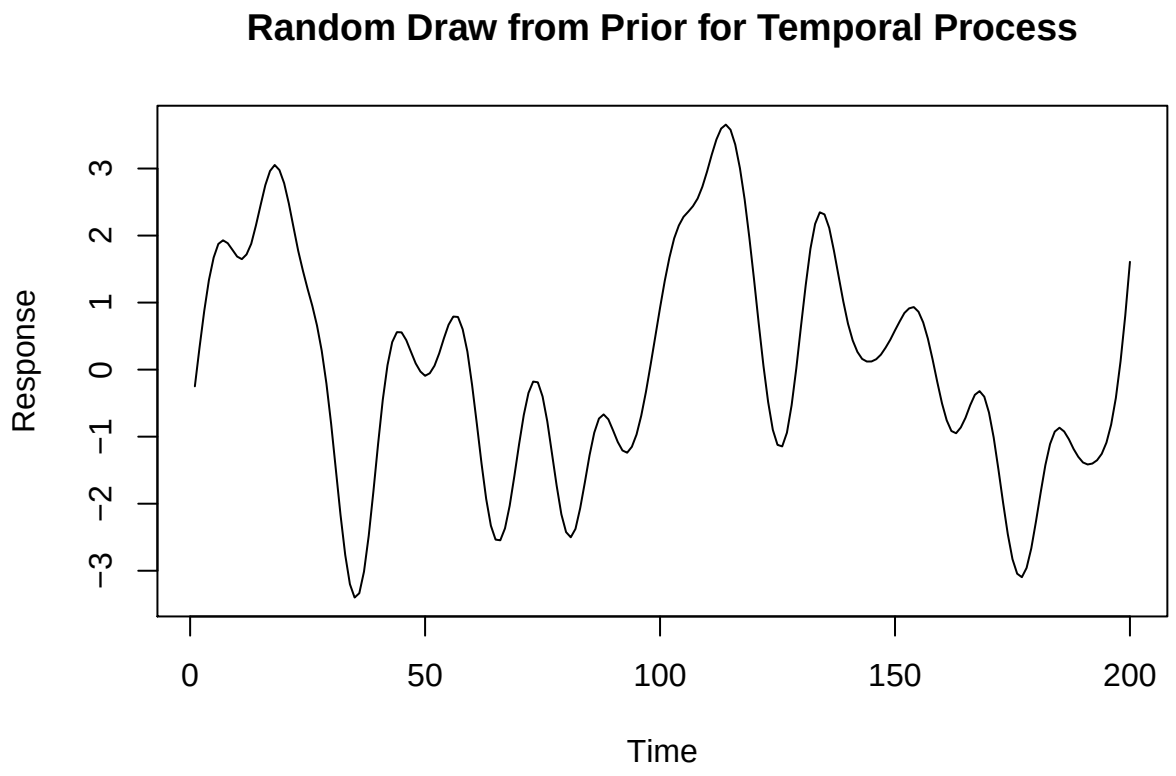
Spatio-Temporal Gaussian Process Demo

David Arthur

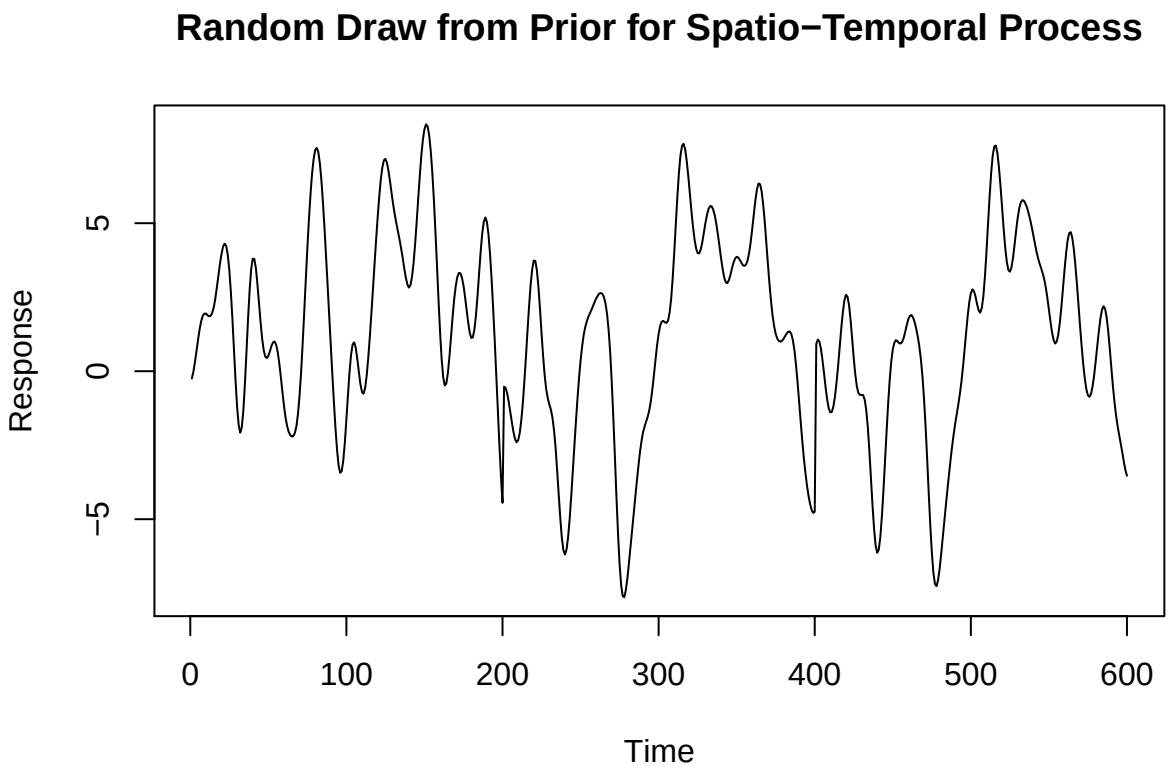
November 30, 2020

```
library(mvtnorm)
set.seed(2)

space.kern <- function(D, l2){
  exp(-1/(2*l2)*D)
}

space.D <- function(lat_long){
  D <- as.matrix(dist(lat_long, diag = TRUE, upper = TRUE))^2
}

time.kern <- function(D, l2){
  exp(-1/(2*l2)*D)
}

time.D <- function(tt){
  D <- as.matrix(dist(tt, diag = TRUE, upper = TRUE))^2
}

nloc <- 3
ntime <- 200
ntime_sub <- 100
N <- ntime*nloc
Nsub <- ntime_sub*nloc

lat_long <- cbind(runif(nloc, 35, 45),
                  runif(nloc, 65, 75))

