Open Healthcare Data and Tools in Practice

Janos G. Hajagos, PhD
Chief of Data Analytics
Research Assistant Professor
Dept. Biomedical Informatics
Stony Brook University
@jhajagos

AHIMA Data Institute: Making Information Meaningful
Las Vegas, Nevada
12/9/2016
Still Literally True in 2016!

https://www.ted.com/talks/eric_dishman_take_health_care_off_the_mainframe
The examples shown here build on open health data and open source tools

https://www.flickr.com/photos/122127718@N08/2040238088
https://www.flickr.com/photos/arbre_evolution/8286785236/
The legacy way: analytic software

If you were using their product for just internal uses and they charge you a big up front cost followed by about 28% of that amount as an ongoing license fee every subsequent year (it used to be 50%).

Their basic windows Analytics package costs $8,700 for the first year. It includes BASE, STAT, and GRAPH products for basic data processing, advanced statistics, and automated production graphics. You will need to pay a chunk extra for modules to access databases directly or using ODBC. See Order SAS® Software

For more than one user or versions that run on Linux, Unix (or mainframes) expect to pay more. License fees can easily reach past $100,000/year. Even a single user license for tools like Enterprise Miner can cost something like $140,000 for the first year.

Generally you need to speak to someone from SAS to get a quote for more complicated licenses. The SAS sales folks are exceedingly arrogant in many cases and you might find yourself lucky to even get them to spend the time to give you a quote. As a consultant, I tried to get a quote so my client at Morgan Stanley could order the product, and it was a multi-day challenge to get a response. At other times with less impressive sounding clients, I had a hard time even getting a response.
The open way: software

• Builds on long term investment of open source tools
  • BSD, GNU, Linux kernel, R, Python, LAPACK
• International community of developers from commercial, academic, and government stakeholders
• Collaborative internet tools are used to coordinate development
  • Git and Github
• Source code is made available
Example: OHDSI Software Stack

http://www.ohdsi.org/
NO TRESPASSING
WETLANDS REST

Digging No Shell-Fish
No Entry Within 300 of Trespassing Will Result
$500 Fine Plus Court
B Ordinance 140.110
2 Hour Day Nigh Camp
The legacy way: healthcare data
The open way: Data portal
Example 1: Enriching your local patient data

<table>
<thead>
<tr>
<th>patient_id</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>100 Mains Street, Springfield, MA 01103</td>
</tr>
<tr>
<td>1022</td>
<td>12 Oak Drive, Springfield, MA 01105</td>
</tr>
<tr>
<td>3033</td>
<td>1001 East Main St., Greenfield, MA 01301</td>
</tr>
<tr>
<td>4010</td>
<td>101 Route 9a, Deerfield, MA 01342</td>
</tr>
</tbody>
</table>

Going beyond the zip code
11746 – Huntington Station and Dix Hills
Two Census-Designated Places
Understanding socio-economic determinants health for your patients

Google Map’s Street View
Combining with Tiger Shape Files

AT THE BLOCK GROUP LEVEL
SNAP By Geographic Region

B22010: RECEIPT OF FOOD STAMPS IN THE PAST 12 MONTHS BY DISABILITY STATUS FOR HOUSEHOLDS
B22010: RECEIPT OF FOOD STAMPS IN THE PAST 12 MONTHS BY DISABILITY STATUS FOR HOUSEHOLDS – Percent of households who received Food Stamps/SNAP in the past 12 months
Using the PostGis Tiger Geocoder in PostGreSQL

```
SELECT (tt.geo).geomout, (tt.geo).rating,
     ST_Y((tt.geo).geomout) as latitude,
     ST_X((tt.geo).geomout) as longitude,
     tiger.pprint_addy((tt.geo).addy) as matched_address, (tt.geo).addy.zip as matched_zip5
FROM (select tiger.Geocode('?? Suncrest Dr., Dix Hills, NY 11746', 1) as geo) tt;
```
Geocoding Results
Example 1: Data sources and tools

