Homework 1

Hyeonki Seo

September 2021

1-(a)

Linear function of multivariate normal random variable follows multivariate normal distribution. Thus, linear combination of$Z \sim N_p(0, I)$ also follows multivariate normal distribution. Also, Multivariate normal distribution is characterize by its mean and covariance matrix. So, if it can be proved that $Y = \mu + LZ$ by showing its expectation and covariance is same as μ, Σ

$$\begin{aligned}
E[Y] &= E[\mu + LZ] \\
&= \mu + E[LZ] \\
&= \mu + LE[Z] \\
&= \mu + 0 \qquad\qquad (E[Z] = 0) \\
&= \mu
\end{aligned}$$

$$\begin{aligned}
Var(Y) &= Var(\mu + LZ) \\
&= LVar(Z)L' \\
&= LL' \qquad\qquad (Var(Z) = I) \\
&= \Sigma
\end{aligned}$$

$$Y \sim MVN(\mu, \Sigma)$$

1-(b)

I made a function 'ysampgen'. This function needs 3 parameters which are mu, sigma and seed. It works in 4 steps. (1) It decompose sigma using Cholesky decomposition to get L. (2) Set seed to regenerate samples. (3) Standard normal random variables are sampled to get Z (4) Linear combination of μ, L and Z ($\mu + LZ$)is calculated to get Y . Consequently, multivariate normal samples with mean μ, covariance Σ is sampled . Codes is given below

```
x <- seq(0, 1, length = 500) # fine grid
d <- as.matrix(dist(x)) # create distance matrix
sigma1 <-  Matern(d, range = 1, nu = 0.5) # make covariance matrix
mu1 <-  rep(0,500) # make mu vector

y_samp_gen <- function(mu, sigma,seed){
  L <-  t(chol(sigma))
  set.seed(seed)
  Z <-  rnorm(dim(sigma)[1], mean = 0, sd = 1)
  y <- mu + L %*% Z
  return(y)
}
```

1-(c)

At first, I changed Σ. there are 2 ways to change covariance matrix. One is changing range(ρ). the other is changing smoothparameter($=\nu$). So I changed range from 1 to 0.5, changed ν from 0.5 to 2. After that I changed μ from 0 to 0.5. codes and results are below

```
# change range
sigma1 <- Matern(d, range = 1, nu = 0.5) # same as exp
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

sigma2 <- Matern(d, range = 0.5, nu = 0.5) # change range
y2 <- y_samp_gen(mu = mu1, sigma = sigma2, seed = 2021)

ylim <- range(c(y1, y2))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(rho*"=1"))
matplot(x, y2, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(rho*"=0.5"))
# scale becomes bigger
```

