmer-friendly) alternative is APIs, (i.e. Application Programming Interfaces)

- Websites volunteer their data in a structured way
- Rate limits (and costs) often apply
- Typically, results are returned using:
  - .json = JavaScript Object Notation; flexible, dictionary-like
  - .xml = Extensible Markup Language; flexible, HTML-like
  - .txt = plain text
  - Sometimes, custom Python packages

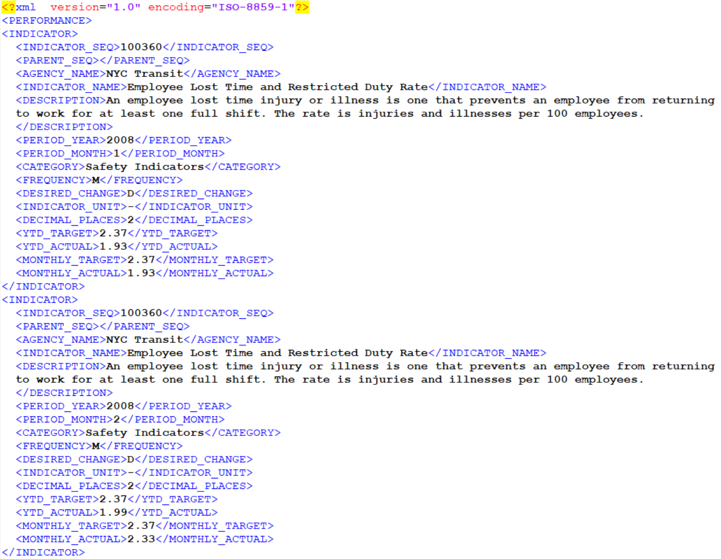
.json example



```
{
"type": "FeatureCollection",

"features": [
    { "type": "Feature",
    "id": 0,
    "properties": { "boroughCode": 5,
                    "borough": "Staten Island",
                    "@id": "http:\/\/nyc.pediacities.com\/Resource\/Borough\/Staten_Island"
                },
    "geometry": { "type": "Polygon",
                    "coordinates": [
                        [ [ -74.050508064032471, 40.566422034160816 ],
                          [ -74.049983525625748, 40.566395924928273 ],
                          [ -74.049316403620878, 40.565887747780437 ],
                          [ -74.049236298420453, 40.565362736368101 ],
                          [ -74.050026201586434, 40.565318180621134 ],
                          [ -74.050906017050892, 40.566094342130597 ],
                          [ -74.050679167486138, 40.566310845736403 ],
                          [ -74.05107159803778, 40.566722493397798 ],
                          [ -74.050508064032471, 40.566422034160816 ] ]
                       ]
                }
    }
```

