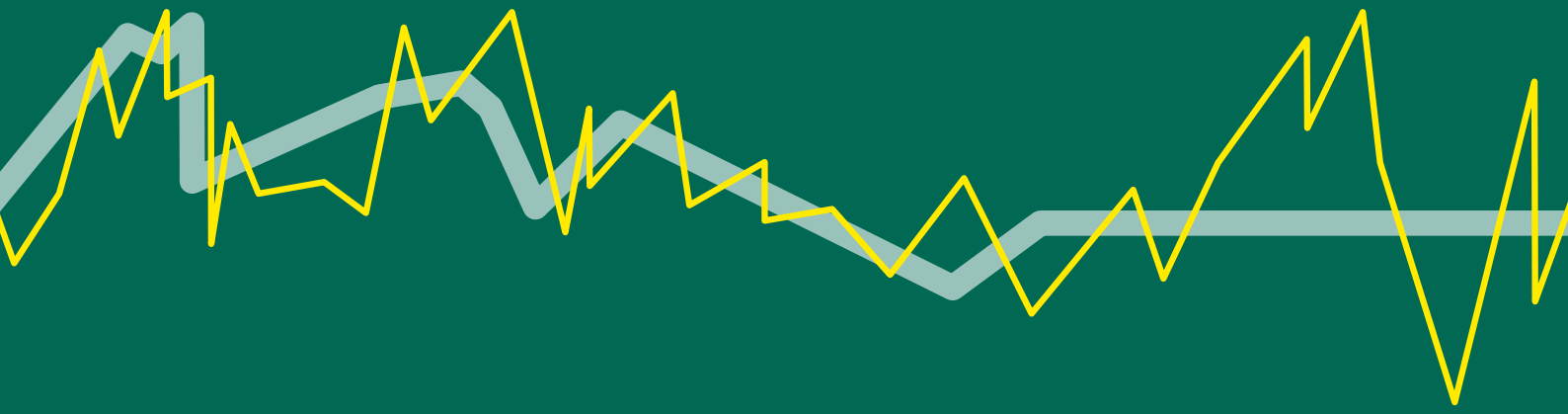


# Einführung und Verarbeitung von Zeitserien

**Isabella Tran**
**Proseminar – Programmierung in R**
**Betreuer: Dr. Julian Kunkel**

# Inhaltsverzeichnis


**Kapitel 0:** Zusammenfassung **3**

**Kapitel 1:** Einführung und Definition **4**

**Kapitel 2:** Arbeitsphasen bei Zeitreihenanalysen **5**

**Kapitel 3:** Programmierung in R **6**

**Kapitel 4:** ARIMA Modell **10**

**Kapitel 5:** Quellenverzeichnis **12**

# Zusammenfassung

Bei der Einführung und Verarbeitung von Zeitserien werden folgende Fragen behandelt:

1. Was ist eine Zeitreihe(nanalyse)?
2. Welchen Nutzen haben sie?
3. Wo werden sie verwendet?
4. Wie ist der Ablauf einer Zeitreihenanalyse?
5. Wie funktioniert es in R?
6. Was ist das ARIMA-Modell?

**1.**
Eine Zeitreihe ist eine Abfolge gemessener Datenpunkte in Abhängigkeit zu einem bestimmten Zeitraum.
Eine Zeitreihenanalyse ist die Verarbeitung und Auswertung dieser Daten.

**2.**
Zeitserien werden verwendet, um zukünftige Prognosen zu erstellen oder vergangene Daten auszuwertenund so über historische Aspekte zu prognostizieren.

**3.**
Anwendung finden Zeitserien bei Wetterbeobachtungen, Börsenkurse, Messungen zum Bevölkerungswachstum, EKG- und EEG-Messungen, in der Finanzmathematik und vielen weiteren Bereichen.

**4.**
Zeitreihenanalysen kann man in vier Phasen unterteilen:

- Identifikationsphase
- Schätzphase
- Diagnosephase
- Einsatzphase

**5.**

1. vector <- c(...) / read.csv(...)
2. Ggf. ts(vector)
3. plot(...) / boxplot(...)

