

Pengenalan Library Seaborn

Seaborn adalah library untuk visualisasi data yang dibangun di atas matplotlib. Seaborn memiliki lebih banyak fungsi untuk visualisasi data dan lebih mudah digunakan

Untuk menggunakan library seaborn kita harus install library ini terlebih dahulu dengan menggunakan PIP

```
pip install seaborn
```

Untuk menggunakannya import terlebih dahulu dengan perintah seperti ini

```
import seaborn as sns
```

Dalam Library Seaborn terdapat banyak fungsi grafik yang bisa digunakan untuk melakukan visualisasi data

Ada beberapa jenis plot yang dapat dilakukan dengan seaborn seperti

- Categorical plots
- Distribution plots
- Relational plots
- Regression plots
- Matrix plots

Pengenalan Library Seaborn

Seaborn adalah library untuk visualisasi data yang dibangun di atas matplotlib. Seaborn memiliki lebih banyak fungsi untuk visualisasi data dan lebih mudah digunakan

Untuk menggunakan library seaborn kita harus install library ini terlebih dahulu dengan menggunakan PIP

```
pip install seaborn
```

Untuk menggunakannya import terlebih dahulu dengan perintah seperti ini

```
import seaborn as sns
```

Dalam Library Seaborn terdapat banyak fungsi grafik yang bisa digunakan untuk melakukan visualisasi data

Ada beberapa jenis plot yang dapat dilakukan dengan seaborn seperti

- Categorical plots
- Distribution plots
- Relational plots
- Regression plots
- Matrix plots

Perbedaan Antara Matplotlib VS Seaborn

Visualisasi Data adalah representasi grafik dari data. Ini mengubah kumpulan data besar menjadi grafik kecil, sehingga membantu dalam analisis dan prediksi data. Ini adalah elemen yang sangat diperlukan dari ilmu data yang membuat data kompleks lebih mudah dipahami dan diakses. Matplotlib dan Seaborn bertindak sebagai tulang punggung visualisasi data melalui Python.

Matplotlib: Ini adalah pustaka Python yang digunakan untuk merencanakan grafik dengan bantuan pustaka lain seperti Numpy dan Pandas. Ini adalah alat yang ampuh untuk memvisualisasikan data dengan Python. Ini digunakan untuk membuat interferensi statis dan merencanakan grafik array 2D. Ini pertama kali diperkenalkan oleh John D. Hunter pada tahun 2002. Ini menggunakan Pyplot untuk menyediakan MATLAB seperti antarmuka gratis dan open-source. Ia mampu menangani berbagai sistem operasi dan backend grafisnya.

Seaborn: Ini juga merupakan pustaka Python yang digunakan untuk merencanakan grafik dengan bantuan Matplotlib, Pandas, dan Numpy. Itu dibangun di atas Matplotlib dan dianggap sebagai superset dari perpustakaan Matplotlib. Ini membantu dalam memvisualisasikan data univariat dan bivariat. Ini menggunakan tema yang indah untuk mendekorasi grafik Matplotlib. Ini bertindak sebagai alat penting dalam menggambarkan Model Regresi Linier. Ini berfungsi dalam membuat grafik data Time-Series statis. Ini menghilangkan tumpang tindih grafik dan juga membantu mempercantiknya.

| | | |
|-----------------|---|--|
| Kegunaan | Ini digunakan untuk membuat grafik dasar. Kumpulan data divisualisasikan dengan bantuan grafik batang, histogram, diagram grafik, plot pencar, garis, dan sebagainya. | Seaborn berisi sejumlah pola dan plot untuk visualisasi data. Ini menggunakan tema yang menarik. Ini membantu dalam mengumpulkan seluruh data menjadi satu plot. Ini juga menyediakan distribusi data. |
|-----------------|---|--|

| | | |
|------------------|---|---|
| Sintaksis | Ini menggunakan sintaks yang relatif kompleks dan panjang. Contoh: Sintaks untuk bargraph- matplotlib.pyplot.bar(x_axis, y_axis). | Ini menggunakan sintaks yang relatif sederhana yang lebih mudah dipelajari dan dipahami. Contoh: Sintaks untuk bargraph- seaborn.barplot(sumbu_x, sumbu_y). |
|------------------|---|---|

