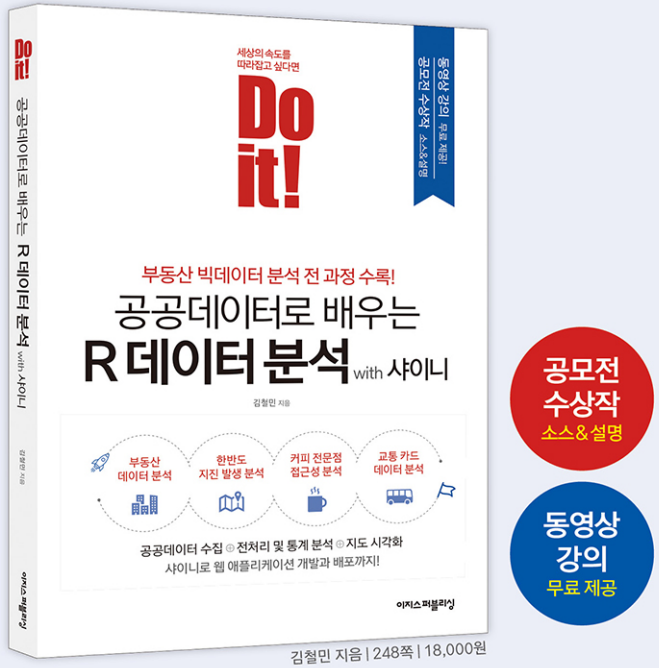
R로 공공데이터를 분석하는 전 과정 실습!

공모전 수상작으로 배우는 **R 데이터 분석**



공모전 수상작 소스&설명

동영상 강의 무료 제공

김철민 지음 | 248쪽 | 18,000원

10

데이터 분석 애플리케이션 개발하기

이번 장에서는 09장에서 배운 샤이니를 이용하여 지도 기반 웹 애플리케이션을 만들어 봅니다. 먼저 R에서 반응형 지도를 구현하는 방법을 살펴보고 이를 확장하여 샤이니 안에 지도를 포함하는 과정을 살펴봅니다. 마지막으로 반응형 함수를 추가하여 실제 활용할 수 있는 애플리케이션을 구현해 봅니다.

10-1 반응형 지도 만들기

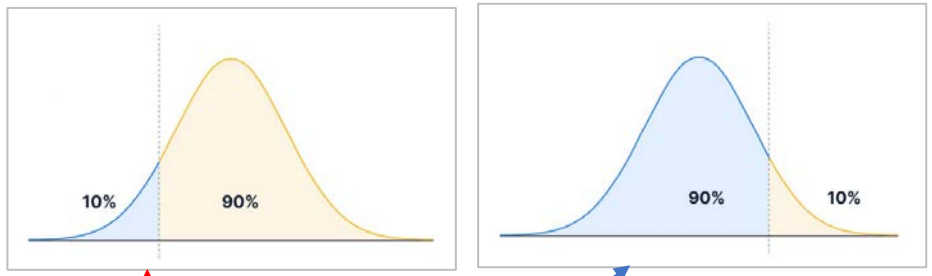
1단계 데이터 불러오기

Do it! 데이터 불러오기 10_웹애플리케이션.R

```
08: setwd(dirname(rstudioapi::getSourceEditorContext()$path))
09: load("./06_geodataframe/06_apt_price.rdata")      # 아파트 실거래 데이터
10: library(sf)
11: bnd <- st_read("./01_code/sigun_bnd/seoul.shp")     # 서울시 경계선
12: load("./07_map/07_kde_high.rdata")    # 최고가 래스터 이미지
13: load("./07_map/07_kde_hot.rdata")     # 급등 지역 래스터 이미지
14: grid <- st_read("./01_code/sigun_grid/seoul.shp")   # 서울시 1km 그리드
```

