A Beginner’s Guide to Data Analysis in Python

[[](https://365datascience.com/blog/authors/natassha-selvaraj/)Natassha Selvaraj](https://365datascience.com/blog/authors/natassha-selvaraj/)21 Apr 202310 min read

In this day and age, [data](https://365datascience.com/trending/techniques-for-processing-traditional-and-big-data/) surrounds us in all walks of life. And so, with our growing treasure trove of information, the need to interpret what it tells us. However, it’s nearly impossible to decipher the vast amount of data we accumulate each day. When dealing with millions of data points, there are often patterns than come up that cannot be detected by the human eye.

This is where [data analysis](https://365datascience.com/trending/analysis-vs-analytics/) comes in – a quintessential [skill](https://365datascience.com/career-advice/career-guides/5-skills-data-science-job/) for any [aspiring data scientist](https://365datascience.com/career-advice/career-guides/entry-level-data-scientist/).

As a [data analyst](https://365datascience.com/career-advice/career-guides/data-analyst-intern/), you would use programming tools to break down large amounts of data, uncover meaningful [trends](https://365datascience.com/trending/trend-analysis/), and help companies make effective business decisions. The results are then presented in a way that is simple and comprehensive so that stakeholders can take action immediately.

In this beginner-friendly article, we will teach you how to use [Python](https://365datascience.com/tutorials/python-tutorials/learning-python-programming/) step-by-step and shed some light on why it’s so important through real-life examples. What is more, we will provide you with the code and all the necessary resources you need to get started.

Table of Contents

* [Real-life Data Analysis Example](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#1)
* [What Software to Use for Data Analysis?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#2)
* [How to Prepare for Data Analysis in Python?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#3)
  + [Python Installation Pre-Requisites](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#4)
  + [The Titanic Dataset](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#5)
  + [Loading the Dataset](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#6)
  + [Dataset Summary Statistics](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#7)
  + [Data Cleaning and Preprocessing](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#8)
  + [Data Imputation](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#9)
* [How to Perform Univariate Analysis?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#10)
* [How to Analyze the Relationship Between Variables?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#11)
* [How to Visualize the Relationship Between Variables?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#12)
* [What Are the Data Analysis Outcomes?](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#13)
* [Data Analysis in Python: Next Steps](https://365datascience.com/tutorials/python-tutorials/data-analysis-python/#14)

Real-life Data Analysis Example

Let’s take a simple example to understand the workflow of a real-life [data analysis project](https://365datascience.com/tutorials/python-tutorials/essential-python-projects/).

Suppose that Store A has a [database](https://365datascience.com/tutorials/sql-tutorials/basic-database-terminology/) of all the customers who have made purchases from them in the past year. They plan to use it to come up with personalized promotions and products to target different customer groups.

For this reason, they hire a [data analyst](https://365datascience.com/career-advice/career-guides/data-analyst-vs-business-analyst/) to compile all the purchase data and, based on [customer analytics](https://365datascience.com/courses/customer-analytics-in-python/), will advise the business on how each group should be targeted.

The analyst can provide recommendations in many different ways:

* **First, they can group customers based on data points.** This includes variables like time of purchase, frequency of purchase, and when the customer last visited the store.
* **Second, they can come up with promotions based on income level.** Customers who only make purchases when there are sales going on can be attracted with bulk promotions or package discounts.
* **Finally, the analyst can give recommendations based on preferred items.** In other words, these are the products customers purchase most. If they are often bought together, then they should be sold next to each other, or displayed on the same aisle.

Consumer behavior can also be taken into consideration.

For example, Andrew goes to Store A for frozen foods and packaged meat on a weekly basis. As of recently, he has started to shop for salads, vegetables, and protein shakes. Perhaps, then, we can conclude that Andrew is on a weight loss journey, or he desires to live a healthier lifestyle.

With this knowledge, Store A can now promote a larger variety of products. They can send Andrew coupons and promote items like gym equipment, sneakers, protein bars, and a variety of different sportswear.

