Lab 5/6/7: Intro to Stan

Griffin Shelor

2024-02-22

```
## loading packages, reading in data
library(pacman)
p_load(tidyverse, here, rstan, shinystan, quarto, bayesplot)
portal <- read.csv(here("UNR-EcoForecast-main", "data", "portal_timeseries.csv"))
```

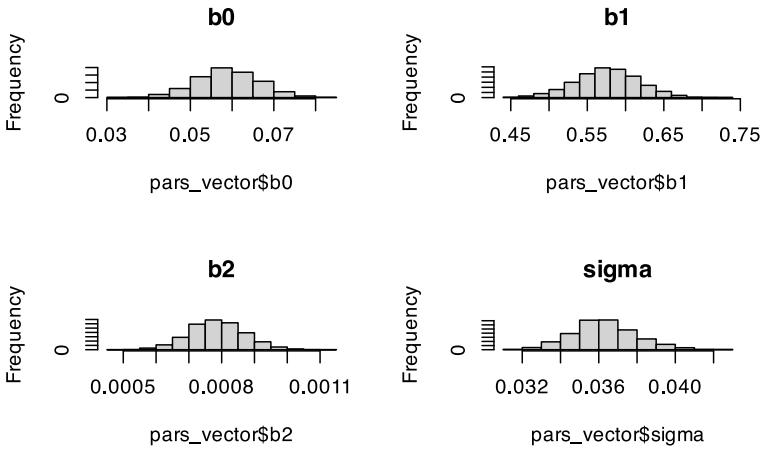
Lab 5

Question 1

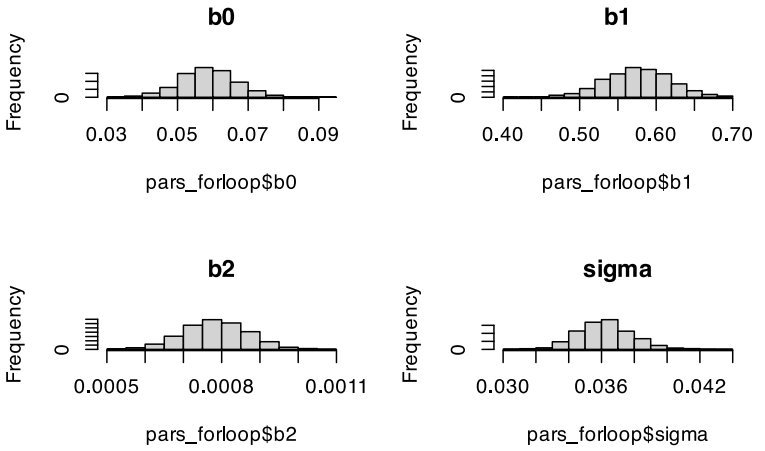
```
## preparing data for model fitting
rows <- nrow(portal)
portal_fit <- portal[1:(rows - 10),]
portal_fit_rows <- nrow(portal_fit)

## making list of data to declare what goes into stan model
ndvi_datalist <- list(N = nrow(portal_fit) -1, y = portal_fit$NDVI[-1], x1 = portal_fit$NDVI[-
portal_fit_rows], x2 = portal_fit$rain[-portal_fit_rows])

## fitting stan model
set.seed(802)
fit_vector <- stan(file=here("Labs", "Lab5", "Lab567NDVI.stan"),data=ndvi_datalist, chains=3,iter=2000,
warmup=1000)
fit_vector

## extracting parameters from fit_vector
pars_vector <- rstan::extract(fit_vector, c('b0','b1','b2','sigma'))


## fitting stan model but with a for loop as the model instead of a vector
set.seed(802)
fit_forloop <- stan(file=here("Labs", "Lab5", "Lab567NDVI_ForLoop.stan"),data=ndvi_datalist,
chains=3,iter=2000, warmup=1000)
fit_forloop

## extracting parameters from fit_forloop
pars_forloop <- rstan::extract(fit_forloop, c('b0','b1','b2','sigma'))
```

