

9주차

지도상에 데이터 표현하기

오 세 종

Contents

1. 특정지역 지도보기
2. 마커, 텍스트 출력
3. 데이터의 크기 지도에 출력

개요

- 구글맵 API 기능을 이용하여 구글지도상에 정보를 표시할 수 있다.



출처

<https://vijaybarve.wordpress.com/tag/ggplot2/>

- 설치가 필요한 패키지
 - ggmap : 구글맵과 연동을 위해 필요
 - ggplot2 : 구글맵 위에 그래프 출력을 위해 필요

1. 특정 지역 지도 보기

```
library(ggmap)
gc <- geocode(enc2utf8("용인")) # 지점의 경도위도
cen <- as.numeric(gc)          # 경도위도를 숫자로
map <- get_googlemap(center=cen) # 지도생성
ggmap(map)                     # 지도 화면에 보이기
```

geocode()

: 지역명을 경도와 위도로 변환

as.numeric(gc)

: 경도와 위도를 숫자로 변환

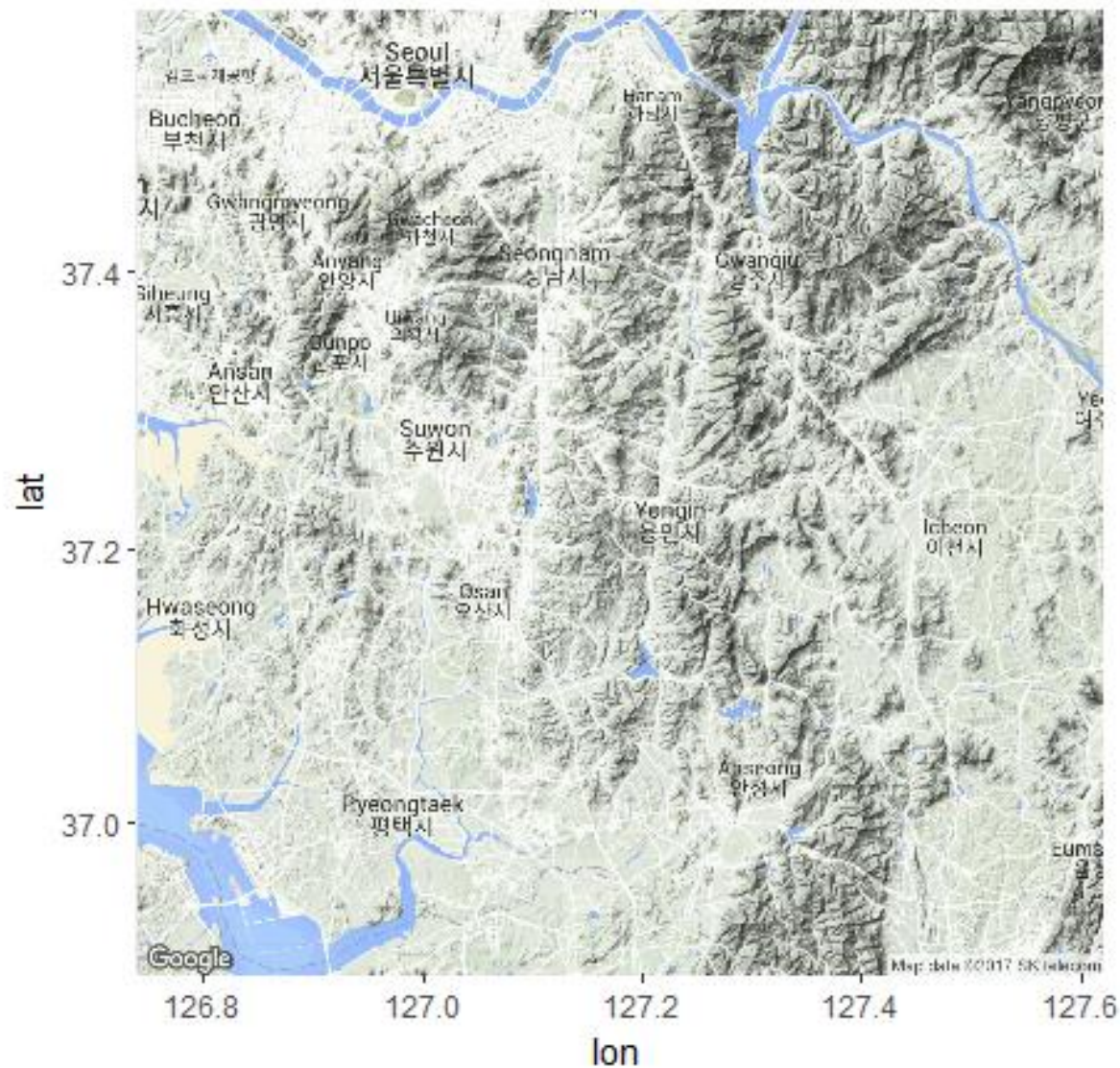
get_googlemap()

: 지정된 지역의 구글 지도를 가져온다

center=cen : 지도의 중심점 지정

```
> gc
      lon      lat
1 127.1776 37.24109
> cen
[1] 127.17755 37.24109
      경도      위도
```

