Supplemental Material

Island colonization and environmental sustainability in the postglacial Mediterranean

This document contains all the code necessary for data processing and analysis presented in the paper "Island colonization and environmental sustainability in the postglacial Mediterranean." Additional data files required for analysis and can be found in the online data repository: https://osf.io/98gxh/. These files are also linked directly in the code below.

Creating dataset of islands

We begin by generating our dataset of Mediterranean islands. This dataset is derived from GADM's global vector data file, which we subsequently query to subset only those polygons that share no boundary with any other geometry (i.e. islands). In creating this dataset, we also calculate our first predictor variable (Area) from the vector geometry of each island.

```
#load packages
library(tidyverse)
library(sf)
library(rgeos)
library(raster)

#OSF data files
MedCountriesOSF <- "https://osf.io/6uzr8/download"
MedSeaOSF <- "https://osf.io/raswx/download"
NPP <- "https://osf.io/xgfnu/download"
dates <- "https://osf.io/j5g7s/download"
biodiversity <- "https://osf.io/ucdxq/download"

#load list of countries bordering Mediterranean
MedCountries <- read_csv(MedCountriesOSF)

#add column with ISO3 code to subset GADM data
ISO <- as.data.frame(getData("ISO3"))
MedCountries <- merge(MedCountries, ISO, by.x = "Countries", by.y = "NAME")

#download data
gadm_subset <- do.call("bind", lapply(MedCountries$ISO3,
                                      function(x) getData('GADM', country=x, level=0)))

#convert to sf object
gadm_subset <- st_as_sf(gadm_subset)

#project Mediterranean subset to UTM N33
gadm_subset <- st_transform(gadm_subset, 32633)
gadm_subset$area <- st_area(gadm_subset)
```

```r
#dissolve all features to remove internal divisions (i.e.
#Cyprus and Northern Cyprus) and then convert multi-part
#polygon to single part. This effectively creates a
#separate feature for each isolated geometry (i.e. island)
islands <- gadm_subset %>%
  st_buffer(0) %>%
  summarise(area = sum(area)) %>%
  st_cast("POLYGON")

#read and project shapefile for Mediterranean
MedSea <- MedSeaOSF %>%
  download.file("MedSea.zip")
unzip("MedSea.zip")
MedSea <- st_read("MedSea.shp") %>%
  st_transform(32633)

#create table identifying which island geometries
#intersect with the Mediterranean. This will remove
#those geometries outside the Mediterranean (such
#as in the Red Sea, Black Sea, or on the Atlantic coast)
Med_subset <- st_intersects(islands, MedSea)
islands <- islands[which(lengths(Med_subset)==1),]

#calculate area for each geometry, in km2
islands$area <- islands %>%
  st_area() %>%
  units::set_units(value = km^2) %>%
  as.numeric()

#we save the pre-subset file for isolation index calculation later
islands_all <- islands

#subset islands by those geometries greater than or equal in area
#to 10 km2
islands <- islands[islands$area >= 10,]

#remove Europe and Asia-Africa
islands <- islands[!islands$area == sort(islands$area, decreasing = T)[1],]
islands <- islands[!islands$area == sort(islands$area, decreasing = T)[1],]
```

The final result is a dataset 161 islands, with an average area of 643.8 square kilometers.

Calculate Isolation Index

Isolation index is calculated here as the minimum distance between each island and the nearest landmass of equal or greater size.

```r
library(nngeo)

#calculate isolation index value for each island
islands$ii <- sapply(1:nrow(islands), function(i){
  min(
```

```
    st_distance(
      islands[i,],
      islands_all[islands_all$area > islands$area[i],])
    )
  }
  )

#convert to kilometers
islands$ii <- islands$ii / 1000
```

NPP

Net Primary Productivity data were collected in 2019 by the Moderate Resolution Imaging Spectroradiometer (MODIS) on NASA's Terra satellite and subsequently made available at 500m resolution in the MOD17A3HGF Version 6 product. This product was downloaded through Google Earth Engine.

```
#read in NPP raster, reproject it, and crop it to extent of islands
NPP %>%
  download.file("NPP.tif")

NPP <- raster("NPP.tif") %>%
  projectRaster(crs = crs(islands)) %>%
  crop(extent(islands))

#rescale data
NPP <- NPP * 0.0001

#create new column for values
islands$npp = 0

#extract and average NPP values for each island
islands$npp <- raster::extract(NPP, islands, fun = mean, na.rm = T)[,1]
```

