Data simulation and analysis of population trends with GLMMs

Supporting documentation for: Unreliable inferences about chinstrap penguin population trends: a statistical critique and reanalysis

Chris Oosthuizen, Murray Christian, Azwianewi B. Makhado, Mzabalazo Ngwenya

2023-06-19

Purpose

This script simulates population counts over time to test glmer (lme4) and MCMCglmm (MCMCglmm) mixed model specifications. This simulation study shows some of the errors in Krüger (2023)'s MCMCglmm analysis code and show why the results from models used in that paper are not useful. (Citation: Krüger, L. (2023). Decreasing Trends of Chinstrap Penguin Breeding Colonies in a Region of Major and Ongoing Rapid Environmental Changes Suggest Population Level Vulnerability. Diversity, 15(3), 327.)

Simulate data

```
# load library
library(tidyverse)
library(lme4)
library(effects)
library(MCMCglmm)

#-----------------------------------
# Simulate data
#-----------------------------------

# Simulation based on Chapter 5 from State-Space Models for Population Counts from
# Bayesian Population Analysis using Winbugs by Marc Kery and Michael Schaub

# Make an empty list to save output in
list1 = list()   # for population counts
list2 = list()   # for lambda of each population

# Set seed for reproducibility
set.seed(1234)

# Choose how many populations and how many years you want to simulate
n.populations = 26 # Number of populations (max = 26)
n.years <- 60 # Number of years
start.year = 1960  # Start year
years = start.year:(start.year+n.years-1)  # Year sequence
```

```r
# simulate
for(i in 1:n.populations) {

  N1 <- runif(1, 500, 5000) # Initial population size
  mean.lambda <- runif(1, 0.95, 1.02) # Mean annual population growth rate
  sigma2.lambda <- 0.0001 # Process (temporal) variation of the growth rate
  sigma2.y <- 0 # Variance of observation error (0 assumes 100% accurate counts)

  y <- N <- numeric(n.years)
  N[1] <- N1

  lambda <- rnorm(n.years-1, mean.lambda, sqrt(sigma2.lambda))

  for (t in 1:(n.years-1)){
    N[t+1] <- N[t] * lambda[t]
  }

  for (t in 1:n.years){
    y[t] <- rnorm(1, N[t], sqrt(sigma2.y))
  }

  # Save output in list for each iteration
  list1[[i]] = as.data.frame(y)
  list2[[i]] = as.data.frame(mean.lambda)
}

# list1
# list2

# Build data frame from simulations of count
df = bind_rows(list1)
names(df) = "count"
# add year to simulated counts
df$year = as.integer(rep(years,n.populations))
# add site to simulated counts
df$site = rep(LETTERS[1:n.populations], each = n.years)
# assign a random latitude to each site
# df$latitude.random = rep(runif(n.populations, -70, -60), each = n.years)
df = as_tibble(df)

# Use list 2 (lambda) to generate a latitude value for each site that
# correlate with the site's growth rate (lambda)
lambda = bind_rows(list2)
lambda$site = LETTERS[1:n.populations]
lambda$noise = runif(n.populations, -1, 1)

df = merge(df, lambda, by = "site")

df$r = df$mean.lambda-1   # convert lambda to growth rate r
df$r100 = df$r * 100   # rescale
range(df$r100)
```

```
## [1] -4.646079  1.965539
```

```r
head(df)
```

```
##   site     count year mean.lambda      noise            r       r100
## 1    A 1011.6654 1960    0.993561 -0.3990841 -0.006439042 -0.6439042
## 2    A 1007.9579 1961    0.993561 -0.3990841 -0.006439042 -0.6439042
## 3    A 1012.3983 1962    0.993561 -0.3990841 -0.006439042 -0.6439042
## 4    A  982.1316 1963    0.993561 -0.3990841 -0.006439042 -0.6439042
## 5    A  980.0222 1964    0.993561 -0.3990841 -0.006439042 -0.6439042
## 6    A  978.6712 1965    0.993561 -0.3990841 -0.006439042 -0.6439042
```

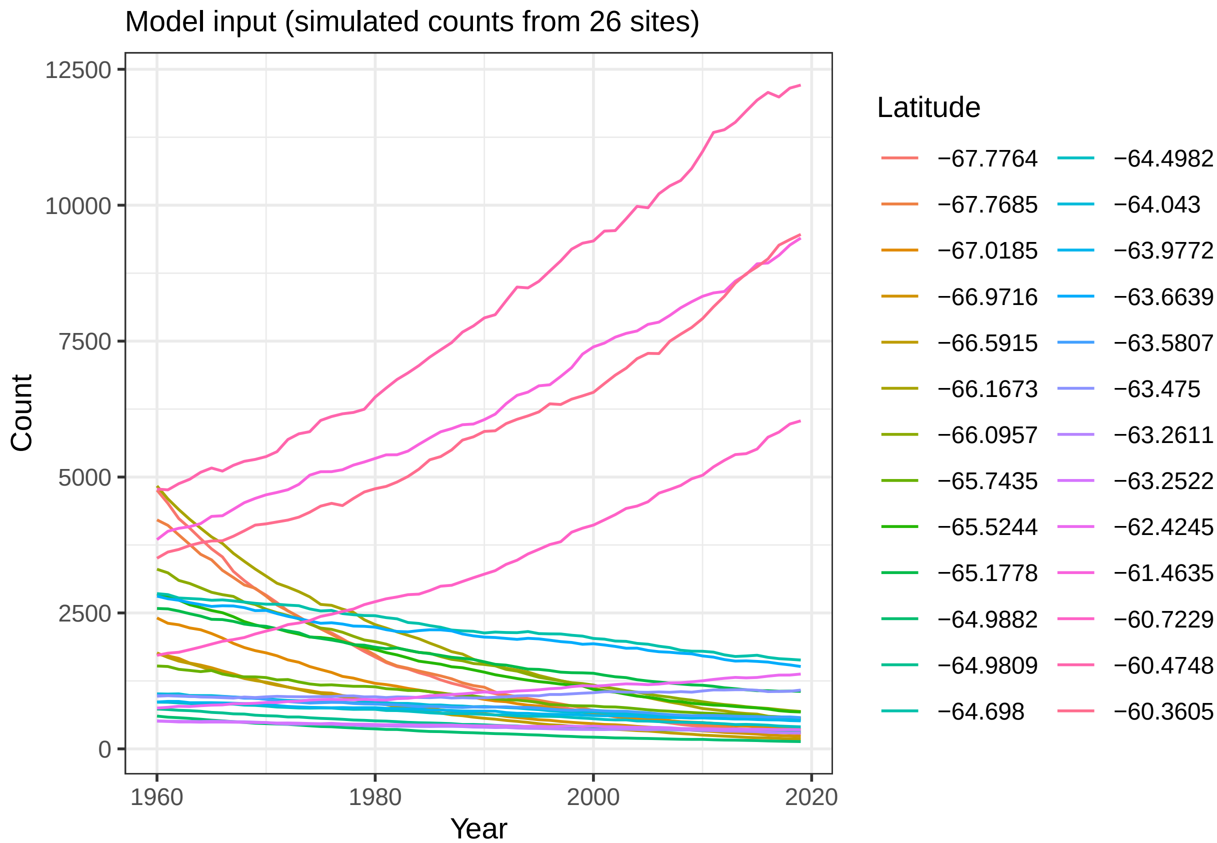
```r
# create one latitude value per site where the mean latitude (-63 degrees S)
# increase or decrease based on the growth rate of the population plus a small
# random component

df = df %>%
  group_by(site) %>%
  mutate(latitude = -63 + r100 + noise) %>%
  ungroup()

range(df$latitude)
```

```
## [1] -67.77641 -60.36049
```

```r
# Inspect df: every site should have 1 unique latitude value
df %>%
  group_by(site) %>%
  summarise(count = n_distinct(latitude)) %>%
  summarise(max_sites_per_lat = max(count))
```

```
## # A tibble: 1 x 1
##   max_sites_per_lat
##               <int>
## 1                 1
```

```r
# Counts must be positive and integers
df = df %>%
  dplyr::filter(count > 0) %>%
  dplyr::select(site, count, year, latitude) %>%
  mutate_at(vars(latitude), round, 4) %>%
  mutate_at(vars(count), round, 0)


# Plot data used for fitting models
ggplot(df, aes(x = year, y = count, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude")+
  theme_bw()+
  labs(subtitle = "Model input (simulated counts from 26 sites)")
```



glmer (lme4) analysis

```
# Fit in frequentist framework - lme4

#library(lme4)

# random slope model with interaction between year and latitude that test
# whether latitude influence population trend
m1 = glmer(count ~ year * latitude + (year|site), family = "poisson", data = df)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
## Error in (function (fr, X, reTrms, family, nAGQ = 1L, verbose = 0L, maxit = 100L, : Downdated VtV is
```

```
# model convergence problems occur. Can be avoided by scaling variables
# to mean = 0, sd = 1
df$zyear = scale(df$year)
df$zlatitude = scale(df$latitude)

# Refit with scaled predictors variables
m2 = glmer(count ~ zyear * zlatitude + (zyear|site), family = "poisson", data = df)
```

```r
# random slope model: assume latitude does not affect overall count (intercepts),
# only the slope of the year effect
m3 = glmer(count ~ zyear + zyear:zlatitude + (zyear|site), family = "poisson", data = df)

# Frequentist representation of the Kruger (2023) model
m4 = glmer(count ~ zyear + (zlatitude|site), family = "poisson", data = df)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly uniden
##  - Rescale variables?
```

```r
# Note the warning message, even with z-standardized covariates

# Are populations nested in latitude? Nested random effects occur when a lower
# level factor appears only within a particular level of an upper level factor.

# How good is the relative fit of the models?
AIC(m2, m3, m4)  # Note the AIC difference
```

```
##    df       AIC
## m2  7  15193.10
## m3  6  15199.14
## m4  5 395741.79
```

```r
summary(m2)
```

