

rayshader CHEAT SHEET

R Package for 2D and 3D mapping and data visualization



ggplot charts in 3D

gg_plot

Creates 3-D ggplot chart

data

created ggplot chart

width = 3

Width of ggplot, in `units`.

scale = 150

affects the height of the 3D transformation.

multicore = FALSE

allow to use multicore

window size

Two-dimensional vector of window size

sunangle = 315

If raytracing: the angle of light source.

zoom

objects size in a window

phi

An angle with Z-axis

theta

An angle with X-axis

`plot_gg(mplot, phi = 30, theta = 45)`

Creating Hillshades and Color Mappings

sphere_shade

Creates a 3-D terrain model

heightmap

A 2-D matrix, where each number is the elevation at that point.

sunangle = 315

The direction of the main highlight color

texture

sq. matrix of the spherical texture or string indicating built-in palette

normalvectors = NULL

Cache of the normal vectors

colorintensity = 1

The intensity of the color mapping.
Higher values increase the intensity.

zscale = 1

The ratio between the x and y spacing

progbar = interactive()

activates progress bar

`sphere_shade(montereybay,
texture='desert')`

Shadows/Overlays

add_water(hillshade, watermap, color = 'imhof1')

Adds a water layer

hillshade

A three-dimensional RGB array.

watermap

Matrix indicating whether water was detected at that point. 1/0 - water/no water.

color = 'imhof1'

A hexcode, recognized color string or palettes included in sphere_shade

`add_water(montereybay,
where_is_water_matrix)`

Get data

load data 'montereybay' in R
`data <- montereybay`

rayshader CHEAT SHEET

R Package for 2D and 3D mapping and data visualization



Plotting and Saving 2D and 3D Visualizations

plot_3d(...)

Plot 3D

plot_map(...)

Plot Map

save_3dprint(...)

Save 3D Print

save_obj(...)

Save OBJ

save_png(...)

Save PNG

Creating Hillshades and Color Mappings

ambient_shade(...)

Calculate Ambient Occlusion Map

height_shade(...)

Calculate Terrain Color Map

lamb_shade(...)

Calculate Lambert Shading Map

ray_shade(...)

Calculate Raytraced Shadow Map

calculate_normal(...)

Calculate Normal

create_texture(...)

Create Texture

Capture 3D Maps

render_camera(...)

Render Camera

render_depth(...)

Render Depth of Field

render_highquality(...)

Render High Quality

render_label(...)

Render Label

render_movie(...)

Render Movie

render_water(...)

Render Water Layer

Adding Shadows and Overlays to Maps

add_overlay(...)

Add Overlay

add_shadow(...)

Add Shadow

detect_water(...)

Detect water