# Visualizing ebird observation data of Sandhill Cranes in Wisconsin

```
library(tidyverse)
```

```
## -- Attaching packages -----------------------------------------------------------------------
--------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.3.0
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## -- Conflicts --------------------------------------------------------------------------------
--------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.3-6, (SVN revision 773)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Users/12147/Documents/R/win-library/3.5/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/12147/Documents/R/win-library/3.5/rgdal/proj
##  Linking to sp version: 1.3-1
```

```
library(leaflet)
```

# Data wrangling

```
#reading ebird data in its original requested format
SCdata <- read.csv("ebird_WI_sandhillcranes.csv", header = TRUE, sep=",")

#month
SC.ObsMonthName <- months(mdy(SCdata$OBSERVATION.DATE))
#month number - to be used for sorting
SC.ObsMonthNumber <- month(mdy(SCdata$OBSERVATION.DATE))

#replace X with 1 (seen but how many not reported)
SC.ObsCount <- as.numeric(str_replace(SCdata$OBSERVATION.COUNT, "X", "1"))
SC.County <- SCdata$COUNTY
SC.Latitude <- SCdata$LATITUDE
SC.Year <- year(mdy(SCdata$OBSERVATION.DATE))

#adding all the necessary columns into a dataframe
SCdata <- cbind.data.frame(SC.ObsMonthName, SC.ObsMonthNumber, SC.ObsCount, SC.County, SC.Latitude, SC.Year)

#order by month
SCdata <- SCdata[order(SC.ObsMonthNumber),]

#filter by year
SCdata <- subset(SCdata, SC.Year == 2017)
```

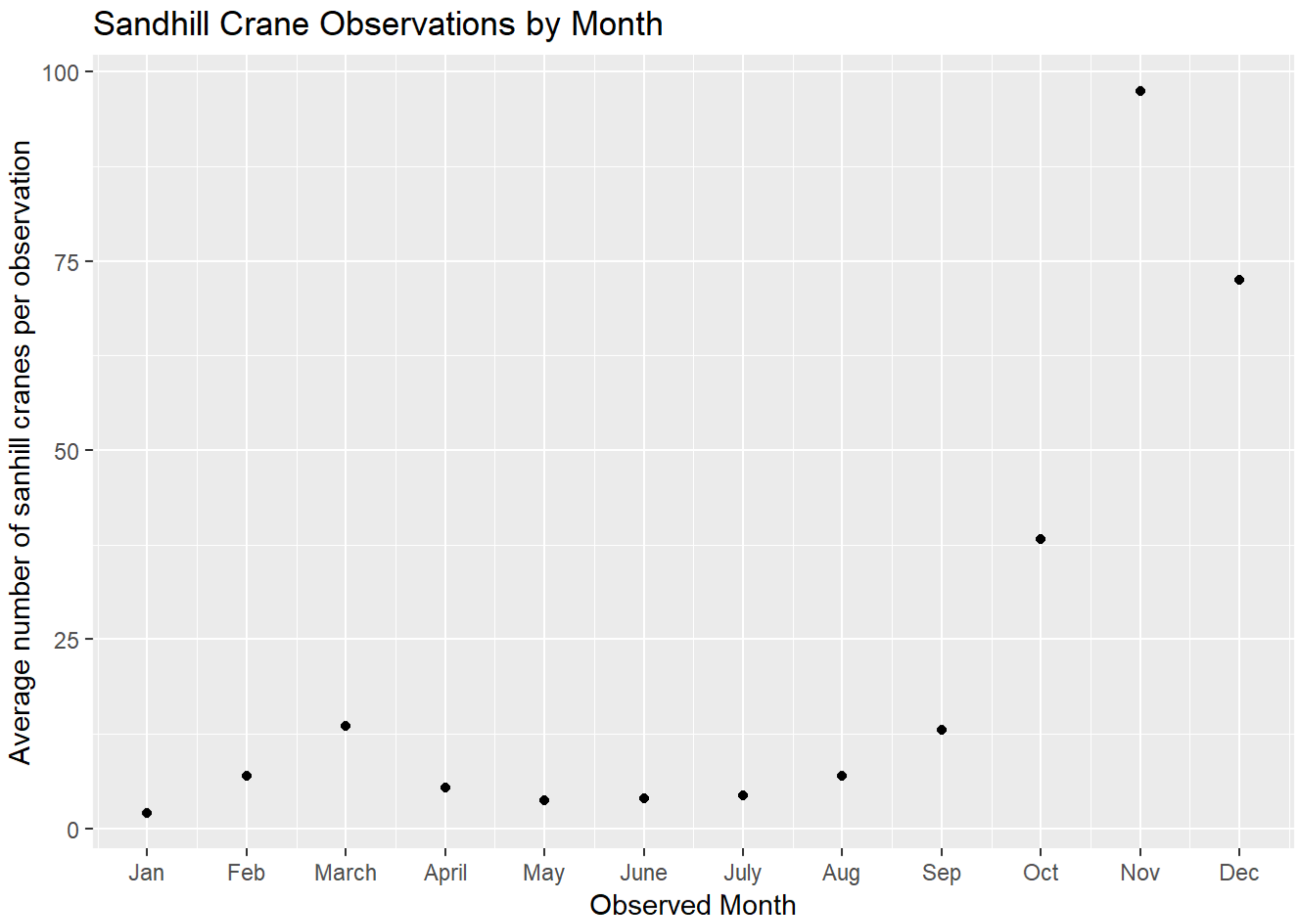
# Calculating average count by Month

```
#calculating average number of sandhill cranes seen per observation each Month of the year
CountByMonth <- SCdata %>% group_by(SC.ObsMonthNumber) %>% summarise(SC.ObsCount = mean(as.numeric(SC.ObsCount)))
```

