UNIVERSITY OF MARYLAND

INST447 -0101
Fall 2020
Lecture 1

Location: Virtual

Instructor: Bill Farmer

TA: Jonathan Chen

Grader: Jeffrey Chen

1

Today

Great day to start a new semester

2

Introductions

Who am I?

Who are you?

3

Goals of the class

4

Syllabus

Review of syllablus

5

Next Class

Today

New Semester

Fall 2020

Introductions

Syllabus

Right Class

INST447-0101

Examines approaches to locating, acquiring, manipulating, and disseminating data. Imperfection, biases, and other problems in data are examined, and methods for identifying and correcting such problems are introduced. The course covers other topics such as automated collection of large data sets, and extracting, transforming, and reformatting a variety of data and file types.

If you are tired, stand up in the back of class, or during COVID just stand up wherever you are.

COLLEGE OF INFORMATION STUDIES

Introductions

Who am I?

Bill Farmer

- Vice President of Engineering -
  - Clarity Business Solutions, Inc.
- MS Information Technology in Software, Carnegie Mellon University
- BS Computer Science, University of Pittsburgh
- 21 years Software Engineering experience
- Maryland Data Works Meetup
- Parallel tinkerer

UTA - Jonathan Chen

- Currently is a senior in Information Science
- email: jonnyapple985@gmail.com
- Office hours:
  - https://umd.webex.com/umd/j.php?MTID=maa654cf56a69519872456151b7d2c073

Grader - Jeffrey Chen

- iSchool Alumni - 2019
- Currently in second semester for the Master's in Information Systems program at the R.H. Smith School of Business
- jeffrey.chen@rhsmith.umd.edu

COLLEGE OF INFORMATION STUDIES

Goals of the class

What is this course?

"There's the joke that 80 percent of data science is cleaning the data and 20 percent is complaining about cleaning the data."

- Kaggle founder Anthony Goldbloom

What is meant by “data cleaning”?

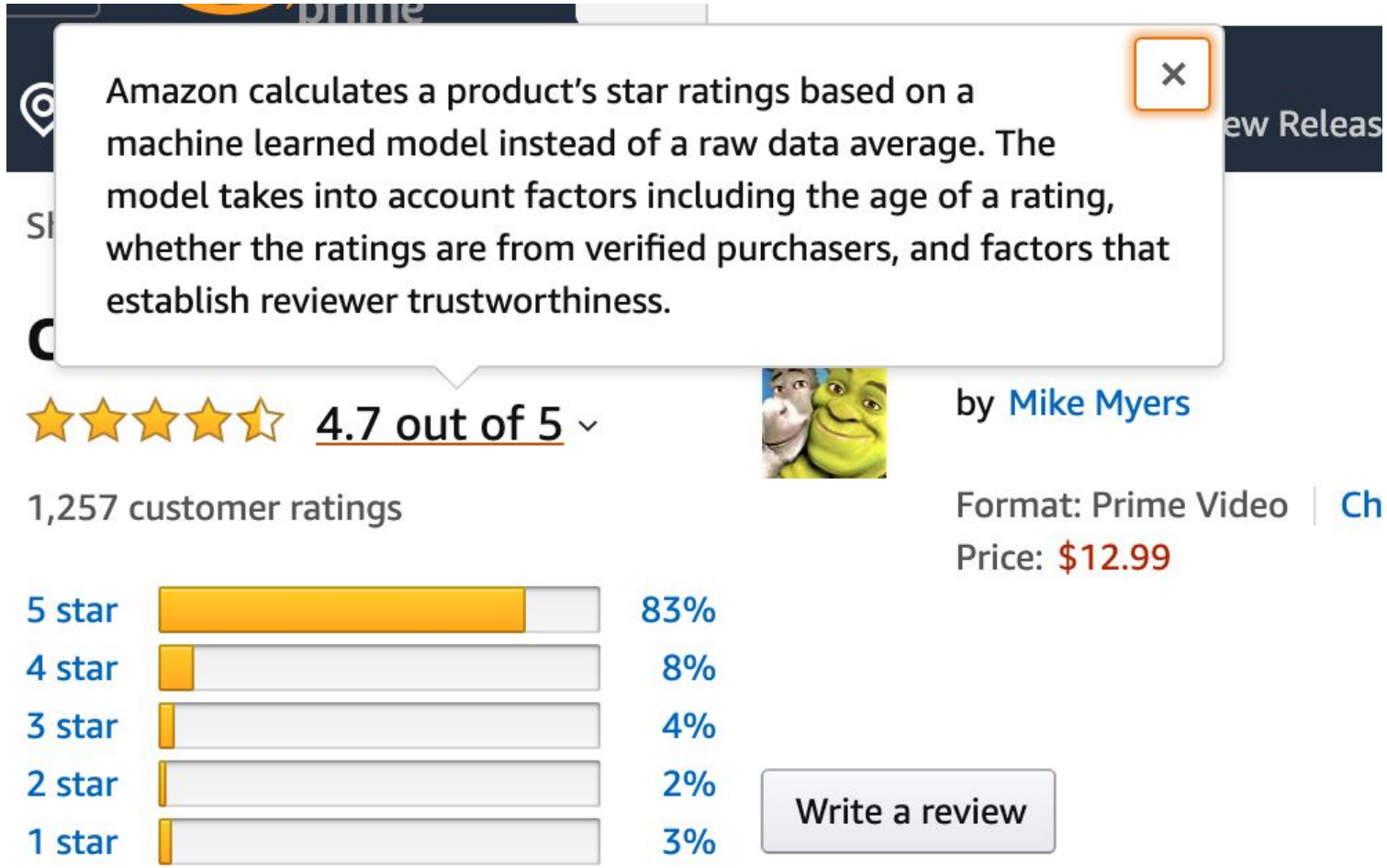
- Data Ingestion & Storage
  - Collecting data
    - web scraping
    - using APIs
- Converting data from sub-optimal format to an easier one to work with
  - json -> SQL
  - html -> json
  - json files that differ in format -> Normalized json format

What is meant by “data cleaning”?

- Data Cleaning and Validation
  - Inspecting data to remove or handle issues
    - identifying missing values
    - identifying outliers
- Standardizing data
  - fixing dates to be of same format “2018-09-01”

What does it mean if you don’t rate a movie?

- Understanding missing ratings
  - Movies you’ve never watched and have no opinion about
  - Movies you do not want to watch and have a negative opinion about



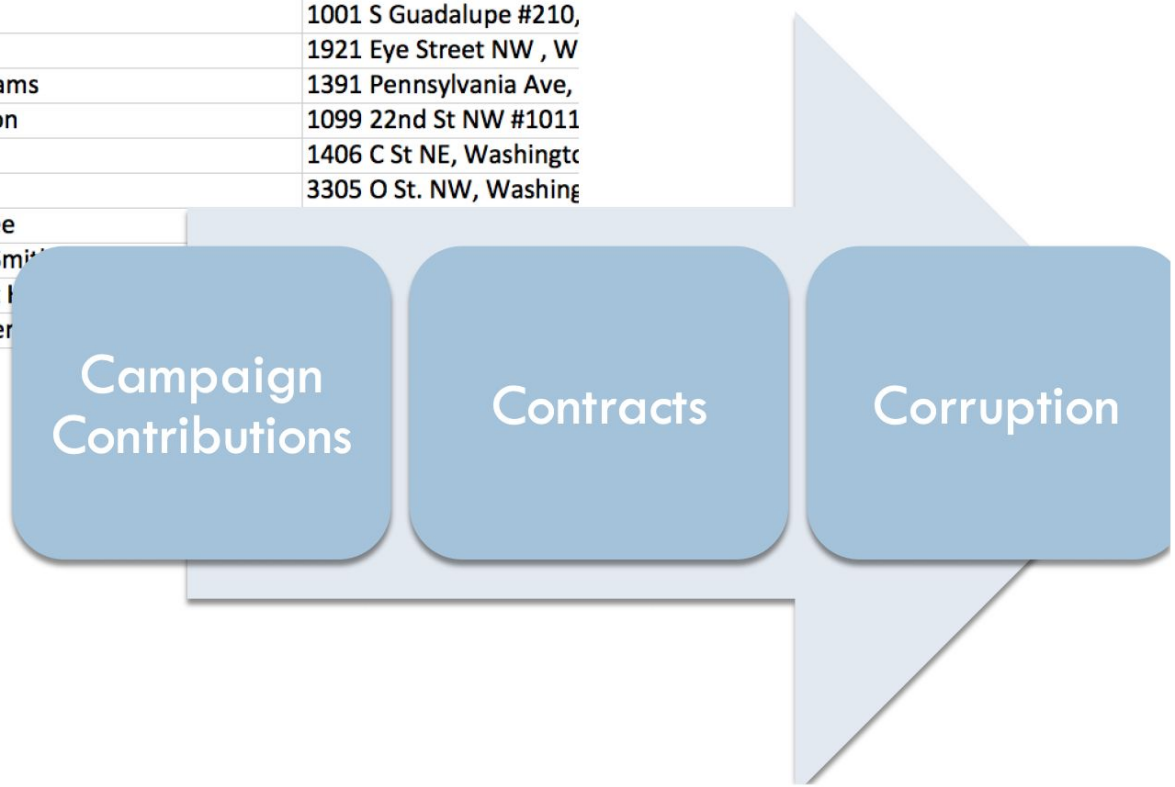
COLLEGE OF INFORMATION STUDIES

What is meant by “data cleaning”?

- Data Wrangling & Processing:
  - Combining data sets
    - Combining weather data with traffic data
  - Aggregating data
    - Calculating spending by week instead by day
  - Reshaping data

Combining Data

	Candidate Name	Contributor Name	Address Raw	Contributo
1				
2	Ron Austin	AMAR GROUP LLC -- Remitted by Genell A	6230 3rd Street, NW Sui	
3	Drew Franklin	Adam Eidinger	1858 Mintwood Pl NW #	
4	Drew Franklin	Sara Callahan	1001 3rd St sw Apt 502	
5	Drew Franklin	Matt Kirkland	850 Quincy St NW #624,	
6	Drew Franklin	Regina Mirabito	1622 3rd St NW, Washir	
7	Drew Franklin	Kim Daniel	1001 S Guadalupe #210,	
8	David Garber	Jetties Inc.	1921 Eye Street NW , W	
9	LaRuby May	Kaneedreck Adams	1391 Pennsylvania Ave,	
10	Jack Jacobson	Florence Harmon	1099 22nd St NW #1011	
11	Jack Jacobson	Joey Weedon	1406 C St NE, Washingtc	
12	David Garber	Gavin Young	3305 O St. NW, Washing	
13	Yvette Alexander	Herman Foushee		
14		Windy Carson-Smi		
15	David Grosso	Premium Select I		
16	David Grosso	MD-DE-DC Bever		
17	David Grosso	Michael Raab		



COLLEGE OF INFORMATION STUDIES

COLLEGE OF INFORMATION STUDIES

Syllabus & Class Activities

Course Structure “Data Science Lab”

- At home
  - Readings (due before class)
  - Assignments
- In class
  - Short lecture
  - Work on “lab exercise”
  - Turn in “lab exercise” using ELMS

Syllabus

- Lectures
- Labs
- Assignments
- Group Projects
- Midterm and Final

Labs

- Focus on skills related to topic of week
  - e.g. regular expressions
- New data set(s) and research questions
- Work together in pairs in class
- Perform manipulations on data set(s) to answer research questions.
- Enter your answers in ELMS. Turn in *own* Jupyter notebook.

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

Group Project

- Groups 2-4 people
- Project proposal
- Project update
- Project presentation and report
- Four main criteria
  - Theme redevelopment in DC area
  - Centered on data using at least one technique from class
  - Must tell us something interesting
  - To get an A must go beyond what is covered in class

Exams

- Midterm
  - Thursday, October 15
- Final
  - TBD
- Structure - TBD

Next Class

Next Class

- Readings (due before next class)
  - See ELMS
  - You will be asked to summarize in class! Come prepared.
- Software install will be posted as a video Thursday

I’m looking forward to the rest of the semester and seeing what projects we choose to work on!

COLLEGE OF INFORMATION STUDIES

Supporting Slides

Who are you?

COLLEGE OF INFORMATION STUDIES

Install Software

Software Tools

- Python & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine
  - http://openrefine.org/download.html

Data Science Projects

Data science - the ability to take large amounts of data in many different formats and be able to understand it, to process it, to extract value from it, to summarize it, to visualize it, and to communicate it to others.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value. -datarobot.com

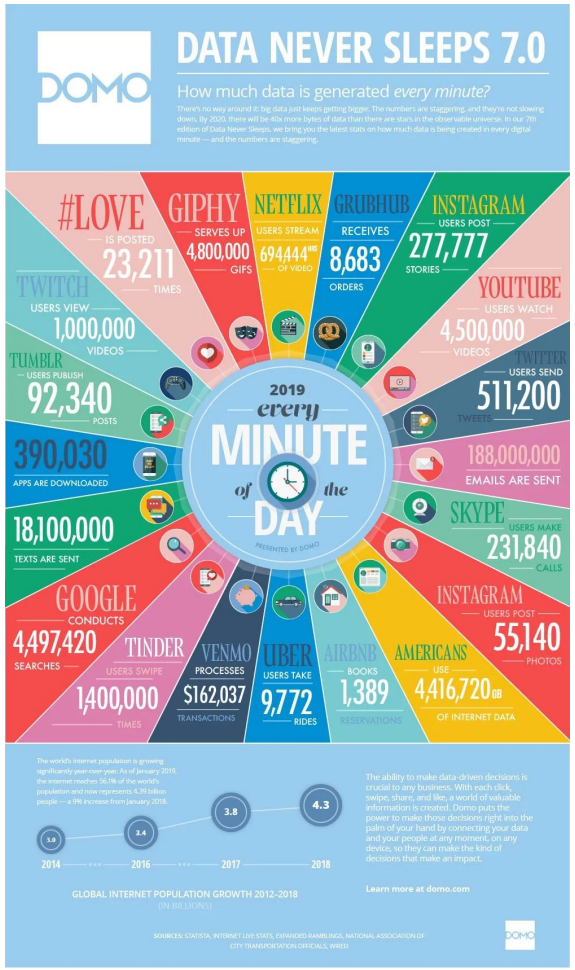
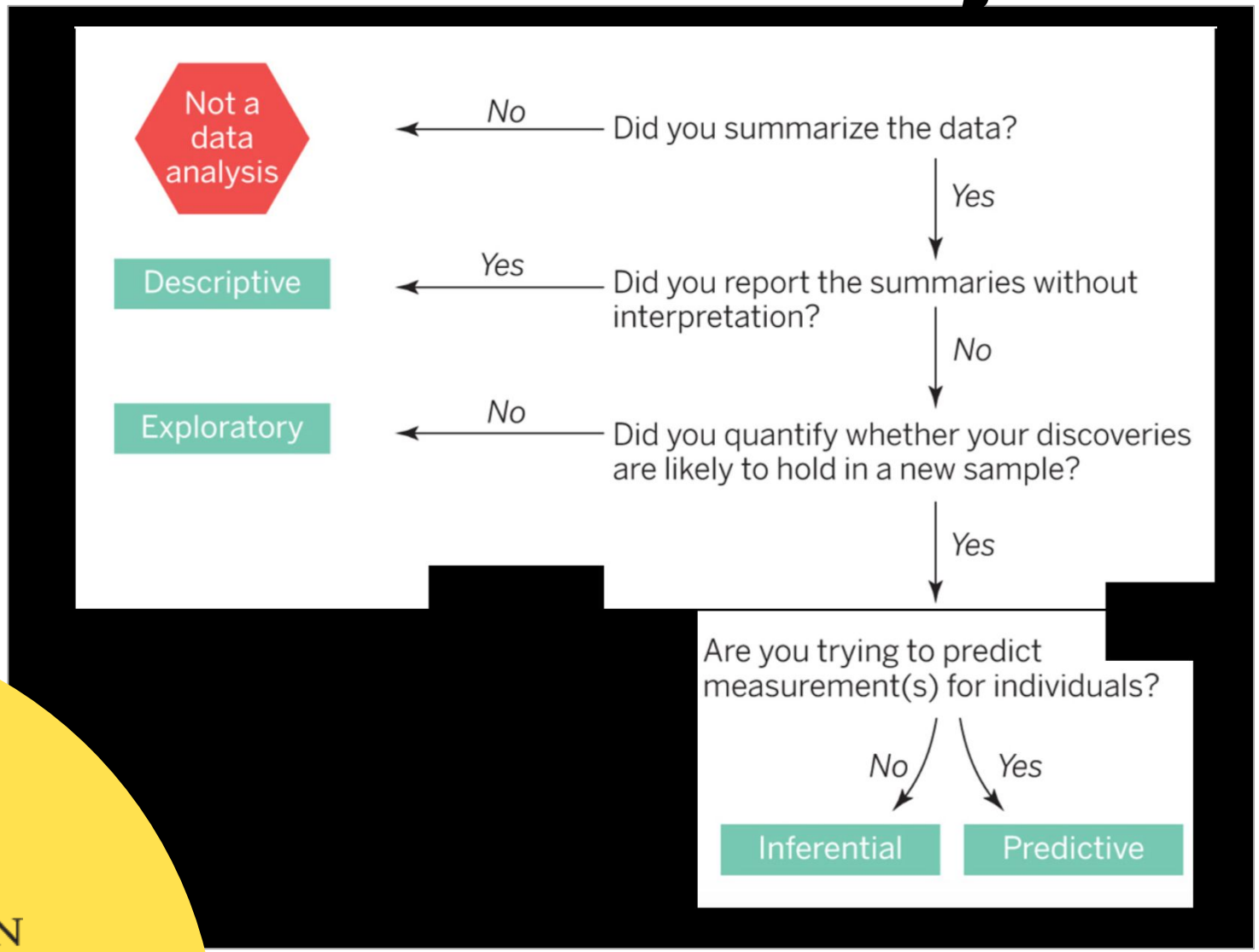


Illustration from: https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2019-version-infographic/558793/

Data Science Projects



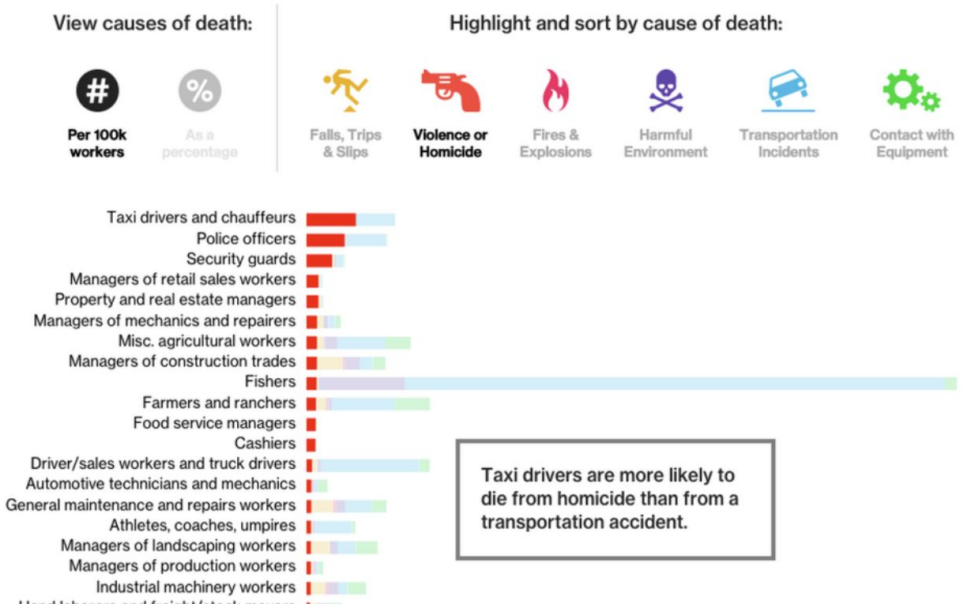
Data Science Project Outputs

Report - Summary of findings

- e.g. Blog
- e.g. Internal or Customer presentation
- e.g. Computation journalism article

Bloomberg: Most dangerous jobs

When it comes to building easy to understand, informative and interactive data vizzes; no one does it better than Bloomberg. Their data graphics section is full of highly interesting data vizzes that are actually fun to play with. A recent one that caught our eye is there analysis of the most dangerous jobs in America.



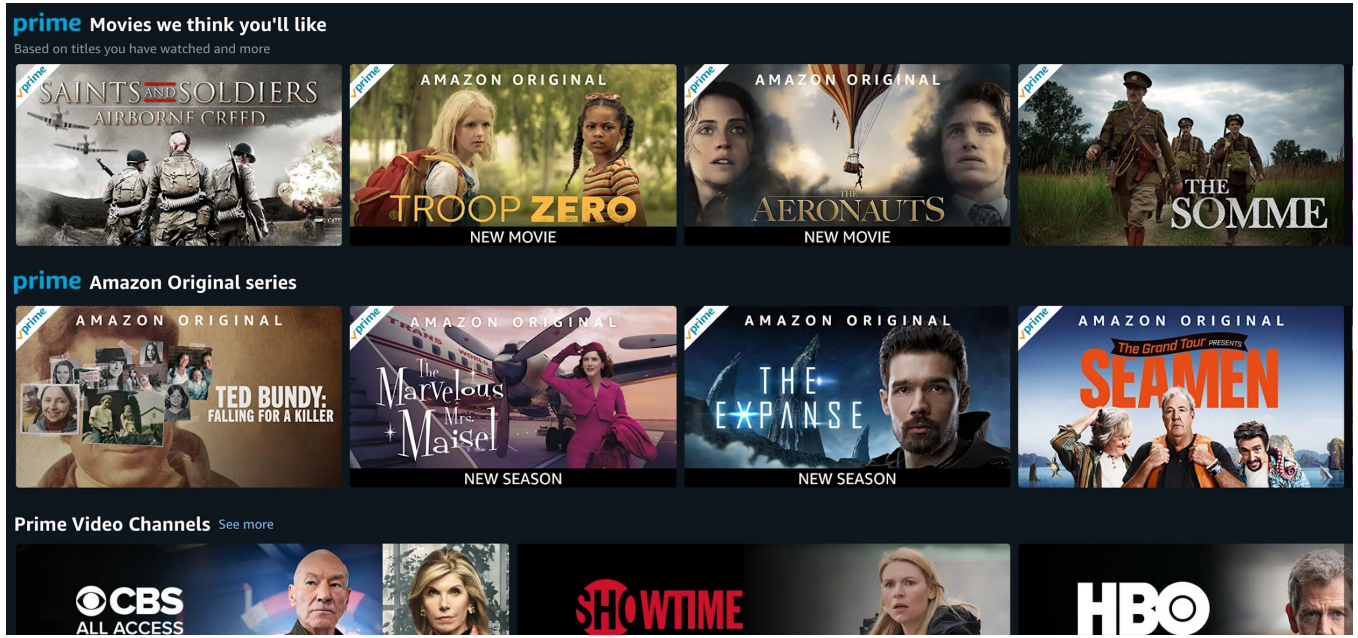
https://www.import.io/post/8-fantastic-examples-of-data-journalism/

COLLEGE OF INFORMATION STUDIES

Data Science Project Outputs

Data product - Tool or component that uses findings

- e.g. Algorithm
- e.g. Interactive visualization



01 Admin

02 Projects

03 Data Science Pipelines

04 Python and Jupyter Notebooks

05 Review Papers

06 Lab

07 Next Class

This Week

Time: Tuesday virtual

- Admin
  - Re-introduction
  - Class Structure
  - Office Hours
  - Syllabus Updates
- Projects
- Data science Pipelines
- Python and Jupyter Notebooks
- Simple Numpy and Pandas

Time: Thursday Virtual w/ optional live session

- Live session
  - Review of paper?
  - Qs about software install
- Lab
- Next Week
- Questions

If you are tired, stand up in the back of class.

Use of phone during class for non-class purposes is rude.

COLLEGE OF INFORMATION STUDIES

Admin

Re-Introductions

- Vice President of Engineering -
  - Clarity Business Solutions, Inc.
- MS Information Technology in Software, Carnegie Mellon University
- BS Computer Science, University of Pittsburgh
- 22 years Software Engineering experience
- Maryland Data Works Meetup
- Professional Internet & Data Person since 1998
- Parallel tinkerer - RaspberryPi, Arduino, Prusa 3D printer, SDR (beginner)
- Dad to 4 growing children and 3 dogs and 1 cat
- Books currently reading and/or audiobooks
  - Weapons of Math Destruction (UMD suggestion)
  - Infinite Powers - The Story of Calculus, the Language of the Universe

My expectations of you

- Watch the lectures.
  - If it's important enough for me to put in the lecture. You should pay attention to it.
- Do the assigned readings.
- Self Directed Learning
  - A **learning** strategy which allows learners take charge of their own **learning** process (diagnosis **learning** needs, identify **learning** goals, select **learning** strategies, and evaluate **learning** performances and outcomes).
  - I will provide reading assignments and additional videos. You have to let me know if you are not understanding the material.
  - It's ok to not understand at first. It can be frustrating but don't let that get you down. The instructional staff is here to help.
- Keep up with the labs and assignments.
- Communicate with me and the TA (please give me 24-48 hours to respond).
- If you are having difficulty, don't wait till the last moment, let me, Jonathan, or Jeffrey know as early as possible.

Your expectations of me

- Provide an inclusive and equitable classroom climate. (virtual - hybrid of recorded lectures and ‘in-person’ time)
- We have a shared goal of academic progress.
- Approachable
  - If I’m not reaching out to you as much as you would like, it is a two-way street!
    - REACH OUT TO ME or TA (but please give me 24-48 hours to respond)
- Provide clear weekly direction
  - I am still learn fairly new to ELMS/Canvas and am just finding all of the capabilities included with it
    - You may see some usability changes within the first two weeks.
- Provide fair assessment (grades)
  - If you do the work and you do it on time then you WILL pass this course.
  - It’s not just about ‘passing’ the course, it’s about learning the material.

