Python for Economics

Zhenhua He

Afternoon session, 9/17/2021

Table of Contents

This course is divided into numbered lessons

13. Matplotlib
14. Pandas
    - Wrap up

Python Libraries Covered

matplotlib	Plotting data
pandas	Analyzing, cleaning, and manipulating data

Lesson 13
Data visualization with Matplotlib

Use Python Matplotlib library for data visualization

Learning Objectives

After this lesson, you will know how to make:

- Scatter plot and Line plot
- Color map
- Contour figures
- 3D figures
  - Surface plots
  - Wire-frame plot
  - Contour plots with projections

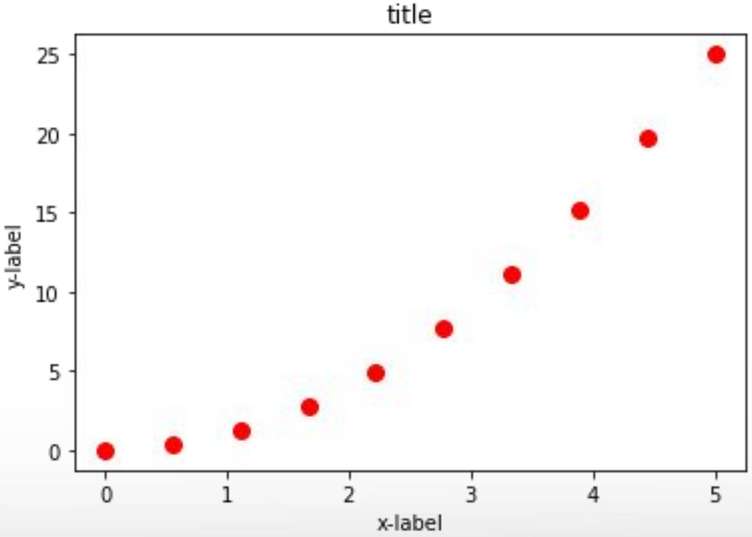
Anatomy of a Scatter Plot

Marker

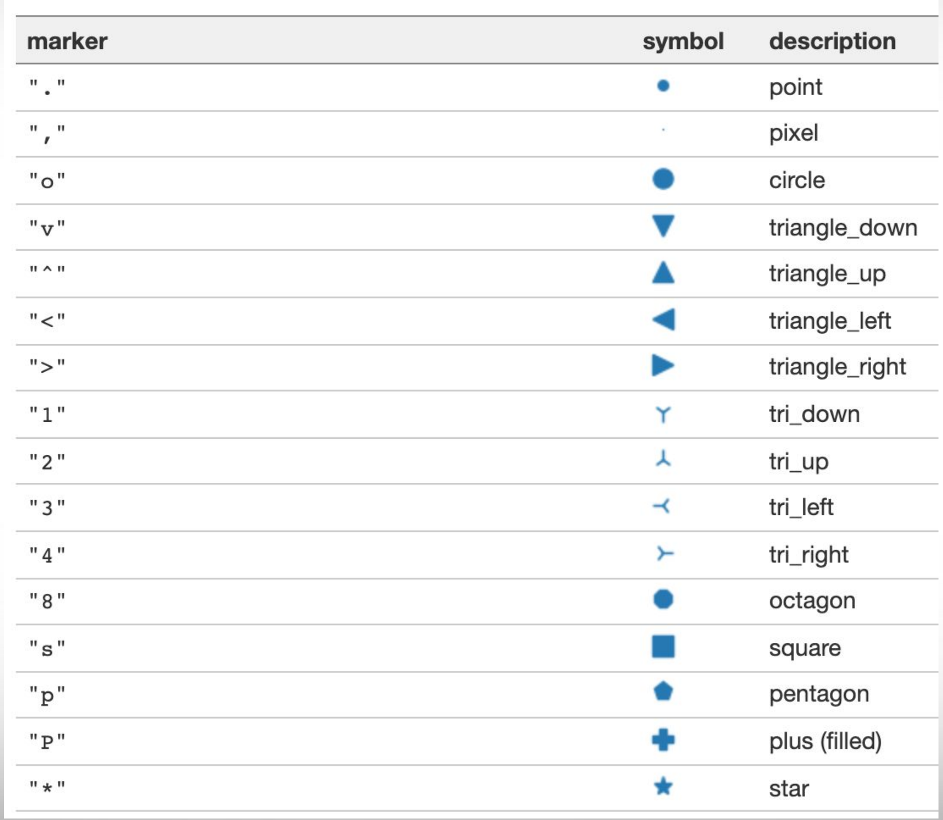
- style
- size
- color

Figure

- title
- xlabel
- ylabel



Scatter plot - Marker symbols



marker	symbol	description
"."	•	point
","	·	pixel
"o"	●	circle
"v"	▼	triangle_down
"^"	▲	triangle_up
"<"	◀	triangle_left
">"	▶	triangle_right
"1"	Y	tri_down
"2"	⅄	tri_up
"3"	≺	tri_left
"4"	≻	tri_right
"8"	⯃	octagon
"s"	■	square
"p"	⬟	pentagon
"P"	✚	plus (filled)
"*"	★	star

Hot Tip!

