Python for Data Science and AI

Pandas - Data Analysis

INSTITUTO INFNET
1994

Prof. Charles Prado

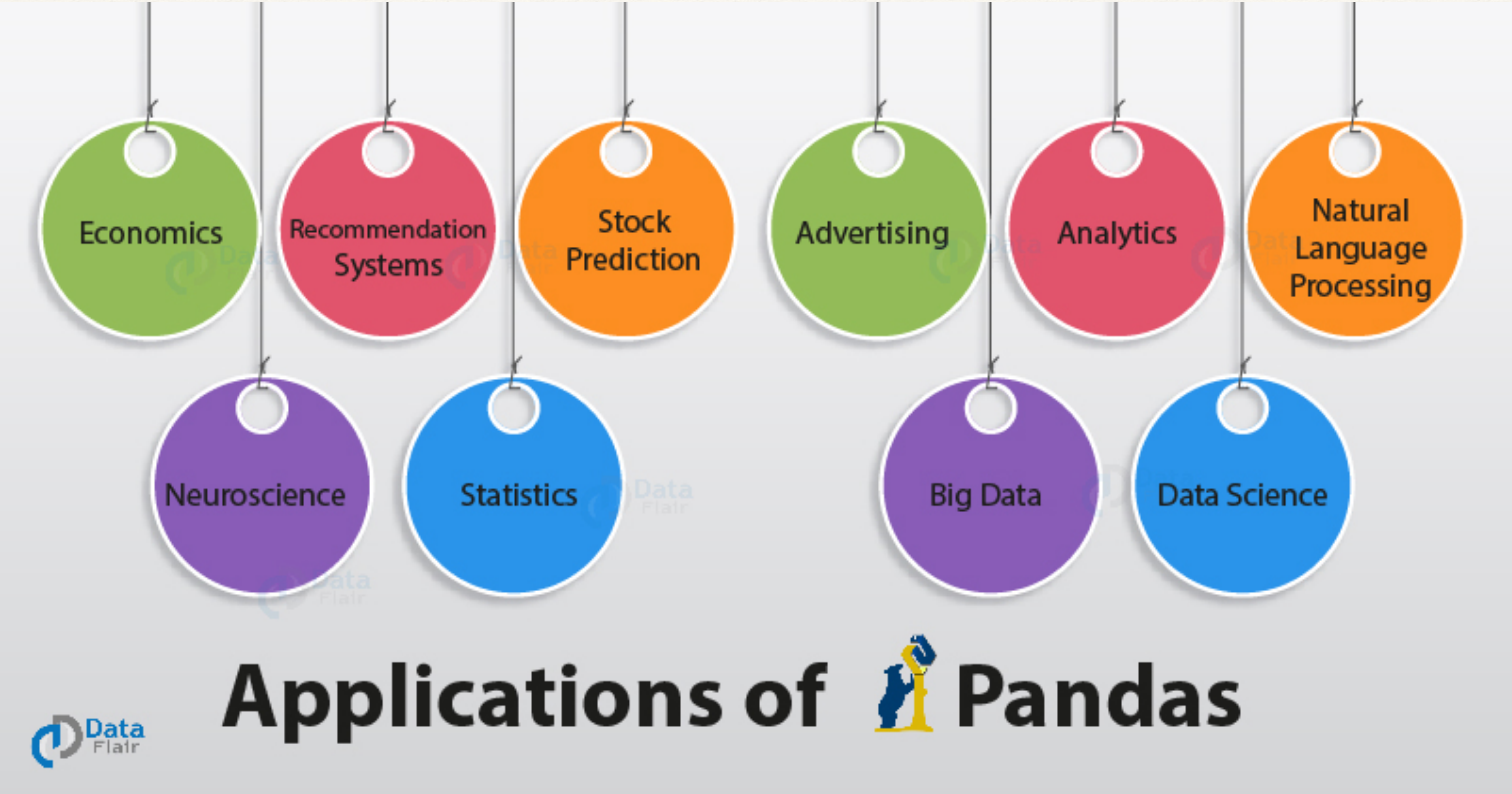
Pandas

- Motivation
- Loading Data with Pandas
  - Importing Pandas
  - Dataframes
  - Using loc, iloc
- Saving Data with Pandas
- Comparison with SQL (Some examples)
  - SELECT
  - WHERE
  - GROUP BY
  - UNION
- Data Analysis - COVID19

