Data Science Using Python

What Does "Data Science" mean?

(1) Collecting large amounts of data (Big Data)

Via computers, sensors, people, events …

(2) Doing something (Useful) with it

Making decisions, confirming hypotheses, gaining insights, predicting future …

- More specifically,

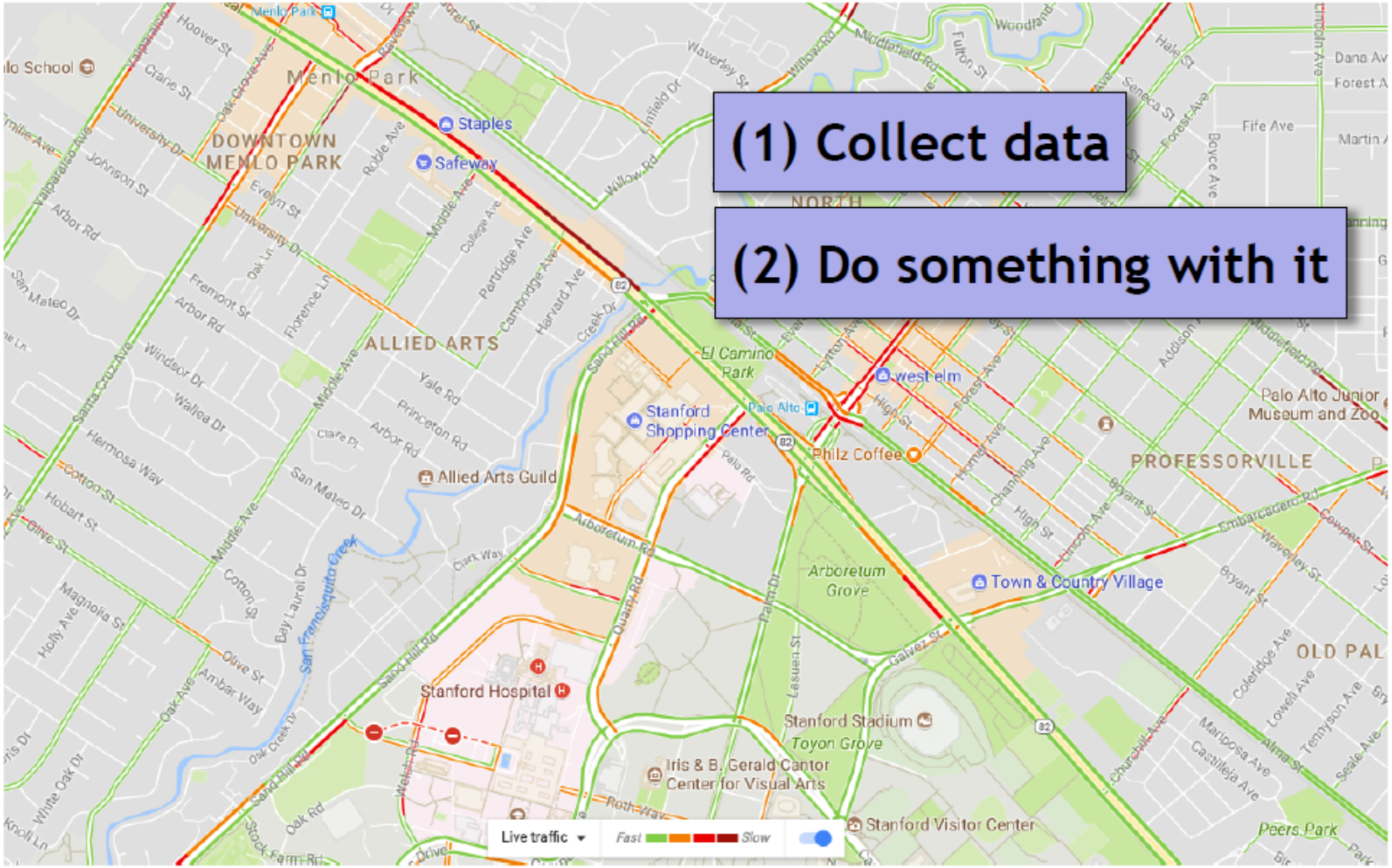
  Data Science = Going from (1) to (2)