Question 2

```
## Plots of the posterior for the vector model
par(mfrow=c(2,2))
hist(pars_vector$b0, main = "b0")
hist(pars_vector$b1, main = "b1")
hist(pars_vector$b2, main = "b2")
hist(pars_vector$sigma, main = "sigma")
```




```
## plots of the posterior for the for loop model
par(mfrow=c(2,2))
hist(pars_forloop$b0, main = "b0")
hist(pars_forloop$b1, main = "b1")
hist(pars_forloop$b2, main = "b2")
hist(pars_forloop$sigma, main = "sigma")
```



Question 3

```
## looking at means for extracted parameters
mean(pars_vector$b0)
mean(pars_vector$b1)
mean(pars_vector$b2)
mean(pars_vector$sigma)

## means for extracted parameters from for loop model
mean(pars_forloop$b0)
mean(pars_forloop$b1)
mean(pars_forloop$b2)
mean(pars_forloop$sigma)
```

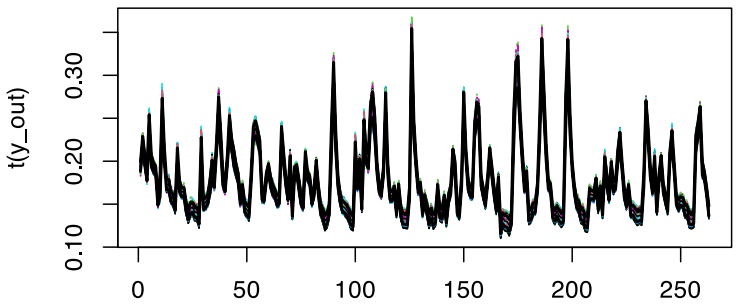
```
## visualizations to check convergence. Have chains converged?
# launch_shinystan(fit_vector)
```

My mean parameter estimates are very similar to the parameter estimates produced by the linear model with the intercept (b0) being approximately 0.059, the NDVI parameter (b1) being approximately 0.57, and the rain parameter (b2) being approximately 0.0007.

Lab 6

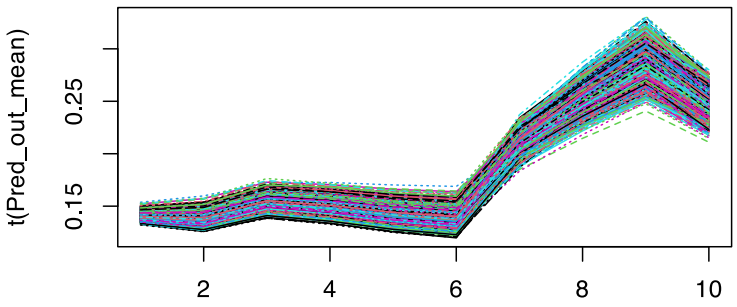
Question 1

```
b0 <- pars_vector$b0
b1 <- pars_vector$b1
b2 <- pars_vector$b2
y_out <- matrix(NA, nrow = nrow(b0), ncol = nrow(portal_fit))
for (p in 1:length(pars_vector$b0)){
  for (t in 1:nrow(portal_fit)){
    y = b0[p] + b1[p] * portal_fit$NDVI[t] + b2[p] * portal_fit$rain[t]
    y_out[p,t] <- y
  }
}

par(mfrow = c(1,1))
matplot(t(y_out), type = 'l')

mean_dist <- apply(y_out, 2,mean)
# mean_dist
lines(mean_dist)
# plot(mean_dist, col = 'blue')

UL_dist <- apply(y_out, 2,quantile, prob = 0.975)
lines(UL_dist, lwd = 2)

LL_dist <- apply(y_out, 2,quantile, prob = 0.025)
lines(LL_dist, lwd = 2)
```

