Table S1. Variables with significance (+) and non-significance (−) regarding EGV set 5 for *P. undulatum* and *M. faya* in Pico Island. PU: *P. undulatum*; MF: *M. faya*; P: Pico; D: Deleting variables.

Pico Island	
P. undulatum	
Model	**EGV (GLM)**
EGV5_PU_P Full	+
Reduction 5000	
EGV5_PU_P	+
Reduction 500	
EGV5_PU_P	(-) TRA; RHRA; DL 4
EGV5_PU_P_D	+
Reduction 50	
EGV5_PU_P	(-) DEM; PMRA; DL 4
EGV5_PU_P_D	(-) SLP; PM
EGV5_PU_P_D_D	(-) TRA
EGV5_PU_P_D_D_D	(-) RHRA
EGV5_PU_P_D_D_D_D	+
M. faya	
Model	
EGV5_MF_P Full	+
Reduction 3000	
EGV5_MF_P	+
Reduction 300	
EGV5_MF_P	+
Reduction 30	
EGV5_MF_P	(-) DEM; SLP; TM; TRA; DL 1
EGV5_MF_P_D	+

Table S2. Variables with significance (+) and non-significance (−) regarding EGV set 5 for *P. undulatum* and *M. faya* in São Miguel Island. PU: *P. undulatum*; MF: *M. faya*; SM: São Miguel; D: Deleting variables.

São Miguel Island	
P. undulatum	
Model	**EGV (GLM)**
EGV5_PU_SM Full	+
Reduction 3000	
EGV5_ PU_SM	+
Reduction 300	
EGV5_ PU_SM	(-) RHM
EGV5_ PU_SM _D	+
Reduction 30	
EGV5_ PU_SM	(-) FLA; WHS; TM; TMRA; RHM; DL 2
EGV5_ PU_SM _D	+
M. faya	
Model	
EGV5_ MF_SM Full	+
Reduction 300	
EGV5_ MF_SM	+
Reduction 30	
EGV5_ MF_SM	(-) RHRA; DL1
EGV5_ MF_SM _D	+

R script

```
########## Species distribution modelling: comparison of fixed and mixed effects models using INLA ##########

########## An example: Pittosporum undulatum_Pico Island ##########

### We following the guidelines of The R-INLA tutorial on SPDE models
### Elias T. Krainski, Finn Lindgren, Daniel Simpson and Havard Rue


### Data preparation
#### Species occurrence data
### Importing occurrence data
### load extra code
library (mgcv)
library (ROCR)
library (dismo)
library (csvread)
library (raster)
library (INLA)


### Load and read species data (presence + absence _ total)in ascii
### Data to be used for absences

Pittosporum <- readAsciiGrid("Pittosporum_IFT_Pico_Tr.asc",as.image=T)
summary(Pittosporum)


### Load and read species data in csv

Pittosporum <- read.table("Pittosporum_IFT_Pico_Tr.csv",header=T ,sep=",")
head(Pittosporum)


### Load occurrence data
### Data to be used for presences

occurrencedata <- read.table("Pittosporum_IFT_Pico_Tr.csv", header=TRUE, sep=",")

# Remove the first column
occurrencedata <-occurrencedata[,2:3]
# Extracts the environmental variables for each presence
# Second option to remove the first column
# occurrencedata <- occurrencedata[,-1]

dim(occurrencedata)
head(occurrencedata)
names(occurrencedata)
```

```
### Load ecogeographical variables(EGVs)

grids <- list.files(pattern='asc', full.names=T)# All "asc" files in folder
variables <- stack(grids)# Make a "stack" of EGV #Install "stack"
variables
names(variables)<-
c("Altimetry_Pico_100","Aspect_Pico_100","Curvature_alt_Pico_100",
"Curvature_alt_Pico_100m","Dist100_class1_Pico",
"Dist100_class2_Pico","Dist100_class34_Pico",
"Dist100_class57_Pico","Dist100_class6_Pico",
"Dist100_class8_Pico","Flow_acum_Pico_100","Flow_acum_Pico_100m",
"Hillshade_summer_Pico_100",
"Hillshade_winter_Pico_100","Landuse_ord_Pico_100","Pittosporum",
