Ultimate Data Science Programming in Python

Master data science libraries with 300+ programs, 2 projects, and EDA GUI tools

Saurabh Chandrakar

www.bpbonline.com

First Edition 2025

ISBN: 978-93-65895-667

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true to correct and the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

www.bpbonline.com

Dedicated to

My parents

***Dr. Surendra Kumar Chandrakar** and*
Smt. Bhuneshwari Chandrakar
my brother
Shri Pranav Chandrakar

my sister-in-law

Smt. Silky Chandrakar

my wife

Smt. Priyanka Chandrakar

and my lovely son

Master Yathartha Chandrakar

About the Author

Saurabh Chandrakar is a Research and Development Engineer (Dy. Manager) at Bharat Heavy Electricals Limited (BHEL) Hyderabad. He is the winner of the best executive award in the Operations Division by BHEL Hyderabad. He has been awarded the prestigious BHEL Excellence Award under Anusandhan category for the projects "Redundant Composite Monitoring System of Power Transformers project" and "Innovation and demonstration of Digital Sub Station with in-house developed IEC61850 compliant intelligent electronic devices and optical current transformers for the year 2020-21". He has 25 copyrights, 5 patents granted, and 2 patents filed.

Moreover, he has published six books in reputed publications such as BPB New Delhi (Programming Techniques using Python, Python for everyone, Building Modern GUIs with tkinter and Python, Python GUI with PyQt, Scitech Publications Chennai (Programming Techniques using Matlab) and IK International publishers (Microcontrollers and Embedded System Design). Additionally, he has also launched one video course on BPB titled "First Time Play with Basic, Advanced Python Concepts and Complete Guide for different Python certification exams all under one umbrella."

About the Reviewers

- **Nirakar Padhy** is an experienced Senior Data Scientist with a strong academic background, holding a Bachelor's degree in Mathematics from Mithibai College and a Master's degree in Machine Learning and Artificial Intelligence from Liverpool John Moores University.

  He specializes in advanced **Machine Learning** (**ML**) and **Artificial Intelligence** (**AI**) algorithms, mathematics, and statistics, along with strong skills in Python programming, data analysis, and distributed computing. Nirakar is also highly proficient in ML engineering, MLOps, model deployment and monitoring, CI/CD pipelines, and cloud computing platforms such as **Amazon Web Services** (**AWS**).

  Dedicated to mentoring the next generation of data professionals, Nirakar conducts workshops where he shares his extensive knowledge and practical insights, particularly in statistics, data science, machine learning, and MLOps.

  Outside of his professional pursuits, Nirakar is an avid movie enthusiast with a keen interest in personal finance. He enjoys playing cricket to stay active and relax and is always on the lookout for new hobbies and interests to explore.

- **Prasenjeet Damodar Patil** received a B.E in E and TC Engineering from Sant Gadgebaba Amravati University and an M. Tech. from Walchand College of Engineering Sangli, India. He did his PhD degree in E&TC Engineering from Sant Gadgebaba Amravati University. He has 15+ years of teaching experience. Currently, he is working as a Professor at the School of Computing, M.I.T A.D.T University, Pune. He has published more than 20+ papers in reputed Journals. His research interests include the Internet of Things and Digital Image Processing. He is also a certified Google Data Analytics Professional.

Acknowledgement

First and foremost, I would like to thank you all for selecting this book. It has been written with a beginner reader in mind. I take this opportunity to greet and thank my mentor "Prof. Nilesh Bahadure Sir" for motivating me and always communicating his expertise on topics related to Python. I am very thankful for being his protégé. I appreciate his belief in me, for always standing behind me and pushing me to achieve more. The phrase "Journey of a thousand miles begins with a single step" is something he always reminds me of.

My parents, Dr. Surendra Kumar Chandrakar and Smt. Bhuneshwari Chandrakar, my brother, Shri Pranav Chandrakar, my sister-in-law, Silky Chandrakar, my beloved wife, Mrs. Priyanka Chandrakar, my adorable son, Yathartha Chandrakar, and all of my friends who have inspired me and given me the confidence over the years. Last but not least, I would like to express my sincere gratitude to the staff at BPB Publications for their contributions and insights that made parts of this book possible.

Preface

The purpose of this book is to introduce individuals to the dynamic field of data science in Python, unraveling the capabilities of essential libraries that drive data analysis and manipulation. For those with varied levels of programming experience, this book serves as a gateway to the intricate world of NumPy, SciPy, Matplotlib, Pandas, Polars, Seaborn, and usage of ChatGPT for all these open-source libraries. Readers will embark on a hands-on journey, navigating through a multitude of solved examples that illuminate the practical application of each library. Beginning with foundational concepts, the chapters progressively explore advanced functionalities, empowering readers to harness the full potential of these libraries. Whether it is mastering the manipulation of data with pandas, visualizing insights with Matplotlib, or leveraging the performance-oriented polars library, this book provides a comprehensive guide for both beginners and seasoned data scientists. With a blend of theoretical understanding and practical implementation, readers will acquire the skills necessary to tackle real-world data challenges and unlock the vast possibilities within the Python data science ecosystem. By mastering open-source libraries in data science, readers will be able to apply this knowledge to solve real-world problems and work on various useful projects according to their needs.