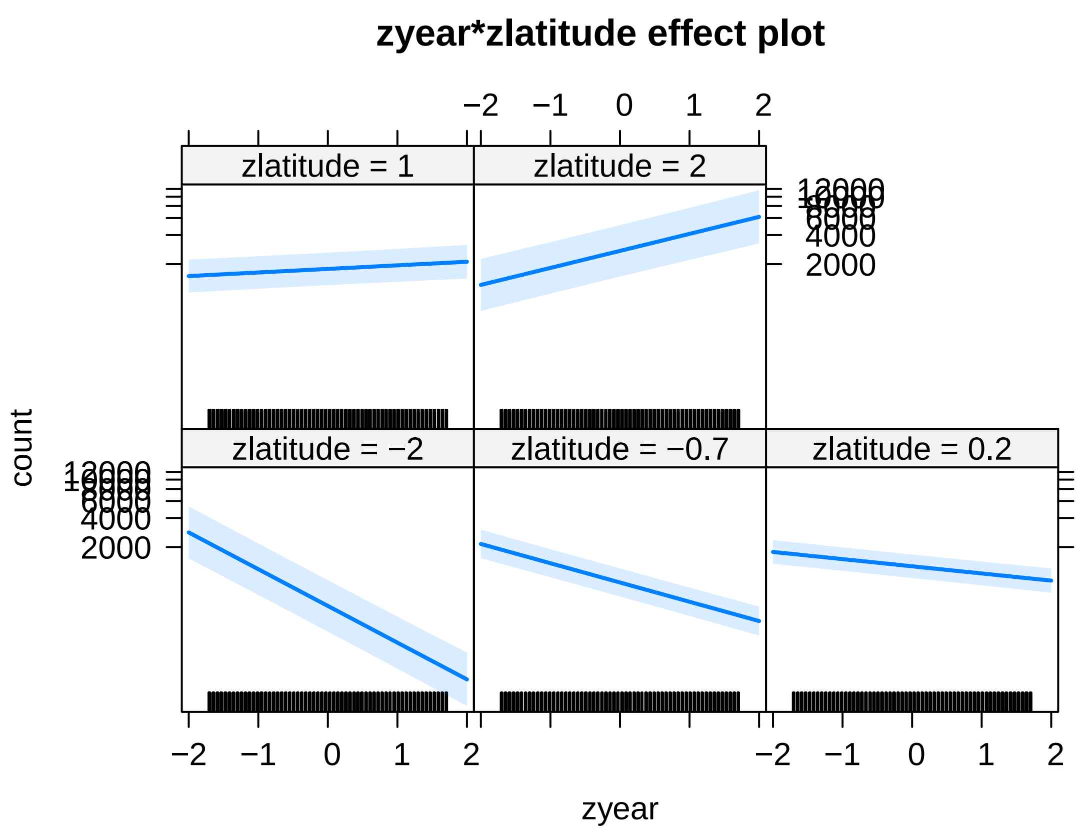
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: count ~ zyear * zlatitude + (zyear | site)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##  15193.1  15230.6  -7589.6  15179.1     1553
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6649 -0.4108  0.0011  0.4291  3.2434
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  site   (Intercept) 0.509685 0.71392
##         zyear       0.006321 0.07951  0.05
## Number of obs: 1560, groups:  site, 26
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      7.05742    0.14001  50.406  < 2e-16 ***
## zyear           -0.23504    0.01562 -15.051  < 2e-16 ***
## zlatitude        0.42972    0.14006   3.068  0.00215 **
## zyear:zlatitude  0.32035    0.01561  20.520  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) zyear  zlattd
## zyear         0.051
## zlatitude     0.000  0.000
## zyear:zlttd  0.000 -0.001  0.051
```

```
# significant zyear:zlatitude interaction indicates the count~year regression
# slope is positive when latitude is (more) positive,
# and negative when latitude is (more) negative
#library(effects)
ae = allEffects(m2)
```

