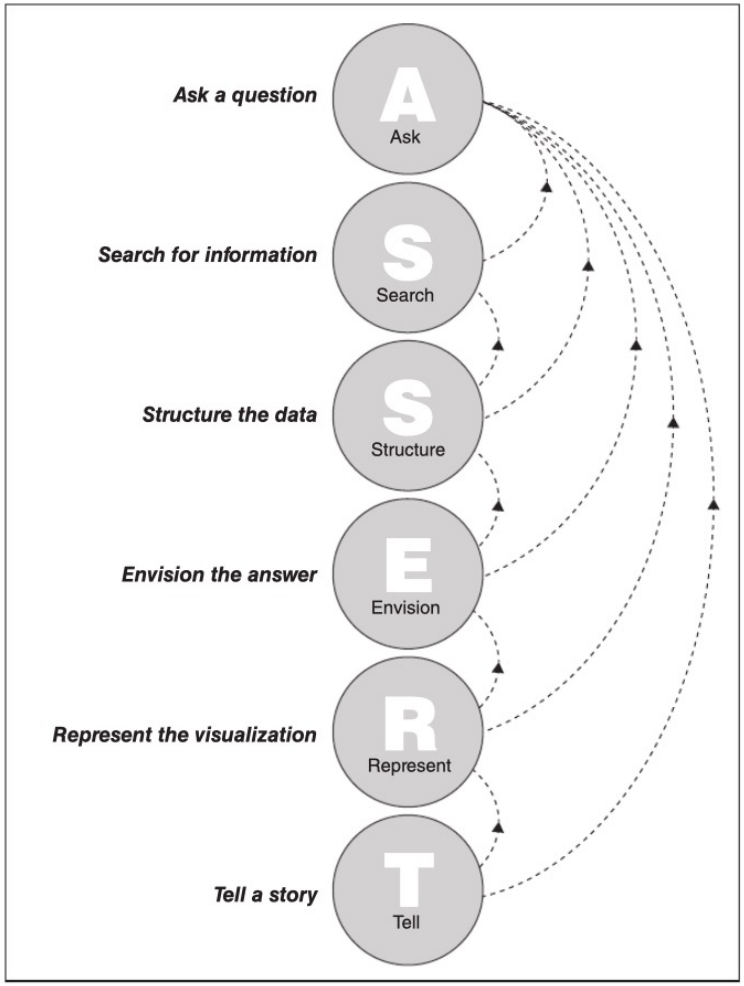
CP321 Data Visualization

- Structure the Data

Jiashu (Jessie) Zhao

Structure the data

- from raw data
  to data for visualization



- Information in its raw form needs to be abstracted and structured in order to tell a compelling story.
- Determine which and in what manner data are to be used to answer the question
- Make the process transparent to the viewer

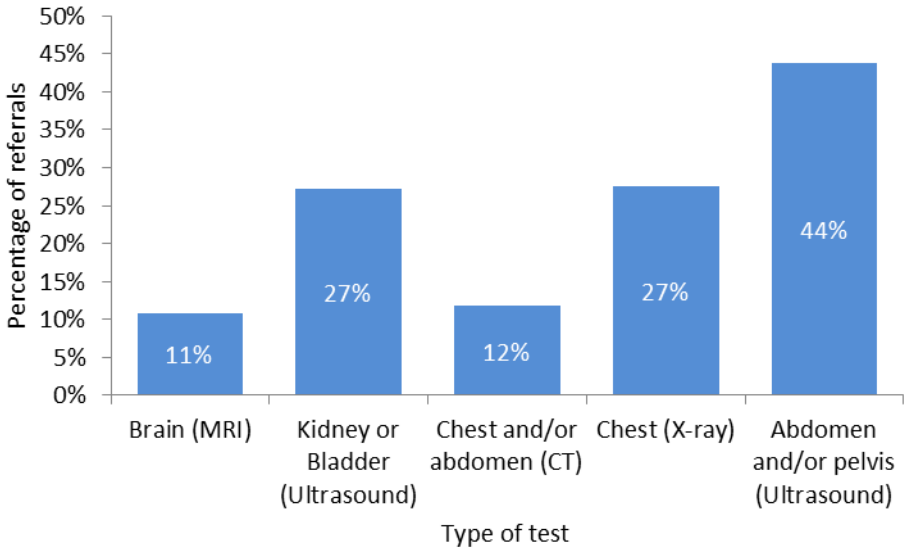
- Unstructured information is data that are lumped without organization into a single document
- Structured information is data that are organized, which means that data are sorted and grouped together into meaningful groups

