Sources,Datasets,Attributes,BigData

August 23, 2020

[1]:
```
pip install sshtunnel
```

```
Requirement already satisfied: sshtunnel in /opt/conda/lib/python3.7/site-
packages (0.1.5)
Requirement already satisfied: paramiko>=1.15.2 in
/opt/conda/lib/python3.7/site-packages (from sshtunnel) (2.7.1)
Requirement already satisfied: pynacl>=1.0.1 in /opt/conda/lib/python3.7/site-
packages (from paramiko>=1.15.2->sshtunnel) (1.4.0)
Requirement already satisfied: cryptography>=2.5 in
/opt/conda/lib/python3.7/site-packages (from paramiko>=1.15.2->sshtunnel) (2.7)
Requirement already satisfied: bcrypt>=3.1.3 in /opt/conda/lib/python3.7/site-
packages (from paramiko>=1.15.2->sshtunnel) (3.2.0)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from pynacl>=1.0.1->paramiko>=1.15.2->sshtunnel) (1.12.0)
Requirement already satisfied: cffi>=1.4.1 in /opt/conda/lib/python3.7/site-
packages (from pynacl>=1.0.1->paramiko>=1.15.2->sshtunnel) (1.12.3)
Requirement already satisfied: asn1crypto>=0.21.0 in
/opt/conda/lib/python3.7/site-packages (from
cryptography>=2.5->paramiko>=1.15.2->sshtunnel) (0.24.0)
Requirement already satisfied: pycparser in /opt/conda/lib/python3.7/site-
packages (from cffi>=1.4.1->pynacl>=1.0.1->paramiko>=1.15.2->sshtunnel) (2.19)
Note: you may need to restart the kernel to use updated packages.
```

0.1 Purpose:

- Demonstrate how to:
  - Get metadata database
  - See dataset sources
  - See current datasets
  - Get attribute look-up table for a specific dataset
  - Get data of a specific dataset from bigdata table
  - Example of how to use data to create graph

Imports / Settings / Connect to Mongodb

[57]:
```
#  ---------------
#   I M P O R T S
```

```
#  ---------------
from sshtunnel import SSHTunnelForwarder
import pymongo
import pprint

import pandas as pd

#  ------------------
#    S E T T I N G S
#  ------------------

pd.set_option('max_colwidth', 100)


#  ----------------
#   M O N G O D B
#  ----------------


MONGO_HOST = "128.206.117.150"
MONGO_USER = "haithcoatt"
MONGO_PASS = "Ke11ieJean"

server = SSHTunnelForwarder(
    MONGO_HOST,
    ssh_username=MONGO_USER,
    ssh_password=MONGO_PASS,
    remote_bind_address=('127.0.0.1', 27017)
)

server.start()
client = pymongo.MongoClient('127.0.0.1', server.local_bind_port) # server.
 ↪local_bind_port is assigned local port
```

0.2 Get metadata database

[58]:
```
db = client.metadata
```

What Collections are in metadata database?**

[59]:
```
db.collection_names()
```

[59]: ['attributes', 'metadata', 'bigdata', 'sources', 'datasets']

0.3 Create dataframes for sources and metadata collections

- The **sources collection** contains the names of all the sources that have datasets in the the bigdata collection.
- The **metadata collection** contains information about each dataset and each dataset object has an "attributes" object that is used as the look-up table for each attribute.

[60]:
```
sources = pd.DataFrame(list(db.sources.find()))
metadata= pd.DataFrame(list(db.metadata.find()))
```

0.4 What are the data sources?

[8]:
```
sources[['originator','originator_prefix']]
```

[8]:
```
                                                originator  \
0                                                 USA Facts
1  United States Deparment of Agriculture Economic Research Service
2                                 Harvard Global Health Institute

  originator_prefix
0         USA_FACTS
1              USDA
2              HGHI
```

0.5 What data sets are available?

[11]:
```
metadata[['originator_id','dataset_name','originator']]
```

[11]:
```
     originator_id  \
0       USDA_02_01
1       USDA_01_01
2       USDA_03_01
3  USA_FACTS_01_01
4  USA_FACTS_02_01
5       HGHI_02_01
6       HGHI_01_01

                                                                dataset_name
\
0                                                      Food Environment Atlas
1                                                   Food Access Research Atlas
2                                    Atlas of Rural and Small-Town America
3          Confirmed Covid 19 Cases in US by State and County (Jan-July 21 2020)
4         Confirmed Covid 19 Deaths in US by State and County (Jan-July 21 2020)
5  Harcard Global Health Institute COVID-19 Hospital Referral Regions (HRR) 2020
```

```
6      Harvard Global Health Institute COVID-19 Hospital Capacity Estimates 2020

  originator
0       USDA
1       USDA
2       USDA
3  USA_FACTS
4  USA_FACTS
5       HGHI
6       HGHI
```

0.6 What are the attributes for a specific dataset?

- The attr_label is the name of the attribute in the bigdata collection.

