Assessing the Impact of Severe Weather Events on Public Health and the Economy

An Analysis using NOAA Storm Data

Saurabh Kelkar

2024-09-29

Contents

Introduction

Severe weather events, such as hurricanes, floods, and droughts, have widespread and far-reaching impacts. Economically, they cause extensive damage to infrastructure and agriculture, leading to business disruptions, increased insurance costs, and significant financial losses [1]. The impact on human health is equally severe, as they lead to injuries, fatalities, and heightened mental health challenges. Furthermore, waterborne and vector-borne diseases often surge in the aftermath of floods. These events also displace populations, especially in marginalized communities that lack the resources to recover quickly.

The summer of 2023 was marked by unprecedented extreme weather events, including record heat, wildfires, and intense storms across North America. This season was noted as the hottest on record for the Northern Hemisphere, exacerbating conditions for severe weather [2]. High temperatures also led to increased rainfall intensity and flooding events in various regions [3].

This project investigates severe weather events in the United States. The primary objective is to determine which types of weather events have the greatest impact on public health and the economy. This may help allocate resources more effectively and enhance preparedness strategies to mitigate future risks. Understanding these patterns is crucial for improving response measures and reducing the overall consequences of severe weather on communities.

Data and Methodology

Region

The analysis focuses on five US states: Colorado, Florida, Idaho, Nevada, and Utah, which *US News* ranked as the *top 5 states of the US Economy sector* in 2024 [4]. The rankings take into account various surveys on each state's business environment, labor market, and overall economic growth conducted from 2020.

Severe Event Data: NOAA Storm Events Database

The data for the top 5 states, Colorado, Florida, Idaho, Nevada, and Utah, for all events from June 1 to June 30, 2023, was downloaded in `CSV` format from the NOAA Storm Events Database [5].

The NOAA Storm Events Database contains the records on:

1. The occurrence of storms and other significant weather phenomena having sufficient intensity to cause loss of life, injuries, significant property damage, and disruption to commerce.
2. Rare, unusual weather phenomena that generate media attention.
3. Other significant meteorological events, such as record maximum or minimum temperatures or precipitation in connection with another event.

The database contains data from **January 1950 to June 2024** and is regularly updated by NOAA's National Weather Service (NWS). For this analysis, I have focused specifically on **June 2023**, as it marks the beginning of the summer season, a period when the US typically experiences more severe weather events due to intense weather fronts. Moreover, **June 2023** represents the most recent complete data for the start of the US summer season, because the data for **June 2024** was still being recorded during the data analysis.

R Packages for Analysis and Visualization

I will use multiple packages to process and analyze data. For data handling and cleaning, functions from `dplyr`, `readr`, and `VIM` will be used. Data visualization will be carried out with the `ggplot2` library. I will also write some functions to speed up the analysis.

```r
suppressPackageStartupMessages({
  suppressWarnings({
    library(dplyr)
    library(readr)
    library(VIM)
    library(ggplot2)
  })
})
```

