

# Practise\_FII\_Modelling

Patrick Alexander Walkden

25/02/2021

## Functional Intactness Index Model

To model FII I am using a two stage modeling framework. The first model will take site functional diversity, measured as Rao's Q, as function of land-use type and intensity. The second model will focus on functional similarity between sites of primary minimal habitat and all other land use types. Together this is will get a measure of overall functional diversity and the proportion of that function that is retained from primary minimal sites, which once projected on to current maps of the modeled pressures and multiplied together will (hopefully) give a measure of functional intactness.

### 1.1 Modelling: Functional Diversity (PGLMM)

Before modeling I considered whether sites that are more phylogenetically related to each other may respond similarly to landuse change and therefore may introduce non-independence into the modeling that would need to accounted for. To do so I first calculated pairwise phylogenetic similarity of sites with unifracs (the percentage of shared branch length between two communities). Using a clustering algorithm within `hclust` that does not assume an ultrametric structure we are able to see whether the sites are clustered phylogenetically and to what extent they form a heirachial structure. A high amount of clustering would indicate that there are sites that are a lot more similar to each other than expected if they are to be independent and therefore could show correlated responses. Low clustering would show that sites aren't that phylogenetically similar to each other anyway and therefore it is not a large problem to consider.

Secondly, I checked whether the responses to landuse within study showed phylogenetic non-independence within a model that does not consider phylogenetic relatedness. I did this by calculating the differences between random slopes within studies and then identifying whether studies that are more phylogenetically similar to each other (as calculated by unifracs) have significantly more similar slopes than the global distribution of slopes. Doing this iteratively from the closest related study to the most distant I can plot the probability of rejecting the null hypothesis of that there is no differences between the similarity of slopes. A clear trend from rejecting the null hypothesis in the most related studies to accepting the null hypothesis in the most distantly related studies will be indicative of non-independence.

I am using community phylogenetic generalized linear mixed effects models (PGLMM) using the package `phyr`. `Phyr` allows for bayesian methods of model fitting using INLA, which is advised with larger datasets. I also considered the UN-subregion, human population density ( $\ln + 1$  transformed) and density of roads at both a 1km and 50km radius surrounding the site. I included study and study block as additional random effects and tested for the inclusion of random slopes of the other pressures within study by comparing the DIC of the maximal models. The fixed effect structure was then determined through backwards stepwise model simplification by looking at whether the removal of fixed effects significantly reduced the DIC of the model.

```
rm(list = ls())
require(phytools)
require(picante)
require(phyr)
require(tidyverse)
```

```
require(raster)
require(ggbridges)
require(motmot)
require(lmerTest)
require(poolr)
require(car)
require(robustlmm)
require(ggResidpanel)
require(GGally)
require(broom.helpers)
```

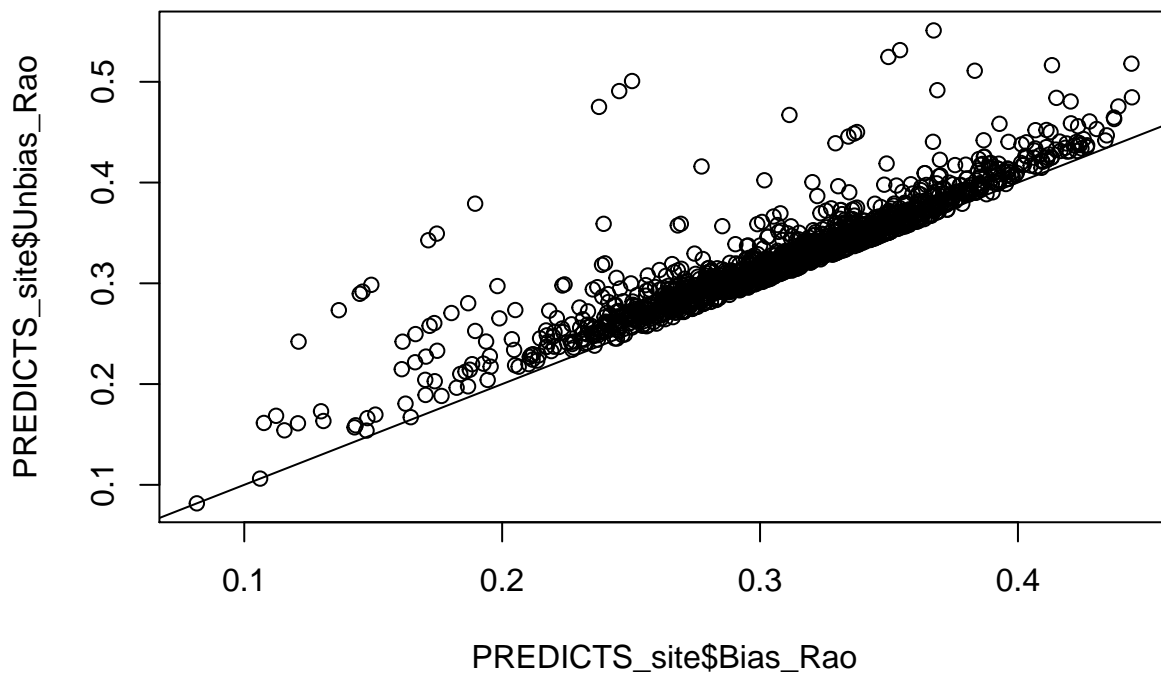
Load in the datasets from the previously calculated metrics.

```
## Load in the datasets
```

```
PREDICTS_site <- readRDS("../Functional_Intactness_Index/Outputs/PREDICTS_Site_Rao.rds")
PREDICTS_abundance <- readRDS("../Functional_Intactness_Index/Outputs/abundance_data.rds")
```

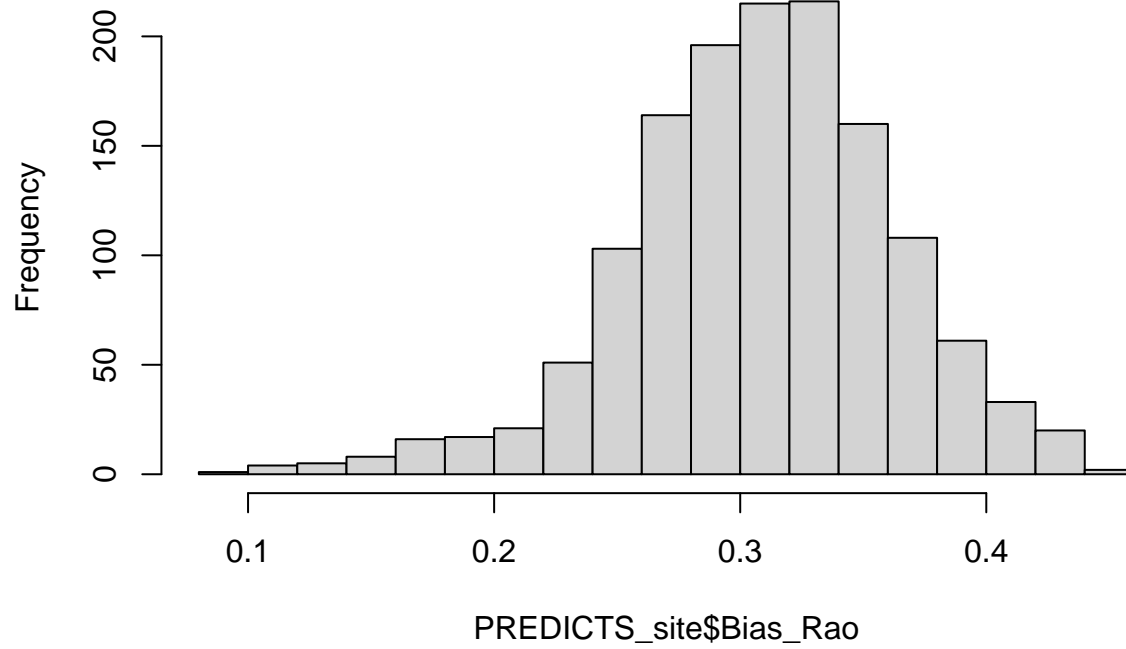
I calculated Rao's Q based on pairwise functional distances of species within a site weighted by their relative abundance. Chen et al, 2018 highlighted that the Rao estimator might be inherently bias underestimating the true value of functional diversity and proposed an unbiased metric. I have calculated both.

```
plot(PREDICTS_site$Unbias_Rao ~ PREDICTS_site$Bias_Rao)
abline(a=0, b=1)
```



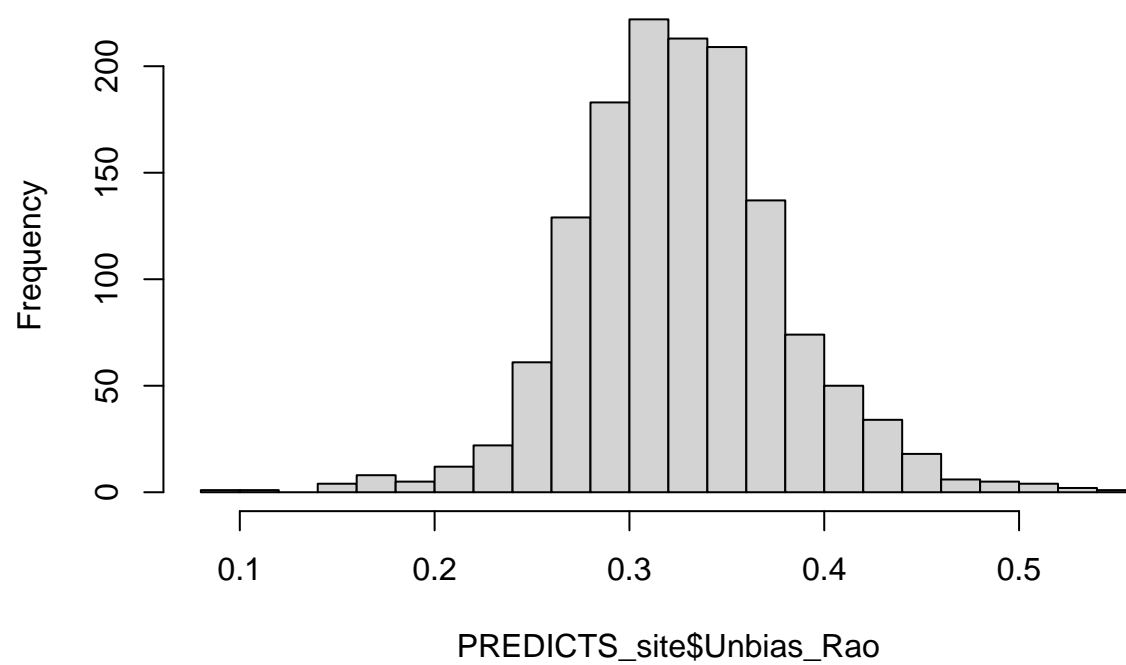
```
hist(PREDICTS_site$Bias_Rao, breaks = 20)
```

