数据获取、分析与表达

Getting, Cleaning and Visualizing Data

Dong Lei | School of Arch

2017.06.12 | Weibo: LEI_DONG

arch.dongl@gmail.com

Web : donglei.org

内容安排 | Outline

1. 总述、软件，简单的流程（例子）；
2. 有了数据，如何画图（统计回归，地图）；
3. 如何获取、处理数据；
4. 关于定性与定量研究。

技术、问题与系统| Preface

1. 技术（Technique）

研究时间越长，技术占的重要性越低。我的理解是这样：随着研究的进行，研究者可以学会更多的技术，当你掌握了一项技术后，它就变成了一个工具，对于研究来说，技术是“工具”层面的重要。但是前提是要“掌握技术”，而且技术（相对于问题和系统）比较好教，好学，好分享。

2. 问题（Question）

研究领域的选择，未来方向的确定是提问题提出来的。但提问题要注意，不是随随便便的问题，是要想Big Question。

3. 系统（System）

一个人永远学不完新东西，所以更重要的是要有自己的研究体系，否则只能要么看别人做什么跟着做什么，或者自己会什么就做什么，东打一枪、西放一炮，很难有系统发现。对于体系的培养，本学科领域的科学史（不是泛的科学史）非常重要，看科学发展的过程，会让人有个所在领域的全局观，能培养一种思维方式。

——施一公

第一章安排 | Chapter 1

1.1 目标 (Why)

1.2 软件 (Who)

1.3 过程 (How)

1.4 例子 (How+)

1.5 Tips

1.6 进一步学习的资料

目标 | Target

Step1 一个想要研究的问题 -> Step2 将问题抽象成可以科学研究的对象 -> Step3 找对应数据 -> Step4 分析 -> Step5 返回问题

目标 | Target

Step1 一个想要研究的问题 -> Step2 将问题抽象成可以科学研究的对象 -> Step3 找对应数据 -> Step4 分析 -> Step5 返回问题

用一盘红烧肉告诉你学士、硕士、博士论文的区别

本科生说把肉放锅里加些东西煮；研究生说这个是不行的，要写出多少肉，多少其他的佐料，怎么煮，煮多长时间；过了一个月，博士出了一本书，书名叫“如何做红烧肉”，打开目录，“第一章， 如何养猪”。

--摘自：http://www.biomart.cn/news/10/104974.htm

目标 | Target

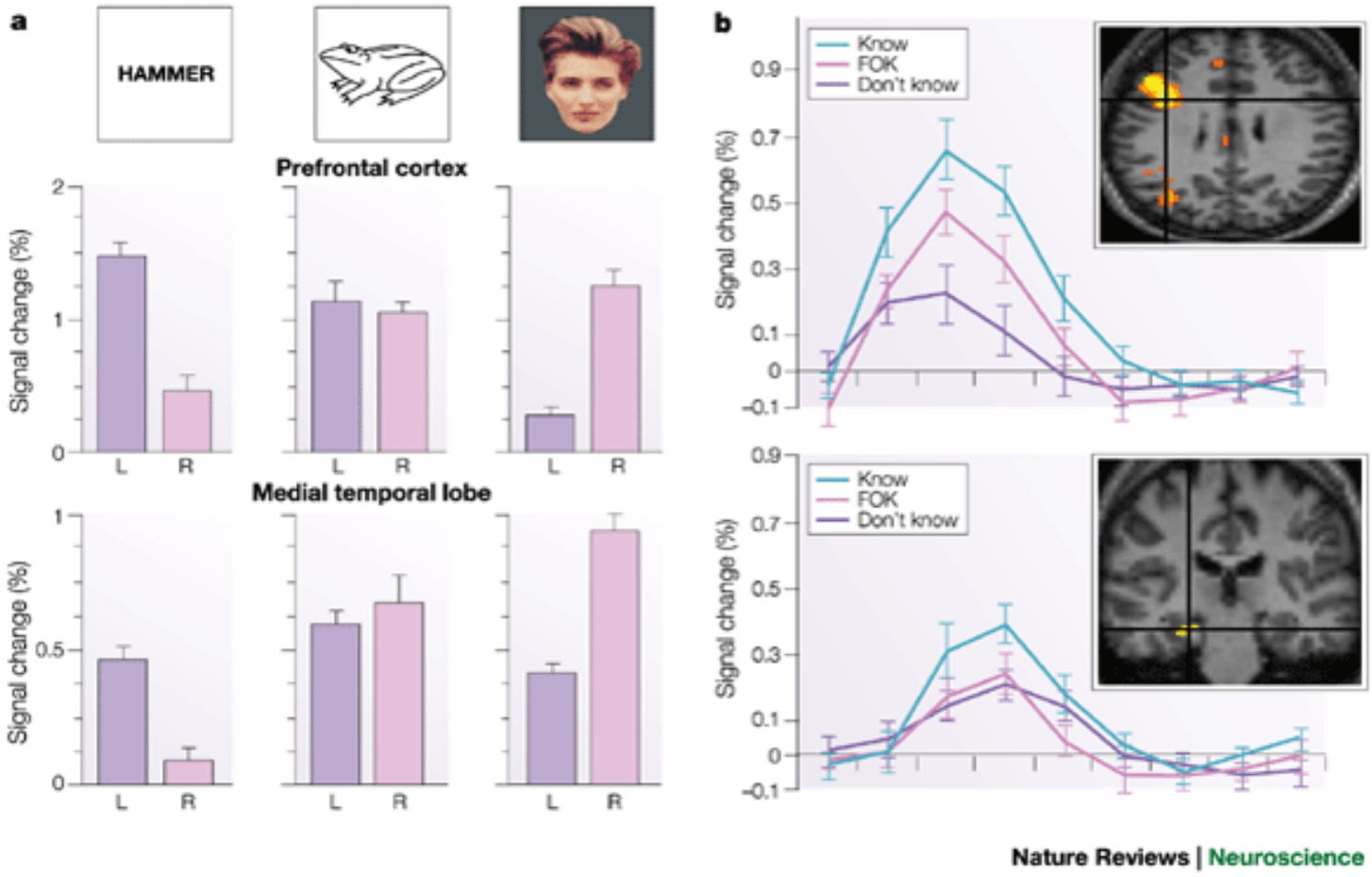
Step1 一个想要研究的问题 -> Step2 将问题抽象成可以科学研究的对象 -> Step3 找对应数据 -> Step4 分析 -> Step5 返回问题

前两步**非常非常**重要，但不同个体，不同研究方向差异很大，所以我们在这里重点关注Step3 Step4。

如果非常认真，一到两个学期（6-12个月，500-1000小时，如果按大学一门3学分的课是100小时左右的话，相当于5-10门三学分的课），可以掌握用于自身研究的最基本技能。

目标 | Target

为什么要做可视化?