tt <- seq(0, 24-24/ntime, length = ntime)
tt <- tt-mean(tt)
ind <- seq(1, length(tt), by = 2)
ind_all <- c(ind, ind+ntime, ind+2*ntime)
tt_sub <- tt[ind]

l2_time <- 0.5
l2_space <- 10

sig2_time <- 4
sig2_space <- 2
sig2 <- 0.5

D_time <- time.D(tt)
D_space <- space.D(lat_long)

K_time_true <- sig2_time*time.kern(D_time, l2_time) + diag(10e-6, ntime)
K_space_true <- sig2_space*space.kern(D_space, l2_space) + diag(10e-6, nloc)
```

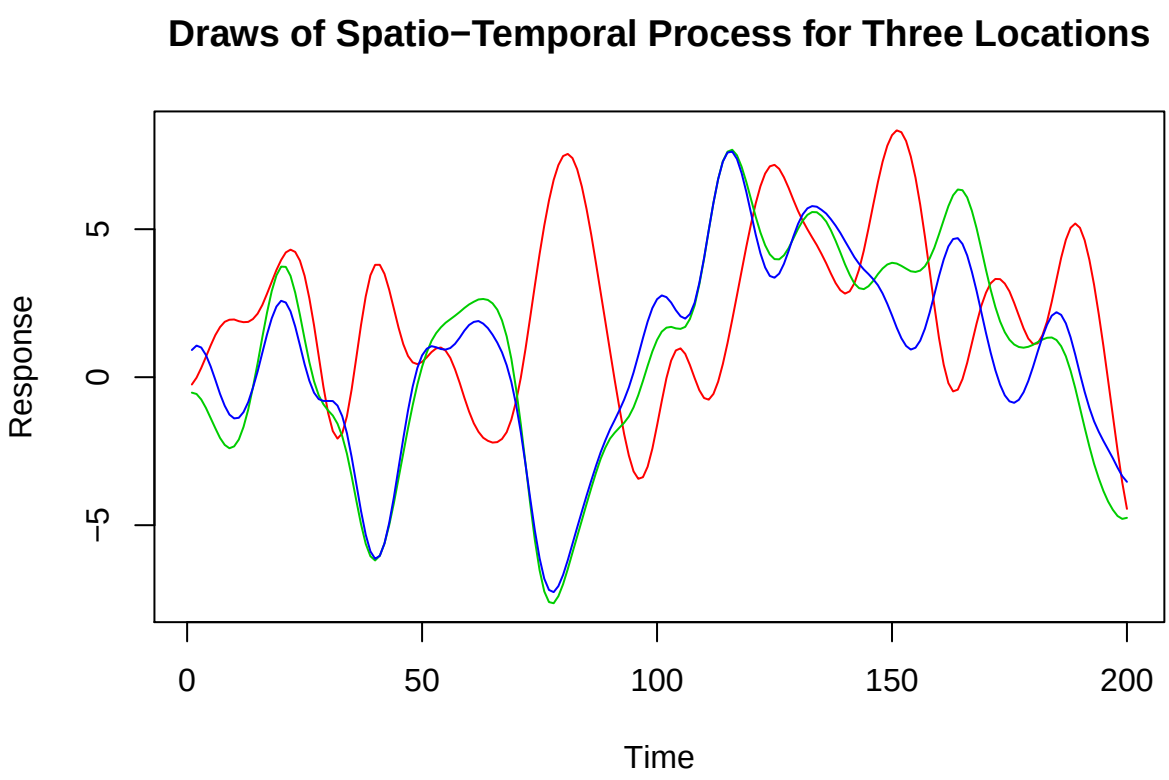
```r
par(mfrow = c(1, 1))

f_time <- t(rmvnorm(1, rep(0, ntime), K_time_true))
ts.plot(f_time, main = "Random Draw from Prior for Temporal Process",
        ylab = "Response")
```



