ST440/540 – Exam 2 – Solution

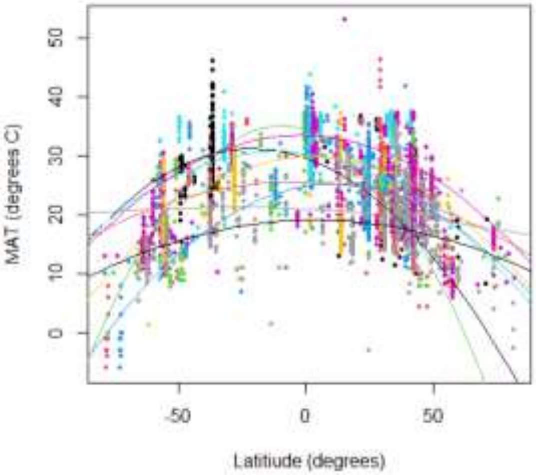
(1a) The model is $Y_i = g_{fi}(X_i) + e_i$ where the errors are $e_i \sim N(0,\sigma^2)$ independent over i and the mean curve for time slice k is $g_k(X) = \beta_{k1} + \beta_{k2}X + \beta_{k3}X^2$. In this expression, X=latitude/90 is standardized. The parameters for each time slice are treated as random effects with distribution $\beta_{kj} \sim \text{Normal}(\mu_j,\tau_1^2)$. The hyperpriors are $\mu_j \sim \text{Normal}(\mu_{0,} \tau_2^2)$, $\mu_0 \sim \text{Normal}(0,100)$ and σ^2, τ_1^2, $\tau_2^2 \sim \text{InvGamma}(0.1,0.1)$.

(1b) I compared three models using DIC: the **random effects** model described in (1a), the **fixed effects** model that sets prior $\beta_{kj} \sim \text{Normal}(0,100)$ and the **constant** model with the same mean for all eras and $\beta_{kj} = b_j$ for all k with priors $b_j \sim \text{Normal}(0,100)$. The random effects model has the smallest DIC and is thus used for the remainder of the analysis.

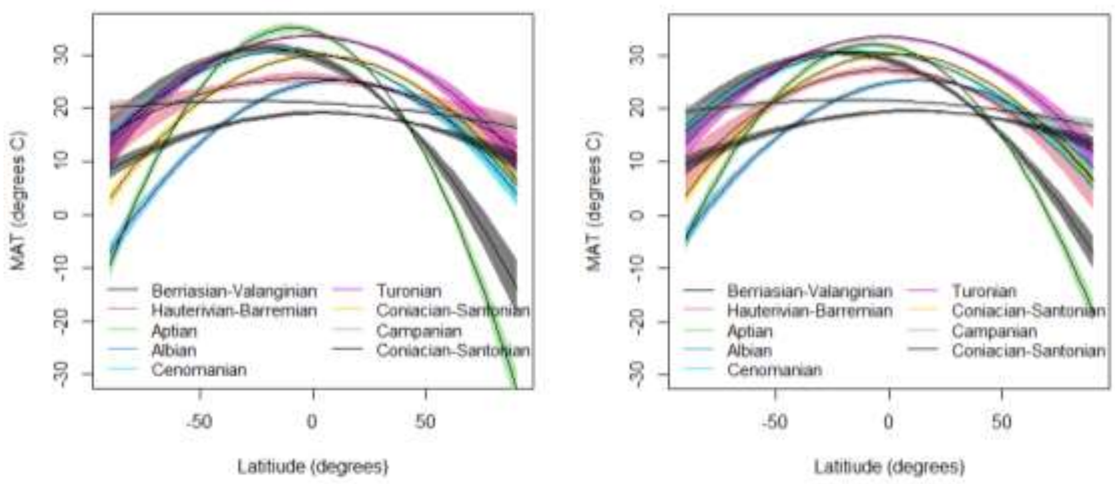
	D-bar	PD	DIC
Random effects	**47565**	27.6	**47593**
Fixed effects	47568	27.6	47596
Constant	51986	**4.0**	51990

(1c) To monitor convergence I computed the effective sample size and Gelman's R statistic for each β_{kj}. For the two chains of length 10000 (after 5000 burn in) iterations, for all β_{kj} the effective sample size was larger than 1200 and the R statistic was smaller than 1.02. Therefore, the chains appear to have converged and mixed sufficiently.

(1d) Below is a plot of the data (dots colored by time slice) versus posterior mean of g_k for each time slice. The lines generally fit the patterns of the data. I also computed posterior predictive checks using the minimum, maximum, range and standard deviation (across all observations and eras). The Bayesian p-value for the maximum was 0.02 and the rest were between 0.1 and 0.9. Therefore, there is some evidence of a lack of fit in the upper tail. To address this, I might fit a higher-order polynomial and check the validity of the largest observation with MAT ≈ 50.