"Rain_all_pca1_Pico_c1_100","Rain_all_pca2_Pico_c2_100",
"Rain_annual_Pico_100","Rain_max_Pico_100",
"Rain_min_Pico_100","Rain_pca_Pico_c1_100",
"Rain_range_annual_Pico_100","Rain_range_Pico_100",
"RH_max_annual_Pico_100","RH_med_annual_Pico_100",
"RH_min_annual_Pico_100","RH_pca_Pico_c1_100",
"RH_pca_Pico_c2_100","RH_range_annual_Pico_100",
"RH_range_med_Pico_100","Slope_Pico_100","T_max_annual_Pico_100",
"T_max_max_Pico_100","T_max_pca1_Pico_c1_100",
"T_med_annual_Pico_100","T_med_annual_Pico_100_2",
"T_min_annual_Pico_100","T_min_pca1_Pico_c1_100",
"T_min_pca1_Pico_c2_100","T_pca_Pico_c1_100",
"T_pca_Pico_c2_100","T_range_annual_Pico_100",
"T_range_med_Pico_100")


### Extracting values from rasters: Species data x EGVs
### Obtain the presences

presvals <- extract(variables, occurrencedata)
mean(presvals[,13])
dim(presvals)
head(presvals)
names(presvals)


### Read of presences and absences from ascii(grid)
### Write a matrix from ascii
### Matrix composed by elements: presence(1); abscence (0); NA; coordinates
### Three dimensional matrix - 3layers


Pittosporum <- readAsciiGrid("Pittosporum_IFT_Pico_Tr.asc",as.image=T)# ascii total of
presences and absences
dim(Pittosporum)
head(Pittosporum)
names(Pittosporum)
head(Pittosporum[z])
```

```
### Avoid deformation(map)
all<-matrix(nrow=129762,ncol=3,NA)
n=0
for(c in 1:534) {
for(l in 1:243) {
n=n+1
all[n,1]=Pittosporum$x[c]
all[n,2]=Pittosporum$y[l]
all[n,3]=Pittosporum$z[c,l]
}}
### Eliminate "NA"
all<- all[-which(all[,3] %in% NA),]
### Abscences matrix
absences<-all[all[,3]==0,]
head(absences)
dim (absences)
### Presences matrix
presences<-all[all[,3]==1,]
head(presences)
dim(presences)



### Raster data

presvals <- extract(variables, occurrencedata)
dim(presvals)

### Setting random seed to always create the same
### Random set of points

set.seed(0)

### Eliminate the third column (x,y)
absences <- absences[,-3]
samp <- sample(nrow(absences), 10000)# Should we set a number
psedo_absences <- absences[samp,]
absvals <- extract(variables, psedo_absences)
dim(absvals)
head(absvals)


### 1 corresponds to presences and 0 abscences

pb <- c(rep(1, nrow(presvals)), rep(0, nrow(absvals)))
length(pb)

### Create the presence absence matrix

sdmdata <- data.frame(cbind(pb, rbind(presvals, absvals)))
```

```
dim (sdmdata)



########## Fixed effects models- GLM ##########
### Model EGV set 5 = glm1
### Prediction

glm1 <- glm(pb ~ Altimetry_Pico_100 +
+ Slope_Pico_100 +
Dist100_class2_Pico
+ Dist100_class57_Pico
+ Rain_annual_Pico_100
+ Rain_range_annual_Pico_100
+ RH_range_annual_Pico_100 + T_med_annual_Pico_100
+ T_range_annual_Pico_100, data=sdmdata, family = "binomial")

summary(glm1) # values of AIC


### Evaluation
### Evaluate the model (AUC)
### It is necessary to divide the data in two random sets, one for training
### and one for evaluating the model (75%; 25%)
### Sdmata subsample - AUC

samp <- sample(nrow(sdmdata), round(0.1 * nrow(sdmdata))) # Should we set a number
sdmdata_10<-sdmdata[samp,]
head(sdmdata_10)
dim(sdmdata_10)

samp <- sample(nrow(sdmdata_10), round(0.75 * nrow(sdmdata_10)))
