

HW1-1 공공데이터 포털 데이터 다뤄보기

김서준

2023 11 11

문제1. 공공데이터 포털에서 데이터 가져오기

```
#출처: https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15039250
#install.packages("xml2")
library(xml2)
serviceKey = "qzyPoJOnu9Kh1QHn8bcEq99acdp7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rnfcmeiPYPmiA%3D%3D" # 본인의 키 값을 입력하세요.
numofRows = ""
pageNo = ""
dataType = 'xml'

url_xml <- paste0('http://apis.data.go.kr/1741000/EmergencyAssemblyArea_Earthquake2/getArea1List?serviceKey=',serviceKey,
                  '&numOfRows=',numofRows,
                  '&pageNo=',pageNo,
                  '&type=', dataType)

# url 링크를 xml로 읽어들이기.
# read_xml(): xml 파일 읽어들이기.
raw_xml <- read_xml(url_xml)

# occrrncDt, dataVal 변수 모두 찾아서 저장하기
# xml_find_all(): xml 파일에서 원하는 '태그'를 찾아 저장.
xml_1 <- xml_find_all(raw_xml, '//ctprvn_nm') # 시도명
xml_2 <- xml_find_all(raw_xml, '//vt_acmdfcilty_nm') # 시설명
xml_3 <- xml_find_all(raw_xml, '//dtl_adres') # 상세 주소
xml_4 <- xml_find_all(raw_xml, '//fcilty_ar') # 시설 면적
xml_5 <- xml_find_all(raw_xml, '//ycord') # 위도
xml_6 <- xml_find_all(raw_xml, '//xcord') # 경도
xml_7 <- xml_find_all(raw_xml, '//acmdfcilty_sn') # 시설일련번호

# text로 읽어 저장하기
# xml_text(): 텍스트 추출.
ctprvn_nm <- xml_text(xml_1)
vt_acmdfcilty_nm <- xml_text(xml_2)
dtl_adres <- xml_text(xml_3)
fcilty_ar <- xml_text(xml_4)
ycord <- xml_text(xml_5)
xcord <- xml_text(xml_6)
acmdfcilty_sn <- xml_text(xml_7)
```

문제 2. 데이터프레임으로 만들기

```
shelter <- data.frame("시도명" = ctprvn_nm, "시설명" = vt_acmdfcilty_nm, "상세주소" = dtl_adres,
"시설면적" = fclty_ar, "위도" = ycord, "경도" = xcord, "시설일련번호" = acmdfcilty_sn)
head(shelter)
```

##	시도명	시설명	상세주소	시설면적
## 1	인천광역시	부평서여자중학교 운동장	인천광역시 부평구 십정동 186-519	2520
## 2	인천광역시	십정초등학교 운동장	인천광역시 부평구 십정동 298-1	1620
## 3	부산광역시	동원초등학교운동장	부산광역시 부산진구 당감동 786-2	2240
## 4	부산광역시	당감초등학교운동장	부산광역시 부산진구 당감동 271	5363
## 5	부산광역시	동평초등학교운동장	부산광역시 부산진구 당감동 705-1	9809
## 6	부산광역시	부산국제고등학교운동장	부산광역시 부산진구 당감동 897	2171

##	위도	경도	시설일련번호
## 1	37.486108	126.705455	8
## 2	37.47436571096047	126.69763569508791	51
## 3	35.17306045874873	129.03118969056064	25
## 4	35.16404012072213	129.04059131137745	6
## 5	35.167111006163374	129.03520793225897	8
## 6	35.16758349649663	129.02384548602052	15

```
dim(shelter)
```

```
## [1] 100 7
```

문제 3. 데이터 문제점 파악 및 제거

```
str(shelter)
```

```
## 'data.frame': 100 obs. of 7 variables:
## $ 시도명 : chr "인천광역시" "인천광역시" "부산광역시" "부산광역시" ...
## $ 시설명 : chr "부평서여자중학교 운동장" "십정초등학교 운동장" "동원초등학교운동장"
"당감초등학교운동장" ...
## $ 상세주소 : chr "인천광역시 부평구 십정동 186-519" "인천광역시 부평구 십정동 298-1"
"부산광역시 부산진구 당감동 786-2" "부산광역시 부산진구 당감동 271" ...
## $ 시설면적 : chr "2520" "1620" "2240" "5363" ...
## $ 위도 : chr "37.486108" "37.47436571096047" "35.17306045874873" "35.1640401207221
3" ...
## $ 경도 : chr "126.705455" "126.69763569508791" "129.03118969056064" "129.0405913113
7745" ...
## $ 시설일련번호: chr "8" "51" "25" "6" ...
```

문제점: 시설면적, 위도, 경도, 시설일련번호 등 numeric이나 int형 자료형이 필요한 컬럼들이 있지만 해당 컬럼의 value들은 모두 chr 형태로 저장되어 있어서 수들의 연산이나 평균, 중앙값등의 점추정치를 알기 어렵다.

