Data 100

Lecture 5: Data Cleaning & Exploratory Data Analysis

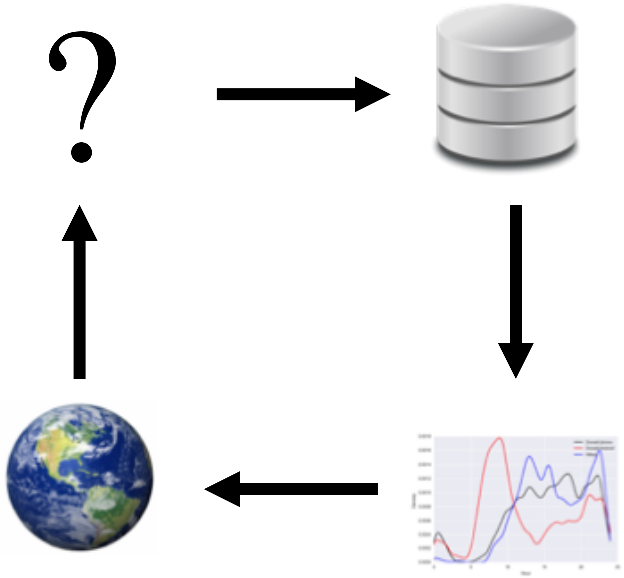
Slides by:

Joseph E. Gonzalez, Deb Nolan, & Joe Hellerstein

jegonzal@berkeley.edu

deborah_nolan@berkeley.edu

hellerstein@berkeley.edu



Last Week

Jupyter Notebooks

https://www.nbcnews.com/news/world/giant-pandas-are-no-longer-endangered-n643336

Pandas and Jupyter Notebooks

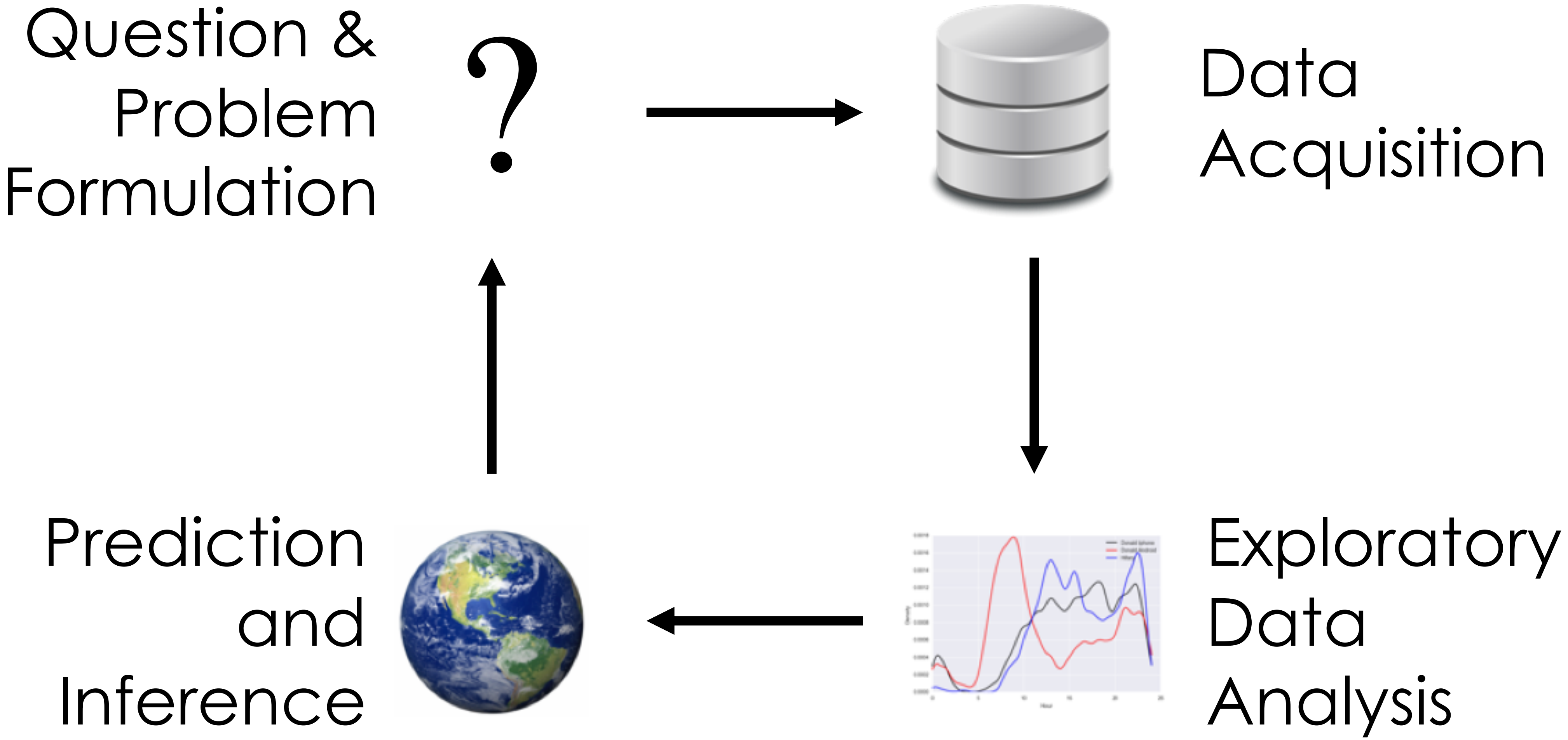
- Reviewed Jupyter Notebook Environment
- Introduced DataFrame concepts
  - **Series:** A named column of data with an index
  - **Indexes:** The mapping from keys to rows
  - **DataFrame:** collection of series with common index
- Dataframe access methods
  - **Filtering** on predicts and **slicing**
  - **df.loc**: location by index
  - **df.iloc**: location by integer address
  - **groupby** & **pivot** (we will review these again today)

Today

Congratulations!

You have **collected** or **been given** a box of data?

What do you do next?



