HW6

Cheng

5/21/2019

## Question1

data("salamanders")  
d <- salamanders; remove(salamanders)

Try the model:

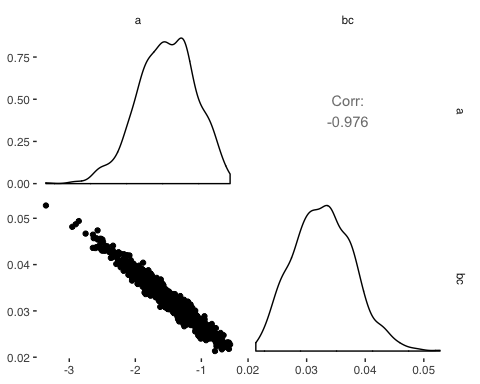
# Rstan setup  
m1='  
 data {  
 int N;  
 int T[N];  
 int C[N];  
 }  
 parameters {  
 real a;  
 real bc;  
 }  
 model {  
 vector[N] lambda;  
 a ~ normal(0,100);  
 bc ~ normal(0,1);  
 for(i in 1:N) lambda[i] = a + bc \* C[i];  
 T ~ poisson\_log(lambda);  
 }  
 generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] lambda;  
 for(i in 1:N) {  
 lambda[i] = a + bc \* C[i];  
 log\_lik[i] = poisson\_log\_lpmf(T[i] | lambda[i]);  
 }  
 }  
 }  
 '

Output1

dat = list(N = NROW(d),   
 T = d$SALAMAN,  
 C = d$PCTCOVER)  
  
fit1 <- stan(model\_code = m1,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)  
  
print(fit1)

