Introducción

1. Un viaje del sol a los pixeles.

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 estadistico de la imagen como un todo y de las distintan coberturas digitalizar en R.

1.1. Exploración de imagenes con el qgis

Comenzamos abriendo la imagen `LC82240782016304LGN00.vrt` que se encuentra en la carpeta `raster_data/LC82240782016304`. Esta image corresponde al departamento de Iguazu en la provincia de corriente. La misma fue obtenida por el satelite Landsat 8 durante el mes de noviembre de 2016.

Para esto vamos al menú *Capa* → *Añadir capa* → *Añadir capa ráster*. Navegamos hasta la carpeta `raster_data/LC8224078201630` y abrimos el archivo `LC82240782016304LGN00.vrt`. Una vez abierto el mismo podremos encontrarlo en el *Panel de capas* de q-gis donde podremos manejar la visualización del mismo y estudiar las propiedades de dicha capa.

Figura 1 – Herramientas para moverse dentro de la imagen. De izquierda a derecha: 1. Desplazar mapa, 2. Desplazar mapa a la seleccion, 3. Acercar zum, 4. Alejar zum, 5. Zum a la resolucion nativa, 6. Zum general, 7. Zum a la seleccion, 8. Zum a la capa, 9. Zum anterior, 10. Zum siguiente, 11. Actualizar.

Para realizar cambiar la visualizacion y explorar los datos de una capa, hacemos click derecho sobre la misma y luego seleccionamos la opciopn *Propiedades*. Dentro de las propiedades de la capa podemos ir a la pestaña *General* para ver datos como el nombre de la capta[1], la cantidad de filas y columnas del archivo, el valro digital no valido, el sistema de referencias de coordenadas entro otros.

Podemos ir luego a la pestaña de estilo para cambiar la visualizacion de la imagen. En la misma podemos elegir de que color mostraremos cada una de las capas ademas de elegir el tipo de realce que deseamos aplicar. Es importante remarcar en este caso que una vez elegidas las badnas debemos hacer click en el boton *Cargar* para seleccionar los valores maximos y minimos de las bandas para el realcec.

Actividad 1.1. Cambie la combinación de bandas de la imagen L8 y muévase dentro de la misma. Identifique zonas de coberturas uniformes. Pruebe cambiar de combinacion de bandas y decida si dichas zonas siguen siendo uniformes luego del cambio.

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

Para identificar la informacion correspondiente a un punto en el espacio podemos utilizar la herramienta *Identificar un objeto espacial*. Al habilitarla y hacer click sobre un punto de la imagen veremos datos de la misma como por ejemplo los valores de reflectancia del pixel seleccionado. Dichos valores pueden mostrarse como Arbol, Tabla o Grafo segun como sea mass comodo.

[1] Es un buen momento para ponerle uno mas sencillo

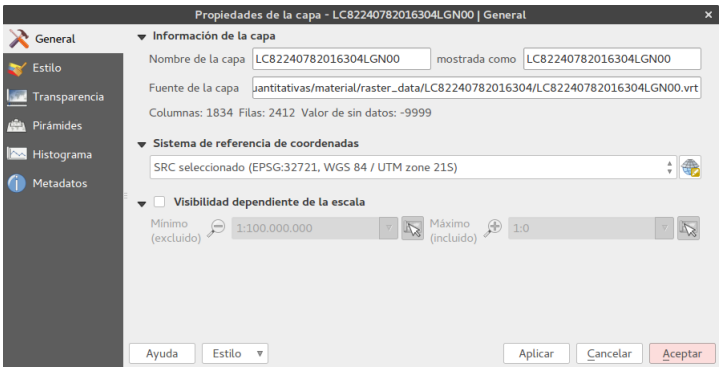

Figura 2 – Pestaña general de propiedades de una capa. En la misma se pueden ver los datos mas importantes sobre la misma como la cantidad de filas y columnass, el nombre y el sistema de referencia de coordenadas.

