Data Science (COSE471) Spring 2021

Data Cleaning and EDA

Dept. of Computer Science and Engineering

Korea University

"This is not what I meant when I said 'we need better data cleansing!'"

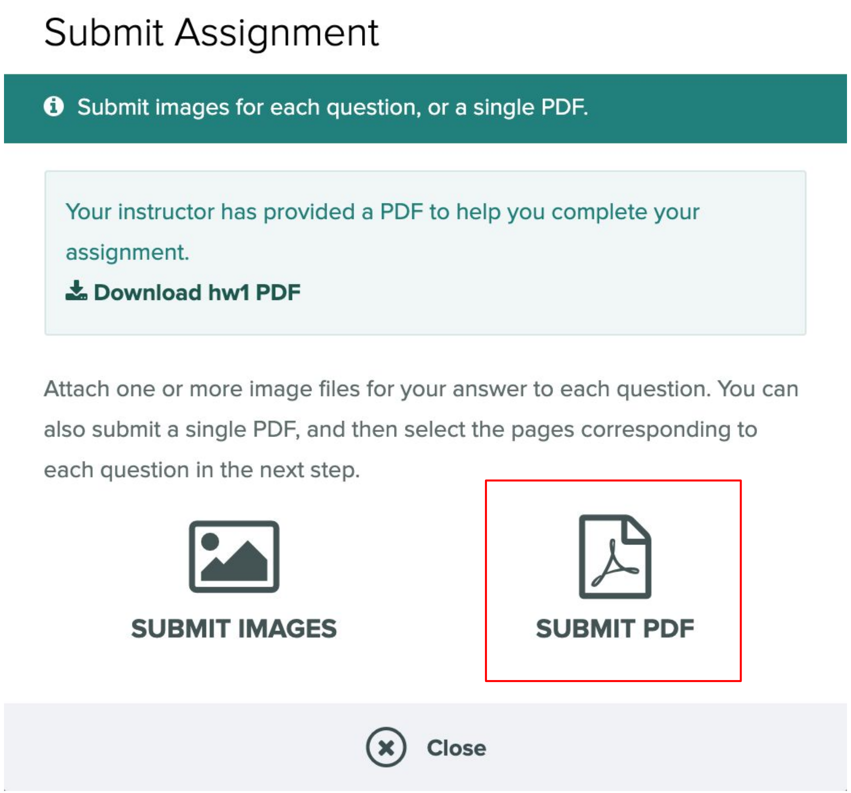
www.iwaysoftware.com/go/dataquality

* This material is adapted from Berkeley CS 100 (ds100.org) and may be copyrighted by them.

Announcements

- Homework #1 is due on March 26 (Fri) 11:59pm
- Office hour sign-up sheet is posted in Blackboard

How to submit your hw to Gradescope?



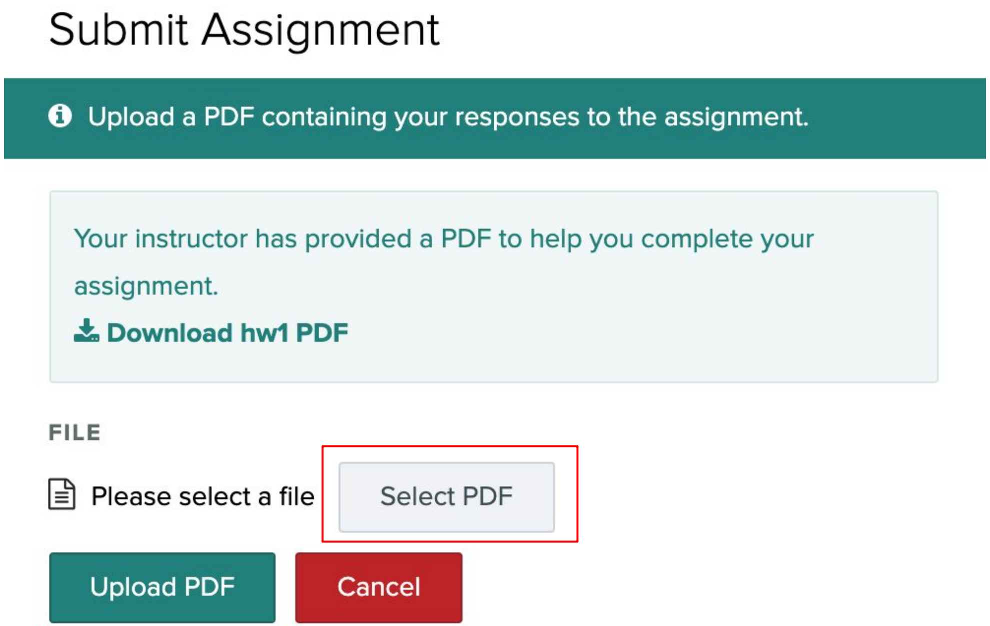
Submit Assignment

Submit images for each question, or a single PDF.

Your instructor has provided a PDF to help you complete your assignment.

Download hw1 PDF

Attach one or more image files for your answer to each question. You can also submit a single PDF, and then select the pages corresponding to each question in the next step.



Submit Assignment

Upload a PDF containing your responses to the assignment.

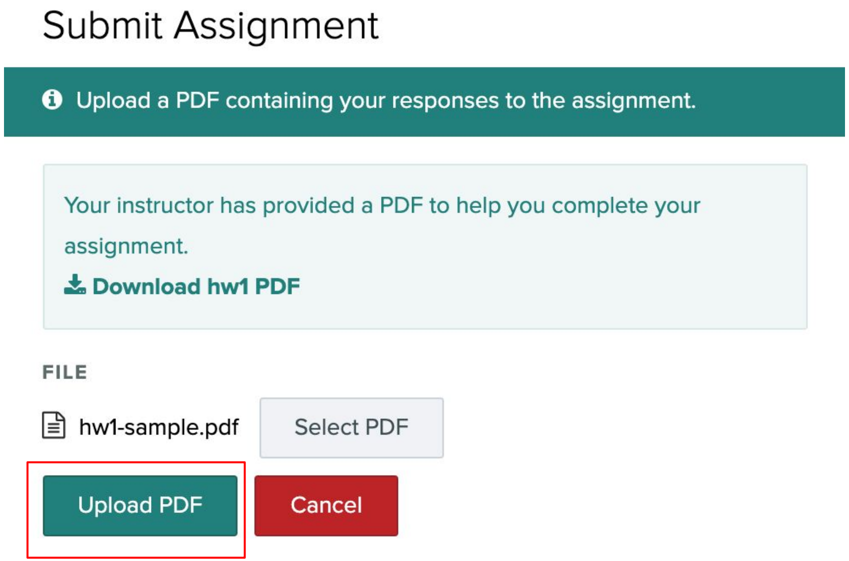
Your instructor has provided a PDF to help you complete your assignment.

Download hw1 PDF

FILE

Please select a file

How to submit your hw to Gradescope?



Submit Assignment

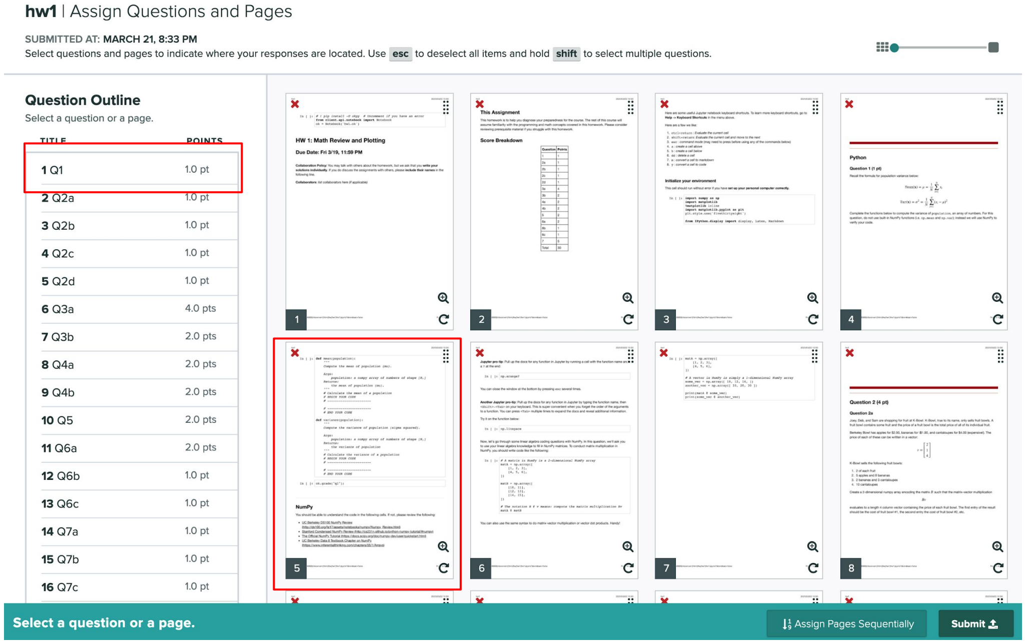
Upload a PDF containing your responses to the assignment.

Your instructor has provided a PDF to help you complete your assignment.

Download hw1 PDF

FILE

hw1-sample.pdf



hw1 | Assign Questions and Pages

SUBMITTED AT: MARCH 21, 8:33 PM

Select questions and pages to indicate where your responses are located. Use esc to deselect all items and hold shift to select multiple questions.

