

Bayesian Spatial Models using the INLA Approximation

Corey S. Sparks, Ph.D. - University of Texas at San Antonio

May 14, 2018

- The INLA Approach to Bayesian models
- Model setup
 - Map of spatial random effects
 - Exceedence probabilities
- Multi - Level Models
 - Higher level predictors
- Create spatial information for higher level units
 - Multi-level model in INLA
 - References

The INLA Approach to Bayesian models

The Integrated Nested Laplace Approximation, or INLA, approach is a recently developed, computationally simpler method for fitting Bayesian models [(Rue et al., 2009 (<http://onlinelibrary.wiley.com/store/10.1111/j.1467-9868.2008.00700.x/asset/j.1467-9868.2008.00700.x.pdf?v=1&t=ih5b86ev&s=9078c3b0adb48d4c15bc49ae3ededc6d1cd684c5>), compared to traditional Markov Chain Monte Carlo (MCMC) approaches. INLA fits models that are classified as latent Gaussian models, which are applicable in many settings (Martino & Rue, 2010 (<http://www.bias-project.org.uk/gmrfcourse/inla-program.pdf>). In general, INLA fits a general form of additive models such as:

$\eta = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ij}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \epsilon_i$

$$\eta = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ij}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \epsilon_i$$

where η is the linear predictor for a generalized linear model formula, and is composed of a linear function of some variables u , β are the effects of covariates, z , and ϵ is an unstructured residual (Rue et al., 2009). As this model is often parameterized as a Bayesian one, we are interested in the posterior marginal distributions of all the model parameters. Rue and Martino (2007) (<http://www.sciencedirect.com/science/article/pii/S0378375807000845>) show that the posterior marginal for the random effects (x) in such models can be approximated as:

$$\tilde{p}(x_i|y) = \sum_k \tilde{p}(x_i|\theta_k, y) \tilde{p}(\theta_k|y) \Delta_k$$

via numerical integration (Rue & Martino, 2007; Schrodle & Held, 2011a (<http://onlinelibrary.wiley.com/doi/10.1002/env.1065/full>), 2011b (<http://link.springer.com/article/10.1007/s00180-010-0208-2>)). The posterior distribution of the hyperparameters (θ) of the model can also be approximated as:

$$\tilde{p}(\theta|y) \propto \frac{p(x, \theta, y)}{\tilde{p}_G(x|\theta, y)} \Big|_{x=x^*(\theta)}$$

, where G is a Gaussian approximation of the posterior and $x^*(\theta)$ is the mode of the conditional distribution of $p(x|\theta, y)$. Thus, instead of using MCMC to find an iterative, sampling-based estimate of the posterior, it is arrived at numerically. This method of fitting the spatial models specified above has been presented by numerous authors (Blangiardo & Cameletti, 2015 (<https://books.google.com/books?hl=en&lr=&id=--HuBgAAQBAJ&oi=fnd&pg=PA259&dq=Blangiardo+%26+Cameletti,+2015&ots=VSDEJ7wfM2&sig=graudrEK2HR7nozMY5N0Q#v=onepage&q=Blangiardo%20%26%20Cameletti%2C%202015&f=false>); Blangiardo et al., 2013 (<http://www.sciencedirect.com/science/article/pii/S1877584513000336>); Lindgren & Rue, 2015 (<http://www.sciencedirect.com/science/article/pii/S2211675315000780>); Martins et al., 2013 (<http://www.sciencedirect.com/science/article/pii/S0167947313001552>); Schrodle & Held, 2011a, 2011b), with comparable results to MCMC.

(<http://www.sciencedirect.com/science/article/pii/S1877584513000336>); Lindgren & Rue, 2015 (<http://www.sciencedirect.com/science/article/pii/S2211675315000780>); Martins et al., 2013 (<http://www.sciencedirect.com/science/article/pii/S0167947313001552>); Schrodle & Held, 2011a, 2011b), with comparable results to MCMC.

Below, I show examples of using INLA to fit Bayesian regression models for areal data from Texas counties and another example of using INLA to estimate multilevel models and perform small area estimation.

```
#Library(rgdal)  
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: Matrix
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge  
## package with: `install.packages('spDataLarge',  
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
library(RColorBrewer)  
library(lattice)  
library(INLA)
```

```
## This is INLA_17.06.20 built 2017-06-20 03:42:30 UTC.  
## See www.r-inla.org/contact-us for how to get help.
```

```
library(tigris)
```

```
## To enable  
## caching of data, set `options(tigris_use_cache = TRUE)` in your R script or .Rprofile.
```

```
##  
## Attaching package: 'tigris'
```

```
## The following object is masked from 'package:graphics':  
##  
##   plot
```

```
library(tidycensus)  
library(ggplot2)  
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:ggplot2':  
##  
##   vars
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
files<-list.files("~/Google Drive/a&m_stuff/workshop_5_14_18/vita_stat/", pattern = "*.csv", full.names =
T)
vital<-lapply(files, read.csv, header=T)
library(plyr)
```

```
## -----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## -----
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
```

```
df <- ldply(vital, data.frame)
df$cofips<-paste(substr(df$GISJOIN, 2,3), substr(df$GISJOIN, 5,7), sep="")
df<-df%>%
  filter(YEAR %in%2000:2007, STATEA==480)%>%
  mutate(births=AGWE001, deaths=AGWG001)%>%
  select(YEAR, cofips, births, deaths)
head(df)
```

```
##   YEAR cofips births deaths
## 1 2000 48001    673    626
## 2 2000 48003    210     94
## 3 2000 48005   1311    824
## 4 2000 48007    266    298
## 5 2000 48009     75     71
## 6 2000 48011     25     24
```

```

popurl<-url("http://www2.census.gov/programs-surveys/popest/datasets/2000-2010/intercensal/county/co-est0
0int-tot.csv")
pops<-read.csv(popurl)
names(pops)<-tolower(names(pops))
pops<-pops%>%
  mutate(cofips = paste(sprintf(fmt = "%02d", state), sprintf(fmt = "%03d",county), sep=""))%>%
  filter(sumlev==50, state==48)

pops$struct<-1:dim(pops)[1]
pops.long<-reshape(data = pops, idvar = "cofips", varying = list(names(pops)[9:16]), direction="long", dr
op = names(pops)[c(2,3,4,5,6,8,17,18,19,20)], v.names = "population")
pops.long$year<-pops.long$time+1999
head(pops.long)

```

```

##          sumlev      ctyname cofips struct time population year
## 48001.1    50 Anderson County 48001     1     1     55062 2000
## 48003.1    50 Andrews County 48003     2     1     12949 2000
## 48005.1    50 Angelina County 48005     3     1     80270 2000
## 48007.1    50 Aransas County 48007     4     1     22452 2000
## 48009.1    50 Archer County 48009     5     1      8966 2000
## 48011.1    50 Armstrong County 48011     6     1      2166 2000

```

```

dat.long<-merge(pops.long, df, by.x=c("cofips", "year"), by.y=c("cofips", "YEAR"))

#v00<-Load_variables(year=2000, dataset = "sf3", cache = T)
cov_dat<-get_decennial(geography = "county", state = "TX", year = 2000, sumfile = "sf3",
  summary_var = "P001001",
  variables = c("P007003", "P007004", "P007010", "P053001", "P089001", "P089002" ),
  output = "wide")

```

```
## Getting data from the 2000 decennial Census
```

```

cov_dat<-cov_dat%>%
  mutate(cofips=GE0ID,pwhite=P007003/summary_value, pblack=P007004/summary_value, phisp=P007010/summary_v
alue,medhhinc=as.numeric(scale(P053001)), ppov=P089002/P089001)

final.dat<-merge(dat.long, cov_dat, by="cofips")

rates<-aggregate(cbind(deaths, births,population)~1, final.dat,sum)
rates$dr<-rates$deaths/rates$population
rates$br<-rates$births/rates$population

final.dat$E_d<-final.dat$population*rates$dr

final.dat$E_b<-final.dat$population*rates$br
final.dat<-final.dat[order(final.dat$cofips, final.dat$year),]
final.dat$id<-1:dim(final.dat)[1]

head(final.dat)

```

```
##   cofips year sumlev      ctyname struct time population births deaths
## 1  48001 2000     50 Anderson County      1  1     55062     673    626
## 2  48001 2001     50 Anderson County      1  2     54263     651    576
## 3  48001 2002     50 Anderson County      1  3     54740     712    616
## 4  48001 2003     50 Anderson County      1  4     56068     611    596
## 5  48001 2004     50 Anderson County      1  5     56245     670    597
## 6  48001 2005     50 Anderson County      1  6     56873     670    551
##   GEOID      NAME P007003 P007004 P007010 P053001 P089001 P089002
## 1 48001 Anderson County  34832  12849   6614   31957   40401   6654
## 2 48001 Anderson County  34832  12849   6614   31957   40401   6654
## 3 48001 Anderson County  34832  12849   6614   31957   40401   6654
## 4 48001 Anderson County  34832  12849   6614   31957   40401   6654
## 5 48001 Anderson County  34832  12849   6614   31957   40401   6654
## 6 48001 Anderson County  34832  12849   6614   31957   40401   6654
##   summary_value  pwhite  pblack  phisp  medhhinc  ppov
## 1          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
## 2          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
## 3          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
## 4          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
## 5          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
## 6          55109 0.6320565 0.2331561 0.1200167 -0.09855534 0.1646989
##      E_d      E_b id
## 1 382.8695 942.6150  1
## 2 377.3137 928.9368  2
## 3 380.6305 937.1027  3
## 4 389.8646 959.8369  4
## 5 391.0954 962.8670  5
## 6 395.4621 973.6178  6
```

```
options(scipen=999)
```

Next we make the spatial information, we get the polygons from census directly using `counties` from the `tigris` package.

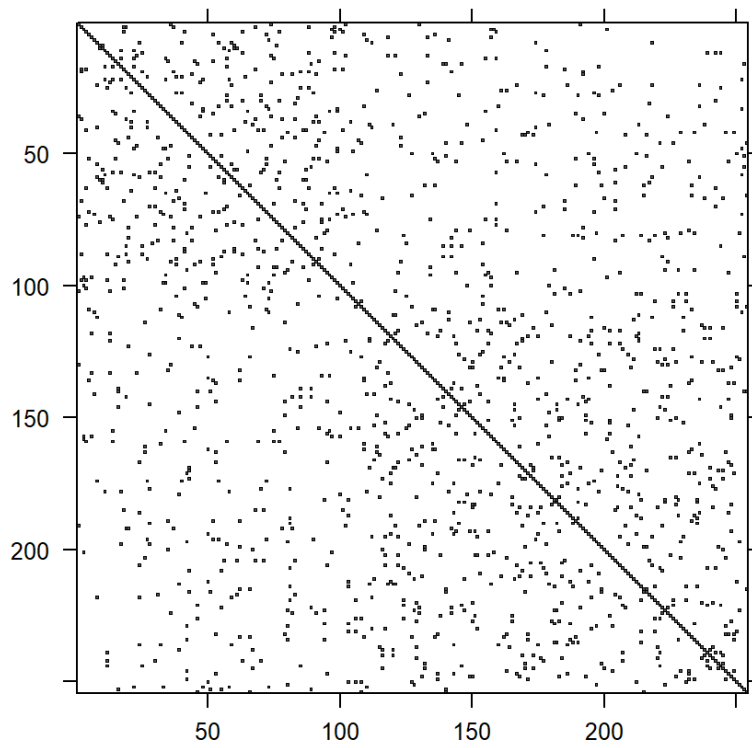
```
us_co<-counties(state = "TX", cb = T)
```

```
#In INLA, we don't need FIPS codes, we need a simple numeric index for our counties
us_co$struct<-1:dim(us_co@data[1])
```

```
## Warning in 1:dim(us_co@data[1]): numerical expression has 2 elements: only
## the first used
```

```
nbs<-poly2nb(us_co, queen = T, row.names = us_co$struct)
mat <- nb2mat(nbs, style="B",zero.policy=TRUE)
colnames(mat) <- rownames(mat)
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
```

```
nb2INLA("am_graph",nbs)
am_adj <-paste(getwd(),"/am_graph",sep="")
H<-inla.read.graph(filename="am_graph")
image(inla.graph2matrix(H), xlab="", ylab="", main="")
```

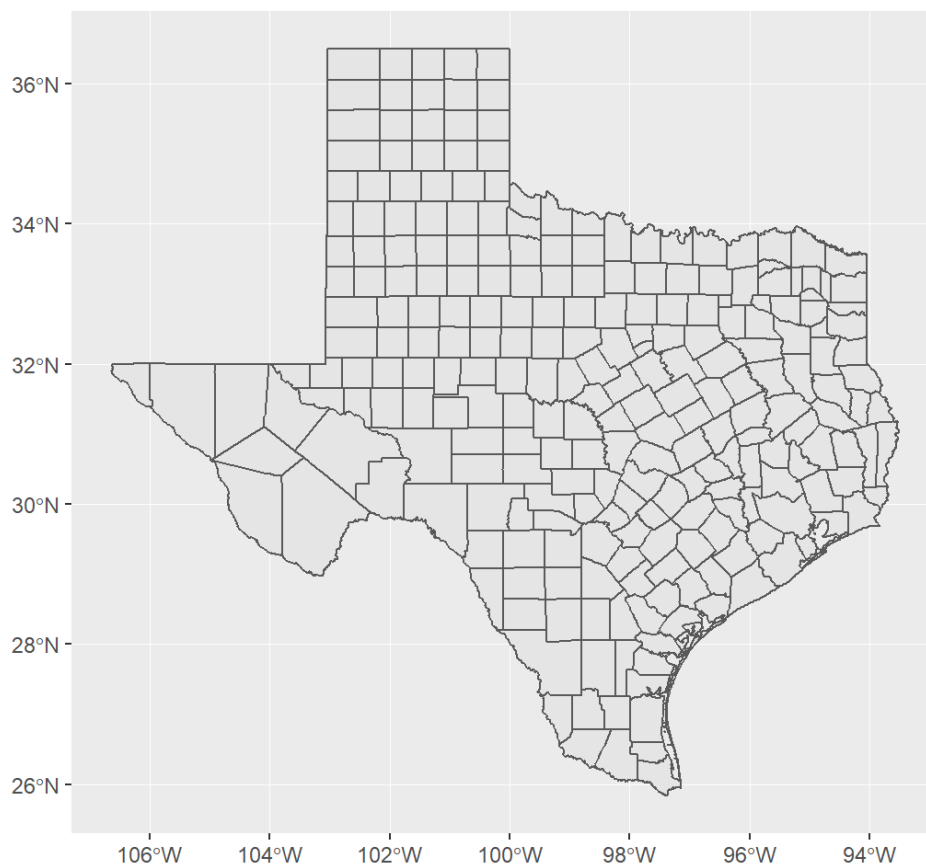


Dimensions: 254 x 254

```
library(sf)
```

```
## Linking to GEOS 3.6.1, GDAL 2.2.3, proj.4 4.9.3
```

```
tx<-st_as_sf(us_co)  
tx%>%  
  ggplot()+geom_sf()
```



```
final.dat<-merge( tx,final.dat, by="struct")
```

Model setup

- We have a count outcome (deaths and births), in counties over time, and a set of time-constant covariates.
- We have several options in the GLM framework with which to model these data, for example:
- Binomial -

$$y_{ij} \sim \text{Bin}(\pi_{ij}): \text{logit}(\pi_{ij}) = \beta_0 + x' \beta_k$$

- Poisson -

$$y_{ij} \sim \text{Pois}(\lambda_{ij} E_{ij}): \log(\lambda_{ij}) = \log(E_{ij}) + \beta_0 + x' \beta_k$$

