Datenanalyse

Rohdaten von Sensoren können in den seltensten Fällen ohne Vorverarbeitung für maschinelles Lernen verwendet werden. In diesem Schritt wird diese wichtige Vorarbeit in Form einer umfassenden Datenanalyse behandelt.

Aufgabe 1: Daten laden und plotten

Zuerst wird die Datei `aussentemperatur_ucb.csv` mit den Rohdaten mithilfe der Funktion `readtable` geladen.

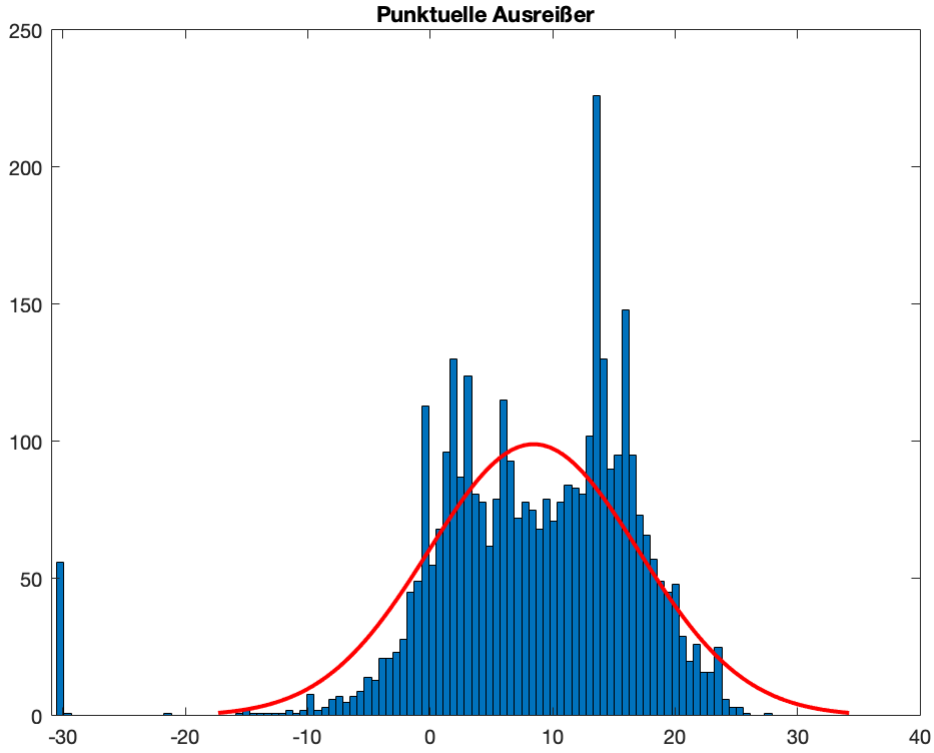
```
clearvars;
clc;

data = readtable('aussentemperatur_ucb.csv', 'Delimiter', ',', 'TreatAsEmpty', 'true');
dates = data{:, 1};
temperatures = data{:, 2};
```

Aufgabe 2: Punktuelle Ausreißer - Histogramm Plot

Ein Histogramm Plot mit einer Normalverteilungskurve eignet sich gut um punktuelle Ausreißer in Daten zu finden. Hierfür wird die Funktion `histfit` verwendet. Punktuelle Ausreißer können dann einfach an den Rändern des Histogramms gefunden werden. Die Ausreißer sollen dann als logische Indizes gespeichert werden.

```
figure
histfit(temperatures, 100, 'Normal')
hold on
title('Punktuelle Ausreißer')
hold off
```




```
punktuelleAusreisser = temperatures < -20;
```

Aufgabe 3: Kontextuelle Ausreißer - Zeitreihenzerlegung (Time Series Decomposition)

Für kontextuelle Ausreißer soll eine Zeitreihenzerlegung (Time Series Decomposition) durchgeführt werden. Bei dieser Methode wird jeder Wert eines Datensatzes in drei Komponenten (Trend T_t, Saisonwert S_t, Restfehler I_t) zerlegt, die entweder addiert (additive Methode) oder multipliziert (multiplikative Methode) den ursprünglichen Messwert ergeben. Da die multiplikative Methode besser zu Zeitreihen mit einem exponentiellen Wachstum passt, soll hier die additive Methode verwendet werden. Die Zerlegung ergibt sich also wie folgt: $y_t = T_t + S_t + I_t$

Bei der Bestimmung der Trendkomponente reicht es nicht nur einen Wert zu betrachten. Stattdessen müssen für jeden Wert auch mehrere Werte davor und danach betrachtet werden. Hier werden für jeden Wert immer auch die Werte der 15 Tage davor und danach betrachtet.