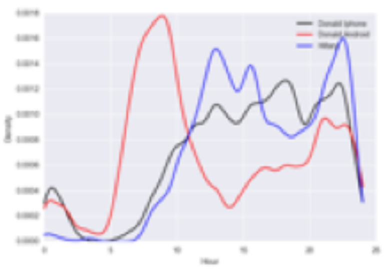
Question & Problem Formulation

Data Acquisition

Exploratory Data Analysis

Prediction and Inference

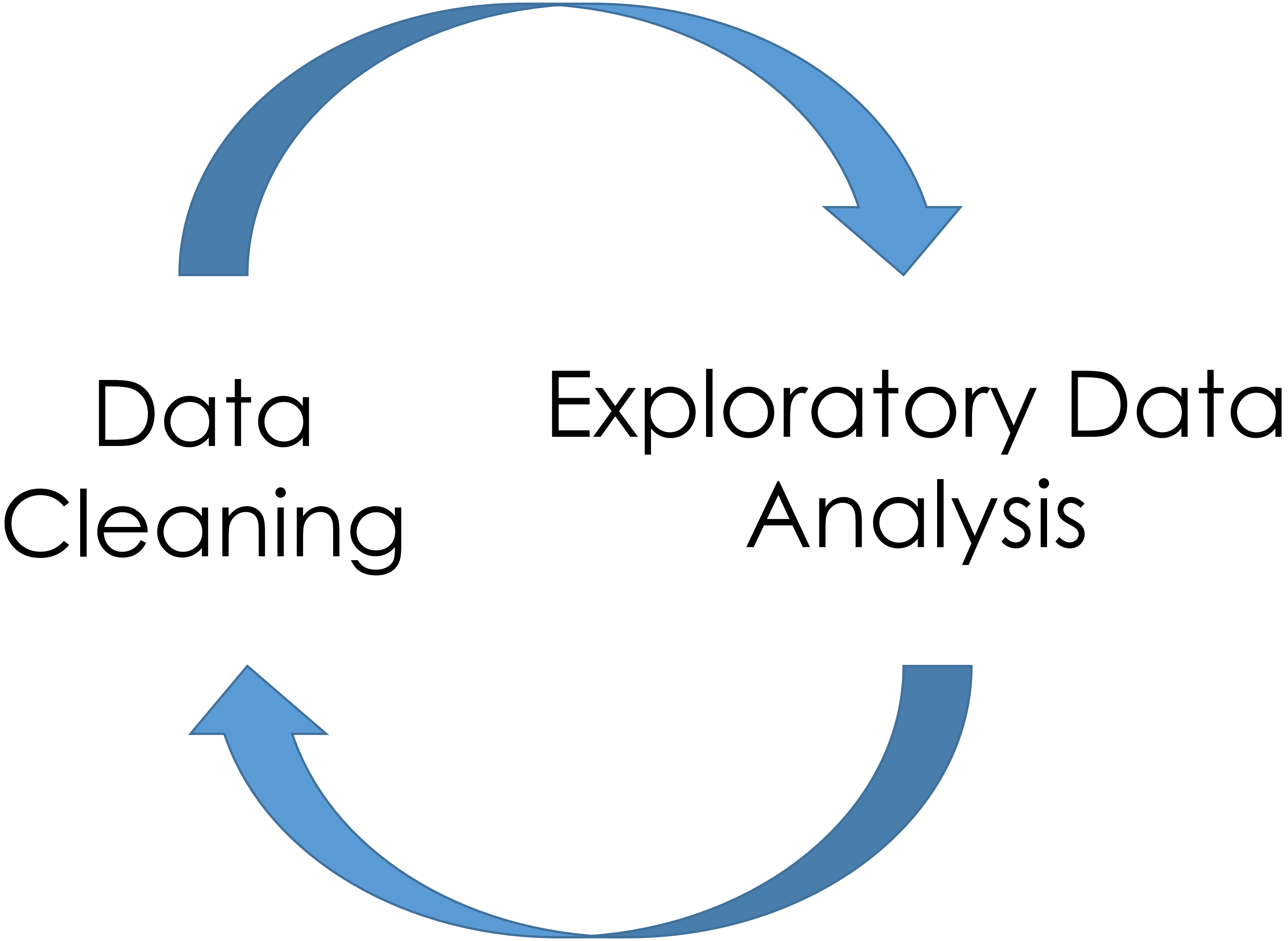
Data Acquisition



Exploratory Data Analysis

Topics For Lecture Today

- ➢ Understanding the Data
  - ➢ Data Cleaning
  - ➢ Exploratory Data Analysis (EDA)
  - ➢ Basic data visualization
- ➢ Common Data Anomalies
  - ➢ … and how to fix them



Data Cleaning

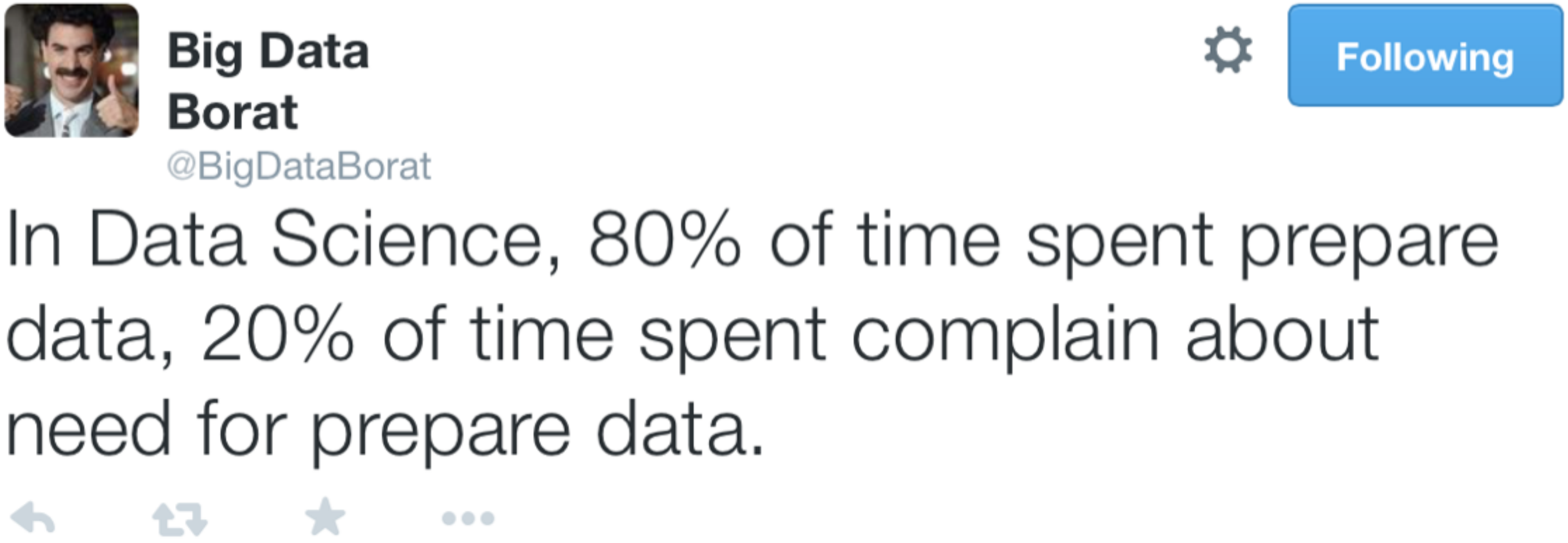
Exploratory Data Analysis

… the infinite loop of data science.

Data Cleaning

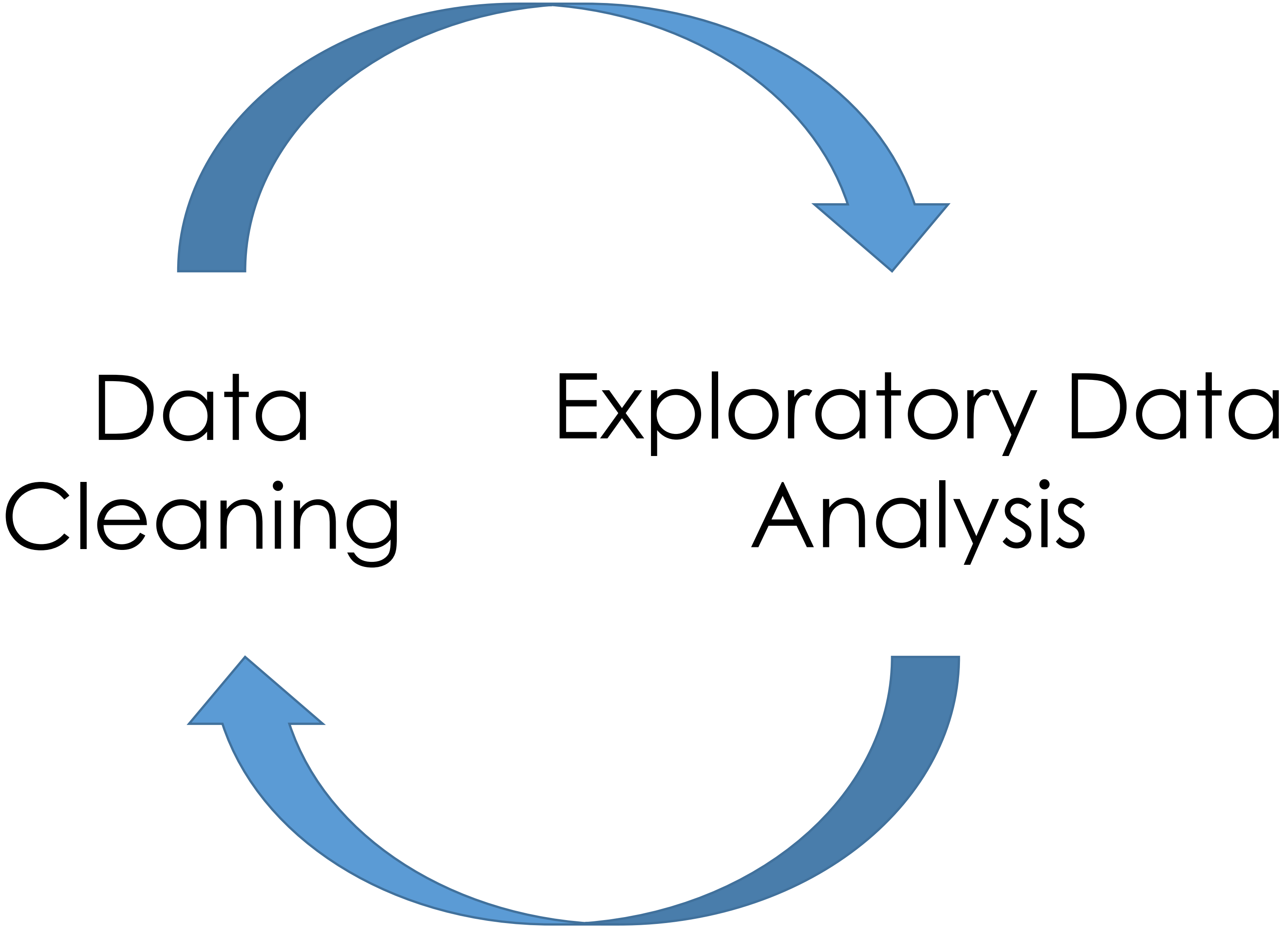
- The process of transforming raw data to facilitate subsequent analysis
- Data cleaning often addresses
  - structure / formatting
  - missing or corrupted values
  - unit conversion
  - encoding text as numbers
  - …
- Sadly data cleaning is a big part of data science…

- Data cleaning often addresses
  - 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

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



Data Cleaning

Exploratory Data Analysis

… the infinite loop of data science.