```
## Warning in Analyze.model(focal.predictors, mod, xlevels, default.levels, : the
## predictors zyear, zlatitude are one-column matrices that were converted to
## vectors
```

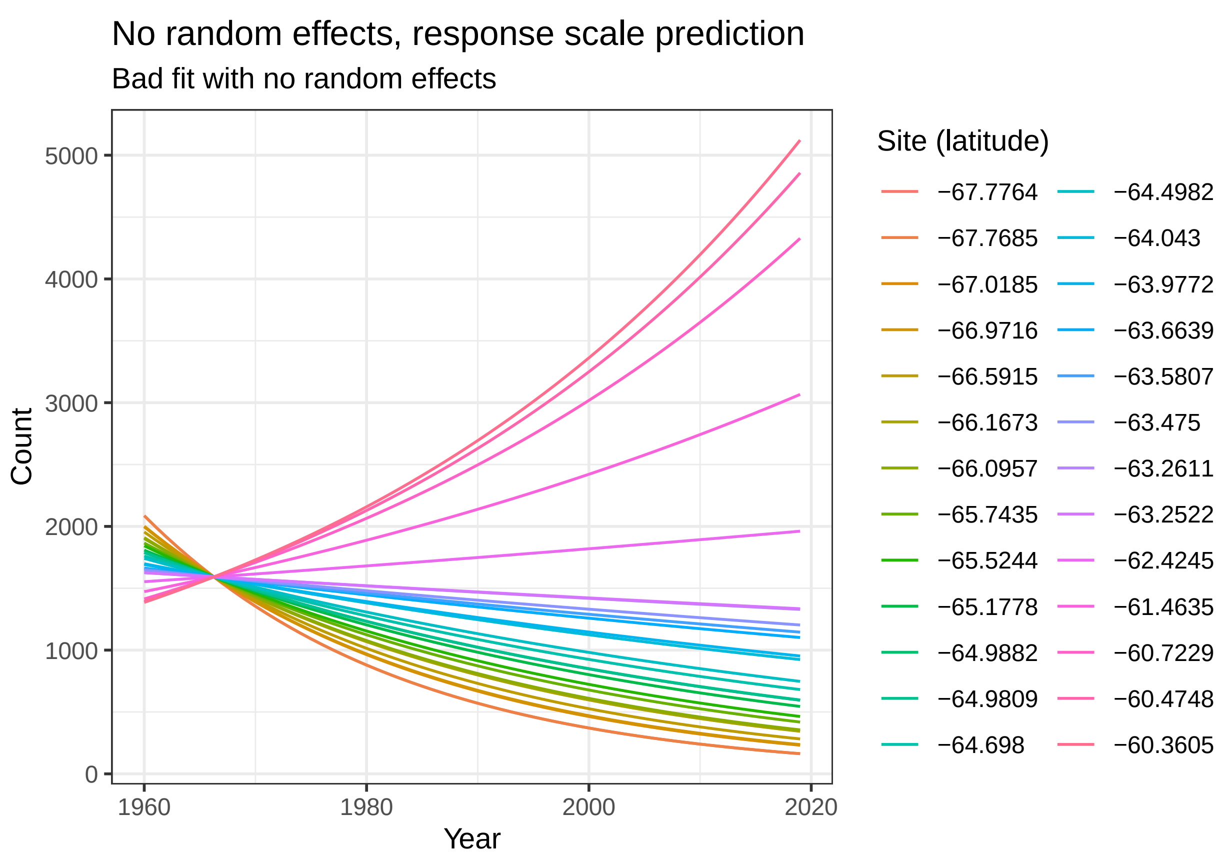
```
plot(ae)
```



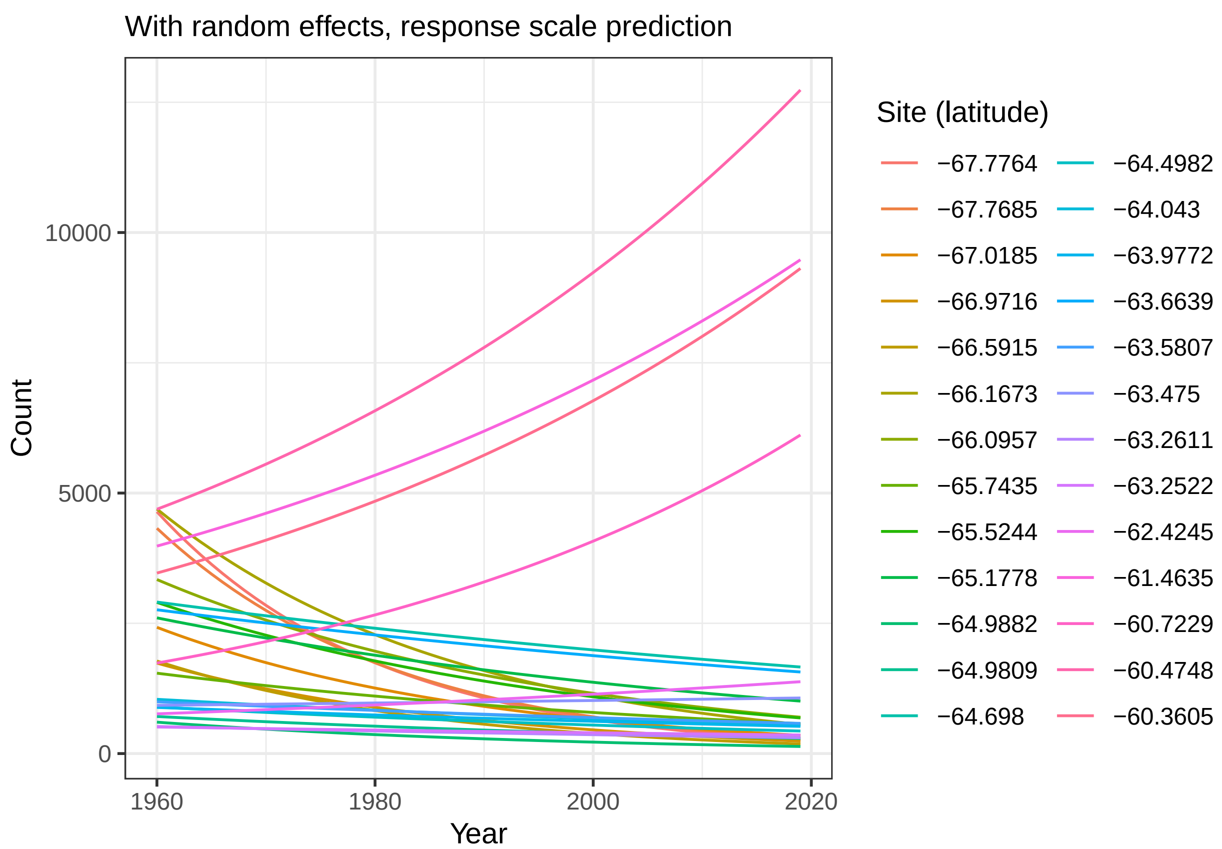
glmer (lme4) prediction

Prediction from m2

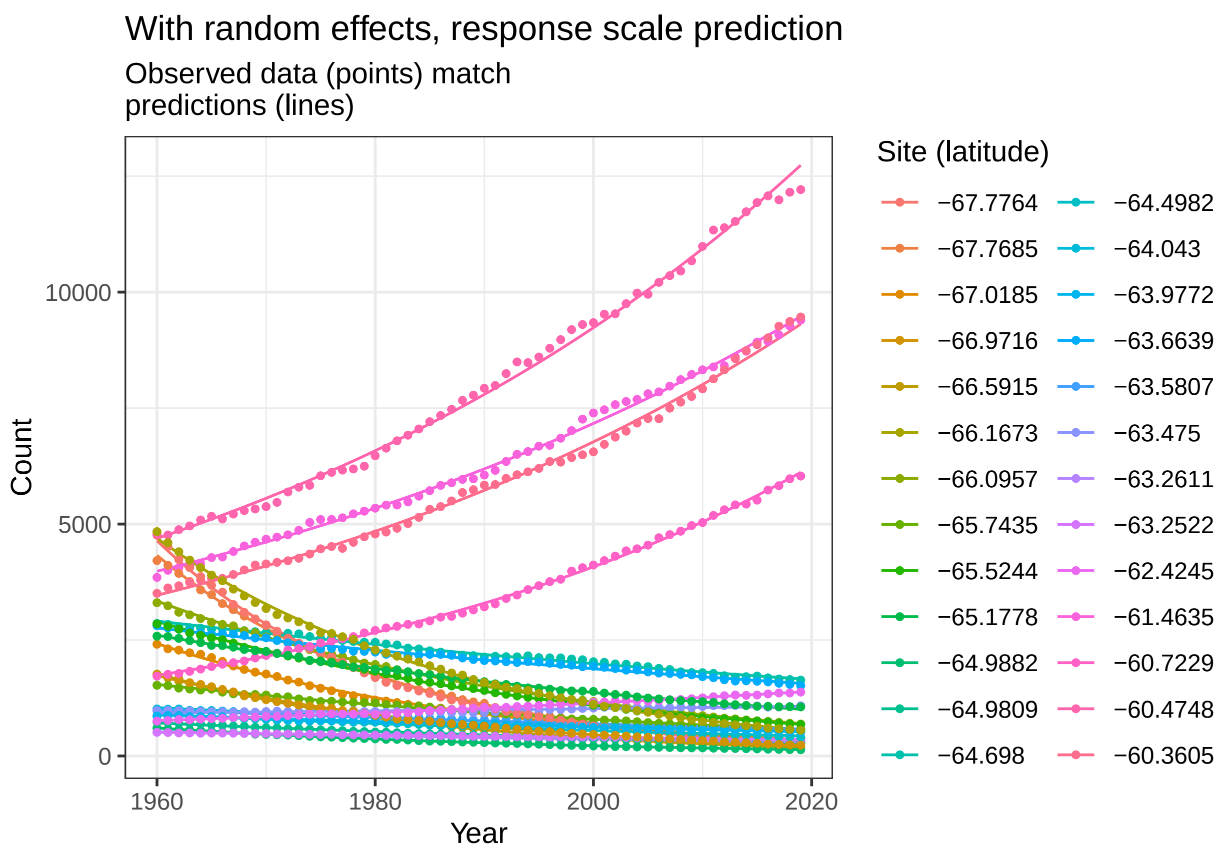
```
# glmer predict function: https://rdrr.io/cran/lme4/man/predict.merMod.html
# predict(object, newdata = NULL, newparams = NULL,
#         re.form = NULL, ReForm, REForm, REform,
#         random.only=FALSE, terms = NULL,
#         type = c("link", "response"), allow.new.levels = FALSE,
#         na.action = na.pass, ...)

# Predictions can be made with, or without, the contribution of random effects.
# The following plots shows the influence of this choice.

# no random effects, response scale prediction
df$fit.m2_norand <- predict(m2, df, re.form=NA, type = "response")

ggplot(df, aes(x = year, y = fit.m2_norand, color = as.factor(latitude))) +
  geom_line() + theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Site (latitude)")+
  labs(title = "No random effects, response scale prediction",
       subtitle = "Bad fit with no random effects")
```



```r
# with random effects, response scale
df$fit.m2 <- predict(m2, df, re.form=NULL, type = "response")

ggplot(df, aes(x = year, y = fit.m2, color = as.factor(latitude))) +
  geom_line() +  theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Site (latitude)") +
  labs(subtitle = "With random effects, response scale prediction")
```

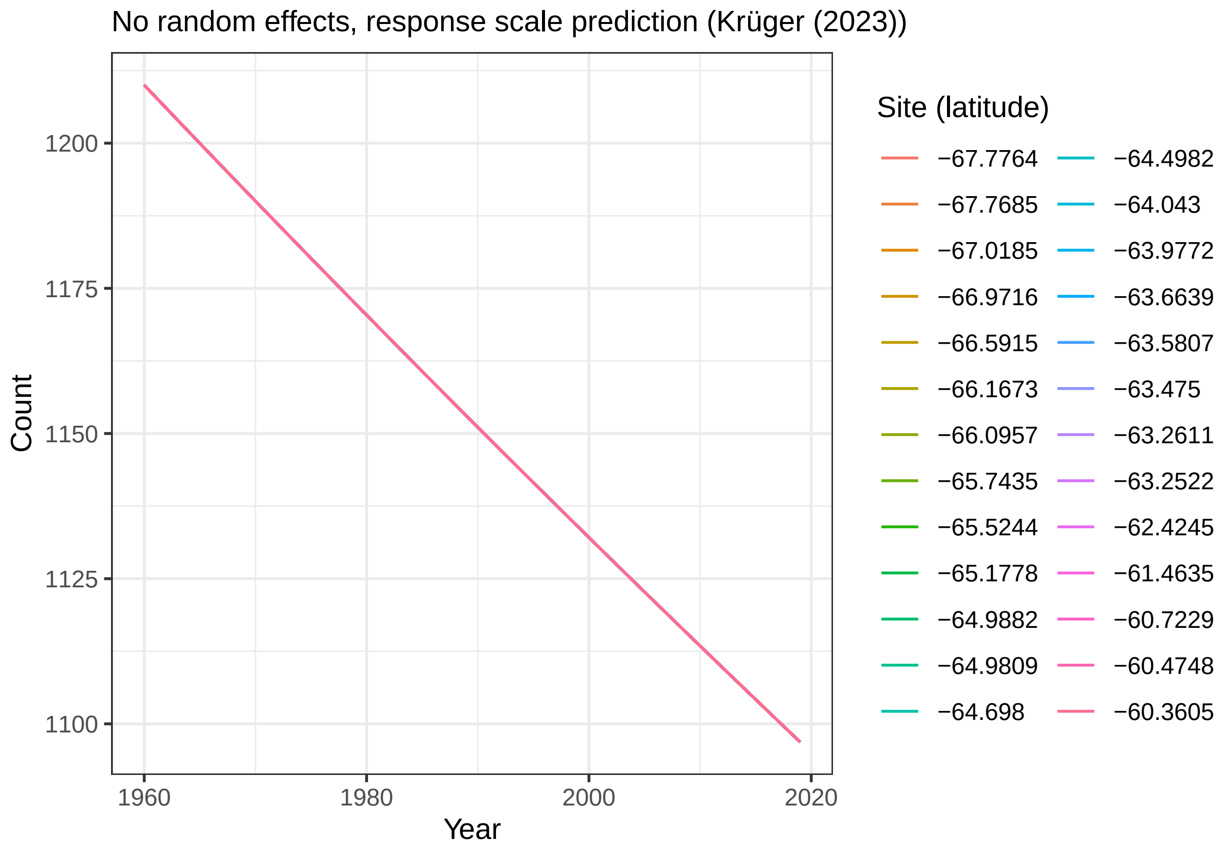