**Histogram of PREDICTS\_site\$Bias\_Rao**



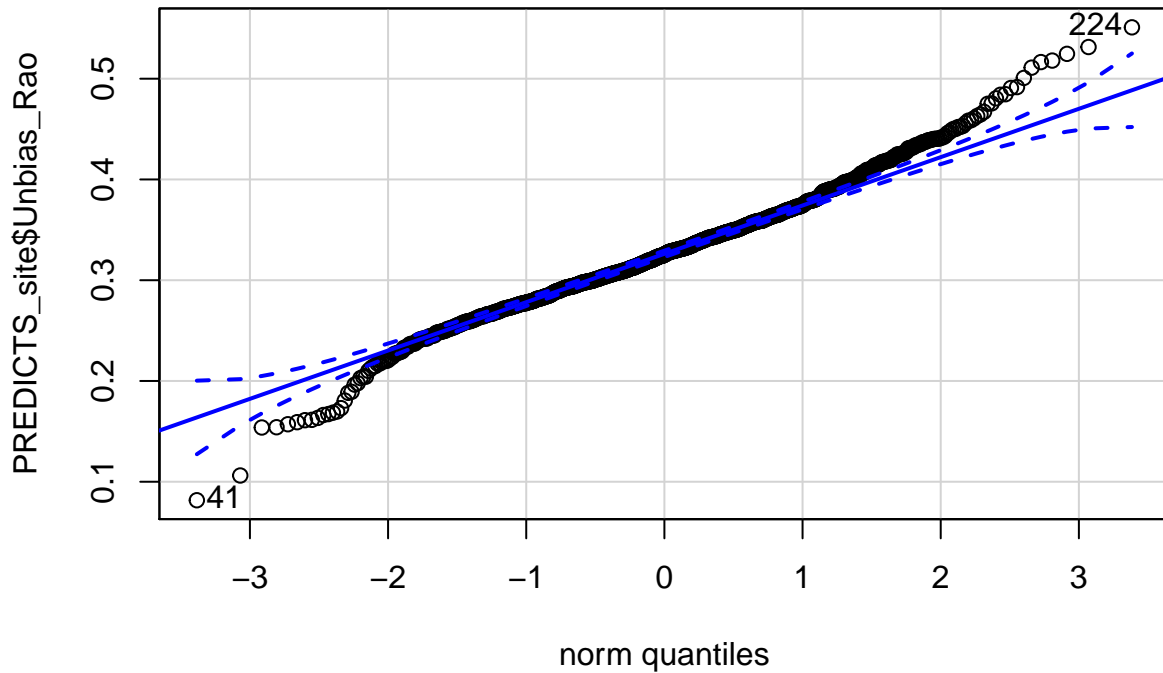
```
hist(PREDICTS_site$Unbias_Rao, breaks = 20)
```

**Histogram of PREDICTS\_site\$Unbias\_Rao**



### Outlier check

```
qqPlot(PREDICTS_site$Unbias_Rao)
```



```
## [1] 41 224
```

QQ plot has identified DI1\_2010\_\_Milder 2 41 and GN1\_2007\_\_VanBael 1 9 64 as potential outliers. Looking at the makeup of the sites it seems that DI1\_2010\_\_Milder 2 41 is completely dominated by a single species *Dives dives* and from that I noticed DI1\_2010\_\_Milder 2 48 is also dominated by *Bubulcus ibis* as well which is really skewing the calculation of the estimator. I looked at the original study published by Milder et al 2010 and it looks like these large values are due to large flocks of birds being counted within the site. There could be a couple of ways to deal with these influential results due to a single species dominating a site and therefore skewing the FD metric, first we could reduced the influence of the species in the calculation of Rao's Q (by what means I'm unsure - possibly introduce a maximum threshold for the number of individuals of a single species that can occur within a site?) or reduce the influence of the final data point at the modelling stage - this can be done by producing "robust" estimates of the final model using robust linear mixed effects models that downweight points that are considered to be outliers. For now I have gone with the latter.

With GN1\_2007\_\_VanBael 1 9 64 I don't know whether to drop it may be a true value of biological relevance.

Next we'd like to have a look at how the distribution of land use types and intensities we have across our modeled sites.

```
table(PREDICTS_site$LandUse, PREDICTS_site$Use_intensity)
```

```
##
##           Minimal use Light use Intense use
## Cropland           33          38           0
## Pasture            28          34           60
## Plantation forest  119          36           0
## Primary           378         308          29
```

##	Secondary Vegetation	125	47	17
##	Urban	93	56	0

Although Plantation forest exists as a land use class within PREDICTS it is rarely incorporated in global land-use layers therefore I included Plantation minimally used habitat in secondary vegetation light use and Plantation Light use in secondary intense use following De Palma et al, in Review. Because we are collapsing some land use classes and intensities in on each other I have calculated another factor or LUI

*#### collapse Plantation forest into secondary vegetation and releve the LUI variable to have Primary*

```
PREDICTS_site$SSBS <- factor(PREDICTS_site$SSBS) %>% droplevels()

PREDICTS_site <- PREDICTS_site %>% dplyr::mutate(LandUse_Intensity = ifelse(grepl(LandUse_Intensity, pattern = "Secondary Vegetation_Light use",
LandUse_Intensity = ifelse(grepl(LandUse_Intensity, pattern = "Secondary Vegetation_Intense use",
LandUse_Intensity = releve(factor(LandUse_Intensity),
ref = "Primary_Minimal use"))

table(PREDICTS_site$LandUse_Intensity)
```

##		
##	Primary_Minimal use	Cropland_Light use
##	378	38
##	Cropland_Minimal use	Pasture_Intense use
##	33	60
##	Pasture_Light use	Pasture_Minimal use
##	34	28
##	Primary_Intense use	Primary_Light use
##	29	308
##	Secondary Vegetation_Intense use	Secondary Vegetation_Light use
##	53	166
##	Secondary Vegetation_Minimal use	Urban_Light use
##	125	56
##	Urban_Minimal use	
##	93	

*##### Going to rename for some ease of outputs*

```
PREDICTS_site <- PREDICTS_site %>% dplyr::rename(LUI = LandUse_Intensity)
levels(PREDICTS_site$LUI) <- c("PriMin", "CrpLig", "CrpMin", "PasIn", "PasLig",
"PasMin", "PriIn", "PriLig", "SecIn", "SecLig", "SecMin", "UrbLig", "UrbMin")
```

*##### Then if the modelling of LUI is not sufficient there is also a LandUse factor that has plan*

```
PREDICTS_site <- PREDICTS_site %>% dplyr::mutate(LandUse = ifelse(LUI == "PriMin", "PriMin", paste(LandUse, "Min", sep = " ")),
LandUse = ifelse(LandUse == "Plantation forest", "Secondary Vegetation", "Plantation forest"),
LandUse = releve(factor(LandUse), ref = "PriMin"))

table(PREDICTS_site$LUI)
```

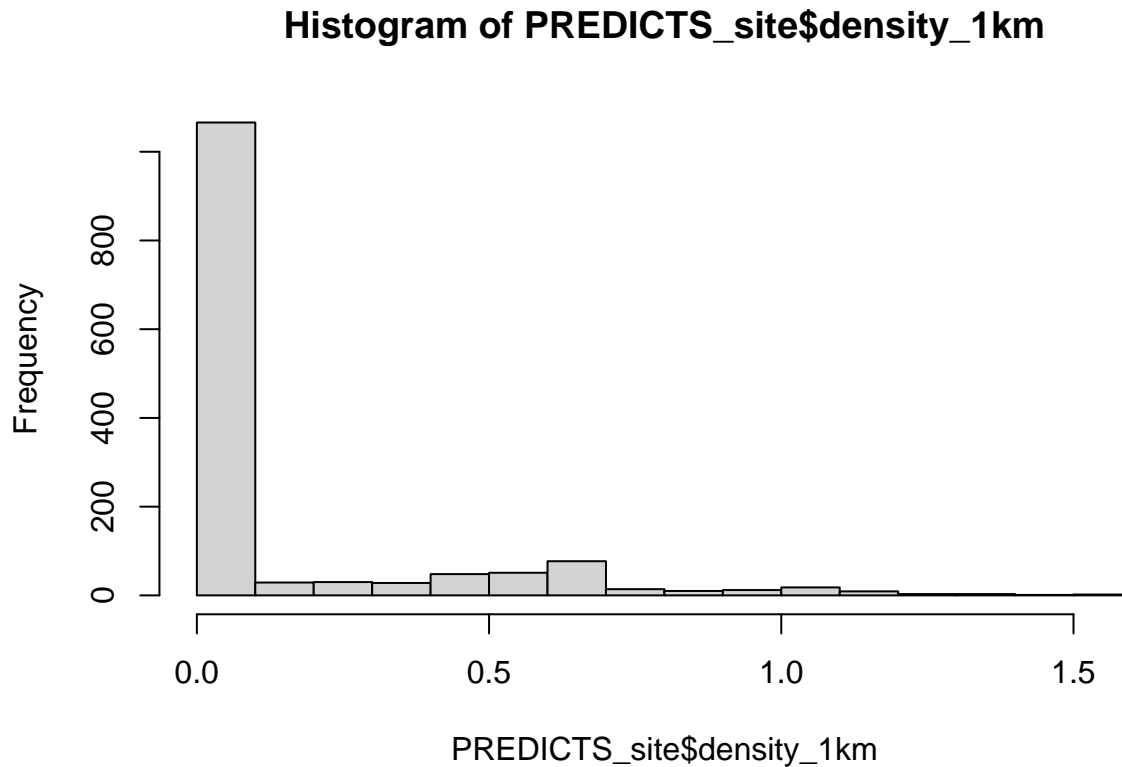
##									
##	PriMin	CrpLig	CrpMin	PasIn	PasLig	PasMin	PriIn	PriLig	SecIn
##									

```
##      378      38      33      60      34      28      29      308      53      166      125
## UrbLig UrbMin
##      56      93
```

### Variable transformations

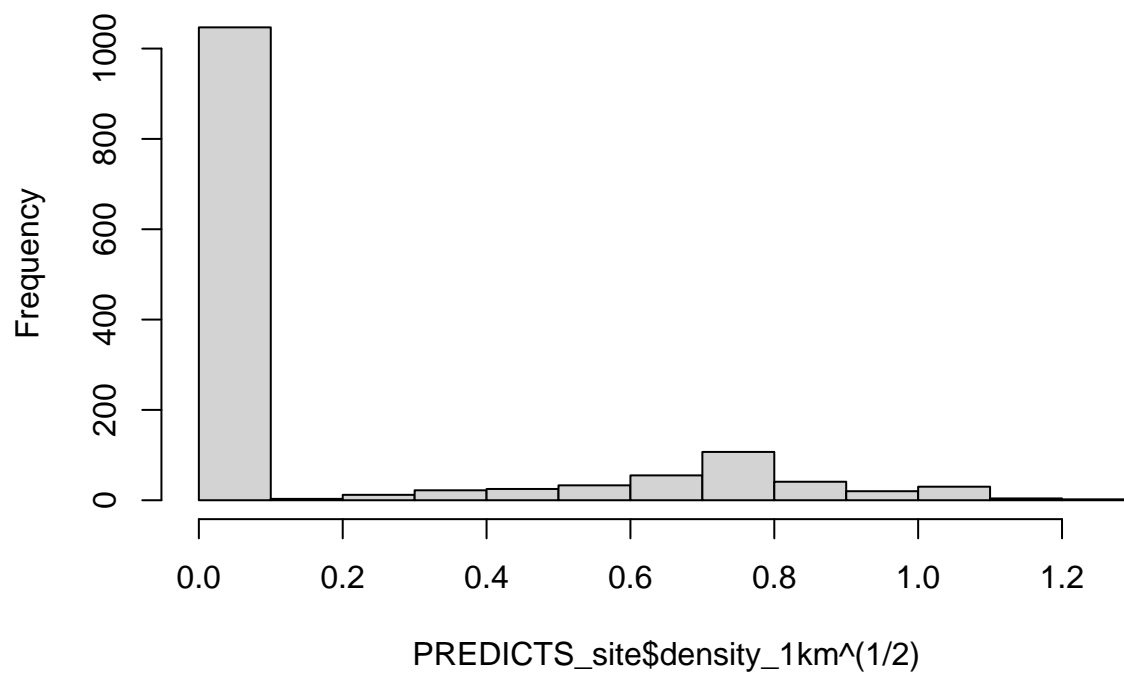
The other continuous variables that I will be including in my model will be human population density, which is already log transformed, and the density of roads at 1km and 50km radius from sites.

```
hist(PREDICTS_site$density_1km)
```



```
hist(PREDICTS_site$density_1km^(1/2))
```