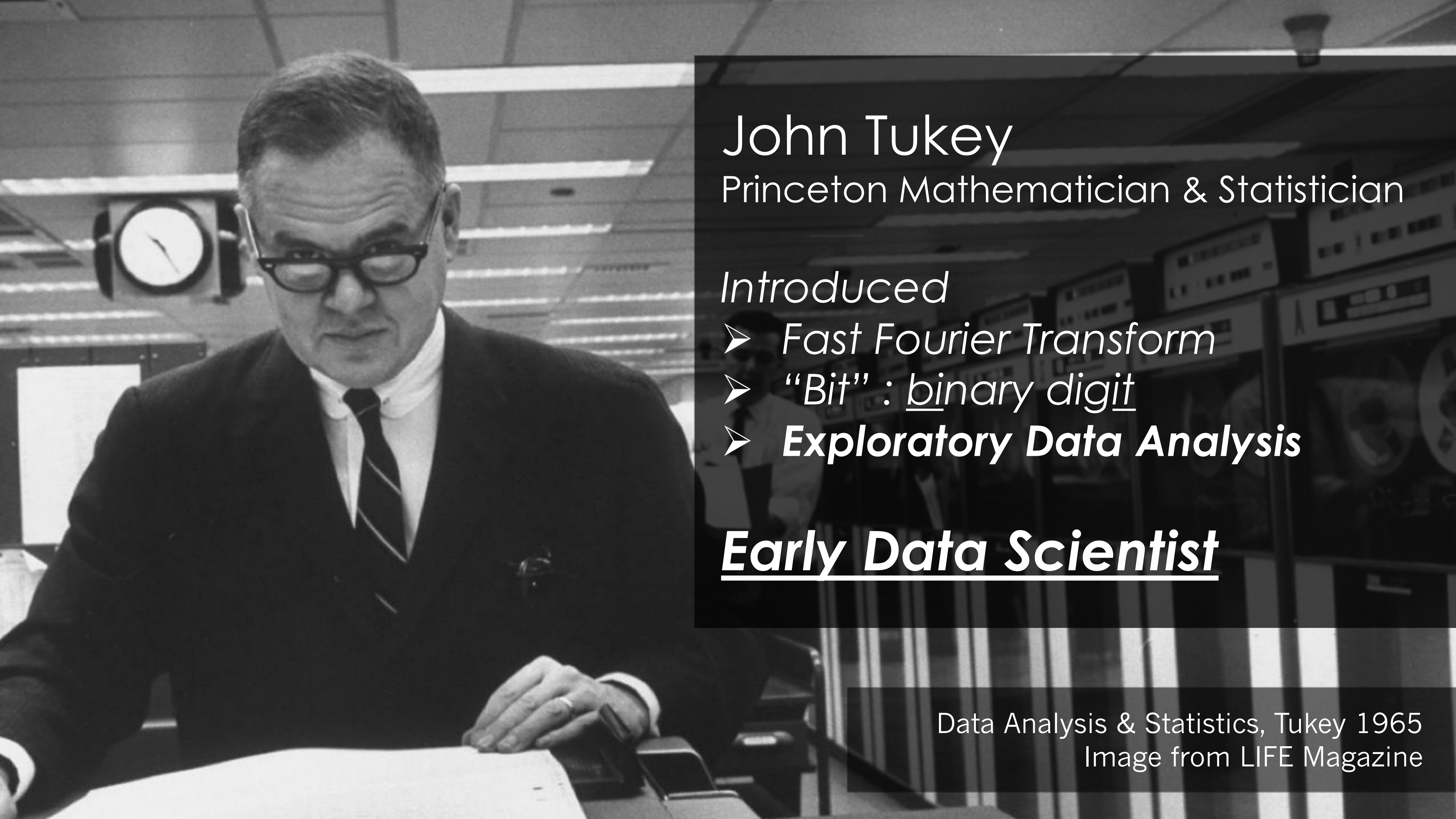
Exploratory Data Analysis (EDA)

"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)

