Introduction to Time Series Data

Friday, April 4, 2025 11:32 AM

1. Time Series Data

- Time Series data is a **sequence of observations collected at regular time intervals** (e.g., every minute, hour, day, week, etc.).
- Each data point is **associated with a specific timestamp**, and the **order of data points matters**.

Examples:

- **Finance**: Daily closing stock prices (e.g., AAPL stock on Jan 1, Jan 2…)
- **IoT/Sensors**: Hourly temperature readings from a weather sensor
- **Web Analytics**: Number of website users per minute/hour
- **Energy Sector**: Electricity consumption recorded every 15 minutes

2. Time Index

- A **Time Index** is the **timestamp or datetime column** in the dataset that uniquely identifies the point in time for each record.
- This acts as the **primary axis** to perform time-based operations.

Importance:

- Without a proper time index, time-based calculations (like rolling averages or resampling) won't work.
- Helps in chronological ordering and detecting missing intervals.

Operations that require a Time Index:

- **Resampling**: Aggregating data (e.g., converting daily data into monthly averages)
- **Rolling Window Calculations**: Applying functions over a moving window (e.g., 7-day rolling average)
- **Time Difference (lag/lead)**: Comparing past/future values for a given time point

3. Decomposition breaks a time series:

Time series decomposition is a statistical method used to break down a time series into **several distinct components** to better understand the underlying patterns in the data.

1. Trend
The long-term progression of the series (e.g., an upward or downward slope over years).
2. Seasonality
Regular, repeating patterns (e.g., temperature increases in summer and decreases in winter).
3. Residual (or Noise)
The random variation or irregularities not explained by trend or seasonality.
4. Observed
The actual original time series (which is the sum of the above parts in an additive model).

A) Trend

long-term movement or direction in the data over time.
It shows whether the variable is **increasing, decreasing, or stable** in the long run.

Characteristics:

- Does **not** repeat periodically.
- Can be linear or non-linear.
- Often due to external factors like economic growth, product popularity, etc.

Examples:

- Increase in electric vehicle sales over the past 10 years.
- Gradual decline in birth rates over decades.

B) Seasonality

repetitive and predictable patterns observed at **fixed time intervals**, such as daily, weekly, monthly, or annually.

Examples:

- More **ice cream sales in summer** and less in winter.
- Higher electricity usage **every evening**.

Characteristics:

- **Fixed frequency** (e.g., every 24 hours, every 7 days).
- Often related to **human behavior, natural phenomena, or calendar effects**.

Important Note:

Seasonality and trend can coexist in the same time series.

C) Noise (Residuals)

random or irregular variations in the data that **cannot be explained by trend or seasonality**.

Characteristics:

- Unpredictable and **doesn't follow a pattern.**
- Could be due to **measurement errors**, **random events**, or **unexpected behavior**.

Example:

- A sudden spike in sales due to a viral ad campaign that wasn't planned.
- Sensor errors in IoT readings.

D) Observed

The actual original time series (which is the sum of the above parts in an additive model).

Summary Table:

Component	Description	Repeats?	Examples
Time Series	Data indexed by time	-	Daily temperature, hourly traffic
Time Index	Timestamp used to align data	-	Date, Datetime, Timestamp column
Trend	Long-term movement (up/down/stable)	❌	Rising EV sales
Seasonality	Recurring patterns at fixed intervals	✅	Summer sales, weekend website visits
Noise	Irregularities that can't be modeled	❌	Sensor glitches, unplanned events

ACF (Autocorrelation Function) – *How much the data is related to its past*

- ACF **checks how today's value is related to previous days** (like yesterday, 2 days ago, 3 days ago...).
- If there's a strong link, it means **past values influence the current value.**
- It helps find **patterns** or **seasonal cycles** in the data.
- It's used to choose the **MA (Moving Average)** part of an ARIMA model.

PACF (Partial Autocorrelation Function) – *How much one specific past day affects today*

- PACF checks how much **one specific past value** (say, 3 days ago) affects today, **without** the influence of the days in between (like 1 or 2 days ago).
- It helps find the **true influence** of a particular lag.
- It's used to choose the **AR (AutoRegressive)** part of an ARIMA model.

