Geographic Mapping Techniques

Jedidiah Harwood[*]
Kurt Wydrinski[†]

11/22/2019

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(maps)
```

```
##
## Attaching package: 'maps'
## The following object is masked from 'package:purrr':
##
##     map
```

Introduction

There are a number of techniques available for creating geographic maps within R. Techniques using both the `maps`, `ggplot` and other `tidyverse` packages are demonstrated in the following sections.

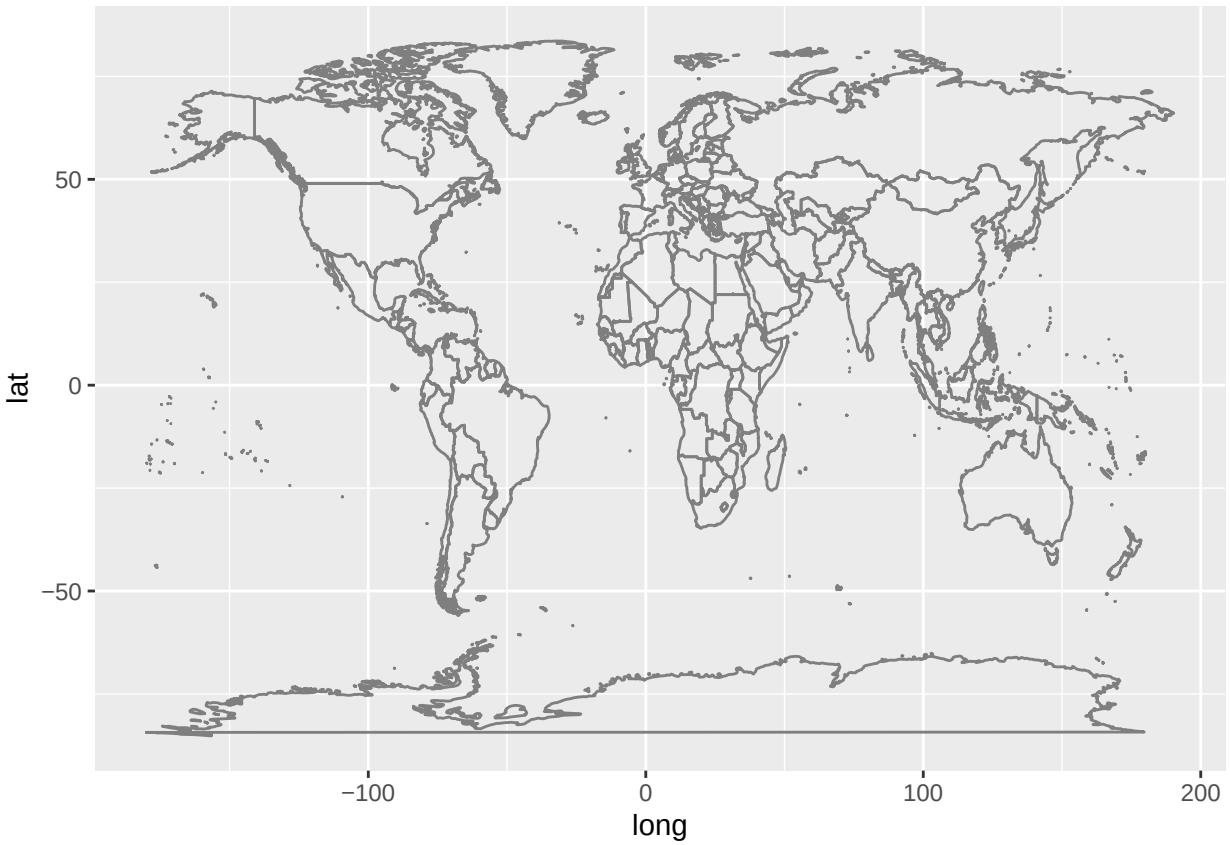
Basic Techniques

World Maps

The code below demonstrates the simplest world map. By default, borders are drawn 1 unit thick in a medium grey color. Also notice that the map is centered on the Prime Meridian at 0° longitude.

