ST 540 – Applied Bayesian Analysis

Model Definition

For enhanced vegetation index (EVI), X was chosen to be representative time, which is analyzed at the day of the year at which the EVI measurement was taken. Y represents the EVI measurement and is assumed to be normally distributed about $u(X)$. Analyzing Day of Year (DOY) without the inclusion of year was considered reasonable, as mean EVI was not found to have a significant relationship with year. This allows for the full dataset to inform on the 365 day EVI curve. Therefore, a non-linear regression model was constructed as follows:

$$Y_i \sim N(u(X_i), \sigma^2)$$

Where the function u is splines of X

$$u(X) = \mu + \sum_{j=1}^{J} B_j(X)\beta_j$$

Because no additional information is available on EVI, uninformative priors are utilized:

$$\mu_1 \sim N(0{,}100)$$

$$B_j \sim N(0, \sigma^2\tau^2)$$

$$\sigma^2, \tau^2 \sim InvGamma(0.1{,}0.1)$$

The model chosen in the Model Comparisons section also allows log variance to change with X. This requires an embedded secondary spline model:

$$log(\sigma^2(X)) = \mu_2 + \sum_{j=1}^{J} B_j(X)\alpha_j$$

$$\mu_2 \sim N(0{,}100)$$

$$B_j \sim N(0, \sigma_b^2)$$

$$\alpha_j \sim N(0, \sigma_a^2)$$

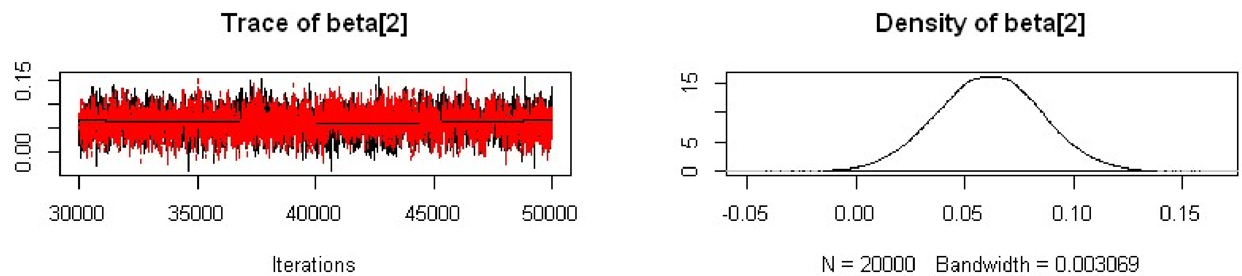
$$\sigma_a^2, \sigma_b^2 \sim InvGamma(0.1{,}0.1)$$

This non-linear regression model is appropriate to the data, which is neither linear nor transformable to linear. The splines serve to provide a fit that is continuous and flexible to the data.

Outlying data was observed at year_DOY (1986_167, 1992_177, and 2016_235). However, without any expert knowledge to support a decision, this data was retained.

MCMC Convergence

The MCMC algorithm cycles through candidate values for the distribution and accepts/rejects with a probability relative to the “legitimacy” of the previous candidate allowing for a thorough exploration of the posterior assuming sufficiently large iterations and convergence.



The trace and density plots above are an excerpt from the chosen model and are examples of excellent convergence; the chains (red and black) are in close agreement and the regular movement of the samples between 0 and 0.14 show that the posterior distribution was thoroughly explored.

