LECTURE 07

Data Wrangling and EDA

Exploratory Data Analysis and its role in the data science lifecycle

Today's Roadmap

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- Faithfulness (and Missing Values)

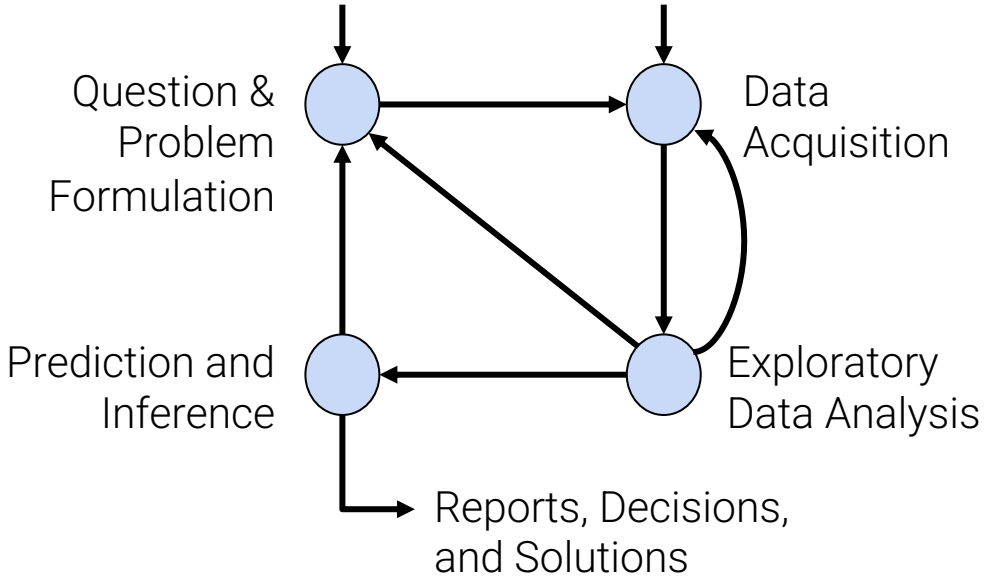
EDA Demo: Mauna Loa CO2

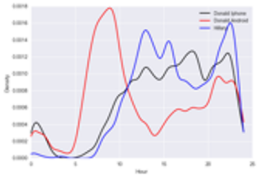
Now

You **have collected** or **<u>have been given</u>** a box of data.

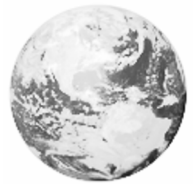
What do you do next?

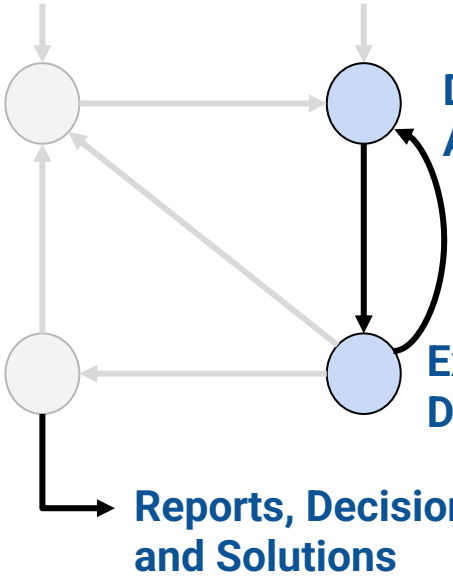
Plan for next few lectures

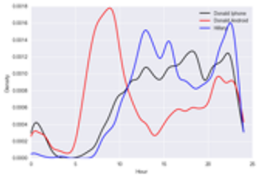




Plan for next few lectures







(today)

Working with Numerical and Text Data Regular Expressions	Data Wrangling Intro to EDA	Plots and variables Seaborn	Viz principles KDE/Transformations

(Part I: Processing Data) (Part II: Visualizing and Reporting Data)