- Negative Binomial -

$$y_{ij} \sim \text{Neg Bin}(\mu_{ij}, \alpha, E_{ij}): \log(\mu_{ij}) = \log(E_{ij}) + \beta_0 + x' \beta_k$$

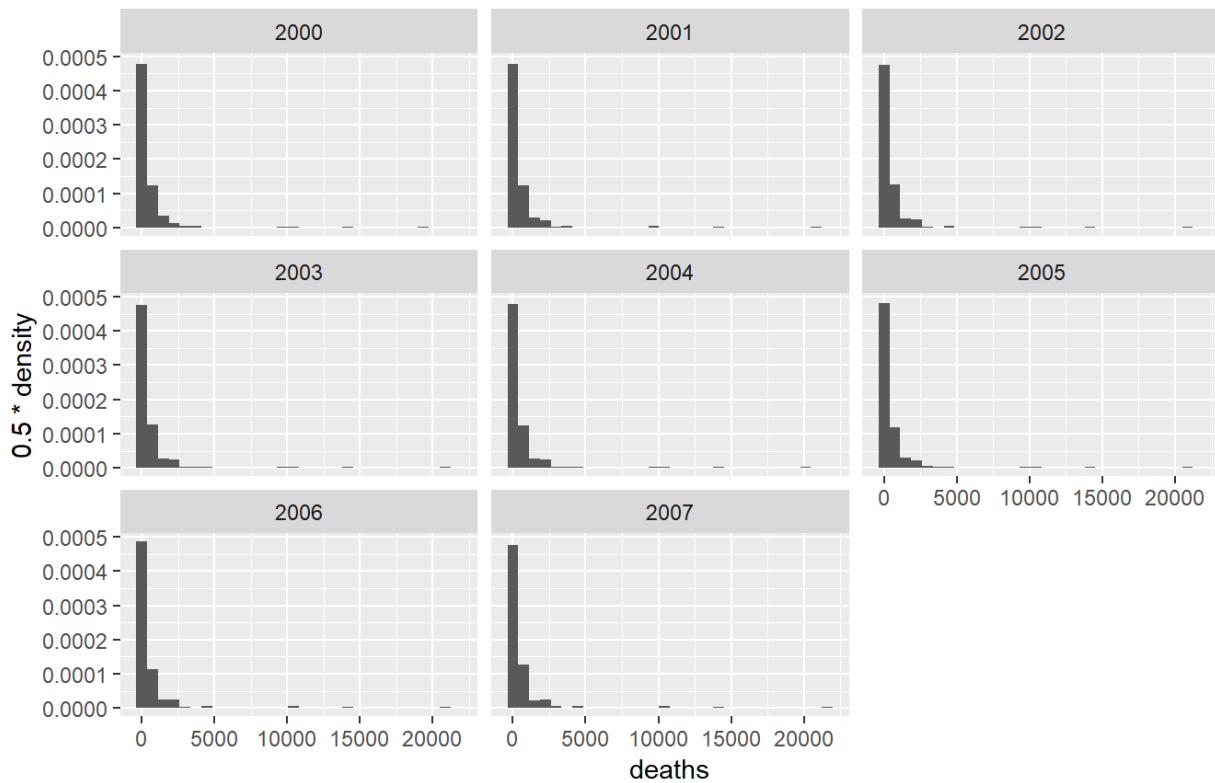
- In addition to various zero-inflated versions of these data.

```
ggplot(data = final.dat)+geom_histogram(aes(x =deaths , y=0.5*..density..))+facet_wrap(~year)+
  ggtitle(label = "Distribution of Deaths by Year", subtitle = "Texas Counties, 2000-2007")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Distribution of Deaths by Year

Texas Counties, 2000-2007

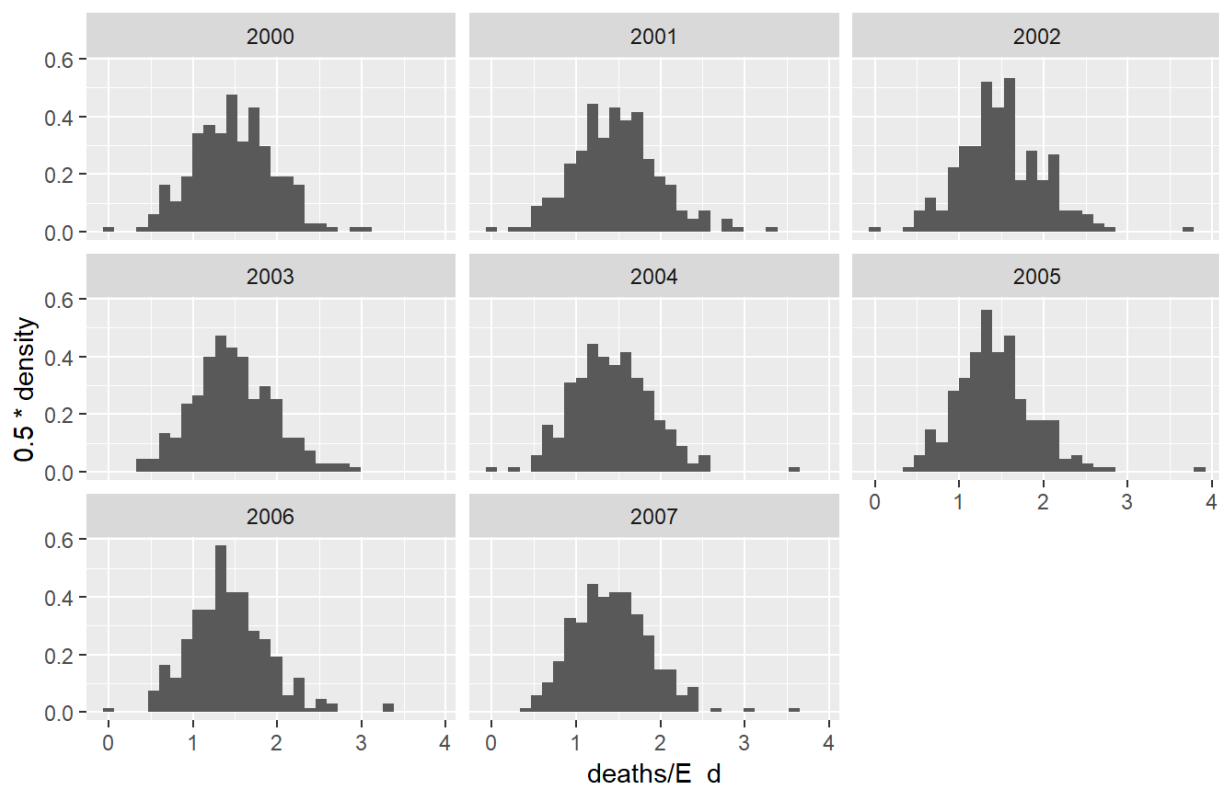


```
ggplot(data = final.dat)+geom_histogram(aes(x =deaths/E_d , y=0.5*..density..))+facet_wrap(~year)+
  ggtitle(label = "Distribution of Morality Relative Risk by Year", subtitle = "Texas Counties, 2000-2007")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Distribution of Morality Relative Risk by Year

Texas Counties, 2000-2007



We can fit these model using the bayesian framework with INLA.

First, we consider the basic GLM for the mortality outcome, with out any hierarhical structure. We can write this model as a Negative Binomial model, for instance as:

$$\text{Deaths}_{ij} = \log(E_d) + X' \beta$$

INLA will use vague Normal priors for the β 's, and we have not other parameters in the model to specify priors for. INLA does not require you to specify all priors, as all parameters have a default prior specification.

```
#Model specification:
f1<-deaths~scale(pblack)+scale(phisp)+scale(ppov)+year

#Model fit
mod1<-inla(formula = f1,data = final.dat, #linear predictor - fixed effects
           family = "nbinomial", E = E_d, #marginal distribution for the outcome, expected count
           control.compute = list(dic=T), # compute DIC or not?
           control.predictor = list(link=1)) #estimate predicted values & their marginals or not?

#model summary
summary(mod1)
```

```

##
## Call:
## c("inla(formula = f1, family = \"nbinomial\", data = final.dat, E = E_d, \" \" control.compute = list(dic = T), control.predictor = list(link = 1))" )
##
## Time used:
## Pre-processing      Running inla Post-processing      Total
##           1.2490           2.4679           0.1527           3.8696
##
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept)  16.7813 4.9367    7.0870  16.7815   26.4657  16.7822  0
## scale(pblack) -0.0858 0.0063   -0.0983 -0.0858   -0.0733 -0.0858  0
## scale(phisp)  -0.3171 0.0095   -0.3357 -0.3171   -0.2985 -0.3172  0
## scale(ppov)   0.2005 0.0093    0.1821  0.2005    0.2188  0.2005  0
## year         -0.0082 0.0025   -0.0130 -0.0082   -0.0034 -0.0082  0
##
## The model has no random effects
##
## Model hyperparameters:
##           mean      sd
## size for the nbinomial observations (1/overdispersion) 18.54 0.6598
##           0.025quant 0.5quant
## size for the nbinomial observations (1/overdispersion)  17.27  18.53
##           0.975quant  mode
## size for the nbinomial observations (1/overdispersion)  19.86 18.51
##
## Expected number of effective parameters(std dev): 5.192(0.0061)
## Number of equivalent replicates : 391.34
##
## Deviance Information Criterion (DIC) ....: 21420.19
## Effective number of parameters .....: 6.233
##
## Marginal log-Likelihood: -10747.31
## Posterior marginals for linear predictor and fitted values computed

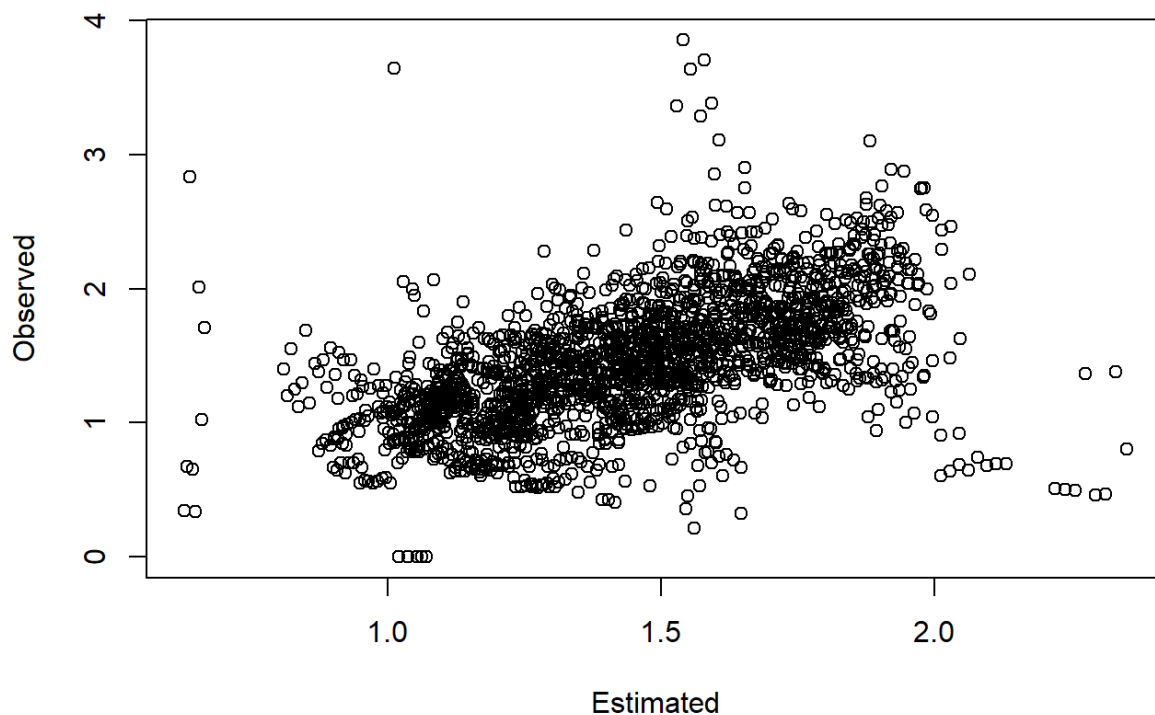
```

Plot our observed vs fitted values

```

plot(x= mod1$summary.fitted.values$mean, y=final.dat$deaths/final.dat$E_d , ylab="Observed", xlab="Estimated" )

```



Now we add basic nesting of rates within counties, with a random intercept term for each county. This would allow there to be heterogeneity in the mortality rate for each county, over and above each county's observed characteristics.

This model would be:

$$\text{Deaths}_{ij} = \log(E_{_d}) + X' \beta + u_j$$

$$u_j \sim \text{Normal}(0, \tau_u)$$

where τ_u here is the precision, not the variance and **precision = 1/variance**. INLA puts a log-gamma prior on the the precision by default.

```
f2<-deaths~scale(pblack)+scale(phisp)+scale(ppov)+year+ #fixed effects
f(struct, model = "iid") #random effects
mod2<-inla(formula = f2,data = final.dat,
  family = "nbinomial", E = E_d,
  control.compute = list(dic=T),
  control.predictor = list(link=1))

#total model summary
summary(mod2)
```

```

##
## Call:
## c("inla(formula = f2, family = \"nbinomial\", data = final.dat, E = E_d, ", " control.compute = list(dic = T), control.predictor = list(link = 1))" )
##
## Time used:
## Pre-processing      Running inla Post-processing      Total
##           1.2208           16.6113           0.2227           18.0548
##
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept)  15.9592 1.2344   13.5361  15.9588   18.3825 15.9580  0
## scale(pblack) -0.0771 0.0176   -0.1117 -0.0772   -0.0425 -0.0772  0
## scale(phisp)  -0.2986 0.0254   -0.3484 -0.2986   -0.2486 -0.2987  0
## scale(ppov)   0.1806 0.0236    0.1341  0.1806    0.2269  0.1807  0
## year         -0.0078 0.0006   -0.0090 -0.0078   -0.0066 -0.0078  0
##
## Random effects:
## Name      Model
## struct    IID model
##
## Model hyperparameters:
##           mean      sd
## size for the nbinomial observations (1/overdispersion) 1343.36 201.60
## Precision for struct           16.99  1.59
##           0.025quant 0.5quant
## size for the nbinomial observations (1/overdispersion)  999.10 1324.76
## Precision for struct           14.04  16.93
##           0.975quant   mode
## size for the nbinomial observations (1/overdispersion) 1790.78 1285.36
## Precision for struct           20.29  16.83
##
## Expected number of effective parameters(std dev): 249.36(0.6697)
## Number of equivalent replicates : 8.149
##
## Deviance Information Criterion (DIC) ...: 17342.39
## Effective number of parameters .....: 250.06
##
## Marginal log-Likelihood: -9130.00
## Posterior marginals for linear predictor and fitted values computed

```

Marginal Distributions of hyperparameters We can plot the posterior marginal of the hyperparameter in this model, in this case $\sigma_u = 1/\tau_u$

```

m2<- inla.tmarginal(
  function(x) (1/x), #invert the precision to be on variance scale
  mod2$marginals.hyperpar$`Precision for struct`)

inla.hpdmarginal(.95, marginal=m2)

