

**Python**

**Data Visualization**

**Essentials Guide**

*****Become a Data Visualization Expert by Building Strong Proficiency in Pandas, Matplotlib, Seaborn, Plotly, Numpy, and Bokeh*

****Kalilur Rahman**

****

www.bpbonline.com

**FIRST EDITION 2021**

**Copyright © BPB Publications, India**

**ISBN: 978-93-91030-063**

All Rights Reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means or stored in a database or retrieval system, without the prior written permission of the publisher with the exception to the program listings which may be entered, stored and executed in a computer system, but they can not be reproduced by the means of publication, photocopy, recording, or by any electronic and mechanical means.

**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.

**Distributors:**

**BPB PUBLICATIONS**

20, Ansari Road, Darya Ganj

New Delhi-110002

Ph: 23254990/23254991

**MICRO MEDIA**

Shop No. 5, Mahendra Chambers,

150 DN Rd. Next to Capital Cinema,

V.T. (C.S.T.) Station, MUMBAI-400 001

Ph: 22078296/22078297

**DECCAN AGENCIES**

4-3-329, Bank Street, Hyderabad-500195

Ph: 24756967/24756400

**BPB BOOK CENTRE** 376 Old Lajpat Rai Market, Delhi-110006

Ph: 23861747



Published by Manish Jain for BPB Publications, 20 Ansari Road, Darya Ganj, New Delhi-110002 and Printed by him at Repro India Ltd, Mumbai

www.bpbonline.com

**Dedicated to**

*My family and teachers*

*My wonderful spouse and the two angelic daughters who make me a complete human. A special dedication to all great mentors, teachers, professionals, and all the wonderful innovators, authors, and leaders who make this world a better place!*

**About the Author**

**Kalilur Rahman** has a Master's Degree in Business Administration, preceded by an Engineering Degree in Computer Science and over two decades of experience in software development, testing, program/ project management, and management consultancy. Kalilur has been a developer, designer, technical architect, test program manager, delivery unit head, IT services, and COE/factory services head of varying complexity across telecommunications, life sciences, retail, and healthcare industries. Kalilur has advised CXO-level leaders in market-leading firms for testing, business, and technology transformation programs. As a thought-leader, Kalilur is a regular invitee at many industry events across technology and domains.

Kalilur is a firm believer in "knowledge is power" and is passionate about writing and sharing his knowledge. Kalilur is an active member of various technical and professional forums and contributed to internal improvement initiatives in the organizations he worked for. Kalilur has varied interests includes technology, testing in the digital world, artificial intelligence, machine learning, DevOps, continuous delivery, agile, mobility, IoT, and analytics. He is a regular contributor at LinkedIn – a site for professionals and has over 800,000+ followers at the time of publishing. He has published over 200 articles across LinkedIn, DevOps.Com, and other leading magazines.

Kalilur is also an active quizzing enthusiast who participates and contributes at corporate level quizzing at competitive and information levels.

**Acknowledgement**

First and foremost, I would like to thank my family for supporting me in this journey to write this book. I would like to thank everyone at BPB Publications for giving me this opportunity to publish this book and helping me keep focused on the goal, and supporting me along the way.

I would also like to thank my colleagues and extended professional community for giving me endless support and resources to write this book. My learning from all the teachers (upwards, peers, subordinates, and industry experts) has been phenomenal and helpful in quenching a little bit of my thirst for knowledge and sharing the same with extended readers.

I would like to thank all the life-long learners like me who have an insatiable thirst for knowledge that permits people like me to embark on a journey of knowledge sharing and wisdom searching and making ourselves better.

I would be missing my duty if I did not thank all the wonderful people at BPB publications who made this book a reality. This includes the management team (Nrip Jain and others), the acquisition Editor, who persistently reached out to me to get this book going, the content editor who was very methodical, meticulous, friendly, and strict to make sure that tasks were done on time and for brilliant inputs to chisel the book for the first draft. My wonderful technical editor's technical prowess and attention to detail made this a better book to read. The copyediting team did a fantastic job in building the final version of the book, and the entire marketing and management team from BPB made a wonderful effort to get the best out of this book.

Lastly, I would like to thank my critics, my mentors, and my guides. Without their criticism, feedback and inputs, I would never be able to write this book.

*- Kalilur Rahman*

**Preface**

Over the past Techade (a NASSCOM term for the decade 2011-20), data science has been very popular and has a key focus area for technologists. Leading technology giants, massively successful startups, and software product engineering companies have leveraged the power of data science to build great products and services. One of the key elements for the success of data science and big data analytics is the growth in the tools and features to make complex tasks simple and the growth in computing infrastructure. Data visualization is one key concept used for various core aspects of data science in a very intelligent and simple way.

Python has opened the doors for making data visualization simple by exponentiating resourceful libraries to automate various visualization needs. Complexities such as the need to visualize complex data sets and data types are addressed by various libraries efficiently using Python. With an easy-to adapt framework and architecture, Python automation offers the capability to write good visualization with a few lines of code. Python offers several benefits when it comes to data visualization.

The primary aim of this book is to create a scholastically simple and easy to-use guide that spotlights the core concepts of data visualization using Python. This book can be used to understand the data visualization concepts easily to prepare the professional reading this book to become ready to do their visualization on various datasets. This book contains many examples to showcase a particular concept and key principles that may get asked as a potential interview question. This book will be a good pilot to learn the basic concepts of data visualization using Python. This book is divided into ten chapters, and it provides a detailed description of the core concepts of data visualization.

**Chapter 1** introduces the concepts of data visualization. It covers data visualization, key elements of data visualization, and the importance of using test automation.

**Chapter 2** introduces the power of visual storytelling, covers some examples of powerful data visualization, reinforces the benefits of data visualization, and some starter recommendations on various resources and tools to use.

**Chapter 3** covers the elements of data visualization and the tools available. It covers different types of charts and graphs used in data visualization, a recommendation on choosing the right charts, different methods to consider for choosing different visualization elements, and suggested implementation approaches. It also covers various data visualization tools available in the market to consider.

**Chapter 4** covers the hands-on examples using the matplotlib library. It covers an introduction to the wildly popular matplotlib library, covers the introduction to the library's architecture, and covers 50+ examples for data visualization. It also gives a list of matplotlib resources for use.

**Chapter 5** covers hands-on examples using NumPy and Pandas libraries. It covers an introduction to Pandas plotting and examples and case studies of various visualization elements using Pandas plotting functions, modules, and toolkits. It also gives a list of Pandas resources for use.

**Chapter 6** covers hands-on examples using the Seaborn library. It covers an introduction to the Seaborn library. Using different datasets examples and case studies of various visualization elements of Seaborn is showcased. It also gives a list of Seaborn resources for use.

**Chapter 7** covers hands-on examples using the Bokeh library. It covers an introduction to the Bokeh library, followed by examples and resources for Bokeh.

**Chapter 8** covers hands-on examples using plot.ly, folium, MPLFinance. This chapter has case studies for Plot.ly covering interactive visualization capabilities, case studies using folium for geographic mapping visualization, and MPLFinance for stock market data visualization of some financial charts.

**Chapter 9** covers case studies and hands-on examples and exercises for the reader to try out. This chapter has examples to work out using all libraries covered (matplotlib, seaborn, bokeh, plot.ly. folium) and Altair. This

chapter is a complete hands-on guide for the readers to apply the learning from the first eight chapters.

**Downloading the code**

**bundle and coloured images:**

Please follow the link to download the

***Code Bundle*** and the ***Coloured Images*** of the book:

**https://rebrand.ly/h9lvo5w**

**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.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at **www.bpbonline.com** and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at **business@bpbonline.com** for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

**BPB is searching for authors like you**

If you're interested in becoming an author for BPB, please visit **www.bpbonline.com** and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight 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.

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

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

**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**.

**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**.

**Table of Contents**

**1. Introduction to Data Visualization**

Structure

Objective

What is data visualization?

*Brilliant use of data visualization in history*

Key elements of data visualization

*Elements of data visualization*

*Strategy*

*Story*

*Style*

*Structure*

*Data*

*User*

Importance of data visualization

Conclusion

Questions

**2. Why Data Visualization?**

Structure

Objective

The power of visual storytelling

Good examples of data visualization

Benefits of visualization

Recommendations and resources

Conclusion

Questions

**3. Various Data Visualization Elements and Tools** Structure

Objectives

Different types of charts and graphs used in data visualization

*Other types of charts and diagrams used for visualization* Different methods for selection of the right data visualization elements Grouping and categorization of data visualization tools

Software tools and libraries available for data visualization Conclusion

Questions

**4. Using Matplotlib with Python**

Structure

Objective

Introduction to Matplotlib

The definition of figure, plots, subplots, axes, ticks, and legends Matplotlib plotting functions

*Plotting functions*

*Subplot functions*

*Coloring functions*

*Config functions*

Matplotlib modules

Matplotlib toolkits

Examples of various types of charts

*Line plot*

*Exercise 4-3*

*Bar plots*

*Scatter plots*

*Histogram plot*

Mid-chapter exercise

*Box plots*

*An exploratory question:*

*Pie charts*

*Donut/Doughnut charts*

*Area charts*

*Matshow*

*Violin plot*

*Treemap charts*

*Saving a file in Matplotlib*

*Annotating a plot*

Exercises and Matplotlib resources

*Example: Use of subplots and different charts in a single visualization*

*Example: Simple use of Error Bars – Using a horizontal bar chart and a standard error bar*

*Example 4-27: Code to use error bars as a standalone feature Example 4-28: Use of log charts in Matplotlib*

*Exercise 4-29: Contour plots on various mathematical equations Exercise 4-30: An example of a comparison of pseudocolor, contour, and filled contour chart for a common dataset*

*Exercise 4-31: An example of a quiver plot using arrows Exercise 4-32: An example of a lollipop plot*

*Exercise 4-33: An example of 3D surface plots*

*End of the chapter exercise*

*Exercise 4-34: A stock market example*

Matplotlib resources

Conclusion

Questions

**5. Using Pandas for Plotting**

Structure

Objective

Introduction to Pandas plotting

*Pandas features and benefits*

Pandas plotting functions

*Dataframe.plot*

*Pandas.plotting functions*

Pandas modules and extensions

*Pandas extensions*

Examples for various types of charts using Pandas

*Bar charts*

*Horizontal bar chart*

*Pie chart*

*Scatter plot*

*Exercises 5-10 to 5-40 – An exploration of the stock market using Pandas charts*

*Exercise 5-13: Building a scatter matrix from the dataset Exercise 5-14: Generating a stock returns graph*

*Exercise 5-15*

*Exercise 5-16: Histogram of all stocks*

*Exercise 5-17: Individual line plot of all stocks*

*Exercise 5-18 - Generating a simple HexBin chart comparing two stocks*

*Exercise 5-19: Subplots of HexBin charts comparing stocks Exercise 5-20: Generating a density chart for stock values Exercise 5-21: Bootstrap plot for stocks*

