Convert_XML

June 2, 2019

0.1 Required Libraries/Packages

[1]:
```
import xml.etree.ElementTree as ET
#import xml.etree.cElementTree as ET
from bs4 import BeautifulSoup
import pandas as pd
import re
import os
```

[2]:
```
filepath = r'S:\Code\School\WGU_DataAnalyst_NanoDegree\02 - Data Wrangling with␣
↪MongoDB'
```

0.2 XML Conversion to Pandas DataFrame Object

Create function to retrieve relavant fields for each of the target tables. Once data is retrieved from cleaned XML file, data is then inserted into a DataFrame object for easy analysis/manipulation and exporting to .CSV

[3]:
```
def xml_retrieve_to_dataframe(search_tag):
    with open(os.path.join(filepath, 'Cleaned_Henderson.osm'),encoding="utf8")␣
↪as osm_file:
        nodes = []
        node_tags = []
        for event, elem in ET.iterparse(osm_file):
            for items in elem:
                    if items.tag == search_tag:
                        nodes.append(items.attrib)
                        if list(items):
                            for x in list(items):
                                temp_dict = {}
                                temp_dict.update(x.attrib)
                                temp_dict['type'] = x.tag
                                temp_dict['id'] = items.attrib['id']
                                node_tags.append(temp_dict)
        df = pd.DataFrame(nodes)
        tag_df = pd.DataFrame(node_tags)
        return df, tag_df
```

0.3 Node Related Datasets

Retrieval/exploration of Nodes dataset

[4]:
```
nodes, nodes_tags = xml_retrieve_to_dataframe('node')
```

[5]:
```
nodes.head()
```

[5]:
```
   changeset        id         lat           lon             timestamp     uid  \
0    5929701  54315452  35.3111691  -116.2478189  2010-10-01T17:14:39Z  194231
1    5929701  54315453  35.3316775  -116.2260267  2010-10-01T17:14:53Z  194231
2    5929701  54315455  35.3599244  -116.1782845  2010-10-01T17:14:35Z  194231
3    5929701  54315457  35.3624572  -116.1739125  2010-10-01T17:13:35Z  194231
4    5929701  54315458  35.3649766  -116.1696890  2010-10-01T17:14:52Z  194231

                        user version
0  Chris Bell in California       4
1  Chris Bell in California       3
2  Chris Bell in California       3
3  Chris Bell in California       3
4  Chris Bell in California       3
```

[6]:
```
nodes_tags.head()
```

[6]:
```
         id      k type      v
0  54315452  power  tag  tower
1  54315453  power  tag  tower
2  54315455  power  tag  tower
3  54315457  power  tag  tower
4  54315458  power  tag  tower
```

0.4 Way Related Datasets

Retrieval/exploration of Way dataset

[7]:
```
ways, ways_tags = xml_retrieve_to_dataframe('way')
```

[8]:
```
ways.head()
```

[8]:
```
   changeset        id             timestamp      uid           user version
0   63816665  14278359  2018-10-24T04:09:08Z  1330847  TheDutchMan13       3
1   63650721  14278368  2018-10-18T15:58:23Z  8446886       arpremna       4
2   49150987  14278402  2017-06-01T01:12:47Z  1330847  TheDutchMan13       3
3   51597855  14278416  2017-08-31T00:37:37Z  1240864        Howpper       7
4   63081661  14278432  2018-10-01T04:51:05Z  8407155        addatla       6
```

[9]:
```
ways_tags.head()
```

[9]:
```
         id    k        ref type    v
0  14278359  NaN  137032593   nd  NaN
1  14278359  NaN  137032596   nd  NaN
2  14278359  NaN  137032598   nd  NaN
3  14278359  NaN  137032600   nd  NaN
```

```
4  14278359  NaN  137032602   nd  NaN
```

Create separate DataFrame for key/value pairs that have the tag "ND" for node, then remove node dataset from tags dataset

[10]:
```
ways_nodes = ways_tags.loc[ways_tags['type'] == 'nd', :]
ways_tags = ways_tags.loc[ways_tags['type'] != 'nd', :]
```

[11]:
```
ways_nodes.drop(columns=['k', 'type', 'v'], inplace=True)
ways_nodes.head()
```

[11]:
```
         id        ref
0  14278359  137032593
1  14278359  137032596
2  14278359  137032598
3  14278359  137032600
4  14278359  137032602
```

[12]:
```
ways_tags.drop(columns=['ref'], inplace=True)
ways_tags.head()
```

[12]:
```
          id                k type            v
19  14278359          highway  tag  residential
20  14278359             name  tag  Rue De Parc
21  14278359       tiger:cfcc  tag          A41
22  14278359     tiger:county  tag    Clark, NV
23  14278359  tiger:name_base  tag  Rue de Parc
```

Exploration of different keys that exist in Ways tags dataset, several of these key/value pairs were cleaned during the audit phase of our analysis.

