



Unit - 1

Data Manipulation with Pandas



Objective

- Data Manipulation with Pandas
- Introduction to Pandas Objects
- Pandas Series Object
- Pandas DataFrame Object
- Data Storage Format
- Reading data from file
- Load CSV file to Pandas
- Groupby Function
- Pandas Data visualization

Data Manipulation with pandas

- *Pandas* is a fast, powerful, flexible and easy to use open-source data analysis and manipulation tool.
- Pandas is a newer package built on top of NumPy and provides an efficient implementation of a **DataFrame**.

Series			Series			DataFrame	
	apples			oranges			
0	3	+	0	0	=	0	3
1	2		1	3		1	2
2	0		2	7		2	0
3	1		3	2		3	1
							oranges

Introducing Pandas Objects

- Pandas objects can be thought of as enhanced versions of NumPy structured arrays in which the rows and columns are identified with labels rather than simple integer indices
- There are three fundamental Pandas data structures:
 - Series
 - DataFrame
 - Index.

Series			Series			DataFrame		
	apples			oranges			apples	oranges
0	3	+	0	0	=	0	3	0
1	2		1	3		1	2	3
2	0		2	7		2	0	7
3	1		3	2		3	1	2

The Pandas Series Object

- A Pandas Series is a one-dimensional array of indexed data.
- The Series wraps both a sequence of values and a sequence of indices, which we can access with the values and index attributes.

```
pd.Series(['p', 'q', 'r', 's'], index=[3, 2, 4, 5])
```

↓
sort_index()

Data	
2	q
3	p
4	r
5	s

dtype: object

Series

The Pandas DataFrame Object

- The next fundamental structure in Pandas is the DataFrame.
- The DataFrame can be thought of either as a generalization of a NumPy array, or as a specialization of a Python dictionary.
- *DataFrame as a generalized NumPy array*
- If a Series is an analog of a one-dimensional array with flexible indices, a DataFrame is an analog of a two-dimensional array with both flexible row indices and flexible column names.

Creating Series from simple datatypes

Creating a Pandas Series

- The Pandas Series can be defined as a one-dimensional array that can store various data types. We can easily convert the list, tuple, and dictionary into series using "series" method.
- The row labels of series are called the index. A Series cannot contain multiple columns. It has the following parameter:
 - data
 - index
 - dtype
 - copy

Data Storage Formats in Pandas

- The different data storage formats available to be manipulated by Pandas library are text, binary and SQL.
- Below is a table containing available 'readers' and 'writers' functions of the pandas I/O API set with data format and description

Format Type	Data Description	Reader	Writer
text	CSV	<code>read_csv</code>	<code>to_csv</code>
text	Fixed-Width Text File	<code>read_fwf</code>	
text	JSON	<code>read_json</code>	<code>to_json</code>
text	HTML	<code>read_html</code>	<code>to_html</code>
text	Local clipboard	<code>read_clipboard</code>	<code>to_clipboard</code>
binary	MS Excel	<code>read_excel</code>	<code>to_excel</code>
binary	OpenDocument	<code>read_excel</code>	
binary	HDF5 Format	<code>read_hdf</code>	<code>to_hdf</code>
binary	Feather Format	<code>read_feather</code>	<code>to_feather</code>
binary	Parquet Format	<code>read_parquet</code>	<code>to_parquet</code>
binary	ORC Format	<code>read_orc</code>	
binary	Msgpack	<code>read_msgpack</code>	<code>to_msgpack</code>
binary	Stata	<code>read_stata</code>	<code>to_stata</code>
binary	SAS	<code>read_sas</code>	
binary	SPSS	<code>read_spss</code>	
binary	Python Pickle Format	<code>read_pickle</code>	<code>to_pickle</code>
SQL	SQL	<code>read_sql</code>	<code>to_sql</code>
SQL	Google BigQuery	<code>read_gbq</code>	<code>to_gbq</code>

CSV file and JSON file

What is CSV file?

- A CSV is a comma-separated values file, which allows data to be saved in a tabular format.
- CSV files can be used with many spreadsheets program, such as Microsoft Excel or Google Spreadsheets.
- They differ from other spreadsheet file types because you can only have a single sheet in a file, they cannot save cell, column, or row. Also, you cannot save formulas in this format.

JSON (JavaScript Object Notation)	vs	CSV (Comma Separated Values)
File size		
Larger file size		Compact file size
Hierarchy		
Supports hierarchical and relational data		Errors when displaying hierarchical data
Scalability		
Allows scalability and integrates with APIs easily		Not easily scalable and difficult to integrate
Best for		
Works best for complex and large-scale datasets		Convenient for small datasets

Why are .CSV files used?