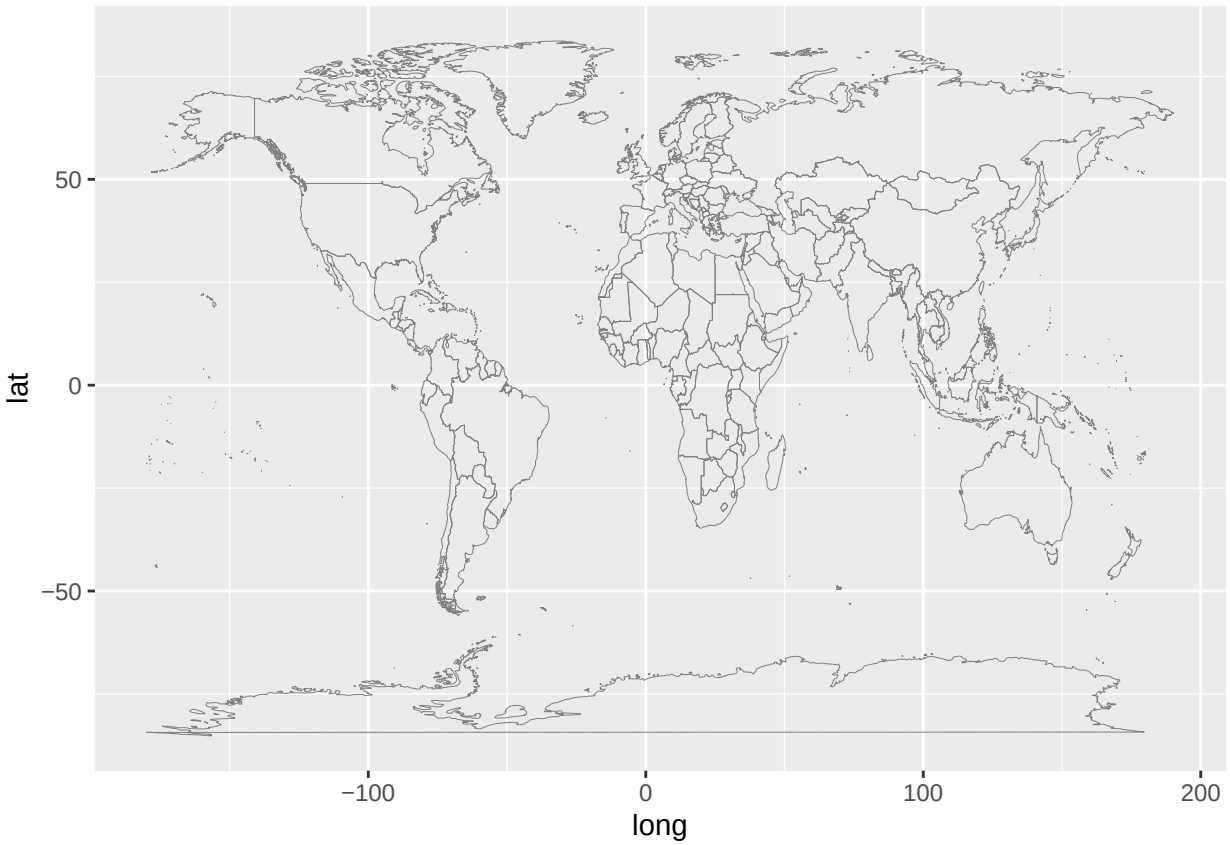
```
ggplot() +
  borders()
```

[*]California State University - East Bay
[†]California State University - East Bay



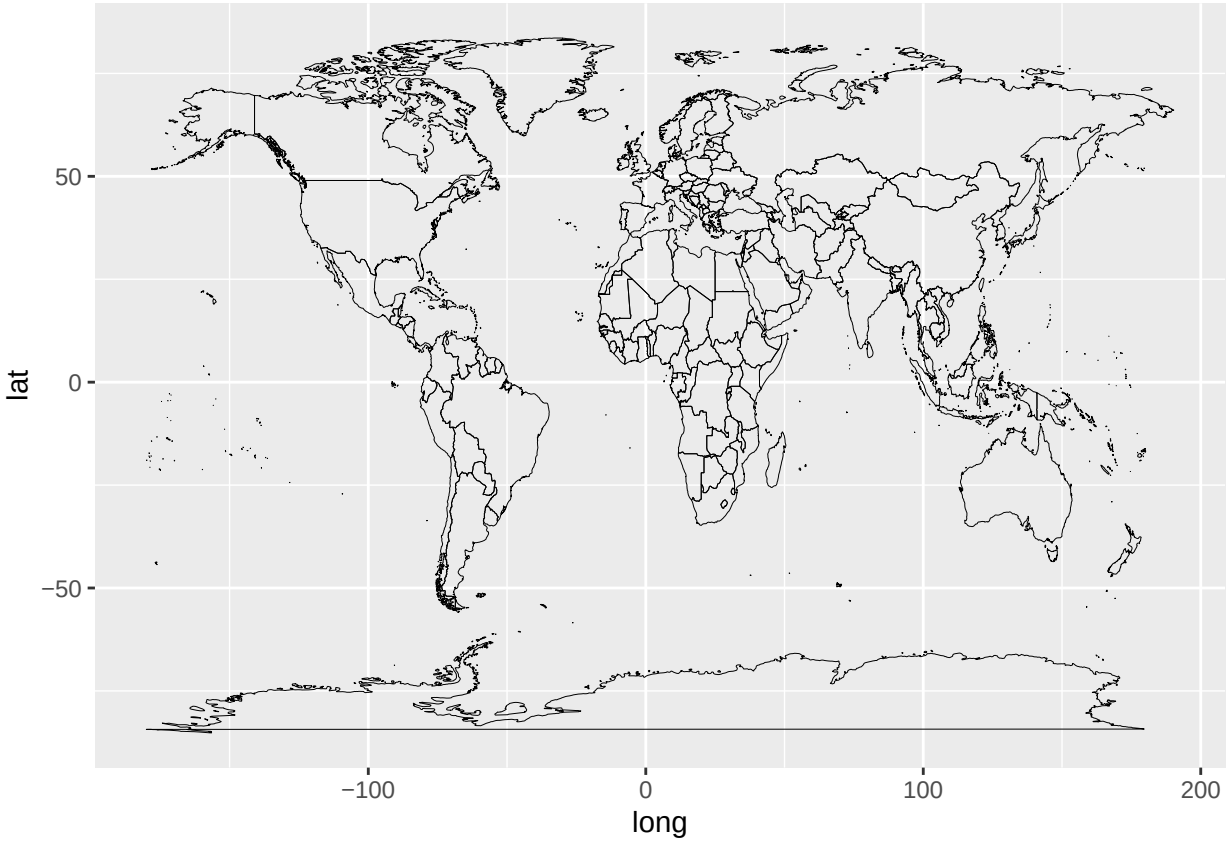
The simple map is just that, simple but it's not very appealing. Adding the `size` graphic aesthetic parameter to the borders call, thinner border line can be used in which improves the appeal of the map.

```
ggplot() +
  borders(size = 0.1)     # thin borders
```



While the thin borders are an improvement, they become harder to see due to the default medium-grey color. Using the `colour` parameter of `borders()` allows the borders to be rendered in different colors. The code below demonstrates using a thin, black border. This fine-line border looks cleaner and more refined than the default border rendering.

