



Sevilla R - Sevilla, 12 de Noviembre de 2019

Teledetección con R.

Jerónimo Carranza Carranza

jeronimo.carranza@asterionat.com



www.asterionat.com

Índice

I. Introducción a la Teledetección

II. Principales paquetes de R para Teledetección

1) raster

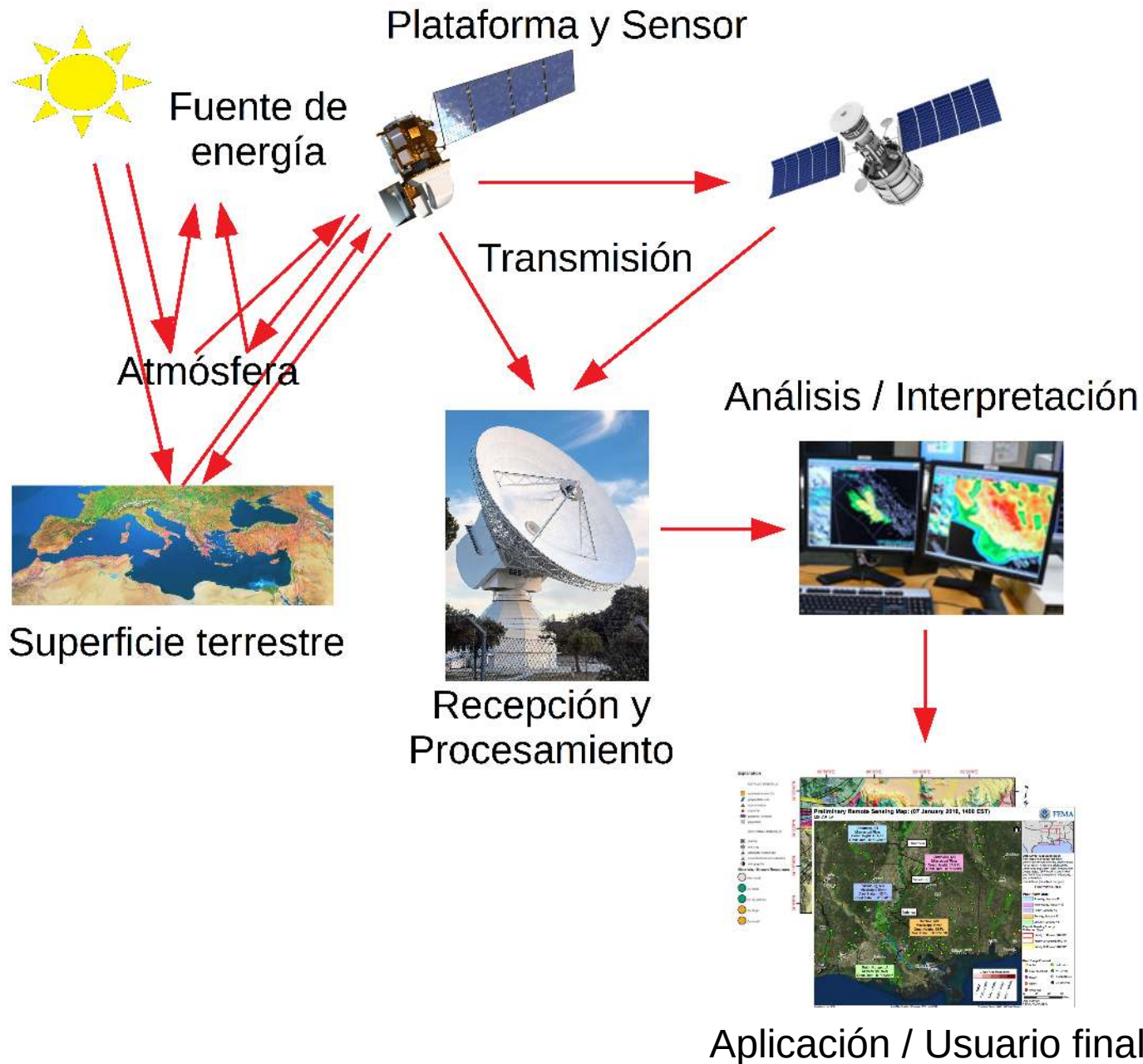
2) gdalcubes

3) sen2r

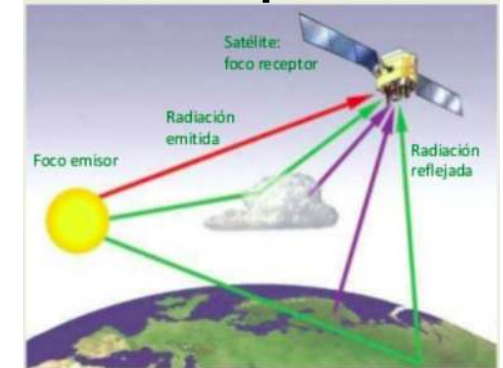
Introducción a la Teledetección

- *La teledetección es un modo de obtener información acerca de objetos tomando y analizando datos sin que los instrumentos empleados para adquirir los datos estén en contacto directo con el objeto. (Agencia Europea del Espacio -ESA-)*
- *La teledetección es el proceso de detectar y monitorizar las características físicas de un área midiendo su radiación reflejada y emitida a una distancia del área objetivo. (Servicio Geológico de Estados Unidos -USGS-)*
- *La teledetección es la ciencia y la tecnología de capturar, procesar y analizar imágenes, junto con otros datos físicos de la Tierra y los planetas, desde sensores en el espacio, en el aire y en el suelo. (Sociedad Internacional de Fotogrametría y Teledetección -ISPRS-).*

