Computação Urbana:
Técnicas para o Estudo de Sociedades com Redes de Sensoriamento Participativo

Thiago H Silva
Antonio A F Loureiro

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

UFMG

Sobre mim

Thiago H Silva

Professor da UTFPR (Campus Curtiba)

Sobre mim

Thiago H Silva

Professor da UTFPR (Campus Curtiba)

Apaixonado por:

Sobre mim

Thiago H Silva

Professor da UTFPR (Campus Curtiba)

Apaixonado por:

Pessoas

Sobre mim

Thiago H Silva

Professor da UTFPR (Campus Curtiba)

Apaixonado por:

Pessoas

Culturas

Tópicos

Computação urbana

Algumas perguntas:
O que é?
Qual a sua importância?

Tópicos

Computação urbana

Redes de sensoriamento participativo

Algumas perguntas:
Como utilizar?
Quais as propriedades?

Tópicos

Computação urbana

Redes de sensoriamento participativo

Técnicas comumente utilizadas

Apresentar ponteiros que podem ser úteis em pesquisas nesta área

Tópicos

Computação urbana

Redes de sensoriamento participativo

Técnicas comumente utilizadas

Desafios e oportunidades

Tópicos

Computação urbana

Rede

Objetivo central:
Motivar a construção de novas aplicações e pesquisas na área de computação urbana

Técnicas comumente utilizadas

Desafios e oportunidades

Cidades

Mais de 50% da população do mundo hoje vive em cidades

Cidades

Sistemas não triviais

Cidades

Muitos desafios!

Cidades

Muitas oportunidades!

Computação urbana

Processo de aquisição, integração e análise de um grande volume de dados heterogêneos de fontes urbanas diversas

Computação urbana

Processo de aquisição, integração e análise de um grande volume de dados heterogêneos de fontes urbanas diversas

Ajudar na solução de diversos problemas das cidades

Melhorar a qualidade de vida das pessoas que vivem em ambientes urbanos

Computação urbana

Auxilia a compreender a natureza dos fenômenos urbanos e prever o futuro das cidades.

Fusão de diversas áreas

- Ciência da Computação
- Transportes
- Economia
- Sociologia

Computação urbana

Estudo e tratamento de questões enfrentadas pelas cidades utilizando tecnologia de computação

Ciência da Computação

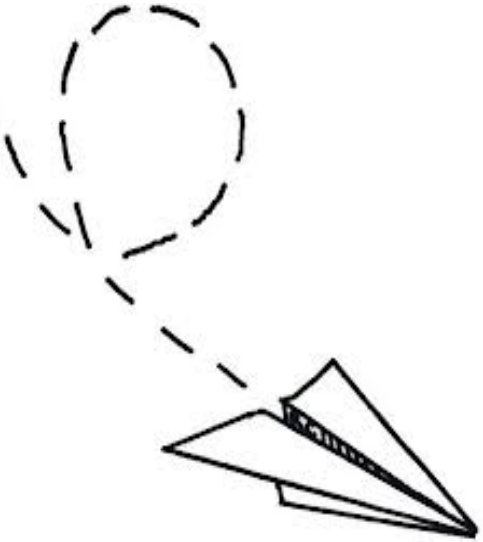
- Sistemas distribuídos
- Inteligência artificial
- Interação humano - computador
- Redes de computadores / sensores
- Sistemas cooperativos

Aplicação

Alice gosta de frequentar Camden Town - Londres

Aplicação

Alice gosta de frequentar
Camden Town - Londres



SÃO PAULO

Aplicação

Alice gosta de frequentar
Camden Town - Londres

Augusta

SÃO PAULO

Aplicação

Voltando pra casa

Aplicação

Voltando pra casa

Acidente na rota planejada

Aplicação

Voltando pra casa

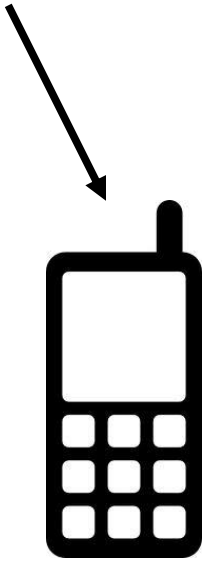
O carro identifica o problema e traça uma nova rota

Aplicação

A área que você planejava jantar e beber estava completamente lotada ☹

Aplicação

Aplicação

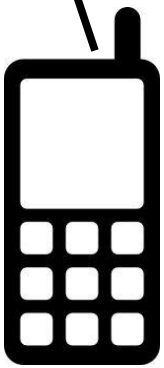


Verificar outras áreas menos cheias

Aplicação

Situação em tempo real

Você está aqui



7km

7,5km

Passo fundamental é o sensoriamento do ambiente

Passo fundamental é o sensoriamento do ambiente

Nos exemplos citados, já temos os sensores?

Arcabouço da Computação Urbana

Arcabouço da Computação Urbana

Fontes de Dados Urbanos

Dados estatísticos oficiais: estudo estatístico sobre uma população, como dados demográficos, econômicos e sociais

Fontes de Dados Urbanos

Dados estatísticos oficiais: estudo estatístico sobre uma população, como dados demográficos, econômicos e sociais

Diversas fontes de dados na Web

Fontes de Dados Urbanos

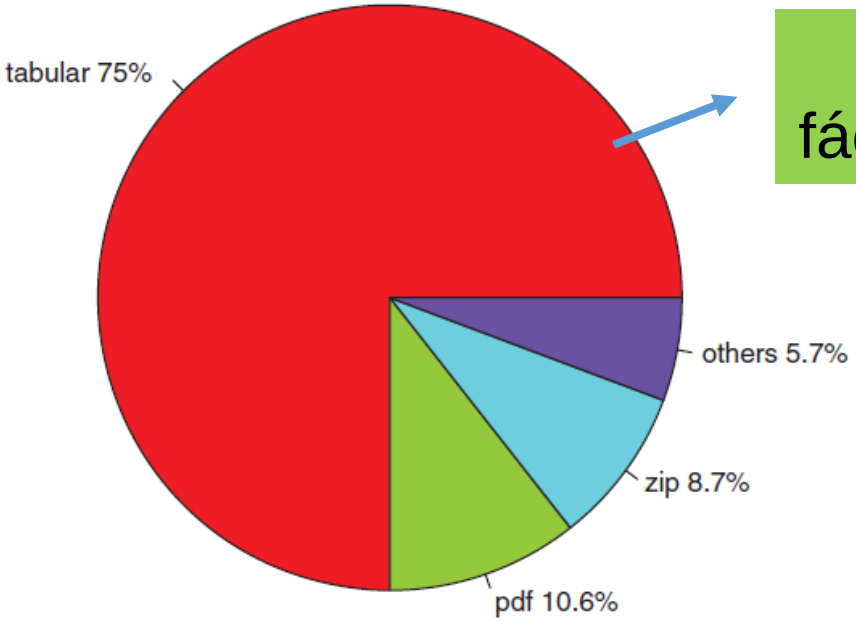
Nem sempre esses dados estão disponíveis para a localidade que se deseja estudar

Grande diversidade de formatos: tabelas, mapas, gráficos, formulários, etc

Fontes de Dados Urbanos

Nem sempre esses dados estão disponíveis para a localidade que se deseja estudar

Grande diversidade de formatos: tabelas, mapas, gráficos, formulários, etc



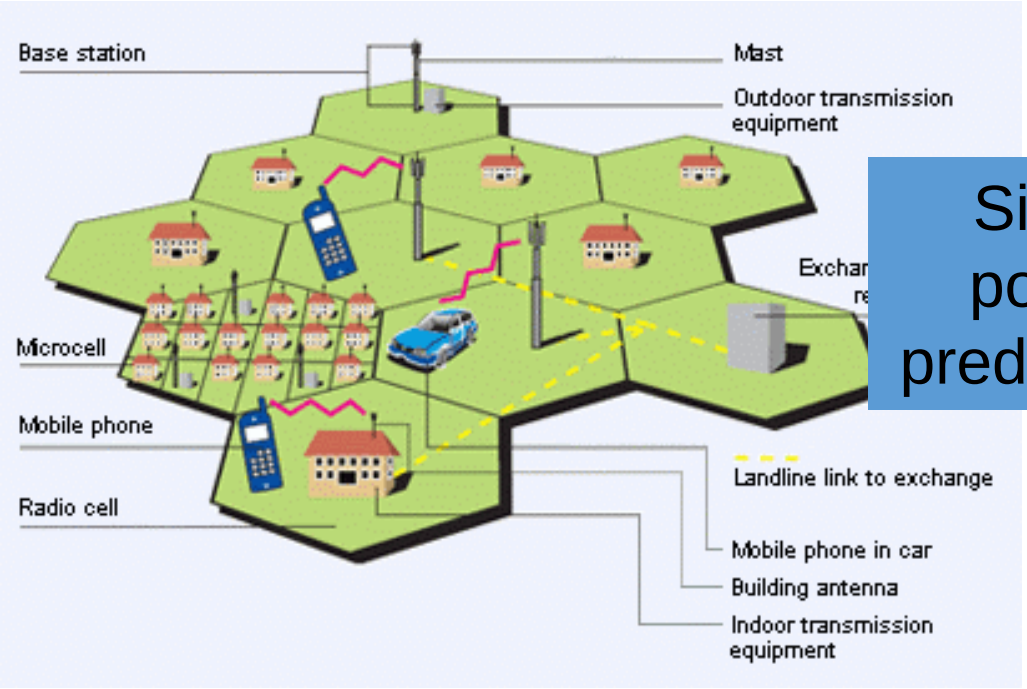
Muitos são de fácil interpretação

Fontes de Dados Urbanos

Infraestrutura das cidades: dados das infraestruturas existentes da cidade, que são criadas para outros propósitos.

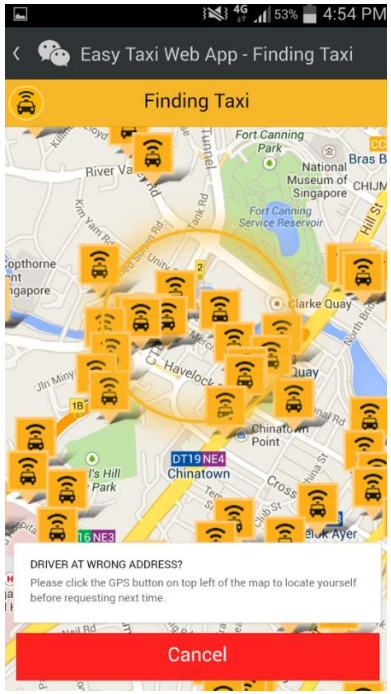
Fontes de Dados Urbanos

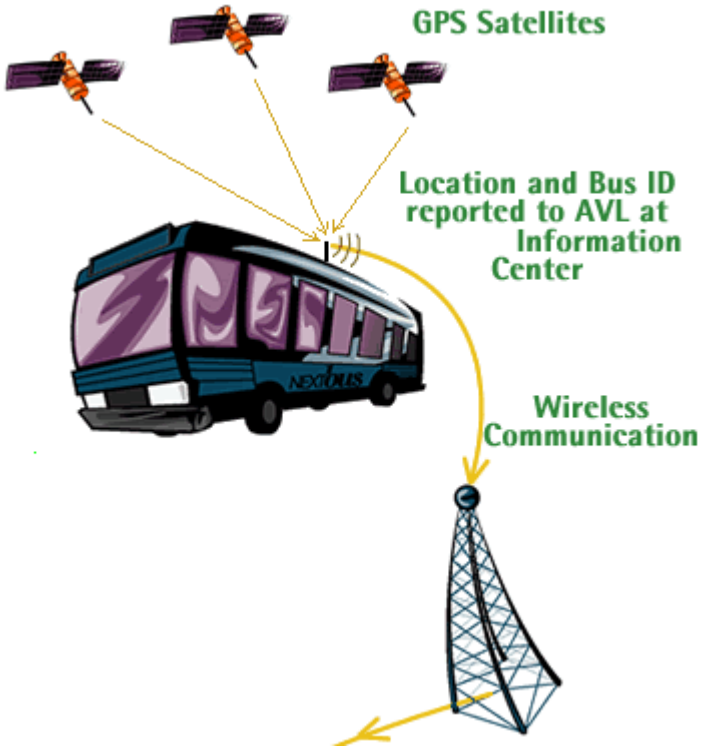
Infraestrutura das cidades: dados das infraestruturas existentes da cidade, que são criadas para outros propósitos.



Sinais dos telefones celulares podem ser usados para tentar predizer a mobilidade dos usuários

Fontes de Dados Urbanos





Localização de veículos que possuem GPS

Possibilita, por exemplo, o entendimento do tráfego

Fontes de Dados Urbanos

Dados de utilização do sistema de transporte público

É comum utilizar cartões RFID para registrar o uso de ônibus e metrô

Fontes de Dados Urbanos

A dificuldade de acesso a dados é também um problema dessa fonte

Somente a prefeitura ou empresas responsáveis, tipicamente, possuem acesso a esse tipo de dado

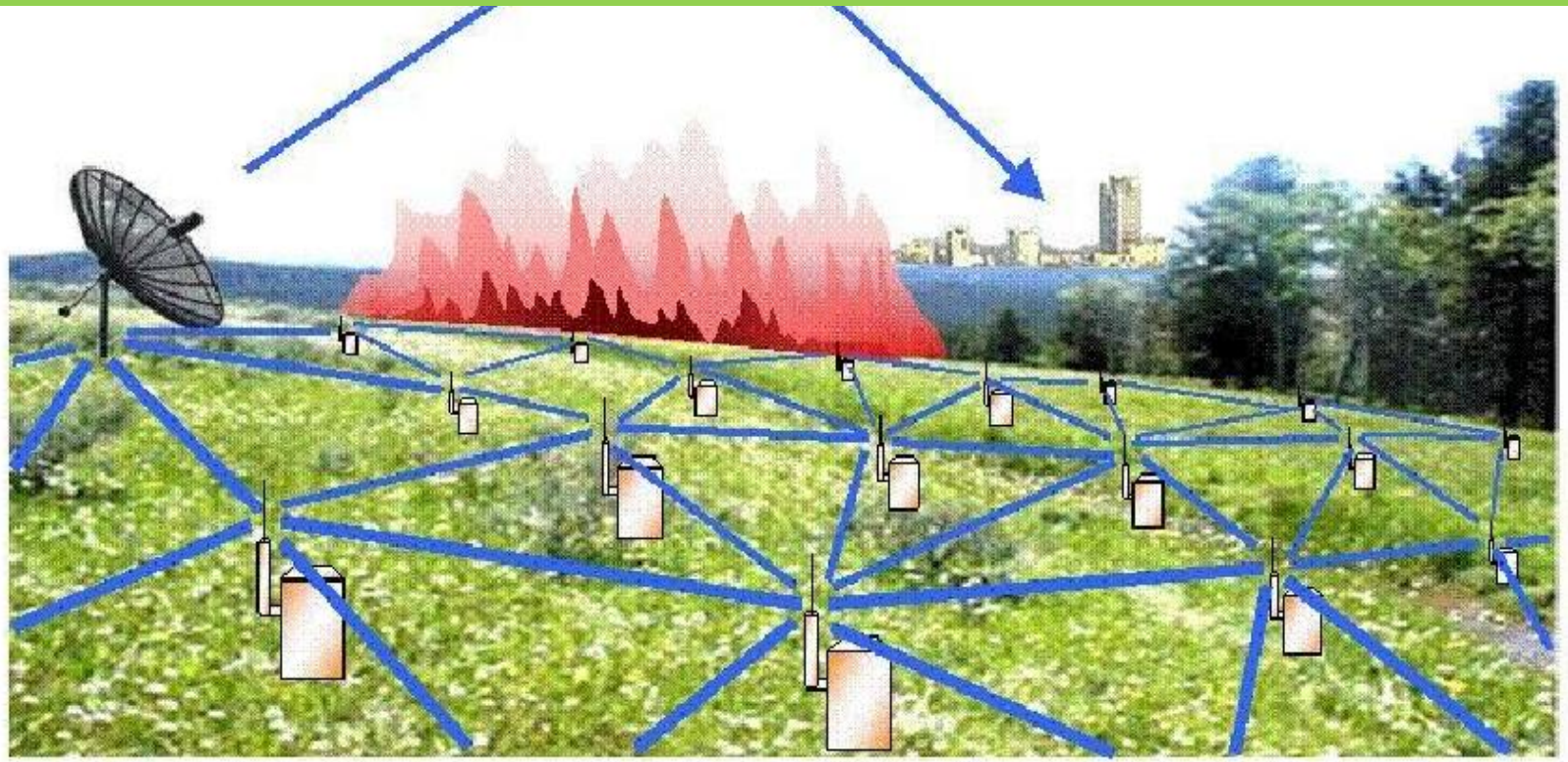
Fontes de Dados Urbanos

Redes de sensores sem fio: instalação de sensores específicos para algumas aplicações

- Volume de tráfego
- Qualidade do ar em diversos pontos da cidade
- Níveis de ruídos

Redes de Sensores sem Fio

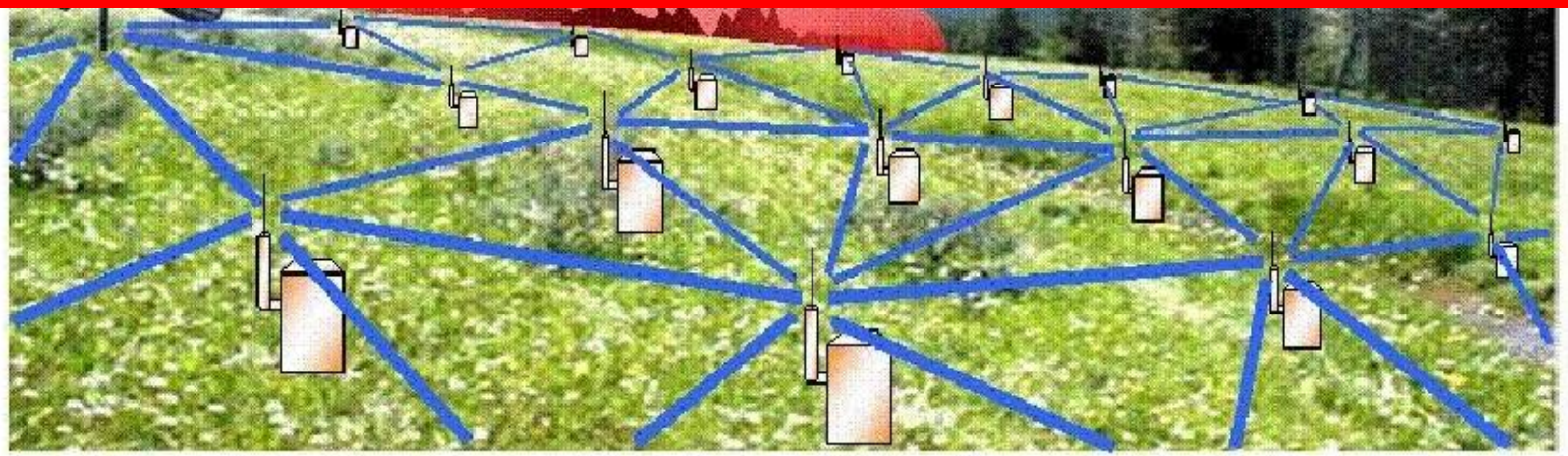
A pesquisa evoluiu bastante para o sensoriamento de áreas de tamanho restrito



Redes de Sensores sem Fio

A pesquisa evoluiu bastante para o sensoriamento de áreas de tamanho restrito

O sensoriamento de áreas grandes (e.g., metrópole) traz grandes desafios



Fontes de Dados Urbanos

Redes de sensoriamento participativo: fornecem dados urbanos que possuem uma escala bastante abrangente

Podem ser mais fáceis de obter

Contam com a colaboração dos usuários na coleta de dados

Humanos como sensores

Vários sensores

Humanos como sensores

Vários sensores

+

Observações

Humanos como sensores

Vários sensores

+

Observações

Sensoriamento participativo

Humanos como sensores

Sensoriamento participativo

Observações são expressadas
Experiências de vida diária são capturadas

Humanos como sensores

Sensoriamento participativo

Observações são compartilhadas
Experiências de vida diária são capturadas

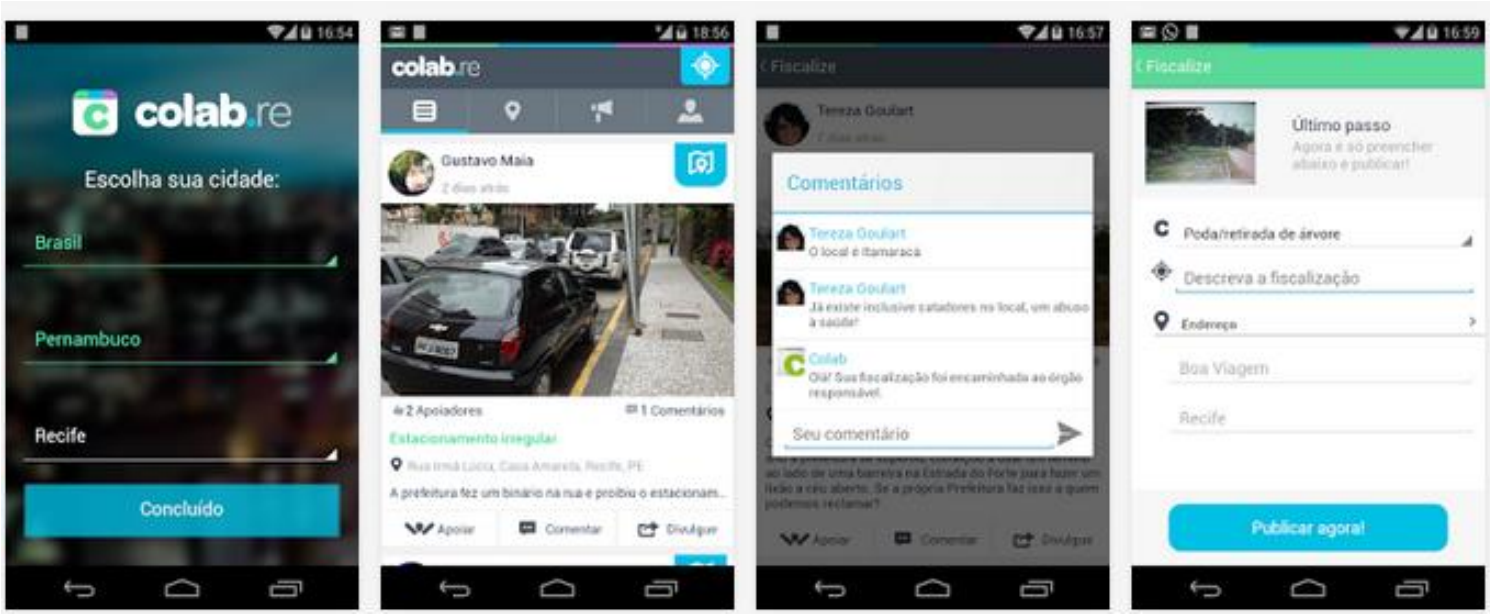
Processo voluntário

Humanos como sensores

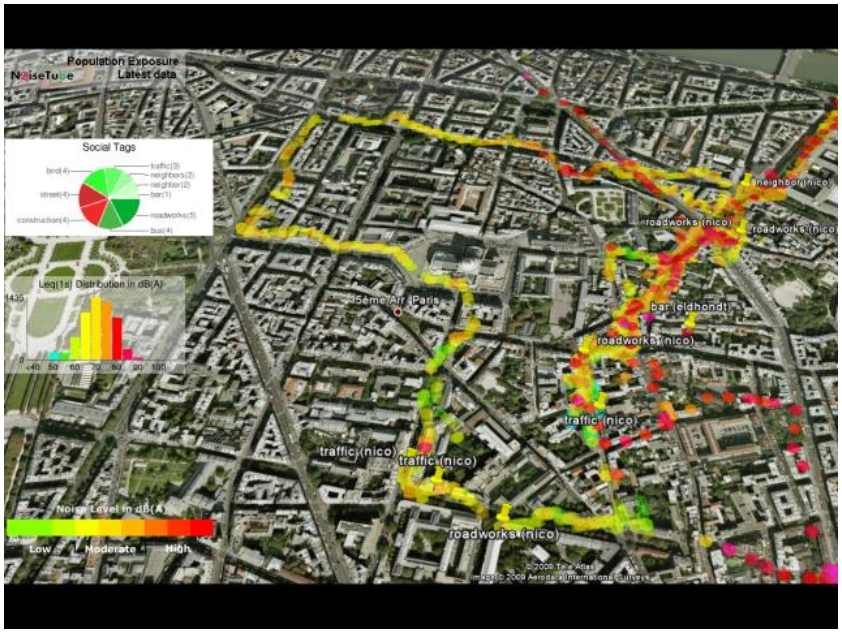
Redes de sensoriamento participativo (RSP)

Combinam características de **computação móvel**, **mídias sociais** e **serviços baseados em localização**

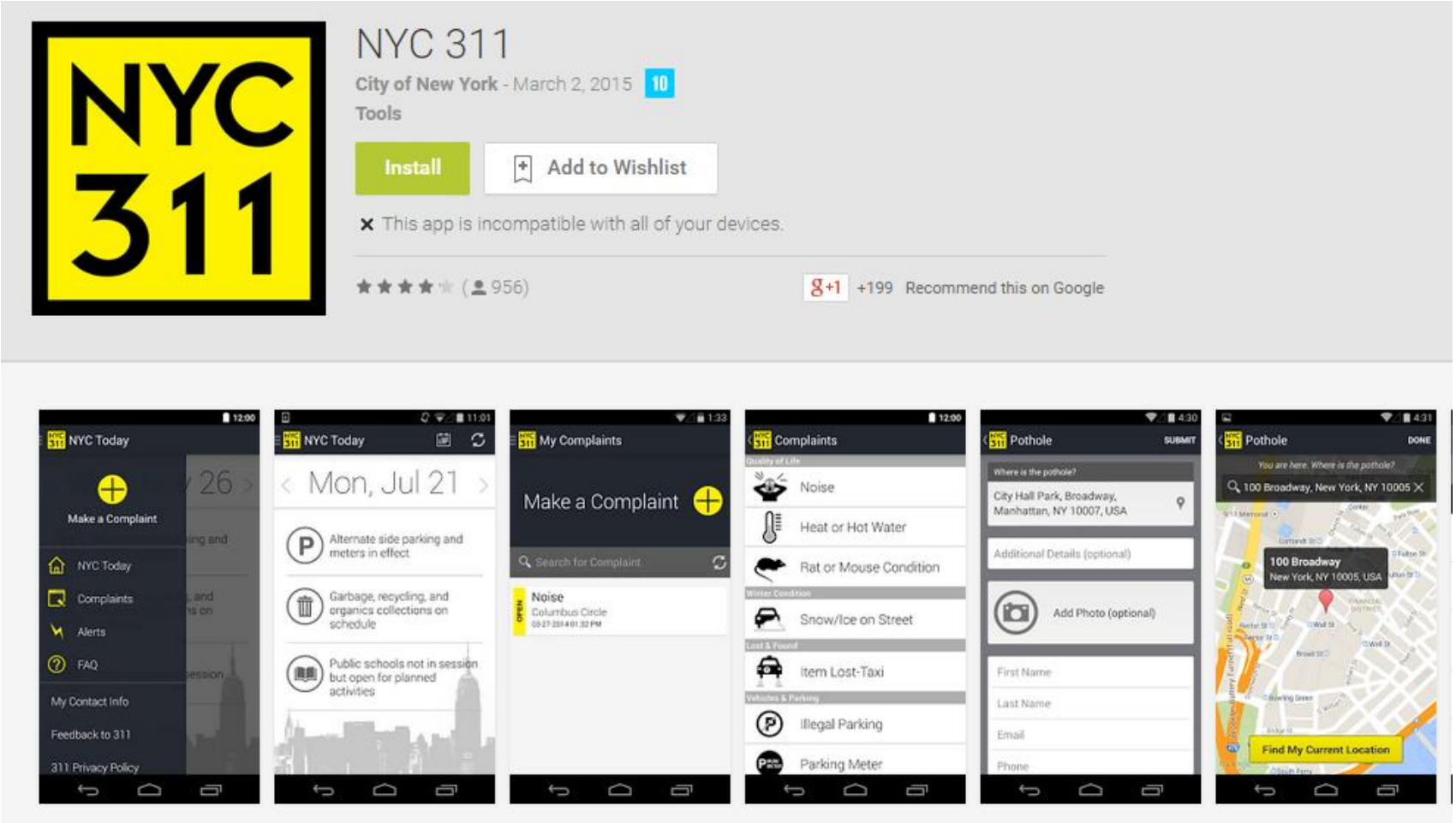
Redes de sensoriamento participativo



Redes de sensoriamento participativo

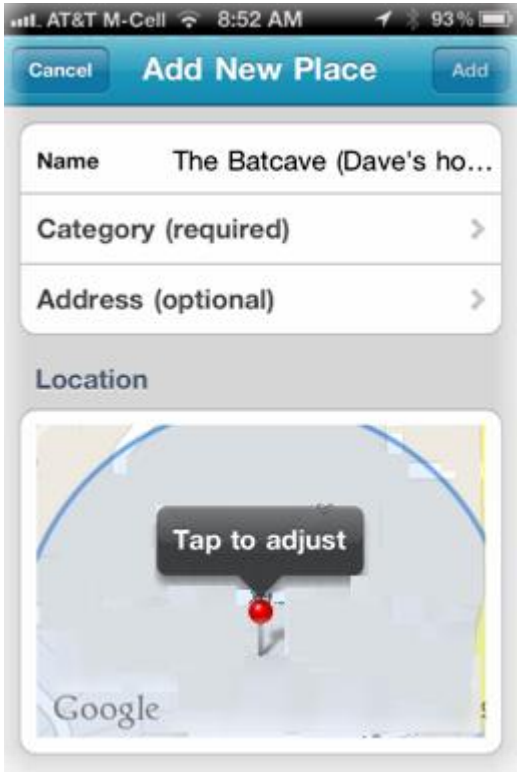


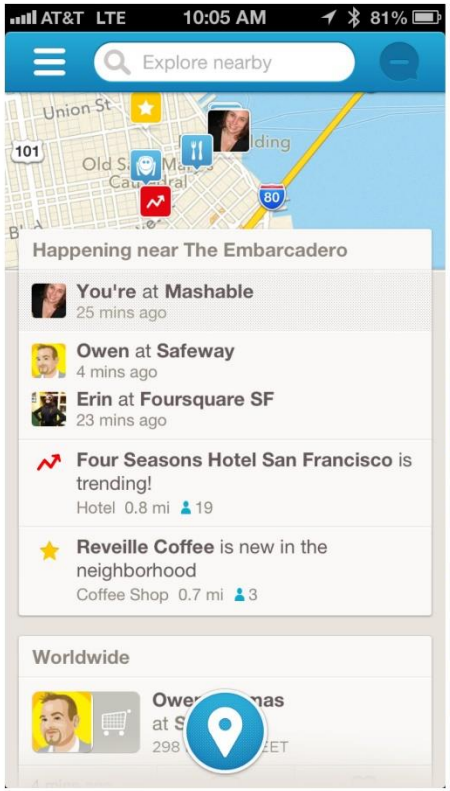
Redes de sensoriamento participativo

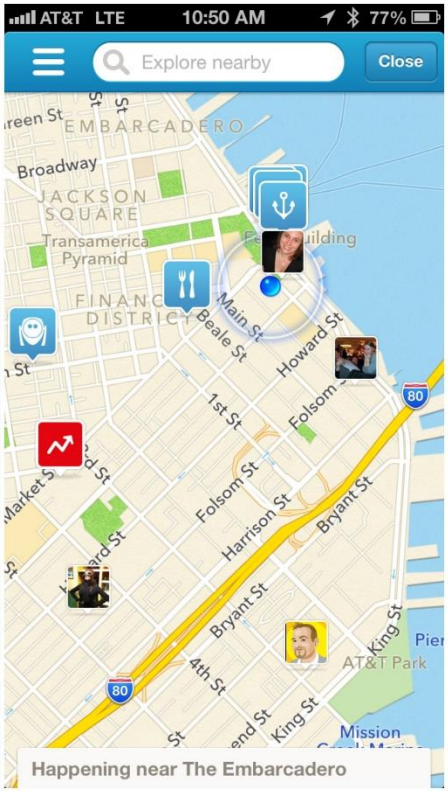


Redes de sensoriamento participativo

Redes de sensoriamento participativo

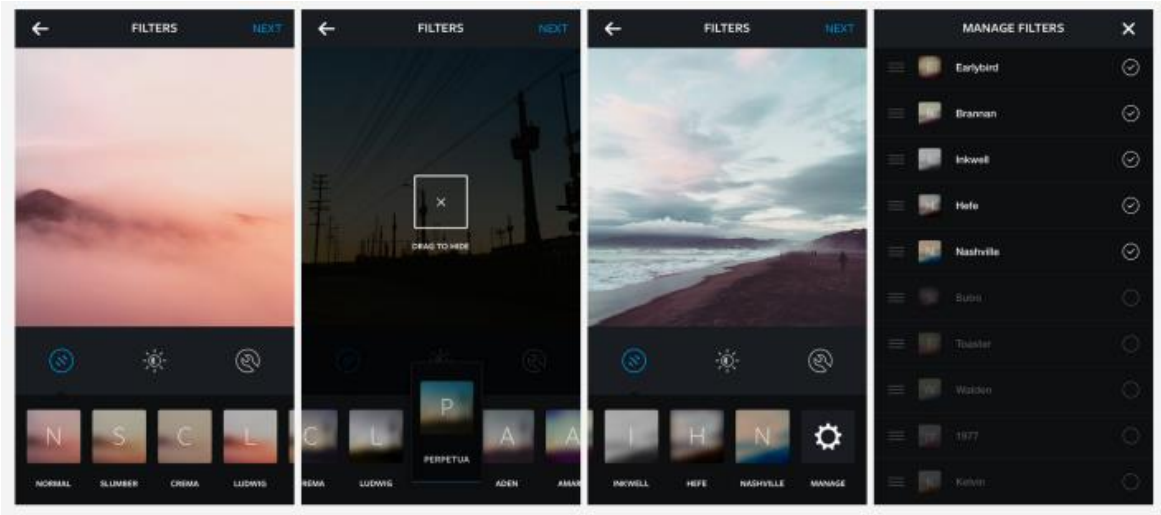




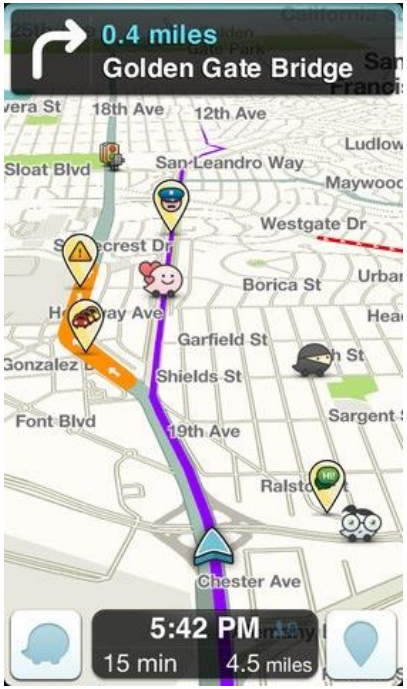


Redes de sensoriamento participativo

Instagram



Redes de sensoriamento participativo



Redes sociais baseadas em localização
(*Location-based social networks – LBSNs*)

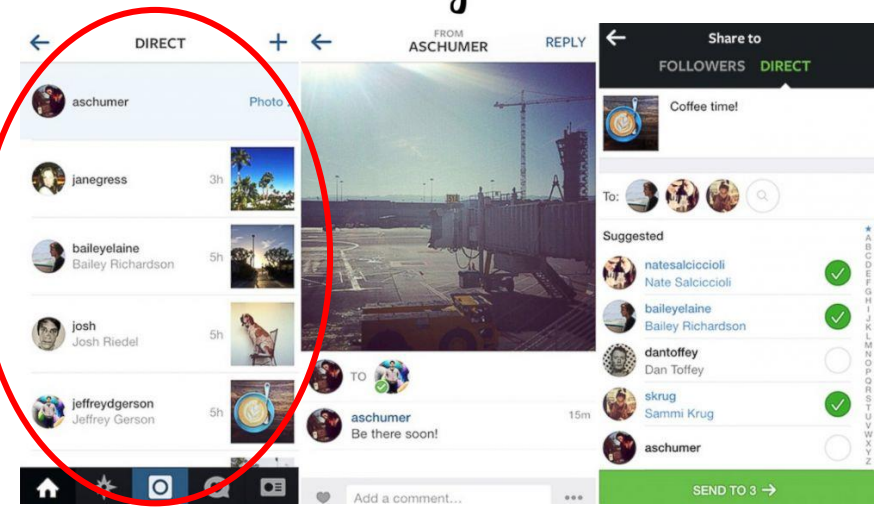
Constroem e refletem as relações sociais da vida real

São os exemplos mais populares

Redes sociais baseadas em localização (*Location-based social networks – LBSNs*)

Constroem e refletem as relações sociais da vida real

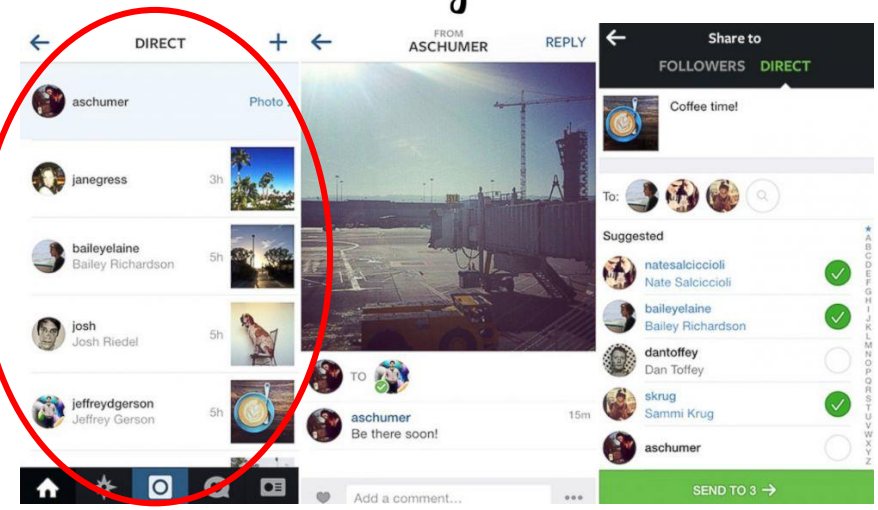
Instagram



Redes sociais baseadas em localização (*Location-based social networks – LBSNs*)

Constroem e refletem as relações sociais da vida real

Instagram



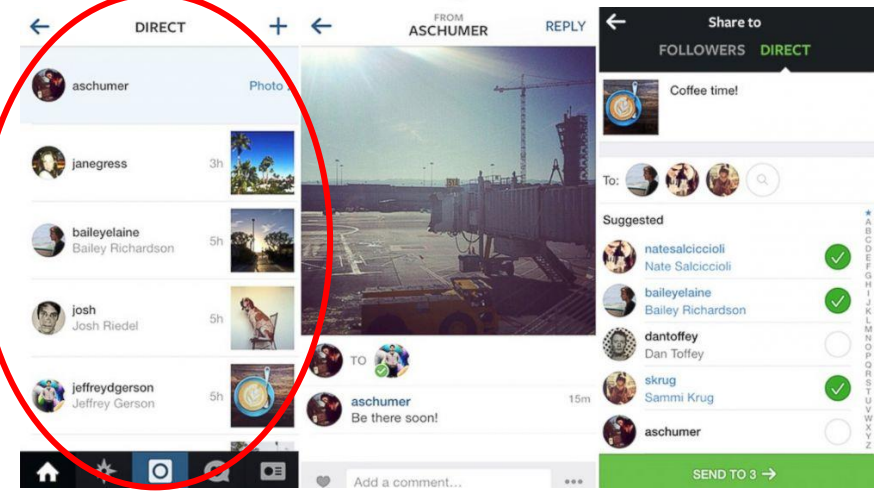
São exemplos de RSPs

Redes de sensoriamento participativo

Redes sociais baseadas em localização (*Location-based social networks – LBSNs*)

