Assign4

Ryan Huang

4/30/2019

## Question\_1

data('rugged', package = 'rethinking')  
d <- rugged; rm(rugged)  
d <-  
 d %>%  
 mutate(loggdp = log(rgdppc\_2000))  
d <-  
 d %>%   
 filter(complete.cases(rgdppc\_2000))

Question\_1(a): Using rstan to fit just the interaction model

m1.1 = "  
data{  
 int N;  
 vector[N] loggdp;  
 vector[N] rugged;  
 vector[N] cont\_africa;  
}  
parameters{  
 real alpha;  
 real bA;  
 real bR;  
 real bAR;  
 real sigma;  
}  
model{  
 // model  
 vector[N] mu;  
 for (i in 1:N){  
 mu[i] = alpha + bA \* cont\_africa[i] + bR \* rugged[i] + bAR \* cont\_africa[i] \* rugged[i];  
 }  
 //prior  
 alpha ~ normal(8,100);  
 bR ~ normal(0,1);  
 bA ~ normal(0,1);  
 bAR ~ normal(0,1);  
  
 //likelihood  
 loggdp ~ normal(mu,sigma);  
}  
generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] mu;  
 for(n in 1:N) {  
 mu[n] = alpha + bA \* cont\_africa[n] + bR \* rugged[n] + bAR \* cont\_africa[n] \* rugged[n];   
 log\_lik[n] = normal\_lpdf(loggdp[n] | mu[n], sigma);  
 }   
 }  
}  
"  
# Not for now  
"  
generated quantities {  
 vector[N] log\_lik;  
 {  
 vector[N] mu;  
 for(n in 1:N) {  
 mu[n] = alpha + bA \* cont\_africa[n] + bR \* rugged[n] + bAR \* cont\_africa[n] \* rugged[n];   
 log\_lik[n] = normal\_lpdf(loggdp[n] | mu[n], sigma);  
 }   
 }  
}"  
  
dat1.1 <- list(  
 N = NROW(d),  
 loggdp = d$loggdp,  
 rugged = d$rugged,   
 cont\_africa = d$cont\_africa  
)  
fit1.1 = stan(model\_code = m1.1, data = dat1.1, cores = 4, chains = 4, iter = 3000)

print(fit1.1, probs = c(0.1, 0.5, 0.9), pars = c('alpha', 'bA', 'bR', 'bAR  
', 'sigma'))

## Inference for Stan model: 2c64889b973c65746a42b5204d1853b9.  
## 4 chains, each with iter=3000; warmup=1500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 10% 50% 90% n\_eff Rhat  
## alpha 9.18 0 0.14 9.00 9.18 9.36 3040 1  
## bA -1.84 0 0.22 -2.13 -1.84 -1.56 3229 1  
## bR -0.18 0 0.08 -0.28 -0.18 -0.09 3014 1  
## bAR 0.35 0 0.13 0.18 0.35 0.51 3070 1  
## sigma 0.95 0 0.05 0.88 0.95 1.02 4014 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 2 01:52:15 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).

log\_lik\_1.1 <- extract\_log\_lik(fit1.1, merge\_chains = FALSE)

Fit model w/o Seychelles

d1.2 = d %>% filter(country != "Seychelles")  
dat1.2 <- list(  
 N = NROW(d1.2),  
 loggdp = d1.2$loggdp,  
 rugged = d1.2$rugged,   
 cont\_africa = d1.2$cont\_africa  
)  
fit1.2 = stan(model\_code = m1.1, data = dat1.2, cores = 4, chains = 4, iter = 3000)

print(fit1.2, probs = c(0.1, 0.5, 0.9), pars = c('alpha', 'bA', 'bR', 'bAR  
', 'sigma'))

## Inference for Stan model: 2c64889b973c65746a42b5204d1853b9.  
## 4 chains, each with iter=3000; warmup=1500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 10% 50% 90% n\_eff Rhat  
## alpha 9.19 0 0.14 9.00 9.19 9.37 2500 1  
## bA -1.79 0 0.23 -2.07 -1.78 -1.50 2614 1  
## bR -0.19 0 0.08 -0.29 -0.19 -0.08 2470 1  
## bAR 0.26 0 0.14 0.07 0.26 0.44 2674 1  
## sigma 0.94 0 0.05 0.88 0.94 1.02 3571 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 2 01:52:19 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).

log\_lik\_1.2 <- extract\_log\_lik(fit1.2, merge\_chains = FALSE)

