

Introducción

1. Un viaje del sol a los píxeles.

En esta primera práctica nos familiarizaremos con las interfaces gráficas del qgis y de R-studio. Para esto comenzaremos a analizar la imagen correspondiente a la zona de estudio del año 2015 desde el punto de vista espectral. Son nuestros objetivos

- Poder cargar una imagen en qgis.
- Digitalizar coberturas en qgis.
- Poder cargar un archivo raster y uno vectorial en R.
- Realizar un análisis estadístico de la imagen como un todo y de las distintas coberturas digitalizadas en R.

1.1. Exploración con el qgis

Comenzamos abriendo la imagen 18 correspondiente a la zona de interés durante el mes de mayo de 2015. Para esto vamos al menú $\rightarrow \rightarrow$. Navegamos hasta la carpeta `raster_data/LC82300772015071` y abrimos el archivo `LC82300772015071.brk`. Una vez abierto el mismo podremos encontrarlo en el `de q-gis` donde podremos manejar la visualización del mismo y estudiar las propiedades de dicha capa.

Usando la barra de herramientas del qgis podremos movernos en la imagen, realizar zoom en la misma e incluso otros.

Actividad 1.1. Cambie la combinación de bandas de la imagen L8 y muévase dentro de la misma.

Actividad 1.2. Encuentre el sistema de coordenadas en el cual se encuentra la imagen.

Actividad 1.3. Utilizando la herramienta identificar objetos espaciales encuentre los valores de reflectancia de distintas coberturas. Grafique estos valores en una firma espectral y en el espacio de fases nirrojo.

Con la herramienta nueva capa de archivo shape es posible digitalizar zonas de la imagen para su posterior análisis. Para esto puede hacer click en el botón del panel lateral y agregar una capa nueva para lo cual el qgis pedirá un nombre de la misma. Guárdela en la carpeta `vector_data/` con el nombre `firmas.shp`. Recuerde elegir el sistema de coordenadas correspondiente a la imagen anterior.

Una vez creada la nueva capa podemos utilizar la barra de herramientas de qgis para agregar nuevas geometrías a la misma. Para esto hacemos click en el botón de agregar geométrica y digitalizamos una zona uniforme dentro de la imagen. Al terminar de hacerlo qgis pedirá un número de ID para la capa que debe ser correlativo.

Actividad 1.4. Digitalice coberturas uniformes dentro de la imagen. Recuerde obtener al menos una por cada categoría de uso y cobertura presente dentro de la misma.

Actividad 1.5. Utilizando la herramienta de tabla de datos de la imagen modifique la tabla de datos para incluir la categoría de la cobertura, la subcategoría y una pequeña descripción.

1.2. Exploración en R

Para trabajar en R deberemos primero cargar las librerías necesarias con el comando

```
1 library(raster)
```

además, deberemos situar nuestra carpeta de trabajo donde se encuentran las carpetas que descargamos. Para esto nos movemos en el explorar de archivos hasta la misma y hacemos click en usar la carpeta como carpeta de trabajo.

Una vez en dicha carpeta, existen varias maneras de abrir una imagen segun queramos hacerlo solo para una banda, varias bandas en archivos separados o un solo archivo multibanda.

Los comandos para esto son **raster**, para abrir una unica banda, **brick**, para abrir un archivo multibanda, y **stack** para abrir distinas bandas por separado. Veamos algunos ejemplo de esto:

Ejemplo 1.1. Abrimos la imagen completa del archivo de landsat 8 y consultamos sus propiedades.

```
1      l8 <- brick(filename)
2      l8
```

Una vez abierta la imagen en el R podemos empezar a trabajar con la misma utilizando distintos comandos.