traindata <- sdmdata_10[samp,]
dim(traindata)
testdata <- sdmdata_10[-samp,]
dim(testdata)

e <- evaluate(testdata[testdata==1,], testdata[testdata==0,], glm1)
e
plot(e, 'ROC')


### k fold validation AUC

k <-10
group <- kfold(sdmdata, k)
group[1:10]
unique(group)

# Now we can fit and test the model five times
# In each run, the records corresponding to one of the five groups
# is only used to evaluate the model,
```

```
# while the other four groups are only used to fit the model
# The results are stored in a list called 'e'

e_K_AUC <- list()
for (i in 1:k) {
train <- sdmdata[group != i,]
test <- sdmdata[group == i,]
glm102_D_13 <- glm(pb ~ Altimetry_Pico_100 +
+ Slope_Pico_100 +
Dist100_class2_Pico
+ Dist100_class57_Pico
+ Rain_annual_Pico_100
+ Rain_range_annual_Pico_100
+ RH_range_annual_Pico_100 + T_med_annual_Pico_100
+ T_range_annual_Pico_100, data=sdmdata, family = "binomial")
e_K_AUC[[i]] <- evaluate(test[test==1,],test[test==0,],glm1)}

auc <- sapply( e_K_AUC, function(x){slot(x, 'auc')} )
mean(auc)
sd(auc)


### Boyce Index

p1 <- predict(variables, glm1)
plot(p1)
head(p1)
dim(p1)
summary(p1)

size<-500
X<-as.vector(p1)
length(X)
summary(X)
X<- X[-which(X %in% NA)]
X<-(X-min(X))/(max(X)-min(X))*100
Z<-
c(0.791611559,0.500665612,0.446202121,0.417446816,0.38967264,0.368580517,0.34915674
7,0.330277569,0.314034906,0.29502174,0.282617152,0.268423191,0.254211941,0.24148319
1,0.230500592,0.219163576,0.210808848,0.198049843,0.189608672,0.178474798,0.1698002
3,0.158692288,0.148375301,0.141883423,0.135707062,0.128104389,0.122122524,0.1154793
7,0.110824408,0.104846866,0.09882178,0.096021023,0.092399056,0.088647424,0.08601955
3,0.082190123,0.079419621,0.07480788,0.072512815,0.069599682,0.066241366,0.06453843
7,0.063168313,0.061171477,0.059671689,0.056533804,0.054640699,0.052596319,0.0506816
04,0.049104017,0.048706379,0.047323289,0.045624681,0.044829404,0.044025483,0.043493
858,0.042754769,0.042045936,0.04093082,0.040438094,0.039677394,0.039003138,0.039059
326,0.037416907,0.036993335,0.036777227,0.036474676,0.035657789,0.034901412,0.03429
1988,0.03369553,0.033319502,0.033146616,0.032403205,0.031733271,0.031037404,0.03012
9751)
W<-
c(12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,
```

```
42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,7
2,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88)
boot.sample<-numeric()
Y<-matrix(ncol=77,nrow=size)
B<-100
Boyce<-matrix(ncol=77,nrow=B)
for (i in 1:B){
boot.sample<-sample(X,size,T)
for (n in 1:size){
for (j in 24:100){
if (boot.sample[n]<=j & boot.sample[n]>=j-24)
Y[n,j-23]=1 else Y[n,j-23]=0
}
}
for (n in 1:77){
Boyce[i,n]<-sum(Y[1:size,n])/size/Z[n]
}
}

Media<-numeric()
Pquantil<-numeric()
Uquantil<-numeric()
Desvio<-numeric()

for (n in 1:77){
Desvio[n]<-sd(Boyce[1:100,n])
Media[n]<-mean(Boyce[1:100,n])
Pquantil[n]<-quantile(Boyce[1:100,n], probs = 0.05)
Uquantil[n]<-quantile(Boyce[1:100,n], probs = 0.95)
}

Spearman<-numeric()
for (n in 1:B){
Spearman[n]<-cor(W,Boyce[n,], method = "spearman")
}

Spearman.Media<-mean(Spearman)
