

# Map-making for ecologists, tutorial 2: Inset maps

Hannah K. Hoff

2022-04-14

The goal of this document is to provide an extension to our previous map-making tutorial. Here, we take a “finalized” map version from our last tutorial and create an inset map, which allows us to zoom in on areas of interest (sampling points, population ranges, etc.). This may be particularly useful for studies where sampling occurred in a very small area relative to the broader landscape, or in cases where there is substantial visual overlap between sample points. Using the function “geom\_rect” in ggplot2, we define a bounding box that can be used to show the sampling area in the context of the broader park, but also used to zoom in on the sampling area itself. Thanks to Dr. Elin Videvall for the inspiration to make an inset map tutorial!

As usual, we place an emphasis on packages such as ggplot2, sf, ggspatial, and tidyterra. These packages unlock great flexibility to customize map features using coding languages that are widely used and well-documented. Exploration of the Comprehensive R Archive Network (CRAN) will help users to explore the full capability of these packages beyond those reflected in this tutorial, and future map tutorials will build upon this framework to explore other ways to visualize spatial patterns in ecological data. Happy mapping!

Packages that may be generally useful:

```
library(knitr)
library(ggplot2)
library(dplyr)
library(rgdal)
library(raster)
library(ggsn)
library(rworldmap)
library(sf)
#library(gdalUtils)
library(gdalUtilities)
library(ggspatial)
library(rgeos)
library(sp)
library(ggspatial)
#remotes::install_github("dieghernan/tidyterra")
library(tidyterra)
library(RColorBrewer)
```

```

library(shades)
library(elevatr)
#install.packages('terra', repos='https://rspatial.r-universe.dev')
library(terra)
#install.packages('raster')
library(raster)
library(tidyverse)

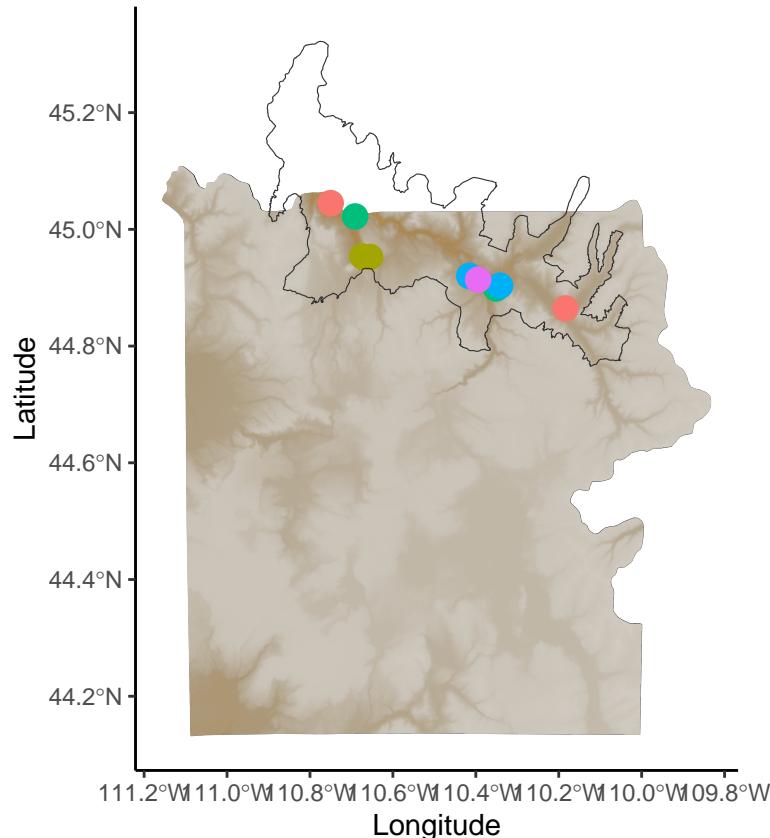
```

