big_structures_analyses

2025-06-29

Load the libraries

```
library(tidyverse)
library(lme4)
library(glmmTMB)
library(mgcv)
library(DHARMa)
library(sjPlot)
library(MASS)
library(GGally)
library(emmeans)
library(performance)
library(patchwork)
```

Load the data

```
setwd("C:/Users/User/Desktop/Git/CRG_spider_nurseries")

df <- read_csv("./Datasets/big_spiders.csv")
paleta <- c("#192639","#4EBCB8", "#FA9938", "#F4F4F5")
```

Format the data

```
df<-df |>
  mutate(structure=as.factor(structure),
         structure_fixed=as.factor(structure_fixed),
         site=as.factor(site),
         face=as.factor(face),
         origin=as.factor(origin),
         data_entry_person=as.factor(data_entry_person),
         month=as.factor(month),
         year=as.factor(year),
         .keep = "unused")

str(df)
```

```
## tibble [8,182 x 16] (S3: tbl_df/tbl/data.frame)
##  $ photo.code       : chr [1:8182] "DSCN8590" "DSCN8591" "DSCN8592" "DSCN8594" ...
##  $ site             : Factor w/ 3 levels "Guiri","Playa Blanca",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ origin           : Factor w/ 3 levels "Jicaro","Marina",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ month            : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year             : Factor w/ 4 levels "2021","2022",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ area.cm2         : num [1:8182] 148.5 90.3 169.5 122.7 100.5 ...
```

```
##  $ num.frag         : num [1:8182] 6 4 7 4 4 4 17 6 5 5 ...
##  $ mort.            : num [1:8182] 0 0 0 0 0 0 0 0 0 0 ...
##  $ mort.parc        : num [1:8182] 0 1 1 1 0 1 0 0 2 2 ...
##  $ perd             : num [1:8182] 0 0 0 0 0 0 0 0 0 0 ...
##  $ blanq            : num [1:8182] 0 0 0 0 0 0 0 0 0 0 ...
##  $ structure        : Factor w/ 56 levels "10","101","102",..: 28 28 28 28 28 28 28 27 27 27 ...
##  $ structure_fixed  : Factor w/ 43 levels "6","7","8","9",..: 42 42 42 42 42 42 42 41 41 41 ...
##  $ face             : Factor w/ 7 levels "1","2","3","4",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ data_entry_person: Factor w/ 15 levels "Digitador_1",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ date             : Date[1:8182], format: "2021-01-01" "2021-01-01" ...
```

```r
# glimpse(df)
```

Glance the data

```r
head(df)
```

```
## # A tibble: 6 x 16
##   photo.code site     origin month year  area.cm2 num.frag mort. mort.parc  perd
##   <chr>      <fct>    <fct>  <fct> <fct>    <dbl>    <dbl> <dbl>     <dbl> <dbl>
## 1 DSCN8590   Playa B~ Marina 1     2021     149.         6     0         0     0
## 2 DSCN8591   Playa B~ Marina 1     2021      90.3        4     0         1     0
## 3 DSCN8592   Playa B~ Marina 1     2021     170.         7     0         1     0
## 4 DSCN8594   Playa B~ Marina 1     2021     123.         4     0         1     0
## 5 DSCN8595   Playa B~ Marina 1     2021     101.         4     0         0     0
## 6 DSCN8596   Playa B~ Marina 1     2021     135.         4     0         1     0
## # i 6 more variables: blanq <dbl>, structure <fct>, structure_fixed <fct>,
## #   face <fct>, data_entry_person <fct>, date <date>
```

```r
tail(df)
```

```
## # A tibble: 6 x 16
##   photo.code site     origin month year  area.cm2 num.frag mort. mort.parc  perd
##   <chr>      <fct>    <fct>  <fct> <fct>    <dbl>    <dbl> <dbl>     <dbl> <dbl>
## 1 P6200183   Playa P~ <NA>   6     2024      890.        8     0         0     0
## 2 P6200184   Playa P~ <NA>   6     2024      925.        8     0         0     0
## 3 P6200185   Playa P~ <NA>   6     2024      863.        7     0         0     0
## 4 P6200186   Playa P~ <NA>   6     2024      881.        7     0         0     0
## 5 P6200187   Playa P~ <NA>   6     2024     1254.        8     0         0     0
## 6 P6200188   Playa P~ <NA>   6     2024     1997.       23     0         0     0
## # i 6 more variables: blanq <dbl>, structure <fct>, structure_fixed <fct>,
## #   face <fct>, data_entry_person <fct>, date <date>
```

```r
min(df$area.cm2)
```

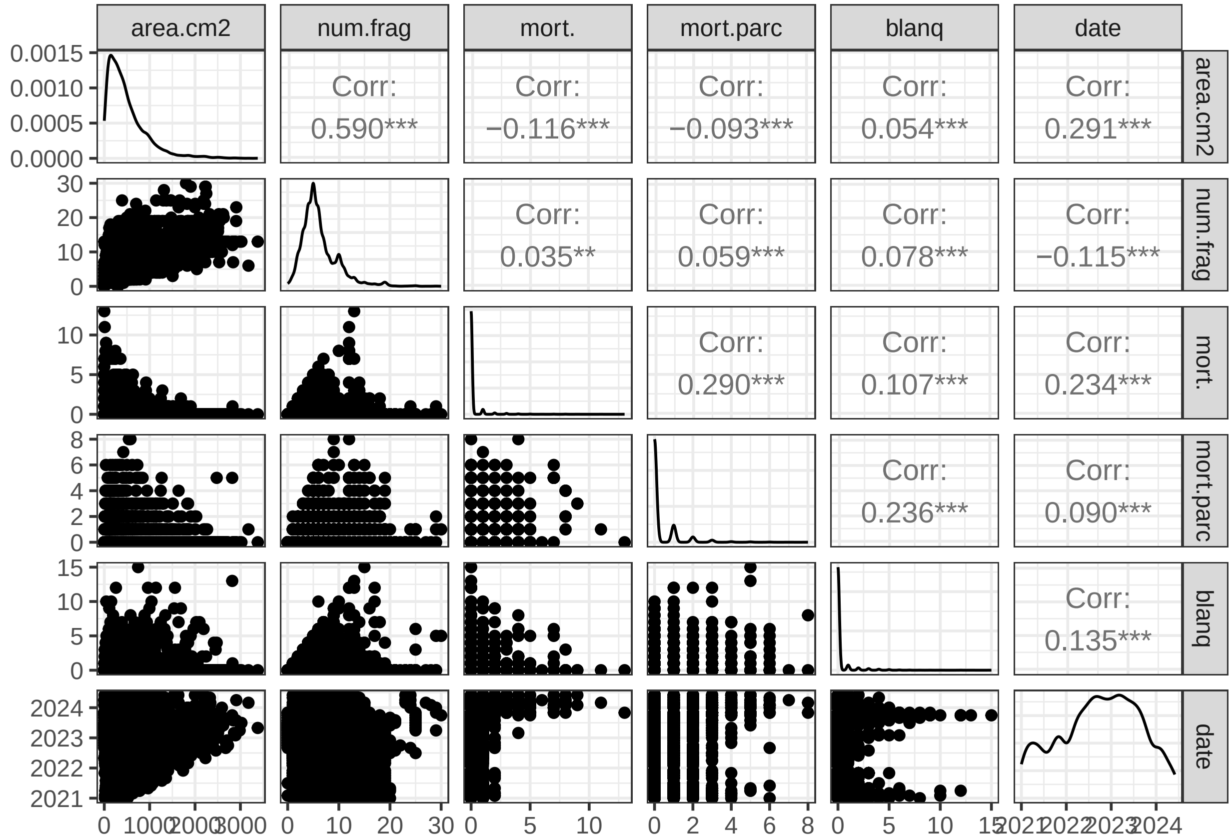
```
## [1] 0
```

```r
max(df$area.cm2)
```

```
## [1] 3381.91
```

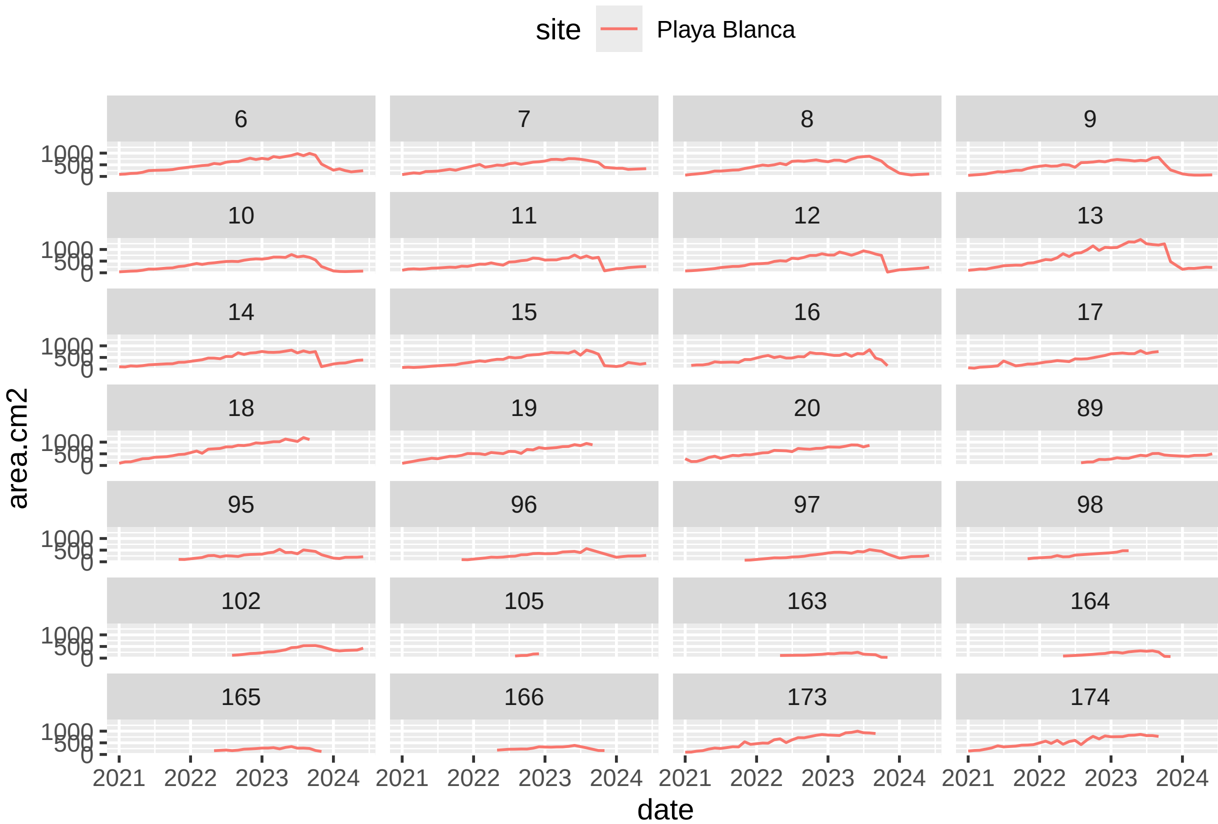
Some exploratory graphs

```
df |>
  dplyr::select(area.cm2, num.frag, mort., mort.parc, blanq, date) |>
  ggpairs()+theme_bw()
```