```
ggplot() +
  borders(size = 0.1,
          colour = "black")     # thin, black borders
```



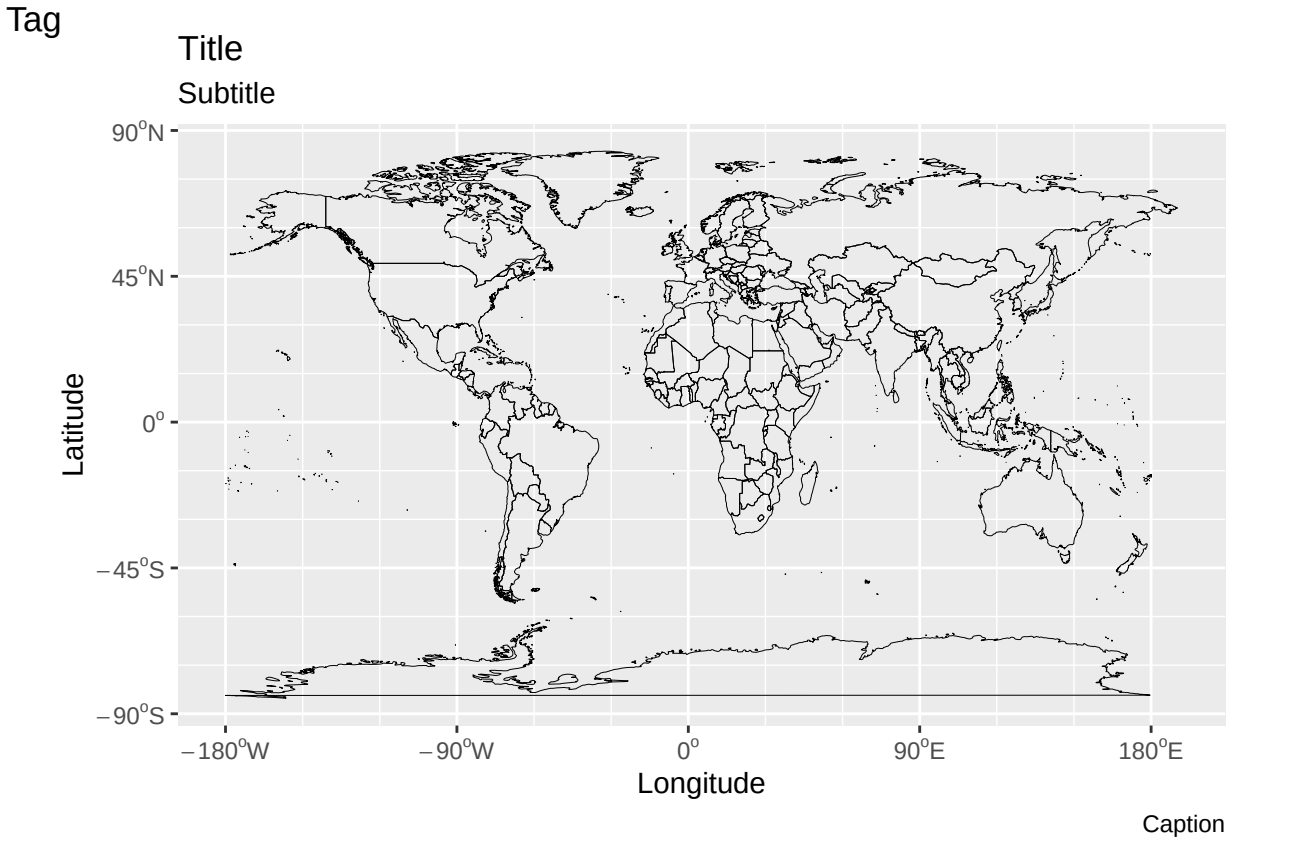
The axes, titles, captions, etc. are either missing or not labelled very clearly in the above plot. The following actions will address this:

1. Re-scale axes and label with proper units-of-measure.
2. Clearly label variables represented on the axes.
3. Add placeholder titling elements for further clarity. However, these are placeholders because it is expected that each world map data visualization will be appropriately titled for the information it is conveying. These guidelines should be followed when choosing appropriate titling elements:
    1. `title` - Indicate the main subject/point of data visualization (e.g. *Country Population*)
    2. `subtitle` - Indicate constraints, filters or transformations applied to the underlying data (e.g. *(by most recent reported census)*, *(in thousands of people)*, etc.)
    3. `caption` - **TODO: define standard**
    4. `tag` - **TODO: define standard**

The code below applies to actions to improve the ease in understanding the map.