해결방법: as.자료형(데이터프레임이름\$컬럼명) 을 통해 알맞은 자료형으로 바꾸어준다. 데이터를 잠시 살펴본 결과 시설면적과 시설일련번호는 정수들로만 구성되어 있기에 as.integer()함수를 사용 실수로 이루어진 위도와 경도는 as.numeric() 함수를 사용한다.

```
shelter$시설면적 <- as.integer(shelter$시설면적)
shelter$시설일련번호 <- as.integer(shelter$시설일련번호)
shelter$위도 <- as.numeric(shelter$위도)
shelter$경도 <- as.numeric(shelter$경도)
class(shelter$시설면적)
```

```
## [1] "integer"
```

```
class(shelter$시설일련번호)
```

```
## [1] "integer"
```

```
class(shelter$위도)
```

```
## [1] "numeric"
```

```
class(shelter$경도)
```

```
## [1] "numeric"
```

문제 4. 반복문을 사용하여 더 많은 데이터 가져오기

```

library(xml2)
shelter <- data.frame( 시도명 = character(), 시설명 = character(), 상세주소 = character(), 시
설면적 = integer(), 위도 = numeric(), 경도 = numeric(), 시설일련번호 = numeric(), stringsAsFacto
rs =
FALSE
)
serviceKey = "qzyPoUonu9Kh1QHn8bcEq99acdp7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rnfc
meiPYPMiA%3D%3D" # 본인의 키 값을 입력하세요.
numofRows = ""
for (i in 1:10){
  pageNo = i
  dataType = 'xml'

  url_xml <- paste0('http://apis.data.go.kr/1741000/EmergencyAssemblyArea_Earthquake2/getArea1L
ist?serviceKey=', serviceKey,
                    '&numOfRows=', numofRows,
                    '&pageNo=', pageNo,
                    '&type=', dataType)

  # urls 링크를 xml로 읽어들이기.
  # read_xml(): xml 파일 읽어들이기.
  raw_xml <- read_xml(url_xml)

  # occrrncDt, dataVal 변수 모두 찾아서 저장하기
  # xml_find_all(): xml 파일에서 원하는 '태그'를 찾아 저장.
  xml_1 <- xml_find_all(raw_xml, '//ctprvn_nm') # 시도명
  xml_2 <- xml_find_all(raw_xml, '//vt_acmdfcilty_nm') # 시설명
  xml_3 <- xml_find_all(raw_xml, '//dtl_adres') # 상세 주소
  xml_4 <- xml_find_all(raw_xml, '//fcilty_ar') # 시설 면적
  xml_5 <- xml_find_all(raw_xml, '//ycord') # 위도
  xml_6 <- xml_find_all(raw_xml, '//xcord') # 경도
  xml_7 <- xml_find_all(raw_xml, '//acmdfcilty_sn') # 시설일련번호

  # text로 읽어 저장하기
  # xml_text(): 텍스트 추출.
  ctprvn_nm <- xml_text(xml_1)
  vt_acmdfcilty_nm <- xml_text(xml_2)
  dtl_adres <- xml_text(xml_3)
  fcilty_ar <- xml_text(xml_4)
  ycord <- xml_text(xml_5)
  xcord <- xml_text(xml_6)
  acmdfcilty_sn <- xml_text(xml_7)
  shelter2 <- data.frame("시도명" = ctprvn_nm, "시설명" = vt_acmdfcilty_nm, "상세주소" = dtl_adr
es, "시설면적" = fcilty_ar, "위도" = ycord, "경도" = xcord, "시설일련번호" = acmdfcilty_sn, strin
gsAsFactors = FALSE)
  shelter2$시설면적 <- as.integer(shelter2$시설면적)
  shelter2$시설일련번호 <- as.integer(shelter2$시설일련번호)
  shelter2$위도 <- as.numeric(shelter2$위도)
  shelter2$경도 <- as.numeric(shelter2$경도)
  shelter <- rbind(shelter, shelter2)
}
dim(shelter)

```

[1] 1000 7

head(shelter)

##	시도명	시설명	상세주소	시설면적
## 1	인천광역시	부평서여자중학교 운동장	인천광역시 부평구 십정동 186-519	2520
## 2	인천광역시	십정초등학교 운동장	인천광역시 부평구 십정동 298-1	1620
## 3	부산광역시	동원초등학교운동장	부산광역시 부산진구 당감동 786-2	2240
## 4	부산광역시	당감초등학교운동장	부산광역시 부산진구 당감동 271	5363
## 5	부산광역시	동평초등학교운동장	부산광역시 부산진구 당감동 705-1	9809
## 6	부산광역시	부산국제고등학교운동장	부산광역시 부산진구 당감동 897	2171
##	위도	경도	시설일련번호	
## 1	37.48611	126.7055	8	
## 2	37.47437	126.6976	51	
## 3	35.17306	129.0312	25	
## 4	35.16404	129.0406	6	
## 5	35.16711	129.0352	8	
## 6	35.16758	129.0238	15	

