FROM SOCIAL SCIENCE TO DATA SCIENCE

By Bernie Hogan

Opening lecture:

Social Data Science as the Science of Measurement

Data science as a paradigm

Data science is an emerging approach to science that extending beyond statistics towards maths, engineering, and computer science. It also draws liberally from the social sciences (e.g., linguistics to computational linguistics).

Statistics similarly focus on data. It's a deep body of knowledge that tends to focus (via RA Fisher certainly) on the generalizability and representativeness of samples of a population.

By contrast, data science tends to have access to vast, streaming, or complete data and instead focuses on how to understand, classify, or predict within these vast samples, but be careful in how they are constructed and filtered. In practice it often focuses on workflows rather than claims.

It’s less useful to press further in distinguishing fields and get territorial. Rather we might consider these as *approaches to* knowledge rather than *domains of* knowledge. So how would data science approach knowledge?

What is data science (and social data science)?

As a provisional (and necessarily incomplete) definition we might say **data science** is the ‘science of operationalisation of phenomena’. For example:

- How many messages shared between two people would be enough to represent friendship?
- Does it matter the time, content, peer relationships, or context?
- What data can help us answer this question and how can we test that answer?

Social data science is thus the ‘science of operationalisation of social life’. We often have ideas about the world that need to be translated into data or data that can offer us new ideas of the world.

This is not just finding the ‘true’ concept, or attaching a dollar value, but understanding a workable value for the accretion of knowledge.

DIKW

From the Information Visualisation (INFOVIS) literature is a schema called DIKW standing for Data -> Information -> Knowledge -> Wisdom (Rowley, 2007).

- **Data**: measurements of phenomena,
- **Information**: Signals from that data / “differences that make a difference”,
- **Knowledge**: Information situated in or understood in context,
- **Wisdom**: Information outside of or in an analogous context.

It’s not bad but it’s certainly not perfect. Do we start with data? Does all knowledge need to go through this schema to become wisdom? What of things we learn from experience rather than through testing?

(PO)DIKW

We might modify this schema and start not from data but from ‘the world’ (not the earth, but the known or knowable Universe). The world simply exists. To interact with the world we take in information, encode it, and react to it. So prior to data we should think of:

Phenomena: What is it we think we are testing, observing, expecting from the world? How are we using our intuitions, prior academic literature, and analytical tools to perceive (or collect) phenomena?

Operationalisation: How can we encode and thus measure or compare different phenomena.

- E.g., Is every friendship the same? How would we know what to measure for a friendship? A Facebook friendship? “Mutuals” + real name? Verbally stated “is my friend”, observed through proximity at an event?

Thinking beyond the interface

Social media platforms host data and present it using specific apps or webpages. Yet, the data does not necessarily need to be presented in this way.

To think beyond the interface is to ask “what can I learn if we represent the same data in a different way”?

- We can consider **performative** features: are we making our lives more amenable to data collection? Did the clock make the workday 9 to 5?
- We can ask about **consequences**: Do some sites structure data in ways that change our behaviour? Do we speak to people less on their birthdays and use a Facebook greeting instead?
- We can consider inclusion and exclusion: Not just as a technical matter, but who is excluded when we observe certain data? Does their **exclusion** (or someone else’s **inclusion**) create a biased claim? Does it reinforce unequal **power** relations?

Coding in practice

“To program” or “to code” is a practice of specifying some consistent phrases that will perform an operation that can be reliably repeated. It is like writing a recipe.

Data refers to measurements. When coding for data we are expecting consistent measurements from the world. We then expect to make claims about those measurements.

We do not need computers to do this. However, computers can do this in a very consistent (often fussy) way and very fast.

The world is full of exceptions and surprises. This means that our coding needs to make compromises. If code is too complex and catches everything it might also be too hard to understand or slow to write. If it is too simple or coarse, we risk misunderstanding the world.

Code as ordering the world

There are three essential forms of structure to consider:

- **The list**: a sequential order of elements.
- **The dictionary**: a relational mapping of elements.
- **The set**: a marking of in versus out.

Virtually all more complex forms of data come out of these three ways of ordering.

Understanding how these relate enables you to appreciate how data comes in from the world and how you can then shape it to ask meaningful questions.

These are basic collection types in Python. Then in the Python data analysis package pandas are more complex objects that can be used with these forms of ordering.

