Data Visualization Guide

Vol 1

Data Visualization

Data visualization is the graphic representation of data. It involves producing images that communicate relationships among the represented data to viewers. Visualizing data is an essential part of data analysis and machine learning, but choosing the right type of visualization is often challenging. This guide provides an introduction to popular data visualization techniques, by presenting sample use cases and providing code examples using Python.

Types of graphs covered:

- Line graph
- Scatter plot
- Histogram and Frequency Distribution
- Heatmap
- Contour Plot
- Box Plot
- Bar Chart

Import libraries

- [Matplotlib](): Plotting and visualization library for Python. We'll use the `pyplot` module from `matplotlib`. As convention, it is often imported as `plt`.

- [Seaborn](): An easy-to-use visualization library that builds on top of Matplotlib and lets you create beautiful charts with just a few lines of code.

```
# Uncomment the next line to install the required libraries
# !pip install matplotlib seaborn --upgrade --quiet
```

```
# Import libraries
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Configuring styles
sns.set_style("darkgrid")
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

Line Chart

A line chart displays information as a series of data points or markers, connected by a straight lines. You can customize the shape, size, color and other aesthetic elements of the markers and lines for better visual clarity.

Example

We'll create a line chart to compare the yields of apples and oranges over 12 years in the imaginary region of Hoenn.

```
# Sample data
years = range(2000, 2012)
apples = [0.895, 0.91, 0.919, 0.926, 0.929, 0.931, 0.934, 0.936, 0.937, 0.9375, 0.9372,
oranges = [0.962, 0.941, 0.930, 0.923, 0.918, 0.908, 0.907, 0.904, 0.901, 0.898, 0.9, 0

# First line
plt.plot(years, apples, 'b-x', linewidth=4, markersize=12, markeredgewidth=4, markeredg

# Second line
plt.plot(years, oranges, 'r--o', linewidth=4, markersize=12,);

# Title
plt.title('Crop Yields in Hoenn Region')

# Line labels
plt.legend(['Apples', 'Oranges'])

# Axis labels
plt.xlabel('Year'); plt.ylabel('Yield (tons)');
```
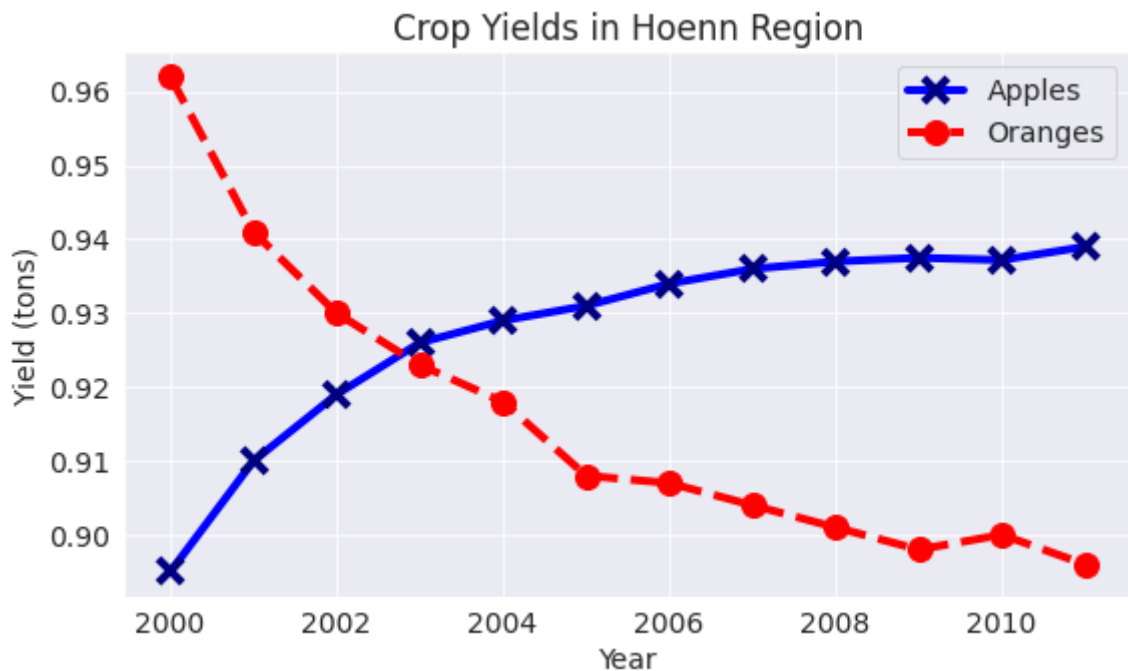
Crop Yields in Hoenn Region

Scatter Plot

In a scatter plot, the values of 2 variables are plotted as points on a 2-dimensional grid. Additonally, you can also use a third variable to determine the size or color of the points.

Example

The Iris flower dataset provides samples measurements of sepals and petals for 3 species of flowers. The Iris dataset is included with the `seaborn` library, and can be loaded as a `pandas` dataframe.

