Analysis of storm events in the U.S. that causes the highest amount of losses on population and economics

Michaela Spiegel

7 Juni 2018

Synopsis

In this analysis two questions are adressed:

1. Across the United States, which types of events are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

Therefore, data from the National & Atmospheric Administration's (NOAA) storm database is used. This data contains storm events from the U.S. between 1950 till end of 2011. The number of attributes for each entry in the database is quite high and the data is kind of messy which needs an excessive data cleansing, before it is possible to analyze the data further. The relevant attributes are selected, storm events are grouped into new categories and amounts of damge/crop costs are unified, that the two questions could be adressed. Therefore, the data is summarized by summing up fatalities, injuries, property and crop damage costs for the new event categories.

The tornado is the storm event which resulted into the highest number of fatalities and injured people in the U.S. between 1950 and 2013, followed by Heat and Flood. The three most expensive storm events are Flood, Hurricane/Typhoon and Tornado.

Data Processing

Reading in Data

First I download the storm data from the National & Atmospheric Administration's (NOAA) storm database.

```
data_url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
doc_url <-
  "https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf"
faq_url <-
  "https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf"
download.file(data_url, "stormdata.csv.bz2")
download.file(doc_url, "doc.pdf")
download.file(faq_url, "faq.pdf")
storm_data <- read.csv("stormdata.csv.bz2")
```

The dataframe is quite huge with 37 columns and 902297 rows.

Getting an overview about the data

First looking at the different attributes we have in this storm data set and select the ones we are interested in.

```
library(dplyr)
names(storm_data) <- tolower(names(storm_data))
names(storm_data)
```

```
##  [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"
```

```
storm_data <- storm_data %>%
  select(evtype, fatalities, injuries, propdmg, propdmgexp, cropdmg, cropdmgexp)
```

We are interested in population health and economic consequences for different event types. Therefore we filter out only the attributes we are interested in: the number of fatalitites, injuries, the property damage (propdmg, propdmgexp) and the crop damage (cropdmg, cropdmgexp).

Exploring and Cleaning up event types

The first attribute I took a closer look at is the event type.

```
events <- table(storm_data$evtype)
head(events[order(-events)],12)
```

```
## 
##               HAIL          TSTM WIND  THUNDERSTORM WIND 
##             288661             219940              82563 
##            TORNADO        FLASH FLOOD              FLOOD 
##              60652              54277              25326 
## THUNDERSTORM WINDS          HIGH WIND          LIGHTNING 
##              20843              20212              15754 
##         HEAVY SNOW         HEAVY RAIN       WINTER STORM 
##              15708              11723              11433
```

If we just look at the twelve most commont event types, we can see that the notation in the database is not consistent. For example "TSTM WIND", "THUNDERSTORM WIND", "THUNDERSTORM WINDS" can be grouped together for sure as the same type of event. So we have to deal with singular and plural forms and spelling errors. **In total there are 985 different event types.**

```
events_smaller_ten = 0
names_smaller_ten = ""
for (i in 1:nrow(events)){
  if (events[[i]] %in% c(1:10)){
    events_smaller_ten = events_smaller_ten + 1
    names_smaller_ten[events_smaller_ten] <- names(events[i])
    }
}
```

Some values for event types appear only 1 to 10 times. This will end up in a very tedious work. They actually make up 826 different event type values.

Examples of these low fequency event type values are:

```
head(names_smaller_ten,12)
```

```
##  [1] "?"                              "ABNORMALLY DRY"                
##  [3] "ABNORMALLY WET"                 "ABNORMAL WARMTH"               
##  [5] "ACCUMULATED SNOWFALL"           "AGRICULTURAL FREEZE"
##  [7] "APACHE COUNTY"               "AVALANCE"
##  [9] "BEACH EROSIN"                "Beach Erosion"
## [11] "BEACH EROSION"               "BEACH EROSION/COASTAL FLOOD"
```

In this example we see, we have to deal with small / capital letters, spelling errors and adjectives / adverbs in front of events like e.g. abnormal, abnormally.

