Session 01
Welcome to Data Science

wifi: GA-Guest, yellowpencil

Today’s session plan

1800-1820	Introductions
1820-1830	Course structure & classroom culture
1830-1845	What is data science?
1845-1900	Technical set up
1900-1920	Break
1920-2000	The data science workflow
2000-2015	Introducing Python & Jupyter
2015-2100	Python practise
Homework: More Python practise	

At the end of the session, you will be able to ...

Understand the steps in the data science workflow

Apply your fundamental Python skills to perform simple calculations

Launch Anaconda and work with Jupyter Notebook

Understand frequently used data science terminology

Data Science Part Time

About us

About GA

Partner Exercise:

Getting to know you

15 minutes

Turn to your partner

Introduce yourself, and tell them about:

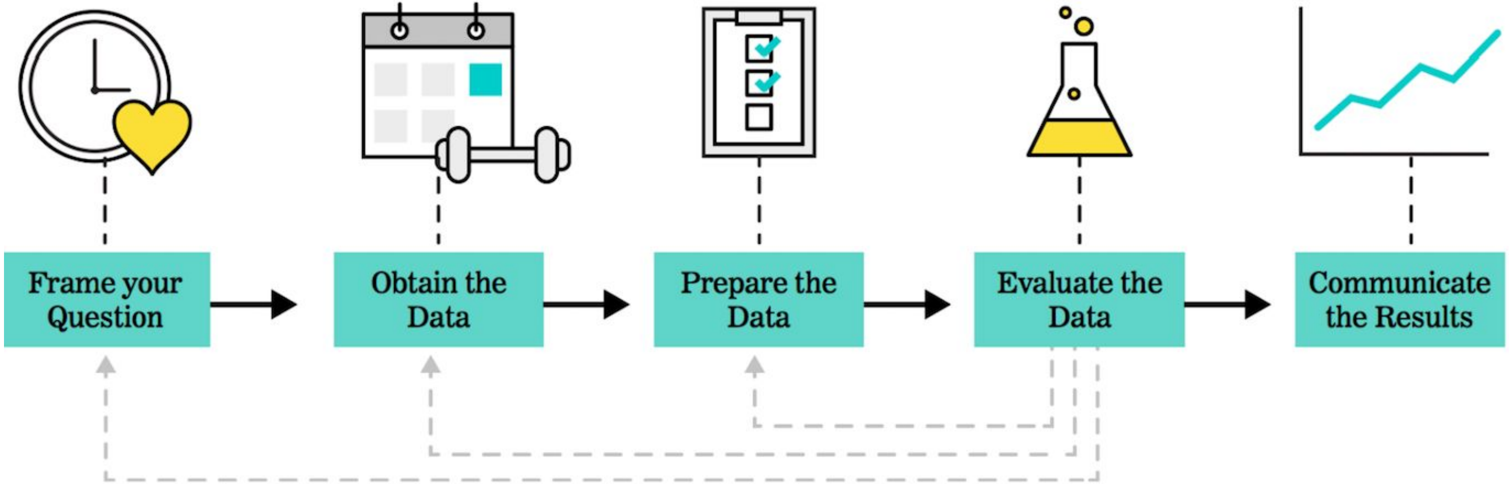
- Why you’re interested in data science
- What you’re most looking forward to learning about
- What you’re having for dinner tonight

You’ll then introduce your partner to the rest of the class

Data Science Part Time

Course structure

Data science workflow



Course structure

Unit 1

Data Foundations

Python and Pandas basics, Git, Bash

Unit 2

Working with Data

Statistics, visualisation, exploratory data analysis

Unit 3

Modelling

Regression, classification, K-nearest neighbours

Unit 4

Applications

Natural language processing, decision trees, k-means clustering

Individual final project work

Teaching Philosophy

Each session will involve

- Theory
- Paper exercises
- Practical Python exercises
- A halfway break

Data Science Part Time

Classroom culture

Partner Exercise:

Good teaching experiences

Turn to your partner.

Tell each other about your best classroom experience. What made it great?

Report your partner’s answer back to the class.

Discussion:

Good teaching experiences

As a class, let’s discuss how we can translate your best classroom experiences into a set of expectations.

- Your expectations of us
- Our expectations of you

Your To Do List

- Be engaged and ask questions
- Communicate with us
- Don’t be afraid of breaking things
- Don’t worry about writing things down
- Try not to get distracted by your inbox or phone calls
- Help your peers out, don’t compete with them

Our To Do List

- Address your questions in person or via Slack
- Hold office hours on request
- Give feedback about your progress on request
- Take feedback about the course structure or content, through weekly surveys
- Explain concepts as many times as you need

Communication

Slack is our main communication channel outside and during lessons.

We’ll use Slack to:

- Post resources and interesting reading/listening/watching materials
- ‘Park’ questions to be addressed later

You can use Slack to:

- Communicate directly with us to set up office hours, let us know if you’re running late, etc
- Post your own interesting links to the class channel

Computers Out:

Slack check

Let’s make sure everyone is set up on Slack

1. Check your emails for an invitation to ga-ldn-datascience.slack.com
2. Follow the steps to sign up
3. Check you can see the ds37 channel

Feedback

We’ll circulate links to feedback surveys at the midpoint and endpoint of the course.

Optional weekly surveys will also be circulated at the end of every week.

We’ll begin every session with ‘standup’. This is a chance for you to discuss with your peers how you’re getting on, what you’re enjoying and what you’re struggling with.

Data Science Part Time

What is data science?

Discussion:

What is Data Science?

- On your tables, agree on a definition for ‘data science’
- How does this differ from data analytics?
- Keep it concise, about the length of a Tweet!

Data Science

Producing insights, value and predictions based on data.

Skills

- Mathematics and statistics
- Programming skills (R, Python)
- Domain knowledge

Data Analytics

Analysing complex data and communicating results.

Skills

- Statistical analysis
- Communicating statistical results
- Tools (Excel, Power BI)
- Data access and query (SQL)

Group Exercise:

Matching Data Roles

On your tables, match each of these job descriptions to one of these roles: **Advanced Analyst, Data Engineer, Machine Learning Engineer, Quantitative Researcher**

Responsible for acquiring, organizing, and delivering complex data and making it reliably accessible to users.

Skills:

- Database storage and retrieval
- Big data storage and processing
- Data modeling
- Coding (Python, Java, SQL)

Responsible for understanding, modeling, and interpreting complex data quantitatively.

Skills:

- Statistical analysis
- Predictive modeling
- Communicating statistical results
- Statistical languages (SAS, R, SQL)
- Combining datasets

Responsible for creating, implementing, and productionalizing predictive models.

Skills:

- Databases
- Big data
- Machine learning
- Software engineering
- Coding (Python, Java, SQL)

Responsible for analyzing complex data and communicating results.

Skills:

- Statistical analysis
- Communicating statistical results
- Tools (Excel, Tableau, Data Dashboards)
- Data access and query

Explain It Like I’m Five

Group Exercise:

Explain It Like I’m Five

One of the most challenging things as an expert can be taking complex concepts and making them easy to understand.

Each group will assigned one of these topics:

- Model
- Variable
- Independent variable
- Dependent variable

Research and write a simple definition that a five year old would understand (ELI5).

Terminology

Variable

Measurements of a single quantity, taken from many subjects e.g. company share price measured across 1000 companies

Dependent variable (or target/response/label/output)

A variable we want to predict e.g. company share price

Independent variable (or feature/predictor/covariate/input)

A variable we're using to predict something e.g. company profits

Model

A mathematical relationship between two variables. Models can be very simple, or very complex.

Discussion:

Terminology

In these situations, identify the independent and dependent variable:

- Modelling the effect of coffee consumption on a student’s final grade
- Predicting a company’s share price using their profits
- Predicting a company’s gender pay gap using the proportion of women on the company’s board and the company sector

Terminology

Variables can be categorical or continuous

Continuous variables

Numerical variables, with an infinite number of values between any two values, e.g. there are an infinite number of values between a height of 2m and 2.1m, so height is a continuous variable

Categorical variables

Variables with a finite number of categories or groups, e.g. t-shirt size can be small, medium or large

Discussion:

Terminology

Are these variables categorical or continuous?

