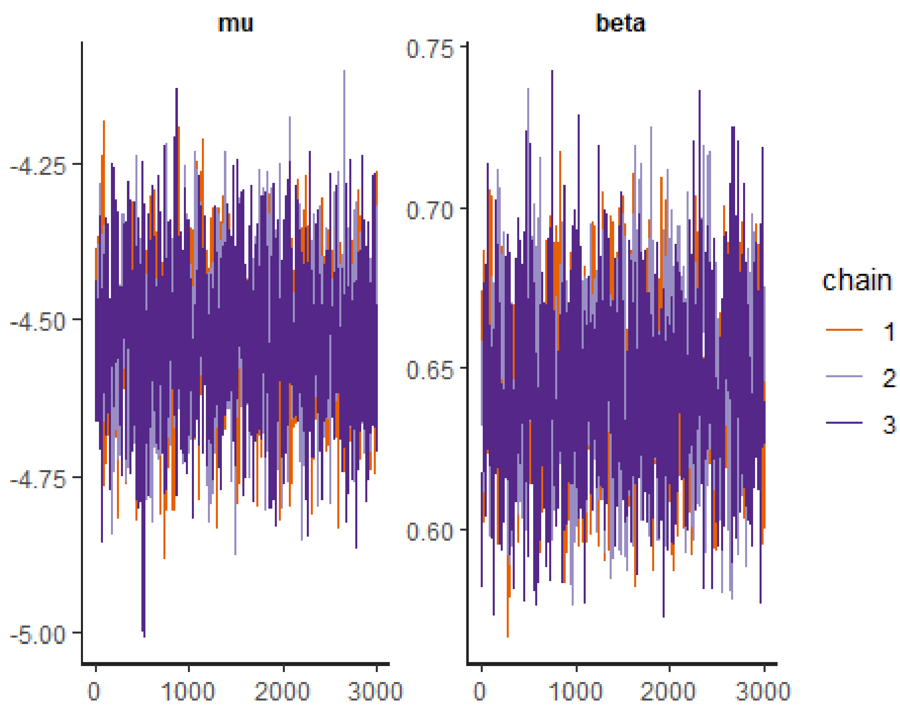
Overall model of extinction risk from climate change

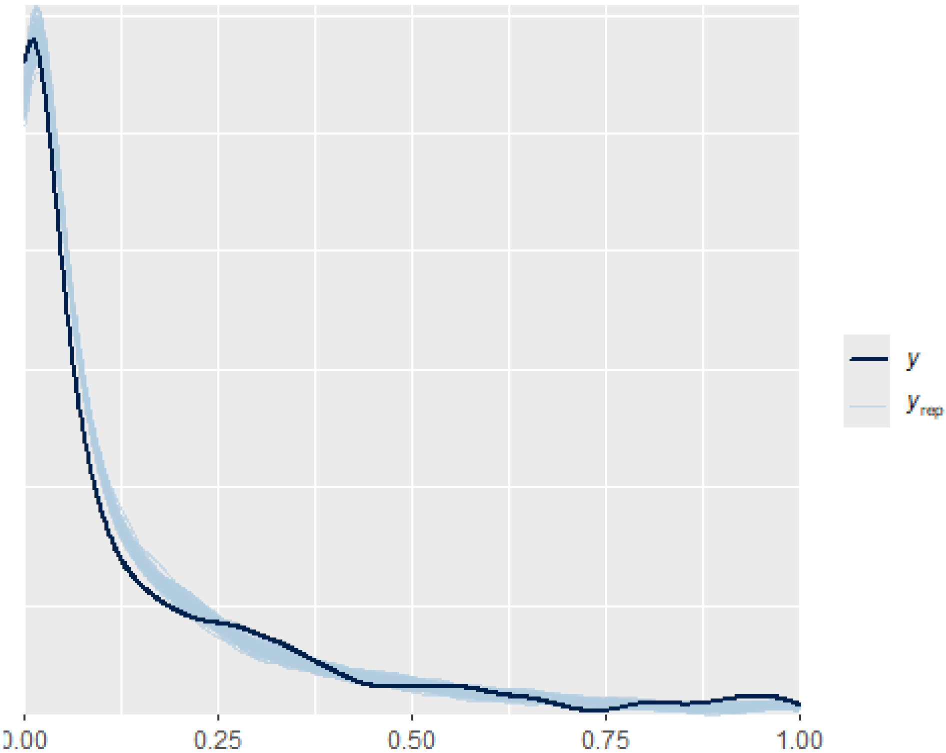
Jan. 5, 2024

```
Includes overall predictions plus tests of methods and assumptions

#Setup

knitr::opts_chunk$set(echo = TRUE, cache.lazy = FALSE)
rm(list = ls())
 root.dir = "C:/Users/mcu08001/Documents/1New Research/CC MetaRisk2/Analysis"
library(MCMCglmm); library(coda); library(ggplot2); library(rstan);
library(bayesplot); library(shinystan); library(loo); library(rstanarm);

options(mc.cores = parallel::detectCores())
rstan_options(auto_write = FALSE)
dataP<-read.table("Metarisk2 aggthres 5.txt",header=T); #newest data with
slight fix to dispersal data

"number of unique studies"
length(unique(dataP$Study))

dataP2<-dataP[is.finite(dataP$Pre.Ind.Rise),]; attach(dataP2) # need to
eliminate NA s for pre-industrial rise or stat programs crash

#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;
dataP2$percent2 <- percent2;

data.use<-dataP2

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
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise
min(Pre.Ind.Rise)= 0.4

stan.data<-list(N = N, percent = data.use$percent2, Ind =
data.use$Pre.Ind.Rise, phi = phi, S = n.Study, Study = Studyint)
params.to.monitor=c("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")
params.to.monitor2=c("mu","beta")#
```

```r
# mod=stan(file="MetaRisk2 RSTAN betareg
2b.stan",data=stan.data,pars=params.to.monitor,
#           chains = 3, warmup=7000, cores=7,iter=10000,
#           control=list(adapt_delta = 0.9, max_treedepth = 15))

load("2pre_lowb.rds") #mu prior (-50,1)
modx = mod

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
Rhat
## mu   -4.5273890 0.003521920 0.10455352 -4.7373329 -4.3237051 881.289
1.004510
## beta  0.6457854 0.000897318 0.02389391  0.5995963  0.6929307 709.058
1.003601

#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'
)
```

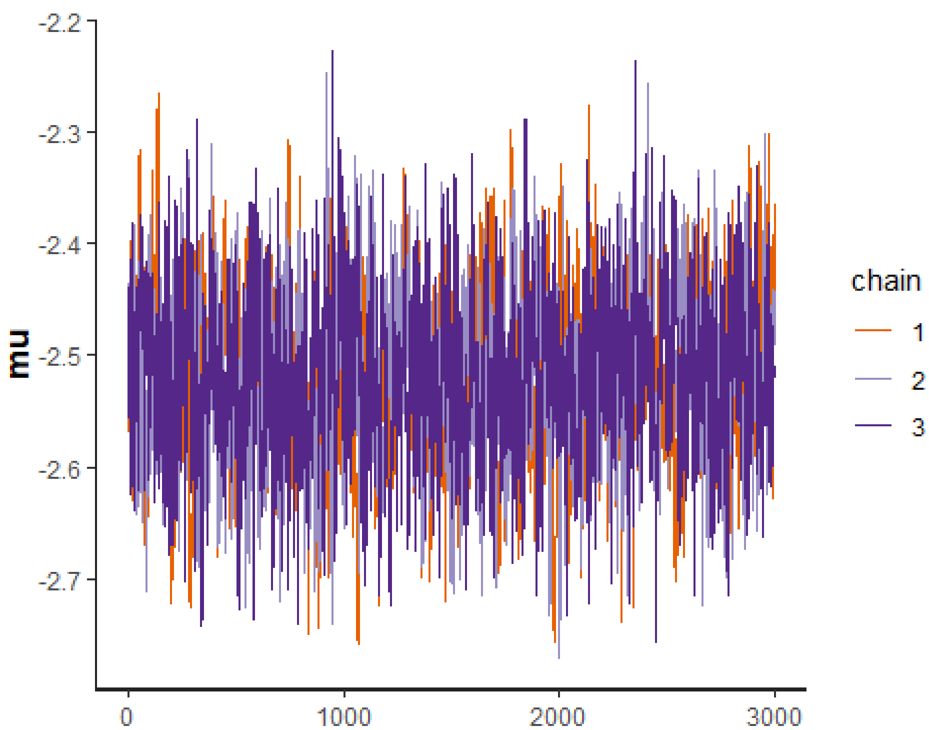



```
#ggsave("Fig S1b koffset 001.png",width=8,height=5.5,unit="in",dpi="print")
#offset = 0.001
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   7720.4  95.5
## p_loo      1849.4  24.6
## looic    -15440.8 191.0
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      811  25.2%   257
##  (0.5, 0.7]   (ok)        741  23.0%   144
##    (0.7, 1]   (bad)      1427  44.3%   14
##    (1, Inf)   (very bad)  241   7.5%   3
## See help('pareto-k-diagnostic') for details.
```