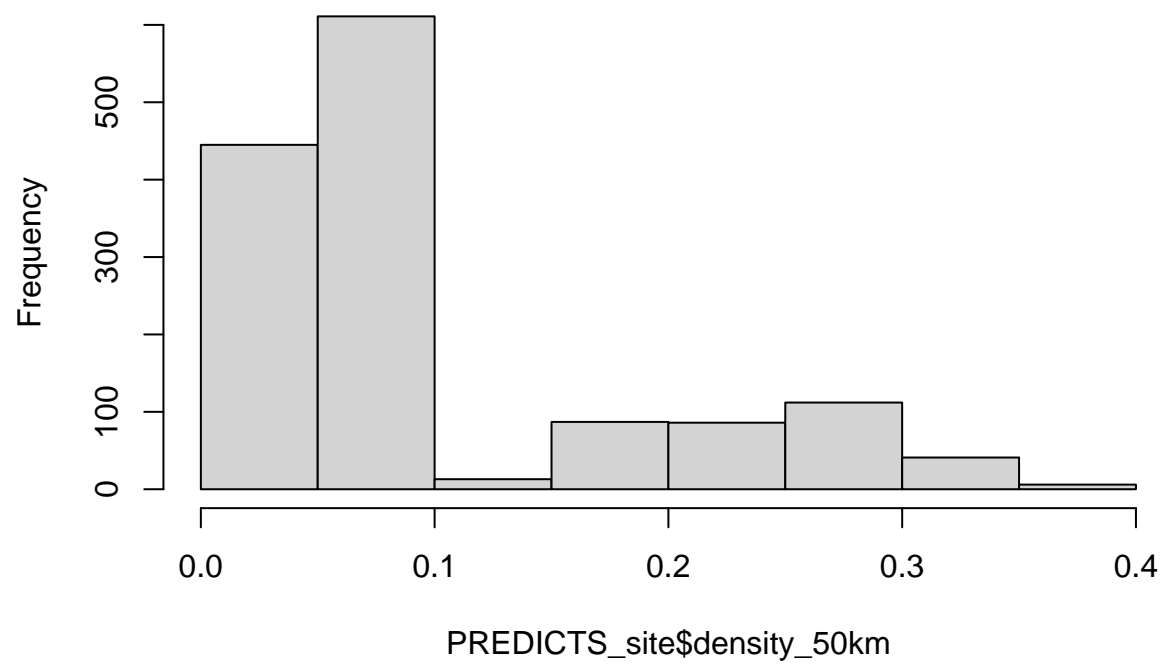
**Histogram of PREDICTS\_site\$density\_1km<sup>1/2</sup>)**



```
hist(PREDICTS_site$density_50km)
```

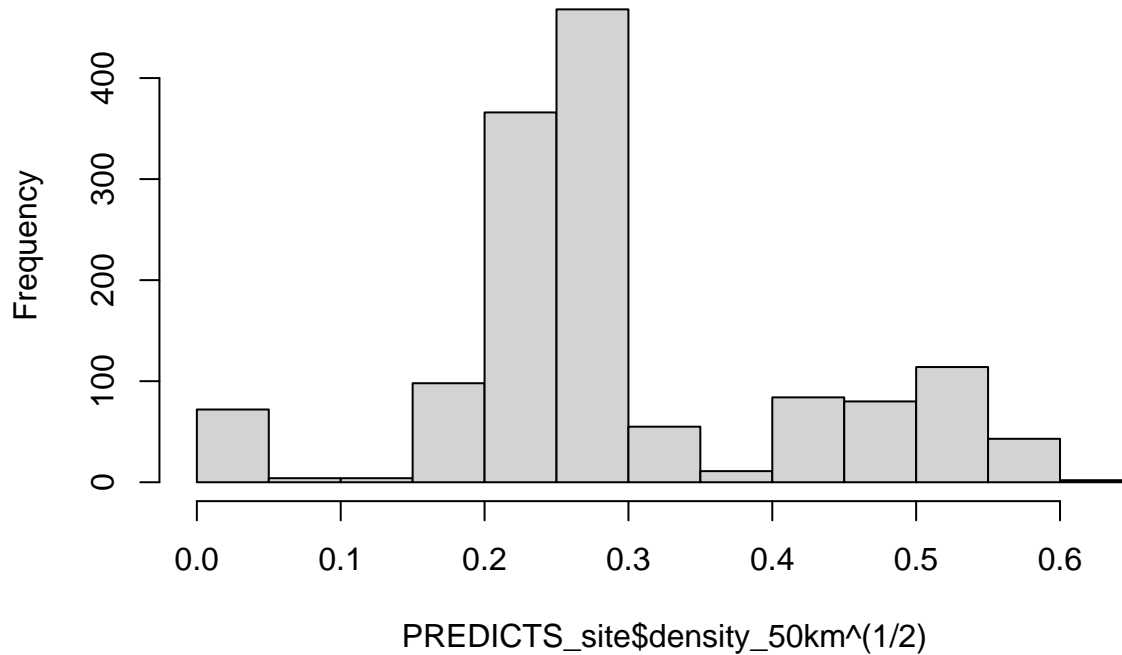


**Histogram of PREDICTS\_site\$density\_50km**



```
hist(PREDICTS_site$density_50km^(1/2))
```

## Histogram of PREDICTS\_site\$density\_50km<sup>(1/2)</sup>



### Colinearity

Next I am going to check for colinearity in the predictor variables, but beforehand I apply the square root transformation to the density of roads and scale all variables to reduce possible colinearity issues.

```
PREDICTS_site$logHPD <- scale(PREDICTS_site$logHPD)
PREDICTS_site$RD1k <- scale(PREDICTS_site$density_1km(1/2))
PREDICTS_site$RD50k <- scale(PREDICTS_site$density_50km(1/2))
PREDICTS_site$CNTRLlogHPD <- scale(PREDICTS_site$CNTRLlogHPD)

#source("https://highstat.com/Books/Book2/HighstatLibV10.R")
#corvif(PREDICTS_site[,c("LUI", "logHPD", "RD1k", "RD50k", "UN_subregion")])

### a lot of colinearity try removing UN-subregion
#corvif(PREDICTS_site[,c("LUI", "logHPD", "RD1k", "RD50k")])

## removing UN_subregion has reduced the colinearity to acceptable levels
```

### Phylogenetic Similarity

Finally I calculate the pairwise phylogenetic similarity between sites, as mentioned before I will be using 1-unifrac.

```
#### Load in Phylogeny - just tae the first one when running the inital models
```

```

all_bird_tree <- read.tree("../Datasets/AllBirdsHackett1.tre")
all_bird_tree <- all_bird_tree[[1]]

#####
#### Unifrac for all sites among studies ####
#####

create_vcv <- function(data, level) {

  data <- data.frame(droplevels(data))
  species <- sub(unique(data$Jetz_Name), pattern = " ", replacement = "_")
  drop.species <- all_bird_tree$tip.label[which(!(all_bird_tree$tip.label %in% species))]

  overall_tree <- drop.tip(all_bird_tree, drop.species)

  site_data <- data.frame(PREDICTS_site)
  ID <- as.character(unique(site_data[, level]))

  comm_data <- t(species)
  colnames(comm_data) <- species
  comm_data <- data.frame(comm_data[-1,])

  for(i in 1:length(ID)){

    ID_data <- data.frame(data[data[,level] == ID[i], c("Jetz_Name", "Effort_Corrected_Measurement")])

    for(spp in species){
      comm_data[i, paste(spp)] <- ifelse(any(ID_data[ID_data$Jetz_Name == sub(spp, pattern = "_", replacement = " ")]),
      1, 0)
    }

    rownames(comm_data)[i] <- ID[i]
  }

  for(i in 1:ncol(comm_data)){
    comm_data[,i] <- as.numeric(comm_data[,i])
  }

  comm_data <- as.matrix(comm_data)

  suppressWarnings(memory.limit(120000))

  vcv <- 1 - as.matrix(unifrac(comm = comm_data, tree = overall_tree))

```

```

    return(vcv)
}

#site_vcv <- create_vcv(PREDICTS_abundance, level = "SSBS")

# write_rds(file = "../Functional_Intactness_Index/Outputs/site_vcv.rds", site_vcv)

```

## Checking for phylogenetic signal in functional diversity and the responses of sites to land use change

```

##### Identify whether there is phylogenetic signal of in the responses of sites of LU change #####
##### first we are going to create a cluster dendrogram based on the phylogenetic distances between sites
##### Using an non-ultmetric tree we will be able to see just how heirarchial the resulting tree is, and
##### in the modelling.

PREDICTS_site <- PREDICTS_site %>% droplevels()

site_vcv <- readRDS("Outputs/site_vcv.rds")

studies <- as.character(unique(PREDICTS_site$SS))

## get the first site from each study as it is very difficult to observe the full tree with all sites

first_site <- c()
for(i in 1:length(studies)){
  first <- which(grepl(studies[i], colnames(site_vcv)))[1]
  first_site <- c(first_site, first)
}

first_sites <- site_vcv[first_site, first_site]

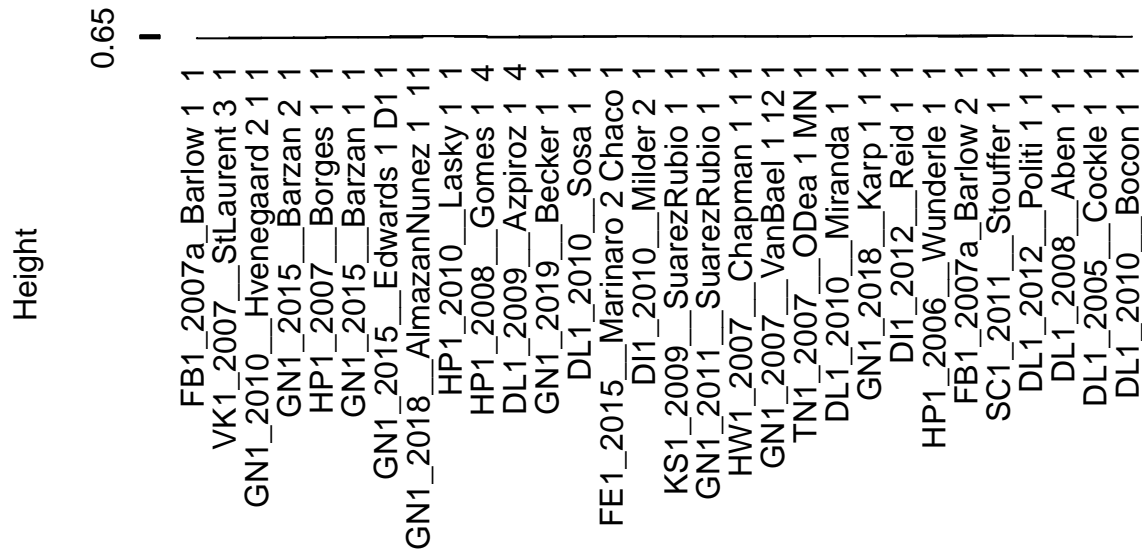
### performing a clustering algorithm based on the distances between sites - first convert back to diss.

site_dendro <- hclust(as.dist(1 - first_sites), method = "median")

plot(site_dendro)

```

## Cluster Dendrogram



```
as.dist(1 - first_sites)
hclust (*, "median")
```

### Can actually see that there is not a large amount of heirarchical structure within the sites with on

The two Lasky studies are related to each other because they are studies at the same sites but between the wet and the dry season - should I then collapse them in on each other? The Hvenegaard studies are again at the same sites but conducted at a 50m radius and unlimited radius again should probably remove or collapse the studies into each other? I have since done this by merging the two sites for lasky and then discarded the study that conducted their point count survey at 50m radius.

#### This initially does not suggest that there is a large amount of phylogenetic similarity between sites  
 #### correlated responses to land-use change - but another check I am going to conduct is to see whether  
 #### slopes of landuse within studies are more similar in studies that are more phylogenetically similar  
 #### of slope differences.

