SPDE one dimensional example

Elias T. Krainski and Håvard Rue

Created: September 26th 2016 - Last update: August 9th 2017

In this example we show row to analyse a time series of daily temperature using a one dimension SPDE model. More details about it are on the paber at <https://www.jstatsoft.org/article/view/v063i19>

1 The data

We consider the daily weather data available at <http://www.yr.no/>. We have the following set the URL for the daily data for Trondheim considerint in the last 13 months

```
u0 <- paste0("http://www.yr.no/place/Norway/S%C3%B8r-Tr%C3%B8ndelag/",
             "Trondheim/Trondheim/detailed_statistics.html")
### browseURL(u0) ### to visit the web page
```

One can read and extract the desired data table (the second one at the URL) using the **XML** package with the `readHTMLTable()` function. However, it still need some pre-processing.

We do not consider it and just read the web page as a text and play with the text and its structure directly.

```
d0 <- readLines(u0) ### read it as text (Done at 26 September 2016)
```

First, we have to find the index for each table line and consider only those for the main table:

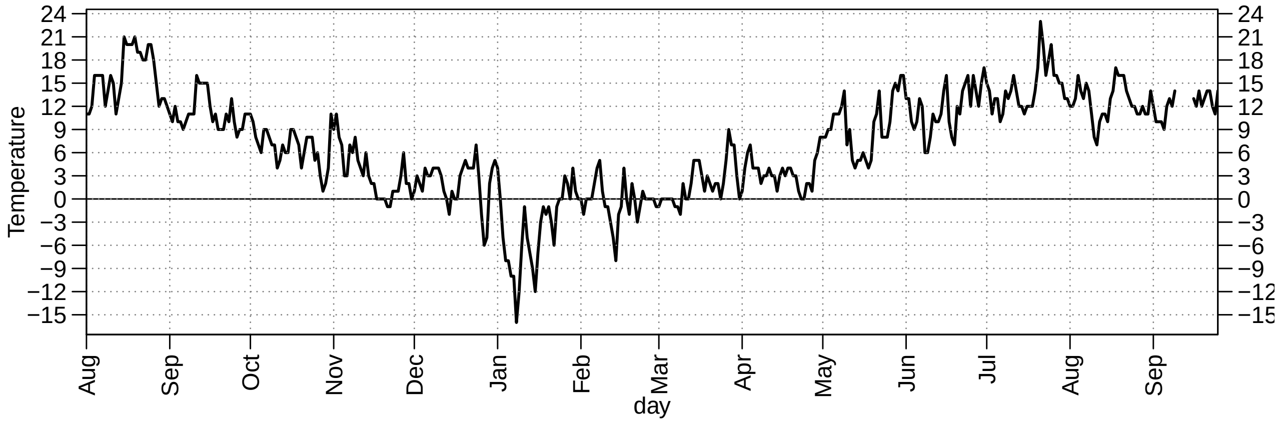
```
i <- grep("<tr>", d0) ### index for each table line
i <- i[i>grep("<tbody>", d0)[2]] ### select those for the second table
```

The desired data we would like to analyse is the minimum and maximum temperature. Commands to extract and pre-process these data

```
if (Sys.getlocale("LC_TIME")!="C")
    Sys.setlocale("LC_TIME", "C")

## [1] "C"

dates <- as.Date(d0[i+1], format="      <th>%b %d, %Y</th>")
tmed <- as.numeric(gsub("<td>", "", gsub("..</td>", "", d0[i+4])))
(n <- length(dates)) ### it is daily over last 13 months

## [1] 422
```

Visualize it with the following commands

```
pd <- pretty(c(dates, max(dates+30)), n=13)
par(mfrow=c(1,1), mar=c(3,3,0.5,2), mgp=c(2,.7,0), las=2, xaxs="i")
plot(dates, tmed, type="l", lwd=2,
     axes=FALSE, xlab="day", ylab="Temperature")
abline(h=0)
abline(h=3*(-8:9), v=pd, lty=3, col=gray(.5))
box()
axis(2, 3*(-8:9)); axis(4, 3*(-8:9))
axis(1, pd, months(pd, TRUE))
```



2 Model fitting

Mesh in 1d it is a matter of chosing a set of knots, the order of the basis functions and the boundary. Choosing first order basis function and Neumann boundary.

```
coo <- as.numeric(dates-min(dates)) ## have numeric temporal coordinates
mesh <- inla.mesh.1d(loc=seq(min(coo), max(coo), by=7), ## knots (7 days)
                  boundary=c("neumann", "neumann"), ## boundaries
                  degree=2)  ### basis function degree
```

