INTRODUCTION TO DATA SCIENCE

JOHN P DICKERSON

Lecture #2 – 09/01/2021

CMSC320
Tuesdays & Thursdays
5:00pm – 6:15pm

https://cmsc320.github.io/

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

ANNOUNCEMENTS

Register on Piazza: piazza.com/umd/fall2021/cmsc320

- 221 have registered already ♡
- ~80 have not registered yet 💔

If you were on Piazza, you'd know …

- Project 0 is out! It is "due" next Tuesday evening.
- Link: **https://github.com/cmsc320/fall2021/tree/master/project0**

We've also linked some reading for the week!

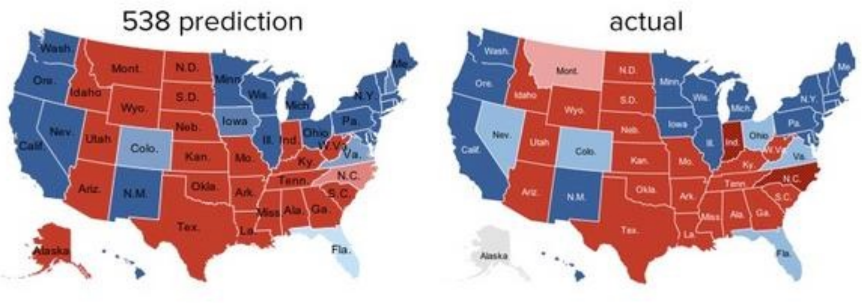
- First quiz is due Tuesday at noon.
- (Quiz is up on ELMS now.)

(A FEW) DATA SCIENCE SUCCESS STORIES & CAUTIONARY TALES

POLLING: 2008 & 2012

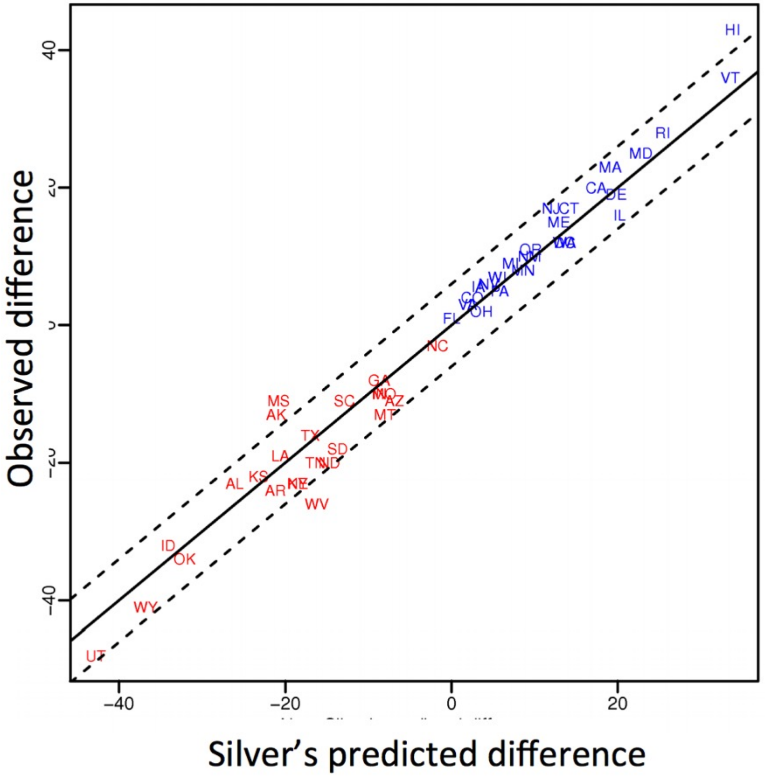
Nate Silver uses a simple idea – taking a principled approach to aggregating polling instead of relying on punditry – and:

- **Predicts 49/50 states in 2008**
- **Predicts 50/50 states in 2012**



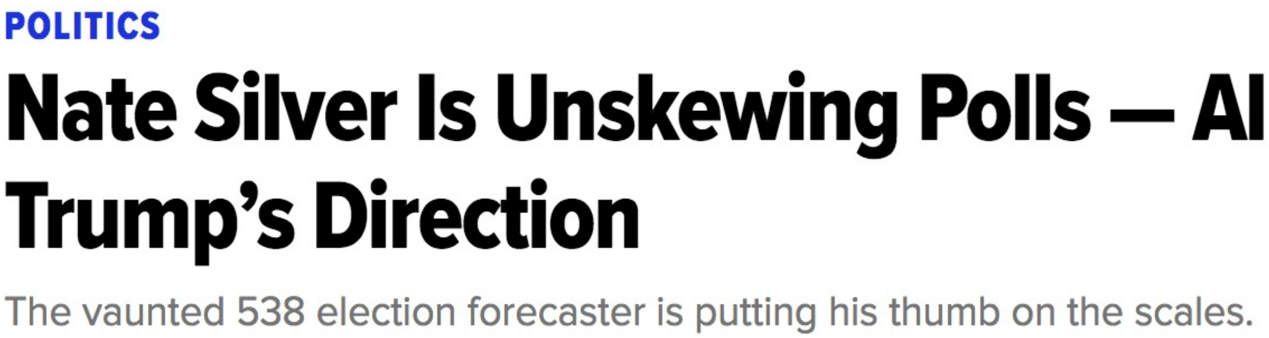
- **(He is also a great case study in creating a brand.)**

https://hbr.org/2012/11/how-nate-silver-won-the-2012-p



Democrat (+) or Republican (-) in 2012

POLLING: 2016



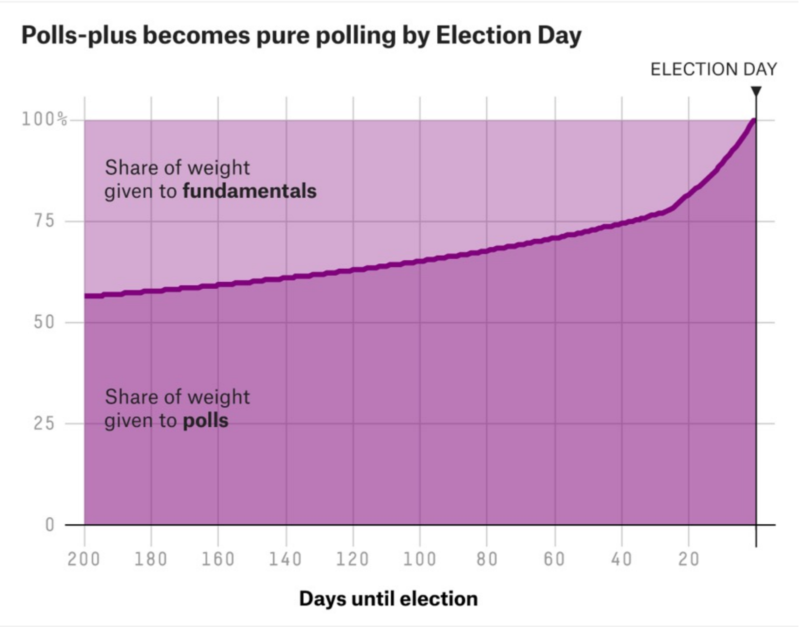
POLITICS

Nate Silver Is Unskewing Polls — All Of Them — In Trump's Direction

The vaunted 538 election forecaster is putting his thumb on the scales.

HuffPo: "He may end up being right, but he's just guessing. A "trend line adjustment" is merely political punditry dressed up as sophisticated mathematical modeling."

538: Offers quantitative reasoning for re-/under-weighting older polls, & changing as election approaches



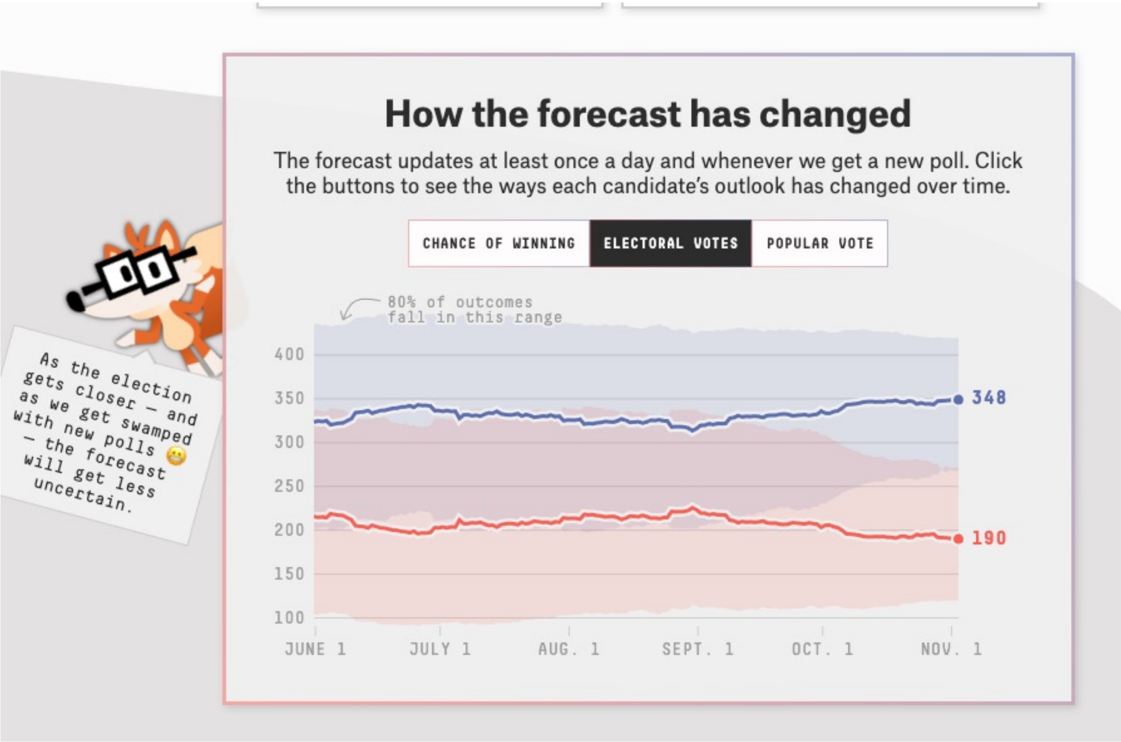
Polls-plus becomes pure polling by Election Day

http://www.huffingtonpost.com/entry/nate-silver-election-forecast_us_581e1c33e4b0d9ce6fbc6f7f
https://fivethirtyeight.com/features/a-users-guide-to-fivethirtyeights-2016-general-election-forecast/