- Unstructured text to structured data

Bob is a male student aged 22. He scored 100 on the test. Ted is a 43-year-old male student who scored 40 on the test. Carol is a female student age 33 and scored 90 on the test. Finally, Alice is a female student, age 23. She scored 75 on the test.

name	sex	age	grade
Bob	male	22	100
Ted	male	43	40
Carol	female	33	90
Alice	female	23	75

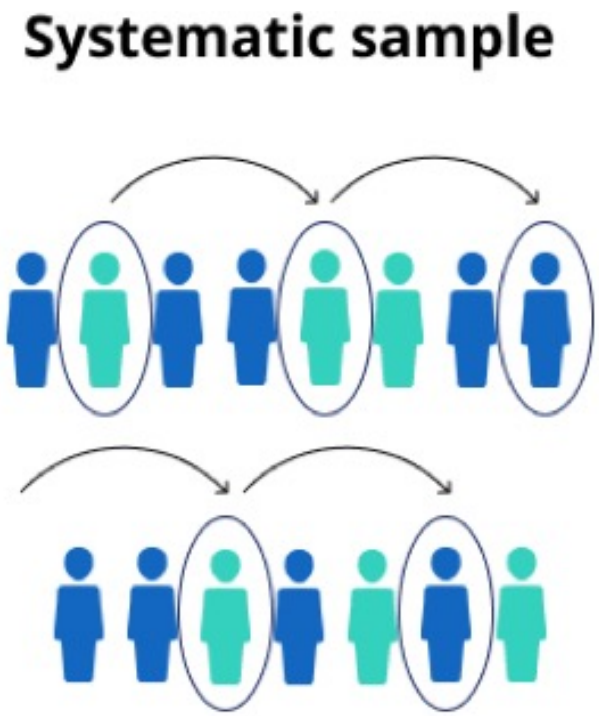
- Domain Space Abstraction
  - The data need to be abstracted from the domain space to quantitative and qualitative information for visualization
  - Information that serves as an effective source for information visualizations often is multidimensional.
  - By comparing subsets of these attributes, insights can be gleaned from exploring their interaction.



- Data Selection: Many information sets are large and cover data that are not useful for the question to be answered, so a subset of the data needs to be extracted from the full set
  - Defined by *minimum and maximum* values of a particular attribute
  - *Sampled*
  - *Aggregated*
  - *Clustered* by grouping attributes together by similar characteristics

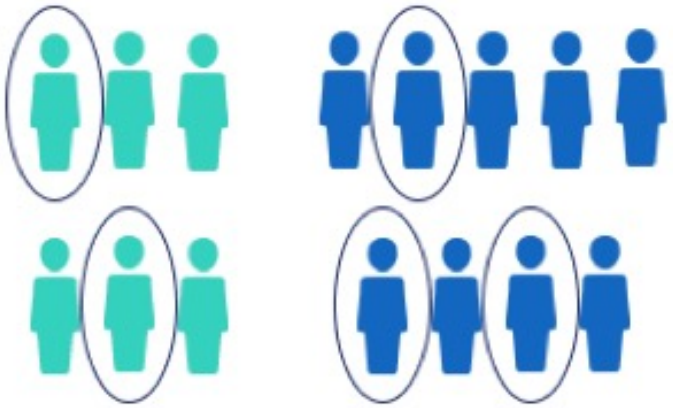
Probability sampling methods

- Simple random sampling: every member of the population has an equal chance of being selected. Your sampling frame should include the whole population.

