Line Graphs

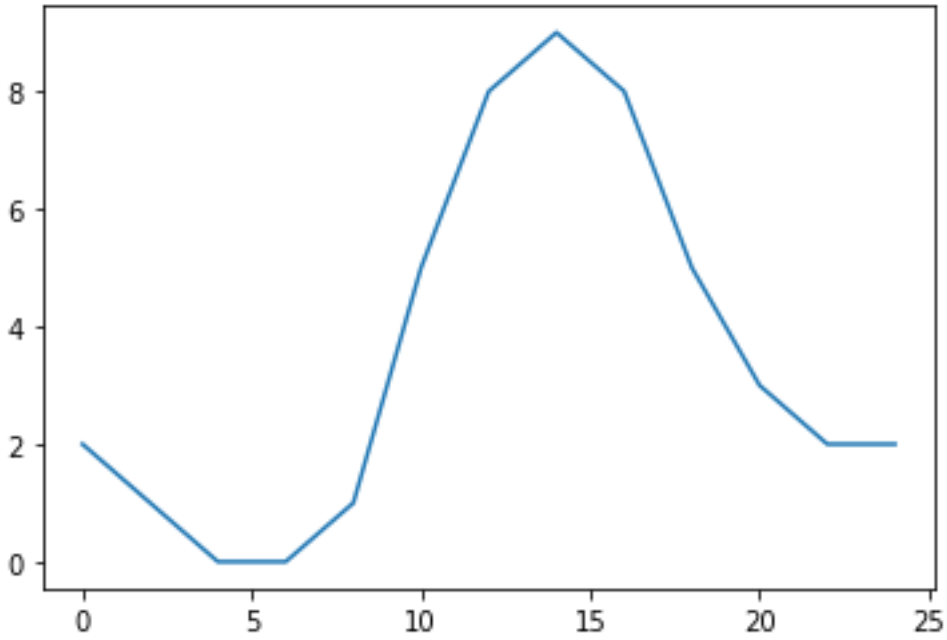
Line Graph adalah bentuk visualisasi lainya selain diagram lingkaran dan diagram batan. Meskipun diagram lingkaran dan diagram batang berguna untuk menunjukkan bagaimana kelas data saling terkait, diagram garis lebih berguna untuk menunjukkan bagaimana kemajuan data selama beberapa periode. Misalnya, grafik garis dapat berguna dalam membuat grafik suhu dari waktu ke waktu, harga saham dari waktu ke waktu, berat menurut hari, atau metrik berkelanjutan lainnya.

Kita akan membuat grafik garis yang sangat sederhana di bawah ini. Data yang kita miliki adalah suhu dalam celsius dan jam dalam sehari untuk satu hari dan lokasi. Anda dapat melihat bahwa untuk membuat grafik garis kita menggunakan metode plt.plot ().

```
import matplotlib.pyplot as plt

temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]

plt.plot(
  hour,
  temperature_c
)
plt.show()
```



Gambar 0.13. Contoh line graph

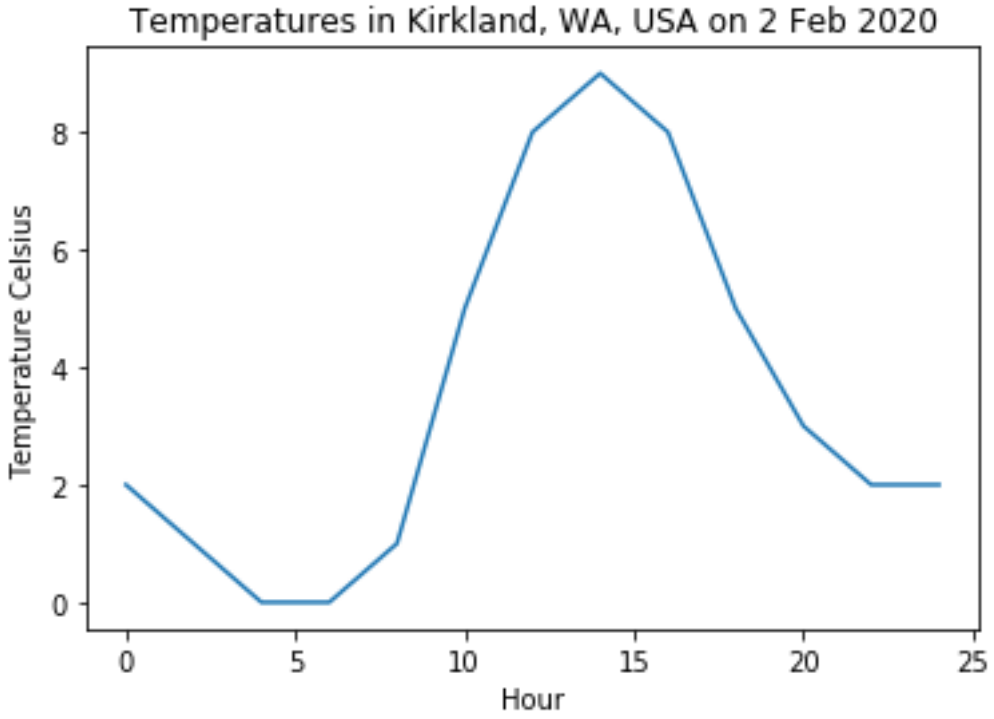
Kita dapat melihat bahwa suhu mulai sekitar 2 derajat celcius pada tengah malam, sedikit turun menjadi beku sekitar pukul 05:00, naik menjadi sekitar 9 derajat celcius pada pukul 15:00, dan kemudian turun kembali menjadi sekitar 2 derajat pada tengah malam (Gambar 0.13).

