Internet and Data

Resources and Risks and Power

Kenneth W. Regan

CSE199, Fall 2017

Outline Week 1 of 2: Data and the Internet

- What is *data* exactly? How much is there? How is it growing?
- Where data resides—in reality and virtuality. The Cloud. The Farm.
- How data may be accessed. Importance of **structure** and **markup**.
- Structures that help algorithms "crunch" data.
- Formats and protocols for *enabling* access to data.
- Protocols for *controlling* access and changes to data.
- **SQL**: `Select.` `Insert.` `Update.` `Delete.` `Create.` **Drop.**
- Dangers to privacy.
- Dangers of crime.
- (Dis-)Advantages of online data.
- [Week 1 Activity: Trying some SQL queries.]

What Exactly Is "Data"?

Several different aspects and definitions:

1. The entire track record of (your) online activity.
   - Note that any "real data" put online was part of online usage. Exception could be burning CD/DVDs and other hard media onto a server, but nowadays dwarfed by uploads. So this is the most inclusive and expansive definition.
   - Certainly what your carrier means by "data"—if you re-upload a file, it counts twice.
2. Structured information for a particular context or purpose.
   - What most people mean by "data."
   - Data repositories often specify the context and form.
   - Structure embodied in *formats* and *access protocols*.
3. In-between is what's commonly called "Unstructured Information"
   - Puts the *M* in *Data Mining*.
   - Hottest focus of consent, rights, and privacy issues.

How Much Data Is There?

- That is, How Big Is the Internet?
  - Searchable Web
  - Deep Web
  - (I maintain several gigabytes of deep-web textual data. . . tracking chess tournaments for possible cheating. Only tournament staff know the link—for their event only.)
- World Wide Web Size.
  - One **terabyte** = 1,000 **gigabytes**.
  - One **petabyte** = 1,000 **terabytes**. "**Big Data**"
  - One **exabyte** = 1,000 **petabytes**.
  - One **zettabyte** = 1,000 **exabytes**.
  - Next level is called **yottabyte**.
- Google currently holds about 15 exabytes.
- Internet on the whole is said to have entered the "Zettabyte Epoch."

Growth Rate of the Internet

- How much data is being added per minute?
- This widget quickly counts up 1TB added data.
- This graphic shows how all the burgeoning data divides into categories.
  - One vast category partly weaves through the graphic, but is largely off it.
  - Estimated here as comprising **30%** of all Internet *traffic*.
  - The musical "Avenue Q" says the Internet was made for it...
  - Is it Data? OK, not for the rest of these lectures...
- How can the Net's architecture absorb this expansion? (Other lectures)
- *Access* to data: *who* and *how*, is key.

Where Data Lives

- Data physically resides on "hard media" in computer systems.
- **Data Centers**
  - Often service governments—hopefully with redundancy.
  - Service multiple agencies and companies...
  - ...as opposed to a data warehouse organized by one company or partnership.
- Largest floor space is Range 1 in Langfang, China. Over 6.3M sq. ft., as big as the Pentagon.
- Nevada SuperNAP Reno got narrowly beaten at 6.2M sq. ft.
- Chicago Lakeside Technology Center, past champion at 1.1M sq. ft.

But for many users, where it lives virtually is in the Cloud.

Data Management and the Cloud

- The Cloud fits under the larger heading of data management services.
- Can be called an internetwork of data management services with common structures.
- Services are contracted to subscribers of all kinds: individuals to huge consortia.
- Responsible for:
  - physical maintenance of data;
  - recoverability in event of mutation or loss;
  - governing access to data;
  - security mechanisms against unauthorized access...
  - ...**and also improper usage**;
  - compatibility and interoperability;
  - algorithmic services.
- Many data centers are augmented with **server farms** to do the processing.

Access to Data

- Some data you own—and you (or your group) have sole access to it.
- Other data you own but wish to share outside your group, even publicly.
- Access to **read**...
- Access to **modify**...
- Not just permission, but ease of interpreting data is paramount.
- Owner and/or provider are responsible for *structuring* data.

