

ggmap_tutorial

- **Authors:** Catrina Loucks
- **Research field:** Genetics
- **Lesson Topic:** In this lesson we will be using ggmap to create maps showing detailed spatial information

Motivation

- Plotting points on maps becomes highly accurate, flexible and reproducible
- ggmap uses the grammar of graphics that underlies the ggplot2 package

First make sure you have ggplot2 and ggmap

```
#Make sure ggplot2 is installed
if(!require("ggplot2")) {
  install.packages("ggplot2")
  library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
#Make sure ggmap is installed
if(!require("ggmap")) {
  install.packages("ggmap")
  library(ggmap)
}
```

```
## Loading required package: ggmap
```

```
## Warning: package 'ggmap' was built under R version 3.2.3
```

- Currently, Tiffany Timbers and I are looking for the genetic basis of individual phenotypic differences
- We are using worms as a model system and investigating 40 strains sampled from various locations around the world

Then load the data

```
#load and take a look at data
#set working directory to folder containing the worms_strains.csv file for this tutorial
setwd("~/Documents/PhD/Classes/Scientific_programming_study_group/ggmap")
#make an object called worm_strains to hold the table
worm_strains <- read.table("worm_strains.csv", header=TRUE, sep=",")
#take a look at worm_strains, containing four columns (strain name, location sampled, latitude sampled, longitude sampled)
head(worm_strains)
```

```
##   strain      location
## 1 AB1 Adelaide, Australia
## 2 AB3 Adelaide, Australia
## 3 CB4853 Altadena, USA
## 4 CB4854 Altadena, USA
## 5 CB4856 Hawaii
## 6 CB4858 Pasadena, USA
```

Now find the location of the first strain: AB1

```
