Create Interactive Data Visualization with Plotly in R

2/06/2020

# Overview

* This is an RMarkdown explaining how we use the Plotly package to more effectivley communicate clinical and translational research via interactive graphs
* Plotly is a computing company that develops online data analytics and visualization tools
* It has open sourced many useful interactive visualization products.
* Plotly can be used in several programming languages (e.g. Python, R and JavaScript)
  + This tutorial will be written with instruction for R users
  + Although <https://plot.ly/> is fabulous resource, we found that there were elements from those tutorials that were missing and we hope that this post will provide a useful resource to get started in Plotly in R
* Why use Plotly?
  + Plotly allows us to create visually appealing interactive plots.
  + The ability to export to html and retain all interactive functionality is easily accomplished with Plotly.
  + It is focused around the ability to generate interactive plots with a few lines of code.
  + Finally, all interactive features are compatible with modern web browsers.

# Step 1, Download Plotly from CRAN

* Use the install.package() function to install the plotly R package from [CRAN](https://cran.r-project.org/) install.packages("plotly")

# Load Relevant Packages

library(tidyverse)  
library(knitr)  
library(plotly)  
library(readxl)  
library(scales)

# Load dataset

* For the first example we will use mtcars (Motor Trend Car Road Tests), which is built into base R

mtcars <- mtcars

## View the mtcars Data Frame

mtcars %>% kable

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 3 | 4 |
| Merc 240D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 | 20.00 | 1 | 0 | 4 | 2 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 | 22.90 | 1 | 0 | 4 | 2 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.30 | 1 | 0 | 4 | 4 |
| Merc 280C | 17.8 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.90 | 1 | 0 | 4 | 4 |
| Merc 450SE | 16.4 | 8 | 275.8 | 180 | 3.07 | 4.070 | 17.40 | 0 | 0 | 3 | 3 |
| Merc 450SL | 17.3 | 8 | 275.8 | 180 | 3.07 | 3.730 | 17.60 | 0 | 0 | 3 | 3 |
| Merc 450SLC | 15.2 | 8 | 275.8 | 180 | 3.07 | 3.780 | 18.00 | 0 | 0 | 3 | 3 |
| Cadillac Fleetwood | 10.4 | 8 | 472.0 | 205 | 2.93 | 5.250 | 17.98 | 0 | 0 | 3 | 4 |
| Lincoln Continental | 10.4 | 8 | 460.0 | 215 | 3.00 | 5.424 | 17.82 | 0 | 0 | 3 | 4 |
| Chrysler Imperial | 14.7 | 8 | 440.0 | 230 | 3.23 | 5.345 | 17.42 | 0 | 0 | 3 | 4 |
| Fiat 128 | 32.4 | 4 | 78.7 | 66 | 4.08 | 2.200 | 19.47 | 1 | 1 | 4 | 1 |
| Honda Civic | 30.4 | 4 | 75.7 | 52 | 4.93 | 1.615 | 18.52 | 1 | 1 | 4 | 2 |
| Toyota Corolla | 33.9 | 4 | 71.1 | 65 | 4.22 | 1.835 | 19.90 | 1 | 1 | 4 | 1 |
| Toyota Corona | 21.5 | 4 | 120.1 | 97 | 3.70 | 2.465 | 20.01 | 1 | 0 | 3 | 1 |
| Dodge Challenger | 15.5 | 8 | 318.0 | 150 | 2.76 | 3.520 | 16.87 | 0 | 0 | 3 | 2 |
| AMC Javelin | 15.2 | 8 | 304.0 | 150 | 3.15 | 3.435 | 17.30 | 0 | 0 | 3 | 2 |
| Camaro Z28 | 13.3 | 8 | 350.0 | 245 | 3.73 | 3.840 | 15.41 | 0 | 0 | 3 | 4 |
| Pontiac Firebird | 19.2 | 8 | 400.0 | 175 | 3.08 | 3.845 | 17.05 | 0 | 0 | 3 | 2 |
| Fiat X1-9 | 27.3 | 4 | 79.0 | 66 | 4.08 | 1.935 | 18.90 | 1 | 1 | 4 | 1 |
| Porsche 914-2 | 26.0 | 4 | 120.3 | 91 | 4.43 | 2.140 | 16.70 | 0 | 1 | 5 | 2 |
| Lotus Europa | 30.4 | 4 | 95.1 | 113 | 3.77 | 1.513 | 16.90 | 1 | 1 | 5 | 2 |
| Ford Pantera L | 15.8 | 8 | 351.0 | 264 | 4.22 | 3.170 | 14.50 | 0 | 1 | 5 | 4 |
| Ferrari Dino | 19.7 | 6 | 145.0 | 175 | 3.62 | 2.770 | 15.50 | 0 | 1 | 5 | 6 |
| Maserati Bora | 15.0 | 8 | 301.0 | 335 | 3.54 | 3.570 | 14.60 | 0 | 1 | 5 | 8 |
| Volvo 142E | 21.4 | 4 | 121.0 | 109 | 4.11 | 2.780 | 18.60 | 1 | 1 | 4 | 2 |