POLLING: 2020

Lessons learned: don’t just communicate a binary prediction (X loses, Y wins):

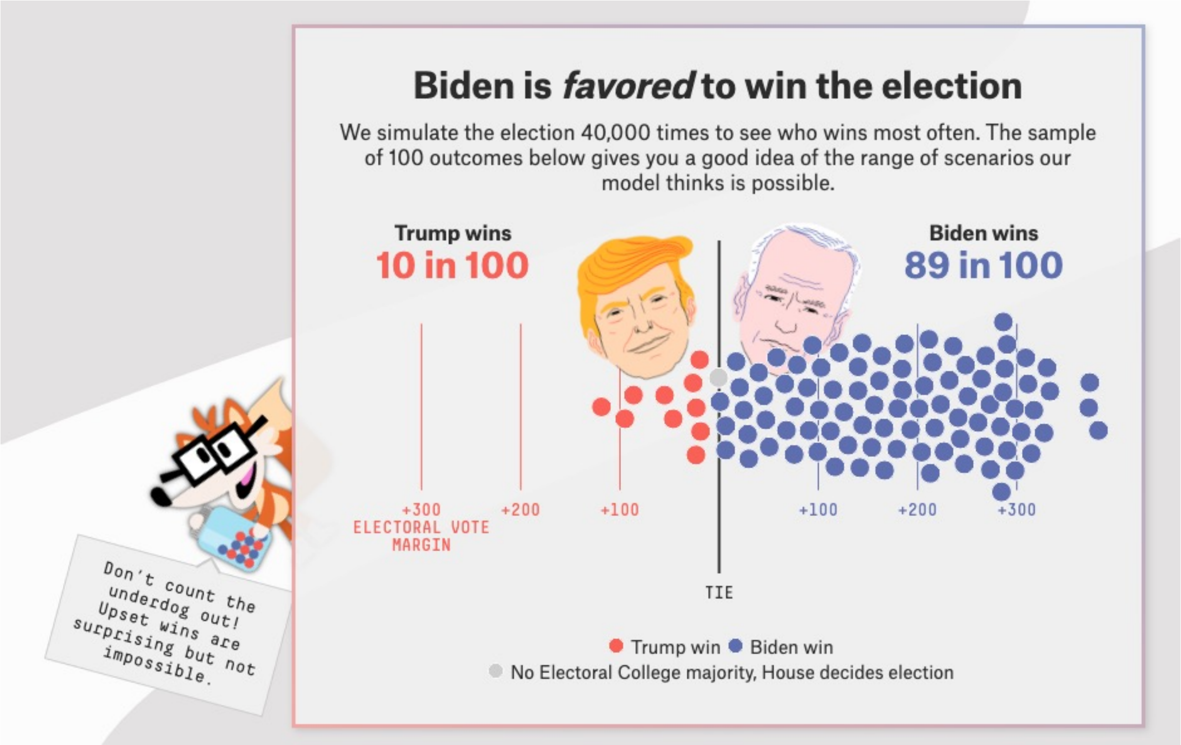
- Communicate uncertainty over that prediction
- Communicate variance in the underlying model(s), modeling errors, etc



How the forecast has changed

The forecast updates at least once a day and whenever we get a new poll. Click the buttons to see the ways each candidate’s outlook has changed over time.

As the election gets closer — and as we get swamped with new polls 😬 — the forecast will get less uncertain.



Biden is *favored* to win the election

We simulate the election 40,000 times to see who wins most often. The sample of 100 outcomes below gives you a good idea of the range of scenarios our model thinks is possible.

Don’t count the underdog out! Upset wins are surprising but not impossible.

AD TARGETING

TARGET

Pregnancy is an expensive & habit-forming time

- Thus, valuable to consumer-facing firms

2012:

- Target identifies 25 products and subsets thereof that are commonly bought in early pregnancy
- Uses purchase history of patrons to predict pregnancy, targets advertising for post-natal products (cribs, etc)
- Good: increased revenue
- Bad: this can expose pregnancies – as famously happened in Minneapolis to a high schooler

http://www.businessinsider.com/the-incredible-story-of-how-target-exposed-a-teen-girls-pregnancy-2012-2

AUTOMATED DECISIONS OF CONSEQUENCE

[Sweeney 2013, Miller 2015, Byrnes 2016, Rudin 2013, Barry-Jester et al. 2015]

textio
DOXA
gild
GJ GapJumpers

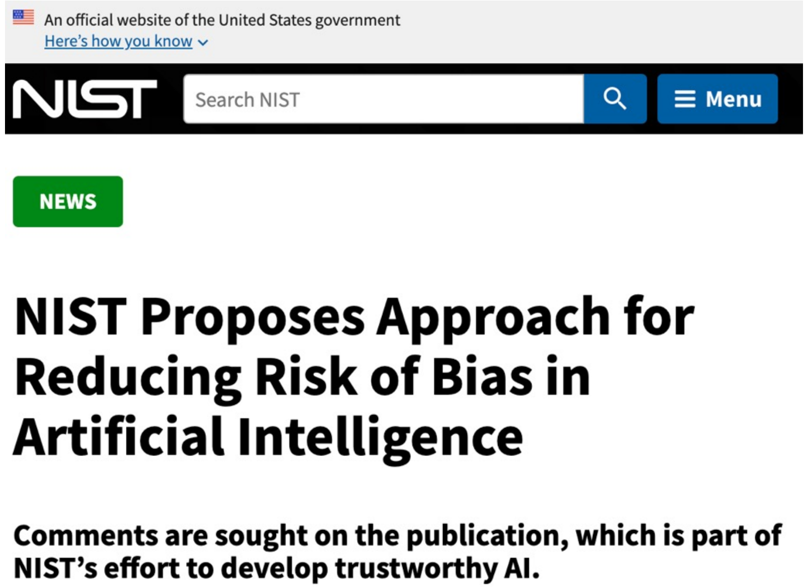
Hiring

zest finance
LendingClub
Kabbage

Lending

PREDPOL
NORTHPOINTE
Advancing Justice. Embracing Community.

Policing/ sentencing



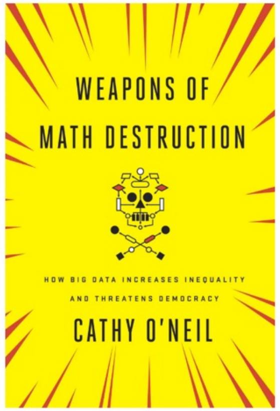
An official website of the United States government
Here's how you know

NIST Proposes Approach for Reducing Risk of Bias in Artificial Intelligence

Comments are sought on the publication, which is part of NIST's effort to develop trustworthy AI.

Search for minority names →
ads for DUI/arrest records

Female cookies →
less freq. shown professional job opening ads



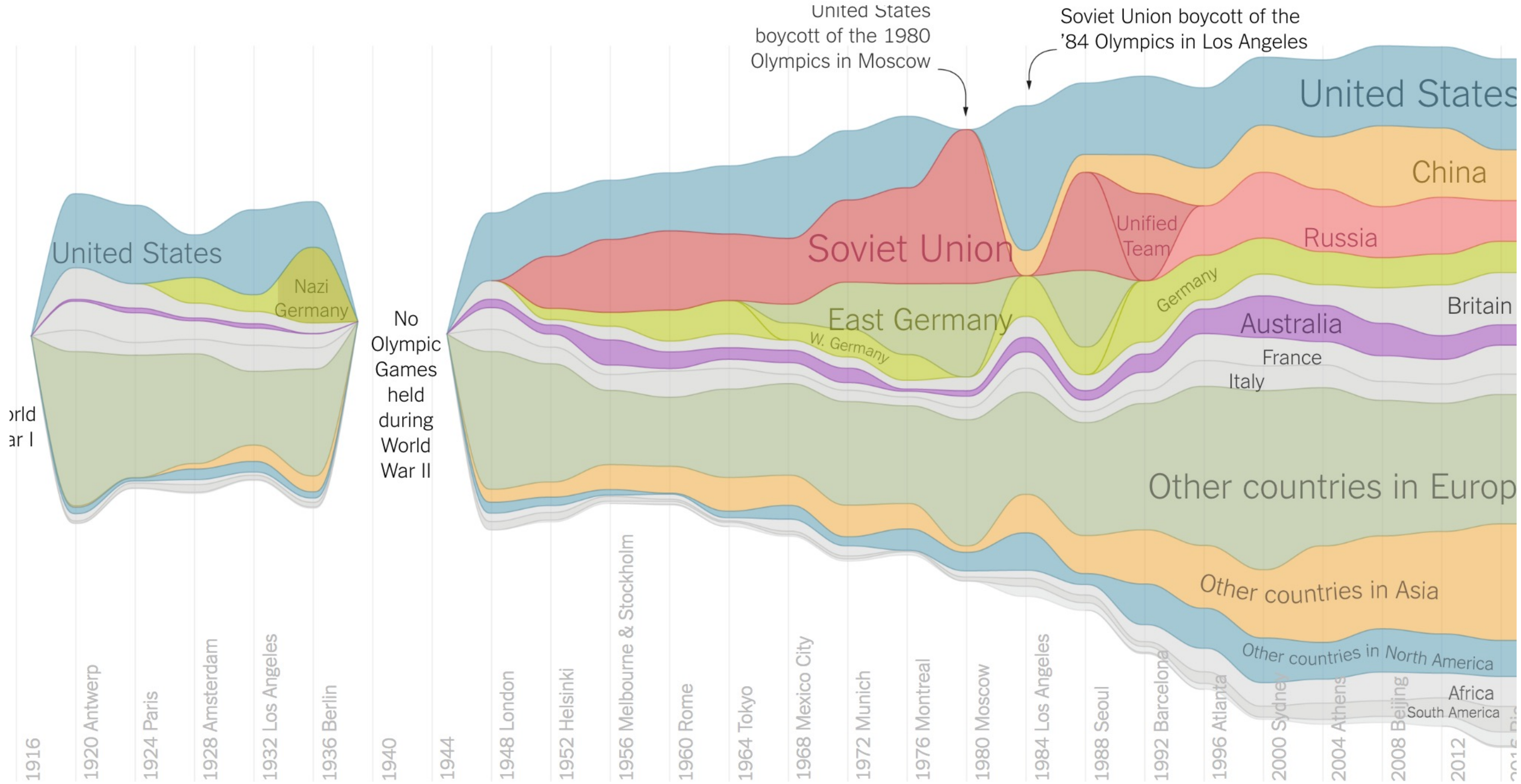
"… a lot remains unknown about how big data-driven decisions may or may not use factors that are proxies for race, sex, or other traits that U.S. laws generally prohibit from being used in a wide range of commercial decisions … What can be done to make sure these products and services–and the companies that use them treat consumers fairly and ethically?"

