

Rayshader를 이용한 지도 3D 시각화

- 실습예제 5가지 -

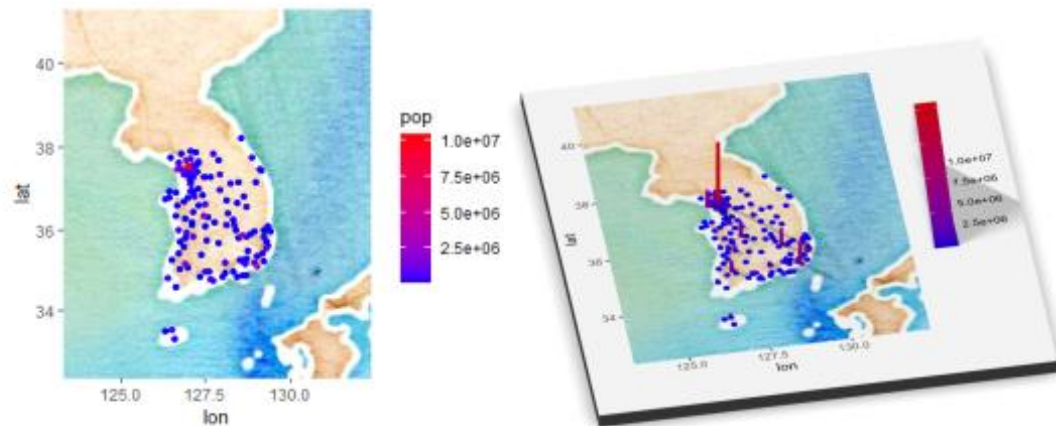
김철민

cmman75@gmail.com

Rayshader를 이용한 지도 3D 시각화

- 개요

- 최근 컴퓨터 기술의 발달과 함께 3D 그래픽 기술이 발달하면서 지리정보 시스템(GIS)에서의 3D 기법 적용이 증가하고 있음
- 이러한 시각화 기법은 공간적 맥락에서 효과적으로 정보를 제시하는 방법으로 지리공간 데이터 분석에서는 추상 정보(종종 숫자)가 지리적 맥락에서 제시되고 평가되며, 여기서 지도는 지리적 수량과 패턴 사이의 관계 직관적으로 제시함으로써 통찰력을 제공할 수 있음
- 초기의 3D bar chart는 단순히 지도 위에 시각화 기법을 적용하는데 그쳤으나 최근에는 이를 입체적으로 보여줌으로써 보다 다양한 통찰을 제공함
- 아래 그래프는 대한민국의 주요 도시별 인구를 나타내는 데이터이며, 과거의 경우 왼쪽과 같이 지도 위에 단순 포인트 데이터를 표현하는데 머물렀으나, 최근 분석기법과 도구가 발달하면서 오른쪽 그림과 같이 3D 시각화를 통한 직관적 이해를 높일 수 있음



<그림 3.41> 포인트 데이터를 이용한 3D bar-chart 예시

```
#####
### Rayshader 패키지
#####

#-----
# 0_패키지 설치와 소개
#-----

# 0-01_소개

# rayshade 패키지는 raytracing과 hillshading algorithms을 기반으로 하는 R 기반 2D / 3D 매핑 소프트웨어임
# 특히 지도와 관련하여, 이 패키지는 ggplot2 오브젝트를 3D 시각화 할 수 있다는 장점을 가지고 있음
# 이 모델링은 카메라 시점을 움직여서 자유롭게 3D 애니메이션으로 만들 수 있다는 장점을 가지고 있음
# 안내: https://www.rayshader.com/
```

```
24 # 1_실습 1: 단계도(Choropleth) 3D: North Carolina 출생률
25 #-----
26
27 # 1-01_작업디렉토리 세팅
28
29 rm(list = ls())
30 setwd(dirname(rstudioapi::getSourceEditorContext()$path))
31
32 # 1-02_라이브러리 불러오기
33
34 library(sf)          # install.packages("sf")
35 library(ggplot2)     # install.packages("remote")
36 library(rayshader)  # remotes::install_github("tylrmorganwall/rayshader")
37 library(viridis)    # install.packages("viridis")
38
39 # 1-03_데이터 불러오기
40
41 nc <- st_read(system.file("shape/nc.shp", package="sf"), quiet = TRUE)
42 head(nc)             # North Carolina주의 카운티별 유아돌연사증후군(Sudden Infant Death Syndrome: SIDS) 샘플 데이터
```

