

Prüfung

CAS Datenanalyse Modul B2

Stefan Schmidt

10.06.2020

a)

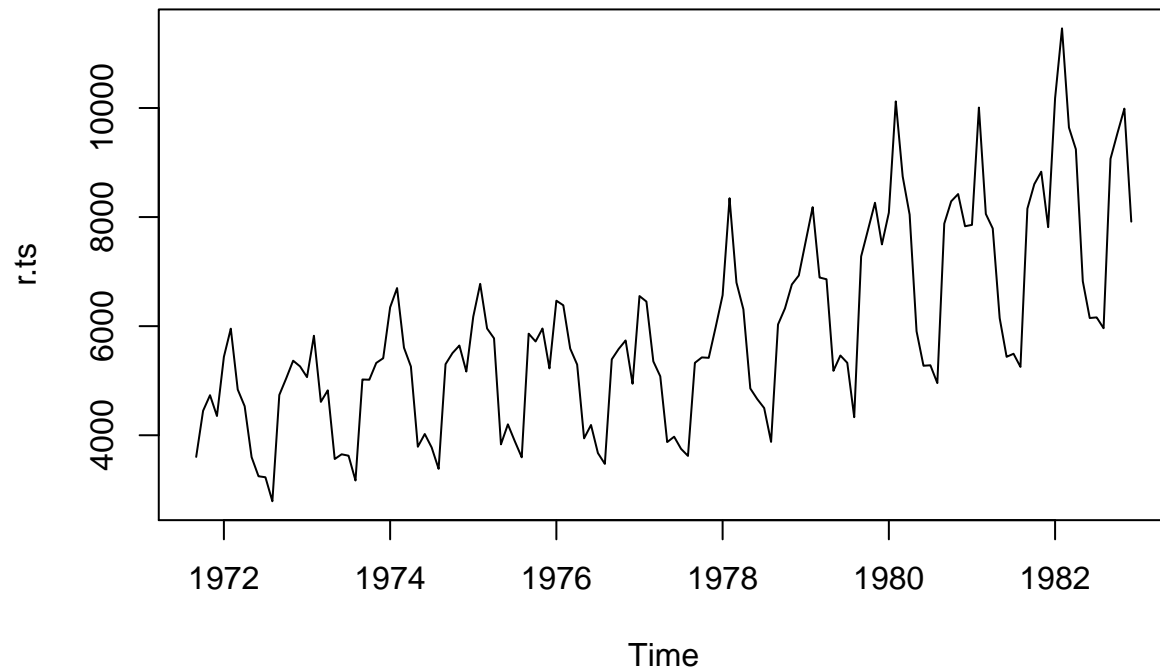
```
r.ts <- ts(riders$avgnumber, start = c(1971, 9), frequency = 12)
r.ts
```

##	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
## 1971									3603	4448	4734	4353
## 1972	5438	5954	4838	4532	3599	3248	3230	2790	4738	5043	5367	5262
## 1973	5065	5824	4612	4823	3564	3651	3625	3170	5022	5018	5326	5412
## 1974	6345	6698	5605	5260	3789	4023	3775	3384	5303	5506	5645	5168
## 1975	6173	6776	5957	5777	3833	4200	3891	3595	5862	5719	5957	5226
## 1976	6467	6383	5589	5296	3943	4188	3673	3475	5392	5583	5738	4944
## 1977	6552	6451	5354	5081	3874	3973	3751	3621	5328	5427	5419	5983
## 1978	6568	8346	6800	6308	4857	4663	4499	3878	6031	6325	6764	6929
## 1979	7557	8182	6892	6860	5181	5462	5328	4331	7279	7772	8262	7499
## 1980	8081	10121	8745	8045	5906	5274	5283	4956	7879	8289	8422	7832
## 1981	7857	10008	8057	7791	6147	5438	5494	5253	8154	8604	8832	7815
## 1982	10179	11460	9641	9243	6824	6149	6161	5962	9068	9538	9988	7916