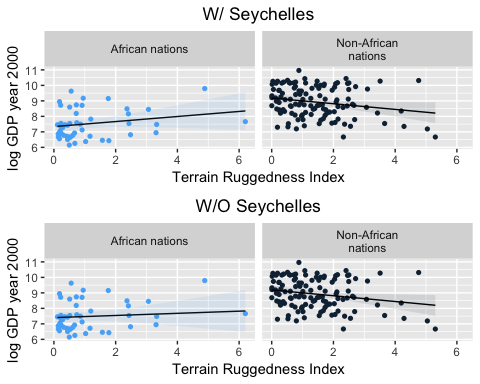
coeftab(fit1.1, fit1.2)

## fit1.1 fit1.2   
## alpha 9.18 9.19  
## bA -1.84 -1.79  
## bR -0.18 -0.19  
## bAR 0.35 0.26  
## sigma 0.95 0.94  
## log\_lik[1] -0.88 -0.87  
## log\_lik[2] -0.95 -0.94  
## log\_lik[3] -1.32 -1.31  
## log\_lik[4] -0.95 -0.94  
## log\_lik[5] -1.33 -1.32  
## log\_lik[6] -0.88 -0.87  
## log\_lik[7] -1.42 -1.42  
## log\_lik[8] -2.56 -2.59  
## log\_lik[9] -1.46 -1.46  
## log\_lik[10] -1.68 -1.55  
## log\_lik[11] -1.55 -1.55  
## log\_lik[12] -1.02 -1.04  
## log\_lik[13] -1.01 -1.02  
## log\_lik[14] -2.78 -2.81  
## log\_lik[15] -0.90 -0.89  
## log\_lik[16] -1.04 -1.03  
## log\_lik[17] -1.06 -1.05  
## log\_lik[18] -0.89 -0.89  
## log\_lik[19] -1.13 -1.13  
## log\_lik[20] -0.96 -0.96  
## log\_lik[21] -1.74 -1.74  
## log\_lik[22] -0.91 -0.90  
## log\_lik[23] -1.09 -1.09  
## log\_lik[24] -2.28 -2.21  
## log\_lik[25] -0.94 -0.95  
## log\_lik[26] -1.64 -1.64  
## log\_lik[27] -3.14 -3.19  
## log\_lik[28] -0.96 -0.96  
## log\_lik[29] -1.05 -1.04  
## log\_lik[30] -0.88 -0.88  
## log\_lik[31] -0.89 -0.88  
## log\_lik[32] -1.34 -1.36  
## log\_lik[33] -1.02 -1.05  
## log\_lik[34] -0.92 -0.91  
## log\_lik[35] -1.00 -0.93  
## log\_lik[36] -1.21 -1.36  
## log\_lik[37] -0.91 -0.91  
## log\_lik[38] -1.73 -1.74  
## log\_lik[39] -1.09 -1.08  
## log\_lik[40] -1.51 -1.51  
## log\_lik[41] -0.91 -0.89  
## log\_lik[42] -1.02 -1.01  
## log\_lik[43] -1.57 -1.57  
## log\_lik[44] -0.88 -0.87  
## log\_lik[45] -1.65 -1.63  
## log\_lik[46] -1.25 -1.25  
## log\_lik[47] -1.18 -1.18  
## log\_lik[48] -1.37 -1.21  
## log\_lik[49] -1.56 -1.57  
## log\_lik[50] -0.88 -0.87  
## log\_lik[51] -1.6 -1.5  
## log\_lik[52] -1.46 -1.46  
## log\_lik[53] -0.97 -0.97  
## log\_lik[54] -1.64 -1.64  
## log\_lik[55] -1.89 -1.83  
## log\_lik[56] -1.55 -1.55  
## log\_lik[57] -1.42 -1.41  
## log\_lik[58] -0.90 -0.88  
## log\_lik[59] -0.89 -0.88  
## log\_lik[60] -0.88 -0.87  
## log\_lik[61] -1.20 -1.21  
## log\_lik[62] -3.57 -3.58  
## log\_lik[63] -1.62 -1.63  
## log\_lik[64] -0.88 -0.88  
## log\_lik[65] -1.05 -1.05  
## log\_lik[66] -1.25 -1.25  
## log\_lik[67] -2.02 -2.03  
## log\_lik[68] -1.39 -1.39  
## log\_lik[69] -0.90 -0.89  
## log\_lik[70] -1.75 -1.76  
## log\_lik[71] -0.95 -0.94  
## log\_lik[72] -1.42 -1.42  
## log\_lik[73] -1.69 -1.69  
## log\_lik[74] -1.73 -1.73  
## log\_lik[75] -0.88 -0.87  
## log\_lik[76] -1.90 -1.91  
## log\_lik[77] -1.69 -1.69  
## log\_lik[78] -1.96 -1.97  
## log\_lik[79] -1.11 -1.11  
## log\_lik[80] -1.17 -1.17  
## log\_lik[81] -1.94 -1.95  
## log\_lik[82] -1.14 -1.14  
## log\_lik[83] -1.03 -1.03  
## log\_lik[84] -1.51 -1.50  
## log\_lik[85] -2.22 -2.24  
## log\_lik[86] -0.89 -0.88  