Intercept-only model

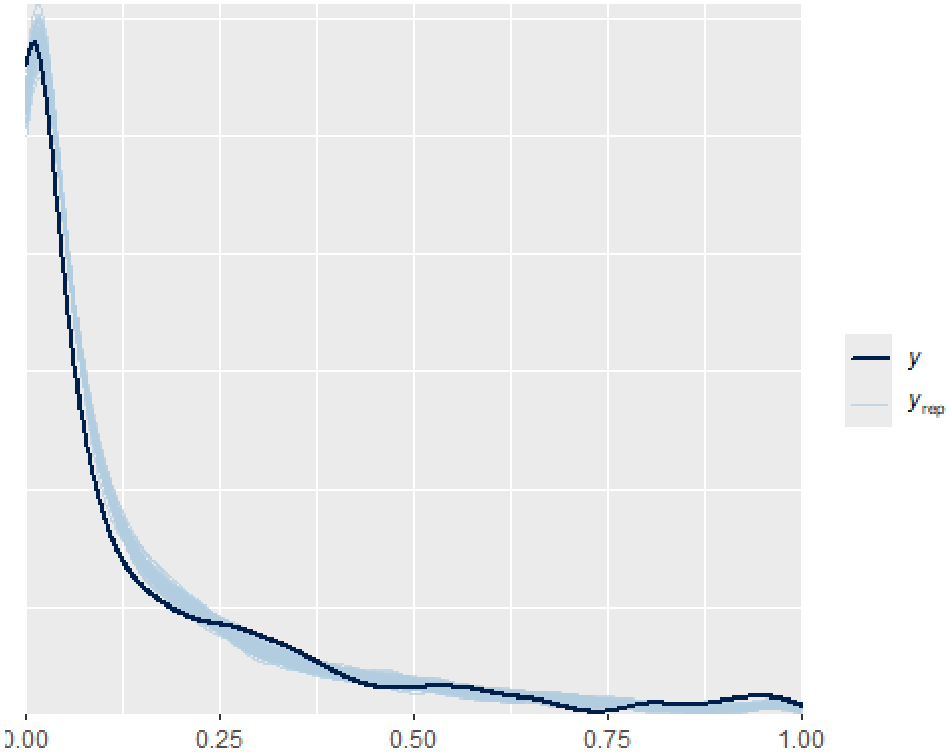
Subset of studies with pre-industrial data, for comparison against the pre-industrial temperature rise model.

```
#Intercept only model, for subset of studies with pre-ind temp data
# mod=stan(file="MetaRisk2 RSTAN int only 
1.stan",data=stan.data,pars=params.to.monitor,
#           chains = 3, warmup=5000, cores=3,iter=8000, save_warmup = FALSE,
#           init = init.fn, control=list(adapt_delta = 0.9, max_treedepth = 
15))

load("2sub_interc.rds")
params.to.monitor2=c("mu")#

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     Rhat
## mu -2.517816 0.002245067 0.0759056 -2.669354 -2.372423 1143.113 1.001435

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



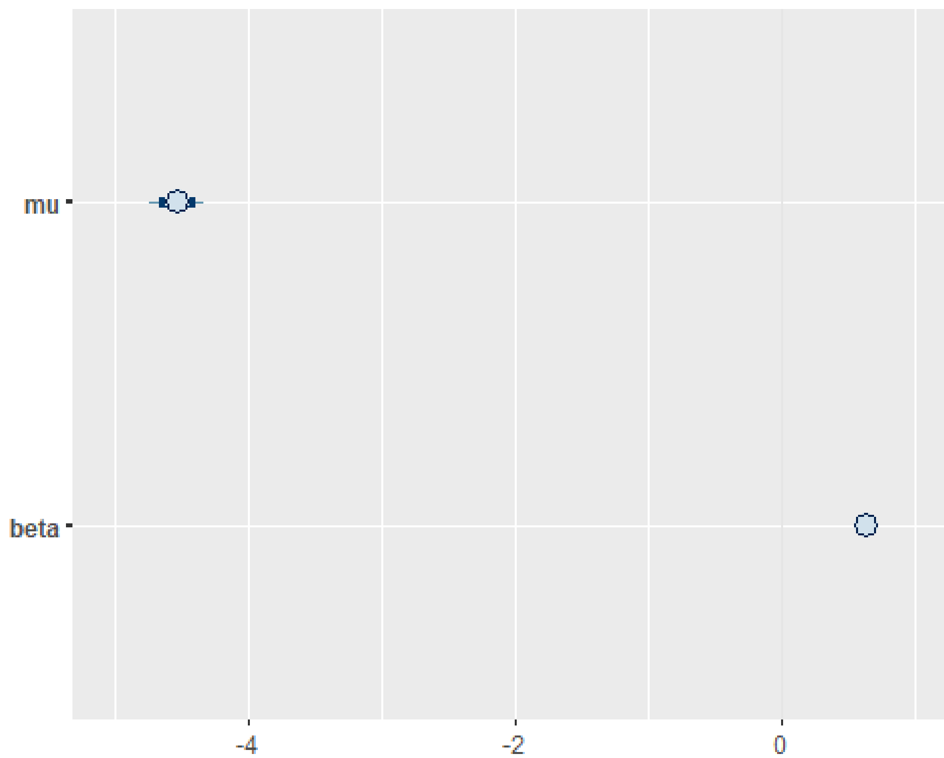

```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
```

```
  fun = 'dens_overlay'
)
```



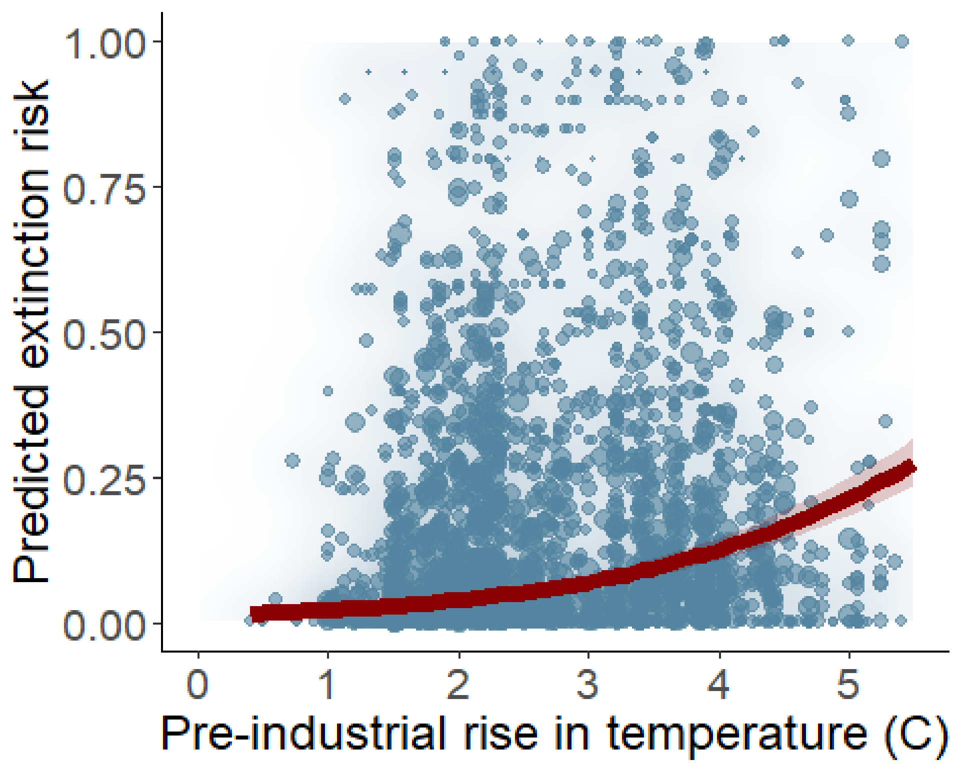

```
#ggsave("Fig S1b koffset 001.png",width=8,height=5.5,unit="in",dpi="print")
#offset = 0.001
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3220 log-likelihood matrix
##
##           Estimate    SE
## elpd_loo    7575.9  96.5
## p_loo       1987.0  25.5
## looic     -15151.7 193.0
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      572  17.8%   348
##  (0.5, 0.7]   (ok)        825  25.6%   106
##    (0.7, 1]   (bad)      1541  47.9%   12
```

```
##    (1, Inf)   (very bad)  282   8.8%   3
## See help('pareto-k-diagnostic') for details.

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line = pred.reg.quant[2,],
                          low_line = pred.reg.quant[1,],
                          hi_line = pred.reg.quant[3,])

#parameter plot with intervals
params.to.show=c("mu","beta")
mcmc_intervals(posterior,prob=.8,prob_outer = .95,pars=params.to.show)
```



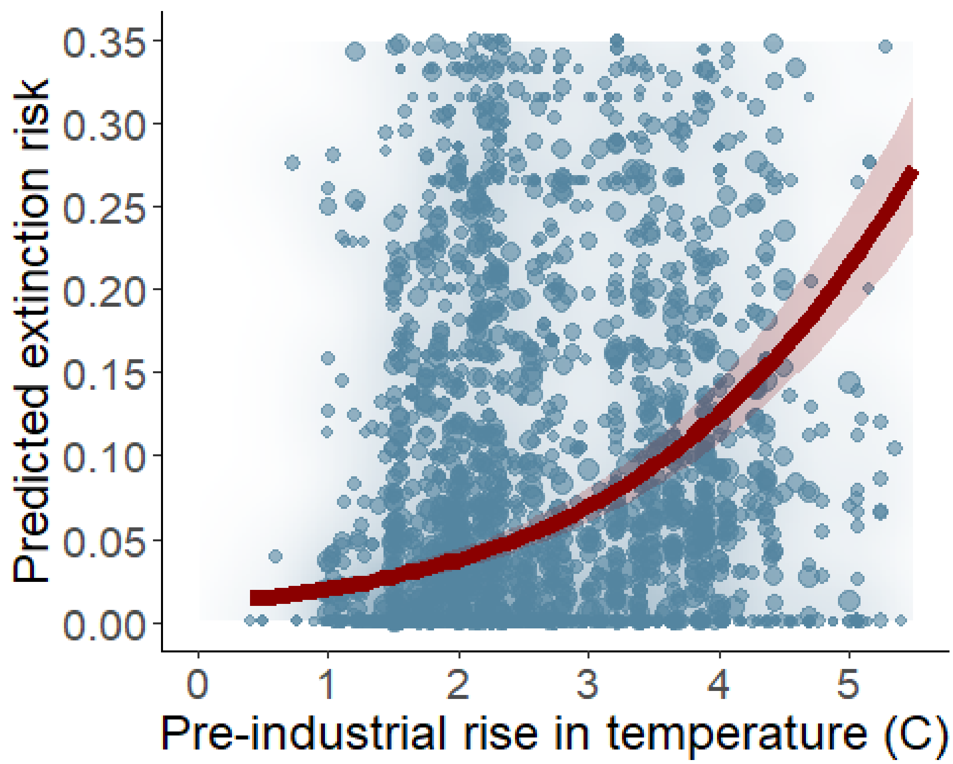

```
Fig1<-ggplot(data = pred.reg.df)+
  stat_density2d(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, fill =
..density..^.5),
                 geom = "tile", contour = FALSE, n = 200, show.legend =
FALSE, alpha = .8) +
  scale_fill_continuous(low = "white", high = "#5485A0") +
  geom_point(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, size =
log(Total.N)), alpha = 0.6, shape = 20, color = "#5485A0") +
  geom_ribbon(data = pred.reg.df,
```

```
aes(x=P.Ind,ymin=low_line,ymax=hi_line),alpha=.2,fill="darkred") +
  geom_line(data = pred.reg.df,
aes(x=P.Ind,y=mean_line),size=3,color="darkred") +
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk") +
  theme_classic()+ ylim(0,1) +  scale_x_continuous(breaks = seq(0,5,1),
limits = c(0,5.5)) + #xlim(0,6) +

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig1
```




```
#ggsave("FigS1 full preind.png",width=8,height=6,unit="in",dpi="print")

#-----------------------
#does not show full y axis for display
Fig1b<-ggplot(data = pred.reg.df)+
  stat_density2d(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, fill =
..density..^.5),
                geom = "tile", contour = FALSE, n = 200, show.legend =
FALSE, alpha = .8) +
  scale_fill_continuous(low = "white", high = "#5485A0") +
  geom_point(data = data.use, aes(x=Pre.Ind.Rise, y=percent2, size =
log(Total.N)), alpha = 0.6, shape = 20, color = "#5485A0") +
  geom_ribbon(data = pred.reg.df,
aes(x=P.Ind,ymin=low_line,ymax=hi_line),alpha=.2,fill="darkred") +
```

```
  geom_line(data = pred.reg.df,
aes(x=P.Ind,y=mean_line),size=3,color="darkred") +
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk") +
  theme_classic() + scale_x_continuous(breaks = seq(0,5,1), limits =
c(0,5.5)) + scale_y_continuous(breaks = seq(0,.35,0.05), limits = c(0,.35)) +

