Maping in R

Tutorial for making maps in R for Marine Ecology 2023 at BMSC

Dominique Maucieri

03 July, 2023

Contents

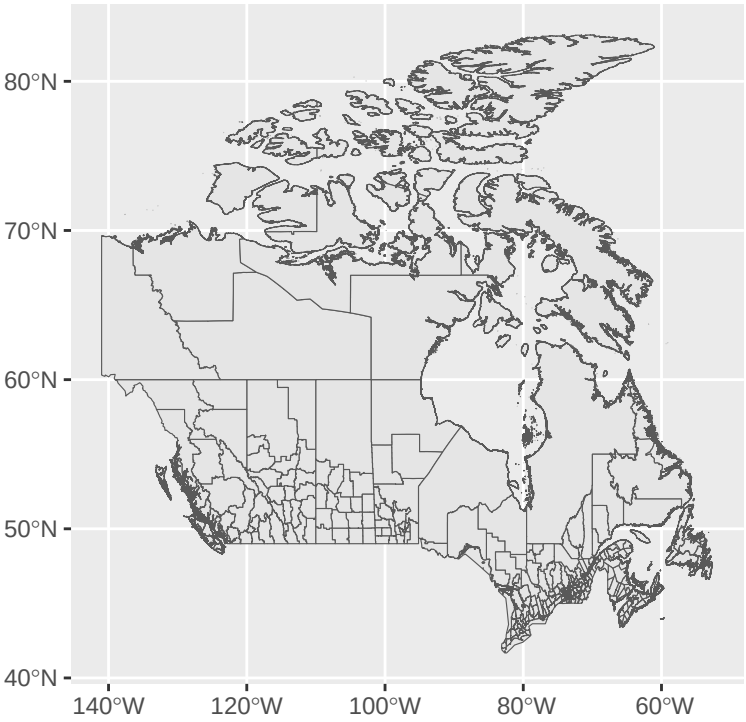
Acquiring base map layers

In order to have spatial shapes on our map, like our land masses, we need to find these "shapefiles" online. We are going to use shape files of Canada for this tutorial from DIVA-GIS https://www.diva-gis.org/gdata. You can look here or other places on the internet for other shapefiles, but we will be using Canada > Administrative Areas. These data have been downloaded for you and added to the folder so you don't have to download it now. Lets read it in and view it.

```
VI_baselayer <- readOGR(dsn = here::here("Data/CAN_adm", ""), stringsAsFactors = F)
```

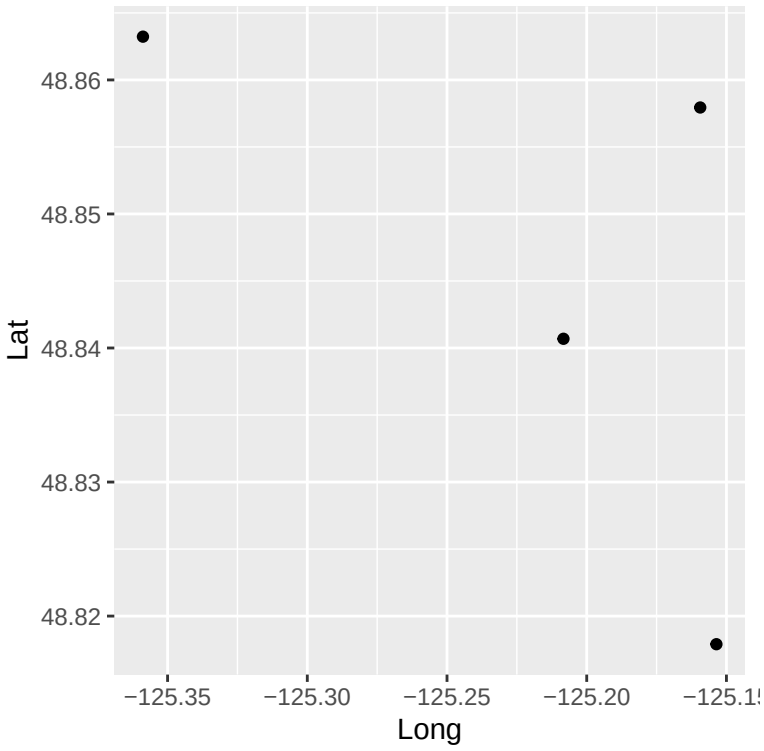
```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/dom/Library/CloudStorage/Dropbox/Documents/GitHub/RMapMaking_Tutorial/Data/CAN_adm",
## with 294 features
## It has 11 fields
## Integer64 fields read as strings:  ID_0 ID_1 ID_2
```

```
VI_baselayer_sf <- st_as_sf(VI_baselayer)

ggplot(VI_baselayer_sf) +
  geom_sf()
```