```r
# now add observed data
ggplot(df, aes(x = year, y = fit.m2, color = as.factor(latitude))) +
  geom_line() +  theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Site (latitude)") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9) +
  labs(title = "With random effects, response scale prediction",
       subtitle = "Observed data (points) match\npredictions (lines)")
```

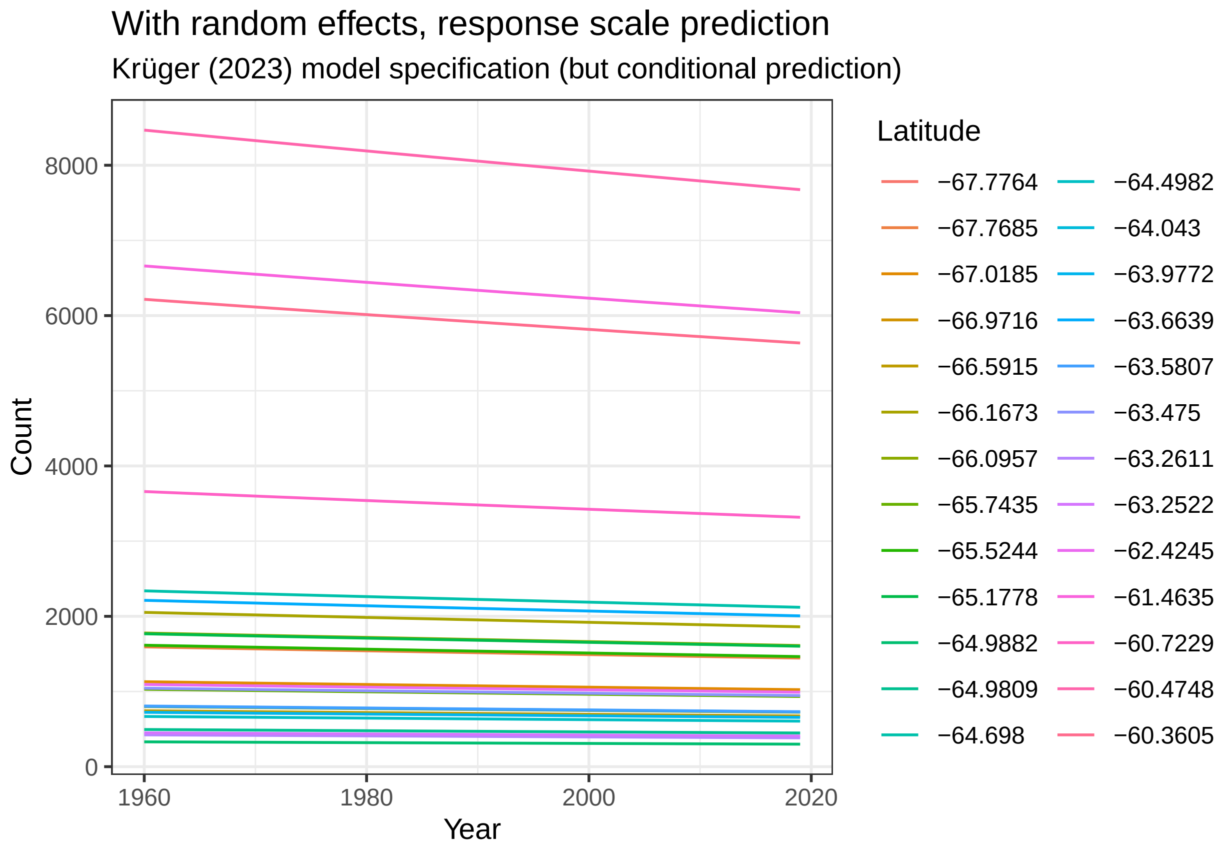


Prediction from m4 (GLMM syntax similar to Krüger 2023)

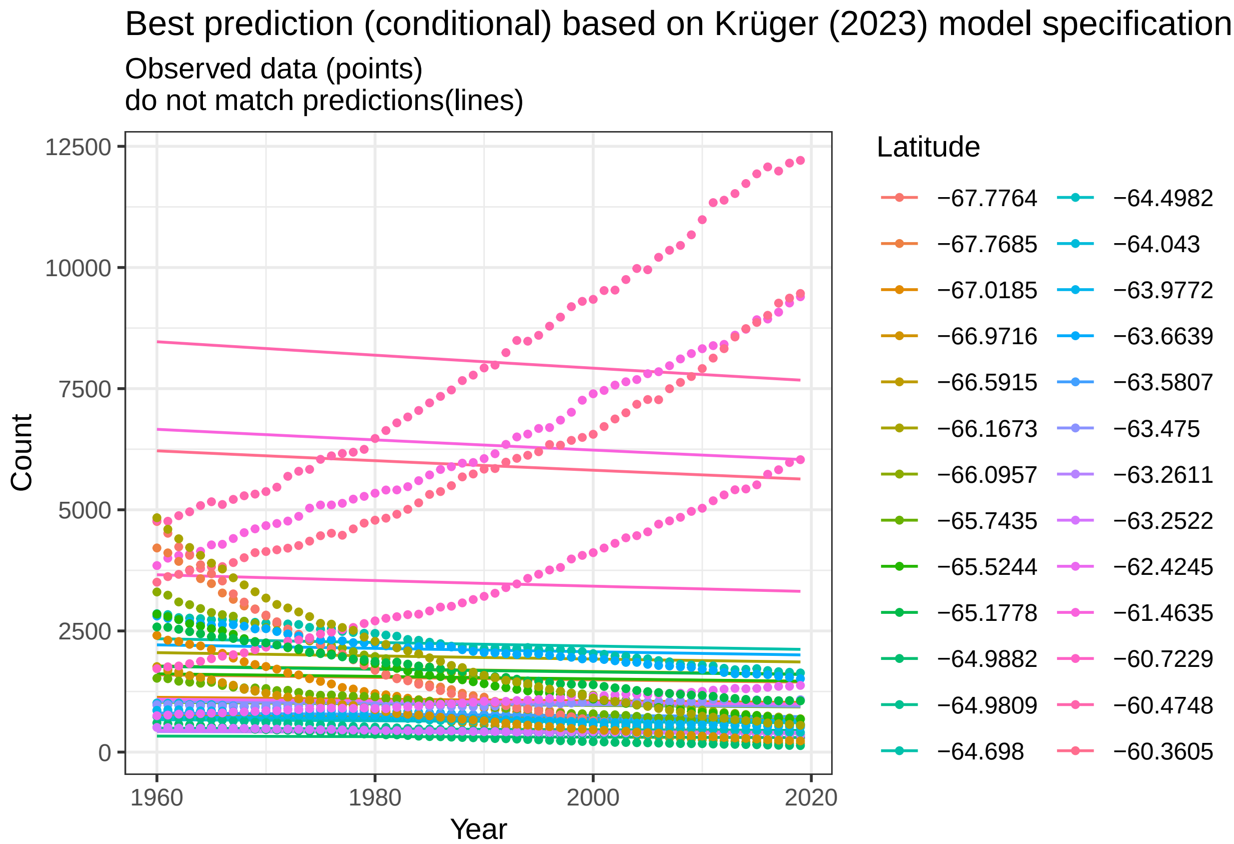
```
# no random effects, response scale prediction
df$fit.m4_norand <- predict(m4, df, re.form=NA, type = "response")

ggplot(df, aes(x = year, y = fit.m4_norand, color = as.factor(latitude))) +
  geom_line() + theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Site (latitude)") +
  labs(subtitle = paste0("No random effects, response scale prediction (Kr",ds4psy::Umlaut["u"], "ger (
```




```
# With random effects, response scale prediction
df$fit.m4 <- predict(m4, df, re.form=NULL, type = "response")

ggplot(df, aes(x = year, y = fit.m4, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  theme_bw() +
  labs(title = "With random effects, response scale prediction",
       subtitle = paste0("Kr",ds4psy::Umlaut["u"], "ger (2023) model specification (but conditional pred
```




```
# now add observed data
ggplot(df, aes(x = year, y = fit.m4, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(title = paste0("Best prediction (conditional) based on Kr",ds4psy::Umlaut["u"], "ger (2023) model
       subtitle = "Observed data (points)\ndo not match predictions(lines)")
```



MCMCglmm analysis of model m2

```
#-----------------------------------
# Repeat analysis with MCMCglmm
#-----------------------------------

#library(MCMCglmm)

df = as.data.frame(df) # a data frame is expected by MCMCglmm

# Use prior from Kruger (2023) (change 133 to 26)
prior <- list(R = list(V = 1, nu = 0.002),
              G = list(G1 = list(V = diag(2), nu = 0.002,
                                 alpha.mu = rep(0, 2),
                                 alpha.V= diag(26, 2, 2))))