- **Data Science is Here to Stay**
  - Ability to collect data will only increase
  - Ability to analyze data will only improve

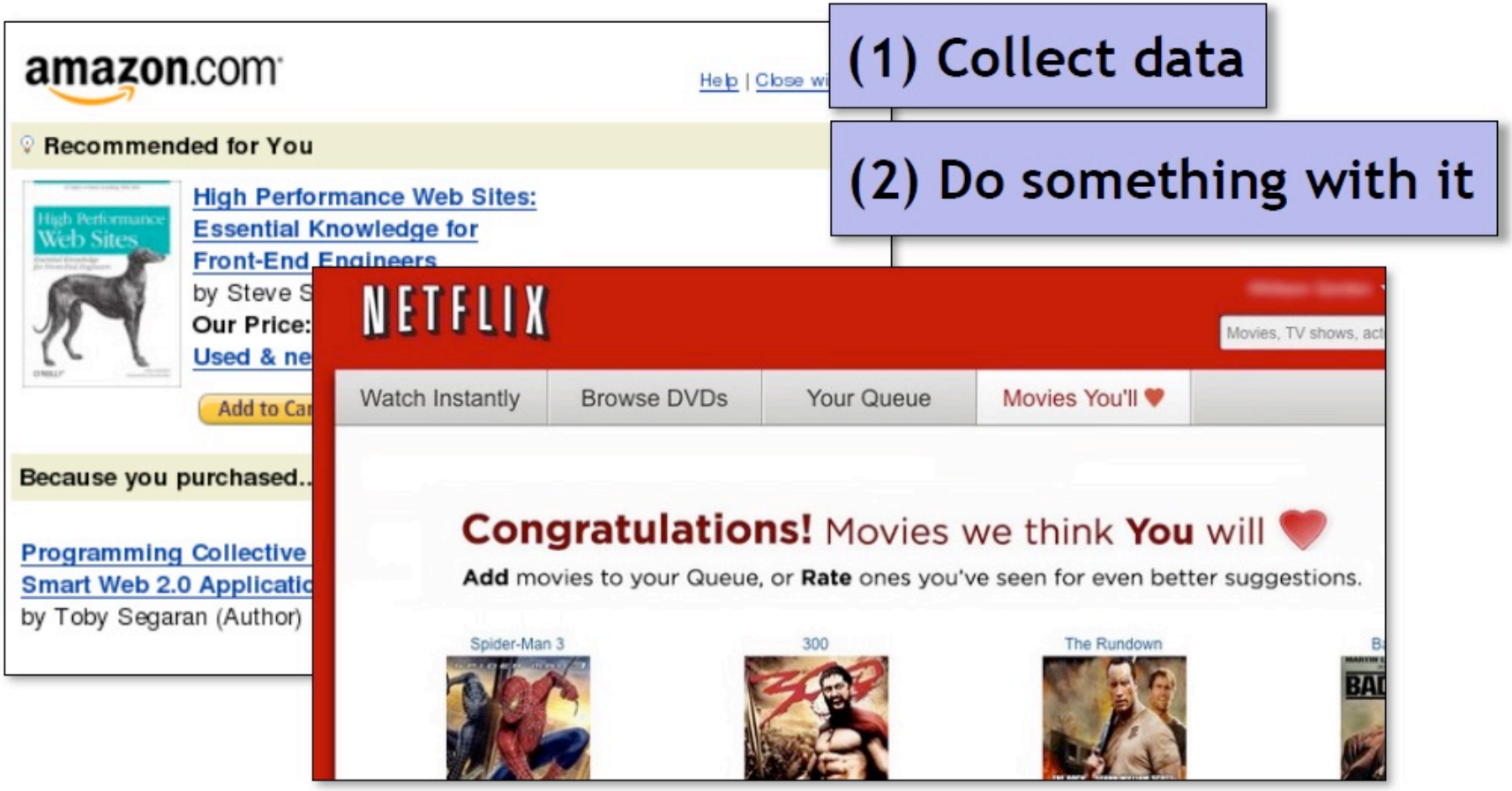
How Big is the Data?