Veamos primero como cambiar los nombres de las bandas por defecto, cambiar la imagen a numeros en reflectancia entre 0 y 1 y luego guardarla nuevamente. Para eso ejecutamos el siguiente codigo.

```
1      l8 <- brick(filename)
2      names(l8) <- c("blue", "gree", "red", "nir", "swirl", "swir2")
3      l8 <- l8/1e4
4      rasterOptions(addheader = "ENVI")
5      writeRaster(l8\_ref, "raster\_data/processed/")
```

Analicemos el codigo linea por linea. La primera de ellas abre la imagen como un raster de multiples bandas. La segunda, cambia los nombres de cada banda a los que figuran en la lista entre parentesis. Es importante resaltar que el numero de nombres debe ser el mismo que el de bandas. En tercer lugar convertimos el archivo de numeros enteros entre 0 y 10000 a numeros entre 0 y 1. En la cuarta linea incluimos el archivo header que nos permitira levantar nuestra imagen con qgis. Por ultimo, guardamos el archivo en el formato nativo de R.

Actividad 1.6. Abra el archivo vrt en qgis y vuelva a mirar la firma espectral para distintas coberturas. Entre que valores se encuentra ahora las mismas.

Hagamos un poco de analisis ahora sobre la imagen. Para comenzar podemos calcular los histogramas de todas las bandas con el comando

```
1      hist(l8)
```

y el scatter plot entre dos bandas como

```
1      plot(l8$red, l8$blue)
```

en caso de querer todos los scatterplots e histogramas en un solo grafico podemos hacerlo con el comando

```
1      pairs(l8)
```

Hasta ahora estamos analizando la imagen completa. Podemos sin embargo analizar solo sectores concretos de la imagen muestreandola en funcion de un shapefile. Para esto debemos primero abrir el shapefile de interes cargando primero la libreria

```
1 library(rgdal)
```

y luego leyendo el vector como

```
1 vector <- readOGR(dsn="vector\_data/", layer="extract")
```

Podemos mostrar las propiedades del vector ejecutando el comando

```
1 vector
```

Actividad 1.7. Muestre las propiedades de la capa raster y el vector abiertos y verifique que los mismos se encuentren en el mismo sistema de coordenadas.

Podemos mostrar las imagenes dentro de R si así nos interesa. Para esto utilizaremos la libreria `ggplot` como se muestra a continuacion

```
1 ggRGB(l8, r="nir", g="red", b="blue", geom_raster=TRUE)
```

Por ultimo mostremos como extraer datos de un archivo raster y veamos un par de ejemplo concretos. La funcion que nos permite extraer datos de un raster segun un vector es `extract` que toma dos argumentos

```
1 raster(l8, vector)
```

Veamos algunos ejemplos que pueden ser utiles de aplicacion de todo lo anterior

Ejemplo 1.2. Graficar en un scatterplot de dos bandas mostrando la zona del espacio ocupada por una cobertura.

```
1 plot(l8$red, l8$nir)
2 points(as.data.frame(datos[1])$red, as.data.frame(datos[1])$nir,
3 col="green")
```

Ejemplo 1.3. Extraer los promedios y desvios standar de un raster y agregarlos a un vector.

```
1 promedio <- extract(l8, vector, fun=mean)
2 desvio <- extract(l8, vector, fun=sd)
3 colnames(promedio) <- paster("mean", colnames("promedio"), sep="_")
4 colnames(desvio) <- paster("sd", colnames("desvio"), sep="_")
5 vector@data <- cbind(vector@data, promedio, desvio)
6 writeOGR(vector, sdn="vector_data/processed/", datos, driver="ESRI
7 Shapefile")
```

