Comprehensive Review of Data Visualization Techniques using Python

Ayush Kumar Rathore
Amity University Uttar Pradesh,
Lucknow Campus

Dr.RanjanaRajnish
Amity University Uttar Pradesh,
Lucknow Campus

Abstract: Big Data has changed the way that data is handled, processed and leveraged in any sector. Healthcare is one of the most important fields where making a transition can be made. Visualization techniques can help to visualize the relationship between data in a visual or graphical manner. Data visualization primarily deals with relationship between data by various statistical parameters and then showing them in the form of graph. When the data is presented in the form of visuals, it becomes easy to understand and makes insights within the data and patterns clearly visible. Big data refers to the vast amounts of information produced by digitizing everything that different technologies store and process. Data visualization finds many applications, healthcare being the upcoming industry embracing technology to reap its benefits. Recent example of use of data visualization in healthcare can be seen in the COVID-19 scenario where lot of visual are appearing to demonstrate various aspects of COVID-19. In the healthcare sector, the implementation of big data analytics has many positive lifesaving outcomes. For individual, a single dashboard for each patient may be created to show his/her entire history, that may help reducing time of treatment. Similarly, if we see data of a bigger group, it may help us to identify some patterns and make predictions regarding some infectious disease or epidemic. Data visualization is all about presenting the data to the right people at the right time. It helps gain deep insights to the data.

In this paper, the objective is to have a comprehensive discussion on Data Visualization and different techniques that are used for data visualisation.

Keywords: Data Visualization, Line Plots, Area Plots, Histograms, Bar Charts, Pie Charts, Box Plots and Scatter Plots, COVID-19

I. INTRODUCTION

In recent times there has been steep rise in the data being produced by almost all areas like social media, healthcare etc. Not only the data is voluminous, it possesses all other characteristics of big data. Data Visualization helps to identify hidden pattern in this huge dataset and shows them in a graphical form with the help of different plots. It has thus recently become a topic of interest for researchers, industry and academicians.

Data visualization has many fold benefits like better understanding of data, information sharing, helps making better decisions, improved ROI and time saving (due to quick glance of data). Visualization can also be presented in the form a dashboard where quick links of important analysis are available, and important information about data can be visualized at a glance.

This paper consists of the six sections: Data Visualization, Tools used for data visualization, Python libraries used for data visualization, Jupiter Notebook- Beginners Toolbox for Visualization and Basic plotting using Matplotlib.

II. DATA VISUALIZATION

In this section we will discuss visualization of data and take an example of how a visual is converted into one that is more accurate, attractive and impactful. So, let's proceed.Now, why should we learn how to view data? You may wonder? Okay, data visualization is a way of displaying complex data in a graphically and easily understandable manner. This can be particularly useful as you seek to discover and familiarize your personal data. Since a picture is worth thousands of words, plots and graphs can be extremely effective, specifically when transmitting results to the public or when sharing the data with other peer data experts. They can also be very helpful in promoting your guidance to consumers, managers and other politicians in your field.

In technical terms Data Visualization can be defined as "*The process of translating data and metrics into charts, graphs and other visual reports. These visualizations let viewers discover patterns and relationships in the data that they otherwise might not see — helping to turn the information into a cohesive story. Data visualization enables organizations and individuals to gain a clearer understanding of their performance and goals.*". [1]

Existing data visualization techniques can be classified as 1D, 2D, 3D, multi-dimensional, temporal, tree and network graphs.

Some basic plots/charts used in Data Visualization are:

- Line Plots
- Area Plots
- Histograms
- Bar Charts
- Pie Charts
- Box Plots
- Scatter Plots

III. TOOLS USED FOR VISULIZATION

Lot of tools are available that help in making visualization task easier. Just like other fields, visualization tools are also evolving and providing different ways to present the data. We can have graphs, video, infographics and even VR-AR components to make the presentations more informative. Some of the commonly used tools, according to Forbes [8], for data visualization are as listed below:

- Tableau
- Qlikview
- FusionCharts
- Highcharts
- Datawrapper
- Plotly
- Sisense

Most of these are paid software, but they offer free trial versions. In this paper, we have not used any of the tools, but have used Python libraries to see how different types of plots can be used to visualize data.