The first part of the book is dedicated to an in-depth exploration of the NumPy library, laying the groundwork for proficient data manipulation in Python. Readers will explore fundamental concepts such as the creation of NumPy arrays, understanding the distinctions between lists and arrays, and the application of arithmetic operations. The chapter also sheds light on advanced topics like broadcasting and matrix multiplication. With practical examples, this section ensures a comprehensive understanding of NumPy's capabilities, equipping readers with the skills necessary to efficiently handle numerical data for various data science applications. Then, we will unfold the powerful functionalities of both SciPy and Matplotlib, extending the readers' capabilities in the realm of scientific computing and data visualization. The exploration of SciPy introduces concepts like optimizers, sparse data handling, graph algorithms, and integration techniques, enabling readers to tackle a broad spectrum of scientific and engineering challenges. Transitioning seamlessly, Matplotlib is unveiled as a quintessential tool for data visualization, covering an array of plots such as line plots, bar plots, pie charts, histograms, scatter plots, and subplots. Various practical examples in this section will provide a comprehensive understanding of how to effectively communicate complex data through visually compelling plots, setting the stage for advanced data exploration.

In the later part of the book, the focus shifts to pandas, a versatile library for data manipulation and analysis. Readers will master the creation of pandas series and dataframes, along with advanced techniques like filtering, sorting, and aggregation. The exploration extends to the polars library, emphasizing its modern approach to dataframe manipulation and its performance advantages. The chapter on seaborn delves into statistical data visualization, covering essential plots like heatmaps, box plots, and scatter plots. Additionally, readers will discover the capabilities of ChatGPT in conjunction with open libraries of data science. With a rich array of solved examples in this section, readers will acquire a holistic skill set, empowering them to tackle diverse data challenges and innovate in their data science endeavors.

This book is divided into **16 chapters**. Each chapter description is listed as follows:

Chapter 1: Environmental Setup for Using Data Science Libraries in Python - This chapter covers the key aspects of setting up an effective Python programming environment, highlighting the importance of using an **Integrated Development Environment** (**IDE**). It begins with a step-by-step guide to installing Jupyter Notebook on a Windows platform, along with an overview of its functionalities. The focus will then move to installing **Visual Studio Code** (**VSCode**) for Python development and exploring its features for coding. Finally, the chapter introduces essential Python data science libraries, providing a foundation for learners to utilize these powerful tools in their programming journey.

Chapter 2: Exploring Numpy Library for Data Science in Python - This chapter covers a clear understanding of the comparisons between lists and numpy arrays and what they entail. We will see the creation of ndarrays through different methods, including utilizing list and tuple data structures. We will also demonstrate various functions using Python code snippets and peep into ndarray creation with random values using a module. The distinction between view and copy in numpy will be explained using examples that are crucial for memory efficiency. We will learn different methods to access individual elements or subsets of elements in ndarrays. We will explore how to iterate over elements of ndarrays using loops and explore various arithmetic operators available in numpy. In the end, we will learn how to use broadcasting to perform operations on ndarrays with different shapes.

Chapter 3: Exploring Array Manipulations in Numpy - This chapter covers an idea about various array manipulation functions and variables available in numpy. We will understand different methods for joining ndarrays and explore how to split ndarrays into smaller ones. Sorting of ndarrays in numpy will also be discussed. Some of the search functions available in numpy will be explored along with the insertion and deletion of elements into/from ndarrays will be carried out. The usage of the dot function in numpy for matrix multiplication will be explored. Finally, the linalg module in numpy for linear algebra will be well understood with various examples.

Chapter 4: Exploring Scipy Library for Data Science in Python - This chapter will cover the basic difference between numpy and scipy array. The aim is to explore the scipy constants which represent physical quantities, mathematical, scientific, and other useful values. We will also discuss the optimizers in scipy where different optimization algorithms available will be discussed with some examples, like finding the roots of an equation and many more. Additionally, we will study sparse data and sparse matrix representation in scipy, enabling efficient storage and manipulation of large, mostly empty matrices. Different sparse matrix types in scipy will be well explored. We will learn about graphs in scipy where the module provides several functions to effectively analyze and work with sparse graphs. The chapter covers integration techniques in scipy which are useful for numerical integration of functions. In the end, we will discuss the interpolation methods in scipy which enables the estimation of values between known data points.

Chapter 5: Line Plot exploration with Matplotlib Library - This chapter covers Python's data visualization tools along with a wide array of techniques and concepts to enhance our ability to represent and analyze data effectively. First to learn will be line plots where we will explore the creation of line plots by passing two ndarrays, adding essential elements like titles, x-labels, and y-labels, and advancing our skills with properties like linestyle, color, alpha, linewidth, markersize, markerfacecolor, and figure size customization. We will also learn how to plot multiple lines in a single plot, creating line plots with single ndarrays, incorporate grid lines (both major and minor), and effectively utilize legends, xlim, and ylim functions.

