



**School of  
Engineering**

IDP Institut für Datenanalyse  
und Prozessdesign

## **Projektarbeit** **(Wirtschaftsingenieurwesen)**

Algorithmischer Handel mit  
makroökonomischen  
Erklärungsvariablen

---

<b>Autoren</b>	Adrian Genoud, Severin Holzer
----------------	-------------------------------

---

<b>Hauptbetreuung</b>	Marc Wildi
-----------------------	------------

---

<b>Industriepartner</b>	InCube AG
-------------------------	-----------

---

<b>Datum</b>	18.12.2015
--------------	------------

# Vorwort

Wir möchten uns ganz herzlich bei unserem Dozenten Herrn Wildi und der Firma InCube, namentlich Herr Blakely, für die grossartige Unterstützung und das Ermöglichen dieser Projektarbeit bedanken.

# Zusammenfassung

Für Langzeitanleger bilden Rezessionen grosse Gefahren. In diesen können sehr hohe Verluste entstehen, da die Aktienmärkte stark darauf reagieren. Das Ziel dieser Arbeit ist es, einen Indikator zu erstellen, welcher eine Rezession möglichst schnell erkennt, um entsprechende Warnsignale geben zu können. Aus den Daten des Bruttoinlandprodukts der USA (**G**ross **D**omestic **P**roduct - GDP) sind die vergangenen Rezessionen sehr gut ersichtlich. Darum wurde es als Ziel für die Berechnung des Indikators verwendet. Die neusten Daten des GDP sind noch nicht korrekt, sondern werden über ca. ein Jahr zurück revidiert. Um dieser Tatsache entgegen zu wirken musste das GDP nachgebildet und für das letzte Jahr geschätzt werden, damit es in allen Zeitpunkten möglichst genau ist. Das GDP wurde mittels einer Regression nachgebildet. Als erklärende Variablen kann im R-Code jede beliebige Zeitreihe benutzt werden. In dieser Arbeit verwendet man hauptsächlich folgende Daten: S&P 500, VIX S&P 500, Initial Claims, Industrial Production Index, Non Farm Payroll, Civilian Unemployment Rate. Anschliessend berechnet der Indikator aus dem geschätzten GDP eine Signalgrenze. Falls das geschätzte GDP heute unter dieser Grenze liegt, besteht die Gefahr einer Rezession und es wird ein entsprechendes Signal übermittelt. In verschiedenen Tests hat der Indikator mit den optimalen Einstellungen 39.8% Vorteil gegenüber dem S&P 500 erzielt. Dieser wurde vor allem in den Rezessionen erreicht. Somit hätte sich eine Investition ab 1997, mit der Strategie des Indikators, mehr gelohnt, als eine fixe Investition auf den S&P 500.

# Abstract

For long-term investors, recessions are a major threat. In this case, high losses could be a result because the share markets strongly react to recessions. The goal of this report is to create an indicator, which detects a recession at an early stage in order to send out alarms accordingly. The last recessions are clearly visible from the data of the gross domestic product (USA). That's the reason why we take it as main part for the calculation of the indicator. The recent GDP data are not correct yet and are being revised one year back. To compensate this fact we had to reproduce the GDP and estimate it for the last year, so that it is as accurate as possible at any point in time. The GDP was simulated by a regression. As explanatory variables, it's possible to use any time series in the R-Code. In this report we mainly used the following data: S&P 500, VIX S&P 500, initial claims, industrial production index, non farm payroll, civilian unemployment rate. Next, from the estimated GDP the indicator calculates a signal limit. If the estimated GDP is under this limit, we have the danger of a recession and a corresponding signal will be sent out. In different tests, the indicator with the optimal settings produced an advantage of 39.8 % over the S&P 500. This was achieved mostly in recessions. As a result, it was more remunerative to invest according to the indicator's strategy than a fix investment in the S&P 500.

# Inhaltsverzeichnis

<b>1 Einleitung .....</b>	<b>3</b>
<b>2 Datensätze .....</b>	<b>4</b>
2.1 Vorgegebene Daten.....	4
2.2 Zusätzliche Daten .....	4
<b>3 Indikator .....</b>	<b>6</b>
3.1 Main.....	6
3.1.1 Beschreibung .....	6
3.1.2 Probleme .....	7
3.2 Read .....	7
3.2.1 Beschreibung .....	7
3.2.2 Probleme .....	8
3.3 Filter.....	8
3.3.1 Beschreibung .....	8
3.3.2 Probleme .....	9
3.4 Regression.....	9
3.4.1 Beschreibung .....	9
3.4.2 Probleme .....	10
3.5 Signal.....	10
3.5.1 Beschreibung .....	10
3.5.2 Problem.....	10
<b>4 Performance .....</b>	<b>11</b>
4.1 R-Code .....	11
4.1.1 Test .....	11
4.1.1.1 Beschreibung.....	11
4.1.1.2 Probleme .....	11
4.1.2 Performance.....	11
4.1.2.1 Beschreibung.....	12
4.1.2.2 Probleme .....	12
4.2 Filter: Gleitender Durchschnitt .....	13
4.2.1 Indikator mit vorgegebenen Daten .....	13
4.2.2 Indikator mit zusätzlichen Daten.....	17
4.2.3 Fazit .....	19

4.3 Filter: Direct Filter Approach .....	19
4.3.1 Indikator mit vorgegebenen Daten .....	19
4.3.2 Indikator mit zusätzlichen Daten.....	23
4.3.3 Fazit .....	25
4.4 Vergleich: DFA vs. gleitender Durchschnitt.....	25
4.5 Fazit .....	28
<b>5 Indikator vs. GDP-Indikator .....</b>	<b>29</b>
<b>6 Optimierung .....</b>	<b>33</b>
<b>7 Schlusswort .....</b>	<b>37</b>
<b>8 Verzeichnisse .....</b>	<b>38</b>
8.1 Literaturverzeichnis.....	38
8.2 Abbildungsverzeichnis .....	38
8.3 Tabellenverzeichnis .....	39
8.4 Datenverzeichnis .....	39
<b>9 Anhang .....</b>	<b>41</b>
9.1 Aufgabenstellung .....	41
9.2 R-Code .....	42
9.2.1 Indikator .....	42
9.2.1.1 Main.....	42
9.2.1.2 Read .....	46
9.2.1.3 Filtern.....	48
9.2.1.4 Regression.....	49
9.2.1.5 Signal.....	51
9.2.2 Unterfunktionen .....	52
9.2.2.1 Lin_interpol .....	52
9.2.2.2 filt_gleit_D.....	53
9.2.2.3 dfa_ms.....	54
9.2.2.4 per .....	54
9.2.2.5 correlation.....	55
9.2.3 Test .....	56
9.2.3.1 Test.....	56
9.2.3.2 Performance .....	58
9.2.3.3 GDP-Indikator.....	60

# 1 Einleitung

Um herauszufinden, ob jetzt ein guter Zeitpunkt ist, um Geld zu investieren, gibt es in der heutigen Zeit viele verschiedene Indikatoren, mit denen man erkennt, ob es der Wirtschaft in der USA gut oder schlecht geht. Das Bruttoinlandprodukt der USA (**G**ross **D**omestic **P**roduct - GDP) ist einer davon. Das Problem ist aber, dass die aktuellen Zahlen sehr ungenau sind und über einen grossen Zeitraum zurück revidiert werden müssen. Darum können nur die Daten bis ungefähr vor einem Jahr für einen verlässlichen Indikator verwendet werden. Folglich ist das GDP für die Echtzeitanwendung eine schlechte Lösung.

Ziel dieser Arbeit ist es, trotz dieses Problems einen Indikator auf Basis des GDP zu erstellen, welcher für den S&P 500 ein Signal gibt, ob man heute „long“ oder „cash“ gehen soll. Hauptsächlich wird dieser von Langzeitanlegern (bspw. Pensionskassen) benutzt, für welche in erster Linie nur grosse Verluste von Bedeutung sind. Diese treten meistens während einer Rezession auf.

Das Problem mit den revidierten Daten wird umgangen, indem der Indikator nicht mit dem originalen GDP rechnet, sondern mit einem geschätzten, welches man mittels einer Regression ermittelt. Der Indikator selbst wird in vier Teile gegliedert. Der erste Teil ist das Einlesen der Zeitreihen, die für die Regression benutzt werden sollen. Sie werden vom öffentlichen Datenportal Quandl heruntergeladen. Die Zeitreihen können vom Benutzer beliebig gewählt werden. Im zweiten Teil folgt das Filtern des Rauschens. Danach findet im dritten Teil eine Regression mit den gefilterten Zeitreihen statt. Im letzten Teil wird mit dem geschätzten GDP das heutige Trading-Signal berechnet. Das Ganze wird durch den Anwender über ein Hauptskript gesteuert, in welchem er die Grundeinstellungen definieren und ausführen kann.

Um herauszufinden wie gut der Indikator ist, wird er mit verschiedenen Einstellungen getestet, die Ergebnisse beschrieben und illustriert. Hauptsächlich soll der Unterschied zwischen den zwei verwendeten Filtern, gleitender Durchschnitt und dem Direct Filter Approach (DFA), aufgezeigt werden.

## 2 Datensätze

Alle in dieser Arbeit verwendeten Daten wurden von dem Datenportal Quandl heruntergeladen. Welche Daten das genau sind wird in diesem Kapitel erläutert.

### 2.1 Vorgegebene Daten

In der Aufgabenstellung wurden einige Daten vorgegeben, die für den Indikator verwendet werden sollen. Man kann erahnen, dass sich diese während einer Rezession auffällig verhalten und somit die Rezessionen der letzten Jahre ersichtlich machen. Darum machen sie als erste Wahl Sinn.

In der folgenden Tabelle sind die Daten kurz beschrieben und die wichtigsten Informationen aufgelistet:

Name	Quandl-Code	Beschreibung	Frequenz	Start
Gross domestic Product	FRED/ GDP	Bruttoinlandsprodukt der USA	quarterly	01.01.1947
S&P 500	YAHOO/ INDEX_GSPC	Alle Aktien der 500 grössten börsennotierten US-Amerikanischen Unternehmen	daily	03.01.1950
VIX S&P 500	YAHOO/ INDEX_VIX	Die Volatilität des S&P 500	daily	02.01.1990
Initial Claims	FRED/ ICSA	Zahl der Erstanträge für Arbeitslosengeld	weekly	07.01.1967
Industrial Production Index	FRED/ INDPRO	Industrieproduktion der USA	monthly	01.01.1919
Non farm payroll	FRED/ PAYEMS	Alle Angestellten in den USA ohne Farm- und non-Profit Unternehmen	monthly	01.01.1939
Civilian Unemployment Rate	FRED/ UNRATE	Zahl aller Arbeitslosen in den USA	monthly	01.01.1948

Tabelle 1 Vorgegebene Daten aus der Aufgabenstellung

### 2.2 Zusätzliche Daten

In einem zweiten Schritt wurde im Internet nach zusätzlichen Daten gesucht, die ebenfalls für den Indikator verwendet werden könnten. Dabei ist man auf den „Personal Consumption Expenditures“-Index gestossen, welcher die privaten Konsumentenausgaben widerspiegelt. Dieser ist ein wichtiger Teil in der Berechnung des offiziellen GDP, folglich könnte er eine gute erklärende Variable für das Regressionsmodell sein.

Als weitere Variable ist man auf den „Manufacturer's New Orders: Durable Goods“-Index aufmerksam geworden, welcher die neuen Auftragseingänge von langlebigen Wirtschaftsgütern aufführt. Steigt der Index, erhöht sich die Beschäftigung von Arbeitern und die Produktionsraten. Dies kurbelt die



Wirtschaft an und beeinflusst das GDP positiv. Damit könnte er ein guter Frühzeitindikator sein und wird als gute erklärende Variable für das Regressionsmodell angenommen.

In der folgenden Tabelle sind diese beiden zusätzlich verwendeten Daten nochmals kurz beschrieben und ihre wichtigsten Informationen aufgelistet:

Name	Quandl-Code	Beschreibung	Frequenz	Start
Personal Consumption Expenditures	FRED/ PCE	Private Konsumausgaben der USA	monthly	01.01.1959
Manufactures New Orders: Durable Goods	FRED/ DGORDER	Neue Auftragseingänge für langlebige Wirtschaftsgüter	monthly	01.02.1992

Tabelle 2 Selbst recherchierte zusätzliche Daten aus dem Internet

Am besten wären Daten mit einer täglichen Frequenz, da dann an jedem Wochentag ein neuer Wert vorhanden wäre und eine Änderung früh erkannt werden könnte. Leider wurden keine solchen Daten auf Quandl gefunden oder sie werden nur für zahlende Kunden zur Verfügung gestellt.

## 3 Indikator

Der Indikator wurde so gebaut, dass er selbstständig die Daten einliest, filtert, eine Regression durchführt und am Ende zusätzlich ein Trading-Signal herausgibt. Zum besseren Verständnis werden die einzelnen Hauptfunktionen genauer erläutert.

### 3.1 Main

Für den Indikator bildet die R-Datei *Main* die Benutzerseite (Steuerungselement), welches die verschiedenen Hauptfunktionen aufruft und ihre Resultate in eine neue Variable abspeichert.

#### 3.1.1 Beschreibung

Zuerst werden einige Voreinstellungen getroffen, damit der Indikator ohne Störungen funktioniert. Er lädt vorgängig die folgenden Packages:

- *devtools*: Wichtige Unterstütlungslibrary
- *quandl/R-package*: Nötig um Daten von der Internetseite Quandl zu laden
- *lubridate*: Möglichkeit bei einem Datum mit Monaten zu rechnen
- *xts*: Ermöglicht das Arbeiten mit xts (**E**xtensible **T**ime **S**eries) Dateien.

Zusätzlich wird ein Pfad definiert und der Quandl-Code geladen. Dieser ist notwendig, um unbegrenzt Datenvolumen von Quandl zu laden. Anschliessend werden noch alle benötigten Hauptfunktionen geladen.

Als nächstes folgen die Einstellungen, welche der Anwender ändern kann. Bei der Variable *asset* können die zu benutzenden Zeitreihen eingegeben werden. Es ist wichtig, dass die Reihe *FRED/GDP* an der ersten Stelle erscheint, da sie die Zielvariable ist.

Der *lag* beträgt für Tages- und Wochendaten 0 und ist daher für diese uninteressant, jedoch für Daten mit einem grösseren Frequenz (z. Bsp. Monatsdaten) von Bedeutung. Normalerweise werden Daten, welche den gesamten Monat erfassen, nicht am letzten sondern am ersten Tag dieses Monats abgebildet. Daher muss dort eine Verschiebung um einen Monat nach hinten (*lag=1*) erfolgen.

Da die Daten von verschiedenen Herausgebern (z. Bsp. *FRED* oder *YAHOO*) stammen, besitzen sie verschiedene Darstellungen. Deshalb muss bei der Variable *select* eingegeben werden, welche Spalte für den Indikator verwendet werden soll (z. Bsp. Aktien: Spalte mit den Schlusskursen).

Die eingegebenen Einstellungen werden zur besseren Übersicht in der Matrix *asset\_mat* zusammengestellt, welche vom Anwender kontrolliert werden sollte.

Nun muss noch das Datum für die folgenden Variablen definiert werden:

- *last\_Rez*: Ende der letzten Rezession (Bsp. 2009-06-15)
- *start*: Startzeit der Zeitreihen
- *end*: Endzeit der Zeitreihen (Für Echtzeitanwendung: *format(Sys.Date())*)

Bei der Variable *filt* kann man entscheiden, welcher Filter (*dfa\_ms* oder *filt\_gleit\_D*) angewendet werden soll.

Im letzten Teil der Benutzerseite folgen die Aufrufe der einzelnen Hauptfunktionen. Am Ende erscheint im R-Output die vom Indikator empfohlene Strategie („cash“ oder „long“), welche heute angewendet werden sollte. Zusätzlich können die gefilterten Daten sowie das gefittete GDP mit der Signalgrenze betrachtet werden.

### 3.1.2 Probleme

Bei der Programmierung traten keine Probleme auf. Die *Main*-Datei wurde jedoch mit fortschreitender Zeit erweitert und neue Variablen hinzugefügt, um neue Einflüsse zu berücksichtigen.

## 3.2 Read

Die Hauptfunktion *Read* ist für das Einlesen der verschiedenen Zeitreihen verantwortlich. Vom *main* wird die Matrix *asset\_mat*, sowie die zwei Variablen *start* und *end* übergeben.

### 3.2.1 Beschreibung

Zuerst wird ein leeres xts mit dem Namen *mydata* erstellt. Dieses enthält im Index jedes Datum zwischen den übergebenen Parametern *start* und *end* inklusive diesen Werten. In der folgenden for-Schleife werden die einzelnen Zeitreihen, mit dem oben erwähnten Zeitraum, von Quandl heruntergeladen und geprüft, ob ein Lag ausgeführt werden muss. Wenn dies der Fall ist, wird direkt nach dem herunterladen der Reihe, der Lag durchgeführt. Informationen diesbezüglich werden der übergebenen *asset\_mat* entnommen.

Nachher findet die Transformation für jede Reihe statt. Für diese wurden die Log>Returns gewählt, da dies in der Aufgabenstellung vorgegeben war.

Zum Schluss finden in der *Read*-Funktion noch einige kosmetische Anpassungen statt. Diese beinhalten die Änderung der Spaltenbezeichnung, Auffüllung der NA's mit den vorhergehenden Werten bei den nicht transformierten Daten und Rückgabe der Resultate.

Die Resultate werden in der *Main*-Datei als Liste in der Variable *data* gespeichert. Mit den folgenden Befehlen können diese aufgerufen werden:

- *data\$mydata*: Eingelesene Zeitreihen (NA's aufgefüllt)
- *data\$mydata\_log*: Transformierte Zeitreihen (NA's nicht aufgefüllt)

### 3.2.2 Probleme

Das Hauptproblem war das Einlesen der Zeitreihen als xts. Zuerst wurden die einzelnen Werte nicht an dem richtigen Zeitpunkt aufgeführt. Dieses Problem konnte jedoch durch die vorgängige Erstellung eines NA Vektors, mit jedem Datum im Index und anschliessendem *cbind* behoben werden. Dadurch wurde zusätzlich das Problem beseitigt, bei welchem es zwingend nötig war, auf die Reihenfolge der einzulesenden Zeitreihen zu achten. Bei diesem konnte das GDP nicht als erstes eingelesen werden weil es Quartalsdaten sind. Dadurch traten beim *cbind* Probleme mit dem Index auf. Mit der Lösung konnten einige Zeilen R-Code eingespart werden.

## 3.3 Filter

Für den Indikator dient diese Hauptfunktion als Filter für die eingelesenen Daten. Vom *Main* wird das xts *mydata\_log* und die Variable *filt* (*dfa\_ms* oder *filt\_gleit\_D*) übergeben.