# Visualizing using ggplot

```
#creating two vectors of month to be used to plot using ggplot
MonthNumber = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
Month = c("Jan", "Feb" , "March", "April", "May", "June", "July", "Aug", "Sep", "Oct", "Nov", "Dec")

#plot average number of sandhill cranes seen per observation each month of the year using ggplot
ggplot(CountByMonth, aes(x=SC.ObsMonthNumber, y=SC.ObsCount))+
      geom_point() +
      scale_x_continuous(breaks = MonthNumber, label = Month) +
      labs(x = "Observed Month", y = "Average number of sanhill cranes per observation") +
      ggtitle("Sandhill Crane Observations by Month")
```



# Calculating average count by county

```
#calculating average number of sandhill cranes seen per observation in each of counties in Wisco
nsin
CountByCounty <- SCdata %>% group_by(SC.County) %>% summarise(SC.ObsCount = mean(as.numeric(SC.O
bsCount)))
```

# Visualizing county level data using leaflet

```
#reading WI counties from USA states shape file
counties <- readOGR(dsn= 'data/counties', layer = 'cb_2017_us_county_500k') %>% .[.$STATEFP == 5
5,]
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\12147\Documents\d drive\Stats2\data\counties", layer: "cb_2017_us_county_50
0k"
## with 3233 features
## It has 9 fields
## Integer64 fields read as strings:  ALAND AWATER
```

```
#joining the county level count data
counties@data <- left_join(counties@data, CountByCounty, by= c('NAME' = 'SC.County'))
```