- **EDA is an open ended analysis**
  - Be willing to find something surprising



John Tukey

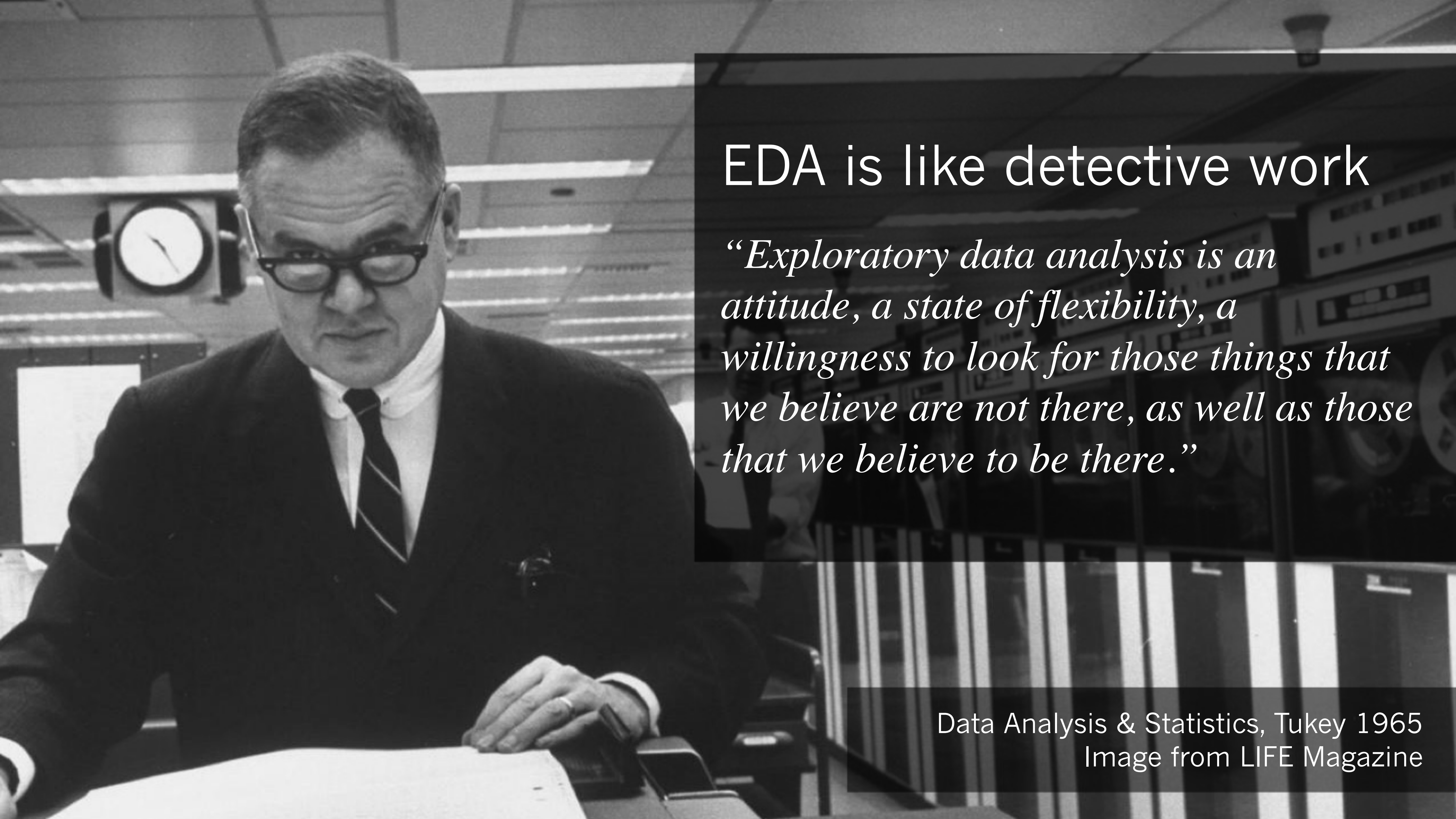
Princeton Mathematician & Statistician

Introduced

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

Early Data Scientist

Data Analysis & Statistics, Tukey 1965
Image from LIFE Magazine



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

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”*

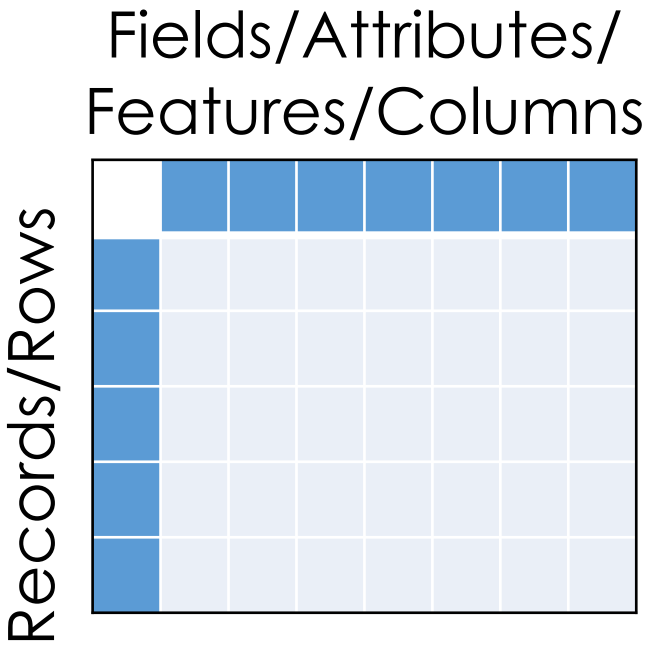
Key Data Properties to Consider in EDA

- **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”*

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

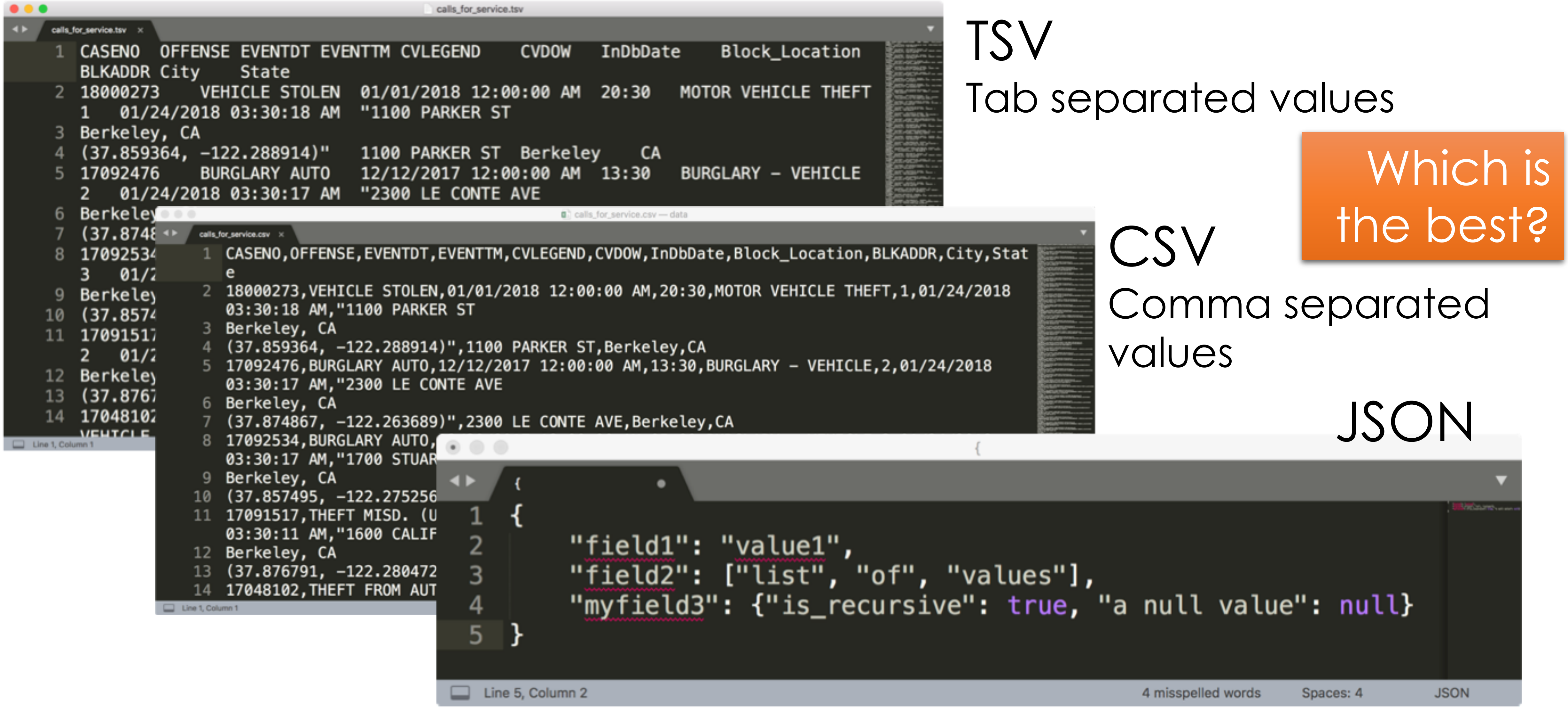


Two kinds of rectangular data: *Tables and Matrices*

(what are the differences?)

1. **Tables** (a.k.a. data-frames in R/Python and relations in SQL)
   - Named columns with different types
   - Manipulated using data transformation languages (map, filter, group by, join, …)
2. **Matrices**
   - Numeric data of the same type
   - Manipulated using linear algebra

How are these data files formatted?



```
CASENO  OFFENSE EVENTDT EVENTTM CVLEGEND     CVDOW   InDbDate     Block_Location
BLKADDR City     State
18000273     VEHICLE STOLEN  01/01/2018 12:00:00 AM  20:30    MOTOR VEHICLE THEFT
1    01/24/2018 03:30:18 AM  "1100 PARKER ST
Berkeley, CA
(37.859364, -122.288914)"   1100 PARKER ST  Berkeley     CA
17092476     BURGLARY AUTO   12/12/2017 12:00:00 AM  13:30    BURGLARY - VEHICLE
2    01/24/2018 03:30:17 AM  "2300 LE CONTE AVE
Berkeley
(37.874
1709253
3    01/2
Berkeley
(37.857
1709151
2    01/2
Berkeley
(37.876
1704810
VEHICLE
```

TSV

Tab separated values

```
CASENO,OFFENSE,EVENTDT,EVENTTM,CVLEGEND,CVDOW,InDbDate,Block_Location,BLKADDR,City,Stat
e
18000273,VEHICLE STOLEN,01/01/2018 12:00:00 AM,20:30,MOTOR VEHICLE THEFT,1,01/24/2018
03:30:18 AM,"1100 PARKER ST
Berkeley, CA
(37.859364, -122.288914)",1100 PARKER ST,Berkeley,CA
17092476,BURGLARY AUTO,12/12/2017 12:00:00 AM,13:30,BURGLARY - VEHICLE,2,01/24/2018
03:30:17 AM,"2300 LE CONTE AVE
Berkeley, CA
(37.874867, -122.263689)",2300 LE CONTE AVE,Berkeley,CA
17092534,BURGLARY AUTO,
03:30:17 AM,"1700 STUAR
Berkeley, CA
(37.857495, -122.275256
17091517,THEFT MISD. (U
03:30:11 AM,"1600 CALIF
Berkeley, CA
(37.876791, -122.280472
17048102,THEFT FROM AUT
```

Which is the best?

CSV

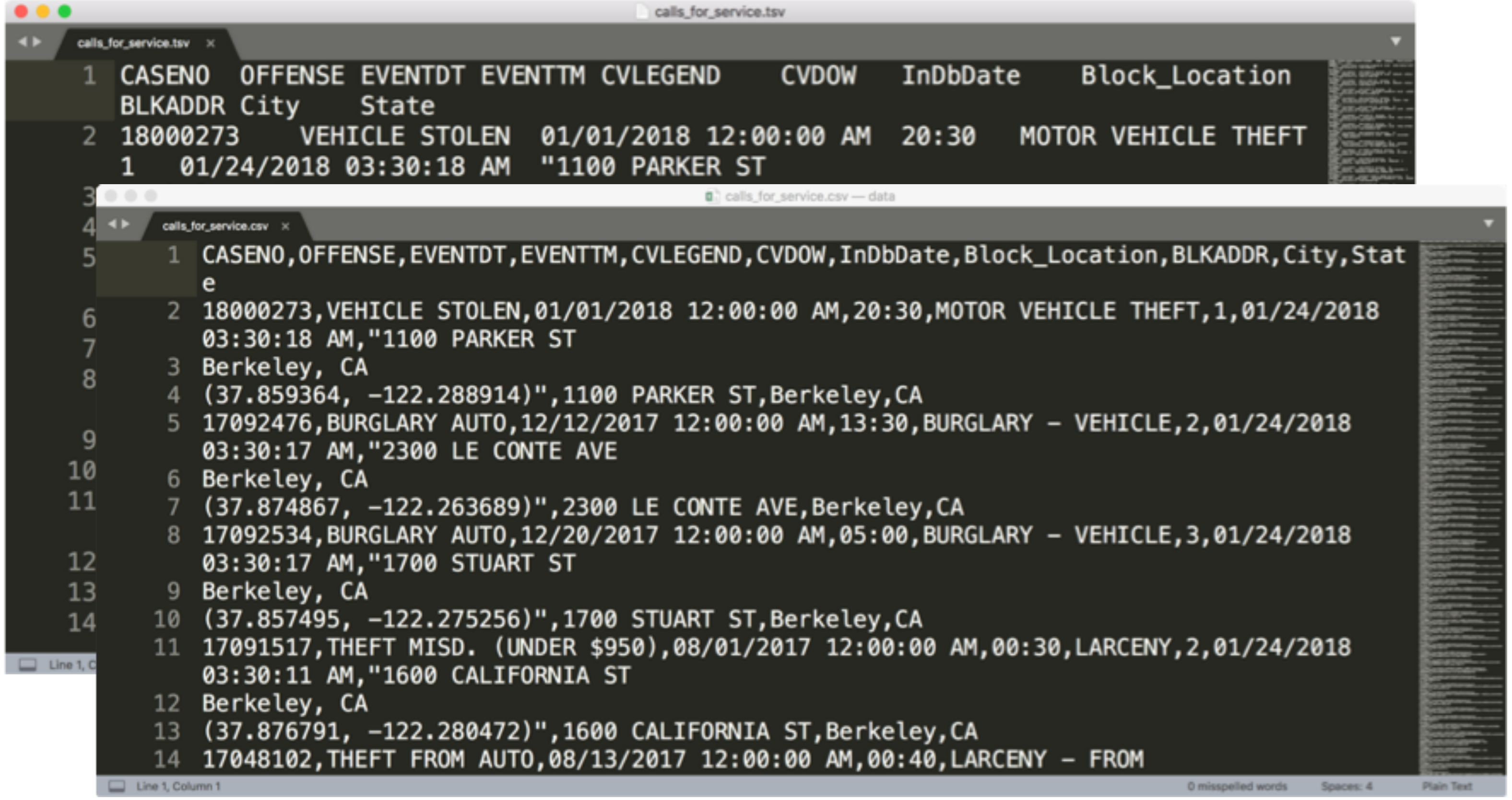
Comma separated values

JSON

```json
{
    "field1": "value1",
    "field2": ["list", "of", "values"],
    "myfield3": {"is_recursive": true, "a null value": null}
}
```