We start with a completed sample map. If you've been following along with previous tutorials, this is the final map version that we created! We will be tailoring our sample data to make it more relevant to this tutorial, and then updating the map before creating an inset.

```

fullmapwithelevation <- ggplot() + geom_sf(data = aoi_boundary_YNP_NAD83, lwd=0,
  color = "grey17", fill=NA) + geom_raster(data = YNPrasterdf, aes(x, y,
  fill = alt)) +geom_point(data = samplecoords3, size = 4, aes(x=x, y=y,
  color=SPECIES)) + xlab("Longitude") + ylab("Latitude") +
  scale_fill_hypso_tint_c(palette = "wiki-2.0_hypso",
  breaks = c(2000, 2500, 3000, 3500)) + labs(fill = "m") +
  geom_sf(fill = "transparent", color = "gray20", size = 1,
  data = northernrange_NAD83) + theme_classic() +
  theme(legend.position = "none")
fullmapwithelevation

```



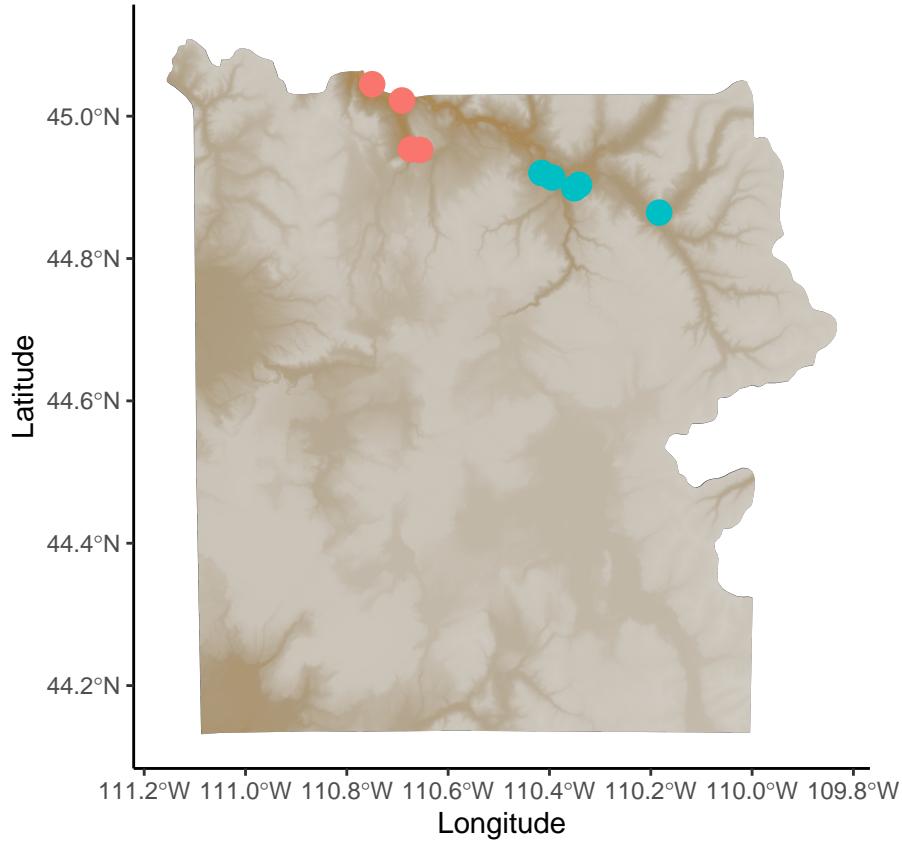
Let's inspect how I've tailored the dataset used in the last tutorial. I'll create a column called "population" and assign each point to population "A" or "B."

```
samples$population <- c("A", "B", "A", "A", "B", "A", "B", "B", "B", "B")  
head(samples)
```

```
##   SampleID SPECIES ANIMAL_ID EASTING NORTHING LOCATION COLLECTION.DATE SEASON  
## 1          S1 Species1      ID1  519692  4987974 Location1      10/15/22  FALL  
## 2          S2 Species1      ID2  564504  4968219 Location2      10/15/22  FALL  
## 3          S3 Species2      ID3  527202  4977748 Location3      10/30/22  FALL  
## 4          S4 Species2      ID4  525774  4977840 Location4      10/30/22  FALL  
## 5          S5 Species3      ID5  551247  4971898 Location5     11/15/22 WINTER  
## 6          S6 Species3      ID6  524345  4985447 Location6     11/15/22 WINTER  
##   population  
## 1          A  
## 2          B  
## 3          A  
## 4          A  
## 5          B  
## 6          A
```

