

Homework 2: Mapping Severe Weather Events

Data Visualization

Jenna Swartz

2024-03-11

- Preparation
- PART 1: Damage From Storms
 - a) State Level Choropleth Maps
 - c) Density Map

Preparation

```
options(repos = 'https://cran.rstudio.com/') # Set CRAN mirror
install.packages("maps")
```

```
## Installing package into 'C:/Users/jenna/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'maps' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'maps'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\jenna\AppData\Local\R\win-library\4.3\00LOCK\maps\libs\x64\maps.dll to
## C:\Users\jenna\AppData\Local\R\win-library\4.3\maps\libs\x64\maps.dll:
## Permission denied
```

```
## Warning: restored 'maps'
```

```
##  
## The downloaded binary packages are in  
## C:\Users\jenna\AppData\Local\Temp\RtmpmW6jdC\downloaded_packages
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.3.3
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —  
## ✓ forcats 1.0.0 ✓ readr 2.1.4  
## ✓ ggplot2 3.4.3 ✓ stringr 1.5.0  
## ✓ lubridate 1.9.3 ✓ tibble 3.2.1  
## ✓ purrr 1.0.2 ✓ tidyr 1.3.0
```

```
## — Conflicts ————— tidyverse_conflicts() —  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::map() masks maps::map()  
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)  
install.packages("ggthemes")
```

```
## Installing package into 'C:/Users/jenna/AppData/Local/R/win-library/4.3'  
## (as 'lib' is unspecified)
```

```
## package 'ggthemes' successfully unpacked and MD5 sums checked  
##  
## The downloaded binary packages are in  
## C:\Users\jenna\AppData\Local\Temp\RtmpmW6jdC\downloaded_packages
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.3.3
```

```
install.packages("remotes")
```

```
## Installing package into 'C:/Users/jenna/AppData/Local/R/win-library/4.3'  
## (as 'lib' is unspecified)
```

```
## package 'remotes' successfully unpacked and MD5 sums checked  
##  
## The downloaded binary packages are in  
## C:\Users\jenna\AppData\Local\Temp\RtmpmW6jdC\downloaded_packages
```

```
remotes::install_github("UrbanInstitute/urbnmapr")
```

```
## Skipping install of 'urbnmapr' from a github remote, the SHA1 (ef9f4488) has not changed since last install.  
## Use `force = TRUE` to force installation
```

```
library(urbnmapr)  
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(leaflet)  
library(RColorBrewer)  
install.packages("geosphere")
```

```
## Installing package into 'C:/Users/jenna/AppData/Local/R/win-library/4.3'  
## (as 'lib' is unspecified)
```

```
## package 'geosphere' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'geosphere'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\jenna\AppData\Local\R\win-library\4.3\00LOCK\geosphere\libs\x64\geosphere.dll  
## to  
## C:\Users\jenna\AppData\Local\R\win-library\4.3\geosphere\libs\x64\geosphere.dll:  
## Permission denied
```

```
## Warning: restored 'geosphere'
```

```
##  
## The downloaded binary packages are in  
## C:\Users\jenna\AppData\Local\Temp\RtmpmW6jdC\downloaded_packages
```

```
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.3.3
```

```
#Loading in storm data
```

```
file_path <- "C:/Users/jenna/Documents/Data Viz/course materials/course_content/Exercises/07_severe_weather_GRADED/data/stor  
ms.csv"
```

```
storm_data <- read_csv(file_path)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
```

```
## e.g.:
```

```
##   dat <- vroom(...)
```

```
##   problems(dat)
```

```
## Rows: 380137 Columns: 49
```

```
## — Column specification —————
```

```
## Delimiter: ","
```

```
## chr (18): STATE, MONTH_NAME, EVENT_TYPE, CZ_TYPE, CZ_NAME, WFO, BEGIN_DATE_T...
```

```
## dbl (26): BEGIN_YEARMONTH, BEGIN_DAY, BEGIN_TIME, END_YEARMONTH, END_DAY, EN...
```

```
## lgl (5): CATEGORY, TOR_OTHER_WFO, TOR_OTHER_CZ_STATE, TOR_OTHER_CZ_FIPS, TO...
```

```
##
```

```
## 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.
```