Data Wrangling and **EDA:** An Infinite Loop

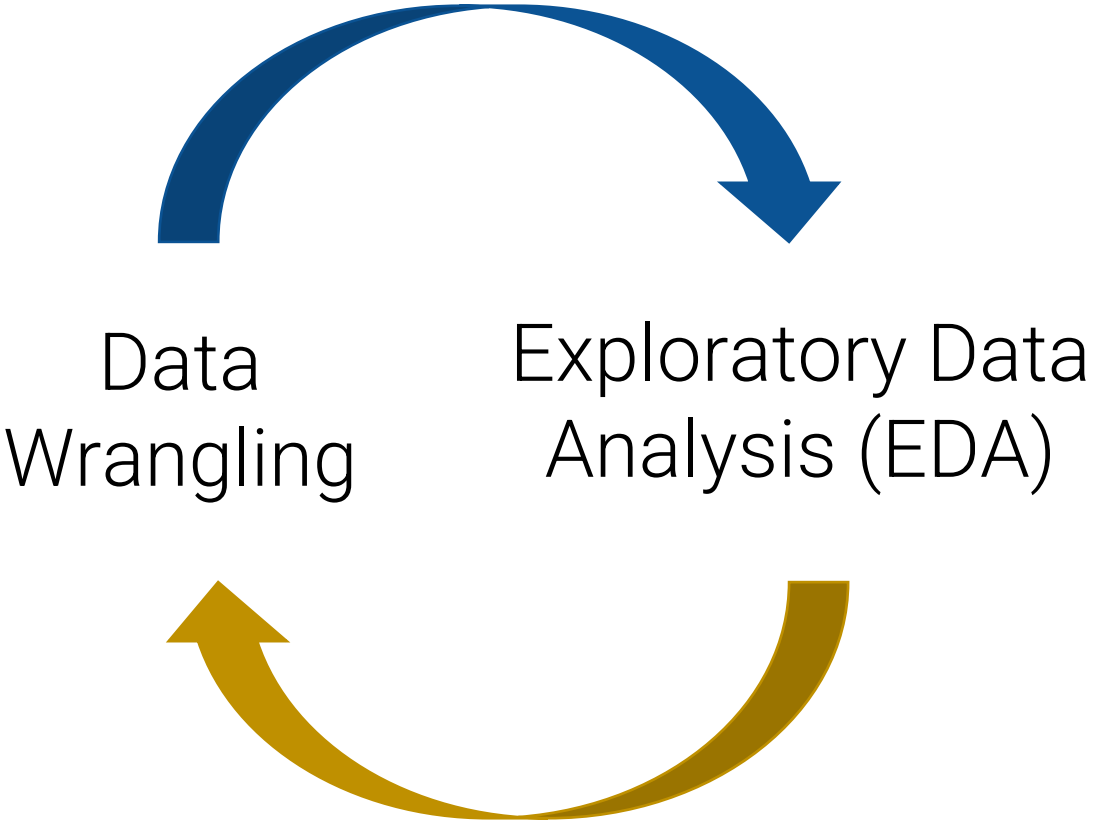
Data Wrangling and **EDA**: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- Faithfulness (and Missing Values)

EDA Demo: Mauna Loa CO2

The Infinite Loop of Data Science



Data Wrangling

Data Wrangling, or **Data Cleaning**:

The process of transforming **raw data** to facilitate subsequent analysis.

Often addresses **issues** like…

- structure / formatting
- missing or corrupted values
- unit conversion
- encoding text as numbers
- …

Sadly, data cleaning is a big part of data science…

Big Data Borat
@BigDataBorat

Following

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

Exploratory Data Analysis (EDA)

"Getting to Know the Data"

The process of **transforming**, **visualizing**, and **summarizing** data to:

- Build/confirm understanding of the data and its **provenance**
- Identify and address potential issues in the data
- Inform the subsequent analysis
- Discover *potential* hypothesis … (be careful…)

Provenance: origin of data; methodology by which data were produced

EDA is an open-ended analysis.

- Be willing to find something surprising!

John Tukey on EDA

John Tukey (1915-2000) was a Princeton Mathematician & Statistician and an **Early Data Scientist**.

Coined/Introduced:

- Fast Fourier Transform algorithm
- “Bit” : binary digit
- **Exploratory Data Analysis**

EDA is like **detective work**:

> Exploratory data analysis is an attitude, a state of flexibility, a willingness to look for those things that we believe are not there, as well as those that we believe to be there.

[Data Analysis & Statistics, Tukey 1965; Image from LIFE Magazine]

Key Data Properties to Consider in EDA

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- Faithfulness (and Missing Values)

EDA Demo: Mauna Loa CO2

What should we look for?

