R code
Introduction to remote sensing and GIS for ecological applications

Duccio Rocchini

May 21, 2018

```
###-- Functions:
https://github.com/bleutner/RStoolbox/tree/master/R

###-- Data:
http://book.ecosens.org/software/rstoolbox/

###-- Landsat bands
https://landsat.usgs.gov/what-are-band-designations-landsat-
    satellites

###-- Code

install.packages("raster") #------ do the same for all packages to
    be recalled later on by library()

library(raster)
library(rgdal)
#  Geospatial Data Abstraction Library
library(RStoolbox)
library(ggplot2)
library(hexbin)

# 2011 image
p224r063_2011m <- brick("~/grassdata/data_book/raster_data/final/
    p224r63_2011_masked.grd")

p224r063_2011m

summary(p224r063_2011m)

# B1_sre      B2_sre      B3_sre     B4_sre     B5_sre B6_bt      B7_
    sre
# Min.     0.00000000 0.01154835 0.00730000 0.0000000 0.0000000
    295.1 0.00000000
# 1st Qu. 0.01371136 0.03108290 0.02000000 0.2516721 0.1137448
    296.4 0.04183780
# Median   0.01654248 0.03526897 0.02386578 0.2842704 0.1290826
    296.9 0.04887195
# 3rd Qu. 0.02157205 0.04421049 0.03364492 0.3121567 0.1567999
    298.1 0.06530904
# Max.     0.10590097 0.14880101 0.24380812 0.5219069 0.3955572
    304.4 0.31459978
# NAs     0.00000000 0.00000000 0.00000000 0.0000000 0.0000000
    0.0 0.00000000

###-- Multitemporal analysis

p224r063_1988m <- brick("~/grassdata/data_book/raster_data/final/
    p224r63_1988_masked.grd")
```

```
# explore the data
pairs(p224r063_2011m)
# wait!

##— RGB spaces
plotRGB(p224r063_2011m, r=3,g=2,b=1,scale=1000,stretch="Lin")
plotRGB(p224r063_2011m, r=4,g=3,b=2,scale=1000,stretch="Lin")

library(ggplot2)
ggRGB(p224r063_2011m,4,3,2)

par(mfrow=c(2,1))
plotRGB(p224r063_2011m, r=3,g=2,b=1,scale=1000,stretch="Lin",main=
    "natural colours")
plotRGB(p224r063_2011m, r=4,g=3,b=2,scale=1000,stretch="Lin", main
    ="infrared")

multitemp <-  p224r063_2011m - p224r063_1988m

plot(multitemp)

###— Spectral indices

ndvi2011 <- (p224r063_2011m$B4_sre-p224r063_2011m$B3_sre) / (
    p224r063_2011m$B4_sre+p224r063_2011m$B3_sre)
# ~time: 1min

ndvi1988 <- (p224r063_1988m$B4_sre-p224r063_1988m$B3_sre) / (
    p224r063_1988m$B4_sre+p224r063_1988m$B3_sre)
# ~time: 1min

par(mfrow=c(2,1))
plot(ndvi1988,main="NDVI 1988")
plot(ndvi2011,main="NDVI 2011")

dif <- ndvi2011-ndvi1988

# par
par(mfrow=c(1,2))
hist(ndvi1988,ylim=c(0,2000000))
hist(ndvi2011,ylim=c(0,2000000))

# Colored histograms (blue and red)
hist(ndvi1988, col=rgb(1,0,0,0.5),main="NDVI frequencies")
hist(ndvi2011, col=rgb(0,0,1,0.5), add=T)
box()
legend("topleft", c("NDVI 1988", "NDVI 2011"), fill=c("red", "
    blue"))
```

```
### Hexagon binning
library(hexbin)
hbin <- hexbin(ndvi1988,ndvi2011, xbins = 40)
plot(hbin)

### PCA

pairs(p224r063_2011m)

library(RStoolbox)

manner <- aggregate(p224r063_2011m, fact=20)

p224r063_2011m_res <- resample(p224r063_2011m, manner)

p224r063_2011m_pca <- rasterPCA(p224r063_2011m_res)