1. 특정 지역 지도 보기



1. 특정 지역 지도 보기

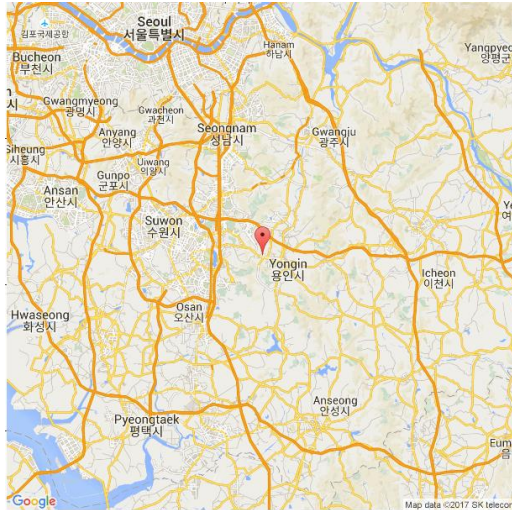
- `get_googlemap`

Parameter	설명
center	지도의 중심좌표
zoom	지도의 확대크기로서 3(대륙)~21(빌딩). 기본값은 10 (도시)
size	지도의 가로와 세로 픽셀 크기. 기본값은 640 x 640 (c(640,640))
maptype	출력될 지도유형 (다음 슬라이드 참조) 기본값은 " terrain "

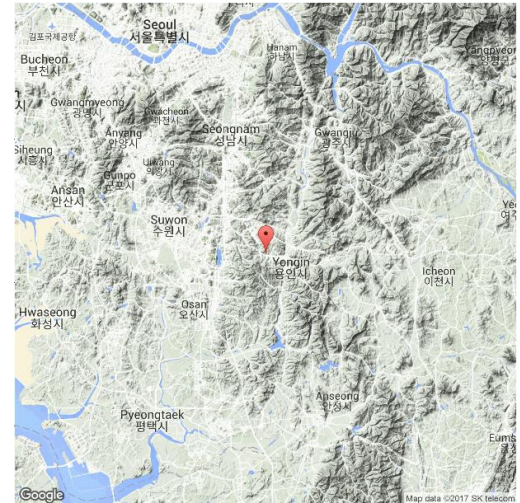
1. 특정 지역 지도 보기

- maptype

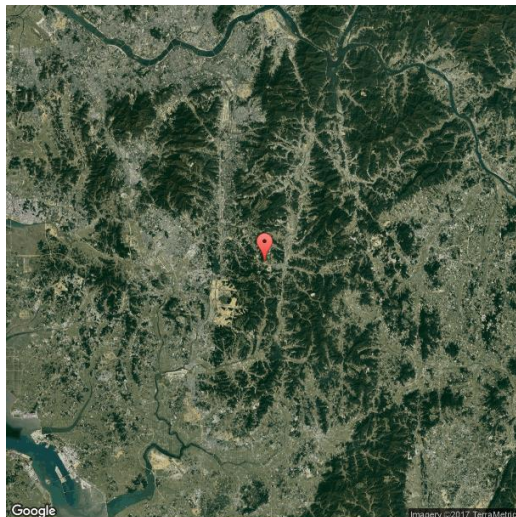
"roadmap"



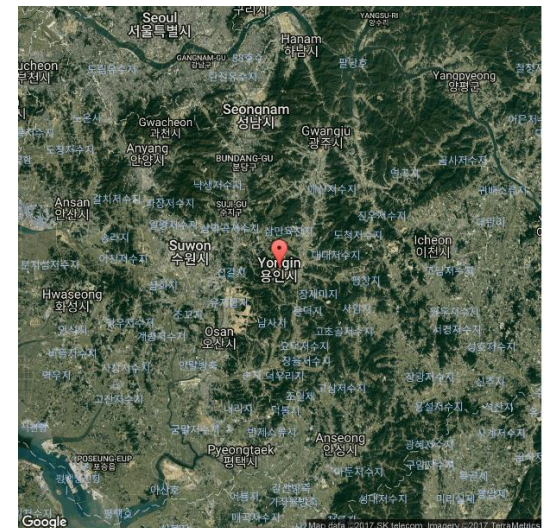
"terrain"



"satellite"

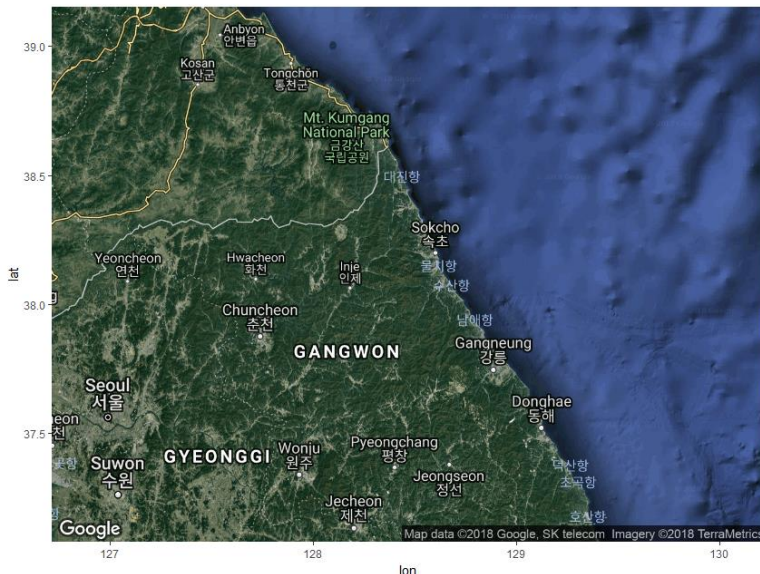


"hybrid"