b)

```
plot(r.ts, main = 'Busfahrgäste pro Monat in Iowa City')
```

Busfahrgäste pro Monat in Iowa City

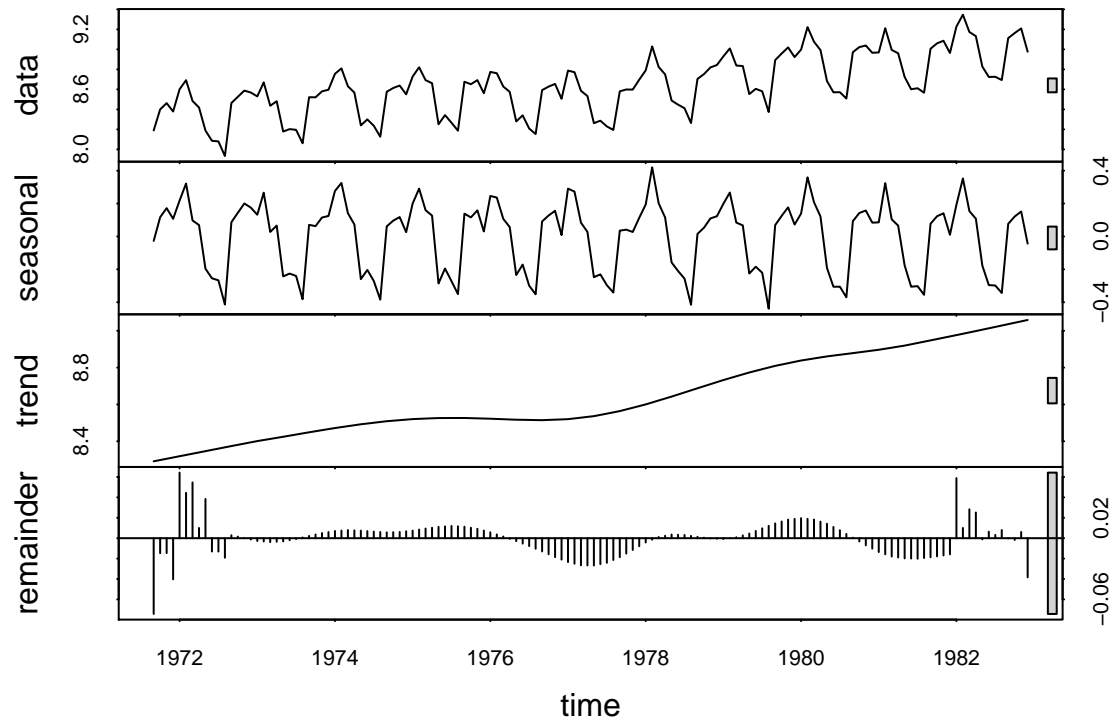


Die Reihe ist nicht stationär, weil sie sowohl einen Trend, als auch eine Saison-Komponente aufweist und ausserdem die Varianz im Verlauf zunimmt.

c)