Now before we clean up the map, lets see what our data we are wanting to map will look like. For this, we are going to import latitude and longitude for our different field sites, along with some site environmental information.

```
SiteInfo <- read_excel(here::here("Data", "SiteInfo.xlsx"))

ggplot(SiteInfo, aes(x = Long, y = Lat)) +
  geom_point()
```



These GPS points look great for now but without the map behind them, its hard to actually visualize what they mean

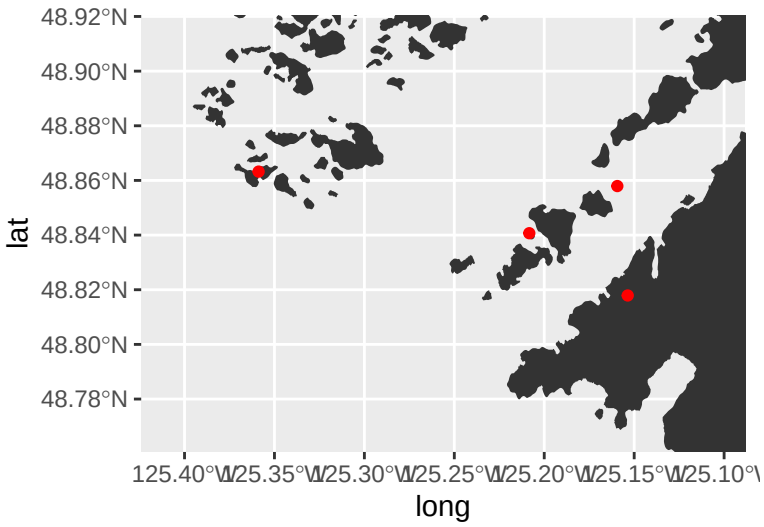
One important aspect of combining any spatial data together is that they all have to have the same projection. Because the Earth is round and not flat, in order to take the round Earth and show it on a 2D map, we have to "project" the 3D object in 2D. This will distort some of the continents, which is why on a world map some countries or land masses might look larger than others when they really aren't. When plotting data you need to make sure you choose a projection that will minimize distortion of your area of interest and all your spatial layer are distorted the same way.

For North America, we will use the World Geodetic System 1984 (WGS84) projection. Lets make sure our basemap is using this projection.

```
st_crs(VI_baselayer_sf) <- "+proj=longlat +datum=WGS84 +no_defs"
```

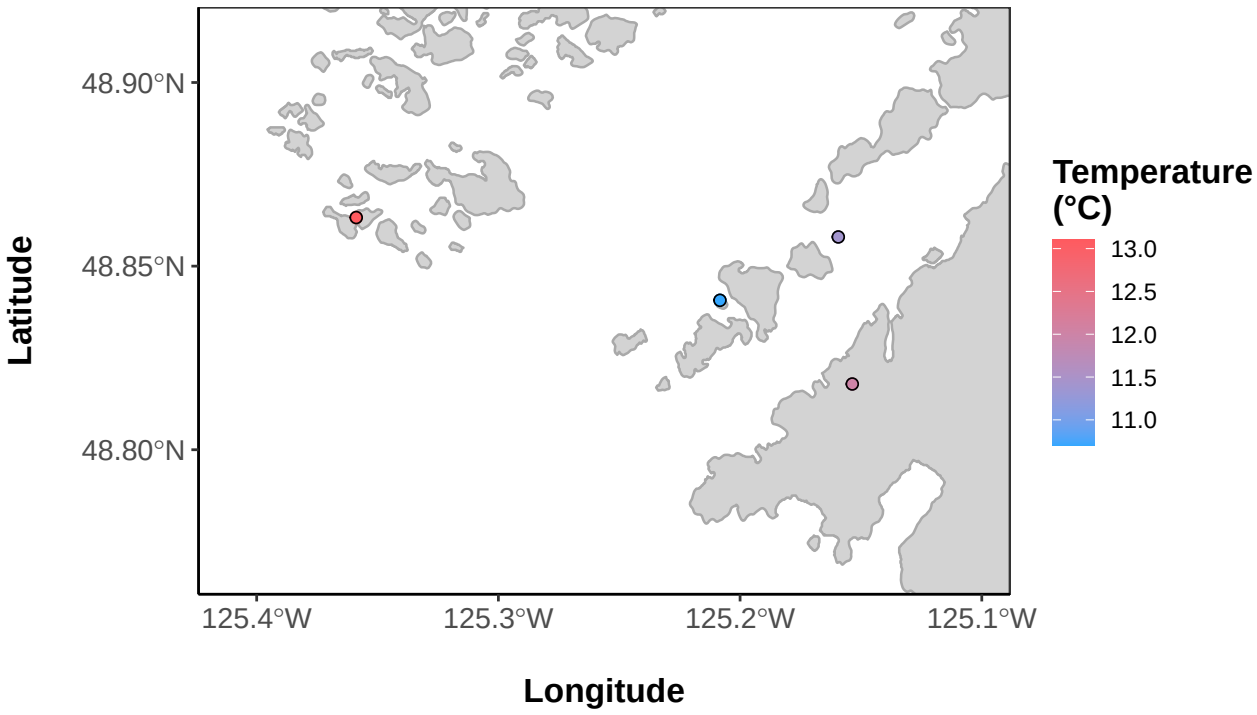
Now lets put the points and the map baselayer together and constrain the basemap to be only a bit larger than the data points of interest. There is no reason to have an entire map of Canada when we work on one small area of Vancouver Island.

```
lat_min = min(SiteInfo$Lat) - 0.05
lat_max = max(SiteInfo$Lat) + 0.05

long_min = min(SiteInfo$Long) - 0.05
long_max = max(SiteInfo$Long) + 0.05

ggplot(VI_baselayer_sf) +
geom_polygon(data = VI_baselayer,
          aes(x = long, y = lat, group = group)) +
  coord_sf(xlim = c(long_min, long_max),
           ylim = c(lat_min, lat_max)) +
  geom_point(data = SiteInfo, mapping = aes(x = Long, y = Lat), color = "red")
```



