

# Gap analysis

Ivan Gonzalez - [igonzalez@humboldt.org.co](mailto:igonzalez@humboldt.org.co)

Wednesday, December 02, 2015

This is an code adaptation to develop an spatial gap index analysis sensu [A methodological framework to quantify the spatial quality of biological databases](#)

The follow R code is an adaptation to get a final gap index map. For save the results in .PDF an .PNG please uncomment the lines starting with #pdf(), #png(), and #dev.off()

The first step is load information. For this is required to have both a database with biological records and the GIS layers. The GIS information can be retived from [climatic folder](#) and [bias factors](#).

The datase must be 'data.frame' class table with the follow columns: \* Unique identifier \* species name \* latitude \* longitude

We validate the original records in [taxonomical](#) and [geographical](#) way with available codes. For this reason we also include an *acceptedNameUsage* field. In order to replicate this analysis we recommend tho change the column names to showed below instead of changing script if you are not R user.

```
DATOS <- read.csv('root/DATOS.csv', as.is = TRUE)
head(DATOS)
```

```
##   ID source  lat    lon      species acceptedNameUsage
## 1 415 GBIF 12.55 -81.73 Passiflora suberosa Passiflora suberosa
## 2 416 GBIF 12.58 -81.70 Scleria melaleuca Scleria gaertneri
## 3 417 GBIF 12.58 -81.70 Eleocharis caribaea Eleocharis geniculata
## 4 419 GBIF 12.58 -81.70 Cyperus rotundus Cyperus rotundus
## 5 421 GBIF 12.58 -81.70 Cyperus luzulae Cyperus luzulae
## 6 423 GBIF 12.53 -81.72 Acalypha alopecuroides <NA>
```

Consider change the folder paths, check the encoding format, tables separator values and install libraries dependences due the compatibility system. In our case we use the follow parameters:

```
## R version 3.0.1 (2013-05-16)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=Spanish_Colombia.1252  LC_CTYPE=Spanish_Colombia.1252
## [3] LC_MONETARY=Spanish_Colombia.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Colombia.1252
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## other attached packages:
## [1] classInt_0.1-21 SPECIES_1.0       rgdal_0.9-2      maptools_0.8-34
## [5] raster_2.3-40  sp_1.0-17
##
## loaded via a namespace (and not attached):
## [1] class_7.3-7      digest_0.6.3     e1071_1.6-1     evaluate_0.5.5
## [5] foreign_0.8-54   formatR_1.0      grid_3.0.1      htmltools_0.2.6
## [9] knitr_1.8        lattice_0.20-15  rmarkdown_0.3.11 stringr_0.6.2
## [13] tools_3.0.1     yaml_2.1.13
```

The subproduct derived from the script could take several minutes to be created. For this reason the most belated objects won't be re-reprocessed if already exist in the outpath folder (**outPlotDir**). If you wan to create it again just remove them from containign folder or set a new outpath.

## 1. Data loading

The initial step is prepare data, paths and functions and load libraries.

```
# -----
# 0. Data loading
# -----
library(raster)
library(maptools)
library(rgdal)
library(SPECIES)
library(classInt)

source('root/GAPfunctions.R')
outPlotDir <- 'root/output'
ruta_factores <- 'root/GIS/FACTORES'
ruta_ambientales <- 'root/GIS/AMBIENTALES'

colombia <- readShapePoly(paste0(ruta_factores, "/COLOMBIA.shp"))
grilla <- raster(paste0(ruta_factores, "/grilla 10k.img"))
grilla[] <- 1:ncell(grilla)

en_area <- mask(grilla, colombia)

## Assign the pixel ID to each coordinate
DATOS2 <- DATOS[, c('ID', 'lat', 'lon')]
coordinates(DATOS2)=~lon+lat
celdas <- raster:::extract(en_area, DATOS2)
DATOS$celdas <- celdas

rm(celdas)
```

As result of this step we shold have a path's to bias factor layers, climate layers and output plots and the follow objects: 1. Original table (DATOS) 2. Point polygon layer (DATOS2) 3. Polygon layer (mask) of the region of interest (colombia) 4. An 'ID' raster of area of interest (grilla) 5. 'Masked' ID raster with the polygon of area of interest (en\_area) 6. A column indicating the pixel in which each pairs of coordinates over ('celdas' column in DATOS table)

## 2. Records density

As the article show, there's tree components to calculet the gap index. The first is the record density and is estimated as the number of records over each pixel:

```
# -----
# 1. Records density
# -----
```

```

DATOS2$CONT <- 1
if (file.exists('INDICE_DENSIDAD.tif')){
  densRegistros <- raster('INDICE_DENSIDAD.tif')
} else {
  densRegistros <- rasterize(DATOS2, en_area, field = DATOS2$CONT, fun = sum, na.rm = TRUE)
  densRegistros[is.na(densRegistros[])] <- 0
  densRegistros <- mask(densRegistros, colombia, filename = "INDICE_DENSIDAD.tif", overwrite = TRUE)
}

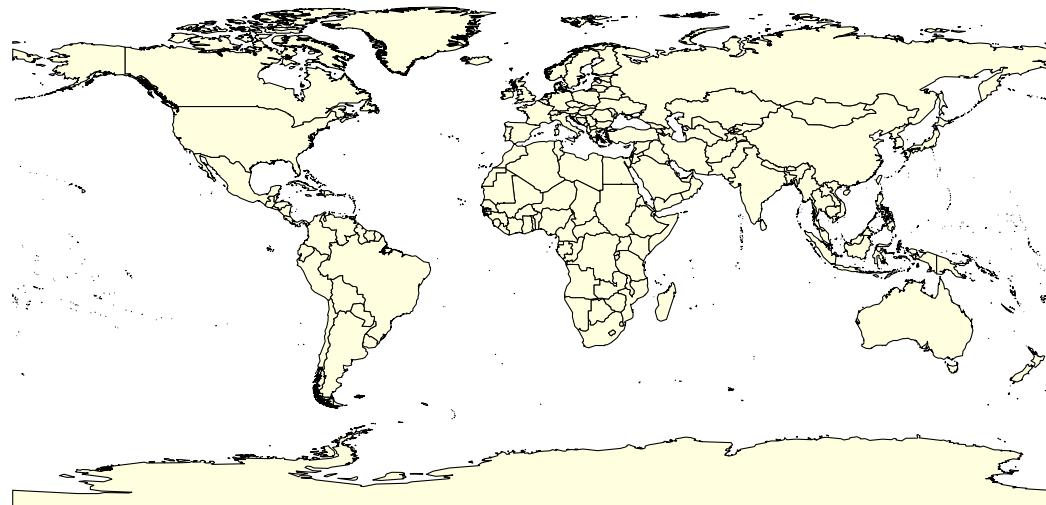
if (file.exists('localidades.tif')){
  localidades <- raster('localidades.tif')
} else {
  localidades <- rasterize(DATOS2[, 'CONT'], en_area, fun = function(x, ...) {
    length(unique(na.omit(x)))
  }, filename = "localidades.tif")[[2]]
}

# Let's generate a png file with the density plot
data(wrld_simpl)
par(mfrow = c(1,1))

#png(file = paste0(outPlotDir, "/1-INDICE_densidad.png"))
plot(wrld_simpl, main="UBICACION DE LOS REGISTROS \n BASE DE DATOS INICIAL", col = 'lightyellow')
points(x=coordinates(DATOS2)[, 1], y=coordinates(DATOS2)[, 2], col = rgb(139, 0, 0, 100, maxColorValue =

```

UBICACION DE LOS REGISTROS  
BASE DE DATOS INICIAL



```
plot(colombia, main = "UBICACION DE LOS REGISTROS COLOMBIA")
plot(DATOS2, add = TRUE, pch = 20, col = rgb(139, 0, 0, 100, maxColorValue = 255, alpha = 0.2))
```

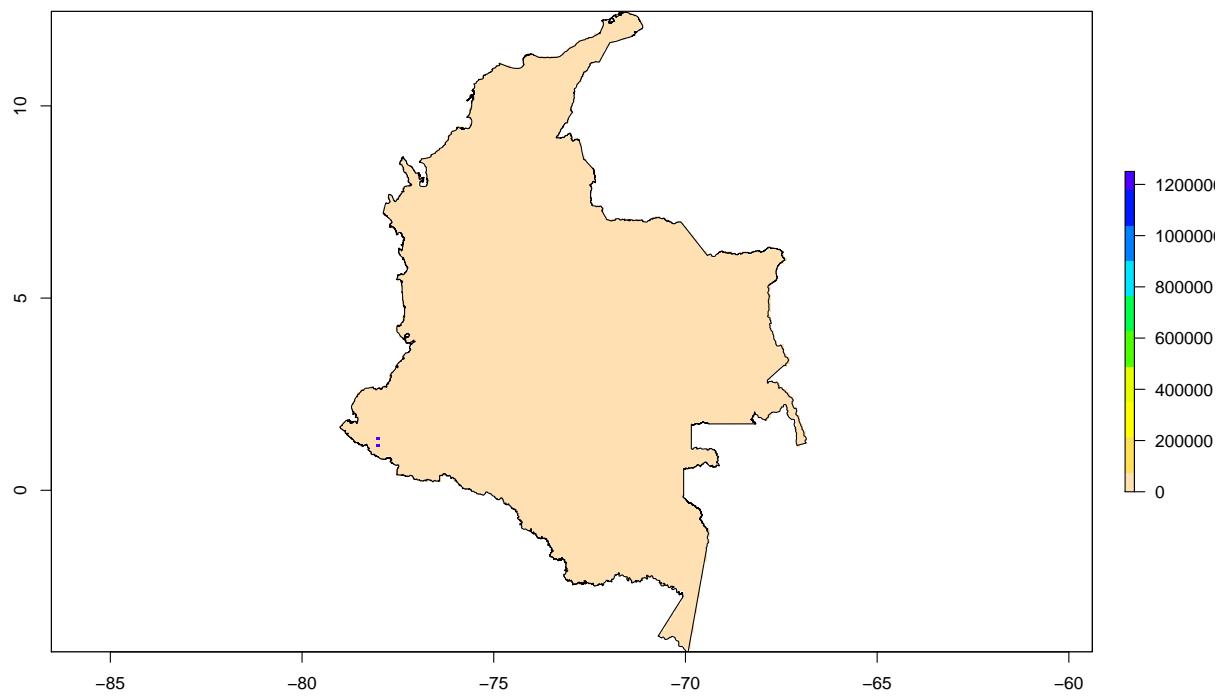
UBICACION DE LOS REGISTROS COLOMBIA



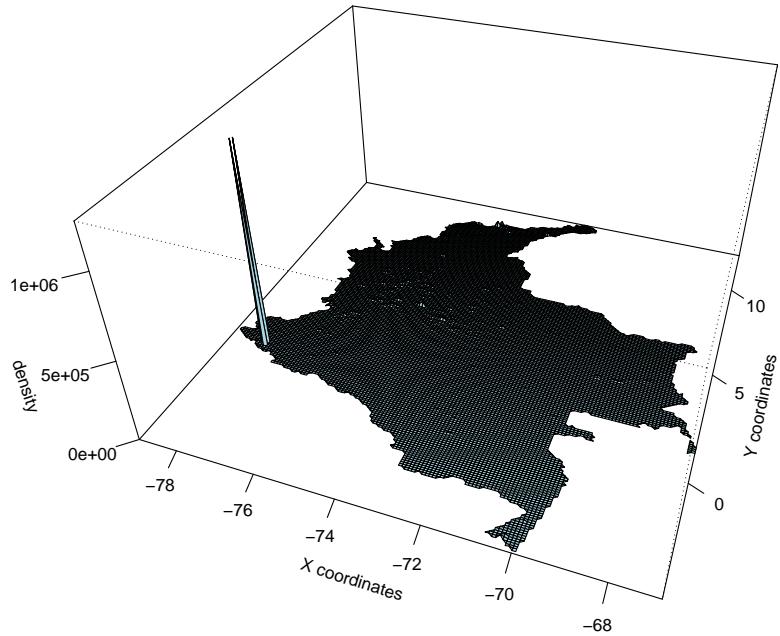
```
#dev.off()
```

```
# Let's generate a png file with the density plot. Could take several minutes because of table size
#pdf(file = paste0(outPlotDir, "/1.INDICE_densidad.pdf"))
plot(densRegistros, col = rev(topo.colors(10)), main="DENSIDAD DE PUNTOS/ 10Km2")
plot(colombia, add = TRUE)
```