General Class Structure

- Readings
- Lecture videos
- Labs
- Assignments
- Tests
- Group project

- Tuesdays
  - Slides and Video
  - Readings
  - Practice
- Thursdays
  - Live ~30 minutes session
    - Will be recorded
  - Labs

Admin

- Office Hours (need to schedule a time slot):
  - Tuesday 6-7:40 pm
  - Friday 8-10 am
  - Saturday 8-10 am
  - Sunday 4-6pm
  - By Appointment *
- Tentative Live class meetings - (This is not a lecture, more of a review to see if anyone needs anything, classroom time, view lectures prior, software issues)
  - Thursdays 12:30-1:15pm (Class originally scheduled to start @ 12:30)
  - We can add a couple of these at different times if needed
    - I am trying to be flexible

UTA - Jonathan Chen

- Currently is a senior in Information Science
- Email: jonnyapple985@gmail.com
- Office hours:
  - Thursday's 4-5
  - https://umd.webex.com/umd/j.php?MTID=maa654cf56a69519872456151b7d2c073

Grader - Jeffrey Chen

- iSchool Alumni - 2019
- Currently in second semester for the Master's in Information Systems program at the R.H. Smith School of Business
- jeffrey.chen@rhsmith.umd.edu

Syllabus Updates

- NTR

INST447 General Schedule

- Tentative as of 9/8

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					8-10am	8-10am
		Noon		12:30-1:00pm		
4-6pm						
		6-7:40pm		11:59pm		

Office Hours

Video Ready

Live Sessions

Lab Due

Data Science Skills

https://towardsdatascience.com/data-science-minimum-10-essential-skills-you-need-to-know-to-start-doing-data-science-e5a5a9be5991

- Coding
- Math (stats)
- Ethics (notice I didn't put this last)
- Team player
- Lifelong learning
- Communication
- Real world project skills (PM)
- ML
- Data visualization
- Data wrangling and preprocessing

COLLEGE OF INFORMATION STUDIES

Projects

Projects

- Teams of 2, 3, or 4
  - I prefer sizes of 3 or 4
  - Project proposals due 10/1, so you have time
- API Keys
  - Twitter (see my submission process)
- Scraping
  - Reddit example (json)

Data Science Projects

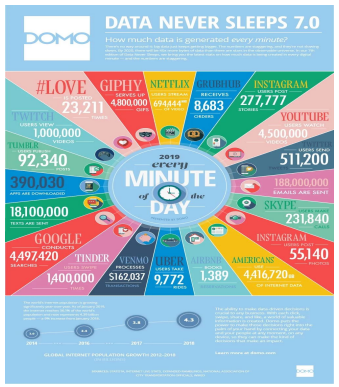
Data science - the ability to take large amounts of data in many different formats and be able to understand it, to process it, to extract value from it, to summarize it, to visualize it, and to communicate it to others.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value. -datarobot.com

FEC.gov

YAHOO! FINANCE

United States® Census Bureau



- Sentiment Analysis
- Customer Segmentation
- Recommending products
- Public Health Issues
- Manufacturing - predicting faults
- Financial Risk Analysis

U.S. DEPARTMENT OF COMMERCE
BUREAU OF THE CENSUS

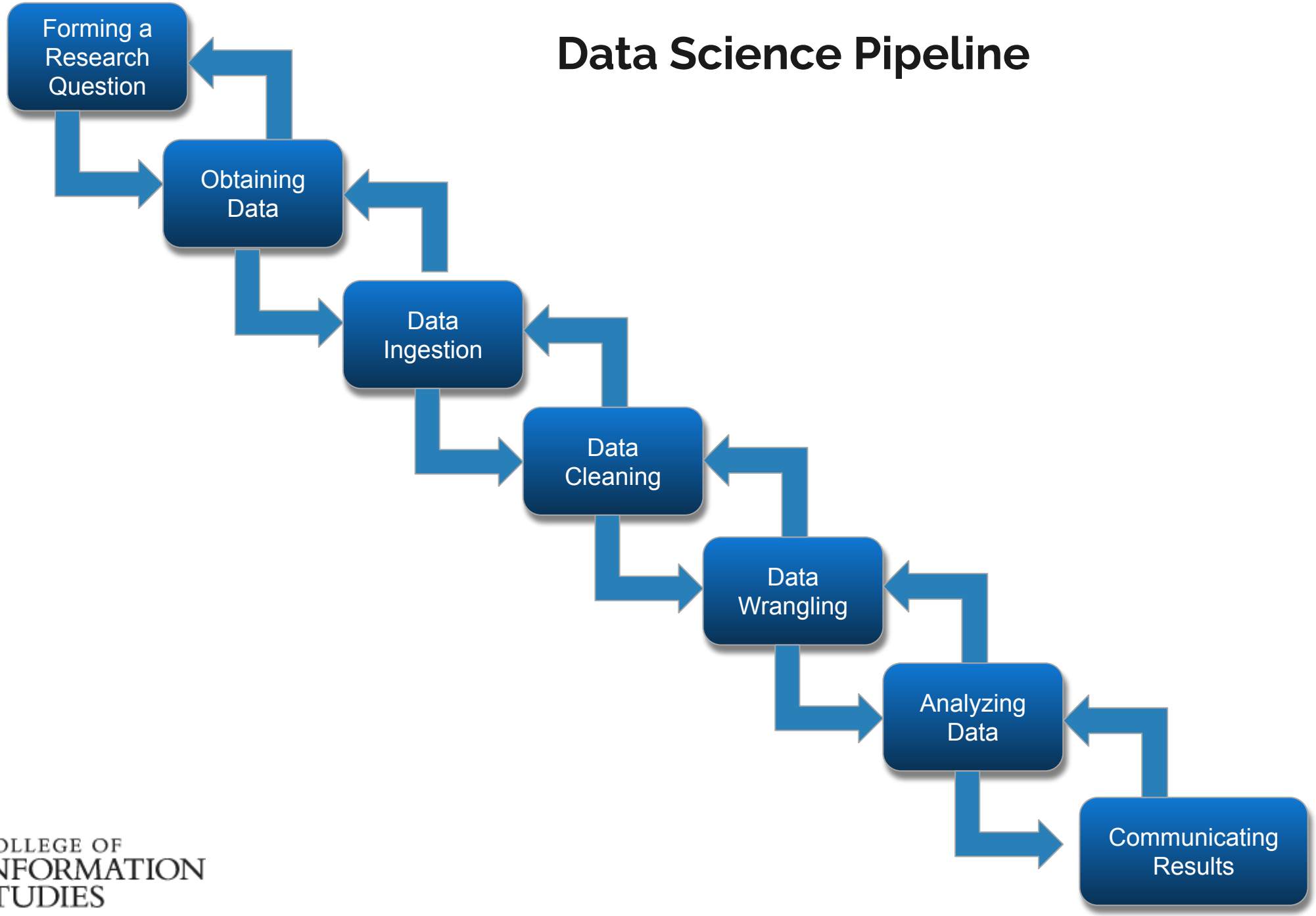
Illustration from:
https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2019-version-infographic/558793/

UNIVERSITY OF MARYLAND COLLEGE OF INFORMATION STUDIES

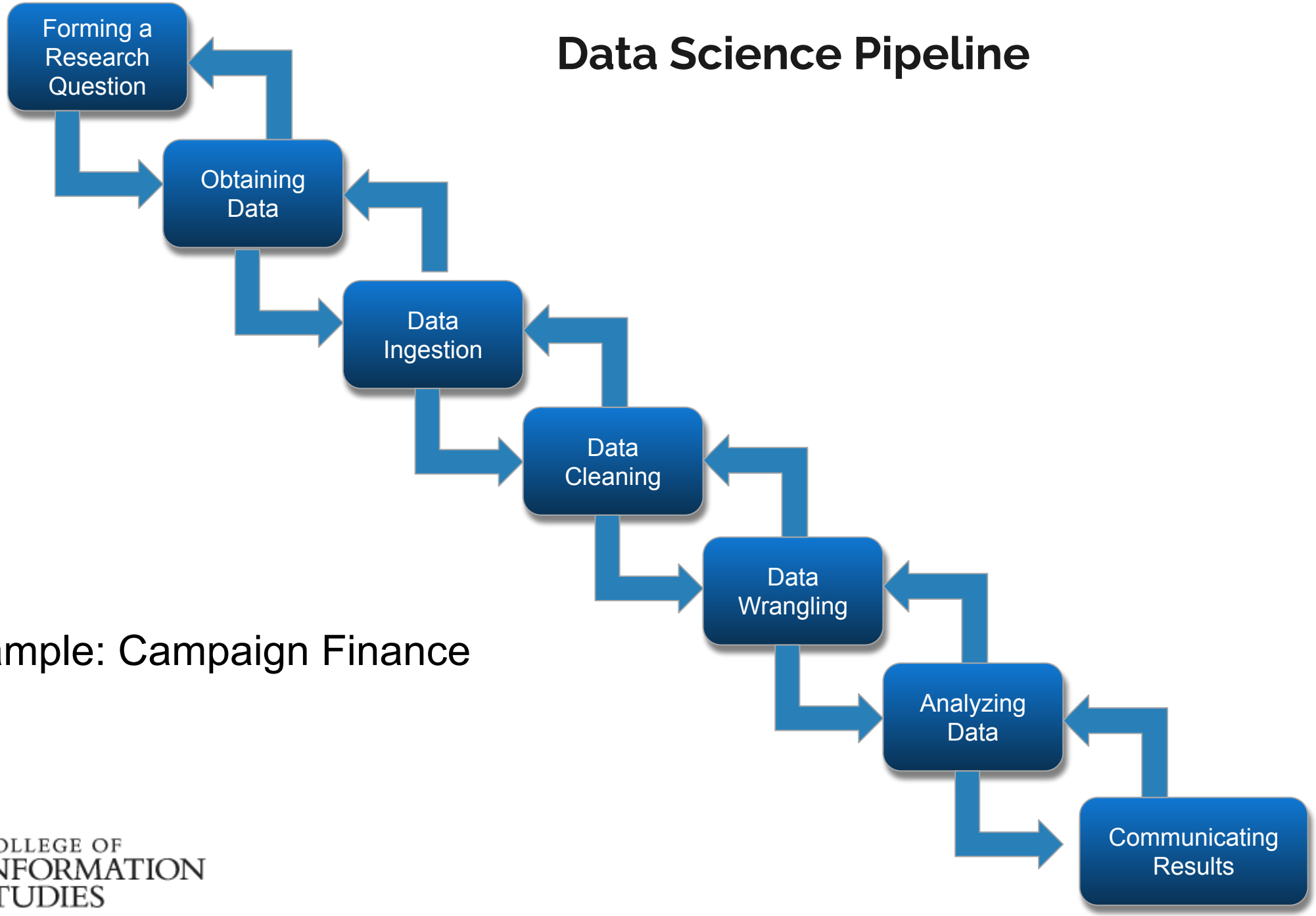
COLLEGE OF INFORMATION STUDIES

Data Science Pipelines

Data Science Pipeline

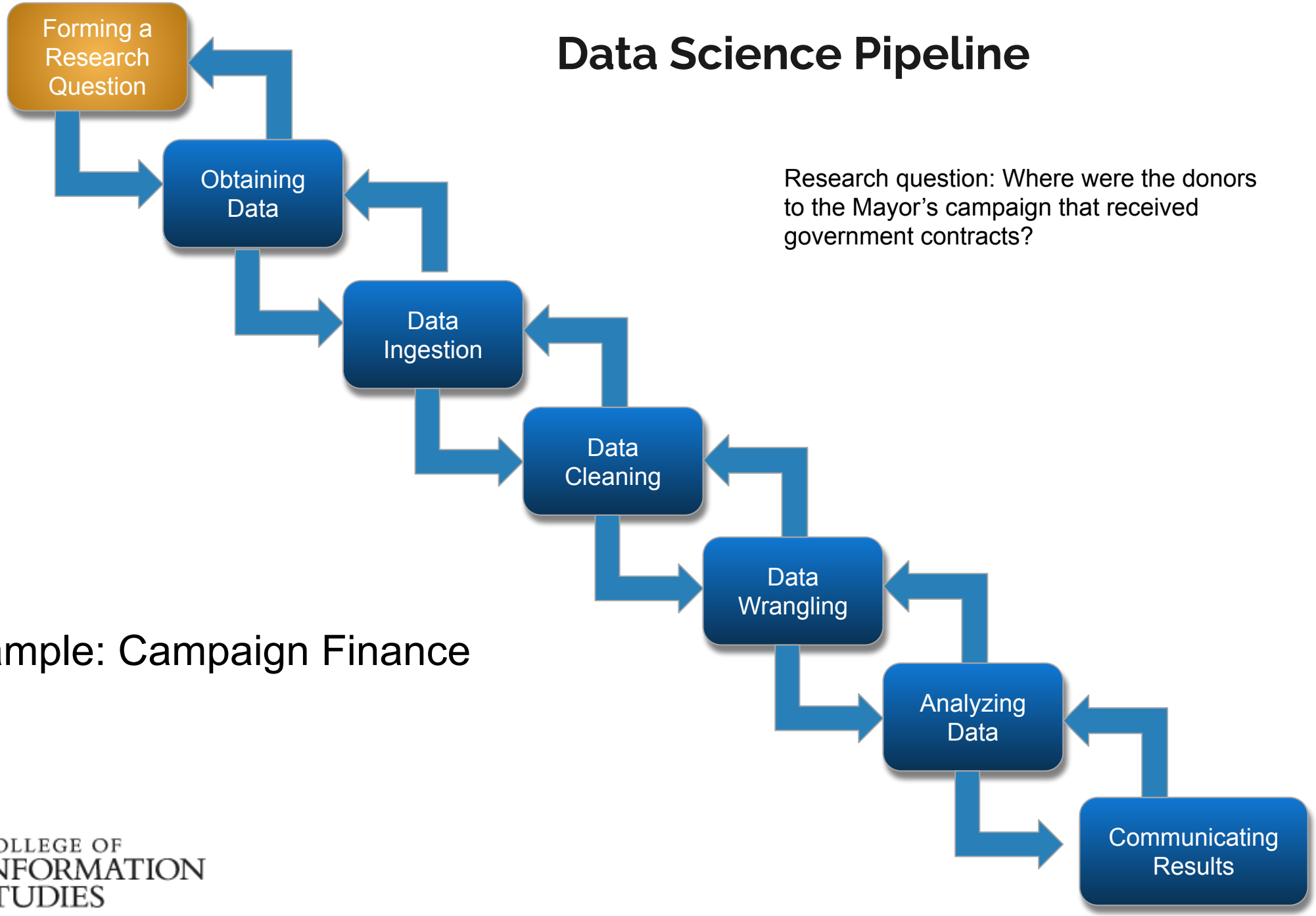


Data Science Pipeline



Example: Campaign Finance

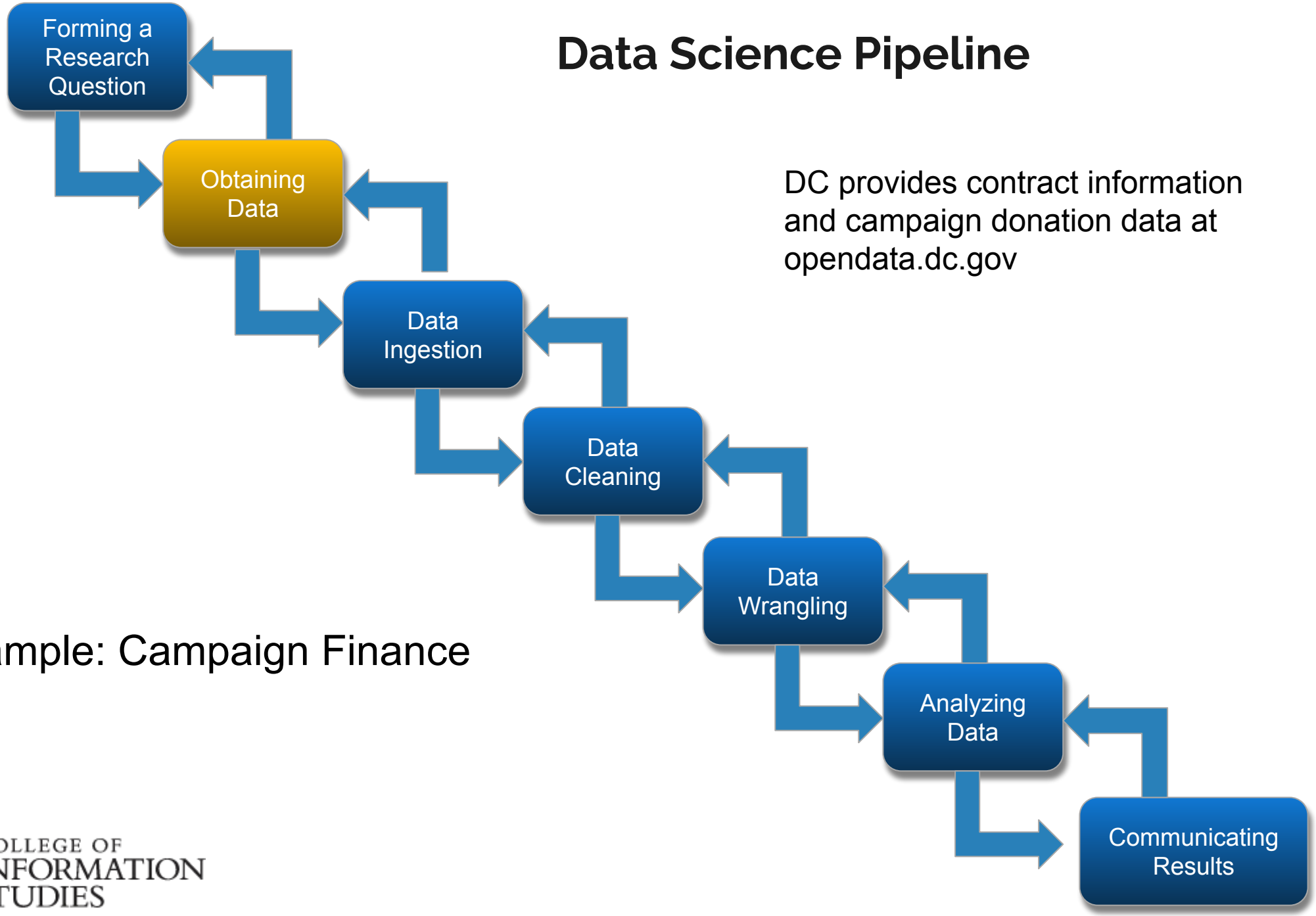
Data Science Pipeline



Research question: Where were the donors to the Mayor's campaign that received government contracts?

Example: Campaign Finance

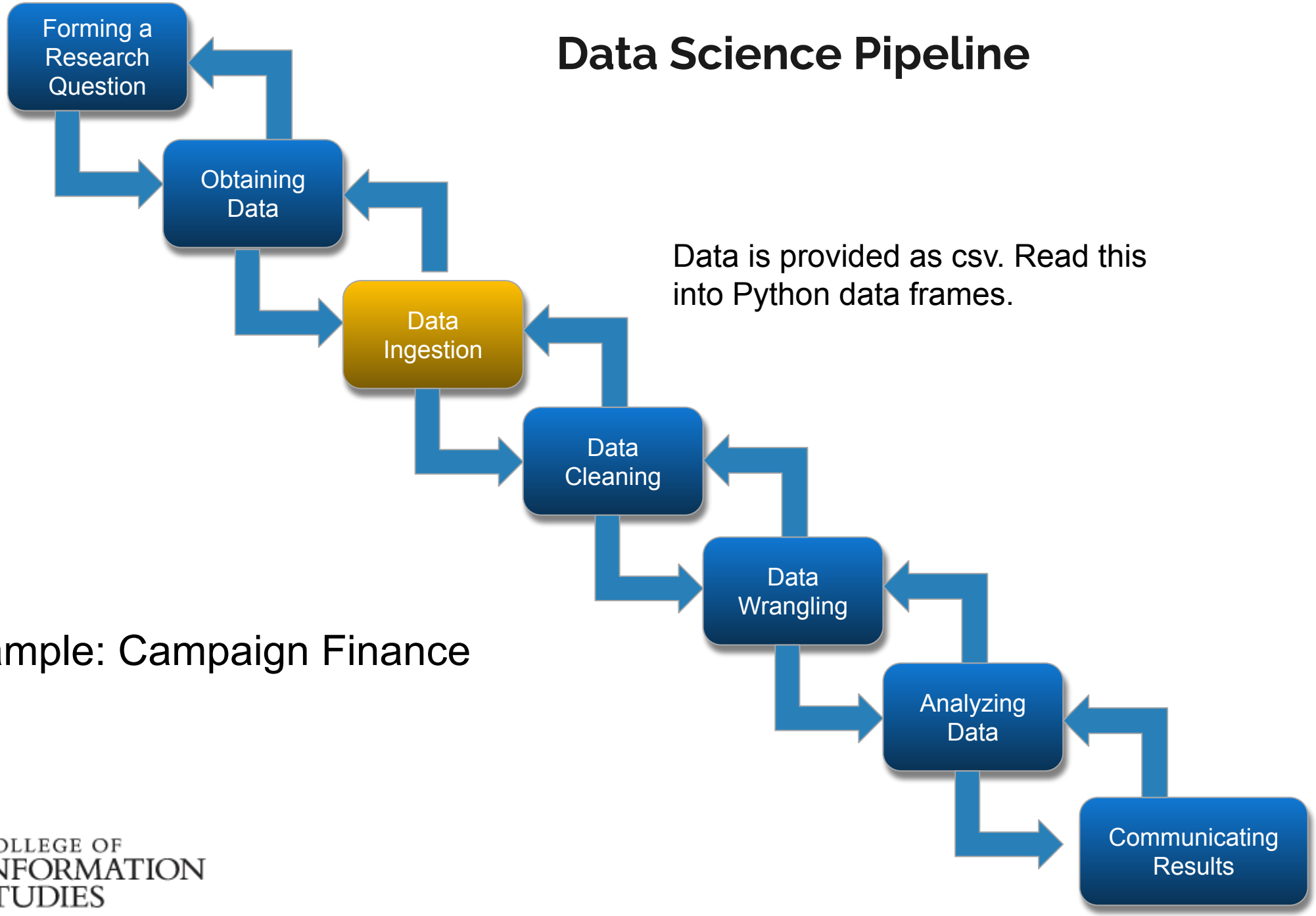
Data Science Pipeline



DC provides contract information and campaign donation data at opendata.dc.gov

Example: Campaign Finance

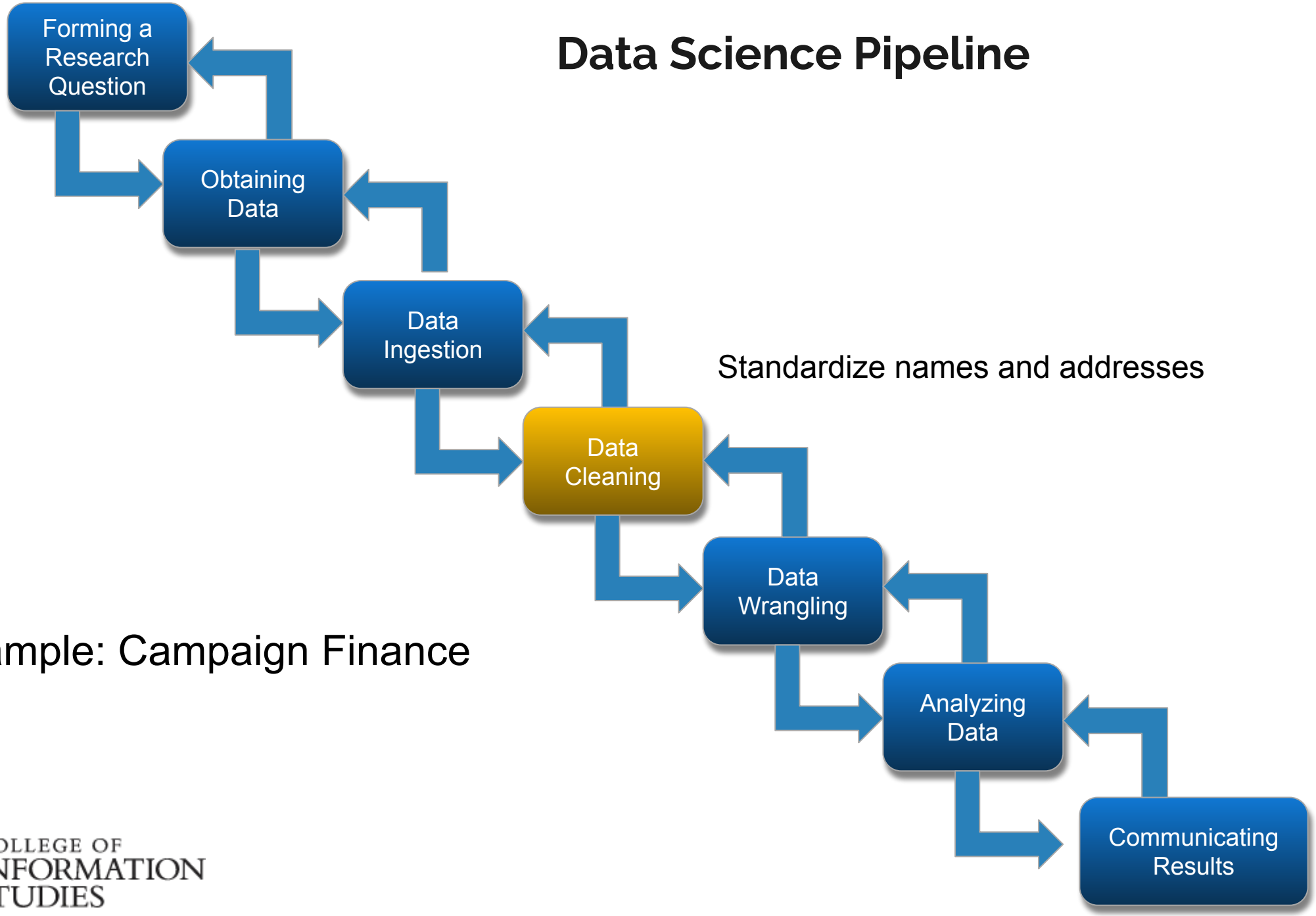
Data Science Pipeline



Data is provided as csv. Read this into Python data frames.