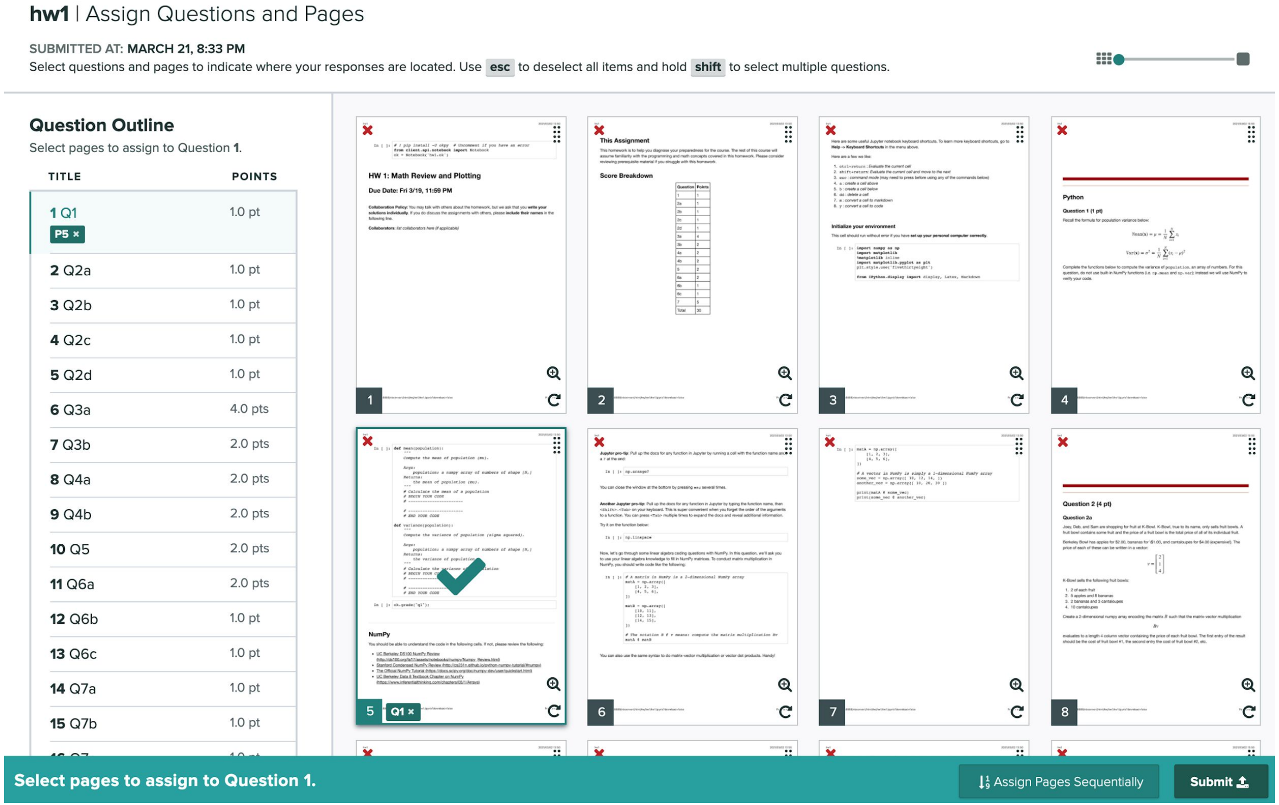
Question Outline

Select a question or a page.

TITLE	POINTS
1 Q1	1.0 pt
2 Q2a	1.0 pt
3 Q2b	1.0 pt
4 Q2c	1.0 pt
5 Q2d	1.0 pt
6 Q3a	4.0 pts
7 Q3b	2.0 pts
8 Q4a	2.0 pts
9 Q4b	2.0 pts
10 Q5	2.0 pts
11 Q6a	2.0 pts
12 Q6b	1.0 pt
13 Q6c	1.0 pt
14 Q7a	1.0 pt
15 Q7b	1.0 pt
16 Q7c	1.0 pt

Select a question or a page.

How to submit your hw to Gradescope?



hw1 | Assign Questions and Pages

SUBMITTED AT: **MARCH 21, 8:33 PM**

Select questions and pages to indicate where your responses are located. Use **esc** to deselect all items and hold **shift** to select multiple questions.

Question Outline

Select pages to assign to Question **1**.

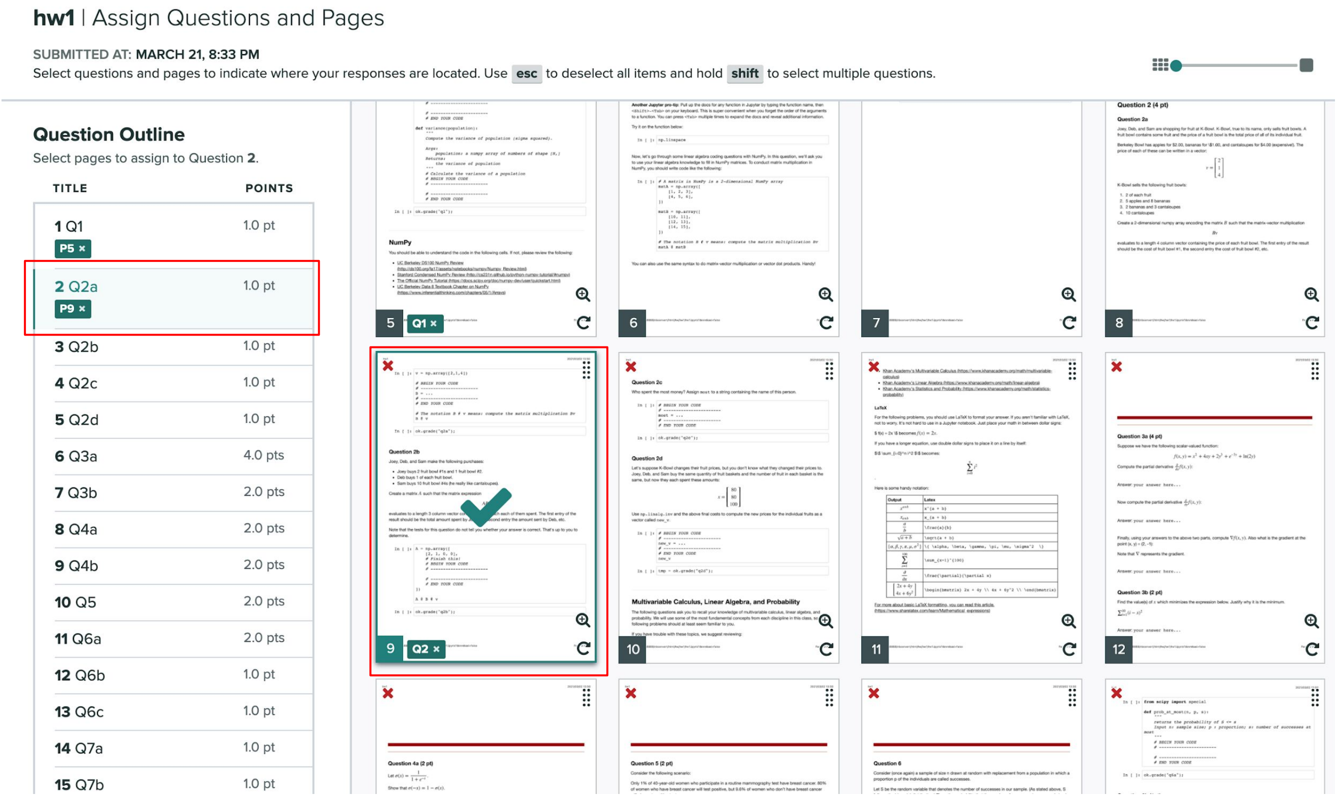
TITLE	POINTS
1 Q1 P5 ×	1.0 pt
2 Q2a	1.0 pt
3 Q2b	1.0 pt
4 Q2c	1.0 pt
5 Q2d	1.0 pt
6 Q3a	4.0 pts
7 Q3b	2.0 pts
8 Q4a	2.0 pts
9 Q4b	2.0 pts
10 Q5	2.0 pts
11 Q6a	2.0 pts
12 Q6b	1.0 pt
13 Q6c	1.0 pt
14 Q7a	1.0 pt
15 Q7b	1.0 pt

Select pages to assign to Question 1.

Assign Pages Sequentially

Submit

How to submit your hw to Gradescope?



hw1 | Assign Questions and Pages

SUBMITTED AT: **MARCH 21, 8:33 PM**

Select questions and pages to indicate where your responses are located. Use **esc** to deselect all items and hold **shift** to select multiple questions.

Question Outline

Select pages to assign to Question **2**.

TITLE	POINTS
1 Q1 P5 ×	1.0 pt
2 Q2a P9 ×	1.0 pt
3 Q2b	1.0 pt
4 Q2c	1.0 pt
5 Q2d	1.0 pt
6 Q3a	4.0 pts
7 Q3b	2.0 pts
8 Q4a	2.0 pts
9 Q4b	2.0 pts
10 Q5	2.0 pts
11 Q6a	2.0 pts
12 Q6b	1.0 pt
13 Q6c	1.0 pt
14 Q7a	1.0 pt
15 Q7b	1.0 pt

Another thing to mention…

Assignments

Data science is a collaborative activity! It is okay to discuss problems with friends.

- List their names at the top of your assignments. We provide a place to do this.
- You must write your solutions individually! Do not copy any other student’s work.
- If we suspect that you have submitted plagiarized work, we will call you in for a meeting. If we then determine that plagiarism has occurred, we reserve the right to give you a negative full score (-100%) or lower on the assignments in question, along with reporting your offense to the University.

Previously ...

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

Pandas and Jupyter Notebooks

- 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

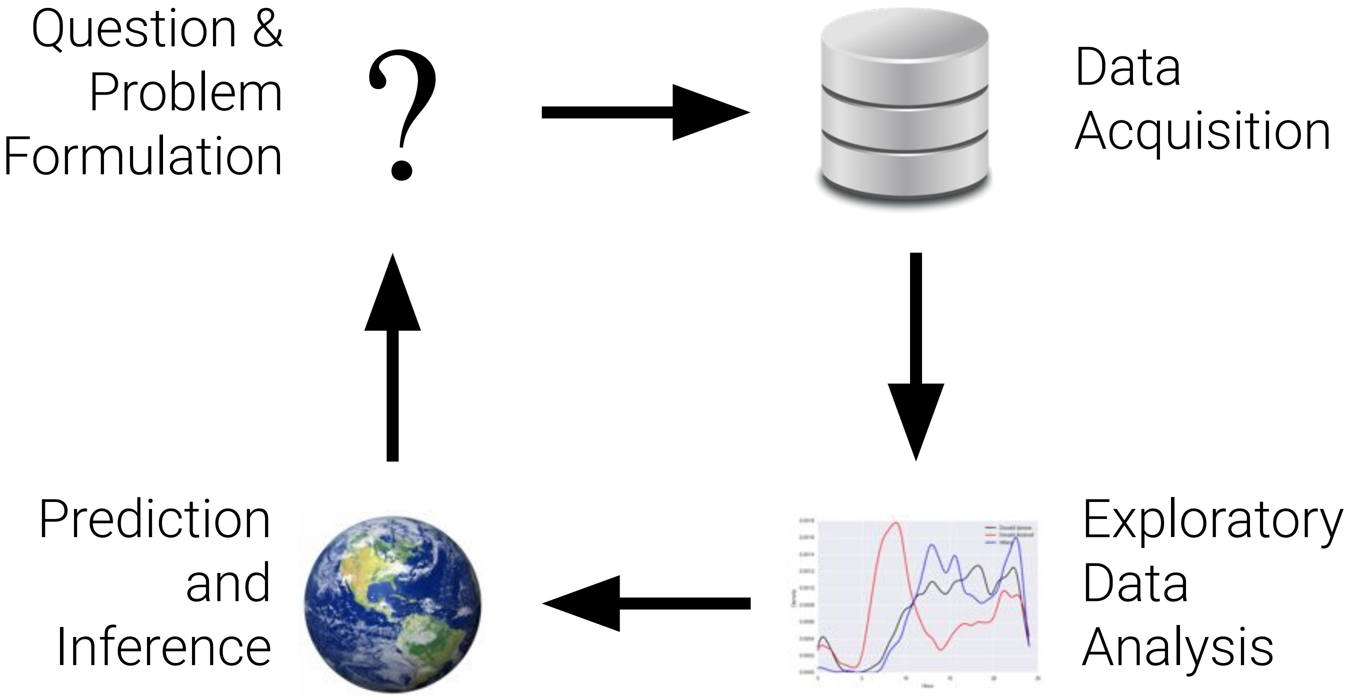
- Dataframe access methods
  - **Filtering** on predicts and **slicing**
  - **df.loc**: location by index
  - **df.iloc**: location by integer address
  - **groupby** data

Today

Congratulations

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

What do you do next?

