As we gear up for Christmas day tomorrow, I wanted to take this time to wish a very happy holiday season to all of my fellow data geeks!

Rather than posting a happy holiday e-Card, I thought I might indulge in creating some gratuitous holiday themed graphs. In this blog we will showcase a tool called [coordinator](https://spotify.github.io/coordinator/), created by [Aliza Aufrichtig](https://twitter.com/alizauf) which allows us to convert SVG graphics to data co-ordinates. This will enable our graphed holiday message. We will leverage the ggimage and magick packages ([again](https://www.littlemissdata.com/blog/bacheloranalysis)) to apply a background image. Finally, we will use gganimate, to create a little holiday gif.

**Load the Packages**

library(data.table)

library(tidyverse)

library(gganimate)

library(ggimage)

library(magick)

library(gifski)

**Download the data**

The csv is created with a tool called [coordinator](https://spotify.github.io/coordinator/), by [Aliza Aufrichtig.](https://twitter.com/alizauf) It allows us to convert SVG graphics to data co-ordinates. More on this in a [previous blog post.](https://www.littlemissdata.com/blog/lmd-loves-cloud)

df= fread('<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/Holidays/Holidays.csv>', stringsAsFactors = FALSE)

**Create a Holiday Graph**

Create a classic ggplot scatterplot. Remove all lines and chart aspects with theme\_void(). Increase the chart size beyond the plotted points using the xlim() and ylim() functions. This will allow us to nicely center the text on our background image.

# Basic scatter plot

p <-ggplot(df, aes(x=x, y=y)) +

geom\_point(colour = '#BF0000', size = 0.3) +

xlim(0, 18000) +

ylim(0, 10000) +

theme\_void()

p

**Add a background Holiday Image**

Add the holiday background using the ggbackground() function in the ggimage package.

img <- "<https://raw.githubusercontent.com/lgellis/MiscTutorial/master/Holidays/%20background.jpg>"

imgRead <- image\_read(img)

ggbackground(p, img)

**Animation**

Create an animated holiday greeting with the gganimate package. Use transition\_manual() vs transition\_reveal() because it allows us to specify cumulative=TRUE to keep all data already plotted. Use the gifski package and anim\_save() function to save the gif.

t <-ggplot(df, aes(x=x, y=y)) +

theme\_void() +

theme(plot.background = element\_rect(fill = '#CA302F')) +

geom\_point(colour = 'white', size = 1) +

transition\_manual(as.numeric(x), cumulative=TRUE)

gif <- animate(t, end\_pause = 25, width = 800, height = 400, fps = 8)

gif

anim\_save("HappyHolidays.gif", gif)

**Thank you!**

Thank you again to my fellow data lovers for such a wonderful 2019. I can’t wait to tackle 2020 with y’all!