Chapter 6: Charting Dta with Various Visuals Using Matplotlib - This chapter covers the bar plots where various aspects such as changing the color, width, bottom position, left alignment and right alignment of individual bars, as well as adding labels to bars will be learned. We will also venture into Horizontal Bar Charts, Stacked Bar Charts (both vertical and horizontal), and Grouped Bar Charts, providing us with a comprehensive understanding of bar chart variations. Additionally, we will master the art of creating i.e. Charts, Histograms, Scatter Plots, and subplots in Matplotlib, equipping us with a versatile toolkit for data visualization and analysis in Python.

Chapter 7: Exploring Pandas Series for Data Science in Python - This chapter covers a comprehensive range of topics related to pandas series and dataframes in Python, equipping them with the skills and knowledge needed for effective data manipulation and analysis. We will start by exploring pandas series, covering fundamental aspects such as series creation, data access, slicing, and filtering. We will also discuss more advanced topics like using callable objects for selection, leveraging useful attributes, applying functions with the ‘apply’ method, aggregating data, and performing basic arithmetic operations. The chapter will further emphasize series transformations and iteration for a well-rounded understanding.

Chapter 8: Exploring Pandas Dataframe for Data Science in Python - This chapter covers the pandas dataframes, readers will gain insights into dataframe construction and exclusive methods and attributes applicable only to dataframes. We will address the critical task of handling missing data, arithmetic operations for dataframes, and the addition of new columns. The usage of the ‘fillna’ method for handling missing values will also be covered. Sorting, ranking, and filtering data within dataframes will be thoroughly explored along with techniques for checking data inclusion, identifying missing values, and handling duplicates.

Chapter 9: Advanced Dataframe Filtering Techniques - This chapter covers various methods for renaming index labels and column names, as well as efficient ways to delete rows and columns from dataframes. It will introduce the powerful ‘query’ method for data filtering and explore additional advanced techniques such as using ‘apply’ to manipulate dataframe data and finding the largest and smallest values with ‘nlargest’ and ‘nsmallest.’ Text data handling within pandas will be addressed, demonstrating how to filter dataframe rows with string methods. Practical examples of data modification in dataframes will provide readers with hands-on experience in real-world data manipulation.

Chapter 10: Exploring Polars Library for Data Science in Python - This chapter covers a comprehensive exploration of the polars data manipulation library in Python. First, we will highlight the key differences between pandas and polars, shedding light on the unique advantages and features that polars bring to the table. Then, we will explore polars data types, categorizing them into Numeric, Nested, Temporal, and Others Groups, providing a solid understanding of the data structures. Readers will then venture into the heart of polars, gaining insights into its data structures, including series and dataframe, and learning how to work with them effectively. Next, we will unveil the concept of contexts in polars, focusing on essential operations, selection, and filtering techniques. The groupby concept is demystified, enabling readers to harness the power of polars for data aggregation and summarization. The concept of a Lazy API will be well elucidated, demonstrating how it enhances performance and resource utilization.

Chapter 11: Exploring Expressions in Polars - This chapter will cover the subtleties of expressions in polars along with basic operators, column selections, functions, casting, string operations, aggregation, handling missing data, and leveraging folds, lists, and arrays. We will also introduce the integration of numpy in polars, offering a bridge between these two powerful libraries. Finally, readers will gain a deep understanding of when different operations will be performed with polars and pandas, thus helping to make informed choices when working with data in Python.

Chapter 12: Exploring Seaborn Library for Data Science in Python - This chapter covers the concept of some basic statistical terms, such as quantitative and qualitative variables, mean, and their significance in data analysis. They will understand the distinction between variables that represent numerical measurements (quantitative) and those that represent categories or labels (qualitative). The reader will also be familiar with various built-in datasets available in seaborn, gaining practical exposure to real-world datasets for data visualization and analysis. Furthermore, the reader will comprehend the principles of plot styling in seaborn, learning how to customize the aesthetics of plots for better presentation and interpretation. The concept of color palettes in seaborn, including qualitative, sequential, and diverging palettes, will be covered, enabling the reader to choose appropriate color schemes for different types of data and visualizations.

Chapter 13: Crafting Seaborn Plots: KDE, Line, Violin, and Facets - This chapter covers the skills to create various advanced plots in seaborn, such as heatmap plots for visualizing matrices of data, KDE plots for exploring the distribution of a single variable, violin plots for combining aspects of box plots and KDE plots, line plots for depicting trends, scatter plots for visualizing the relationship between two variables, joint plots for combining different types of plots in a single grid, and FacetGrid for creating grids of subplots based on categorical variables.