```
# Load data into a Pandas dataframe
data = sns.load_dataset("iris")

# View the data
data.sample(5)
```

	sepal_length	sepal_width	petal_length	petal_width	species
136	6.3	3.4	5.6	2.4	virginica
16	5.4	3.9	1.3	0.4	setosa
36	5.5	3.5	1.3	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
89	5.5	2.5	4.0	1.3	versicolor

We can use a scatter plot to visualize sepal length & sepal witdh vary across different species of flowers. The points for each species form a separate cluster, with some overlap between the Versicolor and Virginica species.

```
# Create a scatter plot
sns.scatterplot(data.sepal_length, # X-axis
                data.sepal_width,  # Y-axis
                hue=data.species,  # Dot color
                s=100);
```

```
# Chart title
plt.title("Flowers");
```


Flowers
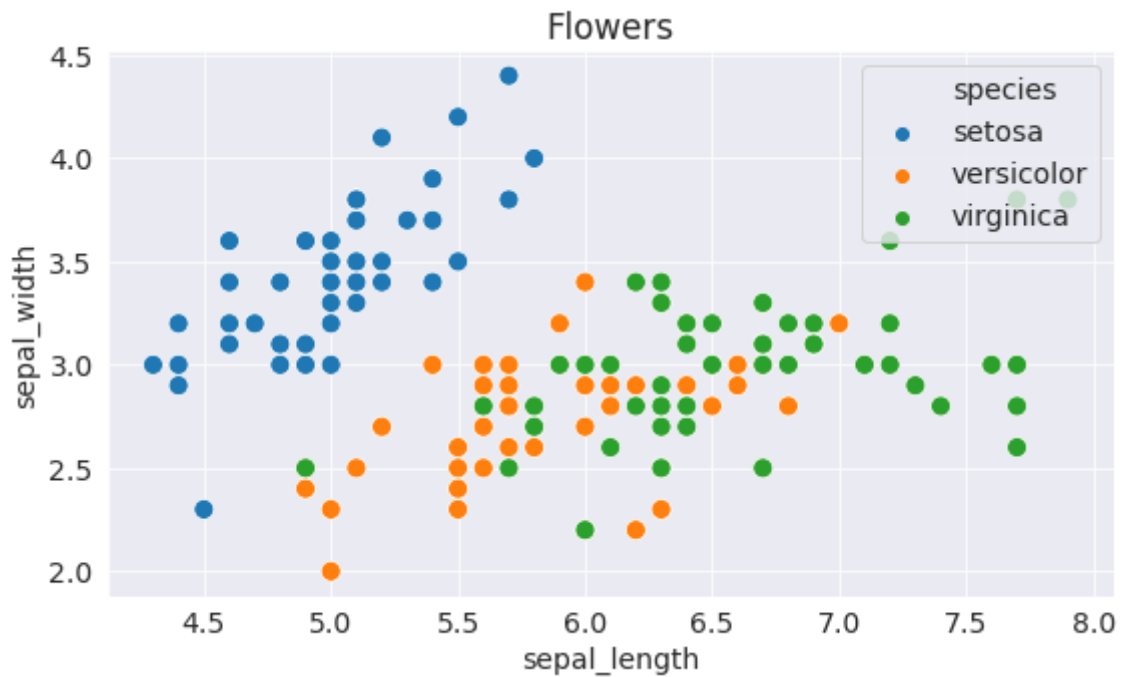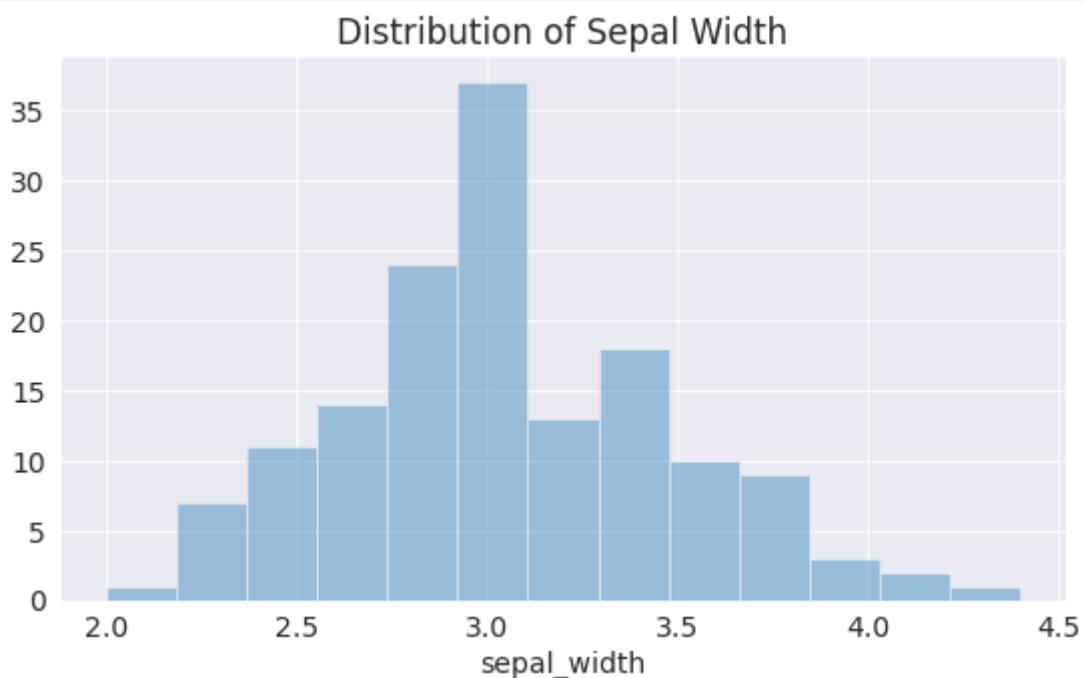
Histogram and Frequency Distribution

A histogram represents the distribution of data by forming bins along the range of the data and then drawing bars to show the number of observations that fall in each bin.

Example

We can use a histogram to visualize how the values of sepal width are distributed.