Example: Campaign Finance

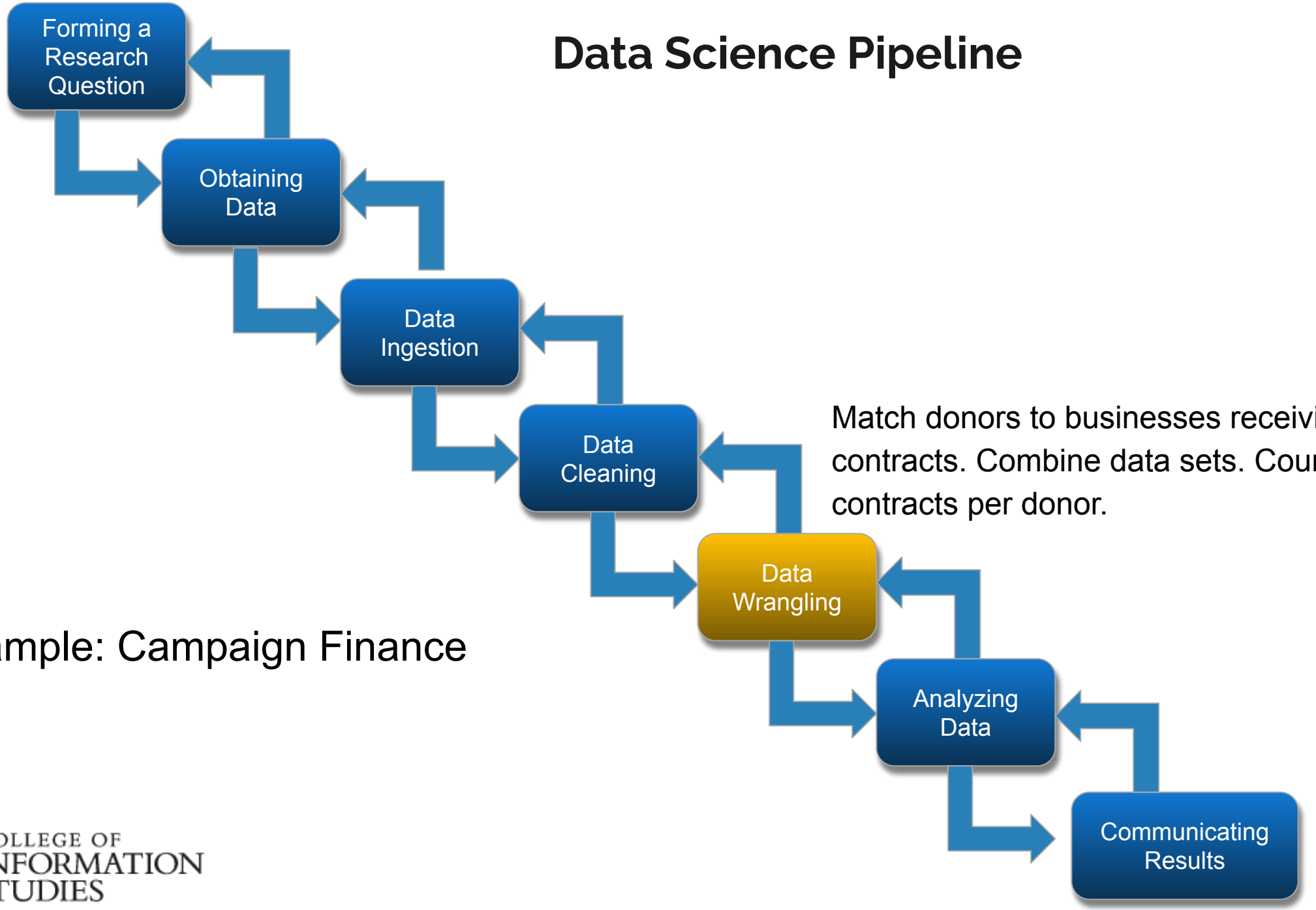
Data Science Pipeline



Standardize names and addresses

Example: Campaign Finance

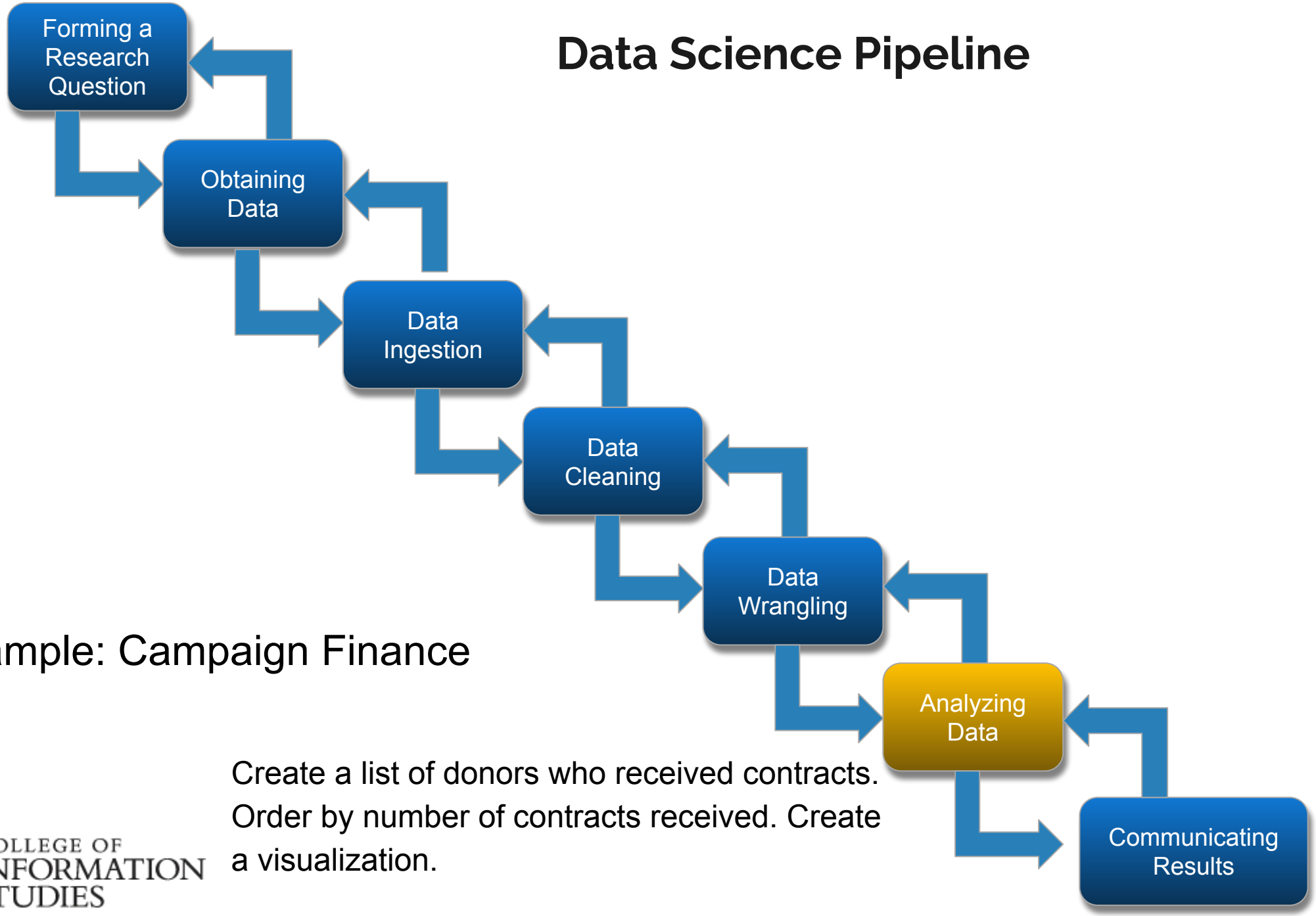
Data Science Pipeline



Match donors to businesses receiving contracts. Combine data sets. Count number of contracts per donor.

Example: Campaign Finance

Data Science Pipeline

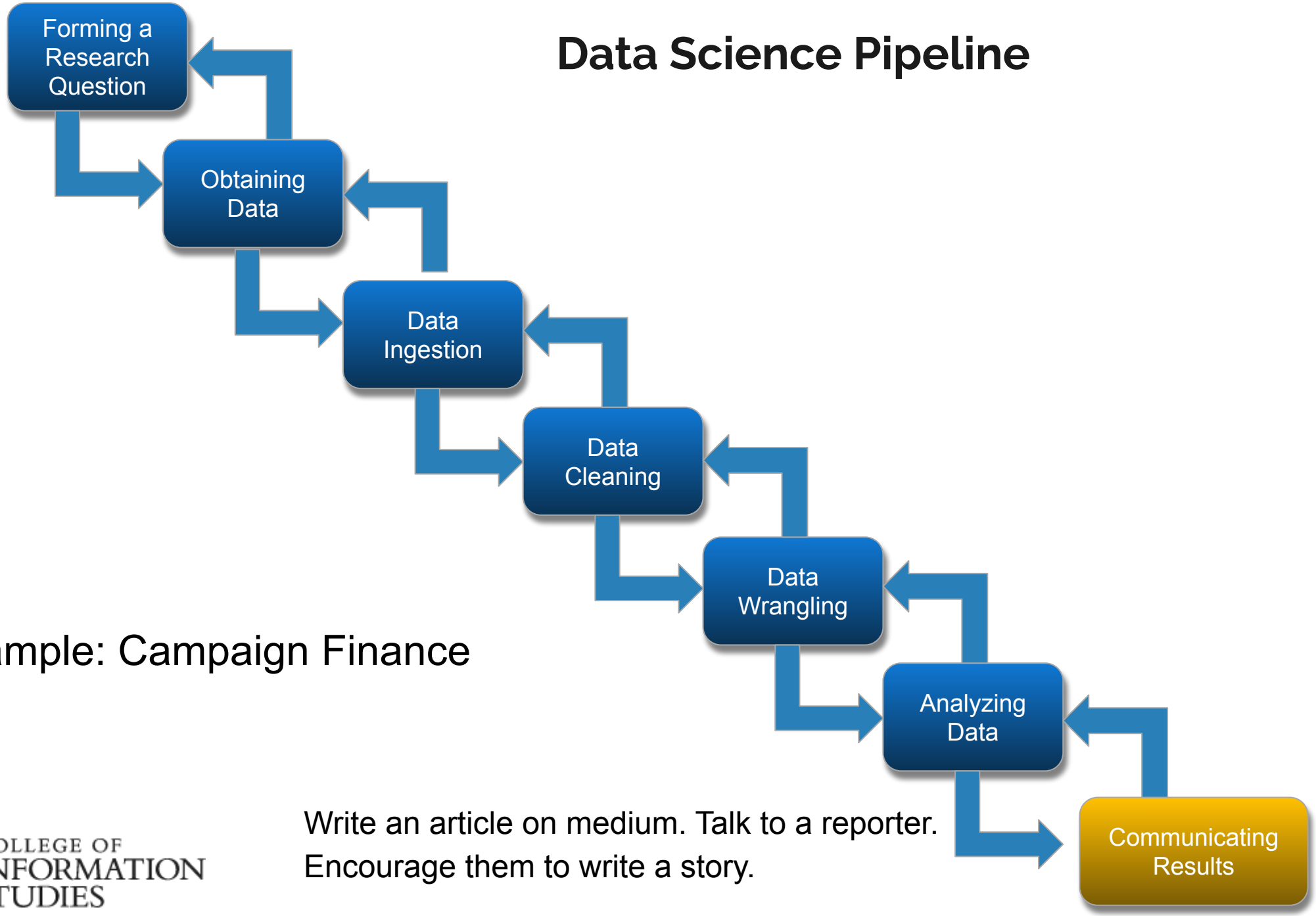


Example: Campaign Finance

Create a list of donors who received contracts. Order by number of contracts received. Create a visualization.

COLLEGE OF INFORMATION STUDIES

Data Science Pipeline



Example: Campaign Finance

Write an article on medium. Talk to a reporter. Encourage them to write a story.

COLLEGE OF INFORMATION STUDIES

COLLEGE OF INFORMATION STUDIES

Software Install

Software Tools

- Python 3 & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine *Week 3
  - http://openrefine.org/download.html

COLLEGE OF INFORMATION STUDIES

Python and Jupyter Notebooks

Python and Jupyter Notebooks review

- Python3 Installed
- Jupyter Notebooks Installed
  - Markdown
  - Save as
- Examples, Numpy and Pandas

Lab

Labs

- Focus on skills related to topic of week
  - e.g. regular expressions
- New data set(s) and research questions
- Work together in pairs in class
- Perform manipulations on data set(s) to answer research questions.
- Enter your answers in ELMS. Turn in *own* Jupyter notebook.

COLLEGE OF INFORMATION STUDIES

Next Week

Next Week

- Data Cleaning
- Proposal Brainstorm
- More Pandas
- Assignment 1 will be available
- * Much more reading for next week

I appreciate your attention

Hope to see you on Thursday!

COLLEGE OF INFORMATION STUDIES

Reference Material
Install Software

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

COLLEGE OF INFORMATION STUDIES

For Thursday

How Twitter can predict an election

DiGrazia, McKelvey, Bollen et al. (2013) More tweets, more votes: Social media as a quantitative indicator of political behavior. PLoS One, 8, 1-5.

- What is this paper about?
- What do they argue and why?

How Twitter can predict an election

- What are the strengths of the paper?
- What are the weaknesses and limitations?

How Twitter can predict an election

How would you go about reproducing this study? What are the steps be detailed. Write down at least one step at each stage in the pipeline

Additional questions to consider to be detailed:

- How can you download data from Twitter?
- How big is a tweet?
- Will there be a problem storing 500 million?
- How will you know what congressional district the person tweeted from?

01 Admin

02 Readings

03 Data Cleaning, transparency, etc.

04 Open Refine

05 Lab

06 Assignment

07 Next Class

This Week

Time: Tuesday virtual

- Admin
  - Office Hours Updates
  - Syllabus Updates
  - Piazza Added
- Readings
  - BadData
  - Scaling Data
- Data cleaning, transparency, etc.
- Lab will be cleaning up data with OpenRefine

Time: Thursday Virtual w/ optional live session

- Live session
  - OpenRefine examples
  - Jupyter Notebooks subjects from reading
    - Indexing
    - Concat & Append
- Lab & Assignment
- Projects - teams

If you are tired, stand up in the back of class.

Use of phone during class for non-class purposes is rude.

COLLEGE OF INFORMATION STUDIES

Admin

Admin

- Office Hours (need to schedule a time slot):
  - Monday 8-9 pm
  - Friday 8-10 am
  - Saturday 6-8 pm (changed from am to pm)
  - Sunday 6-7 pm (changed from 4-6 to 6-7)
  - By Appointment * Anytime
- Live class meetings - Thursdays 12:30-1:30
  - Class originally scheduled to start @ 12:30 so I figure this is a good time
  - We can add a couple of these at different times if/when needed
- Micro videos ? (e.g. running a notebook, Twitter account, other)
- Piazza vs. Canvas discussions

INST447 General Schedule

- Update 9/15/2020

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					 8-10am	
		 Noon-ish		 12:30-1:30pm		
 6-7pm						
	 8-9 pm				 Lab 11:59pm	 6-8pm

 Office Hours Live

 Office Hours By appointment

 Video Ready

 Class Live Sessions

 Lab Due

UTA - Jonathan Chen

- Currently is a senior in Information Science
- Email: jonnyapple985@gmail.com
- Office hours:
  - Thursday's 4-5
  - https://umd.webex.com/umd/j.php?MTID=maa654cf56a69519872456151b7d2c073

Grader - Jeffrey Chen

- iSchool Alumni - 2019
- Currently in second semester for the Master's in Information Systems program at the R.H. Smith School of Business
- jeffrey.chen@rhsmith.umd.edu

Syllabus Updates

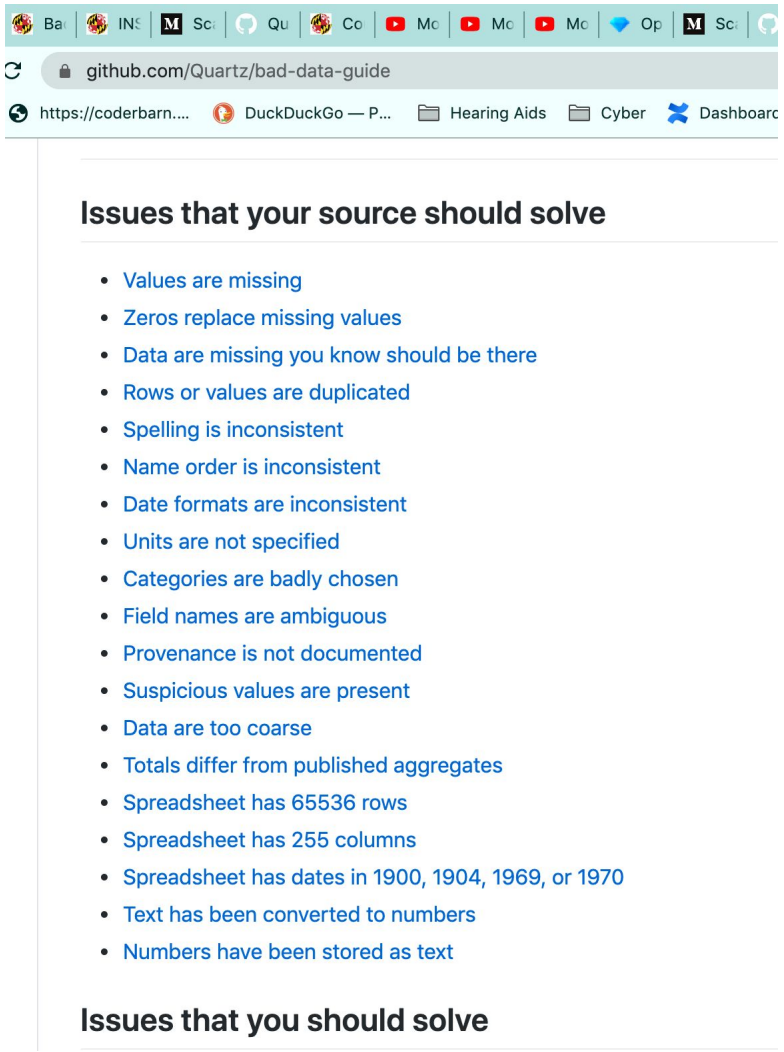
- General syllabus schedule still applies
- Canvas -> Modules

COLLEGE OF INFORMATION STUDIES

Readings

BadData Guide

- https://github.com/Quartz/bad-data-guide
  - Issues that your source should solve
    - Missing values, duplicate rows, ambiguous field names, inconsistent date formats, etc.
  - Issues that you should solve
    - Seasonal variation skews the data, data entered by humans (error prone), non random data (time-of-day, native language, etc.)
  - Issues a third-party should help solve
    - Author is untrustworthy (get two or three sources), inexplicable outliers
  - Issues a programmer should help solve
    - Data are aggregated to the wrong categories or geographies
      - e.g. data aggregated by zip code rather than city neighborhoods



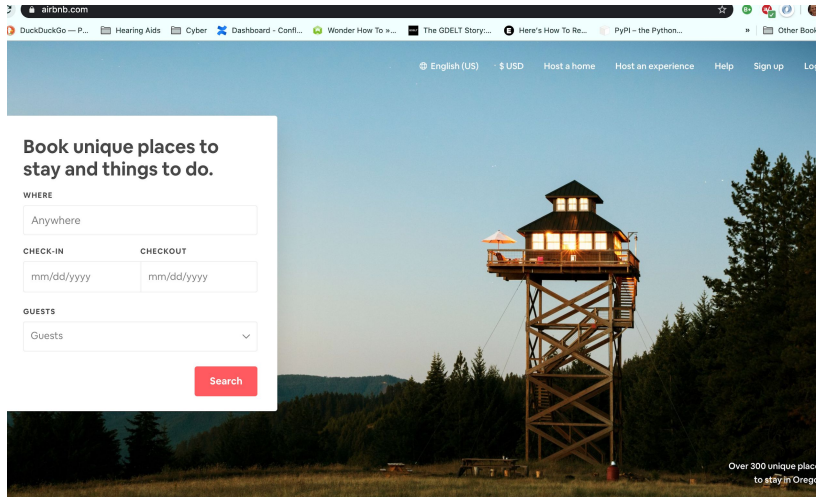
Scaling Knowledge at Airbnb

Data team

Problem: <u>The data team at Airbnb has a responsibility to scale the ability to make decisions using data.</u>

- @ Airbnb - “The democratize data access to empower all employees to make data-informed decisions”
- Give everybody the ability to use experiments to correctly measure the impact of decisions
- Turn those insights on user preferences into data products that improve the experience of using Airbnb.

- ***BUT…***.How to make an insight discovered by one person transfer effectively beyond the target recipient **“scaling knowledge”**
- Issues, just like other DS/SWE teams
  - Disorganized knowledge repos (local, servers, emails)
  - Previous work doesn’t have up to date code.
  - Current version of code isn’t what generated the previous plots
  - General issue of trying to reproduce what someone else did and not being successful
  - She distributes her results in a presentation, email, or doc perpetuating the cycle
- All of this slows down analysis and speed of decision making
- A streamlined approach is needed. Realized that they could do better!

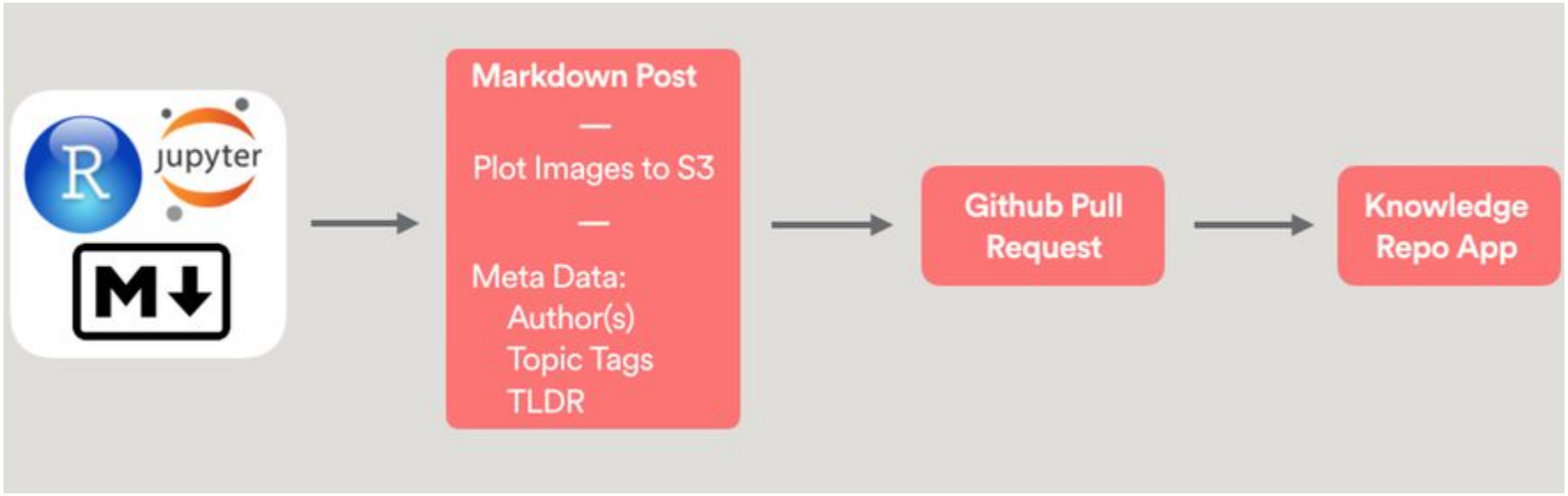


Scaling Knowledge at Airbnb

- Five Key Tenants for DS research going forward
  - *Reproducibility*
    - There should be no opportunity for code forks. The queries, transforms, visualizations and write-ups should be contained in each contribution and be up to date with the results.
  - *Quality*
    - Research should not be shared without being reviewed for correctness and precision (code/peer reviews)
  - *Consumability*
    - The results should be understandable to readers. Aesthetics should be consistent and on brand across research.
  - *Discoverability*
    - Anyone should be able to find, navigate, and stay up to date on the existing set of work on a topic
  - *Learning/Transparency*
    - Other researchers should be able to expand their abilities with tools and techniques from others' work

Scaling Knowledge at Airbnb

“Knowledge Repo”



Combined all of their ideas into one system. It combines a process around contributing and reviewing work, with a tool to present and distribute it. They call it a ‘Knowledge Repo’. Git repo. Posts are written in Markdown. Everything is committed. Templates for code - metadata (author, tags,and a TLDR). A Flask web-app renders the Repo’s contents as an internal blog, organized by time, topic, or contents.