```

```

##           low      high
## level:0.95 0.04883561 0.07047514

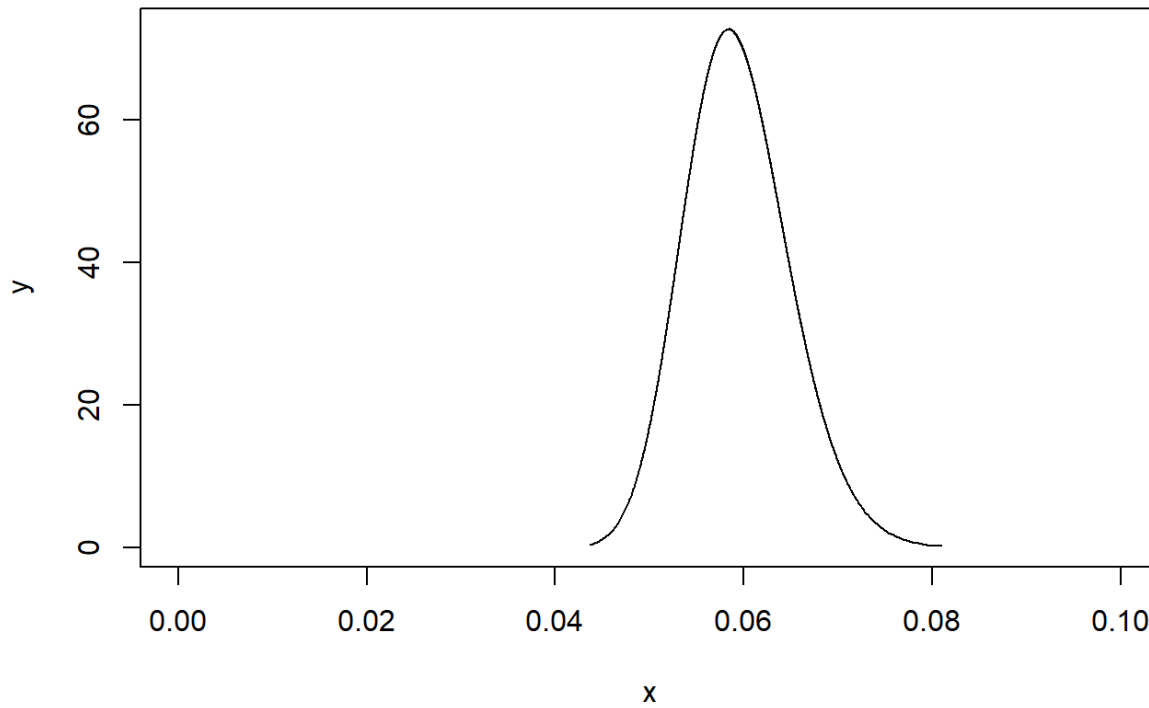
```

```

plot(m2, type="l", main=c("Posterior distribution for between county variance", "- IID model -"), xlim=c(0, .1))

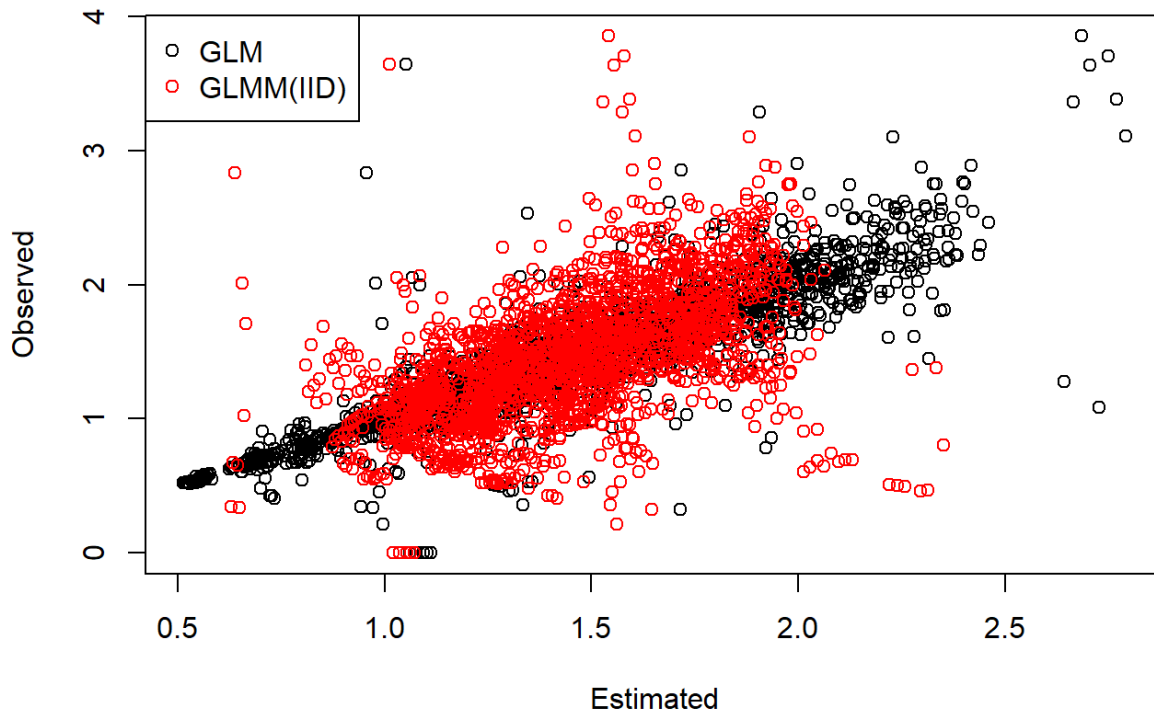
```

Posterior distribution for between county variance - IID model -



Observed vs. Fitted values

```
plot(x= mod2$summary.fitted.values$mean, y=final.dat$deaths/final.dat$E_d , ylab="Observed", xlab="Estimated" )
points(x= mod1$summary.fitted.values$mean, y=final.dat$deaths/final.dat$E_d, col=2)
legend("topleft", legend = c("GLM", "GLMM(IID)"), col=c(1,2), pch=1)
```

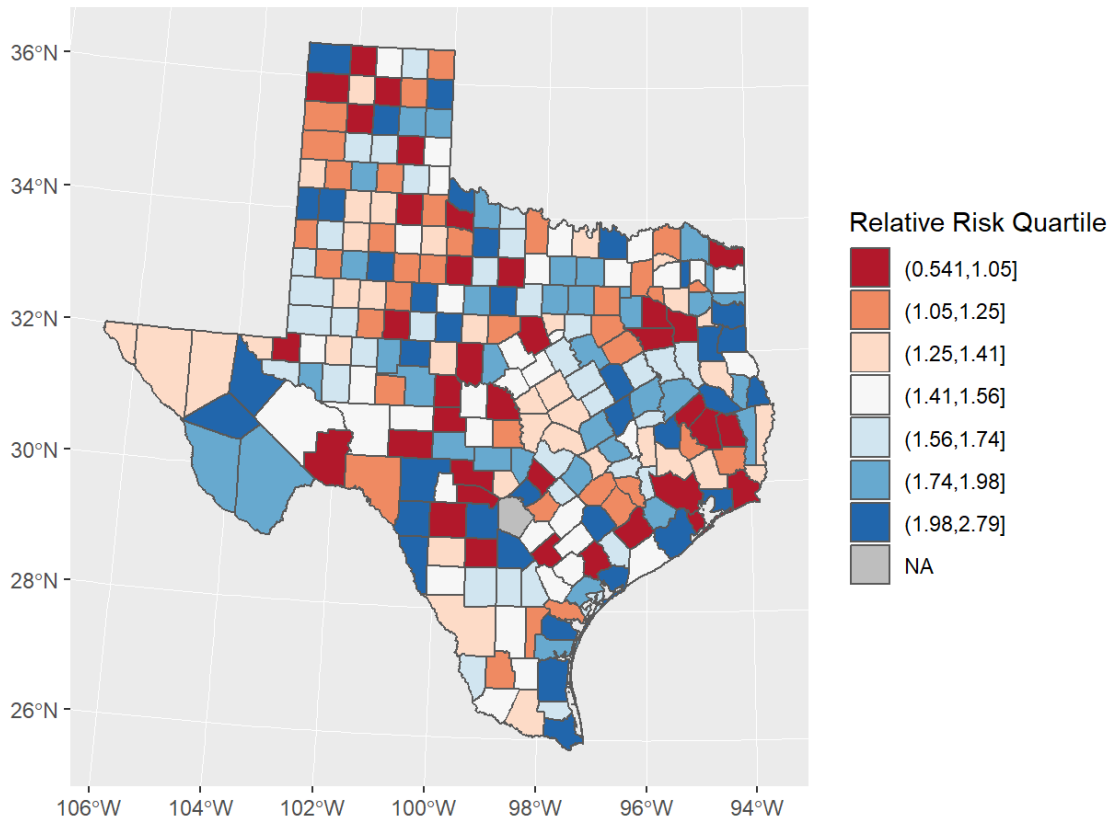


We see a couple of things here. First, we see much closer agreement between the observed and predicted values, and we see the shrinkage of the estimates toward the mean.

```
final.dat$fitted_m2<-mod2$summary.fitted.values$mean

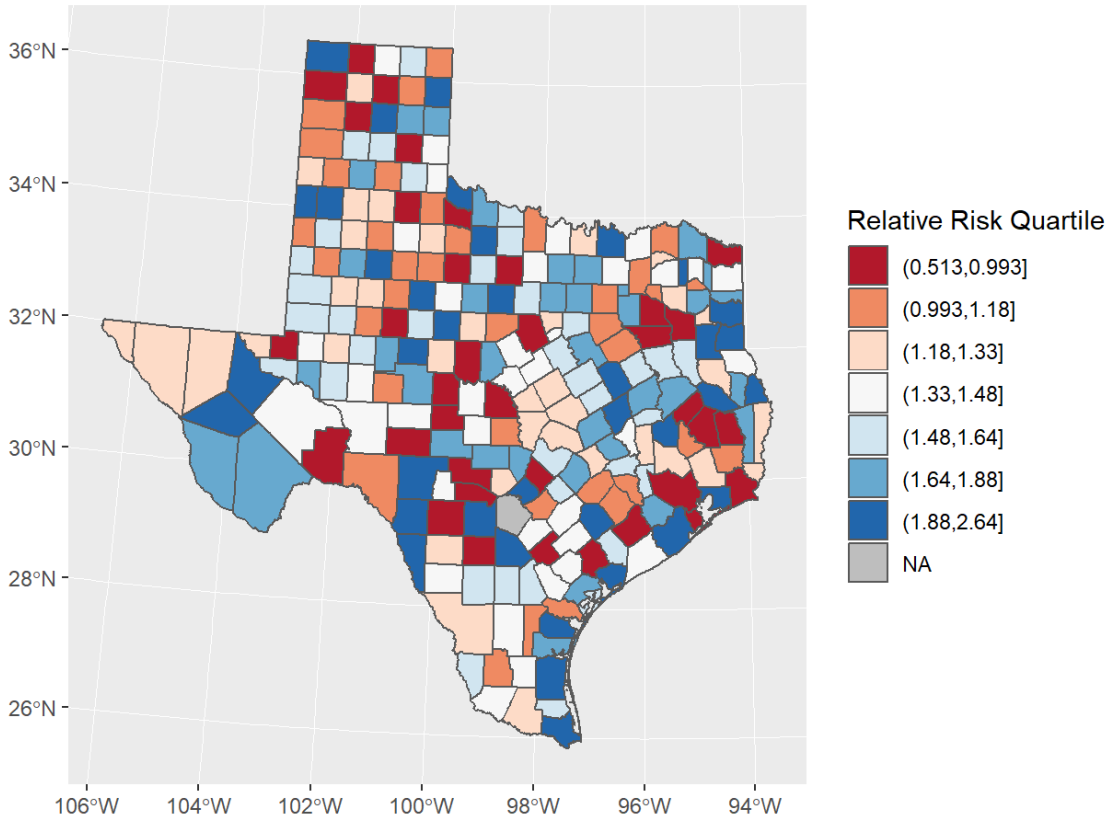
final.dat%>%
  filter(year%in%c(2000))%>%
  mutate(qrr=cut(fitted_m2, breaks = quantile(fitted_m2, p=seq(0,1,length.out = 8))))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "RdBu" )+scale_fill_brewer(palette = "RdBu", na.value="grey")+guides(fill=guide_legend(title="Relative Risk Quartile"))+ggtitle(label="Relative Risk Quartile - IID Model, 2000")+coord_sf(crs = 102008)
```

Relative Risk Quartile - IID Model, 2000



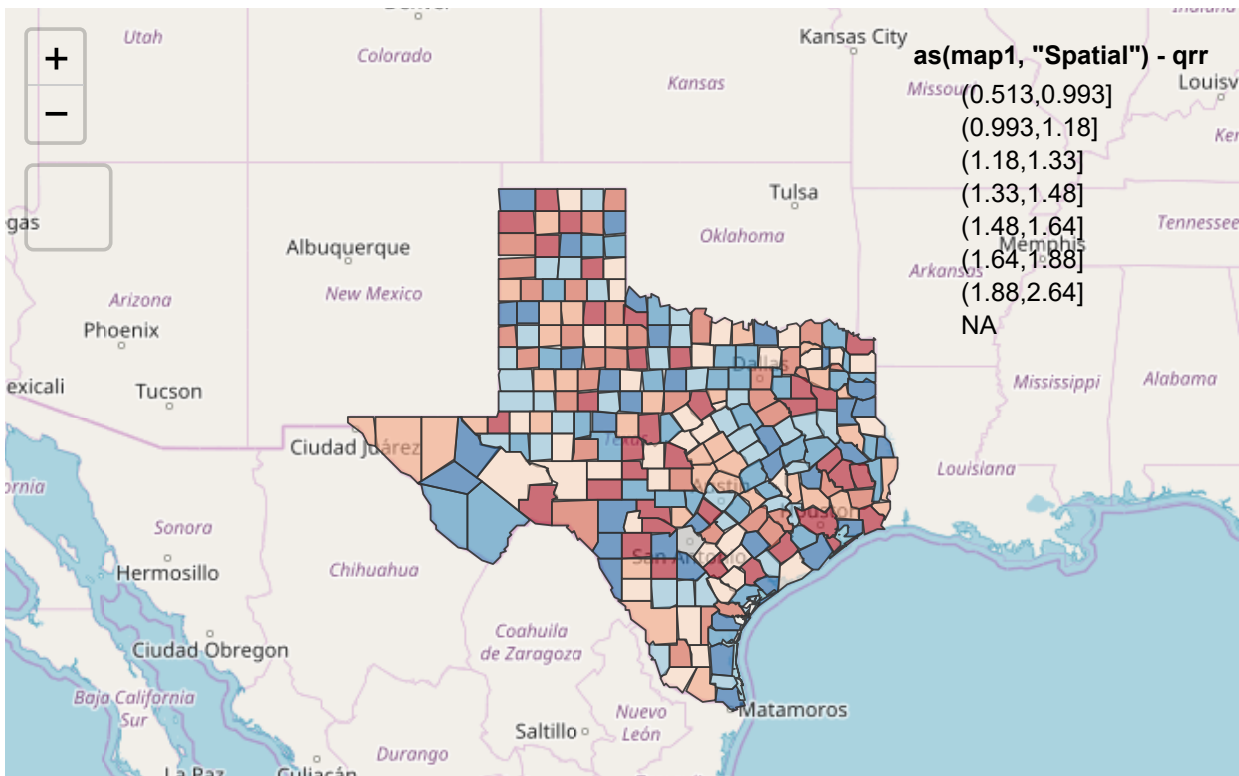
```
final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(fitted_m2, breaks = quantile(fitted_m2, p=seq(0,1,length.out = 8))))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "RdBu" )+scale_fill_brewer(palette = "RdBu", na.value="grey")+guides(fill=guide_legend(title="Relative Risk Quartile"))+ggtitle(label="Relative Risk Quartile - IID Model, 2007")+coord_sf(crs = 102008)
```

Relative Risk Quartile - IID Model, 2007



library(mapview)

```
map1<-final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(fitted_m2, breaks = quantile(fitted_m2, p=seq(0,1,length.out = 8))))
clrs <- colorRampPalette(brewer.pal(8, "RdBu"))
mapView(as(map1, "Spatial"), zcol="qrr", legend=T, col.regions=clrs, map.types="OpenStreetMap")
```





Model with spatial correlation - Besag, York, and Mollie (1991) model and temporal heterogeneity

$$\text{Deaths}_{ij} = \log(E_d) + X'\beta + u_j + v_j + \gamma_t$$

Which has two random effects, one an IID random effect and the second a spatially correlated random effect, specified as a conditionally autoregressive prior for the v_j 's. This is the Besag model:

$$v_j | v_{\neq j}, \sim \text{Normal}\left(\frac{1}{n_i} \sum_{i \sim j} v_j, \frac{1}{n_i \tau}\right)$$

and u_j is an IID normal random effect, γ_t is also given an IID Normal random effect specification, and there are now three hyperparameters, τ_u and τ_v and τ_γ and each are given log-gamma priors.