### 3.3.1 Beschreibung

Zu Beginn werden verschiedene Unterfunktionen geladen, welche die Hauptfunktion *Filter* benutzt. Da die Zielvariable (GDP) quartalsweise vorhanden ist, wird diese linear Interpoliert um einen fließenden Übergang zu erhalten. Das findet statt, da später über drei Monate gefiltert wird. Bei Quartalsdaten hätte dies keinen Effekt. Die lineare Interpolation geschieht mit der Unterfunktion *lin\_interpol*. Mit den restlichen Zeitreihen wird, je nach Eingabe in der Variable *filt*, der gleitende Durchschnitt oder der Direct Filter Approach (DFA) angewendet. Beim gleitenden Durchschnitt (*filt\_gleit\_D*) werden die Zeitreihen exakt über die letzten drei Monate gemittelt. Am Ende findet eine Auffüllung der NA's mit dem jeweiligen vorhergehenden Werten statt. Anders sieht es beim Direct Filter Approach (*dfa\_ms*) aus. Dort werden die NA's der Zeitreihen zu Beginn aufgefüllt, da der DFA für jeden Tag verschiedene Filtergewichte haben kann. Danach folgt für alle erklärenden Zeitreihen, die Berechnung des *Periodogramms*. Da in Echtzeit gefiltert werden soll ist der *Lag* = 0. Die Variable *L*, die Länge des Filters, wurde auf 90 gesetzt. Dies entspricht 90 Tagen, also ca. 3 Monaten. Grund dafür ist, dass man wie beim gleitenden Durchschnitt über drei Monate filtern möchte. Der *cutoff* von  $\pi/45$  ergibt sich, da man alle Frequenzen, die grösser als 90 Tage sind, filtern möchte. Mit diesem *cutoff* wird das *Gamma* erstellt. Anschliessend

kann mit dem *Periodogramm*, dem *Lag*, dem *Gamma* und der Filterlänge  $L$  die Filtergewichte  $b$  des DFA berechnet werden. Diese werden mit den entsprechenden Werten der erklärenden Zeitreihen multipliziert, womit man die gefilterten Zeitreihen erhält. Am Ende erfolgt die Rückgabe der gefilterten Daten.

Die gefilterten Daten werden in der *Main*-Datei als xts in der Variable *mydata.gef* gespeichert.

### 3.3.2 Probleme

In der Hauptfunktion *Filter* traten keine Probleme auf, jedoch in der Unterfunktion *filt\_gleit\_D*. Bei dieser konnte der Index nicht um drei Monate erhöht werden und somit war zuerst ein Filtern von exakt drei Monaten nicht möglich. Dieses Problem konnte dann aber mit der Entdeckung der library *lubridate* behoben werden.

## 3.4 Regression

Bei der Hauptfunktion *Regression* wird mit den vorhandenen Zeitreihen eine Regression durchgeführt. Vom *Main* wird das xts *mydata.gef* übergeben.

### 3.4.1 Beschreibung

Am Anfang wird die Unterfunktion *correlation* geladen. Mit dieser werden die besten Korrelationen zwischen der ersten (Zielvariable) und der jeweiligen folgenden Reihen (erklärende Variable) ermittelt. Dies geschieht durch die Verschiebung in die Vergangenheit. Der Gedanke dazu ist, dass die einzelnen erklärenden Variablen vielleicht früher nach unten streben als die Zielvariable. Es darf nur in die Vergangenheit verschoben werden, da man die Zukunft nicht kennt. Es wurde ein Maximum von 50 Zeitschritten gewählt, da eine zu weite Verschiebung keinen Sinn macht. Nachdem die besten Korrelationen gespeichert wurden, werden die erklärenden Variablen um die ermittelten Schritte verschoben und abgespeichert. Da die jüngsten Daten des GDP noch revidiert werden müssen, geht man davon aus, dass die Daten welche ca. ein Jahr zurückliegen in etwa korrekt sind. Darum wird mit allen Zeitreihen ein Regressionsmodell gebildet, wobei die Daten der letzten 12 Monate weggelassen werden. Mit dem Befehl *step (direction="backward")* findet bei diesem eine Rückwärtsselektion der Variablen mit dem AIC-Kriterium statt. Dabei werden die Variablen aus dem Modell eliminiert, welche die Regression nicht ausreichend beschreiben. Nun findet mit den geschätzten Parametern und den Daten aus den letzten 12 Monaten eine Schätzung (Prognose) für das GDP statt. Als Abschluss werden die gefitteten- und die geschätzten Daten als zwei verschiedene xts-Dateien abgespeichert und zurückgegeben.

In der *Main*-Datei wird das Resultat als Liste in der Variable *reg* gespeichert.

- *reg\$dat.gefittet*: gefittete Zielvariable bis vor einem Jahr (GDP)
- *reg\$prog*: Prognose der Zielvariable bis Heute (GDP)

### 3.4.2 Probleme

Das grösste Problem trat im Zusammenhang mit dem *xts* auf. Die gefitteten Werte wurden nicht mehr in dieser Form zurückgegeben. Um dies zu beheben, musste der Index vorgängig herausgenommen werden, um in nach dem Fitten wieder anzufügen. Ausserdem gab es Probleme bei der Definierung des Ausgangsmodells bei der Regression. Dies konnte dann aber mit dem *past*-Befehl gelöst werden.

## 3.5 Signal

Dieser Teil gibt für den Indikator das Signal („long“ / „cash“). Vom *Main* wird das *xts reg* sowie die Variablen *last\_Rez* und *filt* übergeben.

### 3.5.1 Beschreibung

Zuerst wird der Index mit jedem Datum abgespeichert und zusätzlich das heutige angehängt. Nachher folgt das Zusammenfügen der gefitteten und den prognostizierten Werten des GDP. Zusätzlich wird der Index wieder angefügt. Damit die letzte Rezession das Signal nicht verfälscht, werden die Werte bis zu diesem Datum gelöscht. Ausserdem wird der letzte Eintrag entfernt, da das Signal aufgrund des Wertes von gestern berechnet wird. Bei den Variablen *len* und *versch* erfolgt die Eingabe über wie viele Jahre zurück gerechnet und wie stark die Signalgrenze verschoben werden soll. Diese Werte ergaben sich aus mehreren Versuchen. Sie sind je nach Eingabe des Filters unterschiedlich. In den nächsten Zeilen wird für das Signal der Mittelwert der Zielvariablen (GDP) über die angegebenen Jahre berechnet und verschoben. Wenn nun der aktuelle Wert des geschätzten GDP über oder auf der Signalgrenze liegt, wird in dem Vektor *entscheid* „long“ gespeichert, ansonsten „cash“. Das Signal wird für jeden Tag neu berechnet. Zum Schluss werden die Resultate zurückgegeben.

Die Resultate werden in der *Main*-Datei als Liste in der Variable *sig* gespeichert.

- *sig\$rez\_sig*: Wert des Signals für jeden Tag
- *sig\$entscheid*: Entscheid „long“ oder „cash“ für jeden Tag

### 3.5.2 Problem

Beim Programmieren dieser Hauptfunktion traten keine grösseren Probleme auf. Die einzige Schwierigkeit war das Herausfinden der optimalen Werte für die Variablen *len* und *versch*. Dies konnte jedoch mit einigen Simulationen gelöst werden.

## 4 Performance

Um zu sehen, wie gut die Signale des Indikators sind, wurde er mit verschiedenen Einstellungen getestet. Die Signale wurden auf den S&P 500 angewendet. Die verschiedenen Tests und Ergebnisse werden in diesem Kapitel beschrieben und illustriert. Dafür wurde zusätzlich ein R-Code erstellt.

### 4.1 R-Code

Es wurde eine Datei mit dem Namen *Test* erstellt, welche die Funktion *Performance* benötigt. Damit kann getestet werden, wie der Indikator bei einer Anwendung in Echtzeit abgeschlossen hätte.

#### 4.1.1 Test

Die *Test*-Datei bildet die Testumgebung und simuliert die Performance des Indikators. Je nach Einstellungen können genauere Ergebnisse berechnet werden, jedoch wird dadurch der Rechenaufwand grösser.

##### 4.1.1.1 Beschreibung

Ganz am Anfang wird die Funktion *Performance* geladen. Danach kann selbst entschieden werden, ab welchem Tag man in den S&P 500 investiert hätte (*start\_invest*). Bei der Variable *abstand* (Prognoseintervall) kann man entscheiden, wie weit in die Zukunft die Daten bekannt sind. Um eine Echtzeit Berechnung der Performance zu erhalten, sollte hier *days* definiert werden. Das Problem ist jedoch, dass bei dieser Einstellung sehr lange gerechnet werden muss. Daher bietet sich *months* an, bei der sich der Rechenaufwand in Grenzen hält und die Performance sich nicht gross unterscheidet, weil viele erklärende Variablen Monatsdaten sind. Die gefilterten Daten muss man aus der *Main*-Datei laden. Mit diesen Einstellungen wird nun die Funktion *Performance* aufgerufen. Nach den Berechnungen werden die Trading-Signale geplottet. Ausserdem werden die Differenzen zwischen dem S&P 500 und des Indikators berechnet und grafisch dargestellt, um den Vorteil gegenüber dem S&P 500 zu sehen. Zusätzlich können in der *Test*-Datei einige Werte berechnet werden um quantitative Vergleiche durchführen zu können.

##### 4.1.1.2 Probleme

Es traten keine erwähnenswerten Probleme auf.

### 4.1.2 Performance

Für die Testumgebung dient diese Funktion für die Berechnung der Performance. Vom *Test* werden die gefilterten Daten *mydata.gef*, die Liste *data* und die Variablen *start\_invest* und *abstand* übergeben.

#### 4.1.2.1 Beschreibung

Zu Beginn werden die Hauptfunktionen geladen, welche die Funktion Performance benutzt. Danach wird in der Variablen *ind*, jedes Datum von *start\_invest* bis und mit des letzten Datums in dem übergebenen xts *mydata.gef* abgespeichert. Der Intervall wird aus der Variable *abstand* gelesen. Zusätzlich wird ein Index gespeichert, bei welchem alle Tage des vorher erwähnten Zeitraums vorhanden sind. Als nächstes muss ein Default-Wert (*akt\_Rez*) definiert werden, weil man die letzte Rezession für die Performance nicht berücksichtigt. Daher wird das erste Datum des Zeitraums um 50 Tage aufsummiert. Dies ist nötig weil in der Hauptfunktion eine Zeitverschiebung von maximal 50 Zeitschritten erfolgt. Die Performance wird dadurch aber nicht beeinflusst. In der folgenden for-Schleife, welche über die gesamte Länge der Variable *ind* geht, werden bis zu jedem gespeicherten Datum in *ind* die Daten aus der xts-Datei *mydata.gef* herausgegriffen. Mit diesen wird die Hauptfunktion *Regression* ausgeführt. Das Resultat und die Variable *akt\_Rez* werden an die Hauptfunktion *Signal* übergeben. Das Ergebnis wird in der Variable *ent* abgespeichert, bei welcher jedes neue Ergebnis aus der for-Schleife angehängt wird. Natürlich nur immer die noch nicht vorhandenen Tage. In der neuen Variable *invest* wird die Zeitreihe des nicht transformierten S&P 500 mit dem gleichen Index wie die Variable *ent* gespeichert. Als letztes wird die Variable *gew* erzeugt, welche den Gewinn simulieren soll, der erzielt worden wäre, wenn man in die Strategie des Indikators investiert hätte. Der Gewinn wird mit einer for-Schleife berechnet und abgespeichert.

Die Resultate werden in der *Test*-Datei als Liste unter dem Namen *per* gespeichert. Mit den folgenden Befehlen können die Resultate aufgerufen und zur besseren Darstellung geplottet werden.

- *per\$gew*: Trading-Signal des Indikators
- *per\$gew*: Gewinn bei Anwendung des Indikators

#### 4.1.2.2 Probleme

Zuerst wurde versucht, dass die letzten Rezessionen berücksichtigt werden. Dies konnte aber leider nicht realisiert werden, da es zu kompliziert wurde. Deshalb entschied man sich dieses Kriterium wegzulassen. Das Nichtberücksichtigen sollte keine grossen Unterschiede ergeben, da die Testzeit nur zwei Rezessionen beinhaltet. Ausserdem würde das Resultat besser und nicht schlechter werden.



## 4.2 Filter: Gleitender Durchschnitt

In einem ersten Schritt wurde der Indikator mit den 6 Datensätzen (S&P 500, VIX S&P 500, Initial Claims, Industrial Production Index, Non Farm Payroll, Civilian Unemployment Rate) aus der Aufgabenstellung getestet. Anschliessend noch zusätzlich mit den zwei zusätzlichen Datensätzen (Personal Consumption Expenditures und Manufacturer's New Orders: Durable Goods).

### 4.2.1 Indikator mit vorgegebenen Daten

Für die Signalgrenze wurde  $len=1.5$  und  $versch=0.002$  gesetzt. Diese Werte sind beim mehrmaligen Testen als optimal empfunden worden. Der Indikator hat die Möglichkeit die erklärenden Zeitreihen in die Vergangenheit zu verschieben, um die beste Korrelation zu finden. Damit man herausfinden kann, ob diese Möglichkeit das Resultat verbessert, wurden zwei Varianten getestet:

1. keine Verschiebung
2. max. Verschiebung um 50 Tage

Auf diese zwei Varianten wird in diesem Kapitel eingegangen.

### GDP mit Signalgrenze

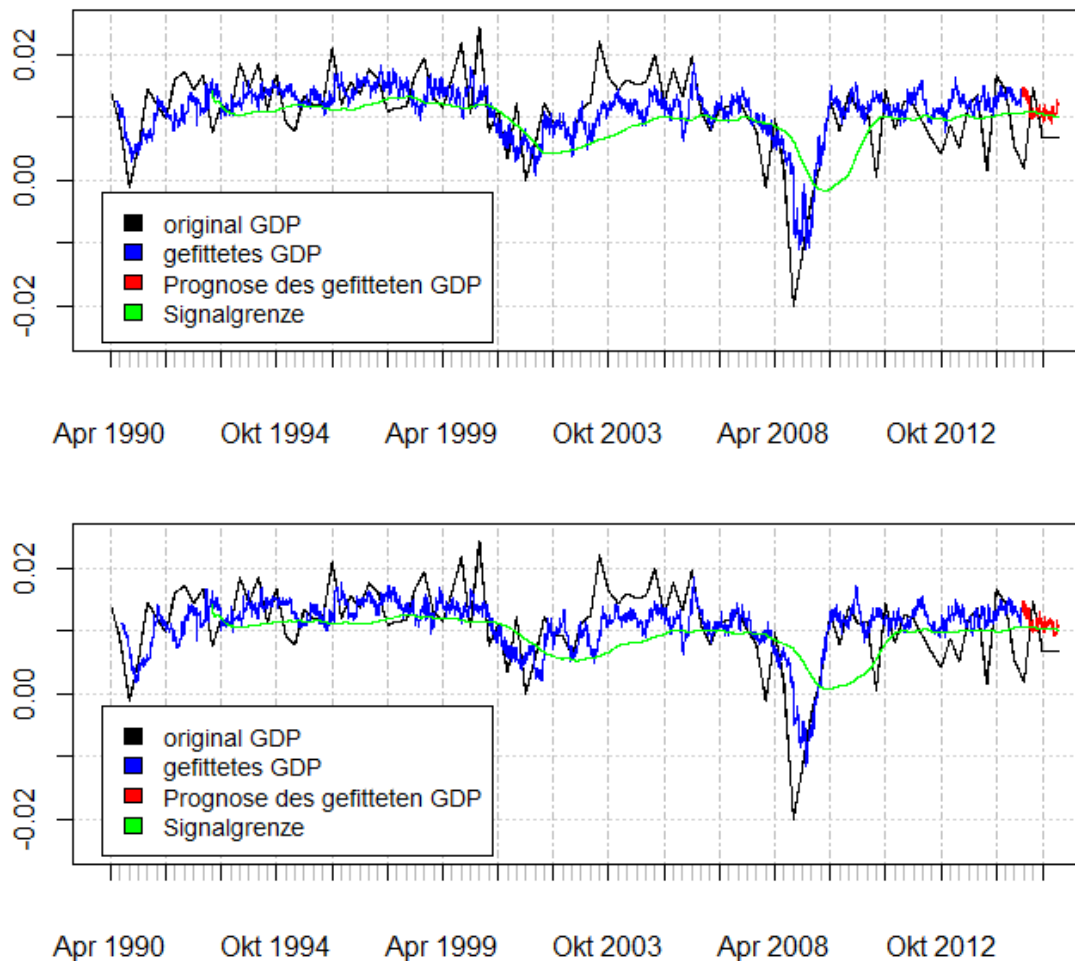


Abbildung 1    Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten  
Unten: Indikator mit gl. Durchsch., mit Verschiebung, mit vorgegebenen Daten

Es muss gesagt werden, dass die zwei Plots für die Signalgrenze nur näherungsweise stimmen, da für die vergangenen Signale das geschätzte GDP bis 09.12.2015 bekannt war. Folglich sieht das Resultat besser aus als es in Echtzeit sein wird. Trotzdem kann man sehr gut sehen, dass der Indikator bei beiden Varianten die Regressionen im Jahr 2001 und 2008 erkennt. Somit hätte er bei beiden ein „cash“-Signal gegeben, weil in diesem Zeitraum die Signalgrenze oberhalb des gefitteten GDP ist. Bei der zweiten Variante sieht die Näherung besser aus. Jedoch ist es schwierig, dies zu beurteilen, da das GDP nur quartalsweise vorhanden ist. Folglich könnte es zwischen den Quartalswerten Schwankungen haben. Um zu sehen wie die beiden Varianten bezüglich dem dauerhaften investieren in den S&P 500 abgeschnitten hätten,

wurden die Signale von dem Indikator auf den S&P 500 angewendet. Dies konnte man mit der *Test*-Datei durchführen. Es wurden die Daten vom 01.01.1990 bis 09.12.2015 eingelesen und gefiltert. Im *Test* wurde ein Prognoseintervall von einem Jahr und das Datum vom Investitionsstart auf den 01.01.1997 gesetzt. Dieser erfolgte nicht früher, weil der Indikator einen gewissen Vorlauf an Daten braucht, um eine gute Regression generieren zu können.

