

Wrangling_OpenStreetMap

October 4, 2017

0.1 Wrangle OpenStreetMap data of Pittsburgh metro with MongoDB

OpenStreetMap is a community built free editable map of the world, inspired by the success of Wikipedia where crowdsourced data is open and free from proprietary restricted use. We see some examples of its use by Craigslist and Foursquare, as an open source alternative to Google Maps. <http://www.openstreetmap.org>

Users can map things such as polylines of roads, draw polygons of buildings or areas of interest, or insert nodes for landmarks. These map elements can be further tagged with details such as street addresses or amenity type. Map data is stored in an XML format. More details about the OSM XML can be found here: http://wiki.openstreetmap.org/wiki/OSM_XML

Some highlights of the OSM XML format relevant to this project are: - OSM XML is list of instances of data primitives (nodes, ways, and relations) found within a given bounds - nodes represent dimensionless points on the map - ways contain node references to form either a polyline or polygon on the map - nodes and ways both contain children tag elements that represent key value pairs of descriptive information about a given node or way

As with any user generated content, there is likely going to be dirty data. I will choose Pittsburgh as area in OpenStreetMap and use data munging techniques, such as assessing the quality of the data for validity, accuracy, completeness, consistency and uniformity, to clean the OpenStreetMap data for a part of the world that you care about.

The reality is this -- anyone who has worked with data extensively knows it is an absolute nightmare to get data from different data sources to play well with each other.

0.1.1 Choosing Map Area and Auditing

In this project, I would choose Pittsburgh, Since I have been living here for over one year and spent quality time in my graduate study in University of Pittsburgh From the beginning, we will read the data, and parse through it with ElementTree and count the number of unique element types. Iterative parsing is utilized since the XML is too large to process in memory.

```
In [2]: from xml.etree import cElementTree as ET
import pprint
#read the .osm data
osm_file=open("pittsburgh_metro.osm","r")
#count the unique element type
tags = {}
for event, elem in ET.iterparse(osm_file):
    if elem.tag in tags:
        tags[elem.tag] += 1
```

```

        else:
            tags[elem.tag] = 1

pprint.pprint(tags)

{'bounds': 1,
 'member': 47796,
 'nd': 2610073,
 'node': 2255583,
 'osm': 1,
 'relation': 4850,
 'tag': 1423165,
 'way': 243060}

In [3]: osm_file = open("pittsburgh_metro.osm", "r")
def process_map(filename):
    users = set()
    for _, element in ET.iterparse(filename):
        for e in element:
            if 'uid' in e.attrib:
                users.add(e.attrib['uid'])

    return users

users = process_map(osm_file)
len(users)

```

Out[3]: 1500

0.2 Problems in the data

0.2.1 Street name problems.

As you would expect from any crowd-sourced data, there is evidence of inconsistent naming conventions as well as evidence of human error when the data was entered. The two main areas of inconsistency that affect this project is in the street names used to describe the type of record in the data and the strings used as values within the data. It isn't feasible to analyse all values in the data, but street names are one type of value where inconsistency in values has a significant impact on the quality of the data.

The following analysis I will attempt to find problems with the street names and correct them with expected values.

```

In [4]: import os
import xml.etree.cElementTree as cET
from collections import defaultdict
import pprint
import re
import codecs

```

```

import json
import string
from pymongo import MongoClient
# some regular expression
lower = re.compile(r'^([a-z]|_)*$')
lower_colon = re.compile(r'^([a-z]|_)*:([a-z]|_)*$')
problemchars = re.compile(r'[=+/&<>;\'\"?%#$@\\,\\. \t\r\n]')
street_type_re = re.compile(r'\b\S+\.?$$', re.IGNORECASE)

# initial version of expected street names
expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Square", "Lane"]

In [5]: osm_file="pittsburgh_metro.osm"
def audit_street_type(street_types, street_name):
    # add unexpected street name to a list
    m = street_type_re.search(street_name)
    if m:
        street_type = m.group()
        if street_type not in expected:
            street_types[street_type].add(street_name)

def is_street_name(elem):
    # determine whether a element is a street name
    return (elem.attrib['k'] == "addr:street")

def audit_street(osmfile):
    # iter through all street name tag under node or way and audit the street name value
    osm_file = open(osmfile, "r")
    street_types = defaultdict(set)
    for event, elem in cET.iterparse(osm_file, events=("start",)):
        if elem.tag == "node" or elem.tag == "way":
            for tag in elem.iter("tag"):
                if is_street_name(tag):
                    audit_street_type(street_types, tag.attrib['v'])
    return street_types

st_types = audit_street(osm_file)
# print out unexpected street names
pprint.pprint(dict(st_types))

{'115': set(['McLaughlin Run Road #115']),
 '1302': set(['Perry Hwy #1302']),
 '18': set(['PA 18', 'Route 18']),
 '19': set(['Route 19', 'US 19']),
 '201': set(['Campbells Run Rd, Ste 201']),
 '202': set(['Wilkins Avenue #202']),
 '217': set(['217']),
 '228': set(['Pennsylvania 228', 'Pennysylvania 228', 'State Route 228']),

```

```

'286': set(['Route 286']),
'30': set(['Route 30', 'State Route 30', 'U.S. 30']),
'400': set(['West Kensinger Drive #400']),
'48': set(['48']),
'51': set(['Route 51', 'State Route 51']),
'519': set(['PA 519', 'Route 519']),
'620': set(['Grant St #620']),
'71': set(['Old Route 71']),
'8': set(['Route 8']),
'837': set(['Route 837']),
'88': set(['Route 88']),
'885': set(['Route 885']),
'910': set(['PA 910', 'Route 910']),
'Allegheny': set(['South Allegheny']),
'Alley': set(['2nd Alley',
              '4th Alley',
              'Beech Alley',
              'Bluff Alley',
              'Center Alley',
              'Chestnut Alley',
              'Church Alley',
              'Oak Alley',
              'Park Alley',
              'Peach Alley',
              'Pine Alley',
              'Plum Alley',
              'Poplar Alley',
              'School Alley',
              'Summit Alley',
              'Taylor Alley',
              'Thompson Alley',
              'Union Alley']),
'Allies': set(['Boulevard of the Allies']),
'Automotive': set(['California Automotive']),
'Av': set(['Center Av']),
'Av.': set(['Fifth Av.']),
'Ave': set(['5th Ave',
            'Arlington Ave',
            'Atlantic Ave',
            'Center Ave',
            'Centre Ave',
            'Elizabeth Ave',
            'Fifth Ave',
            'Forbes Ave',
            'Friendship Ave',
            'Glenn Ave',
            'Highland Ave',
            'Liberty Ave',

```

```

'Lincoln Ave',
'Lynnwood Ave',
'Morewood Ave',
'North Highland Ave',
'Penn Ave',
'Romine Ave',
'S Highland Ave',
'S Millvale Ave',
'S Negley Ave',
'S. Aiken Ave',
'Shadeland Ave',
'Shady Ave',
'South Aiken Ave',
'University Ave']],
'Ave.': set(['4th Ave.',
'5th Ave.',
'Chartiers Ave.',
'Fifth Ave.',
'Macon Ave.',
'Murray Ave.',
'Romine Ave.']),
'B': set(['New Texas Rd #B']),
'Blvd': set(['Beechwood Blvd',
'Clairton Blvd',
'Fort Duquesne Blvd',
'Halstead Blvd',
'Lysle Blvd',
'Monroeville Blvd',
'Pennsbury Blvd',
'Sunset Blvd',
'Washington Blvd']),
'Brdg': set(['Swindell Brdg']),
'CT': set(['Highfield CT']),
'Center': set(['Quaker Village Shopping Center']),
'Circle': set(['Alpine Circle',
'Bardona Circle',
'Bromley Circle',
'Chardonnay Circle',
'Chaucer Circle',
'Constitution Circle',
'Covestro Circle',
'East Russets Circle',
'Gander Circle',
'Glen Spring Circle',
'Golden Circle',
'Haverford Circle',
'Hemlock Circle',
'Hickory Circle',

```

```

'Horseshoe Circle',
'Jefferson Pointe Circle',
'Kettering Circle',
'Lark Tree Circle',
'Laurel Ridge Circle',
'Locust Circle',
'Longview Circle',
'Maple Circle',
'Marion Circle',
'Morning Wind Circle',
'Naughton Circle',
'New York Circle',
'North Pine Circle',
'North Woodland Circle',
'Oak Circle',
'Oakhurst Circle',
'Oakwood Circle',
'Open Meadow Circle',
'Philomena Circle',
'Redrome Circle',
'Ridge Point Circle',
'Sharps Circle',
'South Woodland Circle',
'Steepchase Circle',
'Summit Circle',
'Tor Circle',
'Trotwood Circle',
'Walnut Circle',
'West Grove Circle',
'West Russets Circle',
'Wheatland Circle',
'Wilderness Circle',
'Wildwood Circle',
'Winners Circle']],
'Connector': set(['John Scott Connector']),
'Cove': set(['Pheasant Cove']),
'Crossing': set(['Banbury Crossing',
                 'Butler Crossing',
                 'Grandview Crossing']),
'Ct': set(['Imperial Ct', 'LaPlace Point Ct']),
'DR': set(['MIDWAY DR']),
'Dowling': set(['Dowling']),
'Dr': set(['Berwick Dr',
           'Black Hawk Dr',
           'Camden Dr',
           'Corporate Dr',
           'Douglas Dr',
           'Eastminster Dr',

```

```

        'Fox Ridge Farms Dr',
        'Glengary Dr',
        'Greyfriar Dr',
        'Kirkwall Dr',
        'Selvin Dr']],
'East': set(['Deer Park Drive East',
            'Horseshoe Circle East',
            'Route 31 East',
            'Squaw Run Road East',
            'Waterfront Drive East',
            'Waterman Road East']),
'End': set(['Trails End']),
'Entrance': set(['Towne Hall Entrance']),
'Expressway': set(['Mon/Fayette Expressway', 'Tri-Boro Expressway']),
'Extension': set(['Bowman Street Extension',
            'Brierly Lane Extension',
            'Broadway Avenue Extension',
            'Federal Street Extension',
            'Gulf Lab Road Extension',
            'Hookstown Road Extension',
            'Jacktown Road Extension',
            'Middle Road Extension',
            'Mount Troy Road Extension',
            'South Main Street Extension',
            'Thomas Street Extension',
            'Thorn Hill Road Extension',
            'Virginia Avenue Extension']),
'Harbor': set(['Boothbay Harbor',
            'Marblehead Harbor',
            'Mystic Harbor',
            'Saybrook Harbor']),
'Harding': set(['Harding']),
'Heights': set(['Meadow Heights']),
'Highway': set(['Golden Mile Highway',
            'John Scott Highway',
            'Lincoln Highway',
            'Old Perry Highway',
            'Old William Penn Highway',
            'Perry Highway',
            'William Flinn Highway',
            'William Flynn Highway',
            'William Penn Highway']),
'Hill': set(['Serenity Hill']),
'Hwy': set(['Perry Hwy']),
'Joanne': set(['Joanne']),
'Library': set(['Library']),
'Ln': set(['Meadow Park Ln', 'Mountain Trails Ln']),
'Lysle': set(['Lysle']),