PART 1: Damage From Storms

a) State Level Choropleth Maps

```
# Using maps package, which has geographic information on all U.S. states
us.states <- map_data("state") %>%
  as_tibble(.) %>%
  dplyr::rename(state = region) %>%
  select(-subregion) %>%
  mutate(state = str_to_title(state))

# Add State Abbreviations and Centers
statenames <- as_tibble(
  cbind(state=state.name, state.abb = state.abb,
        state.center.x = state.center$x,
        state.center.y = state.center$y))
statenames #has state, state.abb, state.center.x and state.center.y
```

```
## # A tibble: 50 × 4
##   state      state.abb state.center.x state.center.y
##   <chr>      <chr>      <chr>      <chr>
## 1 Alabama    AL        -86.7509    32.5901
## 2 Alaska     AK        -127.25     49.25
## 3 Arizona    AZ        -111.625    34.2192
## 4 Arkansas   AR        -92.2992    34.7336
## 5 California CA        -119.773    36.5341
## 6 Colorado   CO        -105.513    38.6777
## 7 Connecticut CT        -72.3573    41.5928
## 8 Delaware   DE        -74.9841    38.6777
## 9 Florida    FL        -81.685     27.8744
## 10 Georgia   GA        -83.3736    32.3329
## # i 40 more rows
```

```
statenames <- statenames %>% mutate_each_(funs(as.numeric), #changing centers to numeric
                                           vars=c("state.center.x", "state.center.y"))
```

```
## Warning: `mutate_each()` was deprecated in dplyr 0.7.0.  
## i Please use `across()` instead.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## i Please use a list of either functions or lambdas:  
##  
## # Simple named list: list(mean = mean, median = median)  
##  
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##  
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
us.states <- left_join(us.states, statenames)
```

```
## Joining with `by = join_by(state)`
```

```
us.states #took us.states and added state abb state center x and y
```

```
## # A tibble: 15,537 × 8
##   long  lat group order state  state.abb state.center.x state.center.y
##   <dbl> <dbl> <dbl> <int> <chr>   <chr>          <dbl>          <dbl>
## 1 -87.5  30.4     1     1 Alabama AL          -86.8          32.6
## 2 -87.5  30.4     1     2 Alabama AL          -86.8          32.6
## 3 -87.5  30.4     1     3 Alabama AL          -86.8          32.6
## 4 -87.5  30.3     1     4 Alabama AL          -86.8          32.6
## 5 -87.6  30.3     1     5 Alabama AL          -86.8          32.6
## 6 -87.6  30.3     1     6 Alabama AL          -86.8          32.6
## 7 -87.6  30.3     1     7 Alabama AL          -86.8          32.6
## 8 -87.6  30.3     1     8 Alabama AL          -86.8          32.6
## 9 -87.7  30.3     1     9 Alabama AL          -86.8          32.6
## 10 -87.8  30.3     1    10 Alabama AL          -86.8          32.6
## # i 15,527 more rows
```

```
us.states <- us.states %>% #making title column all caps for easier merging
  rename(STATE = state)
us.states$STATE <- toupper(us.states$STATE) #making states all caps for easier merging

#now I will prepare a simple dataset with total money damage by state
storm_data_filtered <- select(storm_data, STATE, DAMAGE_PROPERTY_USD, DAMAGE_CROPS_USD)
storm_data_filtered <- storm_data_filtered %>%
  mutate(money_damage = DAMAGE_PROPERTY_USD + DAMAGE_CROPS_USD)
state_money_damages <- select(storm_data_filtered, -DAMAGE_PROPERTY_USD, -DAMAGE_CROPS_USD)