- CSV files are plain-text files, making them easier for the website developer to create
- Since they're plain text, they're easier to import into a spreadsheet or another storage database, regardless of the specific software you're using.
- To better organize large amounts of data.

How do I save CSV files?

Under the "File name" section in the "Save As" tab, you can select "Save as type" and change it to "CSV (Comma delimited) (*.csv)".

What is a JSON file?

- A JSON file is a file that stores simple data structures and objects in JavaScript Object Notation (JSON) format, which is a standard data interchange format.

Structures of JSON

- JSON supports two widely used (amongst programming languages) data structures.
 - A collection of name/value pairs.
 - An ordered list of values.



```
Employee - Notepad
File Edit Format View Help
[
  {
    "Name": "John Sins",
    "Gender": "Male",
    "Country": "United States",
    "Age": "21"
  },
  {
    "Name": "Mark Paul",
    "Gender": "Male",
    "Country": "United Kindom",
    "Age": "24"
  },
  {
    "Name": "Martina",
    "Gender": "Female",
    "Country": "Rassia",
    "Age": "24"
  }
]
Ln 20, Col 3 100% Windows
```

Reading data from files



Reference: <https://realpython.com/pandas-read-write-files>

Load CSV files to Python Pandas

- The basic process of loading data from a CSV file into a Pandas DataFrame is achieved using the “read_csv” function in Pandas.
- `pd.read_csv(path/xyz.csv)`

```
In [18]: pd.read_csv("../pokemon.csv", header=[6,3,5,7], squeeze = True,
```

```
Out[18]:
```

	Charizard	Fire
	Venusaur	Grass
	Charmeleon	Fire
	Squirtle	Water
0	Wartortle	Water
1	Blastoise	Water
2	Caterpie	Bug
3	Metapod	Bug

Delimiters in Text Fields – Quote char

- The quote character can be specified in Pandas `read_csv` using the `quotechar` argument.

Semi-colon separated data in text file

```
CustomerId; CustomerName; Address; Age; NickNames
1;Shane Lynn;Dublin, Ireland; 30;"Shaneo;Lynno;Slynn"
2;Johnny Ives;London, United Kingdom;40;"Johnson;Big John;Ivy"
3;Simon Smith;Rue de Rue, Paris, France;50;"Frenchy;Smitho;Hammer"
4;Ronald Mc Donald;The big Farm, McDonalds Farm; 60;"Ronnie;Maccie;Donnie"
5;Jonathan Swift;Celbridge Abbey, Celbridge, Ireland;70;"Jonno;Speedy;Swifter"
```

The data in the column contains semicolons, so quotation char is used to quote the values

Semi-colon separated data loaded into Excel

	A	B	C	D	E
1	CustomerId	CustomerName	Address	Age	NickNames
2	1	Shane Lynn	Dublin, Ireland	30	Shaneo;Lynno;Slynn
3	2	Johnny Ives	London, United Kingdom	40	Johnson;Big John;Ivy
4	3	Simon Smith	Rue de Rue, Paris, France	50	Frenchy;Smitho;Hammer
5	4	Ronald Mc Donald	The big Farm, McDonalds Farm	60	Ronnie;Maccie;Donnie
6	5	Jonathan Swift	Celbridge Abbey, Celbridge, Ireland	70	Jonno;Speedy;Swifter

Semicolons (;) are used here to separate columns

Semi-colon separated data loaded to Pandas

```
pd.read_csv('test_delimited.csv', sep=';', quotechar='"', encoding='utf8')
```

The 'sep' argument tells Pandas how to break up data into columns

Specify the quotechar if necessary - the default is "

	CustomerId	CustomerName	Address	Age	NickNames
0	1	Shane Lynn	Dublin, Ireland	30	Shaneo;Lynno;Slynn
1	2	Johnny Ives	London, United Kingdom	40	Johnson;Big John;Ivy
2	3	Simon Smith	Rue de Rue, Paris, France	50	Frenchy;Smitho;Hammer
3	4	Ronald Mc Donald	The big Farm, McDonalds Farm	60	Ronnie;Maccie;Donnie
4	5	Jonathan Swift	Celbridge Abbey, Celbridge, Ireland	70	Jonno;Speedy;Swifter