[13]:
```
ways_tags['k'].unique()
```

[13]:
```
array(['highway', 'name', 'tiger:cfcc', 'tiger:county', 'tiger:name_base',
       'tiger:reviewed', 'access', 'tiger:name_type', 'source', 'surface',
       'tracktype', 'electrified', 'gauge', 'operator', 'railway',
       'service', 'review', 'bicycle', 'destination', 'lanes', 'oneway',
       'destination:ref', 'tiger:separated', 'tiger:source', 'tiger:tlid',
       'tiger:upload_uuid', 'cables', 'frequency', 'layer', 'power',
       'voltage', 'destination:street', 'junction', 'name_1',
       'tiger:name_base_1', 'maxspeed', 'tiger:name_direction_prefix',
       'tiger:name_type_1', 'sidewalk', 'tiger:name_direction_prefix_1',
       'old_ref', 'source:old_ref', 'bridge', 'is_in', 'source:maxspeed',
       'name_2', 'tiger:name_base_2', 'owner', 'usage',
       'tiger:name_type_2', 'hgv', 'ref', 'NHS', 'alt_name',
       'hgv:national_network', 'old_name', 'source:hgv:national_network',
       'lit', 'addr:postcode', 'tiger:name_direction_prefix_2',
       'cycleway', 'leisure', 'sport', 'foot', 'tunnel', 'segregated',
       'landuse', 'created_by', 'amenity', 'ele', 'gnis:county_id',
       'gnis:created', 'gnis:feature_id', 'gnis:state_id', 'FIXME:name',
       'addr:city', 'addr:housenumber', 'addr:street', 'phone', 'website',
       'wikidata', 'wikipedia', 'embankment', 'fixme', 'noname',
```

```
       'intermittent', 'waterway', 'addr:state', 'description',
       'golf:course', 'golf:par', 'natural', 'religion', 'building',
       'shop', 'denomination', 'brand', 'brand:wikipedia', 'tourism',
       'admin_level', 'boundary', 'postal_code', 'wires', 'mtb:scale',
       'note', 'width', 'horse', 'motor_vehicle', 'barrier', 'smoothness',
       'source_ref', 'cycleway:right', 'brand:wikidata', 'designation',
       'source:name', 'gnis:edited', 'salt', 'tidal', 'water', 'tigis',
       'golf', 'parking', 'cuisine', 'takeaway', 'footway',
       'capacity:disabled', 'FIXME', 'emergency', 'area', 'sac_scale',
       'opening_hours', 'addr:housename', 'email', 'capacity',
       'wheelchair', 'man_made', 'parking:condition:both',
       'parking:lane:both', 'beds', 'healthcare', 'gnis:county_name',
       'attraction', 'lanes:backward', 'lanes:forward',
       'turn:lanes:forward', 'crossing', 'fence_type', 'max_age',
       'min_age', 'abandoned', 'atm', 'dispensing', 'incline',
       'mtb:scale:imba', 'trail_visibility', 'addr:country',
       'toilets:wheelchair', 'building:levels', 'smoking', 'ref:walmart',
       'cycling', 'ford', 'building:min_level', 'old_railway_operator',
       'placement', 'turn:lanes', 'payment:american_express',
       'payment:cash', 'payment:coins', 'payment:discover_card',
       'payment:mastercard', 'payment:visa', 'payment:visa_debit',
       'basin', 'attribution', 'delivery', 'drive_through',
       'outdoor_seating', 'location', 'substance', 'aeroway', 'faa',
       'road_marking', 'modifier', 'name:en', 'generator:source',
       'generator:type', 'covered', 'diaper', 'fee', 'toilets:disposal',
       'unisex', 'hoops', 'generator:method',
       'generator:output:electricity', 'addr:unit', 'craft',
       'service_times', 'operator:wikidata', 'operator:wikipedia',
       'shelter_type', 'culvert', 'maxstay', 'park_ride', 'supervised',
       'swimming_pool', 'opening_hours:url', 'playground', 'residential',
       'golf_cart', 'handicap', 'par', 'kerb', 'traffic_calming'],
      dtype=object)
```

0.5 Data Cleaning/Munging

Discover what values may be good candidates for cleaning (i.e. lots of values) and have values that can be recoded.