Model Comparisons

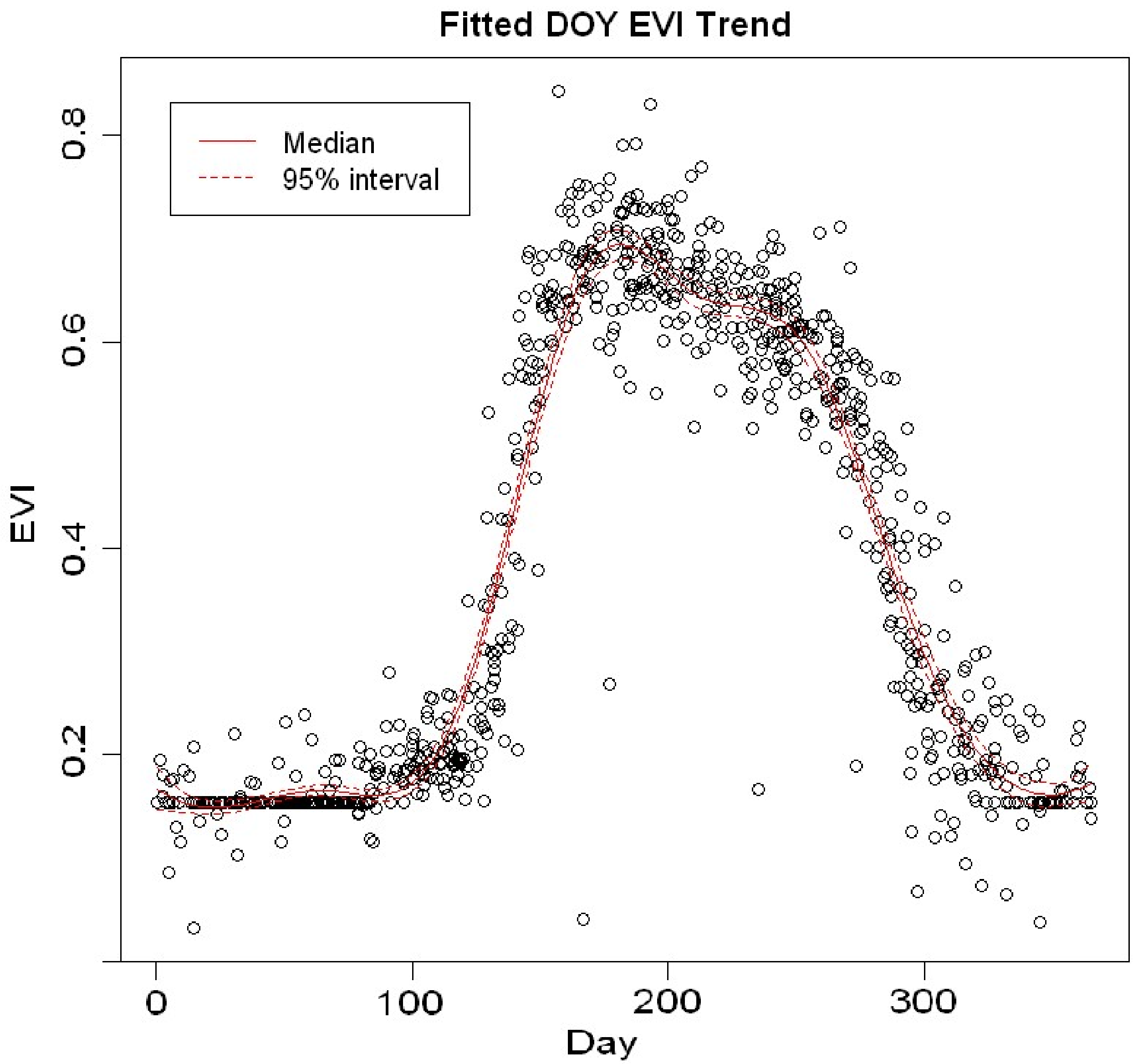
WAIC and DIC are two Bayesian metrics that allows for comparison of models for how well they fit while penalizing for complexity, where the complexity is informed by not only the number of parameters, but the strength of the priors.

Model	WAIC	DIC
4th Degree Polynomial	-1744.23	-1746
Splines (9 df) w/ varying log variance	-2375.15	-2412
Splines (9 df) w/ random effects for year	-2037.73	-2020

The non-linear spline model with 9 degrees of freedom and variable variance has the smallest WAIC and DIC scores and is thus determined to be the best fitting model.

