

R for Mapping

Dr Alex Singleton

Department of Geography and Planning

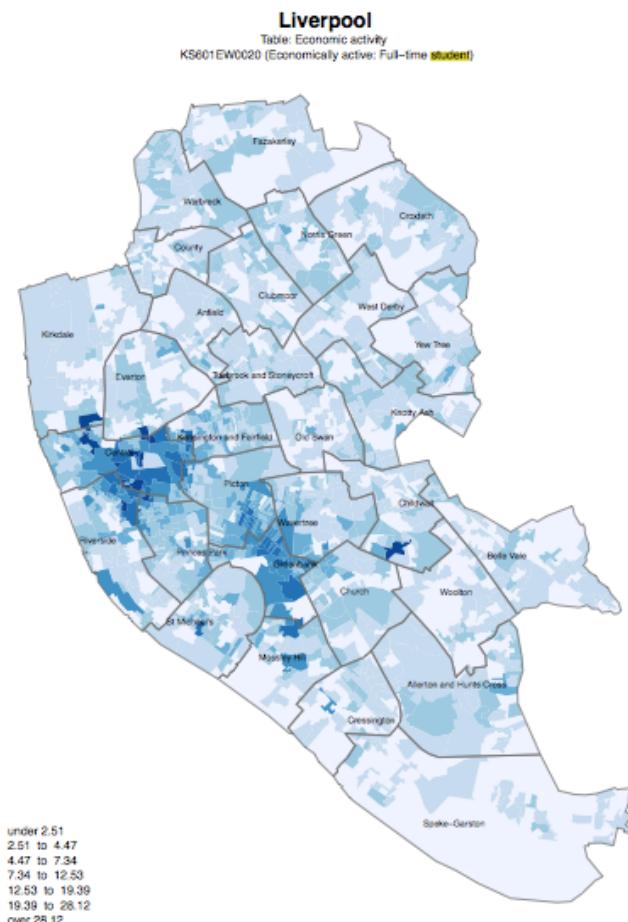


UNIVERSITY OF
LIVERPOOL

Overview and Structure

- R in Action
- R as GIS
- Basics of R for Mapping
- R studio

R in Action



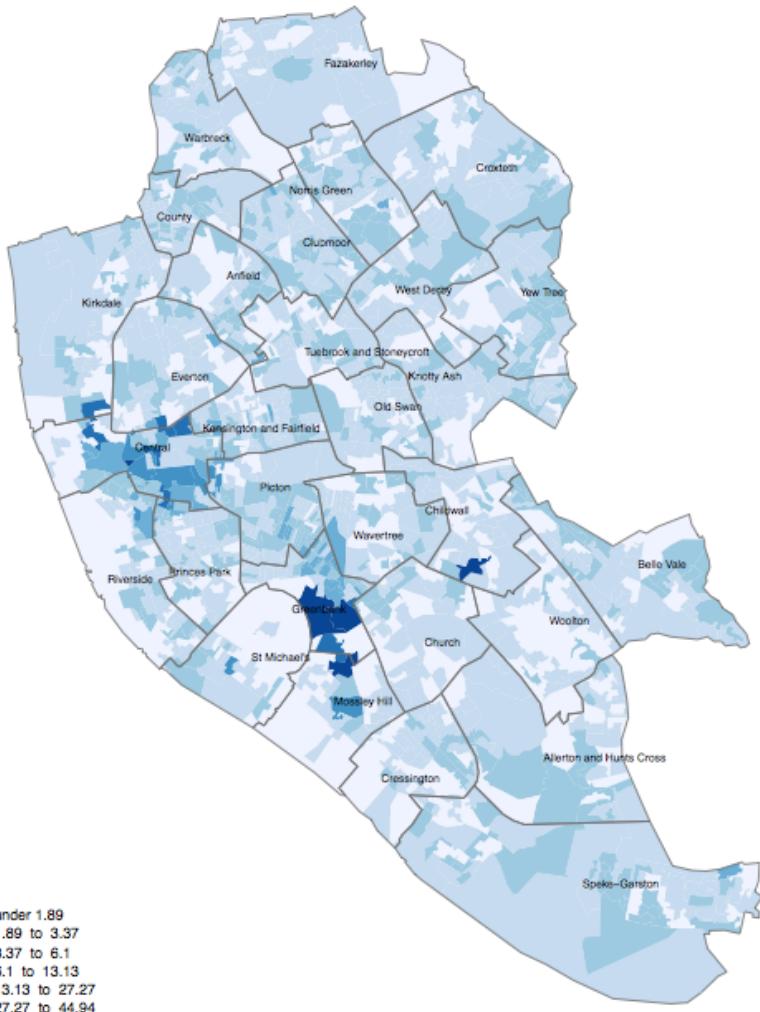
2011 Census Open Atlas



Alex Singleton (www.alex-singleton.com)
Version 1.0

Liverpool

Table: Age structure
KS102EW0026 (Age 18 to 19)



Map created by Alex Singleton (<http://www.alex-singleton.com>)

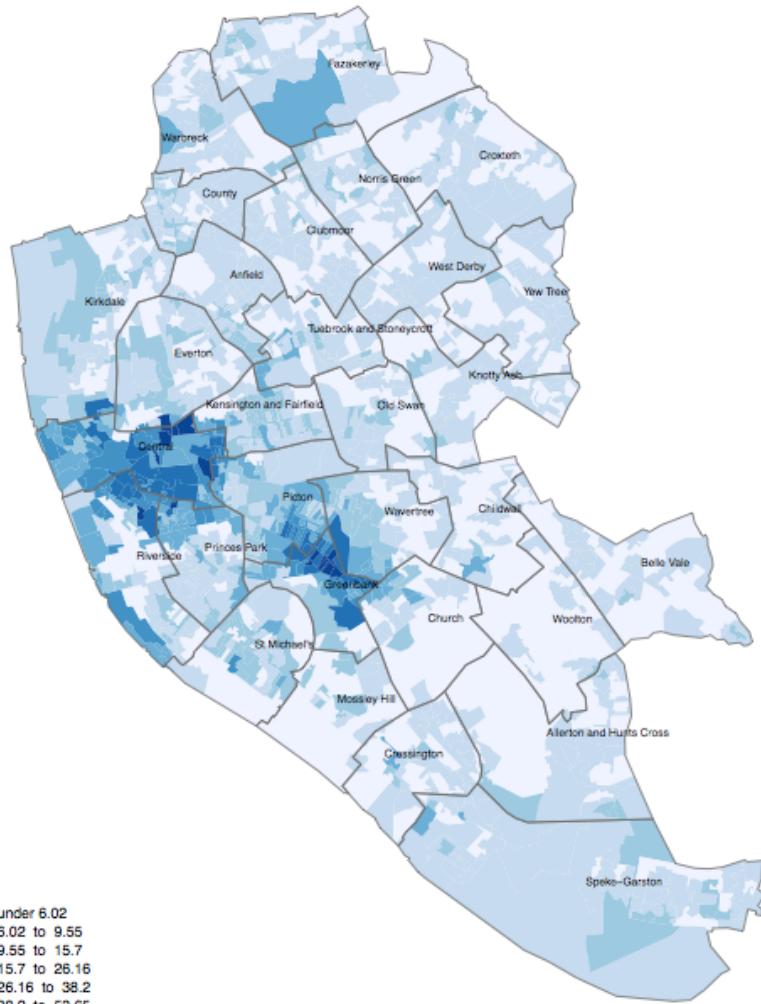
Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013



LIVERPOOL

Liverpool

Table: Age structure
KS102EW0027 (Age 20 to 24)

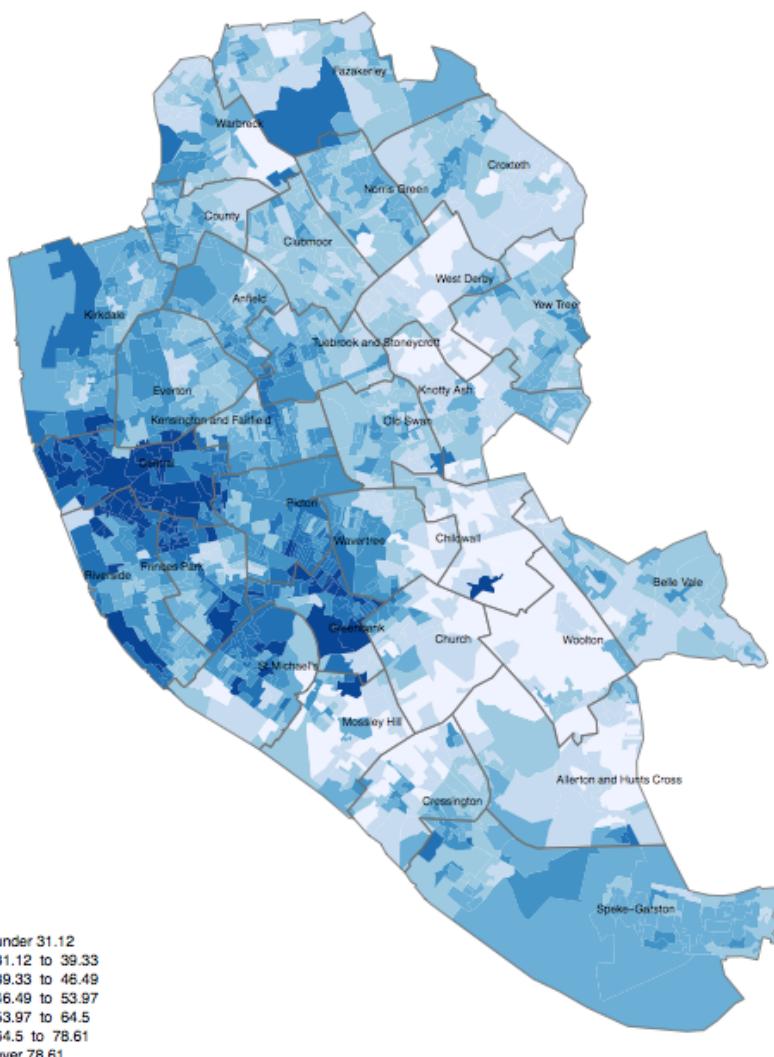


Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Liverpool

Table: Marital and civil partnership status
KS103EW0008 (Single (never married or never registered a same-sex civil partnership))

