UMD DATA605 - Big Data Systems

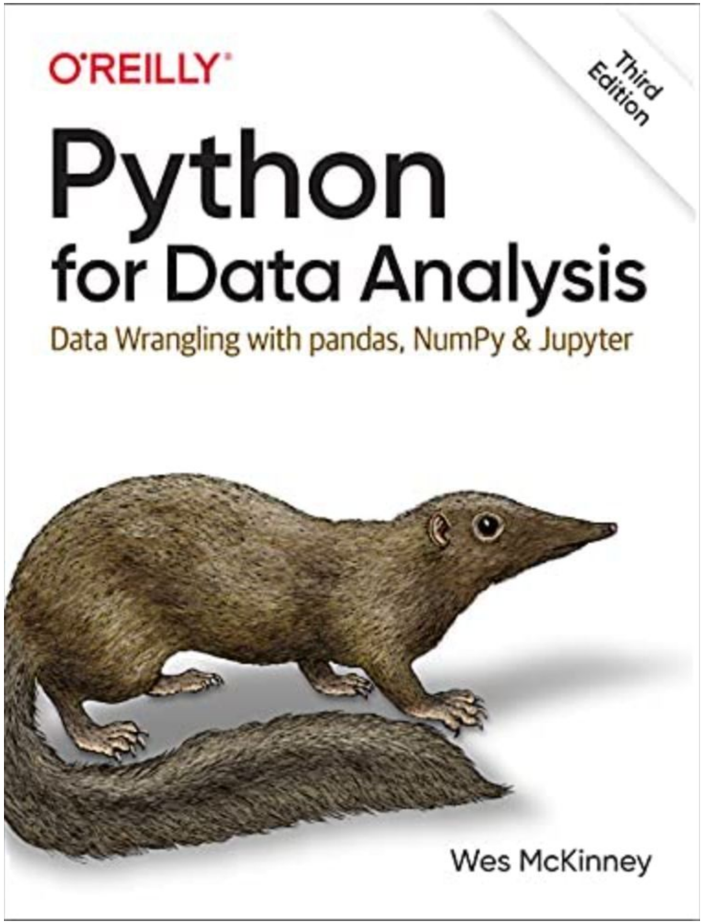
Data Wrangling

Dr. GP Saggese
gsaggese@umd.edu

with thanks to Amol Deshpande

Resources

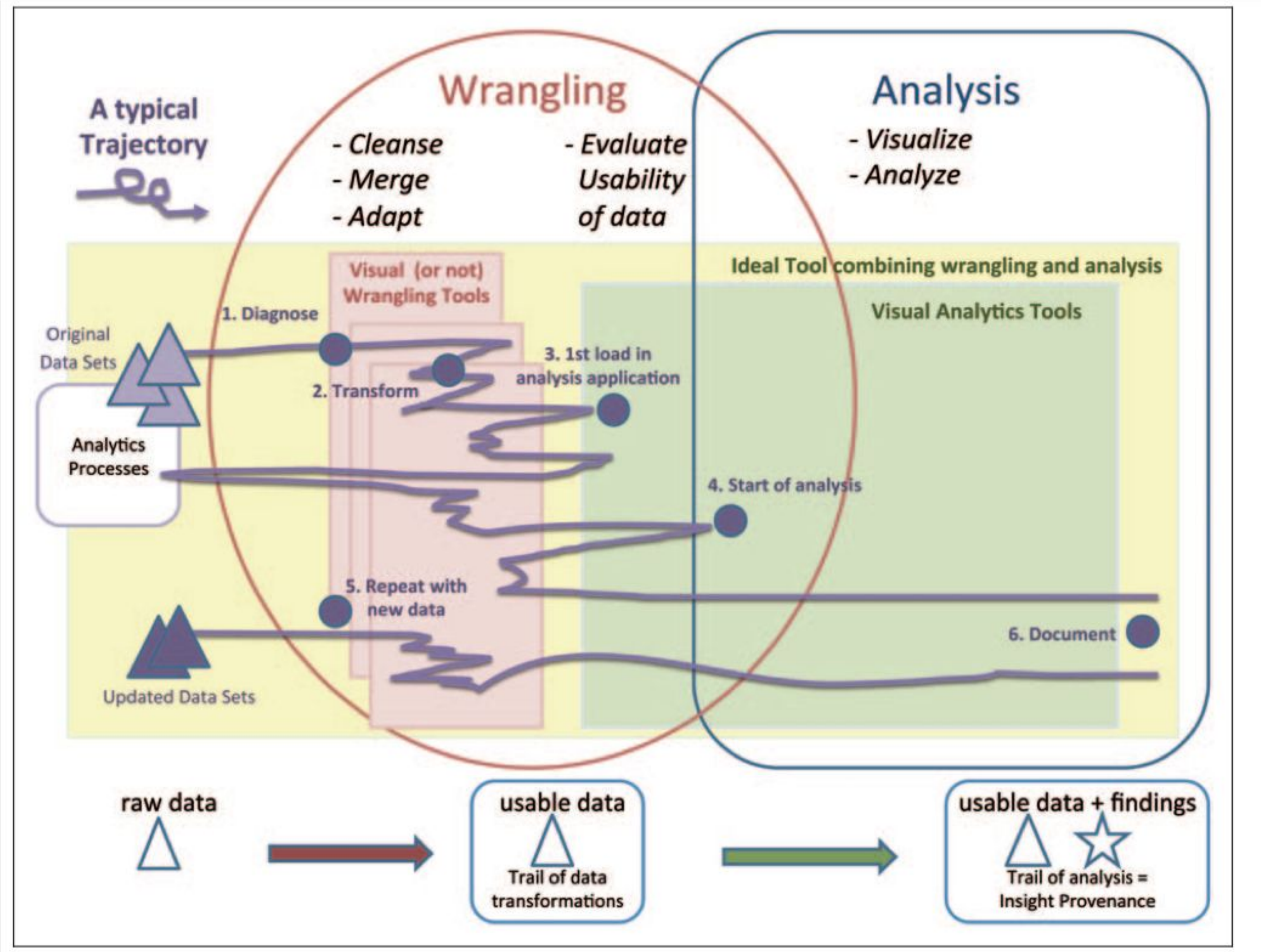
- Pandas tutorial
- Class project
- Web
  - https://pandas.pydata.org
  - Onslaught of free resources
- Mastery
  - https://wesmckinney.com/book