Comma and Tab Separated Values Files

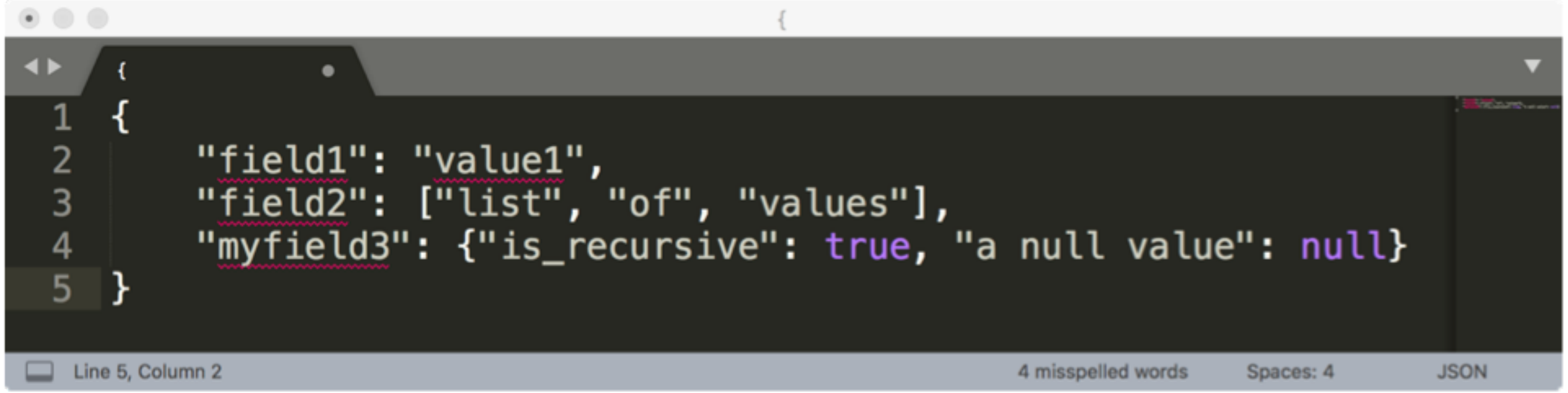
- Tabular data where
  - records are delimited by a *newline*: “\n”, “\r\n”
  - Fields are delimited by ‘,’ (comma) or ‘\t’ (tab)
- Very Common!
- Issues?
  - Commas, tabs in records
  - Quoting
  - …



```
CASENO  OFFENSE EVENTDT EVENTTM CVLEGEND    CVDOW   InDbDate    Block_Location  BLKADDR City    State
18000273    VEHICLE STOLEN  01/01/2018 12:00:00 AM  20:30   MOTOR VEHICLE THEFT 1   01/24/2018 03:30:18 AM  "1100 PARKER ST
```

```
CASENO,OFFENSE,EVENTDT,EVENTTM,CVLEGEND,CVDOW,InDbDate,Block_Location,BLKADDR,City,State
18000273,VEHICLE STOLEN,01/01/2018 12:00:00 AM,20:30,MOTOR VEHICLE THEFT,1,01/24/2018 03:30:18 AM,"1100 PARKER ST
Berkeley, CA
(37.859364, -122.288914)",1100 PARKER ST,Berkeley,CA
17092476,BURGLARY AUTO,12/12/2017 12:00:00 AM,13:30,BURGLARY - VEHICLE,2,01/24/2018 03:30:17 AM,"2300 LE CONTE AVE
Berkeley, CA
(37.874867, -122.263689)",2300 LE CONTE AVE,Berkeley,CA
17092534,BURGLARY AUTO,12/20/2017 12:00:00 AM,05:00,BURGLARY - VEHICLE,3,01/24/2018 03:30:17 AM,"1700 STUART ST
Berkeley, CA
(37.857495, -122.275256)",1700 STUART ST,Berkeley,CA
17091517,THEFT MISD. (UNDER $950),08/01/2017 12:00:00 AM,00:30,LARCENY,2,01/24/2018 03:30:11 AM,"1600 CALIFORNIA ST
Berkeley, CA
(37.876791, -122.280472)",1600 CALIFORNIA ST,Berkeley,CA
17048102,THEFT FROM AUTO,08/13/2017 12:00:00 AM,00:40,LARCENY - FROM
```

JavaScript Object Notation (JSON)



```
{
    "field1": "value1",
    "field2": ["list", "of", "values"],
    "myfield3": {"is_recursive": true, "a null value": null}
}
```

- Widely used file format for nested data
  - Natural maps to python dictionaries (many tools for loading)
  - Strict formatting "quoting" addresses some issues in CSV/TSV
- Issues
  - Each record can have different fields
  - Nesting means records can contain records → complicated

XML (another kind of nested data)

```
<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

We will study XML later in the class

Log data

Is this a csv file? tsv? JSON/XML?

