L15. Geospatial data (2)

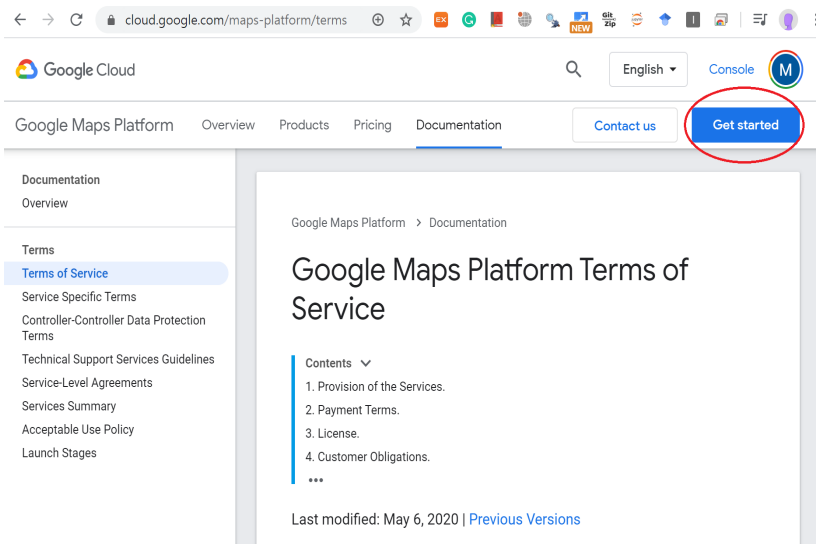
Sim, Min Kyu, Ph.D., mksim@seoultech.ac.kr

SEOULTECH 서울과학기술대학교 데이터사이언스학과

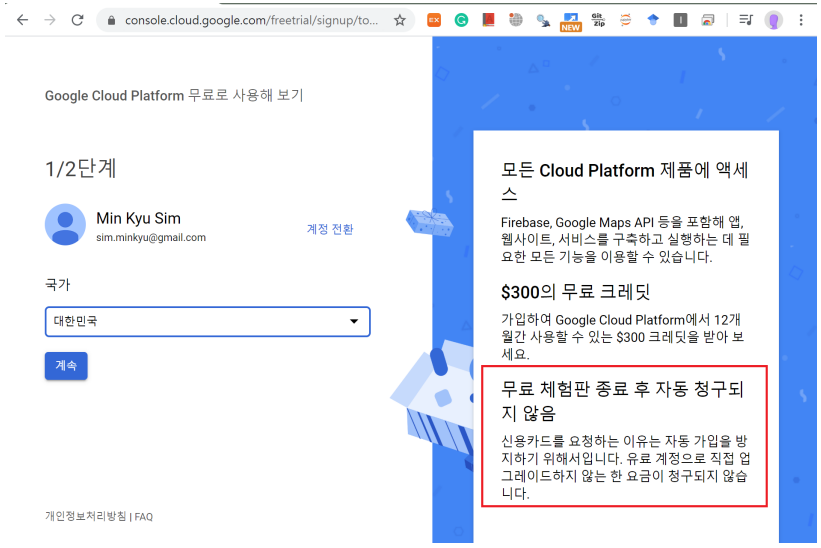
I. Prepare your canvas - maps

Get API key from google

1. https://cloud.google.com/maps-platform/terms에 접속하여 'Get Started' 버튼 누름

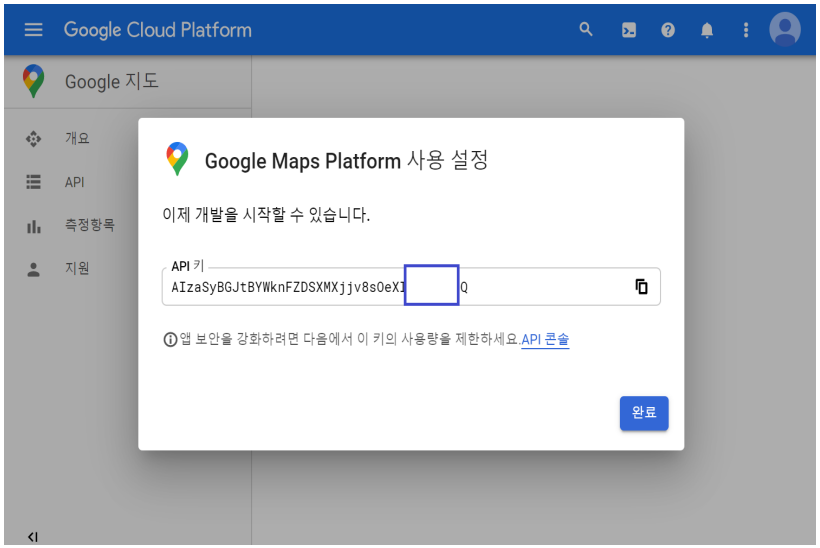


2. 신용카드 정보 입력 (요금 부과는 되지 않음)



