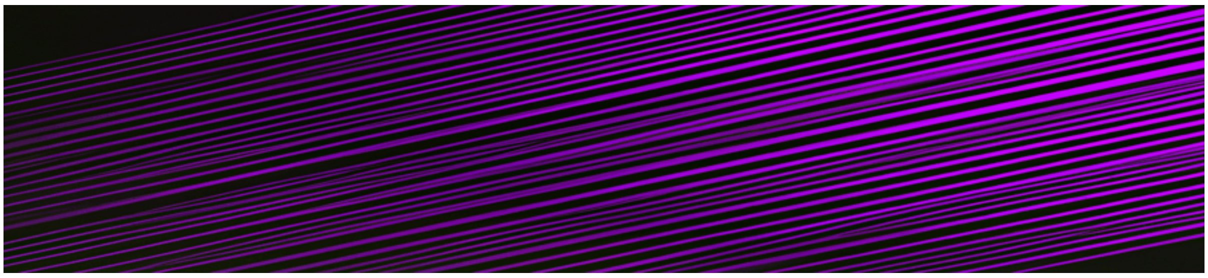
Pierian Training
(https://pieriantraining.com)

PYTHON BASICS (HTTPS://PIERIANTRAINING.COM/PYTHON-BASICS/), TUTORIALS (HTTPS://PIERIANTRAINING.COM/TUTORIALS/)

Seaborn Pie Chart: A Tutorial for Data Visualization

Posted on: 18 May 2023 Updated on: 18 May 2023 Written by: Pierian Training(https://pieriantraining.com/author/ptblogstaff/)



Introduction

Data visualization is an essential part of data analysis, and pie charts are a popular way to represent proportions of data. Seaborn, a Python data visualization library, provides an easy and effective way to create visually appealing pie charts.

Pie charts are useful when you want to show the proportion of each category in a dataset. They are circular in shape, with each slice representing a category and its size representing the proportion of that category in the dataset. Pie charts are commonly used in business reports, market research, and scientific publications.

Seaborn is built on top of Matplotlib (https://matplotlib.org/), another popular data visualization library in Python. Seaborn provides a high-level interface for creating beautiful and informative statistical graphics. It comes with several pre-built themes and color palettes that make it easy to create aesthetically pleasing visualizations.

Unfortunately, while seaborn doesn’t have a direct method for Pie Charts, we can use it in conjunction with Matplotlib to create Pie Charts. Let’s explore this in a blog!

What is Seaborn and why use it for data visualization?

Seaborn (https://seaborn.pydata.org/) is a Python data visualization library that is built on top of matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. Seaborn allows users to quickly explore and understand their data by providing a wide range of visualizations, including scatterplots, line plots, bar plots, heatmaps, and pie charts.

One of the main advantages of using Seaborn for data visualization is its ability to create aesthetically pleasing charts with minimal code. Seaborn provides a number of pre-defined styles and color palettes that can be easily applied to any plot. This means that even beginners can create professional-looking visualizations without spending too much time on design.

Another advantage of Seaborn is its integration with pandas, which is the most popular data manipulation library in Python. This makes it easy to visualize data stored in pandas DataFrames or Series objects. Seaborn also has built-in support for working with categorical variables, making it easy to create charts that show relationships between different groups of data.

In summary, Seaborn is a powerful tool for creating informative and visually appealing data visualizations in Python. Its ease of use, flexibility, and integration with other Python libraries make it an ideal choice for anyone looking to explore and communicate their data effectively.

Pie Chart: An overview

Pie charts (https://en.wikipedia.org/wiki/Pie_chart) are a popular way of representing data in a visual form. They are circular graphs that are divided into slices, with each slice representing a proportion of the whole. Pie charts are commonly used to show how much each category contributes to the overall data set.

In Python, we can use the Seaborn library to create pie charts. Seaborn is a popular data visualization library that is built on top of Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics.