문제 5. pageNo 파라미터를 설정하여 설정한 pageNo까지의 데이터를 가져와 결과를 return 할 수 있는 사용자 정의 함수를 작성할 것

```

set_the_num_of_pages <- function(x) {
  library(xml2)
  shelter <- data.frame(
    시도명 = character(),
    시설명 = character(),
    상세주소 = character(),
    시설면적 = integer(),
    위도 = numeric(),
    경도 = numeric(),
    시설일련번호 = numeric(),
    stringsAsFactors = FALSE
  )
  serviceKey = "qzyPoUonu9Kh1QHn8bcEq99acdp7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rn
fcmeiPYPMiA%3D%3D"
  numofRows = ""

  for (i in 1:x) {
    pageNo = i
    dataType = 'xml'

    url_xml <- paste0('http://apis.data.go.kr/1741000/EmergencyAssemblyArea_Earthquake2/getArea
1List?serviceKey=', serviceKey,
                      '&numOfRows=', numofRows,
                      '&pageNo=', pageNo,
                      '&type=', dataType)

    # urls 링크를 xml로 읽어들이기.
    # read_xml(): xml 파일 읽어들이기.
    raw_xml <- read_xml(url_xml)

    # occrrncDt, dataVal 변수 모두 찾아서 저장하기
    # xml_find_all(): xml 파일에서 원하는 '태그'를 찾아 저장.
    xml_1 <- xml_find_all(raw_xml, '//ctprvn_nm') # 시도명
    xml_2 <- xml_find_all(raw_xml, '//vt_acmdfcilty_nm') # 시설명
    xml_3 <- xml_find_all(raw_xml, '//dtl_adres') # 상세 주소
    xml_4 <- xml_find_all(raw_xml, '//fcilty_ar') # 시설 면적
    xml_5 <- xml_find_all(raw_xml, '//ycord') # 위도
    xml_6 <- xml_find_all(raw_xml, '//xcord') # 경도
    xml_7 <- xml_find_all(raw_xml, '//acmdfcilty_sn') # 시설일련번호

    # text로 읽어 저장하기
    # xml_text(): 텍스트 추출.
    ctprvn_nm <- xml_text(xml_1)
    vt_acmdfcilty_nm <- xml_text(xml_2)
    dtl_adres <- xml_text(xml_3)
    fcilty_ar <- xml_text(xml_4)
    ycord <- xml_text(xml_5)
    xcord <- xml_text(xml_6)
    acmdfcilty_sn <- xml_text(xml_7)
    shelter2 <- data.frame(
      "시도명" = ctprvn_nm,
      "시설명" = vt_acmdfcilty_nm,
      "상세주소" = dtl_adres,
      "시설면적" = fcilty_ar,
      "위도" = ycord,

```

```

"경도" = xcord,
"시설일련번호" = acmdfcilty_sn,
stringsAsFactors = FALSE
)
shelter2$시설면적 <- as.integer(shelter2$시설면적)
shelter2$시설일련번호 <- as.integer(shelter2$시설일련번호)
shelter2$위도 <- as.numeric(shelter2$위도)
shelter2$경도 <- as.numeric(shelter2$경도)
shelter <- rbind(shelter, shelter2)
}
return(shelter)
}
head(set_the_num_of_pages(3))

```

##	시도명	시설명	상세주소	시설면적
## 1	인천광역시	부평서여자중학교 운동장	인천광역시 부평구 십정동 186-519	2520
## 2	인천광역시	십정초등학교 운동장	인천광역시 부평구 십정동 298-1	1620
## 3	부산광역시	동원초등학교운동장	부산광역시 부산진구 당감동 786-2	2240
## 4	부산광역시	당감초등학교운동장	부산광역시 부산진구 당감동 271	5363
## 5	부산광역시	동평초등학교운동장	부산광역시 부산진구 당감동 705-1	9809
## 6	부산광역시	부산국제고등학교운동장	부산광역시 부산진구 당감동 897	2171
##	위도	경도	시설일련번호	
## 1	37.48611	126.7055	8	
## 2	37.47437	126.6976	51	
## 3	35.17306	129.0312	25	
## 4	35.16404	129.0406	6	
## 5	35.16711	129.0352	8	
## 6	35.16758	129.0238	15	

```
dim(set_the_num_of_pages(3))
```

```
## [1] 300 7
```

문제 6. json 형태로 데이터 가져오기

```

library(jsonlite)
set_the_num_of_pages_json <- function(x) {
  df <- data.frame(
    시도명 = character(),
    시설명 = character(),
    상세주소 = character(),
    시설면적 = integer(),
    위도 = numeric(),
    경도 = numeric(),
    시설일련번호 = numeric(),
    stringsAsFactors = FALSE
  )
  serviceKey = "qzyPoUonu9Kh1QHn8bcEq99acdp7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rn
fcmeiPYPMiA%3D%3D"
  numofRows = ""

  for (i in 1:x) {
    pageNo = i
    dataType = 'json'

    url_json <- paste0('http://apis.data.go.kr/1741000/EmergencyAssemblyArea_Earthquake2/getArea
a1List?serviceKey=', serviceKey,
                        '&numOfRows=', numofRows,
                        '&pageNo=', pageNo,
                        '&type=', dataType)

    df2 <- data.frame(fromJSON(url_json))
    df2 <- df2$EarthquakeOutdoorsShelter$row[2]
    df2 <- data.frame(
      "시도명" = ctprvn_nm,
      "시설명" = vt_acmdfcilty_nm,
      "상세주소" = dtl_adres,
      "시설면적" = fcilty_ar,
      "위도" = ycord,
      "경도" = xcord,
      "시설일련번호" = acmdfcilty_sn,
      stringsAsFactors = FALSE
    )
    df2$시설면적 <- as.integer(df2$시설면적)
    df2$시설일련번호 <- as.integer(df2$시설일련번호)
    df2$위도 <- as.numeric(df2$위도)
    df2$경도 <- as.numeric(df2$경도)
    df <- rbind(df, df2)
  }
  return(df)
}
head(set_the_num_of_pages_json(30))