Spearman.sd<-sd(Spearman)

Output<-data.frame(Media,Pquantil,Uquantil,Desvio)


write.table(Output, file = "foo.txt", sep = ",", col.names = TRUE)


dev.new(width=5, height=5)
par(ps=12)
plot(1:77,Output$Media, ylab="Observed/Expected", xlab="Habitat Suitability", ylim=c(0,20),
xlim=c(0,80),pch=20, las=1)
lines(1:77,Output$Pquantil )
lines(1:77,Output$Uquantil)
```

```
text (30,18.7, paste("Boyce index = ",format(round(Spearman.Media,3))," (sd =
",format(round(Spearman.sd,3)),")", sep=""), cex=1.1)



########## Fixed effects models - INLA ##########
### Model EGV set 5 = mod1
### Prediction


mod1 = inla(pb ~ Altimetry_Pico_100 +
+ Slope_Pico_100 +
Dist100_class2_Pico
+ Dist100_class57_Pico
+ Rain_annual_Pico_100
+ Rain_range_annual_Pico_100
+ RH_range_annual_Pico_100 + T_med_annual_Pico_100
+ T_range_annual_Pico_100,family = "binomial",
data=sdmdata,verbose=TRUE, control.compute=list(dic=TRUE), Ntrials=1)

summary(mod1) # values of DIC
plot(mod1)
inla.hyperpar(mod1)
inla.cpo(mod1)



########## Mixed effects models - SPDE ##########

### Preparing data for SPDE
### load R-packages

library (fields)
library (geoR)
library (ggplot2)
library (gridExtra)
library (INLA)
library (lattice)
library (maptools)
library (rgeos)
library (splancs)


presvals <- extract(variables, occurrencedata)
dim(presvals)

### Addition of coordinates

presvals_coord<-cbind(occurrencedata,presvals)
head(presvals_coord)
summary(presvals_coord)
```

```
### Setting random seed to always create the same
### Random set of points

set.seed(0)

### Eliminate the third column (x,y)

absences <- absences[,-3]
samp <- sample(nrow(absences), 10000)# Should we set a number
psedo_absences <- absences[samp,]
absvals <- extract(variables, psedo_absences)

### Criation of data.frame and same names
### Addition of coordinates

psedo_absences<-data.frame(psedo_absences)
colnames(psedo_absences)<-c("X","Y")
absvals_coord <- cbind (psedo_absences, absvals)

head(psedo_absences)
dim(absvals)
head(absvals)

head(absvals_coord)
summary(absvals_coord)

### 1 corresponds to presences and 0 abscences

pb <- c(rep(1, nrow(presvals_coord)), rep(0, nrow(absvals_coord)))
length(pb)
head(pb)
summary(pb)

### Create the presence absence matrix

sdmdata <- data.frame(cbind(pb, rbind(presvals_coord, absvals_coord)))
dim (sdmdata)


### Mesh
### Read shapefile or polygon (island - study area)

nc.sids <- readShapePoly("C:/1.Pittosporum_Pico_correcto/if_pico_hugoWGS8426.shp")

summary (nc.sids)
names(nc.sids)
gArea(nc.sids)

plot(nc.sids, asp=1, main="")

nc.border <- unionSpatialPolygons(nc.sids, rep(1, nrow(nc.sids)))
```

```
summary (nc.border)
names(nc.border)
gArea(nc.border)
plot(nc.border, asp=1, main="")

### Now, we use the inla.sp2segment() to extract the boundary of the
### SpatialPolygons object that contains the border of the map

nc.bdry <- inla.sp2segment(nc.border)
class(nc.bdry)

### and creates the mesh

(nc.mesh <- inla.mesh.2d(boundary=nc.bdry, cutoff=100,
max.edge=c(500, 1000)))$n
plot(nc.mesh, asp=1, main='')

class(nc.mesh)
names(nc.mesh)


### The projector matrix