- FTC Commissioner Julie Brill [2015]

"**Hold yourself accountable – or be ready for the FTC to do it for you.** As we've noted, it's important to hold yourself accountable for your algorithm's performance. … keep in mind that if you don't hold yourself accountable, the FTC may do it for you."

- FTC official blog post, "Aiming for truth, fairness, and equity in your company's use of AI", written by Elisa Jillson [2021]

OLYMPIC MEDALS


United States boycott of the 1980 Olympics in Moscow

Soviet Union boycott of the '84 Olympics in Los Angeles

No Olympic Games held during World War II

https://www.nytimes.com/interactive/2016/08/08/sports/olympics/history-olympic-dominance-charts.html

NETFLIX PRIZE I

Recommender systems: predict a user's rating of an item

	Twilight	Wall-E	Twilight II	Furious 7
User 1	+1	-1	+1	?
User 2	+1	-1	?	?
User 3	-1	+1	-1	+1

Netflix Prize: $1MM to the first team that beats our in-house engine by 10%

- **Happened after about three years**
- **Model was never used by Netflix for a variety of reasons**
  - Out of date (DVDs vs streaming)
  - Too complicated / not interpretable

NETFLIX PRIZE II

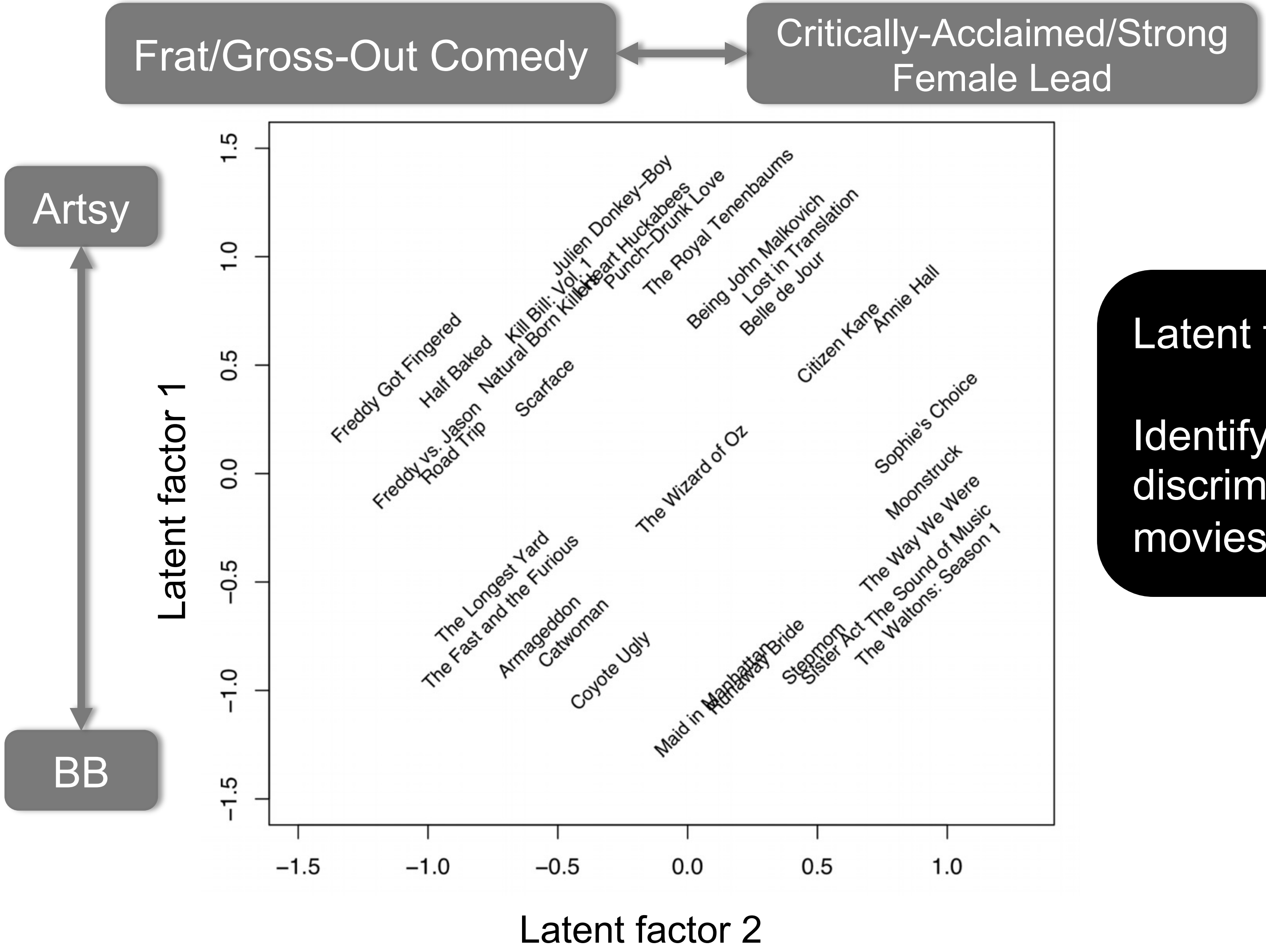


Image courtesy of Christopher Volinsky

Latent factors model:

Identify factors with max discrimination between movies

NETFLIX PRIZE III

Netflix initially planned a follow-up competition

In 2007, UT Austin managed to deanonymize portions of the original released (anonymized) Netflix dataset:

- ????????????
- Matched rating against those made publicly on IMDb

Why could this be bad?

2009—2010, four Netflix users filed a class-action lawsuit against Netflix over

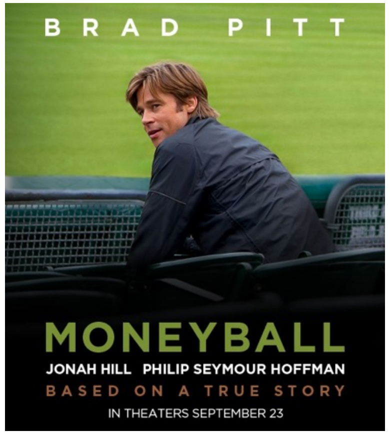
MONEYBALL

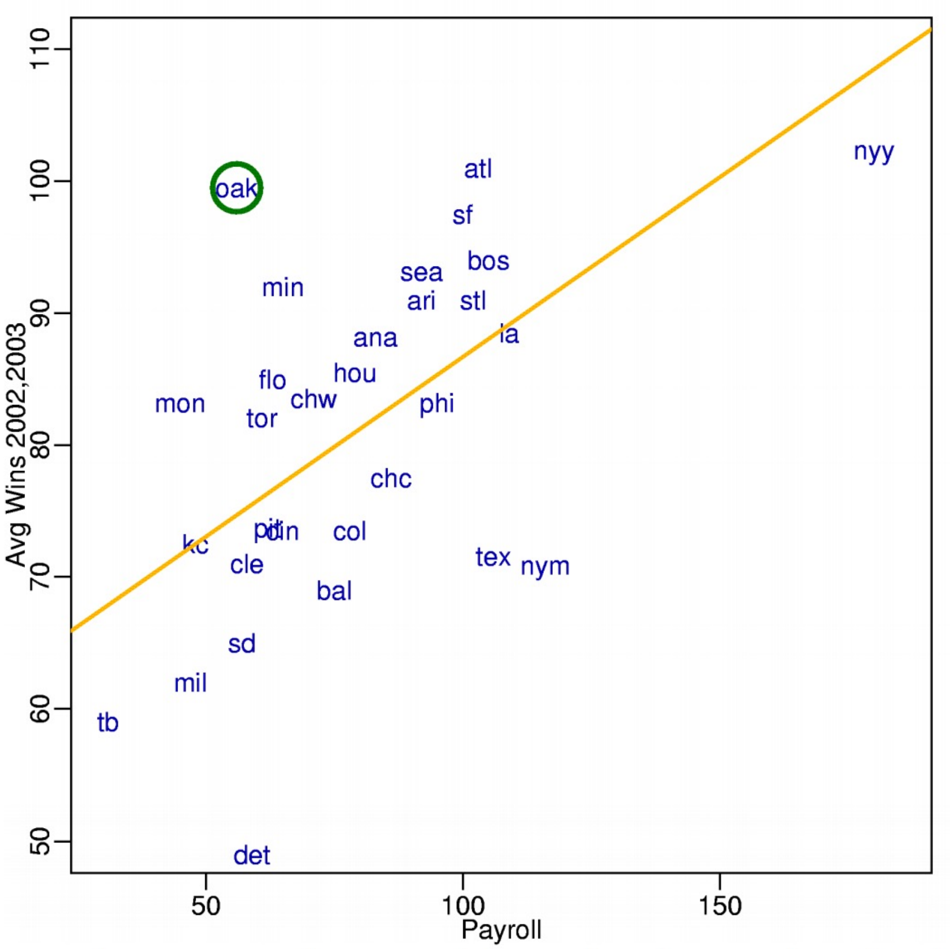
Baseball teams drafted rookie players primarily based on human scouts' opinions of their talents

