Einführung Zeitreihenanalyse

Karsten Lübke

Euro Handelsdaten

Eurostat stellt viele Konjunkturdaten zur Verfügung, u. a. auch Handelsdaten hier. Diese finden sich im Datensatz `euretail` des Paktes `fpp`:

```
# Einmalig installieren:
# install.packages("fpp")

library(fpp) # Paket laden
data(euretail) # Datensatz laden
euretail
```

```
##        Qtr1   Qtr2   Qtr3   Qtr4
## 1996  89.13  89.52  89.88  90.12
## 1997  89.19  89.78  90.03  90.38
## 1998  90.27  90.77  91.85  92.51
## 1999  92.21  92.52  93.62  94.15
## 2000  94.69  95.34  96.04  96.30
## 2001  94.83  95.14  95.86  95.83
## 2002  95.73  96.36  96.89  97.01
## 2003  96.66  97.76  97.83  97.76
## 2004  98.17  98.55  99.31  99.44
## 2005  99.43  99.84 100.32 100.40
## 2006  99.88 100.19 100.75 101.01
## 2007 100.84 101.34 101.94 102.10
## 2008 101.56 101.48 101.13 100.34
## 2009  98.93  98.31  97.67  97.44
## 2010  96.53  96.56  96.51  96.70
## 2011  95.88  95.84  95.79  95.97
```

Es handelt sich also um quartalsweise Daten von 1996, 1 bis 2011, 4.

```
class(euretail)
```

```
## [1] "ts"
```

`ts` steht dabei für ein Zeitreihenobjekt in R. Zeitreihen (d. h. eine Folge von Beobachtungen mit der Zeit – der Wert variiert mit der Zeit) haben viele Besonderheiten, die zur Analyse genutzt werden können.

```
start(euretail) # Startbeobachtung
```

```
## [1] 1996    1
```

```
end(euretail) # Endbeobachtung
```

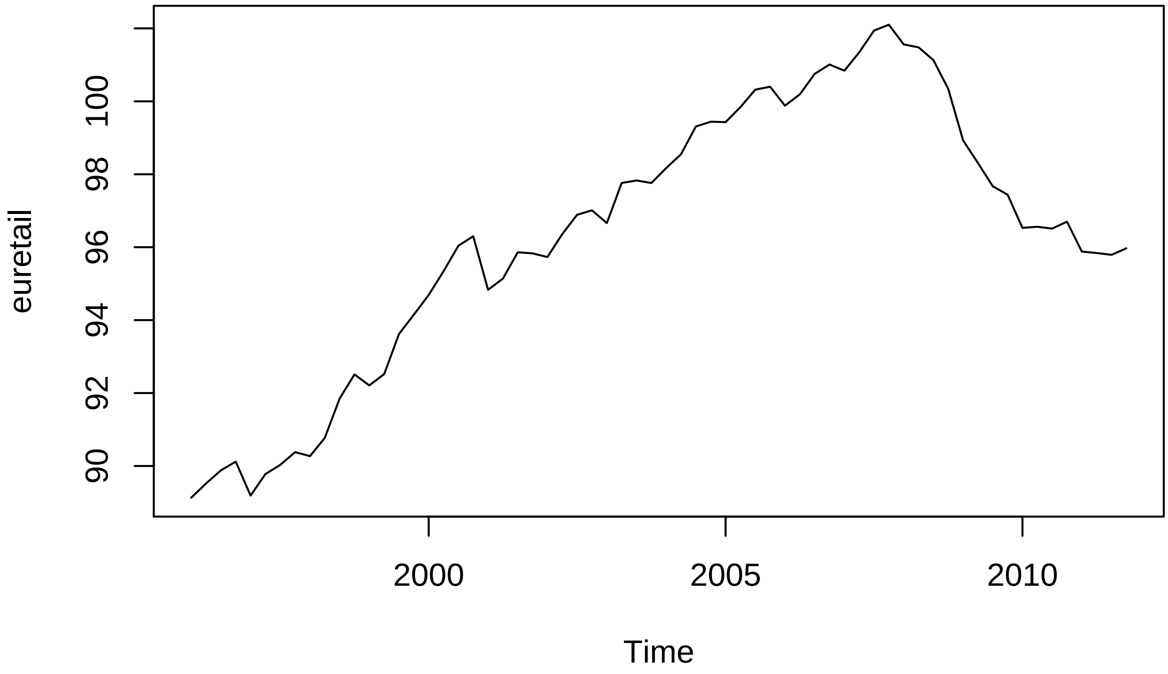
```
## [1] 2011    4
```

```
frequency(euretail) # Beobachtungen je Zeiteinheit
```

```
## [1] 4
```

