R package bsts

Bayesian structural time series

Uses Markov Chain Monte Carlo to sample from the posterior distribution of a Bayesian structural time series model.

Author: Steven L. Scott - stevescott@google.com

References:

Harvey (1990) Forecasting, structural time series, and the Kalman filter

Durbin and Koopman (2001) Time series analysis by state space methods

Scott and Varian (2013) Predicting the Present with Bayesian Structural Time Series
http://people.ischool.berkeley.edu/~hal/Papers/2013/pred-present-with-bsts.pdf

Usage

```
bsts(formula,
     state.specification,
     family = c("gaussian", "logit", "poisson", "student"),
     save.state.contributions = TRUE,
     save.prediction.errors = TRUE,
     data,
     bma.method = c("SSVS", "ODA"),
     prior,
     oda.options = list(
         fallback.probability = 0.0,
         eigenvalue.fudge.factor = 0.01),
     contrasts = NULL,
     na.action = na.pass,
     niter,
     ping = niter / 10,
     timeout.seconds = Inf,
     seed = NULL,
     ...)
```

add.local.linear.trend

Add a local linear trend model to a state specification:

$$\begin{aligned} \mu_t &= \mu_{t-1} + \delta_{t-1} + \omega_{1t} & \omega_{1t} &\sim N(0, \sigma^2_{level}) \\ \delta_t &= \delta_{t-1} + \omega_{2t} & \omega_{2t} &\sim N(0, \sigma^2_{slope}) \end{aligned}$$

The prior distribution is on σ_{level} and σ_{slope}.

```
AddLocalLinearTrend(
    state.specification = NULL,
    y,
    level.sigma.prior = NULL,
    slope.sigma.prior = NULL,
    initial.level.prior = NULL,
    initial.slope.prior = NULL,
    sdy,
    initial.y)
```

add.seasonal

Add a seasonal model to a state specification.
The seasonal model can be thought of as a regression on `nseasons` dummy variables with coefficients constrained to sum to 1 (in expectation). If there are S seasons then the state vector gamma is of dimension $S - 1$:

$$\begin{aligned} \gamma_{1t} &= -\sum_{s=1}^{S-1} \gamma_{s,t-1} + \omega_{3t} \qquad \omega_{3t} \sim N(0, \sigma^2_{seas}) \\ \gamma_{st} &= \gamma_{s-1,t-1} \qquad s = 2, \ldots, S-1. \end{aligned}$$

```
AddSeasonal(
    state.specification,
    y,
    nseasons,
    season.duration = 1,
    sigma.prior,
    initial.state.prior,
    sdy)
```

add.trig

Add a trigonometric seasonal model to a state specification:

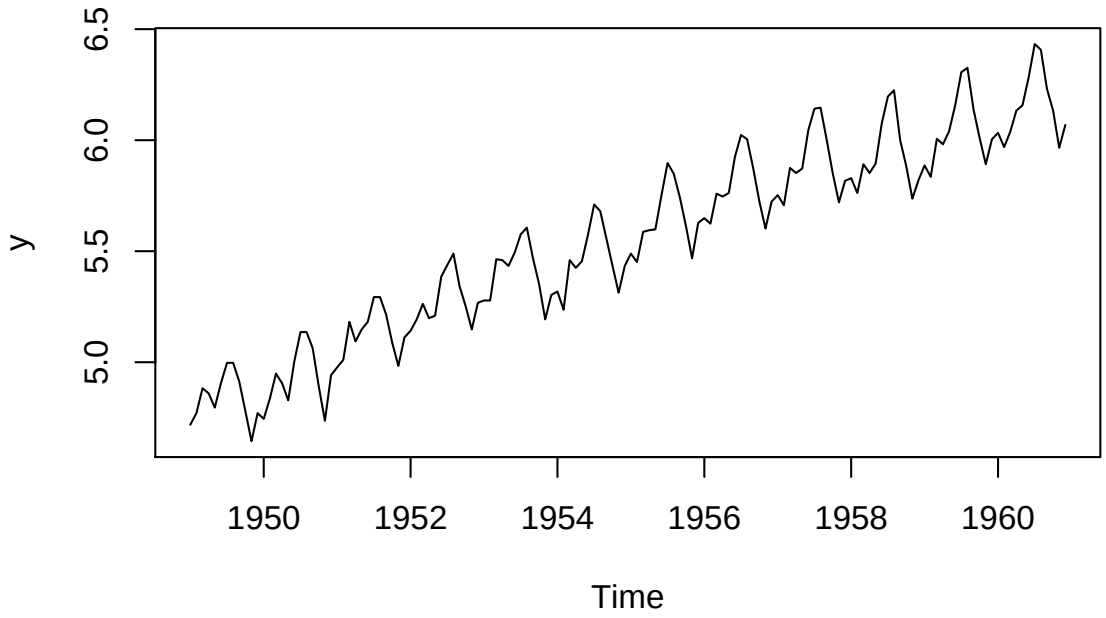
$$\gamma_t = \sum \alpha_{it} \sin(f_i t) + \beta_{it} \cos(f_i t)$$