Area of each structure by site.

```
df %>%
  filter(site=="Playa Blanca")  |>
  ggplot(aes(x=date, y=area.cm2, col=site)) +
  stat_summary(geom = "line", fun = "mean") +
  facet_wrap(~structure_fixed, ncol = 4) +
  theme(legend.position = "top")
```




```r
df %>%
  filter(site=="Playa Pelonas")  |>
  ggplot(aes(x=date, y=area.cm2, col=site)) +
  stat_summary(geom = "line", fun = "mean") +
  facet_wrap(~structure_fixed, ncol = 4) +
  theme(legend.position = "top")
```


```
df %>%
  filter(site=="Guiri")  |>
  ggplot(aes(x=date, y=area.cm2, col=site)) +
  stat_summary(geom = "line", fun = "mean") +
  facet_wrap(~structure_fixed, ncol = 4) +
  theme(legend.position = "top")
```

As we can see, some structures have few records, so those will be filter out

```r
df <- df |>
  filter(structure_fixed!="105")
```

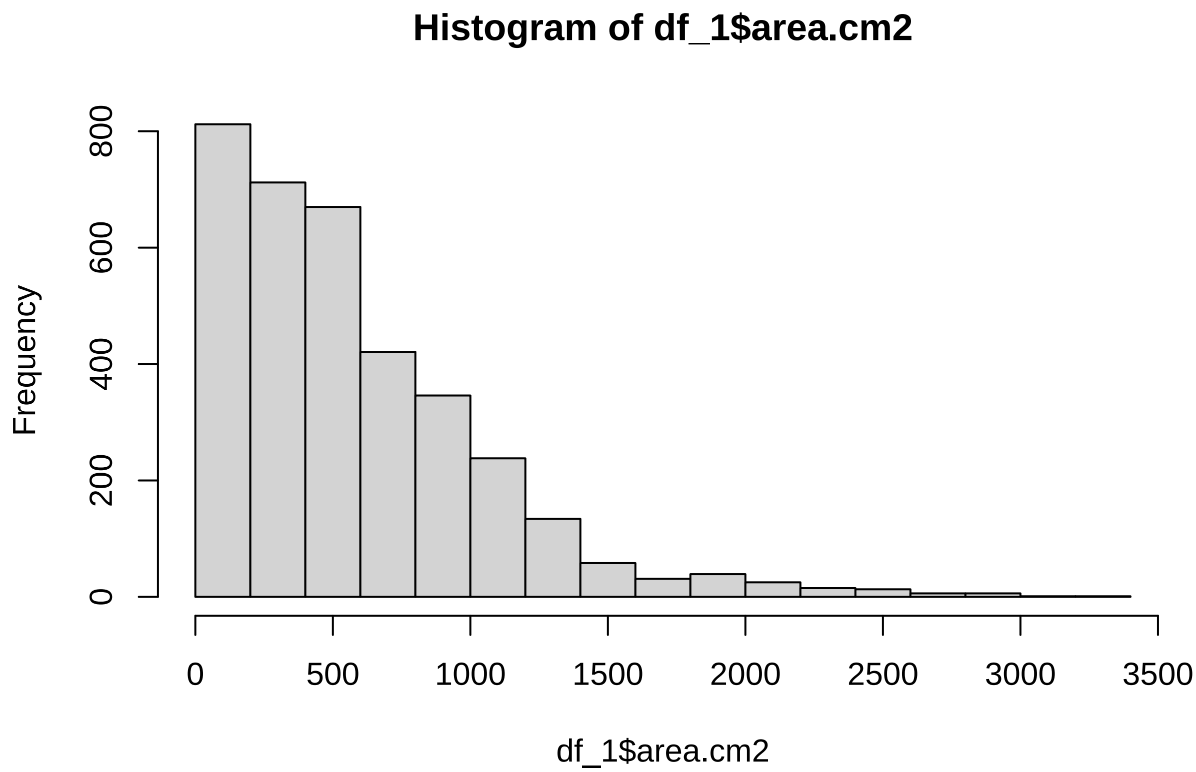
Many other structures have numerous observations but they do not comprehend all the entire time. Thus, only structures with data from january 2021 to may 2024 (because there's no data for june, from Güiri)

```r
df_1 <- df |>
  filter(as.numeric(as.character(structure_fixed))
         %in% c(6:15, 169:172)) |>
  filter(date<="2024-05-01") |>
  droplevels()
```

Model the data

First, we need to check the distribution of the data.

```r
hist(df_1$area.cm2)
```




```r
shapiro.test(df_1$area.cm2) # prueba normalidad
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df_1$area.cm2
## W = 0.87714, p-value < 2.2e-16
```

The data is not normally distributed, so we need to use a different model. Testing different models to find the better one.

linear model

This model is not appropriate because it does not fit the data. But it is used as a comparison with the other models.

```r
m0 <- lm(area.cm2 ~ date + site, data = df_1)
```

A more appropiate model could be the gamma model; however, the data contains zeros and gamma models do not handle zeros. So, a tweedie model is used instead.

```r
m1 <- glmmTMB(area.cm2 ~ date + site,
              data = df_1,
              family = tweedie(link = "log"))
```

The data also contains observations through time, so a random effect is added to the model.

```r
m2 <- glmmTMB(area.cm2 ~ date + site + (1|structure_fixed),
                 data = df_1,
                 family = tweedie())
```

Since the data is not linear, a polynomial term is added to the model.

```r
m3 <- glmmTMB(area.cm2 ~ poly(date, 2) + site + (1|structure_fixed),
                 data = df_1,
                 family = tweedie())
```

As the data includes zeros, a zero-inflated model could be used.

```r
m4 <- glmmTMB(area.cm2 ~ poly(date, 2) + site + (1|structure_fixed),
                 data = df_1,
                 ziformula = ~ 1,
                 family = tweedie())
```

Test for differences in the slopes using an interaction term.

```r
m5 <- glmmTMB(area.cm2 ~ poly(date, 2) * site + (1|structure_fixed),
                 data = df_1,
                 ziformula = ~ 1,
                 family = tweedie())
```

And to compare, a GAM model.

```r
m6 <- gam(area.cm2 ~ s(as.numeric(date), k = 5) + site + s(structure_fixed, bs = "re"),
                 data = df_1,
                 family = tw(link = "log"))
```

Another possible model is the Negative Binomial.

```r
m7 <- glmmTMB(area.cm2 ~ poly(date, 2) + site + (1|structure_fixed),
                 data = df_1,
                 ziformula = ~ 1,
                 family = nbinom2())
```

Now, we can compare all the models using AIC and BIC.

```r
AIC(m0, m1, m2, m3, m4, m5, m6, m7)
```

```
##           df      AIC
## m0   4.00000 52382.91
## m1   5.00000 50371.09
```

```
## m2  6.00000 50256.03
## m3  7.00000 48868.46
## m4  8.00000 48657.79
## m5 10.00000 48437.19
## m6 19.49717 48744.62
## m7  7.00000 48711.20
```

```r
BIC(m0, m1, m2, m3, m4, m5, m6, m7)
```

```
##           df      BIC
## m0  4.00000 52407.59
## m1  5.00000 50401.93
## m2  6.00000 50293.04
## m3  7.00000 48911.64
## m4  8.00000 48707.14
## m5 10.00000 48498.88
## m6 19.49717 48864.89
## m7  7.00000 48754.38
```

```r
anova(m4, m5, test = "Chisq")
```

```
## Data: df_1
## Models:
## m4: area.cm2 ~ poly(date, 2) + site + (1 | structure_fixed), zi=~1, disp=~1
## m5: area.cm2 ~ poly(date, 2) * site + (1 | structure_fixed), zi=~1, disp=~1
##    Df   AIC   BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m4  8 48658 48707 -24321    48642
## m5 10 48437 48499 -24209    48417 224.6      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both AIC and BIC indicate that m5 model is the best. This is the model with the interaction term. The anova test also favors the this model.

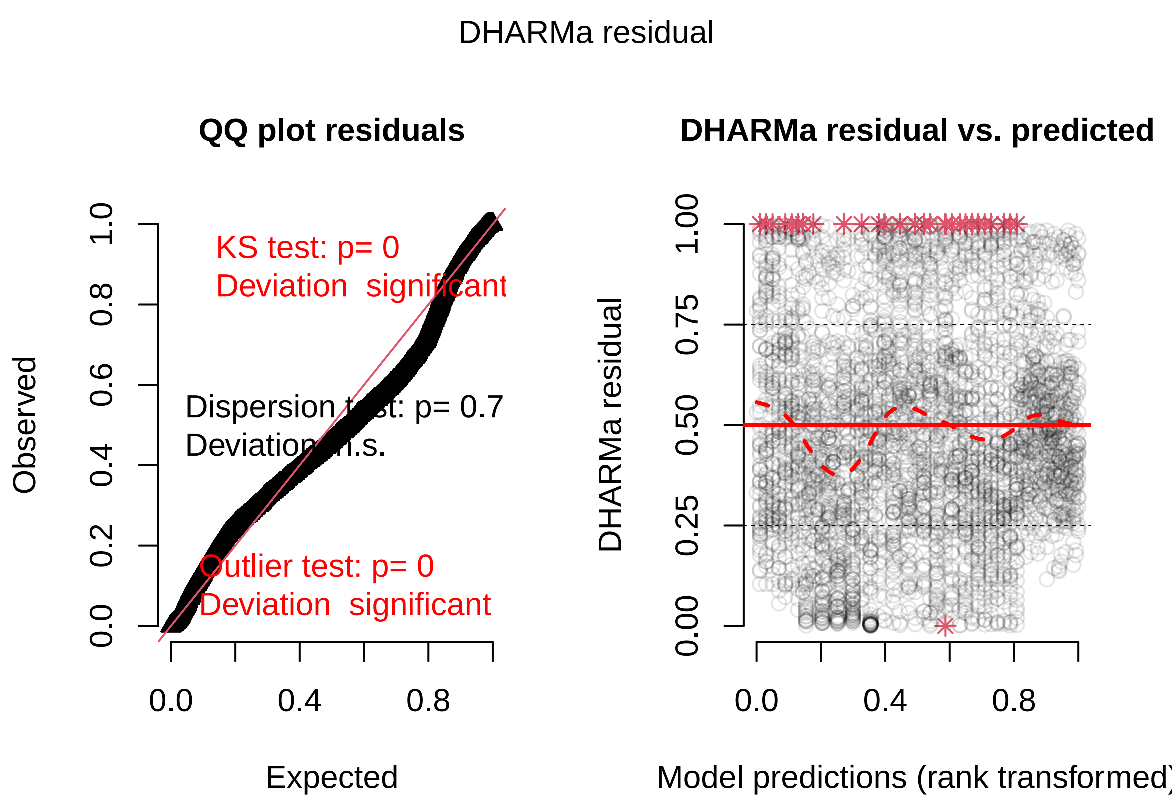
*Note: gamma models were tested by aggregating the data and calculating the mean area per structure, site and date (because gamma does not accept zeros). However, when comparing models those models, the tweedie model was the best one. That is, tweedie model was also fitted to the aggregated data to compare properly with gamma models. Those models showed the lowest AIC and BIC values but since they used summarised data the models with all observations were kept instead.

