Romana Grabner, BSc.

Anwendung statistischer Methoden zur Validierung von Messdaten aus der Wasserwirtschaft

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

Technische Mathematik: Operations Research und Statistik

eingereicht an der

Technischen Universität Graz

Betreuer

Univ.-Prof. Dr. Ernst Stadlober

Institut für Statistik

Graz, November 2017

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZ-online hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum	Unterschrift

Zusammenfassung

In dieser Arbeit wird ein Algorithmus zum Erkennen von untypischen zeitbezogenen Messdaten bzw. Ereignissen entwickelt, der auf verschiedenen in der angewandten Statistik verbreiteten Methoden basiert. Das Verfahren ist robust und liefert in annähernd Echtzeit gute Ergebnisse. Die Methode wird an realen Messdaten getestet, die vom Institut für Siedlungswasserwirtschaft und Landschaftswasserbau der Technischen Universität Graz zur Verfügung gestellt werden.
Zu Beginn der Arbeit wird eine explorative Analyse der Daten durchgeführt, dabei werden durch grafische Analysen zwei Kernprobleme identifiziert: fehlende Werte in den Messdaten und die große Datenmenge aufgrund der feinen zeitlichen Auflösung. Für die Reduktion werden die Daten in halbe Stunden gruppiert und die LOESS-Methode (eine nicht parametrische, lokale Regressionsmethode) für Daten mit saisonalem Muster erweitert und auf die Messdaten angewendet. Nach der Aufbereitung der Daten wird ein Verfahren zum Erkennen von Ereignissen implementiert. Es basiert auf der Zerlegung der Zeitreihe in ihre einzelnen Komponenten (den Trend, den saisonalen Anteil und den Zufallsanteil) durch die R-Routine `stl` und die lokale Analyse der Komponenten. Anschließend werden die gewonnenen Ereignislisten mit Listen bereits bestehender Methoden verglichen.

Abstract

In this thesis an algorithm is developed to identify untypical time series data or so called events. It is based on methods used in applied statistics and delivers robust results in almost real-time. The implementation is tested on real data provided by the Institute of Urban Water Management and Landscape Water Engineering, Graz University of Technology.
Through exploratory data analysis and graphical analysis of data two main issues are identified: a large amount of data and gaps in-between measurements. To reduce the huge amount of data it is binned to half hour intervals. To fill gaps in data the LOESS-method (a non-parametric local regression method) is extended for data with a seasonal pattern and applied to our data sets. The next step is an implementation of a method to identify events based on time series decomposition into a trend, seasonal component and a random component. This is carried out by the R-routine `stl` combined with a local analysis of the decomposition. Finally, the identified events are compared to results from other existing methods.

Inhaltsverzeichnis

1. Einleitung

Die folgende Einführung gibt einen Überblick über die Ausgangssituation und die Motivation für diese Arbeit. Dann werden die Zielsetzung sowie weitere Spezifikationen beschrieben. Anschließend werden Ereignisse definiert und ihre Eigenschaften charakterisiert. Zuletzt folgt eine allgemeine Beschreibung der verwendeten Messreihen. Die hier benutzten Fachbegriffe aus der Siedlungswasserwirtschaft basieren auf [Hofer 2012].

1.1. Ausgangssituation und Motivation

Das Institut für Siedlungswasserwirtschaft und Landschaftswasserbau der Technischen Universität Graz führt in ganz Österreich Messungen bei verschiedenen Frisch- und Abwassersystemen durch. Dabei werden beispielsweise verschiedene hydraulische Daten sowie Kennwerte für Wasserqualität erfasst, die die Basis für viele Simulationen bilden. Mit diesen Daten werden unter anderem theoretische Modelle von Kanalsystemen konstruiert. Da diese Daten die Grundlage für weitere Berechnungen bilden, sollten die Messdaten so plausibel wie möglich sein. Eine von vielen in diesem Fachbereich verwendeten Klassifizierungen der Plausibilität von Messdaten geht auf [Mourand und Bertrand-Krajewski 2002] zurück. Dabei werden die Messdaten in die folgenden drei Kategorien eingeteilt:

A Messdaten sind gültig validiert und können für weitere Zwecke verwendet werden,

B der Status der Messdaten muss noch validiert werden und weitere Analysen sind nötig sowie

C Messdaten sind ungültig und sollten in keinem Fall weiter verwendet werden.

Die Klassifizierung dieser drei Gesichtspunkte ist nach optischen Kriterien für ein geschultes Auge einfach. Eine händische Überprüfung der Daten erfordert allerdings einen erheblichen zeitlichen Aufwand und ist in der Praxis für große Datensätze nicht geeignet. Diese optischen Kriterien in mathematisch saubere und demnach von Algorithmen überprüfbare Kriterien umzuwandeln, ist generell schwierig. Daher erfolgt die Kategorisierung in der Praxis meist über eine versuchte Annäherung der optischen Kriterien durch eine Reihe von einfachen Abfragen. Einen Überblick über gängige Kriterien und ihre grafische Darstellung gibt die Abbildung 2-17 in [Hofer 2012]. Hier seien nur einige Beispiele genannt:

- Das Niveau der Messdaten springt bei mehreren aufeinanderfolgende Beobachtungen auf ein unterschiedliches Level.
- Die Differenz zwischen zwei aufeinanderfolgenden Messwerten wird sehr groß.

- Die Messwerte verlassen den physikalisch plausiblen Bereich (beispielsweise negativer in Liter gemessener Wert).
- Die Messwerte in einem bestimmten Bereich weisen eine übermäßig hohe Streuung bzw. Variabilität auf.
- Die Werte steigen oder sinken kontinuierlich über einen gewissen Zeitraum.

Wenn Grenzwerte für die Abfragen nötig sind, werden sie meistens global vorab festgelegt, zum Beispiel durch Erfahrungswerte aus bereits gültig validierten Messdaten. Bei [Hofer 2012] werden die Daten, die den maximalen bzw. minimalen Wertebereich der gültigen Daten aus den Vorjahren verlassen oder Messungen eine negative Durchflussrate aufweisen, als ungültig erkannt. Zusätzlich werden diese Kriterien bei [Hofer 2012] an das Auftreten von Niederschlag in Graz gekoppelt.

Diese Vorgangsweise versagt, wenn keinerlei Referenzwerte vorliegen oder grundlegende Parameter während des Messvorgangs verändert wurden, z.B. örtliche Umstellung oder Austausch von Messgeräten. Daher wäre eine Kategorisierungsmethode interessant, die sich der Datenlage adaptiv anpasst und keinerlei zusätzliche Informationen benötigt.

1.2. Analyse der Aufgabenstellung

Eine Kategorisierung der Gültigkeit nach exakten mathematischen Kriterien ist, wie im vorigen Abschnitt erläutert, nur bedingt möglich. Dennoch können einige allgemeine Aussagen getroffen werden. Wenn die Daten ausfallen, ist es einfach, diese der Kategorie ungültig zuzuordnen. Wenn die Daten vom ‚normalen' Verhalten abweichen, sind sie potentiell ungültig. Allerdings könnte es ebenfalls durch äußere Einwirkung auf die Daten zu einer Änderung im Verhalten kommen, wobei hier ein (menschliches) Eingreifen in die Messung als ungültig und die Wirkung einer erhöhten Niederschlagsmenge als gültig zu werten ist. Das heißt, im ersten Schritt ist es sinnvoll zu entscheiden, ob die Daten interessant bzw. potentiell ungültig sind. Diese als interessant ausgewählten Daten werden im weiteren Verlauf dieser Arbeit als Ereignisse bezeichnet.
Als nächstes ist zu klären, was ‚normales' Verhalten der Daten ist bzw. wie ‚weit' als interessant eingestufte Daten von den ‚normalen' Daten zumindest abweichen müssen.

1.3. Charakterisierung von Ereignissen

In dieser Arbeit wird ein Ereignis (engl.: Event) durch ein Abweichen vom Trockenwetter-Zustand der Daten charakterisiert. Die Daten sind im Trockenwetter-Zustand, wenn sich kein Niederschlag auf die Messwerte auswirkt, was die meiste Zeit der Fall ist. Somit wird ein Ereignis wie folgt definiert: Ein Ereignis einer Variablen ist ein Zeitraum, in dem ein oder mehrere hintereinander folgende Datenpunkte der Variablen vom Trockenwetterzustand (dem üblichen lokalen Muster der Daten) abweichen.

In dieser Arbeit soll kein Wert für das Erkennen von Ereignissen vorab global festgelegt werden, sondern er soll sich adaptiv den Daten anpassen. Im Ansatz der vorliegenden Arbeit werden die Ereignisse anhand von lokal untypischem Verhalten erkannt. Bei diesem Verfahren wird die Zeitreihe der Messdaten in einzelne Komponenten zerlegt und die Komponenten auf Auffälligkeit bezüglich Ereignissen getrennt analysiert.

Technische Kriterien für Ereignisse

Aus praktischen Gründen ergeben sich zwei technische Kriterien bezüglich der Dauer eines Ereignisses (siehe Tabelle 3-7 in [Hofer 2012]):

- Die Dauer eines Ereignisses soll nicht unter einer Stunde liegen.
- Ein Ereignis soll nicht mehr als 48 Stunden andauern.

1.4. Zielsetzung dieser Arbeit

Ziel dieser Arbeit ist es, einen einfachen Algorithmus zum Erkennen von potentiell ungültigen Daten bzw. sogenannten Ereignissen (siehe Abschnitt 1.3) zu finden, der auf verschiedenen in der angewandten Statistik verbreiteten Methoden basiert. Das Verfahren soll robust sein und in annähernd Echtzeit gute Ergebnisse liefern. Der Algorithmus soll an realen Messdaten getestet und evaluiert werden.

Weitere Spezifikationen

Der Algorithmus soll an einem ausgewählten Anteil der Messdaten getestet werden. Es werden vier Variablen aus dem Zeitraum 1. Januar 2009 bis 31. Dezember 2011 ausgewählt. Dieser Zeitraum wird gewählt, weil zu dieser Periode am Institut für Siedlungswasserwirtschaft und Landschaftswasserbau bereits Abschlussarbeiten zum Thema Validierung von Messdaten verfasst wurden und somit Erfahrungswerte und Referenzlisten von Ereignissen zur Verfügung stehen. Als zusätzliche Referenzwerte können die Niederschlagsmengen von zwei Wetterstationen aus Graz miteinbezogen werden.

1.5. Beschreibung der untersuchten Messreihe

Die konkreten Messdaten, die während dieser Masterarbeit verwendet werden, stammen von den Messstationen im Mischwasserüberlaufwerk CSO-R05, das an der Mur an der Bertha-von-Suttner Friedensbrücke im Süden von Graz liegt. In einem Mischwasserkanal wird Regenwasser sowie Abwasser gemeinsam gesammelt. Dabei wird zwischen einem Trockenwetterabfluss und einem Mischwasserabfluss unterschieden. Wenn sich im Mischwasserkanal zu viel Wasser auf einmal befindet, wird die überschüssige Menge über einen Überlauf direkt in die Mur geleitet. Daher kann es abhängig von der Menge und der Intensität des Regenwassers im Kanal zu einer signifikanten Änderung des Niveaus in den Messdaten kommen. Um über dieses Phänomen mehr Informationen zu sammeln, wurde der standardmäßige Messtakt von 3 Minuten ab dem Jahr 2009 bei großen Einflussmengen in den Mischwasserkanal auf 1 Minute erhöht. Ebenfalls aus diesem Grund können Niederschlagsmessungen in die statistischen Analysen miteinbezogen werden. Hier stammen die erhobenen Niederschlagsdaten aus der Karl-Morre Straße sowie Klusemanngasse in Graz. Der starke Zusammenhang zwischen diesen beiden Niederschlagsmessungen und den Messdaten in diesem Mischwasserüberlauf wurden bereits in anderen Abschlussarbeiten des Instituts für Siedlungswasserwirtschaft und Landschaftswasserbau untersucht, beispielsweise in [Hofer 2012].

Seit 2002 werden in dem Mischwasserüberlaufwerk CSO-R05 bis zu 75 verschiedene hydraulische Daten sowie Kenngrößen für die Wasserqualität gleichzeitig gemessen. Es werden beispielsweise die Temperatur, die elektrische Leitfähigkeit des Abwassers und die Menge des ein- sowie ausfließenden Wassers in den verschiedenen Kammern des Kanals erhoben. Für detaillierte Informationen zu Variablen der gesamten Messreihe sowie weitere Details zum Aufbau der Messanlage und zur physikalischen Funktionsweise der Messsonden siehe [Hofer 2012].

Es wurden folgende 23 Variablen durchgängig während der gesamten Messperiode von 2002 bis 2015 erfasst:

1	`time`	2	`codeq_inflow`
3	`codeq_inflow_quality`	4	`codfeq_inflow`
5	`codfeq_inflow_quality`	6	`cond_bypass`
7	`cond_bypass_quality`	8	`h_cso`
9	`h_sewer_inflow`	10	`h_sewer_overflow`
11	`no3neq_inflow`	12	`no3neq_inflow_quality`
13	`q_bypass`	14	`q_sewer_inflow`
15	`q_sewer_overflow`	16	`sac254_inflow`
17	`sac254_inflow_quality`	18	`sac436_inflow`
19	`sac436_inflow_quality`	20	`temp_cso_air`
21	`tsseq_inflow`	22	`tsseq_inflow_quality` und
23	`v_sewer_inflow_som`.		

Einschränkung der Daten

Um den Umfang der Arbeit einzuschränken, werden die Daten hauptsächlich im Zeitraum zwischen dem 1. Jänner 2009 und 31. Dezember 2011 analysiert. Außerdem werden nur die folgenden Variablen der Messreihe näher untersucht:

1 `time` (verwendet als Identifikator),

2 `codeq_inflow`,

8 `h_cso`,

9 `q_sewer_inflow` (auch: `q_inflow`) und

15 `q_sewer_overflow` (auch: `q_overflow`).

Zusätzlich werden aus den oben genannten Gründen noch Niederschlagsinformationen von zwei Messstationen in Graz (Karl-Morre Straße sowie Klusemanngasse) miteinbezogen:

- `hn_kamo` und
- `hn_klus`.

Eine detaillierte Beschreibung und statistische Analysen der verwendeten Variablen befindet sich in Abschnitt 2.1. Die restlichen Variablen spielen im weiteren Verlauf der Arbeit keine relevante Rolle und werden daher in den anderen Abschnitten nicht mehr genannt. Es können sämtliche Methoden, die in dieser Arbeit verwendet werden, mit kleinen Adaptionen auch auf andere Variablen angewendet werden.

2. Explorative Analyse der Daten

In diesem Kapitel finden sich grundlegende Informationen zu den verwendeten Daten, wie physikalische Bedeutung der Messwerte, ihre Einheit und statistische Basiskennwerte. Dazu gehören Werte wie beispielsweise Mittelwert, verschiedene Quantile, Minimum und Maximum und zeitliche Abhängigkeiten sowie Kennwerte zu Korrelationen. Es wird auf folgende sieben Variablen eingegangen: `time`, `codeq_inflow`, `h_cso`, `q_inflow`, `q_overflow`, `hn_kamo` und `hn_klus`.

2.1. Beschreibung der verwendeten Variablen

Die Variable `time` beschreibt Datum und Uhrzeit der Messungen und dient zur Identifizierung der einzelnen Beobachtungen. Sie wird R-intern als Anzahl der Sekunden seit dem 1. Jänner 1970 gespeichert (engl. Coordinated Universal Time, UTC). Die Variablen `h_cso`, `q_inflow` und `q_overflow` sind Maße für die Wassermenge in den verschiedenen Kammern des Kanals. Die Variable `h_cso` beschreibt die Wasserhöhe in Metern in der Mischwasserkammer. Der Wert der Variable `q_inflow` misst den Zufluss zur Mischwasserkammer. Die Variable `q_overflow` misst den Abfluss im Entlastungskanal, wenn der Mischwasserkanal nicht mehr die gesamte Abwassermenge fassen kann. Diese beiden Variablen werden in Liter pro Sekunde erfasst. Die Variable `codeq_inflow` wird in Milligramm pro Liter angegeben und ist ein Kennwert, der zur Bestimmung der Wasserqualität herangezogen wird. Er beschreibt ein Maß für die Menge der unter bestimmten Umständen oxidierbaren Stoffe im Kanalwasser. Die Variablen `hn_kamo` und `hn_klus` beschreiben die Höhe des Niederschlags in Millimetern pro Minute an den Wetterstationen in der Karl-Morre Gasse und in der Klusemanngasse in Graz.

Name	Einheit	Kurzbeschreibung
`time`		Datum und Uhrzeit der Messung in UTC
`codeq_inflow`	mg/l	Wasserqualitätsparameter (Äquivalenz-Konzentration des Summenparameters CSB)
`h_cso`	m	Wasserstand in der Mischwasserkammer
`q_inflow`	l/s	Abfluss im Zulauf zur Mischwasserkammer
`q_overflow`	l/s	Abfluss im Entlastungskanal
`hn_kamo`	mm/min	Niederschlagsmenge in der Karl-Morre-Straße
`hn_klus`	mm/min	Niederschlagsmenge in der Klusemanngasse.

Tabelle 2.1.: Beschreibung der verwendeten Variablen und deren Einheiten

2.2. Statistische Kenngrößen

In diesem Abschnitt werden zu den verwendeten Variablen folgende statistische Kenngrößen ermittelt:

- der Mittelwert m und Median med,
- das 1. und 3. Quartil ($q_{0.25}$ und $q_{0.75}$) sowie
- der minimale und maximale Variablenwert.

Die Variable `time` dient nur zur Identifikation der Beobachtung, daher werden keine statistischen Kennwerte ermittelt. Sämtliche Grafiken und statistischen Kennwerte werden nicht in der Auflösung der Originaldaten berechnet, sondern basieren auf der Einteilung der Daten in 30-Minuten-Intervalle. Somit werden in die Kennwerte 52555 Datenpunkten miteinbezogen. ($n = 52555$.)
Im Vergleich zur Berechnung der statistischen Werte mit Originaldaten liegen die Quartile, Mittelwert und Median nahe bei den hier berechneten Größen. Vor allem der maximale Wert unterscheidet sich hingegen stark je nach gewähltem Raster. Für eine weitere Begründung der Gruppierung der Daten siehe Abschnitt 3.3.

Variable	Wertebereich	Einheit	m (med)	1. | 3. Quartil
`codeq_inflow`	0.00 ... 1427	mg/l	468.7 (486.7)	327.4 | 592.3
`h_cso`	-0.5 ... 1.43	m	0.247 (0.239)	0.215 | 0.25
`q_inflow`	-4.7 ... 2482	l/s	57.47 (43.02)	27.10 | 53.9
`q_overflow`	-196 ... 2481	l/s	4.570 (0.469)	0.43 | 0.49
`hn_kamo`	0.00 ... 1.29	mm/min	0.0016 (0)	0 | 0
`hn_klus`	0.00 ... 1.33	mm/min	0.0014 (0)	0 | 0

Tabelle 2.2.: Verschiedene statistische Kenngrößen der Variablen aus den Jahren 2009 bis 2011.

Die Wertebereiche der einzelnen Variablen unterscheiden sich sehr stark. Die negativen Werte sind physikalisch bei keiner der Variablen zulässig und können durch Fehler bei der Messung zustande gekommen sein. Das 3. Quantil liegt bei allen Variablen sehr weit vom Maximalwert entfernt. Bei den meisten Variablen liegen Mittelwert und Median nahe beisammen, nur bei `q_inflow` und `q_overflow` ist dies nicht der Fall. Hier liegt der Mittelwert sogar über dem Wert des 3. Quartils. Bei den beiden Niederschlagsmengen ist auffällig, dass bei mindestens 75% der Messwerte kein Niederschlag gemessen worden ist.

Der Anteil der Tage mit Niederschlag im Zeitraum von 1. Jänner 2009 bis 1. Jänner 2012 liegt bei der Messstation in der Karl-Morre-Straße bei 36.4% und in der Klusemanngasse bei 35.7%. Der Anteil der halben Stunden, in denen Niederschlag auftritt, liegt bei der Karl-Morre-Straße bei 5.9% und in der Klusemanngasse bei 6.5%.

2.3. Abhängigkeiten der Variablen

Es gibt verschiedene statistische Methoden, Abhängigkeiten von Variablen zu untersuchen. In diesem Abschnitt werden folgende Abhängigkeitsmerkmale untersucht:

- die Kovarianzmatrix,
- die Korrelationsmatrix,
- die Autokorrelation und
- die partielle Autorkorrelation.

Die Kovarianzmatrix enthält die paarweisen Kovarianzen der Variablen, die ein Maß für den Zusammenhang zwischen zwei Variablen ist. Mit der Kovarianz kann keine Aussage über die relative Stärke des Zusammenhangs getroffen werden. Die Korrelation normiert die Kovarianz auf $(-1, 1)$ und ist daher ein relatives Maß für die lineare Abhängigkeit zweier Variablen, die zum selben Zeitpunkt aufgenommen werden. Die Kovarianz- und die Korrelationsmatrix sind symmetrisch. Die Autokorrelation und die partielle Autokorrelation sind Maße für zeitliche Abhängigkeit der Variablen zu sich selbst.

Mathematische Formeln

Die Formel zur Berechnung der Kovarianz von zwei Variablen X, Y lautet:

$$\mathrm{cov}_{X,Y} = \mathrm{cov}(X, Y) = E[(X - E(X)) \cdot (Y - E(Y))],$$

wobei $E(\cdot)$ den Erwartungswert bezeichnet. Die Korrelation wird durch die Skalierung der Kovarianz durch das Produkt der Varianzen $\mathrm{Var}(\cdot)$ berechnet:

$$\rho_{X,Y} = \frac{\mathrm{cov}_{X,Y}}{\sqrt{\mathrm{Var}(X) \cdot \mathrm{Var}(Y)}}.$$

In der Praxis sind die Größen $E(X)$ bzw. $\mathrm{Var}(X)$ einer Zufallsvariablen X nicht bekannt, daher werden diese beiden Werte in der Berechnung durch Schätzwerte ersetzt:

$$\hat{E}(X) = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ bzw. } \widehat{\mathrm{Var}}(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

wobei $\{x_1, \ldots, x_n\}$ die beobachteten Realisierungen der Variable X sind. Damit ergibt sich für die geschätzte bzw. empirische Kovarianz:

$$\widehat{\mathrm{cov}}_{X,Y} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

In der Praxis wird auch oft der unverzerrter Schätzer mit der Gewichtung $\frac{1}{n-1}$ statt $\frac{1}{n}$ verwendet.