The evolution equation are:

$$\begin{aligned} \alpha_{it} &= \alpha_{i,t-1} + \xi_{it} & \xi_{it} &\sim N(0, \sigma^2_{a,i}) \\ \beta_{it} &= \beta_{i,t-1} + \psi_{it} & \psi_{it} &\sim N(0, \sigma^2_{b,i}) \end{aligned}$$

```
AddTrig(
    state.specification = NULL,
    y,
    period,
    frequencies,
    sigma.prior = NULL,
    initial.state.prior = NULL,
    sdy)
```

Airline data



$$
\begin{aligned}
y_t &= \mu_t + \gamma_{1t} + v_t & v_t &\sim N(0, \sigma^2_{obs}) \\
\mu_t &= \mu_{t-1} + \delta_{t-1} + \omega_{1t} & \omega_{1t} &\sim N(0, \sigma^2_{level}) \\
\delta_t &= \delta_{t-1} + \omega_{2t} & \omega_{2t} &\sim N(0, \sigma^2_{slope}) \\
\gamma_{1t} &= -\sum_{s=1}^{S-1} \gamma_{s,t-1} + \omega_{3t} & \omega_{3t} &\sim N(0, \sigma^2_{seas}) \\
\gamma_{st} &= \gamma_{s-1,t-1} & s &= 2, \ldots, S-1.
\end{aligned}
$$

```
install.packages("bsts")
library(bsts)

data(AirPassengers)
y = log(AirPassengers)

ss = AddLocalLinearTrend(list(), y)
ss = AddSeasonal(ss, y, nseasons = 12)
model = bsts(y, state.specification = ss, niter = 10000)

=-=-=-=-= Iteration 0 Sat May  7 14:45:34 2016 =-=-=-=-=
=-=-=-=-= Iteration 1000 Sat May  7 14:45:36 2016 =-=-=-=-=
=-=-=-=-= Iteration 8000 Sat May  7 14:45:52 2016 =-=-=-=-=
=-=-=-=-= Iteration 9000 Sat May  7 14:45:55 2016 =-=-=-=-=

names(model)

 [1] "sigma.obs"                 "sigma.trend.level"
 [3] "sigma.trend.slope"         "sigma.seasonal.12"
 [5] "final.state"               "state.contributions"
 [7] "one.step.prediction.errors" "log.likelihood"
 [9] "has.regression"            "state.specification"
[11] "family"                    "niter"
[13] "original.series"

dim(model$state.contributions)

[1] 10000     2   144
```