Model Fit

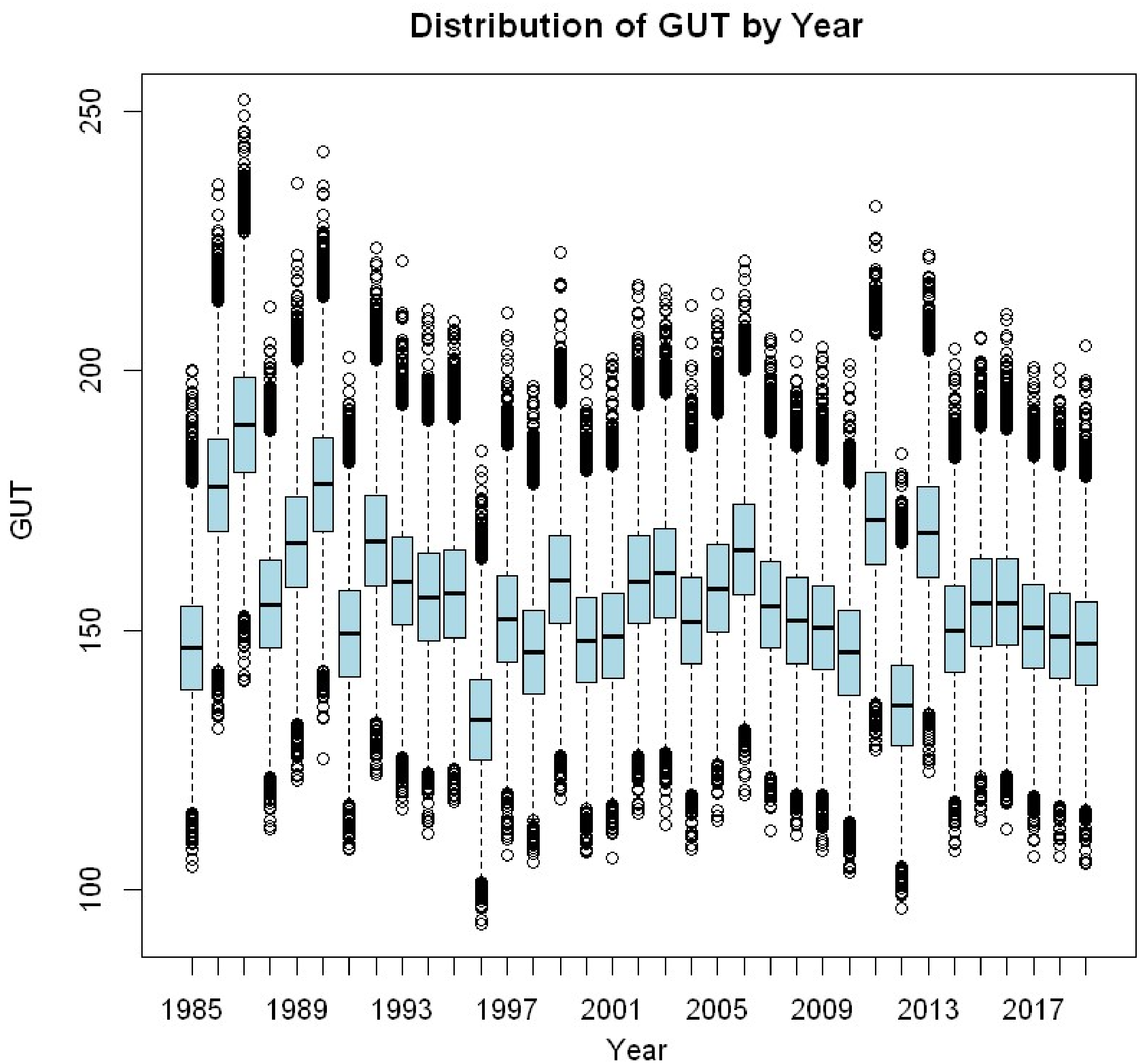
The EVI curve $u(t)$ is similar in shape to a normal distribution, but has some patterning (such as a slightly bimodal peak and long flat tails) that is well fit by the non parametric 9 spline model.



The model yields an excellent fit to the data with a narrow 95% confidence interval (represented by the dashed lines) and no evidence of overfitting.