Constroem e refletem as relações sociais da vida real

Instagram

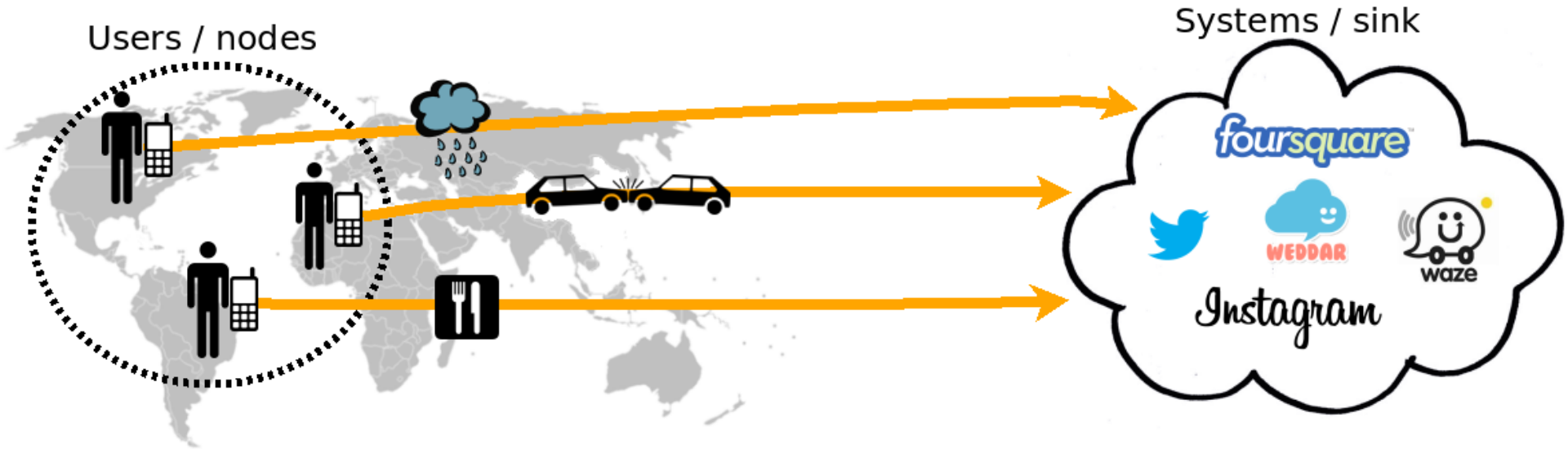


Não é uma LBSN

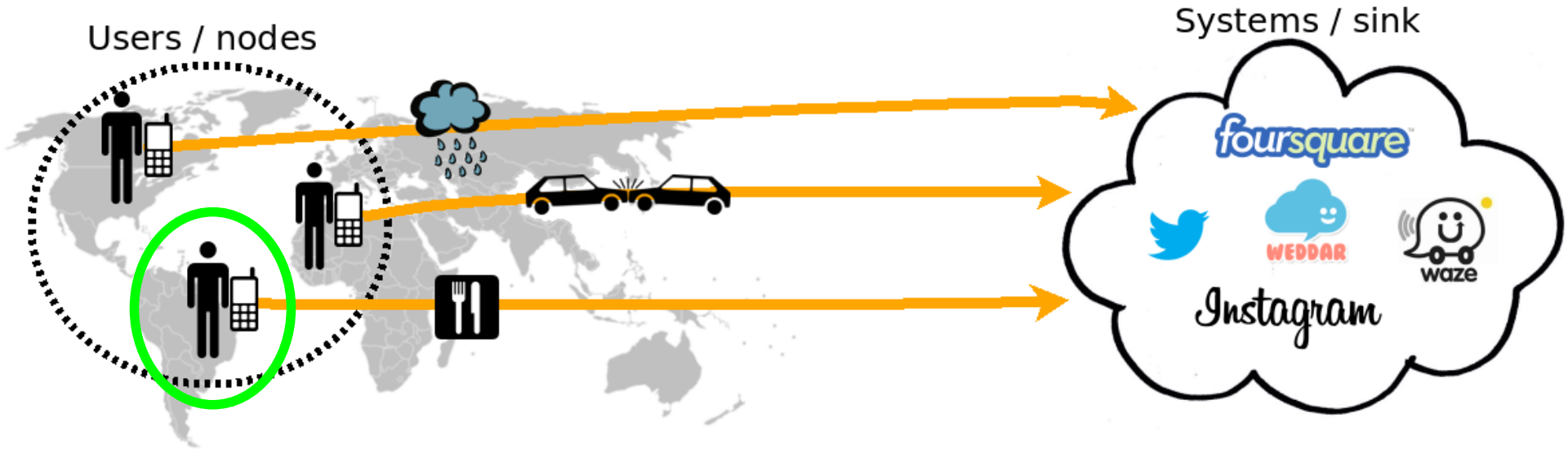
NoiseTube

Redes de sensoriamento participativo

Permitem o compartilhamento de dados pelos usuários sobre áreas urbanas

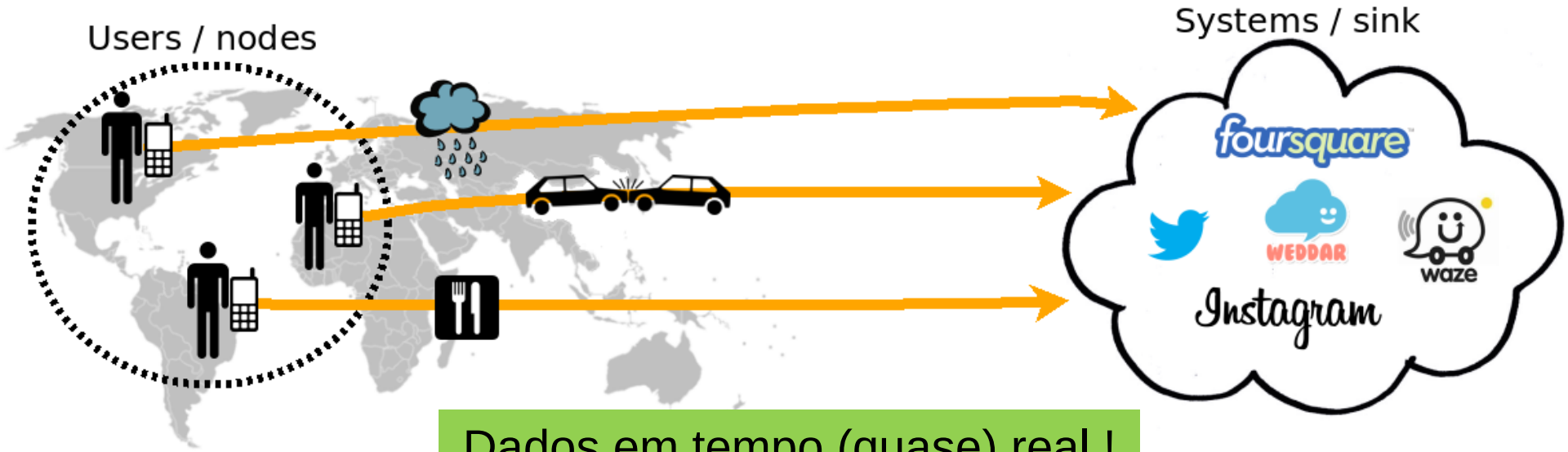


Redes de sensoriamento participativo



Nó:
Usuário + dispositívo móvel

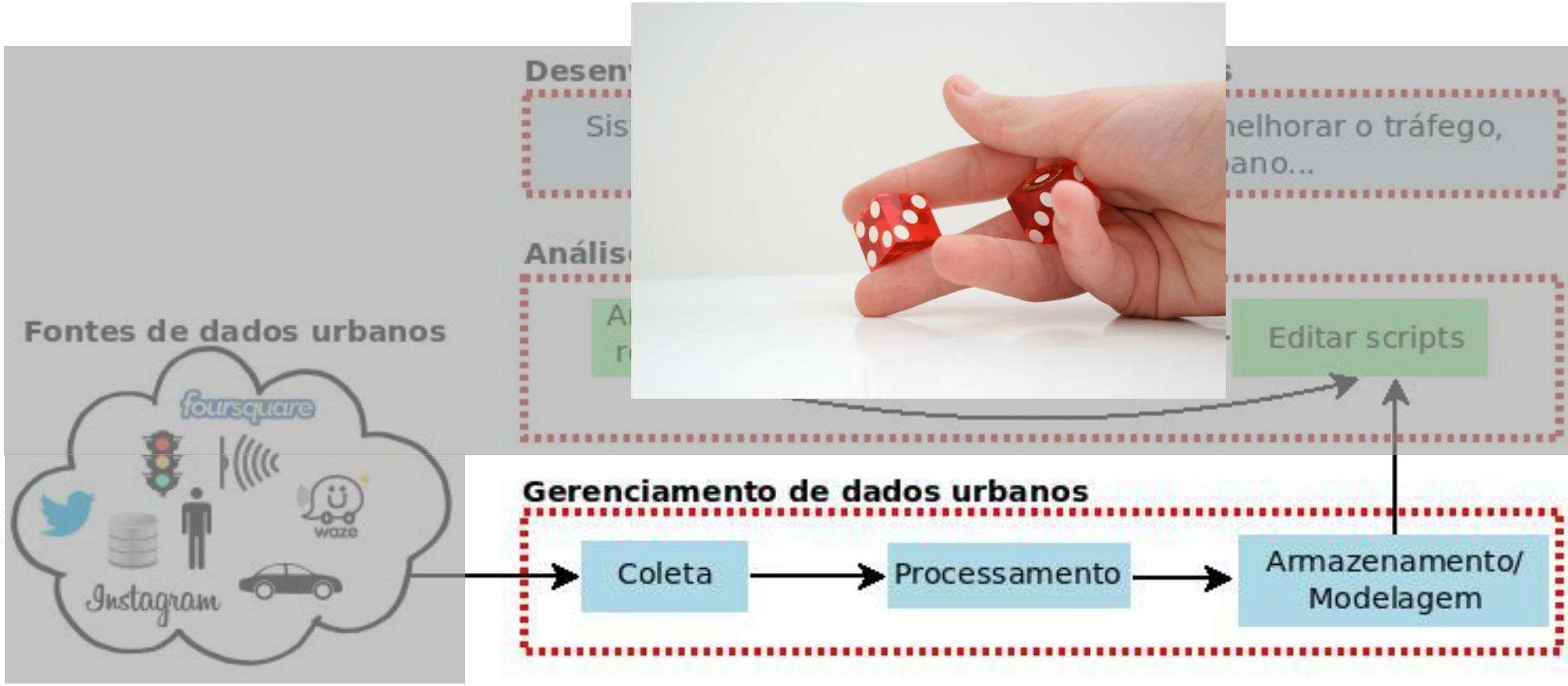
Redes de sensoriamento participativo



Dados em tempo (quase) real !

Arcabouço da Computação Urbana

Arcabouço da Computação Urbana



Coleta

Utilizando APIs

- Permite obter dados facilmente
- **Dois tipos básicos**:
  - *Streaming*
  - Requisições
- **Disponíveis:** Twitter, Foursquare, etc

Utilizando APIs

Exemplo: Usando a API do Twitter

Passo1: Instalar a biblioteca TwitterAPI:

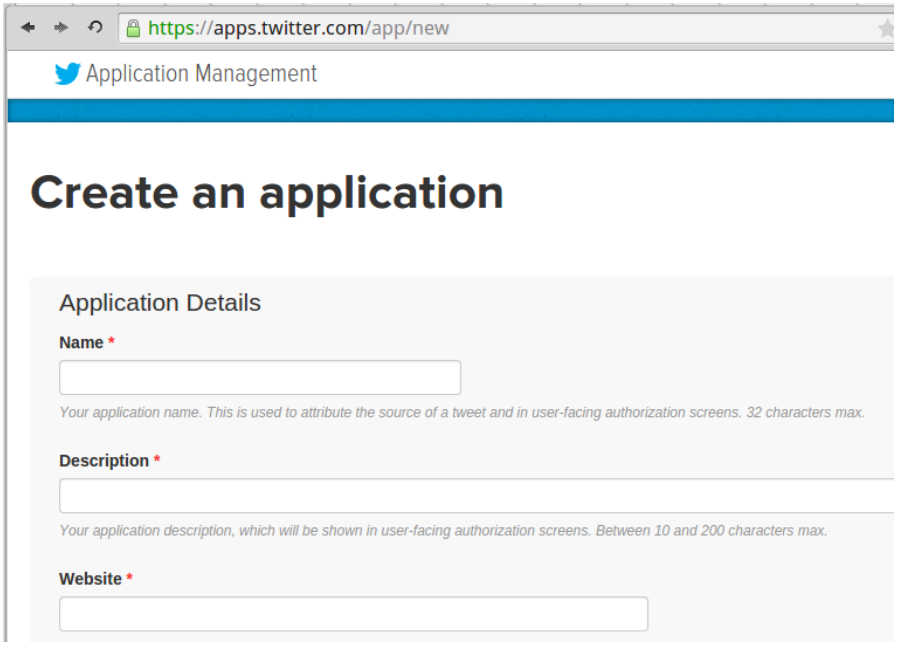
- pip install TwitterAPI
- ou fazer um clone no https://github.com/geduldig/TwitterAPI

Exemplo: Usando a API do Twitter

Passo 2. Criar uma aplicação no Twitter

a. Necessário ter cadastro no Twitter

b. Acessar https://apps.twitter.com e clicar em “Create New App”

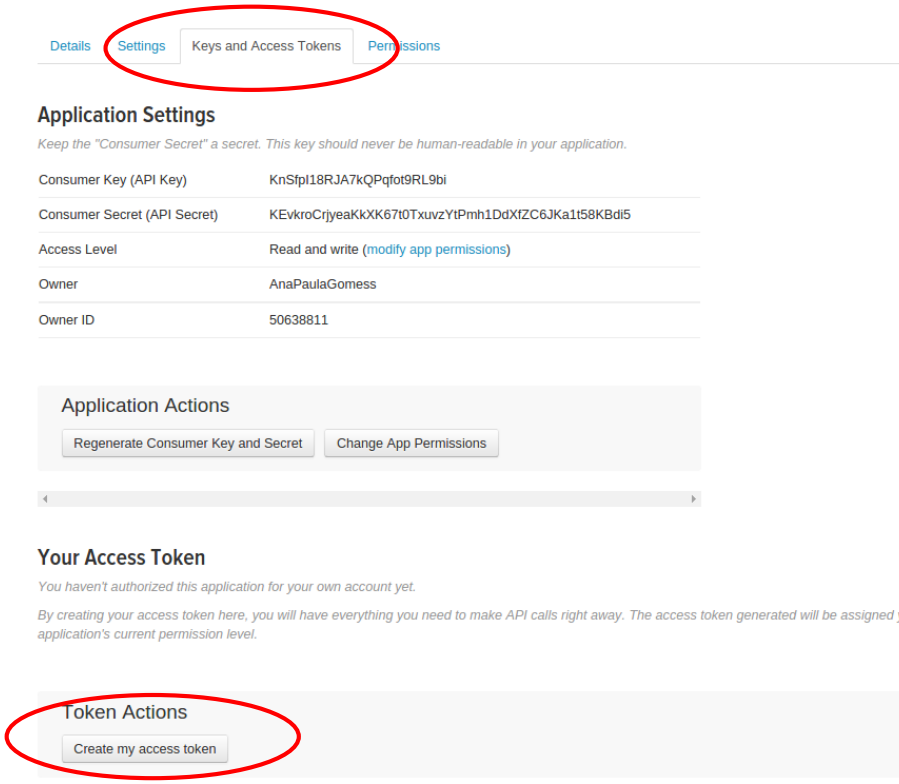


Exemplo: Usando a API do Twitter

Passo 3. Gerar os *tokens* da app

a. Na tela de detalhes da aplicação, clicar na aba "*Keys and Access Token*"

b. Para gerar o *token*, nesta mesma aba clique em "*Create Access Token*"



Exemplo: Usando a API do Twitter

```
from TwitterAPI import TwitterAPI

twitter_api = TwitterAPI(consumer_key='XXXX',
                         consumer_secret='XXXX',
                         access_token_key='XXXX',
                         access_token_secret='XXXX')

for item in twitter_api.request('statuses/filter', {'track':‘csbc'}):
    print item['text']
```

Facilita a comunicação com a API do Twitter

Exemplo: Usando a API do Twitter

```
from TwitterAPI import TwitterAPI

twitter_api = TwitterAPI(consumer_key='XXXX',
                         consumer_secret='XXXX',
                         access_token_key='XXXX',
                         access_token_secret='XXXX')

for item in twitter_api.request('statuses/filter', {'track':‘csbc'}):
    print item['text']
```

Tokens obtidos

Exemplo: Usando a API do Twitter

```
from TwitterAPI import TwitterAPI

twitter_api = TwitterAPI(consumer_key='XXXX',
                         consumer_secret='XXXX',
                         access_token_key='XXXX',
                         access_token_secret='XXXX')

for item in twitter_api.request('statuses/filter', {'track':‘csbc'}):
    print item['text']
```

Requisição dos *tweets* com a termo “csbc”

Exemplo: Usando a API do Twitter

Demonstração – Coleta Streaming

coletaAPIs/streamingTwitter/exemplo1

coletaAPIs/streamingTwitter/exemplo2/coletaTweetsStreaming.py
(Exemplo anterior com pequenas novidades)

Exemplo: Usando a API do Twitter

Exemplo de saída

Slide derivado de Wei Xu

Utilizando APIs

Exemplo: Usando a API do Twitter

```
{"favorited": false, "contributors": null, "truncated": false, "text": "#CFP Workshop on Noisy User-generated Text at ACL - Beiji
ng 31 July 2015. Papers due: 11 May 2015. http://t.co/rcygyEowqH   #NLProc #WNUT15", "possibly_sensitive": false, "in_reply_to_st
atus_id": null, "user": {"follow_request_sent": null, "profile_use_background_image": true, "default_profile_image": false, "id":
237918251, "verified": false, "profile_image_url_https": "https://pbs.twimg.com/profile_images/527088456967544832/DnclpoZO_norma
l.jpeg", "profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333", "followers_count": 226, "profile_sidebar_borde
r_color": "C0DEED", "id_str": "237918251", "profile_background_color": "C0DEED", "listed_count": 13, "profile_background_image_ur
l_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "utc_offset": null, "statuses_count": 120, "description": "I am a
postdoctoral researcher @PennCIS, studying Natural Language Processing and Social Media.", "friends_count": 166, "location": "Phi
ladelphia PA", "profile_link_color": "0084B4", "profile_image_url": "http://pbs.twimg.com/profile_images/527088456967544832/Dnclp
oZO_normal.jpeg", "following": null, "geo_enabled": true, "profile_background_image_url": "http://abs.twimg.com/images/themes/the
me1/bg.png", "name": "Wei Xu", "lang": "en", "profile_background_tile": false, "favourites_count": 88, "screen_name": "cocoweix
u", "notifications": null, "url": "http://www.cis.upenn.edu/~xwe/", "created_at": "Thu Jan 13 23:15:12 +0000 2011", "contributor
s_enabled": false, "time_zone": null, "protected": false, "default_profile": true, "is_translator": false}, "filter_level": "lo
w", "geo": null, "id": 616333141884674048, "favorite_count": 0, "lang": "en", "entities": {"user_mentions": [], "symbols": [], "t
rends": [], "hashtags": [{"indices": [0, 4], "text": "CFP"}, {"indices": [124, 131], "text": "NLProc"}, {"indices": [132, 139],
"text": "WNUT15"}], "urls": [{"url": "http://t.co/rcygyEowqH", "indices": [99, 121], "expanded_url": "http://noisy-text.github.i
o", "display_url": "noisy-text.github.io"}]}, "in_reply_to_user_id_str": null, "retweeted": false, "coordinates": null, "timestam
p_ms": "1435780246598", "source": "<a href=\"http://twitter.com\" rel=\"nofollow\">Twitter Web Client</a>", "in_reply_to_status_i
d_str": null, "in_reply_to_screen_name": null, "id_str": "616333141884674048", "place": null, "retweet_count": 0, "created_at":
"Wed Jul 01 19:50:46 +0000 2015", "in_reply_to_user_id": null}
```


Slide derivado de Wei Xu

Formato JSON

```
{ "Alunos" : [
        { "nome": "João", "notas": [ 8, 9, 7 ]  },
        { "nome": "Maria", "notas": [ 8, 10, 7 ] },
        { "nome": "Pedro", "notas": [ 10, 10, 9 ] }
    ]
}
```

Visualização da hierarquia

Processando um JSON

Utilizando APIs

Exemplo: JSON referente a um *tweet*

pip install simplejson

```python
import simplejson as json

# Arquivo com Tweets
tweets_file = open('tweet.txt', "r")

#le a linha do arquivo
tweet_json = tweets_file.readline()

#imprime a linha lida
print  tweet_json

#remove espacos em branco
strippedJson = tweet_json.strip()

#converte uma string json em um objeto python
tweet = json.loads(strippedJson)

print tweet['id'] # ID do tweet
print tweet['created_at'] # data de postagem
print tweet['text'] # texto do tweet
>>        >>        >>
print tweet['user']['id'] # id do usuario que postou
print tweet['user']['name'] # nome do usuario
print tweet['user']['screen_name'] # nome da conta do usuario
```

coletaAPIs/streamingTwitter/processaExemplo1/

Slide derivado de Wei Xu

Utilizando APIs

Exemplo: JSON referente a um *tweet*

{"favorited": false, "contributors": null, "truncated": false, "text": "#CFP Workshop on Noisy User-generated Text at ACL - Beijing 31 July 2015. Papers due: 11 May 2015. http://t.co/rcygyEowqH #NLProc #WNUT15", "possibly_sensitive": false, "in_reply_to_status_id": null, "user": {"follow_request_sent": null, "profile_use_background_image": true, "default_profile_image": false, "id": 237918251, "verified": false, "profile_image_url_https": "https://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333", "followers_count": 226, "profile_sidebar_border_color": "C0DEED", "id_str": "237918251", "profile_background_color": "C0DEED", "listed_count": 13, "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "utc_offset": null, "statuses_count": 120, "description": "I am a postdoctoral researcher @PennCIS, studying Natural Language Processing and Social Media.", "friends_count": 166, "location": "Philadelphia PA", "profile_link_color": "008484", "profile_image_url": "http://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "following": null, "geo_enabled": true, "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png", "name": "Wei Xu", "lang": "en", "profile_background_tile": false, "favourites_count": 88, "screen_name": "cocoweixu", "notifications": null, "url": "http://www.cis.upenn.edu/~xwe/", "created_at": "Thu Jan 13 23:15:12 +0000 2011", "contributors_enabled": false, "time_zone": null, "protected": false, "default_profile": true, "is_translator": false}, "filter_level": "low", "geo": null, "id": 616333541884674048, "favorite_count": 0, "lang": "en", "entities": {"user_mentions": [], "symbols": [], "trends": [], "hashtags": [{"indices": [0, 4], "text": "CFP"}, {"indices": [124, 131], "text": "NLProc"}, {"indices": [132, 139], "text": "WNUT15"}], "urls": [{"url": "http://t.co/rcygyEowqH", "indices": [99, 121], "expanded_url": "http://noisy-text.github.io", "display_url": "noisy-text.github.io"}]}, "in_reply_to_user_id_str": null, "retweeted": false, "coordinates": null, "timestamp_ms": "1435780246598", "source": "Twitter Web Client", "in_reply_to_status_id_str": null, "in_reply_to_screen_name": null, "id_str": "616333541884674048", "place": null, "retweet_count": 0, "created_at": "Wed Jul 01 19:50:46 +0000 2015", "in_reply_to_user_id": null}

```python
import simplejson as json

# Arquivo com Tweets
tweets_file = open('tweet.txt', "r")

#le a linha do arquivo
tweet_json = tweets_file.readline()

#imprime a linha lida
print  tweet_json

#remove espacos em branco
strippedJson = tweet_json.strip()

#converte uma string json em um objeto python
tweet = json.loads(strippedJson)

print tweet['id'] # ID do tweet
print tweet['created_at'] # data de postagem
print tweet['text'] # texto do tweet
>>        >>        >>
print tweet['user']['id'] # id do usuario que postou
print tweet['user']['name'] # nome do usuario
print tweet['user']['screen_name'] # nome da conta do usuario
```

coletaAPIs/streamingTwitter/processaExemplo1/

Acessando campos do json

Slide derivado de Wei Xu

Utilizando APIs

Exemplo: JSON referente a um *tweet*

{"favorited": false, "contributors": null, "truncated": false, "text": "#CFP Workshop on Noisy User-generated Text at ACL - Beijing 31 July 2015. Papers due: 11 May 2015. http://t.co/rcygyEowqH #NLProc #WNUT15", "possibly_sensitive": false, "in_reply_to_status_id": null, "user": {"follow_request_sent": null, "profile_use_background_image": true, "default_profile_image": false, "id": 237918251, "verified": false, "profile_image_url_https": "https://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333", "followers_count": 225, "profile_sidebar_border_color": "C0DEED", "id_str": "237918251", "profile_background_color": "C0DEED", "listed_count": 13, "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "utc_offset": null, "statuses_count": 120, "description": "I am a postdoctoral researcher @PennCIS, studying Natural Language Processing and Social Media.", "friends_count": 166, "location": "Philadelphia PA", "profile_link_color": "008484", "profile_image_url": "http://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "following": null, "geo_enabled": true, "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png", "name": "Wei Xu", "lang": "en", "profile_background_tile": false, "favourites_count": 88, "screen_name": "cocoweixu", "notifications": null, "url": "http://www.cis.upenn.edu/~xwe/", "created_at": "Thu Jan 13 23:15:12 +0000 2011", "contributors_enabled": false, "time_zone": null, "protected": false, "default_profile": true, "is_translator": false}, "filter_level": "low", "geo": null, "id": 616633141884674048, "favorite_count": 0, "lang": "en", "entities": {"user_mentions": [], "symbols": [], "trends": [], "hashtags": [{"indices": [0, 4], "text": "CFP"}, {"indices": [124, 134], "text": "NLProc"}, {"indices": [132, 139], "text": "WNUT15"}], "urls": [{"url": "http://t.co/rcygyEowqH", "indices": [99, 121], "expanded_url": "http://noisy-text.github.io", "display_url": "noisy-text.github.io"}]}, "in_reply_to_user_id_str": null, "retweeted": false, "coordinates": null, "timestamp_ms": "1435780246598", "source": "Twitter Web Client", "in_reply_to_status_id_str": null, "in_reply_to_screen_name": null, "id_str": "616633141884674048", "place": null, "retweet_count": 0, "created_at": "Wed Jul 01 19:50:46 +0000 2015", "in_reply_to_user_id": null}

```
import simplejson as json

# Arquivo com Tweets
tweets_file = open('tweet.txt', "r")

#le a linha do arquivo
tweet_json = tweets_file.readline()

#imprime a linha lida
print  tweet_json

#remove espacos em branco
strippedJson = tweet_json.strip()

#converte uma string json em um objeto python
tweet = json.loads(strippedJson)

print tweet['id'] # ID do tweet
print tweet['created_at'] # data de postagem
print tweet['text'] # texto do tweet
>>        >>        >>
print tweet['user']['id'] # id do usuario que postou
print tweet['user']['name'] # nome do usuario
print tweet['user']['screen_name'] # nome da conta do usuario
```

```
616633141884674048
Wed Jul 01 19:50:46 +0000 2015
#CFP Workshop on Noisy User-generated Text
at ACL - Beijing 31 July 2015. Papers due:
11 May 2015. http://t.co/rcygyEowqH
#NLProc #WNUT15
237918251
Wei Xu
cocoweixu
```

Slide derivado de Wei Xu

Utilizando APIs

Exemplo: JSON referente a um *tweet*

{"favorited": false, "contributors": null, "truncated": false, "text": "#CFP Workshop on Noisy User-generated Text at ACL - Beijing 31 July 2015. Papers due: 11 May 2015. http://t.co/rcygyEowqH #NLProc #WNUT15", "possibly_sensitive": false, "in_reply_to_status_id": null, "user": {"follow_request_sent": null, "profile_use_background_image": true, "default_profile_image": false, "id": 237918251, "verified": false, "profile_image_url_https": "https://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333", "followers_count": 225, "profile_sidebar_border_color": "C0DEED", "id_str": "237918251", "profile_background_color": "C0DEED", "listed_count": 13, "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "utc_offset": null, "statuses_count": 120, "description": "I am a postdoctoral researcher @PennCIS, studying Natural Language Processing and Social Media.", "friends_count": 166, "location": "Philadelphia PA", "profile_link_color": "008484", "profile_image_url": "http://pbs.twimg.com/profile_images/527088456967544832/Dnclpa2O_normal.jpeg", "following": null, "geo_enabled": true, "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png", "name": "Wei Xu", "lang": "en", "profile_background_tile": false, "favourites_count": 88, "screen_name": "cocoweixu", "notifications": null, "url": "http://www.cis.upenn.edu/~xwe/", "created_at": "Thu Jan 13 23:15:12 +0000 2011", "contributors_enabled": false, "time_zone": null, "protected": false, "default_profile": true, "is_translator": false}, "filter_level": "low", "geo": null, "id": 616633141884674048, "favorite_count": 0, "lang": "en", "entities": {"user_mentions": [], "symbols": [], "trends": [], "hashtags": [{"indices": [0, 4], "text": "CFP"}, {"indices": [124, 134], "text": "NLProc"}, {"indices": [132, 139], "text": "WNUT15"}], "urls": [{"url": "http://t.co/rcygyEowqH", "indices": [99, 121], "expanded_url": "http://noisy-text.github.io", "display_url": "noisy-text.github.io"}]}, "in_reply_to_user_id_str": null, "retweeted": false, "coordinates": null, "timestamp_ms": "1435780246598", "source": "Twitter Web Client", "in_reply_to_status_id_str": null, "in_reply_to_screen_name": null, "id_str": "616633141884674048", "place": null, "retweet_count": 0, "created_at": "Wed Jul 01 19:50:46 +0000 2015", "in_reply_to_user_id": null}

```python
import simplejson as json

# Arquivo com Tweets
tweets_file = open('tweet.txt', "r")

#le a linha do arquivo
tweet_json = tweets_file.readline()

#imprime a linha lida
print  tweet_json

#remove espacos em branco
strippedJson = tweet_json.strip()

#converte uma string json em um objeto python
tweet = json.loads(strippedJson)

print tweet['id'] # ID do tweet
print tweet['created_at'] # data d
print tweet['text'] # texto do twe
>>          >>          >>
print tweet['user']['id'] # id do
print tweet['user']['name'] # nome
print tweet['user']['screen_name'] # nome da conta do usuario
```

```
616633141884674048
Wed Jul 01 19:50:46 +0000 2015
#CFP Workshop on Noisy User-generated Text
at ACL - Beijing 31 July 2015. Papers due:
11 May 2015. http://t.co/rcygyEowqH
#NLProc #WNUT15
237918251
Wei Xu
cocoweixu
```

Processando todos os *tweets* retornados em saidaColetaStream.txt.

coletaAPIs/streamingTwitter/processaExemplo2/

Slide derivado de Wei Xu

Personalização da coleta

```
r = api.request('statuses/filter', {'locations':'-74,40,-73,41'})
for item in r:
        print item
```

Longitude e latitude da área: NYC

Parameter value	Tracks Tweets from...
-122.75,36.8,-121.75,37.8	San Francisco
-74,40,-73,41	New York City
-122.75,36.8,-121.75,37.8,-74,40,-73,41	San Francisco OR New York City

Personalização da coleta

```
r = api.request('statuses/filter', {'locations':'-74,40,-73,41'})
for item in r:
        print item
```

Longitude e latitude da área: NYC

Parameter value	Tracks Tweets from...
-122.75,36.8,-121.75,37.8	San Francisco
-74,40,-73,41	New York City
-122.75,36.8,-121.75,37.8,-74,40,-73,41	San Francisco OR New York City

Combinação

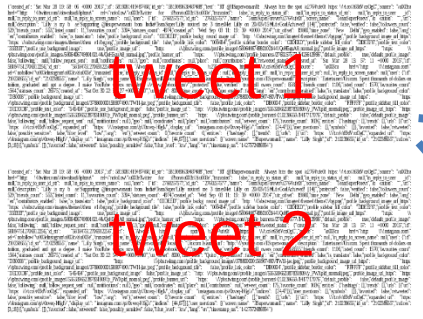
```
track=twitter&locations=-122.75,36.8,-121.75,37.8
```

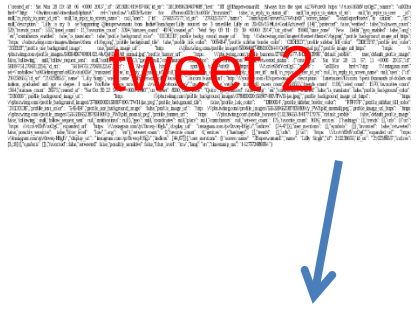
Palavra “twitter” ou tweets de San Fran

Utilizando APIs

Armanezamento dos resultados

Exemplo de saída





Mongo DB

Documentos são similares a objetos JSON

Os valores dos campos podem incluir outros documentos, arrays, arrays de documentos...

```
{
   "_id" : ObjectId("54c955492b7c8eb21818bd09"),
   "address" : {
      "street" : "2 Avenue",
      "zipcode" : "10075",
      "building" : "1480",
      "coord" : [ -73.9557413, 40.7720266 ]
   },
   "borough" : "Manhattan",
   "cuisine" : "Italian",
   "grades" : [
      {
         "date" : ISODate("2014-10-01T00:00:00Z"),
         "grade" : "A",
         "score" : 11
      },
      {
         "date" : ISODate("2014-01-16T00:00:00Z"),
         "grade" : "B",
         "score" : 17
      }
   ],
   "name" : "Vella",
   "restaurant_id" : "41704620"
}
```

Coleções

MongoDB armazena objetos em coleções

Coleções são parecidas com tabelas em bancos relacionais

Uma coleção não requer que os documentos tenham o mesmo esquema

Inserindo dados

Inserir um documento em uma coleção chamada “restaurants”.

```
> use test
switched to db test
> db.restaurants.insert(
...     {
...         "address" : {
...             "street" : "2 Avenue",
...             "zipcode" : "10075",
...             "building" : "1480",
...             "coord" : [ -73.9557413, 40.7720266 ]
...         },
...         "borough" : "Manhattan",
...         "cuisine" : "Italian",
...         "grades" : [
...             {
...                 "date" : ISODate("2014-10-01T00:00:00Z"),
...                 "grade" : "A",
...                 "score" : 11
...             },
...             {
...                 "date" : ISODate("2014-01-16T00:00:00Z"),
...                 "grade" : "B",
...                 "score" : 17
...             }
...         ],
...         "name" : "Vella",
...         "restaurant_id" : "41704620"
...     }
... )
WriteResult({ "nInserted" : 1 })
>
```

Encontrar dados

O método find() faz uma *query* para recuperar dados de uma coleção.

Encontrar dados

Para retornar todos os documentos (find() sem critérios).

db.restaurants.find()

Banco Coleção

Encontrar dados

```
> db.restaurants.find()
{ "_id" : ObjectId("571057054db7bc3a08aae5ba"), "address" : { "building" : "1007", "coord" : [ -73.856077, 40.848447 ], "street" : "Morris
Ave", "zipcode" : "10462" }, "borough" : "Bronx", "cuisine" : "Bakery", "grades" : [ { "date" : ISODate("2014-03-03T00:00:00Z"), "grade" :
"score" : 2 }, { "date" : ISODate("2013-09-11T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2013-01-24T00:00:00Z"), "grad
A", "score" : 10 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2011-03-10T00:00:00Z"),
" : "B", "score" : 14 } ], "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("571057054db7bc3a08aae5bb"), "address" : { "building" : "469", "coord" : [ -73.961704, 40.662942 ], "street" : "Flatbus
ue", "zipcode" : "11225" }, "borough" : "Brooklyn", "cuisine" : "Hamburgers", "grades" : [ { "date" : ISODate("2014-12-30T00:00:00Z"), "gr
 "A", "score" : 8 }, { "date" : ISODate("2014-07-01T00:00:00Z"), "grade" : "B", "score" : 23 }, { "date" : ISODate("2013-04-30T00:00:00Z"
de" : "A", "score" : 12 }, { "date" : ISODate("2012-05-08T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Wendy'S", "restaurant_i
0112340" }
{ "_id" : ObjectId("571057054db7bc3a08aae5bc"), "address" : { "building" : "351", "coord" : [ -73.98513559999999, 40.7676919 ], "street"
   57 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "Irish", "grades" : [ { "date" : ISODate("2014-09-06T00:00:00Z"
ade" : "A", "score" : 2 }, { "date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-07-31T00:0
, "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-12-29T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Dj Reynolds Pub A
aurant", "restaurant_id" : "30191841" }
{ "_id" : ObjectId("571057054db7bc3a08aae5bd"), "address" : { "building" : "2780", "coord" : [ -73.98241999999999, 40.579505 ], "street"
lwell Avenue", "zipcode" : "11224" }, "borough" : "Brooklyn", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-06-10T00:00:0
"grade" : "A", "score" : 5 }, { "date" : ISODate("2013-06-05T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2012-04-13T00
"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-10-12T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Riviera Caterer"
taurant_id" : "40356018" }
{ "_id" : ObjectId("571057054db7bc3a08aae5be"), "address" : { "building" : "97-22", "coord" : [ -73.8601152, 40.7311739 ], "street" : "63
 "zipcode" : "11374" }, "borough" : "Queens", "cuisine" : "Jewish/Kosher", "grades" : [ { "date" : ISODate("2014-11-24T00:00:00Z"), "grade
", "score" : 20 }, { "date" : ISODate("2013-01-17T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-08-02T00:00:00Z"),
" : "A", "score" : 13 }, { "date" : ISODate("2011-12-15T00:00:00Z"), "grade" : "B", "score" : 25 } ], "name" : "Tov Kosher Kitchen", "rest
_id" : "40356068" }
{ "_id" : ObjectId("571057054db7bc3a08aae5bf"), "address" : { "building" : "8825", "coord" : [ -73.8803827, 40.7643124 ], "street" : "Asto
ulevard", "zipcode" : "11369" }, "borough" : "Queens", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-11-15T00:00:00Z"),
```

Encontrar dados

Documentos em que o campo “borough” é "Manhattan“:

```
db.restaurants.find( { "borough": "Manhattan" } )
```

Encontrar dados

```
> db.restaurants.find( { "borough": "Manhattan" } )
{ "_id" : ObjectId("571057054db7bc3a08aae5bc"), "address" : { "building" : "351", "coord" : [ -73.98513559999999, 40.7676919 ], "str
   57 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "Irish", "grades" : [ { "date" : ISODate("2014-09-06T00:0
ade" : "A", "score" : 2 }, { "date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-07-31
, "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-12-29T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Dj Reynolds
aurant", "restaurant_id" : "30191841" }
{ "_id" : ObjectId("571057054db7bc3a08aae5c7"), "address" : { "building" : "1", "coord" : [ -73.96926909999999, 40.7685235 ], "stree
 66 Street", "zipcode" : "10065" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-05-07T00
grade" : "A", "score" : 3 }, { "date" : ISODate("2013-05-03T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-04-3
), "grade" : "A", "score" : 6 }, { "date" : ISODate("2011-12-27T00:00:00Z"), "grade" : "A", "score" : 0 } ], "name" : "1 East 66Th S
n", "restaurant_id" : "40359480" }
{ "_id" : ObjectId("571057054db7bc3a08aae5cc"), "address" : { "building" : "522", "coord" : [ -73.95171, 40.767461 ], "street" : "Ea
et", "zipcode" : "10021" }, "borough" : "Manhattan", "cuisine" : "American ", "grades" : [ { "date" : ISODate("2014-09-02T00:00:00Z"
 "A", "score" : 12 }, { "date" : ISODate("2013-12-19T00:00:00Z"), "grade" : "B", "score" : 16 }, { "date" : ISODate("2013-05-28T00:0
ade" : "A", "score" : 9 }, { "date" : ISODate("2012-12-07T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-03-29
, "grade" : "A", "score" : 11 } ], "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "_id" : ObjectId("571057054db7bc3a08aae5cf"), "address" : { "building" : "759", "coord" : [ -73.9925306, 40.7309346 ], "street" :
"zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Delicatessen", "grades" : [ { "date" : ISODate("2014-01-21T00:00:00Z"),
A", "score" : 12 }, { "date" : ISODate("2013-01-04T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-06-07T00:00:
e" : "A", "score" : 6 }, { "date" : ISODate("2012-01-17T00:00:00Z"), "grade" : "A", "score" : 8 } ], "name" : "Bully'S Deli", "resta
"40361708" }
```

Encontrar dados

Exemplo em python:

mongoExemplos/findData.py

Dados geográficos

GeoJson

Formato para codificar estruturas de dados geográficas

```
"coordinates":
{
  "coordinates":
  [
    -75.14310264,
    40.05701649
  ],
  "type":"Point"
}
```

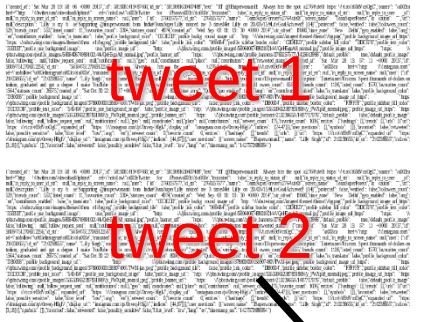
Twitter

Dados geográficos

```
db.places.find(
   {
     location:
       { $near :
          {
            $geometry: { type: "Point",  coordinates: [ -73.9667, 40.78 ] },
            $minDistance: 1000,
            $maxDistance: 5000
          }
       }
   }
)
```

Utilizando os dados

Exemplo de saída



id do usuário

tempo

latitude

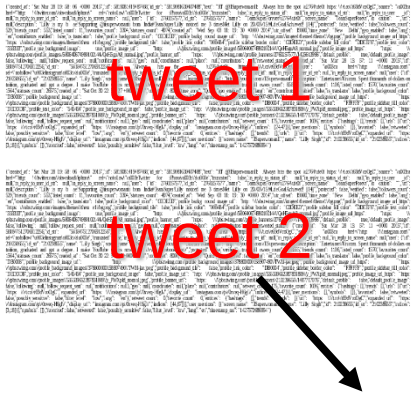
```
296298186        2014-05-01T18:15:09Z    40.87700034
29.23510051      I'm at Başkent Cafe»    4sq.com/1mhXj6u
```

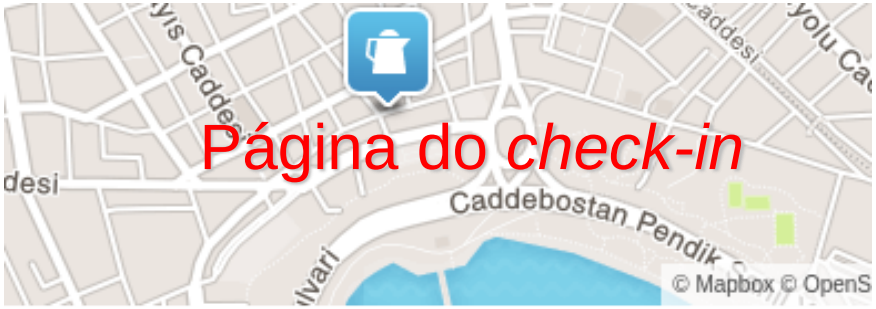
longitude

texto informado

Utilizando os dados

Exemplo de saída





Ümit Yıldız checked in at **Başkent Cafe**

Pendik, Istanbul | May 1, 2014 via Foursquare for Android

id do usuário — tempo — latitude

```
296298186        2014-05-01T18:15:09Z    40.87700034
29.23510051      I'm at Başkent Cafe»    4sq.com/1mhXj6u
```

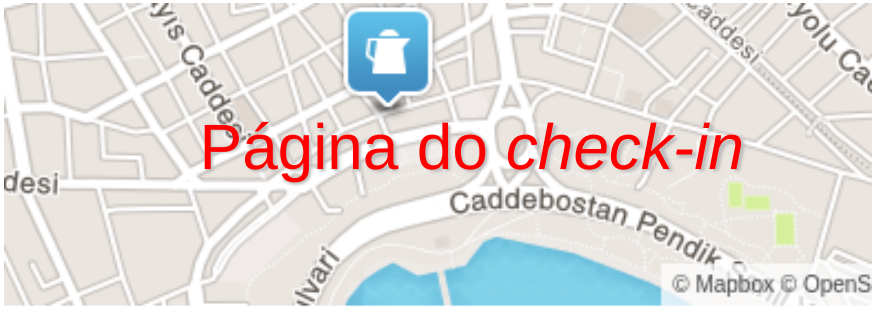
longitude — texto informado

Utilizando os dados

Exemplo de saída

tweet 1

tweet 2



Ümit Yıldız checked in at **Başkent Cafe**

Pendik, Istanbul | May 1, 2014 via Foursquare for Android

Local (pertence à categoria café)

id do usuário

tempo

latitude

```
296298186        2014-05-01T18:15:09Z    40.87700034
29.23510051      I'm at Başkent Cafe»    4sq.com/1mhXj6u
```

longitude

texto informado

Utilizando os dados

Exemplo de saída

tweet 1

tweet 2

Página do *check-in*

Web *crawler*

id do usuário

tempo

latitude

```
296298186        2014-05-01T18:15:09Z    40.87700034
29.23510051      I'm at Başkent Cafe     4sq.com/1mhXj6u
```

longitude

texto informado

Local (pertence à categoria café)

Web *crawler*

Programas que analisam páginas Web em busca do conteúdo desejado

- Alternativa para quando não existe API
- Existem várias estratégias

Coleta com Web *Crawler*

```
# -*- encoding: utf-8 -*-

import urllib

#url obtida através do tweet
url = "http://www.utfpr.edu.br"

pagina = urllib.urlopen(url).read()

print(pagina)

fsaida = open('paginaColetada.html','w')
fsaida.write(str(pagina))
```

Demonstração
coletaCrawler/ColetaPaginaSimples.py

Coleta com Web *Crawler*

```python
# -*- encoding: utf-8 -*-

import urllib

#url obtida através do tweet
url = "http://www.utfpr.edu.br"

pagina = urllib.urlopen(url).read()

print(pagina)

fsaida = open('paginaColetada.html','w')
fsaida.write(str(pagina))
```

Saída do código

```
rt" class="primeiro"><a href="http://www.utfpr.edu.br">Reitoria</a></li>
ap"><a href="http://www.utfpr.edu.br/apucarana">Apucarana</a></li>
cm"><a href="http://www.utfpr.edu.br/campomourao">Campo Mourão</a></li>
cp"><a href="http://www.utfpr.edu.br/cornelioprocopio">Cornélio Procópio</a></
ct"><a href="http://www.utfpr.edu.br/curitiba">Curitiba</a></li>
dv"><a href="http://www.utfpr.edu.br/doisvizinhos">Dois Vizinhos</a></li>
fb"><a href="http://www.utfpr.edu.br/franciscobeltrao">Francisco Beltrão</a></
gp" class="primeiro"><a href="http://www.utfpr.edu.br/guarapuava">Guarapuava</
ld"><a href="http://www.utfpr.edu.br/londrina">Londrina</a></li>
md"><a href="http://www.utfpr.edu.br/medianeira">Medianeira</a></li>
pb"><a href="http://www.utfpr.edu.br/patobranco">Pato Branco</a></li>
pg"><a href="http://www.utfpr.edu.br/pontagrossa">Ponta Grossa</a></li>
sh"><a href="http://www.utfpr.edu.br/santahelena">Santa Helena</a></li>
td"><a href="http://www.utfpr.edu.br/toledo">Toledo</a></li>
```

