Challenge_5: Visualizing Time and Space

AUTHOR
Mehak Nargotra

PUBLISHED
April 1, 2024

Make sure you change the author's name in the above YAML header.

Setup

If you have not installed the following packages, please install them before loading them.

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.4     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(readr)
library(readxl)
library(haven) #for loading other datafiles (SAS, STATA, SPSS, etc.)

#for plotting time
library(ggplot2) # if you have not installed this package, please install it.
library(lubridate)



#for plotting space
library(sp)
library(sf)
```

```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(maps)
```

```
Attaching package: 'maps'

The following object is masked from 'package:purrr':

    map
```

Challenge Overview

In this challenge, we will practice the visualization skills learned in the class with two datasets to capture the temporal and spatial patterns.

There will be coding components and writing components. Please read the instructions for each part and complete your challenges.

Datasets

- Part 1. The UFO Sightings Data (50%)
- Part 2. The Hospital Location Data (50%)

Find the `_data` folder, then read the datasets using the correct R command.

Part 1. The UFO Sightings Data (50%)

This data contains over 80,000 reports of UFO sightings over the last century in six major countries (and other places). You can learn more about this data by checking: https://www.kaggle.com/datasets/NUFORC/ufo-sightings.

1. **Read and Describe the Data (10%)**

   What is the dimension of the data? What do the columns mean? What is the unit of observation?

```
#type of your code/command here.
complete_UFO <- read_csv("~/Desktop/DACSS 601/DACSS_601_datasets/complete_UFO.cs
```

```
New names:
• `` -> `...12`

Warning: One or more parsing issues, call `problems()` on your data frame for
details,
e.g.:
  dat <- vroom(...)
  problems(dat)

Rows: 88875 Columns: 12
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (9): datetime, city, state, country, shape, duration (hours/min), commen...
dbl (3): duration (seconds), longitude, ...12

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dimensions <- dim(complete_UFO)
```

```
    dimensions
```

```
[1] 88875    12
```

```
    print("The UFO dataset has 88875 rows and 12 columns.")
```

```
[1] "The UFO dataset has 88875 rows and 12 columns."
```

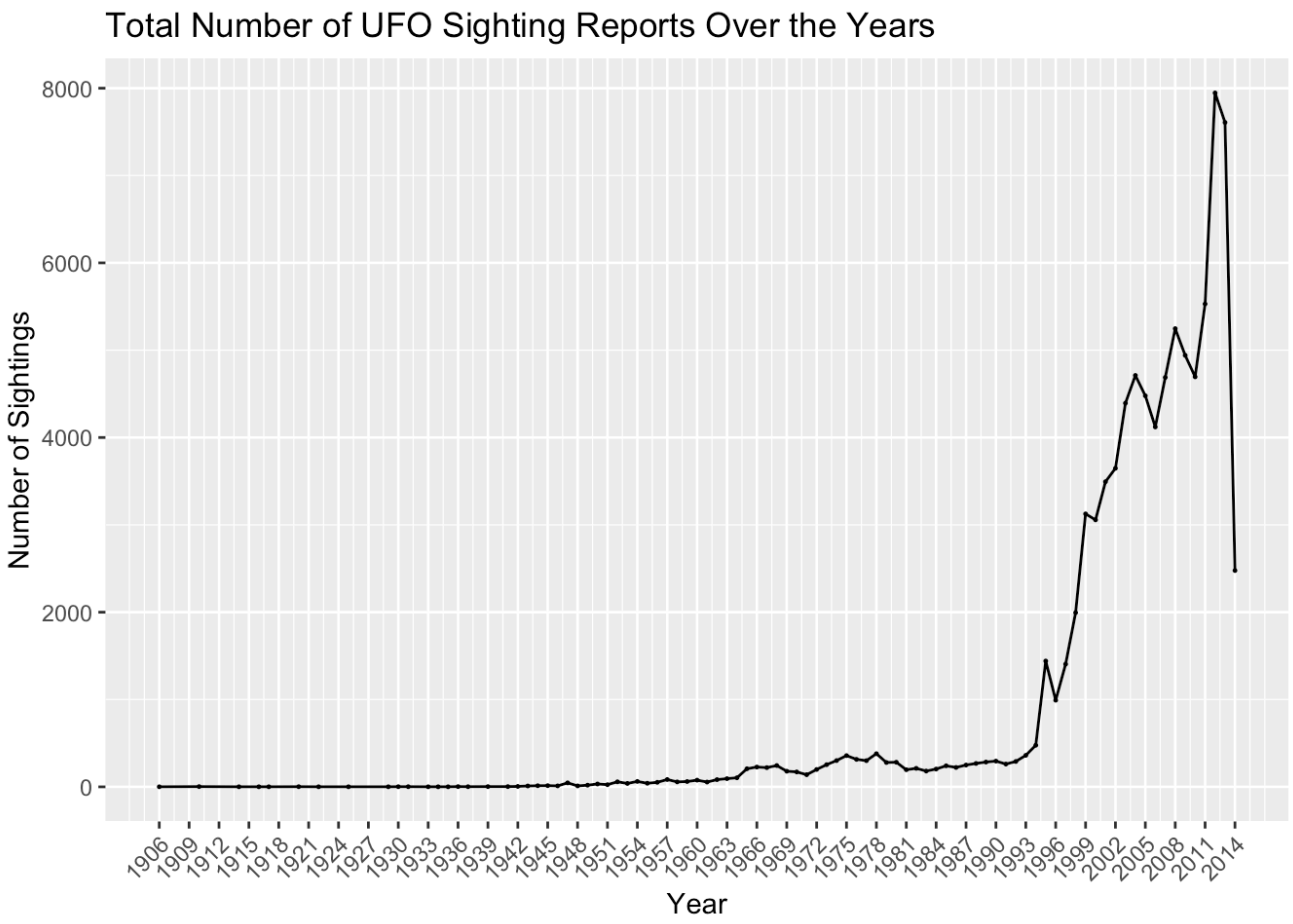
2. Please plot a temporal/time-series graph to present the following patterns. **You may need to subset or mutate the data for graphing.**

   (1) the total number of UFO sighting reports over the years (date_break = year). **(15%)**