#merging sets
moneydamages.merged=left_join(state_money_damages, us.states, by='STATE')
```

```
## Warning in left_join(state_money_damages, us.states, by = "STATE"): Detected an unexpected many-to-many relationship between `x` and `y`.
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 10399 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
moneydamages.merged
```



```
## # A tibble: 137,962,685 × 9
##   STATE      money_damage long   lat group order state.abb state.center.x
##   <chr>          <dbl> <dbl> <dbl> <dbl> <int> <chr>          <dbl>
## 1 NEW JERSEY      NA -74.7  41.4   32  8765 NJ          -74.2
## 2 NEW JERSEY      NA -74.4  41.2   32  8766 NJ          -74.2
## 3 NEW JERSEY      NA -74.2  41.1   32  8767 NJ          -74.2
## 4 NEW JERSEY      NA -74.2  41.1   32  8768 NJ          -74.2
## 5 NEW JERSEY      NA -73.9  41.0   32  8769 NJ          -74.2
## 6 NEW JERSEY      NA -73.9  40.9   32  8770 NJ          -74.2
## 7 NEW JERSEY      NA -73.9  40.9   32  8771 NJ          -74.2
## 8 NEW JERSEY      NA -73.9  40.9   32  8772 NJ          -74.2
## 9 NEW JERSEY      NA -74.0  40.8   32  8773 NJ          -74.2
## 10 NEW JERSEY     NA -74.0  40.8   32  8774 NJ          -74.2
## # i 137,962,675 more rows
## # i 1 more variable: state.center.y <dbl>
```

```

# First, I will replace all NAs with zeros in the "money_damage" column
# Next, group by the "STATE" column and summarize the "money_damage" variable so we see sum of total $ damage
# for that state between 2017 and 2022, for all weather events combined.
moneydamages.merged <- moneydamages.merged %>%
  mutate(money_damage = replace_na(money_damage, 0))
summarized_statedamages <- moneydamages.merged %>%
  group_by(STATE) %>%
  summarise(total_money_damage = sum(money_damage))

#now I'll merge the total money damage for each state with state info from us.states
merged_final_statedamages = left_join(summarized_statedamages, us.states, by='STATE')

#Setting format for my labels so it'll show $ in million, billion, trillion, etc.in the legend
label_format <- function(x) {
  ifelse(x >= 1e12, paste0(format(x / 1e12, scientific = FALSE), " trillion"),
    ifelse(x >= 1e9, paste0(format(x / 1e9, scientific = FALSE), " billion"),
      ifelse(x >= 1e6, paste0(format(x / 1e6, scientific = FALSE), " million"),
        format(x, scientific = FALSE))))
}

# Creating a plot with darkest purple being most money spent and light pink being least money spent
ggplot(merged_final_statedamages,
  aes(x = long, y = lat, group = group, label = state.abb, fill = total_money_damage)) +
  geom_polygon(color = "white") +
  scale_fill_gradientn(colours = c("pink", "purple4"),
    name = "Total Monetary Damage",
    labels = label_format) + # Using label formatting function
  ggtitle("Total Storm-Related Monetary Damage by State 2017-2022") +
  geom_text(aes(x = state.center.x, y = state.center.y, label = state.abb), color = "white", size = 3, inherit.aes = FALSE)
+
  theme_map() +
  theme(
    text = element_text(family = "Times New Roman"), # Change font to TNR
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold")) + # Centered title, size 16, bold
  coord_map(projection = "mercator") +
  guides(fill = guide_colorbar(barwidth = 2, barheight = 5, # Adjust bar width and height
    title.position = "top",
    label.position = "left")) # Adjust legend label position

