Materi Visualisasi

Visualisasi Variabel

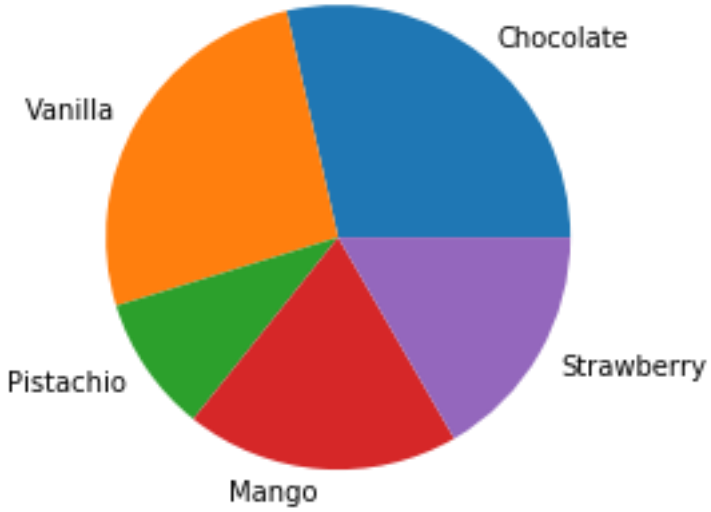
Visualisasi berperan peran penting dalam bidang machine learning dan data science. Seringkali kita perlu menyaring informasi kunci yang ditemukan dalam sejumlah data untuk menjadi bentuk yang bermakna dan mudah dicerna. Visualisasi yang baik dapat menceritakan sebuah cerita tentang data dengan cara yang tidak dapat dilakukan oleh sebuah klaimat.

Di modul ini kita akan mengeksplorasi beberapa teknik visualisasi yang umum. Lab ini akan menggunakan toolkit seperti Matplotlib's Pyplot dan Seaborn untuk membuat gambar informatif yang memberikan informasi dan pengetahuan mengenai dataset.

Pie Charts

Pie chart digunakan untuk menunjukkan seberapa banyak dari setiap jenis kategori dalam dataset berbanding dengan keseluruhan. Pada bagian ini kita akan membuat diagram lingkaran menggunakan kumpulan data sampel. Variabel label berisi tupel rasa es krim. Variabel voting berisi tupel voting. Data tersebut mewakili jumlah voting rase es krim favorit. Kita dapat membuat grafik menggunakan library Pyplot Matplotlib. Method plt.pie () digunakan untuk membuat interface pie chart berdasarkan data rasa es krim dan jumlah voting .

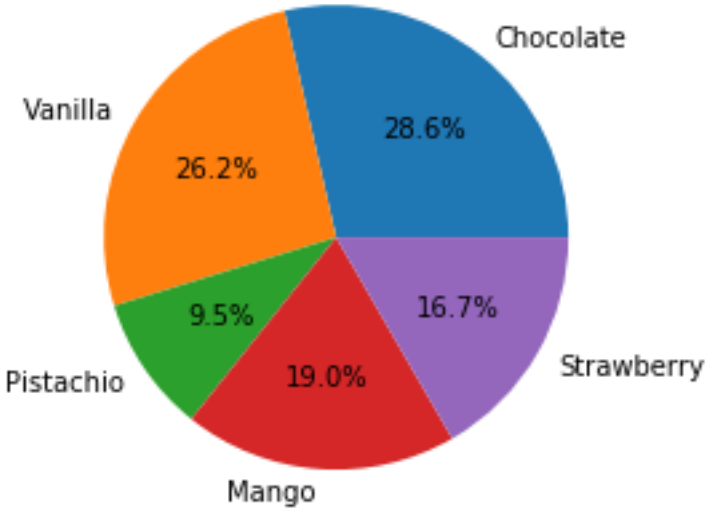
```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)

plt.pie(
    votes,
    labels=flavors,
)
plt.show()
```



Gambar 0.1. Pie Chart perbandingan rasa es krim

Berdasarkan Gambar 0.1, kita dapat dengan mudah melihat bahwa cokelat adalah rasa yang paling populer, diikuti dengan rasa vanila. Kita dapat mengetahui hal ini dengan melihat data mentahnya. Namun, data dalam format diagram lingkaran juga dapat digunakan untuk melihat informasi lain dengan mudah, seperti fakta bahwa kombinasi cokelat dan vanila mewakili lebih dari setengah suara. Apa yang tidak kita lihat adalah persentase sebenarnya. Jika kita ingin melihat berapa persen kontribusi masing-masing rasa es kirim, kita bisa menggunakan argumen autopct. Untuk nilai argumen, ada beberapa string format yang dapat digunakan untuk mengatur ketepatan tampilan data. Coba ubah nilainya menjadi% 1.0 %% dan% 1.2f %%. Apa yang terjadi?