깃허브 파일 다운로드 링크

https://github.com/cmman75/Lecture/tree/main/2304_3D%EC%A7%80%EB%8F%84_%EB%A7%8C%EB%93%A4%EA%B8%B0

```
44 # 1-04_지도 시각화: 1979년 미국 노스 캐롤라이나 카운티별 출생률 시각화
```

```
45
46 gg_nc <- ggplot(nc) + geom_sf(aes(fill = BIR79, color= 'grey')) +
47   scale_fill_viridis("Area") +
48   theme(legend.position = "none") +
49   ggtitle("카운티별 출생률") +
50   theme_bw()
```

```
51 gg_nc
52 class(gg_nc)
```

```
53
54 # 1-05_3D 시각화
```

```
55
56 plot_gg(gg_nc, multicore = TRUE, width = 6, height= 5, scale= 250, raytrace = FALSE, windowsize=c(1400,866), zoom = 0.55, phi = 30, pad = 50)
```

```
57
58 # 옵션 설명
```

```
59 # multicore: 병렬처리 사용여부
```

```
60 # width: 그래프 가로길이(기본 6)
```

```
61 # height: 그래프 세로길이(기본 5)
```

```
62 # scale: 그래프의 크기(기본 250)
```

```
63 # raytrace: 3차원 장면을 사실적으로 렌더링하기 위한 알고리즘
```

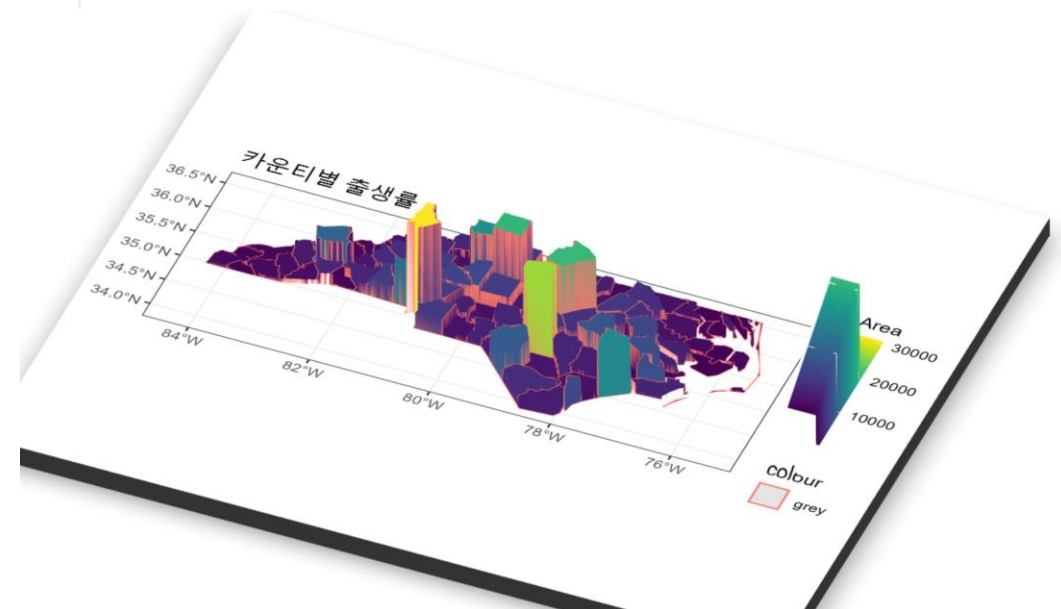
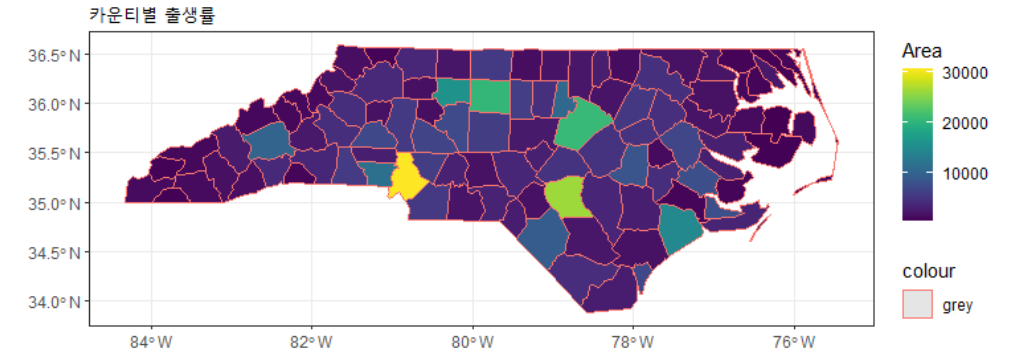
```
64 # (빛의 반사, 굴절 등을 고려하여 그래프를 생성)
```

```
65 # windowsize: 그래프 창 크기(기본값 1400 X 866)
```

```
66 # zoom: 그래프 줌 레벨 (기본 0.55)
```

```
67 # phi: 그래프 각도(시점) (기본 30)
```