Exemplo com a biblioteca BeautifulSoup

BeautifulSoup para copiar dados de páginas Web

Para instalá-la, execute o comando:

```
pip install beautifulsoup4
sudo apt-get install python-bs4
```

Coleta com Web *Crawler*

```
from bs4 import BeautifulSoup

pagina = '<html><body><p>Bom dia, Fafich!</p></body></html>'
soup = BeautifulSoup(pagina)
print soup.prettify()
```

Importa a biblioteca

Lê a página

```
<html>
 <body>
  <p>
   Bom dia, Fafich!
  </p>
 </body>
</html>
```

coletaCrawler/beautifulSoup/example1.py

Coleta com Web *Crawler*

```
from bs4 import BeautifulSoup

pagina = '<html><body><p>Bom dia, Fafich!</p></body></html>'
soup = BeautifulSoup(pagina)

allP = soup.find_all('p')
print allP

print allP[0].text
```

Reconhece a *tag* do elemento desejado

Extrai o texto

```
[<p>Bom dia, Fafich!</p>]
Bom dia, Fafich!
```

coletaCrawler/beautifulSoup/example2.py

Coleta com Web *Crawler*

```
from bs4 import BeautifulSoup

pagina = "<table><tr><td>one</td><td>two</td></tr></table>"
soup = BeautifulSoup(pagina)

allTR = soup.find_all('tr')
print allTR
```

[<tr><td>one</td><td>two</td></tr>]

```
allTD = soup.find_all('td')
print allTD
```

[<td>one</td>, <td>two</td>]

```
for t in soup.find_all('td'):
  print t
```

<td>one</td>
<td>two</td>

```
for t in soup.find_all('td'):
  print t.text
```

one
two

coletaCrawler/beautifulSoup/example3.py

Coleta com Web *Crawler*

http://www.submarino.com.br/produto/109588231/grill-george-foreman-familia-estufa-azul

Pegar os dados indicados nesta página

Coleta com Web *Crawler*

pip install requests

```
url = "http://www.submarino.com.br/produto/109588231/grill-george-foreman-familia-e

##retorna o conteudo da pagina
req = requests.get(url)

##transforma o conteudo da pagina em um objeto BeautifulSoup
soup = BeautifulSoup(req.content,'html.parser')

nomeBruto = soup.find("div",{"class":"mp-title"})
nomeBrutoRefinado = nomeBruto.find("span",{"itemprop":"name"})
print(nomeBrutoRefinado.text)

precoBruto = soup.find("span",{"itemprop":"price"})
print(precoBruto.text)
```

"price/salesPrice"

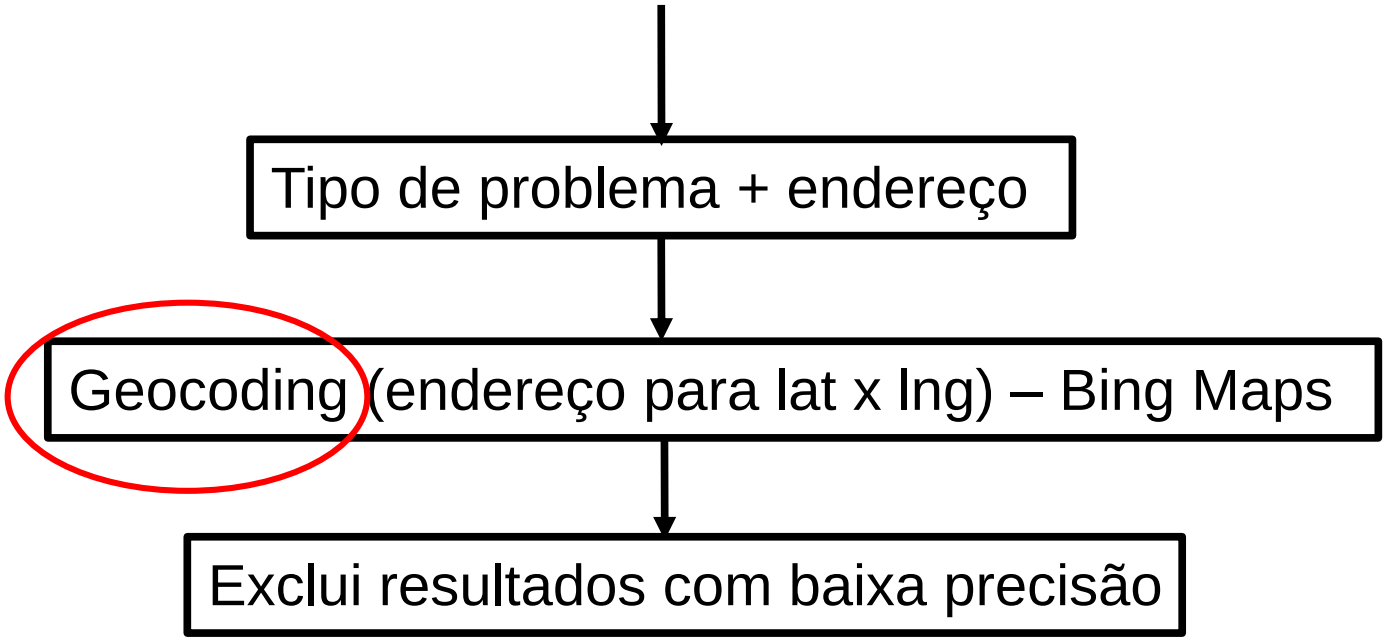
coletaCrawler/beautifulSoup/submarinoCrawler/

Devemos respeitar o termo de uso do site

Coletor de condições do trânsito – Bing Maps

Recursos complementares

Total de *tweets* com alertas do Waze: 212.814



Alertas resultantes (lat x lng): 162.212

Recursos complementares

Geocoding

Google Geocoding
Bing Maps
ArcGIS
Nominatim
Photon
Geopy

Recursos complementares

Google Geocoding

Users of the free API:

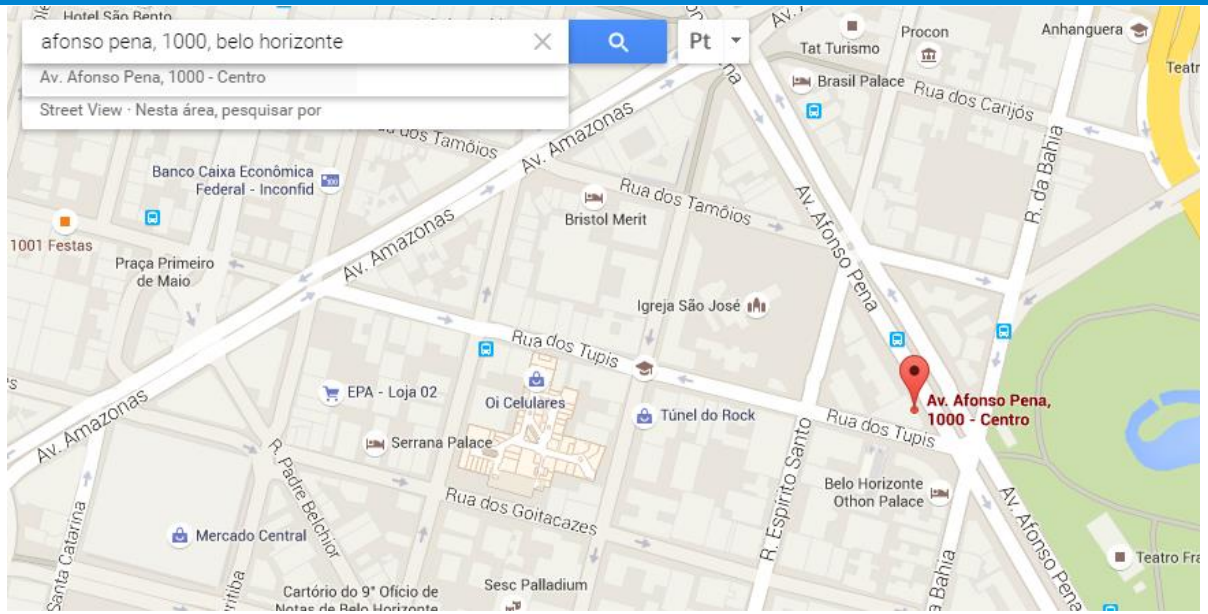
- 2500 requests per 24 hour period.
- 5 requests per second.

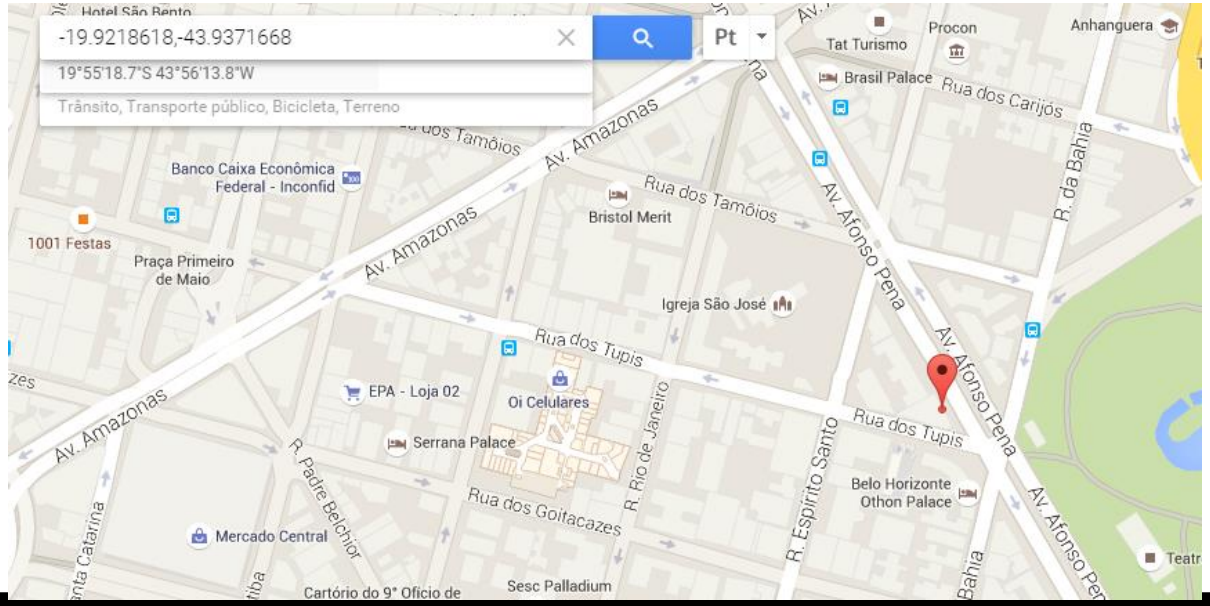
http://maps.googleapis.com/maps/api/geocode/json?address=Afonso Pena 1000, Belo Horizonte

Recursos complementares


```
{
   "results" : [
      {
         "address_components" : [
            {
               "long_name" : "1000",
               "short_name" : "1000",
               "types" : [ "street_number" ]
            },
            {
               "long_name" : "Avenida Afonso Pena",
               "short_name" : "Av. Afonso Pena",
               "types" : [ "route" ]
            },
            {
               "long_name" : "Centro",
               "short_name" : "Centro",
               "types" : [ "neighborhood", "political" ]
            },
            {
               "long_name" : "Belo Horizonte",
               "short_name" : "Belo Horizonte",
               "types" : [ "locality", "political" ]
            },
            {
               "long_name" : "Belo Horizonte",
               "short_name" : "Belo Horizonte",
               "types" : [ "administrative_area_level_2",
            },
            {
               "long_name" : "Minas Gerais",
               "short_name" : "MG",
               "types" : [ "administrative_area_level_1", "political"
            },
            {
               "long_name" : "Brazil",
               "short_name" : "BR",
               "types" : [ "country", "political" ]
            },
            {
               "long_name" : "30130-009",
               "short_name" : "30130-009",
               "types" : [ "postal_code" ]
            }
         ],
         "formatted_address" : "Avenida Afonso Pena, 1000 - Centro, B
         "geometry" : {
            "location" : {
               "lat" : -19.9218618,
               "lng" : -43.9371668
            },
```

Recursos complementares





Recursos complementares

Reverse geocoding

Latitude x Longitude para endereço

```
https://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&key=API_KEY
```


```
            },
            {
               "long_name" : "New York",
               "short_name" : "NY",
               "types" : [ "administrative_area_level_1", "political" ]
            },
            {
               "long_name" : "United States",
               "short_name" : "US",
               "types" : [ "country", "political" ]
            },
            {
               "long_name" : "11211",
               "short_name" : "11211",
               "types" : [ "postal_code" ]
            }
         ],
         "formatted_address" : "277 Bedford Avenue, Brooklyn, NY 11211, USA",
```

Utilizando aplicações

- Plataformas existentes ou novas
- Plataformas existentes: Facebook, Instagram
- Plataforma própria: Noisetube, #311, etc

Reformatação e Limpeza

Reformatação e Limpeza

Estas tarefas tendem a ser demoradas e tediosas

CrowdFlower
2015 Data Scientist Report

Reformatação e Limpeza

Os dados brutos de RSPs podem não estar em um formato conveniente para executar uma análise particular

É possível encontrar erros semânticos, entradas ausentes ou formatação inconsistente

Caso dos Alvarás

Reformatação e Limpeza

Reformatação e limpeza geralmente ocorre utilizando *scripts*

Maior praticidade em linguagens como Python do que em Java ou C++

Reformatação e Limpeza

Reformatação e limpeza geralmente ocorre utilizando *scripts*

Maior praticidade em linguagens como Python do que em Java ou C++

A integração de dados é um desafio que será discutido posteriormente

Reformatação e Limpeza

Pode proporcionar ideias sobre quais suposições são seguras de serem feitas sobre os dados

Peculiaridades existentes no processo de coleta

Quais os modelos e análises são apropriados

Modelagem de Dados

Modelagem de Dados

Os dados gerados em espaços urbanos são geralmente associados com uma propriedade **espacial** e/ou **temporais**

Modelagem de Dados

Os dados gerados em espaços urbanos são geralmente associados com uma propriedade **espacial** e/ou **temporais**

Por exemplo:

A localização de estabelecimentos são dados espaciais

Modelagem de Dados

Os dados gerados em espaços urbanos são geralmente associados com uma propriedade **espacial** e/ou **temporais**

Por exemplo:

Dados meteorológicos são dados temporais

Modelagem de Dados

Os dados gerados em espaços urbanos são geralmente associados com uma propriedade **espacial** e/ou **temporais**

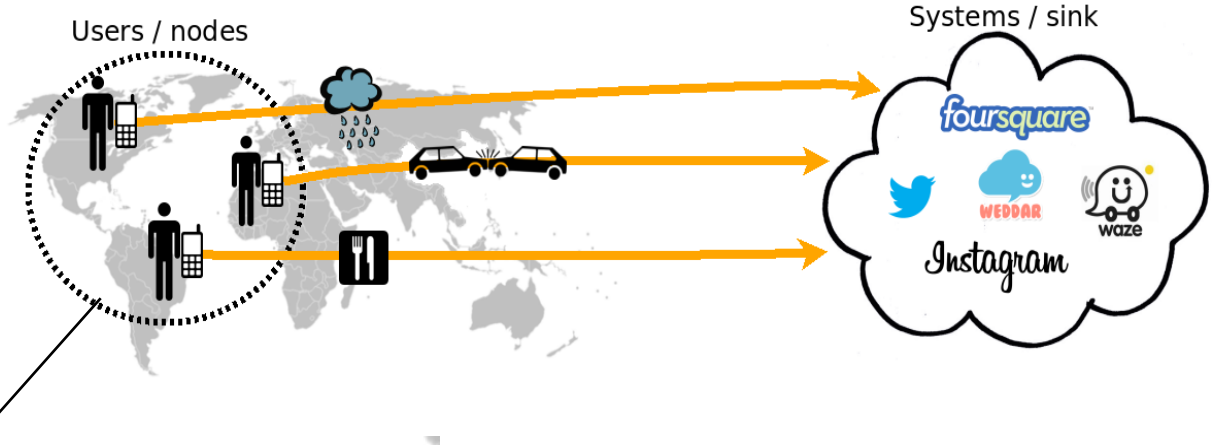
Por exemplo:

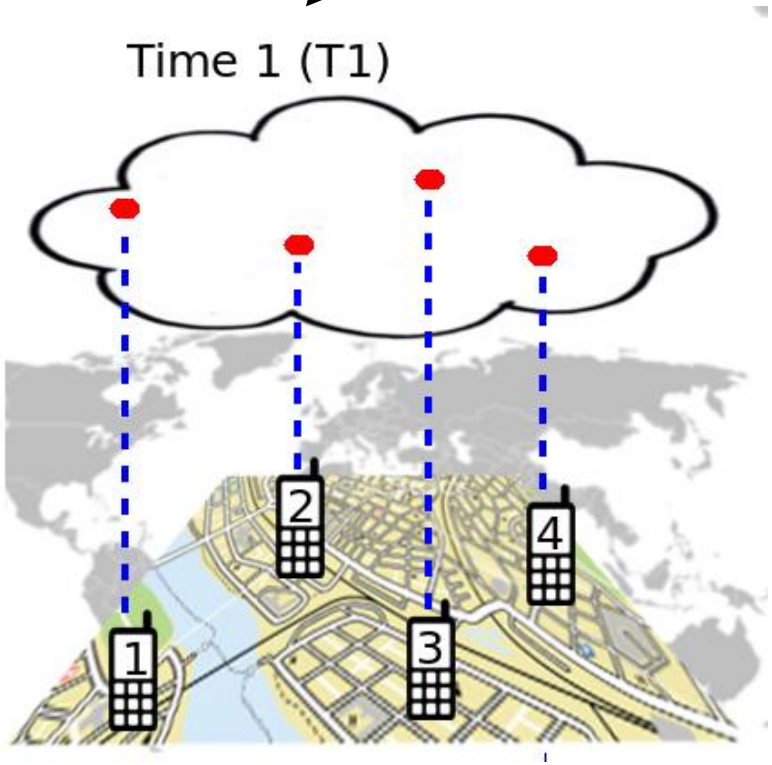
Dados de RSPs possuem propriedades **espaço-temporais** simultaneamente.

Modelagem de Dados

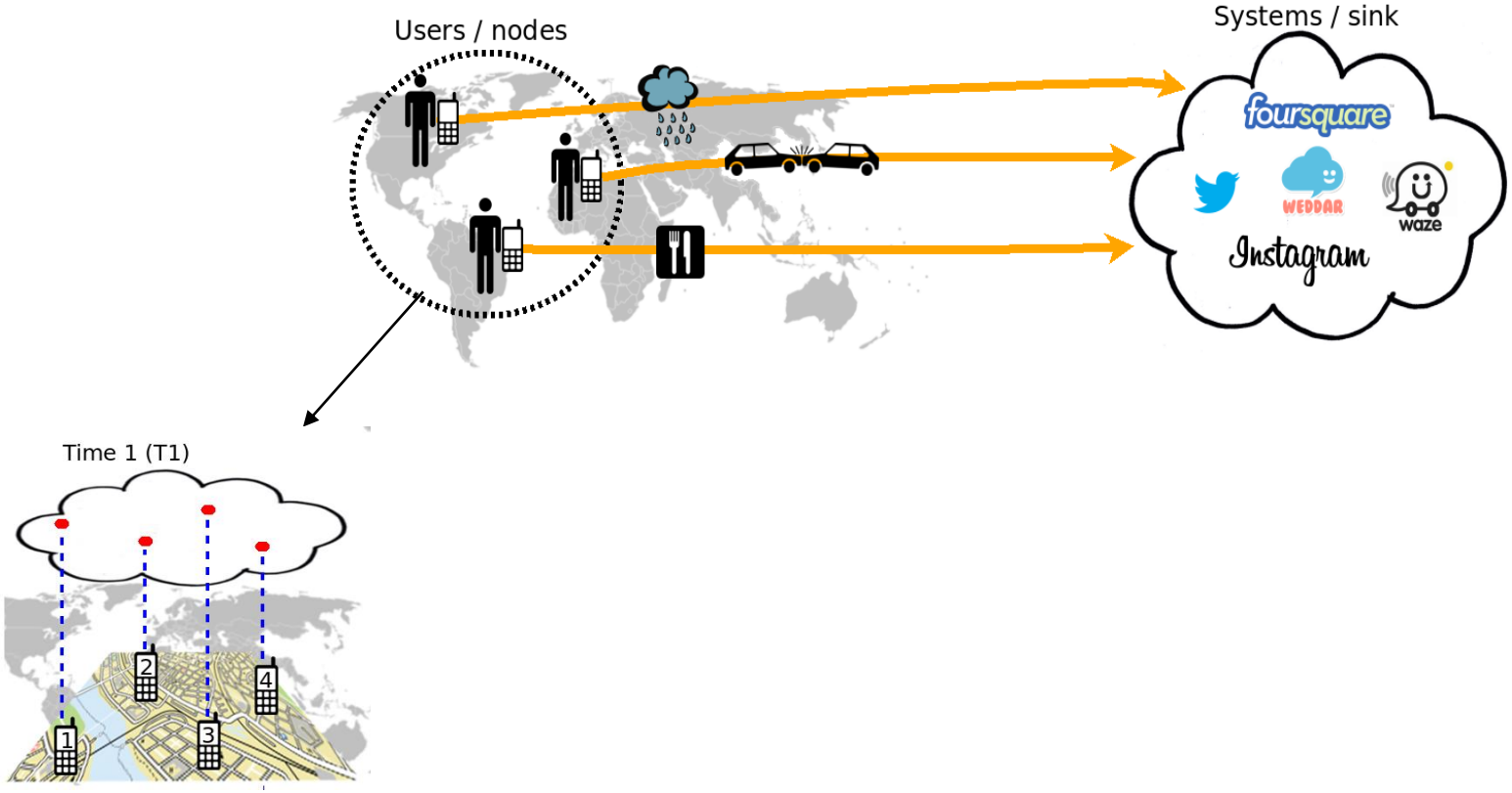
Existem várias formatos de dados para modelar dados de RSPs, sendo bastante popular o uso de grafos

Modelagem de Dados

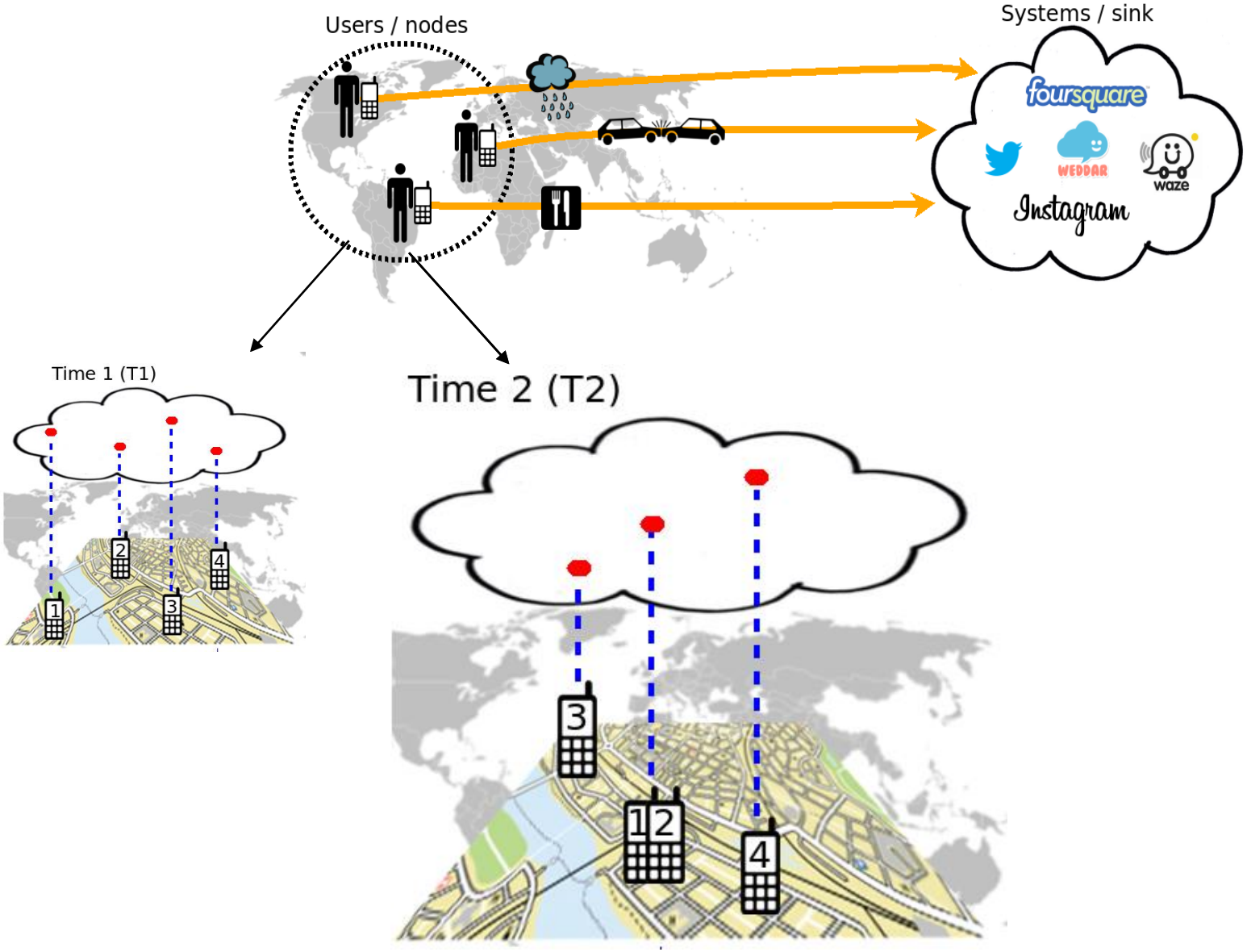




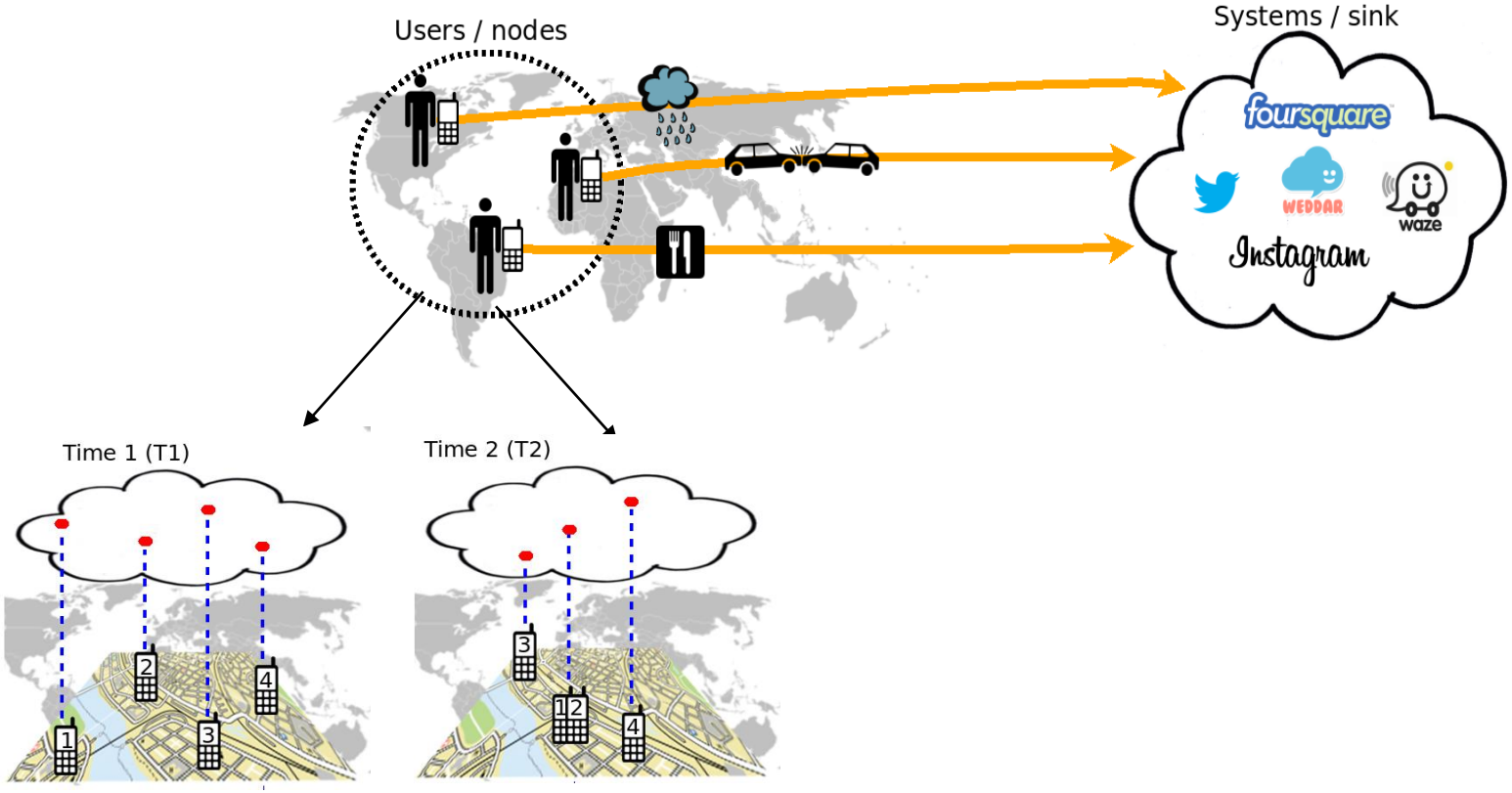
Modelagem de Dados



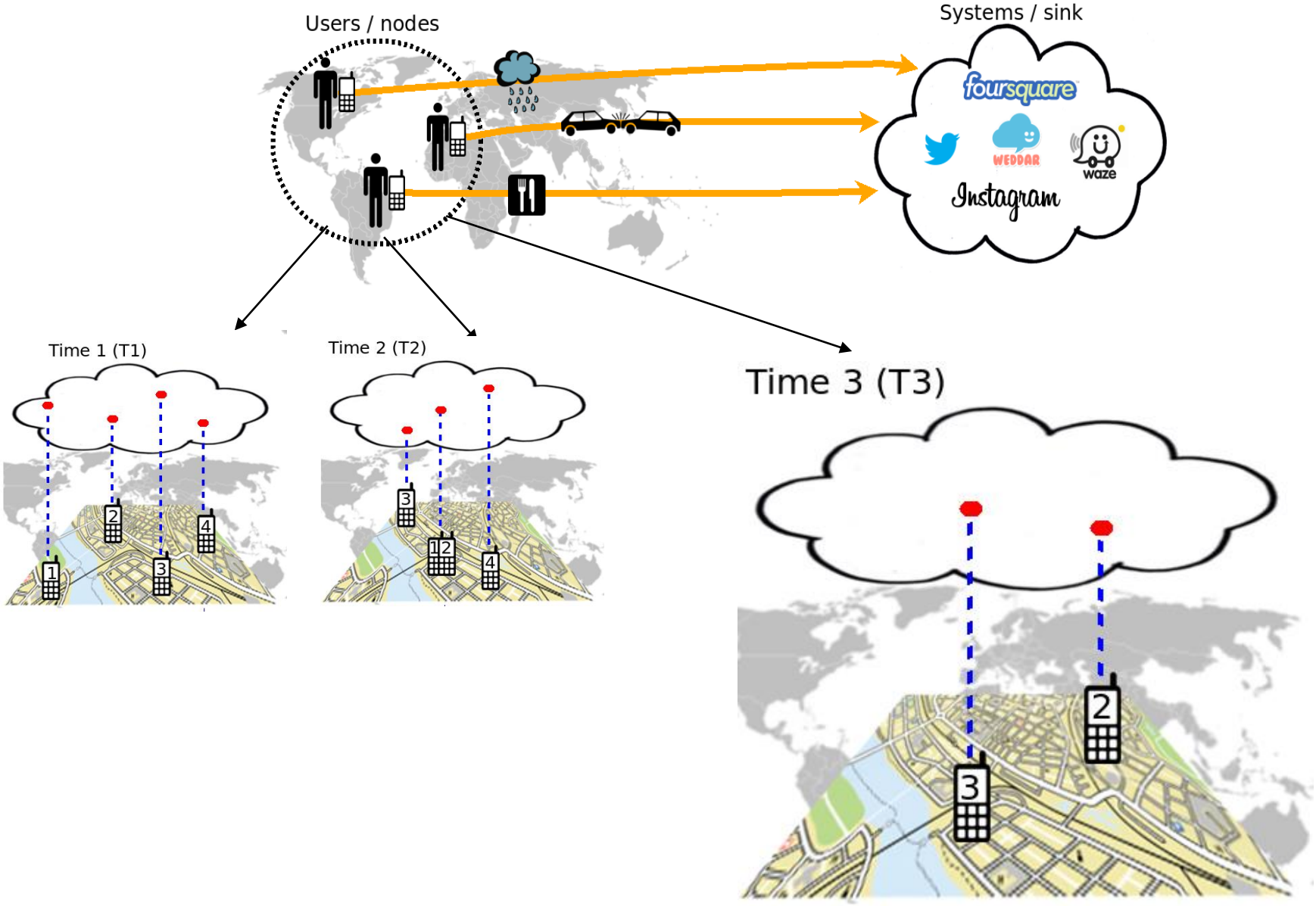
Modelagem de Dados



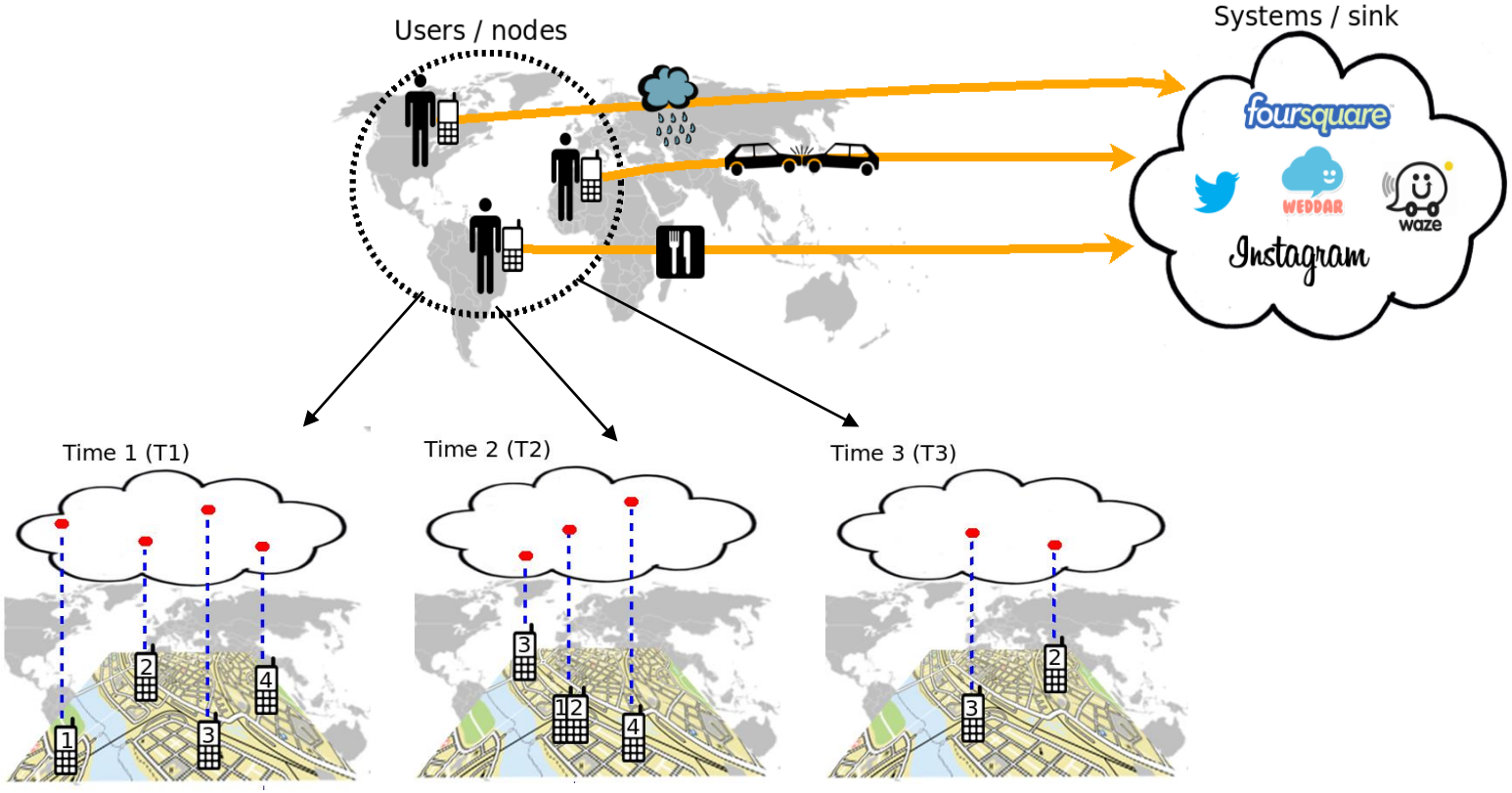
Modelagem de Dados



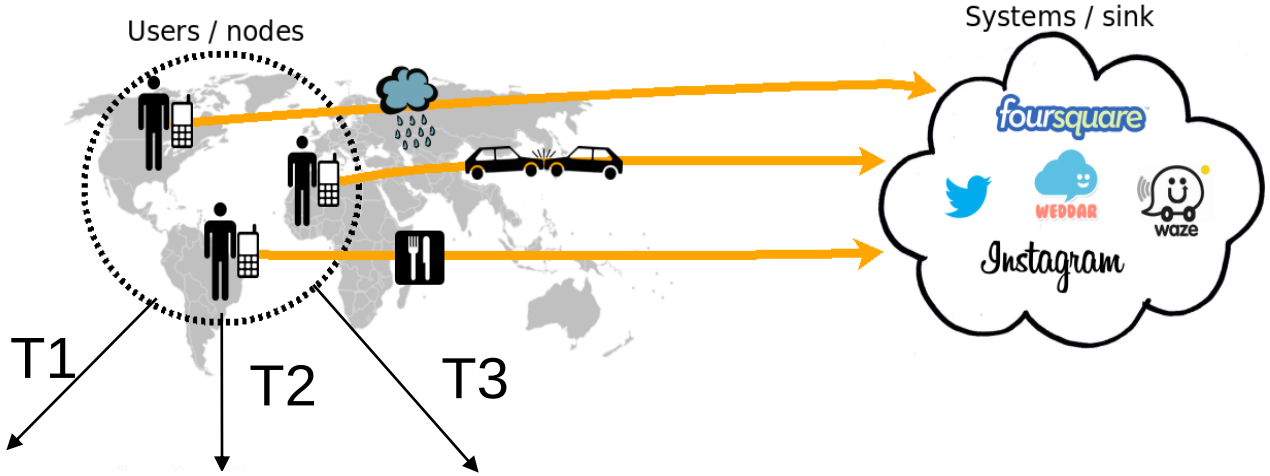
Modelagem de Dados



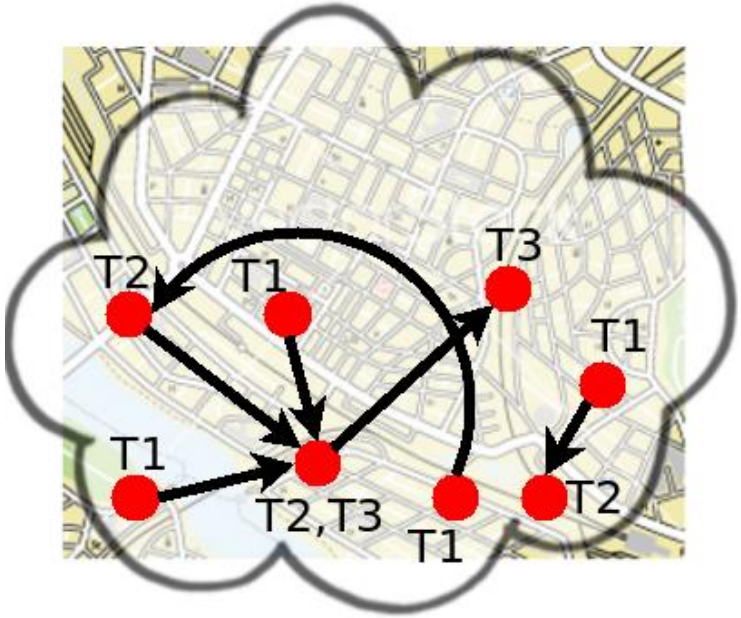
Modelagem de Dados



Modelagem de Dados

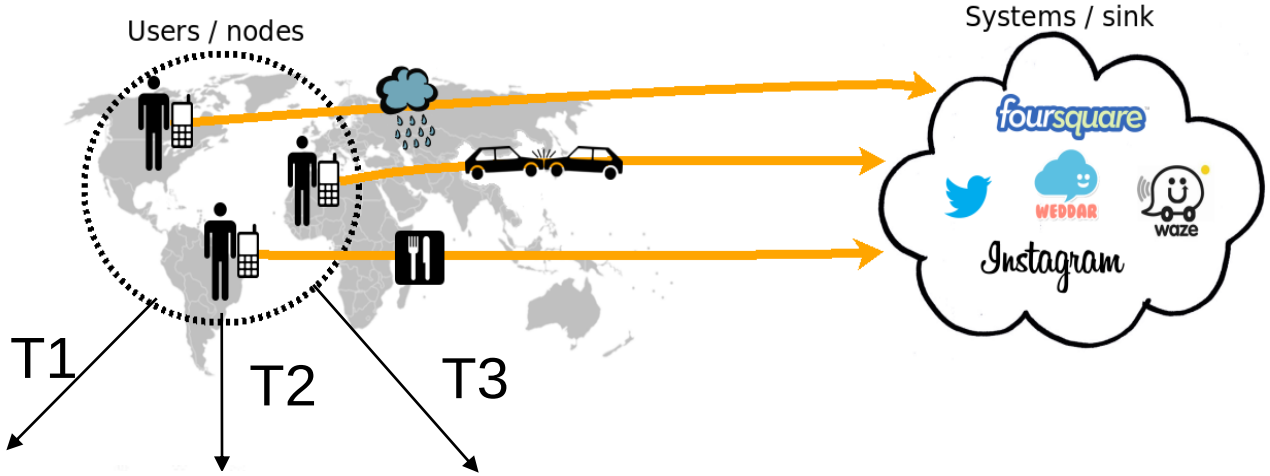


Visão agregada

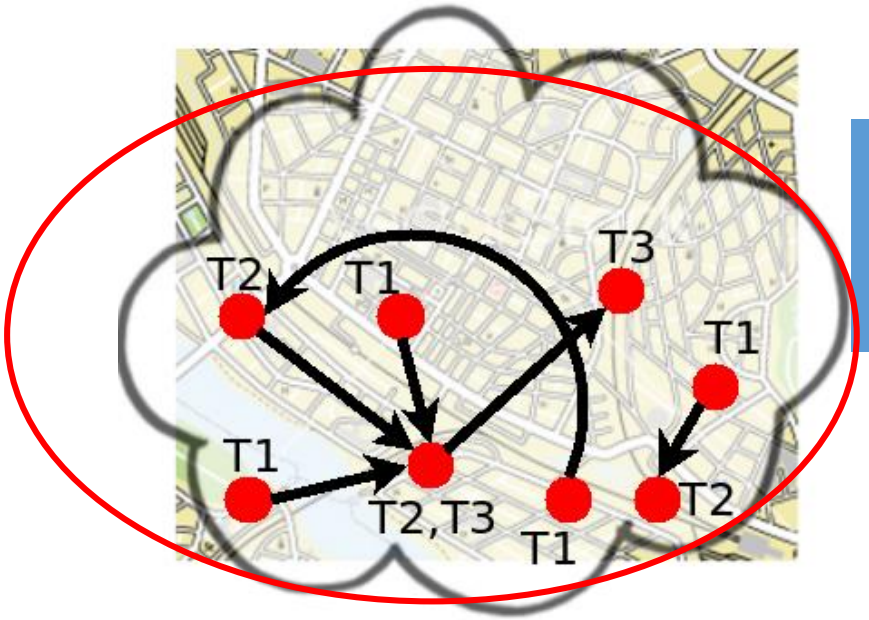


Grafo para estudar trajetórias

Modelagem de Dados



Visão agregada



A descoberta de conhecimento caminha junto com vários estudos que utilizam a teoria de grafos

Modelagem de Dados

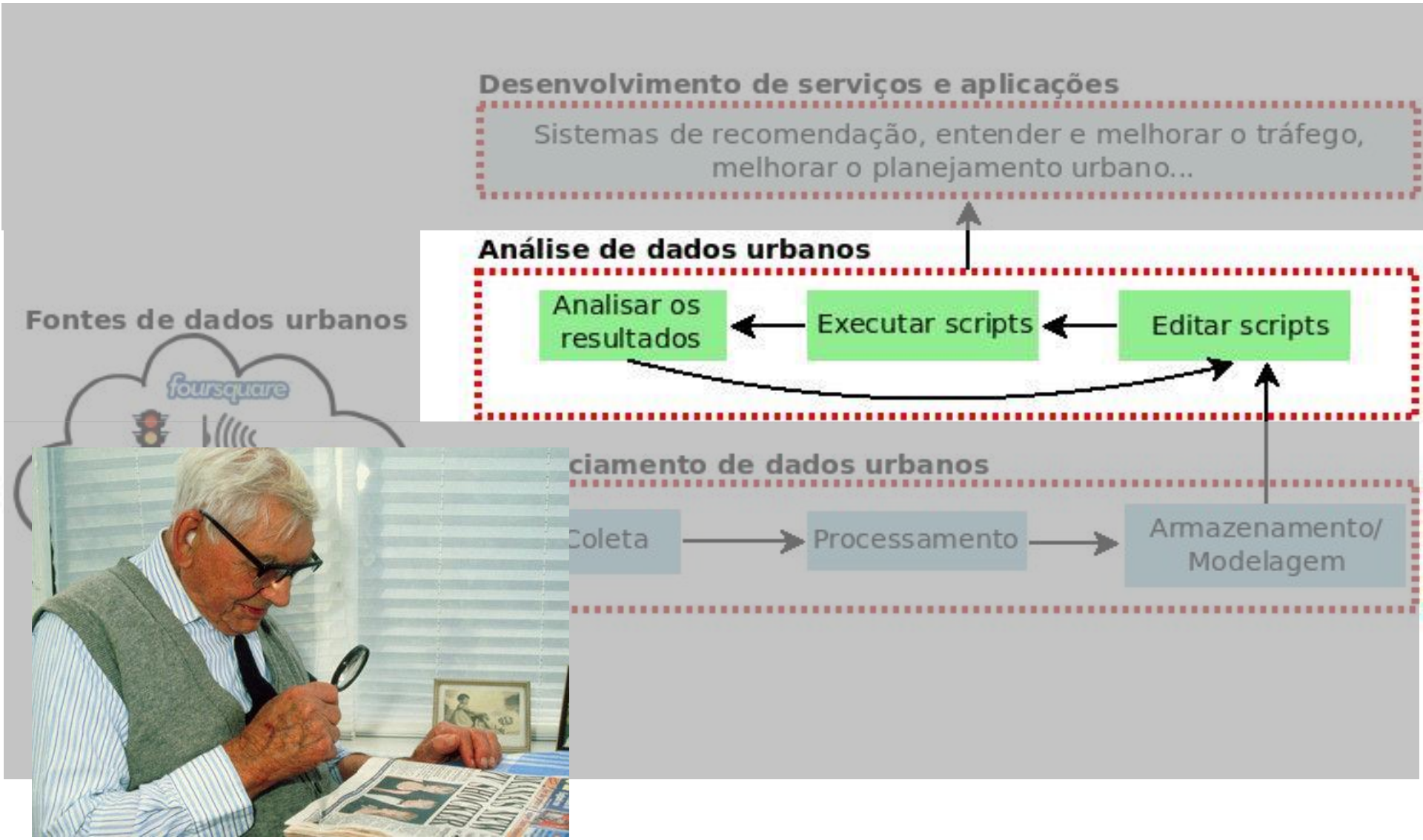
Alguns dos desafios de modelagem relacionados com:

- Dinâmica temporal
- Relacionamento de entidades

Discutidos posteriormente

Arcabouço da Computação Urbana

Arcabouço da Computação Urbana



Análise de dados: Python, Matlab/Octave e R são fáceis para começar a codificar

Qual usar?

python

R

MATLAB

Octave

Quando usar a linguagem Python?

