# rep_res_course_proj_final

Emmanuel

7/26/2021

## Synopsis

This is the final course project of reprodicible research course, which is part of the coursera specialization.

As we all know storms and natural events might affect both the economy and other sectors, causing several damages.

This project is about exploring a database from the NOA, which tracks the natural catastrophies and all the caracteristics as well as its impact on economy and on the crops across the US.

In this analysis we will try to figure out what is natural event that most impact the economy and the people's health.

## Data processing

## Loading libraries

Needed libraries are loaded

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## 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
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.6.3
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

### Setting the working directory and reading the document

The working directory is "C:/Users/rodriguezm.150/Documents/R". To read a bzfile we use the function "bzfile()"

```
setwd("C:/Users/rodriguezm.150/Documents/R")
storm_data <- read.csv(bzfile("repdata_data_StormData.csv.bz2"),header = T)
```

### Examining the structure of the database

```
str(storm_data)
```

```
## 'data.frame':    902297 obs. of  37 variables:
##  $ STATE__   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ BGN_DATE  : Factor w/ 16335 levels "1/1/1966 0:00:00",..: 6523 6523 4242 11116 2224 2224 2260 383
##  $ BGN_TIME  : Factor w/ 3608 levels "00:00:00 AM",..: 272 287 2705 1683 2584 3186 242 1683 3186 318
##  $ TIME_ZONE : Factor w/ 22 levels "ADT","AKS","AST",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ COUNTY    : num  97 3 57 89 43 77 9 123 125 57 ...
##  $ COUNTYNAME: Factor w/ 29601 levels "","5NM E OF MACKINAC BRIDGE TO PRESQUE ISLE LT MI",..: 13513
##  $ STATE     : Factor w/ 72 levels "AK","AL","AM",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ EVTYPE    : Factor w/ 985 levels "   HIGH SURF ADVISORY",..: 834 834 834 834 834 834 834 834 834 8
##  $ BGN_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ BGN_AZI   : Factor w/ 35 levels "","  N"," NW",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ BGN_LOCATI: Factor w/ 54429 levels "","- 1 N Albion",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_DATE  : Factor w/ 6663 levels "","1/1/1993 0:00:00",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_TIME  : Factor w/ 3647 levels "","  0900CST",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ COUNTYENDN: logi  NA NA NA NA NA NA ...
##  $ END_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ END_AZI   : Factor w/ 24 levels "","E","ENE","ESE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ END_LOCATI: Factor w/ 34506 levels "","- .5 NNW",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ LENGTH    : num  14 2 0.1 0 0 1.5 1.5 0 3.3 2.3 ...
##  $ WIDTH     : num  100 150 123 100 150 177 33 33 100 100 ...
##  $ F         : int  3 2 2 2 2 2 2 1 3 3 ...
##  $ MAG       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
##  $ PROPDMG   : num  25 2.5 25 2.5 2.5 2.5 2.5 2.5 25 25 ...
##  $ PROPDMGEXP: Factor w/ 19 levels "","-","?","+",..: 17 17 17 17 17 17 17 17 17 17 ...
##  $ CROPDMG   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CROPDMGEXP: Factor w/ 9 levels "","?","0","2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ WFO       : Factor w/ 542 levels "","  CI","$AC",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ STATEOFFIC: Factor w/ 250 levels "","ALABAMA, Central",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ZONENAMES : Factor w/ 25112 levels "","
##  $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
##  $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
##  $ LATITUDE_E: num  3051 0 0 0 0 ...
##  $ LONGITUDE_: num  8806 0 0 0 0 ...
##  $ REMARKS   : Factor w/ 436781 levels "","-2 at Deer Park\n",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
```

## Variables of interest

The variables to work with should be extracted and then work with a new dataset

```
vars <- c("EVTYPE","FATALITIES","INJURIES","PROPDMG","PROPDMGEXP","CROPDMG","CROPDMGEXP")
targ_data <- storm_data[,vars]
```

## Checking the first and last few recordings

```
head(targ_data)
```

```
##    EVTYPE FATALITIES INJURIES PROPDMG PROPDMGEXP CROPDMG CROPDMGEXP
## 1 TORNADO          0       15    25.0          K       0
## 2 TORNADO          0        0     2.5          K       0
## 3 TORNADO          0        2    25.0          K       0
## 4 TORNADO          0        2     2.5          K       0
## 5 TORNADO          0        2     2.5          K       0
## 6 TORNADO          0        6     2.5          K       0
```

```
tail(targ_data)
```

```
##                EVTYPE FATALITIES INJURIES PROPDMG PROPDMGEXP CROPDMG CROPDMGEXP
## 902292 WINTER WEATHER          0        0       0          K       0          K
## 902293      HIGH WIND          0        0       0          K       0          K
## 902294      HIGH WIND          0        0       0          K       0          K
## 902295      HIGH WIND          0        0       0          K       0          K
## 902296       BLIZZARD          0        0       0          K       0          K
## 902297     HEAVY SNOW          0        0       0          K       0          K
```