For the BYM model we must specify the spatial connectivity matrix in the random effect.

```
#final.dat$year_c<-final.dat$year - 2004
f3<-deaths~scale(pblack)+scale(phisp)+scale(ppov)+
  f(struct, model = "bym", constr = T, scale.model = T, graph = mat)+
  f(year, model="iid") #temporal random effect
mod3<-inla(formula = f3,data = final.dat,
  family = "nbinomial", E = E_d,
  control.compute = list(dic=T),
  control.predictor = list(link=1))

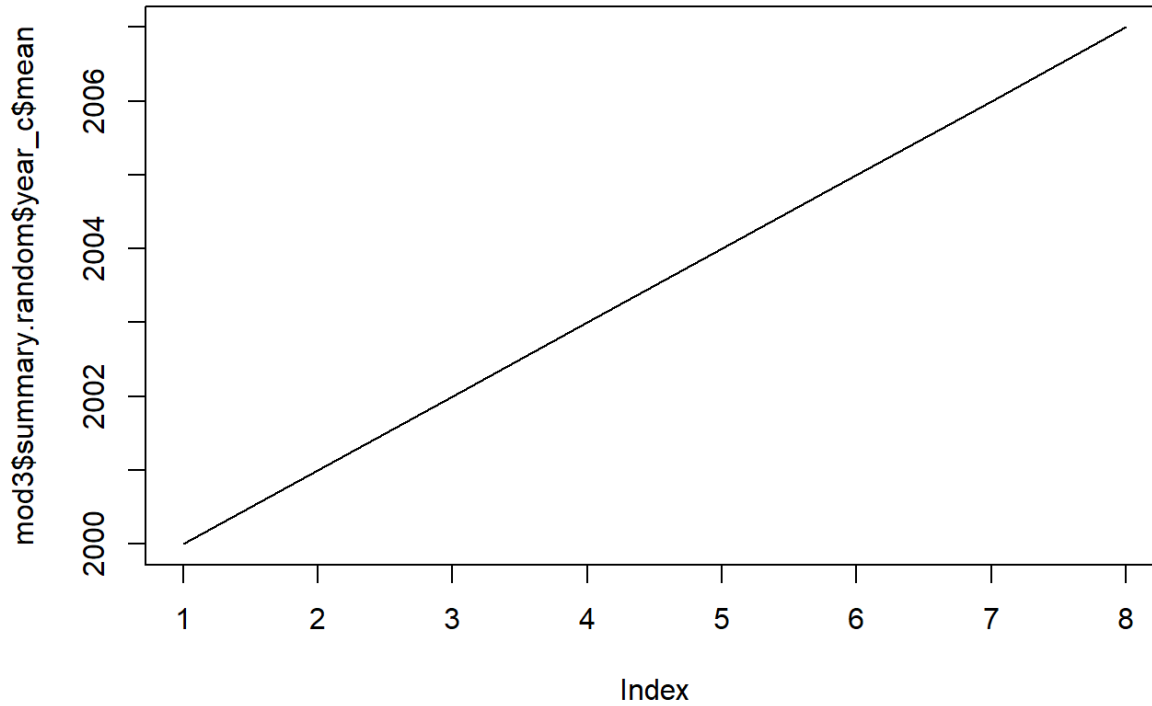
#total model summary
summary(mod3)
```

```

##
## Call:
## c("inla(formula = f3, family = \"nbinomial\", data = final.dat, E = E_d, ", " control.compute = list(dic = T), control.predictor = list(link = 1))" )
##
## Time used:
## Pre-processing      Running inla Post-processing      Total
##           1.5383           15.3501           0.4151           17.3036
##
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant  mode kld
## (Intercept)  0.3312 0.0171   0.2977  0.3312   0.3646 0.3312  0
## scale(pblack) -0.0772 0.0177  -0.1118 -0.0772  -0.0425 -0.0772  0
## scale(phisp)  -0.2985 0.0254  -0.3484 -0.2986  -0.2486 -0.2986  0
## scale(ppov)   0.1805 0.0236   0.1340  0.1805   0.2268  0.1805  0
##
## Random effects:
## Name      Model
## struct    BYM model
## year      IID model
##
## Model hyperparameters:
##
##           mean      sd
## size for the nbinomial observations (1/overdispersion) 1152.93 124.053
## Precision for struct (iid component)                   17.06  1.629
## Precision for struct (spatial component)                 2605.79 2141.720
## Precision for year                                       2874.10 1518.094
##
##           0.025quant 0.5quant
## size for the nbinomial observations (1/overdispersion)  953.83 1136.23
## Precision for struct (iid component)                      14.12  16.97
## Precision for struct (spatial component)                  335.39 2041.52
## Precision for year                                        920.94 2555.57
##
##           0.975quant  mode
## size for the nbinomial observations (1/overdispersion) 1437.98 1091.44
## Precision for struct (iid component)                      20.52  16.77
## Precision for struct (spatial component)                  8222.00 962.02
## Precision for year                                        6710.02 1992.57
##
## Expected number of effective parameters(std dev): 255.12(0.5095)
## Number of equivalent replicates : 7.965
##
## Deviance Information Criterion (DIC) ...: 17298.02
## Effective number of parameters .....: 240.30
##
## Marginal log-Likelihood: -8975.37
## Posterior marginals for linear predictor and fitted values computed

```

```
plot(y=mod3$summary.random$year_c$mean,x=unique(final.dat$year), type="l")
```



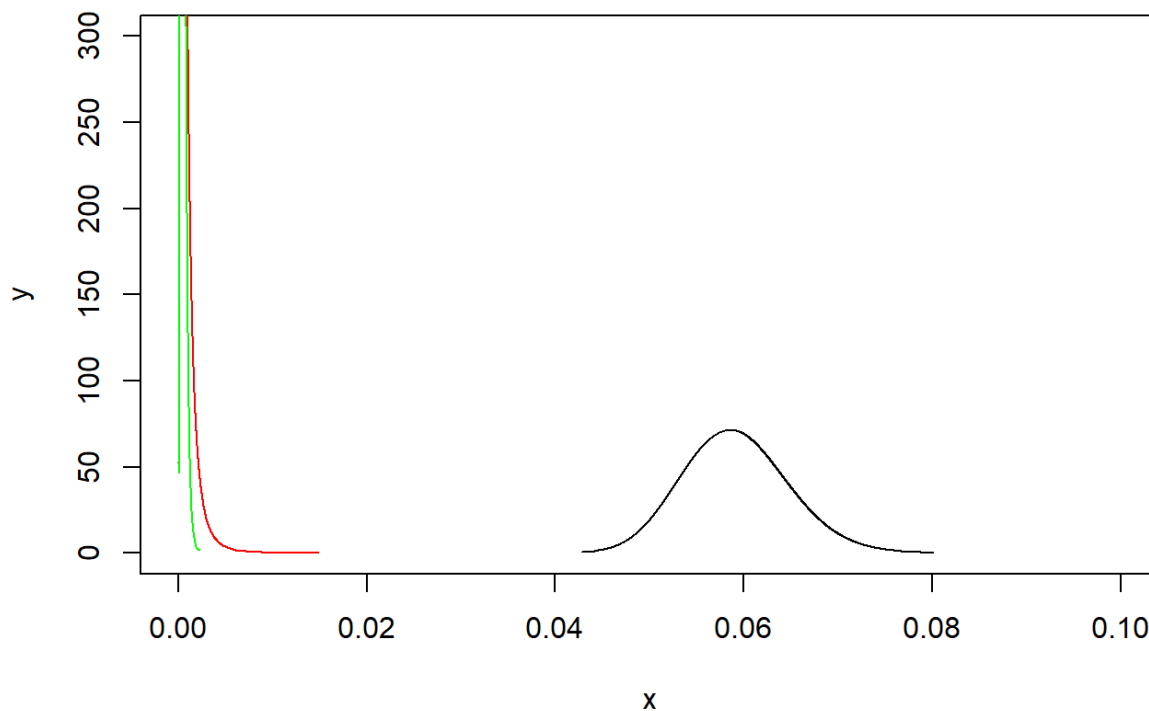
```

m3a<- inla.tmarginal(
  function(x) (1/x),
  mod3$marginals.hyperpar$`Precision for struct (iid component)` )
m3b<- inla.tmarginal(
  function(x) (1/x),
  mod3$marginals.hyperpar$`Precision for struct (spatial component)` )
m3c<- inla.tmarginal(
  function(x) (1/x),
  mod3$marginals.hyperpar$`Precision for year` )

plot(m3a, type="l", main=c("Posterior distribution for between county variance", "- IID model -"), xlim=c(
0, .1), ylim=c(0,300))
lines(m3b, col="red")
lines(m3c, col="green")

```

Posterior distribution for between county variance - IID model -



```
inla.hpdmarginal(.95,m3a)
```

```
##           low      high
## level:0.95 0.04836997 0.07021438
```

```
inla.hpdmarginal(.95,m3b)
```

```
##           low      high
## level:0.95 0.00005184317 0.002186784
```

```
inla.hpdmarginal(.95,m3c)
```

```
##           low      high
## level:0.95 0.0001034234 0.0009307634
```

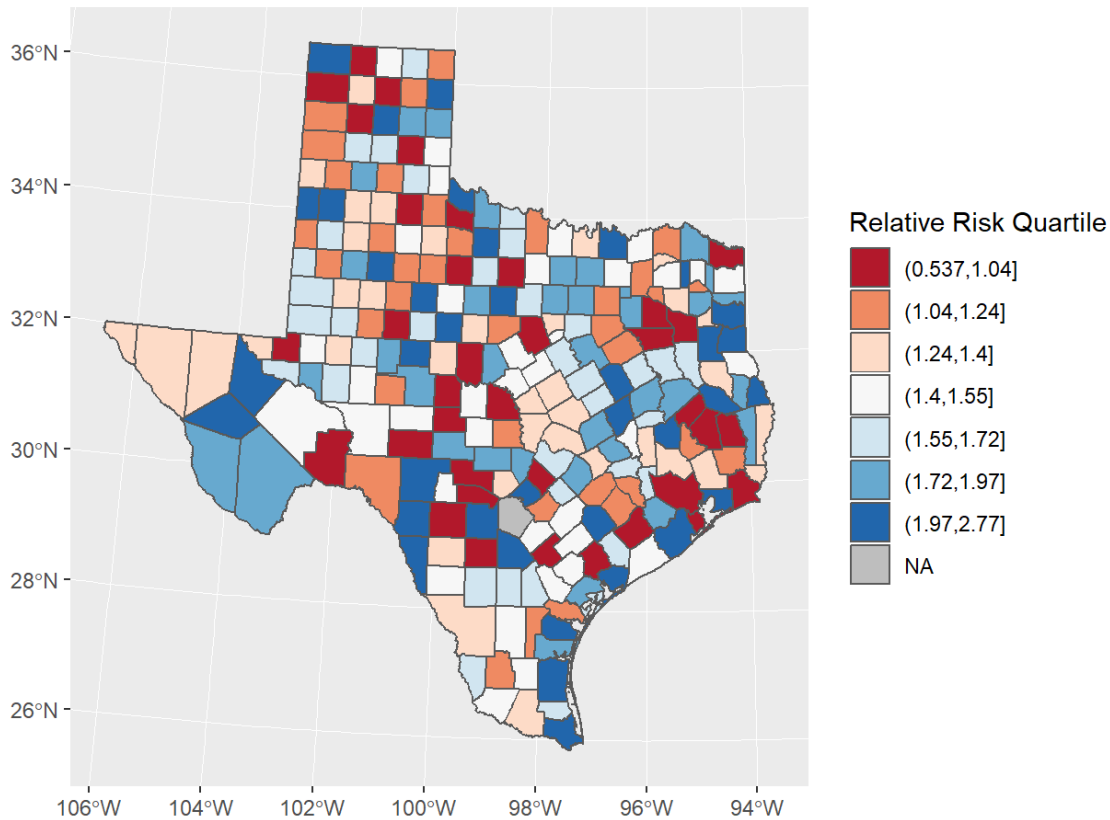
This indicates **very** low spatially correlated variance in these data and very low temporal heterogeneity as well.

Space-time mapping of the fitted values

```
final.dat$fitted_m3<-mod3$summary.fitted.values$mean

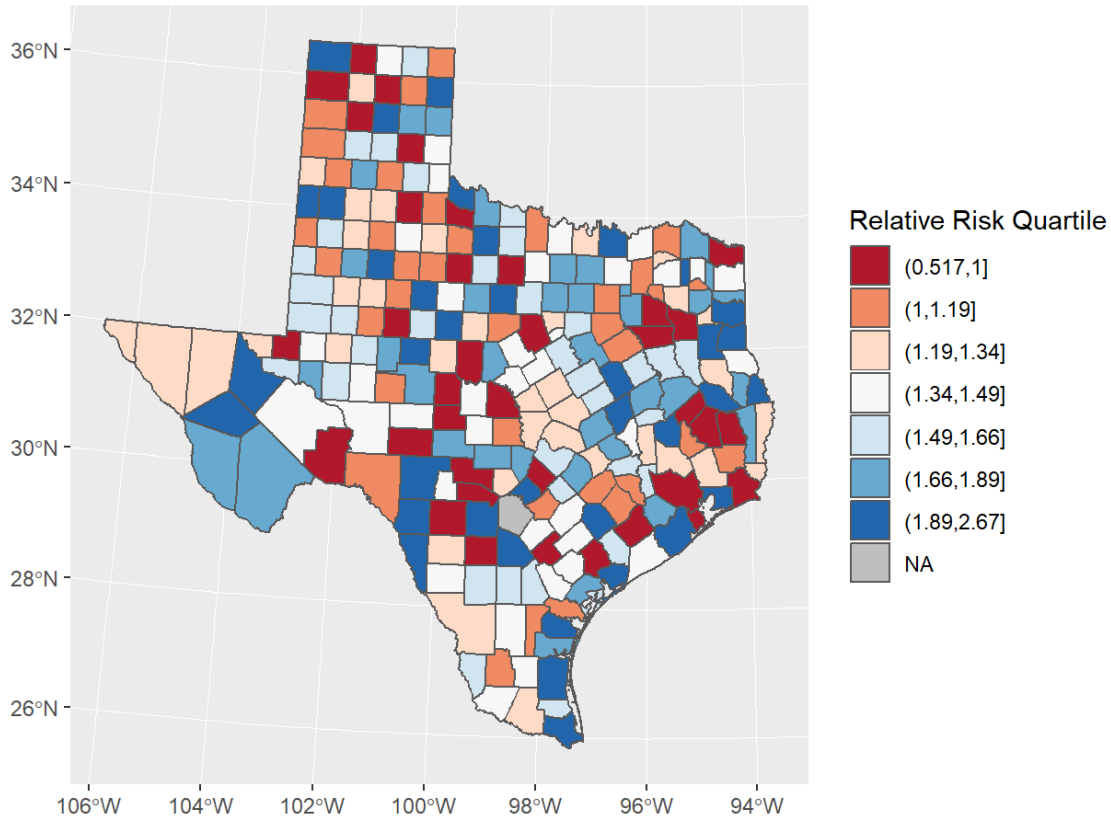
final.dat%>%
  filter(year%in%c(2000))%>%
  mutate(qrr=cut(fitted_m3, breaks = quantile(fitted_m3, p=seq(0,1,length.out = 8))))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "RdBu" )+scale_fill_brewer(palette = "RdBu", na.value="grey")+guides(fill=guide_legend(title="Relative Risk Quartile"))+ggtitle(label="Relative Risk Quartile - IID Model, 2000")+coord_sf(crs = 102008)
```