#### First calculate study level phylogenetic similarity

```
Study_sim <- create_vcv(PREDICTS_abundance, level = "SS")
```

### Next run a model without accounting for phylogenetic signal of sites and extract the random slopes

#1. fit model with no phylogeny in error term

```
test_mod <- lmer(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  (1|SS) + (1|SSB) + (1 + LUI|SS), data = PREDICTS_site)
```

```

## boundary (singular) fit: see ?isSingular
## Warning: Model failed to converge with 2 negative eigenvalues: -1.4e+01 -1.3e+02
## Warning in as_lmerModLT(model, devfun): Model may not have converged with 1
## eigenvalue close to zero: 9.8e-09
#### extract the random effects

Randoms <- lme4::ranef(test_mod)

study_slope <- Randoms$SS

significance <- c()

for(position in 1:c(length(studies)-1)){

  position_diff <- c()
  relatives <- c()

  for(study in studies){

    position_rel <- colnames(Study_sim)[order(Study_sim[study,],decreasing = TRUE)[position + 1]]
    rel_diff <- abs(study_slope[study,] - study_slope[position_rel,])
    rownames(rel_diff) <- paste(position_rel)

    position_diff <- rbind(position_diff, rel_diff)
    relatives <- rbind(relatives, data.frame(stud1 = study, stud2 = position_rel) )

  }

  stud_com <- data.frame(gtools::combinations(v = studies, r = 2, n = c(length(studies)-1)))

  drops <- c()
  for(k in 1:nrow(relatives)){
    drop_row <- which(stud_com$X1 == relatives[k,1] & stud_com$X2 == relatives[k,2] | stud_com$X1 == relatives[k,2] & stud_com$X2 == relatives[k,1])
    drops <- c(drops, drop_row)
  }

  stud_com <- stud_com[-unique(drops),]

  global_differences <- c()
  for(i in 1:NROW(stud_com)){
    abs_diff <- abs(study_slope[stud_com[i,1],] - study_slope[stud_com[i,2],])
    rownames(abs_diff) <- paste(stud_com[i,2])

    global_differences <- rbind(global_differences, abs_diff)
  }

  p_val <- wilcox.test(position_diff[,2], global_differences[,2], alternative = "less")

```

```

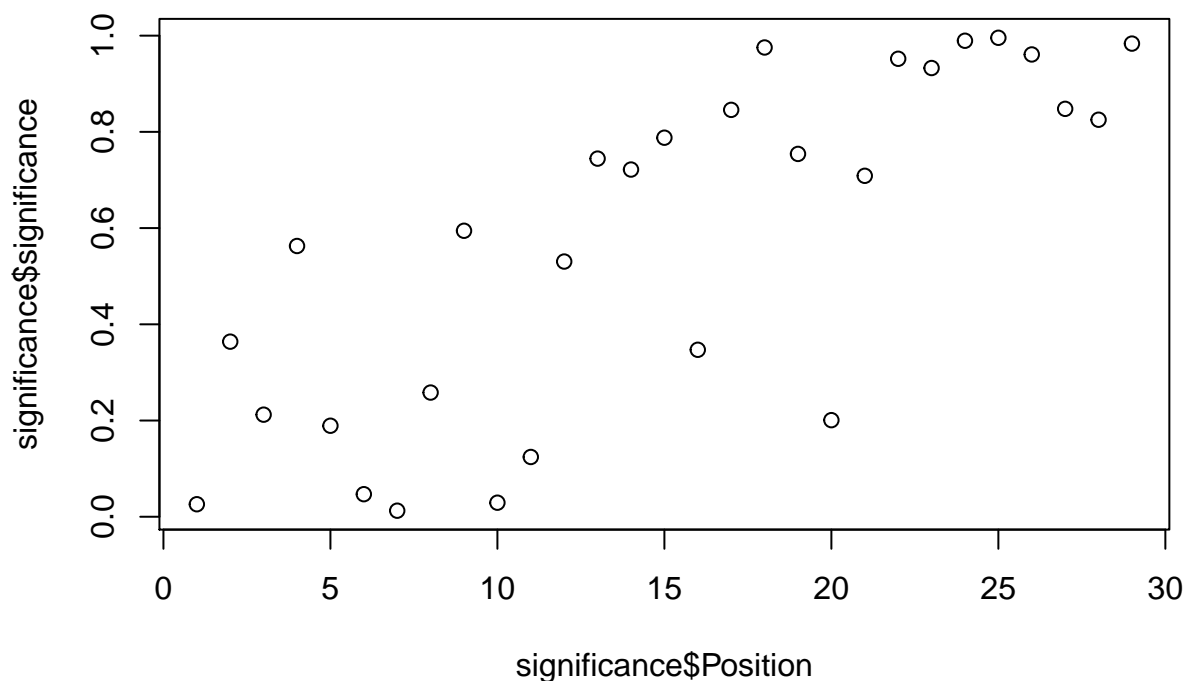
p_val <- data.frame(Position = position, significance = p_val$p.value)

significance <- rbind(significance, p_val)
}

#### plot doesn't clearly indicate that the responses of sites within studies are significantly more similar
#### similar sites

plot(significance$significance ~ significance$Position)

```



```

### fishers method - combined p values testing whether the hypothesis test (are the difference in slopes
### to each other in studies that are more phylogenetically related?)

```

```

### No they are not

```

```

poolr::fisher(significance$significance)

```

```

## combined p-values with:      Fisher's method
## number of p-values combined: 29
## test statistic:              57.803 ~ chi-square(58)
## adjustment:                  none
## combined p-value:            0.483

```

There does seem to be a slight association between the probability of rejecting the null hypothesis and the phylogenetic similarity of the comparison study so that when comparing the differences in slopes in more

phylogenetically related studies there is a greater probability of the differences in slopes being significantly more similar to each compared to the global distribution of differences in slopes. However, when performing a Fishers combined probability there is not a significant difference so we can accept the null hypothesis. — Possibly if I drop the two related studies `lasky` and `hvengaard`... Removing one of each made the p-value from fishers combined increase a good amount.

Therefore, that does not indicate to me at the moment that I need to account for phylogenetic similarity of sites within the modelling framework, as there is not a clear hierarchy of sites with greater phylogenetic similarity to each other and they are not significantly ore similar in their responses to ;and use change therefore I will continue on with GLMMs, however, I will continue to use the `phyr` package as it will allow me to fit models using INLA and bayesian techniques, which helps with some convergence issues.

## INLA/GLMM

With that we now have everything ready to begin modeling. Random effects that I will be including in the models will be study, to account for between study differences in location, method and sampling effort and study block, to account for spatial configuration of sites. Additionally, I will be testing for random slopes for each of the predictor variables within study by identifying which random effect structure has the lowest DIC in the maximal model.

```
#####
# With this we can proceed with modelling using GLMMs as opposed to PGLMMs
#####
PREDICTS_site <- PREDICTS_site %>% droplevels()

Rao_Model_1b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
                                     LUI:logHPD + LUI:RD1k + LUI:RD50k +
                                     (1|SS) + (1|SSB), data = PREDICTS_site, bayes = TRUE)

#### lets see what random effect structure gives the best AIC

### adding random slopes 1) LUI within study

Rao_Model_2b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
                                     LUI:logHPD + LUI:RD1k + LUI:RD50k +
                                     (1|SS) + (1|SSB) + (LUI|SS), data = PREDICTS_site, bayes = TRUE)

#2) logHPD within study

Rao_Model_3b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
                                     LUI:logHPD + LUI:RD1k + LUI:RD50k +
                                     (1|SS) + (1|SSB) + (logHPD|SS), data = PREDICTS_site, bayes = TRUE)

#3) Road Density_1km within study
```



```
Rao_Model_4b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  LUI:logHPD + LUI:RD1k + LUI:RD50k +
  (1|SS) + (1|SSB) + (RD1k|SS), data = PREDICTS_site, bayes = TRUE)

# 4) RD50km within study

Rao_Model_5b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  LUI:logHPD + LUI:RD1k + LUI:RD50k +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)

Random_DIC <- data.frame(mod1 = Rao_Model_1b$DIC, mod2 = Rao_Model_2b$DIC, mod3 = Rao_Model_3b$DIC,
  mod4 = Rao_Model_4b$DIC, mod5 = Rao_Model_5b$DIC)
```

```
Random_DIC
```

```
##      mod1      mod2      mod3      mod4      mod5
## 1 -4806.637 -4804.684 -4807.846 -4802.127 -4809.396
```

The model with the lowest DIC is with a random slope of RD50k with therefore model 5 will be carried forward.

Next to identify the optimal fixed effect structure I removed both interactions in turn to see whether that reduced the DIC of the model. Typically a reduction >2 will provide support for the simplified model.

#### Remove each interaction to see if this improves the model at all.

*#LUI:RD50k*

```
Rao_Model_6b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  LUI:logHPD + LUI:RD1k +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

*#LUI:RD1k*

```
Rao_Model_7b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  LUI:logHPD + LUI:RD50k +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

*# LUI:logHPD*

```
Rao_Model_8b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
  LUI:RD50k + LUI:RD1k +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```

Model_differences <- data.frame(RD50k = Rao_Model_6b$DIC - Rao_Model_1b$DIC,
                                RD1k = Rao_Model_7b$DIC - Rao_Model_1b$DIC,
                                logHPD = Rao_Model_8b$DIC - Rao_Model_1b$DIC)

Model_differences

##          RD50k          RD1k      logHPD
## 1 -13.40589 -10.54026 -7.340598
#### Model DIC is reduced in all but most when removing the interaction between LUI:RD50k so model 6b is best
## RD50k fixed

Rao_Model_9b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + CNTRLlogHPD +
                                      LUI:logHPD + LUI:RD1k +
                                      (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)

# LUI:RD1k

Rao_Model_10b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + RD50k + CNTRLlogHPD +
                                       LUI:logHPD +
                                       (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)

# LUI:logHPD

Rao_Model_11b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD50k + RD1k + CNTRLlogHPD +
                                       LUI:RD1k +
                                       (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)

Model_differences <- data.frame(RD50k = Rao_Model_9b$DIC - Rao_Model_6b$DIC,
                                RD1k = Rao_Model_10b$DIC - Rao_Model_6b$DIC,
                                logHPD = Rao_Model_11b$DIC - Rao_Model_6b$DIC)

Model_differences

##          RD50k          RD1k      logHPD
## 1 -0.6541368 -11.69414 -4.432754
#### Model DIC is reduced in all but most when removing the interaction between LUI:RD1k so model 10b is best
## RD50k fixed

Rao_Model_12b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + CNTRLlogHPD +
                                       LUI:logHPD +
                                       (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)

```

```
### RD1k fixed
```

```
Rao_Model_13b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD50k + CNTRLlogHPD +
  LUI:logHPD +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```
#### LUI:logHPD
```

```
Rao_Model_14b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD50k + RD1k + CNTRLlogHPD +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```
Model_differences <- data.frame(RD50k = Rao_Model_12b$DIC - Rao_Model_10b$DIC,
  RD1k = Rao_Model_13b$DIC - Rao_Model_10b$DIC,
  logHPD = Rao_Model_14b$DIC - Rao_Model_10b$DIC)
```

```
Model_differences
```

```
##          RD50k      RD1k    logHPD
## 1 -0.5398687  3.053603 -2.343013
```

```
#### Model DIC is reduced wehn removing the interaction between LUI:logHPD and is increased in the other
```

```
#### RD50k
```

```
Rao_Model_15b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD1k + CNTRLlogHPD +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```
#### RD1k
```

```
Rao_Model_16b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + logHPD + RD50k + CNTRLlogHPD +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```
### logHPD
```

```
Rao_Model_17b <- phyr::communityPGLMM(Unbias_Rao ~ LUI + RD1k + RD50k + CNTRLlogHPD +
  (1|SS) + (1|SSB) + (RD50k|SS), data = PREDICTS_site, bayes = TRUE)
```

```
Model_differences <- data.frame(RD50k = Rao_Model_15b$DIC - Rao_Model_14b$DIC,
  RD1k = Rao_Model_16b$DIC - Rao_Model_14b$DIC,
```

```
logHPD = Rao_Model_17b$DIC - Rao_Model_14b$DIC)
```

```
####
```

```
Model_differences
```

```
##          RD50k      RD1k    logHPD
```

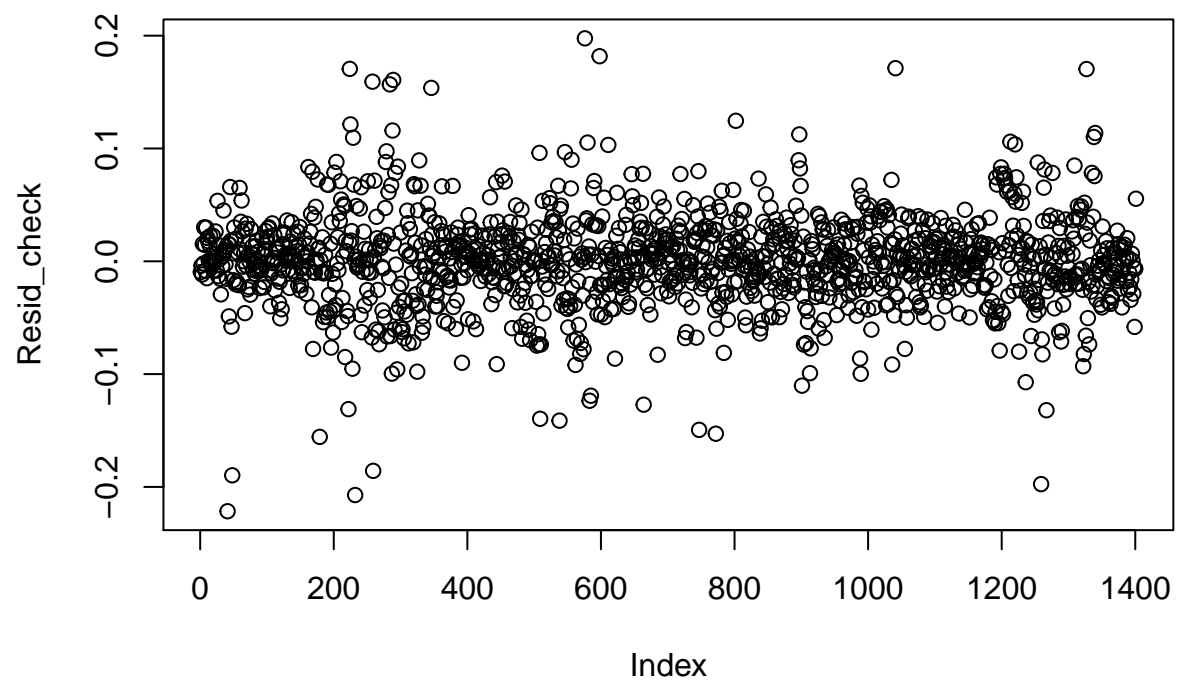
```
## 1 -0.5424302 3.153006 -3.181753
```

## Final Model

After going through model selection the best model seems to be the one with LUI, logHPD and RD1k and RD50k as Fixed effects and no interactions between LUI and any of the other pressures with the best model being Rao\_Model\_14b.