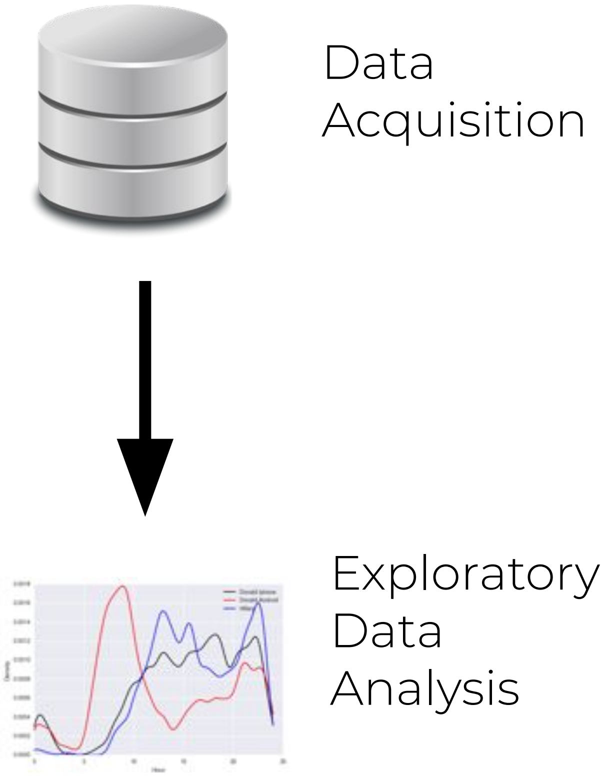


Question & Problem Formulation

Data Acquisition

Exploratory Data Analysis

Prediction and Inference

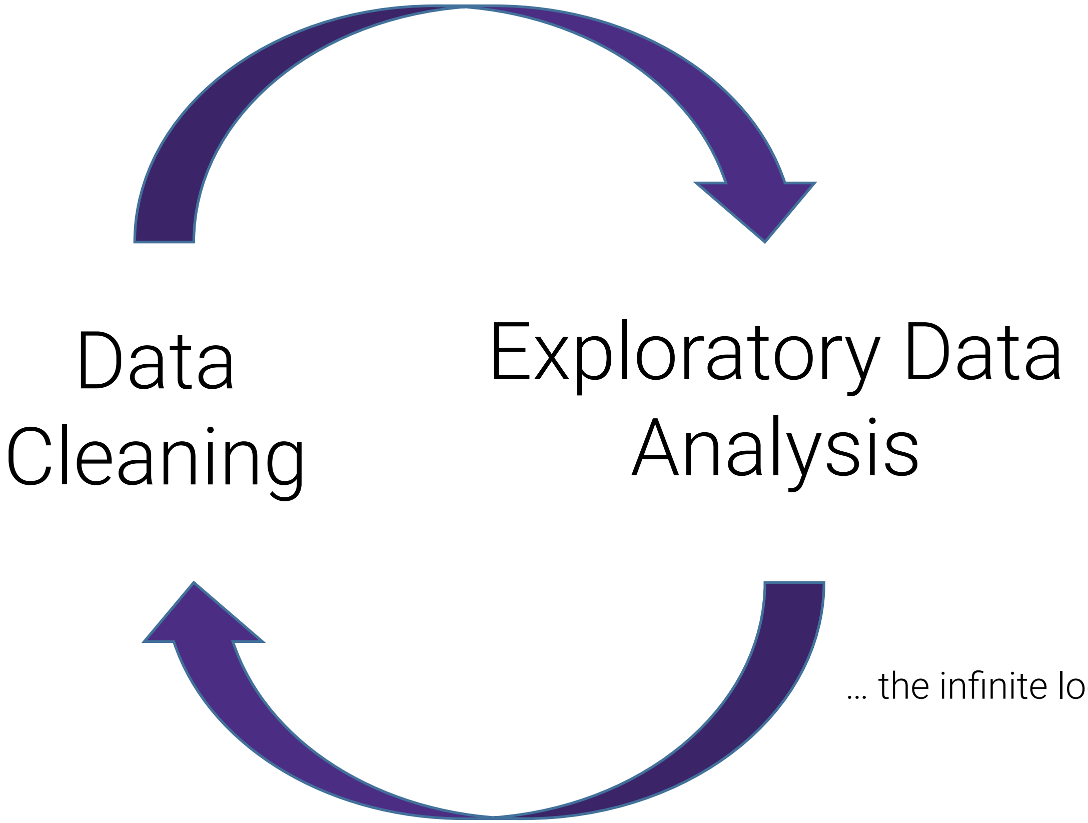


Data Acquisition

Exploratory Data Analysis

Topics For This Lecture

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



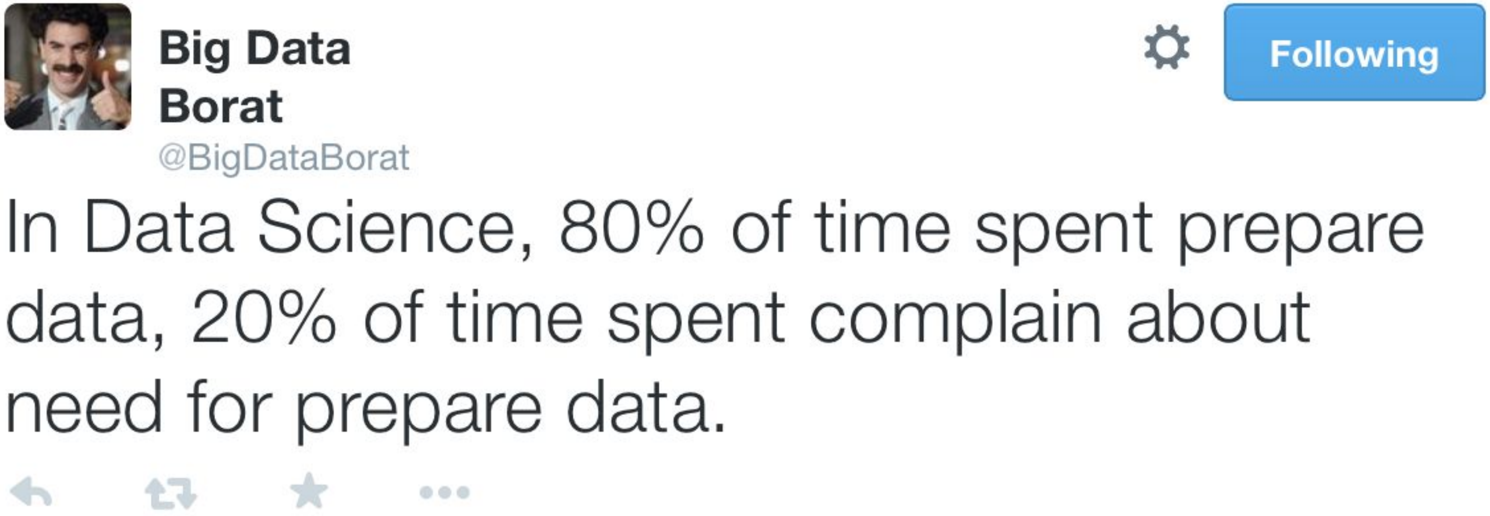
Data Cleaning

Exploratory Data Analysis

… the infinite loop of data science.

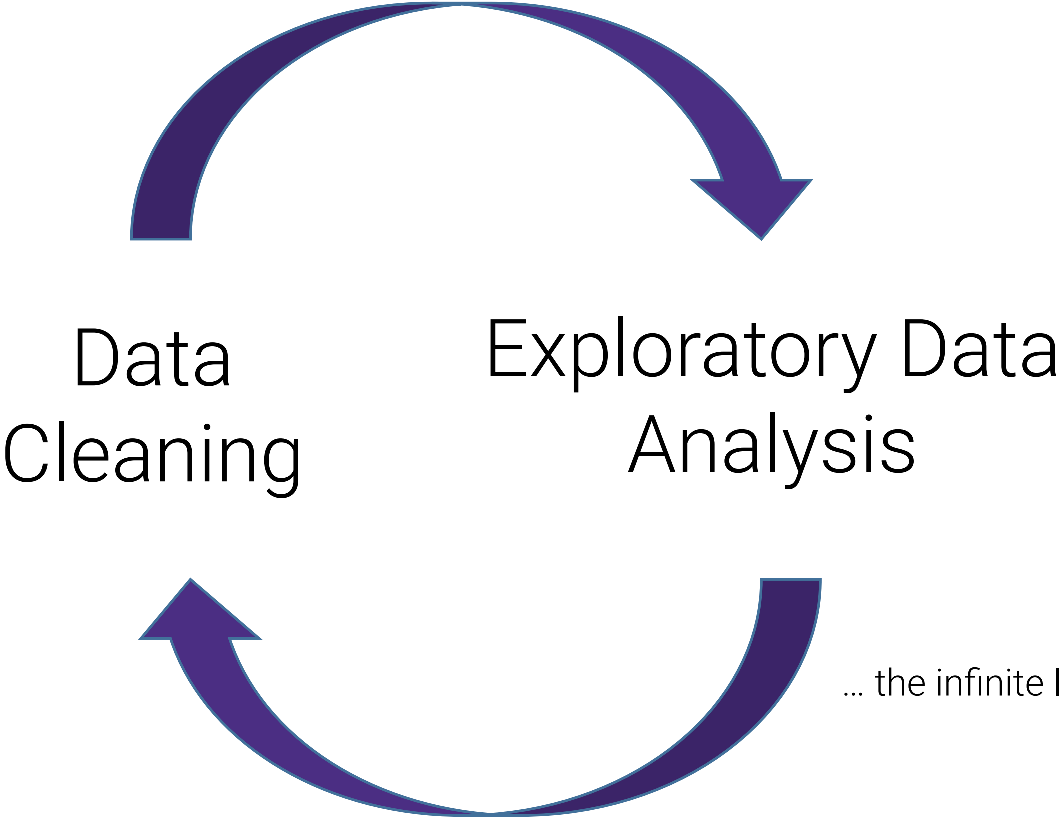
Data Cleaning

- The process of transforming **raw data** to facilitate subsequent analysis
- Data cleaning often addresses **issues**
  - 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