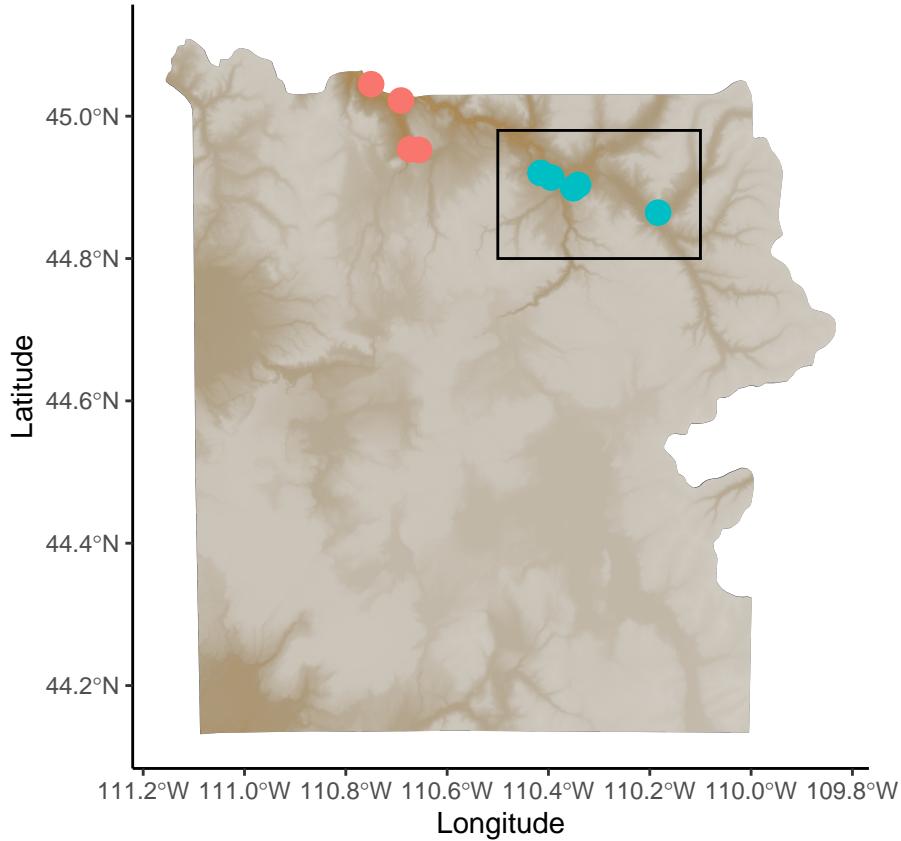
Now, we create a new map, with points colored by population. I will also exclude the outline of the northern range in this version. This will be the map that we feed into code used to create an inset! Note that the map version you have at this step with will be fed directly into the code to make the inset, so make sure you like this version before proceeding :)

```
fullmapwiththelevation <- ggplot() + geom_sf(data = aoi_boundary_YNP_NAD83, lwd=0,  
color = "grey17", fill=NA) + geom_raster(data = YNPrasterdf, aes(x, y,  
fill = alt)) +geom_point(data = samplecoords3, size = 4, aes(x=x, y=y,  
color=population)) + xlab("Longitude") + ylab("Latitude") +  
scale_fill_hypso_tint_c(palette = "wiki-2.0_hypso",  
breaks = c(2000, 2500, 3000, 3500)) + labs(fill = "m") + theme_classic() +  
theme(legend.position = "none")  
fullmapwiththelevation
```



Say I wanted to zoom in on Population B, which has quite a bit of overlap between points on the map. I'll use the “geom\_rect” function to define the area that I want to use for my bounding box.

```
fullmapwithbox <- ggplot() + geom_sf(data = aoi_boundary_YNP_NAD83, lwd=0,
  color = "grey17", fill=NA) + geom_raster(data = YNPrasterdf,
  aes(x, y, fill = alt)) +geom_point(data = samplecoords3, size = 4,
  aes(x=x, y=y, color=population)) +
  scale_color_manual(values = c("A" = "#F8766D", "B" = "#00BFC4")) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_hypso_tint_c(palette = "wiki-2.0_hypso",
  breaks = c(2000, 2500, 3000, 3500)) + labs(fill = "m") +
  theme(legend.position = "none") + geom_rect(aes(xmin = -110.1,
  xmax = -110.5, ymin = 44.98, ymax = 44.8), color = "black", fill = NA) +
  theme_classic() + theme(legend.position = "none")
fullmapwithbox
```