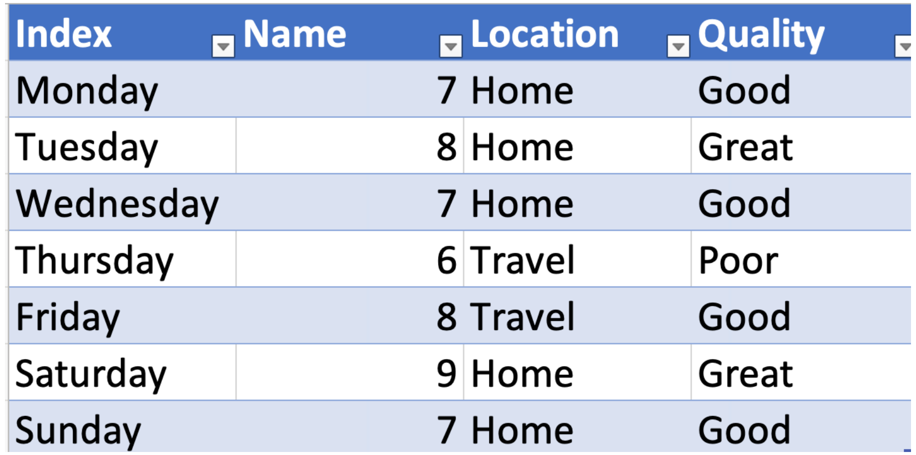
The Series and the DataFrame

In **Chapter 2**, we explore the series as a way to store a single distribution of measurements.

A **Chapter 3** we see how a DataFrame stores many Series, aligned by index.

We begin the code by looking at how to make use of a Series. This is not merely a technical matter. It is because the Series helps us consider that data is not merely an unstructured collection of encodings. It is a means of structuring the world.

The DataFrame is an extension of that by aligning multiple Series. Two measurements about the same person, multiple measurements on each day of the week, etc.



Index	Name	Location	Quality
Monday	7	Home	Good
Tuesday	8	Home	Great
Wednesday	7	Home	Good
Thursday	6	Travel	Poor
Friday	8	Travel	Good
Saturday	9	Home	Great
Sunday	7	Home	Good

Translating the world into a DataFrame

The next two chapters look at the ways in which data from the world can be translated into a DataFrame.

In **Chapter 4**, we look at file types.

- Turns out, data on the web is sent in some pretty consistent formats regardless of platform or context. By covering the bases of these formats, you can think about how to collect data from a variety of contexts, not just 'use a Twitter collector'.
- We focus first on rectangular data, such as CSV and Excel.
- We then turn to nested data, like JSON, HTML, and XML.

In **Chapter 5**, we look at how to group and merge tables.

- Data from different sources or data at different scales can be meaningfully combined. We can take group-level averages or get some country specific measure for the various participants from different countries.

And then the trouble begins

We start using these tools with real live data from the world.

Accessing Data from the Web

No practical book on this topic would be complete without some way of showing how to get data from the world programmatically. We use the Internet and more specifically, the World Wide Web to access (as well as create) data.

The web has a set of conventions that allow you to request not simply 'a webpage', but specific data from a server. These conventions range in complexity.

The simplest are argument strings: simply add things to the end of a URL.

The more complex conventions involve authenticated APIs. These are not merely technically complex, they are politically complex.

- In **Chapter 6** we look at accessing data from the World Wide Web. The last third of the chapter discusses the ethics of online data collection through the principle of data minimisation. It's one of many principles, but it will get you started.
- In **Chapter 7** we look at authenticated access using Reddit and *ahem* Twitter. Importantly, I show how to keep authentication keys out of one's sharable code. I also discuss APIs as political objects and review *Sandvig v. Barr*, an important precedent.

Now we need to ask questions and consider how to answer them

Chapter 8 might be my favourite chapter and yet, there is no code in it at all. It is a chapter on scientific reasoning with the goal of helping create workable research questions.

Research questions are a skill, not simply a hunch. In the chapter I cover a few principles that help us construct useful questions. The key ones are:

- Induction versus deduction versus abduction
- Expectations and observations
- Prediction versus explanation
- Operationalisation
- Boundedness
- Hypotheses versus research questions

Expectations versus observations in Depth

Of all of these principles, the negotiation between expectation and observation is the most central to scientific research. In my opinion, it is *the* scientific practice and everything else are details.