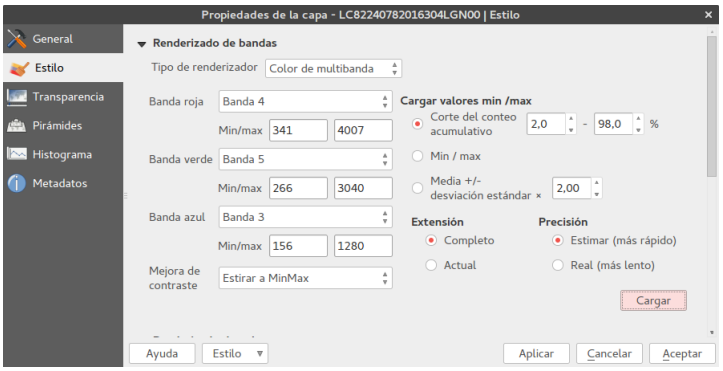

Figura 3 – Estilos de visualizacion de una capa raster. Los estilos posibles son: 1. Color de multibanda, 2. En paleta, 3. Unibanda gris, 4. Unibanda pseudocolor. Puede explorar cada uno por separado ya que todos tendran distintas utilidades.

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.

1.2. Creacion de capas vectoriales

Veamos ahora como crear capas vectoriales con las cuales podamos extraer informacion sobre nuestra zona.

Con la herramienta nueva capa de archivo shape es posible digitalizar zonas de la imagen para su posterior analisis. Para esto puede hacer click en el boton del panel lateral. Podemos agregar los campos que sean necesarios para nuestra capa vectorial en este momento. Crearemos al menos los campos MC_ID como entero de longitud 1 y Comment como texto de 80 caracteres. Guardela 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 geometrias a la misma. Para esto hacemos click en el boton de agregar geometrica y digitalizamos una zona uniforme dentro de la imagen.

Al terminar de acerlo qgis pedira un numero de ID para la capa que debe ser correlativo. Además podremos ingresar en este momento los valores del resto de los campos de nuestro objeto espacial.

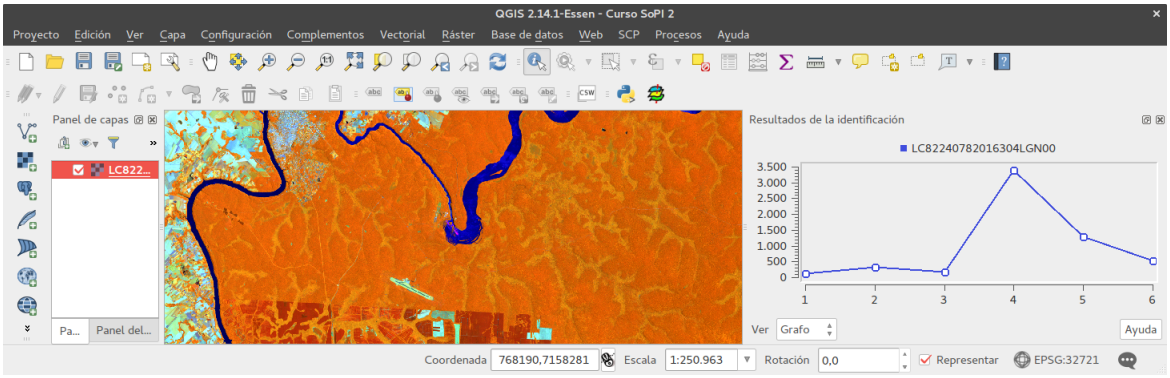

Figura 4 – Identificacion de un pixel correspondiente a la selva paranaense mostrada como grafo.

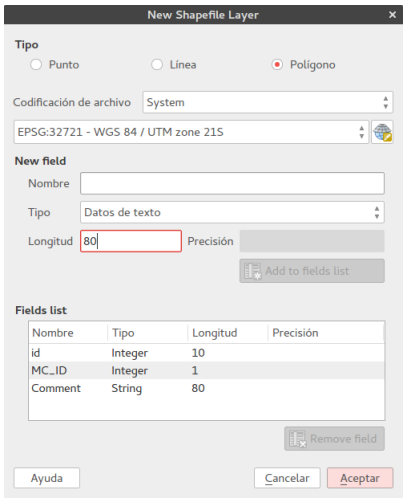

Figura 5 – Creacion de una nueva capa vectorial. Se agregan campos que seran de interes para comparar las firmas espectrales.