*Exercise 5-22: Autocorrelation plot for a stock*

*Exercise 5-23: Question – What will be the outcome? Exercise 5-24: Box plots for stocks*

*Exercise 5-25: Box plots for stock returns*

*Exercise 5-26: Lag plot for stocks*

*Exercise 5-27: Lag plot for stock returns*

*Exercise 5-28: Lag plot for stock returns*

*Exercise 5-29: Calculating the total returns of the stocks Exercise 5-30: Visualizing some area charts*

*Exercise 5-31: Visualizing some bar charts*

*Exercise 5-32: Using a table as a legend*

*Exercise 5-33: Using horizontal bar charts*

*Exercise 5-34: Using parallel coordinates charts*

*Exercise 5-35: Using RadViz charts*

*Exercise 5-36: Question – scatter matrix*

*Exercise 5-37: Using secondary axis in a line chart Exercise 5-38: Using secondary axis in a bar chart Exercise 5-39: Andrews curves using subplots*

*Exercise 5-40 to 5-48: Chapter exercises*

*Exercise 5-40: Question – Random HexBin chart exercise Exercise 5-41 to 5-48: Question on multiple charts Case study 3: Analyzing the Titanic passenger dataset Exercise 5-49: Data exploration and a histogram Exercise 5-50: Histogram of two parameters*

*Exercise 5-51: Histogram of two parameters*

*Exercise 5-52 to 5-68: Further analysis and visualization of the Titanic dataset*

*Exercise 5-69 to 5-71 Generating map outline using scatter plot using latitude/longitude and population data*

*Exercise 5-69: Creation of an approximate outline of India map using a scatter plot*

*Exercise 5-70 and 5-71: Creating an approximate outline of the world and US maps using a scatter plot*

Pandas resources

Conclusion

Questions

**6. Using Seaborn for Visualization**

Structure

Objective

Introduction to Seaborn visualization

Seaborn features and benefits

Examples and case studies for various types of visualizations using Seaborn

*Exercise 6-1: Plotting stock market analysis data*

*Exercise 6-2: Trying Seaborn set\_theme(), color\_codes, despline(), and barplot()*

*Exercise 6-3: Categorical data plotting – boxplot(), violinplot(), and boxenplot()*

*Exercise 6-4: Categorical data plotting – catplot() with the point, bar, strip, and swarm plot*

*Exercise 6-5: Distribution of data plotting – distplot() kernel density estimate*

*Exercise 6-6: Distribution of data plotting – barplot() with NumPy filter functions*

*Exercise 6-7: Simple jointplot() with HexBin option*

*Exercise 6-8: Simple pairplot()*

*Exercise 6-9: A pairplot with the Titanic dataset*

*Exercise 6-10: A pair grid with the Titanic dataset*

*Exercise 6-11: A pair plot with the Titanic sub dataset Exercise 6-12: Application of various plots for real-time stock visualization*

*Exercise 6-13 to 6-20: Application 2 - Data visualization of soccer team rankings using Seaborn*

*Exercise 6-13: A simple pair plot with the soccer team rankings dataset*

*Exercise 6-14: Different types of jointplots() on the soccer rankings dataset*

*Exercise 6-15: Different types of relplots() with the soccer team rankings dataset*

*Exercise 6-16: A clustermap() with the soccer team rankings dataset*

*Exercise 6-17: Write code to create a PairGrid using Soccer team rankings dataset to create Scatter/KDE and histogram across the grid*

*Exercise 6-18: Write code for a simple pairplot() for one of the leagues with a hue on the team name*

*Exercise 6-19: Write a program to create a PairGrid for English Premier League (Barclays Premier League)*

*Exercise 6-20: Simple relationship plot between two variables on the soccer dataset*

*Exercise 6-21: Simple relationship plot between two variables on the Titanic dataset*

*Exercise 6-22: Use of distribution plots*

*Exercise 6-23: Use of histogram and KDE plots*

*Exercise 6-24: Use of Matrix plots – Heatmap()*

*Exercise 6-25: Write a program to generate a heatmap on the stock market dataset*

*Exercise 6-26: Write a program to create jointplot() on the Titanic and soccer datasets*

*Exercise 6-27: Boxenplot() categorical plotting on Titanic dataset Exercise 6-28: Regression plots*

*Exercise 6-29: Write a program to create a jointplot() on the soccer dataset and kdeplot() on the Titanic dataset*

*Exercise 6-30: A PairGrid on the soccer dataset*

*Exercise 6-32: Write a program to create a PairGrid() on the soccer dataset*

Seaborn resources

Conclusion

Questions

**7. Using Bokeh with Python**

Structure

Objective

Introduction to Bokeh visualization

Introduction to Bokeh visualization

*Bokeh API modules and functions*

Examples and case studies for various types of visualizations using Bokeh

*A simple scatter chart*

*Line plot with a square pin marker*

*Bar chart*

*A simple mathematical formula plot*

*Use of patches*

*Use of grid plots and tool selects*

*Simple mathematical plot*

*Use of data for a financial plotting*

*Multiple bar charts with the use of dodge*

*Dynamic line plot*

*Scatter plot*

*Use of colormap, colorbar, and linear colormap for plotting a scatter plot*

*Histogram plot using Bokeh*

*Pie and donut charts in Bokeh*

*Pie chart code*

*Donut chart code*

*Area charts*

*Scatter plot to build a map outline*

*Hex tile plot*

*Dynamic plotting using widgets in Bokeh*

Bokeh case study

*Bar chart and difference in data*

Bokeh – additional resources

Conclusion

Questions

**8. Using Plotly, Folium, and Other Tools for Visualization** Structure

Objective

Introduction to other popular libraries – Plotly, Folium, MPLFinance Examples for various types of visualizations using Plotly *Plotly chart types*

*Simple Plotly scatter plot*

*Simple mathematical plot*

*Line plot using the gapminder() dataset*

*Scatter plot using markers and size*

*Pie chart and donut charts using the gapminder dataset Use of bar charts in Plotly*

*3D Scatter chart and 3D line chart*

*Case study - Use of Plotly for the COVID-19 dataset analysis Plotly animation*

*Scatter matrix plotting capability of Plotly*

*Treemap using Plotly*

Examples for various types of geographic visualizations using Folium *folium features*

*Simple world map*

*Folium heatmap with time animation*

*Simple Heatmap*

Examples for various types of stock and finance data visualizations using MPLFinance

Other examples

Resources

Conclusion

**9. Hands-on Visualization Exercises, Case Studies, and Further Resources**

Structure

Objective

Case studies and examples using Seaborn

*Case studies and examples using Bokeh*

Case studies and examples using Plotly

*Case studies and examples using Folium*

Folium - more choropleth examples

Air travel data case study - Use Bokeh, Matplotlib, Plotly, or any choice Open case study – Using Altair

*Recommended exercises to try out*

Solution file

Resources

Conclusion

**Index**

**CHAPTER 1**

**Introduction to Data Visualization**

**"Human visual perception is grounded on a single set of biological and psychological principles, regardless of culture. Cultural differences should be taken into account but be aware that there are common foundations for what we do."**

*– Alberto Cairo*

**D**

**ata Visualization** and data storytelling have taken the world by storm. Visualization skills are one of the hot skills in the market. Like the introduction, the focus of this book is to give an introduction to data visualization using Python as a primary tool of choice. Specifically, coming to the objective of this chapter, the idea is to introduce data visualization and the difference between various aspects of analysis and the importance of data visualization to the world of business analysis, product management, and its correlation. The importance of data visualization is so big that it determines the winners and losers. Some of the leading technology companies are successful due to their ability to derive insights into their customers' data, products, and services. Some of the large companies with a goldmine are data are not as successful as they should be due to the lack of ability to generate actionable insights.

**Structure**

In this chapter, we will cover the following topics:

What is data visualization

Key elements of data visualization

Importance of data visualization

**Objective**

This chapter aims at giving a good amount of introduction to what data visualization is about, why data visualization is important, how it evolved over time, and its key elements. This chapter will set the context for approaching data visualization from atechnology solution point of view using Python later in the chapters.

**Data visualization** is a visual art of storytelling with an intent to share insights with a meaningful purpose. Data visualization leverages graphical elements such as graphs, charts, maps, and other elements to produce a meaningful graphical representation of data and information. It is a powerful way to share insights on trends, patterns, and outliers in a set of data. The users can analyze the patterns and gain an insight into the data shared. It can be said that visualization is an art due to the creative aspects involved. Data visualization is a very powerful concept in today's world and an important skill to imbibe to succeed.

**What is data visualization?**

Human brains are trained to spot patterns through our experiential learning throughout life. There is a popular adage - *A picture is worth a thousand words*. Our eyes are attracted to patterns and colors. In combination with the brain's cognitive abilities, we are attracted by visuals that make an impact. It could be an image, scenery, or an image in a movie or a TV advertisement.

**Visualization** is a process that transforms the representation of real data from something into something meaningful in a visual representation. The key is meaningful rendering in a simple visualization, even for complex data. Similar to the quote, *a picture is worth a thousand words*, a good visualization tells a story simply and efficiently, like a good painting. It is visual art that can be used as a powerful storytelling tool.

**In literature, I can't say that story A is better than poem B; I have to compare stories with stories and poems with poems, despite being all literature. The same applies to data visualization.**

*- Jorge Camões*

Data visualization, in a metaphorical way, is one way to leverage the visual art of storytelling. Visualization is an intent to share insights with a meaningful purpose. Data visualization leverages graphical elements such as graphs, charts, maps, and other elements to produce a meaningful graphical representation of data and information. It is a powerful way to share insights on trends, patterns, and outliers in a set of data. The users can analyze the patterns and gain an insight into the data shared. It can be said that visualization is an art due to the creative aspects involved.

Data visualization is both an art (visuals) and science (the method of rendering the data) combined. Data visualization can be leveraged to display insights of both quantitative and qualitative data under analysis1. The scientific part of data visualization is done using software or libraries available for rendering graphical visualization. This book is primarily dedicated to this aspect, focusing on a particular language we've chosen – Python.

The French cave paintings, at Chauvet-Pont-d'Arc Cave in France, albeit an art form, showcase some inspiring elements of visualization. Some of the themes used in data visualization were, in a way, let's say, inspired by some of the visual art forms that followed for years. It progressed further with abbreviations used for tax notifications by governments to simple graphs of a line graph or a bar chart before graduating to mind-boggling real-time interactive visualization.

**Brilliant use of data visualization in history**

Some of the greatest visualization examples include the following:

1. Visualization of the cholera deaths by John Snow, known as the father of epidemiology, is a study of disease and patterns to identify measures to solve the issues. He visualized the cholera deaths for a London borough. While visualizing the outbreak in the city, he noted that the number of deaths at a particular street (Broad Street) near a water pump was high. This led to an insight that cholera was caused by germ-contaminated water than particles in the air. This changed the course of medicine and treatments for outbreaks.