3. 상식적인 방법으로 계속 진행

4. API key를 받기 완료



Get started with ggmap

```
install.packages("ggmap")
```

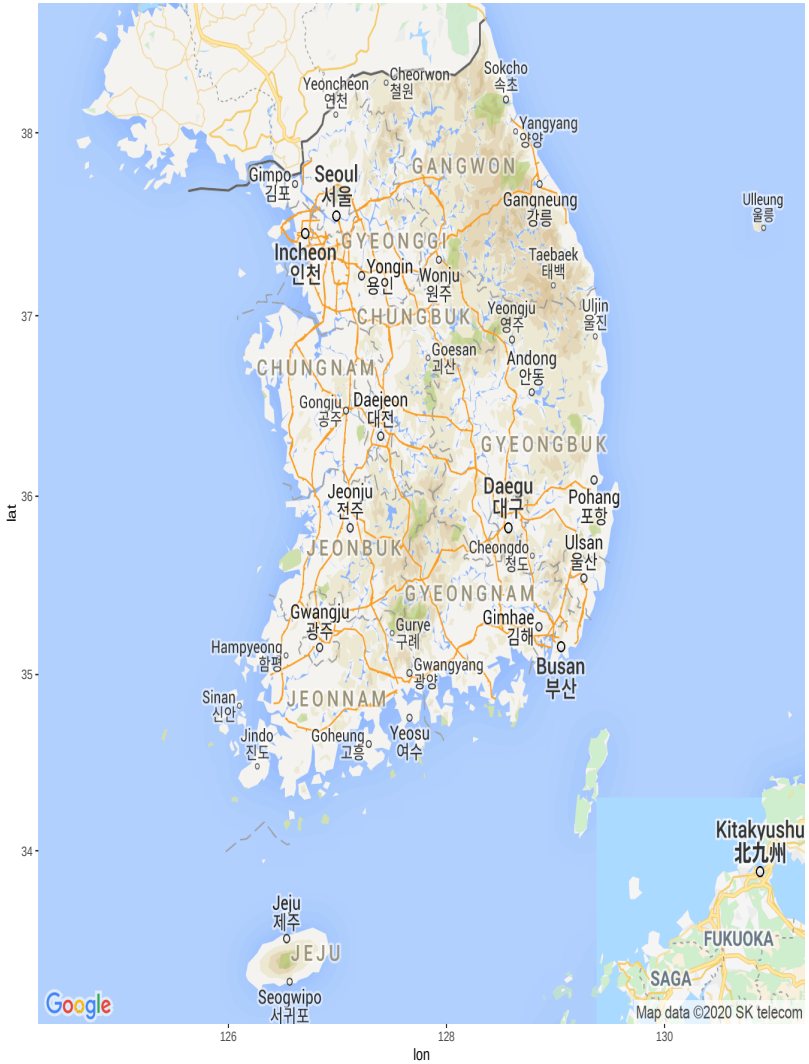
```
library(ggmap)
register_google(key="AIzaSyBGJtBYWknFZDSXMXjjv
```

```
"hide my google api_key"
"hide my google api_key"
"hide my google api_key"
```

```
kr_map <- get_map("south korea", zoom=7, maptype="roadmap")
seoul_map <- get_map("seoul", zoom=11, maptype="roadmap")
```

Our canvases (`kr_map` and `seoul_map`) is ready!

`ggmap(kr_map)`



`ggmap(seoul_map)`

II. Dataset Review

1. 위치 정보 (kr_latlon.csv)

```
kr_latlon <- read_csv("data/kr_latlon.csv", locale = locale('ko', encoding='euc-kr'))
```

- 상단에는 시도별 정보

```
head(kr_latlon)
```

```
## # A tibble: 6 x 4
##   area              lat   lon note
##   <chr>           <dbl> <dbl> <chr>
## 1 서울특별시       37.6  127. <NA>
## 2 제주특별자치도   33.5  127. <NA>
## 3 전라남도         34.8  128. 여수
## 4 전라북도         35.8  127. 전주
## 5 광주광역시       35.2  127. <NA>
## 6 경상남도         35.2  128. 진주
```