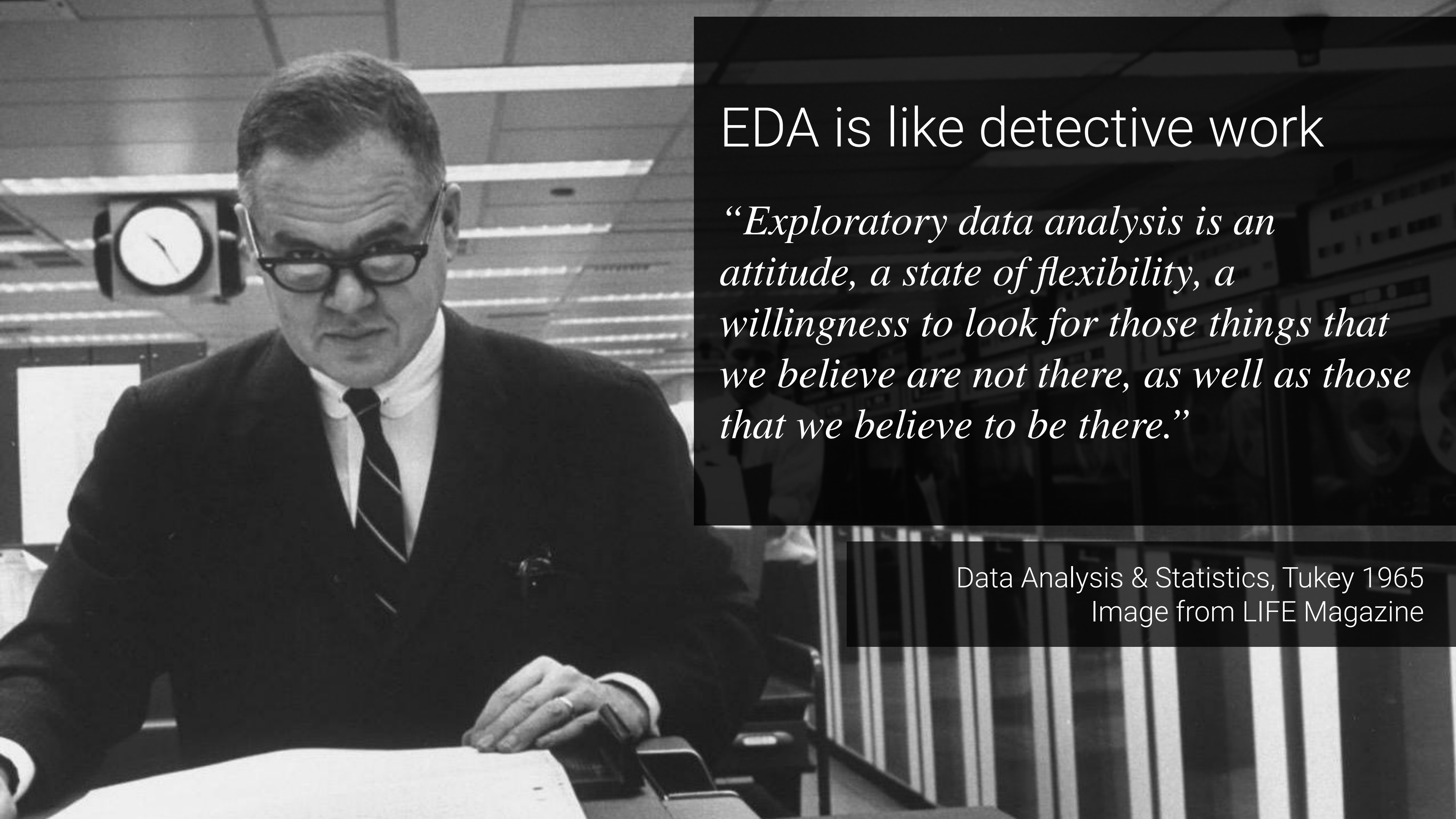
Exploratory Data Analysis

… the infinite loop of data science.

Exploratory Data Analysis (EDA)

“Getting to know the data”

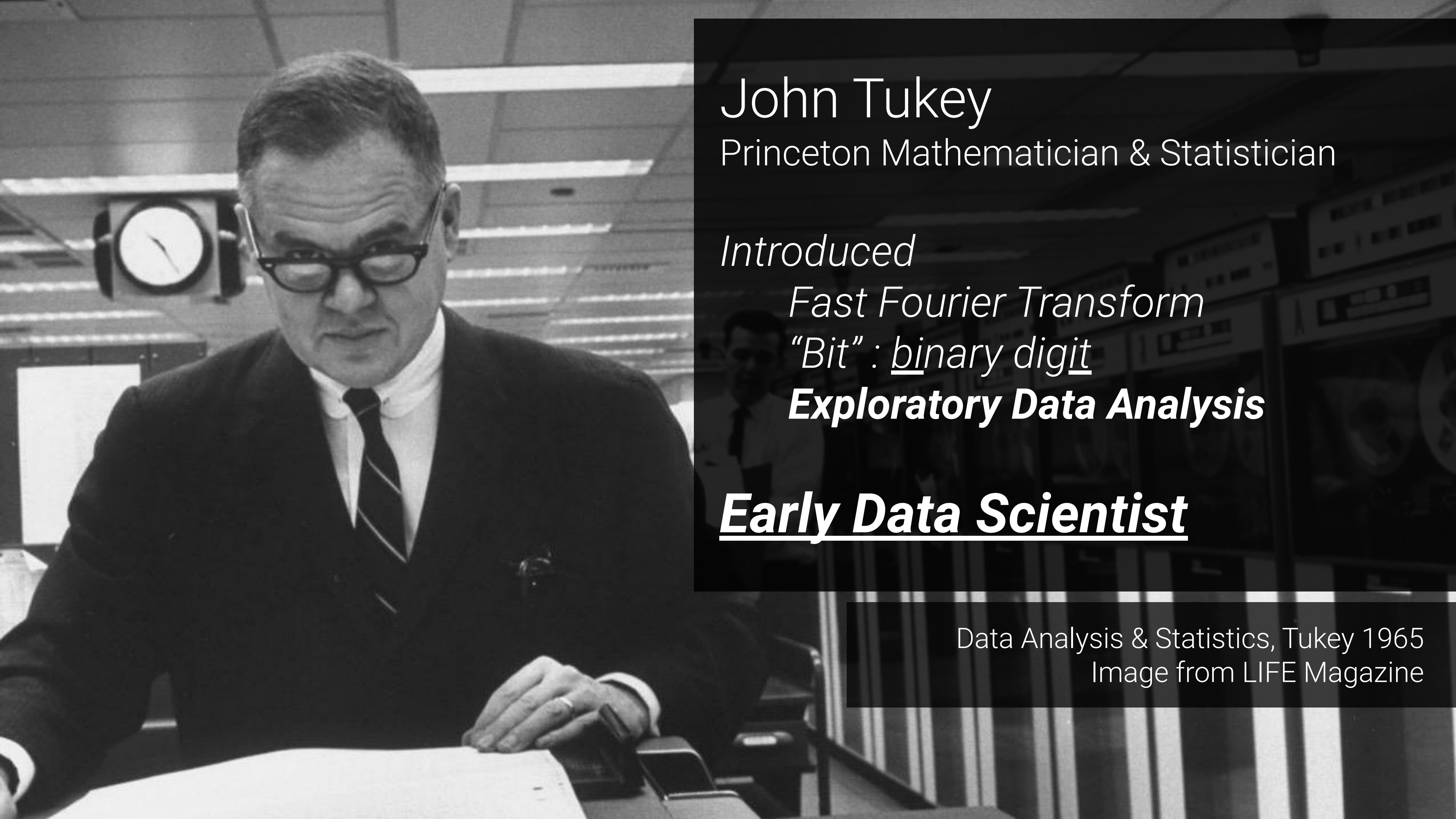
- The process of **transforming**, **visualizing**, and **summarizing** data to:
  - Build/confirm understanding of the data and its provenance
  - Identify and address potential issues in the data
  - Inform the subsequent analysis
  - discover *potential* hypothesis … (be careful)
- **EDA is an open-ended analysis**
  - Be willing to find something surprising



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



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

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”*

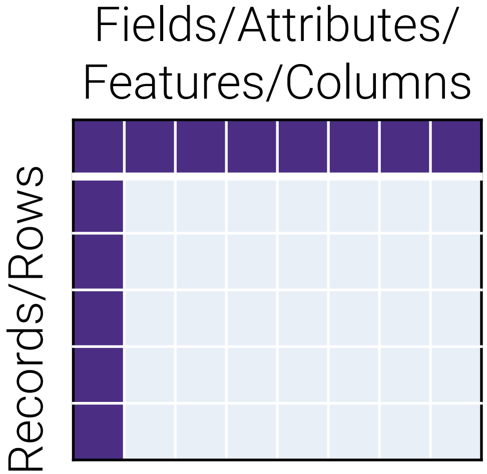
Rectangular Data

We prefer rectangular data for data analysis (why?)

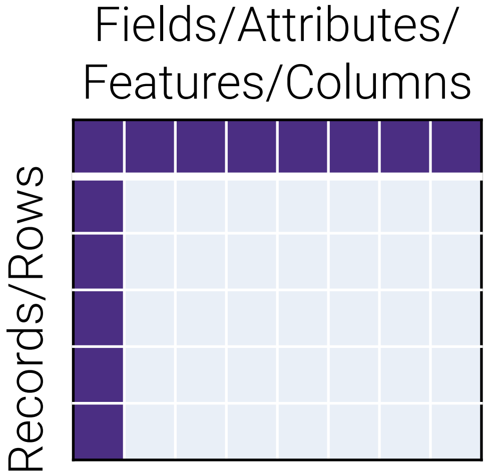
- Regular structures are easy manipulate and analyze
- A big part of data cleaning is about transforming data to be more rectangular

Two kinds of rectangular data: *Tables and Matrices*

(what are the differences?)