```

```

'Maples': set(['The Maples']),
'Marshall': set(['Marshall']),
'Maurers': set(['Maurers']),
'McAleer': set(['McAleer']),
'North': set(['Freedom Drive North',
              'Grandview Drive North',
              'Lakeside Drive North',
              'Randolph Drive North']),
'Oaks': set(['The Oaks']),
'PA-228': set(['PA-228']),
'PA-288': set(['PA-288']),
'PA-910': set(['PA-910']),
'PA-982': set(['PA-982']),
'Park': set(['Arlington Park', 'Blueberry Hill Park', 'Medical Park']),
'Patricia': set(['Patricia']),
'Penco': set(['Penco']),
'Pike': set(['Bethany Pike',
            'Clay Pike',
            'Greensburg Pike',
            'Kittanning Pike',
            'Steubenville Pike',
            'Washington Pike']),
'Plaza': set(['Chamber Plaza',
            'Mckees Rocks Plaza',
            'Moraine Pointe Plaza',
            'Penn Plaza']),
'Rd': set(['49376 Calcutta-Smith Ferry Rd',
          '520 Unity Center Rd',
          'Bayard Rd',
          'Brandt School Rd',
          'Browns Hill Rd',
          'Brownsville Rd',
          'California Rd',
          'Hill-Church Houston Rd',
          'Kerrwood Rd',
          'Latrobe Crabtree Rd',
          'McNeilly Rd',
          'Ravencrest Rd',
          'Robinhood Rd',
          'State Line Rd',
          'Wexford Rd']),
'Rossway': set(['Rossway']),
'Run': set(['Honey Run']),
'ST': set(['12TH ST']),
'South': set(['Freedom Drive South',
            'Grandview Drive South',
            'Penn Circle South',
            'Randolph Drive South']),

```



```

'Sq': set(['Elmer L Williams Sq', 'Harvard Sq', 'Sheridan Sq']),
'St': set(['7th St',
           '8th St',
           'Bellefonte St',
           'Castleman St',
           'Copeland St',
           'First St',
           'Hemlock St',
           'Henry St',
           'James St',
           'Locust St',
           'Main St',
           'Mirror St',
           'N 3rd St',
           'N Craig St',
           'N Dithridge St',
           'N Neville St',
           'North Bellefield St',
           'S Craig St',
           'S Graham St',
           'S Market St',
           'S Whitfield St',
           'Saline St',
           'Smithfield St',
           'South Craig St',
           'South Dithridge St',
           'Stanwix St',
           'Walnut St',
           'West St',
           'Winthrop St']),
'St.': set(['Atwood St.',
            'Bryant St.',
            'Byrant St.',
            'S. Main St.',
            'West 11th St.']),
'Strasse': set(['Rihn Strasse']),
'T566': set(['T566']),
'Ter': set(['Colby Ter', 'Faber Ter']),
'Terrace': set(['Chartiers Terrace',
                'Garden Terrace',
                'Highland Terrace',
                'Mellon Terrace',
                'Pinewood Terrace',
                'Riverview Terrace',
                'Schenk Terrace',
                'Sylvan Terrace',
                'Towne Hall Terrace',
                'West Grove Terrace']),

```

```

'Tr': set(['Mystic Pine Tr']),
'Trillium': set(['The Trillium']),
'US-30': set(['US-30']),
'Walnut': set(['Walnut']),
'Way': set(['Abes Way',
            'Amos Way',
            'Anglon Way',
            'Armitage Way',
            'Armstrong Way',
            'Bailey Way',
            'Baker Way',
            'Ballard Way',
            'Berlin Way',
            'Bizet Way',
            'Braden Way',
            'Burds Way',
            'Carey Way',
            'Carothers Way',
            'Carriage Way',
            'Chapel Way',
            'Cherry Way',
            'Chestnut Way',
            "Children's Way",
            'Colliers Way',
            'Commerce Way',
            'Darby Way',
            'Dix Way',
            'Dodge Way',
            'Duncan Way',
            'Elbow Way',
            'Elrod Way',
            'Euclair Way',
            'Frazee Way',
            'Ganchuk Way',
            'Green Way',
            'Gulch Way',
            'Hager Way',
            'Harmony Way',
            'Hogan Way',
            'Home Way',
            'Hospital Way',
            'Irwin Way',
            'Isleworth Way',
            'Jane Way',
            'Java Way',
            'Kalamazoo Way',
            'Kay Way',
            'Kelly Way',

```

'Kendall Way',
'Klein Way',
'Lacy Way',
'Langer Way',
'Largo Way',
'Legion Way',
'Liberty Way',
'Lincoln Way',
'Links Way',
'Lynn Way',
'Mackenzie Way',
'Marshall Way',
'Maxwell Way',
'Newbury Way',
'Nicklaus Way',
'Nolo Way',
'Orchard Way',
'Pace Way',
'Palmer Way',
'Pansy Way',
'Park Way',
'Partridge Way',
'Peaco Way',
'Player Way',
'Portland Way',
'Raccoon Way',
'Rapidan Way',
'Regis Way',
'Residence Way',
'Roxanna Way',
'Saint Simon Way',
'Sara Way',
'Saxon Way',
'Sentinel Way',
'Snowball Way',
'Stoebner Way',
'Summit Way',
'Sunset Way',
'Swan Way',
'Syrians Way',
'Taylor Way',
'Terminal Way',
'Terrace Way',
'Towne Square Way',
'Tripod Way',
'Trojan Way',
'Tunnel Way',
'Vanilla Way',

```

    'Vermont Way',
    'Veterans Way',
    'Virginia Way',
    'Washington Way',
    'Watercrest Way',
    'Weldin Way',
    'Wellston Way',
    'West Grove Way',
    'West Lock Way',
    'White Way',
    'Wittenberg Way',
    'Wrights Way',
    'Wycoff Way',
    'Yardley Way',
    'York Way']],
    'Weir': set(['Weir']),
    'West': set(['Deer Park Drive West',
                 'Horseshoe Circle West',
                 'Route 40 West',
                 'Waterfront Drive West',
                 'Waterman Road West']),
    'Wheel-In-Campground': set(['Wheel-In-Campground']),
    'center': set(['Hillcrest shopping center']),
    'rd': set(['Broadhead rd'])}

```

0.2.2 Street Name Correction Strategy

Based on the auditing results, I came up with the following mapping dictionary, which addressed the abbreviations and the incorrect names.

In [6]: *# creating a dictionary for correcting street names*

```

mapping = { "Ct": "Court",
            "St": "Street",
            "st": "Street",
            "St.": "Street",
            "St,": "Street",
            "ST": "Street",
            "street": "Street",
            "Street.": "Street",
            "Ave": "Avenue",
            "Ave.": "Avenue",
            "ave": "Avenue",
            "Rd.": "Road",
            "rd.": "Road",
            "Rd": "Road",
            "Hwy": "Highway",
            "HIghway": "Highway",

```

```

    "Pkwy": "Parkway",
    "Pl": "Place",
    "place": "Place",
    "Sedgwick": "Sedgwick Street",
    "Sq.": "Square",
    "Newbury": "Newbury Street",
    "Boylston": "Boylston Street",
    "Brook": "Brook Parkway",
    "Cambridge": "Cambridge Center",
    "Elm": "Elm Street",
    "Webster Street, Coolidge Corner": "Webster Street",
    "Faneuil Hall": "Faneuil Hall Market Street",
    "Furnace Brook": "Furnace Brook Parkway",
    "Federal": "Federal Street",
    "South Station, near Track 6": "South Station, Summer Street",
    "PO Box 846028": "846028 Surface Road",
    "First Street, Suite 303": "First Street",
    "Kendall Square - 3": "Kendall Square",
    "Franklin Street, Suite 1702": "Franklin Street",
    "First Street, Suite 1100": "First Street",
    "Windsor": "Windsor Stearns Hill Road",
    "Winsor": "Winsor Village Pilgrim Road",
    "First Street, 18th floor": "First Street",
    "Sidney Street, 2nd floor": "Sidney Street",
    "Boston Providence Turnpike": "Boston Providence Highway",
    "LOMASNEY WAY, ROOF LEVEL": "Lomasney Way",
    "Holland": "Holland Albany Street",
    "Hampshire": "Hampshire Street",
    "Boylston Street, 5th Floor": "Boylston Street",
    "Fenway": "Fenway Yawkey Way",
    "Charles Street South": "Charles Street"}

# function that corrects incorrect street names
def update_name(name, mapping):
    for key in mapping:
        if key in name:
            name = string.replace(name, key, mapping[key])
    return name

```

0.2.3 Zip code auditing

The zip codes in Pittsburgh should start with 15, audit incorrect zip codes in the data set

```

In [7]: def audit_zipcodes(osmfile):
    # iter through all zip codes, collect all the zip codes that does not start with 15
    osm_file = open(osmfile, "r")
    zip_codes = {}
    for event, elem in cET.iterparse(osm_file, events=("start",)):

```

```

        if elem.tag == "node" or elem.tag == "way":
            for tag in elem.iter("tag"):
                if tag.attrib['k'] == "addr:postcode" and not tag.attrib['v'].startswith("0"):
                    if tag.attrib['v'] not in zip_codes:
                        zip_codes[tag.attrib['v']] = 1
                    else:
                        zip_codes[tag.attrib['v']] += 1
            return zip_codes

zipcodes = audit_zipcodes(osm_file)
for zipcode in zipcodes:
    print zipcode, zipcodes[zipcode]
#print "there number of wrond zipcodes which does not start with 15 is:",len(zipcodes)
#sum(zipcodes)

```

```

16059 2
26062-4500 1
16055 59
44413 1
16056 4
16053 299
43964 1
PA 15033 1
26070 3
16002 16
26062-4598 1
unknown 1
PA 15601 1
16066 179
16001 24
16063 28
26034 2
26050 1
16023 4
26059 3
16027 2
16201 1
14233 1
16046 29
16045 1
43952 68
43953 19
16123 2
26037 2
26062 18
2573 2
16115 1
44432 2

```

```
16117 1
26003 9
16037 44
California PA, 15419 1
PA 15632 1
16157 1
16033 215
26062-4525 2
26047 2
43920 28
```

The mistake zip codes are not that many in our dataset, since we have very large volume of data record, and some of them are zip code of places close to Pittsburgh, PA

0.3 Handle the element and process the OpenStreetMap XML file

The function to update the street name and then handle all the elements in the osm file to process the XML file, so that we can make sure that the file is ready to insert into MongoDB,

```
In [8]: CREATED = [ "version", "changeset", "timestamp", "user", "uid"]
```

```
def shape_element(element):
    node = {}
    node["created"]={}
    node["address"]={}
    node["pos"]=[]
    refs=[]

    # we only process the node and way tags
    if element.tag == "node" or element.tag == "way" :

        if "id" in element.attrib:
            node["id"]=element.attrib["id"]
            node["type"]=element.tag

        if "visible" in element.attrib.keys():
            node["visible"]=element.attrib["visible"]

        # the key-value pairs with attributes in the CREATED list are added under key "c
        for elem in CREATED:
            if elem in element.attrib:
                node["created"][elem]=element.attrib[elem]

        # attributes for latitude and longitude are added to a "pos" array
        # include latitude value
        if "lat" in element.attrib:
            node["pos"].append(float(element.attrib["lat"]))
```

```

        # include longitude value
        if "lon" in element.attrib:
            node["pos"].append(float(element.attrib["lon"]))

    for tag in element.iter("tag"):
        if not(problemchars.search(tag.attrib['k'])):
            if tag.attrib['k'] == "addr:housenumber":
                node["address"]["houenumber"]=tag.attrib['v']

            if tag.attrib['k'] == "addr:postcode":
                node["address"]["postcode"]=tag.attrib['v']

            # handling the street attribute, update incorrect names using the strate
            if tag.attrib['k'] == "addr:street":
                node["address"]["street"]=tag.attrib['v']
                node["address"]["street"] = update_name(node["address"]["street"], m

            if tag.attrib['k'].find("addr")==-1:
                node[tag.attrib['k']]=tag.attrib['v']

    for nd in element.iter("nd"):
        refs.append(nd.attrib["ref"])

    if node["address"] =={}:
        node.pop("address", None)

    if refs != []:
        node["node_refs"]=refs

    return node
else:
    return None

# process the xml openstreetmap file, write a json out file and return a list of diction
def process_map(file_in, pretty = False):
    file_out = "{0}.json".format(file_in)
    data = []
    with codecs.open(file_out, "w") as fo:
        for _, element in cET.iterparse(file_in):
            el = shape_element(element)
            if el:
                data.append(el)
                if pretty:
                    fo.write(json.dumps(el, indent=2)+"\n")
                else:
                    fo.write(json.dumps(el) + "\n")
    return data