Merging Data

Each state has an individual `CSV` file, so merging them will make the data analysis and processing smooth.

```r
# Get a list of all CSV files in the directory
file_list <- list.files("D:/Mini_projects/Project_2", pattern = "*.csv", full.names = TRUE)

# Read, modify, and combine all CSV files into one data frame
df <- file_list %>%
  lapply(read_csv, show_col_types = FALSE) %>%  # Read each CSV file and add the state column
  bind_rows()  # Combine them by rows

# Glimpse the combined data
glimpse(df)
```

```
## Rows: 976
## Columns: 39
## $ EVENT_ID            <dbl> 1106772, 1114089, 1092538, 1114091, 1095326, 11067~
## $ CZ_NAME_STR         <chr> "ELBERT CO.", "KIT CARSON CO.", "PITKIN CO.", "YUM~
## $ BEGIN_LOCATION      <chr> "FONDIS", "BURLINGTON", "(ASE)ASPEN ARPT", "YUMA",~
## $ BEGIN_DATE          <chr> "06/01/2023", "06/01/2023", "06/02/2023", "06/02/2~
## $ BEGIN_TIME          <dbl> 1330, 1556, 1500, 1620, 1100, 1543, 1550, 1608, 16~
## $ EVENT_TYPE          <chr> "Hail", "Hail", "Flood", "Hail", "Flood", "Hail", ~
## $ MAGNITUDE           <dbl> 1.00, 0.75, NA, 0.75, NA, 1.00, NA, 1.00, NA, 1.00~
## $ TOR_F_SCALE         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "EF0", NA,~
## $ DEATHS_DIRECT       <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ INJURIES_DIRECT     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ DAMAGE_PROPERTY_NUM <dbl> 0, 0, 5000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ DAMAGE_CROPS_NUM    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ STATE_ABBR          <chr> "CO", "CO", "CO", "CO", "CO", "CO", "CO", "CO", "C~
## $ CZ_TIMEZONE         <chr> "MST-7", "MST-7", "MST-7", "MST-7", "MST-7", "MST-~
## $ MAGNITUDE_TYPE      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ EPISODE_ID          <dbl> 181742, 182684, 179382, 182685, 179833, 181743, 18~
## $ CZ_TYPE             <chr> "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", ~
## $ CZ_FIPS             <dbl> 39, 63, 97, 125, 37, 35, 89, 5, 89, 5, 123, 101, 4~
## $ WFO                 <chr> "BOU", "GLD", "GJT", "GLD", "GJT", "BOU", "PUB", "~
## $ INJURIES_INDIRECT   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ DEATHS_INDIRECT     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ SOURCE              <chr> "Public", "Public", "Law Enforcement", "Trained Sp~
## $ FLOOD_CAUSE         <chr> NA, NA, "Heavy Rain / Snow Melt", NA, "Heavy Rain ~
## $ TOR_LENGTH          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.01, NA, ~
## $ TOR_WIDTH           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20, NA, NA~
## $ BEGIN_RANGE         <dbl> 3, 0, 4, 0, 0, 1, 2, 2, 2, 1, 3, 1, 2, 3, 4, 1, 6,~
## $ BEGIN_AZIMUTH       <chr> "N", "E", "W", "SSE", "SE", "E", "SW", "NNW", "SW"~
## $ END_RANGE           <dbl> 3, 0, 4, 0, 2, 1, 2, 2, 2, 1, 3, 1, 2, 3, 4, 1, 6,~
## $ END_AZIMUTH         <chr> "N", "E", "W", "SSE", "NW", "E", "SW", "NNW", "SW"~
## $ END_LOCATION        <chr> "FONDIS", "BURLINGTON", "(ASE)ASPEN ARPT", "YUMA",~
## $ END_DATE            <chr> "06/01/2023", "06/01/2023", "06/02/2023", "06/02/2~
## $ END_TIME            <dbl> 1330, 1556, 1600, 1620, 1100, 1543, 1551, 1608, 19~
## $ BEGIN_LAT           <dbl> 39.2600, 39.2990, 39.2107, 40.1191, 39.8653, 39.52~
## $ BEGIN_LON           <dbl> -104.4400, -102.2610, -106.9474, -102.7195, -106.8~
## $ END_LAT             <dbl> 39.2600, 39.2990, 39.2107, 40.1191, 39.8902, 39.52~
## $ END_LON             <dbl> -104.4400, -102.2610, -106.9478, -102.7195, -106.7~
## $ EVENT_NARRATIVE     <chr> NA, "Public report of 0.75 inch hail with visibili~
```

```
## $ EPISODE_NARRATIVE   <chr> "A severe thunderstorm produced hail up to quarter~
## $ ABSOLUTE_ROWNUMBER  <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
```

