Model Definition:

The enhanced vegetation index (EVI) is a satellite-derived measurement that quantifies vegetation greenness at a certain period in time. For this report, the data contains EVI measurements for a single spatial location from 1984 to 2019. Let Y_i represent the EVI measurement at time t_i and $\mu(t)$ be the true EVI curve. Y_i values for this dataset, which are noisy values of $\mu(t_i)$, range from 0 to 1. Since the support Y_i is the interval (0,1), the best likelihood is the beta distribution. Additionally, many papers have proposed a double-logistic function to model $\mu(t)$ (Elmore et al., 2012; Melass et al., 2013; Gao et al., 2021). Therefore, this report will expand on previous research and examine the following model (denoted as Model 1):

$$Y_i \mid \mu(t_i) \sim \text{Beta}[r*\mu(t_i) , r*(1- \mu(t_i))]$$

$$\mu(t_i) = m_{1j} + (m_{2j} - m_7 t_i)\left[\frac{1}{1+exp((m_{3j}-t_i)*m_{4j})} - \frac{1}{1+exp((m_{5j}-t_i)*m_{6j})}\right]$$

$m_{1j} \sim U(0,1)$, seasonal minimum greenness

$m_{2j} \sim U(m_{1j},1)$, seasonal greenness amplitude

$m_{3j} \sim U(1,366)$, start of spring

$m_{5j} \sim U(m_{3j},366)$, start of autumn

$m_{4j} , m_{6j} \sim N(0,10^2)$, slopes of spring and autumn

$m_{7j} \sim N(0,10^2)$, decrease in EVI during summer

$r_j \sim \text{Gamma}(0.1,0.1)$, controls concentration around $\mu(t_i)$ for Y_i

We use uninformative priors for each of the variables within the true EVI curve. For several of the variables, we introduce a restriction in the support. For example, the start of autumn must come after the start of spring. Also, note that all priors will vary by year j and t_i represents the DOY. Finally, we defined $Y_i \mid \mu(t_i)$ such that $E[Y_i \mid \mu(t_i)] = \mu(t_i)$.

MCMC Convergence:

For the model described in the previous section, we will utilize the software package RJAGS to perform the Metropolis-Hastings algorithm in order to sample from the posterior distribution of $\mu(t_i) \mid Y_i$ and ultimately assess the convergence of prior distributions in the model.

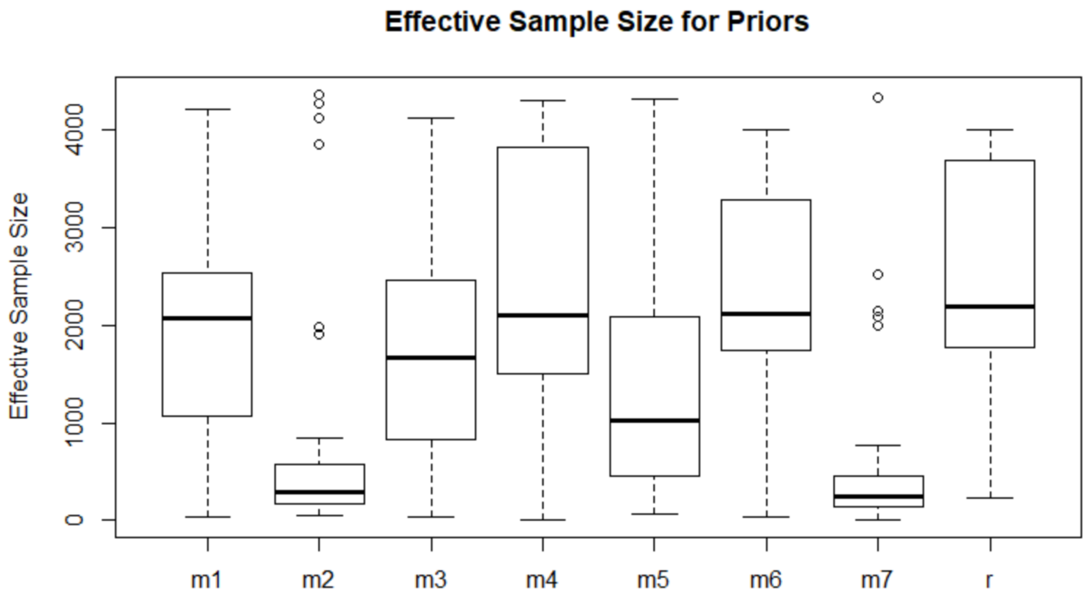

Figure 1 summarizes the distribution of the effective sample size from 1984-2019 for each prior distribution. Ultimately, we see both the m_2 and m_7 distributions are entirely less than 1000.

