Assignment 5 - Interactive Visualizations

Veda Nayak

2025-06-12

Dataset Cleanup

I started by confirming that all of the datapoints were within California, and I found that 3 were not. Since California's latitudinal and longitudinal bounderies are 32° 30' N to 42° N and 114° 8' W to 124° 24' W, I took 32° - 43° N as the latitudinal bounds and 113°- 125° W as the longitudinal bounds.

```
pd = readRDS("PropertyData.rds")

table(pd$lat < 31)
```

```
##
## FALSE  TRUE
##  6656     3
```

```
table(pd$long > -113)
```

```
##
## FALSE  TRUE
##  6656     3
```

```
table(pd$lat < 31 & pd$long > -113)
```

```
##
## FALSE  TRUE
##  6656     3
```

```
pd[which(pd$lat < 31 & pd$long > -113),]
```

```
##                    price bedrooms size
## Napa.price132     999000       NA   NA
## Napa.price414    2900000        6 6494
## Sonoma.price4030 3795000        3 2170
##
## Napa.price132            https://www.realtor.com/realestateandhomes-detail/3027-Foothill-Blvd_Calist
## Napa.price414    https://www.realtor.com/realestateandhomes-detail/4040-Spring-Mountain-Rd_Saint-Hel
## Sonoma.price4030            https://www.realtor.com/realestateandhomes-detail/212-Tucker-St_Healdsb
##                                  type           streetAddress addressLocality
## Napa.price132    SingleFamilyResidence      3027 Foothill Blvd       Calistoga
```

```
## Napa.price414    SingleFamilyResidence 4040 Spring Mountain Rd    Saint Helena
## Sonoma.price4030 SingleFamilyResidence           212 Tucker St      Healdsburg
##                  addressRegion postalCode baths beds lot-size sqft days garage
## Napa.price132               CA      94515    NA   NA   475675   NA  101     NA
## Napa.price414               CA      94574     4    6  1663120 6494   68     NA
## Sonoma.price4030            CA      95448     2    3     2047 2170   33      2
##                        type.1 yearBuilt         id lat long price.1 lastSold
## Napa.price132            Land        NA 2978583468   0    0  999000       NA
## Napa.price414    Single family      1977 2979805670   0    0 2900000   380000
## Sonoma.price4030 Single family      2021 2981168989   0    0 3795000  1325100
##                  lastSoldDate county
## Napa.price132             <NA>   Napa
## Napa.price414       1986-10-31   Napa
## Sonoma.price4030    2018-09-19 Sonoma
```

I removed these datapoints from the dataset since I did not want to doctor the latitude and longitude values and the 3 addresses out of the almost 13,000 seemed like a negligable loss.

I then wanted to see if there were any price discrepencies between the two price variables.

```
table(pd$price != pd$price.1)
```

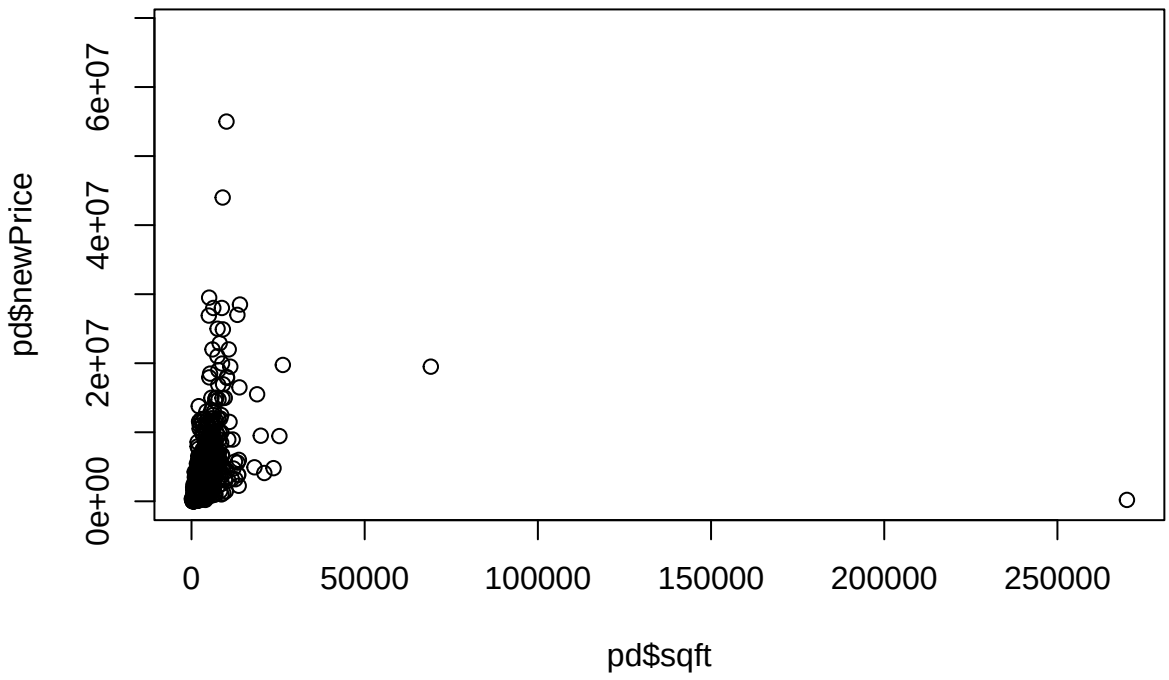
```
##
## FALSE  TRUE
##  6948    27
```