```
ggplot() +
  borders(size = 0.1,
          colour = "black") +    # thin, black borders
  scale_x_continuous(breaks = seq(-180, 180, 90),
                     minor_breaks = seq(-180, 180, 30),
                     labels = unlist(
                       lapply(seq(-180, 180, 90),
                              function(x) ifelse(x < 0,
                                                 parse(text = paste0(x, "^o", "*W")),
                                                 ifelse(x > 0,
                                                        parse(text = paste0(x, "^o", "*E")),
```

```
                                                    parse(text = paste0(x, "^o"))))
                            ))) +
  scale_y_continuous(breaks = seq(-90, 90, 45),
                     minor_breaks = seq(-90, 90, 15),
                     labels = unlist(
                       lapply(seq(-90, 90, 45),
                              function(x) ifelse(x < 0,
                                                 parse(text = paste0(x, "^o", "*S")),
                                                 ifelse(x > 0,
                                                        parse(text = paste0(x, "^o", "*N")),
                                                        parse(text = paste0(x, "^o"))))
                            ))) +
  labs(title = "Title",
       subtitle = "Subtitle",
       caption = "Caption",
       tag = "Tag",
       x = "Longitude",
       y = "Latitude")
```



In the code below, a refactoring has been performed to extract the boilerplate code for the country borders and axes scales. Notice that the code return a list of items that include a `layer` object containing the borders and two `scale` objects for the axes. The `+.gg` operator allows various types of plot components to be added to a plot, including layers and scales. These can either be added individually or a list of this can be added with a single `+` operator.

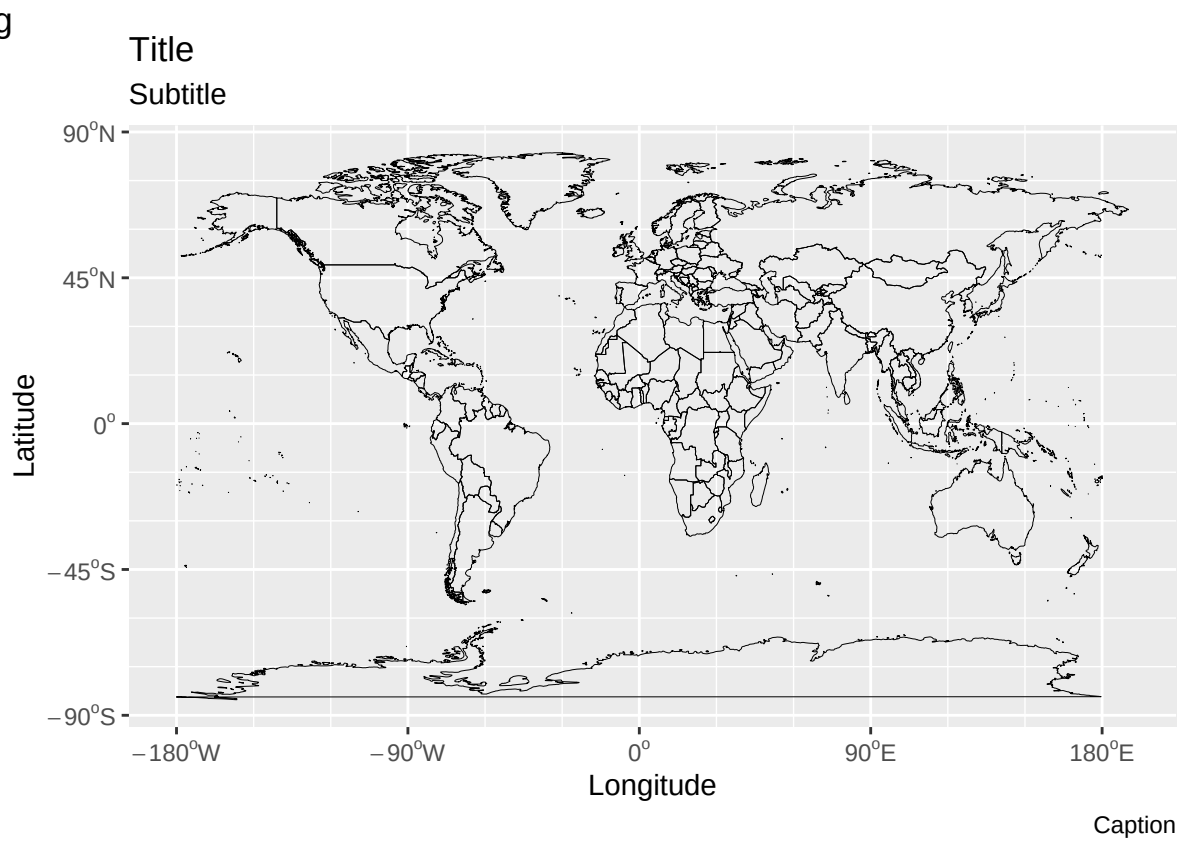
The function below extracts the common plot components into a single function call that can be used as the basis for global maps. By doing so, this code can then be used within a `ggplot()` call sequence to quickly

add the borders and scales that are common to all global maps. Effectively, creating a function like this that returns a list of ggplot components acts like a macro. It is suggested that any functions like this be prefixed with `mg_` by name to indicate they are a macro ggplot operation.