Scaling Knowledge at Airbnb

Code Review of Software Engineering
+
Peer review of academia
=
Trusted, repeatable research going at
'startup speed'

Constantly checking for improvements

Scaling Knowledge at Airbnb

- Results
  - *Reproducibility*
    - All of the work, from the query of the core ETL (Extract, Transform, Load) tables, to the transforms, visualizations, and write-up is contained in one Jupyter NB, RMarkdown, or markdown file.
  - *Quality*
    - Using GitHub's functionality of pull requests prior to publishing, peer review and version control is put directly into the work flow.
  - *Consumability*
    - Markdown served by the web-app hides code and uses their internal branded aesthetics, making the work more accessible to less technical readers. The peer review process provides feedback on writing and communication, improving the quality of work.
  - *Discoverability*
    - The structured metadata (author, tags, TLDR) allows for easy navigation through past work. Tags provide a many-to-one topic inheritance and searching. Users can subscribe to topics. Posts can be bookmarked, browsed by author.
  - *Learning*
    - By having previous work easily searchable, it becomes easier to learn from each other.

Other (optional interesting reads)

- “**Working with Data Across Services is Hard**“ Billions of Messages a Day - Yelp's Real-time Data Pipeline
- news.ycombinator.com - search on data ‘pipeline/sets/cleaning/etc.’
- engineering.salesforce.com - TLS Fingerprints w/ JA3 and JA3S (https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967)
- https://towardsdatascience.com/
- https://www.reddit.com/r/datasets/

Open Refine

- Watch the 3 videos. They are self-explanatory.
- https://openrefine.org/
- “Google refine” -> Google open sourced it

COLLEGE OF INFORMATION STUDIES

Data Cleaning, Validation, Transparency and Reproducibility

Data sources

Types of Sources

- Primary Sources - Data you (or your organization) has created (** Interesting for project)
  - Collected customer data, survey data, interviews
  - Health data
  - IoT, instrument, or log data
  - etc.
- Secondary Sources - Data other people have created or aggregated
  - Government, FOIA requests,
  - Shared scientific data, national surveys
  - Social Media and other third party APIs
  - Finance
  - etc.

Common Data source issues

Secondary data sources

- Corrupted or Untrustworthy Data
    - social media, government provided, shared
- Poorly Documented
    - It’s not often that you find well documented data
- Difficult/Unfamiliar Format - KML, FITS

Data Provenance

- A historical record of where the data came from, how it was collected, and how it was handled.
    - Who collected it?
        - Some orgs are biased and may cherry pick data
    - How was the data collected?
        - Be as detailed as possible, some collection methods are flawed
    - Were there any modifications?
        - Human?
        - Computer?
        - Human & Computer?

Case Study

- Pro-Russia Ads Dataset
    - https://www.reddit.com/r/datasets/comments/8s0wrr/dataset_of_3500_ads_by_prorussia_group/

Follow the links backwards to reconstruct the data provenance

- Who collected the data?
- How was the data collected?
- Was it modified/preprocessed?
    - Computer?
    - Human?

Follow the links backwards to reconstruct the data provenance.

- Look at the summary on the Reddit page
- Description
    - Dems in the US House Intel Committee released 3500 pdfs with texts and images
    - Who collected it? beeeeeeers
    - Facebook and Instagram provided it? We assume.
    - USHIC provided them on their website

Follow the links backwards to reconstruct the data provenance.

- beeeeeeeeers applied OCR to turn pdf -> text
- beeeeeeeeers converted text to json/csv extracting key elements, eg. cost of ad
- Was it modified?
  - We know that it was modified (see previous steps)
  - Not sure exactly how. third party software? Python?
- Were there mistakes introduced?
  - scripting errors, convenience sampling, misunderstanding of fields

What are common Data Formats

- Structured data - organized and machine readable
    - csv, xml, json
    - database
    - spreadsheet
- Unstructured data
    - Images
    - Web pages
    - emails
    - audio files
    - pdfs

Comma Separated Values . csv

File

Animals,Color
Elephant,Grey
Giraffe,Yellow
Dolphin,Blue

Representation

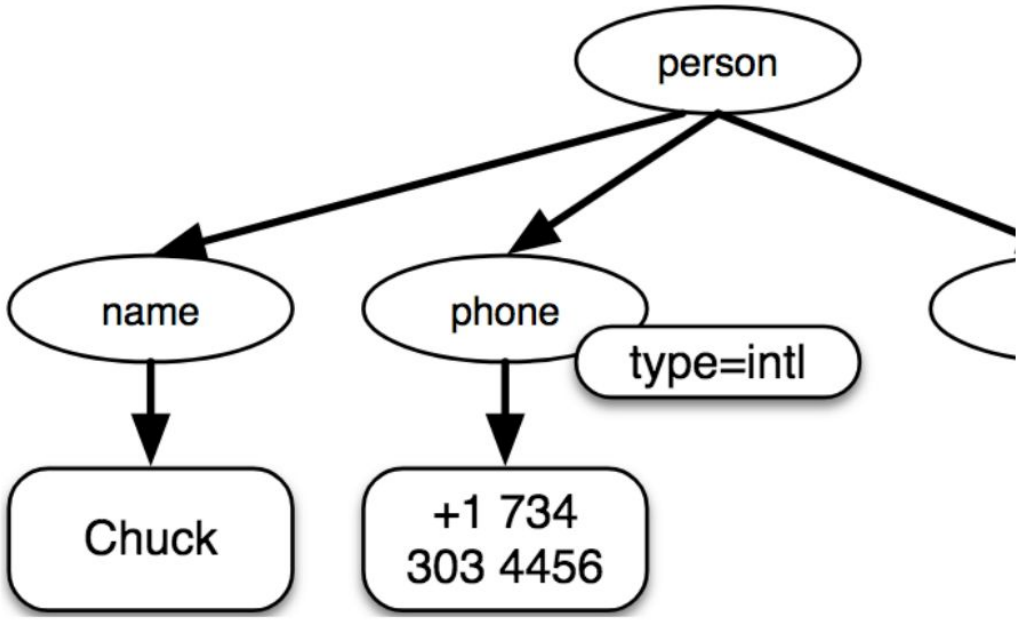
	A	B
1	Animals	Color
2	Elephant	Grey
3	Giraffe	Yellow
4	Dolphin	Blue

JavaScript Object Notation (JSON)

File

Representation

```
{
  "name" : "Chuck",
  "phone" : {
    "type" : "intl",
    "number" : "+1 734 303 4456"
   },
   "email" : {
     "hide" : "yes"
   }
}
```



Unstructured Data

- Data values are not organized in a standardized way.
- This makes it difficult to read and extract values with a computer, humans can read easily
- Examples
  - webpages
  - emails
  - audio files
  - pdfs

Data Orientation

- Get oriented with your data before you start your data cleaning
  - data provenance
  - what is the format?
  - how many observations?
  - how many variables?
  - what are the variables?
  - what do the values mean?
  - what are typical values for each variable?

Bias

Slides adapted from Nekabari Sigalo

How to handle data issues & biases

- Identify issues and biases
  - e.g. misspellings
- Address issues and biases if possible
  - e.g. fix misspellings
- Document issues and biases
  - e.g. record that misspellings were fixed in cells X & Y
- Decide whether results are valid despite issues and biases
  - e.g. results are more accurate because misspellings were fixed

Types of data issues & biases

- 4 Common Types
  - Biases due to data collection methods
  - Missing data
  - Inconsistent data
  - Data errors

Biases due to data collection

- Random Sampling - each entity in the population has an equal chance of being selected
  - e.g. randomly select 20 students from registry
- Convenience Sampling - A set of entities in that population that were easy to gather data from
  - e.g. select 20 students from physics 100
- Most statistical techniques assume data is collected through random sampling. In reality, sampling is almost always collected through convenience sampling which creates bias.

Identify if data collection introduced bias

- Common sources of bias based on flawed data collection methods
  - Sample size is too small
    - e.g. only 20 observations
  - Stopping procedure is based on data
    - e.g. tricking the data into giving you the result. You collect data until you get the result you want.
  - Convenience sampling collects a non-representative group
    - e.g. survey about exercise recruit people at a gym
  - Only capturing data for part of a “season”
    - e.g. collect uber traffic data on M-Th when most traffic happens on weekend

Address bias from data collection methods and decide if problematic

- Address bias from data collection methods
  - Use good data collection methods if you are collecting yourself
  - Only use data sets collected by others that were collected using good methods that minimize bias
- Decide if results are valid given data collection methods
  - Evaluate how much bias is introduced on data collection methods. Evaluate how it affects your results

Missing Data

- Ideally you would have no missing data. There are several reasons why missing data occurs
  - Data was never recorded
    - e.g. sensor failure, human did not respond to question
  - Data was lost or corrupted
    - e.g. value out of range for database, human error

Missing Data can create bias

- Missing data can create biased results because data is rarely missing at random.
  - Extreme values are more likely to be incompatible with database
  - Humans often choose not to answer sensitive questions
    - e.g. low income individuals may not want to answer a question about income
    - e.g. heavy drug users may not want to answer a question about drug use

Identify missing data

- Identify values used to encode missing data (e.g. N/A, blank, null)
  - Check documentation
  - How is it encoded
    - NA, N/A, NULL, Nan, “”, 0, -1
    - 1970-01-01T00:00:00Z
  - Inspect values, there may be a mixture
- Did you receive all of the rows and columns that you expected?
- Get a count of missing values per column

Addressing missing data

- Find out why you don’t have it
  - May not be able to
- Imputation methods
  - Infer it from data around it, make a guess, assume the average
- Exclude it
  - Ignore the entire row.
- Decide whether or not you can use the data if you are missing too many values!

Inconsistent data

- Dates
  - e.g. “2019-03-16” vs “March 16, 2019”
- Variants to represent the same values
  - “USA” vs “United States” vs “United States of America”
- Different units are used
  - e.g. height in cm and inches

Inconsistent data

- Standardize the values
  - e.g. reformat dates
  - e.g. combine variants by recoding values (e.g. “USA”)
  - e.g. use consistent units (inches vs cm)

Identifying and addressing data errors

- Unusual or unexpected values may be errors
  - e.g. out of range date “2070-03-23”
  - e.g. very large integer “99999999999”
  - e.g. negative number for something that should only be positive like height, weight, age
- Outliers may be errors
- How to address?
  - Try to fix
  - Exclude

Transparency and Reproducibility

Transparent & Reproducible Projects

- Transparency – It is easy for you and others to understand the data and how it has been analyzed
  - Documentation
    - Source of data & collection methods
    - Known issues & how they were addressed n Transformations & calculations
  - Justification
    - Why were these methods used

Transparent & Reproducible Projects

- Reproducibility – It is easy for you and others to repeat your analysis.
  - Create copies of data
    - Don’t overwrite your raw data create intermediate data sets
    - Save a final version of your data before analysis ¨ Create an automated pipeline
  - You will want to run your pipeline multiple times
    - Preferably included in only one script
  - OpenRefine automatically creates a list of cleaning steps taken

Design embodies good cleaning practices

- Reproducibility
  - Saves data to new file.
  - Built in versioning that allows you to reverse any action.
  - Can export and apply the same script to a new data set (or same data set again).

Documentation

- Metadata: Structured information describing a dataset.
  - e.g. dates collected, creators of dataset, variables
- Codebook: Description of variables and data values in dataset.
  - e.g. units, missing values, transformations etc.

Reproducibility

- Reproducibility - It is easy for you or others to repeat the steps of your analysis
  - Create copies of data
    - Don’t overwrite your raw data, create intermediate data sets
    - Save final version of your data before analysis
  - Create an automated pipeline
    - You will want to run your pipeline multiple times
    - Preferably included in only one script (often not realistic). Make sure documentation is up to speed!

COLLEGE OF INFORMATION STUDIES

Open Refine

OpenRefine / Google Refine

- Open source source tool to clean messy data originally created by Google.
  - Graphical User Interface
  - Scripting Language

- Using OpenRefine helps you to see the big picture of your data, discover inconsistencies, and fix them.

- Not sure about extremely large data sets; however, you can use a subset of data in combination with OpenRefine to find general problems.
-

Keep Track Changes OpenRefine

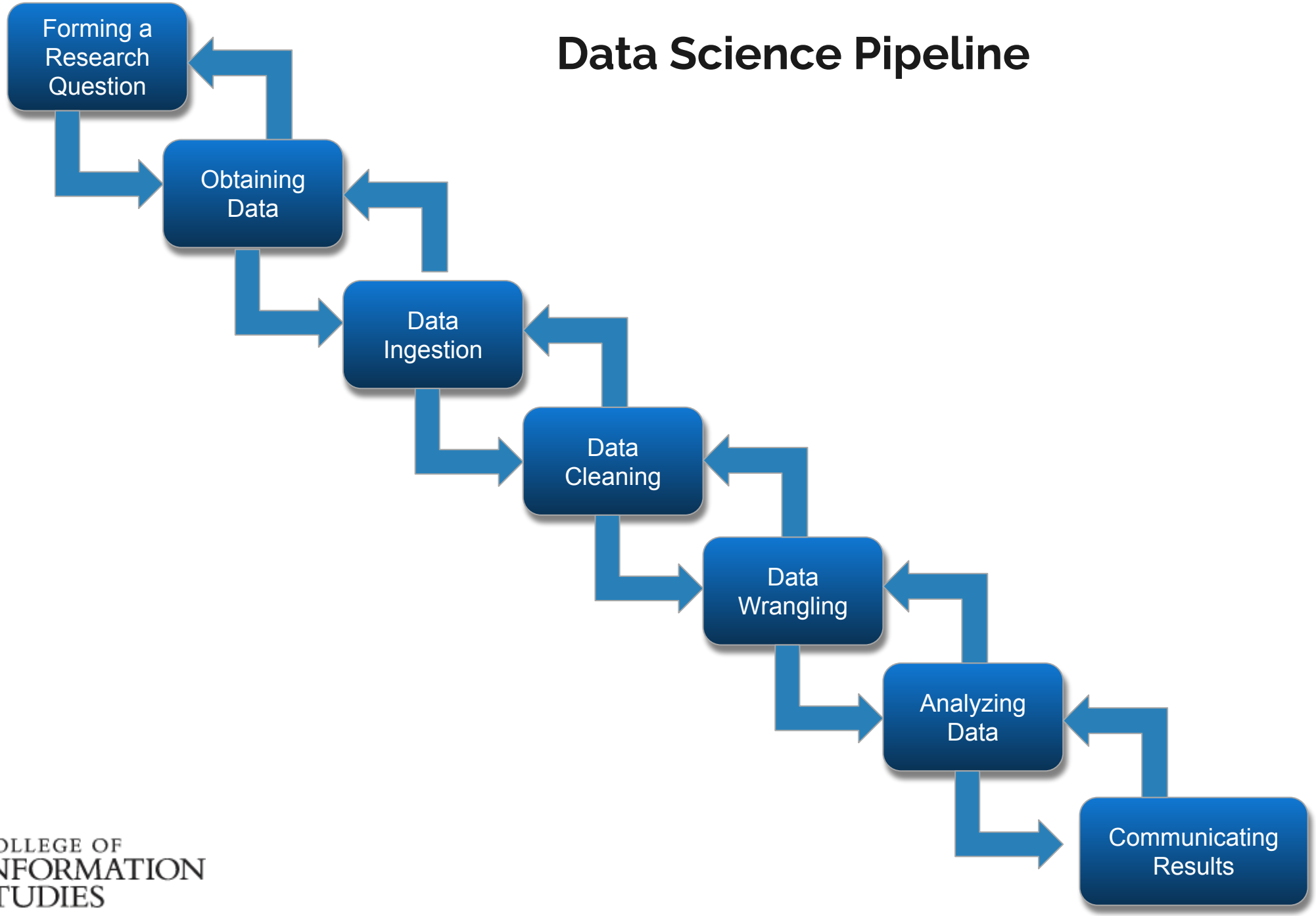
- Use the Facets to fix up standardization issues in the variable EMPLOYER
  - What kind of non-standardization issues do you notice?
- View a list of edits that you've made using the "Undo/Redo" tab
- Export a script with your list of changes using the "Extract" option in the "Undo/Redo" tab

Lab

COLLEGE OF INFORMATION STUDIES

Projects

Data Science Pipeline



Projects

- Teams of 2, 3, or 4
  - Class is small so I prefer sizes of 2 or 3
  - Project proposals due 2/28, so you have time
- API Keys
  - Twitter (see my submission process)
- Scraping
  - Reddit example (json)
  - Reddit group on data sets https://www.reddit.com/r/datasets/

Data Science Projects

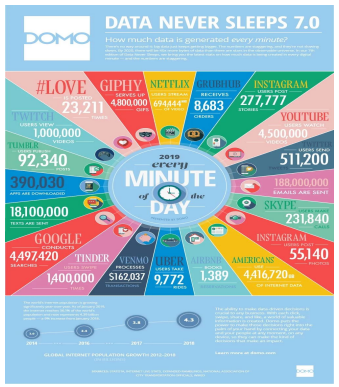
Data science - the ability to take large amounts of data in many different formats and be able to understand it, to process it, to extract value from it, to summarize it, to visualize it, and to communicate it to others.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value. -datarobot.com

FEC.gov

YAHOO! FINANCE

United States® Census Bureau



- Sentiment Analysis
- Customer Segmentation
- Recommending products
- Public Health Issues
- Manufacturing - predicting faults
- Financial Risk Analysis

Illustration from:
https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2019-version-infographic/558793/

COLLEGE OF INFORMATION STUDIES

Projects

https://engineering.salesforce.com/ & TLS Fingerprints JA3 and JA3S

Berkeley SETI Research Center

FITS file handling
https://docs.astropy.org/en/stable/io/fits/index.html

astropy:docs

COLLEGE OF INFORMATION STUDIES

Python and Jupyter Notebooks

Python and Jupyter Notebooks review

- Examples and Pandas

COLLEGE OF INFORMATION STUDIES

Next Week

Next Week

- Assignment 1 due
  - Size of data
- Biases
- Proposal Brainstorm
- More Python/Pandas
  - Data frames
  - Visualizing
  - Summarizing
-

I appreciate your attention

Hope to see you on Thursday!

COLLEGE OF INFORMATION STUDIES

Reference Material
Install Software

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

Software Tools

- Python & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine
  - http://openrefine.org/download.html
- Data sets
  - https://www.reddit.com/r/datasets/
  - https://opendata.dc.gov/
  - https://datasetsearch.research.google.com/
  - https://www.kaggle.com/datasets

01 Admin

02 DataFrames/Series

03 Indexing and Slicing

04 Summarizing Data

05 Lab

06 Projects

07 Next Week

This Week

Time: Tuesday virtual

- Admin
  - Syllabus Updates
- Readings
- Videos
  - Data basics review
  - Indexing
- Jupyter Examples

Time: Thursday Virtual w/ optional live session

- Live session
  - Jupyter Notebooks subjects from reading
- Videos
  - Concat & Append
  - Aggregations
- Lab & Assignment
- Projects - teams

If you are tired, stand up in the back of class.

Use of phone during class for non-class purposes is rude.

COLLEGE OF INFORMATION STUDIES

Admin

Admin

- Office Hours (need to schedule a time slot):
  - Monday 8-9 pm
  - Friday 8-10 am
  - Saturday 6-8 pm (changed from am to pm)
  - Sunday 6-7 pm (changed from 4-6 to 6-7)
  - By Appointment * Anytime
- Live class meetings - Thursdays 12:30-1:30
  - Class originally scheduled to start @ 12:30 so I figure this is a good time
  - We can add a couple of these at different times if/when needed
- Piazza vs. Canvas discussions

INST447 General Schedule

- Update 9/15/2020

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					 8-10am	
		 Noon-ish		 12:30-1:30pm		
 6-7pm						
	 8-9 pm				 Lab 11:59pm	 6-8pm

 Office Hours Live

 Office Hours By appointment

 Video Ready

 Class Live Sessions

 Lab Due

Syllabus Updates

- General syllabus schedule still applies
- Updated Canvas to be cleaner
- Contains fairly set schedule….some adjustments may be made

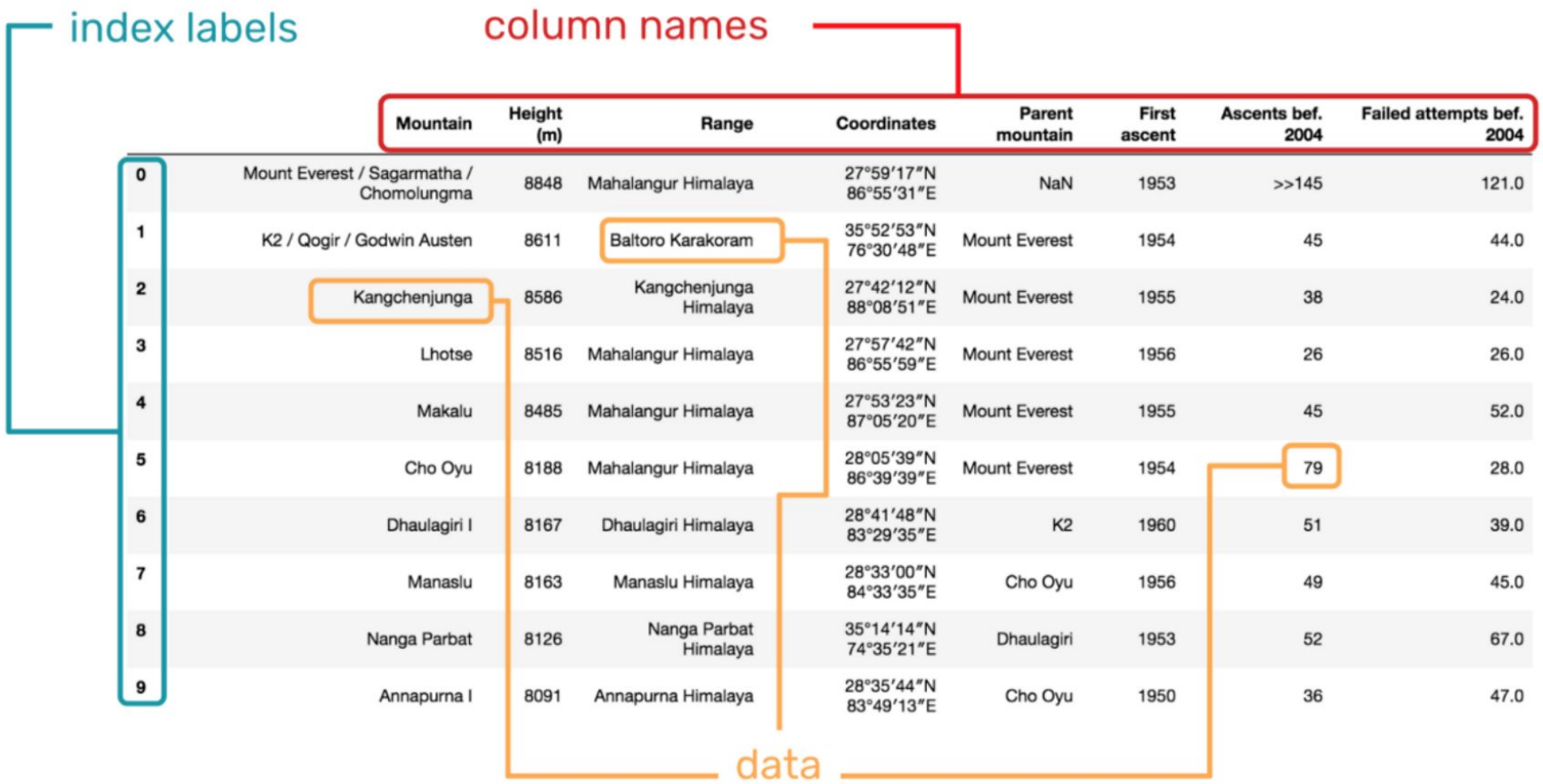
COLLEGE OF INFORMATION STUDIES

DataFrames / Series

Data Frames

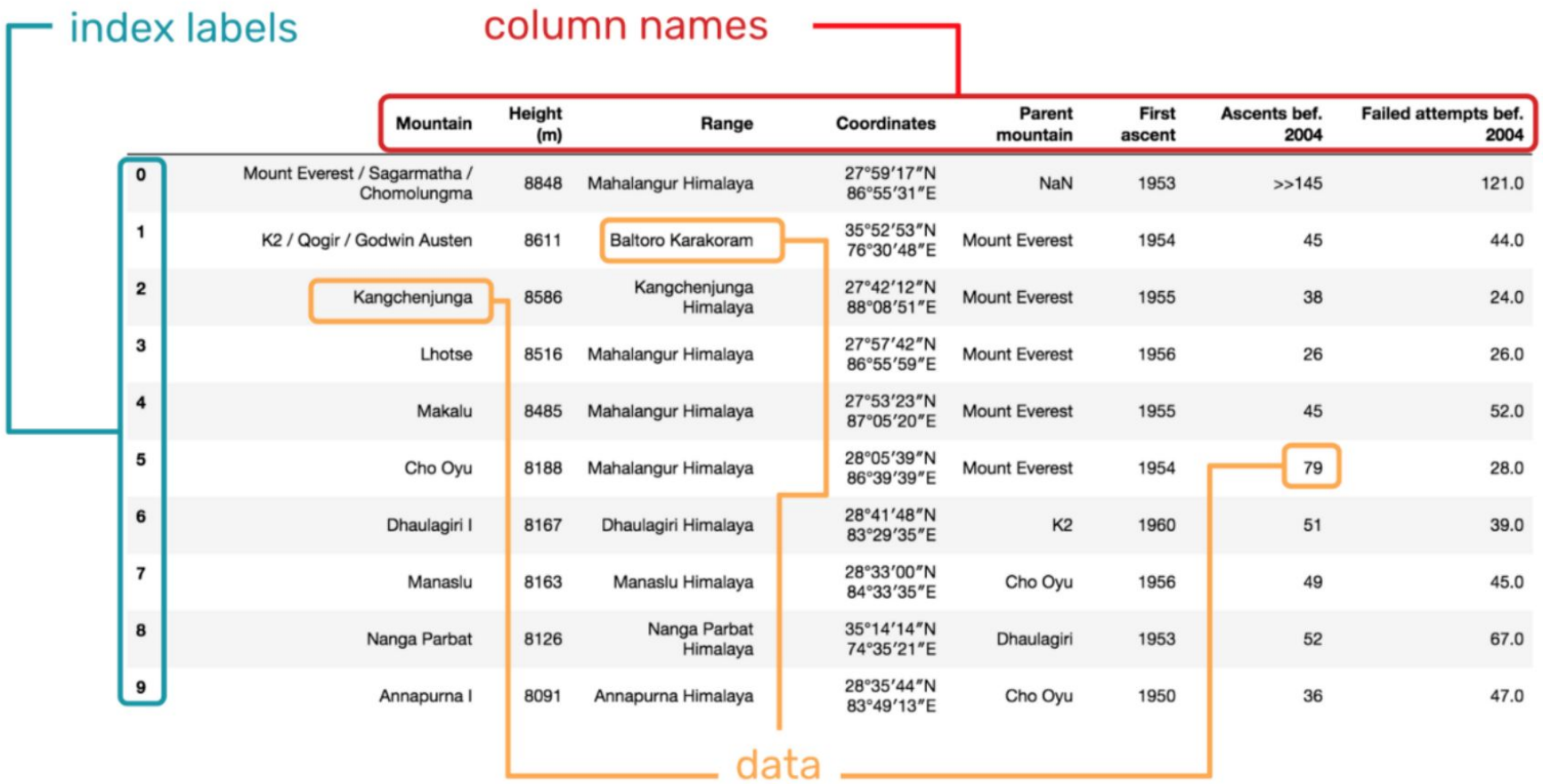
- 2 dimensional arrays with labels
- columns – variables
- rows – each observation
- cell – data value