- 하단에는 서울의 구별 정보

```
tail(kr_latlon)
```

```
## # A tibble: 6 x 4
##   area       lat   lon note
##   <chr>    <dbl> <dbl> <chr>
## 1 서대문구  37.6  127. <NA>
## 2 양천구    37.5  127. <NA>
## 3 영등포구  37.5  127. <NA>
## 4 관악구    37.5  127. <NA>
## 5 성동구    37.6  127. <NA>
## 6 용산구    37.5  127. <NA>
```

2. 지역 정보 (kr_pop_tidy.csv, seoul_pop_tidy.csv)

- 전국 (kr_pop_tidy.csv)

```
kr_pop <- read_csv(
  "data/kr_pop_tidy.csv",
  locale = locale('ko', encoding='euc-kr'))
head(kr_pop)
```

```
## # A tibble: 6 x 5
##      X1 state      year category       pop
##   <dbl> <chr>     <dbl> <chr>          <dbl>
## 1     1 전국       2015 총인구 (명) 51069375
## 2     2 서울특별시 2015 총인구 (명)  9904312
## 3     3 부산광역시 2015 총인구 (명)  3448737
## 4     4 대구광역시 2015 총인구 (명)  2466052
## 5     5 인천광역시 2015 총인구 (명)  2890451
## 6     6 광주광역시 2015 총인구 (명)  1502881
```

- 서울 (kr_pop_tidy.csv)

```
seoul_pop <- read_csv(
  "data/seoul_pop_tidy.csv",
  locale = locale('ko', encoding='euc-kr'))
seoul_pop$district <-
  str_trim(seoul_pop$district)
head(seoul_pop)
```

```
## # A tibble: 6 x 5
##      X1 district  year category       pop
##   <dbl> <chr>    <dbl> <chr>          <dbl>
## 1     1 종로구   2015 총인구 (명) 161521
## 2     2 중구     2015 총인구 (명) 128478
## 3     3 용산구   2015 총인구 (명) 227282
## 4     4 성동구   2015 총인구 (명) 295006
## 5     5 광진구   2015 총인구 (명) 368199
## 6     6 동대문구 2015 총인구 (명) 364787
```

3. 지역+위치 정보

```
kr_pop_2018 <- left_join(kr_pop, kr_latlon, by=c("state"="area")) %>%
  filter(year==2018 & category=="총인구 (명)" & !is.na(lat)) %>%
  select(state, pop, lat, lon)
seoul_pop_2018 <- left_join(seoul_pop, kr_latlon, by=c("district"="area")) %>%
  filter(year==2018 & category=="총인구 (명)" & !is.na(lat)) %>%
  select(district, pop, lat, lon)
```

```
head(kr_pop_2018)
```

```
## # A tibble: 6 x 4
##   state          pop   lat   lon
##   <chr>        <dbl> <dbl> <dbl>
## 1 서울특별시 9673936  37.6  127.
## 2 부산광역시 3395278  35.2  129.
## 3 대구광역시 2444412  35.9  129.
## 4 인천광역시 2936117  37.5  127.
## 5 광주광역시 1490092  35.2  127.
## 6 대전광역시 1511214  36.4  127.
```

```
head(seoul_pop_2018)
```

```
## # A tibble: 6 x 4
##   district    pop   lat   lon
##   <chr>     <dbl> <dbl> <dbl>
## 1 종로구   157967  37.6  127.
## 2 중구     129797  37.6  127.
## 3 용산구   226938  37.5  127.
## 4 성동구   306796  37.6  127.
## 5 광진구   362304  37.5  127.
## 6 동대문구 358141  37.6  127.
```