Relative Risk Quartile - IID Model, 2000



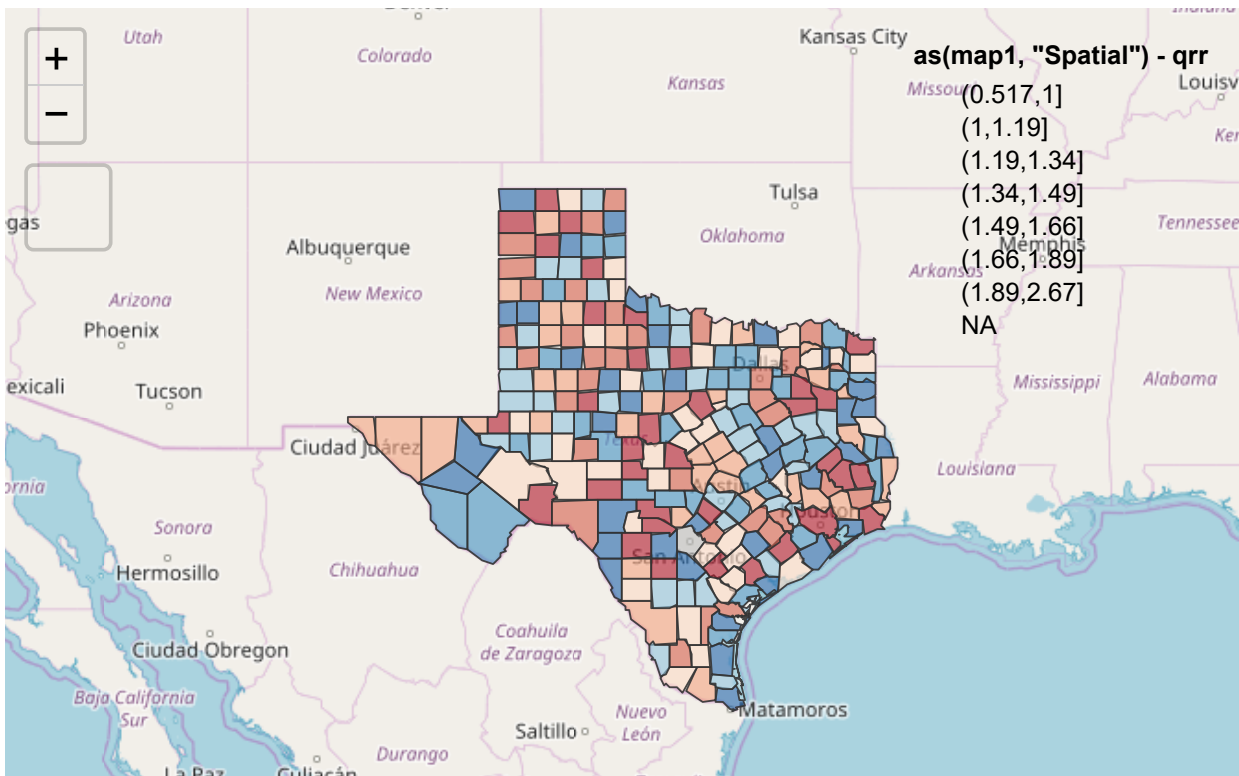
```
final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(fitted_m3, breaks = quantile(fitted_m3, p=seq(0,1,length.out = 8))))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "RdBu" )+scale_fill_brewer(palette = "RdBu", na.value="grey")+guides(fill=guide_legend(title="Relative Risk Quartile"))+ggtitle(label="Relative Risk Quartile - IID Model, 2007")+coord_sf(crs = 102008)
```

Relative Risk Quartile - IID Model, 2007



library(mapview)

```
map1<-final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(fitted_m3, breaks = quantile(fitted_m3, p=seq(0,1,length.out = 8))))
clrs <- colorRampPalette(brewer.pal(8, "RdBu"))
mapView(as(map1, "Spatial"), zcol="qrr", legend=T, col.regions=clrs, map.types="OpenStreetMap")
```



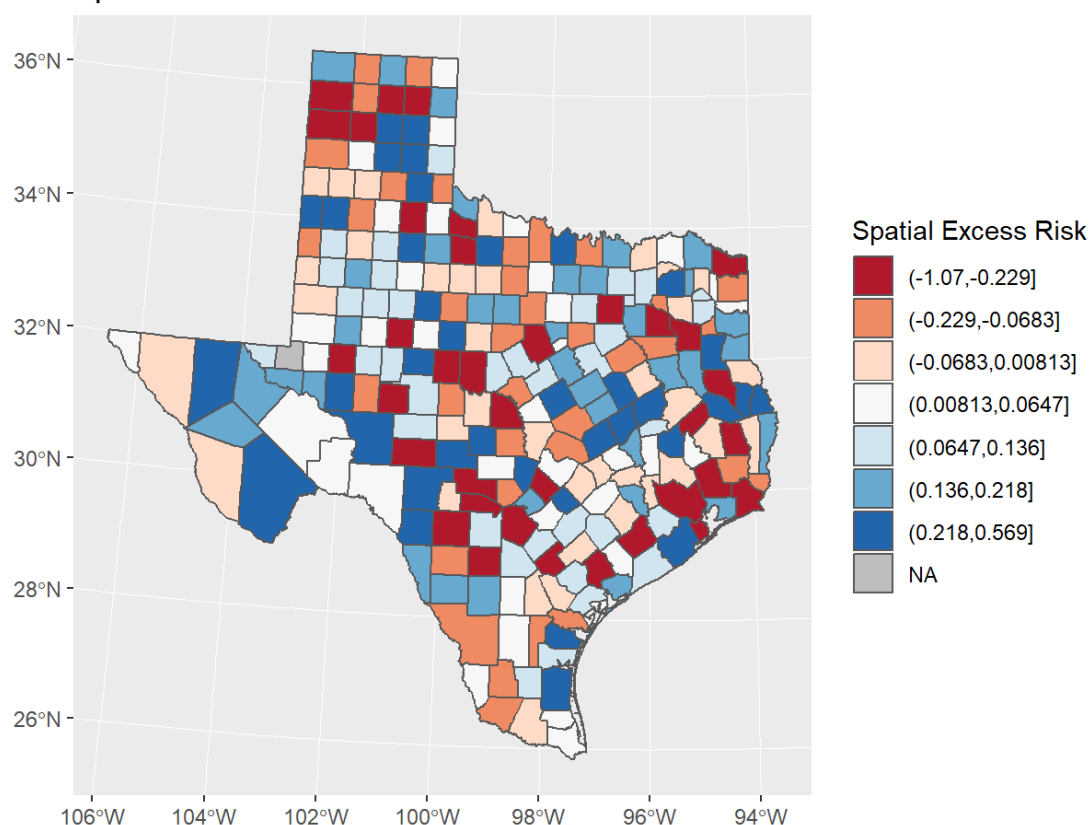


Map of spatial random effects

It is common to map the random effects from the BYM model to look for spatial trends, in this case, there are not strong spatial signals:

```
tx$sp_re<-mod3$summary.random$struct$mean[1:254]
tx%>%
  mutate(qse=cut(sp_re, breaks = quantile(sp_re, p=seq(0,1,length.out = 8))))%>%
  ggplot()+geom_sf(aes(fill=qse))+scale_colour_brewer(palette = "RdBu" )+scale_fill_brewer(palette = "RdBu", na.value="grey")+guides(fill=guide_legend(title="Spatial Excess Risk"))+ggtitle(label="Spatial Random Effect - BYM Model")+coord_sf(crs = 102008)
```

Spatial Random Effect - BYM Model



Exceedence probabilities

In Bayesian spatial models that are centered on an epidemiological type of outcome, it is common to examine the data for spatial clustering. One way to do this is to examine the clustering in the relative risk from one of these GLMM models. For instance if θ is the relative risk

$$\theta = \exp(\beta_0 + \beta_1 * x_1 + u_j)$$

from one of our Negative binomial models above. We can use the posterior marginals of the relative risk to ask $\theta > \theta^*$ where θ^* is a specific level of excess risk, say 50% extra or $\theta > 1.5$. If the density, or $\Pr(\theta > \theta^*)$ is high, then there is evidence that the excess risk is not only high, but **significantly** high.

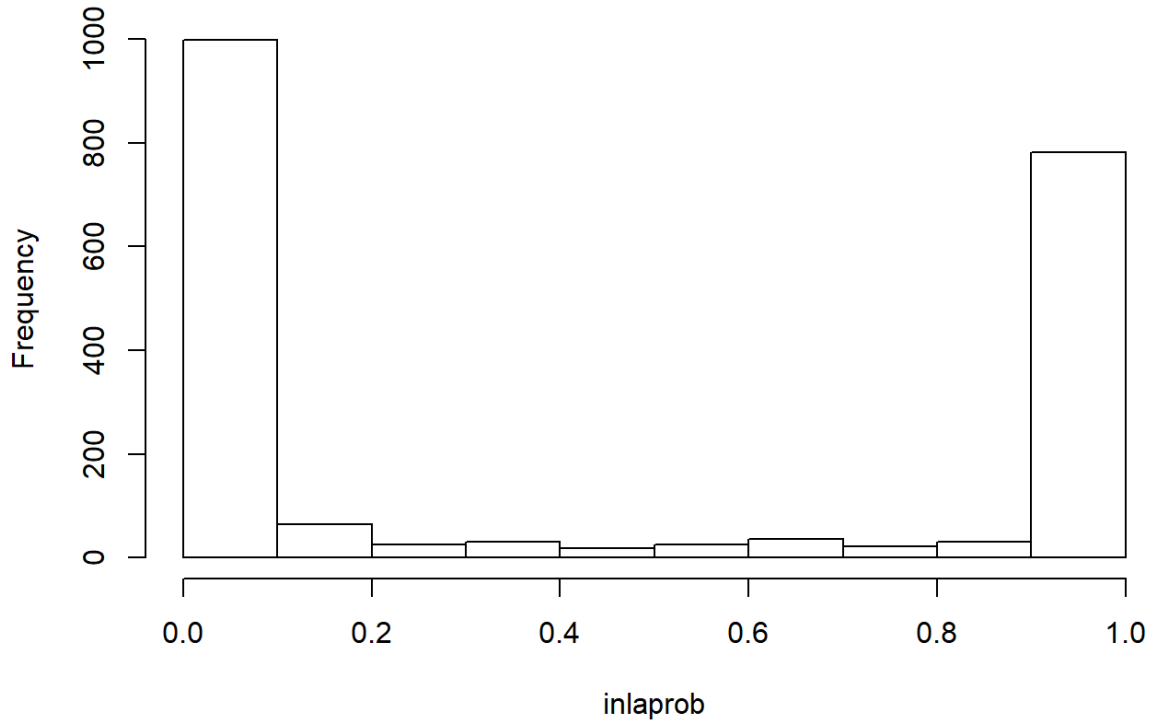
To get the exceedence probabilities from one of our models, we can use the `inla.pmarginal()` function to ask if $\Pr(\theta > \theta^*)$

```

thetastar<-1.5#theta*
inlaprob<- unlist(lapply(mod3$marginals.fitted.values, function(X){
  1-inla.pmarginal(thetastar, X)
}))
hist(inlaprob)

```

Histogram of inlaprob



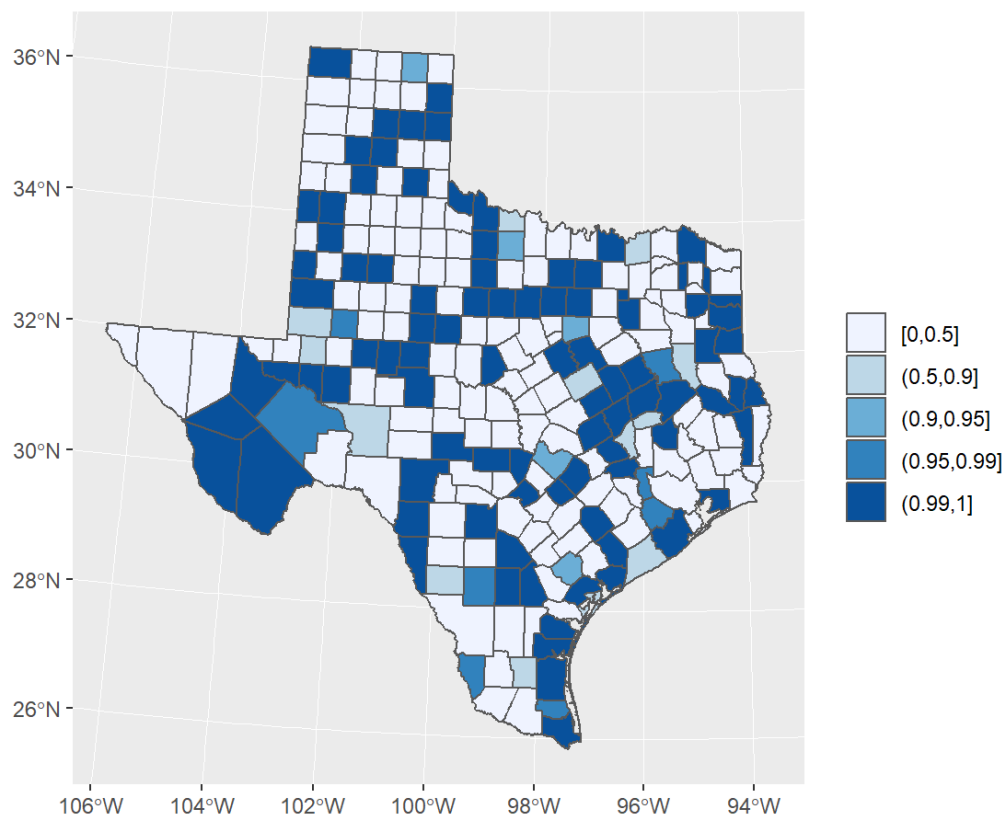
So, we see lots of occasions where the exceedence probability is greater than .9. We can visualize these in a map.

```

final.dat$exceedprob<-inlaprob

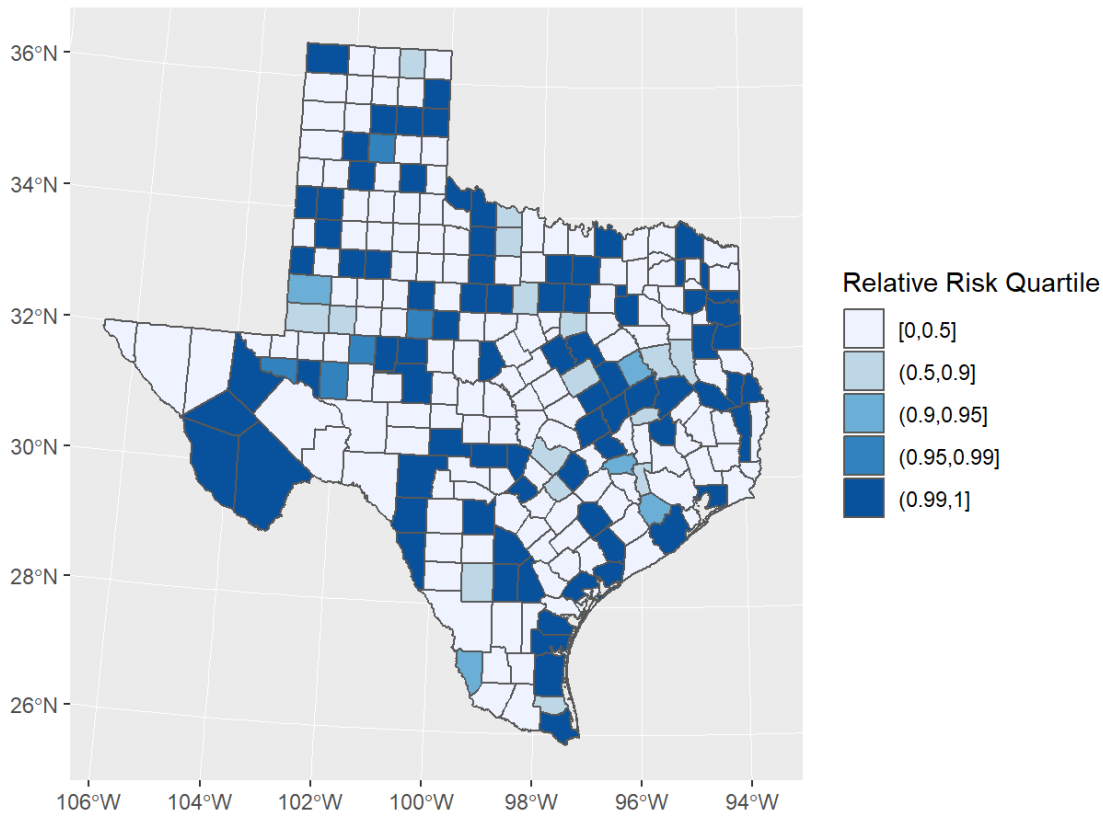
final.dat%>%
  filter(year%in%c(2000))%>%
  mutate(qrr=cut(exceedprob, breaks = c(0, .5, .9, .95, .99, 1), include.lowest = T))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "Blues" )+scale_fill_brewer(palette = "Blues", na.value="grey")+guides(fill=guide_legend(title=""))+ggtitle(label=expression(paste("Exceedence Probability Relative Risk ", "Pr( ",theta," >1.5", " ) - 2000" ) )+coord_sf(crs = 102008)