Die empirische Korrelation lautet somit:

$$\hat{\rho}_{X,Y} = r_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x}) \sum_{i=1}^{n}(y_i - \bar{y})}}.$$

Für die geschätzte Korrelation wird in der Regel die Schreibweise $r_{X,Y}$ oder r verwendet.

Damit die Autokorrelation sinnvoll definiert werden kann, muss die Zeitreihe schwach stationär sein. Eine Zeitreihe X_t $(t = 1, \ldots, n)$ ist schwach stationär, wenn folgende drei Bedingungen erfüllt sind:

1. $E(|X_t|^2) < \infty \; \forall \; t \in \mathbb{Z}$,
2. $E(X_t) = \mu$ unabhängig von t und
3. $\text{cov}(X_s, X_t) = \text{cov}(X_{s+k}, X_{t+k}) \; \forall \; t, s, k \in \mathbb{Z}$.

Sei X_t $(t = 1, \ldots, n)$ eine schwach stationäre Zeitreihe, dann lautet die Autokorrelationsfunktion von X_t mit Verschiebung (engl. lag) h:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \; |\rho(h)| < 1,$$

wobei γ die Autokovarianzfunktion ist. Sie wird auf die folgende Weise berechnet:

$$\gamma(h) = \text{cov}(X_{t+h}, X_t).$$

Auch hier müssen die unbekannten Größen wie oben durch ihre Schätzwerte ersetzt werden. Somit lautet die empirische Autokovarianzfunktion

$$\hat{\gamma}(h) = \frac{1}{t} \sum_{i=1}^{t-h} (x_{i+h} - \bar{x})(x_i - \bar{x})$$

und die empirische Autokorrelationsfunktion

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} = \frac{\sum_{i=1}^{t-h}(x_{i+h} - \bar{x})(x_i - \bar{x})}{\sum_{i=1}^{t}(x_i - \bar{x})^2}.$$

2.3.1. Kovarianz und Korrelation

Wie oben erwähnt basieren die Berechnungen in diesem Abschnitt auf einer Einteilung der Daten in 30-Minuten-Intervalle. Die empirische Kovarianzmatrix $\widehat{\text{cov}}_{X,Y}$ beschreibt die Kovarianzen zwischen den Variablen `codeq_inflow`, `h_cso`, `q_inflow` sowie `q_overflow` in den Jahren 2009 bis 2011. Die Kovarianzen der beiden Variablen `hn_kamo` und `hn_klus` werden hier nicht berechnet, da der extreme Unterschied der Skalierung zu den anderen Variablen keinen sinnvollen Vergleich zulässt.

$$\widehat{cov}_{X,Y} = \begin{bmatrix} 32804.31 & -2.25 & -3304.57 & -1415.9 \\ . & 0.01 & 8.99 & 3.17 \\ . & . & 11330.08 & 5425.79 \\ . & . & . & 4098.81 \end{bmatrix}$$

Die Variable `codeq_inflow` hat zu den anderen drei Variablen einen negativen Zusammenhang. Die Variablen `h_cso`, `q_inflow` und `q_overflow` haben einen positiven Zusammenhang.

$$r_{X,Y} = \begin{bmatrix} 1 & -0.128 & -0.16 & -0.123 & -0.079 & 0.075 \\ . & 1 & 0.9 & 0.57 & 0.318 & 0.28 \\ . & . & 1 & 0.808 & 0.32 & 0.295 \\ . & . & . & 1 & 0.26 & 0.246 \\ . & . & . & . & 1 & 0.74 \\ . & . & . & . & . & 1 \end{bmatrix}$$

Die Variable `codeq_inflow` ist zu allen anderen leicht negativ korreliert ($-0.1 > r > -0.3$). Die Variablen der Wassermenge im Kanal `h_cso`, `q_inflow` und `q_overflow` sind hoch positiv korreliert ($r > 0.8$), nur die Korrelation der beiden Variablen `h_cso` und `q_overflow` liegt etwas darunter ($r \approx 0.6$). Die Niederschlagsvariablen `hn_kamo` und `hn_klus` sind wie erwartet hoch positiv korreliert ($r > 0.7$) und auch zu den Variablen der Wassermenge im Kanal `h_cso`, `q_inflow` und `q_overflow` leicht positiv korreliert ($0.2 < r < 0.4$).

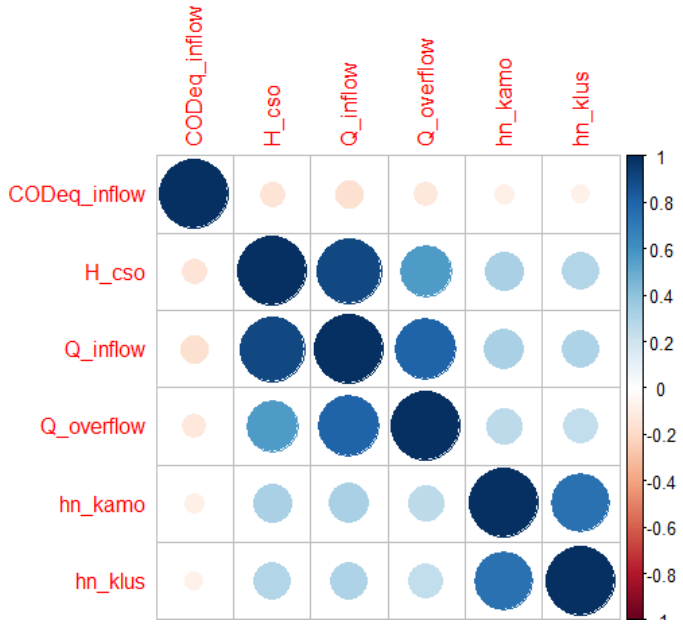

Abbildung 2.1.: Korrelation der Variablen von Jänner 2009 bis Dezember 2011.

2.3.2. Autokorrelation und partielle Autokorrelation

Zuerst ist zu überprüfen, ob die Zeitreihe schwach stationär ist. Dies kann durch einen Hypothesentest überprüft werden, zum Beispiel durch den Augmented Dickey-Fuller Test (ADF). Hier wird in der Nullhypothese überprüft, ob die Zeitreihe eine Einheitswurzel besitzt und somit nicht stationär ist. Für theoretische Ausführungen siehe [Brockwell und Davis 2002]. Die verwendete Implementierung der oben genannten Routine befindet sich in der R-Bibliothek `tseries`.

Variable	Statistik	p-Wert
`codeq_inflow`	-16.375	< 0.01
`h_cso`	-13.329	< 0.01
`q_inflow`	-14.351	< 0.01
`q_overflow`	-17.844	< 0.01

Tabelle 2.3.: R-Output des Augmented Fuller-Dickey Tests (AFD).

Der kleine p-Wert deutet darauf hin, dass die Nullhypothese verworfen werden kann und somit die hier untersuchten Variablen `codeq_inflow`, `h_cso`, `q_inflow` und `q_overflow` schwach stationär sind. Damit ist die Voraussetzung für eine sinnvolle Berechnung der Autokorrelationsfunktion erfüllt.

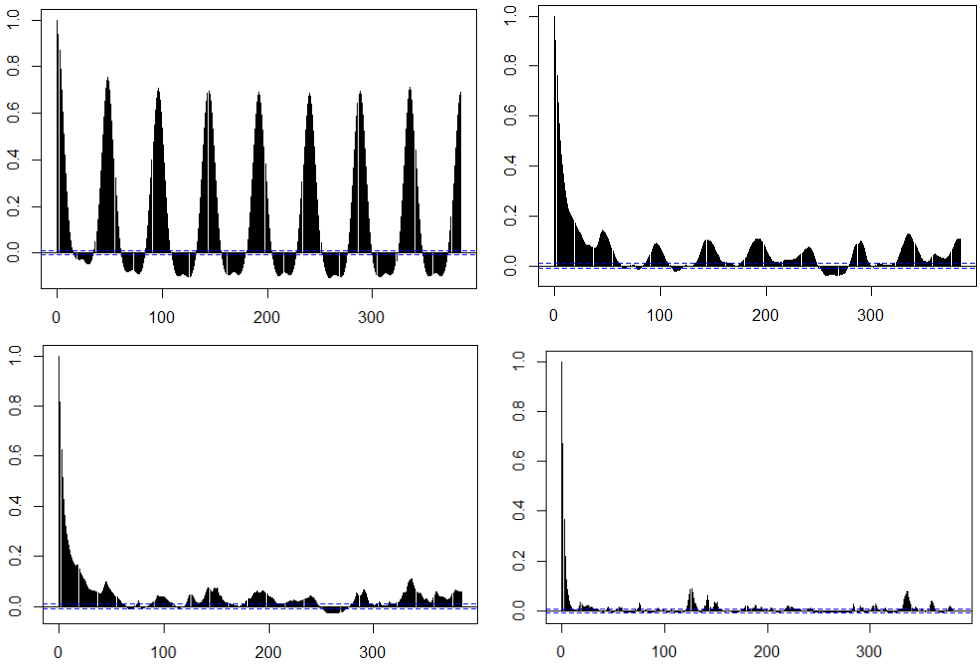

Abbildung 2.2.: Grafische Darstellung der Autokorrelation der Variablen `codeq_inflow`, `h_cso`, `q_inflow` und `q_overflow` (links oben bis rechts unten) über den Zeitraum von 8 Tagen im Jänner 2009. Die Verschiebung wird auf der x-Achse und die Korrelation auf der y-Achse dargestellt.

Die Autokorrelation und partielle Autokorrelation sind Maße für die zeitliche Abhängigkeit der Variablen zu sich selbst. Die Autokorrelation beschreibt die Korrelation der Variablen zu sich selbst, wenn sie um h Positionen verschoben wird und beginnt bei Verschiebung (engl. lag) 0. Die partielle Autokorrelation beschreibt wieder die Korrelation der Variablen zu sich selbst um h Positionen verschoben, allerdings wird der Anteil der Korrelation der dazwischenliegenden Werte abgezogen. Sie beginnt bei Verschiebung 1.
Die Berechnungen in diesem Abschnitt basieren auf einer Einteilung der Daten in 30-Minuten-Intervalle. Daher entspricht eine Verschiebung der Position (engl. lag) um eine Einheit, einer zeitlichen Verschiebung um 30 Minuten. Das heißt eine Verschiebung um 48 Einheiten ($h = 48$) entspricht einem zeitlichen Unterschied von 24 Stunden.

Bei den Variablen `codeq_inflow`, `h_cso` und `q_inflow` gibt es bei der Autokorrelation alle 48 Datenpunkte bzw. 24 Stunden einen Peak (vgl. Abb. 2.2). Bei der Variable `codeq_inflow` ist dieser Peak am stärksten ausgeprägt. Diese Spitzen in den Daten lassen auf ein ausgeprägtes tägliches Muster schließen. Im Gegensatz dazu verläuft die Autokorrelation bei der Variable `q_overflow` sehr flach und es ist kein periodisches Muster vorhanden. Bei allen Variablen fällt die Autokorrelation sehr rasch ab.

Bei der partiellen Autokorrelation der Variablen `h_cso` und `q_inflow` sind nur die Werte mit Verschiebung $1, 2$ sowie 3 auffällig (siehe Abb. 2.3). Die partielle Autokorrelation der Variable `q_overflow` wird nicht geplottet, da bereits die Autokorrelation unauffällig ist.

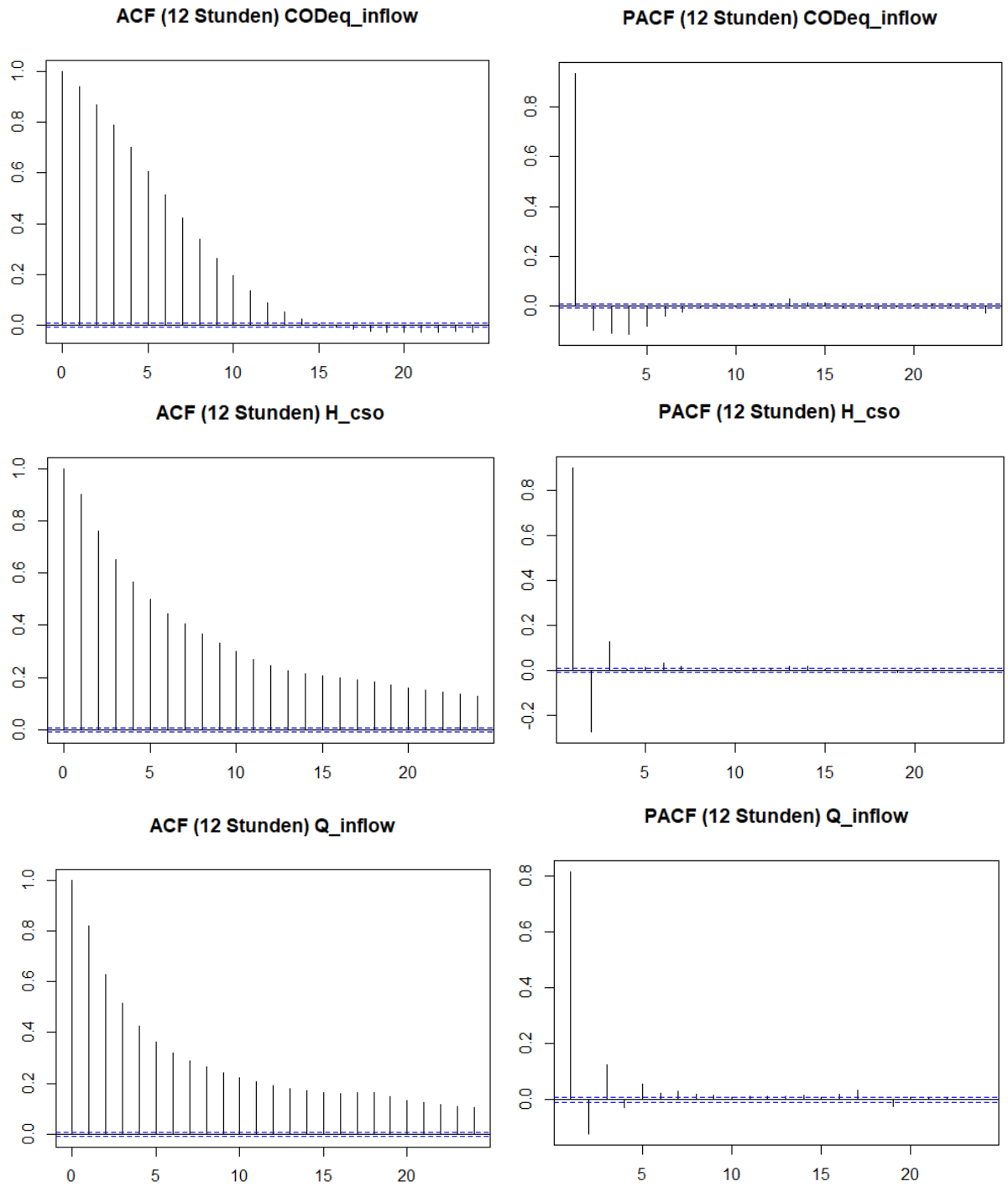

Abbildung 2.3.: Grafische Darstellung der Autokorrelation (links) und der zugehörigen partiellen Autokorrelation (rechts) der Variablen `codeq_inflow`, `h_cso` und `q_inflow` über einen Zeitraum von 12 Stunden im Jänner 2009. Die Verschiebung wird auf der x-Achse und die Korrelation auf der y-Achse dargestellt.

3. Grundlagen zur Behandlung von Messdaten

In diesem Kapitel wird auf einige grundlegende Aspekte hingewiesen, die während dieser Arbeit beim Umgang mit den Messdaten aufgetreten sind. Außerdem werden Methoden beschrieben, um diese Punkte zu behandeln. Auch wenn sich die genannten Strategien auf diese konkrete Situation beziehen, sind sie auch in anderen praktischen Fällen mit geringen Adaptionen anwendbar. Auf folgende Punkte wird in diesem Kapitel näher eingegangen: die Datenverwaltung im Allgemeinen, der Umgang mit Messausfällen und Strategien für den Umgang mit großen Datenmengen.

3.1. Allgemeines zur Datenverwaltung

Bei jedem Projekt mit Messdaten stellt sich zuerst die Frage nach einem einheitlichen und gut lesbaren Datenformat. Zusätzlich werden in dieser Arbeit zeitlich äquidistante Daten benötigt, um gängige Verfahren aus der Zeitreihenanalyse anwenden zu können.

Die Rohdaten sind hier im `.txt` oder `.csv` Format. Mittels eines Python Skripts wurden die Daten in `.rds` Format umgewandelt. Dafür wird das Interface `rpy2` [pyr 2017] verwendet, das R Funktionen[1] aus Python aufruft. Das R-interne Datenformat `.rds`, das für die Speicherung eines einfachen Objekts verwendet wird, ermöglicht eine schnelle und einfache Verarbeitung der Daten. Die Implementierung wird in Anhang A.1 beschrieben.

Die weitere Datenverarbeitung erfolgt mit R, da die Palette an vorprogrammierten, statistischen Routinen sehr vielfältig ist und auch die weitere Handhabung großer Datenmengen benutzerfreundlich ist. Vor allem sei hier die Bibliothek xts [Ryan u. a. 2017] erwähnt, die viele nützliche Funktionen für den Umgang mit Zeitreihen sowie die Manipulation von Zeitreihen implementiert hat.

Zum Schluss noch eine allgemeine Anmerkung zum Konvertieren von Zeitdaten: Besonders sollte auf Kompatibilität von Sommer- und Winterzeit bei verschiedenen Datenquellen geachtet werden.

[1] R ist eine freie Statistik-Software, für weitere Informationen siehe [R 2017].

3.2. Umgang mit Messausfällen

Während der gesamten Messperiode kommt es immer wieder zu Ausfällen bei der Aufzeichnung der Messwerte. Dabei treten die Ausfälle in zwei Varianten auf:

1. Die gesamte Messung fällt aus. Es wird keiner Variablen ein Wert zugewiesen und der Zeitpunkt der Messung wird nicht aufgezeichnet.
2. Mindestens einer Variablen wurde kein Wert zugewiesen. Bei dieser Art des Ausfalls wird zumindest immer der Zeitpunkt der Messung aufgezeichnet. Die fehlenden Messwerte einer Variablen werden durch zwei Platzhalter, NA oder NAN, beschrieben. NAN steht für „not a number" und kommt von einer ungültigen arithmetischen Operation wie beispielsweise $\frac{0}{0}$. NA steht für „not available" und bedeutet, dass der Messung kein Zahlenwert zugeordnet werden konnte. Allerdings ist dies ein allgemeinerer Begriff und wird als eine Überkategorie von NAN verwendet.

Ein vermehrtes Vorkommen von Typ 1 lässt vermuten, dass das Fehlen einer Variable den Ausfall weiterer Variablen zum selben Zeitpunkt begünstigt.

3.2.1. Ansätze zur Vervollständigung des Datensatzes

Ein möglichst vollständiger Datensatz ist eine essentielle Voraussetzung für die Anwendung der meisten gängigen Methoden. Es gibt viele verschiedene Varianten mit fehlenden Messwerten umzugehen. In der Praxis häufig verwendet werden

- auf Interpolation basierende Methoden,
- Modellierung aus den bestehenden Daten und
- das Ersetzen der fehlenden Werte durch einfache Verfahren. Zum Beispiel: Eine Lücke in den Daten wird durch den nächsten bekannten Wert ersetzt (Nearest-Neighbor-Methode).

In der simpelsten Methode werden alle Beobachtungen mit fehlenden Werten gelöscht. Der Vorteil dieser Variante ist, dass sie schnell und einfach angewendet werden kann. Allerdings gehen dabei viele Informationen verloren und im schlimmsten Fall wird die gesamte Messung unbrauchbar. Ein weiterer intuitiver Ansatz ist, einen fehlenden Messwert durch Interpolation der benachbarten bekannten Messwerte zu bilden. Auch diese Methode ist schnell und ohne allzu großem Aufwand umsetzbar. Ein Nachteil dieses Ansatzes ist, dass bei zu großen Lücken oft die lokale Struktur in den Daten zerstört wird. Außerdem werden keinerlei Varianz- bzw. Abhängigkeitsstrukturen zwischen den Variablen berücksichtigt. Hierfür gibt es einige multivariate Verfahren, die aber wieder höheren Rechen- und Speicheraufwand benötigen. Diese Effekte können vernachlässigt werden, wenn der Anteil der fehlenden Werte im Vergleich zu den vorhandenen relativ klein ist, da die ersetzten Werte in der gesamten Messreihe nur eine kleine Rolle spielen. Eine weitere Möglichkeit ist, fehlende Werte mittels eines ARIMA-Modells aus den bestehenden Daten zu prognostizieren.

3.2.2. Die LOESS-Methode

In dieser Arbeit wird eine auf Interpolation basierende Methode verwendet. Die LOESS-Methode ist eine nicht parametrische, lokale Regressionsmethode und geht auf [Cleveland 1981] zurück.
Die LOESS-Methode ist eine Verallgemeinerung der LOWESS-Methode (Englisch für locally weighted scatterplot smoothing). Bei diesem Verfahren werden die Daten in der Lücke nur mittels ihrer vorhandenen Nachbarn interpoliert. In den beiden folgenden Abschnitten werden zuerst die mathematische Theorie und im Anschluss die Anwendung auf die Messdaten beschrieben.

Mathematische Theorie

Die folgende Zusammenfassung der Beschreibung der theoretischen Aspekte stammt aus [Cleveland u. a. 1990].
Seien $\{x_1, \ldots, x_n\}$ und $\{y_1, \ldots, y_n\}$ zwei Gruppen von Beobachtungen. In der Regressionsanalyse wird x als die unabhängige (engl. independent) und y als die abhängige (engl. dependent) Variable oder Response bezeichnet. Es soll ein Schätzer $\hat{g}(x)$ von y konstruiert werden. $\hat{g}$ kann für beliebige Werte von x berechnet werden und ist nicht an die tatsächlichen Beobachtungen x_i $(1 \leq i \leq n)$ gebunden. (Die LOESS-Methode kann Regressionen auch für mehrdimensionale x berechnen. In dieser Arbeit ist die univariate Variante ausreichend.)
$\hat{g}(x)$ wird auf die folgende Weise konstruiert: Wähle eine beliebige positive ganze Zahl für q. Zunächst betrachte den Fall $q \ll n$. (q entspricht der Anzahl der gewählten Nachbarn. Wenn q nahe bei n liegt werden fast alle Beobachtungen zur Berechnung der Regression verwendet und der lokale Effekt dieser Methode geht verloren.) Die q zu x nächsten Werte der x_i werden ausgewählt und diesen Werten wird ein Gewicht basierend auf der Distanz zu x zugeordnet. Sei $\lambda_q(x)$ die Distanz der q-weitesten Beobachtung x_i zu x. Sei W eine Gewichtsfunktion:

$$W = \begin{cases} (1-u^3)^3 & \text{für } 0 \leq u < 1 \\ 0 & \text{für } 1 \leq u. \end{cases}$$

Das Nachbarschaftsgewicht (engl. neighborhood weight) lautet dann:

$$\vartheta_i(x) = W\left(\frac{|x_i - x|}{\lambda_q(x)}\right).$$

Daher haben die zu x nächsten x_i die größten Gewichte.
Im nächsten Schritt wird ein Polynom von Grad d mit Gewichten $\vartheta_i(x)$ bei (x_i, y_i) konstruiert. Dieses lokal gefittete Polynom entspricht an der Stelle x dem Wert $\hat{g}(x)$.

