Extinction risk from climate change: geography figure

Mar. 25th, 2024

Load libraries and data

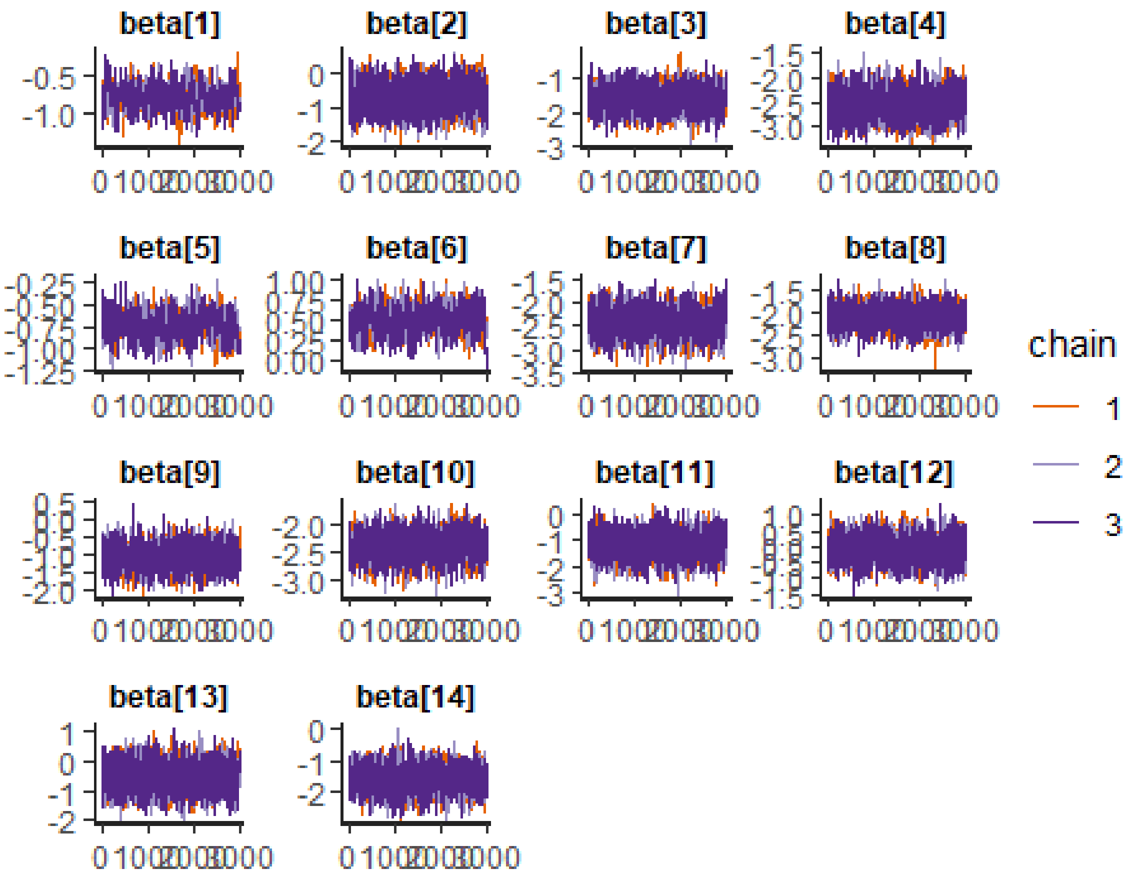
```
rm(list = ls())
 root.dir = "C:/Users/mcu08001/Documents/1New Research/CC MetaRisk2/Analysis"

 #load libraries
library(coda); library(ggplot2); library(rstan); library(bayesplot); library(
shinystan); library(loo); library(rstanarm); library(dplyr); library(ggpubr);
library(rworldmap); library(rnaturalearth); library(sf)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = FALSE)

#load data
dataP<-read.table("MetaRisk2 aggthres 5.txt",header=T); attach(dataP)

#Bayesian stan model proportional and weighted
#betareg requires no 0s or 1s
koffset = 0.001 #the k that gives the best posterior predictive check
percent2 <- adj.percent
percent2[adj.percent == 0] = koffset;
percent2[adj.percent == 1] = 1 - koffset;
dataP$percent2 <- percent2;

data.use<-dataP

N = length(data.use$percent2)
n.Study <- length(unique(data.use$Study)) #number of studies
Studyint<-as.integer(unclass(factor(data.use$Study)))
phi = data.use$Total.N
```

Continent + latitude model

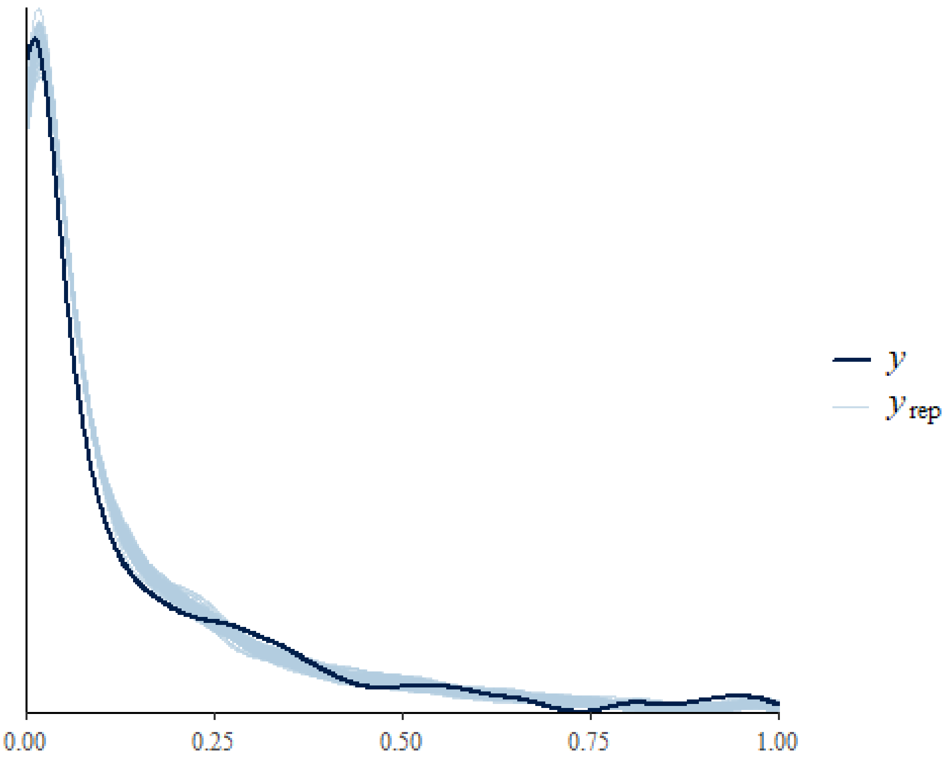
```
#create model matrix for coefficients
#betamat <- model.matrix(~Region + Antarctic + S.Middle + Tropics + N.Middle
+ Arctic, data=data.use) #relative to global mean

betamat <- data.frame(Afr.Tr = rep(1,N),
             Afr.NM = ifelse(data.use$Region == "Africa" & data.use$N.Middle
== "Y",1,0),
             Afr.SM = ifelse(data.use$Region == "Africa" & data.use$S.Middle
== "Y",1,0),
             Asia.NM = ifelse(data.use$Region == "Asia" & data.use$N.Middle =
= "Y",1,0),
             Asia.Tr = ifelse(data.use$Region == "Asia" & data.use$Tropics ==
"Y",1,0),
             ANZ.Tr = ifelse(data.use$Region == "AustraliaNewZealand" & data.
use$Tropics == "Y",1,0),
             ANZ.SM = ifelse(data.use$Region == "AustraliaNewZealand" & data.
use$S.Middle == "Y",1,0),
             Euro.NM = ifelse(data.use$Region == "Europe" & data.use$N.Middle
== "Y",1,0),
             Euro.Arc = ifelse(data.use$Region == "Europe" & data.use$Arctic
== "Y",1,0),
             NAm.NM = ifelse(data.use$Region == "Namerica" & data.use$N.Middl
e == "Y",1,0),
             NAm.Arc = ifelse(data.use$Region == "Namerica" & data.use$Arctic
== "Y",1,0),
             NAm.Tr = ifelse(data.use$Region == "Namerica" & data.use$Tropics
== "Y",1,0),
             SAm.SM = ifelse(data.use$Region == "Samerica" & data.use$S.Middl
e == "Y",1,0),
             SAm.Tr = ifelse(data.use$Region == "Samerica" & data.use$Tropics
== "Y",1,0))

stan.data<-list(N = N, percent = data.use$percent2, betamat = betamat, phi =
phi, S = n.Study, P = ncol(betamat), Study = Studyint)

# params.to.monitor=c("beta","y_rep","stu","sigma_stu", "eta","log_lik")
#
# init.beta=rep(0,ncol(betamat)-1)
# init.fn<- function (chain_id) {
#   list(beta = c(-2.5,init.beta))
# }
#
# mod=stan(file="MetaRisk2 RSTAN beta mat.stan",data=stan.data,pars=params.to
.monitor,
#           chains = 3, warmup=18000, cores=3,iter=21000,
#          init = init.fn, save_warmup = FALSE, control=list(adapt_delta = 0.
9, max_treedepth = 15))#

load("3geo+lat2.rds") #new model with hot coding.

params.to.monitor2=c("beta")#
sumx = summary(mod,probs=c(.025,0.975), digits=4, pars=params.to.monitor2)
sumx$summary

##                 mean     se_mean        sd       2.5%      97.5%     n_ef
f
## beta[1]  -0.77525787 0.009217790 0.1517570 -1.0764836 -0.47043865  271.046
```

```
4
## beta[2]  -0.67739715 0.011147041 0.3714660 -1.4144006  0.05518731 1110.501
0
## beta[3]  -1.60797574 0.010314985 0.3321298 -2.2601086 -0.96633170 1036.760
8
## beta[4]  -2.52903077 0.008683316 0.2648106 -3.0474878 -1.99730686  930.035
7
## beta[5]  -0.73072652 0.008606262 0.1494885 -1.0257281 -0.45272999  301.707
8
## beta[6]   0.50766976 0.008590782 0.1550680  0.2114697  0.81580432  325.821
1
## beta[7]  -2.42146683 0.009408543 0.2653968 -2.9533925 -1.90439132  795.695
2
## beta[8]  -2.08668761 0.008804243 0.2381975 -2.5481217 -1.61955037  731.965
1
## beta[9]  -1.08149862 0.012039921 0.3270982 -1.7207456 -0.43790320  738.089
4
## beta[10] -2.39347006 0.007021656 0.2174032 -2.8230946 -1.97727558  958.634
2
## beta[11] -1.20356311 0.012318846 0.4328687 -2.0705909 -0.35654127 1234.730
7
## beta[12] -0.05681097 0.010098534 0.3600852 -0.7584023  0.65138969 1271.433
8
## beta[13] -0.53952944 0.013163317 0.4261754 -1.3864371  0.30091648 1048.204
4
## beta[14] -1.58003460 0.012096154 0.3626781 -2.2780123 -0.85543993  898.975
7
##              Rhat
## beta[1]  1.008494
## beta[2]  1.002461
## beta[3]  1.001558
## beta[4]  1.001602
## beta[5]  1.003154
## beta[6]  1.006893
## beta[7]  1.004233
## beta[8]  1.003912
## beta[9]  1.002398
## beta[10] 1.000897
## beta[11] 1.002609
## beta[12] 1.001053
## beta[13] 1.000531
## beta[14] 1.000131
```

```
#checks
traceplot(mod,pars=params.to.monitor2,inc_warmup=FALSE)
```



```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```