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

En caso de desear cambiar la visualizacion de la capa vectorial, podemos entrar a las propiedades de la misma[2]. Ademas podemos acceder a la tabla de datos de la capa vectorial haciendo click derecho sobre la misma y eligiendo la opcion *Abrir tabla de atributos*.

1.3. Exploracion raster en R

Busquemos ahora como abrir y trabajar con las imagenes satelitales en R. Para esto comenzamos cargando las librerias que vamos a utilizar con el comando `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.

Tambien podemos utilizar el comando `setwd()` para configurar el directorio 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.

[2]Pueded utilizar el estilo precargado ubicado en la carpeta `aux_data`

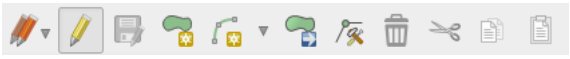

Figura 6 – Herramientas de edición vectorial. De izquierda a derecha: 1. Conmutar edicion, 2. Guardar cambios a la capa, 3. Añadir objeto espacial, 4. Añadir cadena circular, 5. Mover objeto espacial, 6. Herramienta de nodos, 7. Borrar lo seleccionado, 8. Cortar objetos espaciales, 9. Copiar objetos espaciales, 10. Pegar objetos espaciales.

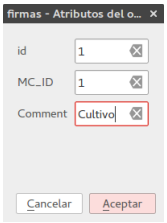

Figura 7 – Valores de los campos del nuevo poligono creado.

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.

```
ref.2016 <- brick("raster_data/LC82240782016304/LC82240782016304LGN00.vrt")
ref.2016
```

obtenemos de resultado el siguiente text

```
class       : RasterBrick
dimensions  : 2412, 1834, 4423608, 6  (nrow, ncol, ncell, nlayers)
resolution  : 30.00402, 30.00265  (x, y)
extent      : 731118.6, 786146, 7101531, 7173897  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=utm +zone=21 +south +datum=WGS84 +units=m +no_defs
              +ellps=WGS84 +towgs84=0,0,0
data source : ./material/raster_data/LC82240782016304/LC82240782016304LGN00.vrt
names       : LC82240782016304LGN00.1, LC82240782016304LGN00.2, ...
min values  :                   -33,                   192, ...
max values  :                  2774,                  3265, ...
```

En el podemos ver la clase a la que corresponde el archivo abierto, en este caso un *RasterBrick*, las dimensiones, el tamaño de pixel, extension de la capa, proyeccion, cual es la ruta al archivo que abrimos, las bandas y sus valores maximos y minimos.

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

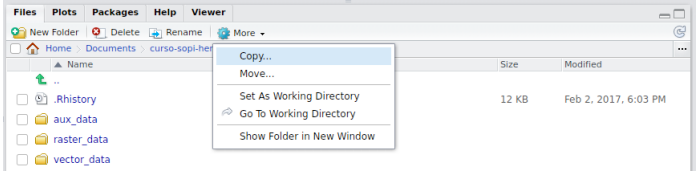

Figura 8 – Configuracion del directorio de trabajo desde la interfaz grafica.

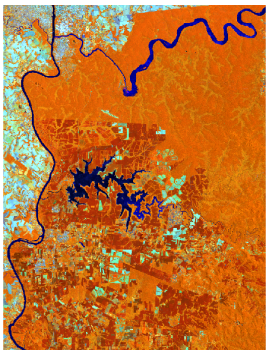

Figura 9 – Combinacion de bandas nir-swir1-red en R.