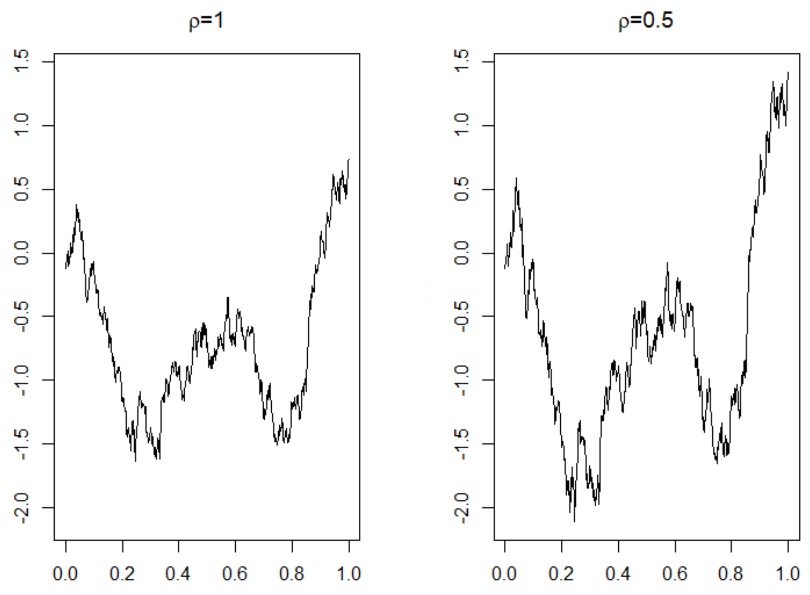


Figure 1 : Change range parameter

In Figure 1, plot in left panel has $\rho = 1$ and right panel has $\rho = 0.5$. Other parameters are same as $\mu = 0, \nu = 0.5$. By this result, It is shown that the lower range value Σ has, the larger oscillation samples have while the shape is maintained.

```r
# change nu
sigma1 <- Matern(d, range = 1, nu = 0.5) # same as exp
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

sigma3 <- Matern(d, range = 1, nu = 2) # change range
y3 <- y_samp_gen(mu = mu1, sigma = sigma3, seed = 2021)

ylim <- range(c(y1, y3))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(nu*"=0.5"))
matplot(x, y3, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(nu*"=2"))
```

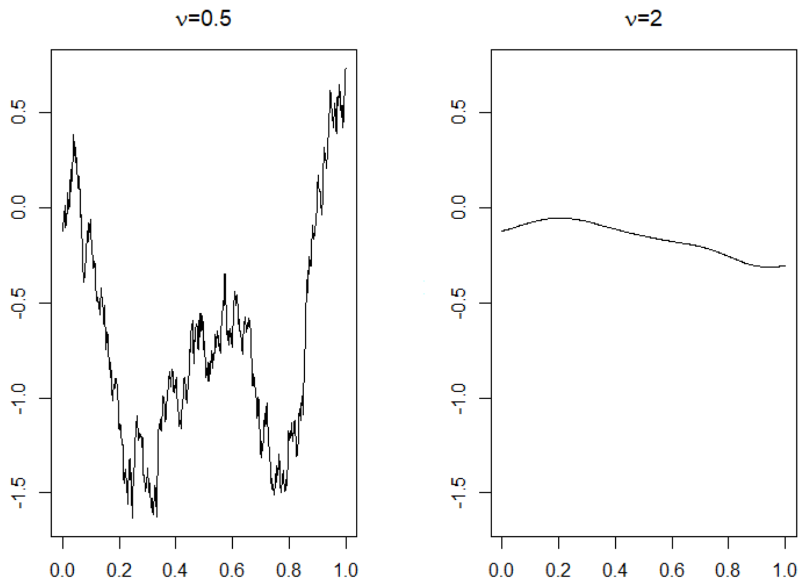


Figure 2 : Change smoothness parameter

In Figure 2, plot in left panel has $\nu = 0.5$ and right panel has $\nu = 2$. Other parameters are same as $\mu = 0, \rho = 1$. By this result, It is shown that the bigger smoothness value Σ has, the more flexible values are sampled.

```r
# change mu
mu1 <-  rep(0,500) # make mu vector
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

mu2 <-  rep(0.5,500)
y4 <- y_samp_gen(mu = mu2, sigma = sigma1, seed = 2021)R

ylim <- range(c(y1, y4))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(mu*"=0"))
matplot(x, y4, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(mu*"=0.5"))
```