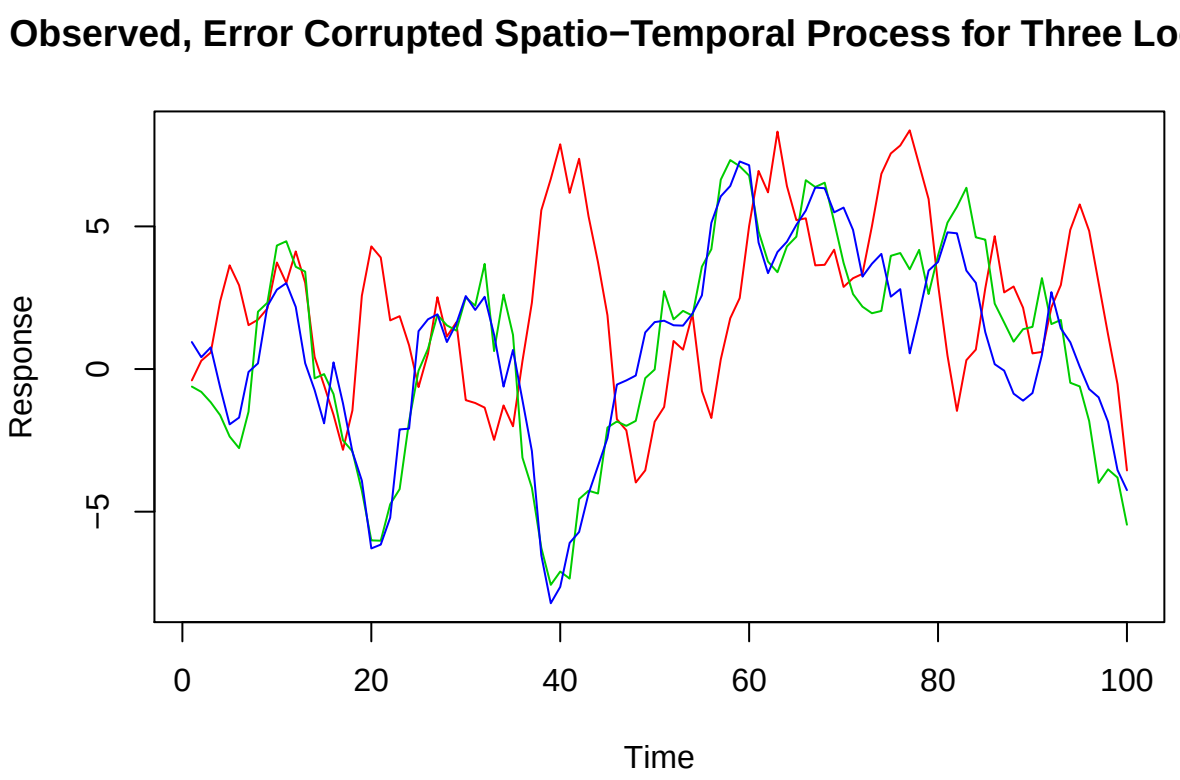

```r
K <- kronecker(K_space_true, K_time_true)

f <- t(rmvnorm(1, rep(0, N), K))
ts.plot(f, main = "Random Draw from Prior for Spatio-Temporal Process",
        ylab = "Response")
```




```
fmat <- matrix(f, nrow = ntime, ncol = nloc, byrow = FALSE)
ts.plot(fmat, col = 2:(nloc+1),
        main = "Draws of Spatio-Temporal Process for Three Locations",
        ylab = "Response")
```




```
Y <- f[ind_all] + rnorm(Nsub, 0, sqrt(sig2))

Ymat <- matrix(Y, nrow = ntime_sub, ncol = nloc, byrow = FALSE)
ts.plot(Ymat, col = 2:(nloc+1),
        main = "Observed, Error Corrupted Spatio-Temporal Process for Three Locations",
        ylab = "Response")
```