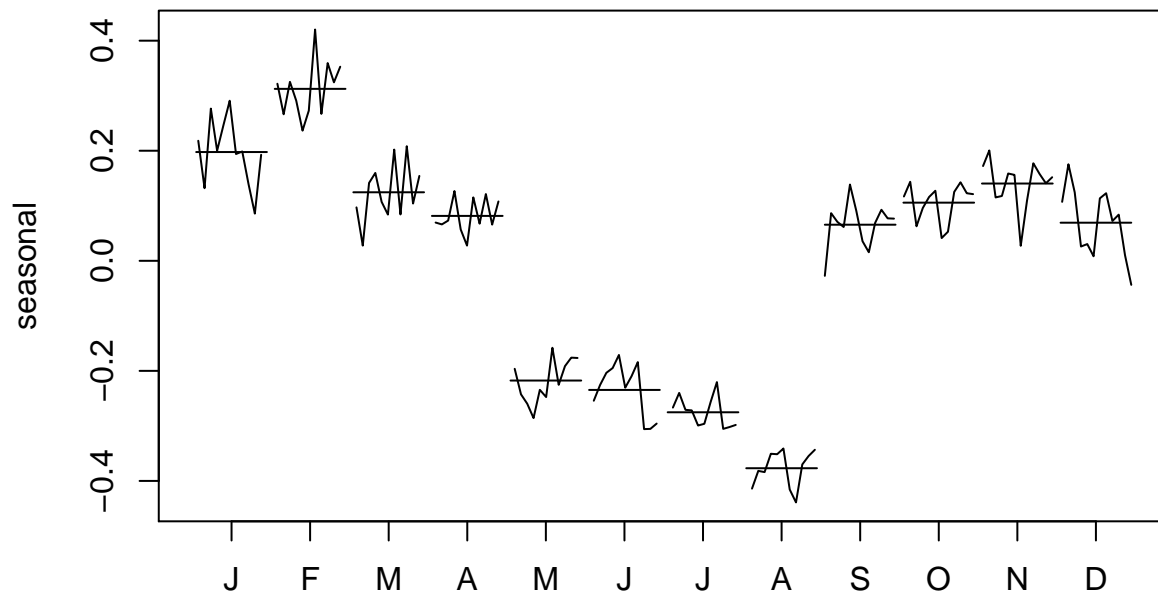
```
r.ts.log <- ts(log(riders$avgnumber), start = c(1971, 9), frequency = 12)
# s.window=3
r.stl.3 <- stl(r.ts.log, s.window = 3)
plot(r.stl.3, main = 'STL Zerlegung mit s.window=3')
```

STL Zerlegung mit s.window=3



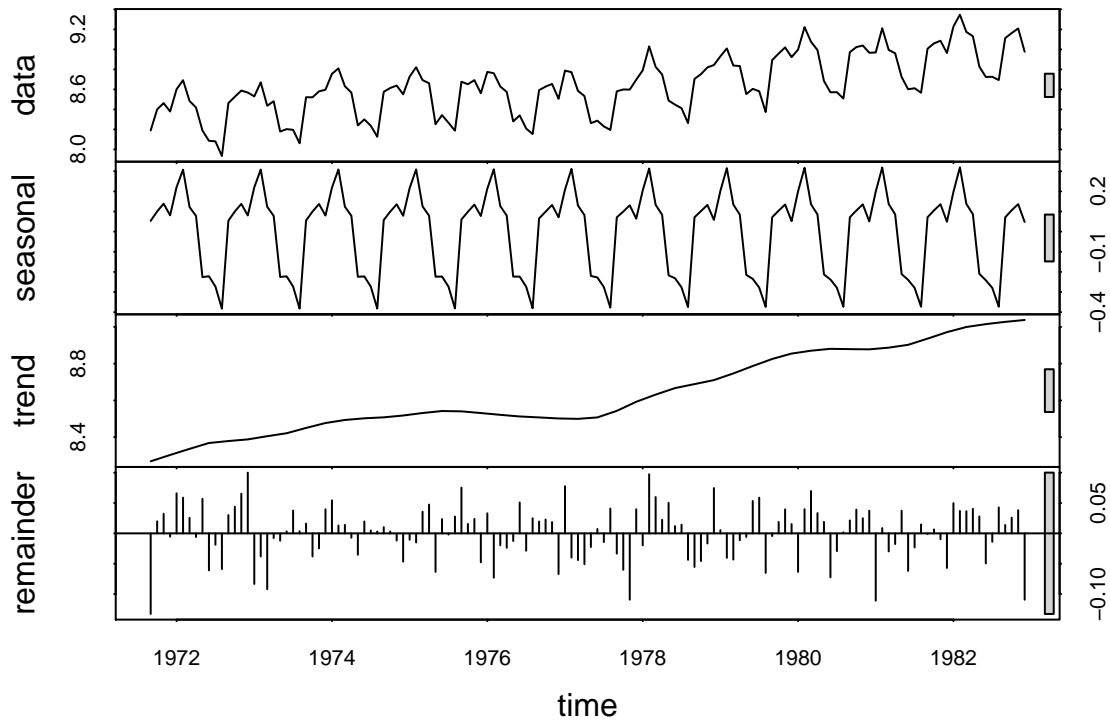
```
monthplot(r.stl.3, main = 'monthplot mit s.window=3')
```

