Animação da mobilidade dos trajectos na tabela tracks

Tópicos Avançados de Bases de Dados

O módulo python-postgis

https://github.com/tilery/python-postgis

python-postgis

PostGIS helpers for psycopg2 and asyncpg.

Install

`pip install postgis`

If you want a compiled version, first install `cython`:

```
pip install cython
pip install postgis
```

Utilização

- Registar a extensão

```
# With psycopg2
> from postgis.psycopg import register
> register(connection)

# With asyncpg
> from postgis.asyncpg import register
> await register(connection)
```

Then you can pass python geometries instance to psycopg:

```
> cursor.execute('INSERT INTO table (geom) VALUES (%s)', [Point(x=1, y=2, srid=4326)])
```

And retrieve data as python geometries instances:

```
> cursor.execute('SELECT geom FROM points LIMIT 1')
> geom = cursor.fetchone()[0]
> geom
<Point POINT(1.0 2.0)>
```

Exemplo com o psycopg2

```
> import psycopg2
> from postgis import LineString
> from postgis.psycopg import register
> db = psycopg2.connect(dbname="test")
> register(db)
> cursor.execute('CREATE TABLE IF NOT EXISTS mytable ("geom" geometry(LineString) NOT NULL)')
> cursor.execute('INSERT INTO mytable (geom) VALUES (%s)', [LineString([(1, 2), (3, 4)], srid=4326)])
> cursor.execute('SELECT geom FROM mytable LIMIT 1')
> geom = cursor.fetchone()[0]
> geom
<LineString LINESTRING(1.0 2.0, 3.0 4.0)>
> geom[0]
<Point POINT(1.0 2.0)>
> geom.coords
((1.0, 2.0), (3.0, 4.0))
> geom.geojson
{'coordinates': ((1.0, 2.0), (3.0, 4.0)), 'type': 'LineString'}
> str(geom.geojson)
'{"type": "LineString", "coordinates": [[1, 2], [3, 4]]}'
```

Animação de todos os trajectos

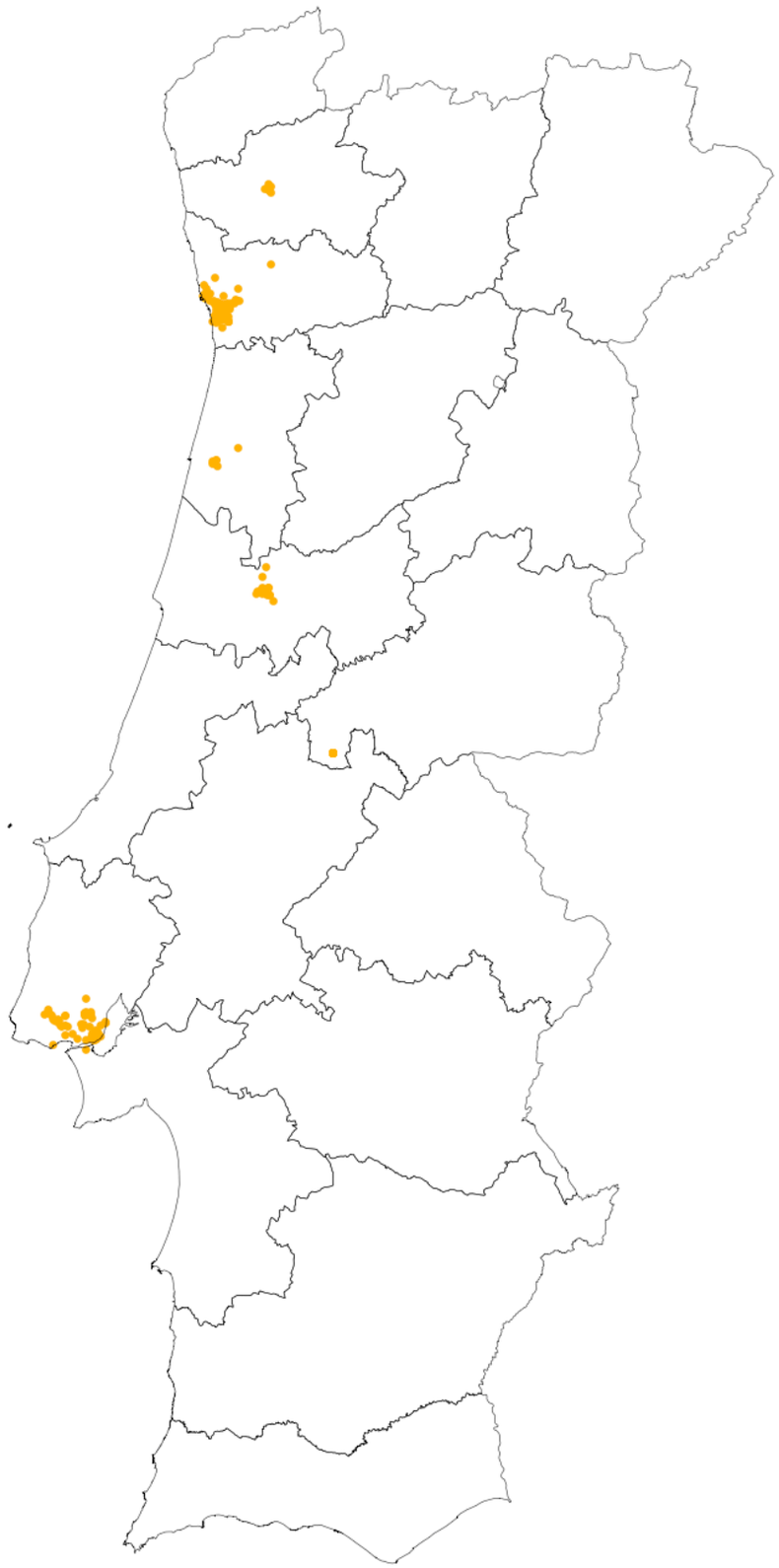
- Separar a geração dos offsets e o código de animação
  - generate_offsets.py
  - tracks_animation.py