1. 특정 지역 지도 보기

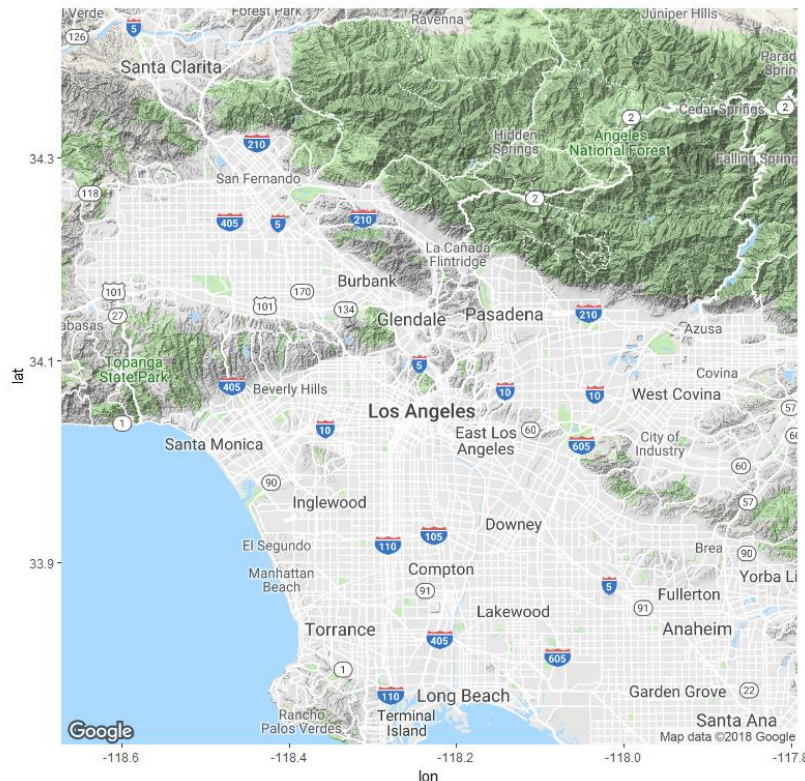
```
library(ggmap)
gc <- geocode(enc2utf8("설악산")) # 지점의 경도위도
cen <- as.numeric(gc)           # 경도위도를 숫자로
map <- get_googlemap(center=cen,
                      zoom=8,
                      size=c(640,480),
                      maptype="hybrid")
ggmap(map)                      # 지도 화면에 보이기
```



1. 특정 지역 지도 보기

- 경도와 위도 값을 알 때
 - 경도 -118.233248, 위도 34.085015 (LA)

```
cen <- c(-118.233248, 34.085015)
map <- get_googlemap(center=cen) # 지도생성
ggmap(map) # 지도 화면에 보이기
```



1. 특정 지역 지도 보기

- 구글지도를 이용하여 특정 지역의 경도, 위도 알아내기
 - <https://maps.google.com> (원하는 지점 클릭)




위도 경도

[연습문제 1]