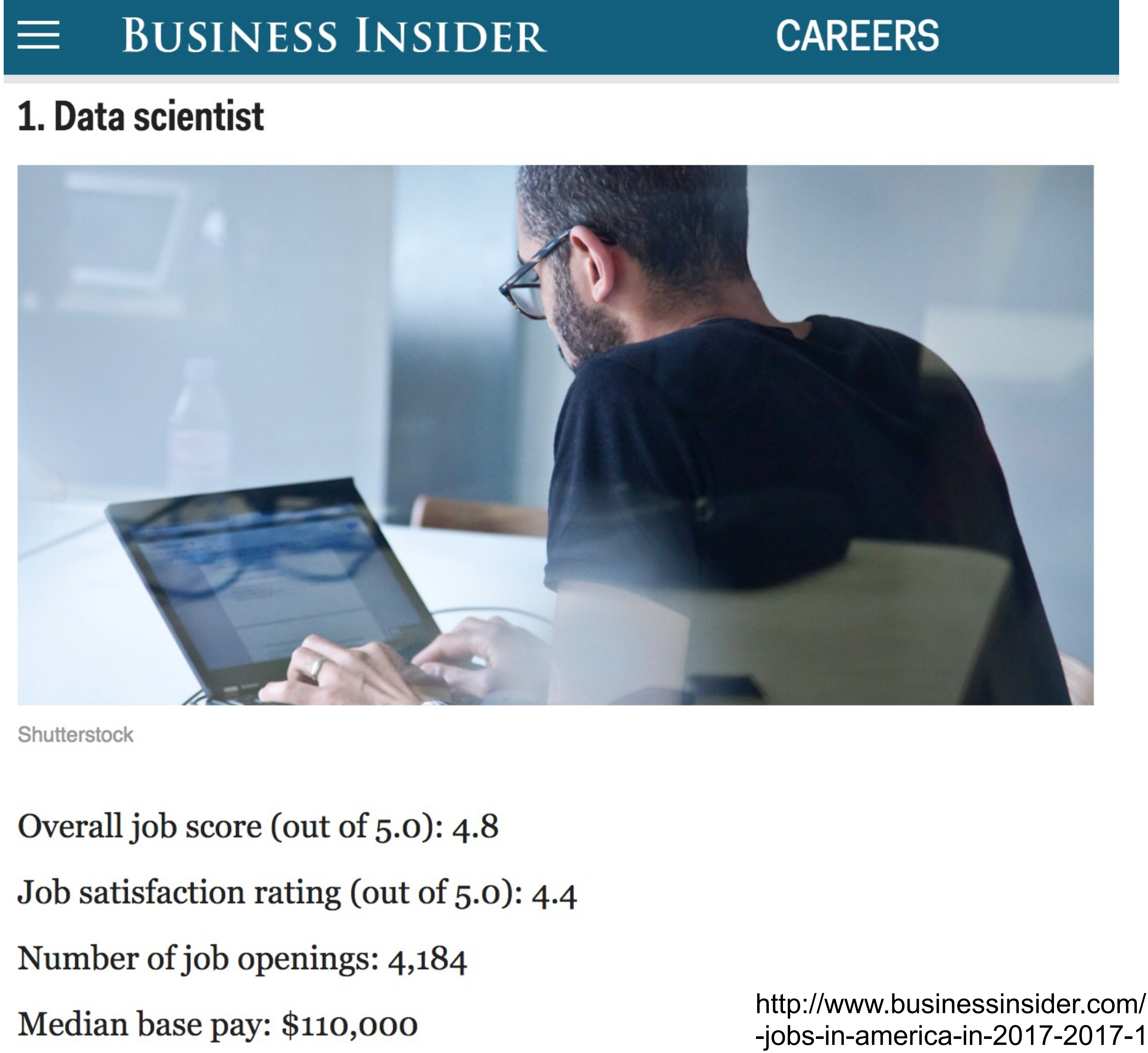
Paul DePodesta, data scientist *du jour*, convinces the {bad, poor} Oakland Athletics to use a quantitative aka sabermetric approach to hiring

(Spoiler: Red Sox offer Brand a job, he says no, they take a sabermetric approach and win the World Series.)

(Spoiler #2: DePodesta is now Chief Strategy Officer for the Browns, and they extended his contract in 2021, so we'll see what happens!)







1. Data scientist

Shutterstock

Overall job score (out of 5.0): 4.8

Job satisfaction rating (out of 5.0): 4.4

Number of job openings: 4,184

Median base pay: $110,000

http://www.businessinsider.com/best-jobs-in-america-in-2017-2017-1/

WRAP-UP FOR NOW

Register on Piazza using your UMD address:

piazza.com/umd/fall2021/cmsc320

Please get in touch with me if you're unsure of whether or not you're at the right {programming, math} level for this course:

- My guess is that you are!
- This is a young class, so we're quite flexible

Tonight, read about Docker & Jupyter!

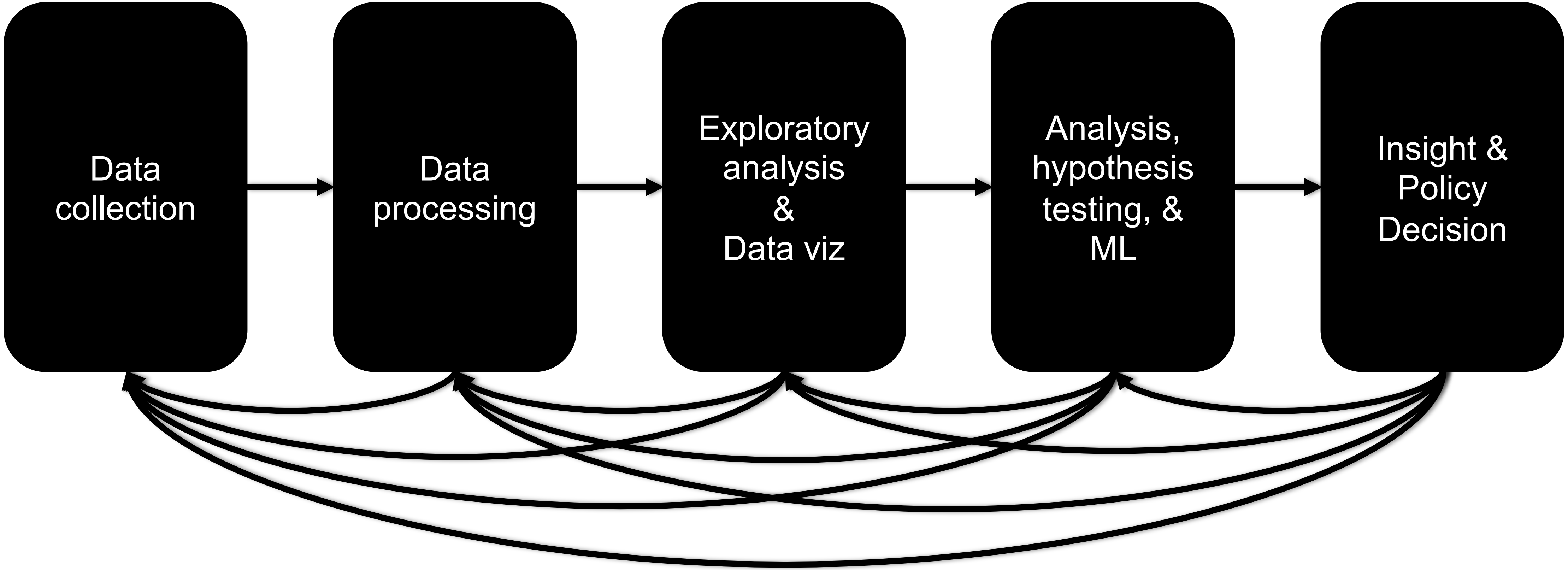
- Works on *nix, OSX, Windows
- https://www.docker.com/

NEXT (AKA THE REST OF THIS) CLASS:

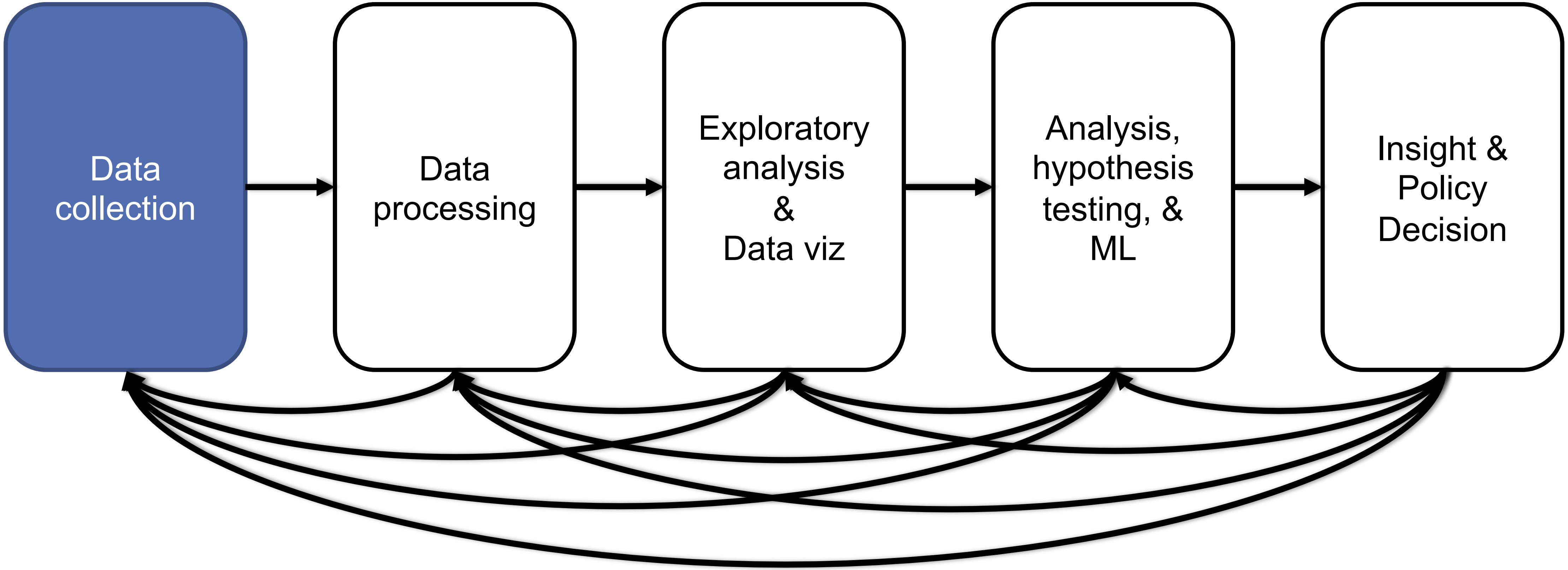
SCRAPING DATA WITH PYTHON

python™

THE DATA LIFECYCLE


Data collection
Data processing
Exploratory analysis & Data viz
Analysis, hypothesis testing, & ML
Insight & Policy Decision

(THE REST OF) TODAY'S LECTURE


Data collection
Data processing
Exploratory analysis & Data viz
Analysis, hypothesis testing, & ML
Insight & Policy Decision

BUT FIRST, SNAKES!

Python is an interpreted, dynamically-typed, high-level, garbage-collected, object-oriented-functional-imperative, and widely used scripting language.

- **Interpreted:** instructions executed without being compiled into (virtual) machine instructions*
- **Dynamically-typed:** verifies type safety at runtime
- **High-level:** abstracted away from the raw metal and kernel
- **Garbage-collected:** memory management is automated
- **OOFI:** you can do bits of OO, F, and I programming

Not the point of this class!