To clean up the data first all types are converted to lower characters.

```r
library(dplyr)
storm_data$evtype <- tolower(storm_data$evtype)
```

In total there are now 898 (before: 985) different event types.

The next thing which we see is the shortcut TSTM for thunderstorm, so we can go and change this and other small spell errors and take a look at the most common events. To group event types like “thunderstorm winds” and “thunderstrom wind”, we can do some stemming, which will change the plural and singular to the stem of the word. Therefore the string is splitted first into singular words and then each word is stemmed individually. Numbers and brackets, which are e.g. often given for hail, are deleted.

```r
library(stringr)
library(dplyr)
library(SnowballC)
storm_data$evtype <- str_replace_all(storm_data$evtype, "tstm", "thunderstorm")
storm_data$evtype <- str_replace_all(storm_data$evtype, "wnd|wi$|win$|windss",
    "wind")
storm_data$evtype <- str_replace_all(storm_data$evtype, "fld", "flood")
storm_data$evtype <- str_replace_all(storm_data$evtype, "[0-9()\\.]", "")
storm_data$evtype <- storm_data$evtype %>% sapply(strsplit, split = " ") %>%
    sapply(wordStem, language = "en")
storm_data$evtype <- storm_data$evtype %>% sapply(unlist) %>% sapply(paste,
    collapse = " ")
storm_data$evtype <- str_replace_all(storm_data$evtype, "heavi surf/high surf",
    "high surf")
storm_data$evtype <- str_replace_all(storm_data$evtype, "frost/freez", "freez")
storm_data$evtype <- str_replace_all(storm_data$evtype, "currents", "current")
storm_data$evtype <- str_replace_all(storm_data$evtype, "storm surge/tid", "storm surg")
storm_data$evtype <- str_replace_all(storm_data$evtype, "rainfal", "rain")
storm_data$evtype <- str_replace_all(storm_data$evtype, "ashfal", "ash")
storm_data$evtype <- str_replace_all(storm_data$evtype, "snowfal", "snow")
storm_data$evtype <- str_replace_all(storm_data$evtype, "ligntn", "lightn")
storm_data$evtype <- str_replace_all(storm_data$evtype, "heavi surf", "high surf")
storm_data$evtype <- str_replace_all(storm_data$evtype, "wild/forest fire",
    "wildfir")
```

In total there are now 689 (before: 985) different event types.

Multiple event categories for one entries

Grouping event type to new categories

The cleaning of the event types would take a lot more time, so the event types are grouped into categories by searching for keywords with regular expressions.

