Data Science

Unit 1-04: Introduction to Pandas

AKADEMI GA

GENERAL ASSEMBLY

Week 1: Data Foundations

- *Now that we have warmed up with Python lists and understand some of the statistics for describing our data, we are ready to play with more data!*
- *Today we will delve into using Pandas which is the most useful library for data wrangling and exploration.*

Week 1 Units
1-01 Installation and Github
1-02 Python Review and Practice
1-03 List Comprehension
1-04 Introduction to Pandas
1-05 Data Wrangling

Our Learning Goals

In this lesson, we will learn how to:

- Define the anatomy of DataFrames
- Explore data with Pandas
- Filter data with Pandas

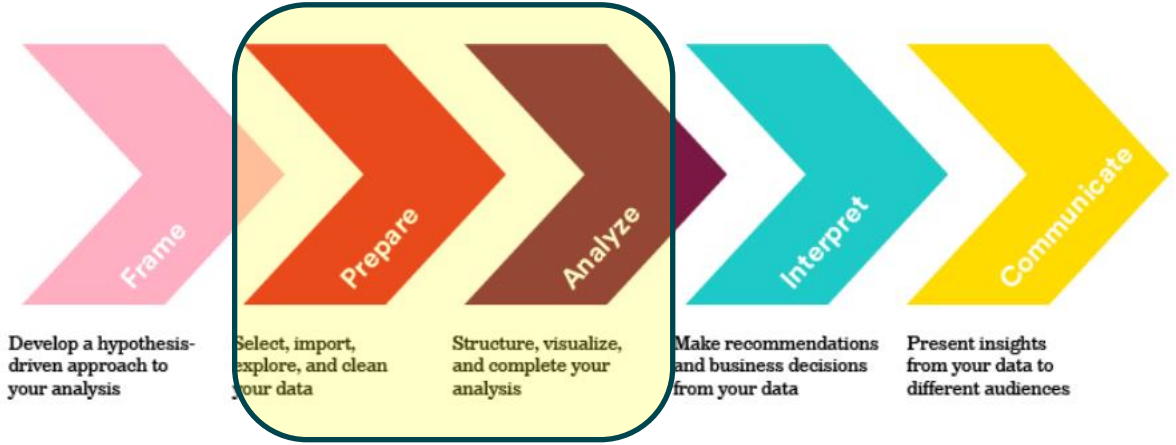
Unit 1-04 Introduction to Pandas

Lesson 1: Pandas DataFrames

AKADEMI GA

GENERAL ASSEMBLY

DATA SCIENCE WORKFLOW



Pandas is an essential Python library for managing our data in the Data Science Workflow.

We can use Pandas to:
- load the data into our Python notebooks
- explore and visualize the data
- clean and transform the data
- filter and sort data

WHAT ARE THE STEPS IN A DATA SCIENCE PROJECT?

Step 0.

- Be as lazy as possible
- Try to find pre-packaged software

This is why we are using python and python data science libraries.

What is Pandas?

A. A group of adorable bears 🐼🐼🐼

B. A Python library for data manipulation

Pandas (Python Data Analysis Library)

- Reads Excel, CSV, databases, HDF5
- Efficient storage and retrieval
- Some SQL-like selection, grouping tools
- Summarises data
- Interpolates time series data

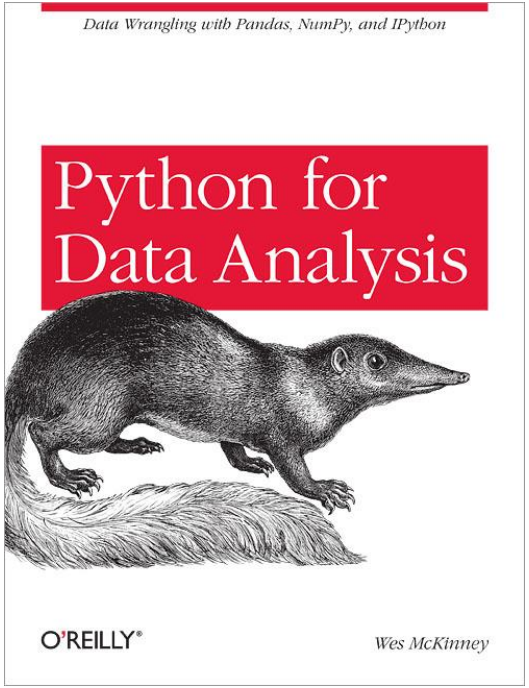
```
pip install pandas
```

or

```
apt-get install python-pandas
```

or

use anaconda



So, Pandas the Library

Quick Backstory from 2009:

- A humble open source project for **Panel Data** (hence “Pandas”) from Wes McKinney.
- A ‘panel’ is the name of the object (in pandas) holding an n-dimensional array
- A 2-dimensional panel is a Dataframe (rows and columns)
- A 1-dimensional panel is a Series (column)

order_id	order_info_id	order_id_number	return_rea	order_date	order_weekday	order_month
AE-2019-1231682	AE-2019	1231682		18/12/2019	Wednesday	December
AE-2019-1263608	AE-2019	1263608		26/12/2019	Thursday	December
AE-2019-303016	AE-2019	303016		18/12/2019	Wednesday	December
AE-2019-304471	AE-2019	304471		27/12/2019	Friday	December
AE-2019-3123605	AE-2019	3123605		30/11/2019	Saturday	November
AE-2019-3179243	AE-2019	3179243		18/11/2019	Monday	November
AE-2019-3323423	AE-2019	3323423		06/12/2019	Friday	December
AE-2019-3371820	AE-2019	3371820		26/12/2019	Thursday	December
AE-2019-350473	AE-2019	350473		25/03/2019	Monday	March
AE-2019-4073208	AE-2019	4073208		13/01/2019	Sunday	January
AE-2019-4135578	AE-2019	4135578		22/03/2019	Friday	March
AE-2019-4220600	AE-2019	4220600		24/03/2019	Sunday	March
AE-2019-4244209	AE-2019	4244209		18/12/2019	Wednesday	December
AE-2019-447833	AE-2019	447833		26/12/2019	Thursday	December

A Dataframe in Pandas contains rows and columns of data, just like in Excel

Exploratory Data Analysis (EDA)

The process of understanding our dataset and producing our first level of insights.

This includes:

- Reading in data
- Checking data types
- Find a summary of the data
- Viewing the distribution
- And more!

Today, we will focus on the most ‘mission critical’ elements of EDA.

Series

- **Have an *index***
  - Automatically created with [0,1,2…] unless index=… is used
  - The index doesn't have to be numeric
- **Can be retrieved by index, index ranges**
- **Can be added, multiplied, subtracted**
  - Done by index
  - Copes with missing data
- **Can be compared against numbers**
  - Returns a shorter Series
  - Which can be used as an index, or bool'd with & and |
- **Interesting functions and Attributes:**
  - mean(), median(), max(), tshift(), describe(), dtype
  - idxmin(), idxmax(), value_counts(), isnull(), notnull()

Example Code

In [1]:
```
1 import pandas as pd
2 kl_temp = pd.Series(index=['Mon','Tue','Wed'], data =[36, 34, 37])
3 pg_temp = pd.Series(index=['Sun','Mon','Tue'], data =[31, 33, 32])
```

In [2]:
```
1 kl_temp - pg_temp
```

Add series together

Out[2]:
```
Mon    3.0
Sun    NaN
Tue    2.0
Wed    NaN
dtype: float64
```

In [3]:
```
1 kl_temp + 1
```

Add a number to a series

Out[3]:
```
Mon    37
Tue    35
Wed    38
dtype: int64
```

Example Code

In [1]:
```
1 import pandas as pd
2 kl_temp = pd.Series(index=['Mon','Tue','Wed'], data =[36, 34, 37])
3 pg_temp = pd.Series(index=['Sun','Mon','Tue'], data =[31, 33, 32])
```

In [2]:
```
1 kl_temp - pg_temp
```

Add series together

Out[2]:
```
Mon    3.0
Sun    NaN
Tue    2.0
Wed    NaN
dtype: float64
```

In [3]:
```
1 kl_temp + 1
```

Add a number to a series

Out[3]:
```
Mon    37
Tue    35
Wed    38
dtype: int64
```

More Pandas Series Examples


```
In [5]:  1  kl_temp.min()
Out[5]: 34
```

```
In [6]:  1  kl_temp.idxmin()
Out[6]: 'Tue'
```

Which index has the minimum number?

```
In [7]:  1  (kl_temp-pg_temp).max()
Out[7]: 3.0
```

Adding two series results in a series

```
In [8]:  1  kl_temp > 35
Out[8]: Mon     True
        Tue    False
        Wed     True
        dtype: bool
```

When you compare, you get a list of Booleans

```
In [9]:  1  kl_temp[kl_temp > 35]
Out[9]: Mon    36
        Wed    37
        dtype: int64
```

If you index by a list of Booleans, you get just the True ones

DataFrames

- **Often created by reading from CSV / Excel with pandas.read_csv()**
  - See also read_table(), read_excel()…
- **Sometimes from a dictionary of Series**
- **Have an *index***
  - Autogenerated from underlying Series if not specified
  - Drops other data if it is specified

DataFrame from Series

In [1]:
```
1 import pandas as pd
2 kl_temp = pd.Series(index=['Mon','Tue','Wed'], data =[36, 34, 37])
3 pg_temp = pd.Series(index=['Sun','Mon','Tue'], data =[31, 33, 32])
```