# 3 minutes needed

summary(p224r063_2011m_pca$model)

# Importance of components:
#                             Comp.1       Comp.2       Comp.3
    Comp.4
# Standard deviation     1.2950291 0.052987610 0.0213916820
   5.551811e-03
# Proportion of Variance 0.9980317 0.001670837 0.0002723173
   1.834234e-05
# Cumulative Proportion  0.9980317 0.999702523 0.9999748401
   9.999932e-01
#                             Comp.5       Comp.6       Comp.7
# Standard deviation     2.621003e-03 1.710617e-03 1.288550e-03
# Proportion of Variance 4.088090e-06 1.741370e-06 9.880696e-07
# Cumulative Proportion  9.999973e-01 9.999990e-01 1.000000e+00


loadings(p224r063_2011m_pca$model)

# Loadings:
#        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
# B1_sre                 0.116  0.317         0.604 -0.721
# B2_sre                 0.161  0.684 -0.276  0.294  0.581
# B3_sre                 0.274  0.559  0.194 -0.697 -0.296
# B4_sre         -0.892 -0.432         0.115
# B5_sre         -0.412  0.663 -0.318 -0.526
# B6_bt   -0.999
# B7_sre         -0.162  0.508 -0.117  0.772  0.238  0.223
#
#                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
```

```
# SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000
# Proportion Var  0.143  0.143  0.143  0.143  0.143  0.143  0.143
# Cumulative Var  0.143  0.286  0.429  0.571  0.714  0.857  1.000

p224r063_2011m_pca

# $call
# rasterPCA(img = p224r063_2011m_res)
#
# $model
# Call:
# princomp(cor = spca, covmat = covMat[[1]])
#
# Standard deviations:
#     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6
# 1.295029050 0.052987610 0.021391682 0.005551811 0.002621003 0.001710617
#     Comp.7
# 0.001288550
#
#  7  variables and  494500 observations.
#
# $map
# class       : RasterBrick
# dimensions  : 500, 989, 494500, 7  (nrow, ncol, ncell, nlayers)
# resolution  : 90, 90  (x, y)
# extent      : 579765, 668775, -522735, -477735  (xmin, xmax, ymin, ymax)
# coord. ref. : +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
# data source : in memory
# names       :        PC1,        PC2,        PC3,        PC4,        PC5,        PC6,        PC7
# min values  : -7.29596861, -0.21213869, -0.12206057, -0.07208760, -0.02024087, -0.03419134, -0.01445724
# max values  :  2.40711155,  0.33669578,  0.16446461,  0.09348111,  0.05662598,  0.02284201,  0.04294400
#
#
# attr(,"class")
# [1] "rasterPCA" "RStoolbox"

##-- Plotting the PCA map
plot(p224r063_2011m_pca$map)

plotRGB(p224r063_2011m_pca$map, r=1,g=2,b=3,scale=1000,stretch="Lin")
```

```
library(ggplot2)
ggRGB(p224r063_2011m_pca$map,1,2,3, stretch="lin")

#--- plot components
plot(p224r063_2011m_pca$map$PC1,p224r063_2011m_pca$map$PC2)

###-- Mapping diversity

# Preparing the dataset
# agg <- aggregate(ndvi2011,fact=50)
# ndvi2011_res <- resample(ndvi2011,agg)
#
# # variance
# varndvi <- focal(ndvi2011_res, w=matrix(1/9,nrow=3,ncol=3), fun
    =var)