Elementos de un Sistema de Teledetección



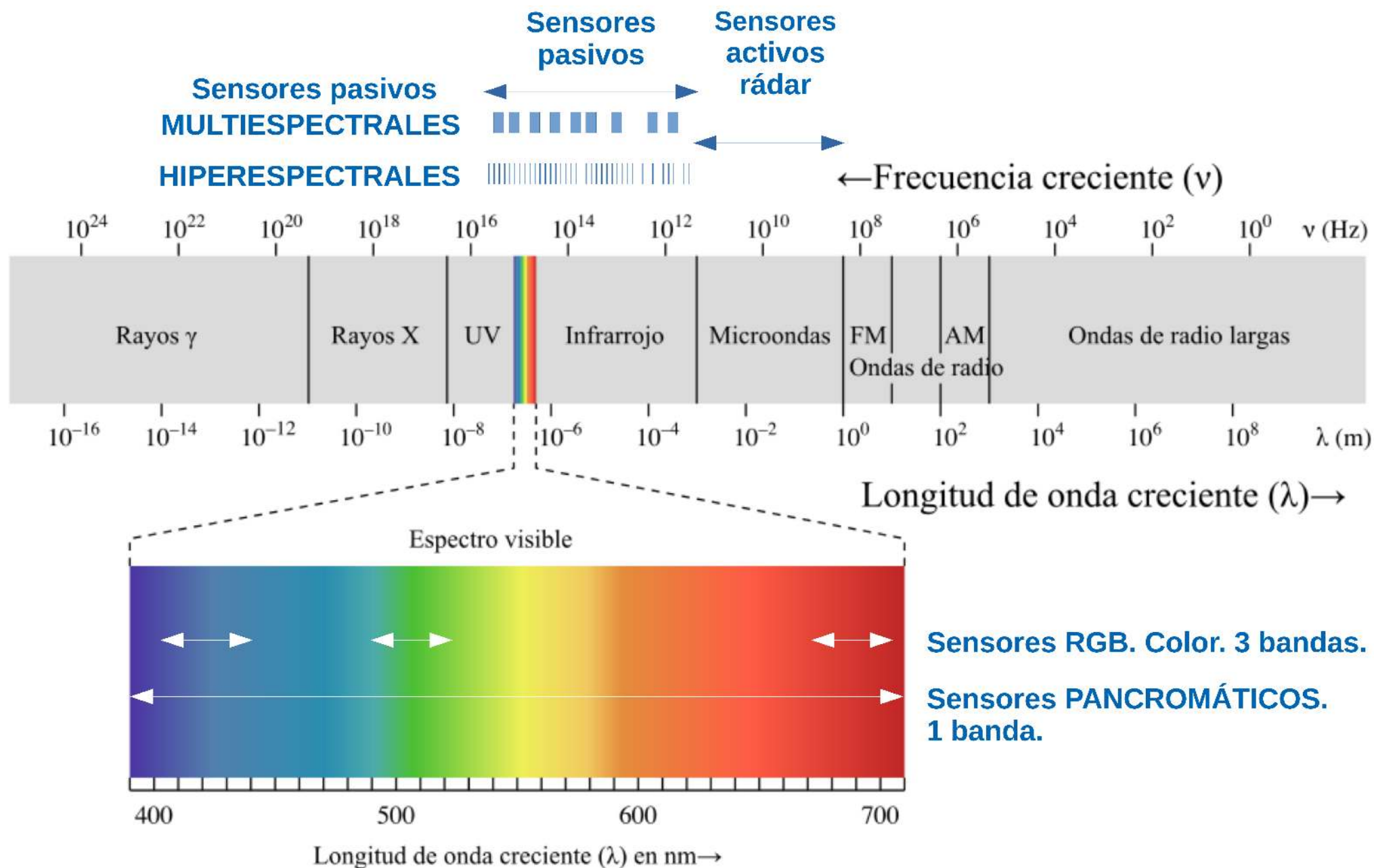
Sensor pasivo



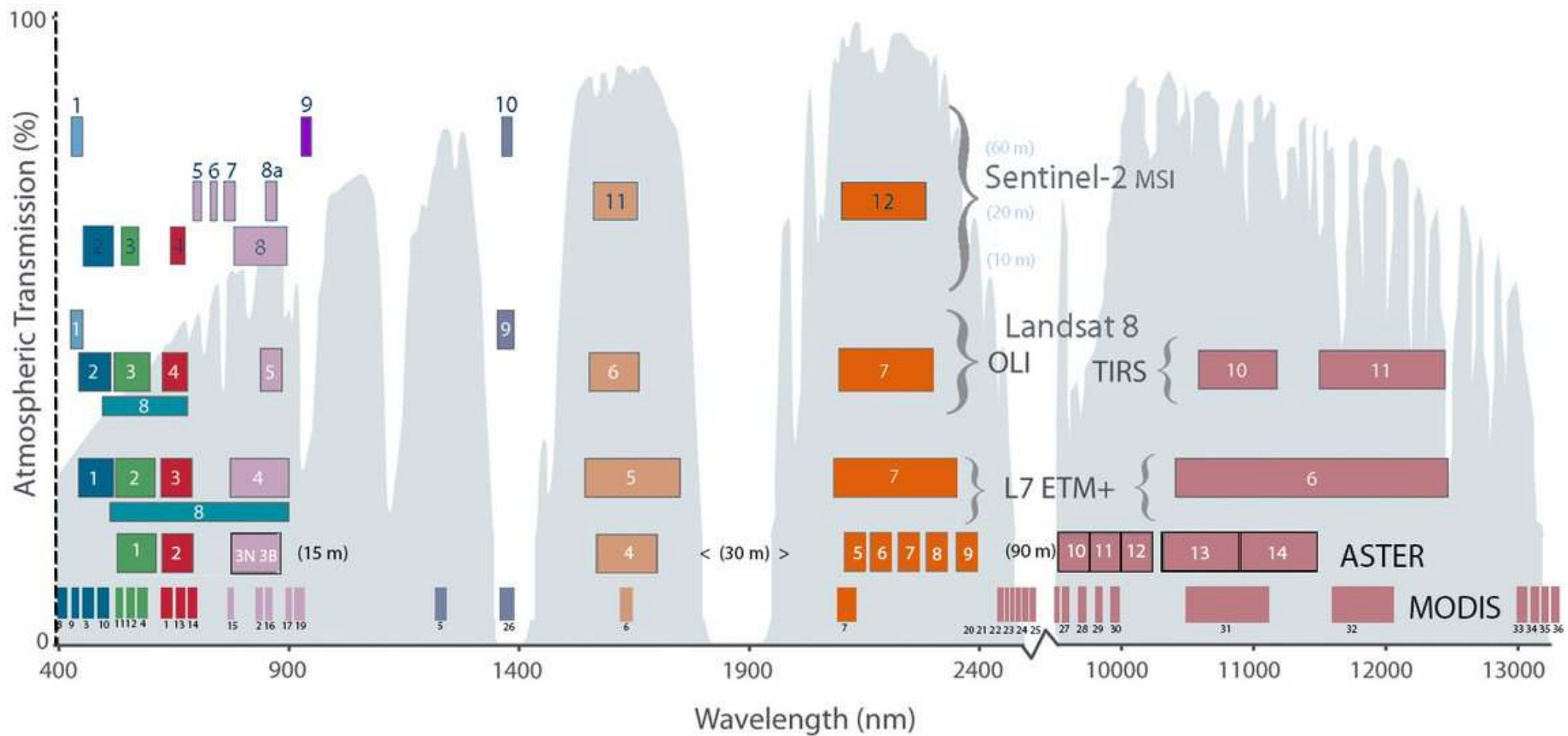
Sensor activo



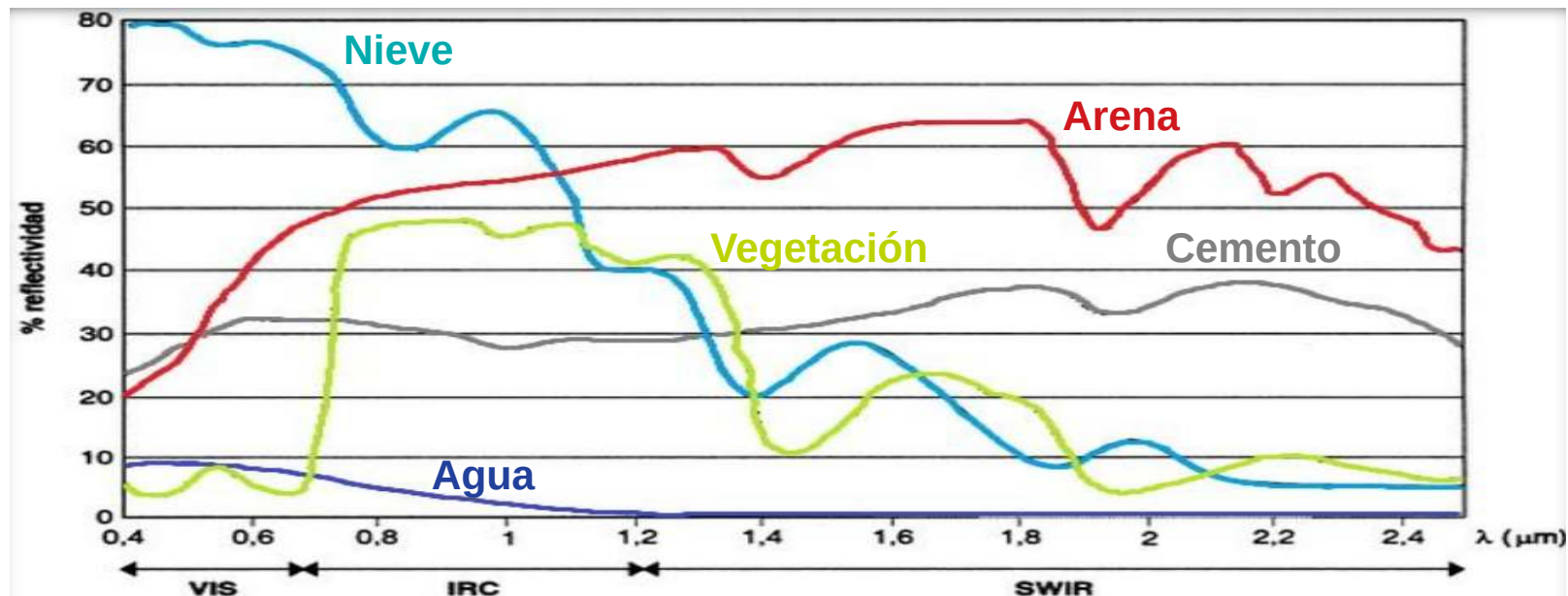
Sensores en el espectro electromagnético



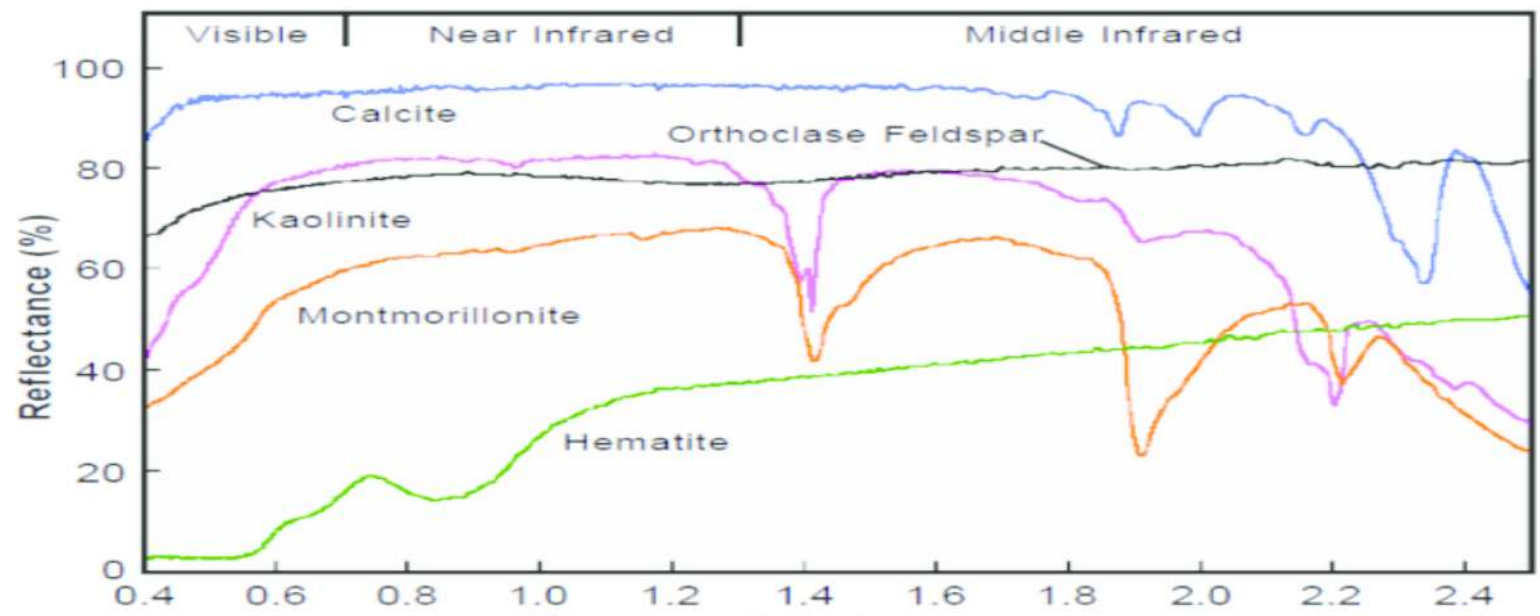
Resolución espectral: Bandas en Sentinel-2, Landsat 7 y 8, ASTER y MODIS