| | | |
|-------------------------------|---|---|
| Menangani Banyak Angka | Kami dapat membuka dan menggunakan banyak gambar secara bersamaan. Namun mereka ditutup dengan jelas. Sintaks untuk menutup gambar satu per satu: matplotlib.pyplot.close(). Sintaks untuk menutup semua gambar: matplotlib.pyplot.close("semua") | Seaborn menetapkan waktu untuk pembuatan setiap gambar. Namun, ini dapat menyebabkan masalah memori (OOM) habis |
|-------------------------------|---|---|

| | | |
|--------------------|--|--|
| Visualisasi | Matplotlib terhubung dengan baik dengan Numpy dan Pandas dan bertindak sebagai paket grafik untuk visualisasi data dalam python. Pyplot menyediakan fitur dan sintaks yang serupa seperti di MATLAB. Oleh karena itu, pengguna MATLAB dapat dengan mudah mempelajarinya. | Seaborn lebih nyaman dalam menangani bingkai data Pandas. Ini menggunakan set metode dasar untuk memberikan grafik yang indah dengan python. |
|--------------------|--|--|

| | | |
|---------------------|--|--|
| Sifat Lembut | Matplotlib sangat disesuaikan dan kuat | Seaborn menghindari tumpang tindih plot dengan bantuan tema defaultnya |
|---------------------|--|--|

| | | |
|-------------------------------|--|--|
| Bingkai dan Array Data | Matplotlib bekerja secara efisien dengan bingkai dan larik data. Ini memperlakukan figur dan ace sebagai objek. Ini berisi berbagai API stateful untuk plotting. Oleh karena itu, metode seperti plot() dapat bekerja tanpa parameter. | Seaborn jauh lebih fungsional dan terorganisir daripada Matplotlib dan memperlakukan seluruh dataset sebagai satu kesatuan. Seaborn tidak begitu stateful dan oleh karena itu, parameter diperlukan saat memanggil metode seperti plot() |
|-------------------------------|--|--|

| | | |
|----------------------|---|---|
| Gunakan Kasus | Matplotlib memplot berbagai grafik menggunakan Pandas dan Numpy | Seaborn adalah versi diperpanjang dari Matplotlib yang menggunakan Matplotlib bersama dengan Numpy dan Pandas untuk merencanakan grafik |
|----------------------|---|---|

```
In [1]: import seaborn as sns
import numpy as np
from numpy import median
import pandas as pd
import matplotlib.pyplot as plt
matplotlib inline
```

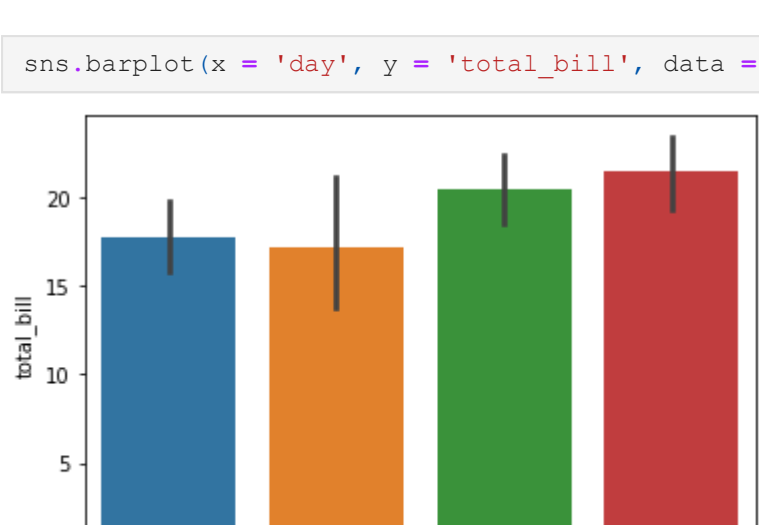
```
In [2]: tips = sns.load_dataset("tips")
```

```
In [3]: tips.head()
```