#make an object for the location where AB1 was sampled
AB1_location <- "Adelaide, Australia"
#you can also define the location using the latitude/longitude coordinates (geocode can determine these)
geocode("Adelaide, Australia")
```

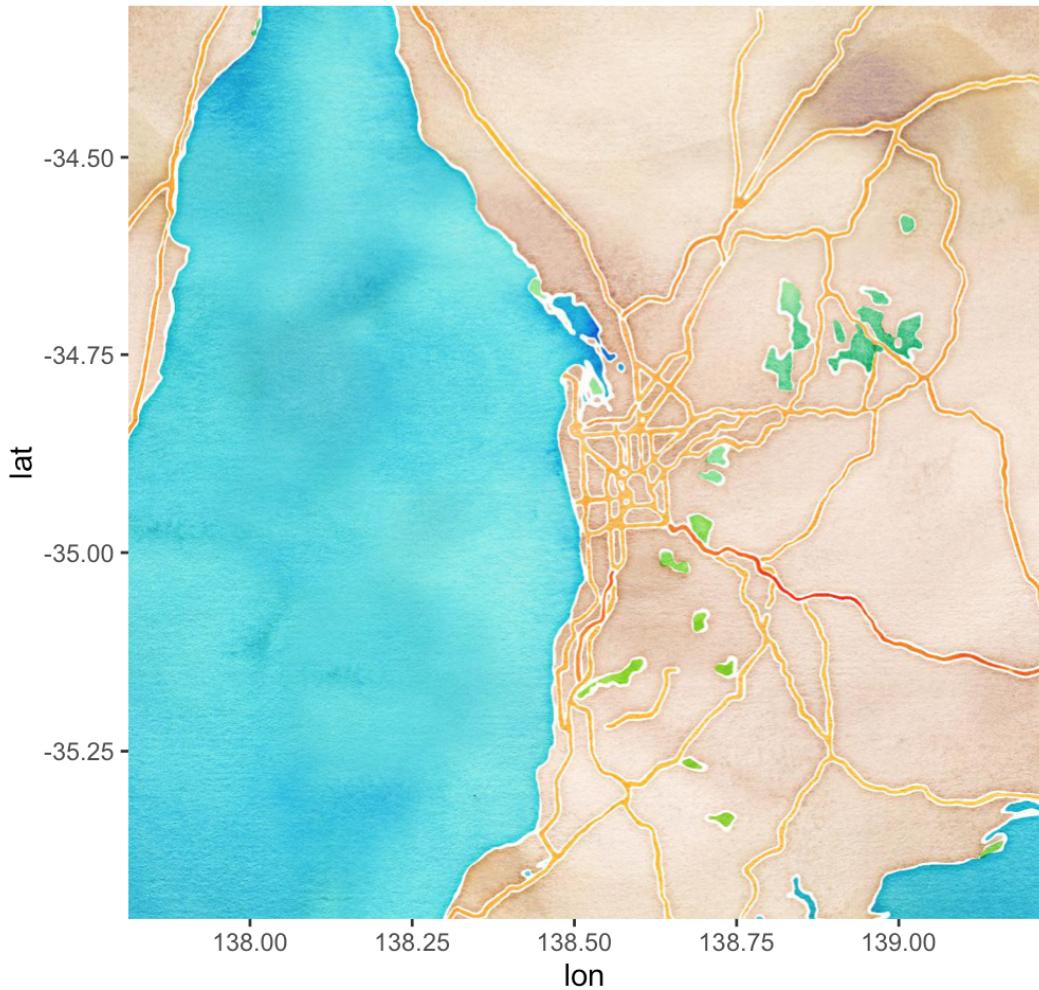
```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Adelaide,%20Australia&sensor=false
```

```
##   lon      lat
## 1 138.6 -34.92862
```

```
AB1_location <- c(lon=138.6, lat=-34.92862)
#find a map including the location of AB1
AB1_map <- get_map(location=AB1_location, source="stamen", maptype="watercolor", crop=FALSE)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=-34.92862,138.6&zoom=10&size=640x640&scale=2&maptype=terrain&sensor=false
## Map from URL : http://tile.stamen.com/watercolor/10/904/616.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/905/616.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/906/616.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/907/616.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/904/617.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/905/617.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/906/617.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/907/617.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/904/618.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/905/618.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/906/618.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/907/618.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/904/619.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/905/619.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/906/619.jpg
## Map from URL : http://tile.stamen.com/watercolor/10/907/619.jpg
```

```
ggmap(AB1_map)
```