- PostgreSQL - https://www.postgresql.org/
- PostGIS and Tiger Geocoder - http://www.postgis.net/
- Shape files - https://www.census.gov/geo/maps-data/data/data/tiger-line.html
- Python Code for extracting ACS variables - https://github.com/jhajagos/CensusGeographyTools
Example 2 - Hospital market analysis – Bakersfield, CA

https://www.flickr.com/photos/27326512@N00/329820320/

https://www.flickr.com/photos/rheinitz/8668769226/
Medicare Teaming Data

https://questions.cms.gov/faq.php?faqId=7977

<table>
<thead>
<tr>
<th>Physician Shared Patient Patterns -</th>
<th>30 day interval</th>
<th>60 day interval</th>
<th>90 day interval</th>
<th>180 day interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For “Big Open Data”* we don’t need to invest in an expensive HIPAA compliant environment

---

*Bigger than supported by a standard business class laptop or desktop or Excel
Gephi Visualization of the teaming data
Eigenvector centrality

• Centrality measures the importance of the node in the network
• Ranks importance of a node (provider) in the network
• Google’s PageRank is a variant of this metric
• A higher page rank in search indicates a more relevant search but in a teaming network does not imply that the physician is of high clinical quality
Example 2: Data sources and tools

- NetworkX
- SQLAlchemy
- MySQL
- Gephi 0.9.1
- ETL – load scripts for NPPES transformation script and teaming table load
- NPPES data: http://download.cms.gov/nppes/NPI_Files.html
- Teaming data: https://questions.cms.gov/faq.php?faqId=7977
Educating Data Scientists to work with healthcare data
Jupyter Notebooks for Analytic Reproducible Analysis

Next we will build the dataframes and corresponding matrices to compute the euclidean distance between two Medicare prescribers in DC. Rather than creating a crosstab by "drug_name" we will use the "generic_name". The branded drug Abilify has the generic name of ARIPiprazole. A great tool for understanding the relationship between active ingredients, generic name and branded products is RxNav.

In [29]: npi_generic_cross_df = pd.crosstab(prescriber_df["npi"], prescriber_df["generic_name"], values=prescriber_df["total_claim_count"],
Example 3: Kidney transplants in NY State

https://www.flickr.com/photos/tareqsalahuddin/7272346858/
Welcome to Health Data NY!

Explore the Health Data NY catalog! See all of the available data assets plus keywords, documentation explaining the data (meta data) and much more by clicking here.
<table>
<thead>
<tr>
<th>Patient Disposition</th>
<th>Discharge Year</th>
<th>Discharge Day of Week</th>
<th>CCS Diagnostic Code</th>
<th>CCS Diagnosis Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>WED</td>
<td>197</td>
<td>SKINSUBCUT TISS INFECT</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>WED</td>
<td>146</td>
<td>DIVERTICUL-OS/SITIS</td>
</tr>
<tr>
<td>Short-term Hospital</td>
<td>2014</td>
<td>SUN</td>
<td>050</td>
<td>DIABETES W/COMPL</td>
</tr>
<tr>
<td>Short-term Hospital</td>
<td>2014</td>
<td>SAT</td>
<td>154</td>
<td>GASTROENTRIS NONINFCT</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>WED</td>
<td>124</td>
<td>TONSILLITIS</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>MON</td>
<td>123</td>
<td>INFLUENZA</td>
</tr>
<tr>
<td>Skilled Nursing Home</td>
<td>2014</td>
<td>WED</td>
<td>122</td>
<td>PNEUMONIA</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>MON</td>
<td>123</td>
<td>INFLUENZA</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>WED</td>
<td>123</td>
<td>INFLUENZA</td>
</tr>
<tr>
<td>Hosp Basd Medicare Approved Swing Bed</td>
<td>2014</td>
<td>FRI</td>
<td>231</td>
<td>OTHER FRACTURE</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>TUE</td>
<td>123</td>
<td>INFLUENZA</td>
</tr>
<tr>
<td>Hosp Basd Medicare Approved Swing Bed</td>
<td>2014</td>
<td>MON</td>
<td>122</td>
<td>PNEUMONIA</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>SAT</td>
<td>122</td>
<td>PNEUMONIA</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>FRI</td>
<td>155</td>
<td>OTHER GI DISORDER</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>THU</td>
<td>122</td>
<td>PNEUMONIA</td>
</tr>
<tr>
<td>Home or Self Care</td>
<td>2014</td>
<td>MON</td>
<td>127</td>
<td>COPD</td>
</tr>
</tbody>
</table>
SOCRATA API with pandas library

```python
In [16]:
print(kt_url)
```


Once a URL is constructed to a data source a GET request over HTTP (HyperText Transfer Protocol) can be executed. The HTTP protocol is how most data is transferred from a host/server to the client. Here the client is not a web browser but the Python kernel running on your computer.

