Time series data visualization

--Trends, Cycles, and Beyond

Département d'informatique

École supérieure d'ingénieurs Léonard-de-Vinci (ESILV)

28 Novembre, 2024

Ruiwen HE

What is Time Data?

- Sequential data indexed by time (e.g., daily sales, monthly revenue, hourly temperatures).
- Examples:
- Business: Quarterly profit trends.
- Healthcare: Daily patient counts during a pandemic.
- Transportation: Hourly traffic flow patterns.

ruiwen.he@devinci.fr

Why Does It Matter?

- Enables us to:
  - Identify long-term **trends** (e.g., growth in user activity).
  - Detect patterns or **anomalies** (e.g., sales spikes during promotions).
  - Make data-driven **predictions** (e.g., forecast next month’s demand).
- Applications Across Industries: Business intelligence, climate analysis, financial forecasting, supply chain optimization.

ruiwen.he@devinci.fr

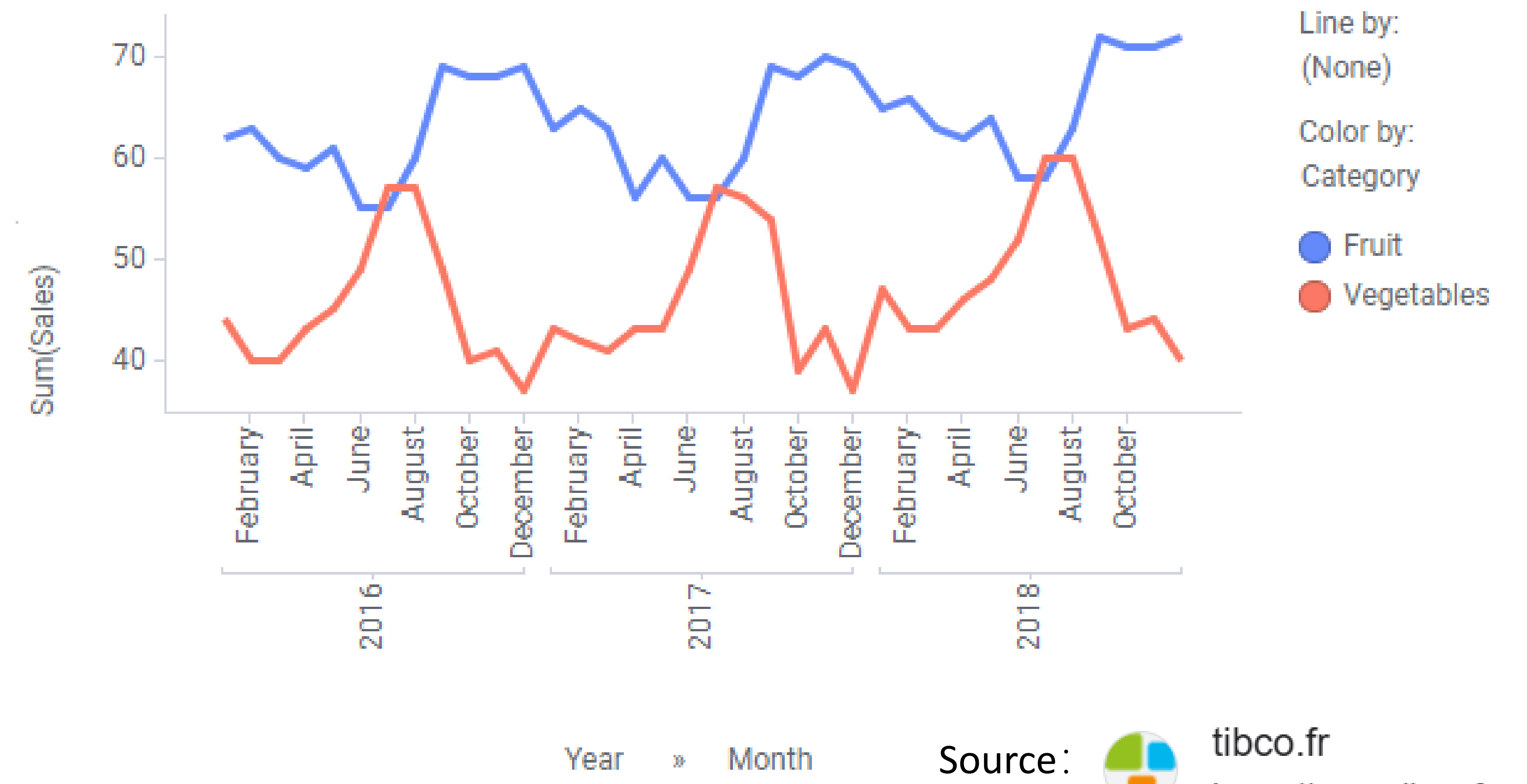
Objectives

- By the end of this session, you will:
  1. Understand the unique characteristics of time data and its challenges.
  2. Learn preprocessing techniques to prepare time data for visualization.
  3. Explore various visualization types for time series analysis.
  4. Grasp the principles of effective design to create insightful visuals.
  5. Identify common pitfalls in time-based visualizations and how to avoid them