(1e) Below are the posterior mean (lines) and 90% intervals (shading) for the g_k. The plot on the left uses only quantitative data and the plot of the right uses all data.



(1f) DIC strongly favors the models with different parameters per time slice and the plots in (1e) show non-overlapping credible intervals, therefore there is evidence of a changing climate.

(2a) The model for the quantitative data and priors are the same as in (1a). The likelihood for interval observation i is $F(u_i,g_{fi}(X_i),\sigma^2) - F(l_i,g_{fi}(X_i),\sigma^2)$ where F is the normal CDF.

(2b) See the figure in (1e).

(2c) The estimated g_k curves in (1e) are fairly similar. However, the average (over time slice and latitude) width of 95% intervals is 2.26 without interval data and 1.90 with interval data, so the interval data reduces uncertainty and therefore provide some value.

```
# Code for question 1
rm(list=ls())
dat <- read.csv("paleo_dat.csv",header=TRUE)

###############################################
# LOAD THE DATA
###############################################

# Put the eras in chronological order
labs <- c("Berriasian-Valanginian","Hauterivian-Barremian",
          "Aptian","Albian","Cenomanian","Turonian",
          "Coniacian-Santonian","Campanian","Coniacian-Santonian")

e <- dat[,2]
e <- ifelse(e == "Albian","4_Albian",e)
e <- ifelse(e == "Aptian","3_Aptian",e)
e <- ifelse(e == "Berriasian-Valanginian","1_Berriasian-Valanginian",e)
e <- ifelse(e == "Campanian","8_Campanian",e)
e <- ifelse(e == "Cenomanian","5_Cenomanian",e)
e <- ifelse(e == "Coniacian-Santonian","7_Coniacian-Santonian",e)
e <- ifelse(e == "Hauterivian-Barremian","2_Hauterivian-Barremian",e)
e <- ifelse(e == "Maastrichtian","9_Maastrichtian",e)
e <- ifelse(e == "Turonian","6_Turonian",e)

dat[,2] <- e
eras    <- as.factor(dat[,2])
era     <- as.numeric(eras)
n_eras  <- max(era)

lat <- dat[,3]/90 # Standardize paleo latitude

# Separate observations into quantitive and interval
 temp <- dat[,7]
 low  <- dat[,8]
 high <- dat[,9]
 type <- ifelse(!is.na(temp),"exact","interval")

 Y_ex    <- temp[type=="exact"]
 era_ex  <- era[type=="exact"]
 lat_ex  <- lat[type=="exact"]
 n_ex    <- length(Y_ex)

# (1b) Model comparisons with DIC

 library(rjags)
 burn   <- 5000
 iters  <- 10000
 data   <- list(Y_ex=Y_ex,era_ex=era_ex,lat_ex=lat_ex,
                n_eras=n_eras,n_ex=n_ex)
 inits  <- list(taue=1)
 params <- c("beta")

# Random effects model

 model_string <- textConnection("model{

   for(i in 1:n_ex){
     Y_ex[i]     ~ dnorm(mu_ex[i],taue)
     mu_ex[i]   <- beta[era_ex[i],1] +
                   beta[era_ex[i],2]*lat_ex[i] +
                   beta[era_ex[i],3]*lat_ex[i]*lat_ex[i]
   }
   for(j in 1:3){
     for(e in 1:n_eras){
      beta[e,j] ~ dnorm(mu[j],taub)
     }
     mu[j]     ~ dnorm(mu0,taum)
   }
   mu0     ~ dnorm(0,0.01)
   taum    ~ dgamma(.1,.1)
   taue    ~ dgamma(.1,.1)
   taub    ~ dgamma(.1,.1)
 }")

 model1   <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)
 update(model1, burn, progress.bar="none")
 samples1 <- coda.samples(model1, variable.names=params,
                          n.iter=iters, progress.bar="none")
 DIC1    <- dic.samples(model1,n.iter=iters, progress.bar="none")
 print(DIC1)

# Fixed effects model

 model_string <- textConnection("model{

   for(i in 1:n_ex){
     Y_ex[i]     ~ dnorm(mu_ex[i],taue)
     mu_ex[i]   <- beta[era_ex[i],1] +
                   beta[era_ex[i],2]*lat_ex[i] +
                   beta[era_ex[i],3]*lat_ex[i]*lat_ex[i]
   }