.xml example



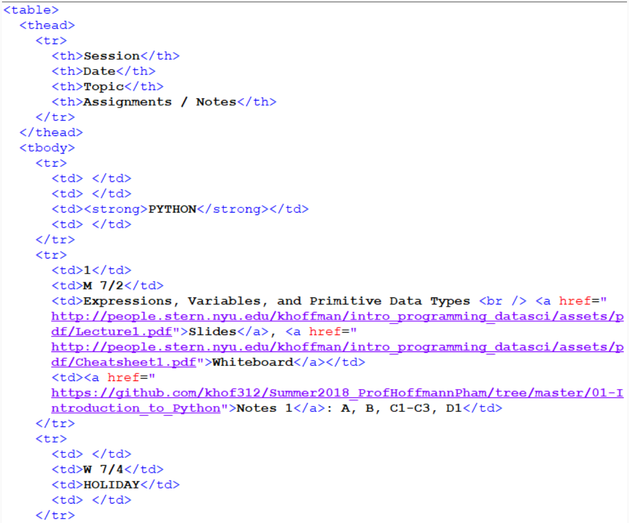
```
<?xml  version="1.0" encoding="ISO-8859-1"?>
<PERFORMANCE>
<INDICATOR>
  <INDICATOR_SEQ>100360</INDICATOR_SEQ>
  <PARENT_SEQ></PARENT_SEQ>
  <AGENCY_NAME>NYC Transit</AGENCY_NAME>
  <INDICATOR_NAME>Employee Lost Time and Restricted Duty Rate</INDICATOR_NAME>
  <DESCRIPTION>An employee lost time injury or illness is one that prevents an employee from returning
  to work for at least one full shift. The rate is injuries and illnesses per 100 employees.
  </DESCRIPTION>
  <PERIOD_YEAR>2008</PERIOD_YEAR>
  <PERIOD_MONTH>1</PERIOD_MONTH>
  <CATEGORY>Safety Indicators</CATEGORY>
  <FREQUENCY>M</FREQUENCY>
  <DESIRED_CHANGE>D</DESIRED_CHANGE>
  <INDICATOR_UNIT>-</INDICATOR_UNIT>
  <DECIMAL_PLACES>2</DECIMAL_PLACES>
  <YTD_TARGET>2.37</YTD_TARGET>
  <YTD_ACTUAL>1.93</YTD_ACTUAL>
  <MONTHLY_TARGET>2.37</MONTHLY_TARGET>
  <MONTHLY_ACTUAL>1.93</MONTHLY_ACTUAL>
</INDICATOR>
<INDICATOR>
  <INDICATOR_SEQ>100360</INDICATOR_SEQ>
  <PARENT_SEQ></PARENT_SEQ>
  <AGENCY_NAME>NYC Transit</AGENCY_NAME>
  <INDICATOR_NAME>Employee Lost Time and Restricted Duty Rate</INDICATOR_NAME>
  <DESCRIPTION>An employee lost time injury or illness is one that prevents an employee from returning
  to work for at least one full shift. The rate is injuries and illnesses per 100 employees.
  </DESCRIPTION>
  <PERIOD_YEAR>2008</PERIOD_YEAR>
  <PERIOD_MONTH>2</PERIOD_MONTH>
  <CATEGORY>Safety Indicators</CATEGORY>
  <FREQUENCY>M</FREQUENCY>
  <DESIRED_CHANGE>D</DESIRED_CHANGE>
  <INDICATOR_UNIT>-</INDICATOR_UNIT>
  <DECIMAL_PLACES>2</DECIMAL_PLACES>
  <YTD_TARGET>2.37</YTD_TARGET>
  <YTD_ACTUAL>1.99</YTD_ACTUAL>
  <MONTHLY_TARGET>2.37</MONTHLY_TARGET>
  <MONTHLY_ACTUAL>2.33</MONTHLY_ACTUAL>
</INDICATOR>
```

Obtaining data: Web scraping

3. A more involved alternative is to use web scraping

   - Basic idea: try to extract elements of the page
   - Collect and parse the page source, typically:
     - .html: Hypertext markup language
     - e.g. with Python's BeautifulSoup library
   - Easily scraped pages:
     - Are well-organized
     - Have a simple structure
     - Include few dynamic elements

.html example



```
<table>
  <thead>
    <tr>
      <th>Session</th>
      <th>Date</th>
      <th>Topic</th>
      <th>Assignments / Notes</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> </td>
      <td> </td>
      <td><strong>PYTHON</strong></td>
      <td> </td>
    </tr>
    <tr>
      <td>1</td>
      <td>M 7/2</td>
      <td>Expressions, Variables, and Primitive Data Types <br /> <a href="
      http://people.stern.nyu.edu/khoffman/intro_programming_datasci/assets/p
      df/Lecture1.pdf">Slides</a>, <a href="
      http://people.stern.nyu.edu/khoffman/intro_programming_datasci/assets/p
      df/Cheatsheet1.pdf">Whiteboard</a></td>
      <td><a href="
      https://github.com/khof312/Summer2018_ProfHoffmannPham/tree/master/01-I
      ntroduction_to_Python">Notes 1</a>: A, B, C1-C3, D1</td>
    </tr>
    <tr>
      <td> </td>
      <td>W 7/4</td>
      <td>HOLIDAY</td>
      <td> </td>
    </tr>
```