[14]:
```
for col_name in ways_tags['k'].unique():
    print(col_name, ' - ', ways_tags.loc[ways_tags['k'] == col_name, :].
 ↪count()['v'])
```

```
highway  -  21646
name  -  6784
tiger:cfcc  -  4140
tiger:county  -  4157
tiger:name_base  -  4079
tiger:reviewed  -  3940
```

```
access  -  850
tiger:name_type  -  3890
source  -  9740
surface  -  1736
tracktype  -  31
electrified  -  95
gauge  -  95
operator  -  104
railway  -  99
service  -  3204
review  -  8741
bicycle  -  951
destination  -  47
lanes  -  452
oneway  -  2583
destination:ref  -  16
tiger:separated  -  80
tiger:source  -  89
tiger:tlid  -  90
tiger:upload_uuid  -  36
cables  -  42
frequency  -  23
layer  -  362
power  -  90
voltage  -  31
destination:street  -  5
junction  -  94
name_1  -  272
tiger:name_base_1  -  275
maxspeed  -  261
tiger:name_direction_prefix  -  340
tiger:name_type_1  -  137
sidewalk  -  31
tiger:name_direction_prefix_1  -  36
old_ref  -  182
source:old_ref  -  5
bridge  -  198
is_in  -  14
source:maxspeed  -  6
name_2  -  17
tiger:name_base_2  -  16
owner  -  29
usage  -  26
tiger:name_type_2  -  9
hgv  -  89
ref  -  238
NHS  -  5
alt_name  -  44
```

```
hgv:national_network  -  5
old_name  -  69
source:hgv:national_network  -  5
lit  -  50
addr:postcode  -  104
tiger:name_direction_prefix_2  -  1
cycleway  -  22
leisure  -  508
sport  -  280
foot  -  664
tunnel  -  135
segregated  -  169
landuse  -  916
created_by  -  8
amenity  -  443
ele  -  52
gnis:county_id  -  40
gnis:created  -  40
gnis:feature_id  -  57
gnis:state_id  -  40
FIXME:name  -  1
addr:city  -  467
addr:housenumber  -  525
addr:street  -  541
phone  -  26
website  -  34
wikidata  -  9
wikipedia  -  5
embankment  -  21
fixme  -  32
noname  -  6
intermittent  -  380
waterway  -  420
addr:state  -  456
description  -  36
golf:course  -  3
golf:par  -  3
natural  -  249
religion  -  13
building  -  9905
shop  -  51
denomination  -  8
brand  -  28
brand:wikipedia  -  26
tourism  -  19
admin_level  -  15
boundary  -  17
postal_code  -  13
```

```
wires  -  19
mtb:scale  -  13
note  -  42
width  -  13
horse  -  36
motor_vehicle  -  67
barrier  -  210
smoothness  -  7
source_ref  -  17
cycleway:right  -  6
brand:wikidata  -  25
designation  -  3
source:name  -  2
gnis:edited  -  5
salt  -  3
tidal  -  3
water  -  46
tigis  -  5
golf  -  468
parking  -  52
cuisine  -  29
takeaway  -  4
footway  -  5906
capacity:disabled  -  17
FIXME  -  7
emergency  -  6
area  -  84
sac_scale  -  9
opening_hours  -  24
addr:housename  -  2
email  -  2
capacity  -  14
wheelchair  -  6
man_made  -  23
parking:condition:both  -  1
parking:lane:both  -  1
beds  -  1
healthcare  -  7
gnis:county_name  -  10
attraction  -  2
lanes:backward  -  5
lanes:forward  -  5
turn:lanes:forward  -  4
crossing  -  197
fence_type  -  4
max_age  -  3
min_age  -  3
abandoned  -  1
```

```
atm  -  6
dispensing  -  3
incline  -  37
mtb:scale:imba  -  3
trail_visibility  -  3
addr:country  -  55
toilets:wheelchair  -  1
building:levels  -  44
smoking  -  1
ref:walmart  -  3
cycling  -  1
ford  -  4
building:min_level  -  1
old_railway_operator  -  1
placement  -  21
turn:lanes  -  21
payment:american_express  -  1
payment:cash  -  1
payment:coins  -  1
payment:discover_card  -  1
payment:mastercard  -  1
payment:visa  -  1
payment:visa_debit  -  1
basin  -  1
attribution  -  2
delivery  -  1
drive_through  -  2
outdoor_seating  -  3
location  -  3
substance  -  2
aeroway  -  3
faa  -  1
road_marking  -  615
modifier  -  98
name:en  -  1
generator:source  -  2
generator:type  -  2
covered  -  3
diaper  -  1
fee  -  2
toilets:disposal  -  1
unisex  -  1
hoops  -  2
generator:method  -  2
generator:output:electricity  -  1
addr:unit  -  1
craft  -  1
service_times  -  1
```

```
operator:wikidata  -  2
operator:wikipedia  -  2
shelter_type  -  8
culvert  -  4
maxstay  -  1
park_ride  -  1
supervised  -  1
swimming_pool  -  1
opening_hours:url  -  1
playground  -  2
residential  -  1
golf_cart  -  35
handicap  -  18
par  -  18
kerb  -  1
traffic_calming  -  1
```

Yes/No now replace with True/False values. These could be boolean if data was of uniform type in target database.