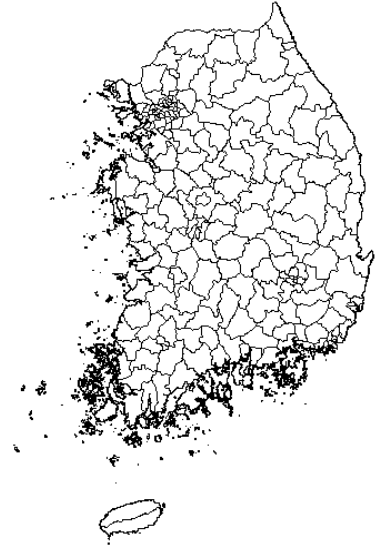
```
68 # pad: 그래프 가장자리 여백(기본 50)
```



```

70 #-----
71 # 2_실습 2: 등고선 3D: 인천시 지도 만들기
72 #-----
73
74 # 2-01_작업디렉토리 세팅
75
76 rm(list = ls())
77 setwd(dirname(rstudioapi::getSourceEditorContext()$path))
78
79 # 2-02_시도 경계선 불러오기
80
81 library(raster)
82 map <-getData('GADM', country='south korea', level=2)
83
84 map$NL_NAME_1 # 시도명 보기
85 map$NL_NAME_2 # 시군구명 보기
86 plot(map)     # 지도 플로팅
87
88 # 2-03_시도 경계선 추출(인천만)
89
90 library(sf)     # install.packages("sf")
91 library(dplyr)  # install.packages("dplyr")
92
93 map <- as(map, "sf") # map 파일 SP 형식 => SF형으로 변환 필요
94 map <- map %>% filter(NAME_1 == "Incheon") # 인천만 추출
95 plot(map$geometry)

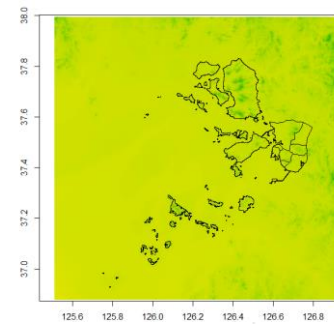
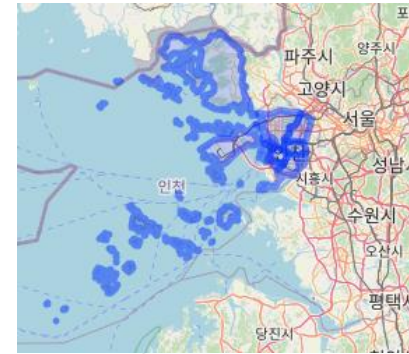
```



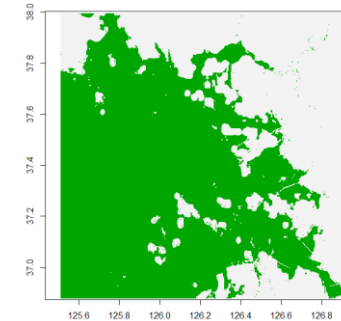
```

97 # 2-04_인천 추출결과 지도 상에서 확인
98
99 library(leaflet) # install.packages("leaflet")
100 leaflet() %>% addTiles() %>% addPolygons(data = map)
101
102 # 2-05_geo-viz로 DEM 자료 불러오기
103
104 library(geoviz) # install.packages("geoviz")
105 dem <- mapzen_dem(37.5, 126.3, square_km = 40, max_tiles = 20)
106 # mapzen_dem은 Mapzen의 지형 타일(terrain tiles)을 기반으로한 DEM(Digital Elevation Model) 데이터
107 # 전세계 지형 및 해저지형 데이터를 제공함
108
109 # 2-06_DEM 자료 지도 시각화
110
111 plot(dem) # DEM 자료 불러오기
112 plot(map$geometry, add=T) # 인천 추출결과 덧붙이기
113
114 # 2-07_DEM 자료 필터링(해수면 위 데이터만 필터링)
115
116 tmp <- dem <= 0 # 해수면 아래만 추출
117 plot(tmp)
118
119 dem <- mask(dem, tmp, maskvalue=1) # 전체 DEM - 해수면 아래 DEM
120 plot(dem)
121 plot(map$geometry, add=T) # 플로팅
122
123 rm("tmp") # 불필요한 변수 제거

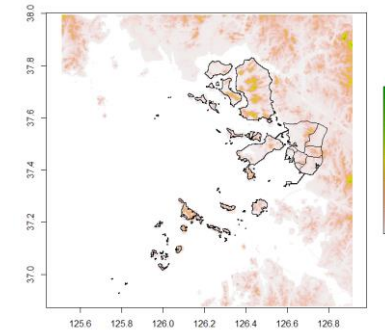
```



전체 DEM



해수면 아래 DEM



해수면 위 DEM


```

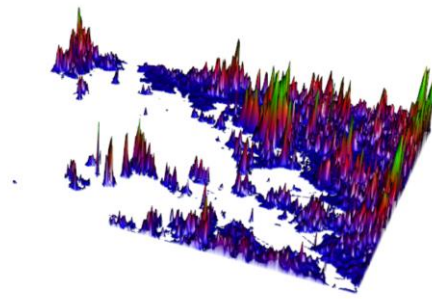
125 # 2-08_3D 시각화
126
127 library(rasterVis) # devtools::install_github('oscarperpinan/rasterVis')
128
129 mycols <- colorRampPalette(c("blue", "red", "green", "yellow"))
130 plot3D(dem, col = mycols(100), adjust=TRUE) # 3D 플로팅
131
132 # 2-09_rasterToPoints
133
134 library(tibble) # install.packages("tibble")
135 dem_pnt <- as_tibble(rasterToPoints(dem))
136 dem_pnt <- na.omit(dem_pnt)
137 plot3d(dem_pnt, maxpixels = 100)
138
139 # 2-10_지도 시각화
140
141 library(leaflet)
142 leaflet() %>% addTiles() %>%
143   addCircleMarkers(data = dem_pnt[50000:100000,], lng = ~x, lat = ~y, radius= 1)
144
145 # 2-11_그리드 2D 지도
146
147 library(rayshader)
148 library(ggplot2)
149
150 plot_out <- ggplot(data=dem_pnt, aes(x = x, y = y, z = layer)) +
151   stat_summary_2d(bins = 50, size = 0, color = "black") + # stat_summary_hex()
152   scale_fill_viridis_c(option = "C") +
153   scale_x_continuous("X", expand = c(0,0)) +
154   scale_y_continuous("Y", expand = c(0,0)) +
155   scale_fill_gradientn("Z", colours = terrain.colors(10)) +
156   coord_fixed()
157
158 plot_out

