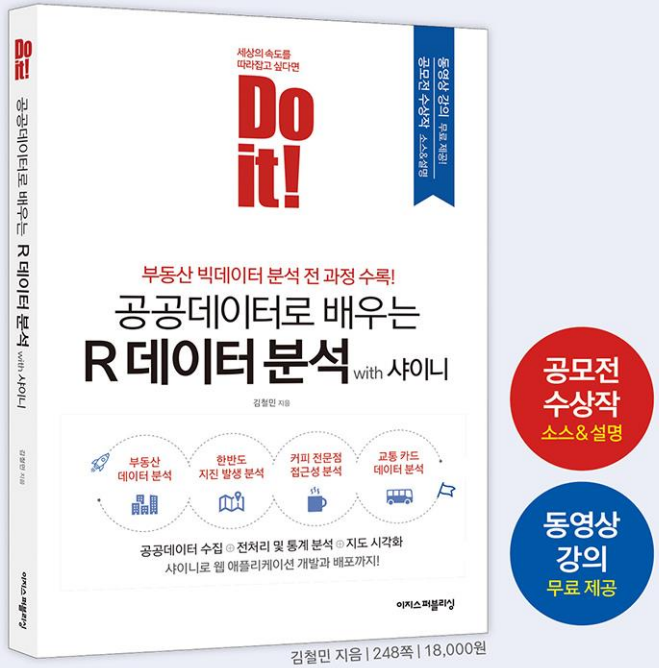
R로 공공데이터를 분석하는 전 과정 실습!

공모전 수상작으로 배우는 **R 데이터 분석**



공모전 수상작 소스&설명

동영상 강의 무료 제공

김철민 지음 | 248쪽 | 18,000원

부동산이나 대중교통 시설, 관광지 등 특정한 위치 정보를 다루는 분석 서비스를 구현할 때는 주소 데이터를 공간 좌표로 변환하는 작업이 필수입니다. 이 책에서 구현할 아파트 실거래 분석 서비스도 지도에 아파트 위치를 표시해야 하므로 좌표가 필요합니다. 이 장에서는 카카오맵에서 제공하는 API를 이용하여 문자로 된 주소를 숫자 좌표로 변환하는 지오 코딩 방법을 살펴봅니다.

05

카카오맵 API로 지오 코딩하기

05-1 지오 코딩 준비하기

05-2 주소를 좌표로 변환하는 지오 코딩

05-1 지오 코딩 준비하기

1단계 카카오 로컬 API 키 발급받기

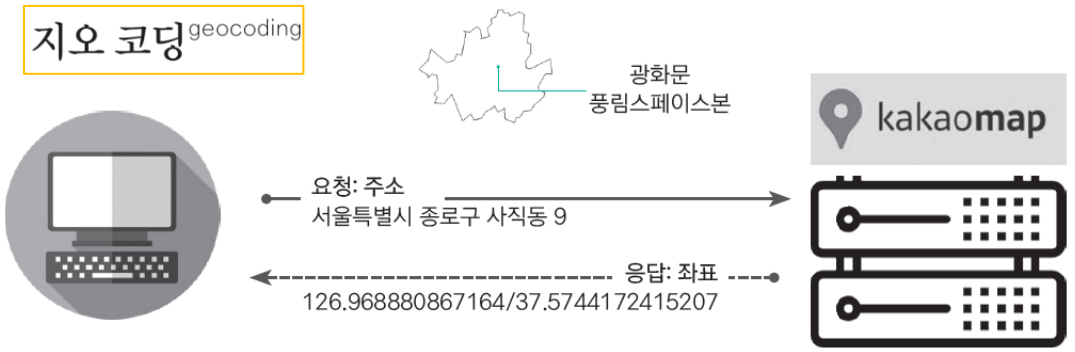


그림 5-1 지오 코딩 개념도

카카오 개발자 사이트(developers.kakao.com)