Posterior of $(\sigma_{obs}, \sigma_{level}, \sigma_{slope}, \sigma_{seas})$

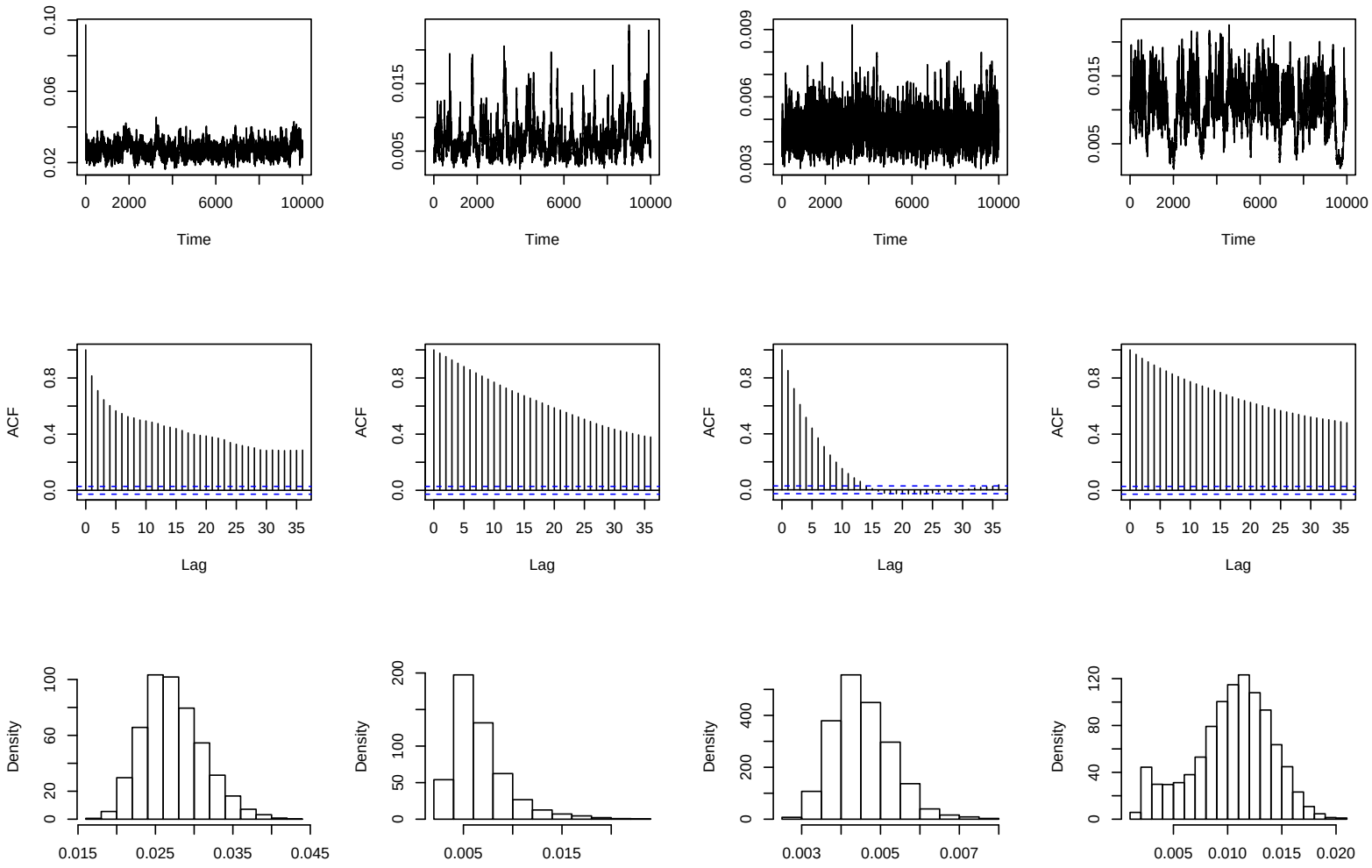
```
sigmas = cbind(model$sigma.obs,
model$sigma.trend.level,
model$sigma.trend.slope,
model$sigma.seasonal.12)

round(cbind(apply(sigmas[5001:10000,],2,mean),
apply(sigmas[5001:10000,],2,median),
apply(sigmas[5001:10000,],2,quantile,0.025),
apply(sigmas[5001:10000,],2,quantile,0.975)),4)

       [,1]   [,2]   [,3]   [,4]
[1,] 0.0269 0.0267 0.0201 0.0346
[2,] 0.0061 0.0054 0.0031 0.0139
[3,] 0.0045 0.0044 0.0033 0.0061
[4,] 0.0107 0.0107 0.0030 0.0176

pdf(file="bsts-graph1.pdf",width=10,height=7)
par(mfrow=c(3,4))
for (i in 1:4)
  ts.plot(sigmas[,i],ylab="")
for (i in 1:4)
  acf(sigmas[5001:10000,i],main="")
for (i in 1:4)
  hist(sigmas[5001:10000,i],xlab="",main="",prob=TRUE)
dev.off()
```

