Data Science

12/9/2023

Intro to Data Science

- Intro to Data Science
- Where does data come from?
- How to use data?
- What is Data Science?
- How to use data?
- Example data science applications

TOPICS WE'LL COVER

- Overview of Data Science
- Data Science Applications
- Essential Skills for Data Science

GOALS FOR THIS DATA CAMP

- Understand the concept and significance of Data Science.
- Explore the Data Science lifecycle and project stages.
- Learn about practical applications of Data Science.
- Understand essential skills for Data Scientists.
- Explore ethical considerations in Data Science.
- Gain insights into emerging trends in the field.

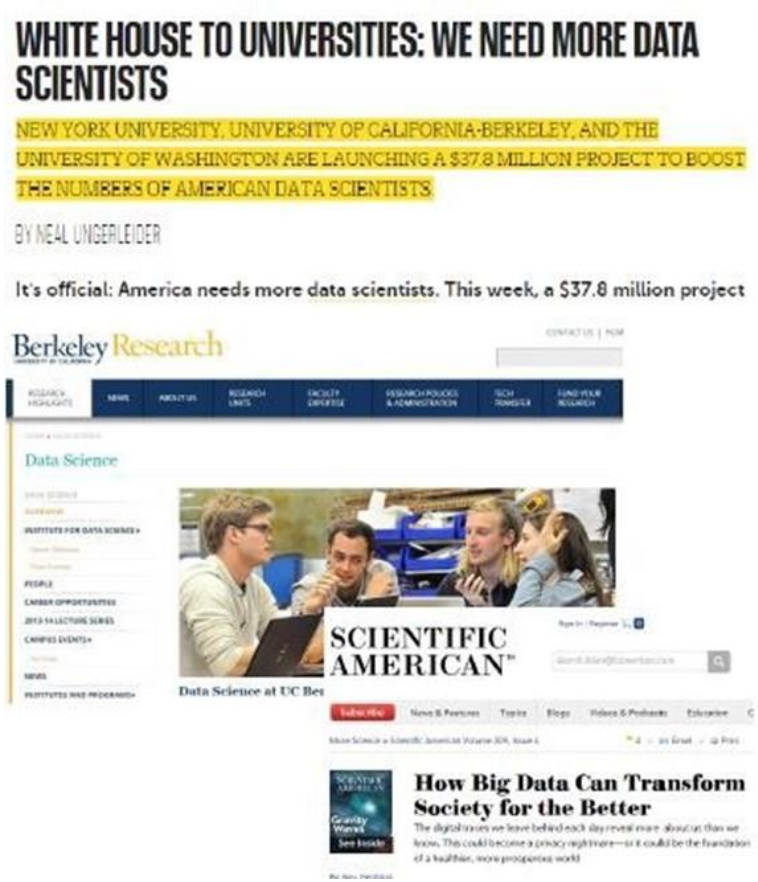
What is data science?