Ejemplo 1.4. Graficar las firmas espectrales en funcion de la longitud de onda para cada geometria de un vector.

```
1 df <- t(promedio)
2 colnames(df) <- vector@data$descripcion
3 df$wl <- as.matrix(c(485,560,660,830,1650,2215))
4 df <- melt(df, id.vars="wl", variable.name="cobertura")
5 names(df) <- c("wl", "Cobertura", "Reflectancia")
6 dfd <- t(desvio)
7 colnames(dfd) <- vector@data$descripcion
8 dfd$wl <- as.matrix(c(485,560,660,830,1650,2215))
```

```

9         dfd <- melt("wl", "Cobertura", "Desvio")
10
11         df$desvio <- dfd$desvio
12
13         ggplot(df, aes(wl, Reflectancia))+
14             geom_line(aes(colour=Cobertura))+
15             geom_poinr(aes(colour=Cobertura))+
16             geom_errorbar(aes(ymin=Reflectancia-2*desvio,
17                               ymax=Reflectancia+2*desvio))

```

Actividad 1.8. Grafique la media y el desvio standar para las distintas coberturas que pudo identificar en el punto uno.

2. Rebotando por la atmosfera

En esta segunda actividad practica nos centraremos en la correccion radiometrica de imagenes satelitales. Son objetivos de la misma

- Poder abrir una imagen satelital desde el metadato.
- Convertir los valores de la imagen a reflectancia tope de la atmosfera.
- Corregir la imagen satelital utilizando los metodos de *dos* y *cost*
- Corregir la imagen satelital utilizando el *6S web*

2.1. Correccion de imagenes en R

Para abrir una imagen satelital desde el metadato utilizaremos las funciones disponibles en **RStoolbox**. Dicho paquete incluye diversas herramientas para trabajar con sensores remotos y ya lo utilizamos antes para graficar imagenes satelitales.

Ejemplo 2.1. Comencemos analizando un ejemplo sencillo, abriremos una imagen landsat 5 desde el metadato y la mostraremos en combinacion de bandas de falso color compuesto, ademas de analizar las propiedades basicas de la misma.

```

1         library(raster)
2         library(RStoolbox)
3         meta.1992 <- readMeta("raster_data/LT52300771992104CUB00/
4                               LT52300771992104CUB00_MTL.txt")
5         dn.1992 <- stackMeta(meta.1992)
6         dn.1992 <- dn.1992[[-6,]]
7         ggRGB(dn.1992,
8               r=4, g=3, b=2,
9               geom_raster = TRUE,
10              stretch = "lin")

```

Cargaremos de esta forma el metadato de la imagen landat 5 del año 1992, abriremos las bandas de la misma y la mostraremos en combinacion de falso color compuesto. Analicemos punto por punto que esta pasando.

1. Las lineas 1 y 2 del script leen las librerias que necesitamos para trabajar con la imagen.
2. La linea 3 crea la variable `meta.1992` con los metadatos correspondientes a la imagen de interes.
3. La linea 4 crea la variable `dn.1992` con las bandas estaqueadas para poder utilizarlas en R. Podemos inspeccionar el elemento poniendo su nombre en la consola.

4. La linea 5 elimina la banda termica de nuestra imagen.
5. Las lineas 6 a 9 nos permiten mostrar la imagen en combinacion falso color compuesto. De ella la linea 6 se refiere a la imagen a mostrar, la linea 7 a la combinacion de colores elegida y la 9 al tipo de realce aplicado.

De esta forma podemos tener el archivo cargado en DN con todos sus metadatos para convertirlo a reflectancia. Para pasar nuestra imagen a reflectancia a tope de la atmosfera tenemos dos maneras de hacerlo. Podemos hacerlo a mano utilizando las herramientas algebraicas de R o podemos hacerlo con la funcion especifica de **RStoolbox**.

Veamos ambas. A mano

```
1 calref.1992 <- meta.1992$calref
2 elev.1992 <- pi*meta.1992$SOLAR_PARAMETERS['elevation']/180
3 dn2ref.1992 <- meta.1992$CALREF
4 toa.1992 <- (dn.1992*dn2ref.1992$gain+dn2ref.1992$offset)/sin(elev.1992)
5 names(toa.1992) <- c("B1-toa","B2-toa","B3-toa","B4-toa","B5-toa","B7-toa")
```

de forma automatica

```
1 toa.1992b <- radCor(dn.1992, metaData = meta.1992, method = "apref")
```

podemos comparar los resultados de ambos metodos inspeccionando los objetos.