In [11]:
```
1 city_data = pd.DataFrame({'kl':kl_temp,
2                           'penang':pg_temp})
```

In [12]:
```
1 city_data
```

Out[12]:

	kl	penang
Mon	36.0	33.0
Sun	NaN	31.0
Tue	34.0	32.0
Wed	37.0	NaN

Create a dataframe with a dictionary of Series

DataFrame from CSV file

Create a dataframe by reading in a CSV (comma-separated-values) file

```
City,Colors Reported,Shape Reported,State,Time
Ithaca,,TRIANGLE,NY,6/1/1930 22:00
Willingboro,,OTHER,NJ,6/30/1930 20:00
Holyoke,,OVAL,CO,2/15/1931 14:00
Abilene,,DISK,KS,6/1/1931 13:00
New York Worlds Fair,,LIGHT,NY,4/18/1933 19:00
Valley City,,DISK,ND,9/15/1934 15:30
Crater Lake,,CIRCLE,CA,6/15/1935 0:00
Alma,,DISK,MI,7/15/1936 0:00
Eklutna,,CIGAR,AK,10/15/1936 17:00
Hubbard,,CYLINDER,OR,6/15/1937 0:00
Fontana,,LIGHT,CA,8/15/1937 21:00
Waterloo,,FIREBALL,AL,6/1/1939 20:00
Belton,RED,SPHERE,SC,6/30/1939 20:00
Keokuk,,OVAL,IA,7/7/1939 2:00
Ludington,,DISK,MI,6/1/1941 13:00
```

In [13]:
```
1 ufo = pd.read_csv('datasets/ufo.csv')
```

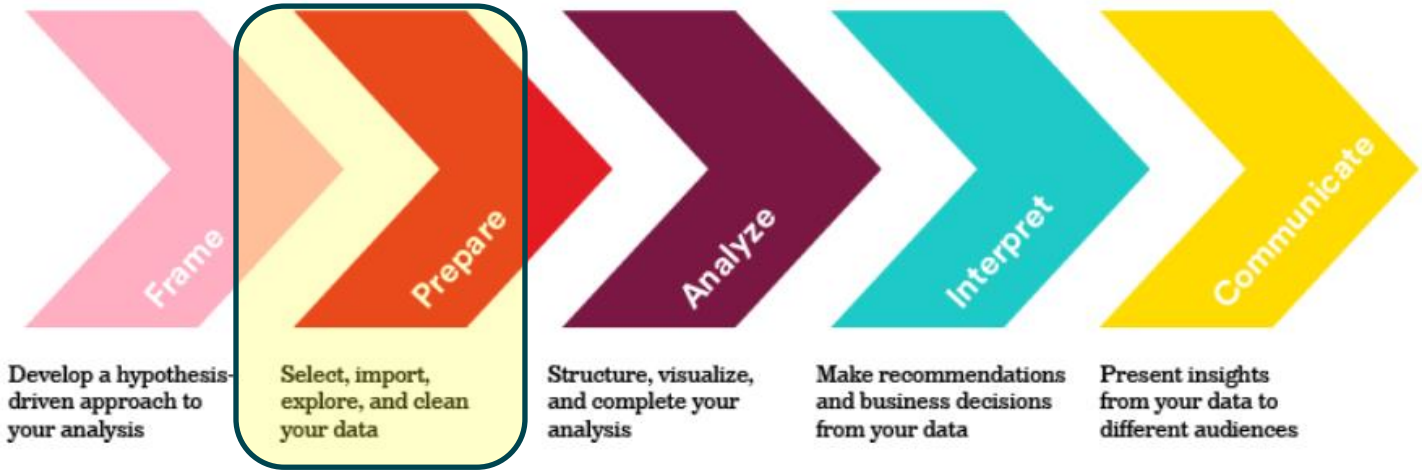
In [14]:
```
1 ufo.head()
```

Out[14]:

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00

Importing Data

- Let's start by selecting, importing and exploring our data.



Notebooks

- Unit 1-04 Lesson 1: Pandas DataFrames
  - Pandas and DataFrames
  - Intro-to-Pandas

Q&A

Unit 1-04 Introduction to Pandas

Lesson 2: Exploratory Data Analysis with Pandas

AKADEMI GA

GENERAL ASSEMBLY

Notebooks

- Lesson 2: Intro to Pandas Lab

Q&A

Homework

- Complete the Pandas lab

Recap

In this unit, we:

- Learned about Pandas Series and DataFrames
- Performed some exploratory data analysis on our dataset.

Looking Ahead

Homework : Intro to Pandas Lab

Up Next: Data Wrangling with Pandas