DENSIDAD DE PUNTOS/ 10Km2



```
persp(densRegistros, xlab = "X coordinates", ylab = "Y coordinates", zlab = "density",
      phi = 35, theta = 20, col = "lightblue", expand = .5, ticktype = "detailed")
```

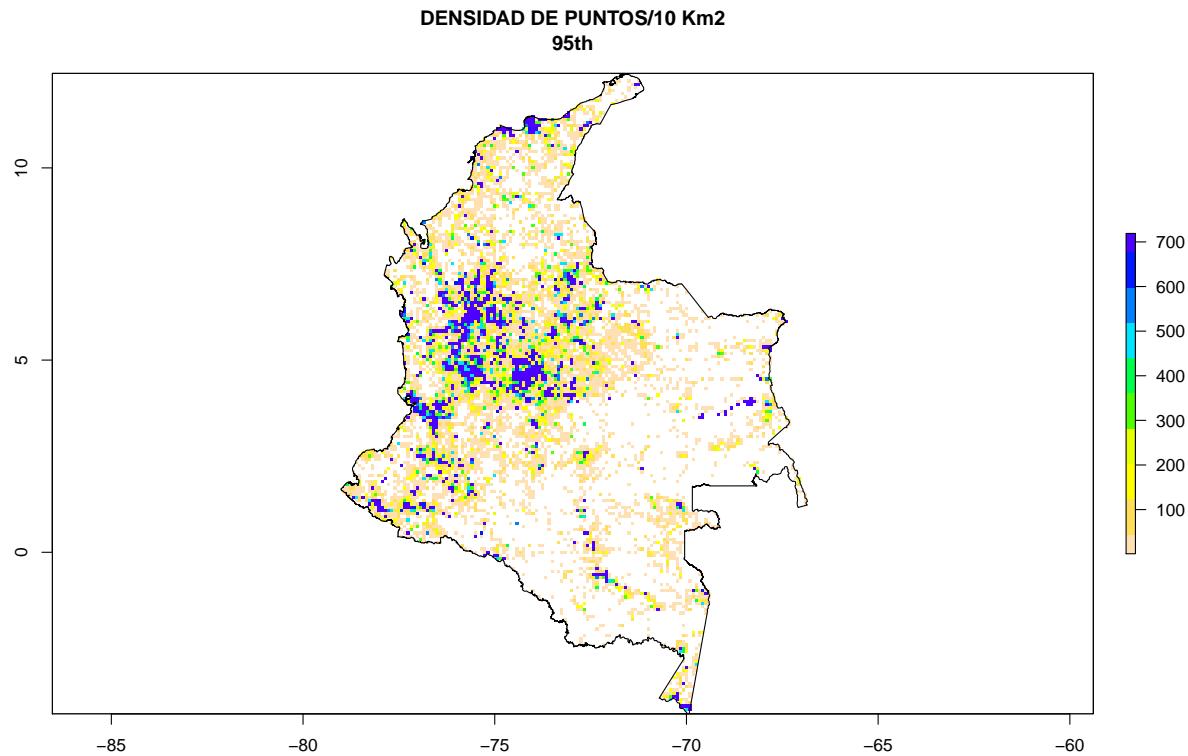


In our data we observe some points with high-density that make a particular configuration of color ramp. For this we are going to set the max value as the 95th percentile and observe in better scale the result. Note the difference in legend magnitude.

```

q95 <- quantile(densRegistros, c(.95), na.rm = TRUE)
maxVal <- cellStats(densRegistros, 'max')
densRegistros2 <- reclassify(x = densRegistros, matrix(c(q95, maxVal, q95, 0, 0, NA), nrow = 2, ncol =
densRegistros2[densRegistros2[] == 0] <- NA
plot(densRegistros2, col = rev(topo.colors(10)), main = "DENSIDAD DE PUNTOS/10 Km2\n95th")
plot(colombia, add = TRUE)

```



### 3. Bias in sampling

The second element for the gap index is a bias layer. Section 2.

The follow code will show the bias as distance function taking some physical layers as reference. The result is just informative will not take in account for GSI index estimation.

The third plot panel *bias* could be interpreted as a z-test score. In this sense values above 1.64 or bellow -1.64 indicate statistical difference in that category.

```
# -----
# 2. Bias in sampling
# -----
DATOS_s <- unique(DATOS[, c('lat', 'lon')])
N <- nrow(DATOS_s)

# -----
# 2.1 Bias by phisical factors
# -----
AP <- readOGR(ruta_factores, "protectedAreas")

## OGR data source with driver: ESRI Shapefile
## Source: "root/GIS/FACTORES", layer: "protectedAreas"
## with 407 features
## It has 7 fields
```

```

urbano <- readOGR(ruta_factores, "urban")

## OGR data source with driver: ESRI Shapefile
## Source: "root/GIS/FACTORES", layer: "urban"
## with 1084 features
## It has 6 fields

rios <- readOGR(ruta_factores, "riversMain")

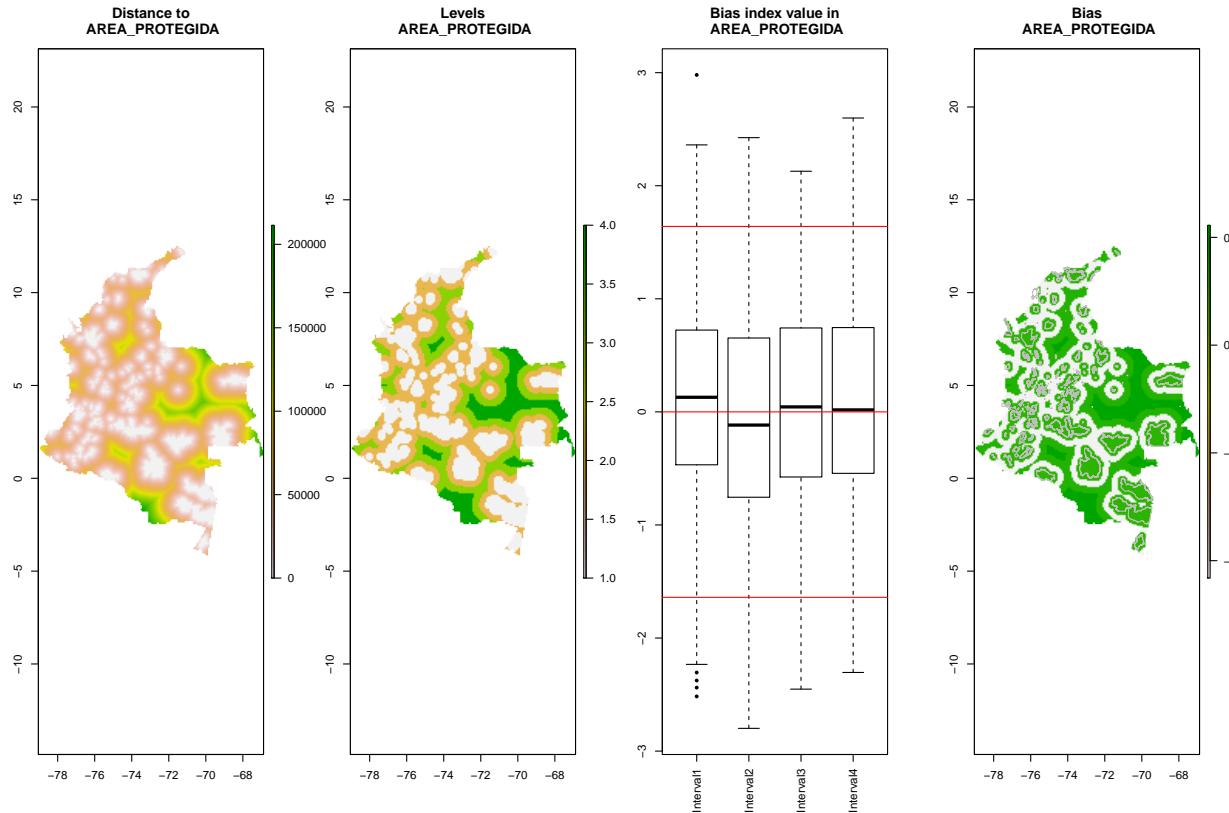
## OGR data source with driver: ESRI Shapefile
## Source: "root/GIS/FACTORES", layer: "riversMain"
## with 261 features
## It has 8 fields

vias <- readOGR(ruta_factores, "roads")

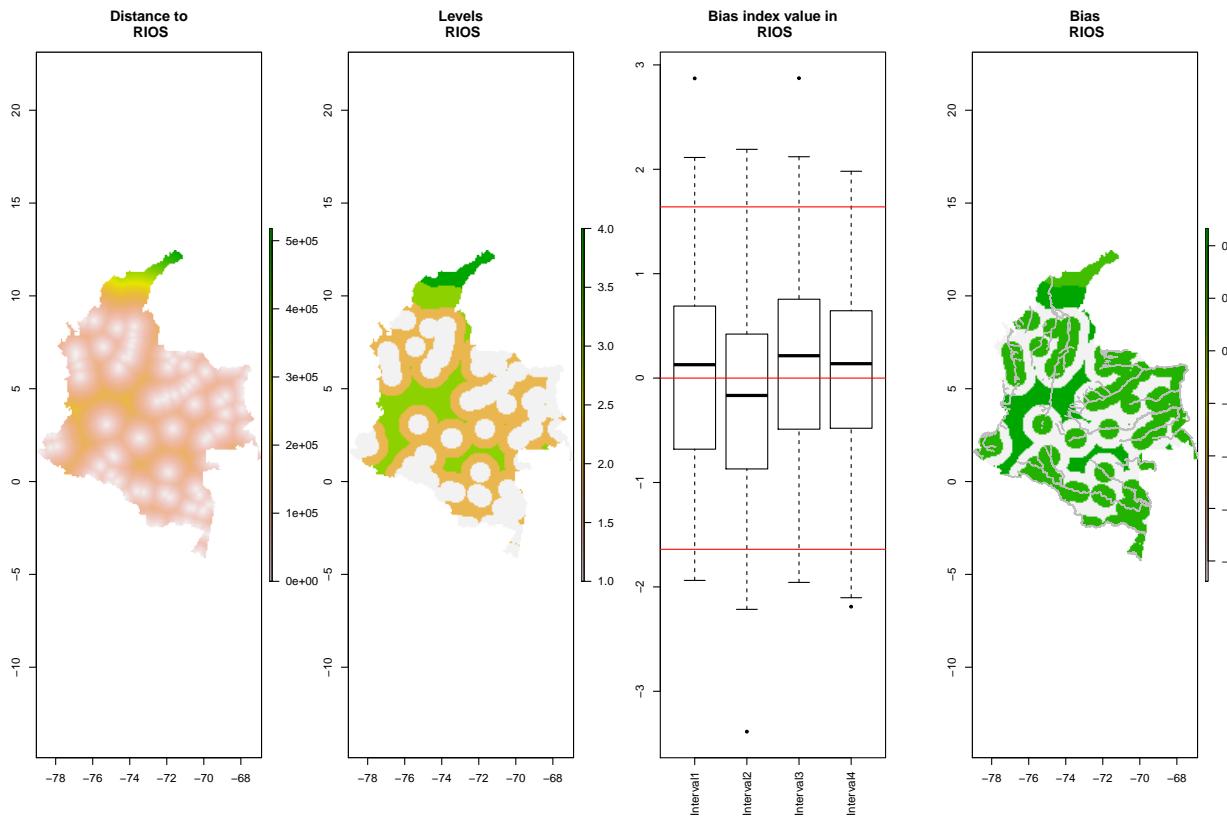
## OGR data source with driver: ESRI Shapefile
## Source: "root/GIS/FACTORES", layer: "roads"
## with 11406 features
## It has 17 fields

#pdf(file = paste0(outPlotDir, "/2.1 INDICE_factores_sesgos.pdf "))
sesgo_ap <- BIAS(biasLayer = AP, rasterMask = grilla, layerName = "AREA_PROTEGIDA", outDir = outPlotDir)

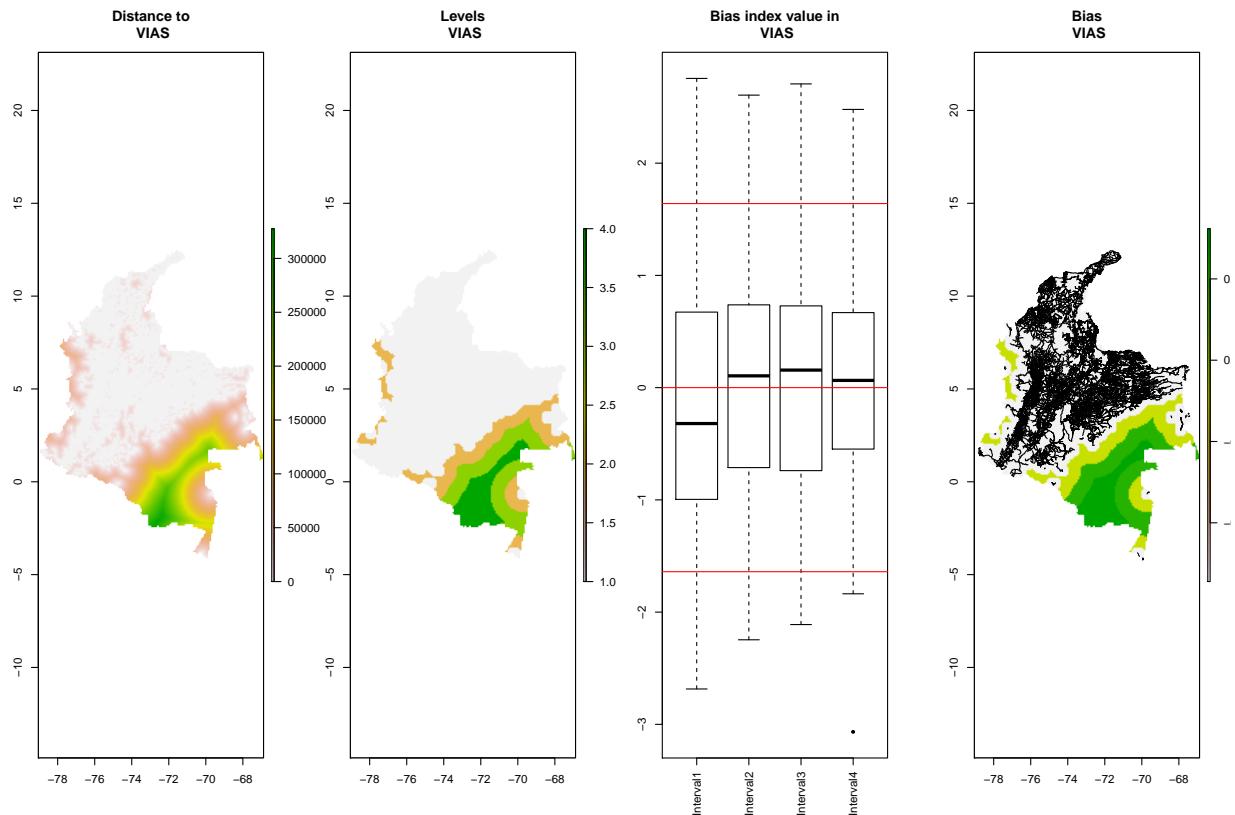
```