### Trading-Signal

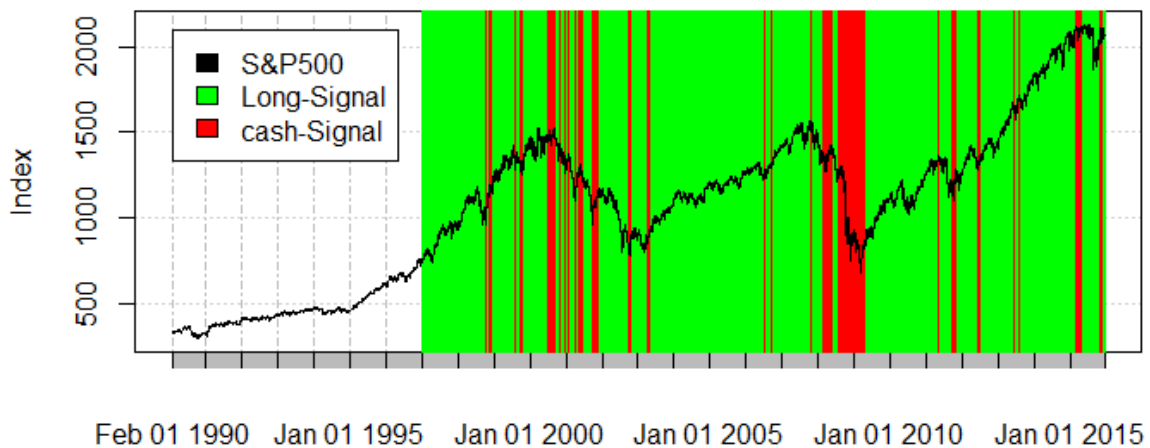
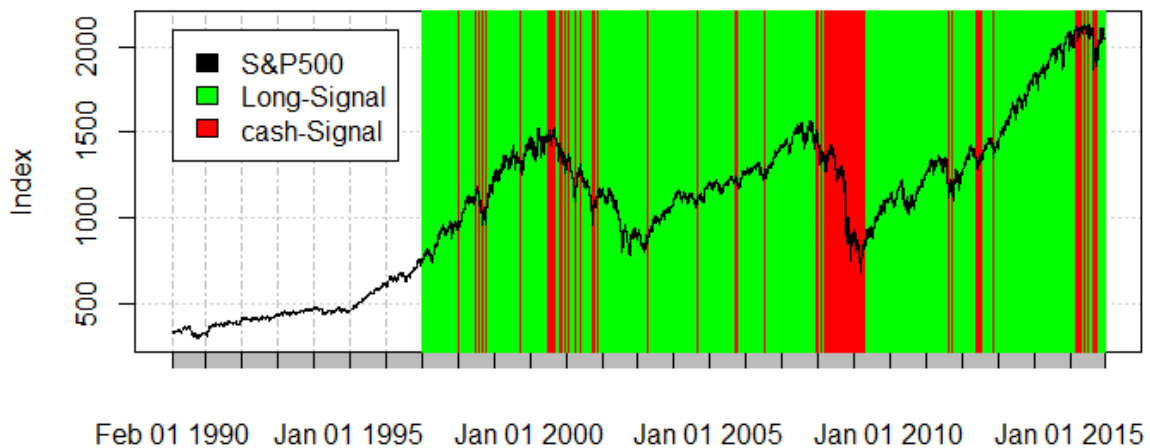


Abbildung 2    Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten  
Unten: Indikator mit gl. Durchsch., mit Verschiebung, mit vorgegebenen Daten

Man sieht, dass der Indikator bei beiden Varianten die Rezessionen erkennt. Er gibt in diesen Perioden mehr „cash“-Signale, als in den Anstiegen. Jedoch scheint es, dass der Indikator in der ersten Rezession Mühe hat. Das Problem ist, dass der Verlauf des S&P 500 langsam nach unten zeigt und sich immer wieder leicht erholt. In der zweiten Rezession ist das Gegenteil der Fall. Dort zeigt

der Verlauf sehr schnell und ohne Pause nach unten. Der grösste Unterschied zwischen den zwei Varianten erkennt man in der zweiten Rezession. Dort hat die zweite Variante eine kurze „long“-Phase.

Im folgenden Plot sieht man die Performance bezüglich des S&P 500.

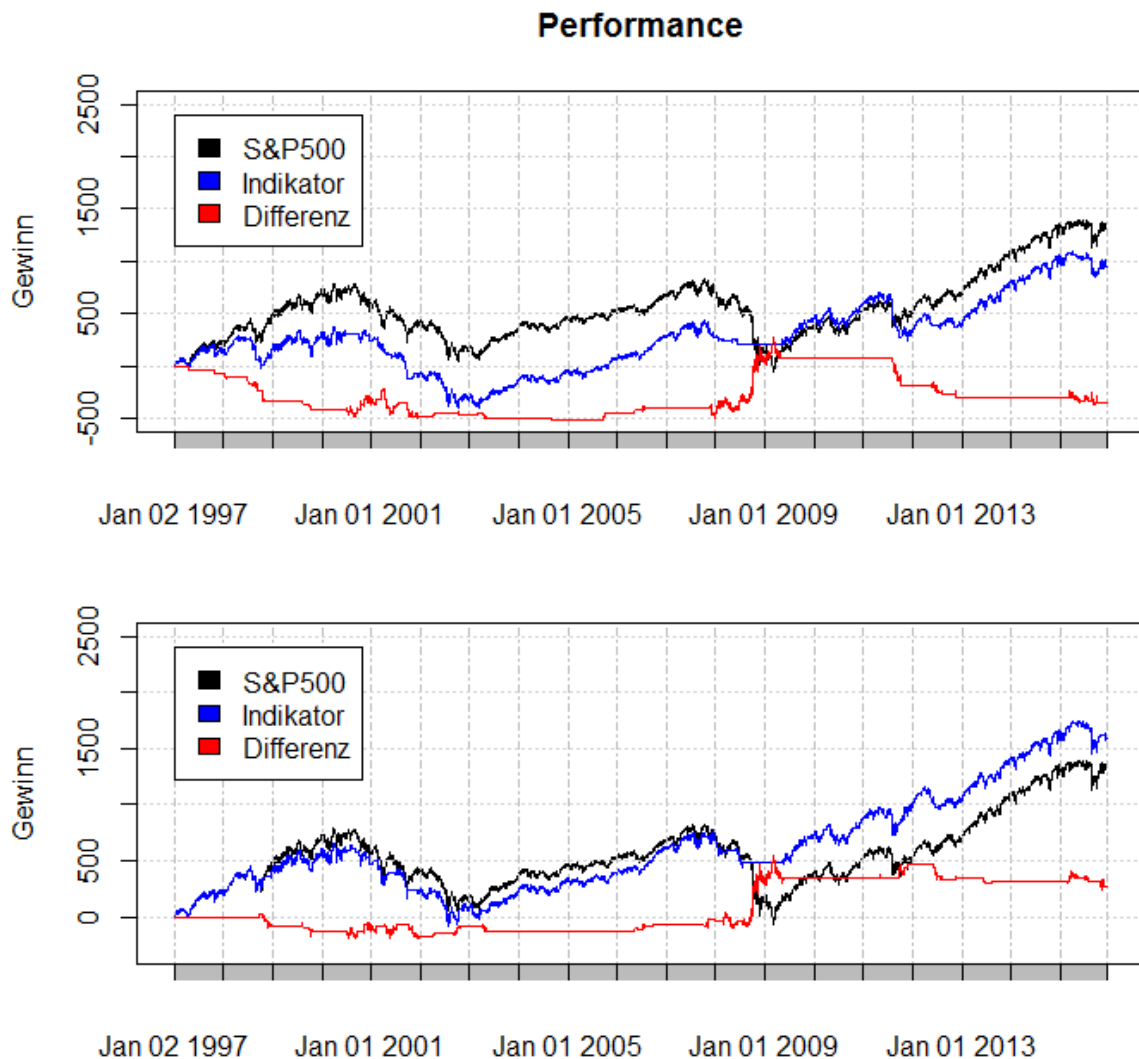


Abbildung 3    Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten  
Unten: Indikator mit gl. Durchsch., mit Verschiebung, mit vorgegebenen Daten

Es ist zu erwähnen, dass diese Performance nicht in Echtzeit durchgeführt wurde. Der Indikator hatte die Daten des geschätzten GDP bereits für ein Jahr voraus, da in der *Test-Datei* ein Prognoseintervall von einem Jahr gewählt wurde. Folglich sehen diese Plots besser aus als es dann später in Echtzeit sein wird. Trotzdem sind sie eine gute Approximation um die Performances zu vergleichen. Wenn man nur die zweite Rezession beachtet, schneidet die zweite Variante gegenüber der ersten besser ab. Der Grund dafür ist, dass die Regression mit

maximal 50 Zeitschritten zurück im Mittel besser ist, aber die Wendepunkte später erkannt werden. Folglich ist die zweite Variante bei der ersten Rezession besser, weil diese keine starken Wendepunkte hat, sondern sehr langsam fällt und sich immer wieder leicht erholt.

Im gesamten Vergleich ist die zweite Variante besser. Deshalb werden die Tests im folgenden Kapitel nur noch mit dieser durchgeführt.

#### 4.2.2 Indikator mit zusätzlichen Daten

Nun wird der Indikator mit den vorgegebenen und den zusätzlich Daten getestet. Da sich im letzten Kapitel der Indikator mit der Verschiebung als besser erwiesen hat, wird in diesem Kapitel nur diese Variante getestet. Für die Signalgrenze wurde  $len=1.5$  und  $versch=0.002$  gesetzt.

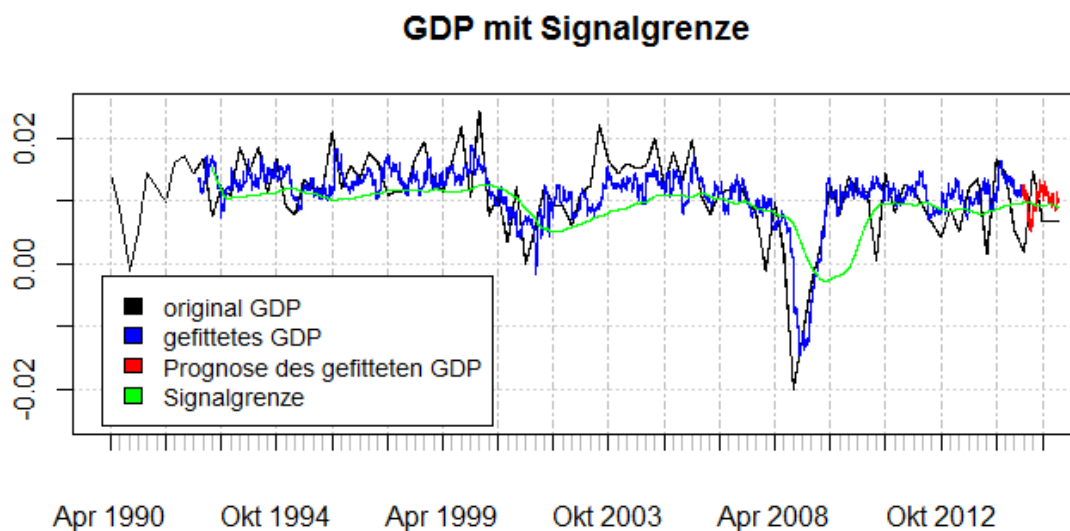


Abbildung 4 Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.

Im Plot mit dem GDP und der Signalgrenze sieht man, dass die Regression erst im Jahr 1992 beginnt. Dies kommt, weil die Daten einer der zusätzlichen Zeitreihen erst ab dem Jahr 1992 zur Verfügung stehen. Die Regression sieht nicht schlecht aus, sie ist sehr nahe am Original. Positiv ist, dass die Regression in den Rezessionen besser mitgeht, als bei den Varianten im vorhergehenden Kapitel.

### Trading-Signal

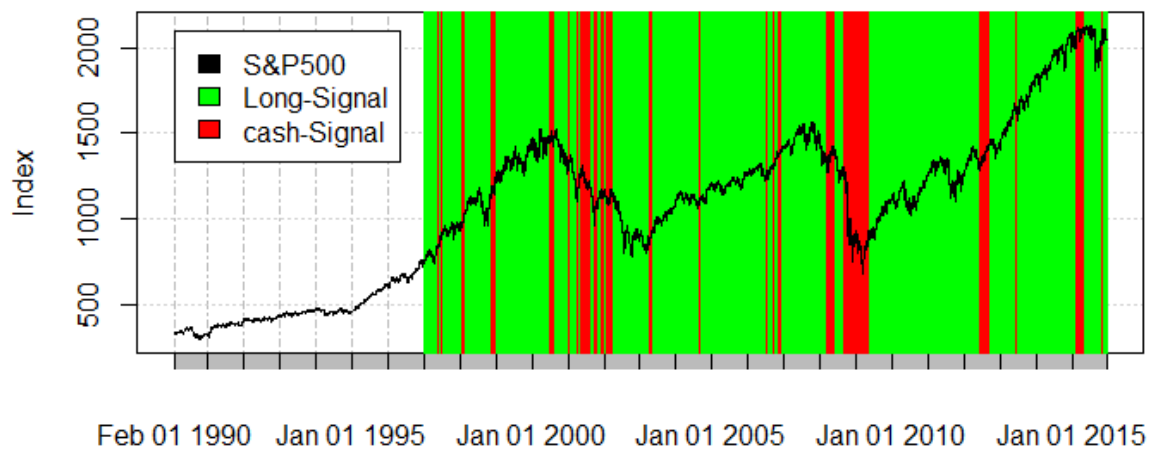


Abbildung 5 Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.

Man erkennt, dass die Trading-Signale nicht nur in den Rezessionen vorkommen sondern vermehrt auch in den Aufwärtstrends. Trotzdem sind die „cash“-Signale in den Rezessionen sehr stark vorhanden. Besonders in der zweiten Rezession kann ein hoher Verlust minimiert werden.

### Performance

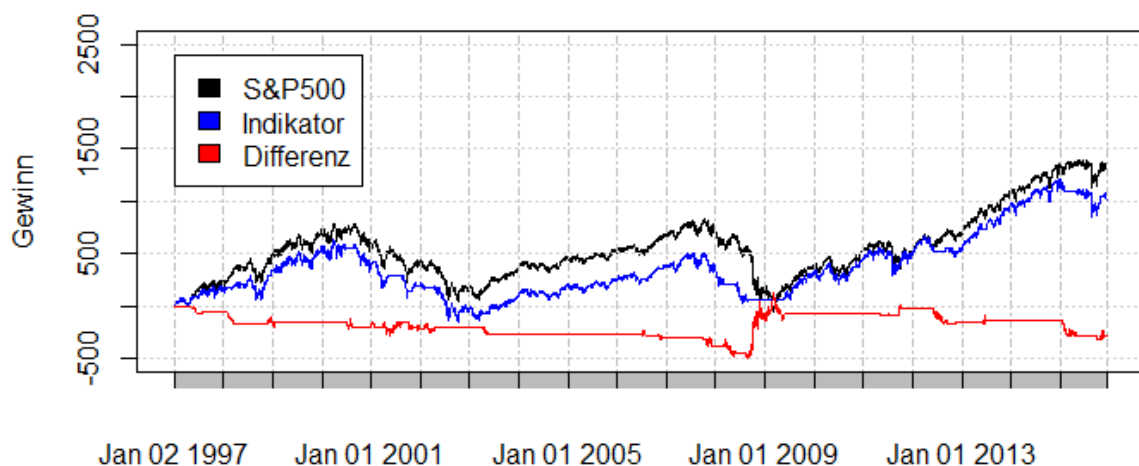


Abbildung 6 Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.

Die Performance sieht insgesamt eher schlecht aus. Jedoch hat der Indikator die zweite Rezession gut erkannt. Wie im vorhergehenden Kapitel, hat er auch in dieser Variante Mühe mit der ersten Rezession, der Grund ist derselbe.

#### 4.2.3 Fazit

Insgesamt schneidet der Indikator besser ab, wenn man nur die vorgegebenen Zeitreihen aus der Aufgabenstellung verwendet. Der Grund dafür muss an der Regression liegen, da nur dort ein Unterschied in der Berechnungen vom Indikator liegt. Man erkennt, dass die Regression mit den zusätzlichen Daten verrauschter ist. Dies liegt an den zusätzlich geschätzten Koeffizienten. Jedoch ist die Regression mit den zusätzlichen Daten eine bessere Näherung an das originale GDP. Trotzdem sieht man in Abbildung 3 und 6, eine Verschlechterung in der Performance.

### 4.3 Filter: Direct Filter Approach

In einem ersten Schritt wurde der Indikator mit den 6 Datensätzen (S&P 500, VIX S&P 500, Initial Claims, Industrial Production Index, Non Farm Payroll, Civilian Unemployment Rate) aus der Aufgabenstellung getestet. Anschliessend noch zusätzlich mit den zwei zusätzlichen Datensätze (Personal Consumption Expenditures und Manufacturer's New Orders: Durable Goods).

#### 4.3.1 Indikator mit vorgegebenen Daten

Für die Signalgrenze wurde  $len=1.5$  und  $versch=0.001$  gesetzt. Diese Werte sind beim mehrmaligen testen als optimal empfunden worden. Der Indikator hat die Möglichkeit die erklärenden Zeitreihen in die Vergangenheit zu verschieben, damit die beste Korrelation gefunden werden kann. Um herauszufinden ob diese Möglichkeit das Resultat verbessert, wurden zwei Varianten getestet:

1. keine Verschiebung
2. max. Verschiebung um 50 Tage

Auf diese zwei Varianten wird in diesem Kapitel eingegangen.

### GDP mit Signalgrenze

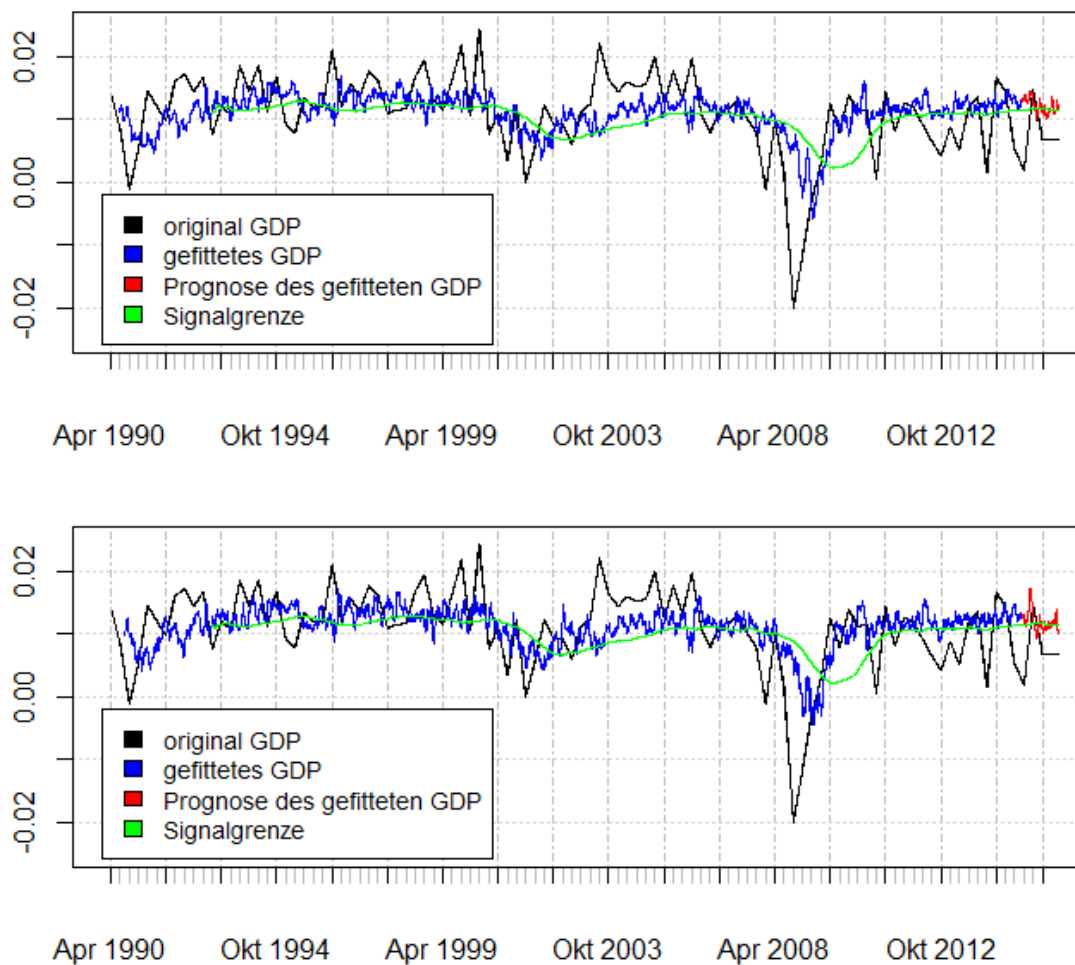


Abbildung 7    Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten.  
Unten: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten.

