구글 지도를 활용하는 ggmap 패키지

구글 서비스를 활용할 수 있는 패키지인 ggmap에 대해 알아보겠습니다. 먼저 ggmap 패키지를 설치 및 로드합니다.

2018년 7월 16일에 출시된 Google Map Platform에 적용된 ggmap은 Google API Key를 입력하는 register_google() 함수가 필요하므로 다음과 같이 두 가지 패키지를 설치하고 로드합 니다.

패키지 설치 및 로드

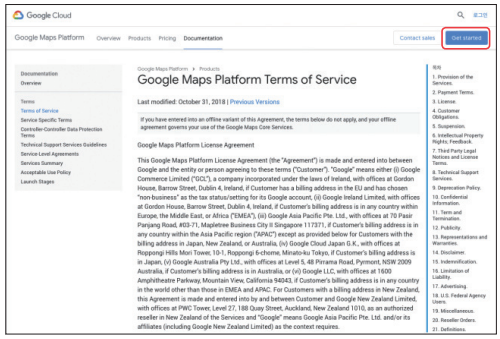
```
install.packages("devtools")      # devtools 패키지 설치
library(devtools)                 # devtools 패키지 로드
install_github("dkahle/ggmap")    # ggmap 패키지 설치
library(ggmap)                    # ggmap 패키지 로드
```

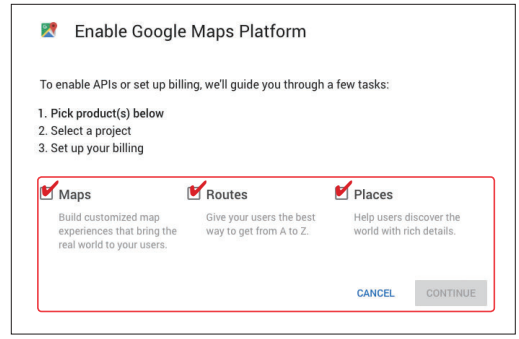
↳ 실행 결과

```
> install.packages("devtools")
> library(devtools)
> install_github("dkahle/ggmap")
> library(ggmap)
Loading required package: ggplot2
Google Maps API Terms of Service: http://developers.google.com/maps/terms.
Please cite ggmap if you use it: see citation("ggmap") for details.
```

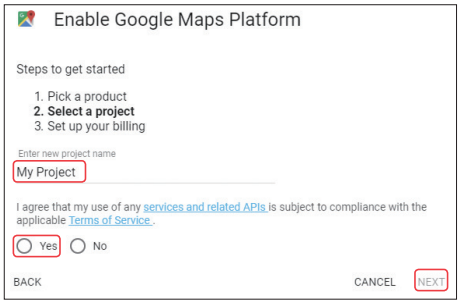
빨간 알림 문구는 API 서비스 이용 안내입니다. 다음 절차를 따라 약관에 동의하고 API Key를 받아봅시다.

01. http://developers.google.com/maps/terms에 접속한 다음 우측 상단의 Get started 버튼을 클릭하고, 사용하고자 하는 API를 체크하면 됩니다. Maps(지도), Routes(길 안내), Places(위치)가 있는데 우리는 Maps만 체크해도 무방하지만, 모두 체크하고 CONTINUE 버튼을 클릭합니다.



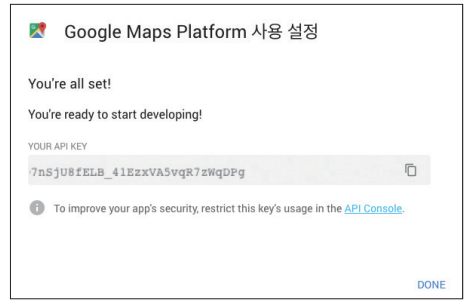


02. 사용하고자하는 프로젝트를 설정하는 화면입니다. Enter new project name에 프로젝트 이름을 입력하고 약관에 동의한다는 의미로 Yes를 클릭하고 다음(Next)으로 넘어갑시다.새로 생성을 하거나 기존에 있는 프로젝트에 연동도 가능합니다. 그리고 결제 연동을 하면, API Key가 발급됩니다.



TIP 결제 연동 과정은 적지 않았습니다. 상단 오른쪽 그림처럼 12개월 간 300달러의 무료 크레딧을 사용할 수 있으며, 평가판 종료 후 자동 청구되지 않는다고 구글은 안내하고 있습니다.

03. 자, 다음과 같이 보인다면 이제 ggmap 패키지를 사용하기 위한 준비는 모두 끝났습니다.



이제부터 ggmap 패키지에서 register_google() 함수, get_googlemap() 함수, ggmap() 함수를 주로 사용합니다. Register_google() 함수는 Google API Key를 등록하고, get_googlemap()는 설정한 위치를 가져오고, ggmap()은 실제 지도를 그려 줍니다.

우선 서울시 지도를 가져와서 그려보도록 하겠습니다.

