

---

**CSE 519: Data Science**  
**Steven Skiena**  
**Stony Brook University**

---

Lecture 6: Assembling Data Sets

---

# Data Munging

---

Good data scientists spend most of their time cleaning and formatting data.

The rest spend most of their time complaining there is no data available.

*Data munging* or *data wrangling* is the art of acquiring data and preparing it for analysis.

---

# Languages for Data Science

---

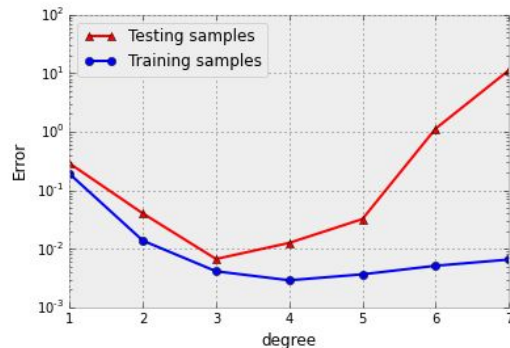
- *Python*: contains libraries and features (e.g. regular expressions) for easier munging.
  - *R*: programming language of statisticians.
  - *Matlab*: fast and efficient matrix operations.
  - *Java/C*: language for Big Data systems.
  - *Mathematica/Wolfram Alpha*: symbolic math.
  - *Excel*: bread and butter tool for exploration.
-

# Notebook Environments

Mixing code, data, computational results, and text are essential for projects to be:

- reproducible
- tweakable
- documented.

```
In [40]: degrees = range(1, 8)
errors = np.array([regressor3(d) for d in degrees])
plt.plot(degrees, errors[:, 0], marker='^', c='r', label='Testing samples')
plt.plot(degrees, errors[:, 1], marker='o', c='b', label='Training samples')
plt.yscale('log')
plt.xlabel("degree"); plt.ylabel("Error")
= plt.legend(loc='best')
```



By sweeping the degree we discover two regions of model performance:

- **Underfitting** (degree < 3): Characterized by the fact that the testing error will get lower if we increase the model capacity.
- **Overfitting** (degree > 3): Characterized by the fact the testing will get higher if we increase the model capacity. Note, that the training error is getting lower or just staying the same!

# Data Pipelines

---

Notebooks make it easier to maintain data pipelines, the sequence of processing steps from start to finish.

**Expect** to have to redo your analysis from scratch, so build your code to make it possible.



# Standard Data Formats

---

Historically, computer scientists would rather share a toothbrush than a data format.

But accepted standards are now available:

- *CSV files*: for tables like spreadsheets
  - *XML*: for structured but non-tabular data.
  - *JSON*: Javascript Object Notation for APIs.
  - *SQL databases*: for multiple related tables.
-

# Where Does Data Come From?

---

The critical issue in any modelling project is finding the right data set.

This will certainly be the case for your projects!

Large data sets often come with valuable **metadata**: e.g. book titles, image captions, Wikipedia edit history...

Repurposing metadata requires imagination.

---

# Sources of Data

---

- Proprietary data sources
  - Government data sets
  - Academic data sets
  - Web search
  - Sensor data
  - Crowdsourcing
  - Sweat equity
-



# Proprietary Data Sources

---

Facebook, Google, Amazon, Blue Cross, etc. have exciting user/transaction/log data sets.

Most organizations have/should have internal data sets of interest to their business.

Getting outside access is usually impossible.

Companies sometimes release rate-limited APIs, including Twitter and Google.

---

# Government Data Sources

---

- City, State, and Federal governments are increasingly committed to open data.
  - Data.gov has over 100,000 open data sets!
  - The Freedom of Information Act (FOI) enables you to ask if something is not open.
  - Preserving privacy is often the big issue in whether a data set can be released.
-

# Academic Data Sets

---

- Making data available is now a requirement for publication in many fields.
  - Expect to be able to find economic, medical, demographic, and meteorological data if you look hard enough.
  - Track down from relevant papers, and ask.
  - Google topic and “Open Science” or “data”
-

# Web Search/Scraping

---

*Scraping* is the fine art of stripping text/data from a webpage.

Libraries exist in Python to help parse/scrape the web, but first search:

- Are APIs available from the source?
- Did someone previously write a scraper?

Terms of service limit what you can legally do.