Projector matrix is the matrix built to project the process at the mesh nodes to the locations

```
A <- inla.spde.make.A( ## projector creator
    mesh=mesh, ## provide the mesh
    loc=coo) ### locations where to project the field
dim(A) ## an "n" by "m" projector matrix

## [1] 422  60

summary(rowSums(A)) ### each line sums up to one

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1

summary(colSums(A)) ### "how many" observations per knot

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.000   7.000   7.000   7.033   7.000   8.500
```

Build the SPDE model on the mesh, choosing the smoothness (α), to define the precision structure, and priors for the hyperparameters. We just set $\alpha = 2$ and keep the default priors.

```
spde <- inla.spde2.pcmatern( ## model for the precision
    mesh=mesh, ## mesh supplied
    alpha=2, ## smoothness parameter
    prior.range = c(1, 0.01), ## P(range < 1) = 0.01
    prior.sigma = c(1, 0.5)) ## P(sigma > 1) = 0.5
```

Create a data stack in order to organize the data. This is a way to allow models with complex linear predictors. In our case, we have a SPDE model defined on m nodes. It must be combined with the covariate (and the intercept) effect at n locations. We do it using different projector matrices.

```
stk.e <- inla.stack( ## stack creator
  data=list(y=tmed),  ## response
  effects=list(## two elements:
    data.frame(b0=rep(1, n)), ## regressor part
    i=1:spde$n.spde),  ## RF index
  A=list(## projector list of each effect
    1, ## for the covariates
    A), ## for the RF
  tag="est") ## tag
```

Fit the posterior marginal distributions for all model parameters, syplying the model formula, data and some additional controls to the main function in the **INLA** package

```
formula <- y ~ 0 + b0 + ## fixed part
  f(i, model=spde) ## RF term
res <- inla( ## main function in INLA package
  formula, ## model formula
  data=inla.stack.data(stk.e), ## dataset
  control.predictor=list( ## inform projector needed in SPDE models
    A = inla.stack.A(stk.e), compute=TRUE)) ## projector from the stack data
```

3 Posterior marginal distributions - PMDs

Summary of the regression coefficients PMDs

```
round(res$summary.fixed, 4)

##      mean     sd 0.025quant 0.5quant 0.975quant   mode kld
## b0 7.0948 1.6702      3.762   7.0962    10.4151 7.0986   0
```

The PMDs summary for the Gaussian likelihood precision and the two RF parameters

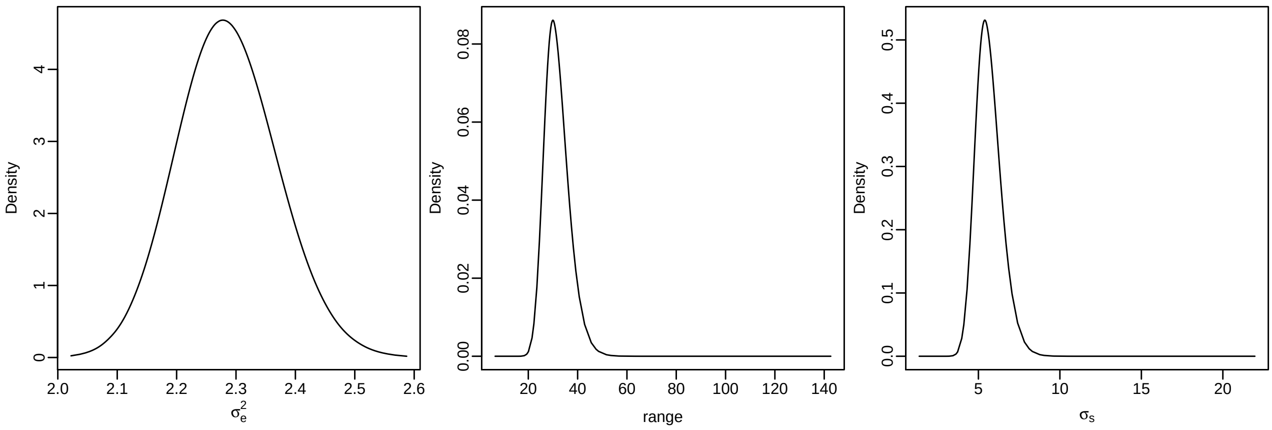
```
round(res$summary.hyperpar, 4)

##                                            mean     sd 0.025quant 0.5quant
## Precision for the Gaussian observations  0.1923 0.0144     0.1653   0.1918
## Range for i                             31.6933 4.9518    23.4692  31.1577
## Stdev for i                              5.6431 0.7921     4.3040   5.5658
##                                         0.975quant    mode
## Precision for the Gaussian observations     0.2219  0.1911
## Range for i                                42.8416 29.9846
## Stdev for i                                 7.4071  5.3939
```

