Impact of weather on human life, health, properties and agriculture

Maciej Gielnik

16/10/2020

Synopsis

Different climate events can have an impact on human life and economic stability. Here I have analyzed the impact of climate events on human life, health, human properties and agriculture. In conclusion tornadoes seem to have to have the biggest impact on human life and health. From the economic point of view floods seem to have the biggest impact on properties and drought seem to be the most harmful for crops.

Data Processing

- Downloading the raw data

```
if (!file.exists("repdata_data_StormData.csv.bz2")) {
  URL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
  download.file(URL, destfile = "./repdata_data_StormData.csv.bz2")
}
```

- Loading libraries

```
library(R.utils)
```

```
## Loading required package: R.oo
## Loading required package: R.methodsS3
## R.methodsS3 v1.8.1 (2020-08-26 16:20:06 UTC) successfully loaded. See ?R.methodsS3 for help.
## R.oo v1.24.0 (2020-08-26 16:11:58 UTC) successfully loaded. See ?R.oo for help.
##
## Attaching package: 'R.oo'
## The following object is masked from 'package:R.methodsS3':
##
##     throw
## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods
## The following objects are masked from 'package:base':
##
##     attach, detach, load, save
## R.utils v2.11.0 (2021-09-26 08:30:02 UTC) successfully loaded. See ?R.utils for help.
##
## Attaching package: 'R.utils'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, inherits, isOpen, nullfile, parse,
##     warnings
```

```
library(data.table)
library(lubridate)
```

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

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

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

```
library(ggplot2)
```

- Loading raw data

```
stormdata <- fread("repdata_data_StormData.csv.bz2")
```

- Changing characters to lower case in columns EVTYPE, PROPDMGEXP and CROPDMGEXP

```
stormdata$EVTYPE <- tolower(stormdata$EVTYPE)
stormdata$PROPDMGEXP <- tolower(stormdata$PROPDMGEXP)
stormdata$CROPDMGEXP <- tolower(stormdata$CROPDMGEXP)
```

- Changing format of column BGN_DATE to Date

```
stormdata$BGN_DATE <- as.Date(stormdata$BGN_DATE, format = "%m/%d/%Y %H:%M:%S")
```

- Due to untidiness in column EVTYPE some types of events were reclassified according to the code below

```
stormdata$EVTYPE[stormdata$EVTYPE %like% "summary"] <- "none"
stormdata$EVTYPE[stormdata$EVTYPE %like% "urban"] <- "none"

stormdata$EVTYPE[stormdata$EVTYPE %like% "flood"] <- "flood"
stormdata$EVTYPE[stormdata$EVTYPE %like% "fld"] <- "flood"

stormdata$EVTYPE[stormdata$EVTYPE %like% "rain"] <- "rain"

stormdata$EVTYPE[stormdata$EVTYPE %like% "cold"] <- "cold"
stormdata$EVTYPE[stormdata$EVTYPE %like% "cool"] <- "cold"
stormdata$EVTYPE[stormdata$EVTYPE %like% "chill"] <- "cold"

stormdata$EVTYPE[stormdata$EVTYPE %like% "hail"] <- "hail"

stormdata$EVTYPE[stormdata$EVTYPE %like% "winter"] <- "winter"
stormdata$EVTYPE[stormdata$EVTYPE %like% "snow"] <- "winter"
```

```
stormdata$EVTYPE[stormdata$EVTYPE %like% "ice"] <- "freeze"
stormdata$EVTYPE[stormdata$EVTYPE %like% "freeze"] <- "freeze"
stormdata$EVTYPE[stormdata$EVTYPE %like% "freez"] <- "freeze"
stormdata$EVTYPE[stormdata$EVTYPE %like% "frost"] <- "freeze"

stormdata$EVTYPE[stormdata$EVTYPE %like% "blizzard"] <- "blizzard"

stormdata$EVTYPE[stormdata$EVTYPE %like% "warm"] <- "heat"
stormdata$EVTYPE[stormdata$EVTYPE %like% "hot"] <- "heat"
stormdata$EVTYPE[stormdata$EVTYPE %like% "heat"] <- "heat"

stormdata$EVTYPE[stormdata$EVTYPE %like% "dry"] <- "drought"

stormdata$EVTYPE[stormdata$EVTYPE %like% "thund"] <- "thunderstorm"
stormdata$EVTYPE[stormdata$EVTYPE %like% "tstm"] <- "thunderstorm"

stormdata$EVTYPE[stormdata$EVTYPE %like% "tornado"] <- "tornado"
stormdata$EVTYPE[stormdata$EVTYPE %like% "funnel"] <- "tornado"

stormdata$EVTYPE[stormdata$EVTYPE %like% "fog"] <- "fog"
stormdata$EVTYPE[stormdata$EVTYPE %like% "vog"] <- "fog"

stormdata$EVTYPE[stormdata$EVTYPE %like% "wind"] <- "wind"
stormdata$EVTYPE[stormdata$EVTYPE %like% "wnd"] <- "wind"

stormdata$EVTYPE[stormdata$EVTYPE %like% "volca"] <- "volcanic-ash"

stormdata$EVTYPE[stormdata$EVTYPE %like% "dust"] <- "dust"

stormdata$EVTYPE[stormdata$EVTYPE %like% "storm"] <- "storm"

stormdata$EVTYPE[stormdata$EVTYPE %like% "spout"] <- "waterspout"

stormdata$EVTYPE[stormdata$EVTYPE %like% "light"] <- "lightning"

stormdata$EVTYPE[stormdata$EVTYPE %like% "hurricane"] <- "hurricane"

stormdata$EVTYPE[stormdata$EVTYPE %like% "lands"] <- "avalanche"

stormdata$EVTYPE[stormdata$EVTYPE %like% "rip"] <- "rip-current"
```