Data Frames



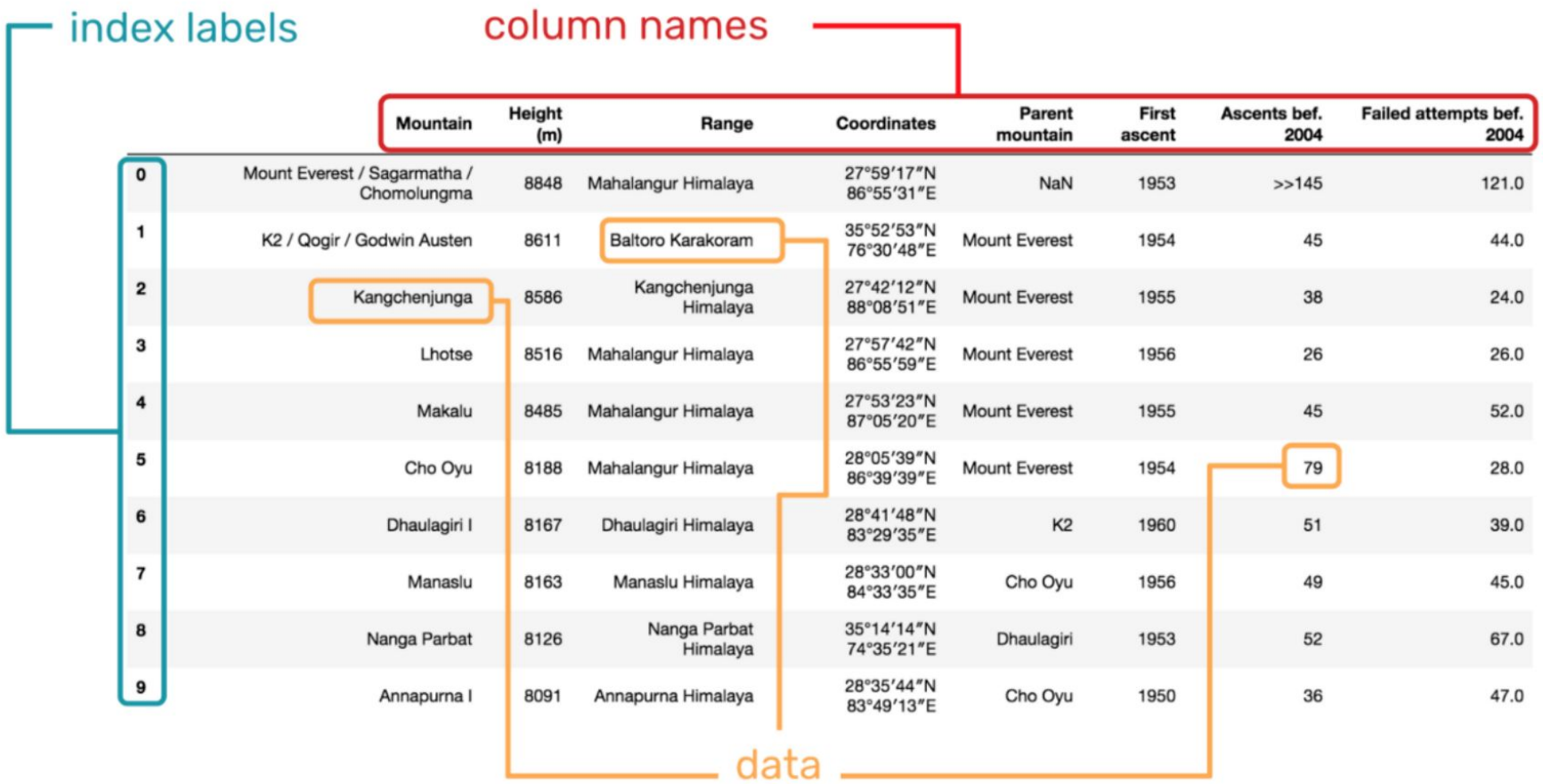
	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

What is a variable in the data set?



	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

What is an observation in the data set?

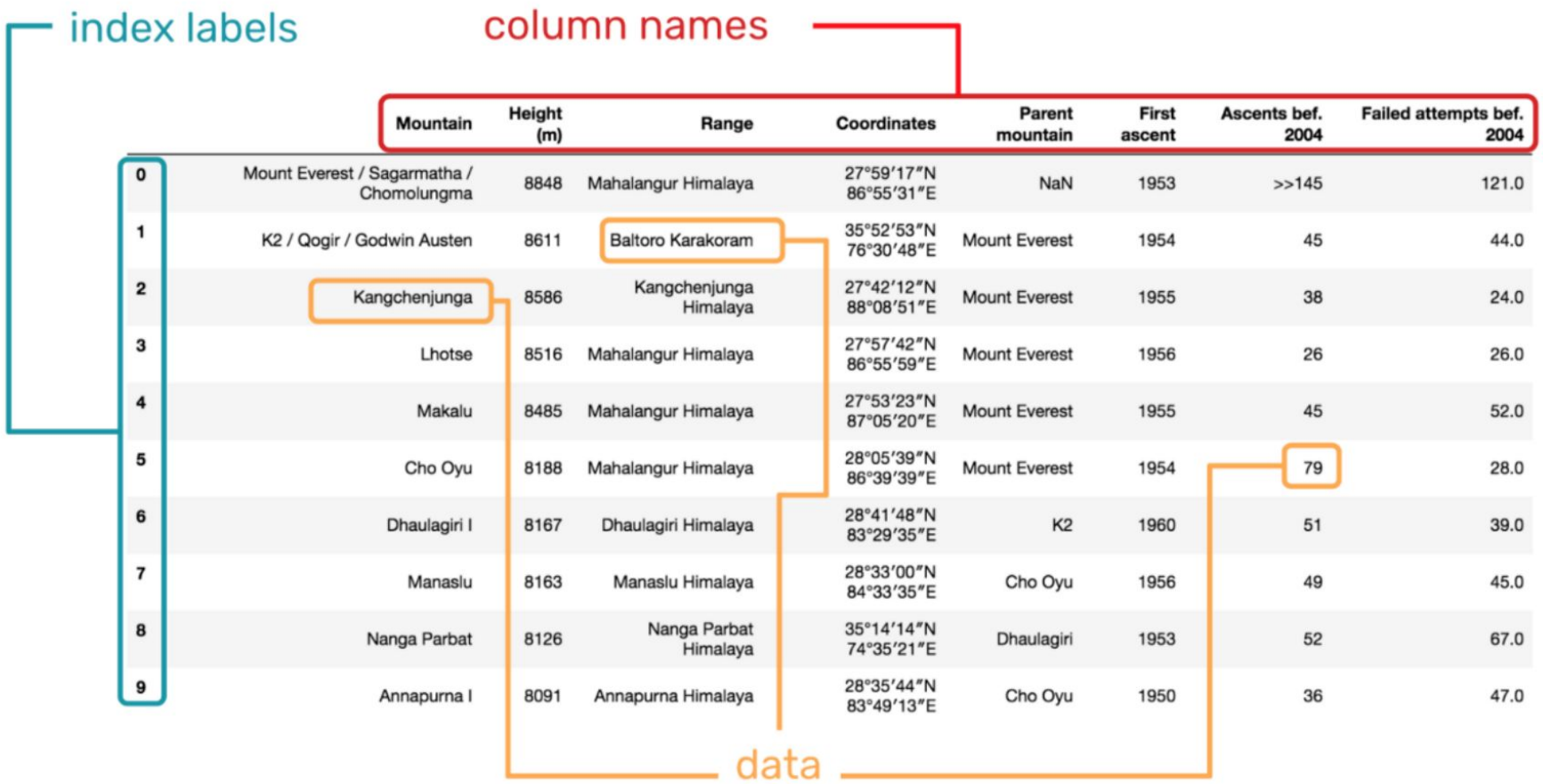


	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

Data Frame Labels

- Column labels aka columns
- Row labels aka index

What is the index for Lhotse?



	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

Create a DataFrame

- Read from file
  - import pandas as pd
  - df = pd.read_csv(‘path/filename’)
- Create using dictionary
  - dict = {‘col1’:[val1,val2],’col2’:[val3,val4]}
  - df = pd.DataFrame(dict)

Create a DataFrame with fruits and colors

- Fruits: apple, banana, orange
  - dict = {‘fruit’:[‘apple’,’banana’,’orange’], ‘color’:[‘red’,’yellow’,’orange’]}
  - fdf = pd.DataFrame(dict)

	fruit	color
0	apple	red
1	banana	yellow
2	orange	orange

Basic information about a DataFrame

- Prints first 5 rows of the data frame
  - df.head()
- Prints the number of rows and columns
  - df.shape (90, 41) 90 rows, 41 columns

Save a DataFrame

- Write data frame to csv
  - df.to_csv(“path/newfilename.csv”)

Series

- 1 dimensional array with labels
  - e.g. a column of a data frame
- best practice all values are same data type (e.g. int, float, string)
- You can think of a data frame as being made up of a series

	Apples
0	3
1	7
2	5
3	9

Series

+

	Oranges
0	12
1	6
2	1
3	13

Series

=

	Apples	Oranges
0	3	12
1	7	6
2	5	1
3	9	13

DataFrame

COLLEGE OF INFORMATION STUDIES

Indexing and Slicing

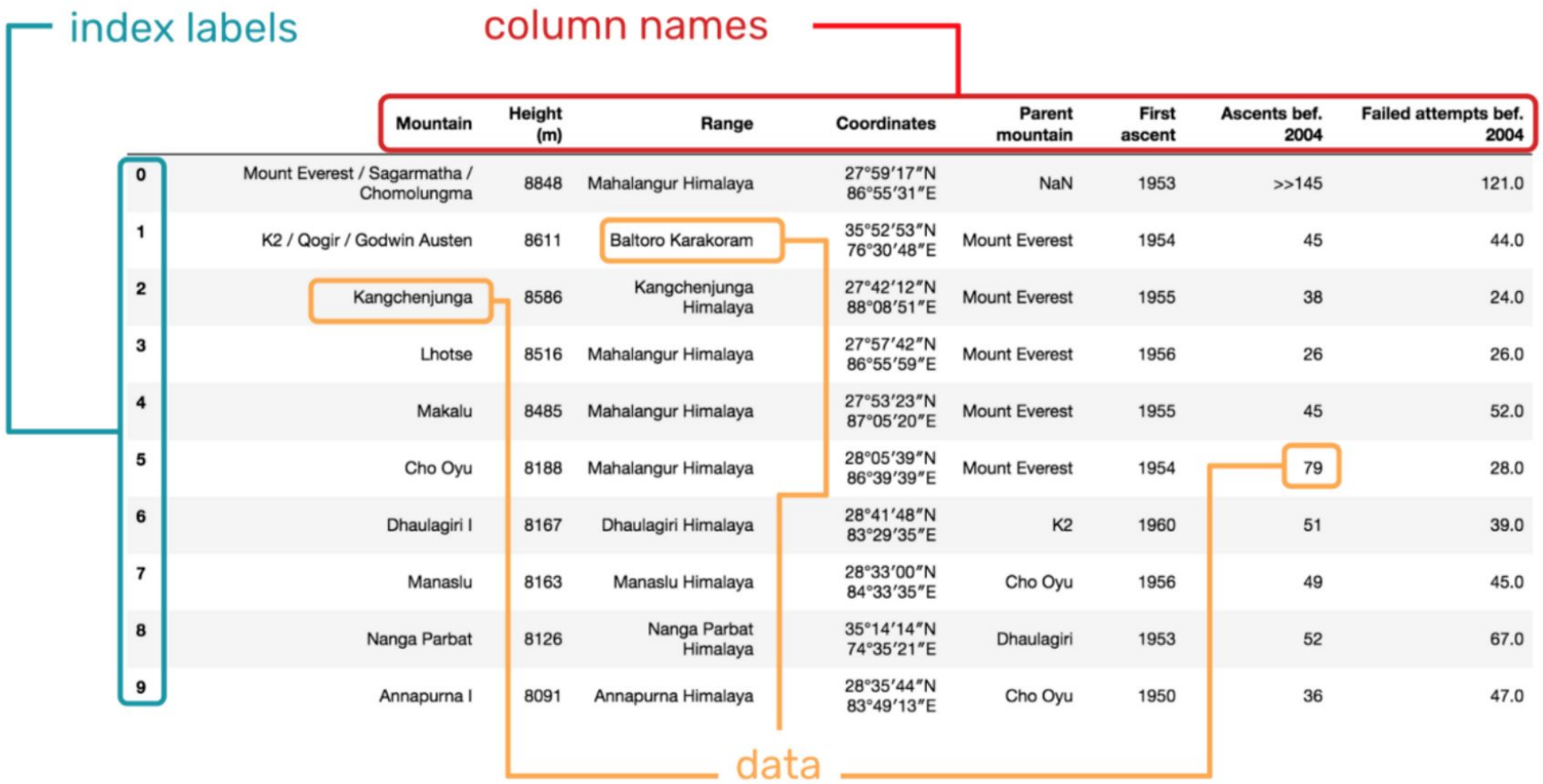
Indexing - rows

- Get first five rows
  - df[:5]
- Get rows 6-10
  - df[6:10]

Slicing - Columns

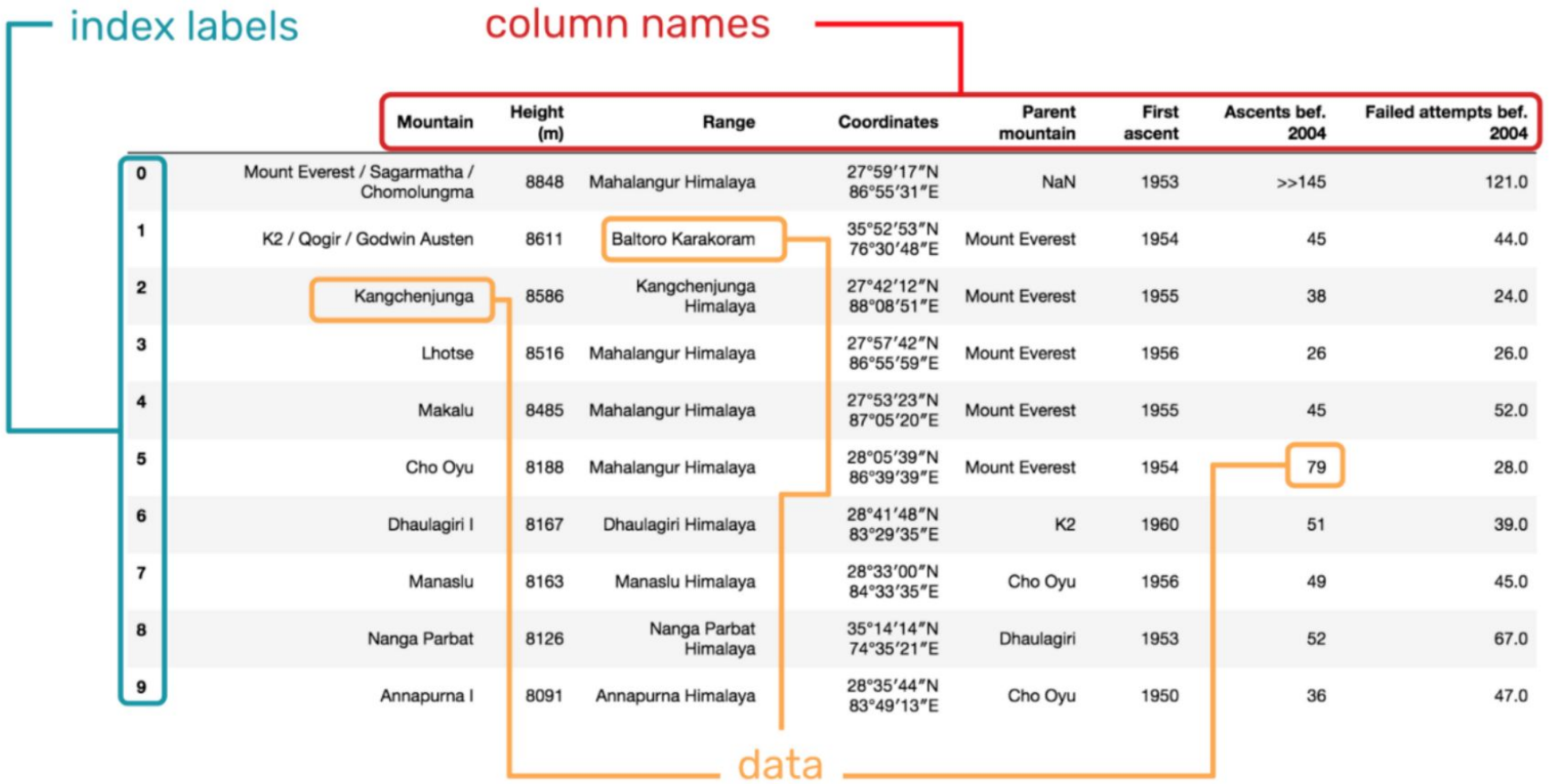
- Get a column
  - df[“Mountain”]
- Get multiple columns
  - df[[“Mountain”,”Range”]]

How can you get the year of first ascent?



	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

How can you get the year of first ascent?



	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59′17″N 86°55′31″E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52′53″N 76°30′48″E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42′12″N 88°08′51″E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57′42″N 86°55′59″E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53′23″N 87°05′20″E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05′39″N 86°39′39″E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41′48″N 83°29′35″E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33′00″N 84°33′35″E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14′14″N 74°35′21″E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35′44″N 83°49′13″E	Cho Oyu	1950	36	47.0

> df[“First ascent”]

COLLEGE OF INFORMATION STUDIES

Summarizing Data

Variables

- Variable – refers to the property of an object or event that can take on different values
  - e.g. Mountain
  - e.g. Height
  - e.g. Coordinates

How many observations?

- Count observations in Python
  - Data Set (get total number of rows)
    - df.shape (90, 41) 90 rows and 41 columns
  - Variable (get count of non-missing values)
    - df.count()

Types of Variables (Scale of Measurement)

- Categorical
  - Nominal scale - a collection of labels (e.g. Mountain “Lhotse”, “K2”, “Makalu”)
  - Ordinal scale - an ordered rank of labels (e.g. Date of first ascent “1954-09-03”, “1954-10-22”, How satisfied are you with our products. (1-very satisfied, 2-somewhat satisfied…))
- Numeric
  - Interval scale - Equal intervals represent equal differences (e.g. Year of first ascent 1953, 1954, 1955)
  - Ratio scale - Equal intervals represent equal differences and has a true zero (e.g. Heigh 8848, 8611, 8586)

Descriptive Statistics

- **Descriptive Statistics** – Ways to meaningfully show or summarize large amounts of data with only a few values.
- **Measures of Central Tendency** - Typical values for a distribution.
  - mean, median, mode
- **Measures of Variability** - The degree to which individual data points are distributed around the mean.
  - range, standard deviation, frequency distributions

Descriptive Stats for Categorical Variables

- **Central Tendency**
  - mode
- **Variability**
  - frequency distribution
- **In Python**
  - df[“variablename”].value_counts()
  - df[“variablename”].mode()

Descriptive Stats for Numeric Variables

- **Central Tendency**
  - mean, median
- **Variability**
  - range, standard deviation
- **In Python**
  - df[“variablename”].describe()
  - df[“variablename”].mean()
  - df[“variablename”].median()
  - df[“variablename”].sd()

Calculations for Numeric Variables

- **Sum in Python**
  - df[“variablename”].sum()

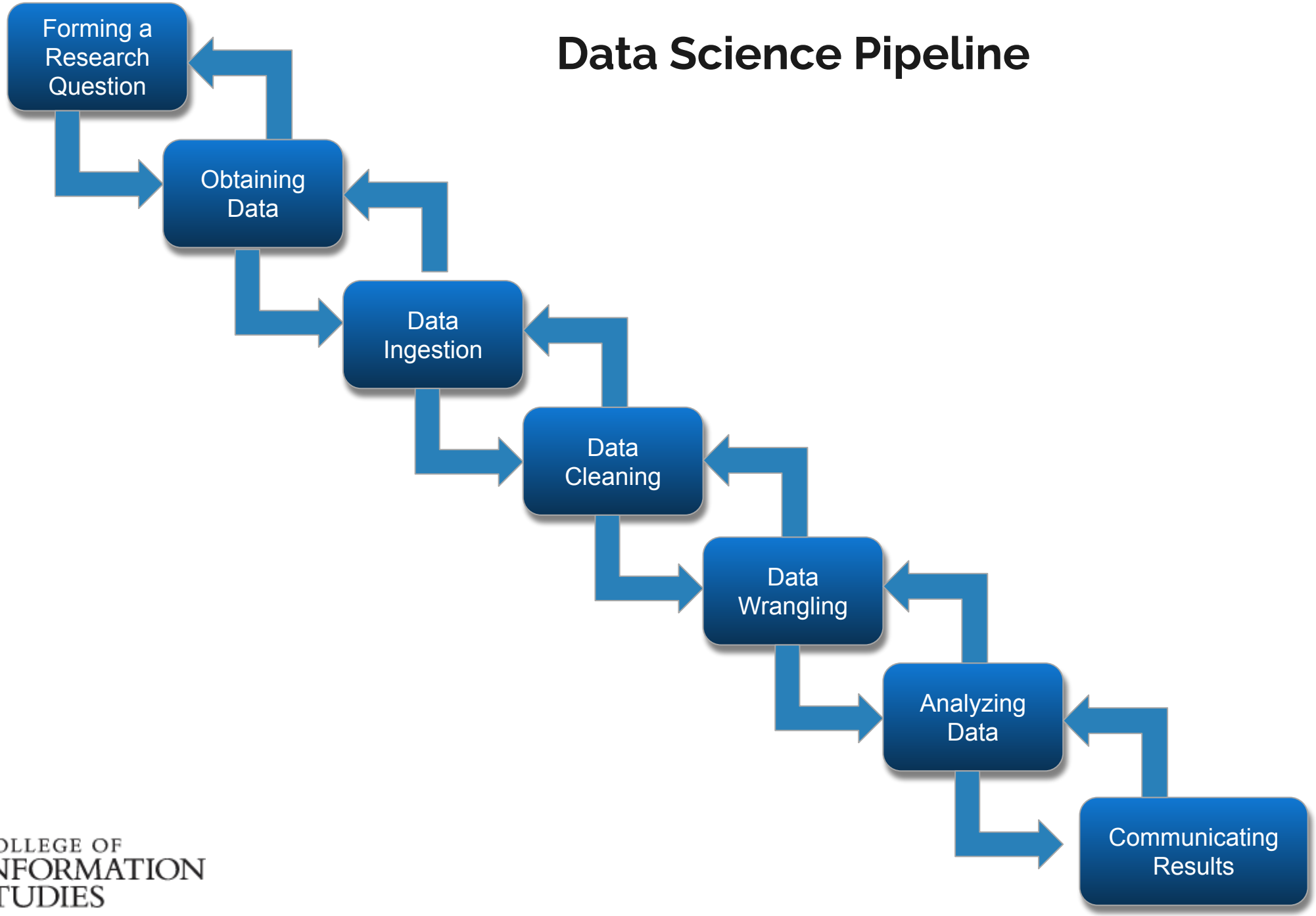
COLLEGE OF INFORMATION STUDIES

Lab

COLLEGE OF INFORMATION STUDIES

Projects

Data Science Pipeline



Projects

- Teams of 2, 3, or 4
  - Class is small so I prefer sizes of 2 or 3
  - Project proposals due 2/28, so you have time
- API Keys
  - Twitter (see my submission process)
- Scraping
  - Reddit example (json)
  - Reddit group on data sets https://www.reddit.com/r/datasets/

Data Science Projects

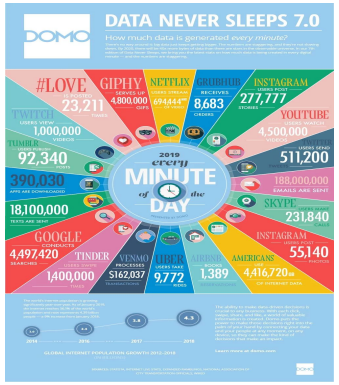
Data science - the ability to take large amounts of data in many different formats and be able to understand it, to process it, to extract value from it, to summarize it, to visualize it, and to communicate it to others.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value. -datarobot.com

FEC.gov

YAHOO! FINANCE

United States® Census Bureau



- Sentiment Analysis
- Customer Segmentation
- Recommending products
- Public Health Issues
- Manufacturing - predicting faults
- Financial Risk Analysis

Illustration from:
https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2019-version-infographic/558793/

Projects

https://engineering.salesforce.com/ & TLS Fingerprints JA3 and JA3S

Berkeley SETI Research Center

FITS file handling
https://docs.astropy.org/en/stable/io/fits/index.html

astropy:docs

COLLEGE OF INFORMATION STUDIES

Next Week

Next Week

- Project Proposal
- More Python/Pandas

I appreciate your attention

Hope to see you on Thursday!