Give a module a nickname with `as`

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

Examples and Exercises

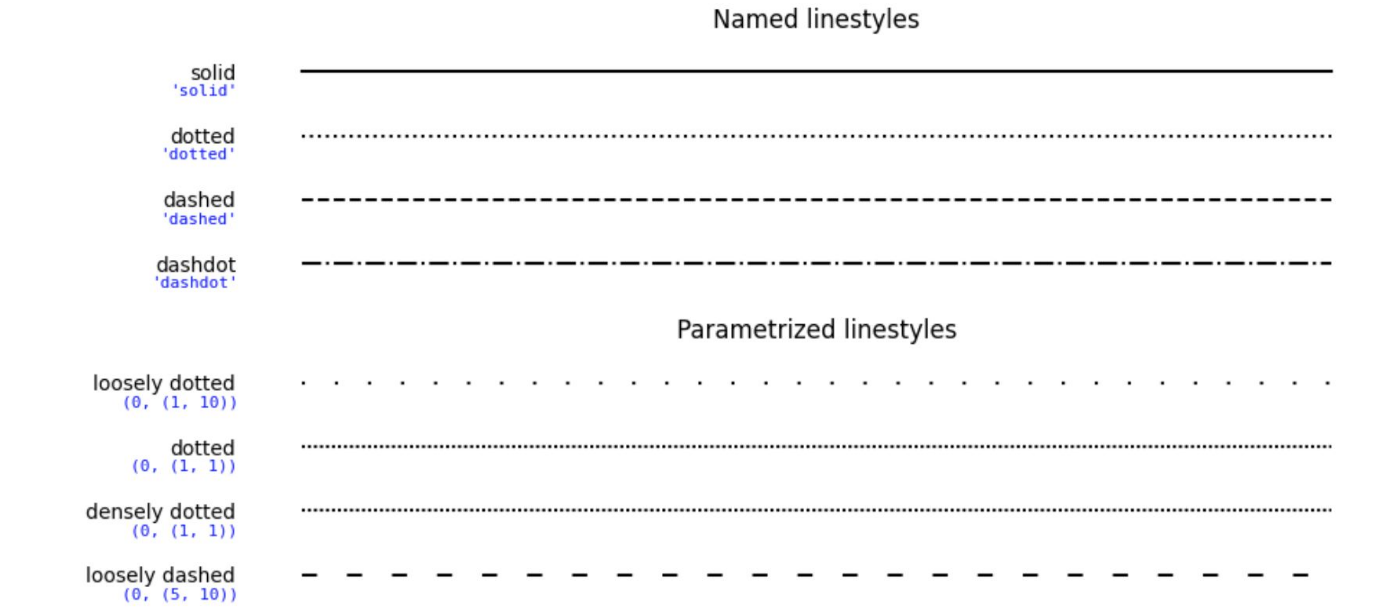
Go to Google Classroom assignment “Scatter Plot”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

Line plot

Simple line styles can be defined using the strings "solid", "dotted", "dashed" or "dashdot".