Key Data Properties to Consider in EDA

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Structure

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- **Structure**
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- Faithfulness (and Missing Values)

EDA Demo: Mauna Loa CO2

File Format

Variable Type

Multiple files
(Primary and Foreign Keys)

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

File Format

Variable Type

Multiple files
(Primary and Foreign Keys)

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

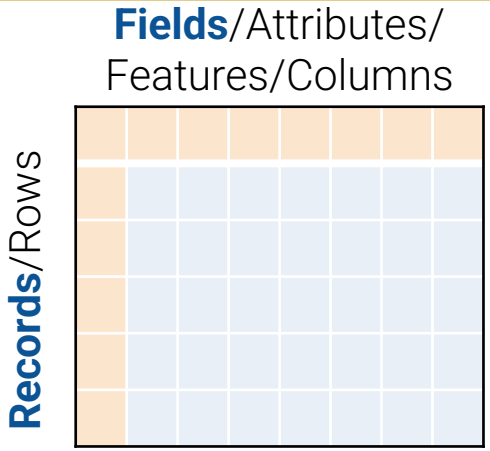
Faithfulness -- how well does the data capture “reality”

Rectangular Data

We prefer rectangular data for data analysis (why?)

- Regular structures are easy manipulate and analyze
- A big part of data cleaning is about transforming data to be more rectangular

Two kinds of rectangular data: **Tables** and **Matrices**.



Tables (a.k.a. dataframes in R/Python and relations in SQL)

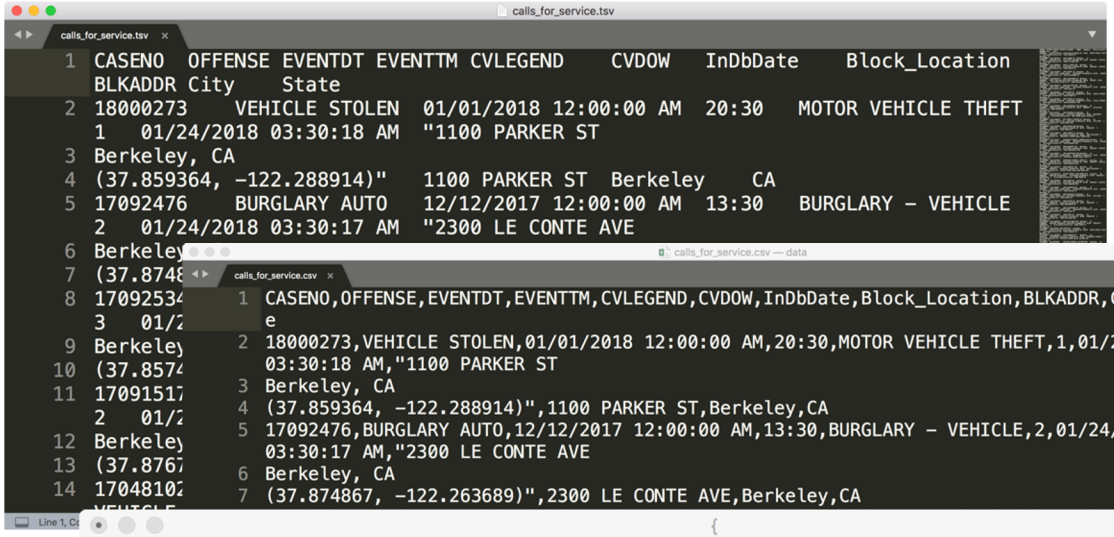
- Named columns with different types
- Manipulated using data transformation languages (map, filter, group by, join, …)

Matrices

- Numeric data of the same type (float, int, etc.)
- Manipulated using linear algebra

What are the differences?
Why would you use one over the other?

How are these data files formatted?

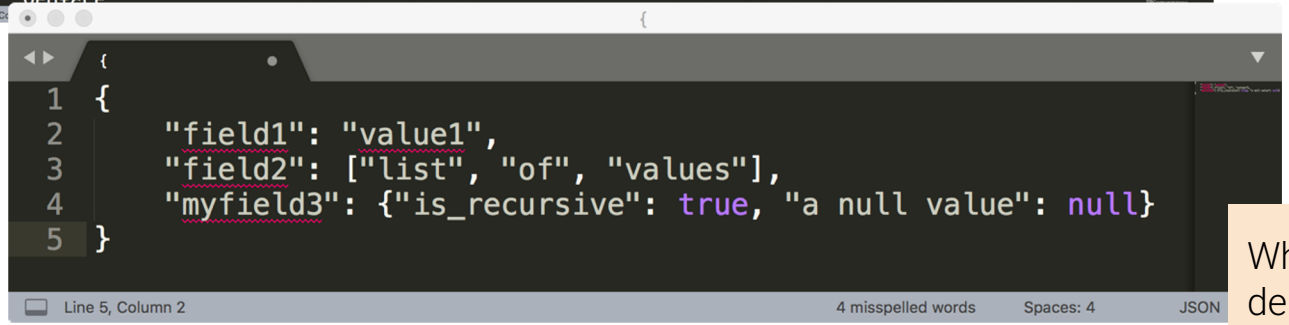


TSV

Tab separated values

CSV

Comma separated values



JSON

Which is the best? It depends on your use case.