We have to transform the likelihood precision PMD to have the variance PMD. It can be done by

```
m.prec <- res$marginals.hyperpar$"Precision for the Gaussian observations" ## the marginal
post.s2e <- inla.tmarginal(## function to compute a tranformation
  function(x) sqrt(1/x), ## inverse transformation and square root
  m.prec) ## marginal to be applied
```

The PMDs for likelihood standard deviation and the RF parameters can be visualized by

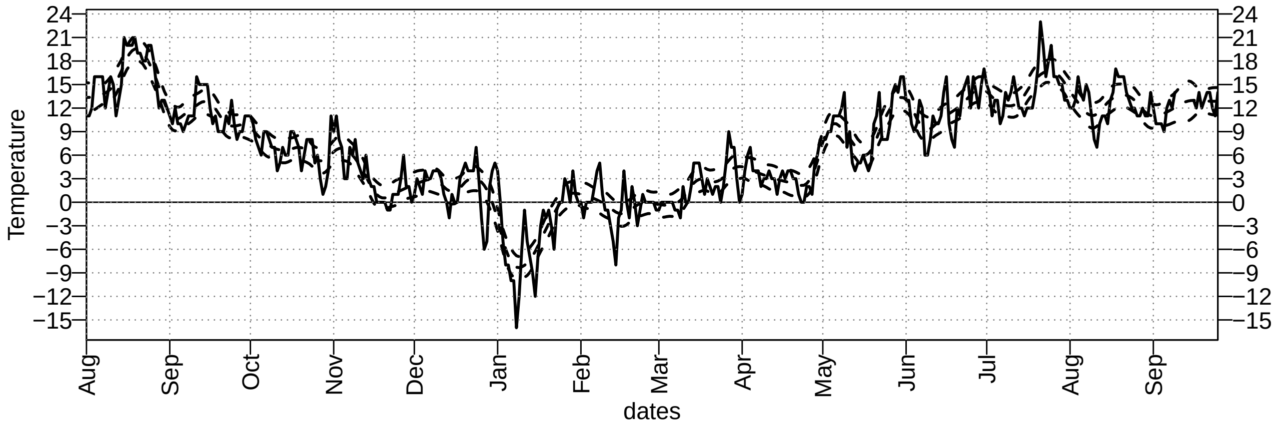
```
par(mfrow=c(1,3), mar=c(3,3,0.3,0.3), mgp=c(2,0.5,0))
plot(post.s2e, type="l", ylab="Density",
     xlab=expression(sigma[e]^2))
plot(res$marginals.hyperpar[[2]], type="l",
     xlab="range", ylab="Density")
plot(res$marginals.hyperpar[[3]], ty="l",
     xlab=expression(sigma[s]), yla="Density")
```



4 Predicted

Visualize it with the commands bellow

```
par(mfrow=c(1,1), mar=c(3,3,0.3,2), mgp=c(2,0.5,0), las=2, xaxs="i")
id <- inla.stack.index(stk.e, tag="est")$data
plot(dates, tmed, type="l", axes=FALSE, ylab="Temperature", lwd=2)
for (j in 3:5)
  lines(dates, res$summary.fitted.values[id, j], lty=2, lwd=2)
box(); axis(2, 3*(-8:9)); axis(4, 3*(-8:9))
axis(1, pd, months(pd, T))
abline(h=0)
abline(h=3*(-8:9), v=pd, lty=3, col=gray(.5))
```