Data analysis, when done correctly, is incredibly powerful. As evidenced by the example we showed, it provides you with actionable insights that you can then use to drive business value to a company.

What Software to Use for Data Analysis?

With the computing power available today, it is possible to perform data analysis on millions of data points in just a couple of minutes. In general, [data scientists](https://365datascience.com/career-advice/career-guides/data-scientist-2021/) use statistical software like R or programming languages like [Python](https://365datascience.com/tutorials/python-tutorials/why-learn-python/).

In this guide, we will show you how to analyze data using 2 popular [Python libraries](https://365datascience.com/tutorials/python-tutorials/best-python-libraries/) — [pandas](https://pandas.pydata.org/) and [Seaborn](https://seaborn.pydata.org/).

How to Prepare for Data Analysis in Python

Python Installation Pre-Requisites

To follow along with this tutorial, you will need to have a Python IDE running on your device. We suggest using a [Jupyter Notebook](https://365datascience.com/tutorials/python-tutorials/why-python-for-data-science-and-why-jupyter-to-code-in-python/) since its interface makes it easier for you to create and view visualizations.

Then, install the [pandas](https://pypi.org/project/pandas/) and [Seaborn](https://pypi.org/project/seaborn/) library on your device.

The Titanic Dataset

We will be using the [Kaggle Titanic dataset](https://www.kaggle.com/c/titanic/data?select=train.csv) for this tutorial, so before we begin, download it on your device.

*Note: Make sure to only download the file named ‘train.csv’ as we won’t need the rest.*

The file contains information about passengers who were on board the Titanic when the collision took place. We will use this data to perform exploratory data analysis in [Python](https://365datascience.com/tutorials/python-tutorials/why-learn-python/) and better understand the factors that contributed to a passenger’s survival of the incident.

Loading the Dataset

Open your [Jupyter Notebook](https://365datascience.com/tutorials/python-tutorials/jupyter-notebook-tutorial/) and navigate to the directory where you’ve saved the dataset. Then, create a new Python file and run the following lines of code:

import pandas as pddf = pd.read\_csv('train.csv')

df.head()

It will generate output that looks like this:

Notice that the data frame has 12 columns. Here’s a description of each of these variables:

1. **PassengerId**: a unique ID assigned to each passenger.
2. **Survived**: whether a passenger survived the collision. A label of 1 indicates that the passenger survived, while a label of 0 – that they didn’t.
3. **Pclass**: a label of 1 indicates that the passenger was traveling in first class, 2 indicates they were traveling second class, and 3 indicates they were traveling third class.
4. **Name**: the passenger’s name
5. **Sex**: the passenger’s gender
6. **Age**: the passenger’s age in years
7. **SibSp**: the number of siblings/spouses aboard
8. **Parch**: the number of parents/children aboard
9. **Ticket**: their ticket number
10. **Fare**: the fare that the passenger paid to board the ship
11. **Cabin**: the cabin number the passenger was in
12. **Embark**: the port which the passenger embarked from

Dataset Summary Statistics

Now that we have a basic understanding of each variable, let’s dive deeper and obtain further insights about them. This will help us find answers to questions such as the average age of a passenger who was aboard the Titanic.

Begin by running the following line of code:

df.describe()

The resulting data frame provides us with descriptive statistics for all the [numeric variables](https://365datascience.com/tutorials/statistics-tutorials/numerical-categorical-data/) in our dataset:

Let’s take a closer look at what each variable means:

* **Count:**the number of rows in the dataset that are populated with non-null values. There are 891 unique passenger IDs in this dataset. All the other variables also have 891 rows of data populated, with the exception of ‘Age’which only has 714 rows. This means that there are 177 passengers in the dataset who aren’t tagged with an age value.
* **Mean:**the mean value in each column. The mean age of passengers aboard the Titanic, for example, was 30.
* **Std:**how much deviation each column has from the mean.
* **Min:**the minimum value of each variable. For example, the minimum value for *‘SibSp’*is 0, meaning that there were passengers who traveled without their siblings and spouses.
* **25%, 50%, and 75%:**the 1st quartile, 2nd quartile (median), and 3rd quartile.
* **Max:**the highest value for each variable in the dataset. From the data frame above, we can see that the oldest passenger aboard the Titanic was 80 years old.

