rayshader package Cheat Sheet: Terrain Visualization and 3D plotting

Introduction

rayshader is a package that allows for 2D and 3D visualizations in R. It can create 2D or 3D maps using elevation data, and can even be combined with ggplot2 to generate 3D visualizations. This cheatsheet introduces the common functions that can be used in this package.

Installation

install.packages('rayshader')

Grammar

```
map %>%
        <shade> %>%
        <overlay> %>%
        plot_map() or plot_3d()
<other renders>
```

Synergy with ggplot2

plot_gg() takes in a ggplot2 object and converts it to 3D, mapping fill or color aes to elevation

Map Object

2D matrix where each entry is the elevation at that point

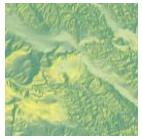
Use the rayshader built-in map "montereybay" or use your own coordinate matrix

raster_to_matrix() converts raster files (TIF extension) to a matrix

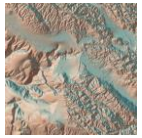
Function to create a map

sphere_shade(): maps an RGB texture to the relief map using built-in or user-defined texture can also be specified using **create_texture()**.

e.g. Textures on built-in montereybay:



'imhof1'



'imhof2'

```
montereybay %>%
        sphere_shade(texture='imhof2') %>%
        plot_map()
```

Other Common Mapping Functions

ray_shade(): takes in argument for light direction and generates a global shadow map

ambient_shade(): makes lower elevation areas (like valleys and ravines) shaded darker than higher elevation areas (ridges and plains)

texture_shade(): directly maps elevation to darkness of shadow, with options to adjust detail, brightness, contrast etc.

height_shade(): elevation to color mapping

add_shadow(): combines two mapping functions

Overlays

detect_water(): detect bodies of water based on user-defined minimum area, maximum height for water to exist.

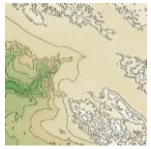
add_water(): uses the output of detect_water() to add water on the map. User can input their own color or use one of the defaults from sphere_shade().

e.g. Adding the ocean to montereybay:

```
montereybay %>%
    sphere_shade(texture='imhof2') %>%
    add_water(detect_water(montereybay,
max_height = 0),color="imhof3") %>%
    plot_map()
```

generate_compass_overlay(): Adds a compass to the map

generate_contour_overlay(): Adds contour lines to map



```
montereybay %>%
    height_shade() %>%
    add_overlay(generate_contour_overlay(montereybay)) %>%
    plot_map()
```

generate_line_overlay(): generates an overlay of lines such as to represent roads

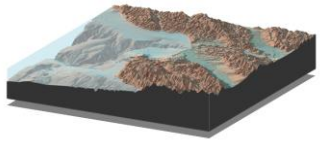
generate_scalebar_overlay(): generates a distance scale bar

Saving and Displaying

plot_map(): plots the current map

write_png(): writes map to specified file name

plot_3d(): plots 3d map, given the height map and a hillshade defined by height_shade()



```
montereybay %>%
    sphere_shade(texture='imhof2') %>%
    plot_3d(montereybay,zscale=50)
render_water(montereybay,zscale=50)
render_snapshot()
```

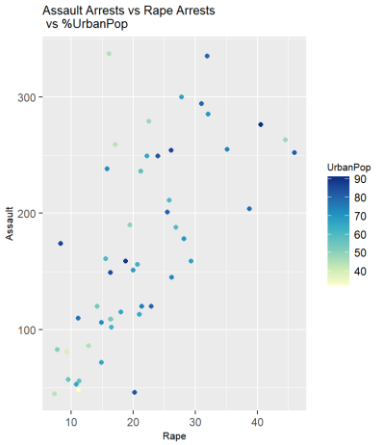
render_camera(): changes angle of camera

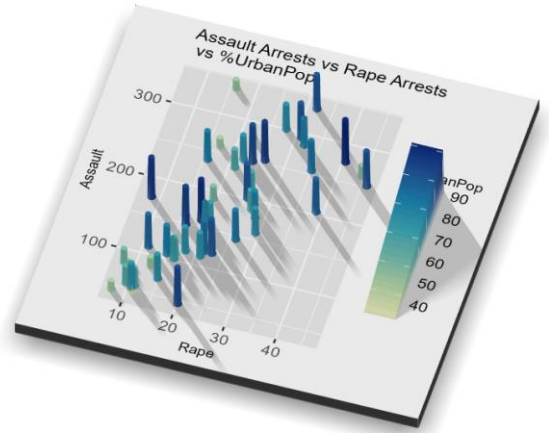
render_snapshot(): display the map on R, or saves to a file

Converting ggplot object to 3D

The rayshader package can also convert ggplot2 objects into 3D. This can be useful in certain applications, when the 2D form of the plot is unable to capture smaller variations between data points.

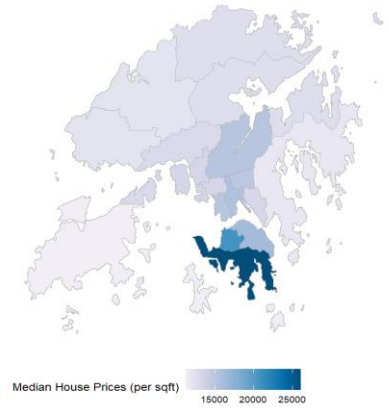
e.g. Scatterplot on R built-in dataset "USArrests"






```
data("USArrests")
gg = ggplot(USArrests, aes(x=Rape, y=Assault, color=UrbanPop)) +
    geom_point()+
    scale_color_distiller(direction=1, palette="YlGnBu")+
    ggtitle('Assault Arrests vs Rape Arrests \n vs %UrbanPop')+
    theme(title=element_text(size=8))
gg %>% plot_gg()
```

e.g. Geospatial Visualization – Median House Prices in Hong Kong



3D visualizations are generally discouraged. However, they can serve their purpose when used in an interactive environment.

Source/Documentation: https://www.rayshader.com/