Richness

```
#read in mammals, birds, and amphibians raster layers
biodiversity %>%
  download.file("biodiversity.zip")
unzip("biodiversity.zip")

mammals <- raster("mammals.tif") %>%
  projectRaster(crs = crs(islands)) %>%
  crop(extent(islands))

birds <- raster("birds.tif") %>%
  projectRaster(crs = crs(islands)) %>%
  crop(extent(islands))

amphibians <- raster("amphibians.tif") %>%
  projectRaster(crs = crs(islands)) %>%
```

```
  crop(extent(islands))

#create new columns for each
islands$mammals_rich <- 0
islands$birds_rich <- 0
islands$amphib_rich <- 0

#extract and average species count for each island for each layer
for (i in 1:nrow(islands)){
  islands$mammals_rich[i] <- raster::extract(mammals,islands[i,],fun = mean, na.rm = T)
  islands$birds_rich[i] <- raster::extract(birds, islands[i,], fun = mean, na.rm = T)
  islands$amphib_rich[i] <- raster::extract(amphibians, islands[i,], fun = mean, na.rm = T)
}

#create new richness column
islands$richness = 0

#sum species counts for each island
for (i in 1:nrow(islands)){
  islands$richness[i] <- sum(c(islands$mammals_rich[i],
                               islands$birds_rich[i],
                               islands$amphib_rich[i]),
                             na.rm = T)
}
```

Longitude

```
#temporarily project data to geographic coordinates and take longitude
#coordinate of island centroid
islands$long <- islands %>%
  st_transform(4326) %>%
  st_centroid() %>%
  st_coordinates() %>%
  .[,1]
```

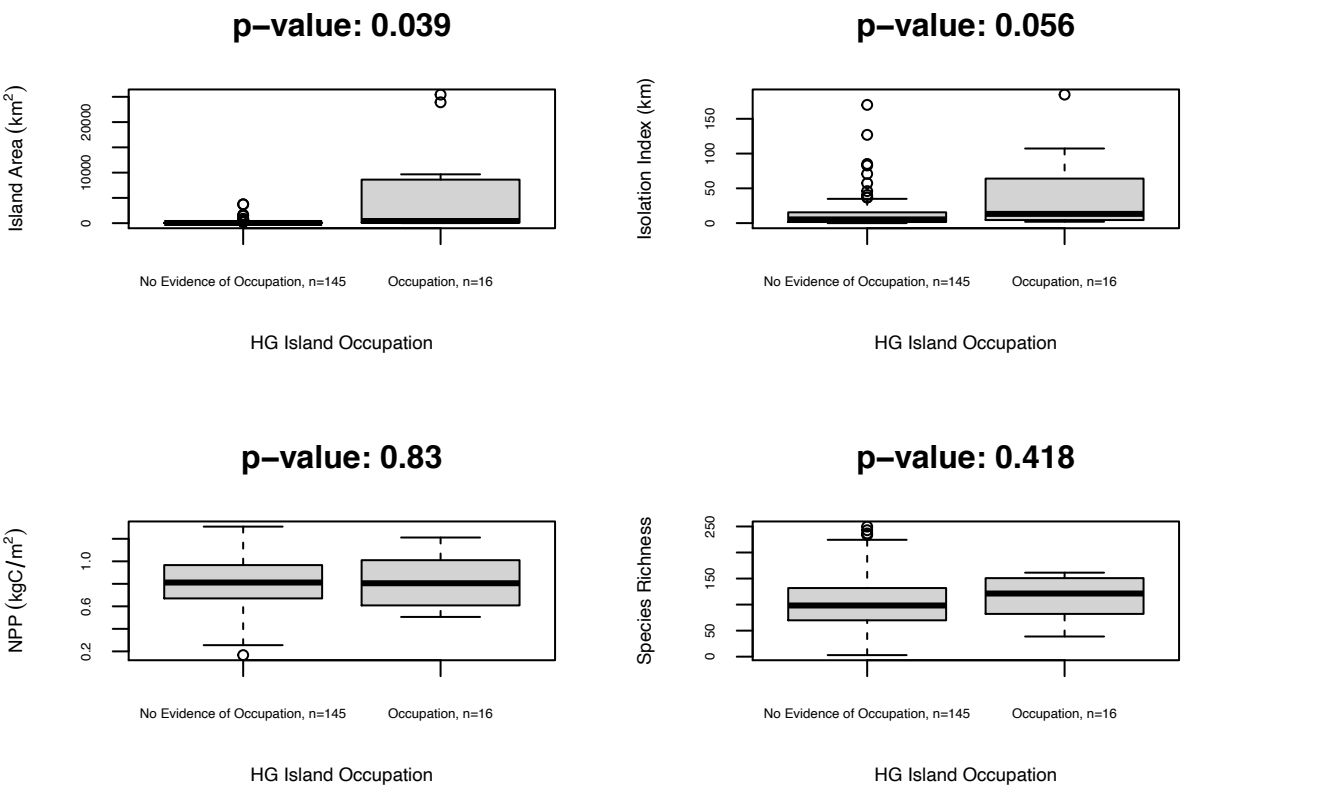
Having calculated our environmental attributes, we now join our island data to a table of occupation dates for both HG and AGR populations.

```
#drop geometry from islands to make processing quicker
islands <- st_drop_geometry(islands)

#import and join dating attributes
dates <- read_csv(dates)

#join dates to islands
islands <- left_join(islands, dates, by = "key")
```