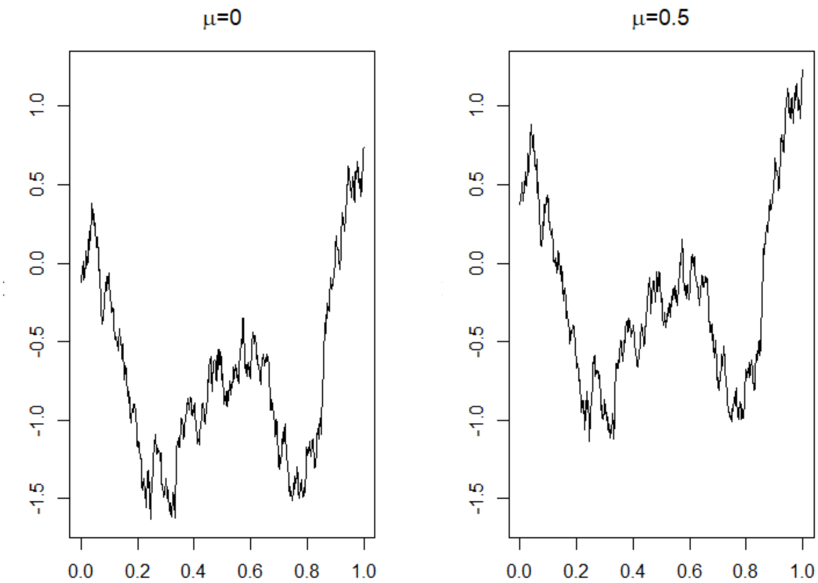


Figure 3 : Change μ

In Figure 3, plot in left panel has $\mu = 0$ and right panel has $\mu = 0.5$. Other parameters are same as $\nu = 0.5, \rho = 1$. By this result, It is shown that μ has an effect on sample's location.

2-(a)

The Average temperatures in CAtemp data are regressed on three covariates(X_1 : lattitude, X_2 : longitude, X_3 : elevation) and intercept by Ordinary Least Square method. However, this result can only take a role of preliminary estimate of β because it did not include spatial correlation among locations. Thus, we have to check residuals if they are having spatial correlation. The result is given below

$$Y = 321.51 + 2.324X_1 + 0.565X_2 - 0.01X_3$$

```
> linmod <- lm(avgtemp ~ lon + lat + elevation, data = CAtemp)
> summary(linmod)

Call:
lm(formula = avgtemp ~ lon + lat + elevation, data = CAtemp)

Residuals:
   Min     1Q Median     3Q    Max 
-6.304 -1.780  0.082  1.687  6.954 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.215e+02  1.602e+01   20.06  < 2e-16 ***
lon          2.324e+00  1.736e-01   13.39  < 2e-16 ***
lat          5.647e-01  1.586e-01    3.56 0.000465 ***
elevation   -9.648e-03  3.923e-04  -24.59  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.583 on 196 degrees of freedom
Multiple R-squared:  0.853,	Adjusted R-squared:  0.8507 
F-statistic: 379.1 on 3 and 196 DF,  p-value: < 2.2e-16
```

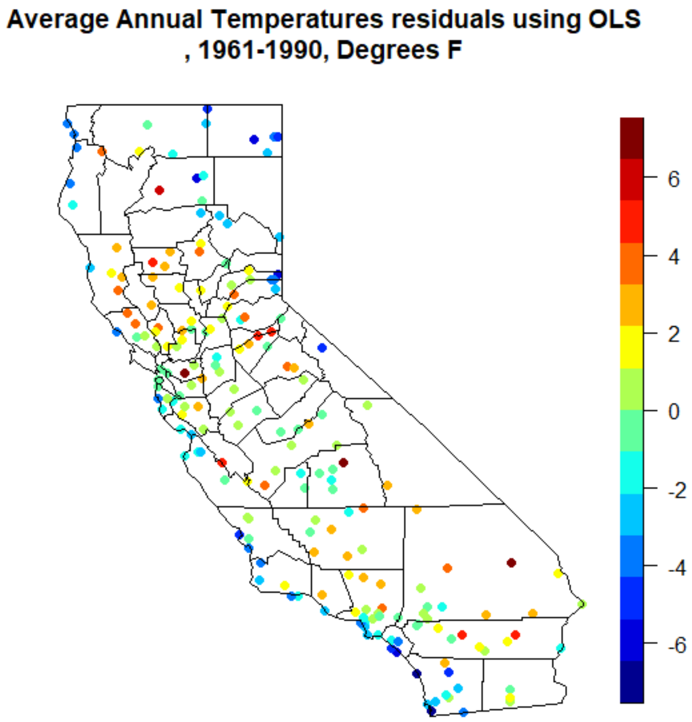


Figure 4 : OLS residuals plot

2-(b)

In order to see if data have spatial correlations, nonparamatric Variogram must be checked first. I take option of 'width = 10' which determines how many distances are grouped together to calculate variogram. In Figure 5 It is shown that semi-variogram become bigger as distance become farther which means correlation becomes smaller as distance become farther which means CAtemp data have spatial correlation.