Eine Abbildung kann wie üblich einfach über `plot()` erzeugt werden.

```
plot(euretail)
```



Man erkennt einen Anstieg bis ca. 2008 (Finanzkrise!), dann einen Abschwung und ab 2010/11 eine evt. Erholung – aber auch saisonale Schwankungen.

Zeitreihenzerlegung

Eine Zeitreihe (`data`) y_t kann in verschiedene Komponenten zerlegt werden:

- Trend (`trend`) M_t
- Saisonkomponenten (`seasonal`)S_t
- Rest-/ Fehlerkomponenten (`remainder`) E_t

Das *additive* Modell lautet dann einfach:

$$Y_t = M_t + S_t + E_t$$

während das *multiplikative* Modell lautet

$$Y_t = M_t \cdot S_t \cdot E_t.$$

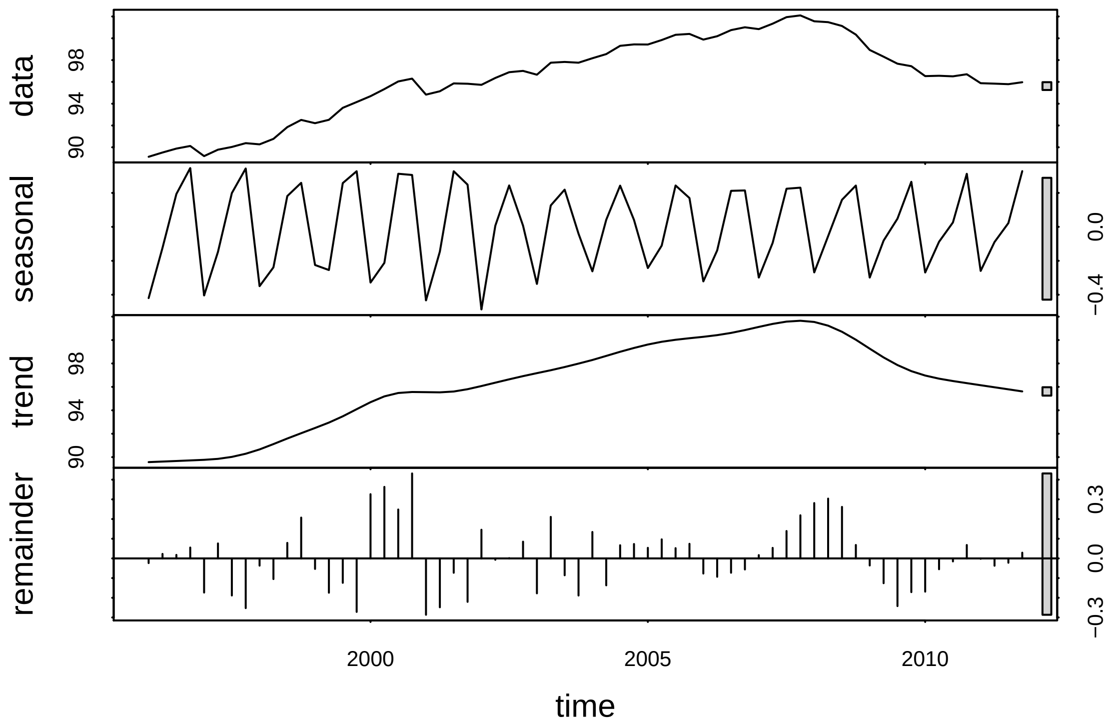
Es gibt viele Möglichkeiten m_t, s_t, z_t zu schätzen, eine gute Umsetzung findet sich in der Funktion `stl()` Hier muss die Option `s.window=` angegeben werden. (Eine Alternative ohne Option ist `decompose()`)

```
fit <- stl(euretail, s.window = 5)
fit
```

```
##  Call:
##  stl(x = euretail, s.window = 5)
##
## Components
##              seasonal    trend    remainder
## 1996 Q1 -0.419860564  89.57375 -0.023884485
## 1996 Q2 -0.123565509  89.62061  0.022953668
## 1996 Q3  0.193599124  89.66846  0.017942030
## 1996 Q4  0.346353935  89.71844  0.055205774
## 1997 Q1 -0.404246348  89.76781 -0.173568415
## 1997 Q2 -0.147484780  89.85132  0.076164034
## 1997 Q3  0.198861912  90.01958 -0.188446028
## 1997 Q4  0.343940521  90.28854 -0.252483392
## 1998 Q1 -0.349545254  90.65637 -0.036827055
```

```
## 1998 Q2 -0.238674319  91.11372 -0.105046188
## 1998 Q3  0.181425885  91.58972  0.078850220
## 1998 Q4  0.259346327  92.04326  0.207394100
## 1999 Q1 -0.224911742  92.48849 -0.053577251
## 1999 Q2 -0.254337444  92.94870 -0.174367468
## 1999 Q3  0.257834148  93.48620 -0.124035417
## 1999 Q4  0.327828800  94.09390 -0.271730904
## 2000 Q1 -0.328353978  94.69217  0.326181855
## 2000 Q2 -0.212084922  95.18876  0.363325738
## 2000 Q3  0.312750693  95.47877  0.248480492
## 2000 Q4  0.305748312  95.56257  0.431679110
## 2001 Q1 -0.433336876  95.54969 -0.286348812
## 2001 