Intro to Data Science

Data Scientists are in high demand

Intro to Data Science

Data Scientists are in high demand

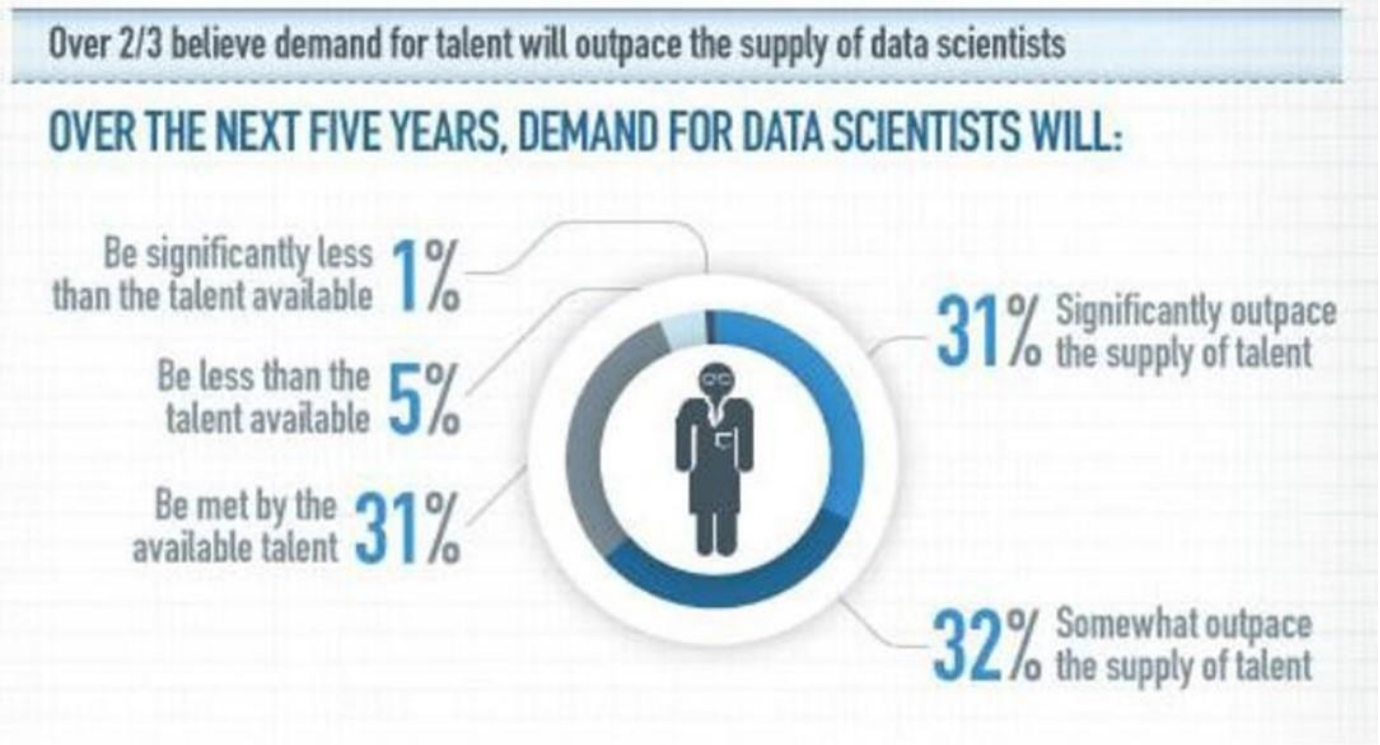


Intro to Data Science

Pays Well

Intro to Data Science

Demand will outpace supply

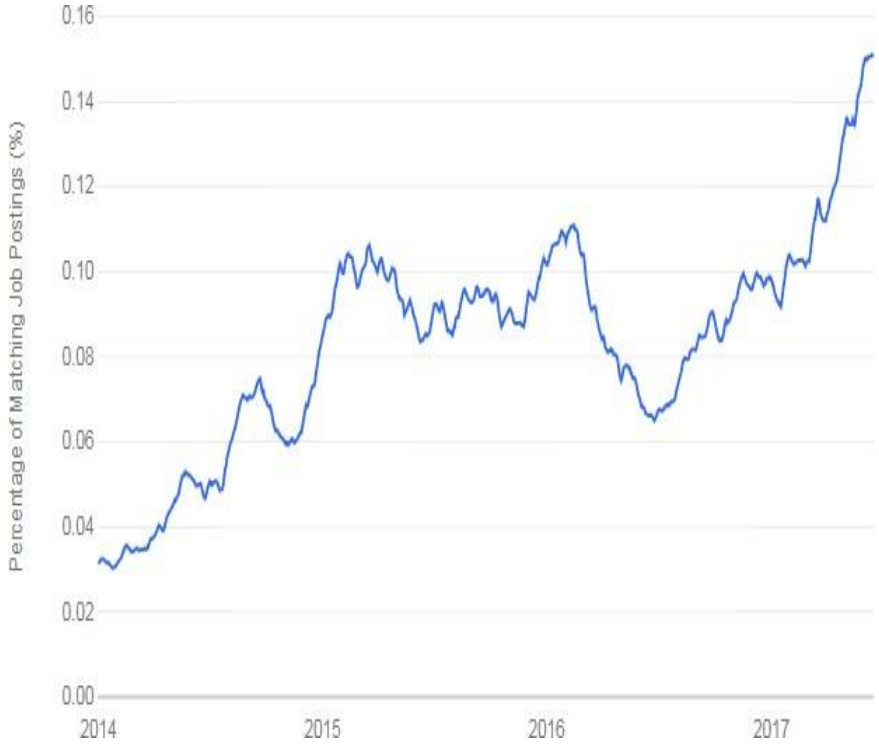


Intro to Data Science

Data Scientist Job Trend in last 3 years

Job postings

0.151%



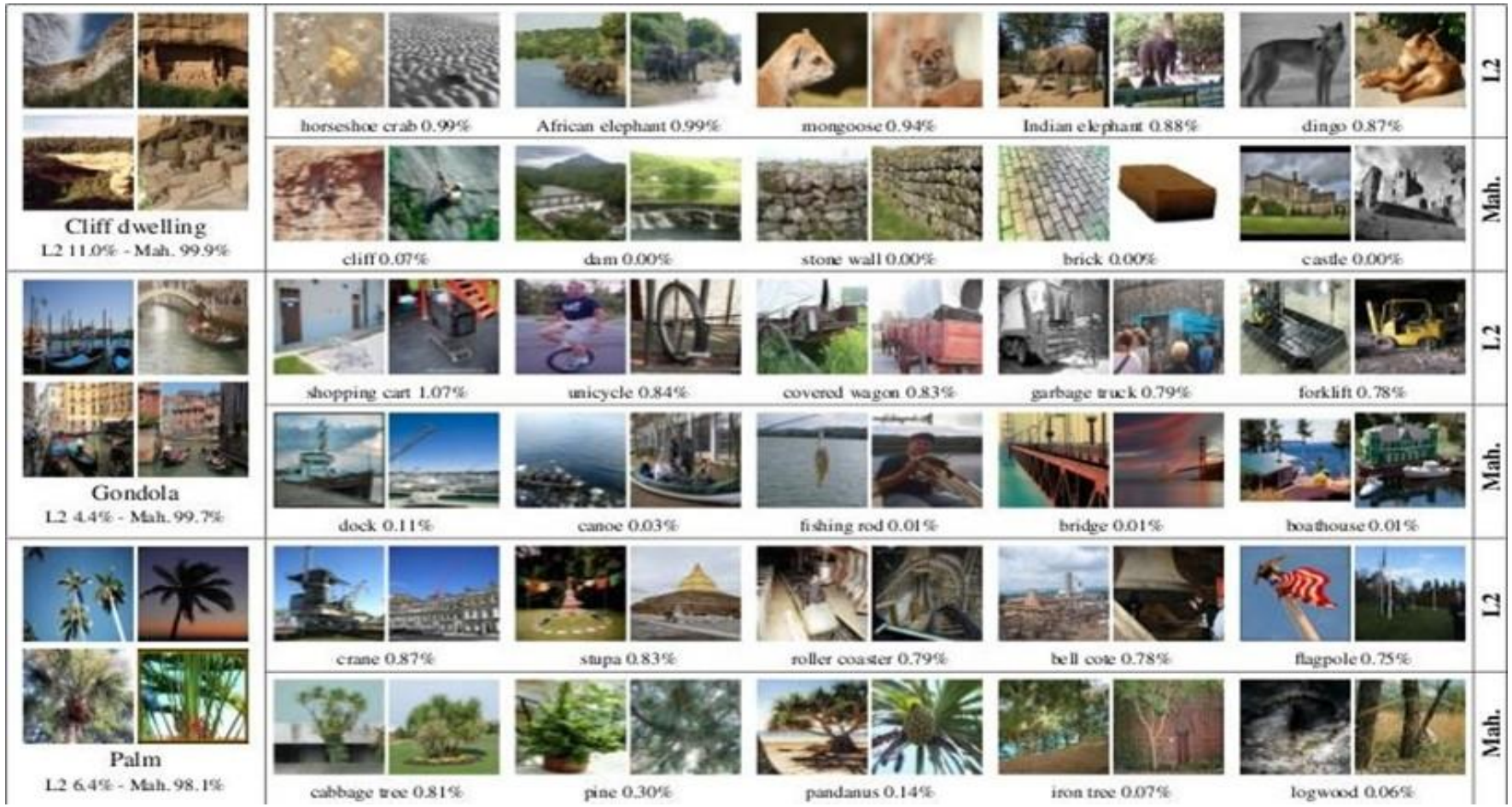
Jobseeker interest

0.074%

Source: indeed.com

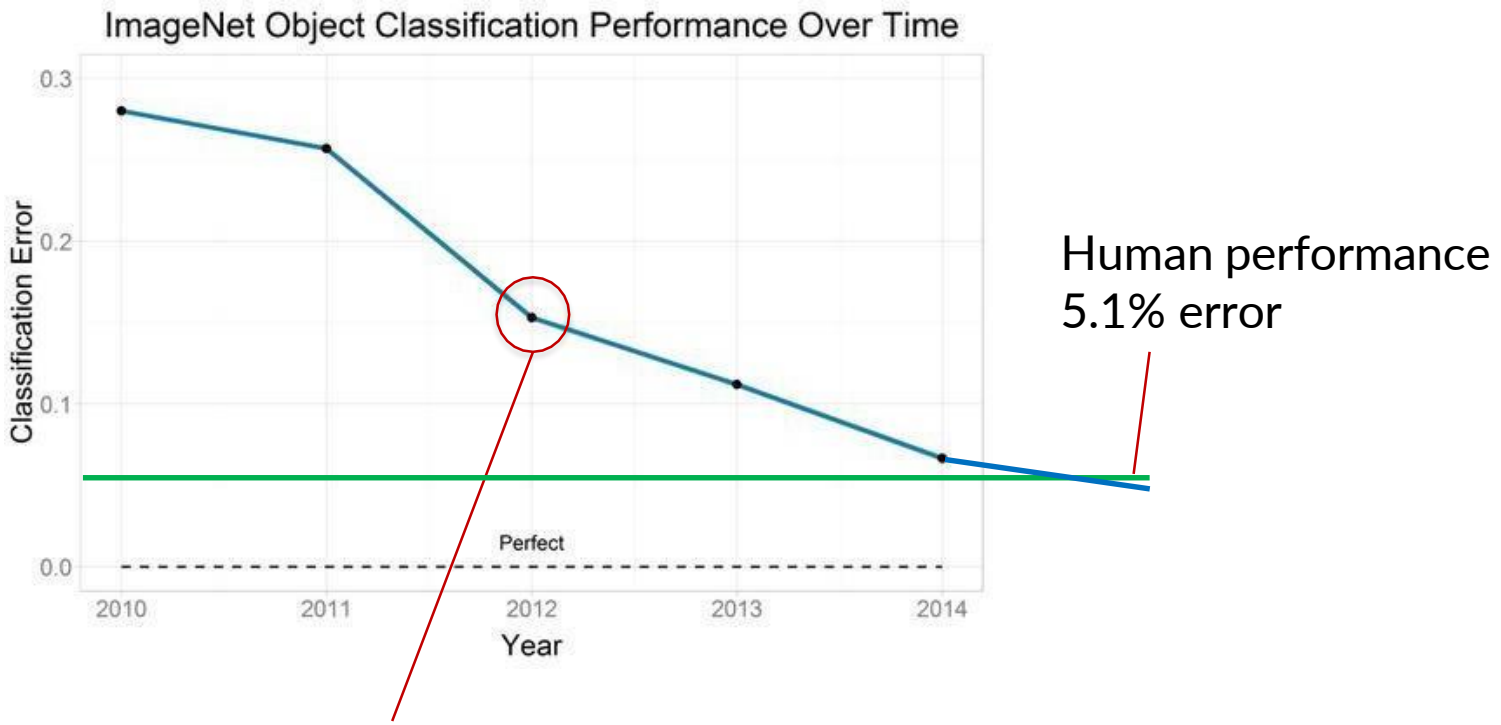
The unreasonable effectiveness of Deep Learning (CNNs)

2012 ImageNet challenge:
Classify 1 million images into 1000 classes.



The unreasonable effectiveness of Deep Learning (CNNs)

Performance of deep learning systems over time:

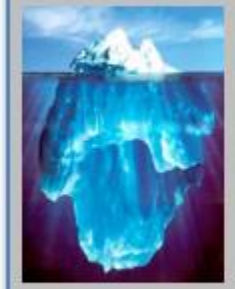


Krizhevsky, Sutskever, and Hinton, NIPS 2012

Where does data come from?

“Big Data” Sources

It’s All Happening On-line

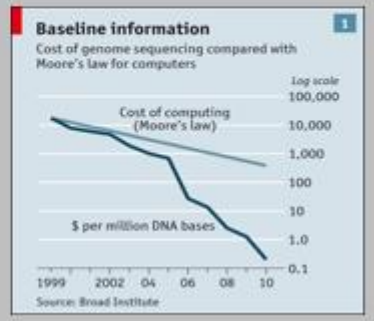


Every:
Click
Ad impression
Billing event
Fast Forward, pause,…
Server request
Transaction
Network message
Fault
…

User Generated (Web & Mobile)

Internet of Things / M2M

Health/Scientific Computing



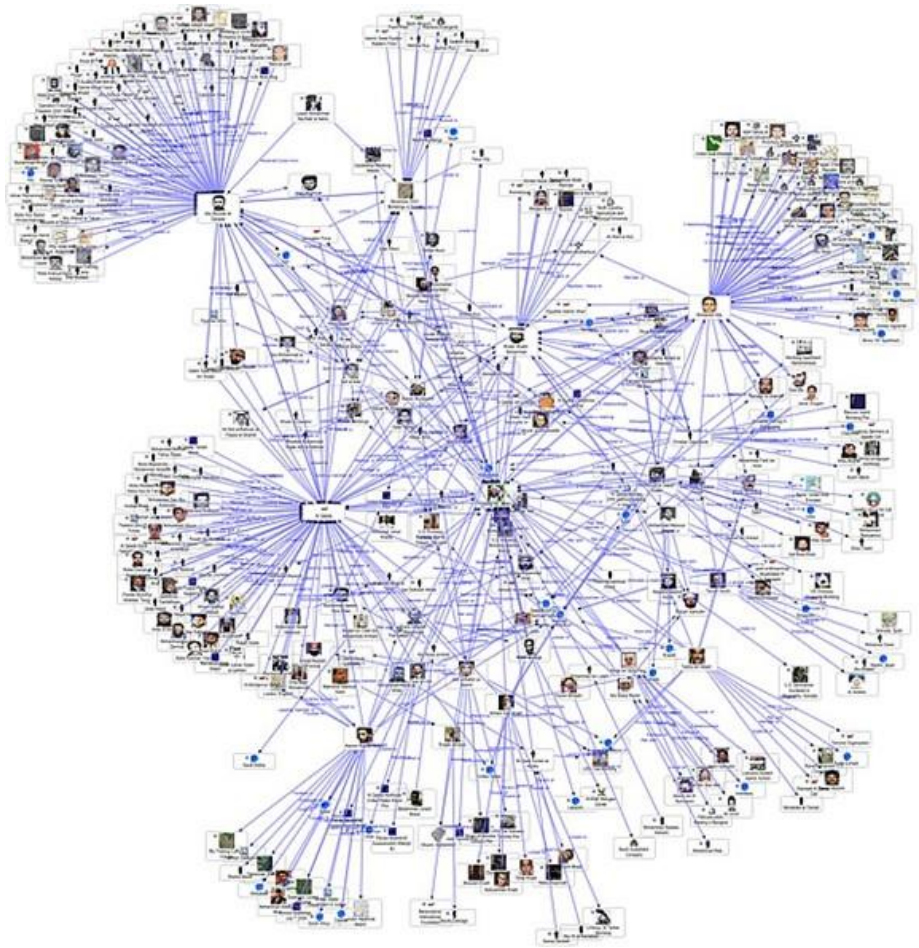
Where does data come from?

Graph Data

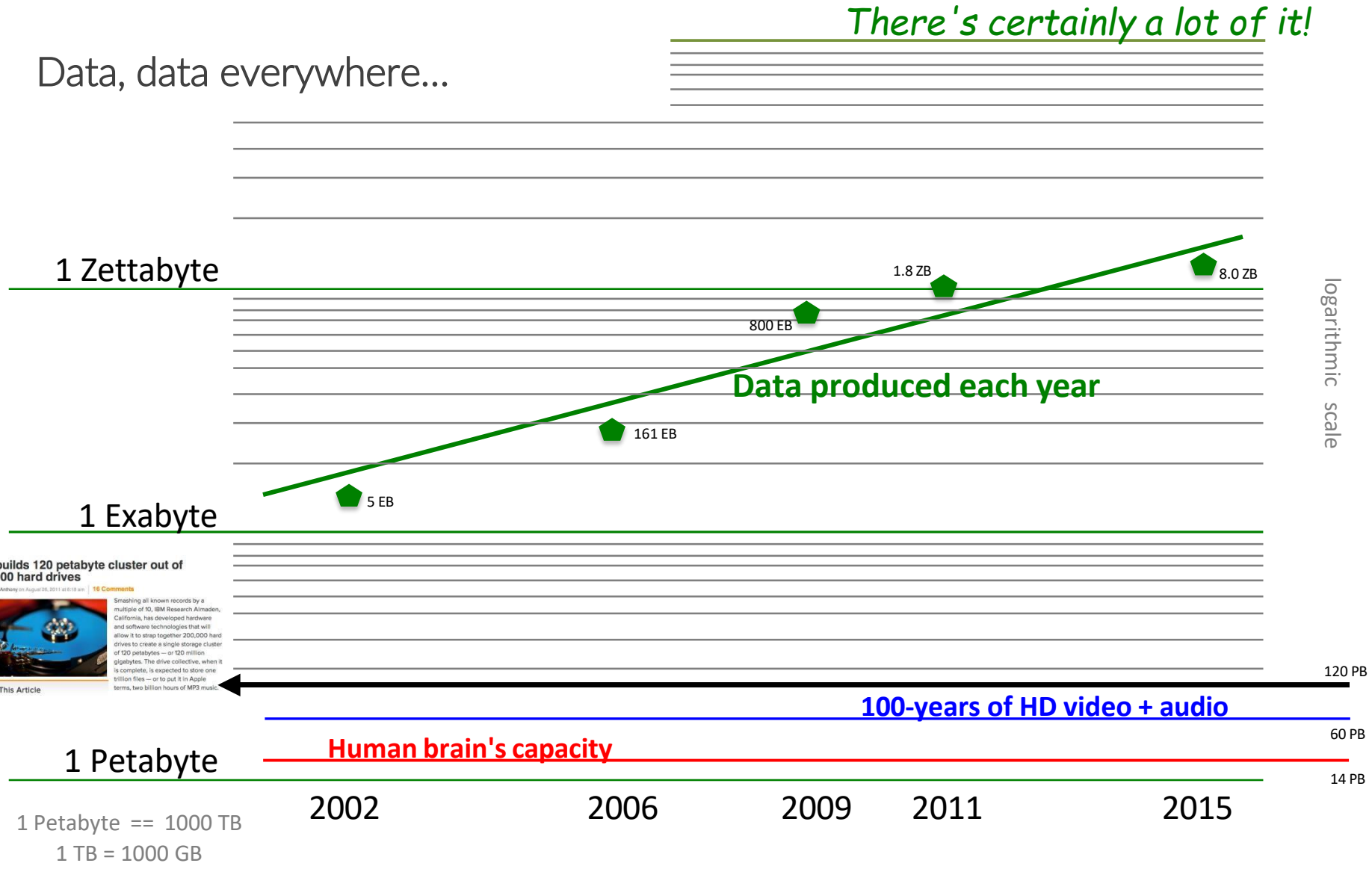
Lots of interesting data has
a graph structure:

- Social networks
- Communication networks
- Computer Networks
- Road networks
- Citations
- Collaborations/Relationships
- …

Some of these graphs can get quite
large (e.g., Facebook* user graph)

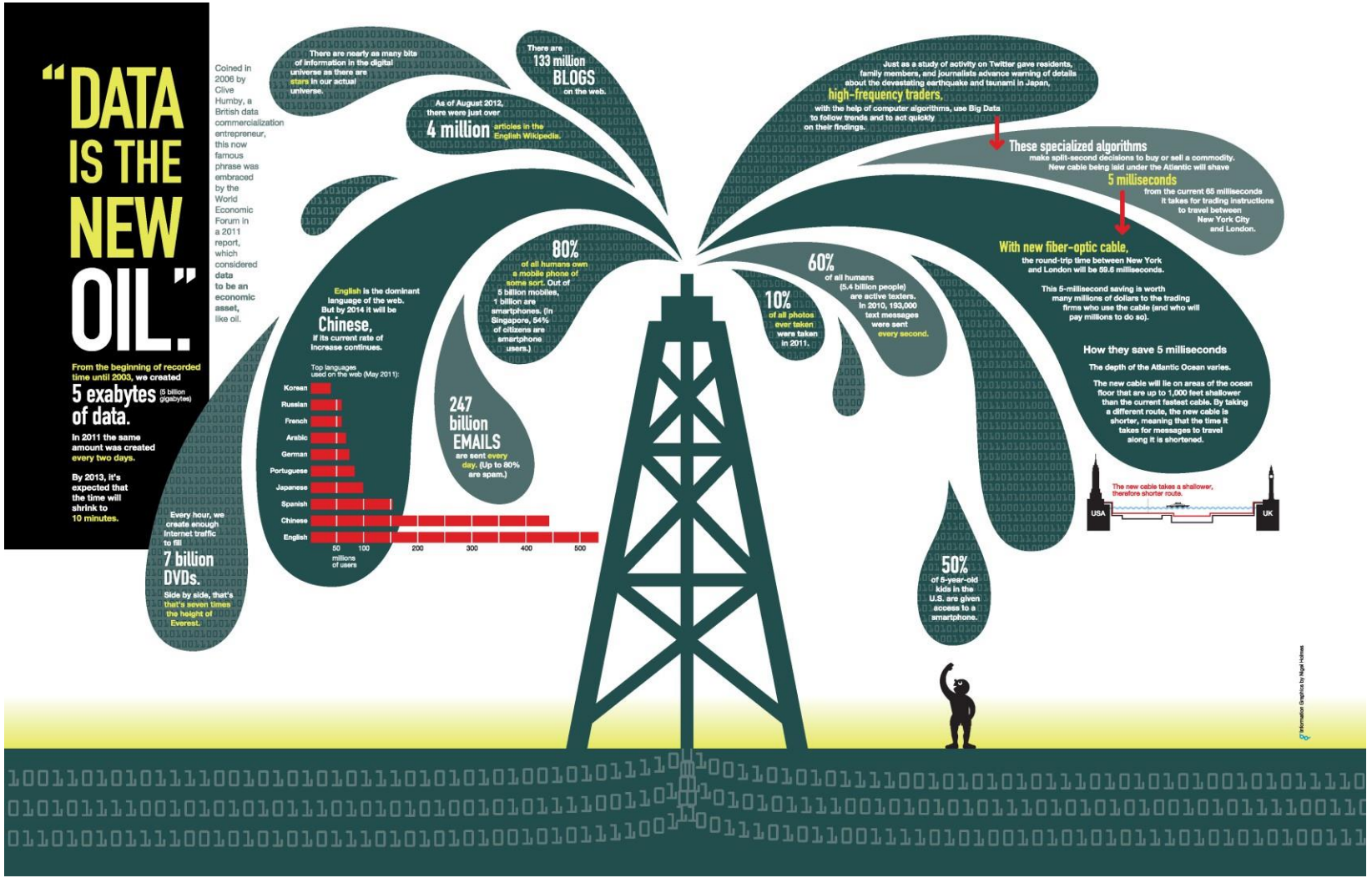


Where does data come from?



Where does data come from?

"Data is the New Oil" – World Economic Forum 2011



What is Data Science?

Data Science – A Definition

Data Science is the science which uses computer science, statistics and machine learning, visualization and human-computer interactions to collect, clean, integrate, analyze, visualize, interact with data to create data products.

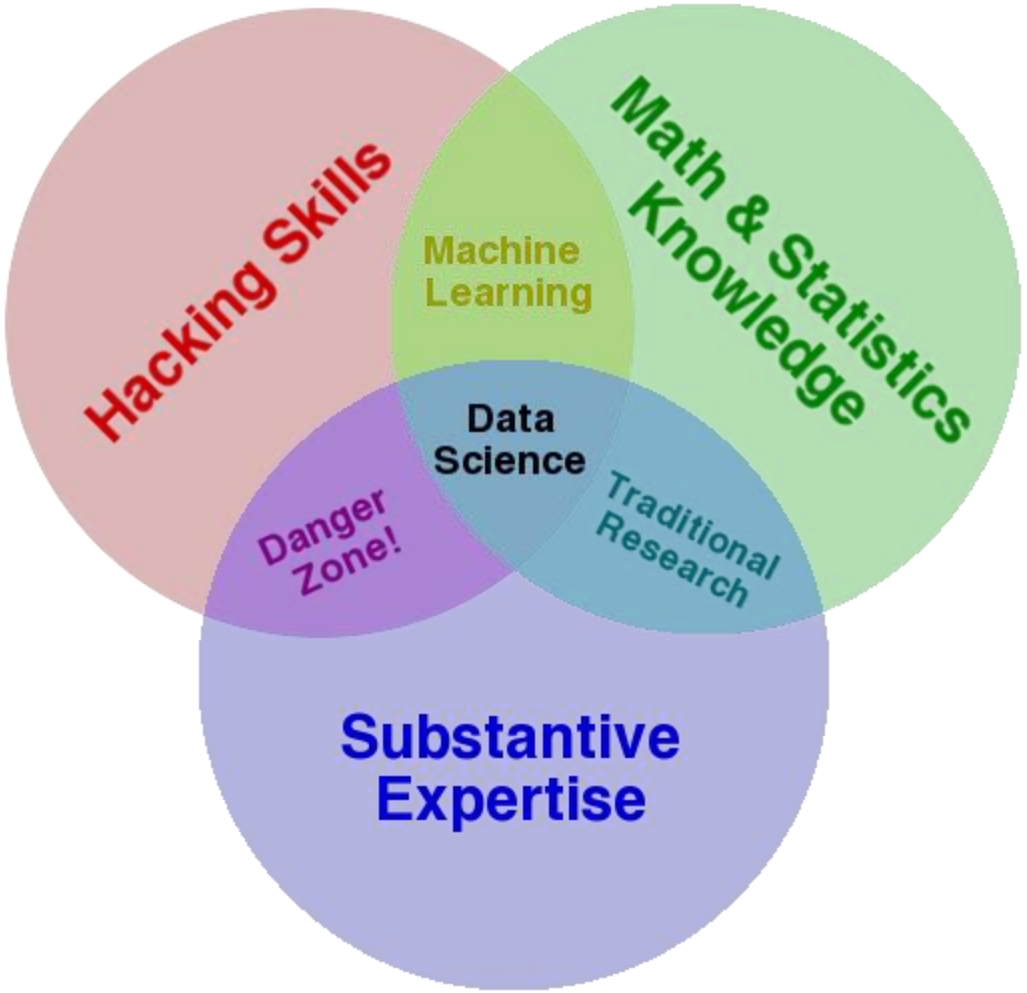
What is Data Science?

Data Science – A Deep Definition

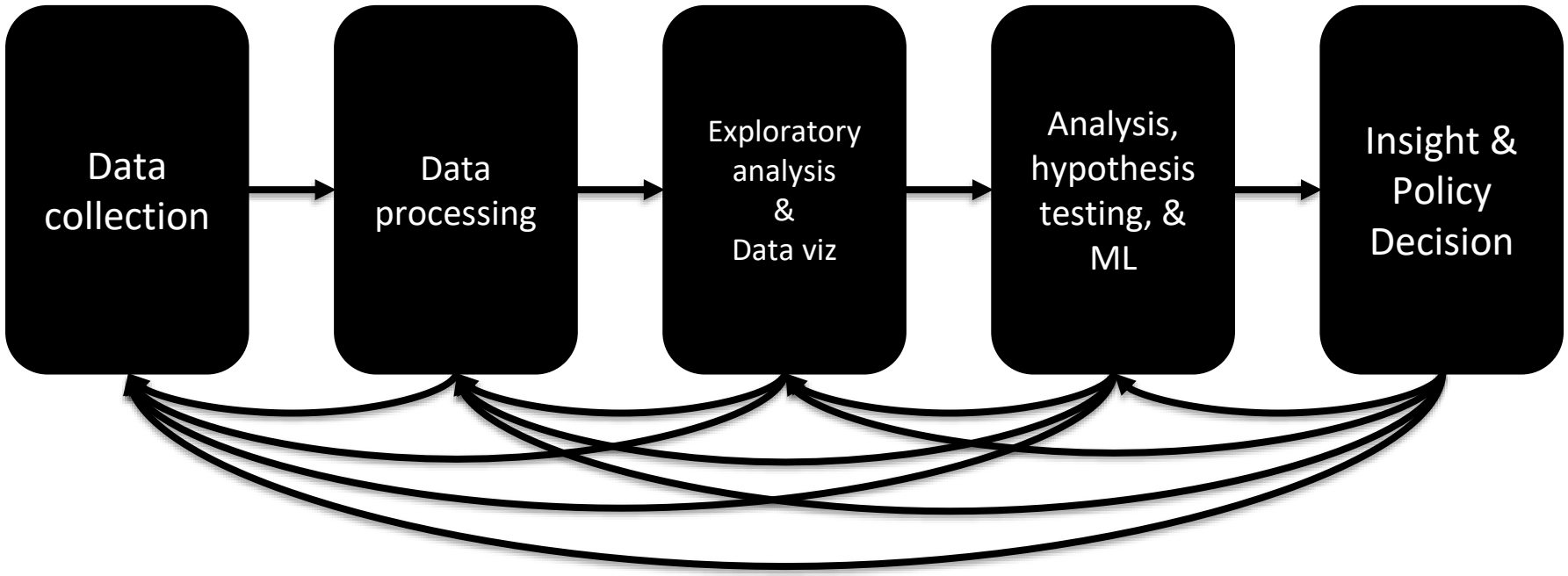
Data science combines math and statistics, specialized programming, advanced analytics, artificial intelligence (AI), and machine learning with specific subject matter expertise to uncover actionable insights hidden in an organization's data. These insights can be used to guide decision-making and strategic planning.

What is Data Science?