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig1b
```




```
#ggsave("FigS1b ylim preind.png",width=8,height=6,unit="in",dpi="print")

load("2pre_lowb.rds")
loo.mod2=loo.mod # rename loo.mod so can load n
mod2 = mod
load("2sub_interc.rds") #intercept only model

table.data<-data.frame(
  Model = c("Intercept-only model","Baseline model"),
  LOOic = c(loo.mod$estimates[3],loo.mod2$estimates[3]),
  SE = c(loo.mod$estimates[6],loo.mod2$estimates[6])
)
knitr::kable(table.data, caption = "Table x: Comparisons of LOOic between
intercept-only and baseline models", format = "markdown")
```

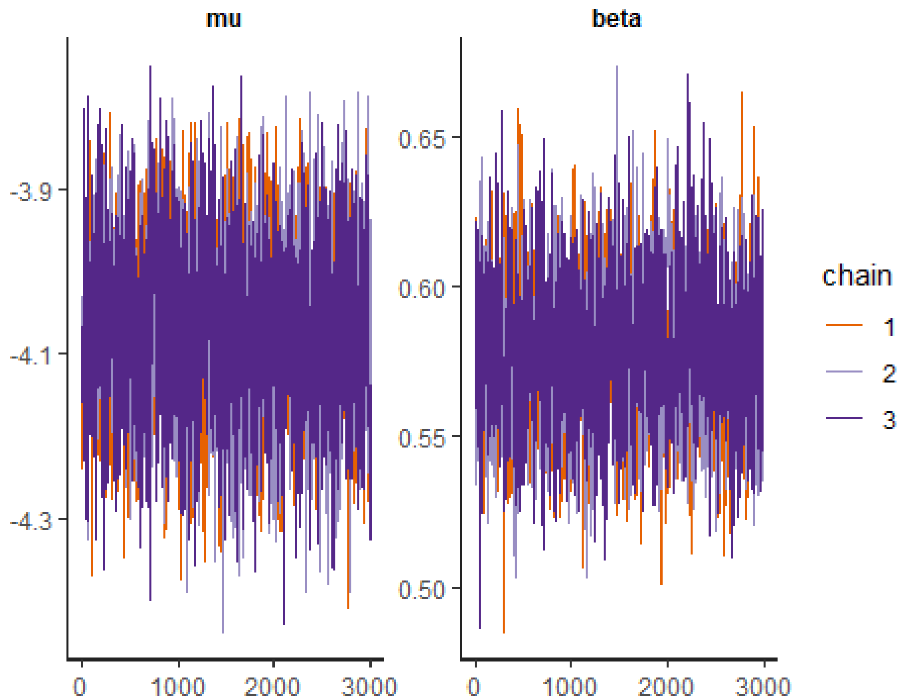
Table x: Comparisons of LOOic between intercept-only and baseline models

Model	LOOic	SE
Intercept-only model	-15151.75	193.0049
Baseline model	-15440.81	190.9764

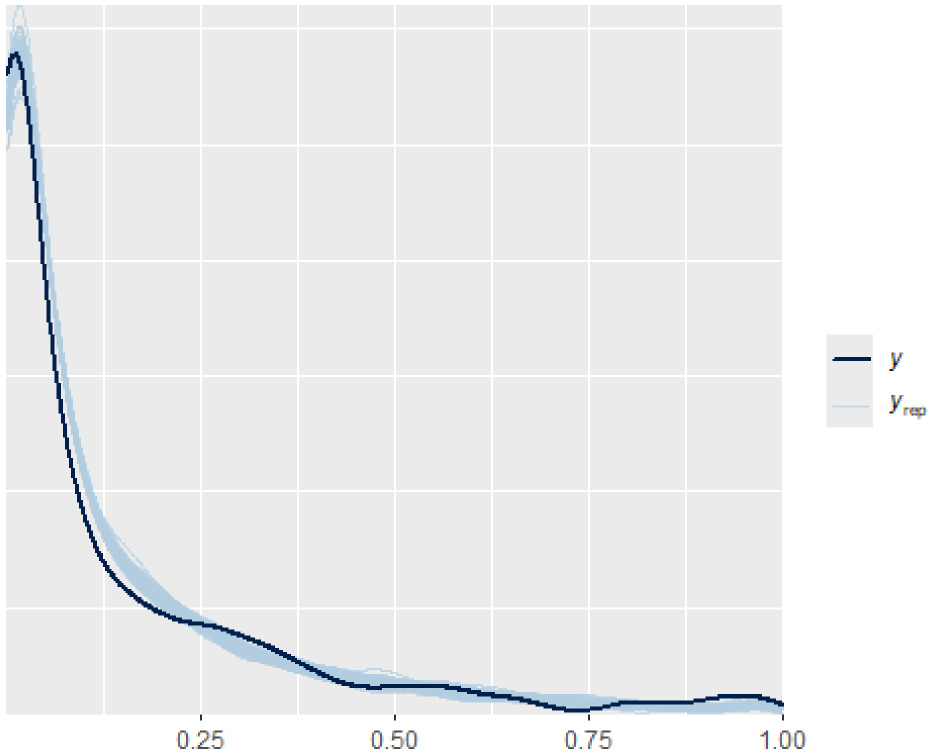
```
Looic.diff = loo.mod2$estimates[3] - loo.mod$estimates[3]
cat("difference in LOOic =", Looic.diff)

## difference in LOOic = -289.0682
```

#LOOic comparison The model with climate change is 289.1 less, and thus a better supported model.

```
# params.to.monitor=c("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")
# mod=stan(file="MetaRisk2 RSTAN
betareg.stan",data=stan.data,pars=params.to.monitor,
#           chains = 3, warmup=5000, cores=3,iter=8000, save_warmup = FALSE,
#           control=list(adapt_delta = 0.9, max_treedepth = 15))

load("2pre_low.rds") #mu prior (-50,5),less restriction on ~0 intercept

params.to.monitor2=c("mu","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
Rhat
## mu   -4.07321 0.0028302342 0.10029056 -4.2735393 -3.8804333 1255.669
1.004015
## beta  0.57809 0.0007381468 0.02368641  0.5341225  0.6252783 1029.706
1.002691

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

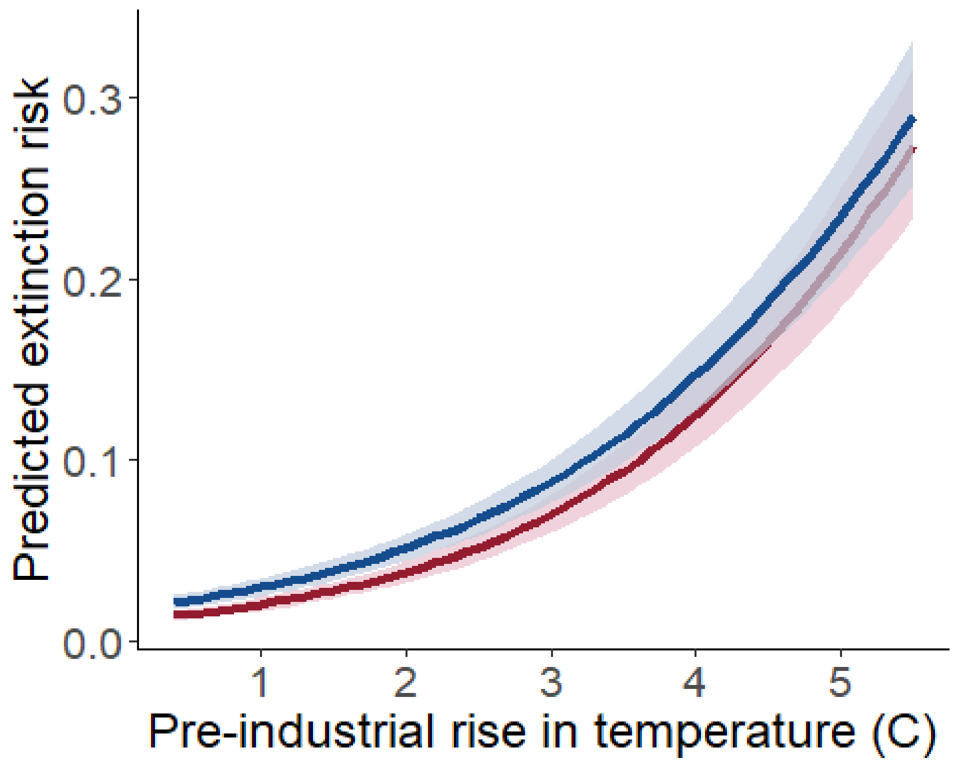



```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```




```
loo.mod #

##
## Computed from 9000 by 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   7719.0  95.3
## p_loo      1854.9  24.9
## looic    -15437.9 190.5
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      779  24.2%   401
##  (0.5, 0.7]   (ok)        753  23.4%   127
##    (0.7, 1]   (bad)      1443  44.8%   10
##    (1, Inf)   (very bad)  245   7.6%   3
## See help('pareto-k-diagnostic') for details.

load("2pre_low.rds") #all non-proportionate analysis
modx = mod

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
```

```
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5, 
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_2 = pred.reg.quant[2,],
                          low_line_2 = pred.reg.quant[1,],
                          hi_line_2= pred.reg.quant[3,])

load("2pre_lowb.rds")
mod2 = mod
posterior=as.data.frame(mod2);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x] 
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5, 
0.975),na.rm=TRUE))
pred.reg.df$mean_line_base = pred.reg.quant[2,]
pred.reg.df$low_line_base = pred.reg.quant[1,]
pred.reg.df$hi_line_base= pred.reg.quant[3,]

Fig3<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+

geom_ribbon(aes(x=P.Ind,ymin=low_line_2,ymax=hi_line_2),alpha=.7,fill="#Bfccd
c")+
  geom_line(aes(x=P.Ind,y=mean_line_2),size=1.5,color="#154c8e")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction 
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text 
= element_text(size=16))+
  guides(size=F)
Fig3
```




```
#ggsave("Fig Sx preind less inf
zero.png",width=8,height=5.5,unit="in",dpi="print")
```

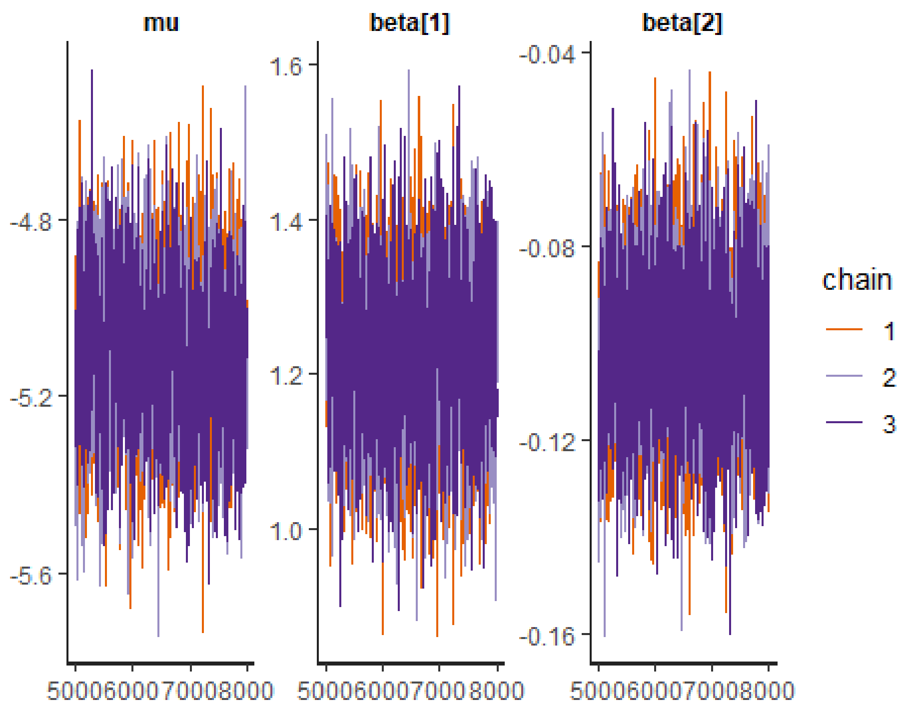
#Comparing models with weakly (blue) and strongly (red) informed priors on a zero intercept. The weakly informed model suggests a slightly higher extinction risk and a less realistic extinction risk at zero temperature, but the two results largely overlapped.