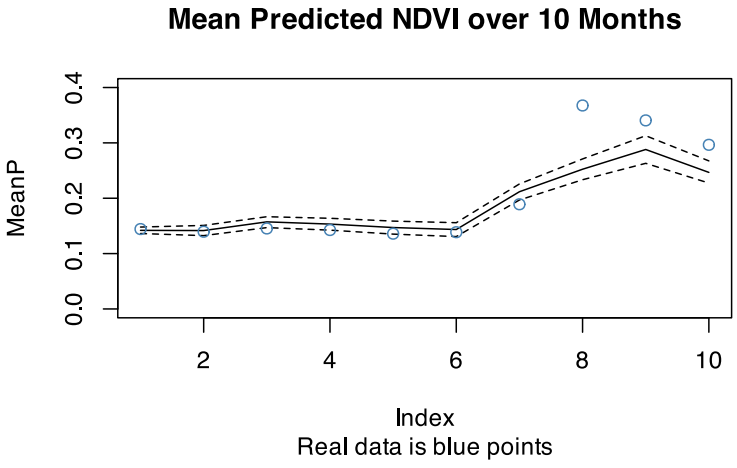



```
## extracting generated quantities from posterior
y_rep <- as.matrix(fit_vector, pars = "y_rep")
# dim(y_rep)

##
Pred_data<- portal[(rows-10):rows,]
Pred_out<- matrix(NA,length(pars_vector$b0),10)
```

```r
set.seed(802)
## Parameter error
Pred_out_mean <- matrix(NA, length(pars_vector$b0), 10)
for (p in 1:length(pars_vector$b0)){
  mean_NDVI <- Pred_data$NDVI[1]
  for (t in 1:10){
   mean_NDVI <- pars_vector$b0[p] + pars_vector$b1[p] * mean_NDVI + pars_vector$b2[p] * Pred_data$rain[t]
    Pred_out_mean[p,t] <- mean_NDVI
  }
}

matplot(t(Pred_out_mean),type='l')
```




```r
MeanP <- apply(Pred_out_mean,2,mean)
Upper <- apply(Pred_out_mean,2,quantile, prob=.975)
Lower <- apply(Pred_out_mean,2,quantile, prob=.025)


plot(MeanP,type='l', ylim = c(0,0.4), main = "Mean Predicted NDVI over 10 Months", sub = "Real data is
blue points")
lines(Upper,lty=2)
lines(Lower,lty=2)
points(Pred_data$NDVI,col='steelblue')
```



Question 2

As we forecast further in time, the SD of the predicted mean change increases over time, but remains fairly small, since it is based strictly on parameter error, and does not account for process variability which would add extra uncertainty due to the possibility of sudden jumps in NDVI not necessarily accounted for by our variables or parameters. Even with just parameter uncertainty, uncertainty increases as we move forward in time.

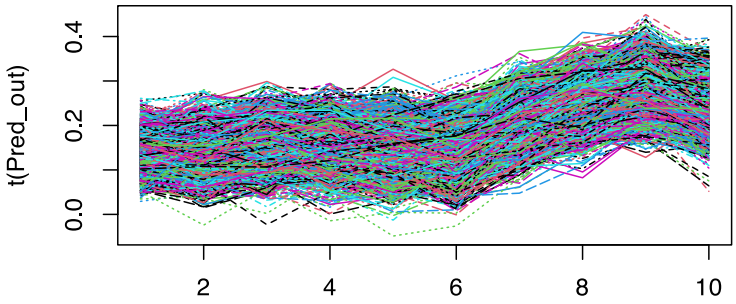
Question 3

It is important to account for covariance in parameter error because if we know how sensitive one variable is to another, we can better understand how change in one variable can lead to change in another. We can then use this to understand how error in one parameter can lead to uncertainty in our response variable or potentially even another predictor parameter, depending on how complex our model is. Because of this, I think not including covariance would increase predicted uncertainty in this NDVI example.