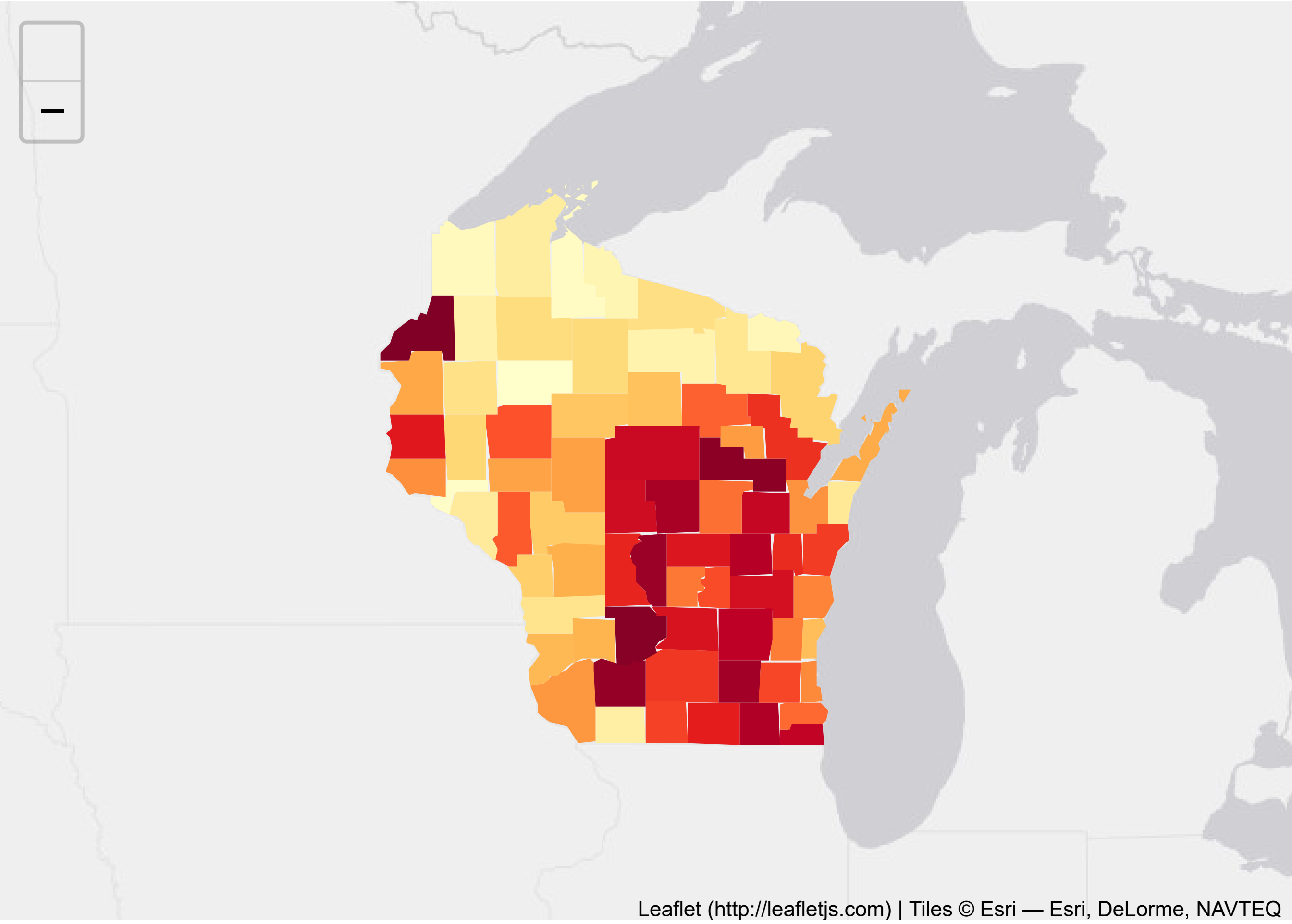
```
## Warning: Column `NAME`/`SC.County` joining factors with different levels,
## coercing to character vector
```

```
#setting popup label
popup <- paste0('<i>', counties$NAME, ' County</i><br><br>',
                '<b>Avg Obs Count:</b>', round(counties$SC.ObsCount,0), '<br>')

#setting colorcode
palette <- colorFactor("YlOrRd", counties$SC.ObsCount)

#using leaflet package to render average count of sandhill cranes by county
leaflet(data = counties) %>%
  addProviderTiles('Esri.WorldGrayCanvas') %>%
  addPolygons(stroke = FALSE,
              fillColor = ~ palette(counties$SC.ObsCount),
              fillOpacity = 1,
              popup = popup)
```

```
## Warning in RColorBrewer::brewer.pal(max(3, n), palette): n too large, allowed maximum for pal
ette YlOrRd is 9
## Returning the palette you asked for with that many colors
```


–
Leaflet (http://leafletjs.com) | Tiles © Esri — Esri, DeLorme, NAVTEQ