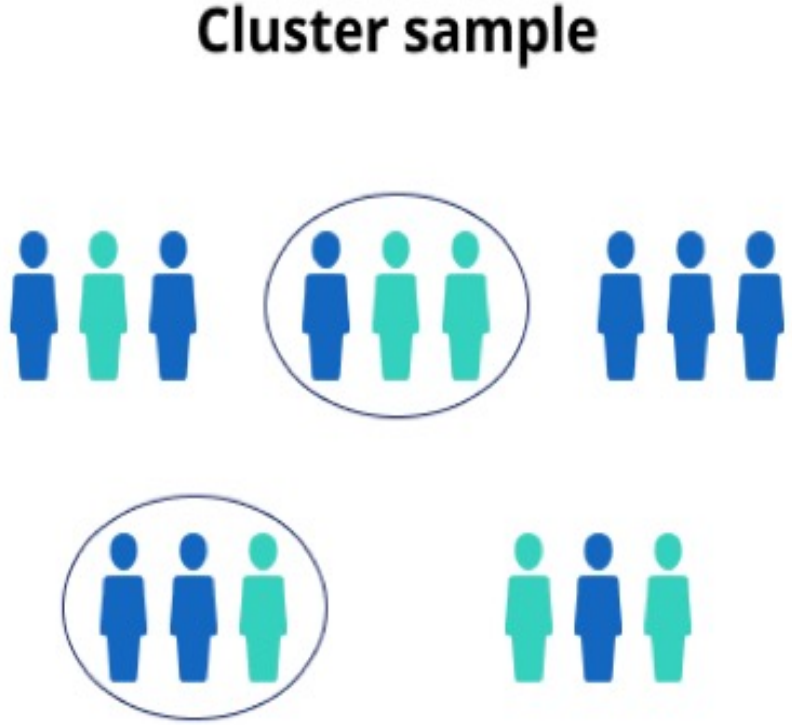


- Systematic/Interval sampling: every member of the population is listed with a number, but instead of randomly generating numbers, individuals are chosen at regular intervals.

Stratified sample



- Stratified sampling:
  - divide the population into subgroups (called strata) based on the relevant characteristic (e.g. gender, age range, income bracket, job role).
  - From the overall proportions of the population, calculate how many people should be sampled from each subgroup.
  - use random or systematic sampling to select a sample from each subgroup.



- Cluster sampling
  - Cluster the population into subgroups
  - Randomly select entire subgroups.
  - If the clusters themselves are large, sample individuals from within each cluster using one of the techniques above.

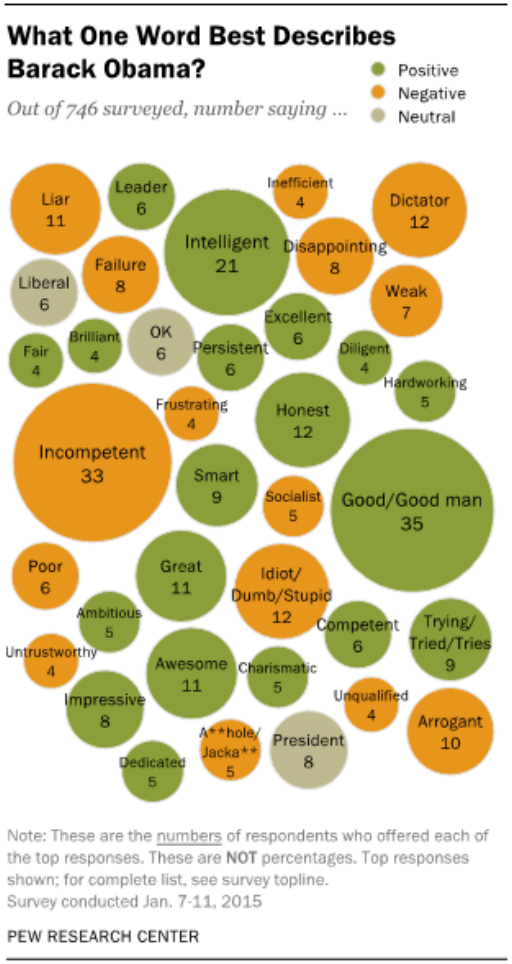
- Low quality data will lead to misleading stories!!
- Data quality criteria
  - Data-type Constraints
  - Range Constraints
  - Unique Constraints
  - Completeness
  - Consistency
  - Uniformity