```

```

125 # 2-08_3D 시각화
126
127 library(rasterVis) # devtools::install_github('oscarperpinan/rasterVis')
128
129 mycols <- colorRampPalette(c("blue", "red", "green", "yellow"))
130 plot3D(dem, col = mycols(100), adjust=TRUE) # 3D 플로팅
131

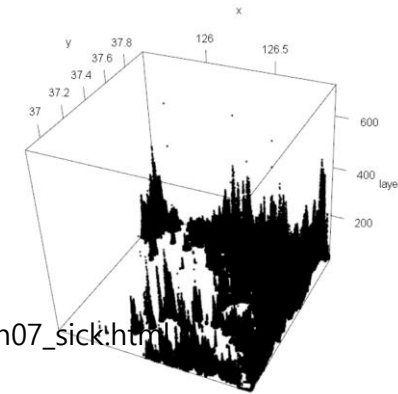
```



```

132 # 2-09_rasterToPoints
133
134 library(tibble) # install.packages("tibble")
135 dem_pnt <- as_tibble(rasterToPoints(dem))
136 dem_pnt <- na.omit(dem_pnt)
137 plot3d(dem_pnt, maxpixels = 100)
138

```



```

139 # 2-10_지도 시각화
140
141 library(leaflet)
142 leaflet() %>% addTiles() %>%
143   addCircleMarkers(data = dem_pnt[50000:100000,], lng = ~x, lat = ~y, radius= 1)
144

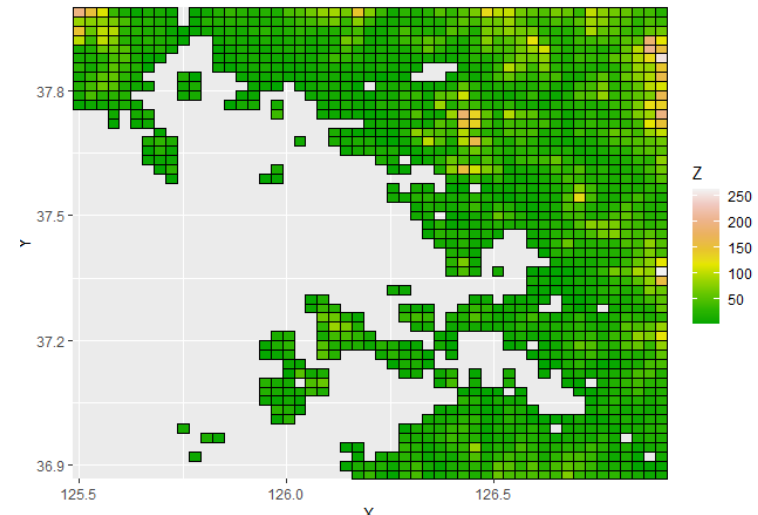
```