Es muss gesagt werden, dass die zwei Plots für die Signalgrenze nur näherungsweise stimmen, da für die vergangenen Signale das geschätzte GDP bis heute bekannt war. Folglich sehen die Resultate besser aus als sie in Echtzeit sind. Man kann sehr gut sehen, dass der Indikator bei beiden Varianten die Regressionen im Jahr 2001 und 2008 erkennt und viele „cash“-Signal gegeben hätte. Beunruhigend ist jedoch, dass in beiden Varianten die Regressionen fast zu spät auf die Rezessionen reagieren. Allgemein sind diese aber sehr ähnlich. Ein kleiner Unterschied ist, dass die zweite Variante ein wenig verrauschter zu sein scheint. Zusätzlich fällt auf, dass in der Prognose des gefitteten GDP ein kleiner Ausschlag nach oben ersichtlich. Um zu sehen wie die beiden Varianten bezüglich dem S&P 500 abgeschnitten hätten, wurden die Signale von dem



Indikator auf den S&P 500 angewendet. Dies konnte man mit der *Test*-Datei durchführen. Es wurden die Daten vom 01.01.1990 bis 09.12.2015 eingelesen und gefiltert. Im *Test* wurde ein Prognoseintervall von einem Jahr und das Datum vom Investitionsstart auf den 01.01.1997 gesetzt. Dieser erfolgte nicht früher, weil der Indikator einen gewissen Vorlauf an Daten braucht, um eine gute Regression generieren zu können.

### Trading-Signal

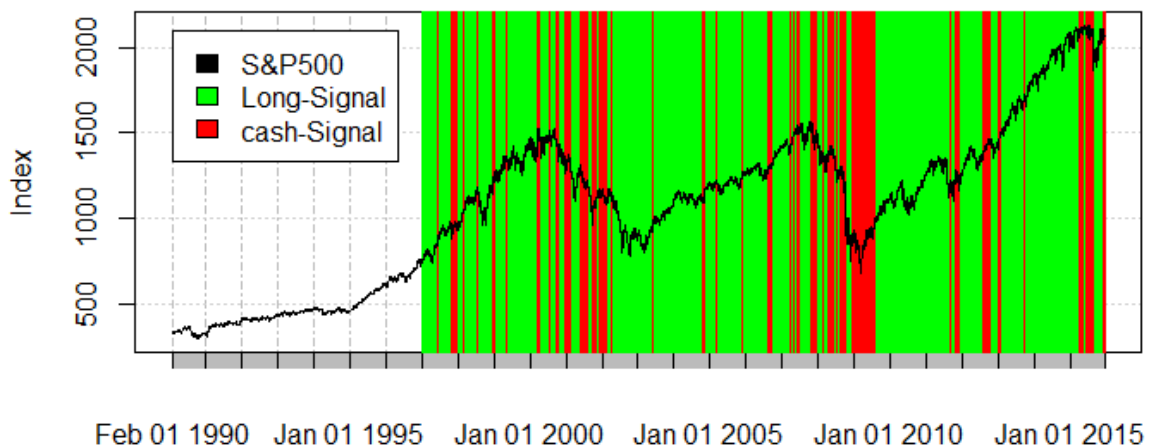
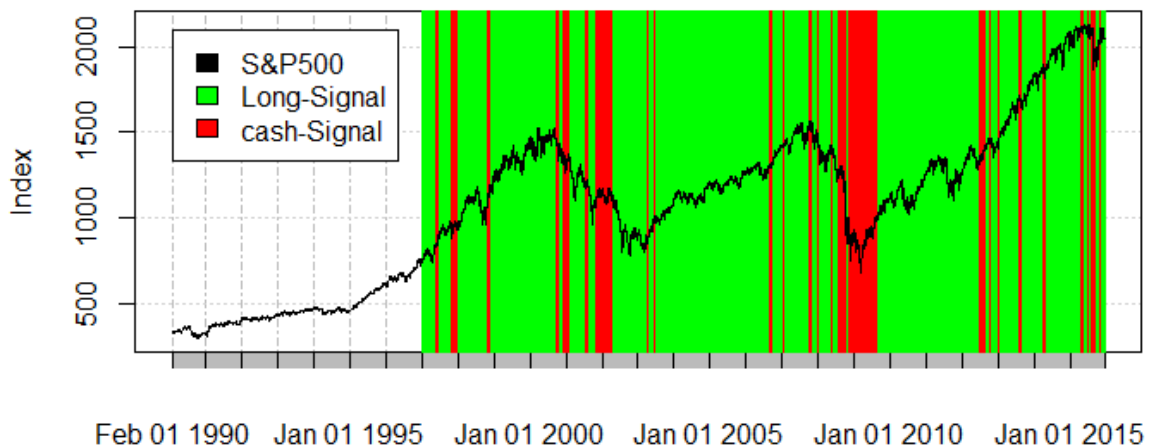


Abbildung 8    Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten.  
                  Unten: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten.

Man sieht, dass der Indikator bei beiden Varianten die Rezessionen erkennt. Er gibt in diesen Perioden mehr „cash“-Signale als in den Anstiegen. Jedoch scheint es, dass der Indikator in der ersten Rezession Mühe hat. Das Problem ist immer noch dasselbe wie beim Indikator mit dem gleitenden Durchschnitt. Der Verlauf des S&P 500 geht nur langsam nach unten und erholt sich immer wieder leicht.

In der zweiten Rezession ist das Gegenteil der Fall. Dort zeigt der Verlauf sehr schnell und ohne Pause nach unten. Weiter fällt auf, dass die zweite Variante viel mehr Signale erzeugt als die erste. Jedoch scheinen diese meistens korrekt zu sein und somit hat es fast nur bei Abwärtstrends „cash“-Signale. Interessant ist auch, dass der Indikator bei beiden Varianten am Anfang des Jahres 2015, während der China-Krise, sehr viele „cash“-Signale gibt. In der folgenden Abbildung sieht man die Performance bezüglich des S&P 500.

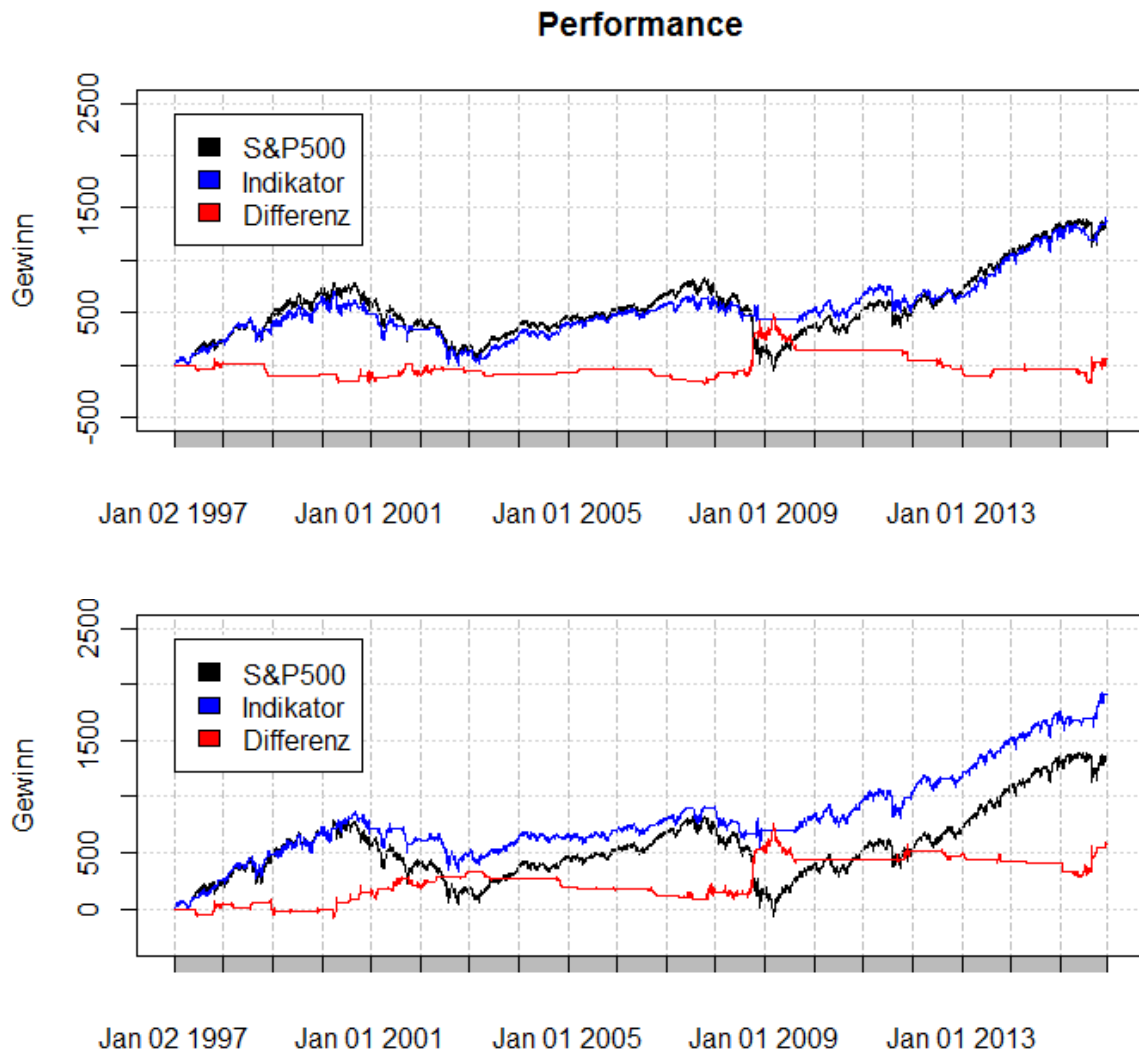


Abbildung 9    Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten.  
                 Unten: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten.

Es ist zu erwähnen, dass diese Performance nicht in Echtzeit durchgeführt wurde. Der Indikator hatte die Daten des geschätzten GDP bereits für ein Jahr voraus, da in der *Test-Datei* ein Prognoseintervall von einem Jahr gewählt wurde. Folglich sehen diese Plots besser aus als sie in Echtzeit wären. Trotzdem sind

sie eine gute Approximation um die Performance zu vergleichen. Bei beiden Rezessionen und der China-Krise schneidet die zweite Variante besser ab. Jedoch sieht man bei genauer Betrachtung, dass die erste Variante in dem Anstieg von 2003 bis 2007 besser ist.

Insgesamt schlägt die zweite Variante die erste klar. Deshalb werden die Tests im folgenden Kapitel nur noch mit der zweiten Variante durchgeführt.

#### 4.3.2 Indikator mit zusätzlichen Daten

Nun wird der Indikator mit den vorgegebenen und den zusätzlich Daten getestet. Da sich im letzten Kapitel der Indikator mit der Verschiebung, als besser erwiesen hat, wird in diesem Kapitel nur diese Variante getestet. Für die Signalgrenze wurde  $len=1.5$  und  $versch=0.001$  gesetzt.

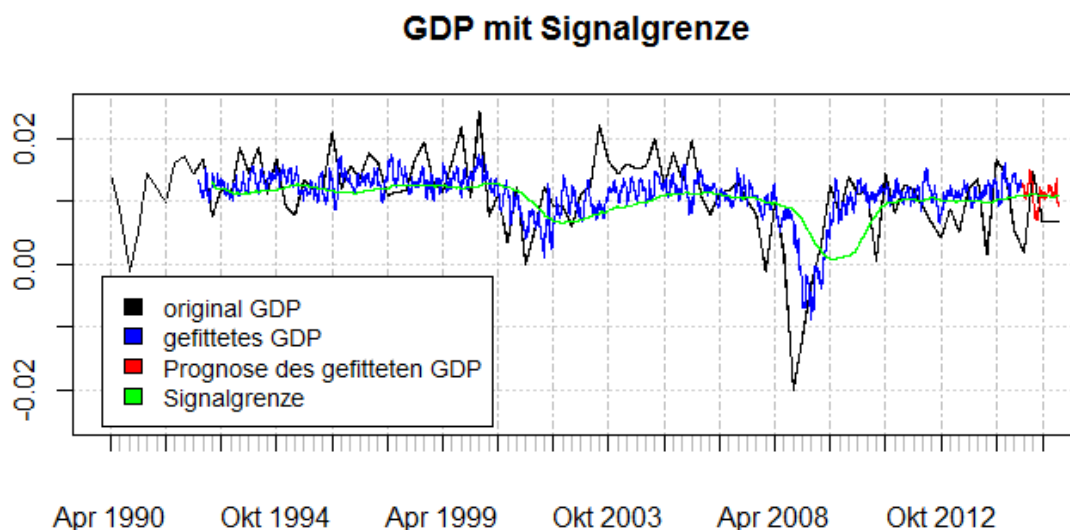


Abbildung 10 Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten

In der Abbildung mit dem GDP und der Signalgrenze sieht man, dass die Regression erst im Jahr 1992 beginnt. Dies kommt, weil die Daten einer der zusätzlichen Zeitreihen erst ab dem Jahr 1992 zur Verfügung stehen. Die Regression sieht allgemein nicht schlecht aus. Beunruhigend ist jedoch, dass man während der zweiten Rezession im Jahr 2008 eine Zeitverzögerung gegenüber dem originalen GDP erkennt.

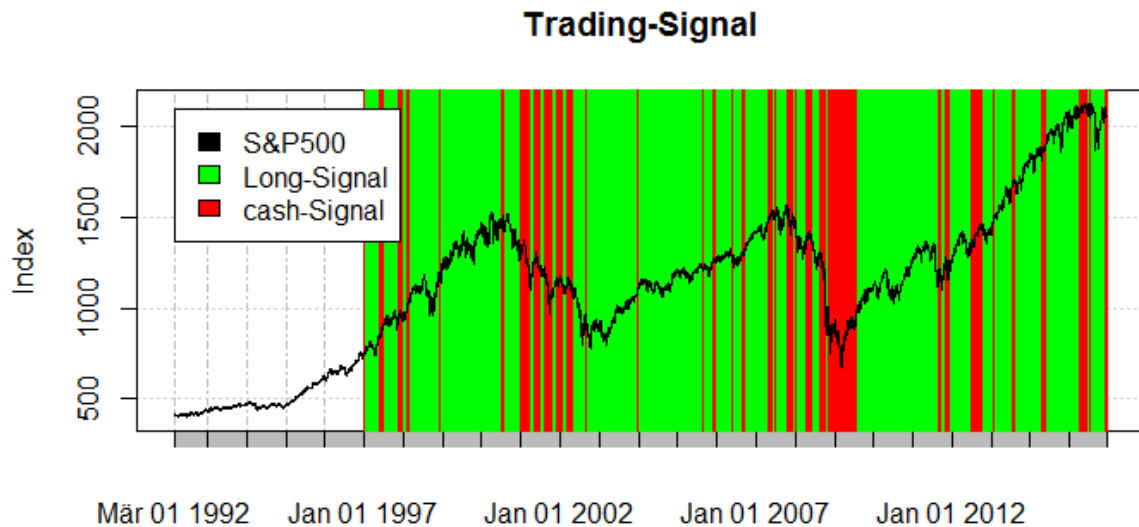


Abbildung 11 Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten

Obwohl es ziemlich viele „cash“-Signale hat, sind sie in den Rezessionen viel dominanter und daher geht man während den Verlustzone aus dem S&P 500. In den Anstiegen sieht man, dass der Indikator leider einige falsche Signale sendet und einen höheren Vorteil verhindert.

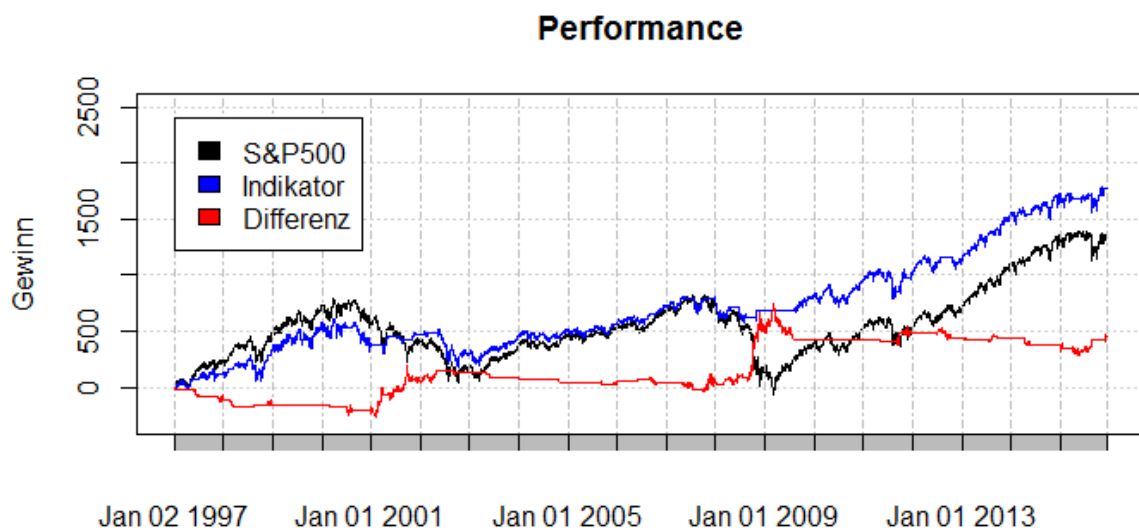


Abbildung 12 Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten

Bezüglich der Performance sieht es insgesamt nicht schlecht aus. Der Indikator scheint jedoch bei den Aufwärtstrends zwischen den Rezessionen Mühe zu

haben, dies wurde bereits in der Abbildung 11 vermutet. In den Rezessionen selber scheint er gute Signale zu geben, besonders bei der zweiten.

#### 4.3.3 Fazit

Die Regression scheint besser mit dem Original mit zu gehen, als wenn nur die vorgegebenen Daten verwendet werden. Schlecht ist allerdings, dass die Regression etwas zeitverzögert ist. Trotzdem schneidet der Indikator insgesamt besser ab, wenn man nur die vorgegebenen Zeitreihen verwendet. Man kann sehen, dass besonders im ersten Anstieg von 1997 bis 2000 und der China-Krise im Jahr 2015 eine bessere Performance erreicht wird. In den übrigen Bereichen schneiden sie ungefähr gleich gut ab.