```
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3235 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo   7602.7  96.0
## p_loo      1993.0  25.0
## looic    -15205.3 192.1
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                         Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1363  42.1%   112
##    (0.7, 1]   (bad)      1619  50.0%   <NA>
##    (1, Inf)   (very bad)  253   7.8%   <NA>
## See help('pareto-k-diagnostic') for details.

mod.add <- mod
loo.mod.add <- loo.mod
```

```
#beta.names = dimnames(model.matrix(~Region + Antarctic + S.Middle + Tropics
+ N.Middle + Arctic, data=data.use))
#cats <- c("Africa", "Asia", "New Zealand/Australia", "Europe", "Oceans", "No
rth America", "South America", "Global", "Antarctic", "S. Middle", "Tropics",
"N. Middle", "Arctic")
#Calculate estimates; ix
posterior=as.data.frame(mod.add);

#absolute risks
Africa.Tr<-posterior[["beta[1]"]]
Africa.N.Mid<-posterior[["beta[1]"]]+posterior[["beta[2]"]]
Africa.S.Mid<-posterior[["beta[1]"]]+posterior[["beta[3]"]]
Asia.N.Mid<-posterior[["beta[1]"]]+posterior[["beta[4]"]]
Asia.Tr<-posterior[["beta[1]"]]+posterior[["beta[5]"]]
Asia.Arct<-NA #base + Asia + Arctic no studies
Aus.NZ.Tr<-posterior[["beta[1]"]]+posterior[["beta[6]"]]
Aus.NZ.S.Mid<-posterior[["beta[1]"]]+posterior[["beta[7]"]]
Euro.N.Mid<-posterior[["beta[1]"]]+posterior[["beta[8]"]]
Euro.Arct<-posterior[["beta[1]"]]+posterior[["beta[9]"]]
N.Am.N.Mid<-posterior[["beta[1]"]]+posterior[["beta[10]"]]
N.Am.Arct<-posterior[["beta[1]"]]+posterior[["beta[11]"]]
N.Am.Tr<-posterior[["beta[1]"]]+posterior[["beta[12]"]]
S.Am.S.Mid<-posterior[["beta[1]"]]+posterior[["beta[13]"]]
S.Am.Tr<-posterior[["beta[1]"]]+posterior[["beta[14]"]]

cats <- c("N. Middle Africa","Tropical Africa", "Middle Africa","Tropical Asi
a","N. Middle Asia","Arctic Asia","Middle Europe","Arctic Europe","Arctic N.
America","Middle N. America","Tropical N. America", "Middle S. America","Trop
ical S. America","Tropical Australia/New Zealand", "Middle Australia/New Zeal
and")
beta.cat<-cbind(Africa.N.Mid,Africa.Tr,Africa.S.Mid,Asia.Tr,Asia.N.Mid,Asia.A
rct,Euro.N.Mid,Euro.Arct,N.Am.Arct,N.Am.N.Mid,N.Am.Tr,S.Am.S.Mid,S.Am.Tr,Aus.
NZ.Tr,Aus.NZ.S.Mid)

#use common median threat
load("2all_interc.rds")
posterior2=as.data.frame(mod)
grand.mean = posterior2[["mu"]]
grand.mean.inv =invlogit(grand.mean)
grand.mean.pred = quantile(grand.mean.inv,probs = c(0.025, 0.5, 0.975))
grand.mean.pred

##       2.5%        50%      97.5%
## 0.06568248 0.07596676 0.08662260

#calculate median and credible intervals
pred.cat = invlogit(apply(beta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975)
,na.rm=TRUE))
pred.cat.df <- data.frame(x = cats,
                          mean = pred.cat[2,],
                          low = pred.cat[1,],
                          hi = pred.cat[3,])

#arrange in increasing order#
pred.cat.df$cats = with (pred.cat.df, reorder(cats, mean))
#pred.cat.df <- pred.cat.df[pred.cat.df$cats != "Oceans",] #remove Oceans, wh
ich is covered by habitat grouping now

rbeta.cat = invlogit(beta.cat)-invlogit(grand.mean)
rpred.cat = (apply(rbeta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975),na.rm
=TRUE))
rpred.cat.df <- data.frame(x = cats,
                          mean = rpred.cat[2,],
                          low = rpred.cat[1,],
                          hi = rpred.cat[3,])
#Differences from zero
glob.mean.over <- rep("n",nrow(rpred.cat.df))
glob.mean.over[rpred.cat.df$mean > 0 & rpred.cat.df$low > 0] = "y"
glob.mean.over[rpred.cat.df$mean < 0 & rpred.cat.df$hi < 0] = "y"
rpred.cat.df$glob.mean.over <- glob.mean.over

#arrange in increasing order#
rpred.cat.df$cats = with (rpred.cat.df, reorder(cats, mean))