```
complete_UFO$datetime <- as.Date(complete_UFO$datetime, format = "%m/%d/%Y")

complete_UFO <- complete_UFO %>%
  mutate(year = lubridate::year(datetime))

sightings_by_year <- complete_UFO %>%
  group_by(year) %>%
  summarise(total_sightings = n())

dim(sightings_by_year)
```

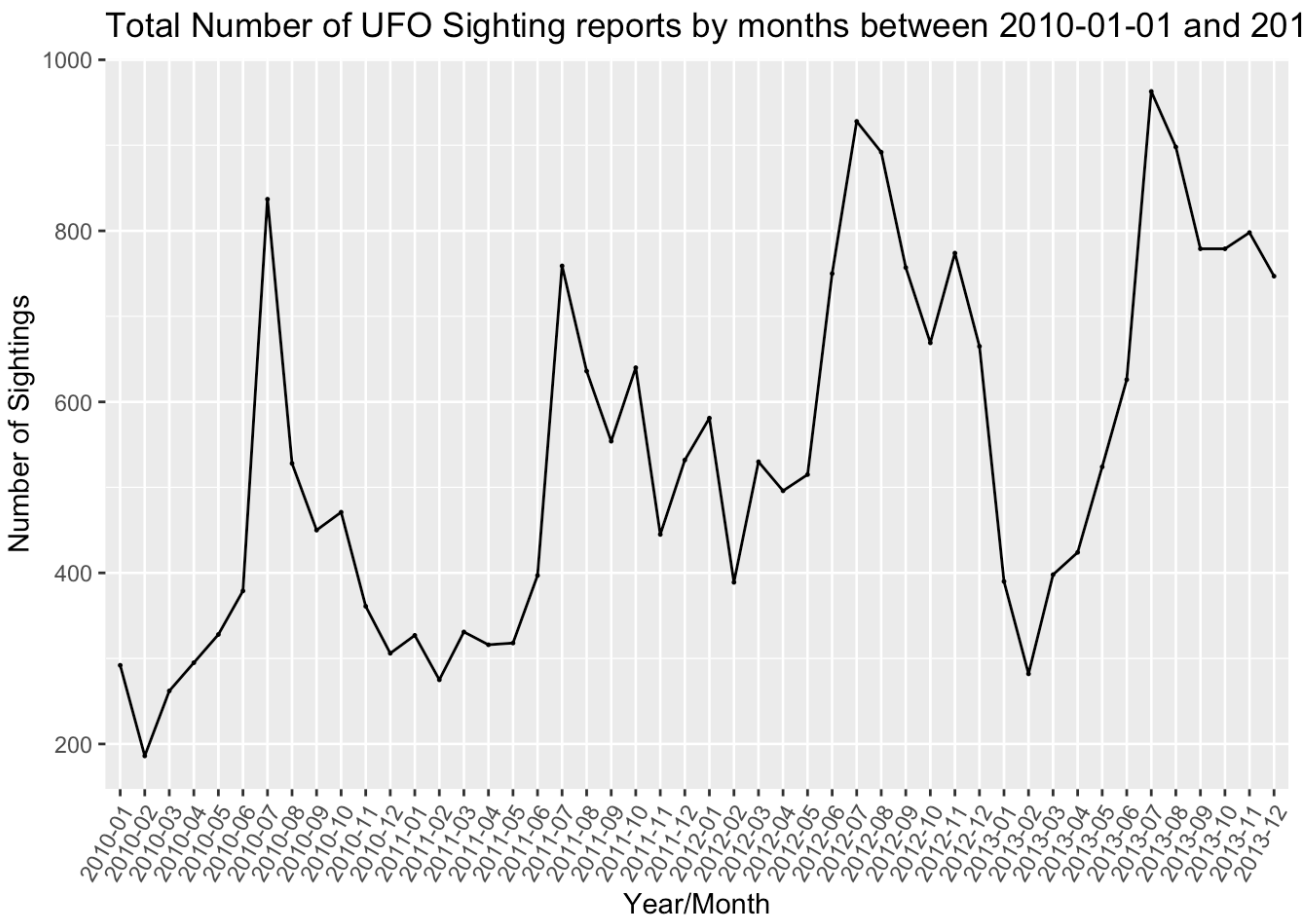
```
[1] 91  2
```