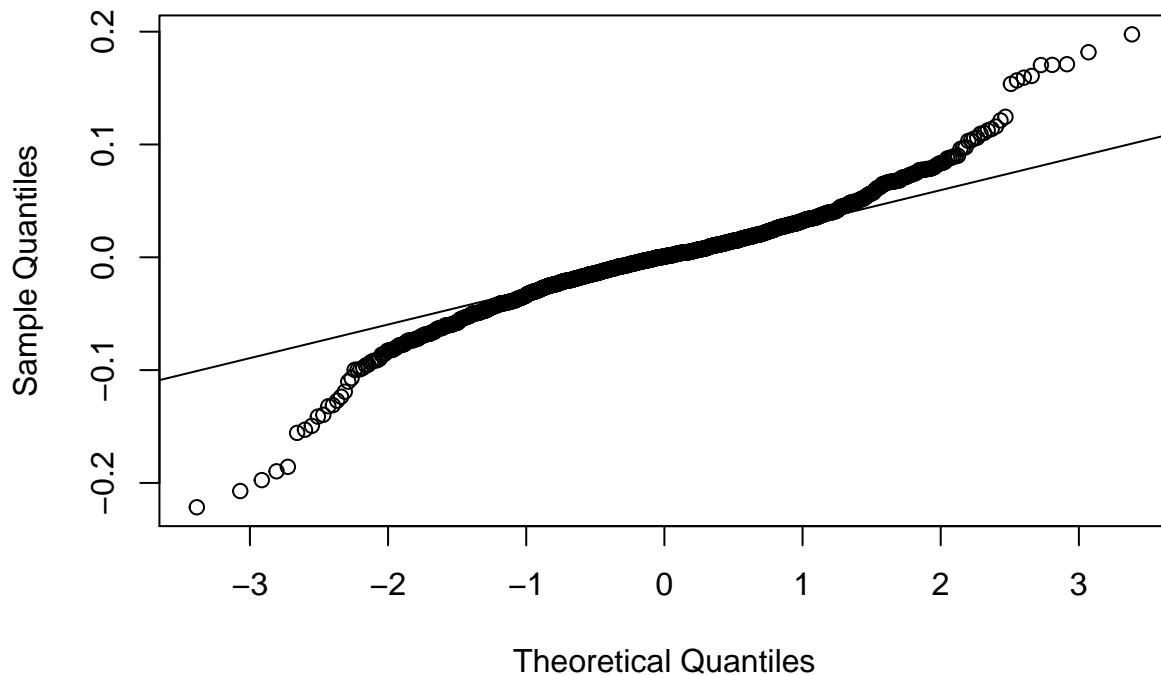
```
## funtion to visualise the residuals from the PGLMM and check whether they are normally distributed
```

```
residuals.communityPGLMM <- function(  
  object,  
  type = if(object$family %in% c("binomial","poisson")) "deviance" else "response",  
  scaled = FALSE, ...){  
  if(object$family == "gaussian"){  
    y <- object$Y  
    mu <- pglmm_predicted_values(object)$Y_hat  
    res <- switch(type,  
                  deviance = stop("no deviance residuals for gaussian model", call. = FALSE),  
                  response = y - mu  
    )  
    if(scaled) res/sqrt(object$s2resid)  
  }  
  
  if(object$family %in% c("binomial","poisson")){  
    y <- as.numeric(object$Y)  
    mu <- unname(object$mu[, 1])  
    if(object$family == "binomial") dres <- sqrt(binomial())$dev.resids(y, mu, 1))  
    if(object$family == "poisson") dres <- sqrt(poisson())$dev.resids(y, mu, 1))  
    res <- switch(type,  
                  deviance = {  
                    dres  
                    ifelse(y > mu, dres, - dres)  
                  },  
                  response = y - mu  
    )  
  }  
  if(object$family %nin% c("gaussian", "binomial", "poisson"))  
    stop("no residual methods for family other than gaussian, binomial and poisson, yet", call. = FALSE)  
  
  unname(res)  
}  
  
Resid_check <- residuals.communityPGLMM(Rao_Model_14b)  
  
plot(Resid_check)
```



```
qqnorm(Resid_check)  
qqline(Resid_check)
```

## Normal Q-Q Plot



```
write_rds(file = "Outputs/Rao_model.rds", Rao_Model_14b)
```

## Robust estimates

looking at the residuals it seems that there are some influential points that are caused by flocks of birds dominating the site in terms of relative abundance resulting in very low estimates of functional diversity. Because these values are true values we want to still include them in the model but we would like to down-weight their contribution to the model therefore I am going to perform a Robustlmm with the final selected model

```
Robust_mod <- rlmer(Unbias_Rao ~ LUI + logHPD + RD50k + RD1k + CNTRLlogHPD +  
                    (1|SS) + (1|SSB), data = PREDICTS_site)  
summary(Robust_mod)
```

```
## Robust linear mixed model fit by DASTau  
## Formula: Unbias_Rao ~ LUI + logHPD + RD50k + RD1k + CNTRLlogHPD + (1 |      SS) + (1 | SSB)  
##   Data: PREDICTS_site  
##  
## Scaled residuals:  
##      Min      1Q  Median      3Q      Max  
## -6.4716 -0.5735  0.0084  0.5613  5.7562  
##  
## Random effects:  
##   Groups   Name                Variance Std.Dev.  
##   SSB      (Intercept)  0.000000  0.00000  
##   SS       (Intercept)  0.001256  0.03544  
##   Residual                        0.001231  0.03509
```

```

## Number of obs: 1401, groups: SSB, 133; SS, 30
##
## Fixed effects:
##           Estimate Std. Error t value
## (Intercept)  0.3244645  0.0076589  42.36
## LUICrpLig    -0.0271857  0.0069105  -3.93
## LUICrpMin    -0.0228550  0.0075125  -3.04
## LUIPasIn     -0.0184758  0.0076104  -2.43
## LUIPasLig     0.0093647  0.0071645   1.31
## LUIPasMin    -0.0176397  0.0078470  -2.25
## LUIPriIn     0.0040981  0.0075297   0.54
## LUIPriLig     0.0047384  0.0037180   1.27
## LUISecIn     0.0059800  0.0065125   0.92
## LUISecLig    -0.0109865  0.0048742  -2.25
## LUISecMin    -0.0105248  0.0047242  -2.23
## LUIUrbLig     0.0077331  0.0103061   0.75
## LUIUrbMin     0.0032144  0.0057172   0.56
## logHPD       0.0089591  0.0026463   3.39
## RD50k        -0.0002842  0.0046109  -0.06
## RD1k         0.0034892  0.0012383   2.82
## CNTRLlogHPD  -0.0185185  0.0072923  -2.54
##
## Correlation of Fixed Effects:
##           (Intr) LUICrL LUICrM LUIPsI LUIPsL LUIPsM LUIPrI LUIPrL LUIScI
## LUICrpLig    -0.096
## LUICrpMin    -0.067  0.215
## LUIPasIn     -0.121  0.108  0.057
## LUIPasLig    -0.105  0.119  0.063  0.155
## LUIPasMin    -0.111  0.092  0.062  0.253  0.219
## LUIPriIn     -0.076  0.063  0.029  0.107  0.178  0.140
## LUIPriLig    -0.199  0.285  0.106  0.350  0.276  0.241  0.190
## LUISecIn     -0.109  0.107  0.053  0.398  0.114  0.139  0.077  0.325
## LUISecLig    -0.213  0.198  0.120  0.230  0.264  0.264  0.134  0.543  0.204
## LUISecMin    -0.172  0.236  0.293  0.141  0.168  0.173  0.072  0.232  0.141
## LUIUrbLig    -0.130  0.071  0.017  0.077  0.097  0.052  0.068  0.183  0.074
## LUIUrbMin    -0.119  0.115  0.036  0.147  0.197  0.127  0.154  0.310  0.124
## logHPD       0.074 -0.038  0.003  0.045 -0.013 -0.040 -0.020  0.023 -0.011
## RD50k        -0.020 -0.031 -0.005 -0.113 -0.111  0.021 -0.057 -0.117 -0.051
## RD1k         0.016  0.020 -0.058 -0.133 -0.051 -0.066 -0.020  0.006  0.009
## CNTRLlogHPD  0.232  0.021  0.018  0.067  0.068  0.026  0.033  0.023  0.031
##           LUIScL LUIScM LUIUrL LUIUrM logHPD RD50k RD1k
## LUICrpLig
## LUICrpMin
## LUIPasIn
## LUIPasLig
## LUIPasMin
## LUIPriIn
## LUIPriLig
## LUISecIn
## LUISecLig
## LUISecMin    0.324
## LUIUrbLig     0.139  0.051
## LUIUrbMin     0.233  0.100  0.383
## logHPD        -0.142 -0.027 -0.248 -0.333

```

```
## RD50k      0.017 -0.012 -0.198 -0.273 -0.441
## RD1k      -0.084 -0.120  0.061  0.034 -0.232  0.105
## CNTRLlogHPD 0.023  0.073  0.034  0.174 -0.045 -0.331 -0.046
##
## Robustness weights for the residuals:
## 1092 weights are ~= 1. The remaining 309 ones are summarized as
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.208  0.581   0.728   0.715   0.898   0.998
##
## Robustness weights for the random effects:
## 161 weights are ~= 1. The remaining 2 ones are
##   151   163
## 0.819 0.938
##
## Rho functions used for fitting:
##   Residuals:
##     eff: smoothed Huber (k = 1.345, s = 10)
##     sig: smoothed Huber, Proposal II (k = 1.345, s = 10)
##   Random Effects, variance component 1 (SSB):
##     eff: smoothed Huber (k = 1.345, s = 10)
##     vcp: smoothed Huber, Proposal II (k = 1.345, s = 10)
##   Random Effects, variance component 2 (SS):
##     eff: smoothed Huber (k = 1.345, s = 10)
##     vcp: smoothed Huber, Proposal II (k = 1.345, s = 10)
```