```r
summary(m5)
```

```
##  Family: tweedie  ( log )
## Formula:          area.cm2 ~ poly(date, 2) * site + (1 | structure_fixed)
## Zero inflation:            ~1
## Data: df_1
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##   48437.2   48498.9  -24208.6   48417.2      3518
##
## Random effects:
##
```

```
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  structure_fixed (Intercept) 0.01801  0.1342
## Number of obs: 3528, groups:  structure_fixed, 14
##
## Dispersion parameter for tweedie family (): 1.28
##
## Conditional model:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       6.68971    0.06904   96.90   <2e-16 ***
## poly(date, 2)1                   25.19276    0.99655   25.28   <2e-16 ***
## poly(date, 2)2                  -13.99426    0.98825  -14.16   <2e-16 ***
## sitePlaya Blanca                 -0.73510    0.08179   -8.99   <2e-16 ***
## poly(date, 2)1:sitePlaya Blanca  -0.94188    1.22347   -0.77    0.441
## poly(date, 2)2:sitePlaya Blanca -18.29073    1.21404  -15.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.494      0.161  -27.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
DHARMa::simulateResiduals(fittedModel = m5, plot = TRUE)
```



```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.332 0.532 0.784 0.812 0.908 0.92 0.956 0.796 0.736 0.272 0.176 0.136 0.3 1
```

Using sjPlot to plot the model

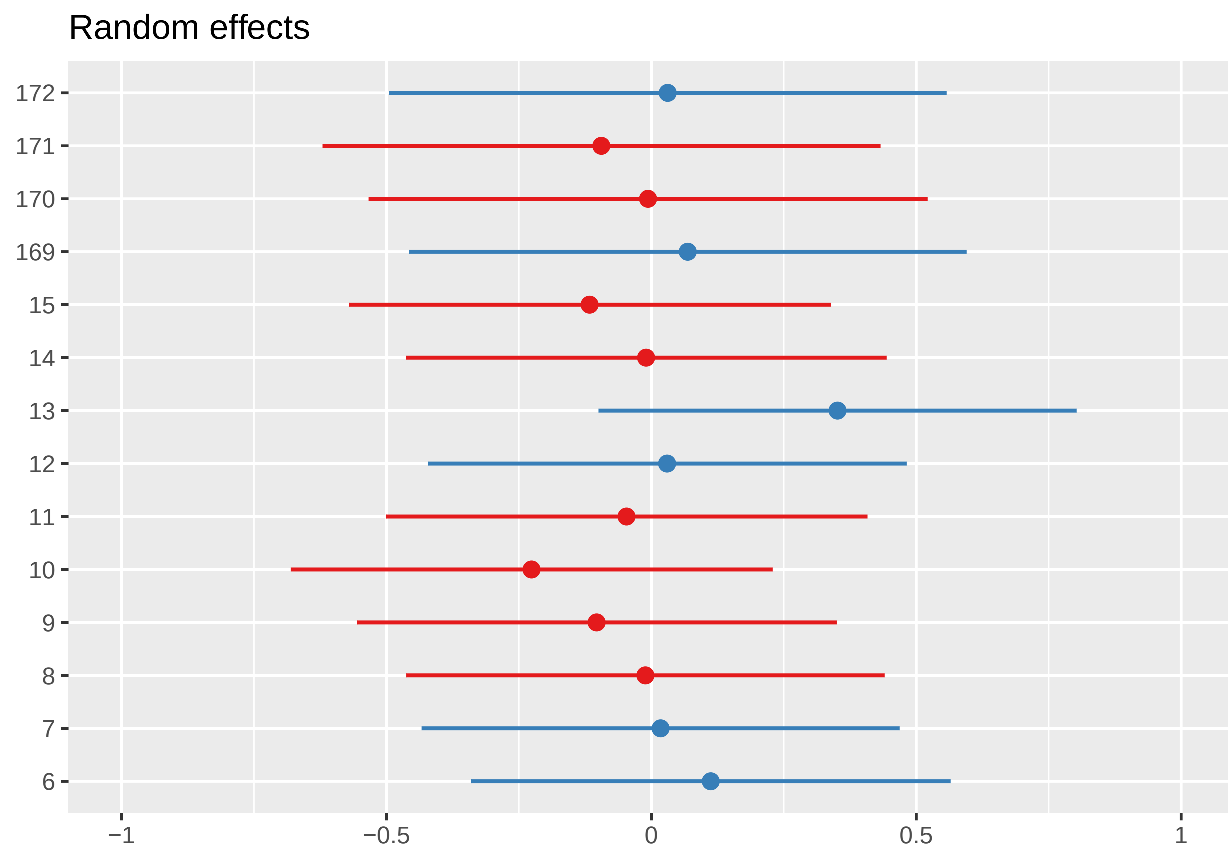
```
plot_model(m5, type = "est", transform = NULL)
```


```
# random effecrs, structure
ranef(m4)
```

```
## $structure_fixed
##      (Intercept)
## 6    0.112132489
## 7    0.017497561
## 8   -0.011228827
## 9   -0.103273622
## 10  -0.226130172
## 11  -0.046879362
## 12   0.029735927
## 13   0.351398791
## 14  -0.009897908
## 15  -0.116554381
## 169  0.068667871
## 170 -0.006257365
```

```
## 171 -0.094373898
## 172  0.030838064
```

```r
plot_model(m4, type = "re", transform = NULL)
```




```r
# If that fails, generate predictions manually:
new_data <- expand.grid(site=levels(df_1$site),
                        date = seq(as.Date("2021-01-01"),
                                   as.Date("2024-05-01"),
                                   by = "month"),
                        structure_fixed=levels(df_1$structure_fixed))

# Include poly() terms manually
new_data$poly1 <- poly(df_1$date, 2)[,1][1:1148]
new_data$poly2 <- poly(df_1$date, 2)[,2][1:1148]

# Rename for consistency with model
names(new_data)[which(names(new_data) == "poly1")] <- "poly(date, 2)1"
names(new_data)[which(names(new_data) == "poly2")] <- "poly(date, 2)2"

new_data$preds <- predict(m5, newdata = new_data, type = "response")

# Plot with ggplot2
p1 <- ggplot(new_data, aes(x = date, y = preds, col=site)) +
  stat_summary(geom = "line", fun = "mean", linewidth = 1, ) +
```

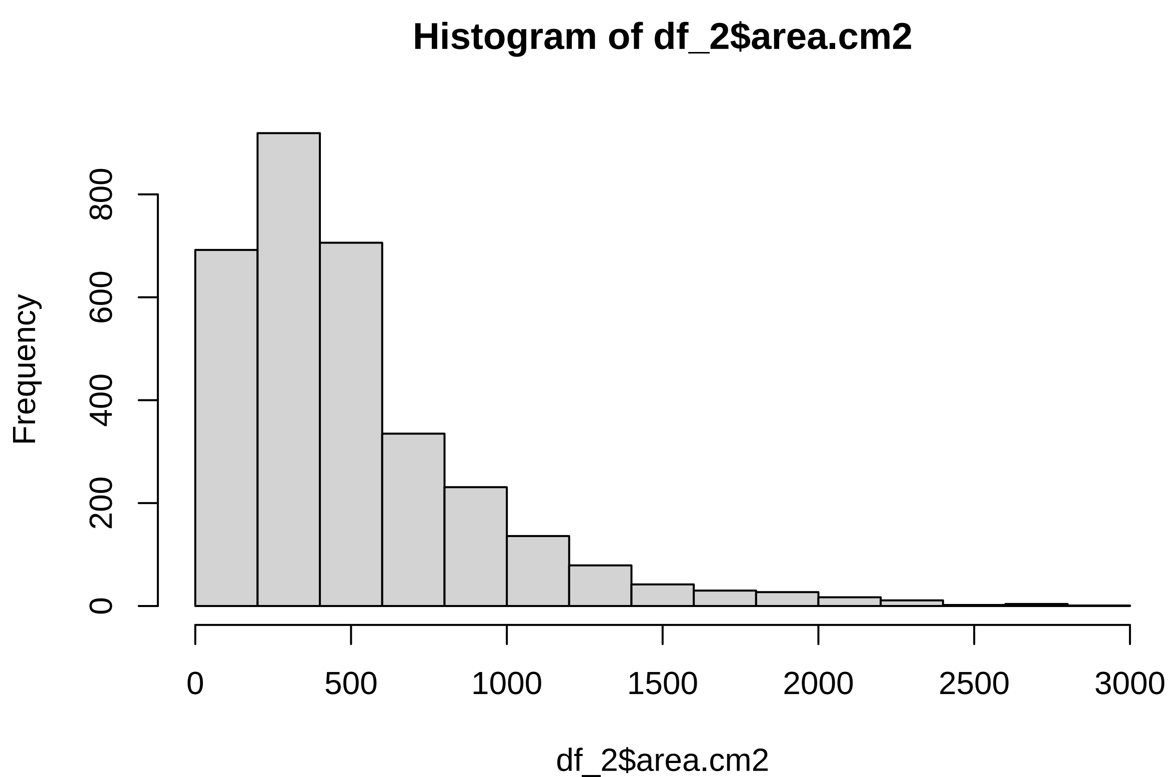
```
  labs(y = "Predicted area (cm²)", x = "Date")+
  scale_color_manual(values = paleta[c(2,3)],
                     labels = c("Güiri", "Playa Blanca"))+
  stat_summary(geom = "line", fun = "mean", data = ) +
  theme_classic()+
  theme(legend.position = "top",
        legend.title = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank())
```

Now, we need to model the data from the 3rd site which was removed before. Other sites are included are a comparison.

Filter data so all sites have the same timeframe.