# Make an Interactive Bar Chart in Plotly

* For this first example, we will make a bar chart of the number of vehicles with “4”, “6” and “8” cylinders
* One approach is to create a new data frame with two vectors (columns)
  + one that we create with the String “Four Cylinders”, “Six Cylinders”, and “Eight Cylinders” and
  + another Vector that is the sum of all of the rows in mtcars that have a “4”, “6” or “8” in them.
  + We’ll call the first vector **“vehicles”** and the second **“cylinders”**
* This allows us to make a table that has a very defined x-axis and y-axis to make construtcting a bar graph very straight forward

vehicles <- c("Four Cylinders","Six Cylinders","Eight Cylinders")  
cylinders <- c(sum(mtcars$cyl==4), sum(mtcars$cyl==6), sum(mtcars$cyl==8)) # of note, the sum() function will allow us to add up all the observations with either a "4" or a "6" or a "8"

## View these two vectors

vehicles %>% kable

|  |
| --- |
| x |
| Four Cylinders |
| Six Cylinders |
| Eight Cylinders |

cylinders %>% kable

|  |
| --- |
| x |
| 11 |
| 7 |
| 14 |

* As you can see, there are 11 vehicles with “4” contained in the columns’ cells, 7 with “6” and 14 with “8”
* Now combine these two vectors into a tibble, we’ll call it “veh\_cyl”
  + Of note, a tibble is a modern rework of the standard data.frame, with some internal improvements to make code more reliable.
    - They are data frames, but do not follow all of the same rules
      * For example, tibbles can have column names that are not normally allowed, such as numbers/symbols.

veh\_cyl <- tibble(vehicles, cylinders)

# View new Tibble

veh\_cyl %>% kable

|  |  |
| --- | --- |
| vehicles | cylinders |
| Four Cylinders | 11 |
| Six Cylinders | 7 |
| Eight Cylinders | 14 |

##### Now that we have a tibble that can be easily turned into a bar graph, let’s use Plotly to make an interactive graph

plot\_ly(data = veh\_cyl, x = vehicles, y = cylinders, type = "bar", text = cylinders, textposition = "auto") %>%   
 layout(title = "Number of Vehicles in mtcars with 4, 6, and 8 Cylinders",  
 titlefont = list(size = 28, color = "orange", family = "Calibri"),  
 yaxis = list(title = "Number of Vehicles",  
 titlefont = list(color = "black", family = "Arial", size = 26),  
 tickfont = list(color = "black", family = "Arial", size = 20)),  
 xaxis = list(title = "Number of Cylinders",  
 titlefont = list(color = "red", family = "Times New Roman", size = 22),  
 tickfont = list(color = "green", family = "Cambria", size = 18)))%>%   
 layout(margin = list(   
 l = 10,  
 r = 10,  
 b = 0,  
 t = 40)) # Use the layout(margin) function to adjust the margins of the graph