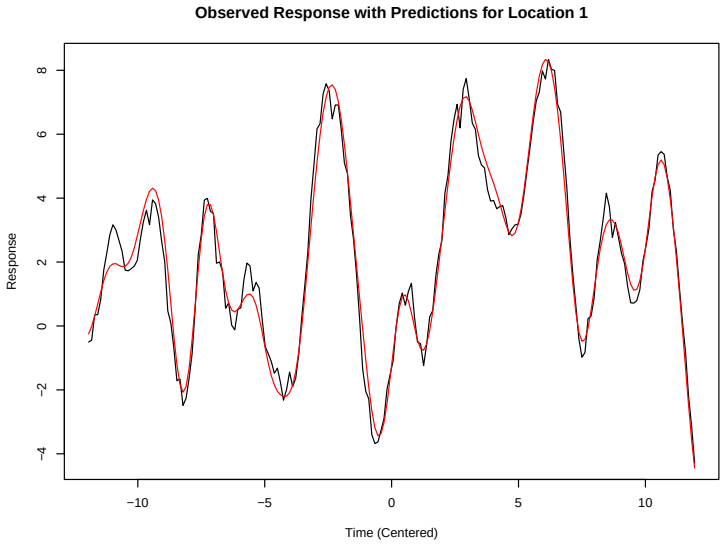


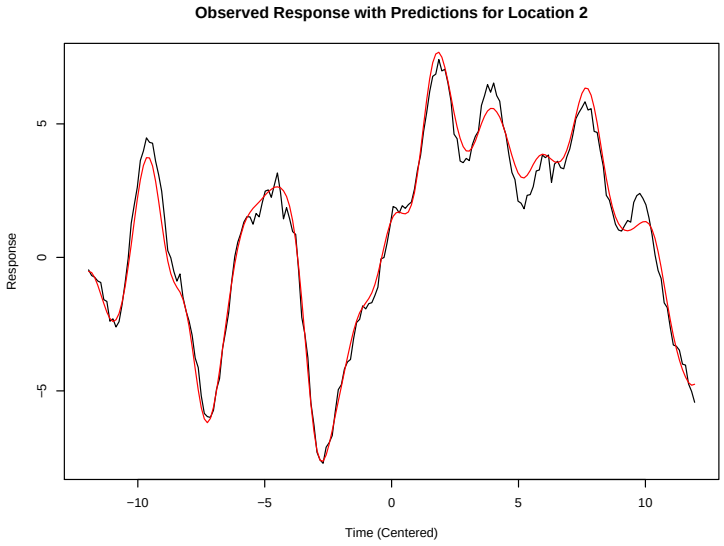

```
niter <- 50

fsave <- matrix(0, niter, Nsub)
sig2_save <- rep(0, niter)
sig2_time_save <- rep(0, niter)
sig2_space_save <- rep(0, niter)

sig2_save[1] <- 2
sig2_time_save[1] <- 2
sig2_space_save[1] <- 2

sig2_a <- 2
sig2_b <- 2
sig2_time_a <- 2
sig2_time_b <- 2
sig2_space_a <- 2
sig2_space_b <- 2

D_space <- space.D(lat_long)
D_time <- time.D(tt_sub)
K_space <- space.kern(D_space, l2_space)
K_time <- time.kern(D_time, l2_time) + diag(10e-02, ntime_sub)
K_space_inv <- solve(K_space)
K_time_inv <- solve(K_time)

for (i in 2:niter){
```

