Method 1: Spatial Model with Time Random Effect (Same Response as Time Fixed)



val2 <- data\_inla(f\_sim2, 4, polyg2, w2, n2, 2) #Get Data set up properly (t=3, int = 1)



val2\_known <- filter(val2, !is.na(xmap))

plot(val2\_known$xmap, val2\_known$ymap)



#Create Mesh - nonconvex (create around known values)



prdomain <- inla.nonconvex.hull(as.matrix(val2\_known[, 5:6]),



convex = -0.03, concave = -0.05,



resolution = c(100, 100))



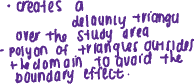
prmesh1 <- inla.mesh.2d(boundary = prdomain,



max.edge = c(2, 3), cutoff = 1,



offset = c(-0.05, -0.05))



par(mfrow=c(1,2))



plot(prmesh1, asp = 1, main = "") #see what mesh looks like



plot(val2\_known$xmap, val2\_known$ymap)



meuse.spde <- inla.spde2.matern(mesh = prmesh1, alpha = 2)



A.meuse <- inla.spde.make.A(mesh = prmesh1, loc = coordinates(val2\_known[,13:14]))



s.index <- inla.spde.make.index(name = "spatial.field",



n.spde = meuse.spde$n.spd



meuse.stack <- inla.stack(data = list(x = val2\_known$xmap),



A = list(A.meuse, 1),



effects = list(c(s.index, list(Intercept = 1)),



list(t = val2\_known$t)),



tag = "meuse.data")



val2\_pred <- filter(val2, is.na(xmap))



val2\_pred2 <- val2\_pred %>% distinct(x\_grid,y\_grid,.keep\_all= TRUE)

A.pred <- inla.spde.make.A(mesh = prmesh1, loc = coordinates(val2\_pred2[,13:14]))

meuse.stack.pred <- inla.stack(data = list(x = NA),



A = list(A.pred, 1),



effects = list(c(s.index, list (Intercept = 1)),



list(t = val2\_pred2$t)),

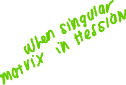
tag = "meuse.pred")

join.stack <- inla.stack(meuse.stack, meuse.stack.pred)



#Fixed time effect

form <- x ~ -1 + Intercept + f(spatial.field, model = meuse.spde) + f(t, model = "rw1")



m\_new5 <- inla(form, data = inla.stack.data(join.stack),

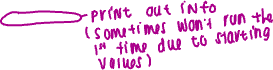


family = "gaussian",



control.predictor = list(A = inla.stack.A(join.stack), compute = TRUE),

control.compute = list(cpo = TRUE, dic = TRUE), verbose = TRUE)



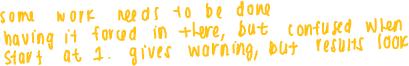
index.pred <- inla.stack.index(join.stack, "meuse.pred")$data



m\_new5$summary.fitted.values[index.pred, "mean"]



m\_new5$summary.fitted.values[index.pred, "sd"]



val2 <- data\_inla(f\_sim2, 3, polyg2, w2, n2, 2) #Get Data set up properly (t=3, int = 1)



val2\_known <- filter(val2, !is.na(xmap))

#Create Mesh - nonconvex (create around known values)

prdomain <- inla.nonconvex.hull(as.matrix(val2\_known[, 5:6]),

convex = -0.03, concave = -0.05,

resolution = c(100, 100))

prmesh1 <- inla.mesh.2d(boundary = prdomain,

max.edge = c(3, 3), cutoff = 0.35,

offset = c(-0.05, -0.05))

plot(prmesh1, asp = 1, main = "") #see what mesh looks like



spde2 <- inla.spde2.pcmatern(mesh = prmesh1,



prior.range = c(0.5, 0.01), # P(range < 0.05) = 0.01



prior.sigma = c(1, 0.01)) # P(sigma > 1) = 0.01

k <- max(val2\_known$t)



iset <- inla.spde.make.index('i', n.spde = spde2$n.spde, n.group = k)

A <- inla.spde.make.A(mesh = prmesh1,

loc = cbind(val2\_known$xval, val2\_known$yval), group = val2\_known$t)

sdat <- inla.stack(

data = list(x = val2\_known$xmap),

A = list(A),

effects = list(c(iset)),



tag = 'stdata')

val2\_pred <- filter(val2, is.na(xmap))

val2\_pred2 <- val2\_pred %>% distinct(x\_grid,y\_grid,.keep\_all= TRUE)

A.pred <- inla.spde.make.A(mesh = prmesh1, loc = cbind(val2\_pred2$xval, val2\_pred2$yval), group = val2\_pred2$t)

iset2 <- inla.spde.make.index('i', n.spde = spde2$n.spde, n.group = max(val2\_pred$t))



sdat.pred <- inla.stack(

data = list(x = NA),

A = list(A.pred),

effects = list(c(iset2)),



tag = 'stdata.pred')

#Join stack

join.stack <- inla.stack(sdat, sdat.pred)

#Fit model

h.spec <- list(rho = list(prior = 'pc.cor1', param = c(0.1, 0.8)))

# Model formula

formulae <- x ~ 0 + f(i, model = spde2, group = i.group,

control.group = list(model = 'ar1', hyper = h.spec))

# PC prior on the autoreg. param.

prec.prior <- list(prior = 'pc.prec', param = c(1, 0.02))

# Model fitting

res2 <- inla(formulae, data = inla.stack.data(join.stack),

control.predictor = list(compute = TRUE,

A = inla.stack.A(join.stack)),

control.family = list(hyper = list(prec = prec.prior)),

control.fixed = list(expand.factor.strategy = 'inla'))

res2$summary.fitted.values[ndex.pred, "mean"]



res2$summary.fitted.values[ndex.pred, "sd"]



What Seems to be happening (in a very general sense):

Model : x = x + y + t

Estimation for missing vales: x\_grid = x\_grid + y\_grid + t