- Data Cleaning
  - Inspection: A summary statistics about the data, called data profiling, is really helpful to give a general idea about the quality of the data.
  - Cleaning:
    - Type conversion
    - Standardize
    - Transformation
    - Missing values: (1) drop (2) Flag
    - Outliers: innocent until proven guilty
  - Verify: Re-inspect the data

- Organize the data: help in structuring it into a more useful form for communicating meaning
- Schemes:
  - Location
  - Alphabet
  - Time
  - Category
  - Hierarchy

- Quantitative Information in Visualization
  - Quantitative data indicate numerically measurable quantities about that base phenomenon, such as age, time, temperature, or number of items
  - Can be displayed through many types of graphs, charts, tables, and maps
  - Data can be displayed over time (such as a line chart)

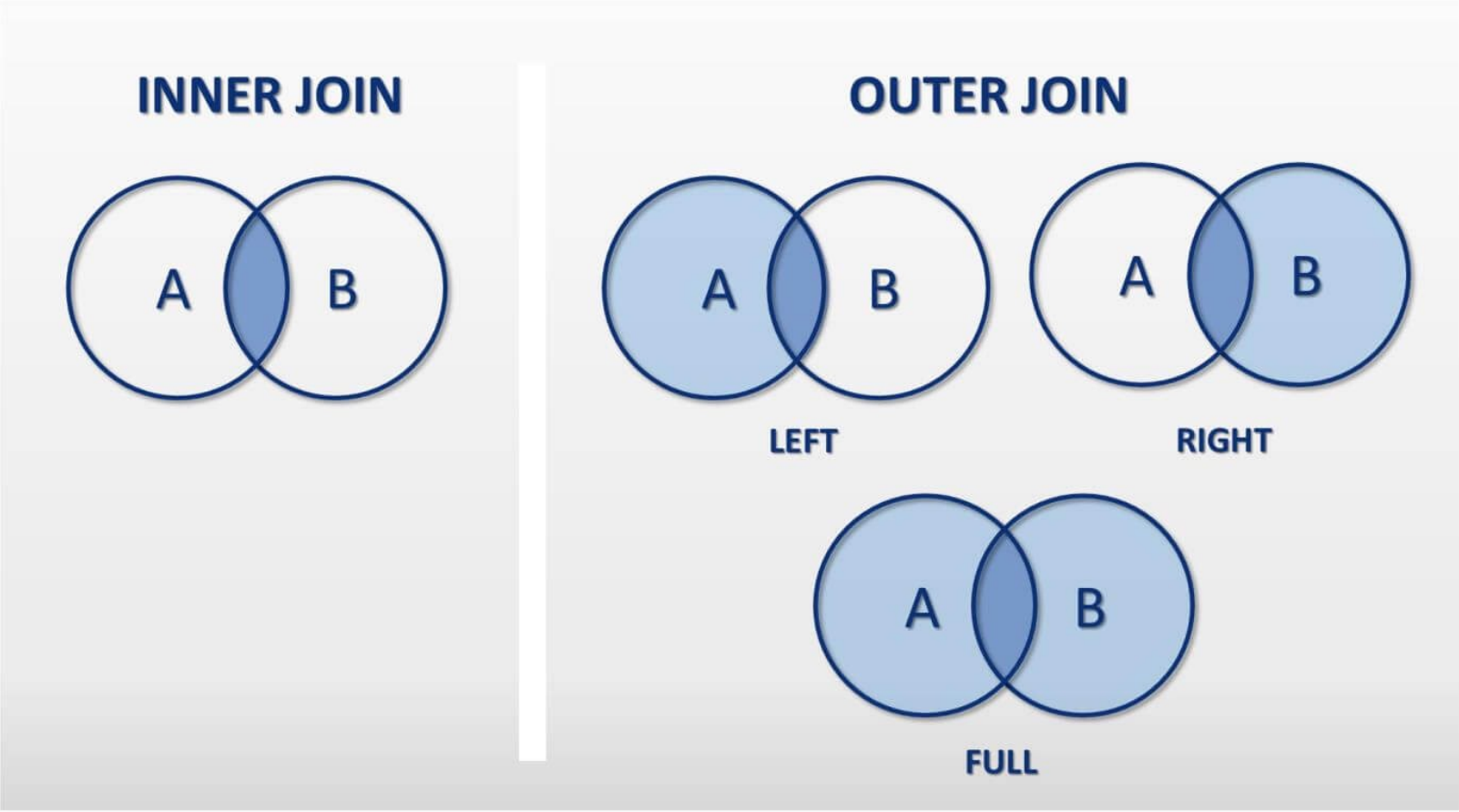
- Qualitative Information in Visualization
  - Qualitative data focus on the more descriptive qualities that explain the underlying phenomenon, such as documents, images, and categorizations.
  - Can tell a story