Using 10,000 burn-in samples and 20,000 iterations, we obtain effective sample size and Gelman-Rubin values (not shown) that indicate the m_1, m_3, m_4, m_5, m_6, and r priors converge for most of the years from 1984-2019, but the m_2 and m_7 priors do not have the best convergence.

Model Comparisons:

Model 1 will be compared to two simpler models using DIC and WAIC metrics; both models have the same likelihood for Y_i, but Model 2 has $logit(\mu(t_i)) = b_{0j} + b_{1j}t_i + b_{2j}t_i^2$ and Model 3 has $logit(\mu(t_i)) = b_{0j}sin(b_{1j}(t_i-b_{2j})) + b_{3j}$ where $b_{ij} \sim N(0,10^2)$ and j is the year for both models.

Metric	Model 1	Model 2	Model 3
DIC	-1932	-1267	-1616
DIC Penalty	596	208	652
Penalized Deviance	-1336	-1059	-964
WAIC	-1729	-1119	-1498
P_W	139	127	129

Table 1 shows the calculated DIC and WAIC metrics for all 3 models

Model 1 has smaller DIC and WAIC values compared to Models 2 and 3 and has only a slightly larger effective model size, therefore we conclude that our initial model is the best fit for the data.

Model Fit:

Using the previously described MCMC algorithm, we obtain estimates of the priors for each year and use these values to estimate the yearly true EVI curve based on the function $\mu(t_i)$ given in Model 1.

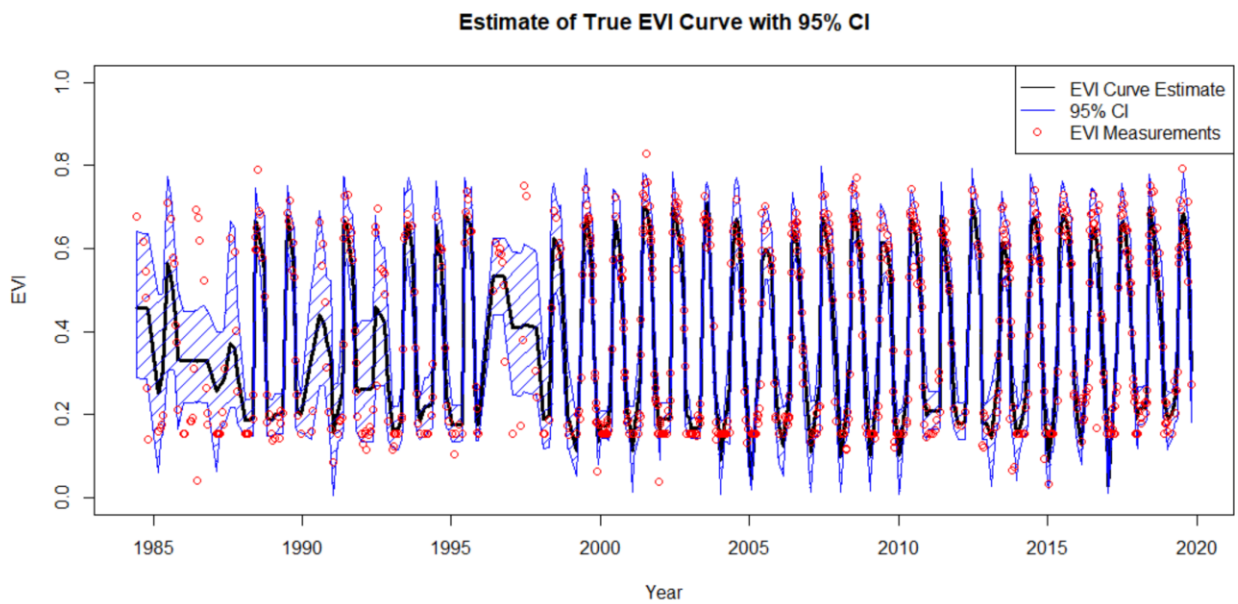

Figure 2 shows the estimate of $\mu(t)$ for the entire dataset along with a 95% confidence interval. The 802 EVI measurements are also shown to get a sense of how well the model fits the data.