Data Cleaning and Preprocessing

[Data preprocessing](https://365datascience.com/courses/data-cleaning-preprocessing-pandas/) is one of the most important steps when conducting any kind of data science activity. Earlier, we noticed that the ‘Age’ column had some missing values in it. Let’s dive deeper to see if there are any further inconsistencies in our dataset.

Run the following lines of code:

df.isnull().sum()

As a result, we see that there are 3 columns with missing values — Age, Cabin, and Embarked:

We can deal with these missing values in a few different ways. The simplest option is to simply **drop all the rows that contain missing values**.

Run the following lines of code to do this:

df2 = df.copy()

df2 = df2.dropna()

df2.info()

*Note: Notice that we are creating a copy of the data frame before removing missing values. This is done so that the original frame isn’t tampered with and we can go back to it anytime without losing valuable data. It is often a best practice to create a copy before performing data manipulation.*

After removing all the rows that contain missing values, we obtain this summary:

Notice that earlier there were 891 rows. By dropping rows with missing values, we have dramatically reduced the size of this data frame by more than half. This isn’t a good practice. We lose a lot of valuable data by simply removing rows that contain missing values.

Data Imputation

Let’s try a second approach —**imputation**. In other words, the process of replacing missing data with substituted values.

First, impute missing values in the ‘Age’column. We will use mean imputation in this case — substituting all the missing age values with the average age in the dataset.

We can do this by running the following line of code:

df3 = df.copy()

df3['Age'] = df3['Age'].fillna(df3['Age'].mean())

Now, let’s move on to the ‘Cabin’column. We will replace the missing values in this column with the majority class:

df3['Cabin'] = df3['Cabin'].fillna(df3['Cabin'].value\_counts().index[0])

We can do the same for ‘Embarked’:

df3['Embarked'] = df3['Embarked'].fillna(df3['Embarked'].value\_counts().index[0])

We have successfully handled missing values in the dataset without losing any valuable data. Let’s now proceed to perform some exploratory data analysis with [Python](https://365datascience.com/tutorials/python-tutorials/learning-python-programming/).

How to Perform Univariate Analysis

Univariate analysis is the process of performing a statistical review on a single variable.

We will start by creating a simple [visualization](https://365datascience.com/trending/data-visualization-project-ideas/) to understand the [distribution](https://365datascience.com/tutorials/statistics-tutorials/distribution-in-statistics/) of the ‘Survived’variable in the Titanic dataset. Our aim is to answer simple questions with the help of available data, such as:

* How many passengers survived the Titanic collision?
* Were there more fatalities than survivors?

In the Seaborn library, we can create a count plot to visualize the distribution of the ‘Survived’variable. Essentially, a count plot can be thought of as a [histogram](https://365datascience.com/tutorials/statistics-tutorials/numerical-data-histogram/) across a categorical variable.

To do this, run the following code:

import seaborn as sns

sns.countplot(x='Survived',data=df)

Now, Python should render the following chart on your screen:

By looking at the results, we can tell that a majority of the passengers didn’t survive the Titanic collision.

To get the exact breakdown of passengers who survived and those who didn’t, we can use an in-built function of the pandas library called ‘value\_counts()’:

df['Survived'].value\_counts()

This function gives us a breakdown of unique values in each category:

Seaborn provides you with many other options for [data visualization](https://365datascience.com/courses/data-visualization/). You can create [pie charts](https://365datascience.com/tutorials/statistics-tutorials/bar-pie-pareto-charts/), violin plots, and box plots to further understand the distribution of every variable in the [dataset](https://365datascience.com/tutorials/python-tutorials/free-public-datasets-python/).