ruiwen.he@devinci.fr

ruiwen.he@devinci.fr

Trends, Cycles, and Beyond



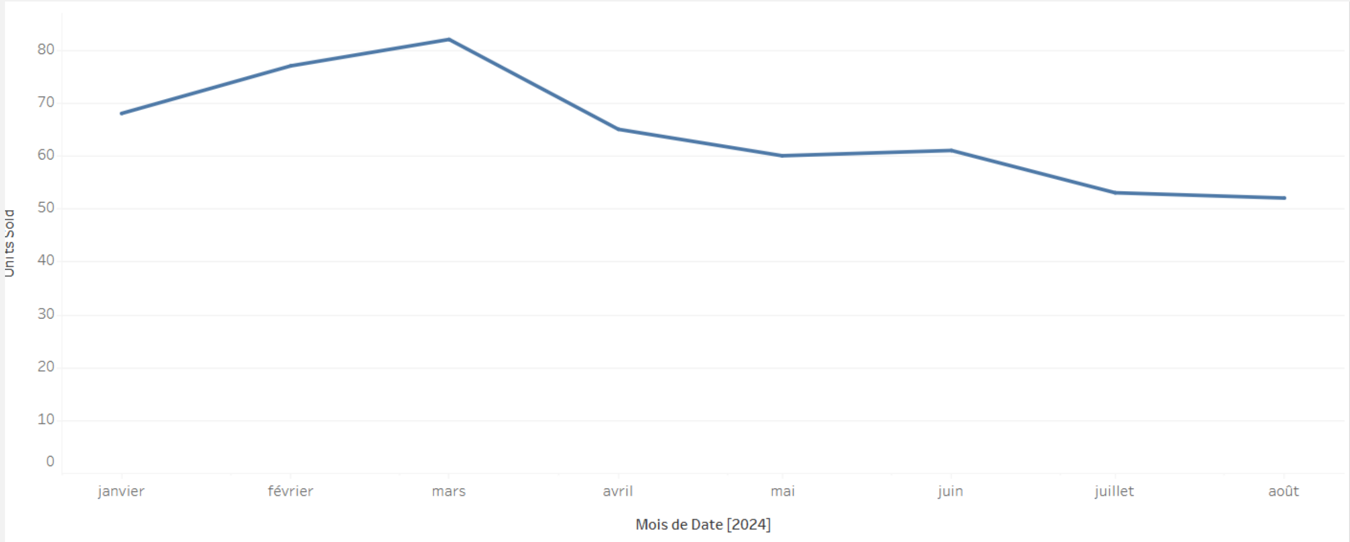
Source : tibco.fr https://www.tibco.fr

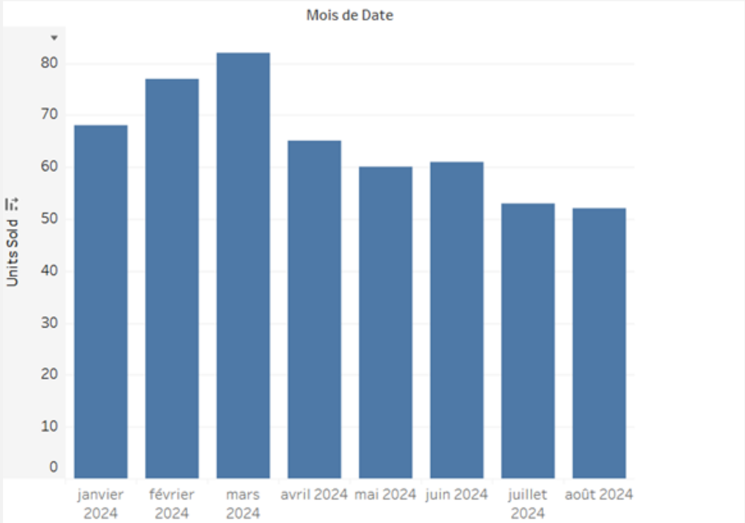
Features of Time Data

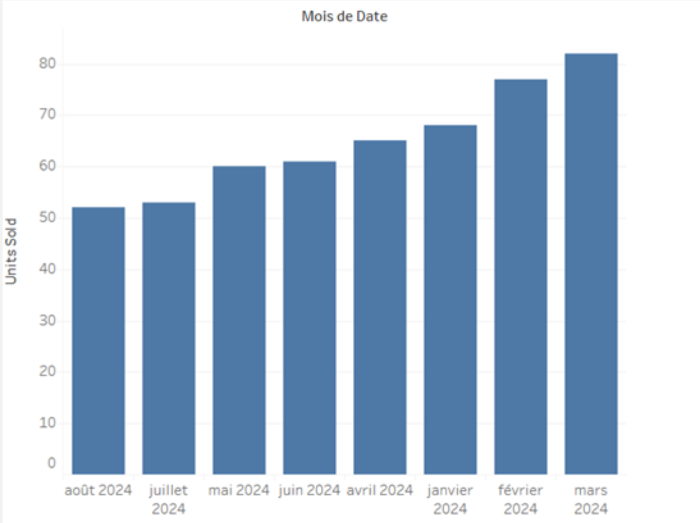
1. Temporal Sequence

- Observations are ordered in time, making their sequence critical.
- But it is not absolu
- Example: Monthly sales data must maintain its order for accurate trend analysis and find out the month with the best sales

Temporal Sequence







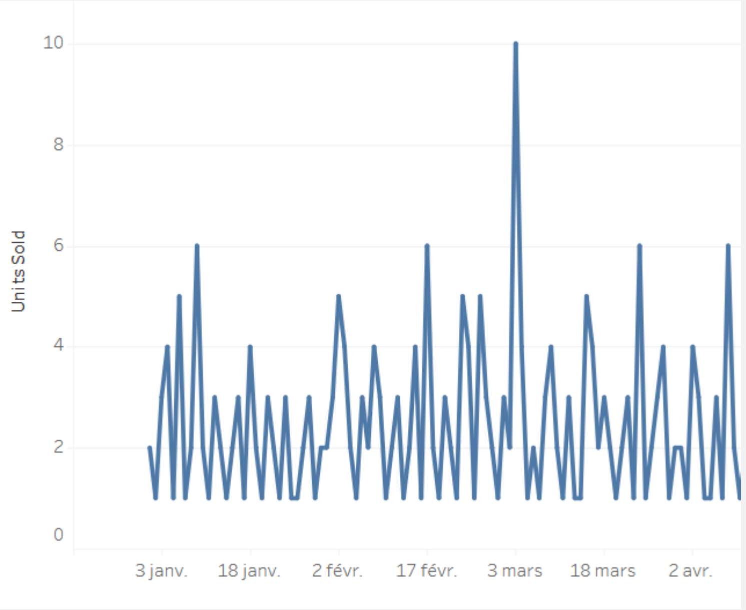
ruiwen.he@devinci.fr

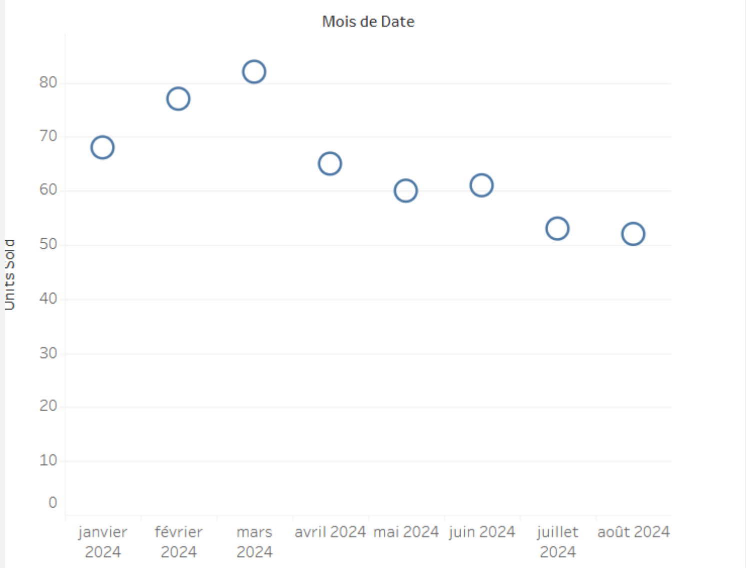
ruiwen.he@devinci.fr

Features of Time Data

2. Continuity or Discreteness

- Continuous: Data collected continuously over time (e.g., temperature readings every minute or eveyday).
- Discrete: Data captured at intervals (e.g., weekly sales reports).
- It do not mean **incomplete data**