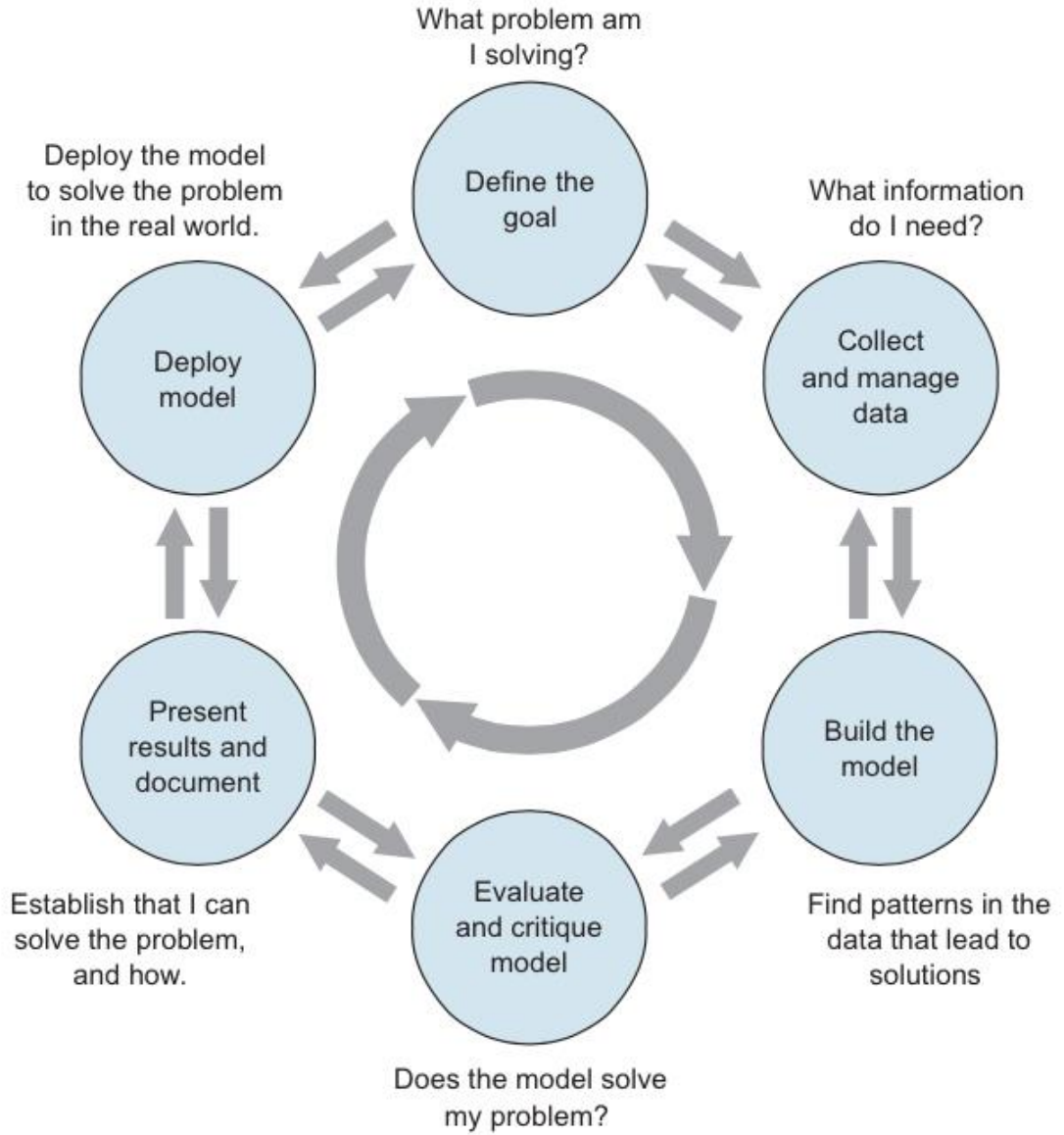
Data Science – One Definition



The Data Lifecycle



Data Science Lifecycle: An Alternate View



How to use data?

Data Science – A Definition

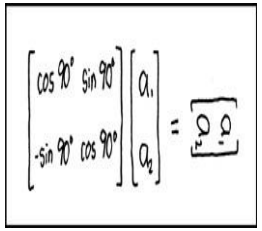
- Data => exploratory analysis => knowledge models => product / decision marking
- Data => predictive models => evaluate / interpret => product / decision making

How to use data?

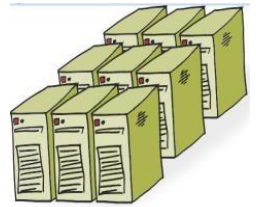
Data Scientist's Practice

Digging Around in Data

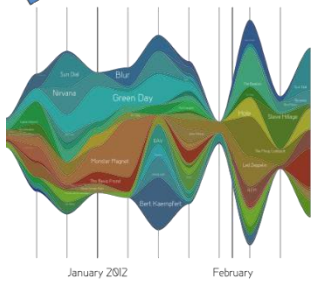
Clean, prep



Hypothesize Model



Large Scale Exploitation



Evaluate Interpret

What is data science used for?

1. Descriptive analysis examines data to gain insights into what happened or what is happening in the data environment.

2. Diagnostic analysis is a deep-dive or detailed data examination to understand why something happened.

3. Predictive analysis uses historical data to make accurate forecasts about data patterns that may occur in the future

4. Prescriptive analysis takes predictive data to the next level. It not only predicts what is likely to happen but also suggests an optimum response to that outcome

Data Science Techniques

Data Science Techniques:

- Classification is the sorting of data into specific groups or categories. Computers are trained to identify and sort data. Known data sets are used to build decision algorithms in a computer that quickly processes and categorizes the data.

- Regression is the method of finding a relationship between two seemingly unrelated data points. The connection is usually modelled around a mathematical formula and represented as a graph or curves. When the value of one data point is known.

- Clustering is the method of grouping closely related data together to look for patterns and anomalies. Clustering is different from sorting because the data cannot be accurately classified into fixed categories. Hence the data is grouped into most likely relationships.

Example of Data Science Applications

- Marketing: predict the characteristics of high life-time value (LTV) customers, which can be used to support customer segmentation, identify upsell opportunities, and support other marking initiatives
- Logistics: forecast how many of which things you need and where will we need them, which enables learn inventory and prevents out of stock situations
- Healthcare: analyze survival statistics for different patient attributes (age, blood type, gender, etc.) and treatments; predict risk of re-admittance based on patient attributes, medical history, etc.

Example of Data Science Applications

- Transaction Databases -> Recommender systems (Netflix), Fraud
- Detection (Security and Privacy)

- Wireless Sensor Data -> Smart Home, Real-time Monitoring, Internet of Things

- Text Data, Social Media Data -> Product Review and Consumer Satisfaction (Facebook, Twitter, LinkedIn), E-discovery

- Software Log Data -> Automatic Trouble Shooting (Splunk)

- Genotype and Phenotype Data -> Epic, 23andme, Patient-Centered Care, Personalized Medicine

Data Scientists Challenges

- Understanding the business problem
- Overcoming assumptions
- Making ad-hoc explanations of data patterns
- Overgeneralizing
- Communication
- Not checking enough (validate models, data pipeline integrity, etc.)
- Using statistical tests correctly
- Prototype -> Production transitions
- Data pipeline complexity (who do you ask?)
- Multiple data sources
- Elimination of bias

Examples of Bias

Genetic testing

- Genetic tests for heart disorder and race-biased risk (NYTimes)
- Race-bias in ancestry reports

Search results / feed optimization

- Google
- Facebook

Combating bias

Fairness through blindness:

- Don’t let an algorithm look at protected attributes

Examples currently in use ??????????

- Race
- Gender
- Sexuality
- Disability
- Religion

Problems with this approach ?????????

Combating Bias

Demographic parity:

- A decision must be independent of the protected attribute
- E.g., a loan application's acceptance rate is independent of an applicant's race (but can be depenedent on non-protected features like salary)

Formally: binary decision variable C, protected attribute A

- P{ C = 1 | A = 0 } = P{ C = 1 | A = 1 }

Membership in a protected class should have no correlation with the final decision.

- Problems ????????

Combating Bias

What if the decision isn't the thing that matters?
"Consider, for example, a luxury hotel chain that renders a promotion to a subset of wealthy whites (who are likely to visit the hotel) and a subset of less affluent blacks (who are unlikely to visit the hotel). The situation is obviously quite icky, but demographic parity is completely fine with it so long as the same fraction of people in each group see the promotion."

Demographic parity allows classifiers that select qualified candidates in the "majority" demographic and unqualified candidate in the "minority" demographic, within a protected attribute, so long as the expected percentages work out.

More: http://blog.mrtz.org/2016/09/06/approaching-fairness.html

FATML

This stuff is really tricky (and really important).

- It's also not solved, even remotely, yet!

New community: Fairness, Accountability, and Transparency in Machine Learning (aka FATML)

"… policymakers, regulators, and advocates have expressed fears about the potentially discriminatory impact of machine learning, with many calling for further technical research into the dangers of inadvertently encoding bias into automated decisions."

F is for Fairness

In large data sets, there is always proportionally less data available about minorities.
Statistical patterns that hold for the majority may be invalid for a given minority group.
Fairness can be viewed as a measure of diversity in the combinatorial space of sensitive attributes, as opposed to the geometric space of features.

A is for accountability

Accountability of a mechanism implies an obligation to report, explain, or justify algorithmic decision-making as well as mitigate any negative social impacts or potential harms.

- Current accountability tools were developed to oversee human decision makers
- They often fail when applied to algorithms and mechanisms instead

Example, no established methods exist to judge the intent of a piece of software. Because automated decision systems can return potentially incorrect, unjustified or unfair results, additional approaches are needed to make such systems accountable and governable.

T is for transparency

Automated ML-based algorithms make many important decisions in life.

- Decision-making process is opaque, hard to audit

A transparent mechanism should be:

- understandable;
- more meaningful;
- more accessible; and
- more measurable.

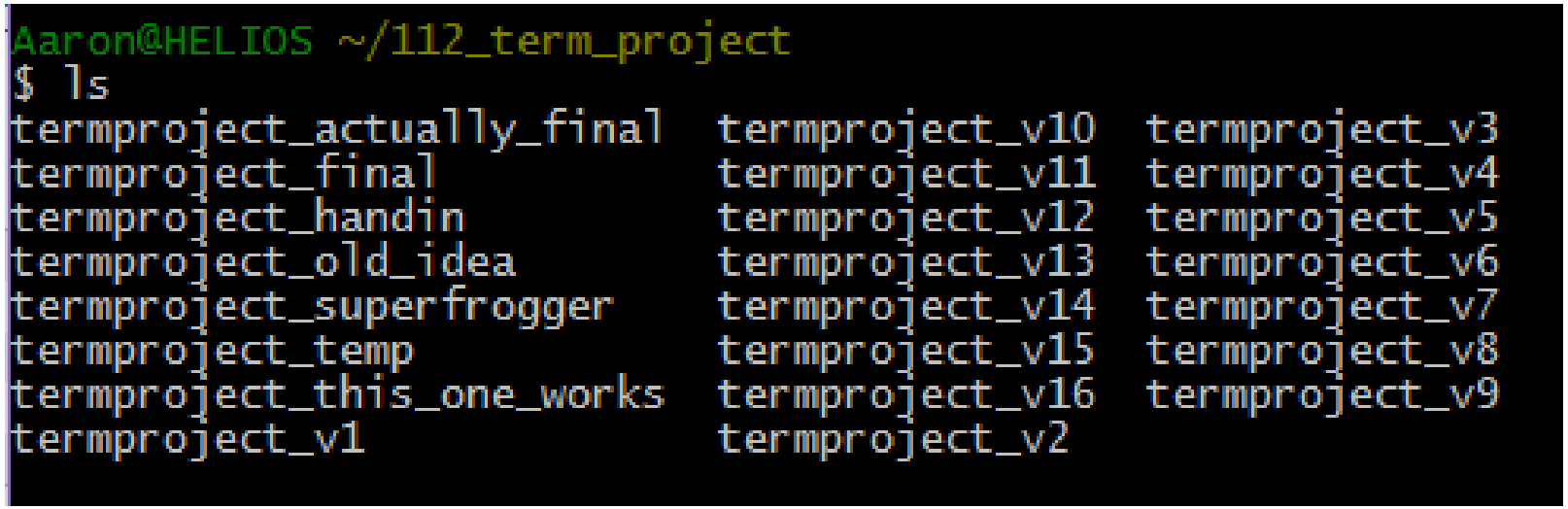
How to use Data Science (Group Discussion)

The Use of Data Science in Military and Law Enforcement?

Version Control Systems

Git and GitHub

What is version control?