generate_offsets.py

```
import numpy as np
import matplotlib.pyplot as plt
import psycopg2
import math
from matplotlib.animation import FuncAnimation
import datetime
from postgis import Polygon,MultiPolygon
from postgis.psycopg import register


#define the step in seconds of the animation
step = 10


conn = psycopg2.connect("dbname=michelferreira user=michelferreira")
register(conn)
cursor_psql = conn.cursor()

sql = """select distinct taxi from tracks order by 1"""
cursor_psql.execute(sql)
results = cursor_psql.fetchall()

taxis_x ={}
taxis_y ={}

ts_i = 1570665600
ts_f = 1570667000

array_size = int(24*60*60/step)

for row in results:
    taxis_x[int(row[0])] = np.zeros(array_size)
    taxis_y[int(row[0])] = np.zeros(array_size)
```

```
for i in range(ts_i,ts_f,10):
    sql = "select taxi,st_pointn(proj_track," + str(i) + "-ts) from tracks where ts<" + str(i) + " and ts+st_numpoints(proj_track)>" + str(i)
    cursor_psql.execute(sql)
    results = cursor_psql.fetchall()
    for row in results:
        x,y = row[1].coords
        taxis_x[int(row[0])][int((i-ts_i)/10)] = x
        taxis_y[int(row[0])][int((i-ts_i)/10)] = y


offsets = []
for i in range(array_size):
    l = []
    for j in taxis_x:
        l.append([taxis_x[j][i],taxis_y[j][i]])
    offsets.append(l)

for i in offsets:
    print("%f %f" %(i[0][0],i[0][1]),end='')
    for j in range(1,len(i)):
        print(",%f %f" %(i[j][0],i[j][1]),end='')
    print("")
conn.close()
```

2019-10-10 00:34:40



tracks_animation.py

```
import numpy as np
import matplotlib.pyplot as plt
import psycopg2
import math
from matplotlib.animation import FuncAnimation
import datetime
import csv
from postgis import Polygon,MultiPolygon
from postgis.psycopg import register

def animate(i):
    ax.set_title(datetime.datetime.utcfromtimestamp(ts_i+i*10))
    scat.set_offsets(offsets[i])

scale=1/3000000
conn = psycopg2.connect("dbname=michelferreira user=michelferreira")
register(conn)

xs_min, xs_max, ys_min, ys_max = -120000, 165000, -310000, 285000
width_in_inches = (xs_max-xs_min)/0.0254*1.1
height_in_inches = (ys_max-ys_min)/0.0254*1.1

fig, ax = plt.subplots(figsize=(width_in_inches*scale, height_in_inches*scale))
ax.axis('off')
ax.set(xlim=(xs_min, xs_max), ylim=(ys_min, ys_max))

cursor_psql = conn.cursor()
```

```
sql = "select distrito,st_union(proj_boundary) from cont_aad_caop2018 group by distrito"

cursor_psql.execute(sql)
results = cursor_psql.fetchall()
xs , ys = [],[]
for row in results:
    geom = row[1]
    if type(geom) is MultiPolygon:
        for pol in geom:
            xys = pol[0].coords
            xs, ys = [],[]
            for (x,y) in xys:
                xs.append(x)
                ys.append(y)
            ax.plot(xs,ys,color='black',lw='0.2')
    if type(geom) is Polygon:
        xys = geom[0].coords
        xs, ys = [],[]
        for (x,y) in xys:
            xs.append(x)
            ys.append(y)
        ax.plot(xs,ys,color='black',lw='0.2')
```

```
offsets = []
with open('offsets3.csv', 'r') as csvFile:
    reader = csv.reader(csvFile)
    i = 0
    for row in reader:
        l = []
        for j in row:
            x,y = j.split()
            x = float(x)
            y= float(y)
            l.append([x,y])
        offsets.append(l)
```

```
x,y = [],[]
for i in offsets[0]:
    x.append(i[0])
    y.append(i[1])

scat = ax.scatter(x,y,s=2,color='orange')
anim = FuncAnimation(
    fig, animate, interval=10, frames=len(offsets)-1, repeat = False)

plt.draw()
plt.show()
```