Reproducible Research: Peer Assessment 2

Reproducible Research: Peer Assessment 2

Storm and Weather Impacts on Health and Property

(USA 1996-2011)

David Modjeska

Synopsis

By way of introduction ...

```
## Warning: package 'knitr' was built under R version 3.2.5
```

Loading and processing the raw data

```r
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
```

```r
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.2.5
```

```r
library(readr)

storm_zip_file <- "StormData.csv.bz2"
storm_csv_file <- "StormData.csv"

col_classes <- c("iccciccncccicnccniciiincnccccicicn")
if (!file.exists(storm_zip_file)) {
    download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2",
                  storm_zip_file, method = "curl")
    storm_data <- read.csv(storm_zip_file, na = "")
    write_csv(storm_data, storm_csv_file)
    classes <- sapply(storm_data, class)
```

```r
} else {
    storm_data <- read_csv(storm_csv_file, na = "", col_types = col_classes)
}


dim(storm_data)
```

```
## [1] 902297     37
```

```r
# The storm data set contains 985 rows and 37 columns.

head(storm_data)
```

```
## Source: local data frame [6 x 37]
##
##   STATE__           BGN_DATE BGN_TIME TIME_ZONE COUNTY COUNTYNAME STATE
##     <int>              <chr>    <chr>     <chr>  <int>      <chr> <chr>
## 1       1  4/18/1950 0:00:00     0130       CST     97     MOBILE    AL
## 2       1  4/18/1950 0:00:00     0145       CST      3    BALDWIN    AL
## 3       1  2/20/1951 0:00:00     1600       CST     57    FAYETTE    AL
## 4       1   6/8/1951 0:00:00     0900       CST     89    MADISON    AL
## 5       1 11/15/1951 0:00:00     1500       CST     43    CULLMAN    AL
## 6       1 11/15/1951 0:00:00     2000       CST     77 LAUDERDALE    AL
## Variables not shown: EVTYPE <chr>, BGN_RANGE <dbl>, BGN_AZI <chr>,
##   BGN_LOCATI <chr>, END_DATE <chr>, END_TIME <chr>, COUNTY_END <int>,
##   COUNTYENDN <chr>, END_RANGE <dbl>, END_AZI <chr>, END_LOCATI <chr>,
##   LENGTH <dbl>, WIDTH <int>, F <chr>, MAG <int>, FATALITIES <int>,
##   INJURIES <int>, PROPDMG <dbl>, PROPDMGEXP <chr>, CROPDMG <dbl>,
##   CROPDMGEXP <chr>, WFO <chr>, STATEOFFIC <chr>, ZONENAMES <chr>, LATITUDE
##   <chr>, LONGITUDE <int>, LATITUDE_E <chr>, LONGITUDE_ <int>, REMARKS
##   <chr>, REFNUM <dbl>.
```

```r
# The columns names are all legal for data frames in R.

# Let's extract the columns that we'll be using in this analysis, to enhance performance.
storm_data_trim <- select(storm_data, BGN_DATE, EVTYPE, FATALITIES, INJURIES, PROPDMG,
                          PROPDMGEXP, CROPDMG, CROPDMGEXP)

# Let's make these column names easier to read.
names(storm_data_trim) <- c("Begin_Date", "Event_Type", "Fatalities", "Injuries",
                            "Property_Damage", "Property_Damage_Exp", "Crop_Damage",
                            "Crop_Damage_Exp")

# The date from 1996 onward is more reliable and complete than earlier data, so let's use that.
# Ref 1: http://www.ncdc.noaa.gov/stormevents/details.jsp
# Ref 2: https://ire.org/media/uploads/files/datalibrary/samplefiles/Storm%20Events/readme_08.doc

library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.2.4

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

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

```
first_year <- 1996
storm_data_dates <- storm_data_trim %>%
    mutate(Date_Time = mdy_hms(Begin_Date)) %>%
    mutate(Year = year(Date_Time)) %>%
    filter(Year >= first_year) %>%
    select(-Date_Time, -Begin_Date)
year_range <- max(storm_data_dates$Year) - first_year

