

Procesamiento de series de tiempo en GRASS GIS

Aplicaciones en Ecología y Ambiente

Dra. Veronica Andreo
CONICET - INMeT

Rio Cuarto, 2018

GRASS and R: Predicting species distribution



Overview

Overview

- Importing species records

Overview

- Importing species records
- Creating random background points

Overview

- Importing species records
- Creating random background points
- Creating environmental layers

Overview

- Importing species records
- Creating random background points
- Creating environmental layers
- Reading data into R

Overview

- Importing species records
- Creating random background points
- Creating environmental layers
- Reading data into R
- Model species distribution

Overview

- Importing species records
- Creating random background points
- Creating environmental layers
- Reading data into R
- Model species distribution
- Model evaluation and visualization

Data

- Records of *Aedes albopictus* (Asian tiger mosquito) in NC
- Environmental layers derived from RS products



Download the **GRASS code** and **R code** to follow this exercise

Importing species records

```
#!/bin/bash
#####
# Commands for GRASS - R interface exercise:
# Modelling Aedes albopictus potential distribution in NC
# Author: Veronica Andreo
# Date: October, 2018
#####

#
# Download data from GBIF for Aedes albopictus
#

# install extension (requires pygbif: pip install pygbif)
g.extension extension=v.in.pygbif

# set region
```

Importing species records

```
#####  
  
#  
# Download data from GBIF for Aedes albopictus  
#  
  
# install extension (requires pygbif: pip install pygbif)  
g.extension extension=v.in.pygbif  
  
# set region  
g.region vector=nc_state align=MOD11B3.A2015001.h11v05.single_LST_Day_6  
r.mask vector=nc_state  
  
# import data from gbif  
v.in.pygbif output=aedes_albopictus taxa="Aedes albopictus" \  
date_from="2013-01-01" date_to="2018-09-30"
```

Install v.in.pygbif

Importing species records

```
#  
# Download data from GBIF for Aedes albopictus  
#  
  
# install extension (requires pygbif: pip install pygbif)  
g.extension extension=v.in.pygbif  
  
# set region  
g.region vector=nc_state align=MOD11B3.A2015001.h11v05.single_LST_Day_6  
r.mask vector=nc_state  
  
# import data from gbif  
v.in.pygbif output=aedes_albopictus taxa="Aedes albopictus" \  
  date_from="2013-01-01" date_to="2018-09-30"  
  
# clip to NC state  
v.clip input=aedes_albopictus clip=nc_state \  
  output=aedes_albopictus_clip
```

Set region and MASK

Importing species records

```
# install extension (requires pygbif: pip install pygbif)
g.extension extension=v.in.pygbif

# set region
g.region vector=nc_state align=MOD11B3.A2015001.h11v05.single_LST_Day_6
r.mask vector=nc_state

# import data from gbif
v.in.pygbif output=aedes_albopictus taxa="Aedes albopictus" \
  date_from="2013-01-01" date_to="2018-09-30"

# clip to NC state
v.clip input=aedes_albopictus clip=nc_state \
  output=aedes_albopictus_clip

#
# Create background points
```

Import data from GBIF

Importing species records

```
# set region
g.region vector=nc_state align=MOD11B3.A2015001.h11v05.single_LST_Day_6
r.mask vector=nc_state

# import data from gbif
v.in.pygbif output=aedes_albopictus taxa="Aedes albopictus" \
  date_from="2013-01-01" date_to="2018-09-30"

# clip to NC state
v.clip input=aedes_albopictus clip=nc_state \
  output=aedes_albopictus_clip

#
# Create background points
#

# create buffer around Aedes albopictus records
```

Clip to NC state

***Task: Explore univariate statistics of downloaded data.
Check the `d.vect.colhist` and `d.vect.colbp` addons.***

Creating random background points

```
#!/bin/bash
#####
# Commands for GRASS - R interface exercise:
# Modelling Aedes albopictus potential distribution in NC
# Author: Veronica Andreo
# Date: October, 2018
#####

#
# Download data from GBIF for Aedes albopictus
#

# install extension (requires pygbif: pip install pygbif)
g.extension extension=v.in.pygbif

# set region
```

