Project 2

2024-04-12

1. Synopsis

This project aims to analyze the NOAA Storm Database and explore the effects of storms and weather events on both population and economy.

This analysis explores which types of events are most harmful on: - Health (injuries and fatalities) - Property and crops (economic consequences)

NOAA Documentation

2. Data Processing

2.1 Loading data

```
#install.packages("dplyr")
library(dplyr)

# Download dataset
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
dataset <- "storm_data.csv"
file <- paste("./", dataset, ".bz2", sep="")
download.file(url, destfile = file, method = "curl")
```

```
# Read dataset
df <- read.csv("./storm_data.csv.bz2")
head(df)
```

```
##   STATE__           BGN_DATE BGN_TIME TIME_ZONE COUNTY COUNTYNAME STATE  EVTYPE
## 1       1  4/18/1950 0:00:00     0130       CST     97     MOBILE    AL TORNADO
## 2       1  4/18/1950 0:00:00     0145       CST      3    BALDWIN    AL TORNADO
## 3       1  2/20/1951 0:00:00     1600       CST     57    FAYETTE    AL TORNADO
## 4       1   6/8/1951 0:00:00     0900       CST     89    MADISON    AL TORNADO
## 5       1 11/15/1951 0:00:00     1500       CST     43    CULLMAN    AL TORNADO
## 6       1 11/15/1951 0:00:00     2000       CST     77 LAUDERDALE    AL TORNADO
##   BGN_RANGE BGN_AZI BGN_LOCATI END_DATE END_TIME COUNTY_END COUNTYENDN
## 1         0                                               0         NA
## 2         0                                               0         NA
## 3         0                                               0         NA
## 4         0                                               0         NA
## 5         0                                               0         NA
## 6         0                                               0         NA
##   END_RANGE END_AZI END_LOCATI LENGTH WIDTH F MAG FATALITIES INJURIES PROPDMG
## 1         0                      14.0   100 3   0          0       15    25.0
## 2         0                       2.0   150 2   0          0        0     2.5
## 3         0                       0.1   123 2   0          0        2    25.0
## 4         0                       0.0   100 2   0          0        2     2.5
## 5         0                       0.0   150 2   0          0        2     2.5
## 6         0                       1.5   177 2   0          0        6     2.5
##   PROPDMGEXP CROPDMG CROPDMGEXP WFO STATEOFFIC ZONENAMES LATITUDE LONGITUDE
## 1          K       0                                         3040      8812
## 2          K       0                                         3042      8755
## 3          K       0                                         3340      8742
## 4          K       0                                         3458      8626
## 5          K       0                                         3412      8642
## 6          K       0                                         3450      8748
##   LATITUDE_E LONGITUDE_ REMARKS REFNUM
## 1       3051       8806              1
## 2          0          0              2
## 3          0          0              3
## 4          0          0              4
## 5          0          0              5
## 6          0          0              6
```

2.2 Variables available

```r
colnames(df)
```

```
##  [1] "STATE__"    "BGN_DATE"   "BGN_TIME"   "TIME_ZONE"  "COUNTY"    
##  [6] "COUNTYNAME" "STATE"      "EVTYPE"     "BGN_RANGE"  "BGN_AZI"   
## [11] "BGN_LOCATI" "END_DATE"   "END_TIME"   "COUNTY_END" "COUNTYENDN"
## [16] "END_RANGE"  "END_AZI"    "END_LOCATI" "LENGTH"     "WIDTH"     
## [21] "F"          "MAG"        "FATALITIES" "INJURIES"   "PROPDMG"   
## [26] "PROPDMGEXP" "CROPDMG"    "CROPDMGEXP" "WFO"        "STATEOFFIC"
## [31] "ZONENAMES"  "LATITUDE"   "LONGITUDE"  "LATITUDE_E" "LONGITUDE_"
## [36] "REMARKS"    "REFNUM"
```

2.3 Filtering variables

Selecting only variables of interest. Additionally, we are filtering the rows with data available.

```r
cols <- c("EVTYPE", "FATALITIES", "INJURIES", "PROPDMG", "PROPDMGEXP",
          "CROPDMG", "CROPDMGEXP")

df <- df %>%
    select(cols) %>%
    filter(EVTYPE != "?" & (INJURIES > 0 | FATALITIES > 0 | PROPDMG > 0 | CROPDMG > 0))
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(cols)
##
##   # Now:
##   data %>% select(all_of(cols))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

2.4 Converting numbers

Cleaning up the PROPDMGEXP and CROPDMGEXP columns to facilitate their use in calculating property and crop costs.