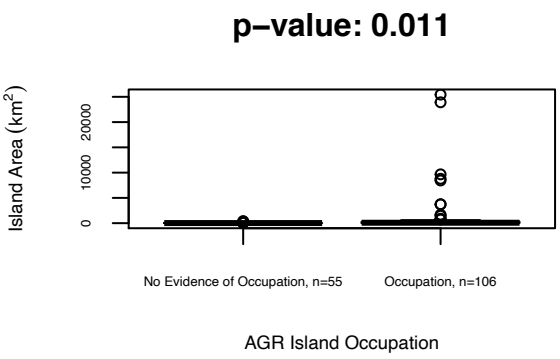
Analysis

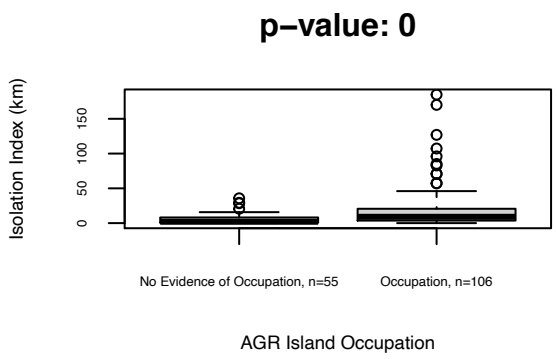
First we plot the distribution of predictor variables of islands with occupation against those without, for both HG and AGR strategies. Boxplot title labels record the p-value of t-test comparison between distributions.

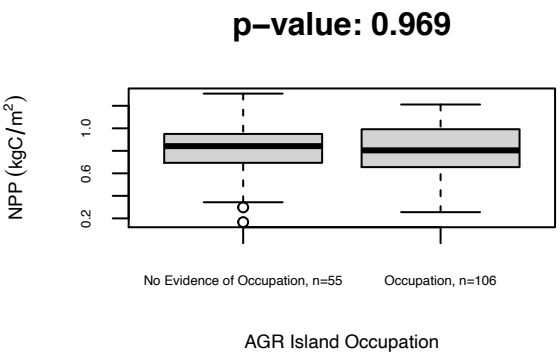
```r
#boxplot distributions of HG start date relative to predictor variables
islands$HG_occ <- as.factor(islands$HG_occ)

par(mfrow=c(2,2), cex.axis=.5, cex.lab=.75)
boxplot(islands$area ~ islands$HG_occ,
        main = paste0("p-value: ", round(t.test(islands$area ~ islands$HG_occ)$p.value, 3)),
        xlab = "HG Island Occupation",
        ylab = expression(Island~Area~(km^2)),
        names=paste(levels(islands$HG_occ), ", n=", table(islands$HG_occ), sep=""))
boxplot(islands$ii ~ islands$HG_occ,
        main = paste0("p-value: ", round(t.test(islands$ii ~ islands$HG_occ)$p.value, 3)),
        xlab = "HG Island Occupation",
        ylab = "Isolation Index (km)",
        names=paste(levels(islands$HG_occ), ", n=", table(islands$HG_occ), sep=""))
boxplot(islands$npp ~ islands$HG_occ,
        main = paste0("p-value: ", round(t.test(islands$npp ~ islands$HG_occ)$p.value, 3)),
        xlab = "HG Island Occupation",
        ylab = expression(NPP~(kg*C/m^2)),
        names=paste(levels(islands$HG_occ), ", n=", table(islands$HG_occ), sep=""))
boxplot(islands$richness ~ islands$HG_occ,
        main = paste0("p-value: ", round(t.test(islands$richness ~ islands$HG_occ)$p.value, 3)),
        xlab = "HG Island Occupation",
        ylab = "Species Richness",
        names=paste(levels(islands$HG_occ), ", n=", table(islands$HG_occ), sep=""))
```