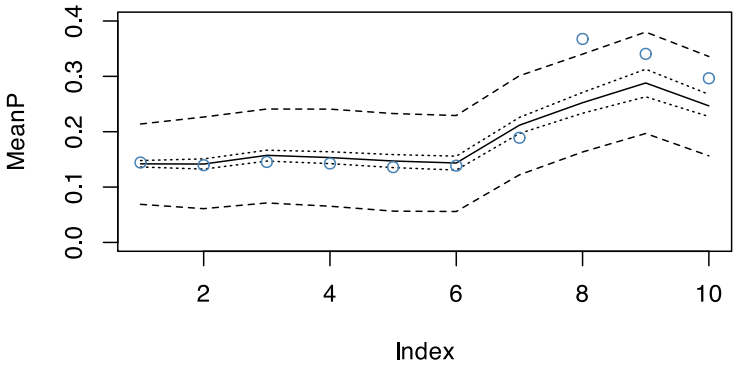
Lab 7

Question 1

```
## adding in process uncertainty
Pred_out <- matrix(NA, length(pars_vector$b0), 10)
Pred_out_mean <- matrix(NA, length(pars_vector$b0), 10)

## Parameter and Process Error
# for (p in 1:length(pars_vector$b0)){
#   NDVI <- Pred_data$NDVI[1]
#   for (t in 1:10){
#     mean_NDVI <- pars_vector$b0[p] + pars_vector$b1[p] * NDVI + pars_vector$b2[p] * Pred_data$rain[t]
#     NDVI <- rnorm(1, mean_NDVI, pars_vector$sigma[p])
#     Pred_out[p,t] <- NDVI
#   }
# }

## process error
for (p in 1:length(pars_vector$b0)){
  NDVI <- Pred_data$NDVI[1]
  for(t in 1:10){
       NDVI <- rnorm(1, mean = pars_vector$b0[p] + pars_vector$b1[p] * NDVI + pars_vector$b2[p] *
Pred_data$rain[t], sd=pars_vector$sigma[p])
  Pred_out[p,t] <- NDVI
  }
}
## Parameter error
for (p in 1:length(pars_vector$b0)){
  mean_NDVI <- Pred_data$NDVI[1]
  for (t in 1:10){
   mean_NDVI <- pars_vector$b0[p] + pars_vector$b1[p] * mean_NDVI + pars_vector$b2[p] * Pred_data$rain[t]
    Pred_out_mean[p,t] <- mean_NDVI
  }
}
par(mfrow = c(1,1))
matplot(t(Pred_out),type='l')
```




```
MeanP<-apply(Pred_out_mean,2,mean)
Upper<-apply(Pred_out,2,quantile, prob=.975)
Upper_mean <-apply(Pred_out_mean,2,quantile, prob=.975)
Lower<-apply(Pred_out,2,quantile, prob=.025)
Lower_mean<-apply(Pred_out_mean,2,quantile, prob=.025)


plot(MeanP,type='l', ylim=c(0,.4))
lines(Upper,lty=2)
lines(Upper_mean,lty=3)
lines(Lower,lty=2)
lines(Lower_mean,lty=3)
points(Pred_data$NDVI,col='steelblue')
```

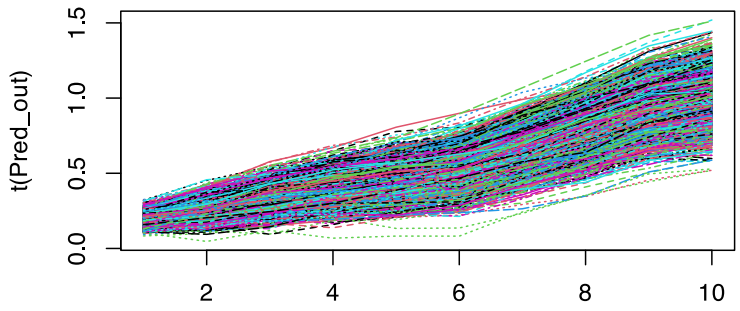


There is 1 point which falls outside of the full predictive interval. Given that there are 10 points, we would expect either 0 or 1 point to be outside of the 95% credible interval.