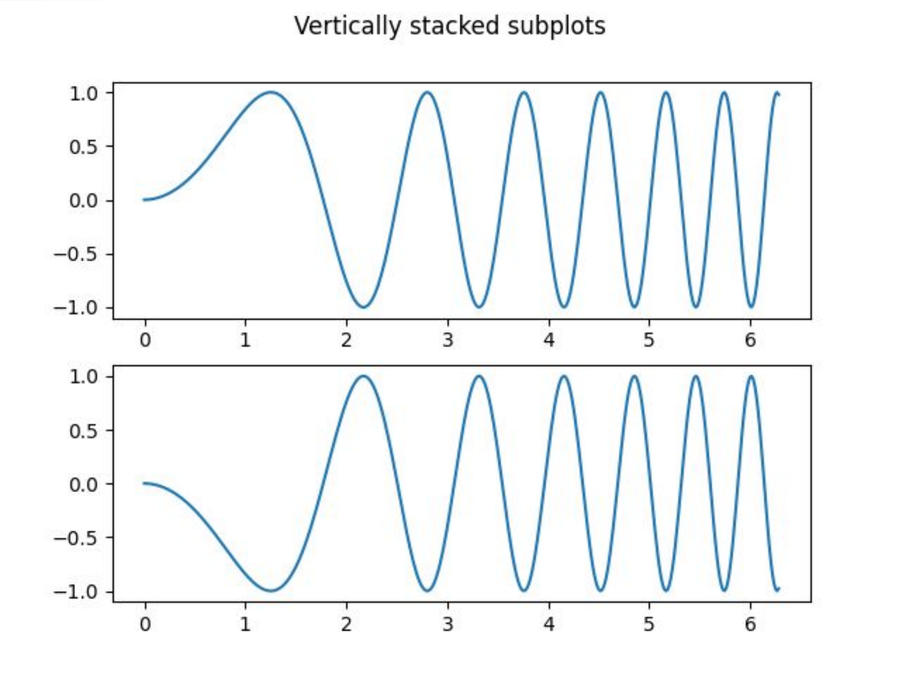
Examples and Exercises

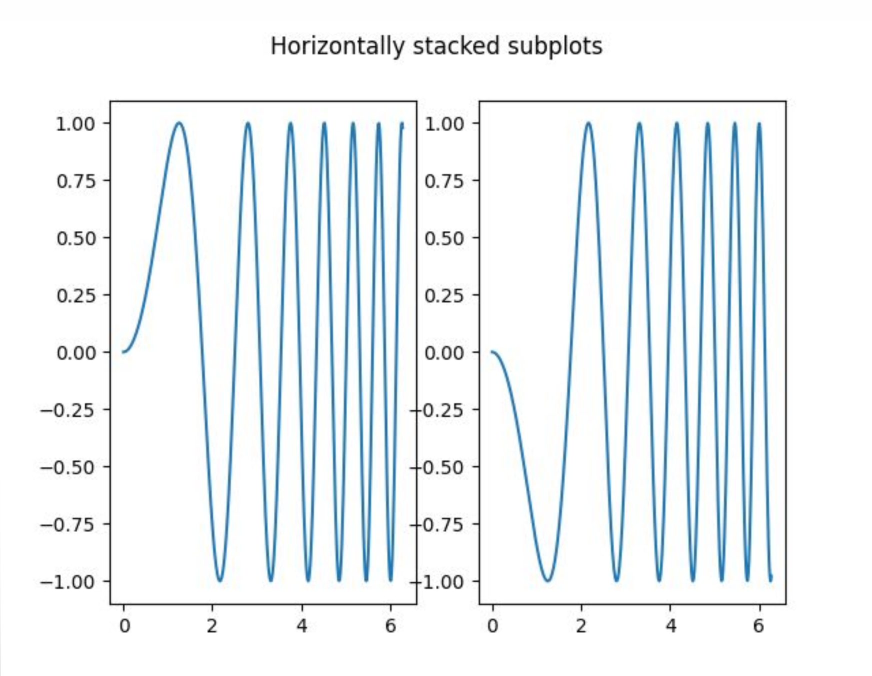
Go to Google Classroom assignment “Line Plot”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

Subplots


Vertically stacked subplots
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
6


Horizontally stacked subplots
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6

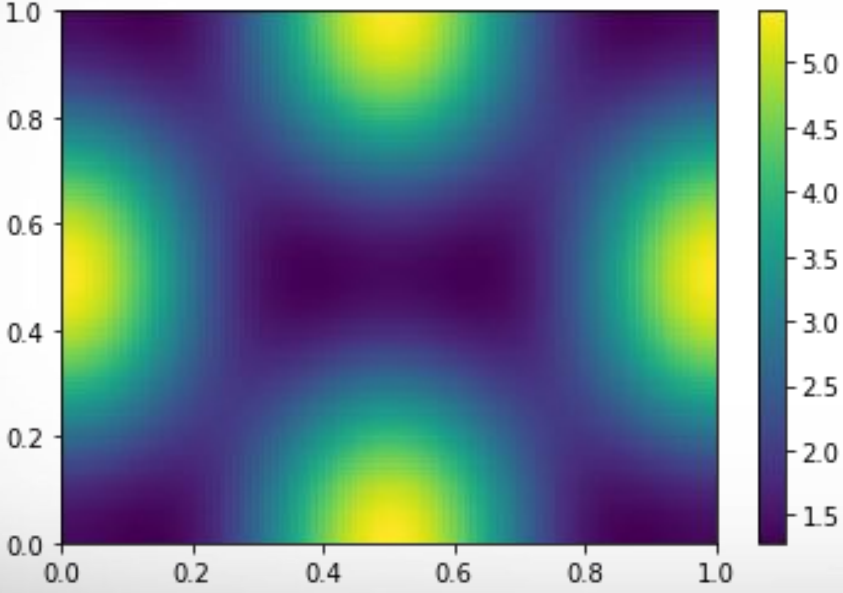
Exercises and Homework

Go to Google Classroom assignment “Subplots”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)
- Work on the homework (**due** by 9/23 11:59 PM)