Ejemplo 1.2. 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.

```
ref.2016 <- brick(filename)
names(ref.2016) <- c("blue","gree","red","nir","swir1","swir2")
ref.2016 <- ref.2016/1e4
rasterOptions(addheader = "ENVI")
writeRaster(ref.2016,"raster\_data/processed/ref2016")
```

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.
- La cuenta linea es necesaria correrla una sola vez por sesion. La misma agrega el header de ENVI a nuestro output para poder abrir el archivo desde qgis
- La sexta linea guarda el archivo raster con el nombre `ref2016`

podemos ademas graficar tanto una combinacion de bandas en qgis

```
plotRGB(ref.2016,r=4,g=5,b=3, stretch='lin')
```

Obtenemos como resultado como tambien todas las bandas por separado

```
plotRGB(ref.2016)
```

obtenemos como resultado

Actividad 1.5. Abra el archivo vrt en qgis y vuelva a mirar la firma espectral para distintas coberturas. Entre que valores se encuentra ahora las mismas.

Ejemplo 1.3. Hagamos un poco de analisis ahora sobre la imagen. En primer lugar podemos calcular un sumario de la estadistica de nuestra imagen

```
summary(ref.2016)
```

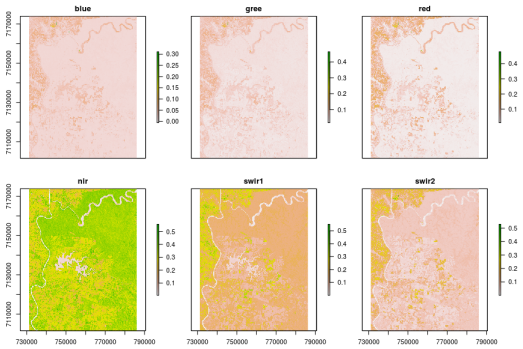

Figura 10 – Grafico de bandas con realce automatico para cada una.

obtenemos como resultado

```
            blue    gree     red      nir    swir1    swir2
Min.    -0.0278 0.0000 0.0000 -0.0128 -0.0069 -0.0038
1st Qu.  0.0128 0.0328 0.0184  0.2763  0.1198  0.0493
Median   0.0138 0.0362 0.0203  0.3287  0.1365  0.0572
3rd Qu.  0.0170 0.0450 0.0329  0.3557  0.1644  0.0749
Max.     0.5548 0.8257 0.8034  0.7542  0.9181  0.9446
NA's     0.0000 0.0000 0.0000  0.0000  0.0000  0.0000
```

Para comenzar podemos calcular los histogramas de todas las bandas con el comando

```
1  hist(ref.2016)
```

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)
```

1.4. Manejo vectorial en R

Hasta ahora estamos analizando la imagen completa. Podemos sin embargo analizar solo sectores concretos de la imagen muestreandola en funcion de un archivo vectorial. Tambien sera posible muestrar la imagen pos zonas definidas por otro raster pero veremos esto mas adelante.

Para poder trabajar con vectores en R utilizaremos la libreria `library(rgal)`.

Ejemplo 1.4. Podemos leer un vector como

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

Notamos en este caso que debemos indicar por separado la carpeta que contiene al shapefilee en *dsn* y el nombre de la capa que queremos abrir como *layer*.

Podemos mostrar las propiedades del vector ejecutando el comando

```
vector
```

obteniendo como resultado

```
class       : SpatialPolygonsDataFrame
features    : 8
extent      : 738692.8, 767774.6, 7133396, 7165265  (xmin, xmax, ymin, ymax)
coord. ref. : +proj=utm +zone=21 +south +datum=WGS84 +units=m +no_defs
              +ellps=WGS84 +towgs84=0,0,0
variables   : 3
names       : id, MC_ID,       Comment
min values  :  0,     1,          Alto
max values  :  9,     8, Suelo desnudo
```

Podemos graficar los vectores obtenidos en R junto aa la imagen de base como

```
plotRGB(ref.2016, stretch="lin")
plot(firmas,add=TRUE,col='red')
```

donde la primera linea grafica la imagen de fondo y la segunda agrega el el shapefile sobre la misma.

Actividad 1.6. Muestre las propiedades de la capa raster y el vector abiertos y verifique que los mismos se encuentren en el mismo sistema de coordenadas.

Por ultimo mostremos como extraer datos de un archivo raster y veamos un par de ejemplo concretos. La funcion que nos permite extrar datos de un raster segun un vector es `extract` que toma dos argumentos, el vector que queremos utilizar y la capa raster sobre la cual hacer la consulta.

Veamos algunos ejemplos

Ejemplo 1.5. Graficar en un scatterplot de dos bandas mostrando la zona del espacio ocupada por una cobertura.