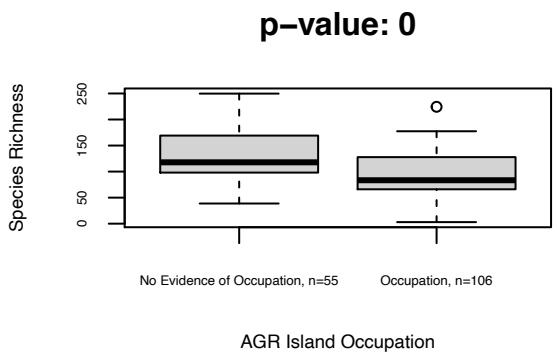


```r
#boxplot distributions of AGR start date relative to predictor variables
islands$AGR_occ <- as.factor(islands$AGR_occ)

par(mfrow=c(2,2), cex.axis=.5, cex.lab=.75)
boxplot(islands$area ~ islands$AGR_occ,
        main = paste0("p-value: ", round(t.test(islands$area ~ islands$AGR_occ)$p.value, 3)),
        xlab = "AGR Island Occupation",
        ylab = expression(Island~Area~(km^2)),
        names=paste(levels(islands$AGR_occ), ", n=", table(islands$AGR_occ), sep=""))
boxplot(islands$ii ~ islands$AGR_occ,
        main = paste0("p-value: ", round(t.test(islands$ii ~ islands$AGR_occ)$p.value, 3)),
        xlab = "AGR Island Occupation",
        ylab = "Isolation Index (km)",
        names=paste(levels(islands$AGR_occ), ", n=", table(islands$AGR_occ), sep=""))
boxplot(islands$npp ~ islands$AGR_occ,
        main = paste0("p-value: ", round(t.test(islands$npp ~ islands$AGR_occ)$p.value, 3)),
        xlab = "AGR Island Occupation",
        ylab = expression(NPP~(kg*C/m^2)),
        names=paste(levels(islands$AGR_occ), ", n=", table(islands$AGR_occ), sep=""))
boxplot(islands$richness ~ islands$AGR_occ,
        main = paste0("p-value: ", round(t.test(islands$richness ~ islands$AGR_occ)$p.value, 3)),
        xlab = "AGR Island Occupation",
        ylab = "Species Richness",
        names=paste(levels(islands$AGR_occ), ", n=", table(islands$AGR_occ), sep=""))
```









The distribution of these environmental variables across all islands can also be viewed in tabular form:

```
variable_summary <- islands %>%
  gather(Variable, Value, area, ii, npp, richness) %>%
  group_by(Variable) %>%
  summarise(min = min(Value, na.rm = T),
            mean = mean(Value, na.rm = T),
            sd = sd(Value, na.rm = T),
            max = max(Value, na.rm = T))

knitr::kable(variable_summary, format="markdown", digits = 2)
```

Variable	min	mean	sd	max
area	10.21	643.80	2992.80	25439.61
ii	0.00	15.31	27.38	184.78
npp	0.17	0.82	0.23	1.31
richness	3.00	107.86	47.34	249.72

Next we turn to model selection. We define our set of variable combinations by which we can build multiple linear regression model sets.

```
library(AICcmodavg)
library(car)

#define variables to be used
indep_param <- c("area", "ii", "npp", "richness")
#create a list of possible combinations
combos <- do.call(c, lapply(seq_along(indep_param), combn, x = indep_param, simplify = F))

#build multiple linear regression model set of each combination
hg <- vector(mode="list", length=length(combos))
for (i in 1:length(combos)){
  hg[[i]] <- eval(
    parse(
      text = paste0("lm(HG_start ~ ", paste(combos[[i]], collapse = "+"),", data = islands)")))
}

#check for collinearity of regression variables
vif(hg[[length(combos)]])
```

```
##     area       ii      npp richness
## 2.216377 1.309731 1.261147 2.101190
```

```
#calculate AIC values and weights for models
hg_aic <- aictab(hg)
knitr::kable(hg_aic, format="markdown", digits = 3)
```

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL	Cum.Wt
6	Mod6	4	279.307	0.000	1.000	0.453	-133.835	0.453
1	Mod1	3	280.978	1.671	0.434	0.197	-136.489	0.650

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL	Cum.Wt
7	Mod7	4	281.479	2.172	0.338	0.153	-134.921	0.803
13	Mod13	5	282.725	3.418	0.181	0.082	-133.362	0.885
11	Mod11	5	283.620	4.313	0.116	0.052	-133.810	0.937
5	Mod5	4	284.605	5.298	0.071	0.032	-136.484	0.969
12	Mod12	5	285.678	6.371	0.041	0.019	-134.839	0.988
15	Mod15	6	287.906	8.599	0.014	0.006	-133.286	0.994
3	Mod3	3	290.666	11.359	0.003	0.002	-141.333	0.996
2	Mod2	3	290.985	11.678	0.003	0.001	-141.493	0.997
10	Mod10	4	291.501	12.194	0.002	0.001	-139.932	0.998
4	Mod4	3	291.932	12.625	0.002	0.001	-141.966	0.999
8	Mod8	4	292.429	13.122	0.001	0.001	-140.396	1.000
9	Mod9	4	294.133	14.826	0.001	0.000	-141.249	1.000
14	Mod14	5	294.565	15.258	0.000	0.000	-139.282	1.000