III. Grammar of Graphics for MAP - ggmap

Going from ggplot to ggmap

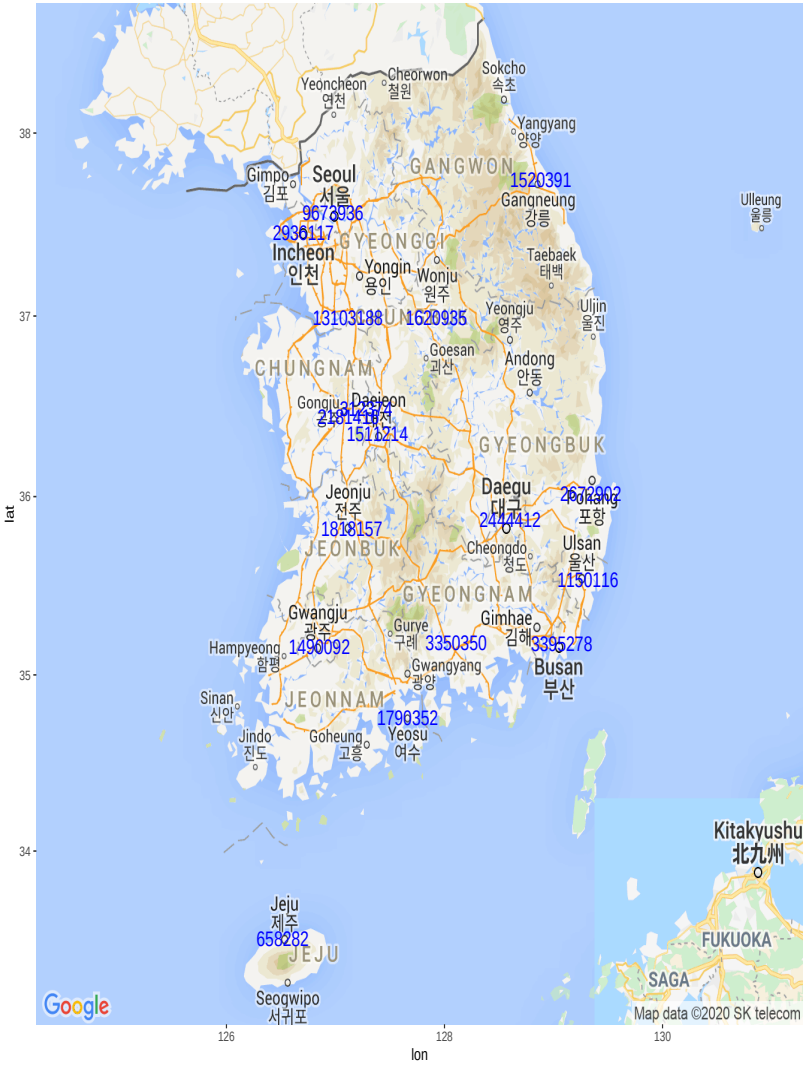
Principle

1. Replace `ggplot()` by `ggmap(<canvas>)`
   - `<canvas>`는 `get_map()`으로 확보한 `ggmap` 객체
2. Each `geom_` needs `<data>`
   - `<data>`는 지역정보(지역의 특성에 대한 수치 정보)와 위치 정보(각 지역의 위경도 정보)를 `join`한 `data.frame` 객체
3. Assign `lon` and `lat` in `<data>` to `x` and `y` aesthetics, respectively.

Minimal examples

```
kr_pop_2018_map <- ggmap(kr_map) + # Principle 1
  geom_text(data = kr_pop_2018, # Principle 2
            aes(x = lon, y = lat, label=pop), # Principle 3
            size = 5, color = "blue")
seoul_pop_2018_map <- ggmap(seoul_map) + # Principle 1
  geom_text(data = seoul_pop_2018, # Principle 2
            aes(x = lon, y = lat, label=pop), # Principle 3
            size = 5, color = "blue")
```