## log\_lik[87] -1.29 -1.29  
## log\_lik[88] -1.04 -1.03  
## log\_lik[89] -1.90 -1.91  
## log\_lik[90] -0.90 -0.89  
## log\_lik[91] -0.89 -0.88  
## log\_lik[92] -1.29 -1.30  
## log\_lik[93] -1.31 -1.10  
## log\_lik[94] -0.88 -0.87  
## log\_lik[95] -2.97 -2.99  
## log\_lik[96] -0.90 -0.89  
## log\_lik[97] -1.21 -1.21  
## log\_lik[98] -0.99 -1.09  
## log\_lik[99] -2.76 -2.79  
## log\_lik[100] -1.25 -1.20  
## log\_lik[101] -0.91 -0.90  
## log\_lik[102] -0.88 -0.87  
## log\_lik[103] -1.16 -1.20  
## log\_lik[104] -1.33 -1.33  
## log\_lik[105] -2.30 -2.31  
## log\_lik[106] -1.12 -1.12  
## log\_lik[107] -0.89 -0.88  
## log\_lik[108] -2.45 -2.51  
## log\_lik[109] -1.59 -1.55  
## log\_lik[110] -0.88 -0.87  
## log\_lik[111] -1.71 -1.75  
## log\_lik[112] -1.56 -1.56  
## log\_lik[113] -1.25 -1.29  
## log\_lik[114] -1.09 -1.11  
## log\_lik[115] -1.33 -1.33  
## log\_lik[116] -1.53 -1.53  
## log\_lik[117] -2.48 -2.50  
## log\_lik[118] -1.55 -1.55  
## log\_lik[119] -1.52 -1.52  
## log\_lik[120] -1.00 -0.99  
## log\_lik[121] -1.76 -1.77  
## log\_lik[122] -0.89 -0.88  
## log\_lik[123] -1.00 -0.99  
## log\_lik[124] -1.02 -1.01  
## log\_lik[125] -1.6 -1.6  
## log\_lik[126] -0.88 -0.88  
## log\_lik[127] -1.54 -1.54  
## log\_lik[128] -1.29 -1.29  
## log\_lik[129] -1.16 -1.16  
## log\_lik[130] -1.82 -1.83  
## log\_lik[131] -0.91 -0.91  
## log\_lik[132] -0.89 -0.88  
## log\_lik[133] -1.40 -1.18  
## log\_lik[134] -0.97 -0.96  
## log\_lik[135] -0.88 -0.87  
## log\_lik[136] -0.89 -0.89  
## log\_lik[137] -1.32 -1.31  
## log\_lik[138] -1.92 -1.93  
## log\_lik[139] -1.78 -1.81  
## log\_lik[140] -0.98 -0.97  
## log\_lik[141] -0.98 -0.97  
## log\_lik[142] -1.44 -1.45  
## log\_lik[143] -1.56 -1.56  
## log\_lik[144] -1.11 -1.30  
## log\_lik[145] -2.50 -1.39  
## log\_lik[146] -1.39 -1.17  
## log\_lik[147] -1.15 -0.89  
## log\_lik[148] -0.89 -0.90  
## log\_lik[149] -0.91 -2.25  
## log\_lik[150] -2.25 -1.36  
## log\_lik[151] -1.35 -0.90  
## log\_lik[152] -0.91 -0.87  
## log\_lik[153] -0.88 -1.81  
## log\_lik[154] -1.80 -0.87  
## log\_lik[155] -0.88 -1.67  
## log\_lik[156] -1.67 -0.94  
## log\_lik[157] -0.95 -1.22  
## log\_lik[158] -1.22 -0.87  
## log\_lik[159] -0.88 -2.05  
## log\_lik[160] -2.04 -2.63  
## log\_lik[161] -2.61 -0.87  
## log\_lik[162] -0.88 -0.97  
## log\_lik[163] -0.97 -1.68  
## log\_lik[164] -1.68 -1.23  
## log\_lik[165] -1.23 -0.95  
## log\_lik[166] -0.96 -3.33  
## log\_lik[167] -3.30 -2.37  
## log\_lik[168] -2.17 -1.22  
## log\_lik[169] -1.21 -0.94  
## log\_lik[170] -0.93 NA  
## lp\_\_ -77.64 -75.74  
## nobs 0 0

Comparing the 2 models, we can tell that the coefficients of the interaction term is decreased from 0.35 to 0.25 in the model w/o Seychelles.