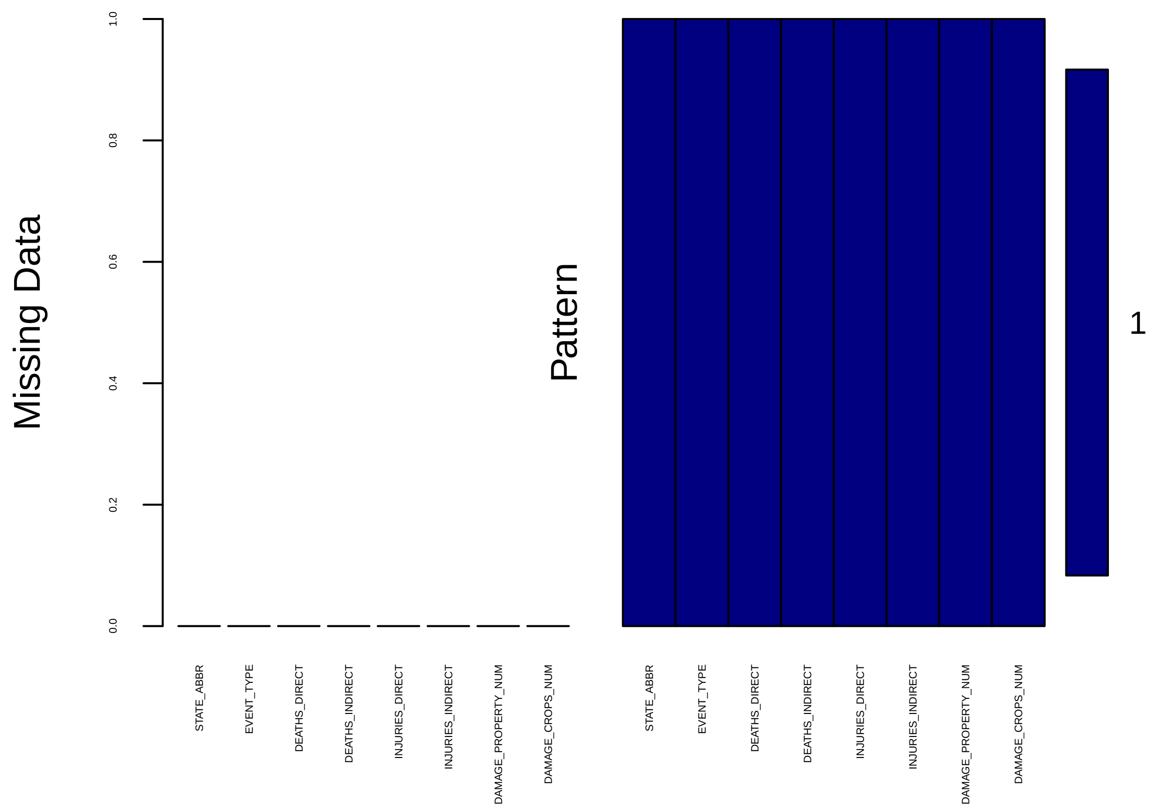
Pre-processing

Before proceeding to the impact estimations, data cleaning and handling missing values must be performed. First, the variables of interest are selected and strings for the names of severe events are re-formatted for consistency.

```
storm_data_clean <- df |>
  # Select the variables of interest
  select(STATE_ABBR, EVENT_TYPE, DEATHS_DIRECT, DEATHS_INDIRECT, INJURIES_DIRECT, INJURIES_INDIRECT, DAI
  # Convert the names in EVENT_TYPE to uppercase
  mutate(EVENT_TYPE = toupper(EVENT_TYPE))
```

Let's check the missing data using `aggr()` function from `VIM` package.

```
aggr(storm_data_clean, col = c('navyblue', 'yellow'), numbers = TRUE, sortVars = TRUE,
     labels = c("STATE_ABBR", "EVENT_TYPE", "DEATHS_DIRECT", "DEATHS_INDIRECT",
                "INJURIES_DIRECT", "INJURIES_INDIRECT", "DAMAGE_PROPERTY_NUM",
                "DAMAGE_CROPS_NUM"),
     cex.axis = 0.3, gap = 1, ylab = c("Missing Data", "Pattern"))
```




```
##
```

```
##  Variables sorted by number of missings:
##             Variable Count
##           STATE_ABBR     0
##           EVENT_TYPE     0
##        DEATHS_DIRECT     0
##      DEATHS_INDIRECT     0
##      INJURIES_DIRECT     0
##    INJURIES_INDIRECT     0
##  DAMAGE_PROPERTY_NUM     0
##     DAMAGE_CROPS_NUM     0
```

Perfect! There seems to be no missing value.

Note: The `aggr()` function generates a plot that provides a comprehensive visual overview of missing data in a dataset. This plot is composed of two primary components. On the left side, a bar plot illustrates the percentage of missing values for each variable, where the height of each bar reflects the proportion of missing data in that specific variable, with the x-axis displaying the variable names. On the right, the missing value patterns are shown in a matrix-like representation. Each row in this matrix corresponds to a unique combination of missing values across variables, with the x-axis labeling the variables and the y-axis showing different missingness patterns. The color intensity of each cell indicates the frequency of that specific pattern, helping users easily detect and understand missing data structures within the dataset.

