Home

GDS4AE - *Geographic Data Science for Applied Economists*

- Dani Arribas-Bel [@darribas]
- Diego Puga [@ProfDiegoPuga]

Contact

Dani Arribas-Bel - D.Arribas-Bel [at] liverpool.ac.uk
Senior Lecturer in Geographic Data Science
Office 508, Roxby Building,
University of Liverpool - 74 Bedford St S,
Liverpool, L69 7ZT,
United Kingdom.

Diego Puga - diego.puga [at] cemfi.es
Professor
CEMFI,
Casado del Alisal 5,
28014 Madrid,
Spain.

Citation

If you use materials from this resource in your own work, we recommend the following citation:

```
@article{darribas_gds_course,
  author = {Dani Arribas-Bel and Diego Puga},
  title = {Geographic Data Science for Applied Economists},
  year = 2021,
  annote = {\href{https://darribas.org/gds4ae}}
}
```

Overview

This resource provides an introduction to Geographic Data Science for applied economists using Python. It has been designed to be delivered within 15 hours of teaching, split into ten sessions of 1.5h each.

How to follow along

GDS4AE is best followed if you can interactively tinker with its content. To do that, you will need two things:

1. A computer set up with the Jupyter Lab environment and all the required libraries (please see the Software stack part in the Infrastructure section for instructions)

Contents

2. A local copy of the materials that you can run on your own computer (see the repository section in the Infrastructure section for instructions)

-
-
-

Content

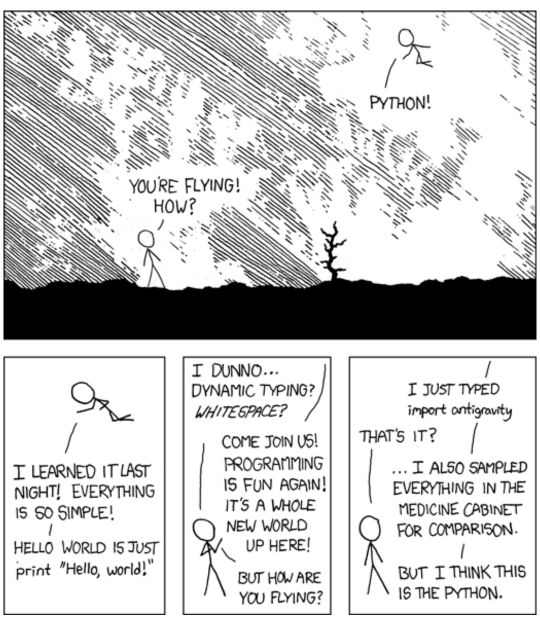
The structure of content is divided in nine blocks:

- *Introduction*: get familiar with the computational envirionment of modern data science
- *Spatial Data*: what do spatial data look like in Python?
- *Geovisualisation*: make (good) data maps
- *Spatial Feature Engineering* (Part I and Part II): augment and massage your data using Geography before you feed them into your model
- *Spatial Networks* (Part I and Part II): understand, acquire and work with spatial graphs
- *Transport Costs*: "getting there" doesn't always cost the same
- *Visual challenges*: all the details nobody told you (but should have) about visualising geographic data

Each block has its own section and is designed to be delivered in 1.5 hours approximately. The content of some of these blocks relies on external resources, all of them freely available. When that is the case, enough detail is provided in the to understand how additional material fits in.

Why Python?

There are several reasons why we have made this choice. Many of them are summarised nicely in this article by The Economist (paywalled).:w



Source: XKCD

Data

All the datasets used in this resource is freely available. Some of them have been developed in the context of the resource, others are borrowed from other resources. A full list of the datasets used, together with links to the original source, or to reproducible code to generate the data used is available in the Datasets page.

License

Infrastructure

This page covers a few technical aspects on how the course is built, kept up to date, and how you can create a computational environment to run all the code it includes.