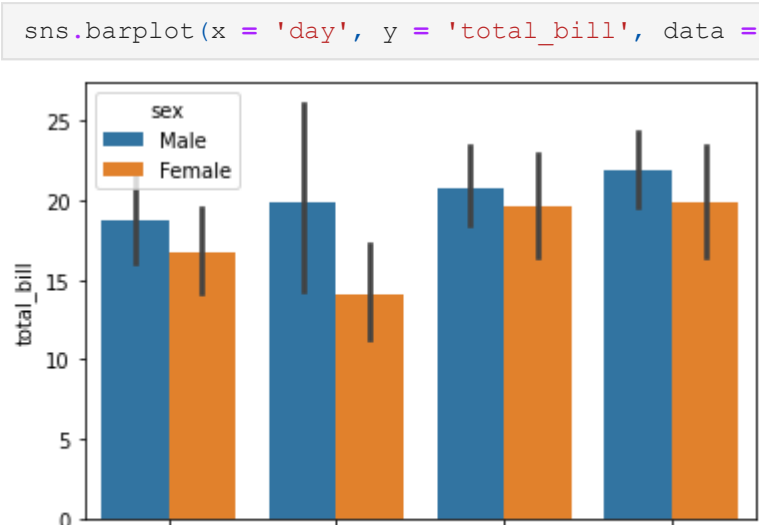
```
Out[3]:   total_bill  tip    sex  smoker  day  time  size
0      16.99   1.01  Female     No   Sun  Dinner     2
1      10.34   1.66   Male     No   Sun  Dinner     3
2      21.01   3.50   Male     No   Sun  Dinner     3
3      23.68   3.31   Male     No   Sun  Dinner     2
4      24.59   3.61  Female     No   Sun  Dinner     4
```

```
In [4]: # Bar Plot
sns.barplot(x = 'day', y = 'tip', data = tips)
```

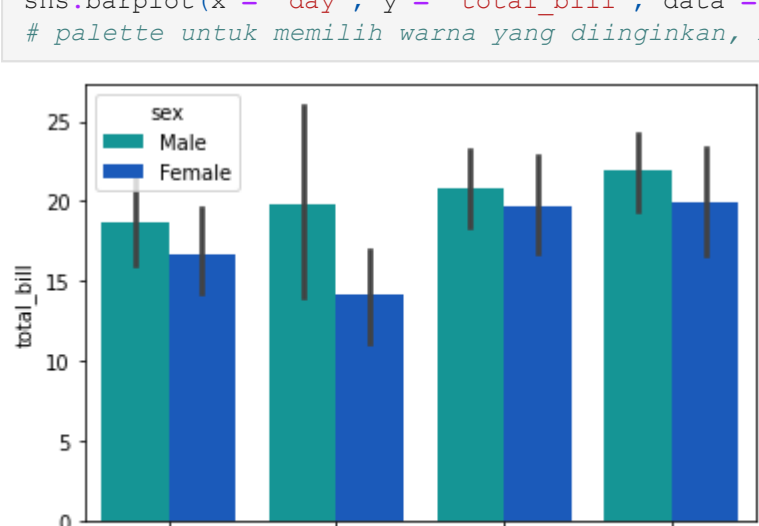
```
Out[4]: <AxesSubplot:xlabel='day', ylabel='tip'>
```



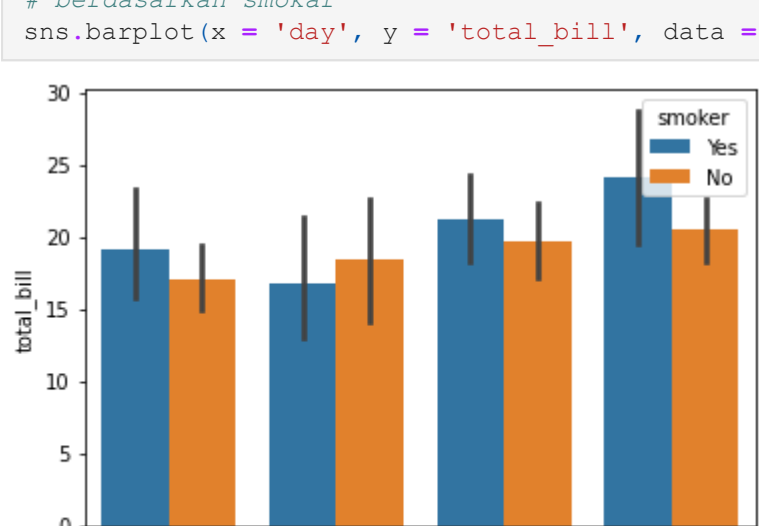
```
In [5]: # untuk menghilangkan tulisan <AxesSubplot:xlabel='day', ylabel='tip'>, pake titik koma di akhir code
sns.barplot(x = 'day', y = 'tip', data = tips);
```