Prime Directive: Eliminate—or at least minimize—the one-off work a client needs to do to interface with your data.

Some Structural and Algorithmic Constraints

A Basic Dilemma—which will echo early on in your courses:

1. Expect certain data points at preappointed positions, or
2. Search open-endedly for **tags** identifying the data points.

- Array lookup `arr[i]` is an example of the first.
- Associative lookup `$table{key}` exemplifies the second.
- But if the tag could be "anywhere" in a mound of data, much waste of time.
- The Cloud cannot serve *Random Access* on a large scale.
- The "Three Rules" of Real Estate (on the Net):
  - **Locality.**
  - **Locality.**
  - **Locality.**
- Whole Net system architectures (MapReduce/Hadoop/Google File System, Amazon Elastic Compute Cloud...) are designed to ensure that data is *Stream-Friendly*.

Data File Formats

- Positional formats typified by CSV, BMP
- Whereas TIFF tags images, XLSX adds markup to XLS…

Markup

- Long predates the Internet.
- Publisher markup for editing and typesetting (and interpretation).
- Jerome Saltzer, 1964: RUNOFF, which led to ROFF.
- Later: TeX, LaTeX... (As opposed to WYSIWYG)
- William Tunnicliff, 1967: "Generic Coding."
- Charles Goldfarb, 1969: organize legal documents.
- Led to IBM's Generalized Markup Language (**GML**), 1973.
- Standard Generalized Markup Language (**SGML**), ISO 1986.
- Extensible Markup Language (**XML**) started as a simpler SGML.
- Hypertext Markup Language (**HTML**) imitated SGML.
  - Introduced by Tim Berners-Lee in a 1991 forum post which linked to a document titled "HTML Tags."
- JavaScript Object Notation (**JSON**), Douglas Crockford, 2001.
- Now main alternative to XML, especially for *object serialization*.

Markup Example: SGML (source)

```
<recipe type="dessert" servings="6" preptime="10">  <!--Ten what?-->
<title>Haupia (Coconut Pudding)</title>
<ingredient-list>
<ingredient>
12 ounces coconut milk
</ingredient>      <!--Parser could allow omitting item close tag-->
<ingredient>
4 to 6 tablespoons sugar
...
</ingredient-list>
<instruction-list>
<step necessary="no">
Thoroughly wash and dry the pot you will use.
</step>
...
</instruction-list>
</recipe>
```

Example: The First HTML Doc (lightly altered)

```
<TITLE>Tags used in HTML</TITLE>
<NEXTID 22>
<H1>HTML Tags</H1>This is a list of tags used in the
<A NAME=0 HREF=MarkUp.html#4>HTML</A> language.
Each tag starts with a tag opener (a less than sign) and ends
with a tag closer (a greater than sign).
Many tags have corresponding closing tags which
identical except for a slash after the tag opener.
(For example, the <A NAME=3 HREF=#2>TITLE</A> tag).<P>
Some tags take parameters, called attributes.
...
Opening list tags are:
<DL>
...
</DL>
the closing tag must obviously match the opening tag.
```

Did not yet have `HEAD` and `BODY` structure. (Yes, word "are" is missing)

Example: XML and JSON Compared

From https://www.w3schools.com/js/js_json_xml.asp, XML first:

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

My Own Format Extending Chess "PGN" Standard

```
[GID "De Castellvi;Vinoles;Valencia;Valencia ESP;1475.??.??;?;1-0"]
[EID "Komodo-8-32bit"]
[Turn "6-w"]
[MovePlayed "h3"]
[EngineMove "Ne5"]
[Eval "+160"]
[Depth "12"]
...
       1    2    3    4    5    6    7    8    9   10   11   12
-------------------------------------------------------------------
Ne5  n.a. n.a. n.a. n.a. n.a. +142 +142 +140 +132 +147 +146 +160
d3   +110 NREC NREC NREC +053 +095 NREC NREC NREC NREC NREC NREC
Bxf7 n.a. n.a. n.a. n.a. +107 +079 NREC NREC NREC NREC NREC NREC
...
```

Mixes position-based and tagged elements. One [...] encloses tag and value.