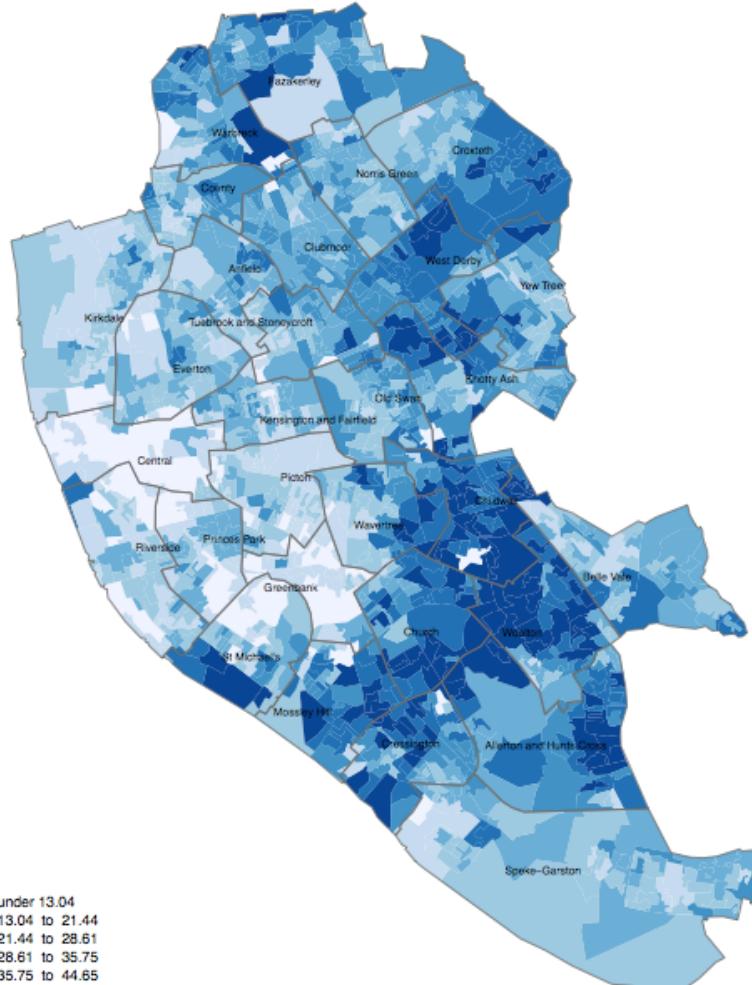


Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Liverpool

Table: Marital and civil partnership status
KS103EW0009 (Married)

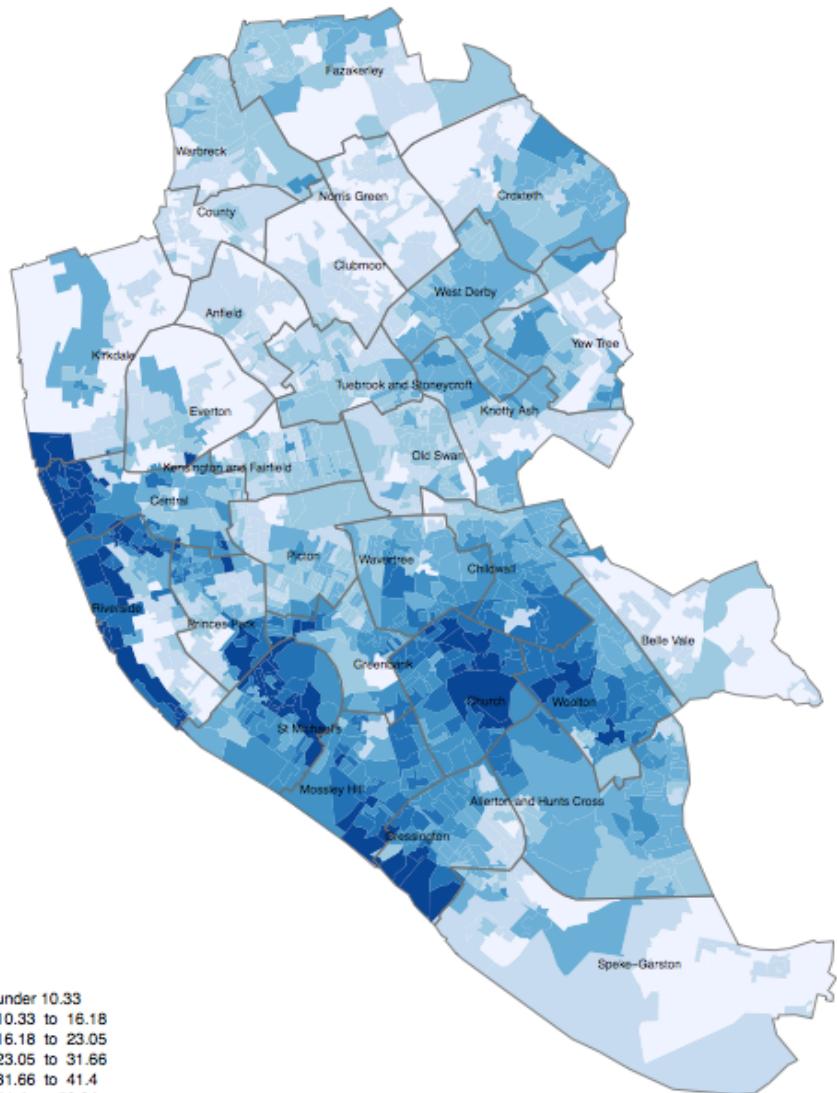


Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Liverpool

Table: Qualifications and students
 KS501EW0019 (Highest level of qualification: Level 4 qualifications and above)

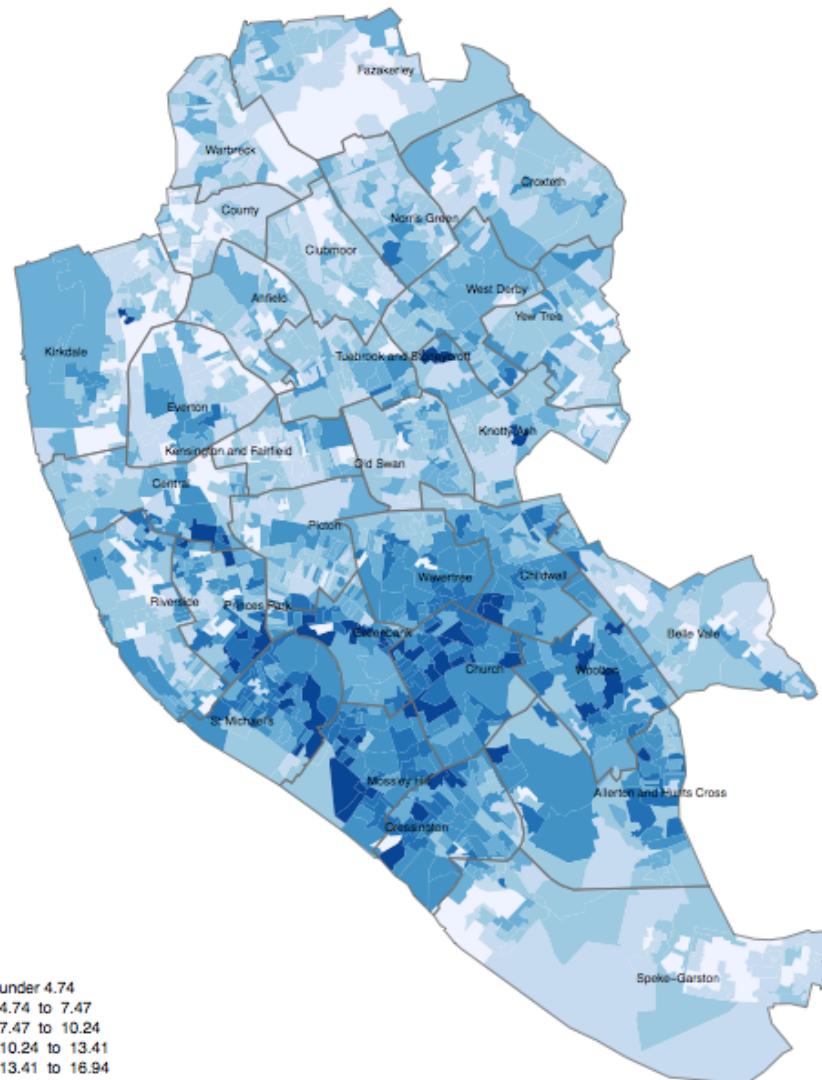


Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Liverpool

Table: Industry
 KS605EW0035 (P Education)



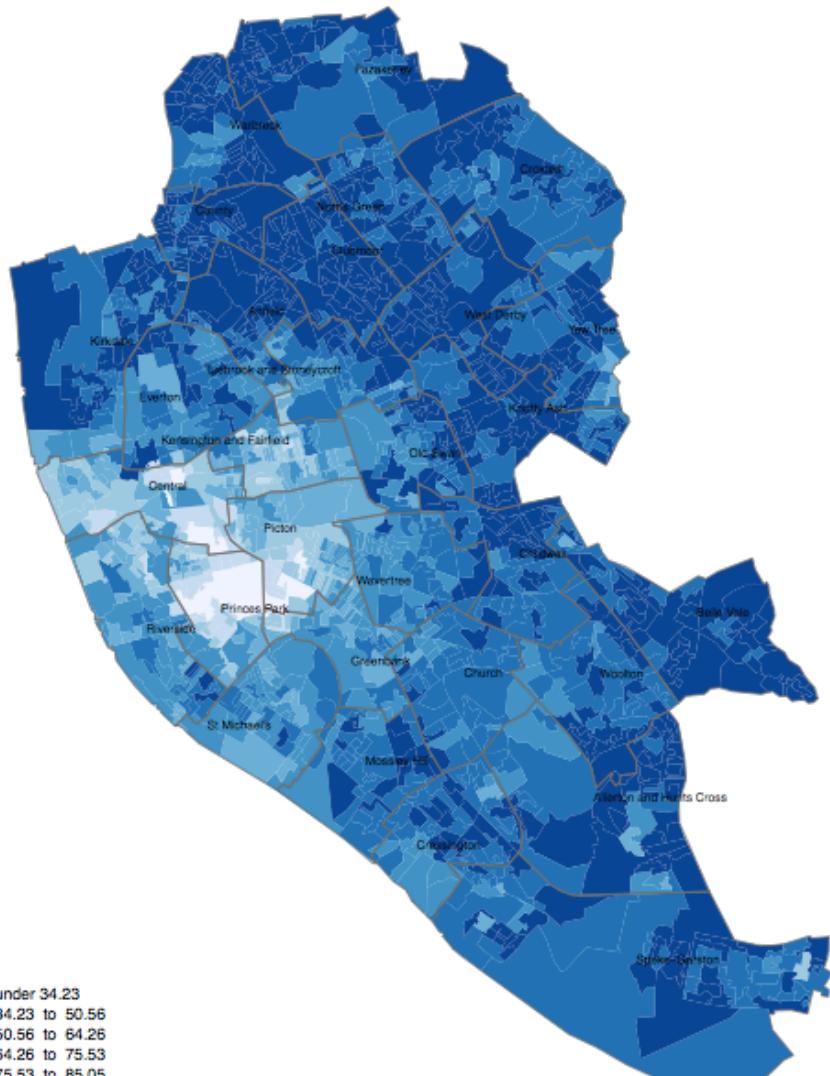
Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Liverpool

Table: Ethnic group

KS201EW0020 (White: English/Welsh/Scottish/Northern Irish/British)