10-1 반응형 지도 만들기

2단계 마커 클러스터링 설정



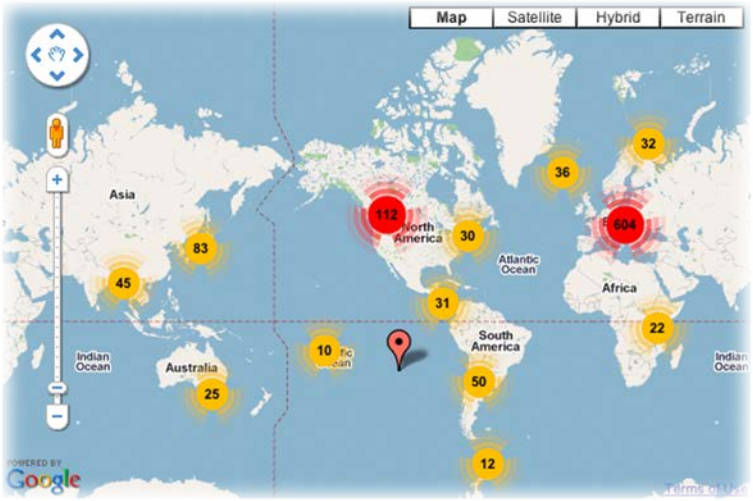
이상치

이상치

Do it! 마커 클러스터링 설정 10_웹애플리케이션.R

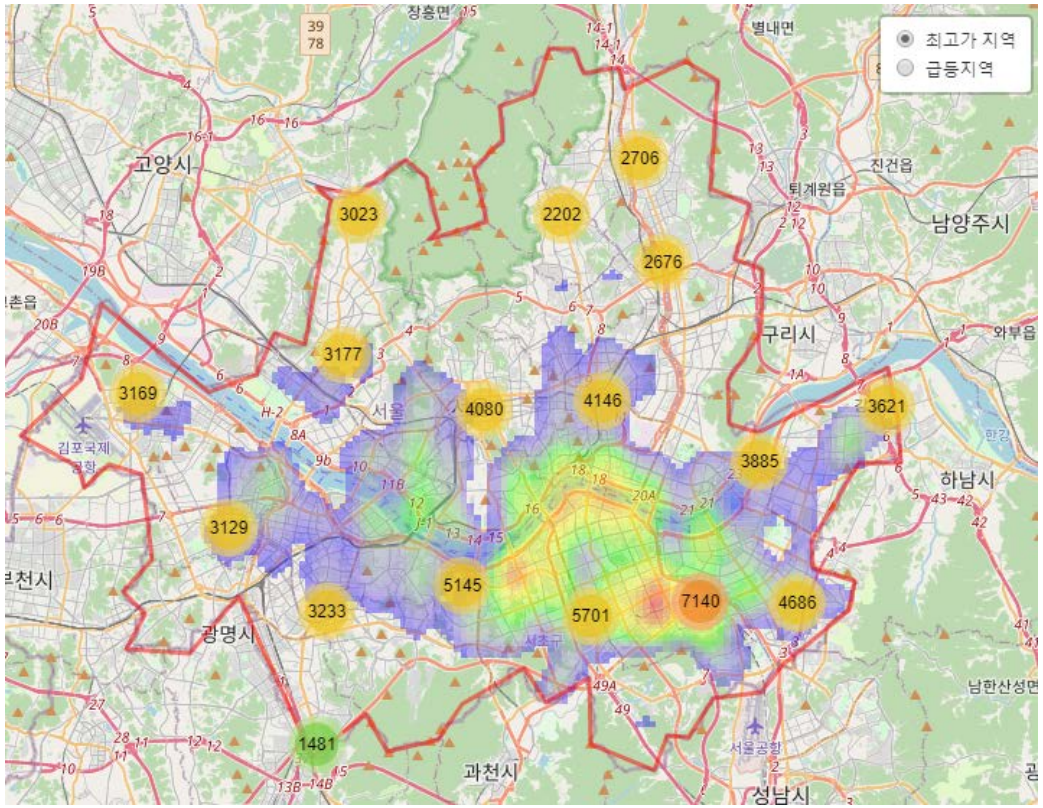
```
18: pcnt_10 <-as.numeric(quantile(apt_price$py, probs=seq(.1,.9,by=.1))[1])  # 하위 10%
19: pcnt_90 <-as.numeric(quantile(apt_price$py, probs=seq(.1,.9,by=.1))[9])  # 상위 10%
20: load("./01_code/circle_marker/circle_marker.rdata")   # 마커 클러스터링 함수
21: circle.colors <- sample(x=c("red","green","blue"), size=1000, replace=TRUE)
```