Creating random background points

```
output=aedes_albopictus_clip

#
# Create background points
#

# create buffer around Aedes albopictus records
v.buffer input=aedes_albopictus_clip \
  output=aedes_buffer distance=3000

# generate random points
v.random output=background_points npoints=100 \
  restrict=nc_state seed=4836

#
# Generate environmental variables
```

Create buffer around Aedes albopictus records

Creating random background points

```
# Create background points
#

# create buffer around Aedes albopictus records
v.buffer input=aedes_albopictus_clip \
  output=aedes_buffer distance=3000

# generate random points
v.random output=background_points npoints=100 \
  restrict=nc_state seed=4836

#
# Generate environmental variables
#

# add modis_lst and modis_ndvi to path in user1.marsdat
```

Generate random points

Task: Display with different colors the GBIF records, the buffer areas and the random points.

Creating environmental layers

```
#!/bin/bash
#####
# Commands for GRASS - R interface exercise:
# Modelling Aedes albopictus potential distribution in NC
# Author: Veronica Andreo
# Date: October, 2018
#####

#
# Download data from GBIF for Aedes albopictus
#

# install extension (requires pygbif: pip install pygbif)
g.extension extension=v.in.pygbif

# set region
```

Creating environmental layers

```
random_output_background_points npoints=100 \
restrict=nc_state seed=4836

#
# Generate environmental variables
#

# add modis_lst and modis_ndvi to path in user1 mapset
g.mapsets mapset=modis_lst,modis_ndvi operation=add

# average LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
output=LST_average --o

# minimum LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=minimum \
output=LST_minimum --o
```

Add modis_lst and modis_ndvi to path in user1 mapset

Creating environmental layers

```
#  
# Generate environmental variables  
#  
  
# add modis_lst and modis_ndvi to path in user1 mapset  
g.mapsets mapset=modis_lst,modis_ndvi operation=add  
  
# average LST  
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \  
  output=LST_average --o  
  
# minimum LST  
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=minimum \  
  output=LST_minimum --o  
  
# average LST of summer  
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \  
  output=LST_summer --o
```

Average LST

Creating environmental layers

```
# add modis_lst and modis_ndvi to path in user1 mapset
g.mapsets mapset=modis_lst,modis_ndvi operation=add

# average LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  output=LST_average --o

# minimum LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=minimum \
  output=LST_minimum --o

# average LST of summer
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='07' OR strftime('%m', start_time)='08' OR strftime('%m', start_time)='09'" \
  output=LST_average_sum --o

# average LST of winter
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='12' OR strftime('%m', start_time)='01' OR strftime('%m', start_time)='02'" \
  output=LST_average_winter --o
```

Minimum LST

Creating environmental layers

```
# average LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  output=LST_average --o

# minimum LST
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=minimum \
  output=LST_minimum --o

# average LST of summer
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='07' OR strftime('%m', start_time)='08' OR strftime('%m', start_time)='09' \
  output=LST_average_sum --o

# average LST of winter
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='12' OR strftime('%m', start_time)='01' OR strftime('%m', start_time)='02' \
  output=LST_average_win --o

# average NDVI
```

Average LST of summer

Creating environmental layers

```
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=minimum \
output=LST_minimum --o

# average LST of summer
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
where="strftime('%m', start_time)='07' OR strftime('%m', start_time)='08' OR strftime('%m', start_time)='09' \
output=LST_average_sum --o

# average LST of winter
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
where="strftime('%m', start_time)='12' OR strftime('%m', start_time)='01' OR strftime('%m', start_time)='02' \
output=LST_average_win --o

# average NDVI
t.rast.series input=ndvi_monthly_patch@modis_ndvi method=average \
output=ndvi_average --o

# average NDWI
t.rast.series input=ndwi_monthly@modis_ndvi method=average \
output=ndwi_average --o
```