Non-hierarchical structure.

What Does 'X'tensible Mean?

- Can tailor (to) data structures and interfaces.
- Can *define* a common user/program interface.
- E.g., common webpage display and protocol for new users of a web-deployed system.
- Close analogy to **Cascading Style Sheets** (CSS).
- Indeed, CSS interfaces with XML as a display and UI front end.
- **Document Type Definition** (DTD) specifies new SGML/XML elements and their syntax rules, which can allow "nesting."
- End-user apps need to be tailored to the DTD but this can be automated, e.g. by an *XML parser generator*.
- **Meta**-level: can extend the language to produce whole hierarchies of DTDs and meta-rules for specifying them.
- (PGN and my AIF have no formal DTD, are minimally extensible.)

Three Functions With Data—All Handled By SQL

1. Data Definition/Creation
2. Data Manipulation (read-only access included in this heading)
3. Data Control.

The **Structured Query Language** (SQL) handles all three.

- Donald Chamberlain, Raymond Boyce, IBM, early 1970s.
- Originally **S**tructured **E**nglish **QUE**ry **L**anguage, but "SEQUEL" trademark was taken. Still often pronounced that way.
- **Oracle Corp.** both extended and "front-ended" SQL.

Largely embodies Edgar F. Codd's **Relational Model** (RM).

Relational not positional. *Declarative* in that users are responsible only for data and queries, not algorithms or code. RM governs how database is built. Queries are built from logic and numerical predicates.

Some SQL Commands

`CREATE`. Note that it creates a structure before you input data.

```
CREATE TABLE Games (
    gid             VARCHAR(128)       PRIMARY KEY,
    white_name     VARCHAR(50)   not null,
    black_name      VARCHAR(50)    not null,
    result          VARCHAR(7)   not null,
    white_rating    INTEGER
    black_rating    INTEGER
);
```

Here `TABLE` is a built-in SQL type, or rather template for the user defined type `employees`. To kill it and all data you give both names:

```
DROP TABLE Games;
```

`TRUNCATE TABLE Games;` would destroy the entries but not the definition.

Inserting, Updating, and Removing Data

```
INSERT INTO Games (white_name, black_name, result)
   VALUES ('DeCastellvi', 'Vinoles', '1-0');

UPDATE Games SET gid = generate_game_id();
```

SQL allows user-defined functions, here to generate the game ID.

Since players didn't have ratings back in 1475, those fields can be left with a default `null` value. We could define a default of 0 but shouldn't—it would throw off `AVG` calculations. The `gid` field had a default which must be immediately changed, else the next insert will violate the `PRIMARY KEY` uniqueness constraint.

`DELETE FROM Games WHERE gid =` followed by the unique key removes just that game.

Can build by generating commands from data in XML/JSON/etc...

Selection and Logic in SQL

Suppose I want just the games where the lower-rated player won. A user-defined predicate `underdog_wins()` could have body:

```
(white_rating < black_rating AND result = '1-0')
   OR (white_rating > black_rating AND result = '0-1')
```

As with a *method* in OOP, the table object is implicit. Then

```
SELECT * FROM Games WHERE underdog_wins() = 1;
```

temporarily makes a table from just those games where the underdog won. In place of we could have listed just some fields to return.

User-defined functions can return whole tables. Tables can be `JOIN`ed together (in various ways) on common field(s).