IV. PYTHON LIBRARIES COMMONLY USED FOR DATA VISULIZATION

Python provides many great graphic libraries filled with many features. Whether you are interested in creating interactive or live stories, Python has some excellent libraries:

Here are a few popular plotting libraries to get a little overview:

- Matplotlib
- Pandas
- Seaborn
- ggplot
- Plotly

Darkhorse Analytics is a company that started out in 2008 and did ground-breaking work on data visualization in a research laboratory at the University of Alberta. In several fields including analysis of data and geospatial analytics Darkhorse Analytics specializes across quantitative consulting. Your visual approach revolves around three key points: less is more successful, more appealing and more impactful. In other words, any function or design that you introduce to your plot will endorse the message that the plot should be conveying and not diverting from it.

Let's take a look at an example. The proceeding pagecontains a picture of the pie diagram of what people like when it comes to various kinds of pig meat. The charts are almost half the preference for bacon in contrast with the other types of pork meat. But I am sure almost everyone agrees that this pie chart is a lot going on, and we're not even aware that it has features like blue background or 3d orientation. These other unwanted features are simply distracting from the main message and can confuse the viewer.

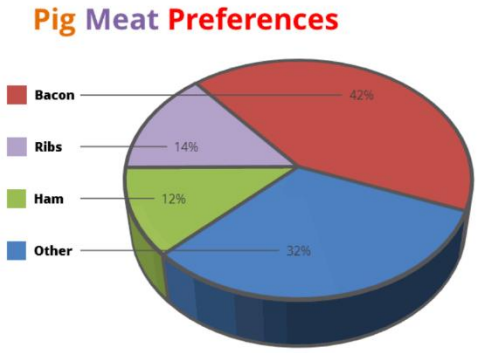

Figure 1: Pig Meat Preferences

So, let's apply Darkhorse Analytics approach to transform this into a visual that's more effective, attractive, and has more impact. As I mentioned earlier, the message here is that people are most likely to choose bacon over other types of pig meat, so let's get rid of everything that can be distracting from this core message. The first thing is let's get rid of the blue background and the grey background. Let's also get rid of borders as they do not convey any extra information. Also let's drop the redundant legend since the pie chart is already color coded. 3D isn't adding any extra information so let's say bye to it. Text bolding is also unnecessary, and let's get rid of the different colours and the wedges. But let's thicken the lines to make them more meaningful.

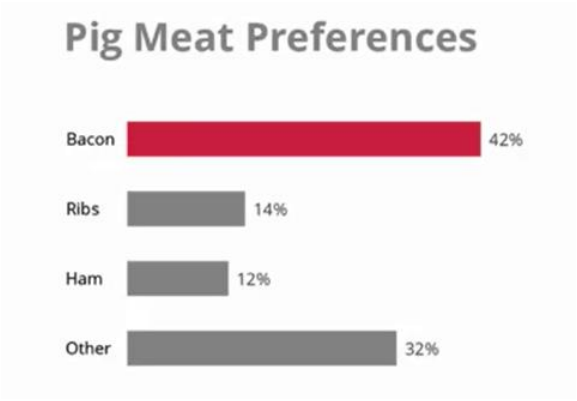

Figure 2: Bar Chart Pig Meat Preferences

It looks a bit familiar now. Yes! After all, this is a bar graph that has horizontal bars. Finally, let's highlight bacon to differentiate itself from the other kinds of pork meat. Let's now bring the pie chart and the bar graph together and compare what is better and more comprehensible. I hope we all agree unanimously that the bar chart is the best of both. It's simpler, smoother, less distractive and readable. Indeed, pie charts have recently been put on fire by experts in data visualization, who argue that they are only of relevance in the rarest of situations. On the other hand, bar graphs and charts are claimed to be far better ways of getting a point across quickly. But don't worry about this for now, we will come back to this point when we learn how to create pie charts and bar graphs with Matplotlib.