Snow Drought vs. Heat Drought.

```
library(stringr)
library(dplyr)
storm_data$event_type <- storm_data$evtype
storm_data$event_type <- "no category"
# clear without ambiguity
storm_data$event_type[storm_data$evtype %>% str_detect("low tide")] <- "Astronomical Low Tide"
storm_data$event_type[storm_data$evtype %>% str_detect("smoke")] <- "Dense Smoke"
storm_data$event_type[storm_data$evtype %>% str_detect("dust devil")] <- "Dust Devil"
storm_data$event_type[storm_data$evtype %>% str_detect("dust storm")] <- "Dust Storm"
storm_data$event_type[storm_data$evtype %>% str_detect("heat")] <- "Heat"
storm_data$event_type[storm_data$evtype %>% str_detect("funnel cloud")] <- "Funnel Cloud"
storm_data$event_type[storm_data$evtype %>% str_detect("freez fog")] <- "Freezing Fog"
storm_data$event_type[storm_data$evtype %>% str_detect("heavi rain")] <- "Heavy Rain"
storm_data$event_type[storm_data$evtype %>% str_detect("hurrican|typhoon")] <- "Hurricane/Typhoon"
storm_data$event_type[storm_data$evtype %>% str_detect("[^trop]ic storm|^ice storm")] <- "Ice Storm"
storm_data$event_type[storm_data$evtype %>% str_detect("lake-effect snow")] <- "Lake-Effect Snow"
storm_data$event_type[storm_data$evtype %>% str_detect("rip curr")] <- "Rip Current"
storm_data$event_type[storm_data$evtype %>% str_detect("seich")] <- "Seiche"
storm_data$event_type[storm_data$evtype %>% str_detect("tropic depress")] <- "Tropical Depression"
storm_data$event_type[storm_data$evtype %>% str_detect("tropic storm")] <- "Tropical Storm"
storm_data$event_type[storm_data$evtype %>% str_detect("tsunami")] <- "Tsunami"
storm_data$event_type[storm_data$evtype %>% str_detect("wildfir")] <- "Wildfire"
storm_data$event_type[storm_data$evtype %>% str_detect("winter weath")] <- "Winter Weather"
###
storm_data$event_type[storm_data$evtype %>% str_detect("avalan")] <- "Avalanche"
storm_data$event_type[storm_data$evtype %>% str_detect("blizzard")] <- "Blizzard"
storm_data$event_type[storm_data$evtype %>% str_detect("cold|wind chil|windchil")] <-
  "Cold/Wind Chill"
storm_data$event_type[storm_data$evtype %>% str_detect("fog")] <- "Fog"
storm_data$event_type[storm_data$evtype %>% str_detect("drought")] <- "Drought"
storm_data$event_type[storm_data$evtype %>% str_detect("flood")] <- "Flood"
storm_data$event_type[storm_data$evtype %>% str_detect("frost|freez")] <- "Frost/Freeze"
storm_data$event_type[storm_data$evtype %>% str_detect("hail")] <- "Hail"
storm_data$event_type[storm_data$evtype %>% str_detect("snow")] <- "Snow"
storm_data$event_type[storm_data$evtype %>% str_detect("high surf")] <- "High Surf"
storm_data$event_type[storm_data$evtype %>% str_detect("high wind")] <- "High Wind"
storm_data$event_type[storm_data$evtype %>% str_detect("lightn")] <- "Lightning"
storm_data$event_type[storm_data$evtype %>% str_detect("sleet")] <- "Sleet"
storm_data$event_type[storm_data$evtype %>% str_detect("storm surg")] <- "Storm Surge/Tide"
storm_data$event_type[storm_data$evtype %>% str_detect("strong wind")] <- "Strong Wind"
storm_data$event_type[storm_data$evtype %>% str_detect("thunderstorm")] <- "Thunderstorm Wind"
storm_data$event_type[storm_data$evtype %>% str_detect("tornado")] <- "Tornado"
storm_data$event_type[storm_data$evtype %>% str_detect("volcan ash")] <- "Volcanic Ash"
storm_data$event_type[storm_data$evtype %>% str_detect("waterspout")] <- "Waterspout"
storm_data$event_type[storm_data$evtype %>% str_detect("winter storm")] <- "Winter Storm"
```

One left over category called “no category” invites to extend the pattern search and sort the leftovers into the other categories. Some events like e.g. “other” can not be assigned.

```
lost<-table(storm_data$evtype[storm_data$event_type=="no category"])
head(lost[order(-lost)],12)
```

```
##
##           landslid               wind     dri microburst
##                608               383               192
##      record warmth     unseason warm astronom high tide
##                154               126               103
##         wintri mix        gusti wind               ice
##                 94                89                61
##       unseason dri             other            funnel
##                 56                52                47
```

Cleaning up the damage cost values

The entry propdmg gives us a numeric value, which describes the property damage. To get the right unit we have to consider propdmgexp and multiply by the definition which can be found here: Link to explanation (How to handle exponent value). The different types of the exponent values for properties are:

```
table(storm_data$propdmgexp)
```

```
## 
##             -      ?      +      0      1      2      3      4      5 
## 465934      1      8      5    216     25     13      4      4     28 
##      6      7      8      B      h      H      K      m      M 
##      4      5      1     40      1      6 424665      7  11330 
```