kr_pop_2018_map

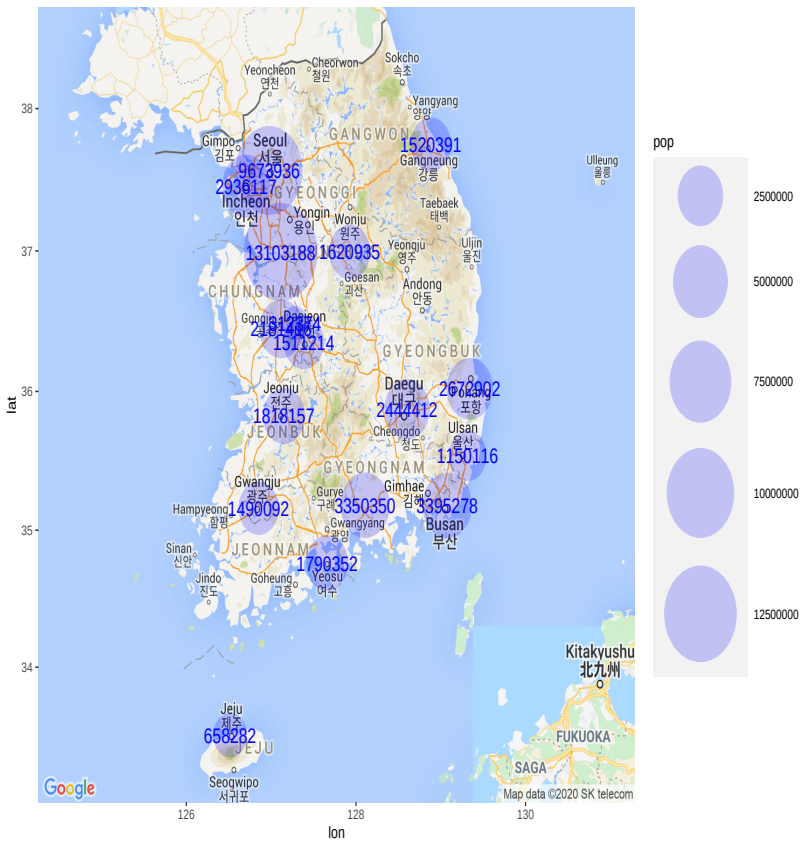


seoul_pop_2018_map

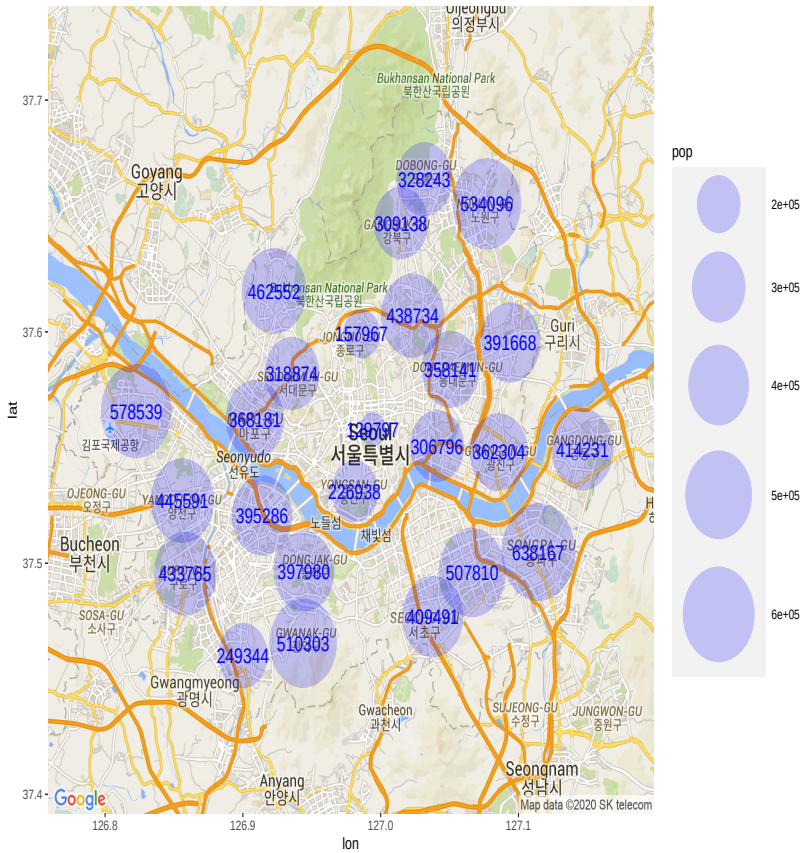
Some improvements over the minimal examples

```
kr_pop_2018_map <- ggmap(kr_map) + # Principle 1
  geom_text(data = kr_pop_2018, # Principle 2
            aes(x = lon, y = lat, label=pop), # Principle 3
            size = 5, color = "blue") +
  geom_point(data = kr_pop_2018,
             aes(x = lon, y = lat, size=pop),
             color = "blue", alpha = 0.2) +
  scale_size(range = c(10, 30))
seoul_pop_2018_map <- ggmap(seoul_map) + # Principle 1
  geom_text(data = seoul_pop_2018, # Principle 2
             aes(x = lon, y = lat, label=pop), # Principle 3
             size = 5, color = "blue") +
  geom_point(data = seoul_pop_2018, # Principle 2
             aes(x = lon, y = lat, size=pop), # Principle 3
             color = "blue", alpha = 0.2) +
  scale_size(range = c(10, 30))
```

```
kr_pop_2018_map
```



