Modelos 3D en R

Rodrigo Tapia McClung
Centro de Investigación en Ciencias de Información Geoespacial - CentroGeo
rtapia@centrogeo.edu.mx

15 de enero, 2020

Crear modelos 3D en R

R *no* es un sistema de información geográfica (SIG), pero se puede usar para crear mapas muy bonitos, tanto en 2D como en 3D. En esta práctica vamos a usar un modelo digital de elevación para hacer un mapa en 3D.

Los datos para esta práctica los puedes descargar de aquí. Una vez descargados los datos, descomprime el archivo en alguna carpeta en tu computadora.

Primero nos cambiamos al directorio de trabajo en donde está un ASCII Grid de una zona del país (¿la puedes reconocer?):

```
# Cambiar directorio de trabajo:
setwd("C:/Descargas/escuela-de-metodos-2020/practica2")
```

Y nos aseguramos de tener las librerías que vamos a usar:

```
if (!require("pacman")) install.packages("pacman")
# para macOS, primero instalar https://www.xquartz.org/
# y ejecutarlo antes de usar plot_3d()
# o usar options(rgl.printRglwidget = TRUE)
pacman::p_load(raster, rayshader)
```

Ahora podemos leer/dibujar el raster:

```
r = raster("dem.asc")
#plot(r)
```

Puedes notar que la parte inferior del raster no tiuene datos. En realidad, todas esas celdas tienen un valor de `NA`. Vamos a reclasificar esas celdas y ponerles un valor de 0:

```
r <- reclassify(r, cbind(NA, 0))
#plot(r)
```

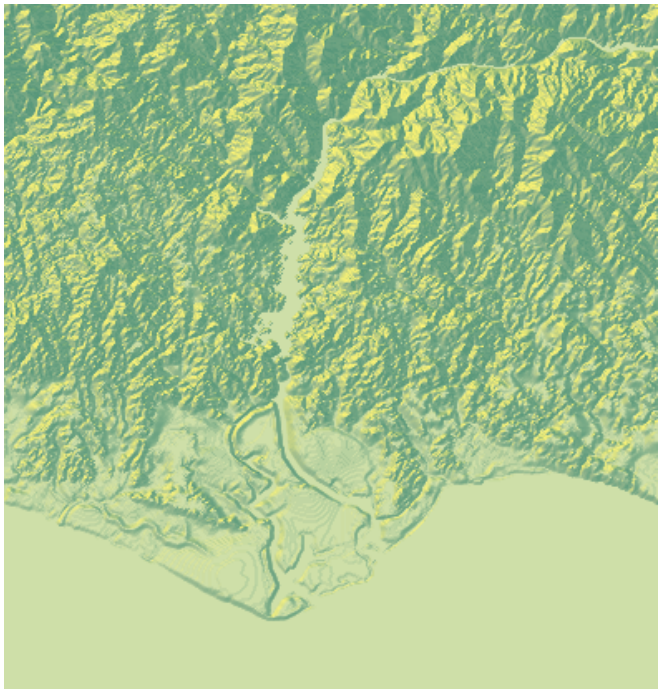
NOTA: en el mejor de los casos podríamos usar datos de batimetría y juntarlos con los del modelo digital de elevación y el resultado sería *mucho* más bonito.

Ahora convertimos ese raster en una matriz:

```
elmat <-  raster_to_matrix(r)
#plot(r)
```

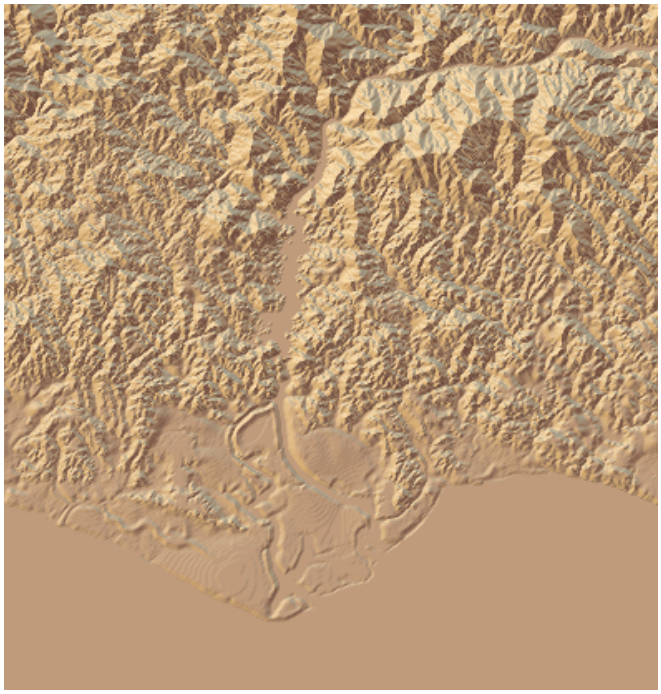
Usando la librería `rayshader`, podemos dibujar este modelo digital de elevación:

```
elmat %>%
  sphere_shade() %>%
  plot_map()
```