```
df_2 <- df |>
  filter(date>="2022-01-01" & date<="2023-02-01") |>
  droplevels()
```

```
hist(df_2$area.cm2)
```

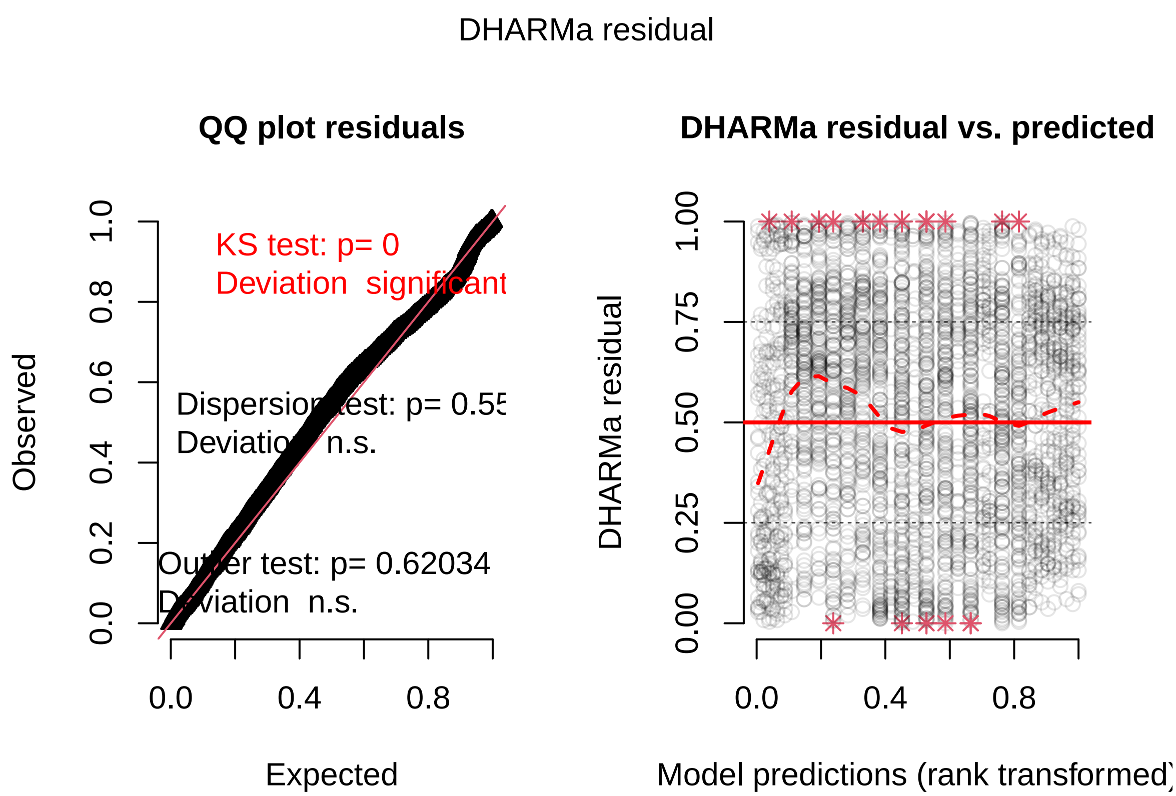



```
min(df_2$area.cm2)
```

```
## [1] 0
```

Same model as before, because data includes zeros but now with polynomial terms of 3rd degree.

```r
m8 <- glmmTMB(area.cm2 ~ poly(date, 3) * site + (1|structure_fixed),
                data = df_2,
                ziformula = ~ 1,
                family = tweedie())


DHARMa::simulateResiduals(fittedModel = m8, plot = TRUE)
```



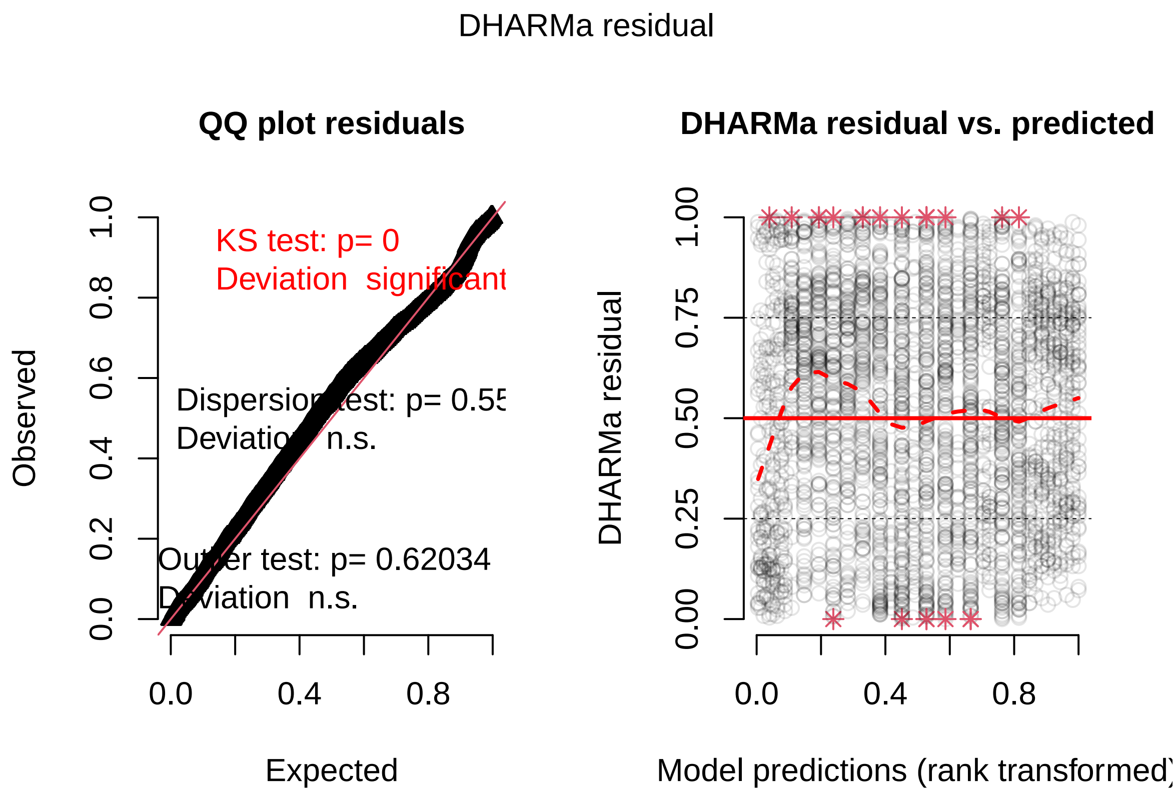

```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DH
##
## Scaled residual values: 0.792 0.756 0.364 0.512 0.94 0.72 0.976 0.7 0.516 0.58 0.772 0.596 0.676 0.98
```

```r
summary(m8)
```

```
##  Family: tweedie  ( log )
## Formula:          area.cm2 ~ poly(date, 3) * site + (1 | structure_fixed)
## Zero inflation:            ~1
## Data: df_2
##
##      AIC      BIC   logLik -2*log(L)  df.resid
##  43740.7  43838.0 -21854.3   43708.7      3216
##
```

```
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  structure_fixed (Intercept) 0.3021   0.5497
## Number of obs: 3232, groups:  structure_fixed, 43
##
## Dispersion parameter for tweedie family (): 0.459
##
## Conditional model:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          6.4333     0.1843   34.91  < 2e-16 ***
## poly(date, 3)1                       9.8537     1.1891    8.29  < 2e-16 ***
## poly(date, 3)2                      -1.2868     1.2762   -1.01 0.313311
## poly(date, 3)3                      -2.0169     1.2752   -1.58 0.113729
## sitePlaya Blanca                    -0.4674     0.2119   -2.21 0.027405 *
## sitePlaya Pelonas                   -1.0967     0.2924   -3.75 0.000177 ***
## poly(date, 3)1:sitePlaya Blanca      1.1237     1.3675    0.82 0.411253
## poly(date, 3)2:sitePlaya Blanca      2.2967     1.4349    1.60 0.109462
## poly(date, 3)3:sitePlaya Blanca      1.0340     1.4328    0.72 0.470487
## poly(date, 3)1:sitePlaya Pelonas  19.4519     2.0337    9.56  < 2e-16 ***
## poly(date, 3)2:sitePlaya Pelonas   2.9598     1.9856    1.49 0.136052
## poly(date, 3)3:sitePlaya Pelonas  -5.1586     2.0497   -2.52 0.011843 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.5922     0.2892  -19.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
m9 <- glmmTMB(area.cm2 ~ poly(date, 3) + site + (1|structure_fixed),
                 data = df_2,
                 ziformula = ~ 1,
                 family = tweedie())


DHARMa::simulateResiduals(fittedModel = m8, plot = TRUE)
```




```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.792 0.756 0.364 0.512 0.94 0.72 0.976 0.7 0.516 0.58 0.772 0.596 0.676 0.98
```