(Yes, basic SQL needs that '`= 1`')

Converting Data to SQL Entry

```
<NFLTeams>
<Team code="ARI" teamName="Cardinals" region="Arizona"
      pop="4438000" lastPlayoffWin="2015"/>
<Team code="ATL" teamName="Falcons" region="Atlanta"
      pop="6462000" lastPlayoffWin="2016"/>
...
</NFLTeams>
```

```
CREATE TABLE NFLTeams (
   _code VARCHAR(50),
   _teamName VARCHAR(50),
   _region VARCHAR(50),
   _pop INT,
   _lastPlayoffWin INT
);
```

```
INSERT INTO NFLTeams VALUES ('ARI', 'Cardinals', 'Arizona', 4438000, 2015);
INSERT INTO NFLTeams VALUES ('ATL', 'Falcons', 'Atlanta', 6462000, 2016);
...
```

SQL Permissions

- The SQL standard finally includes a whole Data Control Language (DCL).
- Maintains a list of user IDs.
- Mostly done by two commands, `GRANT` and `REVOKE`.
- Rather than read-write-execute (`rwx`) permissions, it grants or withdraws allowed SQL commands. E.g.:
- `GRANT UPDATE ON Games TO garry_kasparov;`
- `REVOKE EXECUTE ON Games FROM PUBLIC;`
- Permissions can also be system-wide.
- Permissions can be grouped into *role* specifiers.
- Can build a management system on top of the SQL DCL.
- Permissions can be granted to not just people!
- Your "Al-Go-Rith-Ms" carry lots of SQL commands to submit...
- When "everything is data," those commands are data...and data is commands...

So Is This Data Heaven?

- **Structure**, **Extensibility**, and sheer computing power have built a brave new world.
- "Power Corrupts" is a **theorem** in CS.
- Microsoft Technet article on SQL serving:

> "Security is an exercise in creating enough barriers to the system such that the effort involved to attack a system exceeds the benefit derived from the data."

- It does *not* say, "Security is an exercise in making systems secure."
- Speedy execution cuts corners on safety.
- SQL by itself has several vulnerabilities.
- **Injection:** Trick a system into executing SQL privilege commands embedded in data.
- Show XKCD comic https://xkcd.com/327/

Other Potential Weaknesses

- Although SQL polices its own user-defined functions, it allows functions written in other languages.
- These can possibly import "unsafe code."
- Might exploit details/weaknesses in how the SQL system was implemented.
- Even within SQL, what happens if you give 100 chars to a VARCHAR(50)?
- Implementations "should" either (a) refuse or (b) truncate your string, but (a) can block a whole upload and (b) may cause constraint violations.
- For speed and simplicity too, systems might (c) take your whole string and overflow into another memory region.
- Such "buffer overflows" have bit from the 1988 Internet Worm to 2017's Cloudflare bug.
- I wrote a joint article on the latter.
- More about security in other weeks of this course...

Other Ways to Game a Database

- Even if a database is completely sound, the combination of incautious programming and unseen defaults can leave loopholes.
- Suppose no one under 12 can ride a roller coaster, so they wrote:

```
SELECT * FROM Riders WHERE NOT(age < 12);
```

- And suppose Bart Simpson can upload or finagle his record not to have an `age` field.
- Even if default is `null` or something producing a non-number "`nan`" value, the `age < 12` comparison may fail "gracefully."
- Then `NOT(age < 12)` will *succeed*—and Bart gets to ride!
- Yes, `...WHERE age >= 12` would have averted the problem.
- Database can be vulnerable in-between restoring constraints after upload.
- Point is: we can't escape attention to low-level details.

Data, Metadata, and Privacy

A rough working definition of **metadata** is:

> Data in XML headers and in `<tag ATTR=...>` attributes

In our previous `<recipe>` example this would include:

- It is a dessert.
- Serves 6 people and takes 10 minutes to prepare.
- *Maybe* the title "Haupia (Coconut Pudding)" is public.
- Has 13 ingredients and the recipe takes 17 steps, 3 unnecessary.

Does not give away the ingredients or their amounts or the instructions.

Metadata may be admissible in court when private content isn't.
E.g. time and duration (and recipient??) of cell phone calls.
[Discuss 2010 French chess cheating case and civil vs. criminal law.]

• Major controversy over gathering metadata by law enforcement and intelligence.

Privacy Via Slightly Fake News

- Many databases allow public access to "aggregates" such as mean, median, max, min, "90th percentile" values.
- Typified by allowing students to see the class average on UBLearns.
- Say 98 students average 75 on a test, then 2 in Band make it up.
- Say class average slips to 74.
- Do the math: they scored only 50 between them—they bombed it!
- **Differential Privacy** says to fuzz up aggregate values by $\pm\epsilon$.
- Say $\epsilon = 1\%$. Then 75 vs. 74 could easily have been "random variation." We don't really know.
- Special research topic at UB CSE.

