```
data {
  int Nobs;
  int Npreds;
  int Ngroups;
  vector[Nobs] y;
  matrix[Nobs,Npreds] x;
  vector[Nobs] timevar;
  int group[Nobs];
  matrix[Ngroups,Ngroups] d;
}

transformed data {
vector[Ngroups] mu;
for (i in 1:Ngroups) mu[i] = 0;
}

parameters {
  vector[Npreds] beta;
  real sigmamap;
  real sigmaeps;
  real eta_sq;
  real rho_sq;
  vector[Ngroups] v;
  vector[Ngroups] etamap;
}

model {

  vector[Ngroups] ranint;
  vector[Nobs] yhat;
  matrix[Ngroups,Ngroups] Sigma;
  for (i in 1:Ngroups){
    for (j in 1:Ngroups)
      ////exponential distance
      {Sigma[i,j] = eta_sq * exp(-rho_sq*(d[i,j]));}}

  etamap ~ normal(0, 1);
  eta_sq ~ normal(0.8, 0.1); #half normal due to constraint lower=0
  rho_sq ~ normal(0.03, 0.01);
  v ~ multi_normal(mu,Sigma);

  ranint  = sigmamap * etamap;

  for (i in 1:Nobs)
    yhat[i] = x[i]*beta+ranint[group[i]]+v[group[i]];

  y ~ normal(yhat, sigmaeps);
}

#this is whhat we need to get the posterior predictive distribution

generated quantities{
  real y_rep[Nobs];
  for(n in 1:Nobs)
    y_rep[n] = normal_rng(y[n],sigmaeps);
```

```
}
```