## Rao\_Model\_14b

```
summary(Rao_Model_14b)
```

```
## Linear mixed model fit by Bayesian INLA
##
## Call:Unbias_Rao ~ LUI + logHPD + RD50k + RD1k + CNTRLlogHPD
##
## marginal logLik          DIC          WAIC
##           2250          -4834          -4840
##
## Random effects:
##           Variance Std.Dev lower.CI upper.CI
## 1|SS      6.869e-04 0.026208 3.407e-04 0.0015846
## 1|SSB     4.484e-05 0.006697 1.608e-05 0.0001704
## RD50k|SS  3.325e-05 0.005766 5.707e-06 0.0005149
## residual 1.769e-03 0.042057 1.638e-03 0.0019170
##
## Fixed effects:
##           Value      lower.CI upper.CI
## (Intercept) 0.32538462 0.31584267 0.3348
## LUICrpLig   -0.02807182 -0.04428178 -0.0119
## LUICrpMin   -0.01963325 -0.03724547 -0.0020
## LUIPasIn    -0.01749297 -0.03583307  0.0008
## LUIPasLig    0.00915832 -0.00759894  0.0259
## LUIPasMin   -0.01442213 -0.03275868  0.0039
## LUIPriIn     0.00450485 -0.01314775  0.0221
## LUIPriLig    0.00264896 -0.00670478  0.0119
## LUISecIn     0.00418852 -0.01135909  0.0197
```



```
## LUISecLig -0.01171547 -0.02343746 0.0000
## LUISecMin -0.00862088 -0.01946071 0.0022
## LUIUrbLig -0.00044823 -0.02746654 0.0263
## LUIUrbMin -0.00203201 -0.01634333 0.0122
## logHPD 0.00651562 -0.00134445 0.0141
## RD50k 0.00273714 -0.00904694 0.0147
## RD1k 0.00402016 0.00107077 0.0070
## CNTRLlogHPD -0.01860155 -0.03372255 -0.0034
```

The summary of the final Rao model shows that Functional diversity, compared to primary minimal habitat, is reduced across LUIs except Primary intense use and Secondary intense use where it increases slightly. Significant reductions, as determined by the credible intervals of the parameter distribution not crossing zero, were shown in cropland light and minimal, pasture intense and minimal, secondary light and minimal and urban minimal habitats. In response to the other pressures FD increases with greater logHPD and road density at 1km radius, while FD decreases with Road density at 50km.

Running the model selection framework with the “biased” Rao yielded similar results however, the best model also included a random slope of LUI within study which resulted in FD swapping from observed declines in Primary light use, and urban light use to positive affecting functional diversity.

## 1.2 Modelling: Functional Similarity/Overlap (GLMM)

To assess the impact of landuse change and other pressures on functional similarity, calculated as the overlap between trait probability density hypervolumes in trait space I used GLMMS. The other pressures include human population density, density of roads @ 1 and 50km radii, and geographic and environmental distances. Additionally, since the functional similarity of sites may be influenced by the absolute level of pressure experienced at that site but also the difference in pressure experience between the sites, I included the pressure at site 2 (the site being compared to Primary minimal land use) and the difference in pressures between site 1 and 2.

Because we are interested more in the differences between sites as predictors for functional similarity I considered the absolute value at the second site as an control variable and the differences and their interactions as possible explanatory variables. In model selection if the difference in pressure between the sites drops out of the model the control variable is also removed.

To determine the optimal random effect structure I used the aforementioned selection process picking the structure with the lowest AIC.

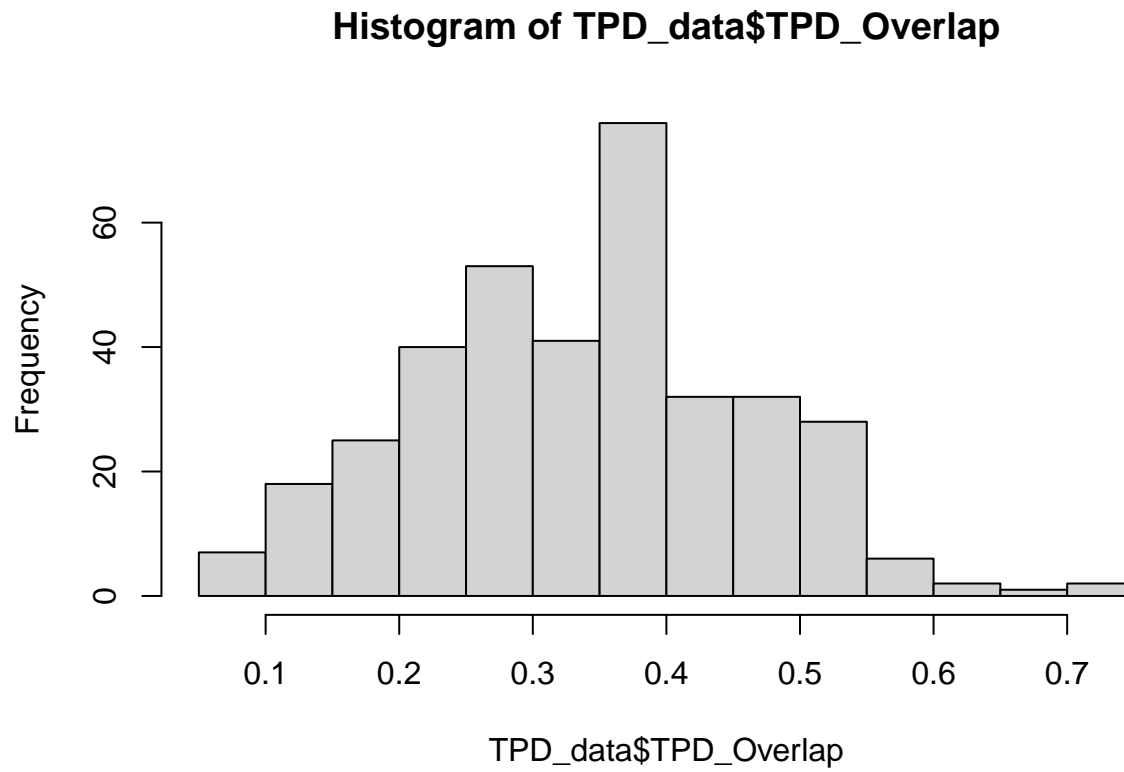
Determining the best fixed effect structure however, is a bit more difficult, as the comparisons are not-independent of each other as a single primary minimal site being compared multiple times, therefore traditional likelihood ratio tests will not be suitable. Again following De Palma et al. In review, I permuted the data 1000 times by randomly shuffling the functional similarity scores within studies and refitted the full and simplified model to each permuted dataset to generate a distribution of Likelihood ratio scores. Comparing the likelihood ratio score between models using the observed data to this distribution, a simplified model would be supported if the score is lower than the 95th percentile of the distribution as it has not significantly reduced the explanatory power of the model.

As the overlap score will be bound between 0 and 1 as it is the proportion of overlap between two communities I logit transformed the values to extend the tails of the distribution.

```
rm(list = ls())
TPD_data <- readRDS("Outputs/Trait_Prob_den.rds")
Similarity_data <- readRDS("Outputs/similarity_data.rds")
```

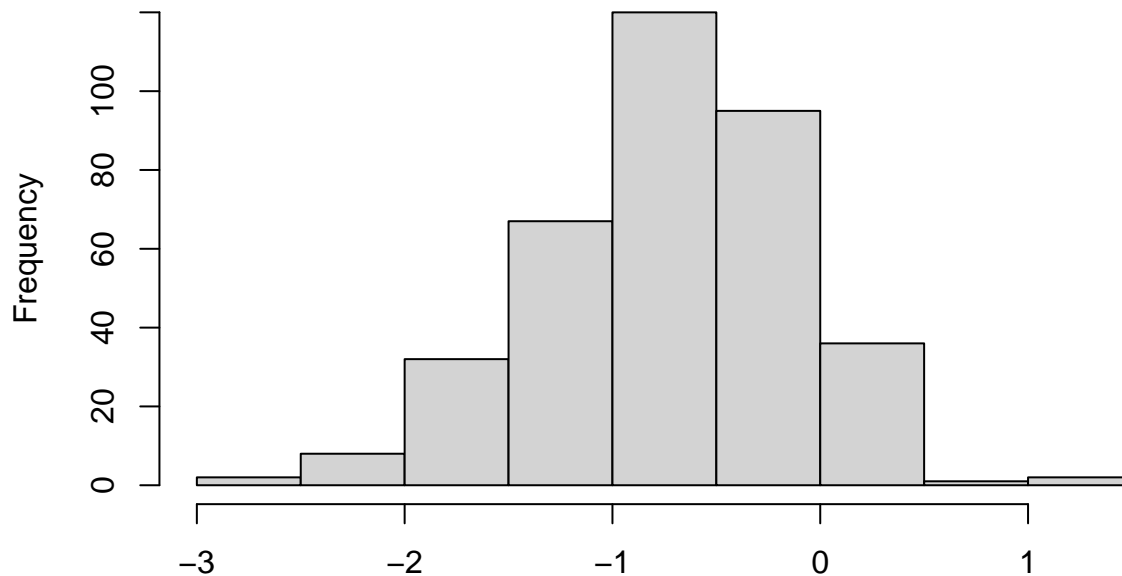
### Also want to add in a variable for the minimum number of species in either of the sites used to con

```
## lets have a look at the other variables and what transformations will be best  
hist(TPD_data$TPD_Overlap)
```



```
hist(car::logit(TPD_data$TPD_Overlap, adjust = 0.001, percents = FALSE))
```

rogram of car::logit(TPD\_data\$TPD\_Overlap, adjust = 0.001, percents =



car::logit(TPD\_data\$TPD\_Overlap, adjust = 0.001, percents = FALSE)

*## because the similarity metric is bound between 0 and 1 and i performed a logit transformation as well.*

*### Want to work out human population density at*

```
TPD_data <- TPD_data %>% dplyr::mutate(logitOver = car::logit(TPD_Overlap, adjust = 0.001, percents = F
  dplyr::mutate(logdist = log(distance + 1),
    rt3env = env_distance^(1/3),
    sqrtS2RD1K = S2RD1K^(1/2),
    sqrtS2RD50K = S2RD50K^(1/2),
    RD1Kdiff = sqrtS2RD1K - (S1RD1K^(1/2)),
    RD50Kdiff = sqrtS2RD50K - (S1RD50K^(1/2)))
```

```
table(TPD_data$Contrast)
```

```
##
##           Primary_Minimal use-Cropland_Light use
##                                     24
##           Primary_Minimal use-Cropland_Minimal use
##                                     27
##           Primary_Minimal use-Plantation forest_Minimal use
##                                     33
##           Primary_Minimal use-Primary_Intense use
```

```

##                                     16
##           Primary_Minimal use-Primary_Light use
##                                     50
##           Primary_Minimal use-Primary_Minimal use
##                                     106
## Primary_Minimal use-Secondary Vegetation_Light use
##                                     12
## Primary_Minimal use-Secondary Vegetation_Minimal use
##                                     94
##           Primary_Minimal use-Urban_Minimal use
##                                     1
TPD_data <- TPD_data %>% dplyr::mutate(Cont = ifelse(grepl(Contrast, pattern = "Plantation forest"),
  "PriMin-SecLig",
  paste(Contrast)),
  Cont = ifelse(grepl(Cont, pattern = "Primary_Light")|grepl(Cont, pattern =
  "PriMin-Primary",
  paste(Cont)),
  Cont = ifelse(grepl(Cont, pattern = "Cropland"),
    "PriMin-Cropland",
    paste(Cont)),
  Cont = ifelse(grepl(Cont, pattern = "Secondary Vegetation_Light use"),
    "PriMin-SecLig",
    paste(Cont)),
  Cont = relevel(factor(Cont), ref = "Primary_Minimal use-Primary_Minimal use")

levels(TPD_data$Cont) <- c("PriMin-PriMin", "PriMin-SecMin", "PriMin-UrbMin", "PriMin-Cropland", "PriMin-SecLig")

### there is only a single Urban minimal site left and is therefore dropped

TPD_data <- TPD_data[-which(TPD_data$Cont == "PriMin-UrbMin"),] %>% droplevels()

table(TPD_data$Cont)

##
## PriMin-PriMin PriMin-SecMin PriMin-Cropland PriMin-Primary PriMin-SecLig
##           106           94           51           66           45

TPD_data$rt3env <- scale(TPD_data$rt3env)
TPD_data$s2logHPD <- scale(TPD_data$s2logHPD)
TPD_data$logHPDdiff <- scale(TPD_data$logHPDdiff)
TPD_data$logdist <- scale(TPD_data$logdist)
TPD_data$sqrtS2RD1K <- scale(TPD_data$sqrtS2RD1K)
TPD_data$sqrtS2RD50K <- scale(TPD_data$sqrtS2RD50K)
TPD_data$RD1Kdiff <- scale(TPD_data$RD1Kdiff)
TPD_data$RD50Kdiff <- scale(TPD_data$RD50Kdiff)

#source("https://highstat.com/Books/Book2/HighstatLibV10.R")

#corvif(TPD_data[,c("Cont", "logdist", "rt3env", "s2logHPD", "logHPDdiff", "sqrtS2RD1K", "sqrtS2RD50K",

