

Geospatial Analytics in R

Spatial Visualisation with Google Maps and OpenStreetMap

Dean Marchiori

Senior Analyst- Wealth Analytics

Introduction

The R `ggmap` package is a package for spatial visualisation with Google Maps and OpenStreetMap.

It allows for the easy visualisation of spatial data and models on top of Google Maps, OpenStreetMaps, Stamen Maps, or CloudMade Maps using `ggplot2`.

The advantages of using R for geo-spatial viz:

1. It's Free, 100% & Open Source
2. It's quick to code and highly customisable
3. Sitting within R, it enables seamless integration and mapping of the statistics & data science outputs from your analysis.

Step 0

Install the `ggmap` package in R

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

Load the `ggmap` library

```
library(ggmap)
```

Step 1

Get your map by calling the `get_map` function:

Here we want to view Victoria with a view to plotting the location of ANZ HQ...

```
basemap <- get_map(location = "Melbourne, Victoria, 3000", zoom = 8, maptype = "hybrid")
```

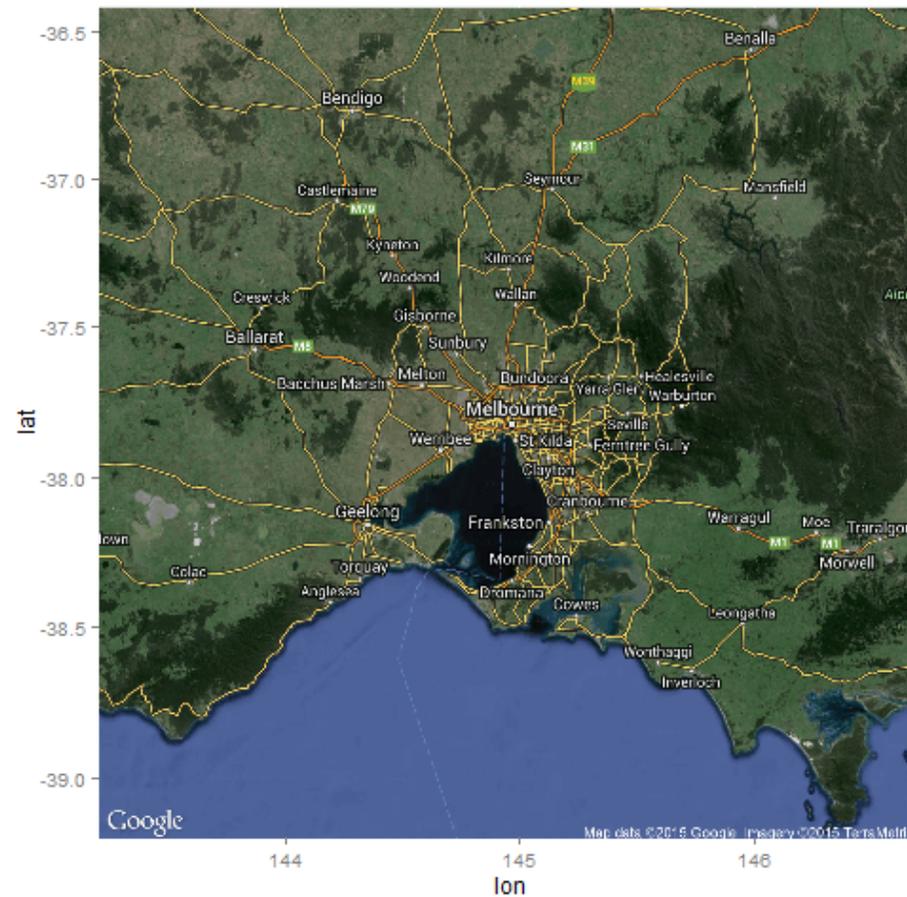
For a full list of customisable options type:

```
??get_map
```

Step 2

To generate the map, call the `ggmap` function, feeding in the 'basemap' variable we assigned in the previous slide:

```
ggmap(ggmap = basemap)
```



Step 3

To Geo-Code a location, feed in a text string of the address.