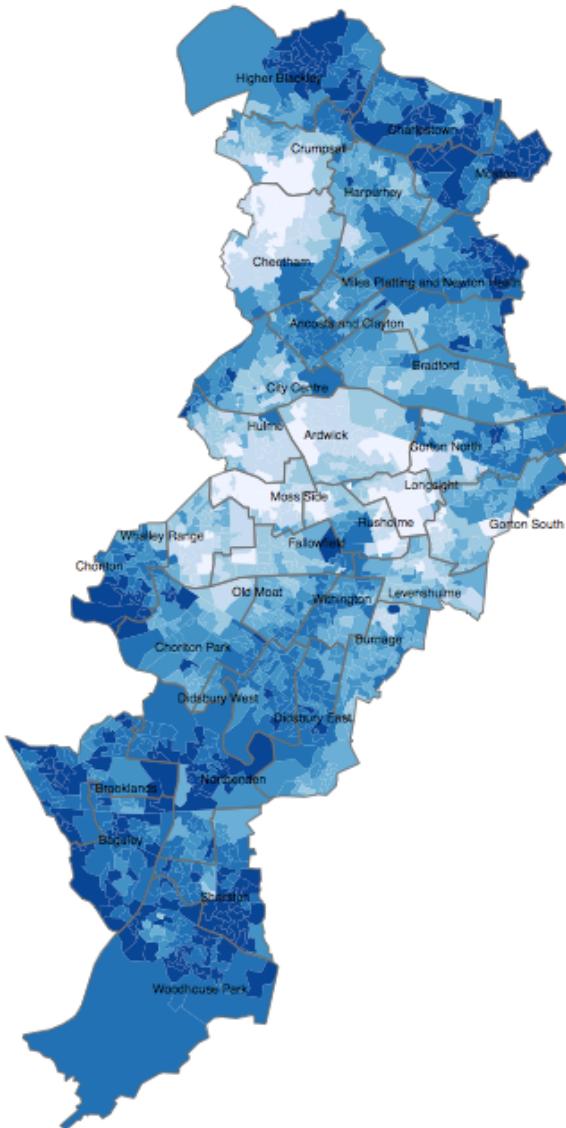
Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Manchester

Table: Ethnic group

KS201EW0020 (White: English/Welsh/Scottish/Northern Irish/British)



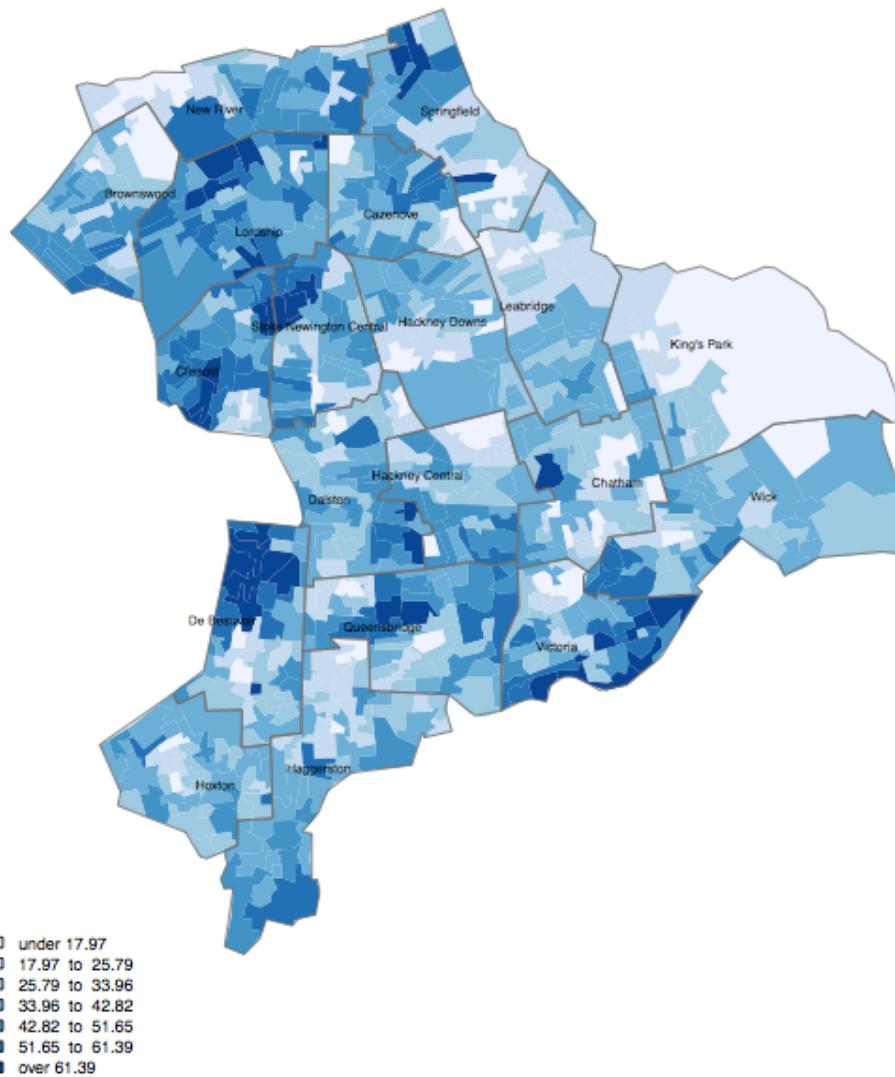
Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

Hackney

Table: Ethnic group

KS201EW0020 (White: English/Welsh/Scottish/Northern Irish/British)

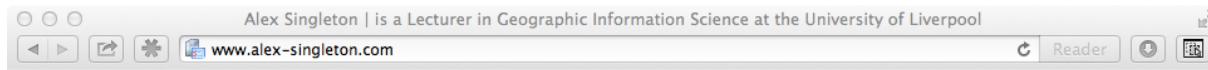


UNIVERSITY
LIVERPC

Map created by Alex Singleton (<http://www.alex-singleton.com>)

Contains National Statistics data © Crown copyright and database right 2013; Contains Ordnance Survey data © Crown copyright and database right 2013

R in Action



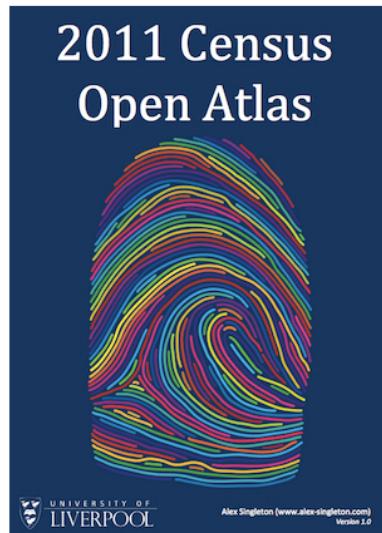
ALEX SINGLETON

is a Lecturer in Geographic Information Science at the University of Liverpool

[RESEARCH INTERESTS](#) [PAPERS](#) [BOOKS](#) [SOFTWARE](#) [CV](#) [CONTACT](#) [R TUTORIALS](#)

FEBRUARY 5, 2013

2011 CENSUS OPEN ATLAS PROJECT



This month has seen the release of the 2011 census data for England and Wales at Output Area Level.

This offers the possibility to map various attributes about people and places for very small geographic areas. Output Areas represent the most detailed geography for which Census data are released and are the building blocks for many popular products such as

geodemographic classifications.

Because the data and boundaries are available under an [open](#)

 Follow @alexsingleton

RECENT TWEETS

- Who Teaches Planning?
[j.mp/TYLvKT](#) 25 days ago
- Still time to join the team here in Liverpool:
[jobs.ac.uk/job/AFQ310/pos...](#)
- Please RT [30 days ago](#)
- Time of travel in the 1800s
[j.mp/13b9dWI](#) 34 days ago
- Nice combination of maps and graphs by Twitter!
[blog.twitter.com/2013/01/celebr...](#)
38 days ago
- Towards a Sociological Understanding of Social Media:
Theorizing Twitter [j.mp/YYqza9](#) 42 days ago
- [Follow Me on Twitter](#)

SOCIAL NETWORKS

R in Action

Rpubs – 2011 Open Atlas Project: Output Area Key Statistics
rpubs.com/alexsingleton/openatlas Sign in Register

2011 Open Atlas Project: Output Area Key Statistics

1. Download and consolidate 2011 Output Area Level Key Statistics

The first stage is to create a list of the tables that you want to download - these are constructed using the Nomis naming conventions for the download files - e.g. ks102ew_2011_oa for Key Statistics table 102 England and Wales at Output Area level.

```
# Create a list of tables
table_list <- c("KS101EW", "KS102EW", "KS103EW", "KS104EW", "KS105EW", "KS106EW",
  "KS107EW", "KS201EW", "KS202EW", "KS204EW", "KS205EW", "KS206EW", "KS207WA",
  "KS208WA", "KS209EW", "KS301EW", "KS401EW", "KS402EW", "KS403EW", "KS404EW",
  "KS405EW", "KS501EW", "KS502EW", "KS503EW", "KS504EW", "KS505EW",
  "KS606EW", "KS607EW", "KS608EW", "KS609EW", "KS610EW", "KS611EW", "KS612EW",
  "KS613EW")
```

```
table_list <- tolower(paste(table_list, "_2011_oa", sep = ""))
```

Download and unzip the files for each of the OA tables.

```
# Set a download folder for the census CSV files setwd('/Volumes/Macintosh
# HD 2/Dropbox/Projects/2011_Census/2011_Data')
setwd("/Volumes/Macintosh HD 2/Dropbox/Projects/2011_census/2011_Data")

# Download Files
for (n in 1:length(table_list)) {
  file <- as.character(table_list[n])
  temp <- tempfile(fileext = ".zip")
  download_file(paste("http://www.nomisweb.co.uk/output/census/2011/", file,
  ".zip", sep = ""), temp)
  unzip(temp, junkpaths = TRUE)
  unlink(temp)
  csv_file <- paste(toupper(sub("_2011_oa", "", file)), "DATA.CSV", sep = "")
  assign(file, read.csv(csv_file))
}
```

Next, compile a single object containing the table variable names and their corresponding descriptions.

```
variable_Desc <- NULL
for (n in 1:length(table_list)) {
  file <- as.character(table_list[n])
  csv_file <- paste(toupper(sub("_2011_oa", "", file)), "DESC0.CSV", sep = "")
  temp <- read.csv(csv_file)
  Variable_Desc <- rbind(Variable_Desc, temp)
  remove(temp)
}
```

Finally, we will create a single file containing all of the 2011 key statistics using the tables that cover both England and Wales.

2011 Open Atlas Project: Output Area Key Statistics by Alex Singleton Last updated 4 days ago Comments (0) Share Hide Toolbars

R in Action

- 2 lists –
 - A) All local authorities
 - B) All census variables – prepared for mapping
- Start Loop 1 – for local authority X in A
 - Start Loop 2 – for census variable Y in B
 - *Make a map for variable Y in local authority X*
 - Close Loop 2 when all Y in B are mapped
- Close Loop 1, and move onto next X in A
- End code when all X in A have been mapped

the guardian

Your search terms... News Search

News > Show and Tell

DATA STORE SHOW AND TELL.