*- mit forecast-package: -*

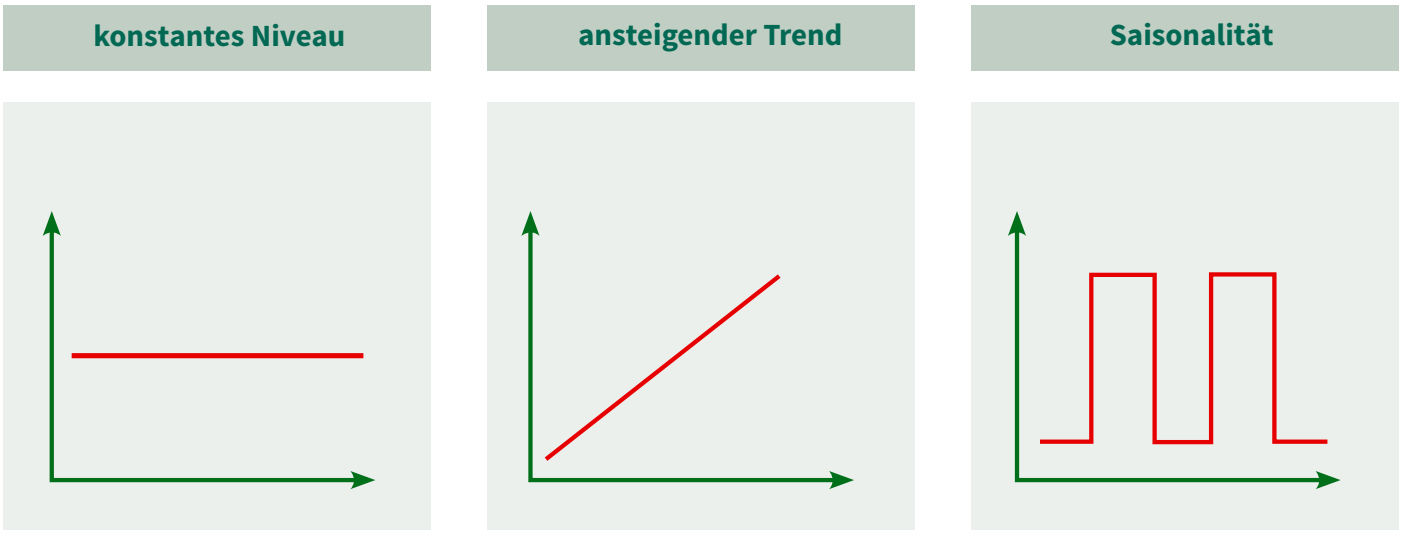
4. seasonplot(...)

**6.**
Das ARIMA-Modell ist eine sehr leistungsstarke Modellklasse, welche auf viele reale Zeitreihen angewendet wird.

# Einführung

Zur Einführung eines Themas gehört wie üblich eine Definition, in diesem Fall eine Definition zum Überbegriff Zeitserien. Was ist eine Zeitserie und welchen Nutzen dient sie? Eine Zeitserie ist wie der Name schon verrät eine Abfolge gemessener Datenpunkte in Abhängigkeit zu einem bestimmten Zeitraum. Beispiele für Zeitserien sind: Wetterbeobachtungen, Börsenkurse, Messungen zum Bevölkerungswachstum, EKG-Messungen...usw. Typischerweise wirken sowohl regelhafte wie auch zufällige Ursachen zusammen auf die Zeitreihe ein. Bei der Auswertung dieser Daten (genannt: Zeitreihenanalyse) können diese periodisch (saisonal) variieren oder auch langfristigen Trends folgen (siehe Abb. 1). In Bereichen wie die Biometrie, die Vegetationsentwicklung, die Finanzmathematik und die Meteorologie  wird man häufig mit Zeitreihenanalysen konfrontiert.

> *„Zeitreihen sind ganz allgemein Sammlungen von Werten, die in zeitlicher Folge beobachtet wurden.“*
>
> (Definition einer Zeitreihe, http://www.fernuni-hagen.de/ksw/neuestatistik/content/MOD_24269/html/comp_24274.html#kopf)