monthplot mit s.window=3



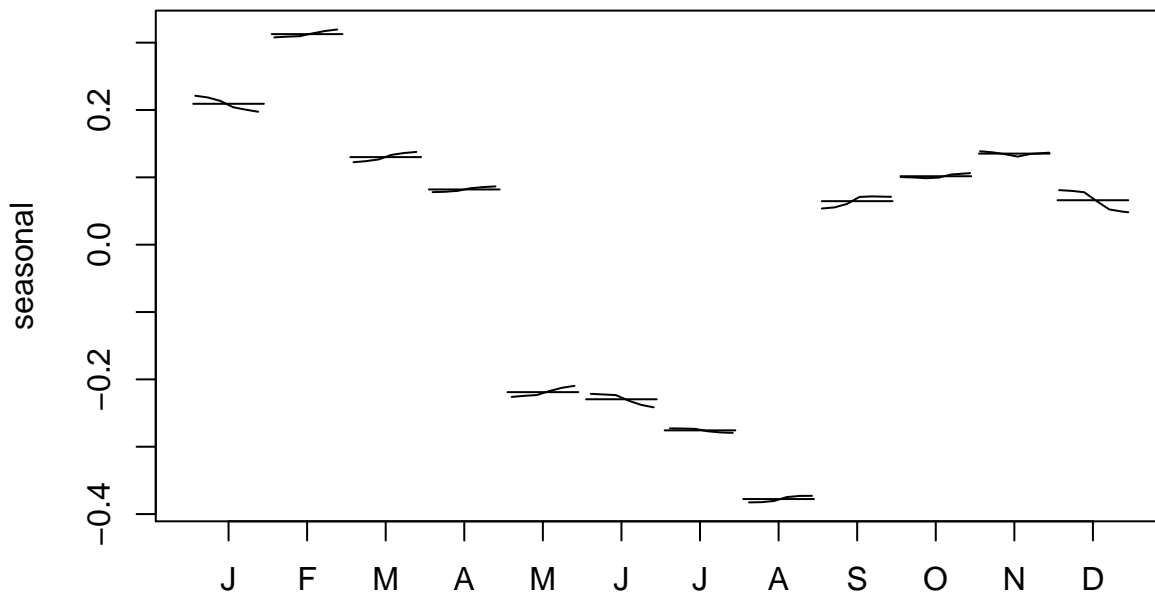
```
# s.window=15
r.stl.15 <- stl(r.ts.log, s.window = 15)
plot(r.stl.15, main = 'STL Zerlegung mit s.window=15')
```