```



```
In [9]: data = process_map(osm_file, True)
        print "finished"
```

finished

0.3.1 Insert the json file into the local mongodb

before we running the following code, we should run the mongodb service successfully in local server. connect the mongodb to the localhost:27017

```
In [10]: from pymongo import MongoClient
         client = MongoClient('localhost:27017')
         db = client.openstreetmap
         collection = db.pittsburgh_metro
         collection.insert(data)
```

/Users/VanessaTong/anaconda/lib/python2.7/site-packages/ipykernel/__main__.py:5: DeprecationWarn

```
Out[10]: [ObjectId('59ba8f9d4276d283e53c03f7'),
          ObjectId('59ba8f9d4276d283e53c03f8'),
          ObjectId('59ba8f9d4276d283e53c03f9'),
          ObjectId('59ba8f9d4276d283e53c03fa'),
          ObjectId('59ba8f9d4276d283e53c03fb'),
          ObjectId('59ba8f9d4276d283e53c03fc'),
          ObjectId('59ba8f9d4276d283e53c03fd'),
          ObjectId('59ba8f9d4276d283e53c03fe'),
          ObjectId('59ba8f9d4276d283e53c03ff'),
          ObjectId('59ba8f9d4276d283e53c0400'),
          ObjectId('59ba8f9d4276d283e53c0401'),
          ObjectId('59ba8f9d4276d283e53c0402'),
          ObjectId('59ba8f9d4276d283e53c0403'),
          ObjectId('59ba8f9d4276d283e53c0404'),
          ObjectId('59ba8f9d4276d283e53c0405'),
          ObjectId('59ba8f9d4276d283e53c0406'),
          ObjectId('59ba8f9d4276d283e53c0407'),
          ObjectId('59ba8f9d4276d283e53c0408'),
          ObjectId('59ba8f9d4276d283e53c0409'),
          ObjectId('59ba8f9d4276d283e53c040a'),
          ObjectId('59ba8f9d4276d283e53c040b'),
          ObjectId('59ba8f9d4276d283e53c040c'),
          ObjectId('59ba8f9d4276d283e53c040d'),
          ObjectId('59ba8f9d4276d283e53c040e'),
          ObjectId('59ba8f9d4276d283e53c040f'),
          ObjectId('59ba8f9d4276d283e53c0410'),
          ObjectId('59ba8f9d4276d283e53c0411'),
          ObjectId('59ba8f9d4276d283e53c0412'),
          ObjectId('59ba8f9d4276d283e53c0413'),
```

ObjectId('59ba8f9d4276d283e53c0414'),
ObjectId('59ba8f9d4276d283e53c0415'),
ObjectId('59ba8f9d4276d283e53c0416'),
ObjectId('59ba8f9d4276d283e53c0417'),
ObjectId('59ba8f9d4276d283e53c0418'),
ObjectId('59ba8f9d4276d283e53c0419'),
ObjectId('59ba8f9d4276d283e53c041a'),
ObjectId('59ba8f9d4276d283e53c041b'),
ObjectId('59ba8f9d4276d283e53c041c'),
ObjectId('59ba8f9d4276d283e53c041d'),
ObjectId('59ba8f9d4276d283e53c041e'),
ObjectId('59ba8f9d4276d283e53c041f'),
ObjectId('59ba8f9d4276d283e53c0420'),
ObjectId('59ba8f9d4276d283e53c0421'),
ObjectId('59ba8f9d4276d283e53c0422'),
ObjectId('59ba8f9d4276d283e53c0423'),
ObjectId('59ba8f9d4276d283e53c0424'),
ObjectId('59ba8f9d4276d283e53c0425'),
ObjectId('59ba8f9d4276d283e53c0426'),
ObjectId('59ba8f9d4276d283e53c0427'),
ObjectId('59ba8f9d4276d283e53c0428'),
ObjectId('59ba8f9d4276d283e53c0429'),
ObjectId('59ba8f9d4276d283e53c042a'),
ObjectId('59ba8f9d4276d283e53c042b'),
ObjectId('59ba8f9d4276d283e53c042c'),
ObjectId('59ba8f9d4276d283e53c042d'),
ObjectId('59ba8f9d4276d283e53c042e'),
ObjectId('59ba8f9d4276d283e53c042f'),
ObjectId('59ba8f9d4276d283e53c0430'),
ObjectId('59ba8f9d4276d283e53c0431'),
ObjectId('59ba8f9d4276d283e53c0432'),
ObjectId('59ba8f9d4276d283e53c0433'),
ObjectId('59ba8f9d4276d283e53c0434'),
ObjectId('59ba8f9d4276d283e53c0435'),
ObjectId('59ba8f9d4276d283e53c0436'),
ObjectId('59ba8f9d4276d283e53c0437'),
ObjectId('59ba8f9d4276d283e53c0438'),
ObjectId('59ba8f9d4276d283e53c0439'),
ObjectId('59ba8f9d4276d283e53c043a'),
ObjectId('59ba8f9d4276d283e53c043b'),
ObjectId('59ba8f9d4276d283e53c043c'),
ObjectId('59ba8f9d4276d283e53c043d'),
ObjectId('59ba8f9d4276d283e53c043e'),
ObjectId('59ba8f9d4276d283e53c043f'),
ObjectId('59ba8f9d4276d283e53c0440'),
ObjectId('59ba8f9d4276d283e53c0441'),
ObjectId('59ba8f9d4276d283e53c0442'),
ObjectId('59ba8f9d4276d283e53c0443'),

ObjectId('59ba8f9d4276d283e53c0444'),
ObjectId('59ba8f9d4276d283e53c0445'),
ObjectId('59ba8f9d4276d283e53c0446'),
ObjectId('59ba8f9d4276d283e53c0447'),
ObjectId('59ba8f9d4276d283e53c0448'),
ObjectId('59ba8f9d4276d283e53c0449'),
ObjectId('59ba8f9d4276d283e53c044a'),
ObjectId('59ba8f9d4276d283e53c044b'),
ObjectId('59ba8f9d4276d283e53c044c'),
ObjectId('59ba8f9d4276d283e53c044d'),
ObjectId('59ba8f9d4276d283e53c044e'),
ObjectId('59ba8f9d4276d283e53c044f'),
ObjectId('59ba8f9d4276d283e53c0450'),
ObjectId('59ba8f9d4276d283e53c0451'),
ObjectId('59ba8f9d4276d283e53c0452'),
ObjectId('59ba8f9d4276d283e53c0453'),
ObjectId('59ba8f9d4276d283e53c0454'),
ObjectId('59ba8f9d4276d283e53c0455'),
ObjectId('59ba8f9d4276d283e53c0456'),
ObjectId('59ba8f9d4276d283e53c0457'),
ObjectId('59ba8f9d4276d283e53c0458'),
ObjectId('59ba8f9d4276d283e53c0459'),
ObjectId('59ba8f9d4276d283e53c045a'),
ObjectId('59ba8f9d4276d283e53c045b'),
ObjectId('59ba8f9d4276d283e53c045c'),
ObjectId('59ba8f9d4276d283e53c045d'),
ObjectId('59ba8f9d4276d283e53c045e'),
ObjectId('59ba8f9d4276d283e53c045f'),
ObjectId('59ba8f9d4276d283e53c0460'),
ObjectId('59ba8f9d4276d283e53c0461'),
ObjectId('59ba8f9d4276d283e53c0462'),
ObjectId('59ba8f9d4276d283e53c0463'),
ObjectId('59ba8f9d4276d283e53c0464'),
ObjectId('59ba8f9d4276d283e53c0465'),
ObjectId('59ba8f9d4276d283e53c0466'),
ObjectId('59ba8f9d4276d283e53c0467'),
ObjectId('59ba8f9d4276d283e53c0468'),
ObjectId('59ba8f9d4276d283e53c0469'),
ObjectId('59ba8f9d4276d283e53c046a'),
ObjectId('59ba8f9d4276d283e53c046b'),
ObjectId('59ba8f9d4276d283e53c046c'),
ObjectId('59ba8f9d4276d283e53c046d'),
ObjectId('59ba8f9d4276d283e53c046e'),
ObjectId('59ba8f9d4276d283e53c046f'),
ObjectId('59ba8f9d4276d283e53c0470'),
ObjectId('59ba8f9d4276d283e53c0471'),
ObjectId('59ba8f9d4276d283e53c0472'),
ObjectId('59ba8f9d4276d283e53c0473'),

ObjectId('59ba8f9d4276d283e53c0474'),
ObjectId('59ba8f9d4276d283e53c0475'),
ObjectId('59ba8f9d4276d283e53c0476'),
ObjectId('59ba8f9d4276d283e53c0477'),
ObjectId('59ba8f9d4276d283e53c0478'),
ObjectId('59ba8f9d4276d283e53c0479'),
ObjectId('59ba8f9d4276d283e53c047a'),
ObjectId('59ba8f9d4276d283e53c047b'),
ObjectId('59ba8f9d4276d283e53c047c'),
ObjectId('59ba8f9d4276d283e53c047d'),
ObjectId('59ba8f9d4276d283e53c047e'),
ObjectId('59ba8f9d4276d283e53c047f'),
ObjectId('59ba8f9d4276d283e53c0480'),
ObjectId('59ba8f9d4276d283e53c0481'),
ObjectId('59ba8f9d4276d283e53c0482'),
ObjectId('59ba8f9d4276d283e53c0483'),
ObjectId('59ba8f9d4276d283e53c0484'),
ObjectId('59ba8f9d4276d283e53c0485'),
ObjectId('59ba8f9d4276d283e53c0486'),
ObjectId('59ba8f9d4276d283e53c0487'),
ObjectId('59ba8f9d4276d283e53c0488'),
ObjectId('59ba8f9d4276d283e53c0489'),
ObjectId('59ba8f9d4276d283e53c048a'),
ObjectId('59ba8f9d4276d283e53c048b'),
ObjectId('59ba8f9d4276d283e53c048c'),
ObjectId('59ba8f9d4276d283e53c048d'),
ObjectId('59ba8f9d4276d283e53c048e'),
ObjectId('59ba8f9d4276d283e53c048f'),
ObjectId('59ba8f9d4276d283e53c0490'),
ObjectId('59ba8f9d4276d283e53c0491'),
ObjectId('59ba8f9d4276d283e53c0492'),
ObjectId('59ba8f9d4276d283e53c0493'),
ObjectId('59ba8f9d4276d283e53c0494'),
ObjectId('59ba8f9d4276d283e53c0495'),
ObjectId('59ba8f9d4276d283e53c0496'),
ObjectId('59ba8f9d4276d283e53c0497'),
ObjectId('59ba8f9d4276d283e53c0498'),
ObjectId('59ba8f9d4276d283e53c0499'),
ObjectId('59ba8f9d4276d283e53c049a'),
ObjectId('59ba8f9d4276d283e53c049b'),
ObjectId('59ba8f9d4276d283e53c049c'),
ObjectId('59ba8f9d4276d283e53c049d'),
ObjectId('59ba8f9d4276d283e53c049e'),
ObjectId('59ba8f9d4276d283e53c049f'),
ObjectId('59ba8f9d4276d283e53c04a0'),
ObjectId('59ba8f9d4276d283e53c04a1'),
ObjectId('59ba8f9d4276d283e53c04a2'),
ObjectId('59ba8f9d4276d283e53c04a3'),