In this case we want to pin-point the location of ANZ Headquarters at Docklands.

```
plotpoint <- geocode(location = "833 Collins Street, Melbourne, VIC, 3000")
```

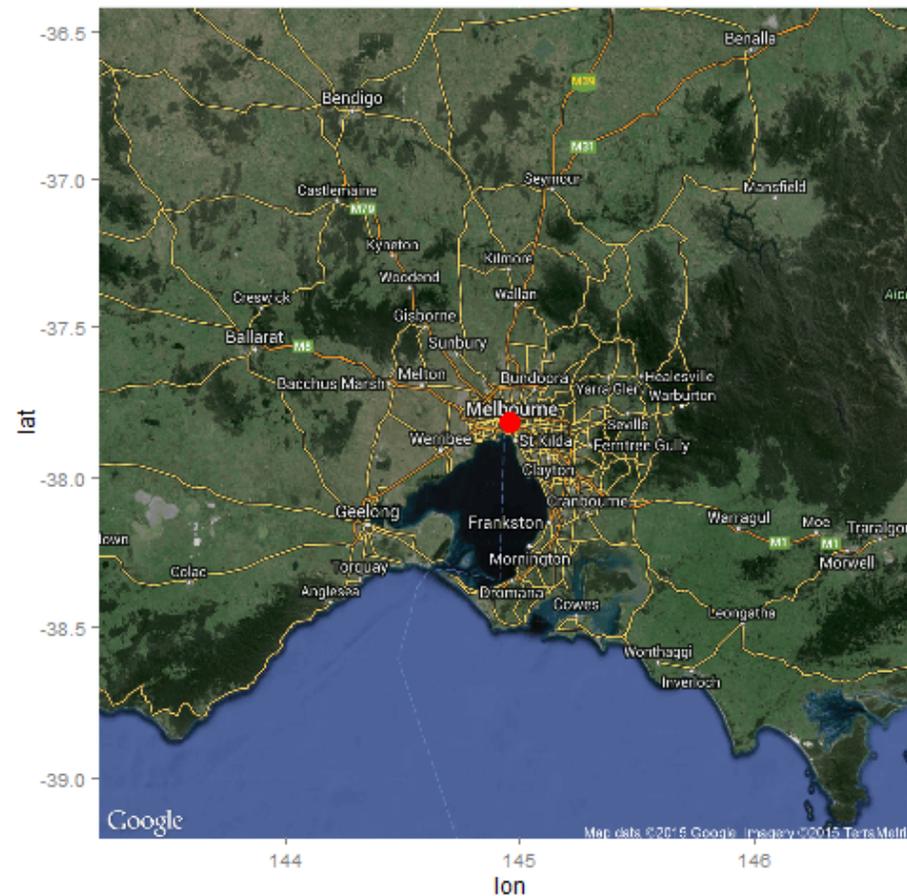
A benefit of using an API is the location data can be semi-structured.

The `geocode` function can also accept a vector of location data and return a dataset of the lat's and lon's, rather than just a single point.

Step 4

To overlay your geocoded data, simply call the `ggmap` function again, + overlay your `geom_point` for the red dot to represent the ANZ HQ location.

```
ggmap(basemap) + geom_point(aes(x = lon, y = lat), data = plotpoint, color = "red", size = 5)
```



Going further

- Incorporating more points of interest can make some really interesting maps. Plotted data can be customised and presented as points, polygons, density maps, heat maps etc.
- The main strength of using R here is not its GIS capabilities but its highly flexible nature and its ability to incorporate the outputs of your statistical methods for your geo-data such as k-means or k-medoids clustering, distance matrices, self-organising maps etc.
- There are also loads of cheat-sheets and help guides by contributors on making the most of the visualisations.
- As a public API is being used, it must be used in conjunction with the Google Maps API Terms of Service : <http://developers.google.com/maps/terms> and comply with internal information security guidelines

Full R Code

```
#load packages
library(ggmap)

#specify variables
maplocation <- "Melbourne, Victoria, 3000"
zoomfactor <- 8
maptype <- "hybrid"
pointsofinterest <- "833 Collins Street, Melbourne, VIC, 3000"

#load the base-map
basemap <- get_map(location = maplocation, zoom = zoomfactor, maptype = maptype )

#geocode your points of interest
plotpoints <- geocode(location = pointsofinterest)

#call the base map with points layered on top
ggmap(basemap) + geom_point(aes(x = lon, y = lat), data = plotpoints, color = "red", size = 5)
```