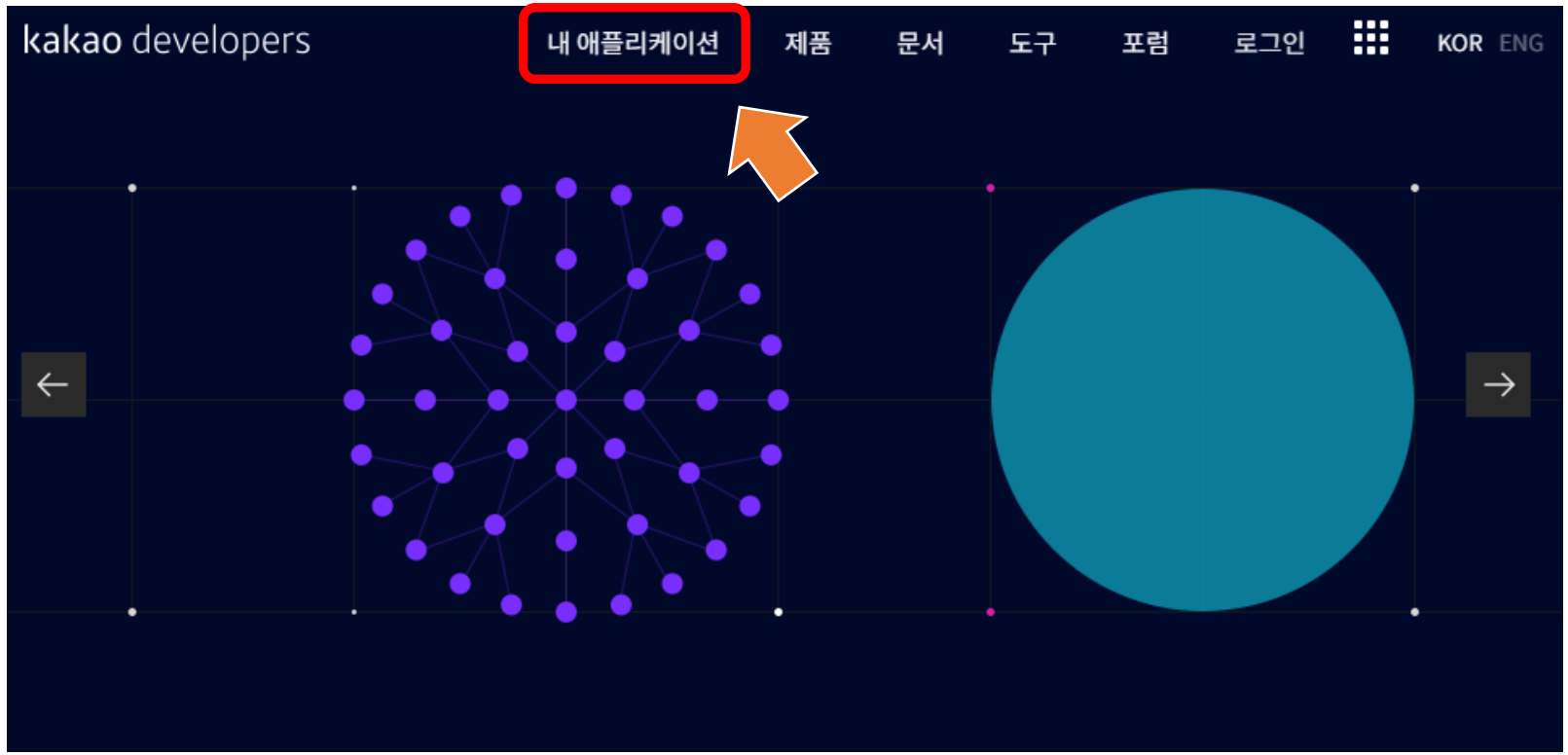


그림 5-2 카카오 개발자 사이트에서 [내 애플리케이션] 클릭

05-1 지오 코딩 준비하기

1단계 카카오 로컬 API 키 발급받기

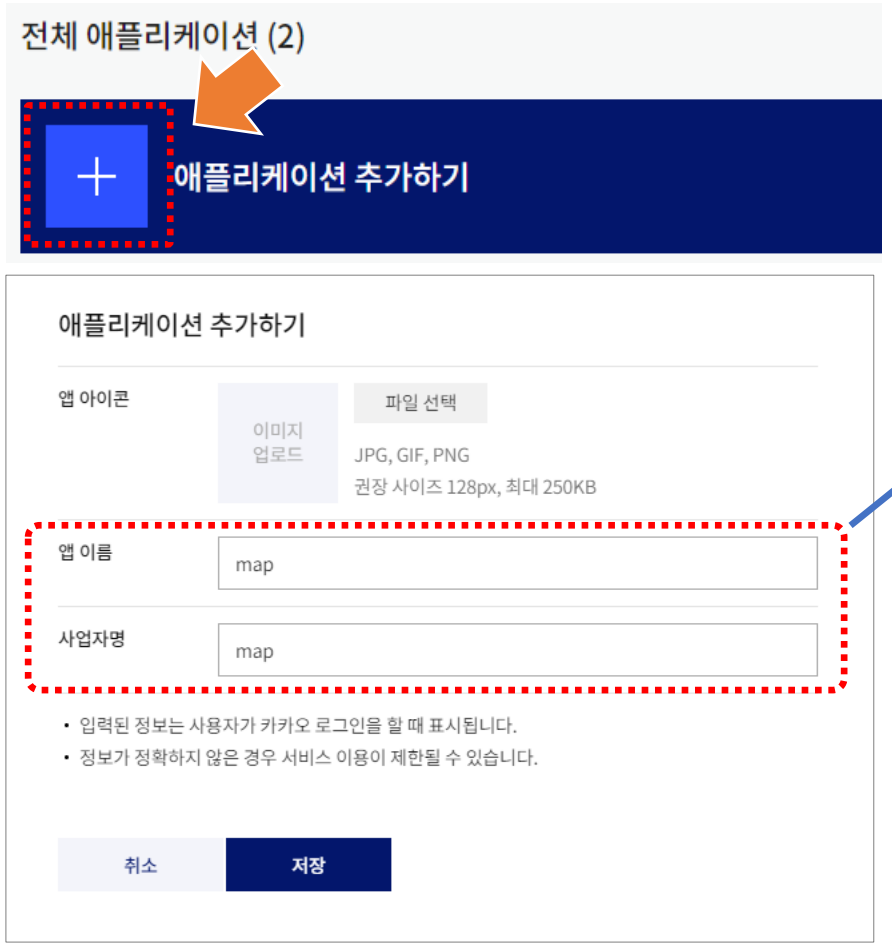


그림 5-3 애플리케이션 추가하기

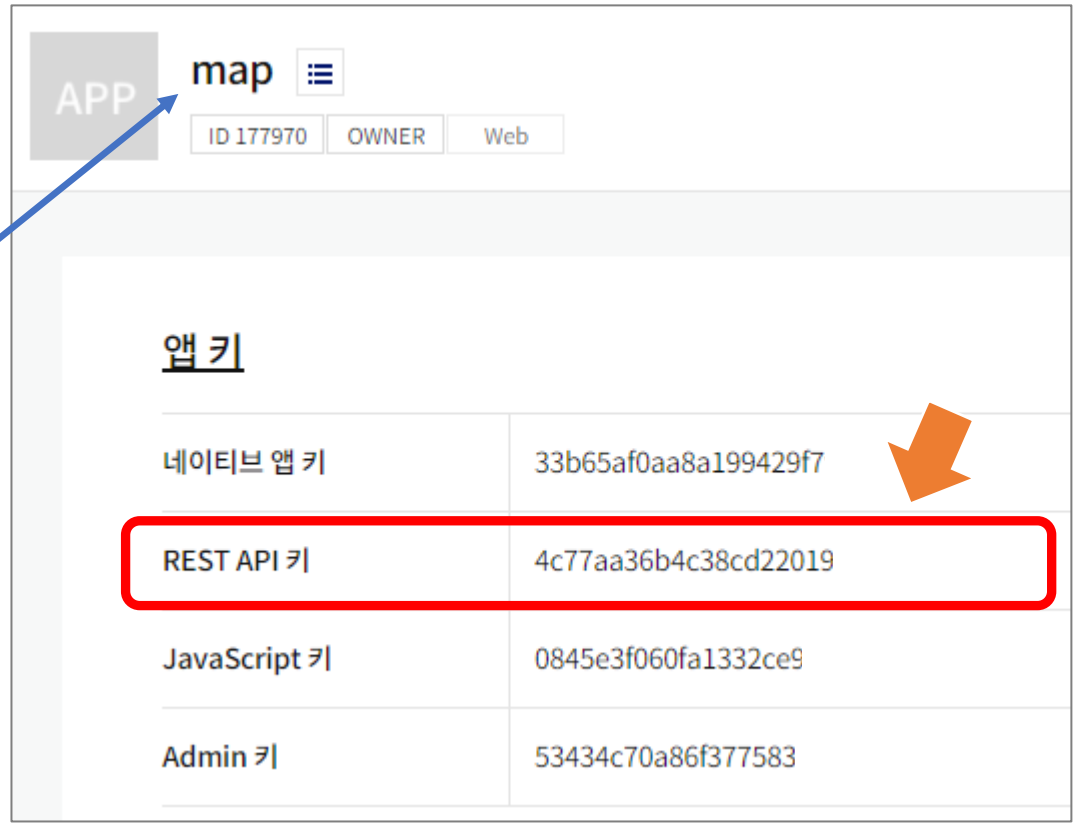


그림 5-4 REST API 키 확인하기

05-1 지오 코딩 준비하기

2단계 중복된 주소 제거하기

Do it! 고유한 주소만 추출 05_지오 코딩.R

```
08: setwd(dirname(rstudioapi::getSourceEditorContext()$path))
09: load( "./04_preprocess/04_preprocess.rdata")  # 실거래 자료 불러오기
10: apt_juso <- data.frame(apt_price$juso_jibun)    # 주소가 있는 칼럼 추출
11: apt_juso <- data.frame(apt_juso[!duplicated(apt_juso), ])   # 고유한 주소만 추출
12: head(apt_juso, 2)   # 추출 결과 확인
```