```

```
## Warning: Removed 28 rows containing missing values (`geom_text()`).
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family  
## not found in Windows font database
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family  
## not found in Windows font database
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family  
## not found in Windows font database
```

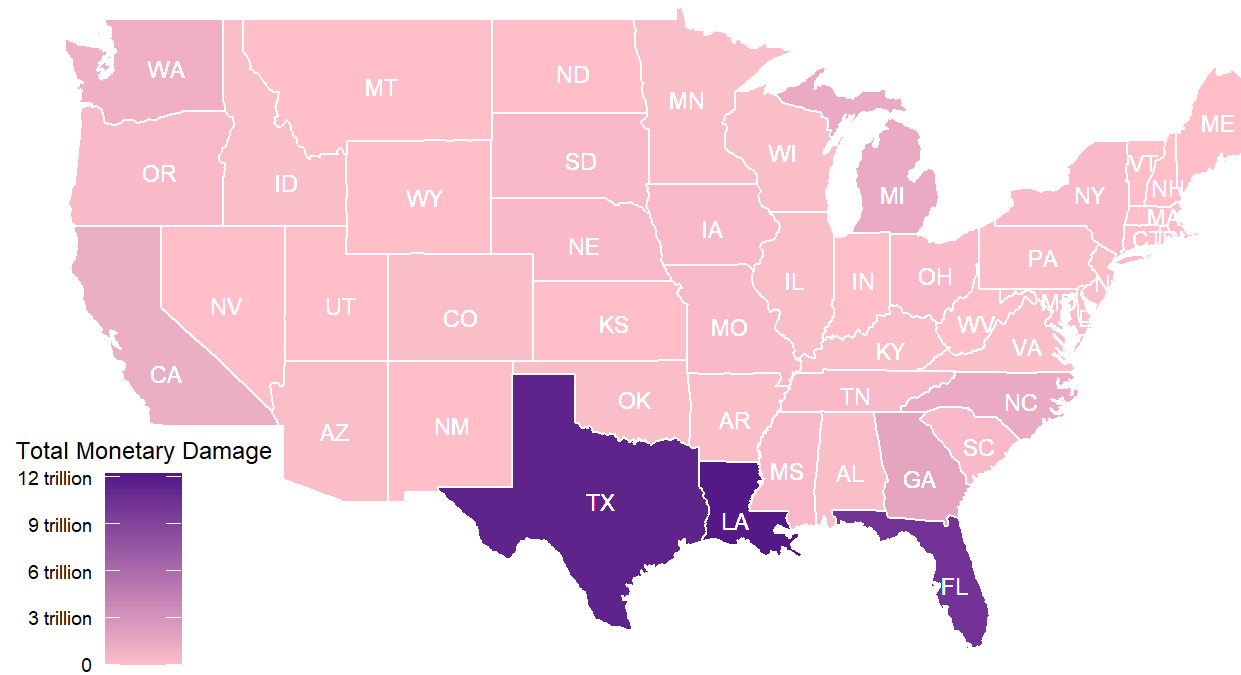
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database
```

Total Storm-Related Monetary Damage by State 2017-2022



b) County Choropleth Maps

```
#Starting by getting counties map and info
uscounties_sf <- get_urbn_map("counties", sf = TRUE)
# changing county fips column name in uscounties_sf for ease of merging later
colnames(counties)[colnames(counties) == "county_name"] <- "CZ_NAME"
counties$CZ_NAME <- toupper(counties$CZ_NAME)

#now preparing simple dataset with money damage by county (for each incident)
storm_data_counties <- filter(storm_data, CZ_TYPE == "C") #only grabbing events that happened in counties
storm_data_filtered2 <- select(storm_data_counties, CZ_NAME, DAMAGE_PROPERTY_USD, DAMAGE_CROPS_USD)
storm_data_filtered2 <- storm_data_filtered2 %>%
  mutate(money_damage2 = DAMAGE_PROPERTY_USD + DAMAGE_CROPS_USD)
county_money_damages <- select(storm_data_filtered2, -DAMAGE_PROPERTY_USD, -DAMAGE_CROPS_USD)
# adding "COUNTY" at ends of county names to prep for merging
county_money_damages$CZ_NAME <- paste(county_money_damages$CZ_NAME, "COUNTY", sep = " ")

# Now I will join the counties info
moneydamages.merged2 <- left_join(county_money_damages, counties, by = 'CZ_NAME')
```

```
## Warning in left_join(county_money_damages, counties, by = "CZ_NAME"): Detected an unexpected many-to-many relationship between `x` and `y`.
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 139360 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship = "many-to-many"` to silence this warning.
```

