

[Home](#)

 **Neelu Tiwari** — Published On June 14, 2021 and Last Modified On March 14th, 2023

[Beginner](#) [Data Cleaning](#) [Programming](#) [Python](#) [Structured Data](#)

50% Off Sale Ends Today | Unlimited Learning, Single Subscription

Introduction

As we know, Data Science is the discipline of study that involves extracting insights from huge amounts of data by the use of various scientific methods, algorithms, and processes. To extract useful knowledge from data, Data Scientists need raw data. This Raw data is a collection of information from various outlined sources and an essential raw material for Data Scientists. It is also known as primary or source data, which is messy and needs cleaning. This beginner's guide will tell you all about data cleaning using pandas in Python.

The primary data consists of irregular and inconsistent values, which lead to many difficulties. When using data, the insights and analysis extracted are only as good as the data we use. Essentially, when irregular data is in, then irregular analysis comes out. Here's where data cleaning comes into play. Data cleansing is an essential part of the data analytics process. Data cleaning removes incorrect, corrupted, garbage, incorrectly formatted, duplicate, or incomplete data within a dataset.

Learning Objectives

- Define data cleaning and its importance in the data analytics process.
- Recognize the importance of accurate, complete, and consistent data for effective analysis and decision-making.
- Learn the various techniques and tools available in the Python Pandas library for data cleaning.

Table of Contents

1. [What Is Data Cleaning?](#)
2. [Why Is Data Cleaning Essential?](#)
3. [Data Cleaning Cycle](#)
4. [Data Cleaning With Pandas](#)
5. [Conclusion](#)
6. [Frequently Asked Questions](#)
 1. [Q1. What do you mean by data type casting in the context of data analysis and data cleaning?](#)
 2. [Q2. When is it appropriate to drop missing values in data rather than imputing them in the context of data cleaning with Pandas?](#)

What Is Data Cleaning?

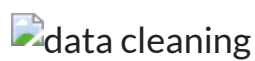
When working with multiple data sources, there are many chances for data to be incorrect, duplicated, or mislabeled. If data is wrong, outcomes and algorithms are unreliable, even though they may look correct. *Data cleaning* is the process of changing or eliminating garbage, incorrect, duplicate, corrupted, or incomplete data in a dataset. There's no such absolute way to describe

Data Cleaning Using Pandas in Python – Complete Guide for Beginners

cleansing, or data scrub is the general data preparation process initiative. Data cleaning plays an important part in developing reliable answers within the analytical process and is observed to be a basic feature of the info science basics. The motive of data cleaning services is to construct uniform and standardized data sets that enable easy access to data analytics tools and business intelligence and perceive accurate data for each problem.

Why Is Data Cleaning Essential?

Data cleaning is the most important task that should be done by a data science professional. Having wrong or bad-quality data can be detrimental to processes and analysis. Having clean data will ultimately increase overall productivity and permit the very best quality information in your decision-making.



Following are some reasons why data cleaning is essential:

1. Error-Free Data: When multiple sources of data are combined, there may be a chance of so much error. Through Data Cleaning, errors can be removed from data. Having clean data which is free from wrong and garbage values can help in performing analysis faster as well as efficiently. By doing this task our considerable amount of time is saved. The results won't be accurate if we use data containing garbage values. When we don't use accurate data, we will surely make mistakes. Monitoring errors and good reporting helps to find where errors are coming from and also makes it easier to fix incorrect or corrupt data for future applications.

2. Data Quality: The quality of the data is the degree to which it follows the rules of particular requirements. For example, if we have imported phone numbers data of different customers, and in some places, we have added email addresses of customers in the data. But because our needs were straightforward for phone numbers, then the email addresses would be invalid data. Here some pieces of data follow a specific format. Some types of numbers have to be in a specific range. Some data cells might require selected quiet data like numeric, Boolean, etc. In every scenario, there are some mandatory constraints our data should follow. Certain conditions affect multiple fields of data in a particular form. Particular types of data have unique restrictions. It will always be invalid if the data isn't in the required format. Data cleaning will help us simplify this process and avoid useless data values.