COLLEGE OF INFORMATION STUDIES

Reference Material
Install Software

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

Software Tools

- Python & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine
  - http://openrefine.org/download.html
- Data sets
  - https://www.reddit.com/r/datasets/
  - https://opendata.dc.gov/
  - https://datasetsearch.research.google.com/
  - https://www.kaggle.com/datasets

01 Admin

02 Projects/Teams

03 Data Wrangling

---

04 Matplotlib

05 Lab

06 Next Week

This Week

Time: Tuesday virtual

- Admin
  - Syllabus Updates
- Readings
- Videos
  - Data Wrangling
- Jupyter Examples

Time: Thursday Virtual w/ optional live session

- Live session
  - Jupyter Notebooks subjects from reading
- Videos
  - Matplotlib
- Lab & Assignment
- Projects - teams

If you are tired, stand up in the back of class.

Use of phone during class for non-class purposes is rude.

COLLEGE OF INFORMATION STUDIES

Admin

Admin

- Office Hours (need to schedule a time slot):
  - Monday 8-9 pm
  - Friday 8-10 am
  - Saturday 6-8 pm (changed from am to pm)
  - Sunday 6-7 pm (changed from 4-6 to 6-7)
  - By Appointment * Anytime
- Live class meetings - Thursdays 12:30-1:30
  - Class originally scheduled to start @ 12:30 so I figure this is a good time
  - We can add a couple of these at different times if/when needed
- Piazza vs. Canvas discussions

INST447 General Schedule

- Update 9/15/2020

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					8-10am	
		Noon-ish		12:30-1:30pm		
6-7pm	8-9 pm					
	Lab 11:59pm					6-8pm

Office Hours Live

Office Hours By appointment

Video Ready

Class Live Sessions

Lab Due

Syllabus Updates

- General syllabus schedule still applies
- Contains fairly set schedule….some adjustments may be made

Project Teams

- Project Teams set up in Canvas
- If you have teams, please sign up
- First one to sign up is the team ‘captain’
- Team contract will be due
- Project Proposal due 10/8

COLLEGE OF INFORMATION STUDIES

Data Wrangling

See Notebook Examples

Slides adapted from N Sigalo and Yla T.

COLLEGE OF INFORMATION STUDIES

Data Wrangling

Aggregating

Slides adapted from Nikki Sigalo and Yla Tausczik

Unit of Analysis

One of the most important ideas in a research project is the *unit of analysis*. The unit of analysis is the major entity that you are analyzing in your study. For instance, any of the following could be a unit of analysis in a study:

- individuals
- an employee record
- artifacts (books, photos, newspapers)
- geographical units (town, census tract, state)
- social interactions (divorces, arrests)

Unit of Analysis - example

- **Unit of Analysis of Data Set** – An entity in your data set.

  e.g. a single capital bikeshare bicycle trip

- **Desired of Unit of Analysis** – The entity that you want to perform your analysis on.

  e.g. a capital bikeshare user

What is the Unit of Analysis?

User Id	Mean Num. Rides Per Day	Mean Ride Length
388478	10.2	5
120760	0.5	92
131227	8.2	23
426446	5.9	34

A capital bikeshare user record

What is the Unit of Analysis?

Date	Day of Week	Num Rides	Num Riders
2015-01-01	Thursday	3064	1446
2015-01-02	Friday	1998	1532
2015-01-03	Saturday	3089	1516
2015-01-04	Sunday	2992	1232

Record of bike rides on a specific date

What Unit of Analysis do you need?

1. Which are the most popular bikeshare stations?

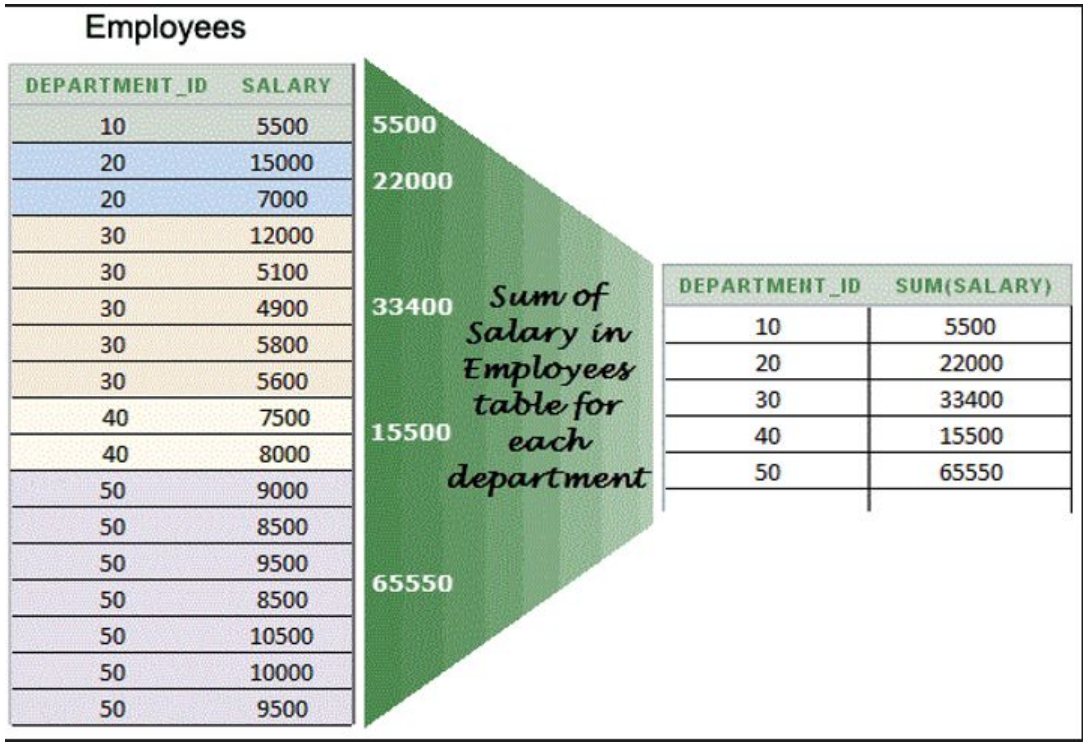
2. Do people take fewer rides when it is raining?

3. Who takes more bike rides per day tourists or residents?

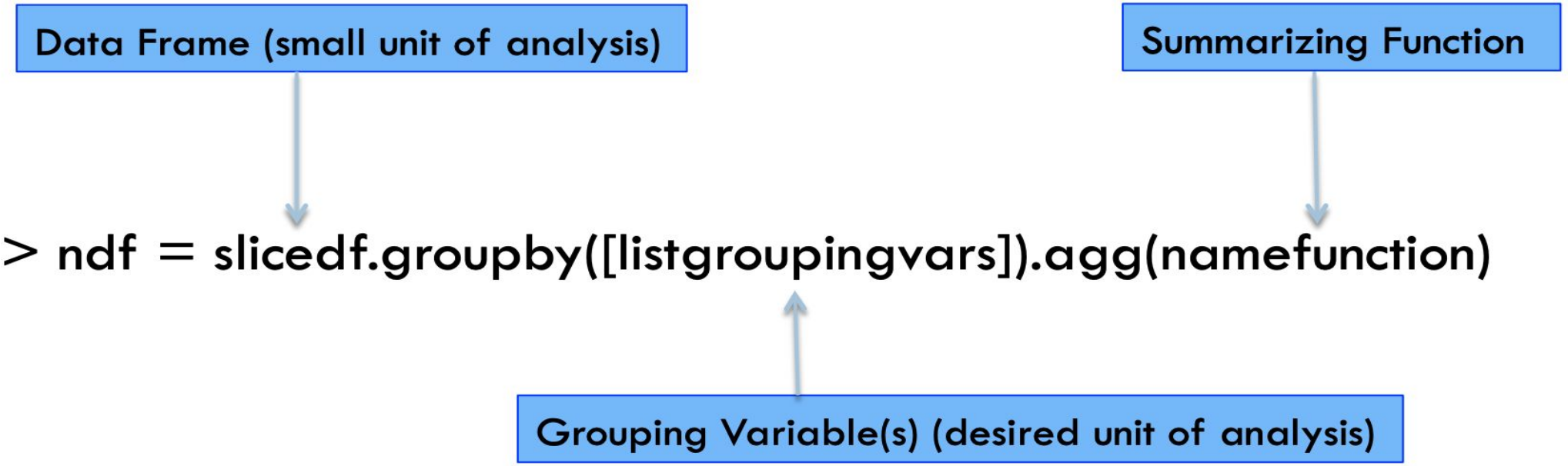
Aggregations

Aggregation - Transformation from a smaller unit analysis to a larger unit of analysis.

Employee record to department record



GroupBy

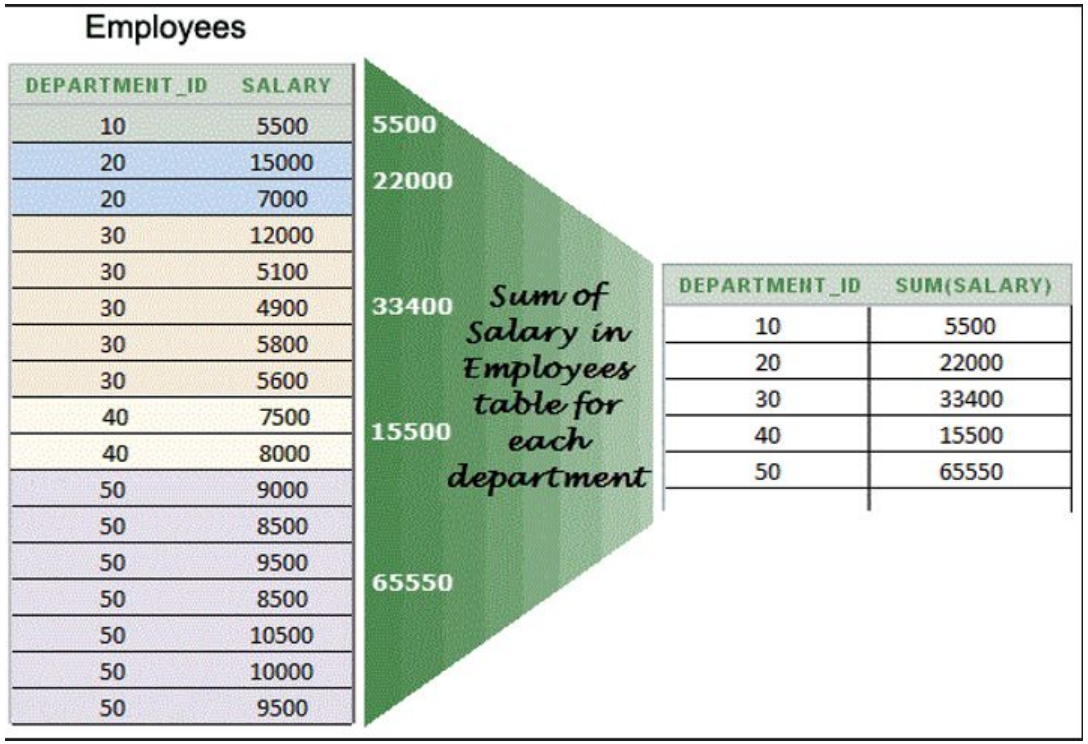


Example:

> stations = trips[["Station","Ride Id"]].groupby(["Station"]).agg("count")

Anatomy of Aggregation

- **Grouping variable(s)**– The variables with unique identification for the unit of analysis you want to perform your analysis on.
  - e.g. department_id
- **Variable(s) to summarize** – What variable(s) do you want to summarize at the new unit of analysis?
  - e.g. salary
- **Summarization function** – How do you want to summarize these variables?
  - e.g. sum



Common Summarizing Functions

- Function to summarize the smaller unit of analysis to create an aggregate value.

- Example Functions
  - Sum - the total sum of salaries per department
  - Mean number of rides per day per user: “mean”
  - Total number of rides per date: “count”

Fixing names after aggregating

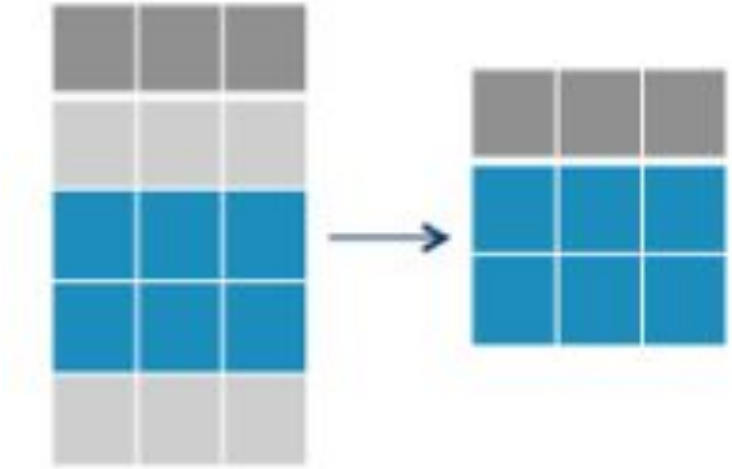
- By default, groupby moves the group by variables from columns to index labels
  - Use reset_index to move index labels to columns
- groupby function does not rename columns that have been summarized. They keep their old name, which might not make sense.
  - rename the columns
- Jupyter Notebook Example - Fixing names after aggregating

COLLEGE OF INFORMATION STUDIES

Subsetting & Merging

Subsetting

- Selecting only some of the rows of your data frame
  - Select only orders of organic produce
  - Select only bike rides from tourists



Boolean Expression

Examples:

Is equal
- df[“variable”] == “value”

Is less than
- df[“variable”] < 100

Is more than
- df[“variable1”] > df[“variable2”]

Is in a set
- df[“variable”].isin([“value1”,”value2”,”value3”])

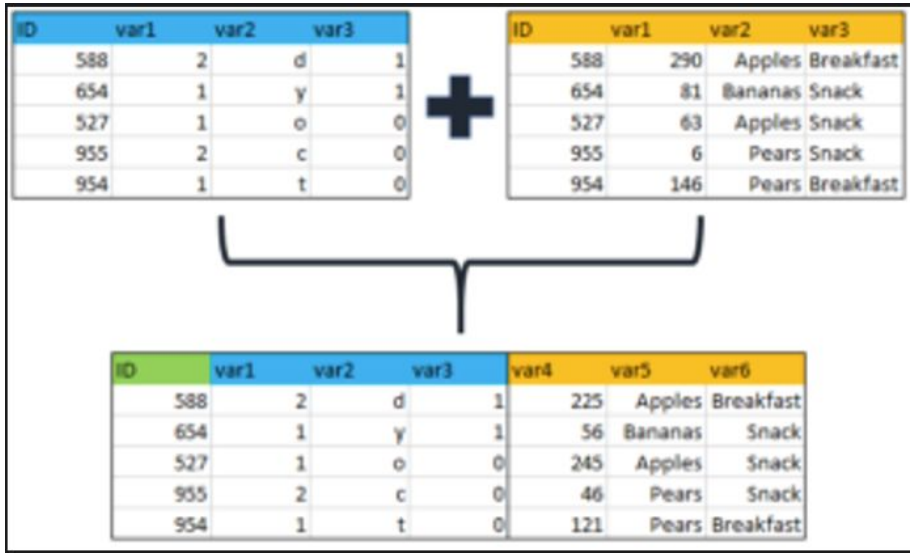
Jupyter Notebook Examples
- Select only orders of organic produce
- Select only bike rides from tourists

Combining Data Sets

- Combine multiple data sets into one data frame
  - e.g. weather data + bikeshare trip data
  - e.g. Uber data + Lyft data + Taxi data
- Typical Purpose
  - Pull information from multiple sources of data
- Functions
  - Combine different columns (join): merge
  - Combine different rows (append): concat

Merge Function (like JOIN MySQL)

- On:
  - Variables that are common to both data sets
- How: How to handle missing rows
  - inner
  - outer
  - left
  - right
- Suffixes:
  - Suffix to distinguish same variable name but different meaning



ID	var1	var2	var3
588	2	d	1
654	1	y	1
527	1	o	0
955	2	c	0
954	1	t	0

ID	var1	var2	var3
588	290	Apples	Breakfast
654	81	Bananas	Snack
527	63	Apples	Snack
955	6	Pears	Snack
954	146	Pears	Breakfast

ID	var1	var2	var3	var4	var5	var6
588	2	d	1	225	Apples	Breakfast
654	1	y	1	56	Bananas	Snack
527	1	o	0	245	Apples	Snack
955	2	c	0	46	Pears	Snack
954	1	t	0	121	Pears	Breakfast

Merge Function

> ndf = pd.merge(df1,df2,how="howtype",on=[mergevars],suffixes=[newvarnames])

Example:

> ndf = pd.merge(weather,trips,how="inner",on=["Date"],suffixes=["_wthr","_trips"])

Trip Data

Date	NumRides
2018-06-15	24
2018-06-16	3

Weather Data

Date	Precipitation
2018-06-15	.9
2018-06-16	0

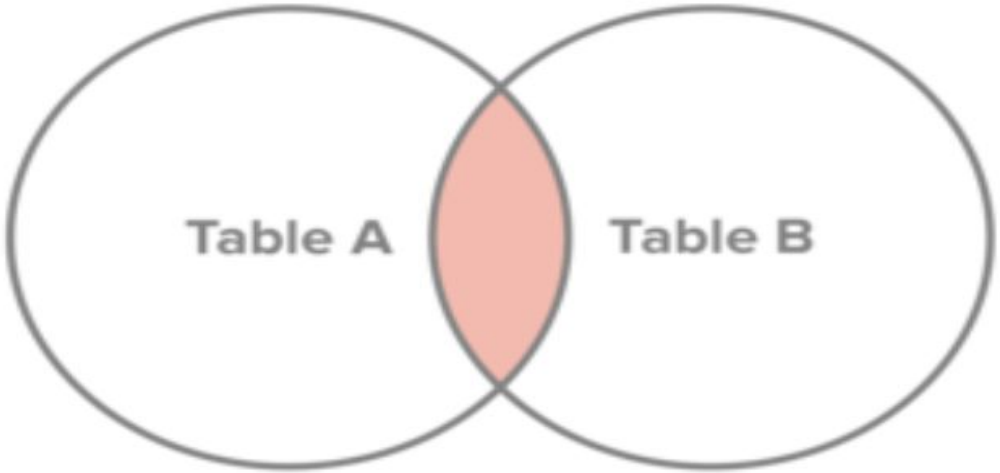
Merge Function - How

How – When there are a different set of observations in the two data frames do you want to include entries for rows missing from the left data frame? from the right data frame? both? neither?

Date	NumRides
2018-06-15	24
2018-06-16	3
2018-06-17	5
2018-06-19	11

Date	Precipitation
2018-06-15	.9
2018-06-16	0
2018-06-17	0
2018-06-18	0

Inner: Include only rows in both data frames

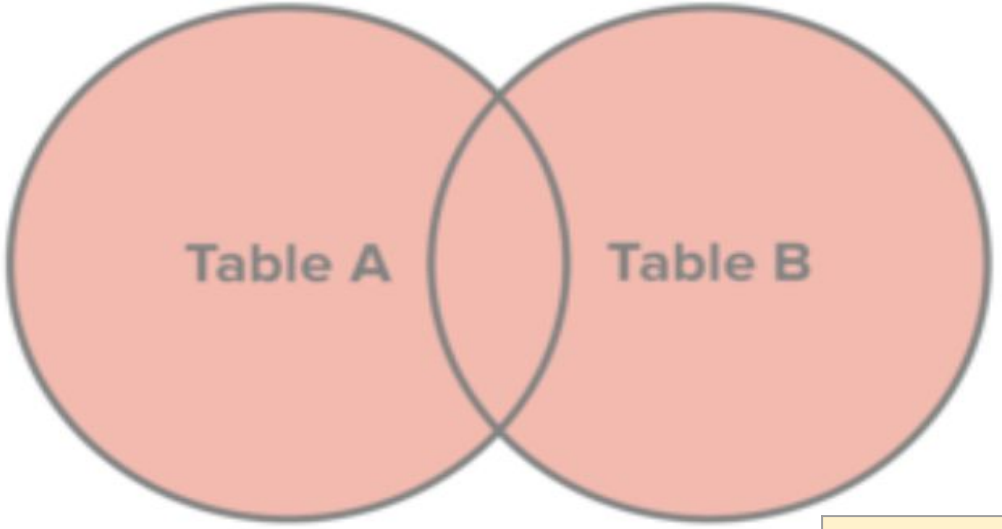


Date	NumRides
2018-06-15	24
2018-06-16	3
2018-06-17	5
2018-06-19	11

Date	Precipitation
2018-06-15	.9
2018-06-16	0
2018-06-17	0
2018-06-18	0

Date	NumRides	Precipitation
2018-06-15	24	.9
2018-06-16	3	0
2018-06-17	5	0

Outer: Include all rows in left and right data frames

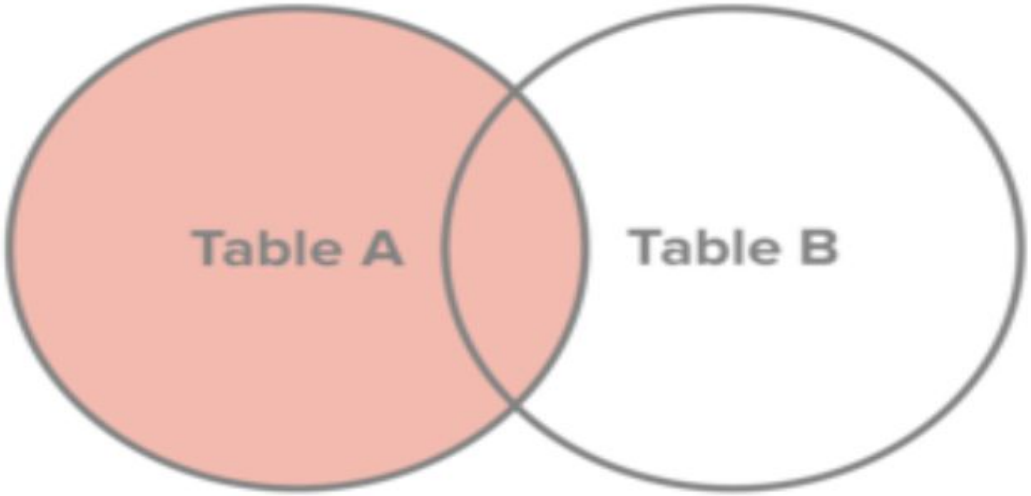


Date	NumRides
2018-06-15	24
2018-06-16	3
2018-06-17	5
2018-06-19	11

Date	Precipitation
2018-06-15	.9
2018-06-16	0
2018-06-17	0
2018-06-18	0

Date	NumRides	Precipitation
2018-06-15	24	.9
2018-06-16	3	0
2018-06-17	5	0
2018-06-18		0
2018-06-19	11	

Left: Include all rows from the left data frame

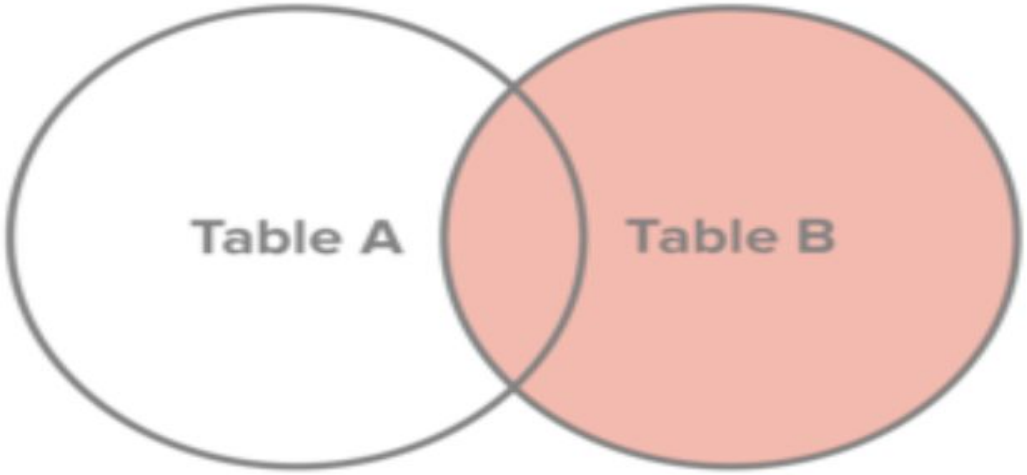


Date	NumRides
2018-06-15	24
2018-06-16	3
2018-06-17	5
2018-06-19	11

Date	Precipitation
2018-06-15	.9
2018-06-16	0
2018-06-17	0
2018-06-18	0

Date	NumRides	Precipitation
2018-06-15	24	.9
2018-06-16	3	0
2018-06-17	5	0
2018-06-19	11	

Right: Include all rows from the right data frame



Date	NumRides
2018-06-15	24
2018-06-16	3
2018-06-17	5
2018-06-19	11

Date	Precipitation
2018-06-15	.9
2018-06-16	0
2018-06-17	0
2018-06-18	0

Date	NumRides	Precipitation
2018-06-15	24	.9
2018-06-16	3	0
2018-06-17	5	0
2018-06-18		0

Concat Function

Example:

- Uber data + Lyft data

Combine rows to get a data set of rideshare trips.