Firmas espectrales

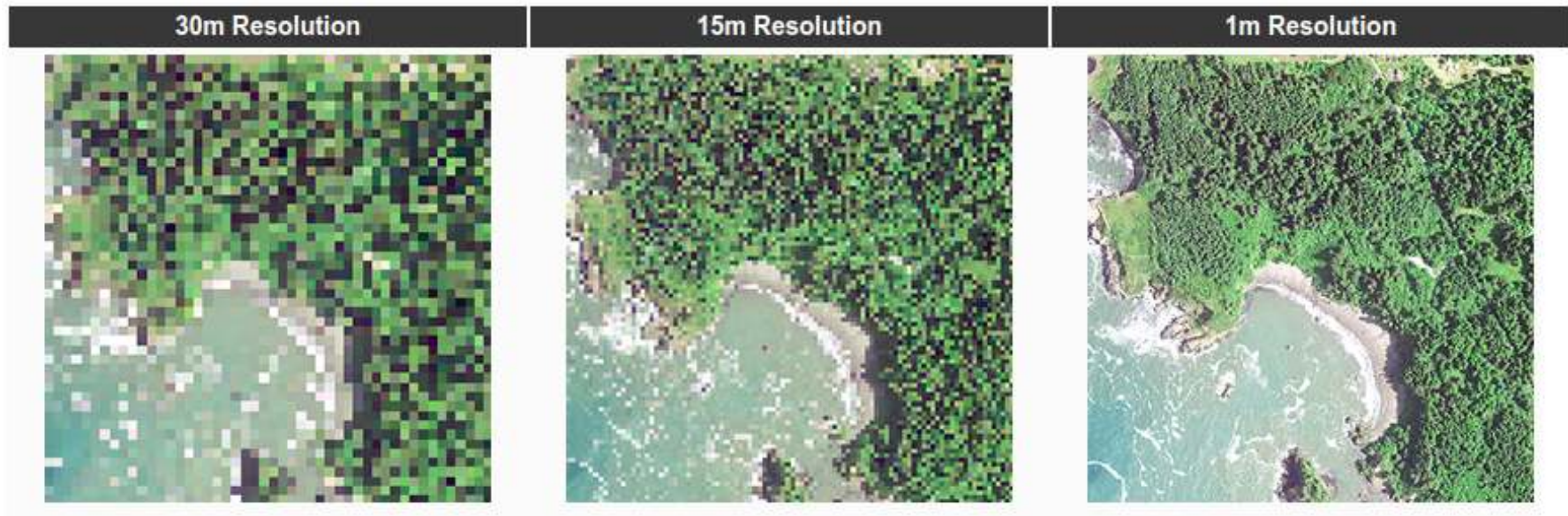


Fuente: Chuvieco, 2008



Fuente: Mendoza, 2014

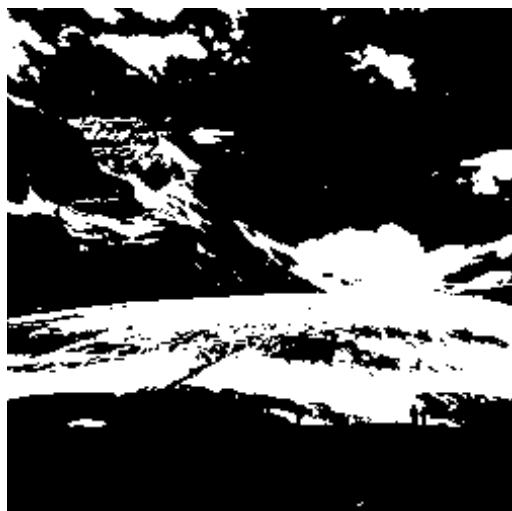
Resolución espacial



Cada pixel de Landsat 8 tiene 30m x 30m

Sensor	Spatial Resolution
MODIS	250-1000 m
Landsat MSS	60 m
Landsat TM, ETM+, OLI	30 m
Sentinel-2	10-60 m
ASTER	15 m
Digital Globe	0.3 m (30 cm)

Resolución radiométrica



1 bit
 $2^1 = 2$ niveles



2 bit
 $2^2 = 4$ niveles



8 bit
 $2^8 = 256$ niveles

Meteosat:	8 bits.	2^8	=	256 niveles
Landsat 7 ETM+:	8 bits.	2^8	=	256 niveles
SPOT 5 HRV:	8 bits.	2^8	=	256 niveles
MODIS:	10 bits.	2^{10}	=	1.024 niveles
IKONOS:	11 bits.	2^{11}	=	2.048 niveles
Landsat 8 OLI:	16 bits.	2^{16}	=	65.536 niveles
Sentinel 2 MSI:	16 bits.	2^{16}	=	65.536 niveles

Resolución temporal



Se refiere a la periodicidad con la que se adquieren imágenes de la misma porción de la superficie terrestre. El ciclo de cobertura está en función de las características orbitales de la plataforma (altura, velocidad, inclinación), así como del diseño del sensor.

- *Alta resolución temporal: < 1 día - 3 días*
- *Media resolución temporal: 4 - 16 días*
- *Baja resolución temporal: > 16 días*

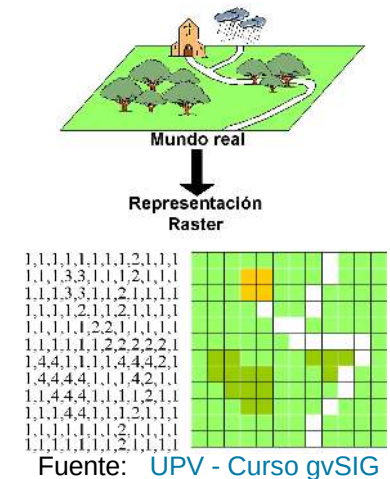
Meteosat:	15 minutos
Landsat 7 ETM+:	16 días
SPOT 5 HRV:	2-3 días
MODIS:	12 horas
Landsat 8 OLI:	16 días
Sentinel 2 MSI:	5 días

Principales paquetes de R para Teledetección

raster	<i>Geographic Data Analysis and Modeling</i>
gdalcubes	<i>Earth Observation Data Cubes from Satellite Image Collections</i>
sen2r	<i>Find, Download and Process Sentinel-2 Data</i>
landsat	<i>Radiometric and topographic correction of satellite imagery</i>
satellite	<i>Handling and Manipulating Remote Sensing Data</i>
rstoolbox	<i>Tools for Remote Sensing Data Analysis</i>
RGISTools	<i>Handling Multiplatform Satellite Images</i>
MODIS	<i>Acquisition and Processing of MODIS Products</i>
hsdar	<i>Manage, Analyse and Simulate Hyperspectral Data</i>
rLiDAR	<i>LiDAR Data Processing and Visualization</i>

El paquete *raster*

- Implementa funciones básicas y de alto nivel para datos geográficos con representación *ráster*:
Lectura, escritura, manipulación, análisis y modelado.



- Admite el procesamiento de archivos muy grandes.
- Da soporte también a operaciones sobre datos vectoriales como las intersecciones. Depende del paquete *sp*.
- Es un *paquete básico* que importan y del que dependen una gran cantidad de paquetes de R, entre otros: Rstoolbox, satellite, leaflet, mapview, MODIS, RGIStools, ...

El paquete *raster*

- Define tres clases principales:
 - **RasterLayer** - representa datos ráster de una sola capa (variable)
 - **RasterStack** - colección de objetos RasterLayer con la misma extensión y resolución espacial, que puede tener varios ficheros origen
 - **RasterBrick** – es un verdadero objeto ráster multicapa, está vinculado a un único fichero

```
library(raster)
```

RasterLayer

```
## Creamos una lista de ficheros de imágenes con extensión tiff  
lista_tiff = list.files('data', pattern = '.tiff', full.names = TRUE)  
  
## Creamos un rasterLayer con la primera de las imágenes  
(L8B1 = raster(lista_tiff[1]))
```

```
class      : RasterLayer  
dimensions : 985, 1283, 1263755  (nrow, ncol, ncell)  
resolution : 29.99692, 29.99395  (x, y)  
extent     : -685640.1, -647154.1, 4478867, 4508411  (xmin, xmax, ymin, ymax)  
crs        : +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +no_defs  
source     : /media/usu1/DATA/Formacion/SevillaR/Teledeteccion_con_R/data/2019-09-27, Landsat 8 USGS, B01.tiff  
names      : X2019.09.27._Landsat_8_USGS._B01
```

El paquete *raster*

```
## Creamos un reasterStack con la lista de imágenes  
(L8RS = stack(lista_tiff))
```

RasterStack

```
class      : RasterStack  
dimensions : 985, 1283, 1263755, 9  (nrow, ncol, ncell, nlayers)  
resolution : 29.99692, 29.99395  (x, y)  
extent     : -685640.1, -647154.1, 4478867, 4508411 (xmin, xmax, ymin, ymax)  
crs        : +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m  
+nadgrids=@null +no_defs  
names      : X2019.09._//_USGS._B01, X2019.09._//_USGS._B02, X2019.09._//_USGS._B03, X2019.09._//_USGS._B04,  
X2019.09._//_USGS._B05, X2019.09._//_USGS._B06, X2019.09._//_USGS._B07, X2019.09._//_USGS._B08,  
X2019.09._//_USGS._B09
```

```
## Creamos un rasterBrick a partir del rasterStack anterior  
(L8RB = brick(L8RS))
```