```


Exceedence Probability Relative Risk $\Pr(\theta > 1.5)$ - 2000

```
final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(exceedprob, breaks = c(0, .5, .9, .95, .99, 1), include.lowest = T))%>%
  ggplot()+geom_sf(aes(fill=qrr))+scale_colour_brewer(palette = "Blues" )+scale_fill_brewer(palette = "Blues", na.value="grey")+guides(fill=guide_legend(title="Relative Risk Quartile"))+ggtitle(label=expression(paste("Exceedence Probability Relative Risk ", "Pr( ", theta, " >1.5", " ) - 2007" ) )+coord_sf(crs = 102008)
```

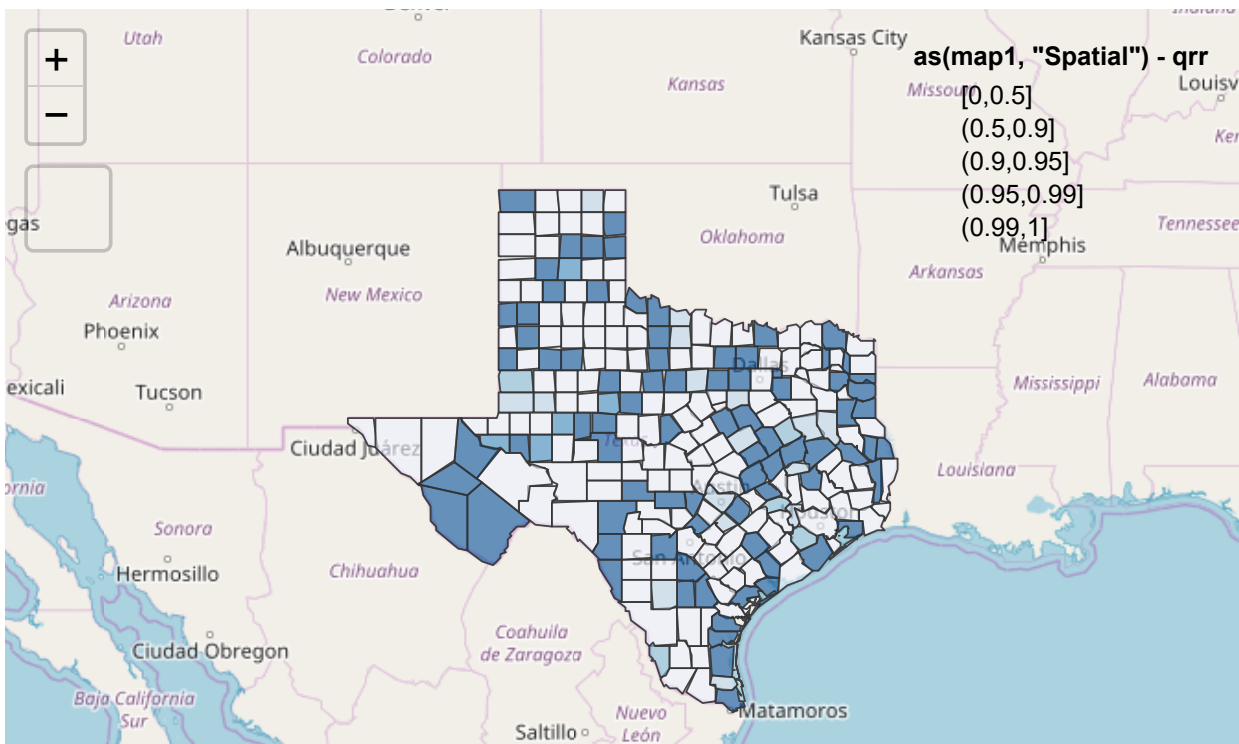
Exceedence Probability Relative Risk $\Pr(\theta > 1.5)$ - 2007



```
library(mapview)
```

```
map1<-final.dat%>%
  filter(year%in%c(2007))%>%
  mutate(qrr=cut(exceedprob, breaks = c(0, .5, .9, .95, .99, 1), include.lowest = T))
```

```
clrs <- colorRampPalette(brewer.pal(6, "Blues"))
mapView(as(map1, "Spatial"), zcol="qrr", legend=T, col.regions=clrs, map.types="OpenStreetMap")
```





Which shows several areas of the state where risk the mortality rate is higher than the state rate.

Multi - Level Models

```

#Load brfss
library(car)
library(knitr)
brfssurl<- "https://github.com/coreysparks/data/blob/master/brfss_14.Rdata?raw=true"
load(url(brfssurl))
set.seed(12345)
#samps<-sample(1:nrow(brfss_14), size = 90000, replace=F)
#brfss_14<-brfss_14[samps,]
#The names in the data are very ugly, so I make them less ugly
nams<-names(brfss_14)
#we see some names are lower case, some are upper and some have a little _ in the first position. This is
a nightmare.
newnames<-gsub(pattern = "x_",replacement = "",x = nams)
names(brfss_14)<-tolower(newnames)
#BMI
brfss_14$bmi<-ifelse(is.na(brfss_14$bmi5)==T, NA, brfss_14$bmi5/100)
brfss_14$obese<-ifelse(brfss_14$bmi>30,1,0)
#Poor or fair self rated health
#brfss_14$badhealth<-ifelse(brfss_14$genhlth %in% c(4,5),1,0)
brfss_14$badhealth<-recode(brfss_14$genhlth, recodes="4:5=1; 1:3=0; else=NA")
#race/ethnicity
brfss_14$black<-recode(brfss_14$racegr3, recodes="2=1; 9=NA; else=0")
brfss_14$white<-recode(brfss_14$racegr3, recodes="1=1; 9=NA; else=0")
brfss_14$other<-recode(brfss_14$racegr3, recodes="3:4=1; 9=NA; else=0")
brfss_14$hispanic<-recode(brfss_14$racegr3, recodes="5=1; 9=NA; else=0")
brfss_14$race_eth<-recode(brfss_14$racegr3, recodes="1='nhwhite'; 2='nh black'; 3='nh other';
4='nh multirace'; 5='hispanic'; else=NA", as.factor = T)
brfss_14$race_eth<-relevel(brfss_14$race_eth, ref = "nhwhite")
#insurance
brfss_14$ins<-ifelse(brfss_14$hlthpln1==1,1,0)

#income grouping
brfss_14$inc<-ifelse(brfss_14$incomg==9, NA, brfss_14$incomg)

#education level
brfss_14$educ<-recode(brfss_14$educa, recodes="1:2='0Prim'; 3='1somehs'; 4='2hsgrad';
5='3somecol'; 6='4colgrad';9=NA", as.factor=T)
#brfss_14$educ<-relevel(brfss_14$educ, ref='0Prim')

#employment
brfss_14$employ<-recode(brfss_14$employ, recodes="1:2='Employed'; 2:6='nilf';
7='retired'; 8='unable'; else=NA", as.factor=T)
brfss_14$employ<-relevel(brfss_14$employ, ref='Employed')

#marital status
brfss_14$marst<-recode(brfss_14$marital, recodes="1='married'; 2='divorced'; 3='widowed';
4='separated'; 5='nm';6='cohab'; else=NA", as.factor=T)
brfss_14$marst<-relevel(brfss_14$marst, ref='married')

#Age cut into intervals
brfss_14$agec<-cut(brfss_14$age80, breaks=c(0,24,39,59,79,99), include.lowest = T)

```

I want to see how many people we have in each MSA in the data:

```

#Now we will begin fitting the multilevel regression model with the msa
#that the person lives in being the higher level
head(data.frame(name=table(brfss_14$mmsaname),id=unique(brfss_14$mmsa)))

```

```
##                                     name.Var1
## 1                Aberdeen, SD, Micropolitan Statistical Area
## 2            Aguadilla-Isabela, PR, Metropolitan Statistical Area
## 3                Albuquerque, NM, Metropolitan Statistical Area
## 4 Allentown-Bethlehem-Easton, PA-NJ, Metropolitan Statistical Area
## 5                Anchorage, AK, Metropolitan Statistical Area
## 6 Atlanta-Sandy Springs-Roswell, GA, Metropolitan Statistical Area
##  name.Freq  id
## 1         620 10100
## 2         544 10380
## 3        1789 10740
## 4         1095 10900
## 5         1785 11260
## 6         2776 12060
```

```
#people within each msa
```

```
#How many total MSAs are in the data?
length(table(brfss_14$mmsa))
```

```
## [1] 132
```

```
#counties
```

Higher level predictors

We will often be interested in factors at both the individual *AND* contextual levels. To illustrate this, I will use data from the American Community Survey measured at the MSA level. Specifically, I use the DP3 table, which provides economic characteristics of places, from the 2010 5 year ACS Link (http://www.census.gov/acs/www/data_documentation/special_data_release/).

```
library(acs)
#Get 2010 ACS median household incomes for tracts in Texas
msaacs<-geo.make(msa="*")

acsecon<-acs.fetch(key=mykey, endyear=2010, span=5, geography=msaacs, variable = c("B19083_001","B17001_001","B17001_002", "B03002_001","B03002_004", "B03002_012" ))

colnames(acsecon@estimate)
```

```
## [1] "B19083_001" "B17001_001" "B17001_002" "B03002_001" "B03002_004"
## [6] "B03002_012"
```

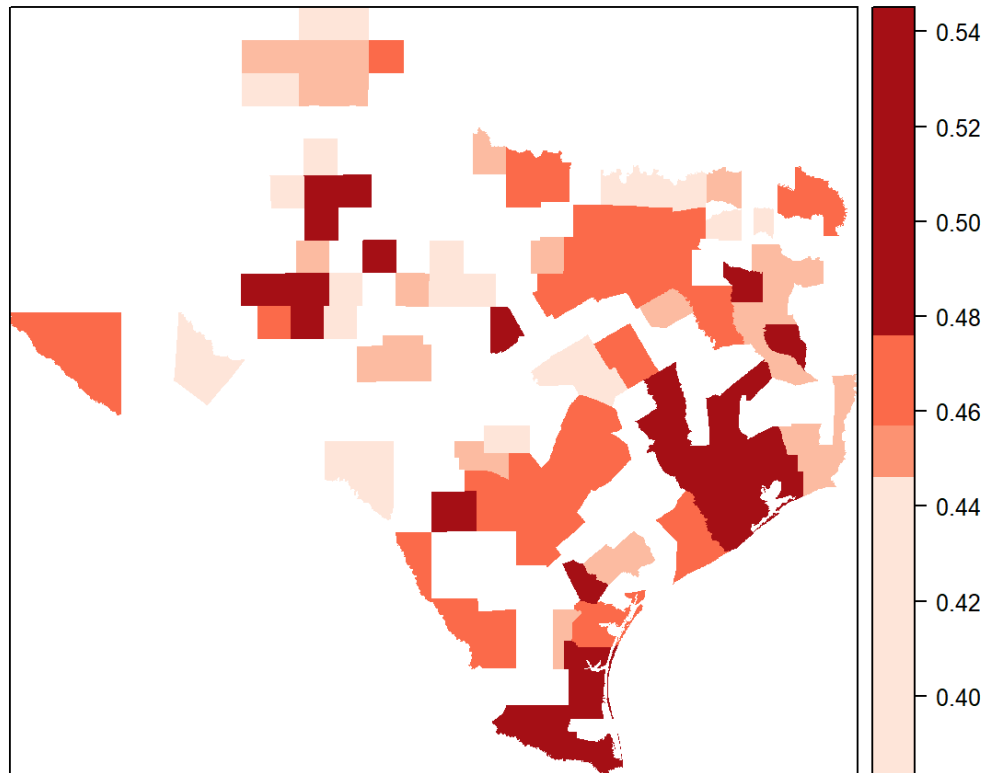
```
msaecon<-data.frame(gini=acsecon@estimate[, "B19083_001"],
ppoverty=acsecon@estimate[, "B17001_002"]/acsecon@estimate[, "B17001_001"],
pblack=acsecon@estimate[, "B03002_004"]/acsecon@estimate[, "B03002_001"],
phisp=acsecon@estimate[, "B03002_012"]/acsecon@estimate[, "B03002_001"],
giniz=scale(acsecon@estimate[, "B19083_001"]),
ppovertyz=scale(acsecon@estimate[, "B17001_002"]/acsecon@estimate[, "B17001_001"]))
msaecon$zblack<-scale(msaecon$pblack)
msaecon$zphisp<-scale(msaecon$phisp)
msaecon$ids<-paste(acsecon@geography$metropolitanstatisticalareamicropolitanstatisticalarea)
```