$$(\sigma_{obs}, \sigma_{level}, \sigma_{slope}, \sigma_{seas})$$

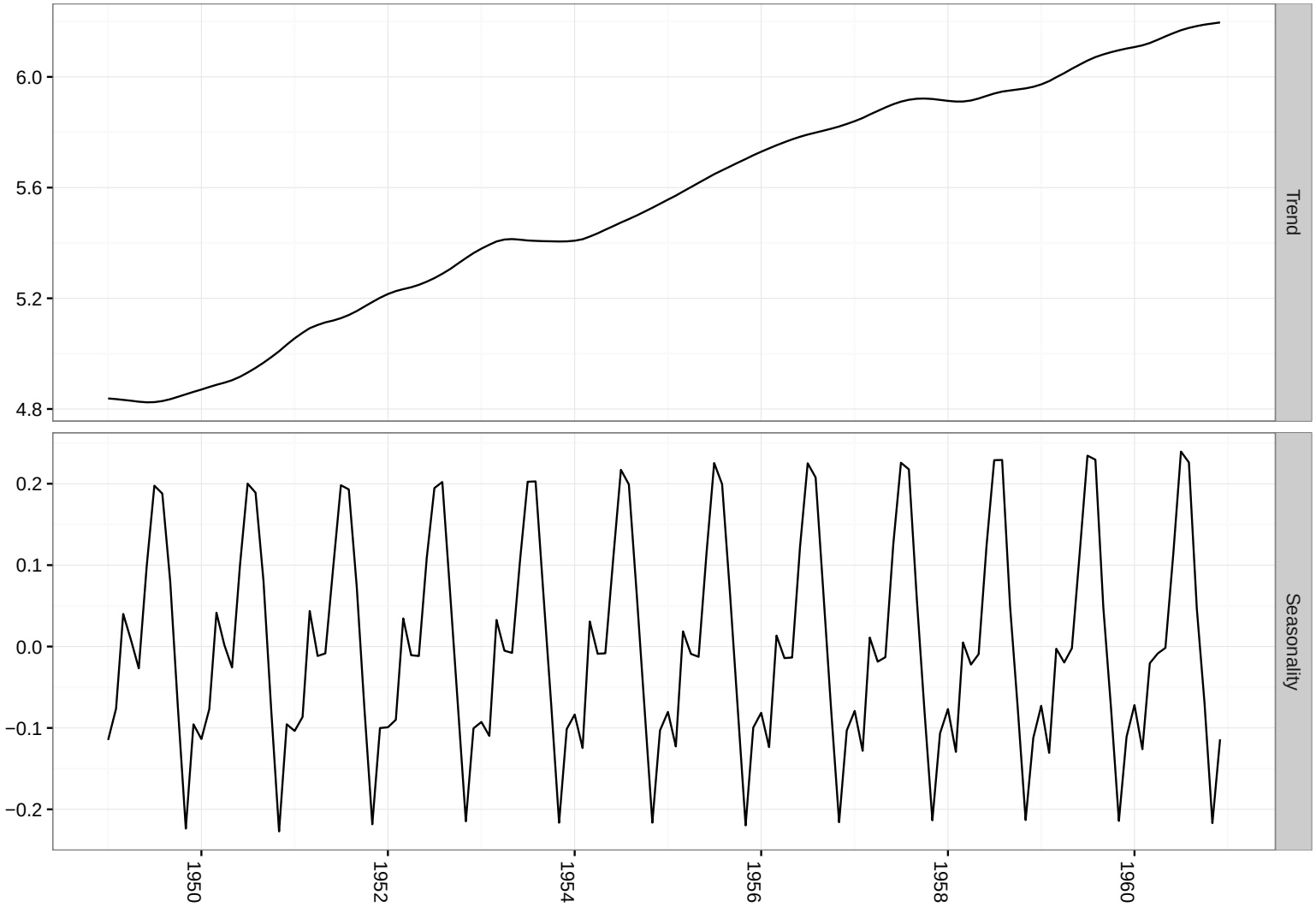


Posterior of σs

```
components = cbind.data.frame(
  colMeans(model$state.contributions[-(1:5000),"trend",]),
  colMeans(model$state.contributions[-(1:5000),"seasonal.12.1",]),
  as.Date(time(y)))
names(components) = c("Trend", "Seasonality", "Date")
components = melt(components, id="Date")
names(components) = c("Date", "Component", "Value")

pdf(file="bsts-graph2.pdf",width=10,height=7)
ggplot(data=components, aes(x=Date, y=Value)) + geom_line() +
  theme_bw() + theme(legend.title = element_blank()) + ylab("") + xlab("") +
  facet_grid(Component ~ ., scales="free") + guides(colour=FALSE) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0))
dev.off()
```

Free-form seasonality



Trigonometric seasonality