```
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)"
```

Data can be **split across files** and **reference other data**.

Structure: Keys

- 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.

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 ..

Merging/joining data across tables

Joining two tables

OrderNum	ProdID	Name
1	42	Gum
2	999	NullFood
2	42	Towel

x

OrderId	Cust Name	Date
1	Joe	8/21/2017
2	Arthur	8/14/2017

Left “key”

Right “key”

OrderNum	ProdID	Name	OrderId	Cust Name	Date
1	42	Gum	1	Joe	8/21/2017
~~1~~	~~42~~	~~Gum~~	~~2~~	~~Arthur~~	~~8/14/2017~~
~~2~~	~~999~~	~~NullFood~~	~~1~~	~~Joe~~	~~8/21/2017~~
2	999	NullFood	2	Arthur	8/14/2017
~~2~~	~~42~~	~~Towel~~	~~1~~	~~Joe~~	~~8/21/2017~~
2	42	Towel	2	Arthur	8/14/2017

Drop rows that don’t match on the key

OrderNum	ProdID	Name
1	42	Gum
2	999	NullFood
2	42	Towel

X

OrderId	Cust Name	Date
1	Joe	8/21/2017
2	Arthur	8/14/2017

Left “key”

Right “key”

OrderNum	ProdID	Name	OrderId	Cust Name	Date
1	42	Gum	1	Joe	8/21/2017
~~1~~	~~42~~	~~Gum~~	~~2~~	~~Arthur~~	~~8/14/2017~~
~~2~~	~~999~~	~~NullFood~~	~~1~~	~~Joe~~	~~8/21/2017~~
2	999	NullFood	2	Arthur	8/14/2017
~~2~~	~~42~~	~~Towel~~	~~1~~	~~Joe~~	~~8/21/2017~~
2	42	Towel	2	Arthur	8/14/2017

Drop rows that don’t match on the key

➡

OrderNum	ProdID	Name	OrderId	Cust Name	Date
1	42	Gum	1	Joe	8/21/2017
2	999	NullFood	2	Arthur	8/14/2017
2	42	Towel	2	Arthur	8/14/2017

Pandas Merge

Demo

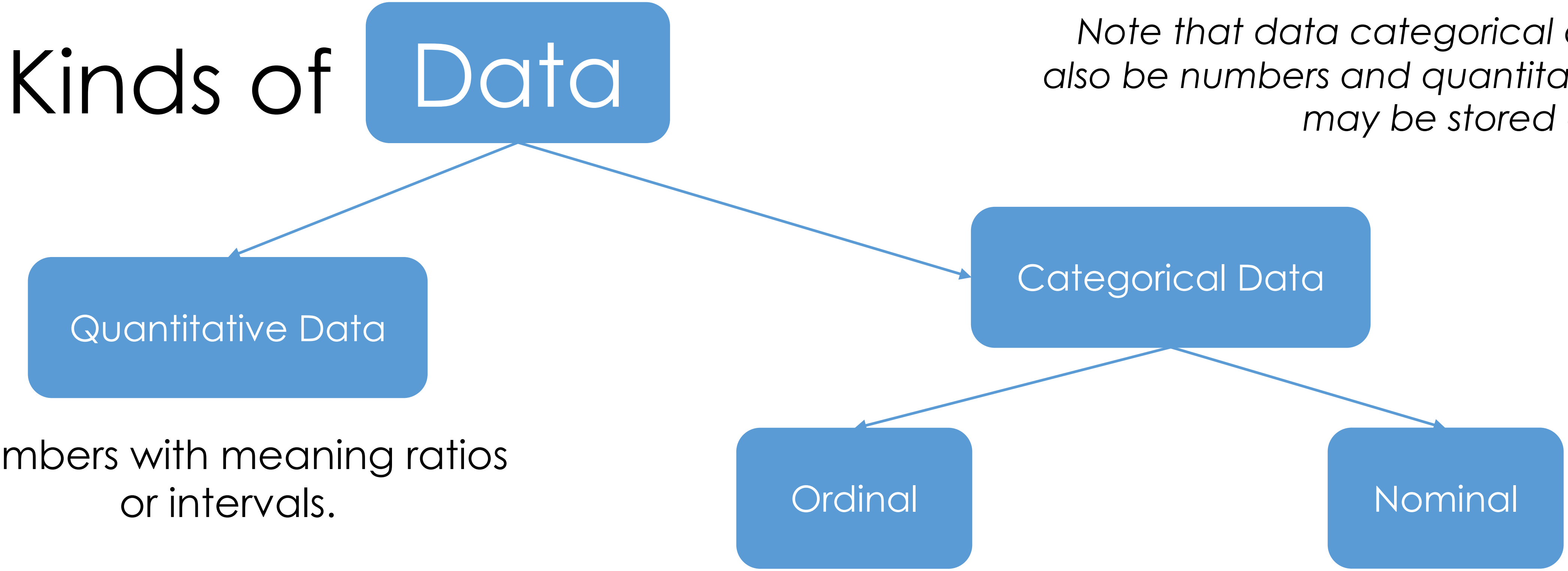
https://www.popsci.com/pandas-have-cute-markings-because-their-food-supply-sucks

Questions to ask about ***Structure***

- 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"?
  - No: Can we define records by parsing the data?
- Are the data nested? (records contained within records…)
  - Yes: Can we reasonably un-nest the data?
- Does the data reference other data?
  - Yes: can we join/merge the data
- What are the fields in each record?
  - How are they encoded? (e.g., strings, numbers, binary, dates …)