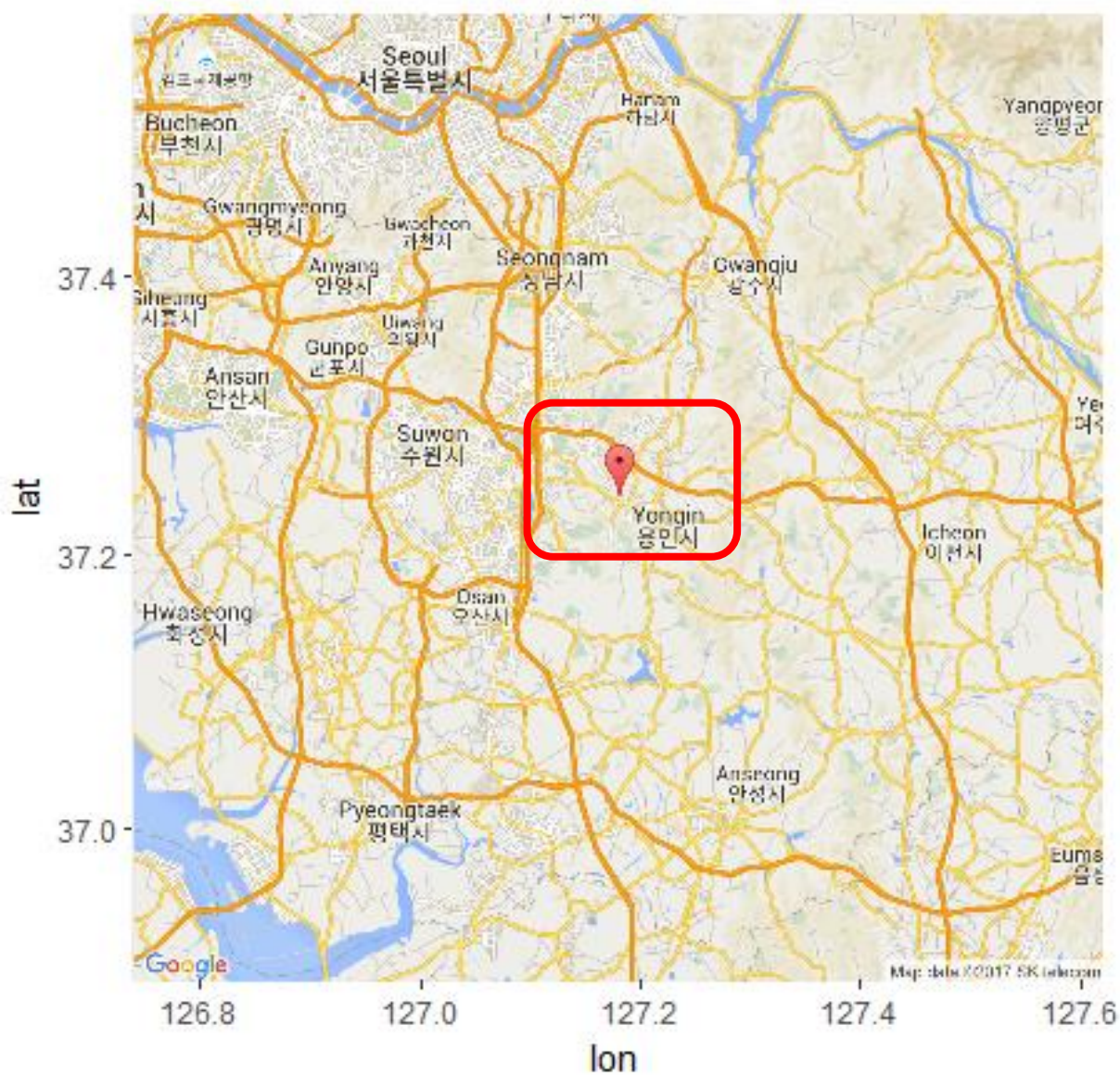
1. 서울시청 지역의 지도를 표시하되 지도 크기는 800x600 으로 하고 maptype 은 "**roadmap**" 으로 하시오
2. 금강산 지역의 지도를 표시하되 지도 크기는 640x480 으로 하고 maptype 은 "**hybrid**" 으로 하시오. zoom 은 8로 하시오
3. 경도 103.867881, 위도 1.331017 지역의 지도를 표시하되 maptype 은 "**roadmap**" 으로 하고, zoom 은 9로 하시오

2. 마커, 텍스트 출력

- 지도의 위도,경도 위치에 마커()를 출력한다
 - 용인에 마커를 표시해보기

```
library(ggmap)
gc <- geocode(enc2utf8("용인")) # 지점의 경도위도
cen <- as.numeric(gc)          # 경도위도를 숫자로
map <- get_googlemap(center=cen, # 지도의 중심
                      maptype="roadmap", # 지도의 형태
                      marker=gc)        # 마커의 위치
ggmap(map)                       # 지도 화면에 보이기
```

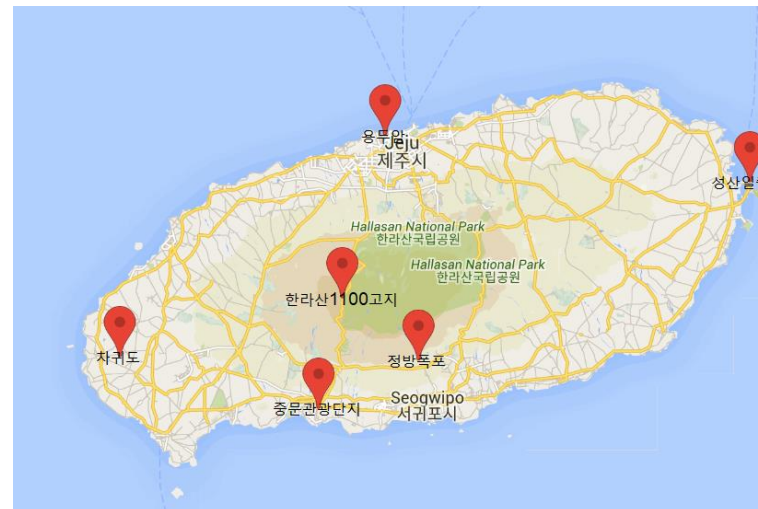
2. 마커, 텍스트 출력



2. 마커, 텍스트 출력

- 여러 지역의 마커 표시하기 : 제주도 관광지

지명	주소
용두암	제주시 용두암길 15
성산일출봉	서귀포시 성산읍 성산리
정방폭포	서귀포시 동홍동 299-3
중문관광단지	서귀포시 중문동 2624-1
한라산110고지	서귀포시 색달동 산1-2
차귀도	제주시 한경면 고산리 125



2. 마커, 텍스트 출력

```
library(ggmap)
library(ggplot2)

names <- c("용두암", "성산일출봉", "정방폭포",
           "중문관광단지", "한라산1100고지", "차귀도")
addr <- c("제주시 용두암길 15",
          "서귀포시 성산읍 성산리",
          "서귀포시 동홍동 299-3",
          "서귀포시 중문동 2624-1",
          "서귀포시 색달동 산1-2",
          "제주시 한경면 고산리 125")
gc <- geocode(enc2utf8(addr)) #주소를 경도, 위도로 변환
df <- data.frame(name=names,
                  lon=gc$lon,
                  lat=gc$lat)
```