## Inference for Stan model: dd4506394d363b76ed8f714ba4468798.  
## 2 chains, each with iter=1000; warmup=500; thin=1;   
## post-warmup draws per chain=500, total post-warmup draws=1000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## a -1.53 0.03 0.43 -2.46 -1.82 -1.52 -1.23 -0.77 178 1.01  
## bc 0.03 0.00 0.01 0.02 0.03 0.03 0.04 0.04 184 1.01  
## log\_lik[1] -9.66 0.03 0.90 -11.47 -10.30 -9.62 -9.06 -7.97 856 1.00  
## log\_lik[2] -6.90 0.02 0.70 -8.35 -7.40 -6.86 -6.42 -5.59 865 1.00  
## log\_lik[3] -5.97 0.02 0.69 -7.34 -6.46 -5.95 -5.49 -4.69 803 1.00  
## log\_lik[4] -4.46 0.02 0.50 -5.48 -4.81 -4.43 -4.10 -3.54 848 1.00  
## log\_lik[5] -3.49 0.01 0.39 -4.30 -3.77 -3.46 -3.21 -2.78 824 1.00  
## log\_lik[6] -3.46 0.01 0.35 -4.19 -3.70 -3.43 -3.22 -2.82 798 1.00  
## log\_lik[7] -2.71 0.01 0.25 -3.26 -2.89 -2.69 -2.54 -2.26 794 1.00  
## log\_lik[8] -2.14 0.01 0.18 -2.53 -2.25 -2.12 -2.01 -1.88 746 1.00  
## log\_lik[9] -1.90 0.00 0.11 -2.16 -1.97 -1.88 -1.81 -1.75 798 1.00  
## log\_lik[10] -1.84 0.00 0.09 -2.05 -1.89 -1.82 -1.77 -1.74 731 1.00  
## log\_lik[11] -1.66 0.00 0.03 -1.75 -1.67 -1.64 -1.64 -1.63 602 1.00  
## log\_lik[12] -1.53 0.00 0.04 -1.63 -1.54 -1.51 -1.50 -1.50 763 1.00  
## log\_lik[13] -1.60 0.00 0.08 -1.81 -1.64 -1.59 -1.54 -1.50 874 1.00  
## log\_lik[14] -1.66 0.00 0.11 -1.93 -1.73 -1.65 -1.58 -1.51 852 1.00  
## log\_lik[15] -1.79 0.01 0.16 -2.16 -1.89 -1.77 -1.67 -1.56 762 1.00  
## log\_lik[16] -1.84 0.01 0.18 -2.26 -1.95 -1.82 -1.71 -1.58 727 1.00  
## log\_lik[17] -3.87 0.06 0.75 -5.54 -4.37 -3.83 -3.34 -2.62 177 1.01  
## log\_lik[18] -1.83 0.01 0.18 -2.23 -1.94 -1.82 -1.70 -1.54 884 1.00  
## log\_lik[19] -2.27 0.01 0.29 -2.89 -2.46 -2.25 -2.06 -1.80 722 1.00  
## log\_lik[20] -1.59 0.02 0.29 -2.27 -1.78 -1.56 -1.38 -1.15 176 1.01  
## log\_lik[21] -1.39 0.02 0.22 -1.94 -1.53 -1.36 -1.23 -1.06 176 1.01  
## log\_lik[22] -1.33 0.02 0.20 -1.83 -1.46 -1.30 -1.18 -1.04 176 1.01  
## log\_lik[23] -1.16 0.01 0.13 -1.49 -1.24 -1.13 -1.06 -1.00 177 1.01  
## log\_lik[24] -1.10 0.01 0.10 -1.34 -1.14 -1.07 -1.02 -1.00 180 1.02  
## log\_lik[25] -1.02 0.00 0.03 -1.10 -1.03 -1.01 -1.00 -1.00 287 1.01  
## log\_lik[26] -1.92 0.01 0.19 -2.33 -2.05 -1.91 -1.79 -1.58 690 1.00  
## log\_lik[27] -2.38 0.01 0.26 -2.92 -2.54 -2.37 -2.19 -1.92 881 1.00  
## log\_lik[28] -2.57 0.01 0.29 -3.18 -2.75 -2.55 -2.35 -2.06 870 1.00  
## log\_lik[29] -2.99 0.01 0.37 -3.76 -3.23 -2.97 -2.72 -2.35 754 1.00  
## log\_lik[30] -0.24 0.01 0.10 -0.46 -0.29 -0.22 -0.16 -0.09 190 1.00  
## log\_lik[31] -0.24 0.01 0.10 -0.48 -0.30 -0.23 -0.17 -0.09 190 1.00  
## log\_lik[32] -0.26 0.01 0.11 -0.50 -0.32 -0.24 -0.18 -0.10 190 1.00  
## log\_lik[33] -0.28 0.01 0.11 -0.52 -0.34 -0.26 -0.19 -0.11 189 1.00  
## log\_lik[34] -0.30 0.01 0.12 -0.57 -0.37 -0.28 -0.21 -0.12 189 1.00  
## log\_lik[35] -0.31 0.01 0.12 -0.58 -0.38 -0.29 -0.22 -0.13 189 1.00  
## log\_lik[36] -0.33 0.01 0.12 -0.61 -0.40 -0.31 -0.24 -0.14 189 1.00  
## log\_lik[37] -0.37 0.01 0.13 -0.66 -0.44 -0.35 -0.27 -0.16 189 1.00  
## log\_lik[38] -0.40 0.01 0.14 -0.71 -0.48 -0.38 -0.30 -0.18 189 1.00  
## log\_lik[39] -0.50 0.01 0.15 -0.84 -0.59 -0.48 -0.38 -0.24 189 1.01  
## log\_lik[40] -0.94 0.01 0.20 -1.36 -1.08 -0.93 -0.79 -0.59 194 1.01  
## log\_lik[41] -1.22 0.02 0.22 -1.67 -1.36 -1.21 -1.05 -0.84 202 1.01  
## log\_lik[42] -2.74 0.01 0.27 -3.28 -2.93 -2.72 -2.56 -2.25 513 1.00  
## log\_lik[43] -2.83 0.01 0.28 -3.40 -3.02 -2.82 -2.64 -2.33 571 1.00  
## log\_lik[44] -3.02 0.01 0.29 -3.62 -3.22 -3.01 -2.82 -2.49 688 1.00  
## log\_lik[45] -3.68 0.01 0.35 -4.41 -3.91 -3.67 -3.42 -3.03 879 1.00  
## log\_lik[46] -4.06 0.01 0.40 -4.92 -4.33 -4.05 -3.76 -3.34 847 1.00  
## log\_lik[47] -4.34 0.02 0.45 -5.26 -4.63 -4.33 -4.02 -3.54 787 1.00  
## lp\_\_ 22.31 0.05 0.87 19.96 21.92 22.55 22.95 23.23 343 1.01  
##   
## Samples were drawn using NUTS(diag\_e) at Tue May 21 23:21:57 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

library(GGally)   
post1 <- as.data.frame(fit1)  
post1[,1:2] %>%  
 ggpairs() + theme\_tufte(base\_family = 'sans')



As you can see, there is the strong correlation between intercept and the coefficient of PCTCOVER. So I would like to try the method of de-mean, centering predictors to see whether it would reduce the correlation among parameters. Let’s try the model 2 with de-mean method:

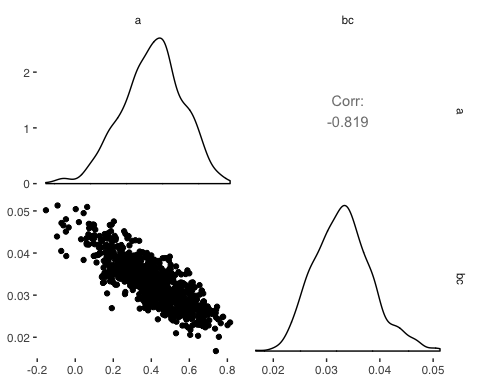
# add de-mean predictors  
d$PCTCOVER\_de <- as.integer(d$PCTCOVER - mean(d$PCTCOVER))  
str(d)

## 'data.frame': 47 obs. of 5 variables:  
## $ SITE : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ SALAMAN : int 13 11 11 9 8 7 6 6 5 5 ...  
## $ PCTCOVER : int 85 86 90 88 89 83 83 91 88 90 ...  
## $ FORESTAGE : int 316 88 548 64 43 368 200 71 42 551 ...  
## $ PCTCOVER\_de: int 26 27 31 29 30 24 24 32 29 31 ...

dat = list(N = NROW(d),   
 T = d$SALAMAN,  
 C = d$PCTCOVER\_de)  
  
fit2 <- stan(model\_code = m1,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)  
  
post2 <- as.data.frame(fit2)

Output2:

post1 <- as.data.frame(fit2)  
post1[,1:2] %>%  
 ggpairs() + theme\_tufte(base\_family = 'sans')

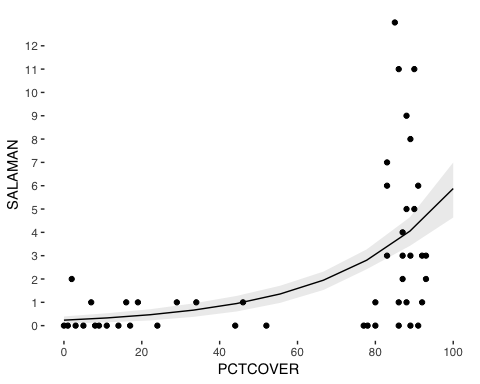


The correlation has slightly decreased, but it’s still strong. But I would use model2 to evalute the model of relationship between density and percent cover.

nd <-   
 expand.grid(pctcover = seq(0, 100, length.out = 10)) %>%  
 mutate(pctcover\_de = pctcover - mean(d$PCTCOVER))  
# get posterior parameters of each  
f\_mu\_2 <- function(C) with(post2, a + bc \* C)  
mu <- mapply(f\_mu\_2, C = nd$pctcover\_de)  
mu <- exp(mu)  
mu\_mean <- colMeans(mu)   
mu\_hpdi <- apply(mu, 2, rethinking::HPDI)  
nd <- nd %>%  
 mutate(mu = mu\_mean,  
 mu\_hpdi\_l = mu\_hpdi[1,],  
 mu\_hpdi\_h = mu\_hpdi[2,])

plot:

ggplot() +   
geom\_point(data = d, aes(x = PCTCOVER, y = SALAMAN)) +  
geom\_ribbon(data = nd, aes(x = pctcover, ymin = mu\_hpdi\_l, ymax = mu\_hpdi\_h), alpha = 0.1) +  
geom\_line(data = nd, aes(x = pctcover, y = mu)) +  
scale\_color\_manual(values = "black", aesthetics = "colour") +  
scale\_x\_continuous(breaks = seq(0, 100, by = 20)) +   
scale\_y\_continuous(breaks = seq(0, 12, by = 1))

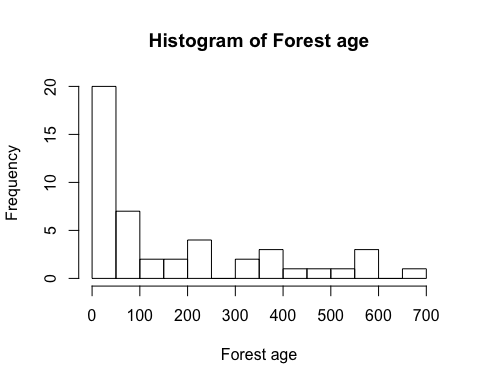


When the percent of cover ground is below 20, there are only few salamanders. When the covered ground is more than 80%, the salamander count rises obviously as you can see from the plot. This show that our model do capture the pattern of observations. However, when the covered ground is more than 80%, our model underestimate the result comparing to the true value. This indicates the model can be improved.