The use and benefit of this is demonstrated in the following code. Notice the structure of the function call that returns a `list`. Also note the conciseness of the ggplot call sequence. As long as these macro calls return a list of addable ggplot components, these calls can be placed anywhere inside the call sequence.

```
mg_global_map <- function() {
  return(list(
    borders(size = 0.1,
            colour = "black"),
    scale_x_continuous(breaks = seq(-180, 180, 90),
                       minor_breaks = seq(-180, 180, 30),
                       labels = unlist(
                         lapply(seq(-180, 180, 90),
                                function(x) ifelse(x < 0,
                                                   parse(text = paste0(x, "^o", "*W")),
                                                   ifelse(x > 0,
                                                          parse(text = paste0(x, "^o", "*E")),
                                                          parse(text = paste0(x, "^o"))))
                                ))),
    scale_y_continuous(breaks = seq(-90, 90, 45),
                       minor_breaks = seq(-90, 90, 15),
                       labels = unlist(
                         lapply(seq(-90, 90, 45),
                                function(x) ifelse(x < 0,
                                                   parse(text = paste0(x, "^o", "*S")),
                                                   ifelse(x > 0,
                                                          parse(text = paste0(x, "^o", "*N")),
                                                          parse(text = paste0(x, "^o"))))
                                )))
    ))
}

ggplot() +
  mg_global_map() +
  labs(title = "Title",
       subtitle = "Subtitle",
       caption = "Caption",
       tag = "Tag",
       x = "Longitude",
       y = "Latitude")
```



World Maps with `ggmap` Package

The code in this section demonstrates how global maps can be rendered using the `ggmap` package. Not a lot of discussion is provided around the code examples because of clear disadvantages in using this approach for current projects.

`ggmap` Risks

1. The package relies on Google and other third party mapping services.
2. The Google mapping services require an API key to be administered within a registered Google account. This key needs to be configured in the application and managed. Changing of the Google account or API key enabled within Google could render the application unusable requiring reconfiguring of the credentials.
3. Requiring a third party key for an API exposes the key owner to risk if the provider's terms-of-service change in any way.
4. The styling of political boundaries (i.e. countries, states, etc.) is not a trivial process when using third party mapping services. The maps lend themselves to data overlays but the political boundaries being displayed in the maps may not be available with certain providers.

`ggmap` Rewards