마커 클러스터링 옵션



10-1 반응형 지도 만들기

3단계 반응형 지도 만들기



Do it! 반응형 지도 만들기 — 10_웹애플리케이션.R

```
25: library(leaflet)
26: library(purrr)
27: library(raster)
28: leaflet() %>%
29:  #---# 기본 맵 설정: 오픈스트리트맵
30: addTiles(options = providerTileOptions(minZoom = 9, maxZoom = 18)) %>%
31:  #---# 최고가 지역 KDE
32: addRasterImage(raster_high,
33:  colors = colorNumeric(c("blue", "green","yellow","red"),
34:  values(raster_high), na.color = "transparent"), opacity = 0.4,
35:  group = "2021 최고가") %>%
36:  #---# 급등 지역 KDE
37: addRasterImage(raster_hot,
38:  colors = colorNumeric(c("blue", "green","yellow","red"),
39:  values(raster_hot), na.color = "transparent"), opacity = 0.4,
40:  group = "2021 급등지") %>%
41: #---# 레이어 스위치 메뉴
42: addLayersControl(baseGroups = c("2021 최고가", "2021 급등지"),
43:  options = layersControlOptions(collapsed = FALSE)) %>%
44: #---# 서울시 외곽 경계선
45: addPolygons(data=bnd, weight = 3, stroke = T, color = "red",
46:  fillOpacity = 0) %>%
47: #---# 마커 클러스터링
48: addCircleMarkers(data = apt_price, lng =unlist(map(apt_price$geometry,1)),
49:  lat = unlist(map(apt_price$geometry,2)), radius = 10, stroke = FALSE,
50:  fillOpacity = 0.6, fillColor = circle.colors, weight=apt_price$py,
51:  clusterOptions = markerClusterOptions(iconCreateFunction=JS(avg.formula)))
```

배경맵 설정

커널추정: 최고가

커널추정: 급등지

선택 레이어

외곽 경계선

10-2 지도 애플리케이션 만들기

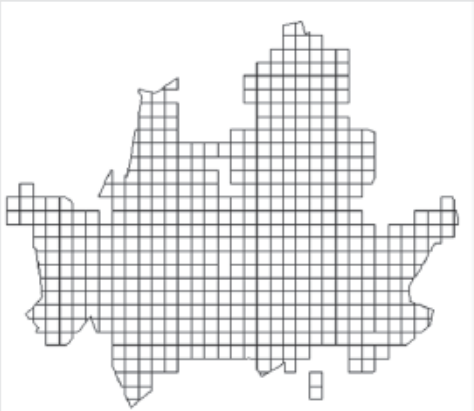
1단계 그리드 필터링하기

Do it! 그리드 필터링 10_웹애플리케이션.R

```
60: grid <- st_read("./01_code/sigun_grid/seoul.shp")        # 그리드 불러오기
61: grid <- as(grid, "Spatial") ; grid <- as(grid, "sfc")    # 변환
62: grid <- grid[which(sapply(st_contains(st_sf(grid),apt_price),length) > 0)]   # 필터링
63: plot(grid)    # 그리드 확인
```