- Height
- Eye colour
- Weight
- Favourite Star Trek episode
- Birth month

Group Exercise:

What is Machine Learning?

On your tables, discuss:

What is machine learning?

Try to come up with a concise definition. Avoid using complicated jargon!

Machine Learning Is Used For...

- Making predictions about the future based on information about the past
- Finding relationships between variables
- Finding clusters and patterns in data
- Automating manual tasks

Supervised vs. Unsupervised Machine Learning

Supervised

Using independent variables to predict a dependent variable.

Vs.

Unsupervised

Finding groups and patterns in data.

Supervised Machine Learning

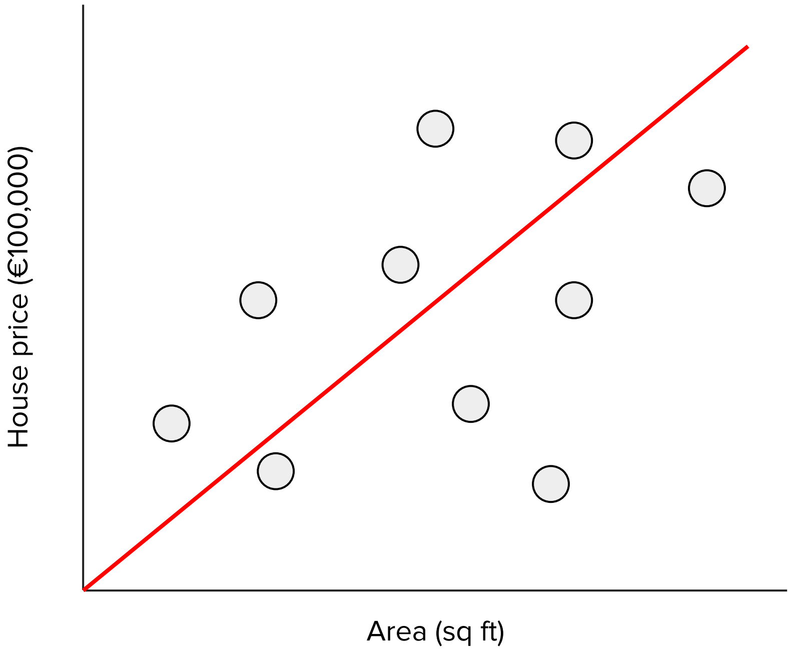
Finding a model that predicts **y (a response)** using values of **X (features)**

We use a **training dataset** which contains many corresponding values of **y** and **X** to find the mathematical relationship between our variables.

This is called **training** or **fitting** a model.

After our model has been trained, it can predict values of **y** when it’s given new values of **X**.

Supervised Machine Learning



Problem: predict house price using internal area

This is a **supervised learning** task because it fits the pattern of 'predict **y** using **X**.

We have **training data** (○) which gives us paired values of house price and area for 10 different houses.

We **fit a model** to this data. In this case, it's a very simple straight line (**linear**) model.

We can use the **trained model** (the red line) to predict the house price of houses that **aren't** in our training dataset, as long as we know their area.

Regression vs Classification

In **regression** tasks, we're predicting a **continuous** output

- Predicting someone's height
- Predicting the weight of a cow*

In **classification** tasks, we're predicting a **categorical** output

- Predicting the winning party in an election
- Predicting whether a customer will buy a product or not

Partner Exercise:

Machine Learning Terminology

For each of these machine learning problems, identify the **feature(s)** and **response**,
then decide whether it's a classification or regression task:

1. Predicting daily rainfall (mm) using temperature, air humidity and wind speed
2. Predicting a person's hair colour using their eye colour
3. Predicting a person's weight (kg) using their height (m)
4. Predicting a person's height using their shoe size and weight (kg)

Unsupervised Machine Learning

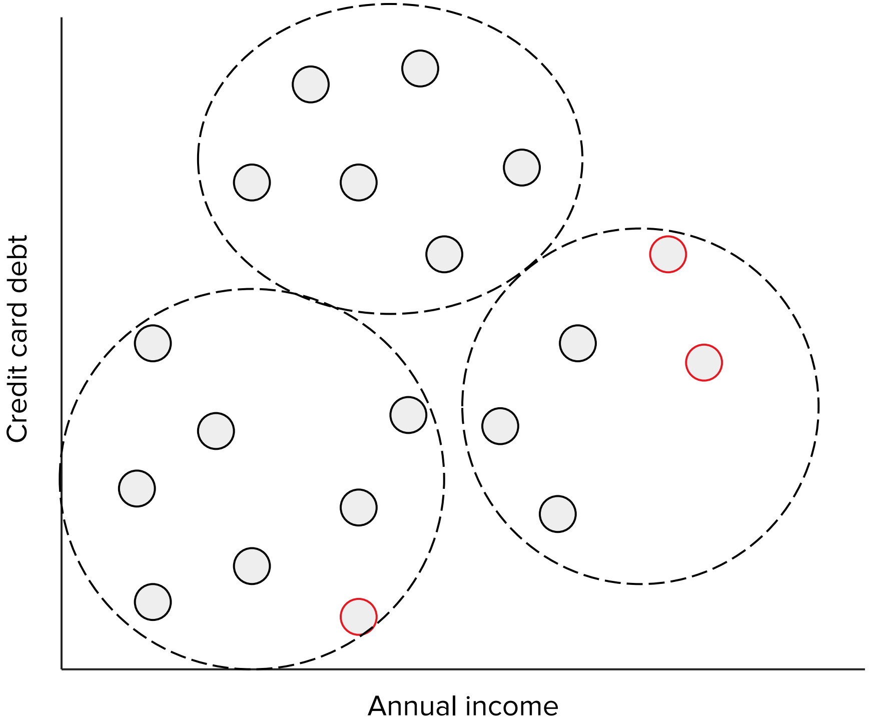
This is used for finding **clusters, groups** and **patterns** in datasets.

It’s also used for **dimensionality reduction**, or reducing the number of variables in a dataset without losing useful information.

We’re not explicitly predicting something.

The results of unsupervised learning often involve a lot of **domain expertise** and **human interpretation**.

Unsupervised Learning



Problem: Find meaningful clusters in customer data.

Use an **unsupervised** clustering method to find groupings in the data.

In this case, the algorithm has found three **clusters**.

Human interpretation is needed to make sense of what these clusters mean.

Discussion:

Case Study

Your client, a large bank, has heard about a lending startup that's doing interesting things with data. Your client is keen to take a similar approach.

Individually, take some time to read and think about this article from the LA Times, **'Some lenders are judging you on much more than finances'**

https://www.latimes.com/business/la-fi-new-credit-score-20151220-story.html?_amp=true