Motivation

- Pandas is a software library (Python)
  - Data manipulation and analysis
  - Numerical Tables
  - Time series
  - Data Visualization

Motivation


Economics
Recommendation Systems
Stock Prediction
Advertising
Analytics
Natural Language Processing
Neuroscience
Statistics
Big Data
Data Science
Applications of Pandas
Data Flair

Importing Pandas

command

Library

- # import pandas as pd
- # csv_path = ‘archive.csv’ or ‘archive.xls’
- # url = (‘https://……’)
- # df = pd.read_csv(csv_path or url)

Pandas
read_csv()
Series()
DataFrame
Values
….

Data - Pandas

- Two primary data structures of pandas:
  - *Series* - (1-dimensional)
  - *DataFrame* - (2-dimensional)
  - Handle the vast majority of typical use cases in ***finance***, ***statistics***, ***social science***, and many areas of ***engineering***.

Dimensions	Name	Description
1	Series	1D labeled homogeneously-typed array
2	DataFrame	General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed column

DataFrame:

- data – The data from which the dataframe will be made
- index – States the index from dataframe
- columns – States the column label
- dtype – The datatype for the dataframe
- copy – Any copied data taken from inputs

Creating DataFrames

- Using Dictionary

```
>>> data={'student': ['Jack','Mike','Rohan','Zubair'], 'year':[1,2,3,1], ‘marks':[9.8,6.7,8,9.9]}
>>> dataflair_df=pd.DataFrame(data)
>>> dataflair_df
```