Let's see the geographic variation in these economic indicators:

```
library(tigris)
msa<-core_based_statistical_areas(cb=T)
msa_ec<-geo_join(msa, msaecon, "CBSAFP", "ids", how="inner")

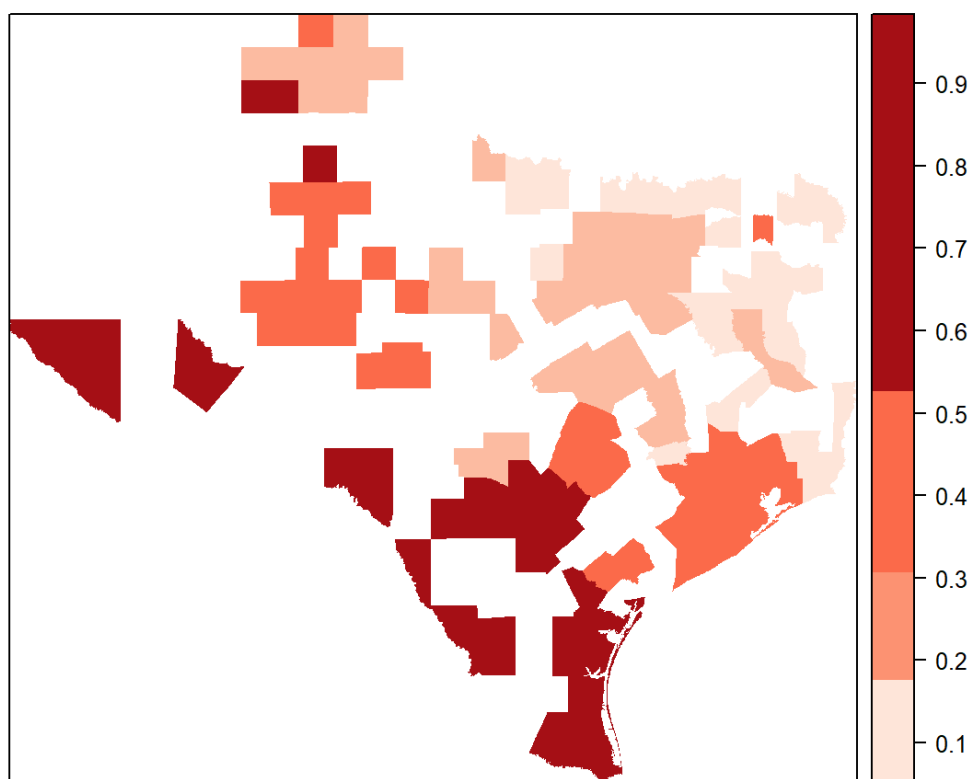
tx_ec<-msa_ec[grep(msa_ec$NAME, pattern = "TX"), ]
library(RColorBrewer)
library(sp)
spplot(tx_ec, "gini", at=quantile(tx_ec$gini), col.regions=brewer.pal(n=6, "Reds"), col="transparent", main="Gini Coefficient")
```

Gini Coefficient



```
spplot(tx_ec, "phisp", at=quantile(tx_ec$phisp), col.regions=brewer.pal(n=6, "Reds"), col="transparent", main="Percent Hispanic")
```

Percent Hispanic



Create spatial information for higher level units

```
#See what counties are in the brfss data
tx_ec$struct<-1:dim(tx_ec)[1]
city.est.dat<-tx_ec@data[,c("giniz","ppovertyz", "zpblack", "zphisp", "struct")]
city.est.dat$obese<-NA
```

```
head(city.est.dat)
```

```
##      giniz  ppovertyz   zpblack   zphisp  struct  obese
## 7  1.6563723  2.68315742 -0.685922045  4.2102849     1    NA
## 10 0.6261542  1.95562295 -0.690319376  4.6557381     2    NA
## 17 -0.2752867  0.38023597 -0.312542768  2.3699550     3    NA
## 24  2.4290359  1.64782952  0.168694184  0.5513964     4    NA
## 28  0.5295712 -0.08529324 -0.244977156  2.3297027     5    NA
## 36  0.7227371 -0.25647600 -0.005019977  0.1616555     6    NA
```

```
brfss_14$cbsa<-as.character(brfss_14$mmsa)
indat<-merge(brfss_14, tx_ec, by.x="cbsa", by.y="CBSAFP", all.x=T)

brf.est<-indat[, c("giniz","ppovertyz", "zpblack", "zphisp", "struct", "obese")]
brf.est<-brf.est[order(brf.est$struct),]
head(brf.est)
```

```
##          giniz ppovertyz  zpblack  zphisp struct obese
## 49956 2.429036  1.64783 0.1686942 0.5513964    4    1
## 49957 2.429036  1.64783 0.1686942 0.5513964    4    1
## 49958 2.429036  1.64783 0.1686942 0.5513964    4    0
## 49959 2.429036  1.64783 0.1686942 0.5513964    4    0
## 49960 2.429036  1.64783 0.1686942 0.5513964    4    1
## 49961 2.429036  1.64783 0.1686942 0.5513964    4    0
```

```
##Here is where I add the cities that need to be estimated to the rest of the data
```

```
m.est<-rbind(city.est.dat, brf.est)
```

```
struct.in<-unique(brf.est$struct)
```

```
m.est$comp<-ifelse(m.est$struct%in%struct.in ,1,0)
```

```
m.est$rm<-ifelse(m.est$comp==1&is.na(m.est$obese)==T,1,0)
```

```
m.est<-m.est[-which(m.est$rm==1),]
```

```
m.est<-m.est[is.na(m.est$struct)==F,]
```

```
m.est<-m.est[order(m.est$struct),]
```

```
#
```

```
# fake_dat<-expand.grid(race_eth=Levels(brfss_14$race_eth), agec=Levels(brfss_14$agec), CBSAFP=Levels(as.factor(tx_ec$CBSAFP) ))
```

```
# fake_dat<-merge(fake_dat, tx_ec, by="CBSAFP")
```

```
library(spdep)
```

```
nbs<-knearneigh(coordinates(tx_ec), longlat = T, k = 4)
```

```
nbs<-knn2nb(nbs, row.names = tx_ec$struct, sym = T)
```

```
mat <- nb2mat(nbs, style="B",zero.policy=TRUE)
```

```
colnames(mat) <- rownames(mat)
```

```
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
```

```
fit_est<- inla(obese~ giniz+zpblack+zphisp+
              f(struct, model="bym", graph=mat,constr=TRUE, scale.model=TRUE),
              family = "binomial",Ntrials = 1,
              data=m.est, num.threads = 2,
              control.predictor = list(link=1))
#              control.inla=list(strategy='gaussian'))
#
summary(fit_est)
```



```

##
## Call:
## c("inla(formula = obese ~ giniz + zpblack + zp hisp + f(struct, model = \"bym\", \" \" graph = mat,
## constr = TRUE, scale.model = TRUE), family = \"binomial\", \" \" data = m.est, Ntrials = 1, control.pr
## edictor = list(link = 1), \" \" num.threads = 2)")
##
## Time used:
## Pre-processing Running inla Post-processing Total
## 1.3010 252.3749 0.4299 254.1058
##
## Fixed effects:
## mean sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept) -1.0249 0.1523 -1.3008 -1.0342 -0.6955 -1.0497 0
## giniz 0.0827 0.0940 -0.1133 0.0858 0.2595 0.0907 0
## zpblack -0.0280 0.1947 -0.4391 -0.0214 0.3465 -0.0113 0
## zp hisp 0.0882 0.0608 -0.0397 0.0903 0.2039 0.0937 0
##
## Random effects:
## Name Model
## struct BYM model
##
## Model hyperparameters:
## mean sd 0.025quant
## Precision for struct (iid component) 394.55 1047.49 16.44
## Precision for struct (spatial component) 2131.41 2414.89 129.05
## 0.5quant 0.975quant mode
## Precision for struct (iid component) 147.89 2296.16 39.30
## Precision for struct (spatial component) 1393.44 8492.95 342.89
##
## Expected number of effective parameters(std dev): 5.788(0.6795)
## Number of equivalent replicates : 1450.66
##
## Marginal log-Likelihood: -5125.44
## Posterior marginals for linear predictor and fitted values computed

```

```

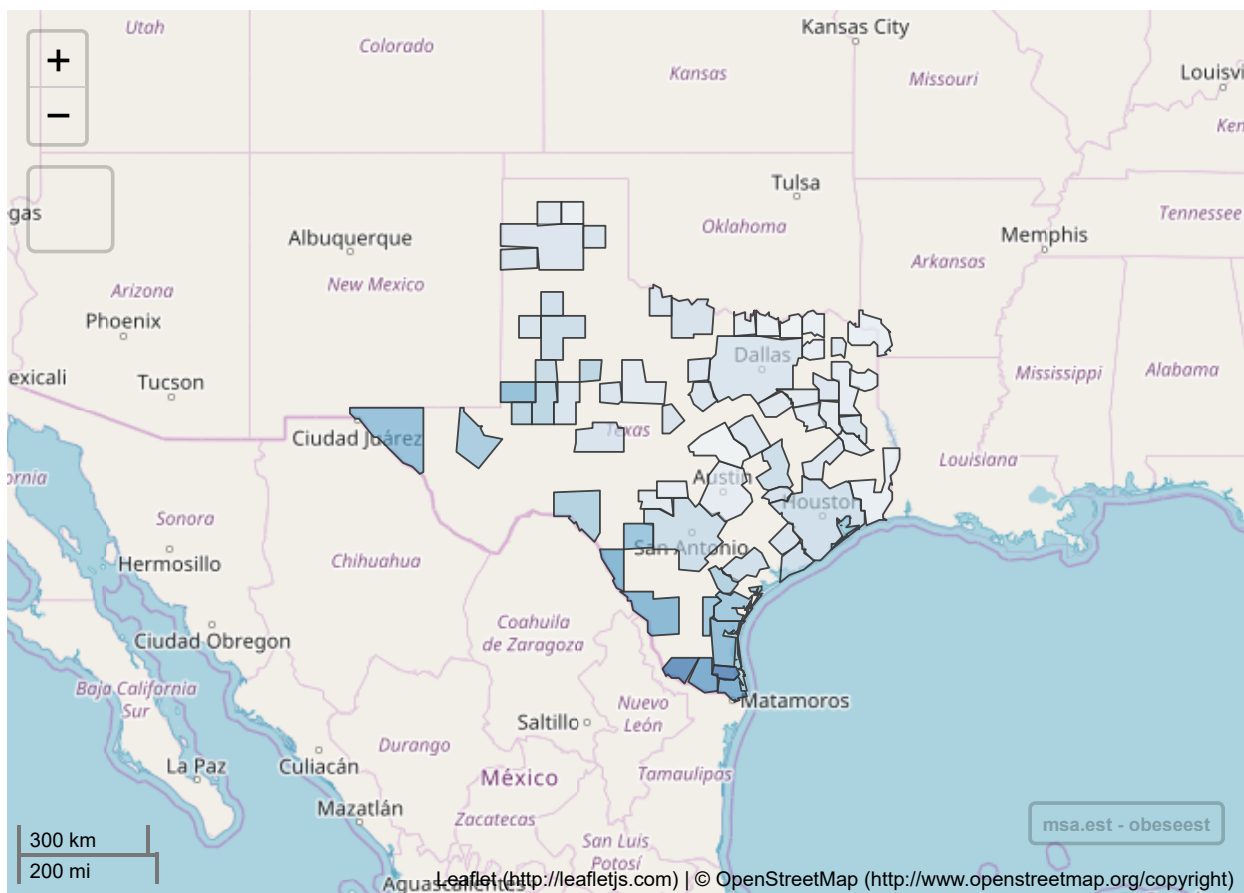
m.est$est.obese<-fit_est$summary.fitted.values$mean
m.est.est<-tapply(m.est$est.obese, m.est$struct, mean, na.rm=T)
m.est.est<-data.frame(struct=names(unlist(m.est.est)), obeseest=unlist(m.est.est))
#m.est<-m.est[is.na(m.est$bmi)==T,]
msa.est<-merge(tx_ec, m.est.est, by.y="struct", by.x="struct", all.x=T, sort=F)
head(msa.est@data)

```

```
## struct CSAFP CBSAFP      AFFGEOID GEOID          NAME
## 1      1   154  15180 310M300US15180 15180      Brownsville-Harlingen, TX
## 2      2 <NA>  29700 310M300US29700 29700          Laredo, TX
## 3      3 <NA>  38380 310M300US38380 38380          Plainview, TX
## 4      4 <NA>  17780 310M300US17780 17780      College Station-Bryan, TX
## 5      5 <NA>  41700 310M300US41700 41700 San Antonio-New Braunfels, TX
## 6      6 <NA>  48660 310M300US48660 48660          Wichita Falls, TX
## LSAD      ALAND      AWATER  gini  ppoverty      pblack      phisp
## 1  M1  2307891059 998035120 0.492 0.3473913 0.002924541 0.8741380
## 2  M1  8706186702 36555096 0.460 0.2976883 0.002346617 0.9544532
## 3  M2  2602115387  246678 0.432 0.1900627 0.051996338 0.5423268
## 4  M1  5439580277 85909281 0.516 0.2766608 0.115243452 0.2144409
## 5  M1  18939926777 150513456 0.457 0.1582592 0.060876225 0.5350693
## 6  M1  6784786637 144176905 0.463 0.1465645 0.092412870 0.1441707
##      giniz  ppovertyz      zpblack  zpblack  ids obeseest
## 1  1.6563723 2.68315742 -0.685922045 4.2102849 15180 0.3837053
## 2  0.6261542 1.95562295 -0.690319376 4.6557381 29700 0.3698816
## 3 -0.2752867 0.38023597 -0.312542768 2.3699550 38380 0.3041471
## 4  2.4290359 1.64782952  0.168694184 0.5513964 17780 0.3058289
## 5  0.5295712 -0.08529324 -0.244977156 2.3297027 41700 0.3056169
## 6  0.7227371 -0.25647600 -0.005019977 0.1616555 48660 0.2970790
```

```
library(mapview)
clrs <- colorRampPalette(brewer.pal(8, "Blues"))

mapview(msa.est,"obeseest", col.regions=clrs, map.types="OpenStreetMap")
```



Multi-level model in INLA

very good at doing multi-level models, and much faster than other modeling strategies. Plus we can build in the random effects, such as the Besag model at the higher level of analysis. Below, we fit three models. A basic intercept model: This corresponds to a multilevel logistic model with a higher level variables as predictors and can be

$$\text{logit}(y_{ij}) = \beta_{0j} + \sum \beta x_i$$

$$\beta_{0j} = \beta_0 + u_j$$

$$u_j \sim N(0, \tau_u)$$

with

```
library(INLA)
indat<-indat[is.na(indat$struct)==F,]
fit_in1<- inla(obese~ race_eth+agec+f(struct, model="iid"),
              family = "binomial", Ntrials =1,
              data=indat, num.threads = 2)
summary(fit_in1)
```

```

##
## Call:
## c("inla(formula = obese ~ race_eth + agec + f(struct, model = \"iid\"), ", " family = \"binomial
\", data = indat, Ntrials = 1, num.threads = 2)" )
##
## Time used:
## Pre-processing      Running inla Post-processing      Total
##           1.4176           3.6431           0.0527           5.1133
##
## Fixed effects:
##              mean      sd 0.025quant 0.5quant 0.975quant  mode
## (Intercept)  -2.0850 0.1451  -2.3750  -2.0834  -1.8043  -2.0800
## race_ethhispanic  0.5357 0.0577   0.4221   0.5358   0.6486  0.5360
## race_ethnh black  0.8600 0.0896   0.6838   0.8601   1.0357  0.8602
## race_ethnh multirace 0.2368 0.2268  -0.2188   0.2404   0.6727  0.2475
## race_ethnh other  -0.6385 0.1809  -1.0046  -0.6346  -0.2938 -0.6268
## agec(24,39)    0.8669 0.1512   0.5756   0.8650   1.1691  0.8612
## agec(39,59)    1.2868 0.1435   1.0114   1.2846   1.5748  1.2802
## agec(59,79)    1.1912 0.1437   0.9154   1.1890   1.4797  1.1847
## agec(79,99)    0.3099 0.1701  -0.0208   0.3088   0.6466  0.3066
##
##              kld
## (Intercept)      0
## race_ethhispanic  0
## race_ethnh black  0
## race_ethnh multirace 0
## race_ethnh other  0
## agec(24,39)      0
## agec(39,59)      0
## agec(59,79)      0
## agec(79,99)      0
##
## Random effects:
## Name      Model
## struct    IID model
##
## Model hyperparameters:
##              mean      sd 0.025quant 0.5quant 0.975quant
## Precision for struct 10259.58 16726.18   41.14 2168.30 59222.95
##              mode
## Precision for struct 69.50
##
## Expected number of effective parameters(std dev): 10.13(1.511)
## Number of equivalent replicates : 829.10
##
## Marginal log-Likelihood: -5012.48