After plotting the estimate of $\mu(t)$, we see that the model fits the dataset fairly well except for years where there are fewer EVI measurements; for years with a lower number of measurements, we notice higher uncertainty by observing the wider confidence interval for these periods.

GUT Analysis:

For each iteration of the MCMC algorithm, we can use our current estimate of $\mu(t)$ to approximate the yearly GUTs and ultimately use these values over all iterations to summarize the posterior distribution of GUT for each year.

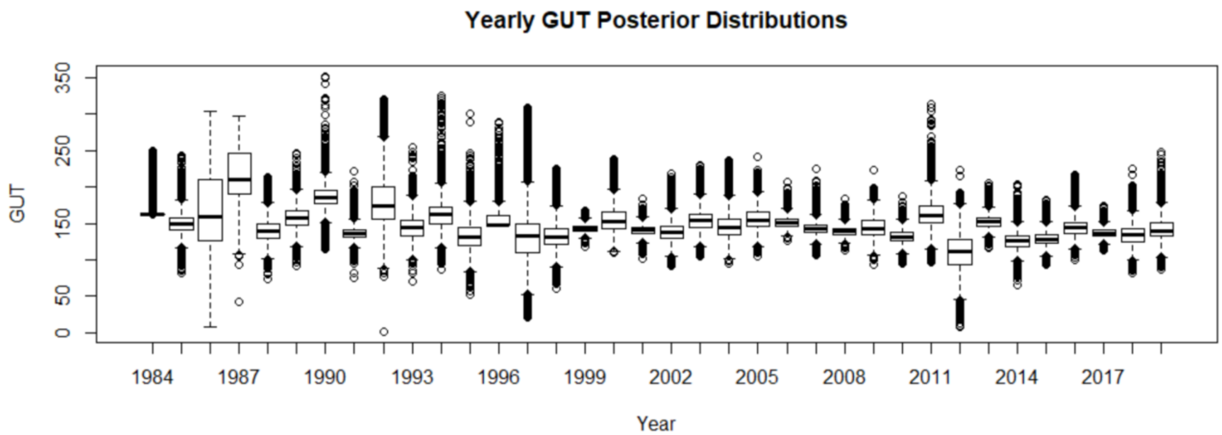

Figure 3 shows the posterior distribution of GUT by year. Note that not all iterations will have μ(t) > 0.5 during a specific year, therefore this analysis only considers iterations where the estimate of μ(t) has a GUT.

Using 20,000 iterations and estimates of μ(t), we obtain the GUT posterior distributions by year and notice that there is generally more uncertainty in the earlier years (1984-1997) compared to more recent years (1998-2019) since there is less data from the early years to train the model on.

Time-trend Analysis:

After finding the posterior distribution of GUT for each year, we will fit a linear regression model on the medians of the distributions to see how the median GUT value change across the years.

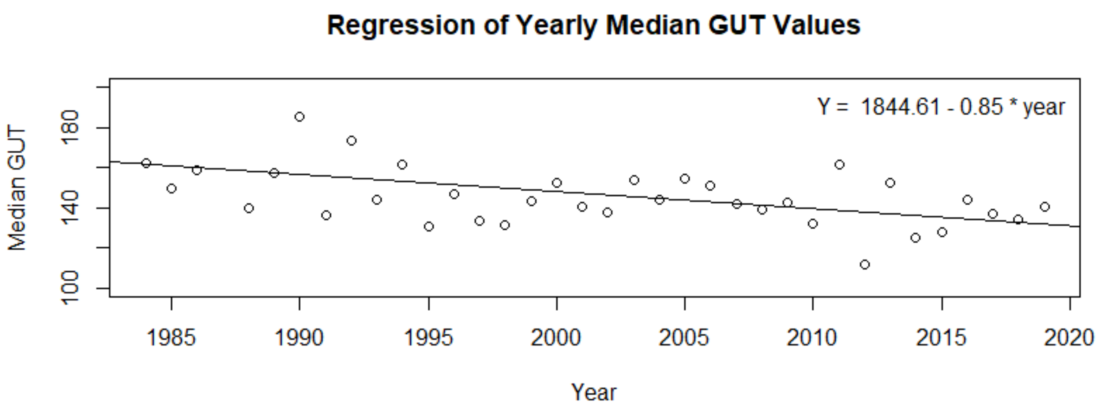

Figure 4 shows the regression of median GUT values from 1984-2019 along with the regression equation.