Challenge 1

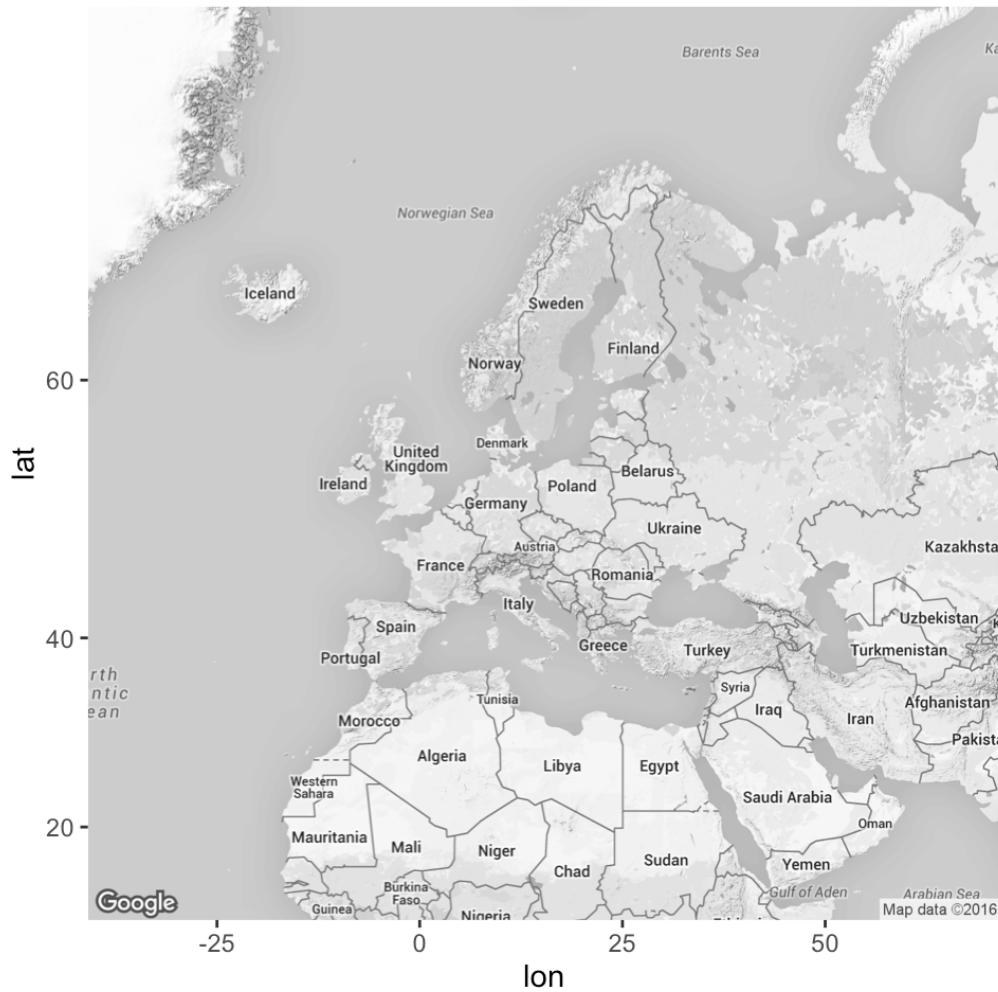
Use the ggmap cheatsheet to get a map showing the location of strain JU263 - use a different source for your map - try making the map black and white - zoom to the level of the continent