Wait a minute! The data seems to have a lot of `0` as seen in `glimpse(df)`. Entries with `0` can be confusing. In some cases, `0` may represent a lack of data or an event that did not yield any results. However, this can easily be tangled up with missing values (`NA`). If `0` is treated as a missing data, but in reality, if it is meaningful value (e.g. 0 m/s wind speed), it will obscure the analysis (e.g., mean, correlations, PDFs). Therefore, handling zeros is crucial to avoid misinterpretations.

Let's look at how many `0` there are.

```r
total_zeros <- storm_data_clean |>
  # Counting zeros
  summarise(across(everything(), ~ sum(. == 0, na.rm = TRUE))) |>
  sum()

print(paste("The total number of 0:", total_zeros))
```

```
## [1] "The total number of 0: 5726"
```

That is a lot of `0`. If I treat them as missing data, this dataset will become unusable. But, it is not like I can label them as meaningful values either. Therefore, I will only select rows with at least one non-zero entry to get some meaningful information.

```r
# Filtering data to select rows with at least one non-zero data entry.
storm_data_clean2 <- storm_data_clean |>
  filter(rowSums(across(where(is.numeric)) != 0) > 0)

# View the cleaned data
glimpse(storm_data_clean2)
```

```
## Rows: 123
## Columns: 8
## $ STATE_ABBR         <chr> "CO", "CO", "CO", "CO", "CO", "CO", "CO", "CO", "C~
## $ EVENT_TYPE         <chr> "FLOOD", "FLOOD", "FLASH FLOOD", "FLASH FLOOD", "F~
## $ DEATHS_DIRECT      <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,~
## $ DEATHS_INDIRECT    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ INJURIES_DIRECT    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 90, 0, 0, 0~
## $ INJURIES_INDIRECT  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ DAMAGE_PROPERTY_NUM <dbl> 5000, 0, 5000, 5000, 50000, 50000, 50000, 50000, 5~
## $ DAMAGE_CROPS_NUM   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 25000, 0, 0, 0, 0~
```

Result and Discussion

Assessing the Impact of Severe Events on Population Health

INJURIES (Direct and Indirect) and DEATHS (Direct and Indirect) are the primary variables in the storm event data, which indicate the state of population health during or post-severe event.

I will explore this data in three cases. To streamline the analysis and save time, I will create a function, `health_impact`, that can be used with different arguments, eliminating the need to rewrite the same code multiple times.

```r
health_impact <- function(...){
  storm_data_clean2 |>
    # Group the data according to the argument passed to it
    group_by(...) |>
    # Calculate total fatalities and total injuries
    summarize(total_fatalities = sum(DEATHS_DIRECT) + sum(DEATHS_INDIRECT),
              total_injuries = sum(INJURIES_DIRECT) + sum(INJURIES_INDIRECT)) |>
    mutate(total_health_impact = total_fatalities + total_injuries) |>
    # Arranging the values in descending order
    arrange(desc(total_health_impact))
}
```

The `...` in the `function(...)` means I can pass any argument to it.

First, Let's check the health impact in each state, which can be examined by grouping the data by `STATE_ABBR`.

```r
# Total health impact due to all severe events in each state
impact1 <- health_impact(STATE_ABBR)
head(impact1)
```

```
## # A tibble: 4 x 4
##   STATE_ABBR total_fatalities total_injuries total_health_impact
##   <chr>                 <dbl>          <dbl>               <dbl>
## 1 CO                        4             95                  99
## 2 FL                       10              4                  14
## 3 ID                        0              9                   9
## 4 NV                        5              0                   5
```

If we want to estimate the health impact of each of the severe events, we just need to change the grouping method to `EVENT_TYPE`.