   for(j in 1:3){
     for(e in 1:n_eras){
      beta[e,j] ~ dnorm(0,0.01)
     }
   }
   taue    ~ dgamma(.1,.1)
 }")

 model2   <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)
 update(model2, burn, progress.bar="none")
 samples2 <- coda.samples(model2, variable.names=params,
                          n.iter=iters, progress.bar="none")
 DIC2    <- dic.samples(model2,n.iter=iters, progress.bar="none")
 print(DIC2)

# Constant model

 model_string <- textConnection("model{

   for(i in 1:n_ex){
     Y_ex[i]     ~ dnorm(mu_ex[i],taue)
     mu_ex[i]   <- beta[1] +
                   beta[2]*lat_ex[i] +
                   beta[3]*lat_ex[i]*lat_ex[i]
   }
   for(j in 1:3){
      beta[j] ~ dnorm(0,0.01)
   }
   taue    ~ dgamma(.1,.1)
 }")

 model3   <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)
 update(model3, burn, progress.bar="none")
 samples3 <- coda.samples(model3, variable.names=params,
                          n.iter=iters, progress.bar="none")
 DIC3    <- dic.samples(model3,n.iter=iters, progress.bar="none")
 print(DIC3)
```

```
# (1c) Check convergence for Model 1

 print(range(effectiveSize(samples1)))
 print(range(gelman.diag(samples1)))

# (1e) Plot the estimated curves

nl    <- 50 # Grid of latitudes
LAT   <- seq(-1,1,length=nl)
beta  <- rbind(samples1[[1]],samples1[[2]])
g     <- array(NA,c(nl,9,3))

for(j in 1:9){
  g_samps <- NULL
  for(i in 1:nrow(beta)){
    b       <- matrix(beta[i,],9,3)
    gi      <- b[j,1] +  b[j,2]*LAT + b[j,3]*(LAT^2)
    g_samps <- rbind(g_samps,gi)
  }
  g[,j,1] <- apply(g_samps,2,quantile,0.05)
  g[,j,2] <- apply(g_samps,2,quantile,0.50)
  g[,j,3] <- apply(g_samps,2,quantile,0.95)
}
plot(NA,xlim=range(90*LAT),ylim=c(-30,35),xlab="Latitiude (degrees)",ylab="MAT (degrees C)")
for(j in 1:9){
  polygon(90*c(LAT, rev(LAT)), c(g[,j,1], rev(g[,j,3])),
            col = adjustcolor(j,alpha.f=0.5),border=F)
}
matplot(90*LAT,g[,,2],type="l",col=1,lty=1,add=TRUE)
legend("bottomleft",labs,col=1:9,lty=1,ncol=2,cex=.9,bty="n")

# (1d) Data versus fitted values
plot(90*lat_ex,Y_ex,col=era,cex=.5,pch=19,
     xlab="Latitiude (degrees)",ylab="MAT (degrees C)")
matplot(90*LAT,g[,,2],type="l",col=1:9,lty=1,add=TRUE)

# Posterior predictive checks

 model_string <- textConnection("model{

   for(i in 1:n_ex){
     Y_ex[i]     ~ dnorm(mu_ex[i],taue)
     mu_ex[i]   <- beta[era_ex[i],1] +
                   beta[era_ex[i],2]*lat_ex[i] +
                   beta[era_ex[i],3]*lat_ex[i]*lat_ex[i]
   }
   for(j in 1:3){
     for(e in 1:n_eras){
      beta[e,j] ~ dnorm(mu[j],taub)
     }
     mu[j]     ~ dnorm(mu0,taum)
   }
   mu0     ~ dnorm(0,0.01)
   taum    ~ dgamma(.1,.1)
   taue    ~ dgamma(.1,.1)
   taub    ~ dgamma(.1,.1)

   for(i in 1:n_ex){
     Yp[i]     ~ dnorm(mu_ex[i],taue)
   }
   D[1] <- max(Yp[])
   D[2] <- min(Yp[])
   D[3] <- D[1]-D[2]
   D[4] <- sd(Yp[])

 }")
 model   <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)
 update(model, burn, progress.bar="none")
 samples <- coda.samples(model, variable.names=c("D"),
                          n.iter=iters, progress.bar="none")
 D       <- rbind(samples[[1]],samples[[2]])
 D0 <-  c(max(Y_ex),min(Y_ex),max(Y_ex)-min(Y_ex),sd(Y_ex))
 mean(D[,1]>D0[1])
 mean(D[,2]>D0[2])
 mean(D[,3]>D0[3])
 mean(D[,4]>D0[4])

mean_width <- mean(g[,,3]-g[,,1])
print(mean_width)
```

```
# Code for question 2

dat <- read.csv("paleo_dat.csv",header=TRUE)

###############################################
# LOAD THE DATA
###############################################