```r
# Change all damage exponents to uppercase
cols <- c("PROPDMGEXP", "CROPDMGEXP")
df <- df %>%
  mutate(across(all_of(cols), toupper))

# Map property damage alphanumeric exponents to numeric values
prop_dmg_key <-  c("\"\"" = 10^0,
                   "-" = 10^0,
                   "+" = 10^0,
                   "0" = 10^0,
                   "1" = 10^1,
                   "2" = 10^2,
                   "3" = 10^3,
                   "4" = 10^4,
                   "5" = 10^5,
                   "6" = 10^6,
                   "7" = 10^7,
                   "8" = 10^8,
                   "9" = 10^9,
                   "H" = 10^2,
                   "K" = 10^3,
                   "M" = 10^6,
                   "B" = 10^9)

# Map crop damage alphanumeric exponents to numeric values
crop_dmg_key <-  c("\"\"" = 10^0,
                   "?" = 10^0,
                   "0" = 10^0,
                   "K" = 10^3,
                   "M" = 10^6,
                   "B" = 10^9)

df <- df %>%
  mutate(PROPDMGEXP = prop_dmg_key[as.character(PROPDMGEXP)]) %>%
  mutate(PROPDMGEXP = ifelse(is.na(PROPDMGEXP), 10^0, PROPDMGEXP)) %>%
  mutate(CROPDMGEXP = crop_dmg_key[as.character(CROPDMGEXP)]) %>%
  mutate(CROPDMGEXP = ifelse(is.na(CROPDMGEXP), 10^0, CROPDMGEXP))
```

2.5 Calculating economic costs

```r
df <- df %>%
    mutate(prop_cost = PROPDMG * PROPDMGEXP,
           crop_cost = CROPDMG * CROPDMGEXP) %>%
    select(EVTYPE, FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, prop_cost,
           CROPDMG, CROPDMGEXP, crop_cost)
```

2.6 Calculating total cost by event

```r
total_cost <- df %>%
    group_by(EVTYPE) %>%
    summarise(prop_cost = sum(prop_cost),
              crop_cost = sum(crop_cost),
              total_cost = sum(prop_cost) + sum(crop_cost)) %>%
    arrange(desc(total_cost)) %>%
    slice_head(n = 10)

head(total_cost)
```

```
## # A tibble: 6 x 4
##   EVTYPE                prop_cost  crop_cost    total_cost
##   <chr>                     <dbl>      <dbl>         <dbl>
## 1 FLOOD             144657709807  5661968450 150319678257
## 2 HURRICANE/TYPHOON  69305840000  2607872800  71913712800
## 3 TORNADO            56947380676.  414953270  57362333946.
## 4 STORM SURGE        43323536000        5000  43323541000
## 5 HAIL               15735267513. 3025954473  18761221986.
## 6 FLASH FLOOD        16822673978. 1421317100  18243991078.
```

2.7 Calculating total fatalities and injuries by event

```r
total_injuries <- df %>%
    group_by(EVTYPE) %>%
    summarise(FATALITIES = sum(FATALITIES),
              INJURIES = sum(INJURIES),
              totals = sum(FATALITIES) + sum(INJURIES)) %>%
    arrange(desc(FATALITIES)) %>%
    slice_head(n = 10)