Each data source in the Socrata environment is identified by a short string or data tag. The SPARCS 2014 data tag is "rmwa-zns4". In the above request we are asking for a JSON document. JSON or Javascript Object Notation is a text based format for exchanging data in a machine readable format. One can think of JSON as a CSV format for the Internet Era. The Pandas' library function "read_json()" can take a URL and makes a remote call to the Socrata server and read the response. If the URL is misspecified than an error will occur. It converts the JSON response into a dataframe. Pandas' dataframes are powerful constructs for working with table based data.

```python
In [17]:
kidney_transplants_df = pd.read_json(kt_url)
```

```python
In [18]:
len(kidney_transplants_df.length_of_stay)
```

Out[18]: 1169
In [31]: kidney_transplants_df.groupby(['facility_name_with_id'])['length_of_stay'].count()

Out[31]:

<table>
<thead>
<tr>
<th>facility_name_with_id</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001 - Albany Medical Center Hospital</td>
<td>55</td>
</tr>
<tr>
<td>0210 - Erie County Medical Center</td>
<td>74</td>
</tr>
<tr>
<td>0245 - University Hospital</td>
<td>59</td>
</tr>
<tr>
<td>0413 - Strong Memorial Hospital</td>
<td>60</td>
</tr>
<tr>
<td>0541 - North Shore University Hospital</td>
<td>29</td>
</tr>
<tr>
<td>0635 - University Hospital SUNY Health Science Center</td>
<td>64</td>
</tr>
<tr>
<td>1139 - Westchester Medical Center</td>
<td>27</td>
</tr>
<tr>
<td>1169 - Montefiore Medical Center - Henry &amp; Lucy Moses Div</td>
<td>155</td>
</tr>
<tr>
<td>1320 - University Hospital of Brooklyn</td>
<td>23</td>
</tr>
<tr>
<td>1456 - Mount Sinai Hospital</td>
<td>162</td>
</tr>
<tr>
<td>1458 - New York Presbyterian Hospital - New York Weill Cornell Center</td>
<td>211</td>
</tr>
<tr>
<td>1463 - NYU Hospitals Center</td>
<td>25</td>
</tr>
<tr>
<td>1464 - New York Presbyterian Hospital - Columbia Presbyterian Center</td>
<td>225</td>
</tr>
</tbody>
</table>

Name: length_of_stay, dtype: int64

In [30]: kidney_transplants_df.groupby(['facility_name_with_id'])['length_of_stay'].mean()

Out[30]:

<table>
<thead>
<tr>
<th>facility_name_with_id</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001 - Albany Medical Center Hospital</td>
<td>9.327273</td>
</tr>
<tr>
<td>0210 - Erie County Medical Center</td>
<td>6.148649</td>
</tr>
<tr>
<td>0245 - University Hospital</td>
<td>6.762712</td>
</tr>
<tr>
<td>0413 - Strong Memorial Hospital</td>
<td>9.250000</td>
</tr>
<tr>
<td>0541 - North Shore University Hospital</td>
<td>6.137931</td>
</tr>
<tr>
<td>0635 - University Hospital SUNY Health Science Center</td>
<td>5.265625</td>
</tr>
<tr>
<td>1139 - Westchester Medical Center</td>
<td>8.703704</td>
</tr>
<tr>
<td>1169 - Montefiore Medical Center - Henry &amp; Lucy Moses Div</td>
<td>6.477419</td>
</tr>
<tr>
<td>1320 - University Hospital of Brooklyn</td>
<td>10.521739</td>
</tr>
<tr>
<td>1456 - Mount Sinai Hospital</td>
<td>7.228395</td>
</tr>
<tr>
<td>1458 - New York Presbyterian Hospital - New York Weill Cornell Center</td>
<td>5.246445</td>
</tr>
<tr>
<td>1463 - NYU Hospitals Center</td>
<td>5.950000</td>
</tr>
<tr>
<td>1464 - New York Presbyterian Hospital - Columbia Presbyterian Center</td>
<td>7.235556</td>
</tr>
</tbody>
</table>