Question\_1(b): Plot the predictions of the interaction model, with and without Seychelles.

# Plot w/ Seychelles.  
post1.1 <- as.data.frame(fit1.1)  
f\_mu\_1.1 <- function(rugged, cont\_africa) with(post1.1,  
alpha + bR \* rugged + bAR \* rugged \* cont\_africa + bA \* cont\_africa )  
  
mu\_1.1 <- mapply(f\_mu\_1.1, rugged = d$rugged, cont\_africa = d$cont\_africa)  
mu\_1.1\_mean <- apply(mu\_1.1, 2, mean)  
mu\_1.1\_pi <- apply(mu\_1.1, 2, rethinking::PI, prob = .97)  
d\_1.1 <- d %>%  
 mutate(mu\_mean = mu\_1.1\_mean,   
 mu\_pi\_l = mu\_1.1\_pi[1,],   
 mu\_pi\_h = mu\_1.1\_pi[2,],  
 inAfrica = ifelse(cont\_africa, 'African nations', 'Non-African  
nations'))  
  
fig1 = d\_1.1 %>%  
 ggplot(aes(x = rugged)) +  
 geom\_point(aes(rugged, loggdp, color = cont\_africa), shape = 16) +  
 theme(legend.position = '') +  
 geom\_line(aes(rugged, mu\_mean)) +  
 geom\_ribbon(aes(x=rugged,ymin=mu\_pi\_l, ymax=mu\_pi\_h,fill = cont\_africa),  
 alpha = .1) +  
 facet\_wrap(~inAfrica) +   
 labs(x = 'Terrain Ruggedness Index', y = 'log GDP year 2000') +  
 ggtitle("W/ Seychelles") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
# Plot w/o Seychelles.  
post1.2 <- as.data.frame(fit1.2)  
f\_mu\_1.2 <- function(rugged, cont\_africa) with(post1.2, alpha + bR \* rugged + bAR \* rugged \* cont\_africa + bA \* cont\_africa )  
  
mu\_1.2 <- mapply(f\_mu\_1.2, rugged = d$rugged, cont\_africa = d$cont\_africa)  
mu\_1.2\_mean <- apply(mu\_1.2, 2, mean)  
mu\_1.2\_pi <- apply(mu\_1.2, 2, rethinking::PI, prob = .97)  
d\_1.2 <- d %>%  
 mutate(mu\_mean = mu\_1.2\_mean,   
 mu\_pi\_l = mu\_1.2\_pi[1,],   
 mu\_pi\_h = mu\_1.2\_pi[2,],  
 inAfrica = ifelse(cont\_africa, 'African nations', 'Non-African  
nations'))  
  
fig2 = d\_1.2 %>%  
 ggplot(aes(x = rugged)) +  
 geom\_point(aes(rugged, loggdp, color = cont\_africa), shape = 16) +  
 theme(legend.position = '') +  
 geom\_line(aes(rugged, mu\_mean)) +  
 geom\_ribbon(aes(x=rugged,ymin=mu\_pi\_l, ymax=mu\_pi\_h,fill = cont\_africa),  
 alpha = .1) +  
 facet\_wrap(~inAfrica) +   
 labs(x = 'Terrain Ruggedness Index', y = 'log GDP year 2000') +  
 ggtitle("W/O Seychelles") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
grid.arrange(fig1, fig2, nrow=2)



Question\_1(c): Conduct a model comparison analysis, using WAIC. Fit three models to the data without Seychelles. Then make model-averaged predictions of the 3 models.

Model\_1: only consider rugged

m3.1 = "  
data{  
 int N;  
 vector[N] loggdp;  
 vector[N] rugged;  
}  
parameters{  
 real alpha;  
 real bR;  
real<lower=0, upper=10> sigma;  
}  
model{  
 // model  
 vector[N] mu;  
 for (i in 1:N){  
 mu[i] = alpha + bR \* rugged[i];  
 }  
 // prior  
 alpha ~ normal(8, 0.5);  
 bR ~ normal(0, 0.25);  
   
 // likelihood  
 loggdp ~ normal(mu, sigma);  
}  
generated quantities{  
 vector[N] log\_lik;  
 {  
 vector[N] mu;  
 for(i in 1:N){  
 mu[i] = alpha + bR \* rugged[i];  
 log\_lik[i] = normal\_lpdf(loggdp[i] | mu[i], sigma);   
 }  
 }  
}  
"