pred.cat.df.r <- pred.cat.df[,1:4]
pred.cat.df.r[,2:4] <- round((pred.cat.df.r[,2:4]), digits = 4)
knitr::kable(pred.cat.df.r, caption = "Table 1: Extinction risk by subregion
and difference from mean, with 95% credible intervals", format = "markdown",
font = 5)
```

Table 1: Extinction risk by subregion and difference from mean, with 95% credible intervals

	x	mean	low	hi
Africa.N.Mid	N. Middle Africa	0.1902	0.1011	0.3217
Africa.Tr	Tropical Africa	0.3154	0.2542	0.3845
Africa.S.Mid	Middle Africa	0.0852	0.0471	0.1432
Asia.Tr	Tropical Asia	0.1816	0.1184	0.2692
Asia.N.Mid	N. Middle Asia	0.0353	0.0205	0.0626
Asia.Arct	Arctic Asia	NA	NA	NA
Euro.N.Mid	Middle Europe	0.0540	0.0319	0.0899
Euro.Arct	Arctic Europe	0.1355	0.0714	0.2429
N.Am.Arct	Arctic N. America	0.1226	0.0521	0.2535
N.Am.N.Mid	Middle N. America	0.0406	0.0241	0.0662
N.Am.Tr	Tropical N. America	0.3028	0.1690	0.4884
S.Am.S.Mid	Middle S. America	0.2116	0.0973	0.3991

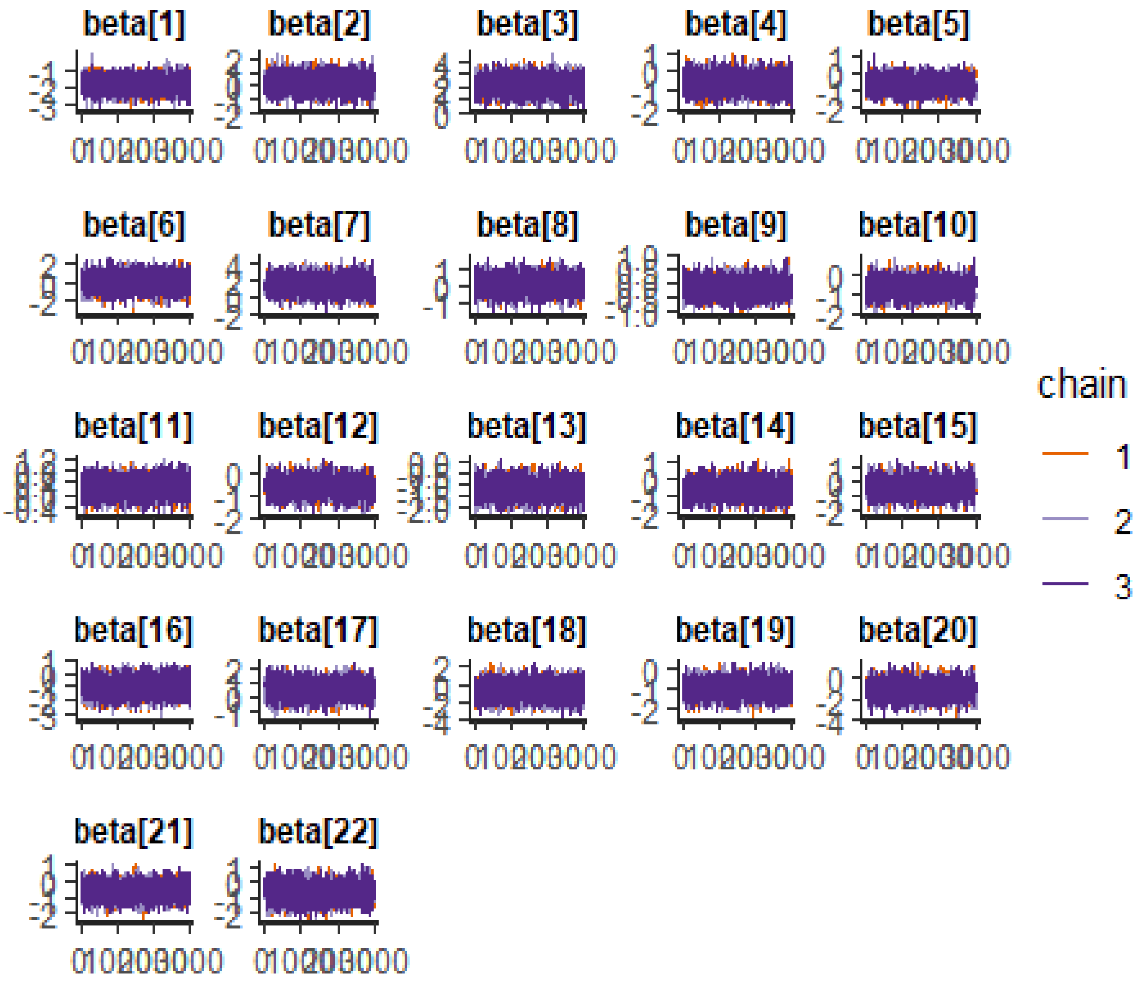
	x	mean	low	hi
S.Am.Tr	Tropical S. America	0.0865	0.0427	0.1701
Aus.NZ.Tr	Tropical Australia/New Zealand	0.4342	0.3197	0.5625
Aus.NZ.S.Mid	Middle Australia/New Zealand	0.0393	0.0226	0.0663

Geography x latitude model

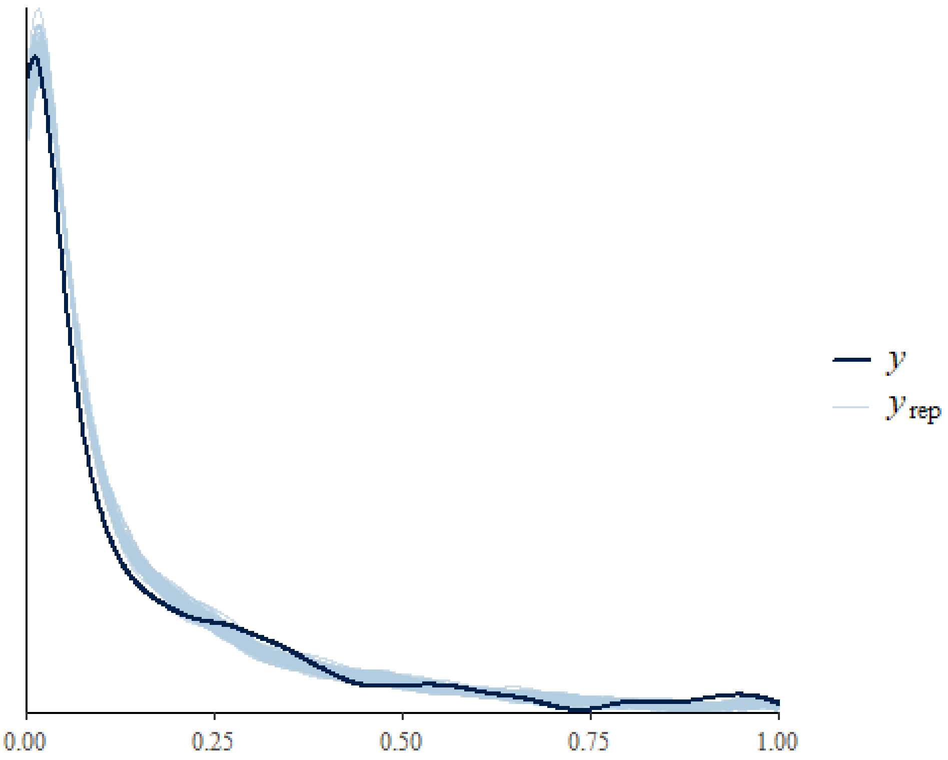
Independent estimate for each continent x latitudinal band region

```
#create model matrix for coefficients
betamat.0 <- model.matrix(~Region * (Antarctic + S.Middle + Tropics + N.Middl
e + Arctic), data=data.use) #relative to global mean
betamat <- betamat.0[,c(1:7,9:13,22,26,28,29,32,33,35,39,44,46)] #cut regions
without a place (e.g., Arctic South America) cut Euro.NM because it exactly =
= Euro

stan.data<-list(N = N, percent = data.use$percent2, betamat = betamat, phi =
phi, S = n.Study, P = ncol(betamat), Study = Studyint)

# params.to.monitor=c("beta","y_rep","stu","sigma_stu", "eta","log_lik")
# init.beta=rep(0,ncol(betamat)-1)
# init.fn<- function (chain_id) {
#   list(beta = c(-2.5,init.beta))
# }
# mod=stan(file="MetaRisk2 RSTAN beta mat.stan",data=stan.data,pars=params.to
.monitor,
#             chains = 3, warmup=18000, cores=3,iter=21000,
#            init = init.fn, save_warmup = FALSE, control=list(adapt_delta = 0.
9, max_treedepth = 15))#

load("3geoxlat3.rds") #fixes europe problem = Europe == Euro.NM
params.to.monitor2=c("beta")#
sumx = summary(mod,probs=c(.025,0.975), digits=4, pars=params.to.monitor2)
sumx$summary
```

```
##                 mean     se_mean        sd       2.5%       97.5%    n_eff
## beta[1]  -1.82079661 0.009114632 0.3749575 -2.5541401 -1.09117895 1692.333
## beta[2]   0.24305565 0.015092021 0.5592585 -0.8710256  1.32094362 1373.189
## beta[3]   2.12942124 0.014851219 0.5987277  0.9596482  3.30419073 1625.303
## beta[4]  -0.60195188 0.011347392 0.4305646 -1.4449670  0.23993365 1439.742
## beta[5]  -0.60839688 0.008241963 0.3836121 -1.3562706  0.14247704 2166.323
## beta[6]   0.07881170 0.016027916 0.7844886 -1.4495209  1.59372716 2395.627
## beta[7]   1.44603305 0.016184881 0.8527150 -0.2002664  3.16276677 2775.804
## beta[8]   0.09033358 0.010951758 0.4844148 -0.8487825  1.04391205 1956.442
## beta[9]  -0.13403095 0.006188036 0.2630643 -0.6399674  0.38410477 1807.250
## beta[10] -0.65670079 0.010145375 0.3690762 -1.3727405  0.06017853 1323.415
## beta[11]  0.26852098 0.008008306 0.2628189 -0.2596476  0.77136718 1077.041
## beta[12] -0.59435768 0.009277630 0.3603180 -1.2993777  0.14259007 1508.335
## beta[13] -1.17425095 0.007717156 0.3238270 -1.8094487 -0.53194559 1760.806
## beta[14] -0.61032492 0.010169358 0.4583525 -1.4843459  0.29687147 2031.478
## beta[15] -0.43098183 0.014873861 0.5505416 -1.4982634  0.67546596 1370.039
## beta[16] -1.01453239 0.014974900 0.5905574 -2.1705106  0.13893255 1555.236
## beta[17]  0.57346311 0.013563834 0.5110024 -0.4351852  1.57679594 1419.322
## beta[18] -0.61670002 0.015403387 0.8088832 -2.2205682  0.96628895 2757.650
## beta[19] -1.04115241 0.010444126 0.3866043 -1.7945775 -0.28479607 1370.216
## beta[20] -1.11159893 0.015691445 0.7259668 -2.5103260  0.32231203 2140.463
## beta[21] -0.56403191 0.011859237 0.4491718 -1.4628874  0.31392568 1434.536
## beta[22] -0.70035003 0.012760103 0.5184468 -1.7119419  0.32105133 1650.821
##               Rhat
## beta[1]  1.0000283
## beta[2]  1.0007366
## beta[3]  1.0001214
## beta[4]  1.0001491
## beta[5]  1.0003062
## beta[6]  1.0008383
## beta[7]  1.0000233
## beta[8]  1.0002222
## beta[9]  1.0008260
## beta[10] 1.0007129
## beta[11] 0.9997631
## beta[12] 1.0004676
## beta[13] 1.0030351
## beta[14] 1.0011682
## beta[15] 1.0005068
## beta[16] 1.0003917
## beta[17] 1.0008833
## beta[18] 1.0000957
## beta[19] 1.0007786
## beta[20] 1.0009837
## beta[21] 1.0008468
## beta[22] 1.0009573