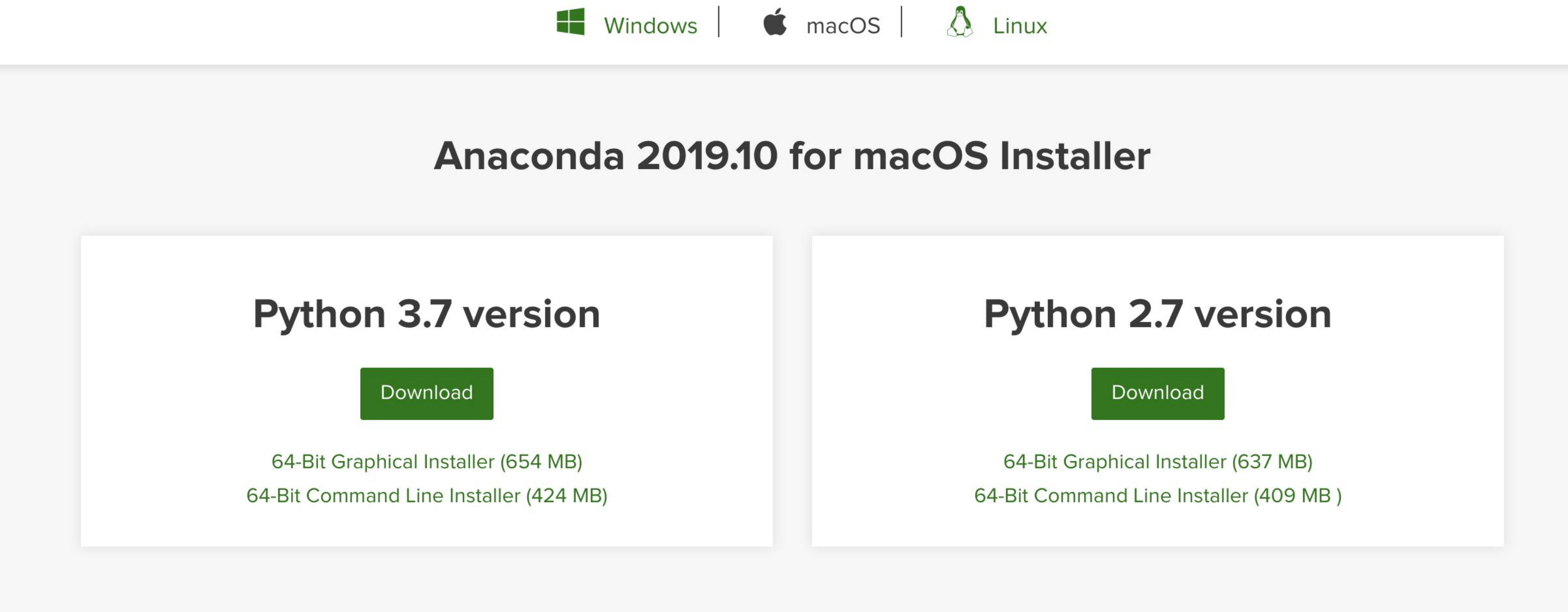
Then, be prepared to discuss with the class:

- What are the features and response in this case?
- Is the company performing a regression or classification task?
- What do you like or dislike about their approach?

Data Science Part Time

Technical set up

https://www.anaconda.com/distribution/#download-section



Anaconda 2019.10 for macOS Installer

Python 3.7 version

64-Bit Graphical Installer (654 MB)

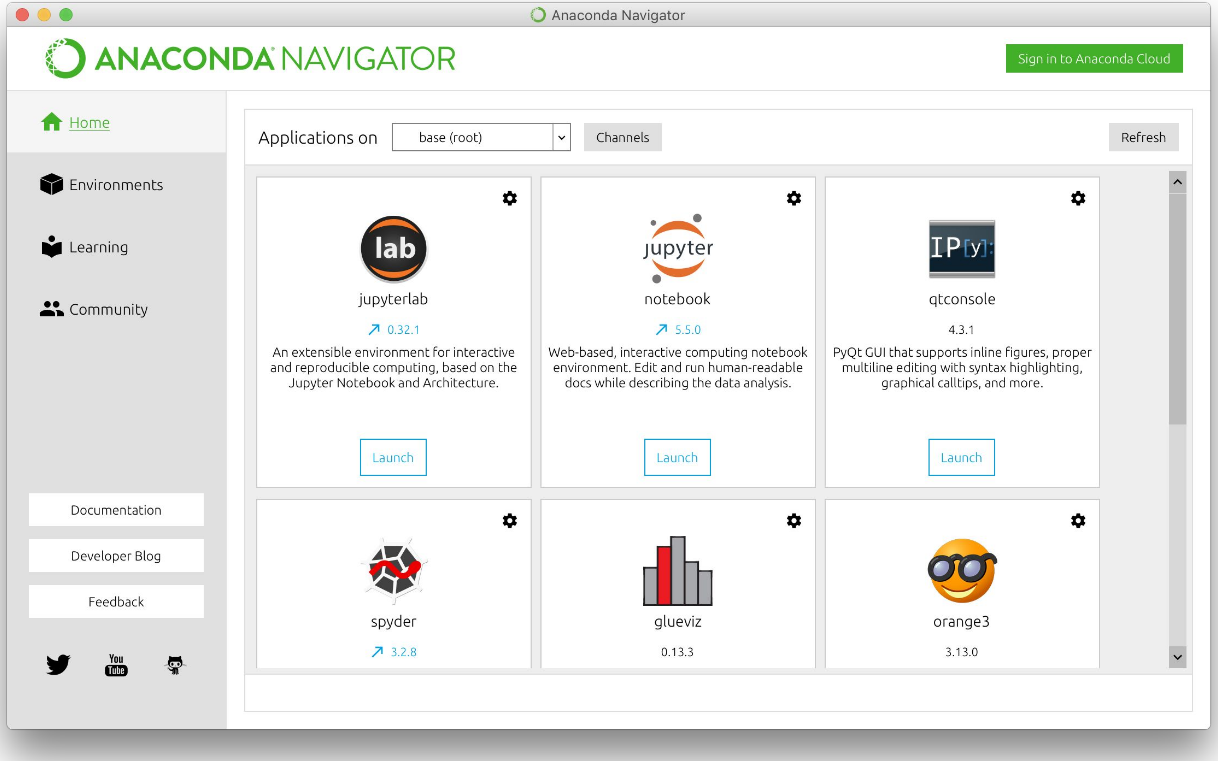
64-Bit Command Line Installer (424 MB)

Python 2.7 version

64-Bit Graphical Installer (637 MB)

64-Bit Command Line Installer (409 MB)

Anaconda Navigator



Anaconda Navigator

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Feedback

Applications on base (root) Channels Refresh

jupyterlab

↗ 0.32.1

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch

notebook

↗ 5.5.0

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch

qtconsole

4.3.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch

spyder

↗ 3.2.8

glueviz

0.13.3

orange3

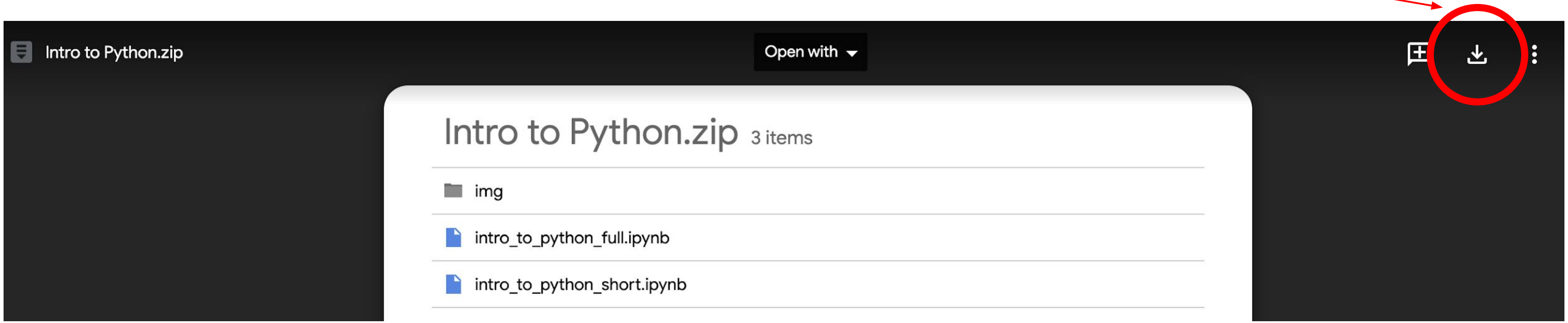
3.13.0

Download some pre-written Python code

Use this link to download a .zip folder containing Python code and exercises.