## Warning: The titlefont attribute is deprecated. Use title = list(font = ...)  
## instead.

##### Comments about code for Titles and Axes

* To adjust the title, axes and margins, plotly uses different code that perhaps the more popular ggpolot package that R users are often very familiar with
  + The layout() and list() functions perform much of the work for these tasks, as seen above
  + In order to highlight how these are used, we’ve made the title Orange and the axis font Green and the axis title Red to emphaise those elements of the code.
* To add the values of the bar on the top of the bars use textposition = "auto" in the first ()
* You don’t have to use those codes to adjust your Titles and Axes, but if you don’t a very basic plot will look like this…

plot\_ly(data = veh\_cyl, x = vehicles, y = cylinders, type = "bar", text = cylinders)

# Make a Time Series in Plolty

* We will use a data set downloaded from GitHub that has dates and corresponding data
* This will also highlight how to download data directly from GitHub into your R Studio
* you will need the packages httr and RCurl

library(httr)  
library(RCurl)

### Download List of publications from Github

* We are going to uses a dataset from **Reproducible Research in Ecology, Evolution, Behaviour, and Environmental Studies** <https://github.com/opetchey/RREEBES/tree/master/Beninca_etal_2008_Nature>

no2 <- read.csv(text=getURL("https://raw.githubusercontent.com/opetchey/RREEBES/master/Beninca\_etal\_2008\_Nature/data/nutrients\_original.csv"), skip=7, header=T)

### Convert first column into dates

no2$Date <- as.Date(no2$Date, "%d/%m/%y") # of note, this %d/%m/%y format is critical, and is specific to the way the data is put into the data frame; b/c here the date in the original df is d/m/2 digit year, you need a lower case "y" to indicate that year is only two digits in the data frame

### Graph a Timeseries in Plotly

plot\_ly( data = no2, x = no2$Date, y = no2$NO2) %>%   
 add\_trace(type = "scatter" ,mode = "lines+markers") %>%   
 layout(  
 title = "Time Series of NO2",  
 xaxis = list(  
 title = "Year"),  
 yaxis = list(  
 title = "NO2"))

# Make a Chloropleth Map in Plotly

* A Chloropleth Map is a map that uses differences in shading, coloring, or the placing of symbols within predefined areas to indicate the average values of a property or quantity in those areas (<https://en.wikipedia.org/wiki/Choropleth_map>)
* For example let’s look at the rate of new cancers in the US according to the **Centers for Disease Control and Prevention (CDC) and the National Cancer Institute (NCI), 2016**
* Visit “<https://gis.cdc.gov/Cancer/USCS/DataViz.html>” and click “export” to download a csv file.
  + Make sure to save it in your project folder
  + Also, please create a column,”state”, and enter the state abbreviations next to the corresponding full state name in “Area”.
    - This is needed to add values to each state

df <- read.csv("USCS\_OverviewMap.csv")