#checks
traceplot(mod,pars=params.to.monitor2,inc_warmup=FALSE)
```



```r
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```



```
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3235 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo   7614.8  96.6
## p_loo      1981.7  25.4
## looic    -15229.6 193.3
## ------
## MCSE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. ESS
## (-Inf, 0.7]   (good)     1383  42.8%   150
##    (0.7, 1]   (bad)      1601  49.5%   <NA>
##    (1, Inf)   (very bad)  251   7.8%   <NA>
## See help('pareto-k-diagnostic') for details.

mod.int <- mod
loo.mod.int <- loo.mod
```

```r
#load continent analysis
load("2geog5.rds")
loo.mod.cont = loo.mod

#load latitudinal band analyss
load("2all_lat.rds")
loo.mod.lat = loo.mod


#load intercept model
load("2all_interc.rds") #use full data (dataP) intercept only model for compa
rison

table.data<-data.frame(
  Model = c("Intercept model","Continent model","Latitude model","Additive mo
del","Interaction model"),
  LOOic = c(loo.mod$estimates[3],loo.mod.cont$estimates[3],loo.mod.lat$estima
tes[3],loo.mod.add$estimates[3],loo.mod.int$estimates[3]),
  SE = c(loo.mod$estimates[6],loo.mod.cont$estimates[6],loo.mod.lat$estimates
[6],loo.mod.add$estimates[6],loo.mod.int$estimates[6])
)

loo.diff = loo.mod.int$estimates[3] - loo.mod.add$estimates[3]; cat("Interact
ion model versus additive model =", loo.diff)

## Interaction model versus additive model = -24.25334

print("                    ")

## [1] "                    "

loo.diff2 = loo.mod.int$estimates[3] - loo.mod$estimates[3]; cat("Interactive
model versus intercept model =", loo.diff2)

## Interactive model versus intercept model = -15.78645

print("                    ")

## [1] "                    "

loo.diff3 = loo.mod.int$estimates[3] - loo.mod.cont$estimates[3]; cat("Intera
ctive model versus continent model =", loo.diff3)

## Interactive model versus continent model = -11.31166

print("                    ")

## [1] "                    "

loo.diff4 = loo.mod.int$estimates[3] - loo.mod.lat$estimates[3]; cat("Interac
tive model versus latitude model =", loo.diff4)
```

```
## Interactive model versus latitude model = -20.5744
```

```
knitr::kable(table.data, caption = "Table 2: Comparisons of LOOic for additiv
e and interactive geographic models", format = "markdown")
```

Table 2: Comparisons of LOOic for additive and interactive geographic models

Model	LOOic	SE
Intercept model	-15213.77	193.3243
Continent model	-15218.24	193.7686
Latitude model	-15208.98	193.3504
Additive model	-15205.30	192.0804
Interaction model	-15229.56	193.2770

LOOIC results

The LOOIC of the interactive model is the best model.

```
betamat.0 <- model.matrix(~Region * (Antarctic + S.Middle + Tropics + N.Middl
e + Arctic), data=data.use) #relative to global mean
betamat <- betamat.0[,c(1:7,9:13,22,26,28,29,32,33,35,39,44,46)]
beta.names = dimnames(betamat)

posterior=as.data.frame(mod.int);

#absolute risks
Africa<-posterior[["beta[1]"]]
Asia<-posterior[["beta[2]"]]
Aus.NZ<-posterior[["beta[3]"]]
Euro<-posterior[["beta[4]"]]
Marine<-posterior[["beta[5]"]]
N.Am<-posterior[["beta[6]"]]
S.Am<-posterior[["beta[7]"]]
#beta.8<-posterior[["beta[8]"]] global don't need
Antarct<-posterior[["beta[8]"]]
S.Mid<-posterior[["beta[9]"]]
Tropics<-posterior[["beta[10]"]]
N.Mid<-posterior[["beta[11]"]]
Arctic<-posterior[["beta[12]"]]

ANZ.SM <-posterior[["beta[13]"]]
S.Am.SM <- posterior[["beta[14]"]]

Asia.Tropics <- posterior[["beta[15]"]]
ANZ.Tropics <- posterior[["beta[16]"]]
N.Am.Tropics <- posterior[["beta[17]"]]
```

```r
S.Am.Tropics <- posterior[["beta[18]"]]

Asia.NM <- posterior[["beta[19]"]]
Euro.NM <- posterior[["beta[20]"]]
N.Am.NM <- Euro #posterior[["beta[21]"]] Euro.NM == Euro

Euro.Arctic <- posterior[["beta[21]"]]
N.Am.Arctic <- posterior[["beta[22]"]]

Africa.Tr<- Africa + Tropics
Africa.S.Mid<- Africa + S.Mid
Africa.N.Mid<- Africa + N.Mid

Asia.Tr<- Africa + Asia + Tropics + Asia.Tropics
Asia.N.Mid<- Africa + Asia + N.Mid + Asia.NM
Asia.Arct<-NA #base + Asia + Arctic no studies

Euro.N.Mid<- Africa + Euro + N.Mid +Euro.NM
Euro.Arct<- Africa + Euro + Arctic + Euro.Arctic


N.Am.Arct<- Africa + N.Am + Arctic + N.Am.Arctic
N.Am.N.Mid<- Africa + N.Am + N.Mid + N.Am.NM
N.Am.Tr<- Africa + N.Am + Tropics + N.Am.Tropics

S.Am.S.Mid<- Africa + S.Am + S.Mid + S.Am.SM
S.Am.Tr<- Africa + S.Am + Tropics + S.Am.Tropics

Aus.NZ.Tr<- Africa + Aus.NZ + Tropics + ANZ.Tropics
Aus.NZ.S.Mid<- Africa + Aus.NZ + S.Mid + ANZ.SM

cats <- c("N. Middle Africa","Tropical Africa", "Middle Africa","Tropical Asia","N. Middle Asia","Arctic Asia","Middle Europe","Arctic Europe","Arctic N. America","Middle N. America","Tropical N. America", "Middle S. America","Tropical S. America","Tropical Australia/New Zealand", "Middle Australia/New Zealand")
beta.cat<-cbind(Africa.N.Mid,Africa.Tr,Africa.S.Mid,Asia.Tr,Asia.N.Mid,Asia.Arct,Euro.N.Mid,Euro.Arct,N.Am.Arct,N.Am.N.Mid,N.Am.Tr,S.Am.S.Mid,S.Am.Tr,Aus.NZ.Tr,Aus.NZ.S.Mid)

#use common median threat
load("2all_interc.rds")
posterior2=as.data.frame(mod)
grand.mean = posterior2[["mu"]]
grand.mean.inv =invlogit(grand.mean)
grand.mean.pred = quantile(grand.mean.inv,probs = c(0.025, 0.5, 0.975))
grand.mean.pred
```

```
##       2.5%        50%      97.5%
## 0.06568248 0.07596676 0.08662260

#calculate median and credible intervals
pred.cat = invlogit(apply(beta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975)
,na.rm=TRUE))
pred.cat.df <- data.frame(x = cats,
                          mean = pred.cat[2,],
                          low = pred.cat[1,],
                          hi = pred.cat[3,])

#arrange in increasing order#
pred.cat.df$cats = with (pred.cat.df, reorder(cats, mean))
#pred.cat.df <- pred.cat.df[pred.cat.df$cats != "Oceans",] #remove Oceans, wh
ich is covered by habitat grouping now

rbeta.cat = invlogit(beta.cat)-invlogit(grand.mean)
rpred.cat = (apply(rbeta.cat, 2, quantile, probs = c(0.025, 0.5, 0.975),na.rm
=TRUE))
rpred.cat.df <- data.frame(x = cats,
                          mean = rpred.cat[2,],
                          low = rpred.cat[1,],
                          hi = rpred.cat[3,])
#Differences from zero
glob.mean.over <- rep("n",nrow(rpred.cat.df))
glob.mean.over[rpred.cat.df$mean > 0 & rpred.cat.df$low > 0] = "y"
glob.mean.over[rpred.cat.df$mean < 0 & rpred.cat.df$hi < 0] = "y"
rpred.cat.df$glob.mean.over <- glob.mean.over

#arrange in increasing order#
rpred.cat.df$cats = with (rpred.cat.df, reorder(cats, mean))

pred.cat.df.r <- pred.cat.df[,1:4]
pred.cat.df.r[,2:4] <- round((pred.cat.df.r[,2:4]), digits = 4)
knitr::kable(pred.cat.df.r, caption = "Table 2: Extinction risk by subregion
and difference from mean, with 95% credible intervals", format = "markdown",
font = 5)
```