Actividad 2.1. Inspeccione la reflectancia a tope de la atmosfera para todas las bandas. Para esto realice los histogramas, graficos de dispersion, calcule la media, el desvio standar y cualquier otra medida estadistica que le guste.

La funcion **radCor** dispone distintos parametros para hacer distintos tipos de correcciones atmosfericas. Ya vimos *apref* que nos permitio calcular la reflectancia a tope de la atmosfera. Veamos como aplicar el metodo de substraccion de cuerpo oscuro.

```
1 haze.1992 <- estimateHaze(dn.1992,darkProp = 0.01, hazeBands = 1:4, plot=TRUE)
2 sdos.1992 <- radCor(dn.1992, metaData = meta.1992,
3 hazeValues = haze.1992,
4 hazeBands = c("B1-dn","B2-dn","B3-dn","B4-dn"),
5 method="sdos")
```

Actividad 2.2. Analice los valores de haze obtenidos por la funcion *estimate haze* y en caso de que sea necesario, corrijalos para la banda indicada.

Actividad 2.3. Utilice el metodo *costz* para corregir la imagen a reflectancia a tope de la superficie.

Actividad 2.4. Guarde los archivos raster generado por cada uno de los metodos de correccion. Abralos en qgis y comparelos visualmente.

2.2. 6S

Veamos ahora como operar con el 6S para obtener una estimacion de los parametros atmosfericos. Para esto utilizaremos la version web del 6S que se encuentra disponible en <http://6s.ltdri.org/pages/run6SV.html>.

Para utilizarla ingresaremos a la pagina y haremos click en el boton . Iremos luego configurando paso a paso nuestro modelo de la atmosfera haciendo siempre luego click en el boton para pasar al paso siguiente.

Los parametros para nuestro modelo son

1. Geometrical conditions

- TM (Landsat)
 - Month: 4, Day:13, GTM decimal hour: 13.60, Longitude: -63.8606, Latitude: -24.9937.
2. Atmospheric Model
- Select Atmospheric Profile: Mid latitude summer
 - Select aerosol model: Continental Model
 - Visibility: 60
3. Target & sensor altitude
- Select target altitude: sea level
 - Select sensor altitude: satellite level
4. Spectral conditions
- Select spectral conditions: choose band
 - Select band: 1st band of tm (landat 5)
5. Ground reflectance
- Ground reflectance type: homogeneous surface
 - Directional effect: no directional effect
 - Specify surface reflectance: input constant value of ro
 - input constant value for ro: 0
6. Signal
- Atmospheric correction mode: no atmospheric correction

En podemos ver el resultado haciendo click en *Output file*

Una vez ejecutado el proceso puede usarse el siguiente código para corregir todas las bandas utilizando R.

```

1  a <- c(0.98,0.90,...)
2  b <- c(0.81,0.90,...)
3  g <- c(0.15,0.10,...)
4  r <- c(0.08,0.05,...)
5  sss.1992 <- (toa.1992/(a*b)-r/b)/(1+g*(toa.1992/(a*b)-r/b))

```

Actividad 2.5. Realice una extracción de firmas espectrales para distintas coberturas de cada uno de los archivos raster obtenidos y grafíquelos en el mismo gráfico. Comparela con la firma espectral obtenida a partir de la imagen corregida por el usgs.