Hacking

- What exactly is "hacking" or "being hacked"?
  - Positive meaning: Finding ingenious *specialized*(one-off) solutions.
  - Derived meaning: Productive coding in less time than it takes to form a plan.
  - Altering code to solve a one-off problem (or game objective) apart from its initial design.
  - Unauthorized *modification* of computer systems. (Entry, access, release all require some mod.)
  - Extensive effort at such modification (as opposed to betrayal or stupidity).
- 1903: Nevil Maskeleyne hacked a demo of Marconi's "wireless telgraphy" by inserting mocking verses in Morse code.
- 1960s: Phone "phreaking" to hijack lines and phone for free.
- 1979: First major computer system break-in, by Kevin Mitnick—now CHO of KnowBe4 Corp. (Surprised so late, given that the movie War Games came out in 1983.)

Crime, Legal Contours, and the Net

- First(?) Major Data Breach to Public: 2006 AOL "*Valdez*" (user search data, ID-ed by number but persons exposed).
- Too Many Examples today, clear thru to Equifax...
- Systems trying to cope by altering *verification* of data and *nature* of data:
  - GLL blog post, "Security Via Surrender"
  - GLL blog post, "Making Public Information Secret"
- Even with authorized access, *fair use* of public data is an issue.
- What does "copyright" mean when copying is so seamless?
- Programming language meanings such as *read-only*, *local copy*, *temporary* are shaping legal contours. (Revisit XKCD link.)
- After a "hack," who bears responsibility—and how much?
- 1998 DMCA: Internet providers not responsible.
- For misuse of Bram Cohen's BitTorrent—not so clear. Cut deal in 2005 with Motion Picture Association of America to follow DMCA.

Outline Rest of Week 2: Inferencing Via Net

One can imagine three (or more) levels of using data on the Internet:

- 'Level 1': Using Internet access to data repositories.
- 'Level 2': Studying Internet traffic: news sites, forums, commentary, research.
- 'Level 3': Analyzing human patterns of Net usage and interaction.