```

```
#####
#### Modelling #####
#####

Model_1 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
                sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
                Cont:logdist + Cont:logHPDdiff +
                + Cont:RD1Kdiff + Cont:RD50Kdiff +
                (1|SS),
                data = TPD_data)

#### random slopes

## logdist

Model_2 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
                sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
                Cont:logdist + Cont:logHPDdiff +
                Cont:RD1Kdiff + Cont:RD50Kdiff +
                (1|SS) + (1 + logdist|SS),
                data = TPD_data )

## boundary (singular) fit: see ?isSingular

## logHPDdiff

Model_3 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
                sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
                Cont:logdist + Cont:logHPDdiff +
                Cont:RD1Kdiff + Cont:RD50Kdiff +
                (1|SS) + (1 + logHPDdiff|SS),
                data = TPD_data )

## boundary (singular) fit: see ?isSingular

## RoadDdiff1k

Model_4 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
                sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
                Cont:logdist + Cont:logHPDdiff +
                Cont:RD1Kdiff + Cont:RD50Kdiff +
                (1|SS) + (1 + RD1Kdiff|SS),
                data = TPD_data)

## boundary (singular) fit: see ?isSingular

## RoadDdiff50k

Model_5 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
                sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
                Cont:logdist + Cont:logHPDdiff +
                Cont:RD1Kdiff + Cont:RD50Kdiff +
                (1|SS) + (1 + RD50Kdiff|SS),
```

```

      data = TPD_data)

## boundary (singular) fit: see ?isSingular
## Cont

Model_6 <- lmer(logitOver ~ Cont + logdist + rt3env + logHPDdiff +
  sqrtS2RD1K + S2logHPD + RD1Kdiff + RD50Kdiff + sqrtS2RD50K + CNTRLlogHPD +
  Cont:logdist + Cont:logHPDdiff +
  Cont:RD1Kdiff + Cont:RD50Kdiff +
  (1|SS) + (1 + Cont|SS),
  data = TPD_data)

## boundary (singular) fit: see ?isSingular
MOD_AIC <- data.frame(Mod1 = AIC(Model_1),Mod2 = AIC(Model_2),Mod3 = AIC(Model_3),
  Mod4 = AIC(Model_4),Mod5 = AIC(Model_5),Mod6 = AIC(Model_6))

```

Model 2 and 6 have the lowest AIC scores, however both they have singular fits therefore, I will proceed with Model 1 which is the next model with the most support so we will carry on without any random slopes. Next stage is to perform backwards stepwise model simplification based on the likelihood ratio distribution test on 1000 permuted datasets.

```

Permuted_data <- rep(list(NA),1000)

for(i in 1:1000){

  sample_data <-c()

  for(study in unique(TPD_data$SS)){
    data <- TPD_data %>% filter(SS == study)

    data$logitOver <- data[sample(NROW(data)),"logitOver"]

    sample_data <- rbind(sample_data,data)

  }

  Permuted_data[[i]] <- sample_data

}

#### function to generate the LR distribution across the 1000 datasets

##### Likelihood ratio function

Permuted_model_simplification <- function(data, model1, remove){

  formula <- as.formula(paste("~.-",remove,sep = ""))
  model2 <- update(model1,formula)

  LRT_dist <- c()

  for(i in 1:length(data)){

    mod1 <- lmer(model1@call, data = data[[i]], REML = FALSE)

```

```

mod2 <- lmer(model2@call, data = data[[i]], REML = FALSE)

LRT <- anova(mod1,mod2)
LRT <- LRT[which(!is.na(LRT$Chisq)), "Chisq"]

LRT_dist <- rbind(LRT_dist, LRT)
}

mod_LRT <- anova(model1, model2)
ChiSq <- mod_LRT[2, "Chisq"]

dist_quant <- quantile(LRT_dist, 0.95)

DROP <- ChiSq < dist_quant

percentile <- 0.01
test <- TRUE
while(test & percentile < 1.01){
  dq <- quantile(LRT_dist, percentile)
  test <- ChiSq > dq
  percentile <- percentile + 0.01
}
perecentile <- percentile - 0.01

res <- data.frame(DROP = DROP, Percentile = perecentile)
rownames(res) <- paste(remove)

return(res)
}
### so lets look at our best maximal model

Anova(Model_1, type = "II")

```

FALSE Analysis of Deviance Table (Type II Wald chisquare tests)

FALSE

FALSE Response: logitOver

FALSE	Chisq	Df	Pr(>Chisq)
FALSE Cont	246.7570	4	< 2.2e-16 ***
FALSE logdist	2.0882	1	0.148442
FALSE rt3env	2.0389	1	0.153318
FALSE logHPDdiff	1.4849	1	0.223007
FALSE sqrtS2RD1K	1.0875	1	0.297034
FALSE S2logHPD	0.4885	1	0.484617
FALSE RD1Kdiff	3.3673	1	0.066502 .
FALSE RD50Kdiff	0.4207	1	0.516608
FALSE sqrtS2RD50K	0.2076	1	0.648676
FALSE CNTRLlogHPD	0.1548	1	0.694009
FALSE Cont:logdist	9.6954	4	0.045883 *
FALSE Cont:logHPDdiff	4.9639	4	0.291025
FALSE Cont:RD1Kdiff	16.8636	4	0.002054 **
FALSE Cont:RD50Kdiff	2.9193	4	0.571427
FALSE ---			

FALSE Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### Because there may be some colinearity issues in the explanatory variables it is not reliable to pic.  
 ### using the highest p-value, therefore I will have to try removing all possibilities and proceeding w

```
##### int Cont:RD50Kdiff
```

```
mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_1, remove = "Cont:RD50Kdiff")
```

```
##### int Cont:RD1Kdiff
```

```
mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_1, remove = "Cont:RD1Kdiff")
```

```
##### int Cont:logHPDdiff
```

```
mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_1, remove = "Cont:logHPDdiff")
```

```
##### int Cont:logdist
```

```
mod_sim4 <- Permuted_model_simplification(Permuted_data,model1 = Model_1, remove = "Cont:logdist")
```

```
Model_simp <- rbind(mod_sim1,mod_sim2,mod_sim3,mod_sim4)
```

```
Model_simp
```

FALSE	DROP	Percentile
FALSE Cont:RD50Kdiff	TRUE	0.24
FALSE Cont:RD1Kdiff	FALSE	1.00
FALSE Cont:logHPDdiff	TRUE	0.74
FALSE Cont:logdist	TRUE	0.71

### Could drop all but Cont:logdist and Cont:S2logHPD shows the lowest probability of significantly low  
 ### power of the model

```
Model_7 <- update(Model_1, ~.-Cont:RD50Kdiff)
```

```
Anova(Model_7, type = "II")
```

FALSE Analysis of Deviance Table (Type II Wald chisquare tests)