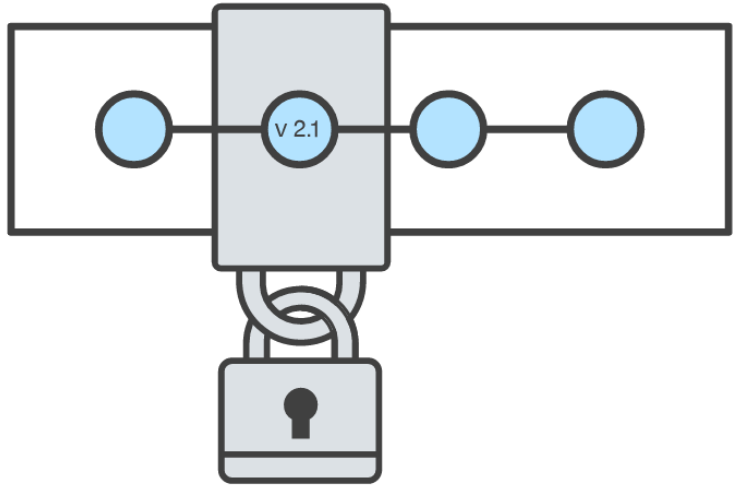
Development Tool

When working with a team, the need for a central repository is essential

- Need a system to allow versioning, and a way to acquire the latest edition of the code
- A system to track and manage bugs was also needed

Goals of Version Control

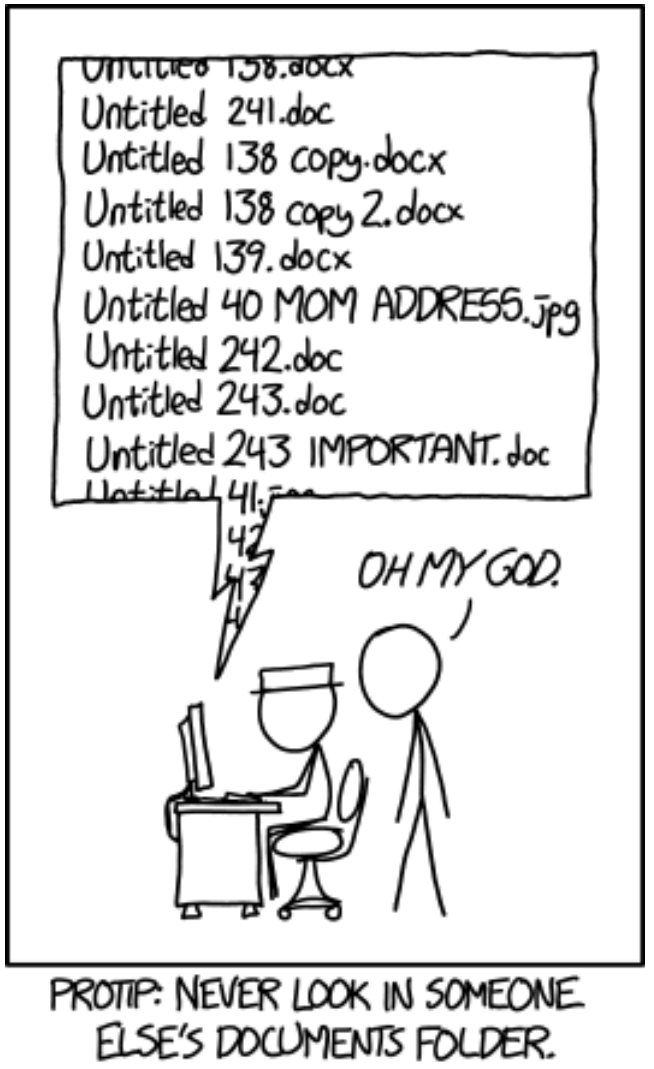
- Be able to search through revision history and retrieve previous versions of any file in a project
- Be able to share changes with collaborators on a project
- Be able to confidently make large changes to existing files



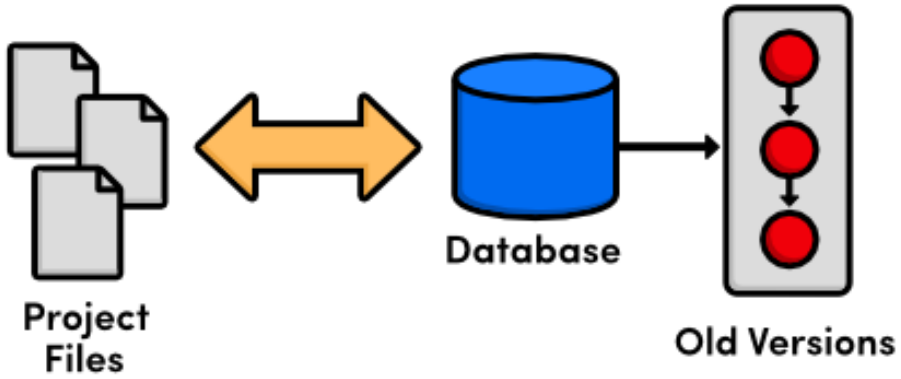
atlassian.com/git/tutorials/what-is-version-control

Named Folders Approach

- Can be hard to track
- Memory-intensive
- Can be slow
- Hard to share
- No record of authorship



Local Database of Versions Approach

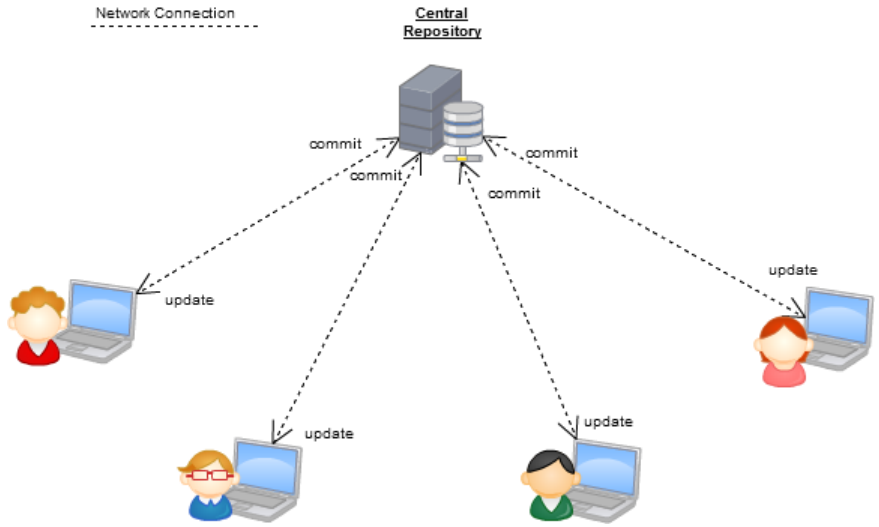


- Provides an abstraction over finding the right versions of files and replacing them in the project
- Records who changes what, but hard to parse that
- Can’t share with collaborators

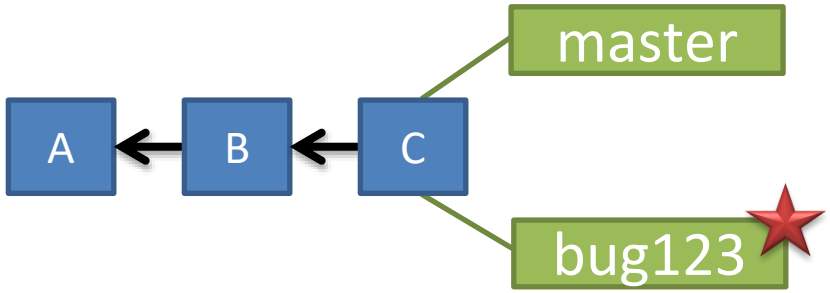
Centralized Version Control Systems

- A central, trusted repository determines the order of commits (“versions” of the project)
- Collaborators “push” changes (commits) to this repository.
- Any new commits must be compatible with the most recent commit. If it isn’t, somebody must “merge” its in.