2. The brilliant use of wonderful data visualization by Florence Nightingale to record the causes of mortality during the Crimean war. Her fact-oriented visualizations proved that more soldiers died due to infections than that of actual fatality in the war. Her visualizations proved the power of inference of data. Data visualizations thrive on the power of insights and inference, and Nightingale's visualization brought the idea of a single picture being more powerful than thousand words to the fore. Florence Nightingale also produced other data visualization charts to prove a point to the government, healthcare professionals, and the public that sanitation is key for healthy lives. Florence Nightingale, also known as "The Lady with the Lamp," was a pioneering icon in statistics and data visualization.

Data visualization has transformed many organizations to become wildly successful and has helped governments make decisions to improve the lives of the citizens. Some of the data used are sales and profit numbers, market coverage, employee productivity, etc. It could be budget and revenue figures, health indicators for citizens, employment data, and education data to make policy decisions for governments. For humans, one major use case of impactful contribution by data visualization is the efficient usage to expand the average lifespan.

By helping the healthcare professionals to do the right type of diagnosis and analyzing to understand the patterns and outliers By focusing on statistically important aspects to build procedures, discover and develop medicines, and choose treatment

By giving an insight into the trends, progress, and to make an informed decision for the betterment of business

Data visualization is a powerful way to tell a visual story that can help determine outliers, patterns, trends, and correlations of data available and make meaningful decisions.

**Key elements of data visualization**

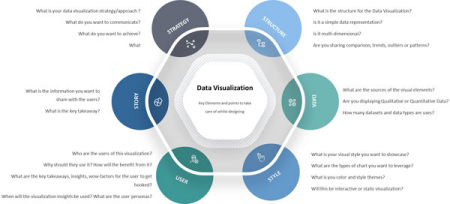
In the 21st century, data visualization has picked a lot of momentum with the advent of increased use of artificial intelligence and data science. The

use of data visualization for research and development, education, and commercial usage have expanded exponentially using interactive dashboards, infographics, and other data rendering tools. It is used in every aspect of our daily life. It is a lot easier to generate powerful visual stories, and I see at least 5 to 10 data visualization elements daily.

One of the keys to the success of the popularity is the evolution of data visualization as an art and science discipline. If you take the examples of visualizations by Florence Nightingale and John Snow, both took a considerable amount of manual analysis and attention to details and great application of key data visualization elements to make them ground breaking. There are many definitions of key elements that vary from designer-to-designer and author-to-author. From a simplicity standpoint, let us see some of the key elements of data visualization.

**Elements of data visualization**

There are plenty of guides available covering the key elements and themes to be considered for effective data visualization. We shall cover some of the essential elements to focus on and consider while designing data visualization. Let us see the key elements for data visualization in a diagram. We can call this *DUSSSS – Data, User, Strategy, Structure, Style, and Story.*

*****Figure 1.1:*** *Elements of data visualization*

*Figure 1.1* shows the key elements in a simple visualization in the form of a pictorial. We shall cover each of the elements in detail. We shall explore each element in a question-and-answer format. At a high level, the six key elements to focus on are:

**Strategy**: What is your data visualization strategy?

**Structure**: How are you planning to structure your story? **Data**: What type of data are you planning to use? How many datasets are you planning to use?

**Style**: A key element on your visualization style, choice of visualization elements such as graphs and charts, choice of colors and other visual elements, use of qualitative and quantitative information to convey a message

**User**: The key to the success of the data visualization exercise. Who are your users? Why should they be using your product? What is the key takeaway for them?

**Story**: Most important aspect of the exercise. What are you trying to convey, what would the key insights, messages, actions, inspirations they can take away to implement actions?

Let us delve into a bit more detail on these elements.

**Strategy**

Having a good strategy for your visualization exercise is very important. This is because the visualization outcome is purely based on the data being used. Like bad content or a theme could derail a movie or an advertisement, bad data can result in poor outcomes in story, elegance, insights, etc. Having a good strategy is important for a visualization exercise. This includes data strategy, design strategy in terms of user persona and visual elements, etc.

Having good data capturing, data extracting, data cleansing, data integration strategy is very important. This strategy is especially important for planning interactive, real-time, update-oriented

dashboards and data visualization. There should be a data strategy for the visualization exercise.

Another element to consider is the user experience and design thinking strategy to address the needs and wants of the users. Using a persona based design of visual elements can help in designing better visualization elements.

As one size fits all does not exist, a designer bias can be avoided by taking the user requirements, and user needs into consideration through the empathy-based user-centric design of elements.

Design elements, style elements, visual themes, templates, messaging, colors, form factors, devices, and gesture-based themes and actions can all be thought in advance.

Having a clear structure, simplicity, better visibility, and consistency in design could be thought through before the design is done.

**Story**

Data visualization can be used for two purposes when it comes to stories and messages for the users. The best stories (or visualizations) are not used only to share information or create a user reaction. They can influence and inspire people.

Some of the advertisements on TV have very strong visuals that tell, inspire, motivate and connect with the viewers at a deep level. Similarly, a good data visualization output should connect with the users. It should cater to the needs of the user. It should address a specific need or wants they have while viewing the data visualization product.

A simple run-of-the-mill output may not appeal to everyone. This is one reason why the current set of successful data visualization tools gives the end-users many choices to play around with and customize the data visualization outputs to suit their needs.

We need to be very clear in what we want to communicate and how the users and viewers will interpret it.

A key takeaway message – be it an executive summary, call-for action, insights should be included in the story getting told

**Style**

Style is very important for a data visualization element. This is similar to branding exercises carried out by various firms. This gives a unique association of the product with the style and a very consistent expectation, and a potential wow factor for the users. The use of essential styling themes will help convey the message in a brilliant and highly influential manner to the users/viewers. There are numerous techniques available for styling. Some of the tips are:

Have an aesthetic element that hooks the users. The style should be impressive and beautiful but simple to connect with the users quickly Have a simple, easy to understand and decipher structure rather than a complex one. You can choose to have an intricate style depending on the audience.

Have a better, simple, effective, and efficient design element that is used in a modular fashion across the board

Have a style guide for the following:

The basic structure of the visualization

Text styles – for header, section, axes, messaging, legend, etc. The text used with context will explain the data visualization story in a better manner

Colors

Background colors and images

Colors to be used for data display

Colors for special elements such as maps, density, outliers, highlighting patterns and trends

It is advisable to minimize the variations in different colors, such as dark and lighter shades

It is advisable to use color palettes available in the software or charting libraries

It is better to use standard colors (recommended by most tools and libraries) for ease of use and readability

**Structure**

Data visualization, as mentioned earlier, is an art and a science combined. Every aesthetic art has a structure to it, and so does a scientific experiment, and it's the outcome. Hence having a structure for data visualization is an important aspect in deciding how the data can be presented by determining the structure of data to be used, structure of the data and insights presentation, the structure of data formatting, structure of how data will be presented. The presentation of the final data step of the data storytelling design journey including data collection, curation, format, visualization, and presentation. It would also include the frequency of data collection.

Having a good data visualization structure also corresponds to keeping the user experience, needs, and wants in mind in delivering a design that matters the most.

Structuring the content to be displayed in the visualization is important – such as what, when, how the users see.

Structuring the timing of data collection and visualization, including the update frequency (if it is one-way visualization; and the data refresh in case of an interactive visualization).

Structuring the approach for data analysis (such as exploratory data analysis used in data science), cleansing, the grouping of data, and most importantly about visualization structure and elements–as per the style and strategy adopted.

The structure would also entail the amount of data to be rendered and the depth and breadth of analysis (mainly in the case of interactive visualization) to be performed.

A structured approach helps build data visualizations that will be cherished forever, like the examples of John Snow and Florence Nightingale discussed earlier in the chapter.

**Data**

Most of the successful firms in the world know how to use their data well. For effective data visualization or storytelling with data, you need to have good data, and you need to know how to use them, build a flow, and convey the message. A story can be boring, mediocre, average, exciting, intriguing, inspiring, motivating, depending on how it is told. Similarly, data can also be visualized in the same aspect.

Do you want to tell an exciting story, or do you want to tell a story that no one is interested in? Due to this important aspect, it is advisable to focus more on the data aspects. We have structured data sources such as tables, flat files, etc. We also have unstructured data in the current context. Plans to use them need to be understood and well thought.

It is also important to understand different data types such as:

Qualitative data (data that cannot be measured quantitatively but is subjective data such as gender, exam results (pass/fail), categories (feedback, happiness, temperature groups, height groups, satisfaction, etc.), color, tastes, preferences, interests, etc.)

Qualitative data can also be called categorical data

Qualitative data are primarily non-numerical and text is used most of the times and can be binary-style (yes/no, pass/fail, male/female, blue/brown/black/green, cold/warm/hot, high/medium/low, highly satisfied/satisfied/neutral/dissatisfied/highly dissatisfied etc.)

Qualitative data can be further categorized as *nominal scale data* or *ordinal scale data*

Nominal scale data is typically named or labeled data type mentioned above (male/female, hot/cold, and pass/fail, etc.)

Further classifications include:

Dichotomous – having two possibilities –

Gender - male/female

Response - yes/no

The result - pass/fail

Temperature - hot/cold

Nominal with order –

Coffee temperature - cold/warm/hot

Probability - high/medium/low

Customer satisfaction - highly satisfied/satisfied/neutral/dissatisfied/highly dissatisfied Nominal without order – such as

Colour of eye - blue/black/brown/green

The blood type of a person – O/A/B/AB +ve/-ve

Political inclination – left/liberal/central/right

Type of house – apartment, independent house, villa, trailer, homeless

Gender – male/female/other

Quantitative data are typically numerical, and they are measured with an accuracy such as height, distance, length, width, speed, profits, salary, duration, etc.

Quantitative data is ordinal data that can be of two types – namely discrete and continuous numbers.

Discrete data are typically whole numbers that can't be divided, and they are typically a count of occurrence – This is like many wins, number of houses, number of computers, etc. You can't have a decimal value in a discrete data

Continuous data can be further fine-tuned and broken into subunits within a range

Example – the distance between the office and home – 7.27 KMs, weight – 72.5 kg

Continuous data can be either interval scales or ratio Interval scales are numeric scales where the values between intervals are the same. For example, in some thermometers, the intervals are typically 10 degrees. Similarly, in some types of scale, the intervals are typically in 5 centimeters

Ratio scales are similar to the interval scale except for the fact that zero does not exist. Examples are age or weight, or a normal room temperature of zero does not exist.