Demo Slides

CSV: Comma Separated Values

Understand high-level structure:

1. How big is the data file?
2. How is the data file formatted?
3. How do we read the data into pandas?

CSV is a very common table file format:

- **Records** (rows) are delimited by a newline: `'\n'`, `"\r\n"`
- **Fields** (columns) are delimited by commas: `','`

Tabular data: **`pd.read_csv`**

Demo Slides

TSV: Tab Separated Values

Another common table file format.

- Records are delimited by a newline: `'\n'`, `"\r\n"`
- **Fields** are delimited by `'\t'` (tab)

`pd.read_csv`: Need to specify `delimiter='\t'`

Issues with CSVs and TSVs:

- Commas, tabs in records
- Quoting
- ...

Demo Slides

JSON: JavaScript Object Notation

A less common table file format.

- Very similar to Python dictionaries
- Strict formatting ”quoting” addresses some issues in CSV/TSV
- Can save metadata (data about the data) along with records in the same file

Issues

- Not rectangular
- Each record can have different fields
- Nesting means records can contain tables – complicated

Tabular data: Find the records using regular Python, then `pd.DataFrame`.

Demo Slides

What is the following file format?

Mauna Loa Observatory CO2 levels (NOAA)

How do we load these data into Pandas?

`pd.read_csv`? `pd.DataFrame`?

Tell me what to explore!
(raise hand/type in chat)

Often files will have mixed file formats, incorrect extensions or no extension at all.
You may need to explore the actual raw data file!

Other types of data formats

we will primarily work with CSV files, but there are other types of non-tabular data out in the wild.

XML (Extensible Markup Language)

```
<catalog>
  <plant type='a'>
    <common>Bloodroot</common>
    <botanical>Sanguinaria
canadensis</botanical>
    <zone>4</zone>
    <light>Mostly Shady</light>
    <price>2.44</price>

<availability>03/15/2006</availability>
    <description>
      <color>white</color>
      <petals>true</petals>
    </description>
    <indoor>true</indoor>
  </plant>
…
</catalog>
```

Nested structure

Log data (usually `.txt`)

```
169.237.46.168 - - [26/Jan/2014:10:47:58 -
0800] "GET /stat141/Winter04 HTTP/1.1" 301 328
"http://anson.ucdavis.edu/courses/"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.0; .NET CLR 1.1.4322)"

169.237.6.168 - - [8/Jan/2014:10:47:58 -0800]
"GET /stat141/Winter04/ HTTP/1.1" 200 2585
"http://anson.ucdavis.edu/courses/"
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.0; .NET CLR 1.1.4322)"
```

CSV? TSV?
JSON? XML?
None of the above?
Make your custom parser!

File Format

Variable Type

Multiple files
(Primary and Foreign Keys)

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Records and Variables/Fields

All data (regardless of format) is composed of **records**.
Each record has a set of **variables** (aka **fields**).

- Tabular: Records == Rows, Variables == Columns
- Non-Tabular: Create Records
  and wrangle into tabular data

