ST540 Final Project

Team Members: Xintong Jiang, Morgan Kuchenbrod, Leah Rohde, Yihang Xu

Question 1

The data should be modeled as

$$Y_i = g(s_{i1}, s_{i2}, t_i) + \varepsilon_i$$

where g is the mean MAT for a given spatiotemporal coordinate and ε_i are independent errors. There are obviously other models for g, but for this exam we will use splines. Let $B_{11}(s_1), \dots, B_{1J}(s_1)$ be a spline basis expansion of longitude, $B_{21}(s_2), \dots, B_{2K}(s_2)$ be a spline basis expansion of latitude and $B_{t1}(t), \dots, B_{tL}(t)$ be a spline basis expansion of time. The mean MAT is modeled using all three sets of the basis functions and their interactions in a multiple linear regression:

$$g(s_{i1}, s_{i2}, t_i) = \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} X_{ijkl}\,\beta_{jkl} \text{ where } \quad X_{ijkl} = B_{1j}(s_{i1})B_{2k}(s_{i2})B_{tl}(t_i).$$

Next we aim to specify the priors of these parameters and errors. The independent errors $\varepsilon_i, i = 1, \dots, n$ are assumed to be distributed as $\varepsilon \sim N\big(\mathbf{0}, \mathrm{Diag}(\sigma_\varepsilon^2)\big)$ and σ_ε^2 is assigned a scaled inverse- χ^2 prior with degree of freedom and scale parameter $\sigma_\varepsilon^2 \sim \chi^{-2}(\sigma_\varepsilon^2 \mid df_\varepsilon, S_\varepsilon)$. Since $X_{ijkl} = B_{1j}(s_{i1})B_{2k}(s_{i2})B_{tl}(t_i)$, the dimension of X can be rewritten as $X \in \mathbb{R}^{n\times(JKL)}$. The dimension of feature is JKL.More precise, we assume that The prior is $\beta_j \sim DE(\tau)$ which has PDF

$$f(\beta) \propto \exp\left(-\frac{|\beta|}{\tau}\right).$$

This is also known as the Bayesian LASSO prior. The remaining default parameters are fixed and same as package.

Question 2

Define the the deviance as twice the negative log likelihood

$$D(\mathbf{Y} \mid \boldsymbol{\theta}) = -2\log[f(\mathbf{Y} \mid \boldsymbol{\theta})].$$

Let $\bar{D} = \mathrm{E}[D(Y \mid \theta) \mid \mathbf{Y}]$ be the posterior mean of the deviance. Denote $\widehat{\boldsymbol{\theta}}$ as the posterior mean of $\boldsymbol{\theta}$. The effective number of parameters is

$$p_D = \bar{D} - D\big(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}\big).$$

DIC can be written

$$DIC = \bar{D} + p_D = D(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}) + 2p_D.$$

We propose to use the Deviance information criteria (DIC) to select the number of basic functions. We will select the model with smallest DICs.

Question 3

For simplicity, we always assume that $J = K = L$. We investigate 5 models as $J = K = L = 6,7,8,9,10,11,12,15,20$. We use package to fit the model. The DICs of the model are

(44323.09,42903.85,42213.94,41408.50,41047.25,40292.67,40029.23,38886.24,37847.71).

Hence, we choose the $J = K = L = 20$ model. In such a case, the average MSE is

$$\sum_{i=1}^{n} 1/n(\hat{y}_i - y_i)^2 = 9.67.$$

To further verify that the model fits well, we first plot the true MAT and the estimated MAT as follows. From Figure 1 below, we can find that BLR predicts the MAT well. The black points represent the true MAT and the red points represent the estimated MAT.