Model 2: Consider both rugged and cont\_africa w/o interaction

m3.2 = "  
data{  
 int N;  
 vector[N] loggdp;  
 vector[N] rugged;  
 vector[N] cont\_africa;  
}  
parameters{  
 real alpha;  
 real bR;  
 real bA;  
 real<lower=0, upper=5> sigma;  
}  
model{  
 //model  
 vector[N] mu;  
 for (i in 1:N){  
 mu[i] = alpha + bA \* cont\_africa[i] + bR \* rugged[i];  
 }  
 //prior  
 alpha ~ normal(8, 0.5);  
 bR ~ normal(0, 0.5);  
 bA ~ normal(0, 0.5);  
   
 //likelihood  
 loggdp ~ normal(mu, sigma);  
}  
generated quantities{  
 vector[N] log\_lik;  
{  
 vector[N] mu;  
 for(i in 1:N){  
 mu[i] = alpha + bR \* rugged[i] + bA \* cont\_africa[i];  
 log\_lik[i] = normal\_lpdf(loggdp[i] | mu[i], sigma);  
 }  
 }  
}  
"

Model 3: Consider R, A and the interaction.

m3.3 = m1.1

Create data sets for 3 models

dat3.1 = list(  
 N = NROW(d1.2),  
 loggdp = d1.2$loggdp,  
 rugged = d1.2$rugged  
)  
  
dat3.2 = list(  
 N = NROW(d1.2),  
 loggdp = d1.2$loggdp,  
 rugged = d1.2$rugged,   
 cont\_africa = d1.2$cont\_africa  
)  
  
dat3.3 = dat3.2

Fit model 3.1

fit3.1 = stan(model\_code = m3.1, data = dat3.1, cores = 4, chains = 4, iter = 3000)

Fit model 3.2

fit3.2 = stan(model\_code = m3.2, data = dat3.2, cores = 4, chains = 4, iter = 3000)

Fit model 3.3

fit3.3 = stan(model\_code = m3.3, data = dat3.3, cores = 4, chains = 4, iter = 3000)

Model Comparison

# extract log likelihood from the fitted model and use WAIC.  
log\_lik\_3.1 = extract\_log\_lik(fit3.1, merge\_chains = FALSE)  
log\_lik\_3.2 = extract\_log\_lik(fit3.2, merge\_chains = FALSE)  
log\_lik\_3.3 = extract\_log\_lik(fit3.3, merge\_chains = FALSE)  
waic\_3.1 = waic(log\_lik\_3.1)  
waic\_3.2 = waic(log\_lik\_3.2)  
waic\_3.3 = waic(log\_lik\_3.3)

## Warning: 1 (0.6%) p\_waic estimates greater than 0.4. We recommend trying  
## loo instead.

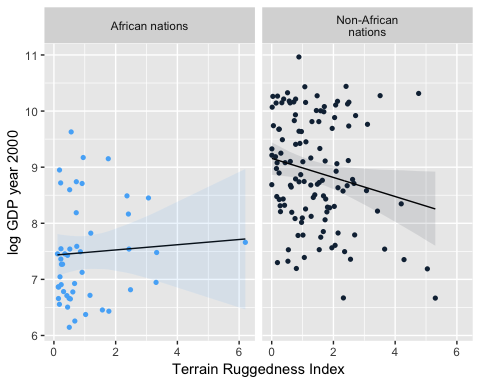
mod\_comp <- loo::compare(waic\_3.1, waic\_3.2, waic\_3.3)  
mod\_comp

## elpd\_diff elpd\_waic se\_elpd\_waic p\_waic se\_p\_waic waic se\_waic  
## waic\_3.3 0.0 -231.6 7.2 4.3 0.6 463.2 14.4   
## waic\_3.2 -1.9 -233.5 6.8 3.6 0.5 467.0 13.5   
## waic\_3.1 -36.4 -267.9 6.5 2.4 0.3 535.9 13.0

From the above comparison report, we can tell that model 3.3 with interaction term is still the best model regarding waic.

Plot the model-averaged predictions of the above 3 models.

# get posterior of parameters  
post3.1 <- as.data.frame(fit3.1) %>% select(alpha, bR)  
post3.2 <- as.data.frame(fit3.2) %>% select(alpha, bR, bA)  
post3.3 <- as.data.frame(fit3.3) %>% select(alpha, bR, bA, bAR)  
  
# posterior predictors for model 3.1  
f\_mu\_3.1 = function(rugged) with(post3.1, alpha + bR \* rugged)  
mu\_3.1 = d1.2$rugged %>% purrr::map(f\_mu\_3.1)  
mu\_3.1\_mean = mu\_3.1 %>% purrr::map(mean) %>% purrr::flatten\_dbl()  
mu\_3.1\_pi = mu\_3.1 %>% purrr::map(rethinking::PI, prob = .97) %>% unlist()  
  