- Newly classified EVTYPES were changed to a factor

```
stormdata$EVTYPE <- as.factor(stormdata$EVTYPE)
```

- Exponent values were fixed according to article "How To Handle Exponent Value of PROPDMGEXP and CROPDMGEXP" available from the following link https://rstudio-pubs-static.s3.amazonaws.com/58957_37b6723ee52b455990e149edde45e5b6.html. Fixed values of PROPDMGEXP were stored in pexp variable and fixed values of CROPDMGEXP were stored in cexp variable.

```
  pexp <- stormdata$PROPDMGEXP

  pexp[pexp == "1" | pexp == "2" | pexp == "3" | pexp == "4" | pexp == "5" |
         pexp == "6" | pexp == "7" | pexp == "8" | pexp == "0"] <- 10
  pexp[pexp == "" | pexp == "-" | pexp == "?"] <- 0
```

```
  pexp[pexp == "+" ] <- 1
  pexp[pexp == "h" ] <- 100
  pexp[pexp == "k" ] <- 1000
  pexp[pexp == "m" ] <- 1000000
  pexp[pexp == "b" ] <- 1000000000

  pexp <- as.numeric(pexp)

  cexp <- stormdata$CROPDMGEXP

  cexp[cexp == "1" | cexp == "2" | cexp == "3" | cexp == "4" | cexp == "5" |
         cexp == "6" | cexp == "7" | cexp == "8" | cexp == "0"] <- 10
  cexp[cexp == "" | cexp == "-" | cexp == "?"] <- 0
  cexp[cexp == "+" ] <- 1
  cexp[cexp == "h" ] <- 100
  cexp[cexp == "k" ] <- 1000
  cexp[cexp == "m" ] <- 1000000
  cexp[cexp == "b" ] <- 1000000000

  cexp <- as.numeric(cexp)
```

- New columns pexp and cexp were added to stormdata

```
stormdata$pexp <- pexp
stormdata$cexp <- cexp
```

- Adding column ecprop - economic consequences for properties as a multiplication of column PROPDMG and pexp from stormdata

```
stormdata$ecprop  <- stormdata$PROPDMG * stormdata$pexp
```

- Adding column - economic consequences for crops as a multiplication of column CROPDMG and cexp from stormdata

```
stormdata$eccrop <- stormdata$CROPDMG * stormdata$cexp
```

- Creating a new data frame sd2001 for stormdata from 2001 to 2011

```
sd2001ss <- stormdata$BGN_DATE >= ymd("2001 01 01")
sd2001 <- stormdata[sd2001ss,]
```

- Defining figure size as a global chunk option

```
knitr::opts_chunk$set(fig.width=12, fig.height=8)
```

Results

- Creating a variable fmt600 storing information about fatalities from different event types from 1950 to 2011

```
fatalities <- with(stormdata, tapply(FATALITIES, EVTYPE, sum, na.rm = TRUE))
fmt600 <- fatalities[fatalities>600]
```

- Creating a variable inj5000 storing information about injured people from different event types from 1950 to 2011

```
injuries <- with(stormdata, tapply(INJURIES, EVTYPE, sum, na.rm = TRUE))
inj5000 <- injuries[injuries > 5000]
```

- Creating a variable fmt40001 storing information about fatalities from different event types from 2001 to 2011

```
fatalities01 <- with(sd2001, tapply(FATALITIES, EVTYPE, sum, na.rm = TRUE))
fmt40001 <- fatalities01[fatalities01>400]
```

- Creating a variable inj120001 storing information about injured people from different event types from 2001 to 2011