2. 마커, 텍스트 출력

```
> df
```

	name	lon	lat
1	용두암	126.5117	33.51496
2	성산일출봉	126.9324	33.46993
3	정방폭포	126.5504	33.29809
4	중문관광단지	126.4348	33.25143
5	한라산1100고지	126.4625	33.35778
6	차귀도	126.2057	33.30110

2. 마커, 텍스트 출력

(continue)

```
cen <- c(mean(df$lon), mean(df$lat))  
map <- get_googlemap(center=cen,  
  maptype="roadmap",  
  zoom=10,  
  size=c(640, 480),  
  marker=gc)
```

```
ggmap(map) # 지도+마커 화면에 보이기
```

명소이름 지도위에 표시하기

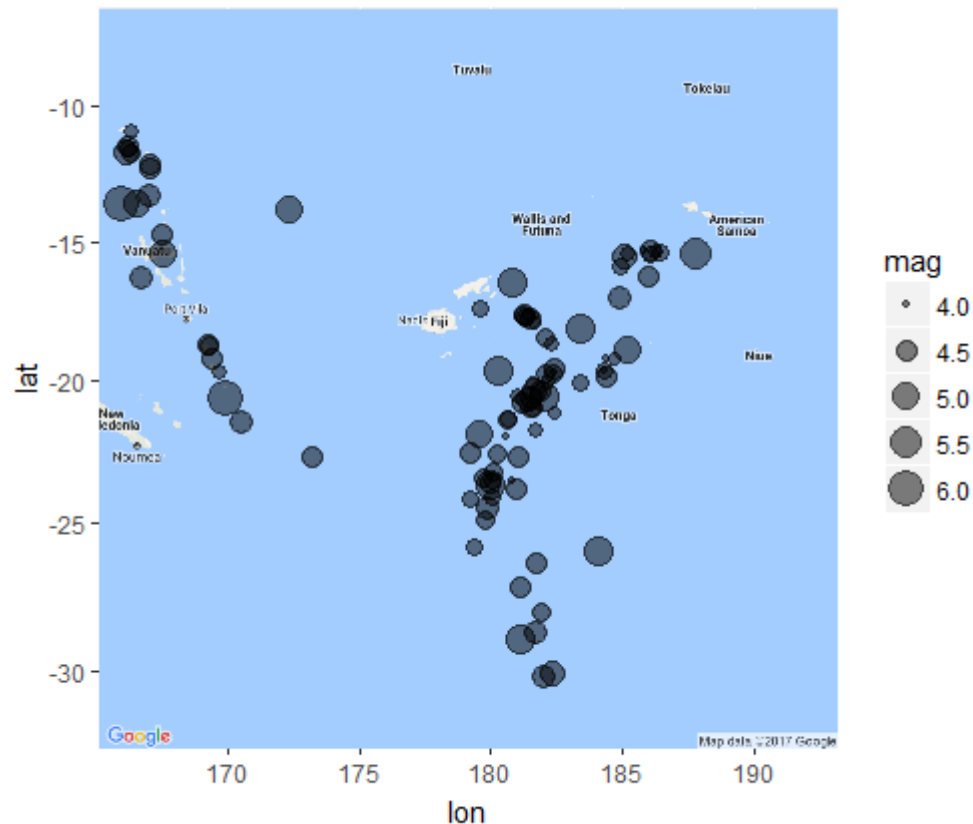
```
gmap <- ggmap(map)  
gmap+geom_text(data=df,  
  aes(x=lon, y=lat), #글씨 위치  
  size=5, #글씨 크기  
  label=df$name) #글씨 내용
```


[연습문제 2]

1. 서울시 각 구청의 위치에 마커, 구청이름을 지도위에 표시하시오
2. 우리나라의 광역시를 마커, 광역시 이름과 함께 지도위에 표시하시오

3. 데이터의 크기를 지도에 표현하기

- R 에서 제공하는 지진 발생 데이터(quakes)를 이용하여 지진 규모를 발생지역에 표시해보자



3. 데이터의 크기를 지도에 표현하기

```
> head(quakes)
```

	lat	long	depth	mag	stations
1	-20.42	181.62	562	4.8	41
2	-20.62	181.03	650	4.2	15
3	-26.00	184.10	42	5.4	43
4	-17.97	181.66	626	4.1	19
5	-20.42	181.96	649	4.0	11
6	-19.68	184.31	195	4.0	12

위도

경도

지진규모(진도)