```r
summary(m8)
```

```
##  Family: tweedie  ( log )
## Formula:          area.cm2 ~ poly(date, 3) * site + (1 | structure_fixed)
## Zero inflation:            ~1
## Data: df_2
##
##      AIC       BIC    logLik -2*log(L)  df.resid
##   43740.7   43838.0  -21854.3   43708.7      3216
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  structure_fixed (Intercept) 0.3021   0.5497
## Number of obs: 3232, groups:  structure_fixed, 43
##
## Dispersion parameter for tweedie family (): 0.459
##
## Conditional model:
##                                   Estimate Std. Error z value Pr(>|z|)
```

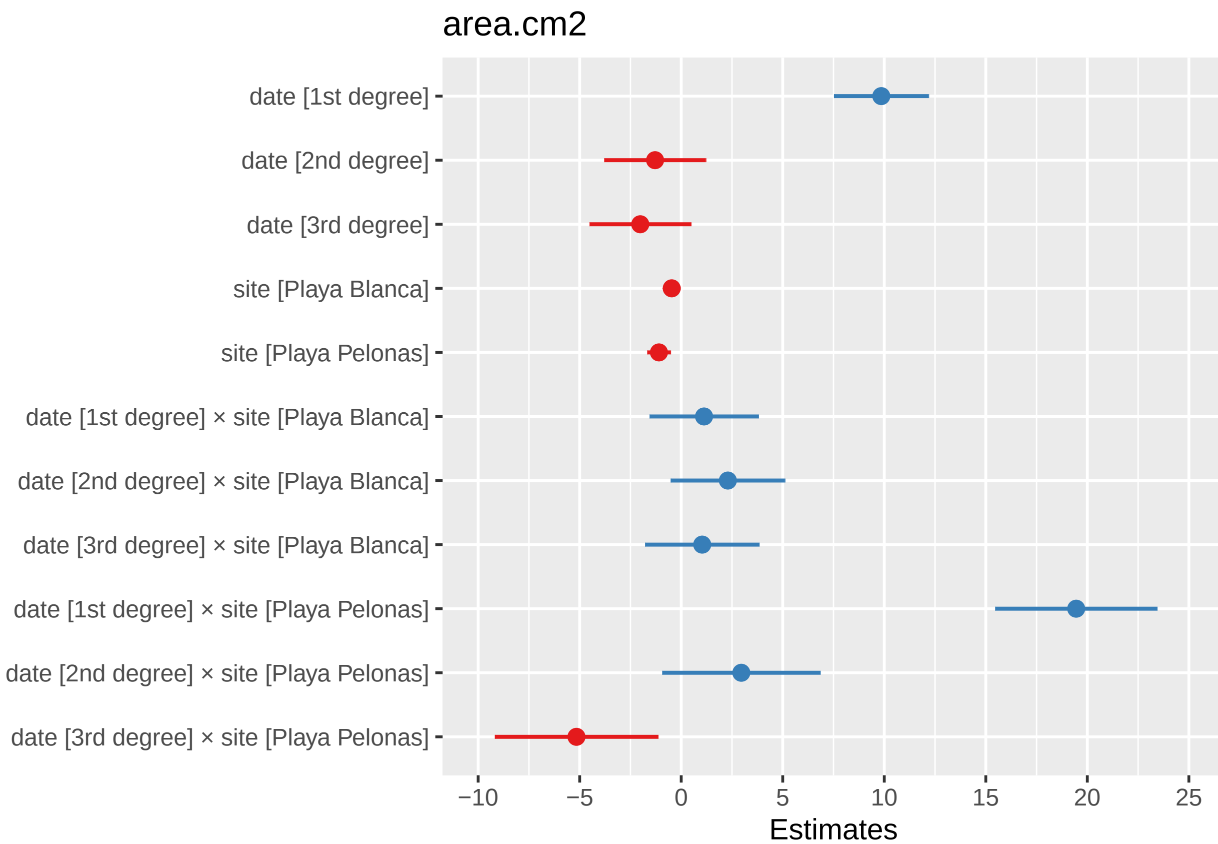
```
## (Intercept)                       6.4333     0.1843   34.91  < 2e-16 ***
## poly(date, 3)1                    9.8537     1.1891    8.29  < 2e-16 ***
## poly(date, 3)2                   -1.2868     1.2762   -1.01 0.313311
## poly(date, 3)3                   -2.0169     1.2752   -1.58 0.113729
## sitePlaya Blanca                 -0.4674     0.2119   -2.21 0.027405 *
## sitePlaya Pelonas                -1.0967     0.2924   -3.75 0.000177 ***
## poly(date, 3)1:sitePlaya Blanca    1.1237     1.3675    0.82 0.411253
## poly(date, 3)2:sitePlaya Blanca    2.2967     1.4349    1.60 0.109462
## poly(date, 3)3:sitePlaya Blanca    1.0340     1.4328    0.72 0.470487
## poly(date, 3)1:sitePlaya Pelonas  19.4519     2.0337    9.56  < 2e-16 ***
## poly(date, 3)2:sitePlaya Pelonas   2.9598     1.9856    1.49 0.136052
## poly(date, 3)3:sitePlaya Pelonas  -5.1586     2.0497   -2.52 0.011843 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.5922     0.2892  -19.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

model 8 is the best one.

```r
# If that fails, generate predictions manually:
new_data <- expand.grid(site=levels(df_2$site),
                        date = seq(as.Date("2022-01-01"),
                                   as.Date("2023-02-01"),
                                   by = "month"),
                        structure_fixed=levels(df_2$structure_fixed))

# Include poly() terms manually
new_data$poly1 <- poly(df_2$date, 3)[,1][1:1806]
new_data$poly2 <- poly(df_2$date, 3)[,2][1:1806]
new_data$poly3 <- poly(df_2$date, 3)[,3][1:1806]

# Rename for consistency with model
names(new_data)[which(names(new_data) == "poly1")] <- "poly(date, 3)1"
names(new_data)[which(names(new_data) == "poly2")] <- "poly(date, 3)2"
names(new_data)[which(names(new_data) == "poly3")] <- "poly(date, 3)3"

new_data$preds <- predict(m8, newdata = new_data, type = "response")

# Plot with ggplot2
p2 <- ggplot(new_data, aes(x = date, y = preds, col=site)) +
  stat_summary(geom = "line", fun = "mean", linewidth = 1) +
  labs(y = "Predicted area (cm²)", x = "Date")+
  scale_color_manual(values = paleta[c(2,3, 1)],)+
  theme_classic()+
  theme(legend.position = "top",
        legend.title = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank())
```

sjPlot

```r
plot_model(m8, type = "est", transform = NULL)
```

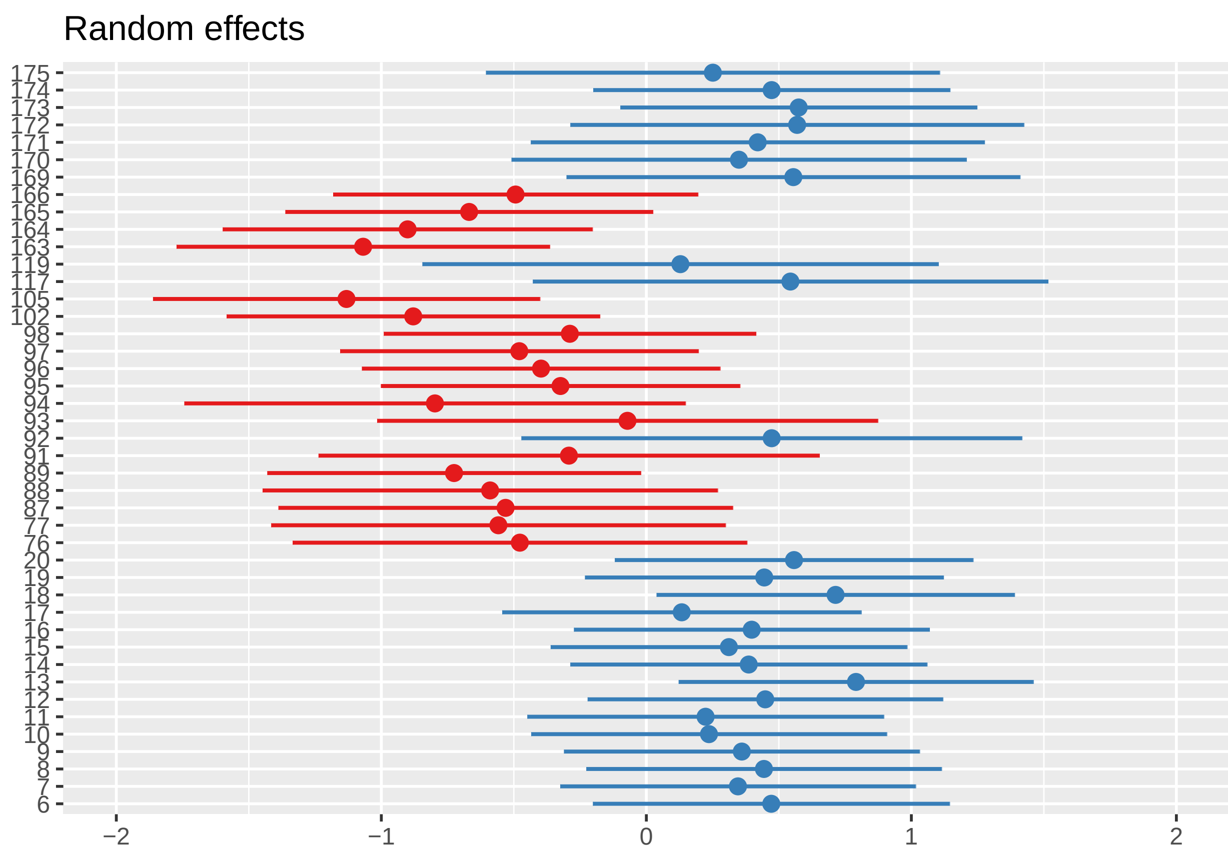



```r
# random effecrs, structure
ranef(m8)
```

```
## $structure_fixed
##     (Intercept)
## 6    0.47137262
## 7    0.34561256
## 8    0.44371107
## 9    0.36020723
## 10   0.23633592
## 11   0.22345542
## 12   0.44861340
## 13   0.79105540
## 14   0.38630319
## 15   0.31142259
## 16   0.39751327
## 17   0.13353893
## 18   0.71398710
## 19   0.44487580
## 20   0.55719509
## 76  -0.47733933
```

```
## 77  -0.55829067
## 87  -0.53113061
## 88  -0.58960134
## 89  -0.72564069
## 91  -0.29214529
## 92   0.47289398
## 93  -0.07128575
## 94  -0.79778442
## 95  -0.32406295
## 96  -0.39741303
## 97  -0.47949991
## 98  -0.28872380
## 102 -0.87967309
## 105 -1.13158767
## 117  0.54359419
## 119  0.12844826
## 163 -1.06877222
## 164 -0.90082890
## 165 -0.66872546
## 166 -0.49362914
## 169  0.55458719
## 170  0.34943812
## 171  0.42010781
## 172  0.56892539
## 173  0.57476706
## 174  0.47272196
## 175  0.25092022
```

```
plot_model(m8, type = "re", transform = NULL)
```



Anovas

```
car::Anova(m5, type = "III")
```

```
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: area.cm2
##                        Chisq Df Pr(>Chisq)
## (Intercept)         9389.537  1  < 2.2e-16 ***
## poly(date, 2)        786.959  2  < 2.2e-16 ***
## site                  80.773  1  < 2.2e-16 ***
## poly(date, 2):site   229.265  2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
car::Anova(m8, type = "III")
```