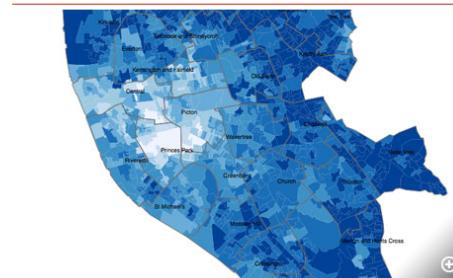
Previous

Blog home

Next

Mapping the census: how one man produced a library for all

Alex Singleton downloaded every single census dataset for every local authority in England - and then produced a free library of downloadable PDFs. Find out what he did
[More data journalism and data visualisations from the Guardian](#)



Census in Liverpool mapped by the Open Atlas project. Click image to embiggen

3

[Share](#)

15

[Tweet](#)

0

[+1](#)

0

[Share](#)



Posted by
 Simon Rogers
 Friday 8 February 2013
 07.30 GMT
[guardian.co.uk](#)
[Jump to comments \(0\)](#)



Article history

UK news

Census

World news

Population

Society

Local government

Series

Show and Tell

More from Datablog on

UK news

Census

World news

Population

Society

Local government

Series

Show and Tell

More blogposts

Great Flight Times

22kg Bag Allowance*

Allocated Seats

Great Package Holidays

*charges apply

BOOK NOW

Holland.

jet2.com
Friendly low fares

Today's best video



Naples driver's u-turn fiasco causes holdup
 A driver attempts a 180-degree turn on a narrow street



Leicester's curry houses: a taste of little India

Marcel Theroux visits

Bobby's restaurant for a

lesson in Gujarati

cooking

75 comments

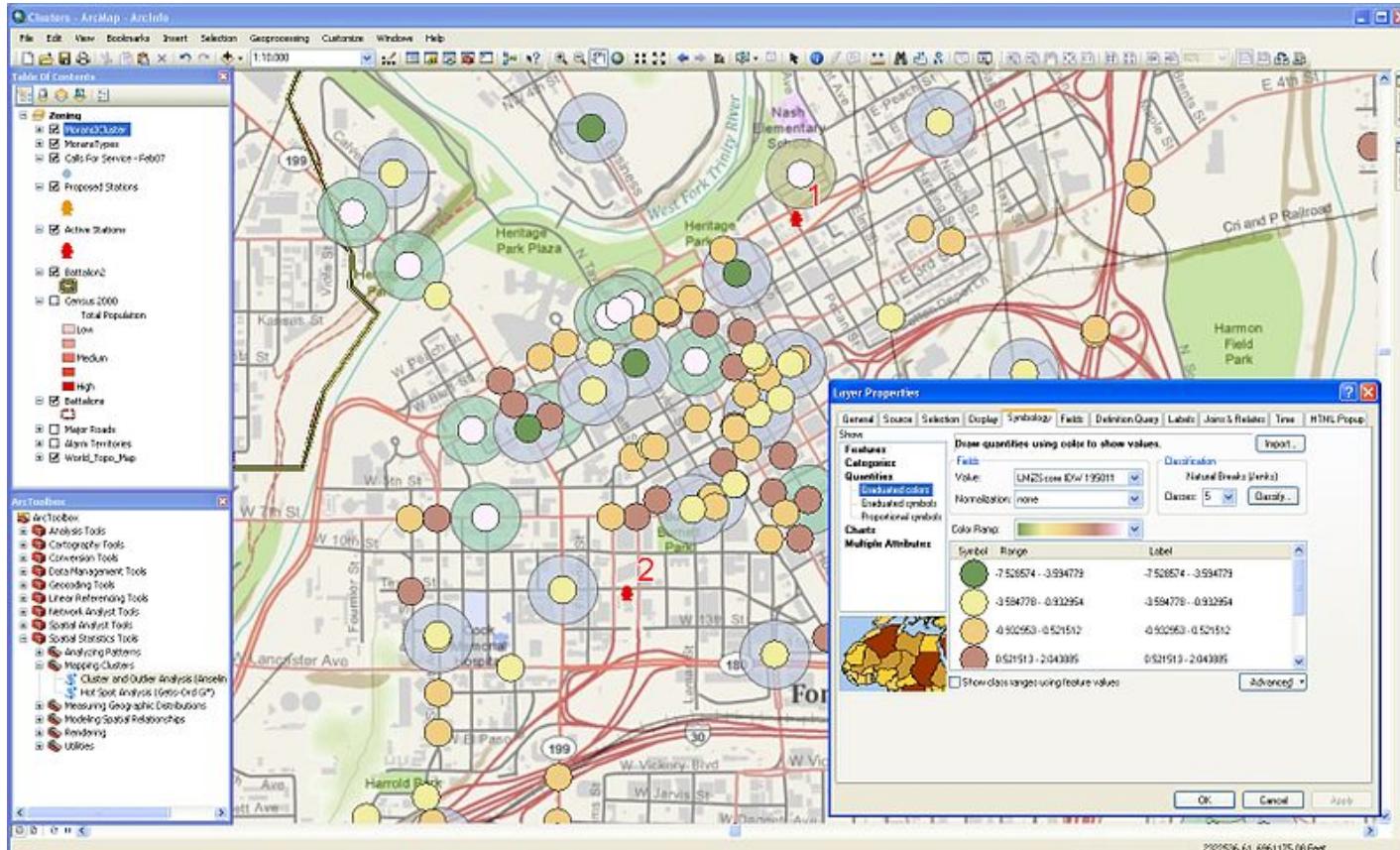


UNIVERSITY OF
 LIVERPOOL

GIS: Geographic Information Systems

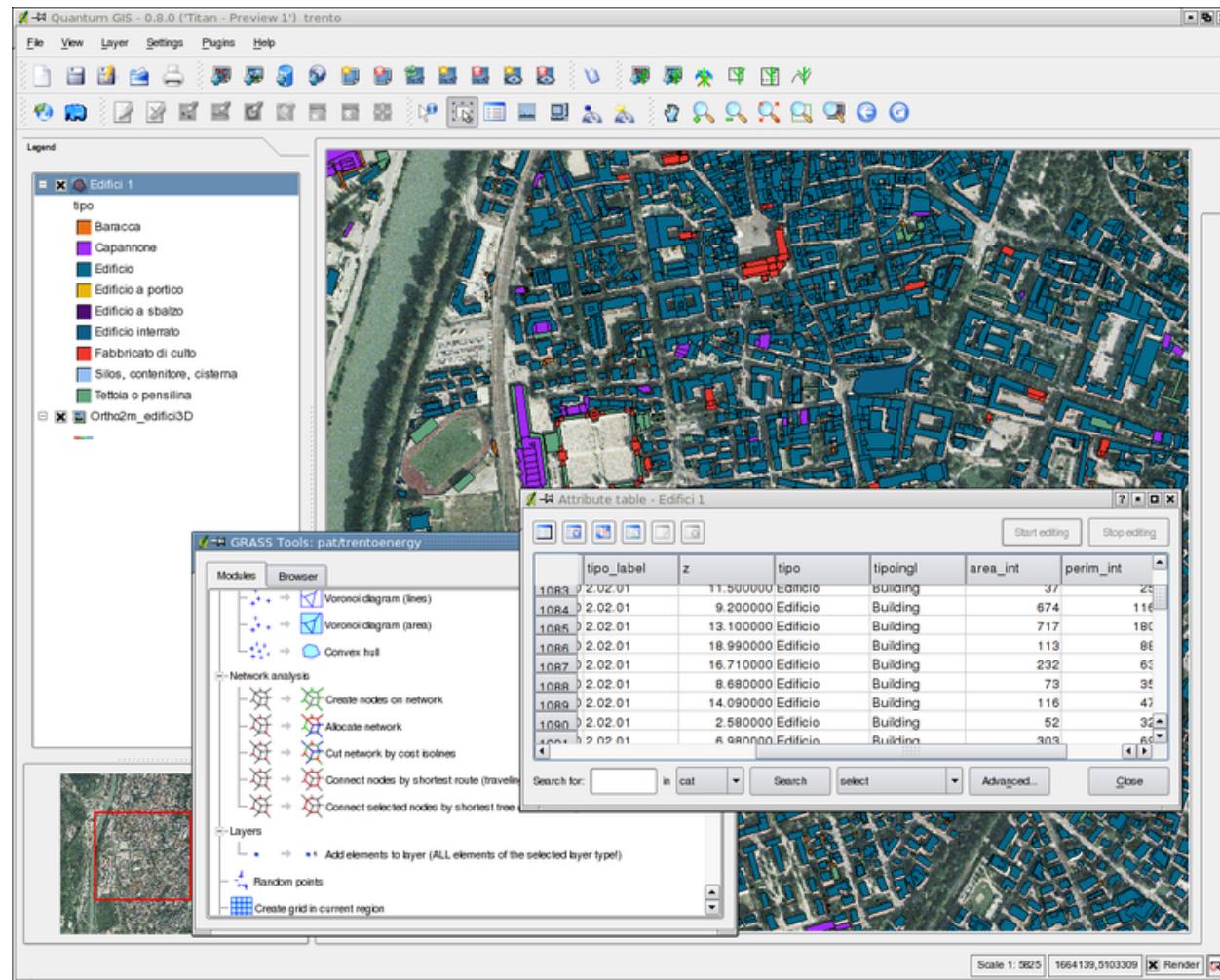
- Almost all human activities and decisions involve an important **geographic component**
- Working with geographic information involves **unique, complex and difficult choices**
 - Why we need specialized “software”
 - **organize** and **store**
 - **access** and **retrieve**
 - **manipulate** and **synthesize**
 - apply to the solution of **problems**