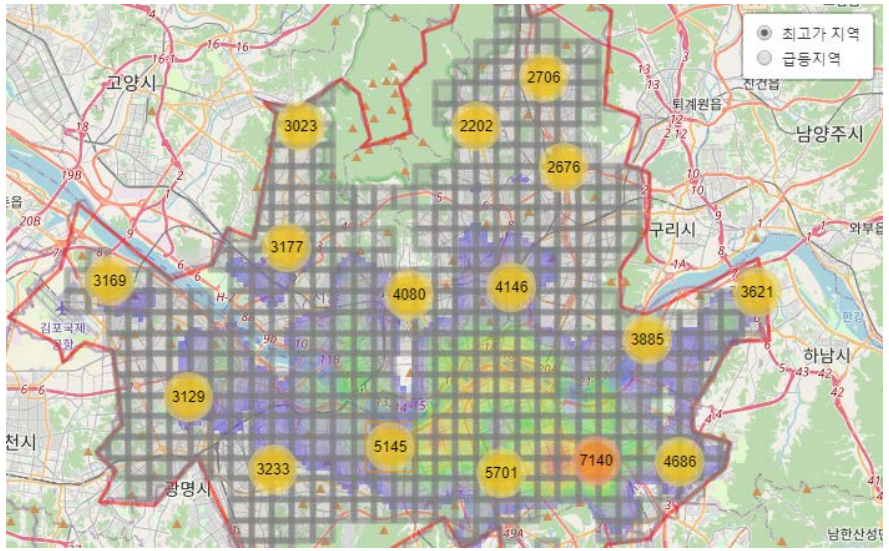
그리드 선택시 which 그리드가 선택되었는지 알아내고 필터링 !!!

☞ 실행 결과



10-2 지도 애플리케이션 만들기

2단계 반응형 지도 모듈화하기



Do it! 반응형 지도 모듈화 — 10_웹애플리케이션.R

```
m <- leaflet() %>%
  #---# 기본 맵 설정: 오픈스트리트맵
  addTiles(options = providerTileOptions(minZoom = 9, maxZoom = 18)) %>%
  #---# 최고가 지역 KDE
  addRasterImage(raster_high,
   colors = colorNumeric(c("blue", "green","yellow","red"),
   values(raster_high), na.color = "transparent"), opacity = 0.4,
   group = "2021 최고가") %>%
  #---# 급등 지역 KDE
  addRasterImage(raster_hot,
   colors = colorNumeric(c("blue", "green","yellow","red"),
   values(raster_hot), na.color = "transparent"), opacity = 0.4,
   group = "2021 급등지") %>%
  #---# 레이어 스위치 메뉴
  addLayersControl(baseGroups = c("2021 최고가", "2021 급등지"),
   options = layersControlOptions(collapsed = FALSE)) %>%
  #---# 서울시 외곽 경계선
  addPolygons(data=bnd, weight = 3, stroke = T, color = "red",
   fillOpacity = 0) %>%
  #---# 마커 클러스터링
  addCircleMarkers(data = apt_price, lng =unlist(map(apt_price$geometry,1)),
  lat = unlist(map(apt_price$geometry,2)), radius = 10, stroke = FALSE,
  fillOpacity = 0.6, fillColor = circle.colors, weight=apt_price$py,
  clusterOptions = markerClusterOptions(iconCreateFunction=JS(avg.formula))) %>%
  #---# 그리드
leafem  ::addFeatures(st_sf(grid), layerId= ~seq_len(length(grid)), color = 'grey')
m
```