Die Trendkomponente ergibt sich wie folgt: $T_t = y \cdot \begin{bmatrix} p-1 \text{ mal} \begin{Bmatrix} \frac{1}{2p} \\ \frac{1}{p} \\ \vdots \\ \frac{1}{p} \\ \frac{1}{2p} \end{Bmatrix} \end{bmatrix}$

Hinweise:

- mit der Funktion `repmat(a,r,c)` kann eine Matrix der Größe `r,c` mit dem Element a erzeugt werden
- die Funktion `conv(u,v,'same')` gibt den mittleren Teil der Faltung u,v mit der Größe `u` zurück

```
l = size(temperatures, 1);
p0 = 15;
p1 = 2 * p0;
p2 = 2 * p1;
sW = [1/p2; repmat(1/p1, p1-1, 1); 1/p2];
yT = conv(temperatures, sW, 'same');
yT(1 : p0) = yT(p0+1);
yT(l-p0 : l) = yT(l-p0-1);
```

Die Saisonkomponente ergibt sich wie folgt: $S_t = \frac{1}{p} \cdot \sum_{i=t-\frac{p}{2}}^{t+\frac{p}{2}} T_i$

Hinweise:

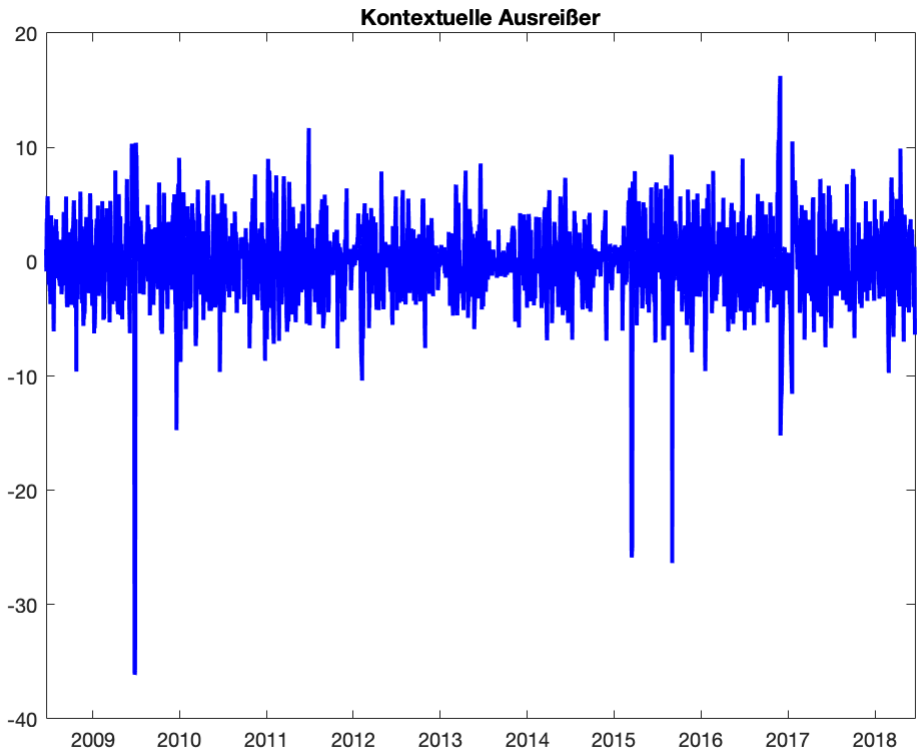
- den Tag einers Jahres liefert die Funktion `day(d, 'dayofyear')`
- mit der Funktion `cellfun(@x, f, c)` kann die Funktion `f` auf jedes Element von `x` angewandt werden welches die Bedingung `c` erfüllt

```
xT = temperatures - yT;
dates = data{:, 1};
sidx = cell(366, 1);
for i = 1 : 366
    sidx{i, 1} = find(day(dates, 'dayofyear') == i);
end
yS = cellfun(@(x) mean(xT(x)), sidx);
yS = table2array(timetable(dates, yS(day(dates, 'dayofyear'), 1)), 'ConvertRowTimes', false);
```