# Fit model m2 from above running 40,000 iterations (~ 1 minute)
# Works with default prior or with prior specified above
mc2 <- MCMCglmm(count ~ zyear * zlatitude,
                random = ~us(1 + zyear):site,
                rcov=~units,
                family="poisson",
```

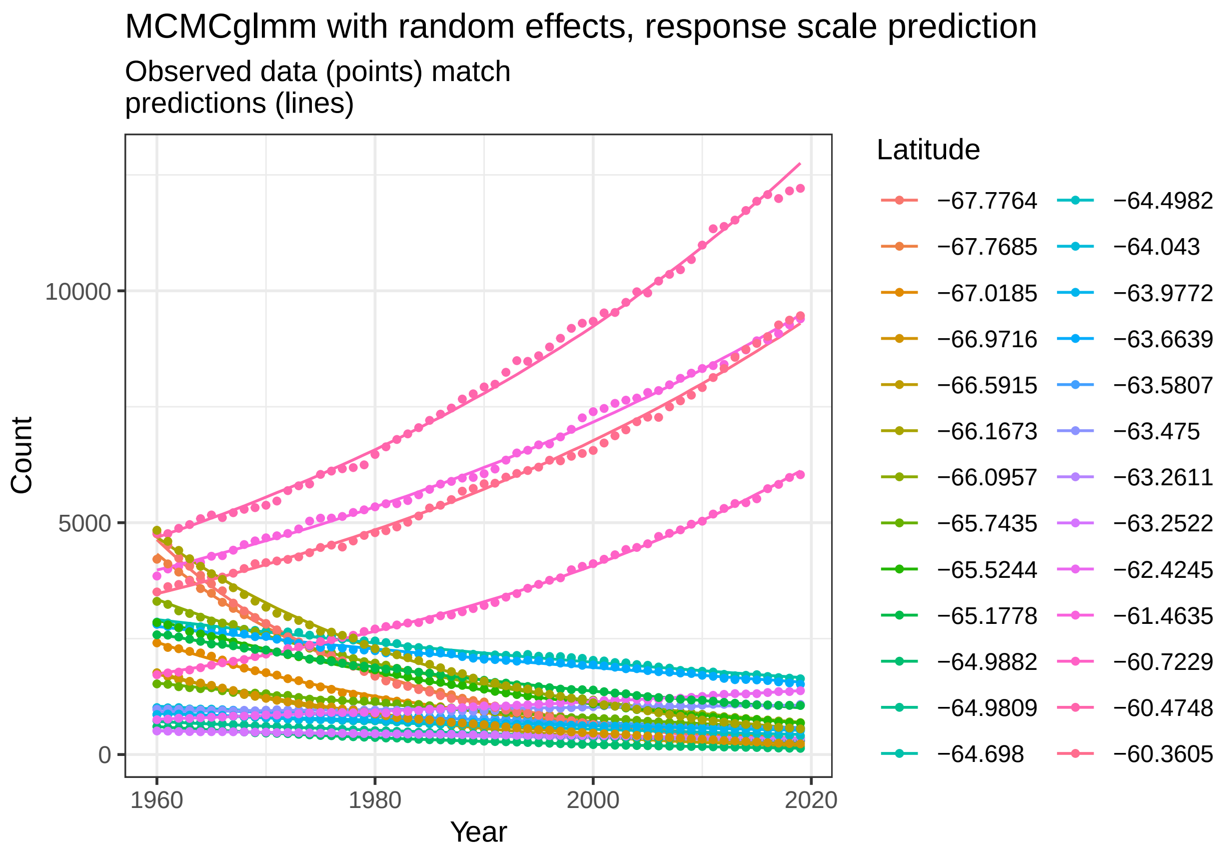
```
                 data = df,
                 mev=NULL,start=NULL,
                 # prior=NULL,
                 prior=prior,
                 nodes="ALL", scale=TRUE,
                 nitt=20000, thin=10, burnin=5000, pr=T,
                 pl=FALSE, verbose=FALSE, DIC=TRUE, singular.ok=FALSE, saveX=TRUE,
                 saveZ=TRUE, saveXL=TRUE, slice=FALSE, ginverse=NULL, trunc=FALSE)

summary(mc2)
```

```
##
##  Iterations = 5001:19991
##  Thinning interval  = 10
##  Sample size  = 1500
##
##  DIC: 14828.51
##
##  G-structure:  ~us(1 + zyear):site
##
##                              post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).site  0.634055  0.31215  0.99245     1500
## zyear:(Intercept).site        0.003924 -0.03058  0.03681     1500
## (Intercept):zyear.site        0.003924 -0.03058  0.03681     1500
## zyear:zyear.site              0.008164  0.00408  0.01373     1050
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI  u-95% CI eff.samp
## units 6.189e-05 4.486e-05 7.878e-05    114.8
##
##  Location effects: count ~ zyear * zlatitude
##
##                 post.mean l-95% CI u-95% CI eff.samp   pMCMC
## (Intercept)        7.0533   6.7297   7.3508     1662 < 7e-04 ***
## zyear             -0.2361  -0.2690  -0.2009     1500 < 7e-04 ***
## zlatitude          0.4299   0.1380   0.7316     1500 0.00667 **
## zyear:zlatitude    0.3200   0.2848   0.3549     1390 < 7e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Predictions with MCMCglmm mc2

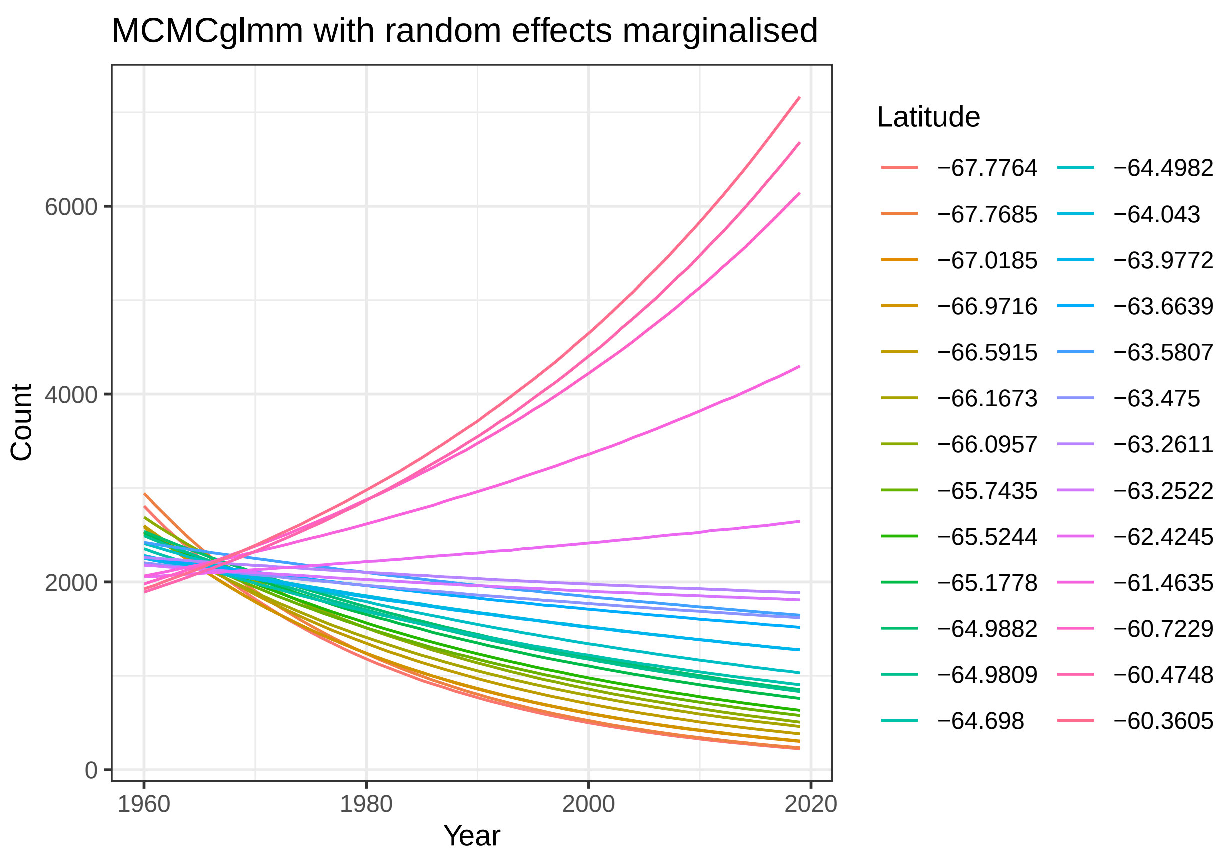
```
# https://www.rdocumentation.org/packages/MCMCglmm/versions/2.34/topics/predict.MCMCglmm
#predict(object, newdata=NULL, marginal=object$Random$formula,
#        type="response", interval="none", level=0.95, it=NULL,
#        posterior="all", verbose=FALSE, approx="numerical", ...)

# marginal = formula defining random effects to be maginalised.
# You don't want the random effects maginalised, so need to set this to NULL.
```

```r
# First do a prediction with random effects:
pred <- data.frame(predict(mc2,
                           newdata=df,
                           type="response",
                           marginal=NULL,
                           interval="prediction",
                           posterior="mean"))  # "all" is better, but Kruger used 'mean'

df$fit.mc2 <- pred$fit
df$lwr.mc2 <- pred$lwr
df$upr.mc2 <- pred$upr

# predict with observed data
ggplot(df, aes(x = year, y = fit.mc2, color = as.factor(latitude))) +
  geom_line() +  theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  labs(title = "MCMCglmm with random effects, response scale prediction",
       subtitle = "Observed data (points) match\npredictions (lines)")
```