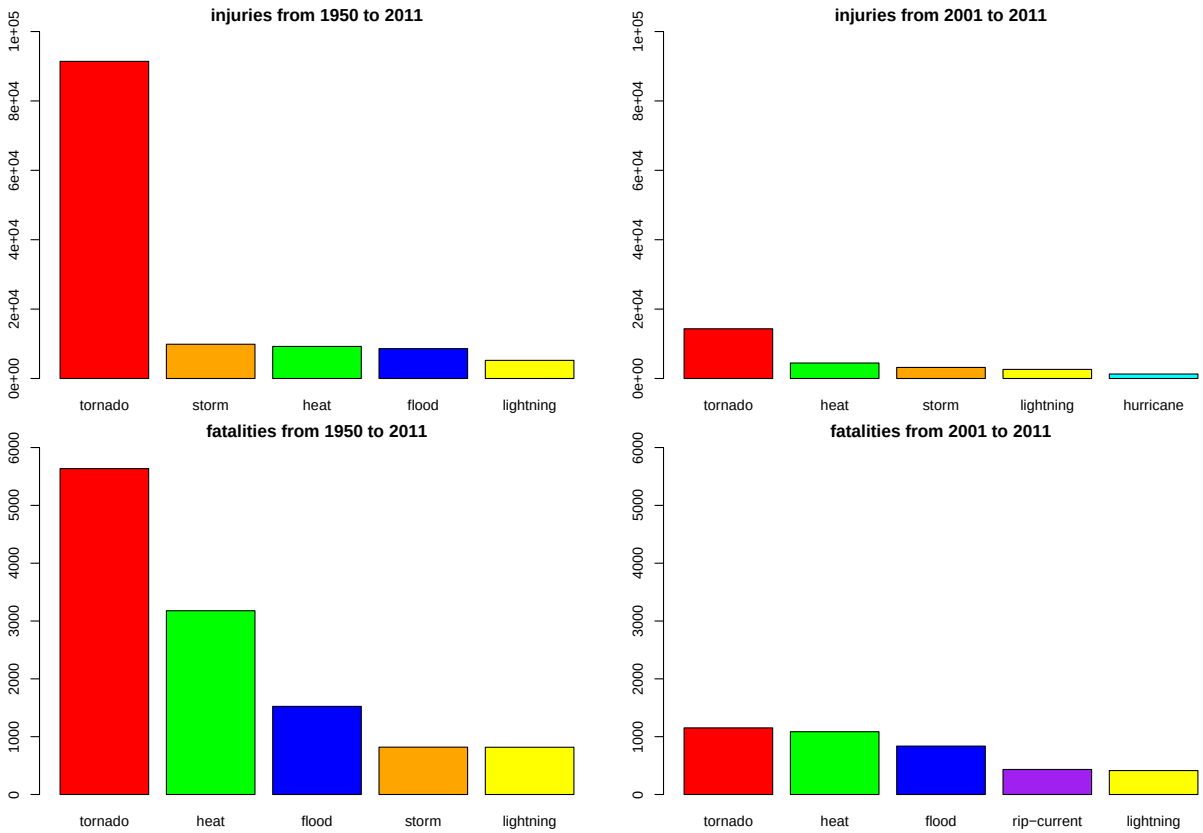
```
injuries01 <- with(sd2001, tapply(INJURIES, EVTYPE, sum, na.rm = TRUE))
inj120001 <- injuries01[injuries01 > 1200]
```

- Sorting variables fmt600, inj5000, fmt40001, inj120001 and making a plot.

```
d1 <- sort(fmt600, decreasing = TRUE)
d2 <- sort(inj5000, decreasing = TRUE)
d3 <- sort(fmt40001, decreasing = TRUE)
d4 <- sort(inj120001, decreasing = TRUE)

par(mfrow = c(2,2), mar = c(2, 2, 2, 2))


barplot(d2, ylim = c (0, 100000), col = c("red", "orange", "green", "blue", "yellow"), main = "injuries
barplot(d4, ylim = c (0, 100000), col = c("red", "green", "orange", "yellow", "cyan" ),  main = "injuri
barplot(d1, ylim = c (0, 6000), col = c("red", "green", "blue", "orange", "yellow"), main = "fatalities
barplot(d3, ylim = c (0, 6000), col = c("red", "green", "blue", "purple", "yellow"), main = "fatalities
```



Tornadoes seem to be the harmful with respect to population health

- Creating a variable ecprop5 storing information about property damaged from different event types

from 1950 to 2011

```
ecprop <- with(stormdata, tapply(ecprop, EVTYPE, sum, na.rm = TRUE))
ecprop5 <- ecprop[ecprop> (10000000000)]
```

- Creating a variable eccrop5 storing information about crops damaged from different event types from 1950 to 2011

```
eccrop <- with(stormdata, tapply(eccrop, EVTYPE, sum, na.rm = TRUE))
eccrop5 <- eccrop[eccrop > 2000000000]
```