### SPDE model is defined on the mesh, we need an appropriate
### specification of the linear predictor for the data response.
### The key is: random field modeled at the mesh vertices, with dimension
### "m", and a response at "n" locations
### The projector matrix A can be builded with the inla.spde.make.A function


coords_Pico <- as.matrix(sdmdata[,c("X","Y")])
head (coords_Pico)
A_Pico <- inla.spde.make.A(nc.mesh, loc=coords_Pico)
dim(A_Pico)


### The stack functionality


pb <- c(rep(1, nrow(presvals_coord)), rep(0, nrow(absvals_coord)))
sdmdata <- data.frame(cbind(pb, rbind(presvals_coord, absvals_coord)))
dim (sdmdata)
coords_Pico <- as.matrix(sdmdata[,c("X","Y")])
head (coords_Pico)
A_Pico <- inla.spde.make.A(nc.mesh, loc=coords_Pico)
spde_Pico <- inla.spde2.matern(nc.mesh, alpha=2)



### Model EGV set 5 = f.mod1
### Prediction
```

```
f.mod1 <- pa ~ 0 + a0 + ALT + S + RA + RRA + RHRA + TMED + TRA + D2 + D57 + f(spatial,
model=spde_Pico) # model with random field "f"

### The stack data is defined to include: intercept (a0); random
field(spatial=1:spde_Pico$n.spde) and covariates

stk1 <- inla.stack(data=list(pa=sdmdata$pb),
A=list(A_Pico, 1),
effect=list(
list(spatial=1:spde_Pico$n.spde),
data.frame(a0=1, ALT=sdmdata[,4],S=sdmdata[,53],
RA=sdmdata[,28],RRA=sdmdata[,43],RHRA=sdmdata[,51],TMED=sdmdata[,69],
TRA=sdmdata[,88],D2=sdmdata[,9],D57=sdmdata[,11])))


### Run the model

r1 <- inla(f.mod1, family="binomial", data=inla.stack.data(stk1),
control.predictor=list(A=inla.stack.A(stk1), link=1),verbose=TRUE,Ntrials=1,
control.compute=list(dic=TRUE, waic = TRUE, cpo= TRUE))


summary(r1) # values of DIC; WAIC; mean; standard deviation; 0.025quant; 0.5quant;
0.975quant (fixed effects);and model hyperparameters (theta 1 and theta 2 - mean; standard
deviation;
#0.025quant 0.5quant 0.975quant)


### Mean Brier score

id.dat <- inla.stack.index(stk1, 'estimation')$data
brier=mean((r1$summary.fitted.values$mean[id.dat]-sdmdata$pb)^2)
brier


### CPO

r1$cpo

### Sum log CPO

-sum(log(r1$cpo$cpo))

### Mean log CPO

-mean(log(r1$cpo$cpo))
r1$failure



### The posterior marginal distribution of variance and range (spatial process)
```

```
r1.field <- inla.spde2.result(r1, 'spatial', spde_Pico, do.transf=TRUE)


### The posterior marginal distribution of log (k)

mean(r1$marginals.hy[[2]][,1])

### The posterior marginal distribution of 1/k

1/exp(-6.504) # the value in parentheses is log (k)


### The posterior marginal distribution of nominal variance of random field

mean(r1.field$marginals.variance.nominal[[1]][,1])


### The posterior marginal distribution of the practical range (meters)

mean(r1.field$marginals.range.nominal[[1]][,1])



### Prediction of the random field

library(INLA)
library (lattice)
library (gridExtra)
summary(nc.border[1][1][1])
summary (nc.border)
names(nc.border)
nc.border[1][1]


### Border of study area

test1<-nc.border@polygons[[1]]@Polygons[[1]]@coords


### Projector matrix to a regular grid with coordinates of study area

gproj_r1<- inla.mesh.projector(nc.mesh, xlim = c(365541.9, 410241.8), ylim = c(4248929.8,
4269055.5),
### Dimensions of grid
dims = c(round((410241.8-365541.9)/100),round((4269055.5-4248929.8)/100)))
### Posterior mean
g.mean_r1 <- inla.mesh.project(gproj_r1, r1$summary.random$spatial$mean)
### Posterior standard deviation
g.sd_r1 <-inla.mesh.project(gproj_r1, r1$summary.random$spatial$sd)
```

```
### We make the values of NA corresponding to the points out of the border (study area)

class(g.mean_r1)
is.na(g.sd_r1)

require(splancs)

table(xy.in <- inout(gproj_r1$lattice$loc,
cbind(test1[,1], test1[,2])))

g.mean_r1[!xy.in] <- g.sd_r1[!xy.in] <- NA


### Projection of maps (mean and standard deviation)

library(gridExtra)
do.call('grid.arrange',
lapply(list(g.mean_r1, g.sd_r1),