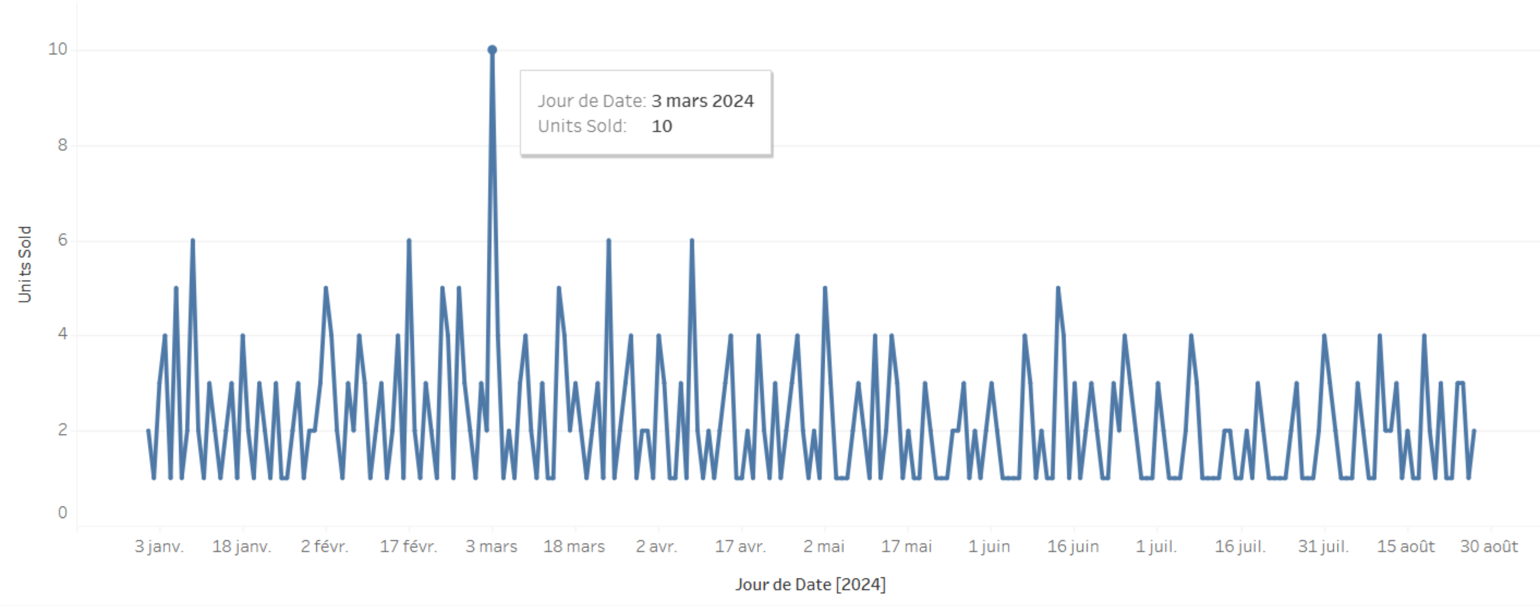




Features of Time Data

3. Granularity

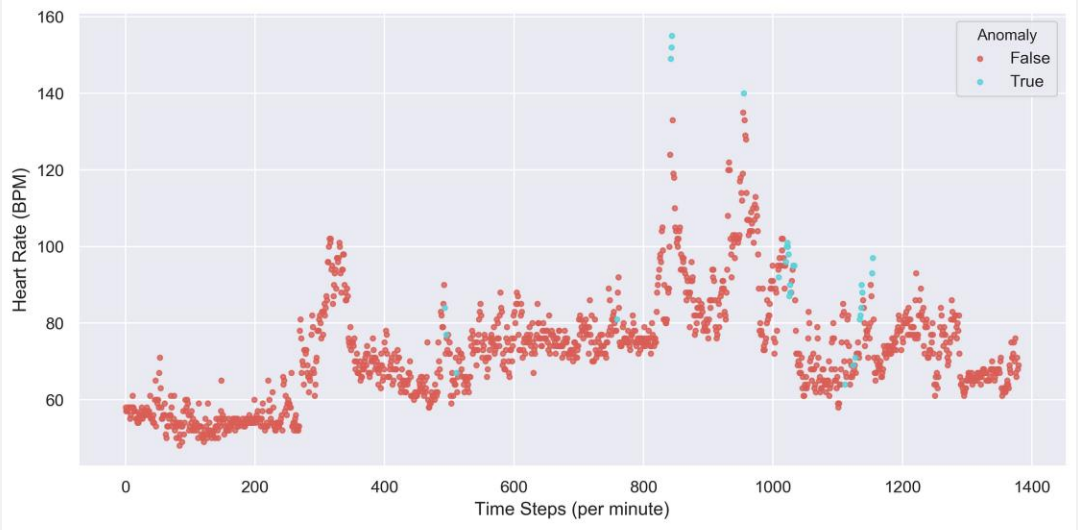
- Represents the level of detail, from high granularity (hourly) to low granularity (yearly).
- Choosing the correct granularity affects the insights you can derive.