Reference: <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files>

Python – Paths, Folders, Files

- When you specify a filename to `Pandas.read_csv`, Python will look in your “current working directory”.

```
In [26]: pd.read_csv('file_not_in_right_place.csv')

FileNotFoundError                                Traceback (most recent call last)
<ipython-input-26-f3409a34b3ff> in <module>()
----> 1 pd.read_csv('file_not_in_right_place.csv')

~/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in parser_f(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, iterator, chunksize, compression, thousands, decimal, li
neterminator, quotechar, quoting, escapechar, comment, encoding, dialect, tupleize_cols, error_bad_lines, warn_bad_li
nes, skipfooter, doublequote, delim_whitespace, low_memory, memory_map, float_precision)
    676         skip_blank_lines=skip_blank_lines)
    677
--> 678     return _read(filepath_or_buffer, kwds)
    679
    680     parser_f.__name__ = name

~/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in _read(filepath_or_buffer, kwds)
    438
    439     # Create the parser.
--> 440     parser = TextFileReader(filepath_or_buffer, **kwds)
    441
    442     if chunksize or iterator:

~/Envs/analysis/lib/python3.6/site-packages/pandas/io/parsers.py in __init__(self, f, engine, **kwds)
    785         self.options['has_index_names'] = kwds['has_index_names']
    786
--> 787         self._make_engine(self.engine)
    788
    789     def close(self):
```

Reference: <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files>



Pandas CSV File Loading Errors

- FileNotFoundError
- UnicodeDecodeError
- pandas.parser.CParserError

Load JSON files to Python Pandas

Product	Price
Desktop Computer	700
Tablet	250
iPhone	800
Laptop	1200

Step 1: Prepare the JSON String



```

{"Product":{"0":"Desktop Computer","1":"Tablet","2":"iPhone","3":"Laptop"}}
    
```

Step 2: Create the JSON File.

	Product	Price
0	Desktop Computer	700
1	Tablet	250
2	iPhone	800
3	Laptop	1200

Finally, load your JSON file into Pandas DataFrame.



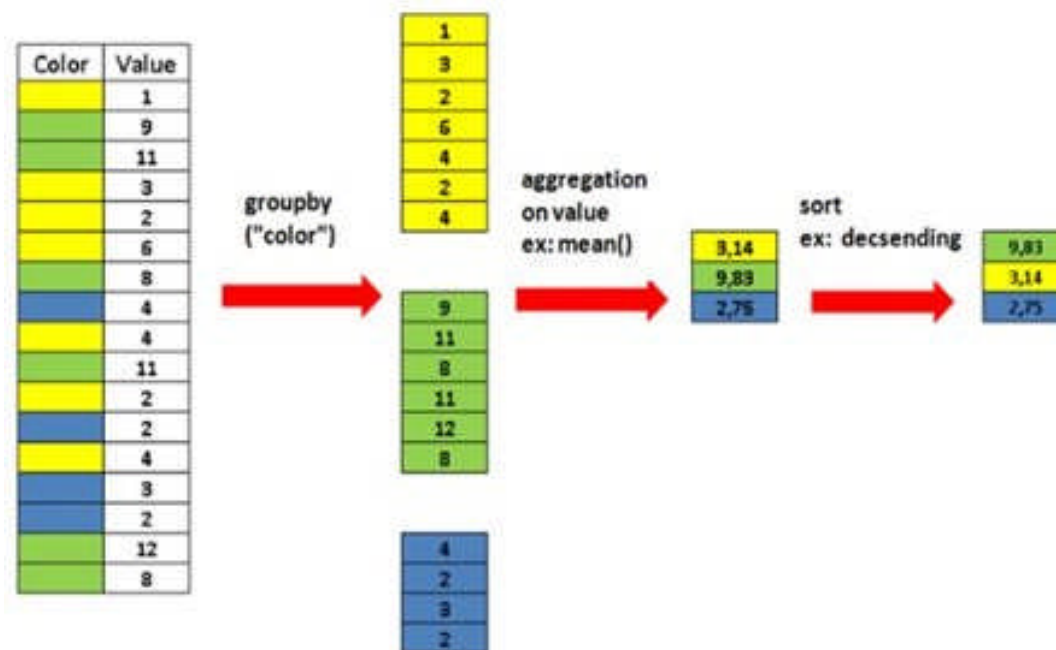
3 different JSON strings

Below are 3 different ways that you could capture the data as JSON strings.

- Index orientation
- Values orientation
- Column's orientation

Groupby Methods

- Pandas `dataframe.groupby()` function is used to split the data into groups based on some criteria. pandas objects can be split on any of their axes.