Rectangular Data



We prefer rectangular data for data analysis (why?)

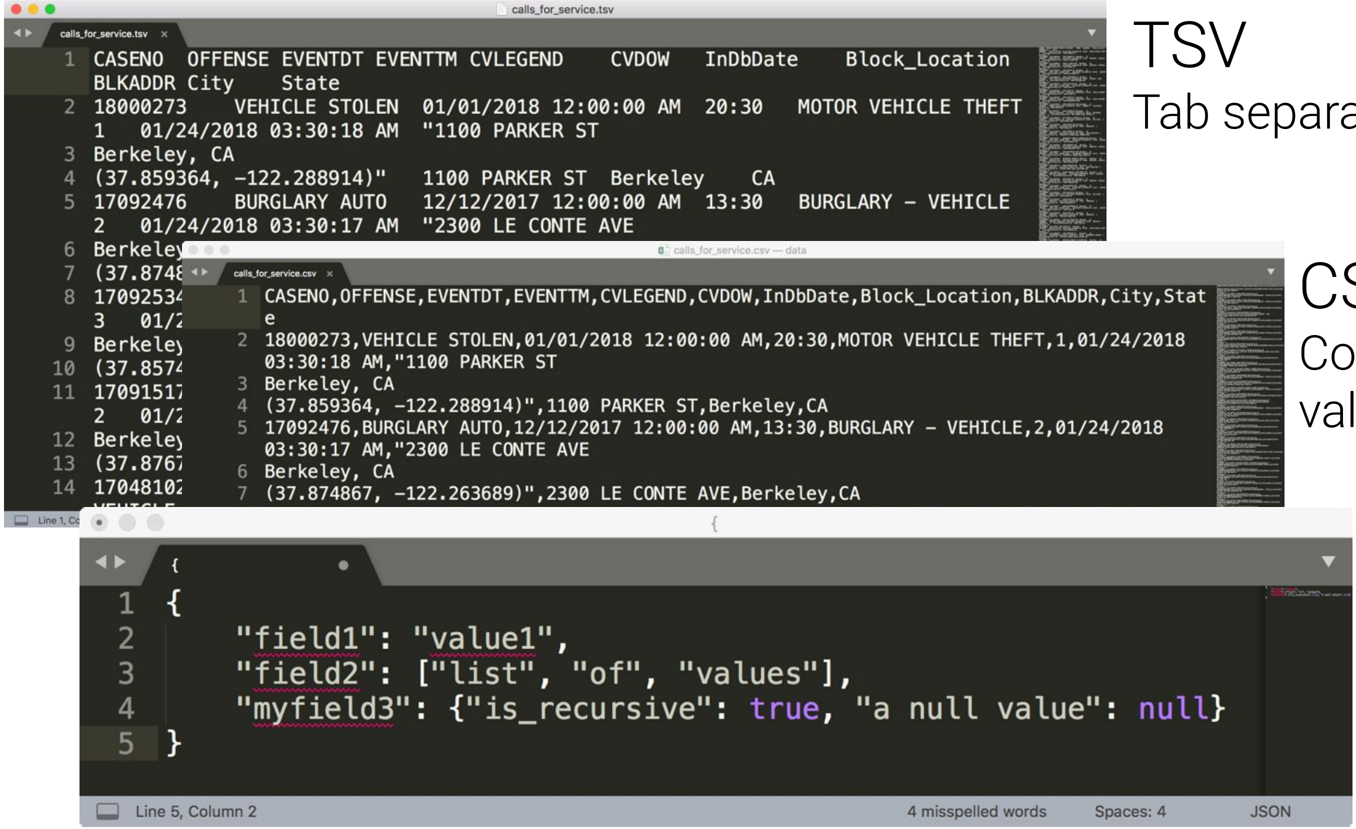
- Regular structures are easy manipulate and analyze
- A big part of data cleaning is about transforming data to be more rectangular

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
(37.8748
1709253
3    01/2
Berkeley
(37.8574
1709151
2    01/2
Berkeley
(37.8767
1704810
```

```
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
```

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

TSV

Tab separated values

CSV

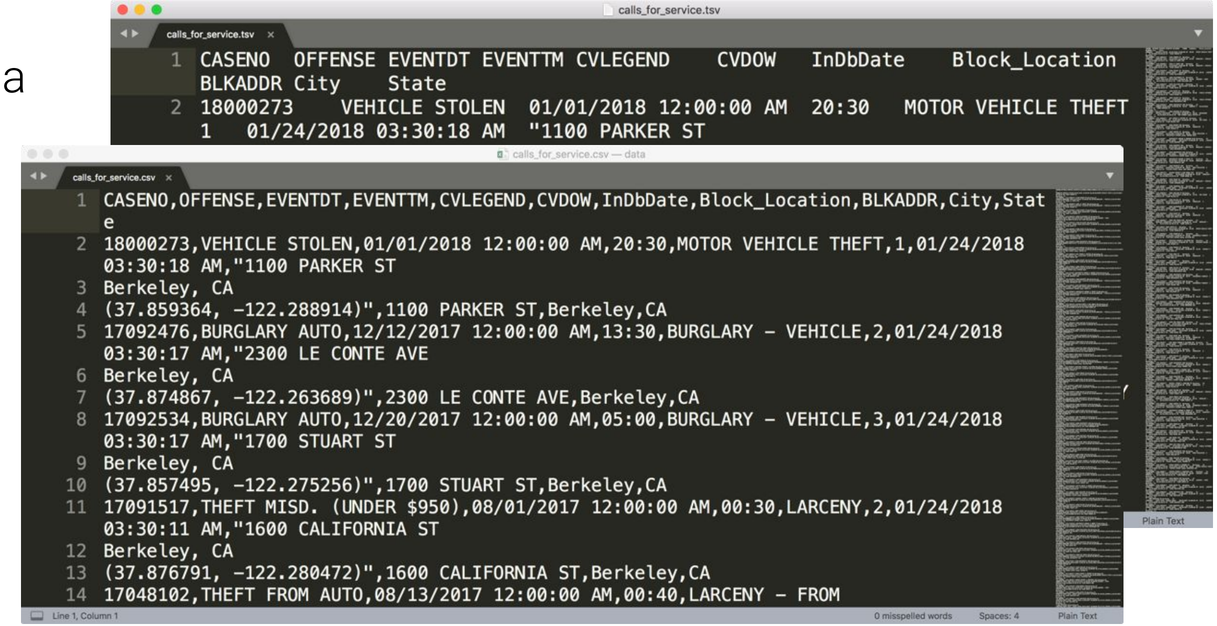
Comma separated values

JSON

Which is the best?

Comma and Tab Separated Values Files

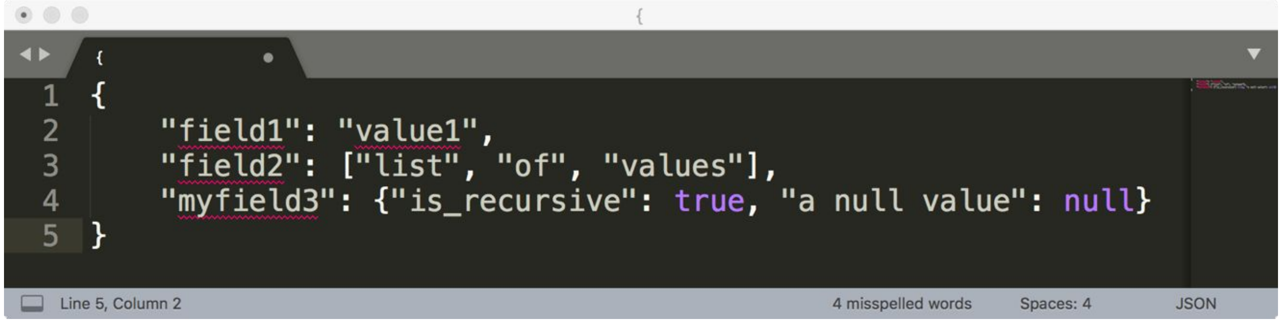
- Tabular data where
  - Records are delimited by a *newline*: “\n”, “\r\n”
  - Fields are delimited by ‘,’ (comma) or ‘\t’ (tab)
- Very Common!
- Issues?
  - Commas, tabs in records
  - Quoting
  - ...



```
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,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,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)



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

- Widely used file format for nested data
  - Very similar to python dictionaries
  - Strict formatting ”quoting” addresses some issues in CSV/TSV
- Issues
  - Not rectangular
  - Each record can have different fields
  - Nesting means records can contain tables – complicated

Extensible Markup Language - 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

