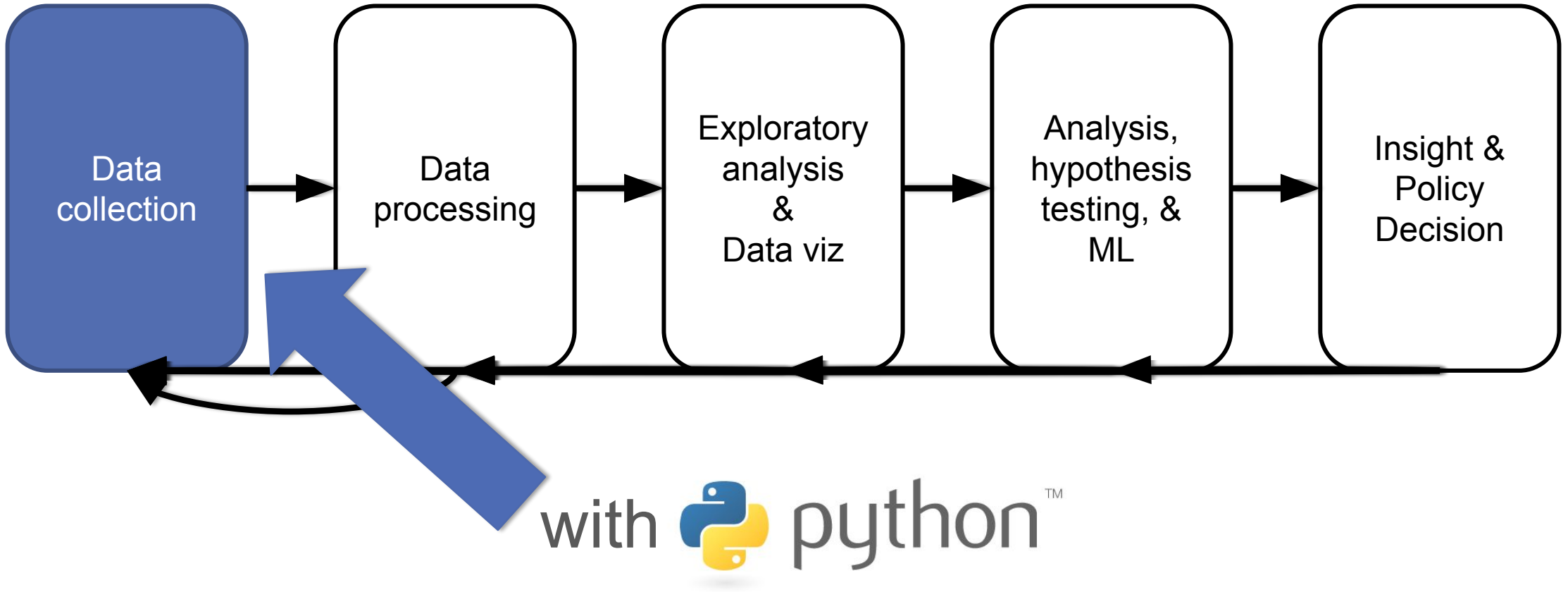
TODAY'S LECTURE



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

data

All | Images | News | Books | Videos | More | Tools | Collections | SafeSe

analysis | science | icon | visualization | cloud | infographic | big | graph | computer | statistics