*(Abb. 1) Muster, die auf Zeitreihen bei der Analyse angewendet werden können*

# Arbeitsphasen bei Zeitreihenanalysen

Eine Zeitreihenanalyse kann man in vier Phasen unterteilen: die Identifikations-, die Schätz, die Diagnose- und die Einsatzphase.
Zuallererst, beginnend mit der Identifikationsphase, versucht man die gemessene Zeitserie grafisch darzustellen. Bei der Modellierung der Zeitserie lassen sich anhand grafischer Analyse oder Anwendung statischer Tests (wie bsp. den Dickey-Fuller-Test[1]) Trends herausfiltern. Auch überprüft man die Zeitreihe auf periodische Wiederholungen (Saisonalität) und vereinzelte Außreißern. Durch Transformation bzw. Differenzierung der Zeitreihe (bsp. der Box-Cox-Transformation[2]) stabilisiert man die durch die Außreißer verursachten Varianzen. Es bildet sich eine gute Grundlage zur weiteren Verarbeitung.
Während der Schätzphase stellt man eine Vermutung von den Modellparametern und -koeffizienten der Zeitserie auf, um in der nachfolgenden Phase die Zeitserie zu diagnostizieren. Bekannte Verfahren sind hierbei die OLS-Methode[3] oder auch der Box-Jenkins-Ansatz[4].
Während der Diagnosephase beurteilt man die Qualität der Zeitreihe. Als mögliche Vorgehensweise bietet es sich an zu überprüfen, ob die geschätzten Koeffizienten sich signifikant von Null unterscheiden. Falls man nach der Box-Jenkins Methode verfährt, überprüft man zusätzlich, inwieweit die erschlossenen Koeffizienten mit den empirischen Autokorrelationskoeffizienten übereinstimmen, sprich: inwiefern sich der gemessene Wert zu sich selbst zu einem früheren Zeitpunkt unterscheidet. Zusätzlich kann man auch das Spektrum analysieren, in welches die Koeffizienten voneinander abweichen. Letztlich analysiert man die Residuen, die Abweichungen vom erwarteten Ergebnis.
Zu guter Letzt in der Einsatzphase formuliert man Prognosegleichung zur befundenen Modellgleichung, um zukünftige Werte zu vermuten wie kommende Trends bei Börsenkursen oder Wettervorhersagen.

[1] Dickey-Fuller-Test: auch Einheitswurzeltest genannt; Test zur Festzustellung, ob integrierte Prozesse vorliegen

[2] Box-Cox-Transformation: Verfahren zur Regressionsanalyse zur Stabilisierung von Varianzen (Ausreißern)

[3] OLS-Methode: Methode der kleinsten Quadrate

[4] Box-Jenkins-Ansatz: auch Momentenmethode genannt, ist ein Schätzverfahren, bei der man Werte in Abhängigkeit vom Moment ihrer Verteilung ausdrückt (in einem Spektrum)

# Programmierung in R

Bei Zeitreihenanalysen in R so wie auch in anderen Programmiersprachen sollte folgendes beachtet werden:

1. Eine vorhandene Quelle von zeitabhängig gemessenen Daten
2. Die Zeitserie muss vorbereitet werden
3. Das Plotten der Zeitserie zur grafischen Darstellung

Als Beispiel nehmen wir die gemessenen Temperaturen der letzten 48 Monate in Hamburg und möchten diese grafisch modellieren. Zur Bereitstellung der Daten gibt es zwei Möglichkeiten. Die erste wäre, die Daten in R einem Vektor zu übergeben. Dies tun wir mit der Funktion *c(...)*. Anschließend wandeln wir die Datei in ein Zeitserienobjekt mithilfe der Funktion *ts(...)* um. Hierbei übergeben wir der Funktion als Argumente zuallererst unseren Vektor, als auch den *start* und *end*-Wert unserer Zeitserie. Mit dem Argument *frequency* bestimmen wir zusätzlich die Anzahl der Messungen in diesem Zeitraum. Der Vorteil dieser Methode ist, dass alle Daten in derselben Datei verfügbar sind. Allerdings kann diese Methode bei größeren Zeitserien schnell unübersichtlich werden, welches uns zur zweiten Möglichkeit bringt.

```
monthVector <- c(2.8,0,7.2,7.8,13.6,14.6,17.2,17.9,13.7,9.6,6,1.6,1.3,0.6, 0.
4,7.6,12.6,15.2,18.9,18,13.5,11.4,5.8,5.3,1.7,5.3,7.3,10.6,12.6,15.9,20.4,16.5
,15.9,13,7.1,3.4,3.2,2.3,5.9,8.2,11.3,14.8,17.9,18.8,13.4,9.1,8.1,7.9)


Monatstemperaturen <- ts(monthVector, start=c(2011, 1), end=c(2014,
12), frequency=12)
```