Feature	ACF	PACF
Measures	Total correlation	Direct correlation only
Includes	Direct + indirect effects	Only direct effect
Use in ARIMA	Helps choose **q** (MA part)	Helps choose **p** (AR part)
Example (Lag 3)	Includes effect of lag 1 & 2	Removes effect of lag 1 & 2

Moving Average & Standard Deviation in Time Series Analysis

Friday, April 4, 2025 1:48 PM

essential for smoothing noisy time series data and detecting trends.

1. Simple Moving Average (SMA)

smooths time series data by calculating the average **over a fixed window size.**
It helps to **remove short-term fluctuations** and highlight long-term trends.

Formula

$$SMA_t = \frac{(X_t + X_{t-1} + X_{t-2} + ... + X_{t-n+1})}{n}$$

where:

- X_t is the value at time t
- n is the window size (number of time steps for averaging)

Example

If we use a 7-day SMA on stock prices, it will return the average price of the last **7 days**, updating each day.

2. Exponential Moving Average (EMA)

more weight to recent observations, making it **more responsive to new data compared to SMA.**

Formula

$$EMA_t = \alpha \times X_t + (1 - \alpha) \times EMA_{t-1}$$

where:

- α is the **smoothing factor** ($\alpha = 2/(n+1)$)
- X_t is the current value
- EMA_{t-1} is the previous EMA

🔷 **EMA reacts faster to changes** than SMA, making it useful for financial markets.

3. Rolling Standard Deviation (Rolling Std)

- Measures the **spread** of the data over a window size.
- Helps detect **volatility shifts** in time series.
- High standard deviation → Data is more **spread out** (volatile).
- Low standard deviation → Data is more **stable**.

Formula

$$\text{Rolling Std}_t = \sqrt{\frac{\sum(X_i - \bar{X})^2}{n}}$$

where:

- X_i are the values in the window
- $\bar{X}$ is the mean of the window

4. Effects of Lag & Window Size

Window Size Effect

- **Smaller Window (e.g., 5 days)** → Reacts quickly but may be noisy.
- **Larger Window (e.g., 30 days)** → Smoother but slower to respond.

Lag Effect

- **SMA has more lag** because it gives equal weight to all values.
- **EMA has less lag** because it emphasizes recent values.

Outlier Detection using Z-Score and IQR

Friday, April 4, 2025 2:34 PM

- Outliers in time series data are **unusual data points that deviate significantly from the normal pattern**.
- Detecting these anomalies is crucial for applications like **sensor monitoring, fraud detection, and system performance analysis**.

1. Z-Score Based Outlier Detection

The **Z-Score** measures **how far a data point is from the mean** in terms of standard deviations.

Student's Score	Z-Score Calculation	Meaning
70 (Mean)	$(70 - 70)/10 = 0$	At the mean
80	$(80 - 70)/10 = 1$	1 SD above mean
90	$(90 - 70)/10 = 2$	2 SDs above mean
60	$(60 - 70)/10 = -1$	1 SD below mean

Formula

$$Z = \frac{X - \mu}{\sigma}$$

where:

- X = Data point
- μ = Mean of the dataset
- σ = Standard deviation of the dataset

- A **high absolute Z-score (>3 or <-3)** indicates an outlier.
- The threshold can be adjusted based on data characteristics.

2. Interquartile Range (IQR) Based Outlier Detection

IQR is a **robust statistical method** that detects outliers **based on quartiles (25th and 75th percentiles).**

Formula

$$IQR = Q3 - Q1$$

$$\text{Lower Bound} = Q1 - 1.5 \times IQR$$

$$\text{Upper Bound} = Q3 + 1.5 \times IQR$$

where:

- $Q1$ = 25th percentile
- $Q3$ = 75th percentile
- Any value **outside** the lower and upper bounds is considered an outlier.

✅ **Z-Score Method:**

- Works well for **normally distributed** data.
- **Sensitive to extreme values** (can be affected by skewed data).
- **Threshold** can be adjusted (**Z>3 by default**).

✅ **IQR Method:**

- Works well for **skewed** or **non-normal** data.
- More robust to extreme outliers.
- **Best for financial or sensor monitoring applications**.

