ST 540: Midterm 2. . . . . John Williams. 4/14/2022

1. Model definition Our goal is to analyze the "green-up time" (GUT) which occurs in the spring. We assume that the growth curve in the first half of each year is only dependent on data in the winter, spring, and early summer. We therefore subset the data for each year to include only data points between December 1 - August 15. Data points from August 16 - November 30 are removed. Data points in the month of December are attached to the following year with negative day-of-year (DOY) values; this provides more reference data in the winter months for years where data points are slim to none.

Model 1: Let Y_ij be the j^{th} observation of EVI in year i. The model is

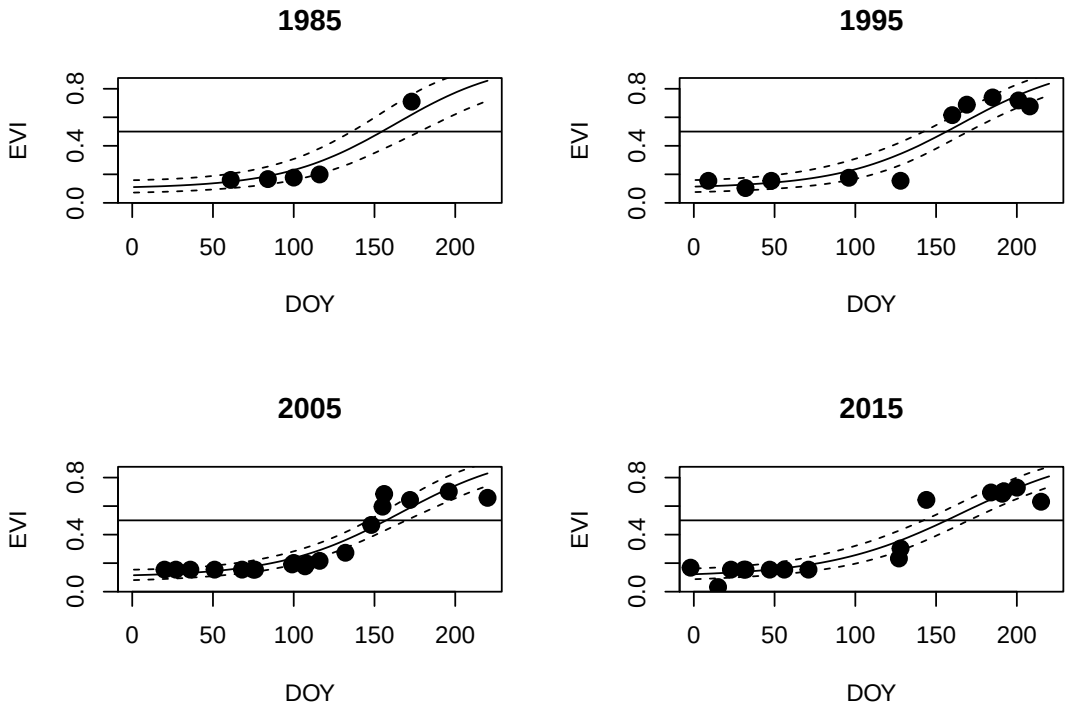
$$Y_{ij} \sim \text{Normal}\left(\mu_i\left(t_{ij}\right), \sigma_\varepsilon^2\right)$$

where $\mu_i\left(t_{ij}\right)$ is the mean growth curve for each year i and σ_ε^2 is the error variance which is constant across all years. For Model 1, we assume the mean growth curve is logistic:

$$\mu_i\left(t_{ij}\right) = a_i + \frac{b_i}{1 + e^{-(t_{ij} - c_i)/d_i}}$$

The growth curve for year i is determined by four parameters: a_i, b_i, c_i, and d_i. Since a_i is the curve's minimum value and b_i represents the curve's maximum value, we set $b_i = 1 - a_i$ **to ensure the inflection point of the logistic curve is at EVI = 0.5**–this is key to our examination of GUT as the GUT for year i is simply c_i. The last parameter d_i is the rate of increase. For the curve to be positive and increasing across DOY, a_i and d_i must be positive. Thus, we represent the growth-curve parameters in terms of unconstrained parameters α_i1, α_i2, and α_i3 as $a_i = e^{\alpha_{i1}}$, $c_i = \alpha_{i2}$, and $d_i = e^{\alpha_{i3}}$. We use uninformative priors separately by year: $\alpha_{ik} \sim \text{Normal}\left(\theta_k, \sigma_k^2\right)$ where $\theta_k \sim \text{Normal}\left(0, 100^2\right)$ and $\sigma_k \sim \text{InvGamma}\left(0.1, 0.1\right)$. Finally, the prior for the error variance is $\sigma_e \sim \text{InvGamma}\left(0.1, 0.1\right)$.