Das entstandene Zeitserienobjekt sieht folgendermaßen aus:

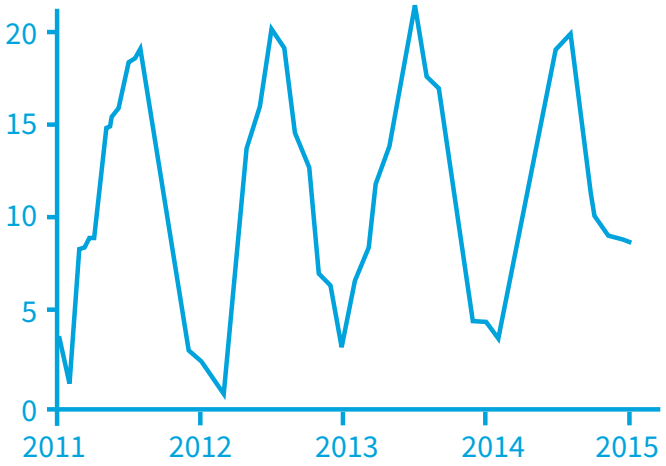
| | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2011 | 2.8 | 0.0 | 7.2 | 7.8 | 13.6 | 14.6 | 17.2 | 17.9 | 13.7 | 9.6 | 6.0 | 1.6 |
| 2012 | 1.3 | 0.6 | -0.4 | 7.6 | 12.6 | 15.2 | 18.9 | 18.0 | 13.5 | 11.4 | 5.8 | 5.3 |
| 2013 | 1.7 | 5.3 | 7.3 | 10.6 | 12.6 | 15.9 | 20.4 | 16.5 | 15.9 | 13.0 | 7.1 | 3.4 |
| 2014 | 3.2 | 2.3 | 5.9 | 8.2 | 11.3 | 14.8 | 17.9 | 18.8 | 13.4 | 9.1 | 8.1 | 7.9 |

Bei größeren Datenmengen können wir in R CSV-Dateien lesen und verarbeiten. Hierzu verwenden wir die Funktion *read.csv(...)*. CSV-Dateien kann man aus einigen Tabellenprogrammen (wie Excel oder OpenCalc) exportieren. Zum Arbeiten mit der Datei sollte diese im Arbeitsverzeichnis *(engl. Work Directory)* gespeichert werden. Mit der Funktion *getwd()* und *setwd(...)* kann man das aktuelle Arbeitsverzeichnis aufrufen und falls erforderlich neu setzen. Ist dies geschafft, lässt sich auch die externe Datei mit *read.csv(...)* in R einlesen. Bei Anwendung dieser Methode muss die eingelesene Datei nicht explizit in ein Zeitserienobjekt umgewandelt werden.

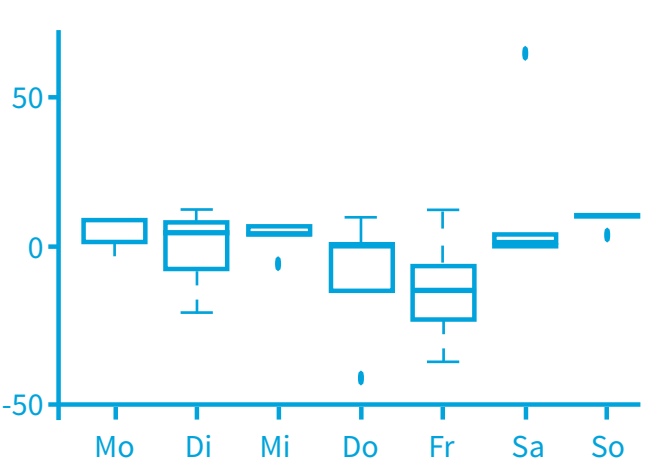
```
getwd()
setwd(„../Proseminar - Programmierung in R/“)
Monatstemperaturen <- read.csv(monatstemperaturen.csv)
```