Color map + savefig()



color map

- pcolor
- imshow

savefig()

- save the current figure

Examples and Exercises

Go to Google Classroom assignment “Color Plot”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

Break Time Reminder Slide

10 minutes break

Lesson 14
Pandas

Use Python Pandas library to manipulate data

Learning Objectives

After this lesson, you should know how to:

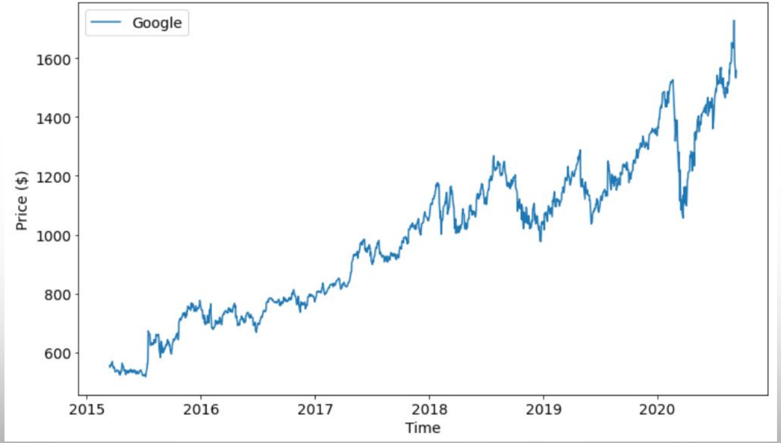
- Create a DataFrame
- Drop Entries
- Index, Select, and Filter data
- Sort data
- Handle missing and duplicate data
- Input and Output

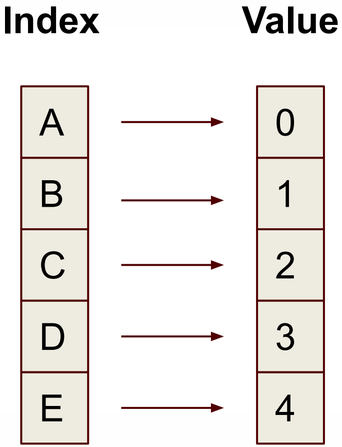
Pandas VS NumPy

NumPy	Pandas
Faster mathematical operations ✅	Slower mathematical operations
Only supports integer index	Customized index ✅
must use structured arrays	Easily handles different data types ✅
better performance when number of rows is 50K or less	better performance when number of rows is 500K or more ✅
more complicated to read and write files	simpler to read and write more file formats ✅

Series

- One-dimensional labeled array
- Capable of holding any data type (integers, strings, floating point numbers, etc.)
- Example: time-series stock price data





Index Value

A → 0

B → 1

C → 2

D → 3

E → 4

Array refresher -> Series

- index
- values
- get a value
- get a set of values
- filtering

Examples and Exercises

Go to Google Classroom assignment “Series”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

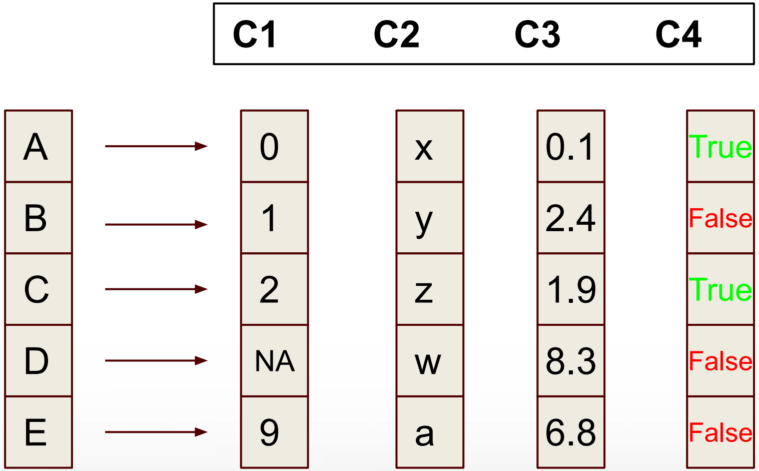
  Create a series -

  index: datetime;

  values: randomly generated stock price.

