How weather related events impacts the US

Mario Azevedo

Sunday, May 24, 2015

### Synopsis

* Synopsis: Immediately after the title, there should be a synopsis which describes and summarizes your analysis in at most 10 complete sentences

### Data Processing

#### Loading the libraries that will be used

require(data.table)  
require(dplyr)  
require(tidyr)  
require(readr)  
require(ggplot2)  
require(stringr)  
require(grid)  
require(gridExtra)

#### Downloading and reading the data

#download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2",  
# "repdata-data-StormData.csv.bz2")  
data <- read.csv('repdata-data-StormData.csv.bz2')  
#Reading a data file prepared from the "Storm Data Event Table"  
type <- readLines('storm.txt')  
data <- data.table(data)  
nr1 <- nrow(data)

#### Cleaning the data

data[data$EVTYPE == "TSTM WIND"]$EVTYPE <- "THUNDERSTORM WIND"  
data[data$EVTYPE == "THUNDERSTORM WINDS"]$EVTYPE <- "THUNDERSTORM WIND"  
data[data$EVTYPE == "MARINE TSTM WIND"]$EVTYPE <- "MARINE THUNDERSTORM WIND"  
data[data$EVTYPE == "MARINE THUNDERSTORM WINDS"]$EVTYPE <- "MARINE THUNDERSTORM WIND"  
data <- filter(data,EVTYPE %in% type)  
perc1 <- 100\*nrow(data)/nr1

#### Preparing data for first question

totFat <- sum(data$FATALITIES,na.rm=TRUE)  
totInj <- sum(data$INJURIES,na.rm=TRUE)  
harm <- group\_by(data,EVTYPE) %>%   
 summarise(NofEV=n(),Fatalities = sum(FATALITIES),  
 pFat= 100\* sum(FATALITIES) / totFat,  
 Injuries=sum(INJURIES),  
 pInj= 100\* sum(INJURIES) / totInj,  
 sevIndex = (10\*sum(FATALITIES)+sum(INJURIES)))   
len <- nrow(harm)  
harm$rankF <- len - ave(harm$Fatalities,FUN=rank) + 1  
harm$rankI <- len - ave(harm$Injuries,FUN=rank) + 1  
harm$rankS <- len - ave(harm$sevIndex,FUN=rank) + 1  
harm

## Source: local data table [46 x 10]  
##   
## EVTYPE NofEV Fatalities pFat Injuries pInj  
## 1 TORNADO 60652 5633 41.3218897 91346 67.883444929  
## 2 THUNDERSTORM WIND 323346 701 5.1423122 9353 6.950647652  
## 3 HAIL 288661 15 0.1100352 1361 1.011422159  
## 4 WINTER STORM 11433 206 1.5111502 1321 0.981696306  
## 5 HEAVY RAIN 11723 98 0.7188967 251 0.186529730  
## 6 LIGHTNING 15754 816 5.9859155 5230 3.886655321  
## 7 DENSE FOG 1293 18 0.1320423 342 0.254156046  
## 8 RIP CURRENT 470 368 2.6995305 232 0.172409949  
## 9 FLASH FLOOD 54277 978 7.1742958 1777 1.320571034  
## 10 FUNNEL CLOUD 6839 0 0.0000000 3 0.002229439  
## .. ... ... ... ... ... ...  
## Variables not shown: sevIndex (dbl), rankF (dbl), rankI (dbl), rankS (dbl)

#### Preparing data for first question

data <- mutate(data,PROPDMGEXP = str\_to\_upper(str\_trim(PROPDMGEXP))) %>%   
 filter(PROPDMGEXP %in% c('K','M','B',''))   
  
data <- mutate(data,CROPDMGEXP = str\_to\_upper(str\_trim(CROPDMGEXP))) %>%   
 filter(CROPDMGEXP %in% c('K','M','B',''))  
  
perc2 <- 100\*nrow(data)/nr1

* There should be a section titled Data Processing which describes (in words and code) how the data were loaded into R and processed for analysis. In particular, your analysis must start from the raw CSV file containing the data. You cannot do any preprocessing outside the document. If preprocessing is time-consuming you may consider using the cache = TRUE option for certain code chunks.

### Results

* There should be a section titled Results in which your results are presented.
* You may have other sections in your analysis, but Data Processing and Results are required.
* The analysis document must have at least one figure containing a plot.
* Your analyis must have no more than three figures. Figures may have multiple plots in them (i.e. panel plots), but there cannot be more than three figures total.
* You must show all your code for the work in your analysis document. This may make the document a bit verbose, but that is okay. In general, you should ensure that echo = TRUE for every code chunk (this is the default setting in knitr).

result1 <- arrange(harm,rankF)[1:10] %>% mutate(EVTYPE = str\_to\_lower(EVTYPE))  
ord1 <- reorder(result1$EVTYPE,result1$Fatalities)  
p1 <- ggplot(result1,aes(x=ord1,y=Fatalities)) +   
 geom\_bar(stat="identity",position="dodge") +   
 theme(axis.text.x=element\_text(size=12,colour="black")) +  
 theme(axis.text.y=element\_text(size=12,colour="black")) +  
 theme(axis.title=element\_text(size=12)) +  
 xlab("") +   
 ylab("Fatalities") +  
 coord\_flip()  
  
result2 <- arrange(harm,rankI)[1:10] %>% mutate(EVTYPE = str\_to\_lower(EVTYPE))  
ord2 <- reorder(result2$EVTYPE,result2$Injuries)  
p2 <- ggplot(result2,aes(x=ord2,y=Injuries)) +   
 geom\_bar(stat="identity",position="dodge") +   
 theme(axis.text.x=element\_text(size=12,colour="black")) +  
 theme(axis.text.y=element\_text(size=12,colour="black")) +  
 theme(axis.title=element\_text(size=12)) +  
 xlab("") +   
 ylab("Injuries") +  
 coord\_flip()  
  
result3 <- arrange(harm,rankS)[1:10] %>% mutate(EVTYPE = str\_to\_lower(EVTYPE))  
ord3 <- reorder(result3$EVTYPE,result3$sevIndex)  
p3 <- ggplot(result3,aes(x=ord3,y=sevIndex)) +   
 geom\_bar(stat="identity",position="dodge") +   
 theme(axis.text.x=element\_text(size=12,colour="black")) +  
 theme(axis.text.y=element\_text(size=12,colour="black")) +  
 theme(axis.title=element\_text(size=12)) +  
 xlab("") +   
 ylab("Fatalities x 10 + Injuries") +  
 coord\_flip()  
  
grid.arrange(p1, p2, p3, ncol = 1,   
 main = "Most Harmful Types of Events",  
 #left="Event Type",  
 widths = unit(c(16), "cm"),  
 heights = unit(c(7,7,7), "cm"),just=c("top"))