Here's a view of how well Model 1 fits the data for selected years:



2. MCMC convergence In an MCMC algorithm, the first few samples are likely not draws from the "true" posterior distribution. Many iterations of the sampler are need to reach convergence, the point when the sampler "truly" begins sampling from the posterior. Of course, we can look at the trace plots of each parameter to determine if the MCMC algorithm has converged (do they look like a fuzzy caterpillar?) But to numerically verify that the MCMC chain has converged, we consider the expected sample size (ESS) and Geweke's statistic. **The minimum ESS (5,296) is greater than 1,000; the absolute maximum Geweke's statistic (z = 1.89) is less than 2.** Thus, based on these two "rules of thumb", we can be confident the MCMC algorithm has converged.

3. Model comparisons Model 2 differs from Model 1 in one aspect, the mean curve is modeled as a cubic polynomial for each year i:

$$\mu_i\left(t_{ij}\right) = a_i + b_i t_{ij} + c_i t_{ij}^2 + d_i t_{ij}^3$$

The priors for the parameters a_i, b_i, c_i, and d_i (indexed by $k = \{1, 2, 3, 4\}$, respectively) for year i are Normal $\left(\theta_k, \sigma_k^2\right)$ where $\theta_k \sim$ Normal $\left(0, 100^2\right)$ and $\sigma_k \sim$ InvGamma $(0.1, 0.1)$.

Here's a view of how well Model 2 fits the data for selected years:

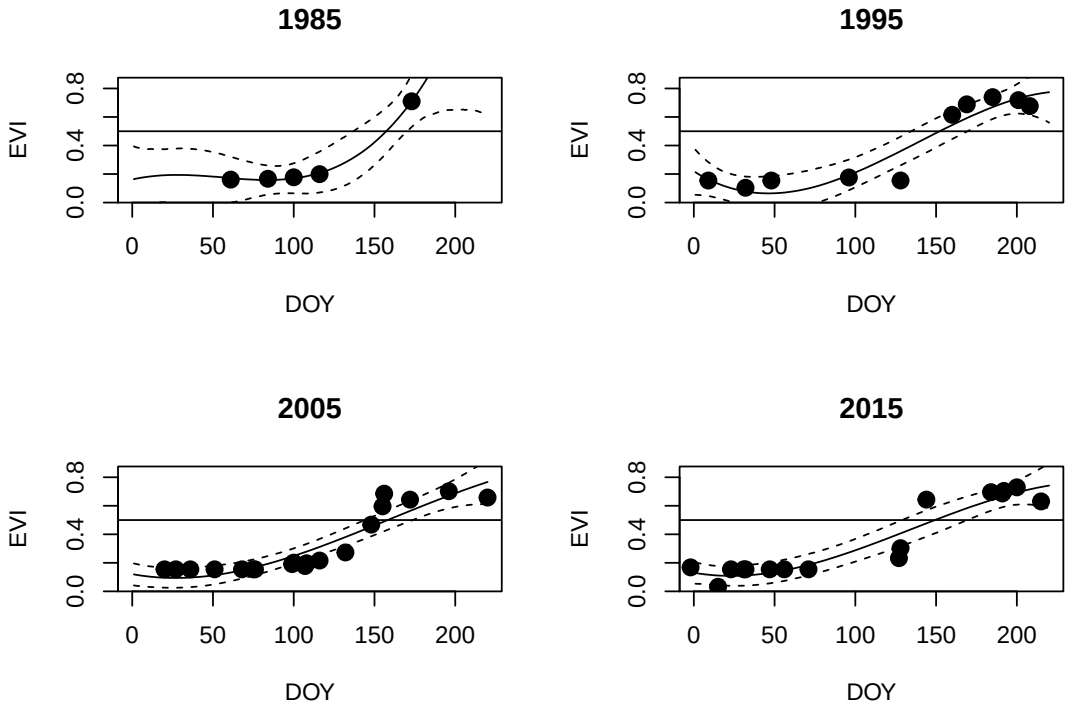
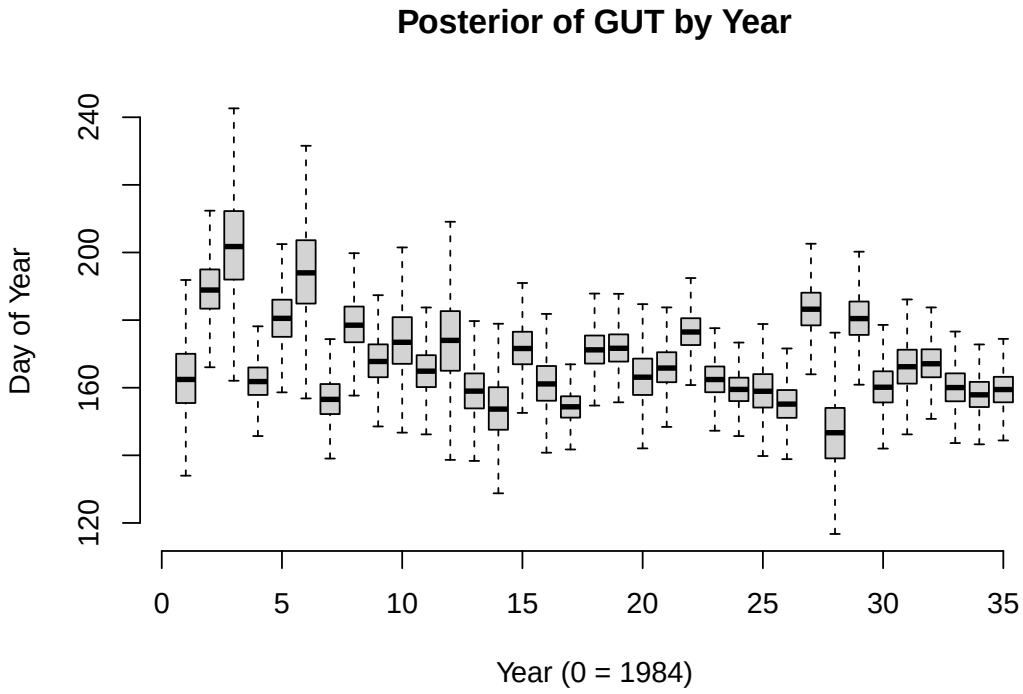