Expectation:

- What we expect comes from past scientific research and our life experience. We need to be critical of past insights, asking in whose interest did they serve or what is the strength of evidence for these claims. We reason by analogy and posit a justification for collecting and analysing new data.

Observation:

- What we observe comes from the availability and sophistication of our tools/practices, our own positionality, the accessibility of data, and the ethical and moral justifications for our own investigation.
- We cannot see if we do not know where (or how) to look.
- We cannot hear if we do not know where (or how) to listen.

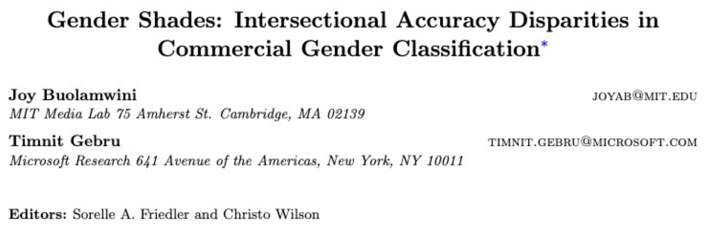
A postmodern appeal

A great deal of work now will consider whether certain people have a privileged or unique vantage point from which to know the world. Power may be revealed more obviously to some than others.

We can lean into this rather than fight against it by considering abduction.

Abduction: Most reasoning is actually by habit and intuition. This intuition does not exclusively come from past literature, but from one's life experience, directly or indirectly. We may have peers, events, or identities that facilitate intuitions about the world that are unorthodox or unexpected. But they still may be valid and produce a meaningful scientific claim.

By translating our abduction into a deductive or inductive research question we can both apply traditional scientific reasoning while including our own (intersectioinal) position.

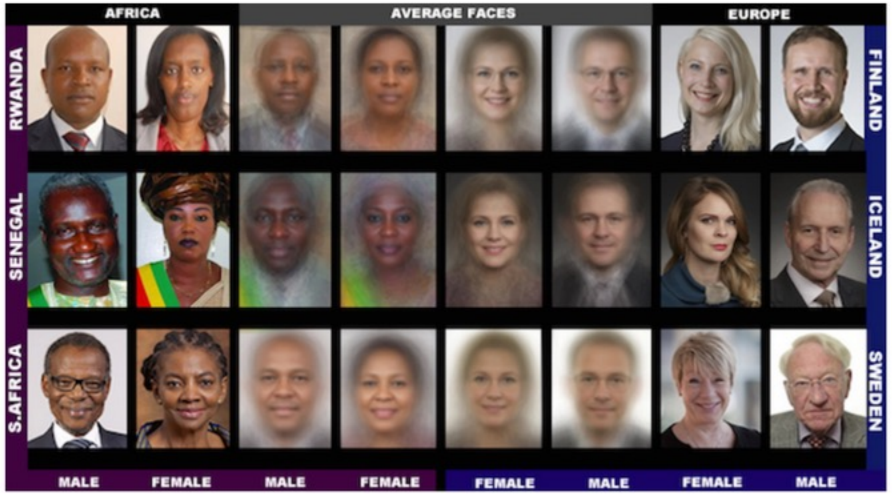


Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*

Joy Buolamwini JOYAB@MIT.EDU
MIT Media Lab 75 Amherst St. Cambridge, MA 02139

Timnit Gebru TIMNIT.GEBRU@MICROSOFT.COM
Microsoft Research 641 Avenue of the Americas, New York, NY 10011

Editors: Sorelle A. Friedler and Christo Wilson



And now back to code

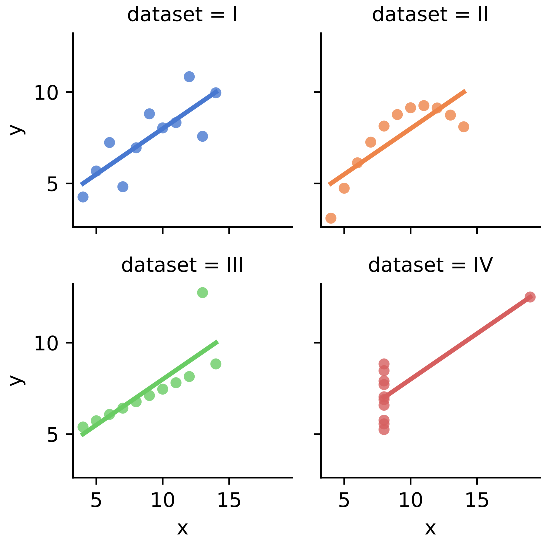
Chapter 9 is the most complex chapter in the book as it tries to do a few things at once, but the most important is:

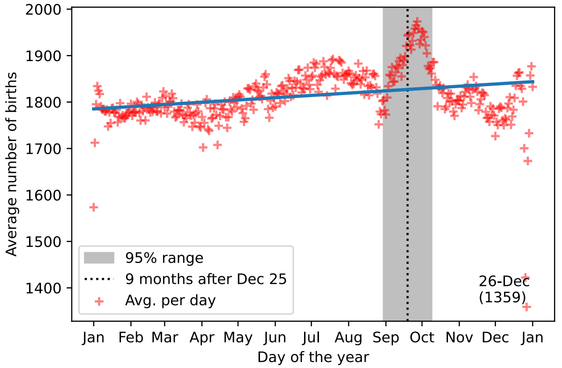
- Practically illustrate how to reason between observations of data and expectations for that data using visualisations and basic statistics.

I cannot cover complex statistics in such a chapter or complex visualisations. Nor should the chapter simply be an index of all possible stats routines or visualisations.

But we do not need complex statistics and visualisations to get us started and to make this point. We only need a single distribution.

In fairness, in the latter parts of the chapter I look at multiple distributions together





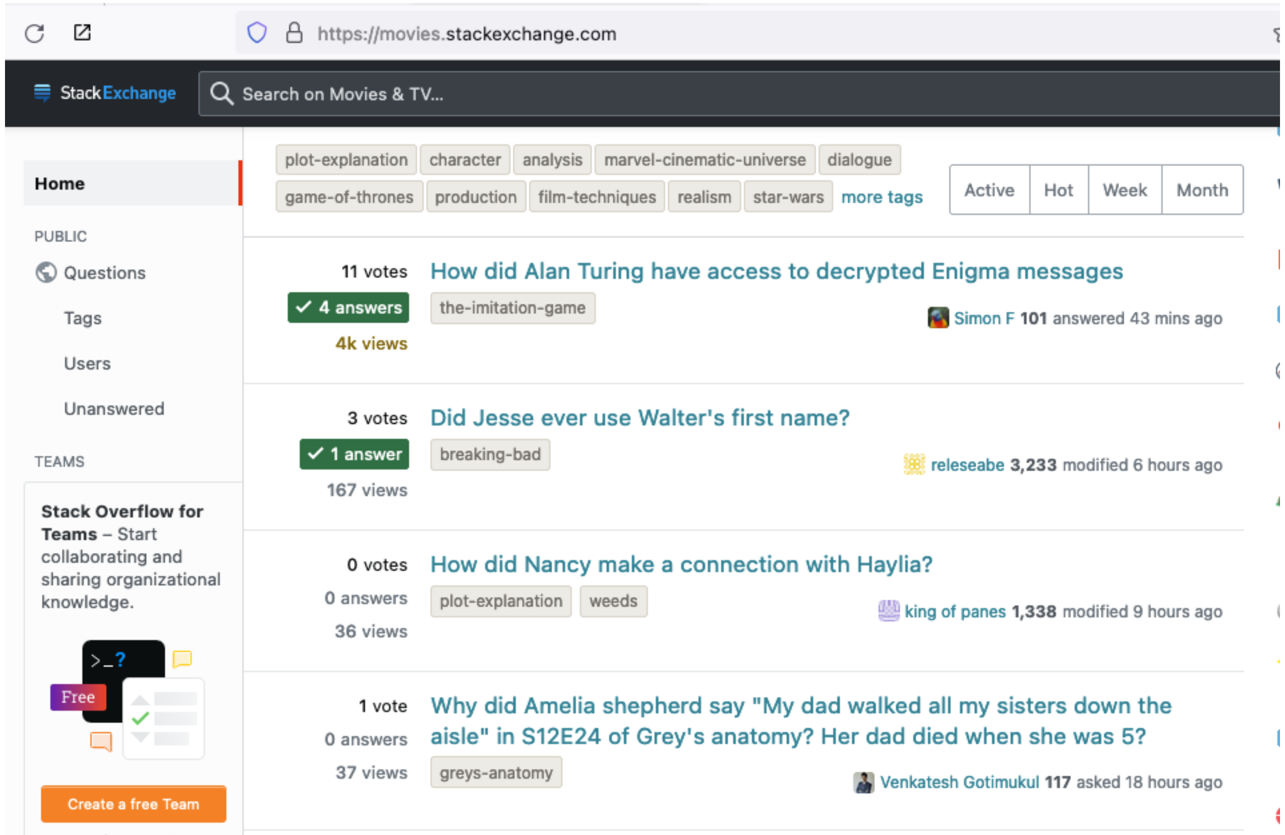
Social Data Science Skills in Practice

The last five chapters in the book all focus on data science skills in practice.

We start with **Chapter 10** on cleaning data. It covers the basics of how we get data from a raw unstructured form into one where the data itself is in a meaningful form for future analysis.

I choose Stack Exchange data.

It's open access, XML, and can be useful for a variety of tasks. The goal is not to discover an exciting finding but to understand how a data set is structured. And it's likely to still be live data available for processing for several years.



https://movies.stackexchange.com

StackExchange — Search on Movies & TV...

Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.

11 votes ✓ 4 answers 4k views — How did Alan Turing have access to decrypted Enigma messages — the-imitation-game — Simon F 101 answered 43 mins ago

3 votes ✓ 1 answer 167 views — Did Jesse ever use Walter's first name? — breaking-bad — releseabe 3,233 modified 6 hours ago

0 votes 0 answers 36 views — How did Nancy make a connection with Haylia? — plot-explanation weeds — king of panes 1,338 modified 9 hours ago

1 vote 0 answers 37 views — Why did Amelia shepherd say "My dad walked all my sisters down the aisle" in S12E24 of Grey's anatomy? Her dad died when she was 5? — greys-anatomy — Venkatesh Gotimukul 117 asked 18 hours ago

The ’silo’ chapters: 11,12,13, and 14

In Chapter 10, I suggested some core ways in which social data is structured: by time, space, relation, and semantics. Each of these is given its own chapter:

- **Language** is covered via **Chapter 11** on Natural Language Processing.

- **Time** is covered via **Chapter 12** on Time Series analysis.

- **Relations** are covered via **Chapter 13** on Social Network Analysis.

- **Space** is covered in **Chapter 14** on Geography.*

* For the chapter in Geography we make use of OxCOVID data instead of Stack Exchange.

Data is the new Garden

The final chapter is a short concluding chapter reflecting on next steps.

I first reflect on the phrase 'data is the new oil', which I find not only offensive but inaccurate. Oil is taken from one place and used elsewhere, but data is recirculated among the very phenomena from where it was taken.

We can think of data more like gardening: we deliberately structure things in a garden, tend to it, prune or harvest it. Take too much and things won't grow on their own, leave it to wild and it's hard to manage. But the process is always cyclical and continual, not extractive and hopefully not exploitative.

And then finally I give pointers to some key conferences, texts, and future sites of learning.

Lessons learned and things excluded

This is not a book on machine learning or artificial intelligence. We only barely scratch the surface of such techniques in the latter chapters.

In the NLP chapter we cover the logic of Naive Bayes, but only as a teaser for thinking about how we classify data as a practice. In the networks chapter we look at community detection, but again, only superficially.

We do not really look at multivariate regression or sophisticated statistical models.

These are deliberate to keep things simple. But were I to do the book again, I think I'd like a chapter on visual analysis, either through object detection or classification. While I have a chapter on scientific reasoning it would be good to clarify for people the steps in training and testing ML models.

Returning to the science of measurement

The book introduces the ideas, tools, and some theories related to the measurement of social life through data.

We do not simply take data from the world, but produce data in the world.

The types of data produced have a structure of their own as well: Data can be nested, sequential, and/or relational.