Table 3: Extinction risk by subregion and difference from mean, with 95% credible intervals

	x	mean	low	hi
Africa.N.Mid	N. Middle Africa	0.1741	0.0895	0.3117
Africa.Tr	Tropical Africa	0.0777	0.0475	0.1236
Africa.S.Mid	Middle Africa	0.1244	0.0599	0.2351
Asia.Tr	Tropical Asia	0.0653	0.0393	0.1061
Asia.N.Mid	N. Middle Asia	0.0872	0.0505	0.1462
Asia.Arct	Arctic Asia	NA	NA	NA

	x	mean	low	hi
Euro.N.Mid	Middle Europe	0.0367	0.0089	0.1418
Euro.Arct	Arctic Europe	0.0271	0.0149	0.0494
N.Am.Arct	Arctic N. America	0.0462	0.0086	0.2017
N.Am.N.Mid	Middle N. America	0.1112	0.0256	0.3721
N.Am.Tr	Tropical N. America	0.1387	0.0487	0.3387
S.Am.S.Mid	Middle S. America	0.2440	0.0784	0.5592
S.Am.Tr	Tropical S. America	0.1615	0.0970	0.2524
Aus.NZ.Tr	Tropical Australia/New Zealand	0.2040	0.1296	0.3026
Aus.NZ.S.Mid	Middle Australia/New Zealand	0.2692	0.1398	0.4560

```
#calculate global mean
g.mean = grand.mean.pred[2]

#Figures
Fig1a<-ggplot(data = pred.cat.df)+
  geom_vline(xintercept=g.mean) +
  geom_errorbar(aes(y = cats, xmin = low, xmax = hi), width = 0) +
  geom_point(stat = "identity", aes(y = cats, x = mean), color = "#416788", s
ize = 3, shape = 15) +
  xlab("Pre-industrial rise \n in temperature (C)") + xlim(c(0,.6)) +
  theme_classic()+
  theme(axis.title.y = element_blank(),axis.title=element_text(size=14),axis.
text = element_text(size=12))+
  guides(size=F)

## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none
" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

Fig1a

## Warning: Removed 1 row containing missing values or values outside the sca
le range
## (`geom_point()`).
```

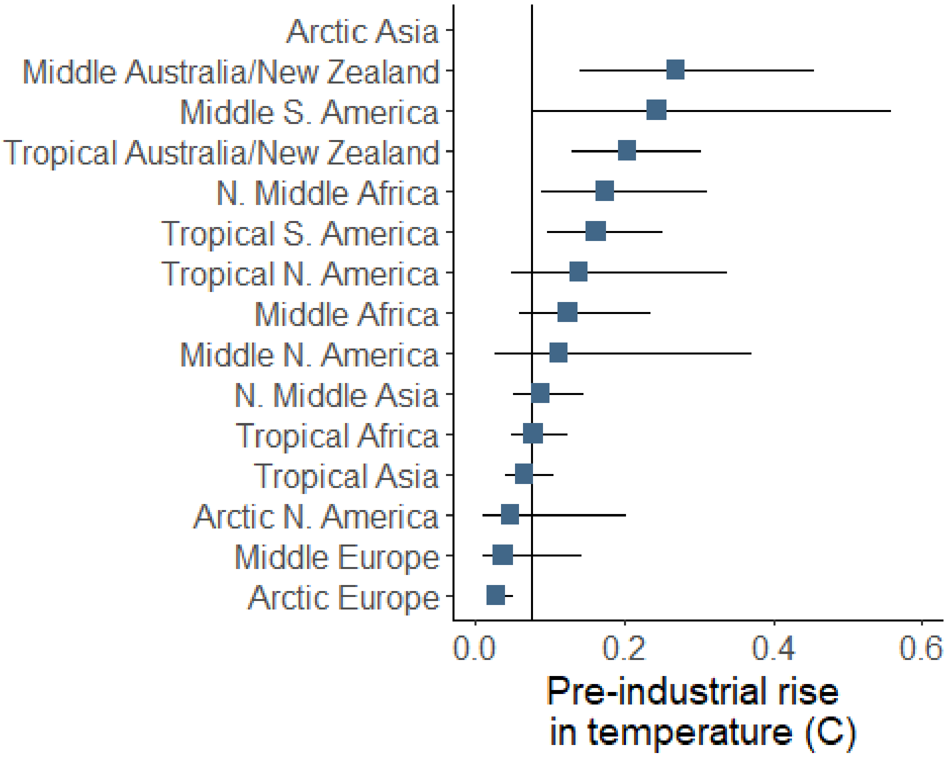


Fig. 1. Predicted extinction risk based on region.

```
Fig1b<-ggplot(data = rpred.cat.df)+
  geom_vline(xintercept=0) +
  geom_errorbar(aes(y = cats, xmin = low, xmax = hi), width = 0) +
  geom_point(stat = "identity", aes(y = cats, x = mean, color = glob.mean.ove
r), size = 3, shape = 15) +
  scale_color_manual(values=c('grey','#E98a15'))+
  xlab("Percent difference \n from global mean") + xlim(c(-.1,.5)) +
  theme_classic()+
  theme(axis.title.y = element_blank(),axis.ticks.y = element_blank(),axis.li
ne.y = element_blank(),axis.text.y = element_blank(),
        axis.title=element_text(size=14),axis.text = element_text(size=12),le
gend.position = "none")+
  guides(size=F)
Fig1b

## Warning: Removed 1 row containing missing values or values outside the sca
le range
## (`geom_point()`).
```

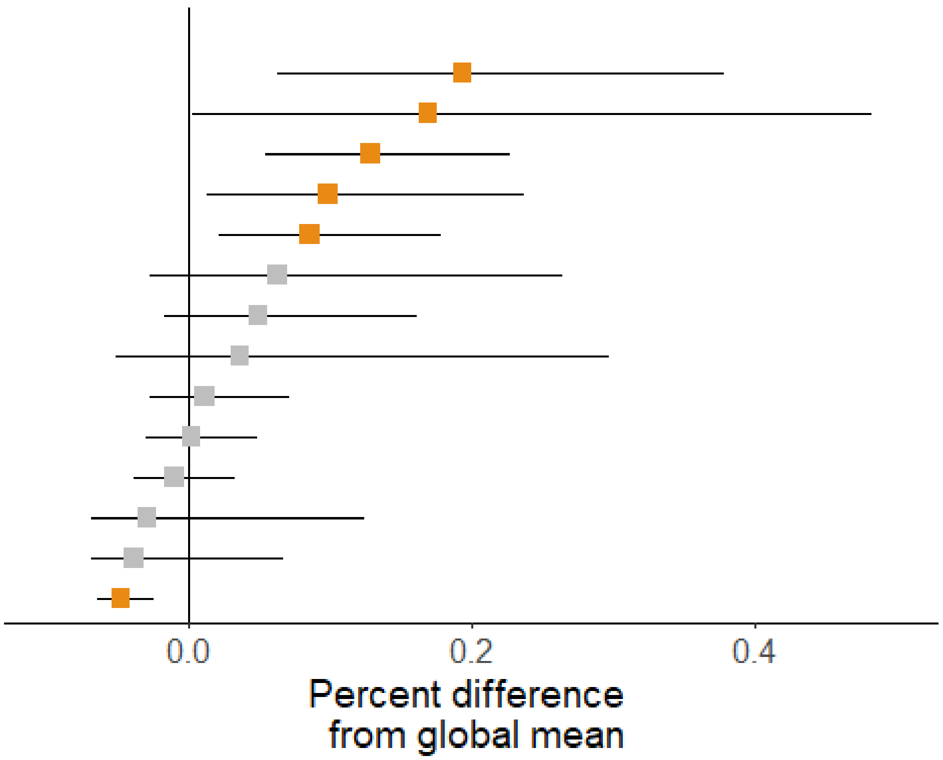


Fig. 2. Predicted extinction risk based on region.

```
ggarrange(Fig1a, NULL, Fig1b, ncol=3, widths = c(4,.4, 2))

## Warning: Removed 1 row containing missing values or values outside the sca
le range
## (`geom_point()`).

## Warning: Removed 1 row containing missing values or values outside the sca
le range
## (`geom_point()`).
```

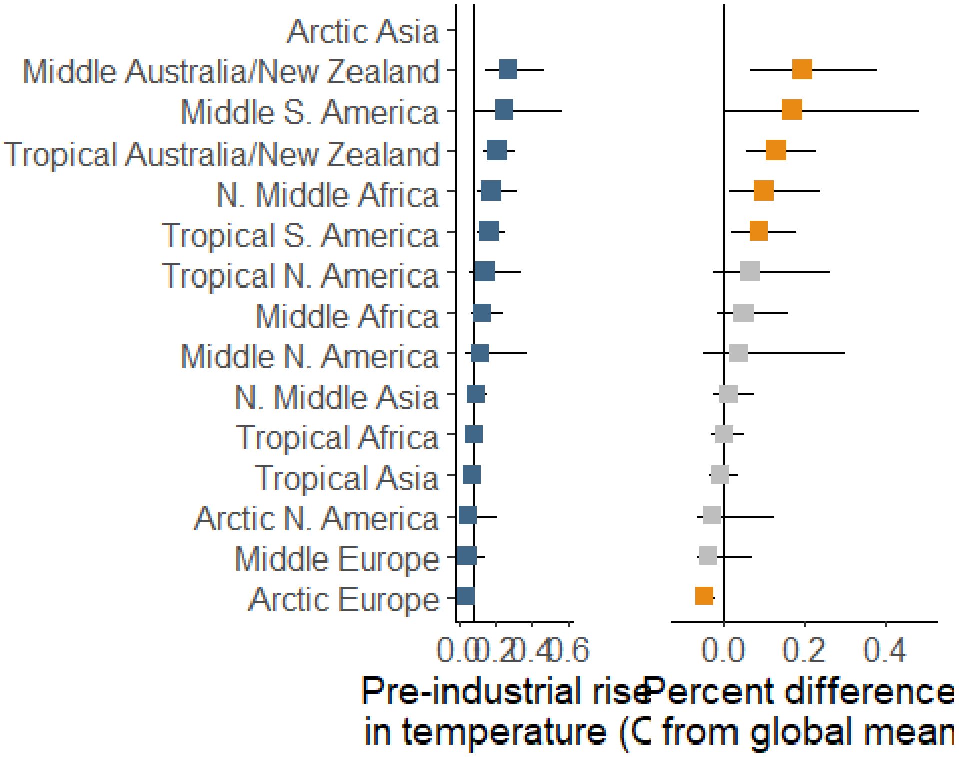


Fig. 3. Predicted extinction risk based on region.

```
ggsave("Metarisk2 region.png",width=6,height=5,unit="in",dpi="print")
```