```

# First, I will replace all NAs with zeros in the "money_damage2" column
#Next, I'll group by the "CZ_NAME" column and summarize the "money_damage2" variable so we see sum of total $
#damage for that county between 2017 and 2022 (combined from all the events)
moneydamages.merged2 <- moneydamages.merged2 %>%
  mutate(money_damage2 = replace_na(money_damage2, 0))
summarized_countydamages <- moneydamages.merged2 %>%
  group_by(CZ_NAME) %>%
  summarise(total_money_damage2 = sum(money_damage2))

#merging back total $ damages for each county with all the county info
merged_final_countydamages = left_join(summarized_countydamages, counties, by='CZ_NAME')

# Create the same plot as above but for counties. The darkest purple means most money spent and light pink means least money
spent
ggplot(merged_final_countydamages,
  aes(x = long, y = lat, group = group, fill = total_money_damage2)) +
  geom_polygon(color = "white") +
  scale_fill_gradientn(colours = c("pink", "purple4"),
    name = "Total Monetary Damage",
    labels = label_format) + # Use custom formatting function
  ggtitle("Total Storm-Related Monetary Damage by County 2017-2022") +
  theme_map() +
  theme(
    text = element_text(family = "Times New Roman"), # Change font to TNR
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold")) + # Centered title, size 16, bold
  coord_map(projection = "mercator") +
  guides(fill = guide_colorbar(barwidth = 2, barheight = 5, # Adjust bar width and height
    title.position = "top",
    label.position = "left")) # Adjust Legend Label position

```

```

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

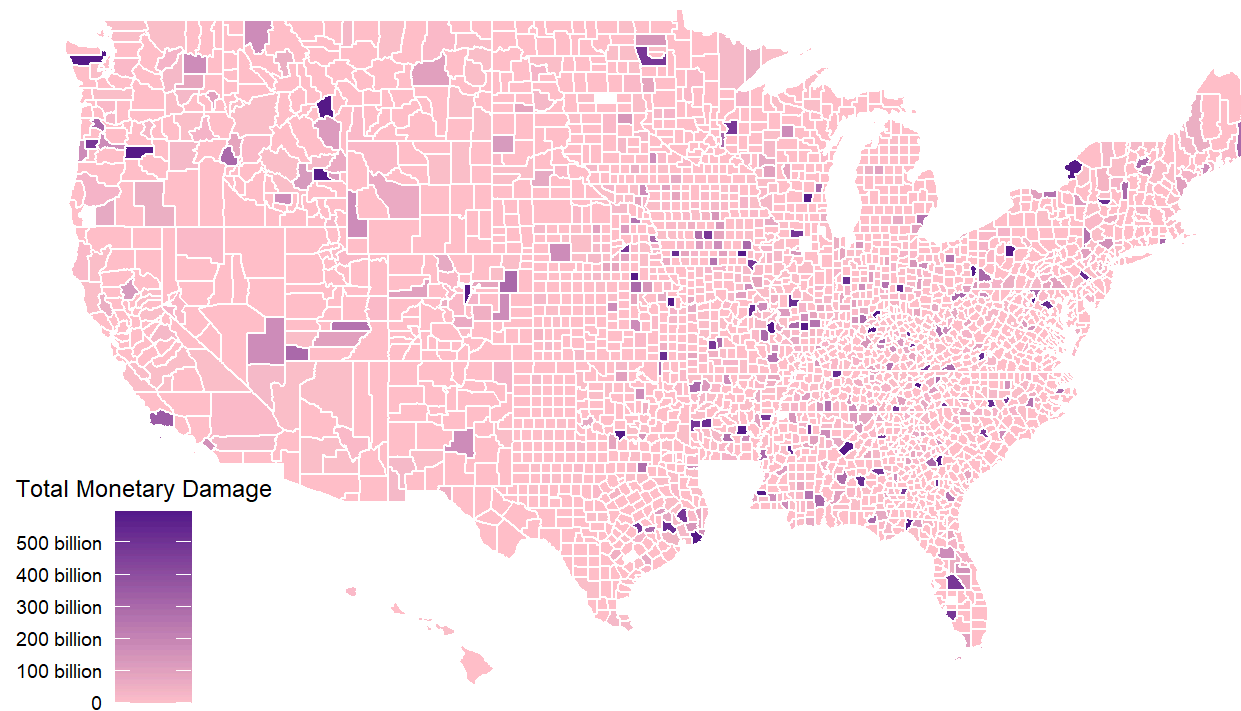
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