Unzip the folder to a convenient location on your computer e.g. in the ‘Documents’ folder.

http://bit.ly/ds37-session-01



Intro to Python.zip 3 items

- img
- intro_to_python_full.ipynb
- intro_to_python_short.ipynb

Data Science Part Time

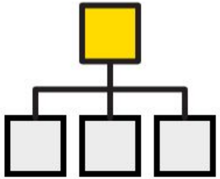
The data science workflow

The Data Science Workflow

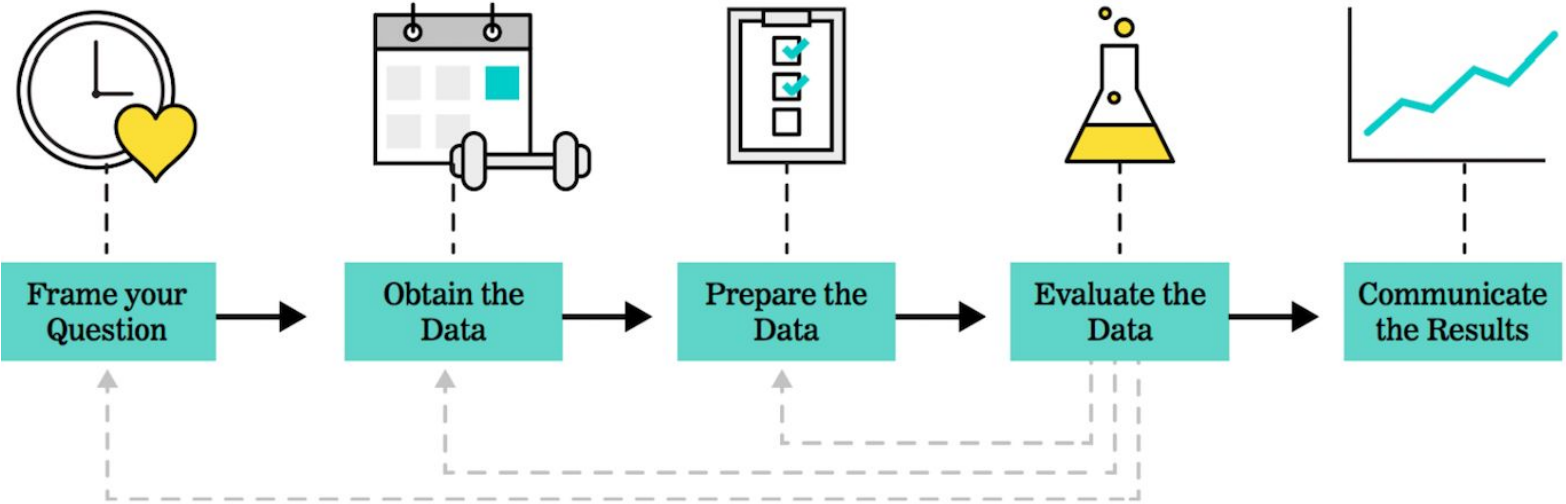
Why use a workflow?

A workflow helps you produce *reliable* and *reproducible* results.

- **Reliable**: Accurate findings
- **Reproducible**: Others can follow your steps and achieve the same results.



The Data Science Workflow



Frame your Question → Obtain the Data → Prepare the Data → Evaluate the Data → Communicate the Results

Data Science Part Time

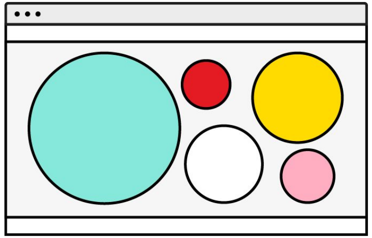
Framing the problem

What's a Good Data Science Question?

Specific

Measurable

Achievable



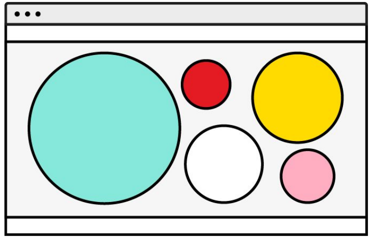
What’s a Good Data Science Question?

Instead of asking:

How good is my local hospital?

You could ask:

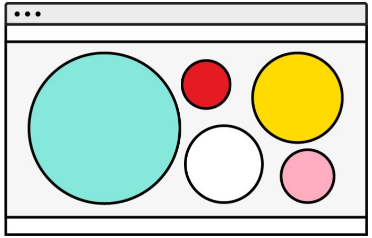
Has the average waiting time for non-emergency operations at my local hospital increased significantly over the past year?



Common Questions Asked in Data Science

From a business perspective:

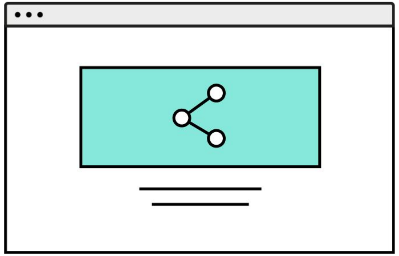
1. What is the likelihood that a customer will buy this product?
2. How much demand will there be for my service tomorrow?
3. What groups of products are customers purchasing together?
4. Can we automate this simple yes/no decision?



Common Questions Asked in Data Science

From a data science perspective:

1. Does X predict Y?
2. Are there any distinct groups in our data?
3. What are the key components of our data?
4. Is one of our observations “weird”?



Partner Exercise:

Asking Good Questions

5 minutes

In pairs, rephrase these questions so they're better data science questions:

1. **How well is Google doing?**

2. **How much will my house sell for?**

3. **Who voted for Donald Trump?**

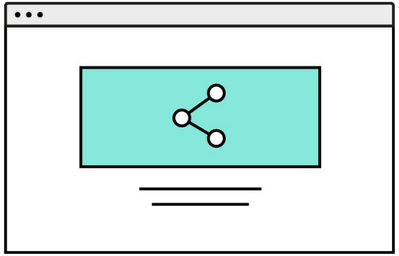
Data Science Part Time

Obtaining data

Obtaining Good Data is Important

Keep in mind:

1. Is the data from an official, trustworthy source?
2. How did the source organisation collect the data?
3. How easy is it to obtain? Is it free to access?
4. Do you have permission to use it?
5. Do we know what the data means?
6. What are the limitations of the data?



Solo Exercise:

Let’s Race to Find Good Data!

Your client wants to understand the demographic factors driving the 2016 US Presidential election results.

As quickly as possible, find a dataset online that can help with this problem **and** meets our ‘good data checklist’ criteria.

Make some noise as soon as you’ve found one!

Partner Exercise:

Identifying Limitations

Open election_results_2016.csv

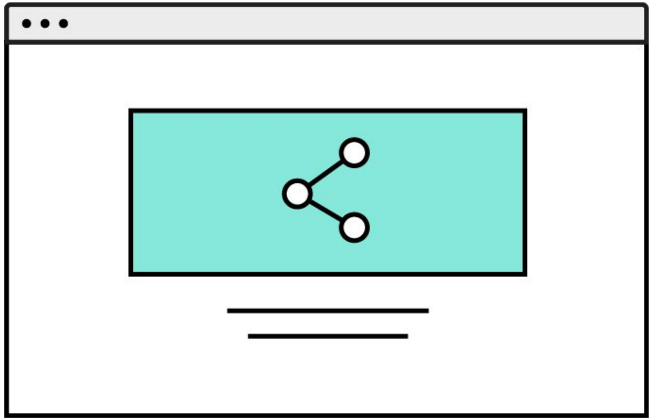
These are the 2016 US Presidential election results, taken from https://www.kaggle.com/benhamner/2016-us-election.
Discuss with your partner:

- What does each row represent? (Hint: look for a data dictionary on the Kaggle site)
- What do the columns mean?
- What are the limitations of the dataset?
- What information is missing?

Data Formats

In this course, we'll be using several sources and formats of data, including:

1. Spreadsheets
2. APIs
3. Web scraping



Data Science Part Time

Analysis

Types of Analysis

There are many types of analysis you can perform:

- Exploratory data analysis
- Calculating descriptive statistics
- Finding correlations
- Modelling
- Machine learning

Group Exercise:

Analysing U.S. Election Data

On your tables, view the US election dataset in Excel.

You want to understand which demographic factors are most strongly associated with a high vote-share for Hillary Clinton.