3. Accurate and Efficient: Ensuring the data is close to the correct values. We know that most of the data in a dataset are valid, and we should focus on establishing its accuracy. Even if the data is authentic and correct, it doesn't mean it is accurate. Determining accuracy helps to figure out whether the data entered is accurate or not. For example, a customer's address is stored in the specified format; maybe it doesn't need to be in the right one. The email has an additional character or value that makes it incorrect or invalid. Another example is the phone number of a customer. This means that we have to rely on data sources to cross-check the data to figure out if it's accurate or not. Depending on the kind of data we are using, we might be able to find various resources that could help us in this regard for cleaning.

4. Complete Data: Completeness is the degree to which we should know all the required values. Completeness is a little more challenging to achieve than accuracy or quality. Because it's nearly impossible to have all the info we need, only known facts can be entered. We can try to complete data by redoing the data-gathering activities like approaching the clients again, re-interviewing people, etc. For example, we might need to enter every customer's contact information. But a number of them

Data Cleaning Using Pandas in Python – Complete Guide for Beginners


all columns, we can try to enter missing or unknown there. But entering such values does not mean that the data is complete. It would still be referred to as incomplete.

5. Maintains Data Consistency: To ensure the data is consistent within the same dataset or across multiple datasets, we can measure consistency by comparing two similar systems. We can also check the data values within the same dataset to see if they are consistent or not. Consistency can be relational. For example, a customer's age might be 25, which is a valid value and also accurate, but it is also stated as a senior citizen in the same system. In such cases, we have to cross-check the data, similar to measuring accuracy, and see which value is true. Is the client a 25-year-old? Or is the client a senior citizen? Only one of these values can be true. There are multiple ways to for your data consistent.

- By checking in different systems.
- By checking the source.
- By checking the latest data.

Data Cleaning Cycle

It is the method of analyzing, distinguishing, and correcting untidy, raw data. Data cleaning involves filling in missing values, handling outliers, and distinguishing and fixing errors present in the dataset. Whereas the techniques used for data cleaning might vary in step with different types of datasets. In this tutorial, we will learn how to clean data using pandas. The following are standard steps to map out data cleaning:

 Data Cleaning Cycle

Data Cleaning With Pandas

Data scientists spend a huge amount of time cleaning datasets and getting them in the form in which they can work. It is an essential skill of Data Scientists to be able to work with messy data, missing values, and inconsistent, noisy, or nonsensical data. To work smoothly, python provides a built-in module, Pandas. Pandas is the popular Python library that is mainly used for data processing purposes like cleaning, manipulation, and analysis. Pandas stand for “Python Data Analysis Library”. It consists of classes to read, process, and write csv files. There are numerous Data cleaning tools present, but the Pandas library provides a really fast and efficient way to manage and explore data. It does that by providing us with Series and DataFrames, which help us represent data efficiently and manipulate it in various ways.

In this article, we will use the Pandas module to clean our dataset.

We are using a simple dataset for data cleaning, i.e., the iris species dataset. You can download this dataset from [kaggle.com](https://www.kaggle.com).

Let's get started with data cleaning step by step.