```


##	시도명	시설명	상세주소
## 1	경상북도	동해초등학교 운동장	경상북도 포항시 남구 동해면 도구리 644-2
## 2	경상북도	대보1리 마을회관	경상북도 포항시 남구 호미곶면 대보리 89-4
## 3	경상북도	대보초등학교 운동장	경상북도 포항시 남구 호미곶면 구만리 123-0
## 4	경상북도	호미곶면 행정복지센터	경상북도 포항시 남구 호미곶면 구만리 47-6
## 5	경상북도	구만2리 마을회관	경상북도 포항시 남구 호미곶면 구만리 576-16
## 6	경상북도	대동배1리 마을회관	경상북도 포항시 남구 호미곶면 대동배리 393-0
##	시설면적	위도	경도 시설일련번호
## 1	4500	35.98837	129.4410 184
## 2	300	36.06847	129.5672 232
## 3	4000	36.07817	129.5572 230
## 4	1200	36.08057	129.5557 229
## 5	200	36.08036	129.5506 455
## 6	200	36.05703	129.5327 227

```
dim(set_the_num_of_pages_json(30))
```

```
## [1] 3000 7
```

문제 7

```

library(xml2)
shelter <- data.frame( 시도명 = character(), 시설명 = character(), 상세주소 = character(), 시
설면적 = integer(), 위도 = numeric(), 경도 = numeric(), 시설일련번호 = numeric(), stringsAsFacto
rs =
FALSE
)
serviceKey = "qzyPoUonu9Kh1QHn8bcEq99acdp7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rnfc
meiPYPmiA%3D%3D" # 본인의 키 값을 입력하세요.
numofRows = ""
for (i in 1:10){
  pageNo = i
  dataType = 'xml'

  url_xml <- paste0('http://apis.data.go.kr/1741000/EmergencyAssemblyArea_Earthquake2/getArea1L
ist?serviceKey=', serviceKey,
                    '&numOfRows=', numofRows,
                    '&pageNo=', pageNo,
                    '&type=', dataType)

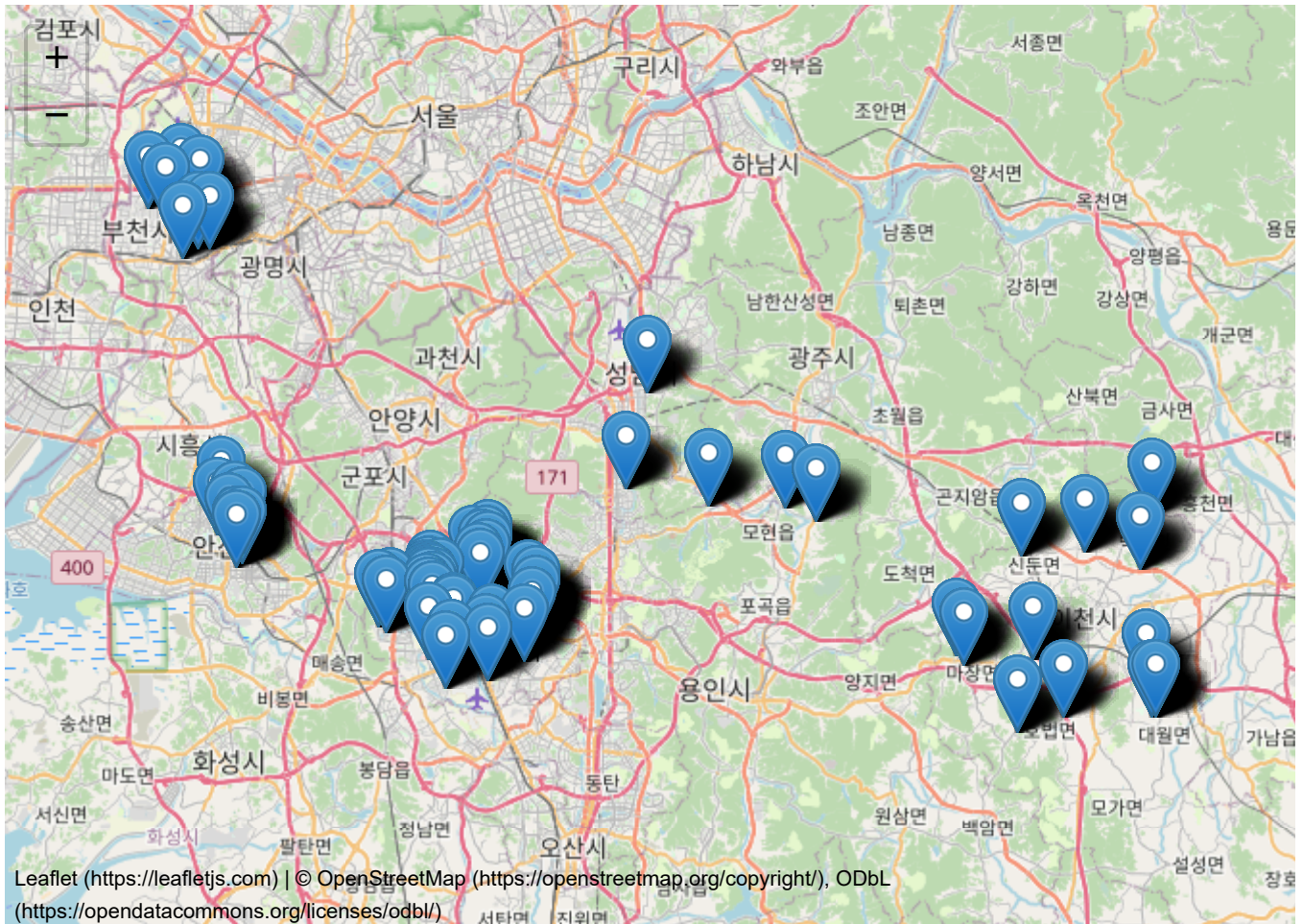
  # urls 링크를 xml로 읽어들이기.
  # read_xml(): xml 파일 읽어들이기.
  raw_xml <- read_xml(url_xml)