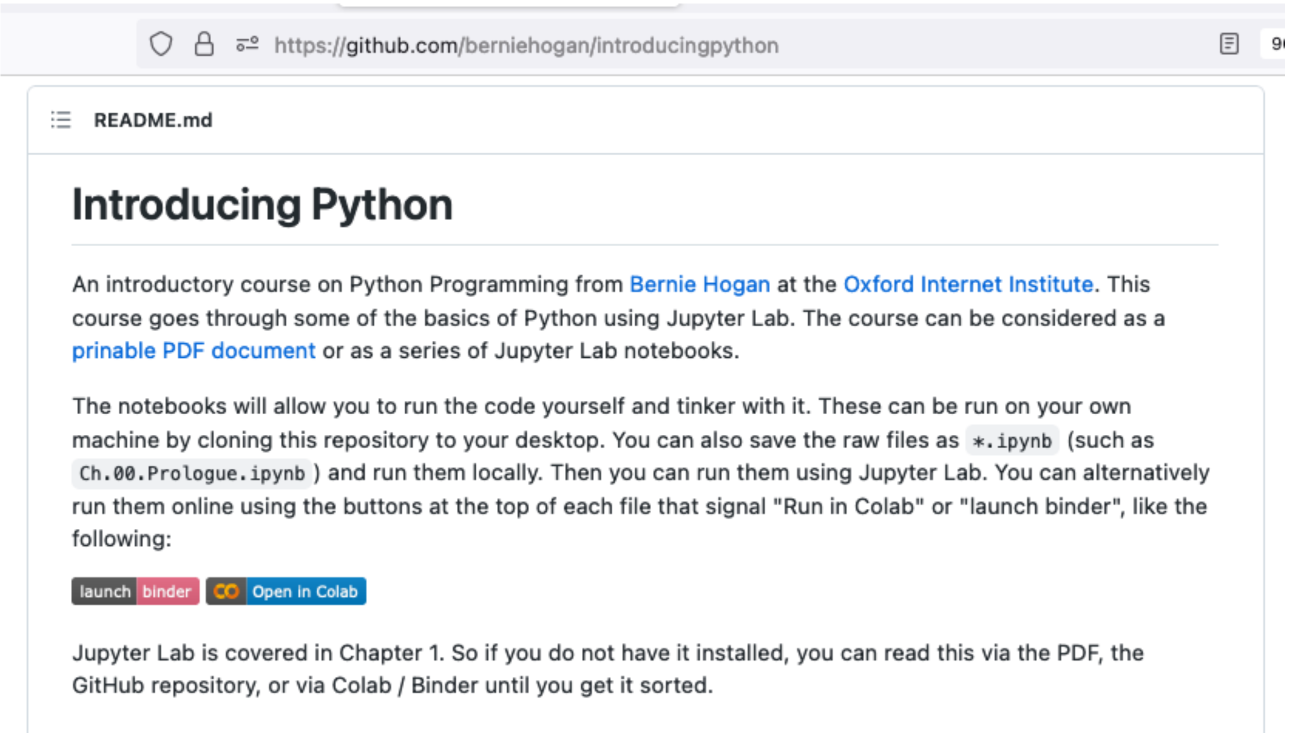
Data has correspondence with the world: it measures time, space, people, their relationships, and the content of their behaviour and communication.

What we are ultimately interested is not data, per se, but phenomena: social cohesion, support, conflict, hate speech, etc…As we measure these things we do not simply understand them but change them. It is up to us to do this in a way that is meaningful, fair, ethical and intelligible.

What you need for this book

Some introductory Python and Jupyter Lab (ideally from https://anaconda.com/).

How to get intro to Python? One example: https://www.github.com/berniehogan/introducingpython/



Introducing Python

An introductory course on Python Programming from Bernie Hogan at the Oxford Internet Institute. This course goes through some of the basics of Python using Jupyter Lab. The course can be considered as a prinable PDF document or as a series of Jupyter Lab notebooks.

The notebooks will allow you to run the code yourself and tinker with it. These can be run on your own machine by cloning this repository to your desktop. You can also save the raw files as `*.ipynb` (such as `Ch.00.Prologue.ipynb`) and run them locally. Then you can run them using Jupyter Lab. You can alternatively run them online using the buttons at the top of each file that signal "Run in Colab" or "launch binder", like the following:

Jupyter Lab is covered in Chapter 1. So if you do not have it installed, you can read this via the PDF, the GitHub repository, or via Colab / Binder until you get it sorted.