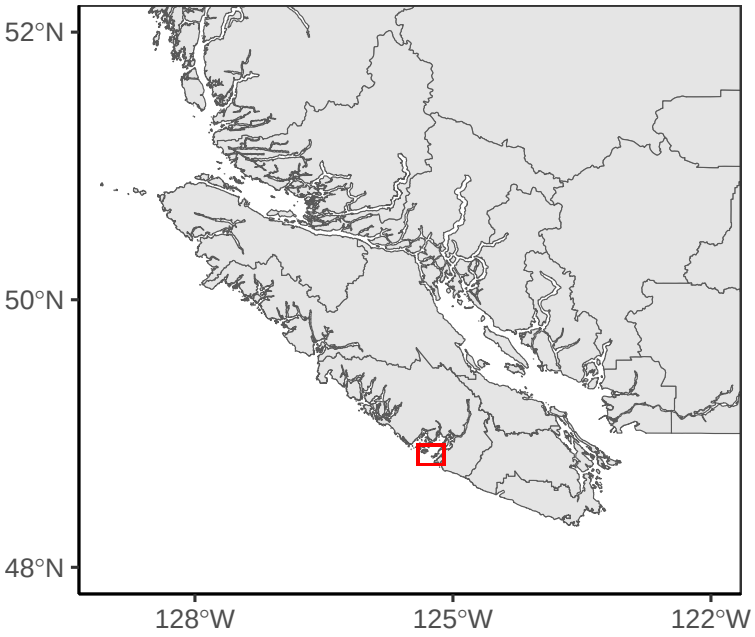
This map is great but its not very pretty, lets make it nicer. We will manually set the latitude and longitude scales and change the land colors. We can also change the points colors, maybe even make the color of the points be based on the sites temperature, or another variable. I will also add on a theme I have made for maps, which will make it aesthetically similar to plots I use in my papers. Feel free to use this theme or customize it to make your own.

```
#Map theme
theme.dgm.map <- function (){
  theme_bw(base_size = 12) +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "black"),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "plain"),
    legend.text = element_text(size = 10, face = "plain"),
    legend.title = element_text(size = 14, face = "bold"))
}

(basemap <- ggplot(VI_baselayer_sf) +
geom_polygon(data = VI_baselayer,
          aes(x = long, y = lat, group = group),
          color = "darkgrey", fill = "lightgrey") +
  coord_sf(xlim = c(long_min, long_max),
           ylim = c(lat_min, lat_max)) +
  geom_point(data = SiteInfo, mapping = aes(x = Long, y = Lat, color = Temp),
             size = 2,) +
  geom_point(data = SiteInfo, mapping = aes(x = Long, y = Lat, color = Temp),
             shape = 1, size = 2,colour = "black")+
  scale_color_gradient(low = "#35A7FF", high = "#FF5A5F" ) +
  labs(x = " \nLongitude", y = "Latitude\n ", color = "Temperature\n(°C)") +
  theme.dgm.map() +
  scale_x_continuous(breaks = c(-125.4, -125.3, -125.2, -125.1)) +
  scale_y_continuous(breaks = c(48.8, 48.85, 48.9)))
```



