

Section 3: Mapping Occurrence Data with leaflet

Maxine Cruz

04 April 2024

Here we will demonstrate how to plot species occurrence data on a leaflet map.

Learning Objectives

1. Be able to plot species occurrence data from GBIF.
 2. Be able to plot occurrence data using **leaflet**.
 3. Understand the value of knowing where a species has been observed.
 4. Understand and practice good file management.
-

Different Ways of Presenting Data

Sometimes different visualizations may be more suitable for the situation. A **ggplot2** map would be best for static maps that would not require someone to zoom in / out for more information — which is perfectly fine in our case, but **ggplot2** still has its limitations. We may want to have a topographical map as the base, or a map that shows roads and state / country names. Such features might help visualizing what type of landscape and where an organism has been observed at the most. Attempting to add this kind of base to a **ggplot2** map might pose some challenges and a less-than-ideal map (at least from the amount of searching how-to's I have done), and an easy way to add these features would be through **leaflet**.

The **leaflet** package operates in a similar fashion to **ggplot2**, but uses a pipe (**%>%**) instead of an addition sign (**+**) in layer additions. The functions used here will also be different. Dynamic and interactive maps might be preferred as part of a website or document where the user can do more than view occurrences. For example, there is an option to have each occurrence point on a leaflet map hold information (accessible by clicking on that point). This feature might provide a quick way to view coordinates, the elevation, or other information that would otherwise be found in the data table. Offering the ability for the user to zoom in to areas would be helpful in planning field research to look for that species, or for more large-scale conservation tasks such as advising where construction might be best avoided to preserve habitat.

Necessary packages

This section of the tutorial requires the following R packages to assist in our mapping:

- **leaflet**: Creates dynamic and interactive maps. Leaflet is actually an open-source JavaScript library, and this R package helps to integrate and control these maps through R.

- **mapview:** Allows for interactive viewing of spatial data. This package also offers a way to save a leaflet map as a .png file, which will be of interest to us later.
- **webshot2:** Required to save the leaflet map through **mapview**.

We can install and use them by running the following lines:

```
# Install packages (Note: This only needs to be done once!)
install.packages("leaflet")
install.packages("mapview")
install.packages("webshot2")
```

Afterwards, let us read these libraries into R so we can use them in the plotting process:

```
# Load dependencies
library(leaflet)
library(mapview)
library(webshot2)
```

Great! Now we have the tools to move forward in this section.

Mapping with leaflet

This package has stack-able components as well, but are called slightly different things than those from **ggplot2**. Like any customizable object, the choices are endless. Here we will focus on the three main pieces of this map:

- **Map tiles:** The default base map that can be added is provided by OpenStreetMap, and is called using **addTiles()**. The map that results can be highly detailed the more you zoom in and includes features such as road maps, landmarks, and city names. Another option for the base map is using one from a third-party provider... of which there are many options (you can find the link to all the options in the Additional Resources section of this lesson). Some of these are variations on OpenStreetMap and others are from groups such as Stadia, Esri, and the United States Geological Survey (USGS). Proceeding with the third-party map tiles would use the function **addProvidertiles()** in place of **addTiles()**.
- **Markers:** These would be the equivalent of **geom_point()** in **ggplot2**. The data can be displayed as circles or icons on the map.
- **Legend:** This is not a necessity, but we will be using the legend box as an area for a simple title on our map. In later sections, the legend will be an important feature because it will hold the key for interpreting our species' environmental suitability values on the map. For now, we will be introducing it for adding a title and aesthetic purposes.

i) Load Our Data

Let us bring in the data (same as before) and remind ourselves of the contents:

```
data <- read.csv("data/cleaned_data.csv")

# Data currently has: 389 observations, 10 variables
```