Chapter 14: Integrating Data Science Libraries with ChatGPT Prompts - This chapter covers the topics to illuminate readers on the versatile applications of ChatGPT in the exploration of open-source data science libraries. Firstly, it aims to guide users in formulating effective prompts for ChatGPT, tailored to each library, including Numpy, Scipy, Matplotlib, Pandas, Polars, and Seaborn to extract valuable insights and coding solutions. We will cover various topics, including leveraging ChatGPT for generating insightful prompts and queries to explore data science libraries, seeking assistance for Python code utilizing these libraries, collaboratively generating comprehensive cheat sheets, and addressing challenges faced by novice users.

Chapter 15: Exploring Automated EDA Libraries for Machine Learning - This chapter will introduce learners to a diverse array of Python libraries tailored for data visualization and analysis, each uniquely contributing to an enriched data exploration experience. Beginning with the PyGWalker Visual library, readers will discover how to seamlessly integrate it with datasets to create visually appealing representations. The exploration continues with the dataprep library, offering insights into efficient data preparation and analysis. Subsequently, learners will explore the capabilities of the autoviz and pandas_visual_analysis, libraries, each providing distinct functionalities and visualization approaches to enhance the understanding and interpretation of datasets.

Chapter 16: Case Studies Using Data Science Libraries - This chapter covers a practical application of data science methodologies by exploring two distinct datasets. The first case study shifts attention to an Electrical Fault Classification dataset. This dataset is designed to help identify and categorize various types of electrical faults in power systems. This dataset typically includes features such as voltage, current, and other relevant electrical parameters collected over time or during specific fault events. By analyzing these features, we will explore no-fault and different fault conditions. Libraries like Pandas and numpy will be used for data manipulation and preprocessing. Visualization libraries such as Matplotlib and Seaborn help in understanding the data distribution and model performance. The second case study focuses on the Titanic dataset, utilizing a range of data science libraries. Learners will employ popular Python libraries such as pandas, numpy and seaborn to preprocess and visualize the data, gaining insights into factors influencing passenger survival and various other factors focusing on the null values.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/bzpl6rq

The code bundle for the book is also hosted on GitHub at **https://github.com/bpbpublications/Ultimate-Data-Science-Programming-in-Python**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

1. Environmental Setup for Using Data Science Libraries in Python 1

Introduction 1

Structure 1

Objectives 2

Introduction to Python 2

Setup installation in Windows for Jupyter Notebook 5

Insights of Jupyter Notebook 11

Demo program using Jupyter Notebook 13

Demo program using VSCode 20

Introduction to data science libraries in Python 22

Conclusion 24

Points to remember 24

Questions 24

2. Exploring Numpy Library for Data Science in Python 25

Introduction 25

Structure 25

Objectives 26

Commonalities shared by lists and numpy arrays 26

Contrasts separating lists from numpy arrays 26

The process of creating numpy arrays 28

Creating a 1-D array using a list 29

Creating a 1-D array using a tuple 29

Creating a 2-D array using nested lists 30

Creating an array with a specific dtype 31

Creating an array of object type 32

Creating a 1-D array with arange() function 32

Array creation using linspace() function 34

Utilizing the zeros() function in numpy 36

Utilizing the ones() function in numpy 37
Utilizing the full() function in numpy 38
Using eye() function 39
Utilizing diag() function 40
Utilizing empty() function 42
Contrast between zeros and empty function 43
Using random module for ndarray creation 43
randint() 43
shuffle() 44
Elements access of ndarray 48
Iterate elements of ndarray 49
Arithmetic operators in numpy 51
Broadcasting concept in numpy 53
Conclusion 55
Points to remember 56
Questions 56
References 57

3. Exploring Array Manipulations in Numpy 59

Introduction 59
Structure 59
Objectives 60
Array manipulation functions/variables in numpy 60
reshape() 60
resize() 62
flatten() 63
flat variable 64
ravel() 65
transpose() 66
Multiple arrays joining into a single array 69
concatenate() 69
stack() 71