Hello World

Java

```java
public class Main {
  public static void main(String[] args) {
    System.out.println("hello world");
  }
}
```

Python

```python
print "hello world";
```

File I/O

Java

```
File dir = new File(".");// get current directory
File fin = new File(dir.getCanonicalPath() + File.separator
                                + "Code.txt");
FileInputStream fis = new FileInputStream(fin);
// //Construct the BufferedReader object
BufferedReader in = new BufferedReader(new InputStreamReader(fis));
String aLine = null;
while ((aLine = in.readLine()) != null) {
        // //Process each line, here we count empty lines
        if (aLine.trim().length() == 0) {
        }
}

// do not forget to close the buffer reader
in.close();
```

Python

```
myFile = open("/home/xiaoran/Desktop/test.txt")

print myFile.read();
```

Vantagens:

- Facilidade de transformar ideias em código
- Simplicidade de manipular *strings*

Python é uma das principais linguagens para a construção de scripts para:

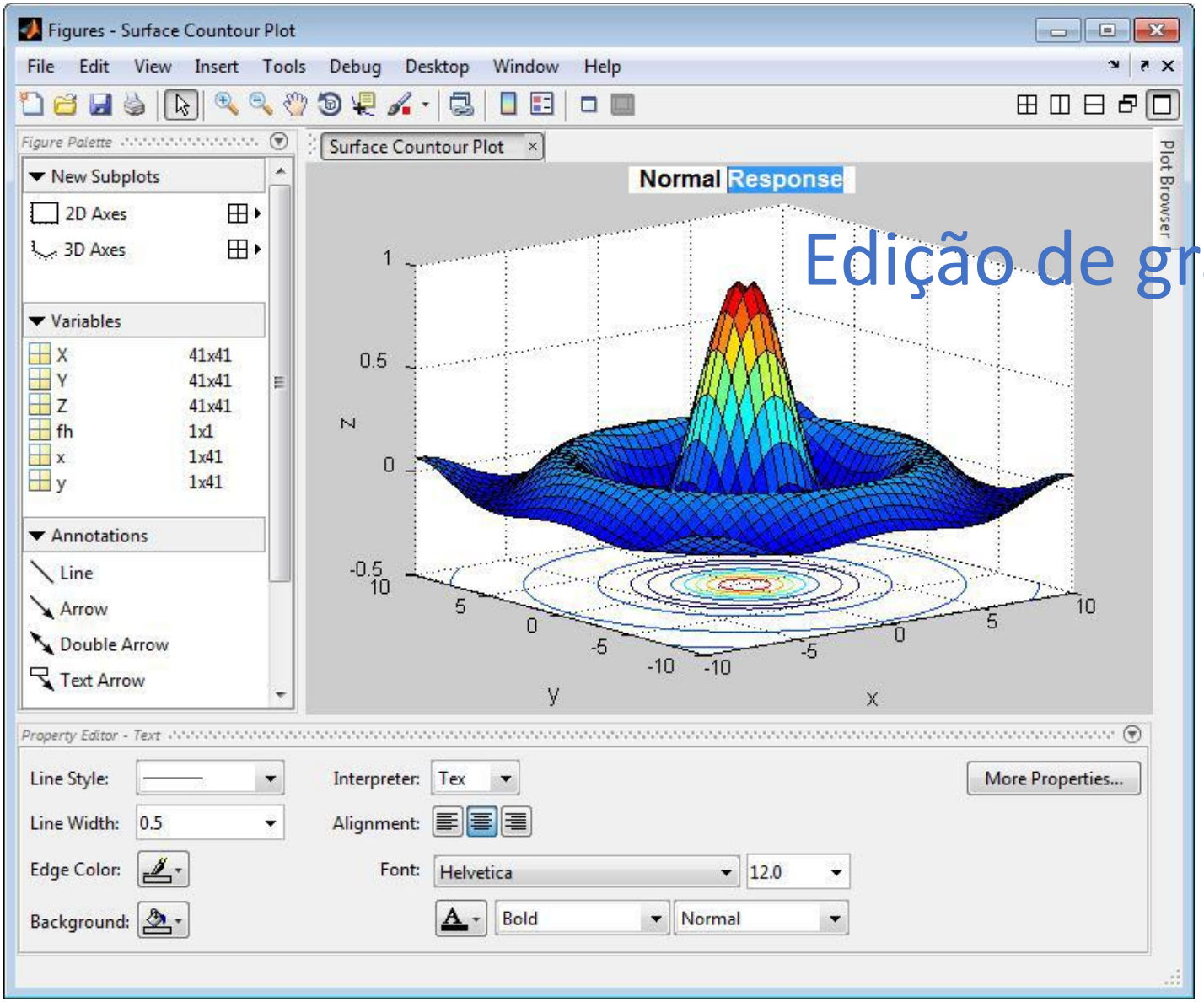
- Limpeza e formatação dos dados
- Obter dados de websites, arquivos ou outras fontes de dados

Quando usar as linguagens Matlab ou R?

Operações matemáticas típicas como multiplicação de matrizes funcionam de forma bem simples e direta

Grande variedade de pacotes extras disponíveis publicamente

Análise estatística “pesada” ou gráficos



Edição de gráficos

Quais são as propriedades dos dados de RSPs?

Datasets

Compartilhamento de localizações

System	# check-ins	Interval
Foursquare1	~ 5 million	April 2012 (1 week)
Foursquare2	~ 12 million	Feb2010-Jan2011
Foursquare3	~ 4 million	2 weeks May 2013
Gowalla	~ 6 million	Feb2009 - Oct2010
Brightkite	~ 4 million	Apr2008 - Oct2010
TOTAL	**+30 million**	

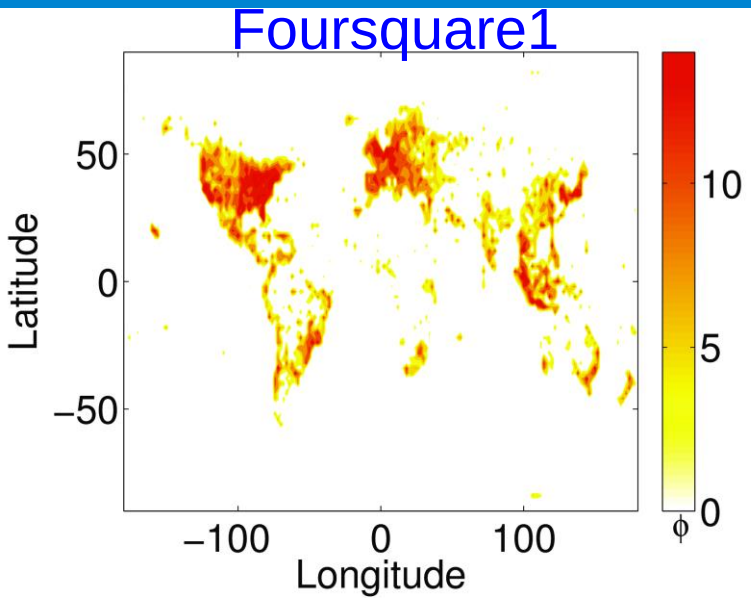
Compartilhamento de fotos

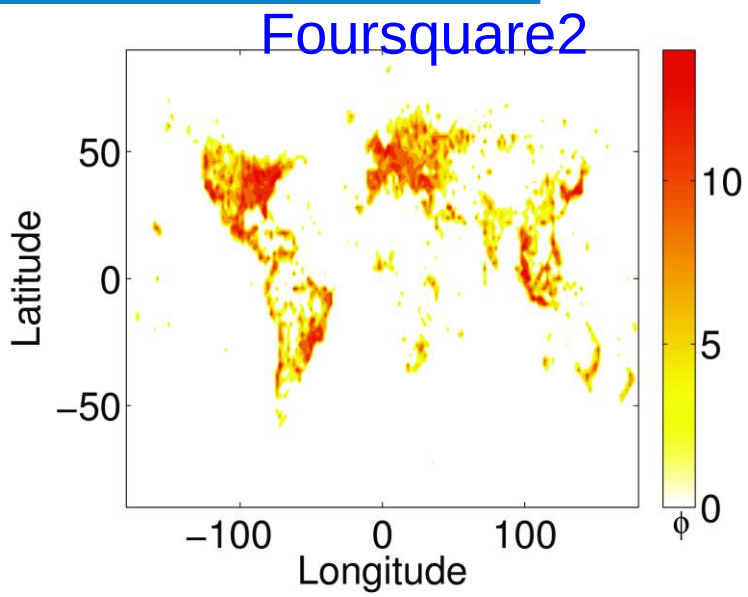
Instagram1	~ 2 million	Jun2012-Jul2012
Instagram2	~ 2 million	2 weeks May 2013
TOTAL	**~4 million**	

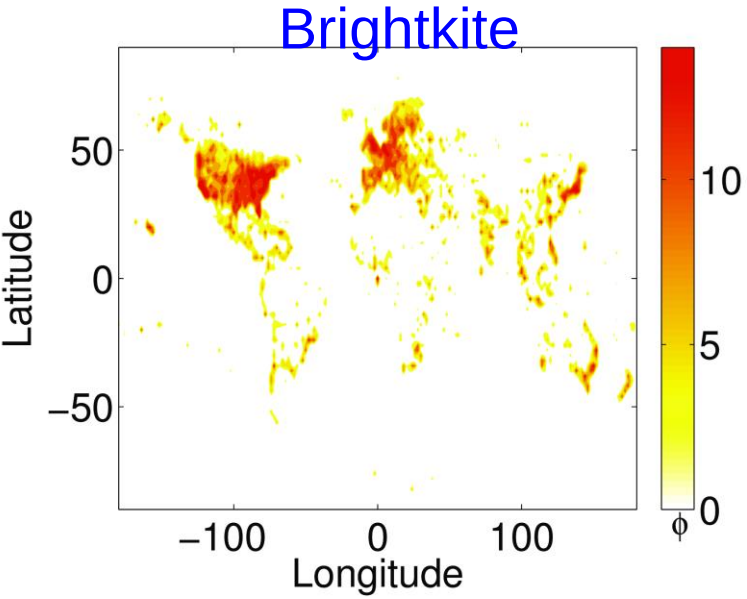
Compartilhamento de problemas de trânsito

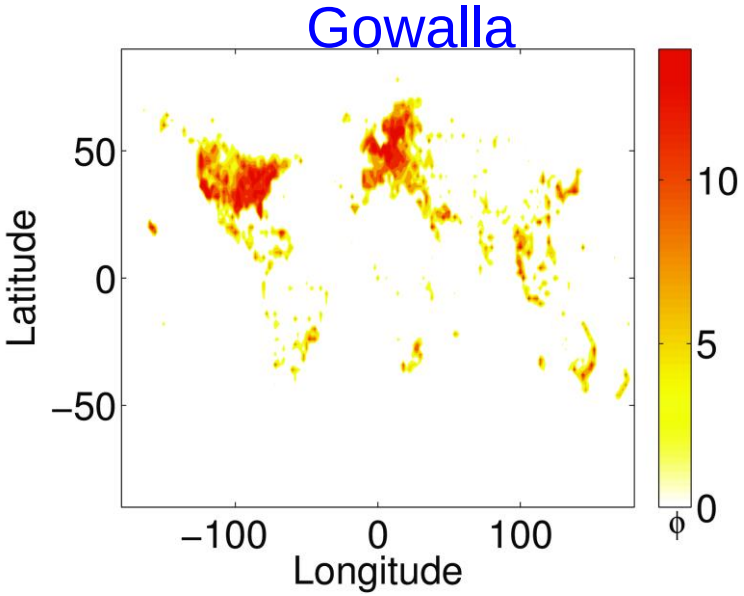
Waze	+ 212 thousands	Dec2012-June2013

Cobertura - global

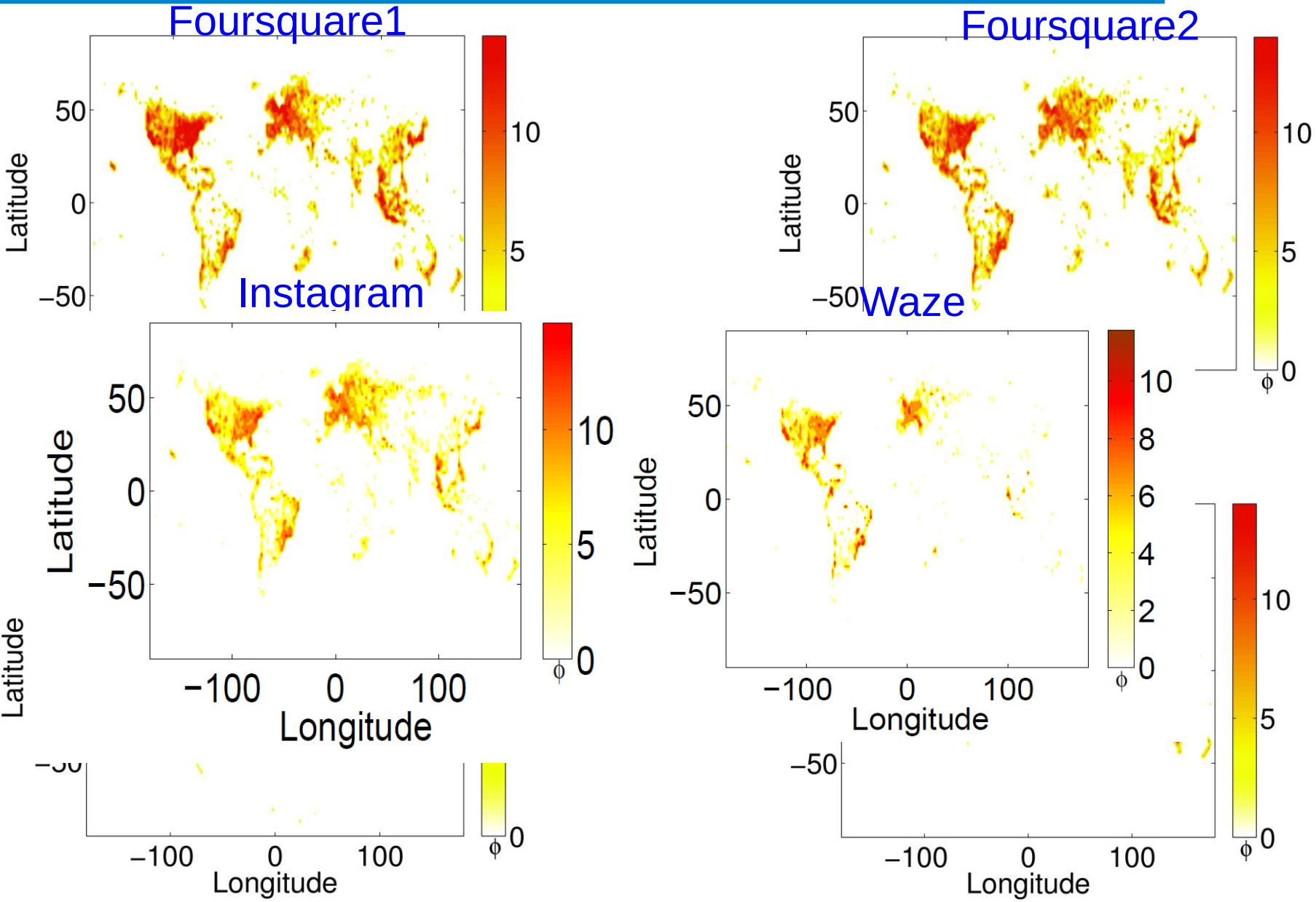








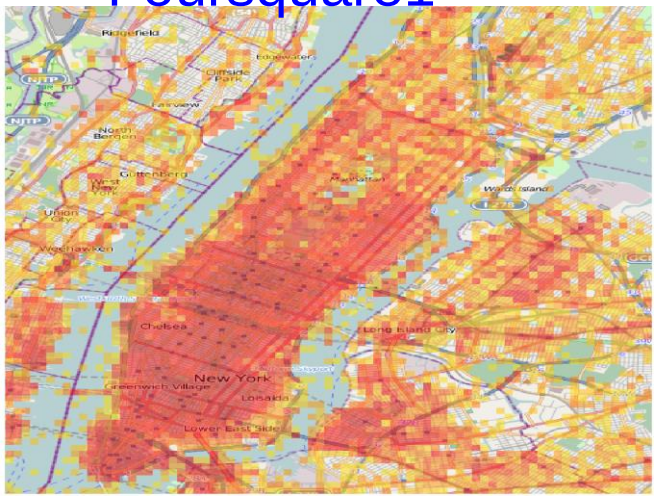
Cobertura - global



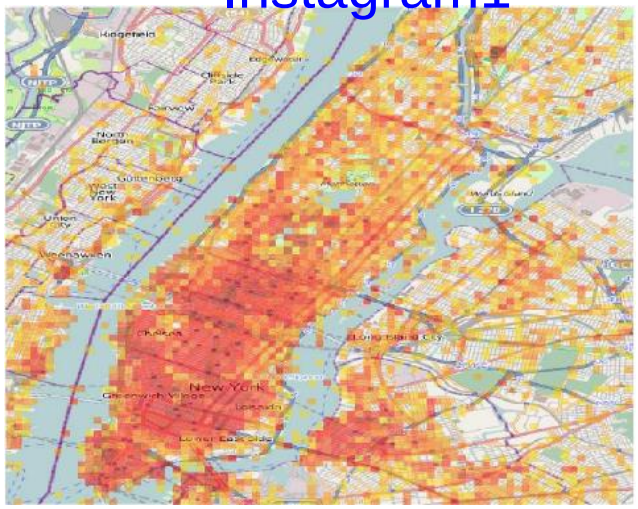
Cobertura - cidades

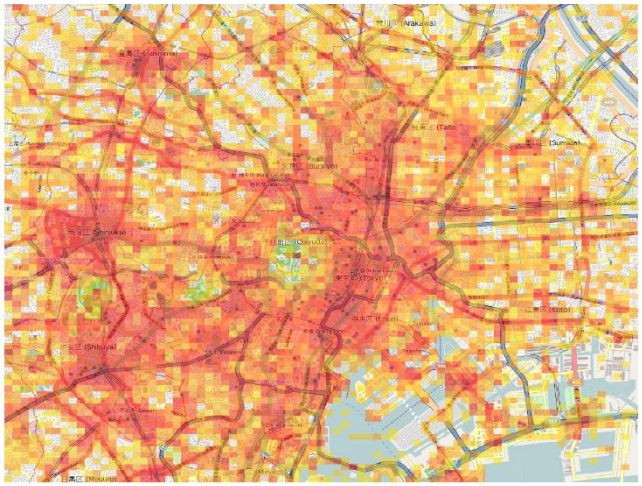
Cobertura abrangente

Foursquare1



Instagram1



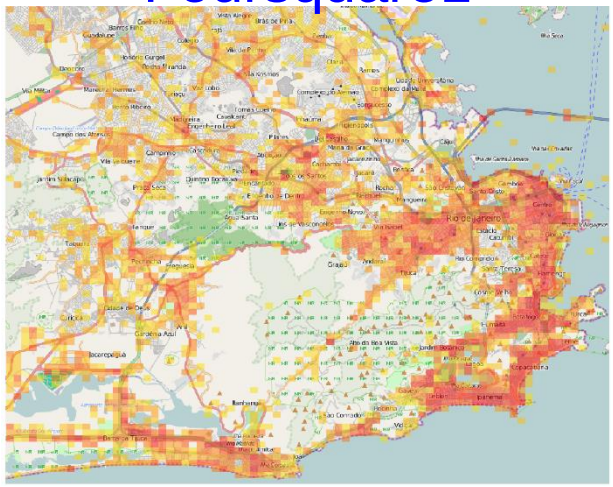


Cobertura - cidades

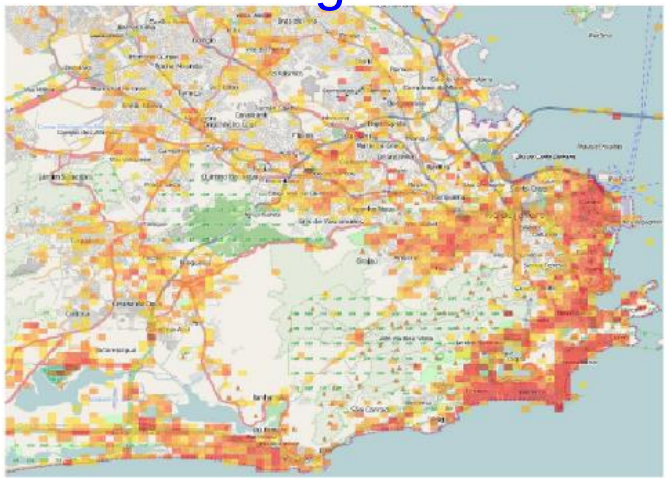
Cores mais escuras indicam maior atividade de sensoriamento em uma área

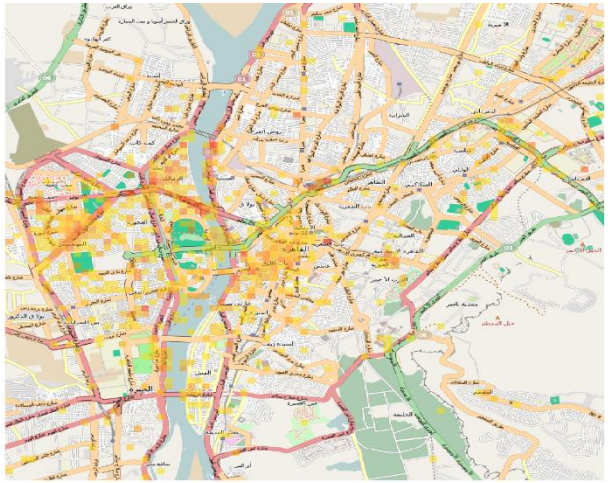
Cobertura - cidades

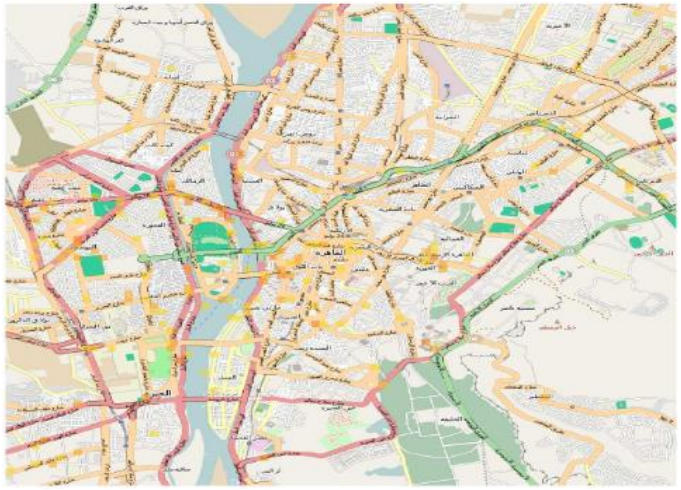
Foursquare1



Instagram1



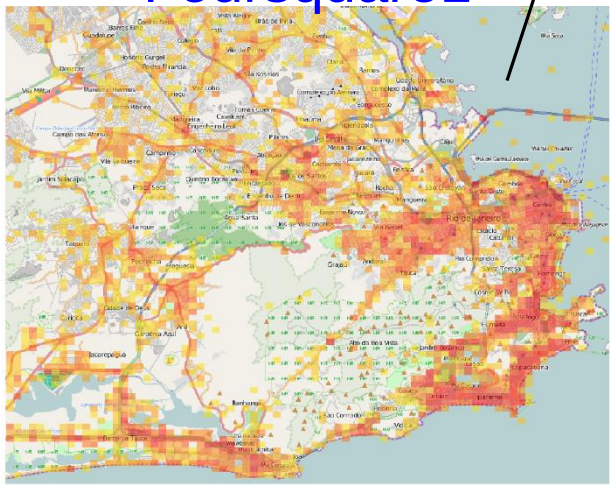




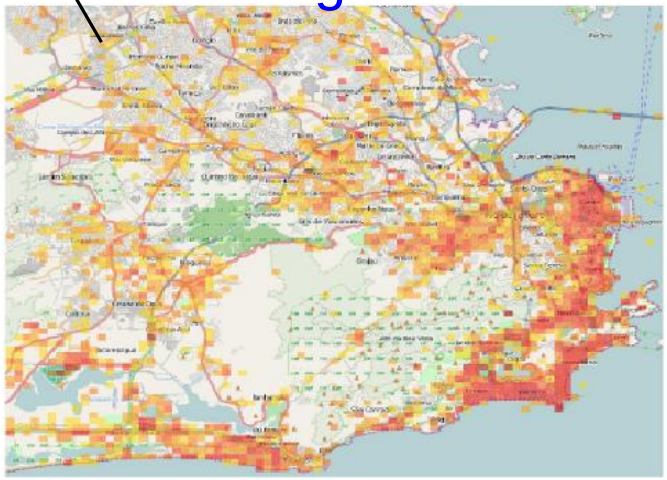
Cobertura - cidades

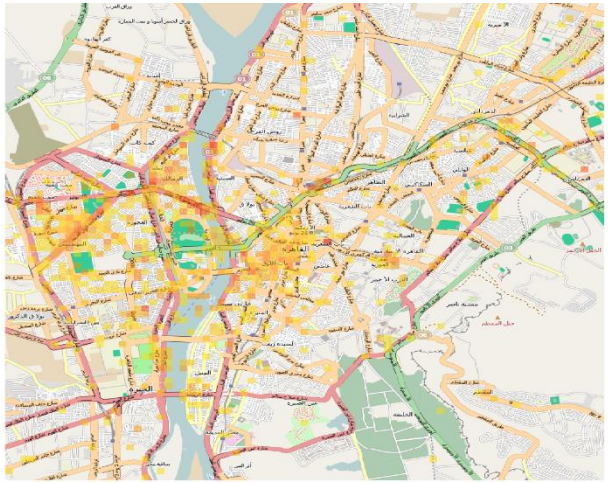
Aspectos geográficos influenciam

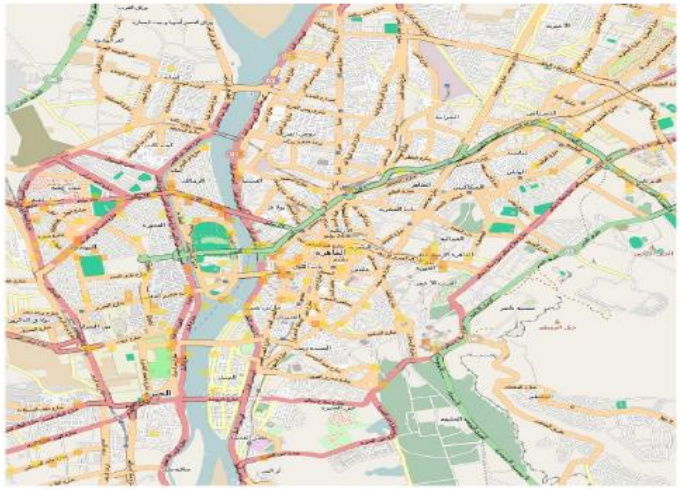
Foursquare1



Instagram1



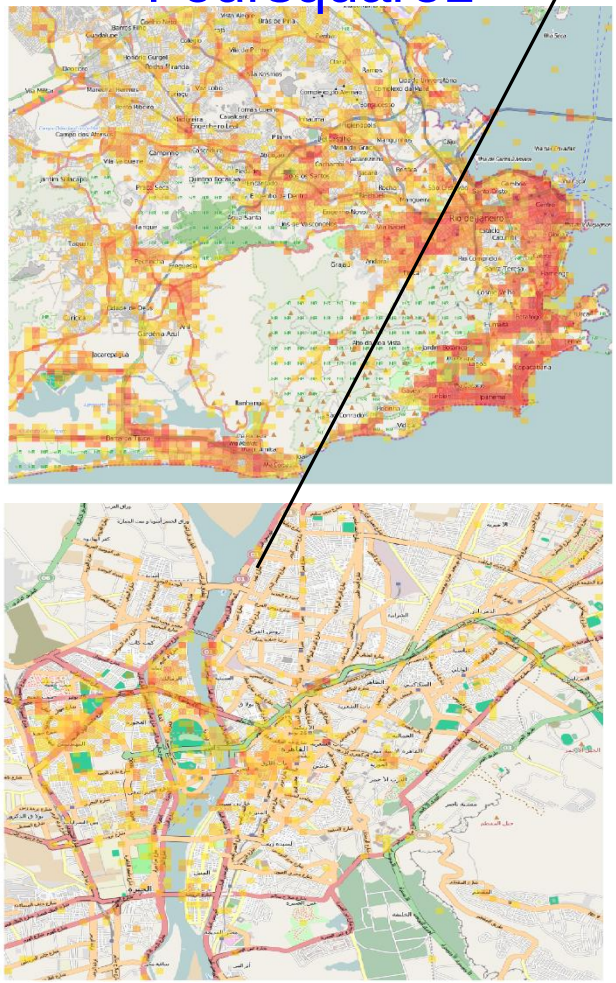




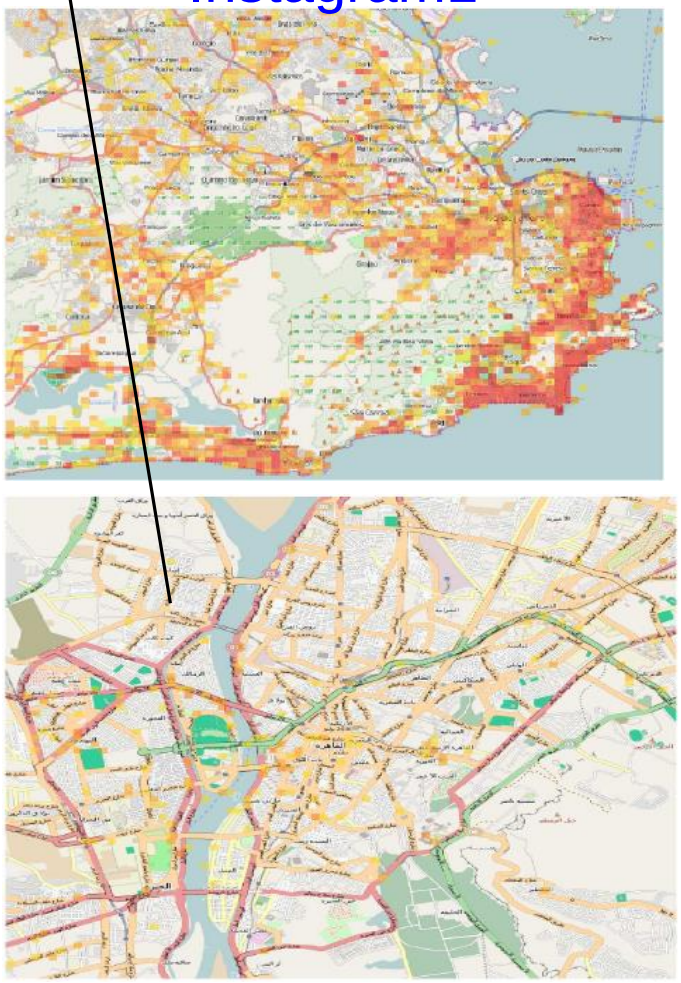
Cobertura - cidades

Além dos aspectos econômicos, diferenças culturais podem explicar

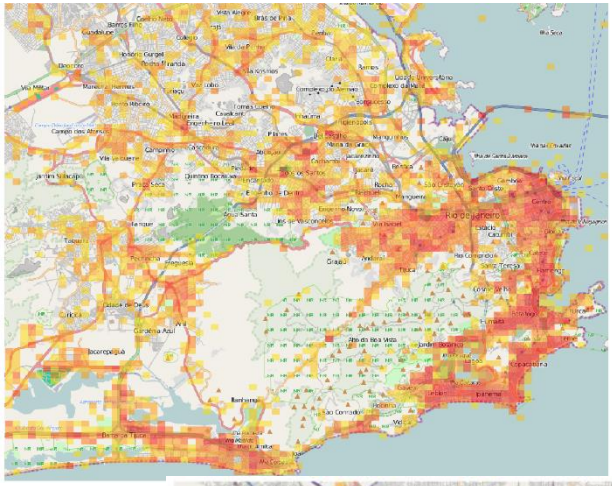
Foursquare1

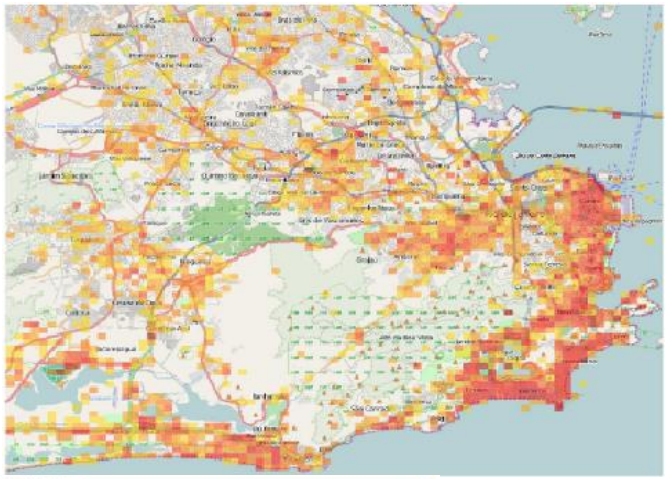


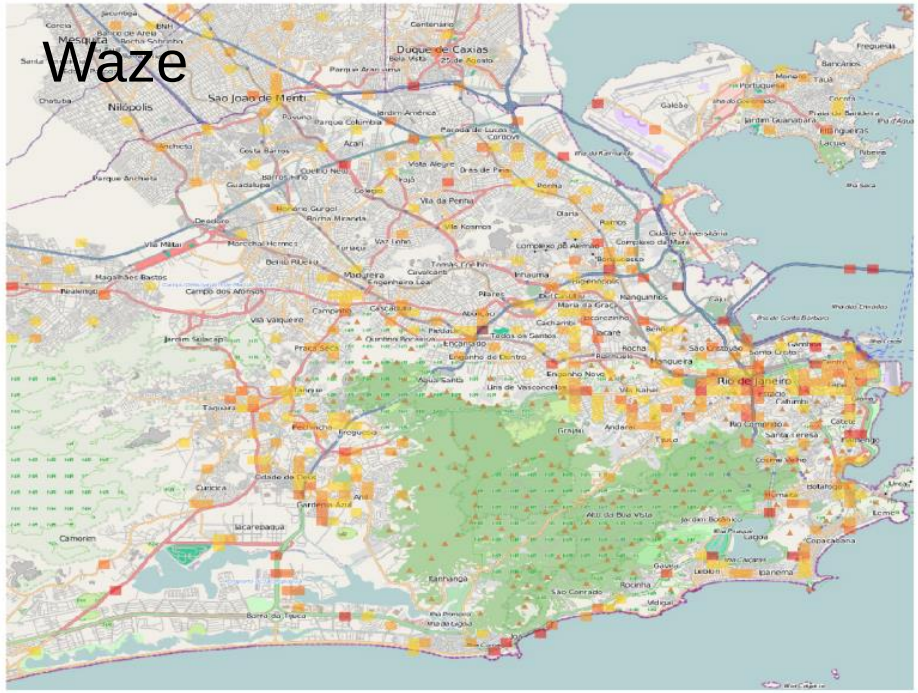
Instagram1



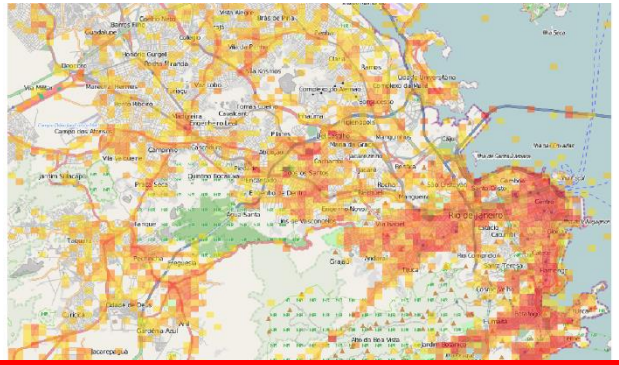
Cobertura - cidades

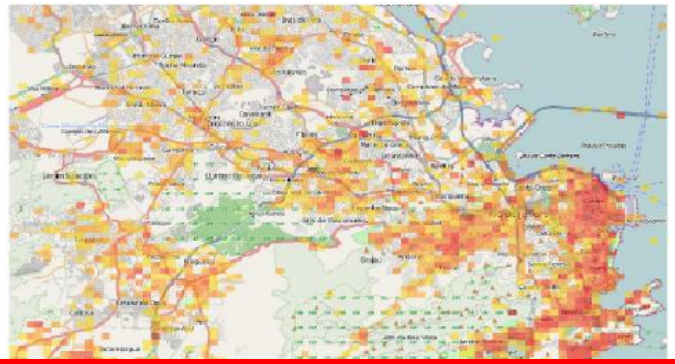






Cobertura - cidades





O propósito da aplicação também influencia
Usuários podem ter menos oportunidades para compartilhar

Mostrar dados em mapas

QGIS

QGIS é um GIS *open-source* bastante popular

Facilita a manipulação, análise e visualização de dados espaciais

Instalação do QGIS:

Linux (ubuntu): sudo apt-get install qgis

Windows: http://qgis.org/pt_BR/site/forusers/download.html

Um mapa simples

Dataset sobre terremotos: signif.txt

signif.txt - Notepad

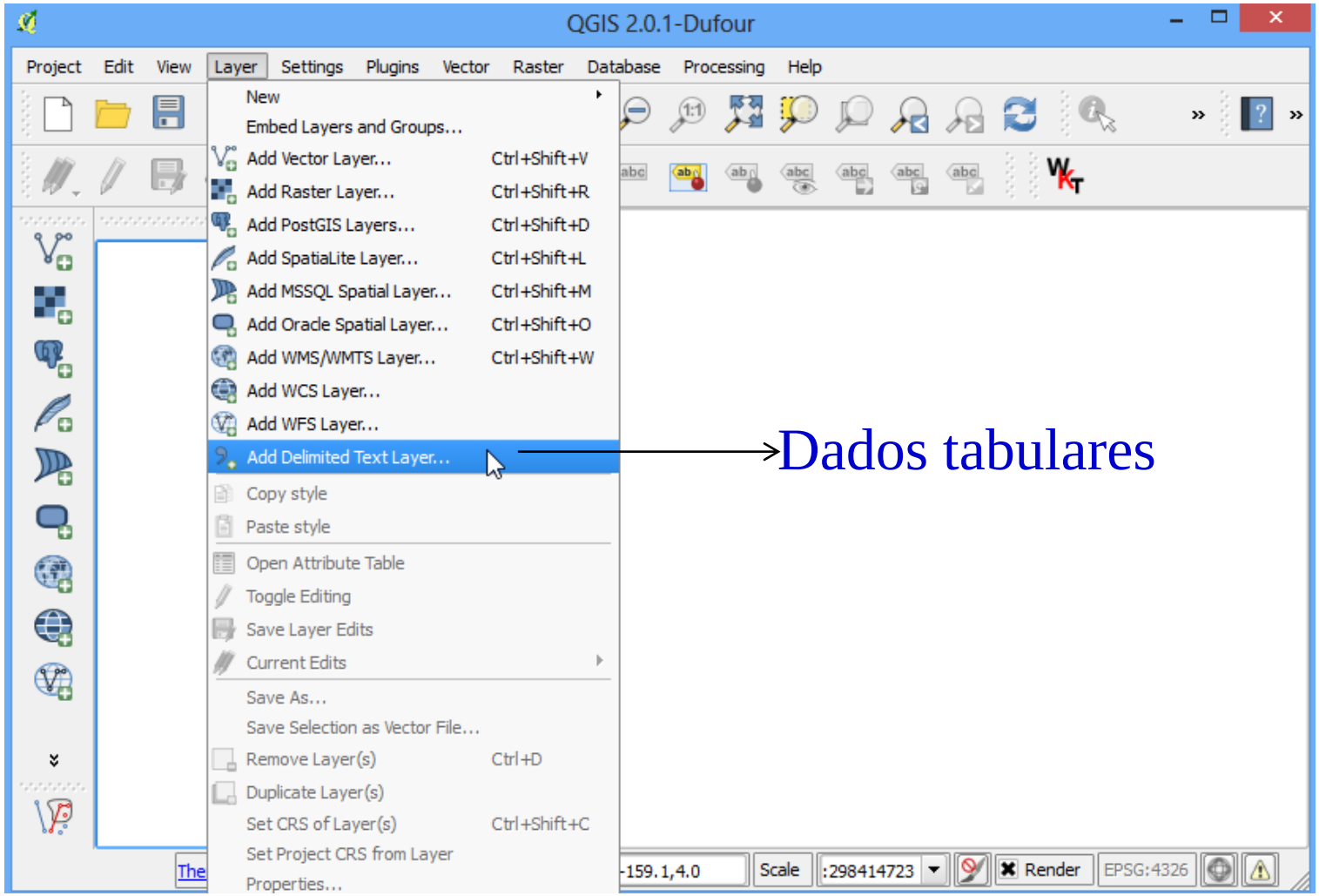
File Edit Format View Help

```
STATE   LOCATION_NAME   LATITUDE        LONGITUDE       REGION_CODE     DEATHS  DEATHS_DESCRIPTION
        10      ISRAEL          ISRAEL:  ARIHA (JERICHO)          31.500          35.300       140
                                                9713    Tsu     -480    9       29
                                                1
  103.900       30                                                                2
GANSU PROVINCE:  LONGXI   34.900          104.700       30              3
        3                               41              23
                                                                                51              57
                                UKRAINE         UKRAINE:  BLACK SEA       44.700          33.300
                                        1                               67              155
                                                3       1001    4
                                                                                79              340
        350     10
                2
        438
        119             477     9       25
                                                                        128             521
                        139     Tsu     551     7       9
                                        147             558     12      25
649     9       12                                              5.5
        715
175             745     6       5                                               7.9
2               3               187             778
        199             811
844     9       18
        219             853
857     4
```

Ln 1, Col 1

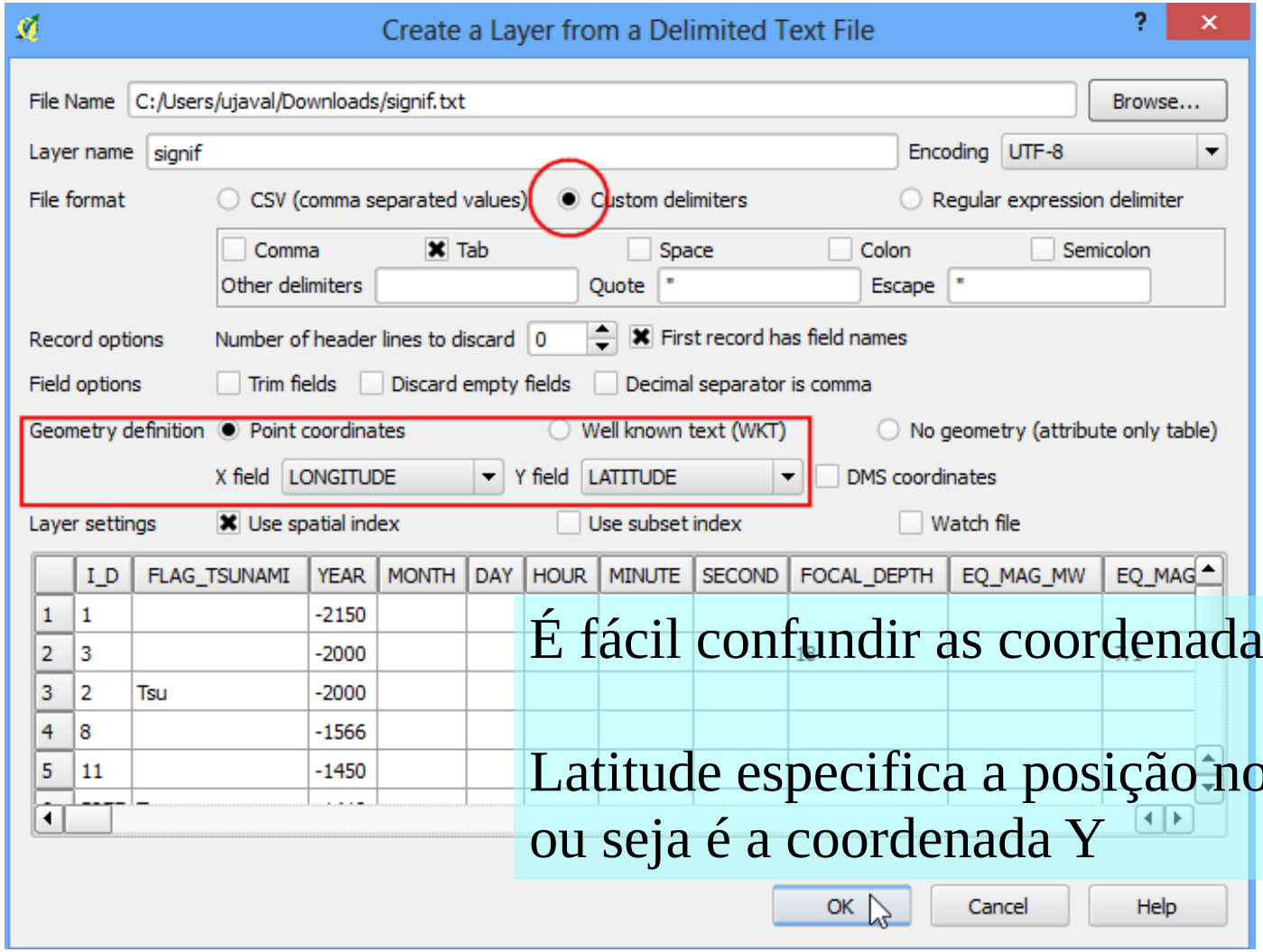
Adaptado de http://www.qgistutorials.com/en/docs/importing_spreadsheets_csv.html

Um mapa simples



Adaptado de http://www.qgistutorials.com/en/docs/importing_spreadsheets_csv.html

Um mapa simples

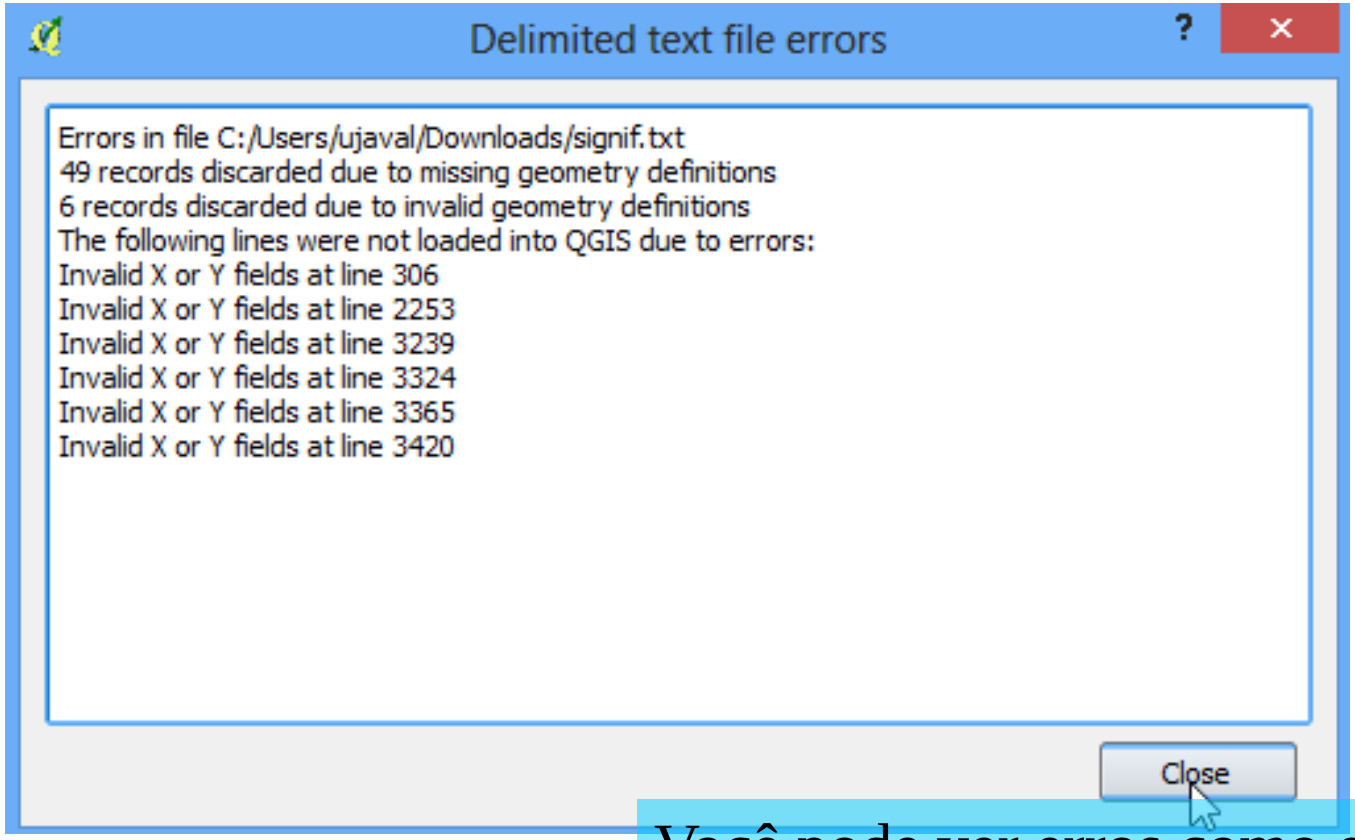


É fácil confundir as coordenadas X e Y

Latitude especifica a posição norte-sul, ou seja é a coordenada Y

Adaptado de http://www.qgistutorials.com/en/docs/importing_spreadsheets_csv.html

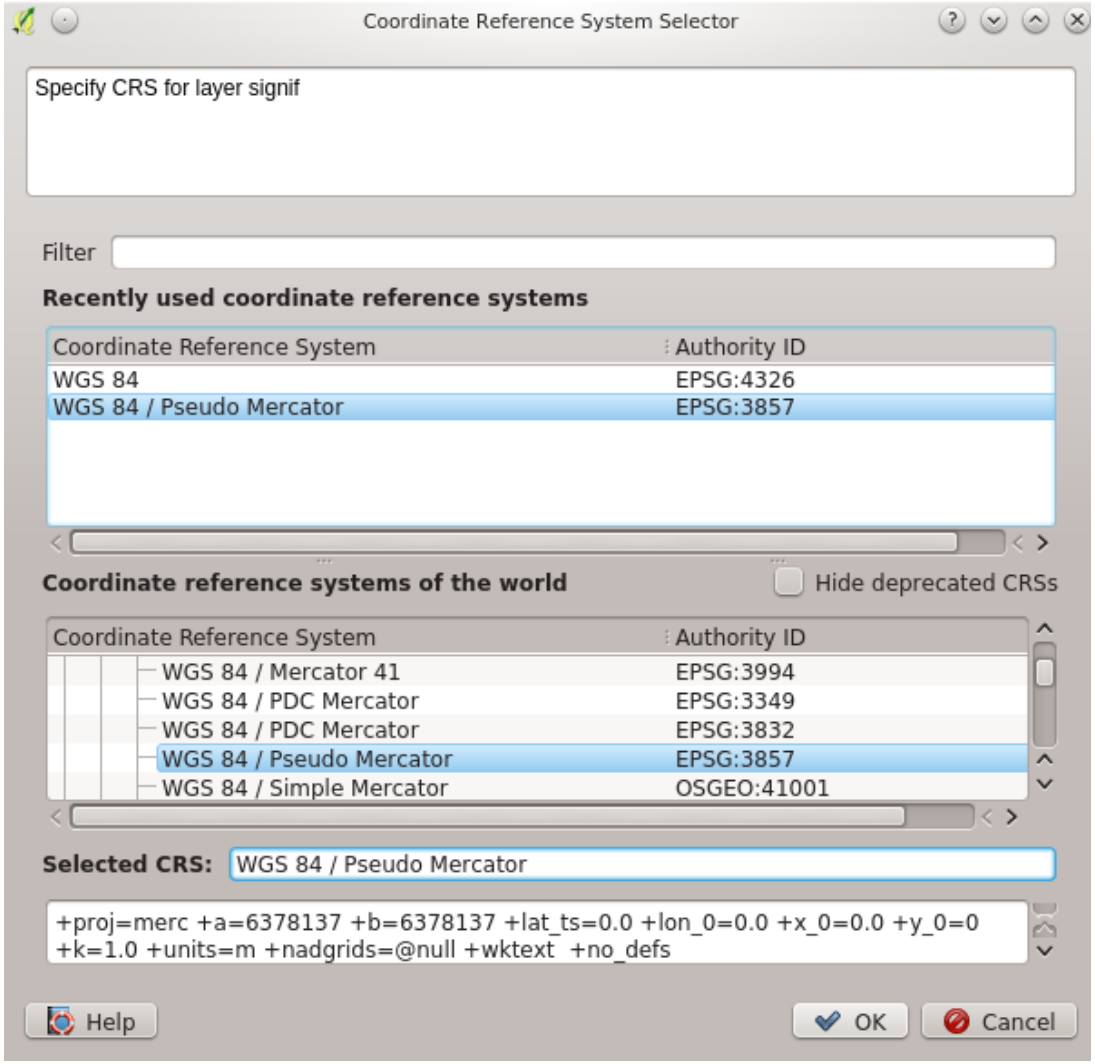
Um mapa simples



Você pode ver erros como estes.
Neste caso é por causa de X e Y faltantes.
Você pode ignorar neste exemplo.

Adaptado de http://www.qgistutorials.com/en/docs/importing_spreadsheets_csv.html

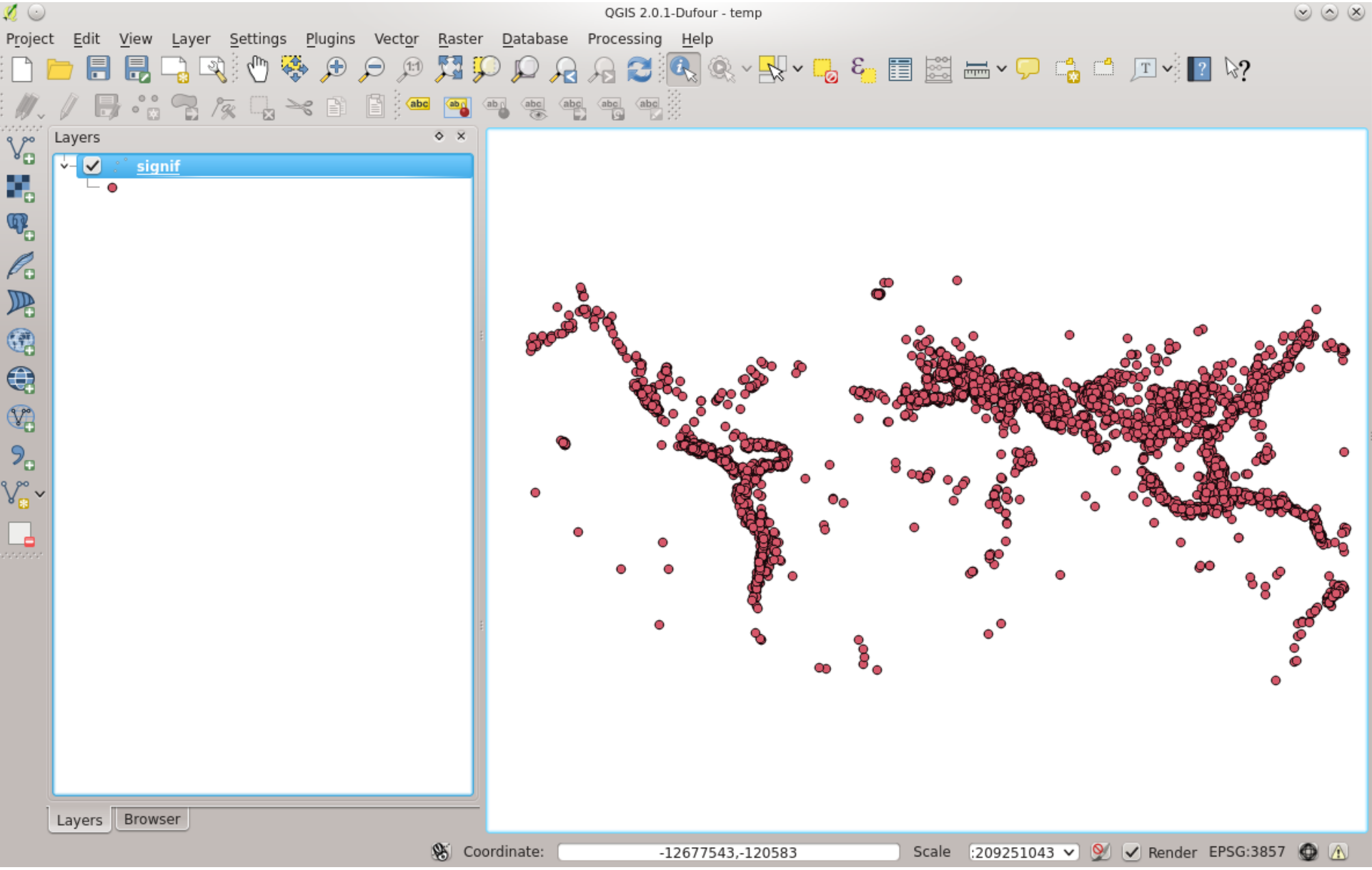
Um mapa simples



Coordinate reference system:

Como o dataset contém dados de latitude e longitude devemos selecionar WGS 84

Um mapa simples



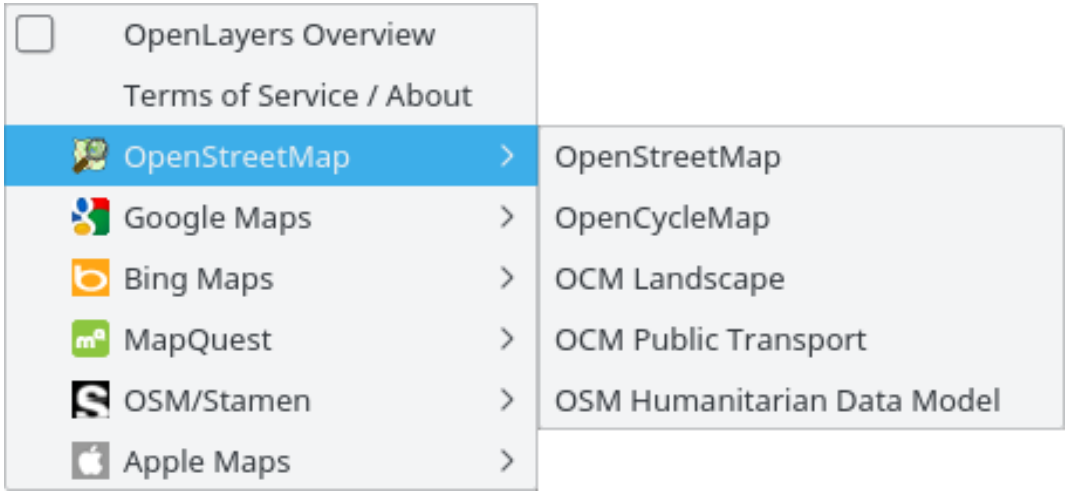
Plugin para mapas

- O plugin que vem no QGIS para mapas do openstreetmaps pode estar com problemas

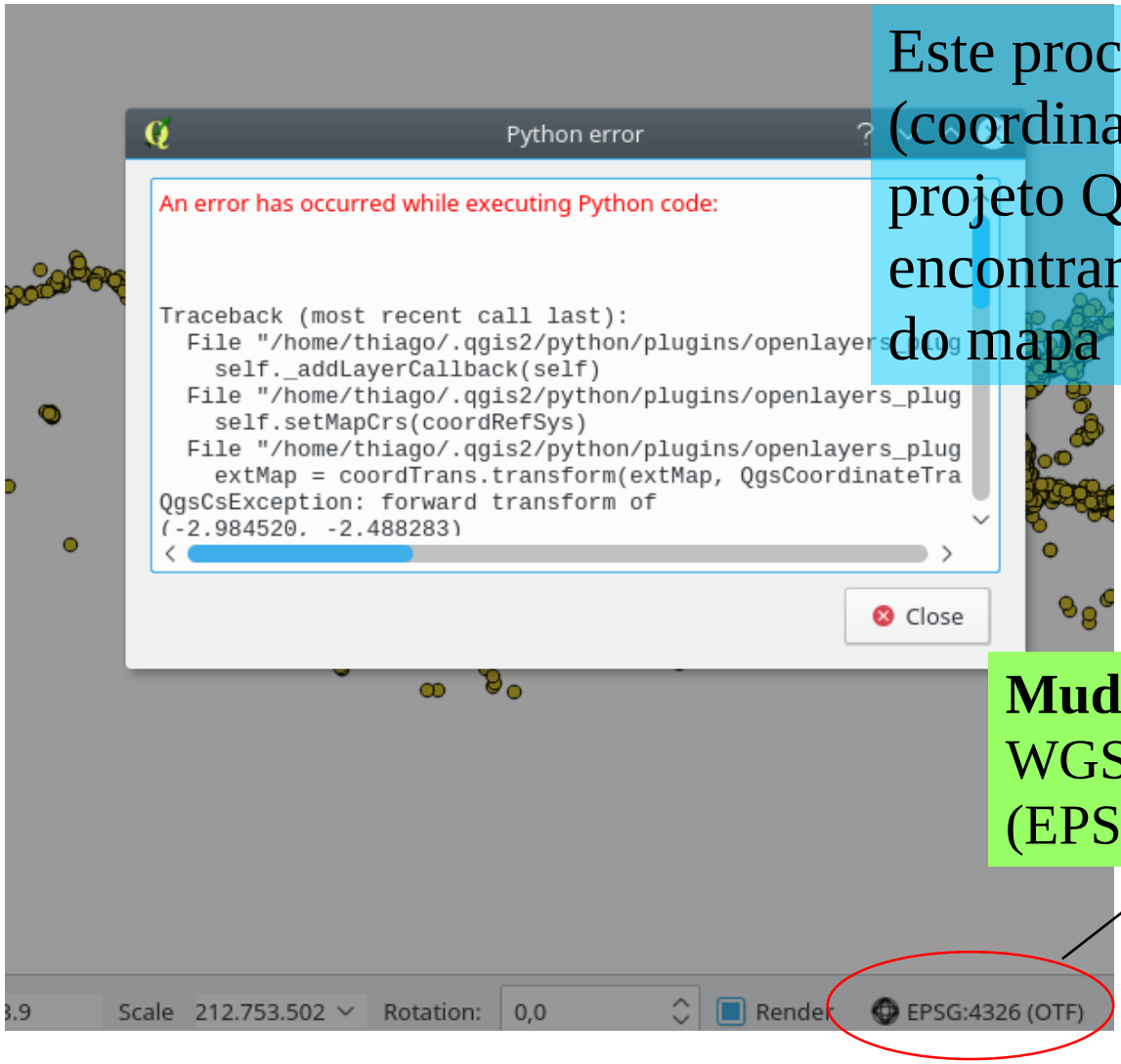
- **Solução:**
- No QGIS2.8 vá em "Plugins" -> "Manage Installed Plugins..."
- Procure por OpenLayers e realize a instalação.

Plugin para mapas

- A lista de camadas OSM aparece em: "Web" -> "OpenLayers plugin"



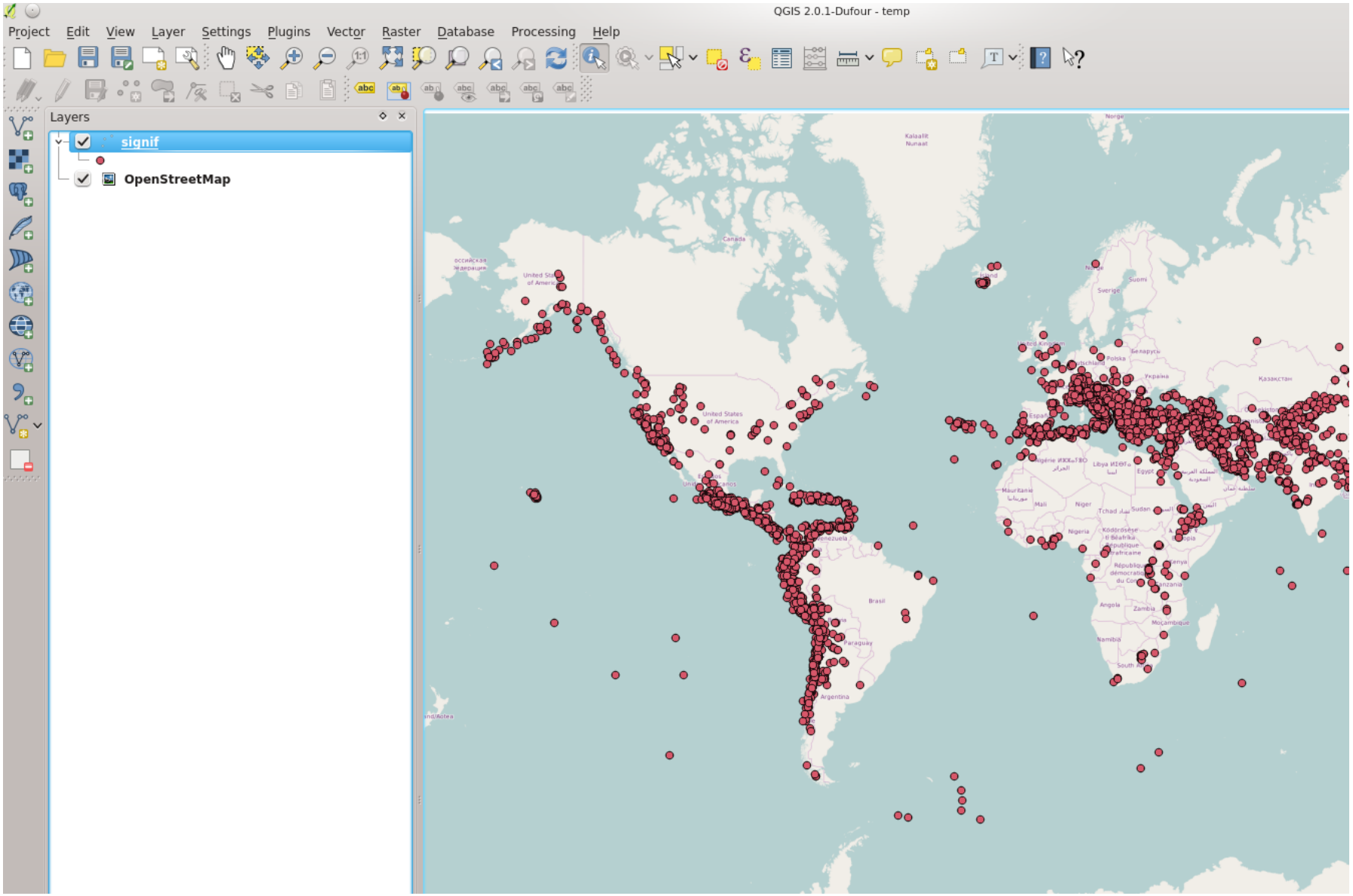
Plugin para mapas



Este processo altera a CRS (coordinate reference system) do projeto QGIS, e você pode encontrar problemas na exibição do mapa

Mudar para:
WGS 84 / Pseudo Mercartor (EPSG:3857)

Plugin para mapas



Mapas com R

- R também pode ser usado para visualização de dados espaciais

GGMAP

Usa uma imagem de um mapa **baixada da Internet**
Podemos exibir conteúdo adicional, como estatísticas, em cima do mapa

Instalação de pacotes exemplificados aqui:

```
install.packages("rworldmap”)
```

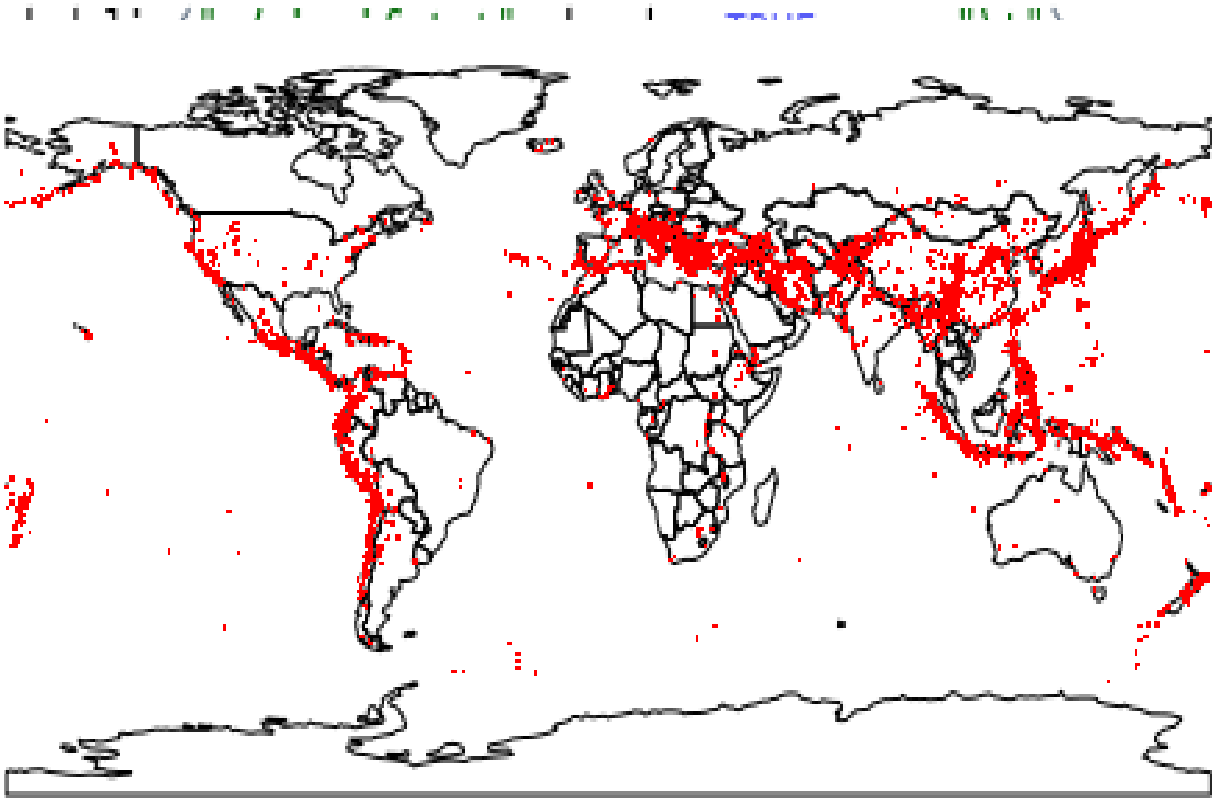
Visualizar pontos no mapa

install.packages("RgoogleMaps")

```
library(rworldmap)

dadosEarthquake <- read.delim("./signif.txt", header=TRUE, sep="\t")

newMap <- getMap(resolution = "less islands")
plot(newMap)

longitude <-as.numeric(unlist(dadosEarthquake["LONGITUDE"]))
latitude <- as.numeric(unlist(dadosEarthquake["LATITUDE"]))
points(longitude,latitude, col="red",pch = 15, cex = 0.2)
```

Visualizar pontos no mapa

```
library(rworldmap)

dadosEarthquake <- r

newMap <- getMap(res
plot(newMap)

longitude <-as.numer
latitude <- as.numer
points(longitude,lat
```



https://cran.r-project.org/web/packages/rworldmap/vignettes/rworldmap.pdf

Visualizar trajetórias

```r
library(ggmap)


df <- round(data.frame(
  x = c(-49.24997,-49.295,-49.2436,-49.26997,-49.28997,-49.24997),
  y = c(-25.43572,-25.46172,-25.48172,-25.49172,-25.45572,-25.41172)
), digits = 5)

map <- get_googlemap('curitiba', markers = df, path = df, scale = 1, zoom = 12)

ggmap(map)
```

Visualizar trajetórias

install.packages("ggmap")

```
library(ggmap)
```

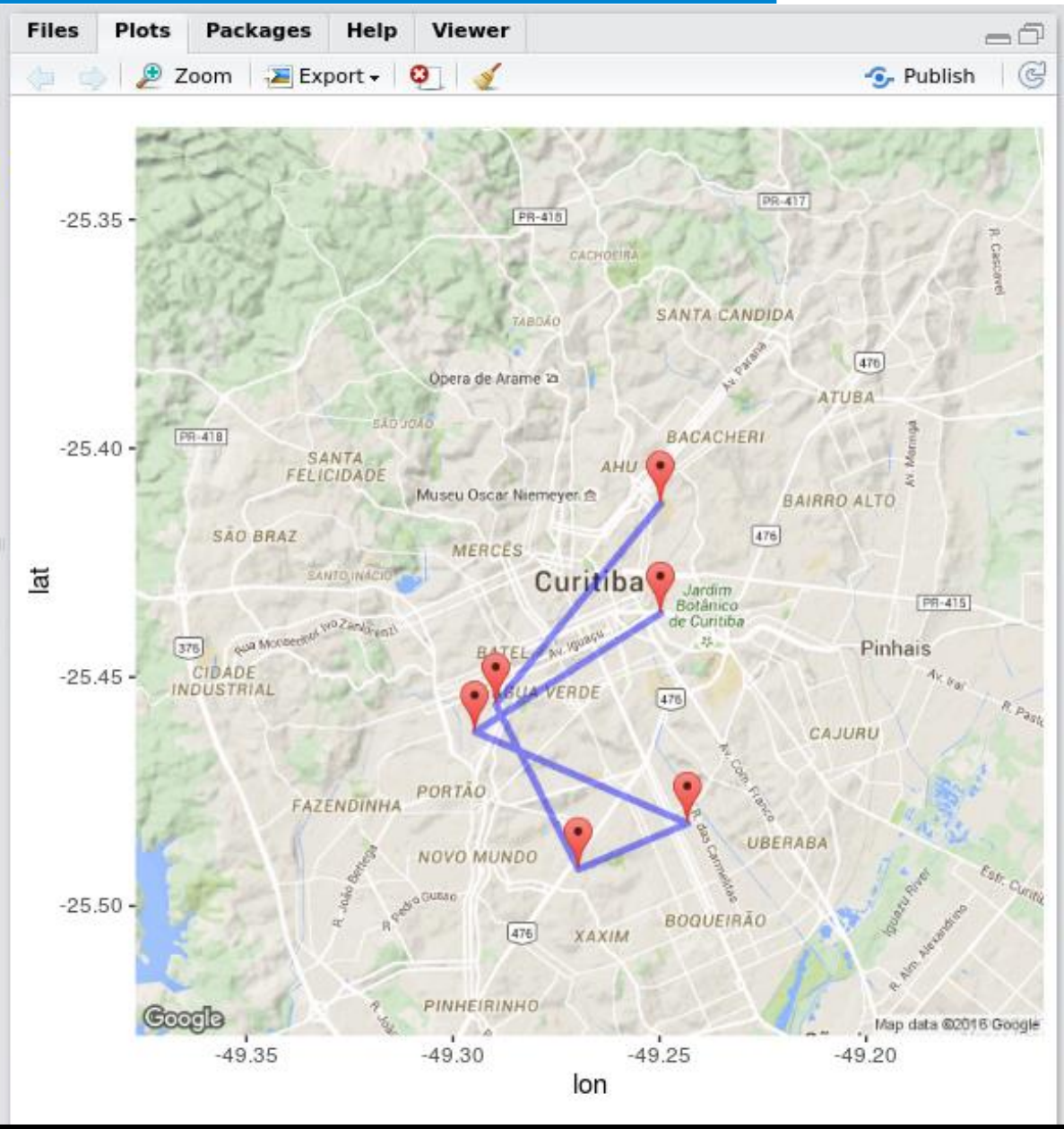
Conjunto de pontos para exibir

```
df <- round(data.frame(
  x = c(-49.24997,-49.295,-49.2436,-49.26997,-49.28997,-49.24997),
  y = c(-25.43572,-25.46172,-25.48172,-25.49172,-25.45572,-25.41172)
), digits = 5)

map <- get_googlemap('curitiba', markers = df, path = df, scale = 1, zoom = 12)

ggmap(map)
```

Recupera um mapa automaticamente

Visualizar trajetórias

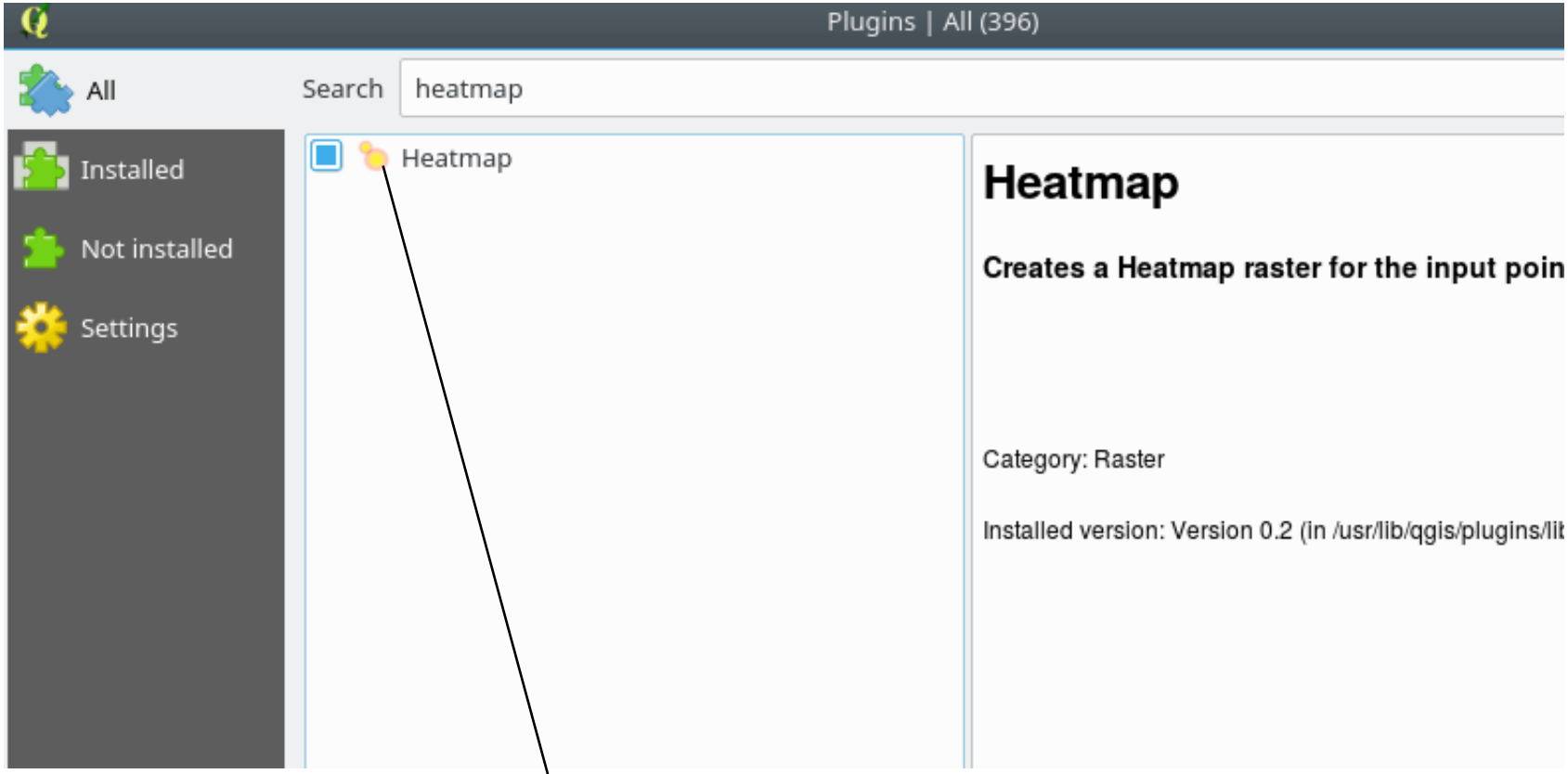
```
library(ggmap)

df <- round(data.frame(
  x = c(-49.24997,-49.29
  y = c(-25.43572,-25.46
), digits = 5)

map <- get_googlemap('CU

ggmap(map)
```



Mapas de calor (Heatmaps)