- Python is fast (developer time), intuitive, and used in industry!

*you can compile Python source, but it's not required

THE ZEN OF PYTHON

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules …
- … although practicality beats purity.
- Errors should never pass silently …
- … unless explicitly silenced.

Thanks: SDSMT ACM/LUG

LITERATE PROGRAMMING

Literate code contains in one document:

- the source code;
- text explanation of the code; and
- the end result of running the code.

Basic idea: present code in the order that logic and flow of human thoughts demand, not the machine-needed ordering

- Necessary for data science!
- Many choices made need textual explanation, ditto results.

Stuff you'll be using in Project 0 (and beyond)!

IP[y]: IPython Interactive Computing

jupyter

JUPYTER PROJECT

Started as iPython Notebooks, a web-based frontend to the iPython Shell

- The “notebook” functionality separated out years ago
- Now supports over 40 languages/kernels
- Notebooks can be shared easily
- Can leverage big data tools like Spark

Apache Zeppelin:

- https://www.linkedin.com/pulse/comprehensive-comparison-jupyter-vs-zeppelin-hoc-q-phan-mba-

Several others, e.g., RStudio (specific to R) – can run R via Jupyter, too (IMO, worse!)

GOOGLE COLAB

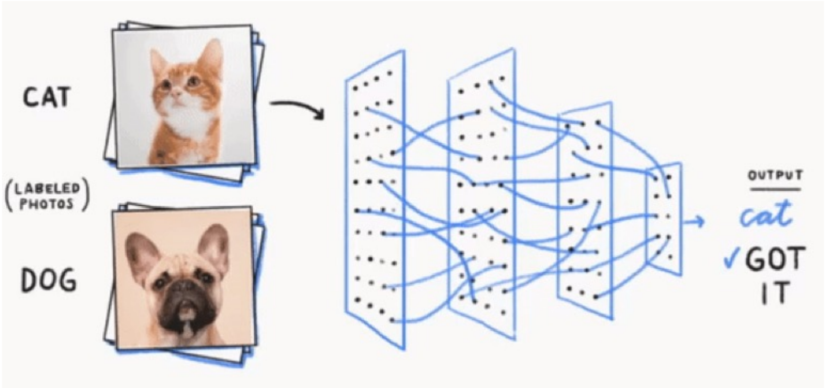
Recall: Jupyter Notebooks are web-based frontends that let you execute arbitrary Python in your browser

- What if you want to run some heavy computation (e.g., train a deep net from scratch, or even just do inference on a pre-trained net)?

Google Colab(oratory) lets you use Google's GPUs and TPUs to do heavy lifting, generally for free

- Notebooks hosted on Google Drive or Github
- Share them with other people with no install needed
- Here's an example CMSC320 tutorial from Fall 2020 that is available on Colab: https://colab.research.google.com/drive/1co_-qwtCmwl_0hCU5Vrz0Lcnp-Qvy_4_

You'll see and use this, and similar cloud services, in ML / data science land.



Google colab

10-MINUTE PYTHON PRIMER

Define a function:

```
def my_func(x, y):
    if x > y:
        return x
    else:
        return y
```

Python is whitespace-delimited

Define a function that returns a tuple:

```
def my_func(x, y):
    return (x-1, y+2)

(a, b) = my_func(1, 2)
```

```
a = 0; b = 4
```

```
def interview(n):

    if n % 3 == 0 and n % 5 == 0:
        return 'FizzBuzz'
    elif n % 3 == 0:
        return 'Fizz'
    elif n % 5 == 0:
        return 'Buzz'
    else:
        return str(n)

print( "\n".join(interview(n) for n
in xrange(1, 101)) )
```

USEFUL BUILT-IN FUNCTIONS: COUNTING AND ITERATING

len: returns the number of items of an enumerable object

```
len( [‘c’, ‘m’, ‘s’, ‘c’, 3, 2, 0] )
```

```
7
```

range: returns an iterable object

```
list( range(10) )
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

enumerate: returns iterable tuple (index, element) of a list

```
enumerate( [“311”, “320”, “330”] )
```

```
[(0, “311”), (1, “320”), (2, “330”)]
```

https://docs.python.org/3/library/functions.html

USEFUL BUILT-IN FUNCTIONS: MAP AND FILTER

`map`: apply a function to a sequence or iterable

```
arr = [1, 2, 3, 4, 5]
map(lambda x: x**2, arr)
```

```
[1, 4, 9, 16, 25]
```

`filter`: returns a list* of elements for which a predicate is true

```
arr = [1, 2, 3, 4, 5, 6, 7]
filter(lambda x: x % 2 == 0, arr)
```

```
[2, 4, 6]
```

We'll go over in much greater depth with pandas/numpy.

**in Python 3, returns Iterable*

PYTHONIC PROGRAMMING

Basic iteration over an array in Java:

```
int[] arr = new int[10];
for(int idx=0; idx<arr.length; ++idx) {
    System.out.println( arr[idx] );
}
```

Direct translation into Python:

```
idx = 0
while idx < len(arr):
    print( arr[idx] ); idx += 1
```

A more “Pythonic” way of iterating:

```
for element in arr:
    print( element )
```

LIST COMPREHENSIONS

Construct sets like a mathematician!

- $P = \{ 1, 2, 4, 8, 16, \ldots, 2^{16} \}$
- $E = \{ x \mid x \text{ in } \mathbb{N} \text{ and } x \text{ is odd and } x < 1000 \}$

Construct lists like a mathematician who codes!

```
P = [ 2**x for x in range(17) ]
```

```
E = [ x for x in range(1000) if x % 2 != 0 ]
```

Very similar to `map`, but:

- **You'll see these way more than `map` in the wild**
- **Many people consider `map`/`filter` not "pythonic"**
- **They can perform differently (`map` is "lazier")**

EXCEPTIONS

Syntactically correct statement throws an exception:

- `tweepy` (Python Twitter API) returns “Rate limit exceeded”
- sqlite (a file-based database) returns `IntegrityError`

```
print('Python', python_version())

try:
   cause_a_NameError
except NameError as err:
   print(err, '-> some extra text')
```

PYTHON 2 VS 3

Python 3 is intentionally backwards incompatible

- (But not *that* incompatible)

Biggest changes that matter for us:

- `print “statement”` → `print(“function”)`
- `1/2 = 0` → `1/2 = 0.5` and `1//2 = 0`
- ASCII `str` default → default Unicode

Namespace ambiguity fixed:

```
i = 1
[i for i in range(5)]
print(i)   # ????????
```

Python 2: prints “4”; Python 3: prints “1” (narrow scope)

TO ANY CURMUDGEONS …

If you're going to use Python 2 anyway, use the `_future_` module:

- Python 3 introduces features that will throw runtime errors in Python 2 (e.g., `with` statements)
- `_future_` module incrementally brings 3 functionality into 2
- https://docs.python.org/2/library/__future__.html

```
from _future_ import division
from _future_ import print_function
from _future_ import please_just_use_python_3
```

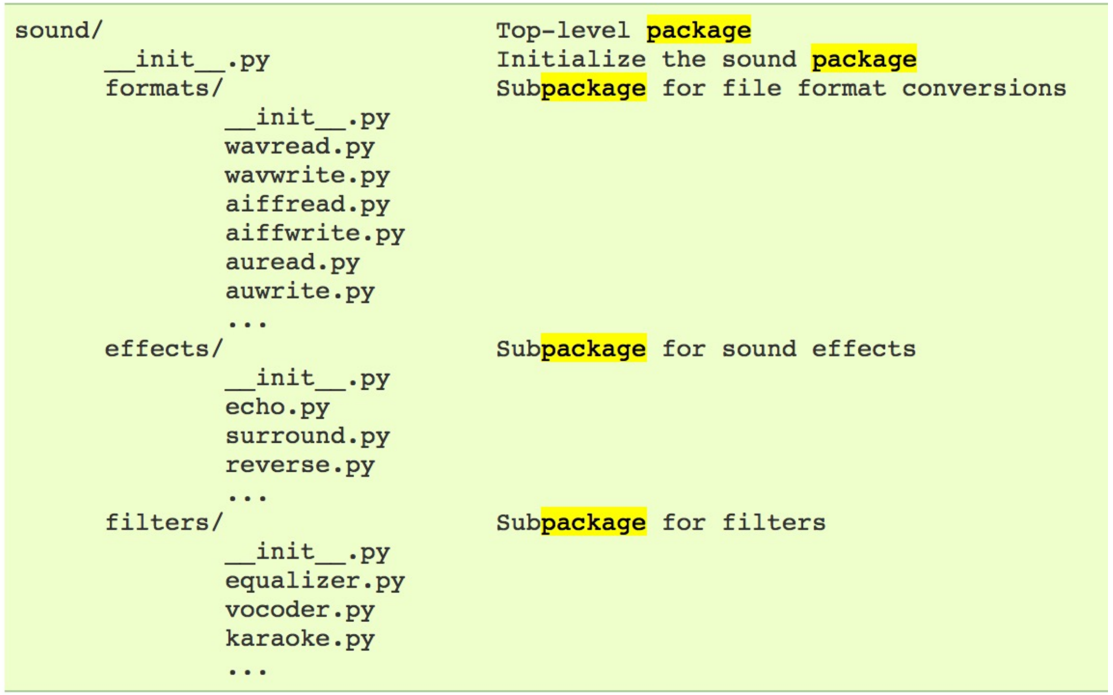
SO, HOW DOES IMPORT WORK?

Python code is stored in module – simply put, a file full of Python code

A package is a directory (tree) full of modules that also contains a file called `__init.py__`

- Packages let you structure Python’s module namespace
- E.g., `X.Y` is a submodule `Y` in a package named `X`

For one module to gain access to code in another module, it must import it



```
sound/                          Top-level package
      __init__.py               Initialize the sound package
      formats/                  Subpackage for file format conversions
              __init__.py
              wavread.py
              wavwrite.py
              aiffread.py
              aiffwrite.py
              auread.py
              auwrite.py
              ...
      effects/                  Subpackage for sound effects
              __init__.py
              echo.py
              surround.py
              reverse.py
              ...
      filters/                  Subpackage for filters
              __init__.py
              equalizer.py
              vocoder.py
              karaoke.py
              ...
```

```
# Load (sub)module sound.effects.echo
import sound.effects.echo
# Must use full name to reference echo functions
sound.effects.echo.echofilter(input, output, delay=0.7)
```

https://docs.python.org/2/tutorial/modules.html

EXAMPLE