Overview

- **Data wrangling**
  - Get data into a structured form suitable for analysis
  - Aka data preparation, data munging, data curation
  - Often it is the step where majority of time (80-90%) is spent
- **Key steps**
  - Scraping: extract information from sources (e.g., webpages, spreadsheets)
  - Data transformation: get data into the right structure
  - Data integration: combine data from multiple sources
  - Information extraction: extract structured information from unstructured / text sources
  - Data cleaning: remove inconsistencies / errors

Overview



A typical Trajectory
Wrangling
- Cleanse
- Merge
- Adapt
- Evaluate Usability of data
Analysis
- Visualize
- Analyze
Ideal Tool combining wrangling and analysis
Visual (or not) Wrangling Tools
Visual Analytics Tools
Original Data Sets
Analytics Processes
1. Diagnose
2. Transform
3. 1st load in analysis application
4. Start of analysis
5. Repeat with new data
6. Document
Updated Data Sets
raw data
usable data
Trail of data transformations
usable data + findings
Trail of analysis = Insight Provenance

Overview

- Many of the data wrangling problems are not easy to formalize, and have seen little research work, e.g.,
  - Data transformation, i.e., put the data in the “right” structure
  - Information extraction: highly domain specific
  - Data cleaning: somewhat studied (e.g., tidy data)
- Others aspects of integration have been studied in depth, e.g.,
  - Schema mapping
  - Data integration
- Typical workflow
  - From Data Cleaning: Problems and Current Approaches
  - Somewhat old: data is mostly coming from structured sources
  - For a data scientist, the data scraping is equally important