## Checking for missing values

Creating a function to summarize the amount of NA's in every column of interest

```
fun_nas <- function(x){
  sum(is.na(x))
}
sapply(targ_data[,2:6],fun_nas )
```

```
## FATALITIES   INJURIES    PROPDMG PROPDMGEXP    CROPDMG
##          0          0          0          0          0
```

## key variable

creating a new key-variale to differenciate the groups, and then getting all the main events by its keyword.

```
targ_data$EVENT <- "OTHER"

targ_data$EVENT[grep("HAIL",targ_data$EVTYPE,ignore.case = T)] <- "HAIL"
targ_data$EVENT[grep("HEAT",targ_data$EVTYPE,ignore.case = T)] <- "HEAT"
targ_data$EVENT[grep("FLOOD",targ_data$EVTYPE,ignore.case = T)] <- "FLOOD"
targ_data$EVENT[grep("WIND",targ_data$EVTYPE,ignore.case = T)] <- "WIND"
targ_data$EVENT[grep("STORM",targ_data$EVTYPE,ignore.case = T)] <- "STORM"
targ_data$EVENT[grep("TORNADO",targ_data$EVTYPE,ignore.case = T)] <- "TORNADO"
targ_data$EVENT[grep("WINTER",targ_data$EVTYPE,ignore.case = T)] <- "WINTER"
targ_data$EVENT[grep("RAIN",targ_data$EVTYPE,ignore.case = T)] <- "RAIN"
```

## Analysing what's in the PROPDMGEXP and CROPDMGEXP

```
sort(table(targ_data$PROPDMGEXP),decreasing = T)
```

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

```
sort(table(targ_data$CROPDMGEXP),decreasing = T)
```

```
##
##             K      M      k      0      B      ?      2      m
## 618413 281832   1994     21     19      9      7      1      1
```

## Organizing the EXP prefixes

Anything except K,M,B is a dollar

```
targ_data$PROPDMGEXP <- as.character(targ_data$PROPDMGEXP)
targ_data$CROPDMGEXP <- as.character(targ_data$CROPDMGEXP)
```

## Organizing the data in propdmgexp variable

Giving the column PROPDMGEXP the number depending on the prefix EXP and calculating a new column that contains the complete number

```
targ_data$PROPDMGEXP[!grepl("K|M|B",targ_data$PROPDMGEXP,ignore.case = T)] <- 0
targ_data$PROPDMGEXP[grepl("K",targ_data$PROPDMGEXP,ignore.case = T)] <- "3"
targ_data$PROPDMGEXP[grepl("M",targ_data$PROPDMGEXP,ignore.case = T)] <- "6"
targ_data$PROPDMGEXP[grepl("B",targ_data$PROPDMGEXP,ignore.case = T)] <- "9"
targ_data$PROPDMGEXP <- as.numeric(as.character(targ_data$PROPDMGEXP))
targ_data$property_dmg <- targ_data$PROPDMG * 10^targ_data$PROPDMGEXP
```

## Organizing the data in CROPDMGEXP variable

Giving the column CROPDMGEXP the number depending on the prefix EXP and calculating a new column that contains the complete number

```
targ_data$CROPDMGEXP[is.na(targ_data$CROPDMGEXP)] <- 0
targ_data$CROPDMGEXP[!grepl("K|M|B",targ_data$CROPDMGEXP,ignore.case = T)] <- 0
targ_data$CROPDMGEXP[grepl("K",targ_data$CROPDMGEXP,ignore.case = T)] <- "3"
targ_data$CROPDMGEXP[grepl("M",targ_data$CROPDMGEXP,ignore.case = T)] <- "6"
targ_data$CROPDMGEXP[grepl("B",targ_data$CROPDMGEXP,ignore.case = T)] <- "9"
targ_data$CROPDMGEXP <- as.numeric(as.character(targ_data$CROPDMGEXP))
targ_data$crop_dmg <- targ_data$CROPDMG*10^targ_data$CROPDMGEXP
```

## Values that most appear in propoerty

printing the first 10 property damage values that most appear in the data crop and property

```
sort(table(targ_data$property_dmg),decreasing = T)[1:10]
```

```
##
##      0   5000  10000   1000   2000  25000  50000   3000  20000  15000
## 663123  31731  21787  17544  17186  17104  13596  10364   9179   8617
```

```
sort(table(targ_data$crop_dmg),decreasing = T)[1:10]
```

```
##
##      0   5000  10000  50000  1e+05   1000   2000  25000  20000  5e+05
## 880198   4097   2349   1984   1233    956    951    830    758    721
```