```
#create model matrix for coefficients
betamat=model.matrix(~-1+data.use$Pre.Ind.Rise + I(data.use$Pre.Ind.Rise^2))

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("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")

init.fn<- function (chain_id) {
  list(mu = -5, beta = c(0.5,0))
}

# mod=stan(file="MetaRisk2 RSTAN quad
2.stan",data=stan.data,pars=params.to.monitor,
#          chains = 3, warmup=5000, cores=7,iter=8000,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth =
15))

load("2poly2b.rds")
```

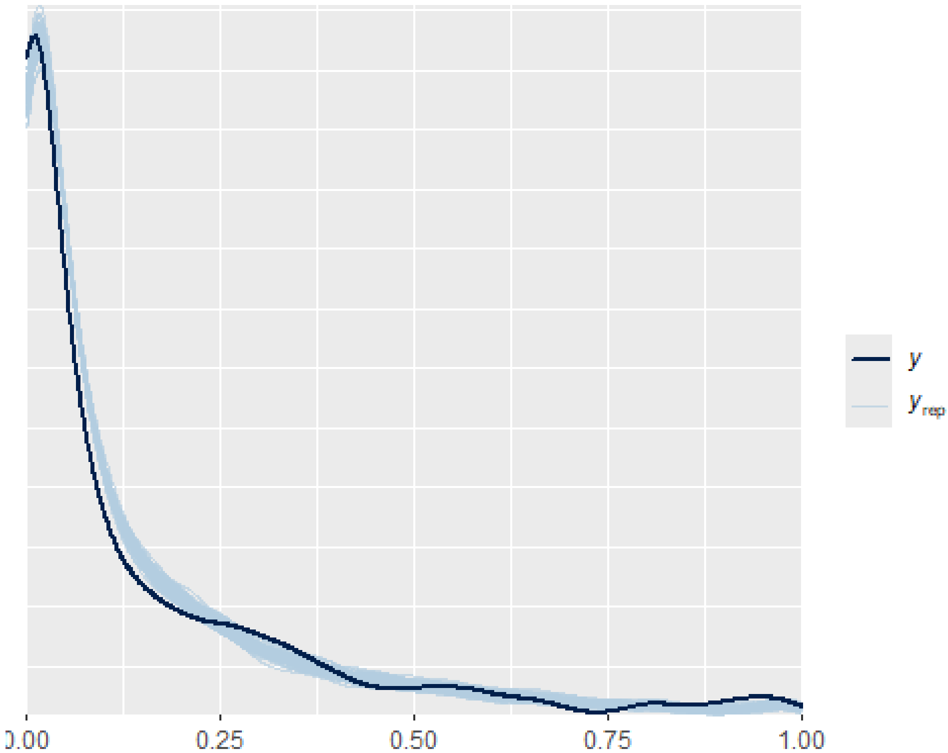
```
params.to.monitor2=c("mu","beta")#
sumx = summary(mod,probs=c(.025,0.975), digits=4, pars=params.to.monitor2)
sumx$summary
```

```
##                 mean      se_mean         sd       2.5%       97.5%    n_eff
## mu        -5.1029840 0.0045824419 0.16357411 -5.4165055 -4.78058972 1274.193
## beta[1]    1.2257276 0.0026266825 0.09734952  1.0331833  1.41175342 1373.575
## beta[2]   -0.1010257 0.0003983665 0.01538871 -0.1307256 -0.07057042 1492.241
##               Rhat
## mu        1.004443
## beta[1]   1.002768
## beta[2]   1.002888
```

```
#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 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   7717.2  95.3
## p_loo      1852.5  24.8
## looic    -15434.4 190.6
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      813  25.2%   449
##  (0.5, 0.7]   (ok)        732  22.7%   119
##    (0.7, 1]   (bad)      1434  44.5%   10
##    (1, Inf)   (very bad)  241   7.5%   3
## See help('pareto-k-diagnostic') for details.

##create data frame of looics from two models
load("2pre_lowb.rds")
loo.mod2=loo.mod # rename loo.mod so can load n
```

```r
load("2poly2b.rds")

table.data<-data.frame(
  Model = c("Baseline model","Polynommial model"),
  LOOic = c(loo.mod$estimates[3],loo.mod2$estimates[3]),
  SE = c(loo.mod$estimates[6],loo.mod2$estimates[6])
)
knitr::kable(table.data, caption = "Table x: Comparisons of LOOic between
linear and quadratic models") #, format = "simple"
```

Table x: Comparisons of LOOic between linear and quadratic models

Model	LOOic	SE
Baseline model	-15434.38	190.5743
Polynommial model	-15440.81	190.9764

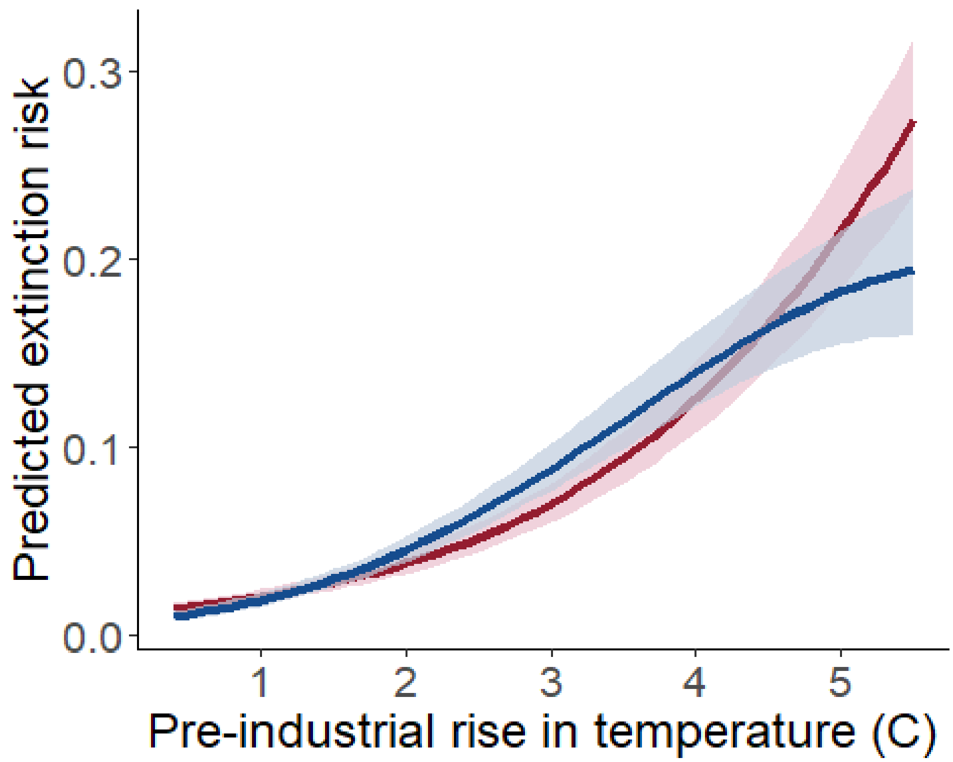
```r
Looic.diff = loo.mod$estimates[3] - loo.mod2$estimates[3]
cat("Difference in LOOic = ", Looic.diff)

## Difference in LOOic =  6.433208
```

Results Although the quadratic coefficient does not overlap zero, the overall model is worse as determined by the increase in LOOic = +6.4.

```r
P.Ind<-seq(from = 0.4, to = 5.5,by = .1)

load("2pre_lowb.rds") #
modx = mod

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_base = pred.reg.quant[2,],
                          low_line_base = pred.reg.quant[1,],
                          hi_line_base= pred.reg.quant[3,])

load("2poly2b.rds")
mod2 = mod
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]  1.2257276 0.0026266825 0.09734952  1.0331833  1.41175342 1373.575
## beta[2] -0.1010257 0.0003983665 0.01538871 -0.1307256 -0.07057042 1492.241
##              Rhat
```

```
## beta[1] 1.002768
## beta[2] 1.002888
```

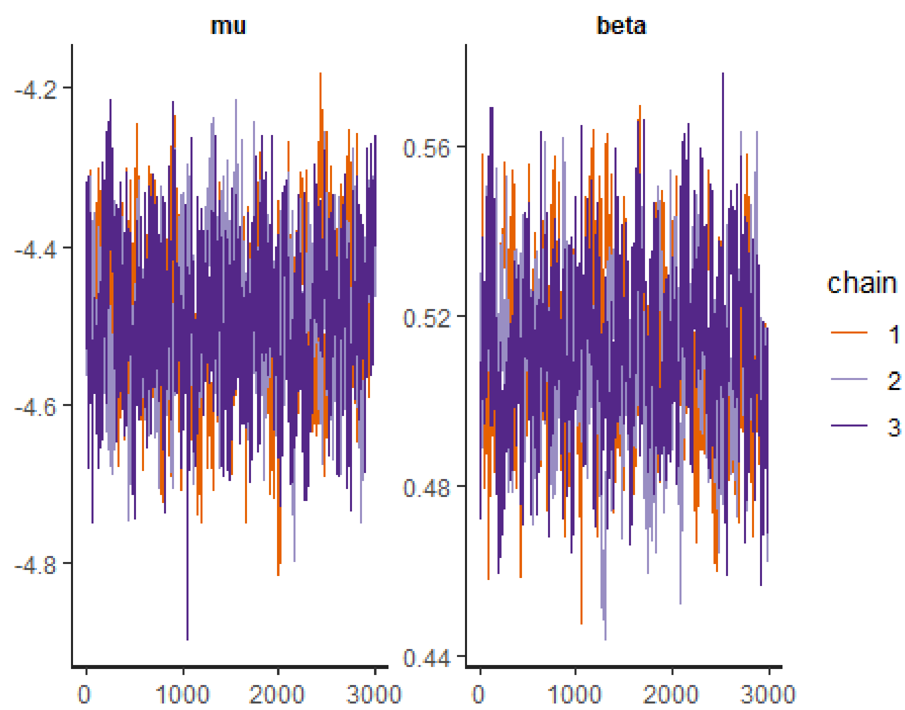
```
posterior=as.data.frame(mod2);
mu<-posterior[["mu"]]
beta1<-posterior[["beta[1]"]]
beta2<-posterior[["beta[2]"]]

# mu<-posterior[["beta[1]"]]
# beta1<-posterior[["beta[2]"]]
# beta2<-posterior[["beta[3]"]]

pred.reg = sapply(1:length(mu), FUN = function(x) {mu[x] + beta1[x]*P.Ind} +
beta2[x]*P.Ind^2)
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_quad = pred.reg.quant[2,]
pred.reg.df$low_line_quad = pred.reg.quant[1,]
pred.reg.df$hi_line_quad= pred.reg.quant[3,]