ObjectId('59ba8f9d4276d283e53c04a4'),
ObjectId('59ba8f9d4276d283e53c04a5'),
ObjectId('59ba8f9d4276d283e53c04a6'),
ObjectId('59ba8f9d4276d283e53c04a7'),
ObjectId('59ba8f9d4276d283e53c04a8'),
ObjectId('59ba8f9d4276d283e53c04a9'),
ObjectId('59ba8f9d4276d283e53c04aa'),
ObjectId('59ba8f9d4276d283e53c04ab'),
ObjectId('59ba8f9d4276d283e53c04ac'),
ObjectId('59ba8f9d4276d283e53c04ad'),
ObjectId('59ba8f9d4276d283e53c04ae'),
ObjectId('59ba8f9d4276d283e53c04af'),
ObjectId('59ba8f9d4276d283e53c04b0'),
ObjectId('59ba8f9d4276d283e53c04b1'),
ObjectId('59ba8f9d4276d283e53c04b2'),
ObjectId('59ba8f9d4276d283e53c04b3'),
ObjectId('59ba8f9d4276d283e53c04b4'),
ObjectId('59ba8f9d4276d283e53c04b5'),
ObjectId('59ba8f9d4276d283e53c04b6'),
ObjectId('59ba8f9d4276d283e53c04b7'),
ObjectId('59ba8f9d4276d283e53c04b8'),
ObjectId('59ba8f9d4276d283e53c04b9'),
ObjectId('59ba8f9d4276d283e53c04ba'),
ObjectId('59ba8f9d4276d283e53c04bb'),
ObjectId('59ba8f9d4276d283e53c04bc'),
ObjectId('59ba8f9d4276d283e53c04bd'),
ObjectId('59ba8f9d4276d283e53c04be'),
ObjectId('59ba8f9d4276d283e53c04bf'),
ObjectId('59ba8f9d4276d283e53c04c0'),
ObjectId('59ba8f9d4276d283e53c04c1'),
ObjectId('59ba8f9d4276d283e53c04c2'),
ObjectId('59ba8f9d4276d283e53c04c3'),
ObjectId('59ba8f9d4276d283e53c04c4'),
ObjectId('59ba8f9d4276d283e53c04c5'),
ObjectId('59ba8f9d4276d283e53c04c6'),
ObjectId('59ba8f9d4276d283e53c04c7'),
ObjectId('59ba8f9d4276d283e53c04c8'),
ObjectId('59ba8f9d4276d283e53c04c9'),
ObjectId('59ba8f9d4276d283e53c04ca'),
ObjectId('59ba8f9d4276d283e53c04cb'),
ObjectId('59ba8f9d4276d283e53c04cc'),
ObjectId('59ba8f9d4276d283e53c04cd'),
ObjectId('59ba8f9d4276d283e53c04ce'),
ObjectId('59ba8f9d4276d283e53c04cf'),
ObjectId('59ba8f9d4276d283e53c04d0'),
ObjectId('59ba8f9d4276d283e53c04d1'),
ObjectId('59ba8f9d4276d283e53c04d2'),
ObjectId('59ba8f9d4276d283e53c04d3'),

ObjectId('59ba8f9d4276d283e53c04d4'),
ObjectId('59ba8f9d4276d283e53c04d5'),
ObjectId('59ba8f9d4276d283e53c04d6'),
ObjectId('59ba8f9d4276d283e53c04d7'),
ObjectId('59ba8f9d4276d283e53c04d8'),
ObjectId('59ba8f9d4276d283e53c04d9'),
ObjectId('59ba8f9d4276d283e53c04da'),
ObjectId('59ba8f9d4276d283e53c04db'),
ObjectId('59ba8f9d4276d283e53c04dc'),
ObjectId('59ba8f9d4276d283e53c04dd'),
ObjectId('59ba8f9d4276d283e53c04de'),
ObjectId('59ba8f9d4276d283e53c04df'),
ObjectId('59ba8f9d4276d283e53c04e0'),
ObjectId('59ba8f9d4276d283e53c04e1'),
ObjectId('59ba8f9d4276d283e53c04e2'),
ObjectId('59ba8f9d4276d283e53c04e3'),
ObjectId('59ba8f9d4276d283e53c04e4'),
ObjectId('59ba8f9d4276d283e53c04e5'),
ObjectId('59ba8f9d4276d283e53c04e6'),
ObjectId('59ba8f9d4276d283e53c04e7'),
ObjectId('59ba8f9d4276d283e53c04e8'),
ObjectId('59ba8f9d4276d283e53c04e9'),
ObjectId('59ba8f9d4276d283e53c04ea'),
ObjectId('59ba8f9d4276d283e53c04eb'),
ObjectId('59ba8f9d4276d283e53c04ec'),
ObjectId('59ba8f9d4276d283e53c04ed'),
ObjectId('59ba8f9d4276d283e53c04ee'),
ObjectId('59ba8f9d4276d283e53c04ef'),
ObjectId('59ba8f9d4276d283e53c04f0'),
ObjectId('59ba8f9d4276d283e53c04f1'),
ObjectId('59ba8f9d4276d283e53c04f2'),
ObjectId('59ba8f9d4276d283e53c04f3'),
ObjectId('59ba8f9d4276d283e53c04f4'),
ObjectId('59ba8f9d4276d283e53c04f5'),
ObjectId('59ba8f9d4276d283e53c04f6'),
ObjectId('59ba8f9d4276d283e53c04f7'),
ObjectId('59ba8f9d4276d283e53c04f8'),
ObjectId('59ba8f9d4276d283e53c04f9'),
ObjectId('59ba8f9d4276d283e53c04fa'),
ObjectId('59ba8f9d4276d283e53c04fb'),
ObjectId('59ba8f9d4276d283e53c04fc'),
ObjectId('59ba8f9d4276d283e53c04fd'),
ObjectId('59ba8f9d4276d283e53c04fe'),
ObjectId('59ba8f9d4276d283e53c04ff'),
ObjectId('59ba8f9d4276d283e53c0500'),
ObjectId('59ba8f9d4276d283e53c0501'),
ObjectId('59ba8f9d4276d283e53c0502'),
ObjectId('59ba8f9d4276d283e53c0503'),

ObjectId('59ba8f9d4276d283e53c0504'),
ObjectId('59ba8f9d4276d283e53c0505'),
ObjectId('59ba8f9d4276d283e53c0506'),
ObjectId('59ba8f9d4276d283e53c0507'),
ObjectId('59ba8f9d4276d283e53c0508'),
ObjectId('59ba8f9d4276d283e53c0509'),
ObjectId('59ba8f9d4276d283e53c050a'),
ObjectId('59ba8f9d4276d283e53c050b'),
ObjectId('59ba8f9d4276d283e53c050c'),
ObjectId('59ba8f9d4276d283e53c050d'),
ObjectId('59ba8f9d4276d283e53c050e'),
ObjectId('59ba8f9d4276d283e53c050f'),
ObjectId('59ba8f9d4276d283e53c0510'),
ObjectId('59ba8f9d4276d283e53c0511'),
ObjectId('59ba8f9d4276d283e53c0512'),
ObjectId('59ba8f9d4276d283e53c0513'),
ObjectId('59ba8f9d4276d283e53c0514'),
ObjectId('59ba8f9d4276d283e53c0515'),
ObjectId('59ba8f9d4276d283e53c0516'),
ObjectId('59ba8f9d4276d283e53c0517'),
ObjectId('59ba8f9d4276d283e53c0518'),
ObjectId('59ba8f9d4276d283e53c0519'),
ObjectId('59ba8f9d4276d283e53c051a'),
ObjectId('59ba8f9d4276d283e53c051b'),
ObjectId('59ba8f9d4276d283e53c051c'),
ObjectId('59ba8f9d4276d283e53c051d'),
ObjectId('59ba8f9d4276d283e53c051e'),
ObjectId('59ba8f9d4276d283e53c051f'),
ObjectId('59ba8f9d4276d283e53c0520'),
ObjectId('59ba8f9d4276d283e53c0521'),
ObjectId('59ba8f9d4276d283e53c0522'),
ObjectId('59ba8f9d4276d283e53c0523'),
ObjectId('59ba8f9d4276d283e53c0524'),
ObjectId('59ba8f9d4276d283e53c0525'),
ObjectId('59ba8f9d4276d283e53c0526'),
ObjectId('59ba8f9d4276d283e53c0527'),
ObjectId('59ba8f9d4276d283e53c0528'),
ObjectId('59ba8f9d4276d283e53c0529'),
ObjectId('59ba8f9d4276d283e53c052a'),
ObjectId('59ba8f9d4276d283e53c052b'),
ObjectId('59ba8f9d4276d283e53c052c'),
ObjectId('59ba8f9d4276d283e53c052d'),
ObjectId('59ba8f9d4276d283e53c052e'),
ObjectId('59ba8f9d4276d283e53c052f'),
ObjectId('59ba8f9d4276d283e53c0530'),
ObjectId('59ba8f9d4276d283e53c0531'),
ObjectId('59ba8f9d4276d283e53c0532'),
ObjectId('59ba8f9d4276d283e53c0533'),

ObjectId('59ba8f9d4276d283e53c0534'),
ObjectId('59ba8f9d4276d283e53c0535'),
ObjectId('59ba8f9d4276d283e53c0536'),
ObjectId('59ba8f9d4276d283e53c0537'),
ObjectId('59ba8f9d4276d283e53c0538'),
ObjectId('59ba8f9d4276d283e53c0539'),
ObjectId('59ba8f9d4276d283e53c053a'),
ObjectId('59ba8f9d4276d283e53c053b'),
ObjectId('59ba8f9d4276d283e53c053c'),
ObjectId('59ba8f9d4276d283e53c053d'),
ObjectId('59ba8f9d4276d283e53c053e'),
ObjectId('59ba8f9d4276d283e53c053f'),
ObjectId('59ba8f9d4276d283e53c0540'),
ObjectId('59ba8f9d4276d283e53c0541'),
ObjectId('59ba8f9d4276d283e53c0542'),
ObjectId('59ba8f9d4276d283e53c0543'),
ObjectId('59ba8f9d4276d283e53c0544'),
ObjectId('59ba8f9d4276d283e53c0545'),
ObjectId('59ba8f9d4276d283e53c0546'),
ObjectId('59ba8f9d4276d283e53c0547'),
ObjectId('59ba8f9d4276d283e53c0548'),
ObjectId('59ba8f9d4276d283e53c0549'),
ObjectId('59ba8f9d4276d283e53c054a'),
ObjectId('59ba8f9d4276d283e53c054b'),
ObjectId('59ba8f9d4276d283e53c054c'),
ObjectId('59ba8f9d4276d283e53c054d'),
ObjectId('59ba8f9d4276d283e53c054e'),
ObjectId('59ba8f9d4276d283e53c054f'),
ObjectId('59ba8f9d4276d283e53c0550'),
ObjectId('59ba8f9d4276d283e53c0551'),
ObjectId('59ba8f9d4276d283e53c0552'),
ObjectId('59ba8f9d4276d283e53c0553'),
ObjectId('59ba8f9d4276d283e53c0554'),
ObjectId('59ba8f9d4276d283e53c0555'),
ObjectId('59ba8f9d4276d283e53c0556'),
ObjectId('59ba8f9d4276d283e53c0557'),
ObjectId('59ba8f9d4276d283e53c0558'),
ObjectId('59ba8f9d4276d283e53c0559'),
ObjectId('59ba8f9d4276d283e53c055a'),
ObjectId('59ba8f9d4276d283e53c055b'),
ObjectId('59ba8f9d4276d283e53c055c'),
ObjectId('59ba8f9d4276d283e53c055d'),
ObjectId('59ba8f9d4276d283e53c055e'),
ObjectId('59ba8f9d4276d283e53c055f'),
ObjectId('59ba8f9d4276d283e53c0560'),
ObjectId('59ba8f9d4276d283e53c0561'),
ObjectId('59ba8f9d4276d283e53c0562'),
ObjectId('59ba8f9d4276d283e53c0563'),