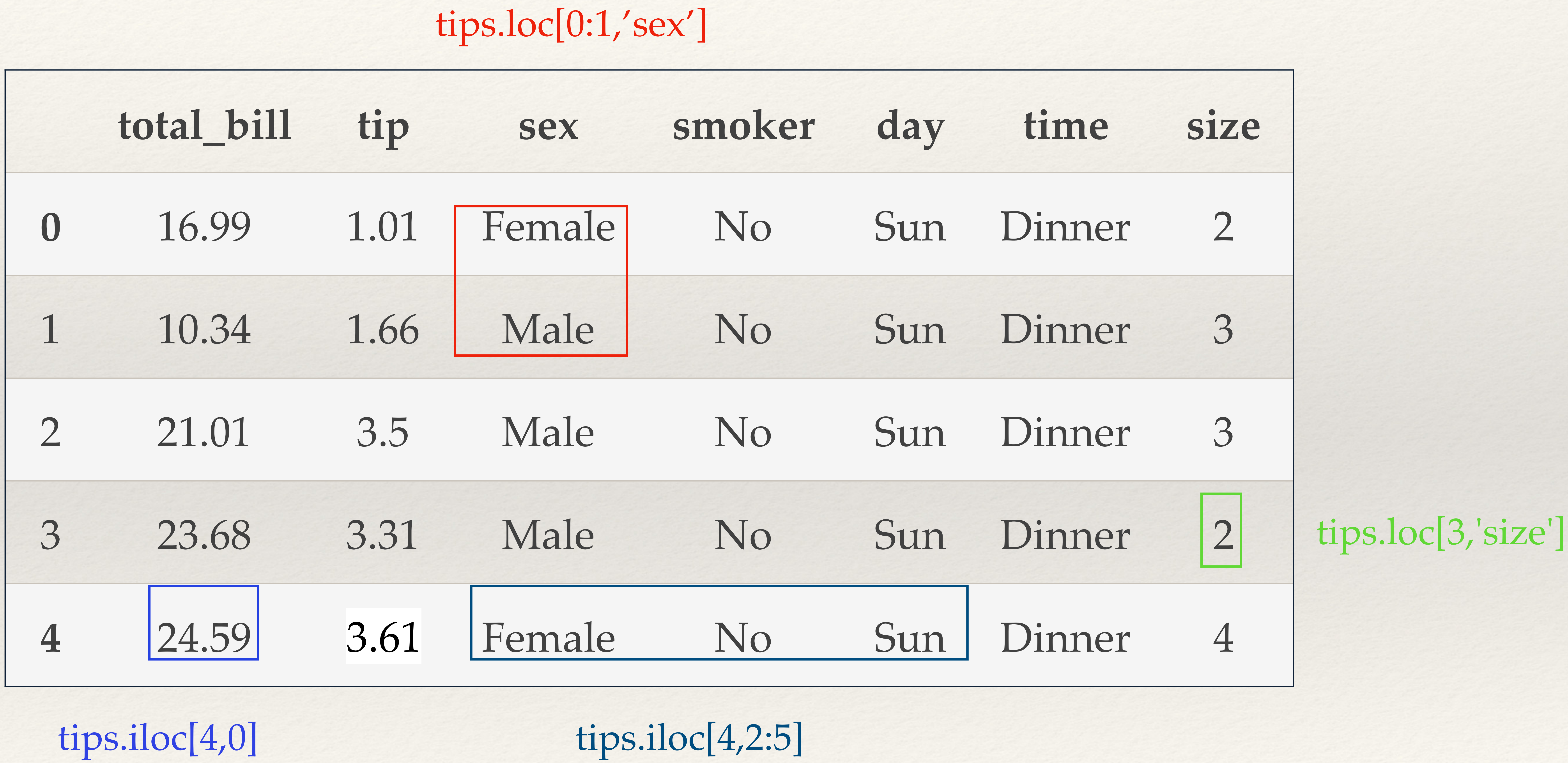
- Using the original DataFrame
  - df2 = tips[[‘sex','day']]
  - df3 = tips.sex
- Reading from File or URL

Columns

Rows

	Total_bill	**Tip**	**Sex**	**Smoker**	**Day**	**Time**	**Size**
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3	Female	No	Thur	Dinner	2

Using loc, iloc



tips.loc[0:1,’sex’]

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.5	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

tips.loc[3,'size']