10-2 지도 애플리케이션 만들기

3단계 **애플리케이션 구현하기**

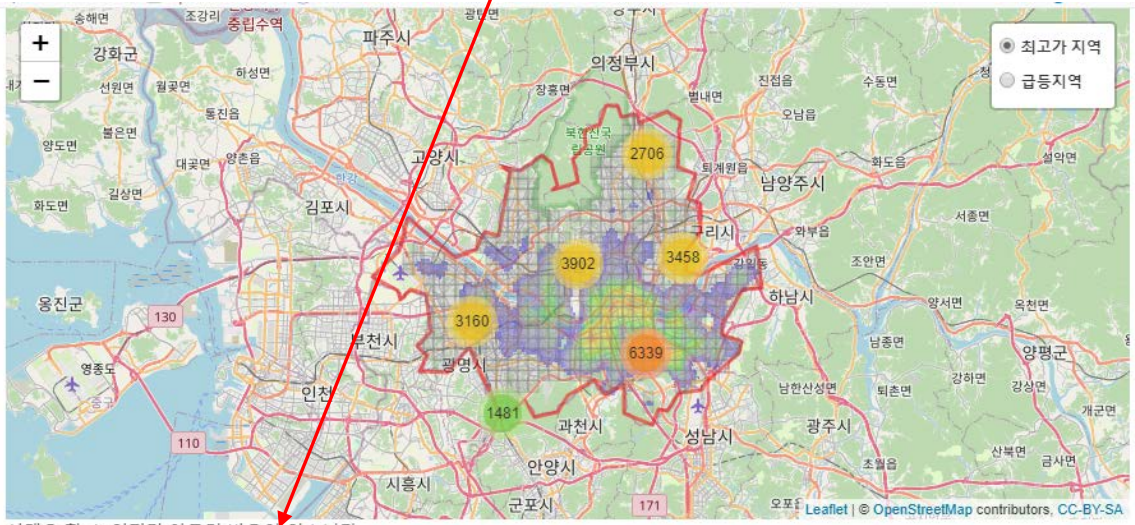
Do it! 샤이니와 mapedit으로 애플리케이션 구현 — 10_웹애플리케이션.R

```
97: library(shiny)   # install.packages("shiny")
98: library(mapedit) # install.packages("mapedit")
99: library(dplyr)   # install.packages("dplyr")
100:  #---# 사용자 인터페이스
101: ui <- fluidPage(
102:  selectModUI("selectmap"),
103:   "선택은 할 수 있지만 아무런 반응이 없습니다.")
104: #---# 서버
105: server <- function(input, output) {
106:  callModule(selectMod, "selectmap", m)}   # 모듈 서버 함수
107: #---# 실행
108: shinyApp(ui, server)
```

그리드 선택 모듈

렌더링 함수가 없어 반응 결과 전달 불가

☞ 실행 결과



선택은 할 수 있지만 아무런 반응이 없습니다.

10-2 지도 애플리케이션 만들기

4단계 반응식 추가하기

Do it! 반응식 추가 — 10_웹애플리케이션.R

```
112: #---# 사용자 인터페이스
113: ui <- fluidPage(
114:   selectModUI("selectmap"),
115:   textOutput("sel")
116:  )
117: #---# 서버
118: server <- function(input, output, session) {
119:    df <- callModule(selectMod, "selectmap", m)
120:    output$sel <- renderPrint({df()[1]})
121:  }
122: #---# 실행
123: shinyApp(ui, server)
```