ObjectId('59ba8f9d4276d283e53c0564'),
ObjectId('59ba8f9d4276d283e53c0565'),
ObjectId('59ba8f9d4276d283e53c0566'),
ObjectId('59ba8f9d4276d283e53c0567'),
ObjectId('59ba8f9d4276d283e53c0568'),
ObjectId('59ba8f9d4276d283e53c0569'),
ObjectId('59ba8f9d4276d283e53c056a'),
ObjectId('59ba8f9d4276d283e53c056b'),
ObjectId('59ba8f9d4276d283e53c056c'),
ObjectId('59ba8f9d4276d283e53c056d'),
ObjectId('59ba8f9d4276d283e53c056e'),
ObjectId('59ba8f9d4276d283e53c056f'),
ObjectId('59ba8f9d4276d283e53c0570'),
ObjectId('59ba8f9d4276d283e53c0571'),
ObjectId('59ba8f9d4276d283e53c0572'),
ObjectId('59ba8f9d4276d283e53c0573'),
ObjectId('59ba8f9d4276d283e53c0574'),
ObjectId('59ba8f9d4276d283e53c0575'),
ObjectId('59ba8f9d4276d283e53c0576'),
ObjectId('59ba8f9d4276d283e53c0577'),
ObjectId('59ba8f9d4276d283e53c0578'),
ObjectId('59ba8f9d4276d283e53c0579'),
ObjectId('59ba8f9d4276d283e53c057a'),
ObjectId('59ba8f9d4276d283e53c057b'),
ObjectId('59ba8f9d4276d283e53c057c'),
ObjectId('59ba8f9d4276d283e53c057d'),
ObjectId('59ba8f9d4276d283e53c057e'),
ObjectId('59ba8f9d4276d283e53c057f'),
ObjectId('59ba8f9d4276d283e53c0580'),
ObjectId('59ba8f9d4276d283e53c0581'),
ObjectId('59ba8f9d4276d283e53c0582'),
ObjectId('59ba8f9d4276d283e53c0583'),
ObjectId('59ba8f9d4276d283e53c0584'),
ObjectId('59ba8f9d4276d283e53c0585'),
ObjectId('59ba8f9d4276d283e53c0586'),
ObjectId('59ba8f9d4276d283e53c0587'),
ObjectId('59ba8f9d4276d283e53c0588'),
ObjectId('59ba8f9d4276d283e53c0589'),
ObjectId('59ba8f9d4276d283e53c058a'),
ObjectId('59ba8f9d4276d283e53c058b'),
ObjectId('59ba8f9d4276d283e53c058c'),
ObjectId('59ba8f9d4276d283e53c058d'),
ObjectId('59ba8f9d4276d283e53c058e'),
ObjectId('59ba8f9d4276d283e53c058f'),
ObjectId('59ba8f9d4276d283e53c0590'),
ObjectId('59ba8f9d4276d283e53c0591'),
ObjectId('59ba8f9d4276d283e53c0592'),
ObjectId('59ba8f9d4276d283e53c0593'),

ObjectId('59ba8f9d4276d283e53c0594'),
ObjectId('59ba8f9d4276d283e53c0595'),
ObjectId('59ba8f9d4276d283e53c0596'),
ObjectId('59ba8f9d4276d283e53c0597'),
ObjectId('59ba8f9d4276d283e53c0598'),
ObjectId('59ba8f9d4276d283e53c0599'),
ObjectId('59ba8f9d4276d283e53c059a'),
ObjectId('59ba8f9d4276d283e53c059b'),
ObjectId('59ba8f9d4276d283e53c059c'),
ObjectId('59ba8f9d4276d283e53c059d'),
ObjectId('59ba8f9d4276d283e53c059e'),
ObjectId('59ba8f9d4276d283e53c059f'),
ObjectId('59ba8f9d4276d283e53c05a0'),
ObjectId('59ba8f9d4276d283e53c05a1'),
ObjectId('59ba8f9d4276d283e53c05a2'),
ObjectId('59ba8f9d4276d283e53c05a3'),
ObjectId('59ba8f9d4276d283e53c05a4'),
ObjectId('59ba8f9d4276d283e53c05a5'),
ObjectId('59ba8f9d4276d283e53c05a6'),
ObjectId('59ba8f9d4276d283e53c05a7'),
ObjectId('59ba8f9d4276d283e53c05a8'),
ObjectId('59ba8f9d4276d283e53c05a9'),
ObjectId('59ba8f9d4276d283e53c05aa'),
ObjectId('59ba8f9d4276d283e53c05ab'),
ObjectId('59ba8f9d4276d283e53c05ac'),
ObjectId('59ba8f9d4276d283e53c05ad'),
ObjectId('59ba8f9d4276d283e53c05ae'),
ObjectId('59ba8f9d4276d283e53c05af'),
ObjectId('59ba8f9d4276d283e53c05b0'),
ObjectId('59ba8f9d4276d283e53c05b1'),
ObjectId('59ba8f9d4276d283e53c05b2'),
ObjectId('59ba8f9d4276d283e53c05b3'),
ObjectId('59ba8f9d4276d283e53c05b4'),
ObjectId('59ba8f9d4276d283e53c05b5'),
ObjectId('59ba8f9d4276d283e53c05b6'),
ObjectId('59ba8f9d4276d283e53c05b7'),
ObjectId('59ba8f9d4276d283e53c05b8'),
ObjectId('59ba8f9d4276d283e53c05b9'),
ObjectId('59ba8f9d4276d283e53c05ba'),
ObjectId('59ba8f9d4276d283e53c05bb'),
ObjectId('59ba8f9d4276d283e53c05bc'),
ObjectId('59ba8f9d4276d283e53c05bd'),
ObjectId('59ba8f9d4276d283e53c05be'),
ObjectId('59ba8f9d4276d283e53c05bf'),
ObjectId('59ba8f9d4276d283e53c05c0'),
ObjectId('59ba8f9d4276d283e53c05c1'),
ObjectId('59ba8f9d4276d283e53c05c2'),
ObjectId('59ba8f9d4276d283e53c05c3'),

ObjectId('59ba8f9d4276d283e53c05c4'),
ObjectId('59ba8f9d4276d283e53c05c5'),
ObjectId('59ba8f9d4276d283e53c05c6'),
ObjectId('59ba8f9d4276d283e53c05c7'),
ObjectId('59ba8f9d4276d283e53c05c8'),
ObjectId('59ba8f9d4276d283e53c05c9'),
ObjectId('59ba8f9d4276d283e53c05ca'),
ObjectId('59ba8f9d4276d283e53c05cb'),
ObjectId('59ba8f9d4276d283e53c05cc'),
ObjectId('59ba8f9d4276d283e53c05cd'),
ObjectId('59ba8f9d4276d283e53c05ce'),
ObjectId('59ba8f9d4276d283e53c05cf'),
ObjectId('59ba8f9d4276d283e53c05d0'),
ObjectId('59ba8f9d4276d283e53c05d1'),
ObjectId('59ba8f9d4276d283e53c05d2'),
ObjectId('59ba8f9d4276d283e53c05d3'),
ObjectId('59ba8f9d4276d283e53c05d4'),
ObjectId('59ba8f9d4276d283e53c05d5'),
ObjectId('59ba8f9d4276d283e53c05d6'),
ObjectId('59ba8f9d4276d283e53c05d7'),
ObjectId('59ba8f9d4276d283e53c05d8'),
ObjectId('59ba8f9d4276d283e53c05d9'),
ObjectId('59ba8f9d4276d283e53c05da'),
ObjectId('59ba8f9d4276d283e53c05db'),
ObjectId('59ba8f9d4276d283e53c05dc'),
ObjectId('59ba8f9d4276d283e53c05dd'),
ObjectId('59ba8f9d4276d283e53c05de'),
ObjectId('59ba8f9d4276d283e53c05df'),
ObjectId('59ba8f9d4276d283e53c05e0'),
ObjectId('59ba8f9d4276d283e53c05e1'),
ObjectId('59ba8f9d4276d283e53c05e2'),
ObjectId('59ba8f9d4276d283e53c05e3'),
ObjectId('59ba8f9d4276d283e53c05e4'),
ObjectId('59ba8f9d4276d283e53c05e5'),
ObjectId('59ba8f9d4276d283e53c05e6'),
ObjectId('59ba8f9d4276d283e53c05e7'),
ObjectId('59ba8f9d4276d283e53c05e8'),
ObjectId('59ba8f9d4276d283e53c05e9'),
ObjectId('59ba8f9d4276d283e53c05ea'),
ObjectId('59ba8f9d4276d283e53c05eb'),
ObjectId('59ba8f9d4276d283e53c05ec'),
ObjectId('59ba8f9d4276d283e53c05ed'),
ObjectId('59ba8f9d4276d283e53c05ee'),
ObjectId('59ba8f9d4276d283e53c05ef'),
ObjectId('59ba8f9d4276d283e53c05f0'),
ObjectId('59ba8f9d4276d283e53c05f1'),
ObjectId('59ba8f9d4276d283e53c05f2'),
ObjectId('59ba8f9d4276d283e53c05f3'),

ObjectId('59ba8f9d4276d283e53c05f4'),
ObjectId('59ba8f9d4276d283e53c05f5'),
ObjectId('59ba8f9d4276d283e53c05f6'),
ObjectId('59ba8f9d4276d283e53c05f7'),
ObjectId('59ba8f9d4276d283e53c05f8'),
ObjectId('59ba8f9d4276d283e53c05f9'),
ObjectId('59ba8f9d4276d283e53c05fa'),
ObjectId('59ba8f9d4276d283e53c05fb'),
ObjectId('59ba8f9d4276d283e53c05fc'),
ObjectId('59ba8f9d4276d283e53c05fd'),
ObjectId('59ba8f9d4276d283e53c05fe'),
ObjectId('59ba8f9d4276d283e53c05ff'),
ObjectId('59ba8f9d4276d283e53c0600'),
ObjectId('59ba8f9d4276d283e53c0601'),
ObjectId('59ba8f9d4276d283e53c0602'),
ObjectId('59ba8f9d4276d283e53c0603'),
ObjectId('59ba8f9d4276d283e53c0604'),
ObjectId('59ba8f9d4276d283e53c0605'),
ObjectId('59ba8f9d4276d283e53c0606'),
ObjectId('59ba8f9d4276d283e53c0607'),
ObjectId('59ba8f9d4276d283e53c0608'),
ObjectId('59ba8f9d4276d283e53c0609'),
ObjectId('59ba8f9d4276d283e53c060a'),
ObjectId('59ba8f9d4276d283e53c060b'),
ObjectId('59ba8f9d4276d283e53c060c'),
ObjectId('59ba8f9d4276d283e53c060d'),
ObjectId('59ba8f9d4276d283e53c060e'),
ObjectId('59ba8f9d4276d283e53c060f'),
ObjectId('59ba8f9d4276d283e53c0610'),
ObjectId('59ba8f9d4276d283e53c0611'),
ObjectId('59ba8f9d4276d283e53c0612'),
ObjectId('59ba8f9d4276d283e53c0613'),
ObjectId('59ba8f9d4276d283e53c0614'),
ObjectId('59ba8f9d4276d283e53c0615'),
ObjectId('59ba8f9d4276d283e53c0616'),
ObjectId('59ba8f9d4276d283e53c0617'),
ObjectId('59ba8f9d4276d283e53c0618'),
ObjectId('59ba8f9d4276d283e53c0619'),
ObjectId('59ba8f9d4276d283e53c061a'),
ObjectId('59ba8f9d4276d283e53c061b'),
ObjectId('59ba8f9d4276d283e53c061c'),
ObjectId('59ba8f9d4276d283e53c061d'),
ObjectId('59ba8f9d4276d283e53c061e'),
ObjectId('59ba8f9d4276d283e53c061f'),
ObjectId('59ba8f9d4276d283e53c0620'),
ObjectId('59ba8f9d4276d283e53c0621'),
ObjectId('59ba8f9d4276d283e53c0622'),
ObjectId('59ba8f9d4276d283e53c0623'),