3. Un abaco espectral

Veamos ahora como realizar operaciones sencillas entre las bandas de una imagen. Usaremos en esta práctica los siguientes paquetes

```

1  library(raster)
2  library(RStoolbox)
3  library(RColorBrewer)
4  library(rgdal)
5  library(ggplot2)
6  library(GGally)

```

Comenzamos primer cargando la imagen desde el metadato y convirtiendola a reflectancia como hicimos en la clase anterior

```
1 xml.2016 <- readMeta("raster_data/LC.../LC...xml")
2 ref.2016 <- stackMeta(xml.2016, quantity = "sre")
3 scaleF <- getMeta(ref.2015,xml.2016, what = "SCALE_FACTOR")
4 ref.2016 <- ref.2016 * scaleF
5 ref.2016 <- ref.2016[[-1,]]
6 names(ref.2016) <- c("blue", "green", "red", "nir", "swir1", "swir2")
```

una vez cargada la imagen podemos realizar operaciones entre las bandas llamando a cada una por separado. Veamos como ejemplo el calculo de NDVI.

Ejemplo 3.1. Calculo de NDVI a mano y grafico del mismo

```
1 ndvi.2016 <- (ref.2016$nir-ref.2016$red)/(nir.2016$nir+ref.2016$ref)
2 cols = colorRampPalette(brewer.pal(9,"YlGn"))(16)
3 plot(ndvi.2016, col=cols, zlim = c(0,1))
```

obteniendo una imagen como la que se ve debajo.

El paquete **RStoolbox** tiene varias herramientas que nos ayudan a calcular los indices espectrales. Veamos por ejemplo como calcular el NDVI y el EVI utilizando dicho paquete

Ejemplo 3.2. Para calcular los indices mediante la funcion `spectralIndices` debemos especificar con que raster trabajamos y que bandas corresponden a cada longitud de onda

```
1 indices.2016 <- spectralIndices(ref.2016,
2                               blues="blue", red="red", nir="nir",
3                               indices=c("NDVI", "EVI"))
4 plot(indices.2016, col=cols, zlim=c(0,1))
```

obtenemos una imagen como se muestra debajo.

Actividad 3.1. Calcule el NDVI para el año 2000 utilizando la imagen landsat 7.

Actividad 3.2. Calcule y grafique todos los indices posibles que involucren a las bandas roja y nir de landsat 8.

Ejemplo 3.3. Veamos ahora como calcular el tSAVI utilizando la linea de suelo obtenida a partir de la imagen. Para esto necesitaremos enmascarar las zonas con cobertura de agua y nubes. Veamos primer como hacer esto.

```
1 mask.2016 <- raster("raster/.../...cfmask.tif")
2 masked.2016 <- mask(ref.2016, mask=mask.2016, inverse=TRUE,
3                    maskvalue=0, updatevalue=255)
4 masked.2016[masked.2016<=0] <- 255
```

de esta forma enmascaramos todos los valores con nubes, agua y donde la reflectancia obtenida es cero. Calculamos ahora la linea de suelo y la mostramos en un scatterplot

```
1 bsl.2016 <- BSL(as.matrix(masked.2016$red), as.matrix(masked.2016$nir),
2               method="quantile", ulimimt=0.99, llimit=0.001)
3 plot(ref.2016$red, ref.2016$nir)
4 abline(bsl.2016$BSL, col="red")
```

Actividad 3.3. Calcule el tSAVI utilizando la linea de suelo obtenida arriba.

Actividad 3.4. Vuelva a obtener la linea de suelo sin enmascarar la imagen y dibujo el scatterplot con la misma y la anterior. Que problema encuentra.

Finalmente, veamos como se puede obtener datos biofisicos a partir de los indices de vegetacion calculados. De esta forma podremos generar mapas de porcentaje de cobertura, productividad, etc.

Actividad 3.5. Cargue la capa vectorial del muestreo de variables biofisicas `muestreo.shp` y haga una extraccion de los valores de NDVI correspondientes a dichos puntos. Guarde estos valores en un dataframe llamado `muestreo`.