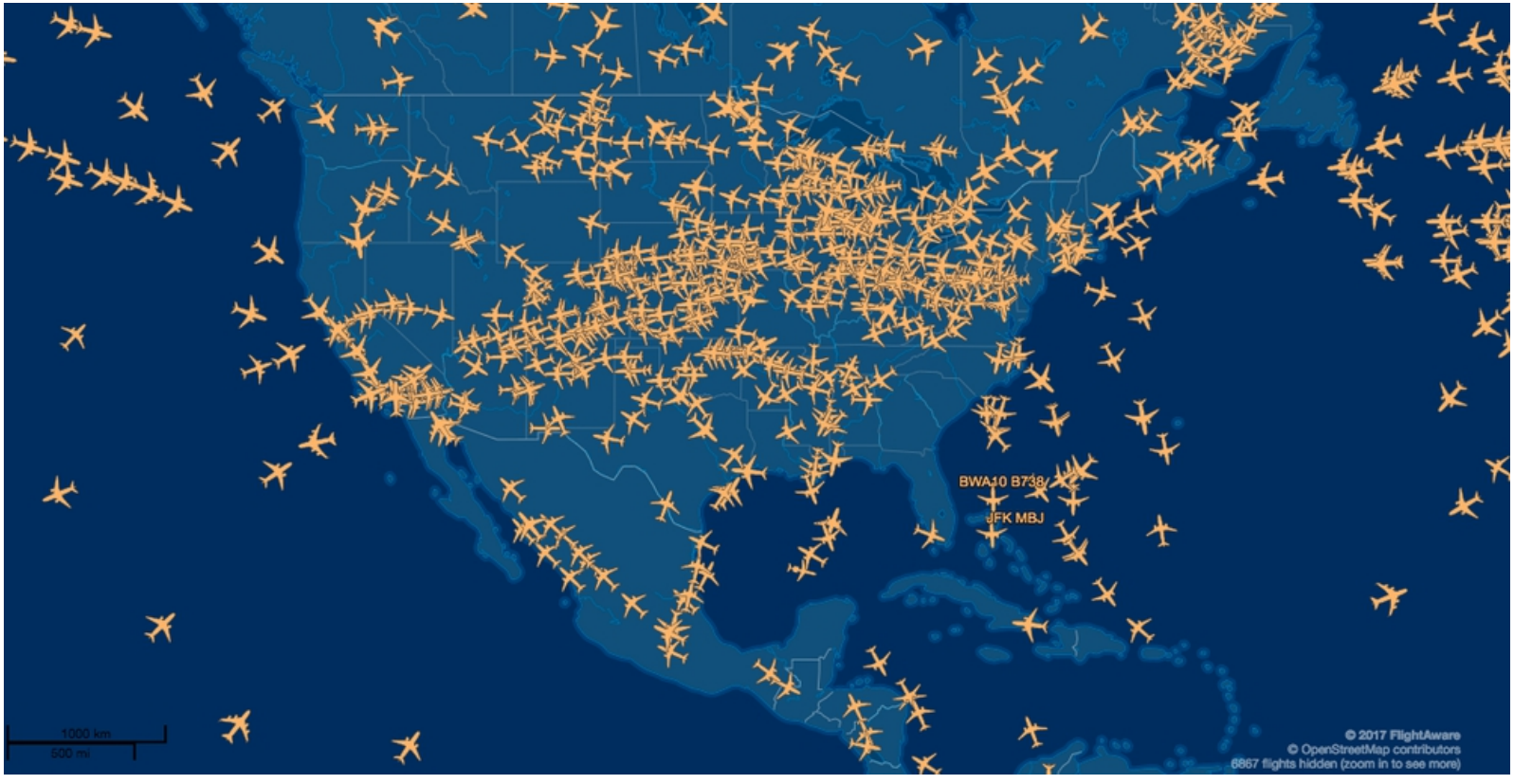
- Level 1 is becoming basic to how science is done.
- Level 2 is important from business (e.g., data-mining reviews, anticipating loads) to Homeland Security applications (e.g., filtering "chatter" for threats, tracking via face recognition).
- In-between 2 and 3 is companies tracking your data and usage patterns via (consented-to) "cookies."
- Level 3 emphasized in 2017 book *Everybody Lies: Big Data, New Data, and What the Internet Tells Us About Who We Really Are.*

A Global Data Village

- "No Man is an Island..." wrote John Donne in 1624.
- Then it was a "Meditation"; now pretty much a statement of fact.
- Article, "What Facebook Knows."
- Even more along Donne's lines, a person in Florida during Hurricane Irma was rescued by someone reading her Tweets in California: http://www.cnn.com/2017/09/11/us/social-media-irma-rescue-trnd/index.html
- Some data is intended to be out there—other data not...
- **Structured Data** has a pre-defined format and (hence) purpose. If it wasn't meant to be out there, a breach has occurred.
- **Unstructured Data** may not have been originally intended as data.

A "Semi-Structred" Example (of Inferencing)

FlightAware Live Tracker, Monday 9/11/17, about 9am:



Why almost no planes over the US Southeast? And Northern Mexico?

Incidentally, from the Monday 9/18 Buffalo News...

"The end of black boxes? Fredonia professor's invention could boost air safety"

> "By having planes connect with multiple servers, Zubairi's Flight Data Tracker system avoids overloading any one central server. These multiple ground servers collect the information and send it back to the origin airport, where the information is stored.
>
> (When a plane is flying over the ocean or a large desert, where there are no air traffic control towers, the more-expensive satellite option will have to be used.)
>
> ...[I]if something went wrong say, a missed landing, or the worst-case scenario, a crash then the black box information is immediately available."

Recall the 2014 Malaysia Airlines Flight 370 mystery.