ObjectId('59ba8f9d4276d283e53c0624'),
ObjectId('59ba8f9d4276d283e53c0625'),
ObjectId('59ba8f9d4276d283e53c0626'),
ObjectId('59ba8f9d4276d283e53c0627'),
ObjectId('59ba8f9d4276d283e53c0628'),
ObjectId('59ba8f9d4276d283e53c0629'),
ObjectId('59ba8f9d4276d283e53c062a'),
ObjectId('59ba8f9d4276d283e53c062b'),
ObjectId('59ba8f9d4276d283e53c062c'),
ObjectId('59ba8f9d4276d283e53c062d'),
ObjectId('59ba8f9d4276d283e53c062e'),
ObjectId('59ba8f9d4276d283e53c062f'),
ObjectId('59ba8f9d4276d283e53c0630'),
ObjectId('59ba8f9d4276d283e53c0631'),
ObjectId('59ba8f9d4276d283e53c0632'),
ObjectId('59ba8f9d4276d283e53c0633'),
ObjectId('59ba8f9d4276d283e53c0634'),
ObjectId('59ba8f9d4276d283e53c0635'),
ObjectId('59ba8f9d4276d283e53c0636'),
ObjectId('59ba8f9d4276d283e53c0637'),
ObjectId('59ba8f9d4276d283e53c0638'),
ObjectId('59ba8f9d4276d283e53c0639'),
ObjectId('59ba8f9d4276d283e53c063a'),
ObjectId('59ba8f9d4276d283e53c063b'),
ObjectId('59ba8f9d4276d283e53c063c'),
ObjectId('59ba8f9d4276d283e53c063d'),
ObjectId('59ba8f9d4276d283e53c063e'),
ObjectId('59ba8f9d4276d283e53c063f'),
ObjectId('59ba8f9d4276d283e53c0640'),
ObjectId('59ba8f9d4276d283e53c0641'),
ObjectId('59ba8f9d4276d283e53c0642'),
ObjectId('59ba8f9d4276d283e53c0643'),
ObjectId('59ba8f9d4276d283e53c0644'),
ObjectId('59ba8f9d4276d283e53c0645'),
ObjectId('59ba8f9d4276d283e53c0646'),
ObjectId('59ba8f9d4276d283e53c0647'),
ObjectId('59ba8f9d4276d283e53c0648'),
ObjectId('59ba8f9d4276d283e53c0649'),
ObjectId('59ba8f9d4276d283e53c064a'),
ObjectId('59ba8f9d4276d283e53c064b'),
ObjectId('59ba8f9d4276d283e53c064c'),
ObjectId('59ba8f9d4276d283e53c064d'),
ObjectId('59ba8f9d4276d283e53c064e'),
ObjectId('59ba8f9d4276d283e53c064f'),
ObjectId('59ba8f9d4276d283e53c0650'),
ObjectId('59ba8f9d4276d283e53c0651'),
ObjectId('59ba8f9d4276d283e53c0652'),
ObjectId('59ba8f9d4276d283e53c0653'),

ObjectId('59ba8f9d4276d283e53c0654'),
ObjectId('59ba8f9d4276d283e53c0655'),
ObjectId('59ba8f9d4276d283e53c0656'),
ObjectId('59ba8f9d4276d283e53c0657'),
ObjectId('59ba8f9d4276d283e53c0658'),
ObjectId('59ba8f9d4276d283e53c0659'),
ObjectId('59ba8f9d4276d283e53c065a'),
ObjectId('59ba8f9d4276d283e53c065b'),
ObjectId('59ba8f9d4276d283e53c065c'),
ObjectId('59ba8f9d4276d283e53c065d'),
ObjectId('59ba8f9d4276d283e53c065e'),
ObjectId('59ba8f9d4276d283e53c065f'),
ObjectId('59ba8f9d4276d283e53c0660'),
ObjectId('59ba8f9d4276d283e53c0661'),
ObjectId('59ba8f9d4276d283e53c0662'),
ObjectId('59ba8f9d4276d283e53c0663'),
ObjectId('59ba8f9d4276d283e53c0664'),
ObjectId('59ba8f9d4276d283e53c0665'),
ObjectId('59ba8f9d4276d283e53c0666'),
ObjectId('59ba8f9d4276d283e53c0667'),
ObjectId('59ba8f9d4276d283e53c0668'),
ObjectId('59ba8f9d4276d283e53c0669'),
ObjectId('59ba8f9d4276d283e53c066a'),
ObjectId('59ba8f9d4276d283e53c066b'),
ObjectId('59ba8f9d4276d283e53c066c'),
ObjectId('59ba8f9d4276d283e53c066d'),
ObjectId('59ba8f9d4276d283e53c066e'),
ObjectId('59ba8f9d4276d283e53c066f'),
ObjectId('59ba8f9d4276d283e53c0670'),
ObjectId('59ba8f9d4276d283e53c0671'),
ObjectId('59ba8f9d4276d283e53c0672'),
ObjectId('59ba8f9d4276d283e53c0673'),
ObjectId('59ba8f9d4276d283e53c0674'),
ObjectId('59ba8f9d4276d283e53c0675'),
ObjectId('59ba8f9d4276d283e53c0676'),
ObjectId('59ba8f9d4276d283e53c0677'),
ObjectId('59ba8f9d4276d283e53c0678'),
ObjectId('59ba8f9d4276d283e53c0679'),
ObjectId('59ba8f9d4276d283e53c067a'),
ObjectId('59ba8f9d4276d283e53c067b'),
ObjectId('59ba8f9d4276d283e53c067c'),
ObjectId('59ba8f9d4276d283e53c067d'),
ObjectId('59ba8f9d4276d283e53c067e'),
ObjectId('59ba8f9d4276d283e53c067f'),
ObjectId('59ba8f9d4276d283e53c0680'),
ObjectId('59ba8f9d4276d283e53c0681'),
ObjectId('59ba8f9d4276d283e53c0682'),
ObjectId('59ba8f9d4276d283e53c0683'),

ObjectId('59ba8f9d4276d283e53c0684'),
ObjectId('59ba8f9d4276d283e53c0685'),
ObjectId('59ba8f9d4276d283e53c0686'),
ObjectId('59ba8f9d4276d283e53c0687'),
ObjectId('59ba8f9d4276d283e53c0688'),
ObjectId('59ba8f9d4276d283e53c0689'),
ObjectId('59ba8f9d4276d283e53c068a'),
ObjectId('59ba8f9d4276d283e53c068b'),
ObjectId('59ba8f9d4276d283e53c068c'),
ObjectId('59ba8f9d4276d283e53c068d'),
ObjectId('59ba8f9d4276d283e53c068e'),
ObjectId('59ba8f9d4276d283e53c068f'),
ObjectId('59ba8f9d4276d283e53c0690'),
ObjectId('59ba8f9d4276d283e53c0691'),
ObjectId('59ba8f9d4276d283e53c0692'),
ObjectId('59ba8f9d4276d283e53c0693'),
ObjectId('59ba8f9d4276d283e53c0694'),
ObjectId('59ba8f9d4276d283e53c0695'),
ObjectId('59ba8f9d4276d283e53c0696'),
ObjectId('59ba8f9d4276d283e53c0697'),
ObjectId('59ba8f9d4276d283e53c0698'),
ObjectId('59ba8f9d4276d283e53c0699'),
ObjectId('59ba8f9d4276d283e53c069a'),
ObjectId('59ba8f9d4276d283e53c069b'),
ObjectId('59ba8f9d4276d283e53c069c'),
ObjectId('59ba8f9d4276d283e53c069d'),
ObjectId('59ba8f9d4276d283e53c069e'),
ObjectId('59ba8f9d4276d283e53c069f'),
ObjectId('59ba8f9d4276d283e53c06a0'),
ObjectId('59ba8f9d4276d283e53c06a1'),
ObjectId('59ba8f9d4276d283e53c06a2'),
ObjectId('59ba8f9d4276d283e53c06a3'),
ObjectId('59ba8f9d4276d283e53c06a4'),
ObjectId('59ba8f9d4276d283e53c06a5'),
ObjectId('59ba8f9d4276d283e53c06a6'),
ObjectId('59ba8f9d4276d283e53c06a7'),
ObjectId('59ba8f9d4276d283e53c06a8'),
ObjectId('59ba8f9d4276d283e53c06a9'),
ObjectId('59ba8f9d4276d283e53c06aa'),
ObjectId('59ba8f9d4276d283e53c06ab'),
ObjectId('59ba8f9d4276d283e53c06ac'),
ObjectId('59ba8f9d4276d283e53c06ad'),
ObjectId('59ba8f9d4276d283e53c06ae'),
ObjectId('59ba8f9d4276d283e53c06af'),
ObjectId('59ba8f9d4276d283e53c06b0'),
ObjectId('59ba8f9d4276d283e53c06b1'),
ObjectId('59ba8f9d4276d283e53c06b2'),
ObjectId('59ba8f9d4276d283e53c06b3'),

ObjectId('59ba8f9d4276d283e53c06b4'),
ObjectId('59ba8f9d4276d283e53c06b5'),
ObjectId('59ba8f9d4276d283e53c06b6'),
ObjectId('59ba8f9d4276d283e53c06b7'),
ObjectId('59ba8f9d4276d283e53c06b8'),
ObjectId('59ba8f9d4276d283e53c06b9'),
ObjectId('59ba8f9d4276d283e53c06ba'),
ObjectId('59ba8f9d4276d283e53c06bb'),
ObjectId('59ba8f9d4276d283e53c06bc'),
ObjectId('59ba8f9d4276d283e53c06bd'),
ObjectId('59ba8f9d4276d283e53c06be'),
ObjectId('59ba8f9d4276d283e53c06bf'),
ObjectId('59ba8f9d4276d283e53c06c0'),
ObjectId('59ba8f9d4276d283e53c06c1'),
ObjectId('59ba8f9d4276d283e53c06c2'),
ObjectId('59ba8f9d4276d283e53c06c3'),
ObjectId('59ba8f9d4276d283e53c06c4'),
ObjectId('59ba8f9d4276d283e53c06c5'),
ObjectId('59ba8f9d4276d283e53c06c6'),
ObjectId('59ba8f9d4276d283e53c06c7'),
ObjectId('59ba8f9d4276d283e53c06c8'),
ObjectId('59ba8f9d4276d283e53c06c9'),
ObjectId('59ba8f9d4276d283e53c06ca'),
ObjectId('59ba8f9d4276d283e53c06cb'),
ObjectId('59ba8f9d4276d283e53c06cc'),
ObjectId('59ba8f9d4276d283e53c06cd'),
ObjectId('59ba8f9d4276d283e53c06ce'),
ObjectId('59ba8f9d4276d283e53c06cf'),
ObjectId('59ba8f9d4276d283e53c06d0'),
ObjectId('59ba8f9d4276d283e53c06d1'),
ObjectId('59ba8f9d4276d283e53c06d2'),
ObjectId('59ba8f9d4276d283e53c06d3'),
ObjectId('59ba8f9d4276d283e53c06d4'),
ObjectId('59ba8f9d4276d283e53c06d5'),
ObjectId('59ba8f9d4276d283e53c06d6'),
ObjectId('59ba8f9d4276d283e53c06d7'),
ObjectId('59ba8f9d4276d283e53c06d8'),
ObjectId('59ba8f9d4276d283e53c06d9'),
ObjectId('59ba8f9d4276d283e53c06da'),
ObjectId('59ba8f9d4276d283e53c06db'),
ObjectId('59ba8f9d4276d283e53c06dc'),
ObjectId('59ba8f9d4276d283e53c06dd'),
ObjectId('59ba8f9d4276d283e53c06de'),
ObjectId('59ba8f9d4276d283e53c06df'),
ObjectId('59ba8f9d4276d283e53c06e0'),
ObjectId('59ba8f9d4276d283e53c06e1'),
ObjectId('59ba8f9d4276d283e53c06e2'),
ObjectId('59ba8f9d4276d283e53c06e3'),