## Exploring the data

aggregating fatalities and injuries by type of event

```
agg_fats_and_injs <- ddply(targ_data, .(EVENT),summarize,
                                  total=sum(INJURIES + FATALITIES,na.rm = T))
agg_fats_and_injs$type <- "Injuries and Fatalities"
```

separating fatalities by type of event

```
agg_fats <- ddply(targ_data,.(EVENT),summarise,total=sum(FATALITIES,na.rm = T))
agg_fats$type <- "Fatalities"
```

separating injuries by type of event

```
agg_injs <- ddply(targ_data,.(EVENT),summarise,total=sum(INJURIES,na.rm = T))
agg_injs$type <- "Injuries"
```

combining all the types, and joining the data

```
agg_health <- rbind(agg_fats,agg_injs)
health_by_event <- join(agg_fats,agg_injs,by="EVENT",type="inner")
```

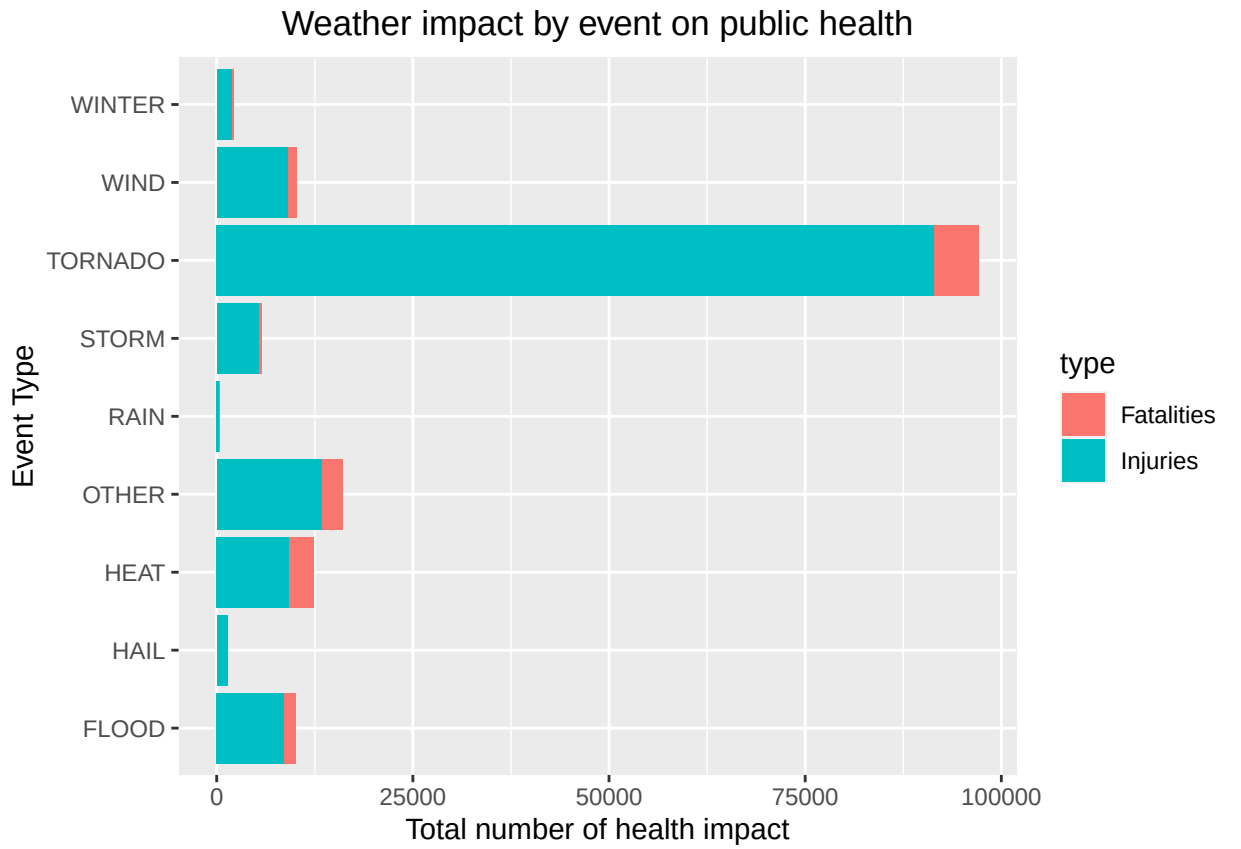
Aggregating events for economic variables

```
agg_prop_and_crop <- ddply(targ_data,.(EVENT),summarise,
                           total=sum(crop_dmg + property_dmg,na.rm = T))
agg_prop_and_crop$type <- "Property and Crop"
agg_prop <- ddply(targ_data,.(EVENT),summarise,
                  total=sum( property_dmg,na.rm = T))
agg_prop$type <- "Property"
agg_crop <- ddply(targ_data,.(EVENT),summarise,total=sum(crop_dmg,na.rm = T))
agg_crop$type <- "Crop"
agg_economic <- rbind(agg_crop,agg_prop)
economic_by_event <- join(agg_crop,agg_prop,by="EVENT",type = "inner")
```