Die einfachste grafische Darstellung der Zeitserie gelingt mit der Funktion plot(...), dem wir unser Zeitserienobjekt bzw. unsere CSV-Datei übergeben. R erzeugt mit der Funktion ein minimalistisches Diagramm der Temperaturen in den vergangenen 4 Jahren. Ein ebenfalls einfache Plot-Funktion ist der sogenannte boxplot(...). Dieser Plot zeigt das Spektrum, in welchem sich die meisten Zeitwerte befinden und auch einige Ausreißer. Vorteilhaft ist dieser Plot bei wiederholenden Zeitreihen, beispielsweise bei Umsatzanalysen in Unternehmen.

```
plot(Monatstemperaturen)
```

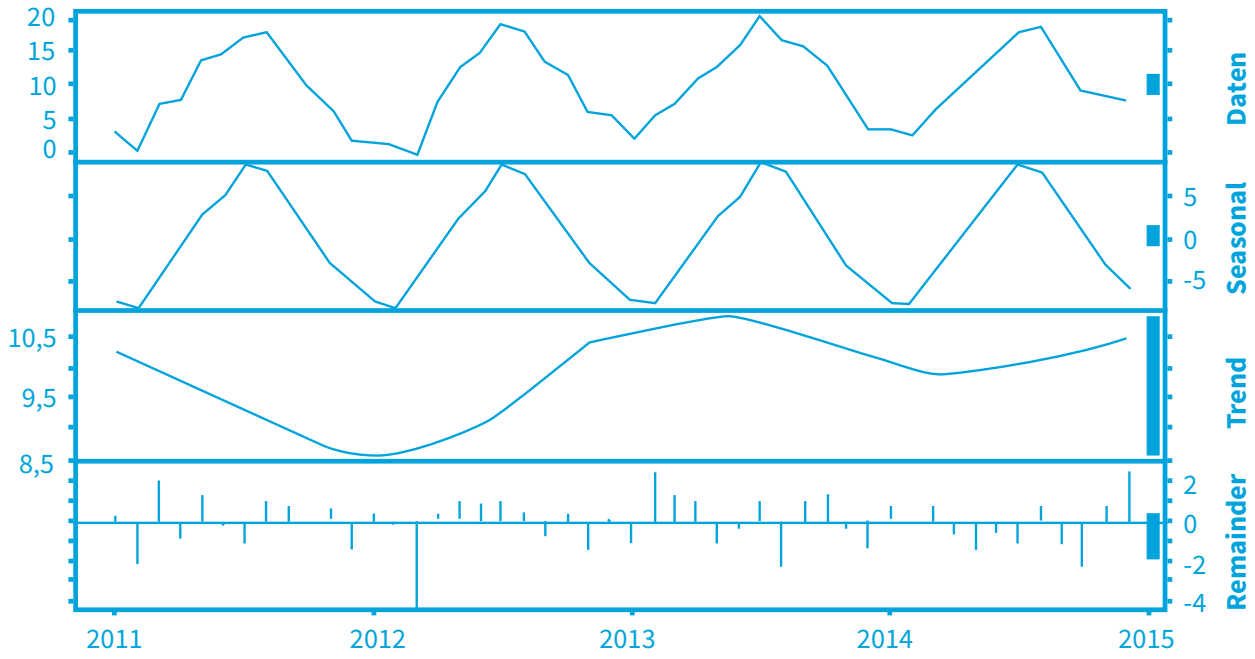



```
boxplot(Umsatz2016)
```



Weitere Verarbeitungsmöglichkeiten von Zeitreihen bieten uns die Funktionen *stl(...)* und *decompose(...)*. Die Funktion *stl(...)* filtert mithilfe des loess-Verfahrens und zerlegt die Zeitserie und vier Komponente: die darliegende Zeitserie, wie wir sie mit der allgemeinen Funktion plot(...) erhalten, den Trend, der Saisonalität und der Restkomponente. Man übergibt der Funktion das Zeitserienobjekt. Es können zusätzliche Argumente hinzugefügt werden wie bsp. *s.window* zur Bestimmung des Bereichs unserer saisonalen Extraktion.
Mit der Funktion *decompose(...)* kann die Zeitserie ebenfalls in die noben genannten Bestandteile zerlegen; sie basiert jedoch auf dem MA-Verfahren und behandelt zudem additive wie multiplikative saisonale Komponente. Additive Modelle sind nützlich bei  relativ konstanten saisonalen Veränderungen in der Zeitreihe. Andernfalls verwendet man multiplikative Methode, um die Zeitreihe für Prognosen anzupassen.