```
# Load (sub)module sound.effects.echo
import sound.effects.echo
# Must use full name to reference echo functions
sound.effects.echo.echofilter(input, output, delay=0.7)
```

```
# Load (sub)module sound.effects.echo
from sound.effects import echo
# No longer need the package prefix for functions in echo
echo.echofilter(input, output, delay=0.7)
```

```
# Load a specific function directly
from sound.effects.echo import echofilter
# Can now use that function with no prefix
echofilter(input, output, delay=0.7)
```

https://docs.python.org/2/tutorial/modules.html

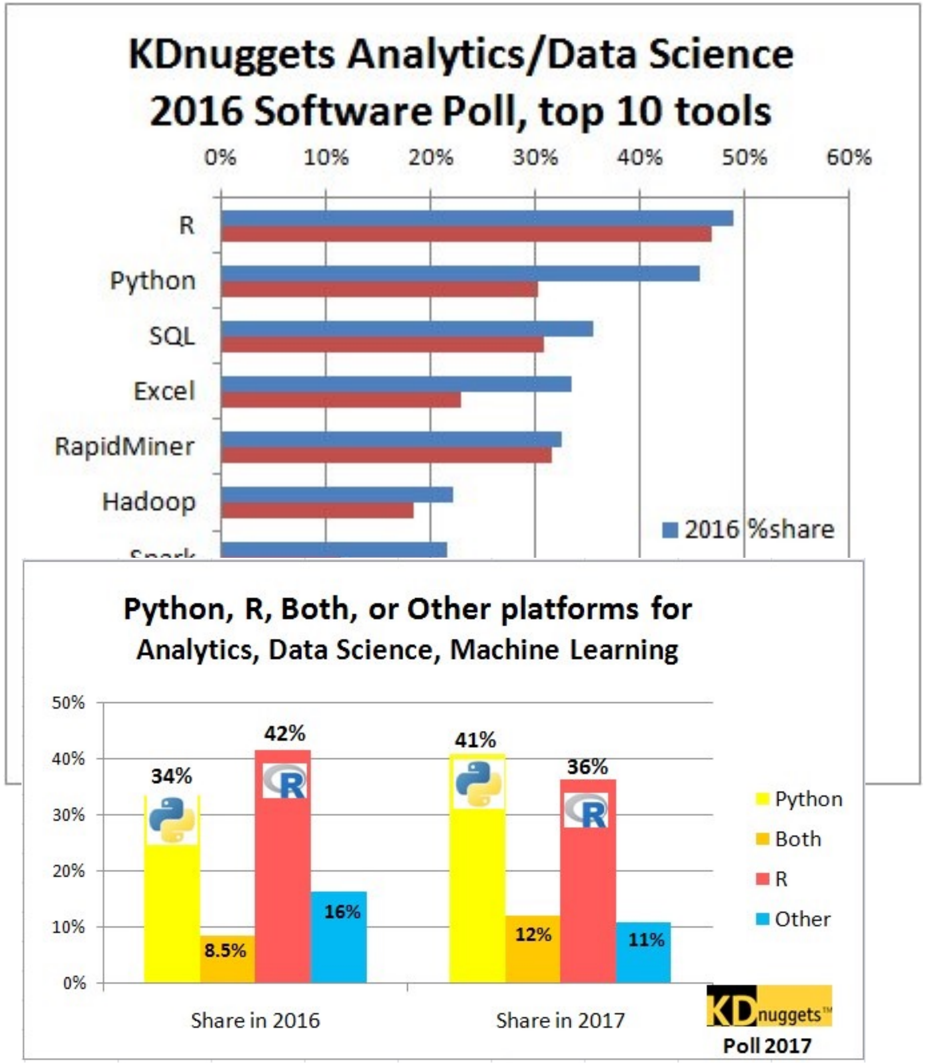
PYTHON VS R (FOR DATA SCIENTISTS)

There is no right answer here!

- Python is a “full” programming language – easier to integrate with systems in the field
- R has a more mature set of pure stats libraries …
- … but Python is catching up quickly …
- … and is already ahead specifically for ML.

You will see Python more in the tech industry.

- https://insights.stackoverflow.com/survey/2021



EXTRA RESOURCES

Plenty of tutorials on the web:

- https://www.learnpython.org/

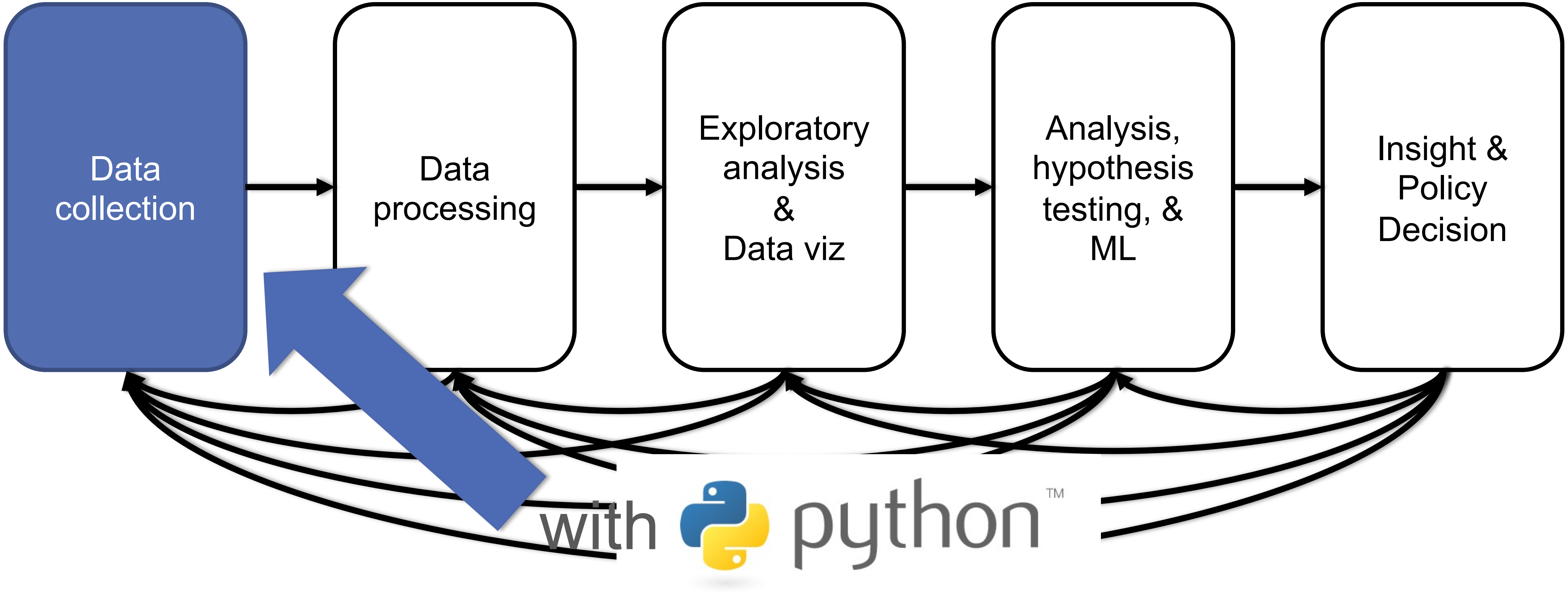
Work through Project 0, which will take you through some baby steps with Python and the Pandas library:

- (We’ll also post some more readings soon.)

Come (virtually!) hang out at office hours:

- All office hours will be on the website/Piazza by early next week.
- Will have coverage MTWThF.

TODAY'S LECTURE



Data collection

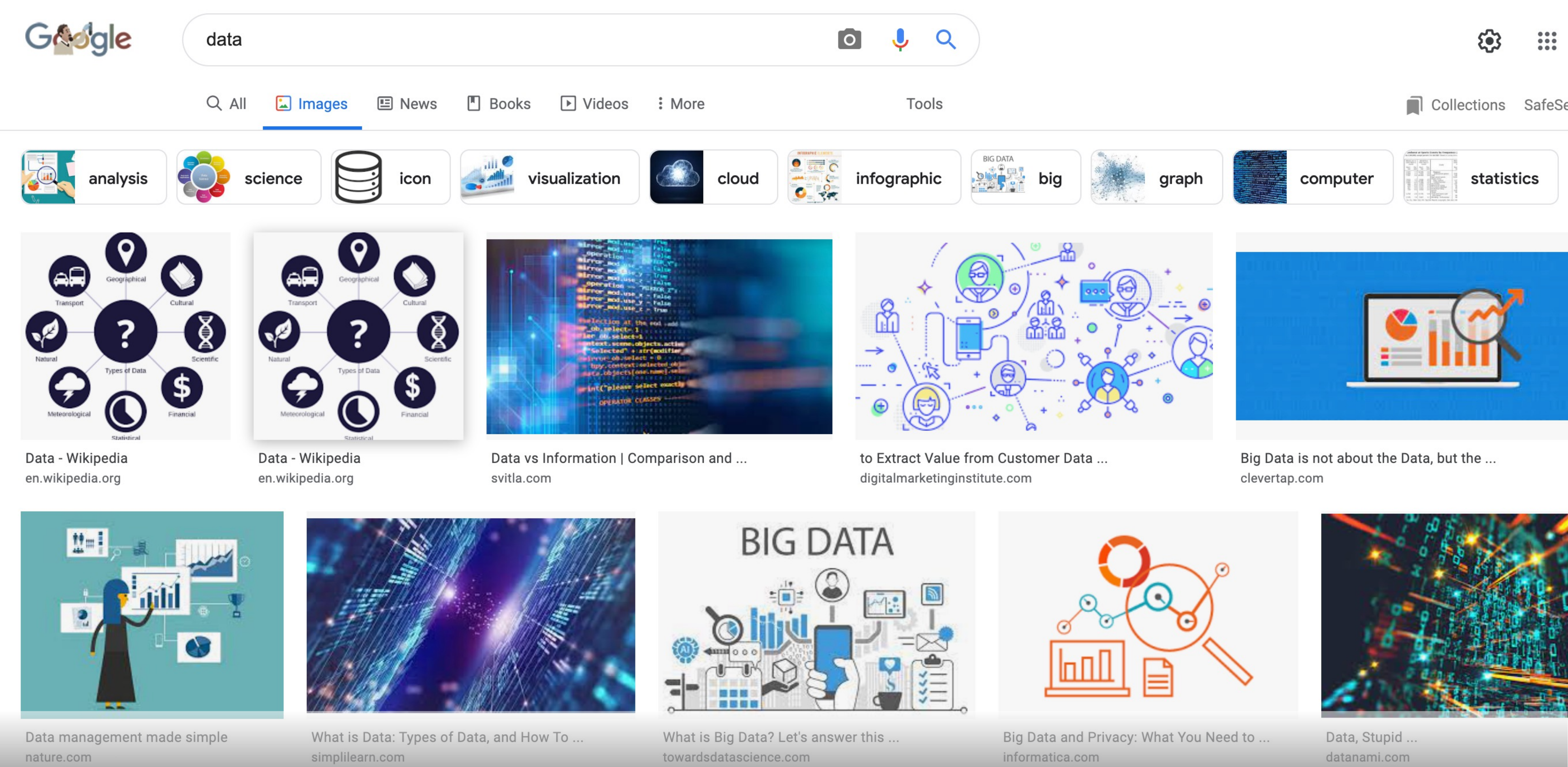
Data processing

Exploratory analysis & Data viz

Analysis, hypothesis testing, & ML

Insight & Policy Decision

Thanks: Zico Kolter's 15-388, Amol Deshpande, Nick Mattei



WHAT IS THIS “DATA”?

TABULAR DATA

Quick teaser. We'll go into greater depth when discussing **tidy data.**

Data is an abstraction of some real world entity.

- Also called: instance, example, record, object, case, individual.

Each of these entities is described by a set of features.

- Sometimes called variables, features, attributes, …

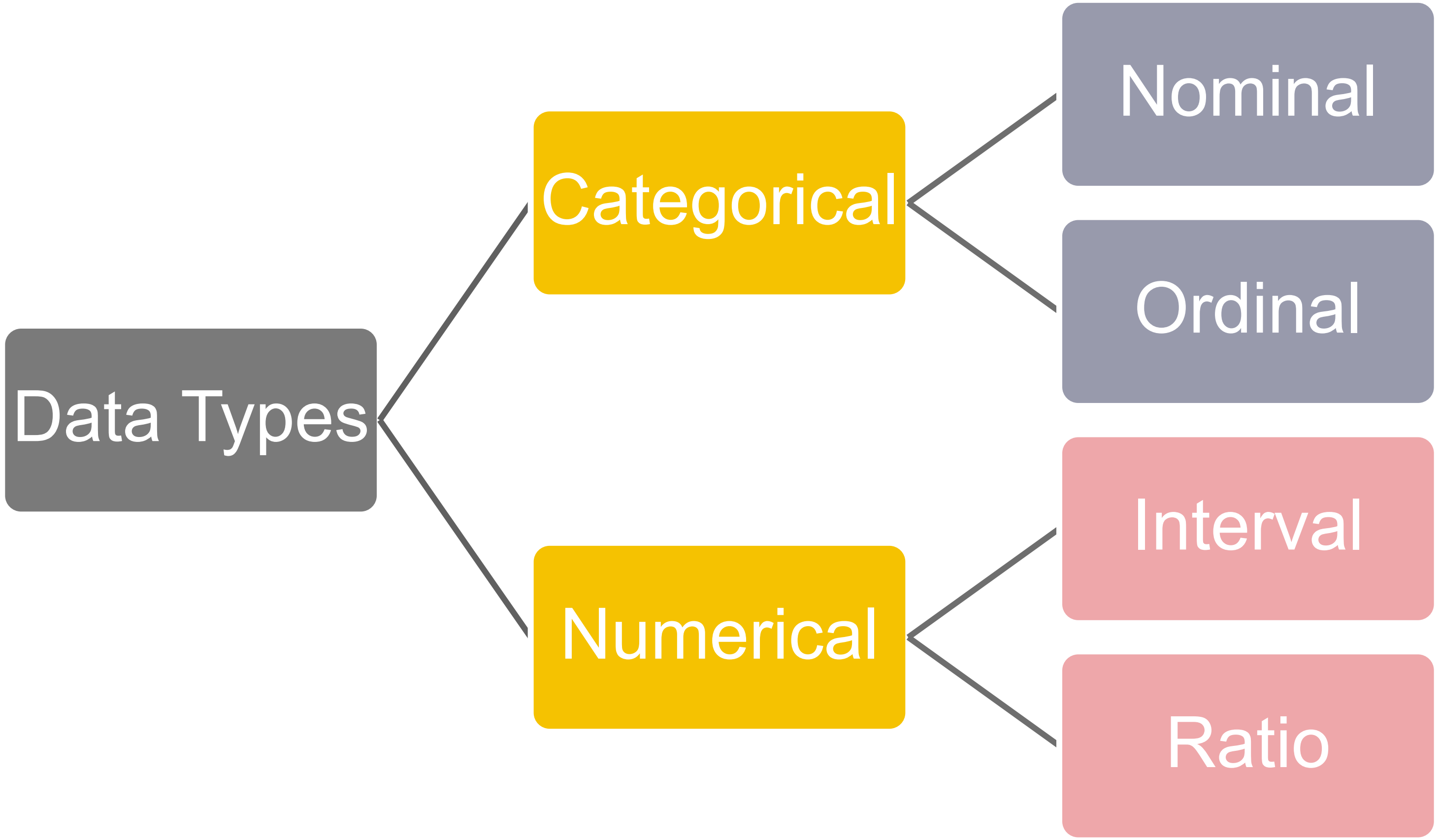
Can be processed into an *n* (number of entities) by *m* (number of attributes) matrix.

- Result of merging & processing different records!
- Picking the data that goes into this table has both technical and ethical concerns (recall: Target, Netflix, AOL examples)

ID	Title	Author	Year	Cover	Edition	Price
1	Emma	Austen	1815	Paper	20th	$5.75
2	Dracula	Stoker	1897	Hard	15th	$12.00
3	Ivanhoe	Scott	1820	Hard	8th	$25.00
4	Kidnapped	Stevenson	1886	Paper	11th	$5.00

CLASSICAL STATISTICAL VIEW OF DATA

There are four classical types of data



Data Types

Categorical

Numerical

Nominal

Ordinal

Interval

Ratio

CATEGORICAL DATA: TAKES A VALUE FROM A FINITE SET

Nominal (aka Categorical) Data:

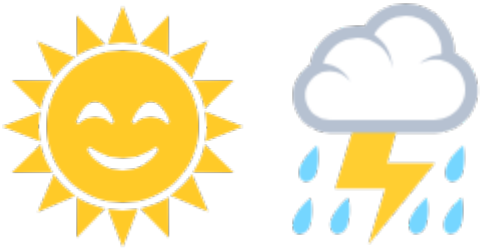
- Values have names: describe the categories, classes, or states of things
- Marital status, drink type, or some binary attribute
- Cannot compare easily, thus cannot naturally order them

Ordinal Data:

- Values have names: describe the categories, classes, or states of things
- However, there is an *ordering* over the values:
  - Strongly like, like, neutral, strongly dislike
- Lacks a mathematical notion of *distance* between the values