Now we are going to plot the locations of all our worm strains on a map

```
#find a map including with the location set as Europe
europe_map <- get_map(location="Europe", source="google", maptype="terrain", color = "bw", zoom=3)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Europe&zoom=3&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Europe&sensor=false
```

```
ggmap(europe_map)
```



```
#first determine latitude/longitude locations for each strain and add them to the
worms_strains datafram
#make sure the location column is in the character class
worm_strains$location <- as.character(worm_strains$location)
#make new columns for the longitude and latitude coordinates for each worm strain
worm_strains$lon <- geocode(worm_strains$location)[1] #lon is first output for geo
code
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Adelaide,%20Australia&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Adelaide,%20Australia&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Altadena,%20USA&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Altadena,%20USA&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hawaii&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Pasadena,%20USA&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Edinburgh,%20Scotland&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Edinburgh,%20Scotland&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Johannesburg,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ceres,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ceres,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Limuru,%20Kenya&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Limuru,%20Kenya&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Wuhan%20City,%20Hubei%20Province,%20China&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Kakegawa,%20Japan&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Concepcion,%20Chile&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Sevilla,%20Spain&sensor=false
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Carmona,%20Spain&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Montevideo,%20Uruguay&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Blanc,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Blanc,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Merlet,%20France&sensor=false
```

```
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Franconville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Franconville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hermanville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hermanville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Primel%20Tregastel,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Perreux-sur-Marne,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lisbon,%20Portugal&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Wuhan%20City,%20Hubei%20Province,%20China&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Madagascar&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lingen,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Mecklenbeck,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Mecklenbeck,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Roxel,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Roxel,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lincoln%20City,%20USA&sensor=false