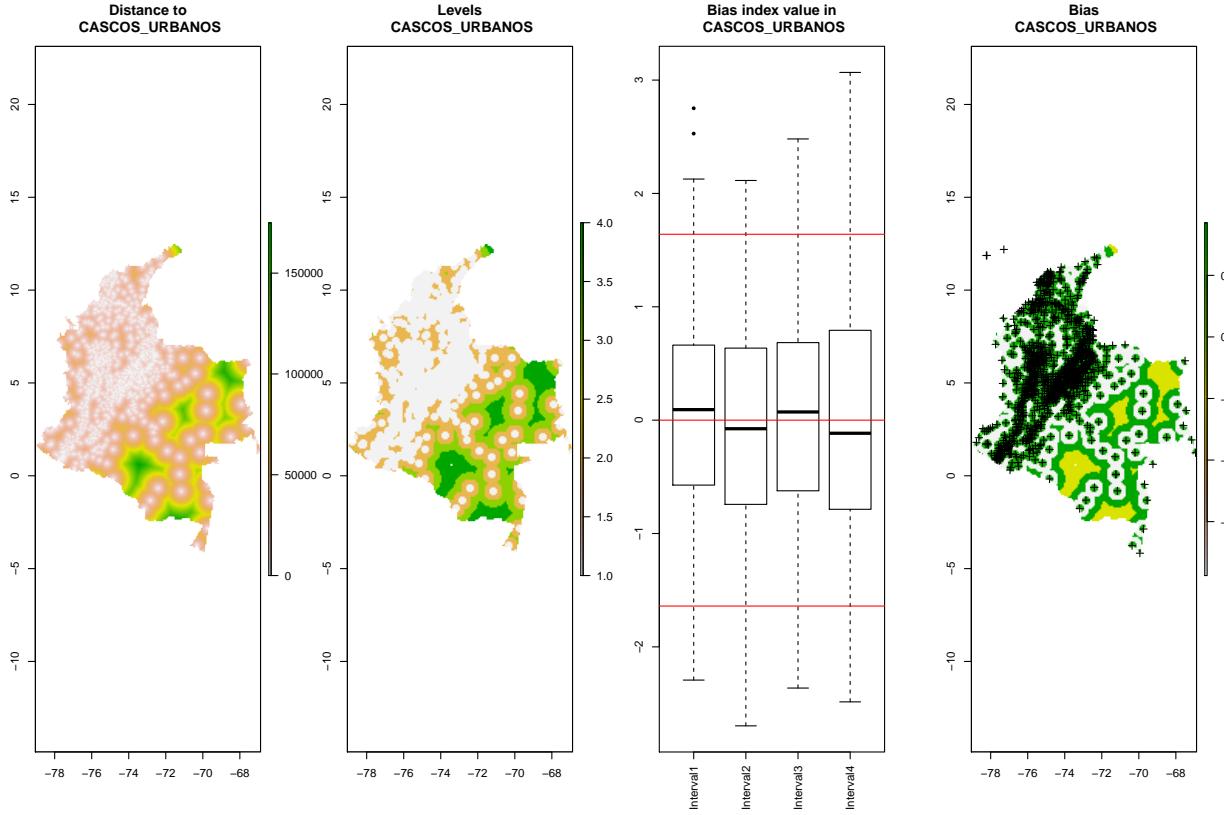
```
sesgo_rios <- BIAS(biasLayer = rios, rasterMask = grilla, layerName = "RIOS", outDir = outPlotDir)
```



```
sesgo_vias <- BIAS(biasLayer = vias, rasterMask = grilla, layerName = 'VIAS', outDir = outPlotDir)
```



```
sesgo_urbano <- BIAS(biasLayer = urbano, rasterMask = grilla, layerName = "CASCOS_URBANOS", outDir = ou
```



```
#dev.off()

# Remove extra files and keep sesgo_ap, sesgo_rios, sesgo_vias, sesgo_urbano
rm(AP, urbano, rios, vias, DATOS_s, sesgo_ap, sesgo_rios, sesgo_vias, sesgo_urbano, grilla)
```