# posterior predictors for model 3.2  
f\_mu\_3.2 = function(rugged, cont\_africa) with(post3.2, alpha + bR \* rugged + bA \* cont\_africa)  
mu\_3.2 = purrr::map2(d1.2$rugged,d1.2$cont\_africa,f\_mu\_3.2)  
mu\_3.2\_mean = mu\_3.2 %>% purrr::map(mean) %>% purrr::flatten\_dbl()  
mu\_3.2\_pi = mu\_3.2 %>% purrr::map(rethinking::PI, prob = .97) %>% unlist()  
  
# posterior predictors for model 3.3  
f\_mu\_3.3 = function(rugged,cont\_africa) with(post3.3, alpha + bR \* rugged + bAR \* rugged \* cont\_africa + bA \* cont\_africa)  
mu\_3.3 = purrr::map2(d1.2$rugged, d1.2$cont\_africa, f\_mu\_3.3)  
mu\_3.3\_mean = mu\_3.3 %>% purrr::map(mean) %>% purrr::flatten\_dbl()  
mu\_3.3\_pi = mu\_3.3 %>% purrr::map(rethinking::PI, prob = .97) %>% unlist()  
  
library(loo)  
r\_eff <- relative\_eff(exp(log\_lik\_3.1))  
loo\_3.1 <- loo(log\_lik\_3.1, r\_eff = r\_eff, cores = 2)  
r\_eff <- relative\_eff(exp(log\_lik\_3.2))  
loo\_3.2 <- loo(log\_lik\_3.2, r\_eff = r\_eff, cores = 2)  
r\_eff <- relative\_eff(exp(log\_lik\_3.3))  
loo\_3.3 <- loo(log\_lik\_3.3, r\_eff = r\_eff, cores = 2)  
  
weights <- loo\_model\_weights(list(loo\_3.1, loo\_3.2, loo\_3.3), method = "pseudobma", BB = F)  
  
ensemble\_mean <- weights[1] \* mu\_3.1\_mean + weights[2] \* mu\_3.2\_mean + weights[3] \* mu\_3.3\_mean  
ensemble\_PI <- weights[1] \* mu\_3.1\_pi + weights[2] \* mu\_3.2\_pi + weights[3] \* mu\_3.3\_pi  
  
# Prepare data for plotting  
  
d1.2 <-   
 d1.2 %>%  
 mutate(ens\_mean = ensemble\_mean,  
 ens\_PI\_l = ensemble\_PI[seq(from=1, to=2\*nrow(d1.2), by = 2)],  
 ens\_PI\_h = ensemble\_PI[seq(from=2, to=2\*nrow(d1.2), by = 2)],  
 inAfrica = ifelse(cont\_africa, 'African nations', 'Non-African  
nations'))  
  
# Plotting the result  
d1.2 %>%  
 ggplot(aes(x = rugged)) +  
 geom\_point(aes(rugged, loggdp, color = cont\_africa), shape = 16,) +  
 theme(legend.position = '') +  
 geom\_line(aes(rugged, ens\_mean)) +  
 geom\_ribbon(aes(x=rugged,ymin=ens\_PI\_l, ymax=ens\_PI\_h, fill = cont\_africa),  
 alpha = .1) +  
 facet\_grid(~inAfrica) +  
 labs(x = 'Terrain Ruggedness Index', y = 'log GDP year 2000')



## Question\_2

data('nettle', package = 'rethinking')  
d2 <- nettle; rm(nettle)  
d2$lang.per.cap <- d2$num.lang / d2$k.pop  
d2$log\_lpc <- log(d2$lang.per.cap)  
d2$log\_area <- log(d2$area)  
d2$log\_area\_c <- d2$log\_area - mean(d2$log\_area)  
d2$mgs\_c <- d2$mean.growing.season - mean(d2$mean.growing.season)  
d2$sgs\_c <- d2$sd.growing.season - mean(d2$sd.growing.season)

Question\_2(a): Is log(lang.per\_cap) positively associated with mean.growing.season? Consider log(area) as a covariate.

m2.1 = "  
data{   
 int N;  
 vector[N] log\_lpc;  
 vector[N] log\_area;  
 vector[N] mgs\_c;  
}  
parameters{  
 real alpha;  
 real bA; //beta of area  
 real bM; // beta of mean  
 real<lower=0, upper=5> sigma;  
}  
model{  
 // model  
 vector[N] mu;  
 for(i in 1:N){  
 mu[i] = alpha + bA \* log\_area[i] + bM \* mgs\_c[i];  
 }  
  
 // Prior  
 alpha ~ normal(-5, 1);  
 bA ~ normal(0, 0.5);  
 bM ~ normal(0, 0.25);  
  