Overview

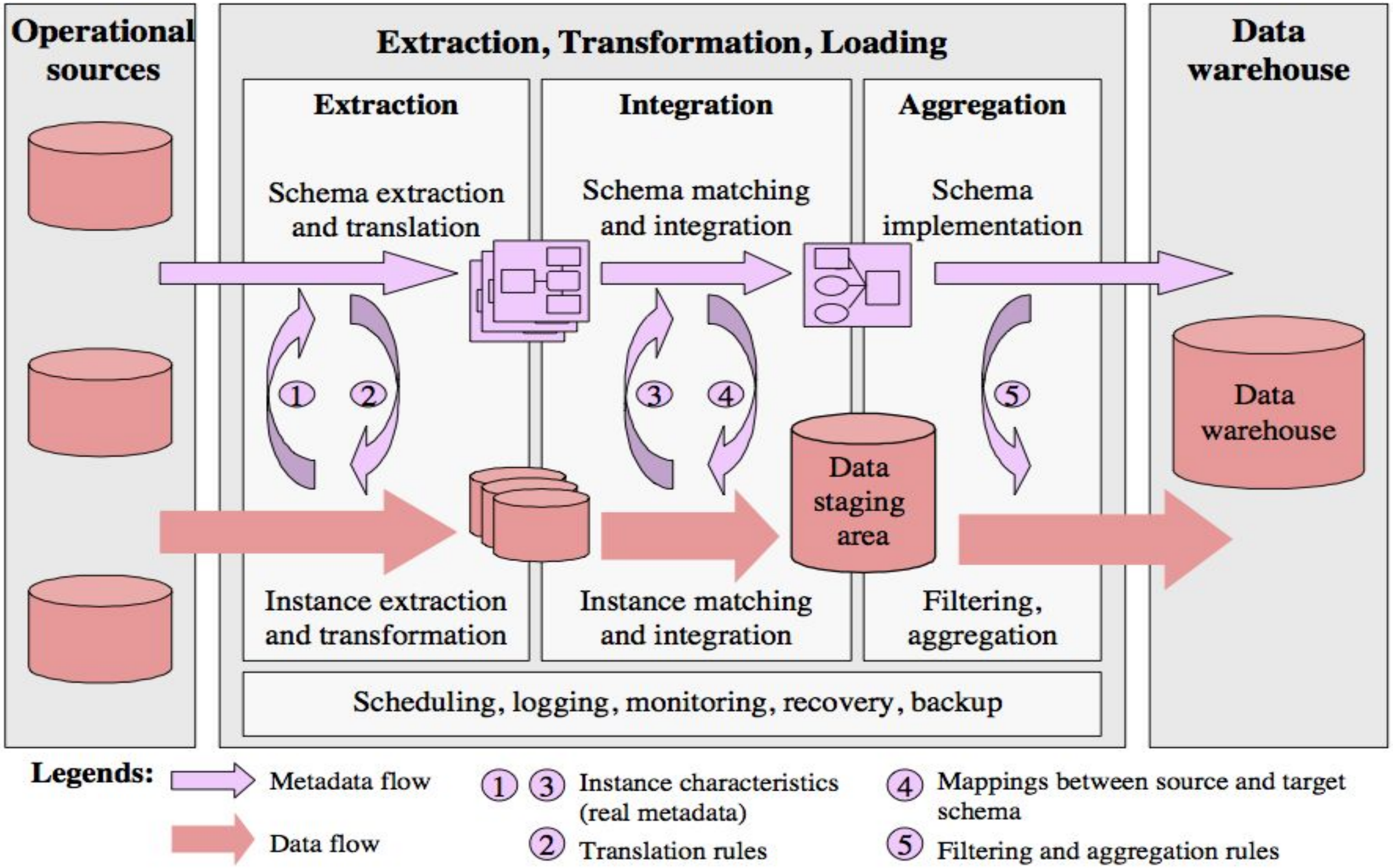


Figure 1. Steps of building a data warehouse: the ETL process

Data Scraping

- Data may reside in a wide variety of different sources
  - Files (e.g., CSV, JSON, XML)
  - Different databases
  - Spreadsheets
- Most analytical tools support importing data from such sources
  - Adapters to load data
- Web scraping: scraping data from web sources is tougher
  - In some cases there may be APIs
  - In other cases data may have to be explicitly scraped
  - Often pipelines are set up to do this on a periodic basis
    - Can be fragile
  - Several tools out there to do this (somewhat) automatically
    - E.g., import.io, portia, ...

Tidy Data

- Tidy data, Wickham, 2014 (here)
  - Each variable forms a column
  - Each observation forms a row
- Wide vs long format