V. JUPITER NOTEBOOK- BEGINNERS TOOLBOX FOR VISUALIZATION

Here we learn how to create plots with Matplotlib and use the Jupyter notebook as our enviroment setting.Matplotlib is a well-established data visualization library, which is well integrated in various environments like the ipython shell, the web-application server, the graphical user interface toolkit and the Jupyter notebook, for example. It is now available in a number of different applications. It's an open source web application that allows you to create and exchange documents with live code views and some informative texts. Jupyternotebook is an IDE for Python development and has specialized support for Matplotlib, so you can simply import Matplotlib and get ready to go if you start a Jupyter Notebook. It is a very good development environment for presentation of your work as well as for those who are beginning with data science. It has rich libraries to support data science and should be part of data scientist toolbox.

VI. BASIC PLOTTING WITH MATPLOLIB

In this section, we learn how to use the Jupiter scripting interface to produce nearly all visualization resources. Once you find, you can really build virtually all traditional visualization tools like histograms, bar charts, box plots and many more, using only one element: the plot function when using the plot feature.

Dataset

In this section, we will learn more about the dataset that we will be using throughout the section. The data set I have chosen is on airline safety which was compiled by **FiveThirtyEight** using their large database of articles and news. It was distributed under the Creative Commons Attribution 4.0 International License as the GitHub kit and their data sets are licensed under the MIT License. To use the dataset, you can go to GitHub and download their dataset library or can download directly using link in the reference [3].

We need to import the data in a pandasdataframe to start creating different types of plots of data. We will have to import the pandas' library for extracting data from Excel table files. Instead we call the read_csvpandas function to read the data in pandas. As an argument, the function read_csv expects filename of the file that contains dataset. And let this dataframe be named df.

```
In [61]: import pandas as pd

         df = pd.read_csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/airline-safety/airline-safety.csv",index_col=0)
```

Figure 3: Importing Pandas and CSV

If you want to confirm that you have imported your data correctly, in pandas, you can always use the head() function to display the first five rows of the dataframe.

Out[63]:

airline	avail_seat_km_per_week	incidents_85_99	fatal_accidents_85_99	fatalities_85_99	incidents_00_14	fatal_accidents_00_14	fatalities_00_14
Aer Lingus	320906734	2	0	0	0	0	0
Aeroflot*	1197672318	76	14	128	6	1	88
Aerolineas Argentinas	385803648	6	0	0	1	0	0
Aeromexico*	596871813	3	1	64	5	0	0
Air Canada	1865253802	2	0	0	2	0	0

Figure 4: Top five rows of the CSV

Line Plots

Line plots are a trace in the form of a set of data points connected by straight line, as its name indicates. It is a most simple type of diagram and is popular not only in data science but in many fields. When to use line plots,

the more important question. The best way of using a line plot is to have an on-going dataset and to display the data over a time period. For an example, say we want to know the Available seat kilometres flown every week of top 5 airlines in the dataframe. Based on this line plot, we can then research for the safety measures which can be taken on these airlines. Okay, now, how can we generate this line plot? Before we go over the code to do that, let's do a quick recap of our dataset. Our dataset contains the following columns:

Airline	Airline (asterisk indicates that regional subsidiaries are included)
avail_seat_km_per_week	Available seat kilometres flown every week
fatal_accidents_85_99	Total number of fatal accidents, 1985–1999
fatalities_85_99	Total number of fatalities, 1985–1999
incidents_00_14	Total number of incidents, 2000–2014
fatal_accidents_00_14	Total number of fatal accidents, 2000–2014
fatalities_00_14	Total number of fatalities, 2000–2014
Incidents_85_89	Total number of incidents, 1985–1999