 // likelihood  
 log\_lpc ~ normal(mu, sigma);  
}  
"  
dat2.1 = list(  
 N = NROW(d2),  
 log\_lpc = d2$log\_lpc,  
 log\_area = d2$log\_area\_c,  
 mgs\_c = d2$mgs\_c  
)  
fit2.1 = stan(model\_code = m2.1, data = dat2.1, cores = 4, chains = 4, iter = 3000)

print(fit2.1)

## Inference for Stan model: 364822bd6f35bc1efdc3e69ae50a6789.  
## 4 chains, each with iter=3000; warmup=1500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## alpha -5.44 0.00 0.17 -5.77 -5.55 -5.44 -5.33 -5.11 5876 1  
## bA -0.19 0.00 0.14 -0.46 -0.28 -0.19 -0.10 0.08 5374 1  
## bM 0.14 0.00 0.06 0.03 0.10 0.14 0.18 0.25 5778 1  
## sigma 1.44 0.00 0.13 1.22 1.35 1.43 1.52 1.71 4888 1  
## lp\_\_ -63.78 0.03 1.52 -67.65 -64.48 -63.42 -62.67 -61.93 2453 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 2 01:55:39 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).

Q2(b)

m2.2 = "  
data{   
 int N;  
 vector[N] log\_lpc;  
 vector[N] log\_area;  
 vector[N] sgs\_c;  
}  
parameters{  
 real alpha;  
 real bA; //beta of area  
 real bS; // beta of mean  
 real<lower=0, upper=5> sigma;  
}  
model{  
 // model  
 vector[N] mu;  
 for(i in 1:N){  
 mu[i] = alpha + bA \* log\_area[i] + bS \* sgs\_c[i];  
 }  
  
 // Prior  
 alpha ~ normal(-5, 1);  
 bA ~ normal(0, 0.5);  
 bS ~ normal(0, 0.25);  
  
 // likelihood  
 log\_lpc ~ normal(mu, sigma);  
}  
"  
dat2.2 = list(  
 N = NROW(d2),  
 log\_lpc = d2$log\_lpc,  
 log\_area = d2$log\_area\_c,  
 sgs\_c = d2$sgs\_c  
)  
fit2.2 = stan(model\_code = m2.2, data = dat2.2, cores = 4, chains = 4, iter = 3000)

print(fit2.2)

## Inference for Stan model: c28848425a824a61ff14775107ac8705.  
## 4 chains, each with iter=3000; warmup=1500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## alpha -5.44 0.00 0.17 -5.78 -5.56 -5.44 -5.33 -5.10 5740 1  
## bA -0.25 0.00 0.15 -0.55 -0.35 -0.25 -0.15 0.04 5036 1  
## bS -0.14 0.00 0.15 -0.44 -0.24 -0.14 -0.03 0.17 4872 1  
## sigma 1.49 0.00 0.13 1.26 1.40 1.48 1.57 1.77 4790 1  
## lp\_\_ -66.44 0.03 1.49 -70.11 -67.18 -66.11 -65.34 -64.58 2577 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 2 01:56:40 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).

Q2(c)

m2.3 = "  
data{   
 int N;  
 vector[N] log\_lpc;  
 vector[N] log\_area;  
 vector[N] sgs\_c;  
 vector[N] mgs\_c;  
}  
parameters{  
 real alpha;  
 real bA; //beta of area  
 real bS; // beta of std  
 real bM; // beta of mean  
 real bSM; // beta of interaction  
 real<lower=0, upper=5> sigma;  
}  
  
model{  
 // model  
 vector[N] mu;  
 for(i in 1:N){  
 mu[i] = alpha + bA \* log\_area[i] + bS \* sgs\_c[i] + bM \* mgs\_c[i] + bSM \* mgs\_c[i] \* sgs\_c[i];  
 }  
  
 // Prior  
 alpha ~ normal(-5, 1);  
 bA ~ normal(0, 0.5);  
 bS ~ normal(0, 0.25);  
 bSM ~ normal(0, 0.5);  
  
 // likelihood  
 log\_lpc ~ normal(mu, sigma);  
}  
"  
dat2.3 = list(  
 N = nrow(d2),  
 log\_lpc = d2$log\_lpc,  
 log\_area = d2$log\_area\_c,  
 sgs\_c = d2$sgs\_c,  
 mgs\_c = d2$mgs\_c  
)  
  
fit2.3 = stan(model\_code = m2.3, data = dat2.3, cores = 4, chains = 4, iter = 3000)

print(fit2.3)