GIS: Geographic Information Systems

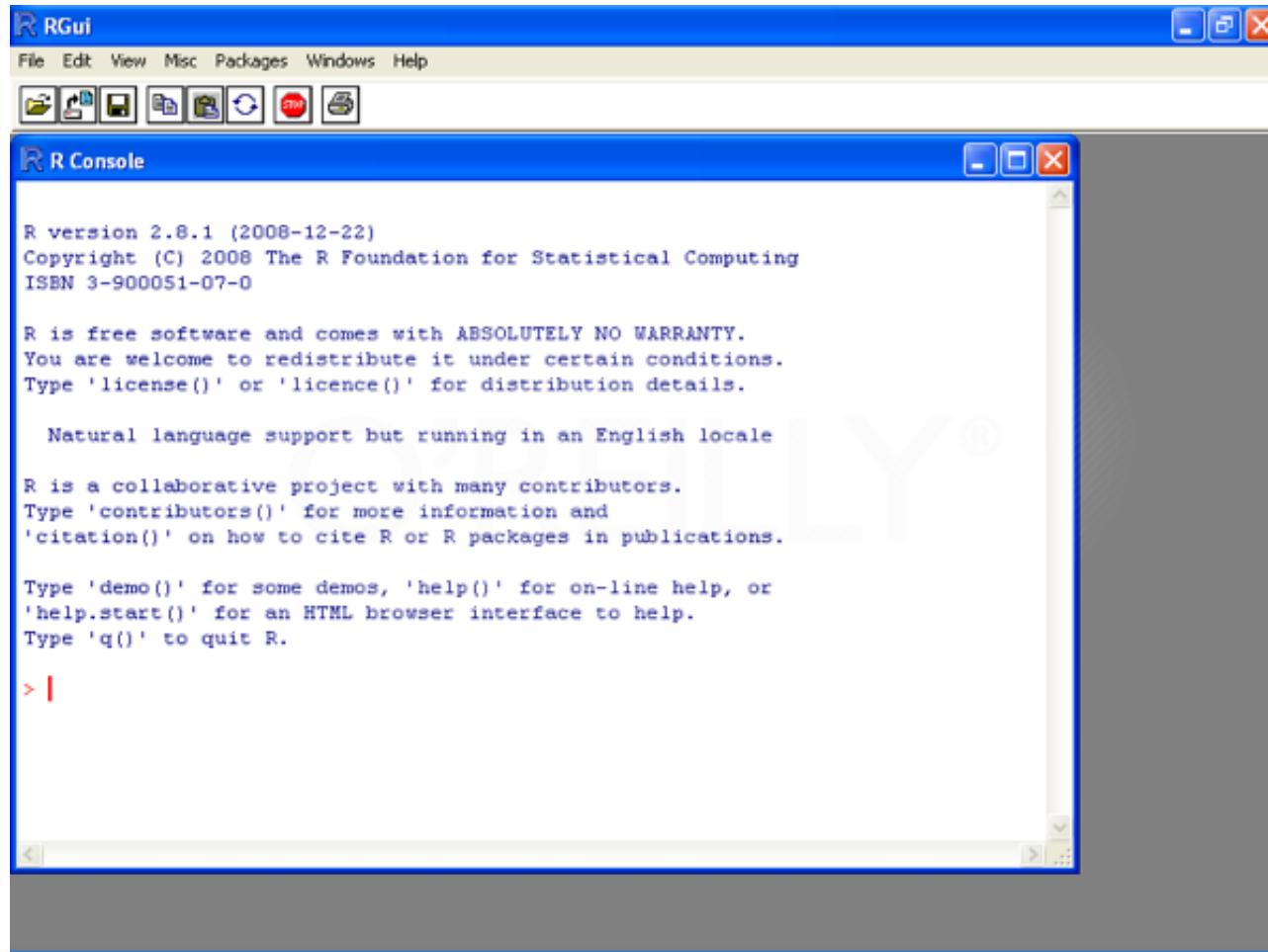


<http://en.wikipedia.org/wiki/File:Arcgisclusters.jpg>

GIS: Geographic Information Systems



GIS: Geographic Information Systems



GIS: Geographic Information Systems

The screenshot shows a web browser displaying the OpenGeo : PostGIS website at opengeo.org/technology/postgis/. The page features a dark header with the OpenGeo logo and navigation links for Products & Services, Technology, Support, Partners, About, and Blog. Below the header, a main content area includes a sidebar with 'PostGIS Resources' (Community Site, Issue Tracker, SVN Repository, Spatial Database Tips & Tricks, Chat Room) and 'OpenGeo Services' (PostGIS training). The main content area has sections for 'About PostGIS' (describing spatial enablement of PostgreSQL), 'PostGIS Features' (listing benefits like high performance, SQL compliance, and spatial operations), and 'PostGIS Support' (listing client/server software). Two examples are shown at the bottom: 'Farallon: San Francisco Enterprise Addressing System' (a map with red circles and address details) and 'Landgate' (a map showing a gas station with edit options).

About PostGIS

PostGIS "spatially enables" the PostgreSQL open source relational database. The database can then be used to store and query spatial data (points, lines and polygons).

 **PostGIS**

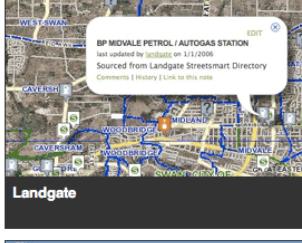
- High performance, robust spatial database built on PostgreSQL
- Simple Features for SQL (SFSQL) compliance
- Proven reliability and transactional integrity (ACID compliance)
- Provides spatial representations of `geometry` types (points, lines, polygons)
- Support for common and advanced spatial operations such as geometry creation and conversion, reprojection, buffer, convex hull, generalization, union, and more
- Geodetic support for measurements across the globe/dateline
- Command-line and graphical tools for flexible management

PostGIS is widely supported as a spatial database back-end to client and server software, including:

- **Open Source Server:** GeoServer, Mapserver, Mapnik, DeeGree, SharpMap
- **Open Source Desktop:** GRASS, QGIS, uDig, gvSIG
- **Proprietary Server:** ArcServer, Ionic Enterprise, MapDotNet Server
- **Proprietary Desktop:** ArcGIS, Manifold, Safe FME, CadCorp SIS, MapInfo Professional

Real-world Examples


Farallon: San Francisco Enterprise Addressing System


Landgate

GIS: Geographic Information Systems



The screenshot shows the GDAL (Geospatial Data Abstraction Library) website. The header reads "GDAL: GDAL – Geospatial Data Abstraction Library". The navigation bar includes links for "Main Page", "Related Pages", "Classes", and "Files". A "Reader" button is also present. Below the header, the title "GDAL – Geospatial Data Abstraction Library" is displayed. A "Select language" dropdown menu offers options in English, Russian, Portuguese, and French/Français. The main content area starts with a brief introduction about GDAL being a translator library for raster geospatial data formats. It mentions the X/MIT style Open Source license by the Open Source Geospatial Foundation, a single abstract data model, and various commandline utilities for data translation and processing. It also notes the NEWS page for the October 2012 GDAL/OGR 1.9.2 release. A note states that the related OGR library provides a similar capability for simple features vector data. URLs for the master site (<http://www.gdal.org>) and download (<ftp://remotesensing.org>, <http://download.osgeo.org>) are provided. A section titled "User Oriented Documentation" lists links to Wiki, Downloads, Supported Formats, Utility Programs, FAQ, Data Model, Governance, Service Provider Listings, Sponsors, and Software Using GDAL. Another section titled "Developer Oriented Documentation" lists links to Building GDAL From Source, Downloads, API Reference Documentation, GDAL API Tutorial, GDAL Driver Implementation Tutorial, GDAL Warp API Tutorial, OGRSpatialReference Tutorial, GDAL C API, GDAL Algorithms C API, GDALDataset C++ API, GDALRasterBand C++ API, and GDAL for Windows CE. A "Mailing List" section describes the gdal-announce mailing list as a low volume way to keep track of major developments. It explains that the gdal-dev@lists.osgeo.org mailing list is used for discussion of development and user issues, with subscriptions available on the web and archives on Nabble. It also mentions the gdal IRC channel on irc.freenode.net.

GDAL: GDAL – Geospatial Data Abstraction Library

Main Page Related Pages Classes Files Reader

GDAL – Geospatial Data Abstraction Library

Select language: [English][Russian][Portuguese][French/Français]

GDAL is a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a **single abstract data model** to the calling application for all supported formats. It also comes with a variety of useful **commandline utilities** for data translation and processing. The **NEWS** page describes the October 2012 GDAL/OGR 1.9.2 release.

The related OGR library (which lives within the GDAL source tree) provides a similar capability for simple features vector data.

Master: <http://www.gdal.org>
Download: <ftp://remotesensing.org>, <http://download.osgeo.org>

User Oriented Documentation

- Wiki – Various user and developer contributed documentation and hints
- Downloads – Ready to use binaries (executables)
- Supported Formats : GeoTIFF, Erdas Imagine, SDTS, ECW, MrSID, JPEG2000, DTED, NITF, ...
- GDAL Utility Programs : gdalinfo, gdal_translate, gdaladdo, gdalwarp, ...
- GDAL FAQ
- GDAL Data Model
- GDAL/OGR Governance and Community Participation
- GDAL Service Provider Listings (not vetted)
- Sponsors, Acknowledgements and Credits
- Software Using GDAL

Developer Oriented Documentation

- Building GDAL From Source
- Downloads – source code
- API Reference Documentation
- GDAL API Tutorial
- GDAL Driver Implementation Tutorial
- GDAL Warp API Tutorial
- OGRSpatialReference Tutorial
- GDAL C API
- GDAL Algorithms C API
- GDALDataset C++ API
- GDALRasterBand C++ API
- GDAL for Windows CE

Mailing List

A gdal-announce mailing list subscription is a low volume way of keeping track of major developments with the GDAL/OGR project.

The gdal-dev@lists.osgeo.org mailing list can be used for discussion of development and user issues related to GDAL and related technologies. Subscriptions can be done, and archives reviewed on the web. The mailing list is also available in read-only format by NNTP at <news://news.gmane.org/gmane.comp.gis.gdal.devel> and by HTTP at <http://news.gmane.org/gmane.comp.gis.gdal.devel>. Archives since 2005 are searchable on Nabble.

Some GDAL/OGR users and developers can also often be found in the gdal IRC channel on [irc.freenode.net](irc://irc.freenode.net).

GIS: Geographic Information Systems

pysal - Python Spatial Analysis Library - Google Project Hosting

alexandersingleton@gmail.com | My favorites | Profile | Sign out

 pysal
Python Spatial Analysis Library

Project Home Downloads Wiki Issues Source

Summary People

Project Information

+13 Recommend this on Google

Starred by 74 users

[Project feeds](#)

Code license [New BSD License](#)

Labels Python, GIS, Geo, Spatial, Analysis

 Members
sjrey.lanse...@gmail.com
7 committers
4 contributors

Featured

 Downloads
[pysal-1.5.0-docs.zip](#)
[pysal-1.5.0.tar.gz](#)
[pysal-1.5.0.win32.exe](#)
[pysal-1.5.0.zip](#)
[Show all »](#)

Links

External links [@geodacenter](#)

Groups

[Users](#) [Developers](#)

Welcome

PySAL is an open source cross-platform library of spatial analysis functions written in Python. It is intended to support the development of high level applications for spatial analysis.

Documentation

[PySAL 1.5](#), released 2013 01 31

All Versions

- [Unreleased development version](#)
- [PySAL 1.4](#), released 2012 07 31
- [PySAL 1.3](#), released 2012 01 31
- [PySAL 1.2](#), released 2011 07 31
- [PySAL 1.1](#), released 2011 01 31
- [PySAL 1.0](#), released 2010 08 01

News

2013-01-30 [PySAL 1.5 Stable released \(Downloads\)](#)

2012-10-24 [PySAL short course at OSGRS 2012](#)

2012-09-18 [PySAL short course at GIScience 2012](#)

2012-07-31 [PySAL 1.4 released](#)

2012-07-30 [Short course on PySAL for Spatial Regression](#)

2012-07-30 [PySAL presentation at Joint Statistical Meetings](#)