GUT Analysis

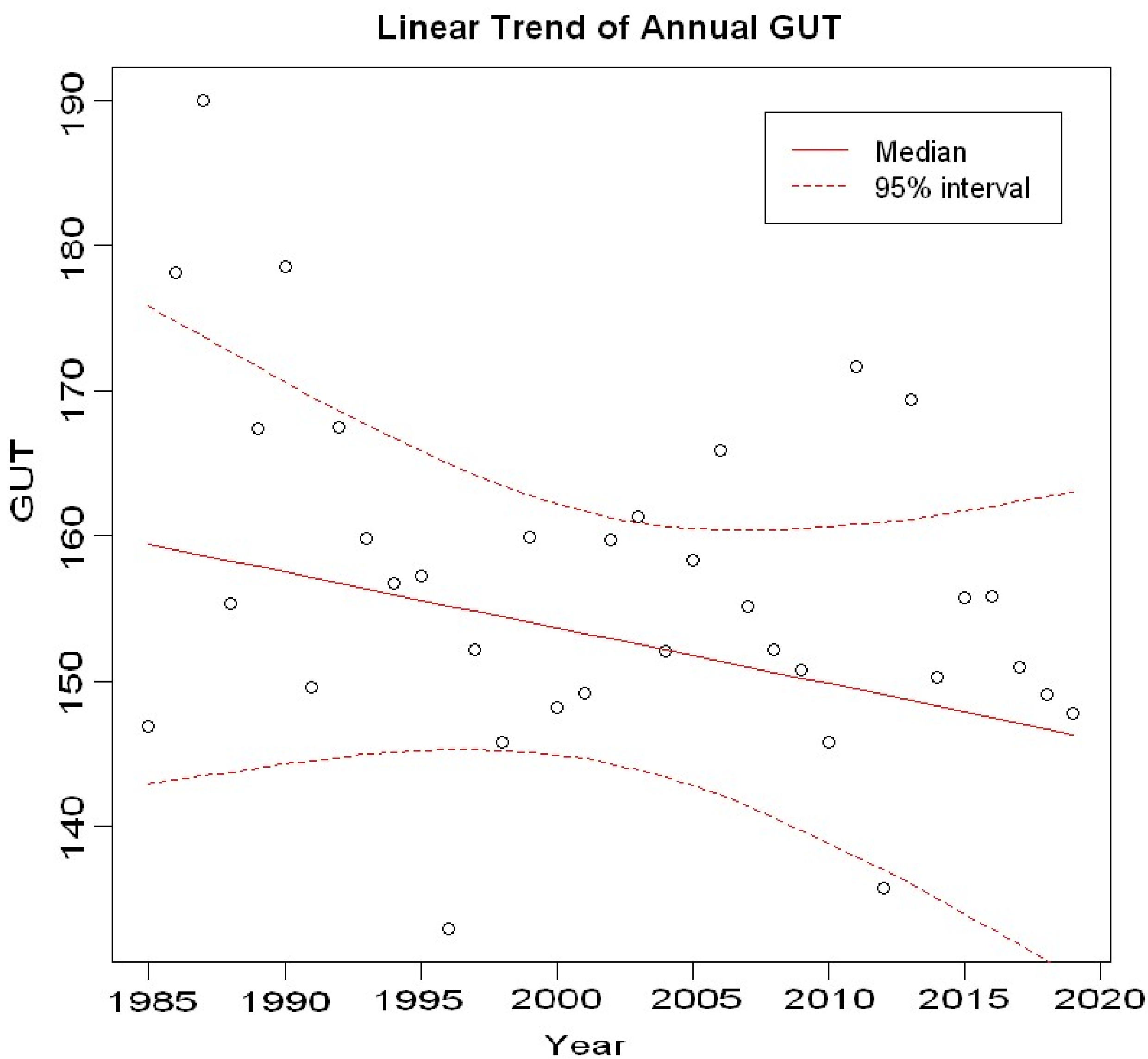
When limiting the data between days 1 and 220 (which is always inclusive of GUT, but exclusive of the end of year EVI decrease), annual GUT can be seen as a count of days with an EVI less than 0.5; therefore, it was modeled as following a Poisson distribution.



The posterior distributions of annual GUT are Poisson distributions with lambda averaging 0.71 and a standard deviation ranging from 11.5 to 13.7 days (mean standard deviation is 12.5 days).

Time-Trend Analysis

A simple linear model for GUT was constructed consisting of X as the year and Y (normally distributed) as GUT.



Although the model appears to indicate a declining trend in GUT year over year, the posterior of Beta2 (slope) includes 0 within the 95% confidence interval indicating that the slope could reasonably be zero (no change in GUT over the years).

