

# Analyse und Visualisierung von Zeitreihen- Daten in Python



Hmetrica



# Analyse und Visualisierung von Zeitreihen-Daten in Python

Tag 2

# Inhalte – Was haben wir vor?

## Tag 1

1. Einführung in Zeitreihendaten in Python
2. Zeitreihen und ihre Merkmale visualisieren
3. Zeitreihen vorhersagen (Statistik I): Exponentielle Glättung und Holt-Winters
4. Zeitreihen vorhersagen (Statistik II): ARIMA-Modelle

## Tag 2

5. Einblick in andere Zeitreihenmodelle
6. Machine Learning für Zeitreihen: Überblick, Vorbereitung und Klassifikation
7. Machine Learning für Zeitreihen: Clustering
8. Deep Learning für Zeitreihen (Einblick)



# Einblick in andere Zeitreihenmodelle

Session 5 (Dienstag 09:15 – 10:45)



# Einblick in andere Zeitreihenmodelle

- Zustandsraumdarstellung (State-space models): Ansatz, Pakete, Kalman-Filter, Anwendungsbeispiel
- Praxisbeispiel additives Regressionsmodell mit Facebook Prophet
- Bsp: Stromverbrauch vorhersagen unter Berücksichtigung von Saisonalität, Wochen- und Feiertagen

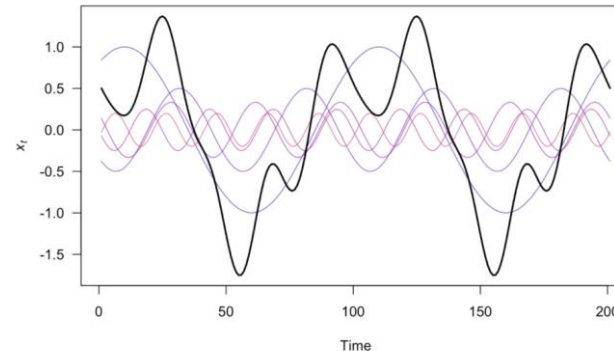
# Andere Ansätze zur Modellierung von Zeitreihen

## Frequenzanalyse (Fourieranalyse)

- Statt Änderungen der Amplitude *mit der Zeit*: Amplitudenänderung *mit der Frequenz*
- Funktionen können durch Summen von einfacheren trigonometrischen Funktionen dargestellt oder angenähert werden
- Fourier-Transformationen

### Beispiel

Unsicherheit in der Erzeugung von Windstrom durch Abschneiden der Hochfrequenzkomponente zuverlässiger modellieren (Oh & Son, 2018)



$$f(t) = \sum_{k=1}^5 \frac{1}{k} \sin(2\pi kt + k^2)$$

Funktioniert gut für Identifizierung von periodischen, durch Rauschen verfälschten Signalen.  
Aber: inkonsistenter Schätzer für die meisten realen Datensätze

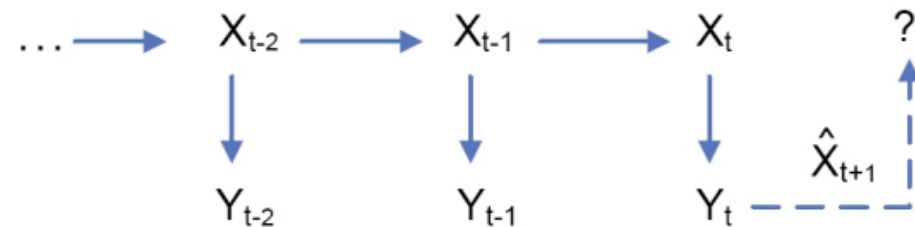
# Andere Ansätze zur Modellierung von Zeitreihen

## Zustandsraumdarstellung (State-Space-Models)

- Aus den Ingenieurwissenschaften: alle Beziehungen der Eingangs-, Ausgangs- und Zustandsgrößen in Matrizen und Vektoren dargestellt
- **Kalman-Filter:** Algorithmus mit Prädiktor-Korrektor-Struktur

### Beispiel

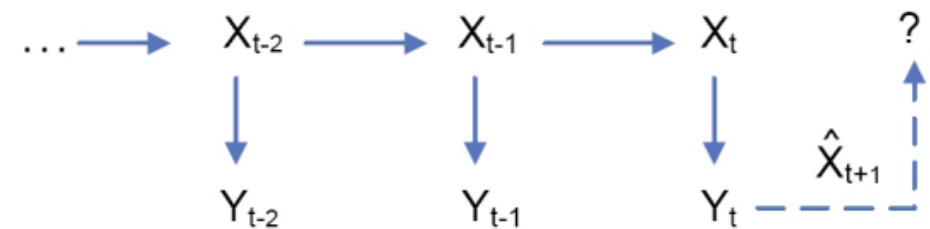
Sie interessiert die Innen-Raumtemperatur  $X_t$ , aber: Messung  $Y_t$  an einer Ecke des Raums mit einem Sensor