This distinction can be blurry…

- Is there an ordering over: sunny, overcast, rainy?



NUMERICAL DATA: MEASURED USING INTEGERS OR REALS

Interval Scale:

- Scale with fixed but arbitrary interval (e.g., dates)
- The difference between two values is *meaningful*:
  - Difference between 9/1/2019 and 10/1/2019 is the same as the difference between 9/1/2018 and 10/1/2018
- Can't compute ratios or scales: e.g., what unit is 9/1/2019 * 8/2/2020?

Ratio Scale:

- All the same properties as interval scale data, but the scale of measurement also possesses a **true-zero origin**
- Can look at the *ratio* of two quantities (unlike interval)
- E.g., zero money is an absolute, one money is half as much as two money, and so on

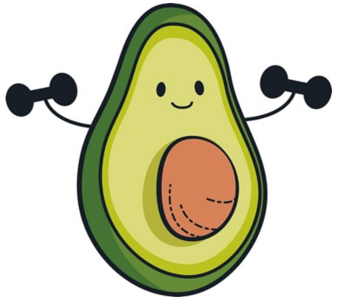
NUMERICAL DATA: EXAMPLES

Temperatures:

- Celsius / Fahrenheit: interval or ratio scale ???????????
  - **Interval:** 0C is not 0 heat, but is an arbitrary fixed point
  - Hence, we can’t say that 30F is twice as warm as 15F.
- Kelvin (K): interval or ratio scale ???????????
  - **Ratio:** 0K is assumed to mean zero heat, a true fixed point

Weight:

- Grams: interval or ratio scale ??????????
- **Ratio:** 0g served as fixed point, 4g is twice 2g, …



GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	?	?	?	?

Thanks to GraphPad

GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	Yes	Yes	Yes	Yes
median and percentiles	?	?	?	?

GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	Yes	Yes	Yes	Yes
median and percentiles	No	Yes	Yes	Yes
addition or subtraction	?	?	?	?

GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	Yes	Yes	Yes	Yes
median and percentiles	No	Yes	Yes	Yes
addition or subtraction	No	No	Yes	Yes
mean or standard deviation	?	?	?	?

GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	Yes	Yes	Yes	Yes
median and percentiles	No	Yes	Yes	Yes
addition or subtraction	No	No	Yes	Yes
mean or standard deviation	No	No	Yes	Yes
ratio, or coefficient of variation	?	?	?	?

GENERAL RULES

OK to compute....	Nominal	Ordinal	Interval	Ratio
frequency distribution	Yes	Yes	Yes	Yes
median and percentiles	No	Yes	Yes	Yes
addition or subtraction	No	No	Yes	Yes
mean or standard deviation	No	No	Yes	Yes
ratio, or coefficient of variation	No	No	No	Yes

DATA MANIPULATION AND COMPUTATION

Data Science == manipulating and computing on data

Large to very large, but somewhat “structured” data

We will see several tools for doing that this semester

Thousands more out there that we won’t cover

Need to learn to shift thinking from:

Imperative code to manipulate data structures

to:

Sequences/pipelines of operations on data

Should still know how to implement the operations themselves, especially for debugging performance (covered in classes like 420, 424), but we won’t cover that much

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data

One-dimensional Arrays, Vectors

0.1	2	3.2	6.5	3.4	4.1

“data”	”representation”	”i.e.”

Indexing
Slicing/subsetting
Filter
‘map’ → apply a function to every element
’reduce/aggregate’ → combine values to get a single scalar (e.g., sum, median)

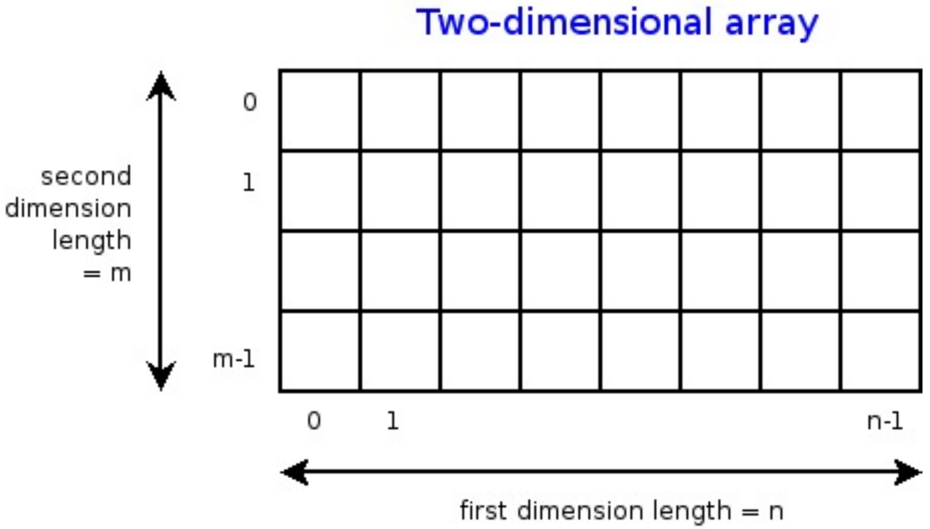
Given two vectors: **Dot and cross products**

2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data

n-dimensional arrays