```
plt.title("Distribution of Sepal Width")

sns.distplot(data.sepal_width, kde=False);
```


Distribution of Sepal Width

We can immediately see that values of sepal width fall in the range 2.0 - 4.5, and around 35 values are in the range 2.9 - 3.1. We can also look at this data as a frequency distribution, where the values on Y-axis are percentagess instead of counts.

```
plt.title("Distribution of Sepal Width")

sns.distplot(data.sepal_width);
```
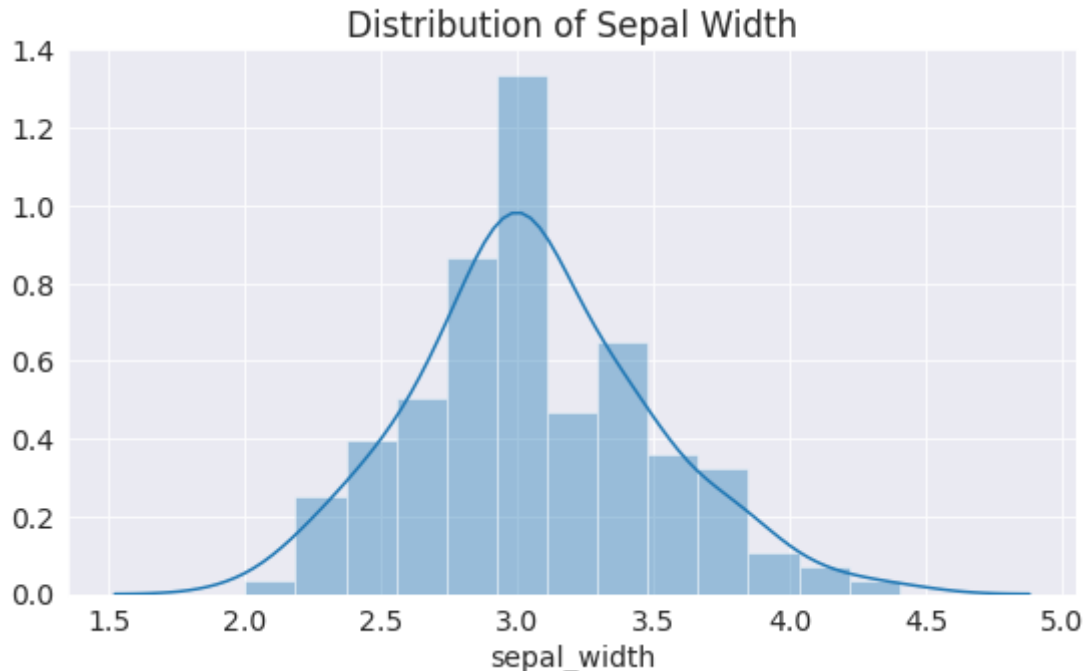

Heatmap

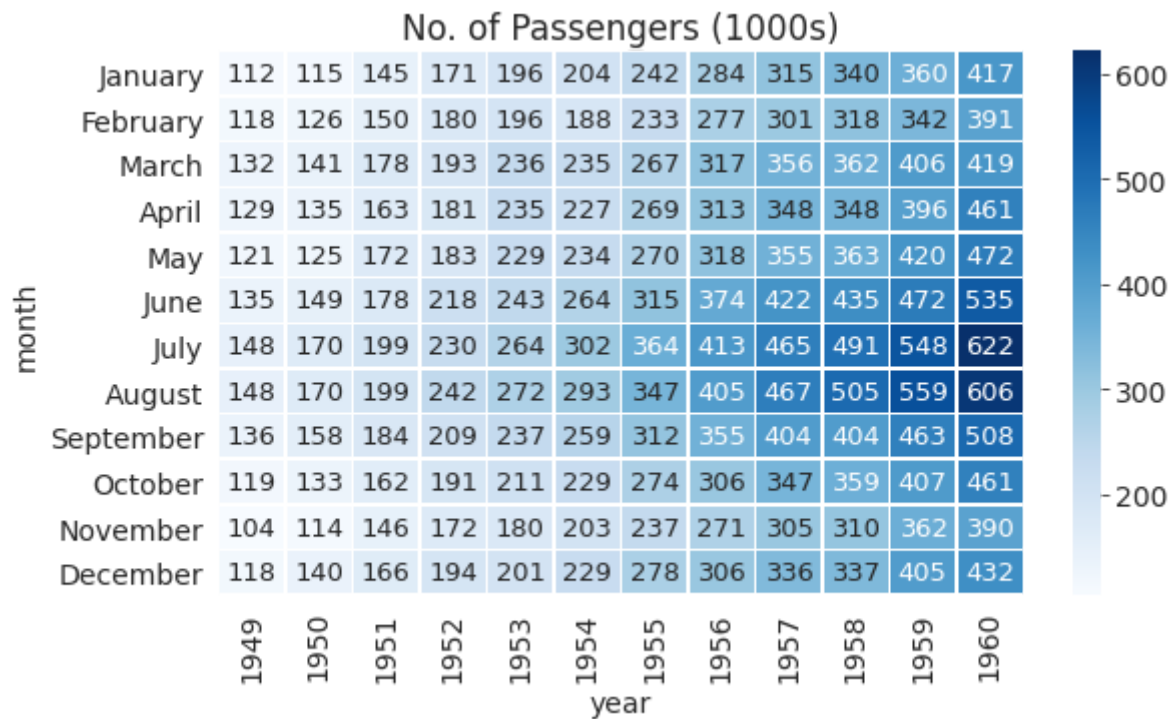A heatamp is used to visualize 2-dimensional data like a matrix or a table using colors.

Example

We'll use another sample dataset from Seaborn, called "flights", to visualize monthly passenger footfall at an airport over 12 years.