#### 4.4 Vergleich: DFA vs. gleitender Durchschnitt

Dieses Kapitel vergleicht die besten Ergebnisse der zwei Filtervarianten miteinander. Um den Vergleich möglichst realitätsnah bewerten zu können, wurde der Indikator nicht mehr mit einem jährlichen Prognoseintervall getestet, sondern mit einem monatlichen. Es ist immer noch kein Echtzeittest, aber die Resultate sind näher an der Realität, da der Indikator nur noch die Regression für einen Monat voraus kennt. Vorher wurde dies vermieden, da der Indikator sehr rechnungsintensiv ist und viele verschiedene Varianten getestet wurden.

### Trading-Signal

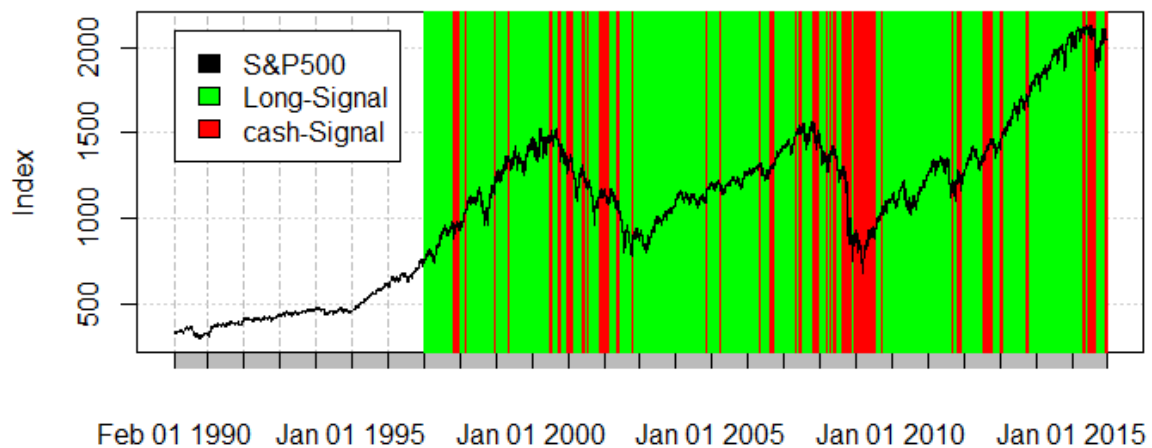
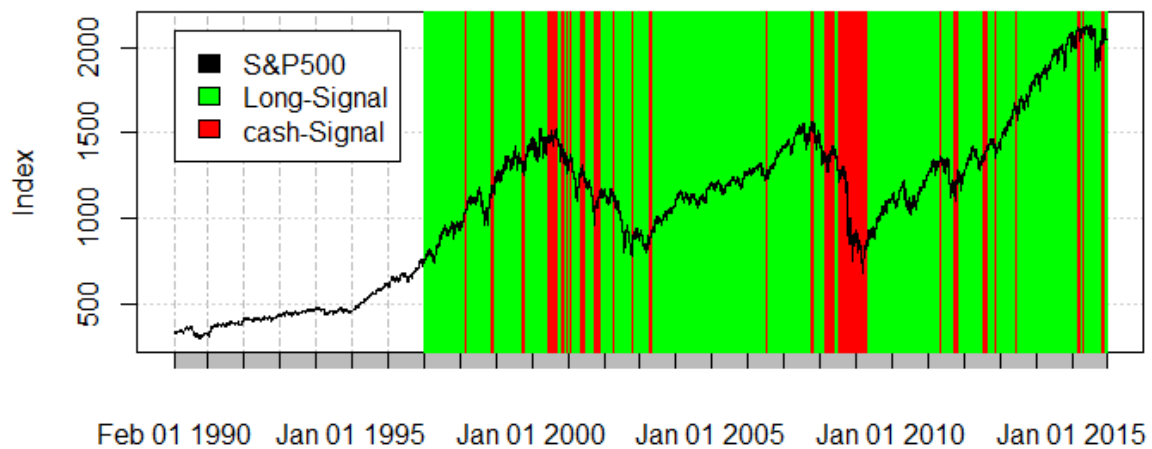


Abbildung 13 Oben: Indikator mit gl. Durchschnitt, mit Verschiebung & vorgegebenen Daten.  
Unten: Indikator mit DFA, mit Verschiebung & vorgegebenen Daten.

Man kann eindeutig sehen, dass der Indikator mit dem DFA viel mehr unterschiedliche Signale gibt, als der mit dem gleitenden Durchschnitt. Weiter erkennt man, dass der DFA die China-Krise im Jahr 2015 zu erkennen scheint. Dort hat er eine ziemlich lange „cash“ Phase, welche positiv zu werten ist. Abgesehen von diesen zwei Punkten, scheinen beide Indikatoren ähnliche und gute Signale zu geben. Besonders in den Rezessionen geben sie häufig das Signal „cash“. In den folgenden Abbildungen wurden die Trading-Signale beider Indikator auf den S&P 500 angewendet.

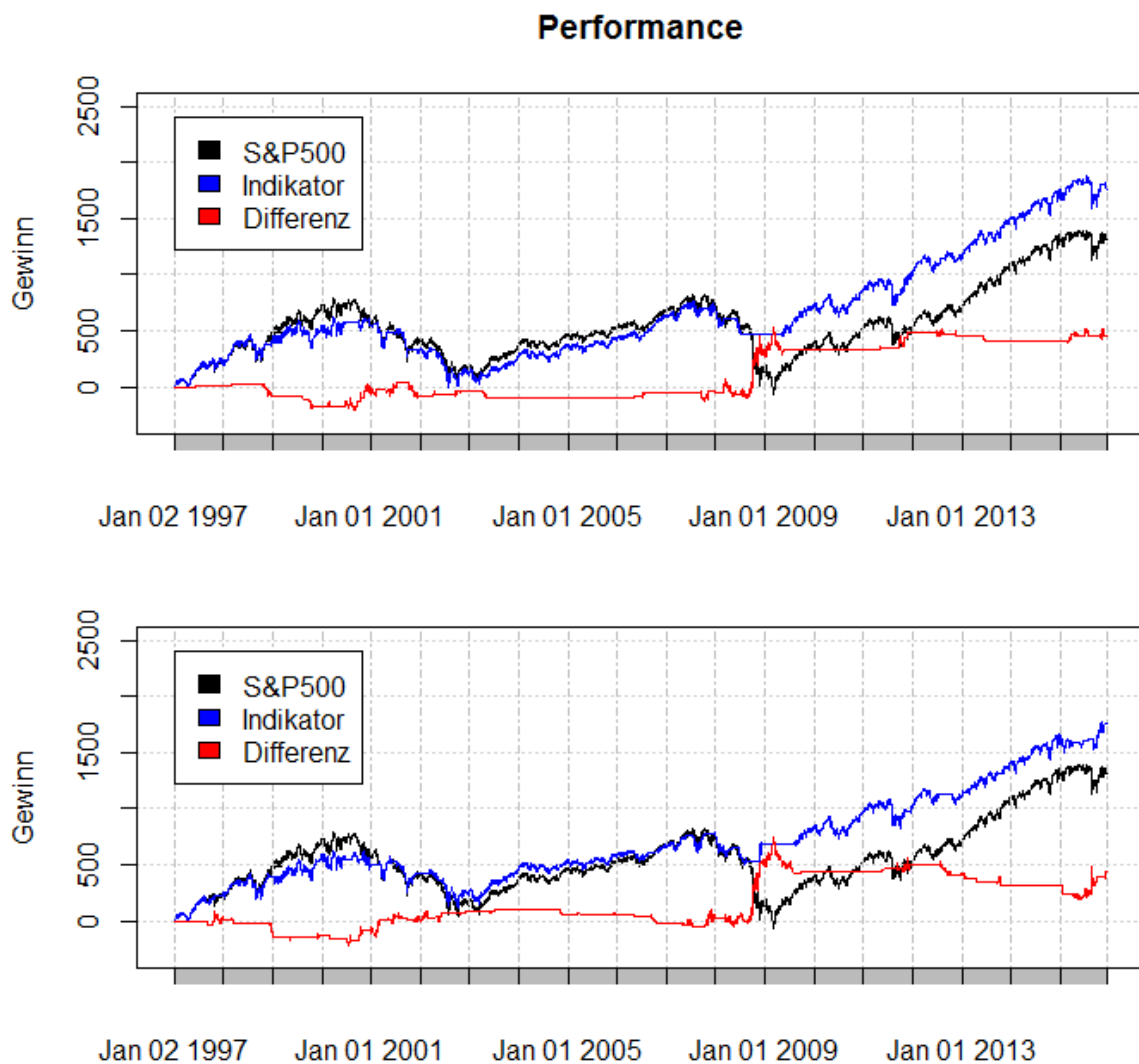


Abbildung 14 Oben: Indikator mit gl. Durchschnitt, mit Verschiebung & vorgegebenen Daten.  
Unten: Indikator mit DFA, mit Verschiebung & vorgegebenen Daten.

Allgemein sieht man, dass beide Indikatoren Mühe mit der ersten Rezession im Jahr 2001 haben. Der Indikator mit dem DFA fängt die Rezession allerdings besser auf als der mit dem gleitenden Durchschnitt. Die zweite Rezession im Jahr 2008 erkennen beide sehr gut. Doch auch hier hat der Indikator mit dem DFA die besseren Signale gegeben. Auch bei der China-Krise im Jahr 2015 ist der Indikator mit dem DFA besser. Der mit dem gleitenden Durchschnitt erkennt diesen kleinen Einbruch fast nicht. Nur bei den steigenden Phasen des S&P 500 ist dieser Indikator besser.

Um den ganzen Vergleich noch ein wenig quantitativer zu bewerten, wurden einige Zahlen berechnet.

- Abschluss: Wert am Ende des Tests
- Prozentualer Vor-/Nachteil: Differenz der Abschlüsse in Prozent.
- Max. Vorteil: Der maximale Vorteil gegenüber dem S&P 500
- Max. Nachteil: Der maximale Nachteil gegenüber dem S&P 500
- Durchschnitt: Der Durchschnittliche Vor-/Nachteil gegenüber dem S&P 500

	S&P 500	DFA	gleit. Durch.
Abschluss	1310.61	1795.92	1771.90
Prozentualer Vor-/Nachteil		37.03%	35.20%
max. Vorteil		784.94	543.51
max. Nachteil		-172.64	-198.1
Durchschnitt		197.22	124.84

Tabelle 3 Performance Werte aus Test-Datei für Indikator mit DFA/gl. Durchschnitt, mit Verschiebung, mit vorgegebenen Daten

Die Vermutung aus den vorhergehenden Abbildungen, dass der Indikator mit dem DFA besser ist, bestätigt sich durch die Tabelle 3. Der Indikator mit dem DFA schneidet bei allen Werten besser ab als der mit dem gleitenden Durchschnitt. Jedoch ist zu bemerken, dass er nur dank der China-Krise den gleitenden Durchschnitt beim Abschluss überholen konnte. Speziell zu bemerken ist, dass der max. Nachteil und der Durchschnitt mit dem DFA besser sind. Dies könnte vor allem für Langzeitanleger von grosser Bedeutung sein.

## 4.5 Fazit

Abschliessend kann man sagen, dass der Indikator mit dem DFA besser ist als mit dem gleitenden Durchschnitt. Er erkennt die Rezessionen insgesamt besser. Jedoch reagiert er in den Anstiegen nicht so gut. Auch in allen anderen Vergleichswerten schneidet der DFA besser ab. Zu dem Problem mit der nicht eindeutig erkannten Rezession im Jahr 2001, stellt sich die Frage, ob das GDP überhaupt die geeignete Zielvariable für das Investieren in den S&P 500 ist. Denn bei allen Varianten die man getestet hat, wurde die erste Rezession schlecht erkannt. Möglicherweise regiert das GDP in diesem Zeitraum nicht stark genug. Um auch in der ersten Rezession gute Ergebnisse zu erzielen, müsste man vielleicht noch andere Zielvariablen in Betracht ziehen.



## 5 Indikator vs. GDP-Indikator

In diesem Kapitel werden die Ergebnisse des besten Indikators in Echtzeit berechnet. Zusätzlich wird das Problem mit der ersten Rezession im Jahr 2001 untersucht. Dazu wurden die Signale des Indikators berechnet, wenn er das originale GDP anstelle einer Regression verwendet. Dazu hat man die Datei *GDP-Indikator* erstellt. In dieser werden wieder alle Variablen definiert und dann die Hauptfunktionen aufgerufen. Am Ende folgt eine grafische Darstellung zur Überprüfung. Für die Performance und die Trading-Signale wurde die schon existierenden Dateien *Test* und *Performance* verwendet. Mit dem Unterschied, dass für den *GDP-Indikator* der Prognoseintervall keine Rolle spielt, da keine wirkliche Regression durchgeführt wird. Folglich ist der *GDP-Indikator* immer in Echtzeit getestet. Die Regression wurde nur durchgeführt, damit das Format für die Hauptfunktion *Signal* stimmt. Dazu wurde das GDP als Zielvariable und als einzige erklärende Variable definiert. Für den Indikator wurde ein tägliches Prognoseintervall gewählt, damit auch dieser in Echtzeit getestet wird.

### Trading-Signal

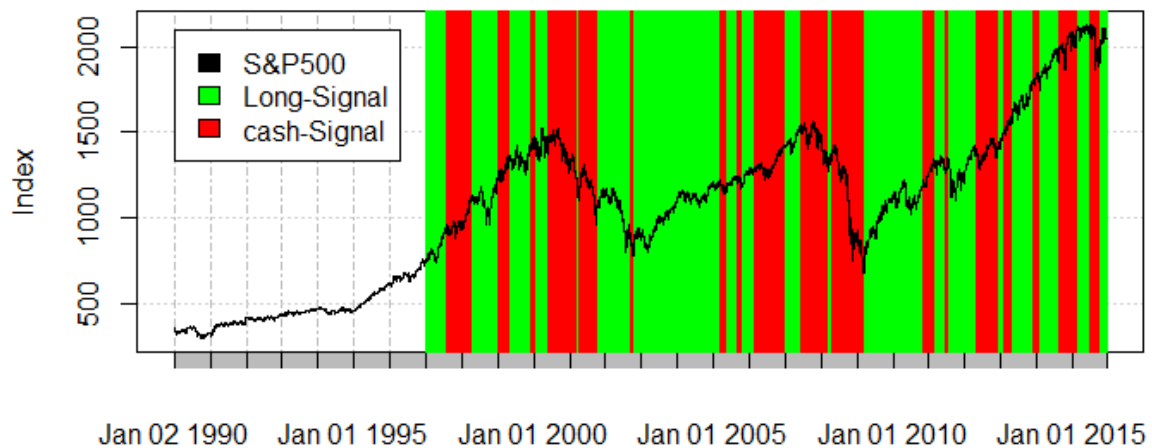
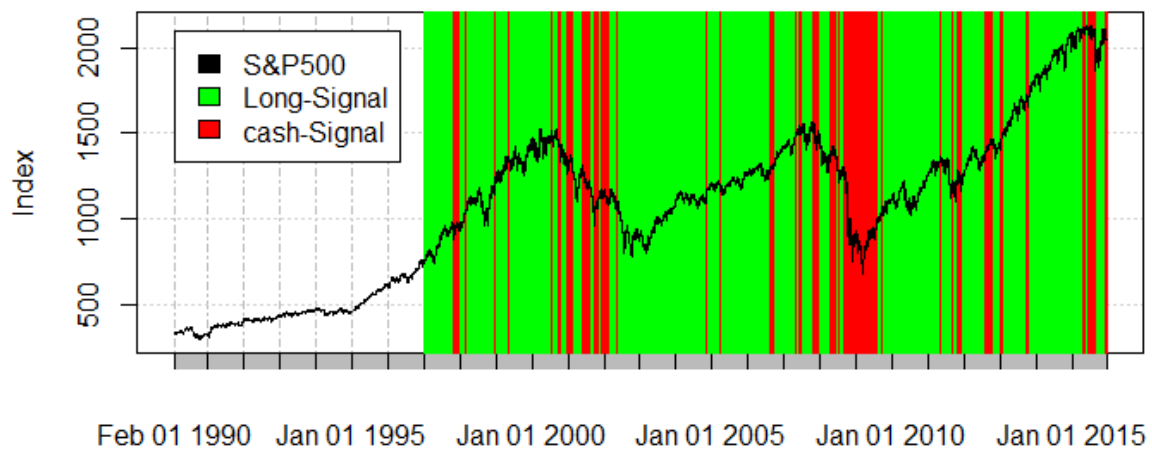


Abbildung 15 Oben: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten.  
Unten: Indikator mit original GDP (versch=0.001, len=1.5)

Man sieht dass der GDP-Indikator viel mehr „cash“-Signale gibt, als der Indikator. Jedoch sind diese nicht nur in den Rezessionen, sondern erscheinen auch häufig in den Anstiegen des S&P 500. Es wird vermutet, dass in den Rezessionen der GDP-Indikator besser sein wird, jedoch nicht in den Anstiegen.