Kita juga bisa menambahkan elemen bagan standar dari title (), ylabel (), dan xlabel () (Gambar 0.14).

```
import matplotlib.pyplot as plt

temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]

plt.plot(
  hour,
  temperature_c,
)
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020')
plt.ylabel('Temperature Celsius')
plt.xlabel('Hour')
plt.show()
```



Gambar 0.14. Line Graph dengan Title dan Atribut

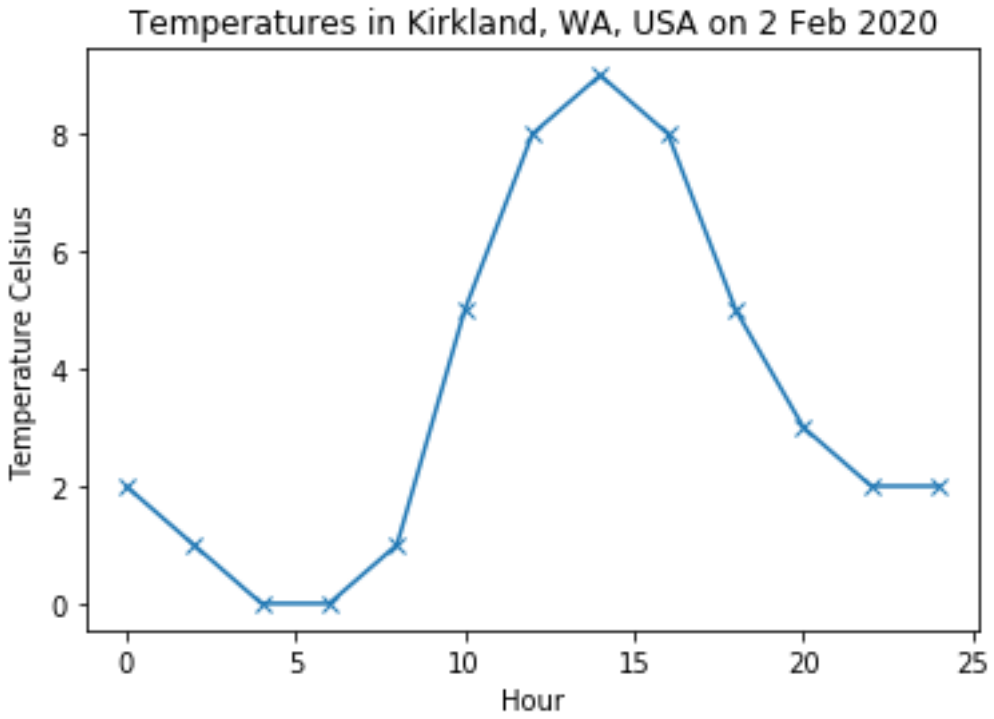
Kita juga dapat menambahkan penanda di setiap titik data (Gambar 0.15). Pada contoh di bawah ini kita menambahkan penanda titik pada setiap titik data menggunakan argumen marker = 'o'.

```
import matplotlib.pyplot as plt

temperature_c = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]

plt.plot(
```

```
  hour,
  temperature_c,
  marker='x',
)
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020')
plt.ylabel('Temperature Celsius')
plt.xlabel('Hour')
plt.show()
```