Note: zero studies from Arctic Asia.

```
cats <- c("N. Middle Africa","Tropical Africa", "Middle Africa","Tropical Asi
a","N. Middle Asia","Arctic Asia","Middle Europe","Arctic Europe","Arctic N.
America","Middle N. America","Tropical N. America", "Middle S. America","Trop
ical S. America","Tropical Australia/New Zealand", "Middle Australia/New Zeal
and")
#catalog sample sizes
N.0.st<-length(unique(dataP$Study[dataP$Region == "Africa" & dataP$N.Middle==
"Y"]))
N.0.mod<-length(dataP$Study[dataP$Region == "Africa" & dataP$N.Middle == "Y"]
)

N.1.st<-length(unique(dataP$Study[dataP$Region == "Africa" & dataP$Tropics ==
"Y"]))
N.1.mod<-length(dataP$Study[dataP$Region == "Africa" & dataP$Tropics == "Y"])

N.2.st<-length(unique(dataP$Study[dataP$Region == "Africa" & dataP$S.Middle =
= "Y"]))
N.2.mod<-length(dataP$Study[dataP$Region == "Africa" & dataP$S.Middle == "Y"]
)

N.3.st<-length(unique(dataP$Study[dataP$Region == "Asia" & dataP$Tropics == "
Y"]))
N.3.mod<-length(dataP$Study[dataP$Region == "Asia" & dataP$Tropics == "Y"])

N.4.st<-length(unique(dataP$Study[dataP$Region == "Asia" & dataP$N.Middle ==
"Y"]))
N.4.mod<-length(dataP$Study[dataP$Region == "Asia" & dataP$N.Middle == "Y"])

N.6.st<-length(unique(dataP$Study[dataP$Region == "Asia" & dataP$Arctic == "Y
"]))
N.6.mod<-length(dataP$Study[dataP$Region == "Asia" & dataP$Arctic == "Y"])

N.7.st<-length(unique(dataP$Study[dataP$Region == "Europe" & dataP$N.Middle =
= "Y"]))
N.7.mod<-length(dataP$Study[dataP$Region == "Europe" & dataP$N.Middle == "Y"]
)

N.8.st<-length(unique(dataP$Study[dataP$Region == "Europe" & dataP$Arctic ==
"Y"]))
N.8.mod<-length(dataP$Study[dataP$Region == "Europe" & dataP$Arctic == "Y"])

N.9.st<-length(unique(dataP$Study[dataP$Region == "Namerica" & dataP$Arctic =
= "Y"]))
N.9.mod<-length(dataP$Study[dataP$Region == "Namerica" & dataP$Arctic == "Y"]
)

N.10.st<-length(unique(dataP$Study[dataP$Region == "Namerica" & dataP$N.Middl
e == "Y"]))
N.10.mod<-length(dataP$Study[dataP$Region == "Namerica" & dataP$N.Middle == "
Y"])

N.11.st<-length(unique(dataP$Study[dataP$Region == "Namerica" & dataP$Tropics
== "Y"]))
N.11.mod<-length(dataP$Study[dataP$Region == "Namerica" & dataP$Tropics == "Y
"])

N.12.st<-length(unique(dataP$Study[dataP$Region == "Samerica" & dataP$S.Middl
e == "Y"]))
N.12.mod<-length(dataP$Study[dataP$Region == "Samerica" & dataP$S.Middle == "
Y"])

N.13.st<-length(unique(dataP$Study[dataP$Region == "Samerica" & dataP$Tropics
== "Y"]))
N.13.mod<-length(dataP$Study[dataP$Region == "Samerica" & dataP$Tropics == "Y
"])

N.14.st<-length(unique(dataP$Study[dataP$Region == "AustraliaNewZealand" & da
taP$Tropics == "Y"]))
N.14.mod<-length(dataP$Study[dataP$Region == "AustraliaNewZealand" & dataP$Tr
opics == "Y"])
```

```
N.15.st<-length(unique(dataP$Study[dataP$Region == "AustraliaNewZealand" & da
taP$S.Middle == "Y"]))
N.15.mod<-length(dataP$Study[dataP$Region == "AustraliaNewZealand" & dataP$S.
Middle == "Y"])


table.data<-data.frame(
  Factor = cats,
  Studies = c(N.0.st,N.1.st,N.2.st,N.3.st,N.4.st,N.6.st,N.7.st,N.8.st,N.9.st,
N.10.st,N.11.st,N.12.st,N.13.st,N.14.st,N.15.st),
  Models = c(N.0.mod,N.1.mod,N.2.mod,N.3.mod,N.4.mod,N.6.mod,N.7.mod,N.8.mod,
N.9.mod,N.10.mod,N.11.mod,N.12.mod,N.13.mod,N.14.mod,N.15.mod)
)
knitr::kable(table.data, caption = "Table 3: Number of studies and models for
each factor", format = "markdown")
```

Table 4: Number of studies and models for each factor

Factor	Studies	Models
N. Middle Africa	6	49
Tropical Africa	40	274
Middle Africa	35	132
Tropical Asia	38	265
N. Middle Asia	56	273
Arctic Asia	0	0
Middle Europe	109	711
Arctic Europe	44	278
Arctic N. America	12	107
Middle N. America	81	552
Tropical N. America	22	127
Middle S. America	34	158
Tropical S. America	58	291
Tropical Australia/New Zealand	32	241
Middle Australia/New Zealand	37	199

Create Fig.2

Pick map colors

```
#Create color pallete
library(colorspace)
#hcl_palettes(plot = TRUE)
ext.risk = seq(0.0,0.40,0.005)
#choose_palette()
```

```r
colors <- sequential_hcl(length(ext.risk), palette = "Lajolla")
map.colors <- data.frame("Ext.risk" = ext.risk,
                         "colors" = colors)
color.ramp <- (map.colors$colors[c(1,11,21,31,41,51)])
#findest nearest color
find_nearest_id <- function(id) {
  nearest_index <- which.min(abs(map.colors$Ext.risk - id))
  return(map.colors$Ext.risk[nearest_index])
}

#find nearest ext risk value
pred.cat.df$color.id <- as.numeric(sapply(pred.cat.df$mean, find_nearest_id))
pred.cat.df$color.id[pred.cat.df$x == "Arctic Asia"] = "NA"

#merge based on nearest value
pred.cat.df.merge <- merge(pred.cat.df, map.colors, by.x = "color.id", by.y =
"Ext.risk")

col.table<-data.frame(
  Factor = pred.cat.df.merge$cats,
  Colors = pred.cat.df.merge$colors
)
knitr::kable(col.table, caption = "Table 4: Colors for each region", format =
"markdown",color = "white")
```