```
datos <- extract(ref.2016,firmas)
```

de esta forma realizamos la extraccion de todos los datos de la imagen a una lista

```
plot(ref.2016$red, ref.2016$nir)
points(as.data.frame(datos[1])$red, as.data.frame(datos[1])$nir,col="green",
       pch = ".")
```

obteniendo como resultado

La funcion `extract` nos permite tambien aplicar una funcion a los datos extraidos antes de entregarlos al usuario. Veamos como usarla para calcular datos de interes sobre las coberturas.

Ejemplo 1.6. Extraer los promedios y desvios standar de un raster y agregarlos a un vector. Primero extraemos los valores de promedio y desvio

```
promedio <- extract(ref.2016,firmas,fun=mean)
desvio <- extract(ref,firmas,fun=sd)
```

renombramos luego las columnas como promedio y devio seguido de la banda a la que pertenecen,

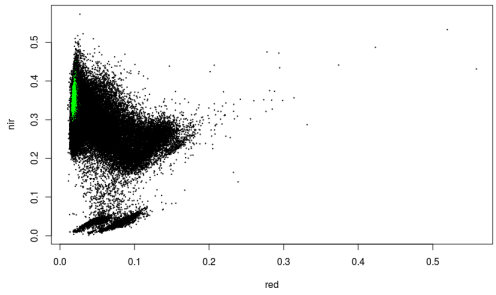

Figura 11 – Resultado del scatterplot para las bandas roja y nir. Se muestra en verde datos correspondientes a la selva paranaense.

```
colnames(promedio) <- paster("mean",colnames("promedio"),sep="_")
colnames(desvio) <- paster("sd",colnames("desvio"),sep="_")
```

finalmente agregamos los archivos a un nuevo shapefile

```
firmas@data <- cbind(firmas@data,promedio,desvio)
writeOGR(firmas, sdn="vector_data/processed/","firmas_datos",
         driver="ESRI Shapefile")
```

Por ultimo, veamos como usar una capa vectorial para graficar para extraer las firmas espectrales y graficarlas para distintas coberturas

Ejemplo 1.7. Graficar las firmas espectrales en funcion de la longitud de onda para cada geometria de un vector. Utilizaremos en este caso una nueva libreria, `reshape2`

Comenzamos convirtiendo en dataframe a nuestros promedios donde cada columna corresponde a una firma espectral

```
df <- t(promedio)
colnames(df) <- vector@data$Comment
```

Agregamos luego una columna con las longitudes de onda en nanometros. Luego reformamos el dataframe para que podamos subsetearlo, poniendo finalmente los nombres a cada columna

```
df$wl <- as.matrix(c(485,560,660,830,1650,2215))
df <- melt(df,id.vars="wl", variable.name="cobertura")
names(df) <- c("wl","Cobertura","Reflectancia")
```

si mostramos el dataframe, el resutado debería ser similar al siguiente

```
     wl     Cobertura Reflectancia
1   485          Alto  0.012926561
2   560          Alto  0.034730646
3   660          Alto  0.018491884
4   830          Alto  0.354564681
5  1650          Alto  0.133750642
...
```

repetimos el proceso para los desvios standar

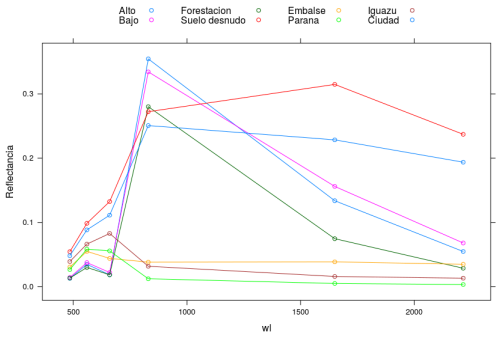

Figura 12 – Firmas espectrales