Anschließend wird der Restfehler durch Subtraktion der Trend- sowie der Saison-Komponente von den Temperaturwerten bestimmt: $I_t = y_t - T_t - S_t$

Die kollektiven Ausreißer können dann am Plot der Restfehlerwerte über die Zeitachse abgelesen werden. Die Ausreißer sollen dann als logische Indizes gespeichert werden.

```
dates = data{:, 1};
yI = temperatures - yT - yS;
figure
plot(dates, yI, 'b', 'Linewidth', 2)
hold on
title('Kontextuelle Ausreißer')
hold off
```



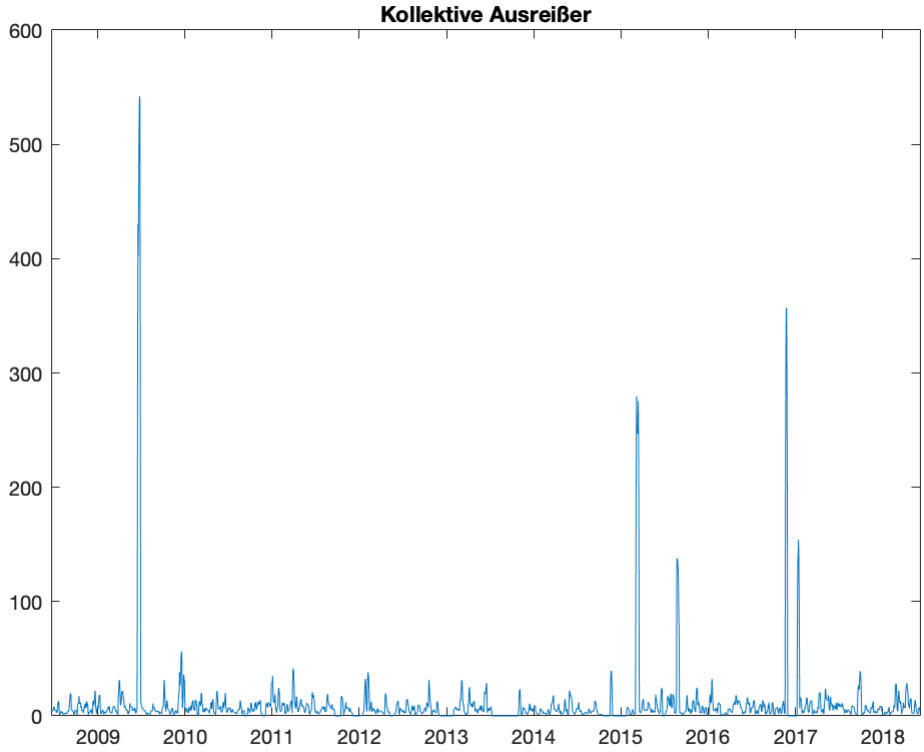

```
kontextuelleAusreisser = yI > 10 | yI < -10;
```

Aufgabe 4: Kollektive Ausreißer - Varianzanalyse

Kollektive Ausreißer können durch die Ermittlung der Varianz in einem festen Zeitfenster, welches schrittweise über die Daten bewegt wird, gefunden werden. Als Fenstergröße werden 11 Tage gewählt. Die Varianz kann mittels der Funktion `var` bestimmt werden. Anschließend können kollektive Ausreißer an einem Plot der Varianzwerte über die Zeitachse abgelesen werden. Die Ausreißer sollen dann als logische Indizes gespeichert werden.

```
dates = data{:, 1};
frame_size = 11;
for f = 1 : size(temperatures, 1) - frame_size
    y(f) = var(temperatures (f : f + frame_size - 1, 1));
end
dates(end - frame_size + 1 : end, :) = [];
temperatures(end - frame_size + 1 : end, :) = [];
```

```
figure
plot(dates, y)
hold on
title('Kollektive Ausreißer')
hold off
```




```
kollektiveAusreisser = (y < 1 | y > 40).';
```

Aufgabe 5: Ausreißer bereinigen und plotten

Zum Schluss sollen die zuvor gefundenen Ausreißer aus den Messwerten entfernt werden. Dann sollen die bereinigten Messwerte geplottet und mit den zu Beginn geplotteten Messwerten verglichen werden.