RasterBrick

```
class      : RasterBrick  
dimensions : 985, 1283, 1263755, 9  (nrow, ncol, ncell, nlayers)  
resolution : 29.99692, 29.99395  (x, y)  
extent     : -685640.1, -647154.1, 4478867, 4508411 (xmin, xmax, ymin, ymax)  
crs        : +proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m  
+nadgrids=@null +no_defs  
source     : /tmp/Rtmpinjn0Q/raster/r_tmp_2019-11-12_094815_7405_52060.grd  
names      : X2019.09._//_USGS._B01, X2019.09._//_USGS._B02, X2019.09._//_USGS._B03, X2019.09._//_USGS._B04,  
X2019.09._//_USGS._B05, X2019.09._//_USGS._B06, X2019.09._//_USGS._B07, X2019.09._//_USGS._B08,  
X2019.09._//_USGS._B09  
min values :           0.25950,           0.18800,           0.11900,           0.04025,  
           0.00400,           0.00000,           0.00000,           0.10225,  
0.00000  
max values :           1.39275,           1.52025,           1.66450,           1.83375,  
           1.97725,           3.27825,           3.22825,           3.23550,  
1.00000X2019.09._//_USGS._B09
```

El paquete *raster*

- Define muchos **métodos y funciones**, entre otros:
 - Álgebra ráster y funciones de superposición: `+`, `-`, `mosaic`, `crop`, `proj`...
 - Distancia, vecindario y funciones de parche: `focal`, `clump`, `Moran`, ...
 - Conversión vectorial - ráster: `rasterize`, `rasterToContour`, ...
 - Resumen de valores ráster: `cellStats`, `summary`, `crosstab`, `zonal`, ...
 - Acceso a valores de celdas ráster: `as.matrix`, `extract`, `sampleRandom`,...
 - Mapas y gráficos: `plot`, `plotRGB`, `hist`, `density`, ...
 - Manipulación de propiedades: `names`, `extent`, `NAvalue`, `res`, `origin`, ...
 - Filas, columnas en coordenadas y viceversa: `rowFromY`, `xFromCol`, ...
 - Escribir varios tipos de archivos ráster

```
## Salvamos a disco en fichero tif multicapa
```

```
writeRaster(L8RB,'data/20190927_L8.tif',format="GTiff", options=c("COMPRESS=DEFLATE"),  
            overwrite=TRUE)
```

```
(L8RB <- brick('data/20190927_L8.tif')) # Leemos desde disco
```

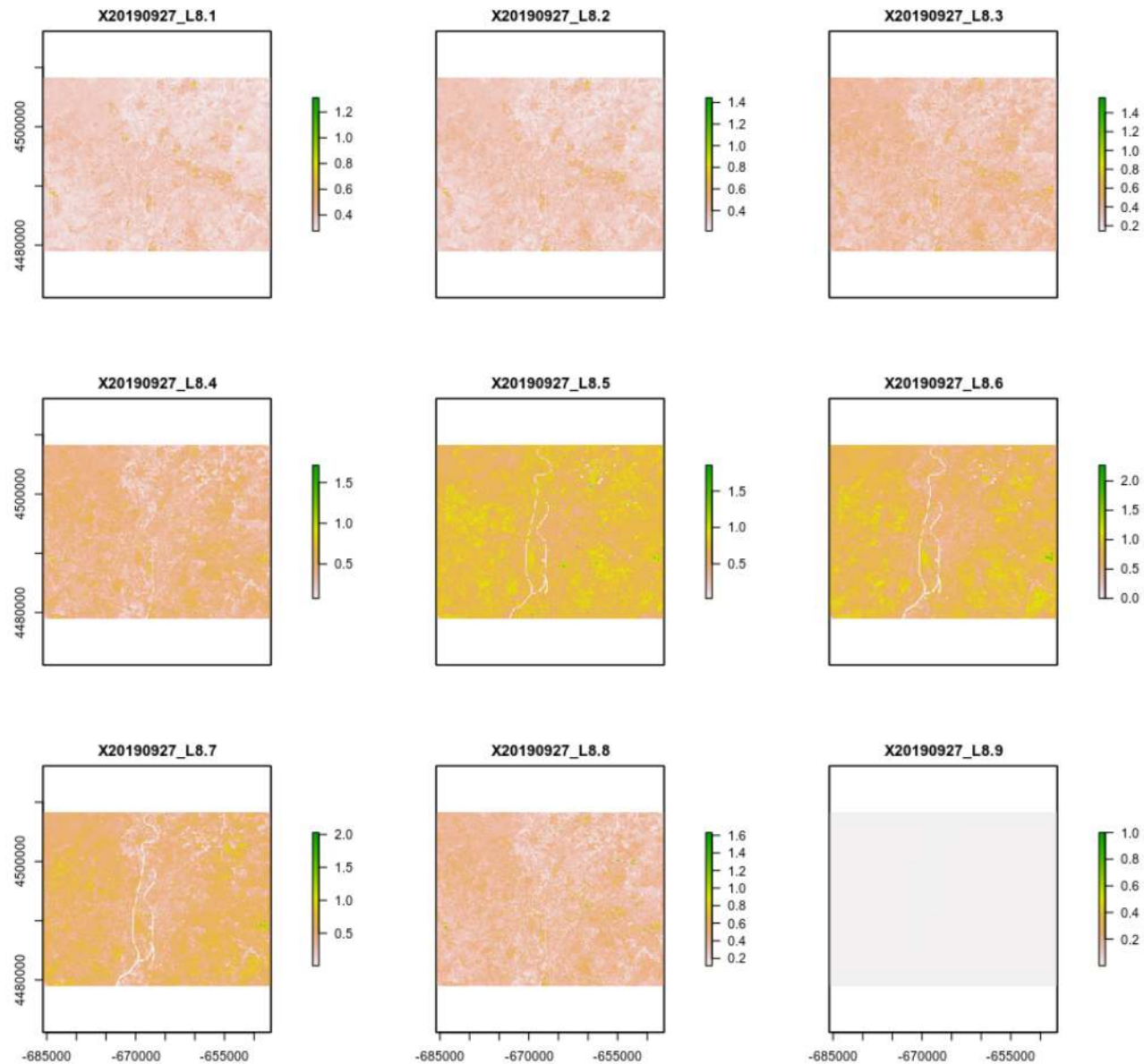
```
removeTmpFiles(h=0.01) # limpieza de temporales
```

```
L8RB
```

```
class       : RasterBrick  
dimensions  : 985, 1283, 1263755, 9  (nrow, ncol, ncell, nlayers)  
resolution  : 29.99692, 29.99395  (x, y)  
extent      : -685640.1, -647154.1, 4478867, 4508411  (xmin, xmax, ymin, ymax)  
crs         : +proj=merc +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 +a=6378137 +b=6378137 +units=m +no_defs  
source      : /media/usu1/DATA/Formacion/SevillaR/Teledeteccion_con_R/data/20190927_L8.tif  
names       : X20190927_L8.1, X20190927_L8.2, X20190927_L8.3, X20190927_L8.4, X20190927_L8.5, X20190927_L8.6, X20190927_L8.7, X20190927_L8.8, X20190927_L8.9  
min values  :      0.25950,      0.18800,      0.11900,      0.04025,      0.00400,      0.00000,      0.00000,      0.10225,      0.00000  
max values  :      1.39275,      1.52025,      1.66450,      1.83375,      1.97725,      3.27825,      3.22825,      3.23550,      1.00000
```

El paquete *raster*

```
## Ploteo simple  
plot(L8RB)
```



El paquete *raster*

```
## Composiciones RGB  
### Color natural  
plotRGB(L8RB,4,3,2,stretch='lin')
```



El paquete *raster*

```
### r=SWIR-2, g=SWIR-1, b=Rojo - Falso color  
plotRGB(L8RB,7,6,4,stretch='lin')
```