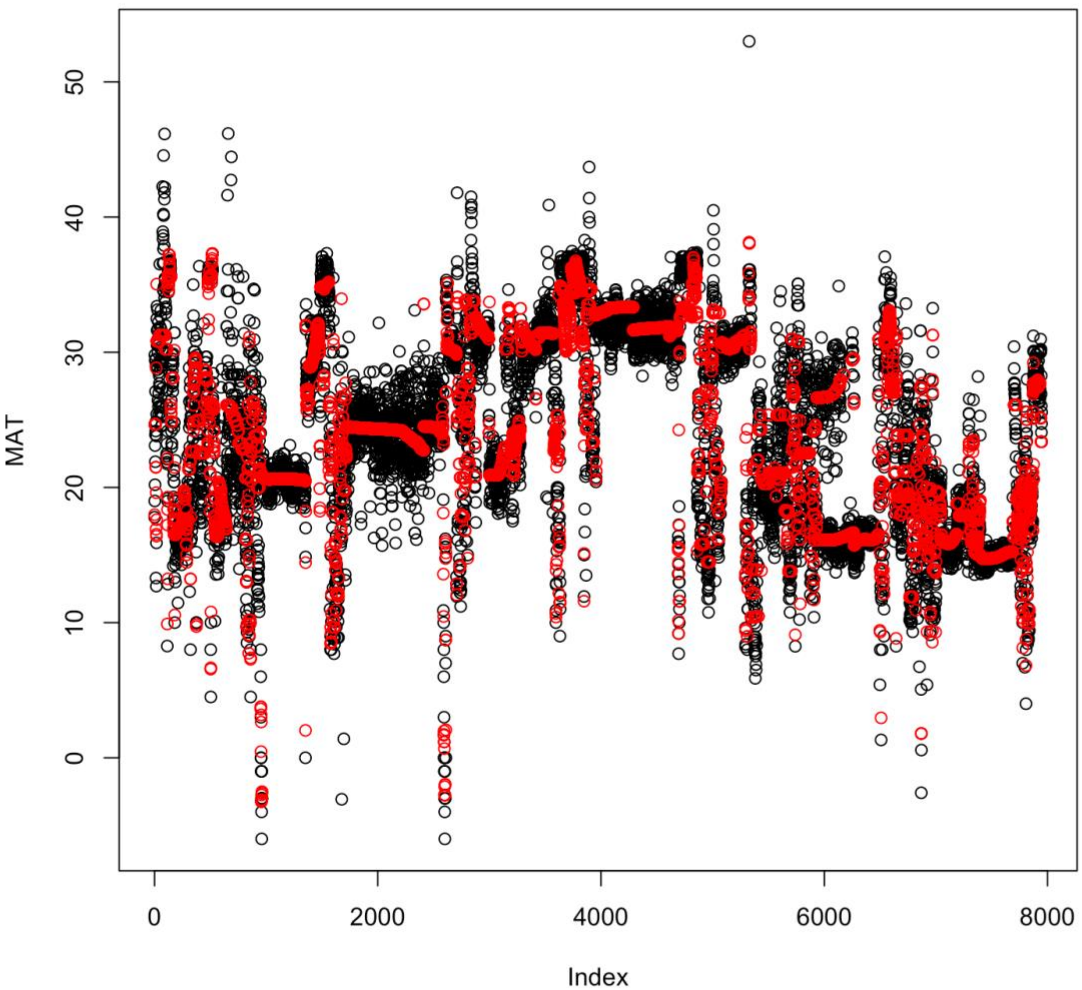

Figure 1

Question 4

We use the posterior mean $\hat{\mu}$ and the posterior standard variation $\hat{\sigma}$ to predict the value of MAT. We construct a $20 \times 20 \times 20$ 3D grid on the domain. The upper bound of the estimated MAT is set as $\hat{\mu} + 3\hat{\sigma}$ and the lower bound of the estimated MAT is set as $\hat{\mu} - 3\hat{\sigma}$. The following Figure 2 shows the estimated MATs of observations. The red points represent the estimated MAT; The blue points represent the upper bound; The yellow points represent the lower bound. The remaining estimated MATs (3D grid) can be seen in

Question 5.