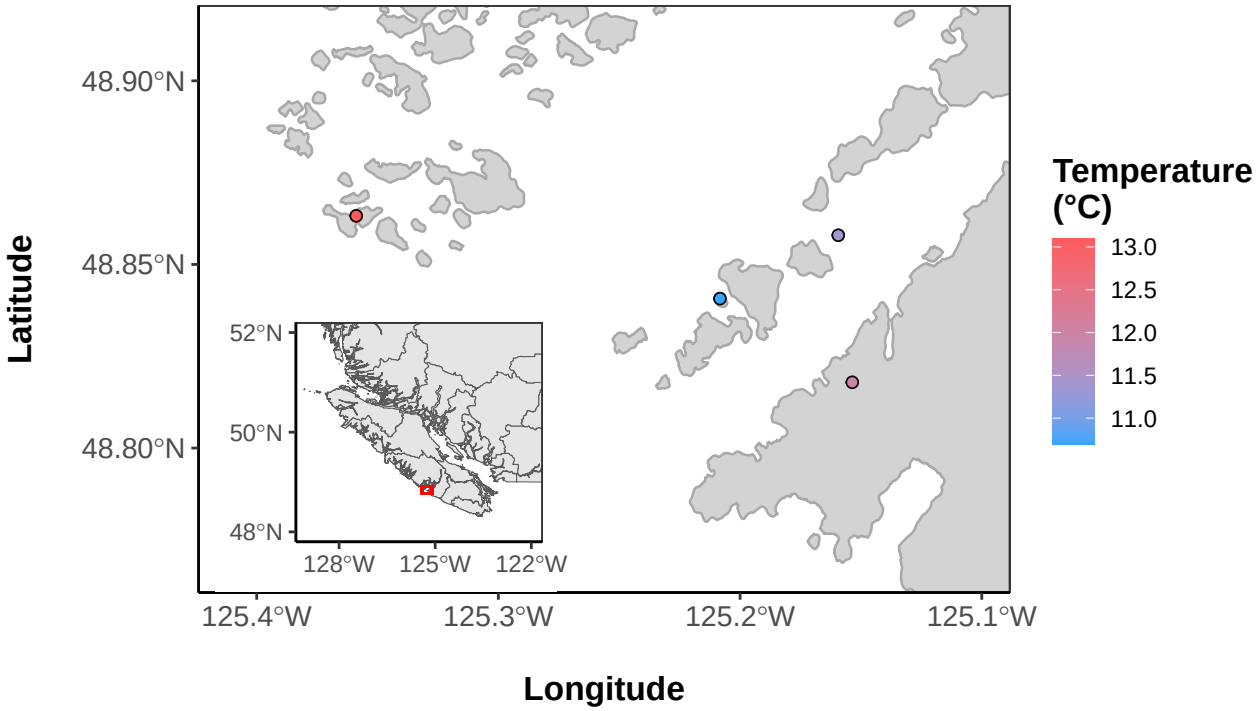
But what if you wanted an inset, because without knowing where Bamfield is, these islands could be confusing. So lets first make the inset map.

```
(VI_Inset <- ggplot(data = VI_baselayer_sf) +
  geom_sf() +
  coord_sf(ylim = c(48, 52), xlim = c(-129, -122)) +
  theme.dgm.map() +
  geom_rect(xmin = long_min, xmax = long_max, ymin = lat_min, ymax = lat_max,
            color = "red", fill = NA) +
  theme(axis.text = element_text(size = 10))+
  scale_x_continuous(breaks = c(-128, -125, -122)) +
  scale_y_continuous(breaks = c(48, 50, 52)))
```



Now that its looking great lets put the two together. This will take a bit of back and forth to place the inset in the right spot, but in the end it will be so pretty.

```
(map_w_inset <- ggdraw(basemap) +
  draw_plot(VI_Inset, x = 0.055, y = 0.251, width = 0.5, height = 0.3))
```



Now lets export this map as a .png file so we can use it in a report.

```
png(filename = here("", "Example_Map.png"), width = 7.5, height = 5.5,
    units = "in", pointsize = 15, res = 600)
```

```
(map_w_inset <- ggdraw(basemap) +
  draw_plot(VI_Inset, x = 0.055, y = 0.251, width = 0.5, height = 0.3))

dev.off()
```

```
## pdf
##   2
```