

ModelTests

Sophie Wulfinf

2022-05-05

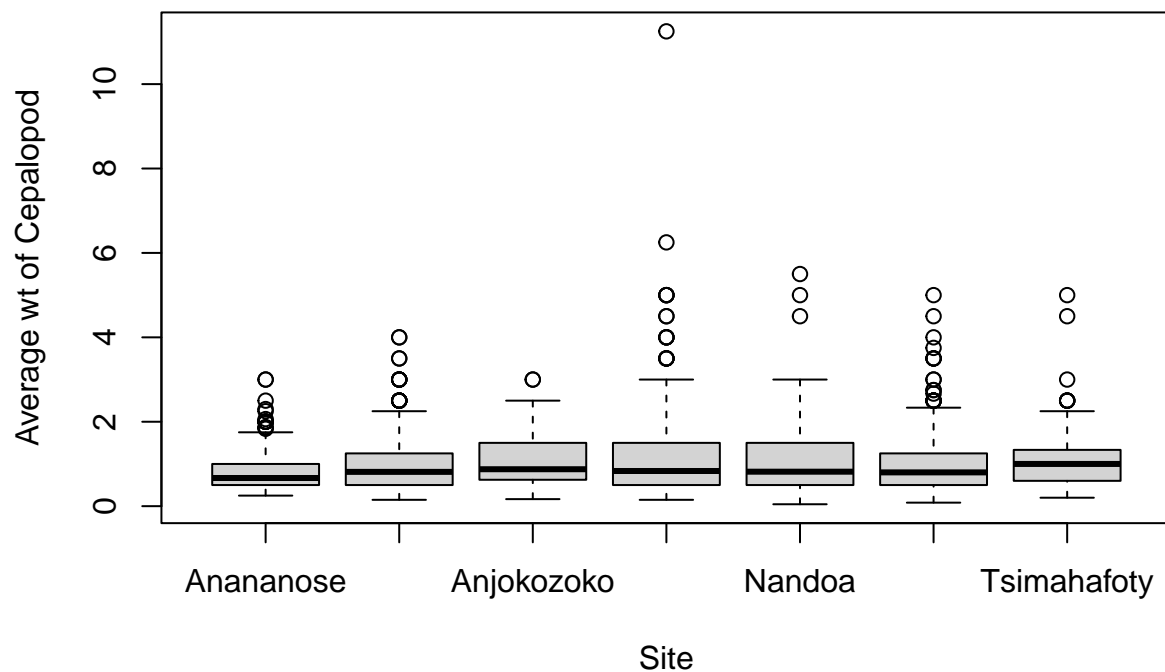
Ok so here's my question, my model is predicting negative values for average weight per site. I have in my notes that i need a link function but I guess I never actually put it into my code when I wrote that. Is it better to throw that link function directly into the jags code? Or should I have the jags run normally and then transform it later when graphing it? I've taken out the jags output because it was just pages and pages of the y.new but if you want to see it I can send that as well! Thank you!

```
#Questions about this one: can you do a p val for an anova? Does the calc look the same?  
#wANT TO REMAKE data next  
#Add random effects? can you do that to an anova
```

```
#ANOVA  
#Only looking at most fished sites  
#Different variances?
```

```
#Visualize Data-FIX THIS  
boxplot(as.numeric(AVG_WT) ~ REGION,  
        ylab = "Average wt of Cephalopod",  
        xlab = "Site",  
        # outline = TRUE,  
        # notch = FALSE,  
        # staplewex = 0.2,  
        # boxwex = 0.4,  
        # border = TRUE,  
        data = ceph_sites2)
```

```
## Warning in eval(predvars, data, env): NAs introduced by coercion
```



```
#sink("cephs_sitewt.txt")
cat("
  model {

    # PRIORS
    for(i in 1:nsites){
      alpha[i] ~ dnorm(mu_site,tau_site)
    }

    mu_site ~ dnorm(0,0.001)
    tau_site <- 1/sigma_site^2
    sigma_site ~ dunif(0,100)

    for(i in 1:nsites){
      tau[i] <- 1/sigma[i]^2
      sigma[i] ~ dunif(0,100)
    }

    # LIKELIHOOD
    for(i in 1:n){
      mass[i] ~ dnorm(mu[i],tau[site[i]])
      mu[i] <- alpha[site[i]] #Mean will vary by site. 3 alphas, 1 per site. loop through every row of y
    } # i

    # DERIVED QUANTITIES
    for(i in 1:nsites){
```

```

avg[i] <- (alpha[i]*stdev_data) + mean_data
stan_dev[i] <- (sigma[i]*stdev_data)
}

mu_site_m <- (mu_site*stdev_data) + mean_data
sigma_site_m <- (sigma_site*stdev_data)

# generate new data from the fitted model
for(i in 1:n){
y.new[i] ~ dnorm(mu[i],tau[i])    #Generates new data for each run of mcmc
}

} # end of model
",fill = TRUE)