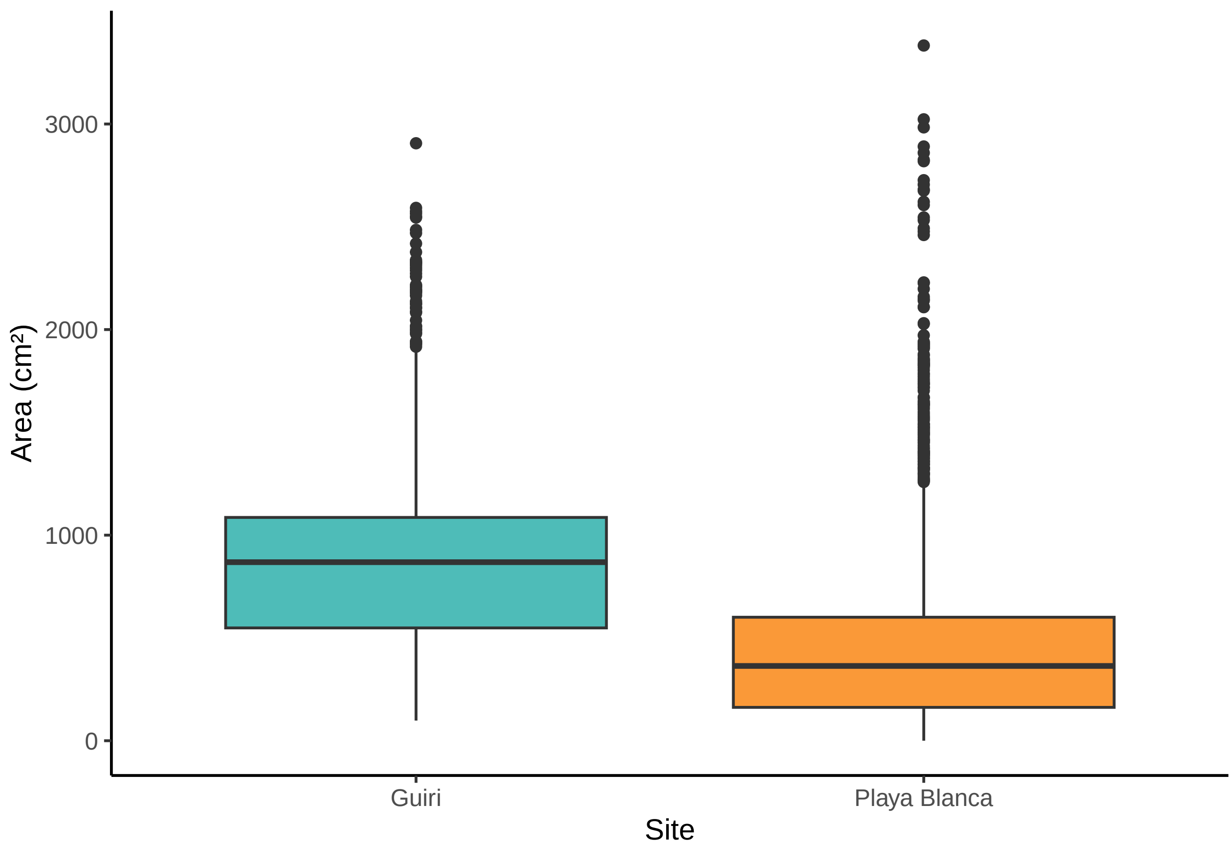
```
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: area.cm2
##                        Chisq Df Pr(>Chisq)
## (Intercept)         1218.778  1  < 2.2e-16 ***
## poly(date, 3)         83.480  3  < 2.2e-16 ***
```

```
## site                     14.096  2   0.000869 ***
## poly(date, 3):site  120.321  6   < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

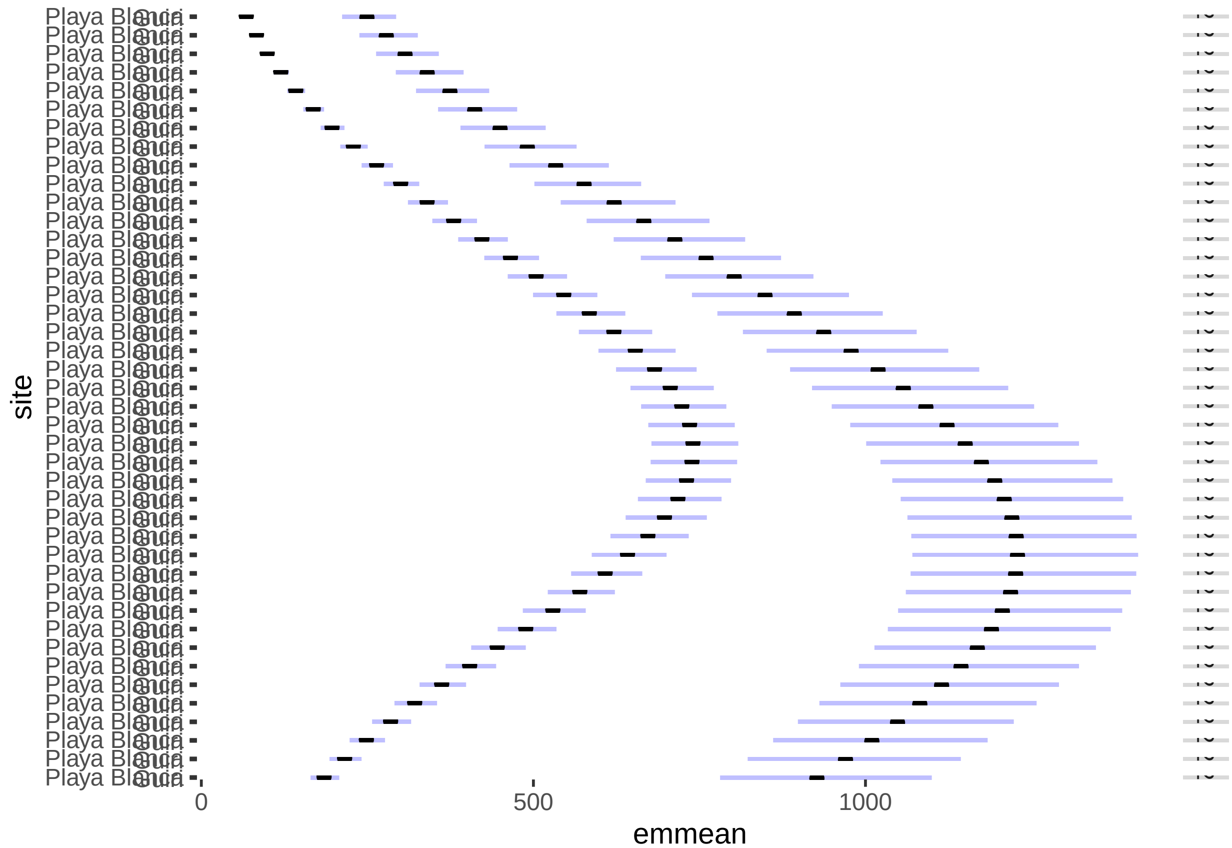
emmeans

```
emm_res <- emmeans(m5,
                  pairwise ~ site | poly(date, 2),
                  at = list(date = seq(as.Date("2021-01-01"),
                                       as.Date("2024-06-01"),
                                       by = "month")),
                  adjust = "tukey",
                  type = "response")
```

```
df_1 |> ggplot()+
  geom_boxplot(aes(x=site, y=area.cm2, fill=site)) +
  scale_fill_manual(values = paleta[c(2,3)])+
  theme_classic()+
  theme(legend.position = "none") +
  labs(x = "Site", y = "Area (cm²)")
```



```r
plot(emm_res)
```



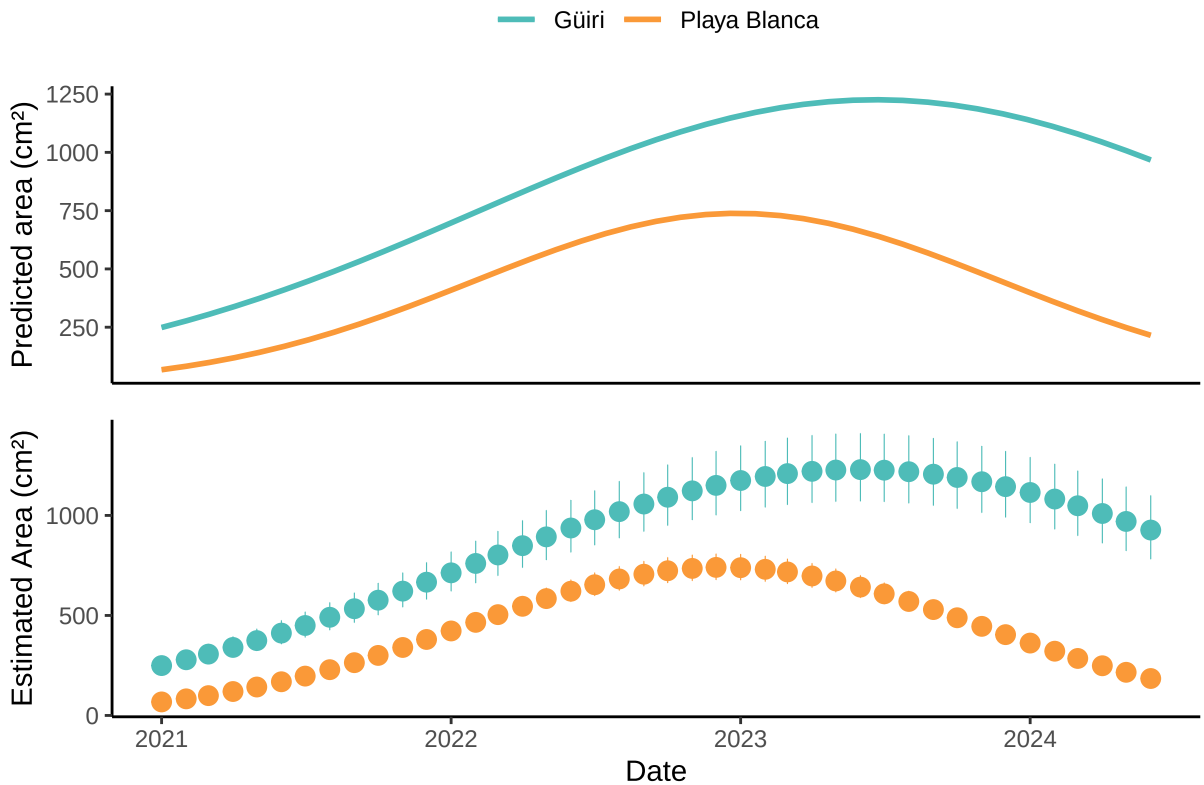

```r
as.Date(19220, origin = "1970-01-01") # to check when is R doing the comparison
```