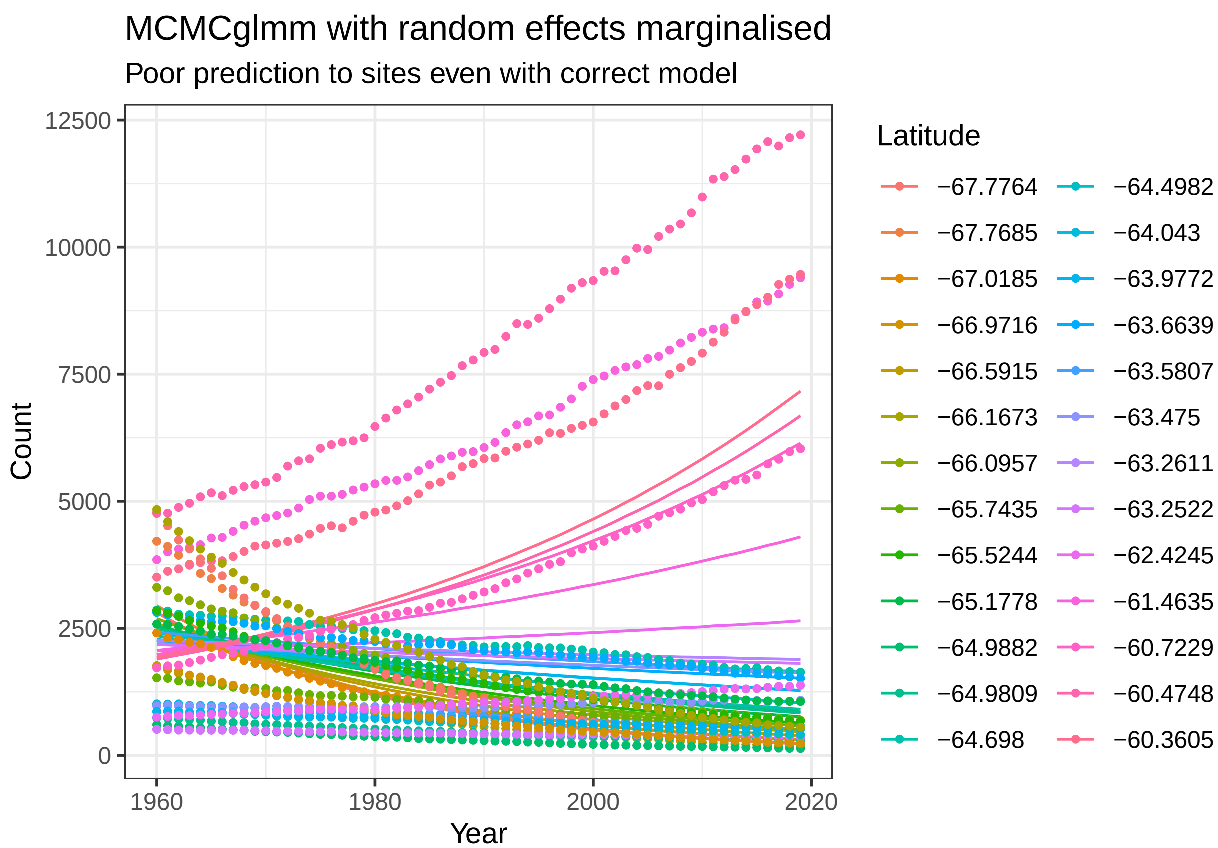



```r
# Kruger (2023) maginalised the random effects.
# Let's also do that to show what the impact is on this data set.
# Now predict with maginalised random effects (Kruger (2023)):
```

```r
pred_margin <- data.frame(predict(mc2,
                          newdata=df,
                          type="response",
                          marginal=mc2$Random$formula,
                          interval="prediction",
                          posterior="mean"))  # "all" is better

df$fit.mc2_margin <- pred_margin$fit

ggplot(df, aes(x = year, y = fit.mc2_margin, color = as.factor(latitude))) +
  geom_line() +   theme_bw() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  labs(title = "MCMCglmm with random effects marginalised")
```




```r
ggplot(df, aes(x = year, y = fit.mc2_margin, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(title = "MCMCglmm with random effects marginalised",
       subtitle = "Poor prediction to sites even with correct model")
```



MCMCglmm analysis (Krüger 2023 model)

```
# Now run MCMCglmm using Kruger (2023) model formulation

mc_Kr <- MCMCglmm(count ~ zyear,
              random = ~us(1+zlatitude):site,
              rcov=~units,
              family="poisson",
              data = df,
              mev=NULL,start=NULL,
              # prior=NULL,
              prior=prior,
              nodes="ALL", scale=TRUE,
              nitt=20000,
              thin=10,
              burnin=5000,
              pr=T,
              pl=FALSE, verbose=FALSE, DIC=TRUE, singular.ok=FALSE, saveX=TRUE,
              saveZ=TRUE, saveXL=TRUE, slice=FALSE, ginverse=NULL, trunc=FALSE)

# Compare the effective sample sizes between mc2 and mc_Kr
summary(mc2)
```

```
##
##  Iterations = 5001:19991
##  Thinning interval  = 10
##  Sample size  = 1500
##
##  DIC: 14828.51
##
##  G-structure:  ~us(1 + zyear):site
##
##                              post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).site  0.634055  0.31215  0.99245     1500
## zyear:(Intercept).site        0.003924 -0.03058  0.03681     1500
## (Intercept):zyear.site        0.003924 -0.03058  0.03681     1500
## zyear:zyear.site              0.008164  0.00408  0.01373     1050
##
##  R-structure:  ~units
##
##       post.mean  l-95% CI  u-95% CI eff.samp
## units 6.189e-05 4.486e-05 7.878e-05    114.8
##
##  Location effects: count ~ zyear * zlatitude
##
##                 post.mean l-95% CI u-95% CI eff.samp   pMCMC
## (Intercept)        7.0533   6.7297   7.3508     1662 < 7e-04 ***
## zyear             -0.2361  -0.2690  -0.2009     1500 < 7e-04 ***
## zlatitude          0.4299   0.1380   0.7316     1500 0.00667 **
## zyear:zlatitude    0.3200   0.2848   0.3549     1390 < 7e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mc_Kr)
```

```
##
##  Iterations = 5001:19991
##  Thinning interval  = 10
##  Sample size  = 1500
##
##  DIC: 16973.07
##
##  G-structure:  ~us(1 + zlatitude):site
##
##                              post.mean l-95% CI u-95% CI eff.samp
## (Intercept):(Intercept).site    0.6219 0.287928   1.0786   1044.5
## zlatitude:(Intercept).site      0.2925 0.076589   0.6130    502.9
## (Intercept):zlatitude.site      0.2925 0.076589   0.6130    502.9
## zlatitude:zlatitude.site        0.1655 0.001637   0.3804    398.2
##
##  R-structure:  ~units
##
##       post.mean l-95% CI u-95% CI eff.samp
## units    0.1104   0.1023   0.1179     1713
##
##  Location effects: count ~ zyear
##
```

```
##             post.mean l-95% CI u-95% CI eff.samp  pMCMC
## (Intercept)    6.9104   6.6583   7.1080     1500 <7e-04 ***
## zyear         -0.2346  -0.2500  -0.2181     1500 <7e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

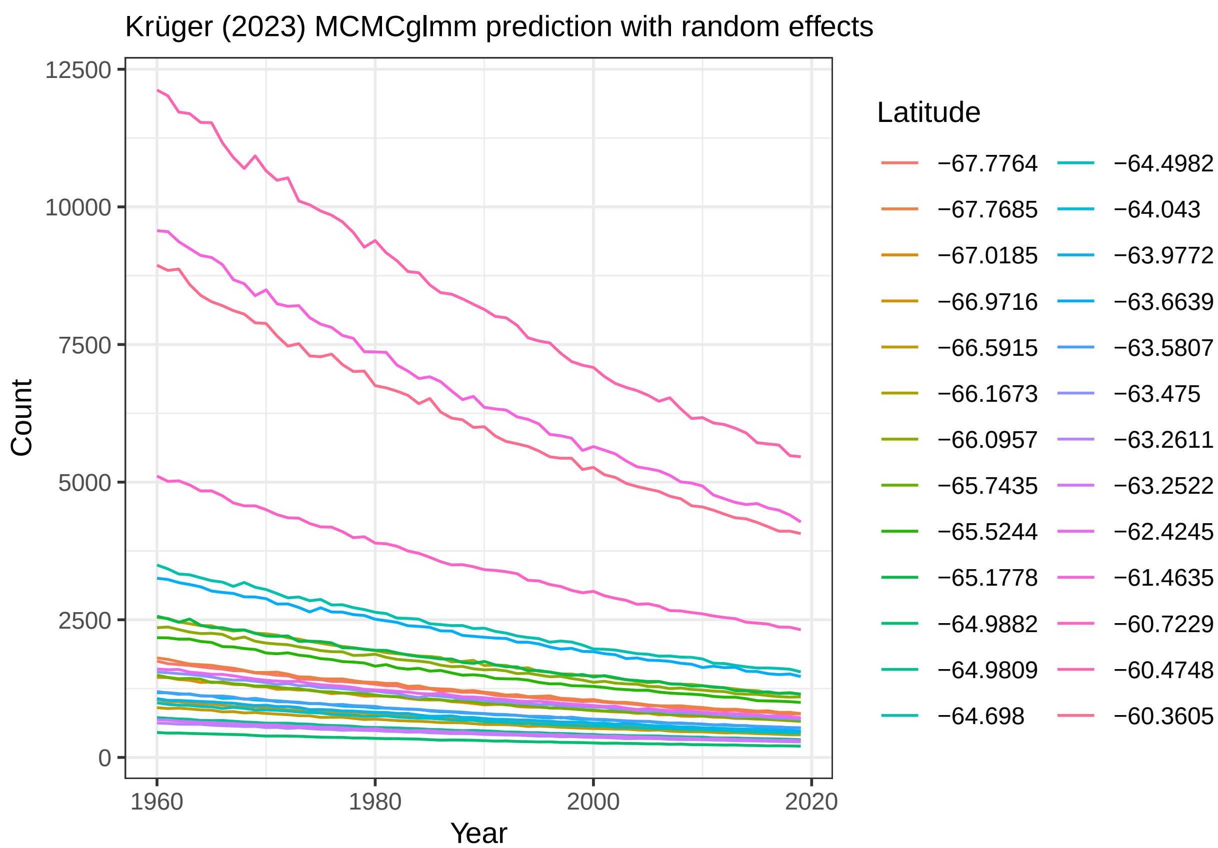
```
# The random effect structure of the Kruger model has MCMC sampling problems
coda::effectiveSize(mc2$VCV)
```