Average LST of winter

Creating environmental layers

```
# average LST of summer
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='07' OR strftime('%m', start_time)='08' OR strftime('%m', start_time)='09' \
  output=LST_average_sum --o

# average LST of winter
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='12' OR strftime('%m', start_time)='01' OR strftime('%m', start_time)='02' \
  output=LST_average_win --o

# average NDVI
t.rast.series input=ndvi_monthly_patch@modis_ndvi method=average \
  output=ndvi_average --o

# average NDWI
t.rast.series input=ndwi_monthly@modis_ndvi method=average \
  output=ndwi_average --o
```

Average NDVI

Creating environmental layers

```
# average LST of summer
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='07' OR strftime('%m', start_time)='08' OR strftime('%m', start_time)='09' \
  output=LST_average_sum --o

# average LST of winter
t.rast.series input=LST_Day_monthly_celsius@modis_lst method=average \
  where="strftime('%m', start_time)='12' OR strftime('%m', start_time)='01' OR strftime('%m', start_time)='02' \
  output=LST_average_win --o

# average NDVI
t.rast.series input=ndvi_monthly_patch@modis_ndvi method=average \
  output=ndvi_average --o

# average NDWI
t.rast.series input=ndwi_monthly@modis_ndvi method=average \
  output=ndwi_average --o
```

Average NDWI

Task: Which other variable could we generate/use?

Just for fun, close GRASS GIS, we'll initialize it again
but from RStudio

Install and load packages

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages  
# install.packages("raster")  
# install.packages("rgrass7")
```

Install and load packages

```
#  
# Install and load required packages  
#  
  
# install packages  
# install.packages("raster")  
# install.packages("rgrass7")  
# install.packages("mapview")  
# install.packages("biomod2")  
  
# load libraries  
library(raster)  
library(rgrass7)  
library(mapview)  
library(biomod2)
```

Install packages

Install and load packages

```
# install.packages
# install.packages("raster")
# install.packages("rgrass7")
# install.packages("mapview")
# install.packages("biomod2")

# load libraries
library(raster)
library(rgrass7)
library(mapview)
library(biomod2)

#
# Set GRASS GIS variables for initialization
#
```

Load packages

Initialize GRASS GIS

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages  
# install.packages("raster")  
# install.packages("rgrass7")
```

Initialize GRASS GIS

```
#  
# Set GRASS GIS variables for initialization  
#  
  
# path to GRASS binaries  
myGRASS <- "/usr/local/grass74"  
# path to GRASS database  
myGISDbase <- "/home/veroandreo/grassdata/"  
# path to location  
myLocation <- "nc_spm_08_grass7"  
# path to mapset  
myMapset <- "user1"  
  
# start GRASS GIS from R  
initGRASS(gisBase = myGRASS,  
          home = tempdir(),  
          gisDbase = myGISDbase,
```

Set parameters to start GRASS

Initialize GRASS GIS

```
myGISbase = "/home/verbanas/grassdata/  
# path to location  
myLocation <- "nc_spm_08_grass7"  
# path to mapset  
myMapset <- "user1"  
  
# start GRASS GIS from R  
initGRASS(gisBase = myGRASS,  
          home = tempdir(),  
          gisDbase = myGISDbase,  
          location = myLocation,  
          mapset = myMapset,  
          SG="elevation",  
          override = TRUE)  
  
#  
# Read raster and vector data  
#
```

Initialize GRASS GIS

Read vector and raster data

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages("raster")  
# install.packages("rgrass7")
```

Read vector and raster data

```
override = TRUE)  
  