Fig2<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+

geom_ribbon(aes(x=P.Ind,ymin=low_line_quad,ymax=hi_line_quad),alpha=.7,fill="
#Bfccdc")+
  geom_line(aes(x=P.Ind,y=mean_line_quad),size=1.5,color="#154c8e")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig2
```



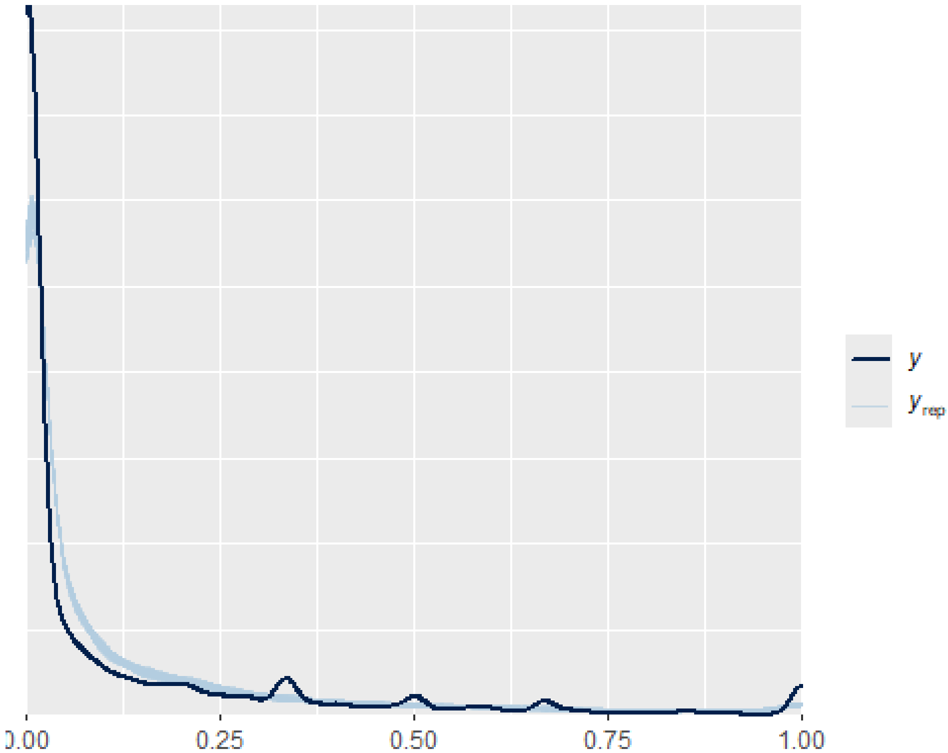

```
#ggsave("Fig Sx preind poly.png",width=8,height=5.5,unit="in",dpi="print")

#Bayesian stan model weighted, not aggregated data
rm(list = ls())
data <- read.table("MetaRisk for aggreg 5.txt",header=T); attach(data)

## The following objects are masked from dataP2:
##
##     Adaptation, Antarctic, Arctic, Author, Climate.Mod, concat,
##     Demography.LH, Disp.Mod, Dispersal, Earth.Sys, Endemic, Fresh,
##     Island, Land.Use.Change, Model.Type, Mtn, N.Middle,
##     Non.clim.threat, Other, Other.Habitat, Physiology, Pop.diff,
##     Pre.Ind.Rise, Region, S.Middle, Scenario, Sp.int, Study, Taxa,
##     Threatened, Time, Total.N, Tropics, version, WtSp, Year, Year.Pred

## The following object is masked from package:base:
##
##     version

dataP1<-data[is.finite(data$Pre.Ind.Rise),]; attach(dataP1) # need to
eliminate NA s for pre-industrial rise or stat programs crash

## The following objects are masked from data:
##
##     Adaptation, adj.perc, Antarctic, Arctic, Author, Climate.Mod,
##     concat, Demography.LH, Disp.Mod, Dispersal, Earth.Sys, Endemic,
##     Fresh, Island, Land.Use.Change, Model.Type, Mtn, N.Ext, N.Middle,
##     Non.clim.threat, Other, Other.Habitat, percent, Physiology,
```

```
##     Pop.diff, Pre.Ind.Rise, Region, S.Middle, Scenario, Sp.int, Study,
##     Taxa, Threatened, Threshold, Time, Total.N, Tropics, version, WtSp,
##     Year, Year.Pred

## The following objects are masked from dataP2:
##
##     Adaptation, Antarctic, Arctic, Author, Climate.Mod, concat,
##     Demography.LH, Disp.Mod, Dispersal, Earth.Sys, Endemic, Fresh,
##     Island, Land.Use.Change, Model.Type, Mtn, N.Middle,
##     Non.clim.threat, Other, Other.Habitat, Physiology, Pop.diff,
##     Pre.Ind.Rise, Region, S.Middle, Scenario, Sp.int, Study, Taxa,
##     Threatened, Time, Total.N, Tropics, version, WtSp, Year, Year.Pred

## The following object is masked from package:base:
##
##     version

#betareg requires no 0s or 1s
koffset = 0.001
percent2 <- percent
percent2[percent == 0] = koffset;
percent2[percent == 1] = 1 - koffset;
dataP1$percent2 <- percent2;
data.use<-dataP1

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

stan.data<-list(N = N, percent = data.use$percent2, Ind =
data.use$Pre.Ind.Rise, phi = phi, S = n.Study, Study = Studyint)
params.to.monitor=c("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")

init.fn<- function (chain_id) {
  list(mu = -4.5, beta = 0.5)
}

# mod=stan(file="MetaRisk2 RSTAN betareg
2b.stan",data=stan.data,pars=params.to.monitor,
#          chains = 3, warmup=5000, cores=3,iter=8000, save_warmup = FALSE,
#          control=list(adapt_delta = 0.9, max_treedepth = 15))

load("2noagg.rds")

params.to.monitor2=c("mu","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
Rhat
## mu   -4.486893 0.004636065 0.09022569 -4.6676879 -4.3112227 378.7576
1.001698
## beta  0.511584 0.001097013 0.01882255  0.4747098  0.5497651 294.3972
1.006725

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

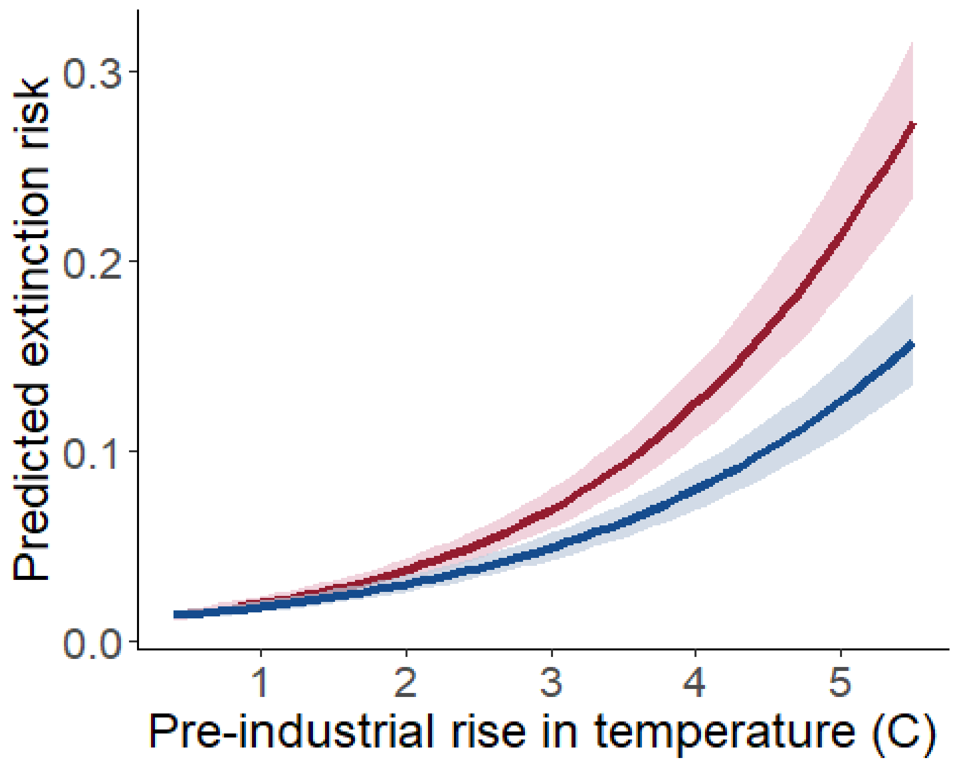



```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```




```
#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 7831 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  22212.8 144.5
## p_loo      4296.2  40.4
## looic    -44425.7 289.1
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     1784  22.8%   283
##  (0.5, 0.7]   (ok)       2279  29.1%   113
##    (0.7, 1]   (bad)      3226  41.2%   12
##    (1, Inf)   (very bad)  542   6.9%   2
## See help('pareto-k-diagnostic') for details.

load("2noagg.rds") #all non-proportionate analysis
modx = mod
P.Ind<-seq(from = 0.4, to = 5.5,by = .1) #don't extrapolate rise
```

```
min(Pre.Ind.Rise)= 0.4

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_all = pred.reg.quant[2,],
                          low_line_all = pred.reg.quant[1,],
                          hi_line_all= pred.reg.quant[3,])

load("2pre_lowb.rds")
mod2 = mod
posterior=as.data.frame(mod2);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_base = pred.reg.quant[2,]
pred.reg.df$low_line_base = pred.reg.quant[1,]
pred.reg.df$hi_line_base= pred.reg.quant[3,]

Fig3<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+

geom_ribbon(aes(x=P.Ind,ymin=low_line_all,ymax=hi_line_all),alpha=.7,fill="#B
fccdc")+
  geom_line(aes(x=P.Ind,y=mean_line_all),size=1.5,color="#154c8e")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig3
```

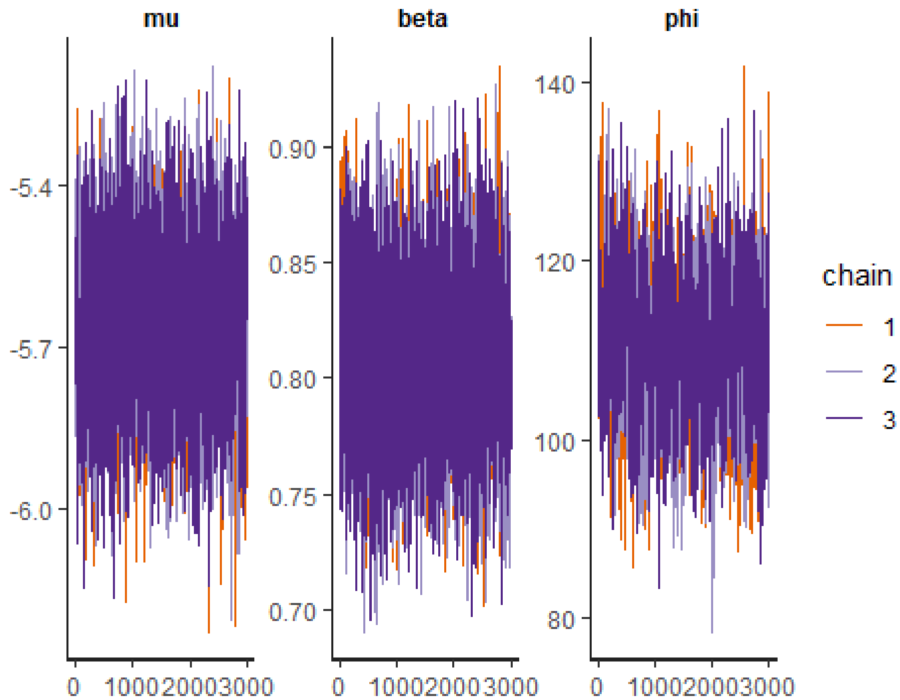