```
dfd <- t(desvio)
colnames(dfd) <- vector@data$Comment
dfd$wl <- as.matrix(c(485,560,660,830,1650,2215))
dfd <- melt("wl","Cobertura","Desvio")
df$desvio <- dfd$desvio
df$MC_ID <- as.character(vector@data$MC_ID[match(df$Cobertura,
                                vector@data$Comment)])
```

el resultado sera ahora

```
    wl     Cobertura Reflectancia       Desvio
1   485          Alto  0.012926561 0.0007772473
2   560          Alto  0.034730646 0.0018113004
3   660          Alto  0.018491884 0.0011561294
4   830          Alto  0.354564681 0.0166801398
5  1650          Alto  0.133750642 0.0075157929
...
```

Veamos algunas opciones para generar ahora los graficos. En primer lugar pondremos todas las firmas juntas, separadas por color

```
xyplot(Reflectancia~wl, data=df, groups = Cobertura,
       auto.key=list(space="top", columns=4),
       ty=c("l", "p"))
```

Aqui la primer linea dice que grafiquemos la reflectancia como funcion de la longitud de onda, obteniendo los datos del dataframe df y agrupandolos segun la columna cobertura. La siguiente linea agrega la leyenda en la parte superior de la figura y con 4 columnas. Por ultimo en la tercer linea pedimos que el grafico tenga lineas y puntos. Si queremos agruparlo por categoria de uso y cobertura cambiamos la formula `Reflectancia  wl` por `Reflectancia  wl | MC_ID`

```
xyplot(Reflectancia~wl | MC_ID, data=df, groups = Cobertura,
       auto.key=list(space="top", columns=4),
       ty=c("l", "p"))
```

y finalmente si queremos graficar solo un subset de los daatos

```
xyplot(Reflectancia~wl | MC_ID, data=df, groups = Cobertura,
       auto.key=list(space="top", columns=4), ty=c("l", "p"),
       subset = Cobertura %in% c("Alto","Bajo"))
```

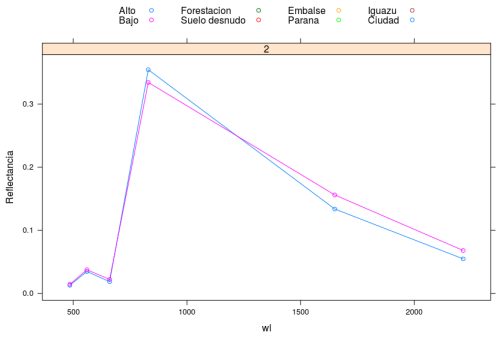

Figura 13 – Firmas espectrales

Actividad 1.7. 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 diversar 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/
             LT52300771992104CUB00_MTL.txt")
4        dn.1992 <- stackMeta(meta.1992)
5        dn.1992 <- dn.1992[[-6,]]
6        ggRGB(dn.1992,
7              r=4, g=3, b=2,
8              geom_raster = TRUE,
9              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

```
calref.1992 <- meta.1992$calref
elev.1992 <- pi*meta.1992$SOLAR_PARAMETERS['elevation']/180
dn2ref.1992 <- meta.1992$CALREF
toa.1992 <- (dn.1992*dn2ref.1992$gain+dn2ref.1992$offset)/sin(elev.1992)
names(toa.1992) <- c("B1_toa","B2_toa","B3_toa","B4_toa","B5_toa","B7_toa")
```

de forma automaica

```
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 obscuro.

```
haze.1992 <- estimateHaze(dn.1992,darkProp = 0.01, hazeBands = 1:4, plot=TRUE)
sdos.1992 <- radCor(dn.1992, metaData = meta.1992,
                    hazeValues = haze.1992,
                    hazeBands = c("B1_dn","B2_dn","B3_dn","B4_dn"),
                    method="sdos")
```

Actividad 2.2. Analice los valores de haze obtenidos por la funcion stimate hace 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 raser 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 *Submit query*. Iremos luego configurando paso a paso nuestro modelo de la atmosfera haciendo siempre luego click en el boton *submit query* 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 targe 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 *7.Results* podemos ver el resultado haciendo click en *Output file*
Una vez ejecutado el proceso puede usarse el siguiente codigo para corregir todas las bandas utilizando R.

