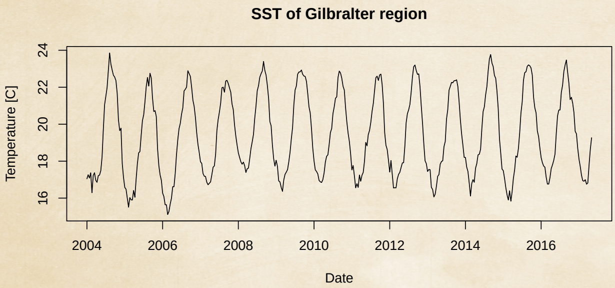
Implementing BSTS Forecasting

Andrew Bates, Josh Gloyd, Tyler Tucker

Time Series Visual

```r
gilbralter <- read_csv("data/gilbralter.csv")
gilT <- ts(gilbralter$tempMean, start=c(2004,1,13),
           end=c(2017, 11, 25), frequency=30)
plot(gilT, main='SST of Gilbralter region',
     xlab='Date',
     ylab='Temperature [C]')
```



Set up BSTS model?

These three lines of code are setting up the model as

$$y_t = \mu_t + \tau_t + \epsilon_t$$

where μ_t is a local linear component, τ_t is a seasonal component, and ϵ_t is a white noise process centered about zero and with standard deviation σ^2.

```
nseasons = 30
ss <- list()
ss <- AddLocalLinearTrend(ss, gilT)
ss <- AddSeasonal(ss, gilT, nseasons=nseasons)
```

Run the model

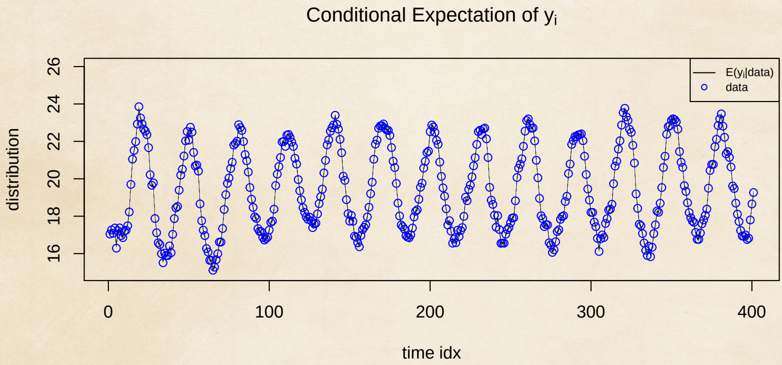
Implement the bsts function with time series, state specification object, and number of iterations runs the MCMC and fits the parameters.

```
modelGilT <- bsts(gilT,
                  state.specification=ss,
                  niter = 1000)
```

Interpret

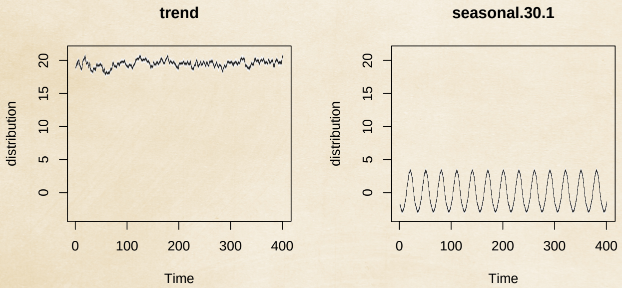
```r
plot(modelGilT,
    main=TeX('Conditional Expectation of $y_i$'),
    xlab='time idx', ylab='distribution',
    ylim=c(15, 26))
legend("topright",
       legend=c(TeX('$E(y_i|data)$'), "data"),
       col=c("black", "blue"),
       lty=c(1,NA), pch=c(NA, 1), cex=.7)
```

Interpret


Conditional Expectation of y_i
E(y_i|data)
data
distribution
16
18
20
22
24
26
time idx
0
100
200
300
400

Trend and Season plots are available too

```r
plot(modelGilT, "components")
```



Forecasting

From here we can perform forecasting 12 time steps in the future (144 days) with the following code. Green dotted line is the 97.5 % prediction interval, the blue line is the 2.5 % PI

```
pred1 <- predict(modelGilT, horizon = 12)
plot(pred1, plot.original = 156)
```