```
#ggsave("Fig Sx preind nonprop.png",width=8,height=5.5,unit="in",dpi="print")
```

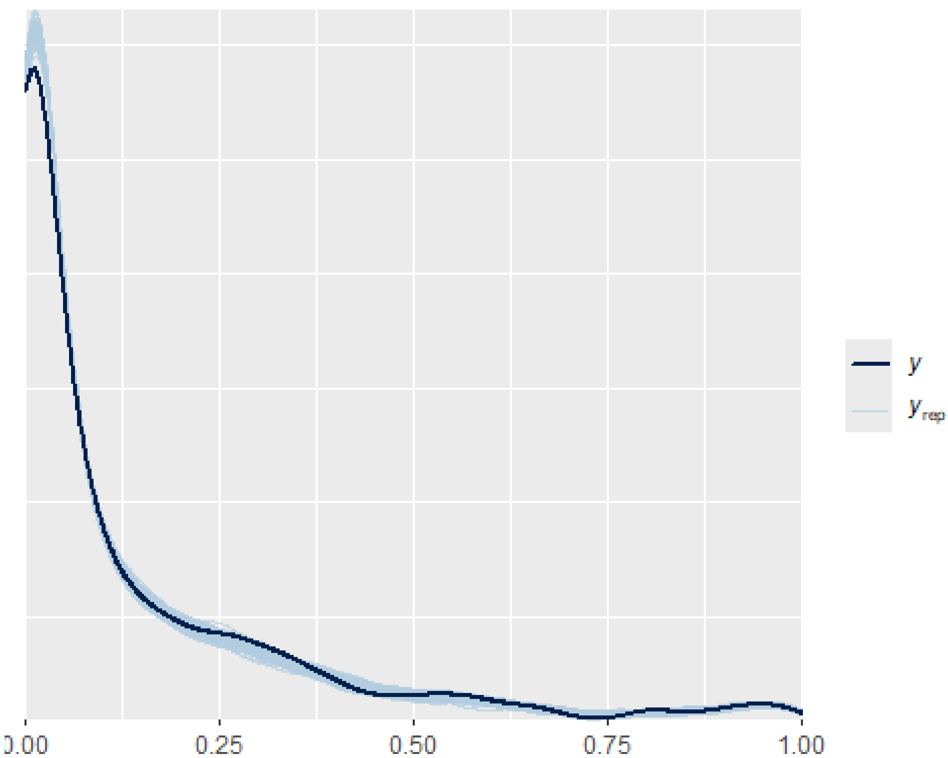
##Comparing all data beta analysis to proportionate beta analysis As expected, the beta regression that uses all data produces a lower prediction than the beta regression based on predictions proportionate to extinction risk. The reasoning is that the former method averages predictions across range loss scenarios, usually 80%, 95%, and 100%, and thus predicts extinction risk at ~ 92%. The latter, and at least in my mind preferred, method sets risk proportional to range loss. For example, if out of 10 species, 5 face extinction at 80% range loss, 2 at 95% range loss, and 1 at 100% range loss, the averaged method would suggest a predicted risk of (5 + 2 +1)/3x10 = 30%, and provides a highly conservative estimate especially if most range losses are between 80 - 95%. A proportionate analysis would suggest a predicted risk of [.8(5-1-1) + .95(2-1) + 1(1)]/10 = 44%. Given that many scientists would use the 80% range loss criterion to define future extinction risk and that this category includes range losses from 80-95%, I think that the proportionate response is still being conservative.

```
#Bayesian stan model proportional and weighted
dataP<-read.table("Metarisk2 aggthres 5.txt",header=T);
dataP2<-dataP[is.finite(dataP$Pre.Ind.Rise),]; attach(dataP2) # need to
eliminate NA s for pre-industrial rise or stat programs crash

## The following objects are masked from dataP2 (pos = 3):
##
##     Adaptation, adj.percent, Antarctic, Arctic, Author, ave.percent,
##     Climate.Mod, concat, Demography.LH, Disp.Mod, Dispersal, Earth.Sys,
##     Endemic, Fresh, Island, Land.Use.Change, max.percent, min.percent,
```

```
##     Model.Type, Mtn, N.Middle, Non.clim.threat, Other, Other.Habitat,
##     Physiology, Pop.diff, Pre.Ind.Rise, Region, S.Middle, Scenario,
##     Sp.int, Study, Taxa, Threatened, Time, Total.N, Tropics, version,
##     WtSp, Year, Year.Pred

## The following object is masked from package:base:
##
##     version

#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;
dataP2$percent2 <- percent2;

data.use<-dataP2

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

stan.data<-list(N = N, percent = data.use$percent2, Ind =
data.use$Pre.Ind.Rise, S = n.Study, Study = Studyint)
params.to.monitor=c("mu","beta","phi","y_rep","stu","sigma_stu",
"eta","log_lik")

# mod=stan(file="MetaRisk2 RSTAN betareg notwtd
5.stan",data=stan.data,pars=params.to.monitor,
#          chains = 3, warmup=7000, cores=3,iter=10000, save_warmup = FALSE,
#          control=list(adapt_delta = 0.9, max_treedepth = 15))

load("2pre_nowt5b.rds")

params.to.monitor2=c("mu","beta","phi")#
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
## mu    -5.6565485 0.0041108524 0.14733084 -5.9513432  -5.3741136 1284.4689
## beta   0.8085877 0.0007533243 0.03438516  0.7419894   0.8773065 2083.4265
## phi  109.9093158 0.3212787960 7.65460609 95.3664701 125.1977191  567.6512
##          Rhat
## mu   1.001016
## beta 1.000847
## phi  1.002911

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

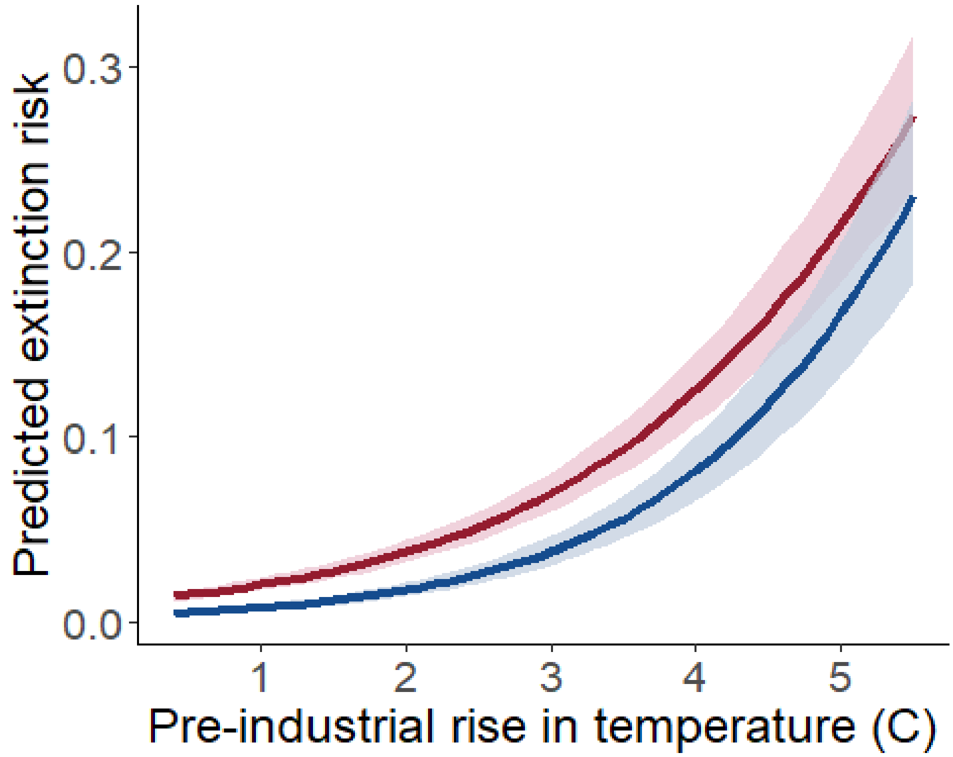



```r
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```




```
#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 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   7898.4  84.9
## p_loo      2322.6  19.9
## looic    -15796.8 169.8
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      149   4.6%   280
##  (0.5, 0.7]   (ok)        704  21.9%   98
##    (0.7, 1]   (bad)      1985  61.6%   11
##    (1, Inf)   (very bad)  382  11.9%   3
## See help('pareto-k-diagnostic') for details.

load("2pre_nowt5b.rds") #all non-proportionate analysis
modx = mod
```

```r
#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_unwt = pred.reg.quant[2,],
                          low_line_unwt = pred.reg.quant[1,],
                          hi_line_unwt = pred.reg.quant[3,])

load("2pre_lowb.rds")
mod2 = mod
posterior=as.data.frame(mod2);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_base = pred.reg.quant[2,]
pred.reg.df$low_line_base = pred.reg.quant[1,]
pred.reg.df$hi_line_base= pred.reg.quant[3,]

Fig4<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+

geom_ribbon(aes(x=P.Ind,ymin=low_line_unwt,ymax=hi_line_unwt),alpha=.7,fill="
#Bfccdc")+
  geom_line(aes(x=P.Ind,y=mean_line_unwt),size=1.5,color="#154c8e")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig4
```

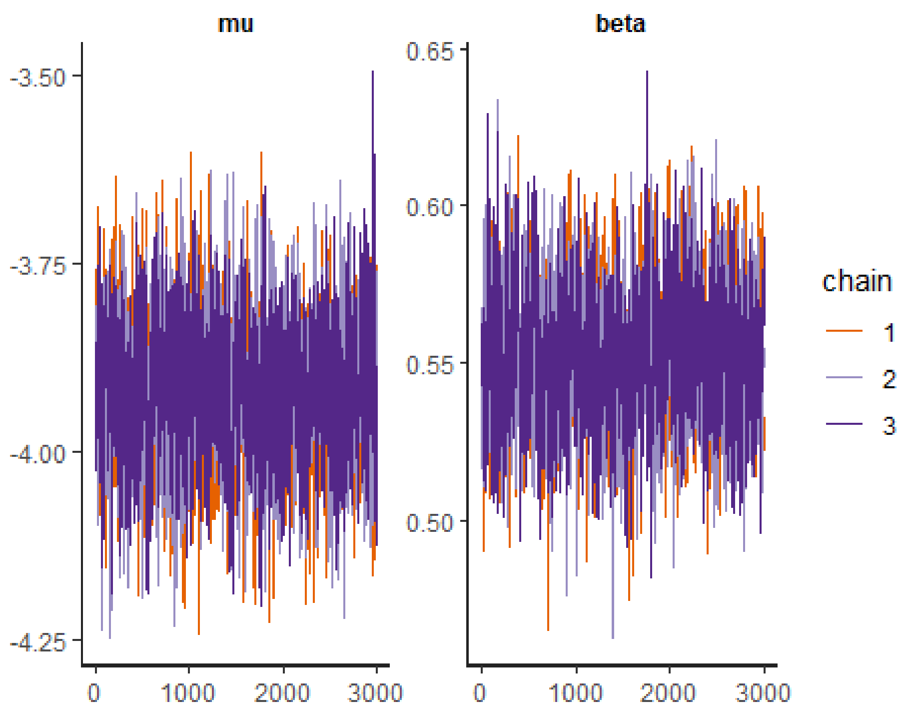



```
#ggsave("Fig S3 preind unwtd.png",width=8,height=5.5,unit="in",dpi="print")
```

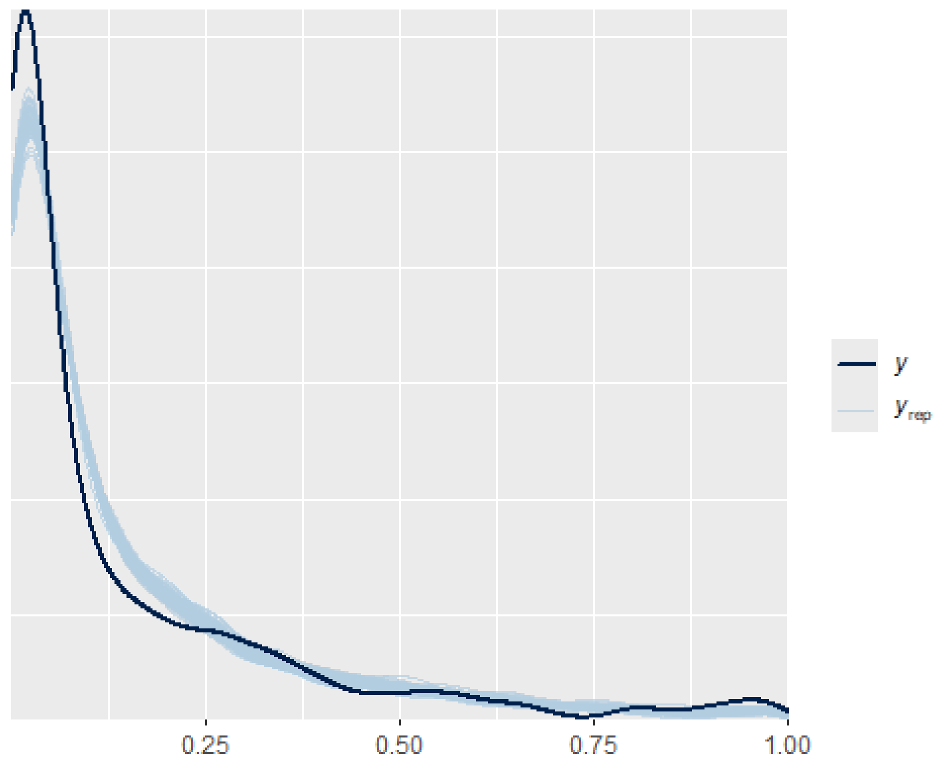
#Comparing unweighted (blue) vs. weighted (red) proportional analyses Without weighting the predictions, the estimated relationship with temperature rise is lower than the version weighted by beta variance (total N). Also, the unweighted version suggests a smaller intercept. The unweighted version overestimates the number of zeros (see posterior check).