# Let's see how many missing values there are in the columns that we'll be using.
data.frame(Event_Type = sum(is.na(storm_data_dates$Event_Type)),
           Fatalities = sum(is.na(storm_data_dates$Fatalities)),
           Injuries = sum(is.na(storm_data_dates$Injuries)),
           Property_Damage = sum(is.na(storm_data_dates$Property_Damage)),
           Property_Damage_Exp = sum(is.na(storm_data_dates$Property_Damage_Exp)),
           Crop_Damage = sum(is.na(storm_data_dates$Crop_Damage)),
           Crop_Damage_Exp = sum(is.na(storm_data_dates$Crop_Damage_Exp)))
```

```
##   Event_Type Fatalities Injuries Property_Damage Property_Damage_Exp
## 1          0          0        0               0                   0
##   Crop_Damage Crop_Damage_Exp
## 1           0               0
```

```
# There are no missing values in these columns, fortunately.

## Let's remove observations with no storm impacts, since they won't affect this analysis.
storm_data_damage <- storm_data_dates %>%
    filter(Injuries > 0 | Fatalities > 0 | Property_Damage > 0 | Crop_Damage > 0)

# Now, we should combine the exponents with the damage values
# Assume: M == m, and digits equal exponents
exp_function <- function(exp_letter) {
    return(ifelse(exp_letter == "B", 1000000000,
                  ifelse(exp_letter == "M" | exp_letter == 'm', 1000000,
                         ifelse(exp_letter == "K", 1000,
                                ifelse(exp_letter == "H" | exp_letter == 'h', 100,
                                       ifelse(is.finite(exp_letter),
                                              10 ^ as.numeric(exp_letter),
                                              1))))))
}
storm_data_exp <- storm_data_damage %>%
    mutate(Property_Damage = Property_Damage * exp_function(Property_Damage_Exp)) %>%
    mutate(Crop_Damage = Crop_Damage * exp_function(Crop_Damage_Exp)) %>%
    select(-Property_Damage_Exp, -Crop_Damage_Exp)

# Let's adjust for inflation. The data come from the U.S. Bureau of Labor Statistics,
# via the Federal Reserve Bank of St. Louis.
# https://research.stlouisfed.org/fred2/series/CPIAUCSL/downloaddata
# On this page, choose "Index 1982-84=100", Annual, Average, 1947-01-01 to 2015-04-01, Excel,
# and then click the Download Data button. Save the file in the current working directory.
```

```r
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: xlsxjars
```

```r
cpi_data <- read.xlsx("CPIAUCSL.xls", sheetIndex = 1, startRow = 55, endRow = 123) %>%
    rename(Date = DATE, Value = VALUE) %>%
    mutate(Date = ymd(Date)) %>%
    mutate(Year = year(Date)) %>%
    select(-Date)
value_2014 <- filter(cpi_data, Year == 2014) %>% select(Value)

storm_data_inflated <- storm_data_exp %>%
    inner_join(cpi_data, by = "Year") %>%
    mutate(Property_Damage = (Property_Damage * (value_2014$Value/Value))) %>%
    mutate(Crop_Damage = Crop_Damage * (value_2014$Value/Value)) %>%
    select(-Year)

# Let's have a quick look at the summary statistics for each output variable.

summary(storm_data_inflated$Fatalities)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   0.00000   0.00000   0.00000   0.04337   0.00000 158.00000
```

```r
summary(storm_data_inflated$Injuries)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    0.000    0.000    0.288    0.000 1150.000
```

```r
summary(storm_data_inflated$Property_Damage)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 2.632e+03 1.052e+04 2.216e+06 3.630e+04 1.351e+11
```

```r
summary(storm_data_inflated$Crop_Damage)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 0.000e+00 0.000e+00 2.235e+05 0.000e+00 1.830e+09
```

```r
# How many unique event types are contained in the data set?
length(unique(storm_data_inflated$Event_Type))
```

```
## [1] 219
```

```r
# There are 985 distinct event types! Let's clean up this list.