진앙지 깊이

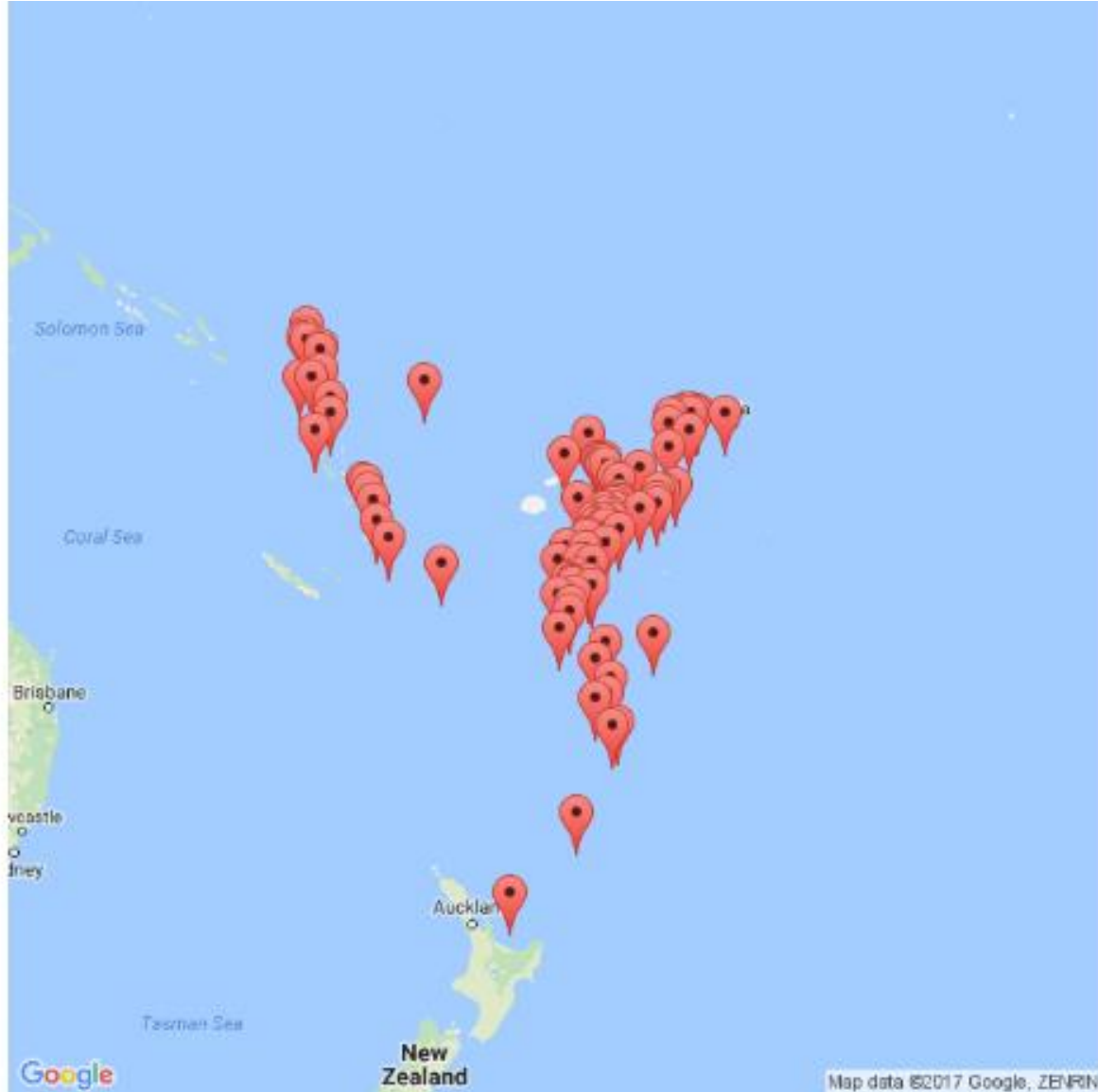
3. 데이터의 크기를 지도에 표현하기

```
library(ggmap)
library(ggplot2)

df <- head(quakes, 100)
cen <- c(mean(df$long), mean(df$lat))
gc <- data.frame(lon=df$long, lat=df$lat)
gc$lon <- ifelse(gc$lon > 180, -(360 - gc$lon), gc$lon)
gc
map <- get_googlemap(center=cen,
  maptype="roadmap",
  zoom=4,
  marker=gc)
ggmap(map) + theme(axis.title.x=element_blank(),
  axis.text.x=element_blank(),
  axis.ticks.x=element_blank(),
  axis.title.y=element_blank(),
  axis.text.y=element_blank(),
  axis.ticks.y=element_blank())
```

④ 경도가 180도를 넘는 경우,
0~-180도 사이로 변환
⇨ ifelse(조건, 조건이 참일 경우
의 값, 조건이 거짓일 경우의 값)