```

```

##
##   model {
##
##   # PRIORS
##   for(i in 1:nsites){
##     alpha[i] ~ dnorm(mu_site,tau_site)
##   }
##
##   mu_site ~ dnorm(0,0.001)
##   tau_site <- 1/sigma_site^2
##   sigma_site ~ dunif(0,100)
##
##   for(i in 1:nsites){
##     tau[i] <- 1/sigma[i]^2
##     sigma[i] ~ dunif(0,100)
##   }
##
##   # LIKELIHOOD
##   for(i in 1:n){
##     mass[i] ~ dnorm(mu[i],tau[site[i]])
##     mu[i] <- alpha[site[i]] #Mean will vary by site. 3 alphas, 1 per site. loop through every row of
##     } # i
##
##   # DERIVED QUANTITIES
##   for(i in 1:nsites){
##     avg[i] <- (alpha[i]*stdev_data) + mean_data
##     stan_dev[i] <- (sigma[i]*stdev_data)
##   }
##
##   mu_site_m <- (mu_site*stdev_data) + mean_data
##   sigma_site_m <- (sigma_site*stdev_data)
##
##   # generate new data from the fitted model
##   for(i in 1:n){
##     y.new[i] ~ dnorm(mu[i],tau[i])    #Generates new data for each run of mcmc
##   }
##
##   } # end of model