  # occrrncDt, dataVal 변수 모두 찾아서 저장하기
  # xml_find_all(): xml 파일에서 원하는 '태그'를 찾아 저장.
  xml_1 <- xml_find_all(raw_xml, '//ctprvn_nm') # 시도명
  xml_2 <- xml_find_all(raw_xml, '//vt_acmdfcilty_nm') # 시설명
  xml_3 <- xml_find_all(raw_xml, '//dtl_adres') # 상세 주소
  xml_4 <- xml_find_all(raw_xml, '//fcilty_ar') # 시설 면적
  xml_5 <- xml_find_all(raw_xml, '//ycord') # 위도
  xml_6 <- xml_find_all(raw_xml, '//xcord') # 경도
  xml_7 <- xml_find_all(raw_xml, '//acmdfcilty_sn') # 시설일련번호

  # text로 읽어 저장하기
  # xml_text(): 텍스트 추출.
  ctprvn_nm <- xml_text(xml_1)
  vt_acmdfcilty_nm <- xml_text(xml_2)
  dtl_adres <- xml_text(xml_3)
  fcilty_ar <- xml_text(xml_4)
  ycord <- xml_text(xml_5)
  xcord <- xml_text(xml_6)
  acmdfcilty_sn <- xml_text(xml_7)
  shelter2 <- data.frame("시도명" = ctprvn_nm, "시설명" = vt_acmdfcilty_nm, "상세주소" = dtl_adr
es, "시설면적" = fcilty_ar, "위도" = ycord, "경도" = xcord, "시설일련번호" = acmdfcilty_sn, strin
gsAsFactors = FALSE)
  shelter2$시설면적 <- as.integer(shelter2$시설면적)
  shelter2$시설일련번호 <- as.integer(shelter2$시설일련번호)
  shelter2$위도 <- as.numeric(shelter2$위도)
  shelter2$경도 <- as.numeric(shelter2$경도)
  shelter <- rbind(shelter, shelter2)
}
#install.packages("leaflet")
library(leaflet)

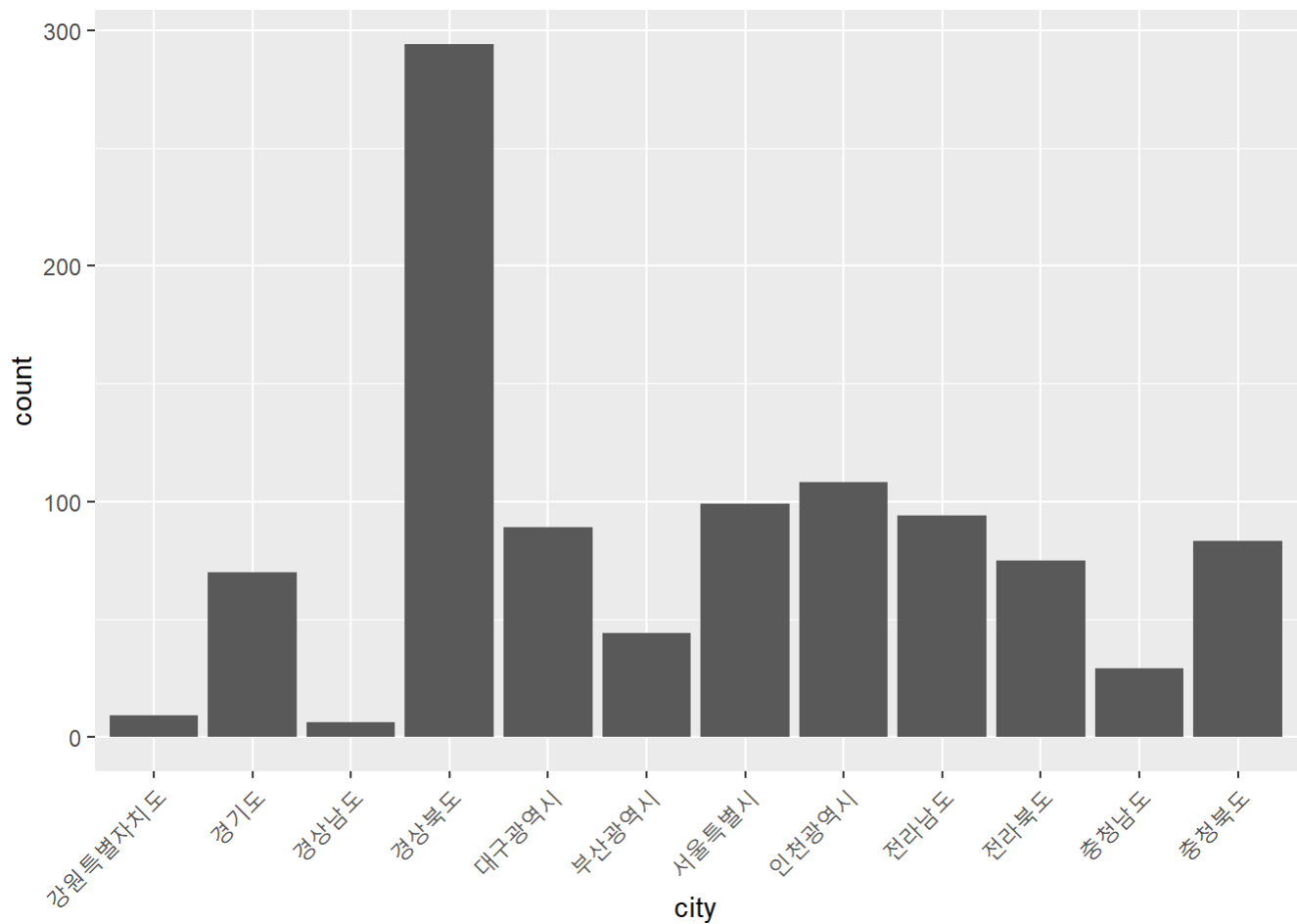
```

```
my_map <- leaflet()
my_map_1 <- addTiles(my_map)
my_map_2 <- addMarkers(my_map_1, lng = shelter[(shelter$시도명 == "경기도"),6], lat = shelter
[(shelter$시도명 == "경기도"),5], popup = shelter$시설명)
my_map_2
```