Gambar 0.15. Line Graph dengan Penanda disetiap Titik

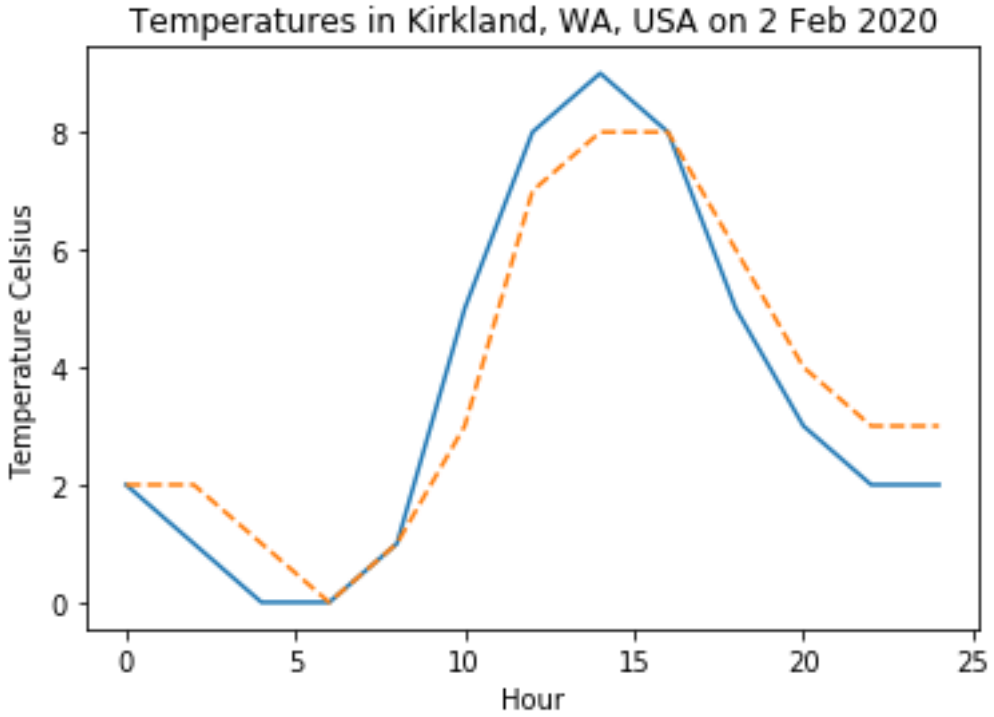
Kita bahkan dapat memiliki beberapa garis pada grafik yang sama (Gambar 0.16). Misalnya, misalnya, kita ingin mengilustrasikan nilai suhu aktual dan prediksi. Kita bisa memanggil plot () dua kali, sekali dengan setiap kumpulan nilai. Perhatikan bahwa dalam panggilan kedua, kita menggunakan argumen lain untuk plot (), linestyle = '-'. Hal ini menyebabkan garis prediksi terlihat seperti garis putus-putus sedangkan nilai sebenarnya tetap solid.
Anda dapat melihat dokumentasi tambahan pada Matplotlib pyplot.plot() documentation.

```
import matplotlib.pyplot as plt

temperature_c_actual = [2, 1, 0, 0, 1, 5, 8, 9, 8, 5, 3, 2, 2]
temperature_c_predicted = [2, 2, 1, 0, 1, 3, 7, 8, 8, 6, 4, 3, 3]
hour = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]

plt.plot(hour, temperature_c_actual)
plt.plot(hour, temperature_c_predicted, linestyle='--')
```

```
plt.title('Temperatures in Kirkland, WA, USA on 2 Feb 2020')
plt.ylabel('Temperature Celsius')
plt.xlabel('Hour')
plt.show()
```

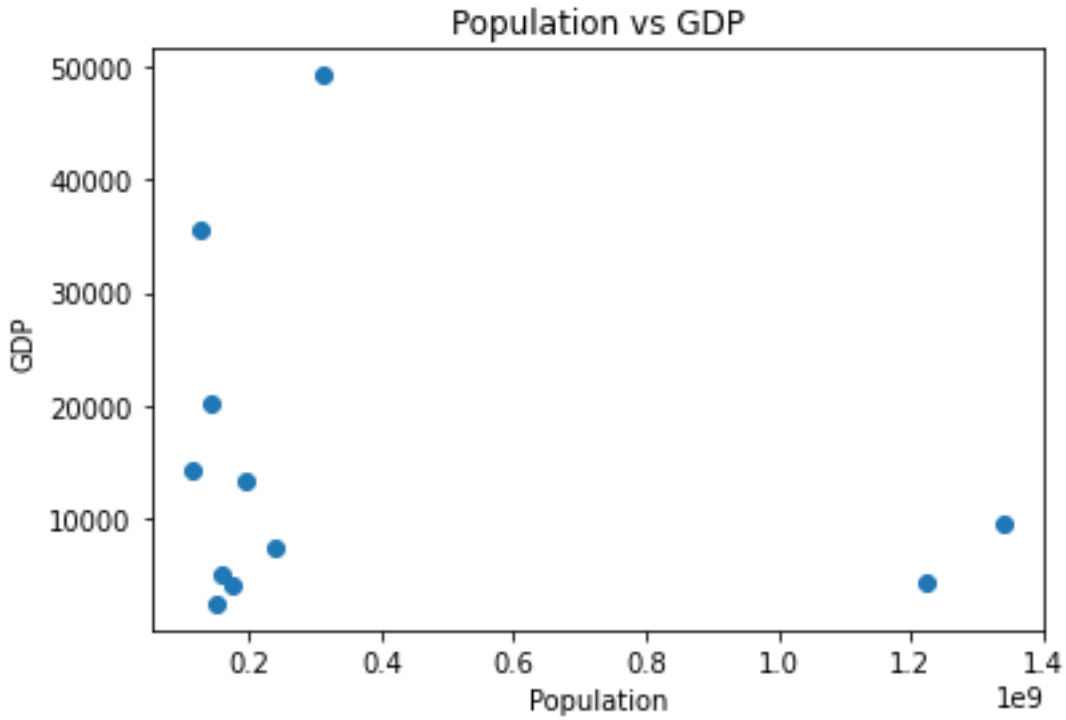


Gambar 0.16. Line Graph dengan Lebih dari satu garis

Scatter Plot

Scatter plot berfungsi baik untuk data dengan dua komponen numerik. Scatter plot dapat memberikan informasi yang berguna terutama mengenai pola atau pencilan. Pada contoh di bawah ini, kita memiliki data yang terkait dengan produk domestik bruto (PDB) dan populasi untuk negara-negara dengan populasi lebih dari seratus juta. PDB adalah total nilai barang dan jasa yang dibuat / disediakan oleh suatu negara selama satu tahun. Kita kemudian menggunakan plt.scatter () untuk membuat sebaran populasi dan PDB (Gambar 0.17).

```
import matplotlib.pyplot as plt
```



Gambar 0.17. Scatter Plot data PDB dan populasi

Scatter plot sangat intuitif karena kita dapat mengumpulkan informasi tentang data kita. Kita dapat melihat bahwa ada dua pencilan populasi (pencilan PDB). Informasi ini dapat membantu kita memutuskan apakah kita perlu mengoreksi atau mengecualikan pencilan dalam analisis kita. Kita juga dapat menambahkan lebih dari satu kumpulan data ke plot (Gambar 0.18).