Als nächstes zum Fall $q > n$. Die Gewichte $\vartheta_q(x)$ werden wie folgt definiert:

$$\vartheta_q(x) = \frac{q}{n}\vartheta_n(x).$$

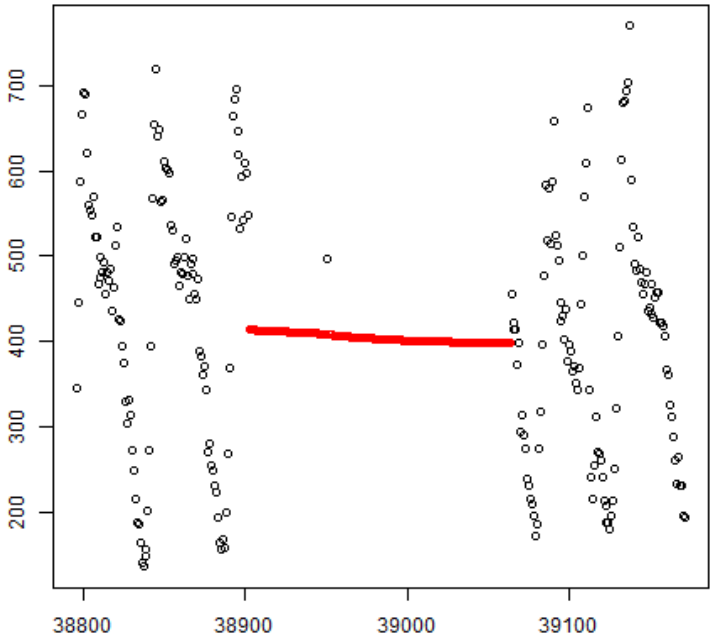

Abbildung 3.1.: Anwendung der Standard-LOESS Methode auf eine Variable mit täglichem Muster: die Interpolation wirkt nicht plausibel.

Die restliche Konstruktion erfolgt wie im ersten Fall.

Zum Schluss einige Anmerkungen zur Wahl von q und d. Für die meisten praktischen Anwendungen ist es ausreichend, ein Polynom mit Grad $d = 1$ für Daten mit einer leichten Krümmung oder $d = 2$ für Daten mit stärkerer Krümmung auszuwählen. Für die Wahl von q gilt, je größer q ist, desto glatter wird die Funktion $\hat{g}(x)$. Wenn also q gegen unendlich strebt, streben die Gewichte $\vartheta_i(x)$ gegen 1 und $\hat{g}(x)$ entspricht einem Interpolationspolynom von Grad d, dass die Quadratsumme minimiert. Die praktische Anwendung auf die Daten hat gezeigt, dass für q zweimal die Breite der Lücke ein sinnvoller Wert ist. Die Begründung folgt im nächsten Abschnitt.

3.2.3. Praktische Anmerkungen und Erweiterung der LOESS-Methode

In diesem Abschnitt werden einige praktische Anmerkungen zur LOESS-Methode beschrieben. Außerdem wird in dieser Arbeit eine nützliche Erweiterung der LOESS-Methode für die Anwendung bei Daten mit periodischen Muster entwickelt.
Ein interessanter Aspekt ist hierbei die Wahl der Fensterbreite, also die Anzahl der Nachbarn der Lücke[2], um eine akkurate Interpolation der Daten zu ermöglichen. Bei der Wahl der Breite des Interpolationsfensters sind folgende Punkte zu bedenken:

[2]Im theoretischen Teil entspricht diese Größe q.

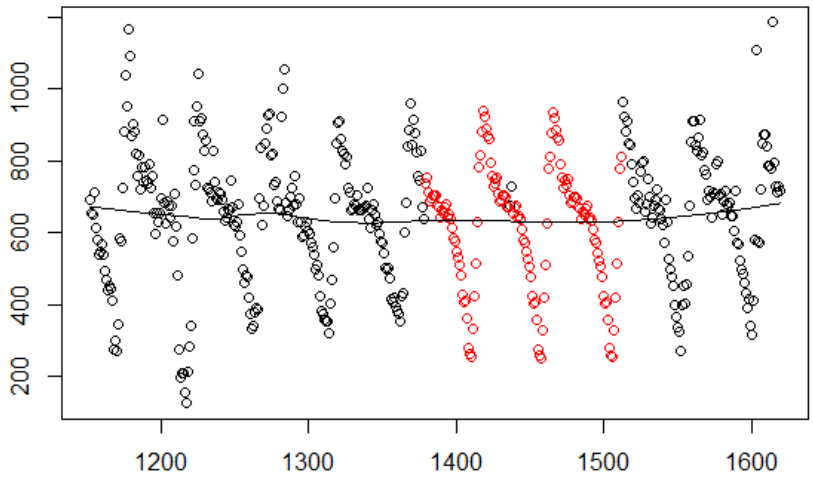

Abbildung 3.2.: Beispielhafte LOESS-Schätzung einer Variablen mit einem täglich wieder kehrenden Muster für eine Lücke (rot) von circa drei Tagen über einen Zeitraum von acht Tagen. Die schwarze Linie entspricht dem interpolierten Trend der Daten ohne das tägliche Pattern.

- Lokale Effekte der Daten sollen erhalten bleiben,
- nicht zu viele Ausreißer sollen in die Interpolation einfließen,
- der Rechenaufwand der Interpolation wächst mit der Fenstergröße und
- die Fenster verschiedener Lücken sollen sich möglichst nicht überlappen.

In der Praxis hat sich als Fensterbreite die zweimalige Größe der Lücke bewährt, die möglichst symmetrisch gewählt werden sollte. Eine natürliche obere Schranke für die Breite der füllbaren Lücken ergibt sich durch wiederkehrende Muster in den Daten: Beispielsweise sollten keine Lücken von Variablen mit einem täglich wiederkehrenden Muster, die größer als ein halber Tag sind, mit dem gewöhnlichen LOESS-Verfahren interpoliert werden. Ansonsten besteht die Möglichkeit, dass das tägliche Muster in den Daten verloren geht. (Zur Veranschaulichung siehe Abbildung 3.1.)
Um dieses Problem zu umgehen, wurde in dieser Arbeit die in R vorhandene LOESS-Routine erweitert, sodass es nun auch möglich ist, wiederkehrende Muster in den Daten miteinzubeziehen. Folgende Schritte wurden bei diesem Verfahren kombiniert: Die Daten werden über R-interne Funktionen in ihre Zeitreihenkomponenten zerlegt, das wiederkehrende Muster wird von den Daten abgezogen, auf die resultierenden Daten wird die LOESS-Schätzung angewendet und im letzten Schritt wird das wiederkehrende Muster wieder zu den geschätzten Daten addiert. In Abbildung 3.2 wird das Ergebnis dieser Vorgangsweise veranschaulicht. Im Anhang A.3 befindet sich die Implementierung der erweiterten Routine in R.
Durch das iterative Anwenden der Methode können auch größere Lücken in den Daten gefüllt werden. Nach jeder Iteration wird die Fensterbreite vergrößert, um so breitere Lücken

füllen zu können. Dies ist möglich, da kleinere Lücken bereits in den vorherigen Durchläufen gefüllt worden sind.

Anwendung der erweiterten LOESS-Methode auf die Daten

Die Reduktion der NA-Werte nach Anwendung der (erweiterten) LOESS-Methode wird in Abbildung 3.3 veranschaulicht. Bei den Variablen `codeq_inflow` (2), `h_cso` (3) und `q_inflow` (4) wurde die LOESS-Methode unter Einbeziehung des täglichen Patterns verwendet. Hier war es möglich, Lücken von bis zu zwei Wochen praktisch in Echtzeit authentisch zu füllen. Bei der Variable `q_overflow` (5) gab es kein wiederkehrendes Muster, hier wurden Lücken bis zu 5 Tagen mit der LOESS-Methode gefüllt.
Insgesamt konnte der Anteil der NA-Werte reduziert werden:

Bei der Variable `codeq_inflow`	von 20.9% auf 13.8%,	
bei `h_cso`	von 20.9% auf 14.5%,	
bei `q_inflow`	von 33.1% auf 26.6%	und
bei `q_overflow`	von 20.9% auf 17.9%.	

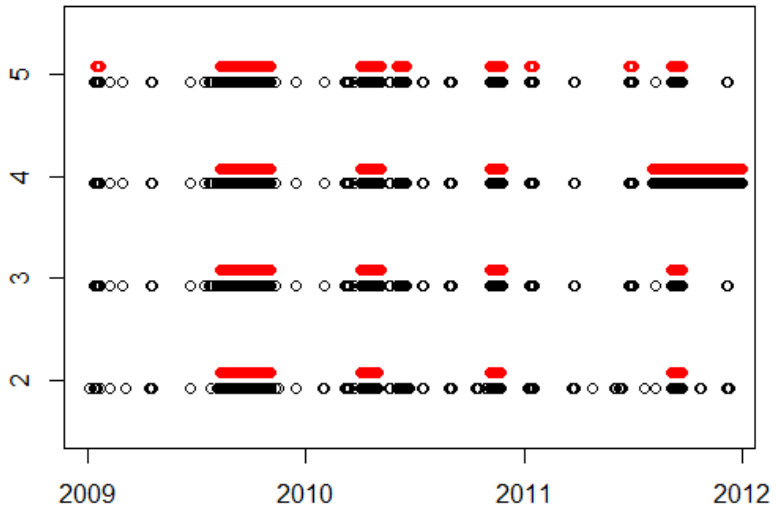

Abbildung 3.3.: NA-Werte der Variablen `codeq_inflow` (2), `h_cso` (3), `q_inflow` (4) und `q_overflow` (5) von 2009 bis 2011 vor (schwarz) und nach (rot) Anwendung der LOESS-Methode.

3.3. Strategien zum Umgang mit großen Datenmengen

Bei einer dreiminütigen Datenerfassung fallen für eine Variable pro Jahr 175 200 Beobachtungen an. Bei einer einminütigen Erhebung ergeben sich somit bis zu 525 600 Beobachtungen pro Jahr pro Variable.
Der originale Messtakt liefert eine sehr feine Auflösung der Daten. Damit können bereits geringfügige Änderungen in den Messungen erfasst werden, allerdings wird das gesamte System anfälliger für kurzfristige Schwankungen in den Daten. Zusätzlich fällt auch ein hoher Speicheraufwand an bzw. wird für Analysen und weitere Bearbeitung der Daten der Rechenaufwand hoch. Auch aus praktischer Sicht wird nicht immer eine sehr feine Auflösung der Daten benötigt. Um sich beispielsweise Jahrestrends anzusehen, reicht meistens ein sehr grobes Raster.
Aus oben genannten Gründen ist es hilfreich, die Daten in Gruppen bzw. sogenannte Klassen einzuteilen. Dabei werden Messdaten aus einem bestimmten zeitlichen Intervall durch ihren Mittelwert repräsentiert. Um die einzelnen Gruppen unabhängig voneinander zu halten, dürfen sich die zeitlichen Intervalle der einzelnen Gruppen nicht überschneiden. Alternativ können auch gleitende Mittelwerte verwendet werden. Dieser Ansatz wird aber in dieser Arbeit nicht weiterverfolgt, weil dadurch die Unabhängigkeit zwischen den einzelnen Gruppen verloren geht. Es gibt mehrere Varianten den Identifikator pro Gruppe zu wählen: In dieser Arbeit wird immer der Startzeitpunkt des Intervalls als Identifikator angegeben. Ein weiterer positiver Nebeneffekt dieser Vorgangsweise ist, dass dadurch vereinzelte fehlende Messwerte eliminiert werden.

3.3.1. Grobes Raster

In der Praxis ist es oft üblich, Tagesmittelwerte als grobes Raster zu verwenden, um einen Überblick über die Eigenschaften der Daten zu bekommen. Durch diese Einteilung der Daten lassen sich generelle Trendentwicklungen und periodische Muster optisch einfach erkennen.
Bei keiner der Variablen in den Abbildungen der Tagesmittel (Abb. 3.4 und 3.5) lässt sich ein jährlich bzw. saisonal wiederkehrendes Muster erkennen. Auffällig ist ebenfalls, dass die Messungen über alle drei Jahre in einem ähnlichen Wertebereich bleiben. Die Variable `q_overflow` nimmt meistens den Wert 0 an. Außerdem lässt sich erkennen, dass die Tagesmittelwerte der Niederschlagsmengen an beiden Messstationen nahezu übereinstimmen. Nur an wenigen Tagen, meist im Herbst, liegen die Werte weiter auseinander. Außerdem wird an den meisten Tagen kein Niederschlag an diesen beiden Wetterstationen gemessen. Der Niederschlag konzentriert sich vor allen auf die Monate zwischen März und November. Vor allem der Winter 2011 scheint sehr trocken gewesen zu sein.

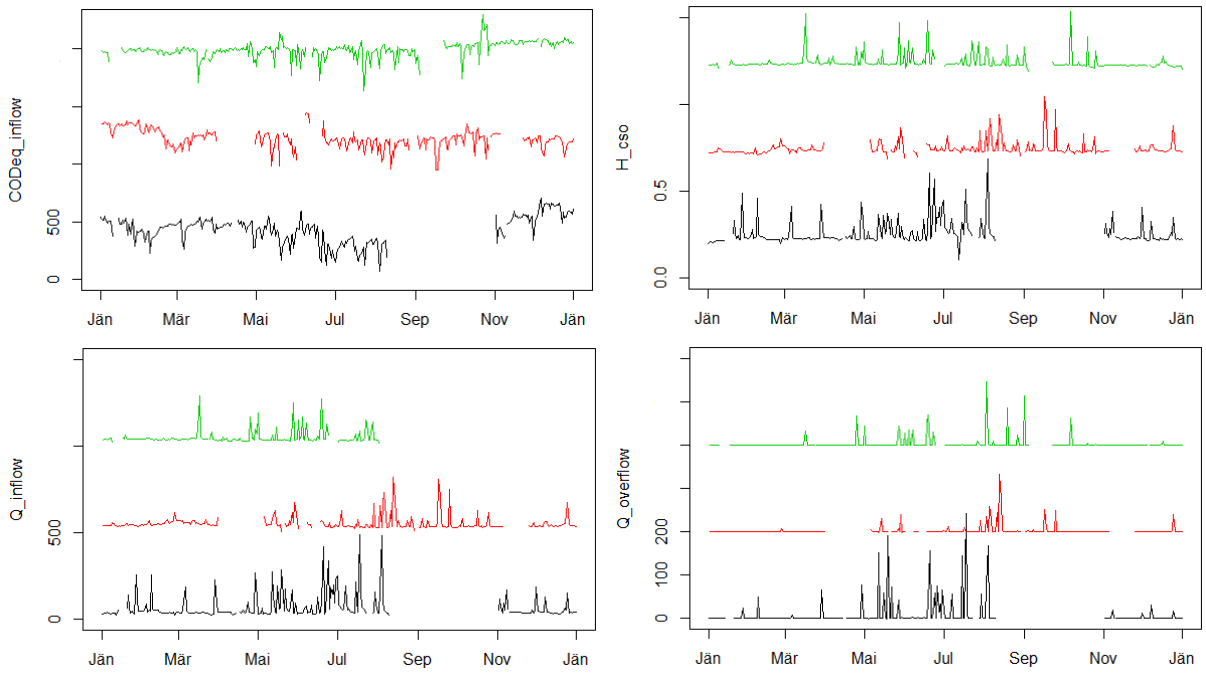

Abbildung 3.4.: Tagesmittelwerte von 2009 (schwarz), 2010 (rot) und 2011 (grün) der vier Variablen `codeq_inflow`, `h_cso`, `q_inflow` und `q_overflow`.

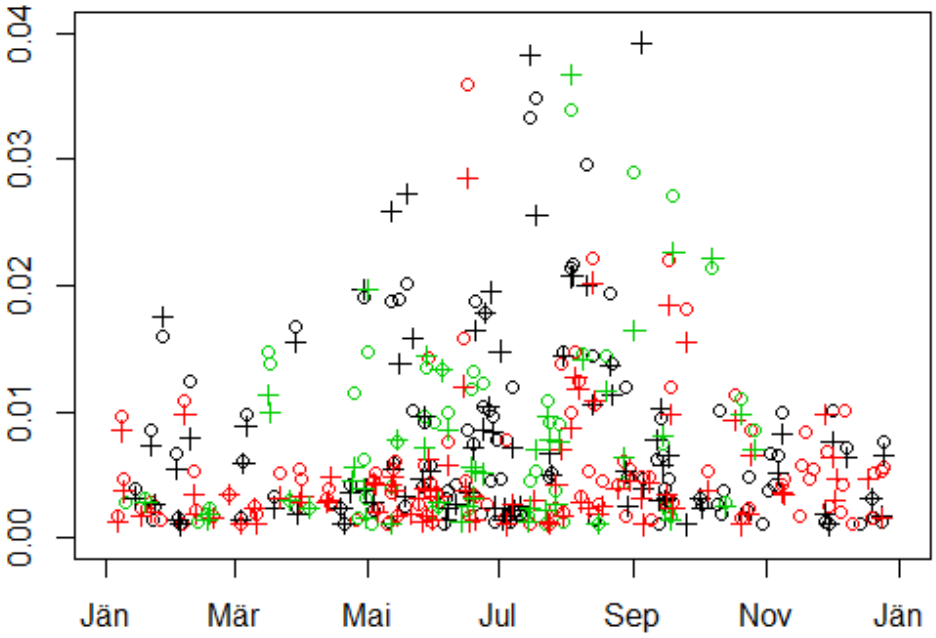

Abbildung 3.5.: Tagesmittelwerte der Niederschlagsmenge in mm/min mit einem Wert größer als 0.001 in der Klusemanngasse (+) und Karl-Morre Gasse (o) von 2009 (schwarz), 2010 (rot) sowie 2011 (grün).

3.3.2. Feines Raster

Für die feinere Einteilung der Daten wurden in dieser Arbeit Intervalle von $15, 30$ und 60 Minuten betrachtet. Eine Unterteilung in 45 Minuten würde sich prinzipiell ebenfalls anbieten, dieser Ansatz wurde aber aus praktischen Gründen nicht weiterverfolgt. Schon bei 15-Minuten-Intervallen wird die originale Datenmenge um Faktor 5 verkleinert.

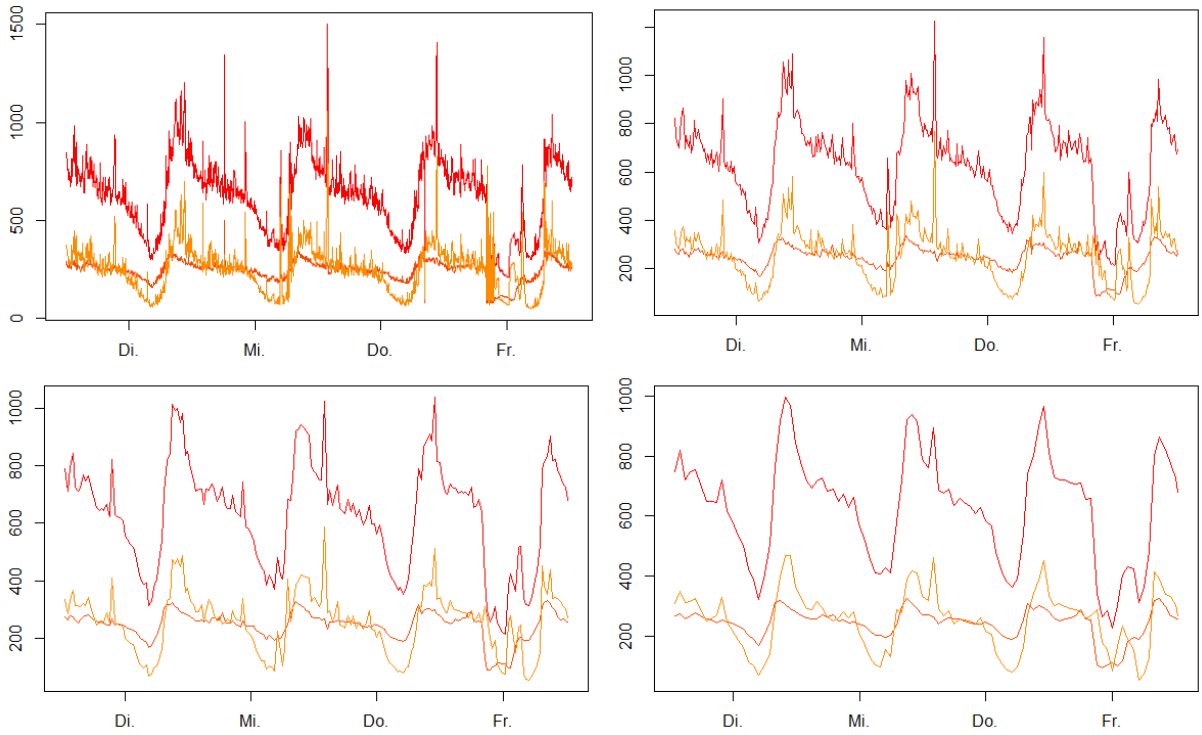

Abbildung 3.6.: Die Abbildungen zeigen beispielhafte Ausschnitte dreier `inflow`-Variablen vom 15. bis 18. März 2005: Vergleich von Auflösungen in 3-Minuten-Intervalle (Originaldaten), 15-Minuten-Intervalle, 30-Minuten-Intervalle, 60-Minuten-Intervalle (von links oben nach rechts unten).