APPENDIX – R Code

```
library(rjags)
library(dplyr)
library(MASS)
library(splines)
library(splines2)

########################################################
## Mean EVI by Year
########################################################

EVIdat <- read.csv('EVI_Data.csv')

EVImeanyr <-
aggregate(EVIdat$EVI,by=list(Year=EVIdat$Year),data=EVIdat,FUN=mean)

plot(EVImeanyr$Year,EVImeanyr$x,xlab="Year",ylab="EVI", main="Mean EVI by
Year",
     cex.lab=1,cex.axis=1)

Y <- EVImeanyr$x
X <- EVImeanyr$Year
X <- cbind(1,X)
n <- length(Y)

data  <- list(Y=Y,X=X,n=n, J=2)

model_lin_mean <- textConnection("model{

   # Likelihood
   for(i in 1:n){
      Y[i]     ~ dnorm(mean[i],taue)
      mean[i] <- mu + inprod(X[i,],beta[])
   }

   # Prior
   mu    ~ dnorm(0,0.01)
   taue ~ dgamma(0.1,0.1)
   for(j in 1:J){
    beta[j] ~ dnorm(0,taue) #*taub)
   }
   taub ~ dgamma(0.1,0.1)

  }")

inits  <- list(mu=mean(Y),beta=rep(0,2),taue=1/var(Y))
model_lmm <- jags.model(model_lin_mean, data = data, inits=inits, n.chains=2,
quiet=TRUE)

update(model_lmm, 100000, progress.bar="none")
samples <- coda.samples(model_lmm,
                        variable.names=c("beta", "mean"),
                        n.iter=200000, progress.bar="none")
tock <- proc.time()[3]
#summary(samples)

plot(samples)
```

```
sum <- summary(samples)
names(sum)
q <- sum$quantiles

plot(EVImeanyr$Year,Y,xlab="day",ylab="EVI", main="Linear Trend: Mean EVI by
Year",
     cex.lab=1.5,cex.axis=1.5)

lines(EVImeanyr$Year,q[3:38,1],col=2,lty=2) # 0.025 quantile (lower bound)
lines(EVImeanyr$Year,q[3:38,3],col=2,lty=1) # 0.500 quantile (median)
lines(EVImeanyr$Year,q[3:38,5],col=2,lty=2) # 0.975 quantile (upper bound)

legend("bottomright",c("Median","95% interval"),
       lty=1:2,col=2,bg=gray(1),inset=0.05,cex=1)


########################################################
## Polynomial Regression
########################################################

EVIdat <- EVIdat[order(EVIdat$DOY),]

Y <- EVIdat$EVI
X <- EVIdat$DOY
X <- scale(X)

X <- cbind(X, X*X, X*X*X, X^4, X^5)
n <- length(Y)

mod_poly <- "model{

    # Likelihood
    for(i in 1:n){
        Y[i]     ~ dnorm(mean[i],taue)
        mean[i] <- mu + inprod(X[i,],beta[])
    }

    # Prior
    mu    ~ dnorm(0,0.01)
    taue ~ dgamma(0.1,0.1)
    for(j in 1:J){
        beta[j] ~ dnorm(0,taue*taub)
    }
    taub ~ dgamma(0.1,0.1)

    # WAIC calculations
    for(i in 1:n){
    like[i] <- dnorm(Y[i],mean[i],taue)}
}"

dat     <- list(Y=Y,n=n,X=X,J=5)
init    <- list(mu=mean(Y),beta=rep(0,5),taue=1/var(Y))
modelp   <- jags.model(textConnection(mod_poly), inits=init,data = dat,
n.chains=2, quiet=TRUE)

update(modelp, 10000, progress.bar="none")

samps_poly  <- coda.samples(modelp, variable.names=c("like"), n.iter=20000,
progress.bar="none")
```

```
# Compute WAIC
like   <- rbind(samps_poly[[1]],samps_poly[[2]]) # Combine samples from the
two chains
fbar   <- colMeans(like)
Pw     <- sum(apply(log(like),2,var))
WAIC   <- -2*sum(log(fbar))+2*Pw

dicp <- dic.samples(modelp,n.iter=20000,progress.bar="none")
dicp

########################################################
## WAIC for Optimizing Spline degrees of freedom
########################################################

EVIdat <- EVIdat[order(EVIdat$DOY),]

Y <- EVIdat$EVI
X <- EVIdat$DOY
X <- scale(X)
n <- length(Y)

splines <- c(5,6,7,8,9,10,11)
chains <- 2
WAICres <- data.frame(matrix(ncol = 2, nrow = length(splines)))

mod1_spline <- "model{

    # Likelihood
    for(i in 1:n){
        Y[i]     ~ dnorm(mean[i],taue)
        mean[i] <- mu + inprod(B[i,],beta[])
    }

    # Prior
    mu    ~ dnorm(0,0.01)
    taue ~ dgamma(0.1,0.1)