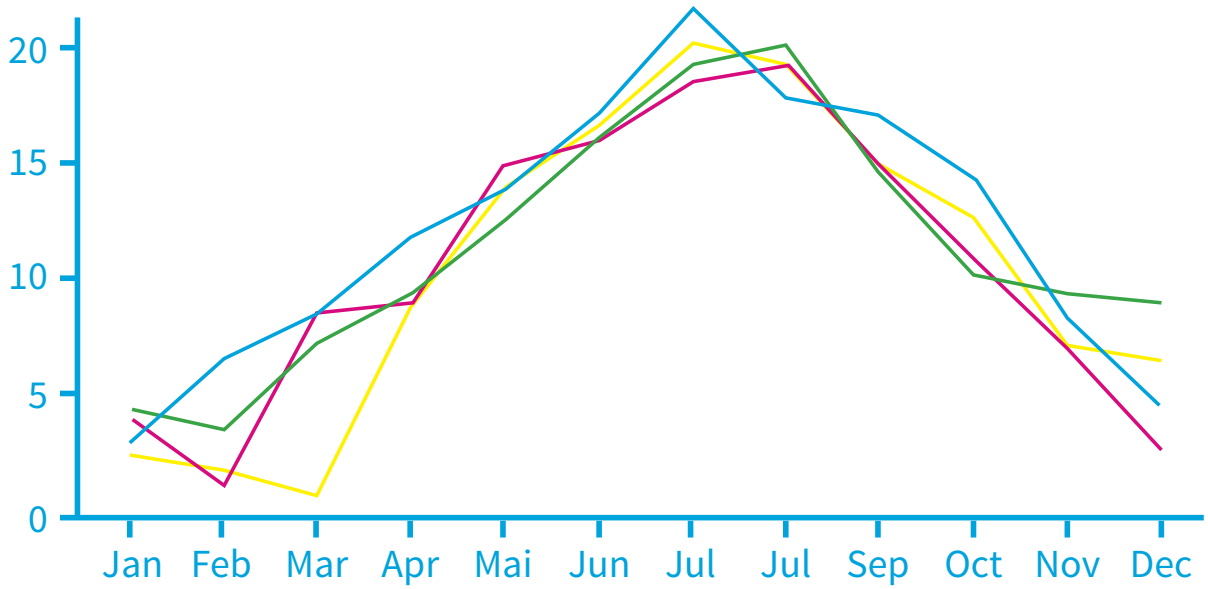
```
fit <- stl(Monatstemperaturen, s.window="period")
plot(Monatstemperaturen)
```




In R stehen noch Pakete zur Verfügung, welche man zur Verarbeitung von Zeitserien ebenfalls verwenden kann. Eines davon ist das Paket forecast. Nach dem Herunterladen und Installieren dieses Paketes und dem Hinzufügen in die R-Bibliothek lässt sich ein saisonaler Plot mit der Funktion seasonplot(...)

auf unsere Zeitserie anwenden. Die einzelnen Messungen lassen sich zur Veranschaulichung mit dem Argument *col* auch farblich voneinander trennen. Die Farben sind nach der Reihenfolge den Zeiten zugeteilt (2011-12=“magenta“, 2012-13=“yellow“, 2013-14=“blue“, 2014-15=“green“).

```
library(forecast)
seasonplot(Monatstemperaturen,
col=c(„magenta“,“yellow“,“blue“,“green“))
```



# ARIMA Modell

Das ARIMA-Modell (engl. „Auto-Regressive Integrated Moving Average Model) ist eine sehr leistungsstarke und verbreitete Modellklasse zur Prognosenbestimmung. Das Modell führt das vorangehende ARMA-Modell weiter. Es besteht aus einem autoregressiven Teil (AR-Modell), dem gleitenden Mittelwertbeitrag (MA-Modell) und der Umfassung der ersten Ableitung der Zeitreihe.
Beim autoregressiven Teil versucht man die Messwerte durch vorangegangene Beobachtungen anhand von linearen Modellen (bsp. x(t) = 2x(t - 1) + x(t - 2)) zu beschreiben. Die allgemeine Formel von AR-Modellen lautet:

$$x(t) = \sum_{i=1}^{p} \alpha x_i(t\text{-}i)$$

t: Zeit
$\alpha$: Parameter
p: Ordnung des Modells

Der gleitende Mittelwertbeitrag (Moving-Average Modell) geht von Schätzungsfehlern bei den Zeitserien aus. Vorangegangene Schätz- oder Vorhersagefehler werden bei der Schätzung des nächsten Wertes in der Zeitserie miteinberechnet. Die allgemeine Formel von MA-Modellen lautet:

$$x(t) = -\sum_{i=1}^{q} \beta_i \varepsilon(t\text{-}i)$$

t: Zeit
ß: Parameter
$\varepsilon$: Unterschied zwischen der Schätzung und dem wirklichen Wert
q: Ordnung des Modells

Durch Kombination der beiden vorgegangenen Modelle entsteht das ARMA-Modell, auch bekannt als Box-Jenkins-Modell. Die allgemeine Formel von ARMA-Modellen lautet also:

$$x(t) = \sum_{i=1}^{p} \alpha x_i(t\text{-}i) - \sum_{i=1}^{q} \beta_i \varepsilon(t\text{-}i)$$

Durch zusätzliche Differenzierung und einer Integration nach Anwendung des Modells kommen wir letztlich zum ARIMA-Modell. Diese werden verwendet, wenn man ein Trend herausfiltern möchte. Der Parameter d des ARIMA-Modells

bestimmt die Zahl der Differenzierungsschritte: Zunächst leitet man die Zeitserie d-mal ab, bis sie stationär (zeitunabhängig) ist.
Dann wird das ARMA-Modell an die abgeleitete Serie angepasst. Zuletzt integriert man d-mal die geschätzten Voraussagen und ist fertig. Dies ist nur eine vieler Varianten des ARIMA-Modells.
In R ist die Funktion *arima(p,d,q)* bereits implementiert.

# Quellenverzeichnis

## elektronische Informationsquellen


**„Definition einer Zeitreihe“**
http://www.fernuni-hagen.de/ksw/neuestatistik/content/MOD_24269/html/comp_24274.html

**„Zeitreihenanalyse“**
de.wikipedia.org/wiki/Zeitreihenanalyse
02.05.2016

**„Time series“**
en.wikipedia.org/wiki/Time_series
02.05.2016

**„Time Series Analysis with R - Part I“**
Walter Zucchini & Oleg Nenadic
www.statoek.wiso.uni-goettingen.de/veranstaltungen/zeitreihen/sommer03/ts_r_intro.pdf
02.05.2016

**„Autoregressive integrated moving average“**
en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
02.05.2016

**„CRAN Task View: Time Series Analysis“**
cran.r-project.org/web/views/TimeSeries.html
02.05.2016

**„ARIMA Modelling of Time Series“**
stat.ethz.ch/R-manual/R-devel/library/stats/html/arima.html
02.05.2016

**„R - Basic Syntax“**
www.tutorialspoint.com/r/r_basic_syntax.htm
02.05.2016

**„Methode der kleinsten Quadrate“**
de.wikipedia.org/wiki/Methode_der_kleinsten_Quadrate
02.05.2016

**„Time Series and Forecasting Methods in NCSS“**
www.ncss.com/software/ncss/time-series-and-forecasting-in-ncss
02.05.2016

**„Methode der kleinsten Quadrate“**
de.wikipedia.org/wiki/Methode_der_kleinsten_Quadrate
02.05.2016

**„Monats- und Jahreswerte für Hamburg“**
www.wetterkontor.de/de/wetter/deutschland/monatswerte-station.asp
02.05.2016

**„Mildes Wetter im Januar: Sind richtige Winter Schnee von gestern?“**
Walter Schmeißer
www.wsgonline.de/winter/mildes-wetter-im-januar-sind-richtige-winter-schnee-von-gestern
02.05.2016

**„Zeitreihen - Definition von ARIMA-Modellen“**
Hans Lohninger
www.statistics4u.info/fundstat_germ/cc_timeser_arima.html
02.05.2016

**„STL: A Seasonal-Trend Decomposition Procedure Based on Loess“**
Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning
cs.wellesley.edu/~cs315/Papers/stl%20statistical%20model.pdf
1990


## literarische Quellen


**„Time Series Analyses With Application in R“**
Jonathan D. Cryer & Kung-Sik Chan
2009