Ejemplo 3.4. Veamos como ajustar con R un modelo lineal a nuestro modelo. Para esto comencemos haciendo un analisis visual con la funcion `ggpairs`.

```
1 ggpairs(muestreo, diag=list(continuous="barDiag"))
```

Obtendremos un grafico que presenta los scatterplots entre las bandas, su correlacion e histogramas. Veamos en el mismo que la superficie cubierta por vegetacion varia linealmente con el NDVI. Por lo tanto utilizaremos estos para hacer un ajuste de nuestro modelo.

```
1 lm.2016 <- lm(fcover ~ ndvi, data=muestreo)
2 plot(muestreo$ndvi, mustreo$fcover)
3 abline(lm.2016, col="red")
4 summary(lm.2016)
```

de esta forma veremos los parametros de nuestro ajuste, y graficaremos al mismo en un scatterplot. Para aplicar el modelo a nuestro raster hacemos

```
1 fcover.2016 <- predict(ndvi.2016, lm.2016)
2 plot(fcover.2016)
```

Obteniendo el mapa de abajo.

Actividad 3.6. Genere los modelos de lai, fapar y fcover para el año 2016 y con los mismos realice mapas de dichas variables.

Actividad 3.7. Utilizando los modelos obtenidos para 2016 aplique los mismos para obtener los mapas de lai, fapar y fcover del año 2000. Que suposicion esta haciendo?

Actividad 3.8. * Utilizando la funcion `spectralIndices` y `ggpairs`, analice si hay otro indice que ajuste que correlacione mejor con las alguna de las variables biofisicas medidas a campo.

4. Rotaciones espectrales

Durante la clase de hoy trabajaremos con rotaciones en el espacio espectral. A diferencia del trabajo con indices las rotaciones pueden interpretarse no como algebra entre las bandas sino como distintas formas de mirar al mismo espacio espectral.

En este caso usaremos las librerias `raster` y `RStoolbox`.

```
1 library(raster)
2 library(RStoolbox)
3 library(bfastSpatial)
```

Ejemplo 4.1. Comencemos analizando la transformada por componentes principales de la imagen de 2016. Que podemos predecir?


```
1      pairs(ref.2016)
```

Mirando el resumen de la imagen vemos que hay varias bandas muy correlacionadas entre si. Por ejemplo las del visible, mientras que otras lo estan poco, por el ejemplo el nir y el swir. Por lo tanto esperamos que no todas las bandas sean necesarias para explicar el comportamiento de la imagen

```
1      pca.2016 <- rasterPCA(ref.2016)
2      summary(pca.2016$model)
3      loadings(pca.2016$model)
4      plot(pca.2016$map)
```

Actividad 4.1. Calcule y analice la transformada por PCA de la imagen Landsat 7 del año 2000.

Ejemplo 4.2. Otra aplicacion de la transformada por componentes principales por componentes principales. Veamos como realizarlo.

```
1      ndvi.list <- list.files("raster_data/MOD13Q1/EVI/", pattern = "*.tif",
2                             full.names = TRUE)
3      ndvi.stack <- stack(ndvi.list)
```

una vez abierta la imagen la convertimos a valores entre -1 y 1 e interpolamos los valores que falten.

```
1      ndvi.stack <- ndvi.stack/1e4
2      ndvi.stack <- approxNA(ndvi.stack)
```

Una vez llenados los espacios donde no habia datos podemos aplicar la transformada por componentes principales y mostrarla

```
1      ndvi.pca <- rasterPCA(ndvi.stack)
```

Actividad 4.2. Grafique las primeras 4 componentes por de la transformada por componentes principales de la imagen del stack de NDVI. Que zonas puede identificar en la primera? que zonas se distinguen en la segunda? que comportamiento encuentra en la tercera y cuarta.

Actividad 4.3. Investigue la funcion `tasseledCap` y calcule la transformada tasseled cap para las imagenes landsat 7 y 8.

Actividad 4.4. Grafique en el scatter-plot la imagen completa y marque en el mismo zonas con vegetacion, agua y suelo sin cobertura vegetal. Vea como cambian estas zonas frente a las transformadas por componentes principales y tasseled cap.