    for(j in 1:J){
        beta[j] ~ dnorm(0,taue*taub)
    }
    taub ~ dgamma(0.1,0.1)

    # WAIC calculations
    for(i in 1:n){
    like[i] <- dnorm(Y[i],mean[i],taue)}
}"

for(s in 1:length(splines)){
  J <- splines[s]
  B <- bs(X, df = J)
  dat    <- list(Y=Y,n=n,B=B,J=J)
  init   <- list(mu=mean(Y),beta=rep(0,J),taue=1/var(Y))
  model1  <- jags.model(textConnection(mod1_spline), inits=init,data = dat,
n.chains=chains, quiet=TRUE)
  update(model1, 10000, progress.bar="none")
  samps_like  <- coda.samples(model1, variable.names=c("like"), n.iter=20000,
progress.bar="none")

  # Compute WAIC
```

```
  like   <- rbind(samps_like[[1]],samps_like[[2]]) # Combine samples from the
two chains
  fbar   <- colMeans(like)
  Pw     <- sum(apply(log(like),2,var))
  WAIC   <- -2*sum(log(fbar))+2*Pw

  WAICres[s,1] <- splines[s]
  WAICres[s,2] <- WAIC
}


########################################################
## Splines on DOY
########################################################

J <- 9
B <- bs(X, df = J)

dat    <- list(Y=Y,n=n,B=B,J=J)
init   <- list(mu=mean(Y),beta=rep(0,J),taue=1/var(Y))
model1 <- jags.model(textConnection(mod1_spline), inits=init,data = dat,
n.chains=2, quiet=TRUE)

update(model1, 10000, progress.bar="none")

samp <- coda.samples(model1, variable.names=c("mean"),
                     n.iter=20000, progress.bar="none")

sum <- summary(samp)
q <- sum$quantiles

plot(EVIdat$DOY,Y,xlab="Day of Year",ylab="EVI", main = "Spline Model of EVI
on DOY",
     cex.lab=1.5,cex.axis=1.5)

lines(EVIdat$DOY,q[,1],col=2,lty=2) # 0.025 quantile (lower bound)
lines(EVIdat$DOY,q[,3],col=2,lty=1) # 0.500 quantile (median)
lines(EVIdat$DOY,q[,5],col=2,lty=2) # 0.975 quantile (upper bound)

legend("topleft",c("Median","95% interval"),
       lty=1:2,col=2,bg=gray(1),inset=0.05,cex=1)

samplebeta   <- coda.samples(model1, variable.names=c("beta"), n.iter=20000,
progress.bar="none")

plot(samplebeta)

samps1_like  <- coda.samples(model1, variable.names=c("like"), n.iter=20000,
progress.bar="none")

# Compute WAIC
like   <- rbind(samps1_like[[1]],samps1_like[[2]]) # Combine samples from the
two chains
fbar   <- colMeans(like)
Pw     <- sum(apply(log(like),2,var))
WAIC   <- -2*sum(log(fbar))+2*Pw

########################################################
## Allowing Log Variance to Vary with X
```

```
########################################################

mod2_spline <- "model{

   # Likelihood
   for(i in 1:n){
      Y[i]          ~ dnorm(mean[i],inv_var[i])
      mean[i]      <- mu1 + inprod(B[i,],beta[])
      inv_var[i]   <- 1/sig2[i]
      log(sig2[i]) <- mu2 + inprod(B[i,],alpha[])
   }

   # Prior
   mu1   ~ dnorm(0,0.01)
   mu2   ~ dnorm(0,0.01)
   for(j in 1:J){
     beta[j]   ~ dnorm(0,taub)
     alpha[j] ~ dnorm(0,taua)
   }
   taua ~ dgamma(0.1,0.1)
   taub ~ dgamma(0.1,0.1)

   # Prediction intervals
   for(i in 1:n){
     low[i]  <- mean[i] - 1.96*sqrt(sig2[i])
     high[i] <- mean[i] + 1.96*sqrt(sig2[i])
   }

    # WAIC calculations
    for(i in 1:n){
    like[i] <- dnorm(Y[i],mean[i],inv_var[i])}
 }"

J <- 9
B <- bs(X, df = J)

dat    <- list(Y=Y,n=n,B=B,J=J)
init   <- list(mu1=mean(Y),beta=rep(0,J), mu2=log(var(Y)),alpha=rep(0,J))
model2  <- jags.model(textConnection(mod2_spline), inits=init,data = dat,
n.chains=2, quiet=TRUE)

update(model2, 10000, progress.bar="none")

samp2  <- coda.samples(model2,
                       variable.names=c("mean","sig2","low","high"),
                       n.iter=20000, progress.bar="none")

q2 <- summary(samp2)$quantiles
high <- q2[1:n+0*n,]
low  <- q2[1:n+1*n,]
mean <- q2[1:n+2*n,]
sig2 <- q2[1:n+3*n,]

