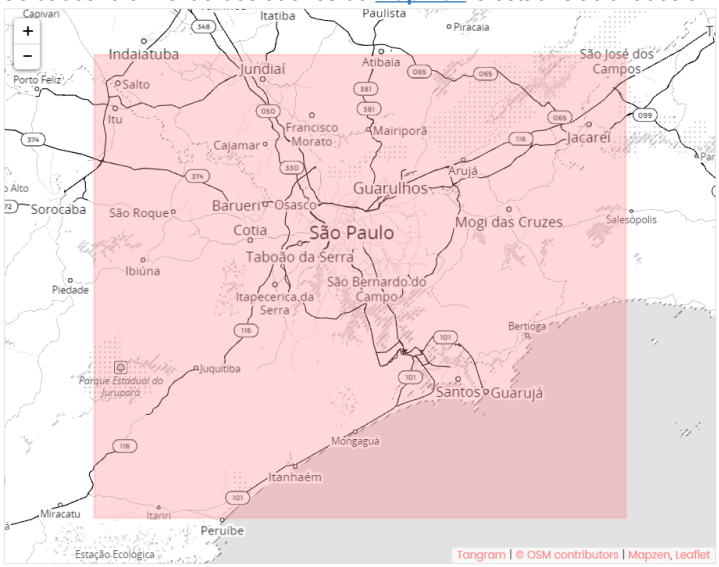
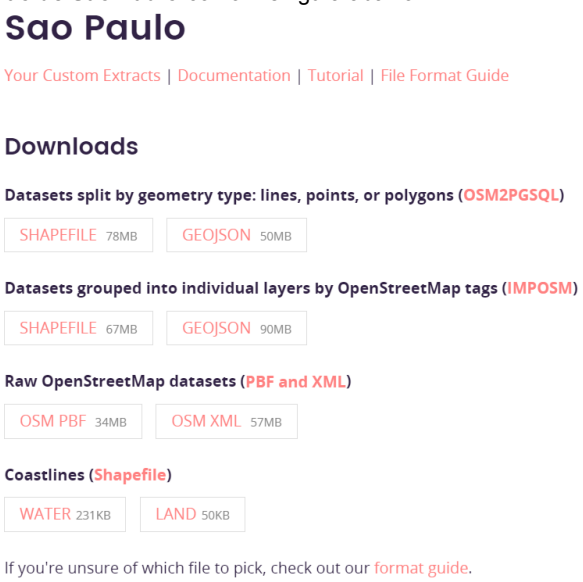
OpenStreetMap Sample Project

Data Wrangling with MongoDB

Fernando Teixeira

Os dados foram extraídos através do [Map Zen](Map Zen) e estão relacionados a **região de São Paulo** conforme figura abaixo:





Problemas encontrados

1.Processamento

1.Tamanho do arquivo
2.Descrições inconsistentes para mesmo grupo de informação
3.Grupo de informação sem descrição correta do tipo de informação
4.Nomes das Cidades

A.Tamanho do arquivo

O tamanho do arquivo dificulta a análise em um primeiro, sendo necessário a criação de uma estratégia para trabalhar com todos os dados.

> Arquivo possui 58 MB compactado e mais de 800 MB descompactado
> Arquivo possui 12.537.231 milhões de linhas
> Em torno de 1.700.000 tags

Assim o acesso a informação teve ser realizado linha a linha, assim como os tratamentos para fazer a conversão de XML to JSON. A função foi utilizado para gerar a conexão com o arquivo de dados evitando descompactar o arquivo.

```
def open_files():
    filename_zip = 'sao-paulo_brazil.zip'
    filename_osm = 'sao-paulo_brazil.osm'


    try:
        myzip.close()
        myfile.close()
    except:
        pass
```

```
myzip = zipfile.ZipFile(filename_zip)
myfile = myzip.open(filename_osm)
return myzip, myfile
```

B.Descrições inconsistentes para mesmo grupo de informação

Antes de importar os dados para MongoDB os seguintes campos foram padronizados para manter a consistência da informação dentro do banco de dados.