☞ 실행 결과

```
1 서울특별시 종로구 사직동 9 광화문풍림스페이스본
2 서울특별시 종로구 사직동 9-1 광화문풍림스페이스본
```

05-2 주소를 좌표로 변환하는 지오 코딩

1단계 지오 코딩하기

Do it! 지오 코딩 준비 — 05_지오 코딩.R

```
21: add_list <- list()          # 빈 리스트 생성
22: cnt <- 0                    # 반복문 카운팅 초깃값 설정
23: kakao_key = "REST API 키"   # 카카오 REST API 키
```

Do it! 라이브러리 불러오기 — 05_지오 코딩.R

```
27: library(httr)          # install.packages('httr')
28: library(rjson)         # install.packages('rjson')
29: library(data.table)    # install.packages('data.table')
30: library(dplyr)         # install.packages('dplyr')
```

- `httr`: 웹(http)으로 자료 요청
- `rjson`: 응답 결과인 JSON형 자료 처리
- `data.table`: 좌표를 테이블로 저장
- `dplyr`: 파이프라인 사용

05-2 주소를 좌표로 변환하는 지오 코딩

1단계 지오 코딩하기

Do it! for 반복문과 예외 처리 시작 — 05_지오 코딩.R

```
32: for(i in 1:nrow(apt_juso)){
33:   # 예외 처리 구문 시작
34:   tryCatch(
35:     {
```

Do it! 주소 요청 — 05_지오 코딩.R

```
36: # 주소로 좌푯값 요청
37: lon_lat <- GET(url = 'https://dapi.kakao.com/v2/local/search/address.json',
38:                query = list(query = apt_juso[i,]),
39:                add_headers(Authorization = paste0("KakaoAK ", kakao_key)))
```

- 서비스 URL: 'https://dapi.kakao.com/v2/local/search/address.json'
- 질의: list(query = apt_juso[i,])
- 헤더: add_headers(Authorization = paste0("KakaoAK ", kakao_key)))