Name: length_of_stay, dtype: float64
In [27]: kidney_transplants_outliers_removed_df = kidney_transplants_df.where(kidney_transplants_df["length_of_stay"] <= 40)

In [28]: sb.violinplot(x="facility_id", y="length_of_stay", data=kidney_transplants_outliers_removed_df)

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0xcdc74940>
Example 3: Data sources and tools

- Anaconda Python distribution - [https://www.continuum.io/downloads](https://www.continuum.io/downloads)
- Seaborn library - [http://seaborn.pydata.org/](http://seaborn.pydata.org/)
- pandas - [http://pandas.pydata.org/](http://pandas.pydata.org/)
- Socrata API - [https://dev.socrata.com/consumers/getting-started.html](https://dev.socrata.com/consumers/getting-started.html)
Example 4: Psychiatric drug prescribers in D.C.

https://www.flickr.com/photos/51274664@N06/6930338021/
Medicare prescribing data
Can we build a distance metric to find similar prescribers

Prescriber 1 = (0,0,0,0,1,0,0,1,0,0,0,1,1)
Prescriber 2 = (0,0,0,0,1,0,0,1,0,0,0,1,0)
Prescriber 3 = (1,1,1,0,0,0,0,0,0,0,0,0,1)

Where $i^{th}$ entry indicates whether the prescriber prescribes Drug $i$

Euclidean distance between providers:

Prescriber 1 and 2 is $\sqrt{1} = 1$
Prescriber 1 and 3 is $\sqrt{6} = 2.44$
Prescriber 2 and 3 is $\sqrt{7} = 2.65$
Prescriber Distance Matrix

White to Black – Small distance to big distance
A slice of the distance matrix

Sorted by increasing distance

Pie image: https://www.flickr.com/photos/aloha75/5953100136/
Sorted list of NPIs with increasing Rx distance

```
In [51]: providers_sorted = np.lexsort((prescriber_dist[:,2010].tolist(),))

In [52]: prescriber_specialty_generic_df.iloc[:,0:2].as_matrix()[providers_sorted[0:40], :]
```

```
Out[52]: array([[1487818670L, u'Psychiatry'],
              [1154970076L, u'Psychiatry & Neurology'],
              [1588810162L, u'Psychiatry & Neurology'],
              [1265692115L, u'Psychiatry'],
              [1366766263L, u'Psychiatry & Neurology'],
              [1285815878L, u'Psychiatry'],
              [1992965537L, u'Certified Clinical Nurse Specialist'],
              [1750616645L, u'Psychiatry'],
              [1366618746L, u'Psychiatry'],
              [1801919659L, u'Certified Clinical Nurse Specialist'],
              [1720241011L, u'Neuropsychiatry'],
              [1326086125L, u'Psychiatry'],
              [1790964948L, u'Psychiatry & Neurology'],
              [1033322730L, u'Psychiatry'],
              [1023267606L, u'Psychiatry'],
              [1780766688L, u'Psychiatry & Neurology'],
              [1316089642L, u'Psychiatry'],
              [1164621363L, u'Psychiatry & Neurology'],
              [1184876948L, u'Psychiatry'],
              [1821259581L, u'Psychiatry'],
              [1992746515L, u'Psychiatry'],
              [1821073776L, u'Psychiatry'],
              [1770868796L, u'Nurse Practitioner'],
```

43
The NPI is the key to rich provider data
Example 4: Data sources and tools


• Medicare Prescriber data - https://data.cms.gov/Public-Use-Files/Medicare-Provider-Utilization-and-Payment-Data-201/4uvc-gbfz