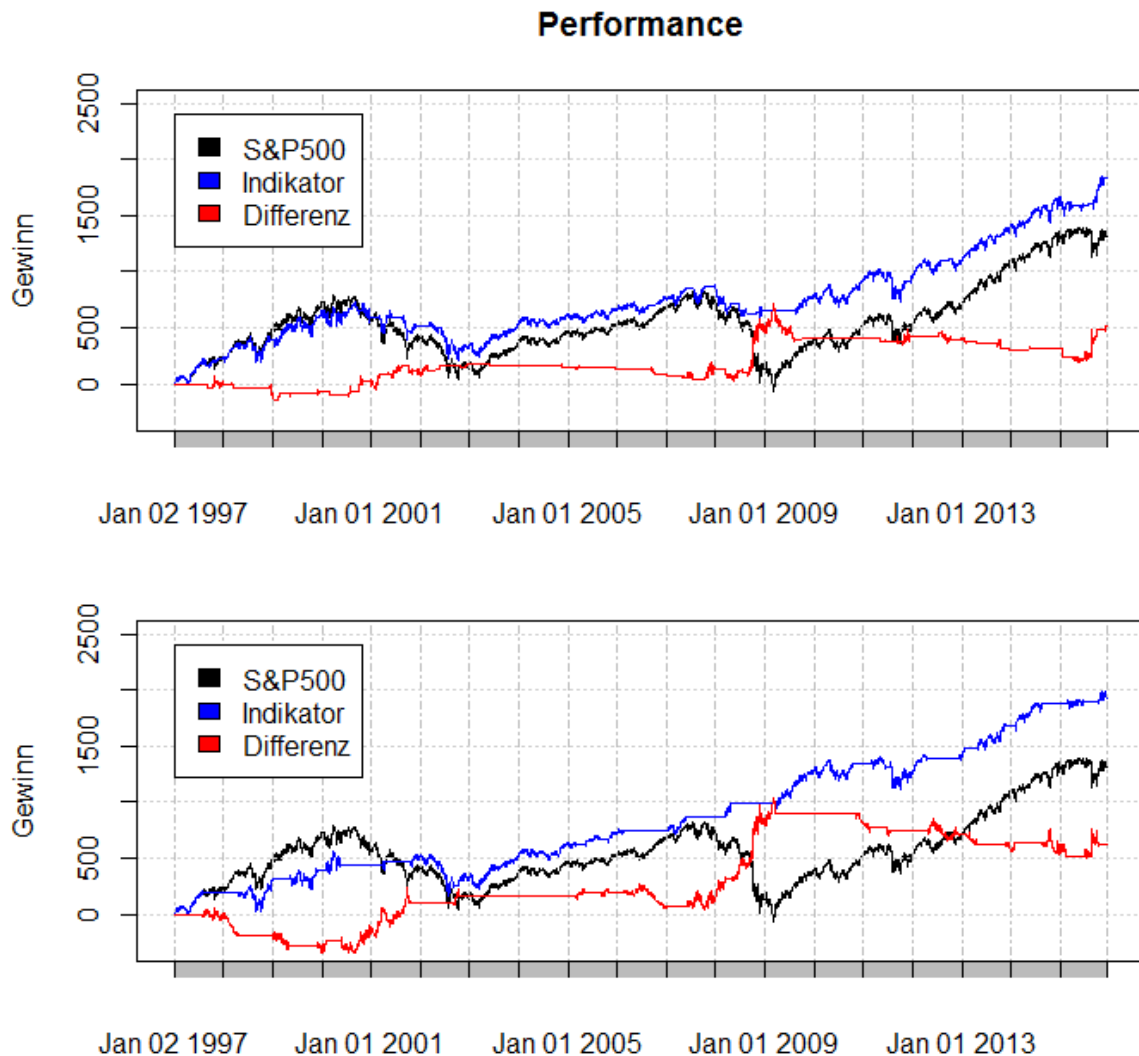


Abbildung 16 Oben: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten.  
Unten: Indikator mit original GDP (versch=0.001, len=1.5)

Wie vermutet sieht man in der Performance, dass der *GDP-Indikator* in den Aufwärtstrends mehr Mühe zu haben scheint als der Indikator. Extrem ist es in der Phase vor der ersten Rezession. Jedoch minimiert er den Verlust in den Rezessionen besser. Der Verdacht, dass das GDP als Zielvariable der Grund für die schlechte Performance in der ersten Rezession ist, bestätigt sich hier trotzdem. Denn der GDP-Indikator gibt bereits beim ersten Anstieg sowie in der darauf folgenden Rezession teilweise schlechte Signale. Dazu ist der Indikator etwas stabiler und weniger verlustanfällig.

Um die Performance noch etwas quantitativer betrachten zu können, wurden wieder einige Werte berechnet.

	S&P 500	Eigener Indikator	GDP-Indikator
Abschluss	1310.61	1832.25	1993.96
Prozentualer Vor-/Nachteil		39.8%	52.14%
max. Vorteil		712.77	1044.25
max. Nachteil		-137.15	-348.25
Durchschnitt		189.29	306.12

Tabelle 4 Performance Werte aus Test-Datei für Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten und für Indikator mit original GDP (versch=0.001, len=1.5)

Trotz des hohen „Durchschnitts“ des GDP-Indikators zeigt sich mit dem „maximalen Nachteil“, dass er verlustanfälliger ist. Erstaunlich ist, dass der Indikator einen tieferen „maximalen Nachteil“ hat. Vermutlich ist das, weil das originale GDP ausserhalb der Rezessionen stärker schwankt und somit immer wieder unter die Signalgrenze fällt, obwohl der S&P 500 steigt. In allen anderen Vergleichswerten ist der GDP-Indikator besser.

Abschliessend lässt sich sagen, dass das GDP zumindest in der ersten Rezession im Jahr 2001 und in den Anstiegen des S&P 500 teilweise keine gute Zielvariable ist.

## 6 Optimierung

Am Ende der Projektarbeit wurde der gesamte R-Code noch einmal verbessert und versucht zu optimieren um ein noch besseres Ergebnis erzielen zu können. Die wichtigste Änderung betrifft die Hauptfunktion *Regression*. Bei dieser wurde ursprünglich die Rückwärtsselektion der Variablen, mit dem AIC-Kriterium, auf das Regressionsmodell angewendet. Diese wurde durch ein Shrinkage-Verfahren ersetzt. Dabei werden die erklärenden Variablen nicht mehr aus dem Modell eliminiert sondern geschrumpft. Dafür wurde das LASSO-Verfahren (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator) verwendet, welches unter der Library lars vorhanden ist. Es wird das Modell mit dem kleinsten RSS-Wert (**R**esidual **S**um of **S**quares) verwendet und mit diesem gefittet.

Die Motivation um diese Optimierung einfließen zu lassen, war die Tatsache, dass bei sehr vielen verschiedenen erklärenden Variablen die Performance des Indikators erheblich nachliess. Diese Wirkung lässt sich erklären, dass bei vielen Variablen die eigentlichen Effekte durch das Rauschen überdeckt werden.

In den folgenden Abbildungen wird der überarbeitete Indikator mit dem LASSO-Verfahren mit dem bisherigen Indikator verglichen. Für beide Indikatoren werden die vorgegebenen und die zusätzlichen zwei Daten verwendet.

### GDP mit Signalgrenze

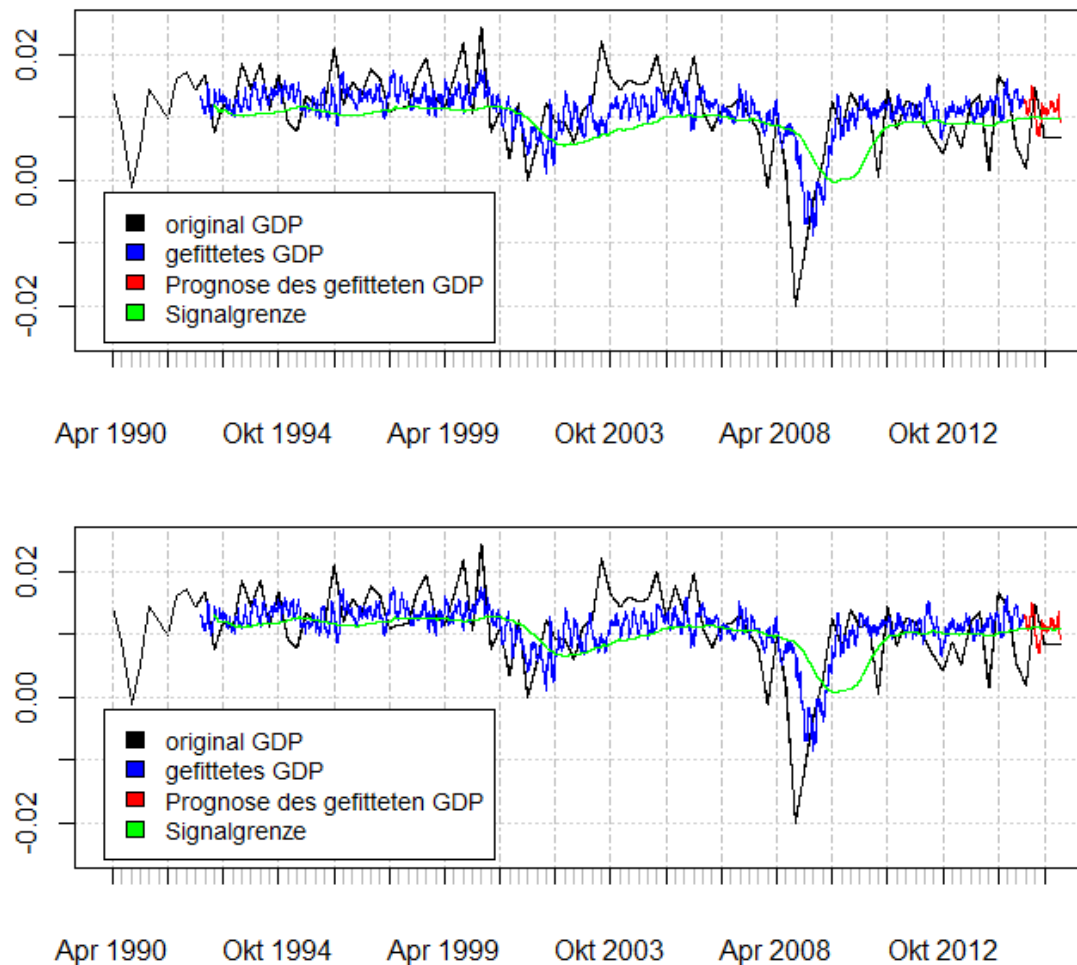


Abbildung 17 Links: Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten. Links mit Backwards Regression, rechts mit Lasso Regression.

Die Regression sieht nahezu identisch aus. Folglich wird die Performance vermutlich auch sehr ähnlich sein.

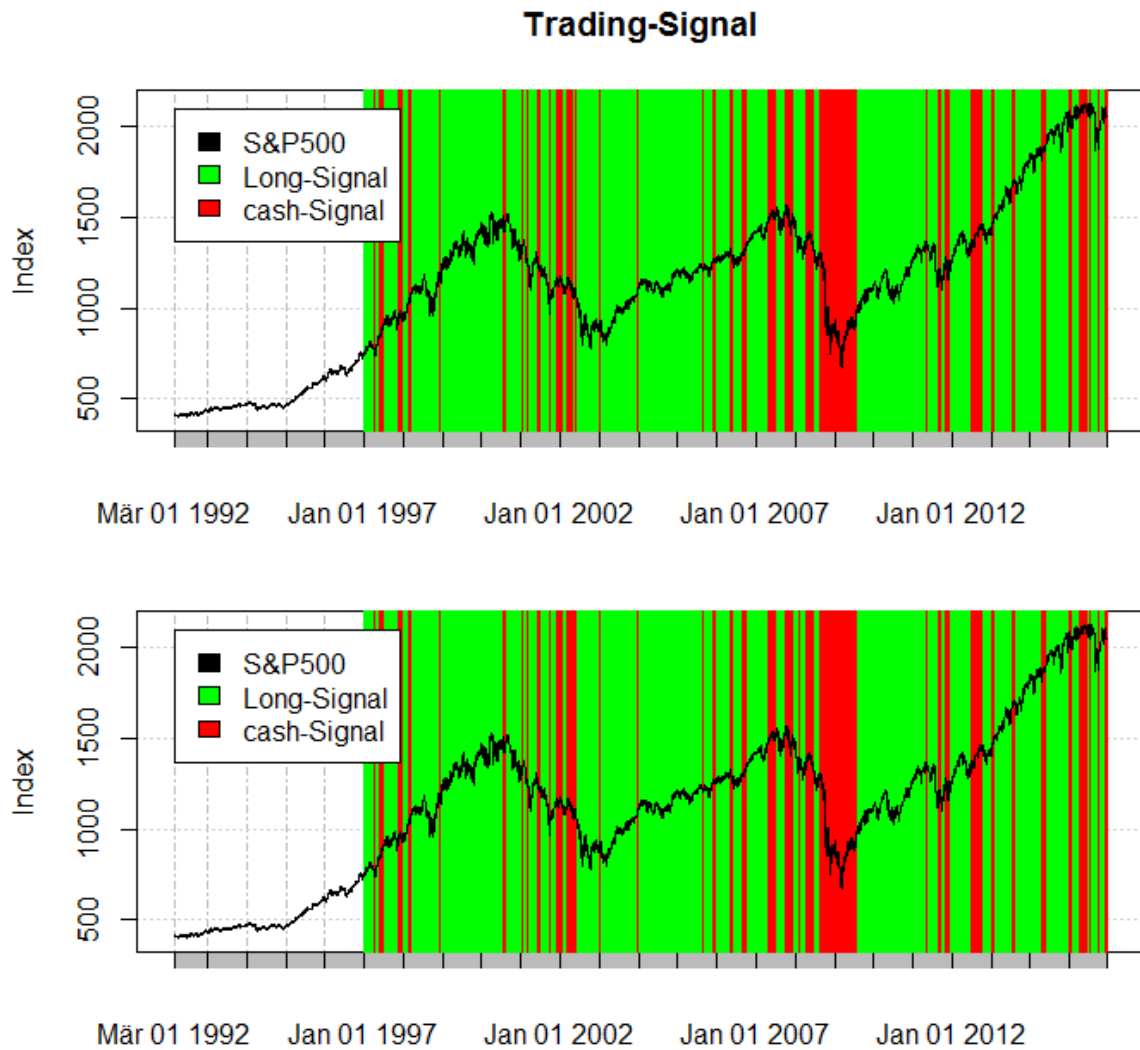


Abbildung 18 Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten.  
Oben mit Backwards Regression, unten mit Lasso Regression.

Auch bei den Signalen setzt sich die Ähnlichkeit der Regression fort. Die Signale sind nahezu identisch.

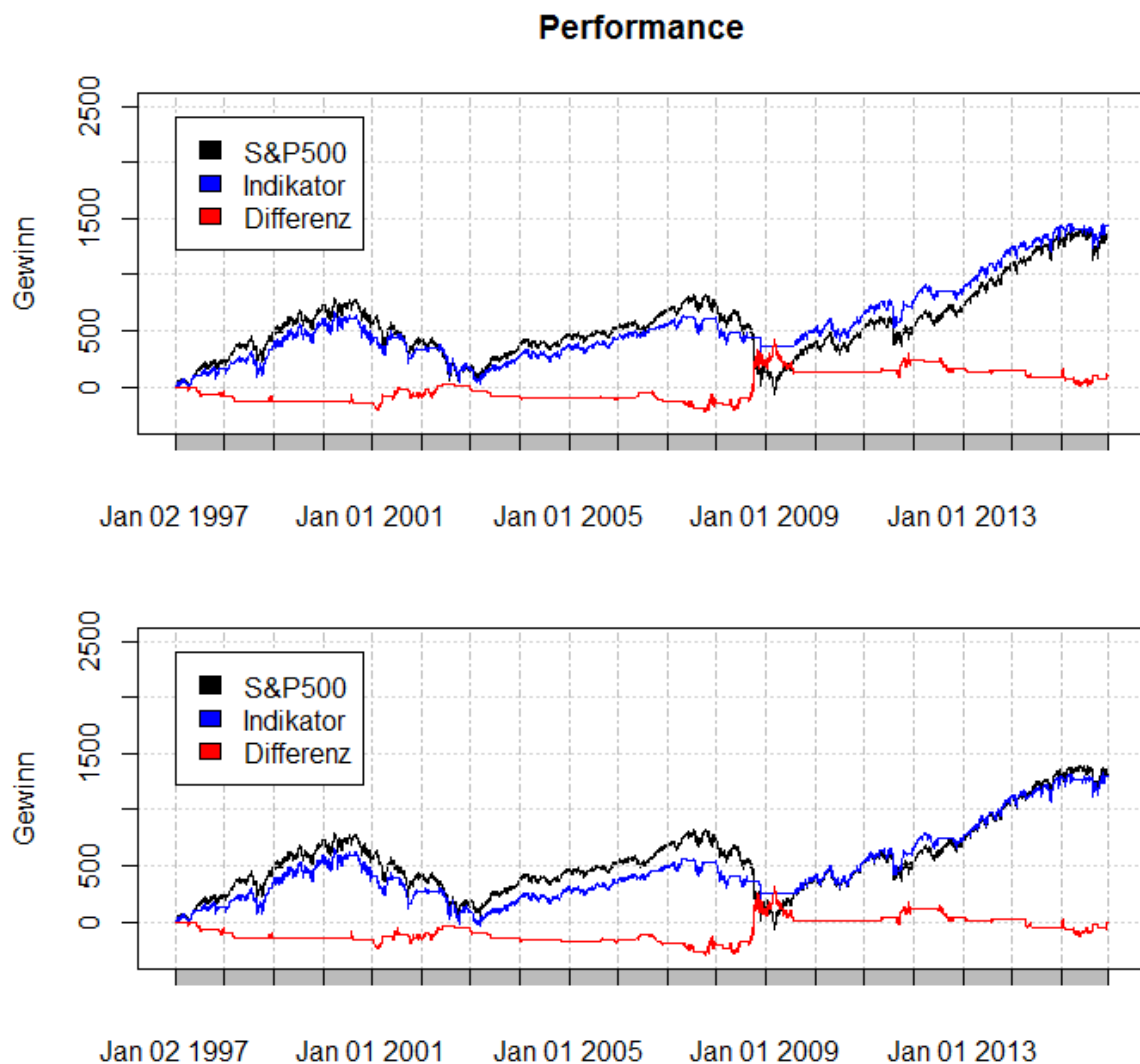


Abbildung 19 Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten. Oben mit Backwards Regression, unten mit Lasso Regression.

Wie vermutet sieht die Performance sehr ähnlich aus. Man kann jedoch erkennen, dass das LASSO-Verfahren bei den Aufwärtstrends ein wenig schlechter ist, als die Rückwärtsselektion der Variablen.

Leider muss man sagen, dass die Änderung auf das LASSO-Verfahren, mit nur den zwei zusätzlichen Daten, nicht viel gebracht hat. Im Gegenteil, es hat sich tendenziell verschlechtert. Möglicherweise lohnt sich ein Shrinkage-Verfahren erst, wenn man bedeutend mehr Variablen zu Verfügung hat.

Die Optimierung wurde so eingefügt, damit bei der Ausführung des Indikators die ursprüngliche Version aufgerufen wird.



## 7 Schlusswort

Das Ziel dieser Arbeit war das Erstellen eines Indikators, welcher auf dem geschätzten GDP beruht und zusätzlich für den S&P 500 die Signale „long“ oder „cash“ berechnet. Dabei sollten vor allem die Rezessionen erkannt werden, damit in diesen Perioden der Verlust minimiert werden kann.

Der erstellte Indikator wurde in folgendem Szenario getestet: Man investiert ab 01.01.1997 bis 09.12.2015 in den S&P 500. Somit sind zwei Rezessionen im gewählten Testzeitraum. Dem Indikator wurden Daten ab 01.01.1990 zur Verfügung gestellt.

In diesem Szenario konnte der erstellte Indikator beide Rezessionen erkennen, wobei er bei der ersten im Jahr 2001 Mühe hatte. Trotzdem konnte er dort den Verlust verkleinern. Die zweite Rezession im Jahr 2008 wurde sehr gut erkannt und der Verlust konnte erheblich minimiert werden. Nicht sehr schön ist, dass der Indikator in den Phasen, in denen der S&P 500 steigt, immer leichte Verluste generiert. Trotzdem konnte über den ganzen Testzeitraum ein prozentualer Vorteil von 39.8% gegenüber dem S&P 500 erzielt werden. Insgesamt hatte er einen maximalen Nachteil von -137.15, einen maximalen Vorteil von 712.77 und im Schnitt einen Vorteil von 189.29 erreicht.