Software stack

This course is best followed if you can not only read its content but also interact with its code and even branch out to write your own code and play on your own. For that, you will need to have installed on your computer a series of interconnected software packages; this is what we call a *stack*.

Instructions on how to install a software stack that allows you to run the materials of this course depend on the operating system you are using. Detailed guides are available for the main systems on the following resource, provided by the Geographic Data Science Lab:

@gdsl-ul/soft_install

Github repository

All the materials for this course and this website are available on the following Github repository:

@darribas/gds4ae

If you are interested, you can download a compressed `.zip` file with the most up-to-date version of all the materials, including the HTML for this website at:

Icon made by Freepik from www.flaticon.com

@darribas/gds4ae_zip

Containerised backend

The course is developed, built and tested using the `gds_env`, a containerised platform for Geographic Data Science. You can read more about the `gds_env` project at:

Binder

Binder is service that allows you to run scientific projects in the cloud for free. Binder can spin up *"ephemeral"* instances that allow you to run code on the browser without any local setup. It is possible to run the course on Binder by clicking on the button below:

launch binder

Warning

It is important to note Binder instances are *ephemeral* in the sense that the data and content created in a session is **NOT** saved anywhere and is deleted as soon as the browser tab is closed.

Binder is also the backend this website relies on when you click on the rocket icon (🚀) on a page with code. Remember, you can play with the code interactively but, once you close the tab, all the changes are lost.

Introduction

Geographic Data Science

Note

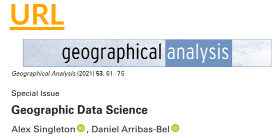
This section is adapted from Block A of the GDS Course [AB19].

Before we learn *how* to do Geographic Data Science or even *why* you would want to do it, let's start with *what* it is. We will rely on two resources:

- First, in this video, Dani Arribas-Bel covers the building blocks at the First Spatial Data Science Conference, organised by CARTO

- Second, *Geographic Data Science*, by Alex Singleton and Dani Arribas-Bel [SAB19]

URL


geographical analysis

Geographical Analysis (2021) 53, 61–75

Special Issue

Geographic Data Science

Alex Singleton, Daniel Arribas-Bel

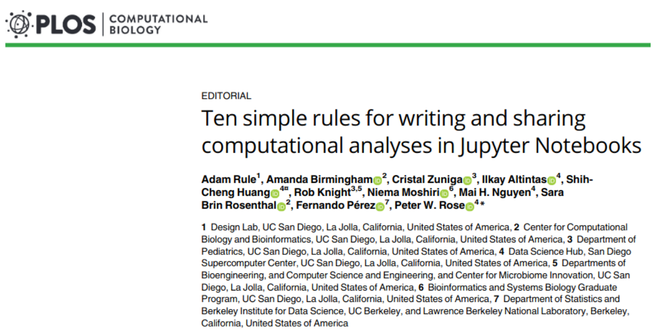
Department of Geography and Planning, University of Liverpool, Liverpool, L69 7ZT, U.K.

The computational stack

One of the core learning outcomes of this course is to get familiar with the modern computational environment that is used across industry and science to "do" Data Science. In this section, we will learn about ecosystem of concepts and tools that come together to provide the building blocks of much computational work in data science these days.

Source: [The Atlantic]

- *Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks*, by Adam Rule et al. [RBZ+19]


PLOS | COMPUTATIONAL BIOLOGY

EDITORIAL

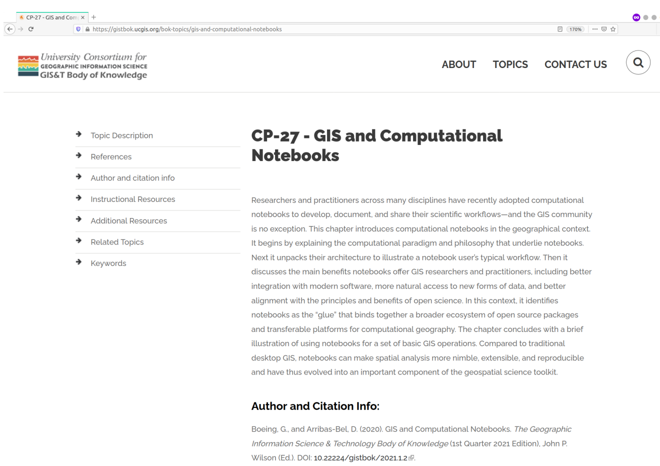
Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks

Adam Rule[1], Amanda Birmingham[2], Cristal Zuniga[3], Ilkay Altintas[4], Shih-Cheng Huang[4¤], Rob Knight[3,5], Niema Moshiri[6], Mai H. Nguyen[4], Sara Brin Rosenthal[2], Fernando Pérez[7], Peter W. Rose[4]*

1 Design Lab, UC San Diego, La Jolla, California, United States of America, 2 Center for Computational Biology and Bioinformatics, UC San Diego, La Jolla, California, United States of America, 3 Department of Pediatrics, UC San Diego, La Jolla, California, United States of America, 4 Data Science Hub, San Diego Supercomputer Center, UC San Diego, La Jolla, California, United States of America, 5 Departments of Bioengineering, and Computer Science and Engineering, and Center for Microbiome Innovation, UC San Diego, La Jolla, California, United States of America, 6 Bioinformatics and Systems Biology Graduate Program, UC San Diego, La Jolla, California, United States of America, 7 Department of Statistics and Berkeley Institute for Data Science, UC Berkeley, and Lawrence Berkeley National Laboratory, Berkeley, California, United States of America

URL

- *GIS and Computational Notebooks*, by Geoff Boeing and Dani Arribas-Bel [BAB20]


University Consortium for Geographic Information Science
GIS&T Body of Knowledge

ABOUT TOPICS CONTACT US

- Topic Description
- References
- Author and citation info
- Instructional Resources
- Additional Resources
- Related Topics
- Keywords

CP-27 - GIS and Computational Notebooks

Researchers and practitioners across many disciplines have recently adopted computational notebooks to develop, document, and share their scientific workflows—and the GIS community is no exception. This chapter introduces computational notebooks in the geographical context. It begins by explaining the computational paradigm and philosophy that underlie notebooks. Next it unpacks their architecture to illustrate a notebook user's typical workflow. Then it discusses the main benefits notebooks offer GIS researchers and practitioners, including better integration with modern software, more natural access to new forms of data, and better alignment with the principles and benefits of open science. In this context, it identifies notebooks as the "glue" that binds together a broader ecosystem of open source packages and transferable platforms for computational geography. The chapter concludes with a brief illustration of using notebooks for a set of basic GIS operations. Compared to traditional desktop GIS, notebooks can make spatial analysis more nimble, extensible, and reproducible and have thus evolved into an important component of the geospatial science toolkit.

Author and Citation Info:

Boeing, G., and Arribas-Bel, D. (2020). GIS and Computational Notebooks. *The Geographic Information Science & Technology Body of Knowledge* (1st Quarter 2021 Edition), John P. Wilson (Ed.). DOI: 10.22224/gistbok/2021.1.2

URL

Now we are familiar with the conceptual pillars on top of which we will be working, let's switch gears into a more practical perspective. The following two clips cover the basics of Jupyter Lab, the frontend that glues all the pieces together, and Jupyter Notebooks, the file format, application, and protocol that allows us to record, store and share workflows.

> **Note**
>
> The clips are sourced from Block A of the GDS Course [AB19]

Jupyter Lab

Jupyter Notebooks

Spatial Data

Ahead of time...

This block is all about understanding spatial data, both conceptually and practically. Before your fingers get on the keyboard, the following readings will help you get going and familiar with core ideas:

- Chapter 2 of the GDS Book [RABWng], which provides a conceptual overview of representing Geography in data
- Chapter 3 of the GDS Book [RABWng], a sister chapter with a more applied perspective on how concepts are implemented in computer data structures

Additionally, parts of this block are based and source from Block C in the GDS Course [AB19].