## Results

## plotting fatalities and injuries

```
agg_health$EVENT <- as.factor(agg_health$EVENT)
healt_plot <- ggplot(agg_health,aes(x = EVENT,y = total,fill=type))+
  geom_bar(stat = "identity")+coord_flip()+xlab("Event Type")+
  ylab("Total number of health impact")+
  ggtitle("Weather impact by event on public health")+
  theme(plot.title = element_text(hjust = 0.5))
print(healt_plot)
```

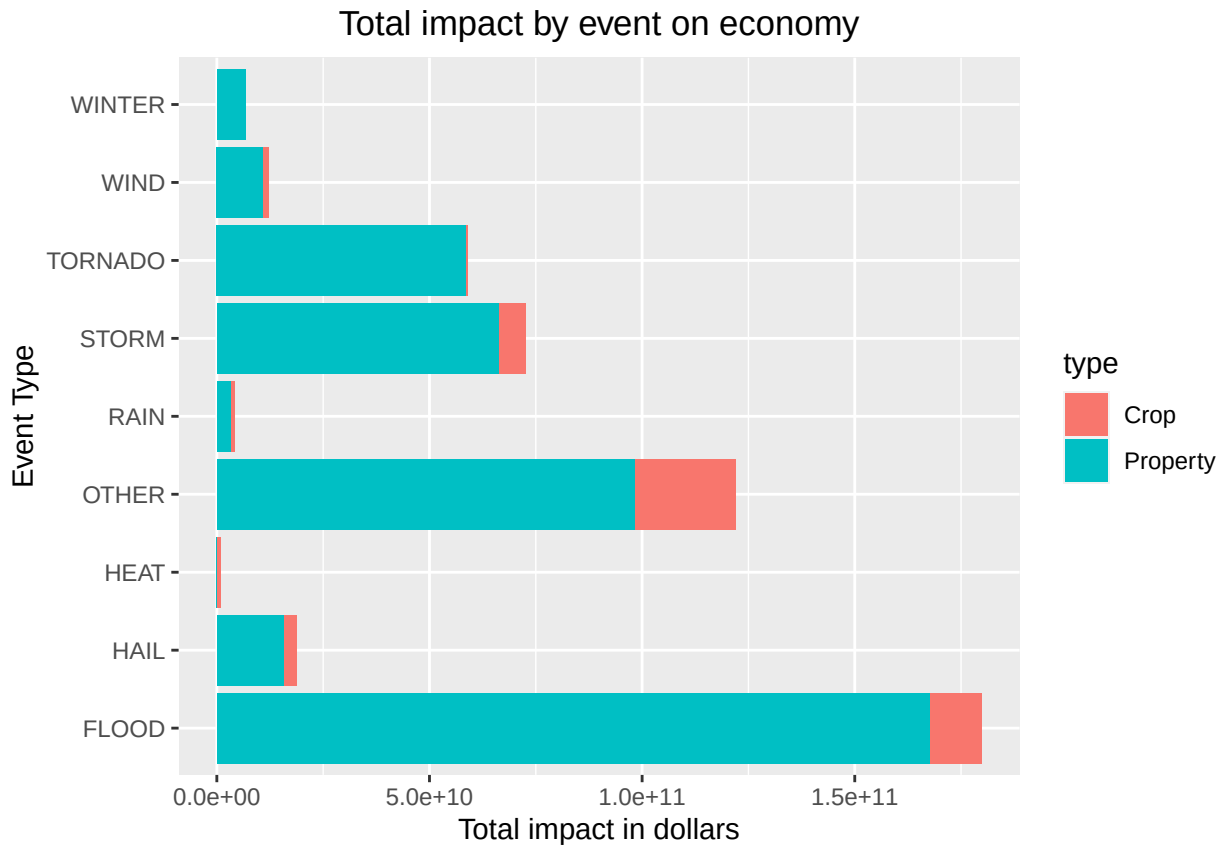



Analysis: As can be seen, the natural that most affect the human's health across the US is in far tornadoes.

### plotting crop and property impact on economy

```
agg_economic$EVENT <- as.factor(agg_economic$EVENT)

economic_plot <- ggplot(agg_economic,aes(x = EVENT,y = total,fill=type))+
  geom_bar(stat = "identity")+coord_flip()+xlab("Event Type")+
  ylab("Total impact in dollars")+
  ggtitle("Total impact by event on economy")+
  theme(plot.title = element_text(hjust = 0.5))
print(economic_plot)
```




Analysis: The Natural event that most have an impact(crops and properties) on the economy are Floods.