[26]:
```
# Use the originator_id of the dataset to get attributes object
org_id="HGHI_01_01"

# Create attributes dataframe
attributes=pd.DataFrame(list(metadata.loc[metadata.originator_id==org_id].
 ↪attributes)[0])

#displaying first 5 rows of attributes table
attributes.head(5)
```

[26]:
```
                        _id originator_id  start_date  end_date update_freq  \
0  5f3c938b85deac56aac95549    HGHI_01_01        2020      2020          NA
1  5f3c938b85deac56aac9554e    HGHI_01_01        2020      2020          NA
2  5f3c938b85deac56aac9554f    HGHI_01_01        2020      2020          NA
3  5f3c938b85deac56aac95550    HGHI_01_01        2020      2020          NA
4  5f3c938b85deac56aac95551    HGHI_01_01        2020      2020          NA

   iso_key iso_key_add     attr_label                      attr_orig  \
0        3          3b  HGHI_01_01_01                          State
1        9          NA  HGHI_01_01_02          20_total_hospital_beds
2        9          NA  HGHI_01_01_03               20_total_icu_beds
3        9          NA  HGHI_01_01_04  20_potentially_avail_icul_beds
4        9          NA  HGHI_01_01_05             20_icu_bed_occ_rate

                                              attr_desc attr_data_type  \
0                                                 State    SQL_VARCHAR
1              20% Hospital Capacity- Total Hospital Beds        SQL_INT
2                   20% Hospital Capacity- Total ICU Beds        SQL_INT
3  20%  Hospital Capacity- Potentially Available ICU Beds*        SQL_INT
4           20%  Hospital Capacity- ICU Bed Occupancy Rate       SQL_REAL
```

```
  scale positional_accuracy spatial_rep datum coordinate_system entity_type
0    NA                  NA     POLYGON    NA                NA       STATE
1    NA                  NA     POLYGON    NA                NA       STATE
2    NA                  NA     POLYGON    NA                NA       STATE
3    NA                  NA     POLYGON    NA                NA       STATE
4    NA                  NA     POLYGON    NA                NA       STATE
```

0.7 Get data for specific dataset

- **To pull in all the data for a dataset**, use the mongodb method "find()" to find all objects that contain one of the fields from the dataset (since each dataset will always have an attribute with the name "[originator_id]-01", I would suggest using that field).
- Use the attributes look up table above, to identify what data each column contains

[28]:
```
dataset_data=pd.DataFrame(list(db.bigdata.find({"HGHI_02_01_01":{"$ne":None}})))


#displaying first five rows of data
dataset_data.head(5)
```

[28]:
```
                        _id      HGHI_02_01_01  HGHI_02_01_02  HGHI_02_01_03  \
0  5f42f6e34b545863438e776d         Abilene TX          980.0          127.0
1  5f42f6e34b545863438e776e           Akron OH         1358.0          186.0
2  5f42f6e34b545863438e776f  Alameda County CA         2695.0          293.0
3  5f42f6e34b545863438e7770          Albany GA          704.0           60.0
4  5f42f6e34b545863438e7771          Albany NY         4804.0          425.0

   HGHI_02_01_07  HGHI_02_01_09  HGHI_02_01_04  HGHI_02_01_05  HGHI_02_01_06  \
0          565.0          772.0           68.0           98.0       226444.0
1          518.0          938.0           94.0          140.0       547990.0
2          665.0         1680.0          139.0          216.0      1310189.0
3          221.0          462.0           27.0           43.0       157143.0
4         1579.0         3191.0          193.0          309.0      1477723.0

   HGHI_02_01_08  …  HGHI_02_01_95  HGHI_02_01_98  HGHI_02_01_99  \
0        50412.0  …           4.27           3.29          209.0
1       111042.0  …           7.09           5.33          496.0
2       214991.0  …          10.56           7.78         1140.0
3        30466.0  …           6.56           4.70          141.0
4       318695.0  …           8.77           6.37         1355.0

   HGHI_02_01_100  HGHI_02_01_101  HGHI_02_01_103  HGHI_02_01_104  \
0            3.07            2.13            1.65           136.0
1            5.28            3.54            2.67           323.0
2            8.20            5.28            3.89           743.0
3            5.22            3.28            2.35            92.0
```

```
4            7.02            4.39            3.19           883.0

   HGHI_02_01_102  HGHI_02_01_105  HGHI_02_01_106
0            2.00            1.39            1.07
1            3.44            2.31            1.74
2            5.35            3.44            2.54
3            3.41            2.14            1.53
4            4.58            2.86            2.08

[5 rows x 107 columns]
```

0.8 Use data to create visual

[55]:
```
ax= dataset_data.loc[0:5].plot.bar(x='HGHI_02_01_01',y="HGHI_02_01_02",␣
↪title="20% Hospital Capacity", figsize=(10,5),color="orange",ec="black")
ax.get_legend().remove()
ax.set_xlabel("City, State")
ax.set_ylabel("Total Hospital Beds")
```

[55]:
```
Text(0, 0.5, 'Total Hospital Beds')
```