Table 5: Colors for each region

Factor	Colors
Arctic Europe	#F5F2B0
Middle Europe	#F3EBA7
Arctic N. America	#F1E59D
Tropical Asia	#EDD889
Tropical Africa	#EBCE7B
N. Middle Asia	#EACB76
Middle N. America	#E6BA5D
Middle Africa	#E4AF4E
Tropical N. America	#E2A43F
Tropical S. America	#DE9529
N. Middle Africa	#DC8916
Tropical Australia/New Zealand	#D47100
Middle S. America	#BB5735
Middle Australia/New Zealand	#A9483E

```r
# Download world shapefile
#
coast <- ne_coastline(scale = 110, returnclass = "sf")
```

```r
world <- ne_download(scale = 10,category = "physical", type = "geography_regi
ons_polys", returnclass = "sf") #category = "cultural", type = "countries", t
ype = "land",
```

```
## Reading layer `ne_10m_geography_regions_polys' from data source
##   `C:\Users\mcu08001\AppData\Local\Temp\RtmpaujqfO\ne_10m_geography_region
s_polys.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1047 features and 37 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -180 ymin: -89.99993 xmax: 180 ymax: 83.97999
## Geodetic CRS:  WGS 84
```

```r
africa <- world[world$NAME == "AFRICA" | world$NAME == "MADAGASCAR",]
europe <- world[world$NAME == "EUROPE" | world$NAME == "ICELAND"| world$NAME
== "IRELAND" | world$NAME == "GREAT BRITAIN",]
asia <- world[world$NAME == "ASIA" | world$NAME == "SRI LANKA"| world$NAME ==
"BORNEO"| world$NAME == "SUMATRA"| world$NAME == "NEW GUINEA",]
# #asia <- world[world$REGION == "ASIA"]
N.Am <- world[world$NAME == "NORTH AMERICA" | world$NAME == "CENTRAL AMERICA"
,]
S.Am <- world[world$NAME == "SOUTH AMERICA",]
#New.Aus <- world[world$NAME == "AUSTRALIA" | world$NAME == "NEW ZEALAND",] #
Include Russia in Asia
New.Aus <- ne_countries(scale = 110, country = c("Australia", "New Zealand"),
returnclass = "sf")
Ant <- world[world$NAME == "ANTARCTICA",]

asia2 <- ne_countries(scale = 110, continent = "Asia", returnclass = "sf")
euro2 <- ne_countries(scale = 110, continent = "Europe", returnclass = "sf")
NM2 <- ne_countries(scale = 110, continent = "North America", returnclass = "
sf")

# Define latitudinal bounds
tropic.box <- c(-180, -23.44, 180, 23.44)  # Adjust these values based on you
r desired bounding box
N.mid.box <- c(-180, 23.44, 180, 66.55)  # Adjust these values based on your
desired bounding box
Arc.box <- c(-180, 66.55, 180, 90)  # Adjust these values based on your desir
ed bounding box
S.mid.box <- c(-180, -23.44, 180, -66.55)  # Adjust these values based on you
r desired bounding box
AntArc.box <- c(-180, -66.55, 180, -90)  # Adjust these values based on your
desired bounding box

# Crop to the bounding box
sf::sf_use_s2(FALSE)
```

```
## Spherical geometry (s2) switched off
```

```
#Crop out lat/continent polygons
africa.tropic <- st_crop(africa, xmin = tropic.box[1], ymin = tropic.box[2],
xmax = tropic.box[3], ymax = tropic.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

africa.S.mid <- st_crop(africa, xmin = S.mid.box[1], ymin = S.mid.box[2], xma
x = S.mid.box[3], ymax = S.mid.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

africa.N.mid <- st_crop(africa, xmin = N.mid.box[1], ymin = N.mid.box[2], xma
x = N.mid.box[3], ymax = N.mid.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

euro.N.mid <- st_crop(europe, xmin = N.mid.box[1], ymin = N.mid.box[2], xmax
= N.mid.box[3], ymax = N.mid.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

euro.Arc <- st_crop(europe, xmin = Arc.box[1], ymin = Arc.box[2], xmax = Arc.
box[3], ymax = Arc.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

euro.Arc2 <- st_crop(euro2, xmin = Arc.box[1], ymin = Arc.box[2], xmax = Arc.
box[3], ymax = Arc.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

asia.tropic <- st_crop(asia, xmin = tropic.box[1], ymin = tropic.box[2], xmax
= tropic.box[3], ymax = tropic.box[4])

## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar

asia.N.mid <- st_crop(asia, xmin = N.mid.box[1], ymin = N.mid.box[2], xmax =
N.mid.box[3], ymax = N.mid.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
asia.Arc <- st_crop(asia, xmin = Arc.box[1], ymin = Arc.box[2], xmax = Arc.bo
x[3], ymax = Arc.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
asia2.tropic <- st_crop(asia2, xmin = tropic.box[1], ymin = tropic.box[2], xm
ax = tropic.box[3], ymax = tropic.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
asia2.N.mid <- st_crop(asia2, xmin = N.mid.box[1], ymin = N.mid.box[2], xmax
= N.mid.box[3], ymax = N.mid.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
asia2.Arc <- st_crop(asia2, xmin = Arc.box[1], ymin = Arc.box[2], xmax = Arc.
box[3], ymax = Arc.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
N.Am.tropic <- st_crop(N.Am, xmin = tropic.box[1], ymin = tropic.box[2], xmax
= tropic.box[3], ymax = tropic.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
N.Am.tropic2 <- st_crop(NM2, xmin = tropic.box[1], ymin = tropic.box[2], xmax
= tropic.box[3], ymax = tropic.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
N.Am.Arc <- st_crop(N.Am, xmin = Arc.box[1], ymin = Arc.box[2], xmax = Arc.bo
x[3], ymax = Arc.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
N.Am.N.mid <- st_crop(N.Am, xmin = N.mid.box[1], ymin = N.mid.box[2], xmax =
N.mid.box[3], ymax = N.mid.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
S.Am.tropic <- st_crop(S.Am, xmin = tropic.box[1], ymin = tropic.box[2], xmax
= tropic.box[3], ymax = tropic.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
S.Am.S.mid <- st_crop(S.Am, xmin = S.mid.box[1], ymin = S.mid.box[2], xmax =
S.mid.box[3], ymax = S.mid.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
New.Aus.tropic <- st_crop(New.Aus, xmin = tropic.box[1], ymin = tropic.box[2]
, xmax = tropic.box[3], ymax = tropic.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

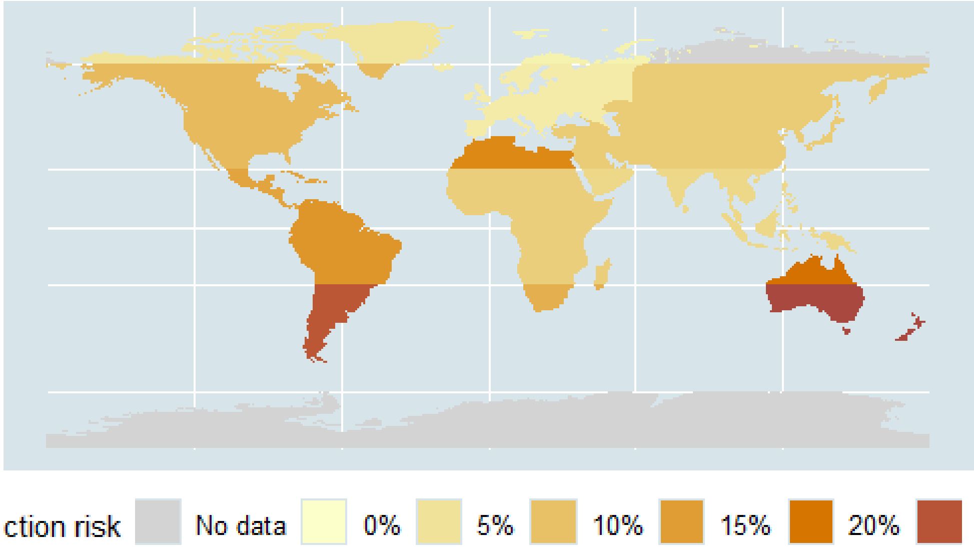
```
New.Aus.S.mid <- st_crop(New.Aus, xmin = S.mid.box[1], ymin = S.mid.box[2], x
max = S.mid.box[3], ymax = S.mid.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
Ant.ant <- st_crop(Ant, xmin = AntArc.box[1], ymin = AntArc.box[2], xmax = An
tArc.box[3], ymax = AntArc.box[4])
```

```
## although coordinates are longitude/latitude, st_intersection assumes that
they
## are planar
```

```
# Display the cropped map using ggplot2
Fig2 <- ggplot() +
    geom_sf(data = N.Am.tropic, aes(fill = color.ramp[6]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = color.ramp[5]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = color.ramp[4]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = color.ramp[3]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = color.ramp[2]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = color.ramp[1]), color = NA) +
  geom_sf(data = N.Am.tropic, aes(fill = "lightgray"), color = NA) +
  geom_sf(data = N.Am.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.mer
ge$cats == "Tropical N. America"], color = "NA") +
  geom_sf(data = N.Am.tropic2, fill = pred.cat.df.merge$colors[pred.cat.df.me
rge$cats == "Tropical N. America"], color = "NA") +
  geom_sf(data = euro.Arc2, fill = pred.cat.df.merge$colors[pred.cat.df.merge
$cats == "Arctic Europe"], color = "NA") +
  geom_sf(data = euro.Arc, fill = pred.cat.df.merge$colors[pred.cat.df.merge$
cats == "Arctic Europe"], color = "NA") +
  geom_sf(data = euro.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.merg
e$cats == "Middle Europe"], color = "NA") +
  geom_sf(data = asia.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.merg
e$cats == "N. Middle Asia"], color = "NA") +
  geom_sf(data = africa.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.m
erge$cats == "Tropical Africa"], color = "NA") +
  geom_sf(data = africa.S.mid, fill = pred.cat.df.merge$colors[pred.cat.df.me
rge$cats == "Middle Africa"], color = "NA") +
  geom_sf(data = africa.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.me
rge$cats == "N. Middle Africa"], color = "NA") +
  geom_sf(data = asia.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.mer
ge$cats == "Tropical Asia"], color = "NA") +
  geom_sf(data = asia2.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.me
rge$cats == "Tropical Asia"], color = "NA") +
  geom_sf(data = asia.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.merg
e$cats == "N. Middle Asia"], color = "NA") +
  geom_sf(data = asia2.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.mer
ge$cats == "N. Middle Asia"], color = "NA") +
  geom_sf(data = N.Am.Arc, fill = pred.cat.df.merge$colors[pred.cat.df.merge$
cats == "Arctic N. America"], color = "NA") +
  geom_sf(data = N.Am.N.mid, fill = pred.cat.df.merge$colors[pred.cat.df.merg
e$cats == "Middle N. America"], color = "NA") +
  geom_sf(data = S.Am.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.mer
ge$cats == "Tropical S. America"], color = "NA") +
  geom_sf(data = S.Am.S.mid, fill = pred.cat.df.merge$colors[pred.cat.df.merg
e$cats == "Middle S. America"], color = "NA") +
  geom_sf(data = New.Aus.tropic, fill = pred.cat.df.merge$colors[pred.cat.df.
merge$cats == "Tropical Australia/New Zealand"], color = "NA") +
  geom_sf(data = New.Aus.S.mid, fill = pred.cat.df.merge$colors[pred.cat.df.m
erge$cats == "Middle Australia/New Zealand"], color = "NA") +
  geom_sf(data = asia.Arc, fill = "lightgray", color = "NA")+
  geom_sf(data = Ant.ant, fill = "lightgray", color = "NA") +
  #geom_sf(data = coast, color = "darkgray") +
  ylim(-80,85) + scale_y_continuous(breaks = c(-66.55, -23.44, 0, 23.44, 66.5
5)) +
  scale_fill_identity(guide = "legend",name ="Exinction risk",labels =c("25%"
,"20%","15%","10%","5%","0%","No data")) +
  theme(panel.background = element_rect(fill = "#D7E5EA"),legend.position = "
bottom",legend.text = element_text(size=10),legend.margin=margin(),legend.dir
ection = "horizontal",plot.margin = margin(t = 0,  # Top margin
                             r = 0,  # Right margin
                             b = 0,  # Bottom margin
                             l = 0)) + # Left margin) + ) +
  guides(fill = guide_legend(nrow = 1, reverse = TRUE)) #b7d1da