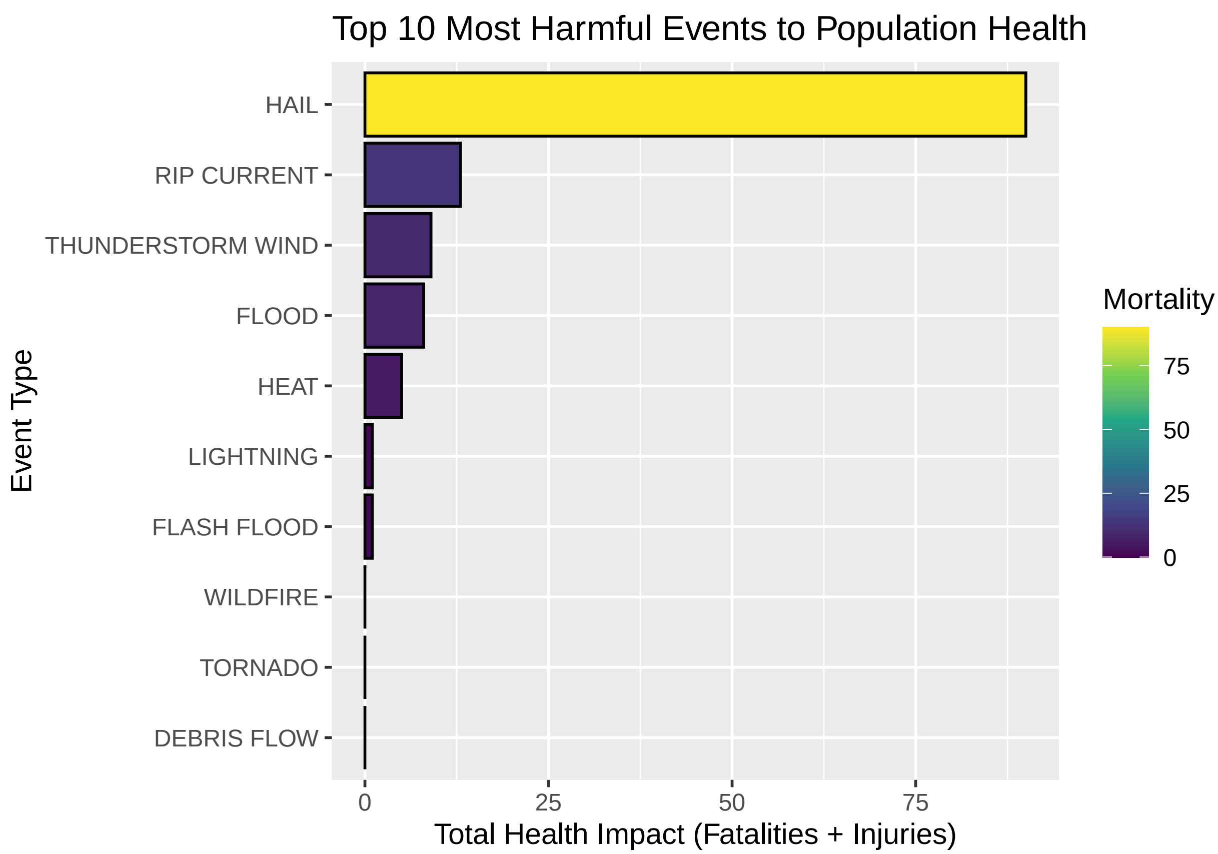
```r
# Total health impact due to each severe event from corresponding states
impact2 <- health_impact(EVENT_TYPE)
head(impact2)
```

```
## # A tibble: 6 x 4
##   EVENT_TYPE        total_fatalities total_injuries total_health_impact
##   <chr>                        <dbl>          <dbl>               <dbl>
## 1 HAIL                             0             90                  90
## 2 RIP CURRENT                     10              3                  13
## 3 THUNDERSTORM WIND                0              9                   9
## 4 FLOOD                            3              5                   8
## 5 HEAT                             5              0                   5
## 6 FLASH FLOOD                      1              0                   1
```

Whoa! That's a lot of fatalities due to the Hail event. Most of them are just injuries, though. Regardless, it was the most devastating event in our data.

Let's visualize it.

```r
ggplot(impact2[1:10, ],
       aes(x = reorder(EVENT_TYPE, total_health_impact),
           y = total_health_impact,
           fill = total_health_impact)) +
  geom_bar(stat = "identity", color = "black") +
  coord_flip() +
  scale_fill_viridis_c(option = "viridis", name = "Mortality") +
  labs(title = "Top 10 Most Harmful Events to Population Health",
       x = "Event Type",
       y = "Total Health Impact (Fatalities + Injuries)")
```