1. Map backgrounds such as terrains, satellite views, or roads are easily obtainable.
2. API provides access to advanced features such as reverse geocoding, travel routes, etc.
3. Automated labelling of geographical boundaries at various levels such as oceans, countries, etc.

```r
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

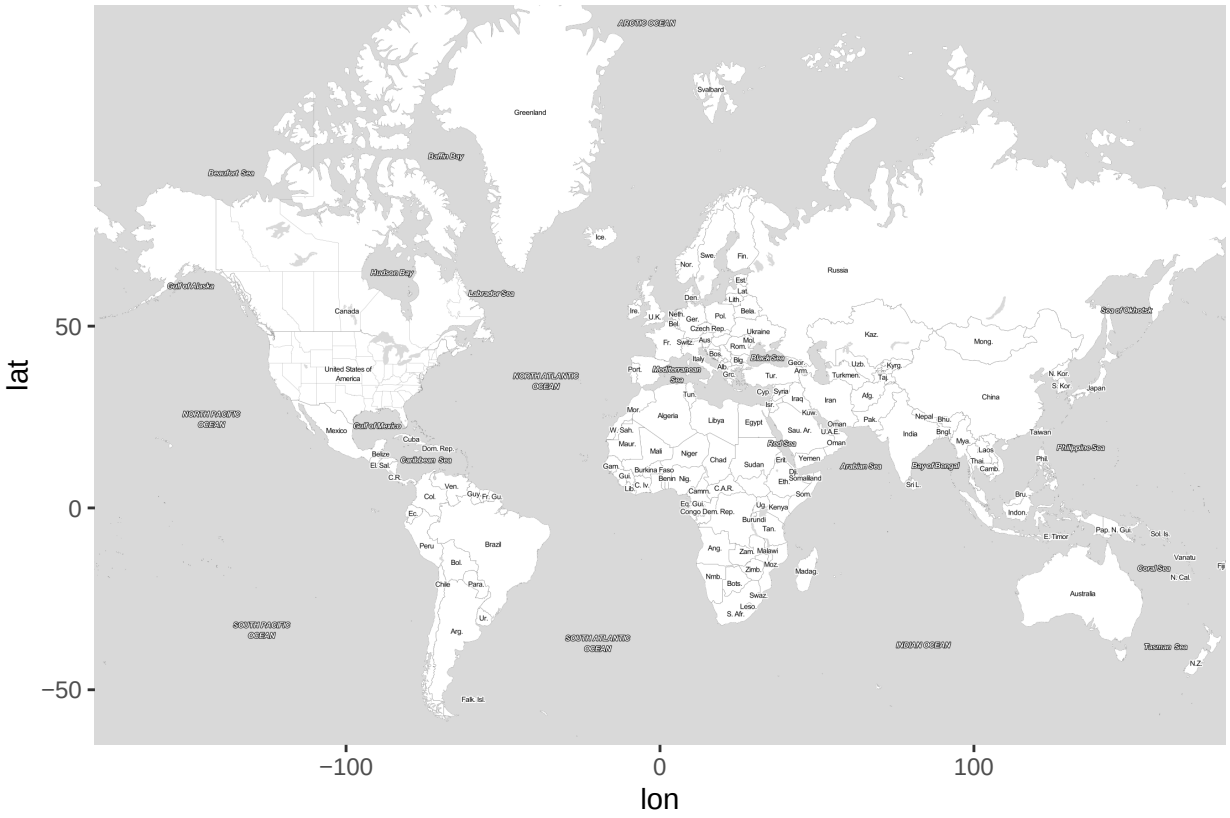
```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```r
world_map <- get_stamenmap(bbox = c(left = -179.99,
                                    bottom = -60.0,
                                    right = 180,
                                    top = 83),
                           maptype = "toner-lite",
                           zoom = 3,
                           force = TRUE,
                           crop = TRUE,
                           messaging = FALSE)
```