# Zustandsraumdarstellung (State-Space-Models)

## Zustandsraummodell (einfachste Form)

- Zustandsgleichung  $X_{t+1} = F_t X_t + W_t$
- Beobachtungsgleichung  $Y_t = G_t X_t + V_t$
- ( $W_t \sim \text{IID}(0, Q_t)$  und  $V_t \sim \text{IID}(0, R_t)$  weißes Rauschen)





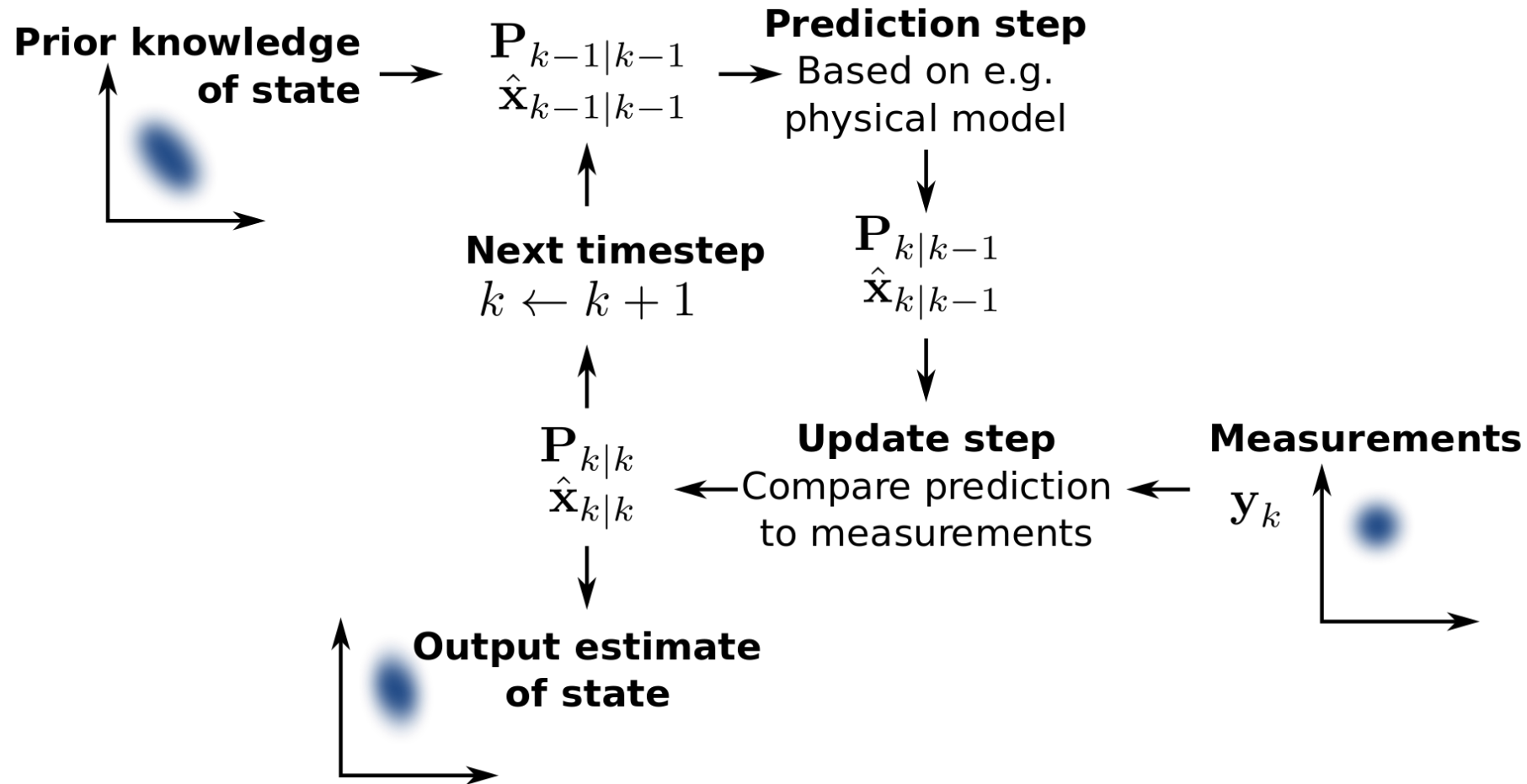
# Kalman Filter

- **Kalman-Filter** (populärster Zustandsraum-Algorithmus): Satz von Rekursionen, zur Prädiktion/ Schätzung, Vorteil: iterative Bauart (PrädiktorKorrektor-Struktur) – ideal für Echtzeitanwendungen

Funktionsweise:

1. auf Basis des vorherigen Zustands  $\hat{X}_{t-1|t-1}$ : Vorhersage wahrscheinlichster Zustand  $\hat{X}_{t|t-1}$  (Prädiktion)
2. Vergleich mit dem tatsächlich gemessenen Wert  $Y_t$
3. Differenz der beiden Werte linear gewichten
4. Damit Verbesserung (Korrektur) des aktuellen Zustandes  $\hat{X}_{t|t}$