Data - Wikipedia
en.wikipedia.org

Data - Wikipedia
en.wikipedia.org

Data vs Information | Comparison and ...
svitla.com

to Extract Value from Customer Data ...
digitalmarketinginstitute.com

Big Data is not about the Data, but the ...
clevertap.com

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, …

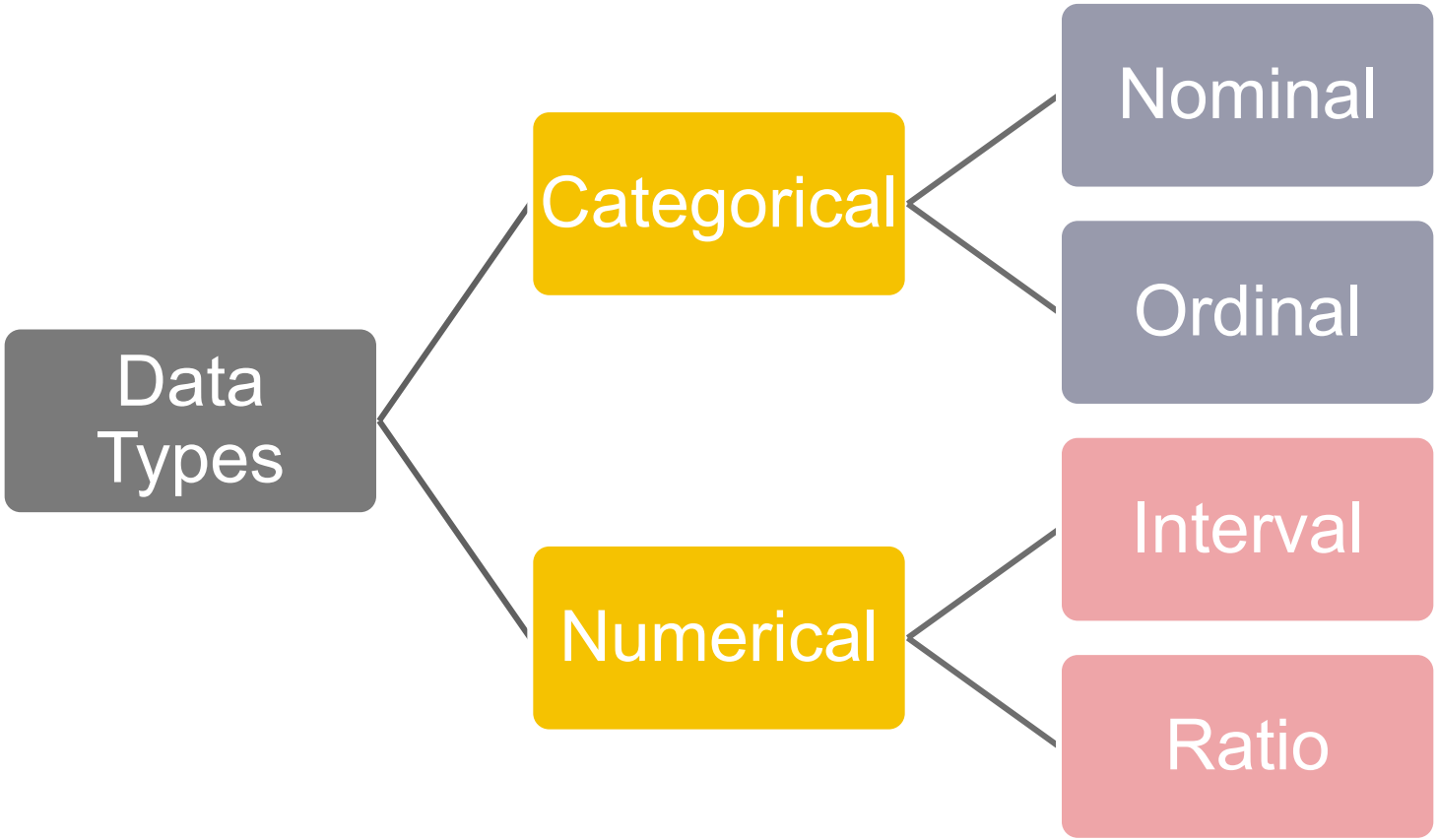
Can be processed into an *n* (number of entities) by *m* (number of attributes) matrix.

- 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



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

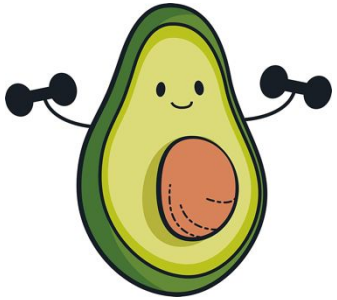
NUMERICAL DATA: EXAMPLES

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.”

