

Visualisasi Data

Pengajar:

Dian Ade Kurnia, M.Kom
Rusnanda Farhan
Rusnandi Fikri
Rika Sahriana



Introduction To Data Visualization



Data Visualization

Data visualization atau visualisasi data adalah tentang bagaimana menampilkan grafis dengan tujuan agar audiens dapat memahami dan menyerap informasinya dengan lebih mudah.



Visualisasi Data

Data bisa divisualisasikan dalam bentuk diagram. Ada tiga jenis diagram yang biasanya dipakai, yaitu:

Diagram Batang

Digunakan untuk menampilkan kuantitas data kategori atau value

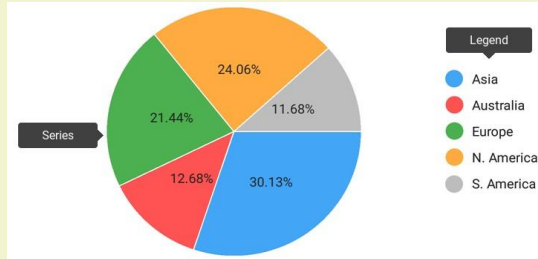


Diagram Pie

Digunakan untuk mengetahui perbandingan antar value.



Diagram Garis

Digunakan untuk mengetahui tren suatu data.

Manfaat Data Visualisasi

Visualisasi data tidak hanya untuk audiens, tetapi data scientist sebagai pengolah data juga sangat membutuhkan data visualisasi.

Pentingnya visualisasi data bagi data scientist:

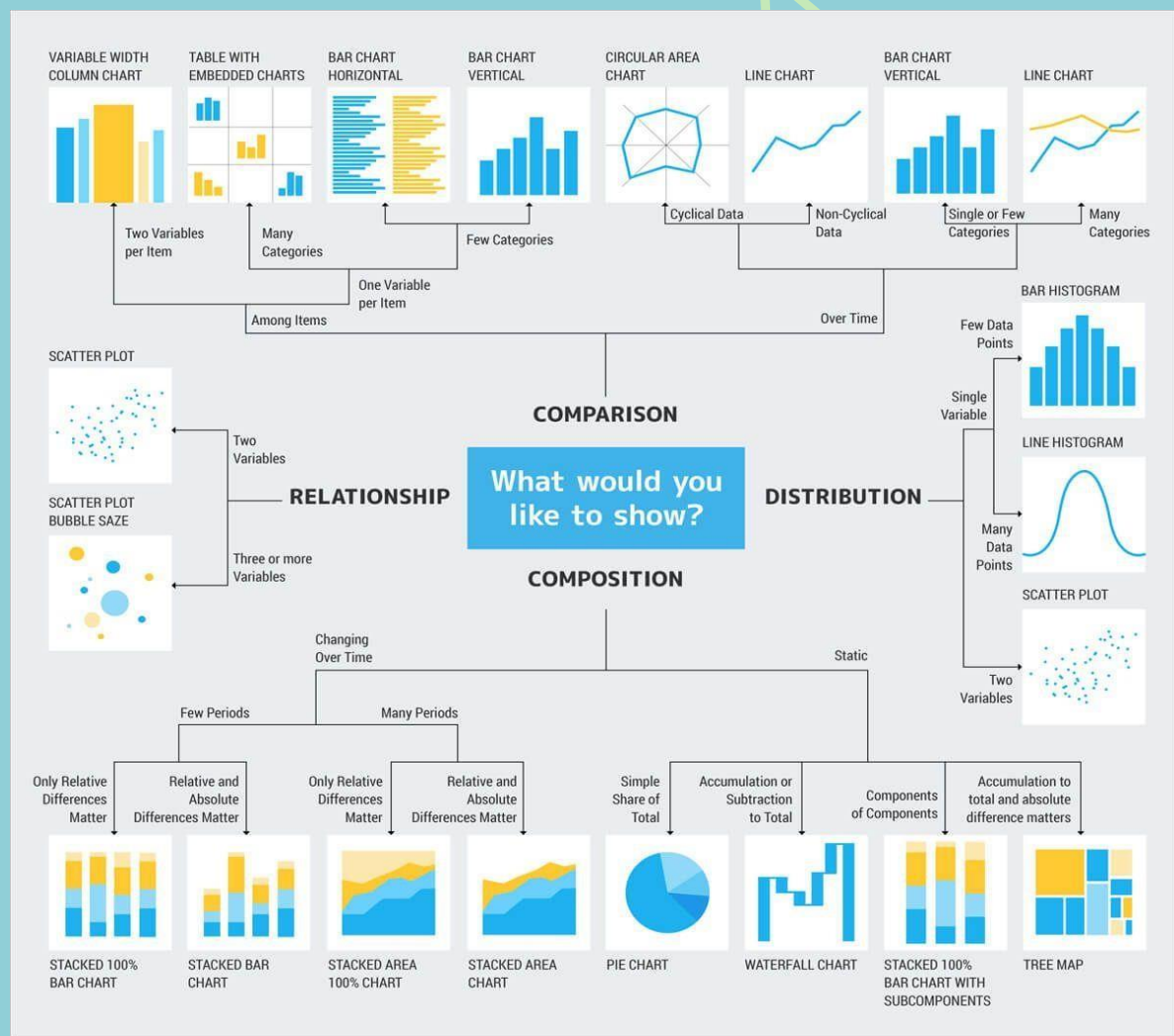
- Memudahkan untuk memahami data yang akan dikerjakan, misalnya saat sedang exploratory data.
- Memudahkan saat membagikan hasil analisis, sehingga tim data bisa memahami informasi data yang sedang dikerjakan.