- Complete works of William Shakespeare
  - 5 megabytes
- USA Library of Congress
  - 10 terabytes (2 million Shakespeares)
- Uploaded to Facebook daily
  - 1 petabyte (200 million Shakespeares)
- Produced by humanity daily (Now)
  - 2.5 exabytes (500 trillion Shakespeares)

Applications of "Data Science": Traffic



Applications of "Data Science": Recommender System



+ music, news, friends, romantic partners, and many more!

And Many More

- Weather prediction
- Medical diagnosis
- Financial markets
- Resource management
- Computational social science
- Smart buildings and cities
- The list goes on and on and it’s still early days.

Data Science Tools and Techniques

- Basic Data Manipulation and Analysis
  - Performing well-defined computations or
  - asking well-defined questions (“queries”)
- Data Mining
  - Looking for patterns in data
- Machine Learning
  - Using data to make inferences or predictions
- Data Visualization
  - Graphical depiction of data
- Data Collection and Preparation

Basic Data Manipulation and Analysis

Performing well-defined computations or asking well-defined questions ("queries")

- Av… r ea…
- Nu… by fer…
- Fre… ing specific symptoms
- The ten stocks whose price varied the most over the past year

- Spreadsheets
- Relational (SQL) database systems
- "NoSQL" / scalable systems
- Programming languages with big-data support (e.g., Python, R)

Data Mining

Looking for patterns in data

- Items X,Y,Z are bought together frequ
- Peop ke movie Y
- Patie edicines X and Y dicine Z
- Students going to the same university are frequently online friends
- Wealthier people are moving from cities to suburbs

- Frequent item-sets
- Association rules
- Specialized techniques for graphs, text, multimedia

Machine Learning

Using data to make inferences or predictions

- Customers wh… over age 20 are likely to respo… rtisement
- Students with… are predicted to do well on the SAT

- Regression
- Classification
- Clustering

- Roughly: Basic data analysis and data mining give answers from the available data, while machine learning uses the available data to make predictions about missing or future data

Data Visualization

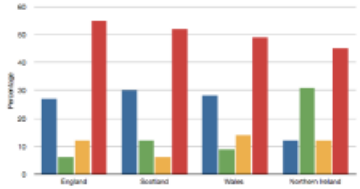
~~A picture is worth a thousand words~~

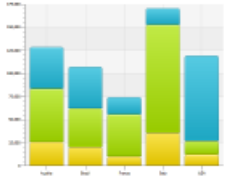
A picture is worth trillion data points

Basic Data Visualizations

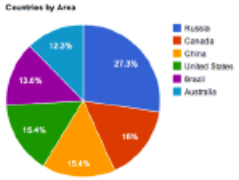
Don’t underestimate the power of basic visualizations