Now let's process the dataframe. We already defined the dataframedf in Last section so using df as the dataframe we will plot the line graph.

```
In [3]: dtf = df.head()
        dtf.loc[:,'avail_seat_km_per_week'].plot(kind="line")
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x4f25240>
```

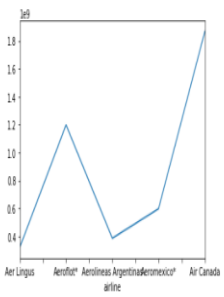

Figure 5: Graph representing Available seat kilometres flown every week

We can see in the graph Air Canada has the highest value so more preference is to be taken for this flight. So, this is how we plot a line graph.

Area Plots

An area plot is also called an area graph is a type of graph representing total accumulations using numbers or proportions over time. It is based on the line plot and is typically used for comparing two or more values. So how can we generate an area plot? It's very simple we will use the same dataframedf to plot area plot. Here we will use an example, say we want to compare incidents of top 5 airlines in file from 1985-1999 and 2000-2014 we will use the two fields incidents_85_99 and incidents_00_14 like this

```
In [7]: dtf = df.head()
        dtf.loc[:,['incidents_00_14','incidents_85_99']].plot(kind="area")
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0xb97de80>
```

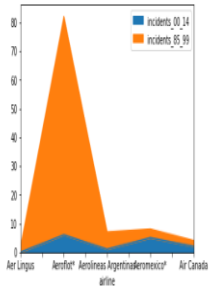

Figure 6: Comparing incidents of top 5 airlines in file from 1985-1999 and 2000-2014

Now from this graph we can infer that the incidents were decreased drastically over time. And this is how we plot Area graphs.

Histograms

A histogram is a way to represent a numerical data set's frequency distribution. The way this works implies that the number of data is split into bins, each datapoint in the dataset is allocated to a bin and then the number of data points assigned to each bin. The vertical axis is the frequency or the number of datapoints in the individual bin.We can understand using an example, say we are interested in knowing the fatalities between 1985-1999 so to do that we will use the same dataframe to plot a histogram.

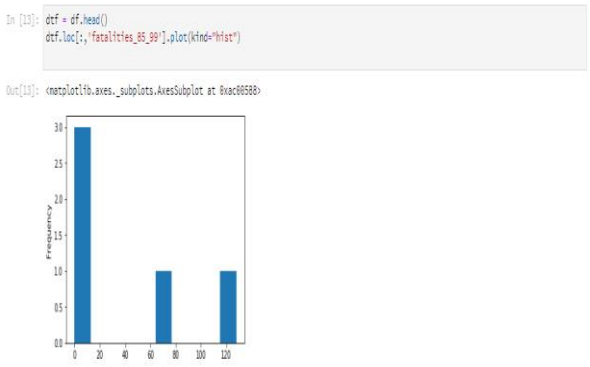

Figure 7: Fatalities between 1985-1999

Note how the tick marks on the horizontal axis are not matched with the tick marks. It makes it difficult to decipher the histogram. Let us try to fix this so that our histogram will be more efficient. The Numpy histogram function is one way to solve this problem. We continue with Matplotlib and its scripting interface as usual, but this time we also import the library of Numpy.

Figure 8: Using Numpy to plot the same graph

And this is how Histogram is plotted in Matplotlib.

Bar Charts

A bar chart sometimes referred to as a bar graph is a kind of plot in which every bar's length is proportionate to its size. The values of a variable are commonly used to compare at certain times.