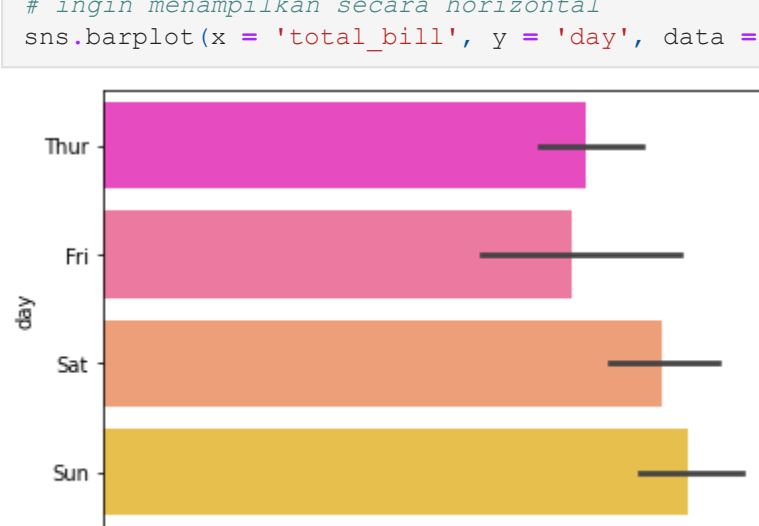
```
In [6]: sns.barplot(x = 'day', y = 'total_bill', data = tips);
```



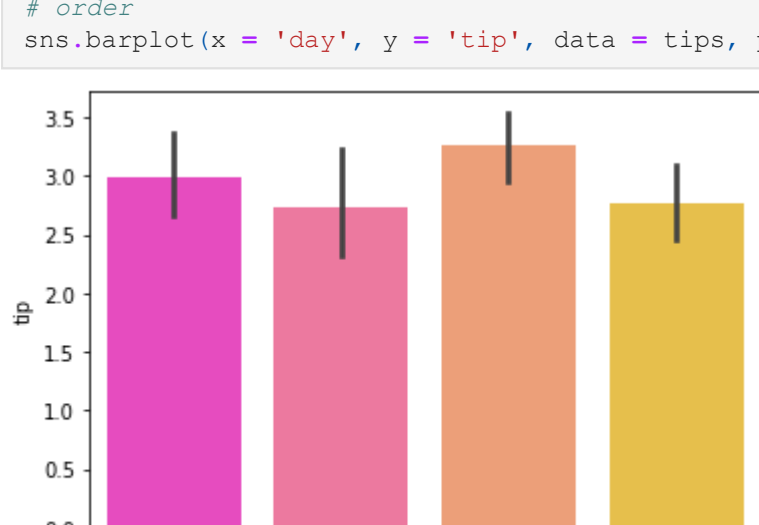
```
In [7]: sns.barplot(x = 'day', y = 'total_bill', data = tips, hue = 'sex');
```



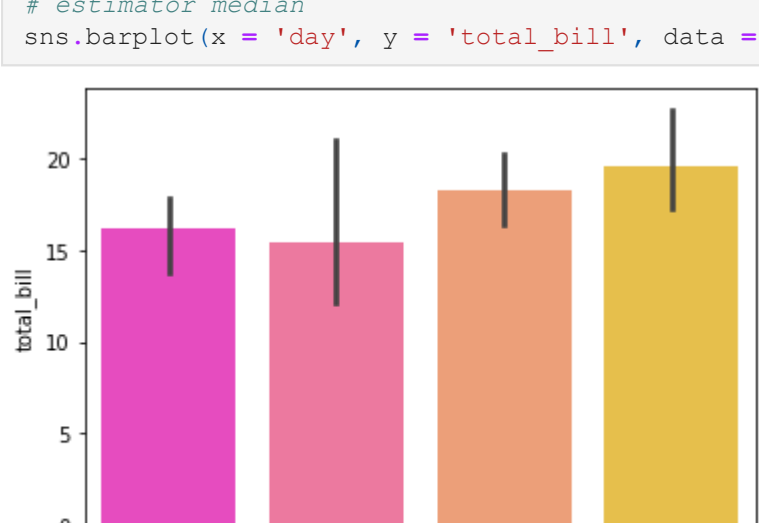
```
In [8]: sns.barplot(x = 'day', y = 'total_bill', data = tips, hue = 'sex', palette = 'winter_r');
# palette untuk memilih warna yang diinginkan, hue untuk membagi, data untuk mengambil data yang ingin divisual.
```