FALSE

FALSE Response: logitOver

FALSE	Chisq	Df	Pr(>Chisq)
FALSE Cont	247.5349	4	< 2.2e-16 ***
FALSE logdist	5.9453	1	0.0147565 *
FALSE rt3env	1.7067	1	0.1914081
FALSE logHPDdiff	1.1812	1	0.2771160
FALSE sqrtS2RD1K	0.7908	1	0.3738580
FALSE S2logHPD	0.5244	1	0.4689561
FALSE RD1Kdiff	10.9054	1	0.0009588 ***
FALSE RD50Kdiff	0.4170	1	0.5184401
FALSE sqrtS2RD50K	0.1542	1	0.6945107
FALSE CNTRLlogHPD	0.1740	1	0.6766030
FALSE Cont:logdist	13.9315	4	0.0075170 **
FALSE Cont:logHPDdiff	6.1261	4	0.1899287
FALSE Cont:RD1Kdiff	25.2816	4	4.416e-05 ***



```

FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##### RD50Kdiff

mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_7, remove = "RD50Kdiff")

##### int Cont:RD1Kdiff

mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_7, remove = "Cont:RD1Kdiff")

##### int Cont:logHPDdiff

mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_7, remove = "Cont:logHPDdiff")

##### int Cont:logdist

mod_sim4 <- Permuted_model_simplification(Permuted_data,model1 = Model_7, remove = "Cont:logdist")

Model_simp2 <- rbind(mod_sim1,mod_sim2,mod_sim3,mod_sim4)

Model_simp2

FALSE          DROP Percentile
FALSE RD50Kdiff      TRUE          0.42
FALSE Cont:RD1Kdiff  FALSE          1.00
FALSE Cont:logHPDdiff TRUE          0.70
FALSE Cont:logdist   TRUE          0.94

### remove Cont:sqrtS2RD1k

Model_8 <- update(Model_7, ~.-RD50Kdiff)

Anova(Model_8, type = "III")

FALSE Analysis of Deviance Table (Type III Wald chisquare tests)
FALSE
FALSE Response: logitOver
FALSE          Chisq Df Pr(>Chisq)
FALSE (Intercept)    0.2775  1  0.598341
FALSE Cont          121.3469  4  < 2.2e-16 ***
FALSE logdist        1.8910  1  0.169092
FALSE rt3env         1.6303  1  0.201661
FALSE logHPDdiff     0.3360  1  0.562120
FALSE sqrtS2RD1K     0.8029  1  0.370224
FALSE S2logHPD       0.5673  1  0.451343
FALSE RD1Kdiff       0.2627  1  0.608291
FALSE sqrtS2RD50K    0.0631  1  0.801655
FALSE CNTRLlogHPD    0.2420  1  0.622732
FALSE Cont:logdist   14.8280  4  0.005072 **
FALSE Cont:logHPDdiff  6.8065  4  0.146472
FALSE Cont:RD1Kdiff  32.2127  4  1.731e-06 ***
FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
##### int Cont:RD1Kdiff

mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_8, remove = "Cont:RD1Kdiff")

##### int Cont:logHPDdiff

mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_8, remove = "Cont:logHPDdiff")

##### int Cont:logdist

mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_8, remove = "Cont:logdist")

##### S2 RD50k

mod_sim4 <- Permuted_model_simplification(Permuted_data,model1 = Model_8, remove = "sqrtS2RD50K")

Model_simp3 <- rbind(mod_sim1,mod_sim2,mod_sim3,mod_sim4)

Model_simp3

FALSE          DROP Percentile
FALSE Cont:RD1Kdiff  FALSE      1.00
FALSE Cont:logHPDdiff TRUE      0.76
FALSE Cont:logdist   TRUE      0.95
FALSE sqrtS2RD50K    TRUE      0.40

### remove interaction Cont:RD50Kdiff

Model_9 <- update(Model_8, ~.-sqrtS2RD50K)

Anova(Model_9, type = "III")

FALSE Analysis of Deviance Table (Type III Wald chisquare tests)
FALSE
FALSE Response: logitOver
FALSE          Chisq Df Pr(>Chisq)
FALSE (Intercept)    0.2361 1  0.627013
FALSE Cont          122.5652 4 < 2.2e-16 ***
FALSE logdist        1.8156 1  0.177832
FALSE rt3env         1.8658 1  0.171962
FALSE logHPDdiff      0.4545 1  0.500183
FALSE sqrtS2RD1K      0.8235 1  0.364153
FALSE S2logHPD        0.6121 1  0.433997
FALSE RD1Kdiff        0.2407 1  0.623709
FALSE CNTRLlogHPD     0.3928 1  0.530809
FALSE Cont:logdist    15.5668 4  0.003659 **
FALSE Cont:logHPDdiff  7.3204 4  0.119895
FALSE Cont:RD1Kdiff   32.4916 4  1.518e-06 ***
FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##### int Cont:RD1Kdiff

mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_9, remove = "Cont:RD1Kdiff")
```

```
##### int Cont:logHPDdiff

mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_9, remove = "Cont:logHPDdiff")

##### int Cont:logdist
mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_9, remove = "Cont:logdist")

##### S2 RD50k
Model_simp4 <- rbind(mod_sim1,mod_sim2,mod_sim3)

Model_simp4

FALSE          DROP Percentile
FALSE Cont:RD1Kdiff  FALSE      1.00
FALSE Cont:logHPDdiff TRUE      0.82
FALSE Cont:logdist   FALSE      0.98

#### Can remove RD50Kdiff

Model_10 <- update(Model_9, ~.-Cont:logHPDdiff)

Anova(Model_10, type = "III")

FALSE Analysis of Deviance Table (Type III Wald chisquare tests)
FALSE
FALSE Response: logitOver
FALSE          Chisq Df Pr(>Chisq)
FALSE (Intercept)    0.3723  1    0.5418
FALSE Cont          148.0974  4 < 2.2e-16 ***
FALSE logdist        1.3852  1    0.2392
FALSE rt3env         0.4695  1    0.4932
FALSE logHPDdiff     0.5414  1    0.4618
FALSE sqrtS2RD1K     1.5450  1    0.2139
FALSE S2logHPD       0.3226  1    0.5700
FALSE RD1Kdiff       0.3265  1    0.5677
FALSE CNTRLlogHPD    0.0963  1    0.7564
FALSE Cont:logdist   10.2911  4    0.0358 *
FALSE Cont:RD1Kdiff  34.0057  4  7.432e-07 ***
FALSE ---
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##### int Cont:RD1Kdiff

mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_10, remove = "Cont:RD1Kdiff")

##### int Cont:logHPDdiff

mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_10, remove = "logHPDdiff")

##### int Cont:logdist
mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_10, remove = "Cont:logdist")

##### S2 RD50k
```

```
Model_simp5 <- rbind(mod_sim1,mod_sim2,mod_sim3)
```

```
Model_simp5
```

```
FALSE          DROP Percentile
FALSE Cont:RD1Kdiff FALSE      1.00
FALSE logHPDdiff  TRUE       0.60
FALSE Cont:logdist TRUE       0.95
```

```
## Can remove logHPDdiff
```

```
Model_11 <- update(Model_10, ~.-logHPDdiff)
```

```
Anova(Model_11, type = "III")
```

```
FALSE Analysis of Deviance Table (Type III Wald chisquare tests)
```

```
FALSE
```

```
FALSE Response: logitOver
```

```
FALSE          Chisq Df Pr(>Chisq)
FALSE (Intercept)    0.5557 1  0.45601
FALSE Cont          148.8378 4 < 2.2e-16 ***
FALSE logdist        1.2784 1  0.25821
FALSE rt3env         0.1444 1  0.70395
FALSE sqrtS2RD1K     2.2293 1  0.13541
FALSE S2logHPD        0.0149 1  0.90290
FALSE RD1Kdiff        0.6023 1  0.43770
FALSE CNTRLlogHPD     0.2225 1  0.63715
FALSE Cont:logdist   12.1622 4  0.01619 *
FALSE Cont:RD1Kdiff  33.5129 4  9.378e-07 ***
```

```
FALSE ---
```

```
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##### int Cont:RD1Kdiff
```

```
mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_11, remove = "Cont:RD1Kdiff")
```

```
##### int Cont:logHPDdiff
```

```
mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_11, remove = "S2logHPD")
```

```
##### int Cont:logdist
```

```
mod_sim3 <- Permuted_model_simplification(Permuted_data,model1 = Model_11, remove = "Cont:logdist")
```

```
##### S2 RD50k
```

```
Model_simp6 <- rbind(mod_sim1,mod_sim2,mod_sim3)
```

```
Model_simp6
```

```
FALSE          DROP Percentile
FALSE Cont:RD1Kdiff FALSE      1.00
FALSE S2logHPD      TRUE       0.21
FALSE Cont:logdist  FALSE      0.97
```

```
##can remove S2logHPD
```

```
Model_12 <- update(Model_11, ~.-S2logHPD)
```

```
Anova(Model_12, type = "III")
```

```
FALSE Analysis of Deviance Table (Type III Wald chisquare tests)
```

```
FALSE
```

```
FALSE Response: logitOver
```

```
FALSE          Chisq Df Pr(>Chisq)
FALSE (Intercept)    0.5536  1  0.456852
FALSE Cont          149.3771  4 < 2.2e-16 ***
FALSE logdist        1.2817  1  0.257591
FALSE rt3env         0.1701  1  0.679981
FALSE sqrtS2RD1K     2.2337  1  0.135032
FALSE RD1Kdiff       0.6284  1  0.427933
FALSE CNTRLlogHPD    0.3867  1  0.534045
FALSE Cont:logdist   14.2856  4  0.006437 **
FALSE Cont:RD1Kdiff  33.8470  4  8.01e-07 ***
```

```
FALSE ---
```

```
FALSE Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##### int Cont:RD1Kdiff
```

```
mod_sim1 <- Permuted_model_simplification(Permuted_data,model1 = Model_12, remove = "Cont:RD1Kdiff")
```

```
##### int Cont:logdist
```

```
mod_sim2 <- Permuted_model_simplification(Permuted_data,model1 = Model_12, remove = "Cont:logdist")
```

```
Model_simp7 <- rbind(mod_sim1,mod_sim2)
```

```
Model_simp7
```

```
FALSE          DROP Percentile
```

```
FALSE Cont:RD1Kdiff FALSE          1.00
```

```
FALSE Cont:logdist  FALSE          0.99
```

Finally testing the removal of all other fixed effects and interactions in the model I have found that `Model_12` is the best model where these removals result in a model with significantly less explanatory power based on the permuted datasets.

Looking more closely at the model output we can see that the similarity of communities compared to that of primary minimally used sites

```
summary(Model_12)
```

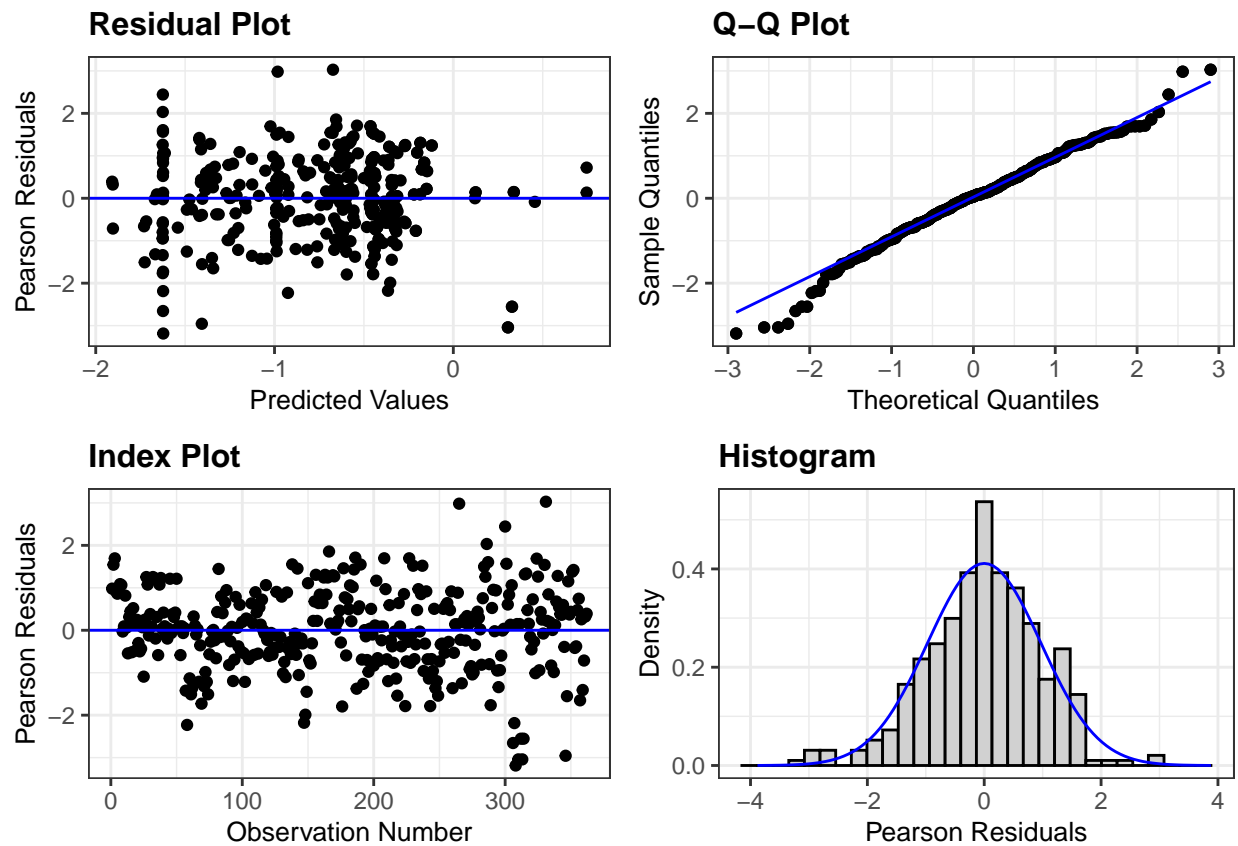
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: logitOver ~ Cont + logdist + rt3env + sqrtS2RD1K + RD1Kdiff +
##          CNTRLlogHPD + (1 | SS) + Cont:logdist + Cont:RD1Kdiff
## Data: TPD_data
##
## REML criterion at convergence: 365.3
##
```

```

## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.1855 -0.5842  0.0422  0.6402  3.0280
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   SS       (Intercept) 0.3710    0.6091
##   Residual                0.1322    0.3636
## Number of obs: 362, groups: SS, 7
##
## Fixed effects:
##                                     Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)                      0.51686    0.69466    4.48942   0.744 0.493922
## ContPriMin-SecMin                -0.91420    0.08374   338.22444 -10.917 < 2e-16
## ContPriMin-Cropland              -1.54807    0.15551   343.99428  -9.955 < 2e-16
## ContPriMin-Primary               -0.07946    0.08411   342.92231  -0.945 0.345449
## ContPriMin-SecLig                -0.62016    0.09482   343.88519  -6.540 2.22e-10
## logdist                         -0.07469    0.06597   316.48875  -1.132 0.258448
## rt3env                          0.01258    0.03049   343.02888   0.412 0.680238
## sqrtS2RD1K                      0.07898    0.05284   343.75379   1.495 0.135949
## RD1Kdiff                        -0.03968    0.05005   341.03832  -0.793 0.428484
## CNTRLlogHPD                     -0.13184    0.21201    4.23739  -0.622 0.565947
## ContPriMin-SecMin:logdist        -0.09668    0.07930   313.13892  -1.219 0.223678
## ContPriMin-Cropland:logdist     -0.13063    0.10103   342.66253  -1.293 0.196895
## ContPriMin-Primary:logdist       0.25287    0.11187   234.38267   2.260 0.024714
## ContPriMin-SecLig:logdist       -0.19445    0.18472   343.84472  -1.053 0.293214
## ContPriMin-SecMin:RD1Kdiff       0.17954    0.05418   339.64868   3.314 0.001020
## ContPriMin-Cropland:RD1Kdiff    -0.27787    0.11614   340.08438  -2.392 0.017278
## ContPriMin-Primary:RD1Kdiff     0.10464    0.18058   214.95750   0.579 0.562863
## ContPriMin-SecLig:RD1Kdiff      0.22724    0.05795   339.62139   3.921 0.000107
##
## (Intercept)
## ContPriMin-SecMin                ***
## ContPriMin-Cropland              ***
## ContPriMin-Primary
## ContPriMin-SecLig                ***
## logdist
## rt3env
## sqrtS2RD1K
## RD1Kdiff
## CNTRLlogHPD
## ContPriMin-SecMin:logdist
## ContPriMin-Cropland:logdist
## ContPriMin-Primary:logdist      *
## ContPriMin-SecLig:logdist
## ContPriMin-SecMin:RD1Kdiff      **
## ContPriMin-Cropland:RD1Kdiff    *
## ContPriMin-Primary:RD1Kdiff
## ContPriMin-SecLig:RD1Kdiff      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 18 > 12.

```

```
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
resid_panel(Model_12)
```



```
write_rds(file = "Outputs/TPD_Overlap_GLMM.rds", Model_12)
```