Campos inconsistentes:

> **addr** e **contact** são campos que descrevem informações gerais com um detalhamento como prefixo como, 'addr:city', 'addr:country', 'addr:district','contact:email', 'contact:facebook', 'contact:fax', assim campos do tipo 'addr' e 'contact' são considerados **inconsistentes** e não foram enviados para o MongoDB
>
> Detalhamento incorretos como 'addr:district:name', ao invés de 'addr:district' também foram considerados **inconsistentes** e portanto não passaram pela auditoria dadose e por consequência também não foram enviados para o MongoDB

C.Grupo de informação sem descrição correta do tipo de informação¶

Nas tags do arquivo XML temos que algumas **keys** contém descrições inconsistentes como highway, highway_1, highway_2, leisure_1, leisure_2, leisure_3, natural, natural_1, natural_2.

Os campos que possuem prefixos podem ser colocados como pertencentes ao um único grupo, como exemplo:

> Highway_1 possui os seguintes valores encontrados {'footway', 'redlight_camera', 'speed_camera', 'traffic_signals'}
> Highway_2 possui os seguintes valores encontrados {'speed_camara'} (erro de grafia)

Portanto, podemos fazer highway_1 e highway_2 serem armazenadas como highway, ou seja, foi necessáirio realizar o **cleaning data** antes de dar upload to MongoDB.

Essa abordagem é utilizada nas demais inconsistências citadas.

D.Nomes das Cidades

```
db.osm.distinct('addr.city')
```

Usando o consulta acima combinado com um conjunto é possível encontrar que, por exemplo, a cidade de **São Paulo** possui nomes como 'Sao paulo', 'SÃO PAULO', 'Sâo Paulo', 'São Paulo', 'São Paulol', 'São paulo', 'sao paulo', 'são Paulo', 'são paulo'.

Vemos que existem erros de grafia.

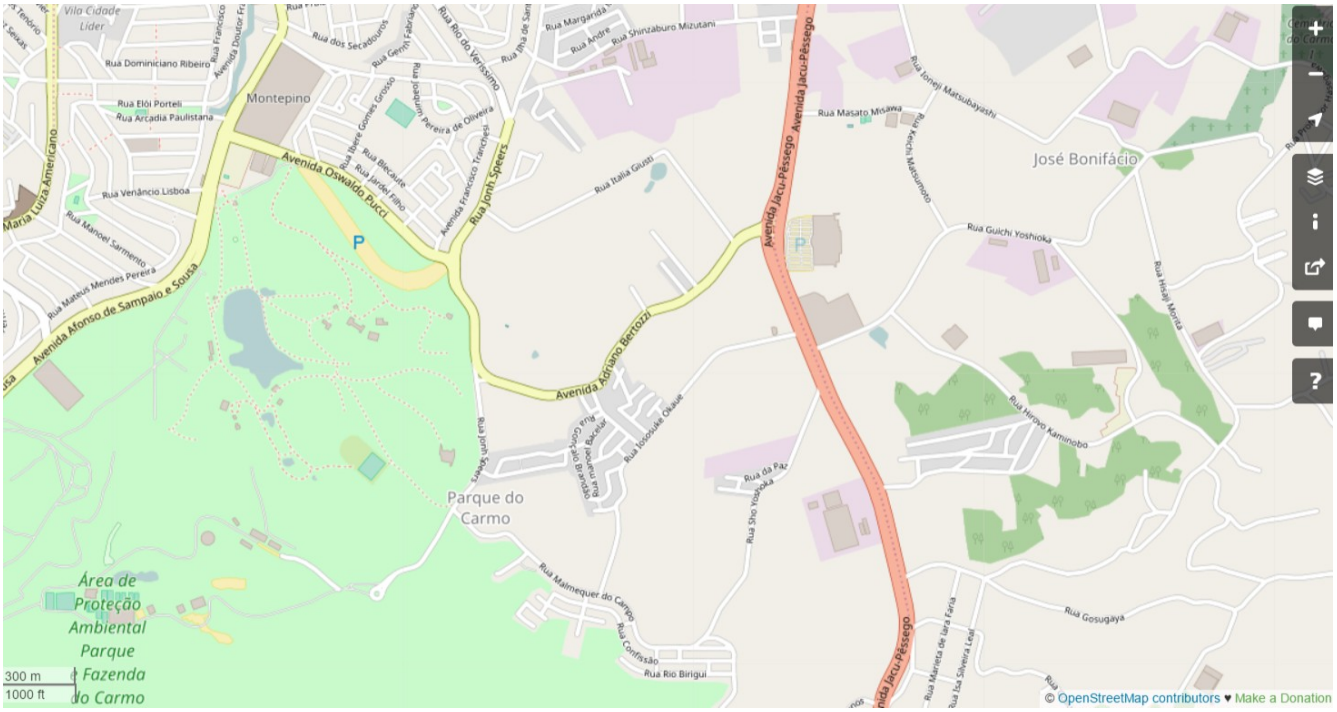
Existe o caso da cidade receber o nome de **'sp'** o que não garante que se trata da cidade de São Paulo.
Usando a consulta do MongoDB:

```
db.osm.find_one({'addr.city':'sp'})
```