```
## [1] "2022-08-16"
```

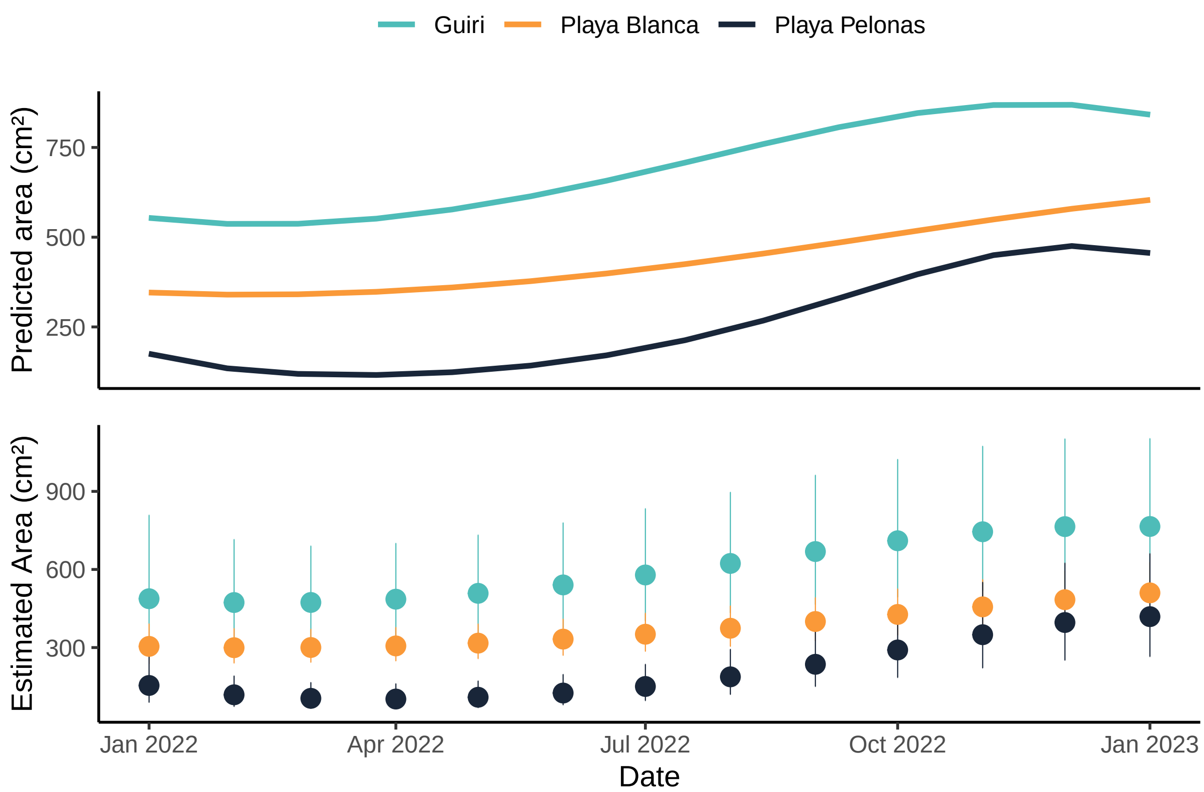
```r
emm_df <- as.data.frame(emm_res$emmeans)
emm_df$date <- as.Date(emm_df$date, origin = "1970-01-01")


p3 <- ggplot(emm_df, aes(x = date, y = response, color = site, fill = site)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, size = 0.2) +
  geom_point(size = 3) +
  scale_color_manual(values = paleta[c(2,3)],
                     labels = c("Güiri", "Playa Blanca"))+
  labs(
    #title = "Estimated Marginal Means Over Time",
    x = "Date",
    y = "Estimated Area (cm²)"
  ) +
  theme_classic() +
  theme(
    legend.position = "none"
  )
```

```
emm_res2 <- emmeans(m8,
                 pairwise ~ site | poly(date, 2),
                 at = list(date = seq(as.Date("2022-01-01"),
                                     as.Date("2023-01-01"),
                                     by = "month")),
                 adjust = "tukey",
                 type = "response")

emm_df2 <- as.data.frame(emm_res2$emmeans)
emm_df2$date <- as.Date(emm_df2$date, origin = "1970-01-01")


p4 <- ggplot(emm_df2, aes(x = date, y = response, color = site, fill = site)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, size = 0.2) +
  geom_point(size = 3) +
  scale_color_manual(values = paleta[c(2,3, 1)],
                     labels = c("Güiri", "Playa Blanca", "Playa Pelonas"))+
  labs(
    #title = "Estimated Marginal Means Over Time",
    x = "Date",
    y = "Estimated Area (cm²)"
  ) +
  theme_classic() +
  theme(
    legend.position = "none"
  )
```

```
p1 / p3
```


Güiri
Playa Blanca
Predicted area (cm²)
1250
1000
750
500
250
Estimated Area (cm²)
1000
500
0
2021
2022
2023
2024
Date

```
p2 / p4
```



Mortality

Model

Basic poisson model

```
# remove zeros, zeros produce errors with the offset in the model
df_1_2 <- df_1 |>
  filter(num.frag>0) |>
  droplevels()

modelo_p <- glmmTMB(
  mort. ~ date + site + offset(log(num.frag)) + (1 | structure_fixed),
  family = poisson,
  data = df_1_2
)
```

Check overdisperssion

```
performance::check_overdispersion(modelo_p)
```

```
## # Overdispersion test
##
##        dispersion ratio =     6.708
##   Pearson's Chi-Squared = 23471.280
##                 p-value =   < 0.001
```

Model is overdispersed so we need to use a Negative Binomial model instead.

```
modelo_nb <- glmmTMB(
  mort. ~ date + site + offset(log(num.frag)) + (1 | structure_fixed),
  family = nbinom2,
  data = df_1_2
)
```

Check overdisperssion again

```
performance::check_overdispersion(modelo_nb)
```

```
## # Overdispersion test
##
##  dispersion ratio = 0.916
##           p-value = 0.856
```

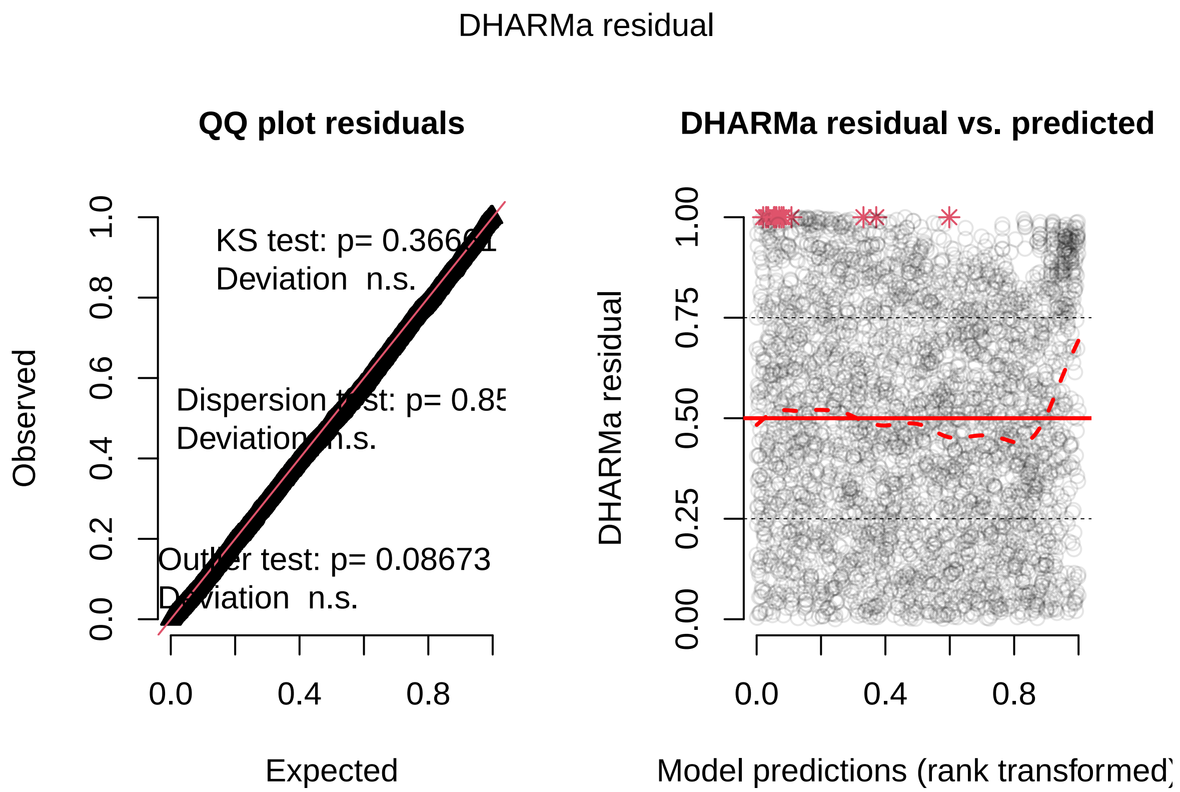
Model is not overdispersed, so we can use it.

Check model

```
summary(modelo_nb)
```

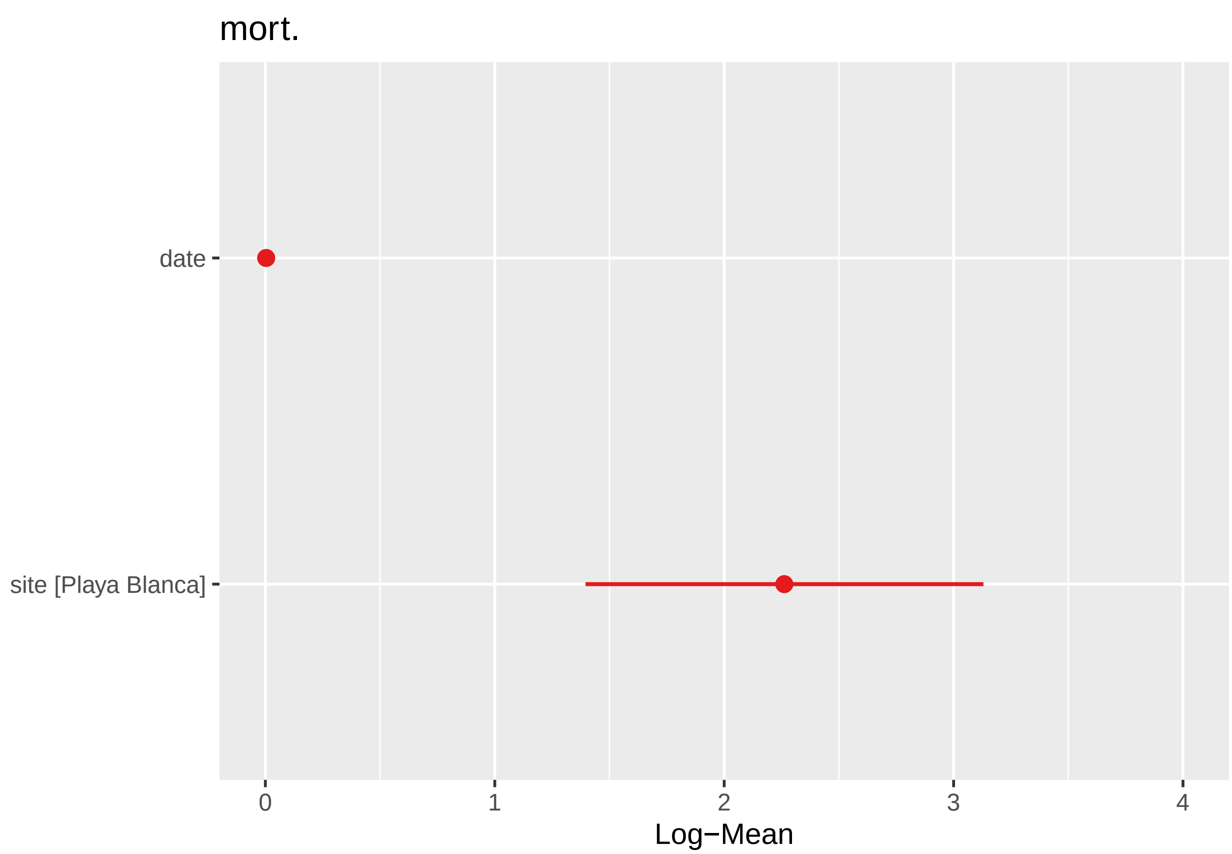
```
##  Family: nbinom2  ( log )
## Formula:
## mort. ~ date + site + offset(log(num.frag)) + (1 | structure_fixed)
## Data: df_1_2
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    2605.7    2636.5   -1297.8    2595.7      3498
##
## Random effects:
##
## Conditional model:
##  Groups          Name        Variance Std.Dev.
##  structure_fixed (Intercept) 0.3966   0.6298
## Number of obs: 3503, groups:  structure_fixed, 14
##
## Dispersion parameter for nbinom2 family (): 0.429
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -7.641e+01  4.211e+00  -18.15  < 2e-16 ***
## date               3.647e-03  2.146e-04   16.99  < 2e-16 ***
## sitePlaya Blanca   2.262e+00  4.418e-01    5.12 3.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
DHARMa::simulateResiduals(fittedModel = modelo_nb, plot = TRUE)
```