Additionally, it is paramount to use the right data and understand the data to be used. It is also important to decide the number of data sources to use. Deciding qualitative vs. quantitative data for visualization and how to represent the data helps. Better handling of data will result in outstanding data visualization results.

Additionally, it is important to explore the data to pick the right strategy to visualize data. This would include steps such as

How to build a model for visualization

Strategy to address the missing data and the exceptions and outliers How to plan and choose the right data elements for analysis How to run an exploratory data analysis

How to plan the visualization approach

How to represent or build a simple data visualization

Let us see the next element from the list to focus on – User.

**User**

A final key element to focus on is the *user*. When we deal with users, it always starts with the end in mind. We can develop better visualization relevant to the users by:

Having a connect – by giving user-centric insights

Giving good insights to the user through the data visualized Motivating the user by connecting them with the data

Giving some actionable insights for the users

Using the elements highlighted above – strategy, story, structure, data, and design, a good programmer doing visualization should understand the audience, have a good framework, use the right tools, and empathize and connect with the user. This will result in the visualization to cater to the

audience's needs better. It will help balance the design better and allow for better outcomes that will certainly imbibe a wow factor!

**Importance of data visualization**

To paraphrase a famous quote – *A picture is worth a thousand words* again, a lot of data and numbers can consume more time from humans than a picture. Data visualization makes humans understand the big picture of big data using a small, impactful visualization.

Visualization makes it easier for humans to detect trends, patterns, correlations, and outliers in a group of data.

A simple and powerful data visualization built with credible data with good analytical modelling can help businesses make quick business decisions.

A good visualization

Does the difficult job of communicating a piece of complex information in a simple, clear, and concise manner to top business leaders in an easy to grasp manner

Can lead the users to focus on actionable insights that need their attention

Provides insights and story to establish a business goal by giving previously unnoticed patterns and correlations

Identifies the gaps, errors and helps businesses make corrective actions for better outcomes

Can help businesses set business goals to drive growth and address any problem areas easily

Helps users identify emerging trends and use them for any actions in the business or personal sphere (such as learning new technology trends, etc.)

Helps businesses such as investment banking and stockbroking to make real-time decisions to make a profit or cut losses

Can help executives run better operations. A good executive dashboard with brilliant data visualizations with actionable

insights helps executives make quicker and better decisions

We shall see more details in upcoming chapters, and a lot of importance and focus areas have been covered in the earlier subsections in this chapter.

**Conclusion**

In this chapter, we covered the introduction to data visualization, its importance, and the key elements to be aware of. We also looked at the key features one needs to be aware of to build data visualization from a storytelling perspective. Good data visualization is like a blockbuster movie or an enthralling ad. It needs a lot of planning, strategy, and messaging skills that will make a huge impact.

We covered the six key elements: strategy, structure, story, data, design, and user. The ability to create a brilliant visualization is a niche skill and helps one propel a career. It is advisable to structure data visualization clearly to get your message across. Adapting the style and content to the occasion and the context relevant to the users is also important. A strategy to the design can be outlined with structured thoughts with end-users in mind before starting the data visualization design. Having a good handle on the data and how to use them is key.

Most importantly, involving the users by highlighting the information that will excite them in the visuals is key. To boost the impact of engagement, it is important to improve the skills to visually convince the audience. It is important to seize the user's attention by removing distractions such as jarring colors, text, and information overload. The user feedback can also be leveraged to strengthen the visualization skills.

Over the next few chapters, we will see how to address the elements discussed in this chapter in a practical manner. The examples provided in this book will be in Python but can be extended to other languages.

**Questions**

1. Why is data visualization important?

2. Why is user experience important while conducting data visualization?

3. Why are design elements important for good data visualization? 4. What are the challenges in dealing with data for data visualization? 5. What are the key benefits of using data visualization?

6. What is the difference between storytelling and data visualization? 7. Why is having a good strategy for data visualization a good thing? 8. What is the need to have a good structure for data visualization? 9. What is minimalism in terms of data visualization? How is it helpful?

10. Why should data visualization have a story element? Why can't we just have a graphic displayed?

11. What is the benefit of giving interactivity to the users? What are the benefits of dynamic data visualization products such as dashboards and applications?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1We shall cover the specifics of different data types a bit later in this book.

**CHAPTER 2**

**Why Data Visualization?**

**"To find signals in data, we must learn to reduce the noise - not just the noise that resides in the data, but also the noise that resides in us. It is nearly impossible for noisy minds to perceive anything but noise in data."**

*― Stephen Few*

W

e covered a high-level introduction to data visualization and key elements to consider and keep in mind while designing data visualization. The objective of this chapter is to showcase some benefits of data visualization and how it leverages the power of storytelling. We shall see some brilliant examples of usage of data visualization that has been very powerful in communication, decision making, and entertaining storytelling, to name a few. This chapter will also give some good resources for further knowledge building.

**Structure**

In this chapter, we will cover the following topics:

The power of visual storytelling

Brilliant examples of data visualization

Benefits of data visualization

Recommendations and resources

**Objective**

This chapter aims at giving an introduction to how powerful visual storytelling is and how good data visualization helps in achieving the same. We will explore some good examples available in the public domain and how benefits were realized. The most important objective of this chapter is

to give a list of good resources to explore and leverage and recommendations to read and upgrade knowledge on visual storytelling, data visualization and become an excellent visual communicator by leveraging data power visualization.

**The power of visual storytelling**

**The power to connect** - Humans need to connect, interpret, learn, understand to debate and accept, and cognize the facts to connect with data or a visual being presented. Most of us connect better with visuals and stories than numbers and text. Hence, good visual storytelling fulfils a basic human need: the eagerness or a hardwired interest for stories.

**The power to derive insights** - Good visualization stories help the audience uncover the messages and stories you are keen to share. They may even uncover some untold stories. A powerful message that conveys untold stories makes the visualization vivid and colorful to remember and understand. Finally, it leaves the audience wanting more information and messages.

**The power to influence** - Good communicators who can tell compelling stories rule the world, and so goes the adage that goes, "Those who tell the stories rule the world." The art of visual storytelling in business is absolutely important as humans get connected at an emotional and psychological level better than at a cognitive level. For example, marketing is no longer a discipline to communicate about the products you sell; but it is about the stories you want to tell and influence.

**The power to inform and make quick decisions** - Human beings are trained by the brain to handle visuals better, quicker, and impactfully as the cognitive effort is lesser than processing text or voice-based information. Author - Ekaterina Walter, in the book "The Power of Visual Storytelling: How to use visuals, videos, and social media for marketing your brand," says that

**"Visuals are processed 60,000 times faster than text by the human brain, and 90% of information transmitted to the brain is visual.**

**Humans evolved over millennia to respond to visual information long before they developed the ability to read text. Images act as shortcuts to the brain: we are visual creatures, and we are programmed to react to visuals more than words."**

Taking A picture is worth a thousand words; a visual story is probably worth more than that. This is why visual information in social media is much more powerful than any media in the world currently due to the reach, impact, and power it possesses. The benefit of good design for visual storytelling is summarized as the *design is about solving and providing elegant solutions. Information desing is about solving information problems* - as per a brilliant quote from McCandless in his wildly popular TED speech.

The "Gapminder'' website – **www.gapminder.org,** the brainchild of Hans Rosling, covers some of the brilliant visualizations delivered. One of the key points outlined by Hans Rosling is that our mind filters what is most dramatic and easy to associate with as the key essence. This means that to connect well to get user attention, content with emotional value add is paramount. Visual connect is much more than facts being passed verbally and is a great connect. Some unerasable memories about movies seen decades ago are an example of this.

There are plenty of examples of compressing tons of information into a single, powerful infographic and outline the power of visual storytelling. One brilliant example is available at **https://www.gapminder.org/tools/** that showcases various trends of the progress of nations in terms of animations, maps, trends, ranks using various types of charts and graphs. While we shall be covering most of the technical aspects of these graphs in the rest of the book through program snippets, it is a good idea to check this tool.

Another powerful summary of how data visualization has evolved and helped over the years can be understood from how the financial summary is shown in famous print media powerhouses such as New York Times and The Wall Street Journal. With the introduction of simple graph/visual elements, they've reduced stock-related pages by 7-8 pages with very powerful indicators. One can visualize stock performance with dots

showing the range and trend for the highs/lows of the stocks. A good visualization can compress 1000s of pages, inputs, billions of data points into a meaningful insight that helps top leaders make big decisions easily.

**Good examples of data visualization**

Now let us see some examples of good data visualization. To showcase this, let us revisit the two examples we covered in *Chapter 1* - The Broad Street cholera outbreak and that of Florence Nightingale's chart. Why did they make a huge impact? They were simple and factual – using good data, represented a lot of data in a single picture, and conveyed a very clear, actionable message. Over the last 20 years, the usage of brilliant information design, computing power, availability of tools has helped in the explosion of multiple ways to render brilliant visualizations.

Let us take an example of a KPI or a metrics dashboard. This sample gives some key metrics. However, the data visualization for business needs could have a myriad of metrics across functions and industries. This dashboard gives some operational statistics for ecommerce websites, such as page views, time spent, visitor locations, etc., that can help the team take action.



***Figure 2.1:*** *A sample dashboard created using data visualization Tools – ecommerce site statistics dashboard*

What we looked at is a simple usage we may see in a business environment.

Let's take another visualization. This example is explained in detail in *Chapter 8* of the book. Is there a possibility of using the words used in a book as a map? We shall check how to do this with the code and explanation later. This code uses Rabindranath Tagore's Gitanjali and a map of India to generate the visualization as in *Figure 2.2*.

***Figure 2.2:*** *A visualization of Tagore's Gitanjali in an India Map using Python*

Finally, another brilliant visualization of all the Oscars awards over the last 90+ years represents a simplified representation graphically. The insights

are very simple for what one needs to have in a picture to win the best actor or actress awards. One intriguing story told is that no sportsperson character wins an award unless the role is a boxer. This visualization can be seen at the following URL - **https://iibawards prod.s3.amazonaws.com/projects/images/000/000/391/page.jpg?**

**1403858126**

The examples given above are just a few of the good examples of brilliant visualization. There are thousands of such brilliant visualizations available. Now let us see some of the benefits of good visualization.

**Benefits of visualization**

We covered some of the benefits in earlier sections in this book. As mentioned earlier in the chapter, the book The Power of Visual Storytelling: How to use visuals, videos, and social media to market your brand says that the human brain processes visuals faster than text (up to 60000 times) and that our brain transmits more visual information than other forms (more than 90%). From the same source, we also infer that the human brain is capable of processing images in a period of fewer than 13 milliseconds.

Let us see some of the benefits of good data visualization.

The first benefit is **rapid/quick decision making** - For rapid decision making, we need to compress the data, narrative, and visuals simplistically, which is aided by data visualization. This results in quick actions being made.