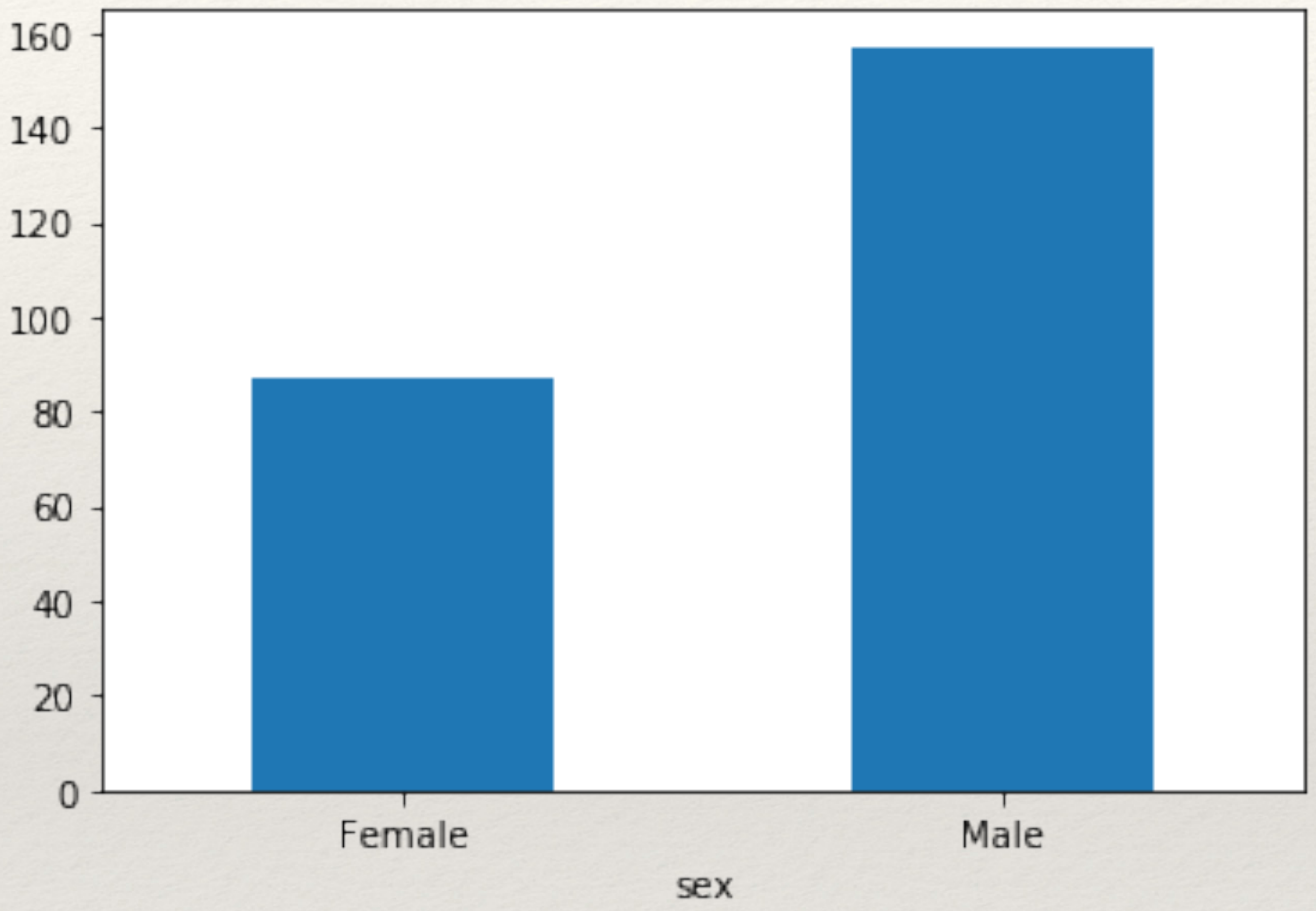
tips.iloc[4,0]