The results of this analysis indicate that the best performing model for predicting hunter-gatherer occupation dates was the one using area, npp as parameters. The AICcWT of this model, or rather the likelihood that it is the best fitting model of the set, was 0.45. The fit of this model, as reflected its R-squared value, is 0.6. As the VIF values are all below or close to 2, variables are only moderately correlated and admissible.

The coefficients for the best performing model can be seen below:

```
best_model <- as.numeric(str_remove(hg_aic$Modnames[1], pattern= "Mod"))
knitr::kable(coefficients(hg[[best_model]]), format="markdown", digits = 3)
```

	x
(Intercept)	12723.895
area	0.158
npp	-2915.605

Now we proceed to investigate these relationships between our environmental variables and AGR occupation dates.

```
#build multiple linear regression model set of each combination
agr <- vector(mode="list", length=length(combos))
for (i in 1:length(combos)){
  agr[[i]] <- eval(
    parse(
      text = paste0("lm(AGR_start ~ ", paste(combos[[i]], collapse = "+"),", data = islands)")))
}

#check again for collinearity of regression variables
vif(agr[[length(combos)]])
```

```
##     area       ii      npp richness
## 1.425977 1.276855 1.097030 1.296388
```

```
#calculate AIC values and weights for models
agr_aic <- aictab(agr)
knitr::kable(agr_aic, format="markdown", digits = 3)
```

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL	Cum.Wt
6	Mod6	4	1858.111	0.000	1.000	0.354	-924.858	0.354
1	Mod1	3	1859.406	1.295	0.523	0.185	-926.585	0.540
13	Mod13	5	1860.237	2.126	0.345	0.122	-924.819	0.662
11	Mod11	5	1860.313	2.202	0.333	0.118	-924.857	0.780
7	Mod7	4	1861.500	3.389	0.184	0.065	-926.552	0.845
5	Mod5	4	1861.567	3.455	0.178	0.063	-926.585	0.908
15	Mod15	6	1862.472	4.361	0.113	0.040	-924.812	0.948
12	Mod12	5	1863.701	5.590	0.061	0.022	-926.551	0.970
3	Mod3	3	1865.551	7.440	0.024	0.009	-929.658	0.978
8	Mod8	4	1866.279	8.168	0.017	0.006	-928.942	0.984
10	Mod10	4	1867.144	9.033	0.011	0.004	-929.374	0.988
2	Mod2	3	1867.567	9.455	0.009	0.003	-930.666	0.991
4	Mod4	3	1867.570	9.458	0.009	0.003	-930.667	0.995
14	Mod14	5	1867.752	9.641	0.008	0.003	-928.576	0.997
9	Mod9	4	1867.961	9.849	0.007	0.003	-929.782	1.000

The results of this analysis indicate that the best performing model for predicting AGR occupation dates was the one again using area, npp as parameters. The AICcWT of this model, or rather the likelihood that it is the best fitting model of the set, was 0.35. The fit of this model, as reflected its R-squared value, is 0.1. As the VIF values are all below 2, variables are only moderately correlated and admissible.

The coefficients for the best performing model can be seen below:

```
best_model <- as.numeric(str_remove(agr_aic$Modnames[1], pattern= "Mod"))
knitr::kable(coefficients(agr[[best_model]]), format="markdown", digits = 3)
```

	x
(Intercept)	4936.328
area	0.126
npp	1219.934

Finally, we add longitude as an additional parameter in our analysis and explore whether the addition of this variable creates a better fitting model for AGR occupation dates.

```
#define variables to be used
indep_param <- c("area", "ii", "npp", "richness", "long")
#create list of possible combinations
combos <- do.call(c, lapply(seq_along(indep_param), combn, x = indep_param, simplify = F))

agr2 <- vector(mode="list", length=length(combos))
for (i in 1:length(combos)){
  agr2[[i]] <- eval(
    parse(
      text = paste0("lm(AGR_start ~ ", paste(combos[[i]], collapse = "+"),", data = islands)")))
}

#check again for collinearity of regression variables
vif(agr2[[length(combos)]])
```

```
##     area       ii      npp richness     long
```

```
## 1.431109 1.327220 1.239518 1.297550 1.211711
```

```
#calculate AIC values and weights for models
agr2_aic <- aictab(agr2)
knitr::kable(agr2_aic, format="markdown", digits = 3)
```