Fields/Attributes/Features/Columns

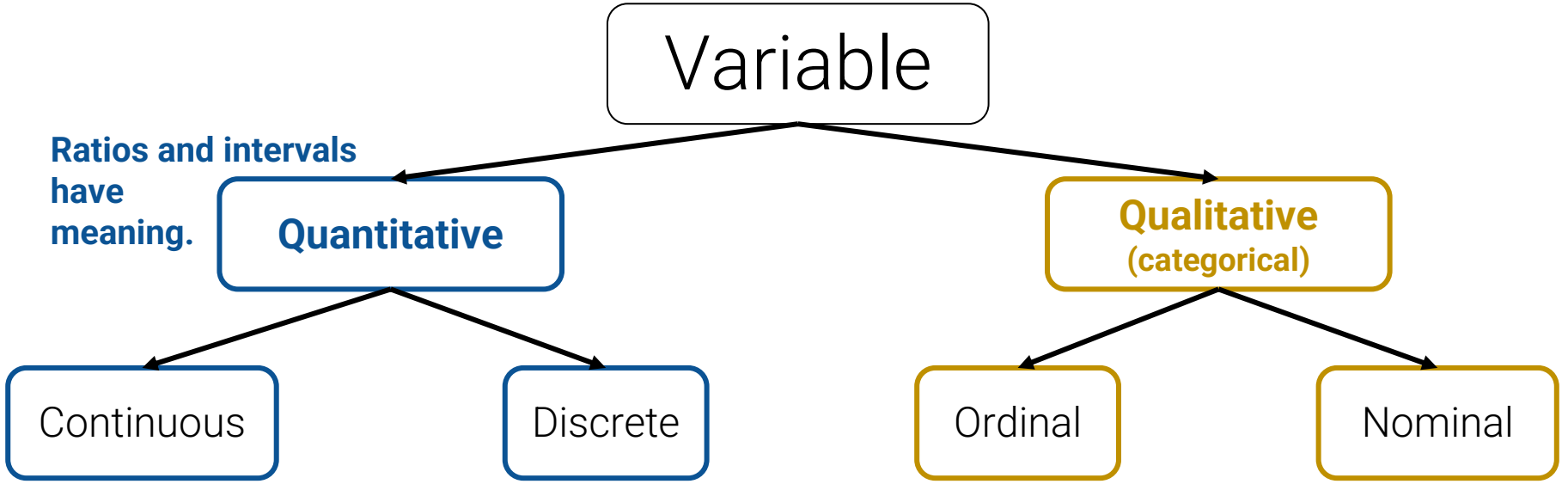
Records/Rows	business_id	business_name
0	835	Kam Po Kitchen
1	905	Working Girls' Cafe'

Variables are defined by their type (2 defs):

- **Storage type** in pandas:
  integer, floating point, boolean, object (string-like), etc.
  `df[colname].dtype`

- **Feature type**: conceptual notion of the information
  Use expert knowledge
  Explore data itself
  Consult data **codebook** (if it exists)

Variable Feature Types



Could be measured to arbitrary precision.

Examples:
- Price
- Temperature

Finite possible values

Examples:
- Number of siblings
- Yrs of education

Categories w/ordered levels; no consistent meaning to difference

Examples:
- Preferences
- Level of education

Categories w/ no specific ordering.

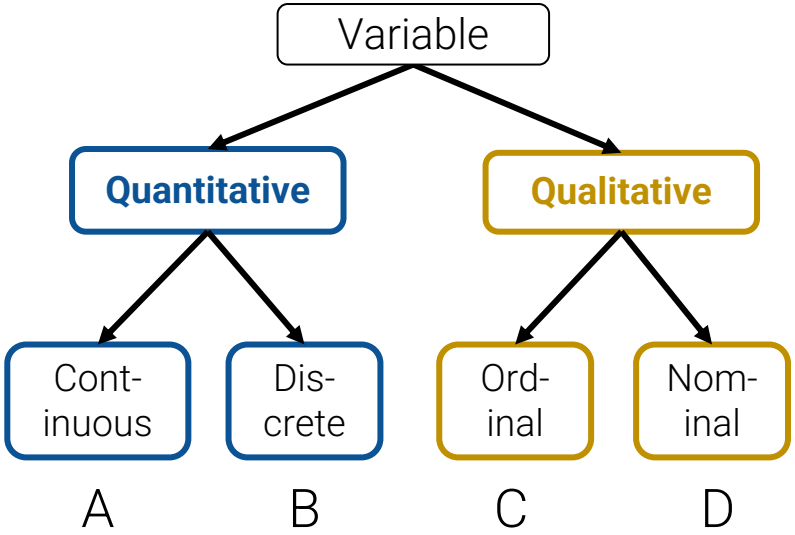
Examples:
- Political Affiliation
- Cal ID number

Note that **qualitative variables** could have numeric levels; conversely, **quantitative variables** could be stored as strings!

Class Exercise

What is the feature type of each variable?

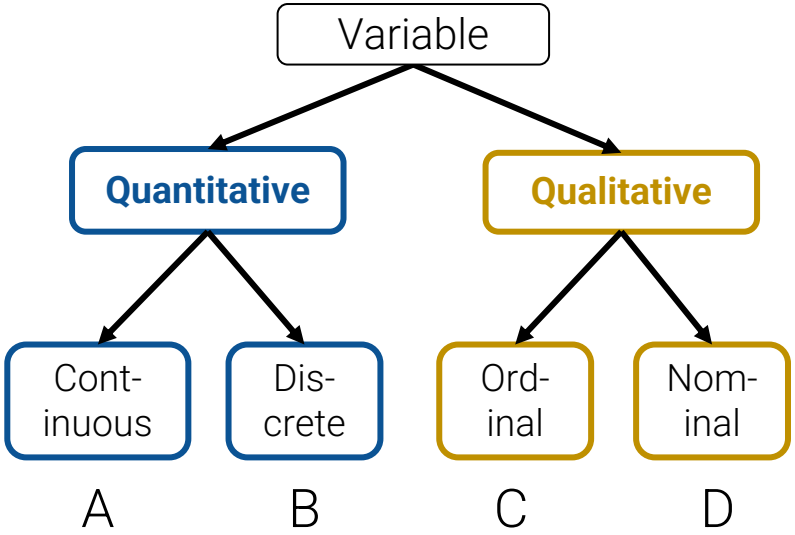
Q	Variable	Feature Type
1	CO_2 level (PPM)	
2	Number of siblings	
3	GPA	
4	Income bracket (low, med, high)	
5	Race	
6	Number of years of education	
7	Dianping (Food) Rating	