Take a look at [Seaborn’s user guide](https://seaborn.pydata.org/tutorial.html) to gain a better understanding of the [different types of visualizations](https://365datascience.com/trending/chart-types-and-how-to-select-the-right-one/) you can create.

How to Analyze the Relationship Between Variables

Now, we can move on to analyzing the relationships between different variables in our dataset.

Before starting any analysis, however, it is important to frame data questions. These will tell us exactly what we want to know from the information we have at hand — and it is useless to start exploring data with no end goal in mind.

In this case, we will run an analysis to try and answer the following questions about Titanic survivors:

* Did a passenger’s age have any impact on what class they traveled in?
* Did the class that these passengers traveled in have any correlation with their ticket fares?
* Were passengers who paid higher ticket fares located in different cabins as compared to passengers who paid lower fares?
* Did ticket fare have any impact on a passenger’s survival?

Using the questions above as a rough guideline, let’s begin the analysis.

How to Visualize the Relationship Between Variables

First, let’s create a boxplot to visualize the relationship between a passenger’s age and the class they were traveling in:

sns.boxplot(data=df,x='Pclass', y='Age')

You will see a plot like this appear on your screen:

If you haven’t seen a boxplot before, here’s how to read one:

* **Quartiles**: the edges of the boxplot represent the 1st and 3rd quartile of the variable. Meanwhile, the line in the middle represents the [median](https://365datascience.com/tutorials/statistics-tutorials/mean-median-and-mode/).
* **Minimum and maximum**: the two lines right at the end of the boxplot tell us the minimum and maximum value of the variable.
* **Outliers**: any point that lies outside the minimum and maximum is considered an outlier.

Taking a look at the boxplot above, notice that passengers traveling first class were older than passengers in the second and third classes. The median age of first-class passengers is around 35, while it is around 30 for second-class passengers, and 25 for third-class passengers.

This makes sense since older individuals are likely to have accumulated a larger amount of wealth and can afford to travel first class. Of course, there are exceptions, which is why you can observe passengers above 70 in the second and third classes – our outliers.

Now, let’s look at the relationship between passenger class and ticket fares. Even before performing this analysis, we understand that first-class tickets are more expensive than second- and third-class ones. Let’s verify this assumption with the help of available data.

Run the following code to create a [bar chart](https://365datascience.com/tutorials/python-tutorials/bar-chart-python-matplotlib/) visualization:

sns.barplot(data=df,x='Pclass',y='Fare')

The chart clearly shows us that first-class passengers paid a lot more for their tickets as compared to second- and third-class passengers:

Now, let’s see if passengers who paid different fare prices were allocated to different cabins.

Before we do that, however, run the following lines of code to see the number of unique cabins in the dataset:

df\_cabin = df[['Cabin','Fare']]

df\_cabin = df\_cabin.dropna()

df\_cabin['Cabin'].nunique()

The output is 147.

‘Cabin’ is a [categorical variable](https://365datascience.com/tutorials/statistics-tutorials/numerical-categorical-data/), which means that the passengers in the dataset have been allocated to 147 different rooms. Essentially, the variable has high cardinality, i.e. it has too many categories.

In this case, we need to perform some [data preprocessing](https://365datascience.com/courses/data-preprocessing-numpy/) before we try to find the relationship between ticket fares and a passenger’s cabin. With such a large number of unique values in the dataset, it is virtually impossible to come up with any meaningful conclusion.

Run the following lines of code to clean and transform the ‘Cabin’ column:

def clean\_cabin(cabin):

return(cabin[0])df\_cabin = df\_cabin['Cabin'].apply(clean\_cabin)

Done! Now, if you check for the number of unique cabin values, there will only be 8.