vstack() *72*

hstack() *73*

dstack() *74*

Splitting of arrays in numpy 76

split() *76*

vsplit() *78*

hsplit() *78*

dsplit() *79*

array_split() *81*

Sorting of ndarray in numpy 81

Elements searching of ndarray in numpy 83

where() *83*

condition based selection *84*

Elements insert into ndarray 84

insert() *85*

append() *86*

Elements delete from ndarray in numpy 87

delete() *87*

Usage of dot function for matrix multiplication 89

Usage of linalg module in numpy 90

Conclusion 91

Points to remember 91

Questions 92

4. Exploring Scipy Library for Data Science in Python 93

Introduction 93

Structure 94

Objectives 94

Differences between numpy and scipy 94

Exploring scipy constants 95

Metric *96*

Force *97*

Binary ... 97

Angle ... 97

Mass ... 98

Time ... 98

Pressure ... 99

Length ... 99

Volume ... 99

Speed ... 100

Temperature ... 100

Energy ... 100

Power ... 101

Area ... 101

Optimizers in scipy ... 101

Sparse data and sparse matrix in scipy ... 106

Block sparse row matrix ... 107

Coordinate list matrix ... 108

Compressed sparse column matrix ... 108

Compressed sparse row matrix ... 109

Diagonal sparse matrix ... 110

Dictionary of keys sparse matrix ... 111

Row-based linked list sparse matrix ... 112

Graphs in scipy ... 113

Adjacency matrix ... 113

connected_components method ... 113

Dijkstra method ... 114

Floyd_Warshall method ... 114

Spatial data in scipy ... 115

Triangulation function ... 116

Integrate in scipy ... 120

Single integration ... 121

Multiple integration ... 122

Interpolation in scipy ... 124

1D interpolation ... *124*

Spline interpolation ... *125*

splrep function ... *128*

splev function ... *128*

Interpolation with radial basis function ... *132*

Conclusion ... 133

Points to remember ... 133

Questions ... 134

5. Line Plot exploration with Matplotlib Library ... 135

Introduction ... 135

Structure ... 135

Objectives ... 136

Python data visualization tools ... 136

Line plot ... *137*

Line plot creation by passing 2 ndarrays ... 138

Adding title, xlabel and ylabel to the line plot ... 139

Advanced line plot ... 140

Linestyle property ... 141

Color property ... 143

Default color ... 144

Quick settings for color, marker and linestyle ... 145

mlc form ... *146*

clm form ... *147*

When unspecified then default color is blue ... *148*

Alpha property ... 149

Linewidth and markersize property ... 150

Markerfacecolor property ... 151

Customizing the figure size ... 152

Plotting multiple lines in a same plot ... 154

Line plot creation by passing a single ndarray ... 156

Addition of grid lines to a plot ... 157

Major and minor grid lines display 160

Legends display to a plot 162

Using xlim and ylim functions to a plot 165

Conclusion 170

Points to remember 170

Questions 171

6. Charting Data With Various Visuals Using Matplotlib 173

Introduction 173

Structure 173

Objectives 174

Barplot 174

Vertical bar chart and simple bar chart 175

Color change of each bar 176

Width change of each bar 177

Bottom change of each bar 179

Left alignment of each bar 180

Right alignment of each bar 181

Enhancing Bar visualization with labels 182

Horizontal bar chart 185

Stacked bar chart 187

Other bar charts 191

Pie chart 193

Histogram 195

Scatter plot 198

Subplot in matplotlib 200

Conclusion 203

Points to remember 204

Questions 204

7. Exploring Pandas Series for Data Science in Python 205

Introduction 205

Structure 206

Objectives 207

Pandas series 207

Pandas series constructor 207

Creating pandas series by passing a list 208

Creating pandas series by passing a dictionary 208

Creating pandas series by passing a numpy array 209

Creating pandas series from scalar value 209

Creating pandas series using RangeIndex 210

Creating pandas series with dtype and name 211

Exploring pandas series using copy parameter 211

Accessing series values using head and tail methods 212

Accessing elements in the pandas series 214

Pandas series slicing 214

Values extraction using loc and iloc indexers 216

Masking using Booleans for condition-based selection 217

Pandas series filtering 218

Callable object usage in selecting elements 219

Usage of the apply method to Pandas series 223

Aggregating of pandas series 224

Series object transformation 229

Iterate elements of the series 232

Conclusion 232

Points to remember 233

Questions 233

8. Exploring Pandas Dataframe for Data Science in Python 235

Introduction 235

Structure 235

Objectives 236

Pandas dataframe 236

Pandas dataframe constructor 236

Pandas dataframe creation 237

Pandas methods and attributes for dataframe 238

Dropping dataframe rows with null values 244

Arithmetic operators for dataframes 248

New columns addition to dataframes 250

Usage of fillna() method to the dataframe 252

Sorting values of the dataframe 256

Sorting dataframe by multiple columns 258

Sorting dataframe based on index 260

Ranking series of values by using the rank() method 262

Filter data from dataframe 264

Inclusion check in dataframe 266

Usage of isnull and notnull method 269

Usage of duplicated method 271

Usage of drop_duplicates method in dataframe 273

Usage of unique and nunique methods 274

Conclusion 275

Points to remember 275

Questions 276

9. Advanced Dataframe Filtering Techniques 277

Introduction 277

Structure 277

Objectives 278

Approaches of setting customized columns as index 278

Rows selection by index label with loc[] indexer 281

Accessing rows of dataframe by using index position 284

Renaming index and column names in dataframe 290

Renaming of index labels 290

Renaming of column names 292

Rows and columns deletion from the Dataframe 293

Filter dataframe using query method 296
Without using the query method 296
in and not in operators using the query method 298
Handling spaces in column names with query 299
Output: 300
Usage of apply() method to pandas dataframe 301
Using nlargest() and nsmallest() in pandas dataframe 304
Filtering dataframe with where() method 305
Handling of text data in Pandas 307
Filtering dataframe rows with string methods 309
Modifying data in pandas dataframe 312
Example of data aggregation in Pandas DataFrame 314
Conclusion 315
Points to remember 316
Questions 316