```

```
worm_strains$lat <- geocode(worm_strains$location)[2] #lat is seconf output for geocode
```

```
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Adelaide,%20Australia&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Adelaide,%20Australia&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Altadena,%20USA&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Altadena,%20USA&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hawaii&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Pasadena,%20USA&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Edinburgh,%20Scotland&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Edinburgh,%20Scotland&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Johannesburg,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ceres,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ceres,%20South%20Africa&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Limuru,%20Kenya&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Limuru,%20Kenya&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Wuhan%20City,%20Hubei%20Province,%20China&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Kakegawa,%20Japan&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Concepcion,%20Chile&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Sevilla,%20Spain&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Carmona,%20Spain&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Montevideo,%20Uruguay&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Blanc,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Blanc,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Merlet,%20France&sensor=false
```

```

## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Franconville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Franconville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hermanville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Hermanville,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Primel%20Tregastel,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Perreux-sur-Marne,%20France&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lisbon,%20Portugal&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Wuhan%20City,%20Hubei%20Province,%20China&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Madagascar&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lingen,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Mecklenbeck,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Mecklenbeck,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Roxel,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Roxel,%20Germany&sensor=false
## .Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Lincoln%20City,%20USA&sensor=false

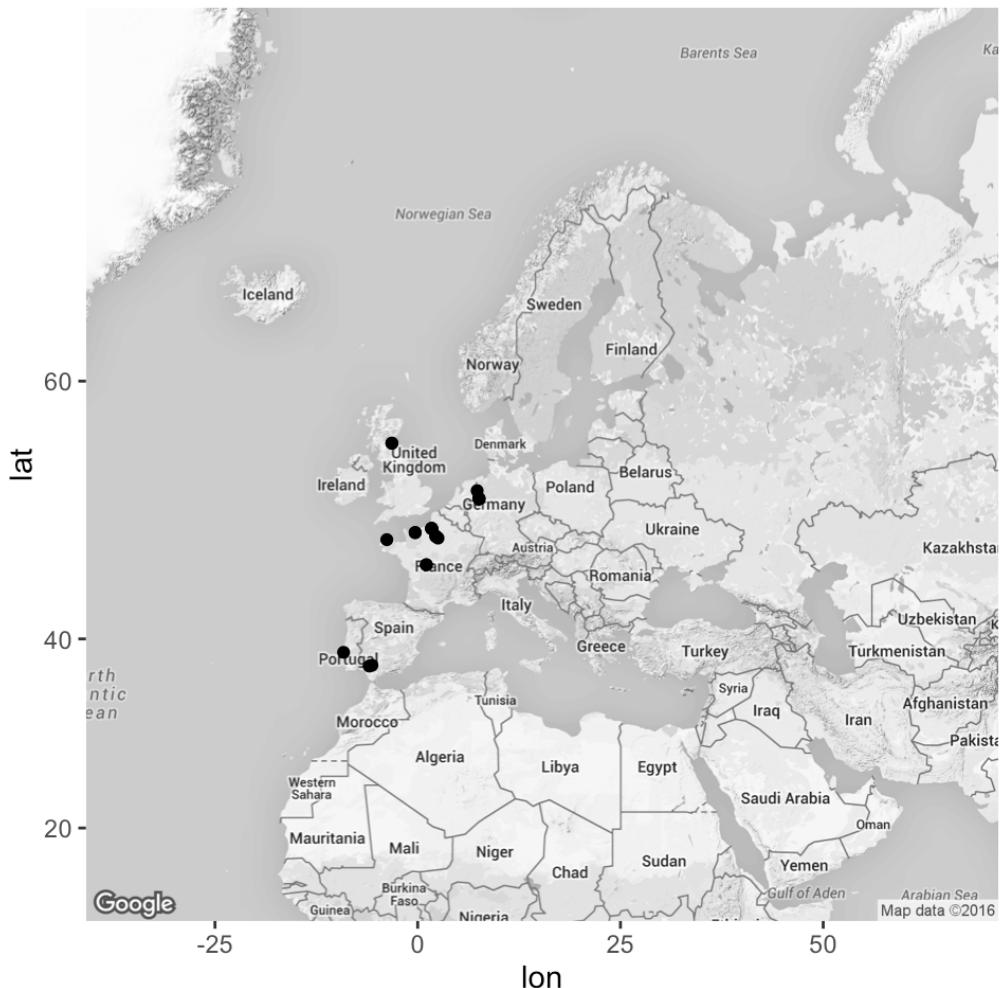
```

```

#now we need to plot the strains on the Europe map we found
ggmap(europe_map) +
  geom_point(data=worm_strains, aes(x=lon, y=lat)) #aes is short for aesthetics

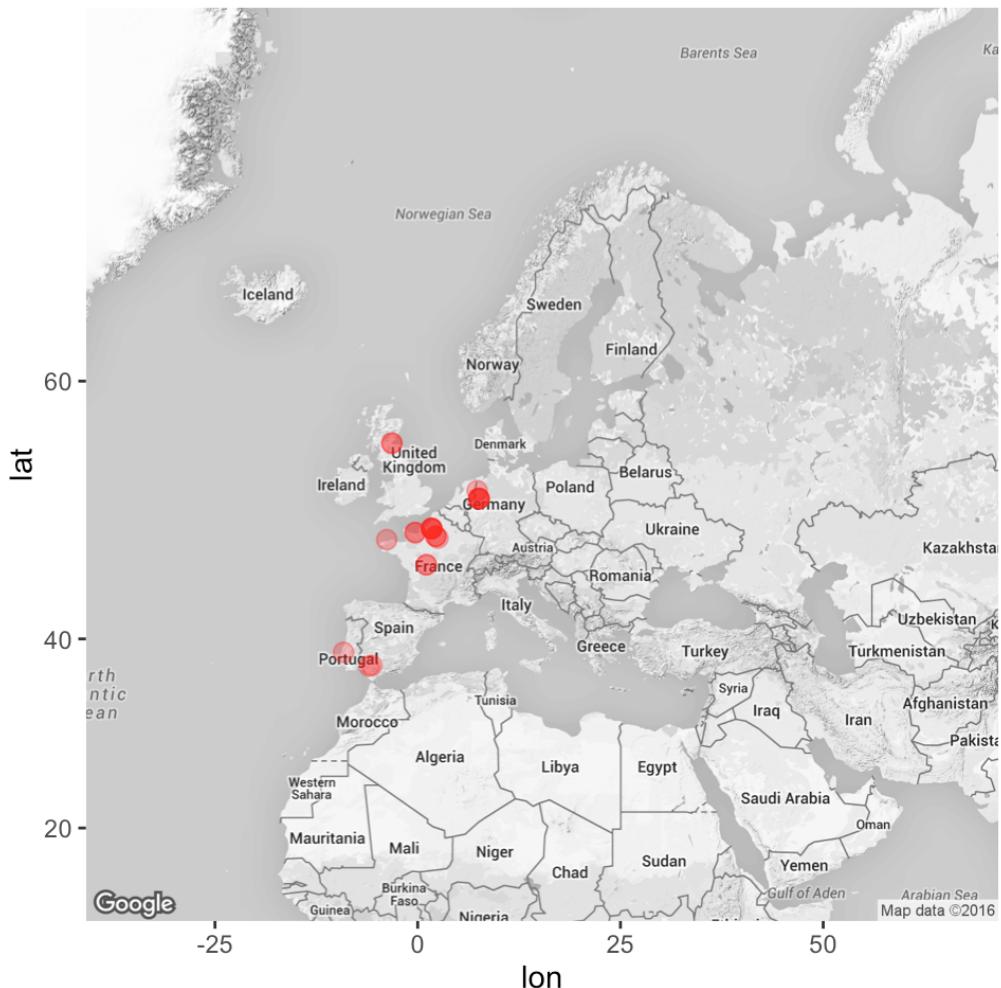
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