Beim Vergleich zwischen den Originaldaten und den in 15 Minuten gruppierten Daten fallen extreme Spitzen weg, die vermutlich einmalige Ausreißer sind. Die Trends werden im 15- und 30-Minuten-Intervallen gut abgebildet und es lassen sich genügend Details erkennen. Der Trend ist im 60-Minuten-Intervall noch gut ablesbar. Allerdings stehen hier pro Tag nur noch 24 Datenpunkte zur Verfügung, was für die meisten detaillierten Analysen einem zu groben Raster entspricht.

3.3.3. Wahl des passenden Rasters

Welches Raster für die Daten gewählt wird, hängt von der weiteren Verarbeitung der Daten ab. Für grobe Trendanalysen und allgemeine Aussagen über die Daten reicht eine Analyse der Tagesmittelwerte. Für Simulationen oder die Anwendung der Daten in Modellen ist es vorteilhaft, ein grobes Raster zu wählen, da die Qualität von Modellen bei einer feinen Unterteilung durch Ausreißern und Peaks in den Daten leidet. In dieser Arbeit wird daher eine Auflösung in 30-Minuten Intervalle als repräsentativer Mittelweg verwendet. Alle Methoden können analog in feineren oder gröberen Rastern verwendet werden.

4. Erkennen von Ereignissen

Der erste Schritt um Daten zu validieren ist, potentielle Ereignisse zu erkennen. In Abschnitt 1.3 wird der Begriff Ereignis näher charakterisiert.
Es gibt verschiedene statistische Methoden, um Ereignisse zu erkennen. Zu den weit verbreiteten Methoden gehören beispielsweise: ARIMA-Modelle, Hauptkomponentenanalyse (engl. Principal component analysis, PCA), auf der T^2-Statistik basierende Analysen, verschiedene Regressionsansätze, Imputation oder andere auf Korrelationen basierende Modellierungen.

In dieser Arbeit wird eine Methode zum Erkennen von Ereignissen gewählt, die auf der Zerlegung der Zeitreihe in ihre einzelnen Komponenten basiert. Die Vorteile des gewählten Ansatzes sind:

- Es werden vorab keine Sollwerte festgelegt, sondern die Erkennung beruht auf dem lokalen Verhalten der Daten.
- Es werden keine meteorologischen Daten für die Auswertung benötigt.
- Verschiedene Kriterien können kombiniert werden.
- Die Methode kann einfach um weitere Kriterien erweitert werden.
- Die Methode erkennt Ereignisse fast in Echtzeit.

Ihre Nachteile sind:

- Die Methode ist hauptsächlich für Variablen mit wiederkehrenden Mustern geeignet, wie die Variablen `h_cso`, `q_inflow` und `codeq_inflow`.
- Es ist ein univariates Verfahren, das keine Beziehungen zwischen Variablen nützt.

Die Variablen des in dieser Arbeit vorliegenden Datensatzes haben eine starke Korrelation untereinander (vergleiche mit Abschnitt 2.3). Somit können, wenn das Verhalten einer Variablen bekannt ist, Aussagen über die anderen Variablen mit hoher Zuverlässigkeit getroffen werden.

Im folgenden Kapitel werden diese Inhalte beschrieben: die Grundlagen zur Zerlegung der Zeitreihe, die Details zur Implementierung, die mathematische Formulierung der verwendeten Kriterien zum Erkennen von Ereignissen sowie deren Anwendung auf die Messdaten und der Vergleich der hier gewonnenen Liste von Ereignissen mit bestehenden Methoden.

4.1. Allgemeines zur Zerlegung von Zeitreihen

Generell lassen sich Zeitreihen in drei Komponenten zerlegen: einen Trend (engl. trend), eine oder mehrere saisonale Komponenten (engl. seasonal component) und den Zufallsanteil (engl. random component oder auch remainder). Die hier verwendete Zerlegung kommt mit einer saisonalen Komponente aus, da nur Variablen mit einem täglichen wiederkehrenden Muster verwendet werden. Eine kurze Zusammenfassung der Theorie der Zeitreihen-Zerlegung folgt im nächsten Abschnitt.

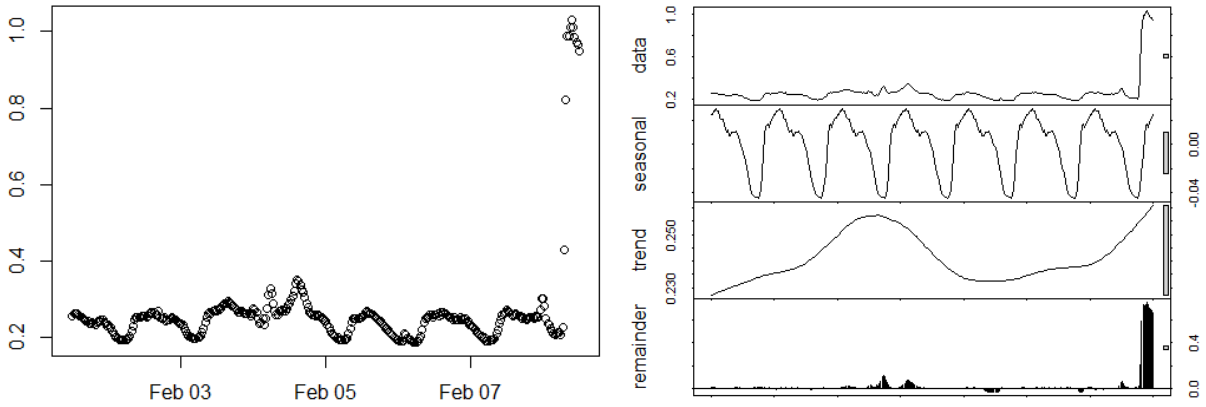

Abbildung 4.1.: Ausschnitt der Zeitreihe der Variablen `h_cso` im Februar 2009 (links) sowie deren Zerlegung mit der R-Routine `stl` (rechts). In der rechten Grafik sind die originalen Daten, die saisonale Komponente, der Trend und der Zufallsanteil (von oben nach unten) dargestellt.

4.1.1. Mathematische Theorie

In diesem Ansatz wird die Zeitreihe $X_t = m_t + s_t + R_t$ in verschiedene Komponenten zerlegt:

- den Trend m_t,
- die saisonale Komponente s_t und
- den sogenannten Zufallsanteil R_t.

Die folgende Definition einer klassischen Zerlegung einer Zeitreihe (engl. classical decomposition model) stammt aus [Brockwell und Davis 2002]. Sei X_t eine Zeitreihe und $t = 1, \ldots, n$ diskrete Zeitpunkte. Dann gilt:

$$X_t = m_t + s_t + R_t, \text{ wobei } \sum_{j=1}^{n} s_j = 0, s_{d+j} = s_j \text{ und } E(R_t) = 0.$$

Der Trend m_t ist eine sich langsam ändernde Funktion, die saisonale Komponente ist eine periodische Funktion mit Periode d.

Eine mögliche Vorgangsweise, um eine solche Zerlegung zu erhalten wird in [Brockwell und Davis 2002] wie folgt beschrieben: Angenommen es liegen die Beobachtungen der Zeitreihe $\{x_1, \ldots, x_n\}$ vor. Zuerst wird der Trend mittels eines endlichen gleitenden Mittelwert-Filters (engl. moving average filter) geschätzt, der so gewählt wurde, dass die saisonale Komponente eliminiert und die Zufallskomponente möglichst klein wird. Mathematisch formuliert lautet der Schätzer des Trends m_t, wenn die Periode gerade ist ($d = 2q, q \in \mathbb{N}$):

$$\hat{m}_t = \frac{1}{2q}(0.5x_{t-q} + x_{t-q+1} + \cdots + x_{t+q-1} + 0.5x_{t+q}),\ q < t \leq n - q.$$

Der Schätzer $\hat{m}_t$ lautet bei einer ungeraden Periode ($d = 2q + 1$):

$$\hat{m}_t = \frac{1}{2q+1}(x_{t-q} + x_{t-q+1} \cdots + +x_{t+q-1} + x_{t+q}),\ q < t \leq n - q.$$

Im zweiten Schritt wird die saisonale Komponente geschätzt.
Dabei werden für jedes $k = 1, \ldots, d$ die gemittelten Abweichungen von Daten zum Trend berechnet, also

$$w_k = \sum_{q<k+dj\leq n-q} x_{k+dj} - \hat{m}_{k+dj}.$$

Da sich die w_k nicht zwangsläufig auf 0 aufaddieren, lautet der Schätzer für die saisonale Komponente s_t:

$$\hat{s}_k = w_k - \frac{1}{d}\sum_{i=1}^{d} w_i.$$

Zusätzlich gilt $\hat{s}_k = \hat{s}_{k-d}$ für $k > d$.
Mit dem Schätzer $\hat{s}_t$ ergibt sich eine Schätzung der Originaldaten ohne die saisonale Komponente (engl. deseasonalized component):

$$d_t = x_t - \hat{s}_t,\ t = 1, \ldots, n.$$

Mittels der konstruierten Daten d_t wird der Trend erneut geschätzt. Hierfür gibt es eine Reihe an verschiedenen Ansätzen, beispielsweise über Differenzen, exponentielle Glättungen oder Polynomansätze. Alternativ kann wie im ersten Schritt ein endlicher gleitender Mittelwert-Filter (engl. finite moving average filter) angewendet werden.
Daraus ergibt sich schlussendlich der Schätzer für die Zufallskomponente R_t:

$$\hat{R}_t = x_t - \hat{m}_t - \hat{s}_t,\ t = 1, \ldots, n.$$

4.1.2. Die R-Routine `stl`

Die `stl`-Routine ist ein Verfahren zur Zerlegung von Zeitreihen in einen Trend, eine saisonale Komponente und eine Zufallskomponente. Sie verwendet iterativ die LOESS-Methode (siehe Abschnitt 3.2.2) und wird in [Cleveland u. a. 1990] erstmals beschrieben. Dieser Ansatz bringt mehrere Vorteile, die für diese Arbeit besonders bedeutsam sind:

- Das einfache Design erlaubt simple Anwendung und eine schnelle Berechnungszeit.
- Der Trend und die saisonale Komponente können robust trotz kurzfristigen, starken Verzerrungen der Daten geschätzt werden.
- Diese Methode kann mit fehlenden Werten in den Daten umgehen.[1]

Im Folgenden wird ein Überblick über den Aufbau und die wichtigsten Eigenschaften der `stl`-Routine gegeben. Die folgende Zusammenfassung wird gänzlich aus [Cleveland u. a. 1990] übernommen.

Die Routine besteht aus einer inneren und einer äußeren Schleife. In der inneren Schleife werden die saisonale Komponente und der Trend aktualisiert. Die äußere Schleife besteht aus einem Durchlauf der inneren Schleife sowie einer Aktualisierung der (robusten) Gewichte. Diese Gewichte werden verwendet, um den Einfluss von starken Schwankungen von Trend und saisonaler Komponente zu verringern. Die Gewichte werden auf 1 initialisiert.

Die innere Schleife

Die innere Schleife, hier beschrieben für den k-ten Durchlauf der äußeren Schleife, besteht aus den folgenden 6 Schritten:

1. Von den Daten wird der aktuelle Trend abgezogen (engl. detrending), also im k-ten Schritt wird $x_t - T_t^{(k)}$ berechnet.[2]
2. Die im vorigen Schritt berechneten Daten werden mittels der LOESS-Methode geglättet. Dadurch wird im k-ten Schritt eine temporäre saisonale Komponente $C_t^{(k)}$ konstruiert, die im Schritt 6 verbessert wird (engl. cycle-subseries smoothing).
3. Es wird ein Tiefpass-Filter auf die temporäre saisonale Komponente angewendet (engl. low-pass filtering of smoothed cycle-subseries). Dieser Filter besteht aus zwei hintereinander folgenden gleitenden Mittelwert-Filtern der Länge der Periode des wiederkehrenden Musters in den Daten sowie einem gleitenden Mittelwert-Filter der Länge 3. Anschließend werden die Daten wieder mit der LOESS-Methode geglättet. Diese in diesem Schritt berechnete Komponente $L_t^{(k)}$ wird im nächsten Schritt für die Berechnung der saisonalen Komponente in Schritt k verwendet.

[1] In dieser Arbeit wird diese Eigenschaft nicht mehr benötigt, da in einem vorigen Schritt die Daten mit der LOESS-Methode bis auf 14-tägige Lücken vervollständigt wurden.

[2] Der Trend wird beim ersten Durchlauf auf $\equiv 0$ initialisiert.

4. In diesem Schritt wird die endgültige saisonale Komponente konstruiert. Dafür werden die in Schritt 2 und 3 erhaltenen Konstruktionen benützt. Es wird $S_t^{(k)}$ berechnet (engl. detrending of smoothed cycled subseries).
5. Es wird von den Daten die saisonale Komponente (engl. deseasonalizing) abgezogen, also $S_t^{(k)} = C_t^{(k)} - L_t^{(k)}$.
6. Die in Schritt 5 erhaltenen Daten werden wieder mit der LOESS-Methode geglättet und die so erhaltenen Werte bilden den Trend im k-ten Schritt (engl. trend smoothing).

Die äußere Schleife

Die äußere Schleife besteht aus zwei Schritten:

1. einem Durchlauf der inneren Schleife, um den aktuellen Trend und die saisonale Komponente zu bestimmen und
2. die Berechnung der sogenannten Robustheits-Gewichte (engl. robustness weight).

Durch die innere Schleife werden Schätzer für Trend und saisonale Komponente berechnet. Damit lässt sich die übriggebliebene Zufallskomponente berechnen:

$$\hat{R}_t = x_t - m_t - s_t.$$

Im nächsten Schritt wird für jeden beobachteten Wert von x_t ein Robustheit-Gewicht berechnet.[3] Diese Gewichte geben ein Maß dafür, wie extrem der Wert der Zufallskomponente ist. Die folgende Konstruktion garantiert, dass das Gewicht sehr klein bzw. Null ist, wenn $|\hat{R}_t|$ sehr groß ist. Die Gewichte zum Zeitpunkt t werden folgendermaßen bestimmt:

$$\rho_t = B\left(\frac{|\hat{R}_t|}{h}\right),$$

wobei $h = 6$ median$(|\hat{R}_t|)$ und

$$B(u) = \begin{cases} (1-u^2)^2 & \text{für } 0 \leq u < 1 \\ 0 & \text{für } u > 1. \end{cases}$$

Anschließend wird die äußere Schleife erneut ausgeführt.

[3] Für fehlende Werte von x_t ist $\hat{R}_t$ nicht definiert und somit kann an diesen Stellen kein Gewicht berechnet werden.

4.2. Implementierung

Der implementierte Code befindet sich in A.4. Ziel dieser Implementierung ist, am Ende eine Liste mit Ereignissen abhängig von verschiedenen Kriterien auszugeben. Zu diesem Zweck wurden mehrere Funktionen in R implementiert, die in diesem Abschnitt beschrieben werden.

Hauptfunktion `pot_events`

Die Hauptfunktion `pot_events` erstellt Listen von potentiellen Ereignissen aufgrund von gewissen Kriterien. Die folgenden Kriterien sind derzeit in dieser Funktion implementiert:

1. Zu hohe Variabilität im Zufallsanteil,
2. ob die Datenpunkte außerhalb der Schwankungsbreite des täglichen Musters liegen,
3. zu große Änderungen im Trend und
4. ob Ereignisse in beiden benachbarten Fenstern erkannt werden.

Die ersten drei Kriterien basieren auf der Zerlegung der Zeitreihe. Das letzte Kriterium ist allgemeiner und kann nur in Kombination mit mindestens einem der ersten drei Kriterien angewendet werden. Eine detaillierte Beschreibung der verwendeten Kriterien befindet sich in Abschnitt 4.3. Die Liste der überprüften Kriterien kann bei Bedarf mit geringen Aufwand erweitert werden.[4]

Der Hauptfunktion können folgende Parameter übergeben werden:

- die zu untersuchenden Daten,
- welche Variable der Daten näher untersucht werden soll,
- die Fensterbreite der betrachteten Zerlegung (Standardeinstellung: eine Woche).

Aus der Fensterbreite ergibt sich direkt die Größe der Verschiebung (engl. shift) der Daten im nächsten Schleifendurchlauf. Es wird standardmäßig um die halbe Fensterbreite verschoben. (`shift` = `window_length`/2.) Dadurch kann die Übereinstimmung von Ereignissen in benachbarten Fenstern einfach überprüft werden.
Die Funktion eignet sich aufgrund des gewählten Ansatzes über die Zerlegung der Zeitreihe, insbesondere für Variablen mit einem wiederkehrenden Muster in den Daten. Ohne wiederkehrendes Pattern in den Daten kann nur Kriterium 1 und 3 verwendet werden.

Ablauf der Funktion `pot_events`

Zunächst werden einige grundlegende Datenstrukturen und Werte initialisiert, wie beispielsweise der Startindex des ersten Fensters, etc.

[4] Zum Beispiel könnten verschiedene Kriterien zur physikalischen Plausibilität eingebaut werden: Prüfung der Werte auf Nicht-Negativität oder auf Einschränkungen der Messgegebenheiten (z.B.:Kanalhöhe, maximales Fassungsvermögen der Wasserkammer).

Die Hauptfunktionalität besteht aus einer Schleife. Sie wird solange ausgeführt bis der Startindex des nächsten Schleifendurchlaufs kein gültiger Index innerhalb der Daten mehr ist. Als erstes wird überprüft, ob die Variable in dem aktuell betrachteten Fenster NAs enthält. Ist dies der Fall, wird das aktuelle Fenster übersprungen, weil bereits in einem Vorbereitungsschritt Lücken kleiner als 14 Tage mit der LOESS-Methode gefüllt wurden und somit bei den übriggebliebenen Lücken nahezu das ganze Fenster fehlt. Ansonsten werden die Daten im aktuellen Fenster in ihre Komponenten der Zeitreihe zerlegt: dem Trend, dem saisonalen Anteil sowie dem Zufallsanteil. Für die Zerlegung wurde in dieser Arbeit die R-Routine `stl` verwendet. Für Details dieser Methode siehe Abschnitt 4.1.2.
Als nächstes werden die für die Kriterien benötigten Werte berechnet und die Daten auf potentielle Ereignisse mittels der Kriterien überprüft. Am Ende des Schleifendurchlaufs werden Kontrollgrafiken erstellt und die Indizes der gefundenen Ereignisse in eine Liste geschrieben. Danach wird das Fenster auf den nächsten Zeitraum verschoben und der nächste Schleifendurchlauf beginnt.

Restliche Funktionen

Die Funktion `kombi_ev` verknüpft zwei Listen von Ereignissen mit logischem „Und“ sowie „Oder“. Dafür wird die R-Routine duplicated(·) verwendet, die aus einem Bool-Vektor der doppelten Einträge eine Liste zurückgibt. Mit dieser Funktion können mehrfach vorkommende Ereignisse aus einer Liste gelöscht werden (Verwendung Oder) oder Kriterium 4 überprüft werden (Verwendung Und).
Die Funktion `kontrollplot` wird in der Hauptfunktion `pot_events` verwendet, um die Kontroll-Grafiken in einen Ordner zu speichern.
Die Funktion `make_event_list` generiert aus einer Liste von (sortierten) Ereignissen eine Übersichtstabelle mit Startzeitpunkt und Dauer des Ereignisses. Diese kann unter anderem dazu verwendet werden, um beispielsweise Ereignisse mit einer Dauer kürzer als eine Stunde zu löschen.

4.3. Mathematische Formulierung der Kriterien

In diesem Abschnitt werden die in der Implementierung verwendeten Kriterien zum Erkennen von Ereignissen beschrieben. Alle verwendeten Kriterien liefern potentiell interessante (= vom lokalen Verhalten der Daten abweichende) Datenpunkte, die Teil eines Ereignisses sein können. Ob der gefundene Datenpunkt Teil eines tatsächlichen Ereignisses ist, wird in Abschnitt 4.5 diskutiert. Die ersten drei Kriterien basieren auf der gewonnenen Zerlegung der Zeitreihe. Das vierte Kriterium ist allgemeiner und kann nur in Kombination mit mindestens einem der anderen Kriterien verwendet werden.

Kriterium Nr. 1: Zufallsanteil groß

Ereignisse zeichnen sich durch eine hohe Variabilität der Daten aus. Bei einer Zerlegung spiegelt sich die hohe Varianz im Zufallsanteil wieder. Daher wird bei einem interessanten Datenpunkt der Zufallsanteil größer als dreimal der IQR des Zufallsfehlers sein. Der IQR ist eine statistische Größe und steht für *Interquantiler Bereich*. Er wird aus der Differenz des 3. und 1. Quartils berechnet, also $\mathrm{IQR} = q_{0.75} - q_{0.25}$. Der IQR enthält 50% der Daten und ist ein (verzerrter) Schätzer für die Streuung σ. Die Streuung σ kann durch $s = \frac{\mathrm{IQR}}{1.349}$ geschätzt werden.
Das Kriterium 1 lautet:

$$random > K_1 \cdot \mathrm{IQR}(random),\ K_1 \in \mathbb{R}_+.$$

Die Konstante K_1 wird in der Implementierung standardmäßig auf den Wert 3 festgesetzt.

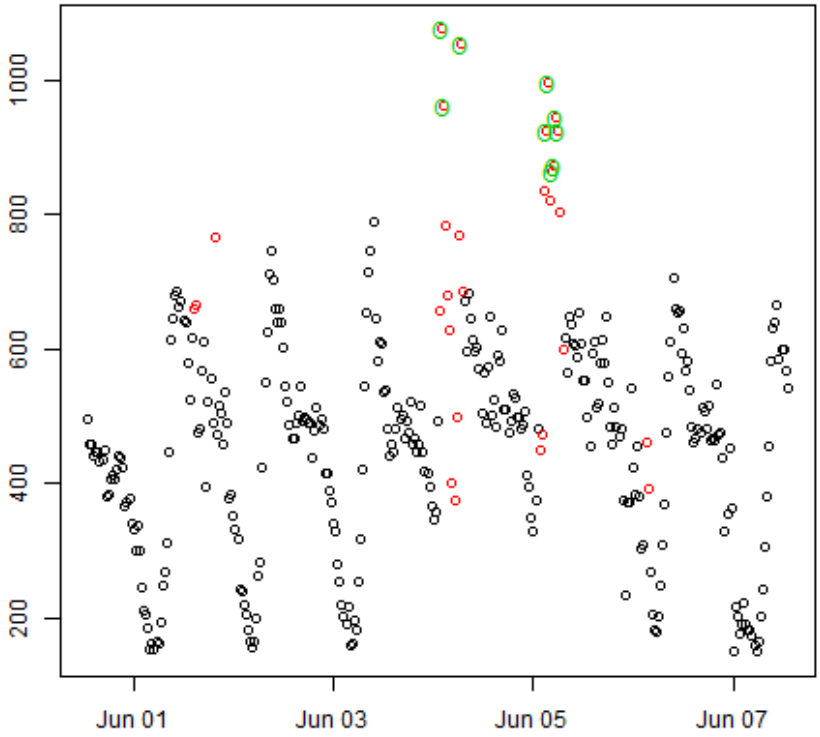

Abbildung 4.2.: Erkannte Ereignisse der Variablen `codeq_inflow` mit Kriterium 1 (rot) und Kriterium 2 (grün) Anfang Juni 2009 bei einer Fensterbreite von 7 Tagen.