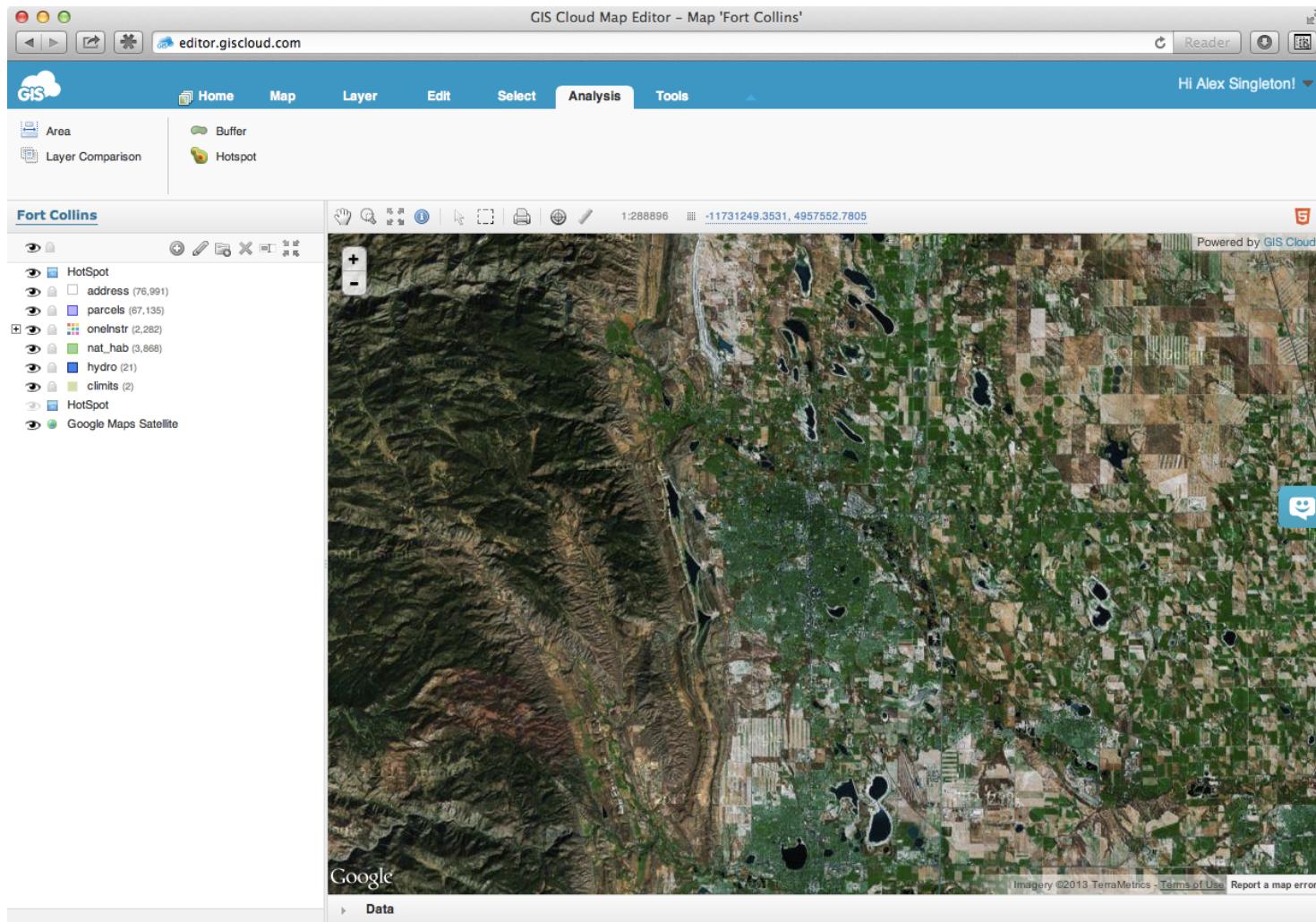
2012-07-18 [PySAL at SciPy 2012](#)

2012-01-31 [PySAL 1.3 released](#)

2012-01-19 [PySAL 1.3 release code sprint on January 23](#)

[Older News](#)

GIS: Geographic Information Systems



Databases & Joins

- Organized / structured collection of data
- Tables hold data
- Relationships between tables
- E.g...

The diagram illustrates a many-to-one relationship between the Person and Location tables. The Person table has five rows with Person IDs 1 through 5. The Location table has five rows with Location IDs 1 through 5, corresponding to Local Authorities Liverpool, Camden, Milton Keynes, Manchester, and Sefton respectively. Red arrows connect the Location ID column of the Person table to the Location ID column of the Location table, indicating that multiple rows in the Person table can map to a single row in the Location table.

| Person ID | Surname | Location ID | Location ID | Local Authority |
|-----------|-----------|-------------|-------------|-----------------|
| 1 | Thomas | 1 | 1 | Liverpool |
| 2 | Carl | 1 | 2 | Camden |
| 3 | Constance | 2 | 3 | Milton Keynes |
| 4 | Frank | 2 | 4 | Manchester |
| 5 | Billy | 4 | 5 | Sefton |

| Person ID | Surname | Location ID | Local Authority |
|-----------|-----------|-------------|-----------------|
| 1 | Thomas | 1 | Liverpool |
| 2 | Carl | 1 | Liverpool |
| 3 | Constance | 2 | Camden |
| 4 | Frank | 2 | Camden |
| 5 | Billy | 4 | Manchester |

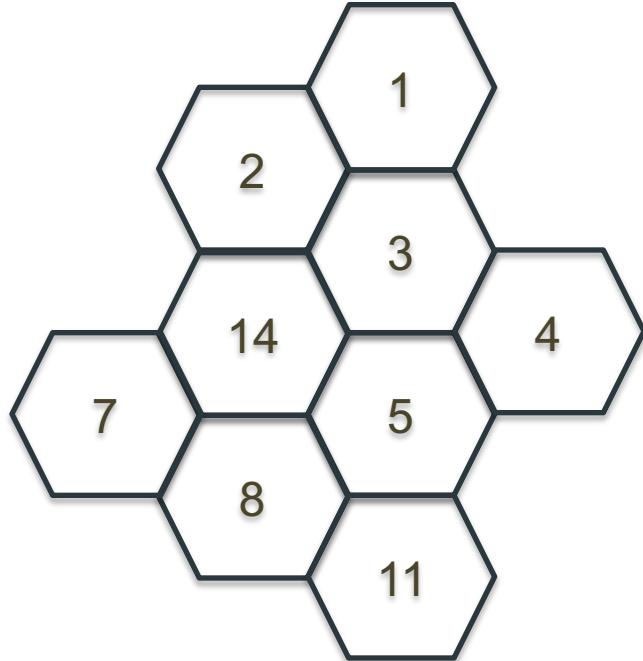
Merging data frames in R

```
New_Object<- merge(object1,object2,  
by.x="variable",by.y="variable", in.x=TRUE)
```

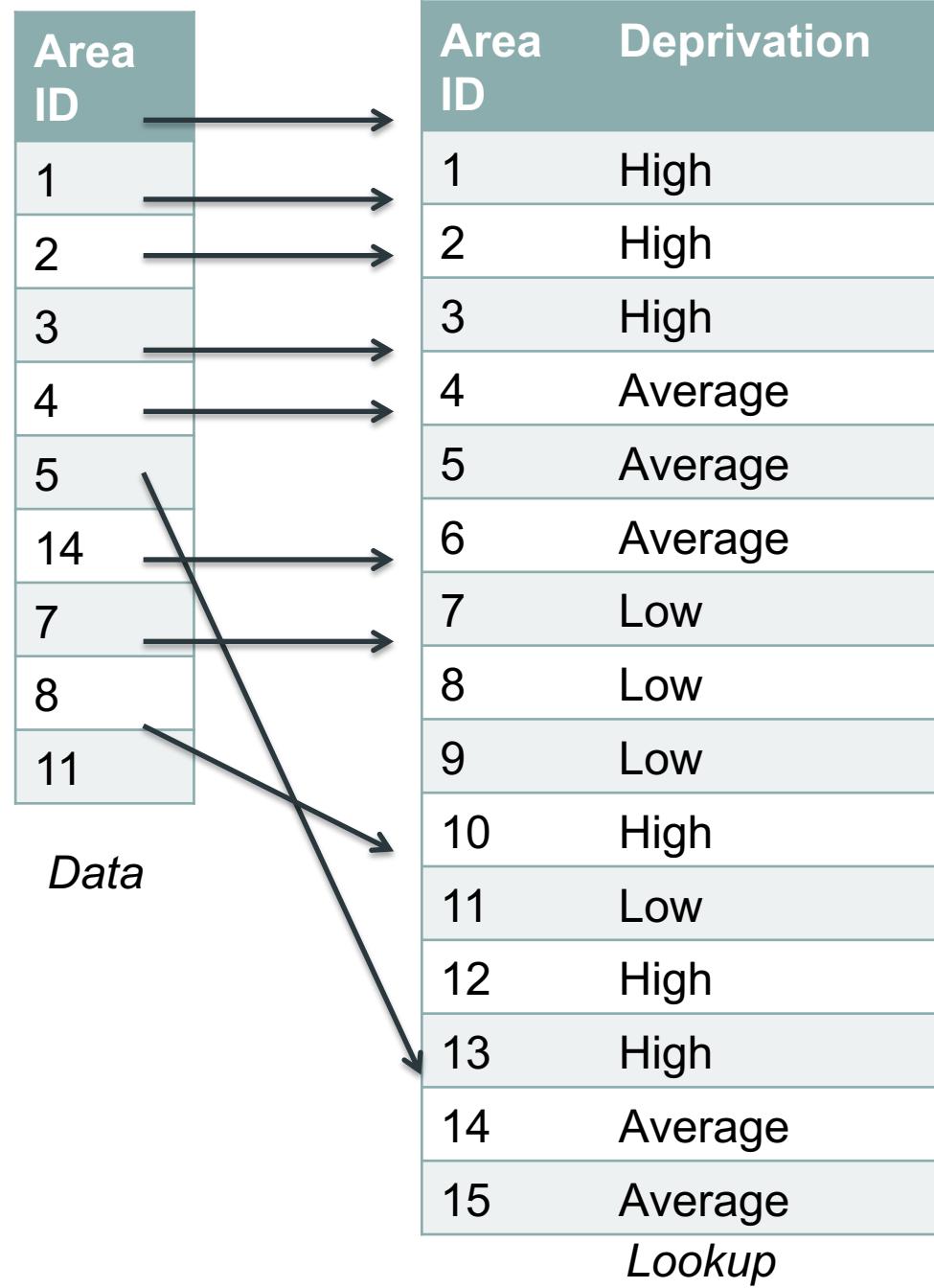
| Person ID | Surname | Location ID |
|-----------|-----------|-------------|
| 1 | Thomas | 1 |
| 2 | Carl | 1 |
| 3 | Constance | 2 |
| 4 | Frank | 2 |
| 5 | Billy | 4 |

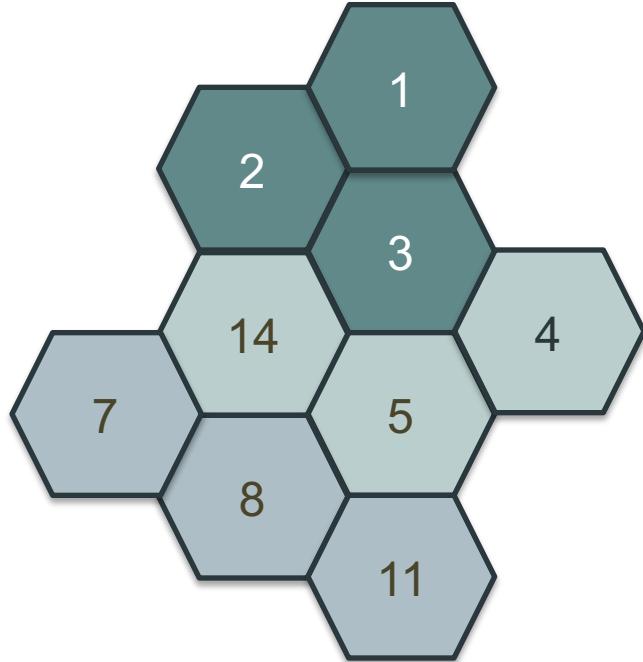
| Location ID | Local Authority |
|-------------|-----------------|
| 1 | Liverpool |
| 2 | Camden |
| 3 | Milton Keynes |
| 4 | Manchester |
| 5 | Sefton |

```
New_Object<- merge(People_Table,Places_Table,  
by.x="Location ID",by.y="Location ID", in.x=TRUE)
```