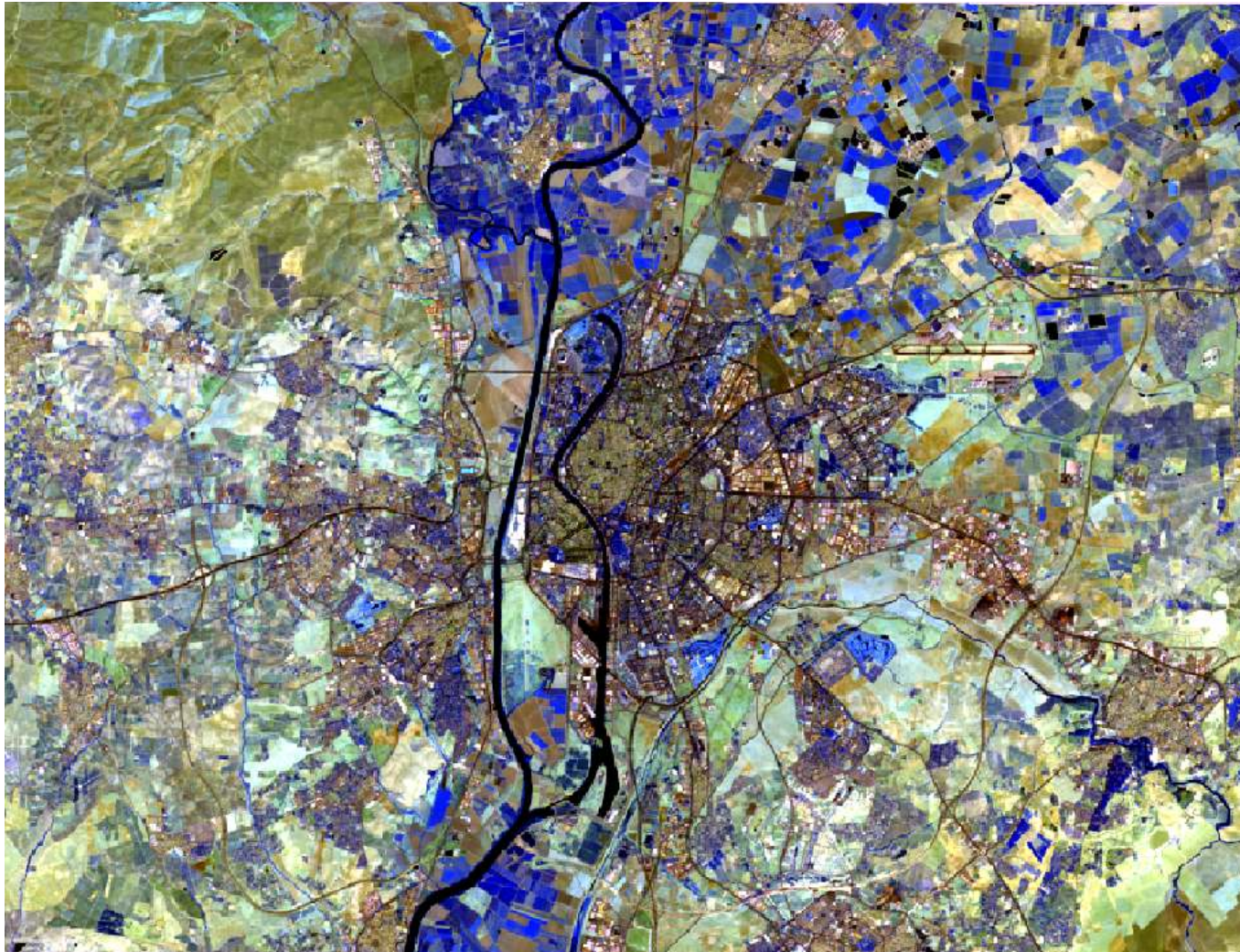
El paquete *raster*

```
### r=NIR, g=Rojo, b=Verde  
plotRGB(L8RB,5,4,3,stretch='lin')
```



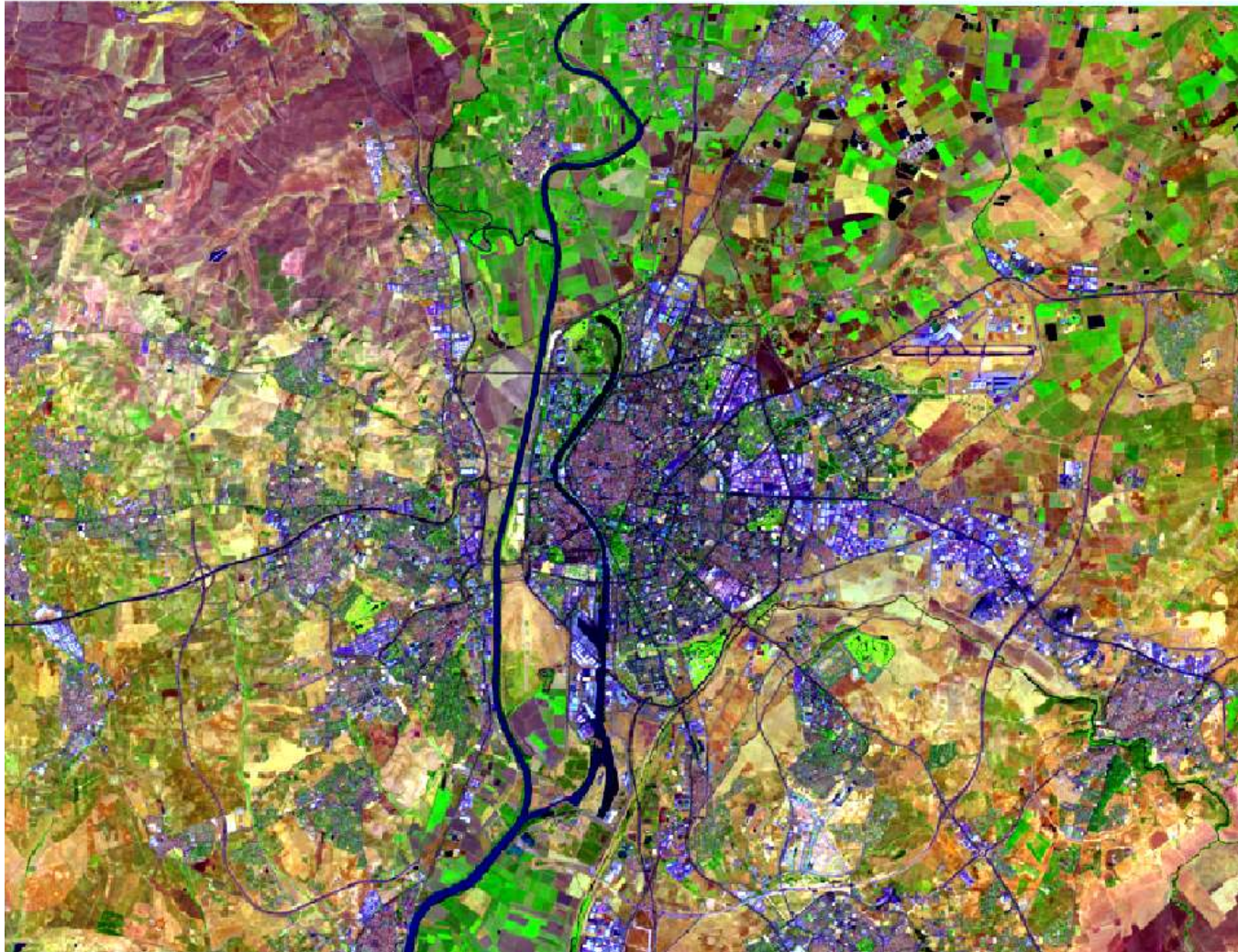
El paquete *raster*

```
### r=SWIR-2, g=SWIR-1, b=IR  
plotRGB(L8RB,7,6,5,stretch='lin')
```



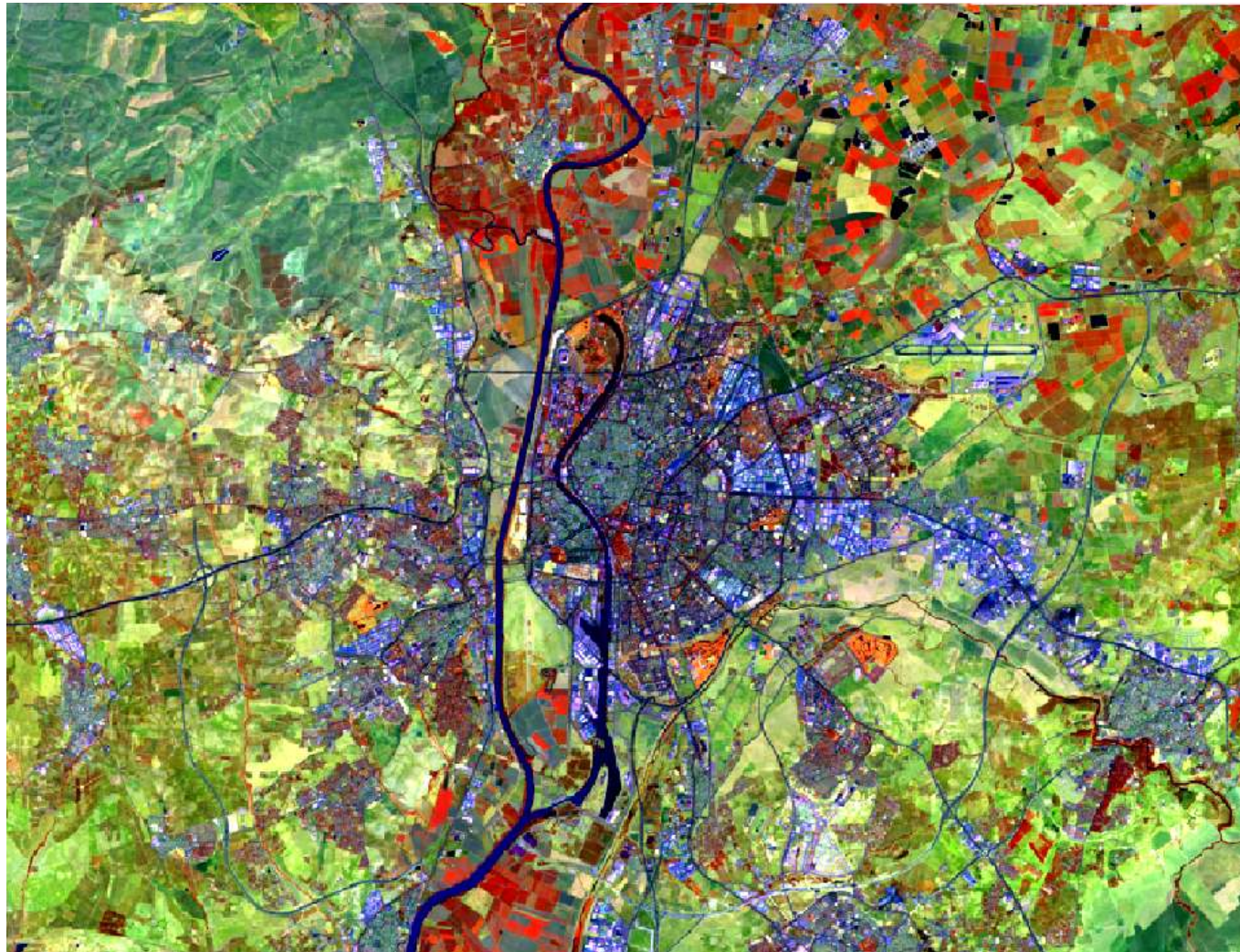
El paquete *raster*

```
### r=SWIR-1, g=NIR-1, b=Azul  
plotRGB(L8RB,6,5,2,stretch='lin')
```



El paquete *raster*

```
### r=NIR, g=SWIR-1, b=Azul  
plotRGB(L8RB,5,6,2,stretch='lin')
```



El paquete *raster*

Componentes principales

```
cp = princomp(as.matrix(L8RB))  
summary(cp)
```

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4	..
Standard deviation	0.3136866	0.1416801	0.10799009	0.05782457	..
Proportion of Variance	0.7226152	0.1474119	0.08564123	0.02455500	..
Cumulative Proportion	0.7226152	0.8700272	0.95566838	0.98022338	..