Heatmap com QGIS

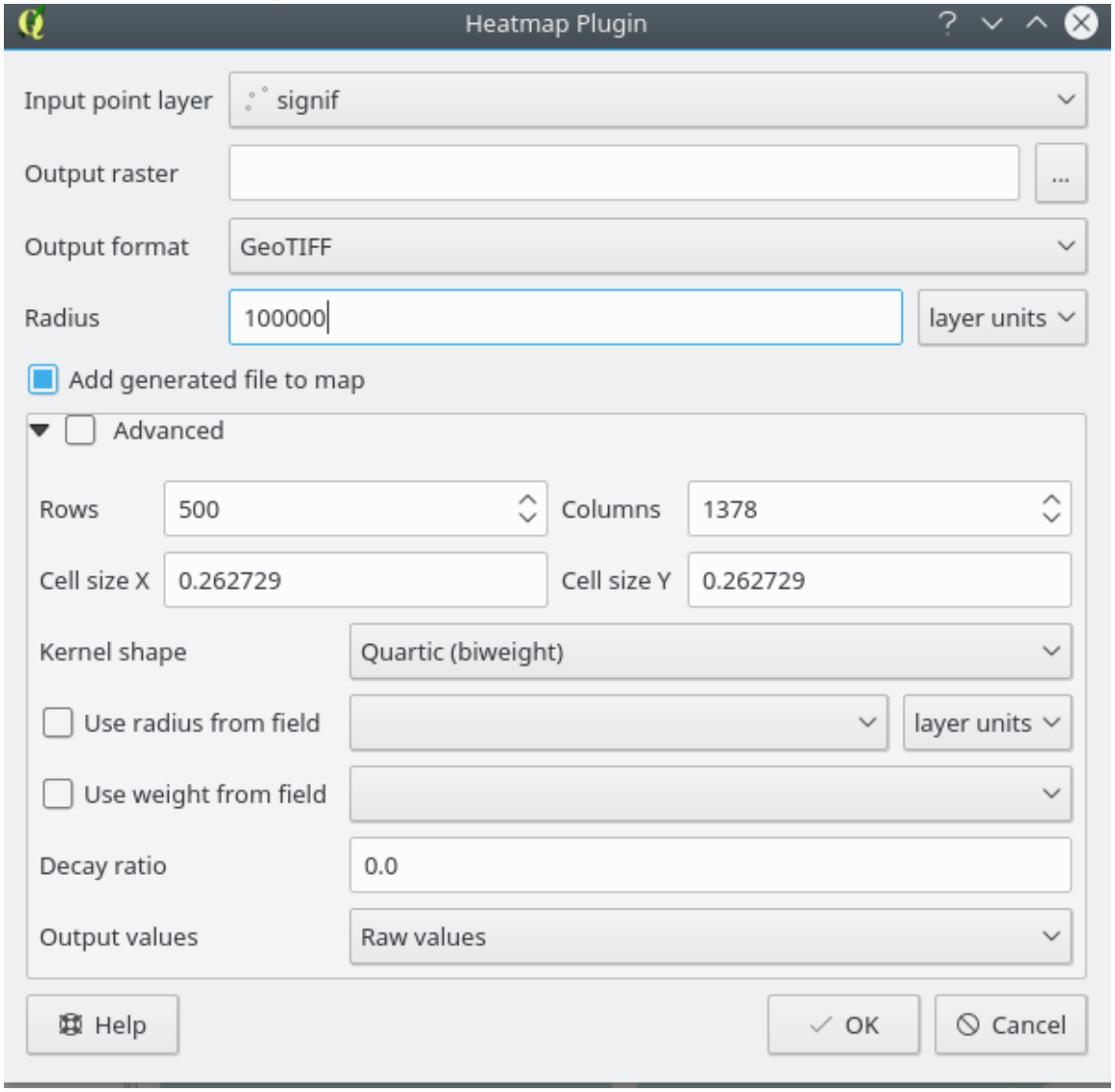
- Raster > heatmap



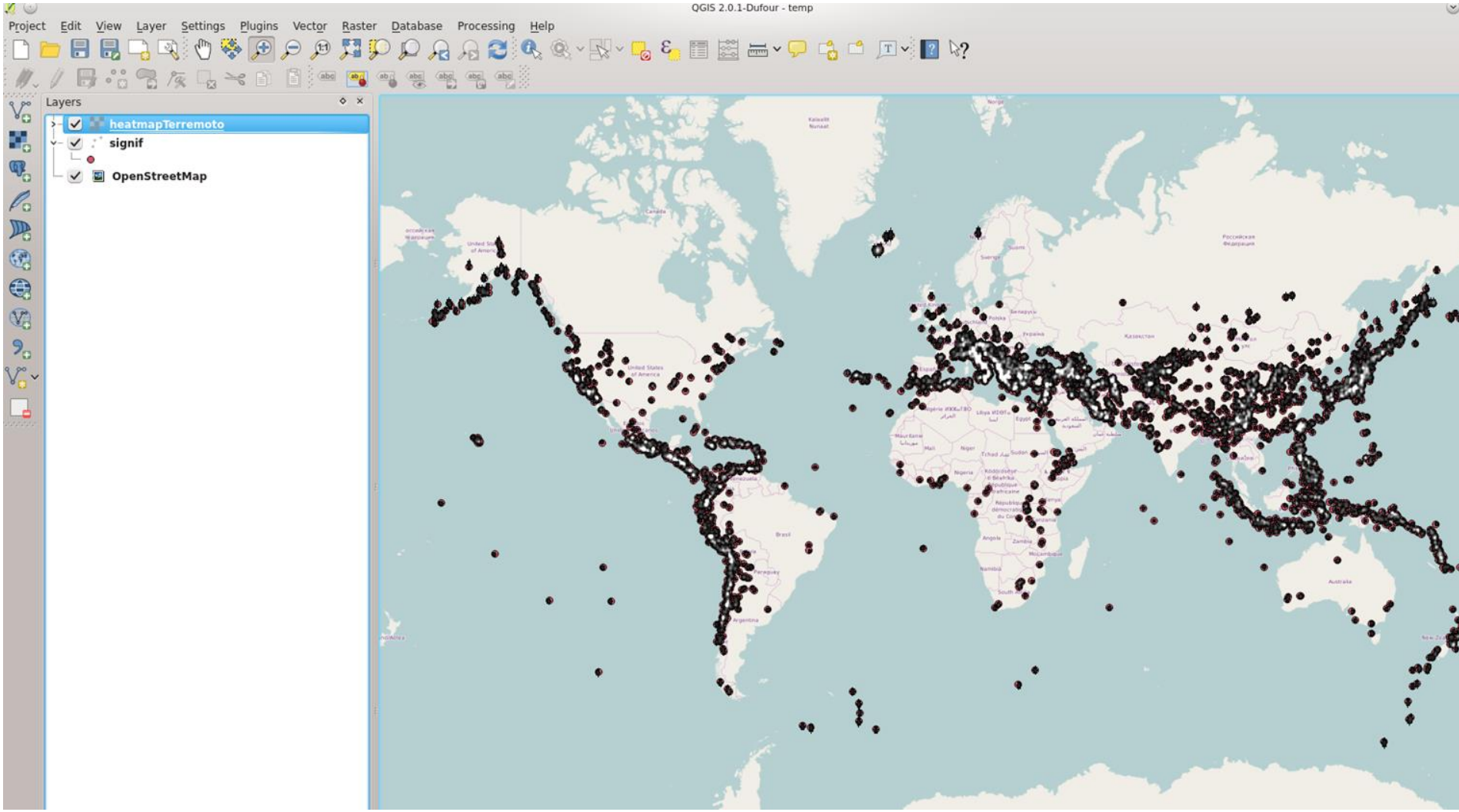
Verificar se está habilitado no menu “Manage Plugins”

Heatmap com QGIS

- Raster > heatmap

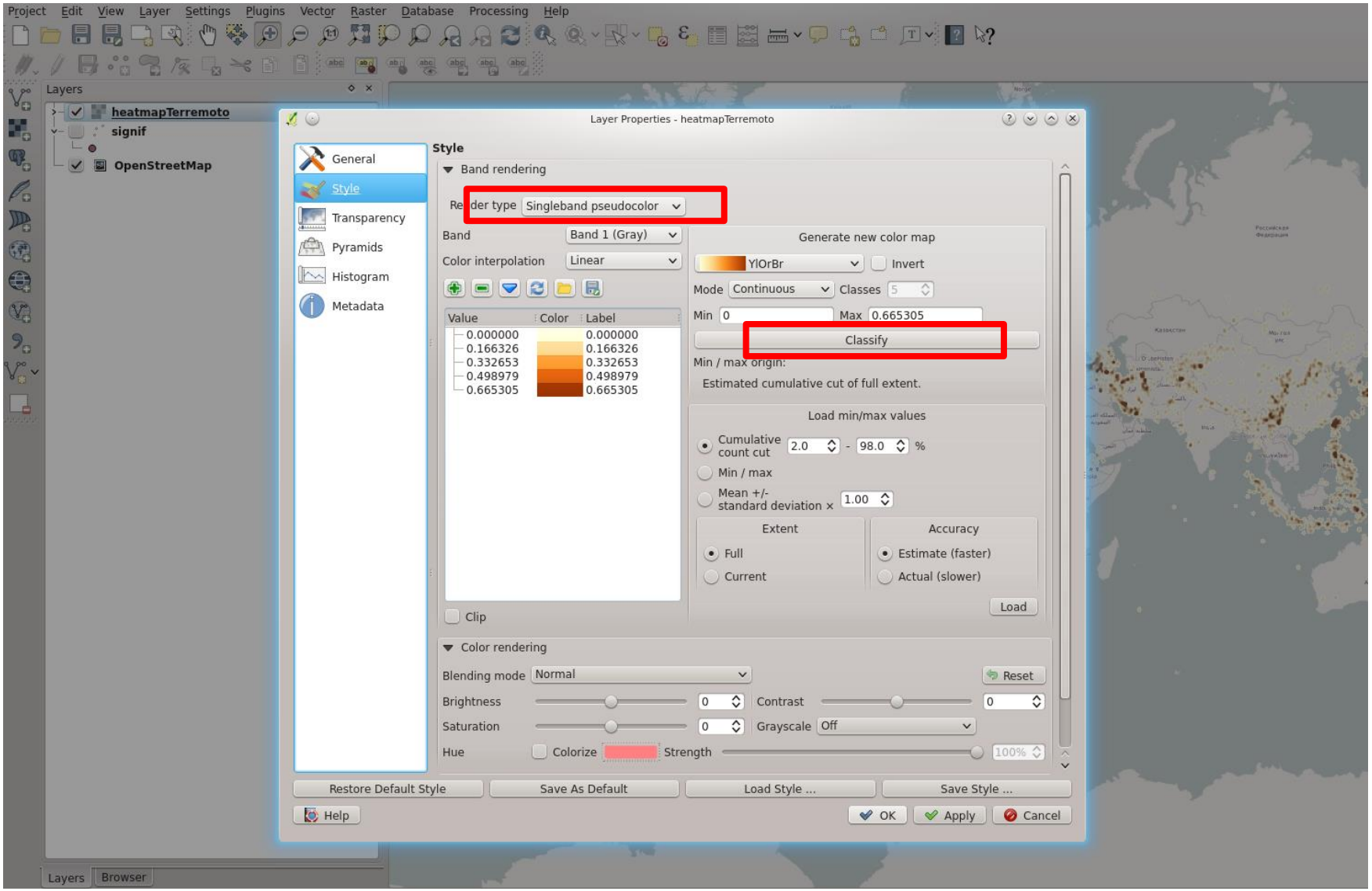


Heatmap com QGIS

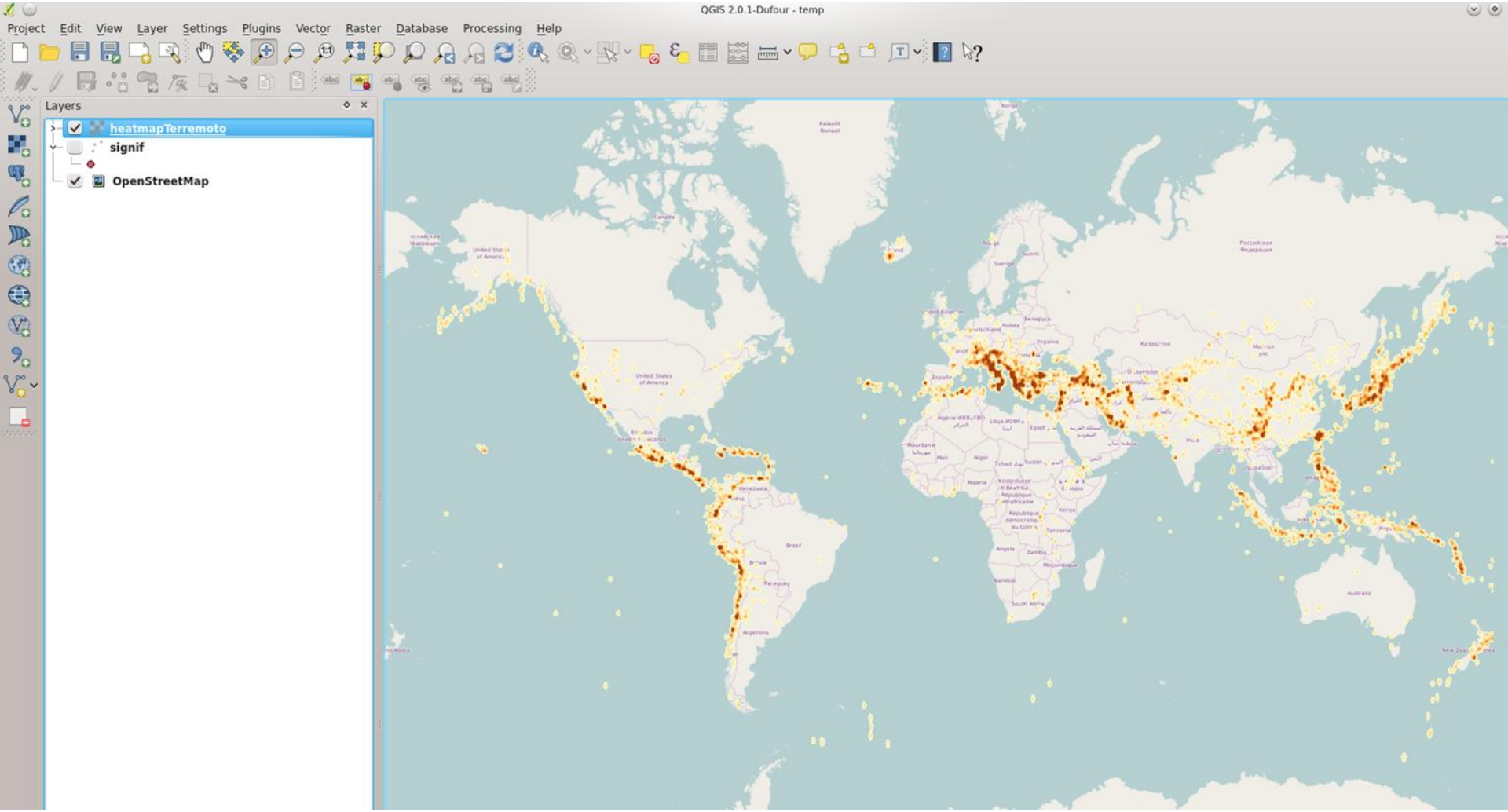


Heatmap com QGIS

- Clicar em propriedades da camada de heatmap



Heatmap com QGIS



Heatmap com R

```
library("ggmap")

checkins <- read.delim("BHcheckins.txt", header=FALSE, sep="\t")

names(checkins) <- c("LAT","LON")

map.in <- get_map(location ="Belo Horizonte", zoom=12,source = "osm")

ggmap(map.in)+
geom_bin2d(data=checkins, aes(x = LON, y = LAT), binwidth = 0.005,alpha = 0.4) +

scale_fill_gradient(low = "yellow", high = "red")
```

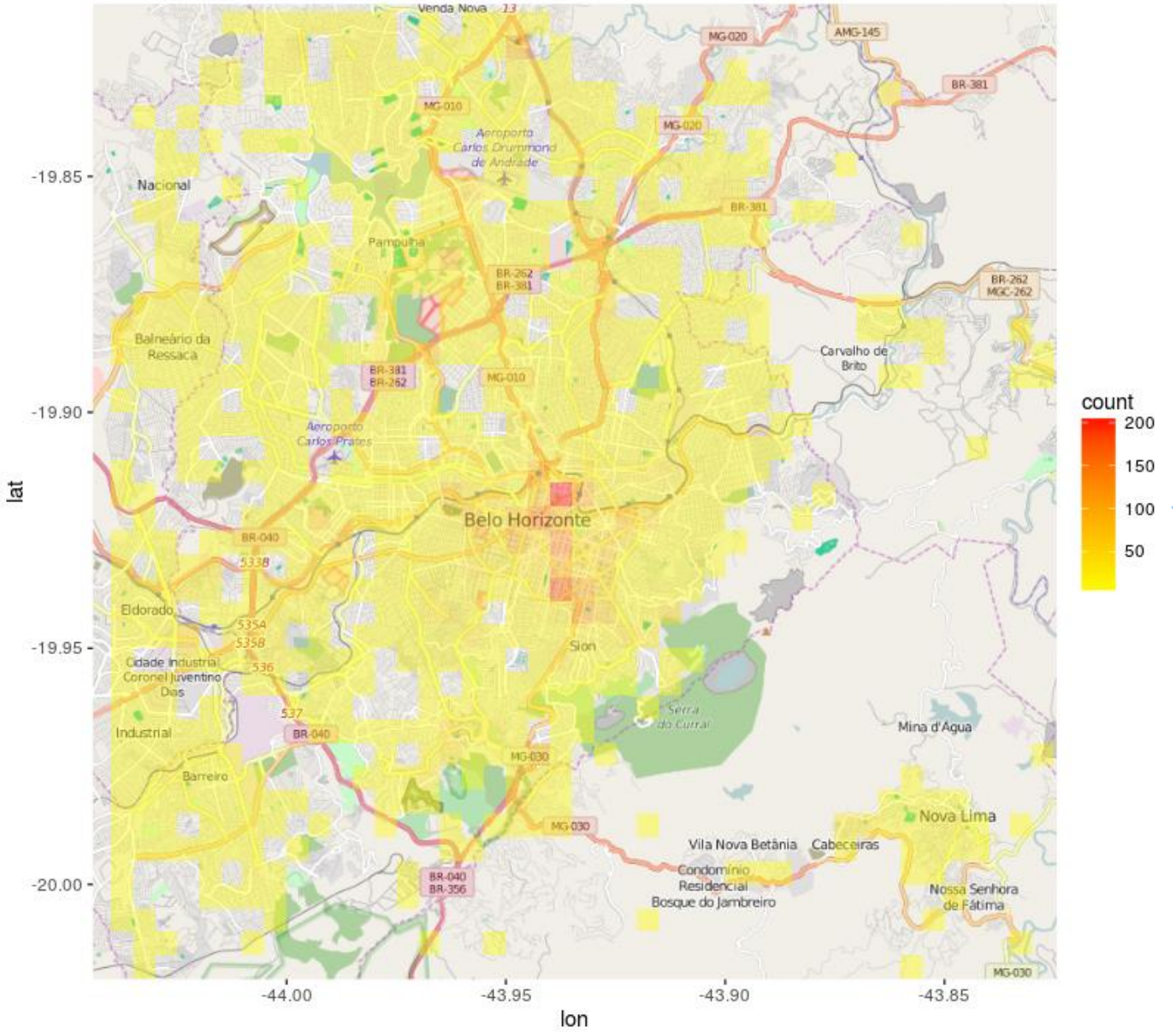
Heatmap com R

```
library("ggmap")

checkins <- read.delim("BHcheckins.txt", header=FALSE, sep="\t")

names(checkins) <- c("LAT","LON")

map.in <- get_map(location ="Belo Horizonte", zoom=12,source = "osm")

ggmap(map.in)+
geom_bin2d(data=checkins, aes(x = LON, y = LAT), binwidth = 0.005,alpha = 0.4) +

scale_fill_gradient(low = "yellow", high = "red")
```

Heatmap da contagem de *bins* 2D

Heatmap com R

```
library("ggmap")

checkins <- read.del

names(checkins) <- c

map.in <- get_map(lo

ggmap(map.in)+
geom_bin2d(data=chec
stat_bin2d(binwidth
scale_fill_gradient(
```



Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

Código disponível

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

Coordenadas

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

Formato demandado

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

Contagem de pontos em *bins* específicos

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

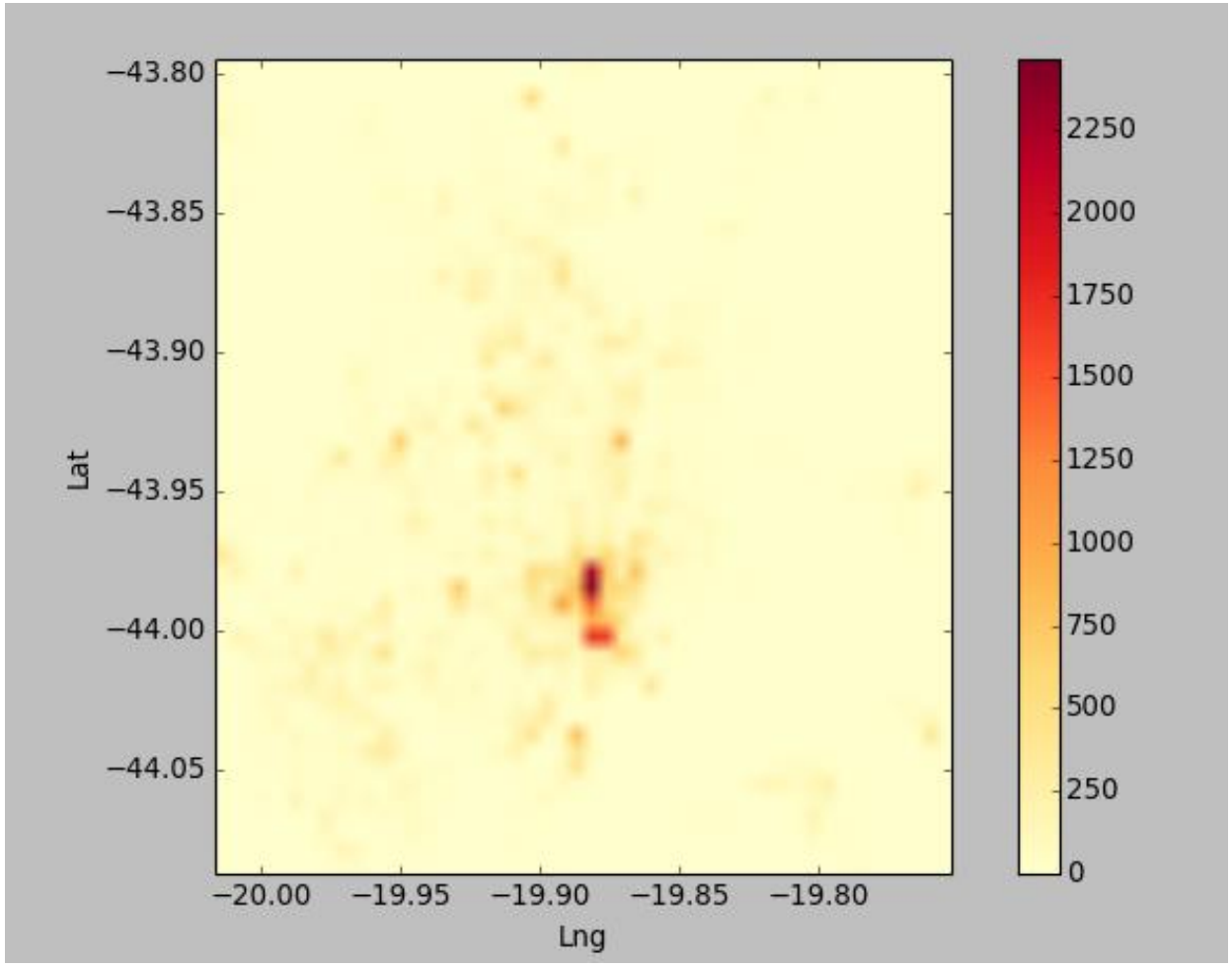
Máximo e mínimo da imagem

Heatmap com Python

```python
import numpy as np
import numpy.random
import matplotlib.pyplot as plt

listlat = []
listlng = []

f = open('../beloHorizonte.dat','r')

while True:
    line = f.readline()
    if len(line) ==0:
        break

    array = line.split()
    listlat.append(float(array[2]))
    listlng.append(float(array[3]))

x = np.array(listlat)
y = np.array(listlng)

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]
heatmap = heatmap[::-1,:] #espelhamento das coordenadas no eixo Y

plt.clf()

plt.imshow(heatmap, extent=extent,cmap='YlOrRd')
plt.colorbar()
plt.xlabel('Lng')
plt.ylabel('Lat')
plt.show()
```

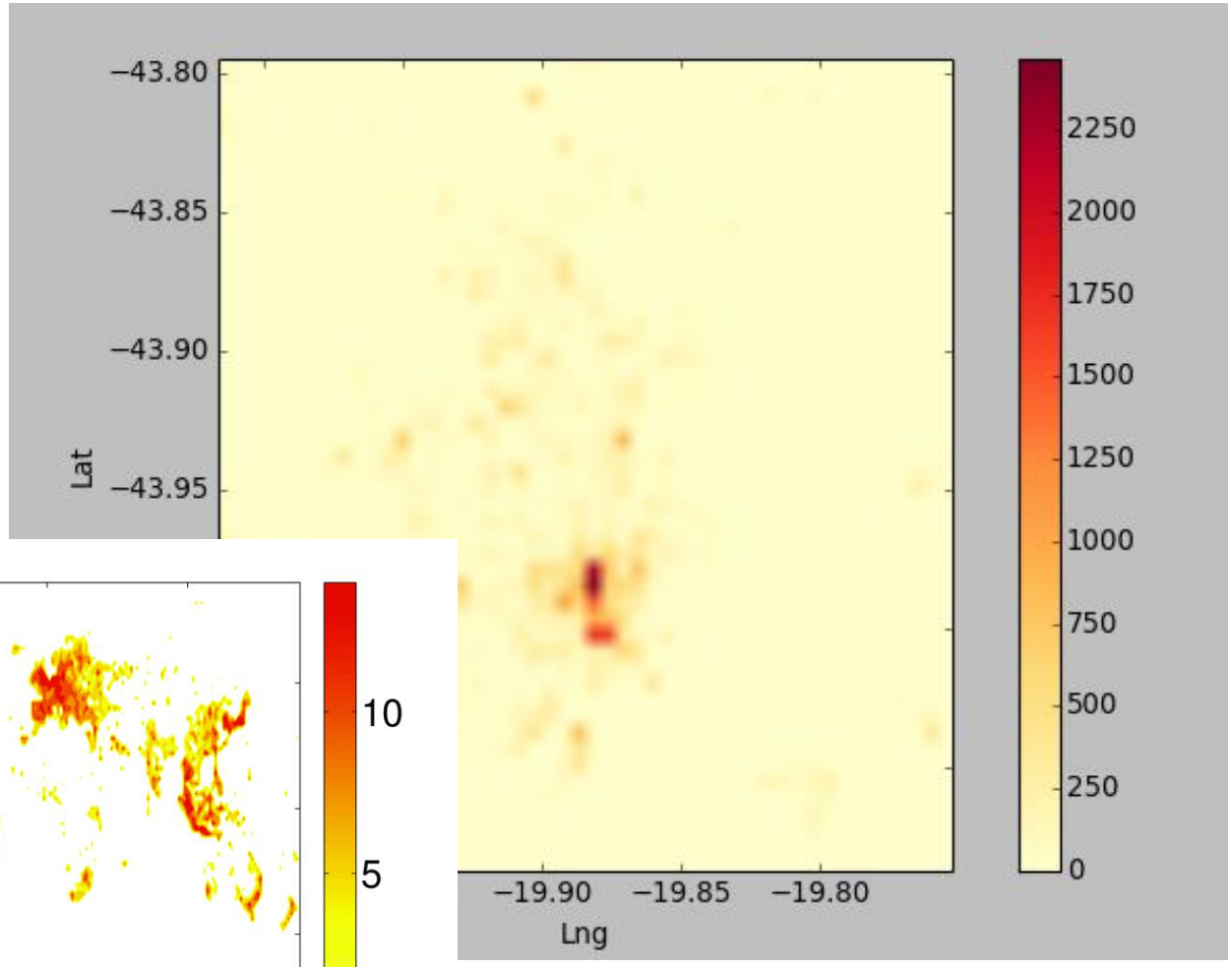
Estilo de cores

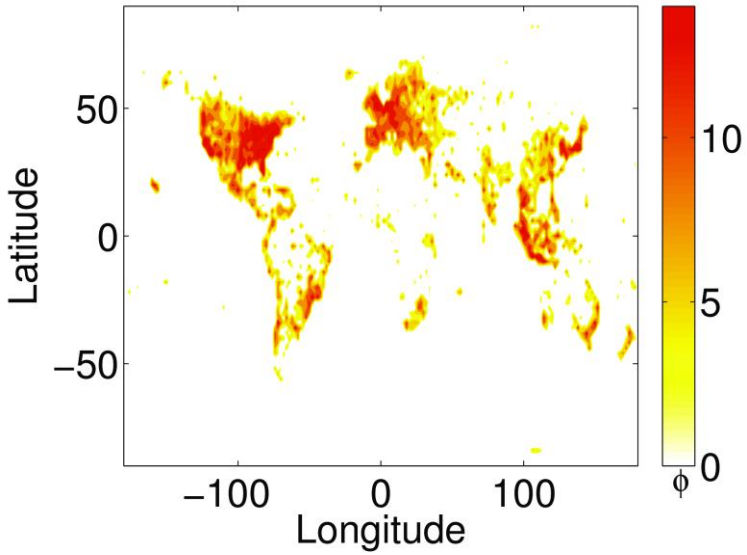
Heatmap com Python



Belo Horizonte

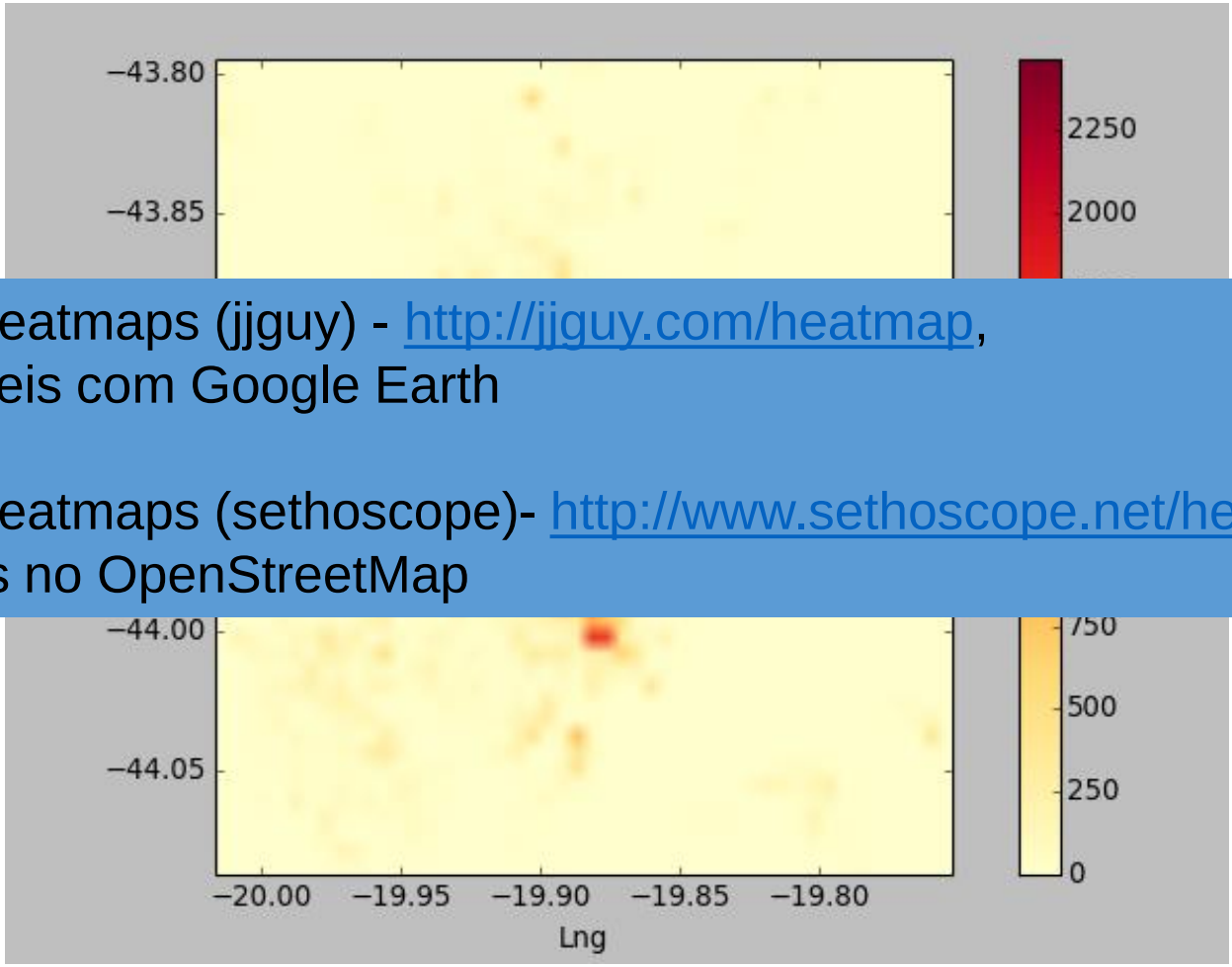
Heatmap com Python





Belo Horizonte

Heatmap com Python



Python Heatmaps (jjguy) - http://jjguy.com/heatmap, compatíveis com Google Earth

Python Heatmaps (sethoscope)- http://www.sethoscope.net/heatmap, baseados no OpenStreetMap

Qual utilizar?

Tabela 2.1. Comparando diferente alternativas para visualizar e manipular dados espaciais.

	GIS (por exemplo, QGIS)	Python	R
Instalação e Configuração do ambiente de trabalho	Fácil instalação. Interfaces gráficas relativamente intuitivas. Disponível para os principais sistemas operacionais (SOs).	Geralmente possui dependências de pacotes na instalação. Independente de SO.	Depende da instalação de pacotes, processo que é geralmente fácil. Independente de SO.
Análise de dados	Bom para iniciantes.	Tende a ser a plataforma estatística preferida para cientistas com elevado conhecimento de programação. Oferece flexibilidade.	Tende a ser a plataforma estatística preferida para cientistas de áreas diversas que usam R. A curva de aprendizado é a mais baixa. Oferece flexibilidade.
Criação de um mapa	Opção mais fácil.	Requer vontade de aprender fundamentos de programação.	Bom para cientistas que já usam R.

Qual utilizar?

Tabela 2.1. Comparando diferente alternativas para visualizar e manipular dados espaciais.

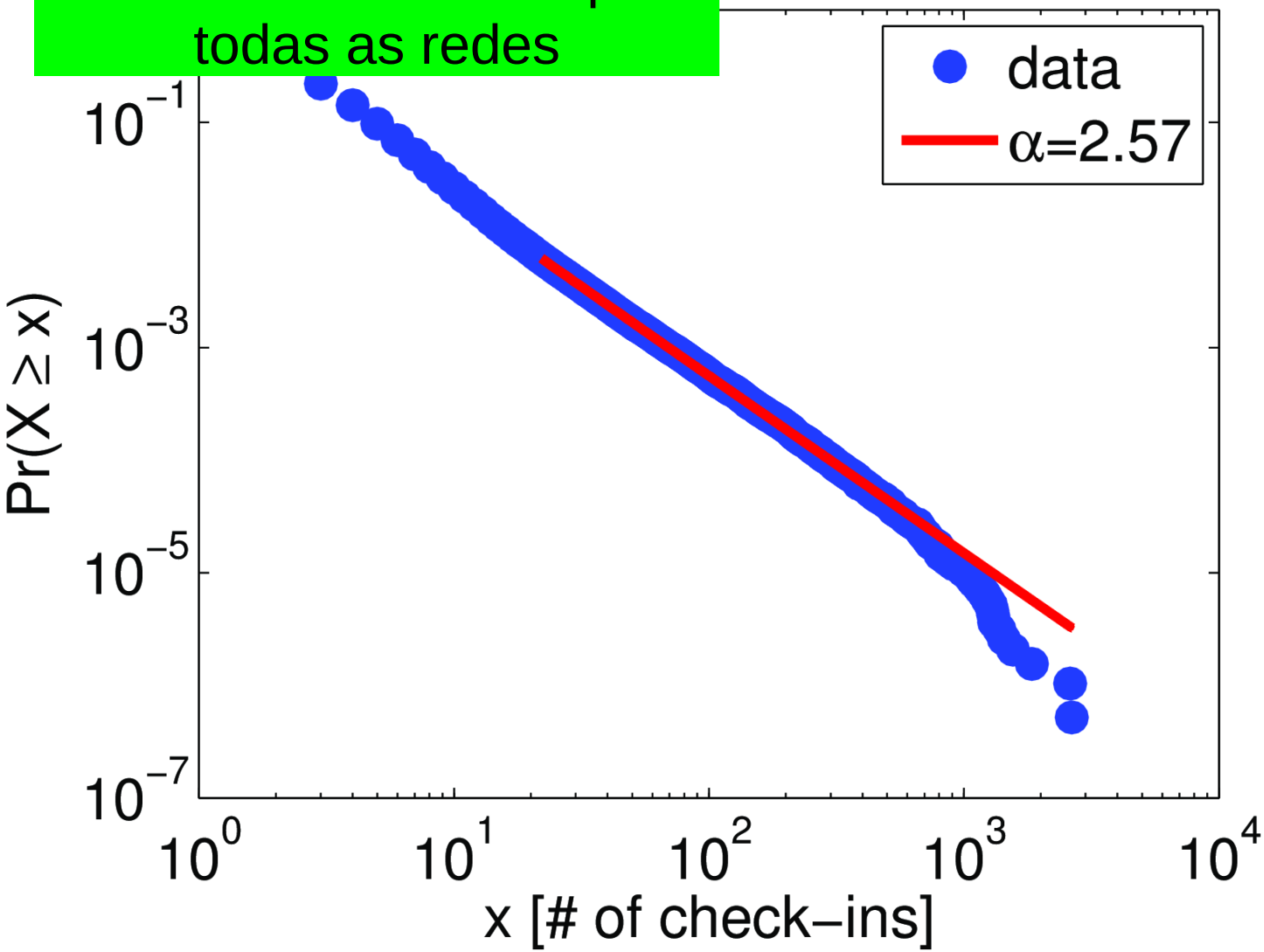
	GIS (por exemplo, QGIS)	Python	R
Instalação e Configur do ambi trabalho	Fácil instalação. ráficas intuiti- vas. Disponível para os principais sistemas operacionais (SOs).	Geralmente possui dependê cotes na Independente de SO.	Depende da instala- s, pro- geral- mente fácil. Independente de SO.
Análise de dados	Bom para iniciantes.	Tende a ser a plataforma estatística preferida para cientistas com elevado conhecimento de programação. Oferece flexibilidade.	Tende a ser a plataforma estatística preferida para cientistas de áreas diversas que usam R. A curva de aprendizado é a mais baixa. Oferece flexibilidade.
Criação de um mapa	Opção mais fácil.	Requer vontade de aprender fundamentos de programação.	Bom para cientistas que já usam R.