5. Clasificacion no supervisada de imagenes

En esta clase vamos a trabajar con clasificaciones no supervisadas de imagenes satelitales. Vamos a usar los paquetes

```
1      library(raster)
2      library(RStoolbox)
```

Cargaremos primero la imagen landsat 8 y habilitaremos la opcion para escribir el header de ENVI.

```

1 rasterOptions(addheader = "ENVI")
2 set.seed(6)
3 kmeans.2016 <- unsuperClass(ref.2016, nClasses = 5, nStarts = 100,
4                             nSamples = 100)
5 writeRaster(kmeans.2016, "raster_data/processed/kmeans2016",
6               datatype="INT1U")

```

Podemos ahora graficar por separado cada una de las clases

```

1 classes.2016 <- layerize(kmeans.2016)
2 plot(classes.2016)

```

Abriremos la imagen ahora en el qgis e identificaremos cada una de las clases realiendo interpretacion visual de la imagen.

Para realizar la identificacion primero vamos al menu → → → . Elegimos de modo Intervalo Igual y en numero de clases ponemos con el minimo en 1 y el maximo en 100. En estilo de color elegimos colores aleatorios. Iremos luego cambiando los colores uno a uno por un color brillante e identificado a que cobertura pertenece dicha clase espectral.

Construiremos con ella una tabla como la siguiente

id	class
1	1
2	1
3	2
4	5
5	7

que guardaremos en un archivo de texto. El mismo lo utilizaremos para realizar la fusion de clases.

Una vez conocidas las categorias de uso y cobertura correspondientes a cada clase espectral podemos combinarlas

```

1 classes.2016 <- read.delim("class")
2 reclas.2016 <- subs(kmeans.2016$map, classes.2016)

```

Actividad 5.1. Clasifique por el metodo de kmeans la imagen en reflectancia con una cantidad de clases espectrales lo suficientemente altas para separar todas las clases espectrales.

Actividad 5.2. Vuelva a repetir la clasificacion utilizando la imagen obtenida de la transformada por componentes principales descartando las bandas que aporten menos informacion.

Podemos ahora utilizar la clasificacion para separar zonas de la imagen en el espacio espectral

```

1 ref.2016$kmeans <- reclas.2016
2 xyplot(nir~red, groups=kmeans, data=ref.2016)

```

Actividad 5.3. Grafique en los cortes del espacio espectral la imagen sin fusionar. Compare la diferencia entre clases espectrales y clases de informacion.

6. Clasificacion supervisada de imagenes

En esta practica seguiremos trabajando con la clasificacion supervisada de imagenes satelitales. Utilizaremos mas paquetes en este caso.

```

1 library(RStoolbox)
2 library(rgdal)
3 library(raster)
4 library(rasterVis)

```

ademas de los paquetes que incorporan los distintos metodos de clasificacion

```

1 library(caret)
2 library(randomForest)
3 library(e1071)
4 library(kernlab)

```

comenzamos abriendo la imagen del año 2016 para las bandas reflectivas como en la clase anterior. Abrimos tambien el vector de entrenamiento.

Empecemos con la clasificacion por el metodo de maxima verosimilitud

```

1 sup.2016 <- superClass(ref.2016, vector, responseCol = "MC_ID",
2                        model = "mlc")

```

y realizar el scatterplot de dichas variables como.

```

1 ref.2016 <- sup.2016
2 xyplot(nir~red, groups=mlc, data=ref.mlc)

```

Cambiando el algoritmo de clasificacion en el parametro `model` podemos calcular distintas clasificaciones supervisadas. Algunas de las vistas en clase son `rf`, `svmRadial`, `kNN`. Cada una de ellas usa alguna libreria adicional de las cargadas antes.

Actividad 6.1. Realice clasificaciones por los distintos metodos y comparelas visualmente.