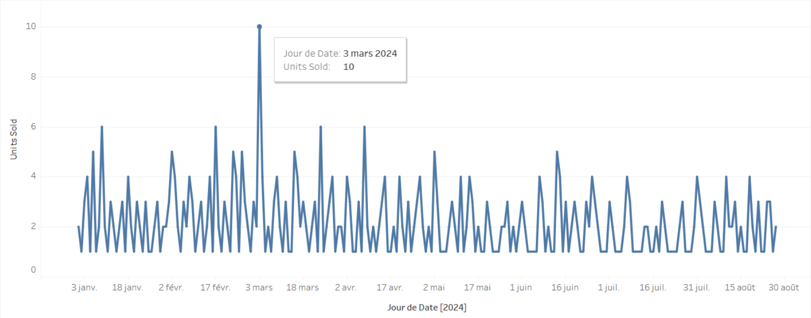


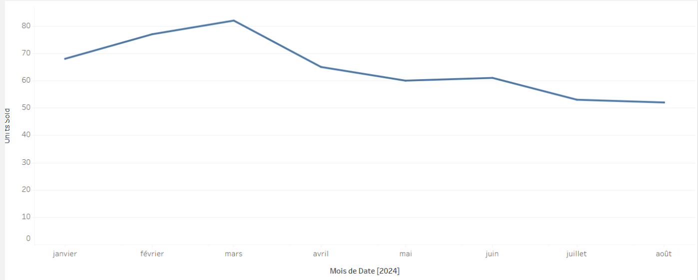
ruiwen.he@devinci.fr

Choosing the correct granularity

- Analysis Objective (Crucial reason): <u>**What question are you trying to answer?**</u>
- High granularity: Detecting anomalies (e.g., hourly web traffic spikes).
- Low granularity: Observing long-term trends (e.g., annual revenue growth).







Choosing the correct granularity

- Data Volume and Storage:
  - High granularity data consumes more storage and processing power.
  - Aggregation can simplify analysis and improve performance.
- Signal vs. Noise:
  - Fine-grained data may have more noise, making trends harder to spot.
  - Aggregated data smooths fluctuations but risks oversimplification.
- Nature of the Phenomenon:
  - Some patterns are visible only at specific granularities:
    - Seasonal sales may require weekly data.
    - Machine efficiency may need minute-by-minute monitoring

ruiwen.he@devinci.fr

ruiwen.he@devinci.fr

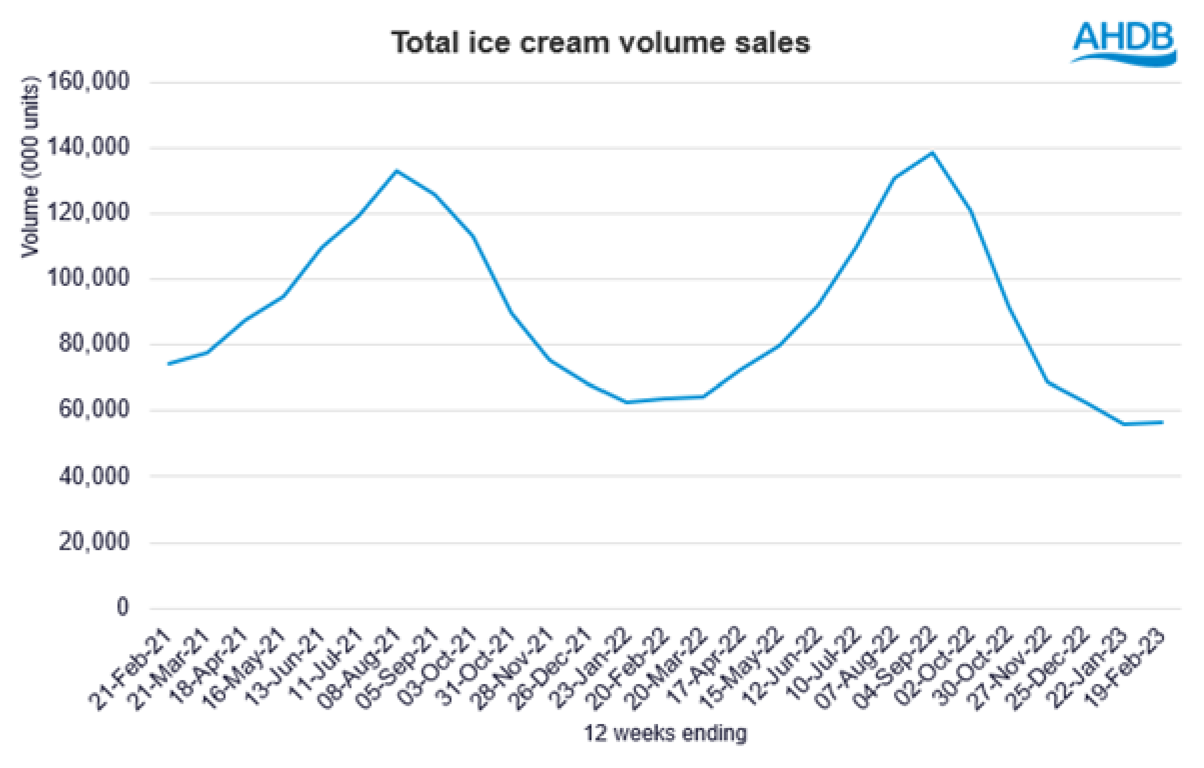
Examples of Granularity Choices

- **Finance:**
  - Hourly data for intraday trading analysis.
  - Monthly or yearly summaries for investment planning.
- **Retail:**
  - Daily data to track sales peaks.
  - Hourly data during promotions to monitor effectiveness.
- **Healthcare:**
  - Hourly patient admissions for staffing.
  - Annual trends for policy decisions.

Features of Time Data

4. Seasonality

- Repeated patterns over time, such as higher ice cream sales in summer or e-commerce peaks during holidays.



ruiwen.he@devinci.fr

Features of Time Data

5. Trend and Stationarity

- Trend: Long-term increase or decrease over time (e.g., rising housing prices).
  - Can be linear (steady growth or decline) or nonlinear (exponential growth).
  - Indicates underlying shifts due to external factors (e.g., population growth, economic changes).
- Stationarity: When statistical properties (mean, variance) do not change over time.
  - No trends or seasonality.
  - Fluctuations are random and independent of time.

Stationarity

- Importance of Stationarity:
- Many statistical models, like ARIMA, require stationarity for accurate forecasting.
- Transforming Non-Stationary Data:
- Detrending: Remove the trend to focus on residual patterns.
- Differencing: Subtract adjacent values to stabilize the mean.