```

```
##

#sink()

win.data <- list(mass = as.numeric(ceph_sites2$AVG_WT),
               site = ceph_sites2$id,
               n = nrow(ceph_sites2),
               nsites = length(unique(ceph_sites2$id)),
               stdev_data = sd(as.numeric(ceph_sites2$AVG_WT), na.rm = TRUE),
               mean_data = mean(as.numeric(ceph_sites2$AVG_WT), na.rm = TRUE))

## Warning: NAs introduced by coercion

## Warning in is.data.frame(x): NAs introduced by coercion

## Warning in mean(as.numeric(ceph_sites2$AVG_WT), na.rm = TRUE): NAs introduced by
## coercion

# Initial values
inits <- function()list(alpha = rnorm(7), sigma = rep(rlnorm(1),7))

# Parameters monitored
params <- c("alpha",
           "sigma",
           "avg",
           "stan_dev",
           "mu_site_m",
           "sigma_site_m",
           #"Bayes.P"
           "y.new"
)

# MCMC settings
ni <- 1000; nt <- 3; nb <- 100; nc <- 3

out_stwt <- jags(win.data, inits, params, "cephs_sitewt.txt",
               n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb,
               working.directory = getwd())

## module glm loaded

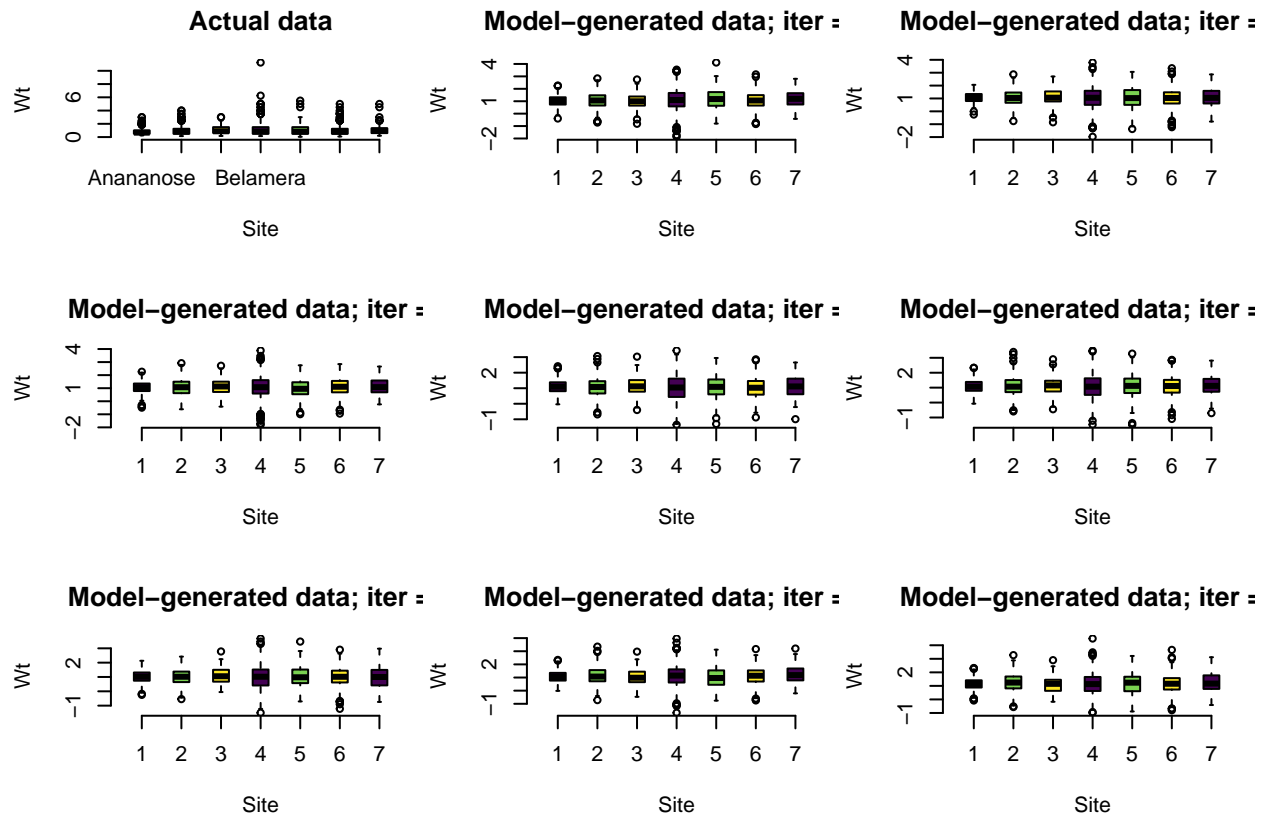
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 3219
##   Unobserved stochastic nodes: 3253
##   Total graph size: 9749
##
## Initializing model
```

```
#print(out_stwt, dig = 1)
```

```
par(mfrow = c(3,3),  
    mar = c(5.1, 4.1, 3.1, 0.5)) #b, l, t, r  
boxplot(as.numeric(AVG_WT) ~ REGION,  
        ylab = "Wt",  
        xlab = "Site",  
        staplewex = 0.2,  
        boxwex = 0.4,  
        frame.plot = FALSE,  
        col = c(purps,limabean,bananas),  
        data = ceph_sites2,  
        main = "Actual data")
```

```
## Warning in eval(predvars, data, env): NAs introduced by coercion
```

```
#MODEL PREDICTING NEGATIVE VALUES FOR WT. NEED LINK FXN?  
for(i in 1:8){  
  boxplot(out_stwt$BUGSoutput$sims.list$y.new[i,] ~ ceph_sites2$id,  
          ylab = "Wt",  
          xlab = "Site",  
          staplewex = 0.2,  
          boxwex = 0.4,  
          frame.plot = FALSE,  
          col = c(purps,limabean,bananas),  
          main = paste0("Model-generated data; iter = ",i))  
}
```



```
dev.off()
```

```
## null device
##      1
```