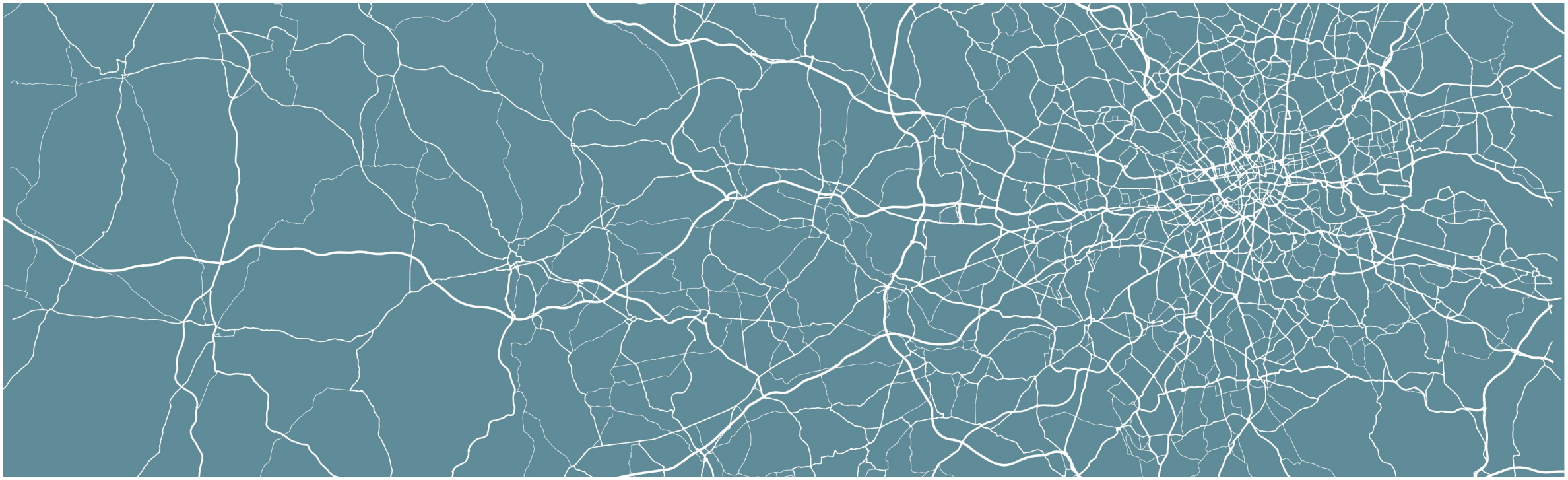
Geocomputation

Programming for Spatial Analysis



UCL

Where are we at?

Part I: Foundational Concepts

W1 Geocomputation: An Introduction

W2 GIScience and GIS Software

W3 Cartography and Visualisation

] QGIS

W4 Programming for Data Analysis

W5 **Programming for Spatial Analysis**

] R

This week

- Data science principles
- Using R as a GIS

Context

- Managing data with the `tidyverse`
- Managing spatial data with `sp` and `sf`

Data management

- Visualising spatial data with `ggplot2` and `tmap`

Visualising spatial data

- Some practical notes on coding

Miscellaneous

Context

Data Science Principles

Repeatability: the same methodology will produce the same (or nearly the same) outputs given the same inputs and reduces opportunity for error.

Reproducibility: work can be easily redone/completed by another (i.e. they have all information needed).

Data Science Principles

Collaboration: easy to share work with others and collaborate, preferably in real- time, with others, alongside easy integration with version control.

Scalability: basic – can re-run work easily, adjusting variables and parameters to include additional data; intermediate – can expand on work to include larger datasets; advanced – suitable for distributed computing.

Programming languages

“Everyone does need to learn to code. It is no longer sufficient for a GI Scientists to just work with a standard GIS interface: menus, buttons and black boxes.”

Brunsdon and Comber 2020

Using R as a GIS

- R is our programming language; RStudio is our Integrated Development Environment to develop and run R code.
- RStudio is now a high functionality piece of software – and can be used in many ways like a traditional GUI statistical software.
- However: there are a lot of differences compared to using a traditional GUI GIS software.