head(total_injuries)
```

```
## # A tibble: 6 x 4
##   EVTYPE         FATALITIES INJURIES totals
##   <chr>               <dbl>    <dbl>  <dbl>
## 1 TORNADO              5633    91346  96979
## 2 EXCESSIVE HEAT       1903     6525   8428
## 3 FLASH FLOOD           978     1777   2755
## 4 HEAT                  937     2100   3037
## 5 LIGHTNING             816     5230   6046
## 6 TSTM WIND             504     6957   7461
```

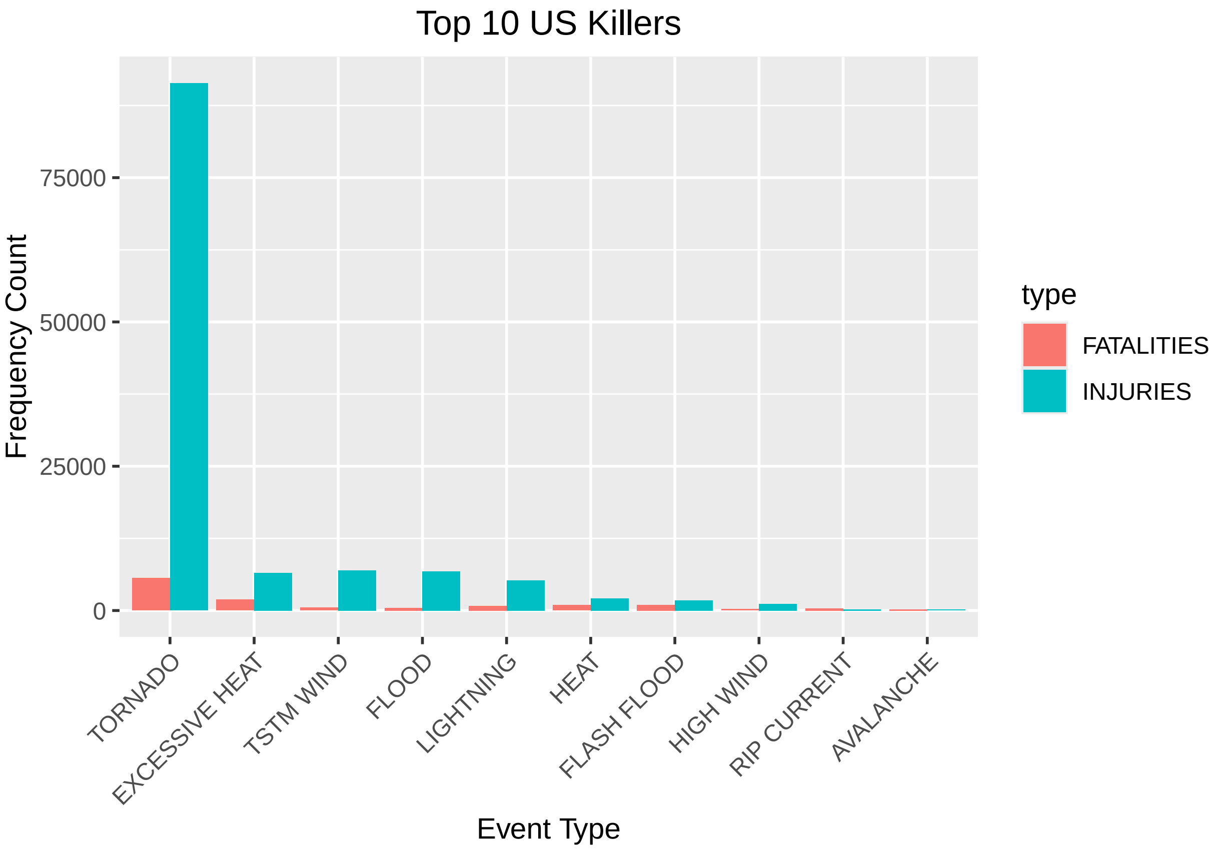
3. Results

3.1 Which types of events are most harmful to population health?

```r
#install.packages('tidyr')
library(tidyr)

df_harmful <- total_injuries %>%
    pivot_longer(cols = c("FATALITIES", "INJURIES"),
                 names_to = "type",
                 values_to = "count")

head(df_harmful)
```

```
## # A tibble: 6 x 4
##   EVTYPE         totals type       count
##   <chr>           <dbl> <chr>      <dbl>
## 1 TORNADO         96979 FATALITIES  5633
## 2 TORNADO         96979 INJURIES   91346
## 3 EXCESSIVE HEAT   8428 FATALITIES  1903
## 4 EXCESSIVE HEAT   8428 INJURIES    6525
## 5 FLASH FLOOD      2755 FATALITIES   978
## 6 FLASH FLOOD      2755 INJURIES    1777
```

```r
library("ggplot2")

# Create chart
health_chart <- ggplot(df_harmful, aes(x=reorder(EVTYPE, -count), y=count))

# Plot data as bar chart
health_chart = health_chart + geom_bar(stat="identity", aes(fill=type), position="dodge")

# Format y-axis scale and set y-axis label
health_chart = health_chart + ylab("Frequency Count")

# Set x-axis label
health_chart = health_chart + xlab("Event Type")

# Rotate x-axis tick labels
health_chart = health_chart + theme(axis.text.x = element_text(angle=45, hjust=1))

# Set chart title and center it
health_chart = health_chart + ggtitle("Top 10 US Killers") + theme(plot.title = element_text(hjust = 0.5