문제 8. 데이터 시각화

```
city <- shelter$시도명
library(ggplot2)
ggplot(data = shelter, mapping = aes(x = city)) + geom_bar()+theme(axis.text.x = element_text(a
ngle = 45, hjust = 1))
```



#x안에 들어갈 변수는 대피소가 위치한 "시도명"으로 하였다.

#이에 따라 x축에는 "shelter"라는 데이터 프레임에 저장된 모든 "시도명"이 위치하고 y축에는 "시도별"로 몇개씩의 대피소가 있는지를 보여준다.

문제 9.

```
#출처: https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15096654
library(jsonlite)
serviceKey <- "qzyPoUonu9Kh1QHn8bcEq99acdP7FQTJeKtS5NC%2BNGbPLfIMbSlzKyAd9W0qLoypM%2F36S7wD0rnf
cmeiPYPmiA%3D%3D"
numofRows <- "1580"
pageNo <- ""
dataType <- 'json'

url_json <- paste0('http://apis.data.go.kr/3740000/suwonEvChrstn/getdatalist?serviceKey=',servi
ceKey,
                    '&numOfRows=',numofRows,
                    '&pageNo=',pageNo,
                    '&type=', dataType)

result <- fromJSON(url_json)
df <- result$items

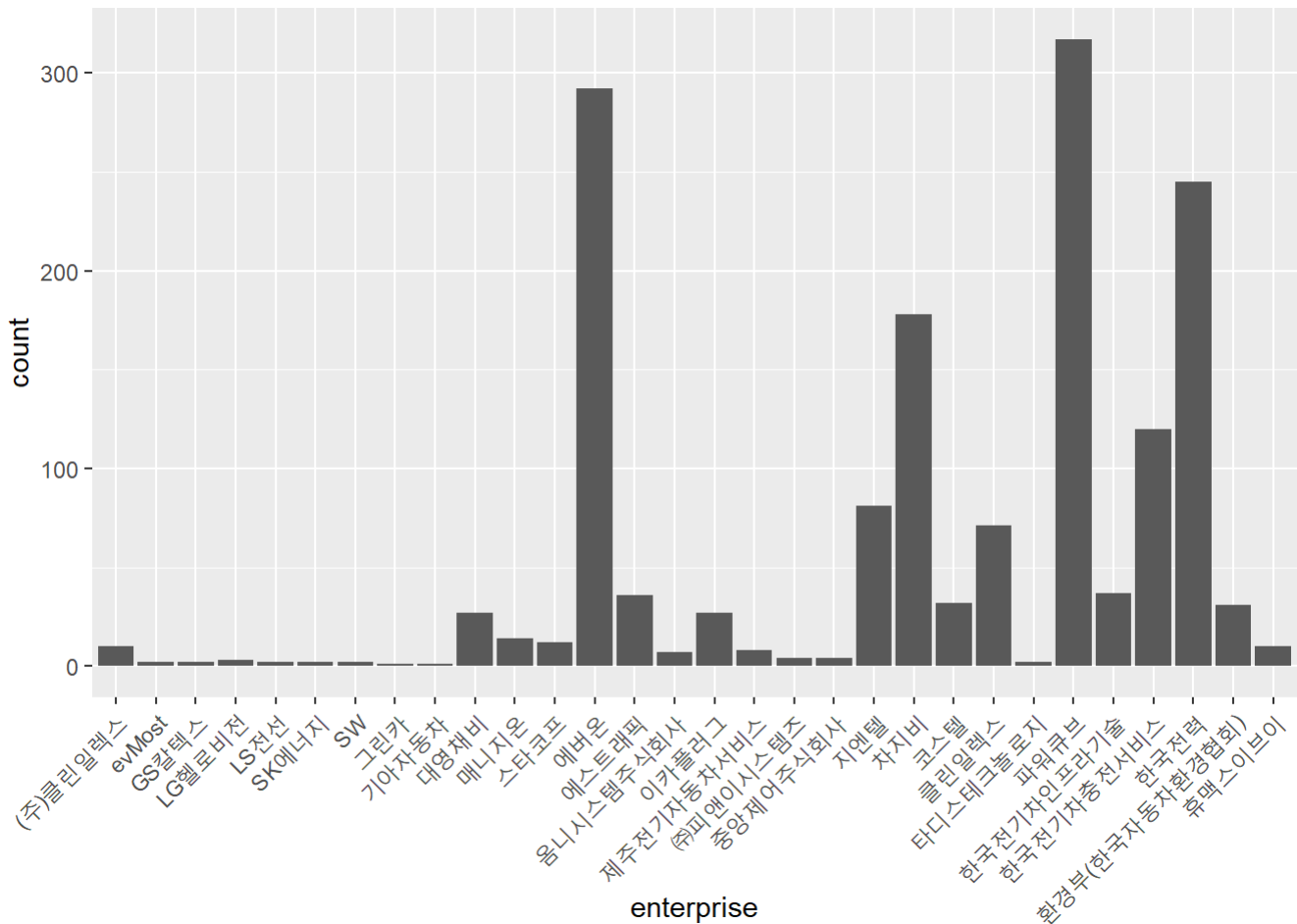
df2 <- data.frame(
  "충전소명" = df$chrstnNm,
  "충전기개수" = df$charger_status,
  "관리업체명" = df$manage_entrps_nm,
  "충전기용량" = df$charger_capacity,
  "충전기타입" = df$chrstnType,
  "위도" = df$latitude,
  "경도" = df$longitude,
  stringsAsFactors = FALSE
)

df2$충전기개수 <- as.integer(df2$충전기개수)
df2$위도 <- as.numeric(df2$위도)
df2$경도 <- as.numeric(df2$경도)
head(df2)
```

```
##                충전소명 충전기개수                관리업체명
## 1 SK엔크린 경진셀프주유소                2 환경부(한국자동차환경협회)
## 2                롯데마트 천천점                2 환경부(한국자동차환경협회)
## 3                롯데마트 천천점                2 환경부(한국자동차환경협회)
## 4                수원종합운동장                2 환경부(한국자동차환경협회)
## 5                수원종합운동장                2 환경부(한국자동차환경협회)
## 6                수원종합운동장                2 환경부(한국자동차환경협회)
##                충전기용량                충전기타입                위도
## 1 DC차데모(50kW)/AC3상(43kW)/DC콤보(50kW) DC차데모+AC3상+DC콤보 37.30813
## 2 DC차데모(50kW)/AC3상(43kW)/DC콤보(50kW) DC차데모+AC3상+DC콤보 37.29595
## 3 DC차데모(50kW)/AC3상(43kW)/DC콤보(50kW) DC차데모+AC3상+DC콤보 37.29595
## 4                DC콤보(100kW)                DC콤보 37.29933
## 5                DC콤보(100kW)                DC콤보 37.29933
## 6                DC콤보(100kW)                DC콤보 37.29933
##                경도
## 1 126.9965
## 2 126.9825
## 3 126.9825
## 4 127.0078
## 5 127.0078
## 6 127.0078
```

