PANDAS

# X

# **Introduction to Pandas**



Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on the top of the NumPy library. Pandas is fast and it has high-performance & productivity for users.

# **Why use we use Pandas?**

* Calculate statistics and answer questions about the data, like
* What’s the average, median, max, or min of each column?
* Does column A correlate with column B?
* What does the distribution of data in column C look like?
* Clean the data by doing things like removing missing values and filtering rows or columns by some criteria
* Visualize the data with help from Matplotlib. Plot bars, lines, histograms, bubbles, and more.
* Store the cleaned, transformed data back into a CSV, other file or database

# **Applications of Pandas**

**1. Economics**

Economics is in constant demand for data analysis. Analyzing data to form patterns and understanding trends about how the economy in various sectors is growing, is something very essential for economists. Therefore, a lot of economists have started using Python and Pandas to analyze huge datasets. Pandas provide a comprehensive set of tools, like dataframes and file-handling. These tools help immensely in accessing and manipulating data to get the desired results. Through these applications of Pandas, economists all around the world have been able to make breakthroughs like never before.

**2. Recommendation Systems**

We all have used Spotify or Netflix and been appalled at the brilliant recommendations provided by these sites. These systems are a [miracle of Deep Learning](https://data-flair.training/blogs/deep-learning-tutorial/). Such models for providing recommendations is one of the most important applications of Pandas. Mostly, these models are made in python and Pandas being the main libraries of python, used when handling data in such models. We know that Pandas are best for managing huge amounts of data. And the recommendation system is possible only by learning and handling huge masses of data. Functions like groupBy and mapping help tremendously in making these systems possible.

**3. Stock Prediction**

The stock market is extremely volatile. However, that doesn’t mean that it cannot be predicted. With the help of Pandas and a few other libraries like [NumPy](https://data-flair.training/blogs/python-numpy-tutorial/) and matplotlib, we can easily make models which can predict how the stock markets turn out. This is possible because there is a lot of previous data of stocks which tells us about how they behave. And by learning these data of stocks, a model can easily predict the next move to be taken with some accuracy. Not only this, but people can also automate buying and selling of stocks with the help of such prediction models.

**4. Neuroscience**

Understanding the nervous system has always been in the minds of humankind because there are a lot of potential mysteries about our bodies which we haven’t solved as of yet. Machine learning has helped this field immensely with the help of the various applications of Pandas. Again, the data manipulation capabilities of Pandas have played a major role in compiling a huge amount of data which has helped neuroscientists in understanding trends that are followed inside our bodies and the effect of various things on our entire nervous system.

**5. Statistics**

Pure maths itself has made much progress with the various applications of Pandas. Since Statistic deals with a lot of data, a library like Pandas which deals with data handling has helped in a lot of different ways. The functions of mean, median and mode are just very basic ones which help in performing statistical calculations. There are a lot of other complex functions associated with statistics and pandas plays a huge role in these so as to bring perfect results.

# **Setup and installation of pandas**

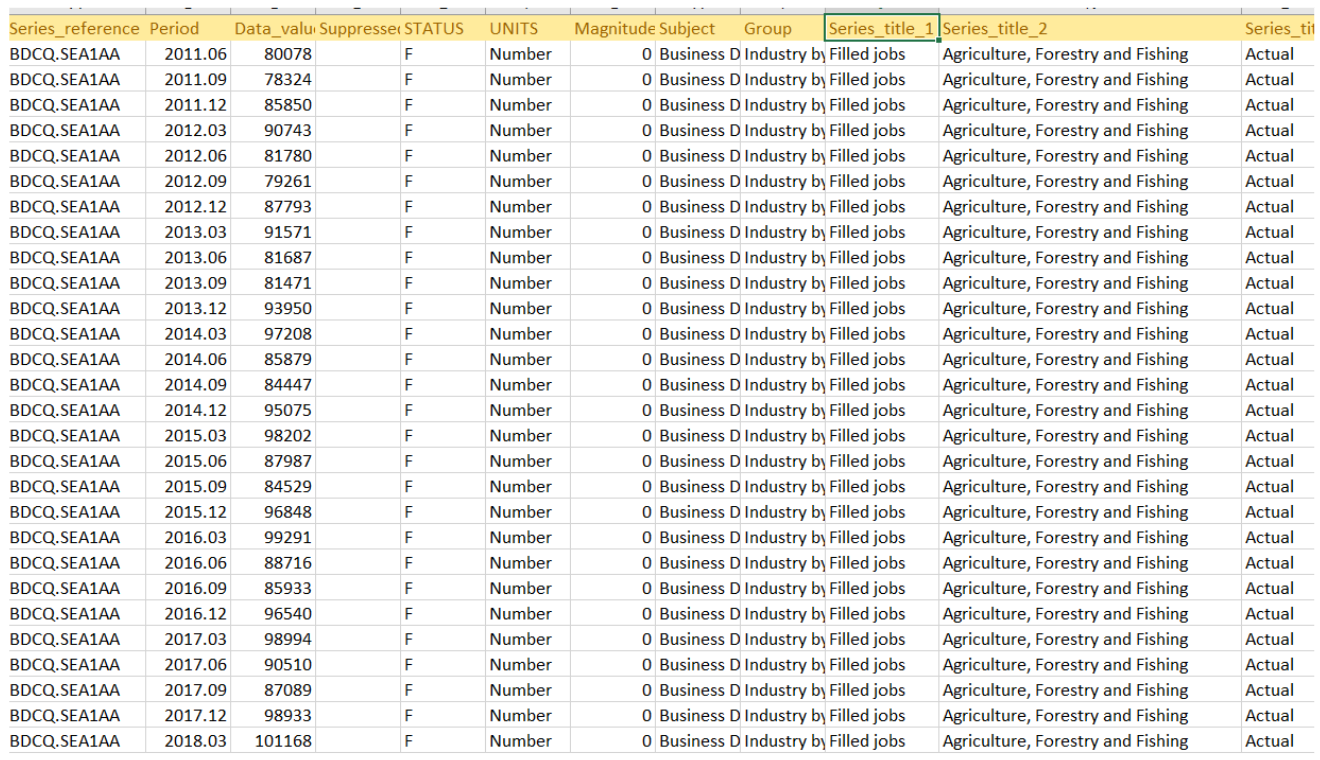
Before we move on with the code for understanding the features of Pandas, let’s get Pandas installed in your system.