# # wait!
# # try the same with GRASS GIS
#
# plot(varndvi)

##-- Shannon's and Rao's Q diversity

require(raster)

source("~/Documents/lectures_and_seminars/glasgow_2018/code/
    spectralrao.r")

# Random simulated spectral matrix
set.seed(26)

r1<-matrix(rpois(2500,lambda=5),nrow=50,ncol=50)

# Plot raster
plot(raster(r1))

# Run the Function
raomatrix<-spectralrao(r1,distance_m="euclidean",window=3,shannon
    =TRUE)

# Comparison
par(mfrow=c(1,3))
plot(raster(r1),main="Synthetic matrix")
plot(raster(raomatrix[[2]]),main="Shannon's Index")
plot(raster(raomatrix[[1]]),main="Rao's Q")

# Real set
# ndvi2011_res
raomatrix_ndvi <- spectralrao(ndvi2011_res,distance_m="euclidean"
```

```
    ,window=3,shannon=TRUE)

# Plot results with a real set
par(mfrow=c(1,3))
plot(ndvi2011_res,main="Synthetic matrix")
plot(raster(raomatrix_ndvi [[2]]),main="Shannon's Index")
plot(raster(raomatrix_ndvi [[1]]),main="Rao's Q")


###-- Field sampling data

require(raster)
require(RStoolbox)

##-- import imagery
p224r063_2011 <- brick("~/grassdata/data_book/raster_data/final/
    p224r63_2011.grd")

##-- define the study area
study <- shapefile("~/grassdata/data_book/vector_data/study_area_
    UTMnorth.shp")

##-- generate random points
RandomPoints <- spsample(study,n=100,type="random")

##-- plot
plotRGB(p224r063_2011, 4,3,2,scale=1000,stretch="lin")

##-- with buffer
points(RandomPoints,cex=2,pch=20,col="red")
points(RandomPoints,cex=20,col="red")

##-- density map: how much dense is your sampling design?
RandomPoints_distmap <- distanceFromPoints(p224r063_2011,
    RandomPoints)

plot(RandomPoints_distmap)
points(RandomPoints,cex=2,pch=20,col="red")

# distance
# spatialdist <- pointDistance(RandomPoints,lonlat=F)

# info on coordinates
coordinates(RandomPoints)

##-- generate regular points
RandomPointsreg <- spsample(study,n=100,type="regular")

##-- with buffer
```

```
points(RandomPointsreg,cex=2,pch=20,col="green")
points(RandomPointsreg,cex=20,col="green")

# Spatial points
# require(rgdal)
# ranPoilsat.sp <- SpatialPoints(RandomPoints)

# a sampling onto an image
# install.packages("dismo")
# require(dismo)
#
# ranPoi_lsat <- randomPoints(p224r063_2011m,100)

##-- extract data
ptVals <- extract(p224r063_2011,y=RandomPoints)

##-- only 4 bands e.g.
pa_values<-extract(p224r063_2011[[1:4]],y=RandomPoints)

##-- dataframe
pa_values_data <- as.data.frame(pa_values)

##-- explore the set
pairs(pa_values_data)
attach(pa_values_data)

boxplot(pa_values_data)

# scatterplot matrix
install.packages("car")
#wait: make a break
require(car)
scatterplotMatrix(pa_values_data)

##-- Detrended Correspondence Analysis
pa_values_data.dca <- decorana(pa_values_data)
pa_values_data.dca
summary(pa_values_data.dca)
plot(pa_values_data.dca)

##-- External samples

field_sampling <- read.csv(file="~/grassdata/data_book/vector_
    data/csv_file_locationdata.csv")

spdf <- SpatialPointsDataFrame(field_sampling[,1:2],
          field_sampling)

##-- Ecological distance decay
```

```
coor <- coordinates(RandomPoints)

require(vegan)

spatialdist <- vegdist(coor,method="euclidean")
spectraldist <- vegdist(pa_values_data,method="bray")

spectralsim <- 1-spectraldist

plot(spatialdist,spectralsim,col="red")

#- Hexagon binning
library(hexbin)
hbin <- hexbin(spatialdist,1-spectraldist, xbins = 40)
plot(hbin)

#- Quantile regression
install.packages("quantreg")

require(quantreg)

plot(spatialdist,spectralsim,col="red")

quant90 <- rq(spectralsim ~ spatialdist, tau=0.90)
abline(quant90,col="blue")

quant10 <- rq(spectralsim ~ spatialdist, tau=0.10)
abline(quant10,col="blue")

mod <- lm(spectralsim~spatialdist)
abline(mod)
```