The shortcut "B" as example stands for billion Dollar. For further analysis we recalculate the damage prices in k$'s and save this values as new attributes (property_damage and crop_damage.

```
storm_data$crop_damage <- 0
storm_data$property_damage <- 0
storm_data$cropdmgexp <- tolower(storm_data$cropdmgexp)
storm_data$propdmgexp <- tolower(storm_data$propdmgexp)
```

```
storm_data$property_damage[storm_data$propdmgexp == "+"] <-
  storm_data$propdmg[storm_data$propdmgexp == "+"]/1000
storm_data$property_damage[storm_data$propdmgexp %in% c(0:8)] <-
  storm_data$propdmg[storm_data$propdmgexp %in% c(0:8)]/100
storm_data$property_damage[storm_data$propdmgexp == "h"] <-
  storm_data$propdmg[storm_data$propdmgexp == "h"]/10
storm_data$property_damage[storm_data$propdmgexp == "k"] <-
  storm_data$propdmg[storm_data$propdmgexp == "k"]
storm_data$property_damage[storm_data$propdmgexp == "m"] <-
  storm_data$propdmg[storm_data$propdmgexp == "m"]*1000
storm_data$property_damage[storm_data$propdmgexp == "b"] <-
  storm_data$propdmg[storm_data$propdmgexp == "b"]*1000000
```

```
storm_data$crop_damage[storm_data$cropdmgexp %in% c(0:8)] <-
  storm_data$cropdmg[storm_data$cropdmgexp %in% c(0:8)]/100
storm_data$crop_damage[storm_data$cropdmgexp == "k"] <-
  storm_data$cropdmg[storm_data$cropdmgexp == "k"]
storm_data$crop_damage[storm_data$cropdmgexp == "m"] <-
  storm_data$cropdmg[storm_data$cropdmgexp == "m"]*1000
storm_data$crop_damage[storm_data$cropdmgexp == "b"] <-
  storm_data$cropdmg[storm_data$cropdmgexp == "b"]*1000000
```

Now, we can delete our other cost attributes and the old event types.

```
library(dplyr)
storm_data <-
```

```
  storm_data %>%
  select(event_type, fatalities, injuries, crop_damage, property_damage)
```

Summarizing the data

We want to summarize the data by summing up the fatalities, injuries and damage costs for different event types. The total damage is calculated by the sum of the property and the crop damage. The new tables are sorted by highest number of fatalities or damage costs for later visualization.

```
library(dplyr)
library(reshape2)
event_consequences <-
  storm_data %>%
  group_by(event_type) %>%
  summarise(fatalities = sum(fatalities), injuries = sum(injuries),
            property_damage = sum(property_damage), crop_damage = sum(crop_damage)) %>%
  mutate(total_damage = crop_damage + property_damage)
# prepare data for plotting
hdata <- event_consequences %>%
  select(event_type, fatalities, injuries) %>%
  arrange(desc(fatalities), desc(injuries)) %>%
  melt(id=c("event_type"))
edata <- event_consequences  %>%
  arrange(desc(total_damage)) %>%
  select(event_type, crop_damage, property_damage) %>%
  melt(id=c("event_type"))
```

Data Results

Across the United States, which types of events are most harmful with respect to population health?