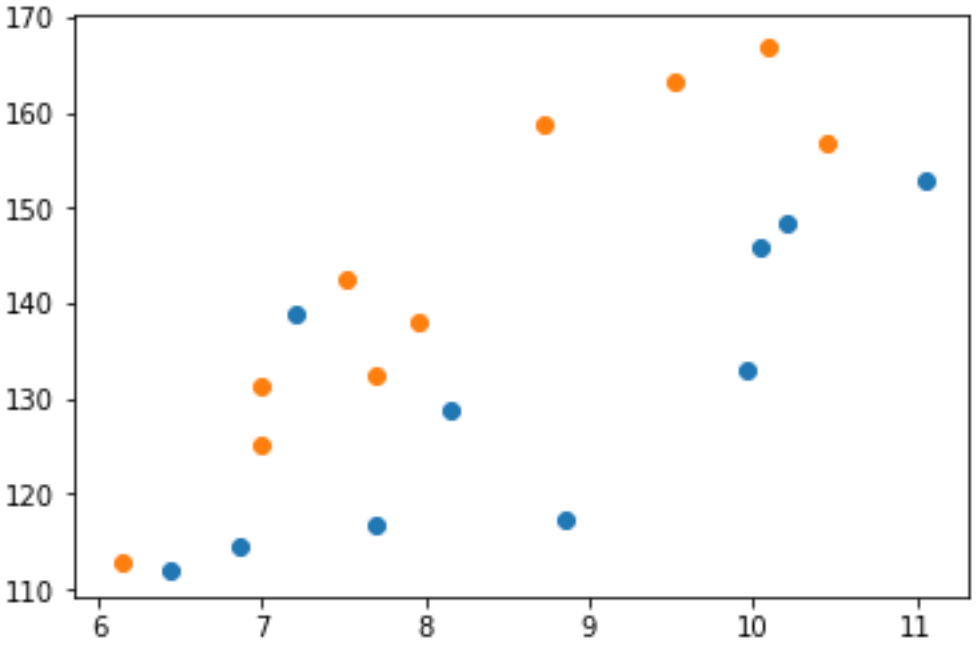
Pada contoh di bawah ini, kita memplot diameter dan berat sekumpulan lemon dan jeruk nipis agar dapat melihat apakah kita dapat menentukan polanya.

```
import matplotlib.pyplot as plt

lemon_diameter = [6.44, 6.87, 7.7, 8.85, 8.15, 9.96, 7.21, 10.04,
10.2, 11.06]
lemon_weight = [112.05, 114.58, 116.71, 117.4, 128.93,
                132.93, 138.92, 145.98, 148.44, 152.81]

lime_diameter = [6.15, 7.0, 7.0, 7.69, 7.95, 7.51, 10.46, 8.72,
9.53, 10.09]
lime_weight = [112.76, 125.16, 131.36, 132.41, 138.08,
               142.55, 156.86, 158.67, 163.28, 166.74]

plt.scatter(lemon_diameter, lemon_weight)
plt.scatter(lime_diameter, lime_weight)
plt.show()
```



Gambar 0.18. Perbandingan lemon dan lime

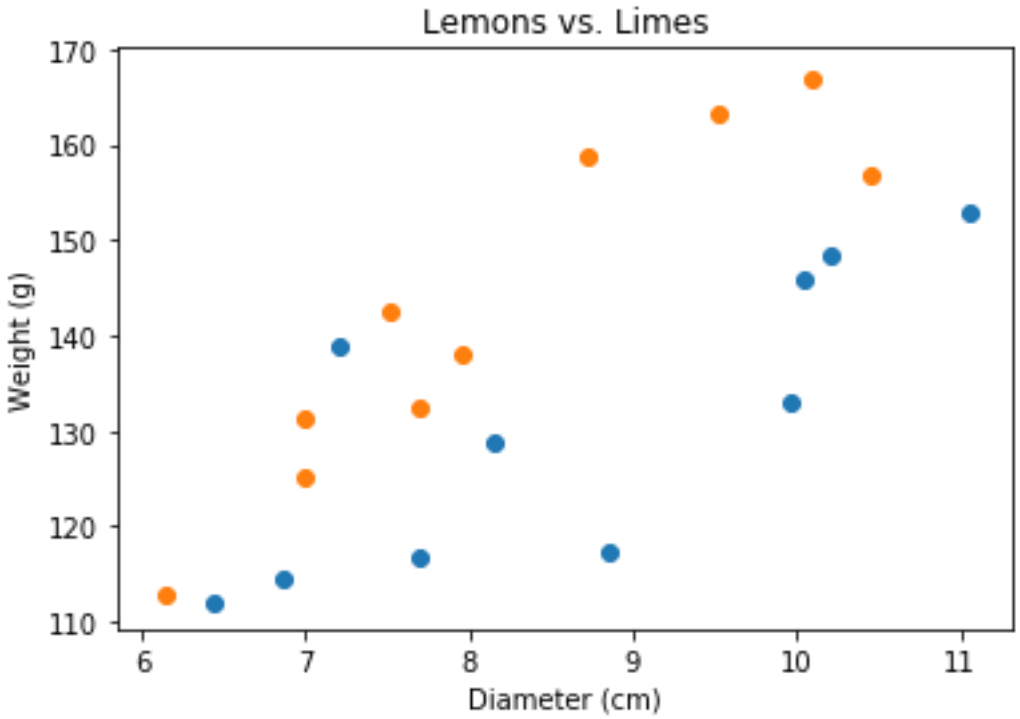
Melihat sampel kita, tidak ada pola yang sangat jelas. Namun, salah satu jenis jeruk tampaknya lebih berat dan diameternya lebih besar. Tapi yang mana? Mari kita bersihkan bagan ini sedikit. Pertama kita akan menambahkan judul menggunakan plt.title (), x-label menggunakan plt.xlabel (), dan y-label menggunakan plt.ylabel () (Gambar 0.19).

```
import matplotlib.pyplot as plt

lemon_diameter = [6.44, 6.87, 7.7, 8.85, 8.15, 9.96, 7.21, 10.04,
10.2, 11.06]
lemon_weight = [112.05, 114.58, 116.71, 117.4, 128.93,
                132.93, 138.92, 145.98, 148.44, 152.81]

lime_diameter = [6.15, 7.0, 7.0, 7.69, 7.95, 7.51, 10.46, 8.72,
9.53, 10.09]
lime_weight = [112.76, 125.16, 131.36, 132.41, 138.08,
               142.55, 156.86, 158.67, 163.28, 166.74]

plt.title('Lemons vs. Limes')
plt.xlabel('Diameter (cm)')
plt.ylabel('Weight (g)')
plt.scatter(lemon_diameter, lemon_weight)
plt.scatter(lime_diameter, lime_weight)
plt.show()
```