```
## (Intercept):(Intercept).site       zyear:(Intercept).site
##                    1500.0000                    1500.0000
##       (Intercept):zyear.site             zyear:zyear.site
##                    1500.0000                    1050.1442
##                        units
##                     114.7672
```

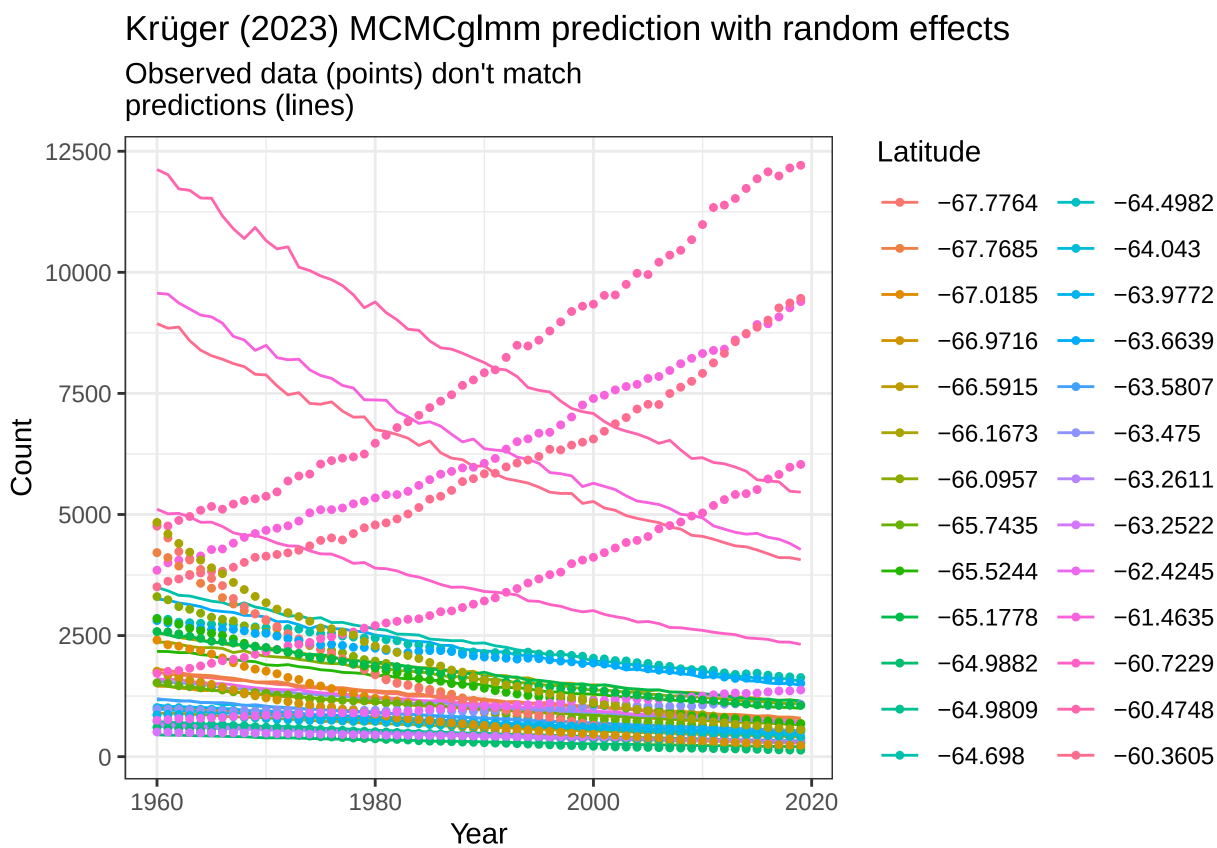
```
coda::effectiveSize(mc_Kr$VCV)
```

```
## (Intercept):(Intercept).site   zlatitude:(Intercept).site
##                    1044.5172                     502.8578
##   (Intercept):zlatitude.site     zlatitude:zlatitude.site
##                     502.8578                     398.2386
##                        units
##                    1713.0864
```

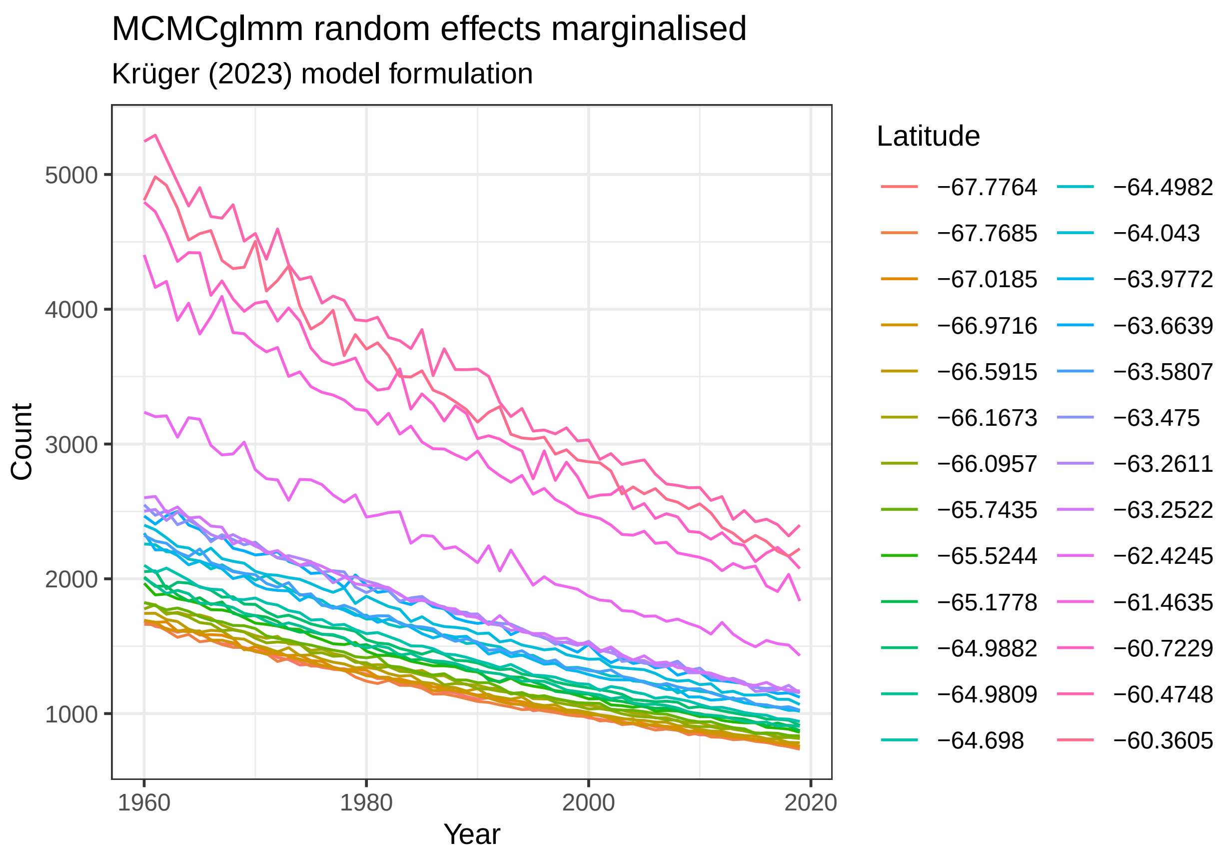
```
# Predict from model akin to Kruger (2023) MCMCglmm, using random effects
pred_Kr <- data.frame(predict(mc_Kr,
                  newdata=df,
                  type="response",
                  marginal=NULL,
                  interval="prediction",
                  posterior="mean"))  # "all" is better, but Kruger used 'mean'

df$fit.mc_Kr <- pred_Kr$fit
df$lwr.mc_Kr <- pred_Kr$lwr
df$upr.mc_Kr <- pred_Kr$upr