---

# Available Data Sources

---

- Bulk Downloads: e.g. Wikipedia, IMDB, Million Song Database.
- API access: e.g. New York Times, Twitter, Facebook, Google.

Be aware of limits and terms of use.

---

# Sensor Data Logging

---

The “Internet of Things” can do amazing things:

- Image/video data can do many things: e.g. measuring the weather using Flickr images.
- Measure earthquakes using accelerometers in cell phones.
- Identify traffic flows through GSP on taxis.

Build logging systems: storage is cheap!

---

# Crowdsourcing

---

Many amazing open data resources have been built up by teams of contributors:

- Wikipedia/Freebase
- IMDB

Crowdsourcing platforms like Amazon Turk enable you to pay for armies of people to help you gather data, like human annotation.

---

# Sweat Equity

---

But sometimes you must work for your data instead of stealing it.

Much historical data still exists only on paper or PDF, requiring manual entry/curation.

At one record per minute, you can enter 1,000 records in only two work days.

Often projects require sweat equity.

---



# Project Data Sources

---

- Miss Universe?
  - Movie gross?
  - Baby weight?
  - Art auction price?
  - Snow on Christmas?
  - Super Bowl / College Champion?
  - Ghoul Pool?
  - Future Gold / Oil Price?
-

# Cleaning Data: Garbage In, Garbage Out

---

Many issues arise in ensuring the sensible analysis of data from the field, including:

- Distinguishing errors from artifacts.
  - Data compatibility / unification.
  - Imputation of missing values.
  - Estimating unobserved (zero) counts.
  - Outlier detection.
-

# Errors vs. Artifacts

---

- Data **errors** represent information that is fundamentally lost in acquisition.
- **Artifacts** are systematic problems arising from processing done to data.

The key to detecting artifacts is the **sniff test**, examining the product closely enough to get a whiff of something bad.

---

# First-time Scientific Authors by Year?

---

In a bibliographic study, we analyzed PubMed data to identify the year of first publication for the 100,000 most frequently cited authors.

*What should the distribution of new top authors by year look like?*

It is important to have a preconception of any result to help detect anomalies.

---

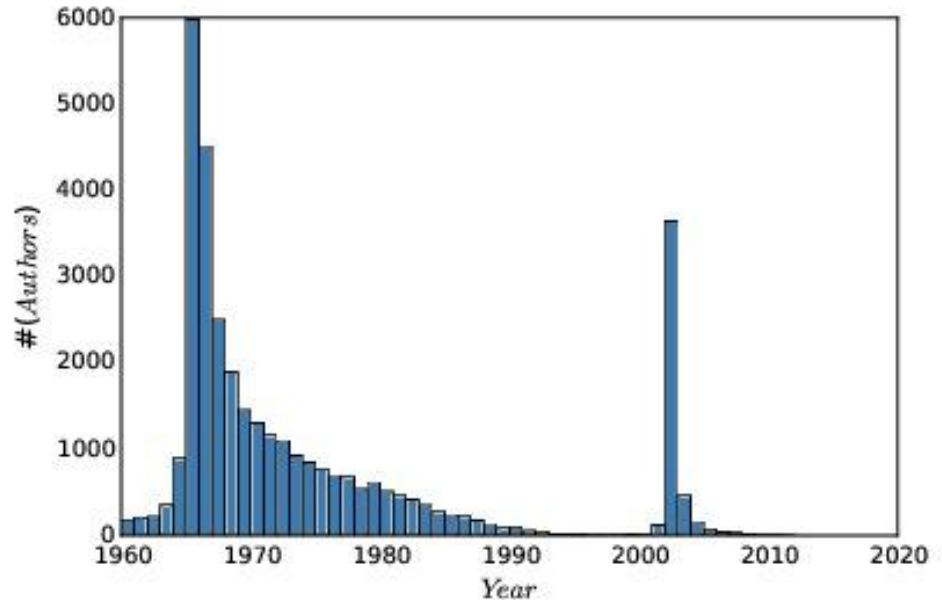
# Might this be Right?

---

A student once tried to foist this off on me.

What artifacts do you see?

What possible explanations could cause them?



# Mystery Solved!

---

Pubmed used author first names starting in 2002.

**SS Skiena** became  
**Steven S Skiena**

Data cleaning gets rid of such artifacts.