	Modnames	K	AICc	Delta_AICc	ModelLik	AICcWt	LL	Cum.Wt
20	Mod20	5	1855.661	0.000	1.000	0.341	-922.531	0.341
29	Mod29	6	1857.698	2.037	0.361	0.123	-922.425	0.464
27	Mod27	6	1857.736	2.075	0.354	0.121	-922.444	0.585
7	Mod7	4	1858.111	2.450	0.294	0.100	-924.858	0.685
1	Mod1	3	1859.406	3.745	0.154	0.052	-926.585	0.737
9	Mod9	4	1859.749	4.088	0.130	0.044	-925.676	0.781
31	Mod31	7	1859.891	4.230	0.121	0.041	-922.374	0.822
19	Mod19	5	1860.237	4.576	0.101	0.035	-924.819	0.857
16	Mod16	5	1860.313	4.652	0.098	0.033	-924.857	0.890
8	Mod8	4	1861.500	5.839	0.054	0.018	-926.552	0.909
6	Mod6	4	1861.567	5.906	0.052	0.018	-926.585	0.926
21	Mod21	5	1861.866	6.205	0.045	0.015	-925.633	0.942
18	Mod18	5	1861.880	6.219	0.045	0.015	-925.640	0.957
26	Mod26	6	1862.472	6.811	0.033	0.011	-924.812	0.968
17	Mod17	5	1863.701	8.040	0.018	0.006	-926.551	0.974
28	Mod28	6	1863.991	8.330	0.016	0.005	-925.571	0.980
23	Mod23	5	1864.650	8.989	0.011	0.004	-927.025	0.983
14	Mod14	4	1865.211	9.550	0.008	0.003	-928.408	0.986
3	Mod3	3	1865.551	9.890	0.007	0.002	-929.658	0.989
30	Mod30	6	1866.162	10.501	0.005	0.002	-926.657	0.991
10	Mod10	4	1866.279	10.618	0.005	0.002	-928.942	0.992
25	Mod25	5	1866.894	11.233	0.004	0.001	-928.147	0.993
13	Mod13	4	1867.144	11.483	0.003	0.001	-929.374	0.995
2	Mod2	3	1867.567	11.906	0.003	0.001	-930.666	0.995
4	Mod4	3	1867.570	11.908	0.003	0.001	-930.667	0.996
22	Mod22	5	1867.752	12.091	0.002	0.001	-928.576	0.997
11	Mod11	4	1867.961	12.299	0.002	0.001	-929.782	0.998
12	Mod12	4	1868.387	12.726	0.002	0.001	-929.995	0.998
5	Mod5	3	1868.432	12.771	0.002	0.001	-931.099	0.999
24	Mod24	5	1868.484	12.823	0.002	0.001	-928.942	1.000
15	Mod15	4	1868.899	13.238	0.001	0.000	-930.252	1.000

The results of this analysis indicate that the best performing model for predicting AGR occupation dates was the one again using area, npp, long as parameters. The AICcWT of this model, or rather the likelihood that it is the best fitting model of the set, was 0.34. The fit of this model, as reflected its R-squared value, is 0.13. As the VIF values are all below 2, variables are only moderately correlated and admissible.

The coefficients for the best performing model can be seen below:

```
best_model <- as.numeric(str_remove(agr2_aic$Modnames[1], pattern= "Mod"))
knitr::kable(coefficients(agr2[[best_model]]), format="markdown", digits = 3)
```