The second benefit is a **better analysis summary** – Good data visualization can share outstanding summary analysis in a very simple manner. It avoids the need to analyze millions of records of data or thousands of pages of text or data. A picture of the data visualization can simplistically tell all the summary.

The third benefit is **better business insights** – One of the main areas where the visualization is used is for the measurements of key business indicators, operational metrics, etc. This gives a better view of the key business focus areas: sales, marketing, finances, operational metrics, and other key parameters.

The fourth benefit is **identifying trends, patterns, and gaps** – Another important benefit of visualization is displaying trends, patterns, and gaps in various visualized parameters. This would help identify the reasons and take actions to improve better outcomes – be it business, operations, or any other areas.

The fifth benefit is **telling a powerful story –** Good data visualization can be leveraged to tell an impactful story in a timeline or based on trends and patterns. This is a very good way to tell stories in a very simple manner.

The sixth benefit is **building a brand image –** Some businesses have built a reputation for telling very powerful stories through their brilliant visualizations. In a way, telling a good story also helps in building an outstanding brand value. There are some great examples of Coca-Cola being attributed to red and attributed color.

The seventh and final benefit is the ability to **telling more with less -** A lot of information and messages can be shared with a simple visualization. This may be impossible to achieve in a narrative or using tons of data. A story can be narrated brilliantly with a single visualization.

**Recommendations and resources**

There are plenty of resources to refer to and benefit from. I am including some of the brilliant resources to read available on the internet.

**Let us start with some brilliant data visualization resources** Information is Beautiful website (**http://informationisbeautiful.com**) Data is Beautiful website (**http://dataisbeautiful.com**)

Data visualization Catalogue (**https://datavizcatalogue.com/index.html**)

Statista website (**http://www.statista.com/https://www.statista.com/chartoftheday/**) Flowing Data (**http://www.flowingdata.com**)

Storytelling with Data website (**http://StoryTellingWithData.com**)

Tableau Public Data visualization Gallery and Datasets (**https://public.tableau.com/**)

QlikView Data visualization Demos - **https://demos.qlik.com/qlikview**

**Excellent Trend Analysis**

Google Trends (**https://trends.google.com**)

**Excellent Data Sources**

Our World in Data (**https://ourworldindata.org/**)

Broad Institute's Datasets – (**http://portals.broadinstitute.org/cgi bin/cancer/datasets.cgi)**

Amazon Web Services – Datasets – **(https://registry.opendata.aws/)** World Health Organization Datasets – (**https://www.who.int/data/gho)**

Pew Research Centre Datasets – (**https://www.pewresearch.org/internet/?post\_type=dataset)** Google Public Datasets – **(https://www.google.com/publicdata/directory)**

US Government's Open Data – (**https://www.data.gov/)**

Europa Open Data – **(http://open-data.europa.eu/en/data/)**

As we coast along with the book, we shall see more resources for our learning.

**Conclusion**

In this chapter, we covered a good amount of introduction to data visualization, its importance, and the key elements to be aware of. We also looked at the key features one needs to be aware of to build data visualization from a storytelling perspective. Good data visualization is like a blockbuster movie or an enthralling ad. It needs many skills, excellent planning, strategy, and messaging skills to make a huge impact.

The six key elements we mentioned in *chapter-1* were strategy, structure, story, data, design, and user. The ability to create a brilliant visualization is a niche skill and helps one propel their career. It is advisable to structure the data visualization clearly to get your message across. Adapting the style and content to the occasion and the context relevant to the users is important. A strategy to the design can be outlined with structured thoughts with end users in mind before starting the data visualization design. Having a good handle on the data and how to use them is key.

Most importantly, involving the users by highlighting the information that will excite them in the visuals is key. To boost engagement, it is important to improve the skills to visually convince the audience. It is important to seize the user's attention by removing distractions such as jarring colors and text and information overload. The user feedback can also be leveraged to strengthen the visualization skills.

**Questions**

1. Why is data visualization important?

2. What are the important needs while doing data visualization? 3. Why are the design elements critical for good data visualization? 4. What are the challenges in dealing with data for data visualization? 5. What are the key benefits of using data visualization?

**CHAPTER 3**

**Various Data Visualization Elements and Tools**

**"By visualizing information, we turn it into a landscape that you can explore with your eyes, a sort of information map. And when you're lost in information, an information map is kind of useful"** *―David McCandless*

I

n the first two chapters, we got introduced to the "what" and "why" aspects of data visualization. In this chapter, we shall broach the topic of how albeit briefly. These topics are covered in detail in the rest of the book. These are the most important aspects and are almost uniform across applications, tools, and libraries to develop data visualization. Understanding the key elements of visualization elements is key to building a successful strategy. A good strategy and approach, and design elements are essential for good storytelling through data visualization, as discussed in the earlier chapters.

**Structure**

In this chapter, we will cover the following topics:

Different types of charts and graphs used in data visualization When to use charts in data visualization

Different methods for selection of the right data visualization elements Suggested implementation approaches using different software and tools available for data visualization

Categorization of data visualization tools for various purposes and business needs

**Objectives**

This chapter aims to give a basic introduction to various data visualization elements such as charts and graphs. We will also touch upon how to choose the right type of element for various needs. We shall complete the chapter to understand the approach through a mind map on approaching the selection of various types of data visualization elements.

**Different types of charts and graphs used in data visualization**

The graphs, charts, and visualizations have come a long way from very simple beginnings. Ever since *William Playfair* published a simple bar graph in 1786, the usage has increased slowly and steadily. We have hundreds of charts to choose from, and innovations continue, and new styles of data visualization charts get created regularly. In this chapter, we shall cover some of the key charts to know (including the ones we shall visualize using Python in the later chapters).

The first question we need to answer while choosing a chart or any data visualization element is – "what is the purpose of the element/chart?" This will allow us to address further queries such as "what are we trying to address, how would it form a part of the story we want to tell," etc. The purpose plays a key role in determining the type of chart we would like to use. A chart helps in achieving the purpose. It can fully satiate the purpose or part of the solution to satiate the purpose with other elements.

A purpose could be to inform about data in a way that is easy to understand. This could be to show comparisons, or to show changes over time, to show relationships between variables, to show organized data visually, show distributions, show geographic data points, financial parameters, key performance indicators (KPIs), trends, the composition of data, the ranking of data, correlations, spatial data, shows a part of a whole set of data, the flow of data, etc. For simplification, we shall use the following types for this book.

**Distribution**: This shows the entire distribution of data, or it could be a count of occurrence of data as well (as in histograms)

**Time-oriented trends**: This shows trends of data movement over time – this could be by second, minutes, hourly, daily, monthly, yearly, etc. **Comparison**: This shows the composition of the data element or to compare two or more set of data of the same type

**Spatial data**: This is to showcase maps and location-specific data to be displayed using various charts

**Flow data**: This is to showcase movement or change of data from one point/position to the next to show how a data element flows in a sequence – such as the flow of funds, immigration data, etc. **Relationship data**: This shows any relationship between two or more sets of data/variables

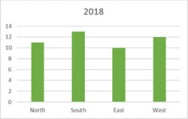
**Part of a whole data**: This shows the composition of data elements that make up the whole data (100%)

**Deviations**: This shows how the data varies from a fixed point of reference to show the trends

**Other types**: This shows financial data charts, KPI charts, word clouds, etc.

*Table 3.1* covers types of charts, their purpose, and where they could be used.

| **Chart Type** | **Purpose / Usage / Description** |
| --- | --- |

Bar chart/ Graph/ Column chart A bar chart is a visual representation of values in horizontal or vertical rectangular bars. The height is 

proportional to the values being represented. Shown

in two axes, one axis shows the element, and the

other axis shows the value of the element (could be

time, company, unit, nation, etc.).

Bar charts can also be combined for multiple values

of an element over time to show the relative

correlation of performance (such as annual revenues

of different divisions within a company/competitor

firms over time.)

Bar charts rendered vertically are also known as

column charts, and horizontal bar charts are referred

to as bar charts in some tools such as Microsoft

Excel.

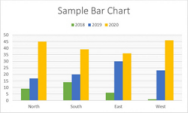
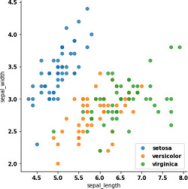
In Python, we have a bar chart and horizontal bar 

chart functions. We shall be covering it in future

chapters.

| Line chart/Graph | The line chart is a two-dimensional plotting of values connected following the order. Values are displayed (or scattered) in an ordered manner and connected. Line charts show the trend of an element in comparison against time.  Line charts can be  A simple line chart (showing value of one element over a reference – such as time)  Multiple line graphs – showing multiple values over a similar reference point – such as stock prices of multiple companies over time (shown in different colors)  Splines – line graph that shows the curved connection of points instead of a straight line Stepped line graph – where connections between points are shown in a step  Line charts are typically used in combination with other types of charts to impact visualization. |
| --- | --- |

Scatter plot A scatter plot is a two-dimensional chart showing the comparison of two variables scattered across two 

axes. The scatter plot is also known as the XY chart

as two variables are scattered across X and Y axes. A

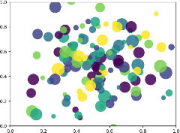
scatter plot can be displayed without connecting

lines or being displayed with smooth curved

connectors or connecting lines. To distinguish

characteristics, a marker can also be used to make it

effective.



| 3D scatter plot | 3D scatter plot is an extension of the scatter plot and adds a third variable to show three dimensions by adding additional axes. An additional axis – Z is added to show the value of the third variable against the two variables compared in a standard scatter plot. |
| --- | --- |
| Bubble chart | A bubble chart is built on a simple scatter plot, so the first two variables can determine the bubble's position on the Y-axis. A third variable represents each data point in a bubble, the value of which determines the size of that point, and the second by the number of data points in that bubble. The second variable determines each point's height and position and the amount of space between them. A bubble chart can be extended to a 3D bubble chart by adding additional axes as well. |

Histogram A histogram is a way to represent the distribution of numerical data elements (mainly statistical) in an

approximate manner. A histogram uses a "bin" or a

"bucket" for a set or range of values to be

distributed. A histogram is discrete and need not be a

contiguous one. Based on the bins and the values of

the data, it can be skewed either to the left or to the

right of the visualization. In a traditional statistical

representation, as per the central limit theorem, data

distribution over a large volume tends to be gaussian.





| Pie chart | A pie chart shows the proportion or percentage of a data element in a circular format. The circular chart is split into various pies based on the value/percentage of the data element to highlight. The pies represent the "part-of-the-whole" data. The overall sum of pies corresponds to the 100% value of the data being visualized.  Pie charts are a very effective tool to show the values of one type of data. They can be further expanded into a pie of pie charts if a particular category of a pie can be shown using the subcategories making the pie (as an example – a pie can be the percentage of the population of a nation, and a pie-of-pie can show the population of states/provinces of a nation chosen to highlight) |
| --- | --- |