软件 | Software

Software: Python -> Canopy, R -> RStudio

如果对数据处理、画图有比较多需求的可以学R；如果对计算机仿真、互联网数据抓取、机器学习有比较多需求的可以试试Python.

Packages: 本章用到的包： Python (Beautiful Soup, csv, urllib2, json); R (ggplot2, ggmap, maps, mapdata)

开源软件的好处就是有不同的人贡献各种“包”，极大方便我们的生活。

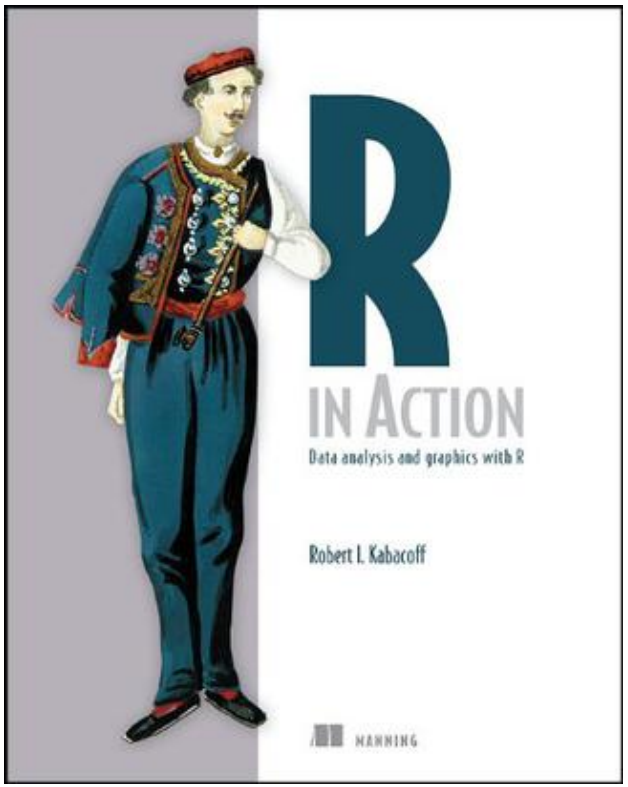
教材 | Books

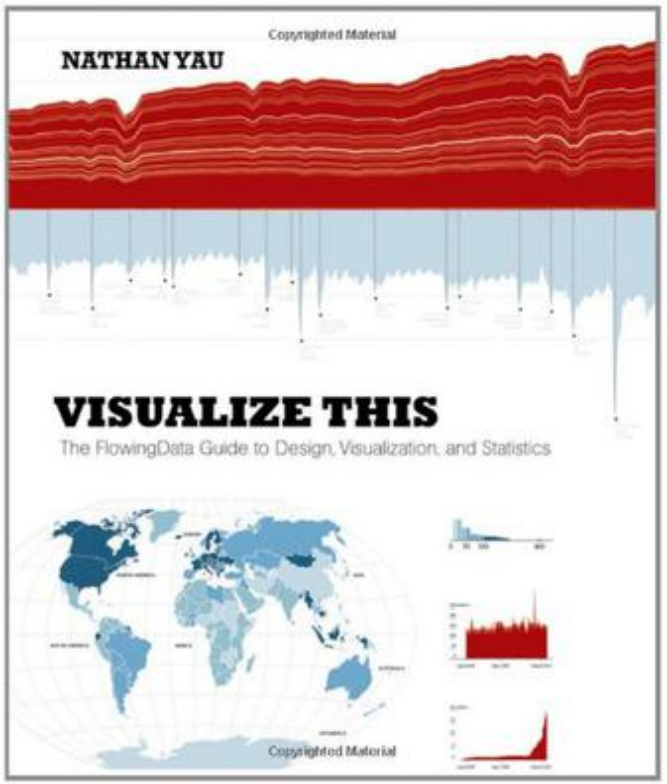
Learn Python the Hard Way (http://learnpythonthehardway.org/)

《用Python做科学计算》（HYRY Studio）

R in Action (http://book.douban.com/subject/6126331/)

Visualize This (http://book.douban.com/subject/5969455/)





教材 | Books

数学之美（http://book.douban.com/subject/10750155/）

公开课 | MOOC

Coursera：

Data Science

[基于R语言的系列课程，约翰霍普金斯大学]，

Model Thinking

[用模型思考问题，密歇根大学]，

Maps and Geospatial Revolution

[简单的地理学介绍及其于ESRI在线平台的地图制作，PennState])

过程 | Process

如何找数据? | Where's the data?

1. 公开数据集（中国的数据可以在这里找到：知网、新浪爱问、人大经济论坛、政府公开数据网站:http://data.stats.gov.cn/、Beijing City Lab 等）

2. 申请政务公开（统计局网站有联系方式，可查经普、人口普查、统计年鉴等）

3. 互联网数据抓取（意识到有**系统误差的存在**，这是由抽样方式造成的，比如微博数据只是针对使用微博的用户）

过程 | Process

互联网数据抓取，以地理信息为例：

如何获取带坐标的数据信息（人口、经济指标、商铺位置等），另一个是如何将这些信息处理，然后在地图上表达。

1. web crawling
2. data clean (structured data vs. unstructured data)
3. geocoding
4. analysis
5. visualize