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)"
```

Keys and Joins

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, …

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

- 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 will need to **join** across tables

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

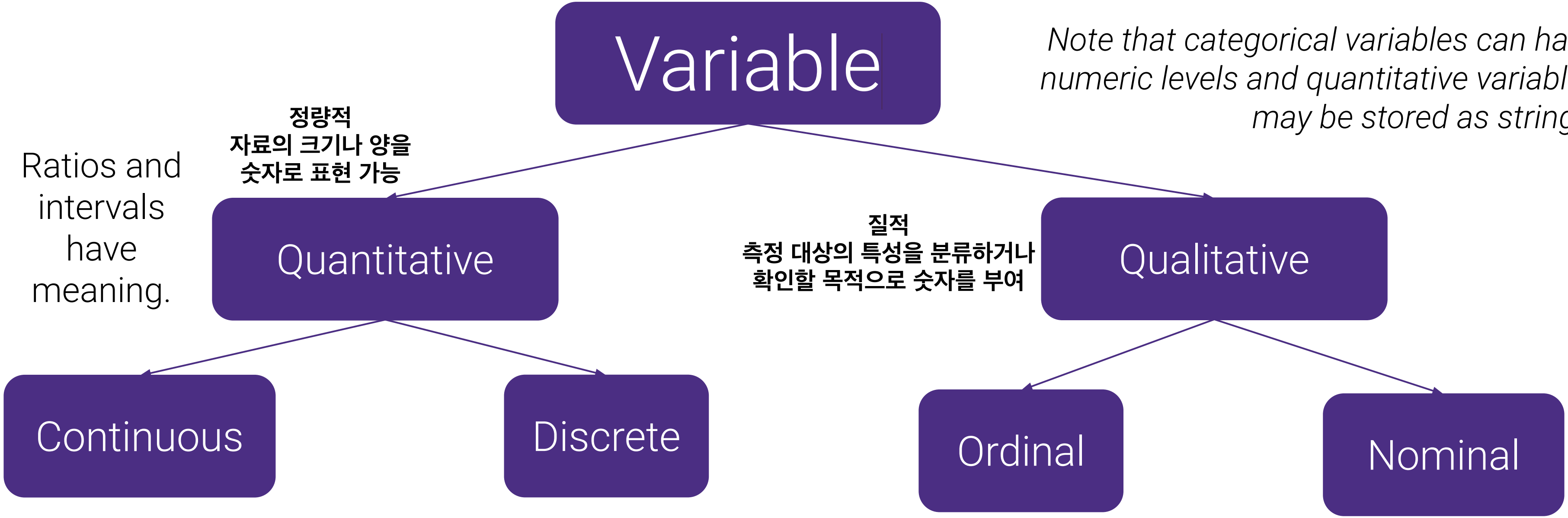
Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

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?

Variable Types



Variable

Note that categorical variables can have numeric levels and quantitative variables may be stored as strings.

정량적
자료의 크기나 양을
숫자로 표현 가능

Ratios and intervals have meaning.

Quantitative

질적
측정 대상의 특성을 분류하거나
확인할 목적으로 숫자를 부여

Qualitative

Continuous

Could be measured to arbitrary precision.

Examples:

- Price
- Temperature

Discrete

Finite possible values

Examples:

- Number of siblings
- Yrs of education

Ordinal

Categories w/ levels but no consistent meaning to difference

Examples:

- Preferences
- Level of education

Nominal

Categories w/ no specific ordering.

Examples:

- Political Affiliation
- ID number

What is the type of variable?

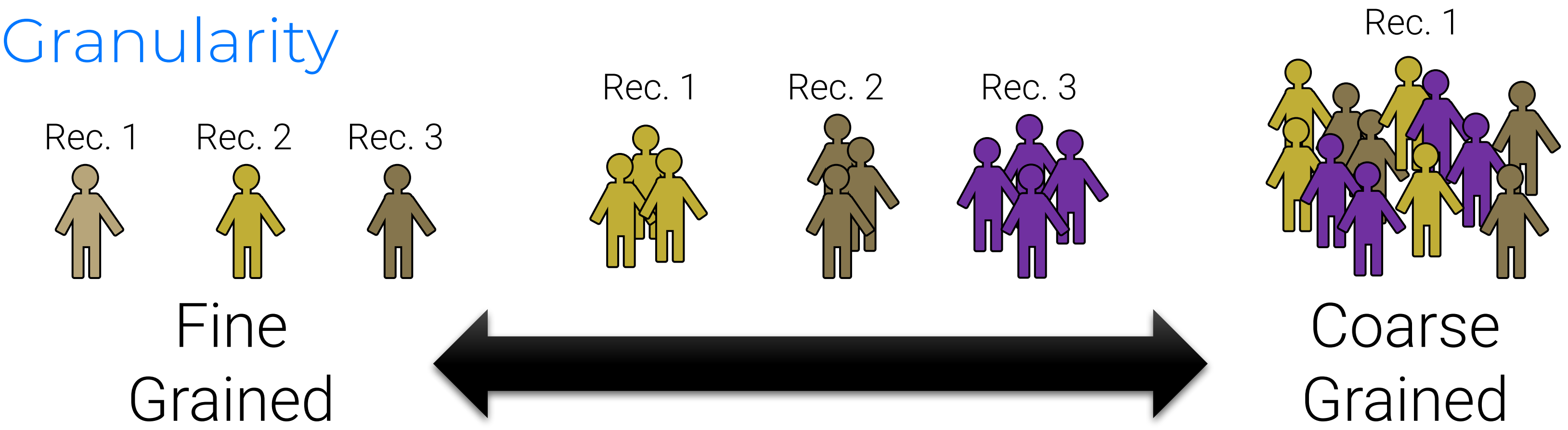
	Quantitative Continuous	Quantitative Discrete	Qualitative Ordinal	Qualitative Nominal
CO_2 level (PPM)	○			
Number of siblings		○		
GPA	○			
Income bracket (low, med, high)			○	
Race				○
Number of years of education		○		
Yelp Rating			○	

Granularity, Scope, and Temporality

Key Data Properties to Consider in EDA

- **Structure --** *the "shape" of a data file*
- **Granularity --** *how fine/coarse is each datum* **단위, 입상**
- **Scope --** *how (in)complete is the data*
- **Temporality --** *how is the data situated in time*
- **Faithfulness --** *how well does the data capture "reality"*

Granularity



Fine Grained

Coarse Grained

- 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 was it aggregated?
  - Sampling, averaging, …

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 Korea but I only have Seoul crime data.*

- Is my data too big?
  - **Example:** *I am interested in student grades for COSE471 but have student grades for all CS 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 …

Revisiting the Sampling Frame

- The **sampling frame** is the **population** from which the data was **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?

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"*

Temporality

- Data changes – when was the data collected?

- What is the meaning of the time and date fields?
  - 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
  - Multiple string representation (depends on region): 07/08/09?

- Are there strange null values?
  - January 1st 1970, January 1st 1900

- Is there periodicity? Diurnal patterns

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"*

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 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

- Missing Values/Default values?
  - What do they look like?
    - "   ",
    - 0,
    - -1, 999, 12345,
    - NaN, Null,
    - 1970, 1900

What to do with the Missing Values?

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

- **Imputation:** (Inferring missing values)
  - **Mean Imputation:** replace with an average value
    - Which mean? 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.

- **Suggestion:**
  - Drop missing values **but check for induced bias (use domain knowledge)**
  - Directly **model missing values** during future analysis

Signs that your data may not be faithful

- **Missing** Values or **default** values
- Truncated data (early 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
- Others…

Summary

Summary

- 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
  - ***Record everything you do! (why?)***

(Optional) Case Study:

CO2 levels at Mauna Loa Observatory

Mauna Loa Volcano

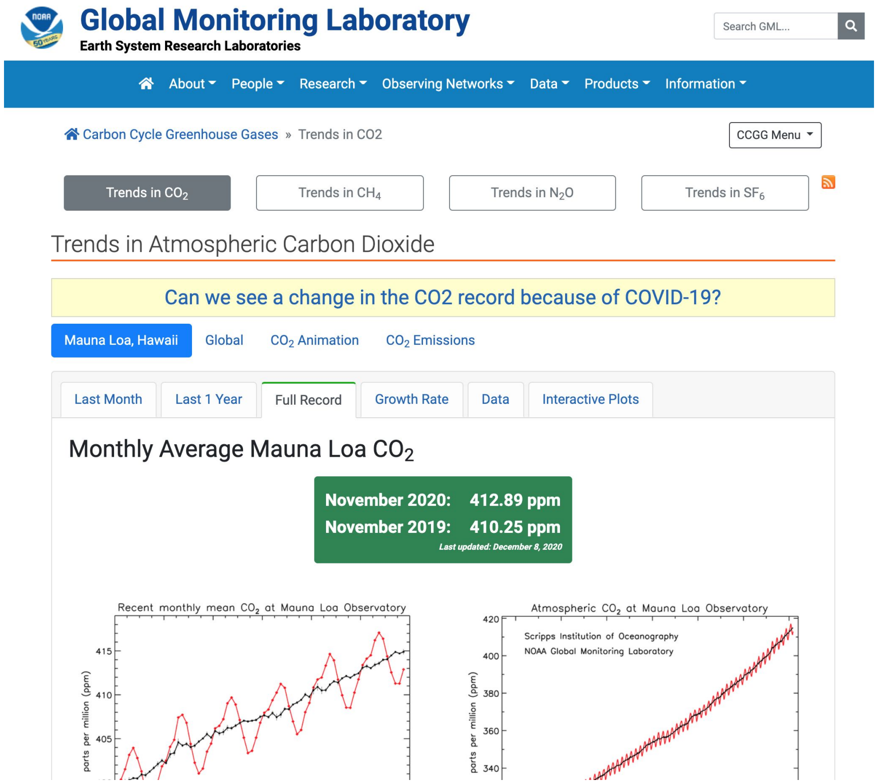
- Largest Volcano in world on the island of Hawaii

Mauna Loa Observatory

- Far from any continent: air sampled is representative of the central Pacific.
- High altitude: above the inversion layer where local effects may be present
- Measurements of atmospheric CO2 since 1958 - longest continuous record

Acquiring the box of data

- Clean
- Well documented
- Simple structure
- Broadly shared
- Reproducibility is key to trusting findings



Trends in Atmospheric Carbon Dioxide

Can we see a change in the CO2 record because of COVID-19?

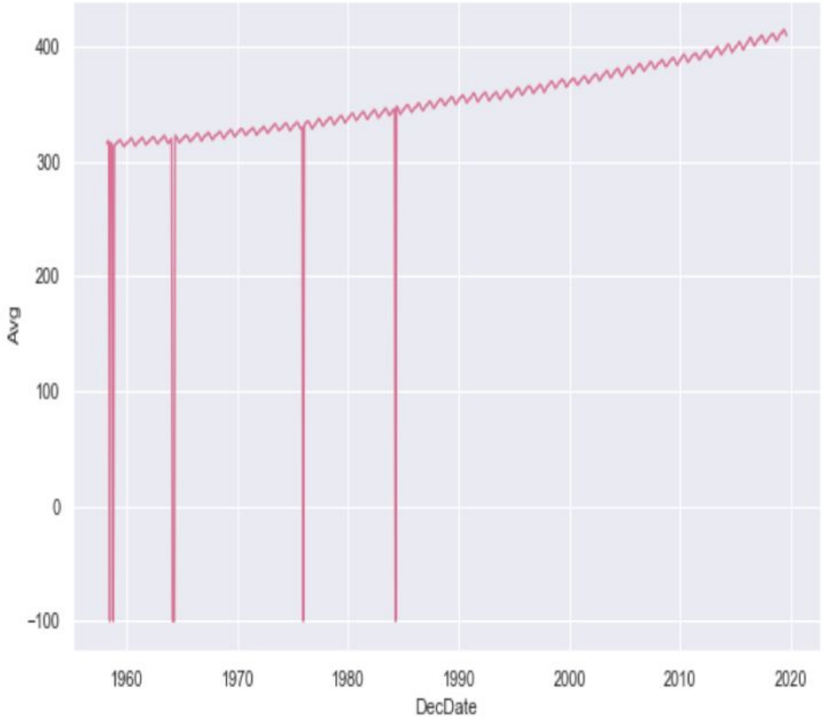
Monthly Average Mauna Loa CO_2

November 2020: 412.89 ppm
November 2019: 410.25 ppm
Last updated: December 8, 2020

Recent monthly mean CO_2 at Mauna Loa Observatory
parts per million (ppm)
415
410
405
Atmospheric CO_2 at Mauna Loa Observatory
Scripps Institution of Oceanography
NOAA Global Monitoring Laboratory
parts per million (ppm)
420
400
380
360
340

Modeling the change in CO2 over time

- YIKES!
- What happened?
- We didn’t clean our data

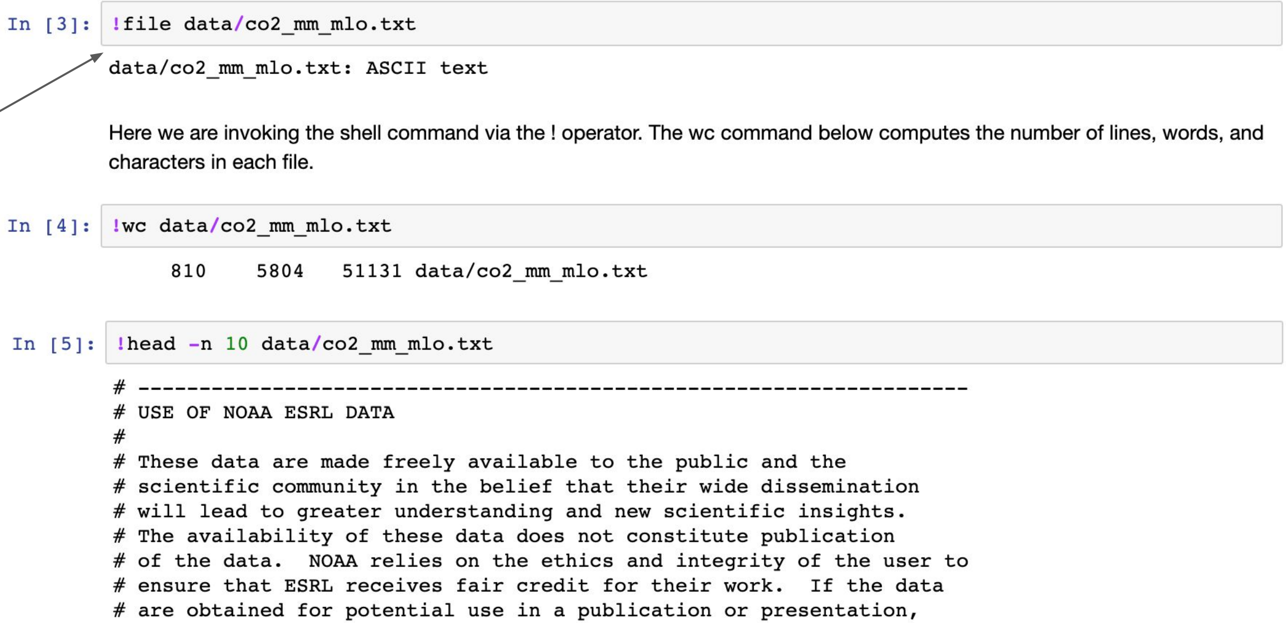


Here’s the Data: co2_mm_mlo.txt

- Start over with more care
- Now what?
    - How big is it?
    - What is the encoding?
    - How is it formatted?

Look at it

These are Unix commands that we run from the Jupyter notebook