Now, this is more visually appealing than just looking at some numbers on the screen. There is another way to examine the impact of severe events: showing the estimated impact with respect to both the State (`STATE_ABBR`) and Severe events (`EVENT_TYPE`). However, I will not go deep into it. I will show you how to perform estimations. Since I have written a function, it is very simple now. Remember, I defined the `health_impact` function using `...` to let it accept any arguments. Now, I can pass the `STATE_ABBR, EVENT_TYPE` argument together.

```r
# Total health impact in each state due to a particular severe event
impact3 <- health_impact(STATE_ABBR, EVENT_TYPE)
```

```
## 'summarise()' has grouped output by 'STATE_ABBR'. You can override using the
## '.groups' argument.
```

```r
head(impact3)
```

```
## # A tibble: 6 x 5
## # Groups:   STATE_ABBR [4]
##   STATE_ABBR EVENT_TYPE      total_fatalities total_injuries total_health_impact
##   <chr>      <chr>                      <dbl>          <dbl>               <dbl>
## 1 CO         HAIL                           0             90                  90
## 2 FL         RIP CURRENT                   10              3                  13
## 3 ID         THUNDERSTORM W~                0              9                   9
## 4 CO         FLOOD                          3              5                   8
## 5 NV         HEAT                           5              0                   5
## 6 CO         FLASH FLOOD                    1              0                   1
```

See! It's very easy! Using defined functions saves a lot of time. This is a part of Functional Programming, about which I will write another article later.

The analysis so far reveals that certain types of severe weather events are particularly detrimental to public health, causing a significant number of fatalities and injuries. The data shows that:

First, considering the five U.S. states examined—Colorado, Florida, Idaho, Nevada, and Utah—Colorado emerges as the state with the highest total impact, recording 99 cases. This was largely due to injuries, whereas Florida, despite having a lower total impact of 14, experienced the highest number of fatalities, with 10 deaths. On the other hand, Nevada and Idaho had relatively lower impacts, with Nevada reporting 5 cases, primarily fatalities, and Idaho recording 9 cases, mostly injuries. Utah, notably, did not report any health impacts during the period analyzed. It's important to note that this could either imply that severe weather events may not be severe enough to lead to mortality or there were no records for our study period.

From the perspective of different weather event types, hailstorms were responsible for the most significant health impacts, resulting in 90 injuries, though no fatalities were associated with these events. Meanwhile, heat events and flash floods had comparatively minor impacts, resulting in 5 and 1 total cases each, while thunderstorm winds and floods contributed to moderate impacts, with 9 and 8 total cases, respectively. Rip currents, however, caused 10 fatalities and 3 injuries, making them particularly deadly. A rip current is a strong, localized, and narrow flow of water that moves directly away from the shore [6]. It pulls anything in its path out to sea, and attempting to swim against it is often futile, resulting in fatality. It is similar to trying to go against the overwhelming force of a truck that has directly hit you.

Now that the health impacts of severe events are examined, it is time to explore another sector where they can significantly impact: the Economy. We will delve into the damages caused by these events and summarize the overall impact on the Economies of the four states.

Assessing the Impact of Severe Events on the Economy

From the Storm events database, there are two variables that represent the damages in USD ($): Property Damage (`DAMAGE_PROPERTY_NUM`) and Crop Damage (`DAMAGE_CROPS_NUM`). Since, data processing functions are the same, except the variables, as in the case of `health_impact`, I will write a function similar to `health_impact()`.

Writing a function `eco_impact()`.

```r
economic_imp <- function(...){
  storm_data_clean2 |>
    group_by(...) |>
    summarize(total_prop_damage = sum(DAMAGE_PROPERTY_NUM, na.rm = TRUE),
              total_crop_damage = sum(DAMAGE_CROPS_NUM, na.rm = TRUE),
              total_damage = (total_prop_damage + total_crop_damage)
              ) |>
    arrange(desc(total_damage))
}
```

Similar to the analysis for public health, I will follow steps to analyze Damage data.