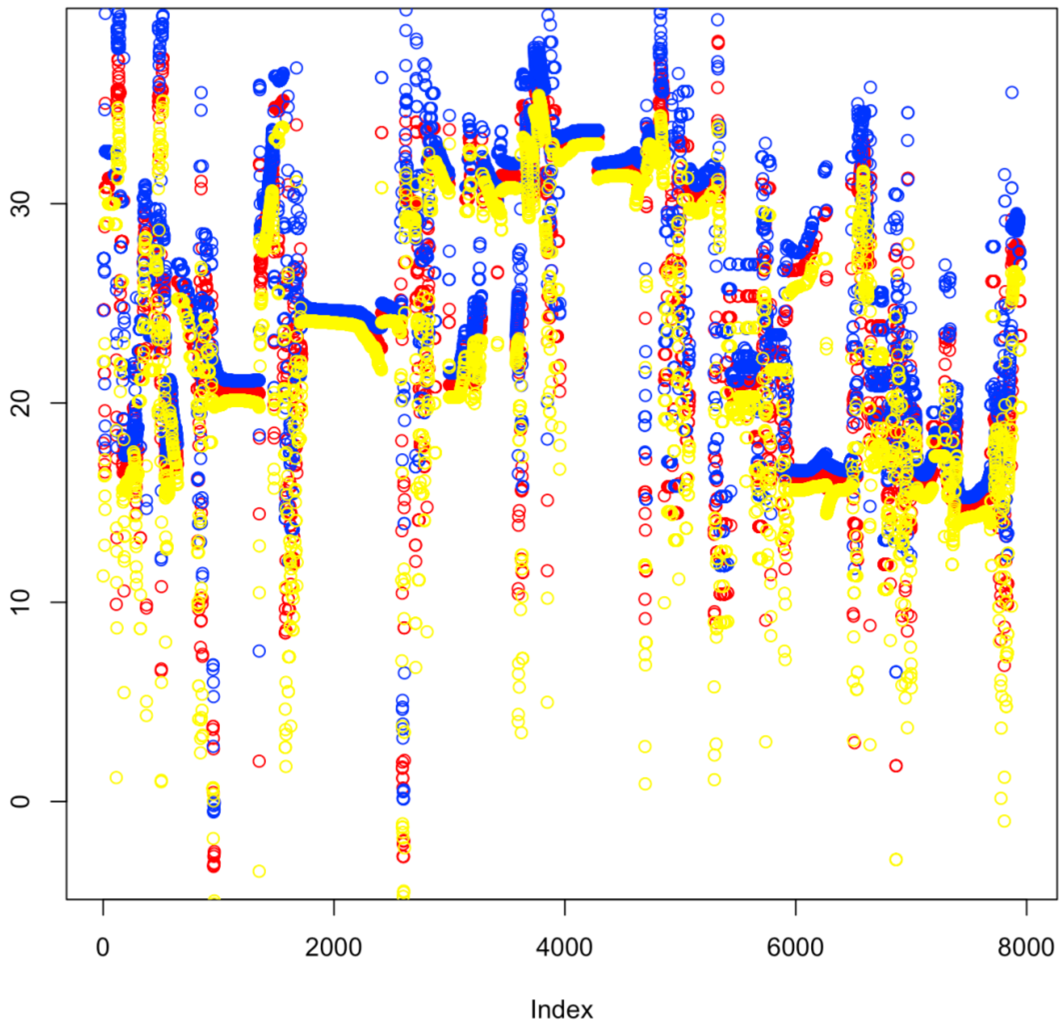

Figure 2

Question 5

We successfully create 2 visualization tools that a user can input a time and see a map of estimated MAT or input a location and see a time series of estimated MAT.

Figure 3 showed an example of a 3D map when the time is age=102.63.