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

"Messy" data

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Tidy data

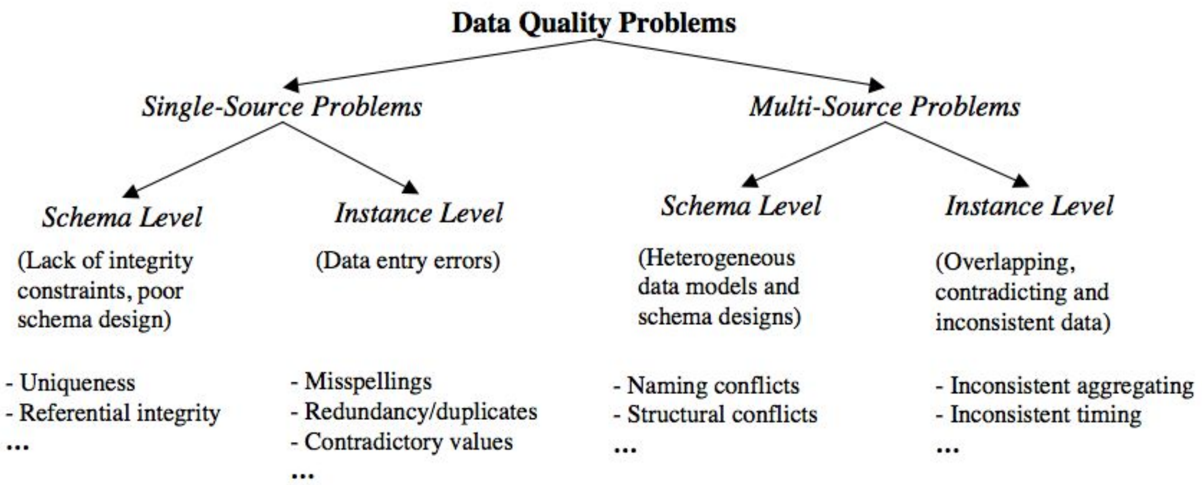
type	date	clicks	conversions	impressions
0	2020-01-01	1.0	NaN	18.0
1	2020-01-02	2.0	NaN	19.0
2	2020-01-03	1.0	1.0	14.0
3	2020-01-04	NaN	NaN	5.0
4	2020-01-05	1.0	NaN	8.0
5	2020-01-06	1.0	1.0	15.0
6	2020-01-07	2.0	NaN	8.0

Wide format

	date	type	count
0	2020-01-01	impressions	18.0
1	2020-01-02	impressions	19.0
2	2020-01-03	impressions	14.0
3	2020-01-04	impressions	5.0
4	2020-01-05	impressions	8.0
...	...	...	...
91	2020-01-28	conversions	NaN
92	2020-01-29	conversions	NaN
93	2020-01-30	conversions	NaN
94	2020-01-31	conversions	NaN
95	2020-02-01	conversions	NaN

Long format

Data Quality Problems



Scope/Problem		Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city="Liipzig"	usually typos, phonetic errors
	Cryptic values, Abbreviations	experience="B"; occupation="DB Prog."	
	Embedded values	name="J. Smith 12.02.70 New York"	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city="Germany"	
Record	Violated attribute dependencies	city="Redmond", zip=77777	city and zip code should correspond
Record type	Word transpositions	$name_1$= "J. Smith", $name_2$="Miller P."	usually in a free-form field
	Duplicated records	emp_1=(name="John Smith",...); emp_2=(name="J. Smith",...)	same employee represented twice due to some data entry errors
	Contradicting records	emp_1=(name="John Smith", bdate=12.02.70); emp_2=(name="John Smith", bdate=12.12.70)	the same real world entity is described by different values
Source	Wrong references	emp=(name="John Smith", deptno=17)	referenced department (17) is defined but wrong

Single-Source Problems

- Depends largely on the source
- Databases can enforce constraints
- Data extracted from files or spreadsheets is often clean
- Data scraped from web-pages is much more messy
- Types of problems:
  - Ill-formatted data (especially from web-pages or files or spreadsheets)
  - Missing or illegal values, misspellings, use of wrong fields, extraction issues (e.g., not easy to separate out different fields)
  - Duplicated records, contradicting information, referential integrity violations
  - Unclear default values
  - Evolving schemas or classification schemes (for categorical attributes)
  - Outliers

Multi-Source Problems

- Different sources are:
  - Developed separately
  - Maintained by different people
  - Stored in different systems
- Issue 1: Schema mapping / transformation
  - Mapping information across sources
  - Naming conflicts: same name used for different objects
  - Structural conflicts: different representations across sources
- Issue 2: Entity resolution
  - Matching entities across sources
- Issue 3: Data quality issues
  - Contradicting information
  - Mismatched information
  - ...