Polygons



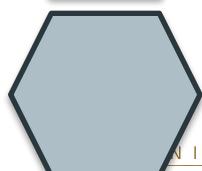
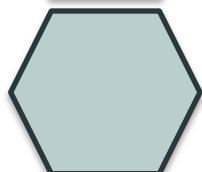
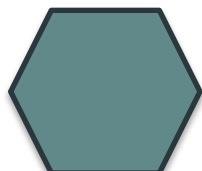


High Deprivation

Average Deprivation

Low Deprivation

| Area ID | Deprivation |
|---------|-------------|
| 1 | High |
| 2 | High |
| 3 | High |
| 4 | Average |
| 5 | Average |
| 14 | Average |
| 7 | Low |
| 8 | Low |
| 11 | Low |



Adding data to polygons in R

```
LSOA <- readShapePoly("LSOA/england_low_soa_2001.shp")
Plot(LSOA)
```



Adding data to polygons in R

```
head(LSOA@data)
```

```
- zonecode
 0 E01006739
 1 E01006687
 2 E01006741
 3 E01006743
 4 E01006528
 5 E01006684
```

```
# Order the LSOA spatial polygon data frame
LSOA <- LSOA[order(LSOA$zonecode), ]
# Order the CT_2001_2011_Liverpool data frame
CT_2001_2011_Liverpool <-
  CT_2001_2011_Liverpool[order(CT_2001_2011_Liverpool$LSOA_CODE), ]
# Join based on ordering
LSOA@data <- cbind(LSOA@data, CT_2001_2011_Liverpool,
POP_2001_2010_Liverpool)
```

Adding data to polygons in R

These two methods are suitable where the rows in the object are different
– e.g. appending a national lookup table

```
# Alternative method 1 (this sometimes causes errors!)
```

```
LSOA@data <- merge(LSOA@data, CT_2001_2011_Liverpool  
, by.x="zonecode", by.y="LSOA_CODE", in.x=TRUE)
```

```
# Alternative method 2
```

```
LSOA@data = data.frame(LSOA@data,  
CT_2001_2011_Liverpool[match(LSOA@data[, "zonecode"] ,  
CT_2001_2011_Liverpool[, "LSOA_CODE"] ) , ] )
```

```
match(LSOA@data[, "zonecode"] ,CT_2001_2011_Liverpool[, "LSOA_CODE"]  
)
```

Making Maps in R

- Create breaks
- Define a colour scheme
- Apply breaks to create colour categories for each area

Making Maps in R

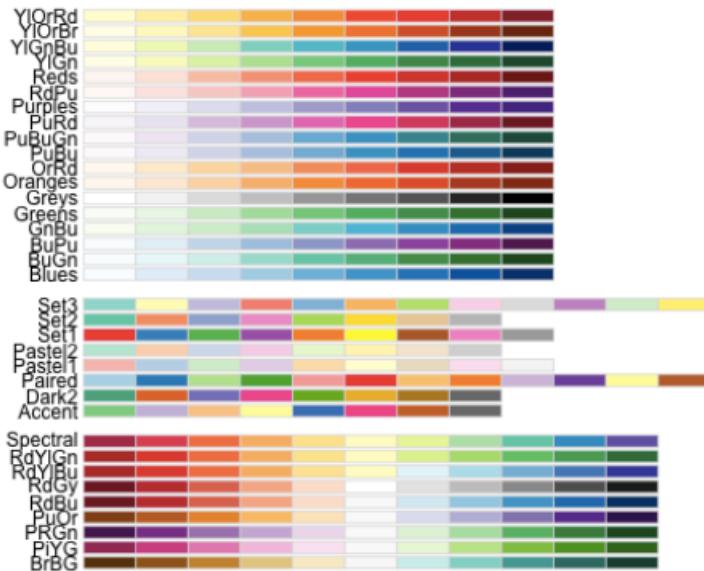
- For a continuous variable, find a sensible set of breaks that can be used to divide up the values into categories

```
#set breaks for 6 categories and then add the values to a single list - style = fisher is jenks  
breaks <- classIntervals(LSOA@data$TotPop_PC, n = 6, style = "fisher")  
breaks
```

```
[1] -79.007 -44.874 -12.964     4.688  30.600  73.195 127.698
```

Making Maps in R

- Choose a colour pallet using color brewer, or, make up your own list.



display.brewer.all()

The screenshot shows the Color Brewer 2.0 website interface. The top navigation bar includes links for "how to use", "updates", and "credits". The main title is "COLORBREWER 2.0" with the subtitle "color advice for cartography".

The interface allows users to specify the "number of data classes on your map" (set to 3) and the "nature of your data" (set to sequential). It provides a grid of color schemes under "pick a color scheme: BuGn", categorized as "multihue" and "single hue".

Optional filters include "colorblind safe", "print friendly", and "photocopy-able".

The "pick a color system" section shows color hex codes (e.g., 229, 245, 249; 153, 216, 201; 44, 162, 95) and options to "adjust map context" (roads, cities, borders), "select a background" (solid color or terrain), and "color transparency".

On the right, there is a map of the United States where each state is colored according to a sequential color scheme. A "SCORE CARD" is visible on the far right. At the bottom, there are "EXPORT YOUR COLORS >>" buttons and copyright information: "© Cynthia Brewer, Mark Harrower and The Pennsylvania State University Support Back to ColorBrewer 1.0".

Making Maps in R

```
# Select six colours from the pallet YlOrRd  
my_colours <- brewer.pal(6, "YlOrRd")  
# Print the colour codes to the terminal  
my_colours
```

```
> "#FFFFB2" "#FED976" "#FEB24C" "#FD8D3C" "#F03B20" "#BD0026"
```



http://en.wikipedia.org/wiki/Web_colors

<http://www.colorhexa.com>

<http://research.stowers-institute.org/efg/R/Color/Chart/ColorChart.pdf>

Making Maps in R

- Remember the list my_colours...
 - "#FFFFB2" "#FED976" "#FEB24C" "#FD8D3C"
"#F03B20" "#BD0026"
- Also that you can refer to items of the list with the square brackets []
 - E.g. my_colours[2] – second item in list

```
# Returns a list of colours  
my_colours[findInterval(LSOA@data$TotPop_PC, breaks,  
all.inside = TRUE)]
```

```
[1] "#BD0026" "#F03B20" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#F03B20"
[8] "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C"
[15] "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C"
[22] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#F03B20" "#FEB24C"
[29] "#FEB24C" "#FEB24C" "#FEB24C" "#FED976" "#FEB24C" "#FEB24C" "#FEB24C"
[36] "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FD8D3C" "#FD8D3C" "#FD8D3C"
[43] "#FD8D3C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FED976" "#FFFFB2"
[50] "#FEB24C" "#FFFFB2" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[57] "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[64] "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FEB24C"
[71] "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FD8D3C"
[78] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[85] "#FEB24C" "#FEB24C" "#FEB24C" "#FED976" "#FED976" "#FEB24C" "#FED976"
[92] "#FFFFB2" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C"
[99] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[106] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FED976"
[113] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C"
[120] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#F03B20" "#FEB24C" "#FEB24C"
[127] "#FD8D3C" "#FED976" "#FEB24C" "#FD8D3C" "#F03B20" "#FED976" "#FEB24C"
[134] "#FD8D3C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FD8D3C" "#BD0026" "#BD0026"
[141] "#FD8D3C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FEB24C"
[148] "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FED976" "#FD8D3C"
[155] "#FD8D3C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FEB24C"
[162] "#FD8D3C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FD8D3C" "#FED976" "#FD8D3C"
[169] "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#F03B20" "#FEB24C"
[176] "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[183] "#FEB24C" "#FD8D3C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C"
[190] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FED976" "#FEB24C" "#FEB24C"
[197] "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
[204] "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#F03B20"
[211] "#FD8D3C" "#FEB24C" "#FEB24C" "#FED976" "#FEB24C" "#FD8D3C" "#FED976"
[218] "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C"
[225] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FED976" "#FEB24C"
[232] "#FEB24C" "#F03B20" "#FEB24C" "#FD8D3C" "#FEB24C" "#BD0026" "#FEB24C"
[239] "#FFFFB2" "#FED976" "#BD0026" "#FED976" "#FEB24C" "#FEB24C" "#FEB24C"
[246] "#FD8D3C" "#FEB24C" "#FEB24C" "#FED976" "#FD8D3C" "#FEB24C" "#FD8D3C"
[253] "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C"
[260] "#FED976" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C"
[267] "#FED976" "#FEB24C" "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C"
[274] "#FEB24C" "#FD8D3C" "#FD8D3C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C"
[281] "#FEB24C" "#FEB24C" "#FD8D3C" "#FEB24C" "#FEB24C" "#FEB24C" "#FEB24C"
```



Making Maps in R

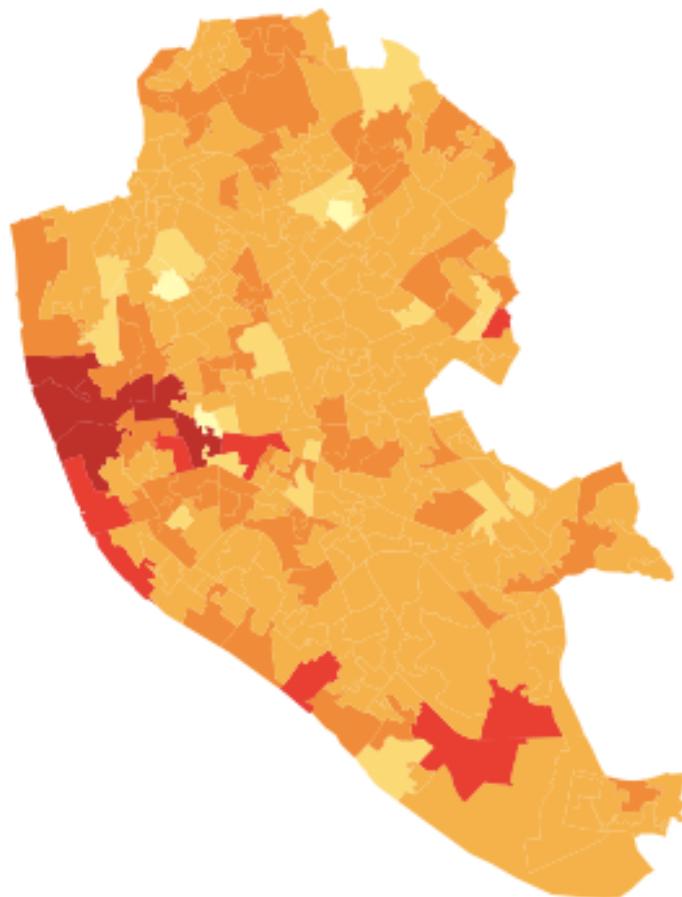
- Create a basic map with `plot()`

```
#Plot colours  
plot_colours <-  
my_colours[findInterval(LSOA@data$TotPop_PC, breaks,  
all.inside = TRUE)]  
  
# Create a basic choropleth map  
plot(LSOA, col = plot_colours, axes = FALSE, border = NA)
```