```
#Bayesian stan model proportional and weighted
#betareg requires no 0s or 1s
koffset = 0.01
percent2 <- adj.percent
percent2[adj.percent == 0] = koffset;
percent2[adj.percent == 1] = 1 - koffset;
dataP2$percent2 <- percent2;

data.use<-dataP2

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

stan.data<-list(N = N, percent = data.use$percent2, Ind =
data.use$Pre.Ind.Rise, phi = phi, S = n.Study, Study = Studyint)
params.to.monitor=c("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")
```

```
# mod=stan(file="MetaRisk2 RSTAN betareg
2b.stan",data=stan.data,pars=params.to.monitor,
#          chains = 3, warmup=5000, cores=3,iter=8000, save_warmup = FALSE,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth =
15))

load("2pre_khi.rds")

params.to.monitor2=c("mu","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
Rhat
## mu   -3.9194651 0.0026243427 0.09169818 -4.1018535 -3.742388 1220.901
1.000689
## beta  0.5522462 0.0006435029 0.02135016  0.5114661  0.594004 1100.782
1.001634

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

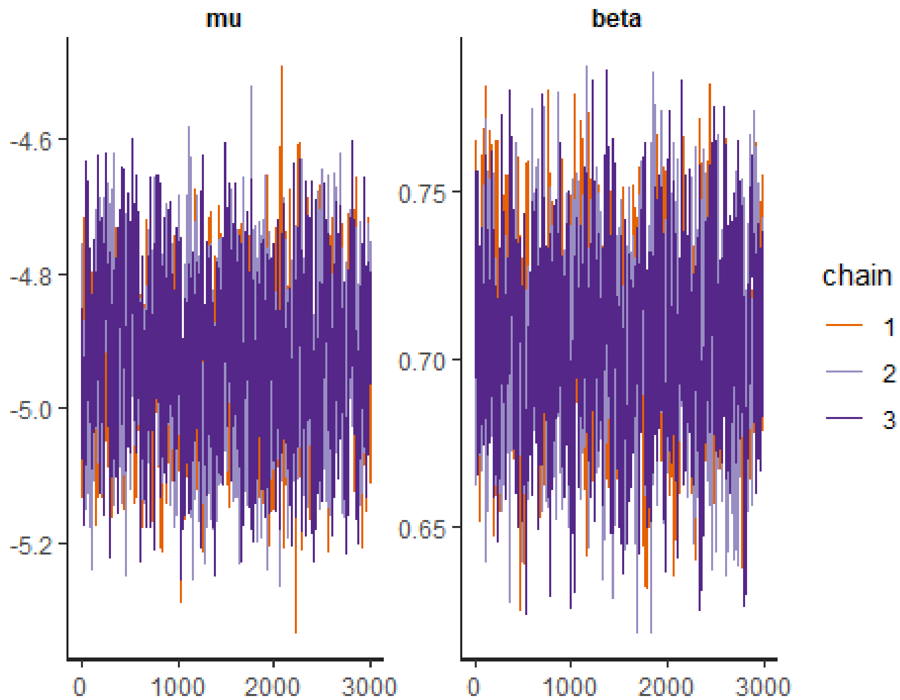



```
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```

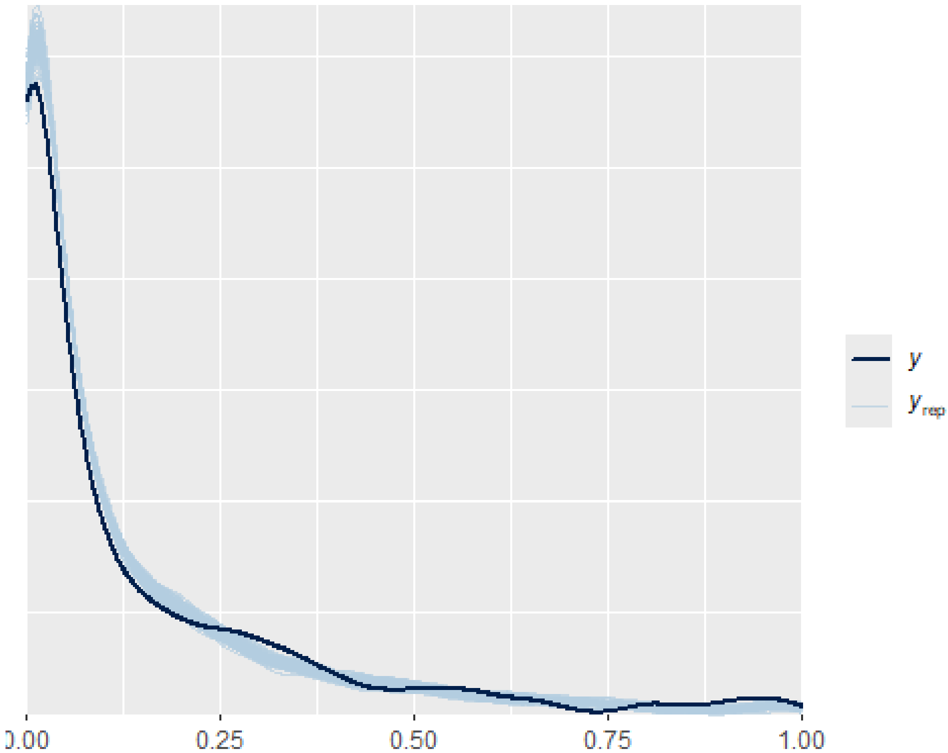



```
#ggsave("Fig S1a koffset 01.png",width=8,height=5.5,unit="in",dpi="print")
#calculate loo
# log_lik_1 <- extract_log_lik(mod, merge_chains = FALSE)
# r_eff <- relative_eff(exp(log_lik_1), cores = 6)
# loo.mod <- loo(log_lik_1, r_eff = r_eff, cores = 6)
loo.mod #

##
## Computed from 9000 by 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   6008.2  75.6
## p_loo      1723.3  24.7
## looic    -12016.4 151.1
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      918  28.5%   547
##  (0.5, 0.7]   (ok)        701  21.8%   116
##    (0.7, 1]   (bad)      1389  43.1%   15
##    (1, Inf)   (very bad)  212   6.6%   3
## See help('pareto-k-diagnostic') for details.

#Bayesian stan model proportional and weighted
#betareg requires no 0s or 1s
```

```
koffset = 0.0001
percent2 <- adj.percent
percent2[adj.percent == 0] = koffset;
percent2[adj.percent == 1] = 1 - koffset;
dataP2$percent2 <- percent2;

data.use<-dataP2

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

stan.data<-list(N = N, percent = data.use$percent2, Ind =
data.use$Pre.Ind.Rise, phi = phi, S = n.Study, Study = Studyint)
params.to.monitor=c("mu","beta","y_rep","stu","sigma_stu", "eta","log_lik")
# mod=stan(file="MetaRisk2 RSTAN betareg
2b.stan",data=stan.data,pars=params.to.monitor,
#          chains = 3, warmup=5000, cores=3,iter=8000, save_warmup = FALSE,
#          init = init.fn, control=list(adapt_delta = 0.9, max_treedepth =
15))
load("2pre_klo.rds")

params.to.monitor2=c("mu","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
Rhat
## mu   -4.9280038 0.004036842 0.11050444 -5.1441473 -4.7119736 749.3349
1.003533
## beta  0.7050835 0.001092128 0.02606376  0.6541766  0.7561688 569.5440
1.007336

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



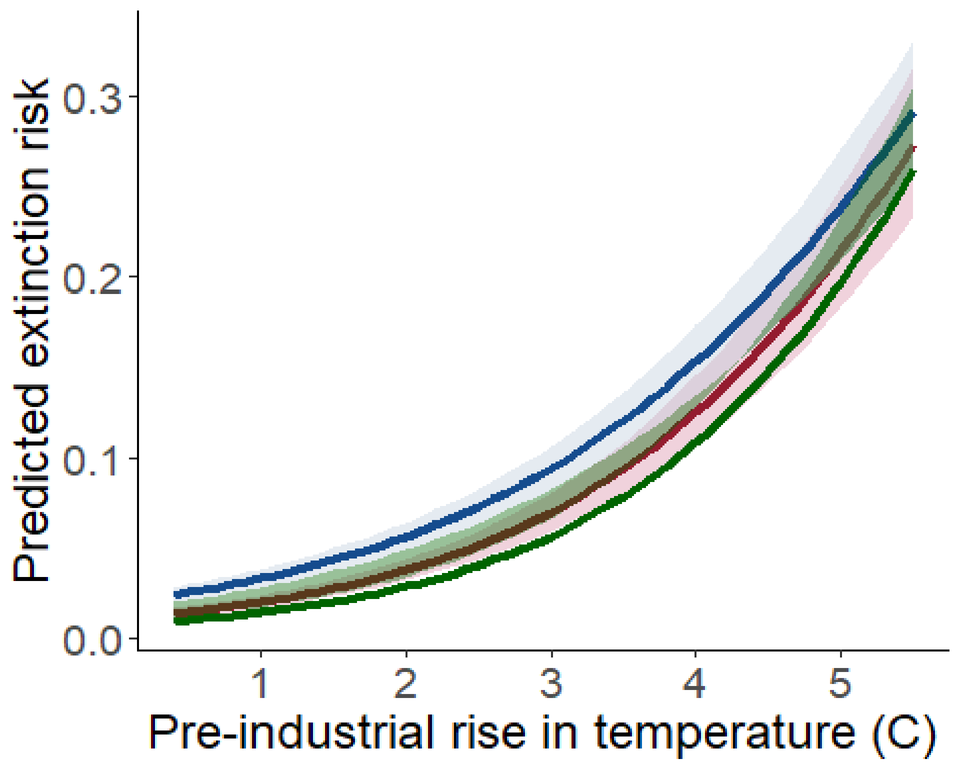

```r
pp_check(
  stan.data$percent,
  rstan::extract(mod, par = 'y_rep')$y_rep[1:100, ],
  fun = 'dens_overlay'
)
```




```
#ggsave("Fig S1c koffset 0001.png",width=8,height=5.5,unit="in",dpi="print")
#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 3220 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo   9680.9 132.6
## p_loo      1961.1  25.0
## looic    -19361.8 265.2
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)      776  24.1%   278
##  (0.5, 0.7]   (ok)        693  21.5%   91
##    (0.7, 1]   (bad)      1487  46.2%   13
##    (1, Inf)   (very bad)  264   8.2%   3
## See help('pareto-k-diagnostic') for details.

#
```

```
#----------------------------------------------------------------------------
-----------
#hi offset
load("2pre_khi.rds") #all non-proportionate analysis
modx = mod

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_bigk = pred.reg.quant[2,],
                          low_line_bigk = pred.reg.quant[1,],
                          hi_line_bigk = pred.reg.quant[3,])
#----------------------------------------------------------------------------
-----------
#small offset
load("2pre_klo.rds") #all non-proportionate analysis
modx = mod

#Calculate estimates
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_lok = pred.reg.quant[2,]
pred.reg.df$low_line_lok = pred.reg.quant[1,]
pred.reg.df$hi_line_lok = pred.reg.quant[3,]
#----------------------------------------------------------------------------
-----------
#just right offset
load("2pre_lowb.rds")
modx = mod
posterior=as.data.frame(modx);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_base = pred.reg.quant[2,]
pred.reg.df$low_line_base = pred.reg.quant[1,]
pred.reg.df$hi_line_base= pred.reg.quant[3,]

Fig5<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+
```

```
geom_ribbon(aes(x=P.Ind,ymin=low_line_bigk,ymax=hi_line_bigk),alpha=.4,fill="
#Bfccdc")+
  geom_line(aes(x=P.Ind,y=mean_line_bigk),size=1.5,color="#154c8e")+

geom_ribbon(aes(x=P.Ind,ymin=low_line_bigk,ymax=hi_line_lok),alpha=.4,fill="d
arkgreen")+
  geom_line(aes(x=P.Ind,y=mean_line_lok),size=1.5,color="darkgreen")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig5
```