# First, let's perform some general lexical cleanup.
storm_data_clean_1 <- storm_data_inflated %>%
    filter(!grepl("summary", Event_Type, ignore.case = TRUE)) %>%
    mutate(Event_Type = tolower(Event_Type)) %>%
    mutate(Event_Type = gsub("^[[:space:]]*", "", Event_Type)) %>%
    mutate(Event_Type = gsub("  [[:space:]]*", " ", Event_Type)) %>%
    mutate(Event_Type = gsub("[[:space:]]?mph$", "", Event_Type)) %>%
    mutate(Event_Type = gsub(" f[[:digit:]]$", "", Event_Type)) %>%
    mutate(Event_Type = gsub(" [[:punct:]]*g[[:digit:]]*[[:punct:]]*$", "", Event_Type)) %>%
    mutate(Event_Type = gsub(" ([[:digit:]]|[[:punct:]])*$", "", Event_Type)) %>%
    mutate(Event_Type = gsub(" advisory| advisories| damage", "", Event_Type)) %>%
    mutate(Event_Type = gsub("&", "and", Event_Type)) %>%
    mutate(Event_Type = gsub("and$", "", Event_Type))

# Second, let's correct some misspellings and word variations.
storm_data_clean_2 <- storm_data_clean_1 %>%
    mutate(Event_Type = gsub("cstl", "coastal", Event_Type)) %>%
    mutate(Event_Type = gsub("devel", "devil", Event_Type)) %>%
    mutate(Event_Type = gsub("hvy", "heavy", Event_Type)) %>%
    mutate(Event_Type = gsub("clou$", "cloud", Event_Type)) %>%
    mutate(Event_Type = gsub("cool", "cold", Event_Type)) %>%
    mutate(Event_Type = gsub("flooding|floods", "flood", Event_Type)) %>%
    mutate(Event_Type = gsub("flood flash|flash flood from ice jams",
                             "flash flood", Event_Type)) %>%
    mutate(Event_Type = gsub("flash flood.?$|flashflood",
                             "flash flood", Event_Type)) %>%
    mutate(Event_Type = gsub("heat waves", "heat wave", Event_Type)) %>%
    mutate(Event_Type = gsub("/ street", "", Event_Type)) %>%
    mutate(Event_Type = gsub("storms", "storm", Event_Type)) %>%
    mutate(Event_Type = gsub("storm surge/tide", "storm surge", Event_Type)) %>%
    mutate(Event_Type = gsub("torndao", "tornado", Event_Type)) %>%
    mutate(Event_Type = gsub("[/]? tree[s]?", "", Event_Type)) %>%
    mutate(Event_Type = gsub("vog", "fog", Event_Type)) %>%
    mutate(Event_Type = gsub("windchill", "wind chill", Event_Type)) %>%
    mutate(Event_Type = gsub("tstm|thun[[:alpha:]]*", "t-storm", Event_Type))

# Third, let's combine some synonyms that don't require a SME.
storm_data_clean_3 <- storm_data_clean_2 %>%
    mutate(Event_Type = gsub("dry|very dry|abnormally dry", "drought", Event_Type)) %>%
    mutate(Event_Type = gsub("hot weather|hot spell|hot pattern", "heat", Event_Type)) %>%
    mutate(Event_Type = gsub("abnormal warmth|extreme heat|unusually warm",
                         "excessive heat", Event_Type)) %>%
    mutate(Event_Type = gsub("unseasonably warm|unseasonably hot|unusual warmth",
                         "excessive heat", Event_Type)) %>%
      mutate(Event_Type = gsub("very warm|record heat|record/excessive heat",
                         "excessive heat", Event_Type)) %>%
    mutate(Event_Type = gsub("low temperature", "cold", Event_Type)) %>%
    mutate(Event_Type = gsub(" temperature", "", Event_Type)) %>%
    mutate(Event_Type = gsub("bitter cold|unusually cold|unseasonably cold",
                         "extreme cold", Event_Type)) %>%
    mutate(Event_Type = gsub("unseasonable cold|severe cold|hypothermia.*",
```