Visualisasi data bagi stakeholder:

- Memudahkan dalam memahami data
- Memberikan ringkasan singkat tentang isi data
- Membantu pengambilan keputusan yang tepat dan akurat berdasarkan data yang ada misalnya untuk pengambilan keputusan strategi bisnis
- Memberikan gambaran mengenai data



Pemilihan Diagram Untuk Visualisasi Data





Data Visualisasi Pada Python

Dalam merepresentasikan data yang kita miliki, kita perlu memilih jenis chart atau grafik yang tepat. Untuk library yang digunakan dalam ***introduction to data visualization*** ini cukup menggunakan pandas.

```
import pandas as pd
```



Bar Chart

```
data_1 = {'Country':['USA','SGP','JPN','IDN'], 'GDP':[40000,35000,50000,25000]}
```

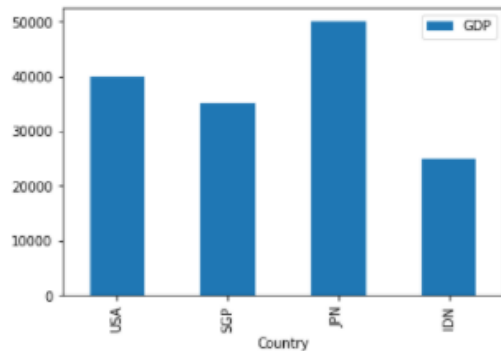
```
df_1 = pd.DataFrame(data_1)
```

```
df_1
```

	Country	GDP
0	USA	40000
1	SGP	35000
2	JPN	50000
3	IDN	25000

```
df_1.plot('Country',kind='bar')
```

<AxesSubplot:xlabel='Country'>

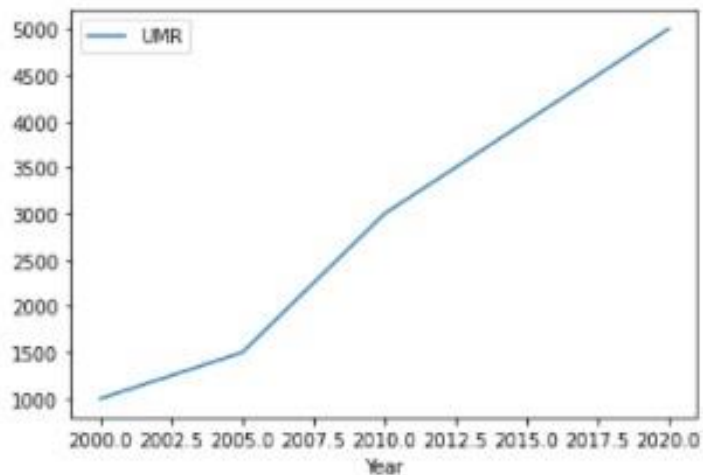


Line Chart

```
data_2 = {'Year': [2010, 2005, 2015, 2020, 2000],  
          'UMR': [3000, 1500, 4000, 5000, 1000]}  
df_2 = pd.DataFrame(data_2)  
df_2_sort = df_2.sort_values('Year')
```

```
df_2_sort.plot(kind='line', x='Year')
```

