



Learning Progress Review 7

8 - 14 November 2025



Outlines



Advanced Statistics

Sampling, Hypothesis Testing, A/B Testing



Data Visualization

Library, Basic Viz, Matplotlib vs Seaborn, Hands On



Google Data Studio

Data Storytelling, BI, Google Locker Studio



Advanced Statistics

Sampling, Hypothesis Testing, A/B Testing

Sampling



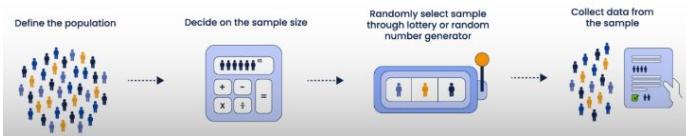
Pengertian Sampling

Proses memilih sebagian elemen dari suatu populasi untuk mewakili populasi tersebut (subset populasi)



Simple Random Sampling

Setiap anggota populasi memiliki kesempatan yang sama untuk dipilih menjadi sampel



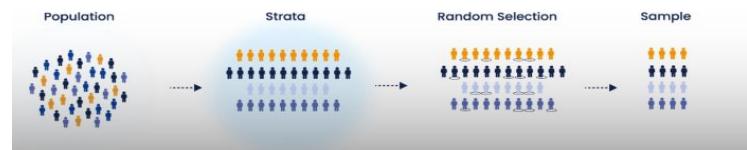
Snowball Sampling

Peserta menggunakan jaringan sosial untuk merekrut sampel tambahan (sering digunakan dalam penelitian sosial untuk populasi yang sulit diakses)



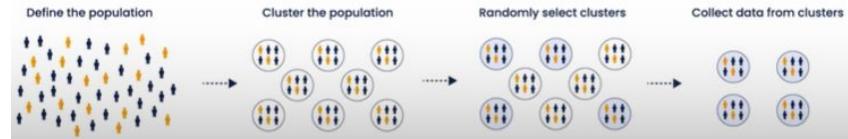
Stratified Sampling

Populasi dibagi menjadi strata berdasarkan karakteristik tertentu dan sampel diambil acak dari tiap strata



Cluster Sampling

Populasi dibagi menjadi kelompok tertentu dan sampel diambil secara acak dari tiap kelompok



Pros & Cons Teknik Sampling

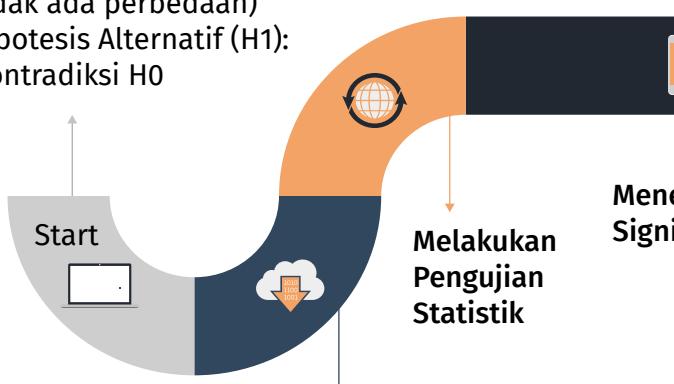
Teknik Sampling	Pros	Cons	Kapan Sebaiknya digunakan?
Simple Random Sampling	- Mudah dipahami dan diterapkan - Mengurangi bias, setiap anggota memiliki peluang yang sama untuk dipilih	Tidak cukup mewakili sub kelompok kecil	Populasi homogen dan tidak tersedia informasi sebelumnya
Stratified Sampling	- Memastikan keterwakilan semua strata - Memberikan perkiraan yang lebih tepat untuk tiap kelompok	- Membutuhkan pengetahuan tentang karakteristik populasi untuk stratifikasi - Lebih kompleks dan memakan waktu	Populasi menunjukkan subkelompok yang jelas
Cluster Sampling	- Hemat biaya dan efisien waktu - Memberikan representasi geografis yang lebih luas	Memerlukan pertimbangan yang cermat terhadap komposisi dan ukuran cluster	Populasi tersebar secara geografis/mengelompok secara alami
Snowball Sampling	- Efektif untuk mempelajari populasi tersembunyi - Memungkinkan akses ke peserta melalui sosial networking	Potensi bias	Mempelajari topik sensitif/populasi dengan akses terbatas

Hypothesis Testing

Pengertian Hipotesis: Dugaan bersifat sementara sehingga harus dibuktikan menggunakan uji hipotesis berdasarkan data (mengambil kesimpulan tentang populasi berdasarkan sampel)

Menetapkan Hipotesis

- Hipotesis Null/Awal (H_0): baseline assumption (tidak ada perbedaan)
- Hipotesis Alternatif (H_1): Kontradiksi H_0



Memiliki satu/dua arah perbandingan nilai

- Menentukan Kriteria Pengujian
- Uji satu arah (one-tailed)
 - Uji dua arah (two-tailed)

Decision	H_0 Is True	H_0 Is False
Fail to reject H_0	No error	Type II error
Reject H_0	Type I error	No error

Aspek	z-test	t-test
Ukuran Sample	≥ 30	< 30
Apakah varians Populasi diketahui?	Ya	Tidak

Rumus

$$Z = \frac{X - \mu}{\sigma}$$
$$T = \frac{\bar{X} - \mu}{s/\sqrt{n}}$$

• X = value of the data point
• μ = mean of the data set
• σ = standard deviation of the data set
• \bar{X} = sample mean
• s = population mean
• n = sample standard deviation
• n = sample size

A/B Testing

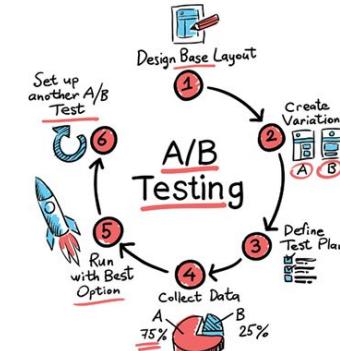
A/B Testing:

Metode eksperimen untuk membandingkan **dua versi** suatu hal — misalnya fitur aplikasi, desain website, atau kampanye marketing — guna mengetahui **mana yang memberikan performa lebih baik**.

- **A** = versi kontrol (yang sudah ada)
- **B** = versi eksperimen (perubahan yang ingin diuji)

Tujuannya:

- Mengambil keputusan berbasis data, bukan asumsi.



Source:

<https://medium.com/@tanyaagarwal9812/ab-testing-always-be-testing-ec0ff4bae1de>

Kenapa A/B Testing Penting?

- Mengurangi risiko saat merilis fitur baru
- Mengukur dampak perubahan secara objektif
- Meningkatkan performa produk secara bertahap (growth optimization)
- Membuktikan sebab-akibat (causality), bukan hanya korelasi

A/B Testing

Bagaimana Cara Kerja A/B Testing?

① Tentukan Hipotesis

Contoh:

“Mengubah warna tombol ‘Checkout’ dari biru menjadi hijau akan meningkatkan conversion rate.”

② Randomisasi Pengguna

Pengguna dibagi acak:

- 50% melihat versi A
- 50% melihat versi B

👉 Ini penting agar hasilnya tidak bias.

③ Kumpulkan Data

Metric yang sering dipakai:

- Conversion rate
- Click-through rate
- Retention
- Revenue per user
- Bounce rate

④ Lakukan Analisis Statistik

Biasanya menggunakan:

- z-test
- t-test
- chi-square test (untuk data kategori)
- p-value & confidence interval

Tujuan: melihat apakah perbedaan A dan B **signifikan secara statistik**, bukan kebetulan.

⑤ Ambil Keputusan

Jika versi B > versi A secara signifikan → versi B di-launch.

Jika tidak signifikan → tetap gunakan A.



Data Visualization

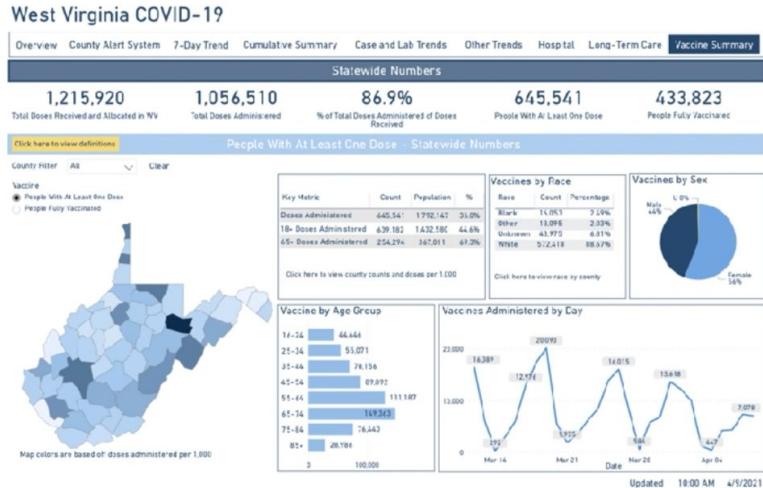
Library, Basic Viz, Matplotlib vs
Seaborn, Hands On

Intro to Data Visualization



Definisi

Visualisasi Data adalah proses untuk **mempresentasikan data** yang terstruktur ataupun tidak dengan menggunakan **grafik** atau **chart** untuk menampilkan informasi yang tersembunyi di dalam data

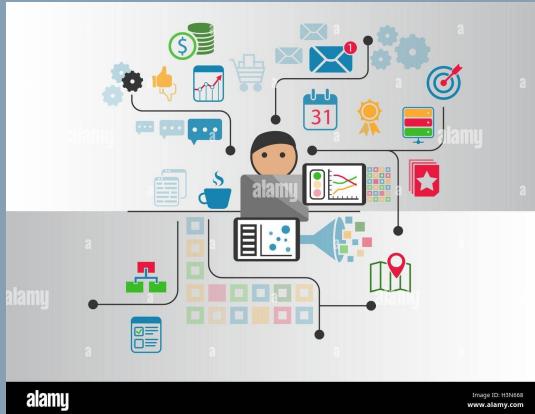


Contoh Dashboard visualisasi data penyebaran Covid 19 dan vaksinasi



Latar Belakang Visualisasi Data

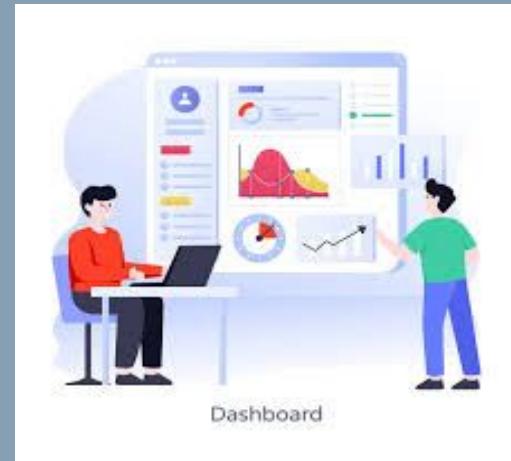
Problem



Dataset ukurannya besar dari segi volume, kecepatan dan juga keragamantipe datanya: Bagaimana agar mudah memahaminya?

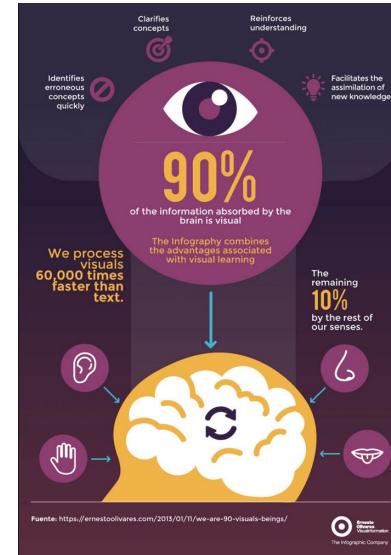
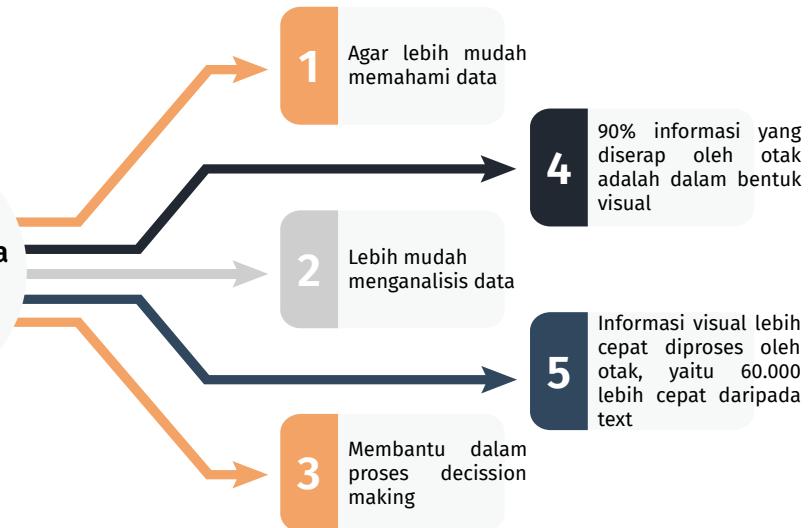
Solusi

1. Faktanya bahwa **manusia senang** dengan tampilan gambar
2. **Ubah** informasi dari **data dalam bentuk grafis**



Pentingnya Visualisasi Data

Mengapa Pentingnya Visualisasi Data ?



Mengapa Membuat Data Menjadi Visual?



Mampu merepresentasikan informasi yang banyak secara singkat dan compact



Menampilkan informasi dari berbagai sudut pandang



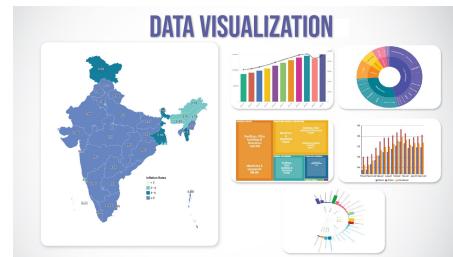
Menampilkan informasi dengan kedalaman tertentu



Mendukung berbagai analisis, misalkan pembanding

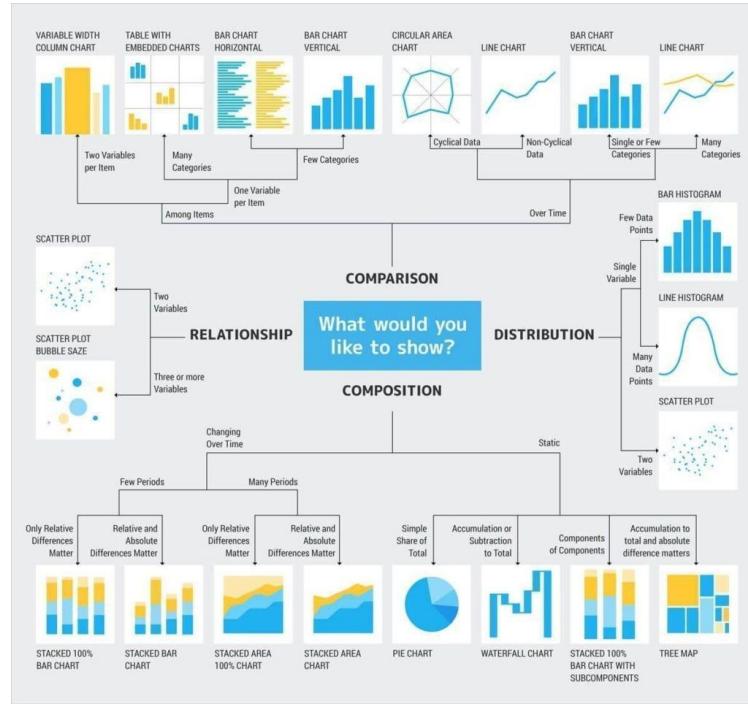


Mendukung cerita dari keseluruhan dan analisis laporan



Source :
<https://www.mospi.gov.in/dataviz-landing-page-1>

Tipe Visualisasi



Source : <https://www.nschaud.fr/2019/08/what-would-you-like-to-show-dataviz/>

Tipe Visualisasi	Tujuan Penggunaan	Jenis Grafik yang Termasuk	Kapan Digunakan
Comparison (Perbandingan)	Membandingkan nilai antar kategori atau waktu	Variable Width Column Chart, Table with Embedded Charts, Horizontal Bar Chart, Vertical Bar Chart, Circular Area Chart (Radar), Line Chart	Ketika ingin mengetahui mana yang paling besar/kecil, tren, atau perbedaan antar item
Relationship (Hubungan)	Menunjukkan hubungan/korrelasi antara dua atau lebih variabel	Scatter Plot, Bubble Scatter Plot	Ketika ingin mengetahui apakah variabel saling mempengaruhi atau memiliki pola tertentu
Distribution (Sebaran)	Melihat variasi, pola distribusi, kepadatan, atau outlier	Bar Histogram, Line Histogram (Density Plot), Scatter Plot (untuk sebaran titik)	Ketika ingin menganalisis kualitas sebaran data seperti normalitas, skew, kluster, atau outlier
Composition (Komposisi)	Menunjukkan bagaimana bagian membentuk total	Stacked 100% Bar Chart, Stacked Bar Chart, Stacked 100% Area Chart, Stacked Area Chart, Pie Chart, Waterfall Chart, Stacked Bar with Subcomponents, Treemap	Digunakan untuk melihat proporsi bagian, perubahan komposisi dari waktu ke waktu, atau bagaimana total terbentuk

Python Library for Visualization: **matplotlib**

Apa itu Matplotlib ?

Matplotlib adalah plotting library 2D yang membantu dalam memvisualisasikan angka.

Mengapa Matplotlib ?

1. Robust, sangat kuat dan serbaguna
2. Open Source / Gratis
3. Mudah untuk digunakan

Cara Import Matplotlib

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

Penjelasan :

matplotlib adalah nama library utamanya
pyplot adalah modul untuk membuat grafik
plt adalah alias (nama pendek) agar mudah dipanggil

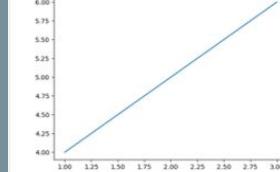
Contoh Penggunaan :

Import Matplotlib ke Python

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

```
❷ plt.plot([1, 2, 3], [4, 5, 6])
```

```
❸ plt.show()
```



❶ import matplotlib.pyplot as plt

```
❷ bulan = ["Jan", "Feb", "Mar", "Apr", "Mai"]
```

```
❸ penjualan = [120, 150, 180, 160, 200]
```

```
❹ plt.plot(bulan, penjualan, marker='o')
```

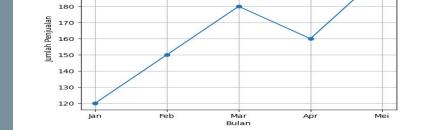
```
❺ plt.title("Line Chart - Penjualan Bulanan")
```

```
❻ plt.xlabel("Bulan")
```

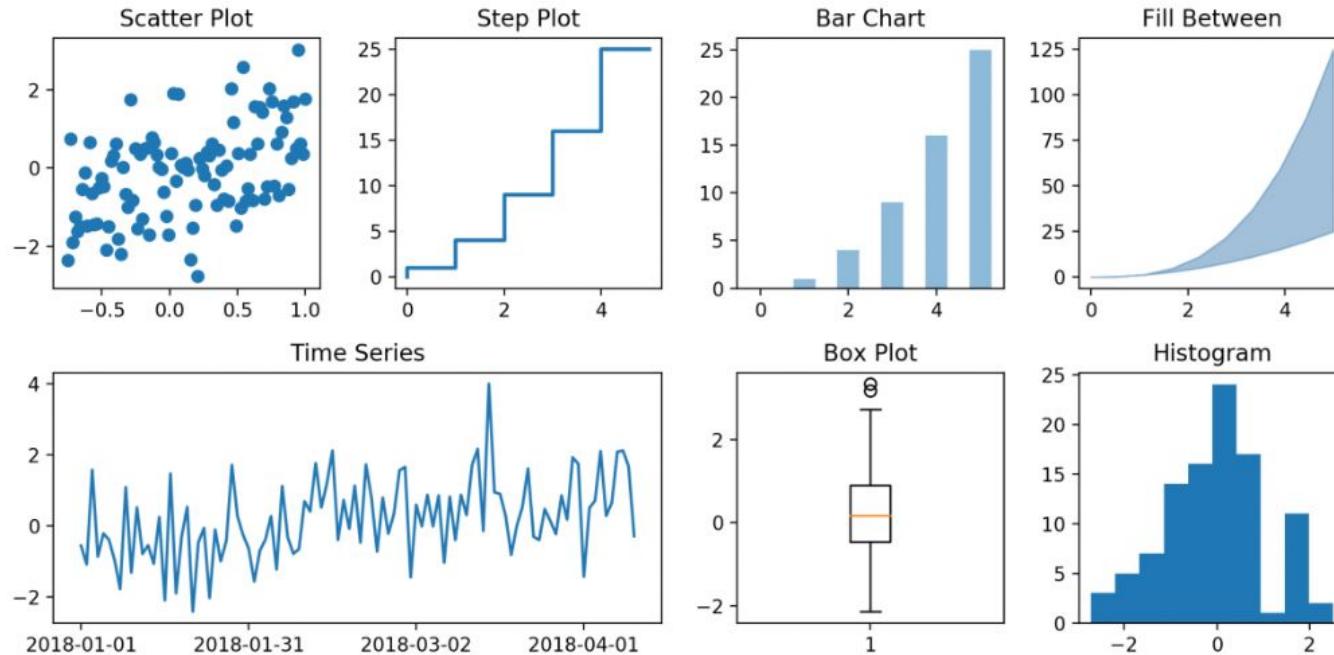
```
❼ plt.ylabel("Jumlah Penjualan")
```

```
❽ plt.grid(True)
```

```
❾ plt.show()
```



Tipe Chart Yang Dibuat Menggunakan Matplotlib



Source : <https://coding-blocks.github.io/DS-NOTES/1.2%20Introduction%20to%20Matplotlib.html>

Python Library for Visualization:



seaborn

Apa itu Seaborn ?

Seaborn adalah library visualisasi data Python berdasarkan matplotlib. Seaborn menyediakan interface tingkat tinggi untuk menggambar grafik statistik yang menarik dan informatif.

Cara Import Seaborn

```
import seaborn as sns
```

Penjelasan :

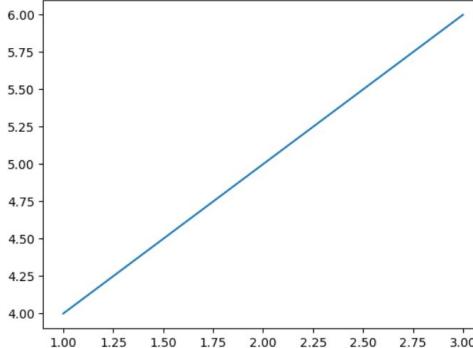
seaborn = nama library

sns = alias (nama pendek) supaya mudah dipanggil

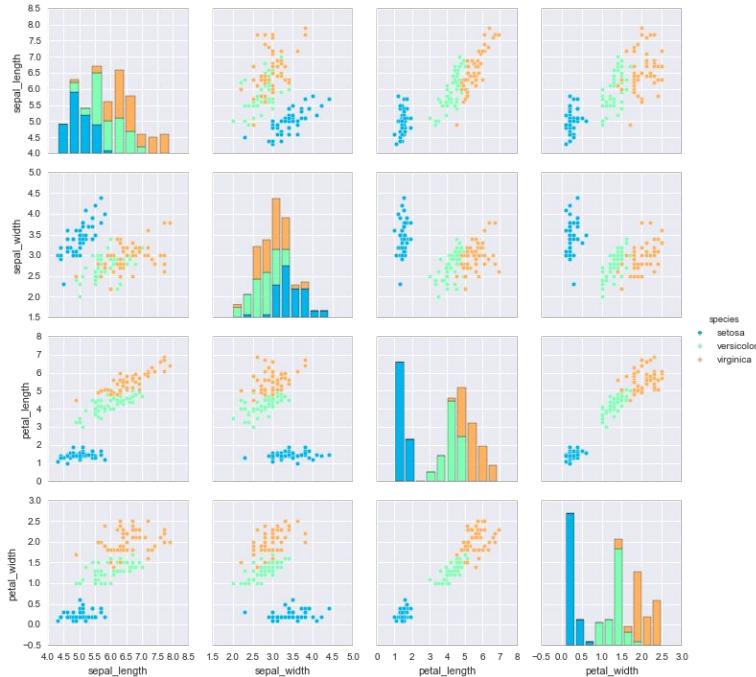
Contoh Penggunaan :

```
1 import seaborn as sns  
  
sns.lineplot(x=[1,2,3], y=[4,5,6])
```

... <Axes: >



Tipe Chart Yang Dibuat Menggunakan Seaborn



Source : <https://www.geeksforgeeks.org/data-science/grid-plot-in-python-using-seaborn/>

Matplotlib Vs Seaborn

matplotlib

Vs

seaborn



- Dapat menggunakan beberapa chart secara simultan
- Mudah untuk dikustomisasi



- Complex dengan syntax yang panjang



- Simple, mudah dipelajari dan dipahami
- Lebih comfortable untuk handling pandas dataframe



- Menggunakan banyak memories
- Tidak ada pie chart

Hands On Data Visualization with Python

1. Buka Google Colaboratory

```
colab.research.google.com/?hl=en-US
```

2. Import Library dan Package Data Visualization

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

3. Buka dataset

```
df = pd.read_csv("https://raw.githubusercontent.com/sivabalanb/Data-Analysis-with-Pandas-and-Python/master/nba.csv")
```

```
df.head()
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0

Hands On Data Visualization with Python

4. Data preparation

```
df.info()  
  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 458 entries, 0 to 457  
Data columns (total 9 columns):  
 #   Column   Non-Null Count  Dtype     
 ---    
 0   Name      457 non-null    object    
 1   Team      457 non-null    object    
 2   Number    457 non-null    float64   
 3   Position   457 non-null    object    
 4   Age       457 non-null    float64   
 5   Height    457 non-null    object    
 6   Weight    457 non-null    float64   
 7   College   373 non-null    object    
 8   Salary    446 non-null    float64  
dtypes: float64(4), object(5)  
memory usage: 32.3+ KB
```

```
df.Age.unique()  
  
array([25., 27., 22., 29., 21., 24., 20., 26., 28., 32., 23., 30., 33.,  
      34., 37., 36., 31., 38., 39., 19., 35., 40., nan])
```

```
df.Age = df.Age.fillna(df.Age.mean().round(0))
```

```
df.Weight.unique()  
  
array([180., 235., 205., 185., 231., 240., 238., 190., 220., 260., 215.,  
      253., 216., 200., 208., 219., 175., 275., 237., 248., 221., 210.,
```

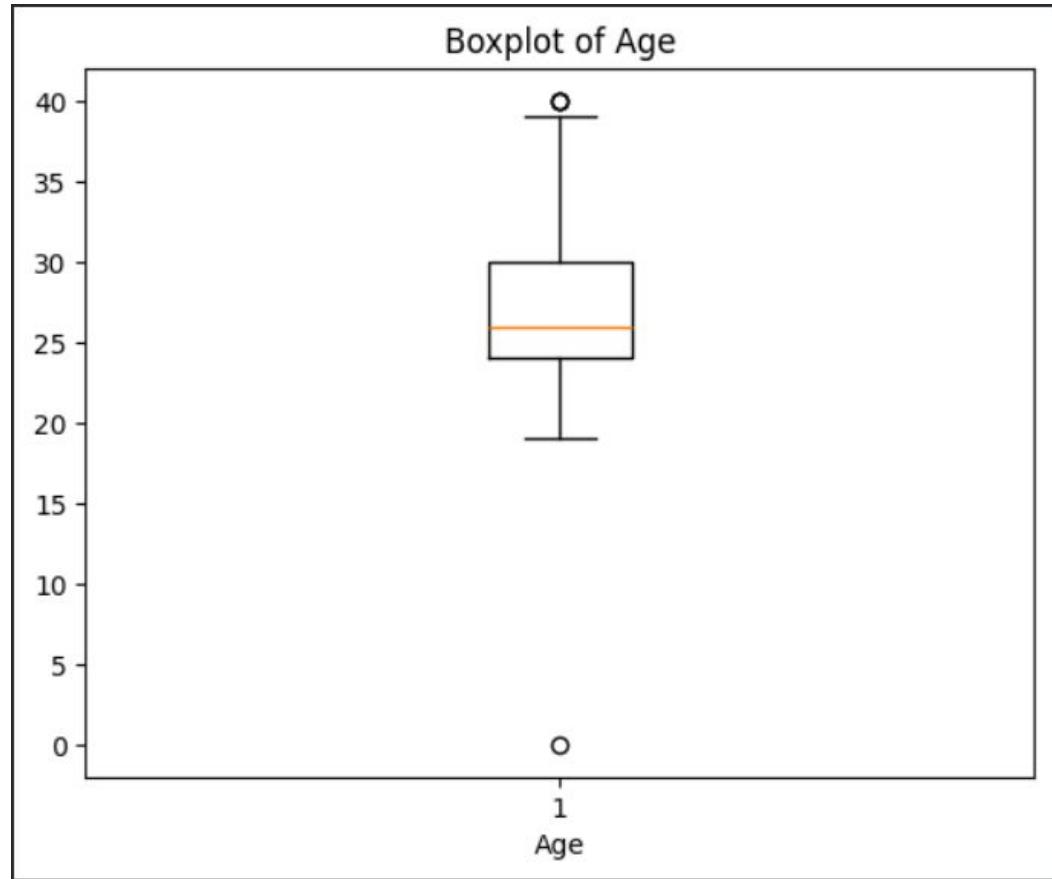
- a. **Membaca dataset**
df.info()
- b. **Membaca data dengan lebih detail, misal per kolom**
df.Age.unique()
- c. **Adakah missing value. Jika ada, handling.**
df.Age =
df.Age.fillna(df.Age.mean().round(0))
- d. **Cek value masing-masing kolom**

Hands On Data Visualization with Python

5. Visualize data

a. ▾ Boxplot

```
▶ import matplotlib.pyplot as plt  
  
# Select variable from dataframe  
data = df.Age  
  
# Creating plot  
plt.boxplot(data)  
plt.xlabel("Age")  
plt.title("Boxplot of Age")  
  
# show plot  
plt.show()
```



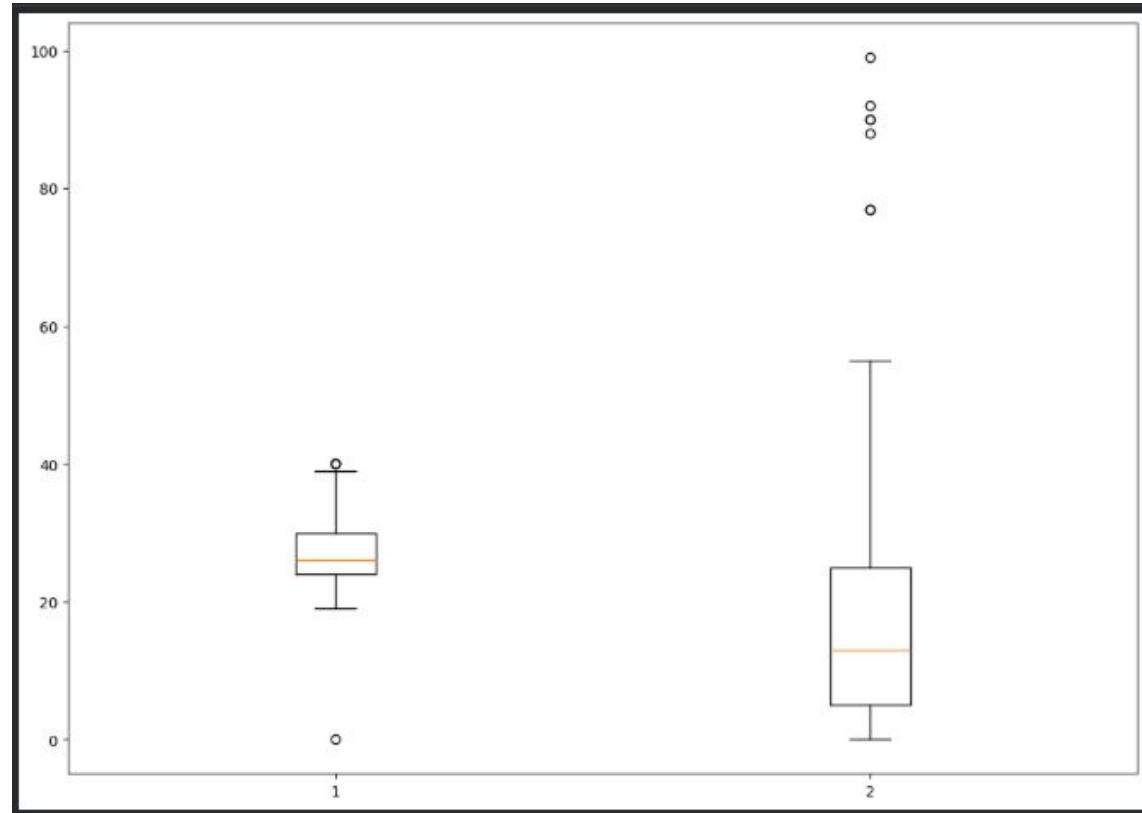
Hands On Data Visualization with Python

5. Visualize data

b. Multiple Boxplot



```
#Select several numerical variables from dataframe  
data = [df.Age, df.Number]  
  
fig = plt.figure(figsize =(10, 7))  
  
# Creating axes instance  
ax = fig.add_axes([0, 0, 1, 1])  
  
# Creating plot  
bp = ax.boxplot(data)  
  
# show plot  
plt.show()
```

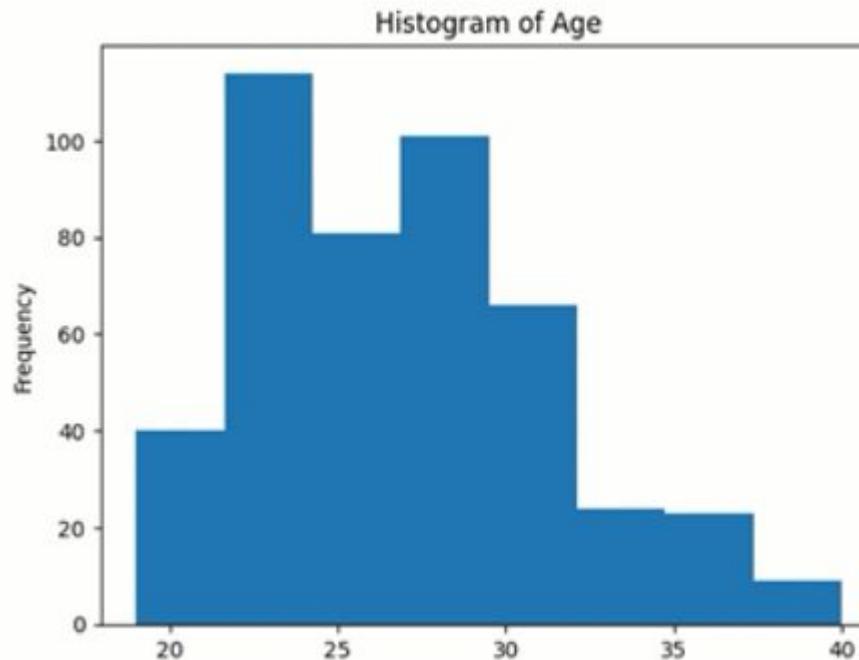


Hands On Data Visualization with Python

5. Visualize data

c. ▾ Histogram

```
▶ # Select variable from the dataset  
data = df.Age  
  
# Creating plot  
plt.hist(data, bins=8)  
plt.xlabel("Age")  
plt.ylabel("Frequency")  
plt.title("Histogram of Age")  
  
# show plot  
plt.show()
```

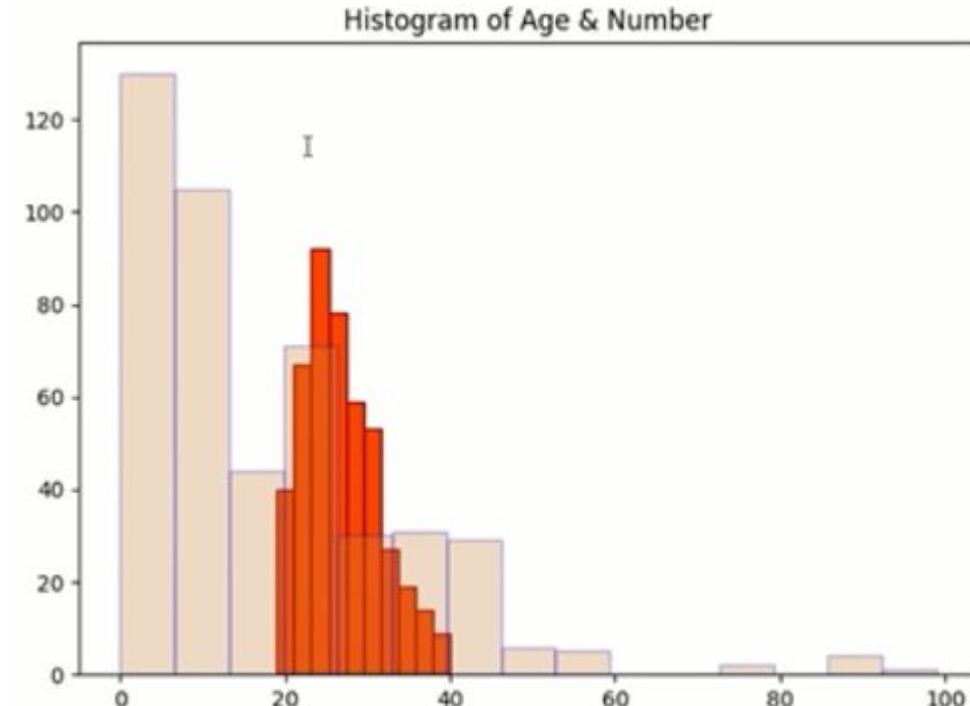


Hands On Data Visualization with Python

5. Visualize data

d. # Multiple Histogram

```
plt.hist(df.Age,
         facecolor = 'orangered',
         edgecolor = 'maroon',
         bins=10)
plt.hist(df.Number,
         facecolor = 'peru',
         edgecolor = 'blue',
         bins=15,
         alpha=0.3)
plt.title("Histogram of Age and Number")
# Adjust the spacing between subplots
plt.tight_layout()
```



Hands On Data Visualization with Python

5. Visualize data

e. Bar plot

```
# Extract the "Position" column from the dataset
position_data = df['Position']

# Get unique values of Position
unique_positions = position_data.unique()

# Count the value of each Position
position_counts = position_data.value_counts().sort_values(ascending=True)

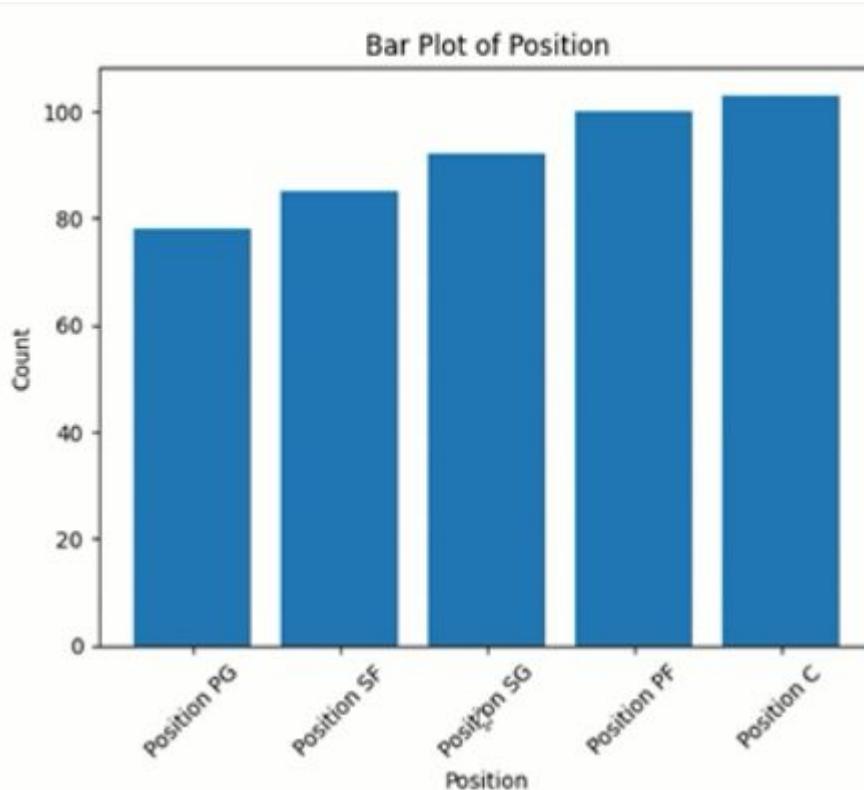
# Define bar labels
bar_labels = [f"Position {position}" for position in unique_positions]

# Create a bar chart
plt.bar(bar_labels, position_counts)

# Add title and axis labels
plt.title("Bar Plot of Position")
plt.xlabel("Position")
plt.ylabel("Count")

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

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



Hands On Data Visualization with Python

5. Visualize data

f. Bar plot antara 2 variabel

```
# Extract the "Position" and "Weight" columns from the dataset
position_data = df['Position']
weight_data = df['Weight']

# Group the data by Position and calculate the average Weight
average_weight_by_position = weight_data.groupby(position_data).mean().sort_values(ascending=True)

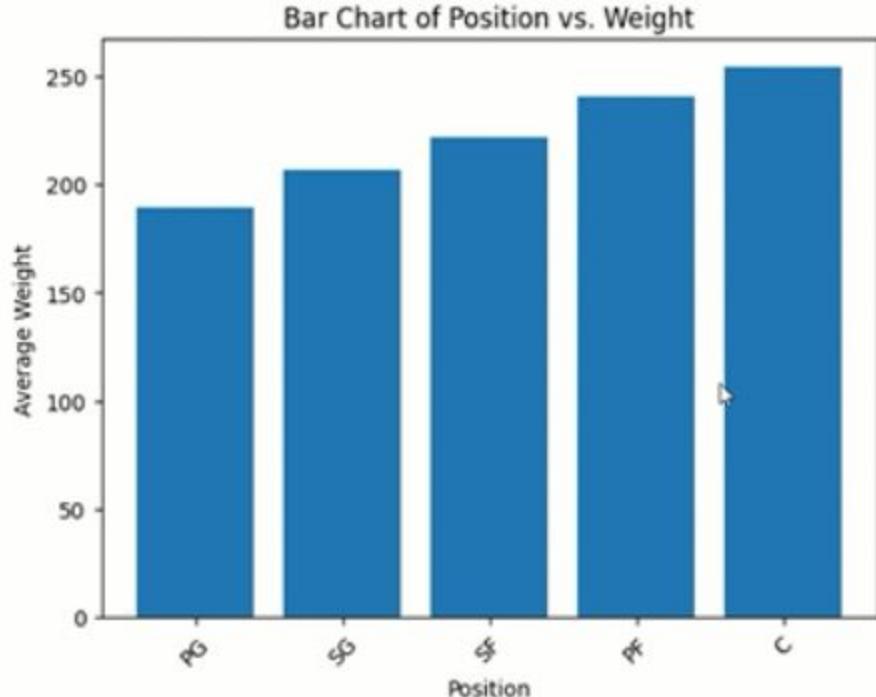
# Define bar labels
bar_labels = average_weight_by_position.index.to_list()

# Create a bar chart
plt.bar(bar_labels, average_weight_by_position)

# Add title and axis labels
plt.title("Bar Chart of Position vs. Weight")
plt.xlabel("Position")
plt.ylabel("Average Weight")

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

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

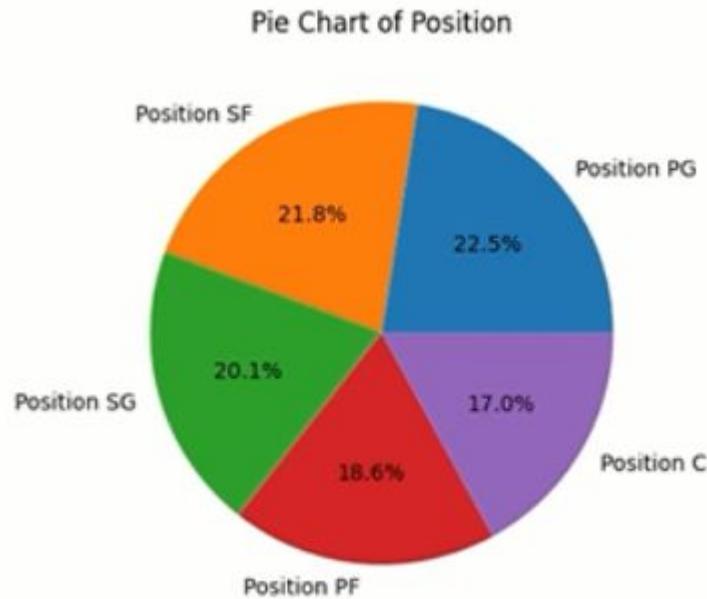


Hands On Data Visualization with Python

5. Visualize data

g. ▾ Pie chart

```
▶ # Extract the "Position" column from the dataset  
position_data = df['Position']  
  
# Get unique values of Position  
unique_positions = position_data.unique()  
  
# Count the frequency of each Position  
position_counts = position_data.value_counts()  
  
# Define labels for the pie chart  
labels = [f"Position {position}" for position in unique_positions]  
  
# Create the pie chart  
plt.pie(position_counts, labels=labels, autopct="%1.1f%%")  
  
# Add title  
plt.title("Pie Chart of Position")  
  
# Show the plot  
plt.show()
```



Hands On Data Visualization with Python

5. Visualize data

h. Scatter plot

```
# Extract the "Age" and "Salary" columns from the dataset  
age_data = df['Age']  
salary_data = df['Salary']  
  
# Create a scatter plot  
plt.scatter(age_data, salary_data)  
  
# Add labels and title  
plt.xlabel("Age")  
plt.ylabel("Salary")  
plt.title("Scatter Plot of Age vs. Salary")  
  
# Show the plot  
plt.show()
```



Hands On Data Visualization with Python

5. Visualize data

i. Line plot

```
# prompt: create a line chart using matplotlib of apple stock price using historical data 1 january 2020 to 14 june 2024 from yahoo finance

!pip install yfinance
import yfinance as yf
import matplotlib.pyplot as plt

# Define the start and end dates
start_date = "2020-01-01"
end_date = "2024-06-14"

# Download Apple stock data
apple_data = yf.download("AAPL", start=start_date, end=end_date)

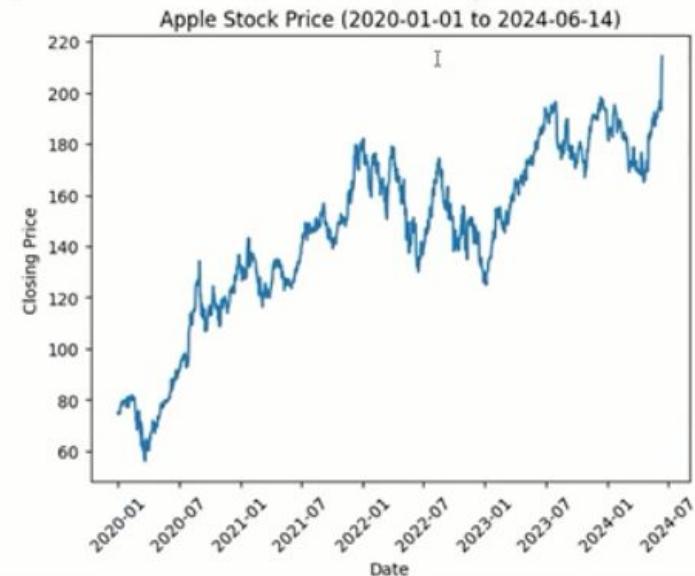
# Extract the closing prices
closing_prices = apple_data["Close"]

# Create the line chart
plt.plot(closing_prices)

# Add title and axis labels
plt.title("Apple Stock Price (2020-01-01 to 2024-06-14)")
plt.xlabel("Date")
plt.ylabel("Closing Price")

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

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



Hands On Data Visualization with Python

6. Mengubah warna



HTML Color Codes

<https://htmlcolorcodes.com> :

HTML Color Codes

Easily find **HTML color codes** for your website using our color picker, color chart and HTML color names with Hex color codes, RGB and HSL values.

Color Names

Green HTML Color Names ; Green, #008000 ; DarkGreen ...

Color Library

Shades of Brown ; Nude, #F2D2BD, rgb(242, 210, 189 ...

Color Chart

Web Safe Color Chart. Web safe colors emerged during the early ...

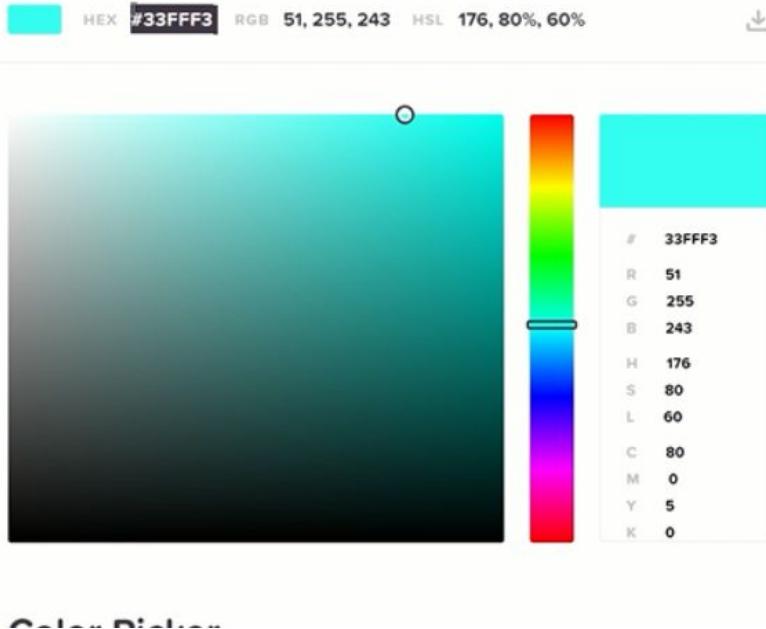
Color Picker

Finding that perfect color is easier than you think. Use our color ...

The screenshot shows the homepage of htmlcolorcodes.com. At the top, there's a navigation bar with links for 'Picker', 'Convert', 'Chart', 'Names', 'Library', 'Tutorials', and 'Resources'. Below the navigation, there's a social sharing section with icons for Twitter, Facebook, Google+, and Email. The main content area features a large heading 'HTML COLOR CODES' and a sub-headline: 'Get HTML color codes, Hex color codes, RGB and HSL values with our color picker, color chart and HTML color names. Let's go!'. Below this, there's a color picker interface. It includes a color wheel, a hex color code field ('#FF5733'), an RGB color code field ('255, 87, 51'), and an HSL color code field ('11, 80%, 60%'). There are also download and share buttons. To the right of the color picker, there's a color palette and a small preview window showing the selected color.

Hands On Data Visualization with Python

6. Mengubah warna

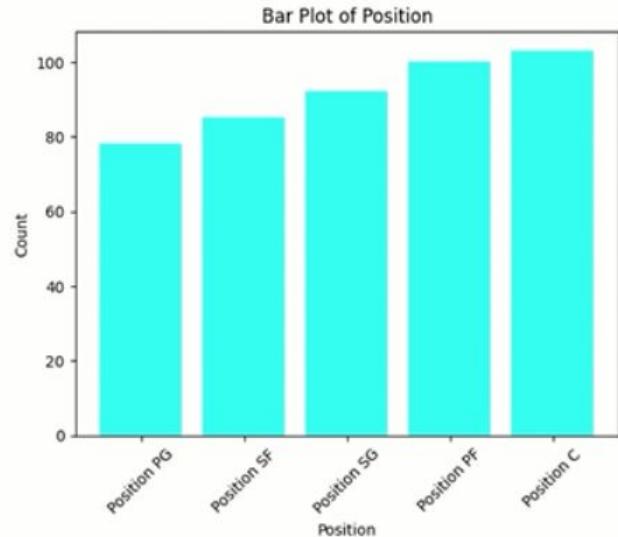


- a. Copy kode Hex
- b. Apply di script

Create a bar chart

```
plt.bar(bar_labels, position_counts, color='#33FFF3')
```

- c. Grafik berubah warna





Google Data Studio

Data Storytelling, BI, Google Locker
Studio

DATA STORYTELLING

Pengertian

Data Storytelling adalah seni dan ilmu dalam menjelaskan suatu data yang berisi kumpulan data menjadi sebuah insight yang berguna melalui visualisasi gambar yang interaktif dan narasi yang mudah dipahami.



Komponen



Visualisasi

Penyampaian insights secara visual agar penyampaian insights lebih efektif dan efisien.



Insights

Kesimpulan dari analisis data



Narasi

Narasi dibangun untuk memperkuat insights yang didapat dan memberikan penjelasan terkait visualisasi data

Google Data Studio

Overview

- Salah satu tools Business Intelligence berbasis cloud berupa visualisasi yang dibuat oleh Google.
- Memungkinkan pembuatan dashboard Business Intelligence yang ringkas dan interaktif.
- Memuat fitur untuk mempermudah visualisasi untuk pengguna non teknis

Integrasi

Nilai lebih Data Studio adalah koneksi dengan berbagai sumber data:

- Produk Google Marketing Platform
- Google Spreadsheet, Youtube, dan Search Console.
- Database: BigQuery, MySQL, PostgreSQL.
- Upload file dari local computer ke Google cloud storage.
- Social media: Facebook, Reddit, Twitter.

Business Intelligence

Pengertian

Teknologi untuk mengumpulkan, mengolah, dan menyajikan data menjadi insight melalui dashboard interaktif guna mendukung keputusan bisnis secara cepat dan akurat.

Mengapa BI Penting?

- Mempercepat keputusan dengan informasi real-time.
- Mengubah data besar menjadi insight yang relevan.
- Meningkatkan efisiensi melalui analisis operasional yang jelas.
- Mendukung keputusan yang data-driven dan meningkatkan profit.

Manfaat Utama BI

- Menyederhanakan data kompleks menjadi informasi strategis.
- Mendukung prediksi dan evaluasi performa bisnis.
- Memungkinkan monitoring proses secara akurat dan berkelanjutan.

Tools Populer BI

- Google Data Studio
- Power BI
- Tableau

Thank You!