To create a pie chart using Seaborn, we first need to import the library and load a dataset that we want to visualize. We can then use the `pieplot()` function from Seaborn to generate the chart.

Here’s an example of how to create a simple pie chart using Seaborn:

```
import seaborn as sns
import matplotlib.pyplot as plt

data = [30, 20, 50] # Sample data
labels = ['A', 'B', 'C'] # Labels for each slice

sns.set_style("whitegrid") # Set style for chart
plt.figure(figsize=(6,6)) # Set figure size
plt.pie(data, labels=labels) # Create pie chart
plt.show() # Show chart
```

In this example, we have created a pie chart with three slices representing the values 30, 20, and 50. We have also added labels for each slice using the `labels` parameter.

Seaborn provides several customization options for pie charts, such as changing the colors of slices or adding a legend. These options can be accessed through various parameters of the `pieplot()` function.

Overall, pie charts are a useful tool for visualizing proportions in datasets. With Seaborn, creating informative and attractive pie charts is easy and straightforward.

Creating a pie chart with Seaborn

Pie charts are a popular way to represent data in a clear and concise manner. Seaborn, a Python data visualization library, offers an easy and intuitive way to create stunning pie charts.

To create a pie chart with Seaborn, we first need to import the necessary libraries:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Next, we need to load our data into a pandas DataFrame:

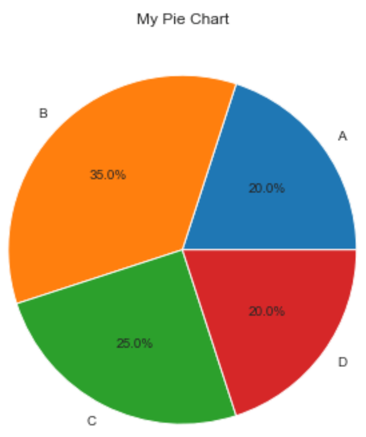
```
import pandas as pd

data = {'labels': ['A', 'B', 'C', 'D'],
        'values': [20, 35, 25, 20]}
df = pd.DataFrame(data)
```

In this example, we have four categories labeled A, B, C, and D with corresponding values of 20, 35, 25, and 20.

Now we can create our pie chart using Seaborn's `pieplot()` function:

```
sns.set_style("whitegrid")
plt.figure(figsize=(6,6))
plt.pie(df['values'], labels=df['labels'], autopct='%1.1f%%')
plt.title('My Pie Chart')
plt.show()
```



Here, we set the style of our plot to "whitegrid" for a clean look. We also specify the size of our plot with `figsize=(6,6)`.

The `pie()` function creates the actual pie chart. We pass in the values from our DataFrame's "values" column and use the "labels" column for the category labels on each slice of the pie. The `autopct` parameter specifies how to format the percentage values displayed on each slice.

Finally, we add a title to our plot with `plt.title()` and display it with `plt.show()`.

With just a few lines of code using Seaborn's `pieplot()` function, we can easily create an informative and visually appealing pie chart.

Customizing the pie chart

Pie charts are a great way to visualize data in a way that is easy to understand. Seaborn provides a simple way to create pie charts using the `pieplot()` function.

Once you have created your pie chart, you can customize it to make it more visually appealing and easier to read. Here are some ways to customize your Seaborn pie chart:

1. Adding a title: You can add a title to your pie chart using the `title()` function. This will make it easier for viewers to understand what the chart is about.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create pie chart
data = [10, 20, 30, 40]
labels = ['A', 'B', 'C', 'D']
sns.set_style("darkgrid")
plt.pie(data, labels=labels)