The follow code will show the bias usi climatic layers, slope and elevation as reference. The main result is the difference between the observed values for each layer given the coordinates *vs.* a random values. A kolmogorov test is used.

```
# -----
# 2.2 Bias by environment
# -----

# 2.2.1 Compare distributions
## Compare environment all distribution values and sampled values by kolmogorov and Kullback-Leibler di

ambientales <- stack(paste0(ruta_ambientales, '/', c(paste0("bio_", 1:19), "alt", "slope")))
envVarNames <- c("TEMPERATURA-MEDIA-ANUAL", "MEDIA-RANGO-DIURNO", "ISOTERMALIDAD", "ESTACIONALIDAD-TEMP",
                 "MAX-T-MES-MAS-CALIDO", "MIN-T-MES-FRIO", "RANGO-ANUAL-T", "T-MEDIA-DEL-CUARTO-HUMEDO",
                 "T-MEDIA-DEL-CUARTO-SECO", "T-MEDIA-DEL-CUARTO-CALIDO", "T-M-DEL-CUARTO-FRIO", "PRECIPITACI",
                 "PP-MES-MAS-HUMEDO", "PP-MES-MAS-SECO", "PP-ESTACIONAL", "PP-CUARTO-HUMEDO", "PP-CUARTO-SE",
                 "PP-CUARTO-CALIDO", "PP-CUARTO-FRIO", "ALTURA", "PENDIENTE")

# Generate data.frame with all values
if (file.exists(paste0(outPlotDir, '/variablesDF.csv'))){
```

```

VARIABLES <- read.csv(paste0(outPlotDir, '/variablesDF.csv'))
} else {
  VARIABLES <- ambientales[1:ncell(ambientales)]
  VARIABLES <- na.omit(VARIABLES)
  write.csv(VARIABLES, paste0(outPlotDir, '/variablesDF.csv'), row.names = FALSE)
}

# Generate data.frame with observed values
if (file.exists(paste0(outPlotDir, '/samplingDF.csv'))){
  MUESTREO <- read.csv(paste0(outPlotDir, '/samplingDF.csv'))
} else {
  MUESTREO <- extract(ambientales, DATOS2)
  MUESTREO <- na.omit(MUESTREO)
  write.csv(MUESTREO, paste0(outPlotDir, '/samplingDF.csv'), row.names = FALSE)
}

#pdf(file = paste0(outPlotDir, "/2.2 INDICE_COMPARACION_VARIABLES.pdf "))
par(mfrow = c(4, 3))

numPredictors <- dim(ambientales)[3]
RESULT <- NULL

for (i in 1:numPredictors){

  # Generate a random vector of complete variable with sampled vector size
  sample.Var.i <- sample(VARIABLES[, i], nrow(MUESTREO), replace = T)

  # Compare both random and sampled vector
  COMPARACION <- ks.test(sample.Var.i, MUESTREO[, i], )

  VAR <- envVarNames[i]
  media_var <- mean(VARIABLES[, i])
  media_DATOS <- mean(MUESTREO[, i])
  des_var <- sd(VARIABLES[, i])
  des_DATOS <- sd(MUESTREO[, i])
  coefvar_var <- des_var/media_var
  coefvar_DATOS <- des_DATOS/media_DATOS
  D <- COMPARACION$statistic
  Pval <- COMPARACION$p.value
  if (Pval <= 0.05){
    DESICION <- "Equal distributions"
  } else {
    DESICION <- "Diferent distributions"
  }
  Nceldas <- ncell(VARIABLES[, i])
  PI <- as.data.frame(table(MUESTREO[, i])/nrow(MUESTREO))
  QTOTAL <- as.data.frame(table(VARIABLES[, i])/Nceldas)
  QI <- QTOTAL[match(PI[, 1], QTOTAL[, 1]), ]

  Q <- cbind(QI[, 2], PI[, 2])
  Q <- as.matrix(Q)
  #KLDiv(Q, overlap=F, method="discrete")
  #DIVERGENCIA=KLDiv(Q, overlap=F, method="discrete", na.rm=T)[1,2]
}

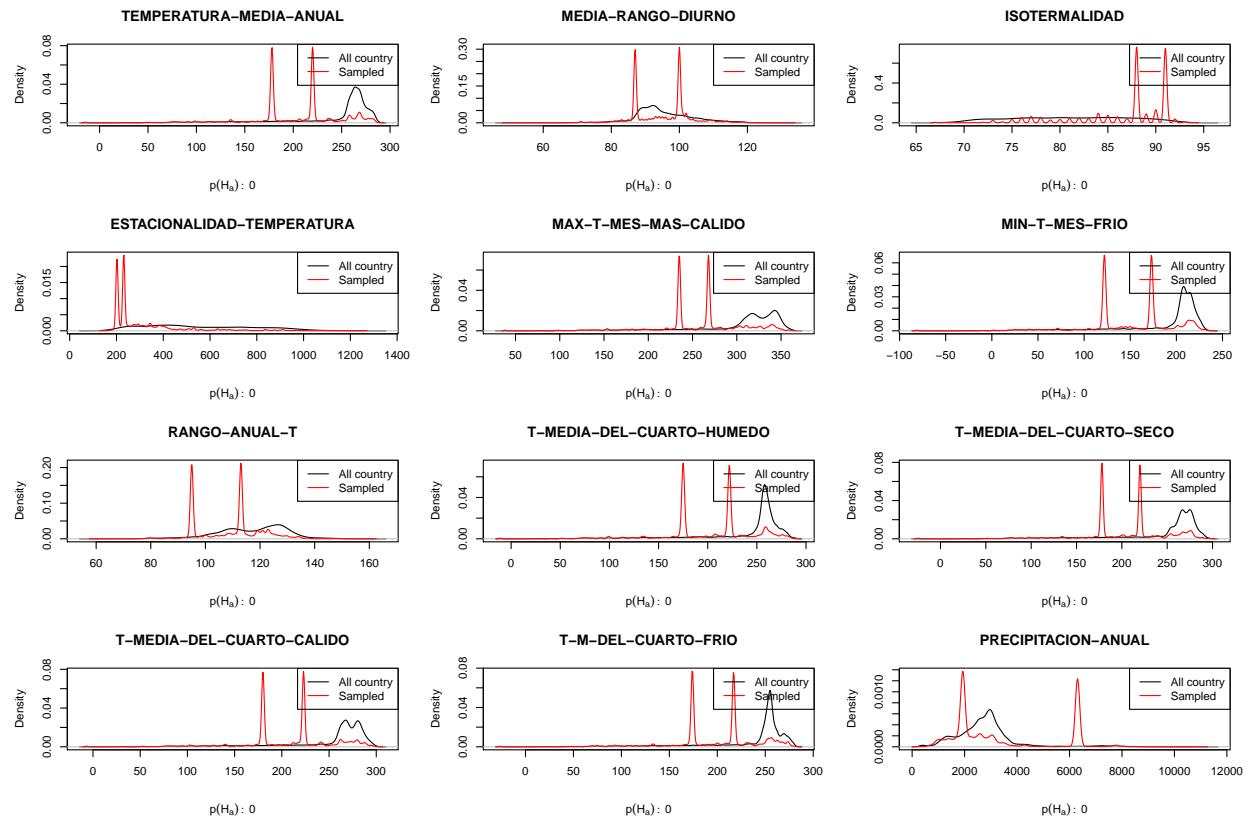
```

```

DENSI_VAR <- density(VARIABLES[, i])
DENSI_MUES <- density(MUESTREO[, i])
minx <- min(min(DENSI_VAR$x), min(DENSI_MUES$x))
maxx <- max(max(DENSI_VAR$x), max(DENSI_MUES$x))
miny <- min(min(DENSI_VAR$y), min(DENSI_MUES$y))
maxy <- max(max(DENSI_VAR$y), max(DENSI_MUES$y))

plot(density(VARIABLES[,i]), col='black', xlab = '', main=VAR, xlim=c(minx, maxx), ylim=c(miny,maxy),
      sub = bquote(p(H[alpha]): ~ .(Pval)))
lines(density(MUESTREO[,i]), col= "red", main=VAR)
legend('topright', lty = c(1, 1), col = c('black', 'red'), legend = c('All country', 'Sampled'))
RESUMEN <- cbind(VAR, media_var, media_DATOS, des_var, des_DATOS, coefvar_var, coefvar_DATOS,
                  D, Pval, DESICION)
RESULT <- rbind(RESULT, RESUMEN)
}

```



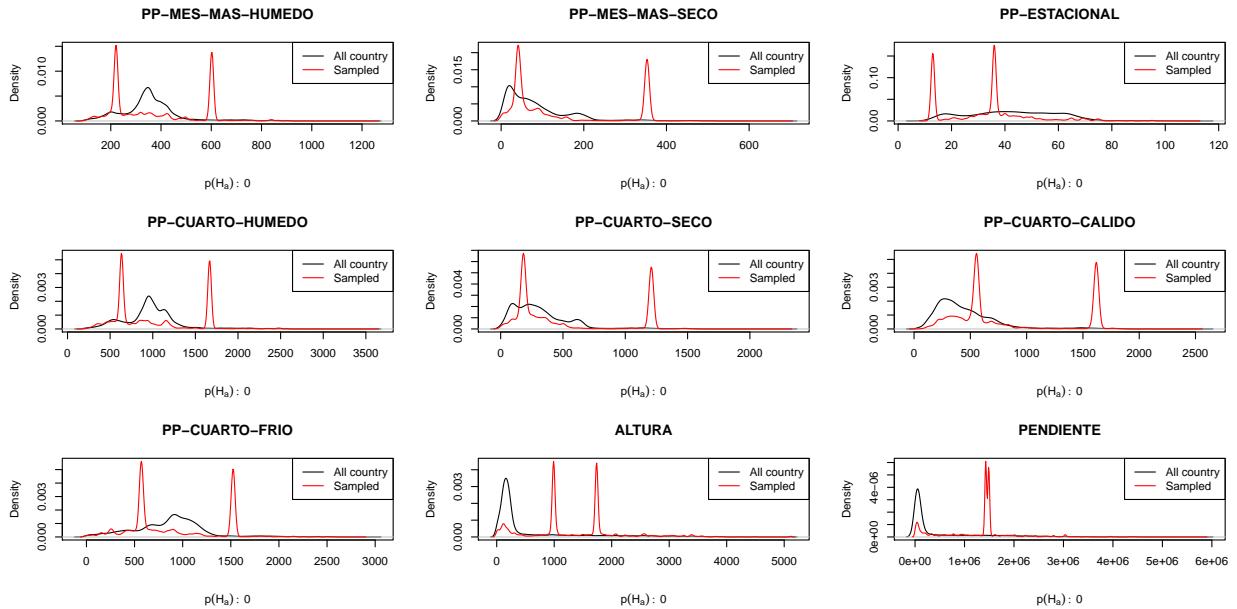
```

#dev.off()

# Write table with summary for each variable
write.csv(RESULT, "goodness_fit.csv", row.names = FALSE)

# Remove extra objects
rm(MUESTREO, VARIABLES, RESULT, RESUMEN, VAR, media_var, media_DATOS, des_var, des_DATOS, coefvar_var,
   D, Pval, DESICION, DENSI_VAR, DENSI_MUES, sample.Var.i, Q, QI, QTOTAL)

```



The climate bias layer will be estimated using `BIAS()` function for each variable. Finally all **bias** layers are summed.