Para poder comparar en que zonas los clasificadores presentan mas o menos dispersion podemos calcular la entropia de las distintas clasificaciones en cada pixel. Para esto utilizaremos la funcion `rasterEntropy`. Para esto comenzamos corriendo la clasificacion para distintos modelos, los apilados y despues calculamos la entropia de los mismos

```

1 modelos <- c("rf", "mlc", "svmRadial", "svmLinear", "kNN")
2
3 ensemble <- lapply(modelos, function(mod){
4   set.seed(5)
5   sc <- superClass(ref.2016, trainData = vector,
6                   responseCol = "MC_ID", model=mod)
7   return(sc$map)
8 })
9 prediction_stack <- stack(ensemble)
10 names(ensemble) <- modelos
11
12 model_entropy <- rasterEntropy(prediction_stack)
13
14 plot(model_entropy)

```

7. Tecnicas pos-clasificacion

Veamos ahora algunas tecnicas de que permiten mejorar las clasificaciones y nos ayudaran a validar y extraer datos de las imagenes clasificadas.

Comenzamos viendo como aplicar un filtro a una imagen. Comenzamos cargando las librerias utilizar.

```

1 library(RStoolbox)
2 library(rgdal)
3 library(raster)
4 library(rasterVis)

```

Para aplicar un filtro a una imagen monobanda, debemos primero definir cual es la ventana en la que trabajaremos y luego cual es la operacion que desamos realizr en dicha ventana.

```

1 window <- matrix(1,nrow=3, ncol=3)
2 clasificacion.3x3<-focal(clasificacion.2016,w=window, fun=modal)

```

En el caso de un filtro por moda, estaremos dejando el mayor que mas veces aparezca entre los que rodean al pixel.

Actividad 7.1. Aplique filtros de 5x5 y 7x7 para filtrar la imagen. Que problemas desaparecen? que dificultad introducen.

Actividad 7.2. Aplique el filtro de 3x3 a la imagen correspondiente al año 2000.

Actividad 7.3. Utilice la funcion raster sieve de qgis para realizar un filtrado espacial. Que diferencias encuentra.

Una vez filtrada la imagen podemos obtener de la misma la matriz de confusion. Para esto debemos cargar el poligono de validacion y la calculamos con la funcion `validateMap`.

```

1 valid.2016 <- readOGR(dsn="vector_data/", layer="validacion")
2 val.unsup.2016 <- validateMap(sup.2016$map, valData = valid,
3                             responseCol = "MC_ID")

```

Actividad 7.4. Construya la matriz de confusion y obtenga la presicion global para todas las clasificaciones. Que algoritmo funciona mejor con la imagen?

Otra forma de construir incorporar contexto espacial a las clasificaciones es contruir una capa de textura. Veamos como construir una banda de textura utilizando la banda pancromatica de Landsat degradada a 30m.

Ejemplo 7.1. Comenzamos cargando la imagen pancromatica

```

1 pan.2016 <- raster("raster_data/LE72240782000188EDC00/LE72240782000188EDC00
  _B8.TIF")

```

Una vez cargada podemos visualizarla como

```

1 plot(pan.2016)

```

calculamos ahora el estimador mas sencillo de la textura mediante el calculo del desvio standar en una ventana de 3x3

```

1 windows <- matrix(1,nrow=3,ncol=3)
2 sd.2016 <- focal(pan.2016,window, fun=sd)

```

desvio que pondemos graficar como

```

1 plot(sd.2016)

```

finalmente, degradamos el mapa obtenido a 30m para poder utilizarlo con la imagen multiespectral.

```
1      sd.agregate.2016 <- aggregate(sd.2016, fact=2, fun=mean)
```

finalmente usamos la funcion **stack** para juntar todas las bandas y proceder a la clasificacion.

```
1      context.2016 <- stack{ref.2016, sd.agregate.2016}  
2      class.2016 <- supClas{}
```