Reference: <https://towardsdatascience.com/pandas-groupby-explained-453692519d0>

Groupby output format – Series or DataFrame?

- As a rule of thumb, if you calculate more than one column of results, your result will be a Data frame.
- For a single column of results, the agg function, by default, will produce a Series.

```
In [35]: data.groupby('month', as_index=False).agg({"duration": "sum"})
```

```
Out[35]:
```

	month	duration
0	2014-11	26639.441
1	2014-12	14641.870
2	2015-01	18223.299
3	2015-02	15522.299
4	2015-03	22750.441

Pivot Tables

- It's a table of statistics that helps summarize the data of a larger table by "pivoting" that data.

Pivot

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

➔

```
df.pivot(index='foo',
          columns='bar',
          values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

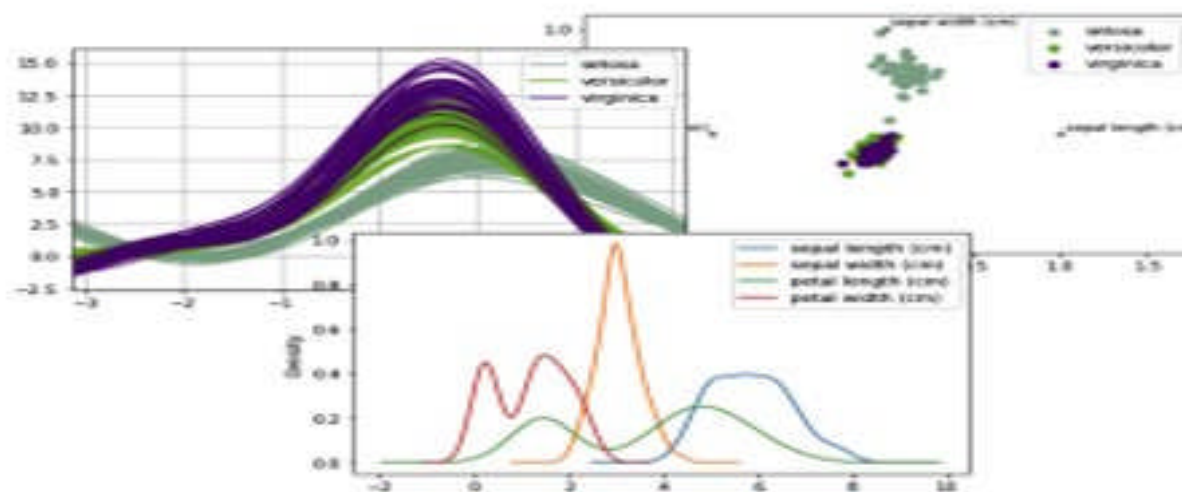
How to Build a Pivot Table in Python

In Pandas, we can construct a pivot table using the following syntax:

```
pandas.pivot_table(data, values=None, index=None, columns=None,  
aggfunc='mean', fill_value=None, margins=False, dropna=True,  
margins_name='All', observed=False)
```

Pandas Plotting

Plotting in pandas utilises the matplotlib API so in order to create visualisations, you will need to also import this library alongside pandas.



<https://towardsdatascience.com/the-best-pandas-plotting-features-c9789e04a5a0>

Plot a Scatter Diagram using Pandas

- Scatter plots are used to depict a relationship between two variables.

Step 1: Prepare the data

Unemployment_Rate	Stock_Index_Price
6.1	1500
5.8	1520
5.7	1525
5.7	1523
5.8	1515
5.6	1540
5.5	1545
5.3	1560
5.2	1555
5.2	1565

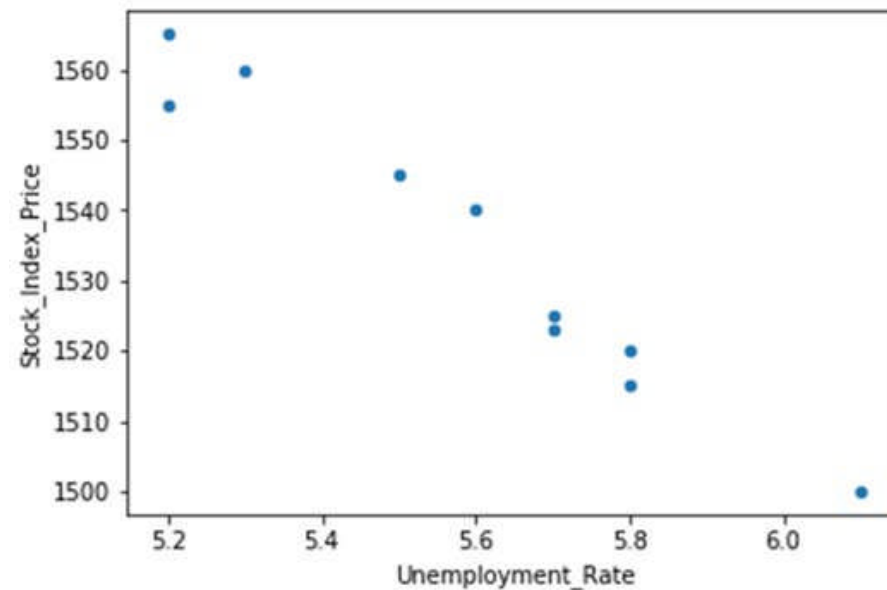
Plot a Scatter Diagram using Pandas

Step 2: Create the DataFrame

	Unemployment_Rate	Stock_Index_Price
0	6.1	1500
1	5.8	1520
2	5.7	1525
3	5.7	1523
4	5.8	1515
5	5.6	1540
6	5.5	1545
7	5.3	1560
8	5.2	1555
9	5.2	1565

Plot a Scatter Diagram using Pandas

Step 3:
Plot the DataFrame using
Pandas



Plot a Line Chart using Pandas

- Line charts are often used to display trends overtime.

Step 1: Prepare the data

Year	Unemployment_Rate
1920	9.8
1930	12
1940	8
1950	7.2
1960	6.9
1970	7
1980	6.5
1990	6.2
2000	5.5
2010	6.3

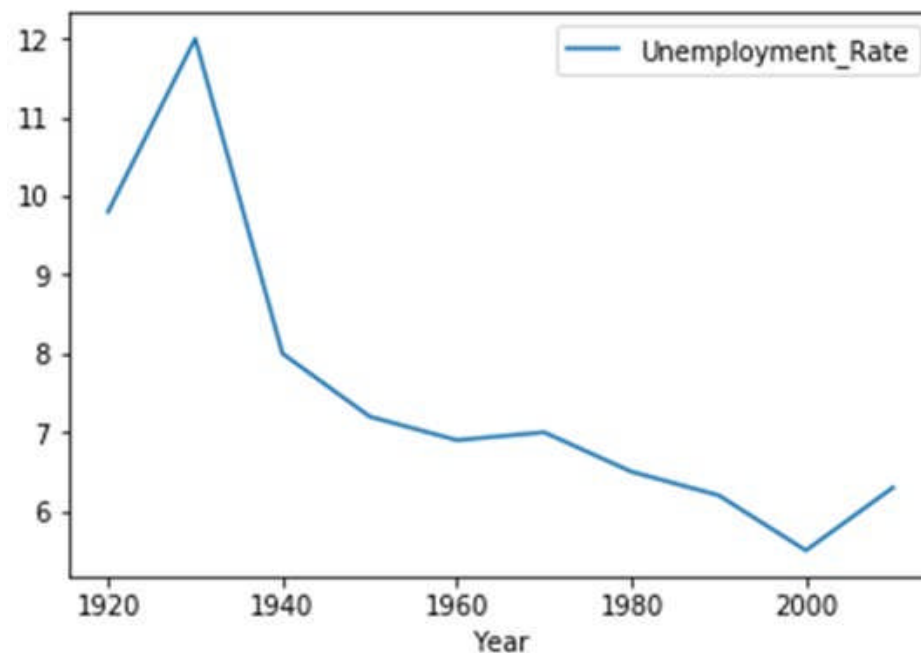
Plot a Line Chart using Pandas

Step 2: Create the DataFrame

	Year	Unemployment_Rate
0	1920	9.8
1	1930	12.0
2	1940	8.0
3	1950	7.2
4	1960	6.9
5	1970	7.0
6	1980	6.5