DataFrame

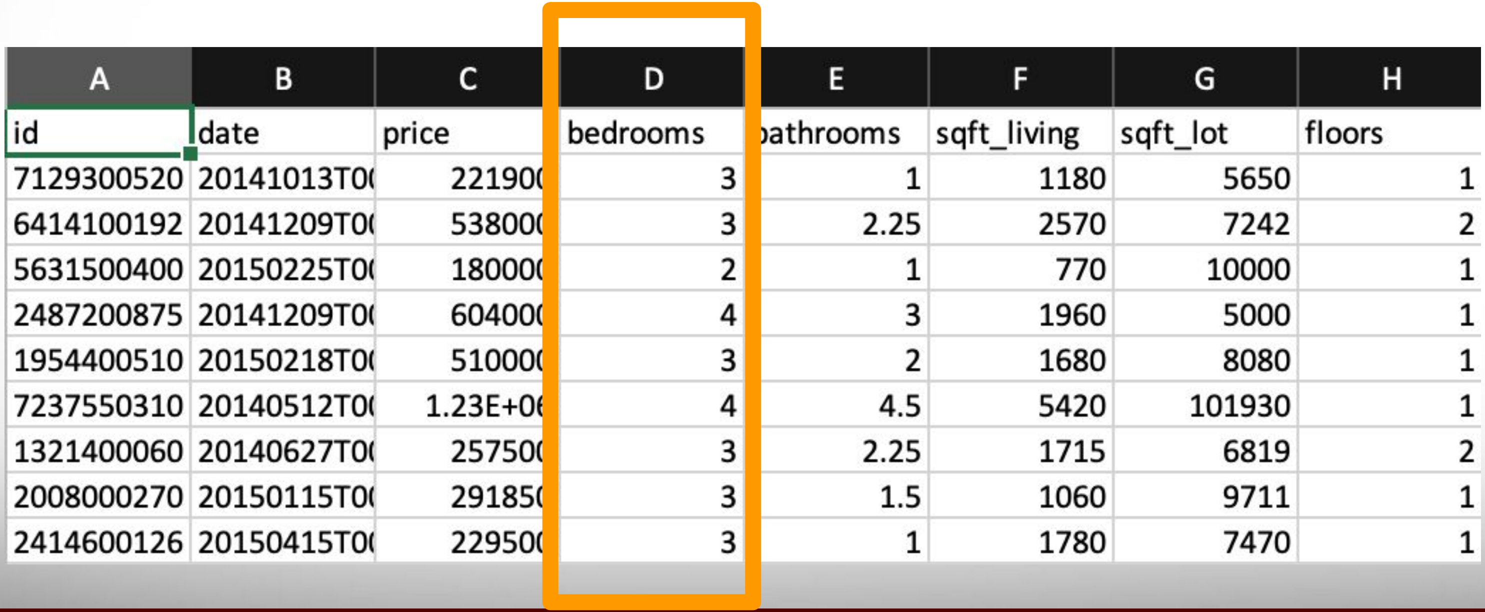
- Primary Pandas data structure
- A dict-like container for Series objects
- Two-dimensional size-mutable
- Heterogeneous tabular data structure



	C1	C2	C3	C4
A	0	x	0.1	True
B	1	y	2.4	False
C	2	z	1.9	True
D	NA	w	8.3	False
E	9	a	6.8	False

DataFrame Example

house sale data



A	B	C	D	E	F	G	H
id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
7129300520	20141013T0(	22190(	3	1	1180	5650	1
6414100192	20141209T0(	53800(	3	2.25	2570	7242	2
5631500400	20150225T0(	18000(	2	1	770	10000	1
2487200875	20141209T0(	60400(	4	3	1960	5000	1
1954400510	20150218T0(	51000(	3	2	1680	8080	1
7237550310	20140512T0(	1.23E+0(	4	4.5	5420	101930	1
1321400060	20140627T0(	25750(	3	2.25	1715	6819	2
2008000270	20150115T0(	29185(	3	1.5	1060	9711	1
2414600126	20150415T0(	22950(	3	1	1780	7470	1

Creating a Data Frame

Ways to do so:

- from Dictionary
- from Numpy array
- Read file (read_csv, read_excel, read_stata, read_html, ...)