Hands-on coding

(Geographic) tables

```
import pandas
import geopandas
```

Points

Local files | Online read

Assuming you have the file locally on the path `../data/`:

```
pts = geopandas.read_file("../data/madrid_abb.gpkg")
```

Point geometries from columns

```
pts.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 18399 entries, 0 to 18398
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             18399 non-null  object
 1   price_usd         18399 non-null  float64
 2   log1p_price_usd   18399 non-null  float64
 3   accommodates      18399 non-null  int64
 4   bathrooms         18399 non-null  object
 5   bedrooms          18399 non-null  float64
 6   beds              18399 non-null  float64
 7   neighbourhood     18399 non-null  object
 8   room_type         18399 non-null  object
 9   property_type     18399 non-null  object
 10  WiFi              18399 non-null  object
 11  Coffee            18399 non-null  object
 12  Gym               18399 non-null  object
 13  Parking           18399 non-null  object
 14  km_to_retiro      18399 non-null  float64
 15  geometry          18399 non-null  geometry
dtypes: float64(5), geometry(1), int64(1), object(9)
memory usage: 2.2+ MB
```

```
pts.head()
```

	price	price_usd	log1p_price_usd	accommodates	bathrooms	bedrooms
0	$60.00	60.0	4.110874	2	1 shared bath	1.0
1	$31.00	31.0	3.465736	1	1 bath	1.0
2	$60.00	60.0	4.110874	6	2 baths	3.0
3	$115.00	115.0	4.753590	4	1.5 baths	2.0
4	$26.00	26.0	3.295837	1	1 private bath	1.0

Lines

```
lines = geopandas.read_file("http://arturo.300000kms.net/data/model.geojson.zip")
```

```
lines.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 66499 entries, 0 to 66498
Data columns (total 59 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   OGC_FID                       66499 non-null  object
 1   geom_pu                       66499 non-null  object
 2   dist_barri                    66483 non-null  object
 3   dm_id                         66499 non-null  object
 4   train                         66499 non-null  int64
 5   land_use_mix                  66499 non-null  float64
 6   closeness_small_parks         66499 non-null  float64
 7   residence_ratio               66499 non-null  float64
 8   block_area                    66499 non-null  float64
 9   intersection_density          66499 non-null  float64
 10  anisotropicity                66499 non-null  float64
 11  average_age                   66499 non-null  float64
 12  age_diversity                 66499 non-null  float64
 13  age_deviation_diversity       66499 non-null  float64
 14  built_density                 66499 non-null  float64
 15  population_density            66499 non-null  float64
 16  ocasional_density             66499 non-null  float64
 17  proximity_density             66499 non-null  float64
 18  leisure_density               66499 non-null  float64
 19  educational_density           66499 non-null  float64
 20  nightlife_density             66499 non-null  float64
 21  culture_density               66499 non-null  float64
 22  closeness_large_parks         66499 non-null  float64
 23  closeness_primary_roads       66499 non-null  float64
 24  closeness_secondary_roads     66499 non-null  float64
 25  closeness_tertiary_roads      66499 non-null  float64
 26  public_space_surface          66499 non-null  float64
 27  parks_surface                 66499 non-null  float64
 28  parking_surface               66499 non-null  float64
 29  warehouse_surface             66499 non-null  float64
 30  commerce_surface              66499 non-null  float64
 31  cultural_surface              66499 non-null  float64
 32  industrial_surface            66499 non-null  float64
 33  industrial_rural_surface      66499 non-null  float64
 34  sports_surface                66499 non-null  float64
 35  hotel_surface                 66499 non-null  float64
 36  garden_surface                66499 non-null  float64
 37  office_surface                66499 non-null  float64
 38  singular_surface              66499 non-null  float64
 39  religious_surface             66499 non-null  float64
 40  spectacle_surface             66499 non-null  float64
 41  housing_surface               66499 non-null  float64
 42  public_service_surface        66499 non-null  float64
 43  rural_surface                 66499 non-null  float64
 44  average_quality               66499 non-null  float64
 45  quality_deviation_diversity   66499 non-null  float64
 46  tertiary_roads_length         66499 non-null  float64
 47  street_length                 66499 non-null  float64
 48  street_one_way                66499 non-null  int64
 49  street_orientation            66499 non-null  float64
 50  street_centrality_degree      66499 non-null  float64
 51  street_centrality_eigenvector  66499 non-null  float64
 52  street_centrality_betweenness  66499 non-null  float64
 53  street_centrality_closeness   66499 non-null  float64
 54  street_hierarchy_primary      66499 non-null  int64
 55  street_hierarchy_secondary    66499 non-null  int64
 56  street_hierarchy_tertiary     66499 non-null  int64
 57  pk                            66499 non-null  object
 58  geometry                      66499 non-null  geometry
dtypes: float64(48), geometry(1), int64(5), object(5)
memory usage: 29.9+ MB
```