Kriterium Nr. 2: Daten liegen außerhalb der Auslenkung des regulären Patterns

Im Kriterium 2 wird überprüft, ob die Auslenkung der Daten ohne den Trend außerhalb der maximalen Auslenkung des saisonalen Patterns liegt.
Das Kriterium 2 ist definiert als:

$$|data - trend| > K_{\min} \cdot |\min(seasonal) - \mathrm{med}(seasonal)|,\ K_{\min} \in \mathbb{R}_+$$

oder

$$|data - trend| > K_{\max} \cdot |\max(seasonal) - \mathrm{med}(seasonal)|,\ K_{\max} \in \mathbb{R}_+.$$

Die Faktoren $K_{\max}$ und $K_{\min}$ auf der rechten Seite werden für die Implementierung symmetrisch gewählt und standardmäßig auf den Wert 2 festgelegt. Es kann aber in manchen Fällen sinnvoll sein, hier unterschiedliche Werte zu wählen.

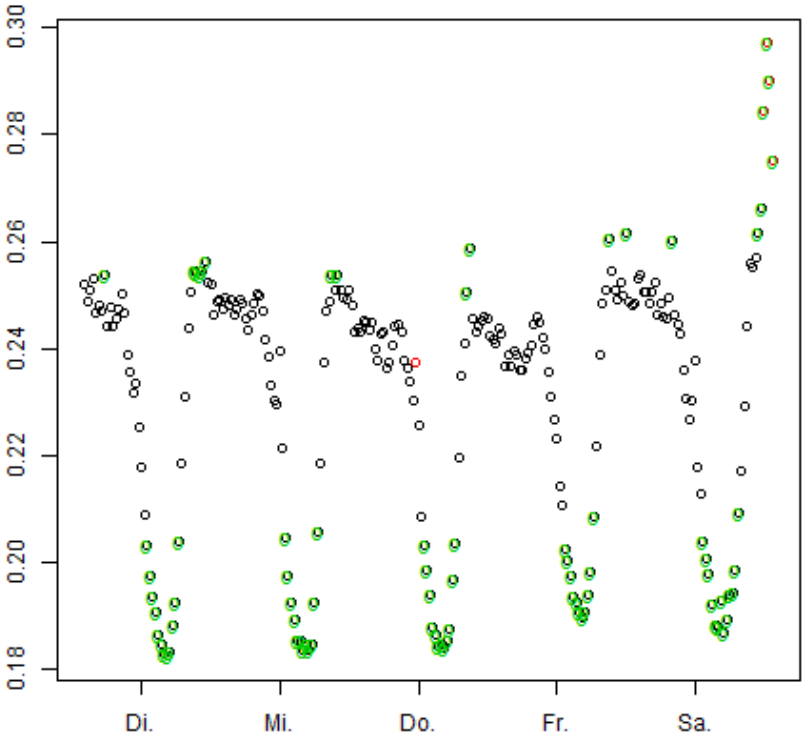

Abbildung 4.3.: Erkannte Ereignisse der Variablen `h_cso` mit Kriterium 1 (rot) und Kriterium 2 (grün) Anfang Februar 2011 bei einer Fensterbreite von 5 Tagen.

Kriterium Nr. 3: Große Änderung des Trends

Im dritten Kriterium wird die Änderung des Trends untersucht. Sprunghafte Änderungen des Trends können ebenfalls auf ein Ereignis hindeuten. Das Kriterium 3 lautet:

$$|\Delta trend| > K_3 \cdot \mathrm{IQR}(\Delta trend),\ K_3 \in \mathbb{R}_+.$$

Die Änderung wird im Algorithmus über eine numerische Näherung der Differenzen bestimmt. In dieser Arbeit wird dafür die R-Routine diff(·) verwendet.[5] Dieses Kriterium ist ohne die robuste Verwendung der `stl`-Routine oft sehr instabil, da die robuste Methode im Vergleich zur regulären Methode einen glatteren Trend verwendet. Hier wird der Wert für K_3 standardmäßig auf 3 festgelegt.

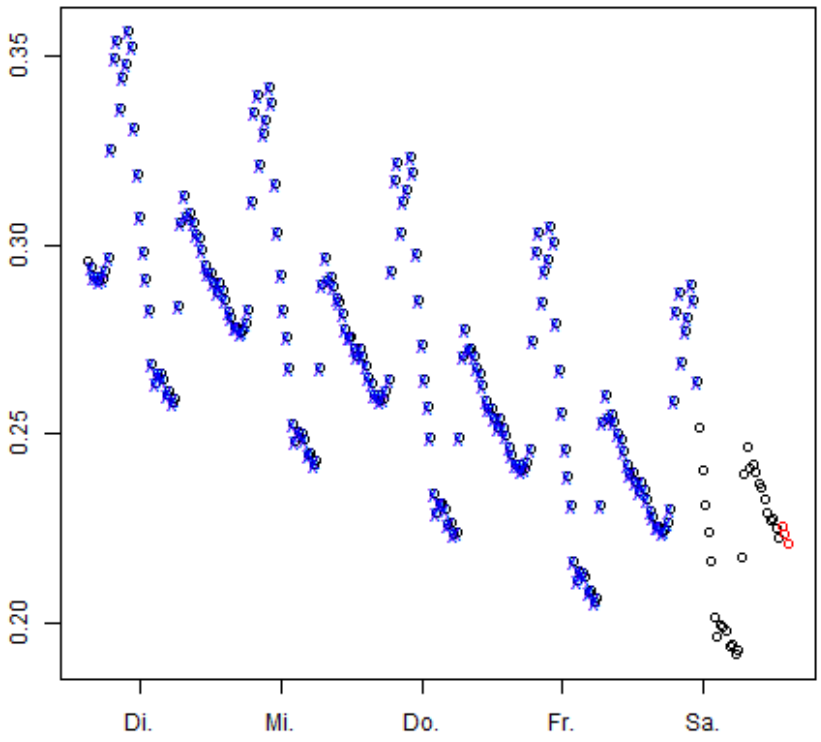

Abbildung 4.4.: Erkannte potentielle Ereignisse der Variablen `h_cso` mit Kriterium 1 (rot) und Kriterium 3 (blau) Ende Juni 2011 bei einer Fensterbreite von 5 Tagen.

[5]Ein technischer Hinweis zur Implementierung: Der Vektor nach der Anwendung der R-Funktion diff(·) ist um einen Eintrag kürzer als der ursprüngliche Vektor. Daher wird dieser Vektor in der Implementierung um einen Eintrag künstlich gestreckt.

Kriterium Nr. 4: Ereignis auch im benachbarten Fenster erkannt

Dieses Kriterium kann nur in Kombination mit mindestens einem der anderen Kriterien verwendet werden. In Kriterium 4 wird ausgenützt, dass die Methode so konstruiert ist, dass sich je zwei benachbarte Fenster der Schleifendurchläufe jeweils um die Hälfte überlappen. Die letzte Hälfte der vorigen Woche entspricht immer dem Zeitraum der ersten Hälfte der aktuellen Woche. In diesem Kriterium werden nur interessante Datenpunkte ausgewählt, die bereits im vorigen Fenster erkannt wurden. Ein Nachteil dieses Kriteriums ist, dass in der derzeitigen Implementierung Fenster, die NAs enthalten, übersprungen werden. Dies führt wiederum dazu, dass Datenpunkte deren überlappendes Gegenstück in einem Fenster mit NAs liegt, mit diesem Kriterium nicht überprüft werden können.

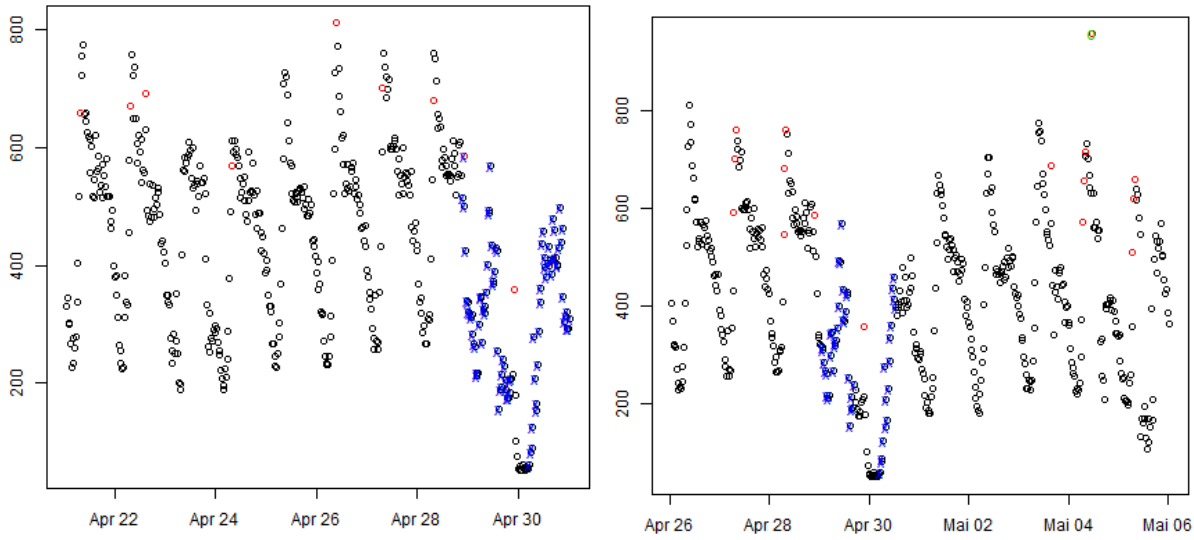

Abbildung 4.5.: Erkannte potentielle Ereignisse der Variablen `codeq_inflow` mit Kriterium 3 (blau) in zwei benachbarten Fenstern Ende April sowie Anfang Mai 2009 bei einer Fensterbreite von 10 Tagen.

4.4. Variationsmöglichkeiten der Parameter

Eine Möglichkeit die Methode zu modifizieren, ist die Fensterbreite der Zerlegung zu variieren. Standardmäßig ist sie auf eine Woche festgelegt. Zu kurze Fensterbreiten sollten nicht verwendet werden, da dann keine sinnvolle Zerlegung für die saisonale Komponente (täglich wiederkehrendes Muster) gefunden werden kann. Wenn die Fensterbreite zu groß gewählt wird, entfallen mehr Fenster durch fehlende NA Werte, da eine plausible Zerlegung bei zu großen Lücken nicht mehr garantiert werden kann. In dieser Arbeit werden Fensterbreiten von 5, 7, 10 sowie 14 Tagen ausgetestet. Außerdem wird eine Fensterbreite von 162 Stunden (bzw. 7 Tage minus 6 Stunden) erprobt, um zu überprüfen, ob unterschiedliche Startzeitpunkte für das Tagespattern bei benachbarten Durchläufen die gefundenen Ereignisse beeinflussen.

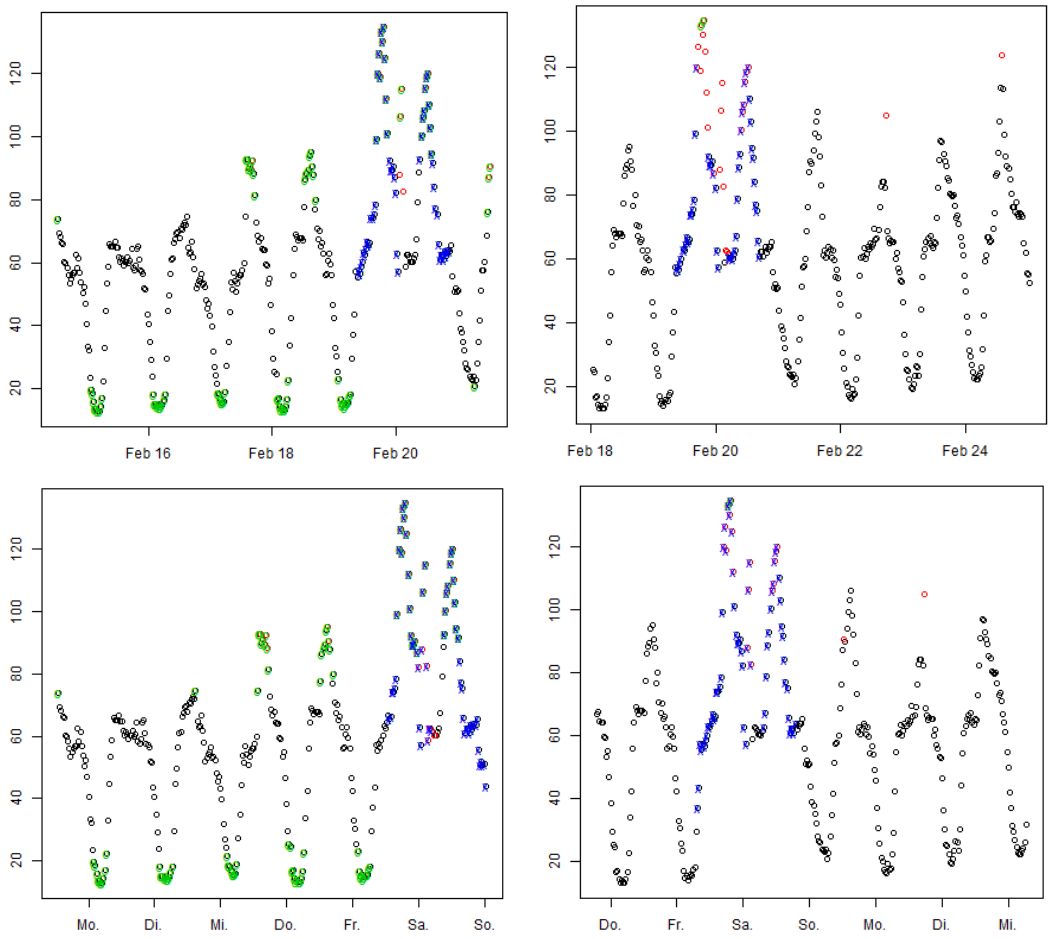

Abbildung 4.6.: Erkannte Ereignisse der Variablen `q_inflow` mit Kriterium 1 (rot), Kriterium 2 (grün) und Kriterium 3 (blau) Mitte Februar 2010 bei einer Fensterbreite von 7 Tagen (oben) sowie 162 Stunden bzw. 6 Tagen und 18 Stunden (unten).

Zusätzlich können bei Kriterien 1 bis 3 die Größe der Konstanten variiert werden. Für die Variablen `h_cso` und `q_inflow` wird nur die Standardeinstellung betrachtet, da diese bereits gute Ergebnisse liefert. Bei der Variable `codeq_inflow` wird zusätzlich eine sensiblere Variante betrachtet.

	Standard	Sensibel
K_1	3	2
$K_{\max}$, $K_{\min}$	2	1.75
K_3	3	2

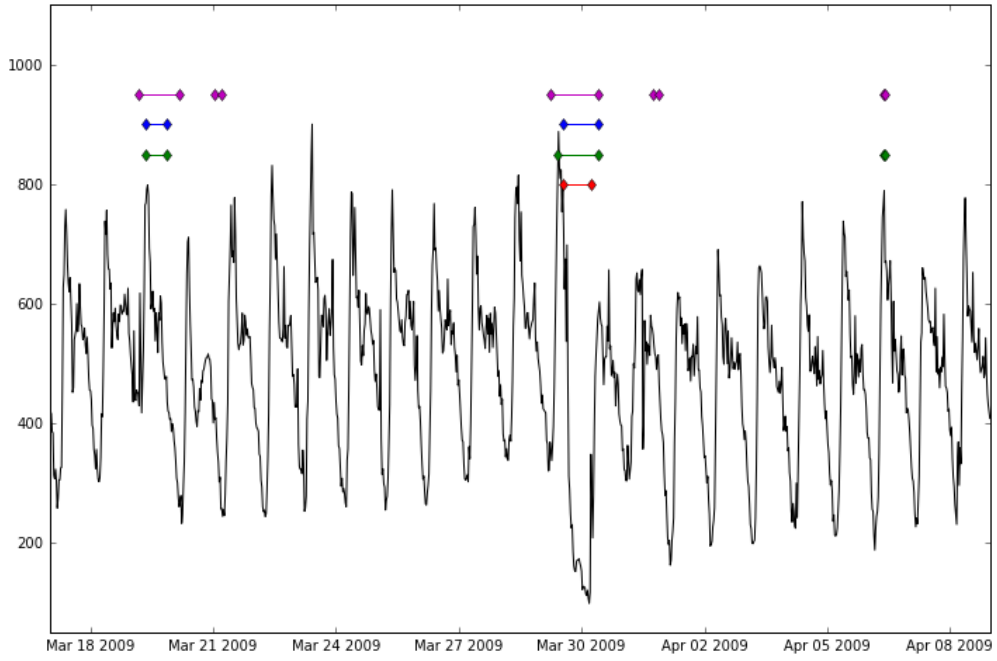

Abbildung 4.7.: Die erkannten Ereignisse der Variablen `codeq_inflow` bei einer Fensterbreite von 7 Tagen (rot), 14 Tagen (grün) mit der Standard-Wahl der Konstanten sowie 7 Tage (blau), 14 Tage (magenta) mit Änderung der Konstanten auf eine sensiblere Variante und deren Verlauf (schwarz) im Zeitraum 18. März bis 9. April 2009 dargestellt.

4.5. Von interessanten Datenpunkten zu Ereignissen

Die in den vorigen Abschnitten genannten Kriterien liefern interessante Datenpunkte. Alle mit den Kriterien gefundenen Datenpunkte weichen mehr oder weniger vom lokalem Muster der Daten an. Ob sie Teil eines Ereignisses sind, kann erst durch die richtige Kombination der Kriterien garantiert werden. Dieser Vorgang wird in diesem Abschnitt beschrieben.

Variable/ Kriterium	1	2	3	1+2	1+3	2+3	1+2+3
codeq_inflow	2410	601	3312	339	435	125	55
h_cso	4189	11361	3211	2962	1045	1680	918
q_inflow	4103	10097	2941	2905	970	1579	858

Tabelle 4.1.: Absolute Anzahl der gefundenen interessanten Datenpunkte bei der Anwendung der Kriterien 1,2,3 allein und in Kombination in den Jahren 2009 bis 2011 bei einer Fensterbreite von 7 Tagen und der Standard-Wahl der Konstanten. Die gesamte Anzahl der betrachteten Datenpunkte liegt bei 52555.

Zunächst folgen einige Anmerkungen zur Verwendung der einzelnen Kriterien: Kriterium 1 sowie 2 allein treten auch bei nicht auffälligen Datenpunkten auf (siehe Abb. 4.2 sowie 4.3). Besonders Kriterium 2 markiert ca. jeden fünften Punkt bei den Variablen `h_cso` und `q_inflow` als interessant. Bei Kriterium 3 werden vereinzelt ganze Fenster als interessant markiert (siehe beispielsweise Abb. 4.4). Dieser Fall liefert in der Regel ein Ereignis, einen sogenannten Shift der Messung, da der Trend monoton über einen längeren Zeitraum ansteigt oder abfällt. Wenn das Kriterium 3 am Rand eines Fensters auftritt, wird es im benachbarten Fenster meistens nicht als interessant erkannt (siehe 4.5 sowie Abb. 4.8). Kriterium 2 wird meistens in beiden benachbarten Fenstern als interessant erkannt.
Die Kombination aus Kriterium 1 und 2 liefert mit hoher Zuverlässigkeit ein Ereignis. Kriterium 3 sollte unbedingt mit Kriterium 4 gemeinsam angewendet werden, weil sonst unterschiedlich gewählte Fensterbreiten teilweise verschiedene Ereignisse mit Kriterium 3 liefern. Je größer die Fensterbreite gewählt wird, desto öfter werden Ereignisse mit Kriterium 3 erkannt. Für Kriterium 1 und 2 sind kleinere Fensterbreiten zu bevorzugen, da bei großen Fensterbreiten die Ergebnisse durch Ausreißer verfälscht werden.
In der Tabelle 4.2 wird die Anzahl der gefundenen Ereignisse, die durch die Zusammenlegung interessanter Datenpunkte der Kriterien entstanden sind, aus den Jahren 2009 bis 2011 beschrieben. Bei den Variablen `h_cso` und `q_inflow` werden mit Kriterium 2 mit Abstand am meisten Ereignisse erkannt. Diese Anzahl sollte mit Vorsicht betrachtet werden, da wie oben beschrieben nicht alle mit Kriterium 2 allein gefundenen Datenpunkte grafisch auffällig sind. Bei der Variablen `codeq_inflow` werden mit Kriterium 1 am meisten Ereignisse erkannt. Bei allen drei Variablen werden mit Kriterium 3 am wenigsten Ereignisse erkannt, da dieses Kriterium tendenziell länger dauernde Ereignisse als die anderen beiden Kriterien findet (vlg. Abschnitt 4.6. Durch die kombinierte Anwendung der gefundenen Kriterien wird die Anzahl deutlich reduziert. Die Gesamtanzahl der Ereignisse (= die durch die hier implementierte Methode gefundene Anzahl) ergibt sich aus den benachbarten Datenpunkten, die entweder aus der kombinierten Anwendung der Kriterien 1 und 2 gemeinsam oder der Kriterien 3 und 4 gemeinsam gefunden wurden.
Bei der Variablen `codeq_inflow` funktioniert das Erkennen der Ereignisse durch die oben genannten Kriterien mit der Standard-Wahl der Konstanten am schlechtesten. Es werden deutlich weniger interessante Datenpunkte als bei den anderen beiden Variablen mit allen Kriterien erkannt. Der Grund liegt darin, dass sich die Daten nicht weit außerhalb der

Kriterien	allein			kombiniert		gesamt
Variable	1	2	3	1+2	3+4	
`codeq_inflow`	1037	290	92	194	18	209
`h_cso`	789	1613	82	477	36	474
`q_inflow`	737	1555	89	511	31	503

Tabelle 4.2.: Anzahl der gefundenen Ereignisse für die Jahre 2009 bis 2011 nach Zusammenlegung der benachbarten interessanten Datenpunkte. Die Gesamtanzahl ergibt sich aus den benachbarten Datenpunkten, die von Kriterien 1 und 2 gemeinsam oder 3 und 4 gemeinsam identifiziert werden.