Using R as a GIS

- No map canvas – we do not ”see” our spatial data when it is loaded.
- When we load spatial data, it is loaded into the memory as a variable – we have to actively plot it using the base `plot()` function or a more advanced visualisation library to see our data.
- We can see the attribute table of our vector data through the `View()` function – this will load as a table.

Using R as a GIS

- When we use spatial analysis tools in QGIS, we often create new data files in the process – or we actively export new data files to save our edits. In R, we use variables in our processing and analysis.
- As a result, our analysis results and outputs are stored as variables. We need to actively export our outputs if we want to:
  - Share them or use within a different programme
  - Avoid re-running our scripts each time we want to use the outputs

Using R as a GIS

- No map composer / print layout to create our final maps.
- We do have visualisation libraries to create maps – that can then be automatically saved into PNGs or PDFs.
- Learning curve, but it is less fiddly than QGIS' Print Layout – and if you spot errors in your processing/analysis, updating your maps becomes a simple case of re-running your code.

When not to use R as a GIS?

- When digitising data (not covered in this module) or editing the geometry of spatial data (e.g. needing to move boundaries, change location of points).
- For fine-tuning cartographic outputs sometimes only a desktop GUI GIS will do.
- Interactive data exploration (although possible with interactive views).

Managing data

Managing data

- Wickham 2014. 80 percent of your time goes to data cleaning and preparation ('data wrangling').
- Tidy data refers to the structure and organisation of your data set.
- The idea boils down to three principles.
- Brought together in the tidyverse.