```
# Load the example flights dataset as a matrix
flights = sns.load_dataset("flights").pivot("month", "year", "passengers")

# Chart Title
plt.title("No. of Passengers (1000s)")

# Draw a heatmap with the numeric values in each cell
sns.heatmap(flights,
            fmt="d",
            annot=True,
            linewidths=.5,
            cmap='Blues',
            annot_kws={"fontsize":13});
```

No. of Passengers (1000s)

month	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
June	135	149	178	218	243	264	315	374	422	435	472	535
July	148	170	199	230	264	302	364	413	465	491	548	622
August	148	170	199	242	272	293	347	405	467	505	559	606
September	136	158	184	209	237	259	312	355	404	404	463	508
October	119	133	162	191	211	229	274	306	347	359	407	461
November	104	114	146	172	180	203	237	271	305	310	362	390
December	118	140	166	194	201	229	278	306	336	337	405	432

Contour Plot

Contour plot uses contours or color-coded regions helps us to visualize 3 numerical variables in two dimensions. One variable is represented on the horizontal axis and a second variable is represented on the vertical axis. The third variable is represented by a color gradient and isolines (lines of constant value).

Example

We can visulize the values of sepal width & sepal length from the flowers dataset using a contour plot. The shade of blue represent the density of values in a region of the graph.

```
plt.title("Flowers")

sns.kdeplot(data.sepal_length, data.sepal_width, shade=True, shade_lowest=False);
```
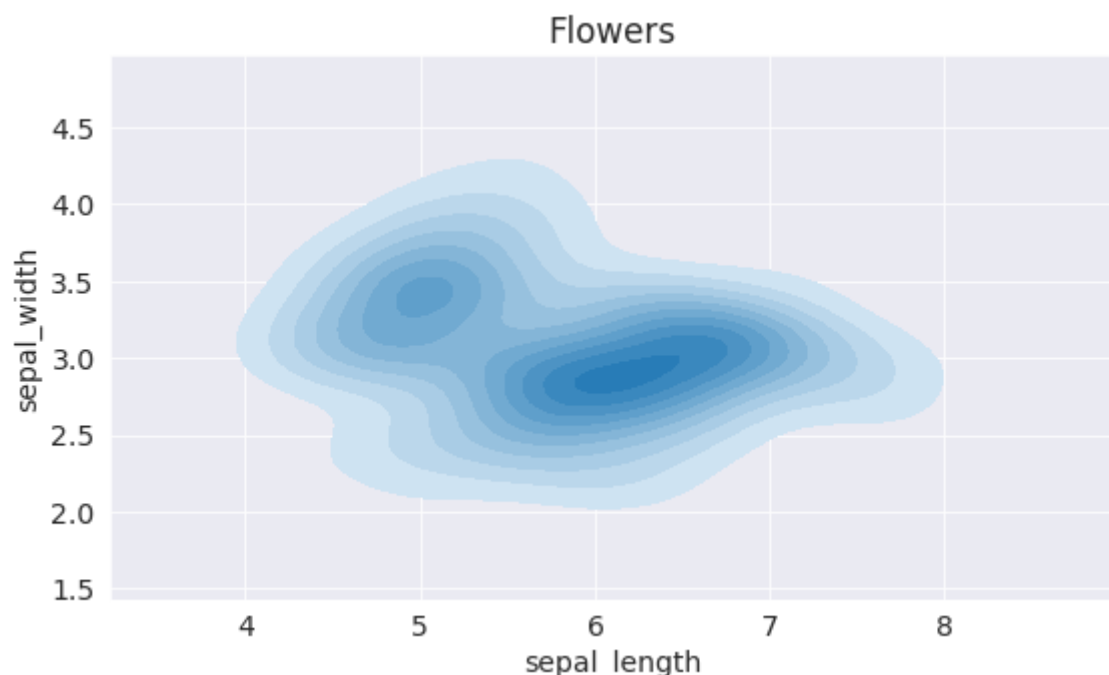
We can segment speicies of flowers by creating multiple contour plots with different colors.

```
setosa = data[data.species == 'setosa']
virginica = data[data.species == 'virginica']

plt.title("Flowers (Setosa & Virginica)")