```
## 54 tiles needed, this may take a while (try a smaller zoom).
## Source : http://tile.stamen.com/toner-lite/3/0/0.png
## Source : http://tile.stamen.com/toner-lite/3/1/0.png
## Source : http://tile.stamen.com/toner-lite/3/2/0.png
## Source : http://tile.stamen.com/toner-lite/3/3/0.png
## Source : http://tile.stamen.com/toner-lite/3/4/0.png
## Source : http://tile.stamen.com/toner-lite/3/5/0.png
## Source : http://tile.stamen.com/toner-lite/3/6/0.png
## Source : http://tile.stamen.com/toner-lite/3/7/0.png
## Source : http://tile.stamen.com/toner-lite/3/8/0.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/0.png.
## Source : http://tile.stamen.com/toner-lite/3/0/1.png
## Source : http://tile.stamen.com/toner-lite/3/1/1.png
## Source : http://tile.stamen.com/toner-lite/3/2/1.png
## Source : http://tile.stamen.com/toner-lite/3/3/1.png
## Source : http://tile.stamen.com/toner-lite/3/4/1.png
## Source : http://tile.stamen.com/toner-lite/3/5/1.png
## Source : http://tile.stamen.com/toner-lite/3/6/1.png
## Source : http://tile.stamen.com/toner-lite/3/7/1.png
## Source : http://tile.stamen.com/toner-lite/3/8/1.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/1.png.
## Source : http://tile.stamen.com/toner-lite/3/0/2.png
## Source : http://tile.stamen.com/toner-lite/3/1/2.png
## Source : http://tile.stamen.com/toner-lite/3/2/2.png
## Source : http://tile.stamen.com/toner-lite/3/3/2.png
## Source : http://tile.stamen.com/toner-lite/3/4/2.png
## Source : http://tile.stamen.com/toner-lite/3/5/2.png
## Source : http://tile.stamen.com/toner-lite/3/6/2.png
## Source : http://tile.stamen.com/toner-lite/3/7/2.png
## Source : http://tile.stamen.com/toner-lite/3/8/2.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/2.png.
## Source : http://tile.stamen.com/toner-lite/3/0/3.png
## Source : http://tile.stamen.com/toner-lite/3/1/3.png
## Source : http://tile.stamen.com/toner-lite/3/2/3.png
```

```
## Source : http://tile.stamen.com/toner-lite/3/3/3.png
## Source : http://tile.stamen.com/toner-lite/3/4/3.png
## Source : http://tile.stamen.com/toner-lite/3/5/3.png
## Source : http://tile.stamen.com/toner-lite/3/6/3.png
## Source : http://tile.stamen.com/toner-lite/3/7/3.png
## Source : http://tile.stamen.com/toner-lite/3/8/3.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/3.png.
## Source : http://tile.stamen.com/toner-lite/3/0/4.png
## Source : http://tile.stamen.com/toner-lite/3/1/4.png
## Source : http://tile.stamen.com/toner-lite/3/2/4.png
## Source : http://tile.stamen.com/toner-lite/3/3/4.png
## Source : http://tile.stamen.com/toner-lite/3/4/4.png
## Source : http://tile.stamen.com/toner-lite/3/5/4.png
## Source : http://tile.stamen.com/toner-lite/3/6/4.png
## Source : http://tile.stamen.com/toner-lite/3/7/4.png
## Source : http://tile.stamen.com/toner-lite/3/8/4.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/4.png.
## Source : http://tile.stamen.com/toner-lite/3/0/5.png
## Source : http://tile.stamen.com/toner-lite/3/1/5.png
## Source : http://tile.stamen.com/toner-lite/3/2/5.png
## Source : http://tile.stamen.com/toner-lite/3/3/5.png
## Source : http://tile.stamen.com/toner-lite/3/4/5.png
## Source : http://tile.stamen.com/toner-lite/3/5/5.png
## Source : http://tile.stamen.com/toner-lite/3/6/5.png
## Source : http://tile.stamen.com/toner-lite/3/7/5.png
## Source : http://tile.stamen.com/toner-lite/3/8/5.png
## Not Found (HTTP 404). Failed to aquire tile /toner-lite/3/8/5.png.
```

```
ggmap(world_map)
```



US State Maps

```
map_data("world") %>%
  filter(region == "USA") %>%
  group_by(subregion) %>%
  summarise(count = n(),
            min_lat = min(lat),
            max_lat = max(lat),
            min_long = min(long),
            max_long = max(long)) %>%
  arrange(min_long)
```

```
## # A tibble: 32 x 6
##    subregion  count min_lat max_lat min_long max_long
##    <chr>      <int>   <dbl>   <dbl>    <dbl>    <dbl>
##  1 Alaska      3133    51.4    71.4   -178.     180.
##  2 Hawaii       130    19.0    22.2   -160.    -155.
##  3 <NA>        1990    25.1    49.4   -125.     -67.0
##  4 Washington    55    47.9    48.7   -123.    -122.
##  5 73            11    48.4    48.6   -123.    -123.
##  6 67             8    47.2    47.3   -123.    -123.
##  7 70            10    47.6    47.7   -123.    -122.
##  8 69            12    47.4    47.5   -123.    -122.
##  9 California    55    32.8    34.1   -120.    -118.
## 10 Texas         44    26.1    29.3    -97.4    -94.8
## # ... with 22 more rows
```

```
map_us <- map_data("world") %>%
  filter(region == "USA" & long < 0)
map_us_48 <- map_data("state")
```

```
ggplot(mapping = aes(x = long, y = lat,
                     group = group)) +
  geom_polygon(data = map_us,
               color = "black",
               size = 0.1,
               fill = NA) +
  geom_polygon(data = map_us_48,
               color = "black",
               size = 0.1,
               fill = NA)
```