```
                         "extreme cold", Event_Type)) %>%
    mutate(Event_Type = gsub("^fog$", "dense fog", Event_Type)) %>%
    mutate(Event_Type = gsub("agricultural freeze|damaging freeze",
                             "frost/freeze", Event_Type)) %>%
    mutate(Event_Type = gsub("^freeze$|hard freeze", "frost/freeze", Event_Type)) %>%
    mutate(Event_Type = gsub("early frost|^frost$", "frost/freeze", Event_Type)) %>%
    mutate(Event_Type = gsub("heavy surf|hazardous surf|heavy surf/high surf",
                              "high surf", Event_Type)) %>%
    mutate(Event_Type = gsub("rough surf|rough wave|rough seas", "high surf", Event_Type)) %>%
    mutate(Event_Type = gsub("hurricane/typhoon|hurricane [[:alpha:]]*",
                             "hurricane", Event_Type)) %>%
    mutate(Event_Type = gsub("debris flow|landslides|landslump|mud slide[s]?",
                         "landslide", Event_Type)) %>%
    mutate(Event_Type = gsub("lightning injury", "lightning", Event_Type)) %>%
    mutate(Event_Type = gsub("rip currents", "rip current", Event_Type)) %>%
    mutate(Event_Type = gsub("urban.*", "urban flood", Event_Type)) %>%
    mutate(Event_Type = gsub("tornados|tornadoes", "tornado", Event_Type)) %>%
    mutate(Event_Type = gsub("tropical storm.*", "tropical storm", Event_Type)) %>%
    mutate(Event_Type = gsub("volcanic.*", "volcanic ash", Event_Type)) %>%
    mutate(Event_Type = gsub("water spout|waterspout funnel cloud|waterspout-$",
                         "waterspout", Event_Type)) %>%
    mutate(Event_Type = gsub("waterspouts|wayterspout|waterspout/",
                         "waterspout", Event_Type)) %>%
    mutate(Event_Type = gsub("wild/forest", "wild", Event_Type)) %>%
    mutate(Event_Type = gsub("wild fire|wildfires|brush fire[s]?", "wildfire",
                             Event_Type)) %>%
    mutate(Event_Type = gsub("wnd|winds|wins|non[ |-]thunderstorm wind", "wind",
                             Event_Type)) %>%
    mutate(Event_Type = gsub("high wind.?$|^wind$", "high wind", Event_Type)) %>%
    mutate(Event_Type = gsub("wintery|wintry", "winter", Event_Type)) %>%
    mutate(Event_Type = gsub("winter weather[ |/]mix", "winter mix", Event_Type))

# Finally, on a discretionary basis, let's combine some high-impact event types that are
# closely related, for the purpose of an interesting exploratory analysis. These are
# differences of degree, rather than of kind. Any promising findings can be probed more
# closely with the advice of a SME in a future analysis, potentially.

storm_data_clean_4 <- storm_data_clean_3 %>%
    mutate(Event_Type = gsub("heat wave", "heat", Event_Type)) %>%
    mutate(Event_Type = gsub("excessive heat", "heat", Event_Type)) %>%
    mutate(Event_Type = gsub("extreme cold", "cold", Event_Type)) %>%
    mutate(Event_Type = gsub("cold/wind chill", "cold", Event_Type)) %>%
    mutate(Event_Type = gsub("strong wind", "high wind", Event_Type)) %>%
    mutate(Event_Type = gsub("avalanche", "landslide", Event_Type)) %>%
    arrange(Event_Type)

#To wrap up this analysis, let's see how the worst storm impacts look
# as stacked bar charts. First, fatalities and injuries ...