In your groups, discuss:

- How you might analyse the data to answer this question
- Any pitfalls or common mistakes you would watch out for

Data Science Part Time

Interpretation and communication

Group Exercise:

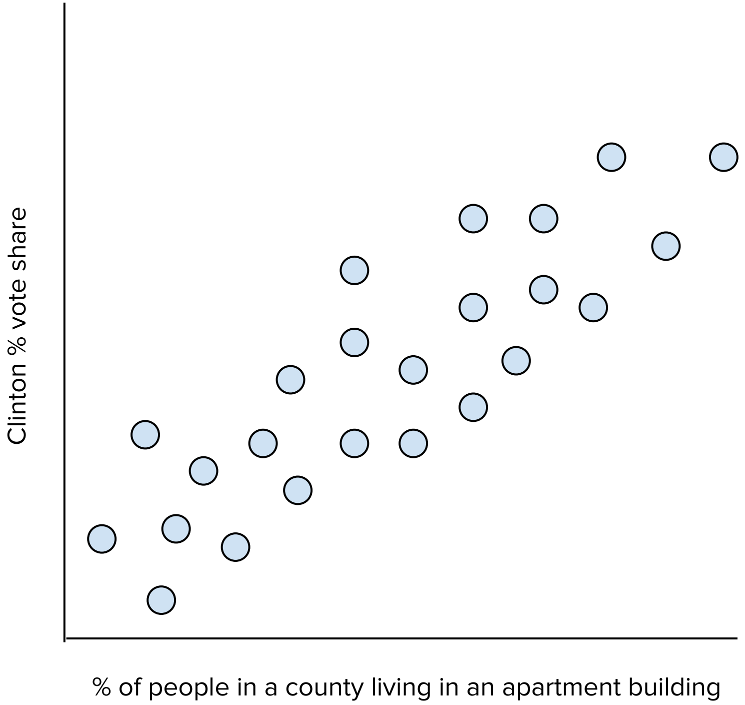
Responsible Interpretation

15 minutes

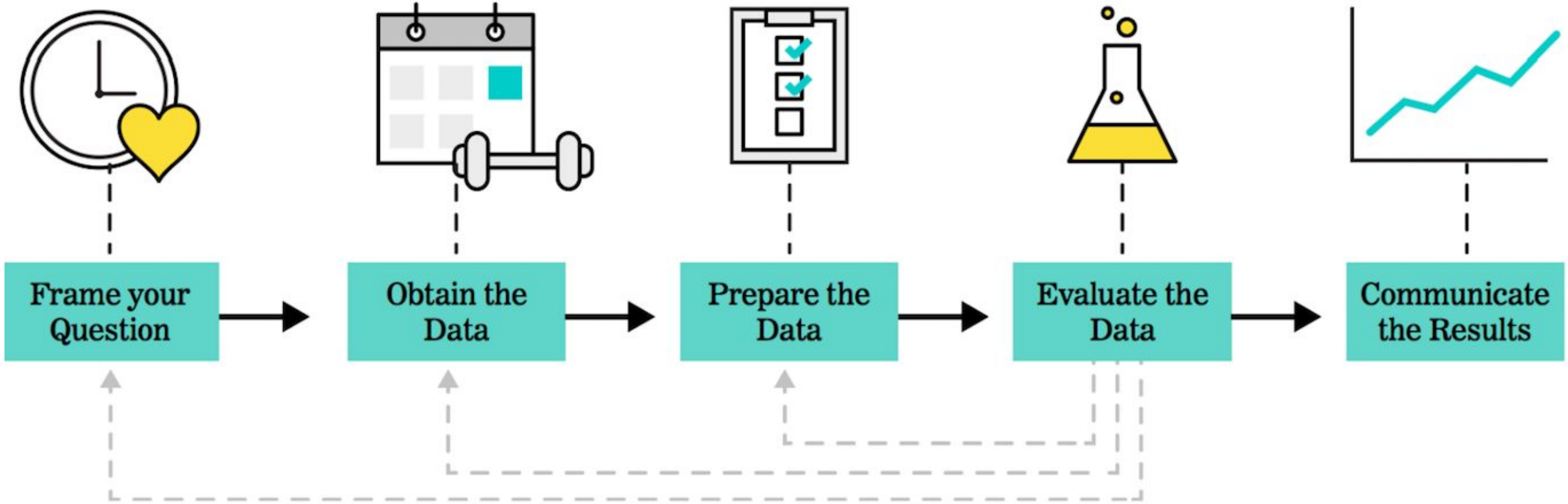
Imagine the results of your analysis show that the **strongest** correlation in your dataset is between the **proportion of people in a county that live in apartment buildings** and **Clinton's vote share**.

Would you interpret this result to mean:

- Apartment buildings are the strongest cause of people voting Democrat
- People who live in apartments were Clinton's biggest supporters
- Something else? Why?



Review: The Data Science Workflow



Frame your Question

Obtain the Data

Prepare the Data

Evaluate the Data

Communicate the Results

Python Explained

Python is a high-level, open source, object-oriented software programming language often used for scripting, data analysis, and rapid software development

python
powered

```
print("Hello, world!")
```

Python Explained

Python is a high-level, open source, object-oriented software programming language often used for scripting, data analysis, and rapid software development

python
powered

```
print("Hello, world!")
```

Group Exercise:

What is programming?

10 minutes

In small groups, discuss:

- What is programming?
- What's an algorithm?
- What are some examples of algorithms you use in everyday life (non-computer related)?
- What makes a good algorithm?

Keep your definitions short and simple

Python Explained

Python is a high-level, open source, object-oriented software programming language often used for scripting, data analysis, and rapid software development

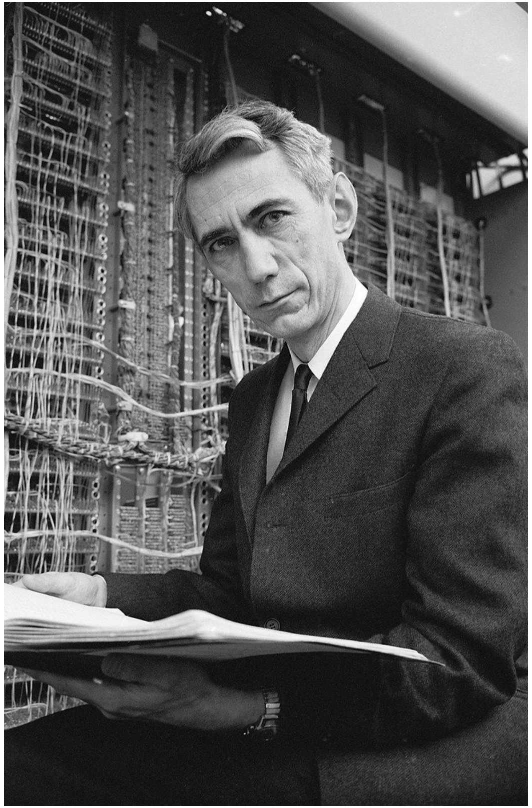
```
print("Hello, world!")
```

Discussion:

The simplest programming language

What’s the simplest, most fundamental language for giving computers instructions?