```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DI
##
## Scaled residual values: 0.5324148 0.7948437 0.1572925 0.05638294 0.5718798 0.04789467 0.1224141 0.16
```

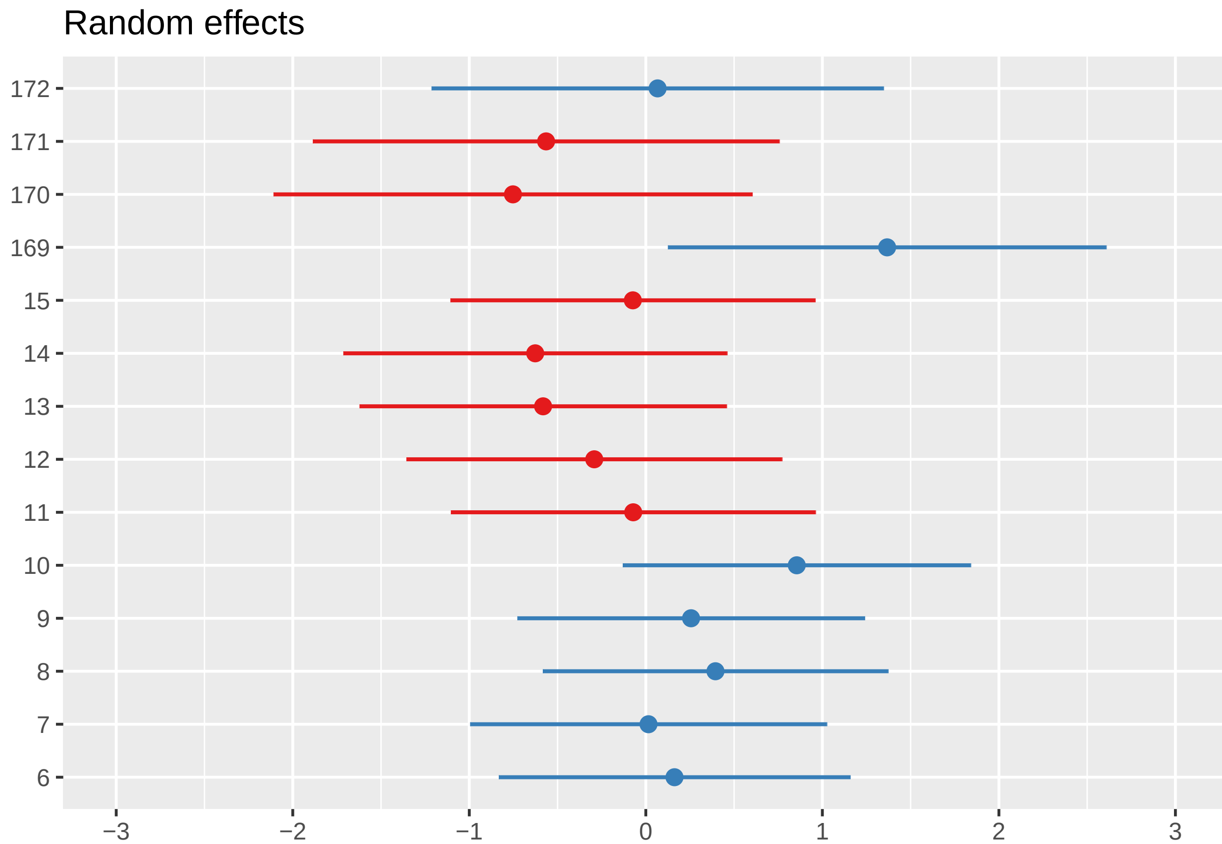
```r
plot_model(modelo_nb, type = "est", transform = NULL)
```




```
# random effecrs, structure
ranef(modelo_nb)
```

```
## $structure_fixed
##     (Intercept)
## 6    0.16239241
## 7    0.01526799
## 8    0.39444187
## 9    0.25629553
## 10   0.85488210
## 11  -0.07136995
## 12  -0.29210740
## 13  -0.58192579
## 14  -0.62656720
## 15  -0.07360507
## 169  1.36696962
## 170 -0.75270608
## 171 -0.56488243
## 172  0.06665128
```

```r
plot_model(modelo_nb, type = "re", transform = NULL)
```

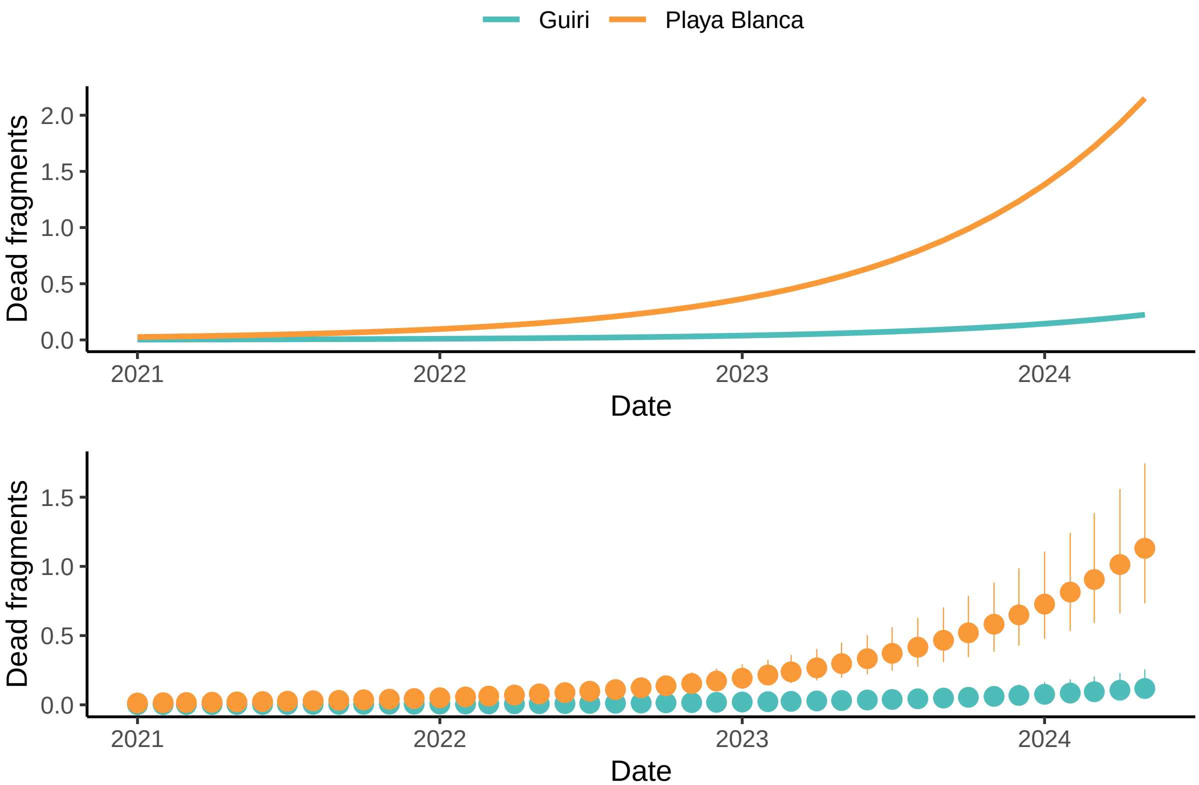


Predictions

```r
# If that fails, generate predictions manually:
new_data <- expand.grid(site=levels(df_1_2$site),
                        date = seq(as.Date("2021-01-01"),
                                        as.Date("2024-05-01"),
                                        by = "month"),
                        num.frag = seq(1, max(df_1_2$num.frag)),
                         structure_fixed=levels(df_1_2$structure_fixed))


new_data$preds <- predict(modelo_nb, newdata = new_data, type = "response")

# Plot with ggplot2
p5 <- ggplot(new_data, aes(x = date, y = preds, col=site)) +
  stat_summary(geom = "line", fun = "mean", linewidth = 1) +
  labs(y = "Dead fragments", x = "Date")+
  scale_color_manual(values = paleta[c(2,3)])+
  theme_classic()+
  theme(legend.position = "top",
        legend.title = element_blank())
```

```
emm_res3 <- emmeans(modelo_nb,
                 pairwise ~ site | poly(date, 2),
                 at = list(date = seq(as.Date("2021-01-01"),
                                     as.Date("2024-05-01"),
                                     by = "month")),
                 adjust = "tukey",
                 type = "response")

emm_df3 <- as.data.frame(emm_res3$emmeans)
emm_df3$date <- as.Date(emm_df3$date, origin = "1970-01-01")


p6 <- ggplot(emm_df3, aes(x = date, y = response, color = site, fill = site)) +
  geom_errorbar(aes(ymin = asymp.LCL, ymax = asymp.UCL), width = 0.2, size = 0.2) +
  geom_point(size = 3) +
  scale_color_manual(values = paleta[c(2,3)],
                     labels = c("Güiri", "Playa Blanca"))+
  labs(
    #title = "Estimated Marginal Means Over Time",
    x = "Date",
    y = "Dead fragments"
  ) +
  theme_classic() +
  theme(
    legend.position = "none"
  )
```

```
p5 / p6
```


Guiri
Playa Blanca
Dead fragments
2.0
1.5
1.0
0.5
0.0
2021
2022
2023
2024
Date
Dead fragments
1.5
1.0
0.5
0.0
2021
2022
2023
2024
Date