STL Zerlegung mit s.window=15



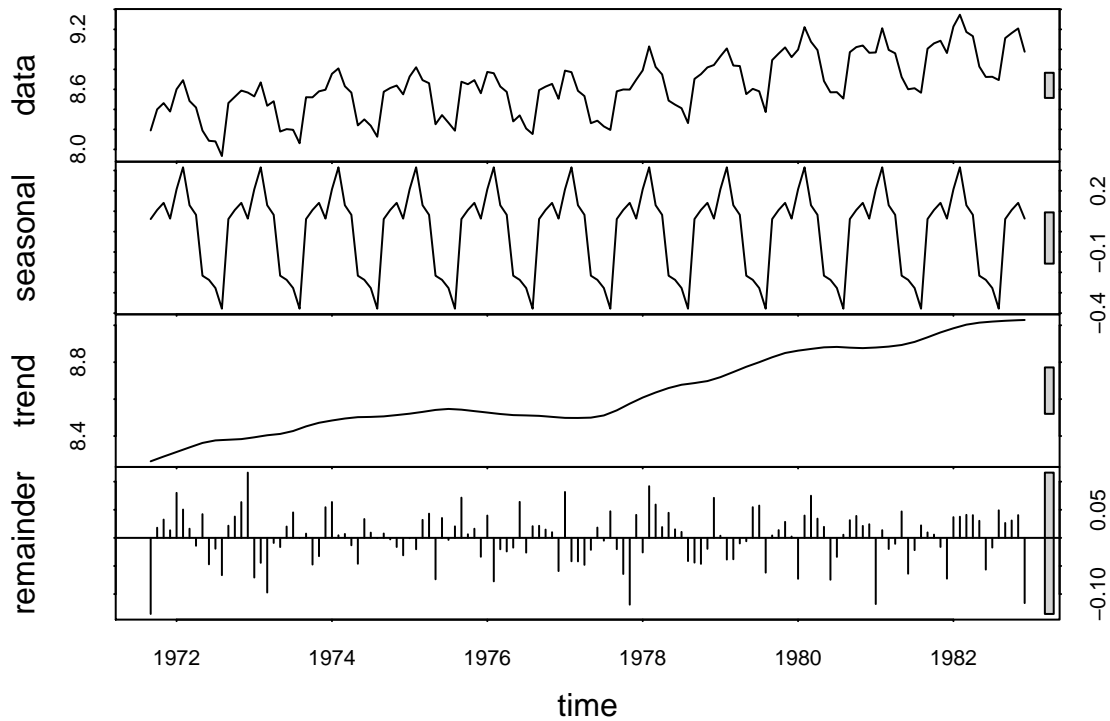
```
monthplot(r.stl.15, main = 'monthplot mit s.window=15')
```

monthplot mit s.window=15



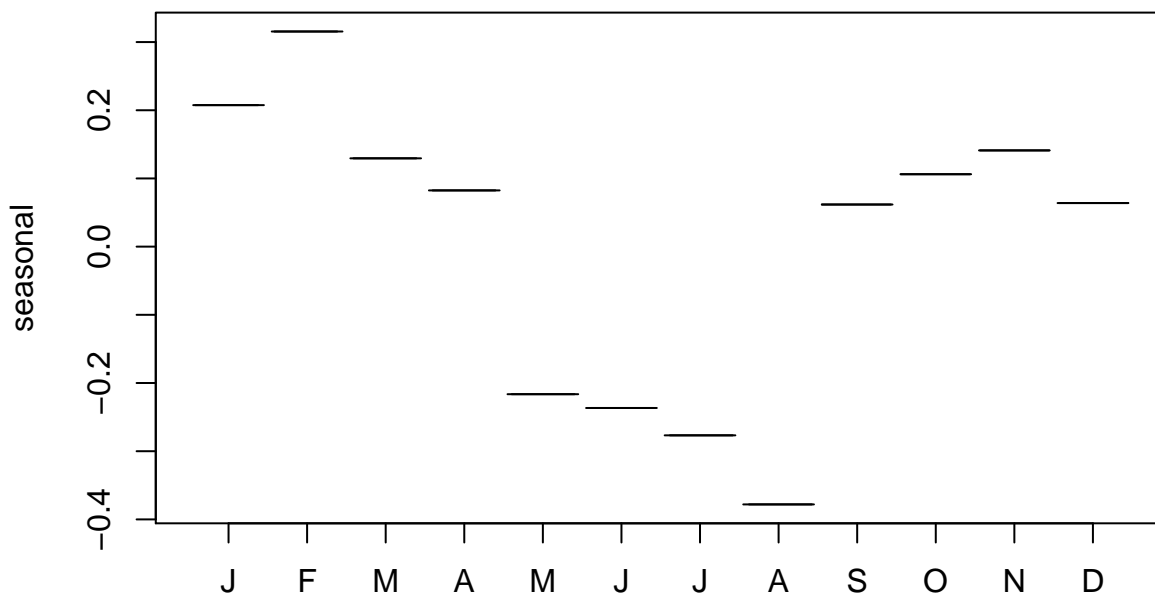
```
# s.window='periodic'
r.stl.p <- stl(r.ts.log, s.window = 'periodic')
plot(r.stl.p, main = 'STL Zerlegung mit s.window=periodic')
```