## Inference for Stan model: 8b423531bbf87612bbe5be2aa6048416.  
## 4 chains, each with iter=3000; warmup=1500; thin=1;   
## post-warmup draws per chain=1500, total post-warmup draws=6000.  
##   
## mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  
## alpha -5.44 0.00 0.16 -5.75 -5.55 -5.44 -5.34 -5.12 8736 1  
## bA -0.06 0.00 0.15 -0.36 -0.16 -0.06 0.04 0.24 6318 1  
## bS -0.23 0.00 0.15 -0.51 -0.32 -0.23 -0.13 0.07 5925 1  
## bM 0.11 0.00 0.06 -0.02 0.06 0.11 0.15 0.23 5828 1  
## bSM -0.11 0.00 0.05 -0.21 -0.14 -0.11 -0.08 -0.01 6760 1  
## sigma 1.38 0.00 0.12 1.16 1.29 1.37 1.45 1.63 6814 1  
## lp\_\_ -60.71 0.04 1.82 -65.07 -61.63 -60.37 -59.40 -58.22 2532 1  
##   
## Samples were drawn using NUTS(diag\_e) at Thu May 2 01:57:52 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).

Plotting the prediction

# Break the continous Xs  
d2$mgsc.group <- cut(  
 d2$mgs\_c,   
 breaks = quantile(d2$mgs\_c, probs = c(0, 1/3, 2/3, 1)),  
 include.lowest = TRUE,   
 dig.lab = 2  
)  
  
d2$sgsc.group <- cut(  
 d2$sgs\_c,   
 breaks = quantile(d2$sgs\_c, probs = c(0, 1/3, 2/3, 1)),  
 include.lowest = TRUE,   
 dig.lab = 2  
)  
  
d2 = d2 %>%   
 group\_by(mgsc.group) %>%   
 mutate(mgsc\_gp\_mean = mean(mgs\_c)) %>% ungroup()  
d2 = d2 %>%   
 group\_by(sgsc.group) %>%   
 mutate(sgsc\_gp\_mean = mean(sgs\_c)) %>% ungroup()  
  
post2.3 = as.data.frame(fit2.3)  
f\_mu\_2.3 = function(log\_area,sgs\_c,mgs\_c) with(post2.3, alpha + bA \* log\_area + bS \* sgs\_c + bM \* mgs\_c + bSM \* mgs\_c \* sgs\_c)  
  
# Grouped mgs\_c  
mu\_2.3\_A = purrr::pmap(list(d2$log\_area\_c, d2$sgs\_c, d2$mgsc\_gp\_mean), f\_mu\_2.3)  
mu\_2.3\_mean\_A = mu\_2.3\_A %>% purrr::map(mean) %>% unlist()  
mu\_2.3\_pi\_A = mu\_2.3\_A %>% purrr::map(rethinking::PI, prob = .97) %>% unlist()  
  
# Grouped sgs\_c  
mu\_2.3\_B = purrr::pmap(list(d2$log\_area\_c, d2$sgsc\_gp\_mean, d2$mgs\_c), f\_mu\_2.3)  
mu\_2.3\_mean\_B = mu\_2.3\_B %>% purrr::map(mean) %>% unlist()  
mu\_2.3\_pi\_B = mu\_2.3\_B %>% purrr::map(rethinking::PI, prob = .97) %>% unlist()  
  
  
figA = d2 %>%   
 mutate(  
 mean = mu\_2.3\_mean\_A,  
 pi\_l = mu\_2.3\_pi\_A[seq(from=1, to=2\*nrow(d2), by=2)],  
 pi\_h = mu\_2.3\_pi\_A[seq(from=2, to=2\*nrow(d2), by=2)]) %>%  
 ggplot(aes(x = sgs\_c)) +  
 geom\_point(aes(sgs\_c, log\_lpc), shape = 16,) +  
 theme(legend.position = '') +  
 geom\_line(aes(sgs\_c, mean)) +  
 geom\_ribbon(aes(x=sgs\_c,ymin=pi\_l, ymax=pi\_h),  
 alpha = .1)+  
 facet\_grid(~mgsc\_gp\_mean)  
  
figB = d2 %>%   
 mutate(  
 mean = mu\_2.3\_mean\_B,  
 pi\_l = mu\_2.3\_pi\_B[seq(from=1, to=2\*nrow(d2), by=2)],  
 pi\_h = mu\_2.3\_pi\_B[seq(from=2, to=2\*nrow(d2), by=2)]) %>%  
 ggplot(aes(x = mgs\_c)) +  
 geom\_point(aes(mgs\_c, log\_lpc), shape = 16,) +  
 theme(legend.position = '') +  
 geom\_line(aes(mgs\_c, mean)) +  
 geom\_ribbon(aes(x=mgs\_c,ymin=pi\_l, ymax=pi\_h),  
 alpha = .1)+  
 facet\_grid(~sgsc\_gp\_mean)  
  
grid.arrange(figA, figB, nrow=2)