From the regression, we obtain a slope of -0.85 with a p-value of 0.002 which indicates the median GUT has decreased across the years, therefore vegetation is getting greener earlier in the year.

Works Cited

Elmore, A.J., Guinn, S.M., Minsley, B.J., Richardson, A.D., 2012. Landscape controls on the timing of spring, autumn, and growing season length in mid-Atlantic forests. Glob. Chang. Biol. 18 (2), 656–674. https://doi.org/10.1111/j.1365- 2486.2011.02521.x.

Melaas, E.K., Friedl, M.A., Zhu, Z., 2013. Detecting interannual variation in deciduous broadleaf forest phenology using Landsat TM/ETM+ data. Remote Sens. Environ. 132, 176–185. https://doi.org/10.1016/j.rse.2013.01.011.

Xiaojie Gao, Josh M. Gray, Brian J. Reich. Long-term, medium spatial resolution annual land surface phenology with a Bayesian hierarchical model. Remote Sensing of Environment, Volume 261, 2021. https://doi.org/10.1016/j.rse.2021.112484.

Code

```
setwd('C:/Users/ryant/Documents/Statistics/ST540/Midterm_2')

library(rjags)

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

Y <- df$EVI

t <- df$DOY

n <- length(Y)

yrs <- df$Year - 1983

unique_years <- unique(df$Year)

k <- length(unique_years)

continuous_time <- df$Year + df$DOY/366


#Fit model

model_string <- textConnection("model{

        for (i in 1:n){

          Y[i] ~ dbeta(r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))

          mu[i] <- m1[yrs[i]] + (m2[yrs[i]]-m7[yrs[i]]*t[i])*(1/(1+exp((m3[yrs[i]]-
t[i])*m4[yrs[i]])) - 1/(1+exp((m5[yrs[i]]-t[i])*m6[yrs[i]])))

        }


        #for each i, we have to account for the year

        for (j in 1:k){

          m1[j] ~ dunif(0,1)

          m2[j] ~ dunif(m1[j],1)

          m3[j] ~ dunif(1,366)

          m4[j] ~ dnorm(0,.01)

          m5[j] ~ dunif(m3[j],366)

          m6[j] ~ dnorm(0,.01)

          m7[j] ~ dnorm(0,.01)

          r[j] ~ dgamma(.1,.1)
```

```
        }
        #WAIC Calculations
        for(i in 1:n){
          like[i] <- dbeta(Y[i],r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))
        }
}")
data = list(Y=Y,t=t,n=n,yrs=yrs,k=k)
model <- jags.model(model_string,data=data,n.chains=2)

#Burn in 10000 samples
update(model, 10000)

#Set paramaters and get samples
params = c('m1','m2','m3','m4','m5','m6','m7','r')
samples <- coda.samples(model, variable.names=params, n.iter=20000,thin=10)
mu_samples <- coda.samples(model,variable.names=c('mu'),n.iter=20000)
summary(samples)
summary(mu_samples)

#Compute DIC
DIC <- dic.samples(model,n.iter=20000,n.thin=10)
#Compute WAIC
waic   <- coda.samples(model,variable.names=c("like"), n.iter=20000,n.thin=10)
like   <- waic[[1]]
fbar   <- colMeans(like)
P      <- sum(apply(log(like),2,var))
WAIC   <- -2*sum(log(fbar))+2*P

#Compute effective sample size and gelman stat for each year
sample_size <- list()
```

```
gelman_stat <- list()
x <- gelman.diag(samples)
for(i in 1:8){
  sample_size[[i]] <- effectiveSize(samples)[(36*i-35):(36*i)]
  gelman_stat[[i]] <- x[[1]][(36*i-35):(36*i)]
}

#Plot effective sample size
names(sample_size) <- c('m1','m2','m3','m4','m5','m6','m7','r')
boxplot(sample_size,main='Yearly Effective Sample Size of Priors',ylab='Effective Sample Size',
        xlab='Variable')

#Fit true curve and data
mu <- summary(mu_samples)
mu_mean <- mu$statistics[,1]
lower_mu <- mu$quantiles[,1]
upper_mu <- mu$quantiles[,5]
plot(continuous_time,mu_mean,type='l',ylim=c(0,1),lwd=3,xlab='Year',ylab='EVI',
     main='Estimate of True EVI Curve with 95% CI')