Dictionary

For example, you have a car and its information is as below,

- brand: Ford
- model: Mustang
- year: 1964

You can create a dictionary as below

```
car_dict = {
 "brand": "Ford",
 "model": "Mustang",
 "year": 1964,
}
```

Examples and Exercises

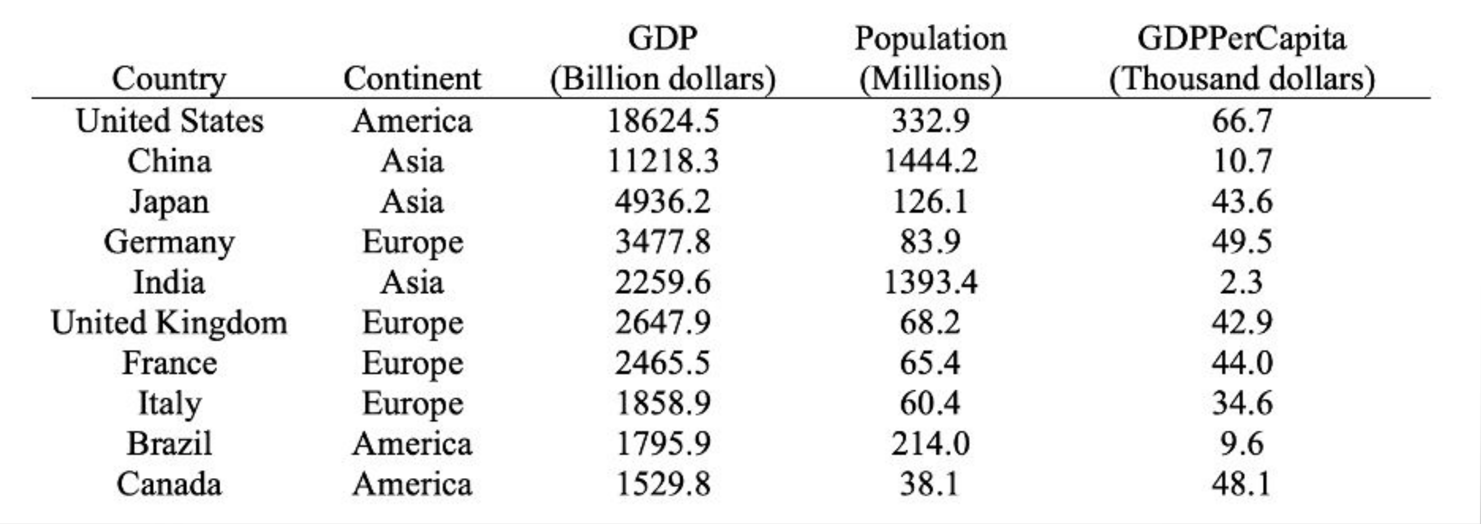
Go to Google Classroom assignment Pandas “DataFrame-1”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)
  1. Create a nation_economics DataFrame - including columns of Country, Continent, GDP, Population, GDPPerCapita
  2. Data on the next slide

Examples and Exercises

nation_economics data



Country	Continent	GDP (Billion dollars)	Population (Millions)	GDPPerCapita (Thousand dollars)
United States	America	18624.5	332.9	66.7
China	Asia	11218.3	1444.2	10.7
Japan	Asia	4936.2	126.1	43.6
Germany	Europe	3477.8	83.9	49.5
India	Asia	2259.6	1393.4	2.3
United Kingdom	Europe	2647.9	68.2	42.9
France	Europe	2465.5	65.4	44.0
Italy	Europe	1858.9	60.4	34.6
Brazil	America	1795.9	214.0	9.6
Canada	America	1529.8	38.1	48.1

Break Time Reminder Slide

10 minutes break

DataFrame: data retrieval

- Retrieve a column
- Retrieve multiple columns
- Retrieve a row
- Retrieve multiple rows
- Drop entries

Examples and Exercises

Go to Google Classroom assignment Pandas “DataFrame-2”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

From the nation_economics DataFrame,

1. Retrieve the ***GDPPerCapita*** column
2. Retrieve the ***United Kingdom*** row
3. Drop the ***Population*** column
4. Drop the ***Canada*** row

DataFrame: operations/manipulation

- Selecting with slicing
- Filtering
- Sorting
    - sort by index
    - sort by values

Examples and Exercises

Go to Google Classroom assignment Pandas “DataFrame-3”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)

  From the national_economics DataFrame

  1. Select the last 5 rows
  2. Select the rows with the population greater than 100M
  3. Sort the DataFrame by GDPPerCapita in descending order
- Work on the homework (**due** by 9/23 11:59 PM)