10. Exploring Polars Library for Data Science in Python 317

Introduction 317
Structure 318
Objectives 318
Preference of polars over pandas 318
Differences between pandas and polars 319
Polars data types 319
Numeric group 319
Nested group 320
Temporal group 321
Others group 321
Data structures in polars 327
Series 327
Dataframe 328
Contexts in polars 330
Basic operations, select and filter concept 331

GroupBy concept *338*

Importance of expressions in polars 339

Concept of lazy API in polars 341

Conclusion 343

Points to remember 344

Questions 344

11. Exploring Expressions in Polars 345

Introduction 345

Structure 345

Objectives 346

Concept of expressions in polars 346

Basic operators *346*

Column selections *349*

Functions *351*

Casting *353*

Strings *358*

Aggregation *360*

Missing data *362*

Folds *366*

List and arrays *369*

Usage of numpy in polars *372*

Operations comparison between pandas and polars 374

Conclusion 374

Points to remember 374

Questions 375

12. Exploring Seaborn Library for Data Science in Python 377

Introduction 377

Structure 378

Objectives 378

Some basic statistical terms to know 378

Quantitative variables *378*

Qualitative variables ... *379*

Built in seaborn datasets ... 379

Plot styling in Seaborn ... 379

Concept of color palette in Seaborn ... *384*

Qualitative palettes ... *386*

Sequential palettes ... *388*

Diverging palettes ... *389*

Heatmap plot in Seaborn ... 391

The box plot in seaborn ... 400

Conclusion ... 405

Points to remember ... 405

Questions ... 406

13. Crafting Seaborn Plots: KDE, Line, Violin and Facets ... 407

Introduction ... 407

Structure ... 407

Objectives ... 408

KDE plot in seaborn ... 408

The violin plot in seaborn ... 416

Line plot in seaborn ... 422

The scatter plot in seaborn ... 424

Facet grid in seaborn ... 430

Conclusion ... 434

Points to remember ... 435

Questions ... 435

14. Integrating Data Science Libraries with ChatGPT Prompts ... 437

Introduction ... 437

Structure ... 438

Objectives ... 438

ChatGPT prompts for exploring data science libraries ... 438

Querying ChatGPT for Python code with data science libraries ... 443

Querying ChatGPT with Python error ... 454

Data science library cheat sheets using ChatGPT 456
Novice challenges querying Python code via ChatGPT 458
Conclusion 459
Points to remember 459
Questions 460

15. Exploring Automated EDA Libraries for Machine Learning 461

Introduction 461
Structure 462
Objectives 462
Usage of PyGWalker visual library with a dataset 462
Usage of dataprep library with a dataset 467
Usage of autoviz library with a dataset 475
Usage of pandas_visual_analysis library with a dataset 478
Conclusion 484
Points to remember 484
Questions 485

16. Case Study Using Python Data Science Libraries 487

Introduction 487
Structure 487
Objectives 487
Case study on electrical fault classification 488
Case study on titanic dataset 504
Conclusion 520
Points to remember 520
Questions 521

Index 523-530

CHAPTER 1

Environmental Setup for Using Data Science Libraries in Python

Introduction

In this chapter, we will explore the data science library in Python. However, before learning about various data science libraries, it is quite important to create an environmental setup for installing and using these data science libraries in Python. Setting up the environment for utilizing data science libraries like NumPy, SciPy, Matplotlib, Pandas, and others in Python ensures effective data analysis and modeling workflows. This entails managing dependencies, version control, and package management to guarantee project compatibility and reproducibility. By creating isolated environments, potential conflicts between different library versions are mitigated, facilitating seamless collaboration and reproducibility of results. So, let us get into the intricate details of Python installation and **Integrated Development Environments** (**IDE**s) like VSCode and Jupyter Notebook for writing and executing the Python code.

Structure

In this chapter, we will discuss the following topics:

- Introduction to Python
- Setup Installation in Windows for Jupyter Notebook
- Insights of Jupyter Notebook

- Demo program using Jupyter Notebook
- Introduction to Data Science Libraries in Python

Objectives

This chapter will discuss the essential aspects of setting up a conducive programming environment for Python, emphasizing the significance of an IDE. The discussion commences with the installation process for the Jupyter Notebook on a Windows platform, guiding learners through the setup and providing insights into its functionality. Subsequently, the focus shifts to installing **Visual Studio Code** (**VSCode**) for Python development, exploring its usage for code development. The chapter will conclude with an introduction to fundamental data science libraries in Python, laying the groundwork for learners to leverage these powerful tools in their programming journey.

Introduction to Python

Before we explore this topic, it is essential to set up our development environment. In this chapter, we will guide you through installing Python on your system, ensuring you have everything you need to start writing and executing Python code. Python is an open-source, high-level programming language renowned for its simplicity, readability, and extensive support for various programming paradigms. Whether you are a beginner taking your first steps into programming or an experienced developer seeking a powerful tool for web development, data analysis, machine learning, or scientific computing, Python has something to offer. To begin harnessing the power of Python, the first step is to install the Python interpreter on your computer. The Python interpreter is the core component that executes Python code and provides access to the vast array of libraries and tools available in the Python ecosystem. In this chapter, we will walk you through the installation process for Python on Windows operating systems, including *Windows*, *macOS,* or *Linux OS*.