https://rpm.snu.ac.kr/dataset/webgl/urban07/urban07_sick.htm

```

145 # 2-11_그리드 2D 지도
146
147 library(rayshader)
148 library(ggplot2)
149
150 plot_out <- ggplot(data=dem_pnt, aes(x = x, y = y, z = layer)) +
151   stat_summary_2d(bins = 50, size = 0, color = "black") + # stat_summary_hex()
152   scale_fill_viridis_c(option = "C") +
153   scale_x_continuous("X", expand = c(0,0)) +
154   scale_y_continuous("Y", expand = c(0,0)) +
155   scale_fill_gradientn("Z", colours = terrain.colors(10)) +
156   coord_fixed()
157
158 plot_out

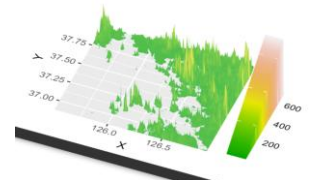
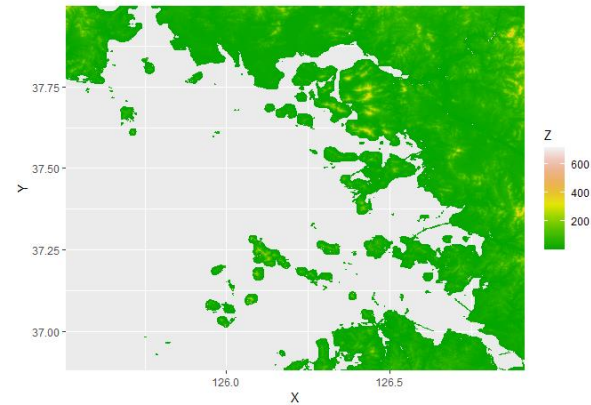
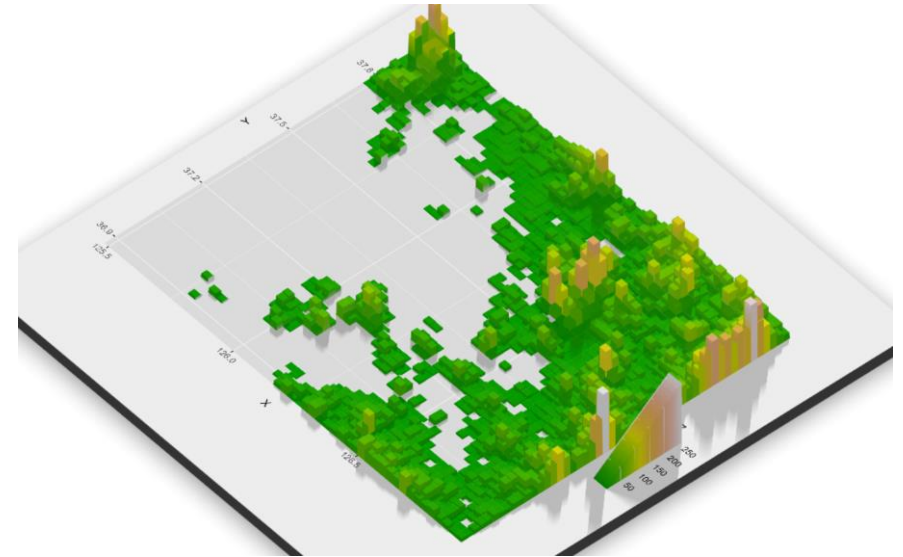
```




```

160 # 2-12_그리드 3D 지도
161
162 plot_gg(plot_out, multicore = TRUE, raytrace = TRUE, width = 8, height = 8,
163         scale = 300, windowsize = c(1400, 866), zoom = 0.6, phi = 30, theta = 30)
164
165
166 # 2-13_동영상 추출
167
168 library(av) # install.packages("av")
169 render_movie(filename = "demo",
170             type = "orbit",
171             phi = 45,
172             theta = 60)
173
174 library(rgl)
175 rgl.clear()
176
177 # 2-14_등고선 2D 지도
178
179 plot_out <- dem_pnt %>% ggplot() +
180   geom_tile(aes(x= x,y= y,fill= layer)) +
181   geom_contour(aes(x= x,y= y,z= layer),color=NA) +
182   scale_x_continuous("X",expand = c(0,0)) +
183   scale_y_continuous("Y",expand = c(0,0)) +
184   scale_fill_gradientn("Z",colours = terrain.colors(10)) +
185   coord_fixed()
186
187 plot_out
188
189 # 2-15_등고선 3D 지도
190
191 plot_gg(plot_out, multicore = TRUE, width = 6, height= 2, scale= 250, raytrace = FALSE, windowsize=c(1400,866), zoom = 0.55, phi = 30, pad = 50)
192
193 # 2-16_정리하기
194
195 library(rgl)
196 rgl.clear()
197 rgl.close()

```



```

160 # 2-12_그리드 3D 지도
161
162 plot_gg(plot_out, multicore = TRUE, raytrace = TRUE, width = 8, height = 8,
163         scale = 300, windowsize = c(1400, 866), zoom = 0.6, phi = 30, theta = 30)
164
165
166 # 2-13_동영상 추출
167
168 library(av) # install.packages("av")
169 render_movie(filename = "demo",
170             type = "orbit",
171             phi = 45,
172             theta = 60)
173
174 library(rgl)
175 rgl.clear()
176
177 # 2-14_등고선 2D 지도
178
179 plot_out <- dem_pnt %>% ggplot() +
180     geom_tile(aes(x= x,y= y,fill= layer)) +
181     geom_contour(aes(x= x,y= y,z= layer),color=NA) +
182     scale_x_continuous("X",expand = c(0,0)) +
183     scale_y_continuous("Y",expand = c(0,0)) +
184     scale_fill_gradientn("Z",colours = terrain.colors(10)) +
185     coord_fixed()
186
187 plot_out
188
189 # 2-15_등고선 3D 지도
190
191 plot_gg(plot_out, multicore = TRUE, width = 6, height= 2, scale= 250, raytrace = FALSE, windowsize=c(1400,866), zoom = 0.55, phi = 30, pad = 50)
192
193 # 2-16_정리하기
194
195 library(rgl)
196 rgl.clear()