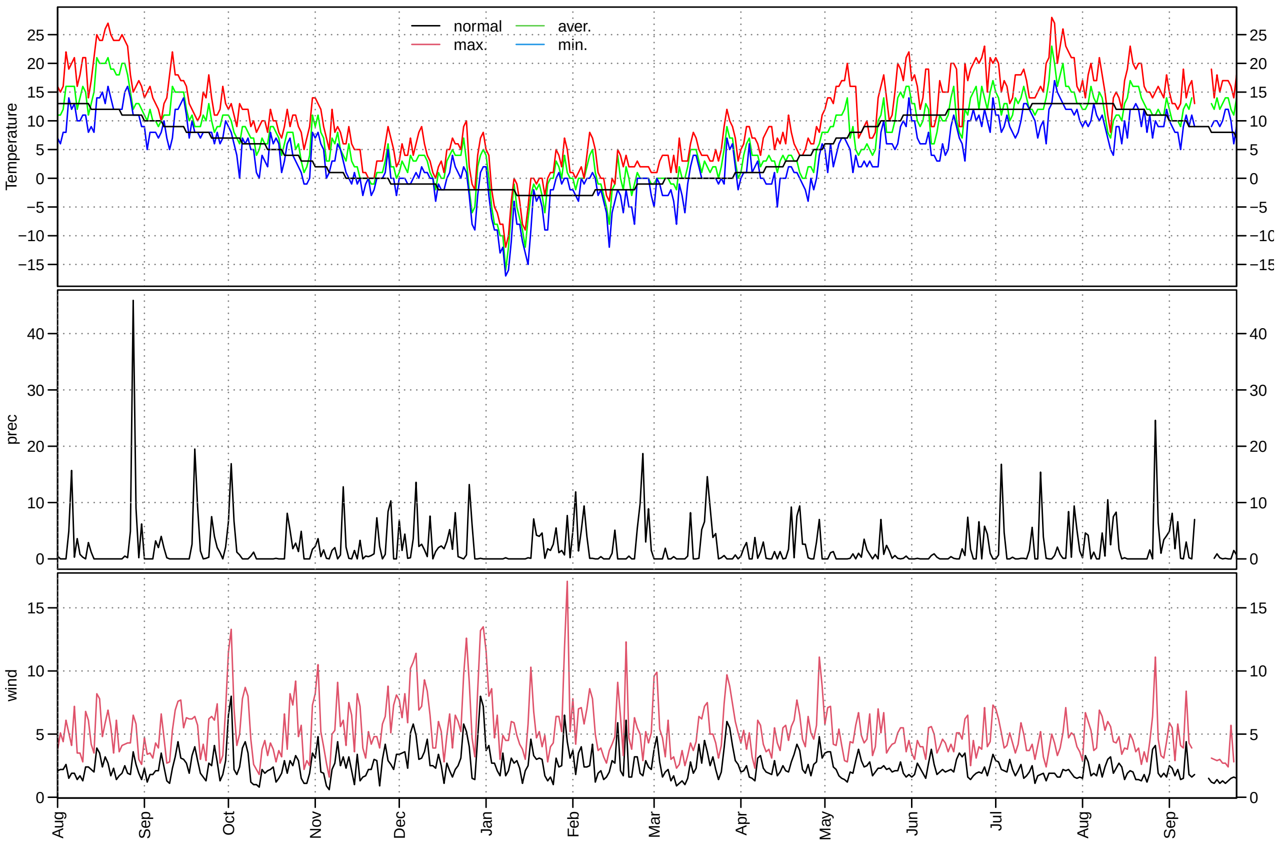
5 Just a look to the rest of the data

Pre-processing the maximum, minimum and normal temperature, the precipitation, and the average and maximum wind:

```
tmax <- as.numeric(gsub("<td>", "", gsub("..</td>", "", d0[i+2])))
tmin <- as.numeric(gsub("<td>", "", gsub("..</td>", "", d0[i+3])))
tnormal <- as.numeric(gsub("<td>", "", gsub("..</td>", "", d0[i+5])))
prec <- as.numeric(gsub("<td>", "", gsub("mm</td>", "", d0[i+6])))
wind <- as.numeric(gsub("<td>", "", gsub("m/s</td>", "", d0[i+10])))
wmax <- as.numeric(gsub("<td>", "", gsub("m/s</td>", "", d0[i+9])))
```

Visualize it

```
par(mfrow=c(3,1), mar=c(0.1,3,0.1,2), mgp=c(2,.7,0), las=2, xaxs="i")
plot(dates, tmed, type="l", ylim=range(tmin, tmax, na.rm=TRUE),
     axes=FALSE, xlab="", ylab="Temperature", col="green")
lines(dates, tmin, col="blue")
lines(dates, tmax, col="red")
lines(dates, tnormal)
legend(dates[which.min(tmin)], par()$usr[4], c("normal", "max.", "aver.", "min."),
      col=1:4, lty=1, ncol=2, xjust=0.5, bty="n")
abline(h=5*(-5:6), v=pd, lty=3, col=gray(.5))
box(); axis(2, 5*(-5:6)); axis(4, 5*(-5:6))

plot(dates, prec, type="l", axes=FALSE, xlab="")
box(); axis(2); axis(4)
abline(v=pd, h=10*(1:4), lty=3, col=gray(0.5))

par(mar=c(3, 3, 0.1, 2), new=FALSE)
plot(dates, wind, type="l", axes=FALSE, xlab="",
     ylim=range(wind, wmax, na.rm=TRUE))
lines(dates, wmax, col=2)
box(); axis(2); axis(4)
abline(v=pd, h=5*(1:3), lty=3, col=gray(0.5))
axis(1, pd, months(pd, TRUE))
```



We can have a look at the difference between the daily mean temperature and the normal temperature. The current normal is the average over the period from 1961 to 1990.

```
par(mar=c(3, 3, 0.1, 2), mgp=c(2,0.7,0), las=2, xaxs="i")
plot(dates, tmed-tnormal, type="l", axes=FALSE,
     xlab="", ylab="Deviation from normal temperature")
box(); axis(2); axis(4)
abline(h=5*(-2:2), v=pd, lty=2, col=gray(0.5))
axis(1, pd, months(pd, TRUE))
```