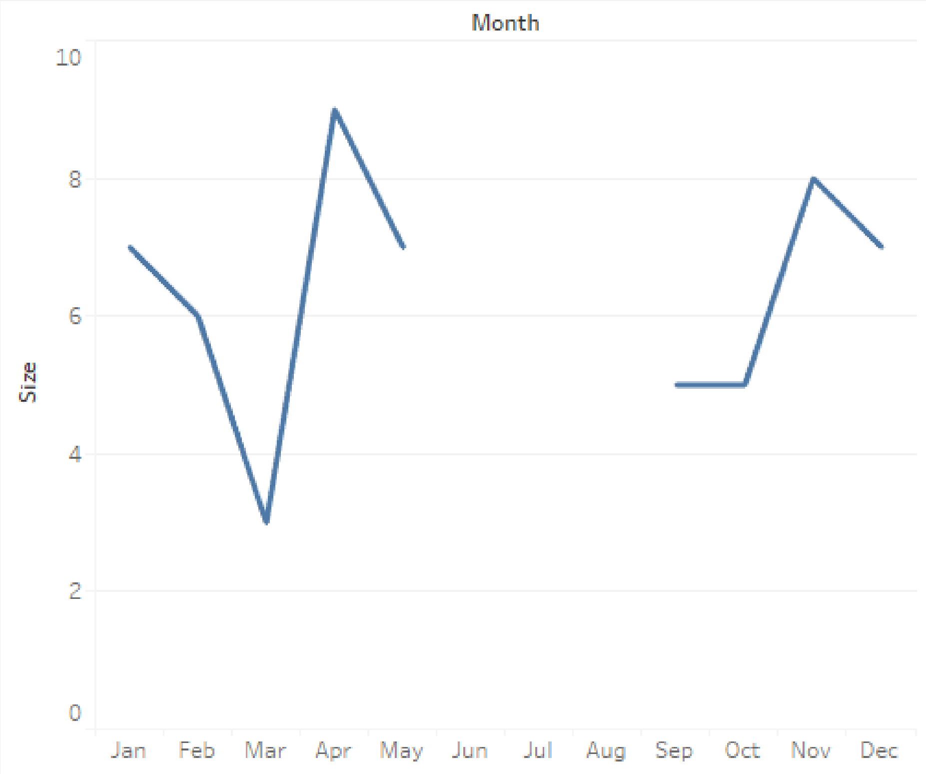
Aspect	Trend	Stationarity
Definition	Long-term upward/downward movement	Consistent statistical properties over time.
Impact on Analysis	Needs to be removed for some analyses	Essential for model stability.

ruiwen.he@devinci.fr

Challenges in Handling Time Data

Missing Data

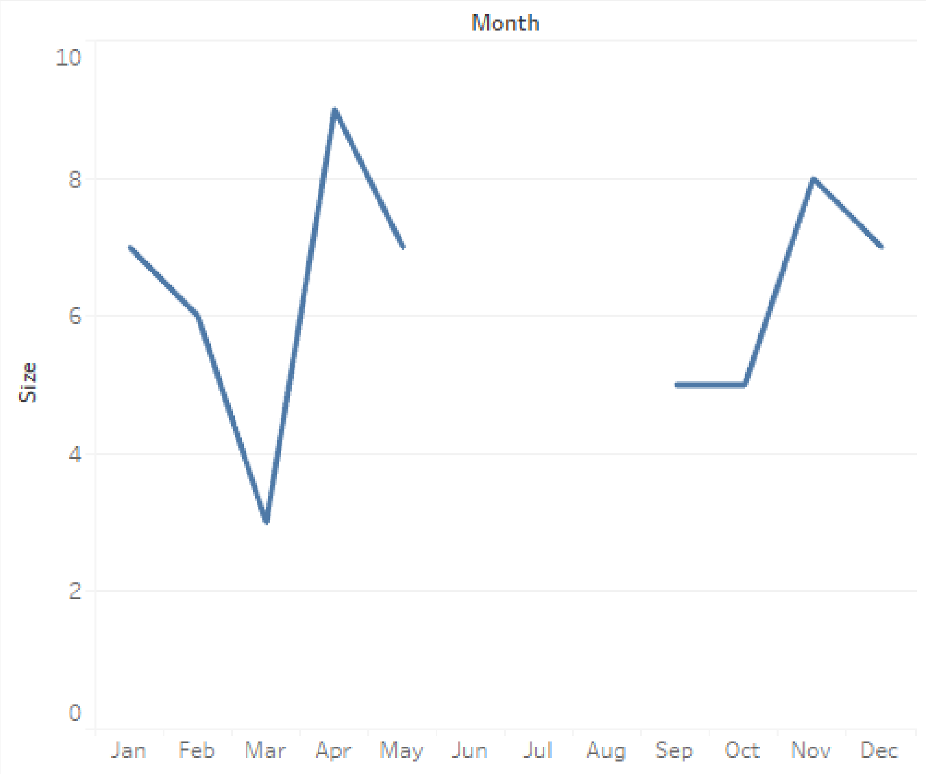
- Gaps in data points due to collection errors or interruptions (e.g., skipped days in a temperature log).
- Solution: Imputation techniques like forward-fill or interpolation (mean, median & mode)



Challenges in Handling Time Data

Missing Data

- Gaps in data points due to collection errors or interruptions (e.g., skipped days in a temperature log).
- Solution: Imputation techniques like forward-fill or interpolation (mean, median & mode)



Challenges in Handling Time Data

Irregular Time Intervals

- Irregular time intervals occur when data points are not collected at uniform time intervals. This often arises from limitations in data collection or external events that affect recording frequency.
- Impact on Analysis: Makes it difficult to directly apply standard time series models that assume constant time intervals.
  - Interpolation vs. Aggregation:
  - Interpolation: Filling gaps by estimating missing values based on available data (e.g., linear interpolation).
  - Aggregation: Summing or averaging data over fixed time intervals to standardize it (e.g., averaging daily traffic data into weekly periods).

Examples of Irregular Time Intervals

- Website Analytics: Traffic data collected during periods of high activity or during specific marketing campaigns, leading to irregular peaks.
- Sensor Data: Temperature or humidity readings where devices might malfunction or have scheduled downtimes, causing data to be recorded at irregular intervals.
- Stock Market:Data points might be missing during market holidays or trading pauses, leading to gaps in time intervals.

ruiwen.he@devinci.fr

Strategies to Handle Irregular DataChoosing

- Between Resampling and Interpolation:
  - Use **resampling** when the intervals are relatively consistent but simply need aggregation.
  - Use **interpolation** when there are gaps in the data that need estimation, especially when dealing with continuous data like temperatures or financial data.