The steps for installing Python on Windows are mentioned below:

1. Visit the Python website: **https://www.python.org/downloads/** to download the latest version of Python. Here, we have downloaded the latest Python version 3.12.2.
2. We use 64-bit Windows OS and will click or execute the Python installer file.
3. The following window will pop out, and the user can choose **Install Now** or **Customize** installation. Here, we will be selecting the **Install Now** option. Also, remember initially, there are two unchecked checkboxes which are:
    a. Use admin privileges when installing `py.exe`.
    b. Add `python.exe` to `PATH`.
4. Use admin privileges when installing `py.exe`. This option grants the Python installer administrative privileges during installation. It is essential for installing

Python in system-wide directories or when installing packages that demand elevated permissions. Enabling this ensures a smooth installation process without encountering permission-related issues. This option can also be used to change the Python installation folder.

5. Add **`python.exe`** to **`PATH`**. By selecting this option, the Python installer adds the directory containing **`python.exe`** to the **`PATH`** environment variable. This inclusion allows easy access to Python commands from any command prompt or terminal window without specifying the full path to the Python executable. It streamlines the usage of Python across the system, enhancing convenience for running Python scripts and commands.

The steps for installing Python on macOS are mentioned below:

1. **Visit the Python website**: Go to **https://www.python.org/downloads/** to download the latest version of Python for macOS.
2. **Run the installer**: Open the downloaded **`.pkg`** file and follow the instructions in the installation wizard.
3. **Verify the installation**: Open a terminal and type **`python3 --version`** to confirm that Python has been installed correctly.

The steps for installing Python on Linux are mentioned below:

1. **Use the package manager**: Most Linux distributions come with Python pre-installed. However, if you need to install or upgrade Python, you can use the package manager. For example, on Ubuntu, you can use the following commands:

   ```
   sudo apt update
   ```

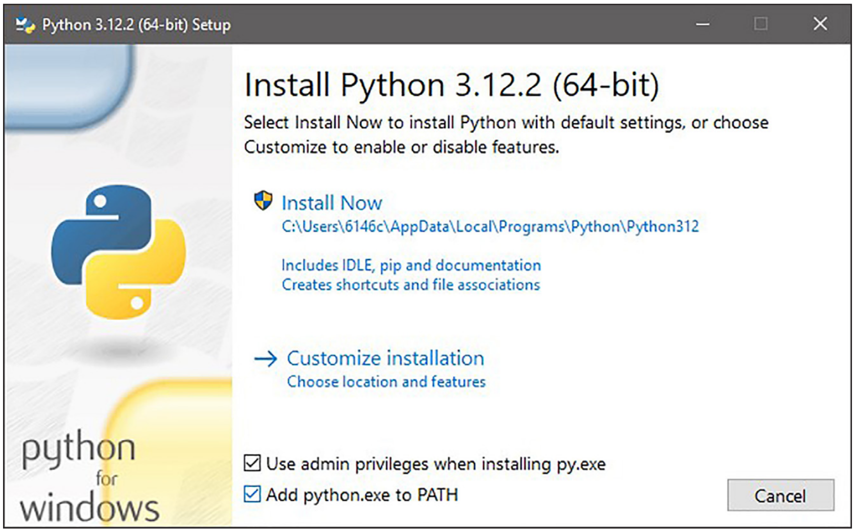
   ```
   sudo apt install python3
   ```

2. **Verify the installation**: Open a terminal and type **`python3 --version`** to confirm that Python has been installed correctly.

We have used Python version 3.11.4 in this book, which means the micro release of Python 3.11 may contain bug fixes and minor enhancements compared to earlier micro releases within the Python 3.11 series. But to show you all the installation steps, we will be showing you Python version 3.12.2, which is the latest version as of 6 February 2024. By the end of this chapter, readers will have a fully functional Python environment set up on their system, ready to embark on their programming journey. Whether you are a student, professional, or hobbyist, Python offers a welcoming and intuitive platform for turning your ideas into reality.

This revision includes installation instructions for Windows, macOS, and Linux users, ensuring that the content is comprehensive and helpful for all readers.