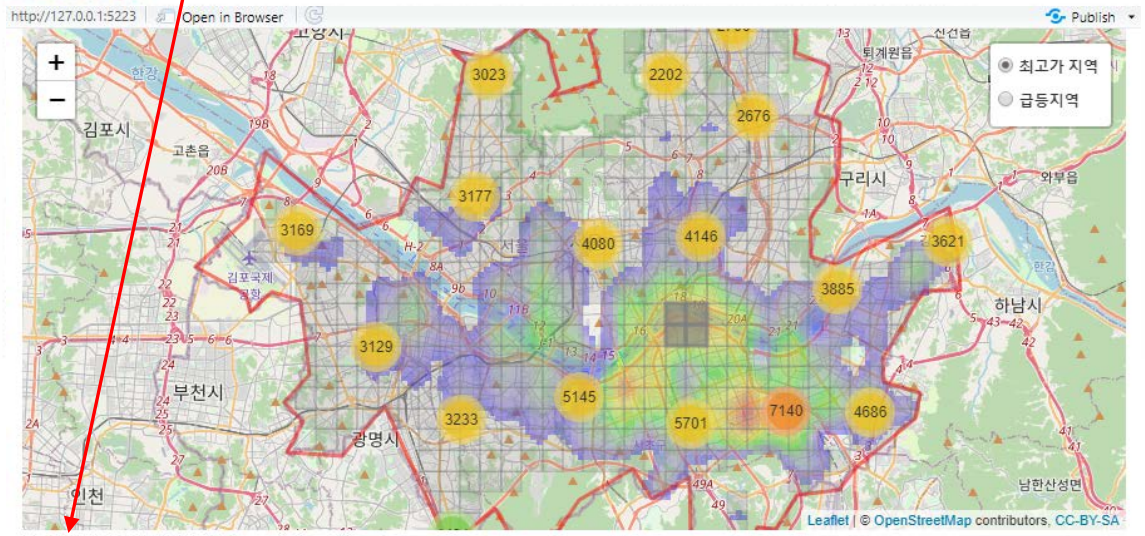
그리드 선택 모듈

렌더링 결과 전달

입력(그리드 선택) 전달

선택된 그리드의 ID 번호를 알려주는 반응식

☞ 실행 결과



id 1 219 2 193 3 194 4 220

10-3 반응형 지도 애플리케이션 완성하기

1단계 사용자 인터페이스 설정하기

Do it! 사용자 인터페이스 설정 — 10_웹애플리케이션.R

```
132: library(DT)   # install.packages("DT")
133: ui <- fluidPage(
134:  #---# 상단 화면: 지도 + 입력 슬라이더
135:  fluidRow(
136:   column( 9, selectModUI("selectmap"), div(style = "height:45px")),
137:   column( 3,
138:    sliderInput("range_area", "전용면적", sep = "", min = 0, max = 350,
139:                value = c(0, 200)),
140:    sliderInput("range_time", "건축 연도", sep = "", min = 1960,  max = 2020,
141:                value = c(1980, 2020)), ),
142:   #---# 하단 화면: 테이블 출력
143:   column(12, dataTableOutput(outputId = "table"), div(style = "height:200px"))))
```

