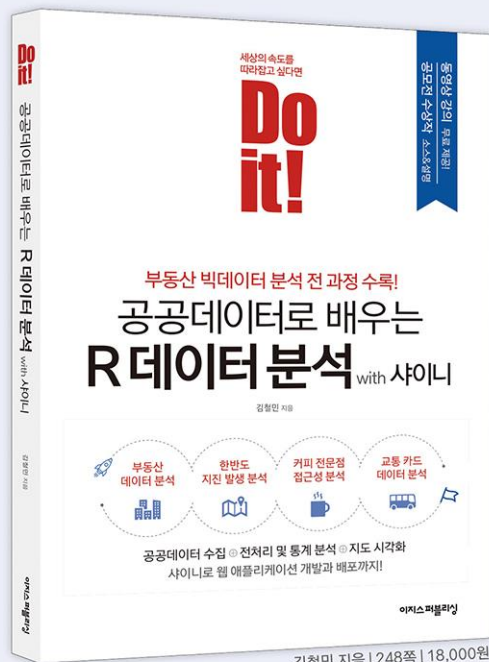


R로 공공데이터를 분석하는 전 과정 실습!  
공모전 수상작으로 배우는 R 데이터 분석



김철민 지음 | 248쪽 | 18,000원

공모전  
수상작  
소스&설명

동영상  
강의  
무료 제공



## 지오 데이터프레임 만들기

06-1 좌표계와 지오 데이터 포맷

06-2 주소와 좌표 결합하기

06-3 지오 데이터 프레임 만들기

일반적인 데이터프레임으로 공간 좌표를 다루기에는 한계가 있습니다. 이 장에서는 05장에서 수집한 위경도 데이터에 좌표계를 결합하여 지도 분석에 효과적인 지오 데이터프레임을 만드는 방법을 살펴봅니다.

## 06-1 좌표계와 지오 데이터 포맷

### 좌표계(CRS)란 무엇일까?

05장에서 지오 코딩으로 평면의 지도 위에 위치를 표현할 수 있는 위경도 정보를 수집하였습니다. 그러나 우리가 살아가는 지구는 불규칙한 타원체라서 실제 좌푯값을 표현하려면 투영<sup>projections</sup> 과정을 거쳐 보정해야 합니다. 이때 보정의 기준이 바로 좌표계<sup>CRS, coordinate reference systems</sup>입니다.

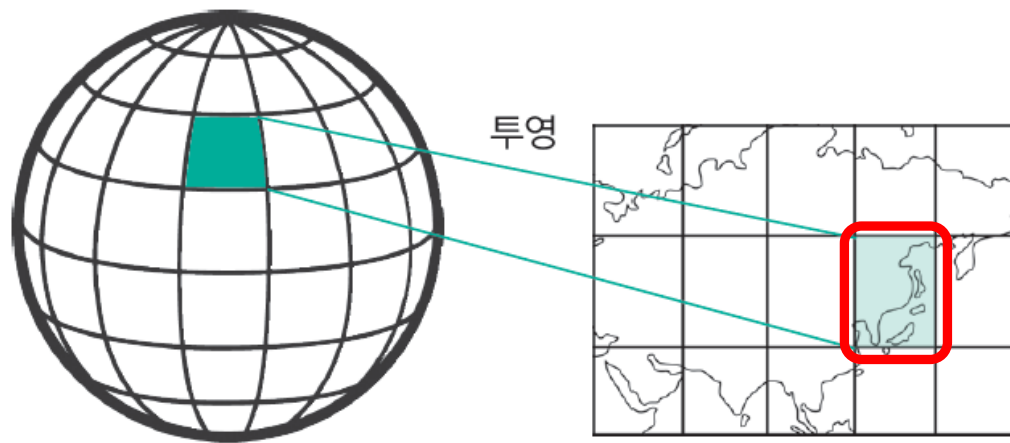


그림 5-1 좌표계 투영

지오 데이터 포맷: sp와 sf

2005년 공개된 sp 패키지는 R에서 점, 선, 면 같은 공간 정보를 처리할 목적으로 만든 데이터 포맷입니다. 테두리 상자<sup>bounding box</sup>나 좌표계 같은 다양한 정보도 함께 저장할 수 있다는 점에서 공간 분석의 새로운 길을 열어 주었습니다. 그러나 데이터 일부를 편집하거나 수정하는 것은 어렵다는 한계가 있습니다.

이러한 sp의 한계를 극복하고자 2016년 sf 패키지가 공개되었습니다. sf는 sp 패키지가 가지고 있던 기능과 속성을 그대로 이어받지만, 기존의 데이터프레임에 공간 속성<sup>geometry</sup>을 가진 칼럼을 추가함으로써 공간 데이터를 일반 데이터프레임과 비슷하게 편집하거나 수정할 수 있게 하였습니다.