- Tools & Libraries:Python:
  - Libraries like pandas for resampling or scipy for interpolation.
  - Tableau: Offers resampling functions for handling irregular data.

ruiwen.he@devinci.fr

Challenges in Handling Time Data

- **Multi-Level Seasonality**
- Complex datasets may exhibit nested patterns (e.g., daily and weekly sales fluctuations).
- **Noisy Data**
- Fluctuations caused by random factors, requiring smoothing techniques for clearer trends.

ruiwen.he@devinci.fr

Examples of Time Data Characteristics in Real Life

- Seasonality: Retail sales peaking in November/December due to holidays.
- Trend: The increase in average global temperatures over decades.
- Noise: Sudden stock price spikes due to market news.
- Multi-Level Granularity: Analyzing daily and hourly customer visits in a store.

ruiwen.he@devinci.fr

Preprocessing Time Data

- Preprocessing time data involves transforming raw time series data into a clean, consistent format suitable for analysis and modeling.
- Steps include handling missing values, resampling, removing trends or seasonality, and converting time formats.

ruiwen.he@devinci.fr

Common Steps in Preprocessing Time Series Data

- Handling Missing Data:
- Missing values are common in time series data due to sensor failures, gaps in reporting, or irregular time intervals.
- Techniques:
  - Forward Fill: Propagate previous valid data point.
  - Backward Fill: Propagate next valid data point.
  - Interpolation: Estimate missing values (e.g., linear, cubic interpolation).
  - Deletion: Remove missing entries if the gaps are minimal or the data is not critical.
- Example: Sales data for a specific day is missing, so interpolation estimates the missing value based on nearby data points.

ruiwen.he@devinci.fr

Resampling or Aggregation

- Transforming data into uniform time intervals (e.g., from hourly data to daily data).
- Methods:
  - Upsampling: Increase the frequency of data (e.g., daily to hourly), often requiring interpolation to fill gaps.
  - Downsampling: Decrease the frequency (e.g., from minute-by-minute data to daily data) by aggregating (mean, sum, etc.).
- Example: If a dataset has hourly temperatures, downsample to daily averages to reveal broader trends.

ruiwen.he@devinci.fr

Time Format Conversion

- Standardizing Date and Time Formats:
- Ensure all date-time information follows a consistent format (e.g., converting all timestamps to a single time zone).
- Convert time into structured formats like year, month, day, hour, minute, etc., for easier analysis.
- Example: Converting timestamps like 2024-11-26 14:32:00 into a simple year-month-day-hour-minute format.

ruiwen.he@devinci.fr

Removing Trends and Seasonality

- Detrending:
  - Remove long-term trends (e.g., gradual increases or decreases) to focus on short-term fluctuations.
- Seasonality Removal:
  - Use decomposition methods (e.g., STL decomposition) to isolate and remove seasonal patterns.
- Example: Identifying holiday-season effects on sales and removing them to better observe underlying business trends.

ruiwen.he@devinci.fr

Normalization and Scaling

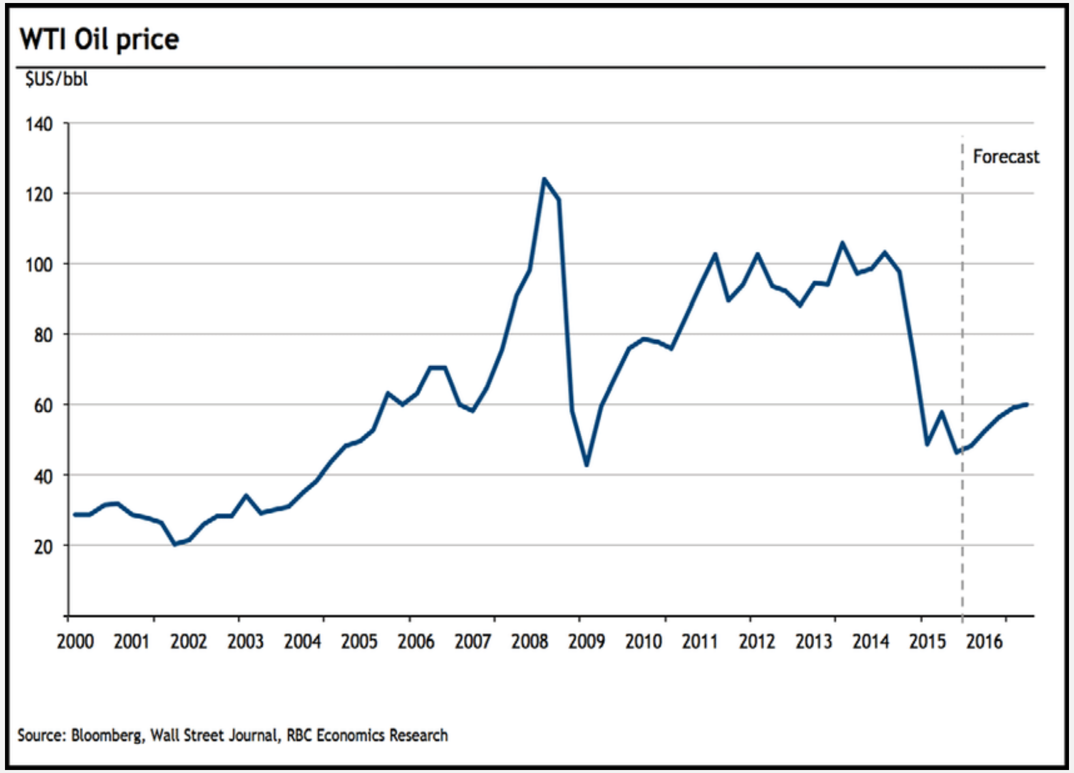
- Scaling data to a consistent range (e.g., between 0 and 1) helps when comparing different datasets.
- Methods:
  - Min-Max Scaling: Rescale the data by subtracting the minimum value and dividing by the range.
  - Z-Score Scaling: Subtract the mean and divide by the standard deviation to standardize the data.
- Example: Normalizing stock prices before combining them with other economic indicators for comparison.

ruiwen.he@devinci.fr

Common Chart Types for Time Data

1. Line Chart

- **Purpose:**
  - Best for showing trends over time, particularly for continuous data.
  - Demonstrates the rise or fall of values over a time period.
- **Use Cases:**
  - Stock prices, temperature changes, website traffic.
- **Visual Example:**
  - Line chart oil prices over several months.