Tidy data

Each variable must have its own column

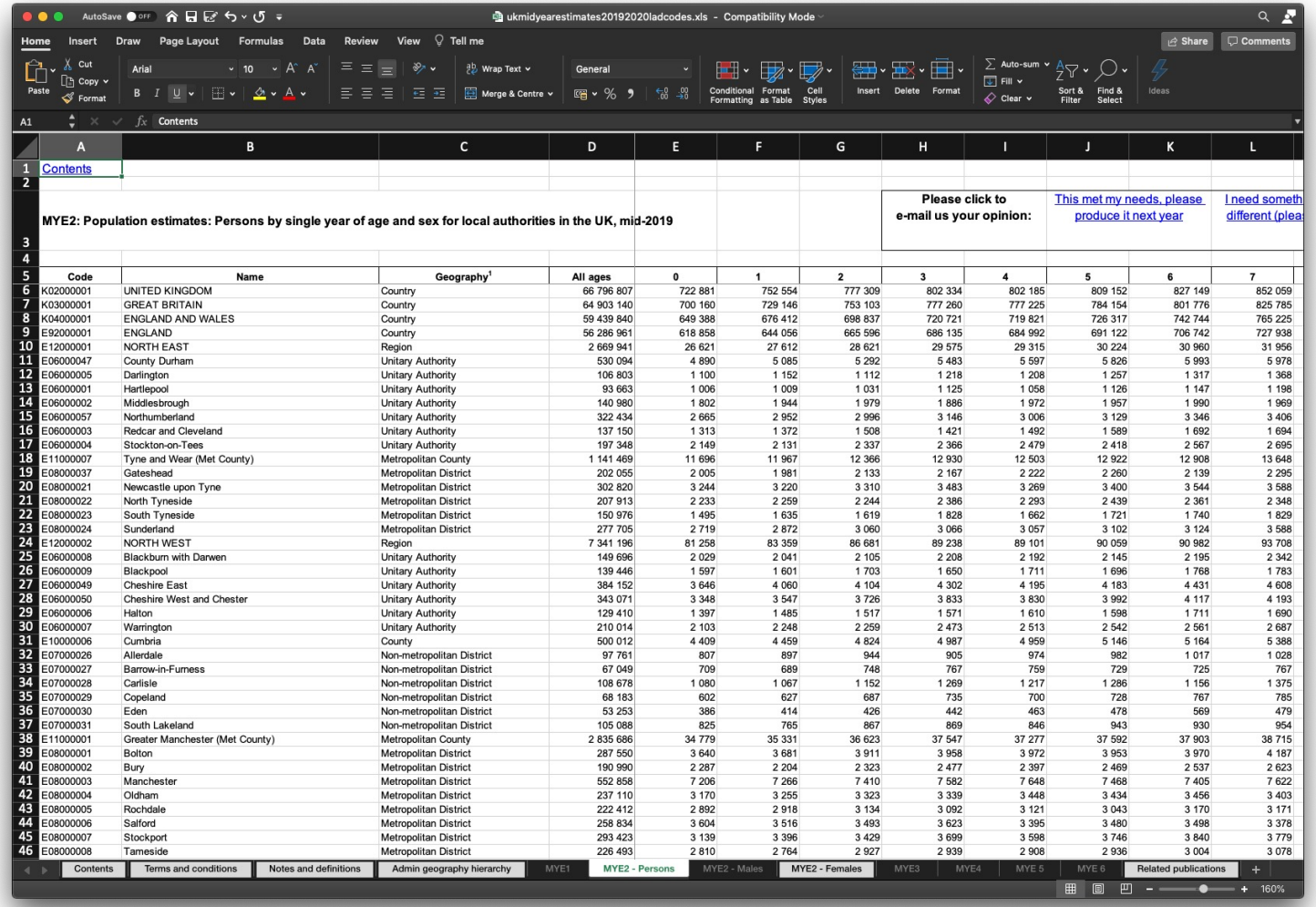
Tidy data

Each observation must have its own row

Tidy data

Each value must have its own cell

Tidy data



ukmidyearestimates20192020ladcodes.xls - Compatibility Mode

Contents

MYE2: Population estimates: Persons by single year of age and sex for local authorities in the UK, mid-2019

Please click to e-mail us your opinion: | This met my needs, please produce it next year | I need someth different (plea