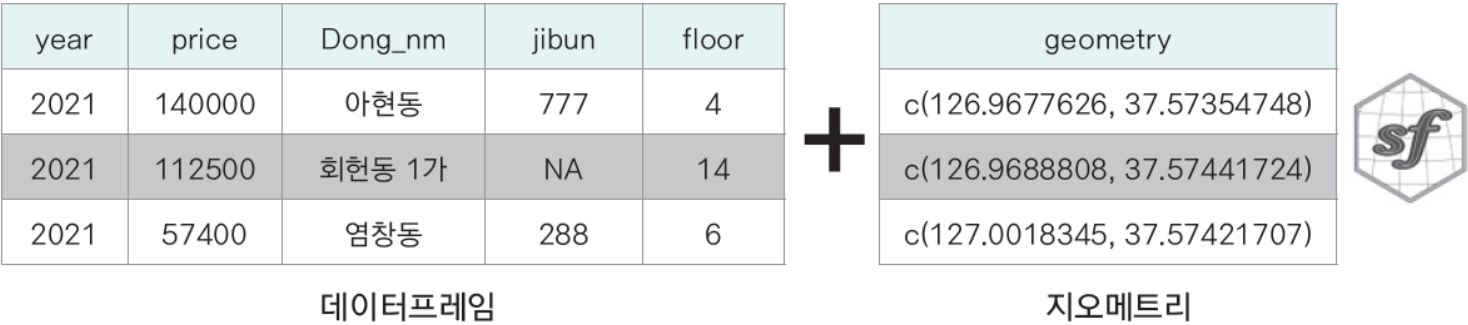


그림 5-2 sf 패키지 자료의 구성

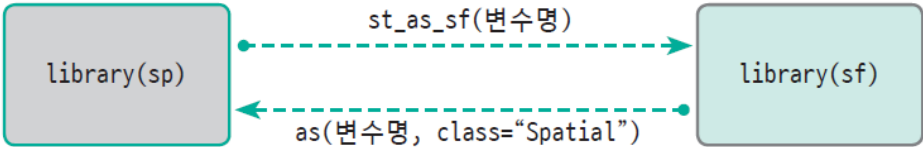


그림 5-3 sp와 sf 변환

## 06-2 주소와 좌표 결합하기

### 1단계 데이터 불러오기

#### Do it! 데이터 불러오기

06\_지오 데이터프레임.R

```
08: setwd(dirname(rstudioapi::getSourceEditorContext())$path))
09: load("./04_pre_process/04_pre_process.rdata")    # 주소 불러오기
10: load("./05_geocoding/05_juso_geocoding.rdata")   # 좌표 불러오기
```

### 2단계 주소와 좌표 결합하기

#### Do it! 주소 + 좌표 결합

06\_지오 데이터프레임.R

```
14: library(dplyr)    # install.packages('dplyr')
15: apt_price <- left_join(apt_price, juso_geocoding,
16:                        by = c("juso_jibun" = "apt_juso")) # 결합
17: apt_price <- na.omit(apt_price)    # 결측치 제거
```

## 06-3 지오 데이터프레임 만들기

### 1단계 지오 데이터프레임 생성하기

**Do it!** 지오 데이터프레임 생성

06\_지오 데이터프레임.R

```
26: library(sp)      # install.packages('sp')
27: coordinates(apt_price) <- ~coord_x + coord_y    # 좌표값 할당
28: proj4string(apt_price) <- "+proj=longlat +datum=WGS84 +no_defs"  # 좌표계(CRS) 정의
29: library(sf)      # install.packages('sf')
30: apt_price <- st_as_sf(apt_price)    # sp형 => sf형 변환
```

## 06-3 지오 데이터프레임 만들기

### 2단계 지오 데이터프레임 시각화

**Do it!** 지오 데이터프레임 시각화

06\_지오 데이터프레임.R

```
34: plot(apt_price$geometry, axes = T, pch = 1)      # 플롯 그리기
35: library(leaflet)  # install.packages('leaflet')  # 지도 그리기 라이브러리
36: leaflet() %>%
37:   addTiles() %>%
38:   addCircleMarkers(data=apt_price[1:1000,], label=~apt_nm)  # 1,000개만 그리기
```

 실행 결과



## 06-3 지오 데이터프레임 만들기

### 3단계 지오 데이터프레임 저장하기

**Do it!** 지오 데이터프레임 저장

06\_지오 데이터프레임.R

```
42: dir.create("06_geodataframe")    # 새로운 폴더 생성
43: save(apt_price, file="./06_geodataframe/06_apt_price.rdata")  # rdata 저장
44: write.csv(apt_price, "./06_geodataframe/06_apt_price.csv")    # csv 저장
```



## 단골 코드 정리하기

지오 데이터프레임 변환 과정에서 다루었던 주요 기능인 일반 데이터프레임을 지오 데이터프레임으로 만들고 지도로 시각화하는 방법을 정리해 보겠습니다.

### • 좌표가 포함된 데이터프레임 만들기

```
df <- structure(
  list(longitude = c(128.6979, 153.0046, 104.3261, 124.9019,
                     126.7328, 153.2439, 142.8673, 152.689),
        latitude = c(-7.4197, -4.7089, -6.7541, 4.7817,
                     2.1643, -5.65, 23.3882, -5.571)),
  .Names = c("coord_x", "coord_y"), class = "data.frame", row.names = c(NA, -8L))
head(df)
```

### 실행 결과

```
   coord_x coord_y
1 128.6979 -7.4197
2 153.0046 -4.7089
```





## 단골 코드 정리하기

### • 지오 데이터프레임으로 변환

```
library(sp)
coordinates(df) <- ~coord_x + coord_y # 좌표값 할당(sp형)
proj4string(df) <- "+proj=longlat +datum=WGS84 +no_defs" # 좌표계(CRS) 정의(sp형)
library(sf) # install.packages('sf')
df <- st_as_sf(df) # sp형 -> sf형 변환
head(df)
```

### ☞ 실행 결과

```
Simple feature collection with 6 features and 0 fields
Geometry type: POINT
Dimension: XY
Bounding box: xmin: 104.3261 ymin: -7.4197 xmax: 153.2439 ymax: 4.7817
CRS: +proj=longlat +datum=WGS84 +no_defs
      geometry
1 POINT (128.6979 -7.4197)
2 POINT (153.0046 -4.7089)
```



## 단골 코드 정리하기

### • 지도 시각화

```
library(leaflet)
leaflet() %>%
  addTiles() %>%
  addCircleMarkers(data=df)
```

### 실행 결과