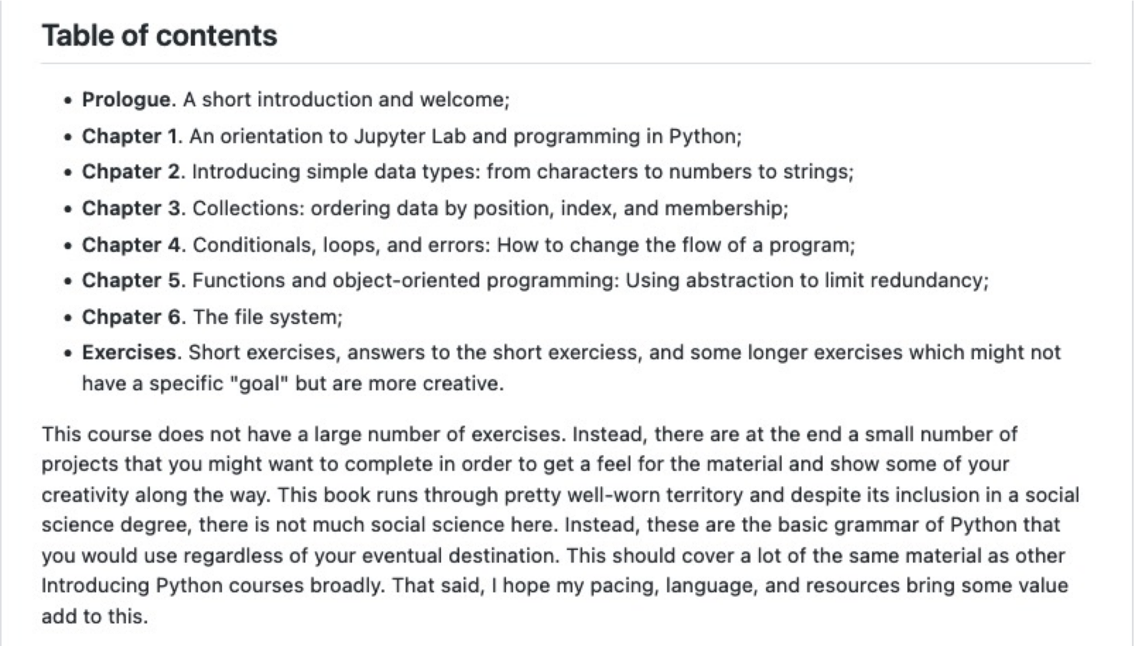


Table of contents

- **Prologue**. A short introduction and welcome;
- **Chapter 1**. An orientation to Jupyter Lab and programming in Python;
- **Chpater 2**. Introducing simple data types: from characters to numbers to strings;
- **Chapter 3**. Collections: ordering data by position, index, and membership;
- **Chapter 4**. Conditionals, loops, and errors: How to change the flow of a program;
- **Chapter 5**. Functions and object-oriented programming: Using abstraction to limit redundancy;
- **Chpater 6**. The file system;
- **Exercises**. Short exercises, answers to the short exerciess, and some longer exercises which might not have a specific "goal" but are more creative.

This course does not have a large number of exercises. Instead, there are at the end a small number of projects that you might want to complete in order to get a feel for the material and show some of your creativity along the way. This book runs through pretty well-worn territory and despite its inclusion in a social science degree, there is not much social science here. Instead, these are the basic grammar of Python that you would use regardless of your eventual destination. This should cover a lot of the same material as other Introducing Python courses broadly. That said, I hope my pacing, language, and resources bring some value add to this.