문제 10

```
enterprise <- df2$관리업체명 #관리업체명컬럼에 있는 데이터를 enterprise(관리업체) 라는 변수에
담음
library(ggplot2)#시각화 패키지 ggplot2사용
ggplot(data = df2, mapping = aes(x = enterprise)) + geom_bar()+theme(axis.text.x = element_text
(angle = 45, hjust = 1)) #데이터는 데이터프레임 df2를 사용하고 x축엔 관리업체들의 이름이 y축엔
각 관리업체들이 관리하는 충전소의 갯수를 나타내는 그래프를 그린다. 텍스트 겹침을 피하기 위해 45
도 돌리고 간격을 조절한다.
```



가장 많은수의 충전소를 관리하는 업체를 알아본ek.

데이터프레임에서 관리업체명의 빈도수 계산

```
company_counts <- table(df2$관리업체명)
```

가장 많은 빈도수를 가진 관리업체명과 그 수

```
most_company <- names(company_counts)[company_counts == max(company_counts)]
```

```
most_num <- max(company_counts)
```

가장 적은 빈도수를 가진 관리업체명과 그 수

```
least_company <- names(company_counts)[company_counts == min(company_counts)]
```

```
least_num <- min(company_counts)
```

```
cat("가장 많은 관리업체명:", most_company, most_num, "\n")
```

가장 많은 관리업체명: 파워큐브 317

```
cat("가장 적은 관리업체명:", least_company, least_num, "\n")
```