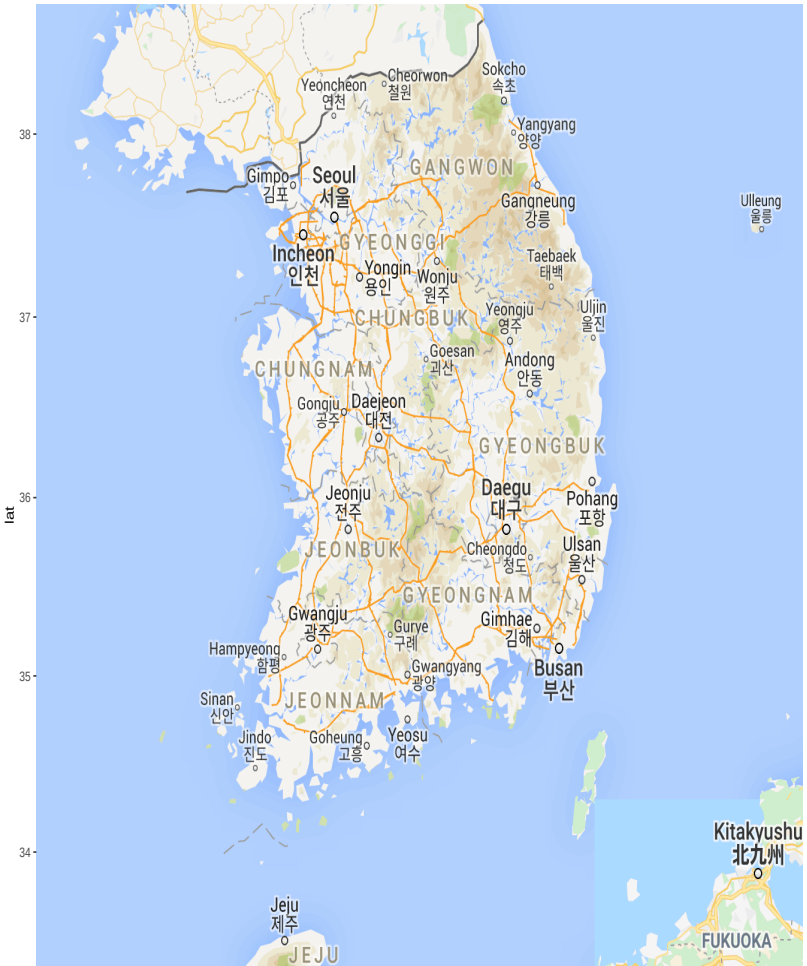

```
seoul_pop_2018_map
```



IV. Some other canvases

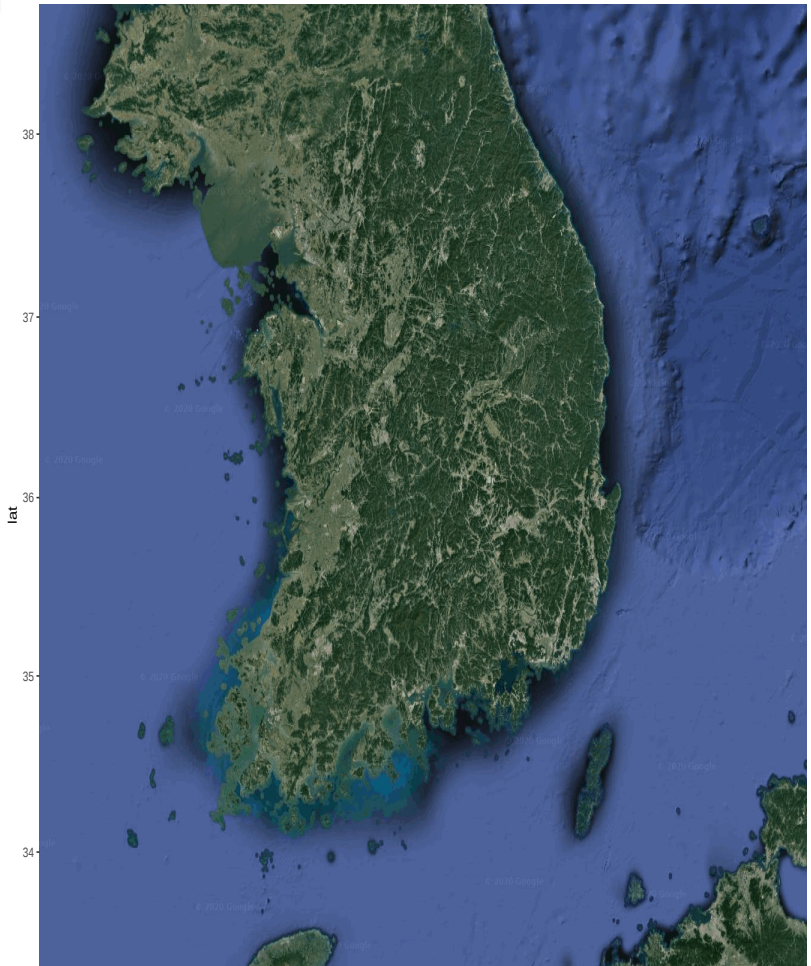
- google - roadmap

```
canvas1 <- get_map(
  "south korea", zoom=7, maptype="roadmap")
ggmap(canvas1)
```