27 non-NA and non-zero prices did not match up with one and another. I took the average of the two prices to be the market price.

```
pd$newPrice = pd$price
pd[weirdPrices, 25] <- sapply(weirdPrices, function(x) mean(pd[x, 1], pd[x, 21]))
```

I then did a basic plot of the data to identify outliers.

```
plot(x = pd$sqft, y = pd$newPrice)
```



I removed the square footage outliers.

```
pd[which(pd$sqft == 100), ] = NA
```

I removed the residance with 256 bedrooms (it was an apartment complex) as well as listings that were farms, coops, land, or multifamily units (apartment complexes and not single apartments) since my interest was in single family homes (an applicable topic for an undergraduate student). I didn't remove the hosues with a large number of bedrooms directly beecause I saw that some of them were actually large houses that sold for a lot of money.

```
max(pd[!is.na(pd[,2]), 2])
```

```
## [1] 256
```

```
m1 = which(pd[,2] == max(pd[!is.na(pd[,2]), 2]))
pd[m1, 2] <- NA

table(pd$type.1)
```

```
##
##          Coop          Farm          Land    Mfd/Mobile        Mobile
##             2             5           902           623             1
##  Multi-Family Single family Single Family     Townhomes     Townhouse
##           238          5029            32           138             2
```

```
weirdLocations = which(pd$type.1 == 'Coop' | pd$type.1 == 'Farm'| pd$type.1 == 'Land' |pd$type.1 ==  'Mu
pd[weirdLocations, ] <- NA
```

I was then ready to start preparing my data for visualization.

Data Visualization

Setup

Dataframe Preparation

I first added two columns to my dataframe: an html-readable link and a row ID. The readable link is to allow users to click on a hyperlink for the viewing and the row ID is to act as a key for the shared dataframe. I also reordered the columns to be in a more readable format for users.

```
pd$url_link <- paste0('<a href="', pd$url, '" target="_blank">View Listing</a>')
pd_new <- pd[, c(1:3, 26, 5:25)]
pd_new$row_id <- 1:nrow(pd_new)
pdShared <- pd_new[!is.na(pd_new$county), c(25, 13, 11, 10, 15, 22, 23, 6, 7, 24, 17, 4, 26, 19, 20)]
```

I then used crosstalk::SharedData$new to create the shared datatable using the row_id as a key to make my widgets compatable with each other.

I initially used this method, and not giraf, because I was going to make my map using leaflet. However, when I tried this, I couldn't get the linking to work, so I switched to using plotly. I had already started working with crosstalk so I decided to keep my original formatting.

Styling

I then created a custom color palette for consistency across my plots.

```
colorPalette = c("Marin" = "#ff595e",
                 "Napa" = "#ff924c",
                 "Sacramento" = "#ffca3a",
                 "Solano"= "#8ac926",
                 "Sonoma" = "#EDFF7A",
                 "Sutter"= "#1982c4",
                 "Yolo" = "#6a4c93",
                 "Yuba" = "#f78ef0"
               )
```

I also made a pastel version of this color palette so that I could color the datatable by the county without it being too busy. I did this so that if you were comparing residences across counties you could more easily see which residence is which.

```
pastelPalette = c("Marin" = "#ffe6e7",
                  "Napa" = "#fff0e6",
                  "Sacramento" = "#fff8e6",
                  "Solano"= "#f0f9e6",
                  "Sonoma" = "#fcffe6",
```

```
                    "Sutter"= "#e6f2fc",
                    "Yolo" = "#f0e6f5",
                    "Yuba" = "#fef0fe")
```

When I initially set up the html visualization, I noticed some words were getting cut off on the side. Once I opened up the developer tools, I realized that it was something going on in the padding and margins. I modified the styling to ensure everything could fit on one page. I also centered the heading, and decreased the line height for my text elements since everything was looking a bit spaced out.

```
css = tags$head(
  tags$style(HTML("
    body {
      margin: 0;
      padding: 20px;
      overflow-x: auto;
    }
    h1 {
      text-align: center;
      line-height: 0.6;
    }
    h2 {
      text-align: center;
      line-height: 0.6;
    }
    p{
      line-height: 0.3;
      }
  "))
)
```

Scatter Plot: Square Fottage by Price

I colored the points based on county to better connect it to the map. I didn't know how to add the hyperlink into the tooltip without the tooltip disappearing upon touch, so I added the hyperlink into the connected data table. In the tooltips, I added the number of beds, baths, and the year the property was built for a quick reference. I made the points a little less opaque so that you could see the overlapping points more clearly.

```
scatter = ggplot(sd, aes(x = sqft,
                         y = newPrice,
                         color = county,
                         texst = paste("Beds:", beds,
                                       "<br>Baths:", baths,
                                       "<br>Year Built:", yearBuilt))) +
  geom_point(size = 3, alpha = 0.9) +
  labs(x = "Square Footage",
       y = "Price",
       color = "County") +
  scale_color_manual(values = colorPalette)
  theme_bw()
```

To make it interactive, I used plotly::ggplotly.

```
scatter_interact = ggplotly(scatter, tooltip = "text")
```

Map

As I mentioned earlier, I used plot_ly to generate the map. I colored by county again for an easier visualization. I added the price, number of beds and baths, and square footage into the tooltips for quick referenece.