<AxesSubplot:xlabel='Year'>



Pie Chart

```
data_3 = {'Tasks': [300,500,700]}  
df_3 = pd.DataFrame(data_3,columns=['Tasks'],  
                    index = ['Tasks Pending','Tasks Ongoing','Tasks Completed'])
```

df_3

Tasks	
Tasks Pending	300
Tasks Ongoing	500
Tasks Completed	700

```
df_3.plot(y='Tasks',  
         kind='pie',  
         autopct='%1.1f%%',  
         figsize=(10,5),  
         startangle=90)
```

<AxesSubplot:ylabel='Tasks'>



Scatter Plot

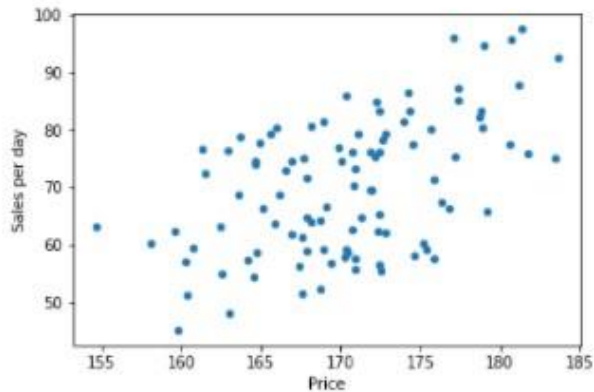
```
np.random.seed(0)
mu = 170 #mean
sigma = 6 #stddev
sample = 100
price = np.random.normal(mu, sigma, sample)
sales_per_day = (price-100) * np.random.uniform(0.75, 1.25, 100)
```

```
data_4 = list(zip(price, sales_per_day))
df_4 = pd.DataFrame(data_4, columns=['Price', 'Sales per day'])
df_4.head()
```

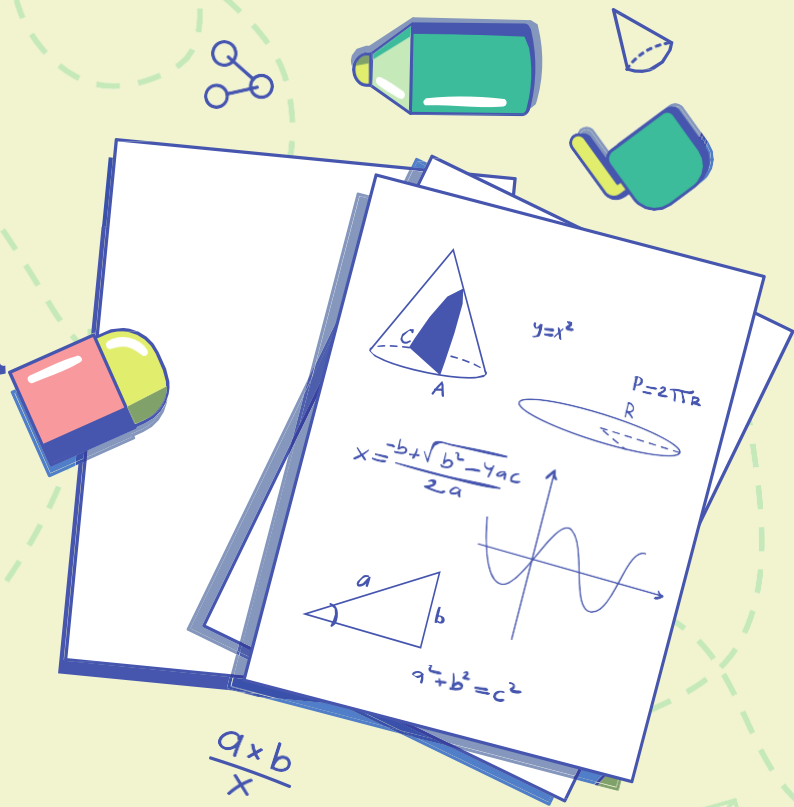
	Price	Sales per day
0	180.584314	77.516270
1	172.400943	76.252428
2	175.872428	57.632438
3	183.445359	75.166529
4	181.205348	87.708822

```
df_4.plot(x='Price', y='Sales per day', kind='scatter')
```

<AxesSubplot: xlabel='Price', ylabel='Sales per day'>



Intermediate Data Visualization





Intermediate Data Visualization

- Library matplotlib dari python berguna untuk melakukan visualisasi data.
- Dataset dapat dipahami dan dipresentasikan dengan mudah dengan visualisasi menggunakan elemen visual seperti chart, graph, maps, dll.
- Visualisasi digunakan untuk melihat tren, mengidentifikasi outlier, dan pola datasetnya seperti apa.
- Dalam mempresentasikan dataset, kita harus dapat menggunakan jenis visualisasi yang tepat.