Class Exercise: Solutions

What is the feature type of each variable?

Q	Variable	Feature Type
1	CO_2 level (PPM)	**A. Quantitative Cont.**
2	Number of siblings	**B. Quantitative Discrete**
3	GPA	**A. Quantitative Cont.** *
4	Income bracket (low, med, high)	**C. Qualitative Ordinal**
5	Race	**D. Qualitative Nominal**
6	Number of years of education	**B. Quantitative Discrete** *
7	Dianping (Food) Rating	**C. Qualitative Ordinal** *



For this exercise,
The Feature Type variable is Qualitative Nominal.

File Format

Variable Type

Multiple files (Primary and Foreign Keys)

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Structure: Keys

Sometimes your data comes in multiple files:

- Often data will reference other pieces of data.

Primary key: the column or set of columns in a table that determine the values of the remaining columns

- Primary keys are unique
- Examples: SSN, ProductIDs, …

Primary Key

Purchases.csv

OrderNum	ProdID	Quantity
1	42	3
1	999	2
2	42	1

Orders.csv

OrderNum	CustID	Date
1	171345	8/21/2017
2	281139	8/30/2017

Products.csv

ProdID	Cost
42	3.14
999	2.72

Primary Key

Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

Structure: Keys

Sometimes your data comes in multiple files:

- Often data will reference other pieces of data.

Primary key: the column or set of columns in a table that determine the values of the remaining columns

- Primary keys are unique
- Examples: SSN, ProductIDs, …

Foreign keys: the column or sets of columns that reference primary keys in other tables.

You may need to join across tables!

`pd.merge`

Primary Key

Purchases.csv

OrderNum	ProdID	Quantity
1	42	3
1	999	2
2	42	1

Foreign Key

Orders.csv

OrderNum	CustID	Date
1	171345	8/21/2017
2	281139	8/30/2017

Products.csv

ProdID	Cost
42	3.14
999	2.72

Primary Key

Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

Are the data in a standard format or encoding?

- Tabular data: CSV, TSV, Excel, SQL
- Nested data: JSON or XML

Are the data organized in **records** or nested?

- Can we define records by parsing the data?
- Can we reasonably un-nest the data?

Does the data reference other data?

- Can we join/merge the data?
- Do we need to?

What are the **fields** in each record?

- How are they encoded? (e.g., strings, numbers, binary, dates …)
- What is the type of the data?

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Summary

You will do the most data wrangling when analyzing the structure of your data.

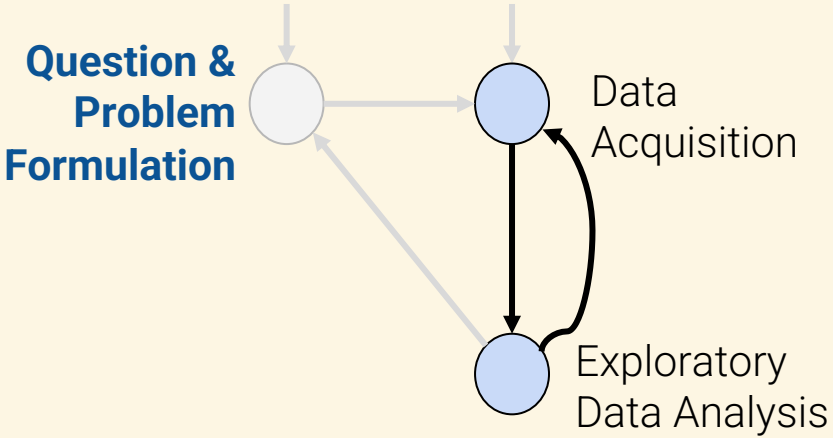
Granularity, Scope, Temporality

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- **Granularity, Scope, Temporality**
- Faithfulness (and Missing Values)

EDA Demo: Mauna Loa CO2



Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Granularity

What does each **record** represent?