A Word Cloud

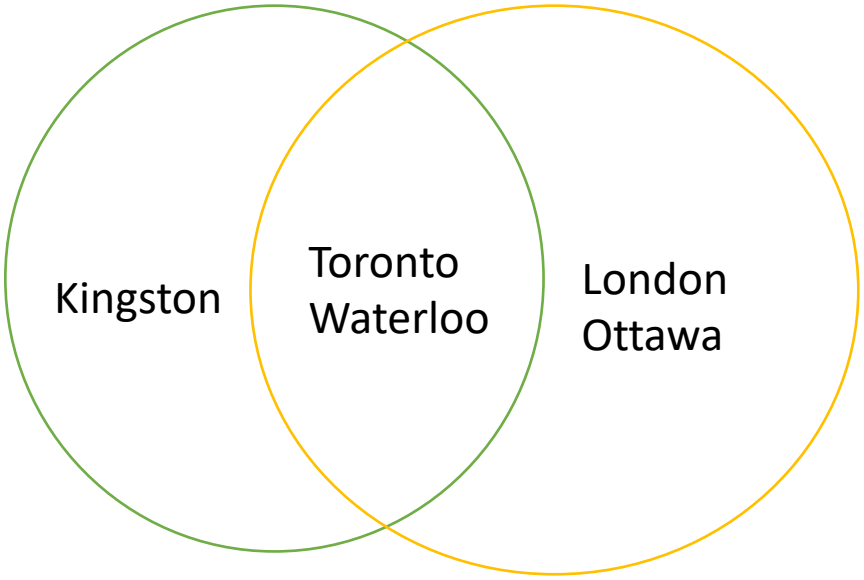
- Using Qualitative and Quantitative Data Together
  - The most effective arguments make use of both quantitative and qualitative types of information
  - Each provides support as appropriate to the base phenomenon and the data that purport to represent it.