#  
# Read raster and vector data  
#  
  
# read vector layers  
Aa_pres <- readVECT("aedes_albopictus_clip")  
Aa_abs <- readVECT("background_points")  
  
# read raster layers  
LST_mean <- readRAST("LST_average")  
LST_min <- readRAST("LST_minimum")  
LST_mean_summer <- readRAST("LST_average_sum")  
LST_mean_winter <- readRAST("LST_average_win")  
NDVI_mean <- readRAST("ndvi_average")  
NDVI_min <- readRAST("ndvi_minimum")
```

Read vector data

Read vector and raster data

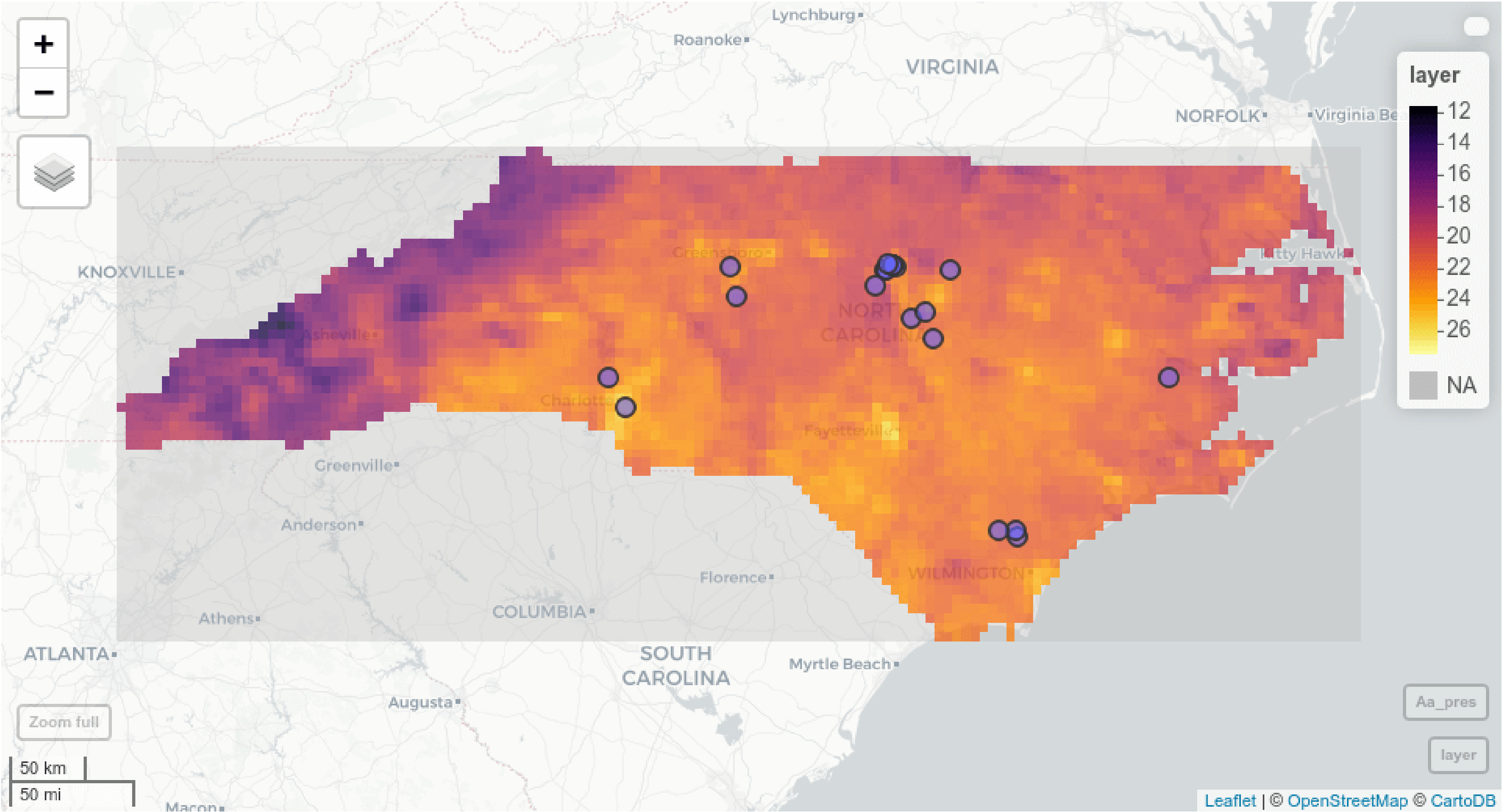
```
# read vector layers
Aa_pres <- readVECT("aedes_albopictus_clip")
Aa_abs <- readVECT("background_points")