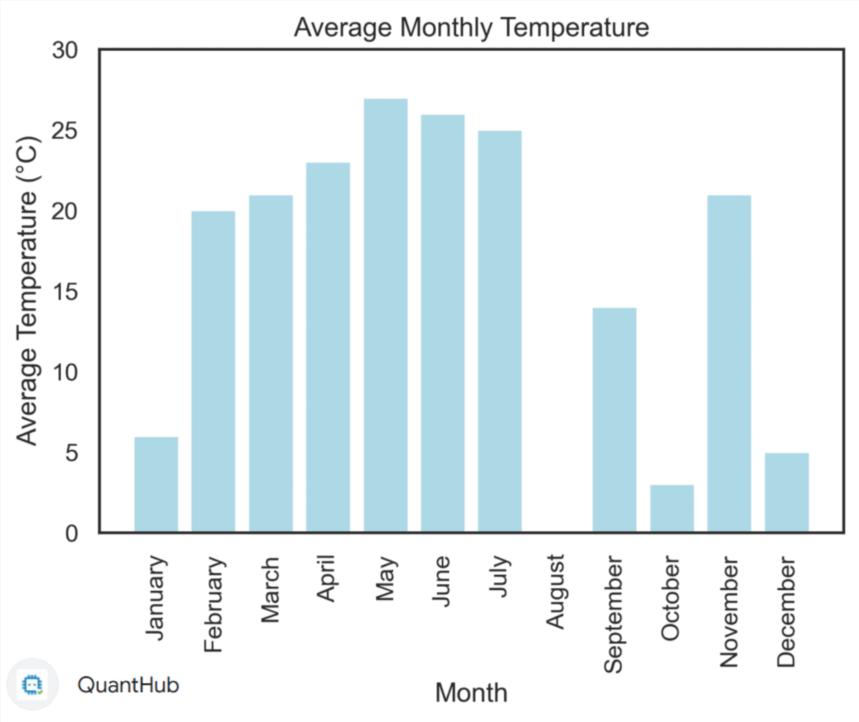
ruiwen.he@devinci.fr

ruiwen.he@devinci.fr

Common Chart Types for Time Data

2. Bar Chart

- **Purpose:**
  - Useful for comparing discrete data points across time intervals.
  - Typically used for categorical time data (e.g., sales by month).
- **Use Cases:**
  - Monthly sales figures, daily traffic counts.
- **Visual Example:**
  - Bar chart comparing temperature for each month of the year.

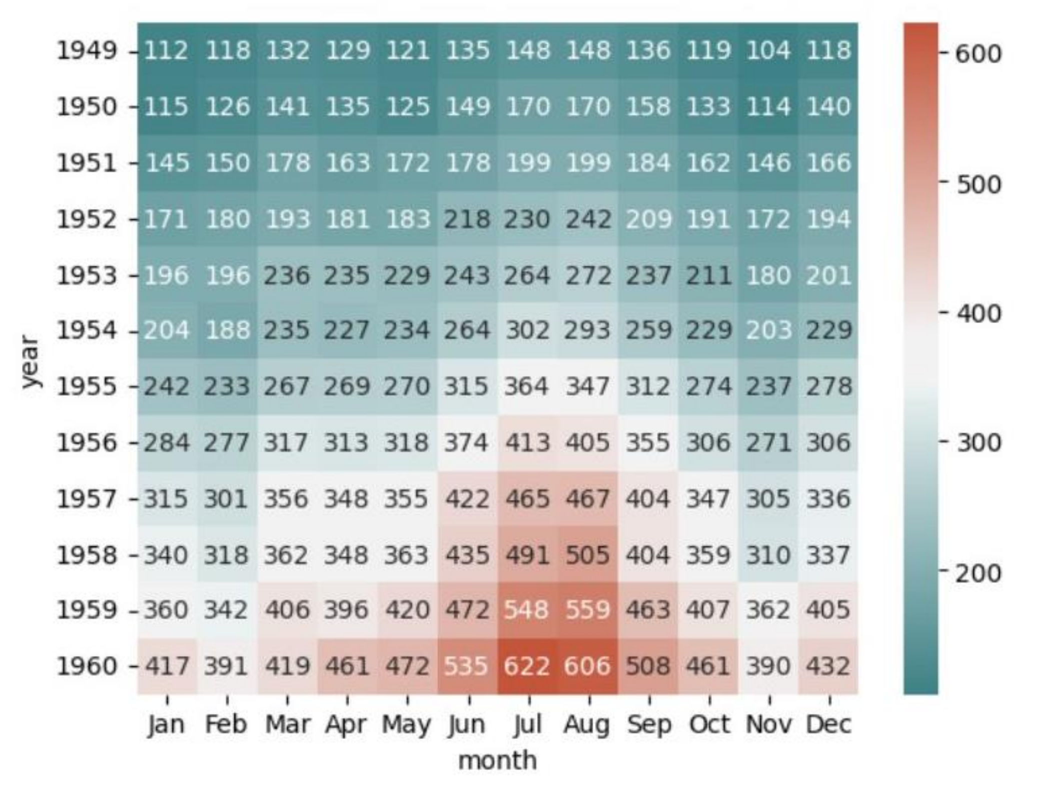


ruiwen.he@devinci.fr

Common Chart Types for Time Data

3. Heatmap

- **Purpose:**
  - Highlights patterns in large datasets by color-coding values.
  - Best for visualizing data intensity or frequency across two-time dimensions (e.g., day vs. time of day).
- **Use Cases:**
  - Website activity heatmap (e.g., user activity by time of day and day of the week).
- **Visual Example:**
  - Heatmap showing vol traffic intensity each month over years



Common Chart Types for Time Data

4. Timeline Chart

- **Purpose:**
  - Displays events or milestones over time. Used for categorical data with specific timestamps.
  - Shows when events happened relative to each other.
- **Use Cases:**
  - Product launch timelines, project milestones.
- **Visual Example:**
  - Timeline of major events in a project or company history.