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

Actividad 2.5. Realice una extraccion de firmas espectrales para distintass coberturass de cada uno de los archivos raster obtenidos y grafiquelos en el mismo grafico. 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 practica los siguientes paquetes

```
library(raster)
library(RStoolbox)
library(RColorBrewer)
library(rgdal)
library(ggplot2)
libyrary(GGally)
```

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

```
xml.2016 <- readMeta("raster_data/LC.../LC....xml")
ref.2016 <- stackMeta(xml.2016, quantity = "sre")
scaleF <- getMeta(ref.2015,xml.2016, what = "SCALE_FACTOR")
ref.2016 <- ref.2016 * scaleF
ref.2016 <- ref.2016[[-1,]]
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

```
ndvi.2016 <- (ref.2016$nir-ref.2016$red)/(nir.2016$nir+ref.2016$ref)
cols = colorRampPalette(brewer.pal(9,"YlGn"))(16)
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

```
indices.2016 <- spectralIndices(ref.2016,
                                blues="blue", red="red", nir="nir",
                                indices=c("NDVI","EVI"))
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.

```
mask.2016 <- raster("raster/.../...cfmask.tif")
masked.2016 <- mask(ref.2016, mask=mask.2016, inverse=TRUE,
                    maskvalue=0, updatevalue=255)
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

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

```
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.

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

```
fcover.2016 <- predict(ndvi.2016,lm.2016)
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 varibles 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`.

```
library(raster)
library(RStoolbox)
library(bfastSpatial)
```

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

```
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 ejeplo el nir y el swir. Por lo tanto esperamos que no todas las bandas sean necesarias para explicar el comportamiento de la imagen

```
pca.2016 <- rasterPCA(ref.2016)
summary(pca.2016$model)
loadings(pcs.2016$model)
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.

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

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

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

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

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

Actividad 4.2. Grafique las primeras 4 componentes por de la transformada por componentes princiales 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 supervisada de imagenes

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

```
library(RStoolbox)
library(rgdal)
library(raster)
library(rasterVis)
```

ademas de los paquetes que incorporan los distintos metodos de clasificacion

```
library(caret)
library(randomForest)
library(e1071)
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

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

y realizar el scatterplot de dichas variables como.

```
ref.2016 <- sup.2016
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 5.1. 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

```
modelos <- c("rf","mlc","svmRadial","svmLinear", "kNN")

ensemble <- lapply(modelos, function(mod){
set.seed(5)
sc <- superClass(ref.2016, trainData = vector,
                 responseCol = "MC_ID", model=mod)
return(sc$map)
})
prediction_stack <- stack(ensemble)
names(ensemble) <- modelos

model_entropy <- rasterEntropy(prediction_stack)

plot(model_entropy
```

6. Clasificacion no supervisada de imagenes

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

```
library(raster)
library(RStoolbox)
```

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

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

Podemos ahora graficar por separado cada una de las clases

```
classes.2016 <- layerize(kmeans.2016)
plot(classes.2016)
```

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

Para realizar la identificacion primero vamos al menu *propiedades de la imagen* → *Estilo* → *Tipo de renderizacion* → *Unibanda pseudocolor*. 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

```
clases.2016 <- read.delim("class")
reclas.2016 <- subs(kmeans.2016$map, clases.2016)
```

Actividad 6.1. 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 6.2. 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

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

Actividad 6.3. Grafique en los cortes del espacio espectral la imagen sin fusionar. Compare la diferencia entre clases espectrales y clases de informacion.

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.

```
library(RStoolbox)
library(rgdal)
library(raster)
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.

```
window <- matrix(1,nrow=3, ncol=3)
clasification.3x3<-focal(clasification.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`.

```
valid.2016 <- readOGR(dsn="vector_data/", layer="validacion")
val.unsup.2016 <- validateMap(sup.2016$map,valData = valid,
                              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

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

Una vez cargada podemos visualizarla como

```
plot(pan.2016)
```

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

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

desvio que pondemos graficar como

```
plot(sd.2016)
```

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

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

finalmente usamos la funcion `stact` para juntar todas las bandas y proceder a la clasificacion.

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