Gambar 0.19 Lemon vs lime dengan label

Sekarang kita dapat menambahkan beberapa warna dan legenda untuk membuat scatter plot sedikit lebih intuitif. Kita menambahkan warna dengan meneruskan color = argumen ke plt.scatter (). Dalam hal ini kita hanya mengatur titik lemon menjadi kuning menggunakan color = 'y' dan titik lime menjadi hijau menggunakan color = 'g'.
Untuk menambahkan legenda, kita panggil plt.legend () dan berikan daftar yang berisi label untuk setiap sebaran data (Gambar 0.20).

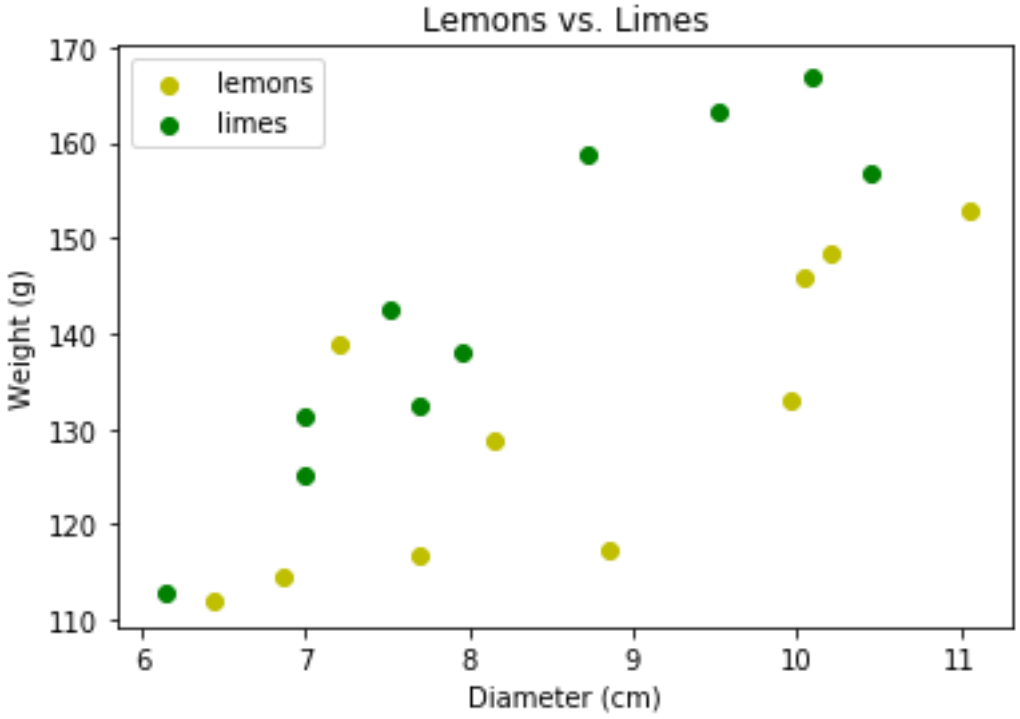
```
import matplotlib.pyplot as plt

lemon_diameter = [6.44, 6.87, 7.7, 8.85, 8.15, 9.96, 7.21, 10.04,
10.2, 11.06]
lemon_weight = [112.05, 114.58, 116.71, 117.4, 128.93,
                132.93, 138.92, 145.98, 148.44, 152.81]

lime_diameter = [6.15, 7.0, 7.0, 7.69, 7.95, 7.51, 10.46, 8.72,
9.53, 10.09]
lime_weight = [112.76, 125.16, 131.36, 132.41, 138.08,
               142.55, 156.86, 158.67, 163.28, 166.74]

plt.title('Lemons vs. Limes')
plt.xlabel('Diameter (cm)')
plt.ylabel('Weight (g)')
plt.scatter(lemon_diameter, lemon_weight, color='y')
plt.scatter(lime_diameter, lime_weight, color='g')
```

```
plt.legend(['lemons', 'limes'])
plt.show()
```



Gambar 0.20. Lemon vs lime perubahan warna

Sekarang kita dapat melihat lebih jelas bahwa jeruk nipis kita cenderung sedikit lebih berat dan diameternya lebih besar sedikit daripada lemon.