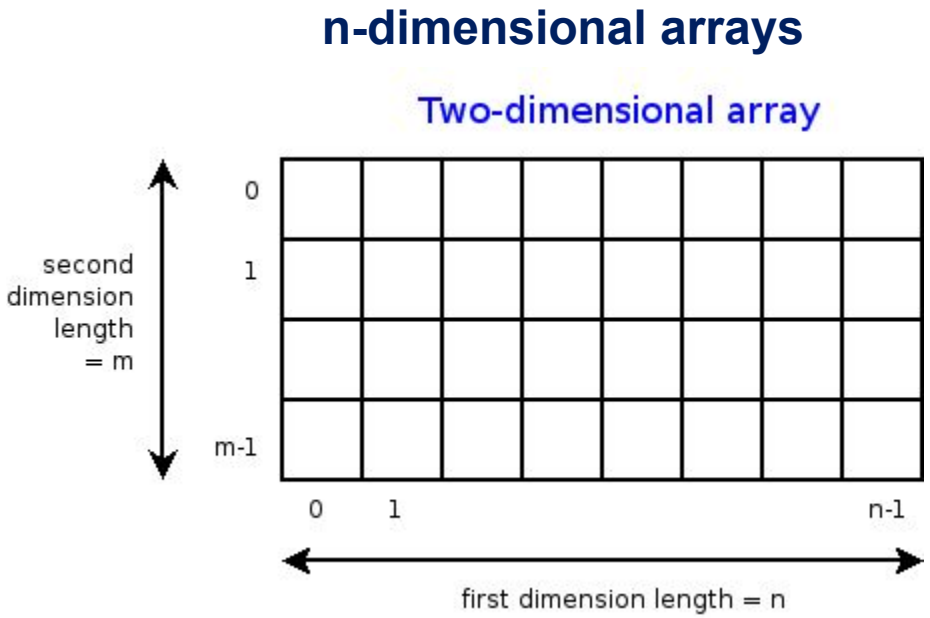
Indexing
Slicing/subsetting
Filter
‘map’ □ apply a function to every element
’reduce/aggregate’ □ combine values to get a single scalar (e.g., sum, median)

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



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

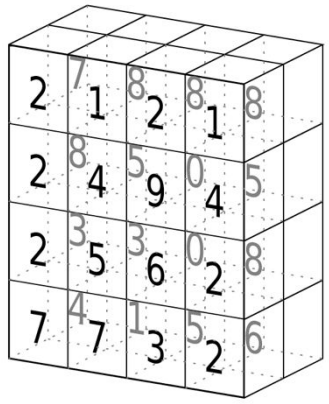
DATA MANIPULATION AND COMPUTATION

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

Filter
Map
Union

Reduce/Aggregate

Sets: of (Key, Value Pairs)

(amol@cs.umd.edu,(email1, email2,…))

(john@cs.umd.edu,(email3, email4,…))

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

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

Filter rows or columns

”Join” two or more relations

”Group” and “aggregate” them

Relational Algebra formalizes some of them

Structured Query Language (SQL)

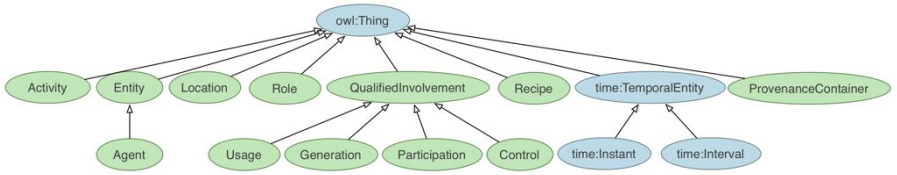
Many other languages and constructs, that look very similar