# Put the eras in chronological order
labs <- c("Berriasian-Valanginian","Hauterivian-Barremian",
          "Aptian","Albian","Cenomanian","Turonian",
          "Coniacian-Santonian","Campanian","Coniacian-Santonian")

e <- dat[,2]
e <- ifelse(e == "Albian","4_Albian",e)
e <- ifelse(e == "Aptian","3_Aptian",e)
e <- ifelse(e == "Berriasian-Valanginian","1_Berriasian-Valanginian",e)
e <- ifelse(e == "Campanian","8_Campanian",e)
e <- ifelse(e == "Cenomanian","5_Cenomanian",e)
e <- ifelse(e == "Coniacian-Santonian","7_Coniacian-Santonian",e)
e <- ifelse(e == "Hauterivian-Barremian","2_Hauterivian-Barremian",e)
e <- ifelse(e == "Maastrichtian","9_Maastrichtian",e)
e <- ifelse(e == "Turonian","6_Turonian",e)

dat[,2] <- e
eras    <- as.factor(dat[,2])
era     <- as.numeric(eras)
n_eras  <- max(era)


# Separate observations into quantitive and interval
 lat  <- dat[,3]/90 # Standardize paleo latitude
 temp <- dat[,7]
 low  <- dat[,8]
 high <- dat[,9]
 type <- ifelse(!is.na(temp),"exact","interval")

 Y_ex    <- temp[type=="exact"]
 era_ex  <- era[type=="exact"]
 lat_ex  <- lat[type=="exact"]
 n_ex    <- length(Y_ex)

 low_int  <- low[type=="interval"]
 high_int <- high[type=="interval"]
 era_int  <- era[type=="interval"]
 lat_int  <- lat[type=="interval"]
 n_int    <- length(low_int)


# Get ready for JAGS

 library(rjags)

 data <- list(Y_ex=Y_ex,era_ex=era_ex,lat_ex=lat_ex,
              low_int=low_int,high_int=high_int,era_int=era_int,
              ones=rep(1,n_int),lat_int=lat_int,
              n_eras=n_eras,n_ex=n_ex,n_int=n_int)

 model_string <- textConnection("model{

   for(i in 1:n_ex){
     Y_ex[i]     ~ dnorm(mu_ex[i],taue)
     mu_ex[i]   <- beta[era_ex[i],1] +
                   beta[era_ex[i],2]*lat_ex[i] +
                   beta[era_ex[i],3]*lat_ex[i]*lat_ex[i]
   }
   for(i in 1:n_int){
     ones[i]    ~ dbern(prob[i])
     prob[i]   <- pnorm(high_int[i],mu_int[i],taue) -
                  pnorm( low_int[i],mu_int[i],taue)
     mu_int[i] <- beta[era_int[i],1] +
                  beta[era_int[i],2]*lat_int[i] +
                  beta[era_int[i],3]*lat_int[i]*lat_int[i]
   }
   for(j in 1:3){
     for(e in 1:n_eras){
      beta[e,j] ~ dnorm(mu[j],taub)
     }
     mu[j]     ~ dnorm(mu0,taum)
   }
   mu0     ~ dnorm(0,0.01)
   taum    ~ dgamma(.1,.1)
   taue    ~ dgamma(.1,.1)
   taub    ~ dgamma(.1,.1)
 }")

 inits <- list(taue=.01)
 model <- jags.model(model_string,data = data, inits=inits, n.chains=2,quiet=TRUE)

# Run JAGS

 burn  <- 5000
 iters <- 10000

 update(model, burn, progress.bar="none")
 params  <- c("beta")
 samples <- coda.samples(model,
            variable.names=params,
            n.iter=iters, progress.bar="none")

# Plot the results

nl    <- 50 # Grid of latitudes
LAT   <- seq(-1,1,length=nl)
beta  <- rbind(samples[[1]],samples[[2]])
g     <- array(NA,c(nl,9,3))

for(j in 1:9){
  g_samps <- NULL
  for(i in 1:nrow(beta)){
    b       <- matrix(beta[i,],9,3)
    gi      <- b[j,1] +  b[j,2]*LAT + b[j,3]*(LAT^2)
    g_samps <- rbind(g_samps,gi)
  }
  g[,j,1] <- apply(g_samps,2,quantile,0.05)
  g[,j,2] <- apply(g_samps,2,quantile,0.50)
  g[,j,3] <- apply(g_samps,2,quantile,0.95)
}
plot(NA,xlim=range(90*LAT),ylim=c(-30,35),xlab="Latitiude (degrees)",ylab="MAT (degrees C)")
for(j in 1:9){
  polygon(90*c(LAT, rev(LAT)), c(g[,j,1], rev(g[,j,3])),
            col = adjustcolor(j,alpha.f=0.5),border=F)
}
matplot(90*LAT,g[,,2],type="l",col=1,lty=1,add=TRUE)
legend("bottomleft",labs,col=1:9,lty=1,ncol=2,cex=.9,bty="n")

mean_width <- mean(g[,,3]-g[,,1])
print(mean_width)
```