Let’s analyze the relationship between a passenger’s ticket fare and the cabin they were allocated with this line of code:

sns.catplot(data=df\_cabin,x='Cabin',y='Fare')

As you can see, a significant portion of passengers in cabin B seem to have paid higher ticket fares than passengers in any other cabin:

Moving on, let’s look into the relationship between a passenger’s ticket fare and survival:

sns.barplot(data=df,x='Survived',y='Fare')

As expected, passengers with higher ticket fares had a higher chance of survival:

This is because they could afford cabins closer to lifeboats, which meant they could make it out on time.

By extension, this should also mean that the first-class passengers had a higher likelihood of survival. Let’s confirm this:

sns.barplot(data=df,x='Pclass',y='Survived')

The chart confirms our assumptions — there were more first-class passengers who survived the Titanic collision:

What Are the Data Analysis Outcomes?

Performing the analysis has helped us come up with answers for the questions we outlined earlier:

* *Did a passenger’s age have any impact on what class they traveled in?*Yes, older passengers were more likely to travel first class.
* *Did the class that passengers traveled in have any correlation with their ticket fares?*Yes, first-class passengers paid more for their tickets.
* *Were passengers who paid higher ticket fares in different cabins as opposed to passengers who paid lower fares?*Yes, passengers who paid higher ticket fares seemed to mostly travel in cabin B. However, the relationship between ticket fare and cabin isn’t too clear because there were many missing values in the *‘*Cabin’ This might have compromised the quality of our analysis.
* *Did ticket fare have any impact on a passenger’s survival?* Yes, first-class passengers were more likely to survive the collision.

Remember, when structuring any [data science project in Python](https://365datascience.com/tutorials/python-tutorials/essential-python-projects/), it is vital to start out with an outline of the type of questions you’d like to answer. If we didn’t set off with the above questions in mind, we would have wasted a lot of time looking into the [dataset](https://365datascience.com/tutorials/python-tutorials/free-public-datasets-python/) without any direction, let alone [identifying patterns](https://365datascience.com/trending/trend-analysis/) that confirmed our assumptions.

Data Analysis in Python: Next Steps

In most real-world projects, [data scientists](https://365datascience.com/career-advice/career-guides/data-scientist-career-path/) are often presented with a [business use case](https://365datascience.com/career-advice/expert-interviews/). They then transform this use case into a set of questions like we did above and validate their assumptions with the help of data. Then, they present their findings in a format that is easy for stakeholders to understand.

Moreover, [pandas](https://pandas.pydata.org/) and [Seaborn](https://seaborn.pydata.org/) are Python tools that most data scientists use for their workflow in large organizations. It is a good idea to build a strong foundation with these libraries.

If you’d like to explore the topic of data cleaning, preprocessing, and analysis further, the 365 [Data Cleaning and Preprocessing with pandas](https://365datascience.com/courses/data-cleaning-preprocessing-pandas/) course is a comprehensive guide that will teach you all the essentials in order to boost your [resume](https://365datascience.com/career-advice/job-interview-tips/data-science-resume/).

However, data analysis is just one piece of the puzzle.  There’s so much more to learn before you can [break into data science](https://365datascience.com/career-advice/career-guides/career-data-science-ultimate-guide/).

Whenever you’re ready for the next step, the 365 Data Science Program offers you self-paced courses led by renowned industry experts. Starting from the very basics all the way to advanced specialization, you will learn by doing with a myriad of practical exercises and real-world business cases. If you want to see how the training works, start with our free lessons by signing up below.

Learn data science with industry experts

[Try For Free](https://365datascience.com/signup/)



[Natassha Selvaraj](https://365datascience.com/blog/authors/natassha-selvaraj/)

Senior Consultant

Natassha is a data consultant who works at the intersection of data science and marketing. She believes that data, when used wisely, can inspire tremendous growth for individuals and organizations. As a self-taught data professional, Natassha loves writing articles that help other data science aspirants break into the industry. Her articles on her personal blog, as well as external publications garner an average of 200K monthly views.