## **Install Pandas**

*pip install pandas*

**Sample Data**

Here I use simple Business employment data. It comprises of data values for each employee and filled jobs in specific field of area. Below is the link you can download the CSV file.



* **Load data into Pandas**

With Pandas, we can load data from different sources. Few of them are loading from CSV or a remote URL. The loaded data is stored in a Pandas data structure called DataFrame. DataFrame’s are usually referred by the variable name df . So, anytime you see df from here on you should be associating it with Dataframe.

* **Load data into Pandas**

With Pandas, we can load data from different sources. Few of them are loading from CSV or a remote URL or from a database. The loaded data is stored in a Pandas data structure called DataFrame. DataFrame’s are usually refered by the variable name df . So, anytime you see df from here on you should be associating it with Dataframe.

## **From CSV File**

import pandas  
df = pandas.read\_csv("path\_to\_csv")

## **From Remote URL**

You can pass a remote URL to the CSV file in read\_csv.

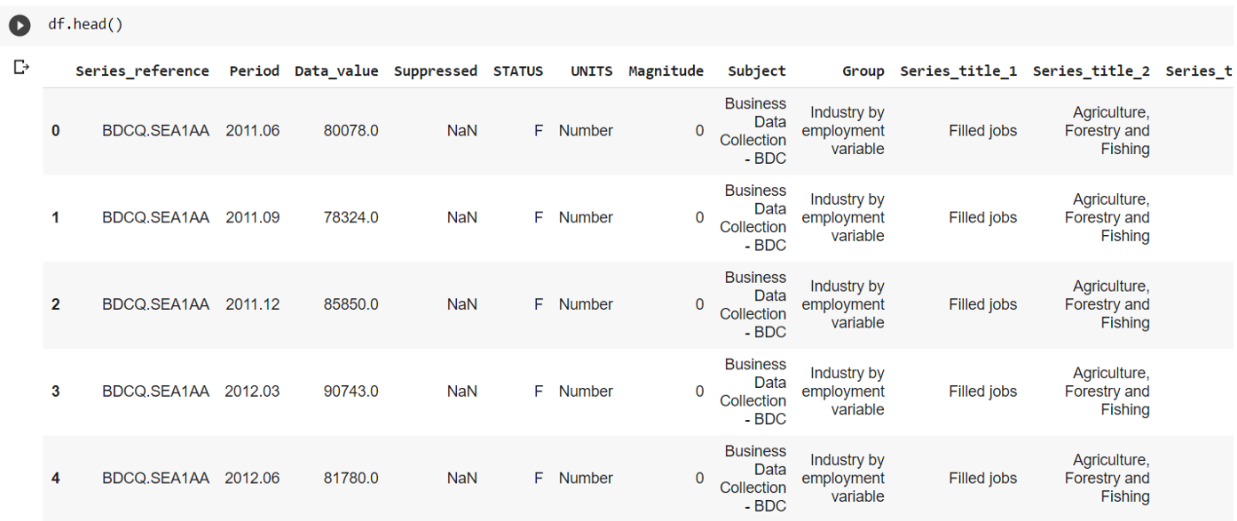
import pandas  
df = pandas.read\_csv("remote/url/path/pointing/to/csv")