Q2 -0.146007433  95.53430 -0.248289419
## 2001 Q3  0.327803754  95.60521 -0.073013319
## 2001 Q4  0.248705735  95.80201 -0.220715694
## 2002 Q1 -0.486645661  96.07108  0.145565611
## 2002 Q2  0.006765829  96.36030 -0.007064601
## 2002 Q3  0.244073433  96.64460  0.001328096
## 2002 Q4  0.006908463  96.91796  0.085127459
## 2003 Q1 -0.336360551  97.17382 -0.177457089
## 2003 Q2  0.126193370  97.42283  0.210975181
## 2003 Q3  0.219287384  97.69652 -0.085805046
## 2003 Q4 -0.040342793  97.98891 -0.188570292
## 2004 Q1 -0.262077562  98.29783  0.134246361
## 2004 Q2  0.041691564  98.64520 -0.136893260
## 2004 Q3  0.242917421  99.00020  0.066885842
## 2004 Q4  0.040843937  99.32594  0.073218914
## 2005 Q1 -0.242801532  99.61905  0.053753440
## 2005 Q2 -0.110382191  99.85354  0.096838071
## 2005 Q3  0.243814592 100.02408  0.052105736
## 2005 Q4  0.169759140 100.15594  0.074303481
## 2006 Q1 -0.321367759 100.27835 -0.076982512
## 2006 Q2 -0.138668418 100.42200 -0.093331710
## 2006 Q3  0.212238735 100.61059 -0.072828685
## 2006 Q4  0.214652787 100.85157 -0.056225555
## 2007 Q1 -0.298954939 101.12212  0.016832862
## 2007 Q2 -0.094424225 101.38092  0.053508420
## 2007 Q3  0.224610074 101.57628  0.139108082
## 2007 Q4  0.230976137 101.64997  0.219053823
## 2008 Q1 -0.268359448 101.54732  0.281044136
## 2008 Q2 -0.054722279 101.23084  0.303880489
## 2008 Q3  0.159364563 100.70890  0.261731581
## 2008 Q4  0.243567144 100.02822  0.068210379
## 2009 Q1 -0.298935355  99.26496 -0.036022855
## 2009 Q2 -0.080976574  98.51702 -0.126046391
## 2009 Q3  0.048157395  97.86404 -0.242192872
## 2009 Q4  0.265446302  97.34591 -0.171352223
## 2010 Q1 -0.268951277  96.96784 -0.168889939
## 2010 Q2 -0.087960678  96.70317 -0.055204499
## 2010 Q3  0.025951837  96.49930 -0.015255811
## 2010 Q4  0.312708115  96.31911  0.068178307
## 2011 Q1 -0.259828801  96.14201 -0.002176461
## 2011 Q2 -0.089538120  95.96674 -0.037205570
## 2011 Q3  0.022179733  95.78986 -0.022036604
## 2011 Q4  0.328374330  95.61244  0.029189348
```