- Examples: SVN, CVS, Perforce

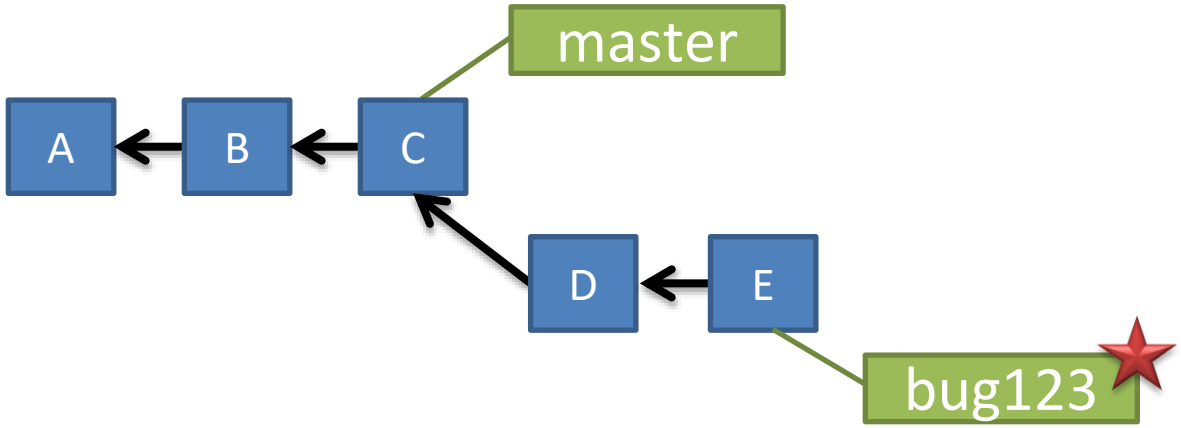


Branches Illustrated



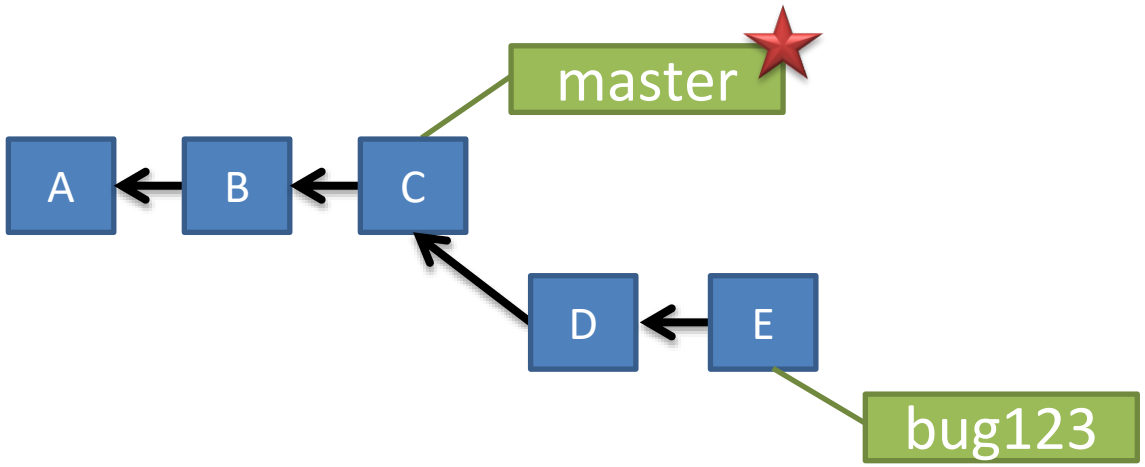

```
> git checkout -b bug123
```

Branches Illustrated



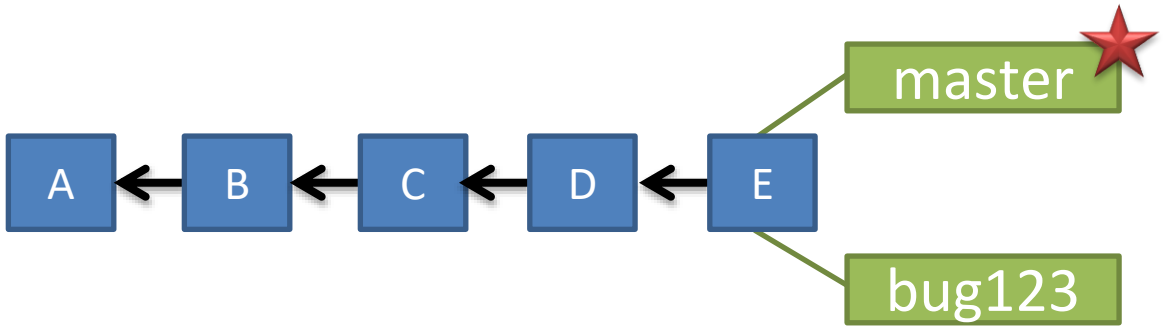

```
> git commit (x2)
```

Branches Illustrated



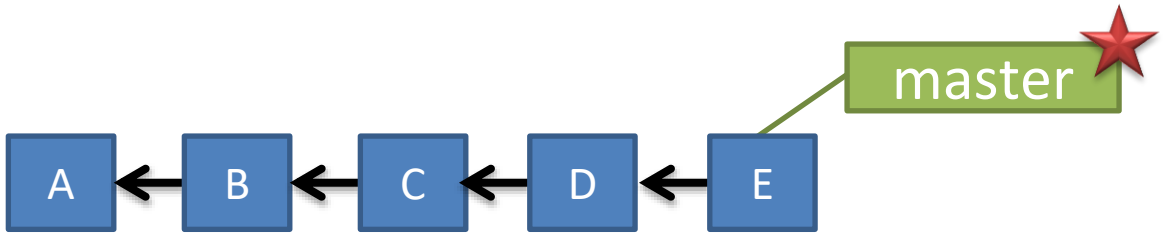

```
> git checkout master
```

Branches Illustrated



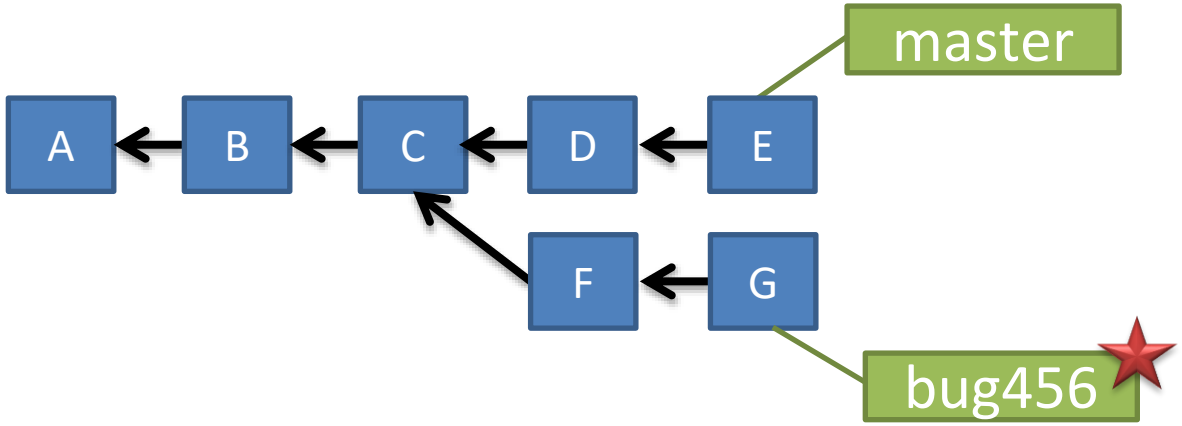

```
> git merge bug123
```

Branches Illustrated

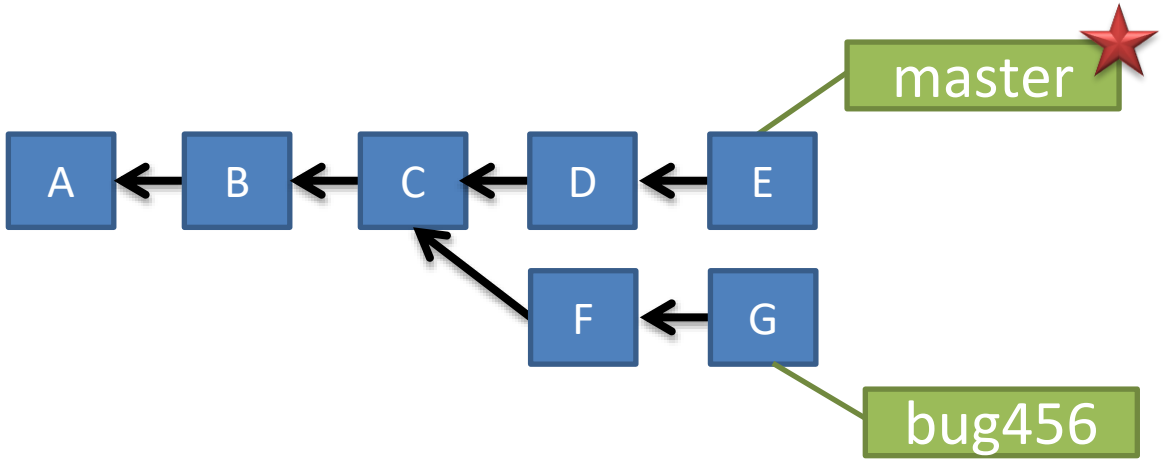



```
> git branch -d bug123
```

Branches Illustrated

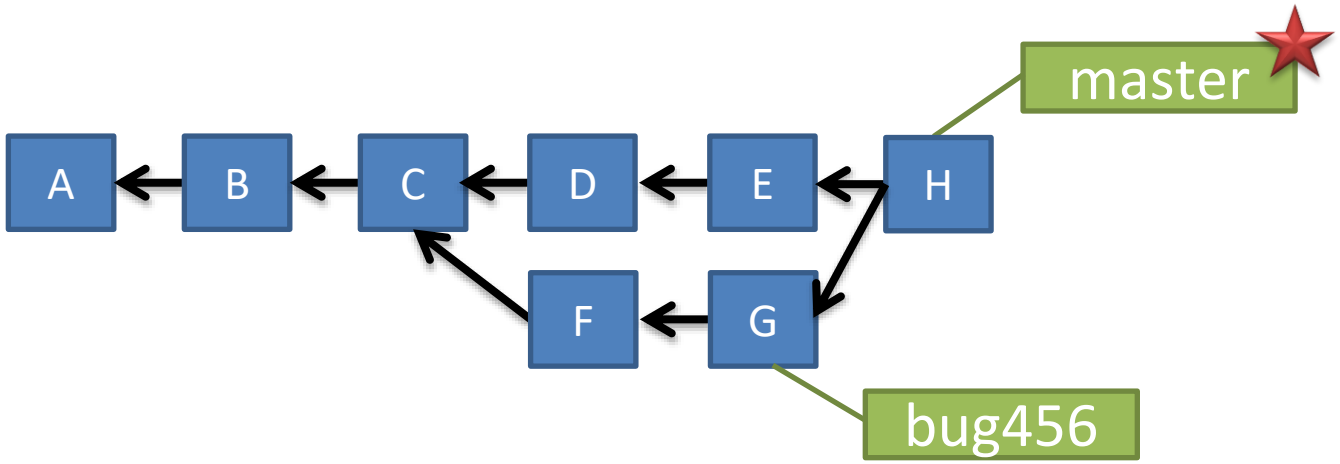


Branches Illustrated



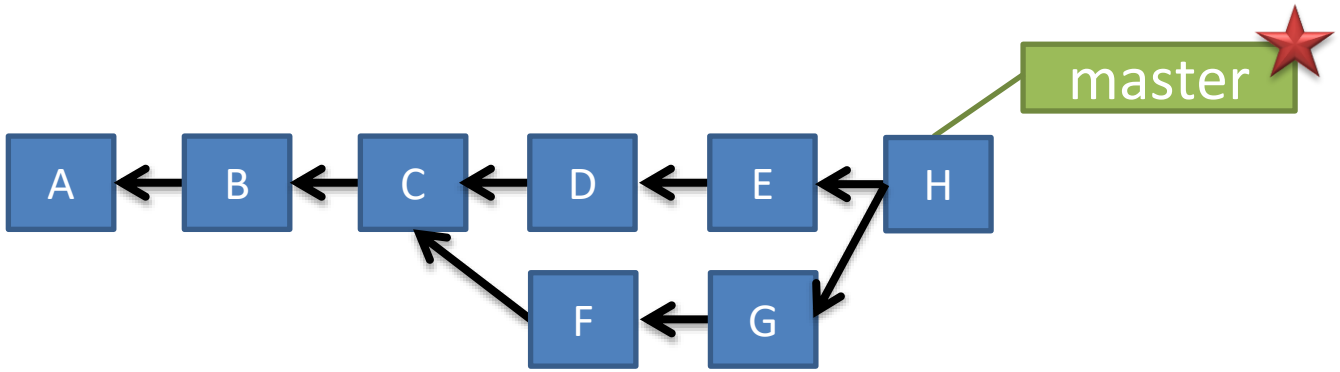

```
> git checkout master
```

Branches Illustrated



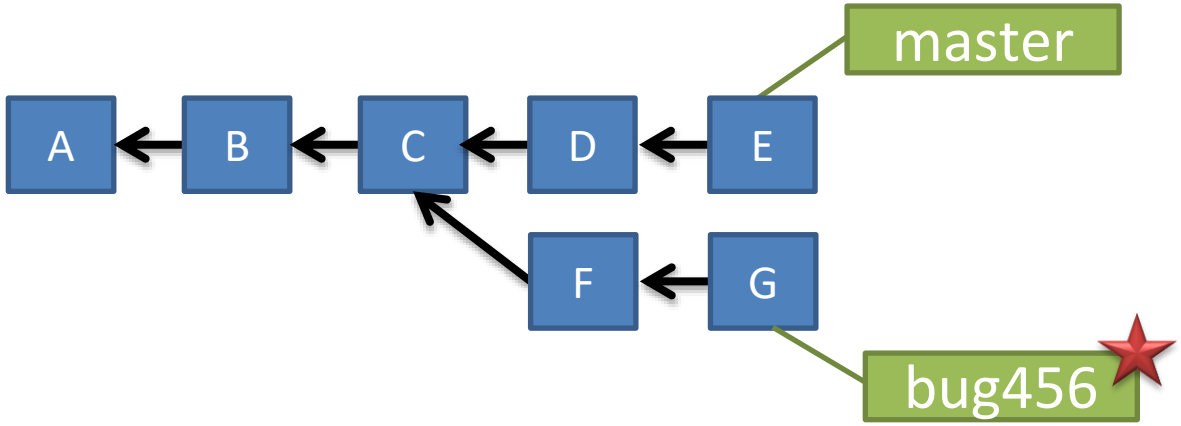

```
> git merge bug456
```

Branches Illustrated

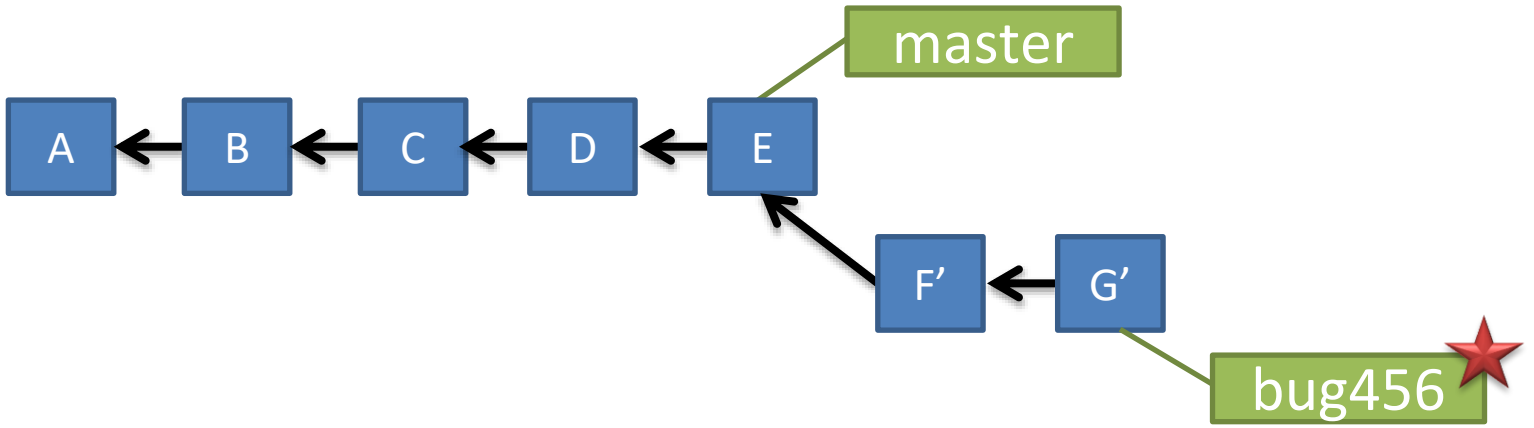



```
> git branch -d bug456
```

Branches Illustrated

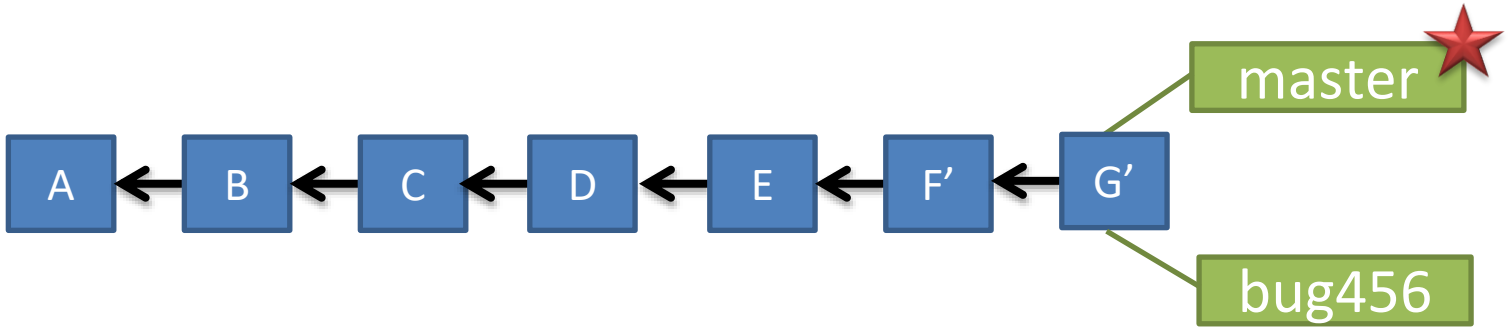


Branches Illustrated




```
> git rebase master
```

Branches Illustrated




```
> git checkout master
> git merge bug456
```

When to branch?

General rule of thumb:

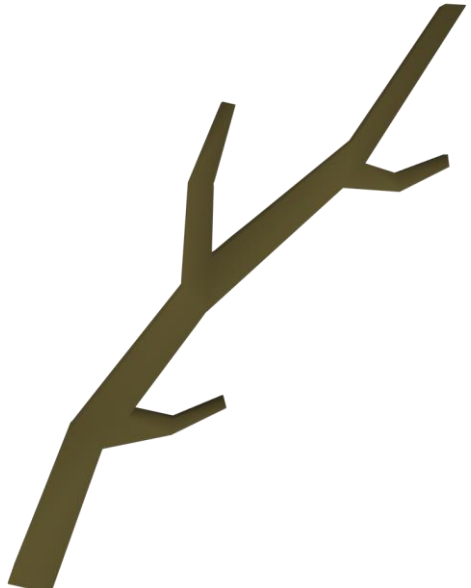
- Anything in the master branch is always deployable.

Local branching is very lightweight!

- New feature? Branch!
- Experiment that you won't ever deploy? Branch!

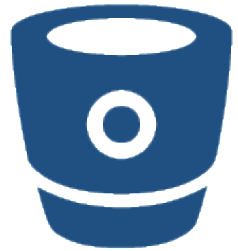
Good habits:

- Name your branch something descriptive (add-like-button, refactor-jobs, create-ai-singularity)
- Make your commit messages descriptive, too!



So, you want somebody else to host this for you

- Git: general distributed version control system
- GitHub / BitBucket / GitLab / …: hosting services for git repositories
