What is Pandas?

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

pandas is a data manipulation package in Python for tabular data. That is, data in the form of rows and columns, also known as DataFrames.

pandas’ functionality includes data transformations, like [sorting rows](https://www.datacamp.com/tutorial/pandas-sort-values) and taking subsets, to calculating summary statistics such as the mean, reshaping DataFrames, and joining DataFrames together. pandas works well with other popular Python data science packages, often called the PyData ecosystem, including

* [**NumPy**](https://www.datacamp.com/tutorial/python-numpy-tutorial)for numerical computing
* [**Matplotlib**](https://www.datacamp.com/tutorial/matplotlib-tutorial-python)**,**[**Seaborn**](https://www.datacamp.com/tutorial/seaborn-python-tutorial)**,**[**Plotly**](https://www.datacamp.com/courses/introduction-to-data-visualization-with-plotly-in-python)**,** and other data visualization packages
* [**scikit**-learn](https://www.datacamp.com/tutorial/machine-learning-python) for machine learning

**What is pandas used for?**

pandas is used throughout the data analysis workflow. With pandas, you can:

* Import datasets from databases, spreadsheets, comma-separated values (CSV) files, and more.
* Clean datasets, for example, by dealing with missing values.
* Tidy datasets by reshaping their structure into a suitable format for analysis.
* Aggregate data by calculating summary statistics such as the mean of columns, correlation between them, and more.
* Visualize datasets and uncover insights.

pandas also contains functionality for time series analysis and analyzing text data.

## Key benefits of the pandas package

1. **Data Structures**:
   * **Series**: A one-dimensional array-like object containing a sequence of values. It is similar to a list or a one-dimensional NumPy array but with labeled axes (index).
   * **DataFrame**: A two-dimensional table of data with labeled axes (rows and columns). It can be thought of as a collection of Series objects sharing the same index.
2. **Data Alignment**: Automatic alignment of data for arithmetic operations on Series and DataFrame objects, making it easy to manage missing data.
3. **Data Cleaning**: Functions for handling missing data, such as filling, replacing, and dropping missing values.
4. **Data Transformation**: Tools for reshaping, pivoting, and transforming data, including functions for merging, joining, and concatenating DataFrame objects.
5. **Data Aggregation and Grouping**: Powerful group-by functionality to split data into groups, apply functions to each group, and combine the results.
6. **Time Series Handling**: Specialized tools for working with time series data, including date range generation and frequency conversion.
7. **Input and Output**: Functions for reading and writing data to and from various file formats, including CSV, Excel, SQL databases, and more.
8. **Visualization**: Integration with plotting libraries like Matplotlib to provide basic data visualization capabilities directly from DataFrame and Series objects.

## How to install pandas?

Installing pandas is straightforward; just use the pip install command in your terminal.

pip install pandas

Import Pandas

Once Pandas is installed, import it in your applications by adding the import keyword:

import pandas

import pandas  
  
mydataset = {  
  'cars': ["BMW", "Volvo", "Ford"],  
  'passings': [3, 7, 2]  
}  
  
myvar = pandas.DataFrame(mydataset)  
  
print(myvar)

Pandas as pd

Pandas is usually imported under the pd alias.

**alias:** In Python alias are an alternate name for referring to the same thing.

Create an alias with the as keyword while importing:

import pandas as pd

Now the Pandas package can be referred to as pd instead of pandas.

import pandas as pd  
  
mydataset = {  
  'cars': ["BMW", "Volvo", "Ford"],  
  'passings': [3, 7, 2]  
}  
  
myvar = pd.DataFrame(mydataset)  
  
print(myvar)

## Checking Pandas Version

The version string is stored under \_\_version\_\_ attribute.

import pandas as pd  
  
print(pd.\_\_version\_\_)