### View the Data Fram df

df %>% kable

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Area | CancerType | Year | Sex | AgeAdjustedRate | CaseCount | Population | X |
| NM | ‘New Mexico’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 359.4 | 9075 | 2085432 | 0.0043516 |
| AZ | ‘Arizona’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 376.3 | 31443 | 6908642 | 0.0045513 |
| CA | ‘California’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 385.6 | 164887 | 39296476 | 0.0041960 |
| CO | ‘Colorado’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 388.8 | 23244 | 5530105 | 0.0042032 |
| DC | ‘District of Columbia’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 378.3 | 2566 | 684336 | 0.0037496 |
| NV | ‘Nevada’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 385.0 | 13054 | 2939254 | 0.0044413 |
| AK | ‘Alaska’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 405.8 | 2882 | 741522 | 0.0038866 |
| HI | ‘Hawaii’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 414.2 | 7395 | 1428683 | 0.0051761 |
| MA | ‘Massachusetts’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 404.2 | 33626 | 6823721 | 0.0049278 |
| OR | ‘Oregon’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 405.3 | 20596 | 4085989 | 0.0050406 |
| TX | ‘Texas’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 391.8 | 109083 | 27904862 | 0.0039091 |
| UT | ‘Utah’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 390.6 | 10494 | 3044321 | 0.0034471 |
| VA | ‘Virginia’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 413.3 | 40322 | 8414380 | 0.0047920 |
| WY | ‘Wyoming’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 402.3 | 2775 | 584910 | 0.0047443 |
| FL | ‘Florida’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 419.3 | 119408 | 20656589 | 0.0057806 |
| ID | ‘Idaho’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 439.2 | 8354 | 1680026 | 0.0049725 |
| IN | ‘Indiana’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 443.0 | 34260 | 6634007 | 0.0051643 |
| MD | ‘Maryland’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 440.9 | 30942 | 6024752 | 0.0051358 |
| MI | ‘Michigan’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 436.8 | 53911 | 9933445 | 0.0054272 |
| MN | ‘Minnesota’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 455.7 | 29619 | 5525050 | 0.0053609 |
| MO | ‘Missouri’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 446.3 | 33171 | 6091176 | 0.0054457 |
| NE | ‘Nebraska’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 446.4 | 9838 | 1907603 | 0.0051573 |
| ND | ‘North Dakota’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 448.7 | 3765 | 755548 | 0.0049831 |
| OK | ‘Oklahoma’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 447.9 | 20167 | 3921207 | 0.0051431 |
| RI | ‘Rhode Island’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 450.6 | 5972 | 1057566 | 0.0056469 |
| SC | ‘South Carolina’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 440.9 | 27313 | 4959822 | 0.0055069 |
| SD | ‘South Dakota’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 449.5 | 4612 | 861542 | 0.0053532 |
| TN | ‘Tennessee’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 455.7 | 36598 | 6649404 | 0.0055040 |
| VT | ‘Vermont’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 442.4 | 3681 | 623354 | 0.0059052 |
| WA | ‘Washington’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 445.1 | 37378 | 7280934 | 0.0051337 |
| AL | ‘Alabama’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 457.8 | 27195 | 4860545 | 0.0055951 |
| AR | ‘Arkansas’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 469.6 | 17053 | 2988231 | 0.0057067 |
| CT | ‘Connecticut’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 465.5 | 21117 | 3587685 | 0.0058860 |
| DE | ‘Delaware’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 487.2 | 6001 | 952698 | 0.0062990 |
| GA | ‘Georgia’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 468.8 | 52056 | 10313620 | 0.0050473 |
| IL | ‘Illinois’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 462.8 | 68954 | 12835726 | 0.0053720 |
| IA | ‘Iowa’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 473.6 | 18146 | 3130869 | 0.0057958 |
| KS | ‘Kansas’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 457.3 | 15312 | 2907731 | 0.0052660 |
| KY | ‘Kentucky’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 509.7 | 27137 | 4436113 | 0.0061173 |
| LA | ‘Louisiana’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 473.1 | 25451 | 4686157 | 0.0054311 |
| ME | ‘Maine’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 473.4 | 8901 | 1330232 | 0.0066913 |
| MS | ‘Mississippi’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 465.8 | 16265 | 2985415 | 0.0054482 |
| MT | ‘Montana’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 456.6 | 6194 | 1038656 | 0.0059635 |
| NH | ‘New Hampshire’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 480.9 | 8442 | 1335015 | 0.0063235 |
| NJ | ‘New Jersey’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 474.8 | 51521 | 8978416 | 0.0057383 |
| NY | ‘New York’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 474.8 | 113026 | 19836286 | 0.0056979 |
| NC | ‘North Carolina’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 458.4 | 55394 | 10156689 | 0.0054539 |
| OH | ‘Ohio’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 456.1 | 65645 | 11622554 | 0.0056481 |
| PA | ‘Pennsylvania’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 482.5 | 80089 | 12787085 | 0.0062633 |
| WV | ‘West Virginia’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 472.0 | 11698 | 1828637 | 0.0063971 |
| WI | ‘Wisconsin’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 458.6 | 32688 | 5772917 | 0.0056623 |

### Add a vector to the Data Frame specifying what will be revealed via hovering

df$hover <- with(df, paste(df$Area, df$Year,"Sex:", df$Sex, "Case count:", df$CaseCount, sep = "<br>"))