  - What is the type of the data?

Kinds of Data

Note that data categorical data can also be numbers and quantitative data may be stored as strings.



Data

Quantitative Data

Categorical Data

Ordinal

Nominal

Numbers with meaning ratios or intervals.

Examples:

- Price
- Quantity
- Temperature
- Date
- …

Categories with orders but no consistent meaning if magnitudes or intervals

Examples:

- Preferences
- Level of education
- …

Categories with no specific ordering.

Examples:

- Political Affiliation
- Product Type
- Cal Id
- …

Quiz

http://bit.ly/ds100-sp18-eda

- Price in dollars of a product?
  - [✓ (A)] Quantitative, (B) Ordinal, (C) Nominal
- Star Rating on Yelp?
  - (A) Quantitative, [✓ (B)] Ordinal, (C) Nominal
- Date an item was sold?
  - [✓ (A)] Quantitative, (B) Ordinal, (C) Nominal
- What is your Credit Card Number?
  - (A) Quantitative, (B) Ordinal, [✓ (C)] Nominal

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"*

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"*

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 as records

- If the data are coarse how was it aggregated?
  - Sampling, averaging, …

- What kinds of aggregation is possible/desirable?
  - From individual people to demographic groups?
  - From individual events to totals across time or regions?
  - Hierarchies (city/county/state, second/minute/hour/days)

- Understanding and manipulating granularity can help reveal patterns.

Reviewing
Group By and Pivot

Manipulating Granularity: Group By

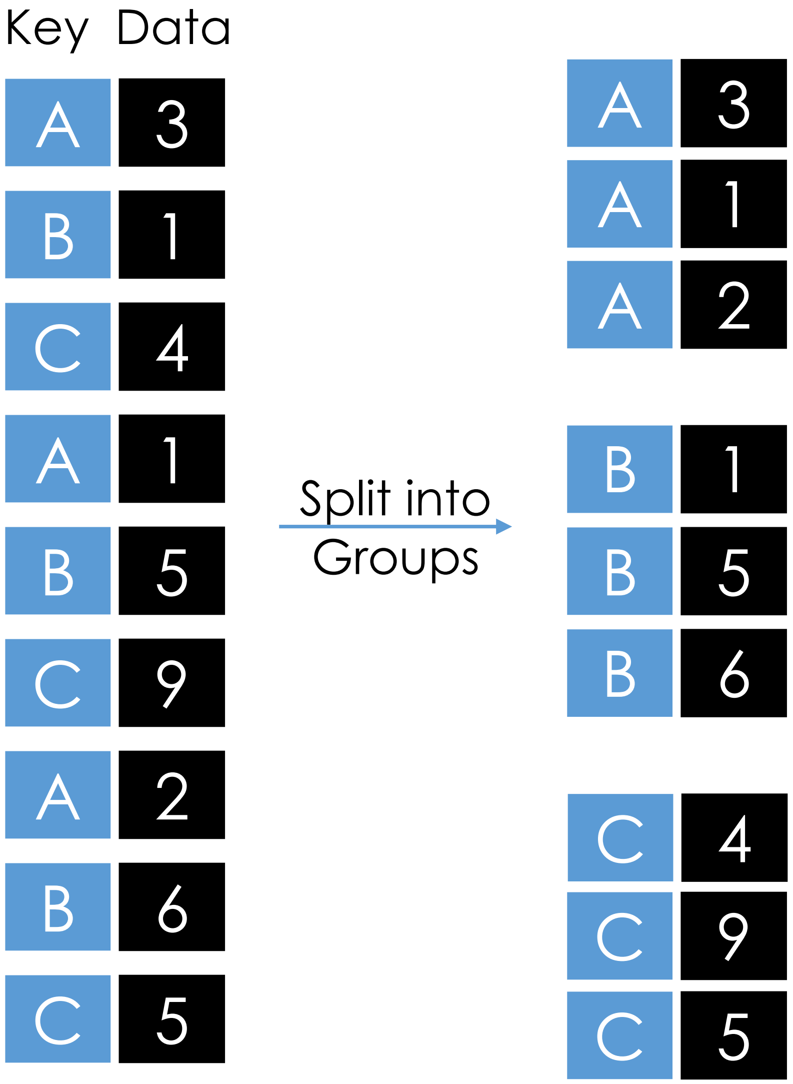
Key	Data
A	3
B	1
C	4
A	1
B	5
C	9
A	2
B	6
C	5

Manipulating Granularity: Group By

Key	Data
A	3
B	1
C	4
A	1
B	5
C	9
A	2
B	6
C	5

A	3
A	1
A	2

Manipulating Granularity: Group By



Key	Data
A	3
B	1
C	4
A	1
B	5
C	9
A	2
B	6
C	5

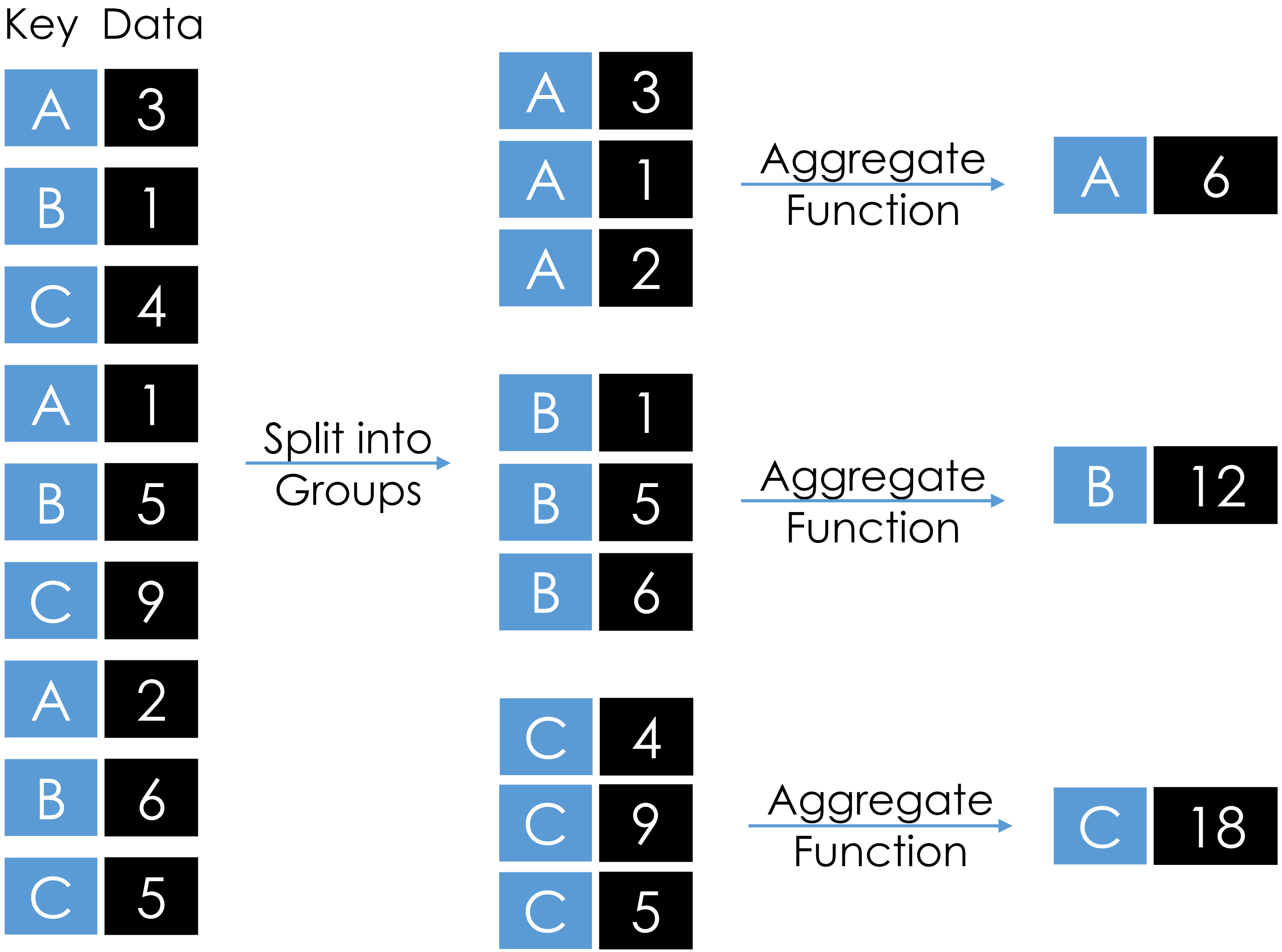
Split into Groups

A	3
A	1
A	2

B	1
B	5
B	6

C	4
C	9
C	5

Manipulating Granularity: Group By