First, let's check the damages in each state by grouping the data by `STATE_ABBR`.

```r
# Total economic impact due to all events in each state
eco_impact1 <- economic_imp(STATE_ABBR)
head(eco_impact1)
```

```
## # A tibble: 4 x 4
```

```
##   STATE_ABBR total_prop_damage total_crop_damage total_damage
##   <chr>                  <dbl>             <dbl>        <dbl>
## 1 ID                  14826000           1000000     15826000
## 2 CO                   5924000             45000      5969000
## 3 FL                    486950                 0       486950
## 4 NV                     10000                 0        10000
```

```r
# Total economic impact due to each severe event from corresponding states
eco_impact2 <- economic_imp(EVENT_TYPE)
head(eco_impact2)
```

```
## # A tibble: 6 x 4
##   EVENT_TYPE        total_prop_damage total_crop_damage total_damage
##   <chr>                         <dbl>             <dbl>        <dbl>
## 1 WILDFIRE                   14450000                 0     14450000
## 2 FLOOD                       3585000             20000      3605000
## 3 FLASH FLOOD                 2500000           1025000      3525000
## 4 THUNDERSTORM WIND            549950                 0       549950
## 5 TORNADO                      100000                 0       100000
## 6 HAIL                          52000                 0        52000
```

```r
# Total economic impact in each state due to a particular severe event
eco_impact3 <- economic_imp(STATE_ABBR, EVENT_TYPE)
```

```
## `summarise()` has grouped output by 'STATE_ABBR'. You can override using the
## `.groups` argument.
```

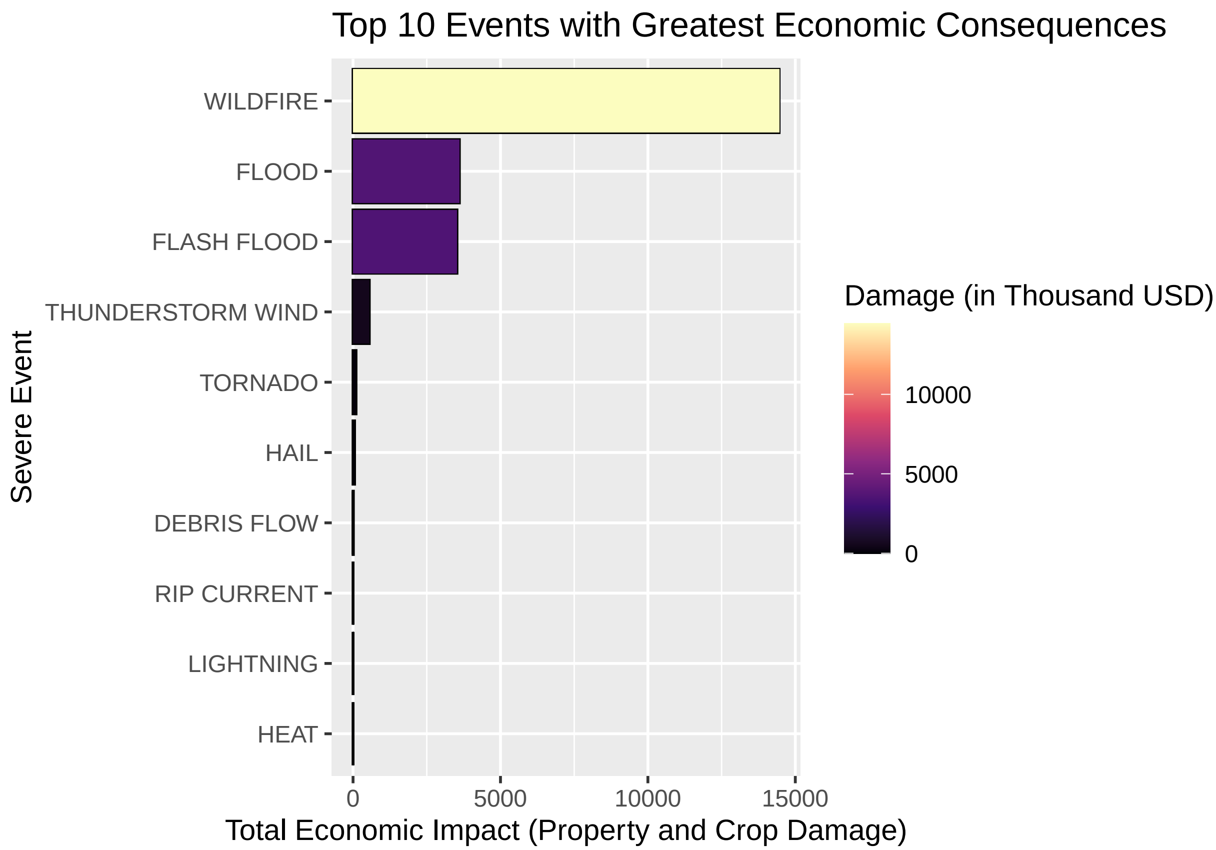
```r
head(eco_impact3)
```

```
## # A tibble: 6 x 5
## # Groups:   STATE_ABBR [3]
##   STATE_ABBR EVENT_TYPE        total_prop_damage total_crop_damage total_damage
##   <chr>      <chr>                         <dbl>             <dbl>        <dbl>
## 1 ID         WILDFIRE                    9200000                 0      9200000
## 2 CO         WILDFIRE                    5250000                 0      5250000
## 3 ID         FLOOD                       3420000                 0      3420000
## 4 ID         FLASH FLOOD                 2000000           1000000      3000000
## 5 CO         FLASH FLOOD                  500000             25000       525000
## 6 FL         THUNDERSTORM WIND            391950                 0       391950
```