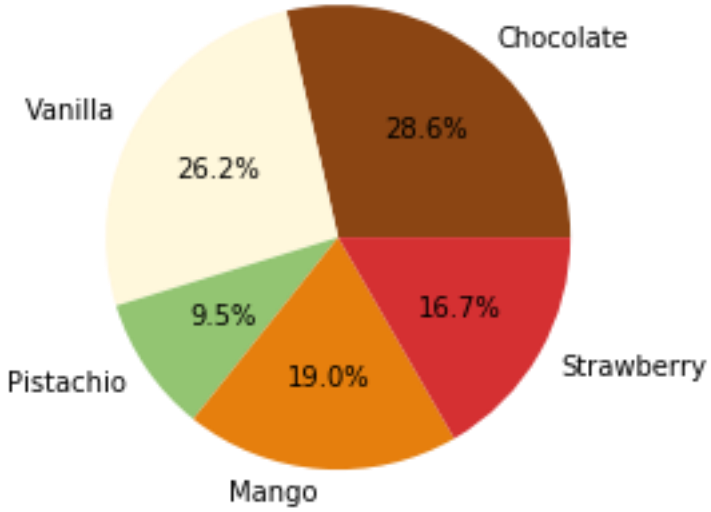
```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango', 'Strawberry')
votes = (12, 11, 4, 8, 7)

plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
)
plt.show()
```



Gambar 0.2. Penambahan argument persentase

Sekarang kita dapat melihat persentase kontribusi setiap rasa es krim secara keseluruhan (Gambar 0.2). Satu hal yang masih sedikit membingungkan tentang bagan ini adalah pilihan warnanya. Kita dapat merubah warna dari setiap rasa es krim Pie Chart. Matplotlib memungkinkan Anda mengubah warna yang ditampilkan pada bagan dengan memasukkan nilai warna. Anda dapat menggunakan shorcut yang telah diprogram seperti 'b' untuk biru dan 'g' untuk hijau.
Dalam kasus ini, kita menggunakan warna html. Warna ini adalah adalah nilai enam karakter di mana dua karakter pertama mewakili jumlah warna merah, dua karakter berikutnya adalah jumlah warna hijau, dan dua karakter terakhir mewakili jumlah warna biru. Anda dapat menemukan custom warna yang lebih banyak dengan mencari kata kunci 'kode warna html'.
Pada code dibawah ini assignment warna dilakukan untuk setiap rasa (Gambar 0.3).

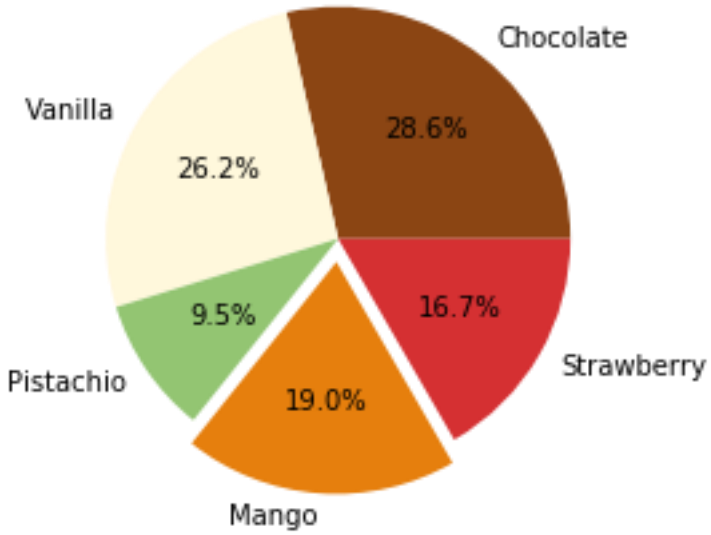
```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango',
'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')

plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
    colors=colors,
)
plt.show()
```



Gambar 0.3. Penambahan warna pada masing-masing kelas

Sekarang mari kita bayangkan kita sedang mempersiapkan bagan ini untuk presentasi, dan kita ingin memanggil salah satu rasa secara khusus. Mungkin mangga baru dipasarkan, dan kita ingin melakukan highlight terhadap data mangga.
Untuk melakukan ini kita bisa menggunakan argumen explode. Ini memungkinkan kita menyetel offset untuk setiap irisan pai dari tengah. Pada contoh di bawah ini kita mendorong mangga keluar sebesar 0,1 sambil menjaga semua potongan lainnya tetap berada ditengah.

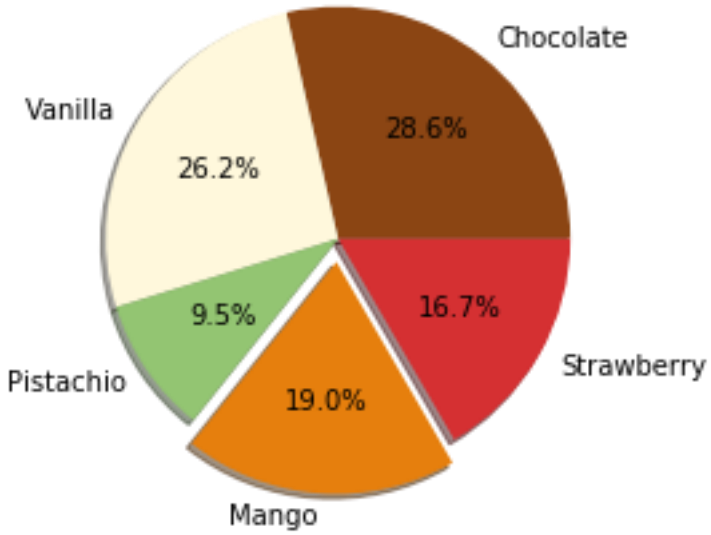
```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango',
'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0)

plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
    colors=colors,
    explode=explode,
)
plt.show()
```



Gambar 0.4. Highlight item mango

Sekarang mangga sudah ditarik keluar sedikit dari Pie Chart, sehingga kita dapat melihat highlight dari data mangga (Gambar 0.4). Diagram lingkaran sudah terlihat cukup bagus, tetapi tampilanya sangat standard. Kita bisa memberinya sedikit tampilan tiga dimensi dengan menambahkan bayangan dengan argumen bayangan (Gambar 0.5).

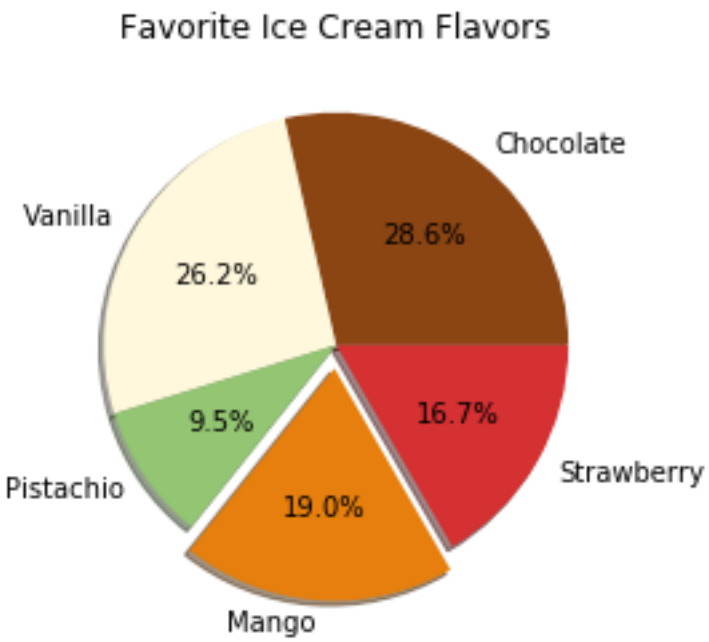
```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango',
'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0)

plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
    colors=colors,
    explode=explode,
    shadow=True
    )
plt.show()
```



Gambar 0.5. Penambahan bayangan pada visualisasi

Untuk menyelesaikannya, kita bisa menambahkan judul menggunakan plt.title (). Perhatikan bahwa ini bukan argumen untuk plt.pie (), melainkan pemanggilan metode terpisah di plt.

```
import matplotlib.pyplot as plt

flavors = ('Chocolate', 'Vanilla', 'Pistachio', 'Mango',
'Strawberry')
votes = (12, 11, 4, 8, 7)
colors = ('#8B4513', '#FFF8DC', '#93C572', '#E67F0D', '#D53032')
explode = (0, 0, 0, 0.1, 0)

plt.title('Favorite Ice Cream Flavors')
plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
    colors=colors,
    explode=explode,
    shadow=True
    )
plt.show()
```



Gambar 0.6. Pie Chart Lengkap

Sekarang kita dapat memiliki Pie Chart (Gambar 0.6) yang menunjukkan semua rasa es krim favorit dalam sebuah survey! Ingat diagram lingkaran bagus untuk menunjukkan bagaimana distribusi kelas pada data yang berbeda (dalam hal ini, rasa es krim) . Pie chart akan sangat efektif jika hanya ada beberapa kelas yang terwakili. Bayangkan jika kita memiliki 100 rasa es krim. Maka tampilan Pie Chart akan sangat penuh

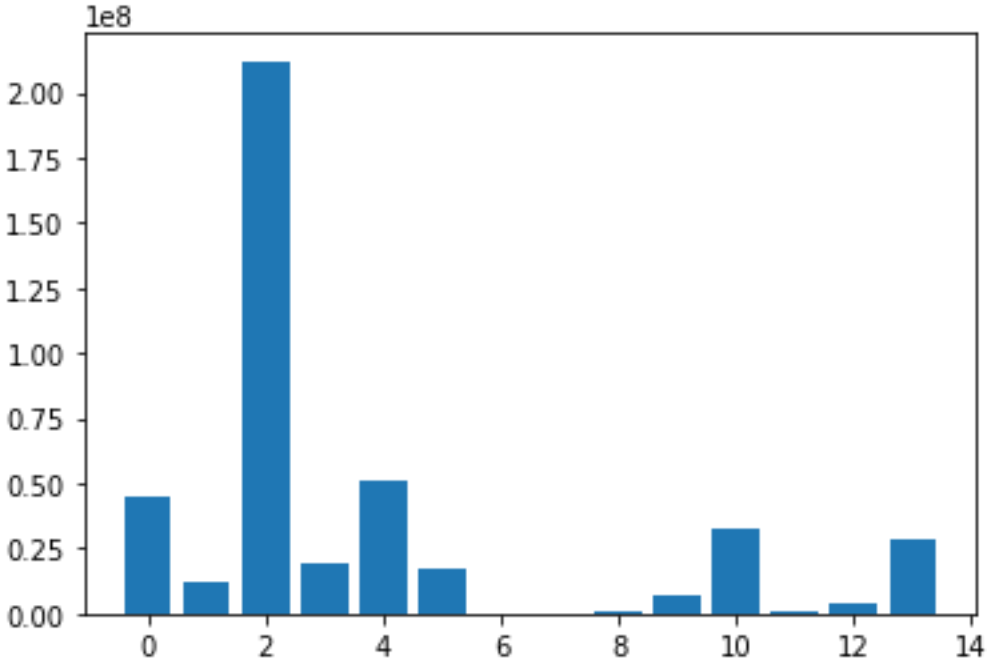
Bar Charts

Bar Chart adalah merupakan tools visualisassi yang dapat digunakan untuk membandingkan data kategorikal. Mirip dengan diagram lingkaran, diagram ini dapat digunakan untuk membandingkan kategori data satu sama lain. Namun, diagram lingkaran sangat spesifik untuk melihat bagaimana satu kategori data dibandingkan dengan keseluruhan. Grafik diagram batang tidak terlalu tepat untuk hal tersebut. Selain itu, diagram batang dapat menampilkan lebih banyak kategori data daripada diagram lingkaran.

Mari kita mulai dengan melihat diagram batang yang menunjukkan populasi setiap negara di Amerika Selatan. Untuk melakukan ini kita akan menggunakan Matplotlib. Kali ini kita akan menggunakan method bar(). bar () memiliki dua argumen yang diperlukan. Argumen pertama berisi koordinat x dari data. Karena kita ingin memplot nama negara pada sumbu x. Dalam kasus ini kita dapat menggunakan fungsi arange () NumPy untuk membuat daftar angka yang memiliki jumlah array sama. Assignment angka antara 0 dan panjang data, yang seharusnya memberi daftar bilangan bulat mulai dari 0 dan berakhir pada len (data) -1, yaitu 13 dalam contoh kasus ini. Argumen berikutnya adalah data numerik untuk dipetakan. Dalam contoh ini kita memplot data populasi..

```
import matplotlib.pyplot as plt
import numpy as np
```

```
countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)

x_coords = np.arange(len(countries))
plt.bar(x_coords, populations)
plt.show()
```

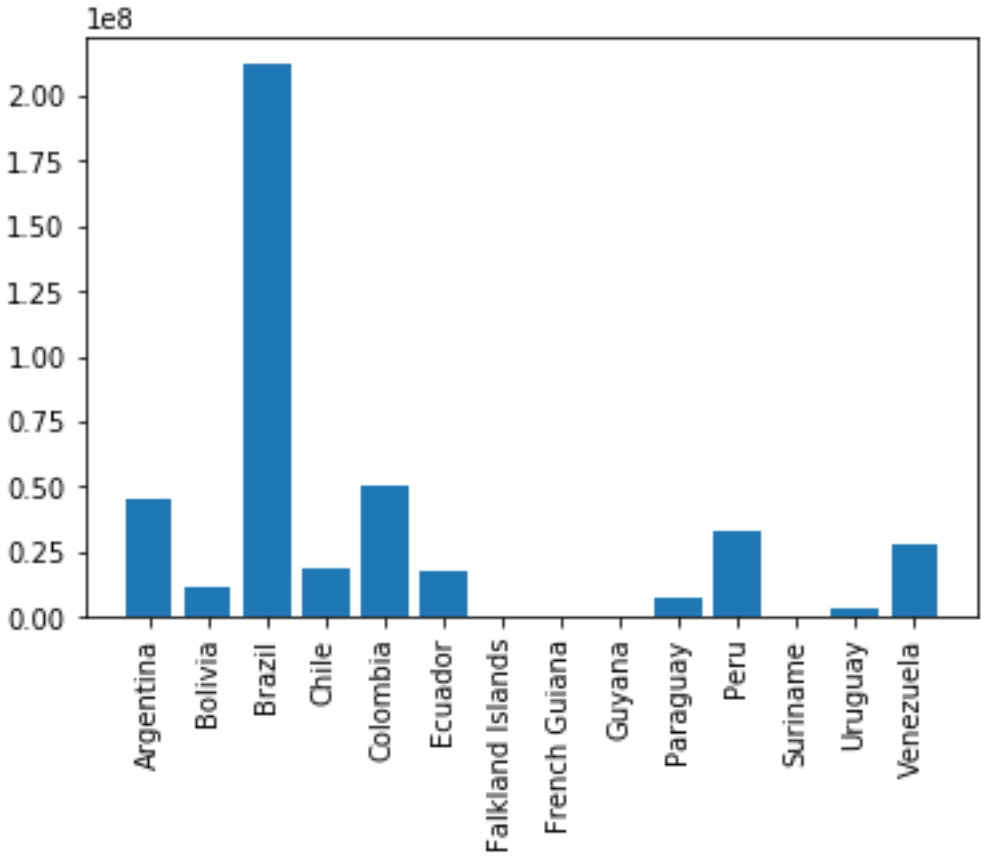


Gambar 0.7. Bar Charts Simple Visualisations

Anda dapat melihat pada Gambar 0.7 bahwa x-label tidak bermakna. Kita bisa memperbaiki ini dengan meneruskan argumen tick_label ke bar (). Karena kita memiliki label yang relatif lebar, akan berguna juga untuk memutar label sejauh 90 derajat agar lebih mudah dibaca. Lakukan panggilan metode plt.xticks (rotation = 90).

```
import matplotlib.pyplot as plt
import numpy as np

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
```

```
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)

x_coords = np.arange(len(countries))
plt.bar(x_coords, populations, tick_label=countries)
plt.xticks(rotation=90) #rotates text for x-axis labels
plt.show()
```

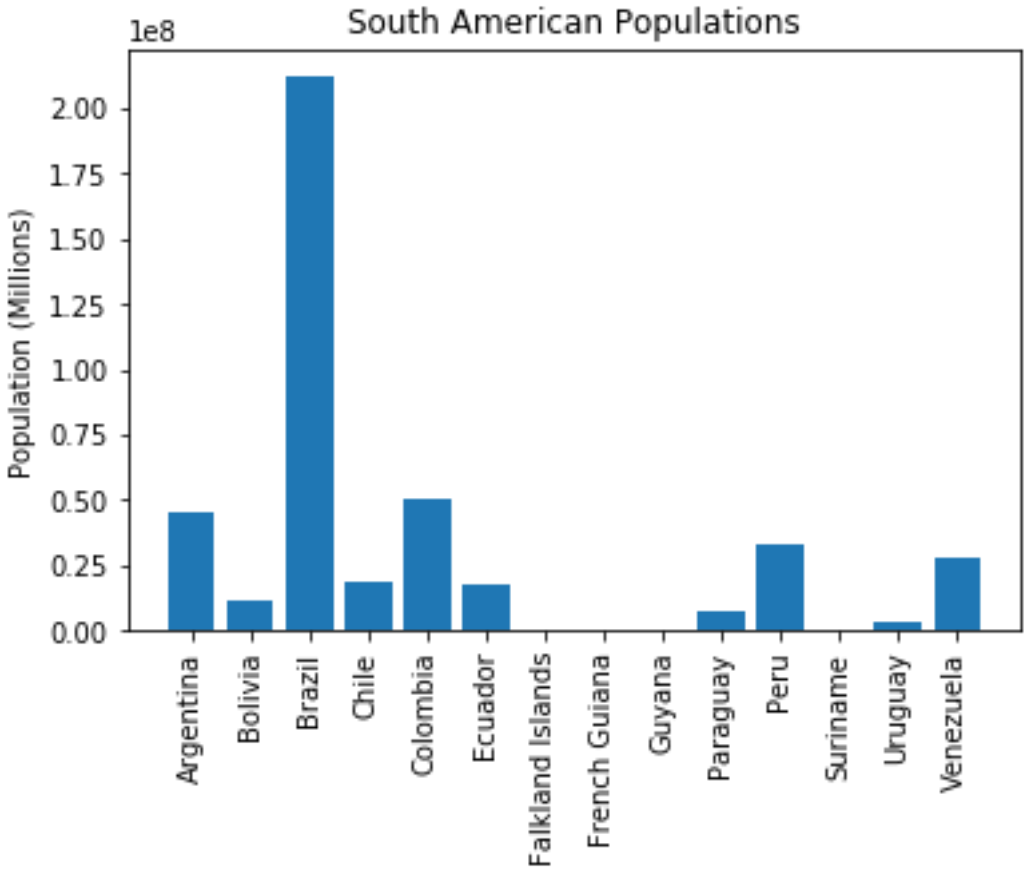


Gambar 0.8. Penambahan label pada bar chart

Kita dapat menambahkan label ke diagram batang (Gambar 0.8) untuk membantu membuat diagram agar lebih mudah dibaca. Pada contoh di bawah ini kita menambahkan label-y menggunakan metode ylabel () dan judul grafik menggunakan metode title () (Gambar 0.9).

```
import matplotlib.pyplot as plt
import numpy as np

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
```

```
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)

x_coords = np.arange(len(countries))
plt.bar(x_coords, populations, tick_label=countries)
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
plt.title('South American Populations')
plt.show()
```

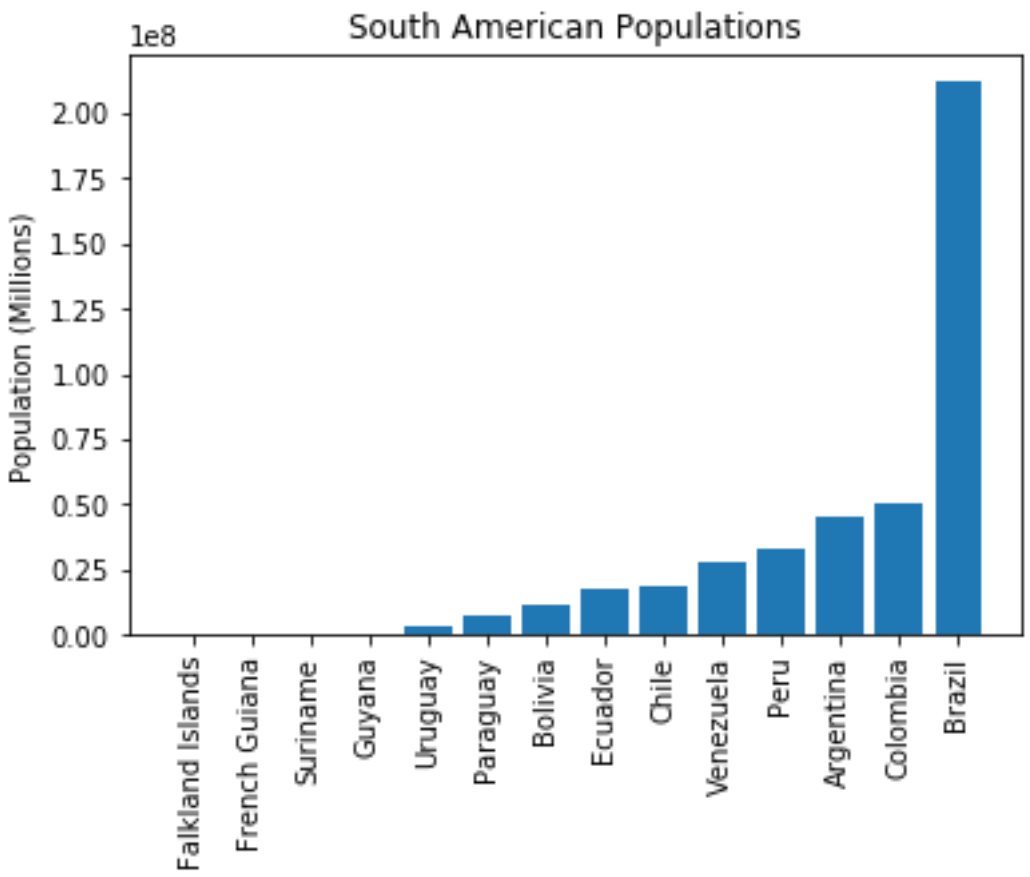


Gambar 0.9. Penambahan label dan title pada bar chart

Bagannya sudah terlihat cukup bagus. Tetapi bagaimana jika pertanyaan: Apa negara terpadat kedua di Amerika Selatan? Anda mungkin harus sedikit menatap Argentina dan Kolombia. Ini karena data diurutkan menurut abjad, yang bukan merupakan pengurutan

yang paling berguna untuk menjawab pertanyaan tentang data. Sayangnya Matplotlib tidak memiliki penyortiran bawaan. Sebagai gantinya, Anda dapat mengimpor Panda dan menggunakannya untuk mengurutkan data (Gambar 0.10).

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)

df = pd.DataFrame({
    'Country': countries,
    'Population': populations,
})
df.sort_values(by='Population', inplace=True)

x_coords = np.arange(len(df))
plt.bar(x_coords, df['Population'], tick_label=df['Country'])
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
plt.title('South American Populations')
plt.show()
```



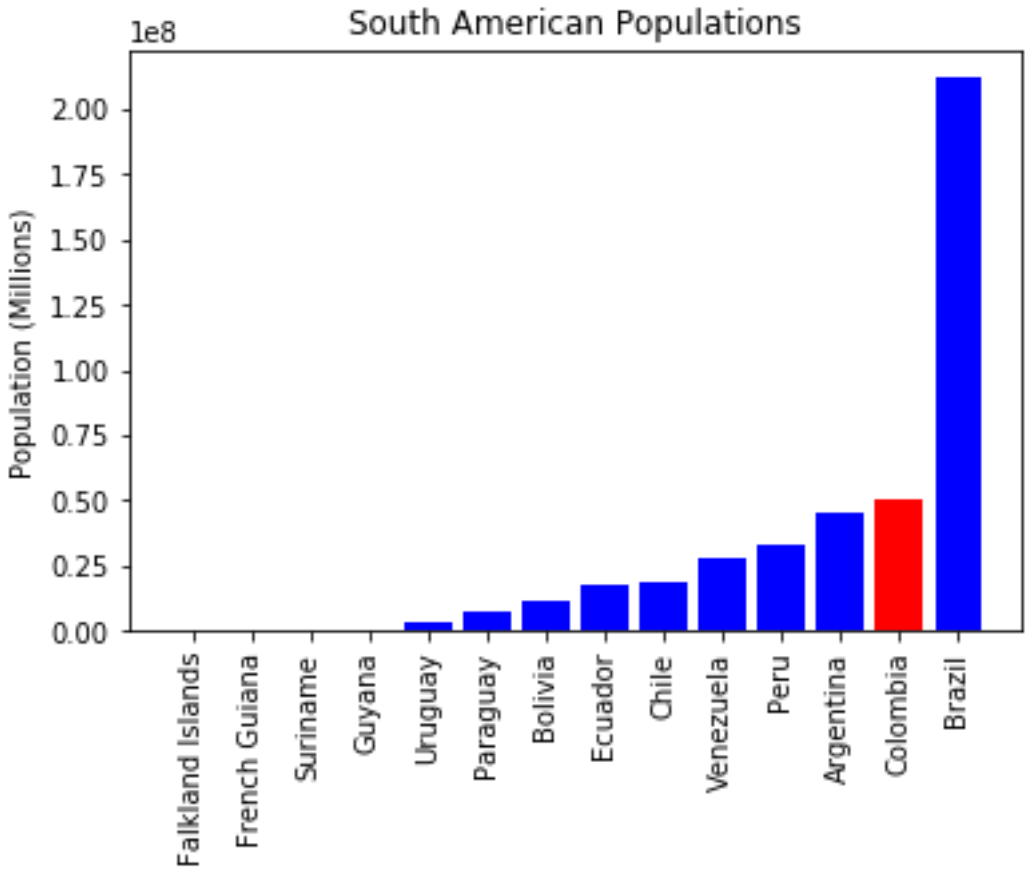
Gambar 0.10. Pengurutan Jumlah Populasi pada Bar Chart

```
len(df)
14
```

Sekarang kita dapat dengan mudah melihat bahwa Kolombia adalah negara terbesar kedua (Gambar 0.11). Jika kita ingin memanggilnya, kita bisa meneruskan daftar warna bar ke metode bar ().

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)
```

```
df = pd.DataFrame({
    'Country': countries,
    'Population': populations,
})
df.sort_values(by='Population', inplace=True)

x_coords = np.arange(len(df))
colors = ['#0000FF' for _ in range(len(df))]
colors[-2] = '#FF0000'
plt.bar(x_coords, df['Population'], tick_label=df['Country'],
color=colors)
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
plt.title('South American Populations')
plt.show()
```



Gambar 0.11. Highlight Populasi Kolombia

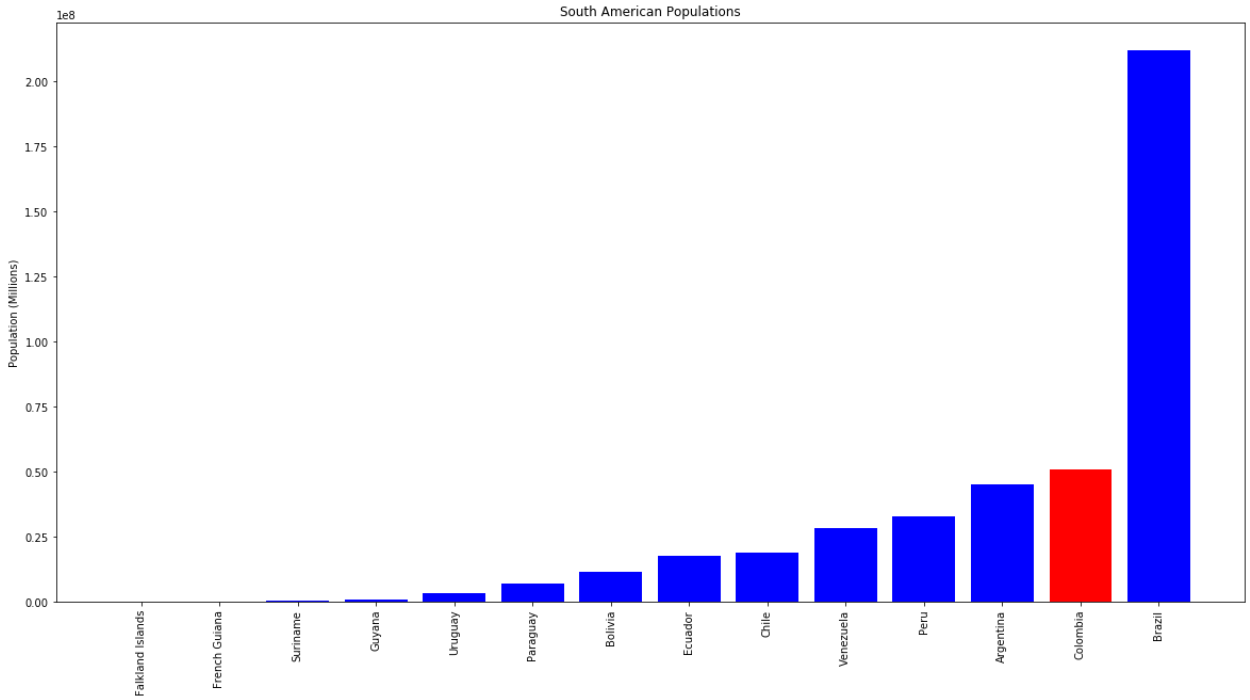
```
colors
['#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
```

```
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#0000FF',
 '#FF0000',
 '#0000FF']
```

Kita juga bisa membuat grafik menjadi lebih besar menggunakan metode figure (). Kita dapat meneruskan argumen figsize = yang mewakili lebar dan tinggi gambar dalam inci (Gambar 0.12).

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

countries = ('Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia',
'Ecuador',
             'Falkland Islands', 'French Guiana', 'Guyana',
'Paraguay', 'Peru',
             'Suriname', 'Uruguay', 'Venezuela')

populations = (45076704, 11626410, 212162757, 19109629, 50819826,
17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169,
3470475,
               28258770)

df = pd.DataFrame({
    'Country': countries,
    'Population': populations,
})
df.sort_values(by='Population', inplace=True)

x_coords = np.arange(len(df))
colors = ['#0000FF' for _ in range(len(df))]
colors[-2] = '#FF0000'
plt.figure(figsize=(20,10))plt.bar(x_coords, df['Population'],
tick_label=df['Country'], color=colors)
plt.xticks(rotation=90)
plt.ylabel('Population (Millions)')
```

```
plt.title('South American Populations')
plt.show()
```



Gambar 0.12. Contoh Hasil Bar Chart Lengkap