```
ggplot(data = sightings_by_year, aes(x = year, y = total_sightings)) +
  geom_point(size=0.2)+
  geom_line() +
  scale_x_continuous(breaks = seq(min(sightings_by_year$year), max(sightings_by_year
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Total Number of UFO Sighting Reports Over the Years", x = "Year", y
```



\(2\) the total number of UFO sighting reports by months between 2010-01-01 and 2014-01-01. **(15%)**

```
#type of your code/command here.
UFO_updated <- complete_UFO %>%
filter(datetime >= as.Date("2010-01-01") & datetime < as.Date("2014-01-01"))

UFO_updated$month <- format(UFO_updated$datetime, "%Y-%m")

sightings_by_month <- UFO_updated %>%
  group_by(month) %>%
  summarise(total_sightings = n())


ggplot(data = sightings_by_month, aes(x = month, y = total_sightings, group = 1))
geom_point(size=0.2)+
geom_line() +
theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
labs(title = "Total Number of UFO Sighting reports by months between 2010-01-01 ar
```



3. Please write a paragraph describing the patterns you find on the two graphs above. **(10%)**

```
print("In the first graph, we can see that there are almost no UFO's sighted before
Hence, we can see a gradual increase even though there are small ups and downs seen.

Whereas, in the second graph, we can see an increasing - decreasing pattern can beer
```

```
[1] "In the first graph, we can see that there are almost no UFO's sighted before
1960's. The number of UFO sightings increased after that. However, it took a peak
after 1996. \nHence, we can see a gradual increase even though there are small ups
and downs seen.\n\nWhereas, in the second graph, we can see an increasing -
decreasing pattern can been observed. There are some peaks which are observed. If we
look closely, we can see that the peaks are observed around the month of July every
year. Hence, we can say that most of the UFO's between the years 2010 to 2014 were
sighted in the month of July. Similarly, we can see that the least number of UFO's
for a particular year were observed in the month of February. "
```

4. **(Optional)** Use `gganimte` and `gifsky` packages to plot gifs of the above time-series plots. You can refer to codes and commands in the Week#8 demo file.

```
#type of your code/command here.
```

5. **(Optional)** Suppose we are interested in describing the country variations in types and numbers of UFO sighting reports. How can we use bar plots to visualize the difference across countries (using the *country* column, you can recode the empty cells with “others”)? **Note that you may need to do the data transformation or mutation needed to help graphing.**

```
#type of your code/command here.
```

Part 2. The Hospital Location Data (50%)

This data contains the locations of hospitals in the United States. You can learn more about this data by checking: https://www.kaggle.com/datasets/andrewmvd/us-hospital-locations

1. **Read and Describe the Data (10%)**

   What is the dimension of the data? What do the columns mean? What is the unit of observation?