Schwankungsbreite des saisonalen Anteils bewegen und sich auch der Trend nicht signifikant ändert. Durch eine Veränderung der Parameter kann ein besseres Ergebnis erzielt werden (vgl. Abb. 4.7). Aus dieser Abbildung geht hervor, dass die Wahl von einer Fensterbreite von 14 Tagen und einer sensibleren Wahl der Konstanten zu einem besseren Ergebnis beim Erkennen führt. Allerdings kann im Allgemeinen eine zu kleine Wahl der Konstanten zu einer zu sensiblen Methode führen, die vermehrt Ereignisse falsch identifiziert.
Allgemein hat die Wahl von unterschiedlichen Startzeitpunkten für das Tagespattern nur einen geringen Einfluss auf die erkannten Ereignisse. Siehe zum Beispiel Abb. 4.6, in der die Fensterbreite 7 Tage bzw. 168 Stunden mit den Ergebnissen der Fensterbreite 162 Stunden verglichen wurde.

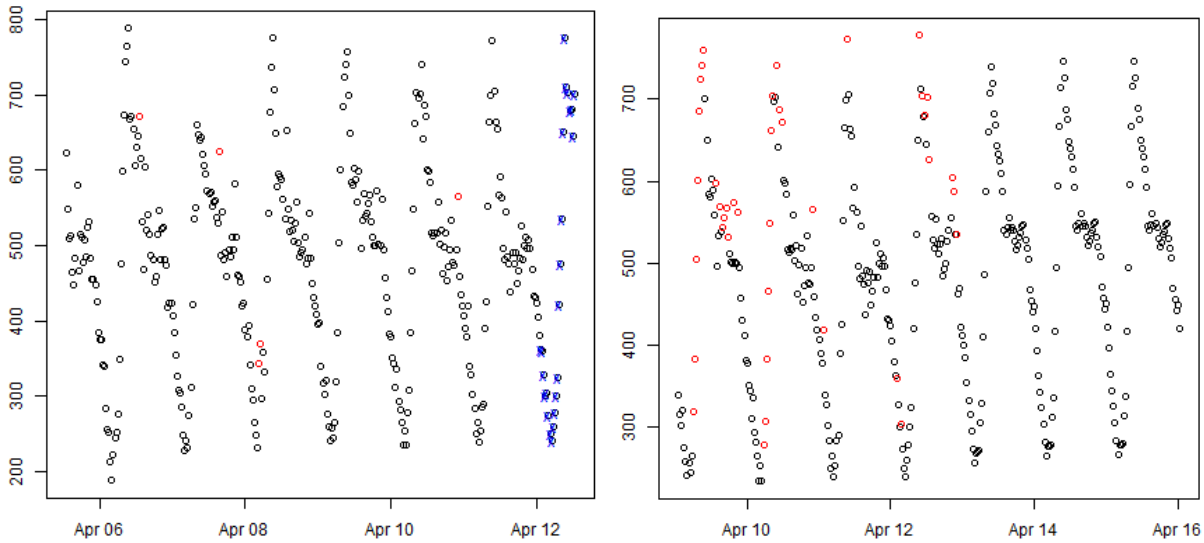

Abbildung 4.8.: Erkannte potentielle Ereignisse der Variablen `codeq_inflow` mit Kriterium 1 (rot), Kriterium 2 (grün) und Kriterium 3 (blau) in zwei benachbarten Fenstern Anfang April 2009 bei einer Fensterbreite von 7 Tagen.

4.5.1. Zusammenfassung der gewonnenen Erkenntnisse aus der Praxis

Die Kriterien liefern interessante Datenpunkte, die vom lokalem Muster der Daten mehr oder weniger abweichen. Benachbarte interessante Datenpunkte ergeben ein Ereignis. Die folgende Kombination der Kriterien liefern gute Ergebnisse: Die Kombination der interessanten Datenpunkte der Kriterien 1 und 2 liefert größtenteils unabhängig von der Wahl der Fensterbreite ein Ereignis. Die Wahl der Fensterbreite hat auf die auf Kriterium 3 basierten Ereignisse einen großen Einfluss. Um diesen zu verringern, sollte Kriterium 3 mit Kriterium 4 kombiniert werden.
Die Anwendung hat gezeigt, dass für die Kriterien 1 und 2 eher eine kürzere Fensterbreite gewählt werden sollte (kürzer als 7 Tage). Für Kriterium 3 ist eher eine längere Fensterbreite zu bevorzugen (länger als 7 Tage). Bei der Variablen `codeq` hat sich aufgrund der geringeren relativen Bandbreite der Daten eine engere Wahl der Konstanten bei den einzelnen Kriterien bewährt. Die Wahl von unterschiedlichen Startzeitpunkten für das Tagespattern (Fensterbreite von 162 Stunden) hat nur einen geringen Einfluss auf die erkannten Ereignisse.

4.6. Analyse der Dauer der Ereignisse

Zum Abschluss dieses Kapitels werden die gefundenen Ereignisse in den Jahren 2009 bis 2011 bezüglich ihrer Dauer analysiert. Die Analyse beruht auf Histogrammen, in denen auf der x-Achse die Dauer und auf der y-Achse die Häufigkeiten dargestellt werden. Alle Histogramme in diesem Abschnitt basieren auf einer Fensterbreite von 7 Tagen und der Standard-Wahl der Konstanten. In der Darstellung entspricht eine Längeneinheit auf der x-Achse einer Dauer von 30 Minuten.

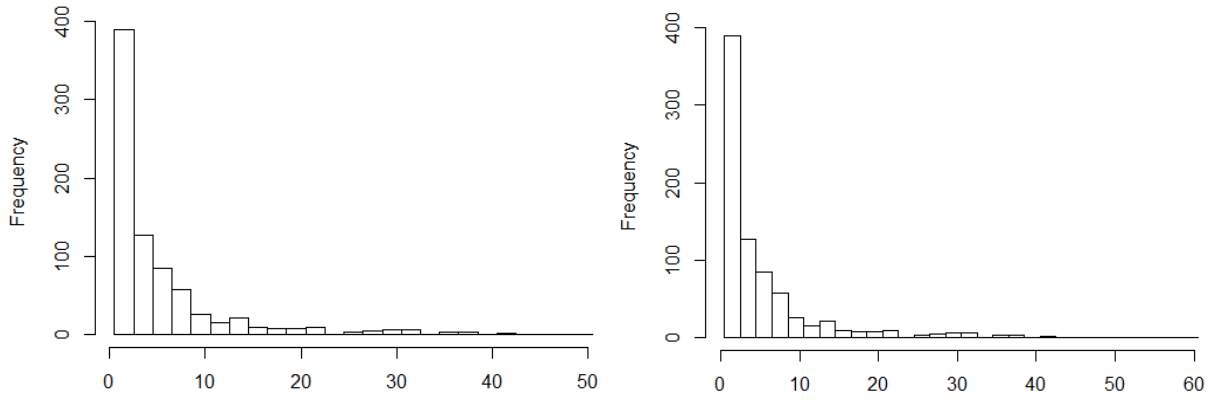

Abbildung 4.9.: Häufigkeiten der Dauer der Ereignisse der Variablen `h_cso` gefunden unter Verwendung der Kriterien 1 (links) und 2 (rechts) allein.

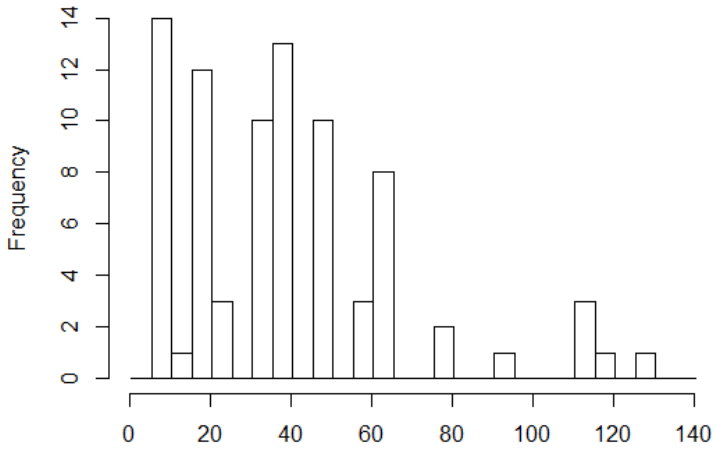

Abbildung 4.10.: Häufigkeiten der Dauer der Ereignisse der Variablen `h_cso` gefunden unter Verwendung des Kriteriums 3 allein.

Zuerst wird die Dauer der Ereignisse der Variablen `h_cso` im Detail betrachtet (siehe Abbildungen 4.9, 4.10 und 4.11). Bei den Kriterien 1 und 2 werden mit Abstand am häufigsten

Ereignisse, die kürzer als 30 Minuten dauern (entspricht Länge 1), gefunden. Danach nehmen die Häufigkeiten mit wachsender Dauer sukzessive ab. Es treten bei diesen beiden Kriterien keine Ereignisse mit einer Dauer länger als 21 Stunden auf (entspricht Länge 42). Bei Kriterium 3 werden mindestens Ereignisse mit einer Dauer von 2 Stunden (entspricht Länge 4) erkannt. Hier schwanken die Häufigkeiten relativ stark. Es treten keine Ereignisse mit einer Dauer länger als 70 Stunden auf (entspricht Länge 140).
Bei der kombinierten Anwendung der Kriterien 1 und 2 reduziert sich die Anzahl der Ereignisse mit einer Dauer von unter 30 Minuten im Vergleich zur alleinigen Anwendung von 400 auf 120. Auch hier nehmen die Häufigkeiten sukzessive mit wachsender Dauer ab. Bei der kombinierten Anwendung der Kriterien 3 und 4 sinkt die maximale Gesamtdauer im Vergleich zur alleinigen Anwendung von Kriterium 3 von 130 auf 80. Hier schwanken die Häufigkeiten deutlich weniger als in Abbildung 4.10.

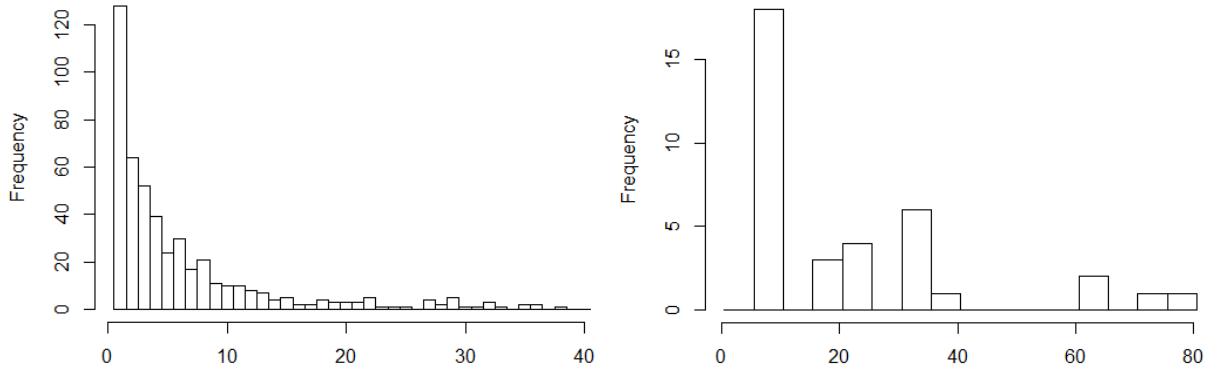

Abbildung 4.11.: Dauer der gefundenen Ereignisse bei Kombination der Kritieren 1 + 2 (links) sowie Kriterien 3 + 4 (rechts).

Zum Abschluss wird für die Variablen `codeq_inflow`, `h_cso` und `q_inflow` das Gesamtergebnis (Kombination 1+2 oder 3+4) der Ereignisse bezüglich ihrer Dauer dargestellt (siehe Abb. 4.12). Bei allen drei Variablen liegt die Dauer der meisten Ereignisse unter 20 Stunden (entspricht Länge 40). Hier treten Ereignisse mit Länge 1 (entspricht Dauer 30 Minuten) am häufigsten auf und nehmen dann sukzessive mit wachsender Dauer ab. Vereinzelt treten Ereignisse mit Dauern über 40 Stunden (Länge 80) auf.

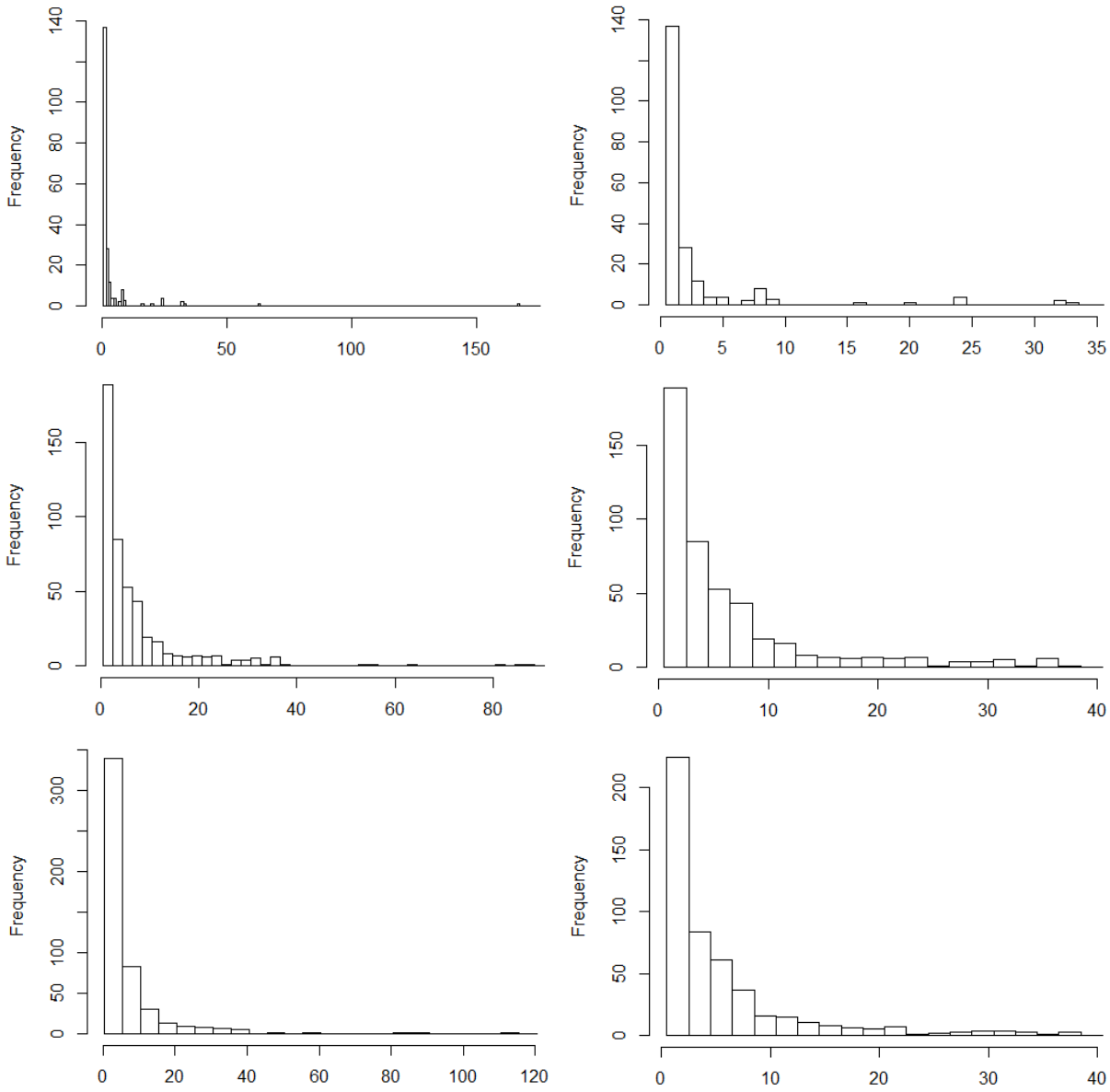

Abbildung 4.12.: Dauer der Ereignisse der Variablen `codeq_inflow` (oben), `h_cso` (Mitte) und `q_inflow` (unten). Gesamtes Histogramm (links) und detailliertere Auflösung mit niedrigeren Häufigkeiten (rechts).

5. Vergleich mit bestehenden Methoden

In diesem Abschnitt sollen die gewonnenen Ergebnisse des in dieser Arbeit implementierten Verfahren zur Erkennung von Ereignissen mit den Ergebnissen anderen bestehenden Methoden verglichen werden. Es werden hier die Ereignislisten der Variablen `codeq_inflow`, `h_cso` und `q_inflow` verglichen. Als Zeitraum für den Vergleich wurde Jänner bis April 2009 ausgewählt, da hier die Daten nahezu vollständig gemessen wurden. In dieser Periode sind nur sehr wenige Lücken vorhanden und das Vervollständigen der Daten durch die LOESS-Methode hat nur wenig Einfluss auf die Ereignislisten. Als Referenz werden ein einfaches Maximum-Kriterium bei der Variable `h_cso` sowie die Ereignislisten aus [Hofer 2012] verwendet.

Maximum-Kriterium

Das erste Kriterium für den Vergleich ist ein simpler Maximumsvergleich, der zuverlässig ein Ereignis liefert und somit in der Praxis oft für schnelle Tests verwendet wird. Der Schwellwert für die Variable `h_cso` ist aus der Praxis bekannt: Wenn der Wasserstand in der Mischwasserkammer über 0.5 Meter ansteigt, wird das Fassungsvermögen der Mischwasserkammer überschritten und der Überlauf wird aktiviert. Das Aktivieren des Überlaufs kann als Ereignis gewertet werden. Bei der anderen Variablen `codeq_inflow` und `q_inflow` ist kein simpler Schwellwertvergleich möglich.

Kriterium Hofer

Bei [Hofer 2012] wird ein umfangreicheres Verfahren zum Erkennen von Ereignissen angewendet: Es werden mehrere Variablen mit einer hohen Korrelation gekoppelt untersucht. Dabei werden für jede Variable Schwellwerte festgelegt, nämlich ein Minimum und ein Maximum, die durch Erfahrungswerte von gültig validierten Daten der Vorjahre gewonnen werden. Wenn eine oder mehrere Variablen den zulässigen Bereich verlassen, wird ein Ereignis identifiziert. Zusätzlich wird das Über- bzw. Unterschreiten der Schwellwerte an das vorherige Auftreten von Niederschlag gekoppelt. Bei der Variablen `h_cso` ergibt sich die untere Schranke aus den natürlichen physikalischen Gegebenheiten: Der Wasserstand soll nicht negativ sein und wird daher auf 0.0 Meter festgesetzt. Die obere Schranke für das Jahr 2009 ergibt sich aus Erfahrungswerten und wird auf 2.5 Meter festgelegt. Auf eine ähnliche Weise werden für das Jahr 2009 bei der Variablen `q_inflow` die Schwellwerte auf 0 und 2500 und bei der Variablen `codeq_inflow` werden die Schwellwerte auf 0 und 1500

festgesetzt. Aufgrund der oben beschriebenen Vorgangsweise ergibt sich für alle drei Variablen `codeq_inflow`, `h_cso` und `q_inflow` die selbe Liste für den Vergleich. Sie basiert auf einer Analyse der Originaldaten und stammt aus [Hofer 2012] (Anhang C).

Eigene Kriterien

Die Liste der Ereignisse entstand aus den benachbarten interessanten Datenpunkten, die entweder durch die Kombination der Kriterien 1 und 2 oder 3 und 4 identifiziert wurden (vgl. Abschnitt 4.3). Zusätzlich wurde mit der Standard-Wahl der Konstanten und einer Fensterbreite von sieben Tagen gearbeitet. Für weitere Details und Begründung der Auswahl siehe Abschnitt 4.5.

Ergebnisse des Vergleichs der Variablen `h_cso`

In der Tabelle 5.1 werden die gefundenen Ereignisse der Variablen `h_cso` aus dem Zeitraum Jänner bis April 2009 der oben beschriebenen Kriterien mit den eigenen Kriterien dieser Arbeit verglichen. In Abbildung 5.1 wird der Vergleich grafisch dargestellt.

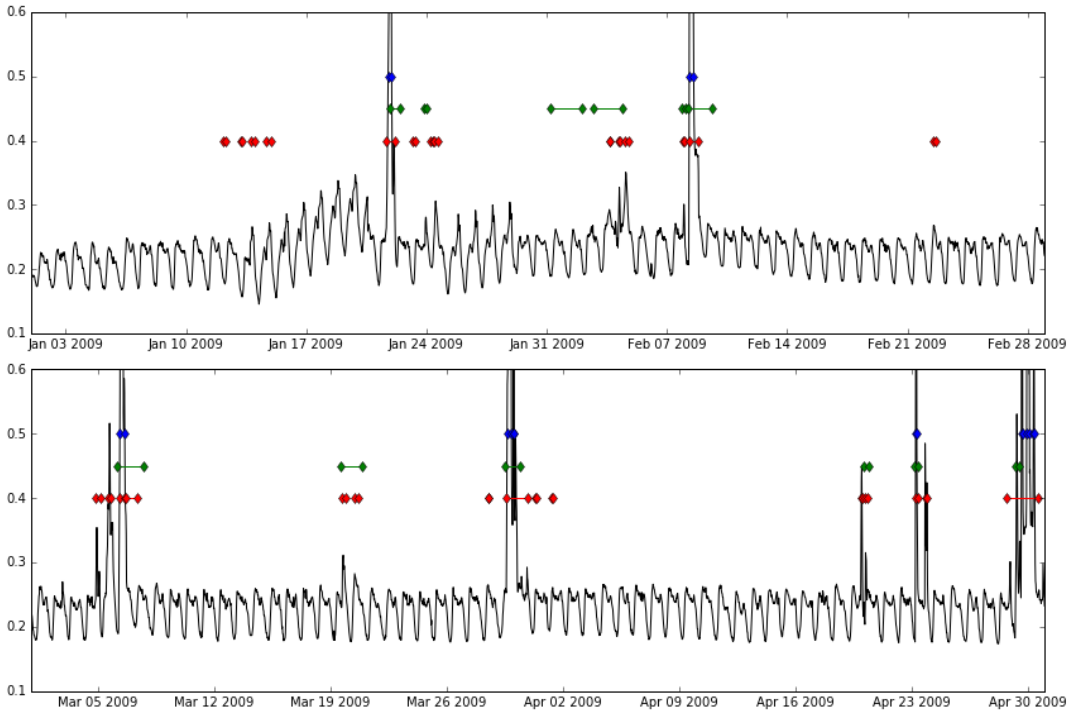

Abbildung 5.1.: Grafischer Vergleich der erkannten Ereignisse mit Maximum-Kriterium (blau), Kriterium Hofer (grün) und den eigenen Kriterien (rot). Zusätzlich wird der Verlauf der Variablen `h_cso` (schwarz) im Zeitraum Jänner bis April 2009 dargestellt.