- Bar charts

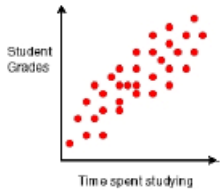


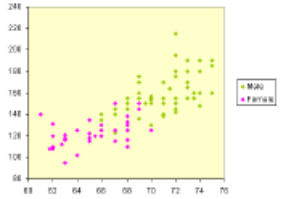


- Pie charts

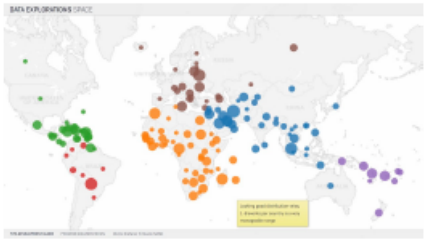


- Scatterplots

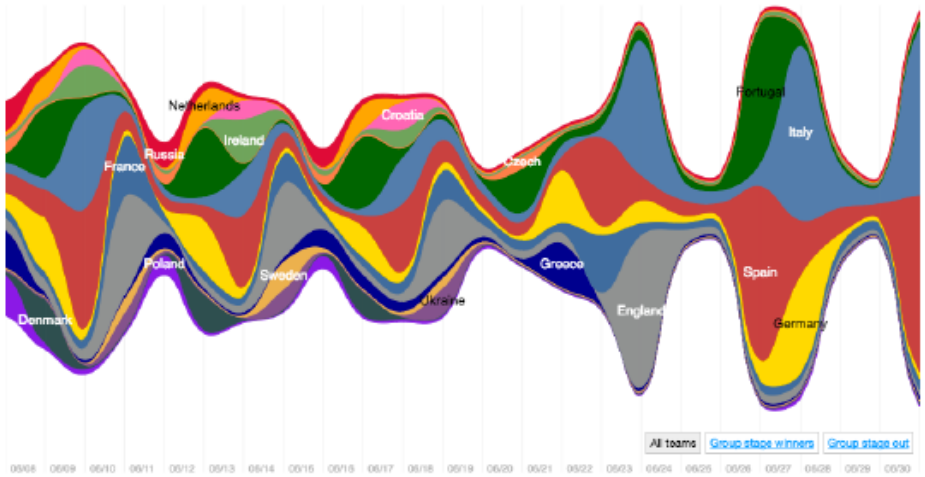


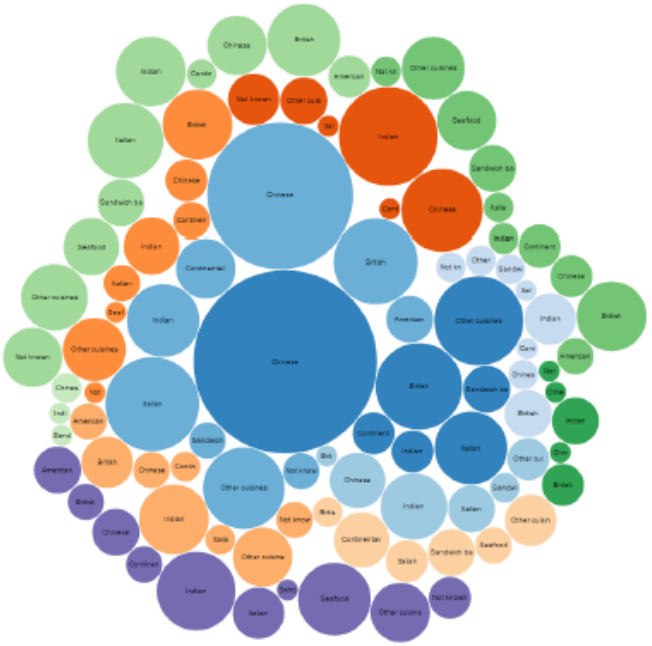


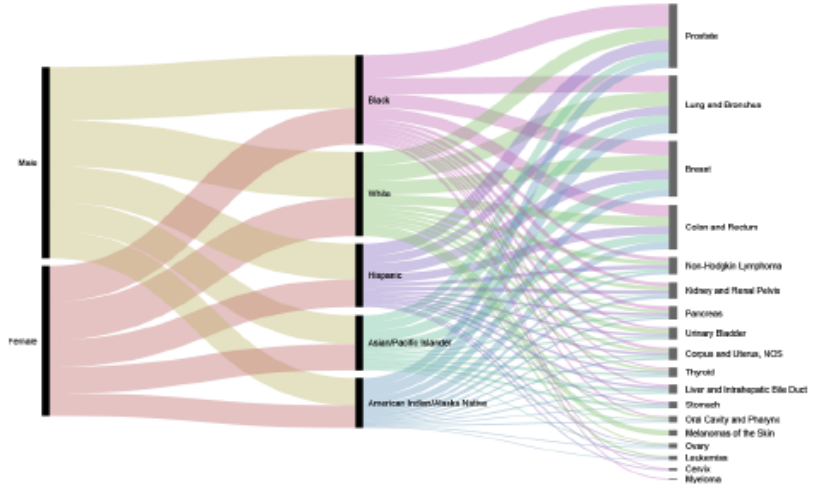
- Maps



Advanced Data Visualizations







Data Collection and Preparation

The "dirty" secret of Big Data

- Extracting data from difficult sources
- Filling in missing values
- Removing suspicious data
- Making formats, encoding, and units consistent
- De-duplicating and matching

Data preparation often consumes 80% or more of the effort in a Big Data project

Languages, Systems, Platforms

- Spreadsheets
  - Surprisingly versatile and powerful for data analysis tasks, but not truly big data
- Programming languages with big-data support
  - R Language – powerful statistical features
  - Python – general-purpose language with R-like add-ons (Pandas, SciPy, scikit-learn)
- Data processing in the cloud
  - Amazon Web Services, Google Cloud, Microsoft Azure

Why do you care?

Big data and social science

Social scientists find themselves facing exponentially larger data sets without suitable tools to deal with them.

Python

- Python is an open-source, general-purpose scripting language.
- Open-source
  - Built by a community
  - Maintained by a community
  - Free to use for all
- General Purpose
  - If you're doing it on a computer and there's some repetitive element, then you can automate it in Python.
  - Python isn't limited to Data Science, but it's very popular with data scientists!
- Scripting
  - Series of commands to automate some task.
  - Like a pipeline: takes some inputs, does some things to these inputs, and gives back some outputs.
  - It's good to keep the input-output framework in your head.

Python (Cont.)

- Scripting
  - Series of commands to automate some task.
  - Like a pipeline: takes some inputs, does some things to these inputs, and gives back some outputs.
  - It’s good to keep the input-output framework in your head.
- Language
  - Python is a language, and not an application.
  - Practical difference for you:
    - most applications provide you options to select from.
    - languages require to generate commands from accepted rules.
  - Bottom line is that you can do nearly anything with Python!

What Can I use Python For?

- Clean up my messy data!
- Run analyses with (hundreds of) millions of data points it won't fit into an excel spreadsheet!
- I want to automate downloading several decades of newspaper articles!
- I want to create beautiful (interactive) visuals to accompany my analyses!
- I want to uncover hidden structures linking parliamentary committees!
- I want to track the changing meaning of a concept over a century!
- Again: any repetitive task done on a computer can be automated with Python.

Tools of the Trade

- Anaconda
  - Environment and software manager.
  - Can be used from the command line (cli) or browser-like interface (anaconda-navigator).