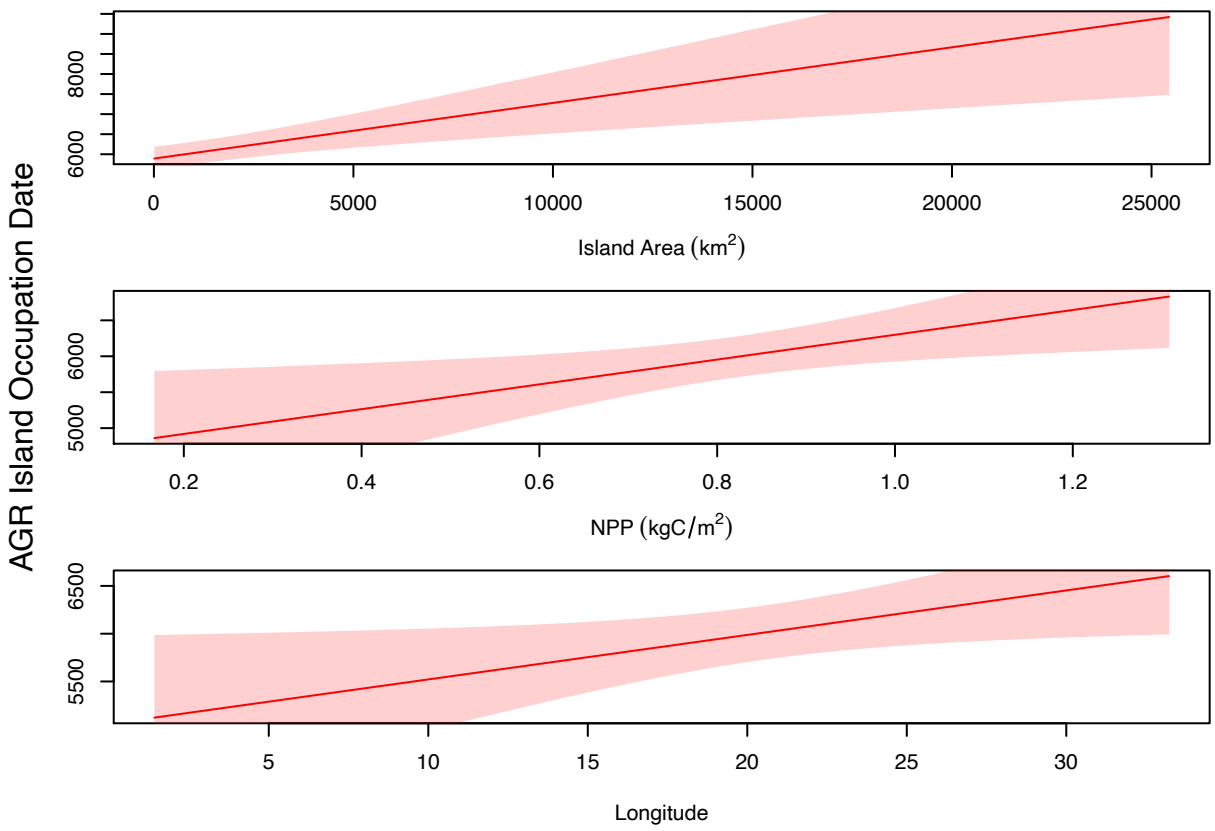
	x
(Intercept)	3559.418
area	0.139
npp	1725.163
long	46.552

Further exploring the best-fitting AGR model

We can visualize what effect our three predictor variables have on the modeled AGR occupation dates by plotting each individually, while holding the rest constant.

```
#define object with best-fitting model
agr_best <- agr2[[as.numeric(str_remove(agr2_aic$Modnames[1], pattern= "Mod"))]]

#we can use the best-fitting model to predict island occupation dates
#given a new dataset made up of sequences of the variables in question
#and the other variables, held constant at their means

#area
area_pred <- predict(agr_best,
                     newdata = data.frame("area"=seq(min(islands$area),
                                                     max(islands$area),
                                                     length.out = 100),
                                          "npp"=rep(mean(islands$npp, na.rm = T), 100),
                                          "long"=rep(mean(islands$long), 100)),
                     se = T, type = "response")
#npp
npp_pred <- predict(agr_best,
                     newdata=data.frame("npp"=seq(min(islands$npp, na.rm = T),
                                                     max(islands$npp, na.rm = T),
                                                     length.out = 100),
                                          "area"=rep(mean(islands$area, na.rm = T), 100),
                                          "long"=rep(mean(islands$long), 100)),
                     se=T, type = "response")

#longitude
long_pred <- predict(agr_best,
                     newdata=data.frame("long"=seq(min(islands$long),
                                                     max(islands$long),
                                                     length.out = 100),
                                          "npp"=rep(mean(islands$npp, na.rm = T), 100),
                                          "area"=rep(mean(islands$area), 100)),
                     se=T, type = "response")


#generate plots
par(mfrow=c(3, 1), oma=c(1,0,0,0), mar=c(4, 5, 1, 1))

#area
plot(area_pred$fit ~ seq(min(islands$area),
                         max(islands$area),
                         length.out = 100),
```

```
     type="l", col="red", xlab=expression(Island~Area~(km^2)), ylab="")
polygon(c(
  rev(seq(min(islands$area), max(islands$area), length.out = 100)),
  seq(min(islands$area), max(islands$area), length.out = 100)),
  c(rev(area_pred$fit + (area_pred$se.fit*1.96)),
    area_pred$fit - (area_pred$se.fit*1.96)),
  col = "#ff000030", border = NA)

#npp
plot(npp_pred$fit ~ seq(min(islands$npp, na.rm = T),
                        max(islands$npp, na.rm = T),
                        length.out = 100),
     type="l", col="red", xlab=expression(NPP~(kg*C/m^2)), ylab="")
polygon(c(
  rev(seq(min(islands$npp, na.rm = T), max(islands$npp, na.rm = T), length.out = 100)),
  seq(min(islands$npp, na.rm = T), max(islands$npp, na.rm = T), length.out = 100)),
  c(rev(npp_pred$fit + (npp_pred$se.fit*1.96)),
    npp_pred$fit - (npp_pred$se.fit*1.96)),
  col = "#ff000030", border = NA)

mtext("AGR Island Occupation Date", side=2, line=3)

#long
plot(long_pred$fit ~ seq(min(islands$long),
                         max(islands$long),
                         length.out = 100),
     type="l", col="red", xlab="Longitude", ylab="")
polygon(c(
  rev(seq(min(islands$long), max(islands$long), length.out = 100)),
  seq(min(islands$long), max(islands$long), length.out = 100)),
  c(rev(long_pred$fit + (long_pred$se.fit*1.96)),
    long_pred$fit - (long_pred$se.fit*1.96)),
  col = "#ff000030", border = NA)
```