While we can visualize all three of the cases, let's direct our attention to the visual representation of `eco_impact2`, the total economic impact due to each severe event from corresponding states.

```r
suppressMessages(
  ggplot(eco_impact2[1:10, ],
       aes(x = reorder(EVENT_TYPE, total_damage),
           y = total_damage/1000,
           fill = total_damage/1000)) +
  geom_bar(stat = "identity", color = "black") +
  coord_flip() +
  scale_fill_viridis_c(option = "magma", name = "Damage (in Thousand USD)") +
```

```r
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Events with Greatest Economic Consequences",
       x = "Severe Event",
       y = "Total Economic Impact (Property and Crop Damage)")
)
```



Just like the Wildfires, `ggplot` is on Fire!

The data reveals a comprehensive view of damages caused by severe weather events across different U.S. states and event types. From a state-level perspective, Idaho suffered the highest total damages, with approximately \$15.83 million in losses, driven largely by property damage from wildfires. Colorado experienced total damage of around \$5.97 million, mostly from wildfires and flash floods, while Florida and Nevada had much smaller impacts, with Florida reporting \$486,950 and Nevada a mere \$10,000 in damages.

From the perspective of event types, wildfires were the most destructive, causing a total of \$14.45 million in damages, all of which were property-related. Floods and flash floods also led to significant losses, with damages totaling \$3.61 million and \$3.53 million, respectively, though flash floods incurred higher crop damage than regular floods. Thunderstorm winds and tornadoes caused relatively lower damages, with thunderstorms accounting for \$549,950 in property damage and tornadoes for \$100,000. Once again, Utah reported no damages.

Summary

When analyzing the impact of severe events across states according to event types, Colorado stands out as disproportionately affected, particularly by hailstorms, which caused all 90 of the state's reported injuries. In addition, Colorado's health burden was exacerbated by floods and flash floods. Similarly, Idaho experienced significant impacts from thunderstorm winds, while Nevada's fatalities were primarily due to heat. On the other hand, Florida saw all its fatalities linked to rip currents, underscoring its vulnerability to coastal hazards.

In terms of damages, Idaho incurred the highest financial losses, with $9.2 million from wildfires, followed by $3.42 million from floods and $3 million from flash floods, the latter heavily affecting crops. Colorado also suffered significant wildfire-related damages, amounting to $5.25 million, alongside $525,000 in flash flood damages. Florida's losses were mostly attributed to thunderstorm winds. This data highlights the uneven distribution of weather impacts, particularly from wildfires, and emphasizes state-specific vulnerabilities, such as Idaho's susceptibility to multiple severe event types.

These findings underscore the need to focus public health resources and emergency response efforts on regions and times of the year most prone to severe weather events.

Reference

1. NOAA National Severe Storms Laboratory. Severe weather 101. https://www.nssl.noaa.gov/education/svrwx101/
2. Berman, N. (2023, August 7). The weather of summer 2023 was the most extreme yet. Council on Foreign Relations. https://www.cfr.org/article/weather-summer-2023-was-most-extreme-yet
3. Center for Climate and Energy Solutions. (2024, January 29). Extreme Weather and Climate Change - Center for Climate and Energy Solutions. https://www.c2es.org/content/extreme-weather-and-climate-change/
4. Best States of the US Economy sector. https://www.usnews.com/news/best-states/rankings/economy
5. NOAA Storm Events Database - https://www.ncdc.noaa.gov/stormevents/
6. NOAA's National Weather Service. RIP Current Science. https://www.weather.gov/safety/ripcurrent-science

---

Note: This project is written in `R Markdown` and kneaded into a document using `knitr`.