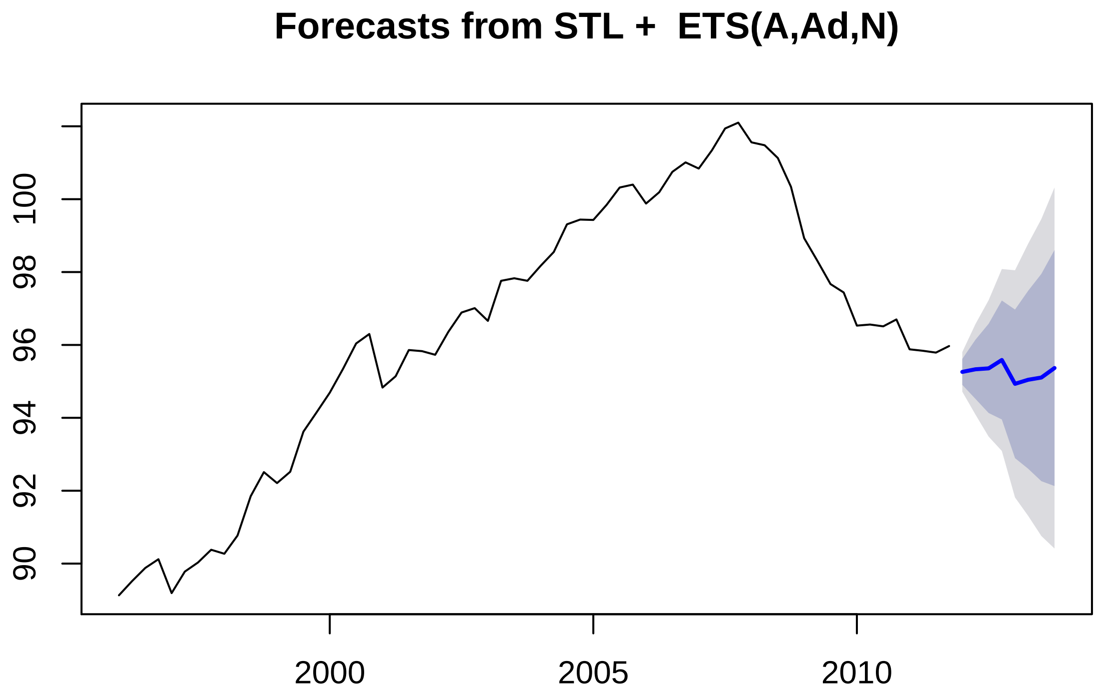
```r
plot(fit)
```



Es gibt eine klare Trendwende in 2008 sowie deutliche Saisoneffekte: der Handel ist am niedrigsten in Q1, am höchsten in Q4.

