



Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Republik Indonesia



MICROCREDENTIAL: ASSOCIATE DATA SCIENTIST

01 November – 10 Desember 2021

Pertemuan ke-6

Data Understanding 2: Menelaah Data dengan Visualisasi



ditjen.dikti



@ditjendikti



ditjen.dikti



Ditjen Dikristek



<https://dikti.kemdikbud.go.id/>

Profil Pengajar: Nama Lengkap dan Gelar Akademik

Poto
Pengajar

Contak Pengajar:

Ponsel:

xxxxxx

Email:

xxxxxxx

Jabatan Akademik:

Latar Belakang Pendidikan:

- S1:
- S2:
- S3:

Riwayat/Pengalaman Pekerjaan:

- Dosen
- Xxx
- Xxx
- Xxx
- xxx

Unit Kompetensi

- Data Understanding: Menelaah Data dengan Metode Statistik
(UK J.62DM100.005.1 - Menelaah Data)
 - Menganalisis tipe dan relasi data
 - Menganalisis karakteristik data
 - Kriteria Unjuk Kerja (KUK) 2.1 Statistika Dasar

Menelaah Data

KODE UNIT : J.62DMI00.005.1

JUDUL UNIT : Menelaah Data

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam menelaah data untuk *data science*.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Menganalisis tipe dan relasi data	<p>1.1 Tipe data yang terkumpul diidentifikasi sesuai tujuan teknis</p> <p>1.2 Nilai atribut data yang terkumpul diuraikan sesuai dengan batasan konteks bisnisnya</p> <p>1.3 Relasi antar data yang terkumpul diidentifikasi sesuai dengan tujuan teknis</p>
2. Menganalisis karakteristik data	<p>2.1 Karakteristik data yang terkumpul disajikan dengan deskripsi statistik dasar</p> <p>2.2 Karakteristik data yang terkumpul disajikan dengan visualisasi grafik</p> <p>2.3 Hasil penyajian data dianalisis karakteristiknya untuk telaah data</p>
3. Membuat laporan telaah data	<p>3.1 Hasil analisis <u>didokumentasikan</u> dalam bentuk laporan sesuai dengan tujuan teknis</p> <p>3.2 Hipotesis disusun berdasar hasil analisis sesuai tujuan teknis <i>data science</i></p>

Data
Understanding
Documentation

1. Konteks variabel

- 1.1 Data yang terkumpul adalah data yang sudah diintegrasikan dari proses mengumpulkan data pada tahap sebelumnya yang sesuai kebutuhan *data science*.
- 1.2 Tipe data termasuk di dalamnya tipe dan nilai datanya.
- 1.3 Deskripsi statistik dasar adalah analisis statistik meliputi nilai maksimum, minimum, rerata, median, modus, *skewing*, persentil, distribusi, *outliers* dan lain sejenisnya.

Tujuan Pembelajaran

- Modul ini berisi penjelasan mengenai modul visualisasi.
- Visualisasi akan dijelaskan dalam bentuk visualisasi variable dan visualisasi untuk menjelaskan statistic dalam suatu dataset.
- Peserta diharapkan mendapat insight, pengalaman, dan memiliki kemampuan untuk melakukan visualisasi data sesuai dengan kebutuhan.



Outline

Visualisasi variabel

- Pie Chart
- Bar Chart
- Line Graphs
- Scatter Plot
- Heatmap

Visualisasi Statistik

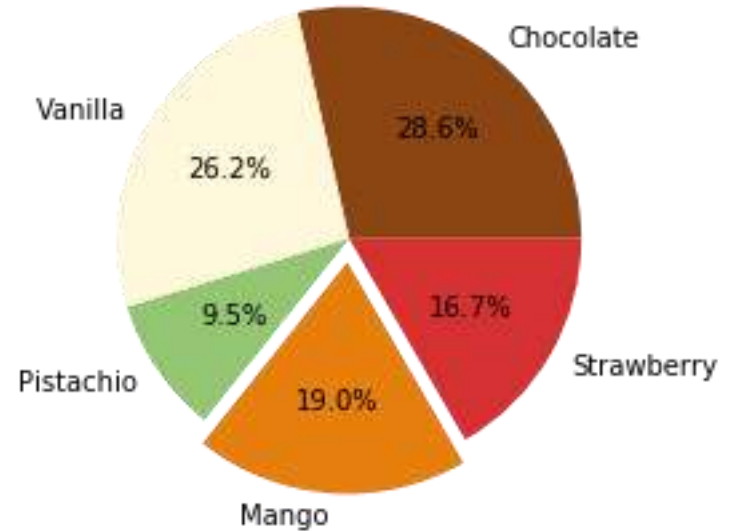
- Histogram
- Correlation
- Descriptive Statistik
- Grouping (Pivot)
- ANOVA

Visualisasi

- **Visualisasi** berperan peran penting dalam bidang machine learning dan data science. Seringkali kita perlu menyaring informasi kunci yang ditemukan dalam sejumlah data data menjadi bentuk yang bermakna dan mudah dicerna.
- **Visualisasi** yang baik dapat menceritakan sebuah cerita tentang data Anda dengan cara yang tidak dapat dilakukan oleh sebuah kalimat.
- Pada pelatihan ini akan mengeksplorasi beberapa teknik visualisasi yang umum. Pelatihan ini menggunakan toolkit seperti Matplotlib's Pyplot dan Seaborn untuk membuat gambar informatif yang memberikan informasi dan pengetahuan mengenai dataset.

Pie Chart

- **Pie chart** digunakan untuk menunjukkan seberapa banyak dari setiap jenis kategori dalam dataset berbanding dengan keseluruhan.
 - Variabel label berisi tupel rasa es krim
 - Variabel voting berisi tupel voting
 - Data tersebut mewakili jumlah voting rasa es krim favorit



Pie Chart (Hands-on)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import library

```
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)
```

Data

Warna chart

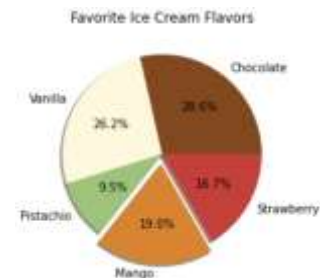
Highlight data "mango"

```
plt.title('Favorite Ice Cream Flavors')
```

Judul pie chart

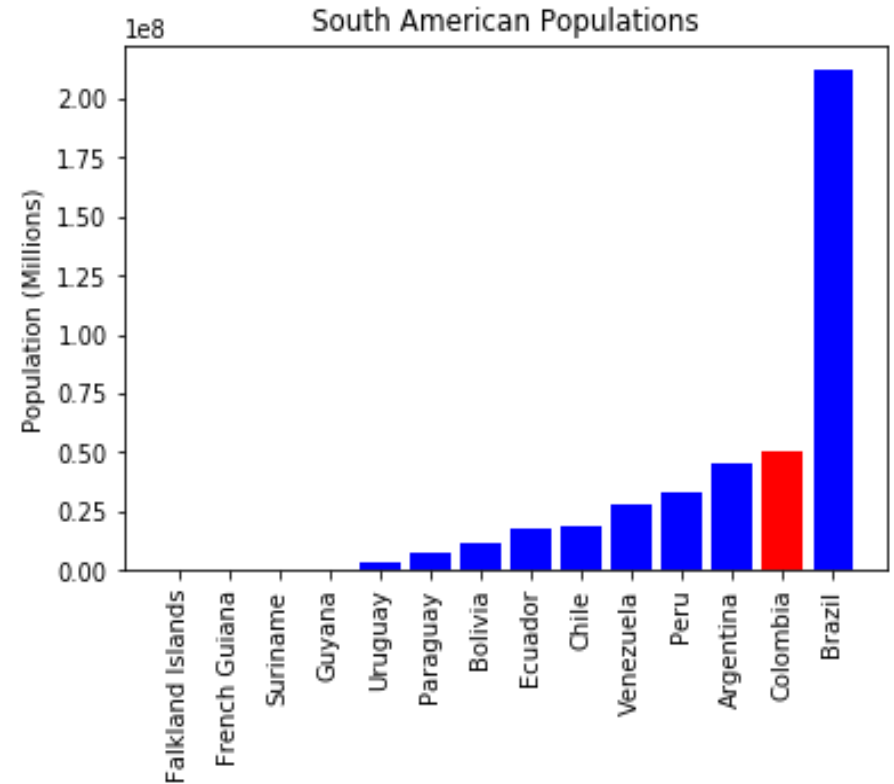
```
plt.pie(
    votes,
    labels=flavors,
    autopct='%1.1f%%',
    colors=colors,
    explode=explode,
    shadow=True
)
plt.show()
```

Visualisasi pie
chart



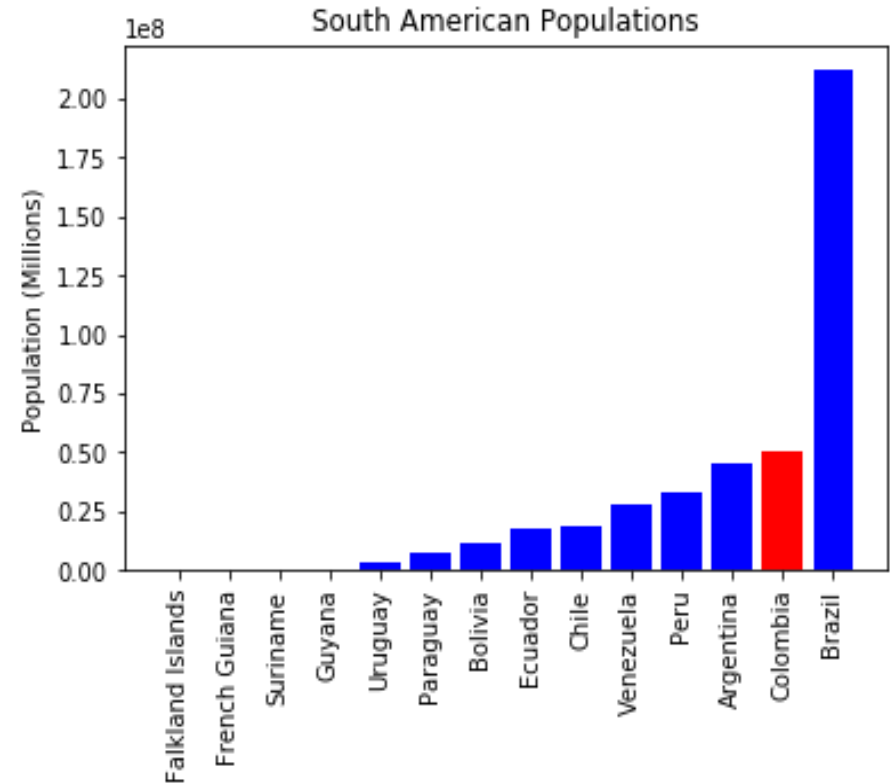
Bar Chart

- **Bar Chart** adalah merupakan tools visualisasi yang dapat digunakan untuk membandingkan data kategorikal.
- Mirip dengan diagram lingkaran, diagram ini dapat digunakan untuk membandingkan kategori data satu sama lain.
- Diagram batang dapat menampilkan lebih banyak kategori data daripada diagram lingkaran.



Bar Chart

- Mari kita mulai dengan melihat diagram batang yang menunjukkan populasi setiap negara di Amerika Selatan.
- Visualisasi ditunjukkan dengan cara mengurutkan dari negara yang memiliki populasi terbesar ke populasi terendah.
- Highlight ditunjukkan untuk negara Colombia



Bar Chart (Hands-on)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import library

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

populations = (45076704, 11626410, 212162757, 19109629, 50019826, 17579085,
               3481, 287750, 785409, 7107305, 32880332, 585169, 3470475,
               28258770)
```

Data

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

Convert menjadi dataframe

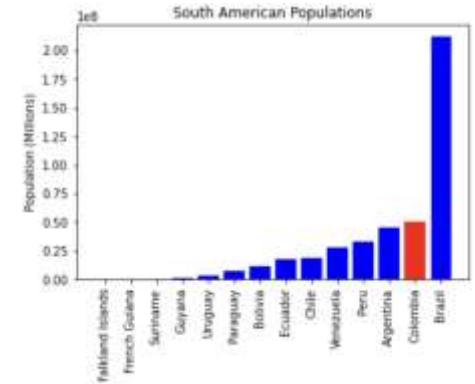
```
df.sort_values(by='Population', inplace=True)
```

Urut berdasarkan "Population"

```
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()
```

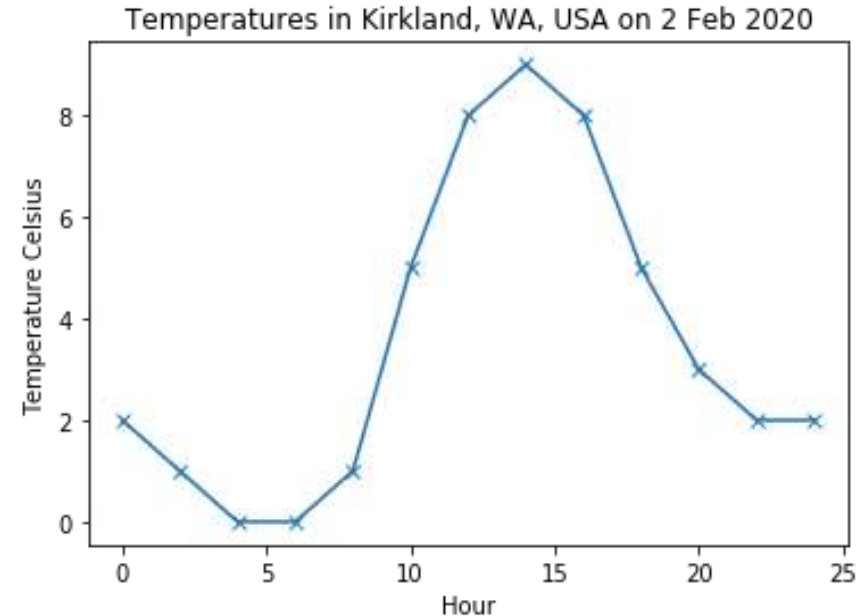
Atur warna chart

Visualisasi Bar Chart



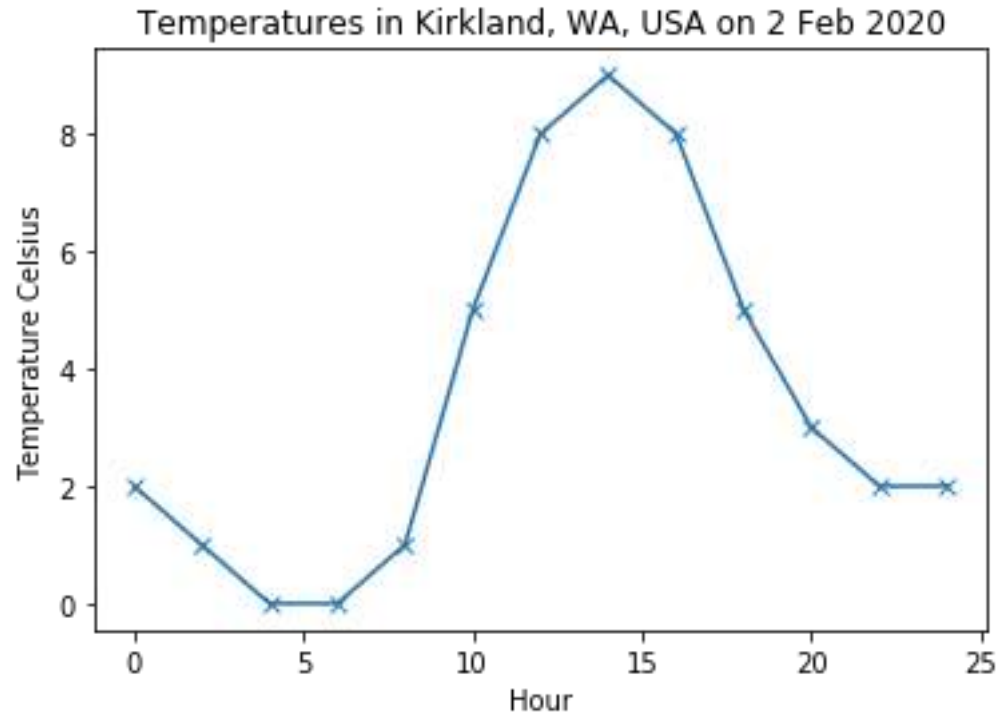
Line Graph

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



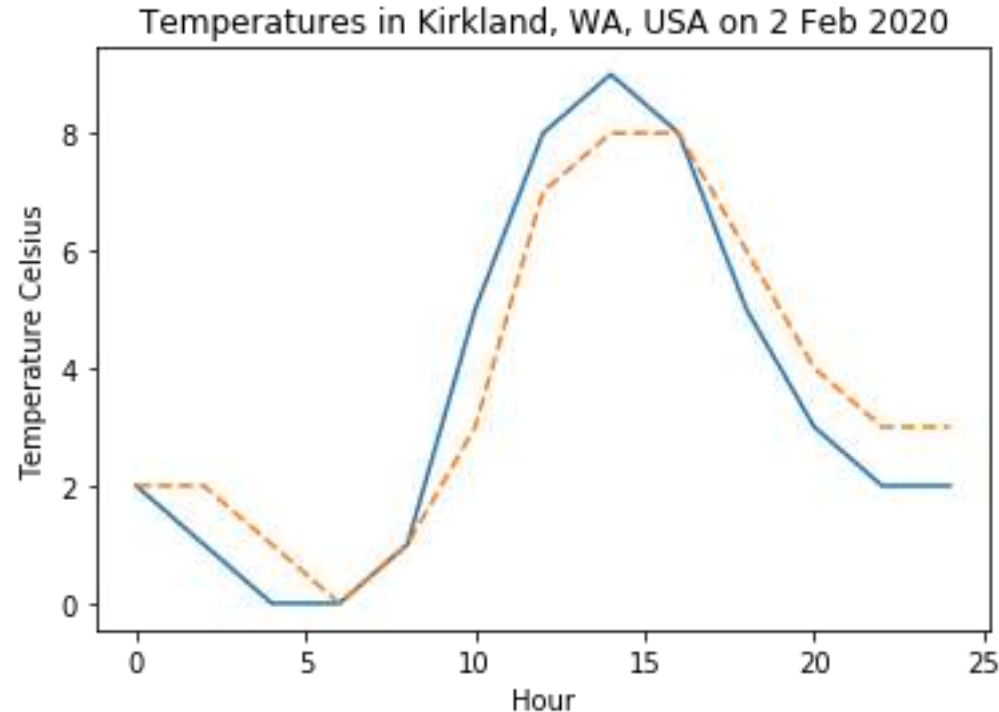
Line Graph

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



Line Graph

- Kita bahkan dapat memiliki beberapa garis pada grafik yang sama didalam satu gambar
- Biasanya kita mengilustrasikan dua line graph untuk menggambarkan dua data yaitu data aktual dan data prediksi.



Line Graph (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```



Import library

```
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()
```



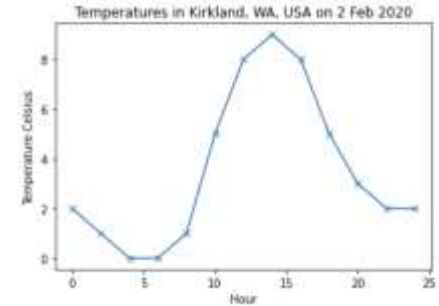
Data



Menambah tanda "x"



Visualisasi linegraph

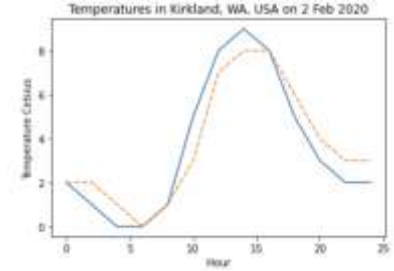


Line Graph (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```



Import library



```
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]
```



Data

```
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()
```



Memberi garis putus-putus



Judul



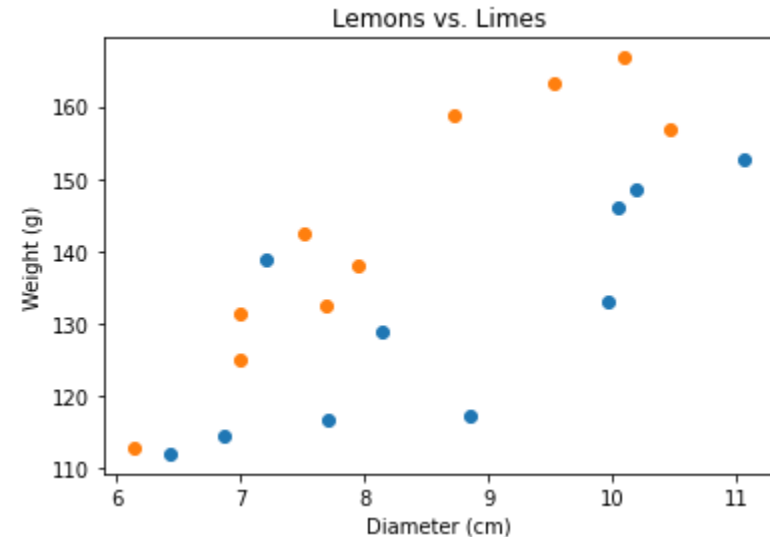
Nama label "Y"



Nama label "X"

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 perbedaan lemon dan lime berdasarkan karakteristik fisiologis.
 - Berat (g)
 - Diameter (cm)



Scatter Plot (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```



Import library

```
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]
```



Data

```
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()
```



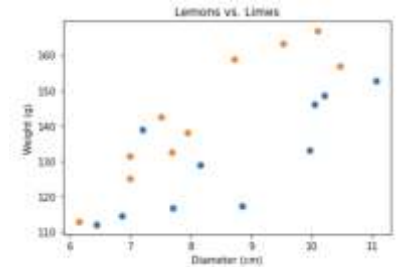
Judul



Nama label

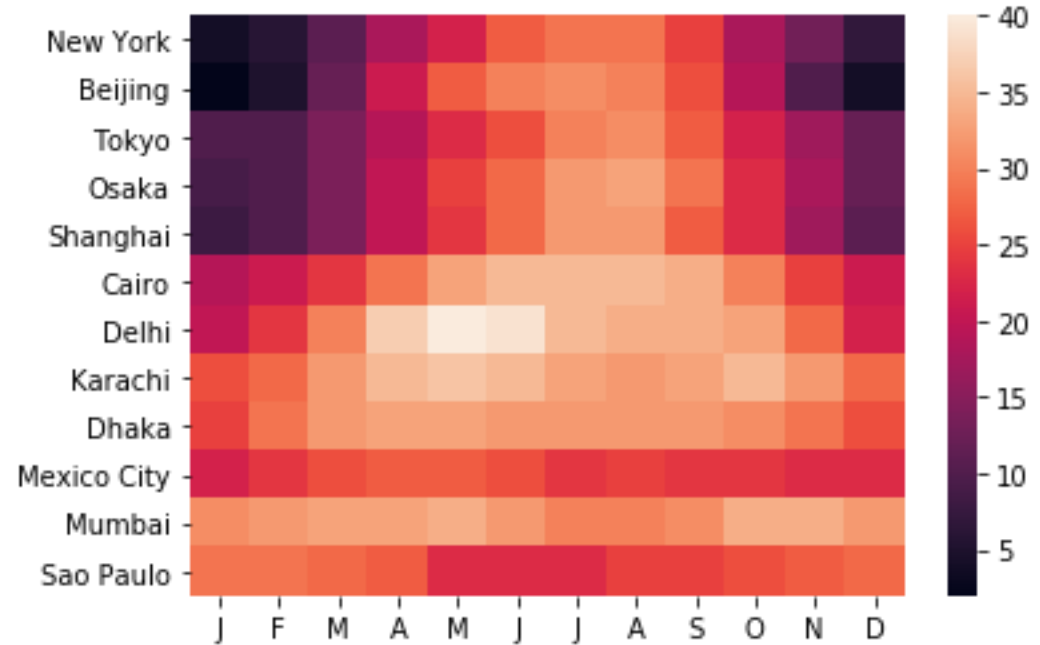


Visualisasi Scatter Plot



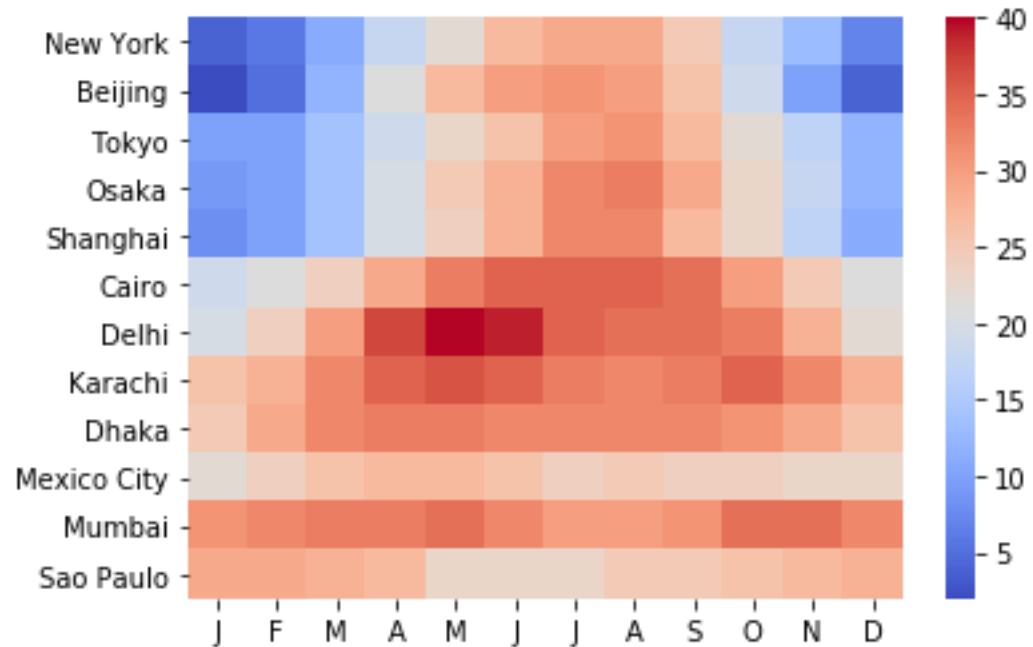
Heatmap

- Heatmap adalah jenis visualisasi yang menggunakan kode warna untuk mewakili nilai / kepadatan relatif data di seluruh permukaan.
- Warna-warna ini kemudian dapat digunakan untuk memeriksa data secara visual guna menemukan kelompok dengan nilai serupa dan mendeteksi trend dalam data.



Heatmap

- 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



Heatmap (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import library

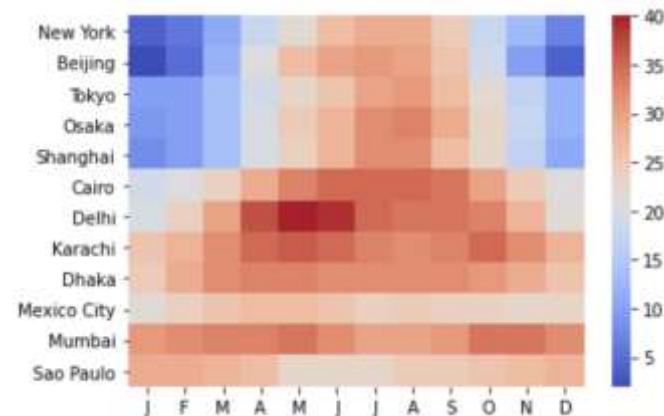
```
cities = ['New York', 'Beijing', 'Tokyo', 'Osaka', 'Shanghai', 'Cairo', 'Delhi',
          'Karachi', 'Dhaka', 'Mexico City', 'Mumbai', 'Sao Paulo']

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',
)
```

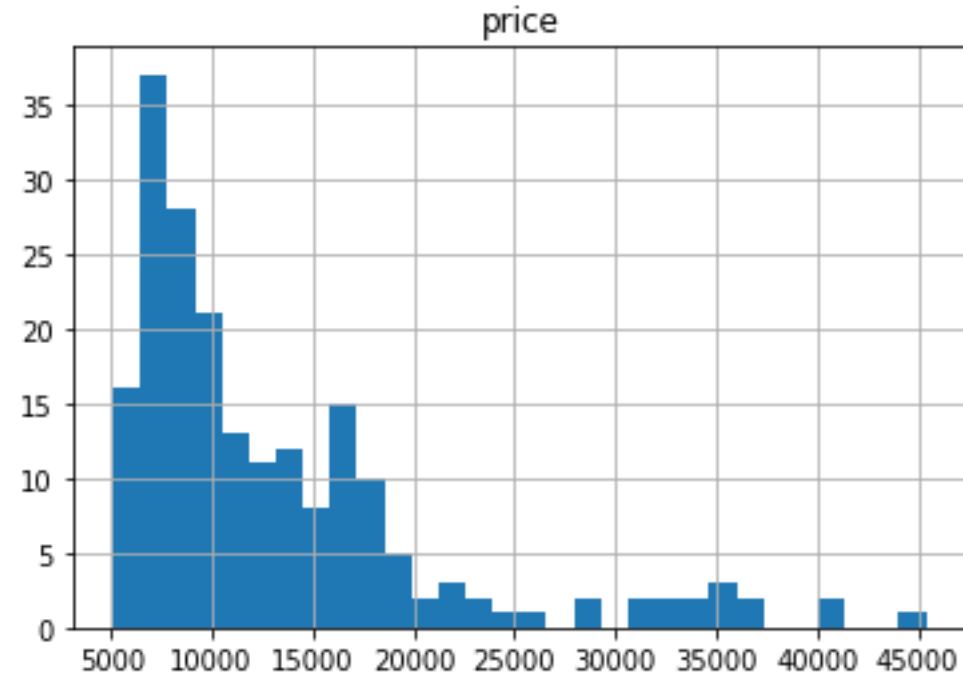
Data

Visualisasi heatmap dengan colormap “coolwarm”



Histogram

- **Histogram** adalah salah satu visualisasi yang cukup penting dalam memahami distribusi pada data kita. Pandas Histogram menyediakan method yang memudahkan kita untuk membuat histogram.
- Plot histogram secara tradisional hanya membutuhkan satu dimensi data.
- Ini dimaksudkan untuk menunjukkan jumlah nilai atau kumpulan nilai secara serial.



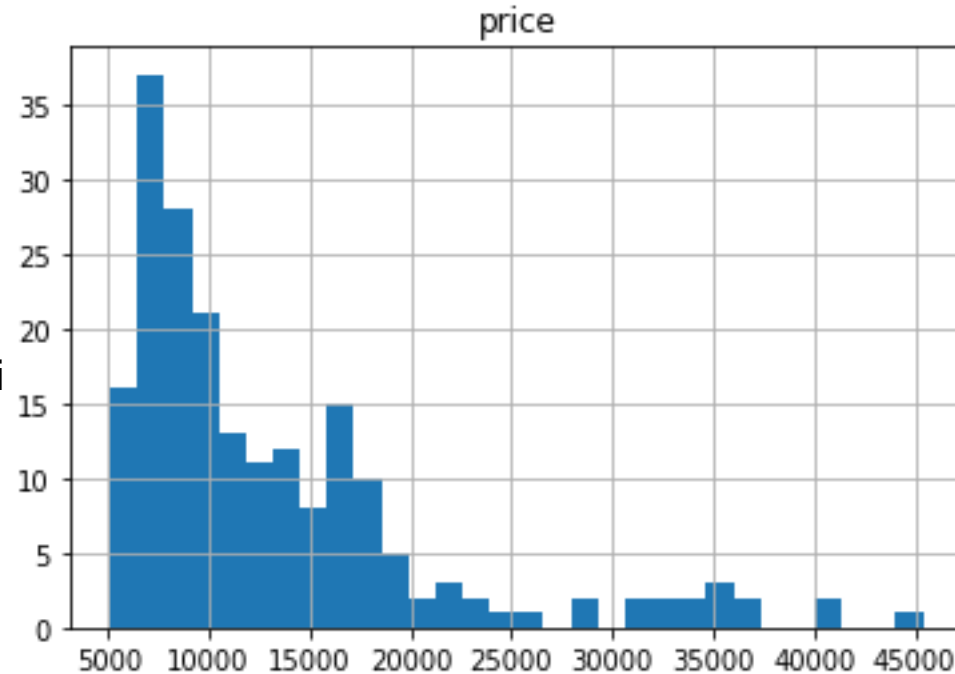
Histogram

- Data yang digunakan adalah data spesifikasi mobil dari berbagai merk

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight	engine-type	num-of-cylinders	engine-size	fuel-system	bore	stroke	compression-ratio
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	0.890278	48.8	2548	dohc	four	130	mpfi	3.47	2.68	
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	0.890278	48.8	2548	dohc	four	130	mpfi	3.47	2.68	
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	0.909722	52.4	2823	ohcv	six	152	mpfi	2.68	3.47	
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	0.919444	54.3	2337	ohc	four	109	mpfi	3.19	3.40	
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	0.922222	54.3	2824	ohc	five	136	mpfi	3.19	3.40	

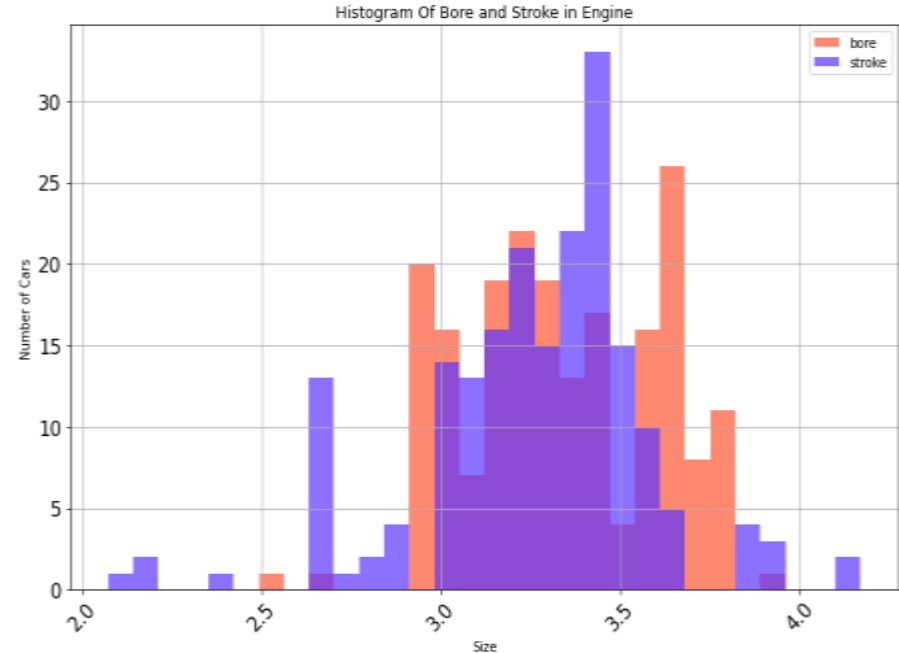
Histogram

- `Pandas DataFrame.hist()` akan mengambil `DataFrame` kita dan menampilkan plot histogram yang menunjukkan distribusi nilai dalam satu seri.
- Untuk membuat histogram di panda, yang perlu kita lakukan adalah memberi tahu panda kolom mana yang ingin kita berikan datanya. Dalam hal ini, saya akan memberi tahu panda bahwa saya ingin melihat distribusi harga (histogram).



Histogram

- Kita juga dapat memplot beberapa grup secara berdampingan. Di sini saya ingin melihat dua histogram, histogram price akan dikelompokkan berdasarkan roda penggerak dari kendaraan (fwd – berpengerak roda depan, 4wd – berpengerak 4 roda, atau rwd – pengerak belakang).



Histogram (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

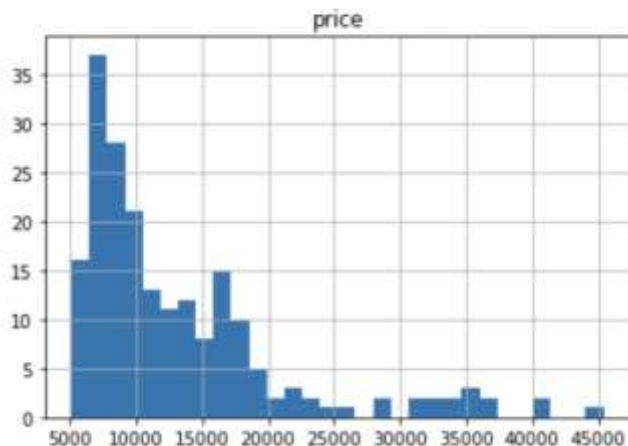


Import library

```
df.hist(column='price', bins=30);
```



Visualisasi histogram
dari kolom "Price"



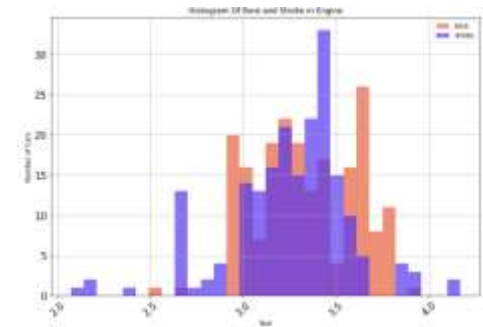
Histogram (Hands-On)

```
# import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import library

```
df[['bore', 'stroke']].plot(kind='hist',
                             alpha=0.7,
                             bins=30,
                             title='Histogram Of Bore and Stroke in Engine',
                             rot=45,
                             grid=True,
                             figsize=(12,8),
                             fontsize=15,
                             color=['#FF5733', '#5C33FF'])
plt.xlabel('Size')
plt.ylabel("Number of Cars");
```

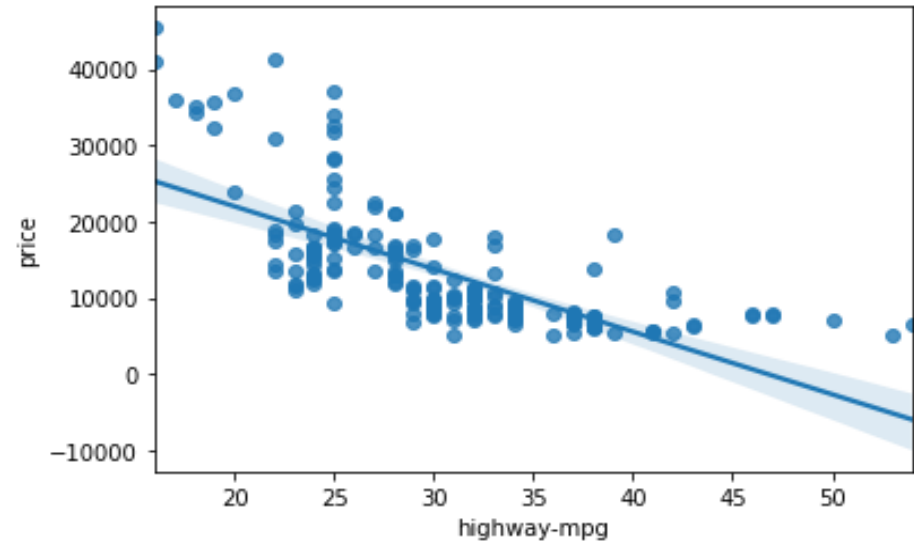
Nama label



Visualisasi
histogram data
"bore" dan "stroke"

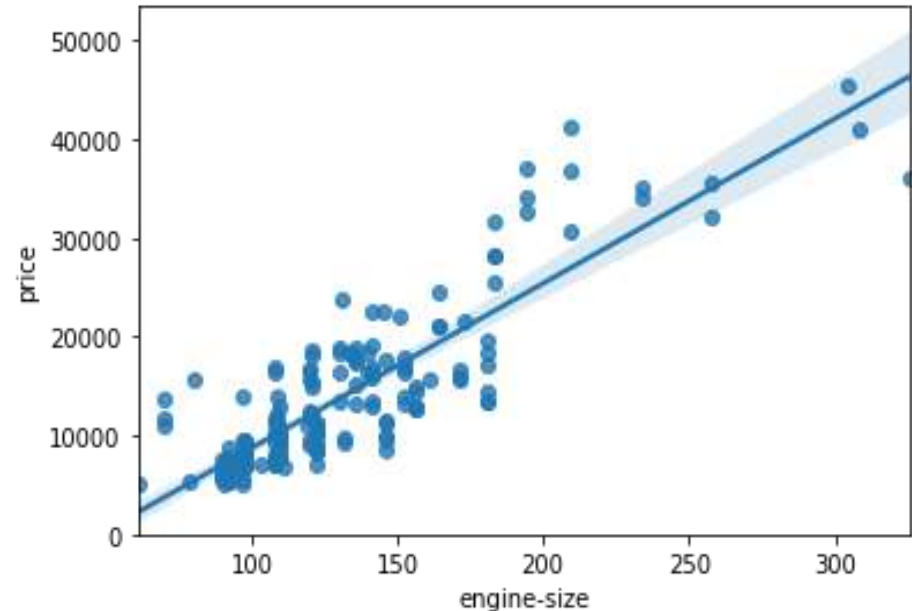
Correlation & Causation

- **Korelasi** merupakan suatu pengukuran sejauh mana nilai saling ketergantungan antar variabel.
- **Causation** merupakan hubungan antara sebab dan akibat antara dua variable
- Penting untuk mengetahui perbedaan antara keduanya dan bahwa korelasi tidak mendeskripsikan sebab-akibat.
- Menentukan korelasi jauh lebih sederhana menentukan sebab memerlukan analisis lebih lanjut



Correlation & Causation

- **Pearson Correlation** adalah metode default dari fungsi "corr". Kita dapat menghitung Korelasi Pearson dari variabel 'int64' atau 'float64'. Terkadang kita ingin mengetahui signifikansi dari estimasi korelasi, kita dapat menggunakan p-value.
- **Korelasi Pearson** mengukur ketergantungan linier antara dua variabel X dan Y.



Correlation & Causation

- **P-Value:**

- Berapa nilai P ini? Nilai P adalah nilai probabilitas bahwa korelasi antara kedua variabel ini signifikan secara statistik. Biasanya, kita memilih tingkat signifikansi 0,05, yang berarti bahwa kami yakin bahwa 95% korelasi antar variabel signifikan.

- **Dengan konvensi, ketika**

- nilai p adalah $p < 0,001$: kami katakan ada bukti kuat bahwa korelasinya signifikan.
- nilai p adalah $p < 0,05$: terdapat bukti moderat bahwa korelasi tersebut signifikan.
- nilai p adalah $p < 0,1$: ada bukti lemah bahwa korelasinya signifikan.
- nilai p adalah $p > 0,1$: tidak ada bukti bahwa korelasi tersebut signifikan.

Correlation & Causation

- Mari kita hitung Koefisien Korelasi Pearson dan nilai-P dari 'wheel-base' dan 'price'.

```
pearson_coef, p_value = stats.pearsonr(df['horsepower'], df['price'])  
print("The Pearson Correlation Coefficient is", pearson_coef, " with a P-value of P = ", p_value)
```

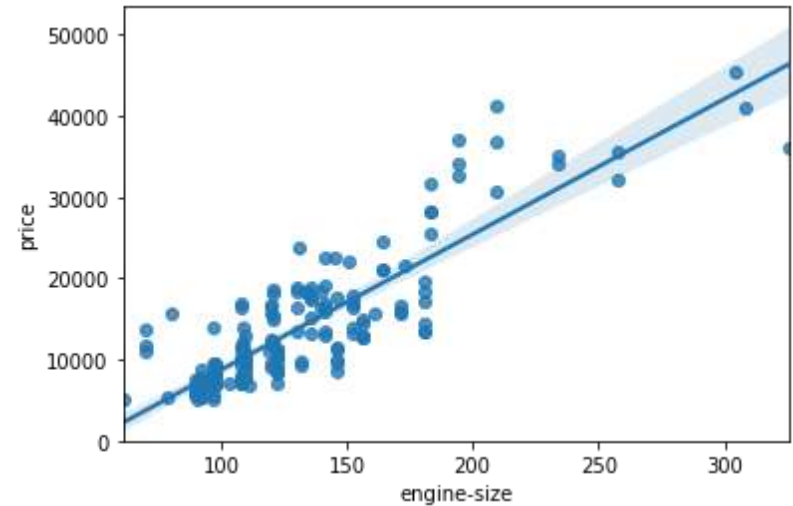
The Pearson Correlation Coefficient is 0.8095745670036559 with a P-value of P = 6.369057428260101e-48

- Karena nilai p adalah $< 0,001$, korelasi antara horsepower dan harga signifikan secara statistik, dengan korelasi linear positif yang cukup kuat ($\sim 0,805$)
- Saat memvisualisasikan variabel individual, penting untuk terlebih dahulu memahami jenis variabel apa yang Anda hadapi. Ini akan membantu kita menemukan metode visualisasi yang tepat untuk variabel tersebut.

Correlation & Causation

- Untuk mulai memahami keterhubungan (linier) antara variabel individu dan harga. Kita dapat melakukan ini dengan menggunakan "regplot".
- Fungsi ini yang memplot scatterplot ditambah garis regresi yang sesuai untuk data.
- Gambar di samping ini memperlihatkan hubungan korelasi positif kuat antara variable.
- Kita dapat memeriksa korelasi antara engine-size dan harga sekitar 0,87

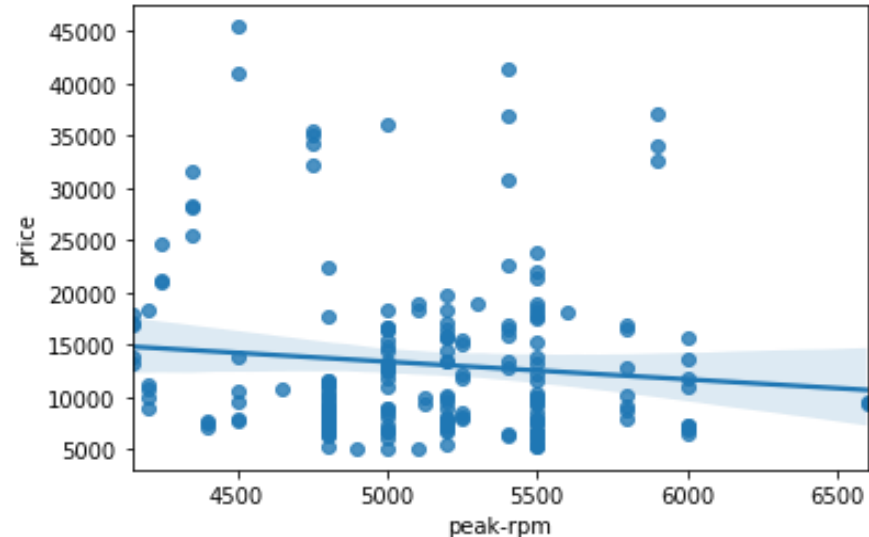
	engine-size	price
engine-size	1.000000	0.872335
price	0.872335	1.000000



- Saat kapasitas mesin naik, harga mobil tersebut juga tinggi: ini menunjukkan hubungan linier antara kedua variabel ini. Ukuran mesin berpotensi menjadi prediktor harga.

Correlation & Causation

- Peak rpm sepertinya bukan merupakan prediktor harga yang baik karena garis regresinya mendekati horizontal.
- Juga, titik-titik data sangat tersebar dan jauh dari garis pas, menunjukkan banyak variabilitas.
- Oleh karena itu itu bukan variabel yang dapat diandalkan untuk memprediksi harga.
- Kita dapat memeriksa korelasi antara 'puncak-rpm' dan 'harga' dan melihatnya kira-kira -0,101616



	peak-rpm	price
peak-rpm	1.000000	-0.101616
price	-0.101616	1.000000

Correlation & Causation (Hands-On)

```
df[["engine-size", "price"]].corr()
```

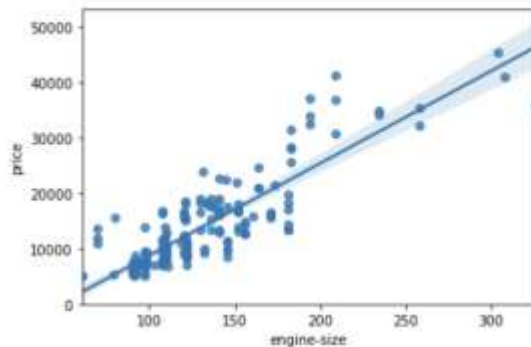
	engine-size	price
engine-size	1.000000	0.872335
price	0.872335	1.000000



Korelasi “engine-size” dan “price”

```
sns.regplot(x="engine-size", y="price", data=df)  
plt.ylim(0,)
```

(0.0, 53209.639179329926)



Visualisasi regplot
“engine-size” dan “price”

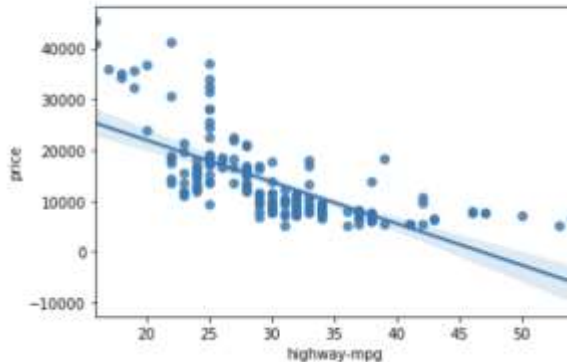
Correlation & Causation (Hands-On)

```
df[['highway-mpg', 'price']].corr()
```

	highway-mpg	price
highway-mpg	1.000000	-0.704692
price	-0.704692	1.000000

Korelasi “highway-mpg”
dan “price”

```
sns.regplot(x="highway-mpg", y="price", data=df)  
<AxesSubplot:xlabel='highway-mpg', ylabel='price'>
```



Visualisasi regplot
“highway-mpg” dan “price”

Correlation & Causation (Hands-On)

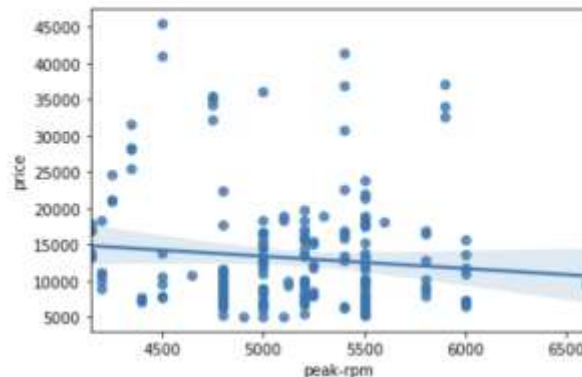
```
df[['peak-rpm', 'price']].corr()
```

	peak-rpm	price
peak-rpm	1.000000	-0.101616
price	-0.101616	1.000000



Korelasi “peak-rpm” dan
“price”

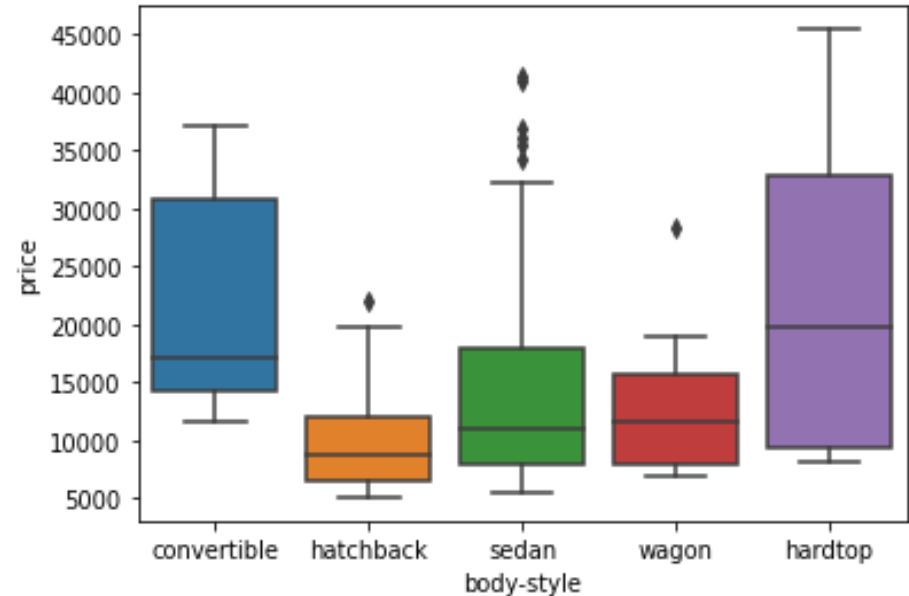
```
sns.regplot(x="peak-rpm", y="price", data=df)  
<AxesSubplot:xlabel='peak-rpm', ylabel='price'>
```



Visualisasi regplot “peak-
rpm” dan “price”

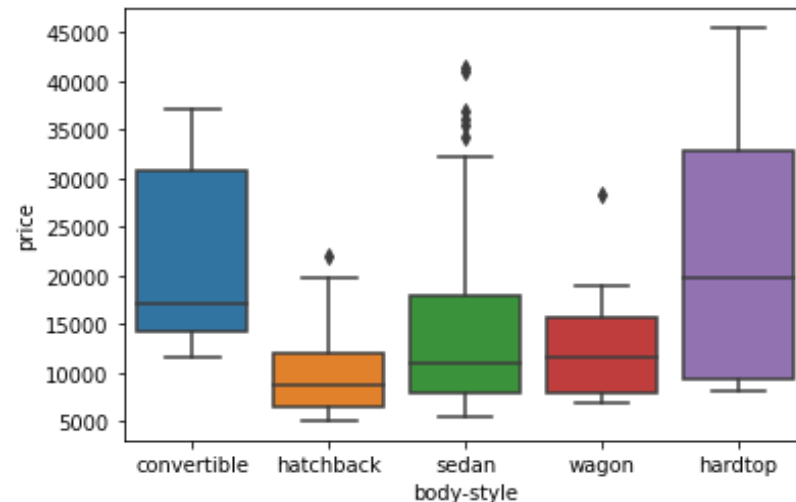
Variabel Kategori Statistik

- Ini adalah variabel yang menggambarkan 'karakteristik' dari unit data, dan dipilih dari sekelompok kategori. Variabel kategori dapat memiliki tipe "objek" atau "int64". Cara yang baik untuk memvisualisasikan variabel kategori adalah dengan menggunakan boxplot.
- Boxplot menggambarkan variable statistic seperti quartil 1, median / quartil 2, quartil 3, nilai maksimum, nilai minimum, dan outlier.



Descriptive Statistic

- Fungsi deskripsi secara otomatis menghitung statistik dasar untuk semua variabel kontinu.
- Analisis yang bisa kita dapatkan dari deskriptif statistik adalah
 - Jumlah variabel
 - Rata-rata
 - Standard deviasi
 - Nilai minimal
 - IQR (Interquartile Range: 25%, 50% and 75%)
 - Nilai Maximal



	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak
count	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	197.000000	201.000000	201.000000	201.000000
mean	0.840794	122.000000	98.797015	0.837102	0.918126	53.788667	2555.688667	126.879627	3.330992	3.258934	10.164278	162.403534	5117.64
std	1.254862	31.996325	6.366366	0.051213	0.029187	2.447822	517.256727	41.546834	0.268072	0.319356	4.004965	37.365780	476.11
min	2.000000	65.000000	86.000000	0.679239	0.837500	47.800000	1488.000000	61.000000	2.540000	2.670000	7.000000	48.000000	4150.00
25%	0.000000	101.000000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000	2.150000	3.170000	8.600000	70.000000	4800.00
50%	1.000000	122.000000	97.000000	0.832292	0.909722	54.100000	2414.000000	120.000000	3.100000	3.240000	9.000000	95.000000	5125.36
75%	2.000000	137.000000	102.400000	0.881788	0.920000	55.500000	2926.000000	141.000000	3.580000	3.410000	9.400000	116.000000	5500.00
max	3.000000	256.000000	120.900000	1.000000	1.000000	59.800000	4064.000000	326.000000	3.940000	4.170000	23.000000	262.000000	6600.00

Descriptive Statistic (Hands-On)

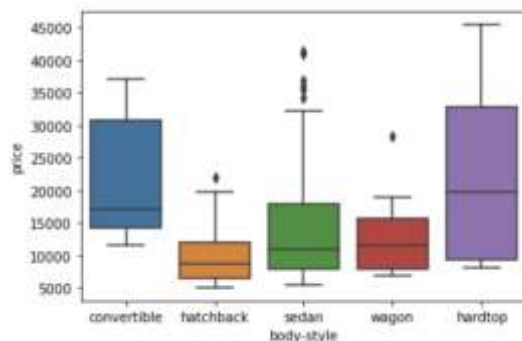
```
df.describe()
```

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower
count	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	197.000000	201.000000	201.000000
mean	0.840796	122.000000	98.797015	0.837102	0.915126	53.766667	2555.666667	126.875622	3.330892	3.258904	10.164279	103.405534
std	1.254802	31.996525	8.066366	0.059213	0.029187	2.447822	517.295727	41.546834	0.268072	0.318256	4.004965	37.365700
min	-2.000000	65.000000	86.600000	0.678039	0.837500	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000
25%	0.000000	101.000000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000	3.150000	3.110000	8.600000	70.000000
50%	1.000000	122.000000	97.000000	0.832292	0.906722	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000
75%	2.000000	137.000000	102.400000	0.881788	0.925000	55.500000	2926.000000	141.000000	3.580000	3.410000	9.400000	116.000000
max	3.000000	256.000000	120.900000	1.000000	1.000000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	262.000000

Menghitung statistik dasar

```
sns.boxplot(x="body-style", y="price", data=df)
```

```
<AxesSubplot:xlabel='body-style', ylabel='price'>
```



Visualisasi boxplot “body-style” dan “price”

Grouping

- Method "**groupby**" digunakan untuk mengelompokkan data menurut kategori yang berbeda. Data dikelompokkan berdasarkan satu atau beberapa variabel dan analisis dilakukan pada kelompok individu.
- Sebagai contoh, mari kita kelompokkan berdasarkan variabel "drive-wheels". Kita melihat bahwa ada 3 kategori roda penggerak yang berbeda.

```
df['drive-wheels'].unique()
```

```
array(['rwd', 'fwd', '4wd'], dtype=object)
```

Grouping

- Anda juga dapat mengelompokkan dengan beberapa variabel. Misalnya, mari kita kelompokkan berdasarkan 'drive-wheels' dan 'body-style'.
- **Grouping** mengelompokkan dataframe dengan kombinasi unik 'drive-wheels' dan 'body-style'. Kita dapat menyimpan hasilnya dalam variabel 'grouped_test1'.

	drive-wheels	body-style	price
0	4wd	hatchback	7603.000000
1	4wd	sedan	12647.333333
2	4wd	wagon	9095.750000
3	fwd	convertible	11595.000000
4	fwd	hardtop	8249.000000
5	fwd	hatchback	8396.387755
6	fwd	sedan	9811.800000
7	fwd	wagon	9997.333333
8	rwd	convertible	23949.600000
9	rwd	hardtop	24202.714286
10	rwd	hatchback	14337.777778
11	rwd	sedan	21711.833333
12	rwd	wagon	16994.222222

body-style	price				
	convertible	hardtop	hatchback	sedan	wagon
drive-wheels					
4wd	0.0	0.000000	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222

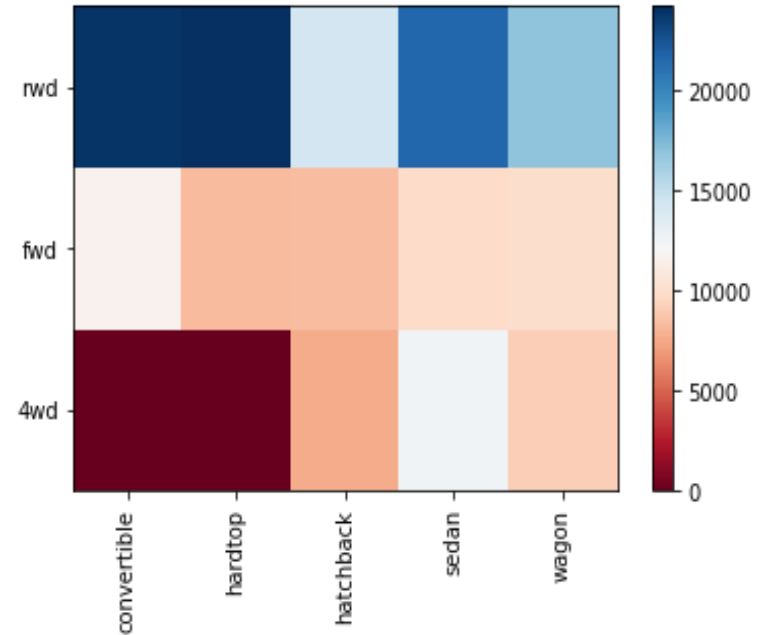
Grouping

- Data yang dikelompokkan ini jauh lebih mudah untuk divisualisasikan ketika dibuat menjadi tabel pivot.
- Tabel pivot yang mirip seperti pada spreadsheet Excel, dengan satu variabel di sepanjang kolom dan variabel lainnya di sepanjang baris.
- Kita dapat mengonversi kerangka data menjadi tabel pivot menggunakan metode "pivot" untuk membuat tabel pivot dari grup.

Grouping

- Dari table pivot kita dapat mengilustrasikan table pivot dalam bentuk heatmap.

body-style	price				
	convertible	hardtop	hatchback	sedan	wagon
drive-wheels					
4wd	0.0	0.000000	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222



Grouping (Hands-On)

```
df['drive-wheels'].unique()  
array(['rwd', 'fwd', '4wd'], dtype=object)
```

Menampilkan nilai unik data series

```
df_group_one = df[['drive-wheels', 'body-style', 'price']]
```

Membuat dataframe baru dengan kolom
'drive-wheels', 'body-style', dan 'price' unik
data series

```
df_gptest = df[['drive-wheels', 'body-style', 'price']]  
grouped_test1 = df_gptest.groupby(['drive-wheels', 'body-style'], as_index=False).mean()  
grouped_test1
```

	drive-wheels	body-style	price
0	4wd	hatchback	7603.000000
1	4wd	sedan	12647.333333
2	4wd	wagon	9095.750000
3	fwd	convertible	11595.000000
4	fwd	hardtop	8249.000000
5	fwd	hatchback	8396.387755
6	fwd	sedan	9811.800000
7	fwd	wagon	9997.333333
8	rwd	convertible	23949.600000
9	rwd	hardtop	24202.714286
10	rwd	hatchback	14337.777778
11	rwd	sedan	21711.833333
12	rwd	wagon	16994.222222

Melakukan grouping dari “drives-
wheels” dan “body-style”
berdasarkan rata-rata “harga”

Grouping (Hands-On)

```
grouped_pivot = grouped_test1.pivot(index='drive-wheels', columns='body-style')  
grouped_pivot
```

drive-wheels	price				
	convertible	hardtop	hatchback	sedan	wagon
4wd	NaN	NaN	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222



Melakukan pivot data

```
grouped_pivot = grouped_pivot.fillna(0)  
grouped_pivot
```

drive-wheels	price				
	convertible	hardtop	hatchback	sedan	wagon
4wd	0.0	0.000000	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222



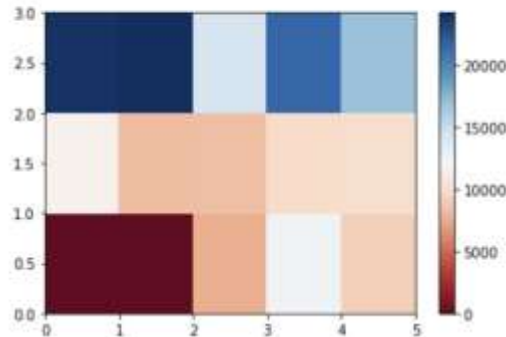
melakukan pivot data
dengan handle missing
value nilai 0

Grouping (Hands-On)

```
df_gpctest2 = df[['body-style', 'price']]  
grouped_test_bodystyle = df_gpctest2.groupby(['body-style'], as_index=False).mean()  
grouped_test_bodystyle
```

	body-style	price
0	convertible	21890.500000
1	hardtop	22208.500000
2	hatchback	9857.441176
3	sedan	14459.755319
4	wagon	12371.960000

```
plt.pcolor(grouped_pivot, cmap='RdBu')  
plt.colorbar()  
plt.show()
```



Melakukan grouping dari
"body-style" berdasarkan
rata-rata "harga"

Visualisasi heatmap

ANOVA

- **Analysis of Varians (ANOVA)** adalah metode statistik yang digunakan untuk menguji apakah ada perbedaan yang signifikan antara rata-rata dua kelompok atau lebih.
- **ANOVA** mengembalikan dua parameter
 - F-Score:
 - P-Value
- **F-Score: ANOVA** mengasumsikan rata-rata semua kelompok adalah sama, anova akan menghitung seberapa jauh rata-rata yang sebenarnya menyimpang dari asumsi, dan melaporkannya sebagai F-Score.
- Skor yang lebih besar berarti ada perbedaan yang lebih besar antara rata-rata.
- P-Value: Nilai-P menunjukkan seberapa signifikan secara statistik nilai skor yang dihitung.

ANOVA

- Jika variabel harga pada dataset mobil sangat berkorelasi dengan variabel lainya, **ANOVA** akan mengembalikan skor F-Score yang cukup besar dan nilai-p yang kecil.
- ANOVA menganalisis perbedaan antara kelompok yang berbeda dari variabel yang sama, fungsi groupby akan berguna dalam kasus ANOVA.
- Mari kita lihat apakah jenis 'roda penggerak' mempengaruhi 'harga'

```
# grouping results
df_gptest = df[['drive-wheels', 'body-style', 'price']]
grouped_test1 = df_gptest.groupby(['drive-
wheels', 'body-style'], as_index=False).mean()
grouped_test1
```

	drive-wheels	body-style	price
0	4wd	hatchback	7603.000000
1	4wd	sedan	12647.333333
2	4wd	wagon	9095.750000
3	fwd	convertible	11595.000000
4	fwd	hardtop	8249.000000
5	fwd	hatchback	8396.387755
6	fwd	sedan	9811.800000
7	fwd	wagon	9997.333333
8	rwd	convertible	23949.600000
9	rwd	hardtop	24202.714286
10	rwd	hatchback	14337.777778
11	rwd	sedan	21711.833333
12	rwd	wagon	16994.222222

ANOVA

```
grouped_test2.get_group('rwd')['price']
```

0	13495.0
1	16500.0
2	16500.0
9	16430.0
10	16925.0
...	
196	16845.0
197	19045.0
198	21485.0
199	22470.0
200	22625.0

Name: price, Length: 75, dtype: float64

```
grouped_test2.get_group('4wd')['price']
```

4	17450.0
136	7603.0
140	9233.0
141	11259.0
144	8013.0
145	11694.0
150	7898.0
151	8778.0

Name: price, dtype: float64

```
[9] grouped_test2.get_group('fwd')['price']
```

3	13950.0
5	15250.0
6	17710.0
7	18920.0
8	23875.0
...	
185	11595.0
186	9980.0
187	13295.0
188	13845.0
189	12290.0

Name: price, Length: 118, dtype: float64

```
# ANOVA
```

```
f_val, p_val = stats.f_oneway(grouped_test2.get_group('fwd')['price'], grouped_test2.  
get_group('rwd')['price'], grouped_test2.get_group('4wd')['price'])  
print("ANOVA results: F=", f_val, ", P =", p_val)
```

ANOVA results: F= 67.95406500780399 , P = 3.3945443577151245e-23

- Hasil ANOVA ini termasuk hasil yang bagus, dengan F-Score yang besar menunjukkan korelasi yang kuat dan nilai P hampir 0 menyiratkan signifikansi statistik yang hampir pasti.
- Tetapi apakah ini berarti ketiga kelompok yang diuji semuanya berkorelasi tinggi?



Quiz / Tugas

Quiz dapat diakses melalui <https://spadadikti.id/>



Terima kasih