DataFrame: input and output

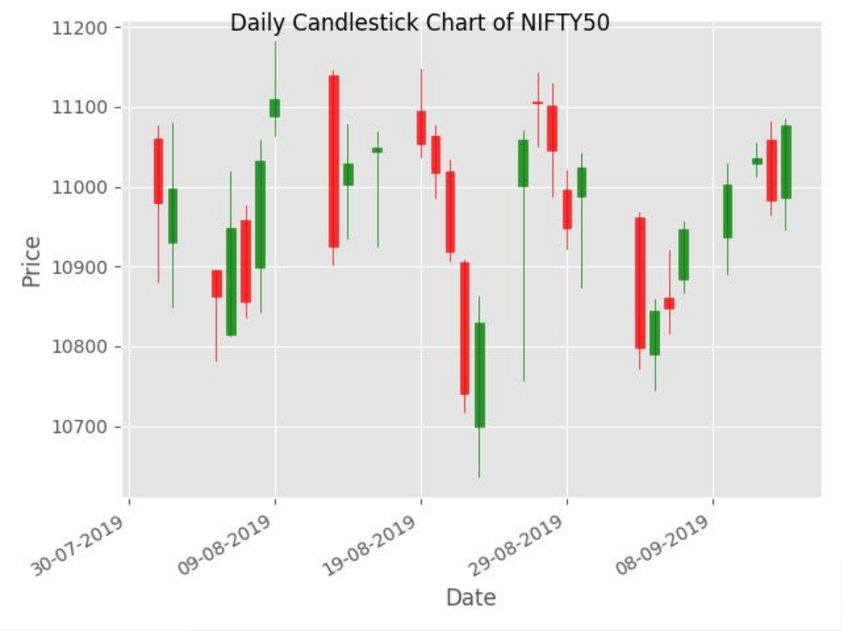
- Read/Write
- Different file formats
- describe()

Capstone - Candlestick Chart

A financial chart to depict price movement.

Four data values per marker:

- High
- Low
- Open
- Close



Exercise and Homework

Go to Google Classroom assignment “Matplotlib-Candlestick”

Tasks

- Follow instructions for the examples
- Work on the exercises (**due** by 9/17 6:00 PM)
- Work on the homework (**due** by 9/23 11:59 PM)

Day 2 wrap-up

almost time to go home

Practice for next week

Most important skills to master

- List loops
- Filtering with conditionals
- Pandas DataFrame structure

Slides from today are available in Google Classroom

Homework Assignments

- Lesson 9: “Lists and Strings”
- Lesson 10: “National Economic Data”
- Lesson 11: “Talking Cats”
- Lesson 12: “Array Quiz”
- Lesson 13: “Matplotlib - Subplots”
- Lesson 13: “Matplotlib - Candlestick chart”
- Lesson 14: “Pandas - DataFrame operations”

Please submit your homework assignments before 9/23 11:59 PM
Turn in your in-class exercises before 6:00 PM today

Office Hours

Please come to our office hours for assistance

- `M` 10 - 11 am Blocker 219B
- `T` 10 - 11 am (on Zoom only)
- `W` 2 - 4:30 pm Blocker 219B
- `R` 2 - 3 pm Blocker 219B

Please join our slack channel for discussion

- Workspace `sweeterworkspace.slack.com`
- Channel `hprc-econ-fall-21` (private channel)

New HPRC Help Resource

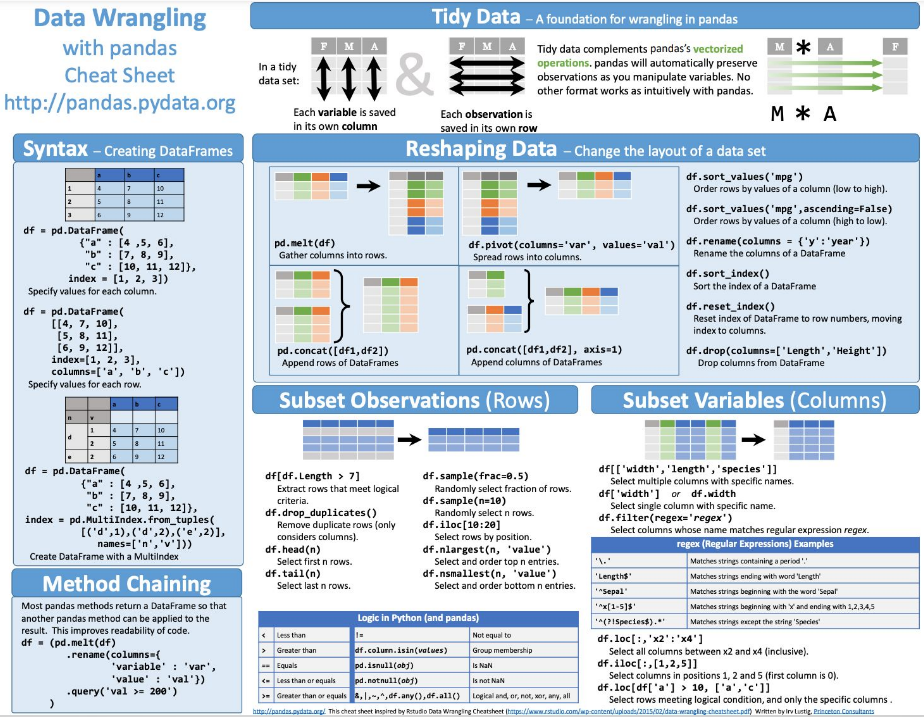
Bring Your Own Code (BYOC) sessions

These sessions are meant to help researchers overcome general Python programming hurdles in their research projects.