Table 1: DIC (Deviance Information Criteria) for Model 1 and Model 2

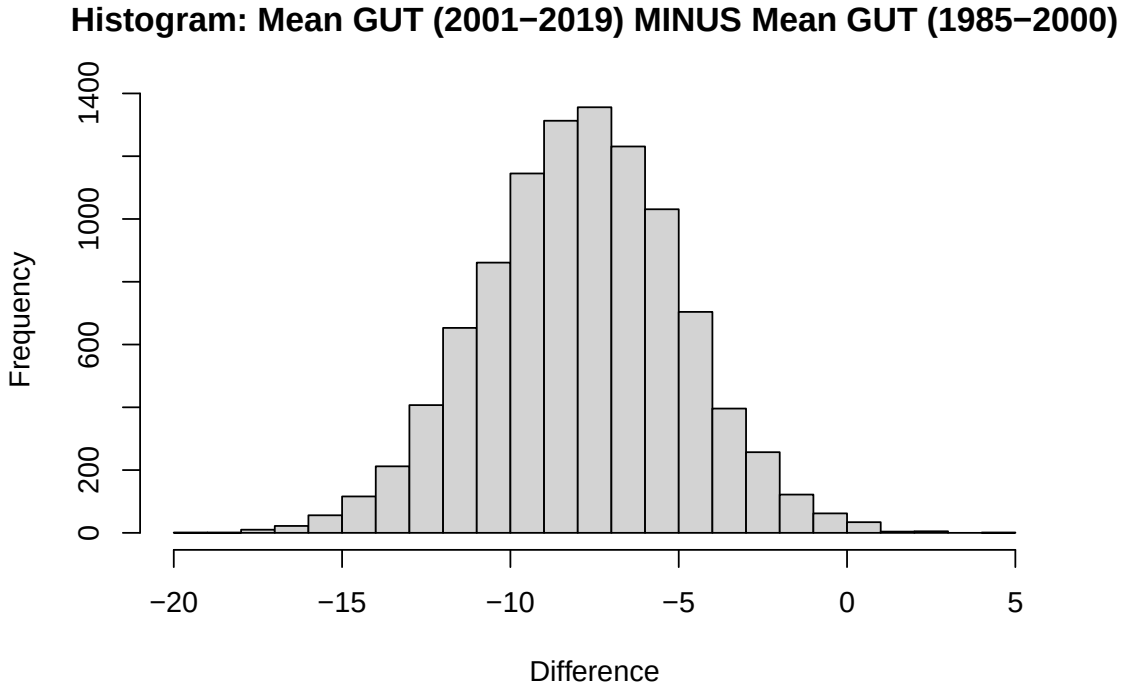
	Mean Deviance	Penalty	Penalized Deviance
Model 1	-1030	59.25	-970.3
Model 2	-1016	128.8	-887.3