After make nonparamatric variograms, paramateric variograms can be estimated. In this paper, we assume that data have exponential covariance function.

$$C(s_i, s_j) = \sigma^2 exp\{- \parallel s_i - s_j \parallel \rho\}$$

Fitting results are as folllows : $\hat{\sigma}^2 = 4.85,\ \hat{\rho} = 85.75,\ \hat{\tau}^2 = 1.90$

```
> # fit exponential variogram using weighted least square
> fitvg <- fit.variogram(vg, vgm(1, "Exp", 500, 0.05))
> print(fitvg)
  model    psill    range
1   Nug 1.895913  0.00000
2   Exp 4.845453 85.74578
```

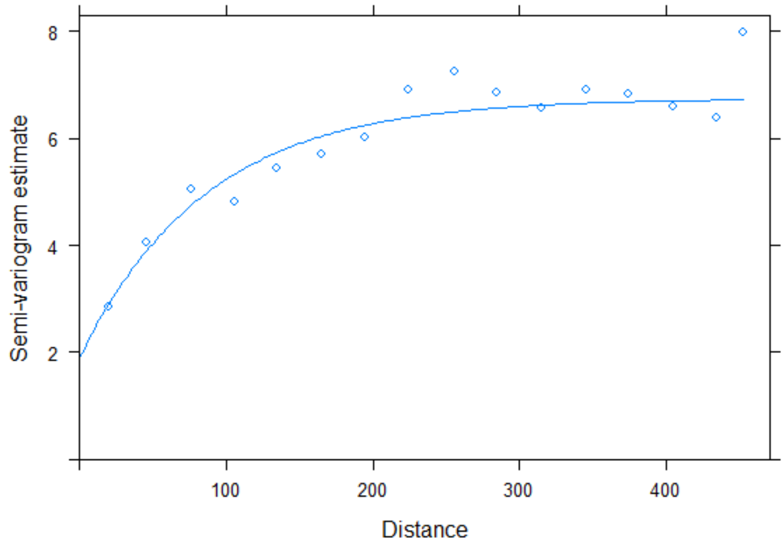


Figure 5 : Variogramgraph

2-(c)

In part 2-(b), We can estimate covariance function. By using this,we can get covariance matrix. At first, calculate distances d between data locations. In R codes, I use rdist.earth function. Second, create $\hat{\Sigma}$ using d. I save this matrix as name of 'cov'. And to calculate faster, I save inverse matrix of cov as name of 'cov.inv'. Third, create design matrix X by combining constant vector and three covariates of CAtemp. Finally, calculate $\hat{\beta}_{gls} = (X'\hat{\Sigma}^{-1}X)^{-1}X'\hat{\Sigma}Y$. Results are given below.

```
> Y = CAtemp.sub$avgtemp
> d <- rdist.earth(coordinates(CAtemp))
> cov <- s2.hat * Matern(d, range = rho.hat,
+                        nu = 0.5) + tau2.hat *diag(dim(d)[1]) # cov matrix
> cov.inv <- solve(cov) # save inverse of covariance matrix
> X <- cbind(rep(1, dim(d)[1]),
+            CAtemp$lon, CAtemp$lat, CAtemp$elevation) # build X using cbind
> beta.hat.gls <-
+   solve(t(X) %*% cov.inv %*% X) %*% t(X) %*% cov.inv %*% Y  # calculate beta hat
> beta.hat.gls
```

$$\hat{\beta}_{gls} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = \begin{bmatrix} 354.72 \\ 2.62 \\ 0.57 \\ -0.01 \end{bmatrix}$$

2-(d)

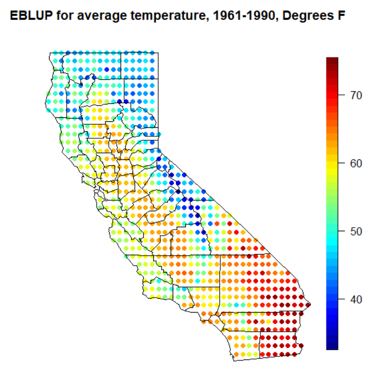
Prediction by using $\hat{\beta}_{gls}$ is Empirical Best Linear Unbiased Predictor. In the codes, dcross means the distance between CAtemp(trained data) and CAgrid(new data). Sigmacross is the covariance between them which also follows Exponential function.It is represented as γ in formula. Because prediction without measurement error is needed, τ^2 is not contained. Xpred is design matrix of CAgrid, which is made as same precedure as X. Ypred is calculated by formula : $\hat{Y(s_0)} = X(s_0)\hat{\beta}_{gls} + \gamma'\Sigma^{-1}(Y - X\hat{\beta}_{gls})$. Its standard error is calculated by