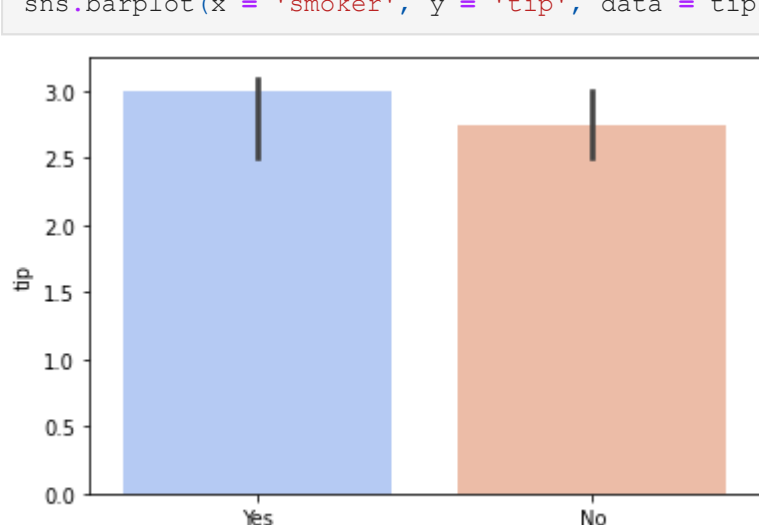
```
In [9]: # berdasarkan smoker
sns.barplot(x = 'day', y = 'total_bill', data = tips, hue = 'smoker');
```



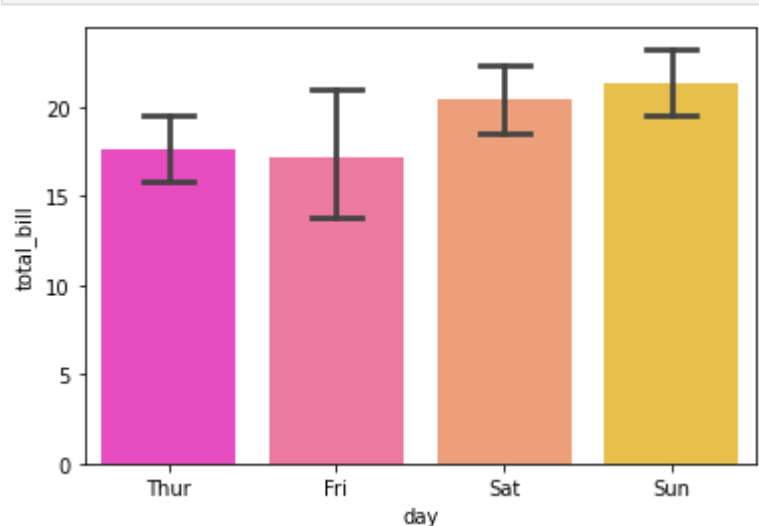
```
In [10]: # ingin menampilkan secara horizontal
sns.barplot(x = 'total_bill', y = 'day', data = tips, palette = 'spring');
```



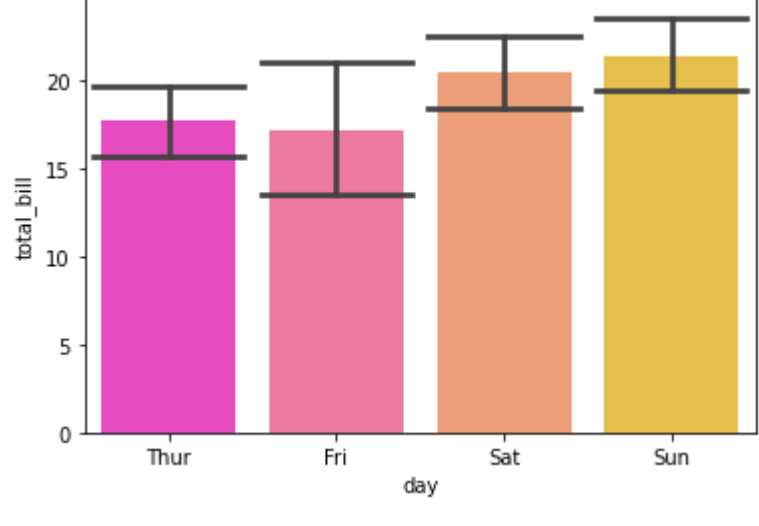
```
In [11]: # order
sns.barplot(x = 'day', y = 'tip', data = tips, palette = 'spring', order = ['Sat', 'Fri', 'Sun', 'Thur']);
```