For example, say we are interested in knowing fatalities by flight Aeroflot* in 1985-1999 and 2000-2014 we will use the same data frame to plot bar chart like this

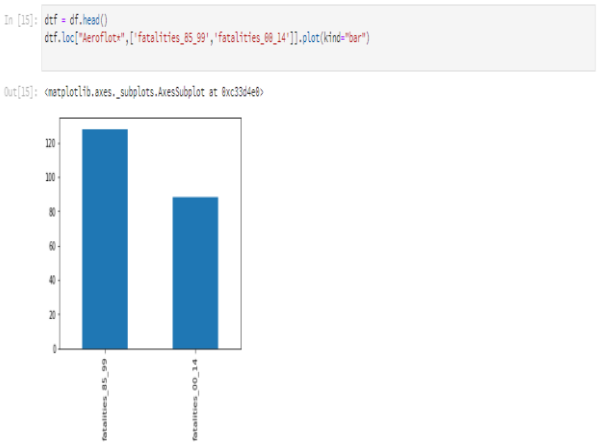

Figure 9: Fatalities by flight Aeroflot in 1985-1999 and 2000-2014

And this is how we plot a bar graph.

Pie Charts

A pie chart is a circular mathematical graph divided into slices to display the numerical proportion. We can understand using an example, say we are interested in knowing incidents between 1985 – 1999 by the top 5 flights in the file we can do like this.

Figure 10: Incidents between 1985 – 1999 by the top 5 flights

Note we are using the same dataframedf. And by the graph we can see that Aeroflot* flight has the highest incident rate between years 1985 – 1999. And this is how we plot pie charts.

Box Plots

A box plot is a mathematical way of dispersing data in five main dimensions. The first calculation is zero, the smallest of the sorted results. The second dimension is the first quartile (25% of the way through the sorted data). A

quarter of the data points, in other words, are less than this amount. Median, which is the mean of the sorted results, is the second dimension. The third quartile is the fourth dimension, which is 75 percent of the time through the sorted information. Three quarters of the data points are less than this amount, in other words. And the ultimate element is the highest number in the sorted results.

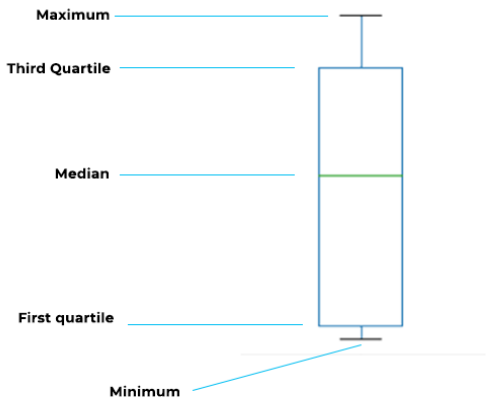

Figure 11: Components of Box Plot

Let's now see how Matplotlib is able to create a box plot. We can understand using an example, say we are interested in knowing Available seat kilometres flown every week of top 5 flights in the descending order we can do like this

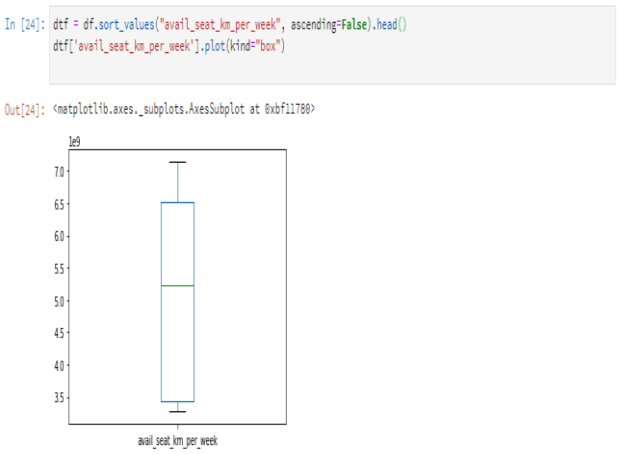

Figure 12: Available seat kilometres flown every week of top 5 flights in the descending order

And this is how we make a box plot. Please Note we are using the same dataframedf to plot the graph

Scatter Plots

A scatter plot is a plot type which shows values of two different variables. It is usually a dependent variable to be measured against an independent variable to determine whether the two variables have any correlation. Of instance, the salary and experience was measured and, looking at the numbers, it can be inferred that a person with more years of experience is likely to earn a higher salary than a person with less years of experience.