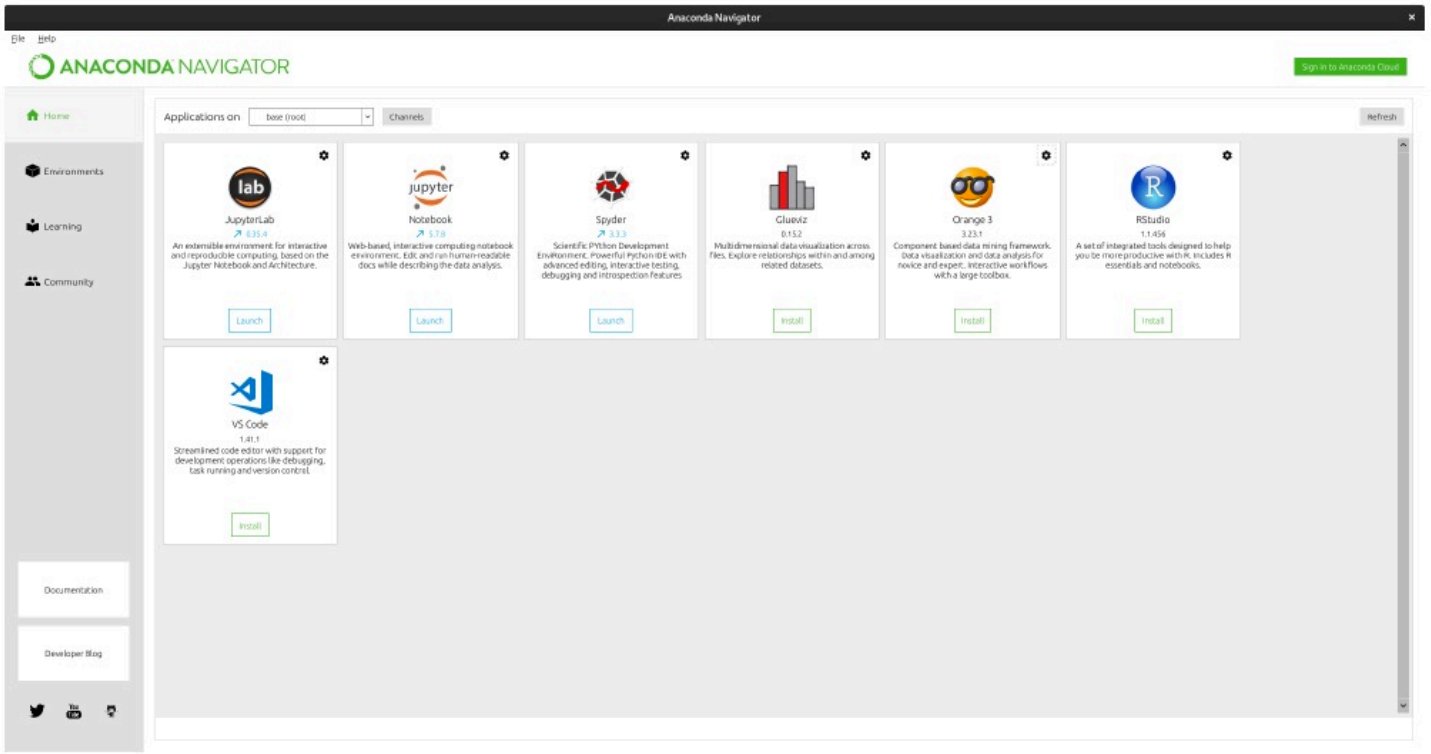
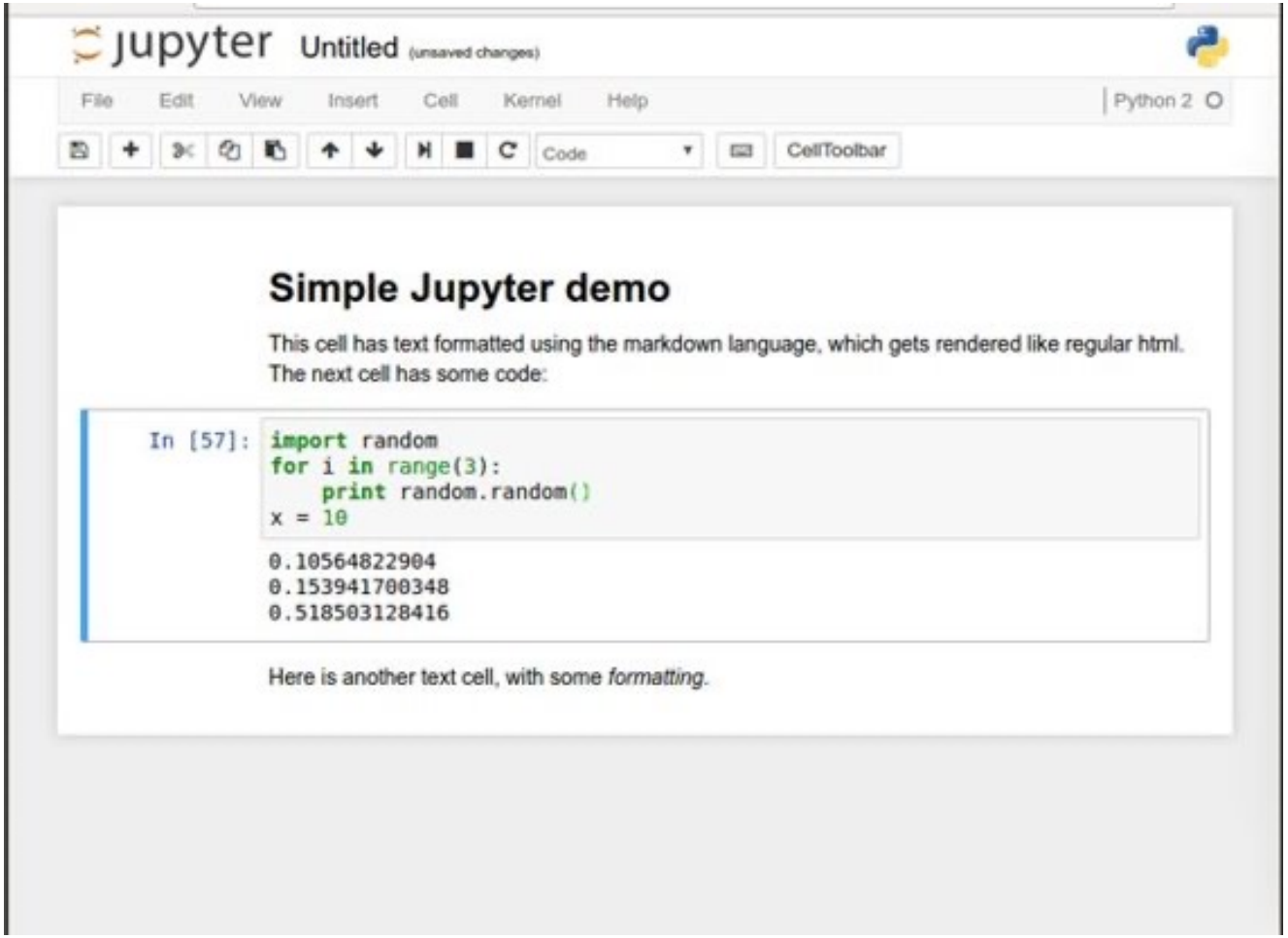