Puntuación de los pixels en las 3 primeras componentes,

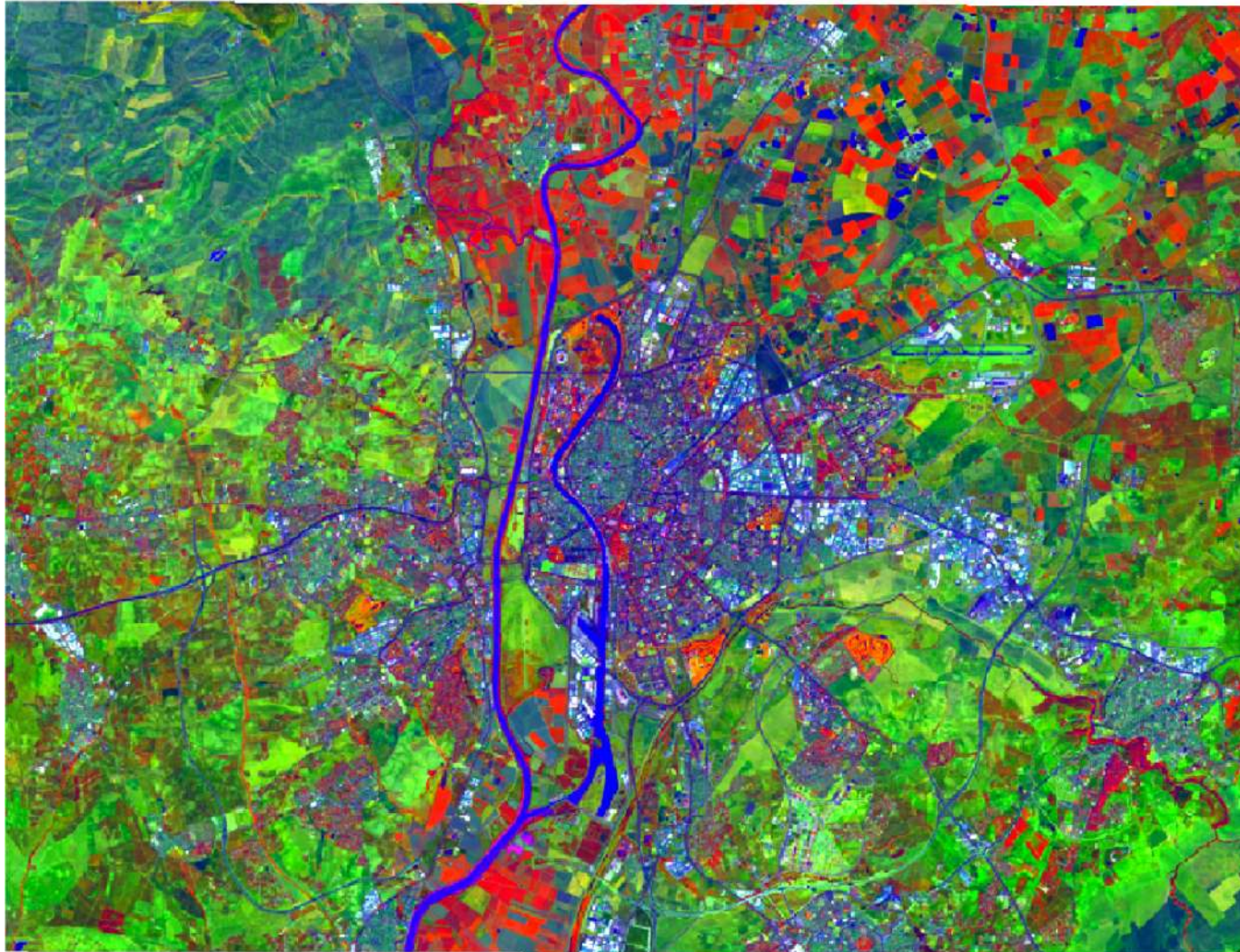
nuevo ráster con las puntuaciones

```
(L8RBcp = predict(L8RB,cp,index=1:3))
```

```
class      : RasterBrick  
dimensions : 985, 1283, 1263755, 3  (nrow, ncol, ncell, nlayers)  
resolution : 29.99692, 29.99395  (x, y)  
extent     : -685640.1, -647154.1, 4478867, 4508411  (xmin, xmax, ymin, ymax)  
crs        : +proj=merc +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 +a=6378137 +b=6378137 +units=m +no_defs  
source     : memory  
names      :      Comp.1,      Comp.2,      Comp.3  
min values : -1.1536160, -1.2592418, -0.8518881  
max values :  4.157295,  1.292503,  1.391324
```


El paquete *raster*

```
### Composición RGB con las 3 primeras componentes principales  
### r=cp3, g=cp1, b=cp2  
plotRGB(L8RBcp,3,1,2,stretch='lin')
```



El paquete *raster*

```
### Clasificación no supervisada con k-means, 10 clases
(km = kmeans(as.data.frame(L8RB),10))
```

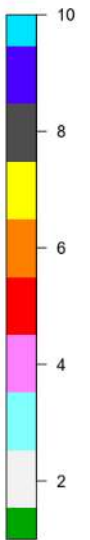
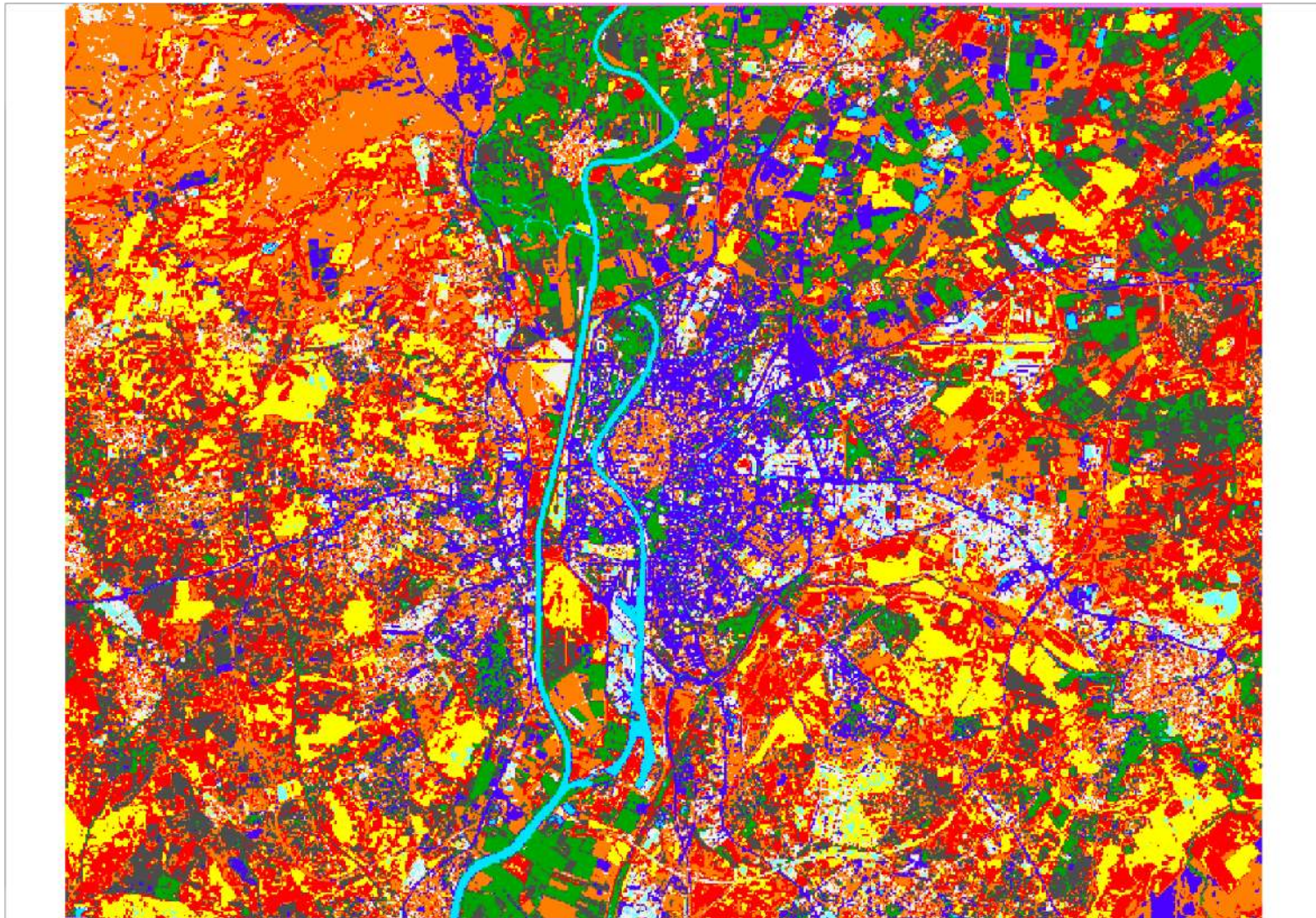
```
K-means clustering with 10 clusters of sizes
98288, 107243, 5737, 114750, 224447, 176679, 287870, 19488, 211247, 18006
:
Within cluster sum of squares by cluster:
[1] 3041.0539 3100.5904 266.0580 2796.3272 3392.6397 4592.0686 4809.4237 3517.9653 3698.6559 821.9051
(between_SS / total_SS = 82.5 %)
```

```
### raster base
L8RBkm = raster(L8RB,1)
### asignamos valores del resultado km y nombre de capa
values(L8RBkm) = km$cluster
names(L8RBkm) = 'Clase'
L8RBkm
```

```
class      : RasterLayer
band       : 1 (of 9 bands)
dimensions : 985, 1283, 1263755 (nrow, ncol, ncell)
resolution : 29.99692, 29.99395 (x, y)
extent     : -685640.1, -647154.1, 4478867, 4508411 (xmin, xmax, ymin, ymax)
crs       : +proj=merc +lon_0=0 +lat_ts=0 +x_0=0 +y_0=0 +a=6378137 +b=6378137 +units=m +no_defs
source    : memory
names     : Clase
values    : 1, 10 (min, max)
```