## Question2

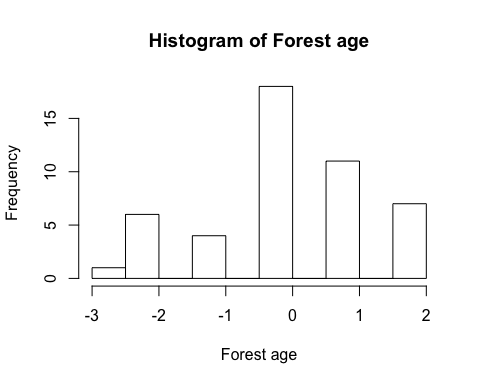
To improve the model by adding variable FORESTAGE, Let’s take a look of it first:

# take a look  
hist(d$FORESTAGE, breaks = 10,  
 main="Histogram of Forest age", xlab="Forest age")



The distribution of FORESTAGE is quite skewed! So I would like to do the log transformation.

# take log  
d$logFORESTAGE <- as.integer(log(d$FORESTAGE + 1))  
d$logFORESTAGE\_de <- as.integer(d$logFORESTAGE - mean(d$logFORESTAGE)) # centered  
hist(d$logFORESTAGE\_de, breaks = 10,  
 main="Histogram of Forest age", xlab="Forest age")



print(str(d))

## 'data.frame': 47 obs. of 7 variables:  
## $ SITE : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ SALAMAN : int 13 11 11 9 8 7 6 6 5 5 ...  
## $ PCTCOVER : int 85 86 90 88 89 83 83 91 88 90 ...  
## $ FORESTAGE : int 316 88 548 64 43 368 200 71 42 551 ...  
## $ PCTCOVER\_de : int 26 27 31 29 30 24 24 32 29 31 ...  
## $ logFORESTAGE : int 5 4 6 4 3 5 5 4 3 6 ...  
## $ logFORESTAGE\_de: int 1 0 2 0 0 1 1 0 0 2 ...  
## NULL

# Rstan Setup  
m3='  
 data {  
 int N;  
 int T[N];  
 int C[N];  
 int F[N];  
 }  
 parameters {  
 real a;  
 real bc;  
 real bf;  
 }  
 model {  
 vector[N] lambda;  
 a ~ normal(0,100);  
 bc ~ normal(0,1);  
 bf ~ normal(0,1);  
 for(i in 1:N) lambda[i] = a + bc \* C[i] + bf \* F[i];  
 T ~ poisson\_log(lambda);  
 }  
 generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] lambda;  
 for(i in 1:N) {  
 lambda[i] = a + bc \* C[i] + bf \* F[i];  
 log\_lik[i] = poisson\_log\_lpmf(T[i] | lambda[i]);  
 }  
 }  
 }  
 '

Run

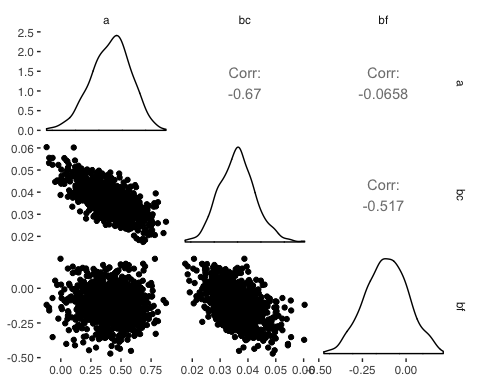
dat = list(N = NROW(d),   
 T = d$SALAMAN,  
 C = d$PCTCOVER\_de,  
 F = d$logFORESTAGE\_de)  
  
fit3 <- stan(model\_code = m3,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)

Parameters

print(fit3)