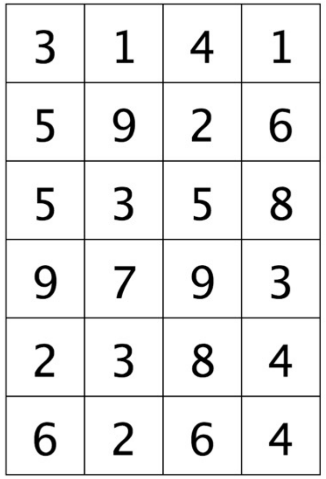
Indexing
Slicing/subsetting
Filter
'map' → apply a function to every element
'reduce/aggregate' → combine values across a row or a column (e.g., sum, average, median etc..)

2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

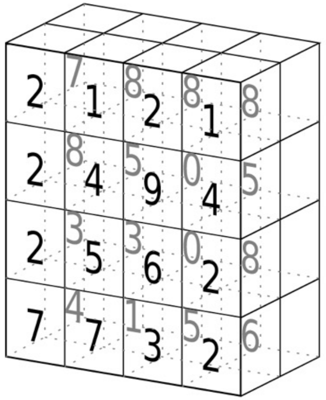
1. **Data Representation**, i.e., what is the natural way to think about given data

Matrices, Tensors



3	1	4	1
5	9	2	6
5	3	5	8
9	7	9	3
2	3	8	4
6	2	6	4

tensor of dimensions [6,4]
(matrix 6 by 4)



tensor of dimensions [4,4,2]

n-dimensional array operations

+

Linear Algebra

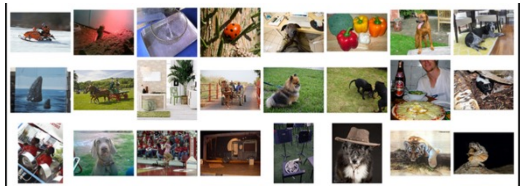
- **Matrix/tensor multiplication**
- **Transpose**
- **Matrix-vector multiplication**
- **Matrix factorization**

2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data

Sets: of Objects



Sets: of (Key, Value Pairs)

(amol@cs.umd.edu,(email1, email2,…))

(john@cs.umd.edu,(email3, email4,…))

Filter
Map
Union

Reduce/Aggregate

Given two sets, **Combine/Join** using "keys"

Group and then aggregate

2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data

Filter rows or columns

”Join” two or more relations

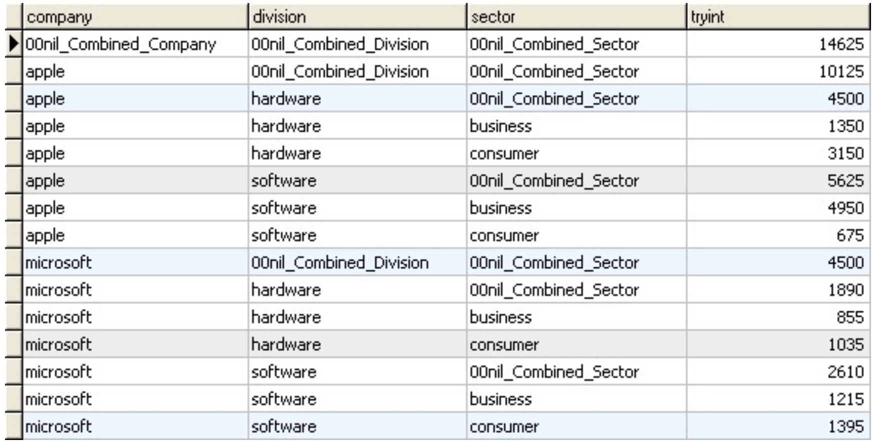
”Group” and “aggregate” them

Relational Algebra formalizes some of them

Structured Query Language (SQL)

Many other languages and constructs, that look very similar

Tables/Relations == Sets of Tuples



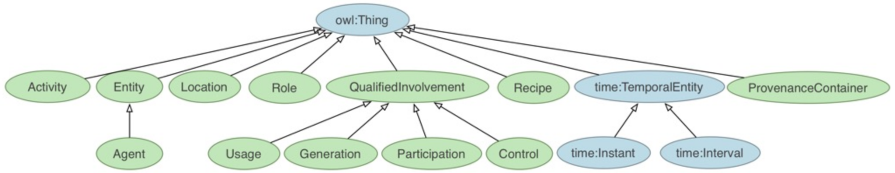
company	division	sector	tryint
00nil_Combined_Company	00nil_Combined_Division	00nil_Combined_Sector	14625
apple	00nil_Combined_Division	00nil_Combined_Sector	10125
apple	hardware	00nil_Combined_Sector	4500
apple	hardware	business	1350
apple	hardware	consumer	3150
apple	software	00nil_Combined_Sector	5625
apple	software	business	4950
apple	software	consumer	675
microsoft	00nil_Combined_Division	00nil_Combined_Sector	4500
microsoft	hardware	00nil_Combined_Sector	1890
microsoft	hardware	business	855
microsoft	hardware	consumer	1035
microsoft	software	00nil_Combined_Sector	2610
microsoft	software	business	1215
microsoft	software	consumer	1395

2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data

Hierarchies/Trees/Graphs



“Path” queries

Graph Algorithms and Transformations

Network Science

Somewhat more ad hoc and special-purpose

Changing in recent years

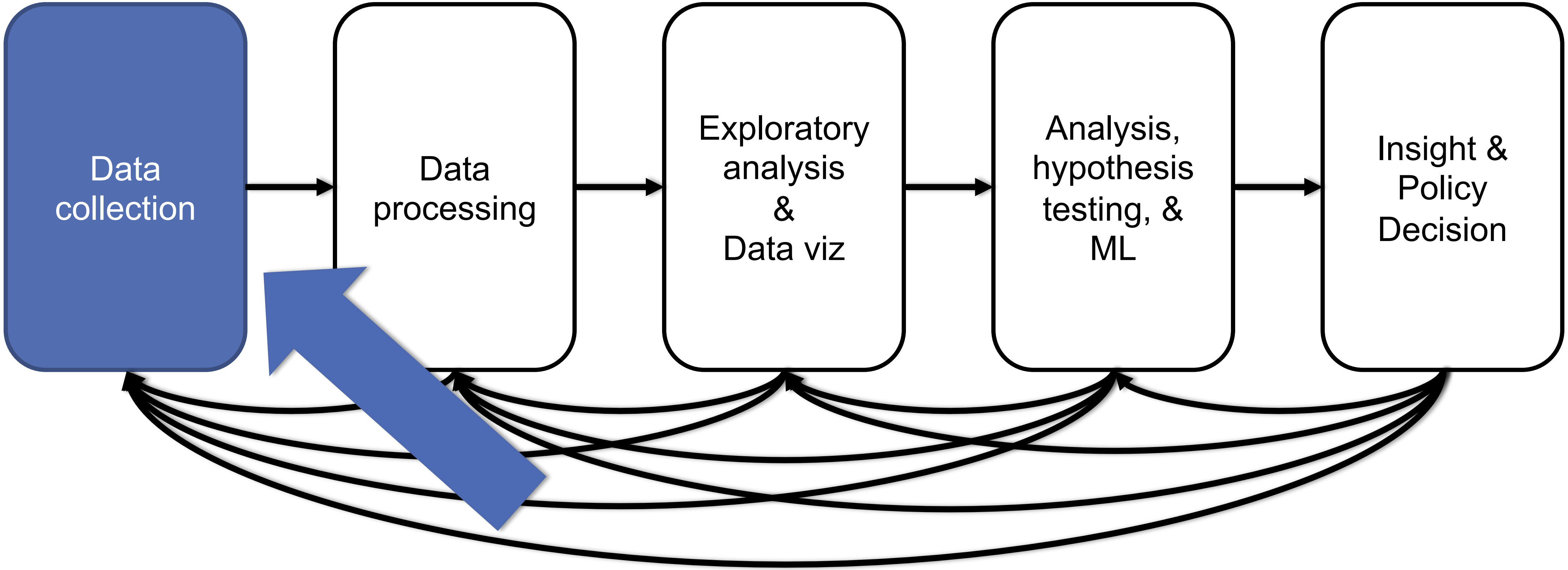
2. **Data Processing Operations**, which take one or more datasets as input and produce one or more datasets as output

DATA MANIPULATION AND COMPUTATION

1. **Data Representation**, i.e., what is the natural way to think about given data
2. **Data Processing Operations**, which take one or more datasets as input and produce

- **Why?**
  - Allows one to think at a higher level of abstraction, leading to simpler and easier-to-understand scripts
  - Provides ”independence” between the abstract operations and concrete implementation
  - Can switch from one implementation to another easily
- **For performance debugging, useful to know how they are implemented and rough characteristics**

NEXT LECTURE



Data collection

Data processing

Exploratory analysis & Data viz

Analysis, hypothesis testing, & ML

Insight & Policy Decision

… on to the “collection” part of things …