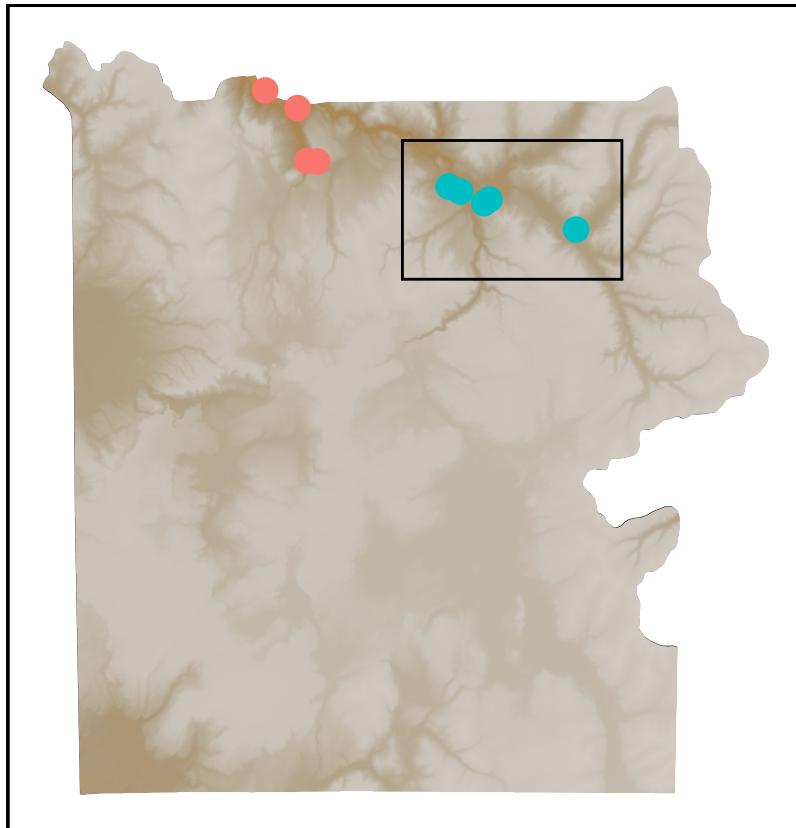
Since my main motivation for including the larger map of Yellowstone in this figure is just to provide some context for the sampling area in my zoomed in plot, and the larger map will be reduced in size and placed in the corner of the figure, I am choosing to remove the axes and coordinate values from it. To do that, we will use the theme() argument to remove each element. I also place a border around the plot itself using theme(panel.border) which will help separate this part of the figure from the main map when we combine the two.

```
fullmapwithbox2 <- ggplot() + geom_sf(data = aoi_boundary_YNP_NAD83, lwd=0,
  color = "grey17", fill=NA) + geom_raster(data = YNPrasterdf,
  aes(x, y, fill = alt)) +geom_point(data = samplecoords3, size = 4,
  aes(x=x, y=y, color=population)) +
  scale_color_manual(values = c("A" = "#F8766D", "B" = "#00BFC4")) +
  xlab("Longitude") + ylab("Latitude") +
  scale_fill_hypso_tint_c(palette = "wiki-2.0_hypso",
  breaks = c(2000, 2500, 3000, 3500)) + labs(fill = "m") +
  theme(legend.position = "none") + geom_rect(aes(xmin = -110.1,
  xmax = -110.5, ymin = 44.98, ymax = 44.8), color = "black", fill = NA) +
  theme(axis.line=element_blank(),axis.text.x=element_blank(),
  axis.text.y=element_blank(),axis.ticks=element_blank(),
  axis.title.x=element_blank(),
  axis.title.y=element_blank(),legend.position="none",
  panel.background=element_blank(),panel.border=element_blank(),
```

```

panel.grid.major=element_blank(), panel.grid.minor=element_blank(),
plot.background=element_blank()) +
theme(panel.background = element_rect(fill = "white")) +
theme(panel.border = element_rect(color = "black",fill = NA,
linewidth = 1))
fullmapwithbox2