Verbesserungspotential hat der Indikator in der Berechnung der Signalgrenzen. Man könnte genauer untersuchen, welches die besten Werte für die Variablen *versch* und *len* sind. Oder man berechnet sie komplett anders. Zum Beispiel könnte man die Werte nicht mehr mit dem Mittelwert berechnen, sondern unterschiedliche Gewichtungen verwenden. Darüber hinaus könnte man untersuchen, ob eine andere Zielvariable als das GDP bessere Ergebnisse erzielt. Interessant wäre es auch noch weitere Filter zu untersuchen, da in dieser Arbeit nur der gleitende Durchschnitt und der Direct Filter Approach untersucht wurden. Natürlich könnte auch die Regression noch weiter verbessert werden.

## 8 Verzeichnisse

### 8.1 Literaturverzeichnis

- Wildi, Marc (2015): Direct Filter Approach (DFA)
- Ruckstuhl, Andreas (2015) : Statistisches Modellieren. Statistische Regressionsrechnung und ihre Anwendung
- Tobias, Liboschik / Verena, Ueberfeld (2009): Verfahren zur Variablenselektion und Modellschrumpfung im linearen Regressionsmodell (13.11.2009). URL: [https://www.statistik.tu-dortmund.de/fileadmin/user\\_upload/Lehrstuehle/Genetik/MW0910/Vortrag20091113\\_Liboschik\\_Ueberfeldt.pdf](https://www.statistik.tu-dortmund.de/fileadmin/user_upload/Lehrstuehle/Genetik/MW0910/Vortrag20091113_Liboschik_Ueberfeldt.pdf) [Stand: 13.12.2015]

### 8.2 Abbildungsverzeichnis

Abbildung 1	Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten.....	14
Abbildung 2	Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten.....	15
Abbildung 3	Oben: Indikator mit gl. Durchsch., ohne Verschiebung, mit vorgegebenen Daten.....	16
Abbildung 4	Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	17
Abbildung 5	Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	18
Abbildung 6	Indikator mit gl. Durchschnitt, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	18
Abbildung 7	Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten. ....	20
Abbildung 8	Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten. ....	21
Abbildung 9	Oben: Indikator mit DFA, ohne Verschiebung, mit vorgegebenen Daten. ....	22
Abbildung 10	Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	23
Abbildung 11	Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	24
Abbildung 12	Indikator mit DFA, mit Verschiebung, mit vorgegebenen und zusätzlichen Daten.....	24

Abbildung 13	Oben: Indikator mit gl. Durchschnitt, mit Verschiebung & vorgegebenen Daten. Unten: Indikator mit DFA, mit Verschiebung & vorgegebenen Daten. ....	26
Abbildung 14	Oben: Indikator mit gl. Durchschnitt, mit Verschiebung & vorgegebenen Daten. Unten: Indikator mit DFA, mit Verschiebung & vorgegebenen Daten. ....	27
Abbildung 15	Oben: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten. ....	30
Abbildung 16	Oben: Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten. ....	31
Abbildung 17	Links: Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten. Links mit Backwards Regression, rechts mit Lasso Regression. ....	34
Abbildung 18	Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten. Oben mit Backwards Regression, unten mit Lasso Regression. ....	35
Abbildung 19	Indikator mit DFA, mit Verschiebung, mit vorgegebenen & zusätzlichen Daten. Oben mit Backwards Regression, unten mit Lasso Regression. ....	36

### 8.3 Tabellenverzeichnis

Tabelle 1	Vorgegebene Daten aus der Aufgabenstellung .....	4
Tabelle 2	Selbst recherchierte zusätzliche Daten aus dem Internet .....	5
Tabelle 3	Performance Werte aus Test-Datei für Indikator mit DFA/gl. Durchschnitt, mit Verschiebung, mit vorgegebenen Daten.....	28
Tabelle 4	Performance Werte aus Test-Datei für Indikator mit DFA, mit Verschiebung, mit vorgegebenen Daten und für Indikator mit original GDP (versch=0.001, len=1.5) .....	32

### 8.4 Datenverzeichnis

- Gross domestic Product, URL: <https://www.quandl.com/data/FRED/GDP-Gross-Domestic-Product-1-Decimal> [Stand: 09.12.2015]
- S&P 500, URL: [https://www.quandl.com/data/YAHOO/INDEX\\_GSPC-S-P-500-Index](https://www.quandl.com/data/YAHOO/INDEX_GSPC-S-P-500-Index) [Stand: 09.12.2015]
- VIX S&P 500, URL: [https://www.quandl.com/data/YAHOO/INDEX\\_VIX-VIX-S-P-500-Volatility-Index](https://www.quandl.com/data/YAHOO/INDEX_VIX-VIX-S-P-500-Volatility-Index) [Stand: 09.12.2015]
- Initial Claims, URL: <https://www.quandl.com/data/FRED/ICSA-Initial-Claims> [Stand: 09.12.2015]

- Industrial Production Index, URL :  
<https://www.quandl.com/data/FRED/INDPRO-Industrial-Production-Index>  
[Stand: 09.12.2015]
- Non farm payroll, URL: <https://www.quandl.com/data/FRED/PAYEMS-All-Employees-Total-nonfarm> [Stand: 09.12.2015]
- Civilian Unemployment Rate, URL :  
<https://www.quandl.com/data/FRED/UNRATE-Civilian-Unemployment-Rate> [Stand : 09.12.2015]
- Personal Consumption Expenditures, URL :  
<https://www.quandl.com/data/FRED/PCE-Personal-Consumption-Expenditures> [Stand: 09.12.2015]
- Manufactures New Orders : Durable Goods, URL :  
<https://www.quandl.com/data/FRED/DGORDER-Manufacturers-New-Orders-Durable-Goods> [Stand : 09.12.2015]

## 9 Anhang

### 9.1 Aufgabenstellung

- Ziel:
  - 'Buy-and hold' Strategie auf S&P500 schlagen
    - Draw-downs während Rezessionen mittels Informationen über das wirtschaftliche Umfeld mildern/abfangen.
- Daten
  - Provider: Quandl
    - Konto einrichten
  - Reihenauswahl
    - Quartal
      - GDP
    - Monat:
      - Industrieproduktion
      - Beschäftigung
      - Non farm payroll
    - Wochen:
      - Weekly initial claims
    - Tagesdaten
      - S&P 500
      - VIX
  - Transformation:
    - Log-returns
  - Zeitspanne:
    - 1990-aktuell
- Problem:
  - Daten sind verrauscht und müssen bereinigt/gefiltert werden.
    - Filter
      - Unterricht: DFA
      - Startphase
        - Einfache Filter (gleitende Durschnitte mit fixen Gewichten, exponential smoothing: einfach/doppelt)
  - Daten aggregieren zu einem Index für den S&P500 (Bsp: gleichgewichtet; Achtung Vorzeichen!)
    - Unterschiedliche Abtastfrequenzen
- Outcome: Index mit Tradingrule

## 9.2 R-Code

### 9.2.1 Indikator

#### 9.2.1.1 Main

```
##Voreinstellungen
#-----

#Packages
install.packages("devtools")
library(devtools)
install_github('quandl/R-package')
library(Quandl)
install.packages("lubridate")
library(lubridate)
install.packages("xts")
library("xts")
#install.packages("lars")
#library("lars")

#Pfade
path.dat <- "C:/Users/holzesev/Dropbox/Projektarbeit/final/"
path.dat <- "C:/Users/Adrian/Dropbox/Projektarbeit/final/"
setwd(path.dat)

#Quandel Code (keine Begrenzung der Dowloadmenge)
Quandl.api_key("zt5r7MUZLsVrqXchV1xA")

#Source
source(paste(path.dat, "Hauptfunktionen/Read.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Filter.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Regression.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Signal.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Plot1.R", sep=""))
#-----

##Durch Benutzer auszufüllen
#-----

#Mögliche Variablen(lag, select):

#FRED/GDP (0, 1):
#   Name: Gross domestic Product
#   Beschreibung: Bruttoinlandprodukt der USA
#   since: 1947-01-01
#   Frequency: quarterly
#
```

#YAHOO/INDEX\_GSPC (0, 4):

# Name: S&P500  
# Beschreibung: Aktien der 500 grössten Börsennotierten US  
# Amerikanischen Unternehmen  
# since: 1950-01-03  
# Frequency: daily  
#

---

#YAHOO/INDEX\_VIX (0, 4):

# Name: Volatilität(VIX) von S&P500  
# Beschreibung: Misst Schwankung vom S&P500  
# since: 1990-01-02  
# Frequency: daily  
#

---

#FRED/ICSA (0, 1):

# Name: Initial Claims  
# Beschreibung: Zahl der Erstanträge für Arbeitslosengeld  
# since: 1967-01-07  
# Frequency: weekly  
#

---

#FRED/INDPRO (1, 1):

# Name: Industrial Production Index  
# Beschreibung: Industrieproduktion der USA  
# since: 1919-01-01  
# Frequency: monthly  
#

---

#FRED/PAYEMS (1, 1):

# Name: Non farm payroll  
# Beschreibung: Alle Angestellten ohne Farm- oder non-Profit-  
# Unternehmen  
# since: 1939-01-01  
# Frequency: monthly  
#

---

#FRED/UNRATE (1, 1):

# Name: Civilian Unemployment Rate  
# Beschreibung: Zahl von Arbeitslosen  
# since: 1948-01-01  
# Frequency: monthly  
#

---

```
#FRED/PCE (1, 1):
#   Name: Personal Consumption Expenditures
#   Beschreibung: Private Konsumausgaben
#   since: 1959-01-01
#   Frequency: monthly
# _____

#FRED/DGORDER (1, 1):
#   Name: Manufactures New Orders: Durable Goods
#   Beschreibung: Neu Auftragseingänge für Gebrauchsgüter
#   since: 1992-02-01
#   Frequency: monthly
# _____

#Zielvariable und erklärende Variablen
asset <- c("FRED/GDP", "YAHOO/INDEX_GSPC", "YAHOO/INDEX_VIX",
          "FRED/ICSA", "FRED/INDPRO", "FRED/PAYEMS", "FRED/UNRATE")

#asset <- c("FRED/GDP", "YAHOO/INDEX_GSPC", "YAHOO/INDEX_VIX",
#          "FRED/ICSA", "FRED/INDPRO", "FRED/PAYEMS", "FRED/UNRATE",
#          "FRED/PCE", "FRED/DGORDER")

#Lag der Zeitreihen (versetzten Daten = 1, sonst 0)
lag <- c(0,0,0,0,1,1,1)
#lag <- c(0,0,0,0,1,1,1,1,1)

#Lag wird benötigt, da bei z.B. Monatsdaten die Daten für den ganzen
#Monat am Anfang des Monats dargestellt werden.

#Eingabe in welcher Spalte gewünschte Daten sind
select <- c(1,4,4,1,1,1,1)
#select <- c(1,4,4,1,1,1,1,1,1)

#Bei Quandl gibt es verschiedene Formate (da verschiedene #Herausgeber)
#für die verschiedenen Zeitreihen. Darum muss man von #Hand eingeben, in
#welcher Spalte sich die gewünschten Daten befinden. #-> siehe Quandl
#(Bsp. FRED/UNRATE (TABLE))

#Zusammenfügen zu einer Matrix
asset_mat <- cbind(asset, lag, select)
View(asset_mat) #zur Kontrolle

#letzte Rezession fertig
last_Rez <- "2009-06-15"

#Startzeit
start <- "1990-01-01"
```



```
#Endzeit
end <- format(Sys.Date())

#Filter (dfa_ms, filt_gleit_D)
filt <- "dfa_ms"
#filt <- "filt_gleit_D"
#-----
##Hauptfunktionen
#-----
#Daten einlesen
data <- Read(asset_mat, start, end)

#Speichern
save(data, file = paste(path.dat, "Daten/data.Rdata", sep=""))
#Daten laden von Festplatte (fakultativ)
load(file = paste(path.dat, "Daten/data.Rdata", sep=""))

#Filtern
mydata.gef <- Filter(data$mydata_log, filt)

#Speichern
save(mydata.gef, file = paste(path.dat, "Daten/mydata.gef.Rdata", sep=""))
#gefilterte Daten laden von Festplatte (fakultativ)
load(file = paste(path.dat, "Daten/mydata.gef.Rdata", sep=""))

#Regression
reg <- Regression(mydata.gef)

#Speichern
save(reg, file = paste(path.dat, "Daten/reg.Rdata", sep=""))
#durchgeführte Regression laden von Festplatte (fakultativ)
load(file = paste(path.dat, "Daten/reg.Rdata", sep=""))

#Signal
sig <- Signal(reg, last_Rez, filt)

#Speichern
save(sig, file = paste(path.dat, "Daten/sig.Rdata", sep=""))
#Signal laden von Festplatte (fakultativ)
load(file = paste(path.dat, "Daten/sig.Rdata", sep=""))

#heutiges Trading-Signal
paste("Gehen Sie bitte", sig$entscheid[length(sig$entscheid),1])

#Plotten1 (transformierte Daten/ gefilterte Daten)
Plot1(data$mydata_log, mydata.gef)
```

```
#Darstellung der Näherung für das GDP
par(mfrow=c(1,1))
plot(na.exclude(data$mydata_log[,1]), ylim=c(-0.025, 0.025), main="GDP
mit Signalgrenze")
lines(mydata.gef[,1])
lines(reg$dat.gefittet, col="blue")
lines(reg$prog, col="red")
lines(sig$rez_sig, col="green")
legend(list(x = .index(data$mydata_log)[1], y=-0.002),
       legend=c("original GDP", "gefittetes GDP", "Prognose des
gefitteten GDP", "Signalgrenze"),
       fill=c("black", "blue", "red", "green"), cex = 0.9)
```

### 9.2.1.2 Read

```
#Daten einlesen
Read <- function(asset_mat, start, end){

  #alle Tage vom Start- bis Enddatum
  alle_tage <- seq.Date(as.Date(start), as.Date(end), "day")

  #xts mit Index (alle_tage) erstellen
  mydata <- as.xts(rep(NA, length(alle_tage)), order.by=alle_tage)

  for(i in 1:nrow(asset_mat)){

    #einzelne Zeitreihen von Quandl laden
    quandli <- Quandl(asset_mat[i,1], start_date=start, end_date=end,
                      type='xts')[, as.integer(asset_mat[i,3])]

    #Überprüfen ob Lag (asset_mat$lag)
    if(as.integer(asset_mat[i,2]) == 0){
    }else{

      #Lag einfügen
      lag_s <- lag.xts(quandli, k=as.integer(asset_mat[i,2]))

      #wieder Anhängen des letzten Eintrages
      quandli <- c(lag_s, quandli[length(quandli)])

      #Änderung des Datums
      index(quandli)[length(quandli)] <-
      index(quandli)[length(quandli)-1]+.1

      #Überprüfen ob Zeitreihen in Zukunft geht
      if(index(quandli)[length(quandli)-1]+.1 > end){

        #Abschneiden der Zukunft
        quandli <- quandli[-length(quandli)]
      }
    }
  }
}
```

```
    }  
  }  
  
  #Anhängen der Zeitreihe  
  mydata <- cbind(mydata, as.xts(na.omit(quandli)))  
}  
  
#NA Werte löschen (bei Erstellung nötig)  
mydata <- mydata[,-1]  
  
mydata_log <- mydata  
  
#Transformation  
for(i in 1:ncol(mydata_log)){  
  
  #Log>Returns und Anhängen  
  mydata_log <- cbind(mydata_log,  
                      diff(log(na.exclude(mydata_log[,i]))))  
}  
  
#ursprüngliche Daten entfernen  
mydata_log <- mydata_log[,-(1:(ncol(mydata_log)/2))]  
  
#Aufsplitten der Namen  
nam <- NA  
for(i in 1:nrow(asset_mat)){  
  nam[i] <- unlist(strsplit(asset_mat[i,1], split = "/"))[2]  
}  
  
#Spaltenbezeichnung ändern  
colnames(mydata) <- nam  
colnames(mydata_log) <- nam  
  
#NA's auffüllen mit vorhergehenden Wert (nur bei mydata)  
for(k in 1:ncol(mydata)){  
  for(i in 2:nrow(mydata)){  
    if(is.na(mydata[i,k])){  
      mydata[i,k] <- mydata[i-1,k]  
    }  
  }  
}  
mydata <- na.exclude(mydata)  
  
return(list(mydata=mydata, mydata_log=mydata_log))  
}
```

### 9.2.1.3 Filtern

```
#Daten filtern
Filter <- function(mydata_log, filt){

  #Source
  source(paste(path.dat, "Unterfunktionen/filt_gleit_D.R", sep=""))
  source(paste(path.dat, "Unterfunktionen/lin_interpol.R", sep=""))
  source(paste(path.dat, "Unterfunktionen/per.R", sep=""))
  source(paste(path.dat, "Unterfunktionen/dfa_ms.R", sep=""))

  #Daten kopieren
  mydata.gef <- mydata_log

  #Lineare Interpolation auf Zielvariable anwenden
  mydata.gef <- cbind(mydata.gef, lin_interpol(mydata.gef[,1]))

  if(filt=="filt_gleit_D"){
    #Filter auf erklärende Zeitreihen anwenden
    for(i in 2:(ncol(mydata.gef)-1)){

      #Durchschnitt über die letzten 3 Monate (inkl. Heute)
      mydata.gef <- cbind(mydata.gef, filt_gleit_D(mydata.gef[,i],3))
    }

    #NA's auffüllen mit vorhergehenden Wert
    for(k in 1:ncol(mydata.gef)){
      for(i in 2:nrow(mydata.gef)){
        if(is.na(mydata.gef[i,k])){
          mydata.gef[i,k] <- mydata.gef[i-1,k]
        }
      }
    }
    mydata.gef <- na.exclude(mydata.gef)
  }

  if(filt=="dfa_ms"){
    #NA's auffüllen mit vorhergehenden Wert
    for(k in 1:ncol(mydata.gef)){
      for(i in 2:nrow(mydata.gef)){
        if(is.na(mydata.gef[i,k])){
          mydata.gef[i,k] <- mydata.gef[i-1,k]
        }
      }
    }
    mydata.gef <- na.exclude(mydata.gef)
    #Filter auf erklärende Zeitreihen anwenden
    for(n in 2:ncol(mydata_log)){
      periodogram <- per(mydata.gef[,n], F)$per
    }
  }
}
```

```
K <- length(periodogram)-1
cutoff<-pi/45
Gamma<-((0:K)*pi/K)<cutoff
L <- 90
Lag <- 0
dfa <- dfa_ms(L,periodogram,Lag,Gamma)
b <- dfa$b
x <- mydata.gef[,n]
for (i in L:length(x))
{
  x[i]<-dfa$b%%mydata.gef[i:(i-L+1),n]
}
mydata.gef <- cbind(mydata.gef, x)
}
mydata.gef <- mydata.gef[-(1:(L-1))]
```

```
}

#ursprüngliche Daten entfernen
mydata.gef <- mydata.gef[,-(1:(ncol(mydata.gef)/2))]
```

```
#Spalten beschriften
colnames(mydata.gef) <- colnames(mydata_log)

return(mydata.gef)
}
```