plot(EVIdat$DOY,Y,xlab="Day",ylab="EVI",
     main="Fitted DOY EVI Trend",
     cex.lab=1.5,cex.axis=1.5)

lines(EVIdat$DOY,mean[,1],col=2,lty=2) # 0.025 quantile (lower bound)
lines(EVIdat$DOY,mean[,3],col=2,lty=1) # 0.500 quantile (median)
```

```
lines(EVIdat$DOY,mean[,5],col=2,lty=2) # 0.975 quantile (upper bound)

legend("topleft",c("Median","95% interval"),
       lty=1:2,col=2,bg=gray(1),inset=0.05,cex=1)

samps2_like  <- coda.samples(model2, variable.names=c("like"), n.iter=20000,
progress.bar="none")

# Compute WAIC
like    <- rbind(samps2_like[[1]],samps2_like[[2]]) # Combine samples from the
two chains
fbar    <- colMeans(like)
Pw      <- sum(apply(log(like),2,var))
WAIC    <- -2*sum(log(fbar))+2*Pw


dic2 <- dic.samples(model2,n.iter=20000,progress.bar="none")

########################################################
## Spline with Random Effects for Year
########################################################

yrN <- n_distinct(EVIdat$Year)

EVIdat$Year <- factor(EVIdat$Year)
EVIdat$Year <- droplevels(EVIdat$Year)
EVIdat$Year <- as.integer(EVIdat$Year)

yr <- EVIdat$Year
yrN <- n_distinct(EVIdat$Year)

mod_spline_randeff <- "model{

   # Likelihood
   for(i in 1:n){
      Y[i]     ~ dnorm(mnY[i],taue)
      mnY[i] <- inprod(X[i,],beta[id[i],])
   }

   # Random slopes
   for(j in 1:p){
     for(i in 1:N){
       beta[i,j] ~ dnorm(mu[j],taub[j])
     }
     mu[j]     ~ dnorm(0,0.01)
     taub[j]   ~ dgamma(0.1,0.1)
   }

   # Priors
   taue ~ dgamma(0.1,0.1)


   # WAIC calculations
   for(i in 1:n){
     like[i]     <- dnorm(Y[i],mnY[i],taue)
   }
  }"

dat     <- list(Y=Y,n=n,N=yrN,X=B,p=J,id=yr)
```

```
init   <- list(beta=matrix(0,yrN,J))
model3 <- jags.model(textConnection(mod_spline_randeff),n.chains=2,
                     inits=init,data = dat,quiet=TRUE)

update(model3, 10000, progress.bar="none")

samps3_like  <- coda.samples(model3, variable.names=c("like"), n.iter=20000,
progress.bar="none")

# Compute WAIC
like   <- rbind(samps3_like[[1]],samps3_like[[2]]) # Combine samples from the
two chains
fbar   <- colMeans(like)
Pw     <- sum(apply(log(like),2,var))
WAIC   <- -2*sum(log(fbar))+2*Pw

dic3 <- dic.samples(model3,n.iter=20000,progress.bar="none")


########################################################
## Extracting GUT
# Several attempts were made to run a full time series model in order to
extract GUT, but consistently froze the computer. These attempts are included
at the bottom of code
# Ultimately, the relatively linear trend between days 80 and 220 was
utilized to extract GUT
########################################################

EVIdat <- read.csv('EVI_Data.csv')
r <- n_distinct(EVIdat$Year) - 1
GUTdat <- data.frame(matrix(ncol = 2,nrow = r))
colnames(GUTdat) <- c("Year", "GUT")
iter <- 1

for (yr in 1985:2019){
  EVIyr <- EVIdat[which(EVIdat$Year == yr),]
  EVIyr <- EVIyr[which(EVIyr$DOY >= 80),]
  EVIyr <- EVIyr[which(EVIyr$DOY <= 220),]
  Y <- EVIyr$EVI
  X <- EVIyr$DOY
  lmod  <- lm(Y ~ X)
  matrix_coef <- summary(lmod)$coefficients
  intercept <- matrix_coef[1,1]
  slope <- matrix_coef[2,1]
  GUTdat[iter,1] <- yr
  GUTdat[iter,2] <- (0.5-intercept)/slope
  iter <- iter + 1
}


########################################################
## GUT Posterior by Year
########################################################

Y <- GUTdat$GUT
n <- length(Y)
yrs <- GUTdat$Year
N <- 220
```

```
model_gutp <- textConnection("model{
    # Likelihood
    for(i in 1:n){
        Y[i] ~ dpois(N*lambda[i])
    }
    # Priors
    for(i in 1:n){
        lambda[i] ~ dgamma(1,gamma)
    }
    gamma    ~  dgamma(a, b)

    for(i in 1:n){
        GUT[i] <- dpois(1,N*lambda[i])