El paquete *raster*

```
### Mapa con resultados de la clasificación  
plot(L8RBkm,  
col=c(terrain.colors(2),cm.colors(2),heat.colors(3),gray.colors(1),topo.colors(2)))
```



El paquete **gdalcubes**

- *Permite el procesamiento de colecciones y series temporales de imágenes de observación de la Tierra como **cubos de datos ráster** multiespectrales a demanda.*

*Los usuarios definen cubos por **extensión espacio-temporal**, **resolución** y **sistema de referencia espacial** y dejan que 'gdalcubes' aplique automáticamente el recorte, la reproyección y el remuestreo utilizando la 'librería de abstracción de datos geoespaciales' ('GDAL').*

- *Las **funciones** implementadas en los cubos de datos incluyen la **reducción en el espacio y el tiempo**, la aplicación de **expresiones aritméticas** en los valores de la banda de píxeles, **agregados focales en ventana temporal**, el **filtrado por espacio, tiempo, bandas** y predicados sobre los valores de los píxeles, la **exportación** de cubos de datos como archivos '**netCDF**' o '**GeoTIFF**' y **ploteo**.*

El paquete *gdalcubes*

```
library(gdalcubes)
```

```
## Crea colección
```

```
L8.col = create_image_collection(files, "L8_SR", "L8.db")
```

```
L8.col
```

```
## A GDAL image collection object, referencing 180 images with 17 bands
## Images:
##               name      left      top      bottom
## 1 LC08_L1TP_226063_20140719_20170421_01_T1 -54.15776 -3.289862 -5.392073
## 2 LC08_L1TP_226063_20140820_20170420_01_T1 -54.16858 -3.289828 -5.392054
## 3 LC08_L1GT_226063_20160114_20170405_01_T2 -54.16317 -3.289845 -5.392064
## 4 LC08_L1TP_226063_20160724_20170322_01_T1 -54.16317 -3.289845 -5.392064
## 5 LC08_L1TP_226063_20170609_20170616_01_T1 -54.17399 -3.289810 -5.392044
## 6 LC08_L1TP_226063_20170711_20170726_01_T1 -54.15506 -3.289870 -5.392083
##               right      datetime
## 1 -52.10338 2014-07-19T00:00:00
## 2 -52.11418 2014-08-20T00:00:00
## 3 -52.10878 2016-01-14T00:00:00
## 4 -52.10878 2016-07-24T00:00:00
## 5 -52.11958 2017-06-09T00:00:00
## 6 -52.09798 2017-07-11T00:00:00
##                               srs
## 1 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## 2 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## 3 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## 4 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## 5 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## 6 +proj=utm +zone=22 +datum=WGS84 +units=m +no_defs
## [ omitted 174 images ]
## :
:
```

El paquete *gdalcubes*

Crea una vista del cubo para chequeo a baja resolución

```
v.overview = cube_view(extent=L8.col, dt="P1Y", dx=1000, dy=1000,  
                        srs="EPSG:3857", aggregation = "median", resampling = "bilinear")  
raster_cube(L8.col, v.overview)
```

A GDAL data cube proxy object

Dimensions:

##	name	low	high	size	chunk_size
## 1	t	2013.0	2019.0	7	16
## 2	y	-764014.4	-205014.4	559	256
## 3	x	-6582280.1	-5799280.1	783	256

##

Bands:

##		name	type	offset	scale	nodata	unit
## 1	AEROSOL	float64	0	1	NaN		
## 2	B01	float64	0	1	NaN		
## 3	B02	float64	0	1	NaN		
## 4	B03	float64	0	1	NaN		
## 5	B04	float64	0	1	NaN		
## 6	B05	float64	0	1	NaN		
## 7	B06	float64	0	1	NaN		
## 8	B07	float64	0	1	NaN		
## 9	EVI	float64	0	1	NaN		
## 10	MSAVI	float64	0	1	NaN		
## 11	NBR	float64	0	1	NaN		
## 12	NBR2	float64	0	1	NaN		
## 13	NDMI	float64	0	1	NaN		
## 14	NDVI	float64	0	1	NaN		
## 15	PIXEL_QA	float64	0	1	NaN		
## 16	RADSAT_QA	float64	0	1	NaN		
## 17	SAVI	float64	0	1	NaN		

El paquete *gdalcubes*

Crea una vista del cubo de una region más pequeña con paso mensual

```
v.subarea = cube_view(extent=list(left=-6320000, right=-6220000, bottom=-6000000,
                                top=-500000, t0="2014-01-01", t1="2018-12-31"),
                      dt="P1M", dx=50, dy=50, srs="EPSG:3857",
                      aggregation = "median", resampling = "bilinear")
raster_cube(L8.col, v.subarea)
```

A GDAL data cube proxy object

Dimensions:

##	name	low	high	size	chunk_size
## 1	t	201401	201812	60	16
## 2	y	-6000000	-5000000	2000	256
## 3	x	-6320000	-6220000	2000	256

##

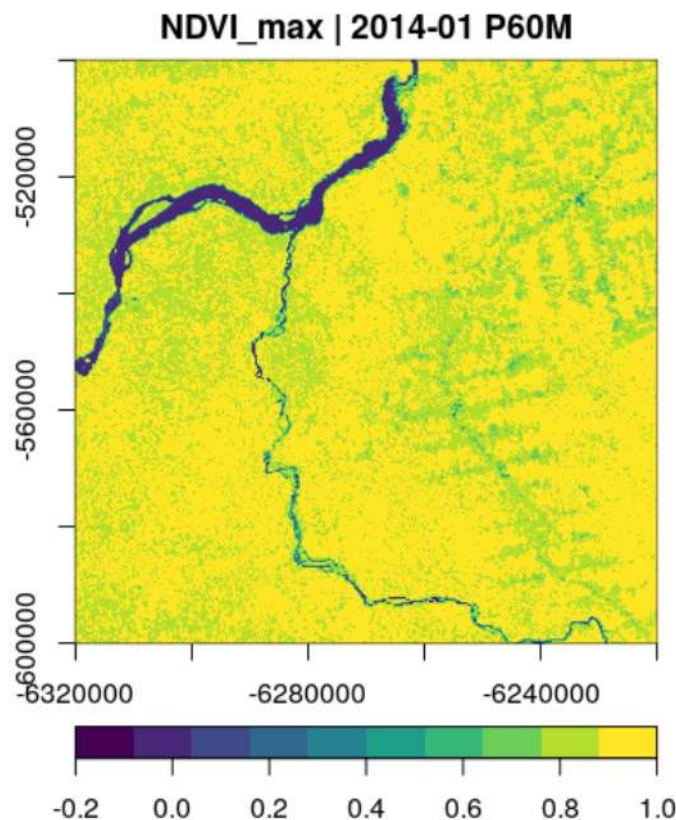
Bands:

##	name	type	offset	scale	nodata	unit
## 1	AEROSOL	float64	0	1	NaN	
## 2	B01	float64	0	1	NaN	
## 3	B02	float64	0	1	NaN	
## 4	B03	float64	0	1	NaN	
## 5	B04	float64	0	1	NaN	
## 6	B05	float64	0	1	NaN	
## 7	B06	float64	0	1	NaN	
## 8	B07	float64	0	1	NaN	
## 9	EVI	float64	0	1	NaN	
## 10	MSAVI	float64	0	1	NaN	
## 11	NBR	float64	0	1	NaN	
## 12	NBR2	float64	0	1	NaN	
## 13	NDMI	float64	0	1	NaN	
## 14	NDVI	float64	0	1	NaN	
## 15	PIXEL_QA	float64	0	1	NaN	
## 16	RADSAT_QA	float64	0	1	NaN	
## 17	SAVI	float64	0	1	NaN	

El paquete *gdalcubes*

Máximo valor NDVI en la región y periodo

```
raster_cube(L8.col, v.subarea) %>%  
  select_bands(c("B04","B05")) %>%  
  apply_pixel(c("(B05-B04)/(B05+B04)"), names="NDVI") %>%  
  reduce_time("max(NDVI)") %>%  
  plot(zlim=c(-0.2,1), col=viridis, key.pos=1)
```



El paquete **sen2r**

- *Proporciona funciones para descargar imágenes ópticas Sentinel-2 y realizar operaciones de procesamiento preliminares.*
- *También se proporciona una interfaz gráfica de usuario para facilitar el procesamiento de datos.*

```
library(sen2r)
```

```
# Lanza interfaz gráfica  
sen2r()
```



Product selection

Spatio-temporal selection

Processing options

Spectral indices selection

RGB images selection

Save options as...