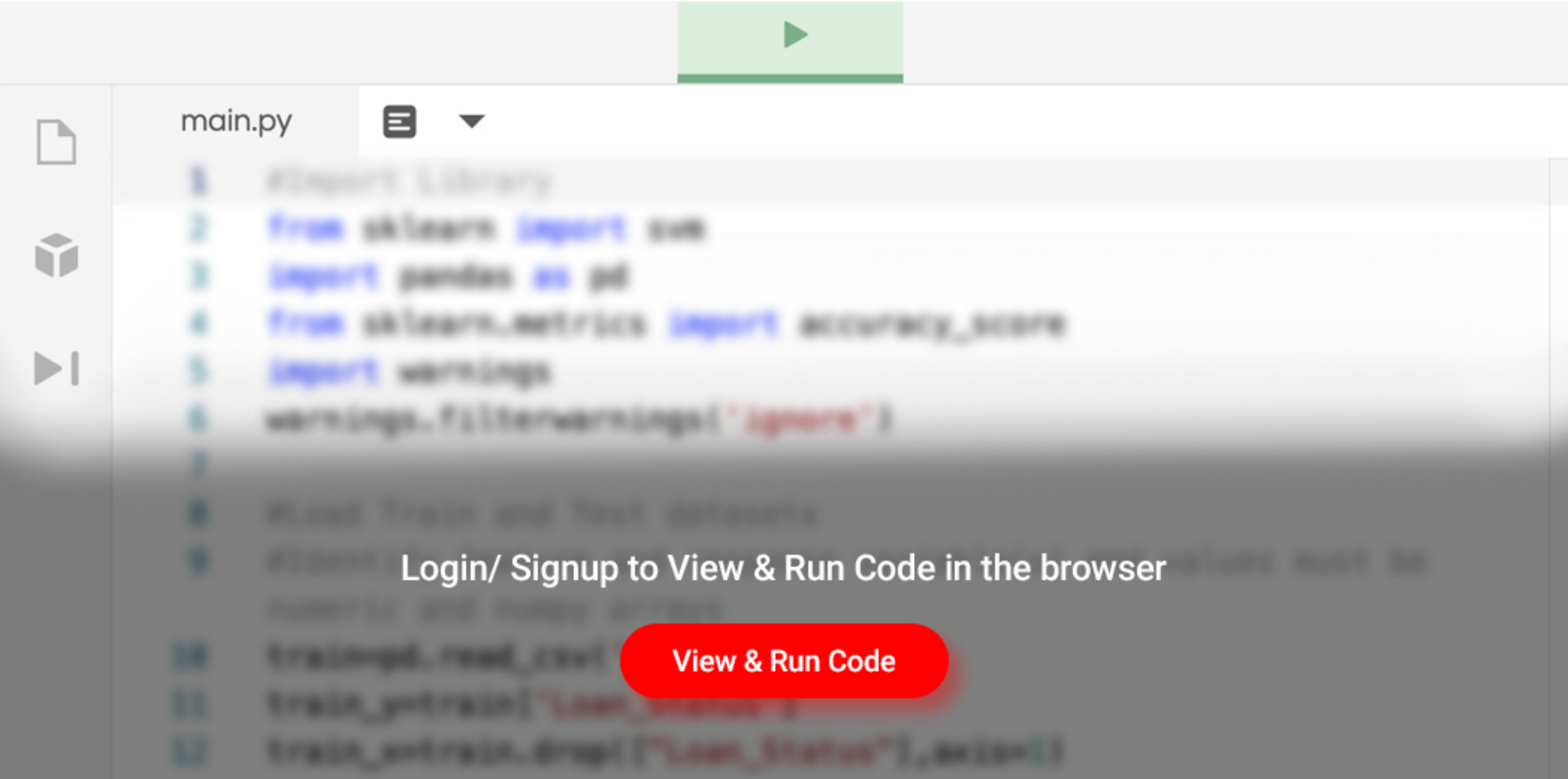
To start working with Pandas, we need to first import it. We are using Google Colab as IDE, so we will import Pandas in Google Colab.

```
#importing module
import pandas as pd
```

Step 1: Import Dataset

To import the dataset, we use the `read_csv()` function of pandas and store it in the pandas DataFrame named as data. As the dataset is in tabular format, when working with tabular data in Pandas, it will be automatically converted into a DataFrame. DataFrame is a two-dimensional, mutable data structure in Python. It is a combination of rows and columns like an excel sheet.