Doughnut chart A doughnut (or a donut) chart is an extension of a pie chart. The center part of the doughnut chart is 

empty to showcase additional data/metrics or

expanded compositions of a pie or showcase another

data element. A doughnut chart addresses the

criticism of pie charts that it is difficult to compare

pie charts due to the central area by deemphasizing

the central portion. A donut chart is efficient in using

space and can easily compare charts using the space

effectively. A pie chart is useful for very simple

visualization.

| Area charts | Area charts are used to plot data trends over a while to show how a value is changing. The area charts can be rendered for a data element in a row or a column of a data table such as the Pandas data frame. An area chart can show the part-of-the-whole by stacking the values of various elements making up 100% through a stacked area chart. An area chart can also be shown in a 3D shape. Some good examples can be the GDP summary or population summary of nations and sales by departments over time. |
| --- | --- |
| Box plots | Box plot is a commonly used chart for business, professional aspects and extensively in data science related visualizations. It is used to show the distribution of two or more data elements in a summarized manner. The key part is a box with a line shown at the median value. The area above the box is the upper quartile, and the area below the box being the lower quartile. The outliers are shown outside the box using an extreme line for both highest and lowest values. The number of values is typically not shown (unlike histogram, where we can define the buckets or bins)  Box plots are used typically for two variables, and the wide format of a boxplot is typically used for three or more variables |

Violin plot A violin plot is handy when the number of data elements is very high in number and where a box

plot, histogram, or scatter plot may not showcase

very meaningful insights. Violin plots give a better

visualization of the density of the data elements and

how closely they are interrelated in a distribution.



| Density plot | The density plot is closely related to a histogram and takes one set of numerical values as inputs. The output of a density plot is the display of the distribution of data. The distribution can be in the format of an exponential or a bell curve-like format. The chart can be skewed either to the left or right based on the volume of data for a particular data range. |
| --- | --- |
| Heat maps | A heat map is a tool to show the magnitude of data elements using colors. The intensity (or hue) of the colors is shown in a two-dimensional manner, showing how close the two elements are correlated. To understand the data implication, a heat map is also tabularized with the correlation value. A heat map can also be used in conjunction with other types. An example is using a heat map with a map to potentially show the intensity of a crime or a particular event in various locations displayed in a map. |

Waterfall Chart A waterfall chart is a visual way of showing the effects of sequential, intermediate time, or category

based values cumulatively. The values are positive or

negative. A waterfall chart is a popular chart in

financial budget visualizations and shows how profit

or loss looks over time.



***Table 3.1:*** *Various types of visualization charts*

**Other types of charts and diagrams used for visualization**

Let us see some of the other charts and diagrams used (and sometimes referred to by various names) in a tabular format.

| **Types of Charts and Diagrams used for Visualization** | | |
| --- | --- | --- |
| 2D density Plot | Barcode plot | Bee swarm charts |
| Binary decision diagrams | Box and whisker charts | Bubble map |
| Bullet chart | Bump chart | Candlestick chart |
| Cartogram | Chord diagram | Choropleth map |
| Combo charts | Connect scatter | Connection map |
| Contour map | Control charts | Correlogram |
| Dendrograms | Diverging bar chart | Dot density plots |
| Dot map plots | Dot strip plot | Error bars |
| Fan charts | Flow charts | Function plots |
| Funnel chart | Gantt chart | Gauge charts |
| Graph visualization chart | Grid plot | Grouped bar plots |
| Grouped symbol chart | Hexbin plot | Hierarchy diagrams |
| Hyper tree diagram | Icicle diagram | Kagi chart |
| Kaleidoscope charts | Mandelbrot set chart | Marimekko chart |

Mosaic charts Multi-level pie charts Network chart

| Ohlc chart | Ordered bar chart | Ordered column chart |
| --- | --- | --- |
| Ordered proportional symbol | Org chart | Pareto chart |
| Pert chart | Pictograms | Point and figure chart |
| Polar area chart | Population pyramid chart | Pyramid chart |
| Radar chart | Radial bar chart | Radial column/bar chart |
| Regression fit scatter plot | Sankey diagram | Scatter line combo |
| Seismogram | Space tree charts | Spaghetti plot |
| Sparklines chart | Spider charts | Spiral plot |
| Stacked bar plot | Station map chart | Stem and leaf plot |
| Stock charts | Streamgraph | Sunburst chart |
| Surface plot | Tally chart | Timeline chart |
| Tree chart | Treemaps | Trellis plots |
| Trellis line charts | Venn diagram | Vertical timeline |
| Voronoi chart | Word cloud | Word trees |

***Table 3.2:*** *Types of charts and diagrams used for data visualization*

**Different methods for selection of the right data visualization elements**

As discussed in the previous chapters, the selection of the right visualization element depends on various aspects. One of the major aspects to consider is the purpose and the data type you are using. We can have a logical approach to selecting the right visualization element. The type of chart to visualize may also be dependent on the type of data, the number of variables, and other aspects. Using these important identifiers, we can categorize various charts and diagrams.

**A mind map is a graphic diagram that is used to organize information visually. It usually follows a hierarchical approach and shows relationships between various pieces of data as a whole.**

We shall leverage a mind map to see how to visualize the charts. The key question we want to address that becomes the central node of the mind-map is the purpose of data visualization. Based on the hierarchical decision tree like mechanism, we reach the terminal node that gives us an option to visualize the data for a particular purpose.

***Figure 3.1:*** *A mind-map of data visualization approaches and chart selection*

There are numerous ways to group or categorize the charts, and all approaches are equally amenable to a good visual representation. A decision tree style table has been constructed to show how we can decide on the key data visualization elements. Suppose we expand the concept or idea further. In that case, we can build a comprehensive mind-map of or a hierarchical table that we can look for various data visualization purposes like *Table 3.3.*

| **Data Visualization -> What is the purpose? - What do you want to visualize?** | | | | |
| --- | --- | --- | --- | --- |
| **No** | **Purpose** | **Criteria-1** | **Criteria -2** | **Chart** |

1 Show distribution of data

Single variable Bar histograms Line histograms

|  |  | Multiple variables | | Scatter plots |
| --- | --- | --- | --- | --- |
| 2 | Spatial data | | | Static maps  Proportional  bubble map Flow maps  Contour maps  Dot density maps Choropleth maps Heat maps |
| 3 | Comparison of data | Ranking | | Ordered bar charts  Ordered column charts  Ordered  proportional  bubble charts |
| Data comparisons | | Slope charts  Lollipop charts Bump charts |
| Overtime | Cyclical data | Circular area chart  Line chart |
| A cyclical data | Bar chart  Line chart |
| Comparison of multiple items | Many items to compare | The variable width bar chart Table with charts Trellis charts  Trellis line charts |
| 4 | Show deviations | Single category | | Diverging bar chart |
| Spline charts |
| More than one category | | The diverging stacked bar chart |
| Surplus/deficit  chart |

5 Relationship of data

Two variables Scatter plots Data tables

Connected scatter

plots

Column + line

combo charts

|  |  | More than two variables | | Bubble charts  Line charts  Data tables  Heat maps |
| --- | --- | --- | --- | --- |
| 6 | Part of a whole data | Simple share/part | | Pie chart  Doughnut charts Arc diagrams  Venn diagrams |
| Addition/subtraction of charts | | Waterfall charts Marimekko  charts  Grid plots  Waffle charts  Treemaps  Venn diagrams |
| Changing data | Some time-based data points | Bar charts / column charts  Lollipop charts Pictograms  Waffle charts |
| Multiple time-based data points | Line charts  Paired bar charts Proportional  bubble charts  Radar charts  Spider charts  Parallel  coordinates |
| 7 | Show time oriented trends | Irregular data |  | Line chart with markers |
|  |  | Regular data | Single variable | Line charts  Column charts  Bar charts |

Multiple variables Column + line

charts

Slope charts

Area charts

Connected scatter

plots

Heat maps

Circle timeline

Vertical timelines

Streamgraph

Seismogram

| 8 | Flows | | Sankey charts  Chord diagrams Waterfall charts Network maps |
| --- | --- | --- | --- |
| 9 | Other types of charts | Financial charts | Candlestick  charts  Ohlc charts  Renko charts  Heiki Ashi charts Point & figure charts  Sharp charts  Seasonality  charts  Performance  charts  Market charts |
| KPI charts | Thermometer  charts  Dial/dashboard chart  Numeric chart  Meter chart  Big numeric chart |
| Word charts | Text summary  Word cloud  Word trees |

***Table 3.3:*** *A hierarchical decision-making tree table for data visualization chart selection* Let us see the different types of data visualization tools available.

**Grouping and categorization of data visualization tools**

About two decades ago, only a few standard data visualization software tools were popular. One of the most popular tools is Microsoft Excel. Statisticians used packages such as SAS and programmed using R. With the growth of the Internet and computing power; many new avenues opened up. Due to increased popularity and the need for data visualization, there are plenty of choices available. There are applications, libraries, APIs, and language-specific options available for visualization.

Some visualization libraries are specific for a particular purpose, such as statistics, machine learning, and financial reporting. Most of the popular libraries have common visualization options and elements for usage. For Python, we have a good number of libraries available. We shall be covering some of the popular tools (Matplotlib, Pandas, Seaborn, and Plotly) in detail and cover the basics of some of the emerging tools. We shall be leveraging some of the charting libraries implicitly in some of the programs we shall discuss.

The following diagram will be very handy to refer to summarize the tools available for data visualization. We will group them into commercial software, visualization applications (with commercial and free options), and visualization libraries. The mind map below covers them in detail:



***Figure 3.2:*** *A representation of the grouping of data visualization tools*

Let us see some of the details of these key visualization tools for consideration. For this book, the scope is primarily on Python as a choice for visualization language. Hence, a separate category on the use of Python specific libraries is included. We can see the details in the next section.