Load options

✓ Launch processing

⏸ Close without processing

📖 Open documentation

Type of processing

- ☐ Raw files in [raw SAFE format](#) (downloaded and/or corrected with sen2cor)
- ☒ Processed spatial files (surface reflectance, spectral indices, ...) in custom format

Select products and sensors

Select products:

- ☐ TOA (top-of-atmosphere) Reflectance
- ☒ BOA (bottom-of-atmosphere) Surface Reflectance
- ☐ SCL (surface classification map)
- ☒ Spectral indices
- ☒ RGB images

SAFE levels needed:

[Level-1C](#)

[Level-2A](#)

Select sensors:

- ☒ Sentinel-2A
- ☒ Sentinel-2B

SAFE options

Directory for level-1C SAFE products:

Select

Directory for level-2A SAFE products:

Select

Download mode ?

- ☒ Online
- ☐ Offline

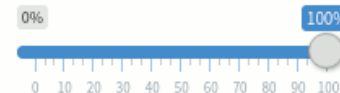
Downloader ?

- ☒ Wget ☐ aria2

Overwrite existing SAFE products? ?

- ☐ Yes ☒ No

Max. SAFE cloud cover ?



👤 Login in SciHub

Delete raw SAFE files after processing?

- ☐ Yes ☐ Only level-1C ☒ No

Atmospheric correction options

Method to obtain level-2A corrected images ?

- ☐ Use only level-2A images available locally or online
- ☒ Use sen2cor only for level-2A products not available locally or online
- ☐ Always correct level-1C images with sen2cor locally



Product selection

Spatio-temporal selection

Processing options

Spectral indices selection

RGB images selection

Save options as...

Load options

✓ Launch processing

⏸ Close without processing

📖 Open documentation

Temporal parameters

Use temporal filter?

☒ Yes ☐ No (process all the input SAFE images)

Time interval

2018-01-01

to

2018-12-31

Time period type ?

☒ Full ☐ Seasonal

Spatial extent

Use spatial filter/clip?

☒ Yes ☐ No (process all the input SAFE images)

Specify the extent:

Set a bounding box

Load a vector file

Draw it on the map

Tiles selected

32TNS

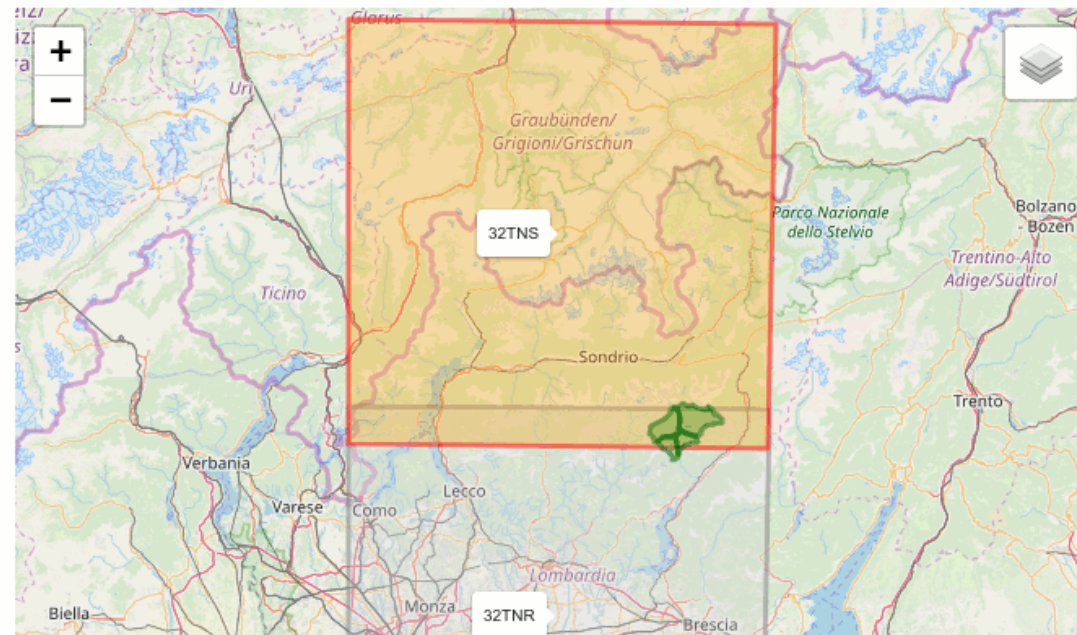
Orbits selected ?

All overlapping orbits

Extent name: ?

sen2r

Refresh the map





Product selection

Spatio-temporal selection

Processing options

Spectral indices selection

RGB images selection

Save options as...

Load options

Launch processing

Close without processing

Open documentation

Output files

Directory for output processed products:

Select

☒ Group products in subdirectories ?

☒ Create thumbnails ?

Overwrite existing outputs

☐ Yes (reprocess all)

☒ No (skip processing if outputs exist)

Output file format

GeoTiff

Output compression

High (deflate)

Output extent

Clip outputs on the selected extent? ?

☒ Yes ☐ No

Mask data outside the selected polygons?

☐ Yes ☒ No

Save single tiles?

☐ Yes ☒ No

Save merged tiles?

☐ Yes ☒ No

Cloud mask

Mask cloud-covered pixels? ?

☐ Yes ☒ No

Output geometry

Use an existing raster to define the output grid?

☐ Yes ☒ No

Spatial resolution

☒ Native ☐ Custom

Specify resolution

☒ 10 metres

☐ 20 metres

☐ 60 metres

Output projection

☒ Input projection (do not reproject)

☐ Custom projection

Resampling method

Nearest neighbour



Product selection

Spatio-temporal selection

Processing options

Spectral indices selection

RGB images selection

Save options as...

Load options

✓ **Launch processing**

⊘ Close without processing

📘 Open documentation

Spectral indices selection

Directory for spectral indices:

Select

The same used for output products

Build indices from: ?

☒ BOA ☐ TOA

Data type

Integer (16 bits)

Filter indices

☒ Show only verified indices ⚠

Indices to be exported

- ☒ MSAVI2 (Modified Soil Adjusted Vegetation Index) ✓
- ☒ NDVI (Normalized Difference NIR/Red Normalized Difference Vegetation Index, Calibrated NDVI - CDVI) ✓
- ☐ ARVI (Atmospherically Resistant Vegetation Index) ✓
- ☐ BNDVI (Normalized Difference NIR/Blue Blue-normalized difference vegetation index) ✓
- ☐ CSI (Char Soil Index) ✓
- ☐ EVI2 (Enhanced Vegetation Index 2 -2) ✓
- ☐ GNDVI (Green Normalized Difference Vegetation Index) ✓
- ☐ MCARI (Modified Chlorophyll Absorption in Reflectance Index) ✓
- ☐ MCARI2 (Modified Chlorophyll Absorption in Reflectance Index 2) ✓
- ☐ MCARI:MTVI2 (MCARI/MTVI2) ✓
- ☐ MIRBI (Mid-Infrared Burn Index) ✓
- ☐ MTVI2 (Modified Triangular Vegetation Index 2) ✓
- ☐ NBR (Normalized Difference NIR/SWIR Normalized Burn Ratio) ✓
- ☐ NBR2 (Normalized Burn Ratio 2) ✓
- ☐ NDFI (Normalized Difference Flood Index B1B7) ✓
- ☐ NDFI2 (Normalized Difference Flood Index B1B6) ✓
- ☐ NDII (Normalized Difference 819/1600 NDII) ✓
- ☐ NDRE (Normalized Difference NIR/Rededge Normalized Difference Red-Edge) ✓
- ☐ NDSI (Normalize Difference Snow Index) ✓
- ☐ OSAVI (Optimized Soil Adjusted Vegetation Index) ✓

MSAVI2

Modified Soil Adjusted Vegetation Index

Formula:

$$\frac{2NIR+1-\sqrt{(2NIR+1)^2-8(NIR-RED)}}{2}$$

[More info](#)

NDVI

Normalized Difference NIR/Red
Normalized Difference
Vegetation Index, Calibrated
NDVI - CDVI

Formula:

$$\frac{NIR-RED}{NIR+RED}$$

[More info](#)



Product selection

Spatio-temporal selection

Processing options

Spectral indices selection

RGB images selection

Save options as...

Load options

✓ Launch processing

⌵ Close without processing

i Open documentation



RGB images selection

Directory for RGB images:

Select

The same used for output products

RGB images: ?

☒ RGBb84B:

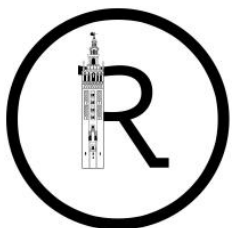
- source: BOA
- red: band 11 (reflectance range 0 – 0.75)
- green: band 8 (reflectance range 0 – 0.75)
- blue: band 4 (reflectance range 0 – 0.25)

☒ RGB432B:

- source: BOA
- red: band 4 (reflectance range 0 – 0.25)
- green: band 3 (reflectance range 0 – 0.25)
- blue: band 2 (reflectance range 0 – 0.25)

+ Define custom RGB image

Remove unselected RGB from list



Sevilla R - Sevilla, 12 de Noviembre de 2019

Teledetección con R.

GRACIAS

Jerónimo Carranza Carranza

jeronimo.carranza@asterionat.com



www.asterionat.com