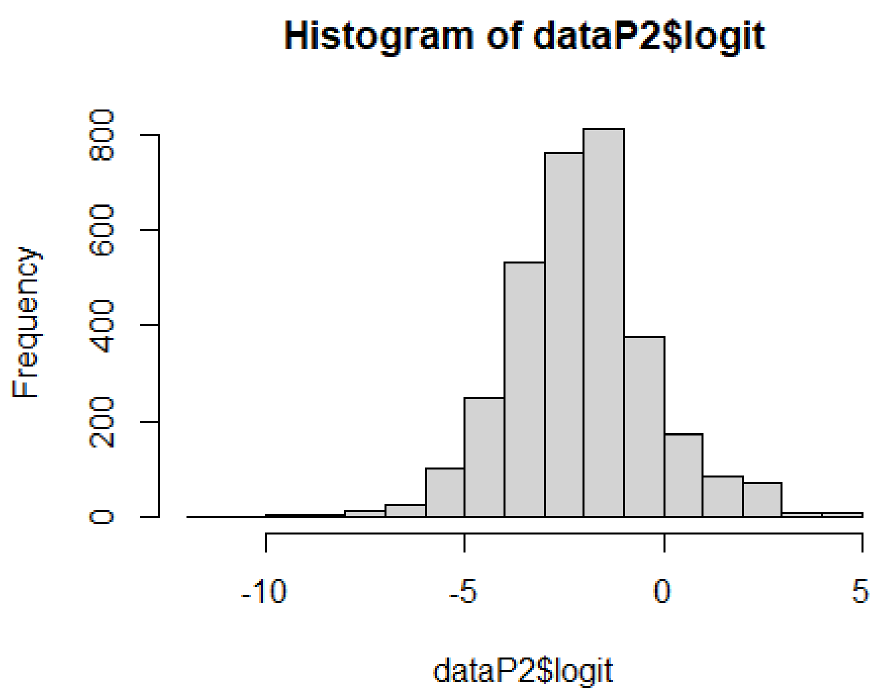



```
#ggsave("Fig S2 preind dfrt
offsets.png",width=8,height=5.5,unit="in",dpi="print")
```

#Offset analysis Changing the offset for zeros has a minor effect on results, with slightly higher predictions for a larger offset relative to baseline and lower predictions for a lower offset than baseline. However, the credible intervals all overlap.

```
#Bayesian stan model proportional and weighted
#betareg requires no 0s or 1s
dataP2$logit<-logit(adj.percent)
```

```
dataP2$logit[dataP2$adj.percent == 0] = log((.5)/((Total.N[dataP2$adj.percent
== 0] + 1)-(.5)))
dataP2$logit[dataP2$adj.percent == 1] = log((Total.N[dataP2$adj.percent == 1]
+ .5)/((Total.N[dataP2$adj.percent == 1] + 1)-(Total.N[dataP2$adj.percent ==
1] + .5))) #original koffset method

hist(dataP2$logit)
```

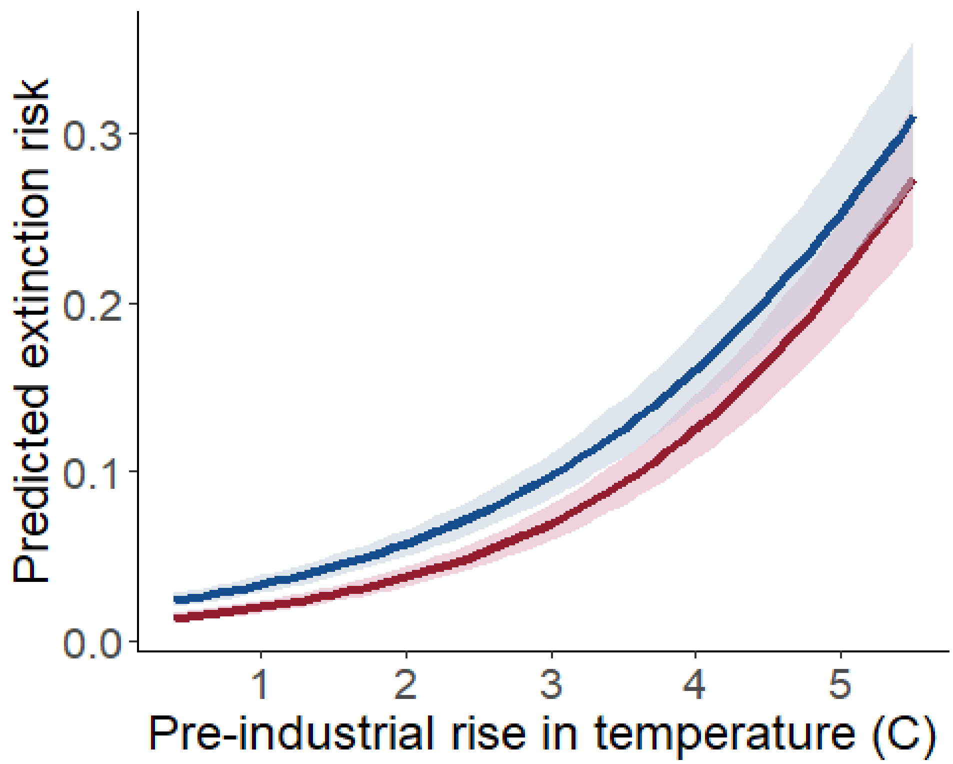



```
vari<-(adj.percent * (1 - adj.percent))/Total.N

betam<-rbind(cbind(-50,0),cbind(0,0))
betav<-rbind(cbind(1,0),cbind(0,100))

prior <- list(B = list(mu = betam, V = betav), R = list(V = 1, nu = 0.002), G
= list(G1=list(V = 1, nu=.002))) #standard weakly informative priors, except
for intercept
 model.log<-MCMCglmm(logit~Pre.Ind.Rise,random=~Study,
mev=vari,nitt=50000,data=dataP2, prior=prior, burnin=40000,thin =10)

#load("1pre ind logit.rds")
summary(model.log)

##
##  Iterations = 40001:49991
##  Thinning interval  = 10
##  Sample size  = 1000
##
```

```
##  DIC: 9503.953
##
##  G-structure:  ~Study
##
##       post.mean l-95% CI u-95% CI eff.samp
## Study     2.187    1.842    2.497     1000
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units    0.9773   0.9208    1.029     1000
##
##  Location effects: logit ~ Pre.Ind.Rise
##
##              post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)    -3.9343  -4.1317  -3.7630     1000 <0.001 ***
## Pre.Ind.Rise    0.5694   0.5320   0.6140     1000 <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#save(model.log,file = "2pre ind logit 2.rds")

load("2pre ind logit 2.rds") #all non-proportionate analysis
posterior <- as.data.frame(model.log$Sol)
pred.reg = sapply(1:1000, FUN = function(x) {posterior[x,1] +
posterior$Pre.Ind.Rise[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df <- data.frame(x = P.Ind,
                          mean_line_log = pred.reg.quant[2,],
                          low_line_log = pred.reg.quant[1,],
                          hi_line_log = pred.reg.quant[3,])

load("2pre_lowb.rds")
mod2 = mod
posterior=as.data.frame(mod2);
pred.reg = sapply(1:length(posterior$mu), FUN = function(x) {posterior$mu[x]
+ posterior$beta[x]*P.Ind})
pred.reg.quant = invlogit(apply(pred.reg, 1, quantile, probs = c(0.025, 0.5,
0.975),na.rm=TRUE))
pred.reg.df$mean_line_base = pred.reg.quant[2,]
pred.reg.df$low_line_base = pred.reg.quant[1,]
pred.reg.df$hi_line_base= pred.reg.quant[3,]

Fig6<-ggplot(data = pred.reg.df)+

geom_ribbon(aes(x=P.Ind,ymin=low_line_base,ymax=hi_line_base),alpha=.7,fill="
#Eabecd")+
  geom_line(aes(x=P.Ind,y=mean_line_base),size=1.5,color="#941C2F")+
```

```
geom_ribbon(aes(x=P.Ind,ymin=low_line_log,ymax=hi_line_log),alpha=.5,fill="#B
fccdc")+
  geom_line(aes(x=P.Ind,y=mean_line_log),size=1.5,color="#154c8e")+
  xlab("Pre-industrial rise in temperature (C)") + ylab("Predicted extinction
risk")+
  theme_classic()+

theme(axis.title=element_text(size=18),title=element_text(size=20),axis.text
= element_text(size=16))+
  guides(size=F)
Fig6
```




```
#ggsave("Fig S4 preind orig vs
new.png",width=8,height=5.5,unit="in",dpi="print")
```

#Comparing baseline analysis (red) with the original Gaussian analysis of logits (blue) with faded 95% credible intervals The two analyses overlap, with a lower result for the baseline analysis.

Variation explained

```
#After Gelman 2019 R2 for Bayesian
#
#Load model and beta matrix - check if mu is modeled separately
load("2pre_lowb.rds")
posterior=as.data.frame(mod); #caution mu/beta model not just beta
```

```
betamat <- (model.matrix(~Pre.Ind.Rise,data=data.use))

#Variables and matrices
S = 9000; #samples
K = ncol(betamat); #factors
p.mat <- as.matrix(posterior[,1:K])
y = dataP2$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$Total.N[i])/(theta[,i] *
data.use$Total.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 =  0.07022201 0.09381807 0.1225552

#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)
```

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

## Overall model R2 =  0.7655744 0.7861943 0.8064076
```