Encontrei uma **única** ocorrência:

```
{'_id': ObjectId('58cc96c6c1a45f129c02c59d'),
        'addr': {'city': 'sp', 'housenumber': '1195', 'postcode': '08265-000', 'street': 'Avenida Adriano Bertozzi', 'suburb': 'Jardim
        Helian'},
        'changeset': '35744724',
        'id': '3876083747',
         'pos': [-46.457189, -23.5747499],
        'timestamp': '2015-12-04T11:05:26Z',
        'type': 'node',
        'uid': '3433372',
         'user': 'thainna', 'version': '1'}
```

Usando a rua descrita, temos que se trata realmente da cidade de São Paulo (imagem abaixo).



A seguinte Query faz o **Update** de todos os campos e traz **consistência** a base de dados.

```
for i,right in enumerate(['Sao paulo', 'SÃO PAULO', 'São Paulo', 'São Paulo', 'São Paulol', 'São paulo', 'sao paulo',
                          'são Paulo', 'são paulo','sp']):

    db.osm.update_many({'addr.city':i},{'$set':{'addr.city':'São Paulo'}})
    sys.stdout.write('\rIteracao:' + str(i))
```

Uma das cidades tem nome '06097-100' ou 'Dona Catarina, Mairinque' e nesses caso foi deletada a primeira foi deletada e a segunda corrigida.

Visão geral dos Dados

Quantidade de documentos

```
>>>db.osm.find().count()
4539325
```

Quantidade de nodes

```
>>>db.osm.find({'type':'node'}).count()
3987104
```

Quantidade de ways

```
>>>db.osm.find({'type':'way'}).count()
552221
```

Quantidade de users¶

```
>>>len(db.osm.distinct('user'))
2158
```

Cidade que é mais citada em endereços

```
>>>db.osm.aggregate([{'$group':{'_id':'$addr.city','count':{'$sum':1}}},{'$sort':{'count':-1}}])
{'_id': None, 'count': 4512415}
{'_id': 'São Paulo', 'count': 21580}
{'_id': 'São Bernardo do Campo', 'count': 2259}
```

Usuários com a maior contribuição

```
>>>db.osm.aggregate([{'$group':{'_id':'$user','count':{'$sum':1}}},{'$sort':{'count':-1}}])
{'_id': 'Bonix-Mapper', 'count': 2343124}
{'_id': 'AjBelnuovo', 'count': 250691}
{'_id': 'cxs', 'count': 192443}
```

Vemos que as maiores contribuições de informação não incluem a cidade.

Ideias adicionais

Vemos que a maioria dos usuários contribuem sem colocar o endereço da cidade, então que seriam os grandes contribuidores para identificação correta da cidade?