```
#type of your code/command here.
us_hospital_locations <- read_csv("~/Desktop/DACSS 601/DACSS_601_datasets/us_hos
```

```
Rows: 7596 Columns: 15
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (9): NAME, ADDRESS, CITY, STATE, TYPE, STATUS, COUNTY, COUNTRY, OWNER
dbl (6): ID, ZIP, POPULATION, LATITUDE, LONGITUDE, BEDS

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
dim(us_hospital_locations)
```

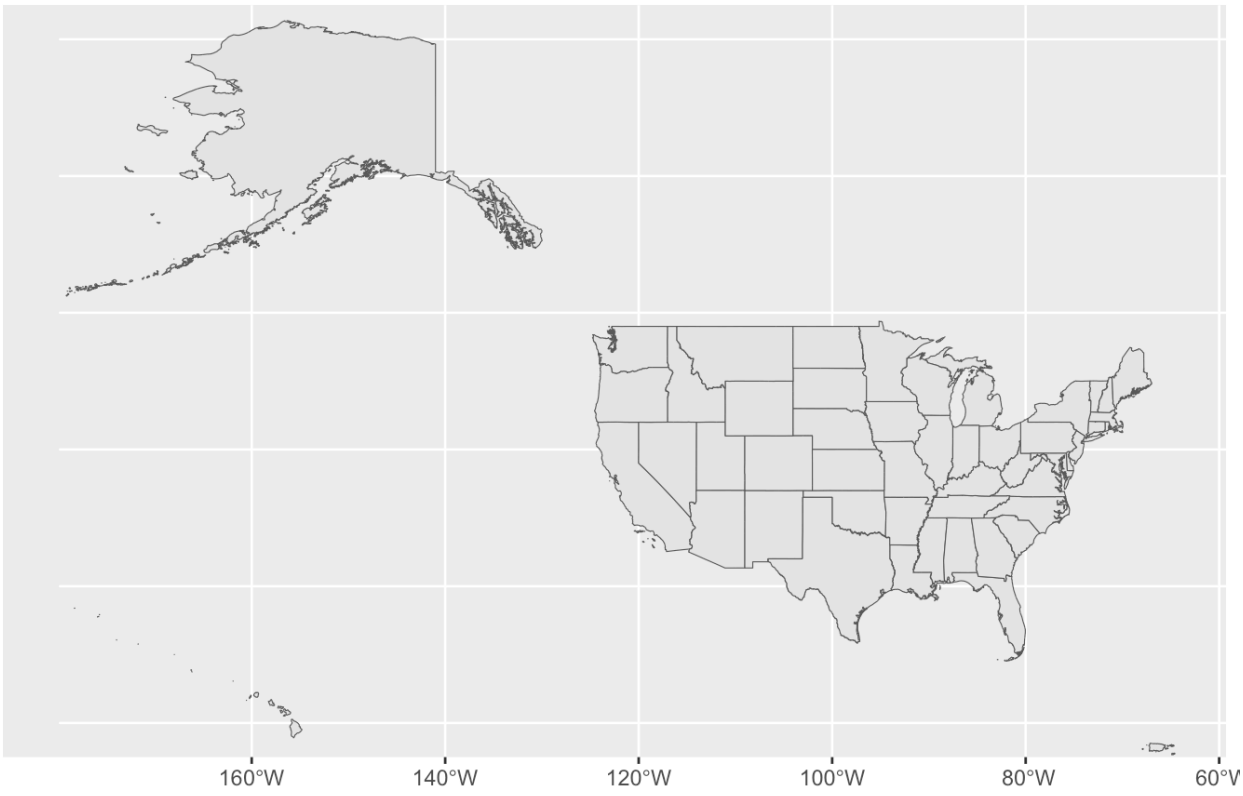
```
[1] 7596   15
```

2. Download the `cb_2018_us_state_500k.shp` data from https://www2.census.gov/geo/tiger/GENZ2018/shp/cb_2018_us_state_500k.zip). Put the whole `cb_2018_us_state_500k` folder under your `_data` folder. Plot a USA map with states boundaries using `geom_sf()`. **(15%)**

```
#type of your code/command here.