Scientific Data

- Example: NIH Gene Expression Omnibus.
- Accepts submissions from Excel, XML, even plaintext but formatted like this.
- NASA Exoplanet Archive
- Key concern is **Reproducibility**.
- For example, someone else analyzing the raw exoplanet data should reach closely similar conclusions.
- Posting data makes this possible by 3rd-parties.
- Center For Open Science—emphasizes rigor and replication in social, medical, and environmental studies.
- Impetus to be public—except mainly for *privacy* concerns.
- Tension over *proprietary* aspects, especially for NSF grants, public universities...
- Look at all these public datasets!

Business Data

- Impetus to be *proprietary*.
- Profit$ replace reproducibility as regards validation.
- Two layers of privacy concerns:
  - Data contracted to be used by clients.
  - Data gathered on customers and competitors.
- Same concerns apply to government agencies.
- Can build *models* based on past record and *correlations*...
- ...with less responsibility than scientists to establish *causation*.
- Example: "Binge-Watching TV Is Killing Us."
- Or do already sick and less-active people watch more TV?
- Either way, can insert targeted ads...

What Is Machine Learning?

> The act of modifying a system or algorithm A via interactions with examples and other data so that A can emulate (and/or predict) the interactions without any more data.

- Your Brain is Included.
- Simple Example: Building a Model.
- Simple Linear Regression Model: $Y = a + bX$.
- E.g. Walk_Likelihood $= a + b\cdot$(Pitch_Count).
- [show graphs from FanGraphs article.]
- The point is that the model can emulate/project the results of pitches by itself—when it goes bad, the manager takes the actual flesh-and-blood pitcher out of the game.

Styles of Machine Learning

- **Supervised Learning:** examples are structured and desired responses are labeled.
  - Regression usually falls into this category.
  - *Classification* according to predetermined criteria. Knowledge Bases.
  - *Training* new Apple iPhone X on labeled datasets of faces.
- **Unsupervised Learning:** responses not labeled, data often unstructured. (Hallmark of "Big Data")
  - *Cluster analysis* is a typical example.
  - Finding patterns in data that are not pre-defined.
  - *Principal Component Analysis* (PCA)—used for face recognition too.
- **Semi-Supervised Learning:** Mix and match these approaches...
- **Reinforcement Learning:** Algorithms A act as autonomous *agents*, receive "rewards" and "demerits," and modify their parameters according to what gives increasing rewards.
- **Deep Learning:** Build layers on successful modeling...

Some Notorious Inferences and Model Decisions

- **Target**ing ads at a pregnant teen: article.
- Amazon often recommends to me the book *Quantum Algorithms Via Linear Algebra*. Problem is—I co-wrote it. Nice to hear...
- Bond and CDO (Collateralized Debt Obligation) ratings before the 2008 crash.
- Book *Weapons of Math Destruction*, by Cathy O'Neill. Thesis: Mathematical models fossilize biases in data from remote history and skewed prior sources.
- Book Everybody Lies: Big Data, New Data, and What the Internet Can Tell Us About Who We Really Are, by Seth Stephens-Davidowitz. Thesis: Formal survey responses are inconsistent with opinions from the same populations mined on social media.
- Insofar as we are the training data for the Internet, the latter has baked in tangible amounts of racism and sexism.

Twitter and Facebook and More

- Using Twitter to predict (postdict) elections: Brexit 2016, Canada 2015, USA 2016 (paper), USA 2016 (BrandsEye).
- Vehicle defect discovery: paper.
- Consumer sentiment analysis: paper.
- Mining Facebook behavior by MasterCard: news article.
- Textbook: *Mining the Social Web*.
- Google Ngram Viewer tracks historical usage of terms and phrases.

Turing's Principle

Alan Turing: Besides his WWII work on the Enigma machine (featured in the movie *The Imitation Game*) and **Turing Machine** theory of computation in his 1936-38 PhD thesis under Alonzo Church, he is considered the **founder** of Artificial Intelligence. The **Church-Turing Thesis** is primarily stated in terms of the class of *computable functions*, but here is Turing's angle:

> **Anything that human beings can consistently deduce or classify can also be achieved by computers acting alone.**

The **Turing Test** involves computers trying to be indistinguishable from humans in ordinary life communications and transactions.