Von allen drei Methoden wurden 6 Ereignisse gefunden. 4 Ereignisse stimmen bei Hofer und den hier verwendeten Kriterien überein. Zwei Ereignisse werden exklusiv von Hofer und 12 von den eigenen Kriterien erkannt. Alle bis auf die beiden Ereignisse, die nur bei Hofer gefunden werden, werden von den hier verwendeten Kriterien ebenfalls als Ereignisse identifiziert. Alle mit dem Maximum-Kriterium gefundenen Ereignisse werden von den anderen beiden Methoden ebenfalls erkannt. Die in dieser Arbeit verwendete Methode findet zusätzlich 12 Ereignisse.
Die Abweichungen zwischen Ereignislisten der verschiedenen Kriterien lassen sich folgendermaßen zusammenfassen: Die Ereignisgruppe um den 13. Jänner (E1 bis E4) lässt sich auf einen Wechsel des Trends sowie des Tagesmusters zurückführen und ist laut der Definition in Abschnitt 1.3 ein Ereignis. Die Ereignisse um den 24. Jänner (H1, E5 und E6) lassen sich auf eine Änderung des täglichen Musters zurückführen. Aus grafischer Sicht ist unklar, warum das Ereignis H2 bei Hofer identifiziert wurde. Das Ereignis um den 22. Februar (E7) geht aus einer Änderung des Tagesmusters hervor. Um das Ereignis #4 Ende März werden von den hier verwendeten Kriterien zwei Ereignisse erkannt (E10 und E11). Dies liegt vermutlich an der überdurchschnittlich langen Dauer der Anomalie von eineinhalb Tagen im Verhältnis zur Fensterbreite von sieben Tagen. Bei einer größeren Fensterbreite, zum Beispiel 14 Tage, werden sie nicht mehr als Ereignis erkannt (siehe Abb. 5.2).

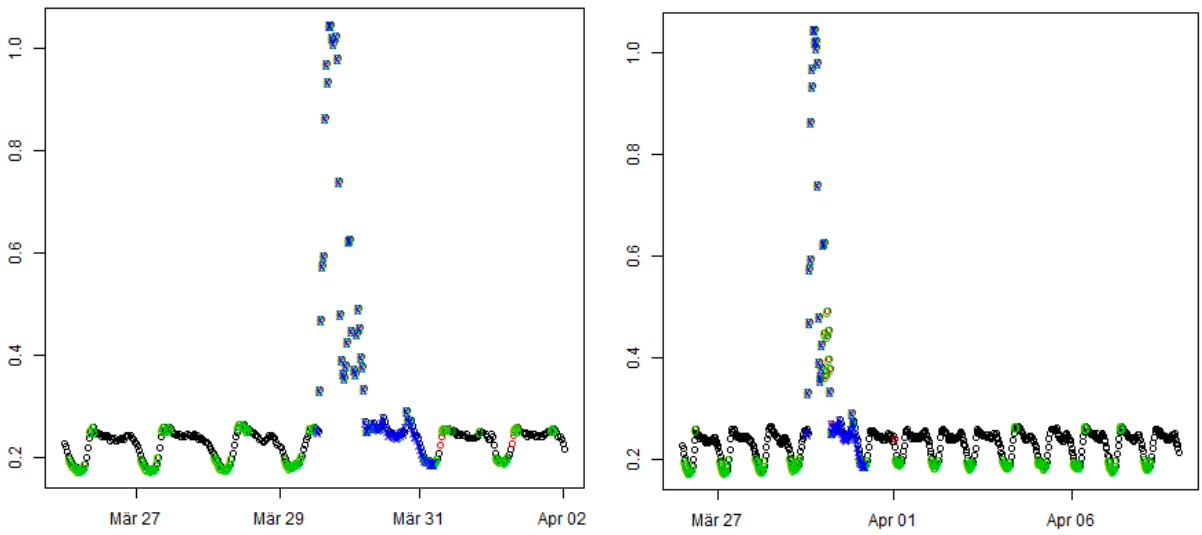

Abbildung 5.2.: Gefundene Ereignisse mit den eigenen Kriterien bei einer Fensterbreite von 7 Tagen (links) und 14 Tagen (rechts) der Variablen `h_cso` Ende März bzw. Anfang April 2009.

Ergebnisse des Vergleichs der Variablen `q_inflow`

In der Tabelle 5.2 werden die gefundenen Ereignisse der Variablen `q_inflow` aus dem Zeitraum Jänner bis April 2009 der oben beschriebenen Kriterien mit den eigenen Kriterien dieser Arbeit verglichen. In Abbildung 5.3 wird der Vergleich grafisch dargestellt. Acht Ereignisse stimmen bei Hofer und den hier verwendeten Kriterien überein. 8 Ereignisse werden exklusiv von Hofer und 17 von den eigenen Kriterien erkannt.

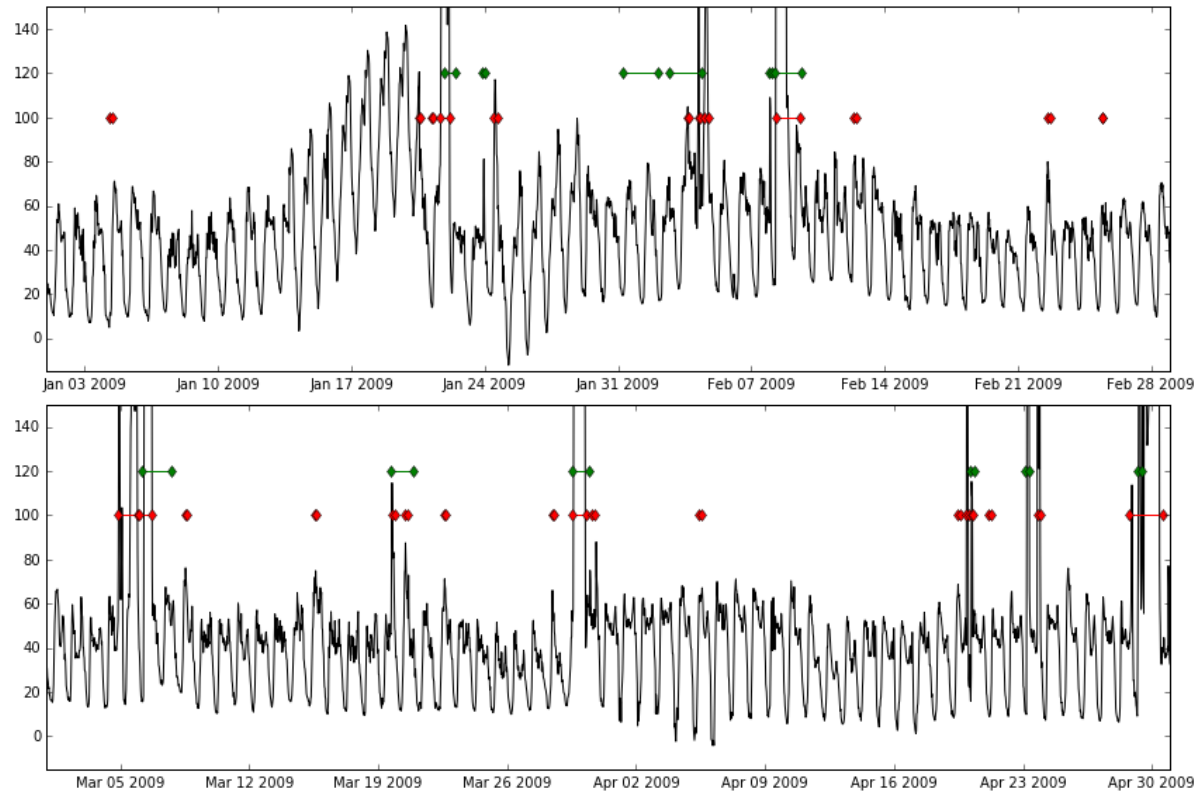

Abbildung 5.3.: Grafischer Vergleich der erkannten Ereignisse mit Kriterium Hofer (grün) und den eigenen Kriterien (rot). Zusätzlich wird der Verlauf der Variablen `q_inflow` (schwarz) im Zeitraum Jänner bis April 2009 dargestellt.

Ergebnisse des Vergleichs der Variablen `codeq_inflow`

In der Tabelle 5.3 werden die gefundenen Ereignisse der Variablen `codeq_inflow` aus dem Zeitraum Jänner bis April 2009 der oben beschriebenen Kriterien mit den eigenen Kriterien dieser Arbeit verglichen. In Abbildung 5.4 wird der Vergleich grafisch dargestellt. 6 Ereignisse stimmen bei Hofer und den hier verwendeten Kriterien überein. 6 Ereignisse werden exklusiv von Hofer und 8 von den eigenen Kriterien erkannt.

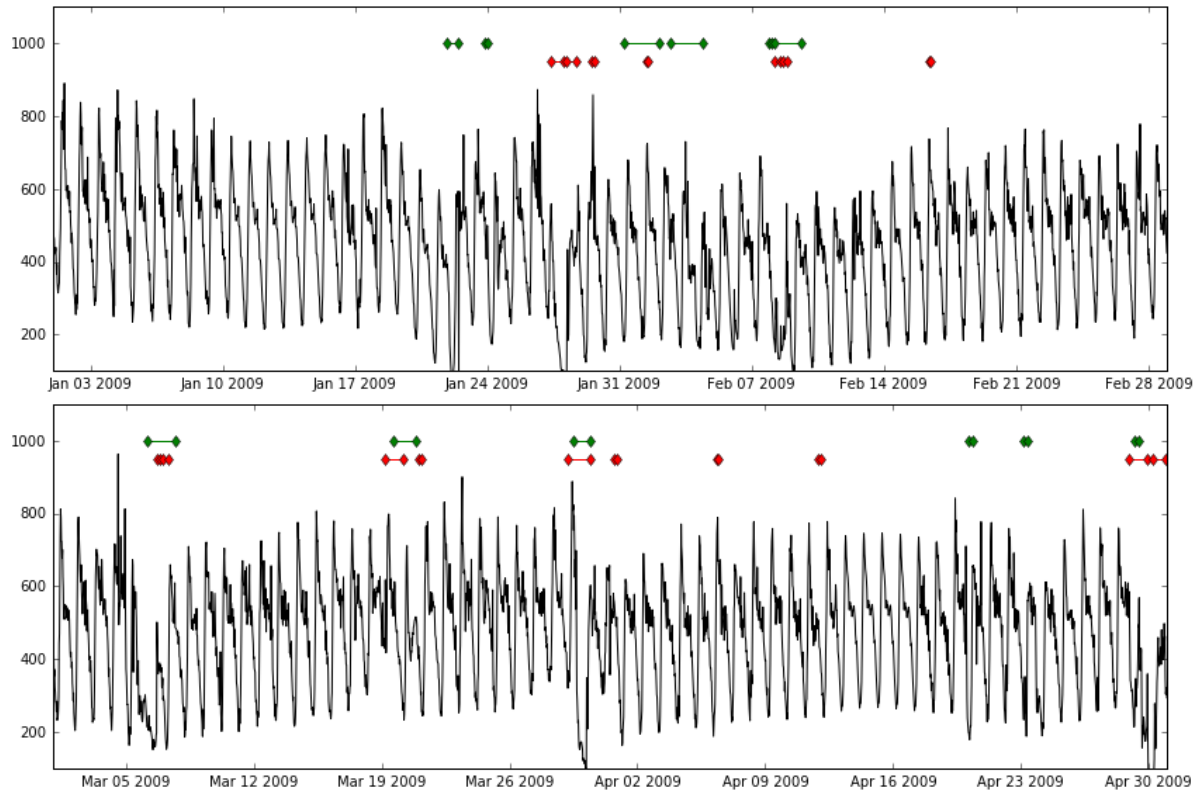

Abbildung 5.4.: Grafischer Vergleich der erkannten Ereignisse der Variablen `codeq_inflow` mit Kriterium Hofer (grün) und den eigenen Kriterien (rot) und deren Verlauf (schwarz) im Zeitraum 1. Jänner bis 9. April 2009.

Zusammenfassung

Die Kriterien stimmen zum größten Teil überein. Bei der Variablen `c_hso` liefert das Maximum-Kriterium am wenigsten Ereignisse, ist somit nur für grobe und schnelle Aussagen geeignet. Die Liste der Ereignisse durch die Methode von Hofer und des hier verwendeten Ansatzes stimmen sehr gut überein bei den Variablen `h_cso` und `q_inflow`. Generell finden die eigenen Kriterien noch zusätzliche Ereignisse, die bei Hofer nicht erkannt werden. Bei der Variable `codeq_inflow` stimmen die gefundenen Ereignisse nur teilweise überein. Dies liegt vermutlich daran, dass beide Methoden zu wenig sensibel auf die schwachen lokalen Änderungen in den Daten reagieren.

ID	Maximum-Krit. Startzeitpunkt	Dauer	Krit. Hofer Startzeitpunkt	Dauer	Eigene Kriterien Startzeitpunkt	Dauer
E1					12.01. 04:00	8
E2					13.01. 04:30	5
E3					13.01. 18:00	11
E4					14.01. 16:30	12
#1	21.01. 19:00	9	21.01. 20:30	28	21.01. 16:30	24
					23.01. 05:30	4
H1			23.01. 20:30	7		
E5					24.01. 06:00	5
E6					24.01. 11:30	10
H2			31.01. 05:30	89		
+1			02.02. 16:30	82	03.02. 15:30	3
					04.02. 05:00	3
					04.02. 12:30	11
+2			07.02. 21:30	8	07.02. 23:00	4
#2	08.02. 07:30	12	08.02. 05:00	67	08.02. 07:00	27
E7					22.02. 12:30	7
E8					04.03. 21:30	13
E9					05.03. 14:30	8
#3	06.03. 08:00	14	06.03. 04:30	75	06.03. 07:30	16
					06.03. 16:30	32
+3			19.03. 15:30	59	19.03. 17:30	9
					20.03. 10:30	12
E10					28.03. 11:30	3
#4	29.03. 15:30	12	29.03. 12:30	44	29.03. 13:30	64
	30.03. 00:30	2				
					31.03. 08:00	3
E11					01.04. 08:00	3
E12					19.04. 22:30	5
+4			20.04. 03:00	12	20.04. 05:00	4
#5	23.04. 05:30	3	23.04. 03:30	11	23.04. 05:30	6
E12					23.04. 19:00	4
#6	29.04. 14:30	3	29.04. 05:30	10	28.04. 17:30	88
	29.04. 22:00	8				
	30.04. 07:30	3				

Tabelle 5.1.: Liste der Ereignisse der Variablen `h_cso` mit den verschiedenen Kriterien von Jänner bis April 2009. Die Dauer eines Ereignisses ist in halben Stunden angegeben. Die ID besteht aus einer fortlaufenden Nummer und einem Symbol. Bei drei sich überschneidenden Ereignissen wird #, bei zwei + und wenn nur ein Kriterium das Ereignis liefert, wird ein Buchstabe als Symbol gewählt (Hofer H, Eigene Kriterien E, Maximum-Kriterium M). Nicht gekennzeichnete Ereignisse überschneiden sich mit einem anderen bereits gekennzeichneten Ereignis.

ID	Eigene Kriterien Startzeitpunkt	Dauer	Krit. Hofer Startzeitpunkt	Dauer
E1	04.01. 08:00	7		
E2	20.01. 12:30	3		
E3	21.01. 05:00	5		
+1	21.01. 16:30	23	21.01. 20:30	28
H1			23.01. 20:30	7
E4	24.01. 11:30	10		
H2			31.01. 05:30	89
+2	03.02. 15:30	3	02.02. 16:30	82
	04.02. 05:00	4		
	04.02. 12:30	12		
H3			07.02. 21:30	8
+3	08.02. 07:00	59	08.02. 05:00	67
E5	12.02. 09:00	5		
E6	22.02. 12:30	8		
E7	25.02. 09:30	3		
E8	04.03. 21:30	49		
+3	06.03. 00:30	34	06.03. 04:30	75
E9	08.03. 12:00	6		
E10	15.03. 13:00	6		
+4	19.03. 17:30	8	19.03. 15:30	59
	20.03. 10:30	8		
E11	22.03. 14:00	4		
E12	28.03. 10:30	3		
+5	29.03. 13:30	34	29.03. 12:30	44
E13	30.03. 13:30	8		
E14	05.04. 10:00	6		
E15	19.04. 11:00	5		
E16	19.04. 22:30	5		
+6	20.04. 05:00	3	20.04. 03:00	12
E17	21.04. 03:00	7		
+7	23.04. 19:00	8	23.04. 03:30	11
+8	28.04. 17:30	88	29.04. 05:30	10

Tabelle 5.2.: Vergleich der erkannten Ereignisse der Variablen `q_inflow` von Jänner bis April 2009. Für Beschreibung siehe Tabelle 5.1.

ID	Krit. Hofer		Eigene Kriterien	
Startzeitpunkt	Dauer		Startzeitpunkt	Dauer
H1	21.01. 20:30	28		
H2	23.01. 20:30	7		
E1			27.01. 08:30	32
E2			28.01. 04:30	24
E3			29.01. 13:30	4
+1	31.01. 05:30	89	01.02. 11:00	2
H3	02.02. 16:30	82		
H4	07.02. 21:30	8		
+2	08.02. 05:00	67	08.02. 04:30	16
			08.02. 16:30	8
E4			16.02. 09:30	2
+3	06.03. 04:30	75	06.03. 16:30	8
			07.03. 00:30	16
+4	19.03. 15:30	59	19.03. 04:30	48
E5			21.03. 00:30	8
+5	29.03. 12:30	44	29.03. 05:30	56
E6			31.03. 17:30	8
E7			06.04. 09:00	2
E8			11.04. 21:30	8
H5	20.04. 03:00	12		
H6	23.04. 03:30	11		
			28.04. 21:30	48
+6	29.04. 05:30	10	30.04. 05:30	32

Tabelle 5.3.: Vergleich der erkannten Ereignisse der Variablen `codeq_inflow` von Jänner bis April 2009. Für Beschreibung siehe Tabelle 5.1.

6. Fazit

Das Ziel dieser Arbeit war, einen Algorithmus zum Erkennen von untypischen Daten bzw. sogenannten Ereignissen zu entwickeln, der auf statistischen Verfahren basiert. Die Methode soll möglichst robust sein und in annähernd Echtzeit gute Ergebnisse liefern. Das Verfahren soll an realen Messdaten getestet werden.

Zu Beginn wurde eine explorative Analyse der Messreihe durchgeführt. Dabei wurden unter anderem verschiedene statistische Kenngrößen sowie die Abhängigkeiten der Variablen untersucht. Vor allem durch grafische Analysen der Daten konnten zwei Kernprobleme identifiziert werden: die Lücken in den Messdaten und die große Datenmenge. Für beide Probleme wurde eine einfach umsetzbare und ressourcenfreundliche Lösung gefunden.

Um die Datenmenge zu reduzieren wurden die Daten in halbe Stunden gruppiert. Dabei wurden alle Messpunkte aus einem 30-Minuten-Intervall durch ihren Mittelwert repräsentiert. Hiermit konnte die Anzahl der Beobachtungen im Vergleich zu den Originaldaten um mehr als Faktor 10 verkleinert werden. Neben der Reduktion des Speicherbedarfs wurde die Rechenzeit für die in dieser Arbeit verwendeten Methoden reduziert. Ein weiterer positiver Nebeneffekt der Gruppierung der Daten war der deutlich verringerte Einfluss der zufälligen Schwankungen der Messwerte auf weiterführende Berechnungen und Schätzungen.

Um die Lücken in der Messreihe zu schließen wurde eine nicht parametrische, lokale Regressionsmethode verwendet, die sogenannte LOESS-Methode. Bei diesem Verfahren werden Lücken in den Daten nur mittels ihrer vorhandenen Nachbarn interpoliert. Hier wurde die LOESS-Methode, die auf [Cleveland 1981] zurückgeht, erweitert, sodass die in dieser Arbeit implementierte Routine auch mit wiederkehrenden Mustern in den Messreihen umgehen kann. Mit der erweiterten LOESS-Methode war es möglich, Lücken von einer Länge von bis zu zwei Wochen zu schätzen und somit mit authentischen Daten zu füllen.

Nach der Aufbereitung der Daten wurde ein Verfahren zur Erkennung von Ereignissen implementiert. Die in dieser Arbeit verwendete Methode basiert auf der Zerlegung der Zeitreihe in ihre einzelnen Komponenten (den Trend, die saisonalen Komponente und den Zufallsanteil) sowie auf der lokalen Analyse der Komponenten. Die Zerlegung der Zeitreihe wurde mit der R-Routine `stl` durchgeführt, in der die LOESS-Methode iterativ verwendet wird.

Die Vorteile des gewählten Ansatzes sind:

- Die Erkennung der Ereignisse beruht auf dem lokalen Verhalten der Daten und benötigt keine zusätzlichen Informationen, wie zum Beispiel meteorologische Daten.
- Die Zerlegung der Zeitreihe wird auch bei kurzfristigen, starken Verzerrungen der Daten robust geschätzt, da durch die iterative Gewichtung der Daten Ausreißern eine geringere Rolle bei der Bestimmung der Zerlegung zugeordnet wird.
- Das simple Design erlaubt eine einfache Anwendung und liefert Ergebnisse fast in Echtzeit. Zusätzlich kann die Methode modular um weitere Kriterien erweitert werden, ohne die Hauptfunktionalität zu ändern.
- Die Methode liefert einfach überprüfbare und nachvollziehbare Ergebnisse.

Eine Einschränkung der gewählten Methode ist, dass sie hauptsächlich für Variablen mit wiederkehrendem Muster geeignet ist.