# read raster layers
LST_mean <- readRAST("LST_average")
LST_min <- readRAST("LST_minimum")
LST_mean_summer <- readRAST("LST_average_sum")
LST_mean_winter <- readRAST("LST_average_win")
NDVI_mean <- readRAST("ndvi_average")
NDWI_mean <- readRAST("ndwi_average")

# visualize in mapview
mapview(NDVI_mean)
mapview(LST_mean) + Aa_pres
```

Read raster data

Task: display maps and points in RStudio using sp or sf and mapview. Explore the manual of each function.



Data formatting

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages("raster")  
# install.packages("rgrass7")
```

Data formatting

```
# data preparation and formatting
#

# response variable
n_pres <- length(Aa_pres[,1])
n_abs <- length(Aa_abs@data[,1])

myRespName <- 'Aedes_albopictus'

pres <- rep(1, n_pres)
abs <- rep(0, n_abs)
myResp <- c(pres,abs)

myRespXY <- rbind(cbind(Aa_pres@coords[,1],Aa_pres@coords[,2]),
                  cbind(Aa_abs@coords[,1],Aa_abs@coords[,2]))

# explanatory variables
myExp1 <- stack(raster(LST_mean), raster(LST_min))
```

Response variable

Data formatting

```
myResp <- c(pres,abs)

myRespXY <- rbind(cbind(Aa_pres@coords[,1],Aa_pres@coords[,2]),
                 cbind(Aa_abs@coords[,1],Aa_abs@coords[,2]))

# explanatory variables
myExpl <- stack(raster(LST_mean),raster(LST_min),
               raster(LST_mean_summer),raster(LST_mean_winter),
               raster(NDVI_mean),raster(NDWI_mean))

myBiomodData <- BIOMOD_FormatingData(resp.var = myResp,
                                     expl.var = myExpl,
                                     resp.xy = myRespXY,
                                     resp.name = myRespName)

#
# Run model
```

Explanatory variables

Run Random Forest model

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages("raster")  
# install.packages("rgrass7")
```

Run Random Forest model

```
resp.xy = myRespXY,  
resp.name = myRespName)  
  
#  
# Run model  
#  
  
# default options  
myBiomodOption <- BIOMOD_ModelingOptions()  
  
# run model  
myBiomodModelOut <- BIOMOD_Modeling(  
  myBiomodData,  
  models = c('RF'), # algoritmos de analisis para hacer el modelo  
  models.options = myBiomodOption,  
  NbRunEval=2,  
  DataSplit=80 # porcentaje de los datos para evaluación
```

Default options

Run Random Forest model

```
# default options
myBiomodOption <- BIOMOD_ModelingOptions()

# run model
myBiomodModelOut <- BIOMOD_Modeling(
  myBiomodData,
  models = c('RF'), # algoritmos de analisis para hacer el modelo
  models.options = myBiomodOption,
  NbRunEval=2,
  DataSplit=80, # porcentaje de los datos para evaluación
  Prevalence=0.5,
  VarImport=3,
  models.eval.meth = c('TSS','ROC'), # metricas para evaluar el modelo
  SaveObj = TRUE,
  rescal.all.models = TRUE,
  do.full.models = FALSE,
  modeling.id = paste(myRespName,"FirstModeling",sep=""))
```

myBiomodModelOut

Run model

Task: Explore the model output

Model evaluation

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages("raster")  
# install.packages("rgrass7")
```