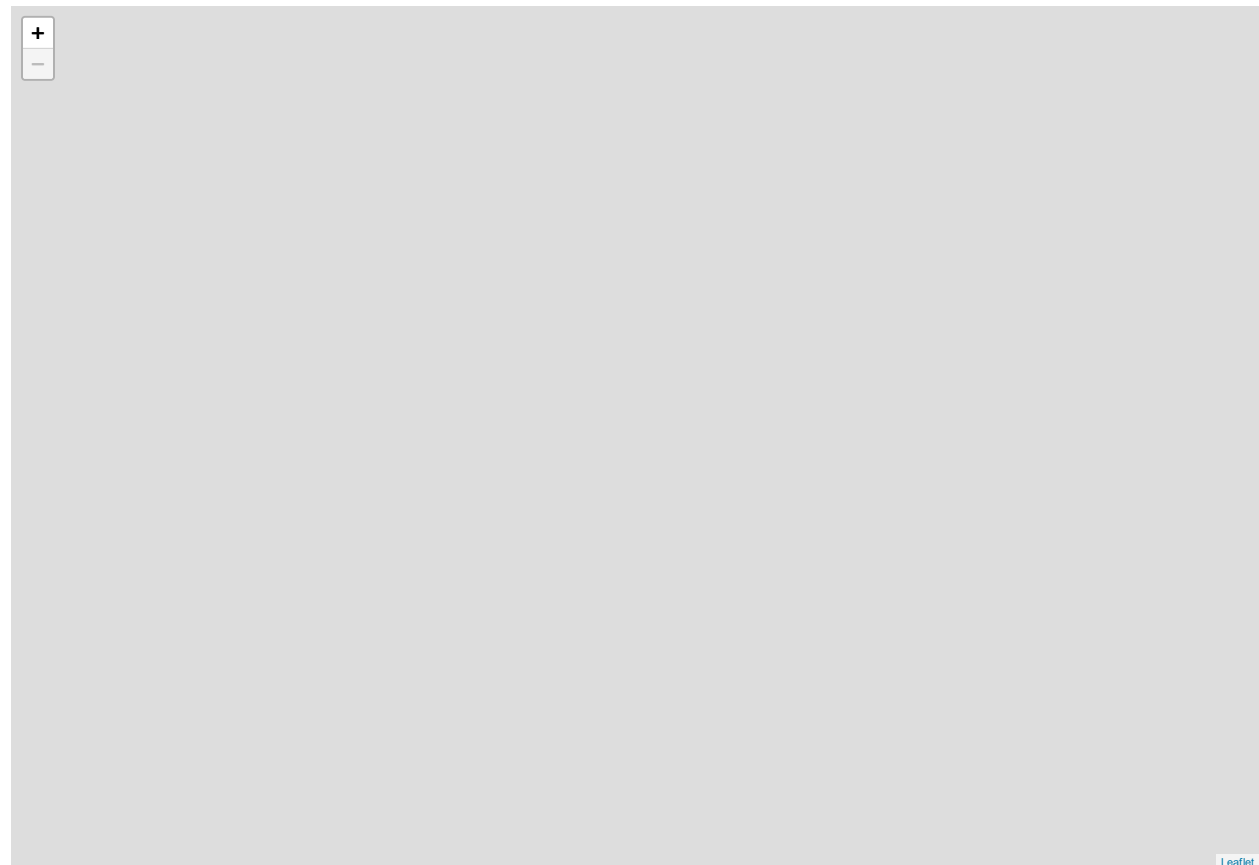
Here are the first few rows of our data (some columns have been omitted for the sake of fitting it in):

```
##   year month day  latitude longitude      species speciesKey  kingdom
## 1 1900    11  NA -29.85417   31.0500 Egybolis vaillantina    1782645 Animalia
## 2 1929    10  NA  -3.21490   40.1218 Egybolis vaillantina    1782645 Animalia
## 3 1930    11   1  -3.21490   40.1218 Egybolis vaillantina    1782645 Animalia
## 4 1930    12   7  -3.21490   40.1218 Egybolis vaillantina    1782645 Animalia
## 5 1936     9  NA  -3.21490   40.1218 Egybolis vaillantina    1782645 Animalia
```

ii) The `leaflet()` Function by Itself

Similar to plots using `ggplot()`, these plots have a base function called `leaflet()`. What happens when you run that by itself is this — note that you will get a blank plot with a zoom control:

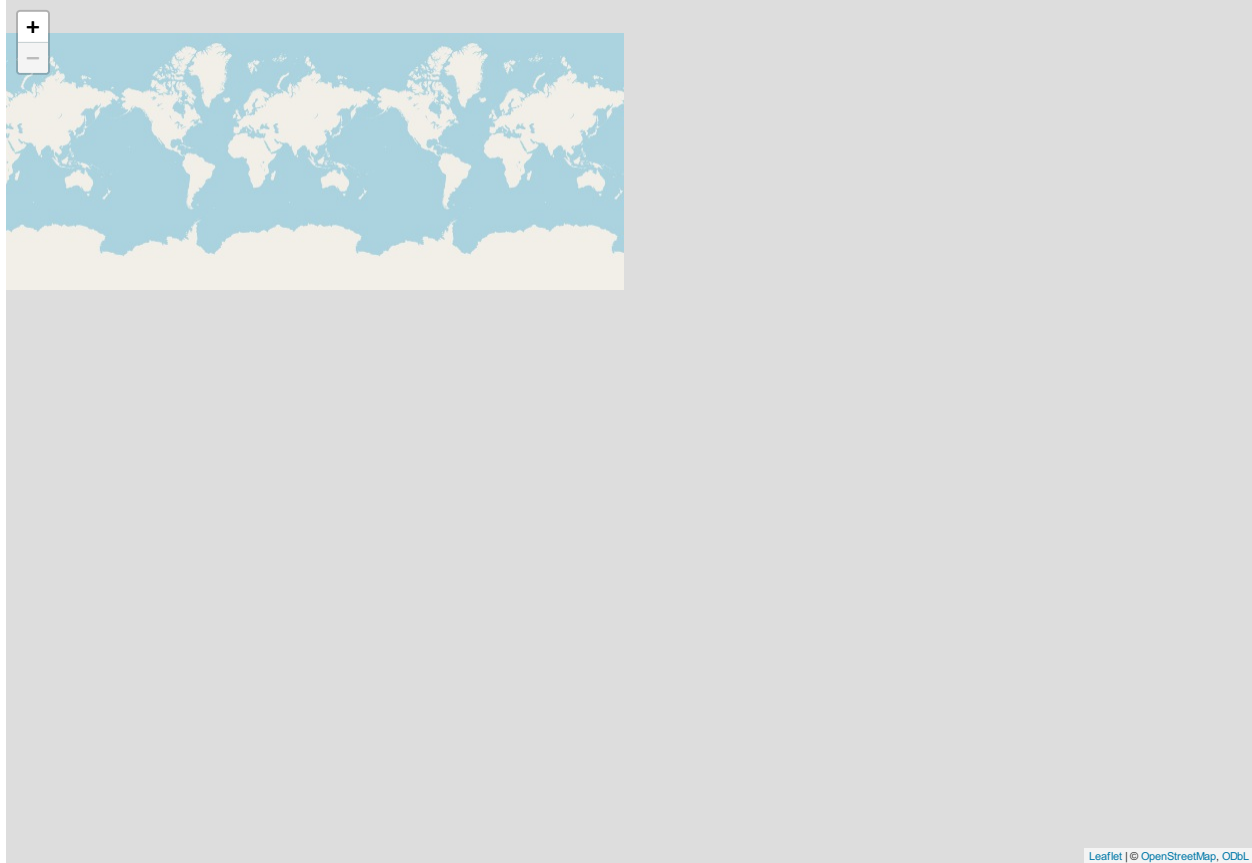
```
# Base function
leaflet()
```



iii) Adding Map Tiles

As said before, there are many options for the map tiles. Feel free to explore those but for the sake of simplicity we will demonstrate the map tile addition using the default OpenStreetMap option:

```
# Base function
leaflet() %>%
  # Add map tiles
  addTiles()
```