Handling Multiple Data Sets



City	Temp
Toronto	-10
Waterloo	-5
Kingston	-7

City	Rainfall
London	15
Waterloo	30
Ottawa	50
Kitchener	20
Toronto	18

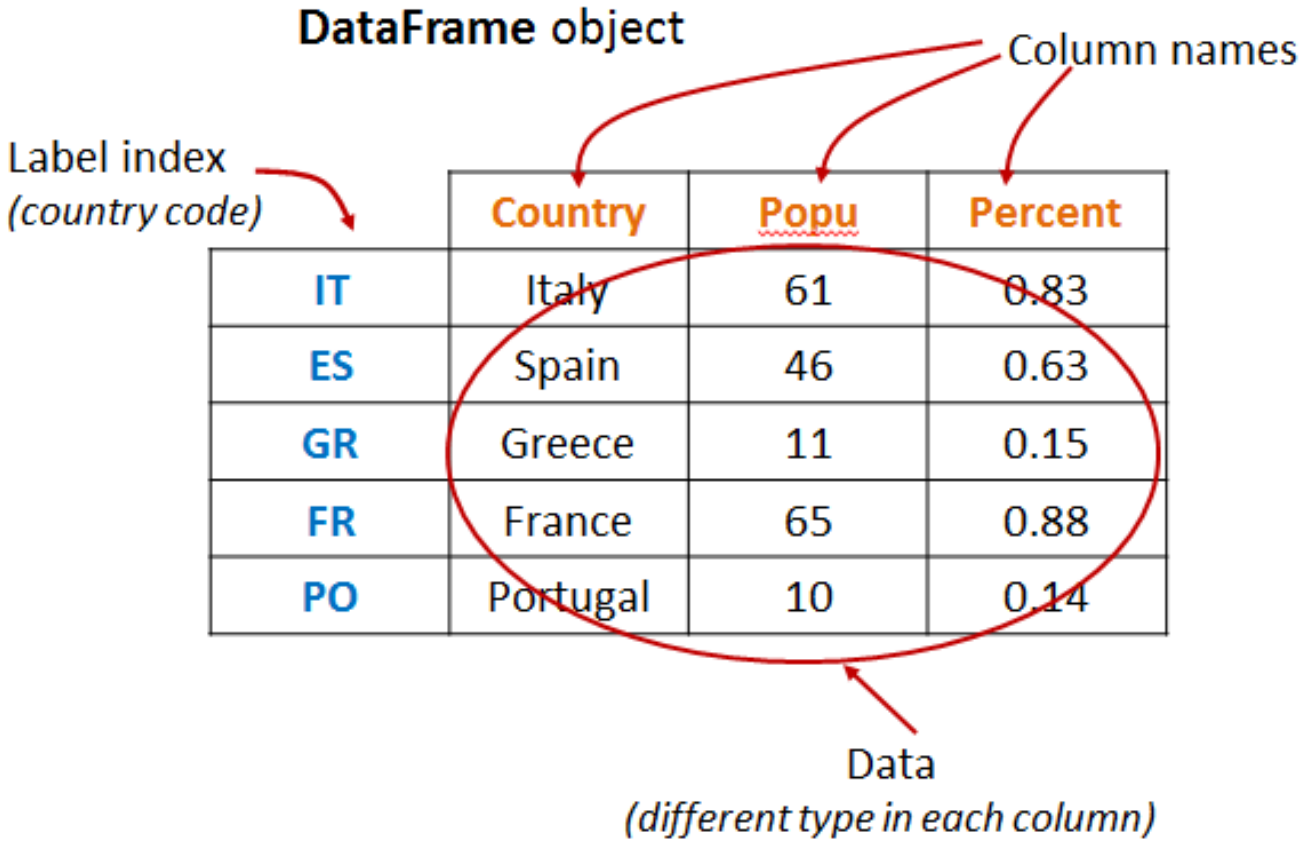


Left Join

City	Temp	Rainfall
Kingston	-10	N/A
Toronto	-5	18
Waterloo	-7	30

Appendix: Python Pandas

- Pandas: a **Python** package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.
- It takes data and creates a Python object with rows and columns called dataframe that looks very similar to table


DataFrame object
Column names
Label index
(country code)
Country
Popu
Percent
IT
Italy
61
0.83
ES
Spain
46
0.63
GR
Greece
11
0.15
FR
France
65
0.88
PO
Portugal
10
0.14
Data
(different type in each column)

Dataframe Basics

- df.shape - returns dimensions of the data tuple(row num, col num)
- df.head() – returns top rows
- df.tail(n) – returns top n rows
- df.index – returns all indexes
- df.columns – returns all columns
- df.to_numpy() - convert to a numpy array
- df.T - transpose