I also changed the style to 'open-street-map' to somewhat mimic leaflet's style of map.

I realized that all of the residences were in the same general area, so I centered on around the middle of the longitude and latitude values to capture most of the data at once.

I also highlighted the selected point in black to make it easier to spot.

```
plotlyMap = plot_ly(sd,
                    x = ~long,
                    y = ~lat,
                    color = ~county,
                    colors = colorPalette,
                    text = ~paste("Price: $", format(newPrice, big.mark = ","),
                                  "<br>Beds:", beds,
                                  "<br>Baths:", baths,
                                  "<br>Sq Ft:", format(sqft, big.mark = ",")),
                    type = "scattermapbox",
                    marker = list(size = 8)) %>%
  layout(
    mapbox = list(
      style = "open-street-map",
      center = list(lon = -122, lat = 38.5),
      zoom = 6
    ),
    showlegend = TRUE) %>%
  highlight(color = 'black')
```

Arranging in the file

I bscols to format my individual pieces in the larger div(). I titled each section using

and added a little more details. I also added a filter for the number of beds and baths since those are common filtering questions people have when looking for homes.

```
a <- div(
   css,
    bscols(widths = c(12),
         h1("Interactive Visualizations"),
         h2("Author: Veda Nayak"),
         p("Clicked and corrsponding residences on the scatter plot, map, and table will show up as foll
         p("   - Scatter Plot: The selected term will be highlighted. Remaing terms will loose opacity.
         p("   - Map: The selected term will be highlighted in black."),
         p("   - Table: The selected term will become the only one visible."),
         p("If you want to select multiple points, hold shift as you click on them."),
         p("To unselect terms and reset your view, double click in the white space on the graph."),
```

```
        p("If you would like to unselect a whole county, click on the county on the legand."),
        p("If you have specifications on what property you are looking for, you can use the filters to
        p("Have a good day!")

  ),

  bscols(widths = c(2, 5, 5),
        h3("Filters:"),
        filter_checkbox("bedrooms", "Bedrooms", sd, ~beds, inline = TRUE),
        filter_checkbox("baths", "Baths", sd, ~baths, inline = TRUE)

  ),

    bscols(widths = c(6, 6),
        div(h3("Square Footage vs. Price"), scatter_interact),
        div(h3("Property Locations"), plotlyMap)
  ),

  div(
    h3("Property Details"),
    datatable(sd,
              filter = 'top',
              colnames = c( 'Property',
                            'Price',
                            'Square Feet',
                            'Beds',
                            'Baths',
                            'Garage Capacity',
                            'Last Selling Price',
                            'Last Sold Date',
                            'Street Address',
                            'Address Locality',
                            'County',
                            'Year Built',
                            'Listing Link',
                            'Row ID',
                            'Longitude',
                            'Latitude'),
              options = list(
                pageLength = 10,
                scrollX = TRUE
              ),
              escape = FALSE) %>%
      formatStyle(columns = 1:ncol(pdShared),
                  valueColumns = 'county',
                  backgroundColor = styleEqual(names(pastelPalette), pastelPalette))
  )
)
```

I was running into errors when I used saveWidget, so I used save_html to generate my final document.

```
save_html(a, "linked.html")
```

Validation

I first made sure that the linking that I set up for it was running as I expected. When I click on a point on the map the scatter plot updates to highlight that point, and visa vera. I also checked that the tooltips on the map and scatter plot match up with eachother and with the information in the table.

I then wanted to make sure that this file could stand alone. After moving it to another directory, I saw that the file was still able to function independently, though if I a new file would be made in my working directory instead of updating the original file.

Acknowledgments

- https://rstudio.github.io/crosstalk/using.html
- https://stackoverflow.com/questions/70746921/arrange-crosstalk-graphs-via-bscols
- https://stackoverflow.com/questions/78929440/using-r-crosstalk-plotly-and-dt-is-there-a-way-of-filtering-a-plotly-using-a-t
- https://stackoverflow.com/questions/71819237/ggplotly-clickable-link-in-r-plot
- https://rdrr.io/cran/crosstalk/man/SharedData.html
- https://rstudio.github.io/DT/010-style.html
- https://www.w3schools.com/css/css_text.asp
- I used ClaudeAI to help fix the css formatting, add line breaks in the tooltips, and to help with the initial setup of 'a'. I also used it to understand how to link the widgets (I didn't realize the SharedData$new() step was important at first).