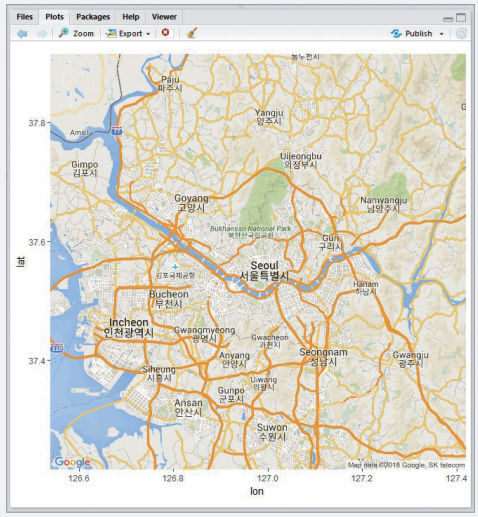
구글 지도 가져오기

```
googleAPIkey = "7nSjU8fELB_41EzxVA5vqR7zWqDPg" #Google API Key 저장
# seoul의 위치 정보를 가져온 후 gg_seoul 변수에 할당
register_google(googleAPIkey)                  #Google API Key 인증
gg_seoul <- get_googlemap("seoul", maptype = "terrain")
ggmap(gg_seoul)     # gg_seoul의 위치 값에 따른 구글 지도 호출
```

↳ 실행 결과

```
> # seoul의 위치 정보를 가져온 후 gg_seoul 변수에 할당
> register_google(googleAPIkey)                  #Google API Key 인증
> gg_seoul <- get_googlemap("seoul", maptype = "roadmap")
Source : https://maps.googleapis.com/maps/api/staticmap?center=seoul&zoom=10&size=6
40x640&scale=2&maptype=roadmap&key=AIzaSyBDWRx1UZjT1i4nVReN0blTNkQU_Q4C8Aw
Source : https://maps.googleapis.com/maps/api/geocode/json?address=seoul&key=AIzaSy
BDWRx1UZjT1i4nVReN0blTNkQU_Q4C8Aw
> ggmap(gg_seoul)
```

구글 지도 API를 호출해서 결과 값을 가져오는 과정입니다.



TIP get_googlemap() 함수로 가져올 수 있는 지도 형태(maptype)에는 "terrain", "satellite", "roadmap", "hybrid"가 있습니다. 각각 옵션을 변경해서 실행한 후 비교해 보세요.

Plots 창에 표시된 지도를 보면 위도와 경도가 함께 출력됩니다. 위도와 경도 데이터가 있다면 우리가 앞에서 실습한 ggplot2 패키지를 추가해서 다른 정보도 함께 보여 줄 수 있습니다. 지도에서 대전역의 위치를 찾아 표시해 보겠습니다. geocode() 함수를 이용해 대전역의 위도와 경도 정보를 저장하고, geom_point() 함수를 이용해 점으로 표시합니다.

구글 지도 위에 산점도 그리기

```
library(dplyr)      # dplyr 패키지 로드
library(ggplot2)    # ggplot2 패키지 로드

# 한글 검색을 위해 utf8로 변환한 후 위도와 경도 데이터를 geo_code 변수에 할당
geo_code <- enc2utf8("대전역") %>% geocode()
geo_data <- as.numeric(geo_code)    # 리스트를 숫자로 변환하여 geo_data 변수에 할당

# 대전역의 위치 정보를 가져온 후 구글 지도 호출
get_googlemap(center = geo_data, maptype = "roadmap", zoom = 13) %>% ggmap() +
    # geo_code에 있는 경도(lon)와 위도(lat) 값으로 산점도 그리기
    geom_point(data = geo_code, aes(x = geo_code$lon, y = geo_code$lat))
```

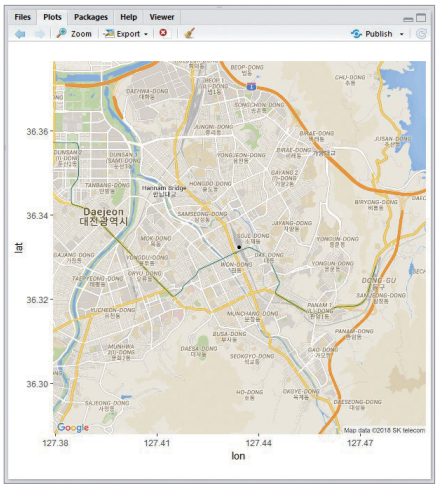
↳ 실행 결과

```
> library(dplyr)
> library(ggplot2)

> geo_code <- enc2utf8("대전역") %>% geocode()
Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=%EB
%8C%80%EC%A0%84%EC%97%AD&sensor=false
> geo_data <- as.numeric(geo_code)

> get_googlemap(center=geo_data, maptype = "roadmap", zoom = 13) %>% ggmap() +
+   geom_point(data = geo_code, aes(x = geo_code$lon, y = geo_code$lat))
Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=36.332344,127.4
342&zoom=13&size=640x640&scale=2&maptype=roadmap&sensor=false
```

%)% 연산자를 사용하기 위해 dplyr 패키지를 로드합니다.



`zoom = 13` 옵션을 추가하여 서울시 지도를 불러왔을 때보다 실행 결과를 확대하여 출력했습니다. 정확한 위치에 표시된 것을 확인할 수 있습니다.

지금까지 R 데이터 시각화를 위한 몇 가지 주요 패키지와 사용 방법을 알아보았습니다. 4부에서는 지금까지 배운 내용을 바탕으로 한 다양한 프로젝트를 실습해 보겠습니다. 실제 데이터를 분석하는 방법과 표현하는 방법을 익힐 수 있을 것입니다.