To adress the most harmful events, the event type with the highest number of fatalities are shown in Figure 1. The tornado is the storm event which resulted into the highest number of fatalities and injured people in the U.S. between 1950 and 2013, followed by Heat and Flood.

```
library(ggplot2)
l <- hdata$event_type[order(hdata$value[hdata$variable=="fatalities"])] %>% unique()
hdata$event_type <- factor(hdata$event_type, levels=l)
f <- ggplot(hdata[hdata$event_type %in% tail(l, n=10),],
            aes(x=event_type, y=value/100, fill=value))
f <- f+ geom_col() + coord_flip()  +  facet_grid(variable ~.)
f + scale_y_log10() +
  labs(y = "number of people [hundred]", x = "type of event", caption = "Figure 1: This figure shows in
```

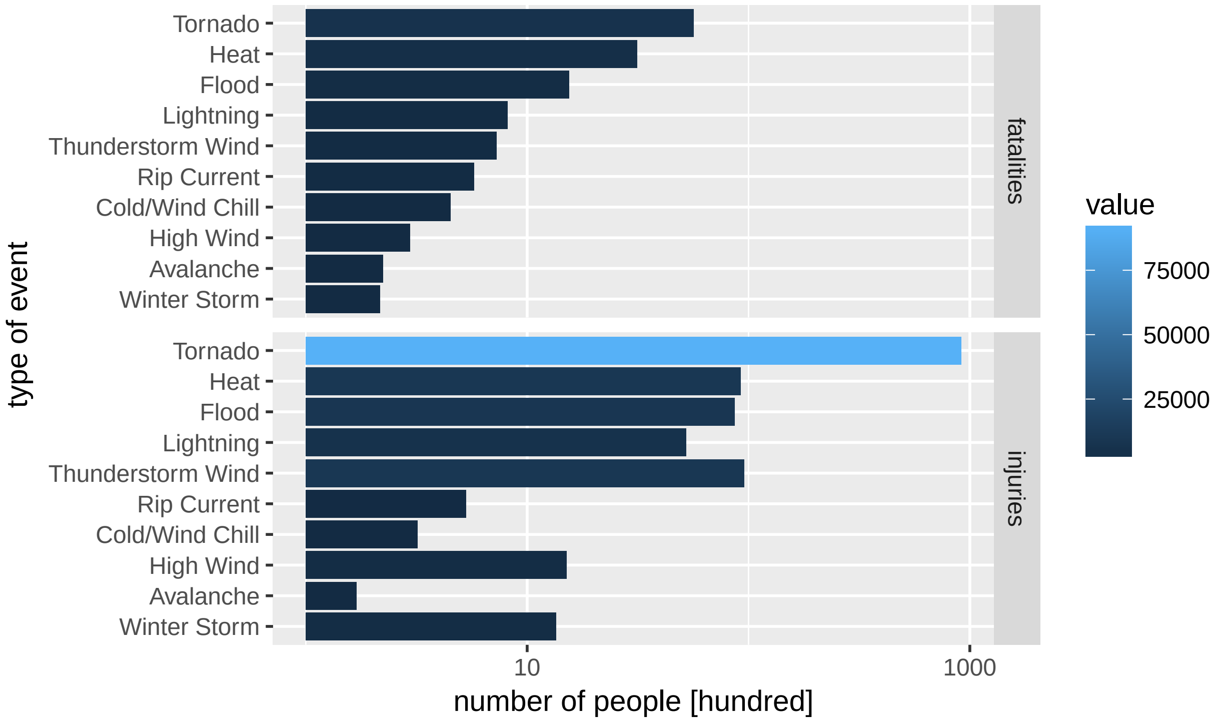

Figure 1: This figure shows in the top plot the sum of all fatalities for different storm events and at the lower plot the sum of all injuries for different storm events. Only the Top 10 storm events in number of fatalities, are shown here.

Across the United States, which types of events have the greatest economic consequences?