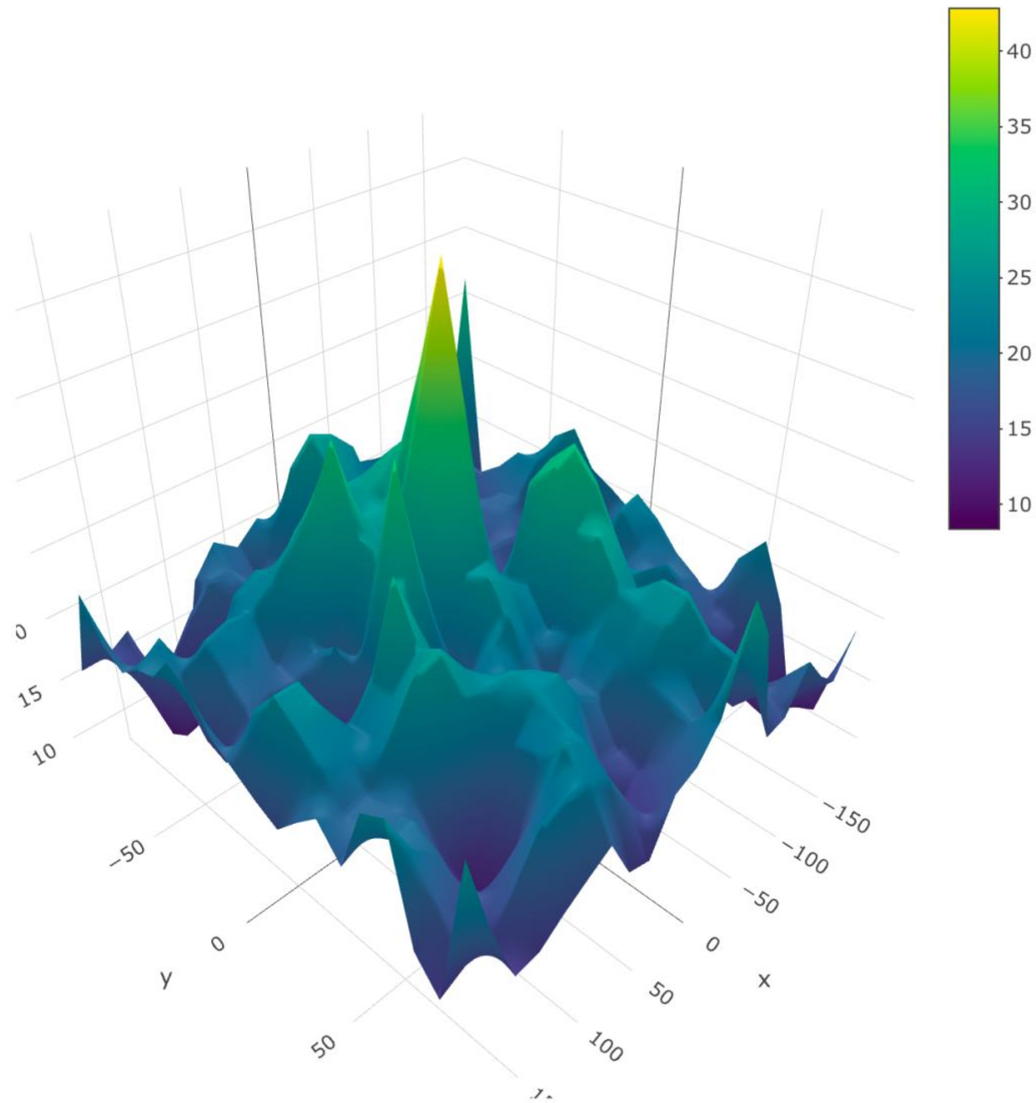

Figure 3. MATs (3D grid)

Figure 4 showed an example of plot for MAT corresponding to time series when the location is (lon=-9.473684, lat=-4.736842).