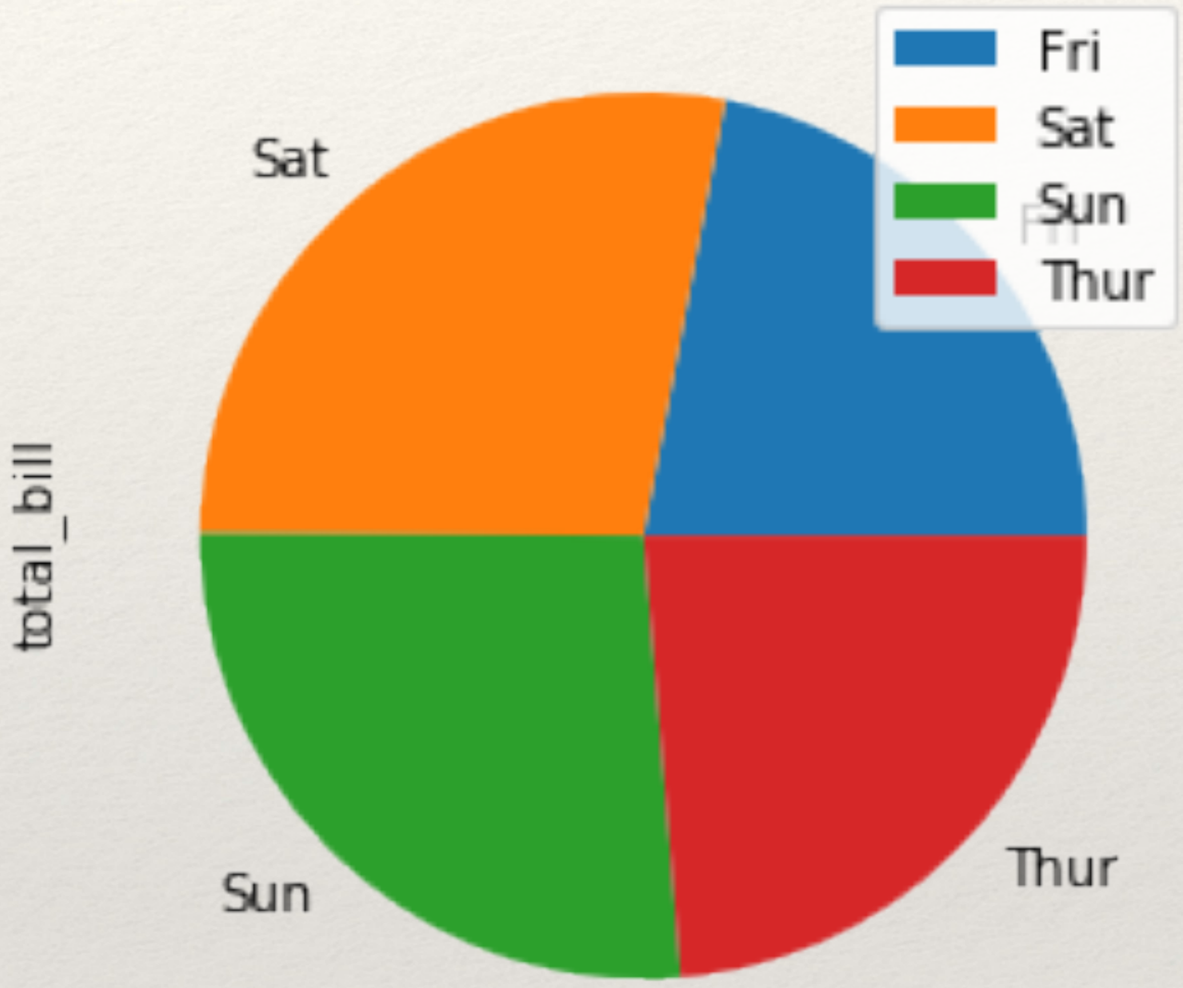
tips.iloc[4,2:5]

Pandas x SQL

- Comparison with SQL
- **SELECT** total_bill, tip, smoker, time FROM tips LIMIT 5;
  - *tips[['total_bill', 'tip', 'smoker', 'time']].head(5)*
- SELECT * FROM tips **WHERE** time = 'Dinner' LIMIT 5;
  - *tips[tips['time'] == 'Dinner'].head(5)*
- SELECT sex, count(*) FROM tips **GROUP BY** sex;
  - *tips_by_sex = tips.groupby('sex').size()*
- SELECT city, rank FROM df1 **UNION ALL** SELECT city, rank FROM df2;
  - *df1 = pd.DataFrame({'city': ['Chicago', 'San Francisco', 'New York City'], 'rank': range(1, 4)})*
  - *df2 = pd.DataFrame({'city': ['Chicago', 'Boston', 'Los Angeles'], 'rank': [1, 4, 5]})*
  - *pd.concat([df1, df2])*
  - *pd.concat([df1, df2]).drop_duplicates()*