```
#play around with parameters
ggmap(europe_map) +
  geom_point(data=worm_strains, aes(x=lon, y=lat), alpha=0.3, color="red", size=3)
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```



Now we can save our Europe map

```
ggsave("Europe.pdf", useDingbats=FALSE, height=6, width=9)
```

```
## Warning: Removed 18 rows containing missing values (geom_point).
```

Unfortunately, ggmap does not handle full maps, but we can use ggplot2 to do this

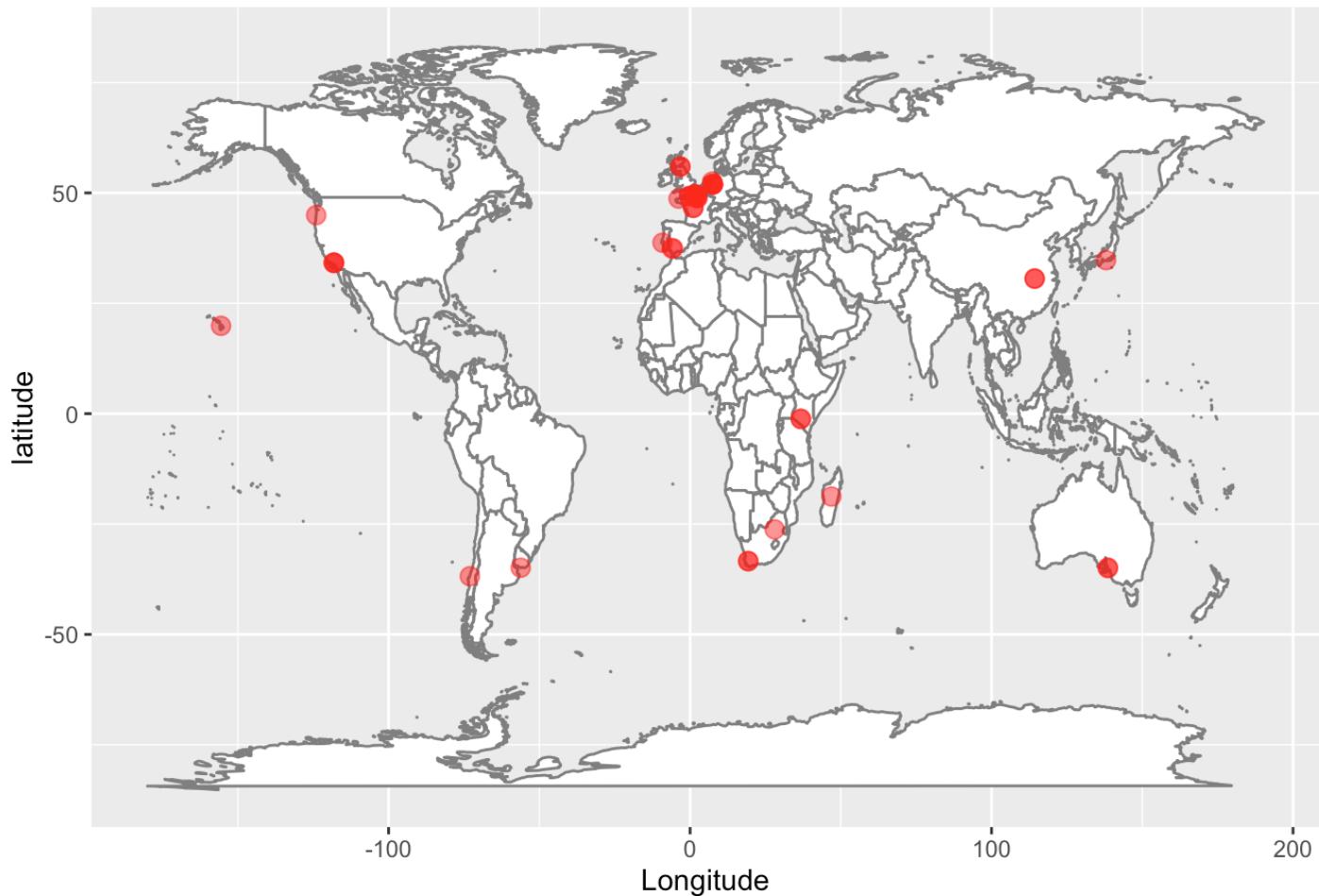
```
world_map <- ggplot() +
  borders("world", fill = "white", colour = "grey50") + # create a layer of borders
  geom_point(data=worm_strains, aes(x = lon, y = lat), colour = "red", size = 3, alpha = 0.5) +
  theme(legend.position="none") +
  ggtitle("Worm strains across the world") +
  labs(x="Longitude", y="latitude")
```

```
## Warning: package 'maps' was built under R version 3.2.3
```

```
##  
## # maps v3.1: updated 'world': all lakes moved to separate new #  
## # 'lakes' database. Type '?world' or 'news(package="maps")'. #
```

world_map

Worm strains across the world



Now we can save our world map

```
ggsave("World.pdf", useDingbats=FALSE, height=6, width=9)
```