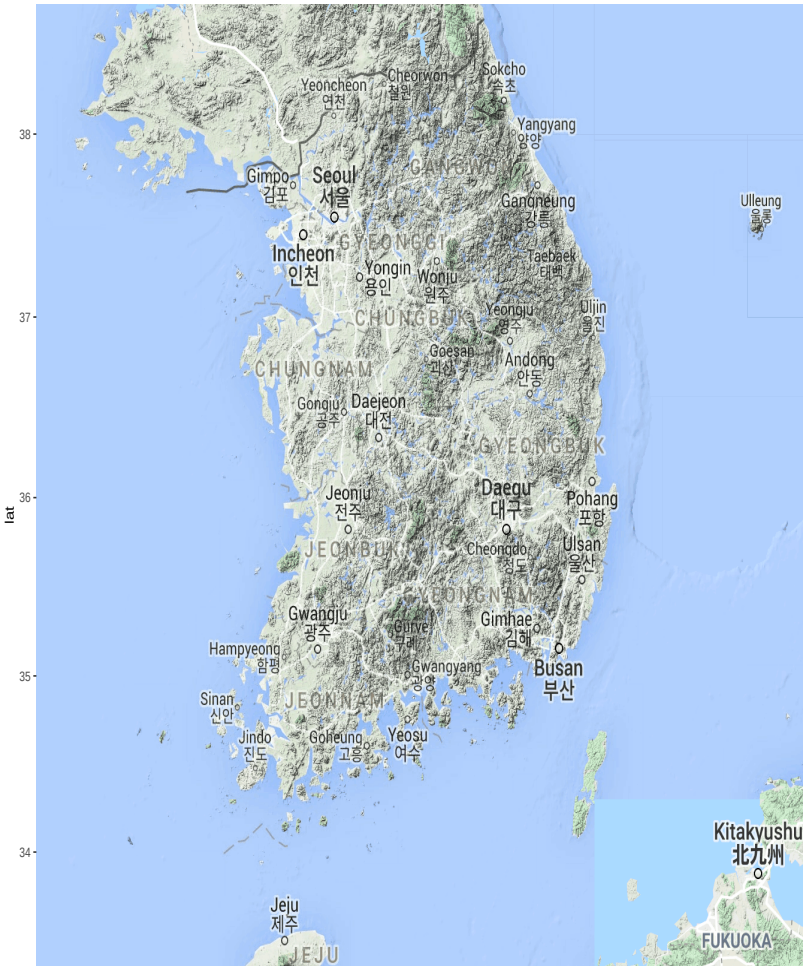
- google - satellite

```
canvas2 <- get_map(
  "south korea", zoom=7, maptype="satellite")
ggmap(canvas2)
```