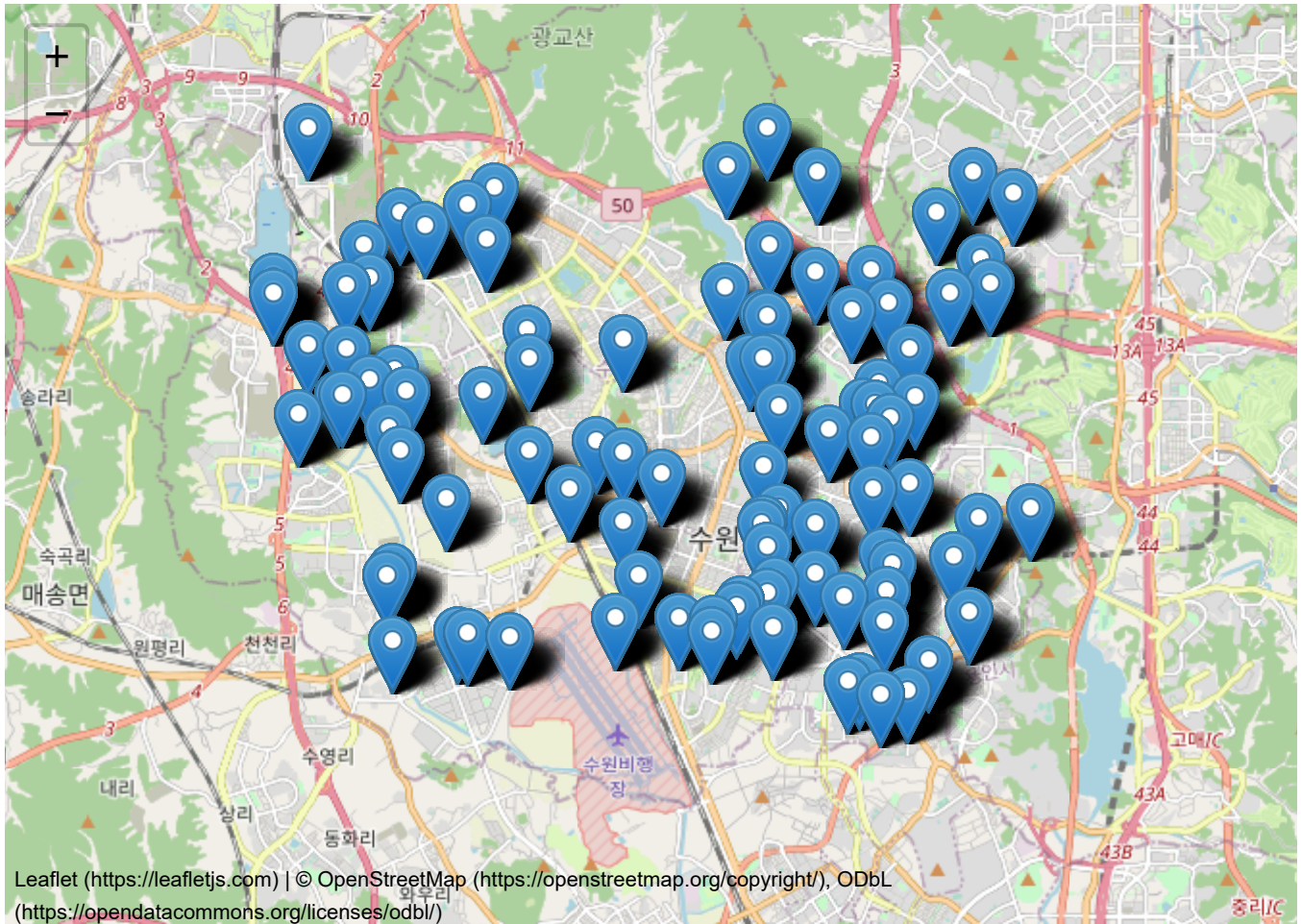
가장 적은 관리업체명: 그린카 기아자동차 1

그래프를 통해 볼 때 가장 많은 관리업체명은 파워큐브로 317개의 충전소를 운영하고

#가장 적은 관리업체명인 그린카 기아자동차는 1개의 충전소를 운영함을 알 수 있다.

문제 10

```
library(leaflet)#지도 시각화 라이브러리를 불러온다.
my_map <- leaflet() # 빈 leaflet 맵 객체 생성
my_map_1 <- addTiles(my_map) # 기본 배경 지도 추가
my_map_2 <- addMarkers(my_map_1, lng = sample(df2$경도,100), lat = sample(df2$위도,100), popup
= df2$충전소명)
#지도에 표시할 위치들을 만드는데 경도는 df2에 경도중 100개를 랜덤으로 가져오고 위도는 df2의 위
도데이터중 100개를 랜덤으로 가져온다. 팝업에 마우스를 가져갔을때 나오는 것은 시설명 컬럼의 데이
터로 한다.
my_map_2
```



#지도를 통해 충전소들의 위치를 본다면 대략적으로 수원의 북쪽보다는 남쪽에 더 많은 전지차충전소가 있고
#서쪽보다는 동쪽에 더 많은 전기차 충전소가 있음을 알 수 있다.

```
mean(df2$충전기개수)#충전기개수의 평균
```

```
## [1] 2.437342
```

```
df2$충전기용량 <- gsub("KW", "kW", df2$충전기용량) #충전기용량컬럼에 KW 와 kW가 대소문자 구문없
이 혼동되어 gsub함수를 통해 kW로 통일시켜줌
result <- aggregate(충전기개수 ~ 충전기용량, data = df2, mean) #충전기 용량별에 따른 충전기 개
수의 평균을 데이터 프레임으로 만들
result
```


##	충전기용량	충전기개수
## 1	DC차데모 (100kW)/DC콤보 (100kW)	2.000000
## 2	DC차데모 (50kW)/AC3상 (43kW)	3.500000
## 3	DC차데모 (50kW)/AC3상 (43kW)/DC콤보 (100kW)	2.000000
## 4	DC차데모 (50kW)/AC3상 (43kW)/DC콤보 (50kW)	2.666667
## 5	DC차데모 (50kW)/AC3상 (50kW)/DC콤보 (50kW)	5.500000
## 6	DC차데모 (50kW)/DC콤보 (100kW)	2.000000
## 7	DC차데모 (50kW)/DC콤보 (50kW)	2.687500
## 8	DC콤보 (100kW)	2.071429
## 9	DC콤보 (50kW)	3.272727
## 10	완속 (3.5kW)	5.500000
## 11	완속 (3kW)	4.625000
## 12	완속 (7.5kW)	2.000000
## 13	완속 (7kW)	2.387120