Heatmap

Heatmap adalah jenis visualisasi yang menggunakan kode warna untuk mewakili nilai / kepadatan relatif data di seluruh permukaan. Seringkali ini adalah bagan tabel, tetapi tidak harus terbatas pada itu. Untuk data tabular, terdapat label pada sumbu x dan y. Nilai di persimpangan label tersebut dipetakan ke warna. Warna-warna ini kemudian dapat digunakan untuk memeriksa data secara visual guna menemukan kelompok dengan nilai serupa dan mendeteksi tren dalam data.

Kita akan bekerja dengan data tentang temperatur rata-rata setiap bulan untuk 12 kota terbesar di dunia. Untuk membuat heatmap ini, kita akan menggunakan library Seaborn. Seaborn adalah library visualisasi yang dibangun di atas Matplotlib.Library ini menyediakan antarmuka tingkat yang lebih tinggi dan dapat membuat grafik yang lebih menarik. Langkah-langkah untuk melakukan visualisasi heatmap (Gambar 0.21) adalah sebagai berikut

1. Pada kode di bawah ini, pertama kita mengimpor seaborn.
2. Kita kemudian membuat daftar yang berisi nama 12 kota terbesar di dunia dan 12 bulan dalam setahun.
3. Selanjutnya kita menetapkan daftar-daftar ke variabel suhu. Setiap baris dalam daftar mewakili sebuah kota.

4. Setiap kolom adalah satu bulan. Nilai-nilai tersebut adalah suhu tinggi rata-rata untuk kota selama bulan tersebut.
5. Pada tahap akhir kita memanggil sns.heatmap () untuk membuat peta panas. Kita mengirimkan data suhu, nama kota sebagai label-y, dan singkatan bulan sebagai label-x.

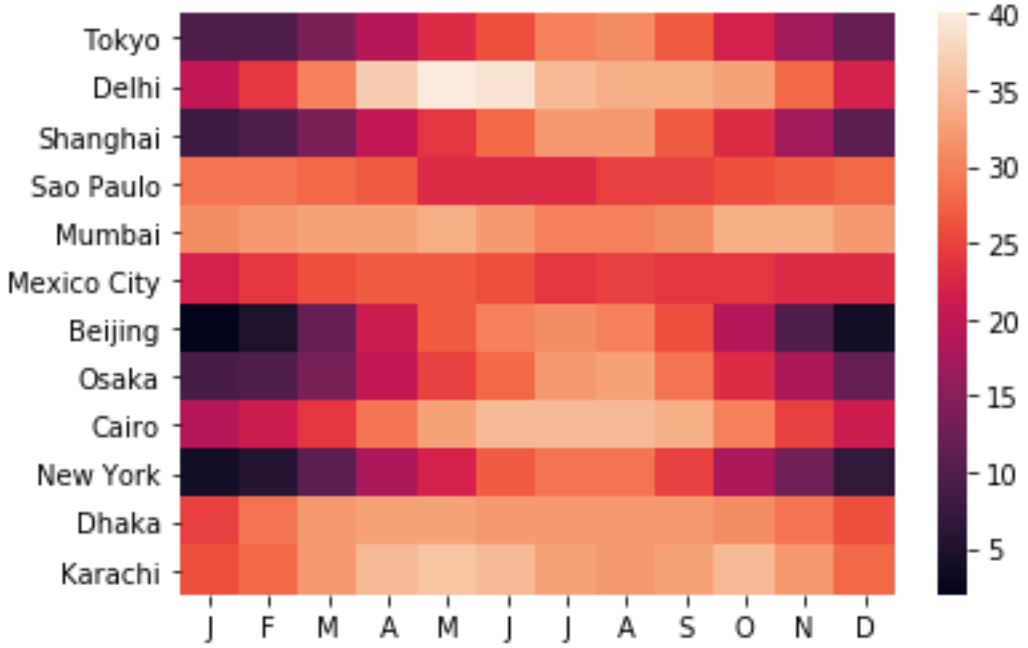
```
import seaborn as sns

cities = ['Tokyo', 'Delhi', 'Shanghai', 'Sao Paulo', 'Mumbai',
'Mexico City',
          'Beijing', 'Osaka', 'Cairo', 'New York', 'Dhaka',
'Karachi']

months = ['J', 'F', 'M', 'A', 'M', 'J', 'J', 'A', 'S', 'O', 'N',
'D']

temperatures = [
  [10, 10, 14, 19, 23, 26, 30, 31, 27, 22, 17, 12], # Tokyo
  [20, 24, 30, 37, 40, 39, 35, 34, 34, 33, 28, 22], # Delhi
  [ 8, 10, 14, 20, 24, 28, 32, 32, 27, 23, 17, 11], # Shanghai
  [29, 29, 28, 27, 23, 23, 23, 25, 25, 26, 27, 28], # Sao Paulo
  [31, 32, 33, 33, 34, 32, 30, 30, 31, 34, 34, 32], # Mumbai
  [22, 24, 26, 27, 27, 26, 24, 25, 24, 24, 23, 23], # Mexico City
  [ 2,  5, 12, 21, 27, 30, 31, 30, 26, 19, 10,  4], # Beijing
  [ 9, 10, 14, 20, 25, 28, 32, 33, 29, 23, 18, 12], # Osaka
  [19, 21, 24, 29, 33, 35, 35, 35, 34, 30, 25, 21], # Cairo
  [ 4,  6, 11, 18, 22, 27, 29, 29, 25, 18, 13,  7], # New York
  [25, 29, 32, 33, 33, 32, 32, 32, 32, 31, 29, 26], # Dhaka
  [26, 28, 32, 35, 36, 35, 33, 32, 33, 35, 32, 28], # Karachi
]

sns.heatmap(temperatures, yticklabels=cities, xticklabels=months)
<matplotlib.axes._subplots.AxesSubplot at 0x22343c81988>
```