Binary



- A way of representing information as a string of 1s and 0s
- The most fundamental way of giving instructions to a computer
- All computer processors (the ‘brains’ of a computer) only understand binary

Group Exercise:

Binary

10 minutes

In small groups, discuss:

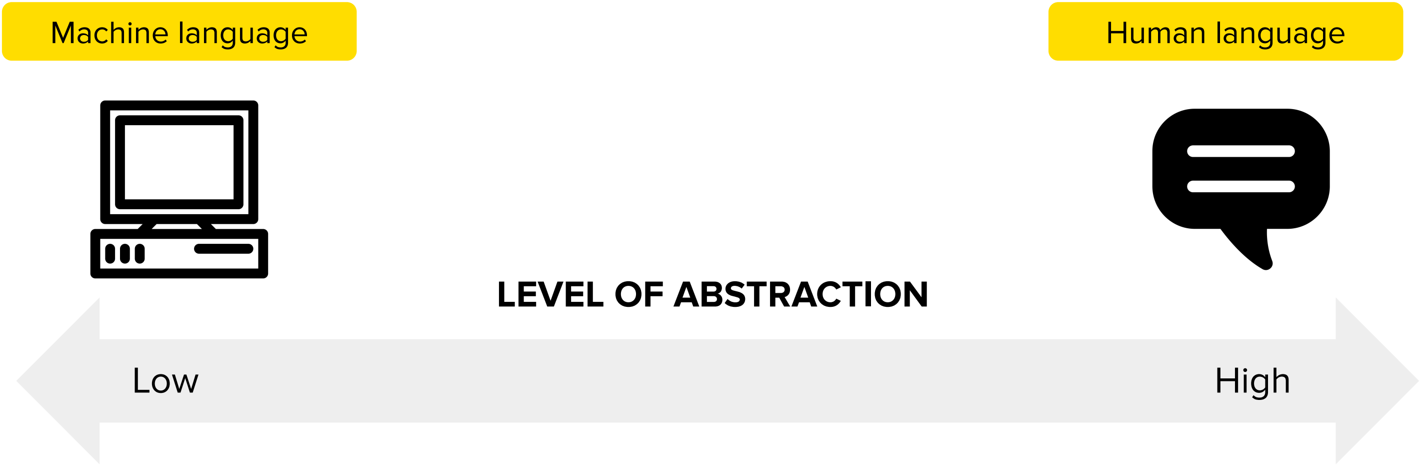
- What are some of the benefits and drawbacks of programming in binary? Would you want to do it?
- If all computers fundamentally only understand binary, how is it that we can write and run code in languages like Python, JavaScript, C++ etc?

Compilers

“It's much easier for most people to write an English statement than it is to use symbols, so I decided data processors ought to be able to write their programs in English, and the computers would translate them into machine code.”

Rear Admiral Grace Hopper (1906-1992)

High-level vs. low-level programming



Machine language

Human language

LEVEL OF ABSTRACTION

Low

High

```
jumpval dd 0 ;
state db 0 ;
```

```
print("hello")
```

Python Explained

Python is a high-level, open source, object-oriented software programming language often used for scripting, data analysis, and rapid software development

```
print("Hello, world!")
```

Open Source

Open source code is free for anyone to use, modify, and distribute.

Python is an open source language, so everyone is free to look ‘under the hood’ at its source code.

People can also add to Python’s functionality by writing their own ‘add on’ source code.

This makes Python a powerful, robust and flexible programming language.

Group Exercise:

Why open source?

10 minutes

Let’s discuss:

The benefits and drawbacks of why a programmer would want to make something they’ve built open source.

The benefits and drawbacks of using an open source technology, tool or app from a user’s perspective.

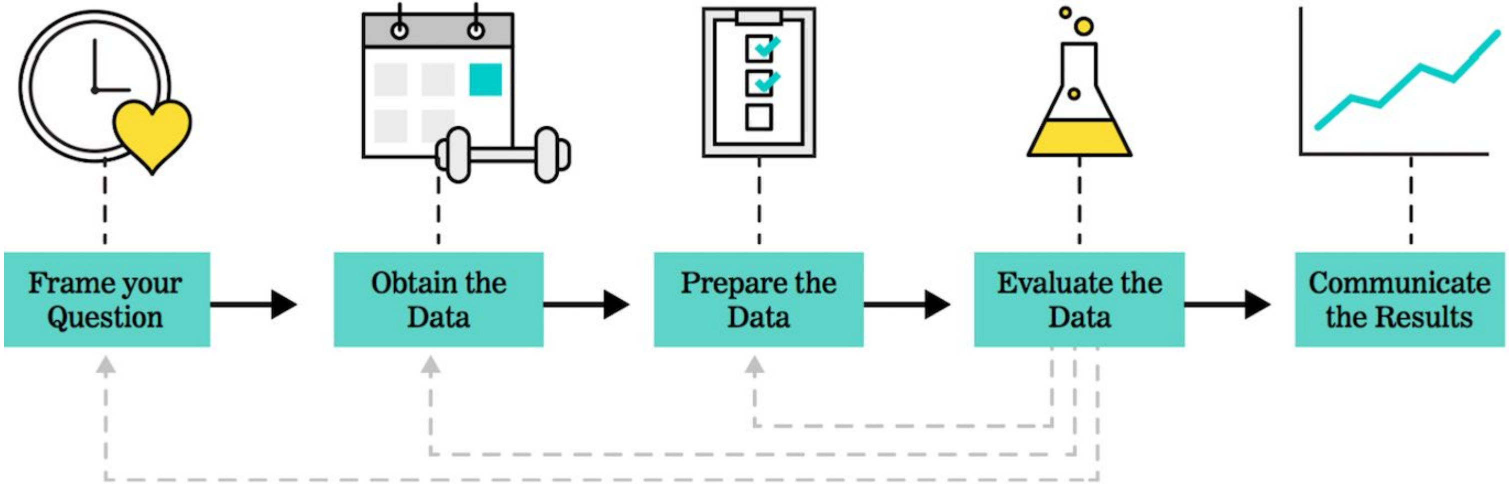
Python Explained

Python is a high-level, open source, object-oriented software programming language often used for scripting, data analysis, and rapid software development

python
powered

```
print("Hello, world!")
```

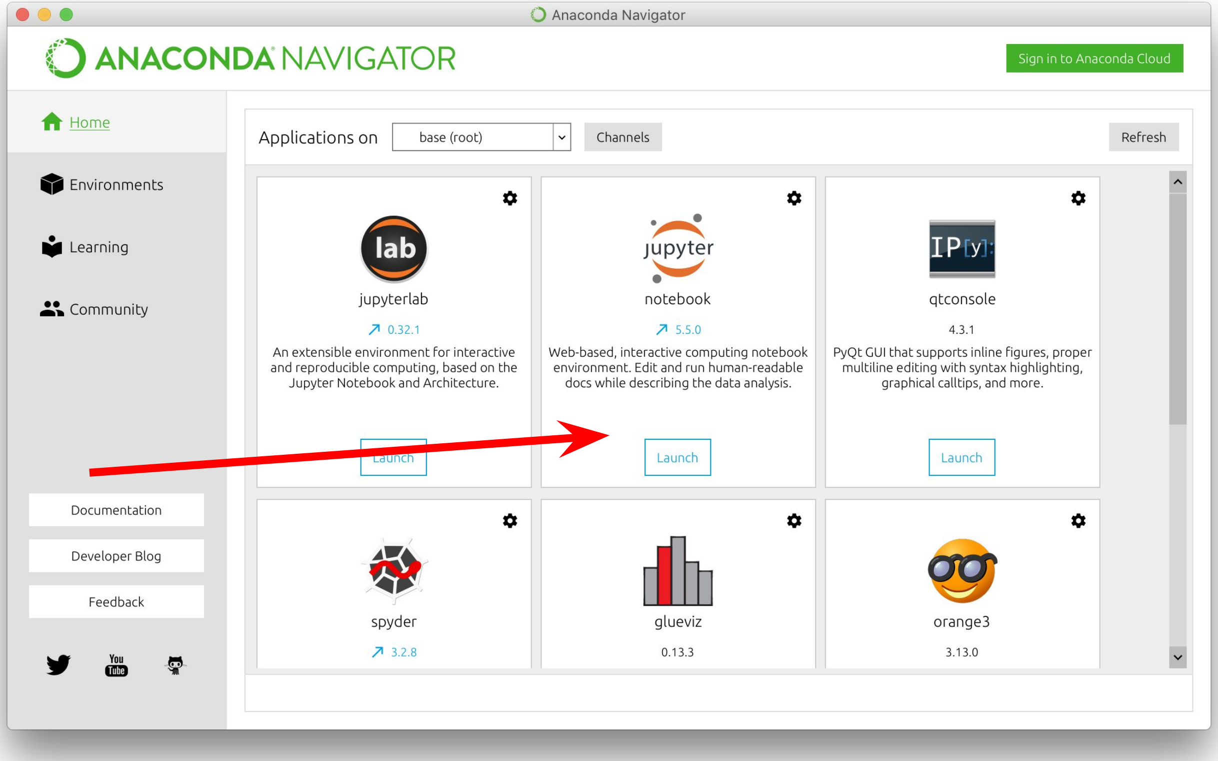
Python is used at every stage of the data science workflow