    }
 }")

inits <- list(lambda=rgamma(n,1),gamma=1)
data  <- list(Y=Y,N=N,n=n,a=0.1,b=0.1)
model <- jags.model(model_gutp,data = data, inits=inits, n.chains=1,
quiet=TRUE)

samples <- coda.samples(model,
                        variable.names=c("lambda"),
                        n.iter=20000, progress.bar="none")

samps <- samples[[1]]
gut <- data.frame(matrix(ncol = n,nrow = 20000))

for (i in 1:n){
  gut[,i] <- as.vector(samps[,i])*220
}

colnames(gut) <- c(yrs)

library(reshape)

gutdata <- melt(gut)

boxplot(data=gutdata,value~variable, col="lightblue",
        xlab="Year", ylab="GUT", main="Distribution of GUT by Year")

########################################################
## Linear Regression GUT over years
########################################################

Y <- GUTdat$GUT
X <- GUTdat$Year
X <- scale(X)
X <- cbind(1,X)
n <- length(Y)

data  <- list(Y=Y, X=X, n=n, J=2)

model_lingut <- textConnection("model{

   # Likelihood
   for(i in 1:n){
      Y[i]    ~ dnorm(mean[i],taue)
      mean[i] <- mu + inprod(X[i,],beta[])
```

```
    }

    # Prior
    mu   ~ dnorm(0,0.01)
    taue ~ dgamma(0.1,0.1)
    for(j in 1:J){
     beta[j] ~ dnorm(0,taue) #*taub)
    }
    taub ~ dgamma(0.1,0.1)

  }")

inits  <- list(mu=mean(Y),beta=rep(0,2),taue=1/var(Y))
model_gut <- jags.model(model_lingut, data = data, inits=inits, n.chains=2,
quiet=TRUE)

update(model_gut, 10000, progress.bar="none")
samples_beta <- coda.samples(model_gut,
                             variable.names=c("beta"),
                             n.iter=20000, progress.bar="none")
samples_mean <- coda.samples(model_gut,
                             variable.names=c("mean"),
                             n.iter=20000, progress.bar="none")

plot(samples_beta)

sum <- summary(samples_mean)
q <- sum$quantiles

plot(GUTdat$Year,Y,xlab="Year",ylab="GUT", main="Linear Trend of Annual GUT",
     cex.lab=1.5,cex.axis=1.5)

lines(GUTdat$Year,q[,1],col=2,lty=2) # 0.025 quantile (lower bound)
lines(GUTdat$Year,q[,3],col=2,lty=1) # 0.500 quantile (median)
lines(GUTdat$Year,q[,5],col=2,lty=2) # 0.975 quantile (upper bound)

legend("topright",c("Median","95% interval"),
       lty=1:2,col=2,bg=gray(1),inset=0.05,cex=1)

########################################################
## Frequentist Timeseries
########################################################

EVIdat <- read.csv('EVI_Data.csv')
EVIdat$Date<-
as.Date(with(EVIdat,paste(EVIdat$Year,EVIdat$Month,EVIdat$Day,sep="-")),"%Y-
%m-%d")
EVIdat$DateN<-as.numeric(EVIdat$Date)

keeps <- c("Date", "EVI")
TSdata <- EVIdat[keeps]

fit <- stl(TSdata)
plot(fit)

plot.ts(TSdata)
abline(h=0.5, col="blue")
```

```
########################################################
## Full Timeseries
########################################################

EVIdat <- read.csv('EVI_Data.csv')
EVIdat <- EVIdat[order(EVIdat$DateVal),]
EVIdat$DateValScaled <- scale(EVIdat$DateVal)

Y <- EVIdat$EVI
X <- EVIdat$DateValScaled
n <- length(Y)

J   <- 324
B   <- bs(X, df = J)
dat <- list(Y=Y,n=n,B=B,J=J)

init   <- list(mu=mean(Y),beta=rep(0,J),taue=1/var(Y))
model1 <- jags.model(textConnection(mod1_spline), inits=init,data = dat,
quiet=TRUE)
update(model1, 10000, progress.bar="none")

samp_full <- coda.samples(model1, variable.names=c("mean"), n.iter=10000)

sum <- summary(samp)
names(sum)
q <- sum$quantiles

plot(EVIdat$DateVal,Y,xlab="day",ylab="EVI",
     cex.lab=1.5,cex.axis=1.5)
abline(v=0.5, col="blue")
lines(EVIdat$DateVal,q[,3],col=2,lty=1) # 0.500 quantile (median)

legend("bottomright",c("Median","95% interval"),
       lty=1:2,col=2,bg=gray(1),inset=0.05,cex=1.5)
```