health_chart
```



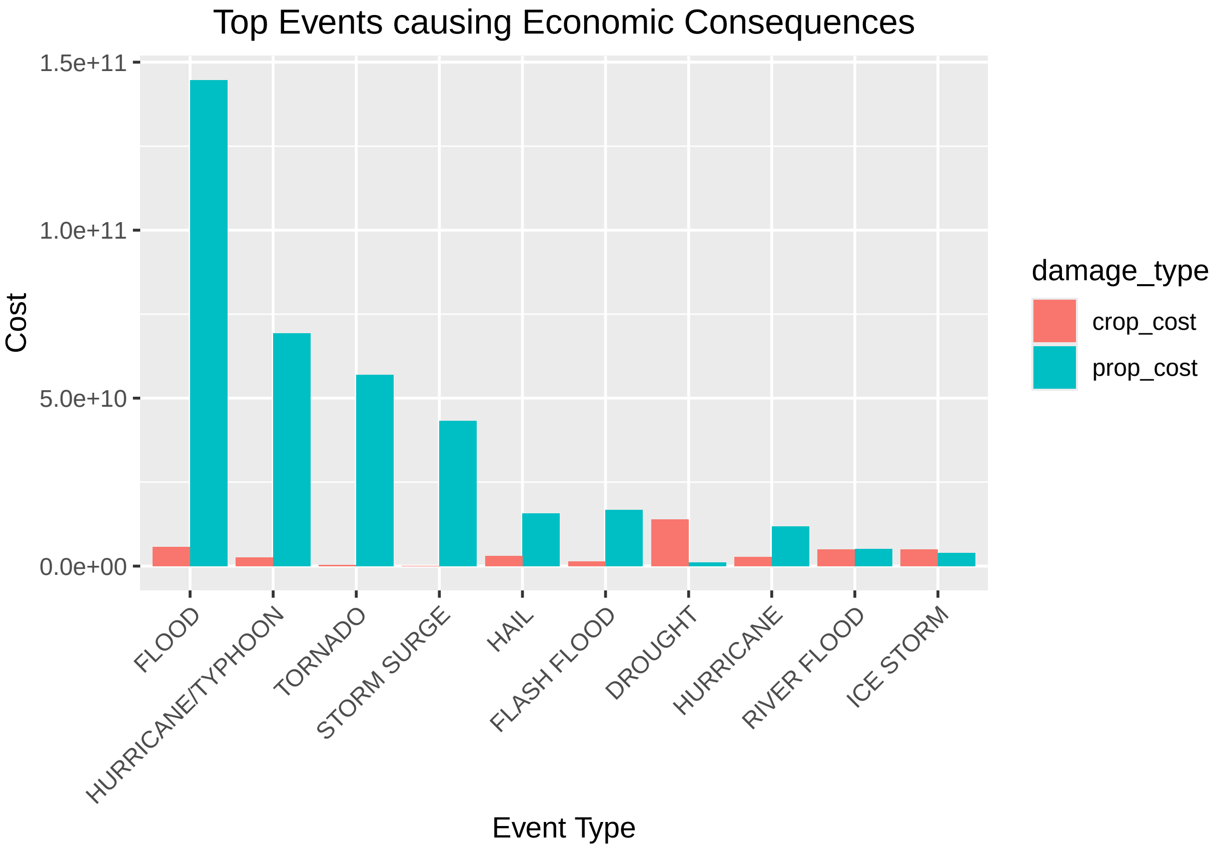
3.2 Which types of events have the greatest economic consequences?

```
df_econ <- total_cost %>%
    pivot_longer(cols = c("prop_cost", "crop_cost"),
                 names_to = "damage_type",
                 values_to = "cost")

head(df_econ)
```

```
## # A tibble: 6 x 4
##   EVTYPE               total_cost damage_type          cost
##   <chr>                     <dbl> <chr>               <dbl>
## 1 FLOOD             150319678257  prop_cost   144657709807
## 2 FLOOD             150319678257  crop_cost     5661968450
## 3 HURRICANE/TYPHOON  71913712800  prop_cost    69305840000
## 4 HURRICANE/TYPHOON  71913712800  crop_cost     2607872800
## 5 TORNADO            57362333946. prop_cost    56947380676.
## 6 TORNADO            57362333946. crop_cost      414953270
```

```
# Create chart
econ_chart <- ggplot(df_econ, aes(x=reorder(EVTYPE, -cost), y=cost))

# Plot data as bar chart
econ_chart = econ_chart + geom_bar(stat="identity", aes(fill=damage_type), position="dodge")

# Format y-axis scale and set y-axis label
econ_chart = econ_chart + ylab("Cost")
```

```r
# Set x-axis label
econ_chart = econ_chart + xlab("Event Type")

# Rotate x-axis tick labels
econ_chart = econ_chart + theme(axis.text.x = element_text(angle=45, hjust=1))

# Set chart title and center it
econ_chart = econ_chart + ggtitle("Top Events causing Economic Consequences") + theme(plot.title = elem

econ_chart
```


Top Events causing Economic Consequences
Cost
1.5e+11
1.0e+11
5.0e+10
0.0e+00
FLOOD
HURRICANE/TYPHOON
TORNADO
STORM SURGE
HAIL
FLASH FLOOD
DROUGHT
HURRICANE
RIVER FLOOD
ICE STORM
Event Type
damage_type
crop_cost
prop_cost