Visualization - Examples

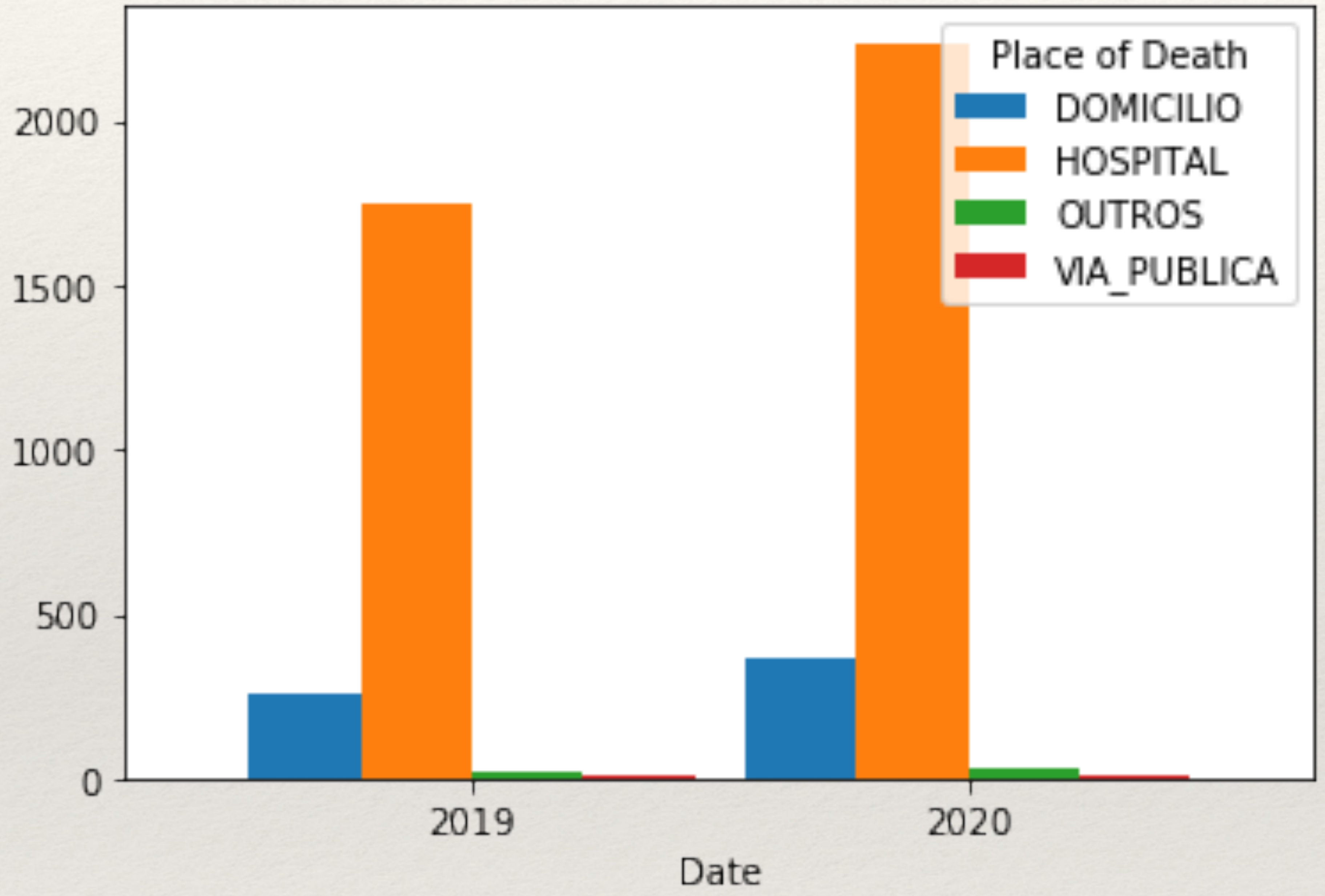


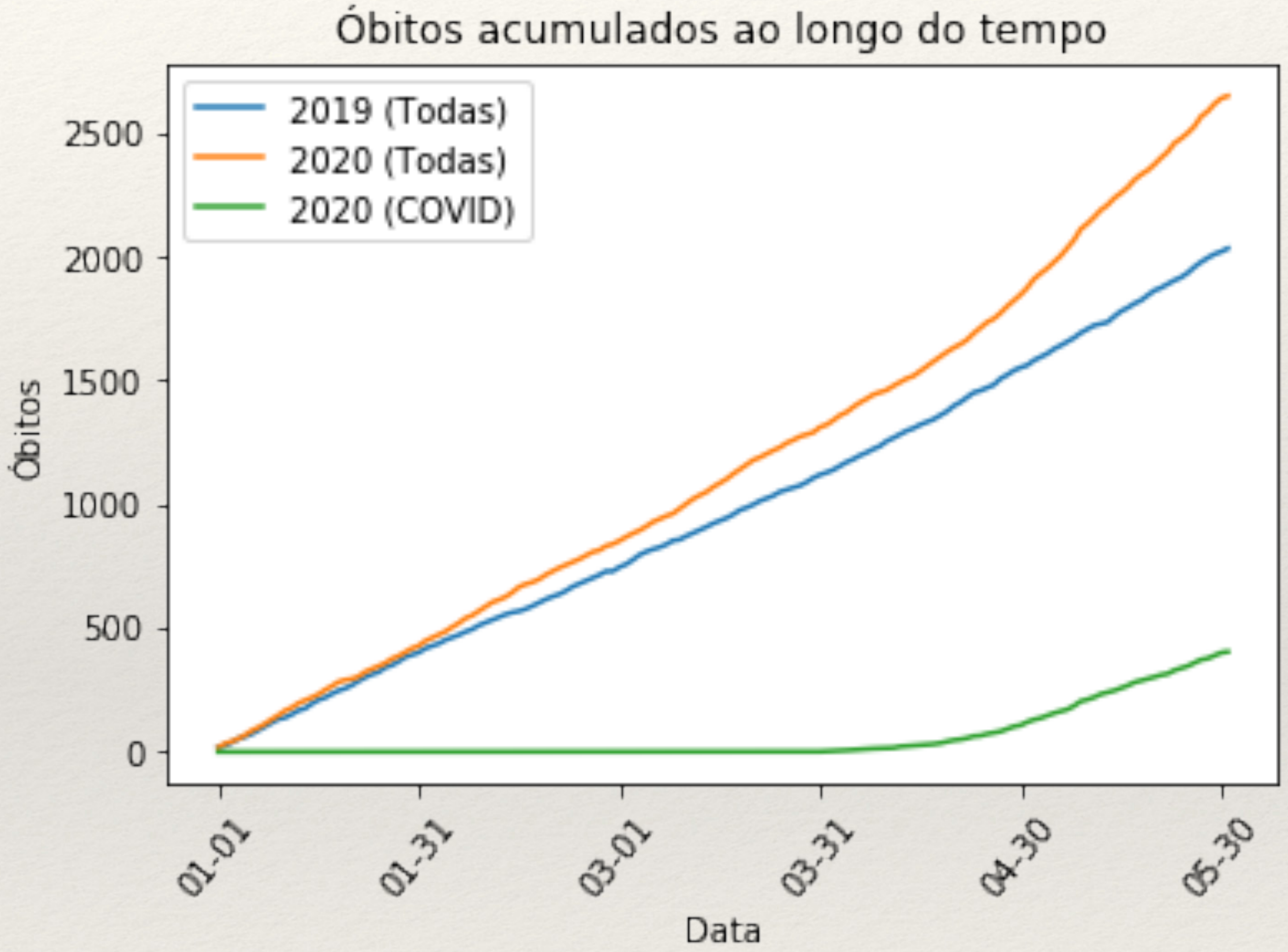


Saving Data with Pandas

- CSV Format - tips.to_csv('tips2.csv')
- Excel Format - tips.to_excel(‘tips2.xlsx’, sheet_name = ‘tips’)

Analyzing COVID19 using Pandas


Place of Death
DOMICILIO
HOSPITAL
OUTROS
VIA_PUBLICA
Date


Óbitos acumulados ao longo do tempo
2019 (Todas)
2020 (Todas)
2020 (COVID)
Óbitos
Data