```



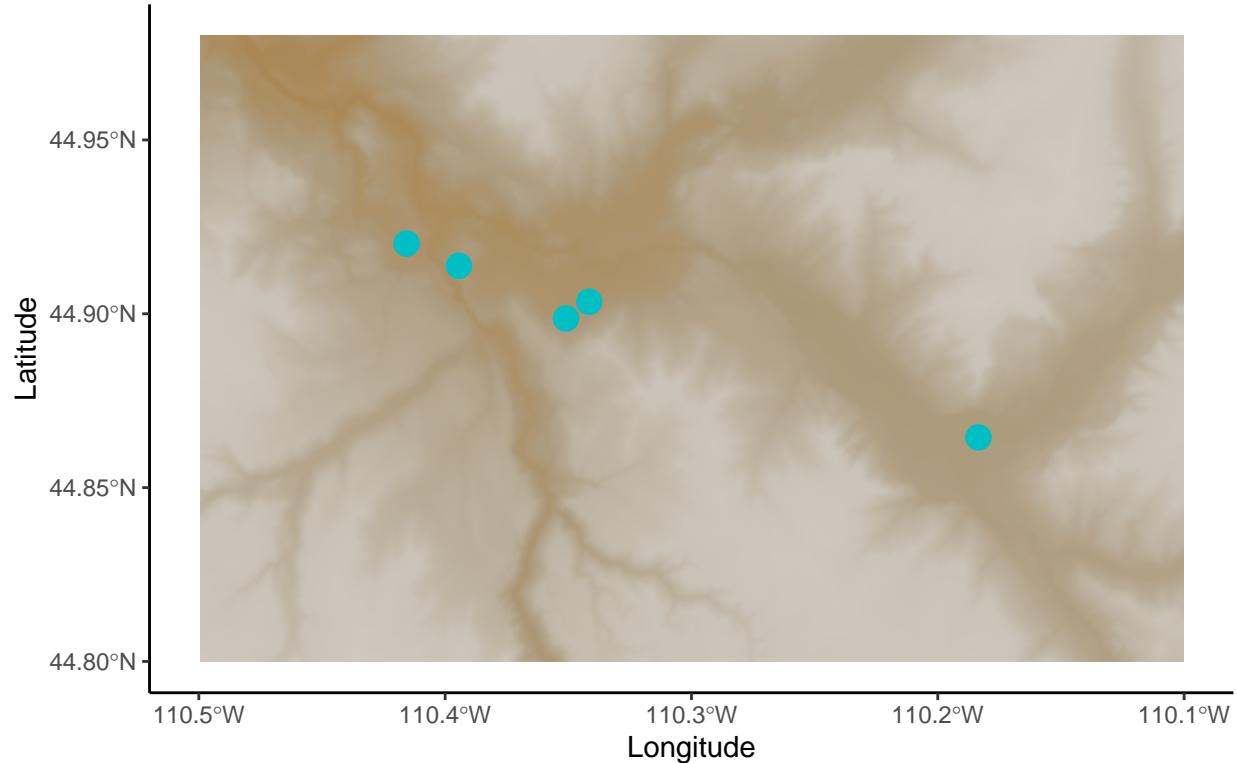
Next, I will create a map of only the area included in that box, using `xlim()` and `ylim()` to confine the map area. Make sure that the same coordinates are used here as in the box in the above map.

```

populationBmap <- ggplot() + geom_sf(data = aoi_boundary_YNP_NAD83, lwd=0,
  color = "grey17", fill=NA) + geom_raster(data = YNPrasterdf, aes(x, y,
  fill = alt)) +geom_point(data = samplecoords3, size = 4, aes(x=x, y=y,
  color=population)) +
scale_color_manual(values = c("A" = "#F8766D", "B" = "#00BFC4")) +
xlab("Longitude") + ylab("Latitude") + scale_fill_hypso_tint_c(palette =
"wiki-2.0_hypso", breaks = c(2000, 2500, 3000, 3500)) + labs(fill = "m")+
xlim(c(-110.5, -110.1)) + ylim(c(44.8, 44.98)) +
theme(panel.border = element_rect(color = "black",fill = NA,
linewidth = 1)) + theme_classic() +
theme(legend.position = "none")
populationBmap

```

```
## Warning: Removed 2625947 rows containing missing values ('geom_raster()').  
## Warning: Removed 4 rows containing missing values ('geom_point()').
```