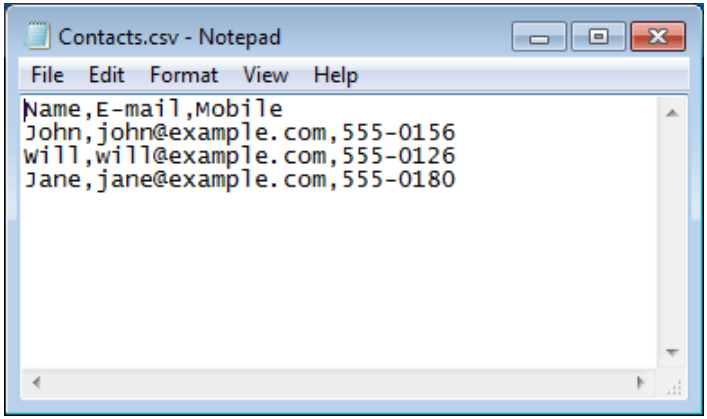
Read from Data Files

- CSV

  import pandas as pd

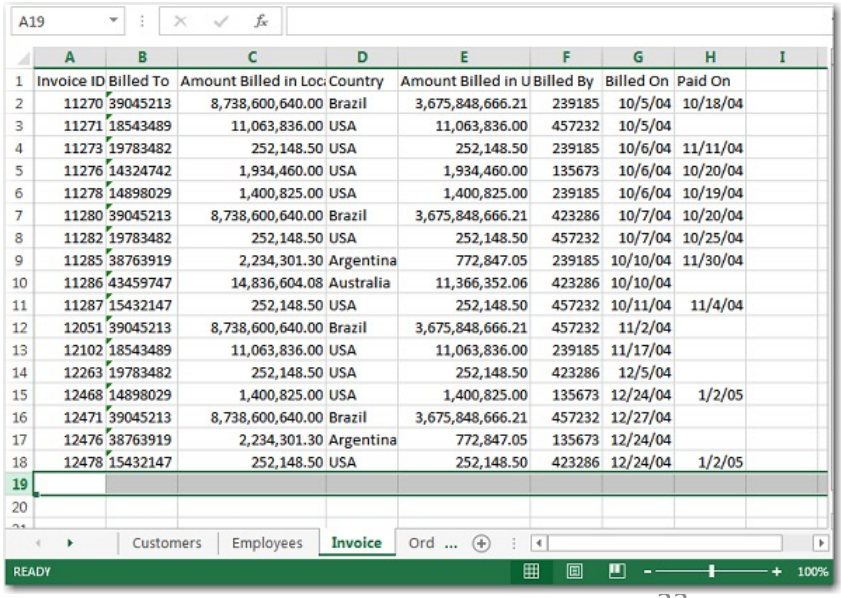
  df =pd.read_csv("/path/data.csv")

  df =pd.read_csv("/path/data.csv", header=None, index=False)



- XLSX

  df = pd.read_excel("/path/data.xlsx", sheetname = "Invoice")



	A	B	C	D	E	F	G	H
1	Invoice ID	Billed To	Amount Billed in Loc	Country	Amount Billed in U	Billed By	Billed On	Paid On
2	11270	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	239185	10/5/04	10/18/04
3	11271	18543489	11,063,836.00	USA	11,063,836.00	457232	10/5/04	
4	11273	19783482	252,148.50	USA	252,148.50	239185	10/6/04	11/11/04
5	11276	14324742	1,934,460.00	USA	1,934,460.00	135673	10/6/04	10/20/04
6	11278	14898029	1,400,825.00	USA	1,400,825.00	239185	10/6/04	10/19/04
7	11280	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	423286	10/7/04	10/20/04
8	11282	19783482	252,148.50	USA	252,148.50	457232	10/7/04	10/25/04
9	11285	38763919	2,234,301.30	Argentina	772,847.05	239185	10/10/04	11/30/04
10	11286	43459747	14,836,604.08	Australia	11,366,352.06	423286	10/10/04	
11	11287	15432147	252,148.50	USA	252,148.50	457232	10/11/04	11/4/04
12	12051	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	457232	11/2/04	
13	12102	18543489	11,063,836.00	USA	11,063,836.00	239185	11/17/04	
14	12263	19783482	252,148.50	USA	252,148.50	423286	12/5/04	
15	12468	14898029	1,400,825.00	USA	1,400,825.00	135673	12/24/04	1/2/05
16	12471	39045213	8,738,600,640.00	Brazil	3,675,848,666.21	457232	12/27/04	
17	12476	38763919	2,234,301.30	Argentina	772,847.05	135673	12/24/04	
18	12478	15432147	252,148.50	USA	252,148.50	423286	12/24/04	1/2/05
19								
20								

Be careful with the header (first row) and index (first column)

- ZIP

```
import zipfile
archive = zipfile.ZipFile('T.zip', 'r’)
data = archive.read('data.csv’)
```

- Plain Text (txt)

```
text_file = open("text.txt", "r")
lines = text_file.read()
```