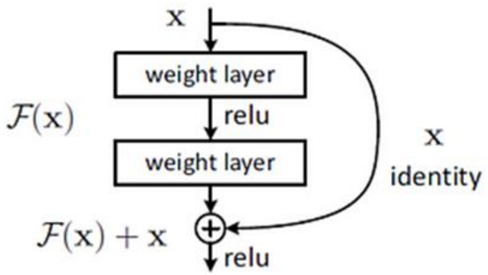
ruiwen.he@devinci.fr

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

ruiwen.he@devinci.fr

Python for Data

Building block of ResNet



ImageNet
2009

AlexNet
(deepCNN)
2012

Fast R-CNN
ResNet
2015

RevNet
YOLOv2
Transformer
2017

Vision
Transformer (ViT)
2021

1998
LeNet

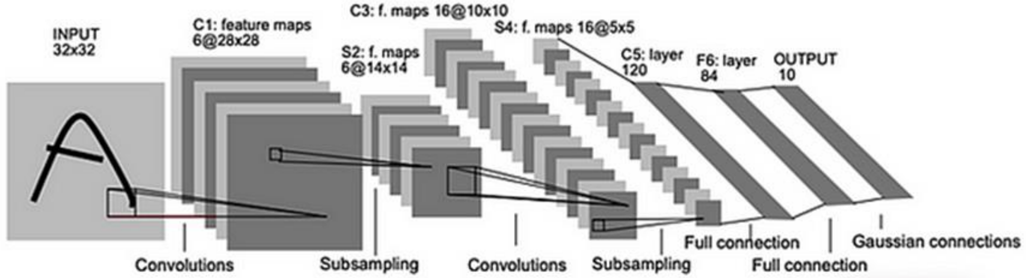
2010
Pascal VOC

2014
VGG
R-CNN
GoogLeNet
(Inception)

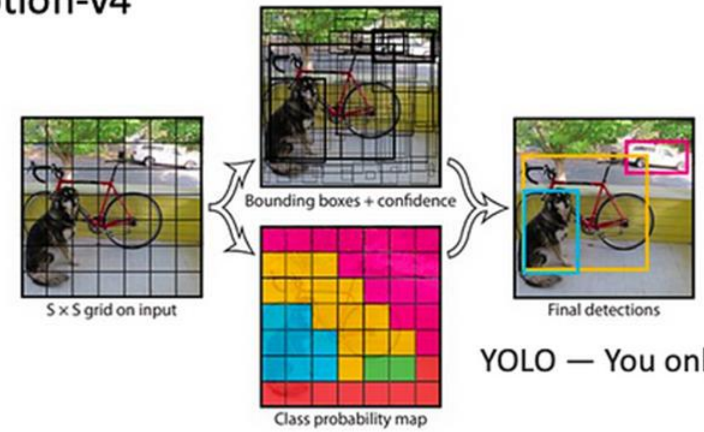
2016
SSD
YOLO
Inception-v4

2018
YOLOv3
BERT

2020
Image GPT



The architecture of LeNet-5 for digit recognitio Bullets nn LeCun



YOLO — You only look once

Patterns and Decomposition in Time Data

- **1. What are Patterns in Time Data?**
- Patterns in time data are recurring behaviors or structures that emerge over time. Understanding these patterns helps in better visualization and analysis.
- Key Patterns:
  - **Trend**
  - **Seasonality**
  - **Residual or Noise**
  - **Anomalies**

ruiwen.he@devinci.fr

Decomposition of Time Series Data

- Decomposition breaks a time series into its components for easier analysis and visualization. It is particularly useful for identifying trends and seasonality.
- Decomposition allows analysts to:
  - **Understand data dynamics:** By separating components, one can see how data behaves over time.
  - **Isolate trends:** Long-term directions can be seen without short-term fluctuations.
  - **Spot seasonality:** Identify repeating cycles that occur over a fixed interval.
  - **Focus on anomalies:** Easier detection of irregular behaviors after removing predictable components.
  - **Prepare data for forecasting:** Cleaned components are useful for predictive models.

Decomposition of Time Series Data

1. **Additive Model:**
   Assumes components add up:

- $Y(t)=T(t)+S(t)+R(t)$
- $Y(t)$: Observed value.
- $T(t)$: Trend component.
- $S(t)$: Seasonal component.
- $R(t)$: Residual (random variation).
  Used when fluctuations remain constant over time

2. **Multiply** Model **:**

- $\boldsymbol{Y(t)=T(t)\cdot S(t)\cdot R(t)}$ **Y(t)**
- Used when fluctuations scale with the trend (e.g., larger values lead to larger variations).

ruiwen.he@devinci.fr

Techniques for Decomposition

- **Moving Averages for Trend Detection**:
- Smooths data by averaging a window of previous data points.
- Eliminates short-term fluctuations to highlight the underlying trend.
- Formula for a simple moving average (SMA)

$$SMA_t = \frac{1}{n}\sum_{i=0}^{n-1} X_{t-i}$$

Techniques for Decomposition

- **Seasonal Component Extraction**:
- Seasonal patterns are identified by grouping data by periodic intervals (e.g., monthly sales trends).
- Compare actual data against averages or medians for each time period

ruiwen.he@devinci.fr

ruiwen.he@devinci.fr

Techniques for Decomposition

- Decomposition Using Libraries:
- Tools like Python's statsmodels or R's forecast package simplify decomposition.

```
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(data, model='additive', period=12)
result.plot()
```

Visualizing Patterns

- Use visual aids to highlight patterns:
- **Line Charts** for trends.
- **Heatmaps** for seasonality (e.g., hourly activity by day).
- **Scatterplots** to spot anomalies.
- **Bar Chart :**Comparing quantities across distinct time periods. Use when you're comparing specific time segments (e.g., monthly sales) rather than trends.
- **Decomposition Graphs** for trend/seasonal breakdowns.

ruiwen.he@devinci.fr

Avoiding Common Visualization Pitfalls

Overcomplicating Visuals

- **Pitfall:** Too many data points or using overly complex chart types that confuse the viewer.
- **Fix:** Keep charts simple and focused on the core message.

ruiwen.he@devinci.fr

ruiwen.he@devinci.fr

Avoiding Common Visualization Pitfalls

Misleading Axes

- **Pitfall:** Using inconsistent time intervals or scaling axes in a way that distorts data trends.
- **Fix:** Maintain consistent time intervals and logical scaling for accurate representation.

ruiwen.he@devinci.fr

Avoiding Common Visualization Pitfalls

Ignoring Audience

- **Pitfall:** Using overly technical charts when the audience is not familiar with the data.
- **Fix:** Tailor the complexity of the chart to the audience’s understanding level.