$$sd(Z) = \sqrt{\sigma^2 - \gamma'\Sigma^{-1}\gamma + b'(X'\Sigma{-}1X)^{-1}b}$$

$$where \;\; b = X(s_0)' - X'\Sigma^{-1}\gamma$$

In figure 6, it is shown that locations which share same regional characteristics have similar temperature. For example, West regions which abut onto the Pacific ocean, have temperature around 58, even if data doesn't contain that information. By this fact, we can infer that random effect of Spatio model can reflect the effects of variables not included in the model. In figure 7, it can be shown that locations near the CAtemp data has smaller Standard error

```
> dcross <-  rdist.earth(coordinates(CAtemp), coordinates(CAgrid))
> Sigmacross <- s2.hat * Matern(dcross, range = rho.hat, nu = 0.5)
> Xpred <- cbind(rep(1,dim(CAgrid)[1]),CAgrid$lon, CAgrid$lat, CAgrid$elevation)
> Ypred <- Xpred %*% beta.hat.gls +
+   t(Sigmacross) %*% cov.inv %*%(Y - X %*% beta.hat.gls)
```



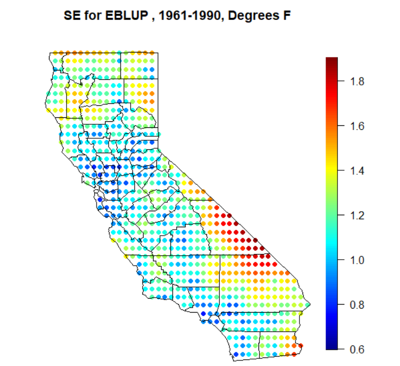


Figure 6 : EBLUP for average temperture
Figure 7 : Standar Error for EBLUP

R Code

```
library(classInt)
library(fields)
library(maps)
library(sp)
library(gstat)
library(geoR)
library(mvtnorm)
library(MCMCpack)
library(coda)

###### 1 ######
```

```
# (b)
library(mvtnorm)
library(fields)

x <- seq(0, 1, length = 500) # fine grid
d <- as.matrix(dist(x)) # create distance matrix
sigma1 <-  Matern(d, range = 1, nu = 0.5)
# make covariance matrix
mu1 <-  rep(0,500) # make mu vector

y_samp_gen <- function(mu, sigma,seed){
  L <-  t(chol(sigma))
  set.seed(seed)
  Z <-  rnorm(dim(sigma)[1], mean = 0, sd = 1)
  y <- mu + L %*% Z
  return(y)
}

Y <-  y_samp_gen(mu1, sigma1, 2021)
dim(Y)
head(Y)


# (c)
# change range
sigma1 <- Matern(d, range = 1, nu = 0.5) # same as exp
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

sigma2 <- Matern(d, range = 0.5, nu = 0.5) # change range
y2 <- y_samp_gen(mu = mu1, sigma = sigma2, seed = 2021)

ylim <- range(c(y1, y2))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(rho*"=1"))
matplot(x, y2, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(rho*"=0.5"))
# scale becomes bigger


# change nu
sigma1 <- Matern(d, range = 1, nu = 0.5) # same as exp
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

sigma3 <- Matern(d, range = 1, nu = 2) # change range
y3 <- y_samp_gen(mu = mu1, sigma = sigma3, seed = 2021)

ylim <- range(c(y1, y3))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(nu*"=0.5"))
matplot(x, y3, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(nu*"=2"))
```

```
# change mu
mu1 <-  rep(0,500) # make mu vector
y1 <- y_samp_gen(mu = mu1, sigma = sigma1, seed = 2021)

mu2 <-  rep(0.5,500)
y4 <- y_samp_gen(mu = mu2, sigma = sigma1, seed = 2021)

ylim <- range(c(y1, y4))
par(mfrow = c(1, 2), mar = c(3,3,3,3))
matplot(x, y1, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(mu*"=0"))
matplot(x, y4, type = "l", ylab = "Y(x)", ylim = ylim, main = expression(mu*"=0.5"))