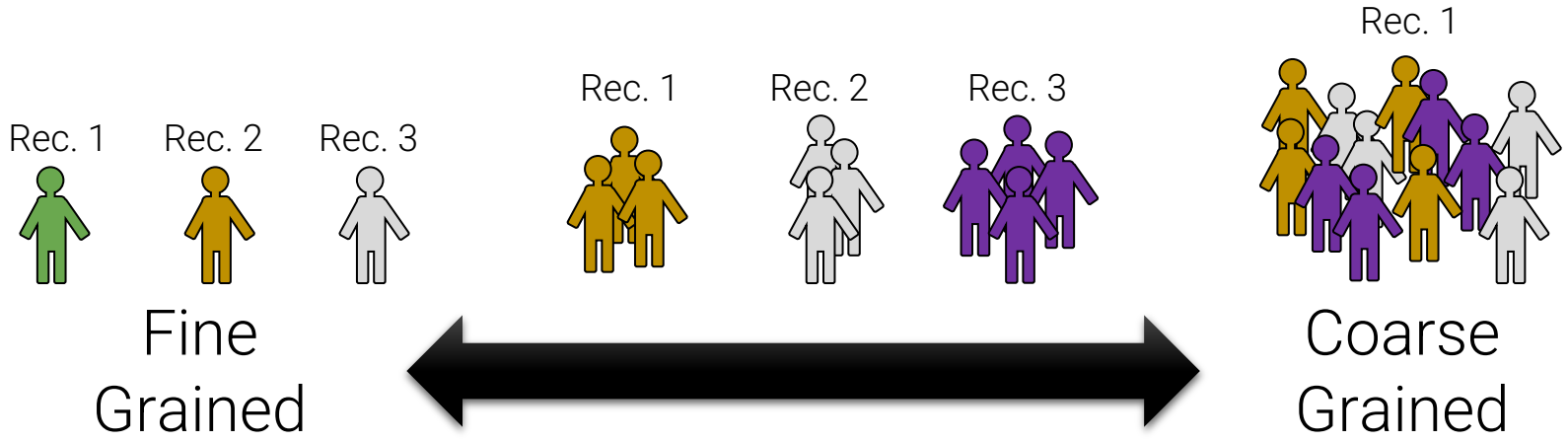
- Examples: a purchase, a person, a group of users

Do all records capture granularity at the same level?

- Some data will include summaries (aka **rollups**) as records

If the data are **coarse**, how were the records aggregated?

- Sampling, averaging, …



Scope

Does my data cover my area of interest?

- **Example**: I am interested in studying crime in China but I only have Shanghai crime data.

Are my data too expansive?

- **Example**: I am interested in student grades for ECE 4710J but have student grades for all statistics classes.
- **Solution: Filtering** ⇒ Implications on sample?
  - If the data is a sample I may have poor coverage after filtering …

Does my data cover the right time frame?

- More on this in Temporality…

Scope

Does my data cover my area of interest?

- Example: I am interested in studying crime in China but I only have Shanghai crime data.

Are my data too expansive?

- Example: I am interested in student grades for ECE 4710J but have student grades for all statistics classes.
- Solution: Filtering ⇒ Implications on sample?
  - If the data is a sample I may have poor coverage after filtering …

Does my data cover the right time frame?

- More on this in Temporality…

(recall) The **sampling frame** is the population from which the data were sampled. Note that this may not be the population of interest.

How complete/incomplete is the frame (and its data)?

- How is the frame/data situated in place?
- How well does the frame/data capture reality?
- How is the frame/data situated in time?

Temporality

Data changes – when was the data collected/last updated?

Periodicity — Is there periodicity? Diurnal (24-hr) patterns?

What is the meaning of the time and date fields? A few options:

- When the “event” happened?
- When the data was collected or was entered into the system?
- Date the data was copied into a database? (look for many matching timestamps)

Time depends on where! (**time zones** & daylight savings)

- Learn to use `datetime` python library and Pandas `dt` accessors
- Regions have different datestring representations: 07/08/09?

Are there strange null values?

- E.g., **January 1st 1970**, January 1st 1900…?