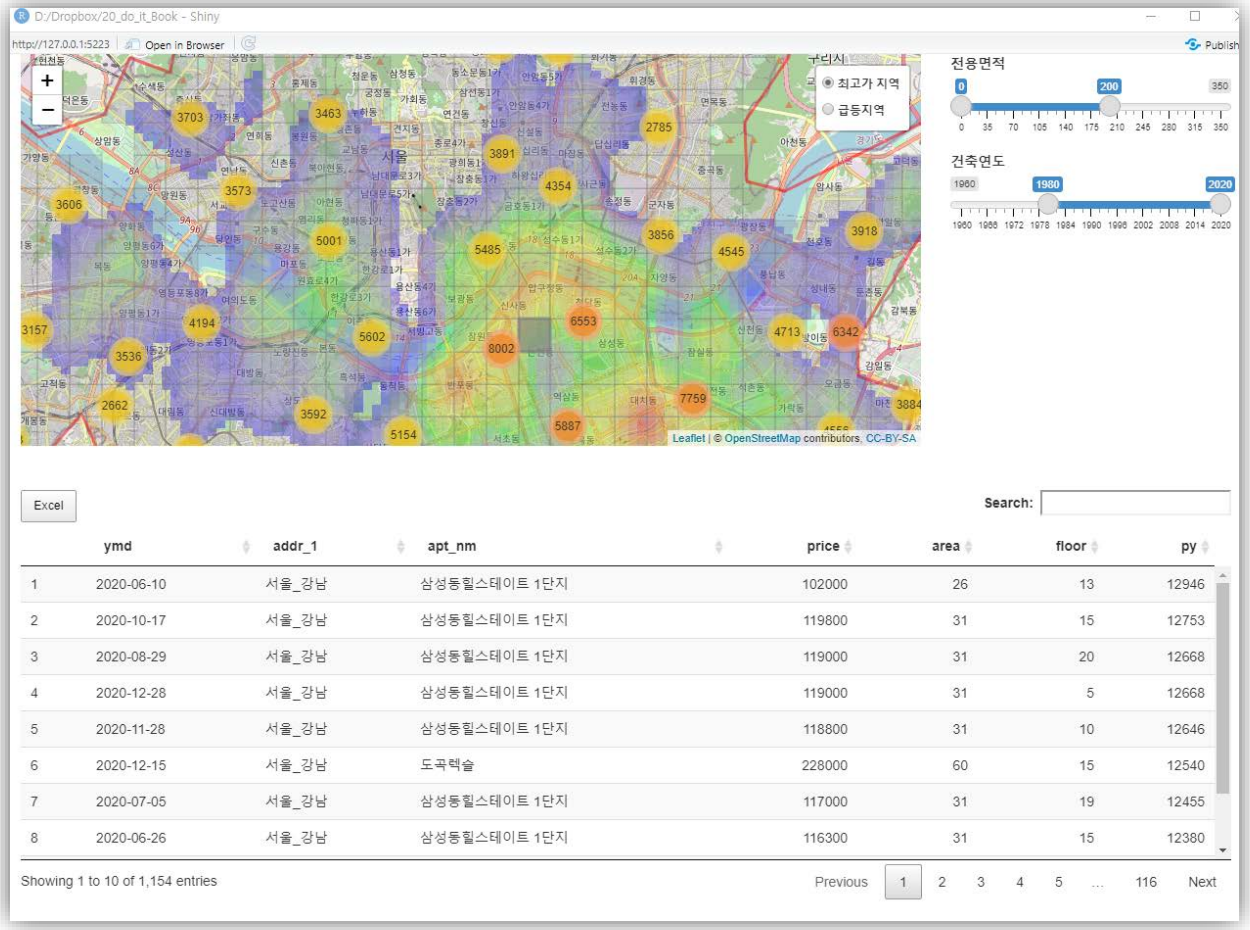
2단계 반응식 설정하기

Do it! 슬라이더 입력 필터링 — 10_웹애플리케이션.R

```
147: server <- function(input, output, session) {
148:  #---# 반응식
149: apt_sel = reactive({
150:   apt_sel = subset(apt_price, con_year >= input$range_time[1] &
151:              con_year <= input$range_time[2] & area >= input$range_area[1] &
152:              area <= input$range_area[2])
153:   return(apt_sel)})
```

10-3 반응형 지도 애플리케이션 완성하기

3단계 지도 입출력 모듈 설정하기



Do it! 그리드 선택 저장 — 10_웹애플리케이션.R

```
157: g_sel <- callModule(selectMod, "selectmap",
158:  leaflet() %>%
159:   #---# 기본 맵 설정: 오픈스트리트맵
160:   addTiles(options = providerTileOptions(minZoom = 9, maxZoom = 18)) %>%
161:   #---# 최고가 지역 KDE
162:   addRasterImage(raster_high,
163:    colors = colorNumeric(c("blue", "green","yellow","red"),
164:    values(raster_high), na.color = "transparent"), opacity = 0.4,
165:    group = "2021 최고가") %>%
166:   #---# 급등 지역 KDE
167:   addRasterImage(raster_hot,
168:    colors = colorNumeric(c("blue", "green","yellow","red"),
169:    values(raster_hot), na.color = "transparent"), opacity = 0.4,
170:    group = "2021 급등지") %>%
171:   #---# 레이어 스위치 메뉴
172:   addLayersControl(baseGroups = c("2021 최고가", "2021 급등지"),
173:    options = layersControlOptions(collapsed = FALSE)) %>%
174:   #---# 서울시 외곽 경계선
175:   addPolygons(data=bnd, weight = 3, stroke = T, color = "red",
176:    fillOpacity = 0) %>%
177:   #---# 마커 클러스터링
178:   addCircleMarkers(data = apt_price, lng =unlist(map(apt_price$geometry,1)),
179:    lat = unlist(map(apt_price$geometry,2)), radius = 10, stroke = FALSE,
180:    fillOpacity = 0.6, fillColor = circle.colors, weight=apt_price$py,
181:    clusterOptions=markerClusterOptions(iconCreateFunction=JS(avg.formula))) %>%
182:   #---# 그리드
183:   leafem::addFeatures(st_sf(grid),layerId= ~seq_len(length(grid)),
184:    color='grey'))
```