```

```

200 #-----
201 # 3_실습 3: 인구 데이터 시각화: 뉴저지 인구밀도
202 #-----
203
204 # 3-01_라이브러리 불러오기
205
206 library(tidycensus) # install.packages("tidycensus")
207 library(tidyverse)  # install.packages("tidyverse")
208 library(rayshader)  # install.packages("rayshader")
209 library(rayrender)  # install.packages("rayrender")
210 library(sf)          # install.packages("sf")
211 library(viridis)     # install.packages("viridis")
212 library(units)       # install.packages("units")
213
214 # 4-02_데이터 불러오기
215
216 data <- get_acs(variables = "B01001_001", geography = "tract", state = "NJ", survey = "acs5", year = 2019, geometry = TRUE )
217 data
218
219 # 4-03_인구밀도 계산
220
221 data <- data %>% mutate(area = set_units(st_area(data), mi^2),
222                        pop_density = as.numeric(estimate/area))
223
224 data %>% ggplot(aes(fill = pop_density)) +
225   geom_sf(color = NA) +
226   scale_fill_viridis_c(option = "plasma", trans = "sqrt")

```

```

228 # 4-04_ggplot 그리기
229
230 plot <- data %>% ggplot(aes(fill = pop_density)) +
231     geom_sf(color = NA) +
232     scale_fill_viridis_c(option = "plasma", trans = "sqrt") +
233     theme(axis.line = element_line(colour = "transparent"),
234           panel.grid.minor = element_blank(),
235           panel.grid.major = element_blank(),
236           panel.border = element_blank(),
237           axis.title.x = element_blank(),
238           axis.title.y = element_blank(),
239           axis.ticks = element_blank(),
240           axis.text = element_blank(),
241           panel.background = element_rect(fill = "transparent", color = "white"),
242           plot.background = element_rect(fill = "transparent", colour = "white"),
243           legend.text = element_text(color = "transparent"),
244           legend.title = element_text(color = "transparent"),
245           legend.position = "blank")
246
247 plot

```

```

249 # 4-03_3D plot 그리기

```

```

251 plot_gg(plot, multicore = TRUE, width = 6, height = 6, scale = 300, fov = 60, theta = 270, phi = 25, zoom = 0.2, pad = 50, raytrace = FALSE, windowsize=c(2500,1000), )

```

```

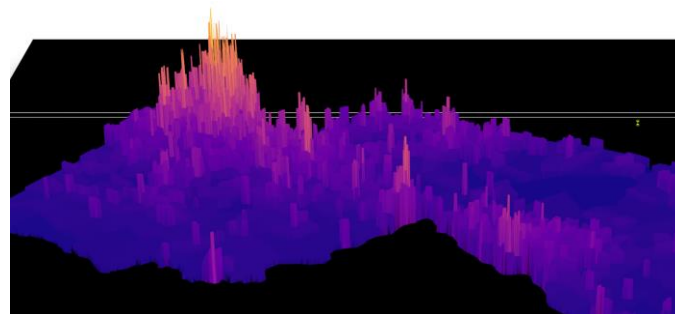
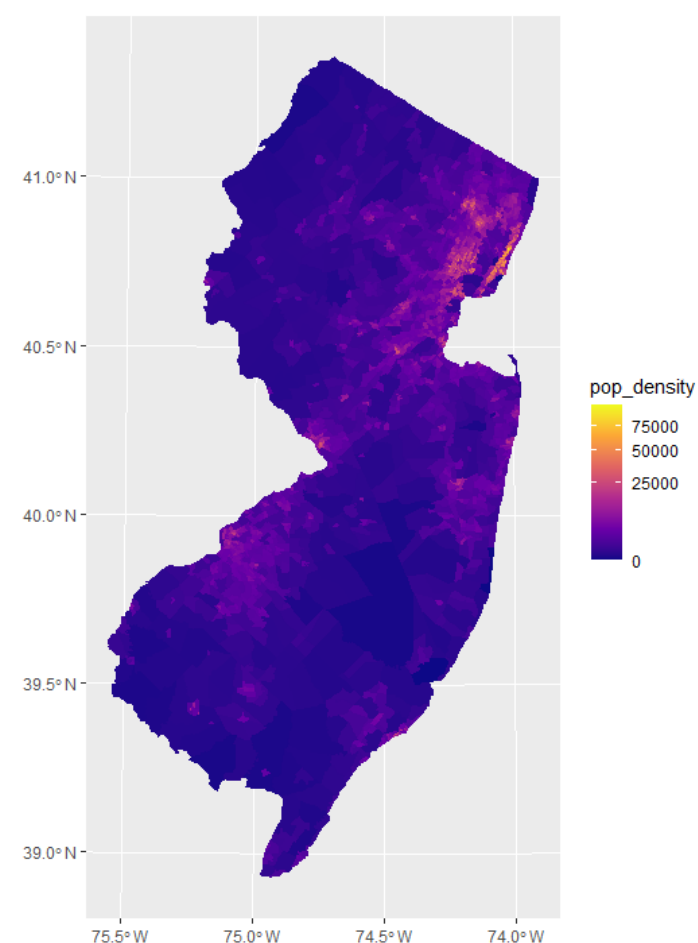
253 # 4-04_정리하기