Eine solche Zerlegung kann übrigens auch als Basis einer Prognose verwendet werden – dabei werden sogar die Prognoseintervalle berechnet. Für Details zur Prognose siehe `?ets`.

```
prog <- predict(fit)
plot(prog)
```



Gleitende Durchschnitte

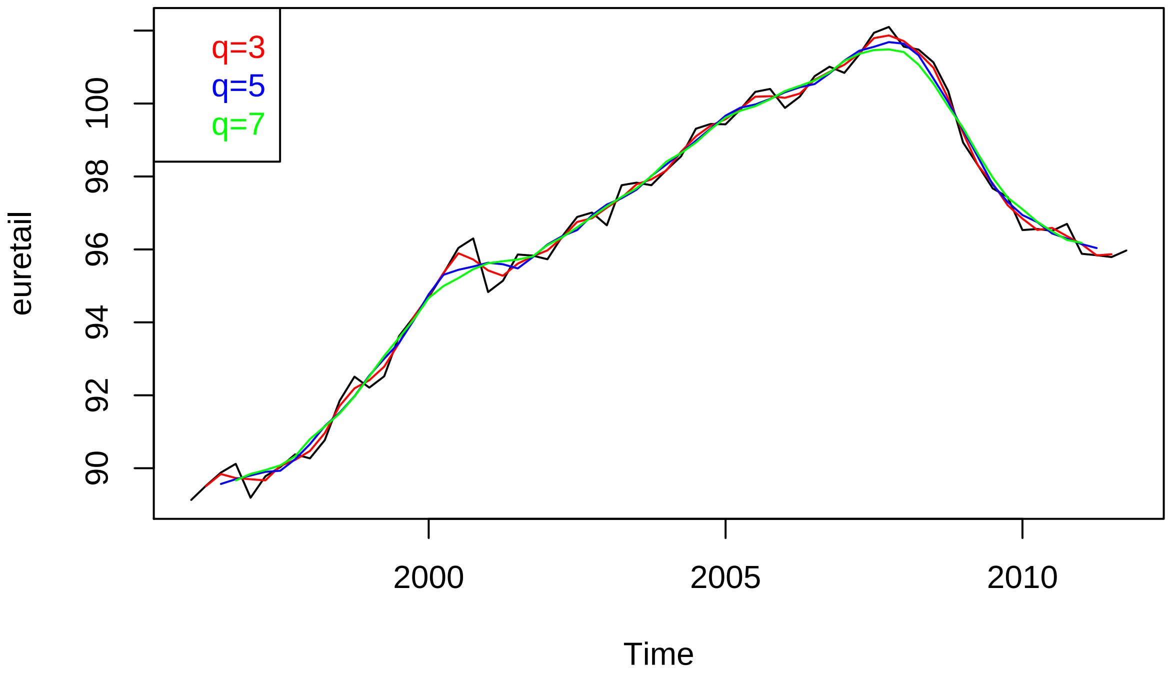
Eine einfache Möglichkeit eine Zeitreihe zu glätten, sind gleitende Durchschnitte (moving-average, Funktion `ma()` aus dem Paket `forecast`). Dazu wird einfach der Mittelwert der q Beobachtungen vor und nach t, inkl. t, berechnet:

$$\hat{y}_{q,t} := \frac{1}{p} \sum_{j=-q}^{q} y_{t+j}$$

mit $p = 2q + 1$.

Je größer q (`order`) ist, desto glatter wird die Zeitreihe:

```
plot(euretail)
lines(ma(euretail, order=3), col="red")
lines(ma(euretail, order=5), col="blue")
lines(ma(euretail, order=7), col="green")
legend("topleft", legend=c("q=3", "q=5", "q=7"), text.col=c("red", "blue", "green"))
```



Übung: Produktionsdaten

Der Datensatz `elecequip` aus dem Paket `fpp` enthält monatliche Produktionsdaten von elektronischen Equipment (näheres siehe hier).

Wie würden Sie die Zeitreihe beschreiben?

Literatur

- Paul S. P. Cowpertwait, Andrew V. Metcalfe (2009): *Introductory Time Series with R*, Kapitel 1
- Rob J. Hyndman, George Athanasopoulos (2012): *Forecasting: principles and practice*, https://www.otexts.org/fpp, Kapitel 6
- Rainer Schlittgen (2012): *Angewandte Zeitreihenanalyse mit R*, Kapitel 2

Lizenz

Versionshinweise:

- Datum erstellt: 2017-03-27
- R Version: 3.3.3