Polygons

Surfaces

Visualisation

Spatial operations

(Re-)Projections
Centroids
Areas

Distances

Next steps

Geovisualisation

Spatial Feature Engineering (I)

Spatial Feature Engineering (II)

Spatial Networks (I)

Spatial Networks (II)

Transport costs

Visual challenges and opportunities

Student presentations

In this session, students will present their projects to the group.

Datasets

This section covers the datasets required to run the course interactively. For archival reasons, all of those listed here have been mirrored in the repository for this course so, if you have downloaded the course, you already have a local copy of them.

Airbnb data

> **Source**
>
> This dataset has been sourced from the course "Spatial Modelling for Data Scientists". The file imported here corresponds to the `v0.1.0` version.

This dataset contains a pre-processed set of properties advertised on the AirBnb website within the region of Madrid (Spain), together with house characteristics.

- Data file `[URL]`
- Code used to generate the file `[URL]`
- Furhter information `[URL]`

Arturo

Further Resources

If this course is successful, it will leave you wanting to learn more about using Python for (Geographic) Data Science. See below a few resources that are good "next steps".

Courses

- The "Automating GIS processes", by Vuokko Heikinheimo and Henrikki Tenkanen is a great overview of GIS with a modern Python stack:

> https://autogis-site.readthedocs.io/

- The "GDS Course" by Dani Arribas-Bel [AB19] is an introductory level overview of Geographic Data Science, including notebooks, slides and video clips.

> https://darribas.org/gds_course

Books

- *"Python for Geographic Data Analysis"*, by Henrikki Tenkanen, Vuokko Heikinheimo and David Whipp:

> https://pythongis.org/

- *"Geographic Data Science in Python"*, by Sergio J. Rey, Dani Arribas-Bel and Levi J. Wolf:

> https://geographicdata.science

Bibliography

[AB19]
Dani Arribas-Bel. A course on geographic data science. *The Journal of Open Source Education*, 2019. doi:https://doi.org/10.21105/jose.00042.

[BAB20]
Geoff Boeing and Dani Arribas-Bel. Gis and computational notebooks. In John P. Wilson, editor, *The Geographic Information Science & Technology Body of Knowledge*. UCGIS, 2020.

[RABWng]
Sergio J. Rey, Daniel Arribas-Bel, and Levi J. Wolf. *Geographic Data Science with PySAL and the PyData stack*. CRC press, forthcoming.

[RBZ+19]
Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H Nguyen, Sara Brin Rosenthal, Fernando Pérez, and others. Ten simple rules for writing and sharing computational analyses in jupyter notebooks. *PLoS Comput Biol*, 2019. doi:https://doi.org/10.1371/journal.pcbi.1007007.

[SAB19]
Alex Singleton and Daniel Arribas-Bel. Geographic data science. *Geographical Analysis*, 2019.