- In general, GitHub is the most popular:
- Lots of big projects (e.g., Python, Bootstrap, Angular, D3, node, Django, Visual Studio)
- Lots of ridiculously awesome projects (e.g., https://github.com/maxbbraun/trump2cash)
- There are reasons to use the competitors (e.g., private repositories, access control)



Bitbucket

Review: How to Use

Git commands for everyday usage are relatively simple

- git pull
  Get the latest changes to the code
- git add .
  Add any newly created files to the repository for tracking
- git add –u
  Remove any deleted files from tracking and the repository
- git commit –m ‘Changes’
  Make a version of changes you have made
- git push
- Deploy the latest changes to the central repository
- Make a repo on GitHub and clone it to your machine: https://guides.github.com/activities/hello-world/

Stuff to click on

- Git http://git-scm.com/
- GitHub https://github.com/ and https://guides.github.com/activities/hello-world/
- GitLab http://gitlab.org/
- Git and SVN Comparison https://git.wiki.kernel.org/index.php/GitSvnComparison

Python IDE environment

installation & setup

12/9/2023

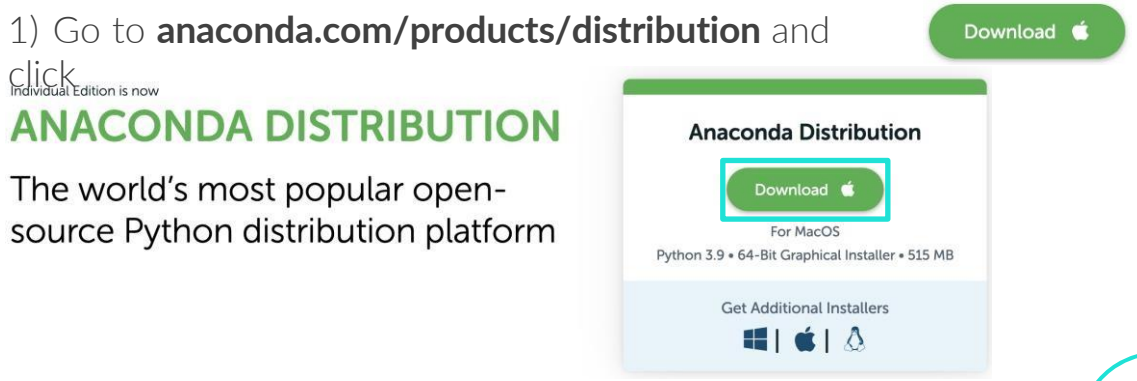
INSTALLING ANACONDA (MAC)

Installing Anaconda

Appending

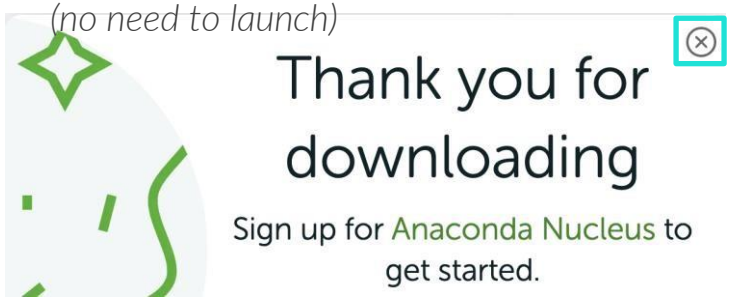
Joining

1) Go to **anaconda.com/products/distribution** and click

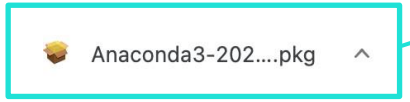


2) Click **X** on the Anaconda Nucleus pop-up

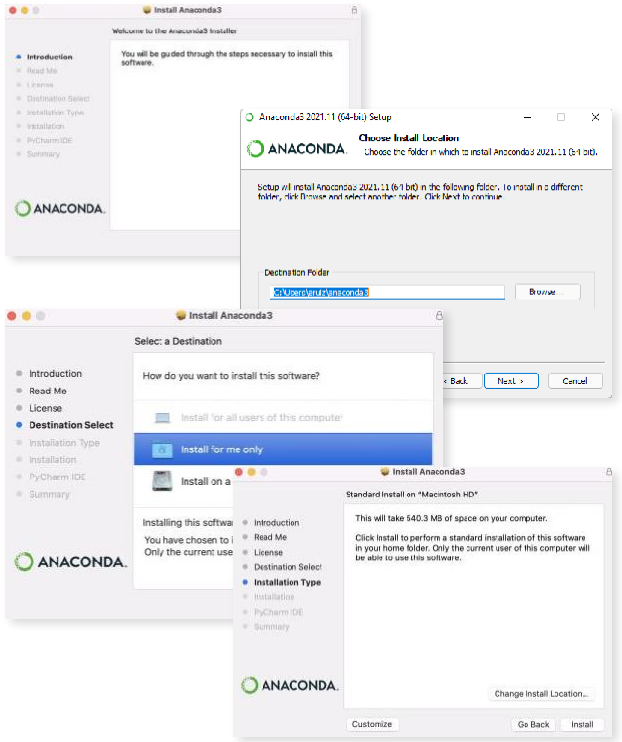
(no need to launch)



3) Launch the downloaded Anaconda **pkg** file



4) Follow the **installation steps**
(default settings are OK)



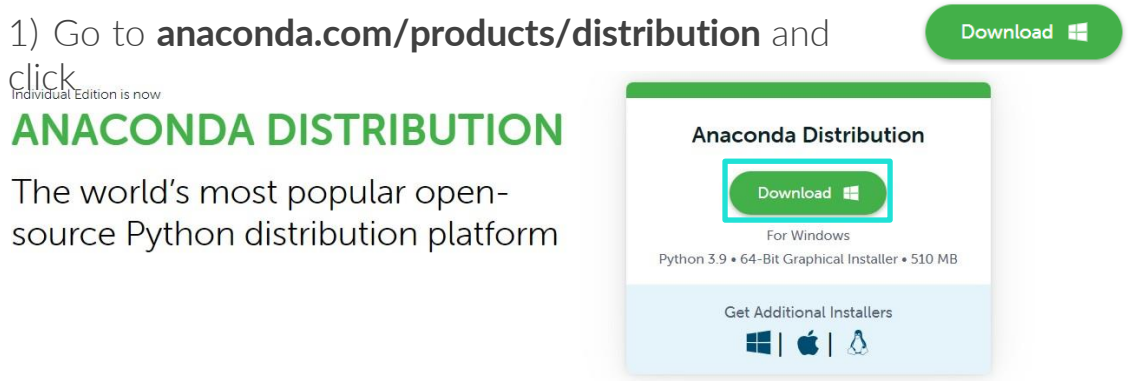
INSTALLING ANACONDA (PC)