Challenge 2

Use the ggmap cheatsheet to make a label on the world map for CB4856

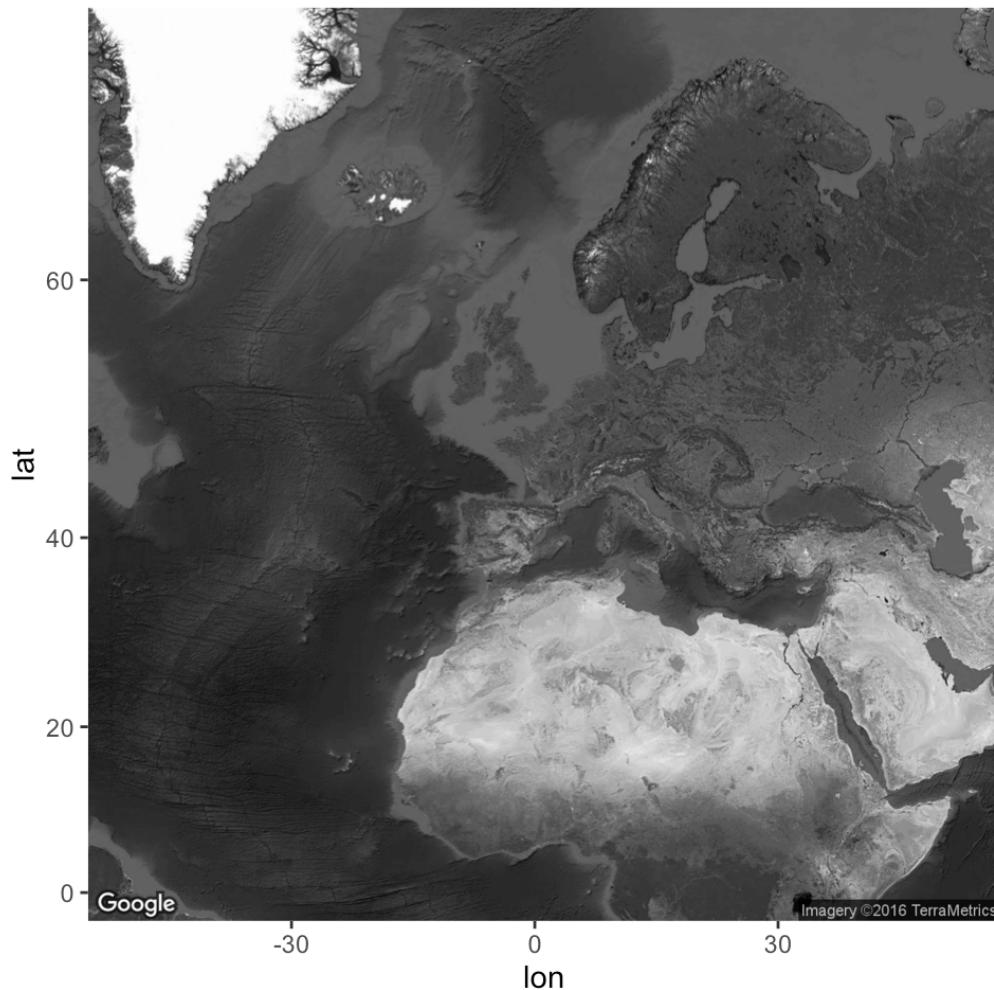
- hint: you will have to use the latitude and longitude coordinates for that strain
- make sure to put the annotation above the title and x/y axis labels in the code

Answer to challenge 1

```
#make an object for the location where JU263 was sampled  
JU263_location <- "Le Blanc, France"  
#find a map including the location of JU263  
JU263_map <- get_map(location=JU263_location, source="google", maptype="satellite", color = "bw", zoom=3)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Le+Blanc,+France&zoom=3&size=640x640&scale=2&maptype=satellite&language=en-EN&sensor=false  
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Le%20Blanc,%20France&sensor=false
```

```
ggmap(JU263_map)
```



Answer to challenge 2

```
world_map <- ggplot() +  
  borders("world", fill = "white", colour = "grey50") + # create a layer of border  
s  
  geom_point(data=worm_strains, aes(x = lon, y = lat), colour = "red", size = 3, a  
lpha = 0.5) +  
  theme(legend.position="none") +  
  annotate("rect", xmin=-173.582782, ymin=26.896766, xmax=-137.582782, ymax=32.896  
766, col="blue", fill="white") +  
  annotate("text", x=-155.582782, y=29.896766, label = 'CB4856', colour = "blue",  
size = 4) +  
  annotate("segment", x=-155.582782, xend=-155.582782, y=26.896766, yend=21.89676  
6,  
         colour="blue", arrow = arrow(length=unit(0.2,"cm")), size = 1) +  
  ggtitle("Worm strains across the world") +  
  labs(x="Longitude", y="latitude")  
world_map
```