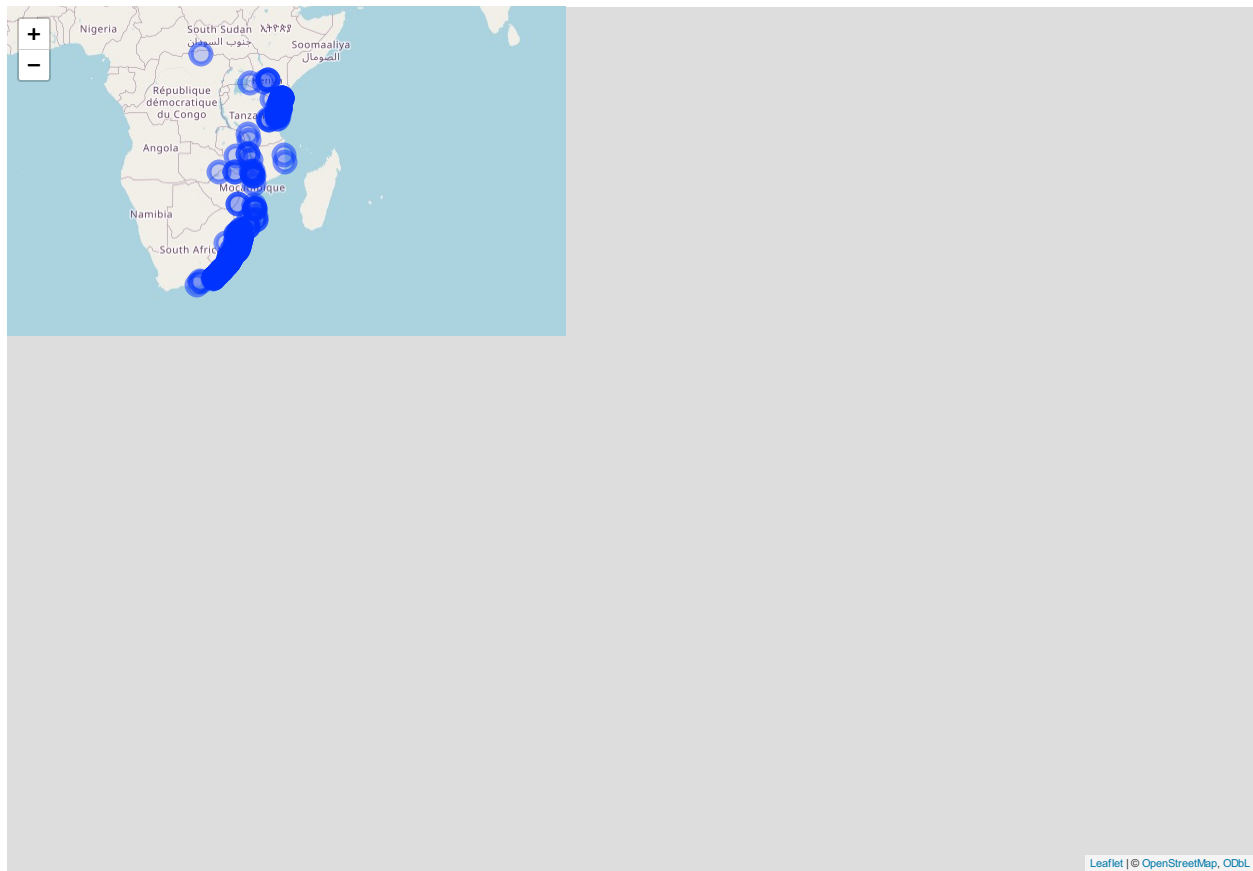


Notice that the default zoom without any other additions to the map shows the full globe. It is possible to adjust the initial zoom that renders but that is not something we will cover here. At this point, it is possible to interact with the map above — to do so, use the zoom control at the top left of the plot to adjust the zoom, and drag the map in a direction to adjust what section of the globe you are viewing.

iv) Adding Markers

Adding the occurrence points to this map can be done by either (1) calling our data within `leaflet()`, or (2) through `addCircleMarkers()`, which is the equivalent to `geom_point()` and also requires a squiggle (“~”) to be used with the latitude and longitude variable names. Here we will practice the addition through the latter:

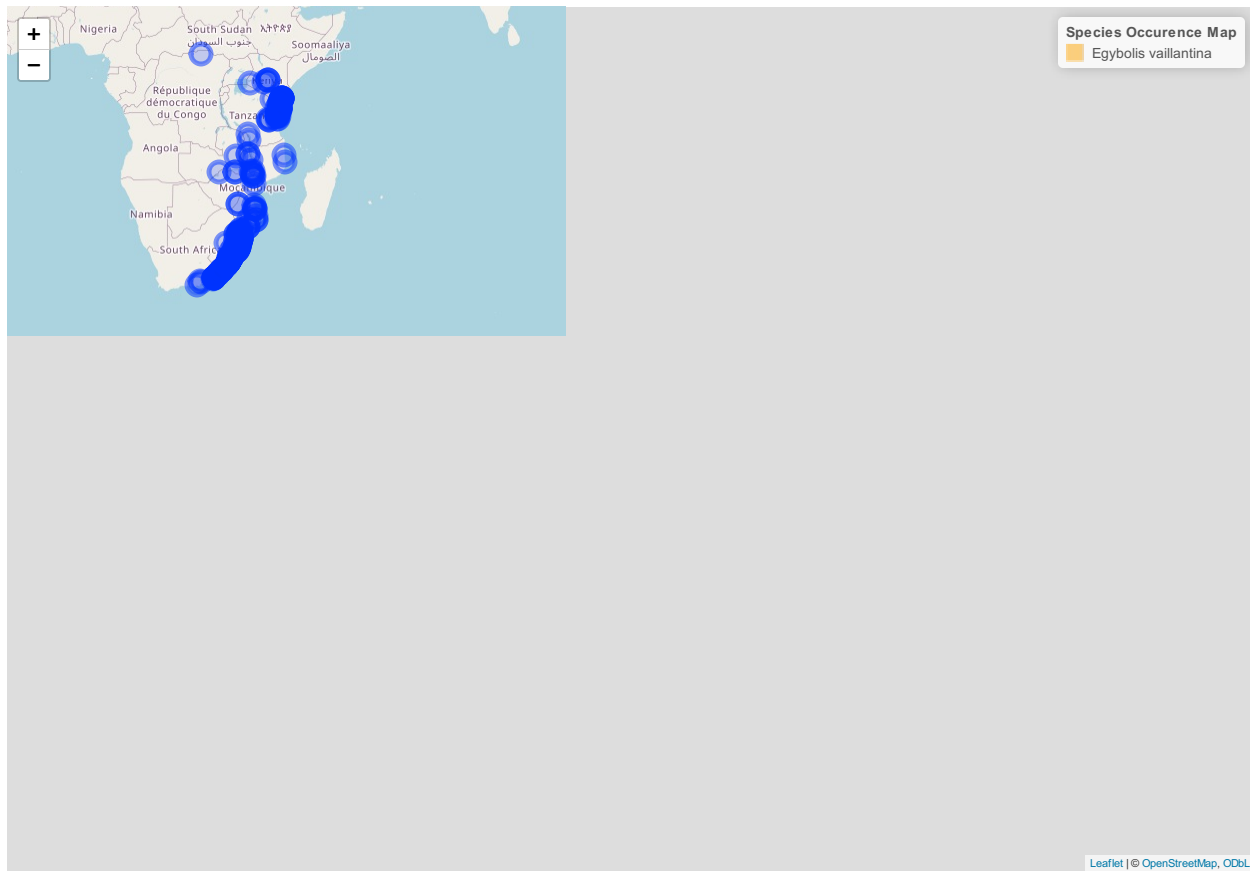
```
# Base function
leaflet() %>%
  # Add map tiles
  addTiles() %>%
  # Add occurrence points
  addCircleMarkers(data = data,
    lat = ~latitude,
    lng = ~longitude)
```



v) Adding a Title

Adding titles to `leaflet` maps can be done in tandem to adding a legend... for this specific case. Since the function we will be using *is* a legend by design, you will be prompted to choose a color for the species we are “labeling”. I will choose orange as my color today, and you will notice that this color will not match the color of the occurrence points. It would seem as though `leaflet` separates the legend from previous plot arguments, which differs from `ggplot()` as the latter tends to derive a legend from the data. For now, we will proceed with the color differences and attend to it later:

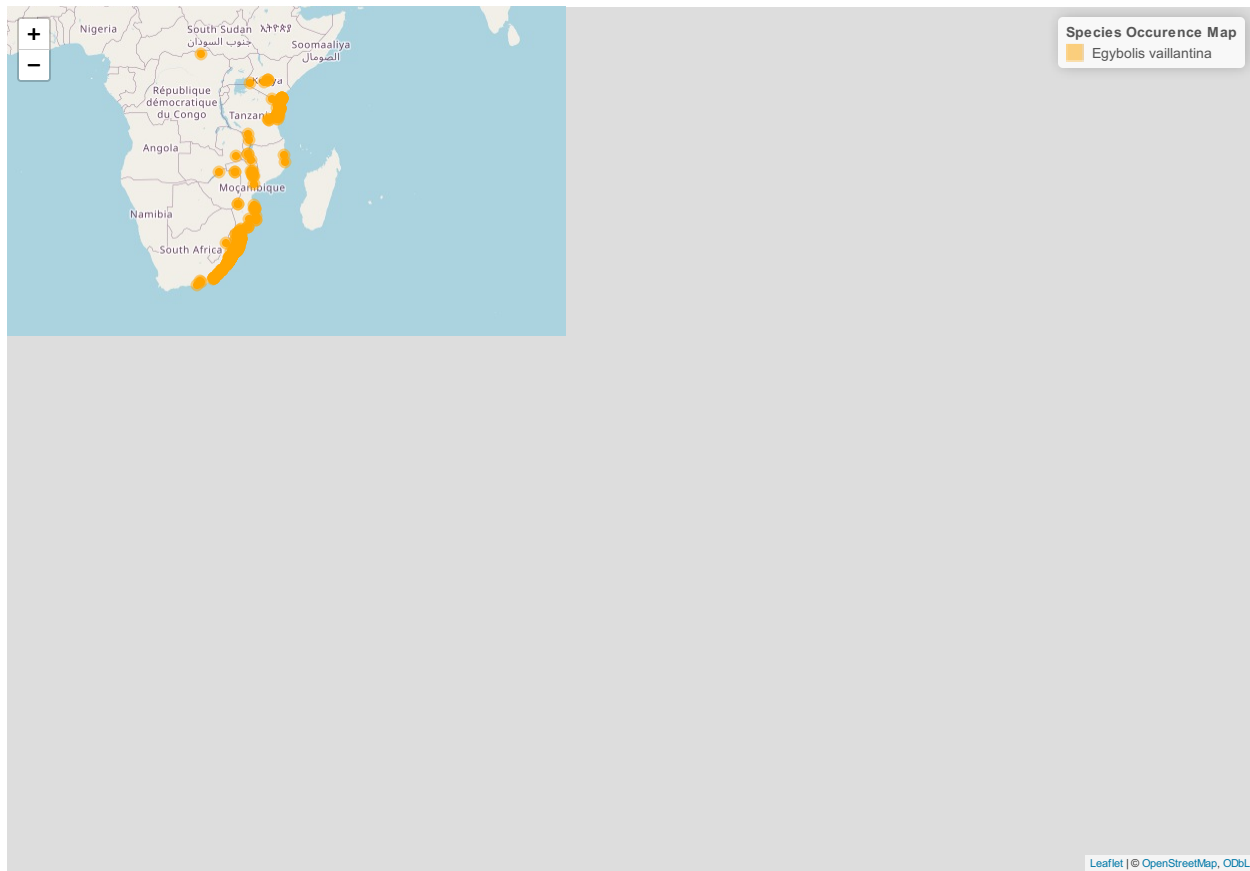
```
# Base function
leaflet() %>%
  # Add map tiles
  addTiles() %>%
  # Add occurrence points
  addCircleMarkers(data = data,
                   lat = ~latitude,
                   lng = ~longitude) %>%
  # Add a legend (which is a title hack, in this case)
  addLegend(position = "topright",
            title = "Species Occurrence Map",
            labels = "Erythrina vaillantiana",
            colors = "orange")
```



vi) Making it Look Pretty

Now that we have a good foundation of a map once again, we can finally direct our attention to the aesthetics. Unlike the `ggplot()` beautification, we will only be changing the color, circle size, and opacity of the markers since this map already looks rather nice. This is done by filling in a few arguments under `addCircleMarkers()`:

```
# Base function
leaflet() %>%
  # Add map tiles
  addTiles() %>%
  # Add occurrence points
  addCircleMarkers(data = data,
                    lat = ~latitude,
                    lng = ~longitude,
                    color = "orange",
                    radius = 4,
                    fillOpacity = 1) %>%
  # Add a legend (which is a title hack, in this case)
  addLegend(position = "topright",
            title = "Species Occurrence Map",
            labels = "Egybolis vaillantina",
            colors = "orange")
```



An additional step that might be taken is removing the zoom control prior to saving. This can be done by modifying `leaflet()` to include the following:

```
leaflet(options = leafletOptions(zoomControl = FALSE))
```

vii) Saving the Map

Lovely! We did a lot of work to build that map, though perhaps it took less steps compared to `ggplot2`... nonetheless, it will now be saved to our output folder alongside the aforementioned version.

```
# Store our map in an object
map <- leaflet(options = leafletOptions(zoomControl = FALSE)) %>%
  addTiles() %>%
  addCircleMarkers(data = data,
    lat = ~latitude,
    lng = ~longitude,
    color = "orange",
    radius = 3,
    fillOpacity = 0.8) %>%
  addLegend(position = "topright",
    title = "Species Occurrence Map",
    labels = "Egybolis vaillantina",
    colors = "orange")

# Save our map
mapshot2(map, file = "output/egybolis_occurrence_map_leaflet.png")
```

Nice job! To conclude — here we introduced a different method of mapping and presenting our species' data. Each method has its pros and cons, and sometimes it boils down to how much time you are willing to put in. That statement really only applies if you were to figure out how to code this yourself, which may be the case if you wanted to add features that were not covered in this tutorial.

Additional Resources

- Want a different base map? There is a leaflet provider testing site so you can check out what each one looks like: [Leaflet Provider Demo](#)
 - Vignette of leaflet: [Introduction to leaflet](#)
 - An R Charts guide to leaflet: [Interactive maps with leaflet in R](#)
-

Next Section: [\[link to next lesson\]](#)

For questions, comments, or concerns (or even tattoo design requests) please email tmcruz@arizona.edu.