ObjectId('59ba8f9d4276d283e53c06e4'),
ObjectId('59ba8f9d4276d283e53c06e5'),
ObjectId('59ba8f9d4276d283e53c06e6'),
ObjectId('59ba8f9d4276d283e53c06e7'),
ObjectId('59ba8f9d4276d283e53c06e8'),
ObjectId('59ba8f9d4276d283e53c06e9'),
ObjectId('59ba8f9d4276d283e53c06ea'),
ObjectId('59ba8f9d4276d283e53c06eb'),
ObjectId('59ba8f9d4276d283e53c06ec'),
ObjectId('59ba8f9d4276d283e53c06ed'),
ObjectId('59ba8f9d4276d283e53c06ee'),
ObjectId('59ba8f9d4276d283e53c06ef'),
ObjectId('59ba8f9d4276d283e53c06f0'),
ObjectId('59ba8f9d4276d283e53c06f1'),
ObjectId('59ba8f9d4276d283e53c06f2'),
ObjectId('59ba8f9d4276d283e53c06f3'),
ObjectId('59ba8f9d4276d283e53c06f4'),
ObjectId('59ba8f9d4276d283e53c06f5'),
ObjectId('59ba8f9d4276d283e53c06f6'),
ObjectId('59ba8f9d4276d283e53c06f7'),
ObjectId('59ba8f9d4276d283e53c06f8'),
ObjectId('59ba8f9d4276d283e53c06f9'),
ObjectId('59ba8f9d4276d283e53c06fa'),
ObjectId('59ba8f9d4276d283e53c06fb'),
ObjectId('59ba8f9d4276d283e53c06fc'),
ObjectId('59ba8f9d4276d283e53c06fd'),
ObjectId('59ba8f9d4276d283e53c06fe'),
ObjectId('59ba8f9d4276d283e53c06ff'),
ObjectId('59ba8f9d4276d283e53c0700'),
ObjectId('59ba8f9d4276d283e53c0701'),
ObjectId('59ba8f9d4276d283e53c0702'),
ObjectId('59ba8f9d4276d283e53c0703'),
ObjectId('59ba8f9d4276d283e53c0704'),
ObjectId('59ba8f9d4276d283e53c0705'),
ObjectId('59ba8f9d4276d283e53c0706'),
ObjectId('59ba8f9d4276d283e53c0707'),
ObjectId('59ba8f9d4276d283e53c0708'),
ObjectId('59ba8f9d4276d283e53c0709'),
ObjectId('59ba8f9d4276d283e53c070a'),
ObjectId('59ba8f9d4276d283e53c070b'),
ObjectId('59ba8f9d4276d283e53c070c'),
ObjectId('59ba8f9d4276d283e53c070d'),
ObjectId('59ba8f9d4276d283e53c070e'),
ObjectId('59ba8f9d4276d283e53c070f'),
ObjectId('59ba8f9d4276d283e53c0710'),
ObjectId('59ba8f9d4276d283e53c0711'),
ObjectId('59ba8f9d4276d283e53c0712'),
ObjectId('59ba8f9d4276d283e53c0713'),

ObjectId('59ba8f9d4276d283e53c0714'),
ObjectId('59ba8f9d4276d283e53c0715'),
ObjectId('59ba8f9d4276d283e53c0716'),
ObjectId('59ba8f9d4276d283e53c0717'),
ObjectId('59ba8f9d4276d283e53c0718'),
ObjectId('59ba8f9d4276d283e53c0719'),
ObjectId('59ba8f9d4276d283e53c071a'),
ObjectId('59ba8f9d4276d283e53c071b'),
ObjectId('59ba8f9d4276d283e53c071c'),
ObjectId('59ba8f9d4276d283e53c071d'),
ObjectId('59ba8f9d4276d283e53c071e'),
ObjectId('59ba8f9d4276d283e53c071f'),
ObjectId('59ba8f9d4276d283e53c0720'),
ObjectId('59ba8f9d4276d283e53c0721'),
ObjectId('59ba8f9d4276d283e53c0722'),
ObjectId('59ba8f9d4276d283e53c0723'),
ObjectId('59ba8f9d4276d283e53c0724'),
ObjectId('59ba8f9d4276d283e53c0725'),
ObjectId('59ba8f9d4276d283e53c0726'),
ObjectId('59ba8f9d4276d283e53c0727'),
ObjectId('59ba8f9d4276d283e53c0728'),
ObjectId('59ba8f9d4276d283e53c0729'),
ObjectId('59ba8f9d4276d283e53c072a'),
ObjectId('59ba8f9d4276d283e53c072b'),
ObjectId('59ba8f9d4276d283e53c072c'),
ObjectId('59ba8f9d4276d283e53c072d'),
ObjectId('59ba8f9d4276d283e53c072e'),
ObjectId('59ba8f9d4276d283e53c072f'),
ObjectId('59ba8f9d4276d283e53c0730'),
ObjectId('59ba8f9d4276d283e53c0731'),
ObjectId('59ba8f9d4276d283e53c0732'),
ObjectId('59ba8f9d4276d283e53c0733'),
ObjectId('59ba8f9d4276d283e53c0734'),
ObjectId('59ba8f9d4276d283e53c0735'),
ObjectId('59ba8f9d4276d283e53c0736'),
ObjectId('59ba8f9d4276d283e53c0737'),
ObjectId('59ba8f9d4276d283e53c0738'),
ObjectId('59ba8f9d4276d283e53c0739'),
ObjectId('59ba8f9d4276d283e53c073a'),
ObjectId('59ba8f9d4276d283e53c073b'),
ObjectId('59ba8f9d4276d283e53c073c'),
ObjectId('59ba8f9d4276d283e53c073d'),
ObjectId('59ba8f9d4276d283e53c073e'),
ObjectId('59ba8f9d4276d283e53c073f'),
ObjectId('59ba8f9d4276d283e53c0740'),
ObjectId('59ba8f9d4276d283e53c0741'),
ObjectId('59ba8f9d4276d283e53c0742'),
ObjectId('59ba8f9d4276d283e53c0743'),

ObjectId('59ba8f9d4276d283e53c0744'),
ObjectId('59ba8f9d4276d283e53c0745'),
ObjectId('59ba8f9d4276d283e53c0746'),
ObjectId('59ba8f9d4276d283e53c0747'),
ObjectId('59ba8f9d4276d283e53c0748'),
ObjectId('59ba8f9d4276d283e53c0749'),
ObjectId('59ba8f9d4276d283e53c074a'),
ObjectId('59ba8f9d4276d283e53c074b'),
ObjectId('59ba8f9d4276d283e53c074c'),
ObjectId('59ba8f9d4276d283e53c074d'),
ObjectId('59ba8f9d4276d283e53c074e'),
ObjectId('59ba8f9d4276d283e53c074f'),
ObjectId('59ba8f9d4276d283e53c0750'),
ObjectId('59ba8f9d4276d283e53c0751'),
ObjectId('59ba8f9d4276d283e53c0752'),
ObjectId('59ba8f9d4276d283e53c0753'),
ObjectId('59ba8f9d4276d283e53c0754'),
ObjectId('59ba8f9d4276d283e53c0755'),
ObjectId('59ba8f9d4276d283e53c0756'),
ObjectId('59ba8f9d4276d283e53c0757'),
ObjectId('59ba8f9d4276d283e53c0758'),
ObjectId('59ba8f9d4276d283e53c0759'),
ObjectId('59ba8f9d4276d283e53c075a'),
ObjectId('59ba8f9d4276d283e53c075b'),
ObjectId('59ba8f9d4276d283e53c075c'),
ObjectId('59ba8f9d4276d283e53c075d'),
ObjectId('59ba8f9d4276d283e53c075e'),
ObjectId('59ba8f9d4276d283e53c075f'),
ObjectId('59ba8f9d4276d283e53c0760'),
ObjectId('59ba8f9d4276d283e53c0761'),
ObjectId('59ba8f9d4276d283e53c0762'),
ObjectId('59ba8f9d4276d283e53c0763'),
ObjectId('59ba8f9d4276d283e53c0764'),
ObjectId('59ba8f9d4276d283e53c0765'),
ObjectId('59ba8f9d4276d283e53c0766'),
ObjectId('59ba8f9d4276d283e53c0767'),
ObjectId('59ba8f9d4276d283e53c0768'),
ObjectId('59ba8f9d4276d283e53c0769'),
ObjectId('59ba8f9d4276d283e53c076a'),
ObjectId('59ba8f9d4276d283e53c076b'),
ObjectId('59ba8f9d4276d283e53c076c'),
ObjectId('59ba8f9d4276d283e53c076d'),
ObjectId('59ba8f9d4276d283e53c076e'),
ObjectId('59ba8f9d4276d283e53c076f'),
ObjectId('59ba8f9d4276d283e53c0770'),
ObjectId('59ba8f9d4276d283e53c0771'),
ObjectId('59ba8f9d4276d283e53c0772'),
ObjectId('59ba8f9d4276d283e53c0773'),

ObjectId('59ba8f9d4276d283e53c0774'),
ObjectId('59ba8f9d4276d283e53c0775'),
ObjectId('59ba8f9d4276d283e53c0776'),
ObjectId('59ba8f9d4276d283e53c0777'),
ObjectId('59ba8f9d4276d283e53c0778'),
ObjectId('59ba8f9d4276d283e53c0779'),
ObjectId('59ba8f9d4276d283e53c077a'),
ObjectId('59ba8f9d4276d283e53c077b'),
ObjectId('59ba8f9d4276d283e53c077c'),
ObjectId('59ba8f9d4276d283e53c077d'),
ObjectId('59ba8f9d4276d283e53c077e'),
ObjectId('59ba8f9d4276d283e53c077f'),
ObjectId('59ba8f9d4276d283e53c0780'),
ObjectId('59ba8f9d4276d283e53c0781'),
ObjectId('59ba8f9d4276d283e53c0782'),
ObjectId('59ba8f9d4276d283e53c0783'),
ObjectId('59ba8f9d4276d283e53c0784'),
ObjectId('59ba8f9d4276d283e53c0785'),
ObjectId('59ba8f9d4276d283e53c0786'),
ObjectId('59ba8f9d4276d283e53c0787'),
ObjectId('59ba8f9d4276d283e53c0788'),
ObjectId('59ba8f9d4276d283e53c0789'),
ObjectId('59ba8f9d4276d283e53c078a'),
ObjectId('59ba8f9d4276d283e53c078b'),
ObjectId('59ba8f9d4276d283e53c078c'),
ObjectId('59ba8f9d4276d283e53c078d'),
ObjectId('59ba8f9d4276d283e53c078e'),
ObjectId('59ba8f9d4276d283e53c078f'),
ObjectId('59ba8f9d4276d283e53c0790'),
ObjectId('59ba8f9d4276d283e53c0791'),
ObjectId('59ba8f9d4276d283e53c0792'),
ObjectId('59ba8f9d4276d283e53c0793'),
ObjectId('59ba8f9d4276d283e53c0794'),
ObjectId('59ba8f9d4276d283e53c0795'),
ObjectId('59ba8f9d4276d283e53c0796'),
ObjectId('59ba8f9d4276d283e53c0797'),
ObjectId('59ba8f9d4276d283e53c0798'),
ObjectId('59ba8f9d4276d283e53c0799'),
ObjectId('59ba8f9d4276d283e53c079a'),
ObjectId('59ba8f9d4276d283e53c079b'),
ObjectId('59ba8f9d4276d283e53c079c'),
ObjectId('59ba8f9d4276d283e53c079d'),
ObjectId('59ba8f9d4276d283e53c079e'),
ObjectId('59ba8f9d4276d283e53c079f'),
ObjectId('59ba8f9d4276d283e53c07a0'),
ObjectId('59ba8f9d4276d283e53c07a1'),
ObjectId('59ba8f9d4276d283e53c07a2'),
ObjectId('59ba8f9d4276d283e53c07a3'),