STL Zerlegung mit s.window=periodic



```
monthplot(r.stl.p, main = 'monthplot mit s.window=periodic')
```

monthplot mit s.window=periodic



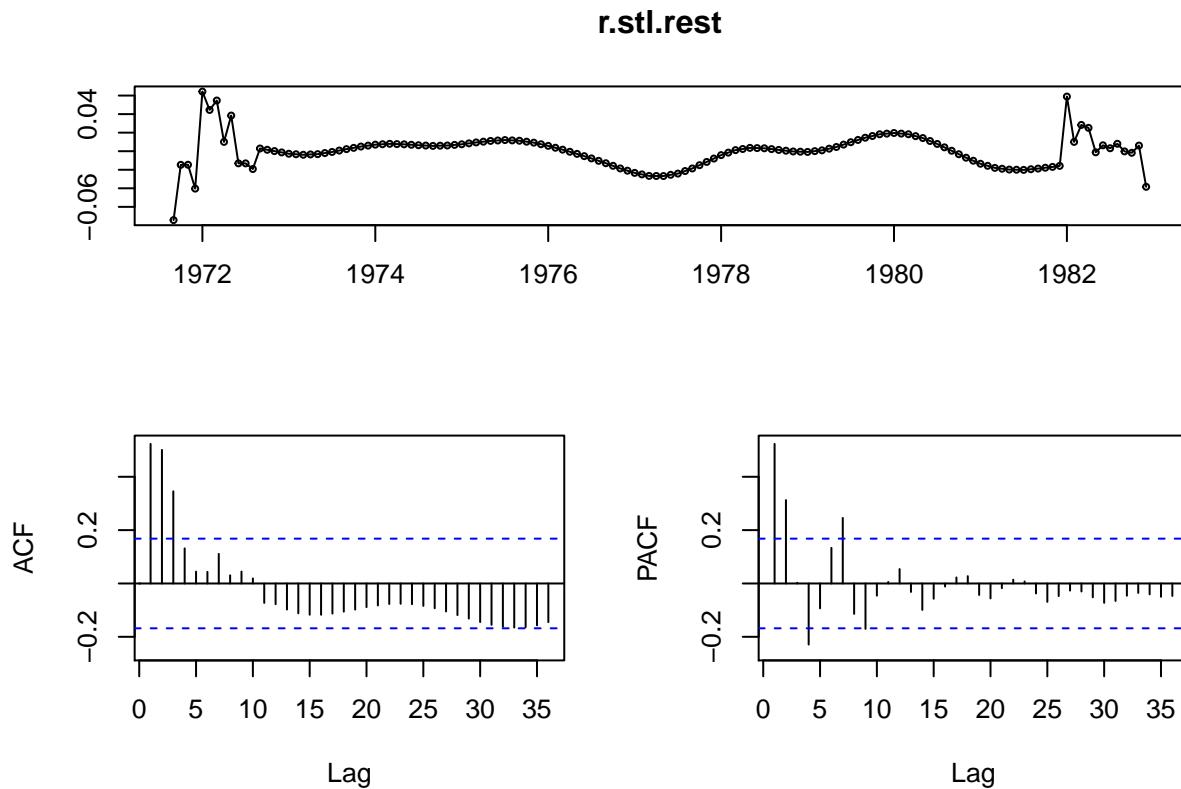
Die Plots für s.window=15 und 'periodic' sind sehr ähnlich, ich würde 'periodic' bevorzugen, weil der monthplot() die besseren Ergebnisse für die Glättung liefert.

d)

```
# Restterm (s.window=3)
r.stl.rest <- r.stl.3$time.series[, 3]
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method      from
## as.zoo.data.frame zoo

tsdisplay(r.stl.rest)
```



i)

Die ACF zeigt im mittleren Teil noch einen Trend.