Data Science Part Time

Introducing Jupyter

Anaconda Navigator



Anaconda Navigator
ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Learning
Community
Documentation
Developer Blog
Feedback
Applications on
base (root)
Channels
Refresh
jupyterlab
0.32.1
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
notebook
5.5.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
qtconsole
4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
spyder
3.2.8
glueviz
0.13.3
orange3
3.13.0

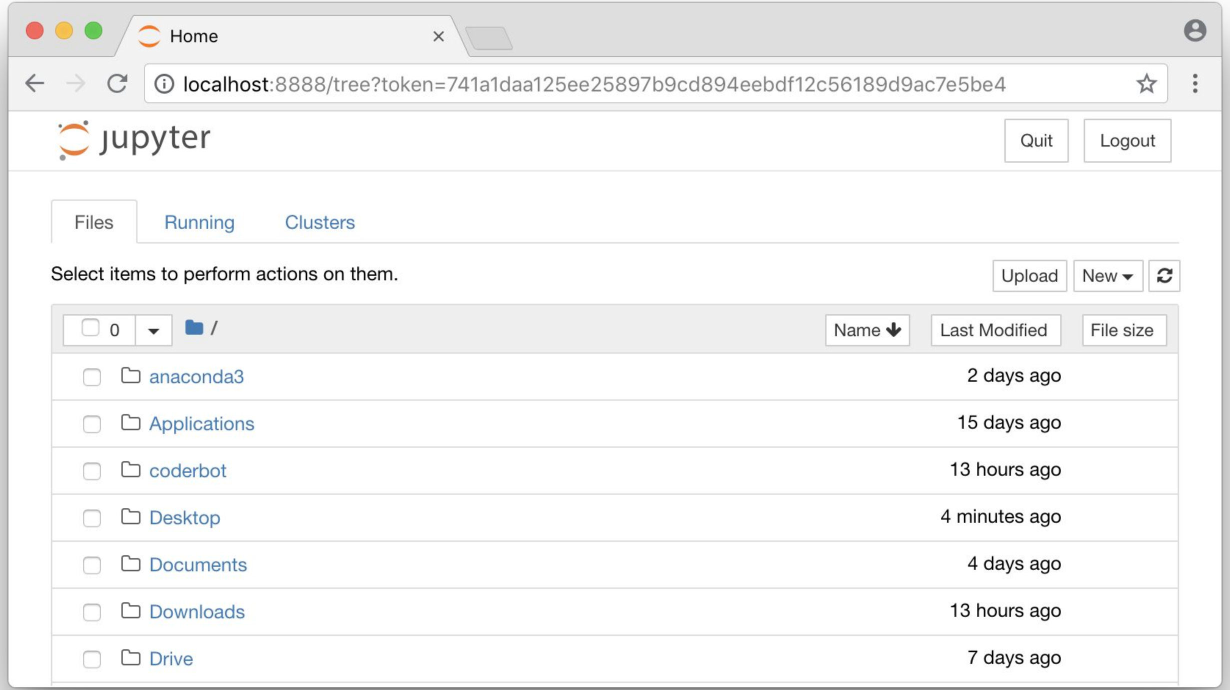
What’s Anaconda?

Anaconda is a **distribution** of Python.

This means it’s a piece of software that includes Python together with some useful **tools** that make programming in Python easier.

Jupyter Notebook is one of those tools.

Navigate to the location of the newly downloaded folder



Select items to perform actions on them.

Name	Last Modified
anaconda3	2 days ago
Applications	15 days ago
coderbot	13 hours ago
Desktop	4 minutes ago
Documents	4 days ago
Downloads	13 hours ago
Drive	7 days ago

What’s Jupyter Notebook?

Jupyter Notebook (or ‘Jupyter’) is an **environment** for writing and running Python code.

It’s widely used in industry and academia, and has lots of handy features that make it easier to use than many other programming **environments**.

Let’s explore them!

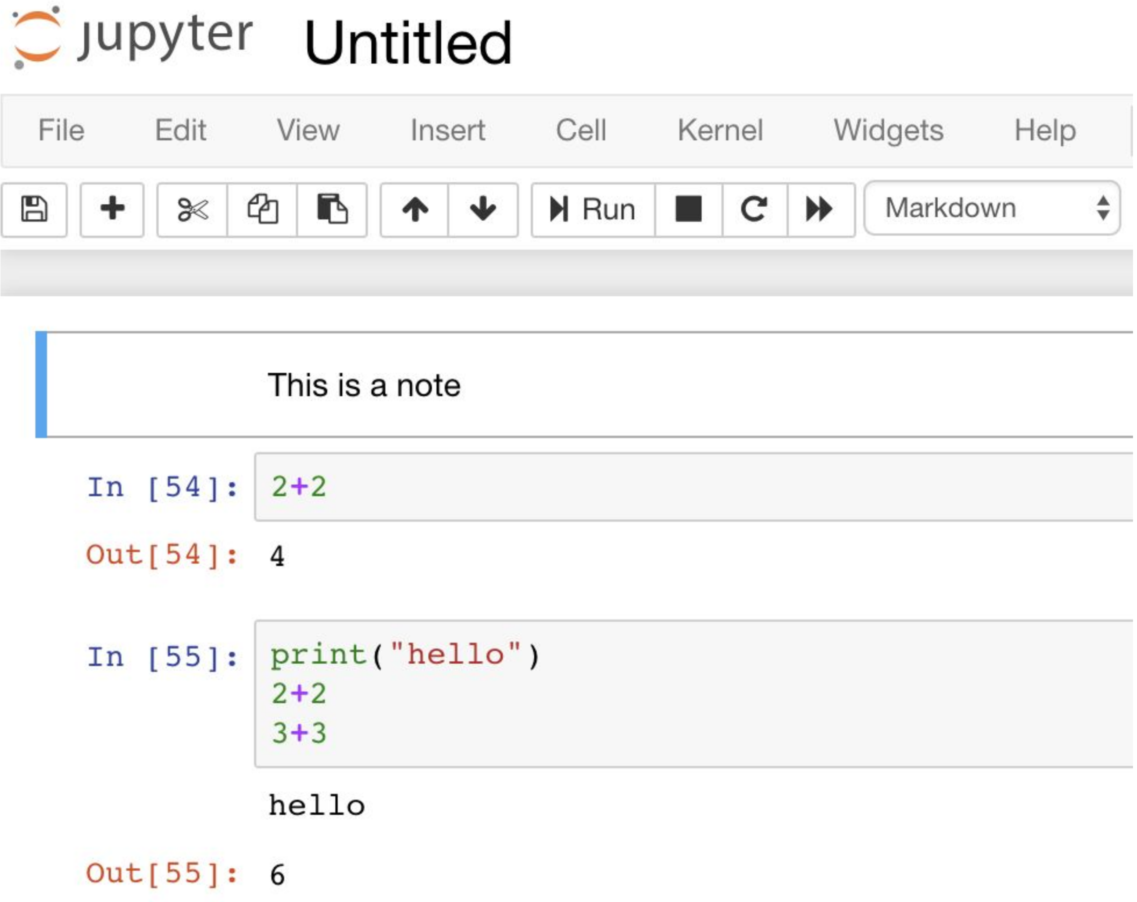
What’s Jupyter Notebook?