Temporality: Unix Time / POSIX Time

Time measured in seconds since **January 1st 1970**

- Minus leap seconds …

Unix time follows Coordinated Universal Time (UTC)

- International time standard
- Measured at 0 degrees latitude
  - Similar to Greenwich Mean Time (GMT)
- No daylight savings
- Time codes

Time Zones:

- San Francisco (UTC-8) without daylight savings

Feb 1, 2022 3:00pm Pacific
1643756400

https://en.wikipedia.org/wiki/Coordinated_Universal_Time

Faithfulness (and Missing Values)

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- **Faithfulness (and Missing Values)**

EDA Demo: Mauna Loa CO2

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Faithfulness: Do I trust this data?

Does my data contain **unrealistic or "incorrect" values**?

- Dates in the future for events in the past
- Locations that don't exist
- Negative counts
- Misspellings of names
- Large outliers

Does my data violate **obvious dependencies**?

- E.g., age and birthday don't match

Was the data **entered by hand**?

- Spelling errors, fields shifted …
- Did the form require all fields or provide default values?

Are there obvious signs of **data falsification**?

- Repeated names, fake looking email addresses, repeated use of uncommon names or fields.

Signs that your data may not be faithful (and Solutions)

Truncated data

- Early Microsoft Excel limits: 65536 Rows, 255 Columns
- Soln: be aware of consequences in analysis ⇒ how did truncation affect sample?

Time Zone Inconsistencies

- Soln 1: convert to a common timezone (e.g., UTC)
- Soln 2: convert to the timezone of the location – useful in modeling behavior.

Duplicated Records or Fields

- Soln: identify and eliminate (use primary key) ⇒ implications on sample?

Spelling Errors

- Soln: Apply corrections or drop records not in a dictionary ⇒ implications on sample?

Units not specified or consistent

- Solns: Infer units, check values are in reasonable ranges for data

Missing Data

- See next slide

How to Address Missing Data/Default Values

Drop records with missing values

- Probably most common
- **Caution**: check for biases induced by dropped values
  - Missing or corrupt records might be related to something of interest

Examples

```
" "
0, -1
999, 12345
1970, 1900
NaN
Null
```

NaN: "Not a Number"

Imputation: Inferring missing values

- **Average Imputation**: replace with an average value
  - Which average? Often use closest related subgroup mean.
- **Hot deck imputation**: replace with a random value
  - Choose a random value from the subgroup and use it for the missing value.

Other Suggestions

1. **Drop** missing values but check for **induced bias** (use domain knowledge)
2. Directly **model missing values** during future analysis

You'll see Suggestion 1 in next week's lab; you'll see Suggestion 2 in the wild.

Demo: Mauna Loa CO2 EDA

Data Wrangling and Exploratory Data Analysis: An Infinite Loop

Key Data Properties to Consider in EDA

- Structure
  - File format
  - Variable types
  - Primary and Foreign Keys
- Granularity, Scope, Temporality
- Faithfulness (and Missing Values)

EDA Demo: Mauna Loa CO2

Demo Slides

What are our Variable Feature Types?

EDA step:
Understand what each record, each feature represents

From file description:

- All measurement variables (**`average`**, **`interpolated`**, **`trend`**) are monthly mean CO2 monthly mean mole fraction, i.e. monthly average CO2 ppm (parts per million)
  - Computed from daily means
- **`#days`**: Number of daily means in a month (i.e., # days equipment worked)

What are the first three columns? How do these columns define each record?

Demo Slides

The Search for the Missing Values

EDA step:
Hypothesize why these values were missing, then use that knowledge to decide whether to drop or impute missing values

From file description:

- **`-99.99`**: missing monthly average **`Avg`**
- **`-1`**: missing value for # **`days`** that the equipment was in operation that month.

Which approach? Drop, NaN, Interpolate

- All 3 are probably fine since few missing values, but we choose interpolation

Granularity of data: What do we want to report? How long is the timescale?

Demo Slides

A Discussion on Data Granularity

From the description:

- Monthly measurements are averages of average day measurements.
- The NOAA GML website has datasets for daily/hourly measurements too.

Which granularity to present?

- You can always go from finer-grained to coarser-grained data (**`groupby.agg`**), but not vice versa.
- Fine-grained data can be computationally expensive: 61 years of seconds is a lot of records!

You want the granularity of your data to match your research question.

Summary: How do you do EDA/Data Wrangling?

Examine **data and metadata**:

- What is the date, size, organization, and structure of the data?

Examine each **field/attribute/dimension** individually

Examine **pairs of related dimensions**

- Stratifying earlier analysis: break down grades by major …

Along the way:

- **Visualize**/summarize the data
- **Validate assumptions** about data and collection process
- Identify and **address anomalies**
- Apply data transformations and corrections (next lecture)
- **Record everything you do**! (why?)