Gambar 0.21. Heatmtp mengenai temperatur di masing-masing kota selama 12 bulan

Kita bisa melihat data di grafik yang dihasilkan. Tapi bagaimana kita menafsirkannya? Sebenarnya cukup sulit untuk memahami data. Bagian kiri dan kanan grafik mungkin berisi warna yang lebih gelap, yang memetakan ke suhu yang lebih dingin, tetapi itu pun sulit untuk ditentukan. Kita perlu mengurutkan secara manual kota-kota tersebut, dari yang terkecil ke yang terbesar. Mari kita coba ubah pengurutan berdasarkan lintang pada garis bumi (Gambar 0.22).

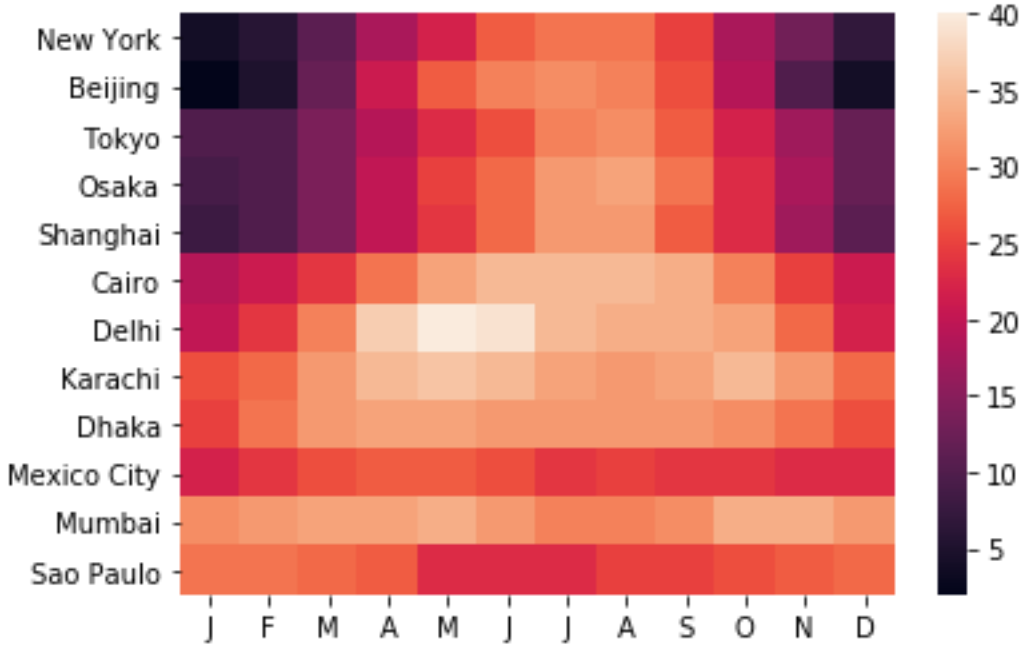
```
import seaborn as sns

cities = ['New York', 'Beijing', 'Tokyo', 'Osaka', 'Shanghai',
'Cairo', 'Delhi',
          'Karachi', 'Dhaka', 'Mexico City', 'Mumbai', 'Sao Paulo']

temperatures = [
  [ 4,  6, 11, 18, 22, 27, 29, 29, 25, 18, 13,  7], # New York
  [ 2,  5, 12, 21, 27, 30, 31, 30, 26, 19, 10,  4], # Beijing
  [10, 10, 14, 19, 23, 26, 30, 31, 27, 22, 17, 12], # Tokyo
  [ 9, 10, 14, 20, 25, 28, 32, 33, 29, 23, 18, 12], # Osaka
  [ 8, 10, 14, 20, 24, 28, 32, 32, 27, 23, 17, 11], # Shanghai
  [19, 21, 24, 29, 33, 35, 35, 35, 34, 30, 25, 21], # Cairo
  [20, 24, 30, 37, 40, 39, 35, 34, 34, 33, 28, 22], # Delhi
  [26, 28, 32, 35, 36, 35, 33, 32, 33, 35, 32, 28], # Karachi
  [25, 29, 32, 33, 33, 32, 32, 32, 32, 31, 29, 26], # Dhaka
  [22, 24, 26, 27, 27, 26, 24, 25, 24, 24, 23, 23], # Mexico City
  [31, 32, 33, 33, 34, 32, 30, 30, 31, 34, 34, 32], # Mumbai
```

```
  [29, 29, 28, 27, 23, 23, 23, 25, 25, 26, 27, 28], # Sao Paulo
]

sns.heatmap(temperatures, yticklabels=cities, xticklabels=months)
<matplotlib.axes._subplots.AxesSubplot at 0x22345cc0a48>
```