Model evaluation

```
myBiomodModelOut

#
# Model evaluation
#

# extract all evaluation data
myBiomodModelEval <- get_evaluations(myBiomodModelOut)

# TSS: True Skill Statistics
myBiomodModelEval["TSS","Testing.data","RF",,]

# ROC: Receiver-operator curve
myBiomodModelEval["ROC","Testing.data",,,]
```

Extract all evaluation data

Model evaluation

```
#  
# Model evaluation  
#  
  
# extract all evaluation data  
myBiomodModelEval <- get_evaluations(myBiomodModelOut)  
  
# TSS: True Skill Statistics  
myBiomodModelEval["TSS", "Testing.data", "RF", ,]  
  
# ROC: Receiver-operator curve  
myBiomodModelEval["ROC", "Testing.data", , ,]  
  
# variable importance  
get_variables_importance(myBiomodModelOut)
```

TSS: True Skill Statistics

Model evaluation

```
#  
  
# extract all evaluation data  
myBiomodModelEval <- get_evaluations(myBiomodModelOut)  
  
# TSS: True Skill Statistics  
myBiomodModelEval["TSS","Testing.data","RF",,]  
  
# ROC: Receiver-operator curve  
myBiomodModelEval["ROC","Testing.data",,,]  
  
# variable importance  
get_variables_importance(myBiomodModelOut)  
  
#  
# Model predictions  
"
```

ROC: Receiver-operator curve



```
#
# Model predictions
#
```

Variable importance

Model predictions

```
#####  
# Commands for GRASS - R interface exercise:  
# Modelling Aedes albopictus potential distribution in NC  
#  
# Original example contributed by Carol Garzon Lopez  
# Adapted by Veronica Andreo  
# Date: October, 2018  
#####  
  
#  
# Install and load required packages  
#  
  
# install.packages  
# install.packages("raster")  
# install.packages("rgrass7")
```

Model predictions

```
#  
# Model predictions  
#  
  
myBiomodProj <- BIOMOD_Projection(  
  modeling.output = myBiomodModelOut,  
  new.env = myExpl,  
  proj.name = "current",  
  selected.models = "all",  
  compress = FALSE,  
  build.clamping.mask = FALSE)  
  
mod_proj <- get_predictions(myBiomodProj)  
mod_proj  
  
# plot predicted potential distribution
```