#### 9.2.1.4 Regression

```
#Regression
Regression <- function(mydata.gef) {

  #Source
  source(paste(path.dat,"Unterfunktionen/correlation.R",sep=""))

  #beste Korrelation finden und richtig verschieben 50 Zeitschritte
  mydata.best_cor <- correlation(mydata.gef, 51)

  #Um gleichlange Zeitreihen zu erhalten
  mydata.reg <- na.exclude(mydata.best_cor)

  #Nur die Daten welche ein Jahr zurück liegen (d.h 12) benutzen, da
  #das GDP bis dorhin plus/ minus richtig ist.
  mydata.reg <- mydata.reg[1:which(index(mydata.reg)==
                                index(mydata.reg[dim(mydata.reg)[1]])
                                %m-%months(12)),]

  #Die Namen der Zeitreihen speichern
  nam <- colnames(mydata.reg)
```

```
#Regressionsmodell definieren
#Zielvariable für lars
#y <- mydata.reg[,1]
#erklärende Variablen für lars
#x <- mydata.reg[,2:ncol(mydata.reg)]

#Regressionsmodell für lm
reg_mod <- paste(c(nam[1], paste(nam[2:length(nam)], collapse="+")),
                collapse="~")

#Regression durchführen
#Lasso (Least Absolute Shrinkage and Selection Operator)
#mod <- lars(x, y, type="lasso")
#best_step <- mod$df[which.min(mod$RSS)]

#AIC-Kriterium
lm_mod <- lm(reg_mod, data=mydata.reg, x=T, y=T)
mod <- step(lm_mod, direction="backward")

#Matrix überschreiben
new.dat <- na.exclude(mydata.best_cor)

#erklärende Variablen, mit welchen die Zielvariable geschätzt werden
#soll
new.dat <- new.dat[(which(index(mydata.reg[dim(mydata.reg)[1]])==
                        index(new.dat))+1):
                  which(index(new.dat[dim(new.dat)[1]])==
                        index(new.dat)),]
new.dat <- new.dat[,-1]

#Prognose
#lars
#prog <- as.xts(predict(mod, new.dat, s=best_step, type="fit")$fit,
#                  order.by=index(new.dat))
#step
prog <- as.xts(predict(mod, new.dat, typ=NULL),
               order.by=index(new.dat))
colnames(prog) <- nam[1]

#gefittete Werte
#lars
#dat.gefittet <- as.xts(predict(mod, x, s=best_step,
#                              type="fit")$fit,
#                      order.by=index(mydata.reg))
#step
dat.gefittet <- as.xts(predict(lm_mod, mydata.reg[,-1], typ=NULL,
                              order.by=index(mydata.reg)))
```

```
colnames(dat.gefittet) <- nam[1]

return(list(dat.gefittet=dat.gefittet, prog=prog))
}
```

### 9.2.1.5 Signal

```
#signal(reg, last_Rez)
Signal <- function(reg, last_Rez, filt){

  #Index speichern bis Heute
  ind <- c(as.Date(index(reg$dat.gefittet))+1,
           as.Date(index(reg$prog)))

  #Daten zusammenfügen (Gefittet, Prognose)
  rez_sig <- rbind(reg$dat.gefittet, reg$prog)

  #Index übergeben
  index(rez_sig) <- ind

  #Daten bis letzte Rezession löschen
  rez_sig <- rez_sig[-(1:(which(index(rez_sig)==last_Rez)-1)),]
  #Dies ist wichtig, damit die Rezession das Signal nicht verfälscht

  #letzter/ erster Eintrag löschen
  dat_rez_sig <- rez_sig[-length(rez_sig)]
  rez_sig <- rez_sig[-1]
  index(dat_rez_sig) <- index(rez_sig)
  #Heutiges Signal wird auf Daten bis und mit Gestern geschätzt

  #Über wieviele Jahre zurück das gerechnet wird
  len <- 1.5

  if(filt=="filt_gleit_D"){
    versch <- 0.002
  }
  if(filt=="dfa_ms"){
    versch <- 0.001
  }

  #Mittelwert der letzten len Jahre (Ausnahme wenn Rezession in
  #Intervall)
  if(nrow(rez_sig)!=0){
    for(i in 1:length(rez_sig)){
      if(i>len*365){
        rez_sig[i] <- mean(dat_rez_sig[(i-len*365+1):i])-versch
      }else{
        rez_sig[i] <- mean(dat_rez_sig[1:i])-versch
      }
    }
  }
```

```
    }

    #Vektor mit long/ short Signalen
    entscheid <- dat_rez_sig-rez_sig
    for(i in 1:length(entscheid)){
      if(entscheid[i]>=0){
        entscheid[i] <- "long"
      }else{
        entscheid[i] <- "cash"
      }
    }
  }else{
    entscheid <- NULL
  }

  return(list(rez_sig=rez_sig, entscheid=entscheid))
}
```

## 9.2.2 Unterfunktionen

### 9.2.2.1 Lin\_interpol

```
#Lineare Interpolation
lin_interpol <- function(dat){

  #nur zu Interpolierende Daten abspeichern (ohne NA's)
  z <- na.exclude(dat)

  for(i in 1:(length(z)-1)){

    #Anzahl Zeilen zwischen zu interpolierenden Daten
    n <- which(index(dat)==index(z[i+1])) -
      which(index(dat)==index(z[i]))-1

    #von interpolierenden Wert
    start_zeile <- which(index(dat)==index(z[i]))

    #bis zu interpolierenden Wert
    end_zeile <- which(index(dat)==index(z[i+1]))

    #Interpolieren zwischen verschiedenen Werten
    dat[start_zeile:end_zeile] <- approx(z[i:(i+1)], n=n)$y
  }
  return(dat)
}
```



### 9.2.2.2 filt\_gleit\_D

```
#gleitender Durchschnitt
filt_gleit_D <- function(dat, a){

  #Index abspeichern
  ind <- index(dat)

  #leerer Vektor für gefilterte Spalten
  z <- NA

  #Datum für erstmöglichen gefilterten Wert
  start_dat <- index(na.exclude(dat)[1])%m+%months(a)-1
  #Bsp. 2015-05-01: Filtern aller Daten von 2015-05-01 bis 2015-02-02
  #(bei a=3 Monate) daher das -1. Der 2015-02-01 gehört nicht zu den
  #zu filternden Daten.

  #entsprechende Zeile zum Datum von oben
  start_zeile <- which(index(dat)==start_dat)

  #Werte filtern!!!!
  for(i in start_zeile:length(dat)){
    if(is.na(dat[i])){
      z[i] <- NA
    }else{

      #Start für filter für jeweiligen Zeitpunkt
      von_Zeile <- which(index(dat)==index(dat[i])%m-%months(a)+1)

      #in vorübergehenden Vektor speichern
      z[i] <- mean(na.exclude(dat[von_Zeile:i]))
    }
  }

  #Index wieder anfügen
  z <- as.xts(z, order.by = ind)
  return(z)
}
```

### 9.2.2.3 dfa\_ms

```
dfa_ms<-function(L,periodogram,Lag,Gamma)
{
  K<-length(periodogram)-1
  X<-exp(-1.i*Lag*pi*(0:(K))/(K))*rep(1,K+1)*sqrt(periodogram)
  X_y<-exp(-1.i*Lag*pi*(0:(K))/(K))*rep(1,K+1)
  for (l in 2:L)      #l<-L<-21
  {
    X<-cbind(X, (cos((l-1-Lag)*pi*(0:(K))/(K)) +
                  1.i*sin((l-1-Lag)*pi*(0:(K))/(K)))*sqrt(periodogram))
    X_y<-cbind(X_y, (cos((l-1-Lag)*pi*(0:(K))/(K)) +
                     1.i*sin((l-1-Lag)*pi*(0:(K))/(K))))
  }
  xtx<-t(Re(X))%*%Re(X)+t(Im(X))%*%Im(X)
  # MA-Filtercoefficients
  b<-as.vector(solve(xtx)%*%(t(Re(X_y))%*(Gamma*periodogram)))
  # Transferfunction
  trffkt<-1:(K+1)
  trffkt[1]<-sum(b)
  for (k in 1:(K)) #k<-1
  {
    trffkt[k+1]<-(b%*%exp(1.i*k*(0:(length(b)-1))*pi/(K)))
  }
  return(list(b=b,trffkt=trffkt))
}
```

### 9.2.2.4 per

```
# Periodogram-Funktion
per<-function(x,plot_T)
{
  len<-length(x)
  per<-0:(len/2)
  DFT<-per

  for (k in 0:(len/2))
  {
    cexp <- complex(arg=-(1:len)*2*pi*k/len)
    DFT[k+1]<-sum(cexp*x*sqrt(1/(2*pi*len)))
  }
  per<-abs(DFT)^2
  if (plot_T)
  {
    par(mfrow=c(2,1))
    plot(per,type="l",axes=F,xlab="Frequency",ylab="Periodogram",
         main="Periodogram")
    axis(1,at=1+0:6*len/12,labels=c("0","pi/6","2pi/6","3pi/6",
                                     "4pi/6","5pi/6","pi"))
  }
}
```

```
axis(2)
box()
plot(log(per), type="l", axes=F, xlab="Frequency",
      ylab="Log-periodogram", main="Log-periodogram")
axis(1, at=1+0:6*len/12, labels=c("0", "pi/6", "2pi/6", "3pi/6",
                                   "4pi/6", "5pi/6", "pi"))
axis(2)
box()
}
return(list(DFT=DFT, per=per))
}
```

### 9.2.2.5 correlation

#höchste betragsmäßige Korrelation finden für shift-1 Verschiebungen

```
correlation <- function(dat, shift){

  #Matrix erstellen für die verschiedenen Korrelationen
  cor_mat <- matrix(NA, ncol=dim(dat)[2]-1, nrow=shift)

  #Namen der Zeitreihen überschreiben
  colnames(cor_mat) <- colnames(dat)[-1]

  #verschiedene Korrelationen berechnen von 0:shift-1.
  #Method=spearman, da die Beobachtungen nicht normalverteilt sind.
  for(n in 1:dim(cor_mat)[2]){
    for(i in 1:shift){
      cor_mat[i,n] <- cor(dat[i:length(dat[,1]),1],
                          dat[1:(length(dat[,n+1])+1-i),n+1],
                          method="spearman")
    }
  }

  #leerer Vektor definieren
  best_lag <- NA

  #den Lag mit der höchsten Korrelation im Vektor abspeichern
  for(i in 1:dim(cor_mat)[2]){
    best_lag[i] <- which(max(abs(cor_mat[,i]))==abs(cor_mat[,i]))-1
  }

  #Daten in neue Matrix speichern
  mydata.best_cor <- dat

  #morgiges Datum herauslesen
  morgiges_dat <- index(dat[dim(dat)[1]])+1

  #Für jede Zeitreihe den Lag ausführen
```

```
for(i in 2:dim(mydata.best_cor)[2]){
  lag <- best_lag[i-1]
  if(lag!=0){

    #Daten abspeichern
    ind <- seq.Date(morgiges_dat, morgiges_dat+lag,"day")

    #Lag durchführen
    lag_s <- lag.xts(dat[,i], k=lag)

    #restliche Daten wieder anfügen
    z <- dat[(dim(dat)[1]-lag):dim(dat)[1],i]
    index(z) <- ind
    lag_s <- c(lag_s,z)
  }else{
    lag_s <- mydata.best_cor[,i]
  }

  #Die neuen Daten anhängen
  mydata.best_cor <- cbind(mydata.best_cor,as.xts(na.omit(lag_s)))
}

#ursprüngliche Daten entfernen
mydata.best_cor <- mydata.best_cor[,-(2:(dim(dat)[2]))]

#Spalten beschriften
colnames(mydata.best_cor) <- colnames(dat)

return(mydata.best_cor)
}
```

## 9.2.3 Test

### 9.2.3.1 Test

```
#Test
source(paste(path.dat,"Test/Performance.R", sep=""))

#start Investition
start_invest <- "1997-01-01"

#bei erster Erhebung Daten bekannt von start_invest+abstand
abstand <- "months"

#gefilterte Daten (aus main)
mydata.gef <- mydata.gef
#nur zur Kontrolle ob mydata.gef existiert.

#Berechnung der Performance
```

```
per <- Performance(start_invest, mydata.gef, abstand, data)

#Entscheidung (long/ cash)
ent <- per$ent

#Gewinn
gew <- per$gew

#Darstellung der Trading-Signale und des S&P 500
plot(data$mydata[,2], type="n", main="Trading-Signal", ylab="Index")
for(i in 1:length(ent)){
  if(ent[i]=="cash"){
    abline(v=.index(ent[i]), col="red")
  }else{
    abline(v=.index(ent[i]), col="green")
  }
}
lines(data$mydata[,2])
legend(list(x = .index(data$mydata[1,2]), y=2100),
       legend=c("S&P500", "Long-Signal", "cash-Signal"),
       fill=c("black", "green", "red"))

#S&P 500 Index-Wert
index <- data$mydata[-(1:(length(data$mydata[,2])-length(gew))),2]
index <- index-as.double(index[1])

#Differenz(Indikator-S&P500) wenn Strategie angewendet
reingewinn <- cumsum(gew)-index

#Darstellung der Performance
plot(index ,ylim=c(-300,2500), main="Performance", ylab="Gewinn")
lines(cumsum(gew), col="blue")
lines(reingewinn, col = "red")
legend(list(x = .index(ent[1]), y=2400),
       legend=c("S&P500", "Indikator", "Differenz"),
       fill=c("black", "blue", "red"))

#Performance Werte:
#letzter Wert des S&P500:
Abschluss_SP500 <- index[length(index)]

#Wert der Differenz(Indikator-S&P500) wenn Strategie angewendet am ende
des Tests:
Abschluss_Indikator <- cumsum(gew)[length(gew)]

#Prozentualer Vorteil des Indikators gegenüber dem S&P500 am Ende des
#Tests:
Proz_Vorteil <- (100/Abschluss_SP500)*Abschluss_Indikator-100
```

```
#Maximaler Vorteil des Indikators gegenüber dem S%P500 über die ganze
#Test dauer:
maxVorteil <- max(reingewinn)

#Maximaler Vorteil des Indikators gegenüber dem S%P500 über die ganze
#Test dauer:
maxNachteil <- min(reingewinn)

#Durchschnittlicher Vorteil des Indikators gegenüber dem S%P500 über
#die ganze Test dauer:
Durchschnitt <- mean(reingewinn)
```

### 9.2.3.2 Performance

```
Performance <- function(start_invest, mydata.gef, abstand, data){

  #Source
  source(paste(path.dat, "Hauptfunktionen/Regression.R", sep=""))
  source(paste(path.dat, "Hauptfunktionen/Signal.R", sep=""))

  #Sequenz von start_invest bis zum Ende Daten mit Intervall=abstand
  ind <- seq.Date(as.Date(start_invest),
                 as.Date(index(mydata.gef[dim(mydata.gef)[1]])),
                 by=abstand)

  #Enddatum hinzufügen
  if(index(mydata.gef[dim(mydata.gef)[1]])>ind[length(ind)]){
    ind <- c(ind, index(mydata.gef[dim(mydata.gef)[1]]))
  }

  #Sequenz für alle Tage in diesem Zeitraum
  ind_all_Tage <- seq.Date(as.Date(start_invest),
                         as.Date(index(mydata.gef[dim(mydata.gef)[1]])),
                         by="days")

  #Default Wert für Signal, damit kein Fehler berechnet wird
  akt_Rez <- index(mydata.gef[1])+51
  #51, weil wir nicht mehr als 51 Zeitschritte verschieben für die
  #beste Korrelation

  #Schlaufe um Signale (long/ cash) zu berechnen
  for(i in 2:length(ind)){

    #gewünschter Zeitraum aus gefilterten Reihen
    zeit <- mydata.gef[1:which(index(mydata.gef)==ind[i]),]

    #Regression durchführen
    reg <- Regression(zeit)
```

```
#Signale berechnen
sig <- Signal(reg, akt_Rez, filt)

#Zusammenbauen des Vektors mit long / cash Signalen für gesamte
#Zeit
if(length(sig$entscheid)!=0){
  if(i==2){
    ent <- sig$entscheid[-(1:which(index(sig$entscheid)==
                                   ind[i-1])))]
  }else{
    ent <- c(ent, sig$entscheid[-(1:which(index(sig$entscheid)==
                                   ind[i-1]))])
  }
}

print(c(as.character(ind[i]),
        as.character(format(Sys.time(), "%X"))))
}

#Name ändern
colnames(ent)[1] <- paste(colnames(mydata.gef)[2], "_gesch", sep="")

#Zeitreihen herauslesen
invest <- data$mydata[,2]

#kürzen damit gleicher Zeitpunkt
invest <- invest[-(1:(length(invest)-length(ent))),]

#damit gleiche Länge
gew <- ent

#bei start_invest Gewinn=0
gew[1] <- 0

#Gewinn ausrechnen pro Tag
for(i in 2:length(ent)){
  if(ent[i]=="long"){
    gew[i] <- as.double(invest[i])-as.double(invest[i-1])
  }else{
    gew[i] <- 0
  }
}
return(list(ent=ent, gew=gew))
}
```

### 9.2.3.3 GDP-Indikator

```
#Packages
install.packages("devtools")
library(devtools)
install_github('quandl/R-package')
library(Quandl)
install.packages("lubridate")
library(lubridate)
install.packages("xts")
library("xts")
install.packages("shrink")
library("shrink")

#Pfade
path.dat <- "C:/Users/holzesev/Dropbox/Projektarbeit/final/"
path.dat <- "C:/Users/Adrian/Dropbox/Projektarbeit/final/"
setwd(path.dat)

#Quandel Code (keine Begrenzung der Dowloadmenge)
Quandl.api_key("zt5r7MUZLsVrqXchV1xA")

#Source
source(paste(path.dat, "Hauptfunktionen/Read.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Filter.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Regression.R", sep=""))
source(paste(path.dat, "Hauptfunktionen/Signal.R", sep=""))

asset <- c("FRED/GDP", "YAHOO/INDEX_GSPC")
lag <- c(0,0)
select <- c(1,4)
asset_mat <- cbind(asset, lag, select)

last_Rez <- "1993-01-01"
start <- "1990-01-01"
end <- "2015-12-09"
filt <- "dfa_ms"

#Hauptfunktionen
data <- Read(asset_mat, start, end)
mydata.gef <- Filter(data$mydata_log, filt)
mydata.gef[,2] <- mydata.gef[,1]
reg <- Regression(mydata.gef)
sig <- Signal(reg, last_Rez, filt)

#Grafiken
par(mfrow=c(1,1))
plot(na.exclude(data$mydata_log[,1]), ylim=c(-0.025, 0.025), main="GDP
mit Signalgrenze")
```



```
lines(mydata.gef[,1])
lines(reg$dat.gefittet, col="blue")
lines(reg$prog, col="red")
lines(sig$rez_sig, col="green")
legend(list(x = .index(data$mydata_log)[1], y=-0.002),
       legend=c("original GDP", "gefittetes GDP", "Prognose des
                gefitteten GDP", "Signalgrenze"),
       fill=c("black", "blue", "red", "green"), cex = 0.9)
```