Code	Name	Geography[1]	All ages	0	1	2	3	4	5	6	7
K02000001	UNITED KINGDOM	Country	66 796 807	722 881	752 554	777 309	802 334	802 185	809 152	827 149	852 059
K03000001	GREAT BRITAIN	Country	64 903 140	700 160	729 146	753 103	777 260	777 225	784 154	801 776	825 785
K04000001	ENGLAND AND WALES	Country	59 439 840	649 388	676 412	698 837	720 721	719 821	726 317	742 744	765 225
E92000001	ENGLAND	Country	56 286 961	618 858	644 056	665 596	686 135	684 992	691 122	706 742	727 938
E12000001	NORTH EAST	Region	2 669 941	26 621	27 612	28 621	29 575	29 315	30 224	30 960	31 956
E06000047	County Durham	Unitary Authority	530 094	4 890	5 085	5 292	5 483	5 597	5 826	5 993	5 978
E06000005	Darlington	Unitary Authority	106 803	1 100	1 152	1 112	1 218	1 208	1 257	1 317	1 368
E06000001	Hartlepool	Unitary Authority	93 663	1 006	1 009	1 031	1 125	1 058	1 126	1 147	1 198
E06000002	Middlesbrough	Unitary Authority	140 980	1 802	1 944	1 979	1 886	1 972	1 957	1 990	1 969
E06000057	Northumberland	Unitary Authority	322 434	2 665	2 952	2 996	3 146	3 006	3 129	3 346	3 406
E06000003	Redcar and Cleveland	Unitary Authority	137 150	1 313	1 372	1 508	1 421	1 492	1 589	1 692	1 694
E06000004	Stockton-on-Tees	Unitary Authority	197 348	2 149	2 131	2 337	2 366	2 479	2 418	2 567	2 695
E11000007	Tyne and Wear (Met County)	Metropolitan County	1 141 469	11 696	11 967	12 366	12 930	12 503	12 922	12 908	13 648
E08000037	Gateshead	Metropolitan District	202 055	2 005	1 981	2 133	2 167	2 222	2 260	2 139	2 295
E08000021	Newcastle upon Tyne	Metropolitan District	302 820	3 244	3 220	3 310	3 483	3 269	3 400	3 544	3 588
E08000022	North Tyneside	Metropolitan District	207 913	2 233	2 259	2 244	2 386	2 293	2 439	2 361	2 348
E08000023	South Tyneside	Metropolitan District	150 976	1 495	1 635	1 619	1 828	1 662	1 721	1 740	1 829
E08000024	Sunderland	Metropolitan District	277 705	2 719	2 872	3 060	3 066	3 057	3 102	3 124	3 588
E12000002	NORTH WEST	Region	7 341 196	81 258	83 359	86 681	89 238	89 101	90 059	90 982	93 708
E06000008	Blackburn with Darwen	Unitary Authority	149 696	2 029	2 041	2 105	2 208	2 192	2 145	2 195	2 342
E06000009	Blackpool	Unitary Authority	139 446	1 597	1 601	1 703	1 650	1 711	1 696	1 768	1 783
E06000049	Cheshire East	Unitary Authority	384 152	3 646	4 060	4 104	4 302	4 195	4 183	4 431	4 608
E06000050	Cheshire West and Chester	Unitary Authority	343 071	3 348	3 547	3 726	3 833	3 830	3 992	4 117	4 193
E06000006	Halton	Unitary Authority	129 410	1 397	1 485	1 517	1 571	1 610	1 598	1 711	1 690
E06000007	Warrington	Unitary Authority	210 014	2 103	2 248	2 259	2 473	2 513	2 542	2 561	2 687
E10000006	Cumbria	County	500 012	4 409	4 459	4 824	4 987	4 959	5 146	5 164	5 388
E07000026	Allerdale	Non-metropolitan District	97 761	807	897	944	905	974	982	1 017	1 028
E07000027	Barrow-in-Furness	Non-metropolitan District	67 049	709	689	748	767	759	729	725	767
E07000028	Carlisle	Non-metropolitan District	108 678	1 080	1 067	1 152	1 269	1 217	1 286	1 156	1 375
E07000029	Copeland	Non-metropolitan District	68 183	602	627	687	735	700	728	767	785
E07000030	Eden	Non-metropolitan District	53 253	386	414	426	442	463	478	569	479
E07000031	South Lakeland	Non-metropolitan District	105 088	825	765	867	869	846	943	930	954
E11000001	Greater Manchester (Met County)	Metropolitan County	2 835 686	34 779	35 331	36 623	37 547	37 277	37 592	37 903	38 715
E08000001	Bolton	Metropolitan District	287 550	3 640	3 681	3 911	3 958	3 972	3 953	3 970	4 187
E08000002	Bury	Metropolitan District	190 990	2 287	2 204	2 323	2 477	2 397	2 469	2 537	2 623
E08000003	Manchester	Metropolitan District	552 858	7 206	7 266	7 410	7 582	7 648	7 468	7 405	7 622
E08000004	Oldham	Metropolitan District	237 110	3 170	3 255	3 323	3 339	3 448	3 434	3 456	3 403
E08000005	Rochdale	Metropolitan District	222 412	2 892	2 918	3 134	3 092	3 121	3 043	3 170	3 171
E08000006	Salford	Metropolitan District	258 834	3 604	3 516	3 493	3 623	3 395	3 480	3 498	3 378
E08000007	Stockport	Metropolitan District	293 423	3 139	3 396	3 429	3 699	3 598	3 746	3 840	3 779
E08000008	Tameside	Metropolitan District	226 493	2 810	2 764	2 927	2 939	2 908	2 936	3 004	3 078

Contents | Terms and conditions | Notes and definitions | Admin geography hierarchy | MYE1 | MYE2 - Persons | MYE2 - Males | MYE2 - Females | MYE3 | MYE4 | MYE 5 | MYE 6 | Related publications

Common errors

- Column headers are values rather than variable names.
- Multiple variables are stored in one column.
- Variables are stored in both rows and columns.
- Multiple observational units are stored in the same column.
- A single observation is stored in multiple tables.

Tidy ?

country	year	type	count
Afghanistan	2019	cases	745
Afghanistan	2019	population	19 987 071
Afghanistan	2020	cases	2 666
Afghanistan	2020	population	20 595 360
Brazil	2019	cases	3,7737
Brazil	2019	population	172 006 362
Brazil	2020	cases	80 488
Brazil	2020	population	174 504 898
China	2019	cases	212 258
China	2019	population	1 272 915 272
China	2020	cases	213 766
China	2020	population	1 280 428 583

Tidy ?

country	year	rate
Afghanistan	2019	745 / 19,987,071
Afghanistan	2020	2,666 / 20,595,360
Brazil	2019	3,7737 / 172,006,362
Brazil	2020	80,488 / 174,504,898
China	2019	212,258 / 1,272,915,272
China	2020	213,766 / 1,280,428,583

Tidy ?