Facilidade

Flexibilidade

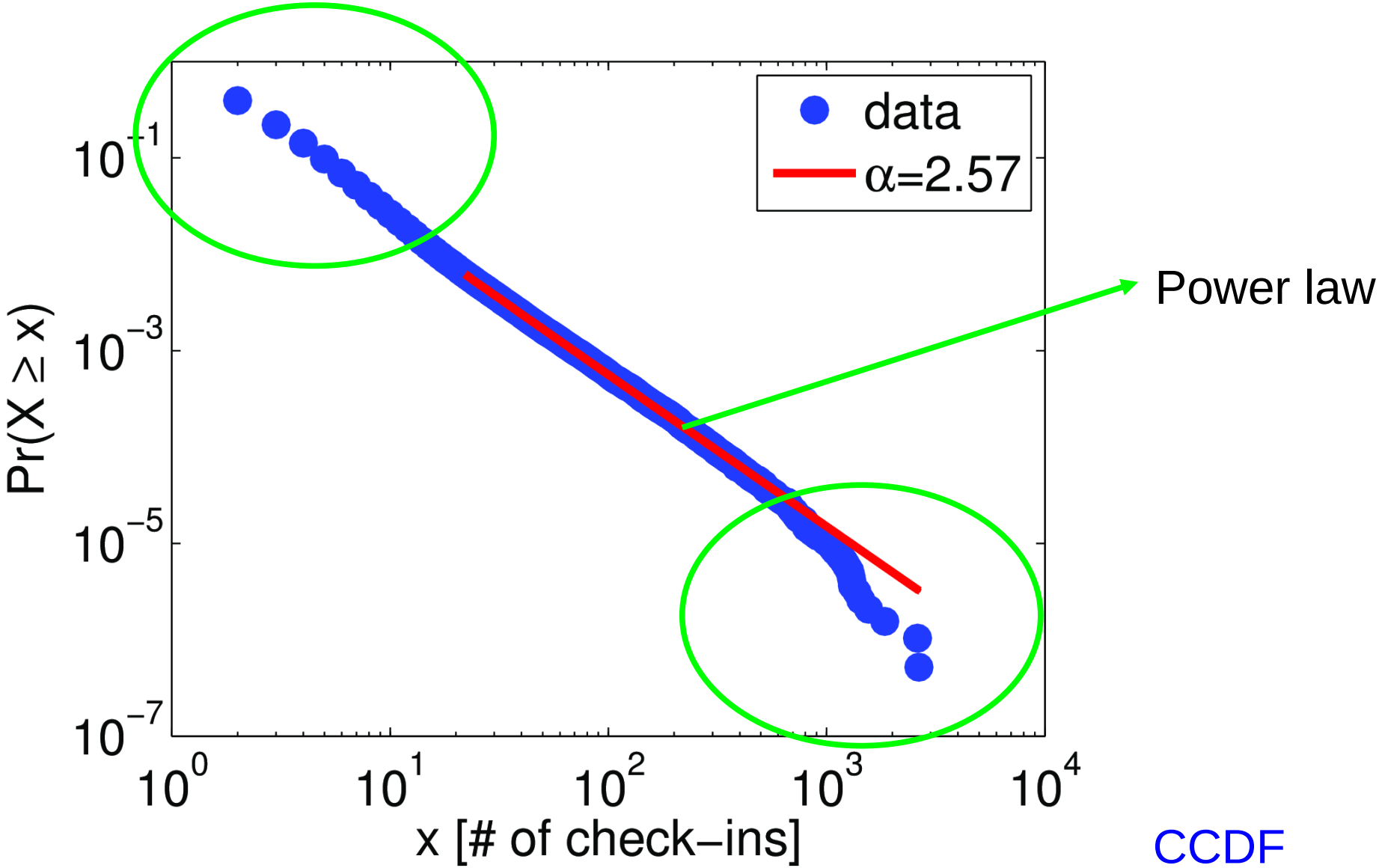
Cobertura em áreas

Resultados similares para todas as redes



CCDF

Cobertura em áreas



Detalhes técnicos

Verificar a existência de lei de potência

Teoricamente interessante devido a ser "cauda pesada”

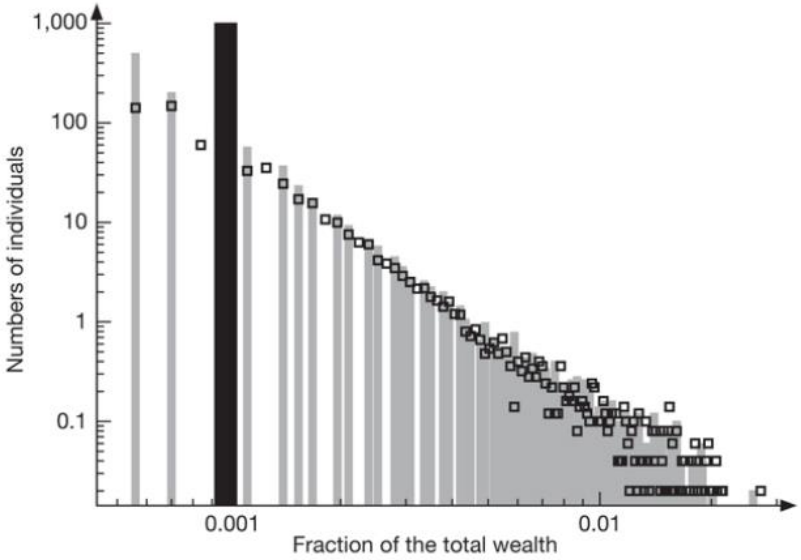
Detalhes técnicos

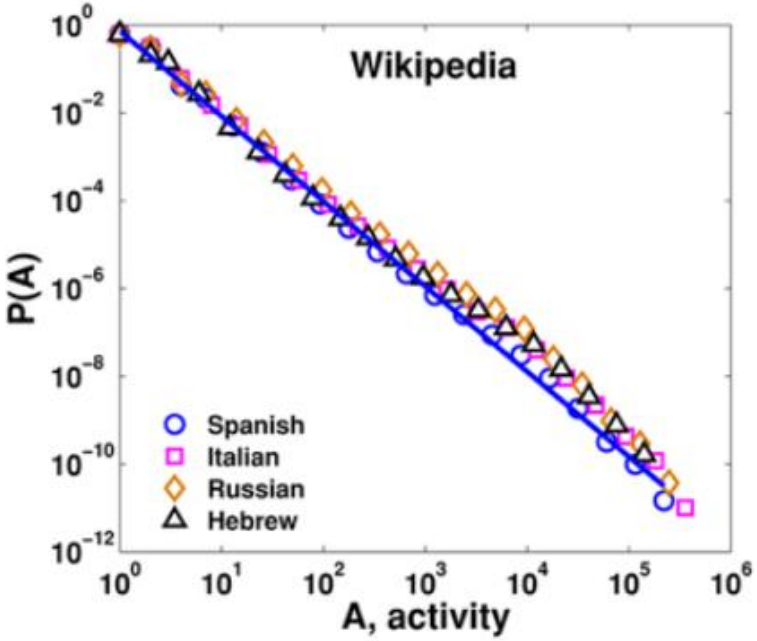
Verificar a existência de lei de potência

$$p(x) \propto x^{-\alpha}$$

Teoricamente interessante devido a ser "cauda pesada"

Estão em vários fenômenos





Detalhes técnicos

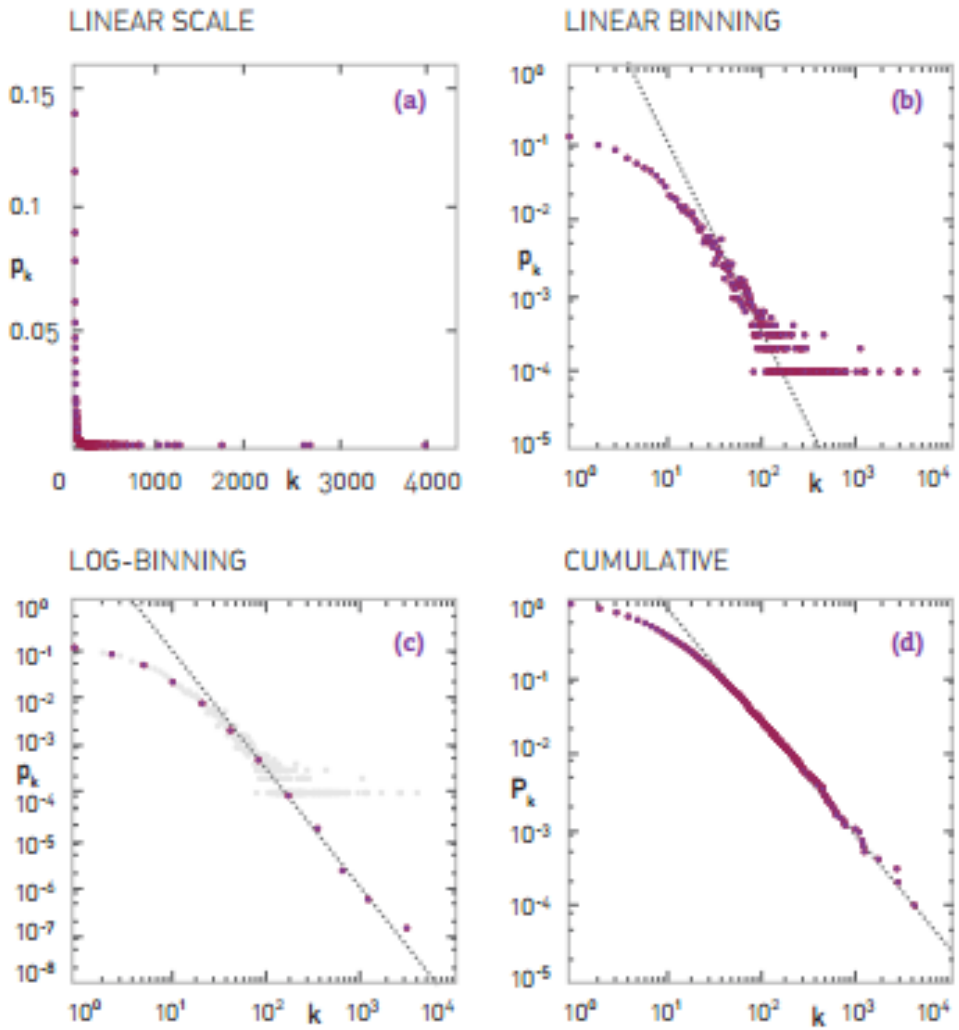
Verificar a existência de lei de potência

$$p(x) \propto x^{-\alpha}$$

Teoricamente interessante devido a ser "cauda pesada”

Fazer o *fitting* correto não é trivial

Detalhes técnicos



$$2 < \alpha < 3$$

Use a Log-Log Plot

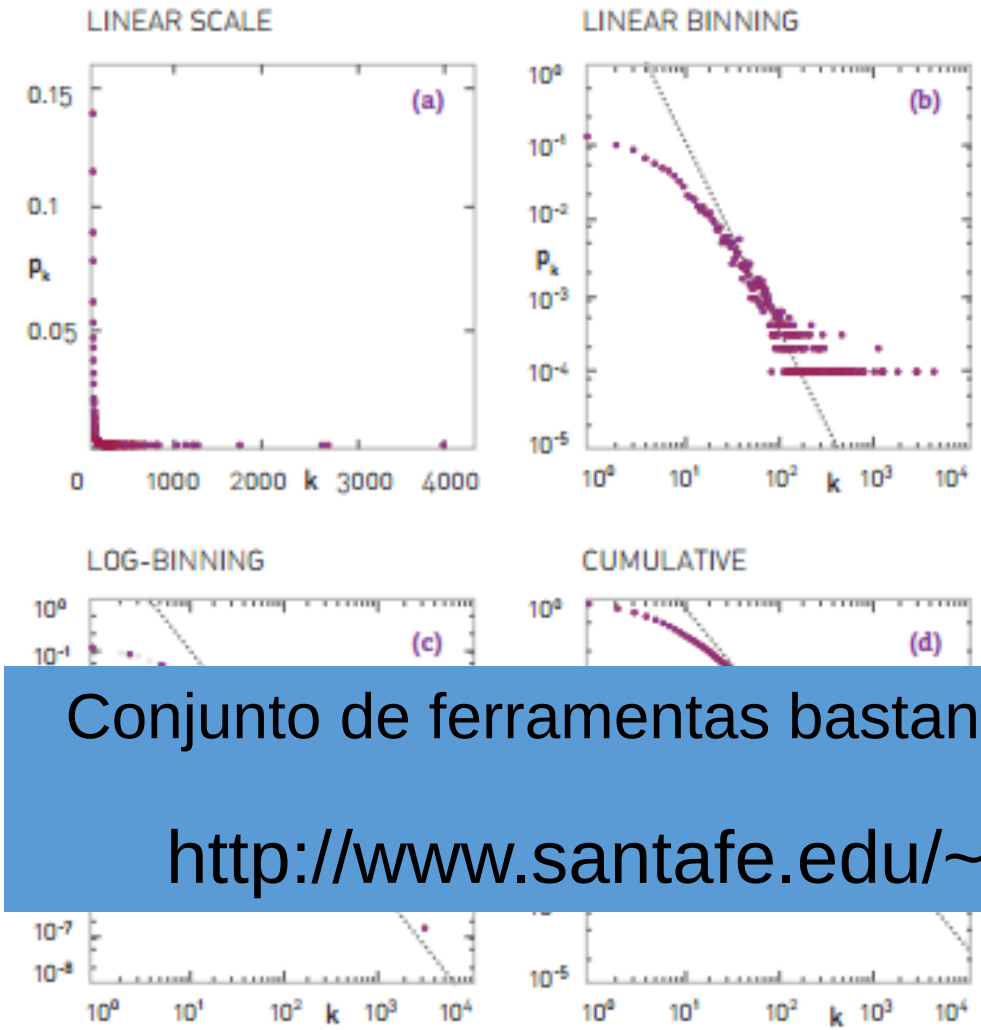
Avoid Linear Binning

Use Logarithmic Binning

Use Cumulative Distribution

Imagem de Barabasi

Detalhes técnicos



$$2 < \alpha < 3$$

Use a Log-Log Plot

Avoid Linear Binning

Use Logarithmic Binning

Conjunto de ferramentas bastante útil disponibilizado em:

http://www.santafe.edu/~aaronc/powerlaws/

Imagem de Barabasi

Detalhes técnicos

Fitting a power-law distribution

This function implements both the discrete and continuous maximum likelihood estimators for fitting the power-law distribution to data, along with the goodness-of-fit based approach to estimating the lower cutoff for the scaling region. Usage information is included in the file; type 'help plfit' at the Matlab prompt for more information.

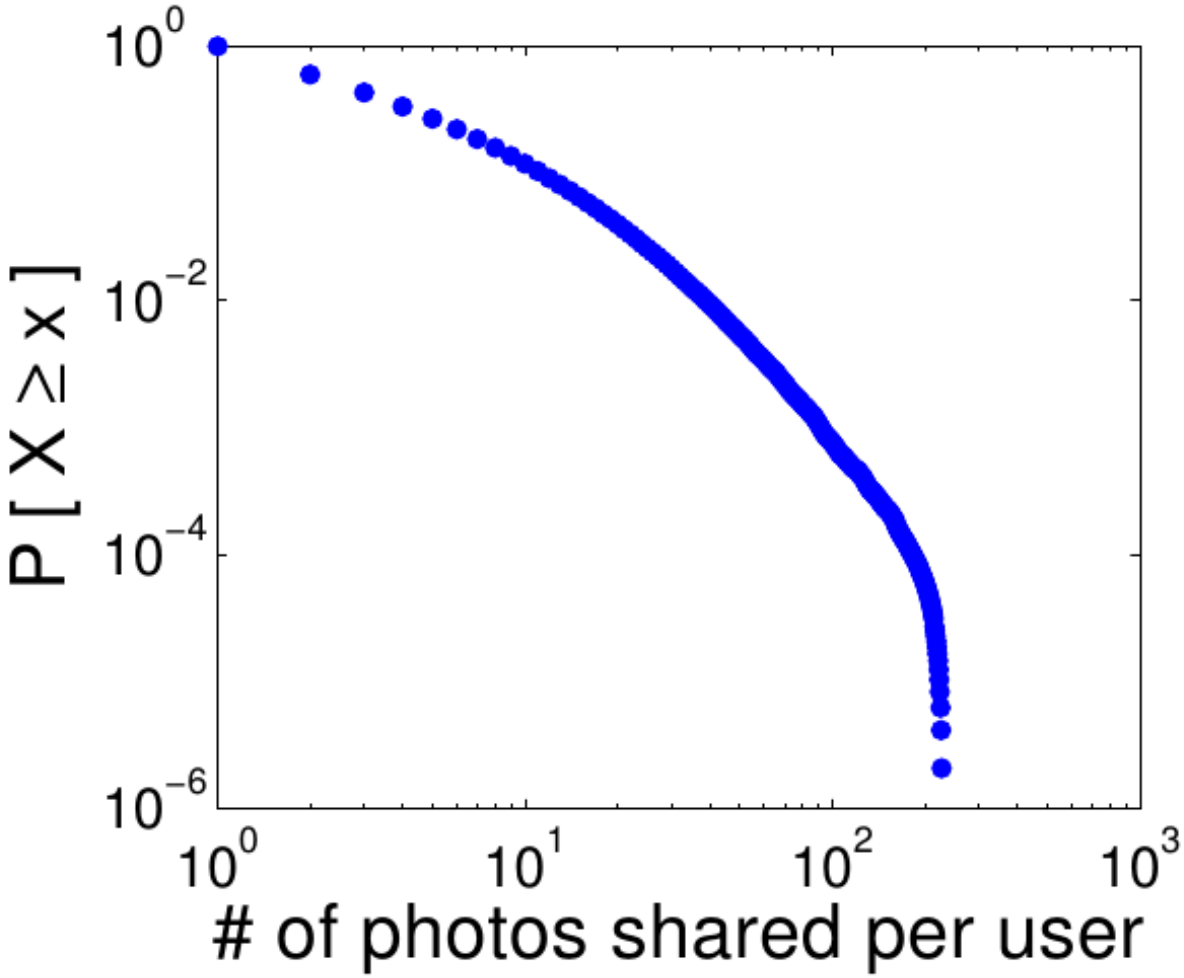
plfit.m (Matlab, by Aaron Clauset)
plfit.r (R, by Laurent Dubroca)
plfit.py (Python, by **Adam Ginsburg**)
plfit.c (C++, by Wim Otte; includes plvar.c)
plfit.c (C++, by Tamas Nepusz)
plfit.py (Python, by Joel Ornstein)

Visualizing the fitted distribution

After several requests, I've written this function, which plots (on log-log axes) the empirical distribution along with the fitted power-law distribution. Usage information is included in the file; type 'help plplot' at the Matlab prompt for more information.

plplot.m (Matlab, by Aaron Clauset)
plplot.py (Python, by Joel Ornstein)

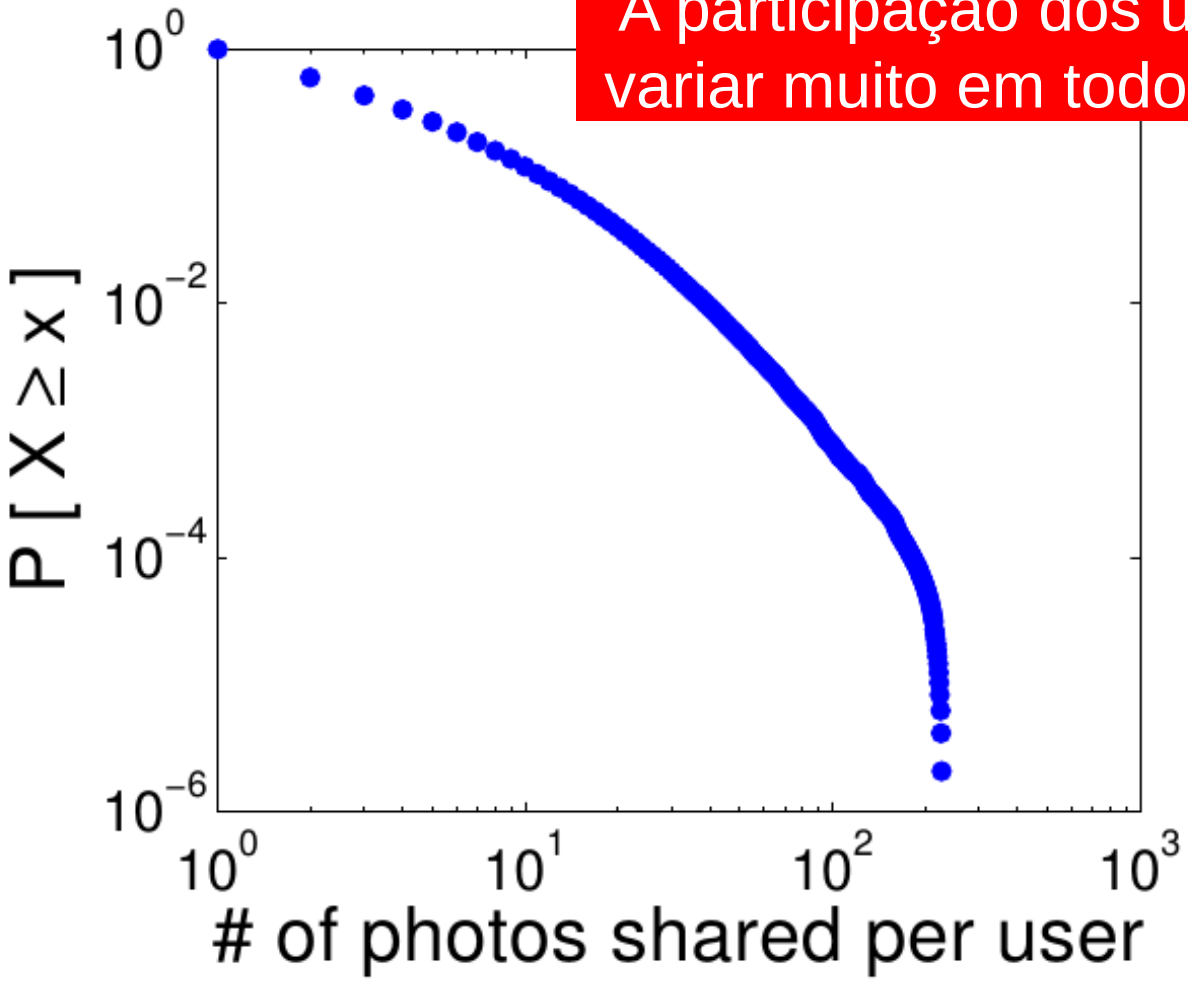
Comportamento dos nós



A picture of Instagram is Worth More than a Thousand Words: Workload Characterization and Application. DCOSS'13

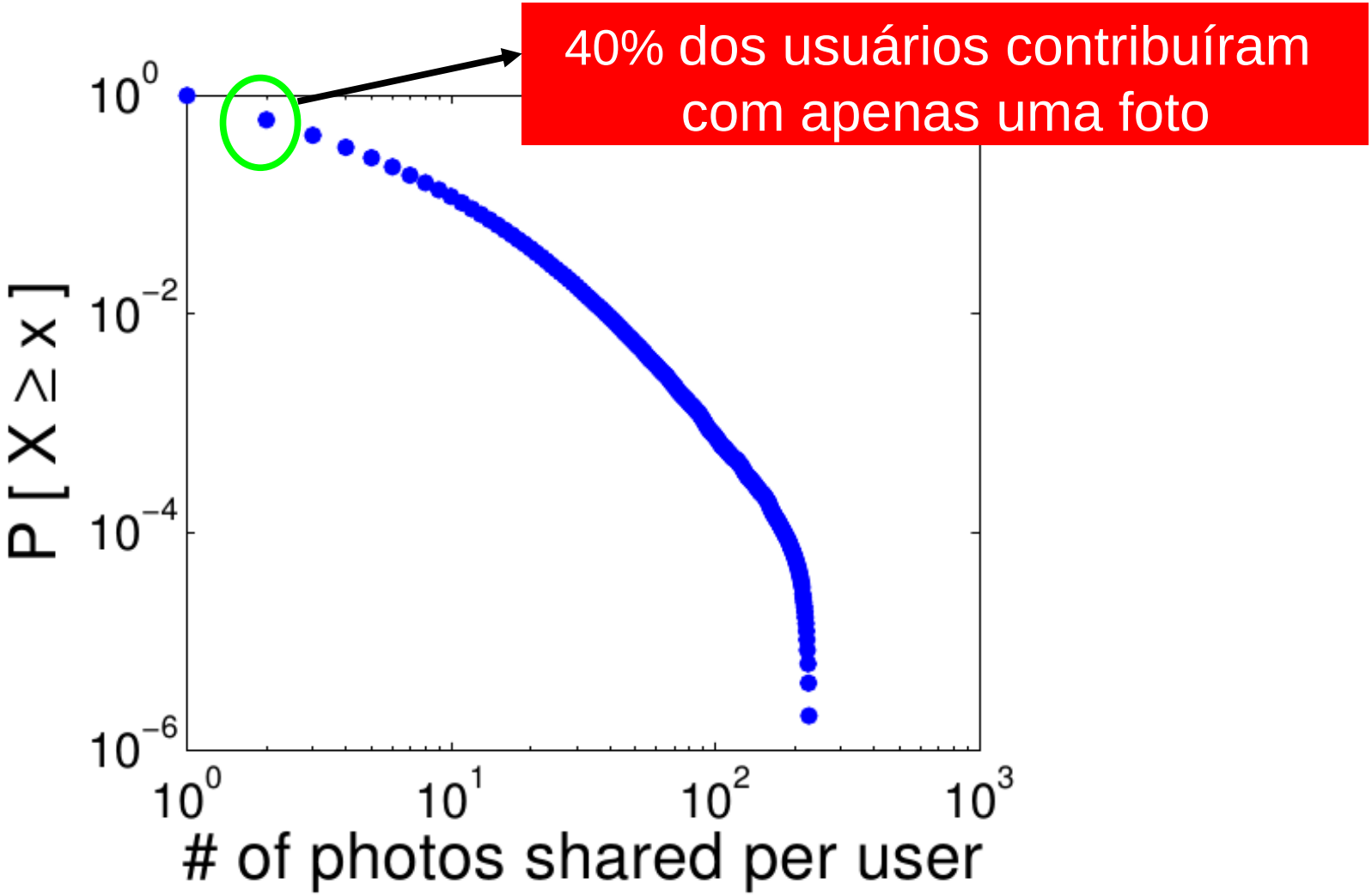
Comportamento dos nós

A participação dos usuários pode variar muito em todos os sistemas

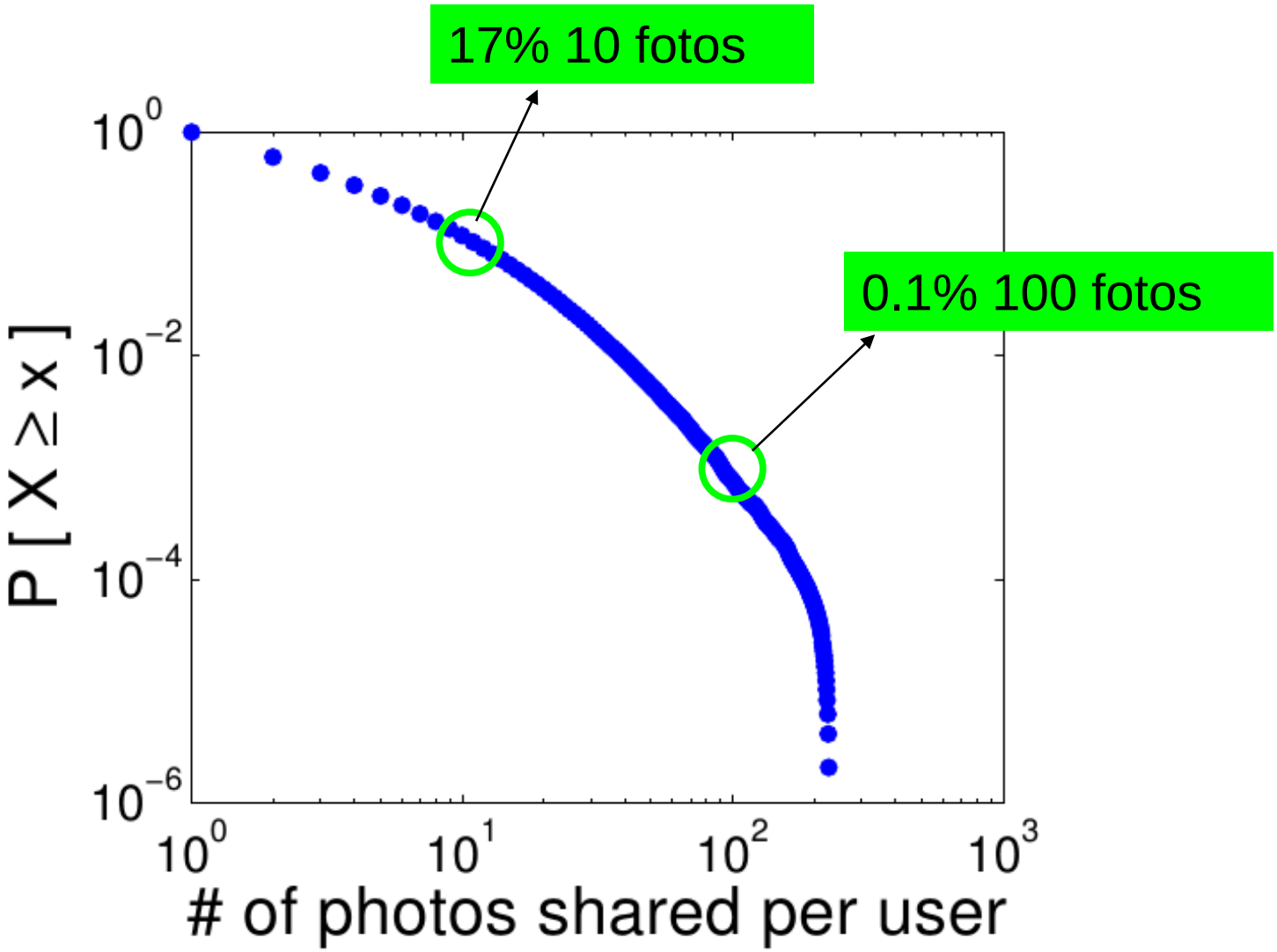


A picture of Instagram is Worth More than a Thousand Words: Workload Characterization and Application. DCOSS'13

Comportamento dos nós



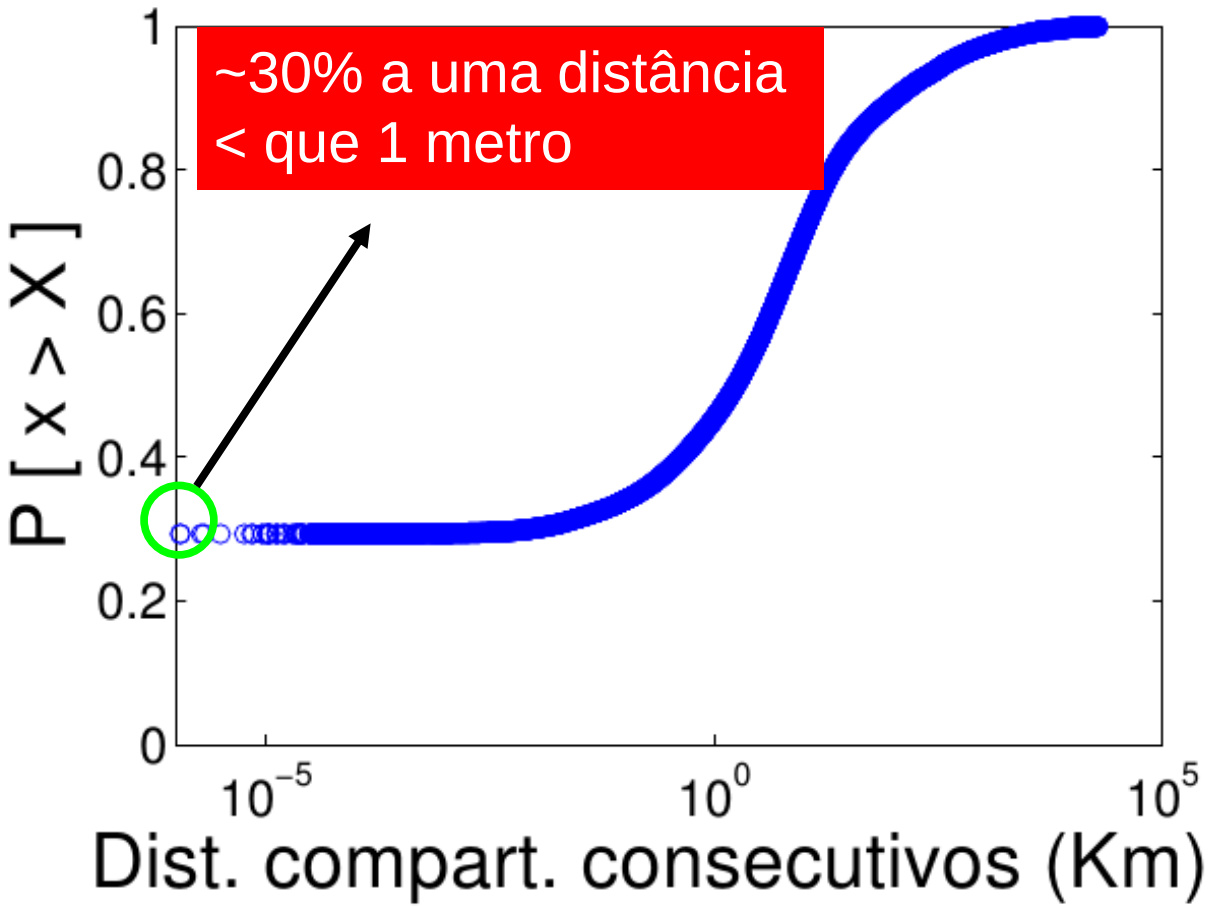
Comportamento dos nós



A picture of Instagram is Worth More than a Thousand Words: Workload Characterization and Application. DCOSS'13

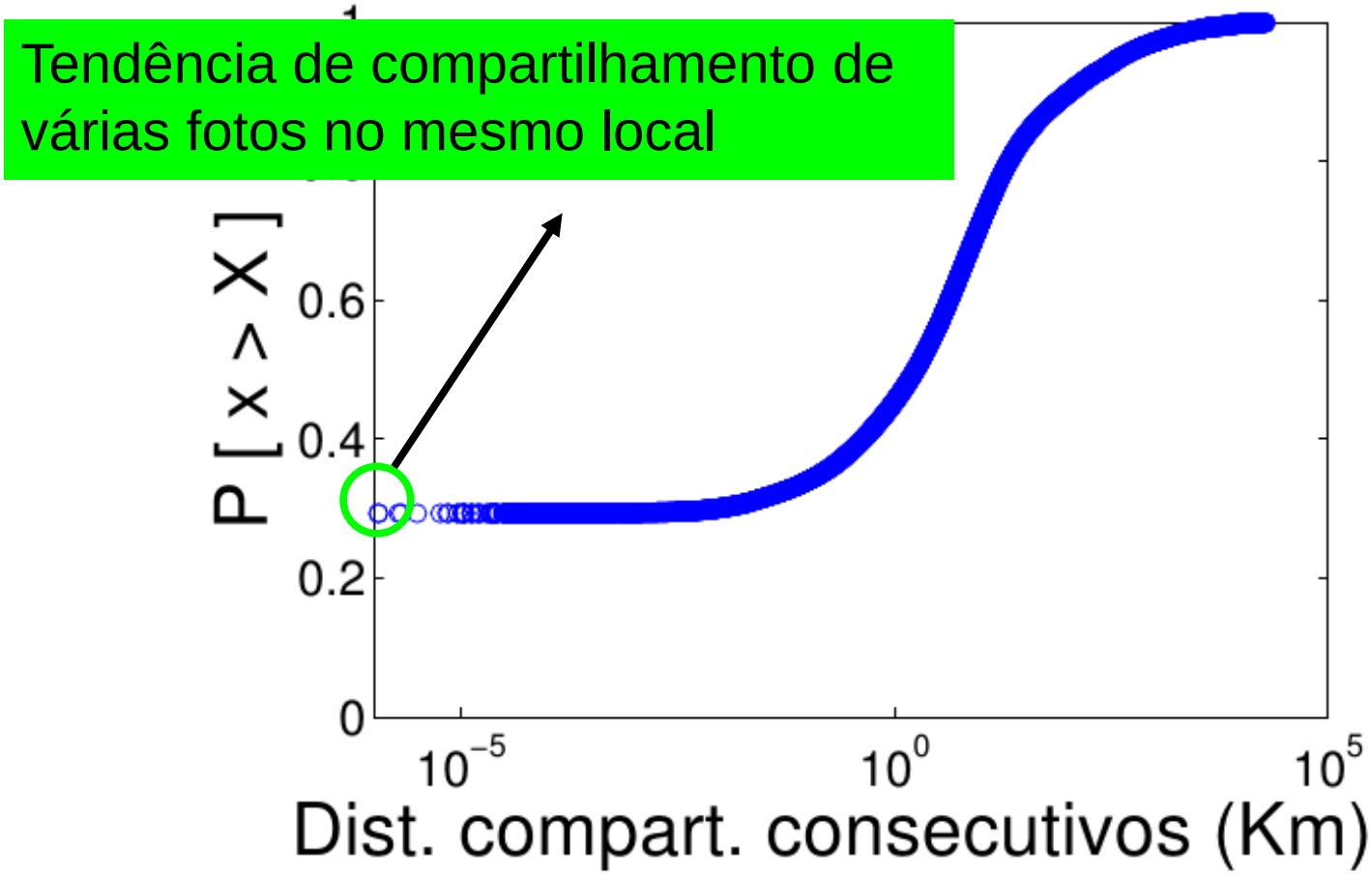
Distância entre check-ins

Distância geográfica entre fotos consecutivas de um mesmo usuário.



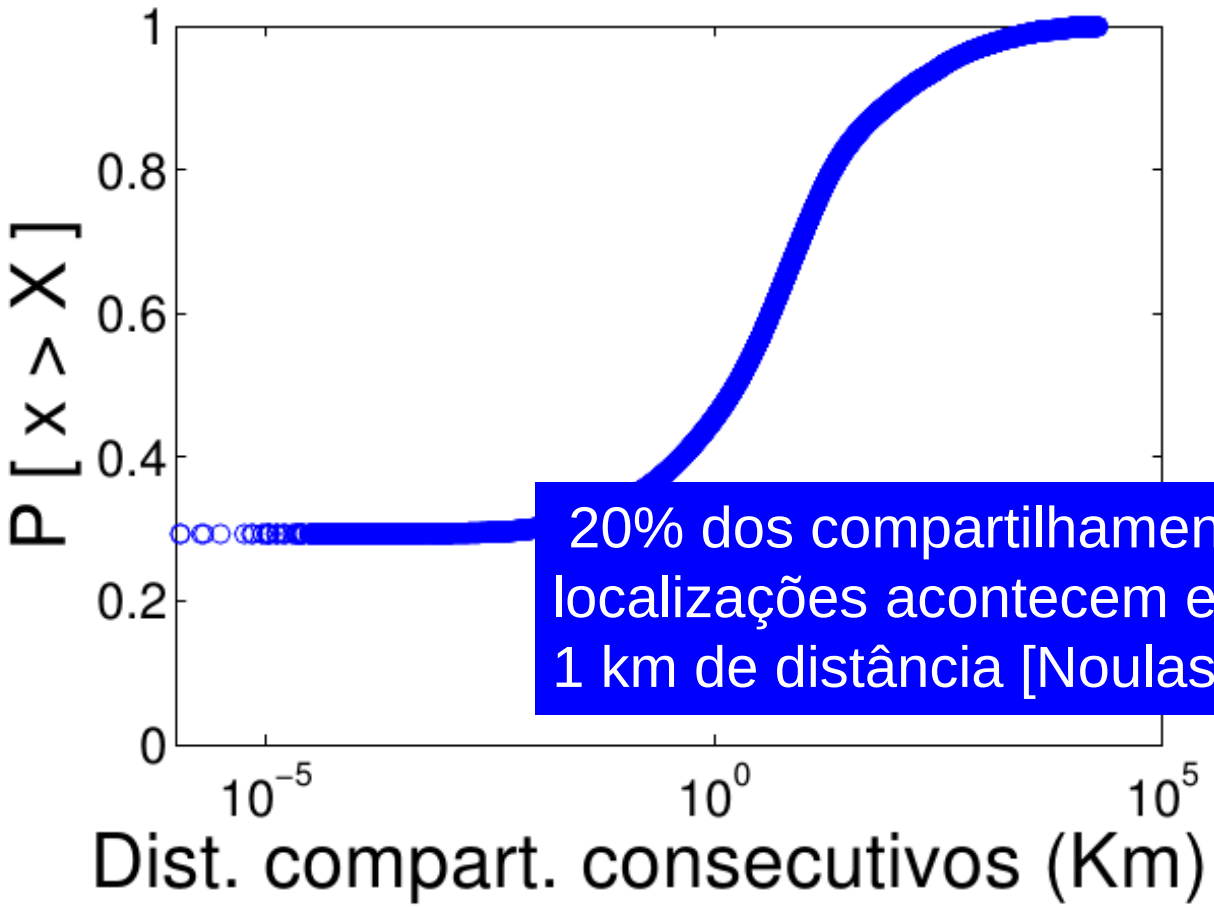
Distância entre check-ins

Distância geográfica entre fotos consecutivas de um mesmo usuário.



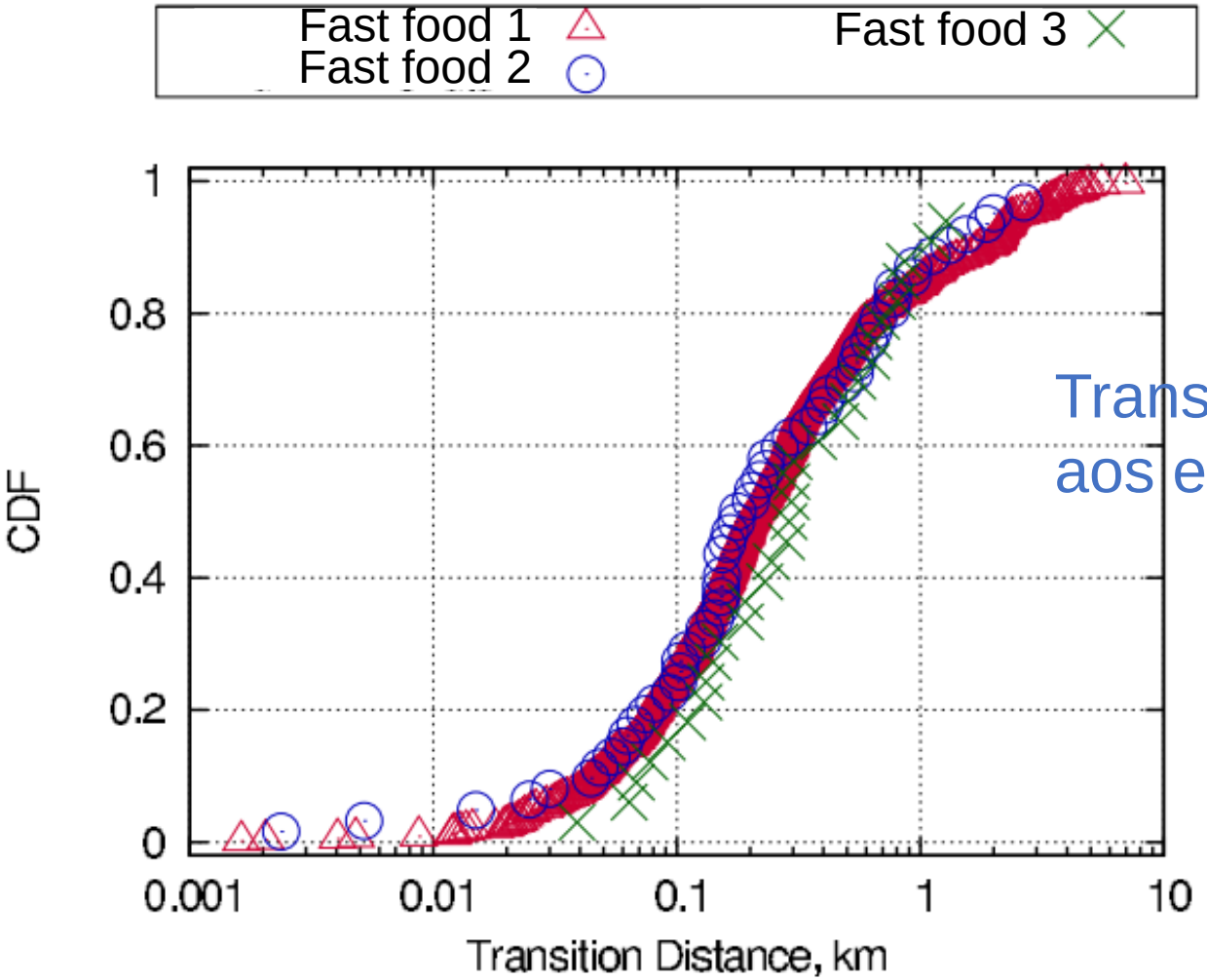
Distância entre check-ins

Distância geográfica entre fotos consecutivas de um mesmo usuário.