### Results

# Now let's get an overview of the data with a set of box plots of the impact variables
# that we're focusing on in this analysis.
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

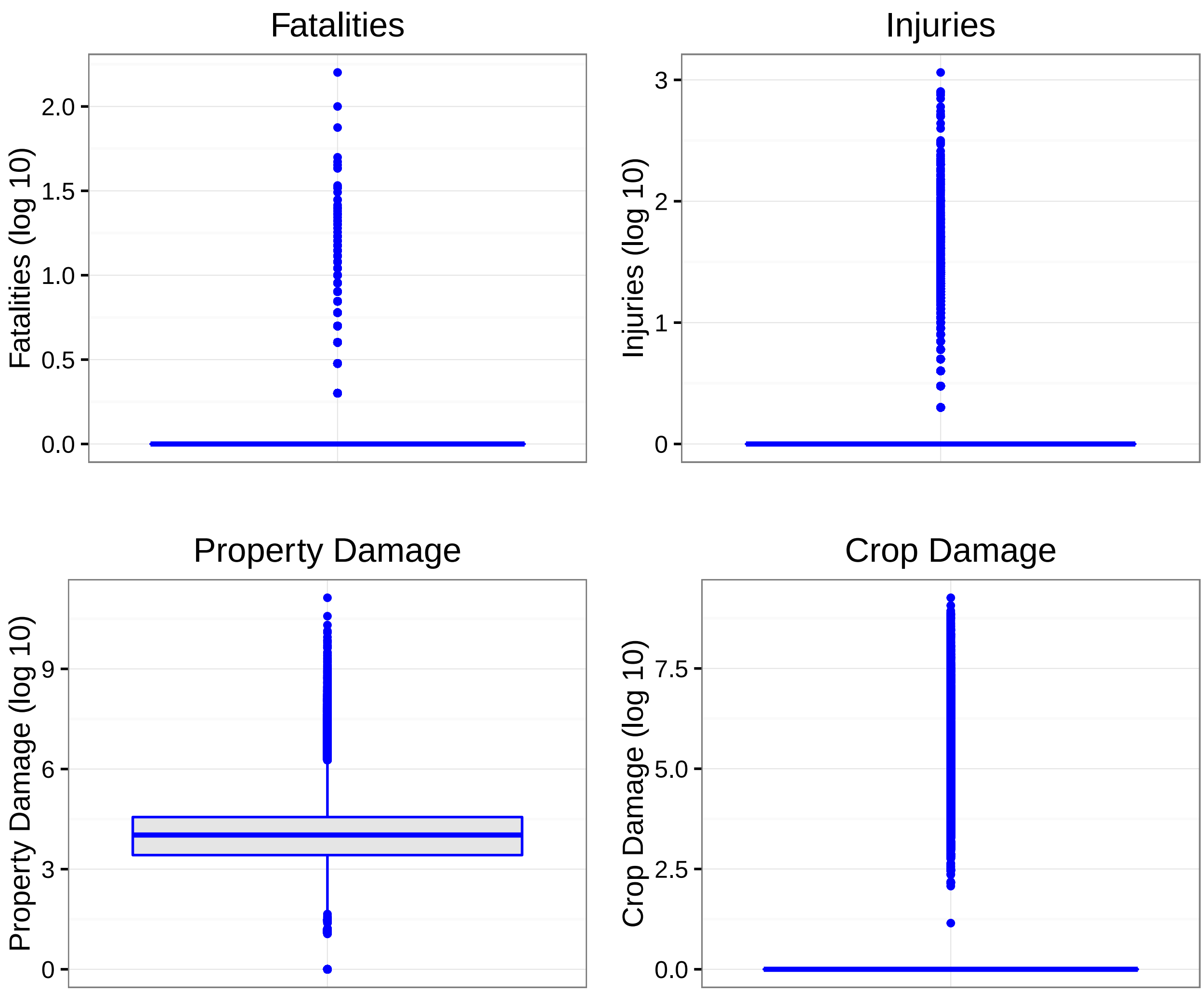
```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.4
```