Cases

country	2019	2020
Afghanistan	745	2 666
Brazil	3,7737	80 488
China	212 258	213 766

Population

country	2019	2020
Afghanistan	19 987 071	20 595 360
Brazil	172 006 362	174 504 898
China	1 272 915 272	1 280 428 583

Tidy ?

country	year	cases	population
Afghanistan	2019	745	19 987 071
Afghanistan	2020	2 666	20 595 360
Brazil	2019	3,7737	172 006 362
Brazil	2020	80 488	174 504 898
China	2019	212 258	1 272 915 272
China	2020	213 766	1 280 428 583

Managing spatial data

- R has the capacity to read, load and store a range of file formats.
- Functions in both the base R library plus a huge host of software-specific packages (e.g. STATA, SPSS) for reading, writing and converting data between different file formats associated with those specific software (e.g. from a SPSS file to a `csv` etc.)
- Base R does not handle the reading, loading, and storing of spatial data.

Managing spatial data

- How do we read in spatial data?
- GDAL: Geospatial Data Abstraction Library (*reading, writing*)
- GEOS: Geometry Engine Open Source (*spatial operations*)

GEOS Geometry Engine Open Source

Managing spatial data

`sp`

- ‘Classes and methods for spatial data’
- First development in using spatial data in R (2005)
- Not fully compliant with the dataframe format

`sf`

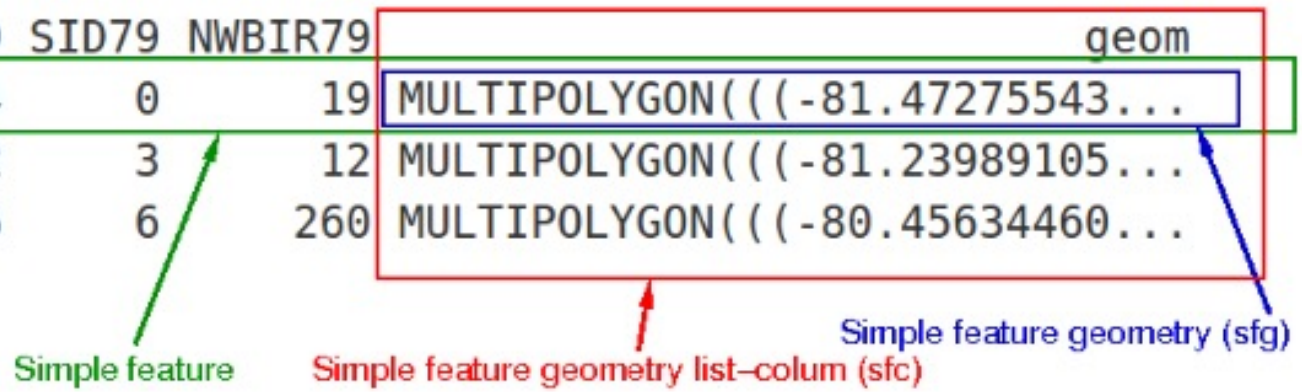
- ‘Support for simple features, a standardized way to encode spatial vector data’
- First development in using spatial data in R
- Fully compliant with the dataframe format

Managing spatial data

- The `sf` (simple features) package facilitates the storage, access and management of geometric objects stored as simple features in R.
- Importantly: sf objects are dataframes with a geometry column, containing WKT geometries at the end.