```
In [3]: !file data/co2_mm_mlo.txt

data/co2_mm_mlo.txt: ASCII text
```

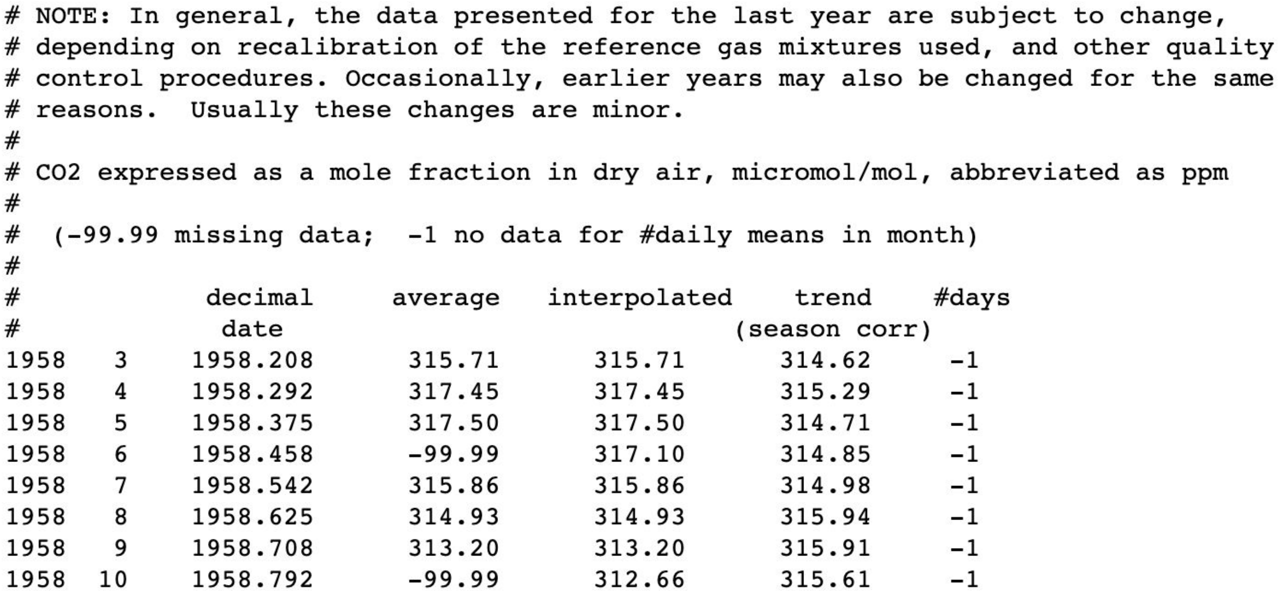
Here we are invoking the shell command via the ! operator. The wc command below computes the number of lines, words, and characters in each file.

```
In [4]: !wc data/co2_mm_mlo.txt

     810    5804   51131 data/co2_mm_mlo.txt
```

```
In [5]: !head -n 10 data/co2_mm_mlo.txt

# --------------------------------------------------------------------
# USE OF NOAA ESRL DATA
#
# These data are made freely available to the public and the
# scientific community in the belief that their wide dissemination
# will lead to greater understanding and new scientific insights.
# The availability of these data does not constitute publication
# of the data.  NOAA relies on the ethics and integrity of the user to
# ensure that ESRL receives fair credit for their work.  If the data
# are obtained for potential use in a publication or presentation,
```

Look at it

What do you see?
Make 4 observations
about these data

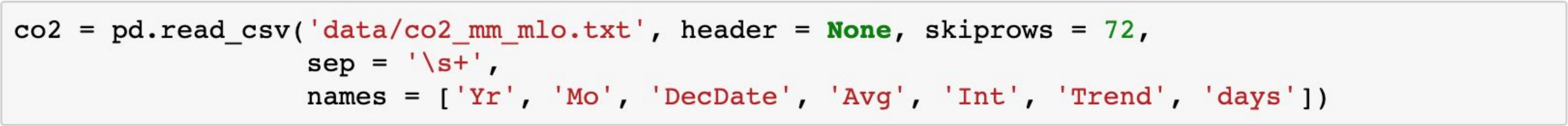


```
# NOTE: In general, the data presented for the last year are subject to change,
# depending on recalibration of the reference gas mixtures used, and other quality
# control procedures. Occasionally, earlier years may also be changed for the same
# reasons.  Usually these changes are minor.
#
# CO2 expressed as a mole fraction in dry air, micromol/mol, abbreviated as ppm
#
#  (-99.99 missing data;  -1 no data for #daily means in month)
#
#            decimal     average   interpolated    trend    #days
#             date                             (season corr)
1958   3    1958.208      315.71      315.71      314.62     -1
1958   4    1958.292      317.45      317.45      315.29     -1
1958   5    1958.375      317.50      317.50      314.71     -1
1958   6    1958.458      -99.99      317.10      314.85     -1
1958   7    1958.542      315.86      315.86      314.98     -1
1958   8    1958.625      314.93      314.93      315.94     -1
1958   9    1958.708      313.20      313.20      315.91     -1
1958  10    1958.792      -99.99      312.66      315.61     -1
```

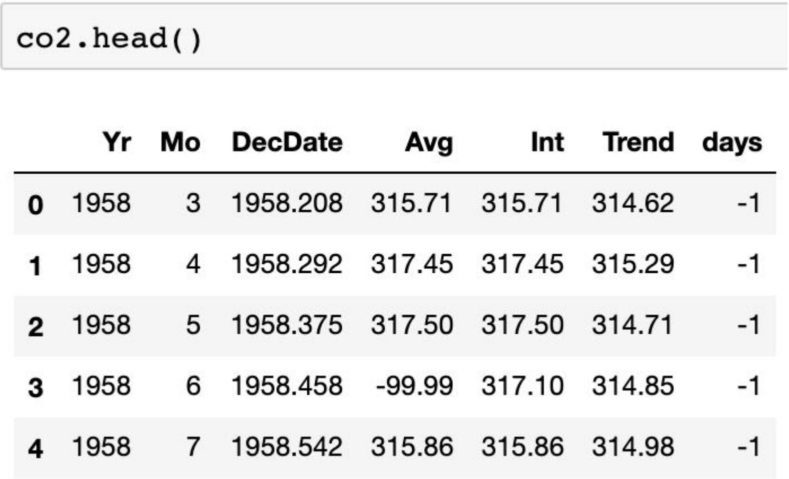
Observations about the file

- File appears to be plain text
- Column names appear on two lines of file
- Fields line up from one row to the next
- White space between fields
- Seven variables
- -99.99 appears in some rows for the “Average”
- -1 appears in all the first 5 rows for “days”

Read the Data into a Data Frame

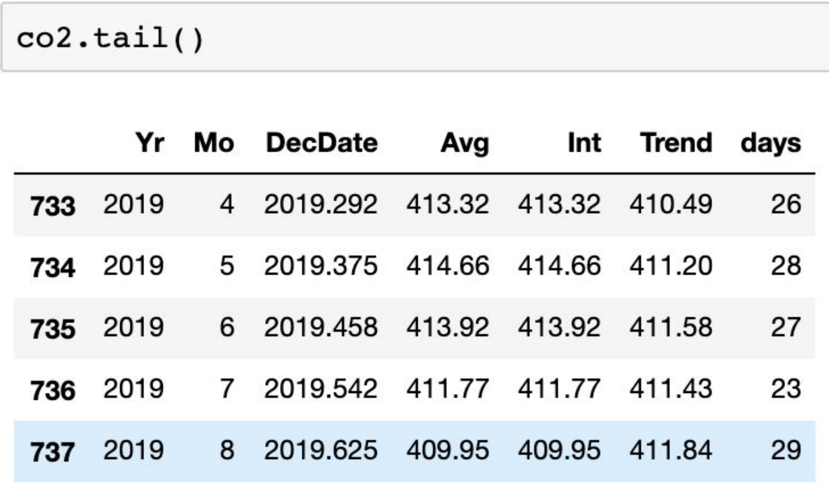