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.

Fig2
```



```
#ggsave("Fig2xb overall.png",width=8,height=6,unit="in",dpi=3600)
#ggsave("Fig2xc overall.eps",width=8,height=6,unit="in",dpi=2400)
```

Variation explained

```
#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
load("2geog5.rds")
posterior=as.data.frame(mod);
betamat <- (model.matrix(~Region,data=data.use, contrasts.arg=list(Region="co
ntr.sum")))

#posterior=as.data.frame(mod.int);
#betamat = as.matrix(rep(1,9000))

#Variables and matrices
S = 9000; #samples
K = ncol(betamat); #factors
p.mat <- as.matrix(posterior[,1:K])
```

```
y = dataP$percent2
y.mat = t(matrix(rep(y,S), nrow = N, ncol = S))
y.mean <- mean(y)

#Calculate y.pred for fixed effects only
y.pred <- matrix(rep(NA, N*S), nrow = S, ncol = N)
theta <- y.pred
for (i in 1:N) {
      theta[,i] = invlogit(p.mat %*% betamat[i,])#rows = samples, cols = i
      y.pred[,i] = (theta[,i] * data.use$Total.N[i])/(theta[,i] * data.use$To
tal.N[i] + (1-theta[,i]) * data.use$Total.N[i])
}

#Calcluate residual variance
res.f = y.mat - y.pred
RSS.f = rowSums((res.f)^2)
res.v.f = 1/(N -1) * RSS.f

#Calculate fit variance
pred.v.f = 1/(N-1) * rowSums((y.pred)^2)

#Calculate R2
R2.v.f = pred.v.f/(pred.v.f + res.v.f)
cat("fixed effects R2 = ", quantile(R2.v.f,probs = c(0.025, 0.5, 0.975),na.rm
= T))

## fixed effects R2 =  0.07863751 0.1053038 0.1374897

#Total model With random effects
y.pred.c <-(as.matrix(posterior[,(K+1):(N+K)])) #calculated in STAN, with all
RE and weightings

#Calculate residual variance
res.c = y.mat - y.pred.c
RSS.c = rowSums((res.c)^2)
res.v.c = 1/(N-1) * RSS.c

#Calculate fit variance

pred.v.c = 1/(N-1) * rowSums(y.pred.c^2)

#Calculate full model R2
R2.v.c = pred.v.c/(pred.v.c + res.v.c)
print("                                                   ")

## [1] "                                                   "

cat("Overall model R2 = ", quantile(R2.v.c,probs = c(0.025, 0.5, 0.975),na.rm
= T))
```

```
## Overall model R2 =  0.76415 0.7845941 0.8047536
```

```
#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
load("2all_lat.rds")
posterior=as.data.frame(mod);
betamat <- (model.matrix(~Antarctic + S.Middle + Tropics + N.Middle + Arctic,
data=data.use,
                          contrasts.arg=list(Antarctic = "contr.sum", S.Middl
e = "contr.sum", Tropics="contr.sum", N.Middle = "contr.sum", Arctic = "contr
.sum")))

#posterior=as.data.frame(mod.int);
#betamat = as.matrix(rep(1,9000))

#Variables and matrices
S = 9000; #samples
K = ncol(betamat); #factors
p.mat <- as.matrix(posterior[,1:K])
y = dataP$percent2
y.mat = t(matrix(rep(y,S), nrow = N, ncol = S))
y.mean <- mean(y)

#Calculate y.pred for fixed effects only
y.pred <- matrix(rep(NA, N*S), nrow = S, ncol = N)
theta <- y.pred
for (i in 1:N) {
      theta[,i] = invlogit(p.mat %*% betamat[i,])#rows = samples, cols = i
      y.pred[,i] = (theta[,i] * data.use$Total.N[i])/(theta[,i] * data.use$To
tal.N[i] + (1-theta[,i]) * data.use$Total.N[i])
}

#Calcluate residual variance
res.f = y.mat - y.pred
RSS.f = rowSums((res.f)^2)
res.v.f = 1/(N -1) * RSS.f

#Calculate fit variance
pred.v.f = 1/(N-1) * rowSums((y.pred)^2)

#Calculate R2
R2.v.f = pred.v.f/(pred.v.f + res.v.f)
cat("fixed effects R2 = ", quantile(R2.v.f,probs = c(0.025, 0.5, 0.975),na.rm
= T))
```

```
## fixed effects R2 =  0.07312864 0.09758071 0.1286844
```

```
#Total model With random effects
y.pred.c <-(as.matrix(posterior[,(K+1):(N+K)])) #calculated in STAN, with all
```

```
RE and weightings

#Calculate residual variance
res.c = y.mat - y.pred.c
RSS.c = rowSums((res.c)^2)
res.v.c = 1/(N-1) * RSS.c

#Calculate fit variance

pred.v.c = 1/(N-1) * rowSums(y.pred.c^2)

#Calculate full model R2
R2.v.c = pred.v.c/(pred.v.c + res.v.c)
print("                                                   ")

## [1] "                                                   "

cat("Overall model R2 = ", quantile(R2.v.c,probs = c(0.025, 0.5, 0.975),na.rm
= T))

## Overall model R2 =  0.7655525 0.7864663 0.8063632

#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
#load("2geo+lat.rds")
load("3geoxlat3.rds")
posterior=as.data.frame(mod);
#betamat <- model.matrix(~Region + Antarctic + S.Middle + Tropics + N.Middle
+ Arctic, data=data.use)
betamat.0 <- model.matrix(~Region * (Antarctic + S.Middle + Tropics + N.Middl
e + Arctic), data=data.use) #relative to global mean
betamat <- betamat.0[,c(1:7,9:13,22,26,28,29,32,33,35,39,44,46)]

#posterior=as.data.frame(mod.int);
#betamat = as.matrix(rep(1,9000))

#Variables and matrices
S = 9000; #samples
K = ncol(betamat); #factors
p.mat <- as.matrix(posterior[,1:K])
y = dataP$percent2
y.mat = t(matrix(rep(y,S), nrow = N, ncol = S))
y.mean <- mean(y)

#Calculate y.pred for fixed effects only
y.pred <- matrix(rep(NA, N*S), nrow = S, ncol = N)
theta <- y.pred
for (i in 1:N) {
      theta[,i] = invlogit(p.mat %*% betamat[i,])#rows = samples, cols = i
      y.pred[,i] = (theta[,i] * data.use$Total.N[i])/(theta[,i] * data.use$To
tal.N[i] + (1-theta[,i]) * data.use$Total.N[i])
}

#Calcluate residual variance
res.f = y.mat - y.pred
RSS.f = rowSums((res.f)^2)
res.v.f = 1/(N -1) * RSS.f

#Calculate fit variance
pred.v.f = 1/(N-1) * rowSums((y.pred)^2)

#Calculate R2
R2.v.f = pred.v.f/(pred.v.f + res.v.f)
cat("fixed effects R2 = ", quantile(R2.v.f,probs = c(0.025, 0.5, 0.975),na.rm
= T))
```

```
## fixed effects R2 =  0.1069424 0.1454799 0.1954363
```

```
#Total model With random effects
y.pred.c <-(as.matrix(posterior[,(K+1):(N+K)])) #calculated in STAN, with all
RE and weightings

#Calculate residual variance
res.c = y.mat - y.pred.c
RSS.c = rowSums((res.c)^2)
res.v.c = 1/(N-1) * RSS.c

#Calculate fit variance

pred.v.c = 1/(N-1) * rowSums(y.pred.c^2)

#Calculate full model R2
R2.v.c = pred.v.c/(pred.v.c + res.v.c)
print("                                                   ")
```

```
## [1] "                                                   "
```

```
cat("Overall model R2 = ", quantile(R2.v.c,probs = c(0.025, 0.5, 0.975),na.rm
= T))
```

```
## Overall model R2 =  0.7648199 0.7852879 0.8057351
```