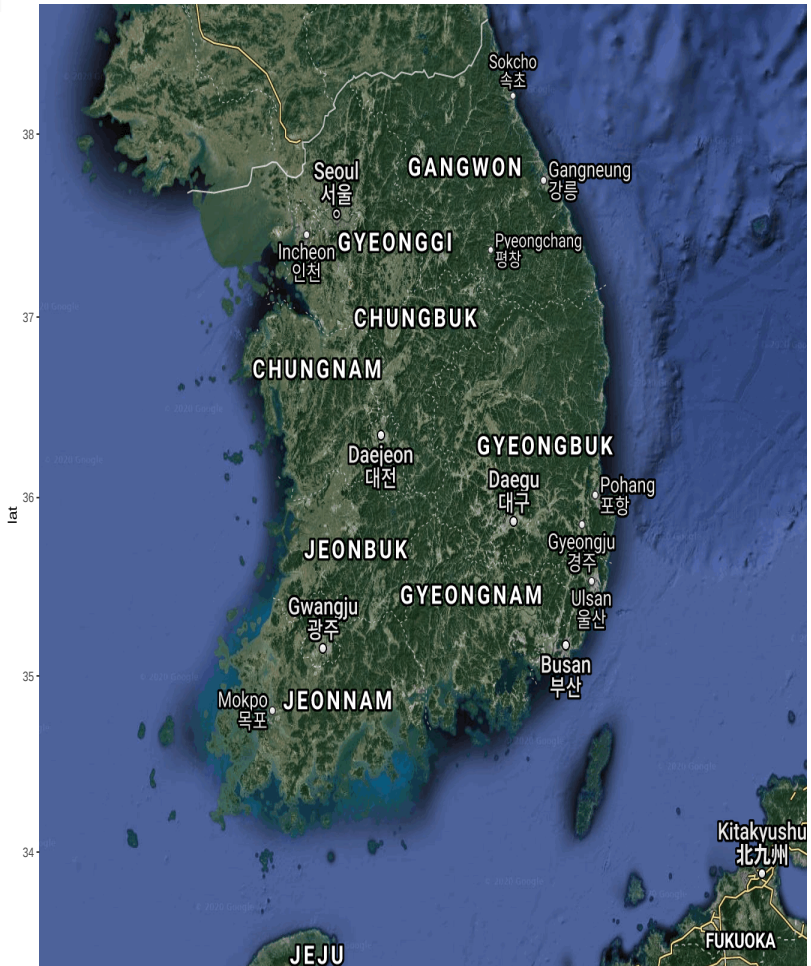
- google - terrain

```
canvas3 <- get_map(
  "south korea", zoom=7, maptype="terrain")
ggmap(canvas3)
```



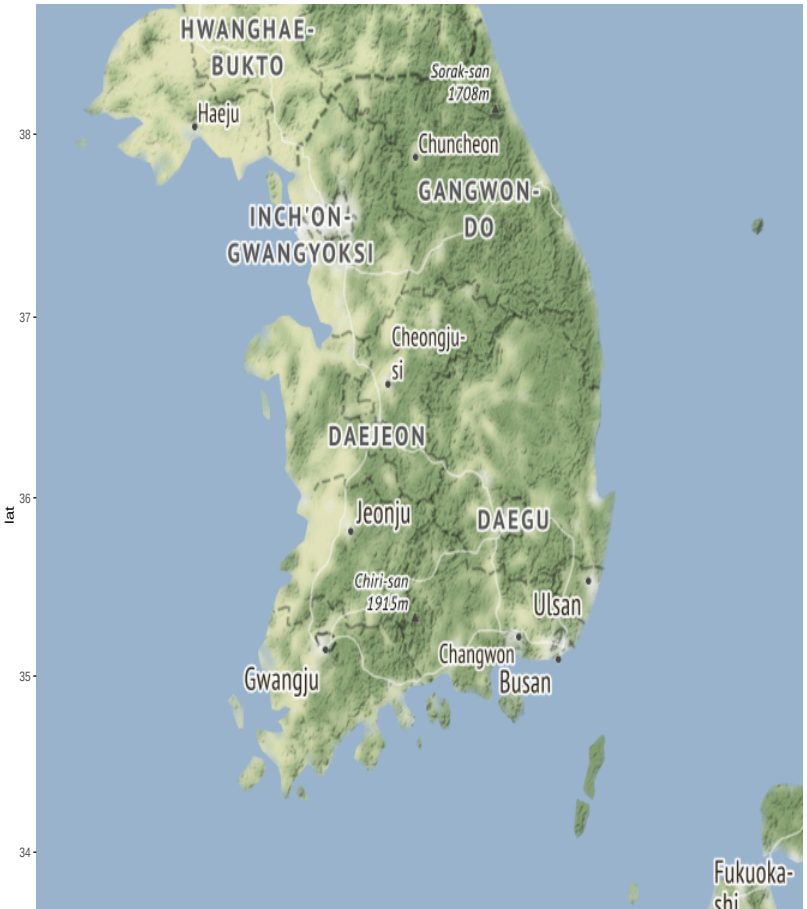
- google - hybrid

```
canvas4 <- get_map(
  "south korea", zoom=7, maptype="hybrid")
ggmap(canvas4)
```



- Stamen - terrain

```
canvas5 <- get_map(
  "south korea", zoom=7,
  source="stamen", maptype="terrain")
ggmap(canvas5)
```



- Stamen - watercolor

```
canvas6 <- get_map(
  "south korea", zoom=7,
  source="stamen", maptype="watercolor")
ggmap(canvas6)
```

- Stamen - toner

```
canvas7 <- get_map(
  "south korea", zoom=7,
  source="stamen", maptype="toner")
ggmap(canvas7)
```

```
"Hello"
```

```
## [1] "Hello"
```