7	1990	6.2
8	2000	5.5
9	2010	6.3

Plot a Line Chart using Pandas

Step 3:
Plot the DataFrame using
Pandas



Plot a Bar Chart using Pandas

Bar charts are used to display categorical data.

Step 1: Prepare the data

Country	GDP_Per_Capita
USA	45000
Canada	42000
Germany	52000
UK	49000
France	47000

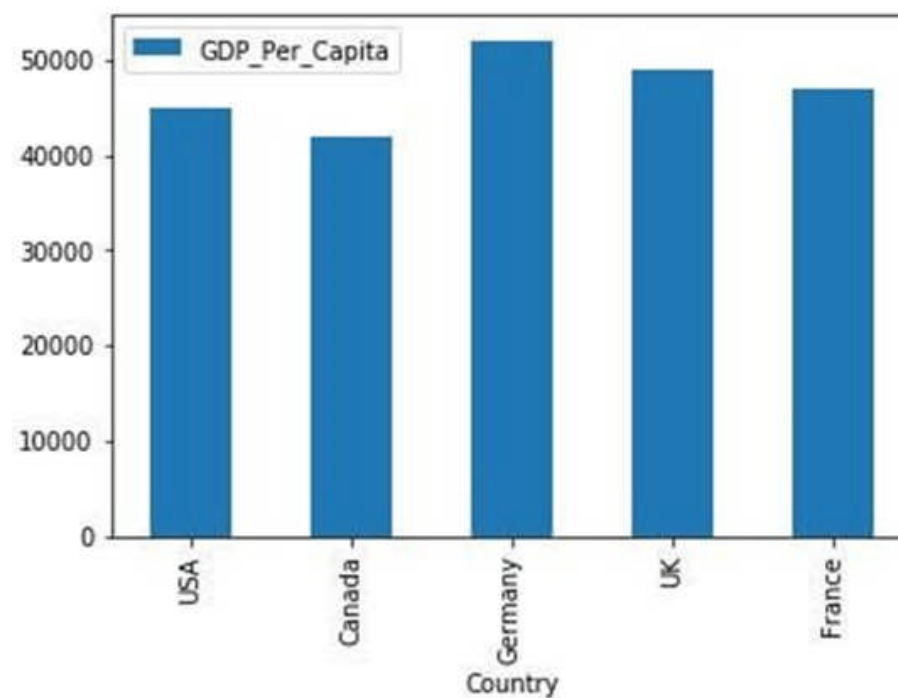
Plot a Bar Chart using Pandas

Step 2: Create the DataFrame

	Country	GDP_Per_Capita
0	USA	45000
1	Canada	42000
2	Germany	52000
3	UK	49000
4	France	47000

Plot a Bar Chart using Pandas

Step 3: Plot the DataFrame using Pandas



Plot a Pie Chart using Pandas

A pie chart (or a circle chart) is a circular statistical graphic, which is divided into slices to illustrate numerical proportion.

Step 1: Prepare the data

Tasks Pending	300
Tasks Ongoing	500
Tasks Completed	700

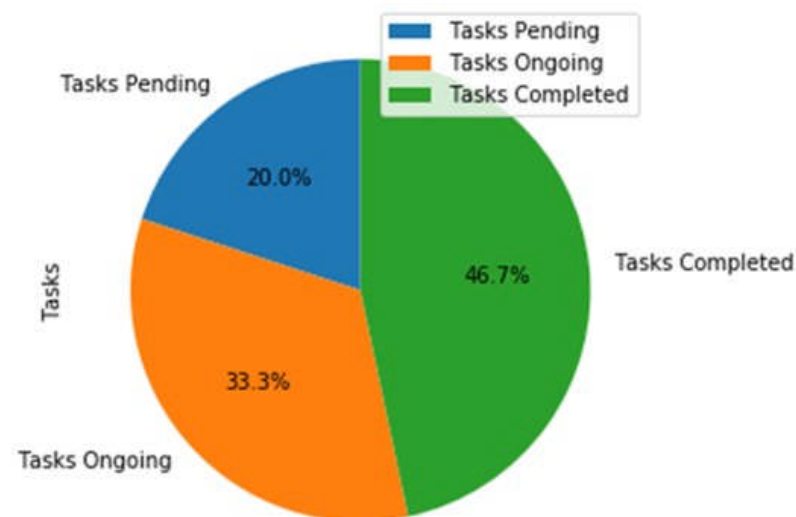
Plot a Pie Chart using Pandas

Step 2: Create the DataFrame

		Tasks
Tasks	Pending	300
Tasks	Ongoing	500
Tasks	Completed	700

Plot a Pie Chart using Pandas

Step 3: Plot the DataFrame using Pandas



References

1. https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html
2. <https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/#:~:text=A%20CSV%20is%20a%20comma,Microsoft%20Excel%20or%20Google%20Spreadsheets.>
3. <https://fileinfo.com/extension/json>
4. <https://www.w3resource.com/JSON/structures.php>
5. <https://www.shanelynn.ie/python-pandas-read-csv-load-data-from-csv-files/>