过程 | Process

geocoding-api

Geocoding API包括地址解析和逆地址解析功能。

http://developer.baidu.com/map/geocoding-api.htm

地址解析是指，根据地址获取坐标。例如： “北京市海淀区中关村南大街27号”地址解析的结果是“lng:116.31985,lat:39.959836”。json

http://api.map.baidu.com/geocoder?address=地址&output=输出格式类型&key=用户密钥&city=城市名

逆地址解析是指，根据坐标获取地址。例如： “lat:31.325152,lng:120.558957”逆地址解析的结果是“江苏省苏州市虎丘区塔园路318号”。

http://api.map.baidu.com/geocoder?location=纬度,经度&output=输出格式类型&key=用户密钥

过程 | Process

geocoding-api

备注：

1. city属于可选参数，通常情况可以不使用，若解析无结果，请尝试增加此字段。

2. 支持名胜古迹、标志性建筑物名称解析返回经纬度坐标。

3. 若解析status字段为OK，若结果内容为空，原因分析及可尝试方法：地址库里无此数据，本次结果为空；加入city字段重新解析；或将过于详细或简单的地址更改至省市区县街道重新解析。

4. 逆地址解析location参数传入的参数格式是(纬度lat，经度lng)。

5. 因为Geocoding和反Geocoding使用的门址数据以及算法都不是一样的，所以会出现不能一一对应的现象。

6. 解析过程中可能会出现一对坐标值对应多个地址门牌信息，将返回距离坐标点最近的一个地址门牌信息。(百度的接口)

过程 | Process

转坐标 WebMercator2LonLat

Google Map, OpenStreetMap等很多地图信息在Web端展示的时候很多是Mercator投影方式，在计算的时候需要转成经纬度。

```
# -*- coding: utf-8 -*-
import math
#墨卡托转经纬度
def Mercator2LonLat(x, y):
    x = x/20037508.34*180
    y = y/20037508.34*180
    Lon = x
    Lat = 180.0/math.pi*(2.0*math.atan(math.exp(y*math.pi/180.0))-math.pi/2.0)
    return Lon, Lat
```

过程 | Process

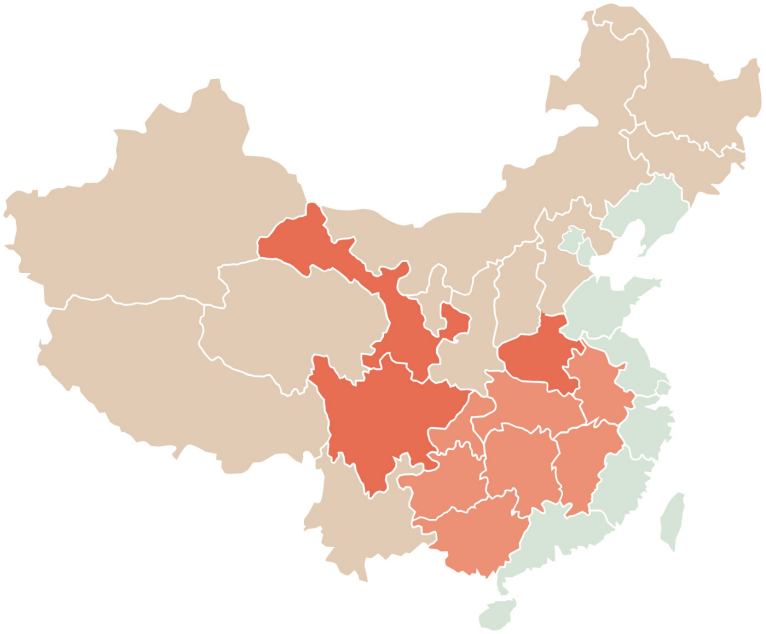
算距离（根据经纬度）

```
import math
Earth_Radius = 6378 * 1000

def rad(d):
    return d * math.pi / 180.0 #注意.0

def GetDistance(lat1, lng1, lat2, lng2):
    a = rad(lat1) - rad(lat2)
    b = rad(lng1) - rad(lng2)
    dis = 2 * Earth_Radius * math.asin(math.sqrt(math.pow(math.sin(a/2),2) +
math.cos(rad(lat1))* math.cos(rad(lat2)) * math.pow (math.sin(b/2),2)))
    return dis
```

例子 | Examples

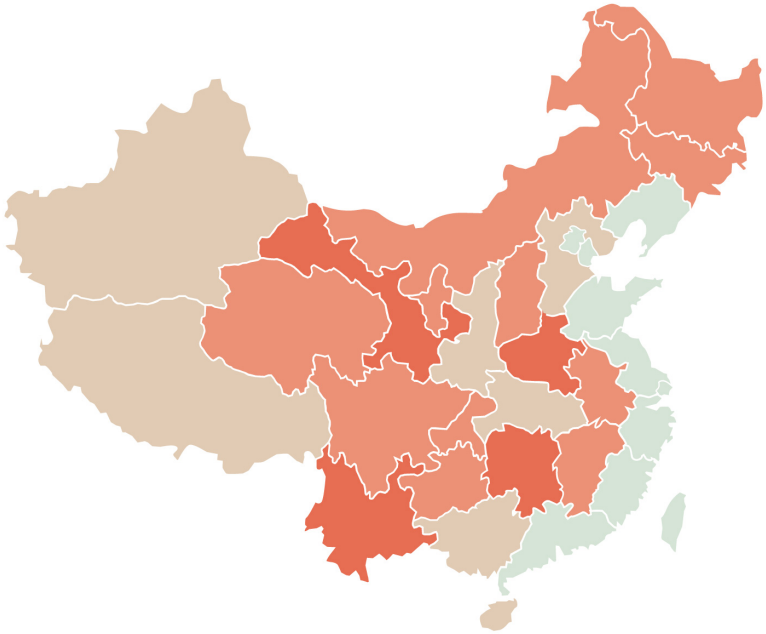
1. 不需要编程的基本表达：

2012年 | Year 2012



21个省市自治区，共59人

2013年 | Year 2013



22个省市自治区，共76人

0人　1-2人　3-4人　>5人

资料来源：XX

例子 | Examples

1. 不需要编程的基本表达：

基础资料：0)数据，因人而异；1）矢量中国地图（搜索引擎很容易找到）；2）配色方案，可参考一个不错的网站ColorBrewer；3）软件工具：AI（illustrator）

强烈推荐经常看一下经济学人杂志和纽约时报的图表，我认为是所有出版物中做的比较好的。（比如个http://www.economist.com/content/chinese_equivalents）

如何搞定中国的地图请看：终于搞定了中国分省市地图（http://yihui.name/cn/2007/09/china-map-at-province-level/），用R软件绘制中国分省市地图（http://cos.name/2009/07/drawing-china-map-using-r/）

例子 | Examples

2. ggmap：(R的一个包)

geocode

```
geocode("China")
       lon      lat
1 104.1954 35.86166
```

返回了中国的经纬度坐标。

mapdist

```
mapdist('Tsinghua University', 'Beijing University', 'walking')
                from              to    m    km    miles seconds
1 Tsinghua University Beijing University 3329 3.329 2.068641    2458
     minutes     hours
1 40.96667 0.6827778
```

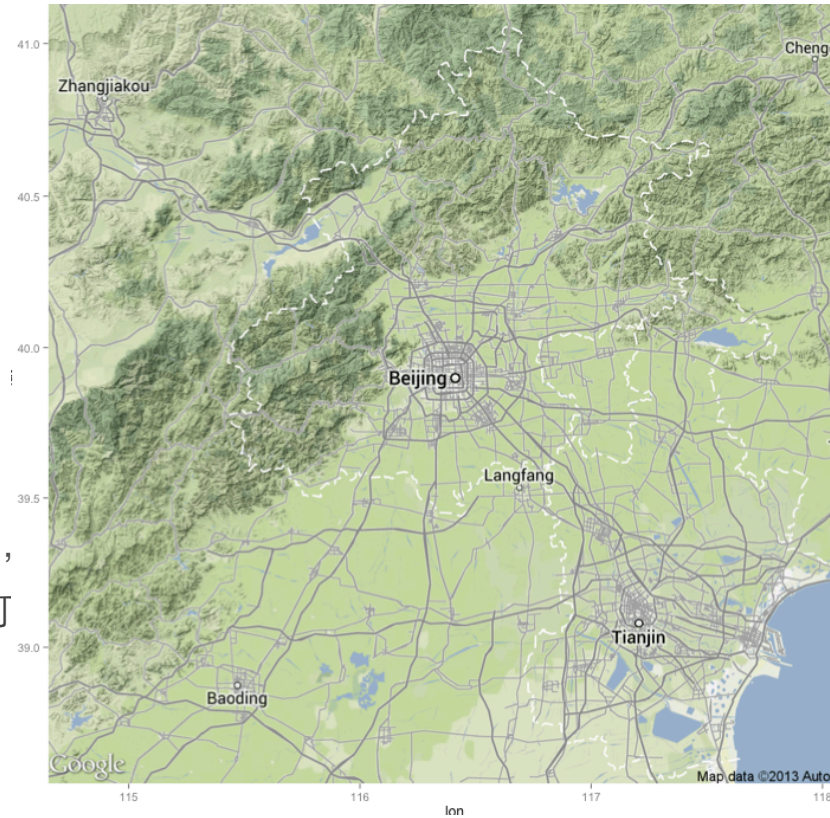
从清华到北大竟然3.3公理，要41分钟，估计是从东门算的吧。.

例子 | Examples

2. ggmap：(R的一个包)

get_map

```
library(ggmap)
library(mapproj)
map <- get_map(location = 'Beijing', zoom = 8)
#以北京为中心，zoom代表类似放大倍数.
```



京津冀地区出现了。GOOGLE地图本身就是个大的GIS系统，如果在上面一行代码里加上 maptype = ‘roadmap’就可以得到北京的路网图。

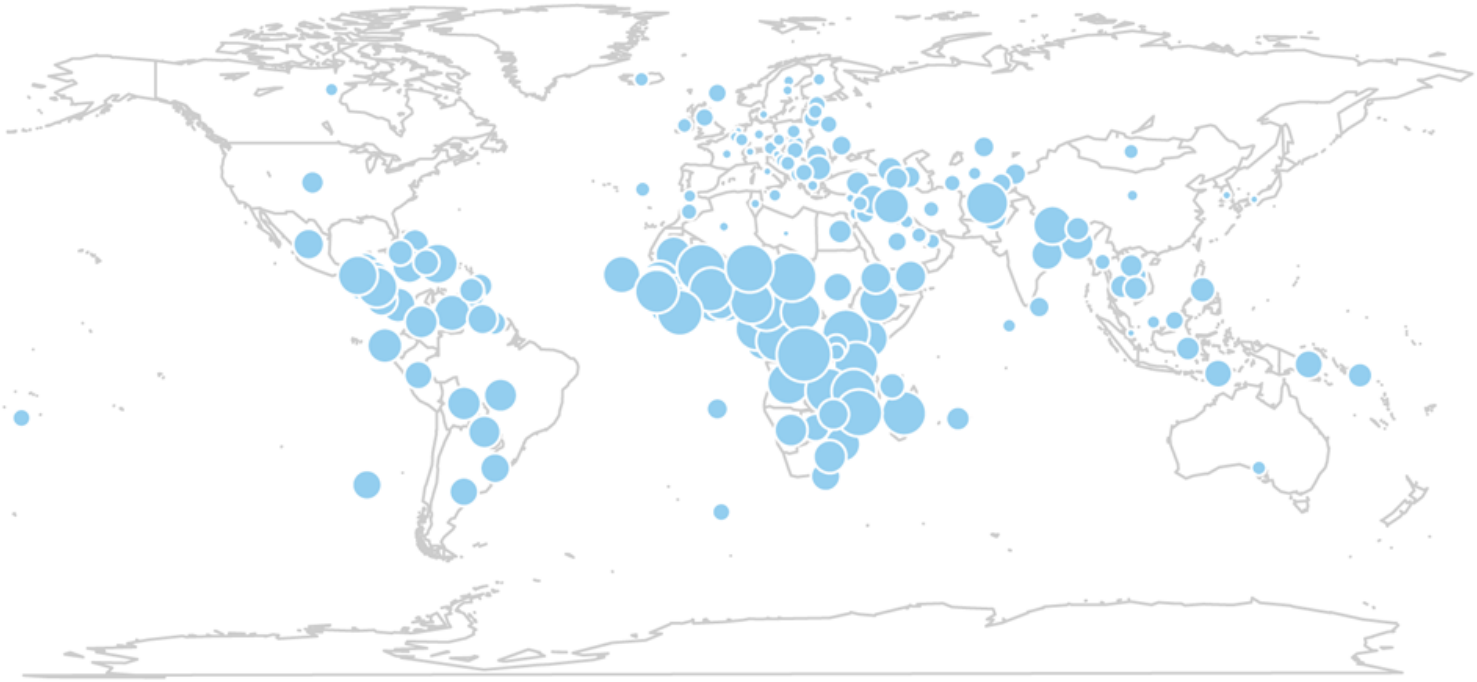
```
map <- get_map(location = 'Beijing', zoom = 10, maptype = 'roadmap') #北京路网
```

参考：http://donglei.name/2013/10/map-data-with-r/

例子 | Examples

3. maps：(R的一个包)

数据：《2008年联合国人类发展报告》中未成年人生育率（每1000名15-19岁女性中生育数量）。



例子 | Examples

3. maps：(R的一个包)

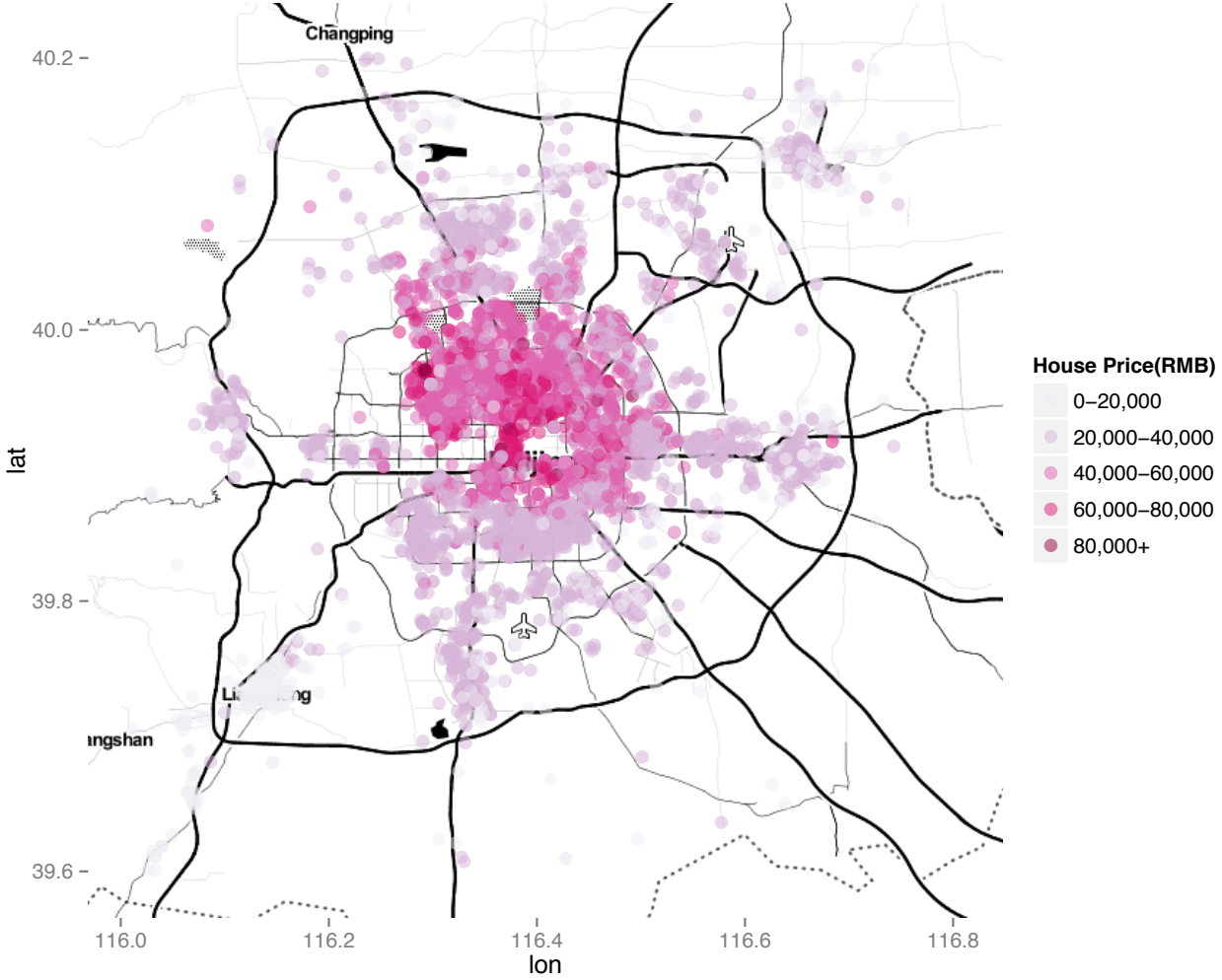
数据：《2008年联合国人类发展报告》中未成年人生育率（每1000名15-19岁女性中生育数量）。

Note: 对CSV数据格式的一点说明。

```
library(maps) #载入maps
fertility <- read.csv("http://book.flowingdata.com/ch08/points/adol-fertility.csv")

map('world', fill = FALSE, col = "#cccccc") #选择世界地图，不填充，底图为灰色

symbols(fertility$longitude, fertility$latitude,
circles = sqrt(fertility$ad_fert_rate), add = TRUE,
inches = 0.1, bg = "#93ceef", fg = "#ffffff") #圆圈的大小（面积而非半径）由生育率决定
```

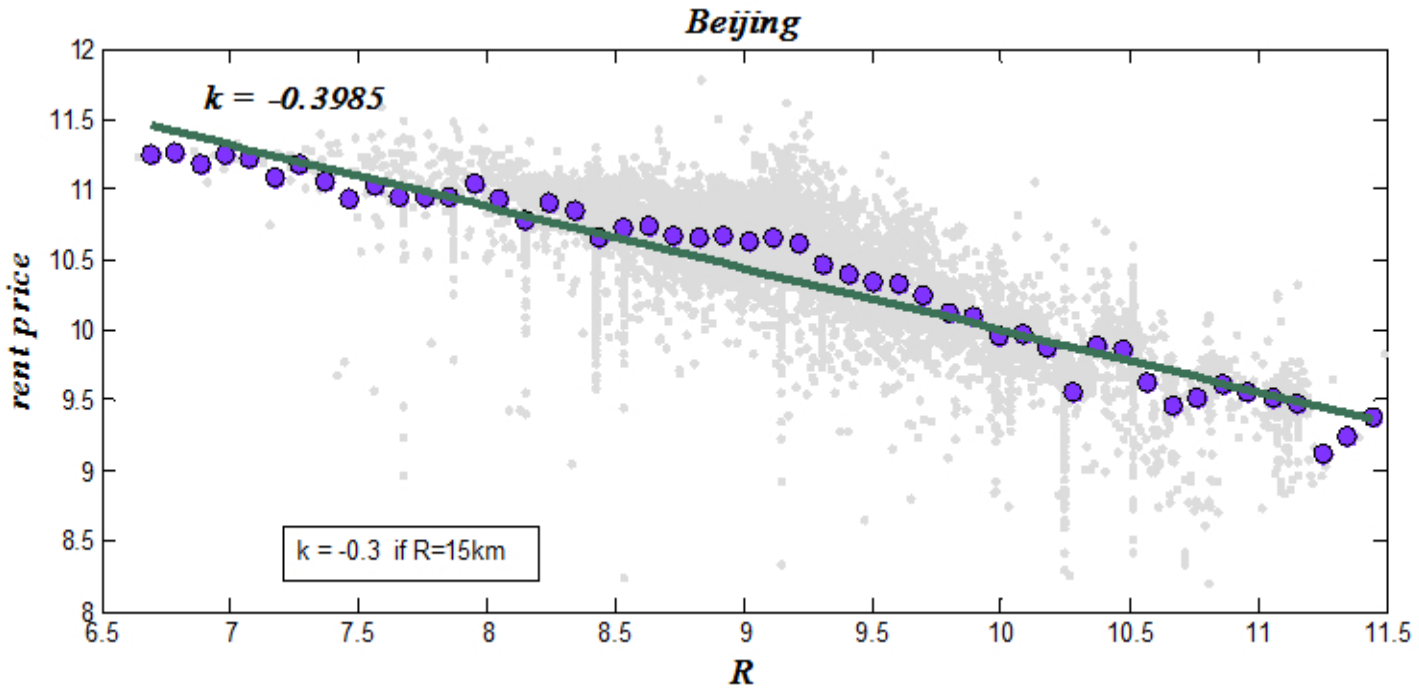
例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap: (R的包)



例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap: (R的包)



例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap: (R的包)

Step 1. Get Data

```
import urllib2
import time

link_start = 'http://beijing.anjuke.com/newmap/search2?model=2&order=null&p=1&bounds='
link_end = '&zoom=14&beh=KW0_DW1_SX0_TD0_SF1_QP0_PX0_JH0_FY1_DQFY_DT0&test= '
rawdata = ''

lat = 39.55
for i in range(30):
    lat += 0.025  lat = lat + 0.025

        lng = 116.05
        for j in range(35):
            lng += 0.025
```

例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap: (R的包)

Step 1. Get Data

```
link = link_start + str(lat) + ',' + str(lat+0.1) + ',' + str(lng) + ',' + str(lng+0.1) + link_end

time.sleep(3)
response = urllib2.urlopen(link)
html = response.read()

index = html.find('],"props")
if  index > 10:
        myItem = html[10:index] + ','
        rawdata += myItem

target = open('center.txt', 'w')
target.truncate()
target.write(rawdata)
target.close()
```

例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap：(R的包)

Step 1. Get Data

```
library(ggplot2)
library(ggmap)

price <- read.csv("HousePrice2013.csv")

price$ID <- (price$Price %/% 20000 + 1)
price$ID[price$ID > 5] <- 5
price$ID <- as.factor(price$ID)

price1 <- price[price$Lng < 116.8 & price$Lng > 116.0,]
price2 <- price1[price1$Lat < 40.2 & price1$Lat > 39.6,]

map <- get_map(location = "beijing", zoom = 10, messaging = FALSE, color = "bw",
maptype = 'toner', source="stamen")
```

例子 | Examples

4. urllib2: (Python的库) ggplot2, ggmap：(R的包)

Step 1. Get Data

```
g <- ggmap(map)

p <-g + geom_point(data=price2, aes(x=price2$Lng, y=price2$Lat,
colour=price2$ID), size=3, alpha=0.5)

p <- p + scale_colour_manual(name="House Price(RMB)",
        values=c("#f1eef6", "#d7b5d8", "#df65b0", "#dd1c77", "#980043"),
        labels=c("0-20,000","20,000-40,000","40,000-60,000","60,000-80,000", "80,000+"))

print(p)
```

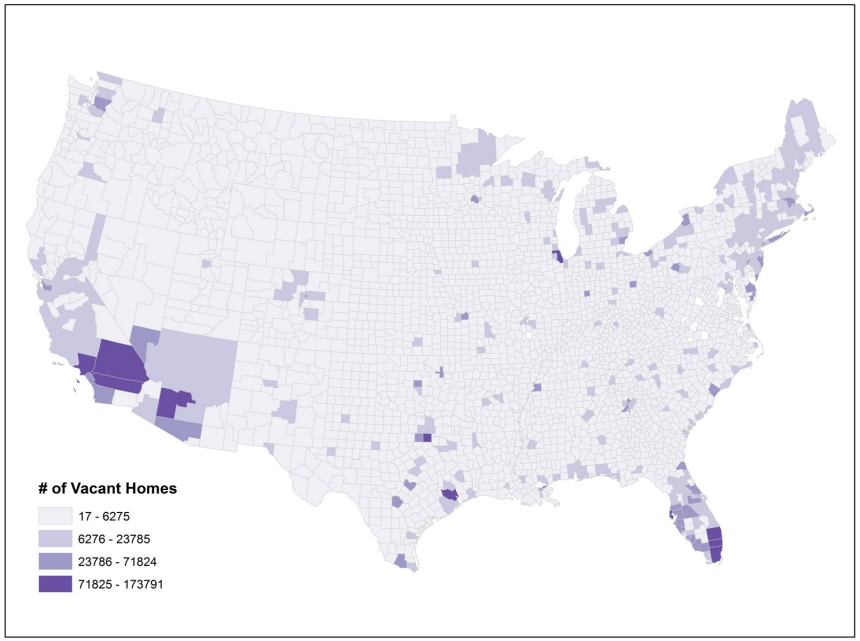
一些可能注意的点 | Tips

- Scale
- Mapping Totals vs. Mapping Rates
- Correlation vs. Causation
- Population Dependence: cause maps of raw values to always highlight heavily populated places

- csv not xls, 汉字编码问题
- About Color (ColorBrewer)
- About Text (http://www.typebrewer.org/)

一些可能注意的点 | Tips

Totals vs. Rates



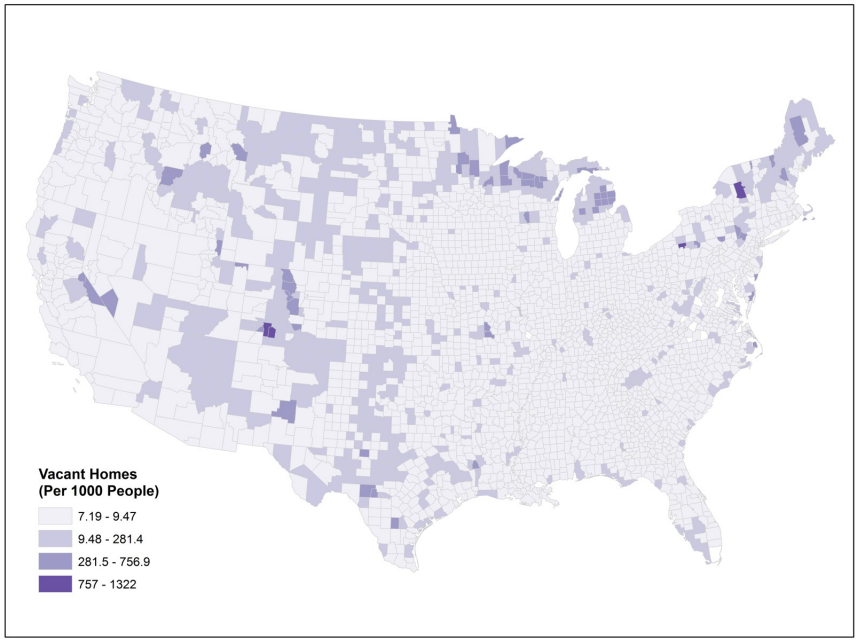
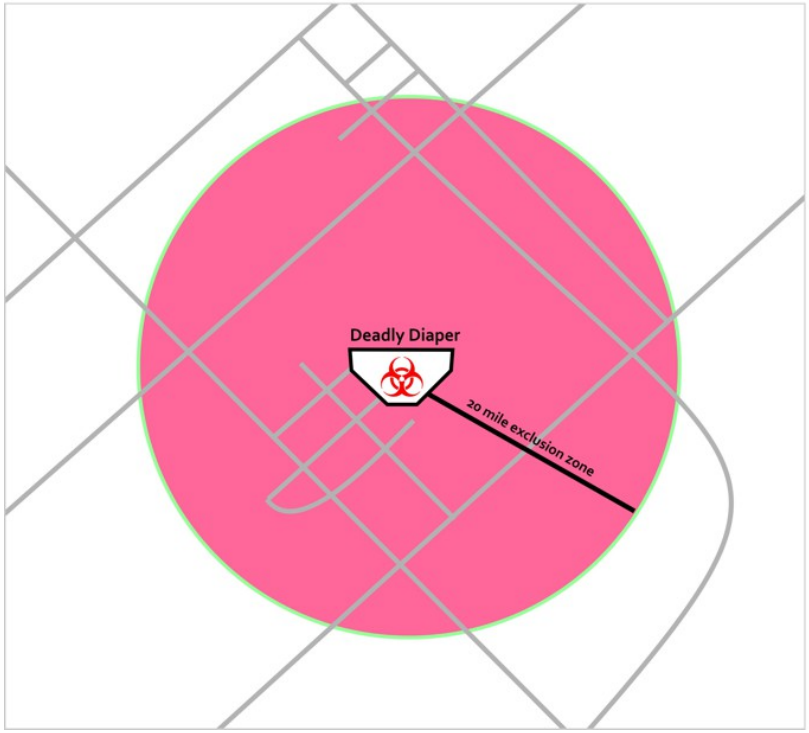