```

```

254 library(rgl)
255 rgl::rgl.clear()
257 rgl.close()

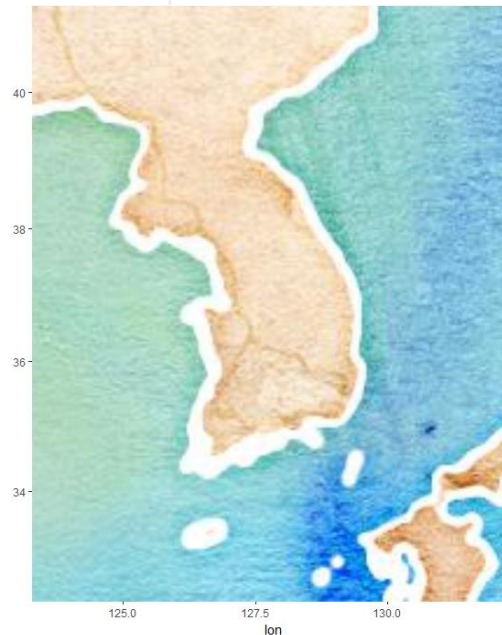
```



```

260 #-----
261 # 4_실습 4: 대한민국 주요도시 인구밀도 현황
262 #-----
263
264 # 4-01_작업디렉토리 세팅
265
266 rm(list = ls())
267 setwd(dirname(rstudioapi::getSourceEditorContext()$path))
268
269 # 4-02_라이브러리 불러오기
270
271 library(rayshader)
272 library(ggplot2)
273 library(dplyr)
274 library(maps)
275 library(ggmap)
276
277 # 4-03_도시 데이터 불러오기
278
279 data(world.cities)
280
281 # 4-04_대한민국 도시 데이터만 가져오기
282
283 world.cities <- world.cities
284 city <- world.cities %>% filter(country.etc == "Korea South")
285
286 # 4-05_대한민국 지도 가져오기(레스터 데이터)
287
288 library(ggmap) # install.packages("ggmap")
289 city_map <- get_stamenmap(bbox = c(left = min(city$long)-3,
290                                   bottom = min(city$lat)-1,
291                                   right = max(city$long)+3,
292                                   top = max(city$lat)+3),
293                           maptype = "watercolor",
294                           zoom = 5)
295 ggmap(city_map)

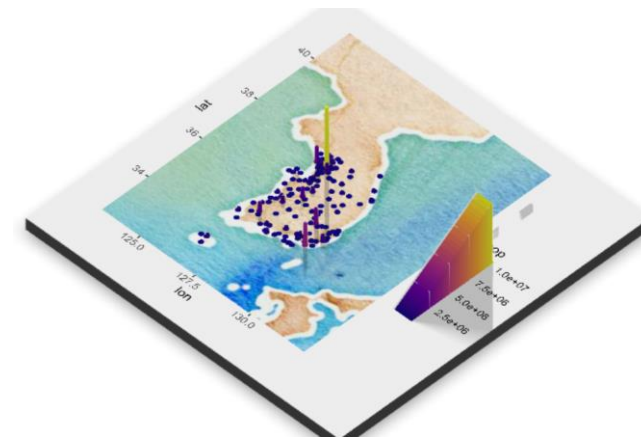
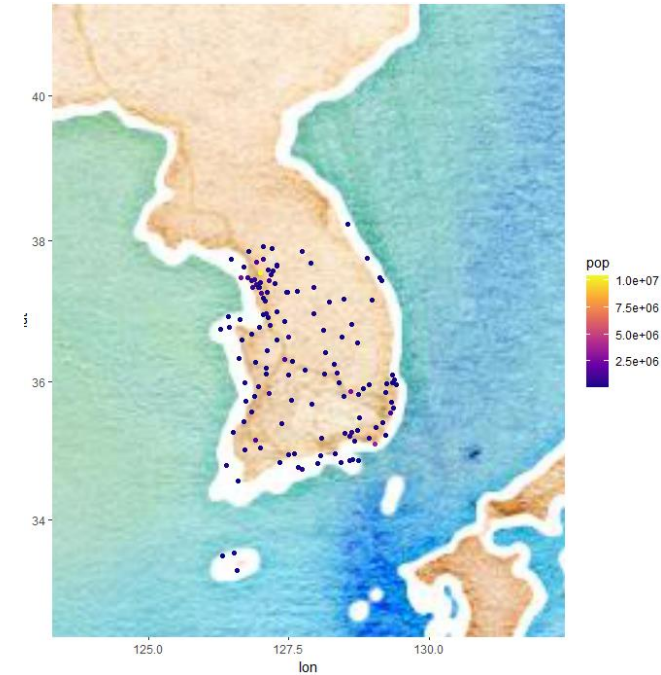
```




```

297 # 4-06_지도 이미지(워터컬러 이미지) 만들기
298
299 city_map_attributes <- attributes(city_map)
300 city_map_trans <- matrix(adjustcolor(city_map, alpha.f = 0), nrow = nrow(city_map))
301 attributes(city_map_trans) <- city_map_attributes