states_sf <-
  st_read("~/Desktop/DACSS 601/DACSS_601_datasets/cb_2018_us_state_500k/cb_2018_
```

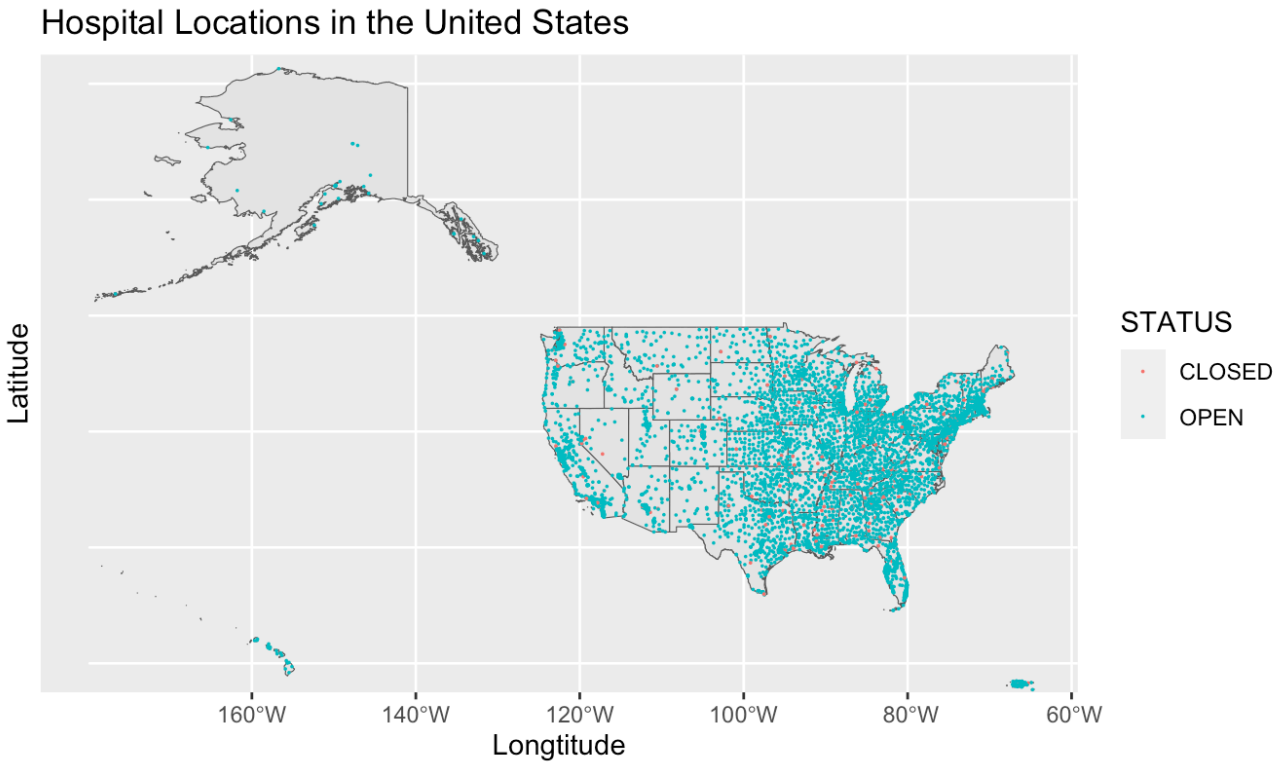
```
Reading layer `cb_2018_us_state_500k' from data source
  `/Users/mehaknargotra/Desktop/DACSS
601/DACSS_601_datasets/cb_2018_us_state_500k/cb_2018_us_state_500k.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 56 features and 9 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -179.1489 ymin: -14.5487 xmax: 179.7785 ymax: 71.36516
Geodetic CRS:  NAD83
```

```r
states <- ggplot()+
  geom_sf(data = states_sf)+
  coord_sf(xlim = c(-180, -65),
     ylim = c(20, 70))

states
```



3. Plot the hospital locations using the *LONGITUDE* and *LATITUDE* with `geom_point()` on the USA map generated above. Mark OPEN and CLOSED hospitals using different colors. **(15%)**

```r
#type of your code/command here.
 states +
  geom_point(data = us_hospital_locations,
         aes(x = LONGITUDE, y = LATITUDE, color = STATUS),
         pch = 19, size=0.00005) +
  labs(x = "Longtitude",
       y = "Latitude",
       title = "Hospital Locations in the United States")
```



4. Please write a paragraph describing the pattern of the hospital locations you find on the map above. **(10%)**

```
#type of your code/command here.
print("Most hospitals are found in big cities on the West Coast (like Los Angele
```

```
[1] "Most hospitals are found in big cities on the West Coast (like Los Angeles
and the San Francisco Bay Area), the East Coast (cities in the Northeast), the
Midwest (like Chicago and cities near the Great Lakes), and the South (like
Dallas, Houston, Orlando, and Miami). In contrast, there are fewer hospitals in
the mountainous areas and the western part of the Great Plains. This trend is
related to where more people live."
```