In most cases, it is recommended that you leave these checkboxes checked to ensure a smooth installation experience and convenient Python usage on your system. Refer to the following figure for a better understanding:



Install Python 3.12.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\6146c\AppData\Local\Programs\Python\Python312

Includes IDLE, pip and documentation
Creates shortcuts and file associations

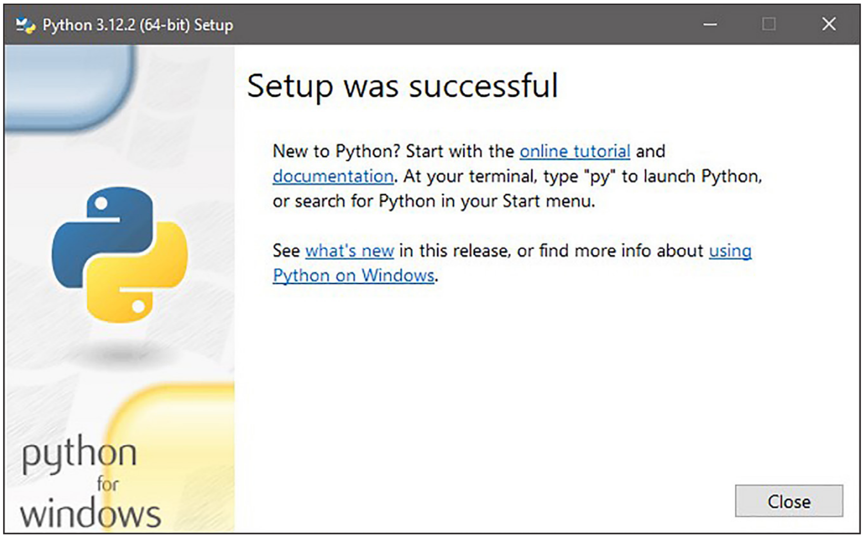
→ Customize installation
Choose location and features

☑ Use admin privileges when installing py.exe
☑ Add python.exe to PATH

Figure 1.1: Python installation on running the Python executable

Refer to the following steps for a better understanding:

1. We will be selecting the **Install Now** option as a recommended option where the default installation path will be **`C:\Users\[user]\AppData\Local\Programs\Python\Python[version]`** for the existing user, which will include the IDLE, pip, and documentation and thus create shortcuts and file associations as mentioned in *Figure 1.1*.
2. Once the installation has been done, the following image will pop out, as displayed in *Figure 1.2*, where the user can close the image.



Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

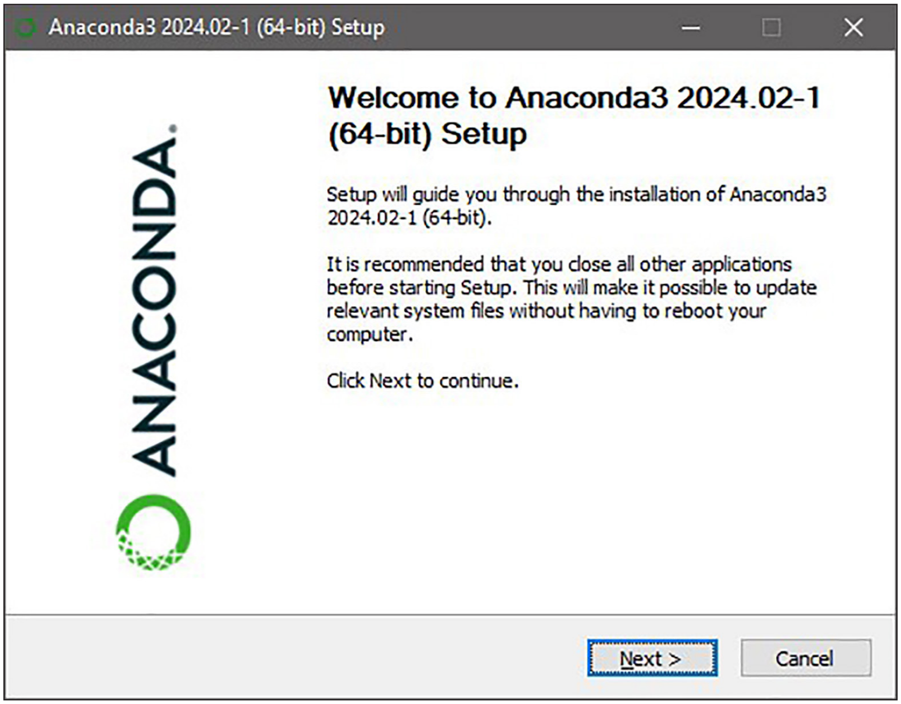
Figure 1.2: Image depicting the successful installation of Python version

Once Python is successfully installed, the user can view the Python version using IDLE (Python 3.12 64-bit), which is installed and can be searched in Windows apps. Python's built-in IDE will be opened when IDLE is run. Another way is that the user can navigate to the directory where Python is installed on the system and double-click `python.exe`.

Setup installation in Windows for Jupyter Notebook

Now, we shall view the steps of installing Anaconda on Windows OS:

1. First, we will download the Anaconda installer by visiting the website **https://www.anaconda.com/download**. Here, we have downloaded Anaconda3 2024.02-1.
2. Once downloaded, we will run this installer file, and we are just getting started with the pop-up of the following image file, as shown in *Figure 1.3*:



Welcome to Anaconda3 2024.02-1 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2024.02-1 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 1.3: *Image depicting Step-1 of getting started of Anaconda3*

3. On clicking the **Next** button of *Figure 1.3*, read the **License Agreement** as shown in *Figure 1.4:*