In person (Rooms 218A and 217B) or via zoom
Weekly on Wednesdays from 3-4:30pm through December 15.

Contact help@hprc.tamu.edu

Pandas Cheat Sheet (continued learning)



Data Wrangling with pandas Cheat Sheet http://pandas.pydata.org

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:

Each **variable** is saved in its own **column**

&

Each **observation** is saved in its own **row**

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

M * A

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
         {"a" : [4 ,5, 6],
          "b" : [7, 8, 9],
          "c" : [10, 11, 12]},
        index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
     [[4, 7, 10],
      [5, 8, 11],
      [6, 9, 12]],
     index=[1, 2, 3],
     columns=['a', 'b', 'c'])
```

Specify values for each row.

		a	b	c
n	v			
d	1	4	7	10
	2	5	8	11
e	2	6	9	12

```
df = pd.DataFrame(
         {"a" : [4 ,5, 6],
          "b" : [7, 8, 9],
          "c" : [10, 11, 12]},
index = pd.MultiIndex.from_tuples(
         [('d',1),('d',2),('e',2)],
            names=['n','v']))
```

Create DataFrame with a MultiIndex

Method Chaining

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
        .rename(columns={
                'variable' : 'var',
                'value' : 'val'})
        .query('val >= 200')
     )
```

Reshaping Data – Change the layout of a data set

`pd.melt(df)`
Gather columns into rows.

`df.pivot(columns='var', values='val')`
Spread rows into columns.

`pd.concat([df1,df2])`
Append rows of DataFrames

`pd.concat([df1,df2], axis=1)`
Append columns of DataFrames

`df.sort_values('mpg')`
Order rows by values of a column (low to high).

`df.sort_values('mpg',ascending=False)`
Order rows by values of a column (high to low).

`df.rename(columns = {'y':'year'})`
Rename the columns of a DataFrame

`df.sort_index()`
Sort the index of a DataFrame

`df.reset_index()`
Reset index of DataFrame to row numbers, moving index to columns.

`df.drop(columns=['Length','Height'])`
Drop columns from DataFrame

Subset Observations (Rows)

`df[df.Length > 7]`
Extract rows that meet logical criteria.

`df.drop_duplicates()`
Remove duplicate rows (only considers columns).

`df.head(n)`
Select first n rows.

`df.tail(n)`
Select last n rows.

`df.sample(frac=0.5)`
Randomly select fraction of rows.

`df.sample(n=10)`
Randomly select n rows.

`df.iloc[10:20]`
Select rows by position.

`df.nlargest(n, 'value')`
Select and order top n entries.

`df.nsmallest(n, 'value')`
Select and order bottom n entries.

Logic in Python (and pandas)

<	Less than	!=	Not equal to
>	Greater than	df.column.isin(*values*)	Group membership
==	Equals	pd.isnull(*obj*)	Is NaN
<=	Less than or equals	pd.notnull(*obj*)	Is not NaN
>=	Greater than or equals	&,|,~,^,df.any(),df.all()	Logical and, or, not, xor, any, all

Subset Variables (Columns)

`df[['width','length','species']]`
Select multiple columns with specific names.

`df['width']` *or* `df.width`
Select single column with specific name.

`df.filter(regex='regex')`
Select columns whose name matches regular expression *regex*.

regex (Regular Expressions) Examples

'\.'	Matches strings containing a period '.'
'Length$'	Matches strings ending with word 'Length'
'^Sepal'	Matches strings beginning with the word 'Sepal'
'^x[1-5]$'	Matches strings beginning with 'x' and ending with 1,2,3,4,5
'^(?!Species$).*'	Matches strings except the string 'Species'

`df.loc[:,'x2':'x4']`
Select all columns between x2 and x4 (inclusive).

`df.iloc[:,[1,2,5]]`
Select columns in positions 1, 2 and 5 (first column is 0).

`df.loc[df['a'] > 10, ['a','c']]`
Select rows meeting logical condition, and only the specific columns .

http://pandas.pydata.org/ This cheat sheet inspired by Rstudio Data Wrangling Cheatsheet (https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf) Written by Irv Lustig, Princeton Consultants

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf