

# Learning Progress Review Basic Statistics

1 - 6 November 2025



TIM SERU

# Table of contents

**01** Intro to Statistics

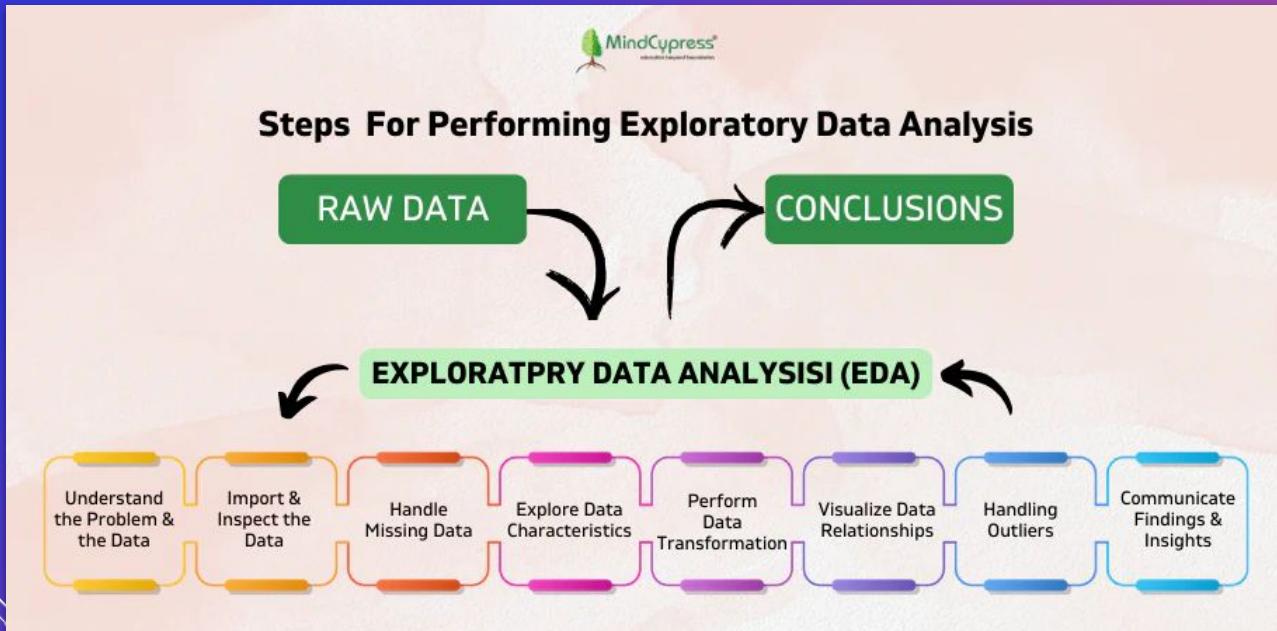
**02** Statistic Data Type

**03** Correlation & Causation

**04** Probability &  
Distributions



# Exploratory Data Analysis



Source:

<https://mindcypress.com/blogs/finance-accounting/understanding-the-primary-goal-of-exploratory-data-analysis>

# Wrangler Project

Live Session

# Wrangler Project

## Pengertian

Wrangler Project merupakan sebuah proyek yang berfokus pada proses pembersihan (cleaning), transformasi (transformation), dan persiapan (preparation) data dari format mentah (raw data) menjadi format yang siap untuk dianalisis.

## DataSets

Dataset yang digunakan adalah dataset gaji Data Scientist (Data Scientist Salaries Dataset), yang berisi informasi mengenai tahun kerja, level pengalaman, tipe karyawan, job title, gaji, mata uang, gaji dalam USD, lokasi karyawan, rasio remote, lokasi perusahaan dan ukuran perusahaan. Dengan bentuk datasets adalah 3755 baris dan 11 kolom.

## Langkah Utama

- Memahami arti setiap kolom (feature understanding)  
Kita harus memahami makna dan konteks dari setiap kolom dalam dataset agar analisis yang dilakukan relevan dan akurat.
- Mendeteksi anomali (data anomaly detection)  
Melihat adanya nilai-nilai yang tidak wajar, inkonsistensi, atau kesalahan input yang mungkin muncul di dalam dataset.
- Verifikasi data (data verification)  
Mengecek validitas data dengan memastikan bahwa data sesuai dengan konteks dan tidak terjadi kesalahan input atau duplikasi yang tidak diinginkan.
- Menangani outlier (outlier handling)  
Dataset sering kali mengandung data outlier, yaitu nilai-nilai ekstrem yang berbeda jauh dari sebagian besar data lainnya. Dalam konteks analisis, kita dapat memilih untuk mempertahankan atau menghapus outlier tergantung pada tujuan analisis dan dampaknya terhadap hasil.
- Menggunakan .copy()  
Selama proses eksplorasi dan transformasi, kita disarankan untuk menggunakan perintah .copy() agar dapat melakukan berbagai variasi percobaan tanpa mengubah dataset utama.

# Wrangler Project

## Exploratory Data Analysis

Check Null Values → Mencari tahu apakah ada nilai yang hilang.

```
[15] ✓ 0s df.isnull().head()
> Show hidden output

[16] ✓ 0s df.isnull().sum()
> Show hidden output

[17] ✓ 0s
  ⏎ # Menampilkan jumlah nilai yang hilang per kolom
  missing_values = df.isnull().sum()

  missing_values_df = pd.DataFrame(
      data=missing_values,
      columns=['missing_values_count'],
  )

  missing_values_df
> ...
  ... Show hidden output
Next steps: Generate code with missing_values_df | New interactive sheet

[18] ✓ 1s
  ⏎ missing_values_df.plot(kind='bar');
> Show hidden output

[19] ✓ 0s
  ⏎ df.info()
> ...
  ... Show hidden output
```

# Wrangler Project

## Exploratory Data Analysis

Check Duplicated Data → Mencari tahu apakah ada nilai duplikat.

```
[20] ✓ 0s
    dm = pd.DataFrame(
        data={
            'col1': [1, 2, 3],
            'col2': [4, 5, 6]
        }
    )

    dm

> ... Show hidden output
Next steps: Generate code with dm New interactive sheet

[21] ✓ 0s
    dm1 = dm

[22] ✓ 0s
    dm1['col1'] = dm1['col1'] + 1

[23] ✓ 0s
    dm

    col1  col2
    0      2      4
    1      3      5
    2      4      6

> ... Show hidden output
Next steps: Generate code with dm New interactive sheet

[24] ✓ 0s
    # Salin data sebelum transformasi
    df_copy = df.copy()
```

```
[25] ✓ 0s
    df.duplicated()
    Show hidden output

[26] ✓ 0s
    df.duplicated().sum()
    np.int64(1171)

[27] ✓ 0s
    df.duplicated()
    Show hidden output

[28] ✓ 0s
    duplicated_row_mask = df.duplicated()
    df[duplicated_row_mask]
    Show hidden output

[29] ✓ 1s
    duplicated_row_mask = df.duplicated(keep=False)
    df[duplicated_row_mask]
    ... Show hidden output

[30] ✓ 0s
    df.iloc[114:116, :]
    Show hidden output

> 1.2.2 Delete Duplicates
    ↳ 1 cell hidden
```

# Wrangler Project

## Exploratory Data Analysis

Check Unique Values in Each Column → Mencari tahu nilai unik dan jumlah nilai unik setiap kolom.

- ✓ 1.3.1 Check Data Shape
- [32] ✓ 0s df\_copy.shape  
  (2584, 11)
- ✓ 1.3.2 Check Data Type
- [33] ✓ 0s df\_copy.dtypes  
  ... Show hidden output
- [34] ✓ 0s df\_copy.info()  
  ... Show hidden output
- ✓ 1.3.3 Check Data Stats
- [35] ✓ 0s df\_copy.describe()  
  ... Show hidden output

- ✓ 1.3.4 Check Unique Values
- [36] ✓ 0s 

```
# for col in df_copy.columns:  
#     if col == 'salary' or col == 'salary_in_usd':  
#         continue  
#     else:  
#         print(f'{col}: {df_copy[col].nunique()}')  
#         print(f'{df_copy[col].unique()}')  
#         print('*'*20)
```
- [37] ✓ 0s 

```
excepted_columns = [  
    'salary',  
    'salary_in_usd'  
]  
  
for col in df_copy.columns:  
    if col not in excepted_columns:  
        print(f'{col}: {df_copy[col].nunique()}')  
        print(f'{df_copy[col].unique()}')  
        print('*'*20)  
    print("")
```

  
... Show hidden output
- ✓ 1.3.5 Check Columns
- [38] ✓ 0s df.columns  
  ... Show hidden output

# Wrangler Project

## Exploratory Data Analysis

Rename Value for Better Understanding → Melakukan perubahan nilai untuk memudahkan dalam memahami data.

```
[39] ✓ 0s
dict_exp_level = {
    'BN': 'Junior',
    'MI': 'Mid',
    'SE': 'Senior',
    'EX': 'Executive'
}

dict_employment_type = {
    'FT': 'Full Time',
    'PT': 'Part Time',
    'CT': 'Contract',
    'FL': 'Freelance'
}

dict_remote_ratio = {
    0: 'No Remote',
    50: 'Partially Remote',
    100: 'Fully Remote'
}

[40] ✓ 0s
1.4.1 Replace Values
df_1 = df_copy.copy()

[41] ✓ 0s
df_1['experience_level'] = (
    df_1['experience_level']
    .replace(dict_exp_level)
)

df_1['employment_type'] = (
    df_1['employment_type']
    .replace(dict_employment_type)
)

df_1['remote_ratio'] = (
    df_1['remote_ratio']
    .replace(dict_remote_ratio)
)

[42] ✓ 0s
df_1.head(10)
Show hidden output
Next steps: Generate code with df\_1 New interactive sheet

[43] ✓ 0s
df_2 = df_copy.copy()

[44] ✓ 0s
df_2.insert(
    loc=list(df_2.columns).index('experience_level') + 1,
    column="experience_level_desc",
    value=df_2['experience_level'].map(dict_exp_level)
)

[45] ✓ 0s
df_2.head()
> Show hidden output
Next steps: Generate code with df\_2 New interactive sheet

[46] ✓ 0s
df_2.insert(
    loc=list(df_2.columns).index('employment_type') + 1,
    column="employment_type_desc",
    value=df_2['employment_type'].map(dict_employment_type)
)

[47] ✓ 0s
df_2.head()
> Show hidden output
Next steps: Generate code with df\_2 New interactive sheet

[48] ✓ 0s
df_2.insert(
    loc=list(df_2.columns).index('remote_ratio') + 1,
    column="remote_ratio_desc",
    value=df_2['remote_ratio'].map(dict_remote_ratio)
)

[49] ✓ 0s
df_2.head()
> ... Show hidden output
Next steps: Generate code with df\_2 New interactive sheet
```

# Wrangler Project

## Exploratory Data Analysis

Group Job Title → Melakukan pengelompokkan data untuk memudahkan dalam menganalisis.

```
Check Job Title
[50] df_2['job_title'].unique()
✓ On
> Show hidden output

Grouping Function
[51] # def assign_broader_category(job_title):
    #     data_engineering = ["Data Engineer", "Data Analyst", "Analytics Engineer", "BI Data Analyst", "Business Data Analyst", "BI Developer", "BI Analyst", "Business Intelligence Engineer", "BI Data Scientist", "Applied Scientist", "Research Scientist", "3D Computer Vision Researcher", "Deep Learning Researcher", "Business Intelligence Engineer"]
    #     machine_learning = ["Machine Learning Engineer", "ML Engineer", "Lead Machine Learning Engineer", "Principal Machine Learning Engineer"]
    #     data_architecture = ["Data Architect", "Big Data Architect", "Cloud Data Architect", "Principal Data Architect"]
    #     management = ["Data Science Manager", "Director of Data Science", "Head of Data Science", "Data Scientist Lead", "Head of Machine Learning", "Manager Data Management", "Data Analytics Manager"]
    #
    #     if job_title in data_engineering:
    #         return "Data Engineering"
    #     elif job_title in data_scientist:
    #         return "Data Science"
    #     elif job_title in machine_learning:
    #         return "Machine Learning"
    #     elif job_title in data_architecture:
    #         return "Data Architecture"
    #     elif job_title in management:
    #         return "Management"
    #     else:
    #         return "Other"

category_mapping = {
    "Data Engineering": [
        "Data Engineer",
        "Data Analyst",
        "Analytics Engineer",
        "BI Data Analyst",
        "Business Data Analyst",
        "BI Developer",
        "BI Analyst",
        "Business Intelligence Engineer",
        "BI Data Engineer",
        "Power BI Developer",
        "Machine Learning Engineer",
        "ML Engineer",
        "Lead Machine Learning Engineer",
        "Principal Machine Learning Engineer",
        "Data Architect",
        "Big Data Architect",
        "Cloud Data Architect",
        "Principal Data Architect",
        "Data Science Manager",
        "Director of Data Science",
        "Head of Data Science",
        "Data Scientist Lead",
        "Head of Machine Learning",
        "Manager Data Management",
        "Data Analytics Manager"
    ]
}
```

# Wrangler Project

## Exploratory Data Analysis

Group Job Title → Melakukan pengelompokkan data untuk memudahkan dalam menganalisis.

```
    "Data Scientist": [
        "Data Scientist",
        "Applied Scientist",
        "Research Scientist",
        "3D Computer Vision Researcher",
        "Deep Learning Researcher",
        "AI/Computer Vision Engineer",
    ],
    "Machine Learning": [
        "Machine Learning Engineer",
        "ML Engineer",
        "Lead Machine Learning Engineer",
        "Principal Machine Learning Engineer",
    ],
    "Data Architecture": [
        "Data Architect",
        "Big Data Architect",
        "Cloud Data Architect",
        "Principal Data Architect",
    ],
    "Management": [
        "Data Science Manager",
        "Director of Data Science",
        "Head of Data Science",
        "Data Science Lead",
        "Head of Machine Learning",
        "Manager Data Management",
        "Data Analytics Manager",
    ],
}

def assign_broader_category(job_title, mapper):
    """
    Assigns a broader category to a given job title.

    Args:
        job_title (str): The job title to categorize.

    Returns:
        str: The broader category the job title belongs to, or "Other" if no
            match is found.
    """
    for category, titles in category_mapping.items():
        if job_title in titles:
            return category
    return "Other"
```

# Wrangler Project

## Exploratory Data Analysis

Group Job Title → Melakukan pengelompokkan data untuk memudahkan dalam menganalisis.

```
[53] ✓ 0s
    df_2.insert(
        loc=list(df_2.columns).index('job_title') + 1,
        column="job_title_simplified",
        value=df_2['job_title'].apply(lambda x: assign_broader_category(x, category_mapping))
    )

[54] ✓ 0s
    df_2.head()
    > Show hidden output
    Next steps: Generate code with df\_2 New interactive sheet

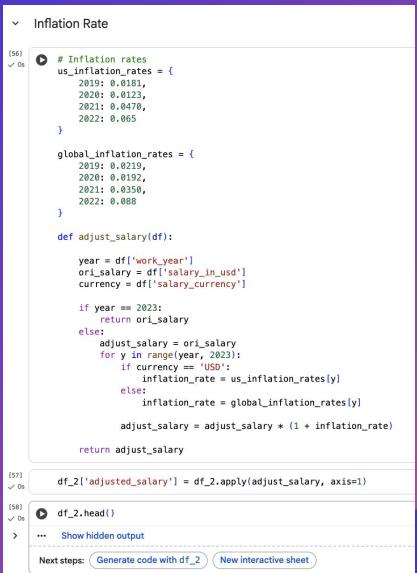
[55] ✓ 0s
    ▶ df_2[df_2['job_title_simplified'] == 'Other']['job_title'].unique()
    > ...
    ... Show hidden output

[55] ✓ 0s
    Start coding or generate with AI.
```

# Wrangler Project

## Exploratory Data Analysis

Adjust Salary to Present Values → Melakukan penyesuaian data dengan nilai sekarang.



```
# Inflation Rate
us_inflation_rates = {
    2019: 0.0181,
    2020: 0.0123,
    2021: 0.0470,
    2022: 0.065
}

global_inflation_rates = {
    2019: 0.0219,
    2020: 0.0192,
    2021: 0.0350,
    2022: 0.088
}

def adjust_salary(df):
    year = df['work_year']
    ori_salary = df['salary_in_usd']
    currency = df['salary_currency']

    if year == 2023:
        return ori_salary
    else:
        adjust_salary = ori_salary
        for y in range(year, 2023):
            if currency == 'USD':
                inflation_rate = us_inflation_rates[y]
            else:
                inflation_rate = global_inflation_rates[y]

            adjust_salary = adjust_salary * (1 + inflation_rate)

        return adjust_salary

df_2['adjusted_salary'] = df_2.apply(adjust_salary, axis=1)

df_2.head()
...
Show hidden output
```

**"Data are just summaries of thousands of stories – tell a few of those stories to help make the data meaningful,"**

Chip and Dan Heath

01

# Intro to Statistics



# Statistik

Statistik adalah ilmu mengumpulkan, mengolah, menganalisis, menafsirkan, menyajikan, mengatur dan mengambil kesimpulan dari data.



# Mengapa Statistik Penting ?

- ▶ 1. Statistik Membantu Pengambilan Keputusan yang Tepat
- ▶ 2. Statistik Mengubah Data Mentah Menjadi Informasi Bermakna
- ▶ 3. Statistik Membantu Mengetahui Pola dan Tren
- ▶ 4. Statistik Diperlukan di Hampir Semua Bidang
- ▶ 5. Statistik Membantu Menghadapi Ketidakpastian
- ▶ 6. Statistik Melatih Cara Berpikir Kritis dan Logis



# Peran Statistik dalam Kehidupan Sehari-hari

Bidang Kehidupan	Peran Statistik
Kehidupan sehari-hari	Membantu membuat keputusan sederhana
Bisnis & Ekonomi	Menganalisis tren dan memprediksi pasar
Kesehatan	Mengevaluasi dan meningkatkan layanan medis
Pendidikan	Menilai prestasi dan efektivitas pembelajaran
Sosial & Politik	Membentuk kebijakan berbasis data
Sains & Teknologi	Menguji teori dan membuat model prediktif



# Contoh Penggunaan Statistik dalam Kehidupan Sehari-hari



# Mengapa Statistik Penting di Bidang Data Science



Statistik adalah fondasi penting dalam Data Science

Termasuk **3 pilar yang harus** dikuasai oleh Data Scientist

Analisis data yang **mendalam**

Hasil analisa menjadi dasar untuk membuat model **prediktif**

Interpretasi hasil yang ilmiah dan terukur

02

# Statistic Data Types



# Tipe Data Statistik

## Kualitatif

- Bersifat non-angka, menggambarkan karakteristik atau kategori.
- Contoh: warna, jenis kelamin, nama, jurusan, merek mobil.

## Kuantitatif

- Bersifat angka dan dapat dihitung atau diukur.
- Terdiri dari dua jenis:
- Diskrit → dapat dihitung secara pasti (contoh: jumlah siswa, kendaraan, buku).
  - Kontinu → dapat diukur dan memiliki nilai di antara dua angka (contoh: tinggi badan, berat badan, suhu, waktu).

# Cabang Statistik

## Statistik Deskriptif

- Berfungsi untuk mendeskripsikan dan memberikan insight dari data yang dikumpulkan.

## Statistik Inferensial

- Bertujuan untuk membuat prediksi atau menarik kesimpulan mengenai populasi berdasarkan data sampel.

# 3 Komponen Statistik Deskriptif

## Measures of Central Tendency (Ukuran Pemusatan)

Mengukur nilai tengah dari dataset dengan nilai representatif seperti mean, median, dan modus.

## Measures of Variability (Ukuran Penyebaran)

Menggambarkan seberapa jauh data tersebar dari nilai tengah, menggunakan range, variance, dan standard deviation.

## Visualisasi Data

Menyajikan data dalam bentuk grafik atau diagram agar lebih mudah dipahami serta membantu mengenali pola, tren, dan outlier.

# Measures of Central Tendency

## Mean (Rata-Rata)

Jumlah seluruh nilai dibagi jumlah data.

Contoh:  $(7 + 5 + 2 + 5 + 4 + 1 + 3) / 7 = 3.86$

## Median (Nilai Tengah)

Nilai di tengah setelah data diurutkan.

Contoh: setelah diurutkan (1, 2, 3, 4, 5, 5, 7), median = 4

## Modus (Nilai Terbanyak)

Nilai yang paling sering muncul.

Contoh: 5 muncul paling sering.

# Measures of Variability

## Range

Selisih antara nilai maksimum dan minimum.

Rumus: Range = Max - Min

Contoh:  $7 - 1 = 6$

## Variance (Ragam)

Rata-rata kuadrat selisih setiap data terhadap mean.

## Standard Deviation (Simpangan Baku)

Akar dari variance, menunjukkan seberapa jauh data menyebar dari rata-rata.

$$\sigma = \sqrt{\frac{\sum(x-\bar{x})^2}{N}}$$

Di mana:

- $\sigma$  = Standar deviasi
- $x$  = Setiap nilai dalam dataset
- $\bar{x}$  = Rerata (mean) dari dataset
- $N$  = Jumlah total nilai dalam dataset

Sample Variance ( $s^2$ )

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$$

$s^2$  = variance

$X_i$  = term in data set

$\bar{x}$  = Sample mean

$\Sigma$  = Sum

$n$  = Sample size

03

# Correlation & Causation



# Correlation vs Causation

## Correlation

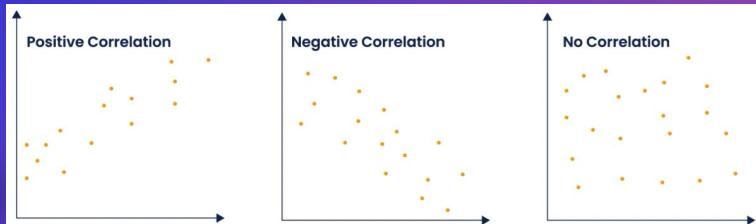
Definisi: Korelasi mengukur kekuatan dan arah hubungan antara dua variabel.

Contoh: Peningkatan penjualan es krim berkorelasi dengan peningkatan jumlah penderita sengatan matahari.

### Pearson correlation coefficient

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

- $r < -0.5$  korelasi negatif kuat
- $-0.5 \leq r \leq 0.5$  korelasi lemah
- $r > 0.5$  korelasi positif kuat

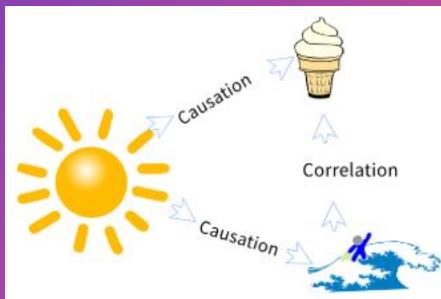


## Causation

Definisi: Kausalitas menunjukkan bahwa suatu peristiwa merupakan akibat dari terjadinya peristiwa lain.

Mekanisme: Ini adalah hubungan sebab akibat antara dua variabel atau lebih, di mana perubahan pada satu variabel menyebabkan perubahan pada variabel lainnya.

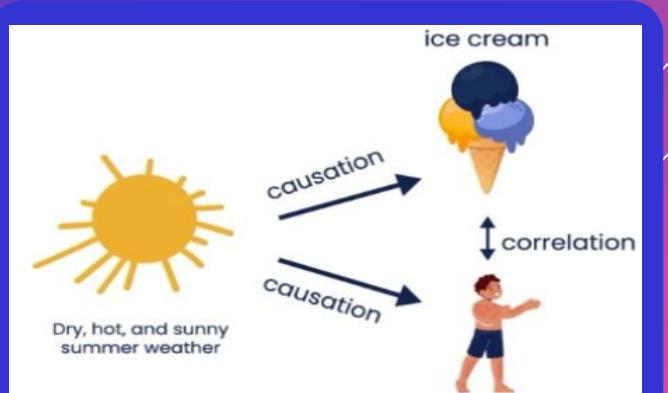
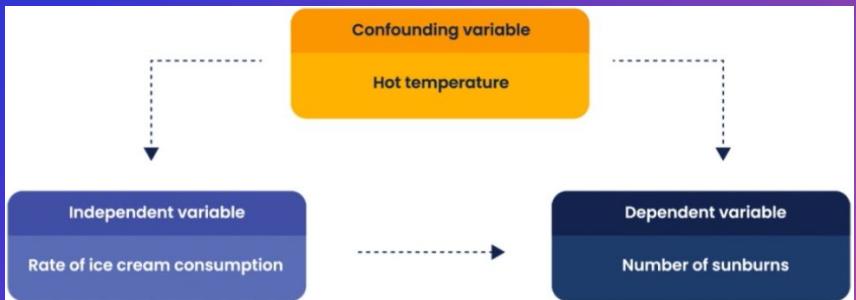
Identifikasi: Menentukan kausalitas lebih rumit daripada mengidentifikasi korelasi. Salah satu caranya adalah dengan melakukan eksperimen terkontrol (memanipulasi satu variabel untuk mengamati pengaruhnya terhadap variabel lain).



Contoh: Cuaca cerah menyebabkan banyak orang berselancar dan meningkatkan penjualan es krim.

# Kesalahan Umum

- 1) Correlation doesn't always imply Causation
- 2) Spurious Correlation
- 3) Confounding Variable



04

# Probability & Distributions



# Probability & Distribution

## Probability (Peluang)

Definisi: Ukuran kemungkinan terjadinya suatu peristiwa.

Contoh: Kemungkinan munculnya sisi head dari dua sisi koin, yaitu sisi head dan tail.

$$P(A) = \frac{\text{Number of times A occurs}}{\text{Total number of possible outcomes}}$$

## Basic Probability Rules



Additive Rules

- Mutually Exclusive**  
 $P(A \cup B) = P(A) + P(B)$

- Non-mutually Exclusive**  
 $P(A \cup B) = P(A) + P(B) - P(A \cap B)$



Multiplicative Rules

- Independent**  
 $P(A \cap B) = P(A) \times P(B)$
- Dependent**  
 $P(A \cap B) = P(A) \times P(B|A)$

Tipe Events	Definisi	Contoh
Mutually Exclusive	Tidak bisa terjadi bersamaan dalam satu waktu tertentu	Pelemparan 1 buah koin
Non-mutually Exclusive	Bisa terjadi bersamaan dalam satu waktu tertentu	Pelemparan 2 buah koin
Independent	Terjadinya suatu event tidak mempengaruhi event lainnya	Pelemparan 1 buah koin & 1 buah dadu
Dependent	Terjadinya suatu event mempengaruhi event lainnya	Pengambilan 2 kartu remi dari deck tanpa pengembalian

# Probability & Distribution

## Distribution (Persebaran)

- Definisi: Pola penyebaran nilai-nilai data
- menjelaskan bagaimana nilai-nilai dalam data tersebar, berapa sering muncul, dan bagaimana peluang tiap nilai terjadi.
  - Dalam analisis data, memahami distribusi adalah langkah awal sebelum membuat kesimpulan, model, atau prediksi.

Contoh:

Distribusi Jumlah Penjualan Harian

- menunjukkan **berapa banyak transaksi** yang terjadi tiap hari dalam sebulan.
- Misalnya: sebagian besar hari punya 200–300 transaksi, tapi akhir pekan bisa naik ke 500.

## Common Distribution

Tipe	Definisi	Contoh
Normal (Gaussian)	Data tersebar simetris di sekitar rata-rata (bentuk lonceng)	Nilai IQ, tinggi badan
Binomial	Hasil dari percobaan ya/tidak	Peluang sukses dari 10 kali percobaan
Poisson	Menghitung kejadian jarang	Jumlah error sistem per jam
Uniform	Semua nilai punya peluang sama	Hasil dadu fair
Eksponensial	Waktu antara dua kejadian	Waktu tunggu pelanggan berikutnya



# THANK YOU!