# Add title
plt.title("Distribution of Data")

# Show plot
plt.show()
```

2. Changing the colors: Seaborn provides a set of default colors for pie charts, but you can also change the colors by specifying a list of colors using the `colors` parameter.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create pie chart
data = [10, 20, 30, 40]
labels = ['A', 'B', 'C', 'D']
colors = ['#ff9999','#66b3ff','#99ff99','#ffcc99']
sns.set_style("darkgrid")
plt.pie(data, labels=labels, colors=colors)

# Add title
plt.title("Distribution of Data")

# Show plot
plt.show()
```

3. Exploding slices: You can highlight specific slices of your pie chart by "exploding" them away from the center of the chart. This is done by specifying a list of values for the `explode` parameter.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create pie chart
data = [10, 20, 30, 40]
labels = ['A', 'B', 'C', 'D']
explode = (0, 0.1, 0, 0)
sns.set_style("darkgrid")
plt.pie(data, labels=labels, explode=explode)

# Add title
plt.title("Distribution of Data")

# Show plot
plt.show()
```

By customizing your Seaborn pie chart, you can create a more visually appealing and informative visualization of your data.

Exploding the slices of the pie chart

Pie charts are a great way to show the composition of a dataset, but sometimes you might want to highlight a particular slice of the chart. This is where exploding the slices comes in handy.

In Seaborn, you can explode one or more slices of a pie chart by passing a list of values to the `explode` parameter of the `pieplot()` function. The values in the list correspond to the degree of separation between each slice and the center of the chart.

Let's take an example where we have a pie chart that shows the percentage of sales for different products in a store. We want to highlight the slice for our best-selling product, which is "Shoes". We can do this by exploding that slice slightly away from the center:

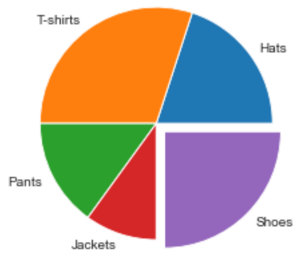
```
import seaborn as sns
import matplotlib.pyplot as plt

# Data
sales_data = [20, 30, 15, 10, 25]
products = ['Hats', 'T-shirts', 'Pants', 'Jackets', 'Shoes']

# Explode Shoes slice
explode = [0, 0, 0, 0, 0.1]

# Create pie chart
plt.pie(sales_data, labels=products, explode=explode)

# Show plot
plt.show()
```



In this example code block, we first import `seaborn` and `matplotlib.pyplot`. Then we define our data and labels for the pie chart. We create a list `explode` with five elements since we have five slices in our chart. The value of `0` means that we don't want to explode that slice while `0.1` means we want to move it slightly away from the center.

Finally, we pass `sales_data`, `products`, and `explode` to the `pieplot()` function and display our chart using `show()`.

By exploding a particular slice in our pie chart, we can draw attention to it and make our data visualization more impactful.

Adding labels to the slices of the pie chart

Labels are a crucial aspect of pie charts, as they help us understand the data represented in each slice. Seaborn makes it easy to add labels to our pie chart using the `pieplot()` function from the `seaborn` library.

To add labels to our pie chart, we first need to create a list of labels that correspond to the data points in our chart. We can then pass this list as a parameter to the `pieplot()` function's `labels` argument.

Let’s say we have a dataset that contains information about the number of pets owned by different people. We want to create a pie chart that shows the percentage of people who own cats, dogs, and birds. Here’s how we can create a labeled pie chart using Seaborn:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Define our dataset
pets = ['Cats', 'Dogs', 'Birds']
counts = [20, 40, 10]

# Create our pie chart with labels
plt.figure(figsize=(6,6))
sns.set_style("whitegrid")
plt.pie(counts, labels=pets)

# Add a title
plt.title("Percentage of People Who Own Cats, Dogs, and Birds")

# Display the plot
plt.show()
```

In this example, we first define our dataset as two lists: `pets` and `counts`. The `pets` list contains the names of each category (cats, dogs, and birds), while the `counts` list contains the corresponding count for each category.

We then create our pie chart using `plt.pie()`, passing in the `counts` list as a parameter. To add labels to our slices, we pass in the `pets` list to the `labels` parameter.