### # 2.2.2. Environmental bias layer

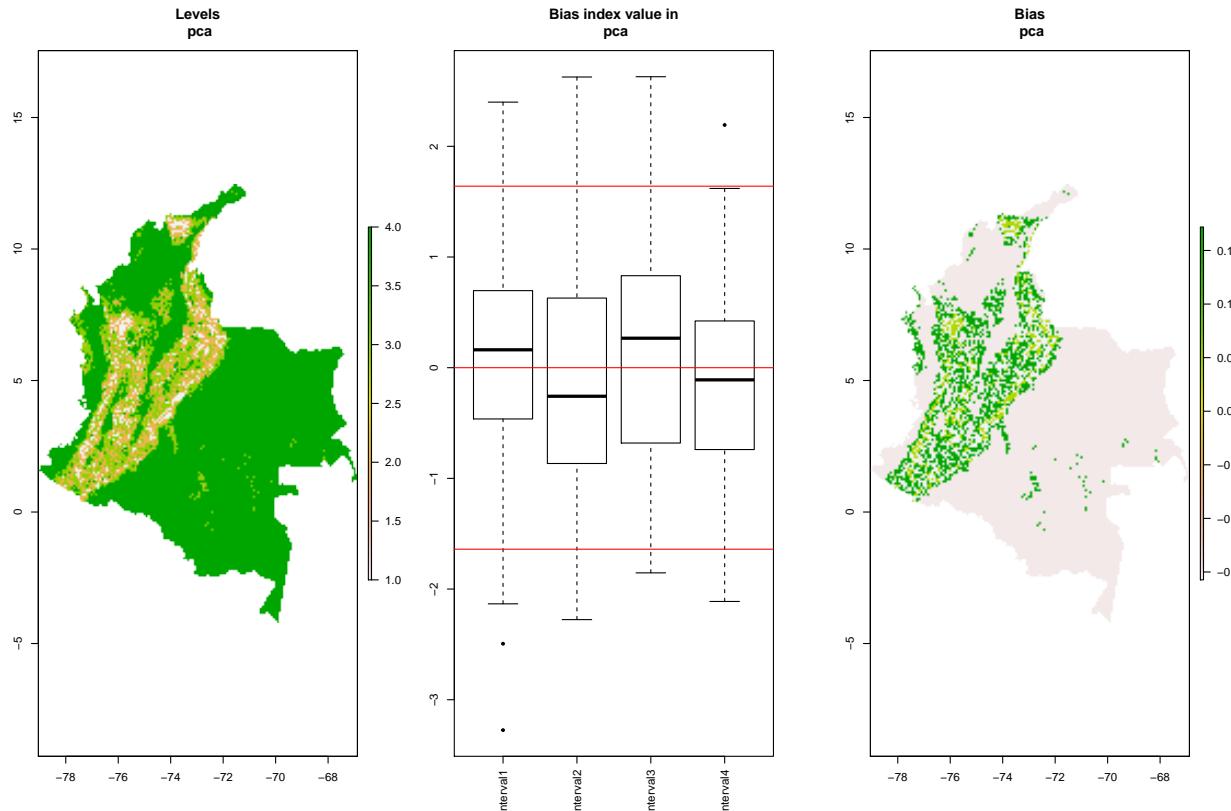
```
#pdf(file = paste0(outPlotDir, "/2.3 INDICE_sesgos_ambientales.pdf")) ##comienza la grafica tipo pdf
par(mfrow = c(1,2))
d <- ambientales[[1]] * 0
resultados <- NULL
#names(ambientales) <- envVarNames
sesgosDF <- NULL
for (k in 1:numPredictors){
  ses_amb <- BIAS(biasLayer = ambientales[[k]], layerName = envVarNames[k], outDir = outPlotDir, doplot = TRUE)
  d <- sum(d, ses_amb$biasLayer, na.rm = TRUE)
  sesgosDF <- cbind(sesgosDF, ses_amb$biasLayer[])
  resultados <- cbind(resultados, ses_amb$biasValues)
}
colnames(sesgosDF) <- names(resultados) <- envVarNames

#dev.off()

pcaLayer <- raster(paste0(ruta_ambientales, '/pcaLayer.tif'))
ses_pca <- BIAS(biasLayer = pcaLayer, layerName = 'pca', outDir = outPlotDir, doplot = TRUE)
d2 <- ses_pca$biasLayer

write.table(resultados, "sesgos_ambientales.txt", sep="\t", col.names = TRUE)
write.table(sesgosDF, "sesgos_tabla_Capas.txt", sep="\t", col.names = TRUE)
```

```
writeRaster(d, "INDICE_AMBIENTAL.tif", overwrite = TRUE)
```



#### 4. Data base completeness

The final component is the species completeness. For this case we are gonna to use Jackknife and bootstrap estimates for two database columns: original species and accepted species name.

```
# -----
# 3. Data base completeness
# -----
```

```
spListByCell <- DATOS[!is.na(DATOS$celdas), c('species', 'celdas')]
spListByCellHQ <- DATOS[!is.na(DATOS$celdas) & !is.na(DATOS$acceptedNameUsage), c('acceptedNameUsage', 'celdas')]

freqTable <- table(spListByCell$celdas)
freqTableHQ <- table(spListByCellHQ$celdas)

threshold <- 0

spListByCell <- spListByCell[spListByCell$celdas %in% names(which(freqTable >= threshold)), ]
spListByCellHQ <- spListByCellHQ[spListByCellHQ$celdas %in% names(which(freqTableHQ >= threshold)), ]

estimateS <- richEst(sppList = spListByCell$species, indexID = spListByCell$celdas)
```

```

estimateSHQ <- richEst(spListByCellHQ$acceptedNameUsage, spListByCellHQ$celdas)

rm(spListByCell, spListByCellHQ)

compRichBoot <- compRichJack <- richJackHQ <- richBootHQ <- richJack <- richBoot <- en_area * 0
richBoot[as.numeric(rownames(estimateS))] <- estimateS$Boot
richJack[as.numeric(rownames(estimateS))] <- estimateS$JNhat
richBootHQ[as.numeric(rownames(estimateSHQ))] <- estimateSHQ$Boot
richJackHQ[as.numeric(rownames(estimateSHQ))] <- estimateSHQ$JNhat

compRichBoot[as.numeric(rownames(estimateS))] <- estimateS$Sobs/estimateS$Boot
compRichJack[as.numeric(rownames(estimateS))] <- estimateS$Sobs/estimateS$JNhat

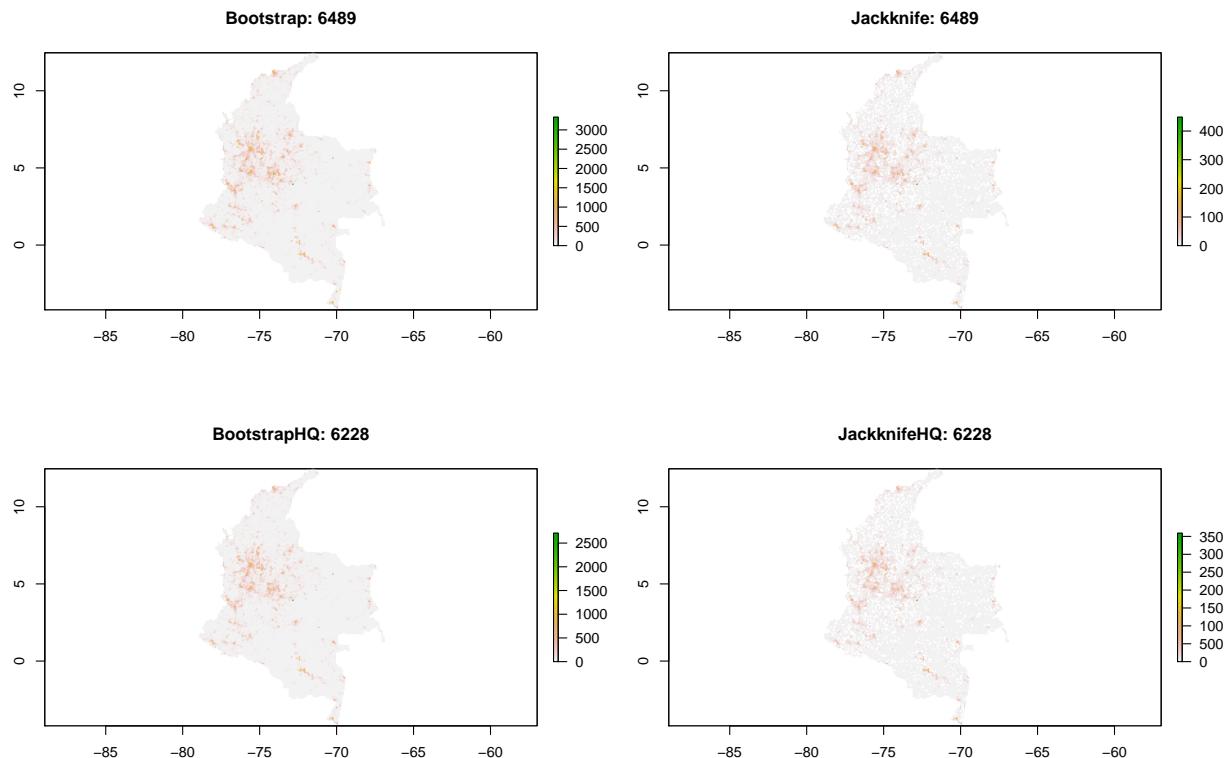
#compRichBoot[as.numeric(rownames(estimateS)[estimateS$Sobs == 1])] <- NA
#compRichJack[as.numeric(rownames(estimateS)[estimateS$Sobs == 1])] <- NA
compRichJack[compRichJack[] >= 1] <- 1
compRichBoot[compRichBoot[] >= 1] <- 1

#pdf(file = paste0(outPlotDir, "/3 INDICE_COMPLEMENTARIEDAD.pdf "))
par(mfrow=c(2, 2))

plot(richBoot, main = paste('Bootstrap:', nrow(estimateS)))
plot(richJack, main = paste('Jackknife:', nrow(estimateS)))

plot(richBootHQ, main = paste('BootstrapHQ:', nrow(estimateSHQ)))
plot(richJackHQ, main = paste('JackknifeHQ:', nrow(estimateSHQ)))

```



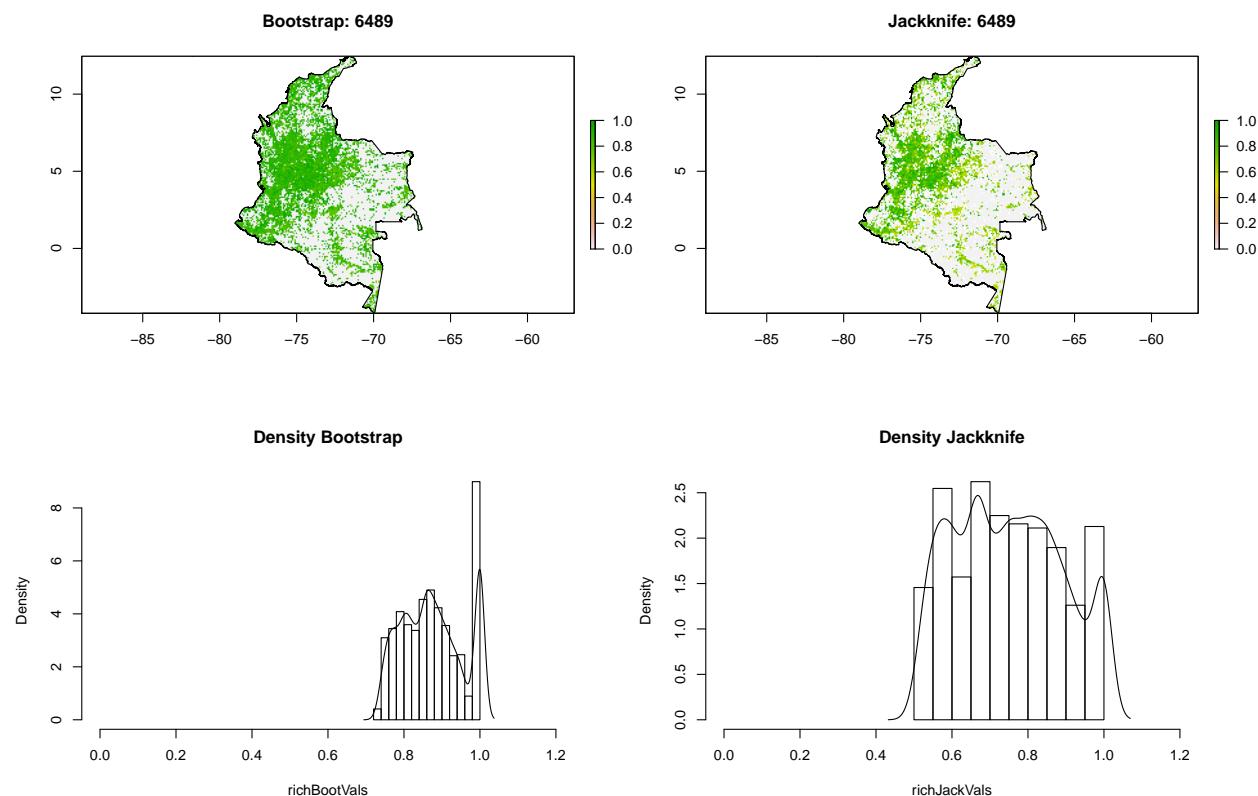
```

plot(compRichBoot, main = paste('Bootstrap:', nrow(estimateS)))
plot(colombia, add = TRUE)
plot(compRichJack, main = paste('Jackknife:', nrow(estimateS)))
plot(colombia, add = TRUE)

richBootVals <- compRichBoot[!is.na(compRichBoot[]) & compRichBoot[] != 0]
richJackVals <- compRichJack[!is.na(compRichJack[]) & compRichJack[] != 0]
hist(richBootVals, main = 'Density Bootstrap', freq = FALSE, xlim = c(0, 1.2))
lines(density(richBootVals), main = 'Density Bootstrap')

hist(richJackVals, main = 'Density Jackknife', freq = FALSE, xlim = c(0, 1.2))
lines(density(richJackVals), main = 'Density Jackknife')