###### 2 #####
# (a)
library(sp)
library(gstat)
library(fields)
library(classInt)
library(maps)

load("CAtemps.RData")
CAtemp
linmod <- lm(avgtemp ~ lon + lat + elevation, data = CAtemp)
summary(linmod)
linmod$coefficients
fitted <- predict(linmod, newdata = CAtemp, na.action = na.pass)
ehat <- CAtemp$avgtemp - fitted

# plotting
ploteqc <- function(spobj, z, breaks, ...){
  pal <- tim.colors(length(breaks)-1)
  fb <- classIntervals(z, n = length(pal),
                       style = "fixed", fixedBreaks = breaks)
  col <- findColours(fb, pal)
  plot(spobj, col = col, ...)
  image.plot(legend.only = TRUE, zlim = range(breaks), col = pal)
}



range(ehat)
breaks <- -7:7
x11()
ploteqc(CAtemp, ehat, breaks, pch = 19)
map("county", region = "california", add = TRUE)
title(main = "Average Annual Temperatures residuals using OLS\n,
      1961-1990, Degrees F")
```

```
# (b)
CAtemp$ehat <- ehat
CAtemp.sub <- CAtemp[!is.na(ehat),] # Remove lines with missing data
head(CAtemp.sub)

# range(CAtemp.sub$ehat)
vg <- variogram(ehat ~ 1, data = CAtemp.sub, width=30) #  width : set bins
plot(vg, xlab = "Distance", ylab = "Semi-variogram estimate", width=15)


# fit exponential variogram using weighted least square
fitvg <- fit.variogram(vg, vgm(1, "Exp", 500, 0.05))
print(fitvg)
# store estimates
s2.hat <- fitvg$psill[2]
rho.hat <- fitvg$range[2]
tau2.hat <- fitvg$psill[1]

# plotting
plot(vg, fitvg, xlab = "Distance", ylab = "Semi-variogram estimate")


#(c)
Y = CAtemp.sub$avgtemp
d <- rdist.earth(coordinates(CAtemp))
cov <- s2.hat * Matern(d, range = rho.hat,
                       nu = 0.5) + tau2.hat *diag(dim(d)[1]) # cov matrix
cov.inv <- solve(cov) # save inverse of covariance matrix
X <- cbind(rep(1, dim(d)[1]),
           CAtemp$lon, CAtemp$lat, CAtemp$elevation) # build X using cbind
beta.hat.gls <-
  solve(t(X) %*% cov.inv %*% X) %*% t(X) %*% cov.inv %*% Y
# calculate beta hat
beta.hat.gls

#(d)
dcross <-  rdist.earth(coordinates(CAtemp), coordinates(CAgrid))
Sigmacross <- s2.hat * Matern(dcross, range = rho.hat, nu = 0.5)
Xpred <- cbind(rep(1,dim(CAgrid)[1]),CAgrid$lon, CAgrid$lat, CAgrid$elevation)
Ypred <- Xpred %*% beta.hat.gls +
  t(Sigmacross) %*% cov.inv %*%(Y - X %*% beta.hat.gls)

# plotting Ypred
range(Ypred)
breaks <- 33:75
x11()
ploteqc(CAgrid, Ypred, breaks, pch = 19)
map("county", region = "california", add = TRUE)
title(main = "EBLUP for average temperature, 1961-1990, Degrees F")
```

```
# MSE
b <-  t(Xpred) - t(X)%*%cov.inv%*%Sigmacross
vpred <- s2.hat - diag(t(Sigmacross) %*% solve(cov, Sigmacross) +
                         t(b) %*% solve(t(X) %*% solve(cov, X), b))
vpred[vpred<0] <- 0
sepred <- sqrt(vpred)
sepred

# plotting MSE
range(sepred)
breaks <- seq(0.6,1.9,by = 0.01)
x11()
ploteqc(CAgrid, sepred, breaks, pch = 19)
map("county", region = "california", add = TRUE)
title(main = "SE for EBLUP , 1961-1990, Degrees F")
```