Some important points about writing code in Jupyter:

1. Notebooks will **autosave** so don’t panic if you accidentally close a tab
2. You don’t need to be connected to the internet to use Jupyter
3. One cell must **finish** running before another cell can run
4. A code won’t be executed until you run the cell

Jupyter Notebook

- **Cells**
  - **Markdown** for notes
  - **Code** for Python
- **Execution**
  - Shift + return
- **Output**
  - Print (all)
  - Return values (last)



This is a note

```
In [54]: 2+2

Out[54]: 4

In [55]: print("hello")
         2+2
         3+3

         hello

Out[55]: 6
```

Jupyter Shortcuts

Shift+Enter Run cell

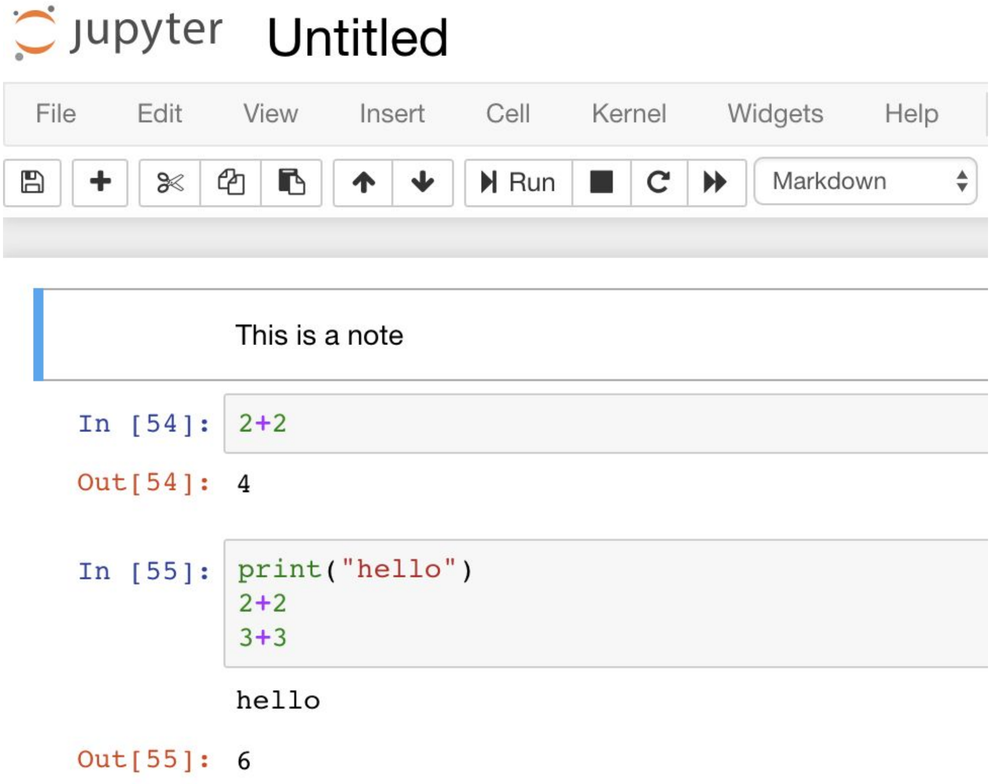
Esc+B Insert cell below

Esc+A Insert cell above

Esc+Y Convert to code cell

Esc+M Convert to markdown cell

Esc+H View all shortcuts

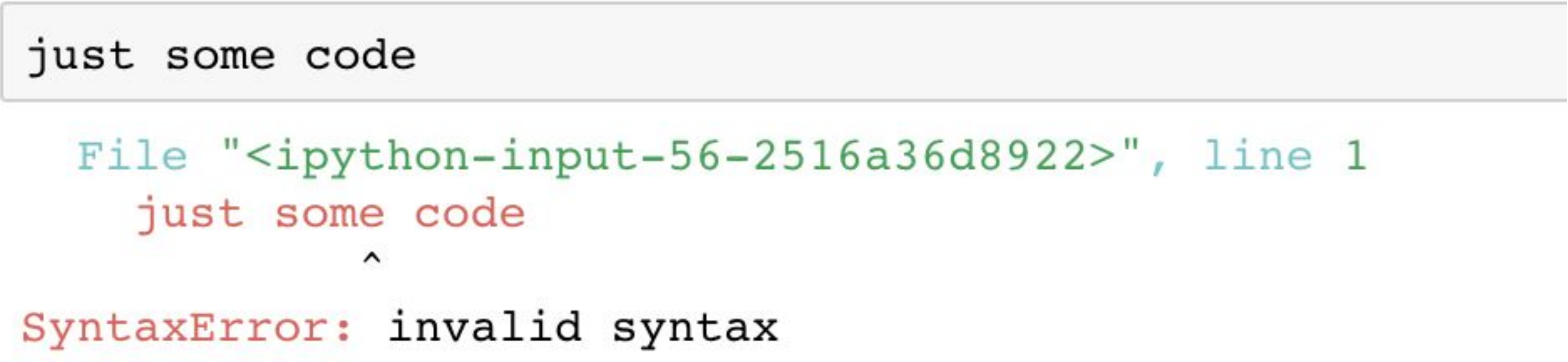


This is a note

```
In [54]: 2+2

Out[54]: 4

In [55]: print("hello")
         2+2
         3+3

         hello

Out[55]: 6
```

Jupyter Notebook errors

Mistakes happen! Here’s what they look like:



```
just some code
```

```
  File "<ipython-input-56-2516a36d8922>", line 1
    just some code
            ^
SyntaxError: invalid syntax
```

1. Try to understand what went wrong
2. Attempt to fix the problem
3. Execute the cell again

Solo Exercise:

Try out Jupyter notebook

Open a new Python 3 Jupyter notebook.

Practise using Jupyter by doing the following:

1. Insert a cell, convert it to a **markdown** cell, insert some text and execute the cell
2. Edit the **markdown** cell so it contains a large heading (hint: use ‘#’ to make a heading)
3. Insert a **code** cell below the **markdown** cell, and execute the calculation ‘2+2’
4. Insert a new cell and delete it immediately

Intro to Python

Python Programming Fundamentals

Computers Out:

Let’s try Python!

1. For Python, let’s write and run code directly in Jupyter!
2. Open the ‘ds37-01-01.ipynb’ file in Jupyter Notebook

Computers Out:

Hello, World!

2 minutes

Let’s get started with our first Python command. It’s useful to think of every Python command in terms of inputs and outputs. Here:

The **input** is the text ‘hello world’

The **output** is the print-out of the text underneath the cell

The **command** is called ‘print’

Discussion:

Types of data

3 minutes

All of these are different **types** of information:

2
2.0
“hello123”
“a”
[3, 4, 5, 6, 7]

Which of these can be…
Divided by 3?
Converted to uppercase?

Different types of data need to be handled in different ways. Python does this by treating different kinds of data as different types.

Variables

A **variable** is a way of assigning a name to a piece of data.

Variables make it easier and more efficient to store data and perform calculations with it.

Variables can be numbers, strings, lists, etc…

The process of creating a variable is called **declaring** a variable.

Strings

A string is a collection of alphanumeric characters, contained inside quotation marks.

Python treats strings like text.

Different methods can be applied to strings, like converting to lowercase or uppercase.

```
‘hello world’
‘123’
“apple”
```

Lists

A **list** is a sequence of **elements**. We declare a **list** using square brackets and separating each element with a comma.

Each element can be a number, string, or even another list.

```
[1.0, 3, ‘hello’, [1, 2, 3]]
```

Lists can hold multiple types of data.

Elements of a list can be retrieved, or **indexed**, using their position.

Python uses **zero indexing**, so the first element in a list is at position or **index** 0.

Dictionaries

A **dictionary** consists of **key value pairs** and is an alternative way of storing data.

```
my_dictionary = {name: ‘Maryam’,
fav_food: ‘pizza’,
fav_drink: ‘orange juice’}
```

Instead of retrieving elements from a dictionary using their **position or index** (as with lists) we retrieve elements using their **keys.**

```
my_dictionary[‘fav_food’]
```

Intro to Python

Let’s Review

Coming up next session...

- Using Git for version control
- Using bash to perform simple file operations

Homework

Work through the exercises in ds37-01-02.ipynb

Solutions are available in ds37-01-02-solutions.ipynb if you get stuck.

Read through this article and be ready to discuss next session: https://www.bloomberg.com/news/articles/2019-09-09/jpmorgan-creates-volfefe-index-to-track-trump-tweet-impact

GA