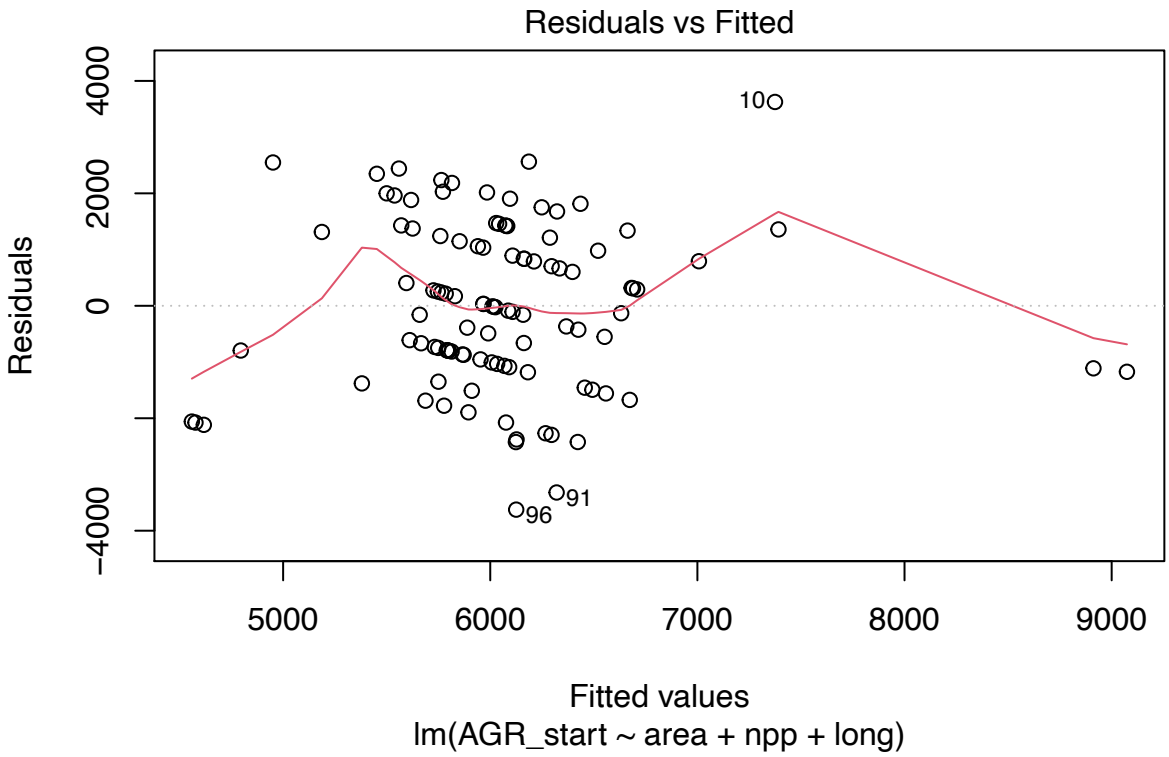

```
dev.off()
```

```
## null device
##           1
```

These models collectively illustrate that all three predictor variables have a positive relationship with island occupation. The confidence intervals here show where the model fits the data better.

Looking closer at the overall model residuals, we see that there is no evident patterning in the residuals, indicating that we are not missing some non-linear relationship between our predictor variables and the dependent variable.

```
plot(agr_best, which = 1)
```



Finally, we can look at the residuals themselves and see how they are spatially distributed. We do so in Figure 4 of the article, created in QGIS using these data.