- JSON

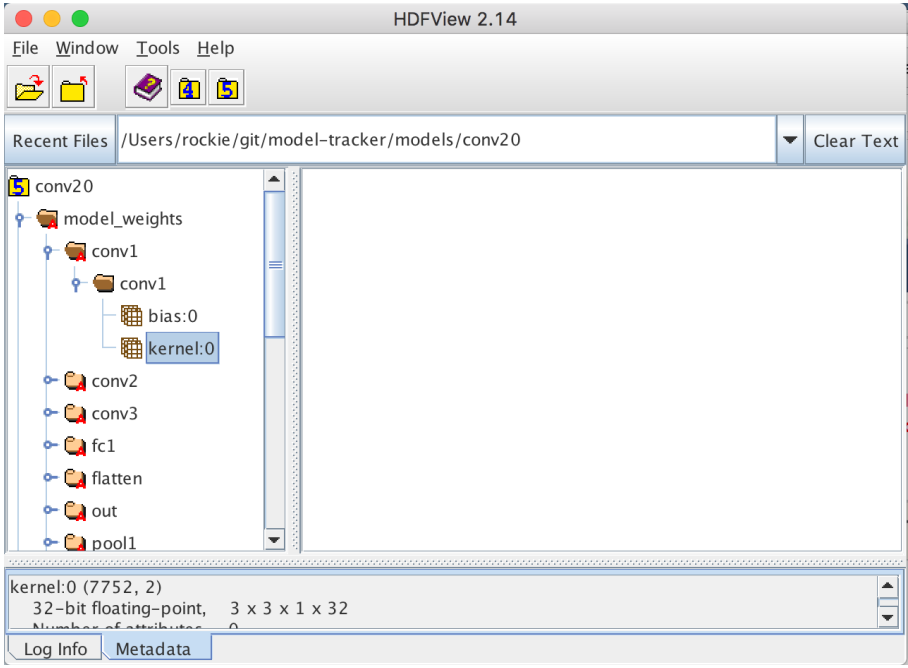
  import pandas as pd

  df = pd.read_json("/path/data.json")

- Hierarchical Data Format

  t = pd.read_hdf('data.h5')

- XML – Beautiful Soup
- HTML– Beautiful Soup



Data Selection with Pandas Dataframe

- Select column(s)
  - df[col], df[[col1, col2]]
- Select rows
  - By position df.iloc[0]
  - Multiple rows by position df.iloc[0,:]
  - By index s.loc['index_one’]
- Select both column and row
  - df.iloc[0,0]

- Filter
  - df[df[year] > 1984]
  - Boolean filtering using & (and) or | (or) to add different conditions
- Sort
  - Sort by a column in ascending order
    df.sort_values(col1)
  - Sort by a column in descending order
    df.sort_values(col2,ascending=False)
  - Sort by two columns
    df.sort_values([col1,col2],ascending=[True,False])
- Groupby
  - Splitting the data into groups based on some criteria
  - df.groupby(col), df.groupby([col1,col2])

Statistics with Pandas Dataframe

- df.mean() Returns the mean of all columns
- df.corr() Returns the correlation between columns in a data frame
- df.count() Returns the number of non-null values in each data frame column
- df.mean() Returns the mean in each column
- df.max() Returns the highest value in each column
- df.min() Returns the lowest value in each column
- df.median() Returns the median of each column
- df.std() Returns the standard deviation of each column
- df.describe() Returns multiple statistical details

Cleaning with Pandas Dataframe

- Check for missing values

  pd.isnull()
- Total number of missing values

  pd.isnull().sum()
- Drop missing values
  - By row: df.dropna()
  - By column: df.dropna(axis=1)
- Fill in missing values
  - df.fillna(x)
- Replace values
  - df.replace(one, 1)
  - df.replace(['one','three’], [1,3])

Handle multiple data sets with pandas dataframes

- df1.append(df2)— add the rows in df1 to the end of df2 (columns should be identical)
- df.concat([df1, df2],axis=1) — add the columns in df1 to the end of df2 (rows should be identical)
- df1.join(df2,on=col1,how='inner') — SQL-style join the columns in df1 with the columns on df2 where the rows for col have identical values. how can be equal to one of: 'left', 'right', 'outer', 'inner'