Finally, we add a title to our plot using `plt.title()`, and display it using `plt.show()`.

With just a few lines of code, we have created a labeled pie chart that clearly shows the percentage of people who own cats, dogs, and birds.

Conclusion

In conclusion, the Seaborn library provides a simple yet powerful way of creating visually appealing pie charts for data visualization. With just a few lines of code, we were able to create a pie chart that effectively conveyed the distribution of data.

It is important to keep in mind that pie charts should be used sparingly and only when appropriate. They work best when there are only a few categories and each slice represents a significant portion of the whole. If there are too many categories or if the slices are too small, it becomes difficult to interpret the chart accurately.

Overall, Seaborn offers a wide range of customization options that allow you to create pie charts that match your specific needs. Whether you need to change the colors, labels, or even the size of the chart, Seaborn makes it easy to do so.

In summary, with Seaborn’s intuitive API and powerful visualization capabilities, creating effective and visually appealing pie charts has never been easier.
Interested in learning more? Check out our Introduction to Python (https://pieriantraining.com/learn/introduction-to-python/) course!

PIERIAN TRAINING

Pierian Training is a leading provider of high-quality technology training, with a focus on data science and cloud computing. Pierian Training offers live instructor-led training, self-paced online video courses, and private group and cohort training programs to support enterprises looking to upskill their employees.

You May Also Like

(https://pieriantraining.com/guide-to-nltk-natural-language-toolkit-for-python/)

DATA SCIENCE (HTTPS://PIERIANTRAINING.COM/DATA-SCIENCE/), TUTORIALS (HTTPS://PIERIANTRAINING.COM/TUTORIALS/)

Guide to NLTK – Natural Language Toolkit for Python (https://pieriantraining.com/guide-to-nltk-natural-language-toolkit-for-python/)

Introduction Natural Language Processing (NLP) lies at the heart of countless applications we use every day, from voice assistants to spam filters and machine translation. It allows machines to

(https://pieriantraining.com/gridse with-scikit-learn-and-python/)

MACHINE LEARNING (HTTPS://PIERIANTRAINING.COM/MACHINE-LEARNING/), TUTORIALS (HTTPS://PIERIANTRAINING.COM/TUTORIALS/)

GridSearchCV with Scikit-Learn and Python (https://pieriantraining.com/gridsearchcv-with-scikit-learn-and-python/)

Introduction In the world of machine learning, finding the optimal set of hyperparameters for a model can significantly impact its performance and accuracy. However, searching through all possible combinations manually can be an incredibly time-consuming and error-prone

(https://pieriantraining.com/plottin time-series-in-python-a-complete-guide/)

PYTHON BASICS (HTTPS://PIERIANTRAINING.COM/PYTHON-BASICS/), TUTORIALS (HTTPS://PIERIANTRAINING.COM/TUTORIALS/)

Plotting Time Series in Python: A Complete Guide (https://pieriantraining.com/p time-series-in-python-a-complete-guide/)

Introduction Time series data is a type of data that is collected over time at regular intervals. It can be used to analyze trends, patterns, and behaviors over time. In order to effectively analyze time series data, it is important to visualize it in a way understand, interpret, and generate human language, bridging the gap between humans and computers. Within the vast landscape of NLP tools and techniques, the Natural Language Toolkit [...]

PIERIAN TRAINING

Read Post (https://pieriantraining.com/guide-to-nltk-natural-language-toolkit-for-python/)

process. This is where GridSearchCV, a powerful tool provided by Scikit-Learn library in Python, comes to the rescue. [...]

PIERIAN TRAINING

Read Post (https://pieriantraining.com/gridsearchcv-with-scikit-learn-and-python/)

that is easy to understand. This is where plotting [...]

PIERIAN TRAINING

Read Post (https://pieriantraining.com/plotting-time-series-in-python-a-complete-guide/)