sns.kdeplot(setosa.sepal_length, setosa.sepal_width, shade=True, cmap='Reds', shade_low
sns.kdeplot(virginica.sepal_length, virginica.sepal_width, shade=True, cmap='Blues', sh
```
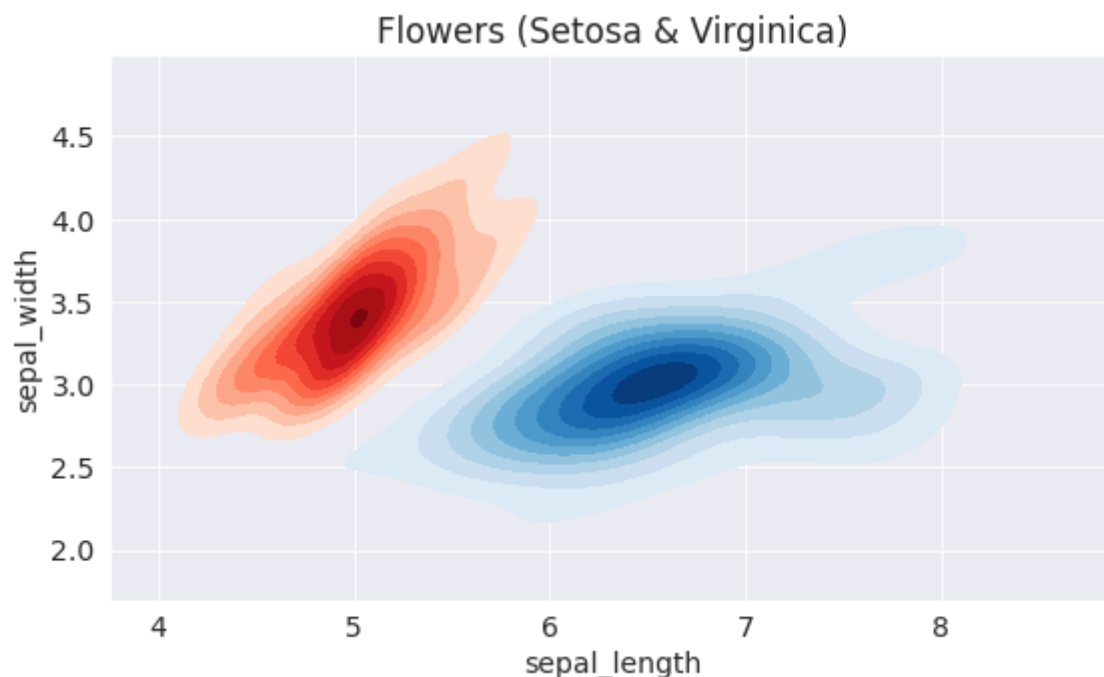

Box Plot

A box plot shows the distribution of data along a single axis, using a "box" and "whiskers". The lower end of the box represents the 1st quartile (i.e. 25% of values are below it), and the upper end of the box represents the 3rd quartile (i.e. 25% of values are above it). The median value is represented via a line inside the box. The "whiskers" represent the minimum & maximum values (sometimes excluding outliers, which are represented as dots).

Example