Data Cleaning Using Pandas in Python – Complete Guide for Beginners



The head() function is a built-in function in pandas for the dataframe used to display the rows of the dataset. We can specify the number of rows by giving the number within the parenthesis. By default, it displays the first five rows of the dataset. If we want to see the last five rows of the dataset, we use the tail()function of the dataframe like this:

```
#displayinf last five rows of dataset
data.tail()
```

Data Cleaning Cycle tail

Step 2: Merge Dataset

Merging the dataset is the process of combining two datasets in one and lining up rows based on some particular or common property for data analysis. We can do this by using the merge() function of the dataframe. Following is the syntax of the merge function:

```
DataFrame_name.merge(right, how='inner', on=None, left_on=None, right_on=None, left_index=False,
right_index=False, sort=False, suffixes=('_x', '_y'), copy=True, indicator=False, validate=None)
```

[\[source\]](#)

But in this case, we don’t need to merge two datasets. So, we will skip this step.

Step 3: Rebuild Missing Data

To find and fill in the missing data in the dataset, we will use another function. There are 4 ways to find the null values if present in the dataset. Let’s see them one by one:

Using isnull() function:


```
data.isnull()
```

Data Cleaning Cycle isnull

This function provides the boolean value for the complete dataset to know if any null value is present or not.

Data Cleaning Using Pandas in Python – Complete Guide for Beginners


```
data.isna()
```

 isna function

This is the same as the isnull() function. Ans provides the same output.

Using isna().any()

```
data.isna().any()
```

 isna().any()

This function also gives a boolean value if any null value is present or not, but it gives results column-wise, not in tabular format.

Using isna().sum()

```
data.isna().sum()
```

This function gives the sum of the null values preset in the dataset column-wise.

Using isna().any().sum()

```
data.isna().any().sum()
```

 isna().any().sum()

This function gives output in a single value if any null is present or not.

There are no null values present in our dataset. But if there are any null values preset, we can fill those places with any other value using the fillna() function of DataFrame. Following is the syntax of fillna() function:

```
DataFrame_name.fillna(value=None, method=None, axis=None, inplace=False, limit=None, downcast=None)
```

[\[source\]](#)

This function will fill NA/NaN or 0 values in place of null spaces. You may also drop null values using the dropna method when the amount of missing data is relatively small and unlikely to affect the overall.

Step 4: Standardization and Normalization

Data Standardization and Normalization is a common practices in machine learning.

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero, and the resultant distribution has a unit standard deviation.

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

To know more about this, [click here](#).

This step is not needed for the dataset we are using. So, we will skip this step.

Step 5: De-Duplicate Data

Data Cleaning Using Pandas in Python – Complete Guide for Beginners

affect the accuracy and efficiency of the analysis result. To find duplicate values in the dataset, we will use a simple dataframe function, i.e., `duplicated()`. Let's see the example:

```
data.duplicated()
```

De-Duplicate

This function also provides bool values for duplicate values in the dataset. As we can see, the dataset doesn't contain any duplicate values. If a dataset contains duplicate values, it can be removed using the `drop_duplicates()` function. Following is the syntax of this function:

```
DataFrame_name.drop_duplicates(subset=None, keep='first', inplace=False, ignore_index=False)
```

[\[source\]](#)

Step 6: Verify and Enrich the Data

After removing null, duplicate, and incorrect values, we should verify the dataset and validate its accuracy. In this step, we have to check that the data cleaned so far is making any sense. If the data is incomplete, we have to enrich the data again by data gathering activities like approaching the clients again, re-interviewing people, etc. Completeness is a little more challenging to achieve accuracy or quality in the dataset.

Step 7: Export Dataset

This is the last step of the data-cleaning process. After performing all the above operations, the data is transformed into a clean dataset, and it is ready to export for the next process in Data Science or Data Analysis.

Conclusion

Data cleaning is a critical task in data science that helps ensure the accuracy and reliability of analysis and decision-making. Through data cleaning, errors can be removed, data quality can be improved, and the data can be made more accurate and complete. By utilizing the various techniques and tools available for data cleaning in the Python Pandas library, data scientists can gain insights from the raw data and make better informed decisions.