Hinweise:

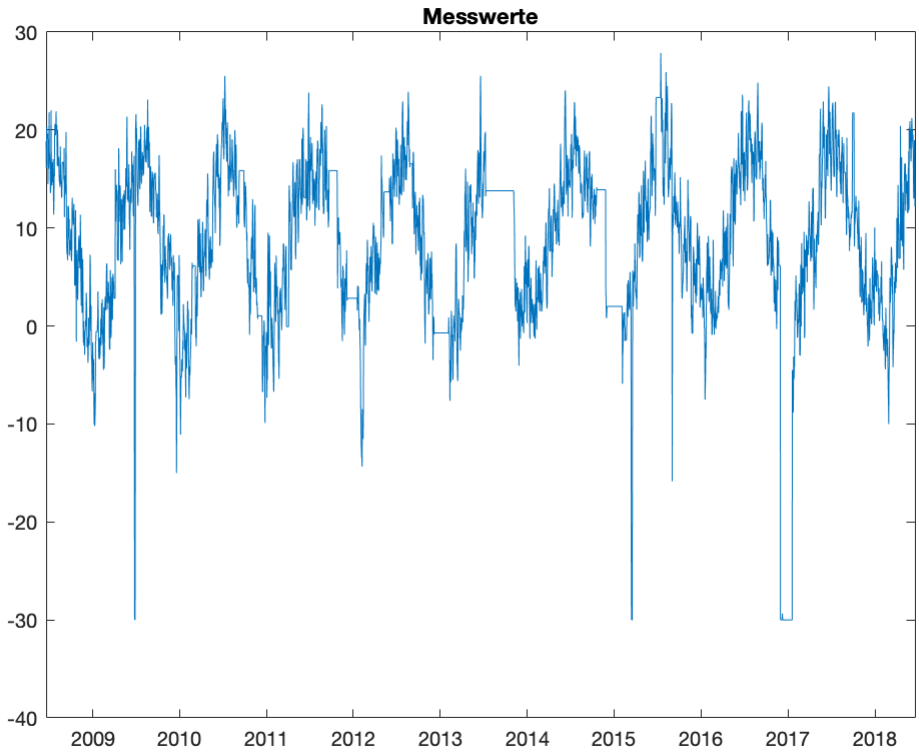
- Zum Löschen sollen die zuvor gespeicherten logischen Indizes verwendet werden.
- Außerdem müssen entsprechend der Fenstergröße der Varianzanalyse Daten am Ende entfernt werden.
- Um Lücken im Plot zu erzeugen kann die Funktion `cellfun(timetable, 'daily')` verwendet werden.

Konnten alle Ausreißer entfernt werden?

Antwort: Nein, auch statistische Verfahren haben keine hundertprozentige Trefferquote.

```
dates = data{:, 1};
temperatures = data{:, 2};

figure
plot(dates, temperatures)
hold on
title('Messwerte')
hold off
```




```
data(end-frame_size+1 : end, :) = [];
punktuelleAusreisser(end-frame_size+1 : end, :) = [];
kontextuelleAusreisser(end-frame_size+1 : end, :) = [];

disp("Anzahl Samples vor Bereinigung: " + size(data, 1))
```

```
Anzahl Samples vor Bereinigung: 3641
```

```
data(punktuelleAusreisser | kontextuelleAusreisser | kollektiveAusreisser, :) = [];
disp("Anzahl Samples nach Bereinigung: " + size(data, 1))
```

```
Anzahl Samples nach Bereinigung: 3107
```

```
data = retime(timetable(data{:, 1}, data{:, 2}), 'daily');

dates = data.Properties.RowTimes;
temperatures = data{:, 1};
```

```
figure
plot(dates, temperatures)
hold on
title('Bereinigte Messwerte')
hold off
```