Polygon colour list

Making Maps in R

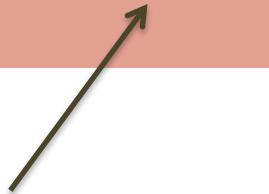


- Missing
 - Labels / context
 - North arrow
 - Scale bar
 - Legend

Making Maps in R

- Add ward borders on top of the previous LSOA plot

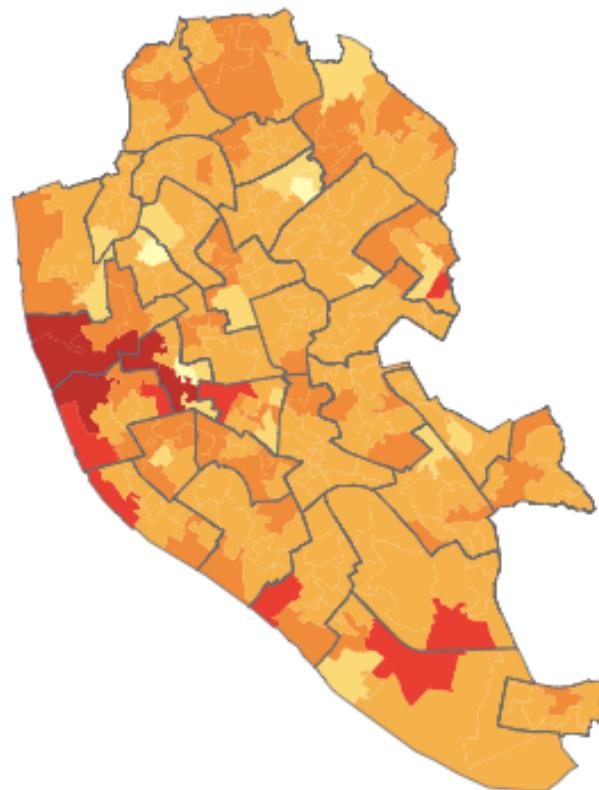
```
# Plot the wards on top of the LSOA choropleth  
plot(CASWARD, axes = FALSE, border = "#6B6B6B", add = TRUE)
```



Specify border colour

Add on top of previous
plot, rather than starting
A new plot

Making Maps in R



Making Maps in R

- Adding labels with `text()`

```
# Add on text labels for the wards  
text(coordinates(CASWARD) [, 1] , coordinates(CASWARD) [, 2] ,  
labels = CASWARD@data$wards ,  
cex = 0.7)
```

Scales label larger or smaller

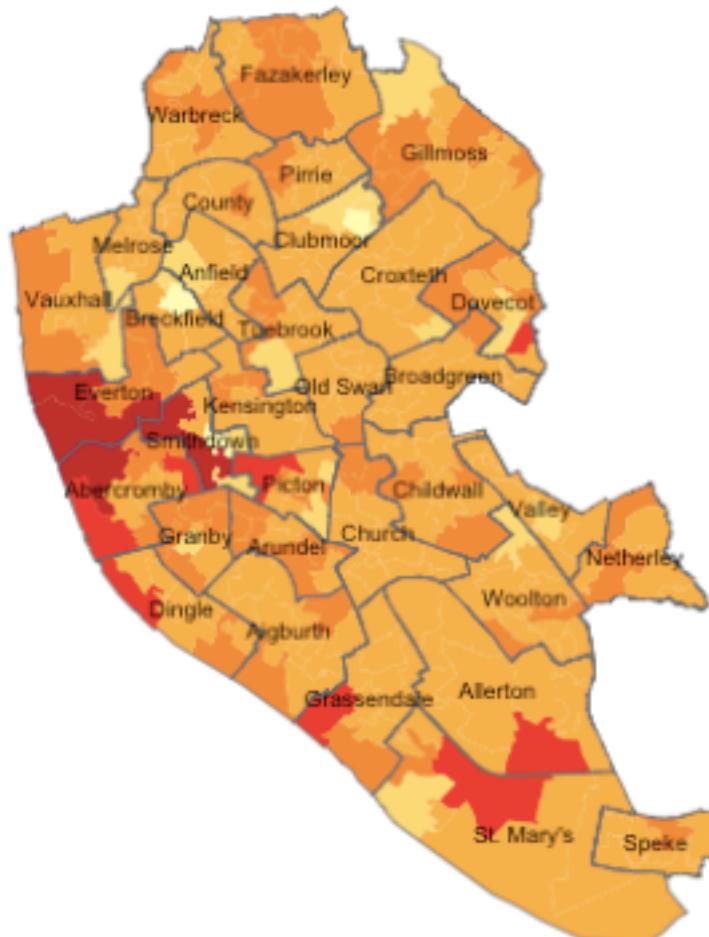
Label values

coordinates(CASWARD) extracts a list of centroids for each polygon

Remember – [row,column]!

Label goes in the centre

Making Maps in R



Making Maps in R

```
# Add the legend  
legend(x = 333603, y = 386096, legend =  
leglabs(round(breaks), between = " to "),  
fill = my_colours, bty = "n")
```

Creates text labels – the round() function reduces decimal places

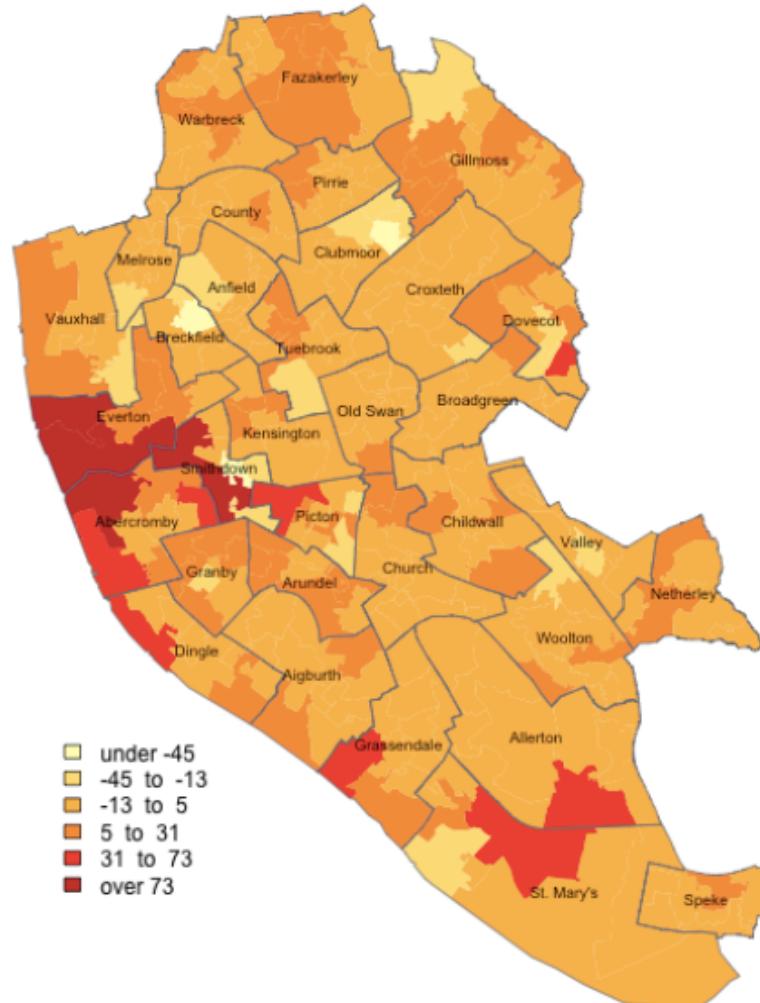
x.y location

Colour pallet

Background colour

```
# Finding locations  
Locator()
```

Making Maps in R



Making Maps in R

```
# Add North Arrow  
SpatialPolygonsRescale(layout.north.arrow(2), offset =  
c(333770, 381184), scale = 1000, plot.grid = F)
```

Offset – c(x,y)

Size of the arrow
– in the map units

Type of arrow

Making Maps in R

Could also use
hex values

Adding scale bar rather
than north arrow

```
# Add Scale Bar
SpatialPolygonsRescale(layout.scale.bar(), offset =
c(335379, 381184), scale = 3000,
fill = c("white", "black"), plot.grid = F)
```

```
# Add text to scale bar
text(335379, 381606, "0km", cex = 0.8)
text(335379 + 1500, 381606, "1.5km", cex = 0.8)
text(335379 + 3000, 381606, "3km", cex = 0.8)
```

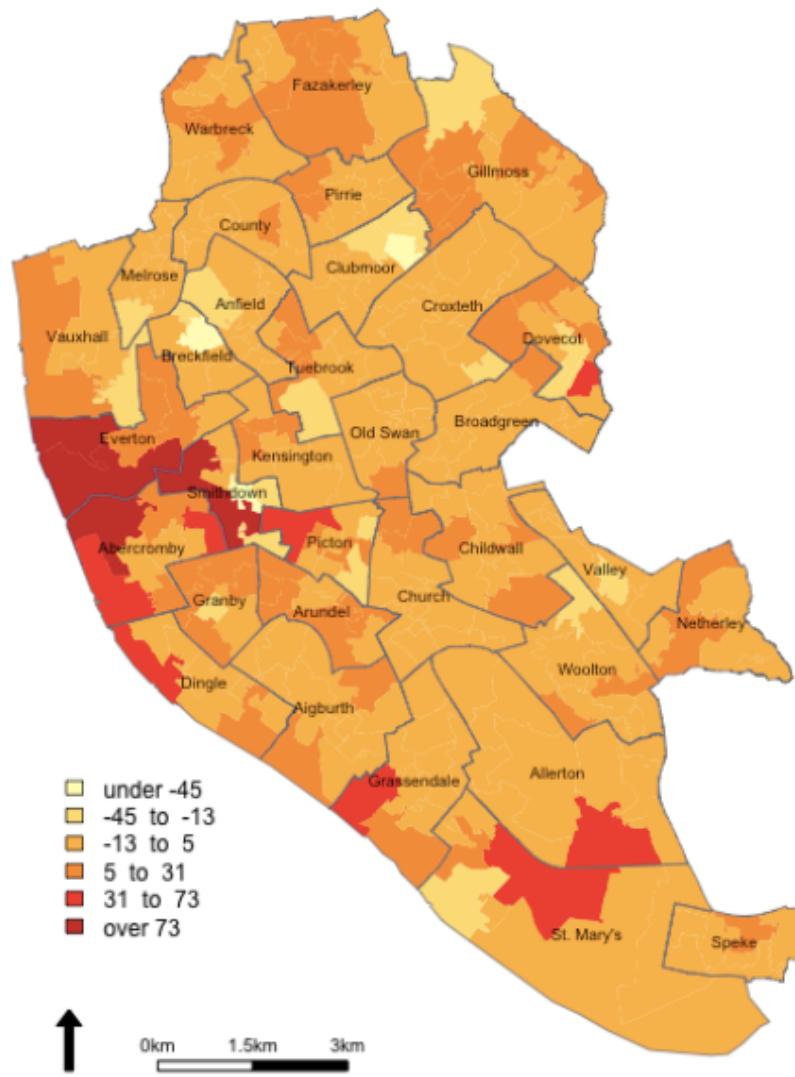
X

Y

Label

Scaling

Making Maps in R



R and Markdown

The screenshot shows the RStudio interface with the following components:

- Left Panel (Code Editor):** An R Markdown document titled "Untitled". The code includes metadata (title, author, date, output), a descriptive note about Markdown, and examples of R code chunks and plots.
- Right Panel (Environment):** The Global Environment pane shows that the environment is empty.
- Bottom Panel (Console):** The R console output shows the R version information, the "World-Famous Astronaut" welcome message, and details about the R distribution.

Many thanks...