Die PACF zeigt cut-offs bei Lag 2, 4, 7 und 9.

ii)

Visuell von der PACF her wären $p = 2, 4, 7$ und 9 Kandidaten für ein $AR(p)$ Modell.

iii)

```
(r.ar.burg <- ar.burg(r.stl.rest))

##
## Call:
## ar.burg.default(x = r.stl.rest)
##
```

```
## Coefficients:
##      1      2      3      4      5      6      7      8
## 0.3494 0.5557 0.1525 -0.3413 -0.3618 0.0649 0.4054 -0.0222
##      9
## -0.2060
##
## Order selected 9  sigma^2 estimated as 0.0001491
```

Es wird ein AR(9) Modell gewählt.

e)

```
(r.train <- window(r.ts.log, end = c(1981, 12)))
```

```
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1971
## 1972 8.601167 8.691819 8.484257 8.418919 8.188411 8.085795 8.080237 7.933797
## 1973 8.530109 8.669743 8.436417 8.481151 8.178639 8.202756 8.195610 8.061487
## 1974 8.755422 8.809564 8.631414 8.567886 8.239857 8.299783 8.236156 8.126814
## 1975 8.727940 8.821142 8.692322 8.661640 8.251403 8.342840 8.266421 8.187299
## 1976 8.774468 8.761393 8.628556 8.574707 8.279697 8.339979 8.208764 8.153350
## 1977 8.787526 8.771990 8.585599 8.533263 8.262043 8.287277 8.229778 8.194506
## 1978 8.789965 9.029538 8.824678 8.749574 8.488176 8.447414 8.411610 8.263075
## 1979 8.930230 9.009692 8.838117 8.833463 8.552753 8.605570 8.580731 8.373554
## 1980 8.997271 9.222368 9.076237 8.992806 8.683724 8.570544 8.572249 8.508354
## 1981 8.969160 9.211140 8.994297 8.960725 8.723719 8.601167 8.611412 8.566555
##      Sep      Oct      Nov      Dec
## 1971 8.189522 8.400210 8.462526 8.378621
## 1972 8.463370 8.525756 8.588024 8.568266
## 1973 8.521584 8.520787 8.580356 8.596374
## 1974 8.576028 8.613594 8.638525 8.550241
## 1975 8.676246 8.651549 8.692322 8.561401
## 1976 8.592672 8.627482 8.654866 8.505930
## 1977 8.580731 8.599142 8.597667 8.696677
## 1978 8.704668 8.752265 8.819370 8.843471
## 1979 8.892749 8.958283 9.019422 8.922525
## 1980 8.971956 9.022685 9.038603 8.965973
## 1981 9.006264 9.059982 9.086137 8.963800
```

```
(r.test <- window(r.ts.log, start = c(1982, 1)))
```

```
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1982 9.228082 9.346618 9.173780 9.131622 8.828201 8.724045 8.725994 8.693161
##      Sep      Oct      Nov      Dec
## 1982 9.112507 9.163039 9.209140 8.976641
```