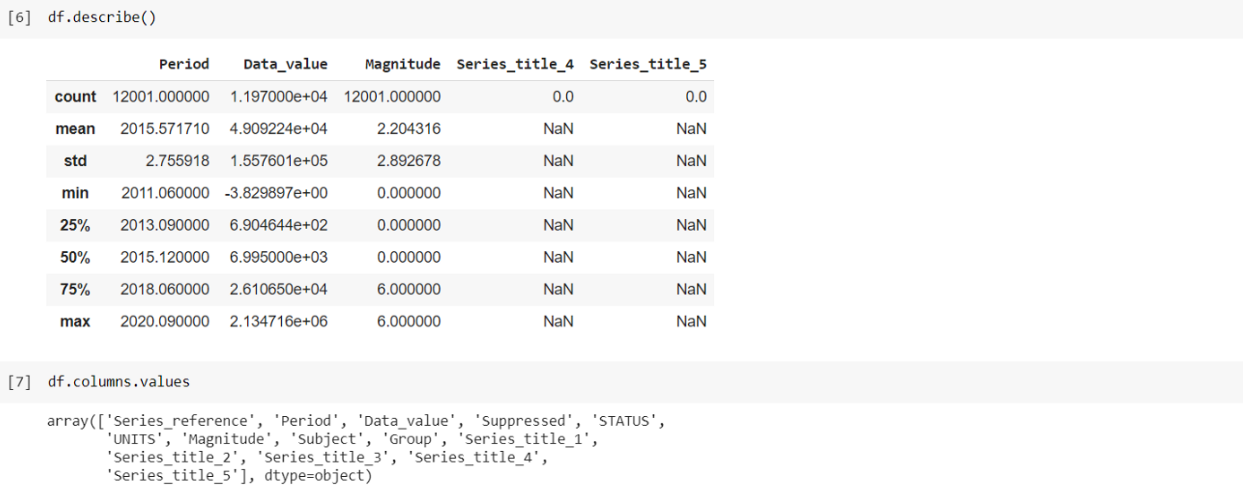
Each of the above snippets reads data from a source and loads it into Pandas’ internal data structure called DataFrame

* **Understanding Data**

Now that we have the Dataframe ready let’s go through it and understand what’s inside it

**# 1. shows you a gist of the data**  
df.head()  
**# 2. Some statistical information about your data**  
df.describe()  
**# 3. List of columns headers**  
df.columns.values



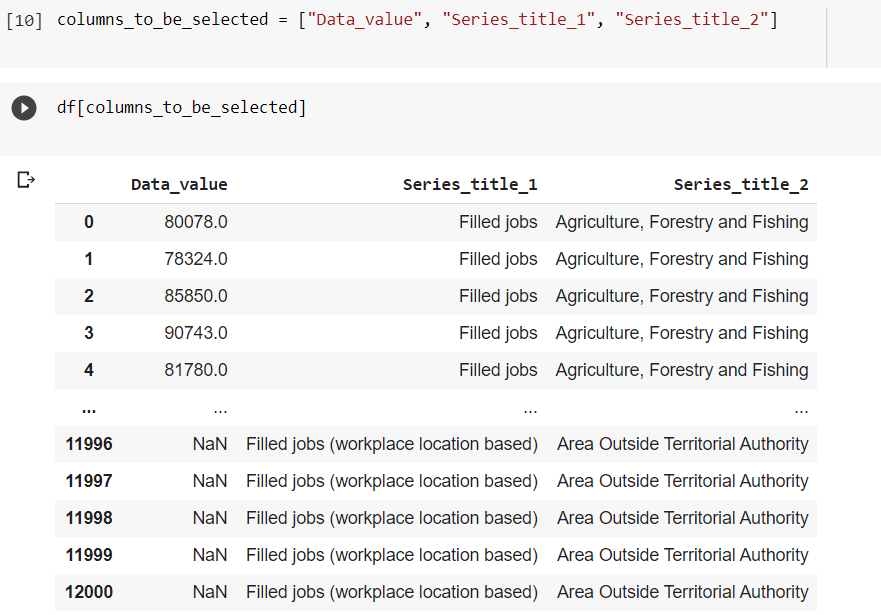


* **Pick & Choose your Data**

Now that we have loaded our data into a DataFrame and understood its structure, let’s pick and choose and perform visualizations on the data.

## **Selecting Columns**

**# 1. Create a list of columns to be selected**  
columns\_to\_be\_selected = ["Data\_value", "Series\_title\_1", "Series\_title\_1"]  
**# 2. Use it as an index to the DataFrame**  
df[columns\_to\_be\_selected]  
**# 3. Using loc method**df.loc[columns\_to\_be\_selected]



## **Selecting Rows**

Unlike the columns, our current DataFrame does not have a label which we can use to refer the row data. But like arrays, DataFrame provides numerical indexing(0, 1, 2…) by default.

**# 1. using numerical indexes - iloc**df.iloc[0:3, :]  
**# 2. using labels as index - loc**row\_index\_to\_select = [0, 1, 4, 5]  
df.loc[row\_index\_to\_select]

* **Grouping**

**Statistical operations**

You can perform statistical operations such as min, max, mean etc., over one or more columns of a Dataframe.

df["Data\_value"].sum()  
df["Data\_value"].mean()  
df["Data\_value"].min()  
df["Data\_value"].max()  
df["Data\_value"].median()  
df["Data\_value"].mode()

Now in a real-world application, the raw use of these statistical functions are rare, often you might want to group data based on specific parameters and derive a gist of the data.

Also there can be multiple fields like data value and we have to perform the operations on both the field. Its quit easy to do so if you have performed the last operation.