Installing Anaconda

Appending

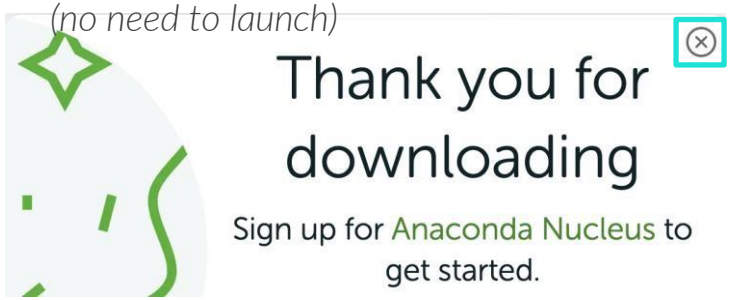
Joining

1) Go to **anaconda.com/products/distribution** and click

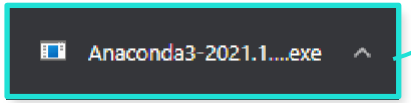


2) Click **X** on the Anaconda Nucleus pop-up

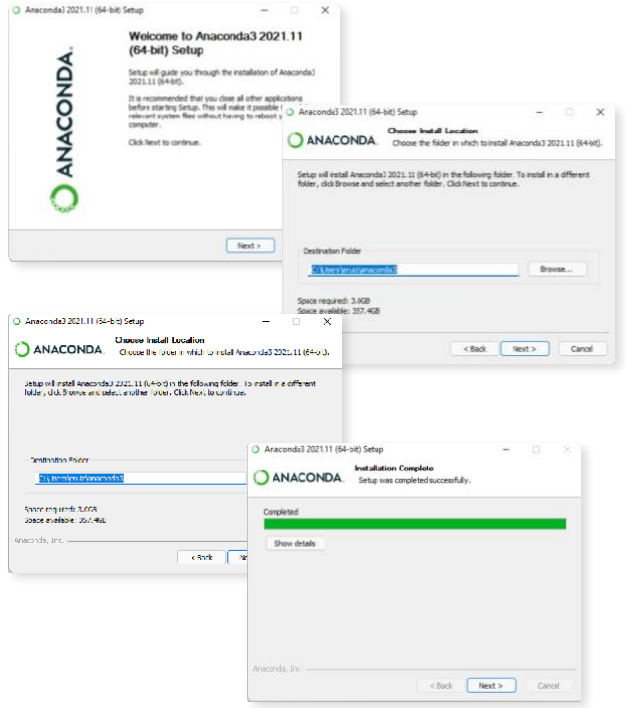
(no need to launch)



3) Launch the downloaded Anaconda **exe** file



4) Follow the **installation steps**
(default settings are OK)



LAUNCHING JUPYTER

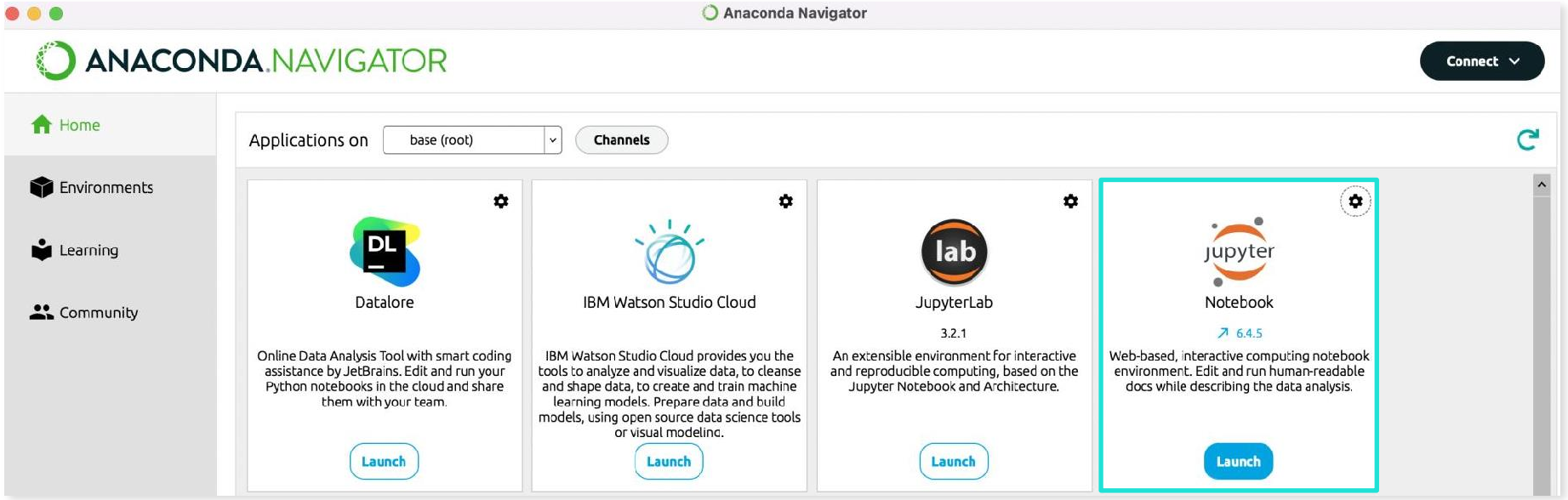
Installing Anaconda

Launching Jupyter

Joining

1) Launch **Anaconda Navigator**

2) Find **Jupyter Notebook** and click Launch



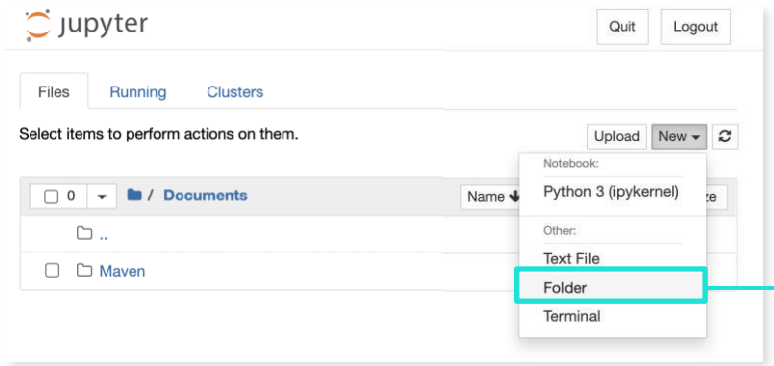
YOUR FIRST JUPYTER NOTEBOOK

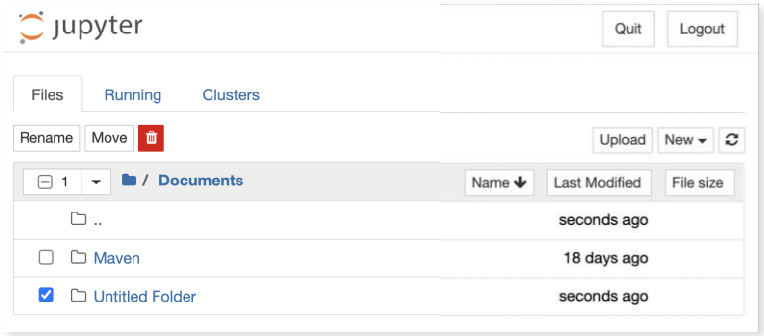
Installing Anaconda

Launching Jupyter

Joining

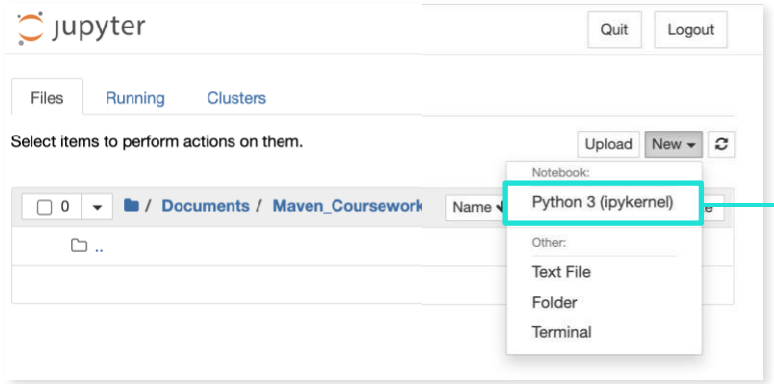
1) Once inside the Jupyter interface, **create a folder** to store your notebooks for the course

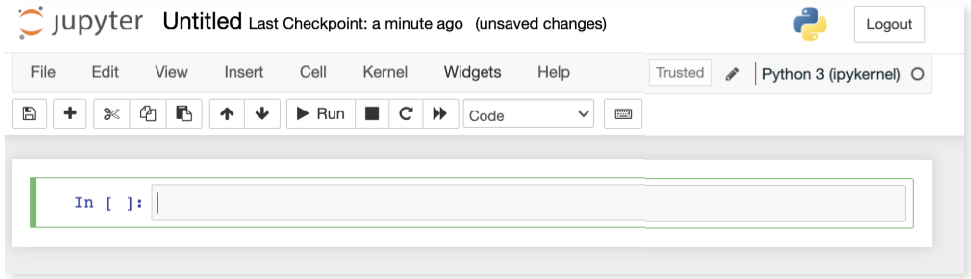




NOTE: *You can rename your folder by clicking "Rename" in the top left corner*

2) Open your new coursework folder and **launch your first Jupyter notebook**!





NOTE: *You can rename your notebook by clicking on the title at the top of the screen*

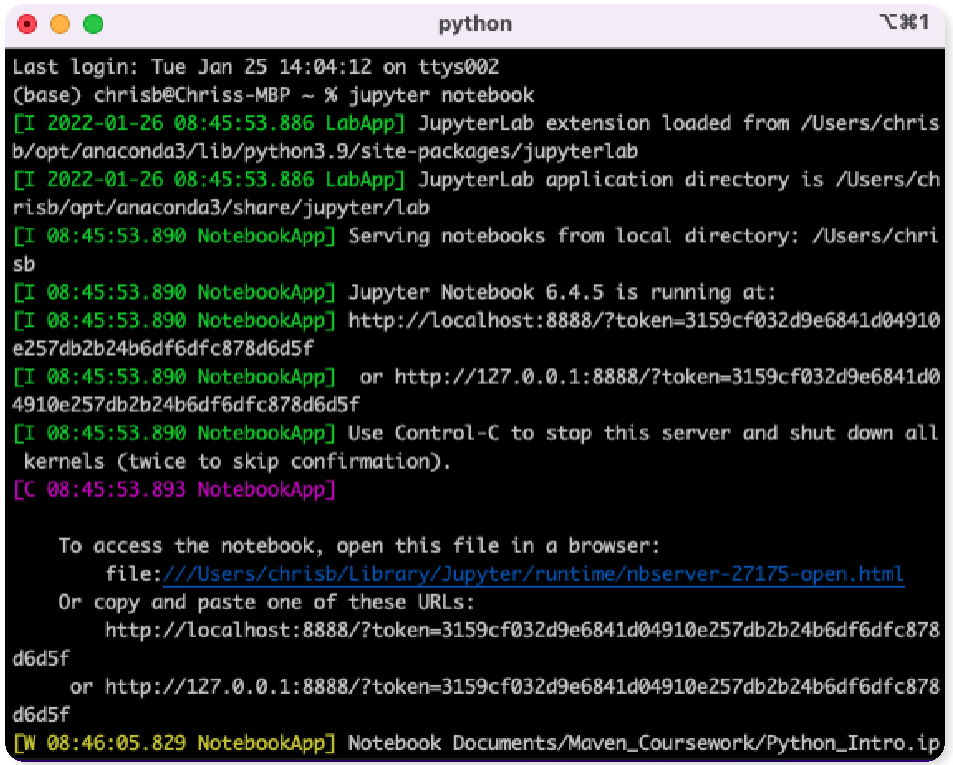
THE NOTEBOOK SERVER

Installing Anaconda

Launching Jupyter

Joining

NOTE: When you launch a Jupyter notebook, a terminal window may pop up as well; this is called a **notebook server**, and it powers the notebook interface



If you close the server window, **your notebooks will not run!**

Depending on your OS, and method of launching Jupyter, one may not open. As long as you can run your notebooks, don't worry!

ALTERNATIVE: GOOGLE COLAB

Installing Anaconda

Launching Jupyter

Google Colab

Google Colab is Google's cloud-based version of Jupyter Notebooks

To create a Colab notebook:

1. Log in to a Gmail account
2. Go to **colab.research.google.com**
3. Click "**new notebook**"

Colab is very similar to Jupyter Notebooks *(they even share the same file extension)*; the main difference is that you are connecting to **Google Drive** rather than your machine, so files will be stored in Google's cloud