Model 1 fits the data *slightly* better than Model 2 since it has a lower mean deviance. But, Model 2 is more complicated than Model 1 as it has a larger penalty, p_D. The penalized deviance balances out complexity and fit to reveal the best performing model overall: Model 1.

5. GUT analysis We summarize the posterior distribution of GUT by year in the boxplot below.



6. Time-trend analysis: To determine if there is a change in the distribution of GUT across the years, let's examine the difference in the mean GUT from 1985 to 2000 and the mean GUI from 2001 to 2019. If the distribution of this difference is NOT centered around zero, then we can conclude the distribution of GUT has changed over time.



The histogram above shows the center of the distribution of the difference is shifted to the left of zero; and the probability that the mean GUT has decreased by 1 full day or more to be 0.9894. Thus, we are confident that GUT has decreased across the years.

Appendix: All code for this report

```
library(tidyverse)
library(rjags)

# Download the data
EVI.data <- read.csv("C:/ST540/EVI_Data.csv")

# Subset data such that only data points between December 1 - August 15 are
# retained for each year. Data points from August 16 - November 30 are removed.
# Data points in the month of December are attached to the following year with
# negative day-of-year (DOY) values.
EVI.data2 <- EVI.data %>%
             mutate(Year = ifelse(Month == 12, Year + 1, Year)) %>%
             mutate(DOY = ifelse(Month == 12, Day - 32, DOY)) %>%
             filter(DOY < 221, Year > 1984)
set.seed(31618)  # For reproducibility

# Set the number of iterations for the MCMC sampler
burn    <- 25000
iters   <- 50000
thin    <- 10

# Load the data
y <- EVI.data2$EVI
x <- EVI.data2$DOY
sp <- as.numeric(as.factor(EVI.data2$Year))
n <- length(y)

# Specify the model
model_string1 <- textConnection("model{

  for(i in 1:n){
    y[i]     ~ dnorm(muY[i],tau[5])
    muY[i]  <- a[sp[i]] + (1-a[sp[i]])/(1+exp(-part[i]))
    part[i] <- (x[i]-c[sp[i]])/d[sp[i]]
  }

  for(j in 1:N){
    a[j]   <- exp(alpha[j,2])
    c[j]   <- alpha[j,3]
    d[j]   <- exp(alpha[j,4])

    for(k in 2:4){
      alpha[j,k]   ~ dnorm(mu[k],tau[k])
    }
  }

  for(j in 2:4){
    mu[j]  ~ dnorm(0,0.01)
    tau[j] ~ dgamma(0.1,0.1)
  }
```

```
  tau[5] ~ dgamma(0.1,0.1)

  for(DOY in 1:220){
    for(j in 1:N){
      PART[DOY,j]   <- (DOY-c[j])/d[j]
      fitted[DOY,j] <- a[j] + (1-a[j])/(1+exp(-PART[DOY,j]))
    }
  }

  mean[1] <- (c[1] + c[2] + c[3] + c[4] + c[5] + c[6] + c[7] + c[8] + c[9] +
              c[10] + c[11] + c[12] + c[13] + c[14] + c[15] + c[16])/16
  mean[2] <- (c[17] + c[18] + c[19] + c[20] + c[21] + c[22] + c[23] + c[24] +
              c[25] + c[26] + c[27] + c[28] + c[29] + c[30] + c[31] + c[32] +
              c[33] + c[34] + c[35])/19
  z <- mean[2] - mean[1]

}")

data    <- list(y=y,x=x,sp=sp,n=n,N=35)
model1   <- jags.model(model_string1,data = data,quiet=TRUE, n.chains=2)
update(model1, burn, progress.bar="none")

samples1 <- coda.samples(model1,
                         variable.names=c("a","c","d","fitted","tau", "z"),
                         n.iter=iters,