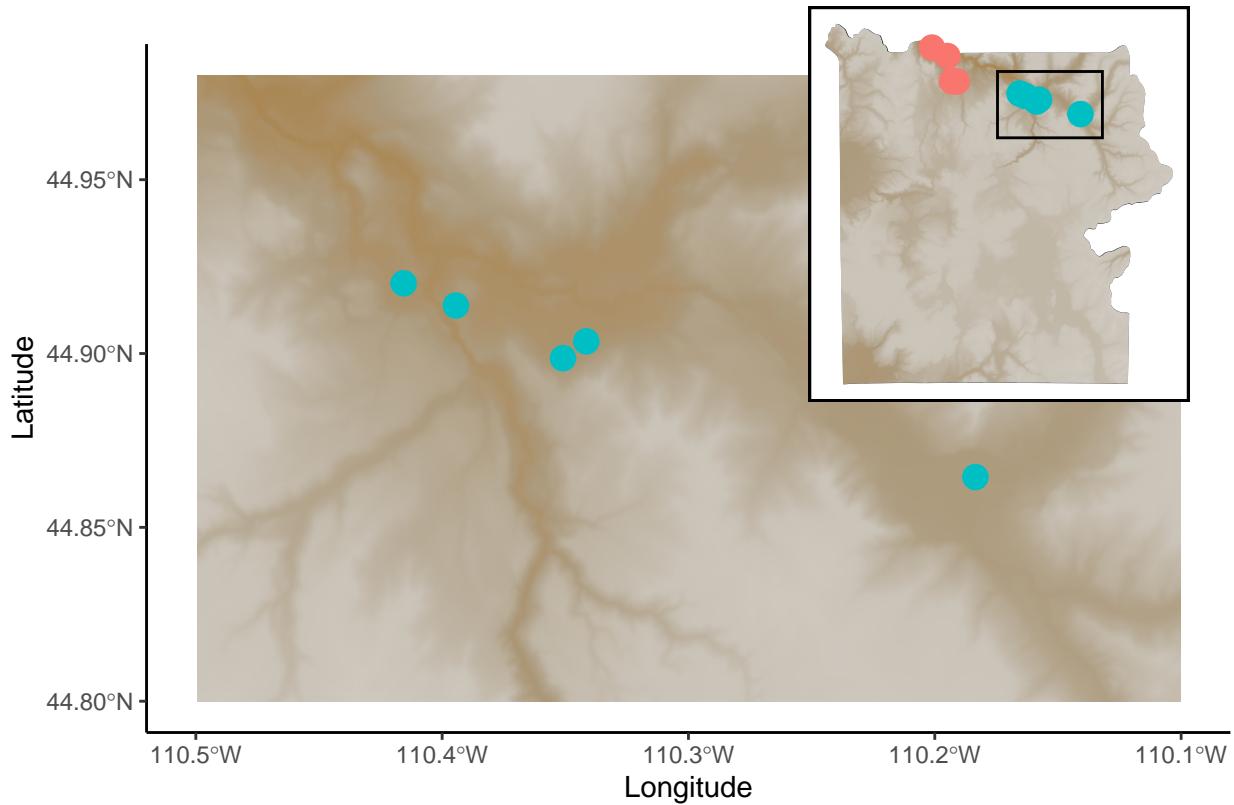


We now have the basic components (full map of Yellowstone plus the specific area we want to zoom in on) of our future inset map. Next, we will combine the two plots and make adjustments to the placement of the full park map so that it fits into a corner and doesn't overlay any of our sample points. We will do this by using the `print()` function and defining the viewport (what are the x and y coordinates we want the smaller map to sit in?) and the size (width and height). We first call the map of our zoomed in sampling area, and then print the smaller map on top of it. Make sure to run these two lines of code together!

```
populationBmap
```

```
## Warning: Removed 2625947 rows containing missing values ('geom_raster()').  
## Warning: Removed 4 rows containing missing values ('geom_point()').
```

```
print(fullmapwithbox2, vp = viewport(0.8, 0.75, width = 0.5, height = 0.5))
```



We now have a figure that includes 1) a full map of Yellowstone National Park  
2) a box around our sampling area that matches 3) the larger map featuring our  
zoomed-in sampling area (and sample points)!