[15]:
```
yes_no_dict = {'yes':'True', 'no':'False', 'YES':'True', 'Yes':'True', 'NO':
 ↪'False', 'No':'False'}
```

[16]:
```
ways_tags['v'].replace(yes_no_dict, inplace=True)
```

For each dataset, ensuring that NA values are filled with empty strings and data types are enforced ensures easy migration into sql.

0.5.1 Nodes Table - Clean/Enforce Data Types

[17]:
```
nodes['id'] = nodes['id'].astype(float)
nodes['lat'] = nodes['lat'].astype(float)
nodes['lon'] = nodes['lon'].astype(float)
nodes['uid'] = nodes['uid'].astype(float)
nodes['changeset'] = nodes['changeset'].astype(float)
nodes['version'] = nodes['version'].astype(int)
nodes.fillna('', inplace=True)
nodes = nodes[['id', 'lat', 'lon', 'user', 'uid', 'version', 'changeset',␣
 ↪'timestamp']]
nodes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 353779 entries, 0 to 353778
Data columns (total 8 columns):
id           353779 non-null float64
lat          353779 non-null float64
lon          353779 non-null float64
user         353779 non-null object
uid          353779 non-null float64
```

```
version      353779 non-null int32
changeset    353779 non-null float64
timestamp    353779 non-null object
dtypes: float64(5), int32(1), object(2)
memory usage: 20.2+ MB
```

0.5.2 Nodes_Tags Table - Clean/Enforce Data Types

[18]:
```
nodes_tags['id'] = nodes_tags['id'].astype(float)
nodes_tags.fillna('', inplace=True)
nodes_tags = nodes_tags[['id', 'k', 'v', 'type']]
nodes_tags.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16534 entries, 0 to 16533
Data columns (total 4 columns):
id      16534 non-null float64
k       16534 non-null object
v       16534 non-null object
type    16534 non-null object
dtypes: float64(1), object(3)
memory usage: 516.8+ KB
```

0.5.3 Ways Table - Clean/Enforce Data Types

[19]:
```
ways['id'] = ways['id'].astype(float)
ways['uid'] = ways['uid'].astype(float)
ways['changeset'] = ways['changeset'].astype(float)
ways.fillna('', inplace=True)
ways = ways[['id', 'user', 'uid', 'version', 'changeset', 'timestamp']]
ways.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35264 entries, 0 to 35263
Data columns (total 6 columns):
id           35264 non-null float64
user         35264 non-null object
uid          35264 non-null float64
version      35264 non-null object
changeset    35264 non-null float64
timestamp    35264 non-null object
dtypes: float64(3), object(3)
memory usage: 1.6+ MB
```

0.5.4 Ways_Tags Table - Clean/Enforce Data Types

[20]:
```
ways_tags['id'] = ways_tags['id'].astype(float)
ways_tags.fillna('', inplace=True)
ways_tags = ways_tags[['id', 'k', 'v', 'type']]
ways_tags.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 105949 entries, 19 to 519222
Data columns (total 4 columns):
id      105949 non-null float64
k       105949 non-null object
v       105949 non-null object
type    105949 non-null object
dtypes: float64(1), object(3)
memory usage: 4.0+ MB
```

0.5.5 Ensure Columns are in Proper Order

[21]:
```
ways_nodes['id'] = ways_nodes['id'].astype(float)
ways_nodes['ref'] = ways_nodes['ref'].astype(float)
ways_nodes.fillna('', inplace=True)
ways_nodes = ways_nodes[['id', 'ref']]
ways_nodes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 413274 entries, 0 to 519219
Data columns (total 2 columns):
id     413274 non-null float64
ref    413274 non-null float64
dtypes: float64(2)
memory usage: 9.5 MB
```

0.6 Export DataFrames to CSV for Import into SQL

Each DataFrame object is exported to CSV for importing in SQLite

[22]:
```
nodes.to_csv(os.path.join(filepath, 'nodes.csv'), index=False)
nodes_tags.to_csv(os.path.join(filepath, 'nodes_tags.csv'), index=False)
ways.to_csv(os.path.join(filepath, 'ways.csv'), index=False)
ways_tags.to_csv(os.path.join(filepath, 'ways_tags.csv'), index=False)
ways_nodes.to_csv(os.path.join(filepath, 'ways_nodes.csv'), index=False)
```