05-2 주소를 좌표로 변환하는 지오 코딩

1단계 지오 코딩하기

Do it! 위경도 정보 추출 05_지오 코딩.R

```
40: # 위경도만 추출하여 저장
41: coordxy <- lon_lat %>% content(as = 'text') %>% fromJSON()
```

Do it! 위경도 정보 저장 05_지오 코딩.R

```
42: # 반복 횟수 카운팅
43: cnt = cnt + 1
44: # 주소, 경도, 위도 정보를 리스트로 저장
45: add_list[[cnt]] <- data.table(apt_juso = apt_juso[i,],
46:                               coord_x = coordxy$documents[[1]]$address$x,
47:                               coord_y = coordxy$documents[[1]]$address$y)
```

05-2 주소를 좌표로 변환하는 지오 코딩

1단계 지오 코딩하기

Do it! 진행상황 알림 메시지 출력 05_지오 코딩.R

```
48: # 진행 상황 알림 메시지
49: message <- paste0("[", i,"/",nrow(apt_juso),"] 번째 (",
50:   round(i/nrow(apt_juso)*100,2)," %) [", apt_juso[i,] ,"] 지오 코딩 중입니다:
51:   X= ", add_list[[cnt]]$coord_x, " / Y= ", add_list[[cnt]]$coord_y)
52: cat(message, "\n\n")
```

Do it! for 반복문과 예외 처리 종료 05_지오 코딩.R

```
53:     # 예외 처리 구문 종료
54:     }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")}
55:   )
56: }
```

☞ 실행 결과

```
[7520/7521 (99.9%)] 번째 [서울특별시 강동구 … X= 127.13152432 / Y= 37.56532325
[7521/7521 (100%)] 번째 [서울특별시 강동구 … X= 127.13152323 / Y= 37.55412345
```

05-2 주소를 좌표로 변환하는 지오 코딩

2단계 지오 코딩 결과 저장하기

Do it! 지오 코딩 결과 저장 05_지오 코딩.R

```
60: juso_geocoding <- rbindlist(add_list)   # 리스트 -> 데이터프레임 변환
61: juso_geocoding$coord_x <- as.numeric(juso_geocoding$coord_x) # 좌표 숫자형 변환
62: juso_geocoding$coord_y <- as.numeric(juso_geocoding$coord_y)
63: juso_geocoding <- na.omit(juso_geocoding)   # 결측치 제거
64: dir.create("./05_geocoding")   # 새로운 폴더 생성
65: save(juso_geocoding, file="./05_geocoding/05_juso_geocoding.rdata") # 저장
66: write.csv(juso_geocoding, "./05_geocoding/05_juso_geocoding.csv")
```

단골 코드 정리하기

• unique() 함수로 중복값 제거하기

```
library(ggplot2)
mpg <- data.frame(mpg$manufacturer)  # 제조사 이름만 추출
data.frame(mpg[!duplicated(mpg), ])  # 제조사 중복값 제거
```

• GET() 함수로 웹 페이지 자료 가져오기

```
library(httr)
library(rjson)
library(dplyr)
# GET()으로 HTML 페이지 가져오기
web_page <- GET('http://www.w3.org/Protocols/rfc2616/rfc2616.html')
web_page <- web_page %>% content(as = 'text')   # HTML 페이지 텍스트만 저장
head(web_page)    # 자료 확인
```

단골 코드 정리하기

• tryCatch() 함수로 예외 처리하기

```
inputs = list(1, 2, 3, 'four', 5, 6)   # 입력 데이터(문자와 숫자 혼합)
# 일반적인 반복문: 중간에 오류 발생으로 멈춤
for(input in inputs) {
  print(paste(input, "의 로그값은 =>", log(input)))
}
# tryCatch() 함수가 포함된 반복문: 중간에 오류 발생해도 예외 처리하고 넘어감
for(input in inputs) {
  tryCatch({
    print(paste(input, "의 로그값은 =>", log(input)))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")} )
}
```