Key	Data
A	3
B	1
C	4
A	1
B	5
C	9
A	2
B	6
C	5

Split into Groups

A	3
A	1
A	2

Aggregate Function

A	6

B	1
B	5
B	6

Aggregate Function

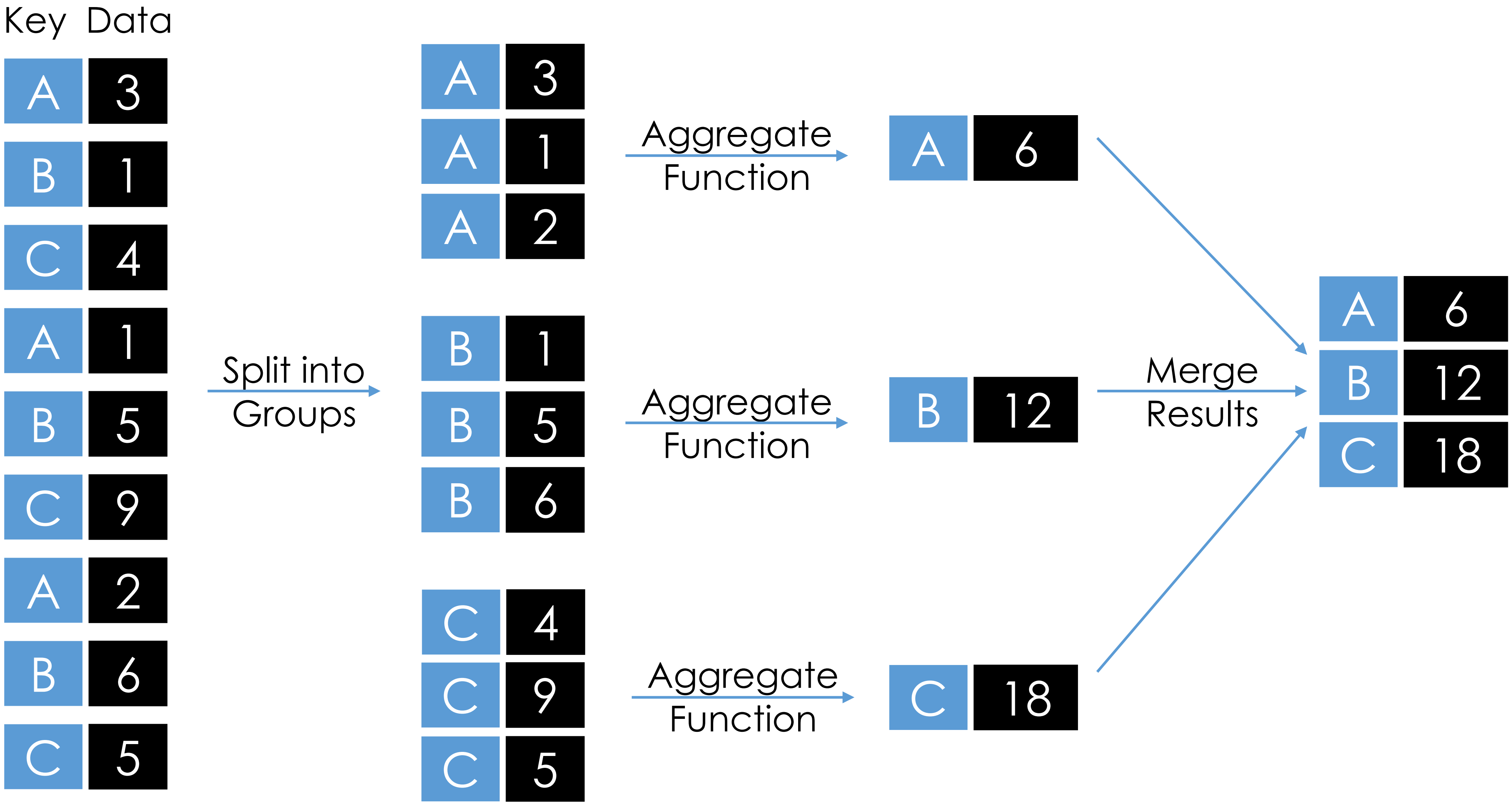
B	12

C	4
C	9
C	5

Aggregate Function

C	18

Manipulating Granularity: Group By



Key	Data
A	3
B	1
C	4
A	1
B	5
C	9
A	2
B	6
C	5

Split into Groups

A	3
A	1
A	2

B	1
B	5
B	6

C	4
C	9
C	5

Aggregate Function

A	6

Aggregate Function

B	12

Aggregate Function

C	18

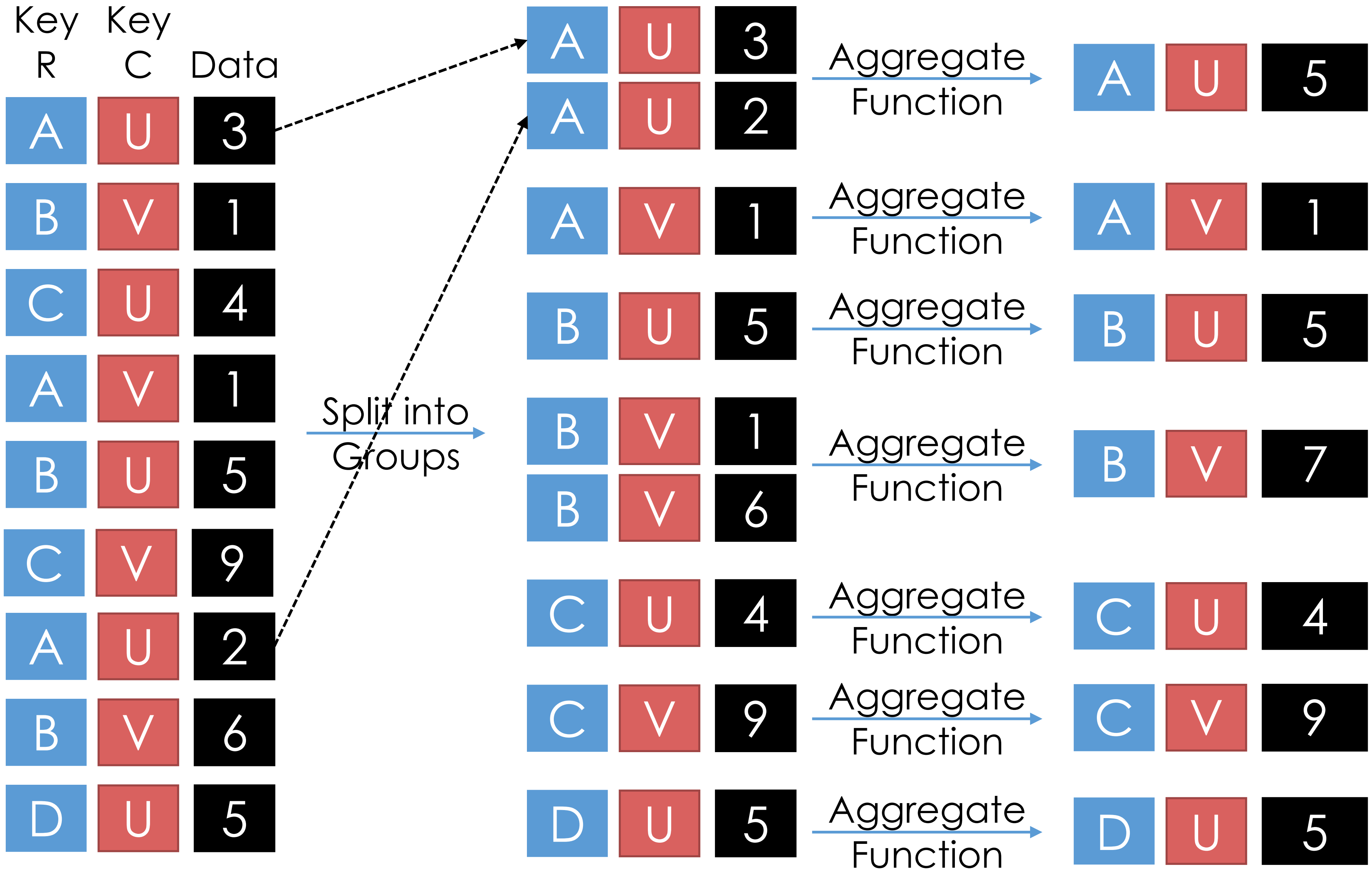
Merge Results

A	6
B	12
C	18

Manipulating Granularity: Pivot

Key R	Key C	Data
A	U	3
B	V	1
C	U	4
A	V	1
B	U	5
C	V	9
A	U	2
B	V	6
D	U	5

Manipulating Granularity: Pivot



Key R	Key C	Data
A	U	3
B	V	1
C	U	4
A	V	1
B	U	5
C	V	9
A	U	2
B	V	6
D	U	5

Split into Groups

A	U	3
A	U	2

Aggregate Function

A	U	5

A	V	1

Aggregate Function

A	V	1

B	U	5

Aggregate Function

B	U	5

B	V	1
B	V	6

Aggregate Function

B	V	7

C	U	4

Aggregate Function

C	U	4

C	V	9

Aggregate Function

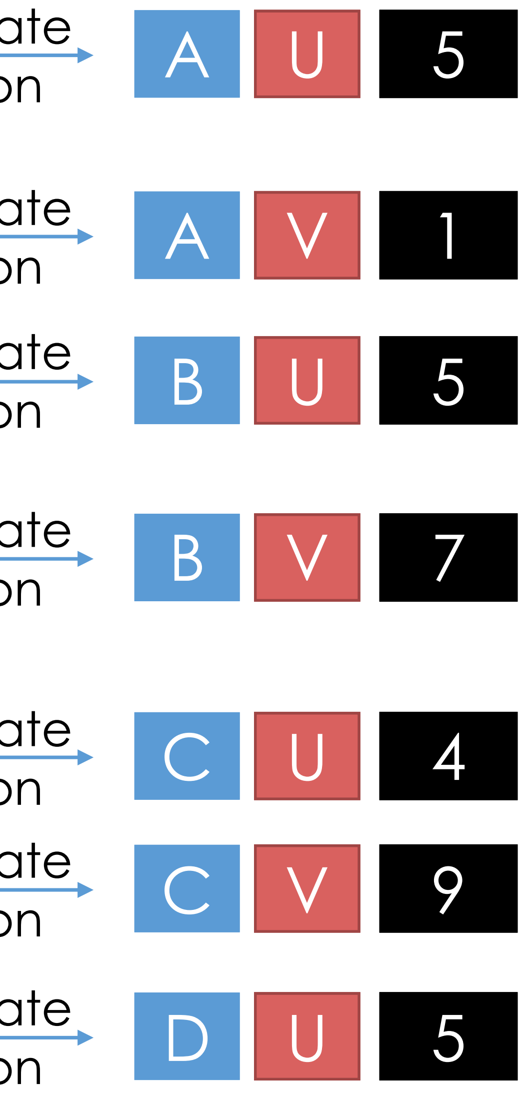
C	V	9

D	U	5

Aggregate Function

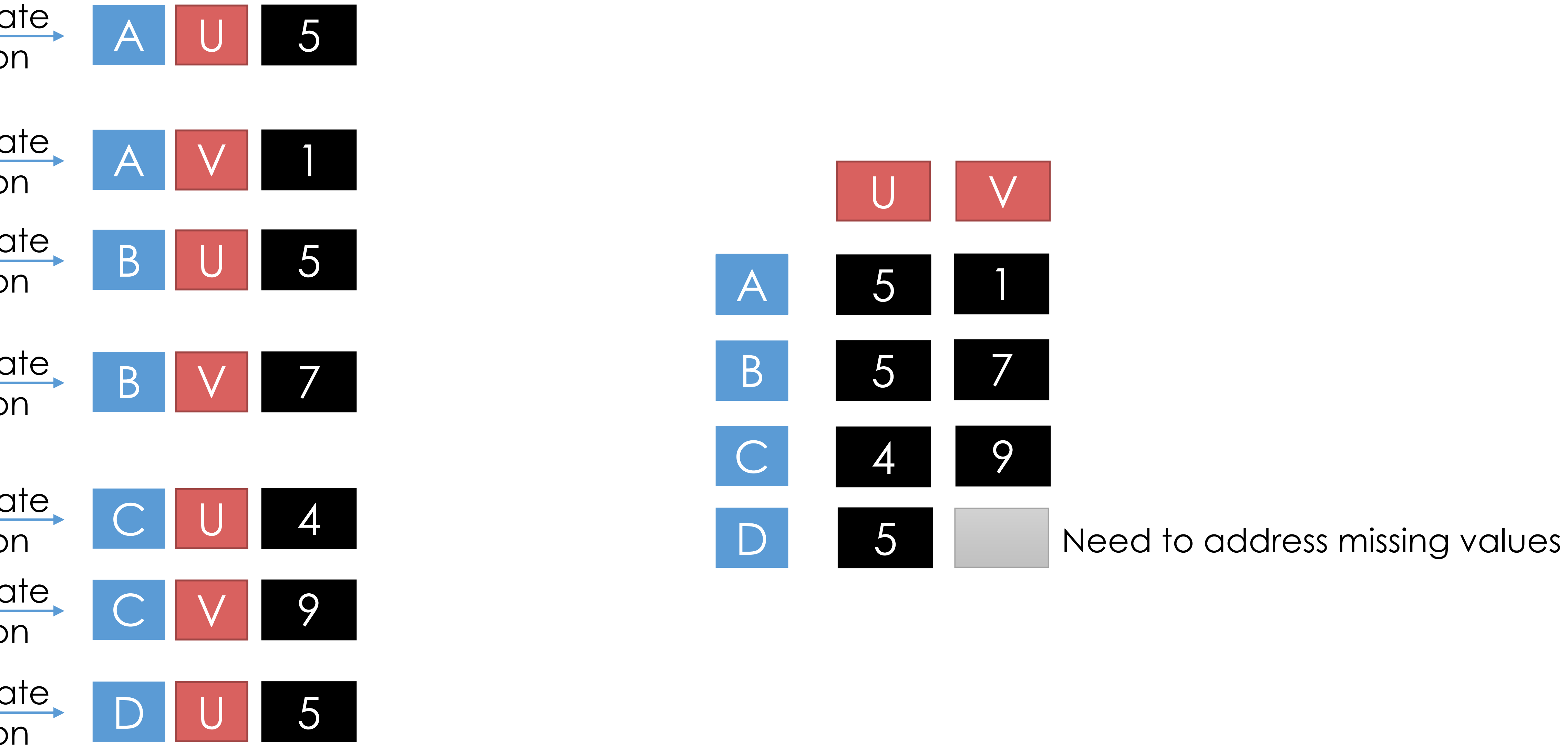
D	U	5

Manipulating Granularity: Pivot



ate on	A	U	5
ate on	A	V	1
ate on	B	U	5
ate on	B	V	7
ate on	C	U	4
ate on	C	V	9
ate on	D	U	5

Manipulating Granularity: Pivot



ate on	A	U	5
ate on	A	V	1
ate on	B	U	5
ate on	B	V	7
ate on	C	U	4
ate on	C	V	9
ate on	D	U	5

	U	V
A	5	1
B	5	7
C	4	9
D	5	

Need to address missing values

Demo

http://abcnews.go.com/Lifestyle/silly-baby-panda-falls-flat-face-public-debut/story?id=42481478

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"*

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"*

Scope

- Does my data cover my area of interest?
  - **Example:** *I am interested in studying crime in California but I only have Berkeley crime data.*
- Is my data too expansive?
  - **Example:** *I am interested in student grades for DS100 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 …

To be continued …

In the next lecture