Figure 1: Anaconda Navigator

Tools of the Trade

- Jupyter Notebook
  - Interactive code editor.
  - Easy to use environment
  - Web-based
  - Combines both text and code into one



Basic Workflow

1. Open up Anaconda Navigator
2. Open up Jupyter Notebook
3. Navigate to relevant directory
4. Open pre-existing notebook, or create new one.
5. Start coding

Coding

Variable Assignment

- Variables can be assigned with =

  x = 1

  y = 2

  Z = 30.5

  Name = “John”

- There are some rules for variable assignment:
  - Variable names cannot contain spaces
  - The first letter of the variable cannot be a number or symbol

Data Types

- There are four basic data types in Python. These are:
  - String
    - A sequence of characters.
    - Behaves like a sequence; can be indexed with [index]
  - Integer
    - Whole numbers.
    - Can be positive or negative
  - Float
    - Decimal numbers.
    - Behave unexpectedly. Remember: 0.1*3==0.3 returns False
  - Boolean
    - True/False
    - Behaves similarly to integers 0 and 1

Tabular Data

- Tabular data consists of an ordered arrangement of rows and columns.
  - A common example is a spreadsheet.
  - In this class will often be referred to as a matrix X_{ij} of i observations of j variables.
    - Note all columns must be of equal length (i), and all rows must be of equal length (j).
- In this class we are primarily focused on tabular data. If your data is not tabular, you may want to figure out some way to coerce it to a tabular format because most statistical/ML models assume tabular data.

Tabular Data

Name	Location	Age	M/F	Covid Positive?
Andy	Greensboro	19	M	Yes
Julia	Winston-Salem	21	F	No
Chris	High Point	22	M	No
Debra	Winston-Salem	20	F	Yes
John	Clemmons	19	M	
Barbara	Winston-Salem	21	F	Yes
Maria	Greensboro		F	
James	High Point	20	M	Yes

Note the missing values: NA (no observation available)

Data Formats

- csv ("comma separated-values") is an extremely common tabular data storage format.
  - Values are delineated by a special character, usually a comma.
  - Has no built-in data types; this needs to be inferred by the parser.
- json (JavaScript Object Notation) is also extremely common, especially when using web data.
  - Stores information as a mixture of key-value pairs and arrays
  - Working with json usually requires us to coerce hierarchical data to tabular data.
  - There are reasons to not use commas, especially when working with text data.

Pandas

- pandas is a very popular library for working with tabular data structures in Python. Before we start using it, let's go over some of the ways it can be useful to you as a social science researcher.
  - Analysis usually takes <30% of your time.
  - >50% of your time will be spent reading, cleaning, checking, storing, and cursing your data.
  - Data cleaning is meticulous work, but that doesn't mean you can't be efficient.
- Advantages of Panda
  - provides fast, flexible data structures
  - extensive array of convenient functions
  - compatible with most data science libraries and data types
- When you should not use Panda
  - Your data is not coercible to a tabular structure.
  - When your dataset is too large to load in your computer's memory (or loading it uses most of your RAM).

Using Pandas

- Import panda to the program

```
import pandas as pd
```

- Data I/O
  - pandas has methods for reading in data from various formats.
  - Read csv file

```
data = pd.read_csv('Project_dataset.csv’)
```

- pandas contains two native data containers:
  - pandas.DataFrame: A two-dimensional* labelled matrix
  - pandas.Series: A one-dimensional labelled array
  - *Can be higher-dimensional with the use of hierarchical indices

Data Exploration

- When working with data, your first step should always be getting to know the data. Ask questions like:
  - What are the dimensions of the dataset?
  - What information is contained in the columns? in the rows?
  - How is my data organized?
  - What data types are each of the columns? Is this expected?
  - How sparse is my data? (Looking for NAs)
  - There’s nothing worse than trying to debug code that’s taken hours to write only to discover that the problem lies in your data!

Query the Data

1. What is the maximum GPA?
2. What is the average age?
3. How many female students?
4. How many students have less than average GPA?
5. How old is the highest GPA holder?
6. What is the average age for Covid=yes vs. Covid=no students
7. What is the maximum GPA for male vs. female students

Summary Functions

- Part of the function and appeal of data analysis is to reduce millions of data points to a few summary numbers that capture key information that you are looking for.
- Summary functions do this: they summarize a large number of observations to one or a few values that tell you what you need to know. They are also known as statistics.
- Basic examples include mean, sum, variance, skew, but also more advanced statistics such as regression coefficients, Kolmogorov-Smirnov tests, and even the output of machine learning algorithms!

Grouped Summaries

- Pandas provides an extremely efficient and clean method for doing group summaries, but the syntax can be difficult to understand.
- To conduct grouped summaries, we use the following syntax:

```
data.groupby("group_col").agg({"value_col": summary_func()})
```

  - group_col is the column we are grouping over.
  - value_col is the column that contains the values we will be applying grouped summary functions to.
  - summary_func is the function that that is applied to each group.

Data Visualization

- The aim of much of data science is to understand the whole picture of your data.
- If you can do this without reading your entire dataset, all the better!
- When making data visuals, I think it’s helpful to remember that they are, in many ways, a form of summary.
- Visualizing data is not just about communicating results; it is also a powerful tool for you to understand important features of your own data.
- matplotlib is the primary library for building data-based visuals in Python.