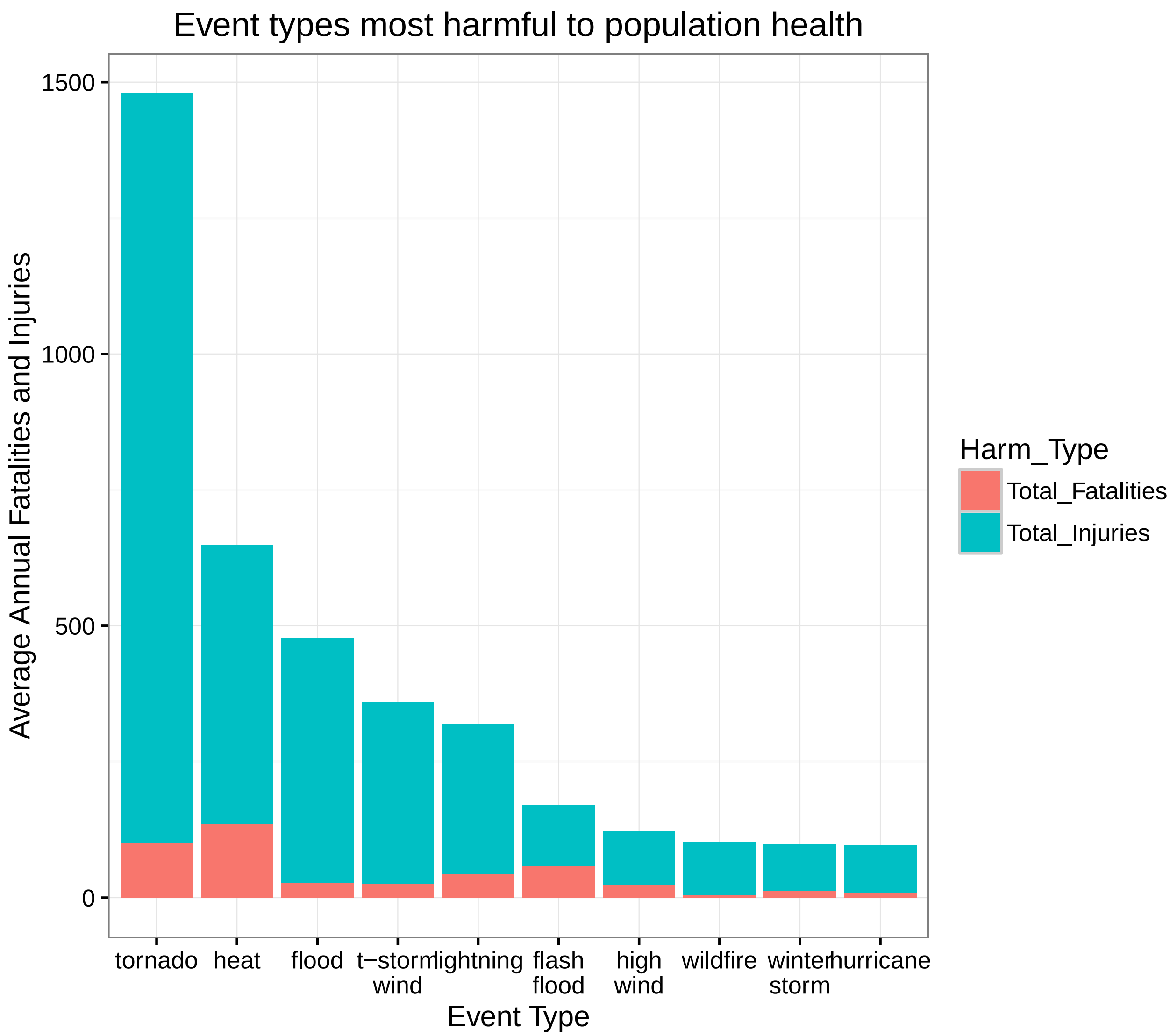
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