polygon(c(continuous_time,rev(continuous_time)),c(lower_mu,rev(upper_mu)),col = 4,density=10)
points(continuous_time,df$EVI,col=2)
legend(x='topright',legend=c('EVI Curve Estimate','95% CI','EVI Measurements'),lty=c(1,1,NA),col=c(1,4,2),pch=c(NA,NA,1))

#Fit model 2
model_string_2 <- textConnection("model{
                for (i in 1:n){
                  Y[i] ~ dbeta(r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))
                  logit(mu[i]) <- b0[yrs[i]] + b1[yrs[i]]*t[i] + b2[yrs[i]]*t[i]^2
```

```
            }
            #for each i, we have to account for the year
            for (j in 1:k){
              b0[j] ~ dnorm(0,.01)
              b1[j] ~ dnorm(0,.01)
              b2[j] ~ dnorm(0,.01)
              r[j] ~ dgamma(.1,.1)
            }
            #WAIC Calculations
            for(i in 1:n){
              like[i] <- dbeta(Y[i],r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))
            }
}")
model2 <- jags.model(model_string_2,data=data,n.chains=2)

#Burn in 10000 samples
update(model2, 10000)

#Calculate DIC and WAIC
DIC2 <- dic.samples(model2,n.iter=20000,n.thin=10)
waic2   <- coda.samples(model2,variable.names=c("like"), n.iter=20000,n.thin=10)
like2   <- waic2[[1]]
fbar2   <- colMeans(like2)
P2      <- sum(apply(log(like2),2,var))
WAIC2   <- -2*sum(log(fbar2))+2*P2
#Fit model 3
model_string_3 <- textConnection("model{
            for (i in 1:n){
              Y[i] ~ dbeta(r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))
              logit(mu[i]) <- b0[yrs[i]]*sin(b1[yrs[i]]*(t[i]-b2[yrs[i]]))+b3[yrs[i]]
```

```
        }
        #for each i, we have to account for the year
        for (j in 1:k){
          b0[j] ~ dnorm(0,.01)
          b1[j] ~ dnorm(0,.01)
          b2[j] ~ dnorm(0,.01)
          b3[j] ~ dnorm(0,.01)
          r[j] ~ dgamma(.1,.1)
        }
        #WAIC Calculations
        for(i in 1:n){
          like[i] <- dbeta(Y[i],r[yrs[i]]*mu[i],r[yrs[i]]*(1-mu[i]))
        }
}")
model3 <- jags.model(model_string_3,data=data,n.chains=2)

#Burn in 10000 samples
update(model3, 10000)

#Calculate DIC and WAIC
DIC3 <- dic.samples(model3,n.iter=20000,n.thin=10)
waic3   <- coda.samples(model3,variable.names=c("like"), n.iter=20000,n.thin=10)
like3   <- waic3[[1]]
fbar3 <- colMeans(like3)
P3      <- sum(apply(log(like3),2,var))
WAIC3   <- -2*sum(log(fbar3))+2*P3

#Fit mu for each iteration and calculate GUT
mu_iterations <- mu_samples[[1]]
time <- data.frame(matrix(nrow=20000,ncol=k))
```

```
names(time) <- unique_years
for(i in 1:k){
  print(i)
  for(j in 1:20000){
    #first calculate the indices that correspond to each year
    year_index <- which(df$Year == unique_years[i])
    #find mu estimate and doy of estimate
    est <- mu_iterations[j,year_index]
    doy <- t[year_index]
    #use approx fn to find first time mu > .5
    approx <- approx(doy,est,n=1000)
    gut <- min(which(approx$y > .5))
    #save first time mu > .5
    time[j,i] <- approx$x[gut]
  }
}
boxplot(time,xlab='Year',ylab='GUT',main='Yearly GUT Posterior
Distributions',ylim=c(120,180))


#Perform regression to see if median GUT changes
medians = data.frame(x = unique_years,y = sapply(time, median, na.rm=TRUE))
regression <-lm(y ~ x, data=medians)
plot(unique_years,medians$y,xlab='Year',ylab='Median GUT',main='Regression of Yearly
Median GUT Values', ylim=c(100,200))
abline(regression)
coef <- round(coef(regression), 2)
text(2015,190,  paste("Y = ", coef[1], "-", abs(coef[2]), "* year"))
summary(regression)
```