* This is a very nice option.
  + You can create a vector (here we are calling it “hover”) that will be the contents displayed when you hover over the map
    - This “hover” vector needs to be specified in the text = part of the code below

#### View the first 2 rows of the data frame df with hover vector

kable(head(df[1:2,]))

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Code | Area | CancerType | Year | Sex | AgeAdjustedRate | CaseCount | Population | X | hover |
| NM | ‘New Mexico’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 359.4 | 9075 | 2085432 | 0.0043516 | ‘New Mexico’‘2016’Sex:‘Male and Female’Case count:9075 |
| AZ | ‘Arizona’ | ‘All Types of Cancer’ | ‘2016’ | ‘Male and Female’ | 376.3 | 31443 | 6908642 | 0.0045513 | ‘Arizona’‘2016’Sex:‘Male and Female’Case count:31443 |

### Plot the Chloropleth Map

# give state boundaries a white border  
l <- list(color = toRGB("white"), width = 2)  
# specify some map projection/options  
g <- list(  
 scope = 'usa',  
 projection = list(type = 'albers usa'),  
 showlakes = TRUE,  
 lakecolor = toRGB('steelblue')  
)  
  
p <- plot\_geo(df, locationmode = 'USA-states') %>%  
 add\_trace(  
 z = ~AgeAdjustedRate, text = ~hover, locations = df$Code,  
 color = ~AgeAdjustedRate, colors = 'Purples'  
 ) %>%  
 colorbar(title = "Rate per 100,000 people") %>%  
 layout(  
 title = 'US Cancer Statistics Rate of New Cancers- 2016 (Source: CDC and NCI)',  
 geo = g  
 )  
  
p

# SessionInfo

sessionInfo()

## R version 3.6.1 (2019-07-05)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS Mojave 10.14.6  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] RCurl\_1.95-4.12 bitops\_1.0-6 httr\_1.4.1 scales\_1.1.0   
## [5] readxl\_1.3.1 plotly\_4.9.0 knitr\_1.26 forcats\_0.4.0   
## [9] stringr\_1.4.0 dplyr\_0.8.3 purrr\_0.3.3 readr\_1.3.1   
## [13] tidyr\_1.0.0 tibble\_2.1.3 ggplot2\_3.2.1 tidyverse\_1.2.1  
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_1.0.3 lubridate\_1.7.4 lattice\_0.20-38 assertthat\_0.2.1   
## [5] digest\_0.6.23 mime\_0.7 R6\_2.4.1 cellranger\_1.1.0   
## [9] backports\_1.1.5 evaluate\_0.14 highr\_0.8 pillar\_1.4.3   
## [13] rlang\_0.4.4 lazyeval\_0.2.2 rstudioapi\_0.10 data.table\_1.12.6   
## [17] rmarkdown\_1.17 htmlwidgets\_1.5.1 munsell\_0.5.0 shiny\_1.4.0   
## [21] broom\_0.5.2 compiler\_3.6.1 httpuv\_1.5.2 modelr\_0.1.5   
## [25] xfun\_0.11 pkgconfig\_2.0.3 htmltools\_0.4.0 tidyselect\_0.2.5   
## [29] fansi\_0.4.1 viridisLite\_0.3.0 crayon\_1.3.4 withr\_2.1.2   
## [33] later\_1.0.0 grid\_3.6.1 nlme\_3.1-140 jsonlite\_1.6   
## [37] xtable\_1.8-4 gtable\_0.3.0 lifecycle\_0.1.0 magrittr\_1.5   
## [41] cli\_2.0.1 stringi\_1.4.5 farver\_2.0.3 promises\_1.1.0   
## [45] xml2\_1.2.2 generics\_0.0.2 vctrs\_0.2.2 RColorBrewer\_1.1-2  
## [49] tools\_3.6.1 glue\_1.3.1 hms\_0.5.2 crosstalk\_1.0.0   
## [53] fastmap\_1.0.1 yaml\_2.2.0 colorspace\_1.4-1 rvest\_0.3.4   
## [57] haven\_2.1.1