```



```

#dev.off()

writeRaster(compRichBoot, "INDICE_COMPLEMENTARIEDAD_JACK.tif", overwrite=TRUE)
writeRaster(compRichJack, "INDICE_COMPLEMENTARIEDAD_BOOTS.tif", overwrite=TRUE)

```

## 5. GSI/Gap selection index

Finally sum the three standardized components using the article formula

```

# -----
# 4. GSI / Gap selection index
# -----

```

```

# Layer standardization

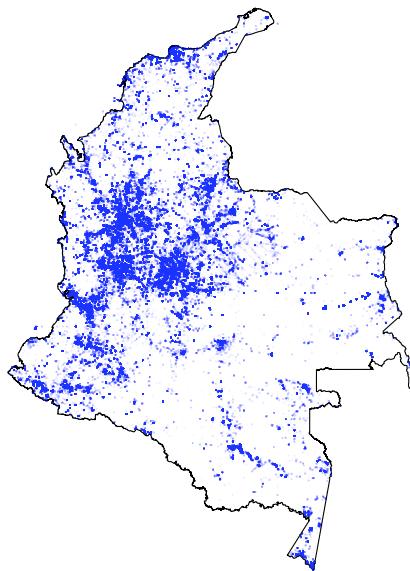
DENSIDAD <- normalize01(densRegistros)
AMBIENTAL <- normalize01(d)
COMPLEMENTARIEDAD_BOOT <- normalize01(compRichBoot)
COMPLEMENTARIEDAD_JACK <- normalize01(compRichJack)

writeRaster(DENSIDAD, "INDICE_DENSIDAD_EST.tif", overwrite=TRUE)
writeRaster(AMBIENTAL, "INDICE_AMBIENTAL_EST.tif", overwrite=TRUE)
writeRaster(COMPLEMENTARIEDAD_BOOT, "INDICE_COMPLEMENTARIEDAD_BOOT_EST.tif", overwrite=TRUE)
writeRaster(COMPLEMENTARIEDAD_JACK, "INDICE_COMPLEMENTARIEDAD_JACK_EST.tif", overwrite=TRUE)

# Plot
## Records ubicacion
#png(file = paste0(outPlotDir, "/4 INDICE_final_ubicacion_registros.png")) ##comienza la grafica tipo p
par(mfrow = c(1, 1))
plot(colombia, main = "UBICACION DE \n LOS REGISTROS")
plot(DATOS2, add = T, pch = 20, cex = 0.1, col = rgb(0.1, 0.2, 1, 0.01))

```

UBICACION DE  
LOS REGISTROS



```

#dev.off()

#pdf(file = paste0(outPlotDir, "/4 INDICE_final.pdf ")) ##comienza la grafica tipo pdf

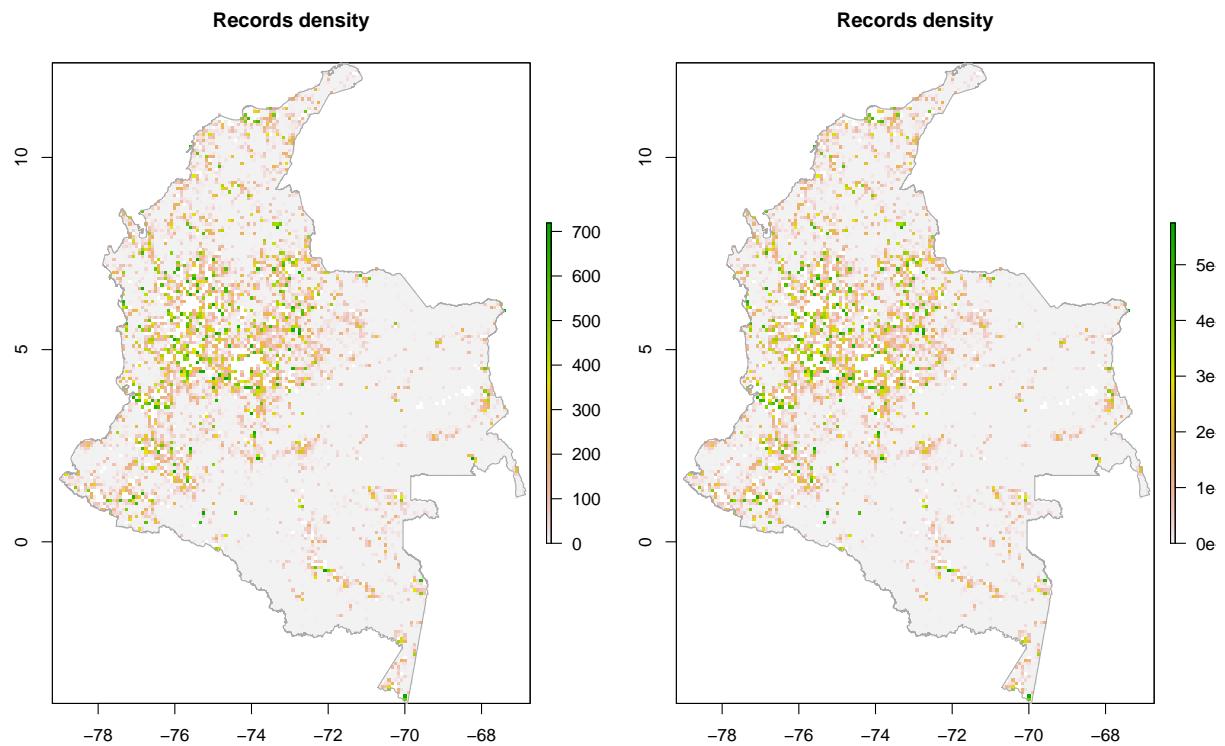
## Records density
par(mfrow=c(1, 2))
plot(densRegistros, main = 'Records density', ylim = c(0, quantile(densRegistros[], .95, na.rm = TRUE)))

```

```

plot(colombia, add = TRUE, border = 'darkgrey')
plot(DENSIDAD, main = 'Records density', zlim = c(0, quantile(DENSIDAD[], .95, na.rm = TRUE)))
plot(colombia, add = TRUE, border = 'darkgrey')

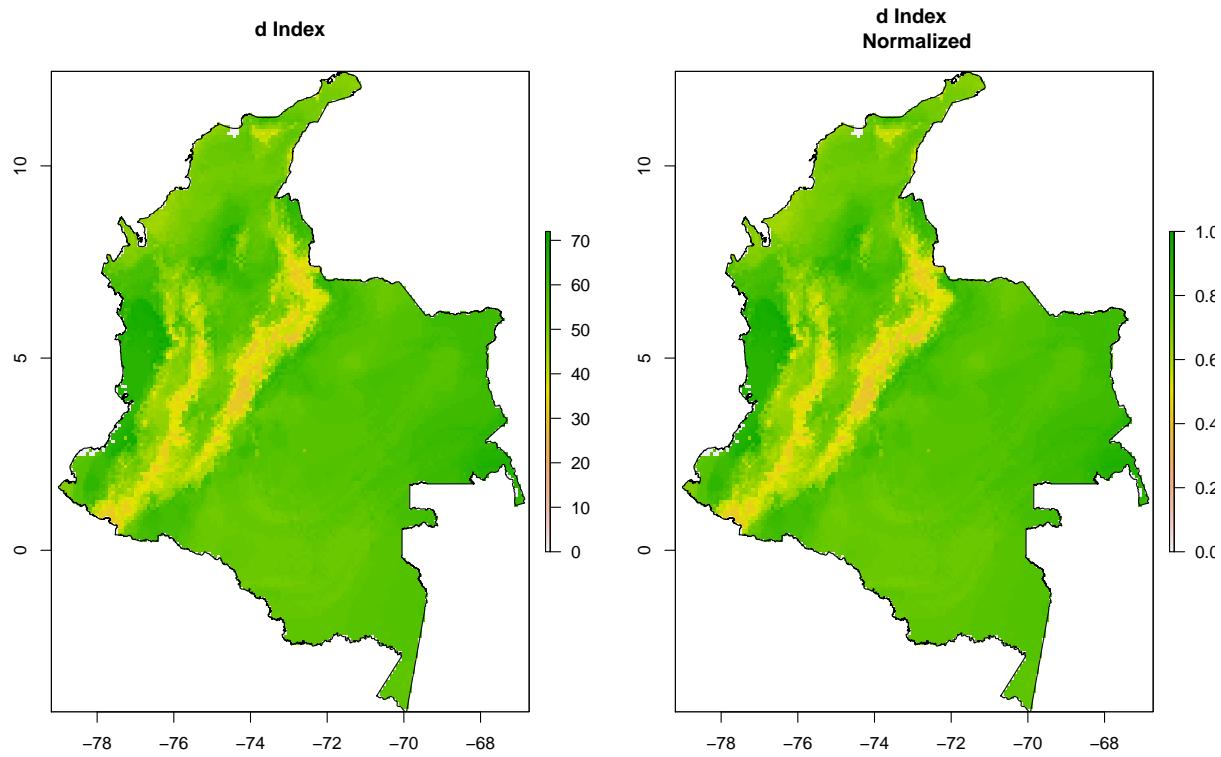
```



```

## Environmental bias
par(mfrow=c(1, 2))
plot(mask(d, colombia), main = 'd Index')
plot(colombia, add = TRUE)
plot(mask(AMBIENTAL, colombia), main = 'd Index \n Normalized')
plot(colombia, add = TRUE)