We'll use another sample dataset included with Seaborn, called "tips". The dataset contains information about the sex, time of day, total bill and tip amount for customers visiting a restraurant over a week.

```
# Load the example tips dataset
tips = sns.load_dataset("tips");
tips
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2

	total_bill	tip	sex	smoker	day	time	size
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

We can use a box plot to visualize the distribution of total bill for each day of the week, segmented by whether the customer was a smoker.

```
# Chart title
plt.title("Daily Total Bill")

# Draw a nested boxplot to show bills by day and time
sns.boxplot(tips.day, tips.total_bill, hue=tips.smoker);
```
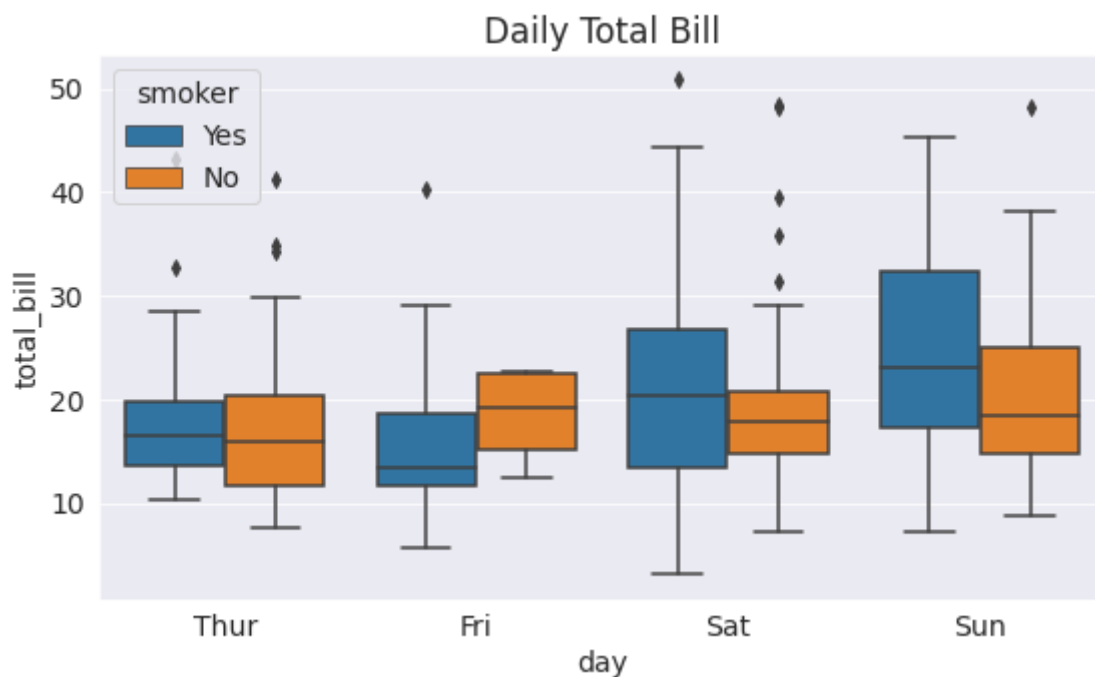