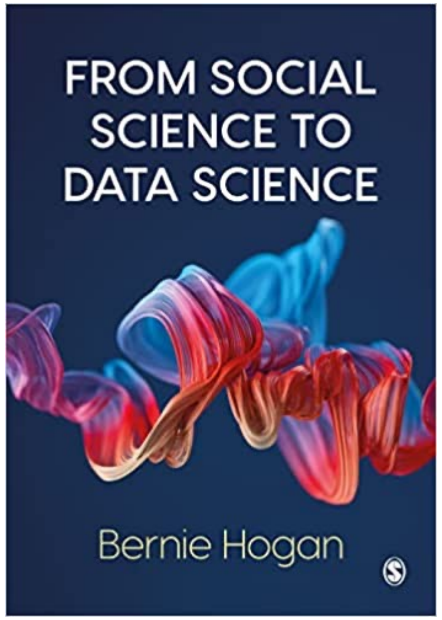
Book Resources

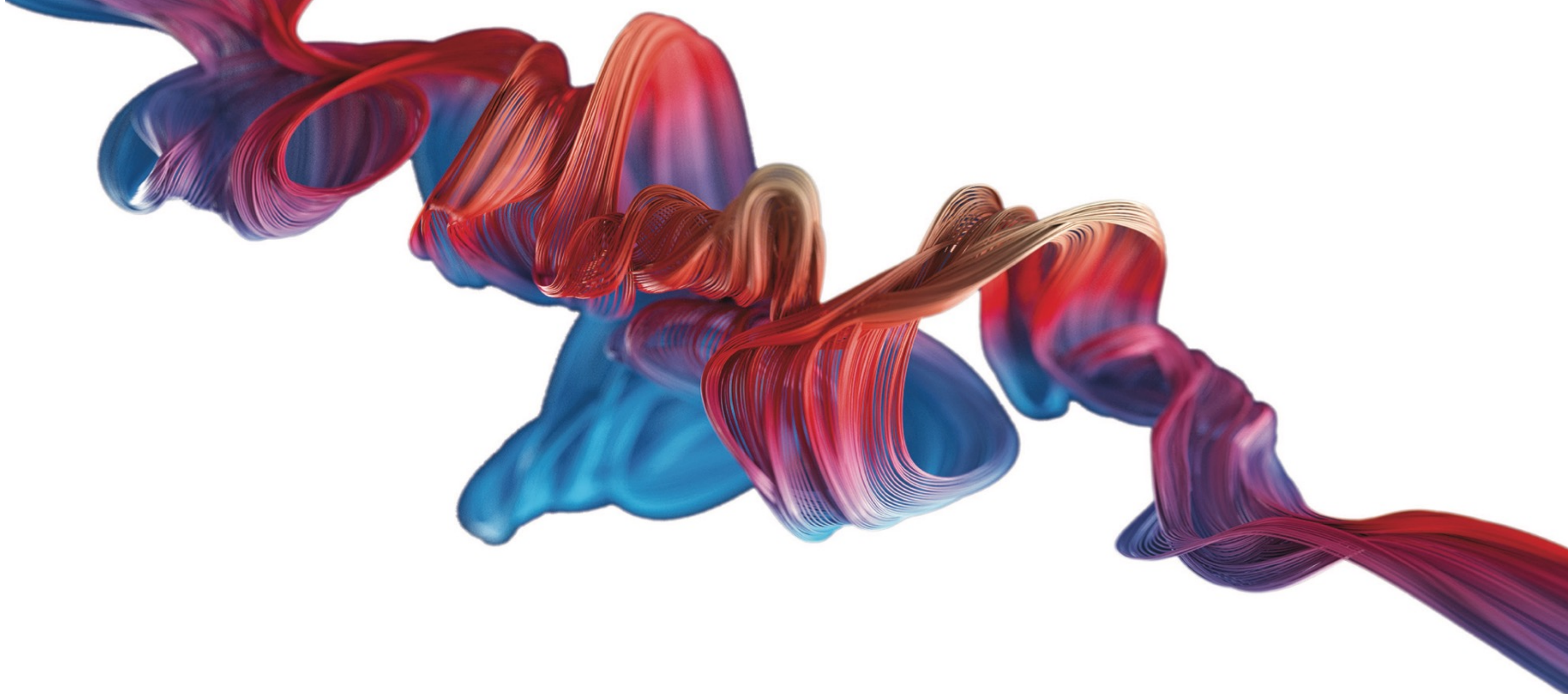
GitHub – https://www.github.com/berniehogan/fsstds

Online – The book is available as an eBook or as a paper copy.

Coming shortly – The Sage learning portal for this book will have PowerPoint slides and multiple choice quizzes by the end of this month. They are mostly done and drafts can be available by request.

Also partially available – I have an unlisted link to many of the chapters using these PowerPoint. The link is on the GitHub page. The rest of the lectures will follow the PowerPoints by Mid-March.





Thank you

Available at all kinds of bookstores around the world.

Here's my Amazon affiliate link: https://amzn.to/3GEj6HT

Feel free to also check Sage's website and order direct (and even contact me for a discount for direct orders): bernie.hogan@oii.ox.ac.uk