Gambar 0.22. Heatmap temperature negara berdasarkan garis bumi (lintang)

Kita dapat melihat bahwa kota-kota di garis lintang yang lebih tinggi, lebih dingin dari bulan September hingga Maret dan suhu cenderung meningkat seiring dengan semakin mengecilnya garis lintang.
Perhatikan juga bahwa Sao Paulo terlihat lebih hangat di tengah tahun meskipun berada di belahan bumi selatan. Memang, skema warnanya sulit dibaca. Anda dapat mengubah skema warna menggunakan argumen cmap =. cmap = menerima daftar warna dan skema warna preset (Gambar 0.23). Anda dapat menemukan skema di Matplotlib colormap documentation.

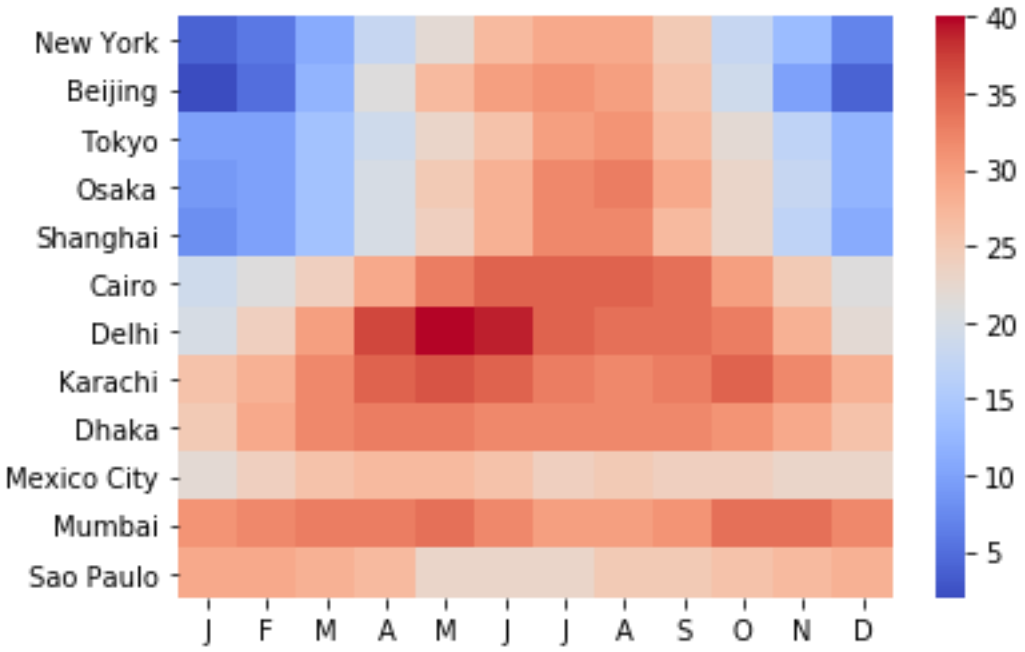
```
import seaborn as sns

cities = ['New York', 'Beijing', 'Tokyo', 'Osaka', 'Shanghai',
'Cairo', 'Delhi',
          'Karachi', 'Dhaka', 'Mexico City', 'Mumbai', 'Sao Paulo']

temperatures = [
  [ 4,  6, 11, 18, 22, 27, 29, 29, 25, 18, 13,  7], # New York
  [ 2,  5, 12, 21, 27, 30, 31, 30, 26, 19, 10,  4], # Beijing
  [10, 10, 14, 19, 23, 26, 30, 31, 27, 22, 17, 12], # Tokyo
  [ 9, 10, 14, 20, 25, 28, 32, 33, 29, 23, 18, 12], # Osaka
```

```
  [ 8, 10, 14, 20, 24, 28, 32, 32, 27, 23, 17, 11], # Shanghai
  [19, 21, 24, 29, 33, 35, 35, 35, 34, 30, 25, 21], # Cairo
  [20, 24, 30, 37, 40, 39, 35, 34, 34, 33, 28, 22], # Delhi
  [26, 28, 32, 35, 36, 35, 33, 32, 33, 35, 32, 28], # Karachi
  [25, 29, 32, 33, 33, 32, 32, 32, 32, 31, 29, 26], # Dhaka
  [22, 24, 26, 27, 27, 26, 24, 25, 24, 24, 23, 23], # Mexico City
  [31, 32, 33, 33, 34, 32, 30, 30, 31, 34, 34, 32], # Mumbai
  [29, 29, 28, 27, 23, 23, 23, 25, 25, 26, 27, 28], # Sao Paulo
]

sns.heatmap(
    temperatures,
    yticklabels=cities,
    xticklabels=months,
    cmap='coolwarm',
)
<matplotlib.axes._subplots.AxesSubplot at 0x22345d9a3c8>
```



Gambar 0.23. Perubahan warna heatmap colormap