levelplot, col.regions=terrain.colors(16),
xlab='', ylab='', scales=list(draw=FALSE)))


### Prediction of the response on a grid
### By computation of the posterior distributions
### Select the coordinates of the grid and the correspondent rows of predictor matrix

(nxyr1 <- c(round((410241.8-365541.9)/100),round((4269055.5-4248929.8)/100)))

prdcoo_r1 <- gproj_r1$lattice$loc[which(xy.in),]
Aprd_r1 <- gproj_r1$proj$A[which(xy.in), ]


names(variables)
colnames(pred.vals)

### Stack of prediction
stk.prd_r <- inla.stack(data=list(pa=NA), A=list(Aprd_r1
,1),
effects=list(
list(spatial=1:spde_Pico$n.spde),
data.frame(a0=1, ALT=pred.vals[,1],S=pred.vals[,50],
RA=pred.vals[,25],RRA=pred.vals[,40],RHRA=pred.vals[,48],TMED=pred.vals[,66],
TRA=pred.vals[,85],D2=pred.vals[,6]
,D57=pred.vals[,8])))

### We join the stack prediction with the stack data

stk.all <- inla.stack(stk1,
stk.prd_r1)


### We fit the model prediction with data (data stack)
```

```
r1_pred <- inla(f.mod1, family="binomial", data=inla.stack.data(stk.all),
control.predictor=list(A=inla.stack.A(stk.all),
compute=TRUE, link=1), quantiles=NULL,
control.results=list(return.marginals.random=FALSE,
return.marginals.predictor=FALSE),
control.compute=list(config=TRUE),verbose=TRUE, Ntrials=1)


### Map projection

stk.all$data$index
stk1$data$index
stk.prd_r1$data$index


nxy<-c(round((410241.8-365541.9)/100),round((4269055.5-4248929.8)/100))
sd.prd <- m.prd <- matrix(NA,  nxy[1], nxy[2])

media=r1_pred$summary.fitted.values$mean[17270:61414]
sd=r1_pred$summary.fitted.values$sd[17270:61414]


m.prd[xy.in]=media
sd.prd[xy.in]=sd


library(gridExtra)
do.call('grid.arrange', lapply(list(m.prd, sd.prd),
levelplot, col.regions=terrain.colors(30), xlab='', ylab='', scales=list(draw=FALSE)))

levelplot (m.prd, col.regions=terrain.colors(16), xlab='', ylab='', )
levelplot (sd.prd, col.regions=terrain.colors(16), xlab='', ylab='', )



### Prediction without random field

### We split the sample into two subsets because of memory pc

### Data

dim(sdmdata)
head(sdmdata)
colnames(prdcoo_r102_D_13)<-c("X","Y")
data_global<-cbind(prdcoo_r1,pred.vals)
pb<-rep(NA, 44145)
data_global_pb<-data.frame(pb,data_global)
dim(data_global_pb)
global<-rbind(sdmdata,data_global_pb)
dim(global)
```

```
subset_1<-global[1:40000,]
dim(subset_1)

subset_2<-global[c(1:17269,40001:61414),]
dim(subset_2)



### Model_Subset_1

mod1 = inla(pb ~ Altimetry_Pico_100 +
+ Slope_Pico_100 +
Dist100_class2_Pico
+ Dist100_class57_Pico
+ Rain_annual_Pico_100
+ Rain_range_annual_Pico_100
+ RH_range_annual_Pico_100 + T_med_annual_Pico_100
+ T_range_annual_Pico_100,family = "binomial",
data=subset_1,verbose=TRUE, control.compute=list(dic=TRUE), Ntrials=1)


### Model_Subset_2

mod1 = inla(pb ~ Altimetry_Pico_100 +
+ Slope_Pico_100 +
Dist100_class2_Pico
+ Dist100_class57_Pico
+ Rain_annual_Pico_100
+ Rain_range_annual_Pico_100
+ RH_range_annual_Pico_100 + T_med_annual_Pico_100
+ T_range_annual_Pico_100,family = "binomial",
data=subset_2,verbose=TRUE, control.compute=list(dic=TRUE), Ntrials=1)



## Map projection


mean_1=mod1$summary.fitted.values$mean[17270:40000]
mean_1_f<-exp(mean_1)/(1+exp(mean_1))

sd_1=mod1$summary.fitted.values$sd[17270:40000]


mean_2=mod$summary.fitted.values$mean[17270:38683]
mean_2_f<-exp(mean_2)/(1+exp(mean_2))

sd_2=mod1$summary.fitted.values$sd[17270:38683]
```

```
mean<-c(media_1_f, media_2_f)
sd<-c(sd_1, sd_2)


### Matrix

nxy<-c(round((410241.8-365541.9)/100),round((4269055.5-4248929.8)/100))

sd.prd <- matrix(NA,  nxy[1], nxy[2])
m.prd <- matrix(NA,  nxy[1], nxy[2])

m.prd[xy.in]=mean
sd.prd[xy.in]=sd


library(gridExtra)
do.call('grid.arrange', lapply(list(m.prd, sd.prd),
levelplot, col.regions=terrain.colors(30), xlab='', ylab='', scales=list(draw=FALSE)))
```