```



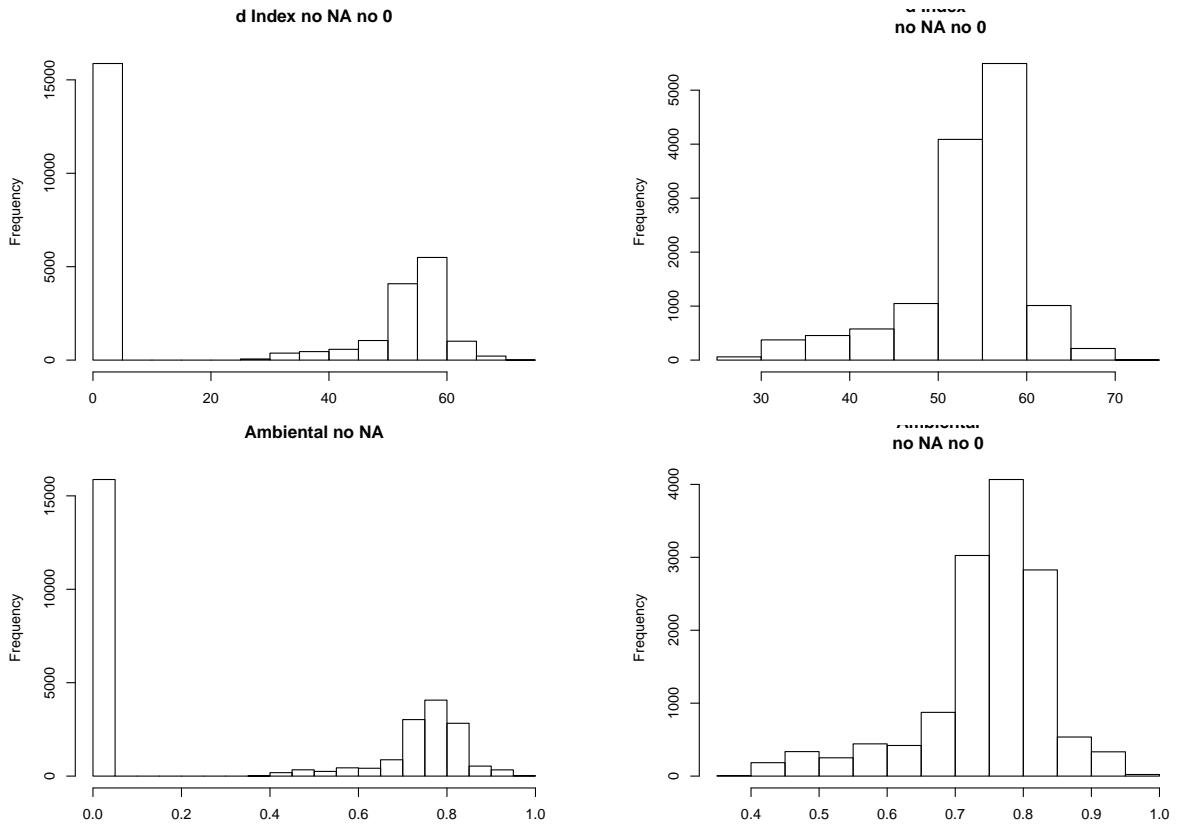
```

par(mfrow=c(2, 2))
dValsNoNA <- d[!is.na(d[])]
dValsNoNaNZero <- d[!is.na(d[])] & d[] != 0
hist(dValsNoNA, main = 'd Index no NA no 0')
hist(dValsNoNaNZero, main = 'd Index \n no NA no 0')

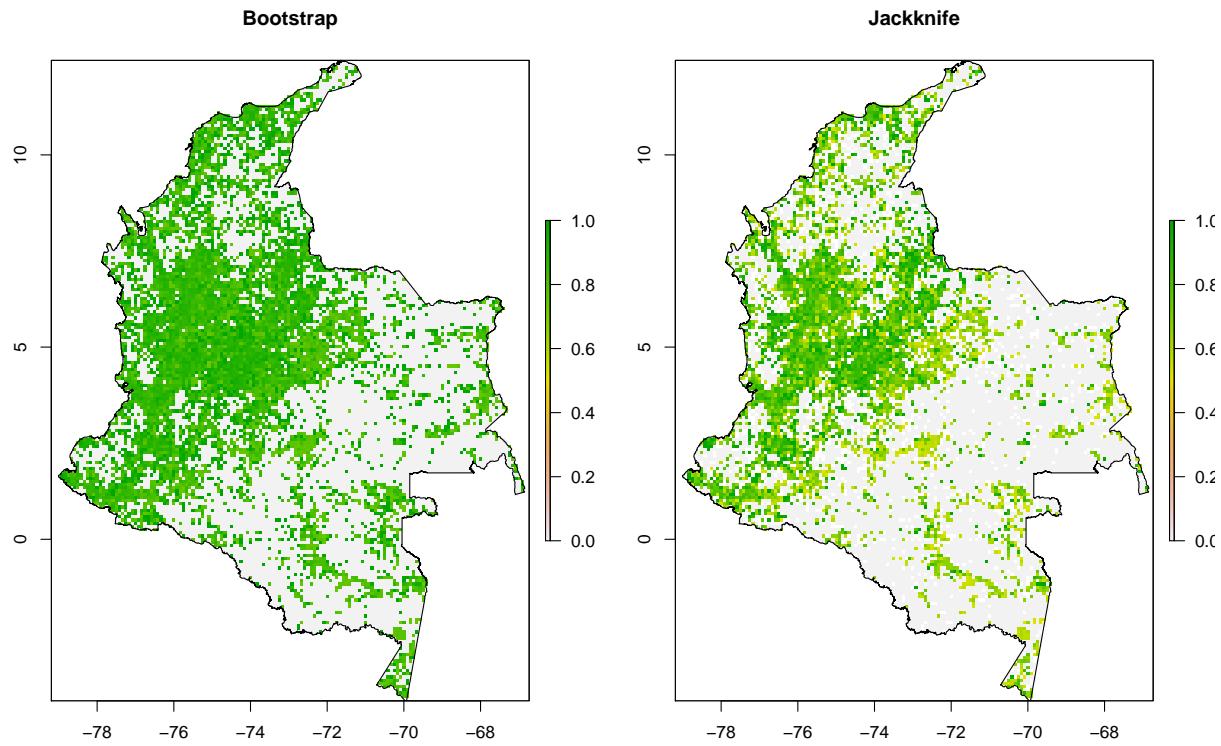
AMBValsNoNA <- AMBIENTAL[!is.na(AMBIENTAL[])]
AMBValsNoNaNZero <- AMBIENTAL[!is.na(AMBIENTAL[])] & AMBIENTAL[] != 0
hist(AMBValsNoNA, main = 'Ambiental no NA')
hist(AMBValsNoNaNZero, main = 'Ambiental\nno NA no 0')

## Completeness
par(mfrow = c(1, 2))

```



```
plot(compRichBoot, main = 'Bootstrap')
plot(colombia, add = TRUE)
plot(compRichJack, main = 'Jackknife')
plot(colombia, add = TRUE)
```



```

par(mfrow=c(2, 2))

bootValsNoNA <- compRichBoot[!is.na(compRichBoot[])]
bootValsNoNANoZeros <- compRichBoot[!is.na(compRichBoot[])] & compRichBoot[] != 0]
hist(bootValsNoNA, main = 'Bootstrap no NA')
hist(bootValsNoNANoZeros, main = 'Bootstrap\nno NA no 0')

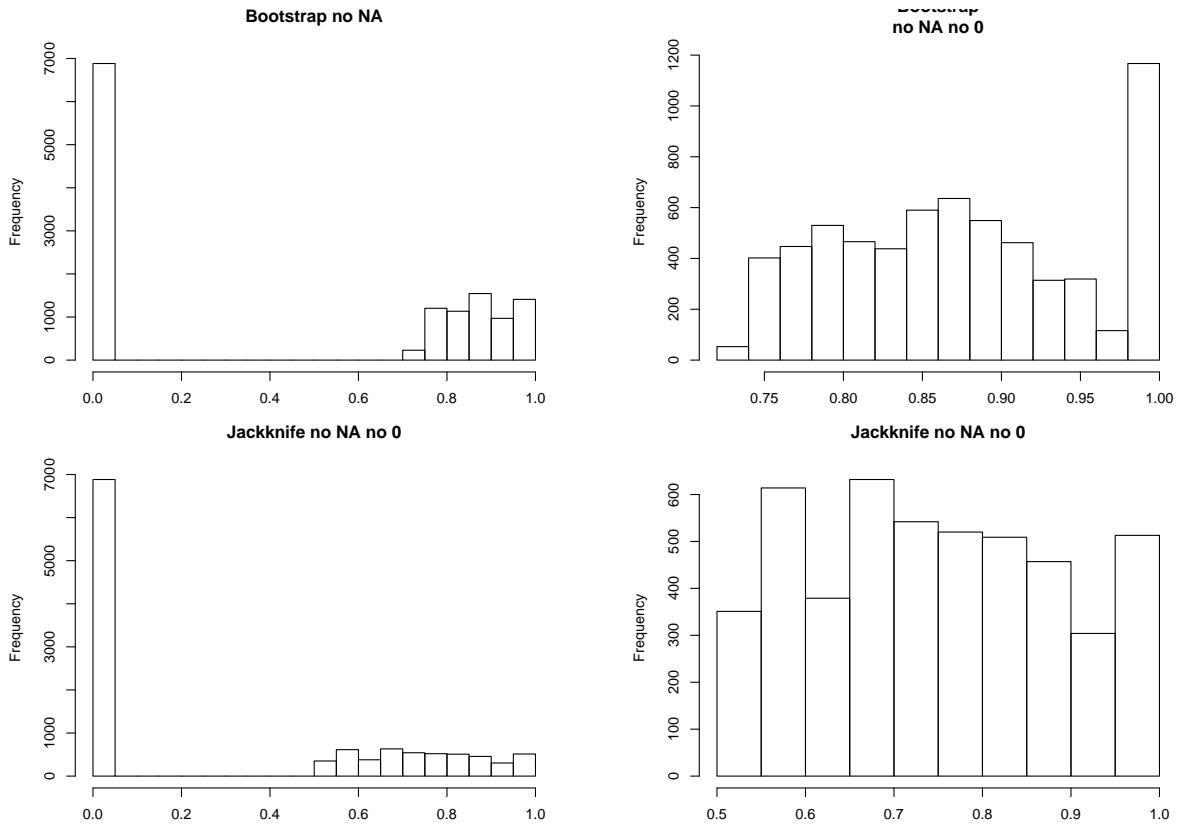
JacValsNoNA <- compRichJack[!is.na(compRichJack[])]
JacValsNoNANoZeros <- compRichJack[!is.na(compRichJack[])] & compRichJack[] != 0]
hist(JacValsNoNA, main = 'Jackknife no NA no 0')
hist(JacValsNoNANoZeros, main = 'Jackknife no NA no 0')

# GSI / GAP INDEX

GSI_BOOT <- (3 - DENSIDAD - AMBIENTAL - COMPLEMENTARIEDAD_BOOT)/3
GSI_JACK <- (3 - DENSIDAD - AMBIENTAL - COMPLEMENTARIEDAD_JACK)/3

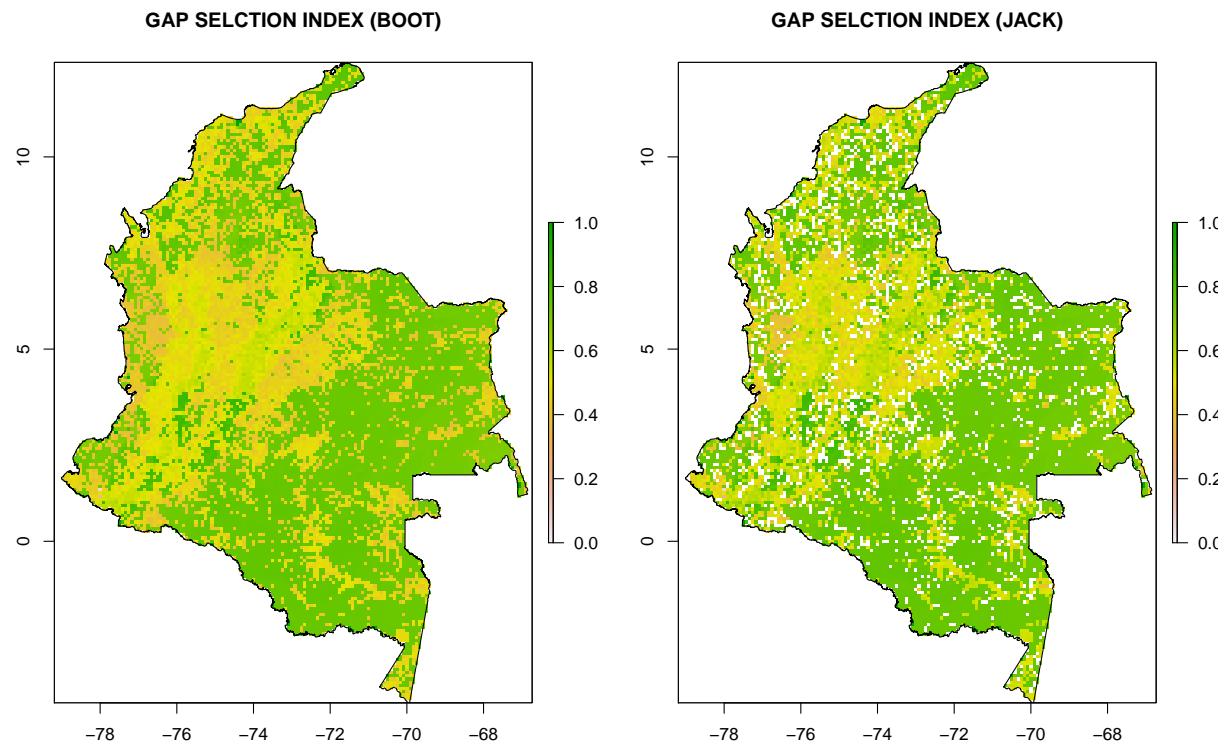
par(mfrow = c(1, 2))