So, for example we say we are interested in correlation between salary and year of experience and to do that we will make a new dataframe and use this dataframe to plot a scatter graph like this

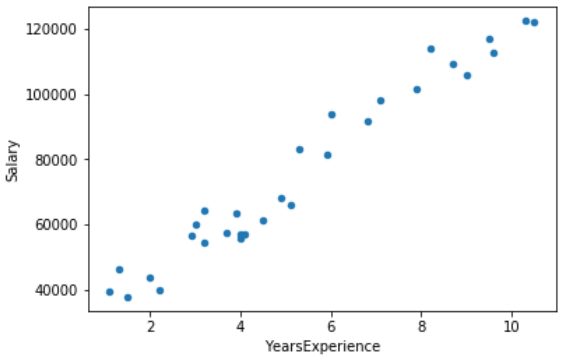

Figure 13: Correlation between salary and year of experience

We will use a new data set which can be found on kaggle (link in the references section). Now we will do like this [4]:

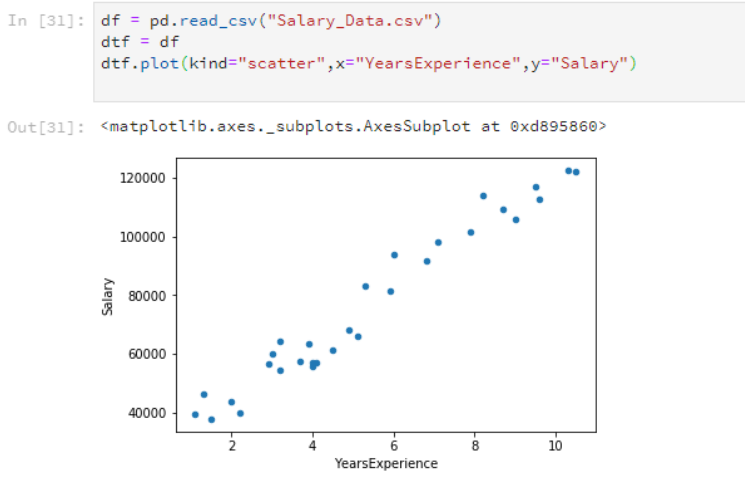

Figure 14: Correlation between salary and year of experience

Note in scatter plot it is necessary to give x and y axis in plot function otherwise it will generate error. And this is how we make a scatter plot.

VII. DISCUSSION

In this paper we discussed the need and importance of data visualization. We discussed the commonly used tools used for data visualization, and about the different types of visualization techniques like 1D, 2D, 3D, multi-dimensional, temporal, tree and network graphs. In this paper, we have explored the basic charts using Python libraries using pandas. We have used Jupiter notebook and dataset from github library to demonstrate various plots.

VIII. CONCLUSION

As seen from the above paragraphs, data visualisation plays an important role in providing visual images that make it easier to comprehend the information stored in the dataset. In this paper we have discussed the concept of data visualisation, use of Jupiter notebook for data visualisation. We also discussed different types of plots and the scenarios where each type of plot can be of help.Data visualization may not be exact solution for analysing. By using various visualization techniques, as per the data, companies, researchers or other using these tools may find insights into their data and find patterns within them. This may help in quick decisions leading to many-fold benefits to the user.

References

[1] https://www.ibm.com/topics/data-visualization
[2] https://www.darkhorseanalytics.com/
[3] https://github.com/fivethirtyeight/data
[4] https://www.kaggle.com/rohankayan/years-of-experience-and-salary-dataset/data
[5] https://matplotlib.org/3.1.1/tutorials/introductory/customizing.html
[6] https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data
[7] https://www.datapine.com/blog/big-data-examples-in-healthcare/
[8] https://www.forbes.com/sites/barnardmarr/2017/07/20/the-7-Best-data-visualization-tools-in-2017/#7d186dc76c30