Model projection settings

Model predictions

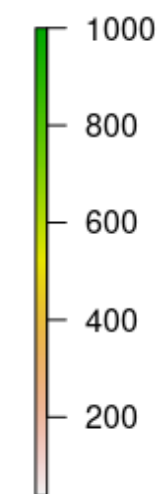
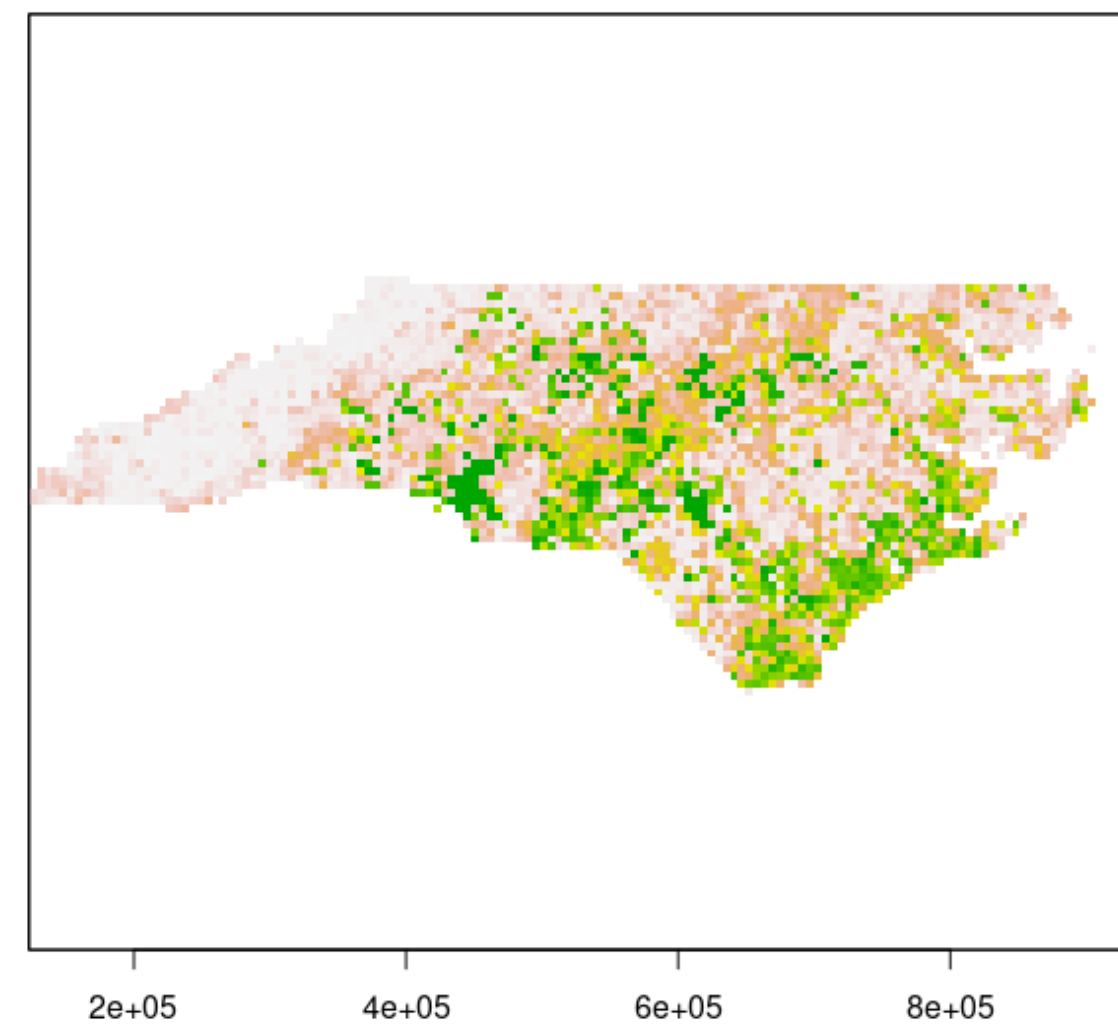
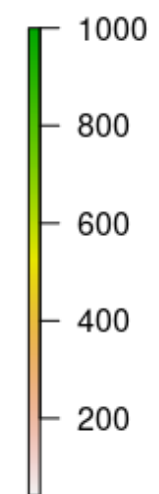
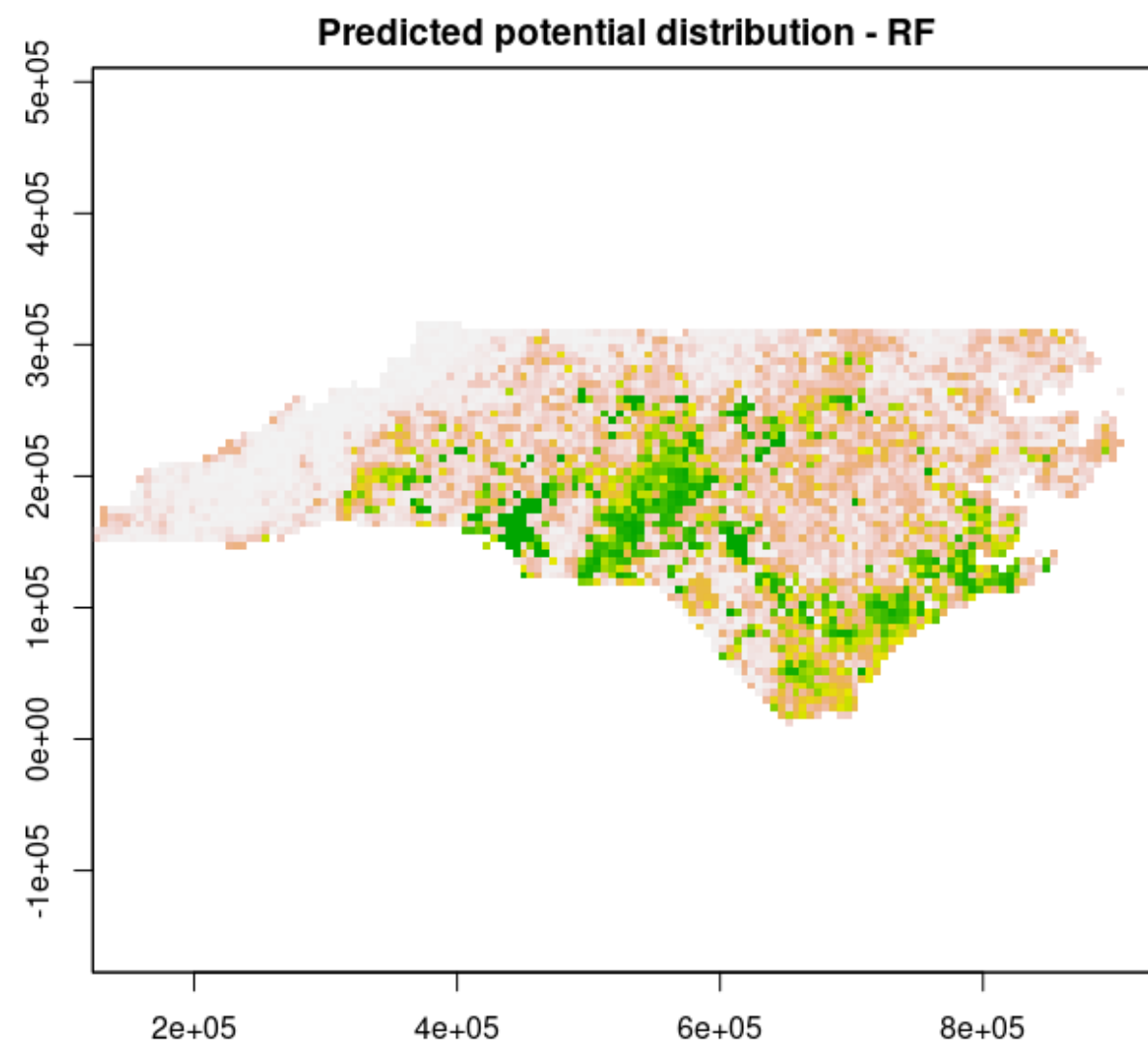
```
#  
# Model predictions  
#  
  
myBiomodProj <- BIOMOD_Projection(  
  modeling.output = myBiomodModelOut,  
  new.env = myExpl,  
  proj.name = "current",  
  selected.models = "all",  
  compress = FALSE,  
  build.clamping.mask = FALSE)  
  
mod_proj <- get_predictions(myBiomodProj)  
mod_proj  
  
# plot predicted potential distribution  
plot(mod_proj, main = "Predicted potential distribution - RF")
```

Obtain predictions from model

Model predictions

```
#  
# Model predictions  
#  
  
myBiomodProj <- BIOMOD_Projection(  
  modeling.output = myBiomodModelOut,  
  new.env = myExpl,  
  proj.name = "current",  
  selected.models = "all",  
  compress = FALSE,  
  build.clamping.mask = FALSE)  
  
mod_proj <- get_predictions(myBiomodProj)  
mod_proj  
  
# plot predicted potential distribution  
plot(mod_proj, main = "Predicted potential distribution - RF")
```

Plot predicted potential distribution



***Task: Explore algorithms available in
BIOMOD_Modeling() and try with a different one.
Compare the results.***

QUESTIONS?



Thanks for your attention!!



Move on to:

Think on the evaluation exercise

Presentation powered by