# Without observed data
ggplot(df, aes(x = year, y = fit.mc_Kr, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
 # geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(subtitle = paste0("Kr",ds4psy::Umlaut["u"], "ger (2023) MCMCglmm prediction with random effects")
```



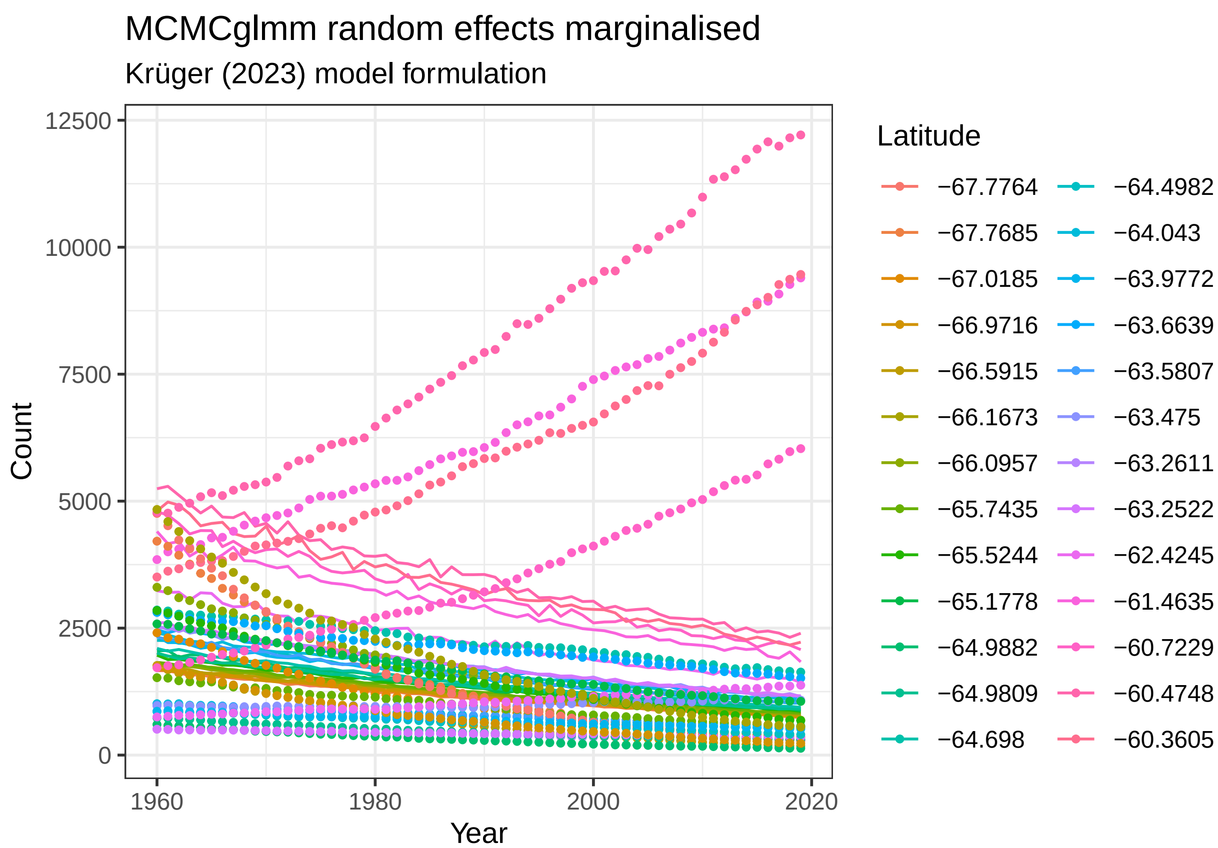

```
# Add observed data
ggplot(df, aes(x = year, y = fit.mc_Kr, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(title = paste0("Kr",ds4psy::Umlaut["u"], "ger (2023) MCMCglmm prediction with random effects"),
       subtitle = "Observed data (points) don't match\npredictions (lines)")
```




```r
# Now predict with maginalised random effects (Kruger (2023)):
# This is the prediction used by Kruger (2023)
pred_margin_Kr <- data.frame(predict(mc_Kr,
                                newdata=df,
                                type="response",
                                marginal=mc_Kr$Random$formula,
                                interval="prediction",
                                posterior="mean"))  # "all" is better

df$fit.mc_Kr_margin <- pred_margin_Kr$fit

ggplot(df, aes(x = year, y = fit.mc_Kr_margin, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
 # geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(title = "MCMCglmm random effects marginalised",
       subtitle = paste0("Kr",ds4psy::Umlaut["u"], "ger (2023) model formulation"))
```



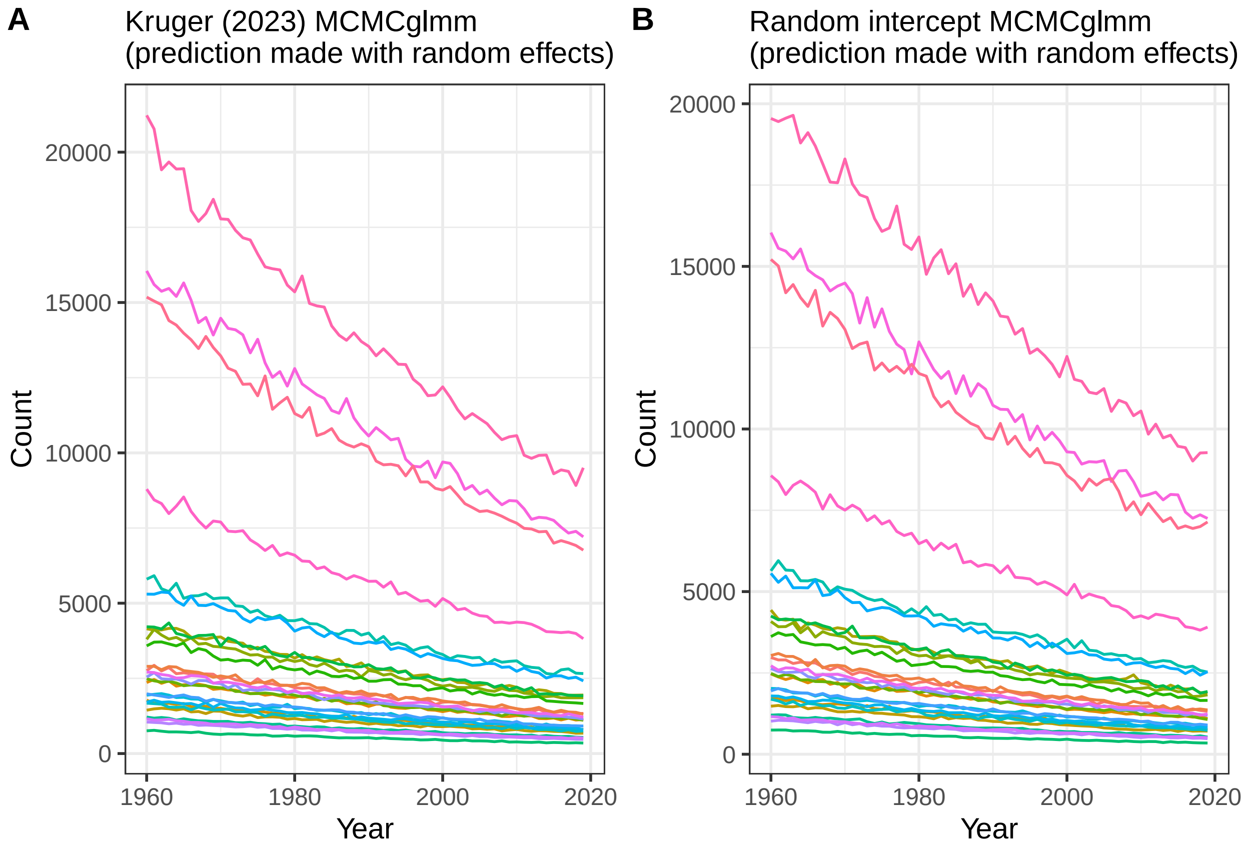

```r
# add observed data
ggplot(df, aes(x = year, y = fit.mc_Kr_margin, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +
  labs(title = "MCMCglmm random effects marginalised",
       subtitle = paste0("Kr",ds4psy::Umlaut["u"], "ger"," (2023) model formulation"))
```



Run intercept-only MCMCglmm model to show equivalency to Krüger (2023) model

```
# Now run MCMCglmm intercept only model, using the default prior

mc_I <- MCMCglmm(count ~ zyear,
                  random = ~site,
                  rcov=~units,
                  family="poisson",
                  data = df,
                  mev=NULL,start=NULL,
                  prior=NULL,
                  #prior=prior,
                  nodes="ALL", scale=TRUE,
                  nitt=20000,
                  thin=10,
                  burnin=5000,
                  pr=T,
                  pl=FALSE, verbose=FALSE, DIC=TRUE, singular.ok=FALSE, saveX=TRUE,
                  saveZ=TRUE, saveXL=TRUE, slice=FALSE, ginverse=NULL, trunc=FALSE)


# Predict from intercept model (include random site)
pred_I <- data.frame(predict(mc_I,
```

```
                              newdata=df,
                              type="response",
                              marginal=NULL,
                              interval="prediction",
                              posterior="mean"))

df$fit.mc_I <- pred_I$fit
df$lwr.mc_I <- pred_I$lwr
df$upr.mc_I <- pred_I$upr

# figure (from above) with Lat as a random effect
lat_id = ggplot(df, aes(x = year, y = upr.mc_Kr, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  # geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +  theme(legend.position="none") +
  labs(subtitle = "Kruger (2023) MCMCglmm\n(prediction made with random effects)")

# Intercept only model
interc.m = ggplot(df, aes(x = year, y = upr.mc_I, color = as.factor(latitude))) +
  geom_line() +
  labs(x = "Year", y = "Count") +
  scale_color_discrete(name = "Latitude") +
  # add observed data
  # geom_point(aes(x = year, y = count, color = as.factor(latitude)), size = 0.9)+
  theme_bw() +  theme(legend.position="none") +
  labs(subtitle = "Random intercept MCMCglmm\n(prediction made with random effects)")

cowplot::plot_grid(lat_id, interc.m, labels = c('A', 'B'), label_size = 12)
```



The two models produce the same estimates for 'fit'.