Bar Chart

A bar chart presents categorical data with rectangular bars with heights proportional to the values that they represent. If there are multiple values for each category, then a bar plot can also represent the average value, with confidence intervals.

Example

We can use a bar chart visulize the average value of total bill for different days of the week, segmented by sex, for the "tips" dataset

```
sns.barplot(x="day", y="total_bill", hue="sex", data=tips);
```
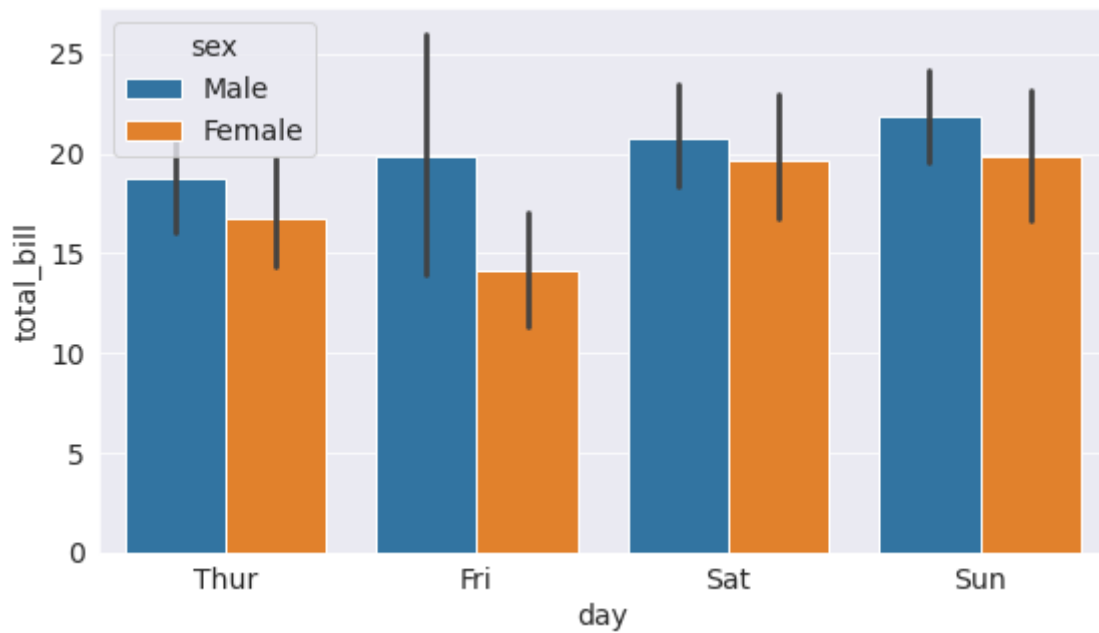

Further Reading

This guide intends to serve as introduction to the most commonly used data visualization techniques. With minor modifications to the examples shown above, you can visualize a wide variety of datasets. Visit the official documentation websites for more examples & tutorials:

- Seaborn: https://seaborn.pydata.org/tutorial.html
- Matplotlib: https://matplotlib.org/tutorials/index.html

To share your data visualzations online, just install the Jovian python library and run `jovian.commit`.

```
!pip install jovian --upgrade --quiet
```

```
import jovian
```

```
jovian.commit(project='dataviz-cheatsheet')
```

```
[jovian] Attempting to save notebook..
```