Total Storm-Related Monetary Damage by County 2017-2022



c) Density Map

Below, I will map deaths associated with storms, creating a density map with each point representing a severe weather event that resulted in at least one death. The darker circles will represent more deaths caused by that storm and lighter circles will represent less deaths. I will combine both direct and indirect deaths to create a total deaths variable. I believe the county-level monetary damage map (part b above) creates the best visualization of the distribution of destructive effects of the storms. The density map I create below is helpful because it pinpoints each event that results in at least one death and gives us an idea of how deadly each of these events was. However, I did filter out all the incidents where there were no deaths, so this portrays only a fraction of the event data. The state map in part a is a little too broad but the counties map gives a comprehensive summary of monetary damage for smaller areas of land so it allows us to zoom in a bit more. It also seems monetary damage is more common than deaths so we are able to use more data in that map to get a picture of the variation between counties.


```

#creating simple dataset with total deaths, lat and lon, state name
storm_data_filtered3 <- mutate(storm_data, total_deaths = DEATHS_DIRECT + DEATHS_INDIRECT)
storm_data_filtered3 <- select(storm_data_filtered3, BEGIN_LAT, BEGIN_LON, total_deaths, STATE)

# Filter out rows with missing latitude or longitude values and where there were zero deaths
storm_data_filtered3 <- storm_data_filtered3[complete.cases(storm_data_filtered3$BEGIN_LAT, storm_data_filtered3$BEGIN_LON),
]
storm_data_filtered3 <- filter(storm_data_filtered3, total_deaths != 0)

#I am only going to plot deaths on my map (in continental U.S.) So, I define the values I want to filter out
#(those not on map of continental US aka filtering out non-first-48 states)
states_to_exclude <- c("PUERTO RICO", "ATLANTIC SOUTH", "ATLANTIC NORTH", "GULF OF MEXICO", "LAKE MICHIGAN", "ALASKA", "HAWAII", "GUAM", "LAKE ERIE")

# Filter out rows where the "state" column equals the specified states to exclude
storm_data_filtered3_continental <- filter(storm_data_filtered3, !(STATE %in% states_to_exclude))

# Load state map data
us.states <- map_data("state")

# Create the plot with state boundaries and storm data points
ggplot() +
  geom_polygon(data = us.states, aes(x = long, y = lat, group = group), color = "grey", fill = "white") +
  geom_point(data = storm_data_filtered3_continental, aes(x = BEGIN_LON, y = BEGIN_LAT, color = total_deaths)) +
  geom_text(data = merged_final_statedamages, aes(x = state.center.x, y = state.center.y, label = state.abb), color = "black", size = 3) +
  scale_color_gradient(low = "lightgreen", high = "darkgreen", name = "Total Deaths") +
  ggtitle("Density Plotting of Death-Causing Storms") +
  theme_map() +
  theme(
    text = element_text(family = "Times New Roman"), # Change font to TNR
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    panel.background = element_rect(fill = "white"), # Set background color to white
    panel.border = element_blank(), # Remove panel border
    panel.grid.major = element_blank(), # Remove major gridlines
    panel.grid.minor = element_blank(), # Remove minor gridlines
    legend.position = c(0.015, 0.015)) + # Adjust legend position (x, y)
  coord_map(projection = "mercator") +

```

```
guides(fill = guide_colorbar(barwidth = 2, barheight = 5, # Adjust bar width and height
                             title.position = "top",
                             label.position = "left")) # Adjust legend label position
```

```
## Warning: Removed 28 rows containing missing values (`geom_text()`).
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

Density Plotting of Death-Causing Storms