# Kalman Filter



# Andere Ansätze zur Modellierung von Zeitreihen

## Facebook Prophet

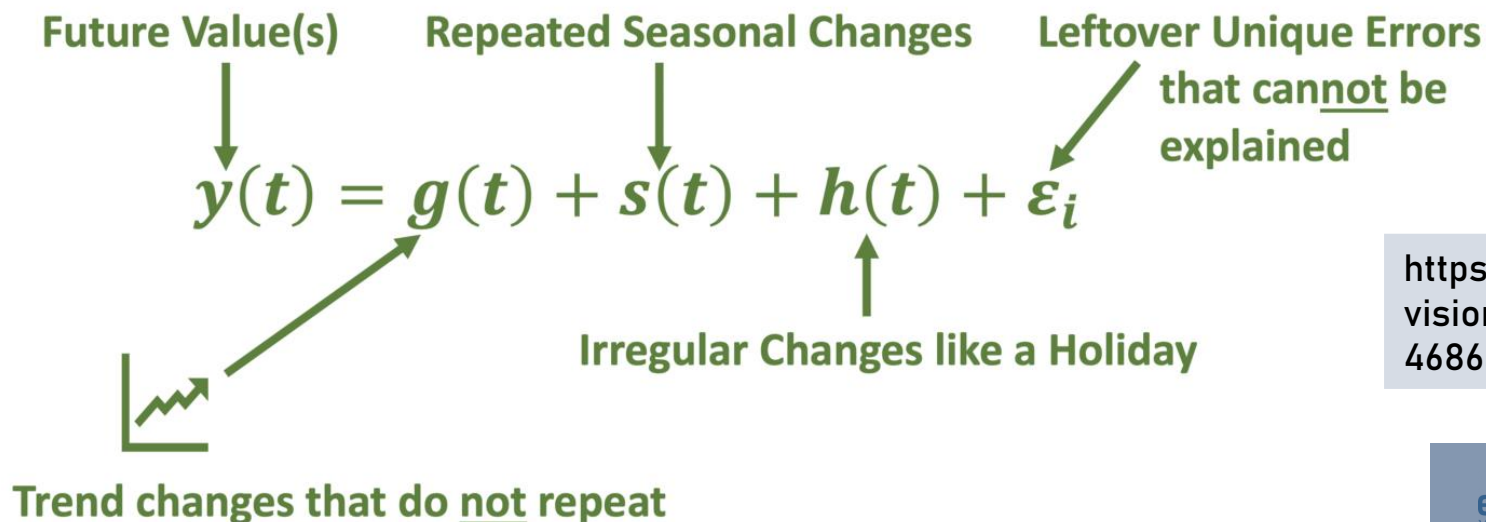
- Prophet ist ein Verfahren zur Vorhersage von Zeitreihendaten auf der Grundlage eines additiven Modells, bei dem nichtlineare Trends mit jährlicher, wöchentlicher und täglicher Saisonalität sowie Ferieneffekten angepasst werden.

Future Value(s)      Repeated Seasonal Changes      Leftover Unique Errors that cannot be explained

$$y(t) = g(t) + s(t) + h(t) + \epsilon_i$$

Irregular Changes like a Holiday

Trend changes that do not repeat



<https://medium.com/future-vision/the-math-of-prophet-46864fa9c55a>

# Growth $g(t)$

Das stückweise lineare Modell wird anhand der folgenden statistischen Gleichungen angepasst,

$$y = \begin{cases} \beta_0 + \beta_1 x & x \leq c \\ \beta_0 - \beta_2 c + (\beta_1 + \beta_2) x & x > c \end{cases}$$

wobei  $c$  der Punkt der Trendänderung ist („Changepoint“).

# Seasonality $s(t)$

- Die Saisonalitätsfunktion ist eine Fourier-Reihe in Abhängigkeit von der Zeit.

$$s(t) = \sum_{n=1}^N \left( a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right)$$



# Holidays $h(t)$

- Die Feiertagsfunktion ermöglicht es Facebook Prophet, die Vorhersage anzupassen, wenn ein Feiertag oder ein wichtiges Ereignis die Vorhersage verändern könnte.
- Sie nimmt eine Liste von Daten (es gibt eingebaute Daten von US-Feiertagen oder Sie können Ihre eigenen Daten definieren) und wenn jedes Datum in der Vorhersage vorhanden ist, addiert oder subtrahiert sie den Wert der Vorhersage von den Wachstums- und Saisonalitätsbedingungen, die auf historischen Daten zu den identifizierten Feiertagsdaten basieren.
- Man kann auch einen Bereich von Tagen um die Daten herum festlegen (z. B. die Zeit zwischen Weihnachten und Neujahr, Feiertagswochenenden, die Verbindung von Thanksgiving mit Black Friday/Cyber Monday usw.).

# Ab ins Jupyter Notebook