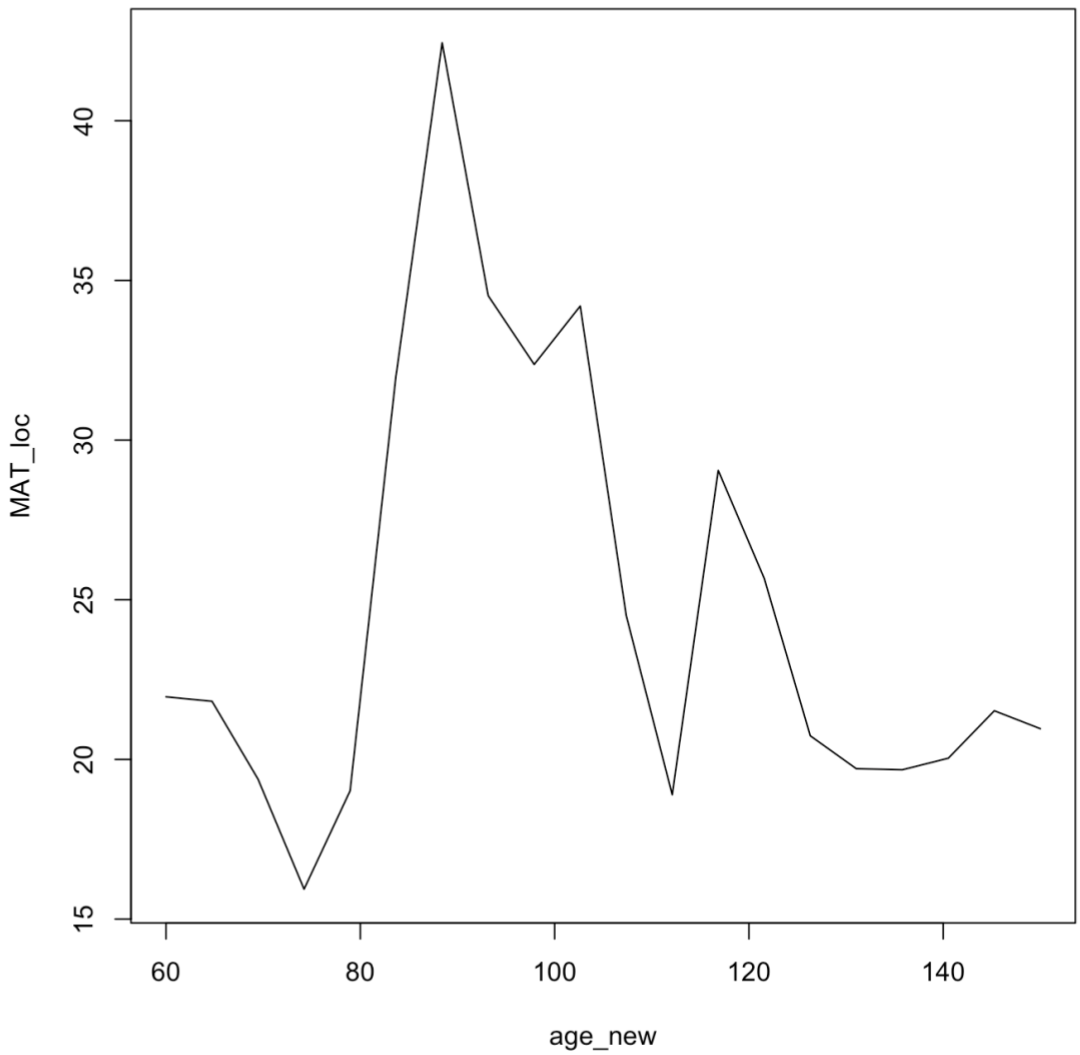

Figure 4

```
###Code Section
library(BGLR)
### Problem 1
 ### loading data data <- read.csv("C:/Users/yxu/Desktop/paleo_dat.csv")
 head(data)
data_BLR = data[!is.na(data$Temperature.C),c(1,3,4,7)]
data_BLR = data_BLR[data_BLR$Sample.Age>=60,]
head(data_BLR)
### Fixing Model
bs_fixed_knots <- function(x, x_min, x_max, df) {
        library(splines)
        B = bs(x, df=df, Boundary.knots = c(x_min,x_max),
                knots = seq(x_min,x_max,length = df -3)
                )
        return(B)
}
 ### Problem 2
### Use DIC as the strategy for selecting the number of basis functions
cal_DIC <- function(knots) {
         range1 = c(-180,180)
        J = knots
        lon = data_BLR$Paleo.Lon
        B1 = bs_fixed_knots(lon, range1[1], range1[2], J)
        range2 = c(-90,90) K = knots lat = data_BLR$Paleo.Lat
        B2 = bs_fixed_knots(lat, range2[1], range2[2], K)
        range3 = c(60,150)
        L = knots age = data_BLR$Sample.Age
        B3 = bs_fixed_knots(age, range3[1], range3[2], L)
```

```
    X = NULL
    for (j in 1:J) {
        for (k in 1:K) {
            for (l in 1:L) {
                X = cbind(X, B1[,j]*B2[,k]*B3[,l])
            }
        }
    }
    MAT = data_BLR$Temperature.C
    out=BLR(y=MAT,XL=X)
    return(out$fit$DIC)
}
###### Problem 3 & 4
# Calculating DIC for several models
DIC_vec = c(cal_DIC(6), cal_DIC(7), cal_DIC(8), cal_DIC(9), cal_DIC(10))
DIC_vec2 = c(cal_DIC(11), cal_DIC(12))
DIC_vec3 = c(cal_DIC(15), cal_DIC(20))
# Further check whether the model with smallest DIC fit well by using plot.
knots = 20
range1 = c(-180,180)
J = knots
lon = data_BLR$Paleo.Lon
B1 = bs_fixed_knots(lon, range1[1], range1[2], J)
range2 = c(-90,90)
K = knots
lat = data_BLR$Paleo.Lat
B2 = bs_fixed_knots(lat, range2[1], range2[2], K)
range3 = c(60,150)
```

```
L = knots
age = data_BLR$Sample.Age
B3 = bs_fixed_knots(age, range3[1], range3[2], L)
X = NULL
for (j in 1:J) {
    for (k in 1:K) {
        for (l in 1:L) {
            X = cbind(X, B1[,j]*B2[,k]*B3[,l])
        }
    }
}
MAT = data_BLR$Temperature.C