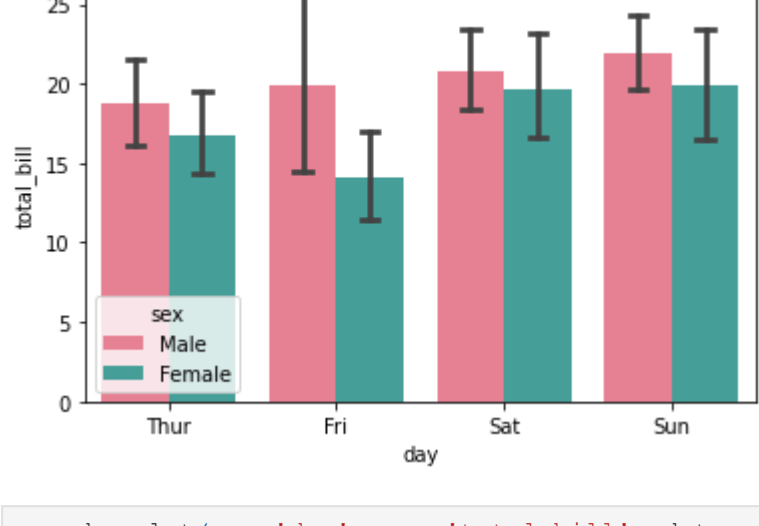
```
In [12]: # estimator median
sns.barplot(x = 'day', y = 'total_bill', data = tips, estimator = median, palette = 'spring');
```



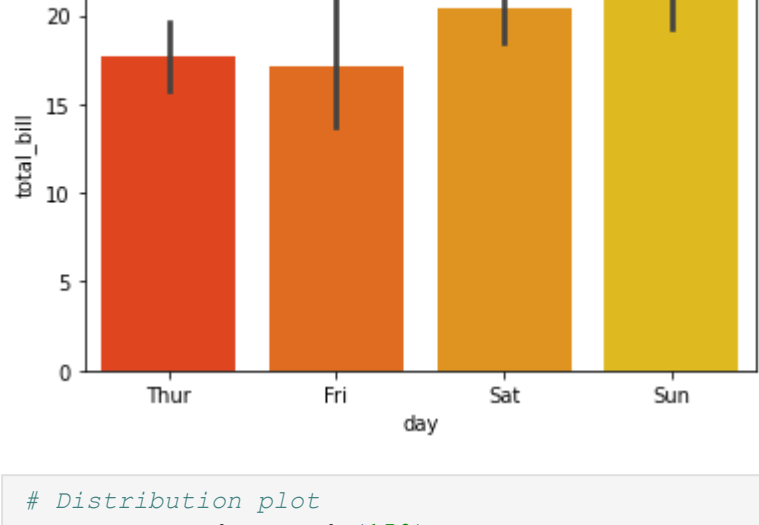
```
In [13]: sns.barplot(x = 'smoker', y = 'tip', data = tips, estimator = median, palette = 'coolwarm');
```



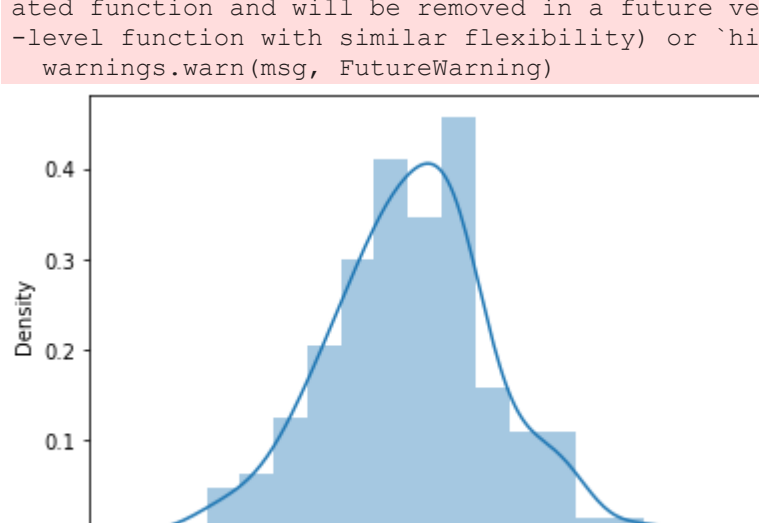
```
In [14]: sns.barplot(x = 'day', y = 'total_bill', data = tips, palette = 'spring', capsize = 0.3);
```



