Spatial visualizations really come to life when you have a real map as a background. In R, ggmap is the package that you’ll want to use to get these maps. In what follows, we’ll demonstrate how to use ggmap with the Sacramento dataset in the caret package. For each city, we are going to keep track of the median price of houses in that city, the number of transactions in that city, as well as the mean latitude and longitude of the transactions.

library(caret)

library(dplyr)

library(ggmap)

data(Sacramento)

df <- Sacramento %>% group\_by(city) %>%

summarize(median\_price = median(price), transactions = n(),

latitude = mean(latitude), longitude = mean(longitude))

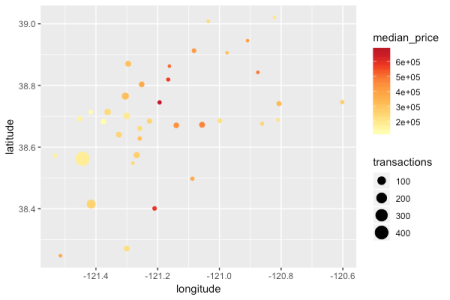
Now, I want to plot these cities on a map, with the color of the points representing the median price and the size representing the number of transactions. I could do this with ggplot2:

ggplot() +

geom\_point(data = df, mapping = aes(x = longitude, y = latitude,

col = median\_price, size = transactions)) +

scale\_color\_distiller(palette = "YlOrRd", direction = 1)



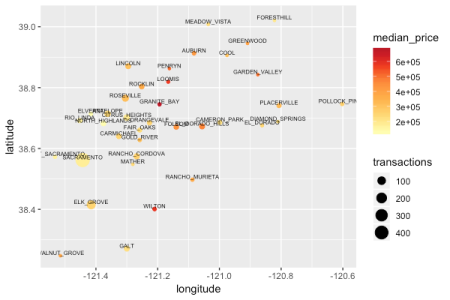
That plots the data that we want, but the visualization isn’t very informative. For it to be useful, I need to know which point represents which city. I could add text labels to each of these points…

ggplot(data = df, mapping = aes(x = longitude, y = latitude)) +

geom\_point(aes(col = median\_price, size = transactions)) +

geom\_text(aes(label = city), size = 2, nudge\_y = 0.01) +

scale\_color\_distiller(palette = "YlOrRd", direction = 1)



… but it would clutter the visualization, and I still think that it does a poorer job of being informative than putting a map as background. Let’s use ggmap to do that!

**Google maps**

A quick note about Google maps: it seems that using Google maps as a background is quite a challenge now. According to the author of the ggmap package, to do so you need to provide an API key and enable billing.

**Stamen maps**

Fortunately there is an easier way to get map backgrounds: Stamen maps. In the code below, I’m getting a base map of the Sacramento region based on the latitudes and longitudes I see in my dataset:

height <- max(Sacramento$latitude) - min(Sacramento$latitude)

width <- max(Sacramento$longitude) - min(Sacramento$longitude)

sac\_borders <- c(bottom = min(Sacramento$latitude) - 0.1 \* height,

top = max(Sacramento$latitude) + 0.1 \* height,

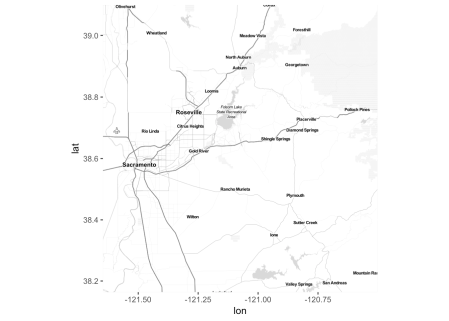
left = min(Sacramento$longitude) - 0.1 \* width,

right = max(Sacramento$longitude) + 0.1 \* width)

map <- get\_stamenmap(sac\_borders, zoom = 10, maptype = "toner-lite")

I can then print it with the ggmap function:

ggmap(map)



The zoom parameter is set to 10 by default. Making it bigger gives a more detailed map, but takes longer to load.

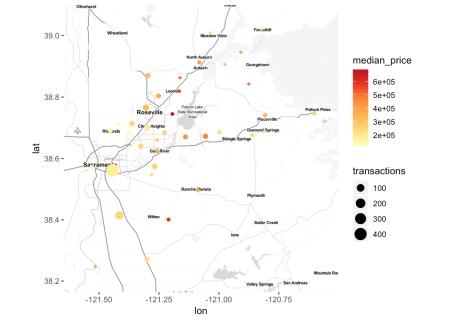
Just like in ggplot2 graphics, we can add layers to the map drawn by ggmap(map) with the + syntax. For example, if we wanted to overlay the map with the data from our dataset, we could do the following:

ggmap(map) +

geom\_point(data = df, mapping = aes(x = longitude, y = latitude,

col = median\_price, size = transactions)) +

scale\_color\_distiller(palette = "YlOrRd", direction = 1)

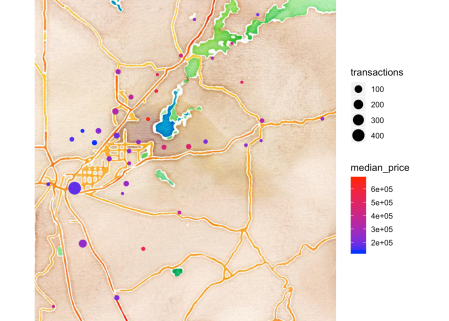


Another way to get Stamen maps as a background is to use the qmplot function. Notice though that the syntax is pretty different from what we had with ggplot. Here is a map plotting the same points as above but with a different color scale and a different type of background map:

qmplot(x = longitude, y = latitude, data = df, maptype = "watercolor",

geom = "point", color = median\_price, size = transactions) +

scale\_color\_gradient(low = "blue", high = "red")



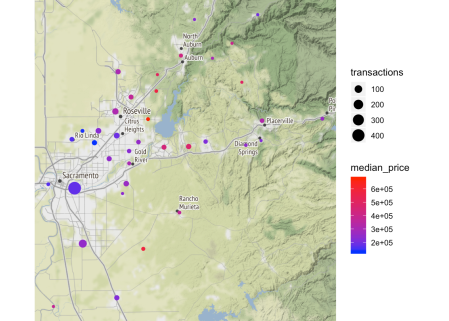
The full list of Stamen map types is “terrain”, “terrain-background”, “terrain-labels”, “terrain-lines”, “toner”, “toner-2010”, “toner-2011”, “toner-background”, “toner-hybrid”, “toner-labels”, “toner-lines”, “toner-lite”, “watercolor”. Note that some of them don’t work with the CRAN version of ggmap (e.g. “terrain”); instead, you will have to download the github version of the package.

Below are examples of what “terrain” and “toner” look like:

qmplot(x = longitude, y = latitude, data = df, maptype = "terrain",

geom = "point", color = median\_price, size = transactions) +

scale\_color\_gradient(low = "blue", high = "red")



qmplot(x = longitude, y = latitude, data = df, maptype = "toner",

geom = "point", color = median\_price, size = transactions) +

scale\_color\_gradient(low = "blue", high = "red")