out=BLR(y=MAT,XL=X)
(mean((out$y - out$yHat)^2))
lon_new = seq(-180,180,length = 20)
lat_new = seq(-90,90,length = 20)
age_new = seq(60,150,length = 20)
lon_pre = NULL
lat_pre = NULL
age_pre = NULL
for (j in 1:20) {
    for (k in 1:20) {
        for (l in 1:20) {
            lon_pre = c(lon_pre, lon_new[j])
            lat_pre = c(lat_pre,lat_new[k])
            age_pre = c(age_pre,age_new[l])
        }
    }
```

```
}
MAT_pre = rep(NA, 8000)
lon_all = c(lon, lon_pre)
lat_all = c(lat, lat_pre)
age_all = c(age, age_pre)
MAT_all = c(MAT, MAT_pre)

lengknots = 20
range1 = c(-180,180)
J = knots lon = data_BLR$Paleo.Lon
B1 = bs_fixed_knots(lon_all, range1[1], range1[2], J)

range2 = c(-90,90)
K = knots
lat = data_BLR$Paleo.Lat
B2 = bs_fixed_knots(lat_all, range2[1], range2[2], K)
range3 = c(60,150)
L = knots
age = data_BLR$Sample.Age
B3 = bs_fixed_knots(age_all, range3[1], range3[2], L)
X = NULL
for (j in 1:J) {
    for (k in 1:K) {
        for (l in 1:L) {
            X = cbind(X, B1[,j]*B2[,k]*B3[,l])
        }
    }
}
```

```
out=BLR(y=MAT_all,XL=X)
yhat = out$yHat
sdhat = out$SD.yHat
setwd("/Users/apple/Desktop/huge/TA/Baysian")
save(yhat, sdhat, file = "BLR.Rdata")

# Plot for problem 3
plot(MAT)
points(yhat[1:7947], col = 'red')

# Plot for problem 4
plot(yhat[1:7947], col="red")
points(yhat[1:7947]+3*sdhat[1:7947], col = "blue")
points(yhat[1:7947]-3*sdhat[1:7947], col = "yellow")

######## Problem 5
### plot with fixed time ###
time = age_pre[10]
ind_time = which(age_pre == time)
lon_time = lon_pre[ind_time]
lat_time = lat_pre[ind_time]
MAT_pre = yhat[7948:15947]
MAT_time = matrix(MAT_pre[ind_time], ncol = 20,nrow = 20, byrow = T)
library(plot3D)
library(tidyverse)
library(plotly)
persp(lon_new,lon_new, MAT_time)
plot_ly(x = lon_new, y = lat_new, z = MAT_time) %>% add_surface()
```

```
### plot with fixed location (longitude and latitude) ###

location = c(lon_new[10], lat_new[10])

ind_loc = which((lon_pre == location[1]) & (lat_pre == location[2]))

MAT_pre = yhat[7948:15947]

MAT_loc = MAT_pre[ind_loc]

plot(age_new, MAT_loc, type = "l")
```