Date	Ride Length	Rideshare Type
2018-06-15	24	Uber
2018-06-15	3	Uber

Date	Ride Length	Rideshare Type
2018-06-15	17	Lyft
2018-06-16	11	Lyft

Concat Function

Data frames must have the same columns

> ndf = pd.concat([df1,df2])

> rideshareTrips = pd.concat([uberdf,lyftdf])

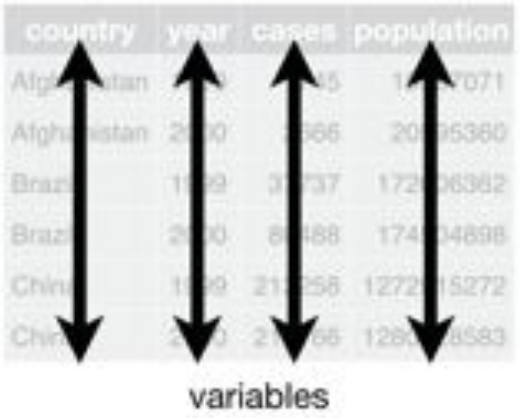
Date	Ride Length	Rideshare Type
2018-06-15	24	Uber
2018-06-15	3	Uber
2018-06-15	17	Lyft
2018-06-16	11	Lyft

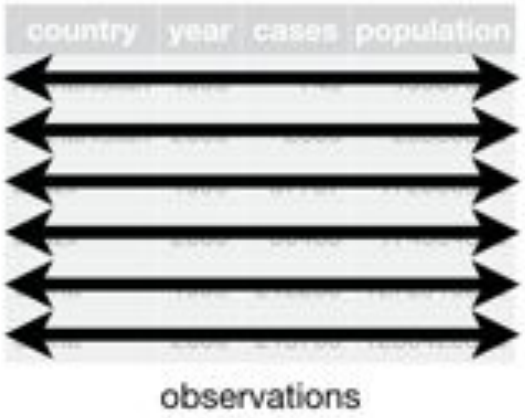
Reshaping

Tidy Data (Hadley Wickham)

Tidy data is organized such that:

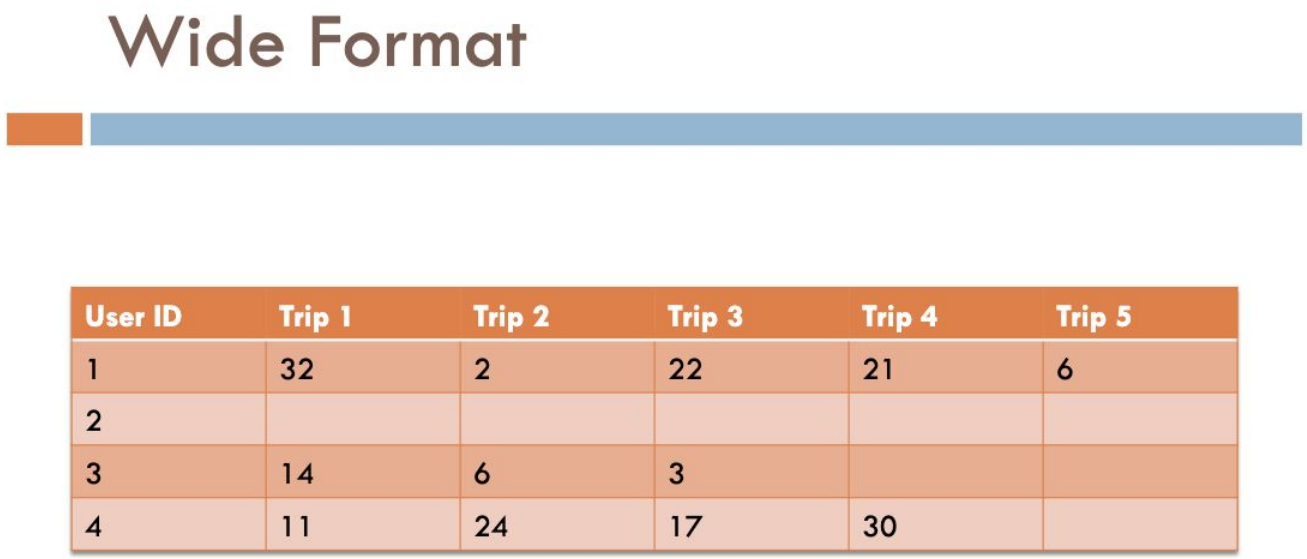
- Each variable has its own column
- Each observation has its own row
- Each value has its own cell





Wide Format

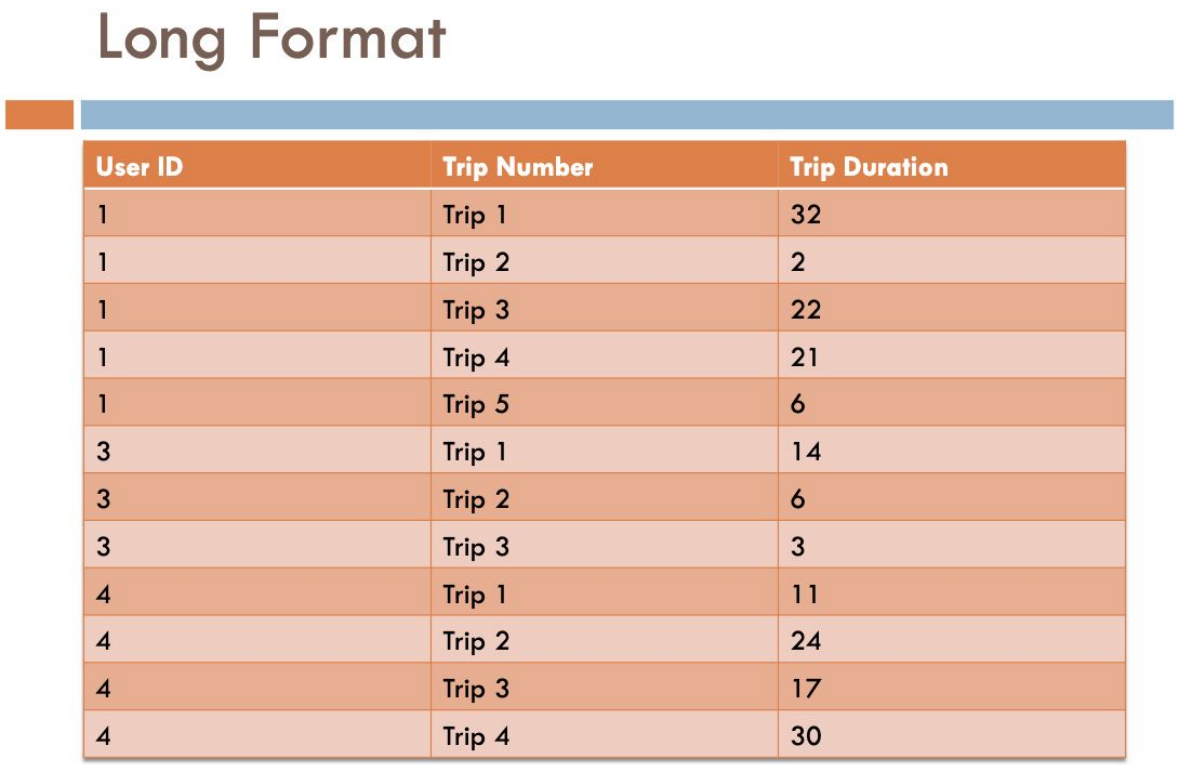
- The same variable is spread across multiple columns
  - e.g. Trip duration
- Multiple observations are collapsed into one row
  - e.g. each row represents data from multiple trips



User ID	Trip 1	Trip 2	Trip 3	Trip 4	Trip 5
1	32	2	22	21	6
2					
3	14	6	3		
4	11	24	17	30	

Long Format

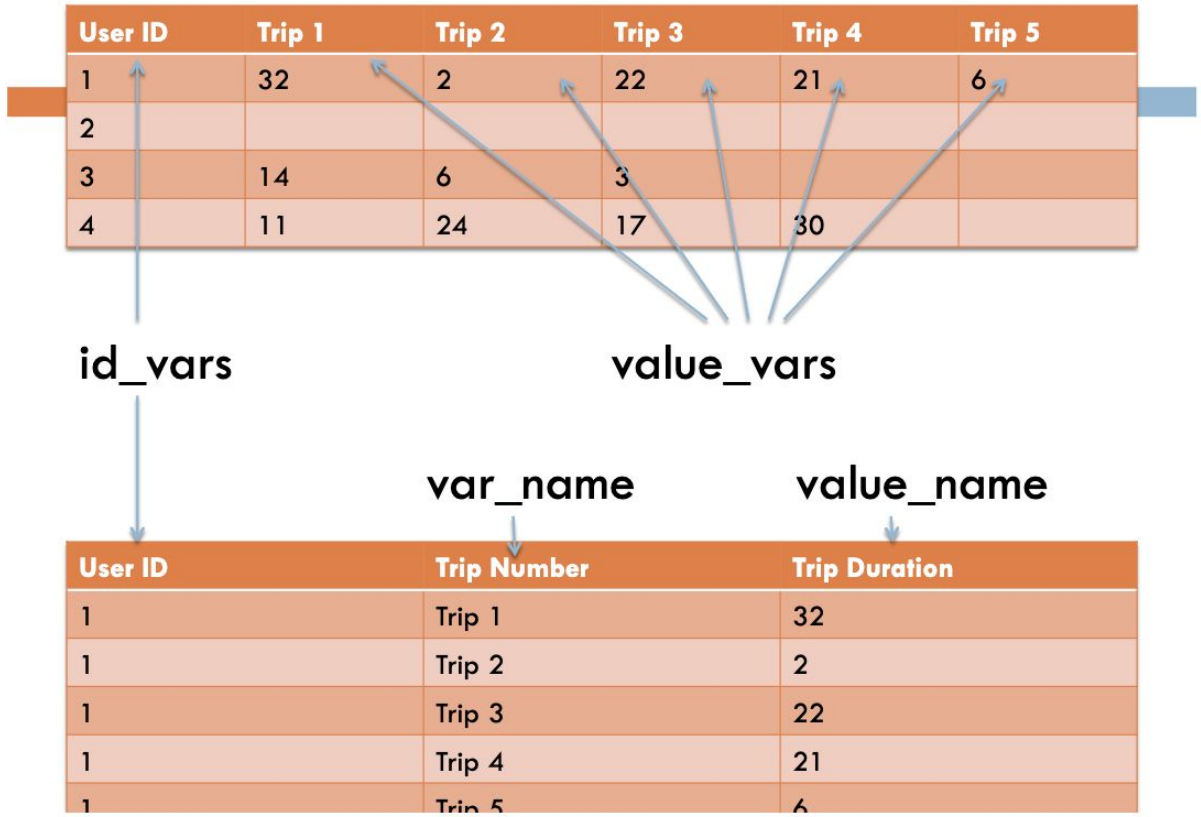
- Each row is one time point per user (not grouped)



User ID	Trip Number	Trip Duration
1	Trip 1	32
1	Trip 2	2
1	Trip 3	22
1	Trip 4	21
1	Trip 5	6
3	Trip 1	14
3	Trip 2	6
3	Trip 3	3
4	Trip 1	11
4	Trip 2	24
4	Trip 3	17
4	Trip 4	30

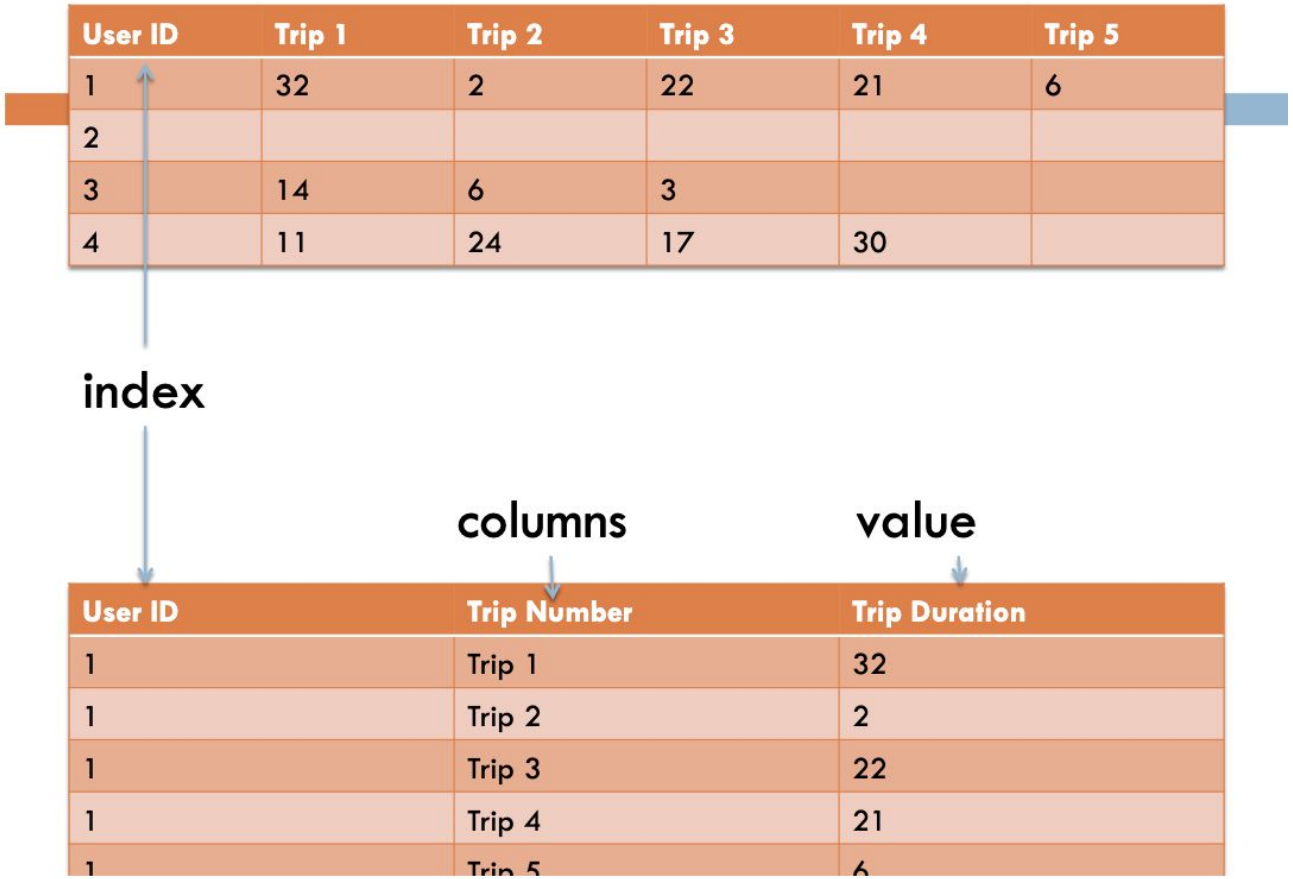
Converting wide to long

- Use the melt function, along with some other specifications:
  - pd.melt(df, id_vars, value_vars, var_name, value_name)
- id_vars: variable that identifies a row in wide format
- value_vars: variables in wide format that contain data to put in one variable in long format
- var_name: new name for column that will have labels for data
- value_name: new name for column that has data in long format



Converting long to wide

- df.pivot(index, columns, values)
- Use the pivot function
- index: variable that identifies a row in wide format (id_var)
- columns: variable that contains labels for data (var_name)
- values: variable that contains data (value_name)



Computing new variables

- Function on one or more variables to generate a new variable
  - e.g. convert scores from 0 to 100 to letter grades
  - e.g. compute trip duration from start and end timestamps
- Functions
  - vector functions
  - apply

apply, lambda, mean

- scores – data frame with student quiz grades
- q1,q2,q3 – variable names for quiz scores

User ID	q1	q2	q3
1	91	95	86
2	70	89	77
3	85	85	92
4	88	80	87
5	0	0	20

apply, lambda, mean

Calculate average quiz score

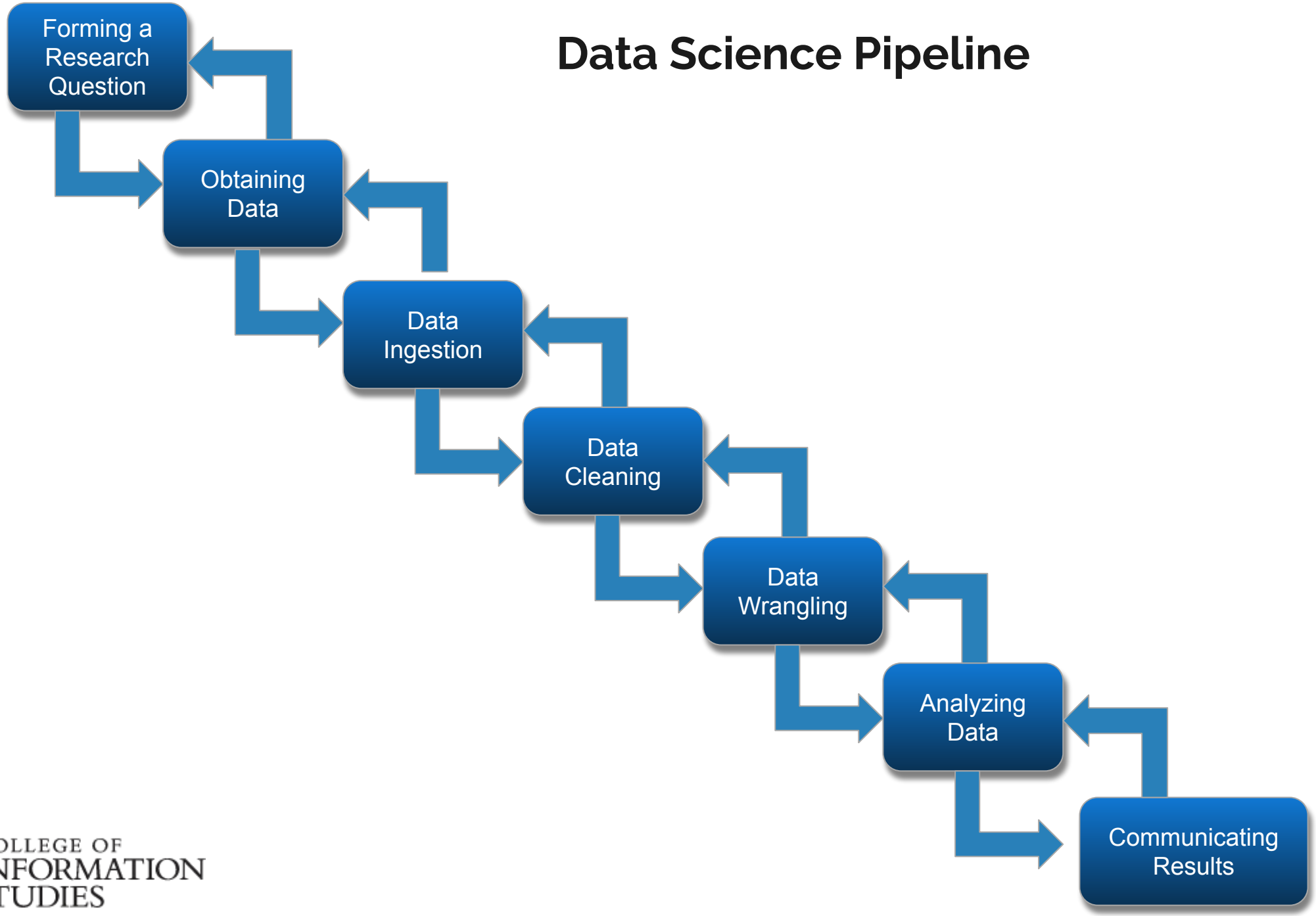
```
scores["avgquiz"] = (scores["q1"]+scores["q2"]+scores["q3"])/3

scores["avgquiz"] = scores[["q1","q2","q3"]].apply(np.mean,1)

scores["avgquiz"] = scores[["q1","q2","q3"]].apply(lambda x:sum(x)/len(x),1)

scores["avgquiz"] = scores[["q1","q2","q3"]].mean(1)
```

Lab

COLLEGE OF INFORMATION STUDIES

Projects

Data Science Pipeline



Projects

- Teams of 2, 3, or 4
  - Select teams
- API Keys
  - Twitter (see my submission process)
- Scraping
  - Reddit example (json)
  - Reddit group on data sets https://www.reddit.com/r/datasets/

Data Science Projects

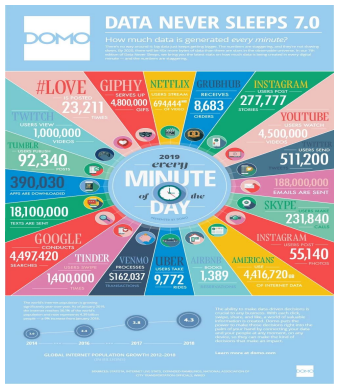
Data science - the ability to take large amounts of data in many different formats and be able to understand it, to process it, to extract value from it, to summarize it, to visualize it, and to communicate it to others.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of math and statistics to extract meaningful insights from data. Data science practitioners apply machine learning algorithms to numbers, text, images, video, audio, and more to produce artificial intelligence (AI) systems that perform tasks which ordinarily require human intelligence. In turn, these systems generate insights that analysts and business users translate into tangible business value. -datarobot.com

FEC.gov

YAHOO! FINANCE

United States® Census Bureau



- Sentiment Analysis
- Customer Segmentation
- Recommending products
- Public Health Issues
- Manufacturing - predicting faults
- Financial Risk Analysis

Illustration from:
https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2019-version-infographic/558793/

Projects

https://engineering.salesforce.com/ & TLS Fingerprints JA3 and JA3S

Berkeley SETI Research Center

FITS file handling
https://docs.astropy.org/en/stable/io/fits/index.html

astropy:docs

COLLEGE OF INFORMATION STUDIES

Next Week

Next Week

- Project Proposal
- Data structures

I appreciate your attention

Hope to see you on Thursday!

COLLEGE OF INFORMATION STUDIES

Reference Material
Install Software

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

Software Tools

- Python & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine
  - http://openrefine.org/download.html
- Data sets
  - https://www.reddit.com/r/datasets/
  - https://opendata.dc.gov/
  - https://datasetsearch.research.google.com/
  - https://www.kaggle.com/datasets

01 Admin

02 XML

03 JSON

04 SqlLite

05 Project/Lab

06 Next Class

This Week

Time: Tuesday virtual

- Admin
  - Syllabus Updates
  - Grades
  - Midterm
- Metadata
- XML
- Json
- SQL -Sqlite

Time: Thursday Virtual w/ optional live session

- Live session
  - Jupyter Notebooks subjects from reading
- Videos
  - Data structures
- Lab & Midterm & Review (Next week)
- Projects - teams

If you are tired, stand up in the back of class.

Use of phone during class for non-class purposes is rude.