Photo Credit: Anthony C. Robinson

一些可能注意的点 | Tips

Buffer Zones,

20 mile buffer



20 minute drive time buffer

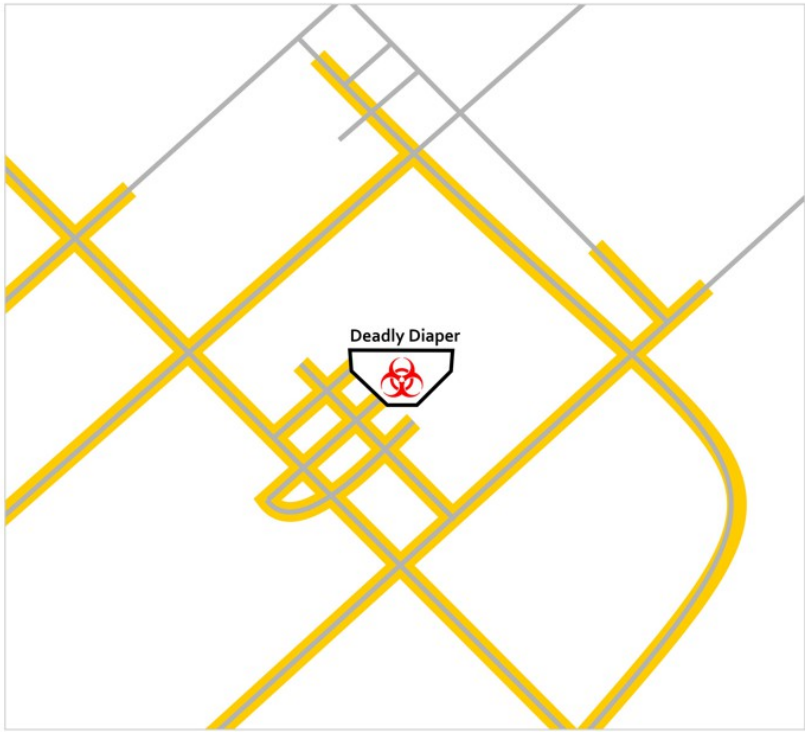
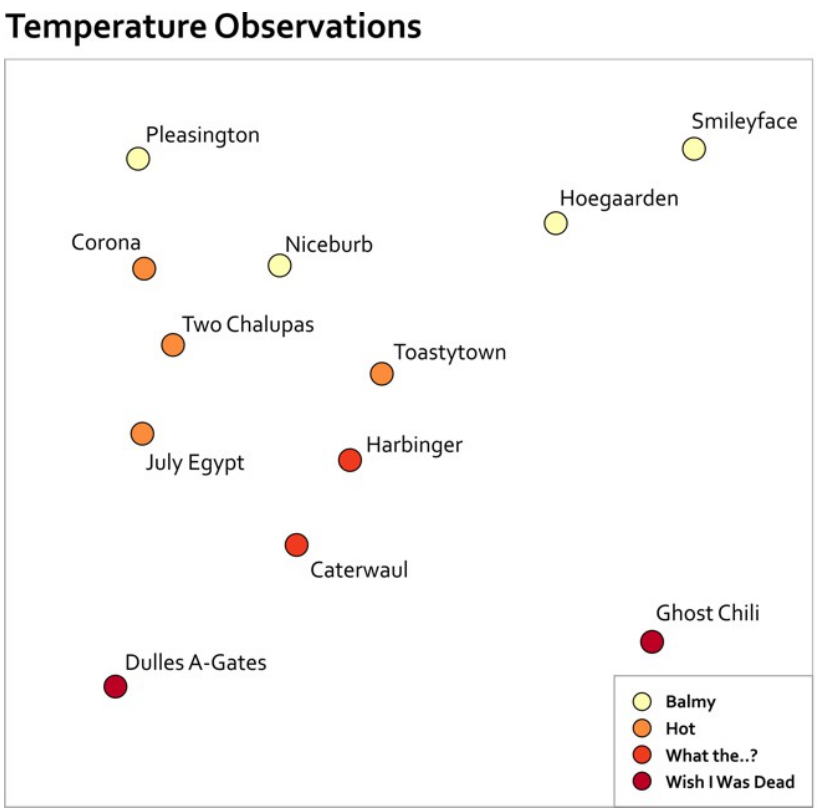


Photo Credit: Anthony C. Robinson

一些可能注意的点 | Tips

Heat Maps



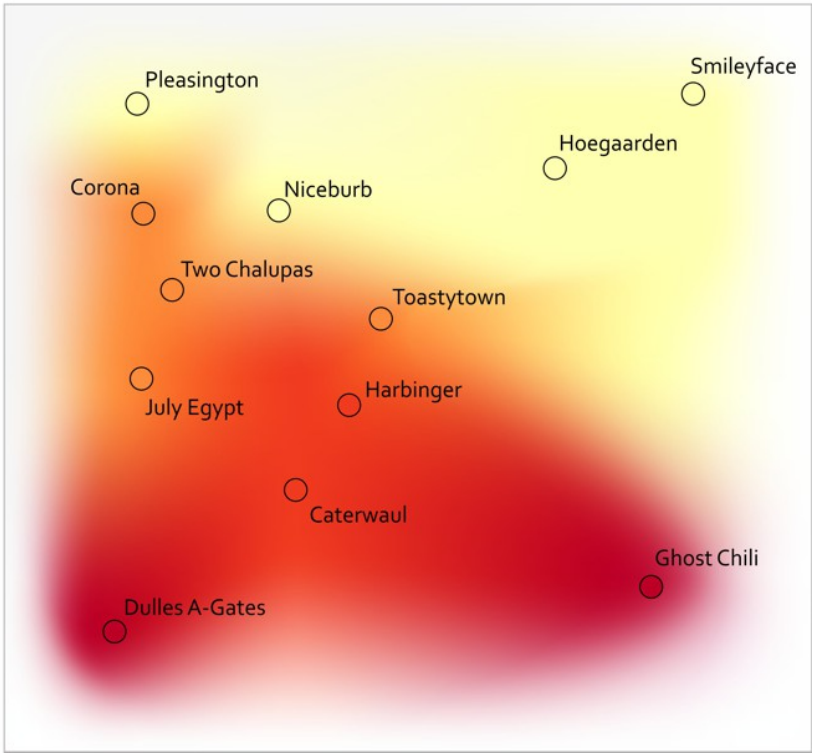


Photo Credit: Anthony C. Robinson

进一步学习 | Reference

Data

OpenStreetMap: http://www.openstreetmap.org/

哥伦比亚大学Socioeconomic data and applications center (SEDAC): http://sedac.ciesin.columbia.edu/data/sets/browse

香港数据： http://www.gov.hk/en/theme/psi/datasets/

United Nations http://data.un.org/

美国数据：http://www.data.gov/ http://www.census.gov/

芝加哥数据：http://thechangelog.com/the-city-of-chicago-is-on-github/

英国数据：http://data.gov.uk/ (伦敦数据：http://data.london.gov.uk/)

德国数据：http://www.govdata.de/

日本数据：http://www.data.go.jp/

其它国家: http://www.data.gov/opendatasites

历史地理：复旦哈佛数据集(禹贡)

人口面板数据：北大，西南财经

进一步学习 | Reference

Visualization

http://visualizing.org/

http://flowingdata.com

http://spatial.ly/r/

http://axismaps.com/portfolio.php

http://www.qgis.org/en/site/

http://developers.cartodb.com/tutorials.html

Code

http://www.pythonchallenge.com/

http://www.codecademy.com/tracks/python