- Creating a variable ecprop501 storing information about property damaged from different event types from 2001 to 2011

```
ecprop01 <- with(sd2001, tapply(ecprop, EVTYPE, sum, na.rm = TRUE))
ecprop501 <- ecprop01[ecprop01> (10000000000)]
```

- Creating a variable eccrop501 storing information about crops damaged from different event types from 2001 to 2011

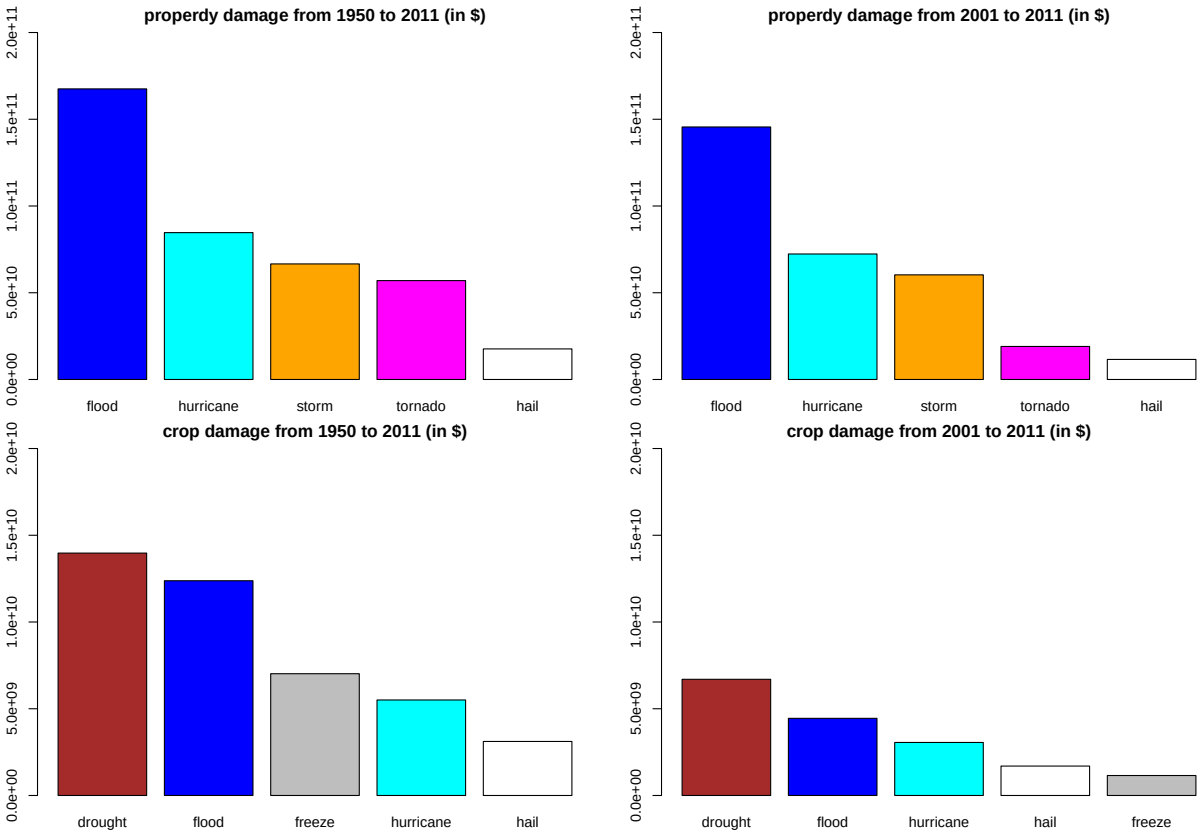
```
eccrop01 <- with(sd2001, tapply(eccrop, EVTYPE, sum, na.rm = TRUE))
eccrop501 <- eccrop01[eccrop01 > 1100000000]
```

- Sorting variables ecprop5, eccrop5, ecprop501, eccrop501 and making a plot.

```
e1 <- sort(ecprop5, decreasing = TRUE)
e2 <- sort(eccrop5, decreasing = TRUE)
e3 <- sort(ecprop501, decreasing = TRUE)
e4 <- sort(eccrop501, decreasing = TRUE)

par(mfrow = c(2,2), mar = c(2, 2, 2, 2))

barplot(e1, ylim = c (0, 200000000000), col = c("blue", "cyan", "orange", "magenta", "white"), main = "
barplot(e3, ylim = c (0, 200000000000), col = c("blue", "cyan", "orange", "magenta", "white"), main = "
barplot(e2, ylim = c (0, 20000000000), col = c("brown", "blue", "gray", "cyan", "white"), main = "crop
barplot(e4, ylim = c (0, 20000000000), col = c("brown", "blue", "cyan", "white", "gray"), main = "crop
```



Floods seem to be the most harmful for properties. **Drought** seem to be the most harmful for crops.