## Inference for Stan model: 94b08db6ea073536797e2e2578501cff.  
## 2 chains, each with iter=1000; warmup=500; thin=1;   
## post-warmup draws per chain=500, total post-warmup draws=1000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## a 0.41 0.01 0.17 0.06 0.30 0.42 0.53 0.72 550 1  
## bc 0.04 0.00 0.01 0.02 0.03 0.04 0.04 0.05 441 1  
## bf -0.11 0.01 0.12 -0.35 -0.20 -0.11 -0.03 0.13 548 1  
## log\_lik[1] -9.87 0.03 0.92 -11.71 -10.47 -9.85 -9.28 -8.04 1034 1  
## log\_lik[2] -6.26 0.04 0.96 -8.08 -6.95 -6.16 -5.60 -4.55 705 1  
## log\_lik[3] -6.83 0.04 1.20 -9.34 -7.53 -6.74 -6.02 -4.76 767 1  
## log\_lik[4] -4.00 0.03 0.68 -5.38 -4.46 -3.91 -3.53 -2.84 679 1  
## log\_lik[5] -3.14 0.02 0.52 -4.24 -3.49 -3.05 -2.76 -2.29 665 1  
## log\_lik[6] -3.57 0.01 0.37 -4.34 -3.80 -3.54 -3.31 -2.87 951 1  
## log\_lik[7] -2.79 0.01 0.27 -3.37 -2.96 -2.77 -2.60 -2.30 951 1  
## log\_lik[8] -2.02 0.01 0.19 -2.50 -2.12 -1.96 -1.87 -1.83 634 1  
## log\_lik[9] -1.84 0.00 0.11 -2.13 -1.89 -1.79 -1.75 -1.74 661 1  
## log\_lik[10] -1.99 0.01 0.22 -2.54 -2.09 -1.93 -1.83 -1.74 789 1  
## log\_lik[11] -1.74 0.00 0.13 -2.09 -1.78 -1.69 -1.65 -1.63 753 1  
## log\_lik[12] -1.52 0.00 0.03 -1.62 -1.53 -1.51 -1.50 -1.50 524 1  
## log\_lik[13] -1.59 0.00 0.08 -1.79 -1.62 -1.57 -1.53 -1.50 926 1  
## log\_lik[14] -1.65 0.00 0.11 -1.93 -1.71 -1.63 -1.58 -1.51 1028 1  
## log\_lik[15] -1.79 0.00 0.16 -2.18 -1.88 -1.76 -1.68 -1.56 1068 1  
## log\_lik[16] -1.72 0.01 0.20 -2.21 -1.81 -1.68 -1.57 -1.50 718 1  
## log\_lik[17] -3.74 0.03 0.80 -5.52 -4.22 -3.66 -3.19 -2.38 539 1  
## log\_lik[18] -2.05 0.01 0.32 -2.75 -2.23 -2.03 -1.81 -1.57 707 1  
## log\_lik[19] -2.05 0.01 0.35 -2.84 -2.23 -2.01 -1.79 -1.50 764 1  
## log\_lik[20] -1.53 0.01 0.31 -2.27 -1.70 -1.48 -1.31 -1.07 543 1  
## log\_lik[21] -1.39 0.01 0.24 -1.96 -1.52 -1.36 -1.22 -1.04 502 1  
## log\_lik[22] -1.39 0.01 0.24 -1.96 -1.52 -1.36 -1.21 -1.05 458 1  
## log\_lik[23] -1.14 0.01 0.13 -1.48 -1.21 -1.11 -1.04 -1.00 510 1  
## log\_lik[24] -1.12 0.01 0.11 -1.42 -1.17 -1.08 -1.03 -1.00 467 1  
## log\_lik[25] -1.02 0.00 0.03 -1.13 -1.03 -1.01 -1.00 -1.00 397 1  
## log\_lik[26] -1.85 0.01 0.20 -2.29 -1.97 -1.83 -1.71 -1.50 779 1  
## log\_lik[27] -2.09 0.02 0.39 -2.94 -2.32 -2.06 -1.83 -1.46 640 1  
## log\_lik[28] -2.90 0.02 0.49 -3.96 -3.19 -2.89 -2.54 -2.10 693 1  
## log\_lik[29] -2.99 0.01 0.37 -3.80 -3.22 -2.95 -2.75 -2.35 1131 1  
## log\_lik[30] -0.29 0.01 0.15 -0.66 -0.37 -0.27 -0.19 -0.09 583 1  
## log\_lik[31] -0.27 0.01 0.12 -0.56 -0.33 -0.25 -0.18 -0.09 538 1  
## log\_lik[32] -0.29 0.01 0.13 -0.59 -0.35 -0.27 -0.20 -0.10 542 1  
## log\_lik[33] -0.31 0.01 0.13 -0.63 -0.38 -0.29 -0.21 -0.11 545 1  
## log\_lik[34] -0.30 0.01 0.13 -0.60 -0.37 -0.28 -0.21 -0.11 472 1  
## log\_lik[35] -0.31 0.01 0.13 -0.61 -0.38 -0.29 -0.22 -0.12 474 1  
## log\_lik[36] -0.38 0.01 0.15 -0.75 -0.46 -0.36 -0.27 -0.14 557 1  
## log\_lik[37] -0.34 0.01 0.14 -0.65 -0.41 -0.31 -0.24 -0.13 418 1  
## log\_lik[38] -0.37 0.01 0.14 -0.69 -0.45 -0.35 -0.27 -0.15 421 1  
## log\_lik[39] -0.47 0.01 0.16 -0.82 -0.56 -0.45 -0.35 -0.21 429 1  
## log\_lik[40] -0.94 0.01 0.22 -1.38 -1.08 -0.93 -0.79 -0.55 468 1  
## log\_lik[41] -1.24 0.01 0.24 -1.74 -1.39 -1.24 -1.07 -0.80 503 1  
## log\_lik[42] -2.92 0.01 0.36 -3.67 -3.16 -2.91 -2.65 -2.29 731 1  
## log\_lik[43] -2.71 0.01 0.30 -3.34 -2.90 -2.69 -2.51 -2.16 701 1  
## log\_lik[44] -2.63 0.02 0.50 -3.65 -2.94 -2.60 -2.30 -1.76 565 1  
## log\_lik[45] -4.06 0.02 0.56 -5.22 -4.41 -4.05 -3.63 -3.13 713 1  
## log\_lik[46] -4.53 0.03 0.67 -5.93 -4.95 -4.52 -4.03 -3.38 679 1  
## log\_lik[47] -4.33 0.01 0.44 -5.27 -4.61 -4.29 -4.04 -3.55 1145 1  
## lp\_\_ 22.13 0.07 1.22 19.01 21.58 22.41 23.02 23.54 330 1  
##   
## Samples were drawn using NUTS(diag\_e) at Tue May 21 23:22:42 2019.  
## For each parameter, n\_eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor on split chains (at   
## convergence, Rhat=1).