**Software tools and libraries available for data visualization**

Let us see some additional details about the tools we covered in the data visualization tool mind-map above. This can be very handy if you would like to explore learning and programming, in addition to the libraries we cover in this book through examples and exercises. The following table covers some of the most popular data visualization tools and corresponding references:

| **Product / Library Name** | **Link** |
| --- | --- |
| Altair | **https://altair-viz.github.io/** |
| Bokeh | **http://bokeh.pydata.org/en/latest/** |
| Cartopy | **http://scitools.org.uk/cartopy/** |
| D3 JS | **https://d3js.org** |
| Datashader | **https://github.com/bokeh/datashader** |
| Folium | **https://github.com/Python-visualization/folium** |
| GeoPandas | **https://geopandas.org/** |
| Geoplotlib | **https://pypi.Python.org/pypi/geoplotlib** |
| ggplot2 | **https://pypi.org/project/ggplot/** |
| Graphviz | **http://www.graphviz.org** |
| Holoviews | **http://holoviews.org** |
| Leather | **https://pypi.Python.org/pypi/leather** |
| Matplotlib | **http://matplotlib.org** |
| Missing No | **https://pypi.Python.org/pypi/missingno/** |

Networkx **https://networkx.github.io**

| Pandas | **http://pandas.pydata.org/index.html** |
| --- | --- |
| Plotly | **https://plot.ly** |
| Pygal | **http://pygal.org/en/stable/** |
| Seaborn | **https://seaborn.pydata.org/** |
| Vega | **https://vega.github.io/vega/** |
| Vega-Lite | **https://vega.github.io/vega-lite/** |
| Vincent | **https://github.com/wrobstory/vincent** |

***Table 3.4:*** *A table of popular Python centric data visualization tools*

Some of the libraries were created due to Python's open-source nature and flexibility to be extended, and some features were non-existent in existing libraries. Once mastered the language and libraries, you can write your extensions and features that can be leveraged globally.

| **Product / Library Name** | **Link** |
| --- | --- |
| D3.js | **https://d3js.org/** |
| Google Charts | **https://developers.google.com/chart/** |
| Excel | **https://office.live.com/start/Excel.aspx** |
| FusionCharts | **http://www.fusioncharts.com** |
| Google Sheets | **http://www.google.com/sheets/** |
| Illustrator | **http://www.adobe.com/products/illustrator.html** |
| Tableau | **https://public.tableau.com/s/** |

***Table 3.5:*** *A table of popular data visualization tools that are not Python-based*

We shall cover the purpose of the top libraries we shall be covering in this book, such as Matplotlib, Bokeh, Plotly, Pandas, and Folium, and other key libraries we shall be using in the respective chapters planned in the book. If you are keen to leverage some of the tools highlighted for learning and example coding, please refer to the links provided above.

**Conclusion**

In this chapter, we covered a good amount of introduction to data visualization elements and got introduced to the key elements of a data visualization – charts and graphs. We also delved into choosing the charts and graphs to support the six key elements we covered in the preceding chapter: strategy, structure, story, data, design, and user. We built a mind map and a decision tree for visualization purposes. The decision tree can be used to decide what we want to use for our visualization needs. We also covered various tools available for data visualization and focused primarily on the tools available for Python.

Over the next few chapters, we shall see how to use the libraries and tools mentioned in a hands-on and practical manner. The examples will be in Python, but if you want to extend your skills in other languages, there are plenty of choices that are covered with relevant information. The sheer volume of the ever-expansive tool list highlights the importance and popularity of data visualization.

**Questions**

1. Why is choosing the right visualization element important? 2. What are some of the benefits of pie charts?

3. What will happen if there are too many variables to display? What will be your approach to visualization?

4. How will you decide what type of chart to use for time-centric data? 5. What are the top commercial tools for data visualization? 6. What are some top applications for data visualization?

7. What are some of the top Python-based Visualization libraries? 8. What would be your choice for map-based visualization in Python? 9. What would be your choice to handle missing data for data visualization in Python?

10. What is the benefit of having a dynamic dashboard as a data visualization?

**CHAPTER 4**

**Using Matplotlib with Python**

**“Most of us need to listen to the music to understand how beautiful it is. But often that’s how we present statistics: we just show the notes, we don’t play the music.”** *- Hans Rosling*

S

o far, we’ve covered aspects of data visualization in a theoretical manner. In this chapter, we shall have some real practical examples of how to build them. We shall take one of the most widely used libraries to do data visualization - Matplotlib. We shall be covering the key features of Matplotlib and cover some of its core functions and features for using it with Python. As Matplotlib has a good number of features that can run into a voluminous book, we shall cover the key essentials to understand how to use Matplotlib for data visualization using Python.

**Structure**

In this chapter, we will cover the following topics:

Introduction to Matplotlib

Definition of figure, plots, subplots, axes, ticks, and legends

Matplotlib plotting functions

Matplotlib modules

Matplotlib toolkits

Examples of various types of charts

Exercises and Matplotlib resources

**Objective**

This chapter aims to give a basic introduction to Matplotlib for experimenting with various data visualization elements such as charts and graphs. We will also touch upon details of how to use Matplotlib elements for effective visualization. We shall start with simple examples of Matplotlib and explore the libraries and toolkits available before trying out some of the important charts and graphs covered in earlier chapters. We shall conclude the chapter with some examples to try out and resources to consume.

**Introduction to Matplotlib**

****

***Figure 4.1:*** *The Matplotlib logo*

**Matplotlib** is the most widely used Python library for various visualizations - static, dynamic, or animated. It is one of the most comprehensive libraries that exist in the Python tool landscape. A good number of libraries have extended Matplotlib for advanced features. These include BaseMap, Cartopy, Mplot3d, and Seaborn, to name a few. Visualization programs using Matplotlib can be done using Python scripts using an IDE, Python Shells, Python Notebooks (such as Jupyter), Web Application servers, and GUI tools.

John D. Hunter originally developed Matplotlib as an experiment to visualize some of the medical images of epilepsy patients. Matplotlib, along with the power of NumPy, SciPy, and iPython, became very popular over the last decade to become the most widely plotted library. The influence of MATLAB is very highly visible in the name of the library and how the interfaces are developed and named. MATLAB heavily influences Matplotlib’s interface.

**Note:** Before we try the examples, there is an assumption that the readers have basic knowledge of the following areas in Python

Python syntax

Variable types

Basic operators

Loops (while, for, nested loops, etc.)

Conditional statements (if, if-else, etc.)

Python data types - numbers, strings, lists, tuples, dictionary

File I/O

Exception handling

Python functions (defining a function, calling a function, etc.)

Arguments handling (required arguments, keyword arguments, default arguments, variable length arguments, etc.)

Python classes and objects

Ability to set up Python environments

Additionally, it is assumed that the reader has a basic knowledge of using some of the key libraries such as:

NumPy – Array handling functions

Pandas – Data handling functions

It is also expected that the reader has knowhow of

Setting up a Jupyter Notebook environment for execution of the exercise

or

Setting up an integrated desktop environment (IDE) such as Visual Code, PyCharm, Spyder, Atom, or any popular IDE of choice by the reader to set up Python for running the exercises

To install matplotlib and the dependencies (libraries it refers to), we can use the following command in the command prompt or a shell. We need to make sure Python or pip utility is in the environment variable or the path.

**Note:** Let us see how to install\* matplotlib using Python installer in various operating systems:

**In Windows:**

***python -m pip install -U pip***

***python -m pip install -U matplotlib***

**Or simply**

***pip install -U matplotlib***

**In Ubuntu Linux:**

***sudo apt-get install python3-matplotlib***

**In macOS:**

***sudo pip3 install matplotlib***

**(\* There is a difference in the installer names of Python installer across operating systems. Suggest you refer to the right version. Ensure the dependent packages are installed before or during the installation of MatPlotlib.)**

The *matplotlib* depends on the following packages to function well. The above commands can ensure dependent packages (*Python, FreeType, libpng, NumPy, setuptools, cycler, dateutil, kiwisolver, and pyparsing)* are downloaded and installed.

Once the installation is completed, we are ready to do programming in Python with *matplotlib.*

**Note:** One of the key requirements for visualization plotting and programming is data. For the examples used throughout the book, we shall be using the following types of data.

Random data generated using random functions in Python (primarily for numeric data)

Sample hard-coded data (for simplicity)

Open datasets available in the public domain (such as IRIS dataset, titanic passenger data)

Realtime datasets available (such as data from stock markets)

For each chapter, we shall introduce the type of data we use for the examples in the chapter.

Let us start with a simple example. In this example, we will create simple **Sine()** and **Cosine()** waves.

1. import numpy as np

2. import matplotlib.pyplot as plt

3. plt.style.use('default')

4.

5. x = np.arange(0, 10.5, 0.2)

6. y = np.sin(x)

7. z = np.cos(x)

8. plt.plot(x, y)

9. plt.plot(x, z)

10. plt.show()

This program produces the following output:



***Figure 4.2:*** *A Python code output for displaying simple Sine and Cosine waves*

As you can see, it is very easy to build a visualization with very few lines of code in *Python*. We shall see the details about various functions used and parameters as we progress along in this chapter. At a very simple level, it uses the *pyplot* library from *matplotlib* to build a simple mathematical line chart based on trigonometric values for *sine* and *cosine* waves.

We import the libraries, set the data configuration, define the parameters, initiate the plot/graph, call the functions for various settings, and draw the plot. This is a simple structure of a *pyplot* visualization.

The next section shows how we can define the figure, plots, subplots, axes, and other parameters needed to define a matplotlib plot.

**The definition of figure, plots, subplots, axes, ticks, and legends**

Let us take the previous example of how the plot is built. This is done systematically, and every visualization element follows a process for the generation of the plot. This includes the figure, axes, the plotting type, title for the figure, x and y-axis label, ticks, and further segregations of the axis elements at a very high level. There are legends and other aspects to consider that we shall cover in the upcoming sections.

***Figure 4.3:*** *Matplotlib Plot – Architectural representation of a Python visualization interaction*

In this diagram, the **Figure** is an element containing one or more than one axes or plots for visualization. An **Axes** can contain mostly one or more axes for rendering the plots. In some cases, it can have more than two axes to plot 3D elements and objects. Typically, each **Axes** has a combination of X-axis and Y-axis with a title, **ticks,** and label. Each axis handles the graph/plot limits that are rendered. Each axis has ticks to define the axes parameters. Finally, each plot is rendered through **Artists** tied to axes and render any text or visual objects rendered in a plot. **Artist,** like the name says, renders visual elements to a canvas. It takes care of the representation of figures, texts, axes, and lines on the canvas.

Let us take the anatomy of a plot in ***pyplot*** in *matplotlib.* To do this, let us get all the commands that exist in matplotlib. We can do this using the following line of code. We need to

understand these key parameters as they form most of the visualizations we will be doing over the next few chapters. Even the extensions of matplotlib refer to some of these core functions implicitly. If we execute **plt.get\_plot\_commands()** in the python console***,*** we get all the commands used for various plotting aspects of the matplotlib library in the console.

***Figure 4.4:*** *Matplotlib plotting functions displayed through the get\_plot\_commands() command* Let us see various aspects of these libraries in a tabularized fashion.

**Matplotlib plotting functions**

Let us see all plotting functions available in *Matplotlib.*

**Note:** The commands are like a library reference. Only some of the plots and functions are used in the examples showcased in this book. The number of functions is too many to cover. Readers are requested to glance through or use the section as a reference.

**Plotting functions**

We shall cover them in various categories serving distinct purposes such as plotting, sub plotting, axes, legends, and general features. The majority of the types of charts were introduced at a high level in *Chapter 3*.

**Plot - plot(\*args[, scalex, scaley, data])**

Creates a general plot of y values versus x as lines. Additionally, parameters can be passed for line markers.