✅ **Choosing the Best Method**

- If data is **normally distributed**, use **Z-score**.
- If data is **skewed**, use **IQR**.
- For **real-world datasets**, compare both methods.

Time Series Specific Outliers

Friday, April 4, 2025 3:44 PM

Unlike standard outlier detection, time series data has unique characteristics. Outliers in time series can be:

1. Types of Time Series Outliers

1 Point Outliers

- A sudden spike or dip that is inconsistent with past data.
- **single data point significantly deviates.**
- Example: A temperature sensor recording **50°C** when the usual range is 20-30°C.

2 Contextual Outliers

- A value that is **only an outlier in a certain context**
- A value that may be normal overall but is **unusual for a specific time period.**
- Example: A **sharp drop in retail sales** on a **holiday**, whereas it would be normal on a non-holiday.

3 Collective Outliers

- A sequence of values that **together deviate from the pattern**.
- A group of values that deviate from the expected pattern.
- Example: **Several days of unusually high temperatures** during winter.

2. Methods for Detecting Time Series Outliers

1 Rolling Window Detection

- Uses a **moving average** or **rolling standard deviation** to detect outliers.
- **Concept:** If a value **differs significantly from the rolling mean**, it may be an outlier.
- **Formula:**

$$\text{Rolling Z-Score} = \frac{X_t - \text{Rolling Mean}}{\text{Rolling Std}}$$

- **Best For:** Detecting sudden spikes/dips (Point Outliers).

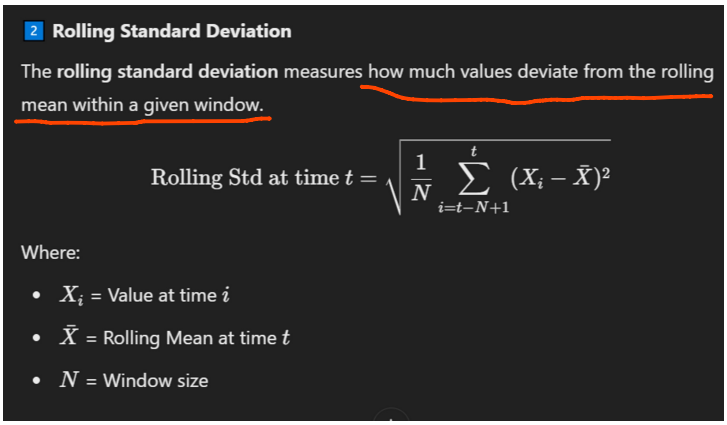
1 Rolling Mean (Moving Average)

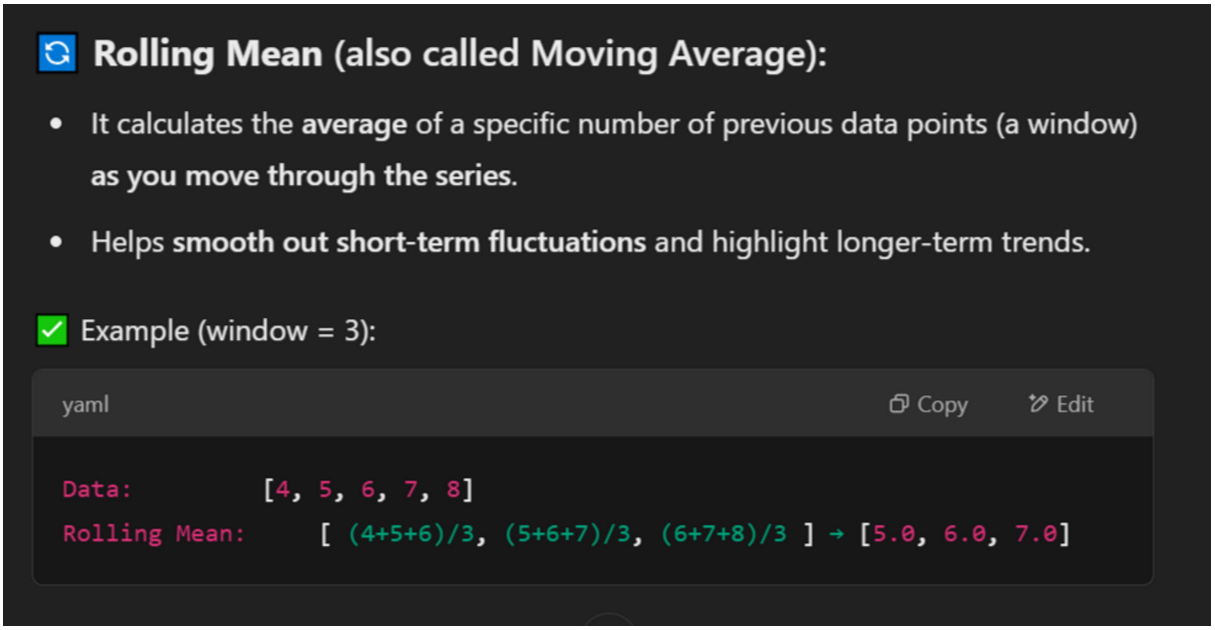
The **rolling mean** (or **moving average**) smooths time series data by calculating the average over a specified window.