302 city_map_trans
303
304 # 4-07_ggplot으로 인구 시각화
305
306 together_plot <- ggmap(city_map_trans) +
307   geom_point(data = city,
308             aes(x = long, y = lat, color = pop)) +
309   scale_color_viridis_c(option = "C")
310
311 together_plot
312
313 # 4-08_인구 + 지도 시각화
314
315 point_plot <- ggmap(city_map) +
316   geom_point(data = city,
317             aes(x = long, y = lat, color = pop)) +
318   scale_color_viridis_c(option = "C")
319
320 point_plot
321
322 # 4-09_3D 지도
323
324 plot_gg(list(point_plot,together_plot),multicore=TRUE, width=4.5, height=4.5, scale=250, windowsize = c(1000,800))

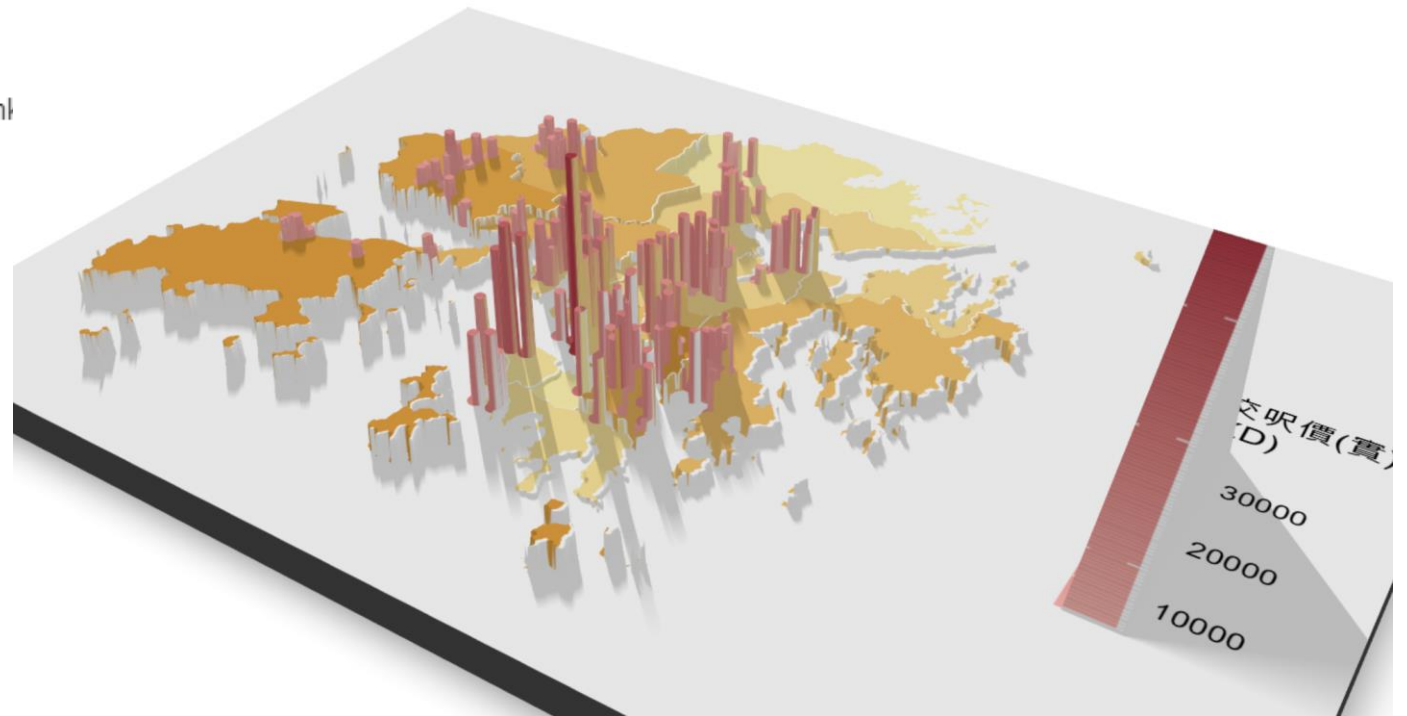
```




```

326 #-----
327 # 5_실습 5: 홍콩 인구밀도 분포
328 #-----
329
330 # 5-01_디렉토리 세팅
331
332 rm(list = ls())
333 setwd(dirname(rstudioapi::getSourceEditorContext()$path))
334
335
336 population <- read.csv("../tmp/District Name.csv")
337 colnames(population) = c("name", "Population")
338
339 hkmap = readRDS("../tmp/HKG_adm1.rds")
340
341 library(ggplot2)
342 # Preprocessing
343
344 map_data = data.frame(id=hkmap$ID_1, Code=hkmap$HASC_1, name=hkmap$name)
345 map_data$Code = gsub('HK.', '', as.character(map_data$Code))
346 map_data = merge(map_data, population, by = 'name')
347 hkmapdf = fortify(hkmap)
348 map_data = merge(hkmapdf, map_data, by="id")
349 map_data$Population = as.numeric(map_data$Population)
350

```



감사합니다.