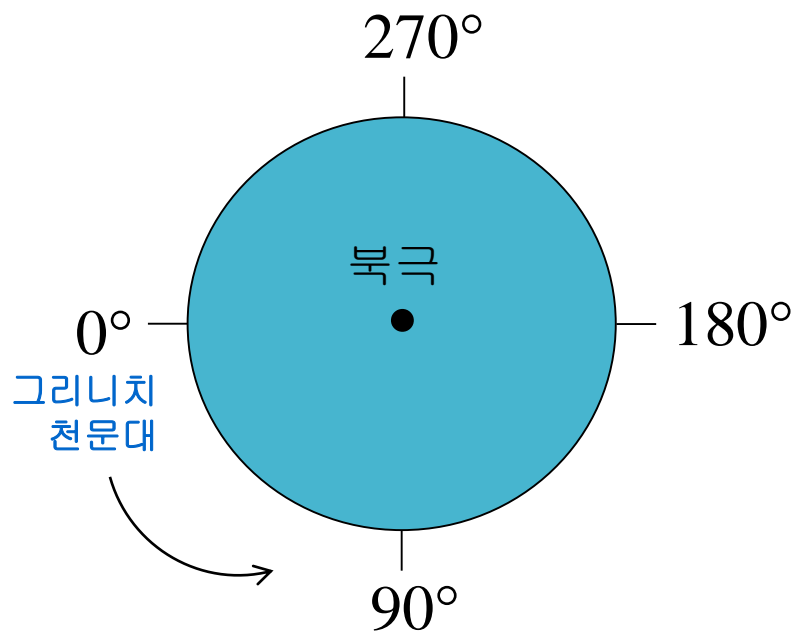
3. 데이터의 크기를 지도에 표현하기



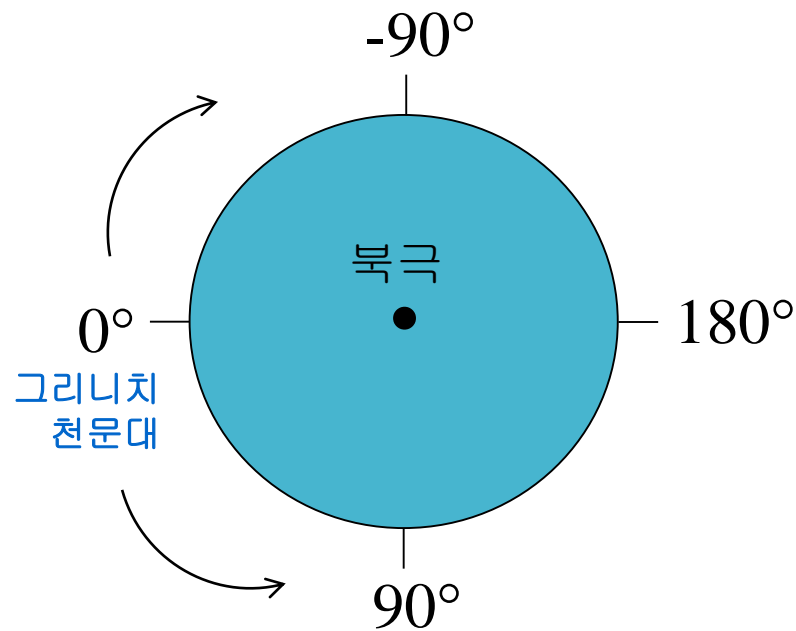
3. 데이터의 크기를 지도에 표현하기

- [note]

360° 경도



-180° ~ 180° 경도 (구글지도)