```

```
1/fit_in1$summary.hyperpar
```

```

##              mean      sd 0.025quant 0.5quant
## Precision for struct 0.00009746986 0.00005978651 0.02431007 0.0004611908
##              0.975quant  mode
## Precision for struct 0.00001688535 0.01438795

```

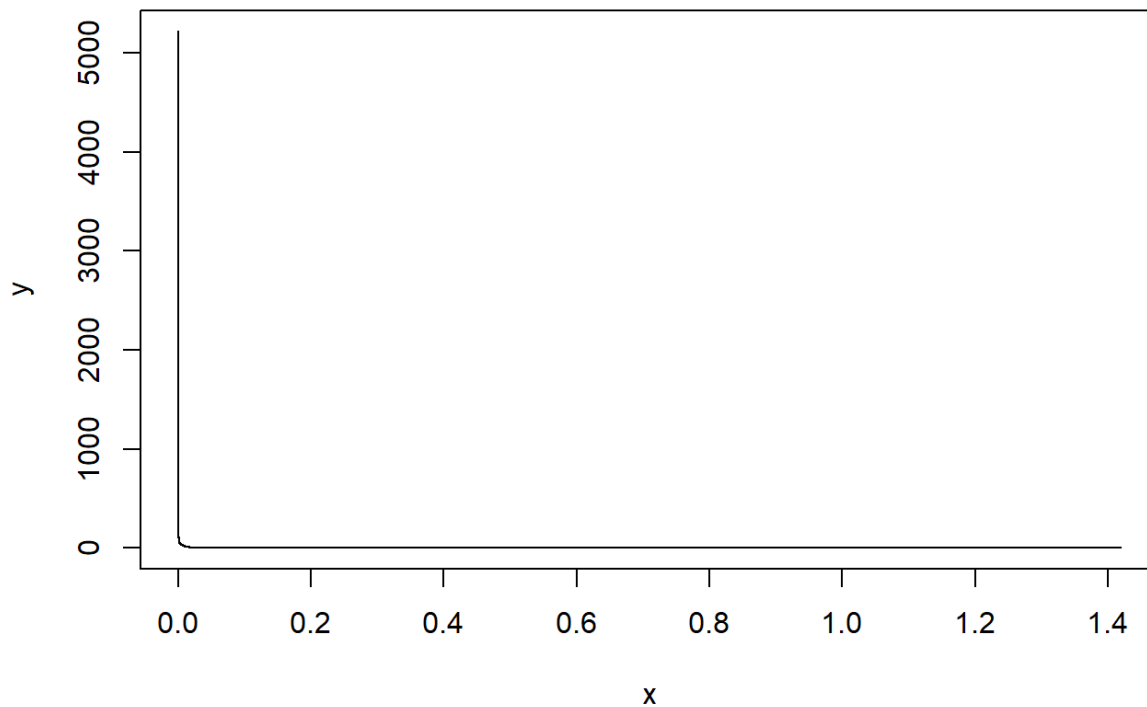
```
m<- inla.tmarginal(
  function(x) (1/x),
  fit_in1$marginals.hyperpar$`Precision for struct`)

inla.hpdmarginal(.95, marginal=m)
```

```
##                low        high
## level:0.95 0.000007273682 0.02080729
```

```
plot(m, type="l", main=c("Posterior distribution for between MSA variance", "- Random Intercept model -"))
```

Posterior distribution for between MSA variance - Random Intercept model -



That is our individual level, random intercept model. Now I will fit a model that includes MSA level demographic characteristics, the Gini index, %black and %hispanic. This corresponds to a multi-level model with higher level predictors:

$$y_{ij} = \beta_{0j} + \sum \beta x_i + \sum \gamma z_j$$

$$\beta_{0j} = \beta_0 + \sum \gamma z_j + u_j$$

with

$$u_j \sim N(0, \tau_u)$$

```
fit_in2<- inla(obese~ race_eth+agec+giniz+zpblack+zphis+f(struct, model="iid"),
  data=indat,
  family="binomial", Ntrials = 1)
summary(fit_in2)
```



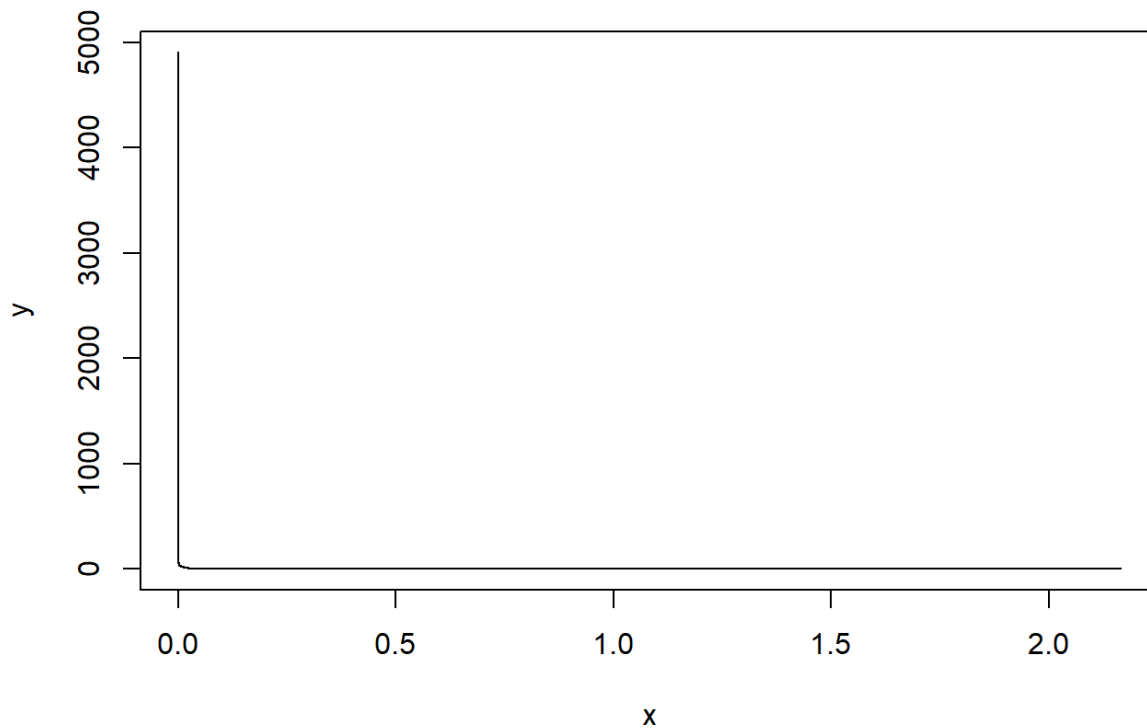
```
m2<- inla.tmarginal(
  function(x) (1/x),
  fit_in2$marginals.hyperpar$`Precision for struct`)

inla.hpdmarginal(.95, marginal=m2)
```

```
##                low        high
## level:0.95 0.000008128605 0.04459525
```

```
plot(m2, type="l", main=c("Posterior distribution for between MSA variance", "- Multi-level model -"))
```

Posterior distribution for between MSA variance - Multi-level model -



Finally, we model the county level means using the Besag-York and Mollie model that we had used on the areal data last week. In this example, we are modeling the correlation between MSAs at the higher level, instead of like last week, where we were modeling the correlation among neighboring counties.

This is a multi level model with correlation between neighboring MSAs, assuming a Besag, York and Mollie, 1991 (<https://link.springer.com/article/10.1007/BF00116466>) convolution prior for the random effects

$$y_{ij} = \beta_0 + \sum \beta x_i + \sum \gamma z_j + u_j + v_j$$

$$\beta_{0j} = \beta_0 + \sum \gamma z_j + u_j + v_j$$

with

$$u_j \sim N(0, \tau_u)$$

$$v_j | v_{\neq j}, \sim \text{Normal}\left(\frac{1}{n_i} \sum_{i \sim j} v_j, \frac{1}{n_i \tau}\right)$$

```
fit_in3<- inla(obese~ race_eth+agec+giniz+zpblack+zphisp+
              f(struct, model="bym", graph=mat),
              family = "binomial", Ntrials = 1,
              data=indat,
              control.predictor = list(link=1))

summary(fit_in3)
```



```
##
## Call:
## c("inla(formula = obese ~ race_eth + agec + giniz + zpblack + zpblack + zpblack + zpblack + zpblack + ", " f(struct, model = \"by
m\", graph = mat), family = \"binomial\", ", " data = indat, Ntrials = 1, control.predictor = list(li
nk = 1))" )
##
## Time used:
## Pre-processing      Running inla Post-processing           Total
##           1.7982          330.3397            0.6054          332.7433
##
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant      mode
## (Intercept) -2.1123 0.2205 -2.5332 -2.1169 -1.6625 -2.1243
## race_ethhispanic  0.5265 0.0609  0.4069  0.5265  0.6460  0.5265
## race_ethnh black  0.8589 0.0900  0.6820  0.8589  1.0352  0.8590
## race_ethnh multirace 0.2387 0.2270 -0.2172  0.2422  0.6748  0.2493
## race_ethnh other -0.6370 0.1809 -1.0033 -0.6331 -0.2921 -0.6253
## agec(24,39)  0.8720 0.1512  0.5806  0.8701  1.1742  0.8662
## agec(39,59)  1.2801 0.1435  1.0047  1.2779  1.5682  1.2736
## agec(59,79)  1.1772 0.1438  0.9012  1.1751  1.4659  1.1707
## agec(79,99)  0.2839 0.1702 -0.0470  0.2827  0.6209  0.2805
## giniz        0.0752 0.1055 -0.1448  0.0785  0.2757  0.0838
## zpblack      -0.0960 0.2227 -0.5681 -0.0877  0.3344 -0.0740
## zpblack      -0.0078 0.0693 -0.1534 -0.0053  0.1250 -0.0010
##
##           kld
## (Intercept)      0
## race_ethhispanic  0
## race_ethnh black  0
## race_ethnh multirace 0
## race_ethnh other  0
## agec(24,39)      0
## agec(39,59)      0
## agec(59,79)      0
## agec(79,99)      0
## giniz            0
## zpblack          0
## zpblack          0
##
## Random effects:
## Name      Model
## struct    BYM model
##
## Model hyperparameters:
##           mean      sd 0.025quant
## Precision for struct (iid component)  153.85 223.49  13.51
## Precision for struct (spatial component) 2450.82 2819.58  171.57
##           0.5quant 0.975quant      mode
## Precision for struct (iid component)   87.86  699.83  34.26
## Precision for struct (spatial component) 1595.63  9844.56 464.10
##
## Expected number of effective parameters(std dev): 13.92(0.6688)
## Number of equivalent replicates : 603.12
##
## Marginal log-Likelihood: -5010.08
## Posterior marginals for linear predictor and fitted values computed
```

```
1/fit_in3$summary.hyperpar
```

```
##
##           mean      sd
## Precision for struct (iid component) 0.0064998242 0.0044745170
## Precision for struct (spatial component) 0.0004080259 0.0003546625
##           0.025quant    0.5quant
## Precision for struct (iid component) 0.074045321 0.0113817337
## Precision for struct (spatial component) 0.005828483 0.0006267101
##           0.975quant      mode
## Precision for struct (iid component) 0.0014289183 0.029186839
## Precision for struct (spatial component) 0.0001015789 0.002154692
```

```
m3_sp<- inla.tmarginal(
  function(x) (1/x),
  fit_in3$marginals.hyperpar$`Precision for struct (spatial component)` )
m3_iid<- inla.tmarginal(
  function(x) (1/x),
  fit_in3$marginals.hyperpar$`Precision for struct (iid component)` )

inla.hpdmarginal(.95, marginal=m3_sp)
```

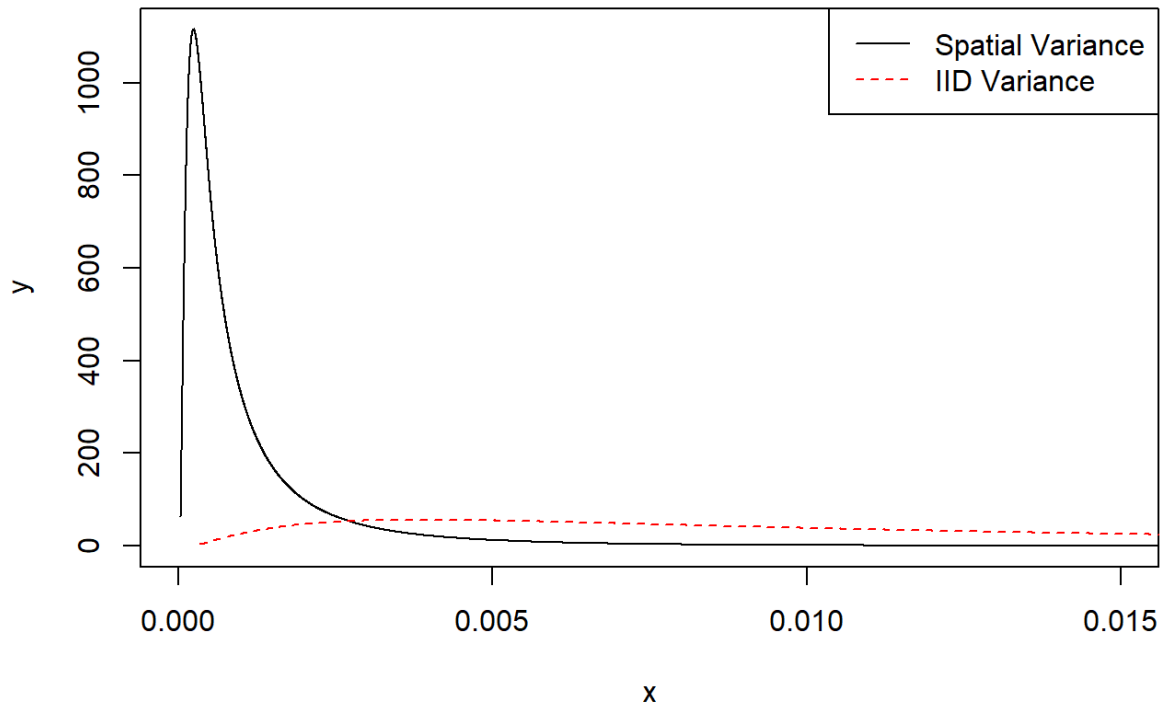
```
##           low      high
## level:0.95 0.00003208837 0.003969693
```

```
inla.hpdmarginal(.95, marginal=m3_iid)
```

```
##           low      high
## level:0.95 0.0003350537 0.05414296
```

```
plot(m3_sp, type="l", main=c("Posterior distribution for between Spatial MSA variance", "- Multi-level model -"), xlim=c(0, .015))
lines(m3_iid, col=2,lty=2)
legend("topright", legend=c("Spatial Variance", "IID Variance"), col=c(1,2), lty=c(1,2))
```

Posterior distribution for between Spatial MSA variance - Multi-level model -



References

Besag, J., York, J., & Mollie, a. (1991). Bayesian image-restoration, with 2 applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1), 1-20. <https://doi.org/10.1007/BF00116466> (<https://doi.org/10.1007/BF00116466>)