Managing spatial data

```
## Simple feature collection with 100 features and 6 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -84.32385 ymin: 33.88199 xmax: -75.45698 ymax: 36.58965
## epsg (SRID):    4267
## proj4string:    +proj=longlat +datum=NAD27 +no_defs
## precision:      double (default; no precision model)
## First 3 features:
##   BIR74 SID74 NWBIR74 BIR79 SID79 NWBIR79                              geom
## 1  1091     1       10  1364     0       19 MULTIPOLYGON(((-81.47275543...
## 2   487     0       10   542     3       12 MULTIPOLYGON(((-81.23989105...
## 3  3188     5      208  3616     6      260 MULTIPOLYGON(((-80.45634460...
```



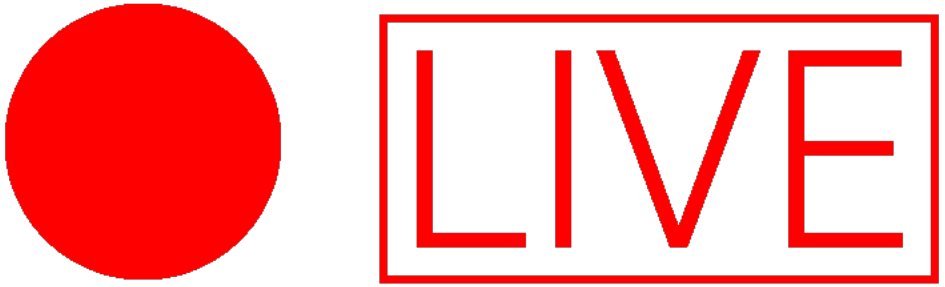
Managing spatial data

The `sf` package contains a huge set of tools for:

- Reading and writing spatial data.
- Querying a range of different point, line and polygon vector geometries
  - `st_` prefix for all functions identical to that used in PostGIS (SQL) queries
  - `st_` originates from SQL association of “Spatial Temporal”.

Managing spatial data

RStudio



Visualising spatial data

Libraries for spatial data visualisation

- A huge variety of packages that facilitate visualisation of spatial data.
- Most common: `tmap` and `ggplot2`
- Both are based on the “Layered Grammar of Graphics”

Grammar of Graphics

- Main concept: graphics can be built up through multiple layers of data.
- Values in a dataset are examples of aesthetics – values that can be viewed in a graphic.
- Data, scales and coordinate systems and plot annotations can then be layered on top of these data values to produce the final graphic.