Data Cleaning: Outlier Detection

- Quantitative Data Cleaning for Large Databases, Hellerstein, 2008 (here)
  - Focuses on quantitative data (i.e., integers/floats that measure some quantities of interest)
- Sources of errors in data
  - Data entry errors: users putting in arbitrary values to satisfy the form
  - Measurement errors: especially sensor data
  - Distillation errors: errors that pop up during processing and summarization
  - Data integration errors: inconsistencies across sources that are combined together

Univariate Outlier Detection

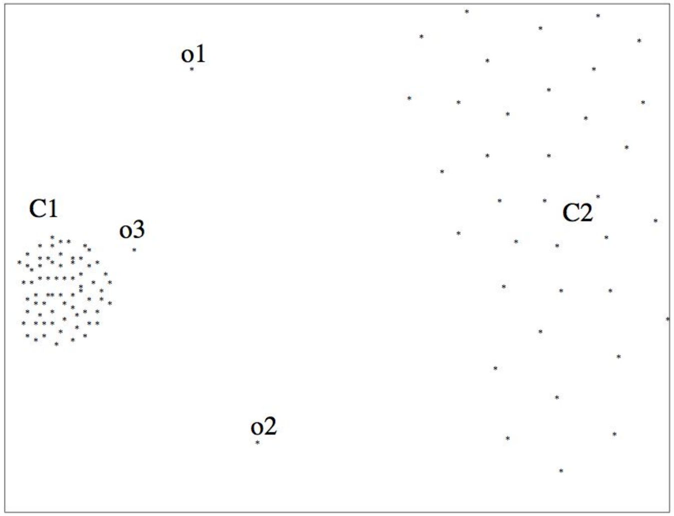
- A set of values can be characterized by metrics such as center (e.g., mean), dispersion (e.g., standard deviation), and skew
- Can be used to identify outliers
  - Must watch out for "masking": one extreme outlier may alter the metrics sufficiently to mask other outliers
  - Use robust statistics: considers effect of corrupted data values on distributions
  - Robust center metrics: median, k% trimmed mean (i.e., discard lowest and highest k% values)
  - Robust dispersion: median absolute deviation (MAD), median distance of all the values from the median value
- A reasonable approach to find outliers (assuming normal distribution)
  - Any data points 1.4826x MAD away from median
  - May need to eyeball the data (e.g., plot a histogram) to decide if this is true

Outlier Detection

- Wikipedia Article on Outliers lists several other normality-based tests for outliers
- If data appears to be not normally distributed:
  - Distance-based methods: look for data points that do not have many neighbors
  - Density-based methods:
    - Define *density* to be average distance to *k* nearest neighbors
    - *Relative density* = density of node/average density of its neighbors
    - Use relative density to decide if a node is an outlier
- Most of these techniques start breaking down as the dimensionality of the data increases
  - *Curse of dimensionality*
  - Can project data into lower-dimensional space and look for outliers there
    - Not as straightforward

Multivariate Outliers

- Analogous to univariate
- One set of techniques based on assuming data follows a *multi-variate normal distribution*
  - Defined by a *mean* μ and a *covariance matrix* Σ
- Mahalanobis distance of a point
  - Square root of $(x - \mu)'\Sigma^{-1}(x - \mu)$
  - Measures how far the point x is from the multivariate normal distribution
  - Outliers are points that are too far away
- Mean / covariance are not *robust* (sensitive to outliers)
  - Iterative approach: remove points with high Mahalanobis distance, recompute the mean and covariance
  - Several other general approaches discussed in the reference by Hellerstein
  - Need to try different techniques based on the data
- Often the volume of data may be too much (e.g., internet routers)
  - Approximation techniques often used



Time Series Outliers

- Often data is in the form of a time series
- Rich literature on *forecasting* in time series data
- Can use the historical values / patterns in the data to flag outliers
  - Rolling standard deviation or MAD (median absolute variation)

Split-Apply-Combine

- The Split-Apply-Combine Strategy for Data Analysis, Wickam, 2011 (here)
- Common data analysis pattern
  - Split: break a big problem into manageable pieces
  - Apply: operate on each piece independently
  - Combine: combine the pieces back together
- Pros
  - Code is compact
  - Easy to parallelize
- E.g.,
  - group-wise ranking
  - group vars (e.g., sums, means, counts)
  - create new models per group
- Supported by many languages
  - Pandas
  - SQL GROUP BY operator
  - Map-Reduce

```
In [94]: animals.groupby("kind").height.agg(
   ....:     min_height="min",
   ....:     max_height="max",
   ....: )
   ....:
Out[94]:
      min_height  max_height
kind
cat          9.1         9.5
dog          6.0        34.0
```