$$\text{Rolling Mean at time } t = \frac{1}{N} \sum_{i=t-N+1}^{t} X_i$$

Where:

- X_i = Value at time i
- N = Window size (number of previous observations)
- t = Current time step





2 Seasonal Z-Score Detection

- **Compares a value to similar points in previous seasons.**
- **Concept:** If a value deviates significantly from past values for the same time (e.g., last year's sales in the same month), it's an outlier.
- **Best For:** Detecting **contextual outliers** in seasonal data.

Isolation Forest for Anomaly Detection

Friday, April 4, 2025 5:05 PM

- Isolation Forest is an **unsupervised machine learning algorithm** used for **anomaly detection**.
- It works by **isolating data points** that are different from the rest of the dataset.
- It is particularly effective for **time-series outlier detection** because it is **fast and doesn't require labeled data.**
- It works by **randomly partitioning the dataset** and isolating anomalies based on how quickly they are separated from the majority of data points.

🔷 How Isolation Forest Works

Random Partitioning:

- The dataset is randomly split into subsets **using decision trees.**
- Each split isolates a data point based on feature values.

Isolation Depth:

- **Outliers (**Anomalies**) get isolated faster (in fewer splits).**
- Normal points take **longer** to be isolated.

Anomaly Score:

- A score is assigned to each point based on how **easily it was isolated**.
- High anomaly scores = Outliers
- Low anomaly scores = Normal values

Scoring:

- Data points that are isolated in **fewer splits** are considered anomalies (-1).
- Normal data points require **more splits** (+1).

Neighborhood Comparison Methods for Anomaly Detection

Friday, April 4, 2025 5:51 PM

- **Neighborhood-based methods** detect anomalies by **comparing each data point to its neighbors**.
- If a point is **far from its neighbors** or in a **low-density region**, it is considered an **anomaly**.

These methods are useful for **time series outlier detection**, especially **when data exhibits clusters or varying densities.**

Method	Concept	Best For
k-NN (k-Nearest Neighbors)	Compares distance to k-nearest points	General anomaly detection
LOF (Local Outlier Factor)	Detects density-based local outliers	Sparse and dense regions
DBSCAN (Density-Based Clustering)	Finds clusters and marks points in low-density areas as anomalies	Clustering + anomaly detection

1 k-NN (k-Nearest Neighbors) for Anomaly Detection

- **Compute the distance** to the k-nearest neighbors.
- If the distance is significantly larger than average, label it an **outlier**.

2 LOF (Local Outlier Factor) for Density-Based Anomaly Detection

Local Outlier Factor (LOF) **compares a point's density to its neighbors**.
If the density **drops significantly**, the point is an **outlier**.

Compute the **density** of each point and its k-nearest neighbors.
If the density is **much lower than its neighbors**, the point is an **anomaly**.

3 DBSCAN (Density-Based Clustering) for Outliers

DBSCAN detects **clusters** and labels **low-density points** as anomalies.

Steps

Identifies **dense clusters** based on eps (distance threshold).
Points **outside clusters** are flagged as **outliers**.

- **DBSCAN automatically finds clusters** and isolates anomalies.
- Works well when anomalies are **in low-density areas**.