Maior contribuidor de informações que contém a cidade no endereço

```
>>>db.osm.aggregate([{'$match':addr.city':{'$exists':1}}},
{'$project':city':'$addr.city','user':'$user'}},
{'$group': {'_id':{'Usuario':{'user':'$user'},'Cidade':{'city':'$city'}},'count':
       {'$sum':1}}},{'$sort':{'count':-1}},
         {'$limit':5}])
{'_id': {'Usuario': {'user': 'O Fim'}, 'Cidade': {'city': 'São Paulo'}}, 'count': 16376}
{'_id': {'Usuario': {'user': 'Bonix-Mapper'}, 'Cidade': {'city': 'São Paulo'}}, 'count': 2887}
{'_id': {'Usuario': {'user': 'AjBelnuovo'}, 'Cidade': {'city': 'São Bernardo do Campo'}}, 'count':
1853}
{'_id': {'Usuario': {'user': 'naoliv'}, 'Cidade': {'city': 'São Paulo'}}, 'count': 556}
{'_id': {'Usuario': {'user': 'AjBelnuovo'}, 'Cidade': {'city': 'São Paulo'}}, 'count': 295}
```

Vemos que o maior na identificação da cidade nas tags que contém endereço é 'O Fim', ao contrário da contribuição geral onde o principal é 'Bonix-Mapper' que pelo tamanho da contribuição realmente dedicou muito tempo.

O usário 'Bonix-Mapper' possui segundo a query abaixo:

```
>>>db.osm.aggregate([{'$group': {'_id':'$user', 'count':{'$sum':1}}},{'$sort':{'count':-1}}])
{'_id': 'Bonix-Mapper', 'count': 2343124}
{'_id': 'AjBelnuovo', 'count': 250691}
```

'Bonix-Mapper' fez 2343124 contribuições, enquanto o segundo colocado 250691, ou seja, 9.34 vezes mais contribuições que se concentram em criar nodes e ways, mas não no detalhamento dessas tags e assim aparece com poucas tags que tem a cidade como endereço. Em geral os usuários se concetram em marcar elementos das regiões mas não em detalhar cada região com dados macros como cidade e até rua, se fizermos uma consulta sobre as contribuições para nomes de ruas:

```
>>>db.osm.aggregate([{'$match':
                {'addr.street':{'$exists':1}}},
               {'$project':{'rua':'$addr.street','user':'$user'}},
               {'$group':{'_id':{'Usuario':{'user':'$user'},'Rua':{'Rua':'$rua'}},'count':
             {'$sum':1}}},{'$sort':{'count':-1}},
               {'$limit':5}])

{'_id': {'Rua': {'Rua': 'Avenida Conceição'}, 'Usuario': {'user': 'O Fim'}}, 'count': 494}
{'_id': {'Rua': {'Rua': 'Rua Oriente'}, 'Usuario': {'user': 'O Fim'}}, 'count': 354}
{'_id': {'Rua': {'Rua': 'Rua Maria Marcolina'}, 'Usuario': {'user': 'O Fim'}}, 'count': 324}
{'_id': {'Rua': {'Rua': 'Rua Bresser'}, 'Usuario': {'user': 'O Fim'}}, 'count': 277}
{'_id': {'Rua': {'Rua': 'Rua Miller'}, 'Usuario': {'user': 'O Fim'}}, 'count': 220}
```

O usuário 'O Fim' é o maior contribuidor para identificação de ruas, mas a quantidade é bem pequena perto da quantidade de nodes que ele criou.

Conclusão

A contribuição de poucos usuários e a falta de padrão de preenchimento criam um problema para o aumento e qualidade do OpenStreetMap.
Uma query que use os dados de localização latitude e longitude poderia adicionar aos endereços as cidades correspondentes e outra query poderia analisar o nome das cidades e corrigir erros de grafia.

Assim o OpenStreetMap poderia ser mais chamativo para o grande público, ao invés das contribuições virem de um pequeno grupo.