Bar charts

**bar - bar(x, height[, width, bottom, align, data])**

Used for a bar chart plotting. Additional parameters specify the height and other parameters.

**barh - barh(y, width[, height, left, align])**

Used for a horizontal bar chart plotting. Additional parameters specify the width and other parameters.

Pie charts

**pie - pie(x[, explode, labels, colors, autopct, …])**

Other key charts/plots

**boxplot** ⇒ Used for box plots

**hexbin** ⇒ Used for hex bin plots

**loglog** ⇒ Used for logarithmic charts

**polar** ⇒ Used for polar charts

**step** ⇒ Used for a step plot - A variant of line plot where two connecting points are shown in a step

**violinplot** ⇒ Used for a violin plot to show data density

**hist** ⇒ Used for a histogram chart

**hist2d** ⇒ Used for a 2D histogram

**matshow** ⇒ Used for an array of matrix display in a chart

**specgram** ⇒ Used for a spectrogram chart

**scatter** ⇒ Used for a scatter chart

**quiver** ⇒ Used for a quiver chart

**contour** ⇒ Used for a contour chart

**stackplot** ⇒ Used for a stack plot - for plotting a stacked area chart

**stem** ⇒ Used for a stem and leaf plot representation

**streamplot** ⇒ Used for drawing a streamplot

**imshow** ⇒ Used to show data as an image

**Subplot functions**

Subplots are used extensively in Matplotlib, and the two key functions to be aware of are as follows.

**subplot(\*args, \*\*kwargs)**

Adds a subplot to the current figure in the display

**subplots([nrows, ncols, sharex, sharey, …])**

Create a figure with the specified number of subplots specified as per rows and columns

**Coloring functions**

The following are some of the coloring functions to use for **Matplotlib**: **set\_cmap(cmap)** ⇒ Sets a colormap style

We can set various colormaps using the available functions -> **autumn(), bone(), cool(), copper(), flag(), gray(), hot(), hsv(), inferno(), jet(), magma(), nipy\_spectral(), pink(), plasma(), prism(), spring(), summer(), viridis(), winter()**

**colorbar** ⇒ Adds a colorbar to the image

**Config functions**

**matplotlibrc** is a configuration file used to set properties by **matplotlib. rc settings** and **rc parameters** are used to set default values of a figure size and DPI (dots per inch), line width, color, and style, axes, axis and grid properties, text, and font matplotlib. This is a useful feature to leverage, and the following functions are key for this purpose. The configurations for plotting are set using

**rc** ⇒ Set current RC parameters

**rc\_context** ⇒ get context for changing current RC parameter

**rcdefaults** ⇒ restore rcparams to matplotlib defaults

Understanding the core functions of ***matplotlib*** is important as it would form the basis for all our exercises in this book. Like mentioned earlier, even the advanced libraries build on these foundations. Now that we have seen all the functions comprehensively let us pick some practical examples of using matplotlib to draw some visualizations.

**Matplotlib modules**

There are other extensions available in the form of modules for Matplotlib that can be leveraged for various features that are not a part of the core *matplotlib* library. Some of the major modules and their features are as follows:

| **Function** | **Purpose** |
| --- | --- |
| matplotlib.artist | Abstract base class for all the objects that are rendered in matplotlib through in a FigureCanvas. Every visible element is a subclass of this. |

matplotlib.axes The Axes contain most of the figure elements rendered, such as Axis, Tick, Line2D, and Text. It is used to coordinate system management in

matplotlib.

| matplotlib.axis | This contains the class to set X and Y-axis ticks. |
| --- | --- |
| matplotlib.cm | Color Maps and Color Map Utilities. |
| matplotlib.colorbar | Has functions to map scalar values into colors for visualization. |
| matplotlib.colors | Has classes and functions to set various color-related elements and themes in Matplotlib. |
| matplotlib.contour | Functions to support contour plots. |
| matplotlib.figure | Implements the top-level Artist Class that manages all the plotting elements in Matplotlib. |
| matplotlib.gridspec | To build a grid-like figure using multiple axes to plot a complex visualization. |
| matplotlib.image | Supports basic image operations. |
| matplotlib.legend | Useful for the creation of legends for the plots. |
| matplotlib.markers | A function to define and use various marker types. |
| matplotlib.patches | Contains utility functions such as different polygons etc. |
| matplotlib.pyplot | The core library is used extensively for plotting in Matplotlib. |
| matplotlib.quiver | Used for plotting quiver plots. |
| matplotlib.rcsetup | Has features to customize the default rc settings of Matplotlib. |
| matplotlib.style | Used for setting various styling themes for the plots. |
| matplotlib.text | Functions for text usage in a plot or figure. |
| matplotlib.ticker | Used for ticks formatting and locating. |
| matplotlib.tight\_layout | Used for layout formatting and beautifying the subplots in a figure. |
| matplotlib.widgets | To build and use GUI neutral widgets. |

***Table 4.1:*** *Matplotlib key modules*

**Matplotlib toolkits**

There are other toolkits available in Matplotlib that can be leveraged for 3D plotting and similar features. They are application-specific extensions that include a collection of functions for plotting. One of the top toolkits available is *mplot3d.* The extension *mplot3d* toolkit is used for leveraging the 3D plotting to extend matplotlib capabilities. This is done by adding an axes object to create a 2D perspective of a 3D perspective using the functions.

**Examples of various types of charts**

**Line plot**

Let us see a simple example (Exercise 4-2 below) of plotting a line plot by generating random numbers using the *NumPy* random function. We assign the random numbers generated using the **randn()** function to a 1-dimensional array using the **cumsum()** function in *Numpy.* The **cumsum()** function is used to compute the cumulative sum of array elements over an axis. This is shown as the line.

This example code has two parts. The first part of the code produces a default line chart displayed based on a 1-dimensional array of random values. We generate 15 random numbers using the *NumPy* function and assign them to a 1-dimensional array. The second part covers some of the plot features such as marker type, markersize, markerfacecolor, linewidth, linestyle, and label, along with the color of the lines.

1. *# import the libraries*

2. import matplotlib.pyplot as plt

3. import numpy as np

4. *# create data - 1 dimensional array of 15 elements using Random Numbers* 5. *# Use Numpy cumsum() function for array and random.randn() for randomization*

6. oneDimArrayValues=np.cumsum(np.random.randn(15,1))

7. *# basic plot function*

8. *# Following Line gives simple chart*

9. plt.plot(oneDimArrayValues)

10. plt.title('Simple Linechart')

11. *# Matplotlib Plot – Exercise 4-2 Output-1*

12. plt.show()

13.

14. *#Following gives a bit more detailed features*

15. plt.plot(oneDimArrayValues, marker='\*', markerfacecolor='red',markersize=10,

16. color='green', linewidth=3, linestyle='dashdot', label="Simple Line Chart")

17. plt.title('Simple Linechart with Markers')

18. *#Matplotlib Plot – Exercise 4-2 Output-2*

19. plt.show()

The code produces the following output shown below in *Figure 4.5* and *Figure 4.6*. The first one is a simple chart, and the second one is a line chart with markers.

**First simple line chart without markers The second one is the line chart with markers**

***Figure 4.5: Matplotlib Plot –*** *Exercise 4-2 Output-1* ***Figure 4.6: Matplotlib Plot –*** *Exercise 4-2 Output-2*

**Exercise 4-3**

How do we add multiple lines in the same chart? Let us take it a bit further to plot multiple lines in the same chart:

1. *# Import the libraries PyPlot, numpy for random numbers, pandas for dataframe*

2. import matplotlib.pyplot as plt

3. import numpy as np

4. import pandas as pd

5. *# Define the Data using a Pandas Dataframe using NumPy random numbers* 6. df=pd.DataFrame({'x': range(1,21),

7.                  'yy': np.random.randn(20),

8.                  'zz': 2 \* np.random.randn(20) + range(1,21), 9.                  'ww': 4 \* np.random.randn(20) + range(1,21), 10.                  'aa': 8 \* np.random.randn(20) + range(1,21), 11.                 })

12. *# Create a multiple line plot by calling 3 plot functions with various markrs*

13. plt.plot('x', 'yy', data=df, marker='p', markerfacecolor='blue', markersize=8,

14.          label="Pentagon", color='skyblue', linewidth=4)

15. plt.plot('x', 'zz', data=df, marker='h', color='green',

16.          linewidth=2, label="Hexagon")

17. plt.plot('x', 'ww', data=df, marker='8', color='red',

18.          linewidth=2, linestyle='dashdot', label="Octagon") 19. plt.plot('x', 'aa', data=df, marker='d', color='violet',

20.          linewidth=2, linestyle='dotted', label="Diamond")

21. plt.title('Linechart with Multiple Line Charts')

22. *#Show the legend that displays colors and markers*

23. plt.legend()

24. plt.show()

This produces the following output. This program takes the previous example by adding multiple elements of comparing the x-axis against various values for four different data arrays to compare with. This chart has a clear title definition and a legend for various lines used in the chart.



***Figure 4.7:*** *Exercise 4-3 Python output – multiple line charts*

In the notebook covering examples and exercises for *Chapter 4*, we have included another code snippet (Exercise 4-4). This code showcases various style types. Let us see some examples using the Bar Chart.

**Bar plots**

**Bar plots** are one of the most popular charts used in any type of visualization as it shows the relative data points in an emphasized manner like a line chart. As explained in *Chapter 3*, a bar chart can be used to compare the progress of a particular element within an entity or across a set of entities to showcase a relative comparison of values.

Let us take a simple bar plot that displays the yearly revenues of a technology firm from 2008 onwards. We have used a simple parameter **color** to color the bar purple.

1. *#import the libraries*

2. import matplotlib.pyplot as plt

3. *#set style = default*

4. plt.style.use('default')

5. *#Let's put the years in an array for y axis*

6. year = [2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020]

7. *#Let's define the revenue in an array for x axis*

8. *#revenus of Technology firm in Billions - available in public sites* 9. revenue = [37.5,42.9,65.2,108.2,156.5,170.9,182.8,233.7,215.6,229.2,265.6,260.2,274 .5]

10. *#Let's build a bar chart with a purple color filling*

11. plt.bar(year, revenue, color='purple')

12. *#set title*

13. plt.title('Technology revenues in billions of US $ dollar') 14. *#set x and y axis labels*

15. plt.xlabel('Year')

16. plt.ylabel('Revenue')

17. *#show the plot*

18. plt.show()

The output of the program will be as follows:



***Figure 4.8:*** *Exercise 4-5 Python Bar chart example - Output*

Let us explore how to take the available features further with a bit more advanced features in a bar chart. This program covers subplots, setting figure size, xticks, xlabel/ylabel, title, and legend we used. It also uses various parameters of the bar function available in Matplotlib.