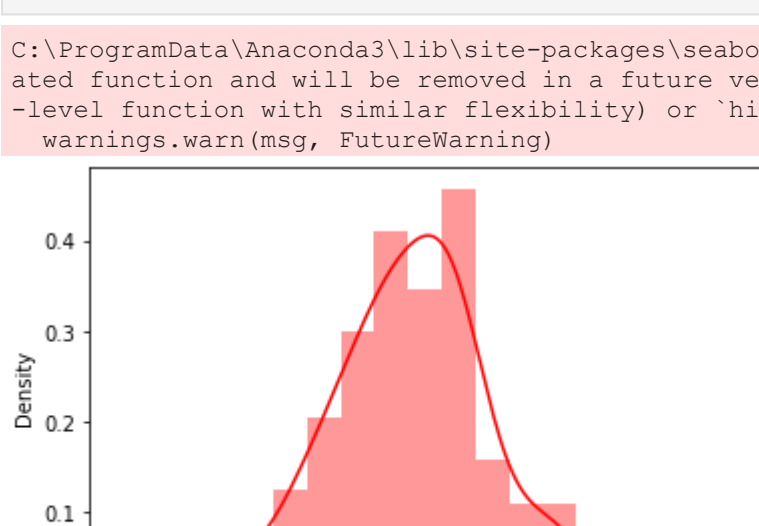
```
In [15]: sns.barplot(x = 'day', y = 'total_bill', data = tips, palette = 'spring', capsize = 0.9);
```



```
In [16]: sns.barplot(x = 'day', y = 'total_bill', data = tips, hue = 'sex', palette = 'husl', capsize = 0.1);
```

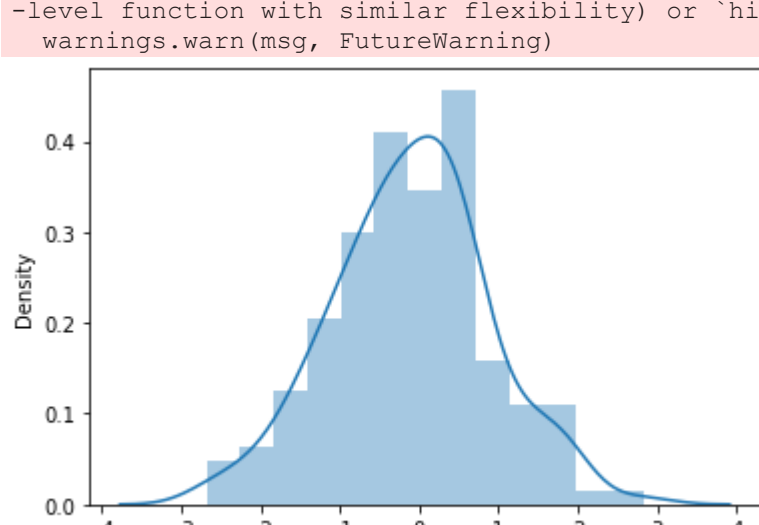


```
In [17]: sns.barplot(x = 'day', y = 'total_bill', data = tips, palette = 'autumn');
```