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

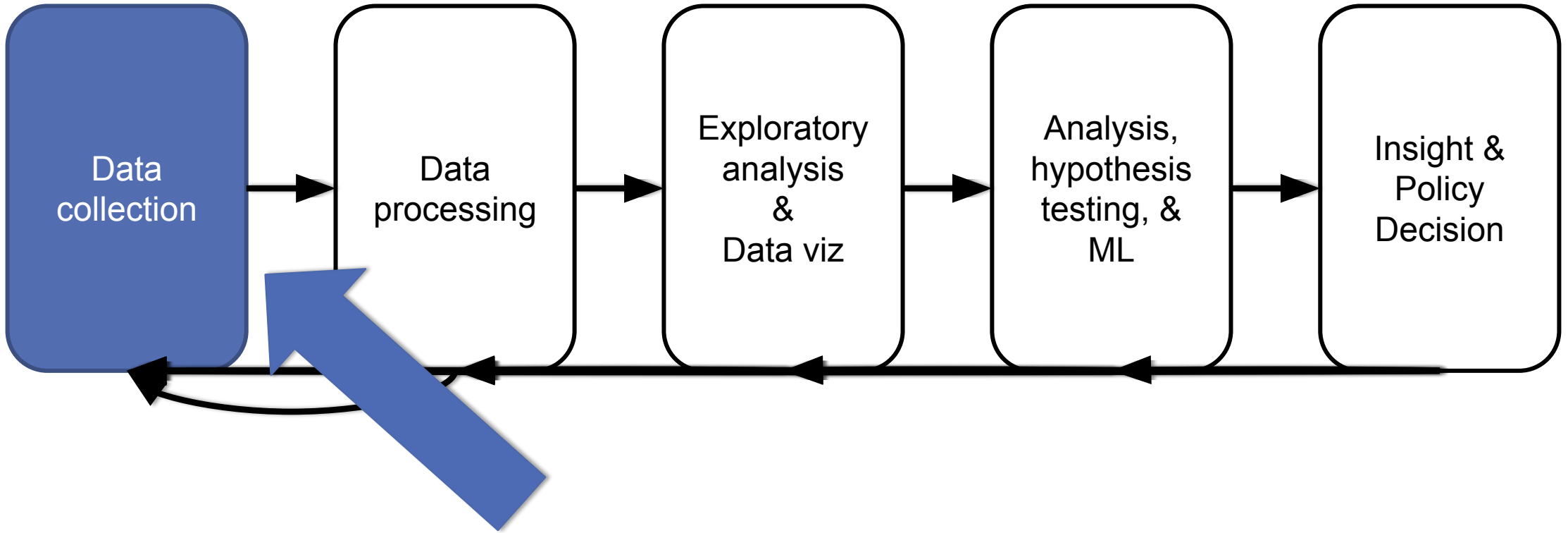
DATA MANIPULATION AND COMPUTATION

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**

THE REST OF TODAY'S LECTURE



… on to the "collection" part of things …

GOTTA CATCH ‘EM ALL

Five ways to get data:

- **Direct download and load from local storage**
- **Generate locally via downloaded code (e.g., simulation)**
- **Query data from a database (covered in a few lectures)**
- **Query an API from the intra/internet**
- **Scrape data from a webpage**

} Covered today.

WHEREFORE ART THOU, API?

A web-based Application Programming Interface (API) like we'll be using in this class is a contract between a server and a user stating:

"If you send me a specific request, I will return some information in a structured and documented format."

(More generally, APIs can also perform actions, may not be web-based, be a set of protocols for communicating between processes, between an application and an OS, etc.)

“SEND ME A SPECIFIC REQUEST”

Most web API queries we’ll be doing will use HTTP requests:

- `conda install -c anaconda requests=2.12.4`

```
r = requests.get(   'https://api.github.com/user',
                    auth=('user', 'pass')   )
```

```
r.status_code
```

```
200
```

```
r.headers['content-type']
```

```
'application/json; charset=utf8'
```

```
r.json()
```

```
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

http://docs.python-requests.org/en/master/

HTTP REQUESTS

https://www.google.com/**?q=cmsc320&tbs=qdr:m**

??????????

HTTP GET Request:

GET /?q=cmsc320&tbs=qdr:m HTTP/1.1
Host: www.google.com
User-Agent: **Mozilla/5.0 (X11; Linux x86_64; rv:10.0.1) Gecko/20100101 Firefox/10.0.1**

```
params = { "q": "cmsc320", "tbs": "qdr:m" }
r = requests.get(    "https://www.google.com",
                     params = params )
```

*be careful with https:// calls; `requests` will not verify SSL by default

RESTFUL APIS

This class will just query web APIs, but full web APIs typically allow more.

Representational State Transfer (RESTful) APIs:

- GET: perform query, return data
- POST: create a new entry or object
- PUT: update an existing entry or object
- DELETE: delete an existing entry or object

Can be more intricate, but verbs (“put”) align with actions