post3 <- as.data.frame(fit3)  
post3[,1:3] %>%  
 ggpairs() + theme\_tufte(base\_family = 'sans')



FOESTAGE has a small negative coefficient but with a large standard deviation. Besides, there is a strong correlation between FORESTAGE and PERCOVER which may imply that adding only FORESTAGE doesn’t improve the model obviously.

Run model 4, which only use FORESTAGE as predictor for comparing.

m4='  
 data {  
 int N;  
 int T[N];  
 int F[N];  
 }  
 parameters {  
 real a;  
 real bf;  
 }  
 model {  
 vector[N] lambda;  
 a ~ normal(0,100);  
 bf ~ normal(0,1);  
 for(i in 1:N) lambda[i] = a + bf \* F[i];  
 T ~ poisson\_log(lambda);  
 }  
 generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] lambda;  
 for(i in 1:N) {  
 lambda[i] = a + bf \* F[i];  
 log\_lik[i] = poisson\_log\_lpmf(T[i] | lambda[i]);  
 }  
 }  
 }  
 '

Run

dat = list(N = NROW(d),   
 T = d$SALAMAN,  
 F = d$logFORESTAGE\_de)  
  
fit4 <- stan(model\_code = m4,   
 data = dat,   
 iter = 1000,   
 chains = 2,   
 cores = 2)

Next, I’m gonna compare the models using WAIC:

library(loo)  
fit\_list <- list(fit2, fit3, fit4)  
# extract log likelihoods(Apply a Function over a List)  
ll\_list <- lapply(fit\_list, extract\_log\_lik)   
# exponentiate  
exp\_ll\_list <- lapply(ll\_list, exp)  
# get relative neff  
rel\_n\_eff\_list <- lapply(exp\_ll\_list, relative\_eff, chain\_id = c(rep(1, 500), rep(2, 500)))  
# loo  
waic\_list <- list()  
for(i in 1:3) {  
waic\_list[[i]] <- waic(ll\_list[[i]], r\_eff = rel\_n\_eff\_list[[i]], cores = 4)  
}  
names(waic\_list) <- c('fit2', 'fit3', 'fit4')  
loo::compare(x = waic\_list)

## elpd\_diff se\_diff elpd\_waic p\_waic waic   
## fit2 0.0 0.0 -106.5 4.5 213.1  
## fit3 -2.0 1.9 -108.5 7.8 217.0  
## fit4 -21.9 9.9 -128.5 7.0 256.9

The first model still performs best, whereas the model using only forest age performs much worse than the other two models. And we can see that adding FORESTAGE to the model doesn’t make the model perform more better.