                         thin=thin,
                         progress.bar="none")

summ1 <- summary(samples1)

fit1 <- summ1$quantiles[1:7700+105,3]
lo1  <- summ1$quantiles[1:7700+105,1]
hi1  <- summ1$quantiles[1:7700+105,5]
id   <- rep(1:35,each=220)

par(mfrow=c(2,2))
year_names <- c(1985:2019)
for(j in c(1,11,21,31)){
  plot(NA,xlim=range(0:220),ylim=range(y),
       xlab="DOY", ylab="EVI", main=year_names[j])
  points(x[sp==j],y[sp==j],pch=19,cex=1.5)
  abline(h=0.5)
  lines(1:220,fit1[id==j],lty=1)
  lines(1:220,lo1[id==j],lty=2)
  lines(1:220,hi1[id==j],lty=2)
}
set.seed(31618)  # For reproducibility

ESS <- effectiveSize(samples1)
ESS2 <- ESS[-(1:7700+105)]
ESS2[which.min(ESS2)]

geweke <- geweke.diag(samples1[[1]])
```

```
geweke2 <- geweke$z[-(1:7700+105)]
geweke2[which.max(abs(geweke2))]
set.seed(31618)  # For reproducibility

# Set the number of iterations for the MCMC sampler
burn    <- 25000
iters   <- 50000
thin    <- 10

# Load the data
y <- EVI.data2$EVI
x <- EVI.data2$DOY
sp <- as.numeric(as.factor(EVI.data2$Year))
n <- length(y)

# Specify the model
model_string2 <- textConnection("model{

  for(i in 1:n){
    y[i]     ~ dnorm(muY[i],tau[5])
    muY[i]  <- a[sp[i]] + b[sp[i]]*x[i] + c[sp[i]]*pow(x[i],2) + d[sp[i]]*pow(x[i],3)
  }

  for(j in 1:N){
    a[j]   <- alpha[j,1]
    b[j]   <- alpha[j,2]
    c[j]   <- alpha[j,3]
    d[j]   <- alpha[j,4]

    for(k in 1:4){
      alpha[j,k]   ~ dnorm(mu[k],tau[k])
    }
  }

  for(j in 1:4){
    mu[j]  ~ dnorm(0,0.01)
    tau[j] ~ dgamma(0.1,0.1)
  }

  tau[5] ~ dgamma(0.1,0.1)

  for(DOY in 1:220){
    for(j in 1:N){
      fitted[DOY,j] <- a[j] + b[j]*DOY + c[j]*pow(DOY,2) + d[j]*pow(DOY,3)
    }
  }
}")

data    <- list(y=y,x=x,sp=sp,n=n,N=35)
model2   <- jags.model(model_string2,data = data,quiet=TRUE, n.chains=2)
update(model2, burn, progress.bar="none")

samples2 <- coda.samples(model2,
```

```
                          variable.names=c("a","b","c","d","fitted","tau"),
                          n.iter=iters,
                          thin=thin,
                          progress.bar="none")

summ2 <- summary(samples2)

fit2 <- summ2$quantiles[1:7700+140,3]
lo2  <- summ2$quantiles[1:7700+140,1]
hi2  <- summ2$quantiles[1:7700+140,5]
id   <- rep(1:35,each=220)

par(mfrow=c(2,2))
year_names <- c(1985:2019)
for(j in c(1,11,21,31)){
  plot(NA,xlim=range(0:220),ylim=range(y),
       xlab="DOY", ylab="EVI", main=year_names[j])
  points(x[sp==j],y[sp==j],pch=19,cex=1.5)
  abline(h=0.5)
  lines(1:220,fit2[id==j],lty=1)
  lines(1:220,lo2[id==j],lty=2)
  lines(1:220,hi2[id==j],lty=2)
}
set.seed(31618)  # For reproducibility

# Compute DIC
dic1 <- dic.samples(model1, n.iter = iters, progress.bar = "none")
dic2 <- dic.samples(model2, n.iter = iters, progress.bar = "none")
GUT <- rbind(samples1[[1]],samples1[[2]])[,1:35+35]

boxplot(GUT, xlab="Year (0 = 1984)", ylab="Day of Year",
        main="Posterior of GUT by Year", outline=FALSE, axes=FALSE)
axis(1)
axis(2)
diff <- rbind(samples1[[1]],samples1[[2]])[,7810]

hist(diff,
     breaks = 20,
     xlab="Difference",
     main="Histogram: Mean GUT (2001-2019) MINUS Mean GUT (1985-2000)")
```