Anschließend wurden die Ereignislisten dieser Methode für die Variable `h_cso` mit herkömmlichen Methoden verglichen. Das erste Vergleichskriterium war ein einfacher Maximums-Vergleich, das zweite ein Minimums- und Maximum-Vergleich, der zusätzlich an das Auftreten von Niederschlag gekoppelt ist. Die durch die anderen beiden Kriterien erkannten Ereignisse decken sich nahezu komplett mit den durch diese Methode erkannten Ereignissen. Abgesehen davon fand die hier verwendete Methode zusätzliche Ereignisse, die von den anderen beiden Methoden nicht identifiziert werden konnten. Aus diesen Gründen liefert der in dieser Arbeit gewählte Ansatz eine Verbesserung zu den bestehenden Methoden.

Ausblick

Die in dieser Arbeit verwendete Methode reagiert sensibel auf lokale Änderungen und liefert einen für die Praxis nützlichen Ansatz zum Erkennen von Ereignissen. Um die Zuverlässigkeit der Methode zu verbessern, könnten folgende Aspekte miteinbezogen werden: Die Wahl der Fensterbreite spielt zum Erkennen von Ereignissen eine essentielle Rolle. Mehrere Variationen dieses Parameters abhängig von der Länge des Ereignisses könnten ausgetestet werden. Ebenfalls ein interessanter Aspekt könnte darin bestehen, zusätzliche periodische Muster in die Zerlegung einzubauen, beispielsweise Wochentag-Werktag Beziehungen oder andere saisonale Komponenten im Jahresverlauf.

A. Routinen zur Datenauswertung

In diesem Anhang finden sich einige von uns implementierte Routinen, die während dieser Arbeit verwendet wurden.

A.1. Rohdaten einlesen in Python

Dieser Code wurde verwendet um die Rohdaten (.txt bzw. .csv-Dateien) in ein R-internes einfaches Datenobjekt (.rds) umzuwandeln. Dafür wird ein Interface namens `rpy2` [pyr 2017] verwendet, dass R von Python aus ausführt.

```
import os
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
import rpy2
import rpy2.robjects as ro
import rpy2.robjects.numpy2ri

rpy2.robjects.numpy2ri.activate()

# Funktion, die x gegeben in Bytes in einen String umwandelt
def convertfunc(x):
    return dt.datetime.strptime(x.decode('ascii'),
                                '%Y-%m-%d %H:%M:%S').timestamp()

# Passenden Ordner angeben in dem Rohdaten liegen
basedir = r'C:\Users\...\Daten_Wetter'

# Daten einlesen
filename = os.path.join(basedir,'data_rain_2011.txt')
data = np.genfromtxt(filename,
                     delimiter = ';',
                     skip_header = 1,
                     converters={0: convertfunc})

# Kontrollplot (optional)
plt.plot(data[:,0],data[:,1])
plt.plot(data[:,0],data[:,2])
```

```
# Titel der zu speichernden Datei generieren
outname =  os.path.splitext(filename)[0] + '.rds'
outname = outname.replace('\\', '/') # unter Windows

# Daten in .r Objekte umwandeln und passende Spaltennamen vergeben
rdata = ro.r.matrix(data,nrow=data.shape[0], ncol=data.shape[1])
rdata.colnames = ro.StrVector(['time','klus_hn','kamo_hn'])
ro.r.assign('data', rdata)
ro.r('data <- data.frame(data)')

#r internen Zeitstempel zuordnen
ro.r('data$time <- as.POSIXct(data$time, tz="UTC", origin="1970-01-01")')

# Schreibe Daten in ein .rds File
ro.r('saveRDS(data,"{}")'.format(outname))
```

A.2. Gruppierung der Daten in R

In diesem Abschnitt befindet sich der Code zur Gruppierung der Daten in 30-Minuten Intervalle sowie zur Berechnung der Tagesmittelwerte.

Tagesmittelwerte berechnen

```
# Unterteilt Daten nach Tagen und gibt UTZ TTMMJJ um 00:00 zurueck
dat2 <- data.frame(table(cut(daten[,1], breaks = "day")))

dat_temp <- list()
start = 1
for (count in 1:nrow(dat2)){
  if(dat2[count,2]== 0){ # keine Eintraege an diesem Tag
    # generischen Eintrag erstellen
    date <- as.POSIXct( dat2[count,1], origin = "1970-01-01")
    dat_temp <-c(dat_temp,date)
    dat_temp <-c(dat_temp,rep(NA,22))
  }else{
        # vorhandene Daten mitteln und an Liste anhaengen
    end = start + dat2[count,2] - 1
    date <- as.POSIXct( dat2[count,1], origin = "1970-01-01")
    dat_temp <-c(dat_temp,date)
    dat_temp <- c(dat_temp,sapply(daten[start:end,2:23],mean,na.rm = T))
    }
    start = start + dat2[count,2]
```

```
  }
}
# in passende Matrix schreiben und Zeitstempel richtig setzen,
Spalten benennen
daten_neu <- as.data.frame(matrix( data = unlist(dat_temp),
 nrow= nrow(dat2), ncol = ncol(daten),byrow = T))
daten_neu[,1] <- as.POSIXct(daten_neu[,1], origin = "1970-01-01")
names(daten_neu) <- names(daten)

# speichern als rds Datei
setwd("C:/Users/.../Master/Daten_grupp")
saveRDS(object = daten_neu, file ="graz_sewer_tagesmittel03bis06.rds")
```

Einteilung in 30-Minuten Intervalle

```
daten # Rohdaten
dat_temp <- list()
  start = 1

  # Startzeit auf volle oder halbe Stunde setzen
  zw <- daten[1,]
  zw[1] <- as.POSIXct(((as.numeric(round(daten[start,1]-15*60,
   unit="hours")) +as.numeric(round(daten[start,1]+15*60,
   unit ="hours")))/2), origin = "1970-01-01")
  zw[1] <- zw[1]- 15*60
  zw[2:ncol(daten)] <- NA
  daten <- rbind(zw,daten)
  rm(zw)

# Unterteilt Daten in 30 Minuten Intervalle und gibt UTZ zurueck
  dat2 <- data.frame(table(cut(daten[,1], breaks = "30 mins")))

  for (count in 1:nrow(dat2)){

    if(dat2[count,2]== 0){ # keine Eintraege in dieser halben Stunde
      # generischen Platzhalter generieren
      date <- as.POSIXct( dat2[count,1], origin="1970-01-01") + 15*60
      dat_temp <-c(dat_temp,date)
      dat_temp <-c(dat_temp,rep(NA,22))
    }else{
      # vorhandene Daten mitteln und an Liste anhaengen
      end = start + dat2[count,2] - 1
      date <- as.POSIXct( dat2[count,1], origin="1970-01-01") + 15*60
      dat_temp <-c(dat_temp,date)
      dat_temp <- c(dat_temp,sapply(daten[start:end,2:23],
      mean,na.rm = T))
```

```
    }
    start = start + dat2[count,2]
  }
}
# in passende Matrix schreiben und Zeitstempel richtig setzen,
# Spalten benennen
daten_neu <- as.data.frame(matrix( data = unlist(dat_temp),
nrow = nrow(dat2), ncol = ncol(daten),byrow = T))
daten_neu[,1] <- as.POSIXct(daten_neu[,1], origin = "1970-01-01")
names(daten_neu) <- names(daten)

# speichern als rds Datei
setwd("C:/Users/.../Daten_grupp")
string <-  paste("graz_sewer_downsample_",jahrliste[counter],".rds",
sep = "")
saveRDS(object = daten_neu, file = string )
```

A.3. LOESS-Interpolation in R

Dieser Code wurde verwendet um Lücken in den Daten zu füllen. Die Routine wird in zwei Schleifen ausgeführt. In der ersten Schleife werden Lücken gefüllt, die kleiner als ein halber Tag sind. Die zweite Schleife kann größere Lücken füllen und bezieht ein tägliches Pattern in die Interpolation der Daten mit ein.

```
var_liste # Nummer der Daten zu vervollstaendigen
# Zwischenspeicherung der Daten fuer den Kontrollplot
daten_backup <- daten
# Standardbreite des Fensters zur Vervollstaendigung der Luecke
std_breite <- 12
alpha = 2/3

library(xts)

# Funktion, die Liste mit Groessen der Luecken erstellt und deren
# Index des ersten fehlenden Eintrags der Luecke
findgaps <- function(daten,var){
  liste_gaps <- list()
  bool_vec <- is.na(daten[,var])
  index = 0
  gap_size = 0

  while(index < nrow(daten)){
    index = index +1
    while(bool_vec[index] == T && index < nrow(daten)){
      gap_size = gap_size +1
```

```
      index = index +1
    }
    if(gap_size > 0){
      liste_gaps <-c(liste_gaps, index -gap_size, gap_size)
      gap_size = 0
    }
  }
  ## als Matrix zwischenspeichern
  dat_temp <- matrix( data = unlist(liste_gaps),
  nrow= length(liste_gaps)/2, ncol = 2,byrow = T)

  # Nahe Luecken zusammenfuegen
  liste_gaps <- list()
  # aktuelle (Start-)Index fuer zusammengefuegte Luecke
  tempI <- dat_temp[1,1]
  # aktuelle Groesse der Luecke
  tempS <- dat_temp[1,2]

  for (zw in 1:(nrow(dat_temp)-1)){
   # Abfrage Luecken nicht zu nahe beisammen
    if(tempI+tempS + 5 < dat_temp[zw+1,1]){
      liste_gaps <- c(liste_gaps, tempI,tempS)
      tempI <- dat_temp[zw+1,1]
      tempS <- dat_temp[zw+1,2]
    }else{
      tempS <- dat_temp[zw+1,1] - tempI + dat_temp[zw+1,2]
    }

    if(zw == nrow(dat_temp)-1)
    {
      liste_gaps <- c(liste_gaps, tempI,tempS)
    }
  }
  ## als Matrix speichern
  dat_gap <- matrix( data = unlist(liste_gaps),
  nrow= length(liste_gaps)/2, ncol = 2,byrow = T)

  return(dat_gap)
}

# Funktion, die .png Files zur optischen Ueberpruefung der Luecken-
# fuellung erstellt
kontrollplot <- function(var, counter,index_vec, daten, daten_backup){
  string = paste("loess",var,"-",counter,"-10tage",".png", sep = "")
  png(string)
```

```
  plot(index_vec,daten[index_vec,var], col = 2, main = paste("Var",var))
  points(index_vec,daten_backup[index_vec,var])
  dev.off()
}

# Hauptfunktion
for (var in var_liste){

  ## erster Schleifendurchlauf ohne Pattern
  # alle Luecken finden
  dat_gap <- findgaps(daten,var)
  # fehlende Daten mit loess fuellen
  for (counter in 1:nrow(dat_gap)) {
    if(dat_gap[counter,2]>25) # Keine Luecke groesser als ein halber Tag
    {
      next
    }

    if(dat_gap[counter,2] < 6){ #1,2,3,4,5
      breite <- std_breite
    }else{
      breite <- dat_gap[counter,2]
    }

    # Rand der Daten beachten sowie gewaehlte Fenster von verschiedenen
    # Luecken sollen sich nicht ueberschneiden
    if (counter == 1){
      pre = 1
    }else{
      pre = dat_gap[counter-1,1]
    }
    if(counter == nrow(dat_gap)){
      post = nrow(daten)
    }else{
      post = dat_gap[counter+1,1]
    }

    # Tatsaechliche Fensterbreite festlegen
    lhs <- max(1, pre, dat_gap[counter, 1] - round(alpha * breite))
    rhs <- min(dat_gap[counter, 1] + dat_gap[counter, 2] +
                        round(alpha * breite), post , nrow(daten))

    daten_temp <- daten[lhs:rhs, var]
    size = length(daten_temp)
```

```
  # Interpolation mit loess
  data_smooth <-loess(daten_temp ~ c(1:size), na.action = na.exclude,
                                              span = 1.0)
  data_fitted <- predict(data_smooth, c(1:size))

  # Interpolation an die NA Stellen der originalen Daten einfuegen
  bool_vec <- is.na(daten_temp)
  index_vec <- c(lhs:rhs)
  daten[index_vec[bool_vec], var] <- data_fitted [bool_vec]

  # Kontrollplots erstellen
  setwd("C:/Users/.../loess")
  kontrollplot(var, counter, index_vec, daten, daten_backup)
}

# fortlaufende Nummer fuer Kontrollplots in 2. Schleife
counter_bak = nrow(dat_gap)

## 2. Schleifendurchlauf, Einbeziehung des Tagespatterns

# neue Liste der Luecken (Groesse und Startindex) erstellen
dat_gap <- findgaps(daten,var)

# fehlende Daten fuellen
for (counter in 1:nrow(dat_gap)){
# Luecke groesser als X Tage ignorieren
  if(dat_gap[counter,2]>48*10)
  {
   next
  }

  if(dat_gap[counter,2] < 6){
    breite <- std_breite
  }else{
    breite <- dat_gap[counter,2]
  }

  # Rand der Daten beachten sowie gewaehlte Fenster von verschiedenen
  # Luecken sollen sich nicht ueberschneiden
  if (counter == 1){
    pre = 1
  }else{
    pre = dat_gap[counter-1,1]
  }
  if(counter == nrow(dat_gap)){
```

```
      post = nrow(daten)
    }else{
      post = dat_gap[counter+1,1]
    }
     # Luecke bei ganzen Tag starten, sonst Tagestrend verfaelscht
    indli = dat_gap[counter,1] - dat_gap[counter,1]%%48
    indre = (dat_gap[counter,1] + dat_gap[counter,2]) +
              (48 -(dat_gap[counter,1] + dat_gap[counter,2])%%48)

    # Tatsaechliche Fensterbreite festlegen
    breite = indre- indli
    lhs <- max(1,indli - 2*breite)
    rhs <- min(nrow(daten),indre + breite, post)

    # Taegliches Pattern extrahieren mittels Zeitreihen
    index_vec <- c(lhs:indli,indre:rhs)
    daten.ts <- ts(xts(daten[index_vec,var],daten[index_vec,1]),
    frequency = 48)
    dects <- decompose(daten.ts,type = "additive")
    se <- dects$seasonal


    # Daten interpolieren ohne Tagespattern
    index_vec <- c( max(48,indli-breite):rhs)
    size = length(index_vec)
    temp <- daten[index_vec,var] - se[1:size]
    data_smooth <- loess(temp~c(1:size), na.action = na.exclude,
    span = 0.5)
    data_fitted <- predict(data_smooth,c(1:size))

    # Tagespattern wieder hinzufuegen
    fit_se <- data_fitted + se[1:size]

    # Interpolation an die NA Stellen der originalen Daten einfuegen
    bool_vec <- is.na(temp)
    daten[index_vec[bool_vec],var] <- fit_se[bool_vec]

    # Kontrollplots erstellen
    setwd("C:/Users/.../loess")
    kontrollplot(var, counter+counter_bak,index_vec, daten, daten_backup)
  }
}
```

A.4. Bestimmung von Ereignissen in R

Dieser Code wurde verwendet um Ereignisse mittels Zerlegung der Zeitreihe in verschiedene Komponenten zu finden. In Standard-R-Bibliotheken sind mehrere Funktionen inkludiert, die Zeitreihen zerlegen können. Die in dieser Routine verwendete robuste Methode basiert auf der LOESS-Methode und geht auf [Cleveland u. a. 1990] zurück. Achtung: Die meisten Routinen zur Zerlegung von Zeitreihen können nur angewendet werden, wenn sich keine NAs in der zu zerlegenden Zeitraum befinden. Die Routine `stl` kann auch mit fehlenden Werten umgehen, allerdings sollten keine zu großen Lücken vorhanden sein, da ein plausible Zerlegung nicht mehr garantiert werden kann.
Folgende Funktionen wurden implementiert:

- `pot_ev`: findet potentielle Ereignisse einer Variablen,
- `kontrollplot`: speichert Kontrollplots in Ordner (wird nur in `pot_ev` verwendet),
- `kombi_ev`: verknüpft Ereignislisten mit logischen ‚Und' bzw. ‚Oder' sowie
- `make_event_list`: generiert aus einer Liste von IDs eine Matrix mit Startzeitpunkt und Dauer des Ereignisses.

```
# Initialisieren
# Bibliothek fuer Zeitreihen und Zerlegung
library(xts)


## Kontrollplots in Ordner speichern
# in Funktion pot_events verwendet
kontrollplot <- function(daten, var,day, index_vec, bool_vec1,
                                    bool_vec2, bool_vec3){
  # Kontrollplot
  setwd("C:/Users/.../events/5Tage")
  string = paste("dec",var,"-",day,".png", sep = "")
  png(string)
  plot(daten$time[index_vec],daten[index_vec,var], main = var)
  points(daten[index_vec[bool_vec1],1],daten[index_vec[bool_vec1],var],
  col = 2)
  points(daten[index_vec[bool_vec2],1],daten[index_vec[bool_vec2],var],
  col = 3, pch = "o")
  points(daten[index_vec[bool_vec3],1],daten[index_vec[bool_vec3],var],
  col = 4, pch = "x")
  dev.off()
}


### Hauptfunktion

## Parameter
```

```
# zu untersuchende Daten
setwd("C:/Users/.../daten_kompakt")
daten <- readRDS("graz_sewer_loess09-11.rds" )

var = 3 # Variable (2...4)
win_len <- 7*48 # gerade Fensterbreite

##Funktion Potentielle Ereignisse bestimmen mittels Kriterien 1 bis 3
# gibt Liste der Ereignisse einer Variable getrennt nach Kriterien zurueck
# Standardfensterbreite: 1 Woche
pot_events <- function(daten, var, win_len = 7*48){
  shift <- win_len/2

  # Ausgabelisten initialisieren
  aus_k1 <- list()
  aus_k2 <- list()
  aus_k3 <- list()

  # day Startindex des aktuell betrachteten Fensters
  day = 1

  while (day + win_len < nrow(daten)){
    print(day)
    # Index Vektor
    index_vec <- c(day:(day + win_len- 1))

    # Wenn NA in aktuellen Fenster, Schleifendurchlauf ueberspringen
    # da Zerlegung nicht mit NAs funktioniert
    if (sum(is.na(daten[index_vec,var]))>0){
      day = day + shift
      print("In current window are NAs included.")
      next
    }

    ## Zerlegen der Zeitreihe
    # rand: zufaelliger Anteil, seas: saisonaler Anteil, trend: Trend
    # dtrend: numerische Aenderung des Trends
    daten.ts <- ts(xts(daten[index_vec,var],daten[index_vec,1]),
    frequency = 48)
    dects <- stl(daten.ts[,1],s.window = "per",robust =T)
    rand <- dects$time.series[,"remainder"]
    seas <- dects$time.series[,"seasonal"]
    trend <- dects$time.series[,"trend"]
    dtrend <- diff(c(trend[1],trend))
```

```
    #potentielle Ausreisser bestimmen mittels Kriterien 1 bis 3

    # Nr 1: Random Anteil nicht zu gross
    # mittels Schaetzer sigma (IQR)
    bool_vec1 <- (rand > 3*IQR(rand,na.rm = T))

    # Nr 2 Daten innerhalb Schwankung Tagespattern

    bool_vec2 <- abs(daten.ts - trend) > 2 * abs(max(seas)- median(seas)) |
                 abs(daten.ts - trend) > 2 * abs(min(seas)- median(seas))

    # Nr 3. zu starke Aenderung des Trends
    bool_vec3 <- (abs(dtrend) > 3 *IQR(dtrend) )


    #Kontrollplot
    if(sum(bool_vec1 + bool_vec2+ bool_vec3) > 1){
      kontrollplot(daten, var, day, index_vec, bool_vec1,
                                  bool_vec2, bool_vec3)
    }

    aus_k1 <- c(aus_k1,index_vec[bool_vec1])
    aus_k2 <- c(aus_k2,index_vec[bool_vec2])
    aus_k3 <- c(aus_k3,index_vec[bool_vec3])

    day = day + shift # Fenster um halbe Woche verschieben
  }
  return(list(aus_k1,aus_k2,aus_k3))
}

## Events kombinieren

# zwei Index-Listen von Events mit and(T), oder (F) verknuepfen
kombi_ev <- function(ev1,ev2,bool_and){
  kombi <- c(ev1,ev2)
  if (bool_and == T){
    ev <- kombi[duplicated(kombi)]
  }else{
    ev <- kombi[!duplicated(kombi)]
  }
  return(ev)
}

## Generiert aus einer Liste von sortierten IDs eine Matrix
# mit Startzeitpunkt und Dauer aller gefundenen Events
```

```
make_event_list <- function(bool_plot){
  ind = 1
  counter = 0
  start_list <- list()

  while (ind < length(bool_plot)){

    while((bool_plot[ind + counter]+1) == (bool_plot[ind+counter+1])){
      counter = counter +1
      if (ind + counter + 1 > length((bool_plot)) ){
        break
      }
    }
    start_list <- c(start_list,bool_plot[ind], counter + 1)
    ind = counter + ind +1
    counter = 0
  }

  events <- matrix(unlist(start_list),ncol = 2,byrow =T)
  rownames(events) <- c("Startindex","Breite")

  return(events)
}
```

Literaturverzeichnis

[R 2017] : *R: an open source programming language for statistical computations and graphics*. `https://cran.r-project.org/`. 2017

[pyr 2017] : *rpy2: an interface to run R embedded in a Python process*. `https://rpy2.bitbucket.io/`. 2017

[Brockwell und Davis 2002] Brockwell, Peter J. ; Davis, Richard A.: *Introduction to Time Series and Forecasting*. 2. Springer, 2002

[Cleveland u. a. 1990] Cleveland, Robert B. ; Cleveland, William S. ; McRae, Jean E. ; Terpenning, Irma: *STL: A Seasonal-Trend Decomposition Procedure based on Loess*. S. 3–73, Journal of Official Statistics Vol. 6 No.1, 1990

[Cleveland 1981] Cleveland, William S.: *LOWESS: A program for smoothing scatterplots by robust locally weighted regression*. The American Statistician 35 (1), 1981

[Hofer 2012] Hofer, Thomas F.: *Validierung, Charakterisierung und Klassifizierung von Mischwasserereignissen im Einzugsgebiet Graz-West R05*. Institut für Siedlungswasserwirtschaft und Landschaftswasserbau an der Technischen Universität Graz, 2012

[Mourand und Bertrand-Krajewski 2002] Mourand, E. ; Bertrand-Krajewski, J.-L.: *A methode for automatic validation of long time series of data in urban hydrology*. Water Science and Tecnology 45 (4-5), 2002

[Ryan u. a. 2017] Ryan, J.A. ; Ulrich, J.M. ; Benett, R.: *xts (R package): eXtensible Time Series*. `https://CRAN.R-project.org/package=xts`. 2017