COLLEGE OF INFORMATION STUDIES

Admin

Admin

- Office Hours (need to schedule a time slot):
  - Monday 8-9 pm
  - Friday 8-10 am
  - Saturday 6-8 pm (changed from am to pm)
  - Sunday 6-7 pm (changed from 4-6 to 6-7)
  - By Appointment * Anytime
- Live class meetings - Thursdays 12:30-1:30
  - Class originally scheduled to start @ 12:30 so I figure this is a good time
  - We can add a couple of these at different times if/when needed
- Midterm Next week and Review Session

INST447 General Schedule

- Update 9/15/2020

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
					8-10am	
		Noon-ish		12:30-1:30pm		
6-7pm	8-9 pm					
	Lab 11:59pm					6-8pm

Office Hours Live

Office Hours By appointment

Video Ready

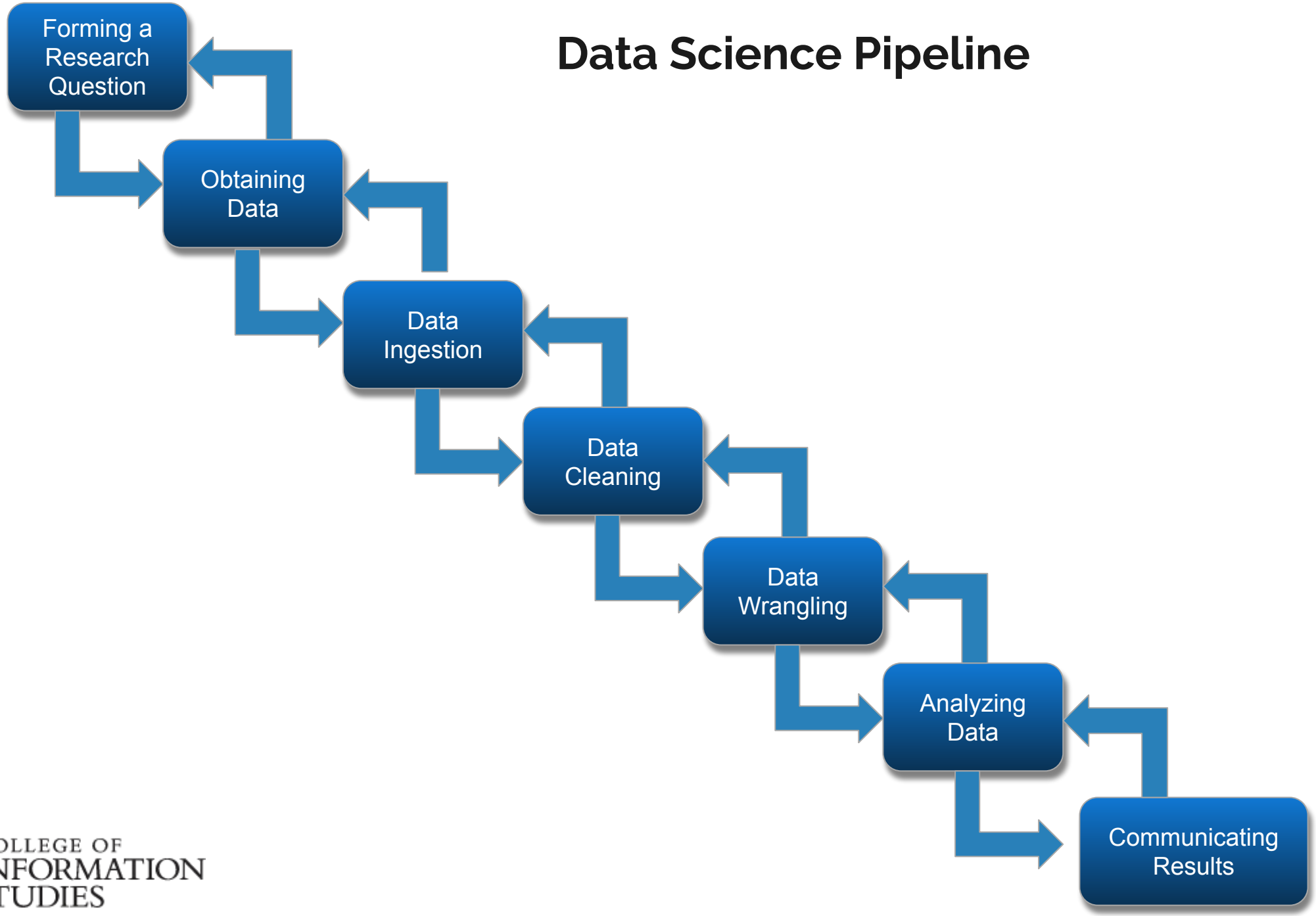
Class Live Sessions

Lab Due

Project Teams

- Project Teams set up in Canvas
- If you have teams, please sign up
- First one to sign up is the team ‘captain’
- Team contract will be due
- Project Proposal due 10/8

Data Science Pipeline



COLLEGE OF INFORMATION STUDIES

Metadata

Metadata

- What is Metadata?
    - Metadata is structured information describing a resource, for example: the dates, title, and creators associated with a dataset.
    - Metadata is "data that provides information about other data". In other words, it is "data about data."
    - More precisely, metadata describes data containing specific information like type, length, textual description and other characteristics.
        - Examples
            - Date of ingest, Date of update
            - Collection method (API, manual, other)
            - Processing servers

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata

Types of Metadata

Metadata falls into **three** main categories:

1. Descriptive: or used for discovery and identification and including such information as title, author, abstract and keywords.
2. Structural: which shows how information is put together – page order to chapters, for example.
    a. **Technical** metadata properties include file types, size, creation date and time, and type of compression. Technical metadata is often used for digital object management and interoperability.
    b. **Preservation** metadata is used in navigation. Example preservation metadata properties include an item's place in a hierarchy or sequence.
3. Administrative: which enables better resource management by showing such information as when and how the resource was created.
    a. **Rights** metadata might include copyright status, rights holder, or license terms.

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata
https://www.lifewire.com/metadata-definition-and-examples-1019177
https://guides.ucf.edu/metadata/intrometadata

Purposes of Metadata

- Resource discovery is one of the most common
  - Effective cataloging
    - Identifying resources.
    - Defining them by criteria.
    - Bringing similar resources together.
    - Distinguishing among those that are dissimilar.
- Effective means of organizing electronic resources
  - Typically, links to resources have been organized as lists and built as static webpages, with the names and resources hardcoded in HTML. A more efficient practice, is to use metadata to build these pages. For Web purposes, the information can be extracted and reformatted through use of software tools.
-

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata

Purposes of Metadata

- Facilitates interoperability and integration of resources.
  - Using metadata to describe resources enables its understanding by humans as well as machines.
- Facilitates digital identification via standard numbers that uniquely identify the resource the metadata defines.
  - A practice is to combine metadata so that it acts as a set of identifying data that differentiate objects or resources, supporting validation needs.
- Metadata is an important way to protect resources and their future accessibility.
  - Lineage: For archiving and preservation purposes, it takes metadata elements that track the object's lineage, and describe its physical characteristics and behavior so it can be replicated on technologies in the future.

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata

Metadata

- Establishing a metadata workflow that sufficiently describes your data is an important part of data management.
- ***** Metadata planning should occur at the beginning of a research project. *****
- **Metadata isn't the data itself!!**

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata

Metadata Standards

- There are a number of metadata standards for datasets.
  - Society for American Archivists
    - https://www2.archivists.org/
    - https://www2.archivists.org/sites/all/files/national_library_of_medicine.pdf
    - https://www2.archivists.org/sites/all/files/UniversityofLouisvilleCDMexportHFox.txt
  - Common Standards (many, one example)
    - https://guides.ucf.edu/c.php?g=79118&p=506121
      - Dublin Core : https://www.dublincore.org/specifications/dublin-core/dces/ -
        - The Dublin Core™ Metadata Element Set is a vocabulary of fifteen properties for use in resource description. The name "Dublin" is due to its origin at a 1995 invitational workshop in Dublin, Ohio; "core" because its elements are broad and generic, usable for describing a wide range of resources.

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata
https://www.lifewire.com/annot-file-2619632

Metadata Examples

- **Metadata and Website Searches**
  - The metadata embedded in websites is critically important to the success of the site. It includes a description of the website, keywords, metatags, and more.
  - Some common metadata terms used when building a web page include meta title and meta description.
    - The meta title briefly explains the topic of the page to help readers understand what they'll get from the page should they open it.
    - The meta description is further information, though brief, about the contents of the page.
    - Both of these metadata pieces are displayed on search engines
    - A web page's metadata might also include the language the page was written in, like whether it's an HTML page or the actual language.
- **Metadata for Tracking**
  - Retailers and online shopping sites use metadata to track consumers' habits and movements.
  - ISPs, governments, and anyone else with access to large collections of metadata information could potentially use the metadata from web pages, emails, and other places there are users online, to monitor web activity.

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata
https://www.lifewire.com/annot-file-2619632

Metadata Examples

- **Metadata in Social Media**
  - Geotagged images
  - You can see a profile image and a short description of the Facebook user to learn just the basics about them before deciding to friend them or send them a message

https://www.villanovau.com/resources/bi/metadata-importance-in-data-driven-world/
https://www.lib.ncsu.edu/do/data-management/metadata
https://www.lifewire.com/annot-file-2619632

XML

XML

- **eXtensible Markup Language**
  - One of the two common formats that we use when exchanging data across the web. (JSON)
  - XML looks very similar to HTML, but XML is more structured than HTML. Here is a sample of an XML document.
  - It was designed to store and transport data. It was designed to be both human- and machine-readable.
- Each pair of opening (e.g., `<person>`) and closing tags (e.g., `</person>`) represents a ***element*** or ***node*** with the same name as the tag (e.g., `person`).
- Each element can have:
  - Text
  - Attributes (e.g., `hide`)
  - Other nested elements.

- If an XML element is empty (i.e., has no content), then it may be depicted by a self-closing tag (e.g., `<email />`).

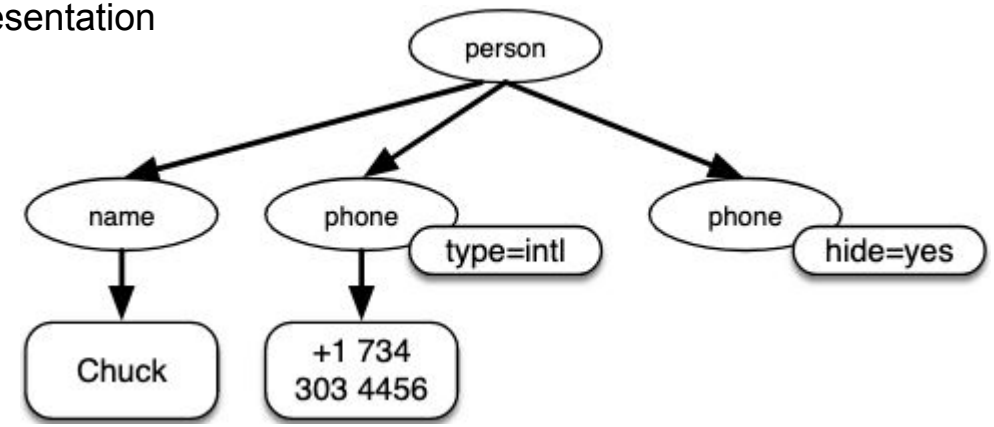
```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```

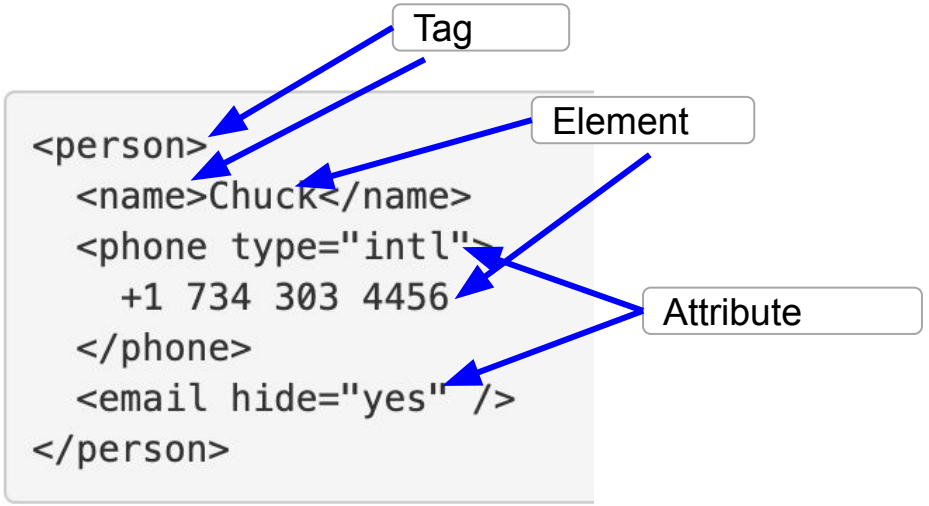

https://www.py4e.com/html3/13-web

XML

- Often it is helpful to think of an XML document as a tree structure where there is a top element (here: `person`), and other tags (e.g., `phone`) are drawn as *children* of their *parent* elements.

Representation





https://www.py4e.com/html3/13-web

XML

Tags: enclosure and label type of element

- Enclose each observation and each value for a variable.
  - Start tag e.g. <person>
  - End tag e.g. </person>
- Give variable names e.g. name, phone

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```


https://www.py4e.com/html3/13-web

XML

- Elements: tags + content between tags
  - provide data about an observation e.g. person information
  - provide data value for variable e.g. Chuck

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```


https://www.py4e.com/html3/13-web

XML

- Attributes: modify tag
  - define additional attributes of a variable e.g. type="intl"

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```


https://www.py4e.com/html3/13-web

XML

- Sequence of elements:
  - observations appear one after each other enclosed in tags e.g. Chuck info Roger info
  - <person>chuck..</person>
  - <person>bob..</person>

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```


https://www.py4e.com/html3/13-web

XML-Document Type Definition (DTD)

- Specifies valid tags for XML

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

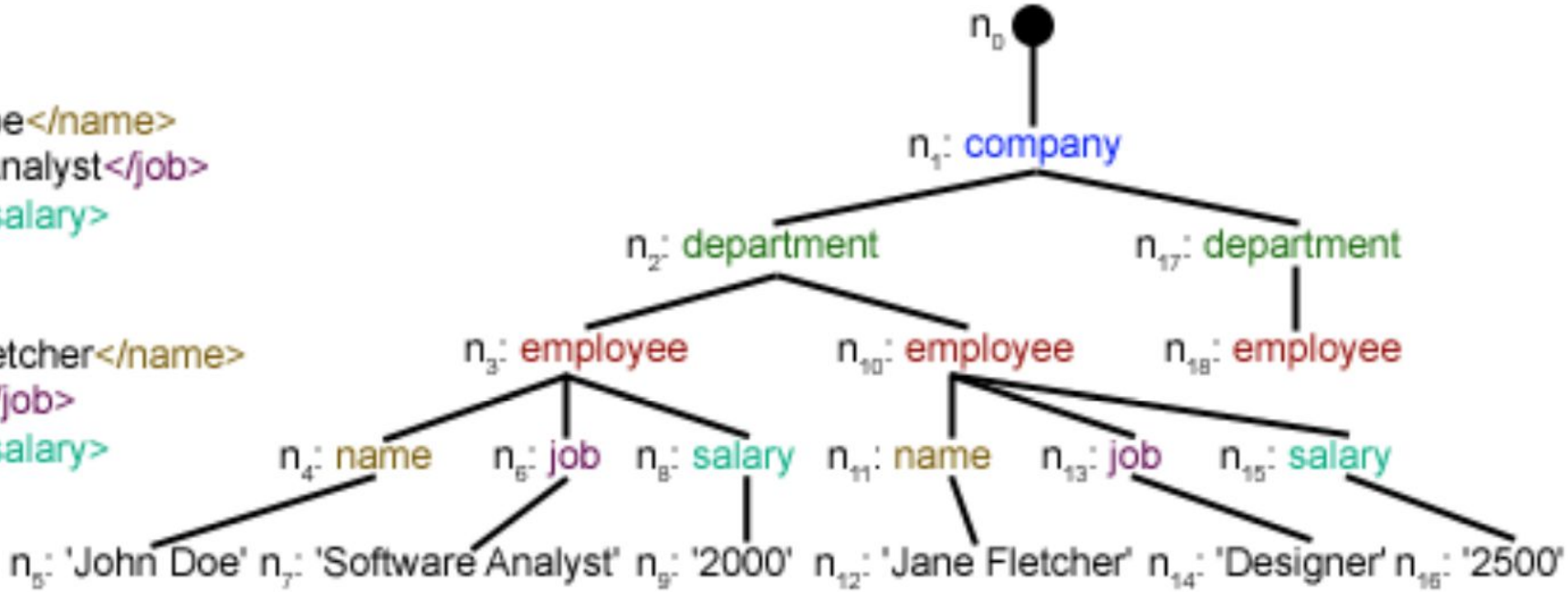
DTD

Defines a tag used in XML

https://www.py4e.com/html3/13-web

XML-Element Tree

```
<company>
 <department>
  <employee>
   <name>John Doe</name>
   <job>Software Analyst</job>
   <salary>2000</salary>
  </employee>
  <employee>
   <name>Jane Fletcher</name>
   <job>Designer</job>
   <salary>2500</salary>
  </employee>
 </department>
 <department>
  <employee>
  </employee>
 </department>
</company>
```



XML-Element Tree Functions

Read in XML as Element Tree

> data_etree = etree.parse(‘filename.xml’)

Search for specific elements by tag name

> listelements = data_etree.findall(“phone”)

XML-Element Tree Elements

Specific Element – A python object which holds information about the tag, content between tags, and attributes.

e.g. phoneelem =’<phone type =”intl”>+1 734 303 4456 </phone>’

Get tag

> phoneelem.tag

phone

Get content

> phoneelem.text

+1 734 303 4456

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```

XML-Element Tree Navigation

Get parent: gets specific element that encloses the element

> personelem = phoneelem.getparent()

Get children: gets all elements enclosed by that element

> listchildren = personelem.getchildren()

```
<person>
  <name>Chuck</name>
  <phone type="intl">
    +1 734 303 4456
  </phone>
  <email hide="yes" />
</person>
```

Parsing XML

- https://www.w3schools.com/xml/xml_examples.asp
- Jupyter

https://www.py4e.com/html3/13-web

JSON

JavaScript Object Notation (JSON)

- Data storage type
- Common syntax to store and transfer data across web applications
- Data is stored and transferred as text
  - lightweight
  - programming language independent

JavaScript Object Notation (JSON)

- Lists
  - Each element represents an observation e.g. person
- Dictionary
  - names represent variable names
    - e.g. phone number
  - values represent data values e.g. +1 734 303 4456

```
[{“name”:”Chuck”,
“phone”:{
  “type”: “intl”,
  “number”: “+1 734 3034456”}
“email”:{“hide”:”yes”}
},
{“name”:”Rodger”,
...
]
```

JavaScript Object Notation (JSON)

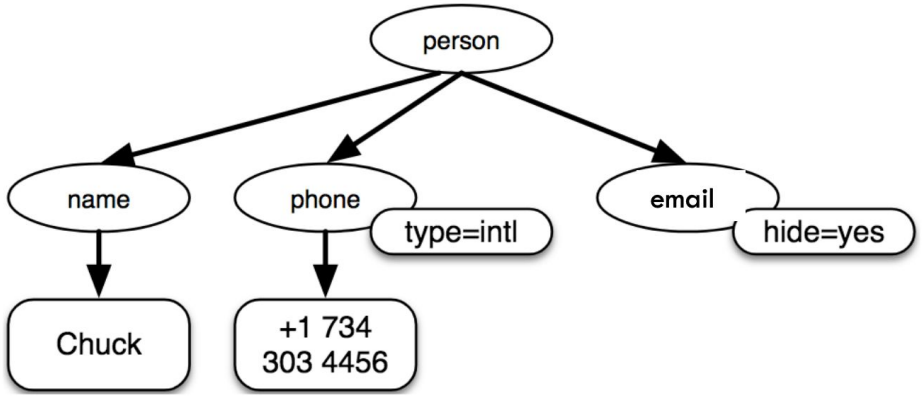
Code

```
{
  "name" : "Chuck",
  "phone" : {
    "type" : "intl",
    "number" : "+1 734 303 4456"
   },
   "email" : {
     "hide" : "yes"
   }
}
```

name

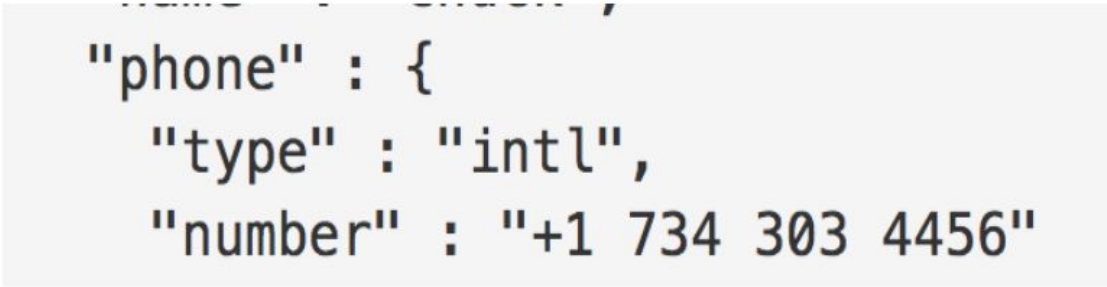
value

Representation



JSON - Data Storage Properties

- Hierarchical Structure:
  - Variables can be nested within other variables
    - e.g. phone has number and a type
  - Observations can be nested within other observations
    - e.g. goals are nested within match
- Inconsistent Variables
  - Variables and values may be missing for some observations
    - e.g. a match may have no goals or multiple goals



JSON - JSON Transfer

Creating and using JSON

> import json

1. Create a string with correct syntax

    > ppl = ‘[{“name”:”george”},{“name”:”mary”}]’

2. Convert string to json to python object

    > pplobj = json.loads(ppl)

    > print(pplobj)

Out: [{‘name’:’george’},{‘name’:’mary’}]

Ingesting JSON

> import json

1. Open text file

    > f = open(“people.json”)

2. Read in text file and convert to python object

    > pplobj = json.load(f)

    > print(pplobj)

    Out: [{‘name’:’george’},{‘name’:’mary’}]

Writing JSON

> import json

1. Open text file to write

    > f = open(“peoplev2.json”,”w”)

2. Convert python object to string

    > pplstr = json.dump(pplobj)

3. Write string to text file

    > f.write(pplstr) > f.close()

JSON Properties

- Hierarchical Structure:
    - Variables can be nested within other variables
        - e.g. phone has number and a type
    - Inconsistent Variables
        - Variables and values may be missing for some observations
    - Not great for data analysis
        - Consistent set of non-nested variables for all observations

Normalization

- Transform JSON to DataFrame
    - Flatten – to remove hierarchical structure
    - Fill in NAs – add missing values to observations with missing entries for variables

Normalization

- Specific Transformations:

Element	JSON	Data Frame
Variable names	keys in dictionaries	column indexes
Observations	elements in list	rows
Data values	values in dictionaries	cell entries

Normalization - Example

JSON

```
[{      “name”:”John”,
        ”age”:31,
        ”city”:”NYC”},
{       “name”:”Cynthia”,
        ”age”:55,
        ”city”:”Atlanda”},
{       “name”:”Bart”,
        ”age”:10,
        ”city”:”SF”}]
```

Data Frame

name	age	city
John	31	NYC
Cynthia	55	Atlanta
Bart	10	SF

json_normalize

json_normalize(jdata, record_path, meta)

- jsondata – JSON data loaded as a python object
- record_path – one or more keys which provide a link to the list that should be converted to rows if list is within a dictionary
- meta – one or more keys which provide additional variable data, if some variables are outside of the record_path list
- from pandas.io.json import json_normalize

Example 1

- JSON String
  - > fruits_str = '[{"kind":"apple","variety":"fuji"}, {"kind":"apple","variety":"jazz"}, {"kind":"orange","variety":"navel"}]'
- JSON Python Object
  - > fruits_json = json.loads(fruits_str)
- Data Frame
  - > fruits_df = json_normalize(fruits_json)

	kind	variety
0	apple	fuji
1	apple	jazz
2	orange	navel

Example 2

- JSON String
  - > fruits_str = ‘{“fruits”: [{"kind":"apple","variety":"fuji"}, {"kind":"apple","variety":"jazz"}, {"kind":"orange","variety":"navel"}]}’
- JSON Python Object
  - > fruits_json = json.loads(fruits_str)
- Data Frame
- > fruits_df = json_normalize(fruits_json)

```
0    [{'kind': 'apple', 'variety': 'fuji'}, {'kind'...
```

Example 2

- JSON String
  - > fruits_str = ‘{“fruits”: [{"kind":"apple","variety":"fuji"}, {"kind":"apple","variety":"jazz"}, {"kind":"orange","variety":"navel"}]}’
- JSON Python Object
  - > fruits_json = json.loads(fruits_str)
- Data Frame
  - > fruits_df = json_normalize(fruits_json,record_path=“fruits”)

Example 3

- JSON String
  - > fruits_str = '[{“kind”:”apple, “varieties”:[{"variety":"fuji"},{"variety":"jazz"}]}, {“kind”:”orange”, ”varieties”:[{"variety":"navel"}]}]’
- JSON Python Object
  - > fruits_json = json.loads(fruits_str)
- Data Frame
  - > fruits_df = json_normalize(fruits_json,record_path=“varieties”)

	variety
0	fuji
1	jazz
2	navel

Example 3

- JSON String
  - > fruits_str = '[{“kind”:”apple, “varieties”:[{"variety":"fuji"},{"variety":"jazz"}]}, {“kind”:”orange”, ”varieties”:[{"variety":"navel"}]}]’
- JSON Python Object
  - > fruits_json = json.loads(fruits_str)
- Data Frame
  - > fruits_df = json_normalize(fruits_json,record_path=“varieties”,meta=“kind”)

	variety	kind
0	fuji	apple
1	jazz	apple
2	navel	orange

Relational Databases: SQL

- Collection of one or more tables
  - Keys
    - Unique id(s)
  - Foreign keys
    - Relations between tables
  - Structured Query Language (SQL)

Similar Data Operations

Operation	Pandas	SQL
Represent data set	Data Frame	Table
Combine data sets	merge	JOIN
Aggregate data set	groupby	GROUP BY
Subset data set	iloc	WHERE

Example

- Write code to get the number of orders per aisle from the order_products data set in:
  - Pandas
  - MySQL

SQL in Python

- Connect to an existing database or create a new one
  - > con = sqlite3.connect(“database.sql”)
- Save tables to database
  - > dataframe.to_sql(“tablename”,con)
- Extract data using SQL queries
  - > newdataframe = pd.read_sql_query(query,con)

I appreciate your attention

Hope to see you on Thursday!

COLLEGE OF INFORMATION STUDIES

Reference Material
Install Software

4 Programming Assignments

- Work independently
- Deeper investigation into a data set and research question
- Turn in a well-structured and written report using Jupyter notebooks

Software Tools

- Python & Jupyter Notebook
  - Method 1
    - Python 3 (https://www.python.org/downloads)
      - Pandas Data Analysis Library (pandas)
      - Other modules (e.g. numpy, plotnine)
    - Jupyter Notebooks (aka ipython) (https://jupyter.org/install)
      - blend narrative text
      - code
      - output
      - visualizations
  - Method 2
    - Install Anaconda (includes both) (https://www.anaconda.com/distribution)
- Open Refine
  - http://openrefine.org/download.html
- Data sets
  - https://www.reddit.com/r/datasets/
  - https://opendata.dc.gov/
  - https://datasetsearch.research.google.com/
  - https://www.kaggle.com/datasets