20% dos compartilhamentos de localizações acontecem em até 1 km de distância [Noulas et al. 2011]

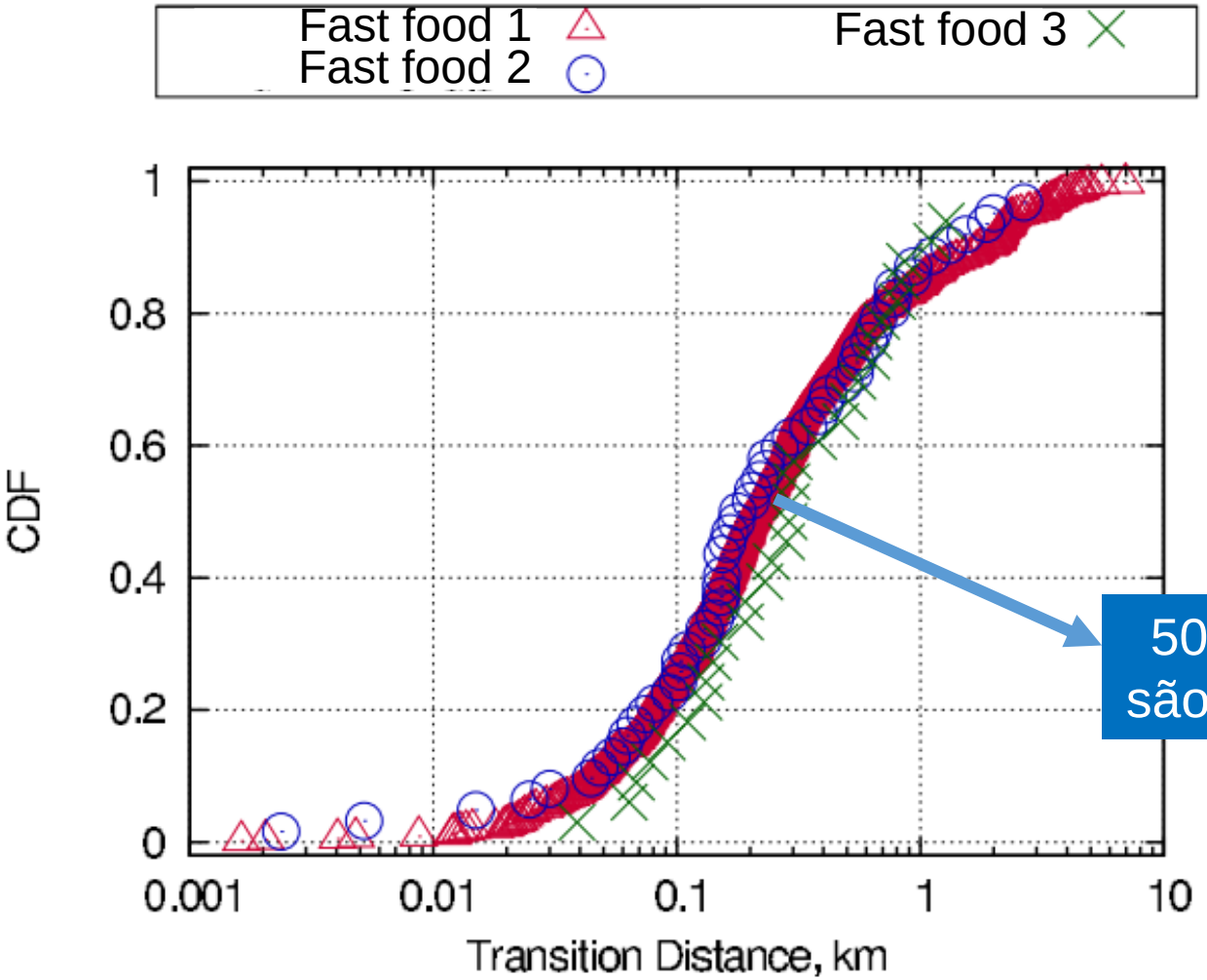
Distância entre check-ins



Transições em direção aos estabelecimentos

Imagem de [Karamshuk et al. 2013]

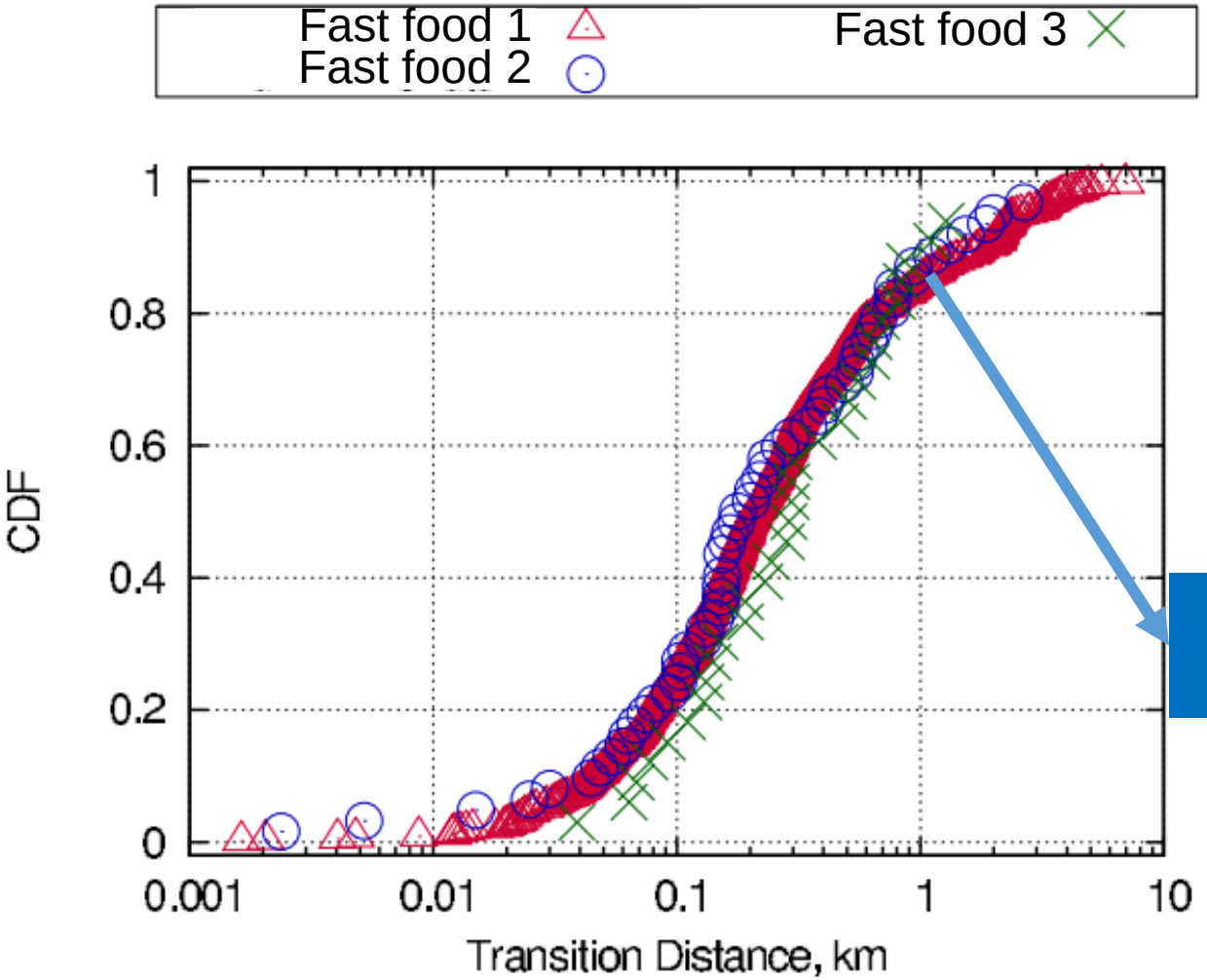
Distância entre check-ins



50 % das transições são de até 300 metros

Imagem de [Karamshuk et al. 2013]

Distância entre check-ins



90 % das transições são de até 1km

Imagem de [Karamshuk et al. 2013]

Detalhes técnicos

A fórmula de Haversine

Importante equação usada em navegação, fornecendo distâncias entre dois pontos de uma esfera

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

d	é a distância entre os dois pontos
r	é o raio da esfera (raio da Terra = 6372,8 km)
ϕ_1, ϕ_2	latitude do ponto 1 e latitude do ponto 2
λ_1, λ_2	longitude do ponto 1 e longitude do ponto 2

Detalhes técnicos

A fórmula de Haversine

Importante equação usada em navegação, fornecendo distâncias entre dois pontos de uma esfera

$$d = 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right)$$

Exemplo de código disponível

d	é a distância entre os dois pontos
r	é o raio da esfera (raio da Terra = 6372,8 km)
ϕ_1, ϕ_2	latitude do ponto 1 e latitude do ponto 2
λ_1, λ_2	longitude do ponto 1 e longitude do ponto 2

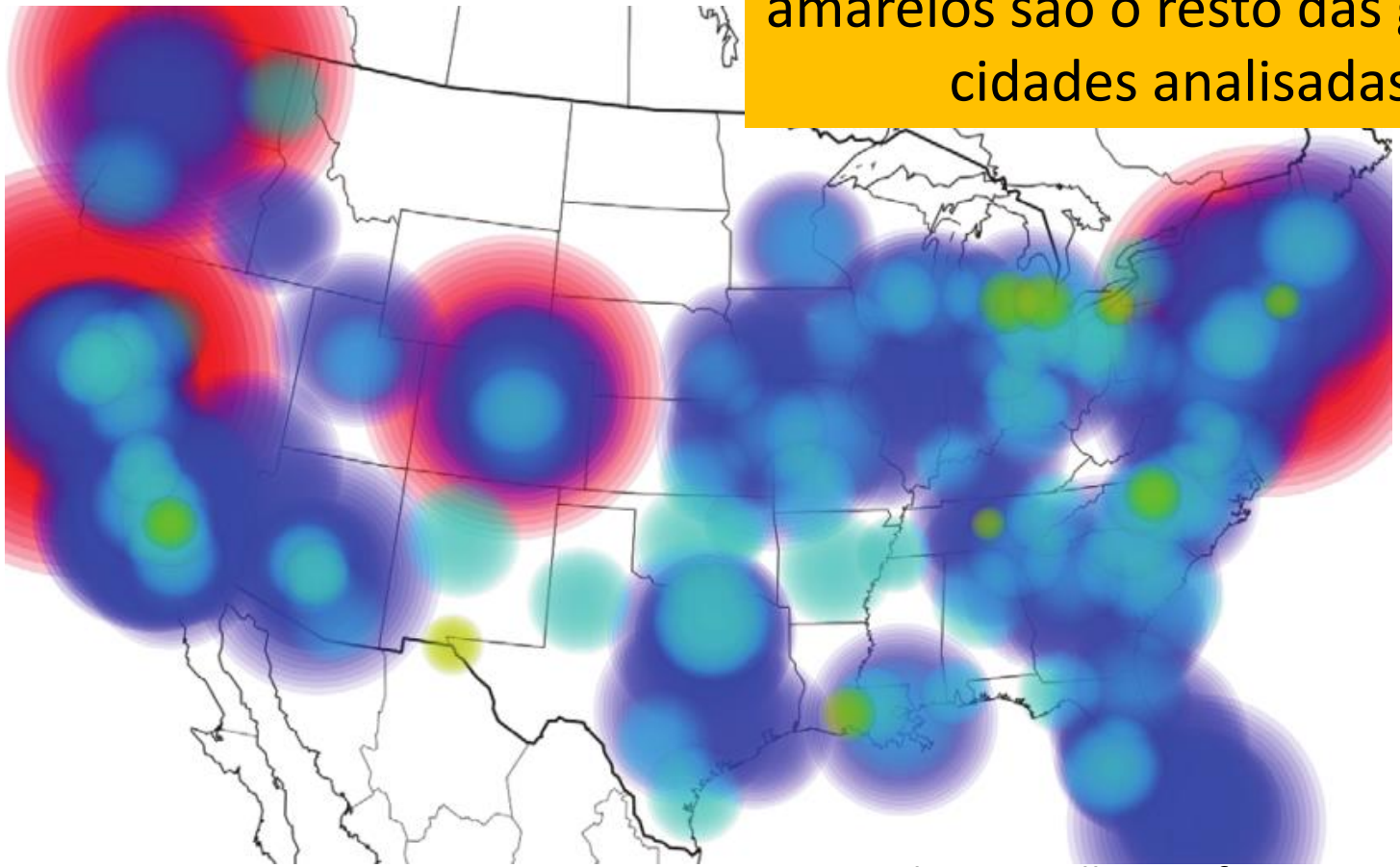
Raio de giro

bolhas vermelhas são cidades com um raio de giro > 500 milhas

azuis >250 milhas

Ciano > 125 milhas

amarelos são o resto das grandes cidades analisadas

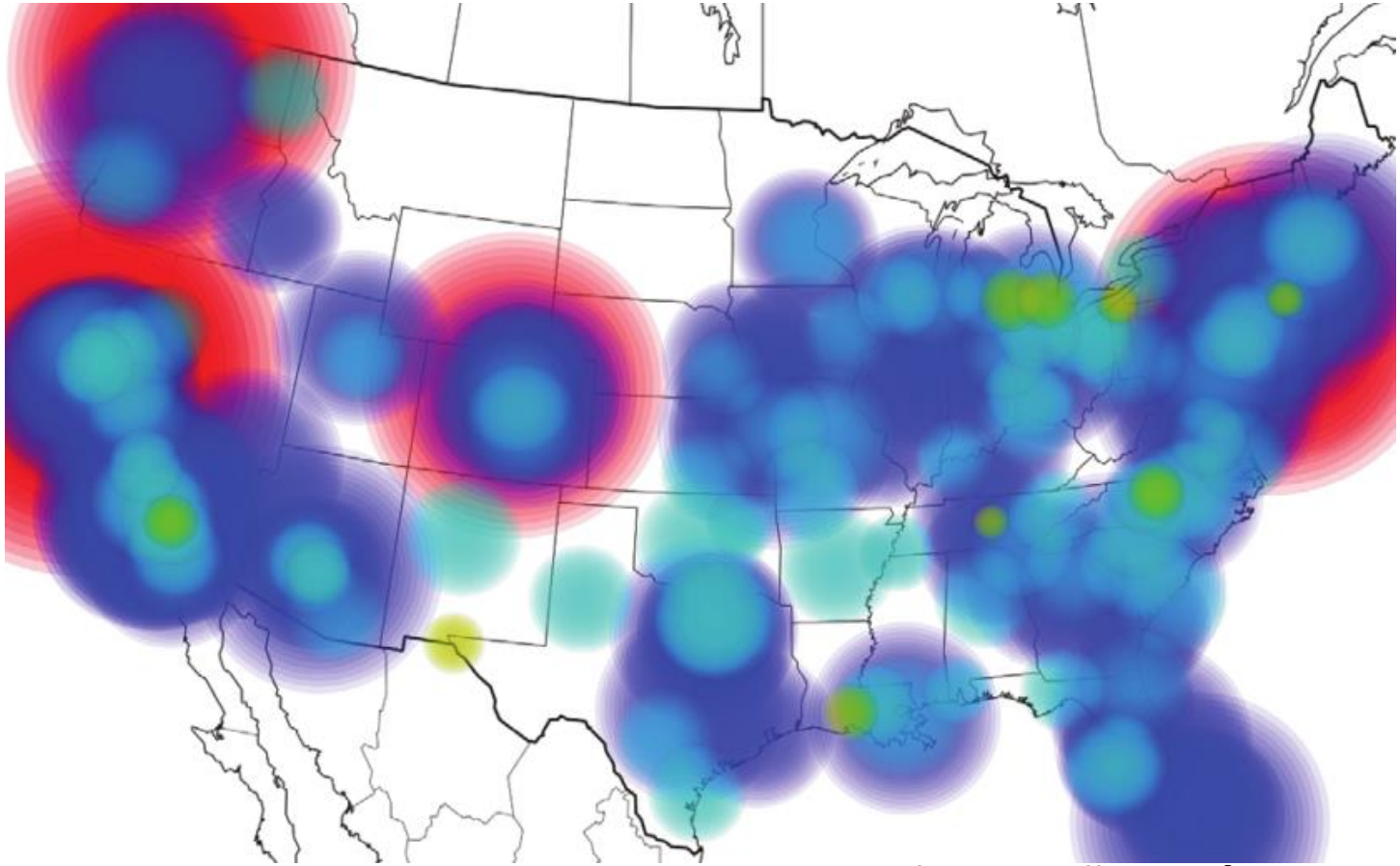


Média de todos os usuários da cidade

Exploring Millions of Footprints in Location Sharing Services – ICWSM 11

Raio de giro

Cidades costeiras e estados centrais possuem maior raio de giro



Média de todos os usuários da cidade

Exploring Millions of Footprints in Location Sharing Services – ICWSM 11

Detalhes técnicos

Baixo raio de giro normalmente indica um usuário que viaja principalmente localmente

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{r}_i - \mathbf{r}_{\mathrm{cm}}\right)^2}$$

n é o número de dados do usuário

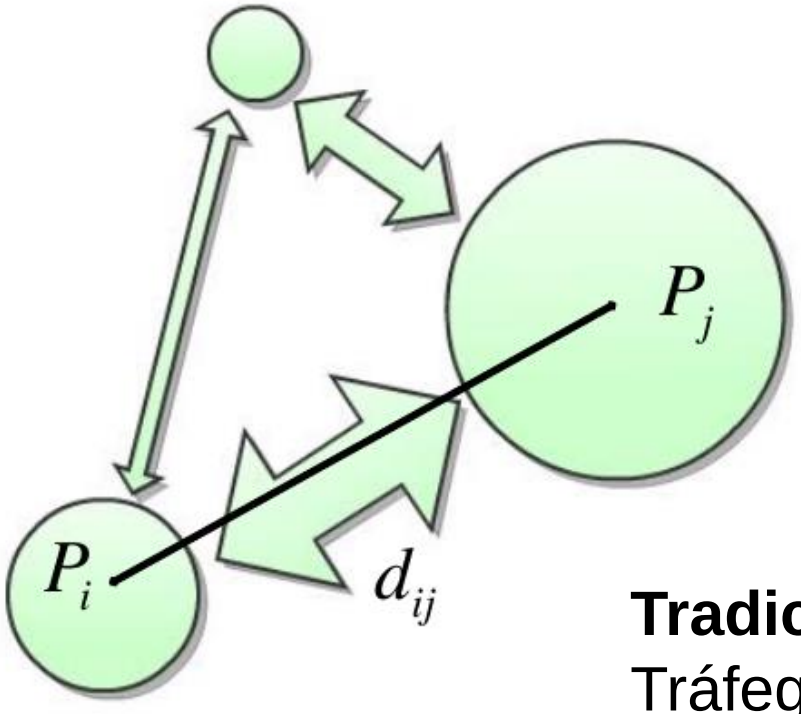
$\mathbf{r}_i$ localização de um dado compartilhado específico

$\mathbf{r}_{\mathrm{cm}}$ centro de massa do usuário (média simples da localização de todos os dados)

Modelo de Gravidade

Newton: "Any two bodies attract one another with a force that is proportional to the product of their masses and inversely proportional to the square of the distance between them."

$$T_{ij} = k \frac{P_i^{\alpha} P_j^{\beta}}{d_{ij}^{\gamma}}$$



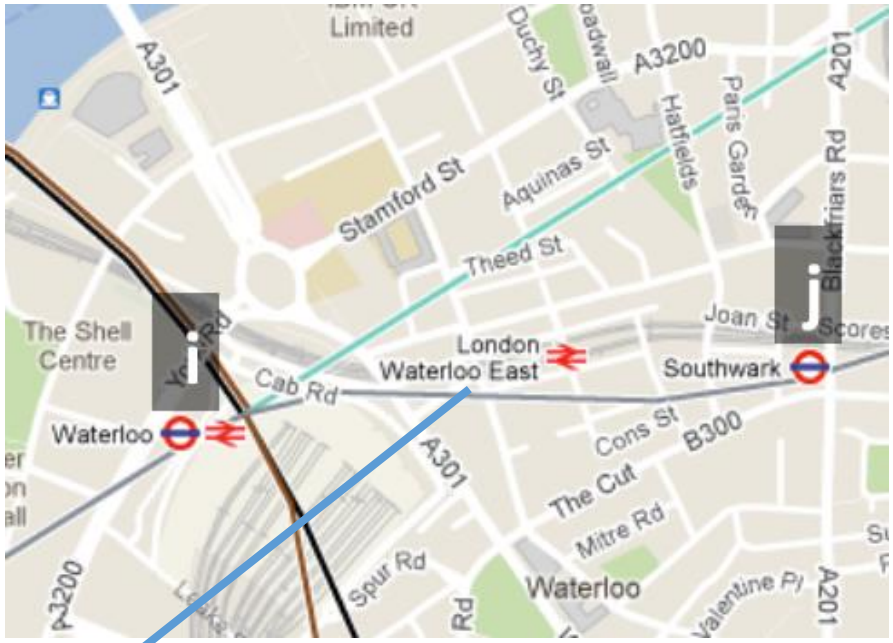
Tradicionalmente:

Tráfego

Migração entre áreas

…

Modelo de Gravidade

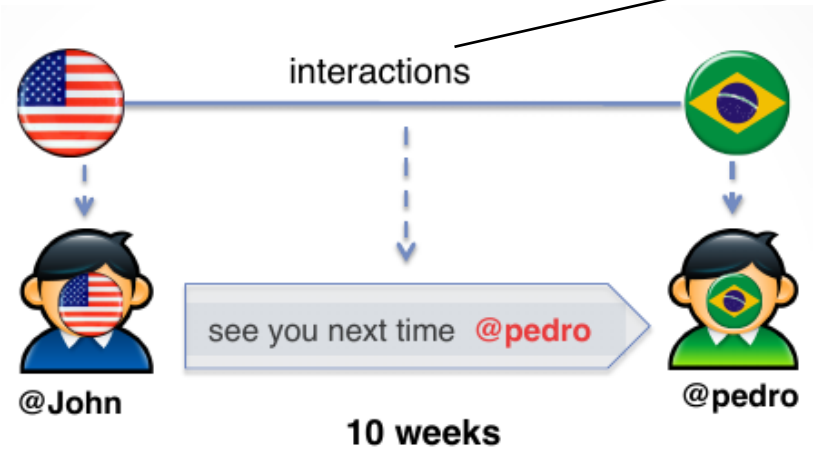


Modelo explica bem

Modelo de Gravidade

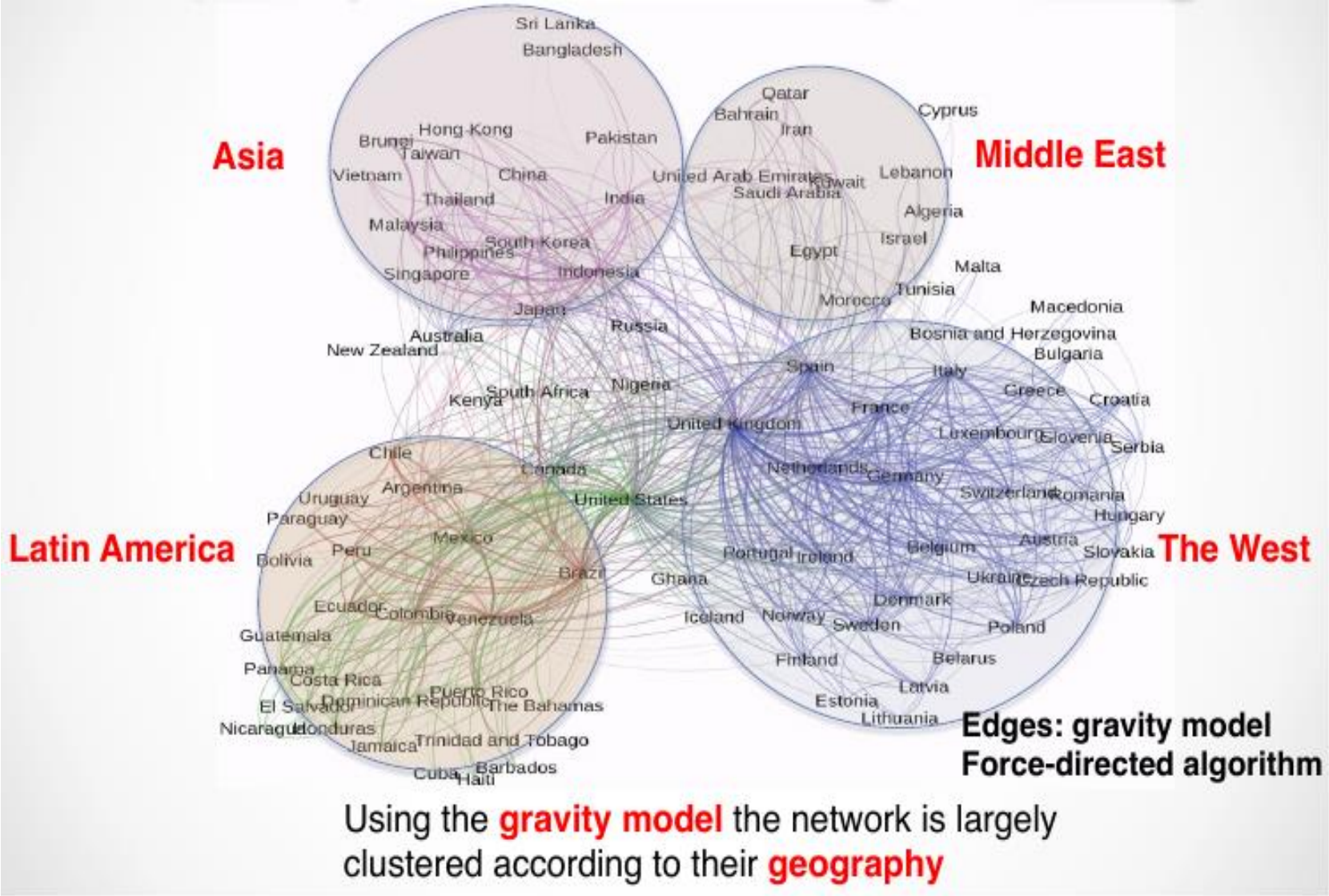
Twitter

Distância



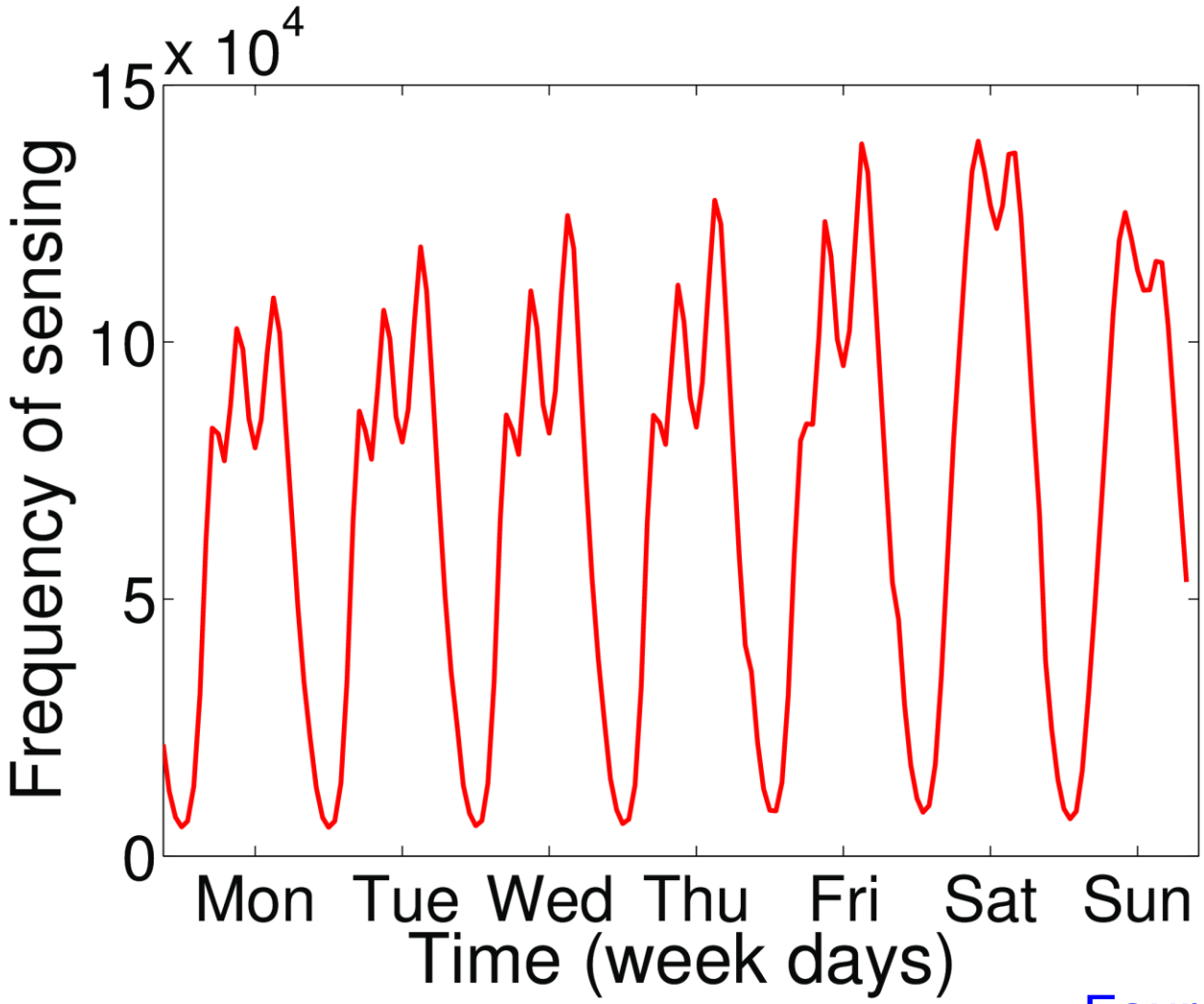
Twitter ain't without frontiers. CSCW 2014

Modelo de Gravidade



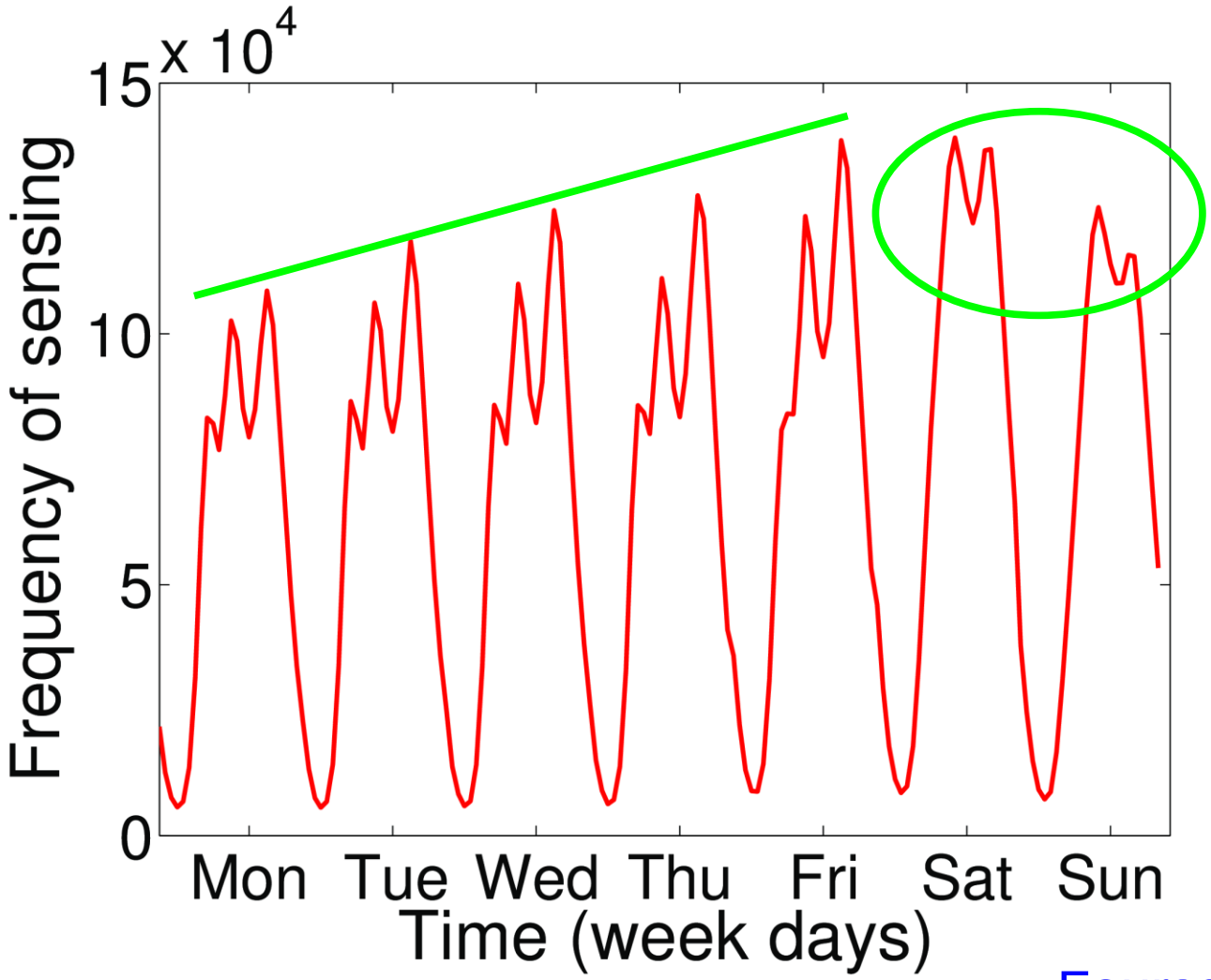
Twitter ain't without frontiers. CSCW 2014

Rotinas e o compartilhamento



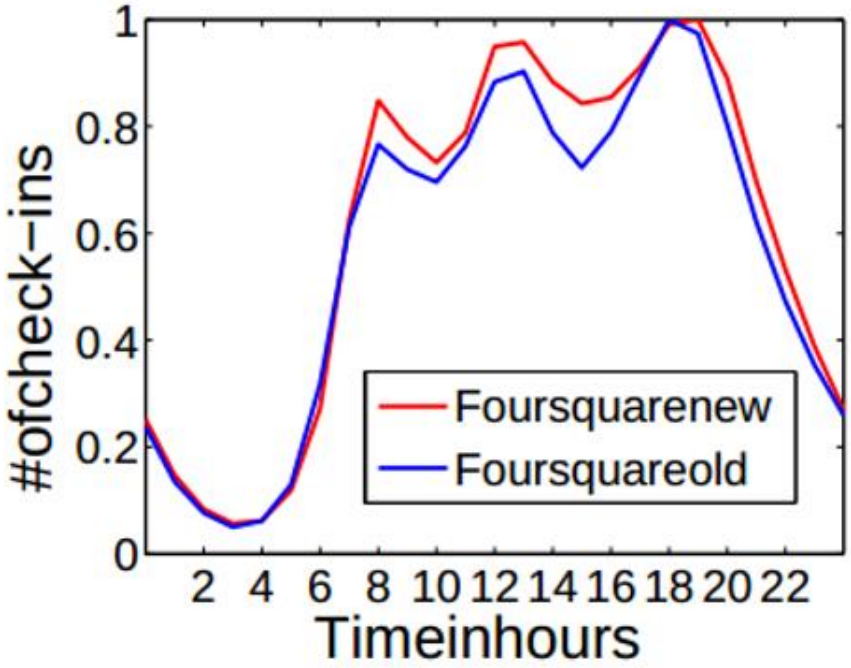
Foursquare dataset

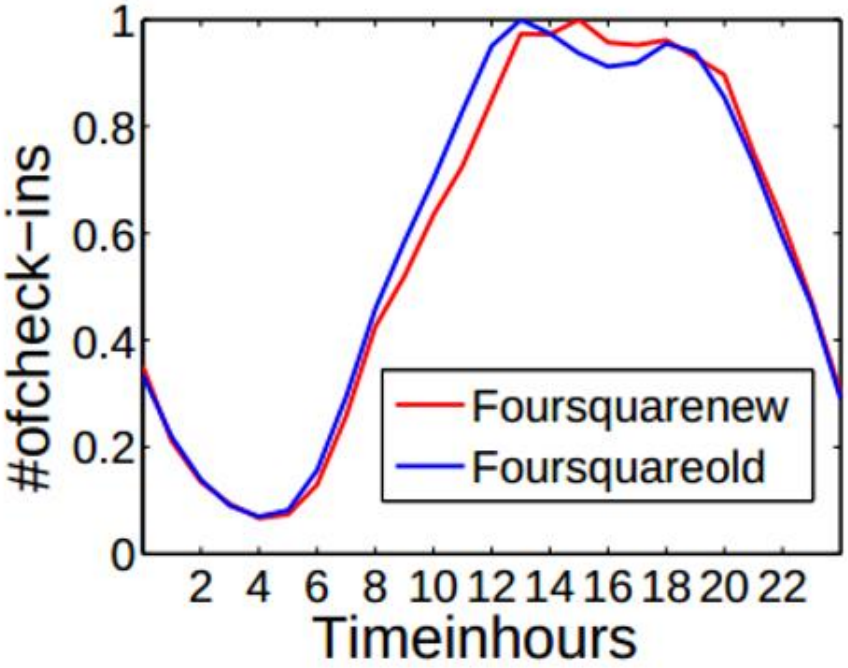
Rotinas e o compartilhamento



Foursquare dataset

Rotinas e o compartilhamento

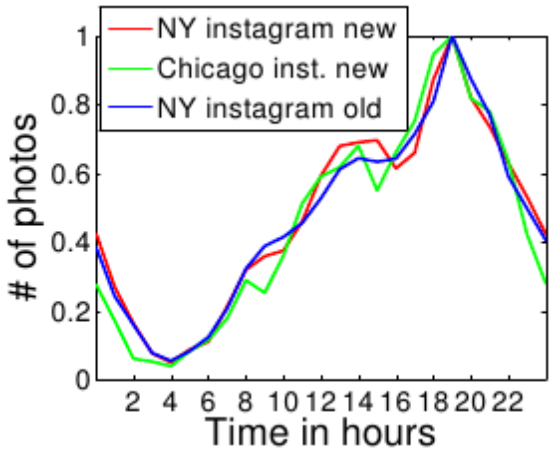




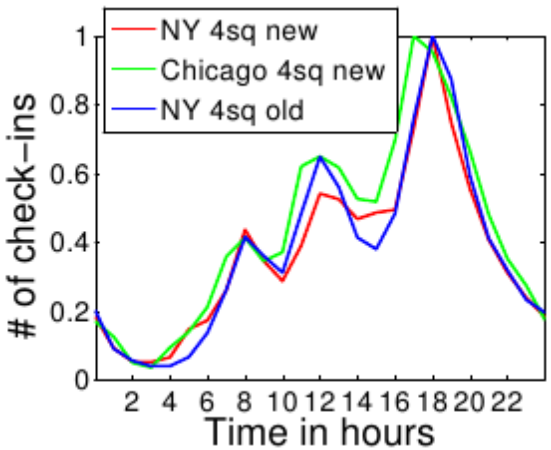
Rotinas e o compartilhamento

E se olharmos para cidades?
O padrão se repete?

Rotinas e o compartilhamento



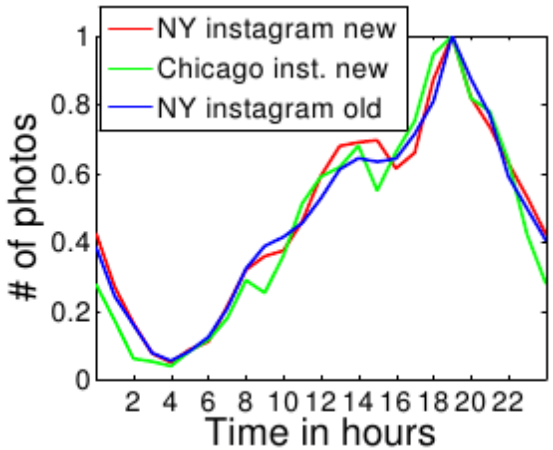
Instagram



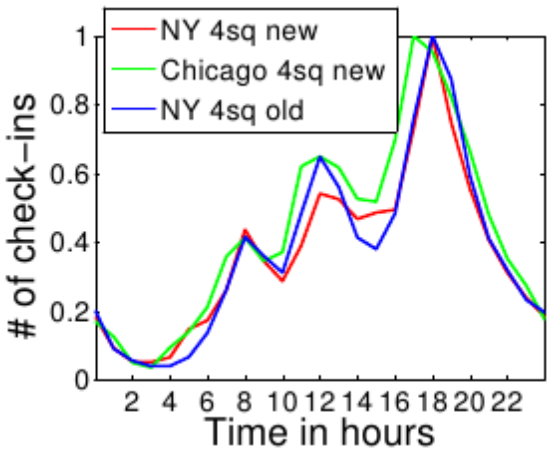
Foursquare

A comparison of Foursquare and Instagram to the Study of City Dynamics and Urban Social Behavior. Urbcomp'13

Rotinas e o compartilhamento



Instagram

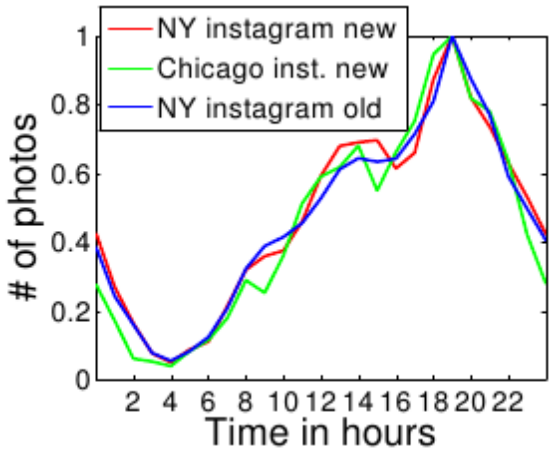


Foursquare

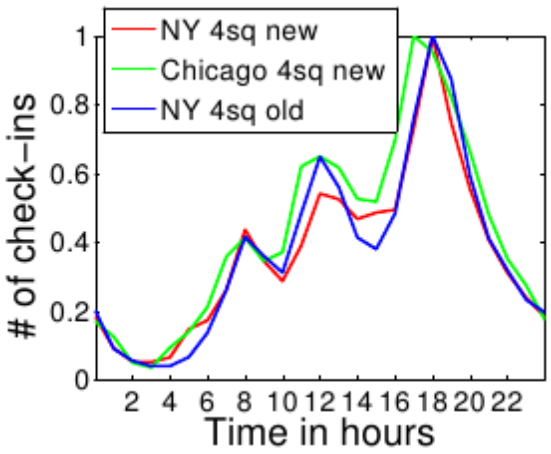
A curva para cada cidade é similar para cada sistema

A comparison of Foursquare and Instagram to the Study of City Dynamics and Urban Social Behavior. Urbcomp'13

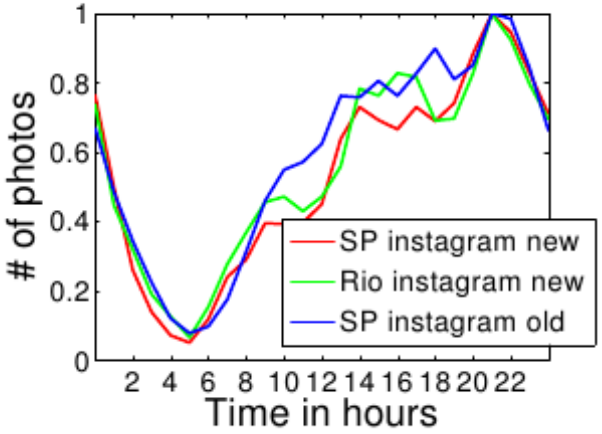
Rotinas e o compartilhamento



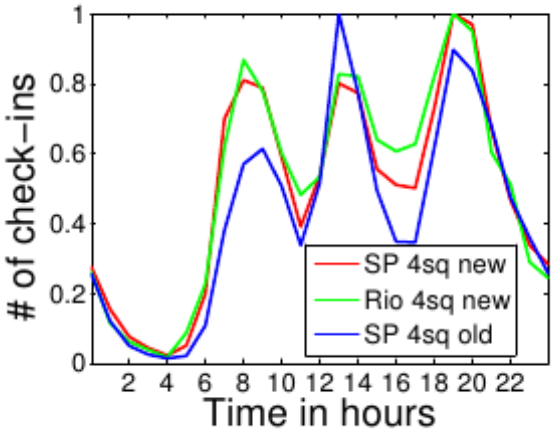
Instagram



Foursquare

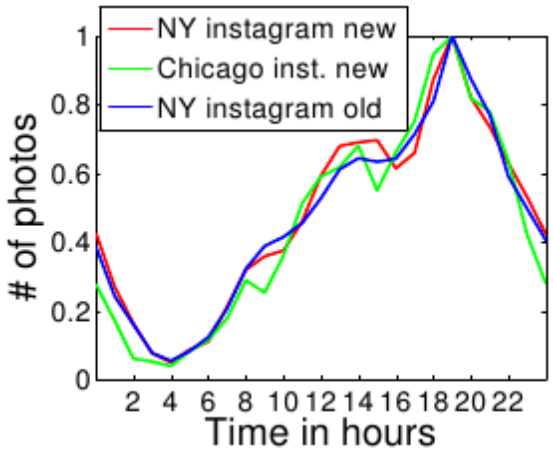


Instagram

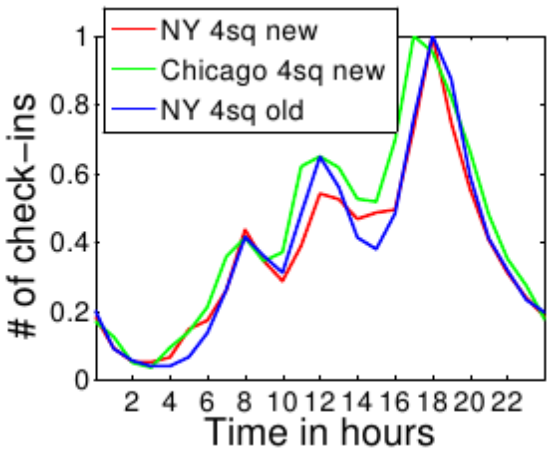


Foursquare

Rotinas e o compartilhamento

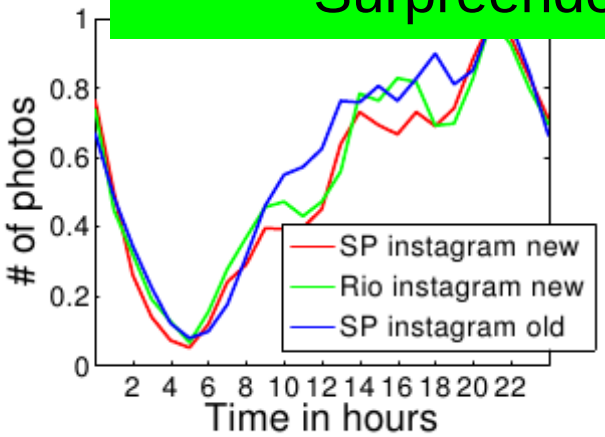


Instagram

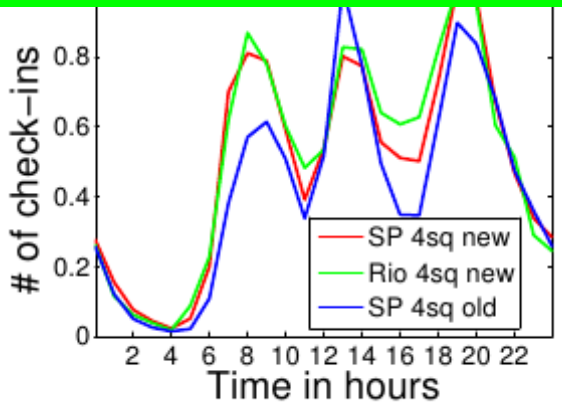


Foursquare

Surpreendente!! (Cultura?)



Instagram



Foursquare

Observações

RSP pode ser uma plataforma **nova** e **barata** para estudar o comportamento social urbano em tempo real e em larga escala