```python
co2 = pd.read_csv('data/co2_mm_mlo.txt', header = None, skiprows = 72,
                  sep = '\s+',
                  names = ['Yr', 'Mo', 'DecDate', 'Avg', 'Int', 'Trend', 'days'])
```



```python
co2.head()
```

	Yr	Mo	DecDate	Avg	Int	Trend	days
0	1958	3	1958.208	315.71	315.71	314.62	-1
1	1958	4	1958.292	317.45	317.45	315.29	-1
2	1958	5	1958.375	317.50	317.50	314.71	-1
3	1958	6	1958.458	-99.99	317.10	314.85	-1
4	1958	7	1958.542	315.86	315.86	314.98	-1



```python
co2.tail()
```

	Yr	Mo	DecDate	Avg	Int	Trend	days
733	2019	4	2019.292	413.32	413.32	410.49	26
734	2019	5	2019.375	414.66	414.66	411.20	28
735	2019	6	2019.458	413.92	413.92	411.58	27
736	2019	7	2019.542	411.77	411.77	411.43	23
737	2019	8	2019.625	409.95	409.95	411.84	29

Identify the Structure and Granularity

- What is the shape?
- What does a record represent?
- Have the data aggregated?
- Do we need to aggregate?

Identify the Structure and Granularity

- What is the shape?
  - Rectangular - 7 columns & 738 rows
- What does a record represent?
  - One month of CO2 measurements
- Have the data aggregated?
  - Yes, they are aggregated to the month, via an average.
- Do we need to aggregate?
  - We don’t need to further aggregate.

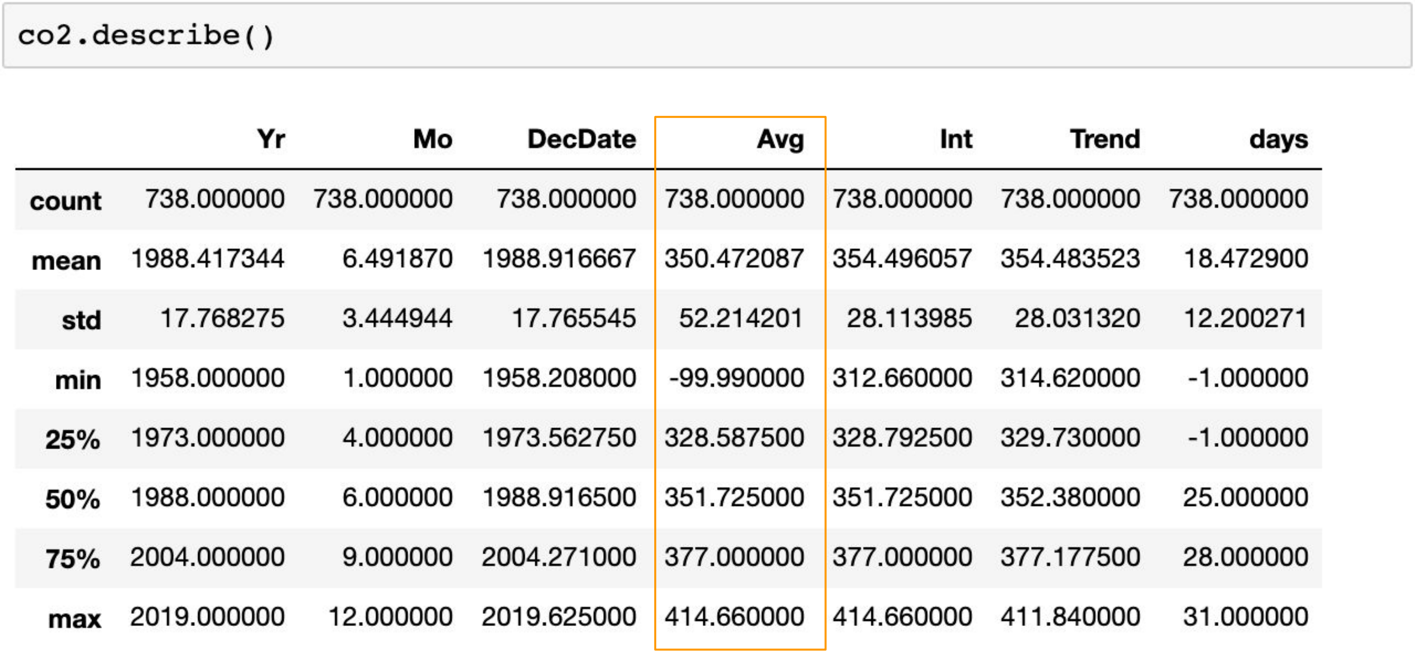
Ideas for confirming data quality?

- Can you think of some ways for us to check that the data are what we expect?
- What about ways to check consistency between the variables?
  - Yr -- 4 digits, from 1958 to 2019
  - Mo -- 1 to 12
  - DecDate -- Jan 1, 1958 = 1958 + 1/365
  - Avg -- Average monthly CO2
  - Int -- Interpolated CO2, if Avg is missing
  - Trend -- fitted trend
  - Days -- days in operation

Ideas for confirming data quality?

- How many records should we have?
    - 12 x (2019 - 1957) - 4 - 2 = 738
- How many records for a month should we have?
    - (2019 - 1957) = 62 or for some 61
- Are there any/many unusual values?
    - We saw -99.99 for Avg and -1 for days but we don’t know how many there are. Do we care about the -1s?

Checking data quality



```
co2.describe()
```

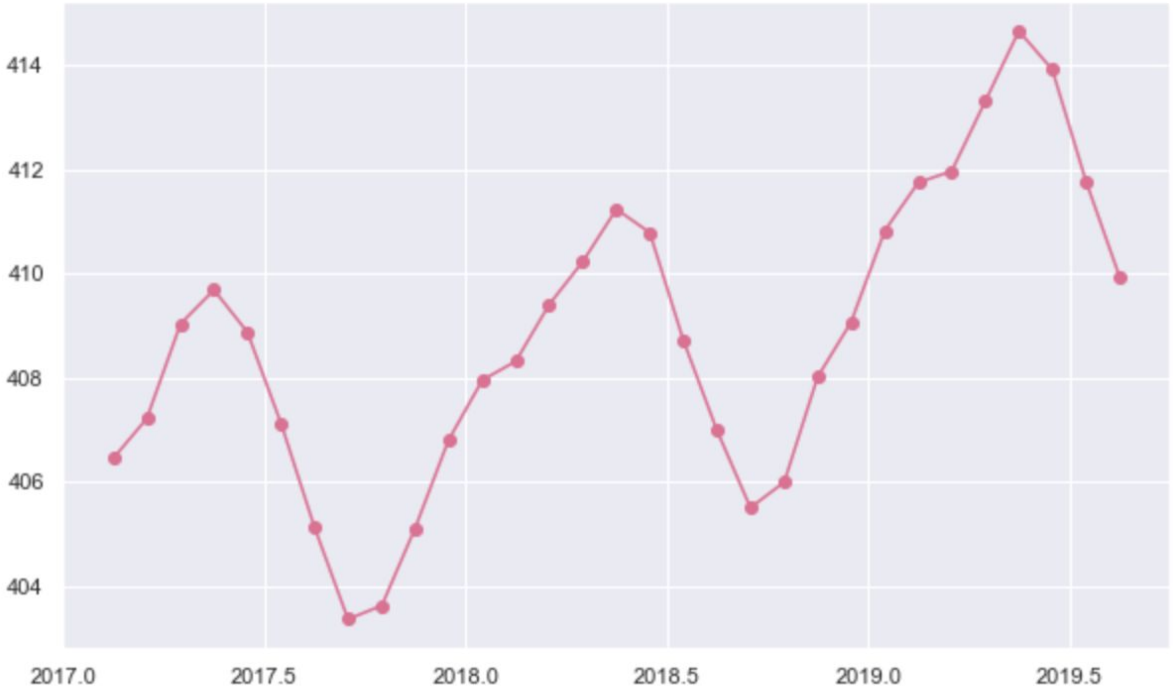
	Yr	Mo	DecDate	Avg	Int	Trend	days
count	738.000000	738.000000	738.000000	738.000000	738.000000	738.000000	738.000000
mean	1988.417344	6.491870	1988.916667	350.472087	354.496057	354.483523	18.472900
std	17.768275	3.444944	17.765545	52.214201	28.113985	28.031320	12.200271
min	1958.000000	1.000000	1958.208000	-99.990000	312.660000	314.620000	-1.000000
25%	1973.000000	4.000000	1973.562750	328.587500	328.792500	329.730000	-1.000000
50%	1988.000000	6.000000	1988.916500	351.725000	351.725000	352.380000	25.000000
75%	2004.000000	9.000000	2004.271000	377.000000	377.000000	377.177500	28.000000
max	2019.000000	12.000000	2019.625000	414.660000	414.660000	411.840000	31.000000

Zoom in a short time period

We see a seasonal component to CO2 measurements

Peak around April/May

Trough around Sep



June 2017 is Missing, What to do?

- Ignore it, and hope it goes away
- Drop the records with missing values
- Replace with the average from the previous 6 months
- Replace with a random June from the previous 6 years

What to do with the Missing Values?

- Drop the records with missing values
  - We typically selectively drop records for one analysis but not drop them for all analyses. For example, if a variable is not a model then we don’t drop records with missing values for that variable.
- Replace with the average from the previous 6 months
  - Typically, we divide the data into subgroups that have the same values for certain variables (e.g. age). Then we impute the missing value with the average value for the group.
- Replace with a random value, hot deck imputation
  - Like mean imputation, we divide the data into subgroups. But, we choose a random value from the subgroup and use it for the missing value.

What happens?

- Ideally, the missingness is at random -- meaning it is not correlated with other variables
- If missingness is correlated, that leads to biased inference
- If too many values for a field are missing, we may need to drop that field from our investigation
- If we impute with averages, the variability is reduced