This question is answered by Figure 2. Here we show the 15 events which became most expensive for the U.S. between 1950 and 2013. The costs of crop and property damaged were summed up over all the years. The three most expensive storm events are flood, hurricane/typhoon and tornado. The property damage for these events is much higher than the crop damage costs. Other storm events like drought or frost/freeze lead to a higher crop than property damage. Additionally all the expenses for treating injured people, are not counted in, this could also shift the ranking of the most expensive storm events.

```
library(ggplot2)
l <- event_consequences$event_type[order(event_consequences$total_damage)]
edata$event_type <- factor(edata$event_type, levels=l)
l <- tail(l, n=15)
#edata[edata$event_type %in% l,]
f <- ggplot(edata[edata$event_type %in% l,],
            aes(x=event_type, y=value/1000000, group=variable, fill=variable ))
f + geom_col()  + coord_flip() +
  labs(y = "costs in [Billion Dollar]", x = "type of event", caption = "Figure 2: This figure shows the
  scale_fill_discrete(name="Cost Type", labels = c("Crop Damage", "Property Damage")) +
  scale_fill_brewer(palette="Set1")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill',
## which will replace the existing scale.
```

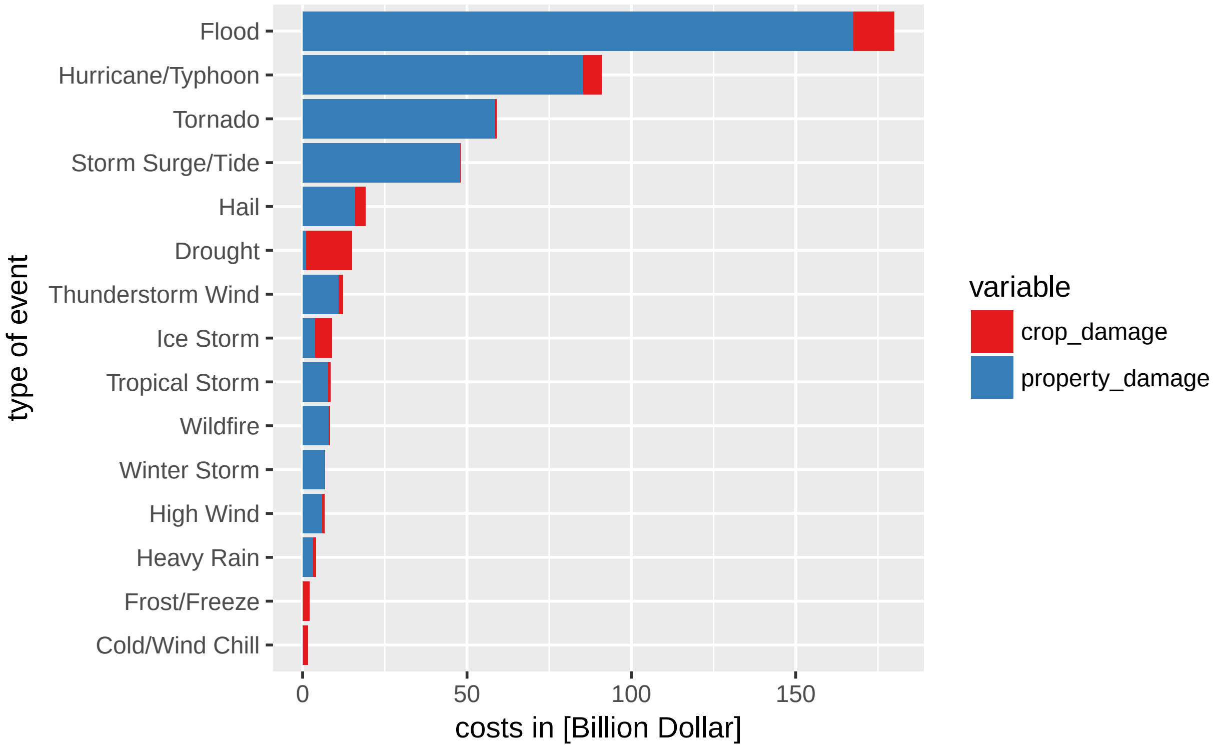

Figure 2: This figure shows the storm events, which cause the 15 highest amounts of economical damage in the U.S. in form of crop damage (red) and property damage(blue).