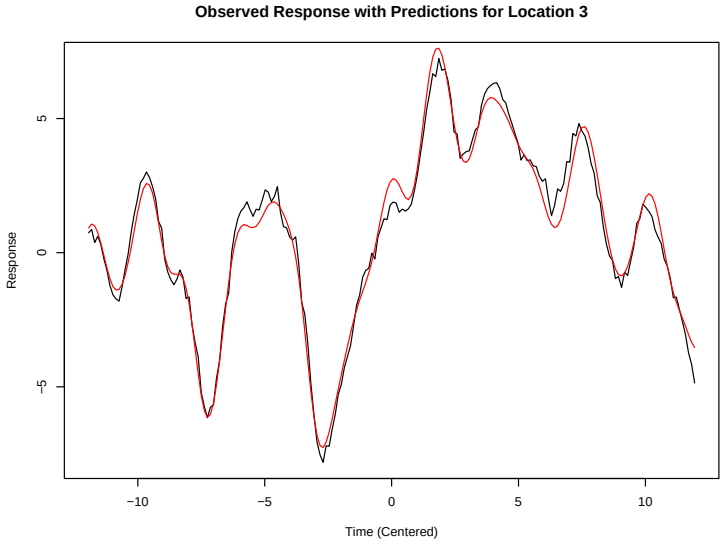
```
  K_space_time <- kronecker(sig2_space_save[i-1]*K_space,
                        sig2_time_save[i-1]*K_time)
  sig2I <- diag(sig2_save[i-1], Nsub)
  f_Sigma <- K_space_time + sig2I

  f_post_Mu <- K_space_time%*%solve(f_Sigma, Y)
  f_post_Sigma <- K_space_time - K_space_time%*%solve(f_Sigma, K_space_time)

  fsave[i,] <- rmvnorm(1, f_post_Mu, f_post_Sigma)

  sig2_save[i] <- 1/rgamma(1, sig2_a + Nsub/2,
                           sig2_b + 0.5*t(Y-fsave[i,])%*%(Y-fsave[i,]))

  sig2_space_Sigma <- kronecker(K_space_inv, 1/sig2_time_save[i-1]*K_time_inv)

  sig2_space_save[i] <- 1/rgamma(1, sig2_space_a + Nsub/2,
                                 sig2_space_b + 0.5*t(fsave[i,])%*%sig2_space_Sigma%*%fsave[i,])

  sig2_time_Sigma <- kronecker(1/sig2_space_save[i]*K_space_inv, K_time_inv)

  sig2_time_save[i] <- 1/rgamma(1, sig2_time_a + Nsub/2,
                                sig2_time_b + 0.5*t(fsave[i,])%*%sig2_time_Sigma%*%fsave[i,])
}

f_pred <- matrix(0, niter, nloc*(ntime-ntime_sub))
In_sub <- diag(ntime_sub*nloc)

for (i in 1:niter){
  K <- kronecker(sig2_space_save[i]*K_space_true, sig2_time_save[i]*K_time_true)
  Kf <- K[ind_all, ind_all]
  Kfpred <- K[-ind_all, ind_all]
  Kpred <- K[-ind_all, -ind_all]

  YSig_inv <- solve(sig2_save[i]*In_sub + Kf, In_sub)
  Kpred_Sigma <- Kpred - Kfpred%*%YSig_inv%*%t(Kfpred)
  Kpred_Mu <- Kfpred%*%YSig_inv%*%Y

  f_pred[i,] <- rmvnorm(1, Kpred_Mu, Kpred_Sigma)
}

f_pred_all <- rep(0, N)
f_pred_all[ind_all] <- colMeans(fsave)
f_pred_all[-ind_all] <- colMeans(f_pred)
```

```
par(mfrow = c(3, 1))

plot(tt, f_pred_all[1:200], type = 'l',
     main = "Observed Response with Predictions for Location 1",
     ylab = "Response",
     xlab = "Time (Centered)")
lines(tt, f[1:200], col = 'red')

plot(tt, f_pred_all[201:400], type = 'l',
```

```
     main = "Observed Response with Predictions for Location 2",
     ylab = "Response",
     xlab = "Time (Centered)")
lines(tt, f[201:400], col = 'red')

plot(tt, f_pred_all[401:600], type = 'l',
     main = "Observed Response with Predictions for Location 3",
     ylab = "Response",
     xlab = "Time (Centered)")
lines(tt, f[401:600], col = 'red')
```


Observed Response with Predictions for Location 1
Response
Time (Centered)


Observed Response with Predictions for Location 2
Response
Time (Centered)


Observed Response with Predictions for Location 3
Response
Time (Centered)