Grammar of Graphics

Dataframe

x	*y*	Shape
2	4	a
1	1	a
4	15	b
9	80	b

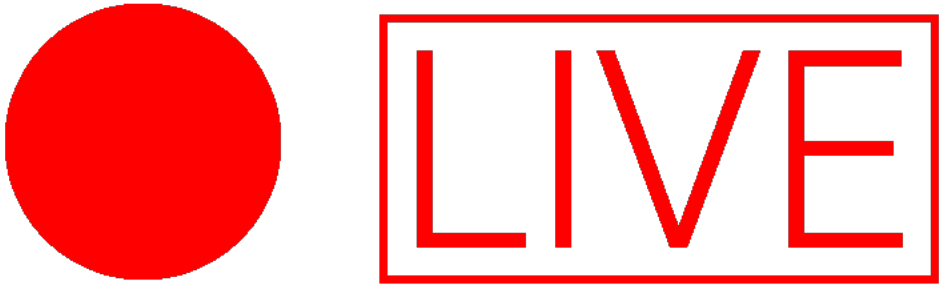
Dataframe values

Dataframe scale

Dataframe annotations

Final Graphic

RStudio

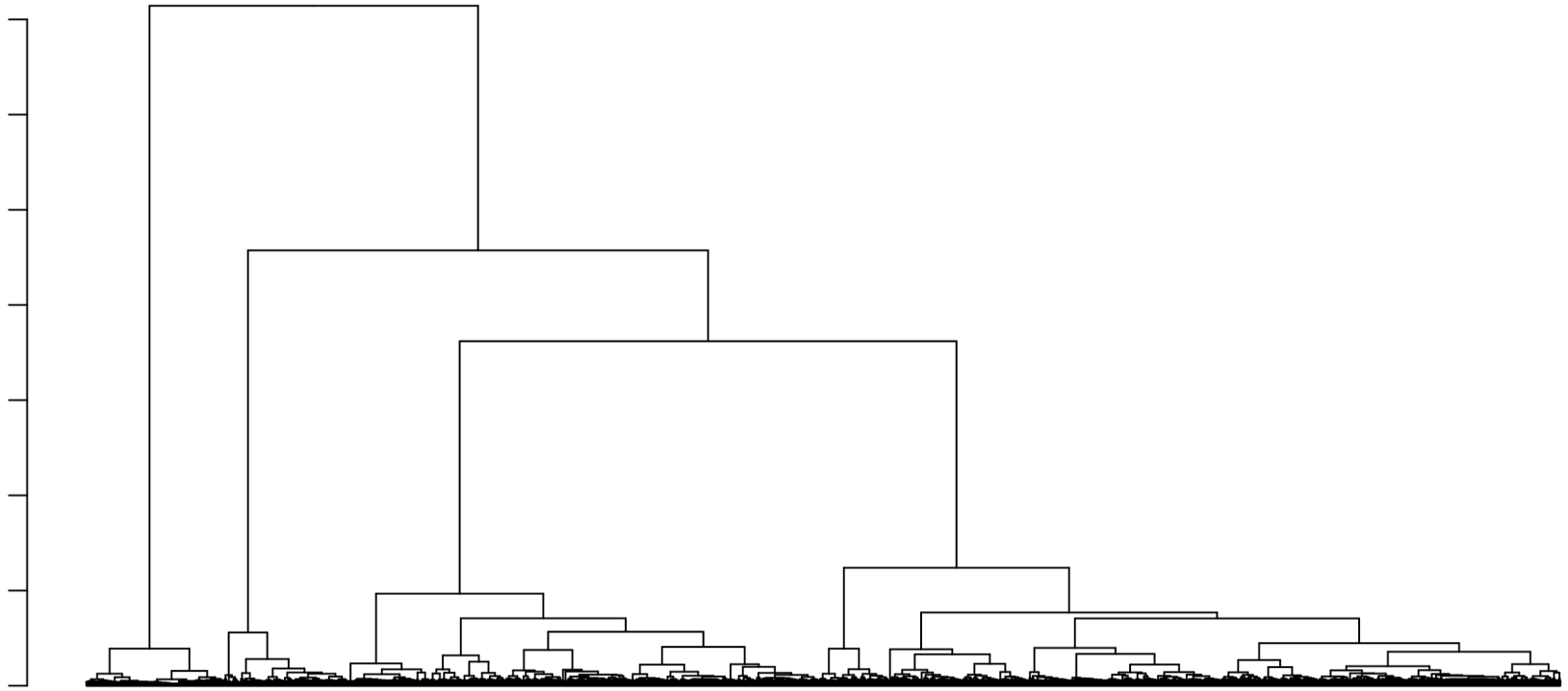


Miscellaneous

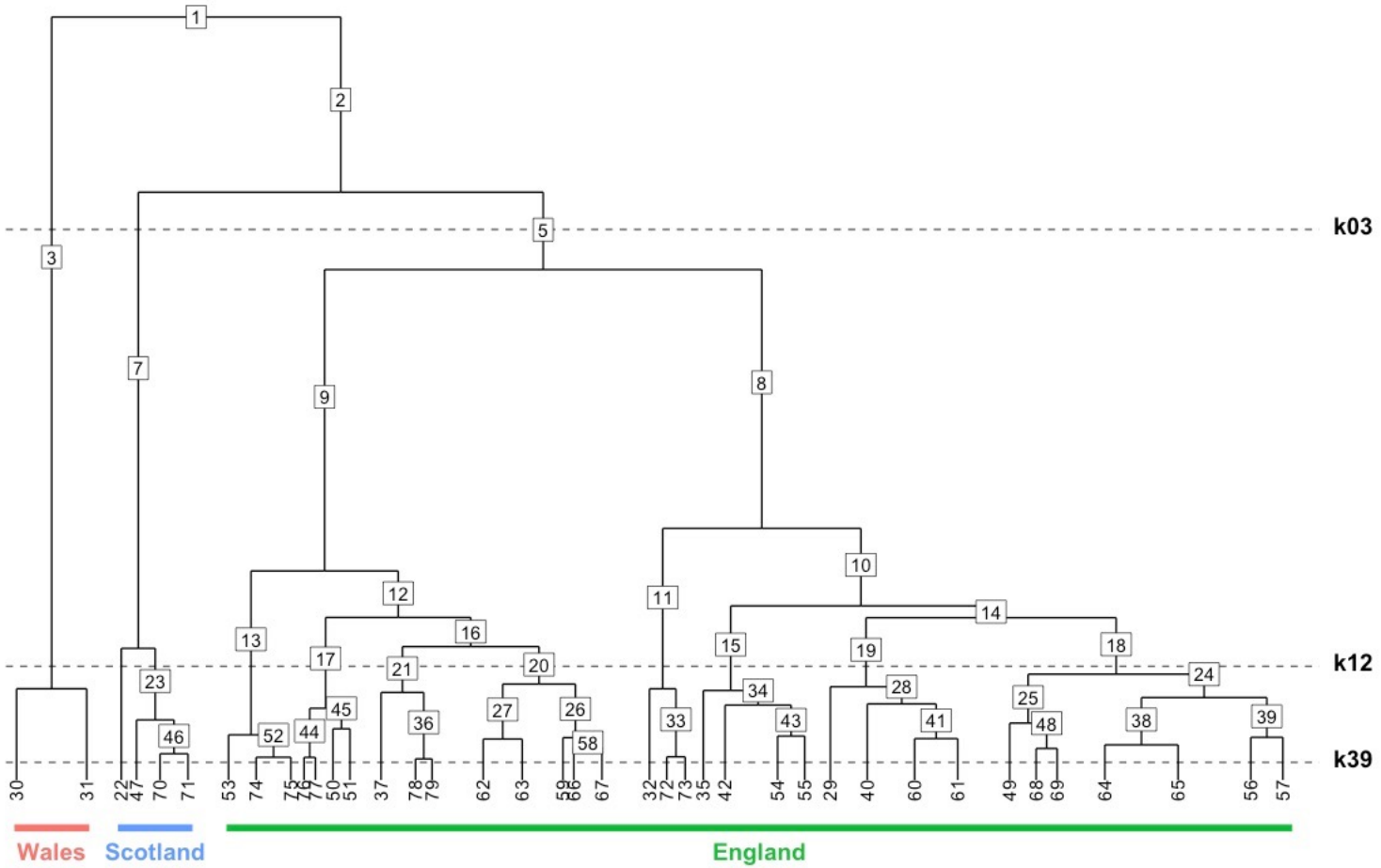
Comment your code

- Ensures that you remember what you did, especially with long pieces of code.
- Helps others to follow what you have done.
- In a way: in-text metadata

Comment your code



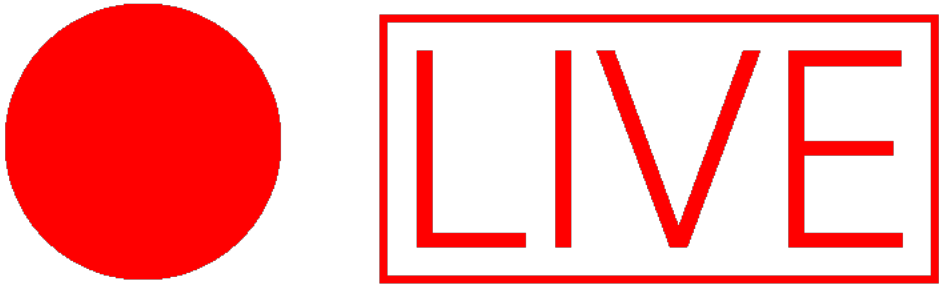
Comment your code



Version control

- The practice of tracking and managing changes to software code.
- Example: Git (GitHub, GitLab).
- “Track changes" on steroids.
- Beyond the scope of this module but you can still set up your own small version control system (”_v0.1”, “v_0.2”, “v_1.0”).

RStudio



Conclusion

- In many ways Geospatial / Geographic Information Systems have evolved considerably since their invention in the 1970s.
- Analysis has become more sophisticated. Processes are very similar, but the scale and size of datasets we are using require more computing resources.
- R was developed as a piece of statistical software, but being open source, coupled with a few key players making early developments in spatial analysis and visualisation means it is now one of the most comprehensive GIS solutions available today.

Further resources

Spatial analysis in R

- Geocomputation in R

Using Git and Github

- Happy Git and GitHub for the useR

Have a good reading week

Questions

Justin van Dijk

j.t.vandijk@ucl.ac.uk