```

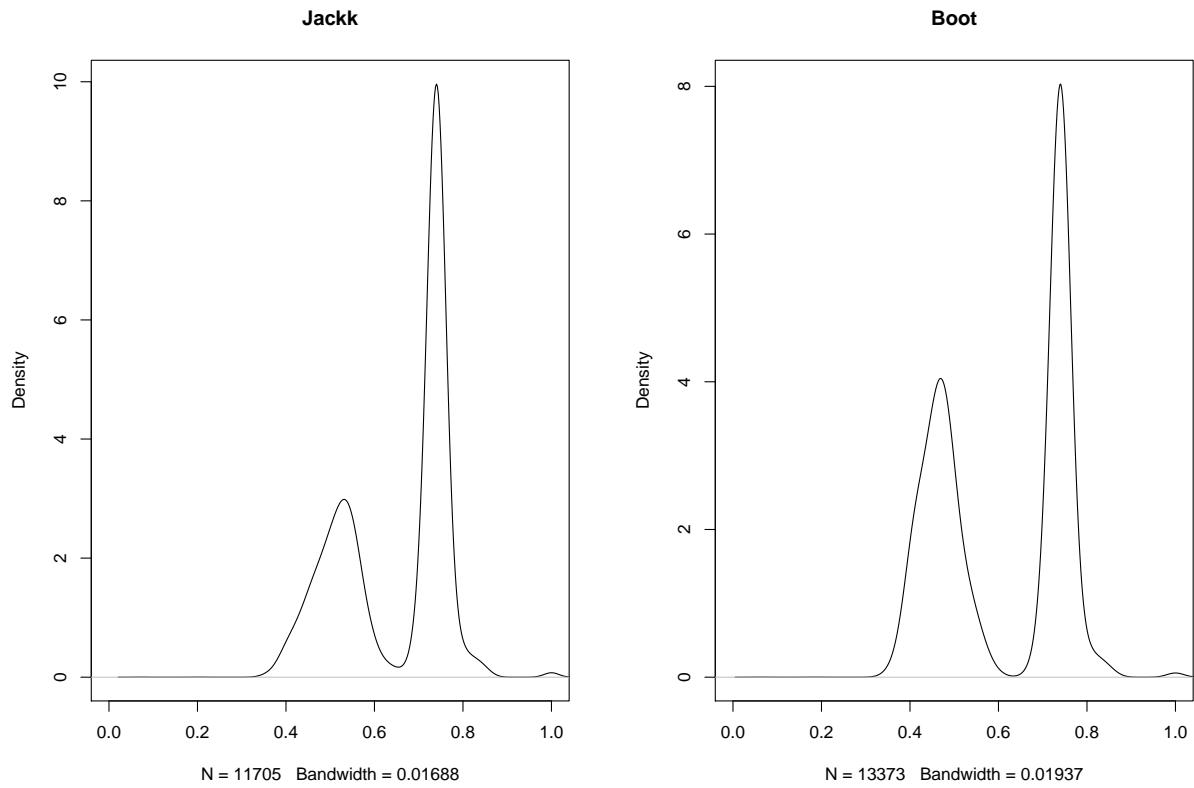


```
plot(GSI_BOOT, main = 'GAP SELCTION INDEX (BOOT)', ylim = c(0, 1))
plot(colombia, add = TRUE)

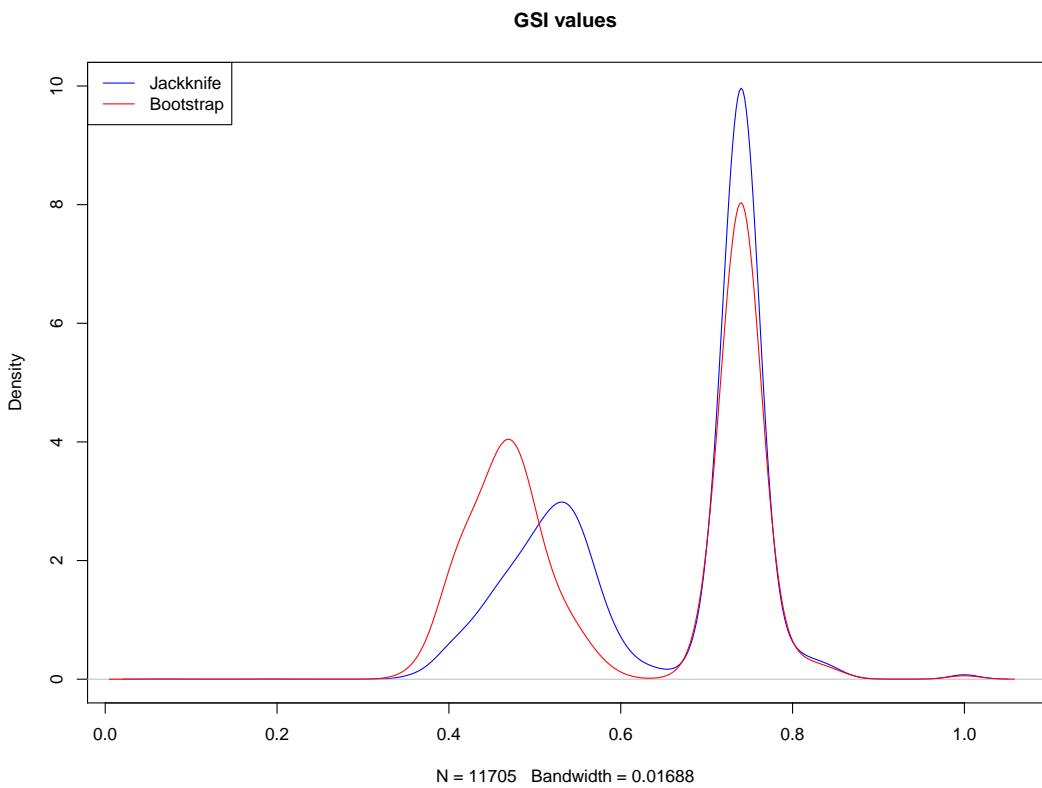
plot(GSI_JACK, main = 'GAP SELCTION INDEX (JACK)', ylim = c(0, 1))
plot(colombia, add = TRUE)
```



```
densJack <- density(GSI_JACK, main = 'Jackk', xlim = c(0, 1))
densBoot <- density(GSI_BOOT, main = 'Boot', xlim = c(0, 1))
```



```
par(mfrow = c(1, 1))
plot(densJack, main = 'GSI values', col = 'blue', ylim = c(0, 10))
lines(densBoot, col = 'red')
legend('topleft', legend = c('Jackknife', 'Bootstrap'),
      lty = c(1, 1), lwd = c(1, 1), col = c('blue', 2))
```



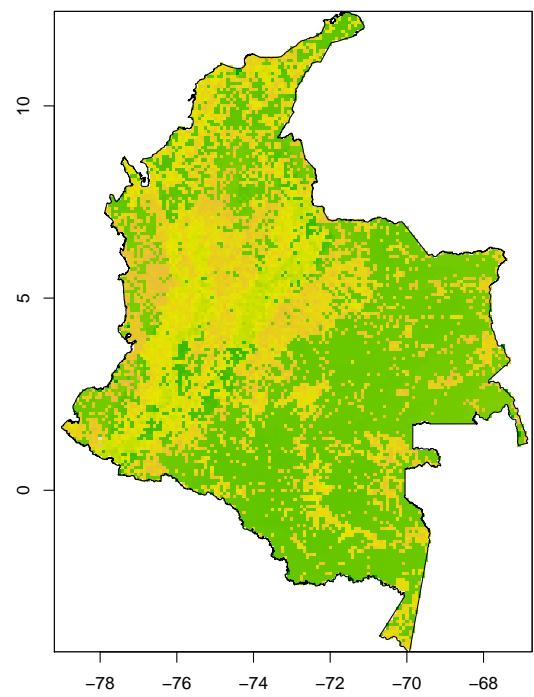
```
#dev.off()

writeRaster(GSI_BOOT,"INDICE_GSI_BOOT.tif", overwrite = TRUE)
writeRaster(GSI_JACK,"INDICE_GSI_JACK.tif", overwrite = TRUE)

par(mfrow = c(1, 2))
plot(GSI_BOOT, main = 'GAP SELCTION INDEX (BOOT)', zlim = c(0, 1))
plot(colombia, add = TRUE)

plot(GSI_JACK, main = 'GAP SELCTION INDEX (JACK)', zlim = c(0, 1))
plot(colombia, add = TRUE)
```

GAP SELECTION INDEX (BOOT)



GAP SELECTION INDEX (JACK)