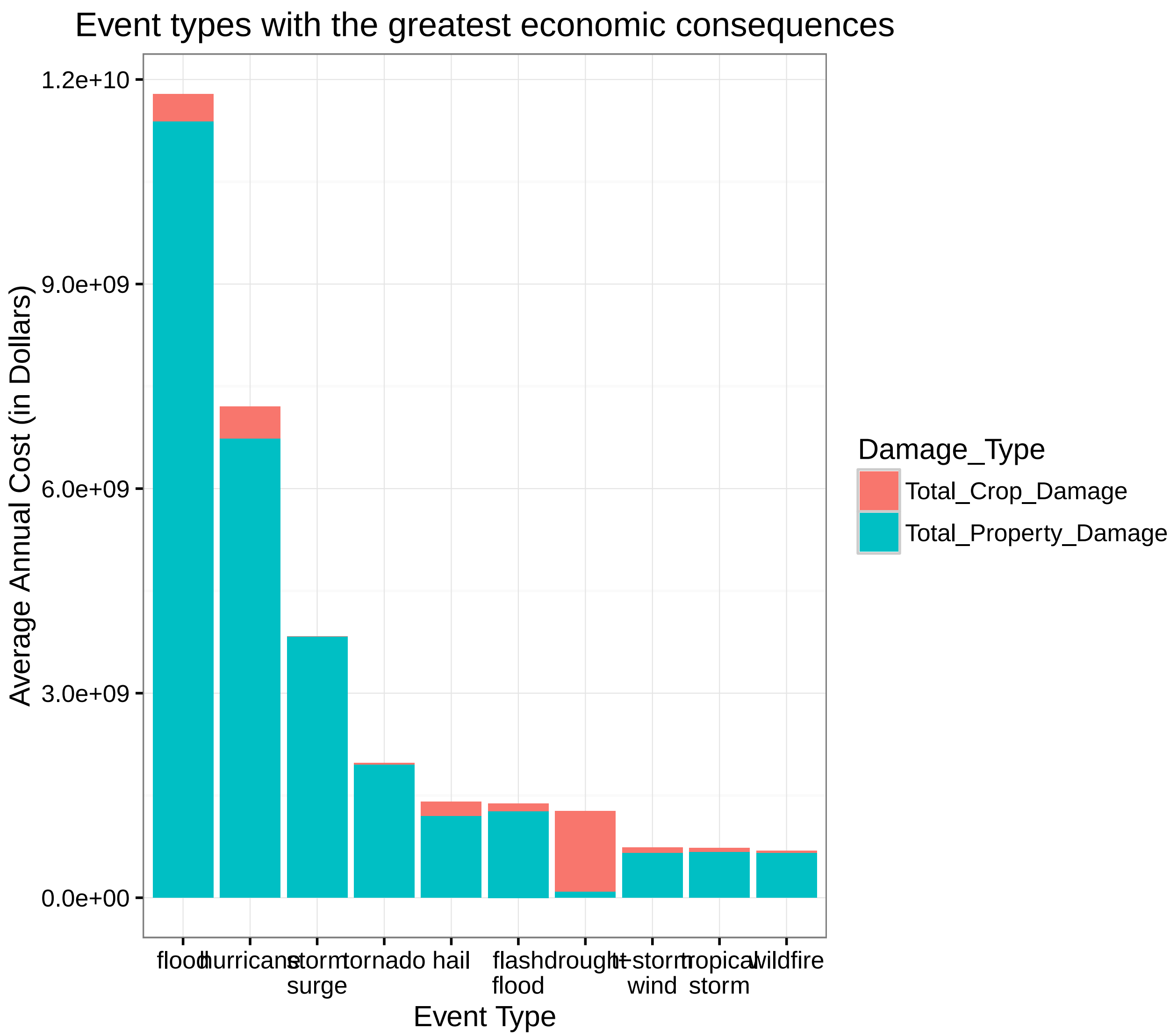
```r
g <- ggplot(storm_data_clean_4, aes(x = factor(0), y = log10(Fatalities + 1)))
plot_1 <- g + geom_boxplot(col = "blue", fill = "gray90", outlier.size = 1) +
    xlab("") + ylab("Fatalities (log 10)") + theme_bw() +
    theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
    ggtitle("Fatalities")

g <- ggplot(storm_data_clean_4, aes(x = factor(0), y = log10(Injuries + 1)))
plot_2 <- g + geom_boxplot(col = "blue", fill = "gray90", outlier.size = 1) +
    xlab("") + ylab("Injuries (log 10)") + theme_bw() +
    theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
    ggtitle("Injuries")

g <- ggplot(storm_data_clean_4, aes(x = factor(0), y = log10(Property_Damage + 1)))
plot_3 <- g + geom_boxplot(col = "blue", fill = "gray90", outlier.size = 1) +
    xlab("") + ylab("Property Damage (log 10)") + theme_bw() +
    theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
    ggtitle("Property Damage")

g <- ggplot(storm_data_clean_4, aes(x = factor(0), y = log10(Crop_Damage + 1)))
plot_4 <- g + geom_boxplot(col = "blue", fill = "gray90", outlier.size = 1) +
    xlab("") + ylab("Crop Damage (log 10)") + theme_bw() +
    theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
    ggtitle("Crop Damage")

grid.arrange(plot_1, plot_2, plot_3, plot_4, ncol=2)
```




```
storm_data_plot <- mutate(storm_data_clean_4, Event_Type = gsub("[ /]", "\n", Event_Type))

storm_data_harm <- storm_data_plot %>%
    group_by(Event_Type) %>%
    summarize(Total_Fatalities = sum(Fatalities),
              Total_Injuries = sum(Injuries),
              Total_Harm = Total_Fatalities + Total_Injuries) %>%