Se ve algo feo, así que lo podemos embellecer un poco con solo cambiar el estilo:

```
elmat %>%
  sphere_shade(texture = "desert") %>%
  plot_map()
```



También podemos calcular un "mapa de sombras" y uno de "oclusión ambiental", esto es, una matriz de valores que nos diga dónde hay sombras dependiendo de dónde esté una fuente de luz (el sol) y otra que nos diga qué tan brillante es la luz que brilla en la superficie que tenemos.

```
shadow <-  ray_shade(elmat, zscale = 50, lambert = FALSE, multicore = TRUE)
ambient <-  ambient_shade(elmat, zscale = 50, multicore = TRUE)
```

Y agregar esas matrices a nuestro dibujo:

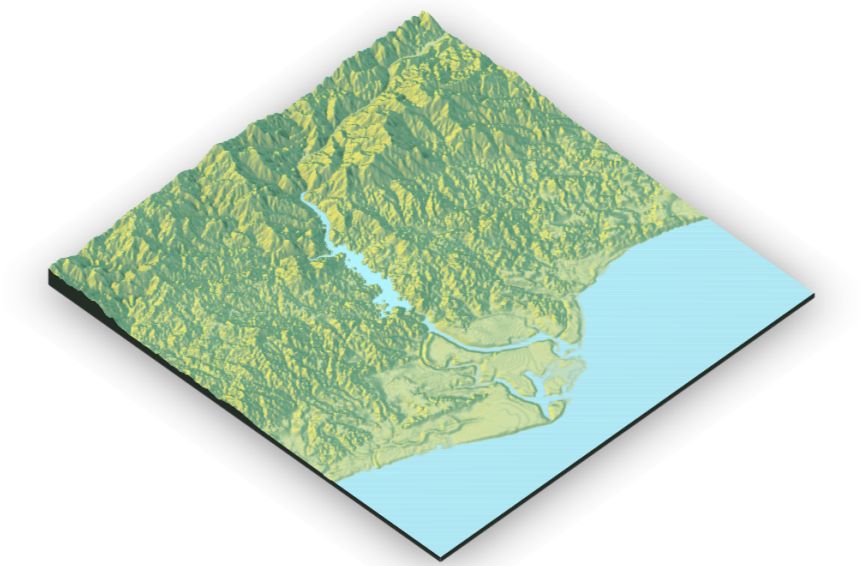
```
elmat %>%
  sphere_shade(texture = "imhof1") %>%
  add_shadow(shadow) %>%
  add_shadow(ambient) %>%
  plot_map()
```

Adicionalmente, podemos usar un algoritmo para detectar agua:

```
water <- detect_water(elmat)
elmat %>%
  sphere_shade(texture = "imhof1") %>%
  add_water(water, color = "desert") %>%
  add_shadow(shadow) %>%
  add_shadow(ambient) %>%
  plot_map()
```

Ahora, para convertir el mapa 2D a 3D, hay que usar la función `plot_3d`. Esto tarda un poco y da como resultado una ventana flotante en donde está la gráfica en 3D:

```
elmat %>%
  sphere_shade(texture = "imhof1") %>%
  add_water(water, color = "desert") %>%
  add_shadow(shadow) %>%
  add_shadow(ambient) %>%
  plot_3d(elmat, zscale = 50, fov = 0, theta = -45, phi = 45,
          windowsize = c(1000, 800), zoom = 0.75,
          water = TRUE, waterdepth = 0, wateralpha = 0.5, watercolor = "lightblue",
          waterlinecolor = "white", waterlinealpha = 0.5)
```



Esta es una gráfica interactiva: puedes hacer zoom in, zoom out, cambiar el campo de visión y rotarla. También puedes "tomarle una foto":

```
Sys.sleep(0.2)
render_snapshot(clear=TRUE)
```