Question 2

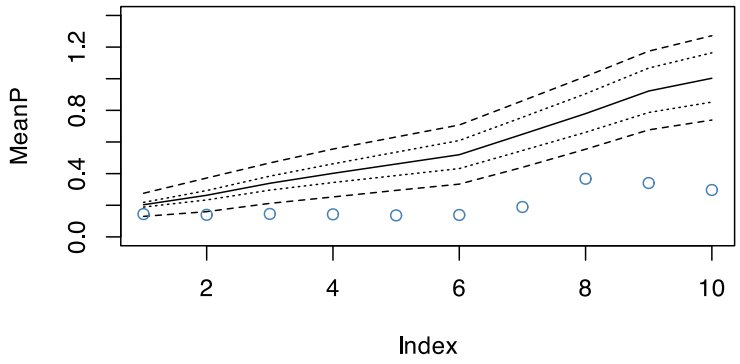
```
## adding in process uncertainty
Pred_out <- matrix(NA, length(pars_vector$b0), 10)
Pred_out_mean <- matrix(NA, length(pars_vector$b0), 10)

## Parameter and Process Error
# for (p in 1:length(pars_vector$b0)){
```

```
#   NDVI <- Pred_data$NDVI[1]
#   for (t in 1:10){
#     mean_NDVI <- pars_vector$b0[p] + pars_vector$b1[p] * NDVI + pars_vector$b2[p] * Pred_data$rain[t]
#     NDVI <- rnorm(1, mean_NDVI, pars_vector$sigma[p])
#     Pred_out[p,t] <- NDVI
#   }
# }

## process error
set.seed(802)
for (p in 1:length(pars_vector$b0)){
  NDVI <- Pred_data$NDVI[1]
  for(t in 1:10){
      NDVI <- rnorm(1, mean = pars_vector$b0[p] + 1 * NDVI + pars_vector$b2[p] * Pred_data$rain[t],
sd=pars_vector$sigma[p])
  Pred_out[p,t] <- NDVI
  }
}
## Parameter error
for (p in 1:length(pars_vector$b0)){
  mean_NDVI <- Pred_data$NDVI[1]
  for (t in 1:10){
    mean_NDVI <- pars_vector$b0[p] + 1 * mean_NDVI + pars_vector$b2[p] * Pred_data$rain[t]
    Pred_out_mean[p,t] <- mean_NDVI
  }
}
par(mfrow = c(1,1))
matplot(t(Pred_out),type='l')
```




```
MeanP <- apply(Pred_out_mean,2,mean)
Upper <- apply(Pred_out,2,quantile, prob=.975)
Upper_mean <- apply(Pred_out_mean,2,quantile, prob=.975)
Lower <- apply(Pred_out,2,quantile, prob=.025)
Lower_mean <- apply(Pred_out_mean,2,quantile, prob=.025)


plot(MeanP,type='l', ylim=c(0,1.4))
lines(Upper,lty=2)
lines(Upper_mean,lty=3)
lines(Lower,lty=2)
```

```
lines(Lower_mean,lty=3)
points(Pred_data$NDVI,col='steelblue')
```



If we do not assume that there is a long-term equilibrium in the model, our model very quickly begins to no longer reflect the actual data. Instead, it continues to project an increasing rise in NDVI with nothing to limit how high our projected NDVI values can go or stabilize it to a standard long-term value or range of values. This is different from our model in Question 1 because our model in question 1 included a parameter value which was less than 1 and thus would cause our projected NDVI values to decrease if they grow too high. This use of the parameter prevents our model from projecting values which are mathematically impossible for NDVI to be, such as any value outside the range of −1 to 1.

Question 3

As we collect more data, we would expect our parameter error to decrease relative to process variability. This is because more data would help our model develop a narrower range for the credible interval that our parameter value is likely to exist within. If the credible interval for the parameter value is narrower, process variability would make up a larger proportion of the total uncertainty between both of these sources of uncertainty.