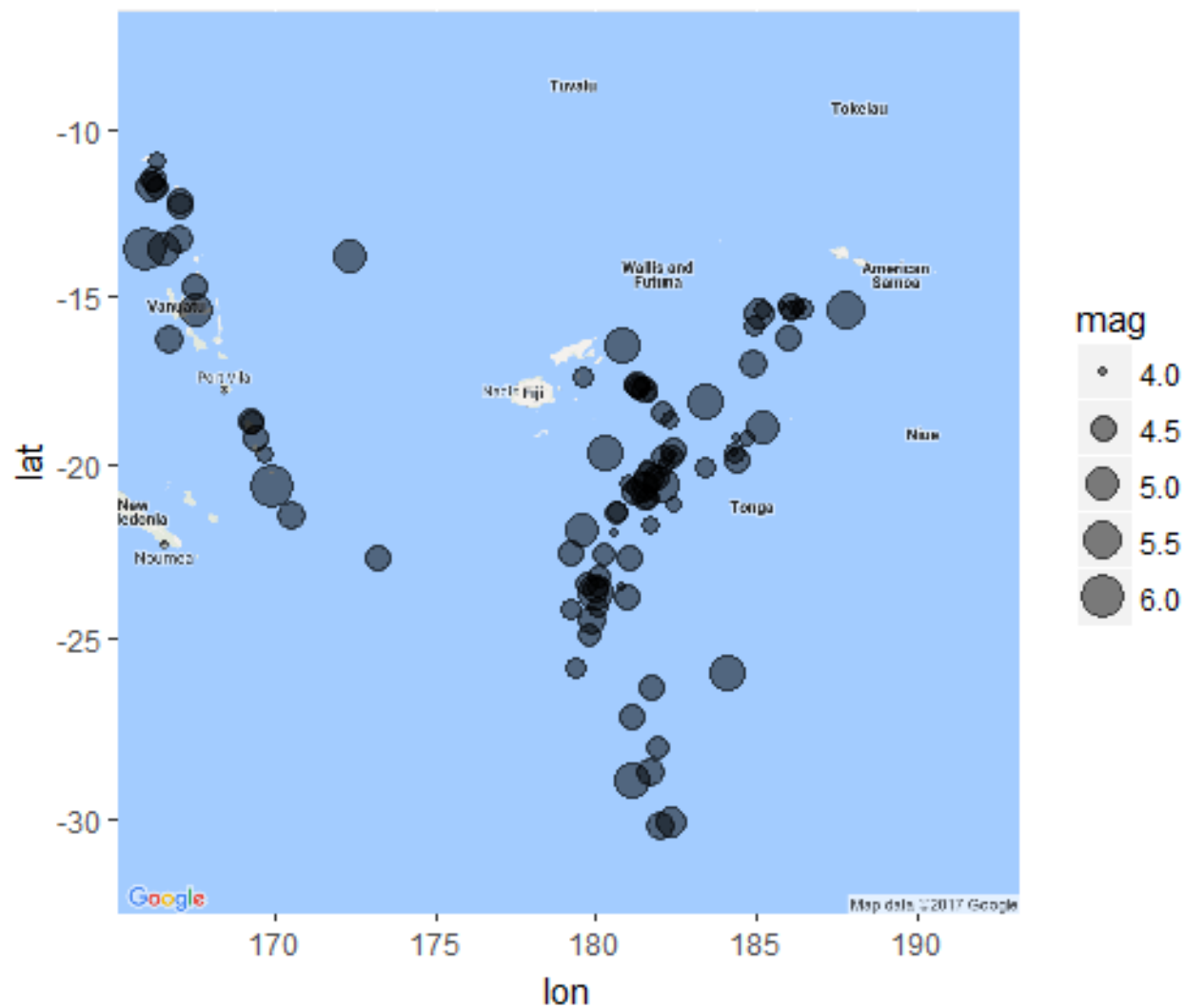
3. 데이터의 크기를 지도에 표현하기

(continue)

```
map <- get_googlemap(center=cen,  
  maptype="roadmap",  
  zoom=5)  
gmap <- ggmap(map)  
gmap+geom_point(data=df,  
  aes(x=long,y=lat,size=mag),  
  alpha=0.5)
```

지도 위에 df 데이터 내의 경도(long)와 위도(lat)를 x, y 좌표로 하는 위치에 원의 크기(mag)로 표시. 투명도(alpha)는 완전 투명(0)과 완전 불투명(1)의 중간 (0.5)

3. 데이터의 크기를 지도에 표현하기



[연습문제 3]

1. treemap 패키지의 GNI2014 데이터셋을 이용하여 유럽대륙의 각 나라의 위치에 그 나라의 인구수를 원의 크기로 표시하십시오. (원의 색깔은 초록색, alpha=0.5)

```
> library(treemap)
warning message:
패키지 'treemap'는 R 버전 3.4.3에서 작성되었습니다
> data("GNI2014")
> head(GNI2014)
```

	iso3	country	continent	population	GNI
3	BMU	Bermuda	North America	67837	106140
4	NOR	Norway	Europe	4676305	103630
5	QAT	Qatar	Asia	833285	92200
6	CHE	Switzerland	Europe	7604467	88120
7	MAC	Macao SAR, China	Asia	559846	76270
8	LUX	Luxembourg	Europe	491775	75990

[Note] 구글 API key 를 이용한 주소변환

- ggmap에서 제공하는 geocode() 함수는 구글의 주소변환 서비스를 이용하는데 하루 2500건으로 제한되어 있음
- 그 이상의 작업을 필요로 하는 경우는 google API 를 이용해야 함

- [구글 API 키 정의]

1. 구글 로그인 후 아래에서 새로운 프로젝트 생성

<https://console.developers.google.com/apis>



[Note] 구글 API key 를 이용한 주소변환

2. 라이브러리에서 Google Maps Geocoding API 사용 신청

검색결과 19개



Google Places API for iOS
Google

Make your iOS app stand out with detailed information about 100 million places



Google Places API for Android
Google

Make your Android app stand out with detailed information about 100 million places



Google Maps Geocoding API
Google

Convert between addresses and geographic coordinates.



Google Static Maps API
Google

Simple, embeddable map image with minimal code.



API API 및 서비스

대시보드

라이브러리

사용자 인증 정보

대시보드

API 및 서비스 사용 설정

사용 설정된 API 및 서비스

일부 API 및 서비스는 자동으로 사용 설정됩니다.

지난 1시간 동안의 활동

1시간 6시간 12시간 1일 2일 4일 7일 14일 30일

트래픽

요청/초

이 기간에는 트래픽이 없습니다.

오류

요청 비율

이 기간에는 오류가 없습니다.

지연 시간 중앙값

밀리초

지연 시간 데이터가 없습니다.

API

요청 수

오류

오류 비율

지연 시간, 중앙값

지연 시간, 98%

Google Maps Geocoding API

-

-

-

-

-

사용 중지



클릭

[Note]

3. 발급받은 API KEY 를 복사하여 둔다

API API 및 서비스

대시보드

라이브러리

사용자 인증 정보

사용자 인증 정보

OAuth 동의 화면

도메인 확인

사용자 인증 정보 만들기

삭제

이 API에 액세스하려면 이러한 사용자 인증 정보 중 하나를 사용하거나 위에서 사용자 인증 정보를 새로 만드세요. 자세한 내용은 [API 문서를 참조](#)하세요.

API 키

<input type="checkbox"/>	이름	생성일	제한사항	키
<input type="checkbox"/>	⚠ API 키 1	2017. 11. 25.	없음	AlzaSyDVGG[REDACTED]-fsY8U2e1fuAs

복사

4. googleway 패키지를 설치한다

[Note]

5. googleway 를 통해 주소를 좌표값으로 변환한다.

```
library(googleway)
mykey <- "발급받은 API key"
gc <- NULL
# 벡터 addr에 변환하고 싶은 주소들 저장
for (i in 1:length(addr)) {
  df <- google_geocode(addr[i], key = mykey)
  gc = rbind(gc, df$results$geometry$location)
}
gc # 변환된 좌표값이 저장되어 있음
```