Plotting In Matplotlib

1. Pie chart

Berbentuk seperti lingkaran dan dibagi menjadi irisan untuk melihat komposisi data biasanya dalam persentase.

2. Bar Plot


Biasanya berfungsi untuk comparison atau membandingkan beberapa variabel, numerik maupun kategori.

3. Histogram

Berfungsi untuk melihat distribusi data frekuensi dan korelasi antar variabel.

4. Scatter Plot

Sama halnya dengan histogram, scatter plot juga berfungsi untuk melihat distribusi data namun dalam bentuk kumpulan titik.





Plotting In Matplotlib

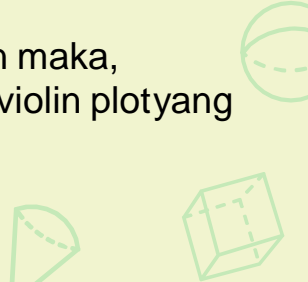
5. Box Plot

Berfungsi untuk melihat sebaran data dengan menggambarkan kumpulan data numerik berdasarkan nilai kuartilnya dan untuk menentukan adanya outlier pada data.

6. Line Plot

Berfungsi untuk membandingkan beberapa variabel dan melihat trend data dari berbagai waktu tertentu.

7. Violin Plot

- Violin plot merupakan penggabungan antara dua metode yaitu boxplot dan Estimasi Kepadatan Kernel (KDE).
 - Berfungsi untuk memudahkan pengguna menganalisis distribusi data yang kontinu untuk setiap kategori.
 - Sesuai dengan KDE, semakin cembung grafik data violin plot yang divisualisasikan maka, kepadatan data peluangnya semakin besar. Sebaliknya, semakin pipih grafik data violin plot yang divisualisasikan maka, kepadatan data peluangnya semakin kecil.
- 

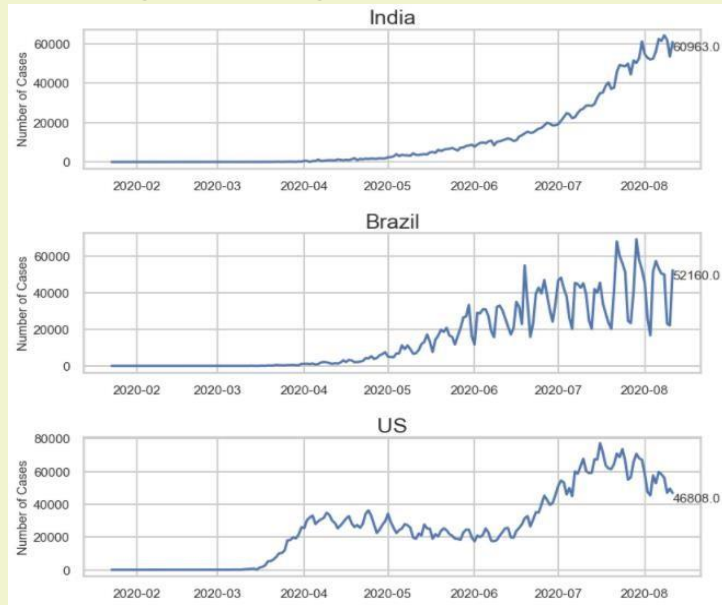
Custom Visualization In Matplotlib

1. Subplot

Fungsi dari subplot adalah membuat multi grafik dalam satu figure.

Syntax:

matplotlib.pyplot.subplot(*args, **kwargs)



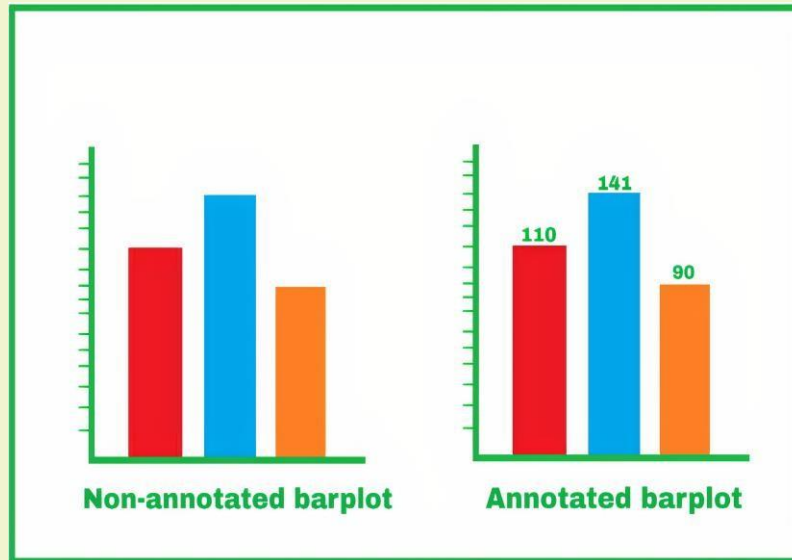
Custom Visualization In Matplotlib

2. Annotation

Berfungsi untuk menampilkan keterangan/arti dari sebuah visualisasi

Syntax:

`matplotlib.pyplot.annotate(text, xy, *args, **kwargs)`



Custom Visualization In Matplotlib

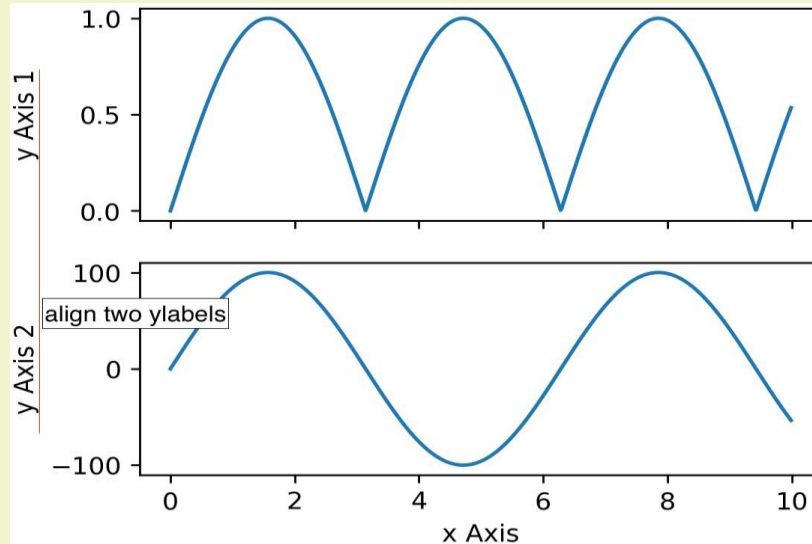
3. Axis

Berfungsi untuk menentukan batas dari sumbu x dan y.

Syntax:

`matplotlib.pyplot.xlabel(xlabel, fontdict=None, labelpad=None, *, loc=None, **kwargs)`

`matplotlib.pyplot.ylim(*args, **kwargs)`



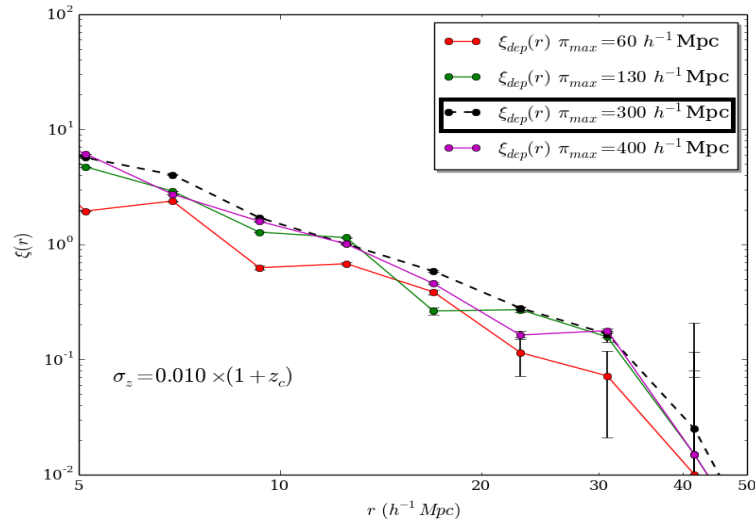
Custom Visualization In Matplotlib

4. Legend

Merupakan keterangan tambahan untuk menjelaskan unsur-unsur pada grafik.

Syntax:

matplotlib.pyplot.legend(*args, **kwargs)



Advance Data Visualization



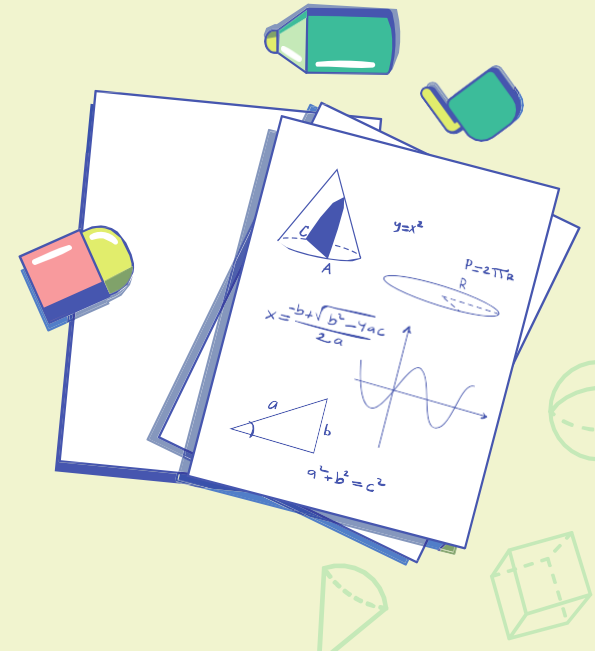
Exploratory Data Analysis

Exploratory data analysis adalah apa yang Anda lakukan untuk membiasakan diri dengan data. Tujuan utama analisis data eksplorasi adalah untuk memenuhi rasa ingin tahu dan menjawab pertanyaan..

Apa yang harus dilakukan dalam Eksplorasi?

Berikut adalah langkah demi langkah analisis eksplorasi:

1. Memahami konteks data.
2. Ajukan pertanyaan dan tuliskan.
3. Pilih metode apa untuk mendapatkan wawasan.
4. Menafsirkan visualisasi.
5. Ulangi.



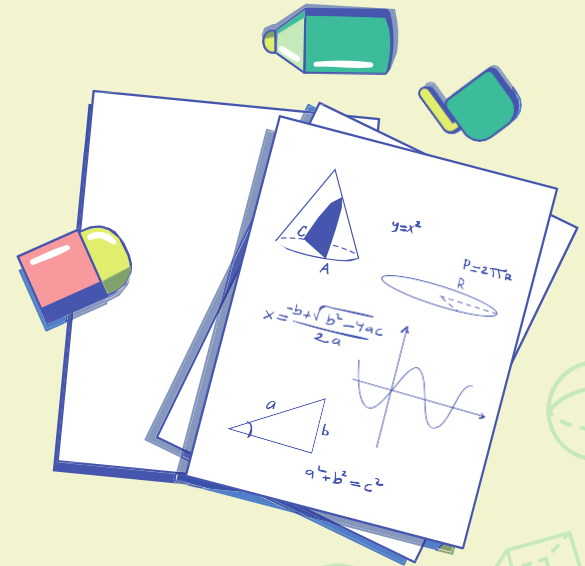
Explanatory Data Analysis

Explanatory Data Analysis adalah apa yang terjadi ketika Anda memiliki sesuatu yang spesifik yang ingin Anda tunjukkan kepada audiens - mungkin tentang 1 atau 2 batu permata berharga itu. Tujuan utama dari analisis data penjelas adalah untuk membuat audiens memahami betapa berharganya permata itu.

Apa yang harus dilakukan di Explanatory?

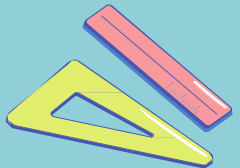
Berikut adalah cara melakukan analisis data eksplanatori:

1. Buat narasi/cerita.
2. Kumpulkan wawasan penting dari proses eksplorasi yang relevan dengan cerita.
3. Pahami siapa audiensnya (mis. CEO, Investor, sesama analis, anak-anak).
4. Pilih grafik yang tepat untuk penonton.
5. Manfaatkan warna, ukuran, ruang dengan bijak untuk menekankan konten.



SweetViz

Sweetviz adalah pustaka Python sumber terbuka yang menghasilkan visualisasi kepadatan tinggi yang indah untuk memulai EDA (Analisis Data Eksplorasi) hanya dengan dua baris kode. Output adalah aplikasi HTML mandiri sepenuhnya. Sistem ini dibangun dengan memvisualisasikan nilai target dengan cepat dan membandingkan kumpulan data. Tujuannya adalah untuk membantu analisis cepat karakteristik target, data pelatihan vs pengujian, dan tugas karakterisasi data lainnya.



SweetViz

Compare 2 dataframes (e.g. Test vs Train),
or 2 subsets of the same dataframe
(e.g. Male vs Female)

Pearson correlation, uncertainty
coefficient and correlation ratio

	Train	Test
ROWS	891	418
DUPPLICATES	0	0
RAM	326.2 kb	150.0 kb
FEATURES	(1 SKIPPED) 11	11 (1 SKIPPED)
CATEGORICAL	5	5
NUMERICAL	2	2
TEXT	3	3

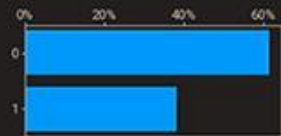
Associations

Associations

Survived

VALUES: 891 (100%)
MISSING: —
DISTINCT: 2 (0%)

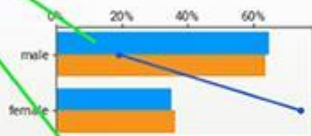
Isolate target feature
and see distribution for every variable



Sex

VALUES: 891 (100%)
MISSING: —
DISTINCT: 2 (0%)

Missing, distinct, numerical and
categorical analysis



Age

VALUES: 714 (80%)
MISSING: 177 (20%)
DISTINCT: 88 (10%)

MAX	80.0	76.0
95%	56.0	57.0
Q3	38.0	39.0
AVG	29.7	30.3
MEDIAN	28.0	27.0
Q1	20.1	21.0
5%	4.0	8.0
MIN	0.4	0.2

RANGE	79.6	75.8
IQR	17.9	18.0
STD	14.5	14.2
VAR	211	201
KURT	0.178	0.084
SKEW	0.389	0.457
SUM	21,205	10,050



Name

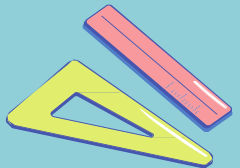
VALUES: 891 (100%)
MISSING: —
DISTINCT: 891 (100%)

0%	—	Harris, Mr. Henry Birkhardt
0%	—	Rice, Mrs. William (Margaret Norton)
0%	—	Seward, Mr. Frederic Kimber
0%	—	Trout, Mrs. William H (Jessie L)
0%	—	Olsson, Mr. Nils Johan Goransson

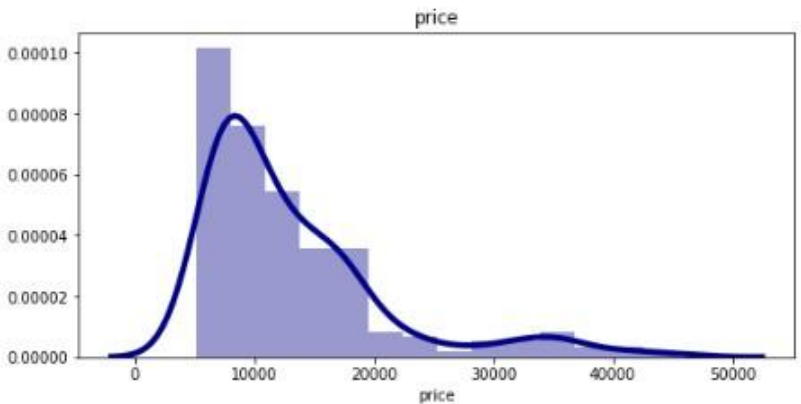
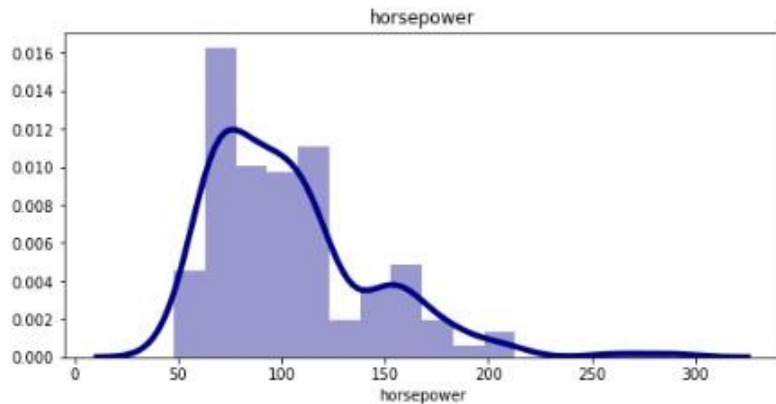
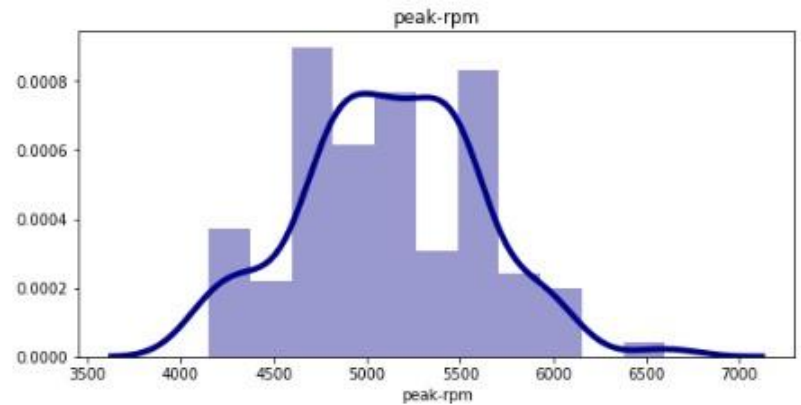
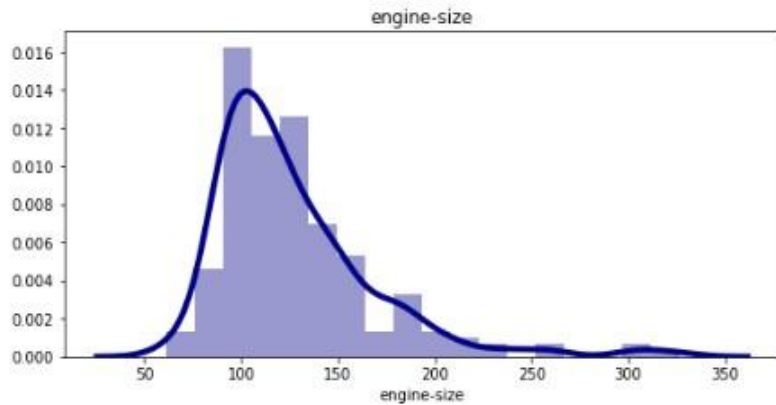
And more!

AutoViz

AutoViz adalah mesin visualisasi sekali klik: Ini menciptakan grafik yang kuat yang siapa pun dari pemula hingga ahli dapat menggunakannya. AutoViz tahu membuat bagan dari data apa pun secara manual itu sulit: Lebih sulit lagi jika Anda tidak tahu apa yang ada di dalamnya. AutoViz dimulai dengan terlebih dahulu menganalisis data Anda untuk mengetahui apakah itu masalah Klasifikasi, Regresi, Tidak Terawasi, atau Seri Waktu. Kemudian memilih grafik terbaik untuk memaksimalkan wawasan Anda.

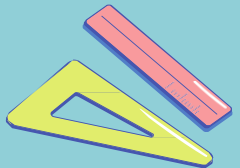


AutoViz



Plotly Express

Pustaka grafik Python Plotly memudahkan pembuatan grafik interaktif berkualitas publikasi. Itu juga dapat membuat bagan serupa seperti Matplotlib dan seaborn seperti plot garis, plot sebar, bagan area, bagan batang, dll. Plotly juga memudahkan untuk membuat plot interaktif. Plot interaktif tidak hanya cantik tetapi juga memudahkan pemirsa untuk melihat lebih dekat setiap titik data.



Plotly Express

Plotly Express Guide

CHART TYPES, ANNOTATIONS, BUTTONS, TOOLTIPS

`px.chart_type(df, parameters)`