    arrange(desc(Total_Harm))
storm_data_harm_high <- storm_data_harm %>%
  top_n(10, Total_Harm) %>%
  gather("Harm_Type", "Number_People", 2:3) %>%
  mutate(Event_Type = reorder(Event_Type, -Total_Harm))

g <- ggplot(storm_data_harm_high,
            aes(Event_Type, Number_People / year_range, fill = Harm_Type))
p <- g + geom_bar(stat = "identity") +
    xlab("Event Type") + ylab("Average Annual Fatalities and Injuries") +
    ggtitle("Event types most harmful to population health") + theme_bw()
print(p)
```




```r
# Second, property damage and crop damage ...

storm_data_damage <- storm_data_plot %>%
    group_by(Event_Type) %>%
    summarize(Total_Property_Damage = sum(Property_Damage),
              Total_Crop_Damage = sum(Crop_Damage),
              Total_Damage = Total_Property_Damage + Total_Crop_Damage) %>%
    arrange(desc(Total_Damage))
storm_data_damage_high <- storm_data_damage %>%
  top_n(10, Total_Damage) %>%
  gather("Damage_Type", "Cost", 2:3) %>%
  mutate(Event_Type = reorder(Event_Type, -Total_Damage))

g <- ggplot(storm_data_damage_high, aes(Event_Type, Cost / year_range, fill = Damage_Type))
p <- g + geom_bar(stat = "identity") +
    xlab("Event Type") + ylab("Average Annual Cost (in Dollars)") +
    ggtitle("Event types with the greatest economic consequences") + theme_bw()
print(p)
```



In conclusion . . .

explain why partial cleanup of the EVTYPE variable is sufficient "My current submission lists the event types, states that there are some overlaps, and calls this out as an area for further investigation in terms of future reporting / research, and that this report is simply (naively) based on the 'as-is' EVTYPE values."