f)

```
(r.fit <- ets(r.train))
```

```
## ETS(M,A,A)
##
## Call:
## ets(y = r.train)
```

```

##
## Smoothing parameters:
##   alpha = 0.4511
##   beta  = 1e-04
##   gamma = 1e-04
##
## Initial states:
##   l = 8.2626
##   b = 0.0058
##   s = -0.3872 -0.2778 -0.2319 -0.2181 0.0861 0.1322
##       0.3131 0.2061 0.0805 0.1422 0.099 0.0558
##
## sigma: 0.0067
##
##      AIC      AICc      BIC
## -94.42680 -88.65322 -46.48201
summary(r.fit)

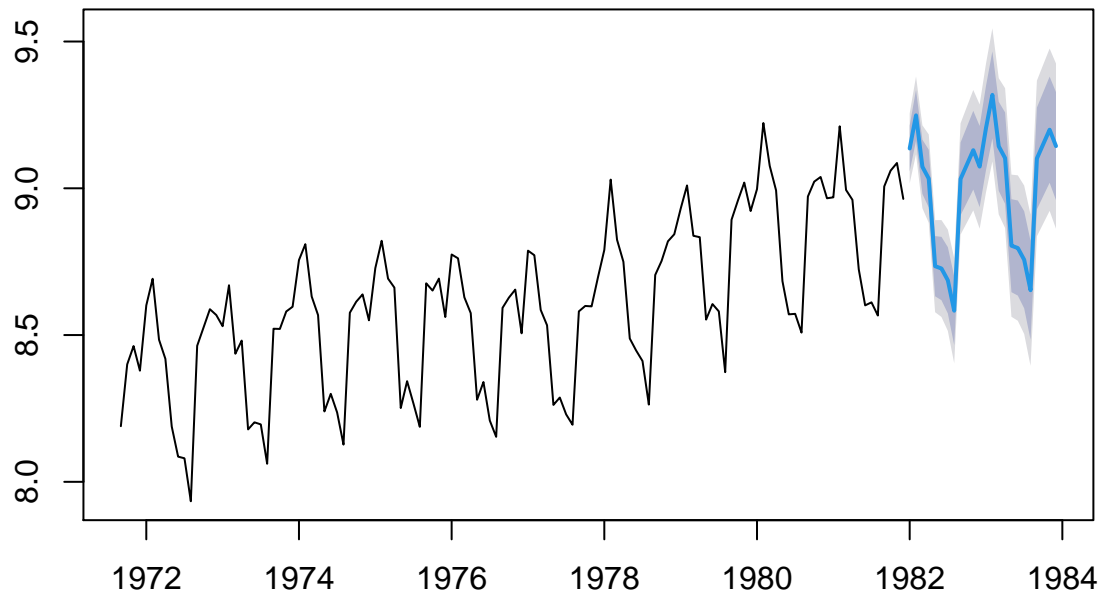
## ETS(M,A,A)
##
## Call:
## ets(y = r.train)
##
## Smoothing parameters:
##   alpha = 0.4511
##   beta  = 1e-04
##   gamma = 1e-04
##
## Initial states:
##   l = 8.2626
##   b = 0.0058
##   s = -0.3872 -0.2778 -0.2319 -0.2181 0.0861 0.1322
##       0.3131 0.2061 0.0805 0.1422 0.099 0.0558
##
## sigma: 0.0067
##
##      AIC      AICc      BIC
## -94.42680 -88.65322 -46.48201
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.001093531 0.05366313 0.04243885 -0.01488697 0.4933156 0.5129506
##              ACF1
## Training set -0.01112208

```

g)

```
plot(forecast(r.fit))
```


Forecasts from ETS(M,A,A)



h)

```
accuracy(forecast(r.fit, h = length(r.test)), r.test)
```

```
##               ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.001093531 0.05366313 0.04243885 -0.01488697 0.4933156 0.5129506
## Test set      0.064607325 0.08637169 0.08126590  0.71146520 0.8971843 0.9822460
##               ACF1 Theil's U
## Training set -0.01112208      NA
## Test set      0.04658647 0.4524071
```

i)

j)

Nein, denn die geschätzten Koeffizienten $\beta_1, \dots, \beta_{12}$ sind erwartungstreu.

k)

Nein.

l)

Nein.