Turing All the Possibilities

> TP: If it is easy for humans then it will soon be easy for computers.

Defied by a CAPTCHA: "Completely Automated Public Turing test to tell Humans and Computers Apart": vision tasks hard for computers.

Logical **contrapositive** of Turing's Principle:

> If it is really hard for computers then it should be hard for humans.

What we fear when worrying that AI will take away our jobs is:

> Stuff that is hard for humans but easy for computers.

The logical **converse** of Turing's Principle acts as a brake, however:

> If X is hard for humans—insofar as we can't consistently agree on answers—then X is hard for computers too.

Some Hard Data Challenges (based on the converse principle)

- Inferring people's opinions and beliefs based on text alone. **Stance Classification**
  - How to do it when grammar and intent may differ?
  - Example: "[*that*—] you didn't build that" video.
- Reliable automatic translation.
  - Google Translate data-mines known translations for corresponding phrases.
- Election status (might not be well-defined).
- Identifying faces conclusively.
  - Apple iPhone X has bet on it.
  - Scotland Yard employs special humans to examine photos.
  - Super-Recognizers.com
- Scene analysis in greater generality.
- General anomaly alert systems.

Two Activities—Elements of Data Science

- Both activities are ready-made Python programs.
- Use of demo Python 3 Trinket can remove need to download and install Python or use CSE machines at command line.
- Both quick to run—activity will be in examining, reflecting, maybe tweaking, finally interpreting (and playing).
- Full setup directions on the activity sheet.
- First activity continues NFL data from last week.
- Is playoff failure (as particularized by the number of years since the last playoff win) correlated to media market size? to high coaching turnover? both? neither?
- Run linear regressions that also report the R^2 correlation index.
- How strong are the correlations? Do all of them "exist"?

Second Activity: Textual Content Analysis

Problem: Take the "emotional temperature" of a webpage.

- We will do this in a simple-minded way. (Often simple is best with data and models.)
- Assign an "affect intensity" to every word in the dictionary.
- Helper words 'a', 'the', 'is'... get 0. Any unlisted word gets 0.
- Add up the scores of all the words—ignore context (being simple).
- Divide by the number of words. (Or # of non-helper words?)
- Get a crude but useful index.

Running the Application

- Need to (a) fetch webpage(s) and (b) compute the index.
- Let's call the whole thing the Simple Communications Analysis Reader And Measure of Unsavory Content Calibration Index. (SCARAMUCCI)
- Get an ensemble of numerical results. *Now the real questions begin:*

1. What do the results signify?
2. Which differences (if any) are significant?
3. Are conclusions in line with model design and justified by methodology?
4. Dos the model meet desired performance specs?

See activity sheet for what we really obtained to use.

Going Beyond: Training and Assessment

- Consider an application that is simpler in vocabulary but now cares about *context*.
- Hotwords include "bomb," "threat," "fuse," "ignition," "laptop" ...
- But could be on a page teaching how to combat threats.
- We want to *distinguish* threat pages from benign ones.
- **Train** the model on pages with known classification.
  - How many does the app mis-classify?
  - How wide/significant are its correct distinctions?
  - Which *combinations* of words matter most to correct classification?
- Can *regress* the numerical word-score multipliers to optimize training results.
- Can *infer* which combinations should be sought / given high weight.

Going Beyond: Training Cycle

- This sets up a cycle of testing and tweaking the model. **Can be automated.**
- A "Rule of Three": Divide the in-house data into 3 parts.
  1. Use first part to *train* the model.
  2. Use second part as a *fresh test* of the trained model. Go back to step 1 if needed.
  3. Use 3rd part as published field test of the application.
- Sometimes the middle step is elided; e.g., *Kaggle* competitions on models of rating chess games had just 2 piles.
- Activity sheet has some supporting discussion toward the end—this could become a larger research project.