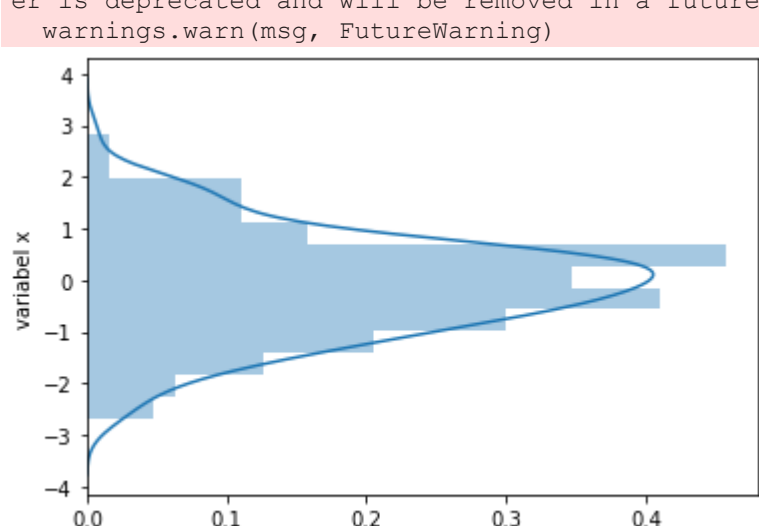
```
In [18]: # Distribution plot
num = np.random.randn(150)
sns.distplot(num);
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for kernel density plots).



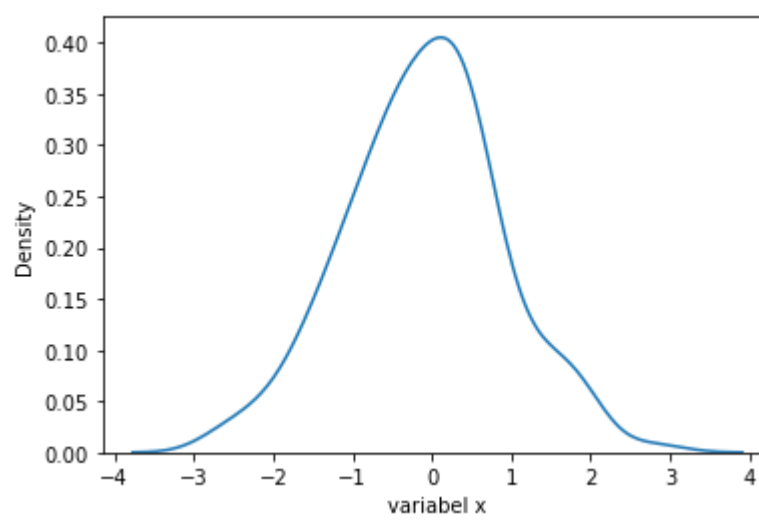
```
In [19]: sns.distplot(num, color = 'red');
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for kernel density plots).



```
In [20]: label_dist = pd.Series(num, name = 'variabel x')
sns.distplot(label_dist);
```

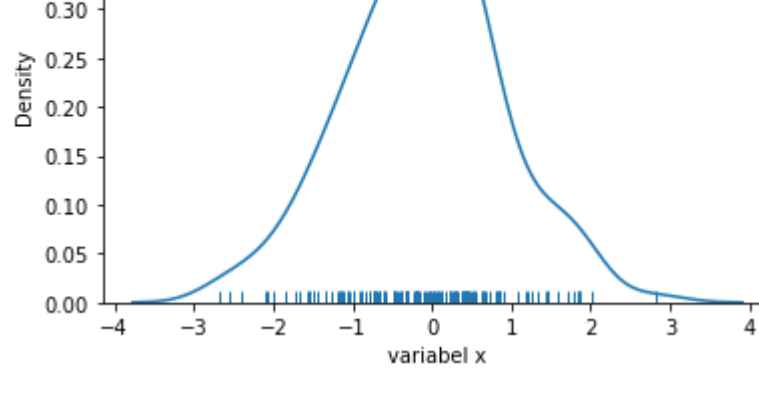
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for kernel density plots).



```
In [21]: sns.distplot(label_dist, vertical = True);
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:1649: FutureWarning: The 'vertical' parameter is deprecated and will be removed in a future version. Assign the data to the 'y' variable instead.



```
In [22]: sns.distplot(label_dist, hist = False)
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'kdeplot' (an axes-level function for kernel density plots).

```
Out[22]: <AxesSubplot:xlabel='variabel x', ylabel='Density'>
```



```
In [23]: sns.distplot(label_dist, rug = True, hist = False)
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: 'distplot' is a deprecated function and will be removed in a future version. Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'kdeplot' (an axes-level function for kernel density plots).

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2055: FutureWarning: The 'axis' variable is no longer used and will be removed. Instead, assign variables directly to 'x' or 'y'.

```
Out[23]: <AxesSubplot:xlabel='variabel x', ylabel='Density'>
```