ObjectId('59ba8f9d4276d283e53c07a4'),
ObjectId('59ba8f9d4276d283e53c07a5'),
ObjectId('59ba8f9d4276d283e53c07a6'),
ObjectId('59ba8f9d4276d283e53c07a7'),
ObjectId('59ba8f9d4276d283e53c07a8'),
ObjectId('59ba8f9d4276d283e53c07a9'),
ObjectId('59ba8f9d4276d283e53c07aa'),
ObjectId('59ba8f9d4276d283e53c07ab'),
ObjectId('59ba8f9d4276d283e53c07ac'),
ObjectId('59ba8f9d4276d283e53c07ad'),
ObjectId('59ba8f9d4276d283e53c07ae'),
ObjectId('59ba8f9d4276d283e53c07af'),
ObjectId('59ba8f9d4276d283e53c07b0'),
ObjectId('59ba8f9d4276d283e53c07b1'),
ObjectId('59ba8f9d4276d283e53c07b2'),
ObjectId('59ba8f9d4276d283e53c07b3'),
ObjectId('59ba8f9d4276d283e53c07b4'),
ObjectId('59ba8f9d4276d283e53c07b5'),
ObjectId('59ba8f9d4276d283e53c07b6'),
ObjectId('59ba8f9d4276d283e53c07b7'),
ObjectId('59ba8f9d4276d283e53c07b8'),
ObjectId('59ba8f9d4276d283e53c07b9'),
ObjectId('59ba8f9d4276d283e53c07ba'),
ObjectId('59ba8f9d4276d283e53c07bb'),
ObjectId('59ba8f9d4276d283e53c07bc'),
ObjectId('59ba8f9d4276d283e53c07bd'),
ObjectId('59ba8f9d4276d283e53c07be'),
ObjectId('59ba8f9d4276d283e53c07bf'),
ObjectId('59ba8f9d4276d283e53c07c0'),
ObjectId('59ba8f9d4276d283e53c07c1'),
ObjectId('59ba8f9d4276d283e53c07c2'),
ObjectId('59ba8f9d4276d283e53c07c3'),
ObjectId('59ba8f9d4276d283e53c07c4'),
ObjectId('59ba8f9d4276d283e53c07c5'),
ObjectId('59ba8f9d4276d283e53c07c6'),
ObjectId('59ba8f9d4276d283e53c07c7'),
ObjectId('59ba8f9d4276d283e53c07c8'),
ObjectId('59ba8f9d4276d283e53c07c9'),
ObjectId('59ba8f9d4276d283e53c07ca'),
ObjectId('59ba8f9d4276d283e53c07cb'),
ObjectId('59ba8f9d4276d283e53c07cc'),
ObjectId('59ba8f9d4276d283e53c07cd'),
ObjectId('59ba8f9d4276d283e53c07ce'),
ObjectId('59ba8f9d4276d283e53c07cf'),
ObjectId('59ba8f9d4276d283e53c07d0'),
ObjectId('59ba8f9d4276d283e53c07d1'),
ObjectId('59ba8f9d4276d283e53c07d2'),
ObjectId('59ba8f9d4276d283e53c07d3'),

```

ObjectId('59ba8f9d4276d283e53c07d4'),
ObjectId('59ba8f9d4276d283e53c07d5'),
ObjectId('59ba8f9d4276d283e53c07d6'),
ObjectId('59ba8f9d4276d283e53c07d7'),
ObjectId('59ba8f9d4276d283e53c07d8'),
ObjectId('59ba8f9d4276d283e53c07d9'),
ObjectId('59ba8f9d4276d283e53c07da'),
ObjectId('59ba8f9d4276d283e53c07db'),
ObjectId('59ba8f9d4276d283e53c07dc'),
ObjectId('59ba8f9d4276d283e53c07dd'),
ObjectId('59ba8f9d4276d283e53c07de'),
...]
```

0.4 Exploration on the dataset

In the following part I will explore the .json file dataset to find some interesting insights by mon-godb and nosql

```

In [11]: #find the len of the dataset
         collection.find().count()
```

```
Out[11]: 4997286
```

```

In [12]: #number of unique user
         len(collection.group(["created.uid"], {}, {"count":0}, "function(o, p){p.count++}"))
```

```
Out[12]: 1485
```

```

In [13]: collection.find({"type":"node"}).count()
```

```
Out[13]: 4511092
```

```

In [14]: collection.find({"type":"way"}).count()
```

```
Out[14]: 486062
```

Find the user name who commit the most of data entry and the number.

```

In [15]: import pprint
```

```

pipeline = [{"$group":{"_id": "$created_by",
                        "count": {"$sum": 1}}}]
pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))
```

```

[{u'_id': u'polyshp2osm-multipoly', u'count': 474},
 {u'_id': u'Potlatch 0.10b', u'count': 2},
 {u'_id': u'Potlatch 0.10e', u'count': 58},
 {u'_id': u'Potlatch 0.10', u'count': 12},
 {u'_id': u'OSM Fixer', u'count': 2},
 {u'_id': u'Potlatch 0.9a', u'count': 8},
```

```
{u'_id': u'Merkaartor 0.12', u'count': 28},
{u'_id': u'Potlatch 0.10d', u'count': 62},
{u'_id': u'Potlatch 0.9c', u'count': 90},
{u'_id': None, u'count': 4995038},
{u'_id': u'Potlatch 0.10f', u'count': 86},
{u'_id': u'JOSM', u'count': 1426}]
```

```
In [17]: #top 10 users who have the most contribution
pipeline = [{"$group":{"_id": "$created.user",
                      "count": {"$sum": 1}}},
            {"$sort": {"count": -1}},
            {"$limit": 10}]
pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))
```

```
[{u'_id': u'GeoKitten', u'count': 646394},
 {u'_id': u'woodpeck_fixbot', u'count': 379988},
 {u'_id': u'balcoath', u'count': 354914},
 {u'_id': u'rickmastfan67', u'count': 234590},
 {u'_id': u'abbafei', u'count': 230168},
 {u'_id': u'Gary Hayden', u'count': 187898},
 {u'_id': u'Fredlyfish4', u'count': 156350},
 {u'_id': u'dchiles', u'count': 139330},
 {u'_id': u'AndrewSnow', u'count': 137950},
 {u'_id': u'aps', u'count': 121474}]
```

```
In [18]: #find the most 10 popular cuisne/ restraurants in the city
pipeline = [{"$match":{"amenity":{"$exists":1}, "amenity":"restaurant", "cuisine":{"$exists":1}},
            {"$group":{"_id":"$cuisine", "count":{"$sum":1}}},
            {"$sort":{"count":-1}},
            {"$limit":5}]
pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))
```

```
[{u'_id': u'american', u'count': 324},
 {u'_id': u'pizza', u'count': 280},
 {u'_id': u'italian', u'count': 142},
 {u'_id': u'chinese', u'count': 100},
 {u'_id': u'mexican', u'count': 70}]
```

```
In [19]: #find the proportion of the top user contributions
pipeline = [{"$group":{"_id": "$created.user",
                      "count": {"$sum": 1}}},
            {"$project": {"proportion": {"$divide" :["$count",collection.find().count()]}},
            {"$sort": {"proportion": -1}},
            {"$limit": 3}]
pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))
```

```
[{'_id': 'GeoKitten', 'proportion': 0.12934901064297702},
 {'_id': 'woodpeck_fixbot', 'proportion': 0.07603887390075333},
 {'_id': 'balcoath', 'proportion': 0.07102135038899114}]
```

In [21]: *#find the number of hospital in pittsburgh*

```
pipeline = [{"$match":{"amenity":{"$exists":1}, "amenity": "hospital", "name":{"$exists":1}},
             {"$group":{"_id":"$name", "count":{"$sum":1}}},
             {"$sort":{"count":1}}]
pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))
```

```
[{'_id': 'Trinity Medical Center West', 'count': 2},
 {'_id': 'UPMC Mercy', 'count': 2},
 {'_id': 'Weirton Medical Center', 'count': 2},
 {'_id': 'Med Express', 'count': 2},
 {'_id': 'Allegheny University-Forbes Regional Hospital', 'count': 2},
 {'_id': 'UPMC East', 'count': 2},
 {'_id': 'Dr. Kamlesh B. Gosai, MD', 'count': 2},
 {'_id': 'Monongahela Valley Hospital', 'count': 2},
 {'_id': 'Southwood Psychiatric Hospital', 'count': 2},
 {'_id': 'Heritage Valley Health System', 'count': 2},
 {'_id': 'Canonsburg General Hospital', 'count': 2},
 {'_id': 'Allegheny General Hospital', 'count': 2},
 {'_id': 'UPMC St. Margaret', 'count': 2},
 {'_id': 'UPMC Passavant \u2013 Cranberry', 'count': 2},
 {'_id': 'Childrens Hospital of Pittsburgh East', 'count': 2},
 {'_id': 'Frick Hospital & Community Health Center', 'count': 2},
 {'_id': 'Hillview Sanatorium', 'count': 2},
 {'_id': 'Armstrong County Memorial Hospital', 'count': 2},
 {'_id': 'Trinity Medical Center East', 'count': 2},
 {'_id': 'Westmoreland Regional Hospital', 'count': 2},
 {'_id': 'Childrens Rehabilitation Center', 'count': 2},
 {'_id': 'Kindred Hospital', 'count': 2},
 {'_id': 'Lifecare Hospital of Pittsburgh', 'count': 2},
 {'_id': 'UPMC McKeesport Hospital', 'count': 2},
 {'_id': 'Wheeling Hospital', 'count': 2},
 {'_id': 'Jeanette District Memorial Hospital', 'count': 2},
 {'_id': 'Butler Memorial Hospital', 'count': 2},
 {'_id': 'Sewickley Valley Hospital', 'count': 2},
 {'_id': 'UPMC Oakland', 'count': 2},
 {'_id': 'Latrobe Hospital', 'count': 2},
 {'_id': 'H. John Heinz III Progressive Care Center', 'count': 2},
 {'_id': 'Morarthy Consultants', 'count': 2},
 {'_id': 'East Liverpool City Hospital', 'count': 2},
 {'_id': 'Ellwood City Hospital', 'count': 2},
 {'_id': 'Citizens General Hospital', 'count': 2},
 {'_id': 'UPMC Rehabilitation Hospital', 'count': 2},
 {'_id': 'Western Psychiatric Hospital', 'count': 2},
```



```
{u'_id': u'Children's Hospital of Pittsburgh of UPMC', u'count': 2},
{u'_id': u'Deshon Veterans Administration Hospital', u'count': 2},
{u'_id': u'Columbia Hospital', u'count': 2},
{u'_id': u'Jefferson Hospital', u'count': 2},
{u'_id': u'Pittsburgh Tuberculosis Sanitarium', u'count': 2},
{u'_id': u'Ohio Valley General Hospital', u'count': 2},
{u'_id': u'Charleroi Monessen Hospital', u'count': 2},
{u'_id': u'Childrens Hospital of Pittsburgh North', u'count': 2},
{u'_id': u'Providence Point', u'count': 2},
{u'_id': u'Cat Hospital of North Hills', u'count': 2},
{u'_id': u'UPMC Shadyside', u'count': 2},
{u'_id': u'MedExpress Urgent Care', u'count': 2},
{u'_id': u'Allegheny Valley Hospital', u'count': 2},
{u'_id': u'Mountain View Medical Park', u'count': 2},
{u'_id': u'Washington Hospital', u'count': 2},
{u'_id': u'Western Pennsylvania Hospital', u'count': 2},
{u'_id': u'Childrens Hospital of Pittsburgh South', u'count': 2},
{u'_id': u'Magee-Womens Hospital', u'count': 2},
{u'_id': u'Veterans Administration Hospital', u'count': 4}]
```

find the number of data entry

```
In [23]: #number of the data entry contributor
        pipeline = [{"$group":{"_id": "$created_by",
                                "count": {"$sum": 1}}}]
        pprint.pprint(list(db.pittsburgh_metro.aggregate(pipeline)))

[{u'_id': u'polyshp2osm-multipoly', u'count': 474},
{u'_id': u'Potlatch 0.10b', u'count': 2},
{u'_id': u'Potlatch 0.10e', u'count': 58},
{u'_id': u'Potlatch 0.10', u'count': 12},
{u'_id': u'OSM Fixer', u'count': 2},
{u'_id': u'Potlatch 0.9a', u'count': 8},
{u'_id': u'Merkaartor 0.12', u'count': 28},
{u'_id': u'Potlatch 0.10d', u'count': 62},
{u'_id': u'Potlatch 0.9c', u'count': 90},
{u'_id': None, u'count': 4995038},
{u'_id': u'Potlatch 0.10f', u'count': 86},
{u'_id': u'JOSM', u'count': 1426}]
```