Key Takeaways

- Data cleaning is the process of removing incorrect, corrupted, garbage, incorrectly formatted, duplicate, or incomplete data within a dataset.
- Data cleaning is essential for ensuring error-free data, data quality, accuracy, completeness, and efficiency in the analysis and decision-making process.
- Pandas is a popular data manipulation library in Python that provides powerful data-cleaning capabilities. It offers functions and methods to handle missing data, remove duplicate data, and fix data formatting issues.

Frequently Asked Questions

Q1. What do you mean by data type casting in the context of data analysis and data cleaning?

A. In the context of data analysis, casting data types means converting data from one type to another. This is often done to ensure consistency and accuracy in data analysis, as well as to enable specific operations or functions that are available for certain data types. For example, casting a string to a numerical data type can enable mathematical operations, while casting a numerical data type to a string can enable string-based operations.

Data Cleaning Using Pandas in Python – Complete Guide for Beginners

context of data cleaning with Pandas?

A. It is appropriate to drop missing values in data when the amount of missing data is small compared to the overall size of the dataset, and the missing data is randomly distributed or when they would skew the analysis. if the amount of missing data is substantial or the missing data is non-random, it may be more appropriate to impute the missing values rather than drop them, as dropping them may result in a biased or incomplete analysis.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author’s discretion.






[blogathon](#) [data cleaning](#)

About the Author



[Neelu Tiwari](#)

Our Top Authors

[Rahul S](#)[Sion Ch](#)[CHRA](#)[Boney](#)[Arind](#)[Prateek](#)[Sainth](#)[Pand](#)[view more](#)

Download

Analytics Vidhya App for the Latest blog/Article



Previous Post

[Offline Data Augmentation for multiple images](#)

Next Post

[Must Known Techniques for text preprocessing in NLP](#)

One thought on "Data Cleaning Using Pandas in Python – Complete Guide for Beginners"



Billy says:
November 07, 2022 at 12:02 pm

Thank you so much for sharing your insightful essays and insights with the world.
[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Data Cleaning Using Pandas in Python – Complete Guide for Beginners

Name*

Email*

Website

☒ Notify me of follow-up comments by email.

☒ Notify me of new posts by email.

Submit

Top Resources



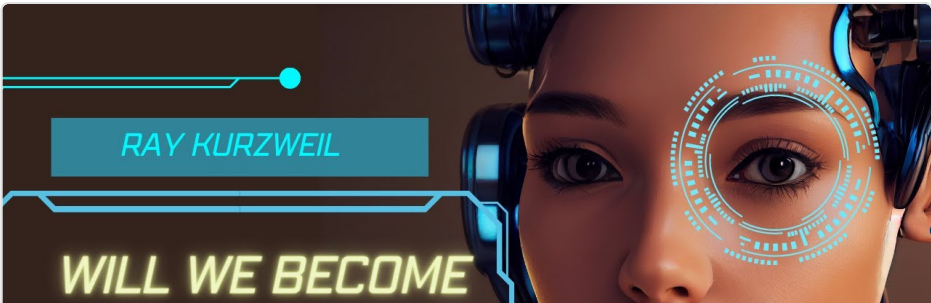
[FreedomGPT: Personal, Bold and Uncensored Chatbot Running Locally on Your..](#)

[K.sabreena](#) - APR 08, 2023



[How to Use ChatGPT as a Data Scientist?](#)

[Aravindpai Pai](#) - APR 08, 2023



[Futurist Ray Kurzweil Claims Humans Will Achieve Immortality by 2030](#)

[K.sabreena](#) - APR 06, 2023



[Understand Random Forest Algorithms With Examples \(Updated 2023\).](#)

[Sruthi E R](#) - JUN 17, 2021

Analytics Vidhya

[About Us](#)

[Our Team](#)

[Careers](#)

[Contact us](#)

Data Scientists

[Blog](#)

[Hackathon](#)

[Discussions](#)

[Apply Jobs](#)

