Reproducible Research: Peer Assessment 2

Health and Economic Impact of Weather Events in the US

Storms and other severe weather events can cause both public health and economic problems for communities and municipalities. Many severe events can result in fatalities, injuries, and property damage, and preventing such outcomes to the extent possible is a key concern.

Synopsis

The analysis on the storm event database revealed that tornadoes are the most dangerous weather event to the population health.

Data Processing

- The data is from a compression types of comma-separated-value file available here.
- The data is downloaded , extracted to the folder and read via CSV reader.
- Some of the data also converted into lower case , converted into dataframe via ddplay , combine via plyr and data preview .
- New basic function introduced to convert into exponential transformation
- Basic command line using KnitR (manual)

```
library(markdown)
library(knitr)
setwd("D:/Google Drive/Coursera/Assignment 5.1/R/coursera-repdata/project2")
knitr::opts_chunk$set(echo=FALSE, fig.path='D:/Google Drive/Coursera/Assignment 5.1/R/coursera-repo
knitr::knit2html("storm.analysis.Rmd", options=c("use_xhtml","smartypants","mathjax","highlight_coc
```

- Load all the required library

```
options( warn = -1 )
is.installed <- function(mypkg) is.element(mypkg, installed.packages()[,1])
if (is.installed('dplyr') == 'FALSE') {install.packages("dplyr");library(dplyr)} else{suppressMessages(l
if (is.installed('ggthemes') == 'FALSE') {install.packages("ggthemes");library(ggthemes)} else{suppressM
if (is.installed('scales') == 'FALSE') {install.packages("scales");library(scales)} else{suppressMessage
if (is.installed('RColorBrewer') == 'FALSE') {install.packages("RColorBrewer");suppressMessages(library
if (is.installed('lubridate') == 'FALSE') {install.packages("lubridate");library(lubridate)} else{suppre
if (is.installed('ggplot2') == 'FALSE') {install.packages("ggplot2");library(ggplot2)} else{suppressMess
if (is.installed('plyr') == 'FALSE') {install.packages("plyr");library(plyr)} else{suppressMessages(libr
if (is.installed('knitr') == 'FALSE') {install.packages("knitr");library(knitr)} else{suppressMessages(l
if (is.installed('lattice') == 'FALSE') {install.packages("lattice");library(lattice)} else{suppressMess
if (is.installed('RCurl') == 'FALSE') {install.packages("RCurl");library(RCurl)} else{suppressMessages(l
if (is.installed('reshape') == 'FALSE') {install.packages("reshape");library(reshape)} else{suppressMess
if (is.installed('car') == 'FALSE') {install.packages("car");library(car)} else{suppressMessages(library
if (is.installed('gridExtra') == 'FALSE') {install.packages("gridExtra");library(gridExtra)} else{suppre
```

```
if (is.installed('grid') == 'FALSE') {install.packages("grid");library(grid)} else{suppressMessages(lib
if (is.installed('xtable') == 'FALSE') {install.packages("xtable");library(xtable)} else{suppressMessag
```

- The first step is to read the data into a data frame.

```
options( warn = -1 )
curdir <-getwd()
file.url<-'http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2'
download.file(file.url,destfile=paste(curdir,'/repdata%2Fdata%2FStormData.csv.bz2',sep=""))
storm <- read.csv(bzfile(paste(curdir,'/repdata%2Fdata%2FStormData.csv.bz2',sep="")))
#storm <- read.csv(bzfile("c://repdata%2Fdata%2FStormData.csv.bz2"))
length(unique(storm$EVTYPE))
```

```
## [1] 985
```

- Get the no. of event types

```
length(unique(storm$EVTYPE))
```

```
## [1] 985
```

- Converting letters to lower casing

```
event_types <- tolower(storm$EVTYPE)
```

- Replace punctation characters with a space

```
event_types <- gsub("[[:blank:][:punct:]+]", " ", event_types)
```

- Get the unique of event types

```
length(unique(event_types))
```

```
## [1] 874
```

- Get the casualities

```
library(plyr)
casualties <- ddply(storm, .(EVTYPE), summarize,
                    fatalities = sum(FATALITIES),
                    injuries = sum(INJURIES))
```

- Find events which causing most injury and death

```
fatal_events <- head(casualties[order(casualties$fatalities, decreasing = T), ], 10)
injury_events <- head(casualties[order(casualties$injuries, decreasing = T), ], 10)
```

- Top 10 Fatal Events List

```r
fatal_events[, c("EVTYPE", "fatalities")]
```

```
##             EVTYPE fatalities
## 834        TORNADO       5633
## 130 EXCESSIVE HEAT       1903
## 153    FLASH FLOOD        978
## 275           HEAT        937
## 464      LIGHTNING        816
## 856      TSTM WIND        504
## 170          FLOOD        470
## 585    RIP CURRENT        368
## 359      HIGH WIND        248
## 19       AVALANCHE        224
```

- Top 10 Injury Events List

```r
injury_events[, c("EVTYPE", "injuries")]
```

```
##                EVTYPE injuries
## 834           TORNADO    91346
## 856         TSTM WIND     6957
## 170             FLOOD     6789
## 130    EXCESSIVE HEAT     6525
## 464         LIGHTNING     5230
## 275              HEAT     2100
## 427         ICE STORM     1975
## 153       FLASH FLOOD     1777
## 760 THUNDERSTORM WIND     1488
## 244              HAIL     1361
```

- Define function for exponent transformation and apply the transformation

```r
exp_transform <- function(e) {
    # h -> hundred, k -> thousand, m -> million, b -> billion
    if (e %in% c('h', 'H'))
        return(2)
    else if (e %in% c('k', 'K'))
        return(3)
    else if (e %in% c('m', 'M'))
        return(6)
    else if (e %in% c('b', 'B'))
        return(9)
    else if (!is.na(as.numeric(e))) # if a digit
        return(as.numeric(e))
    else if (e %in% c('', '-', '?', '+'))
        return(0)
    else {
        stop("Invalid exponent value.")
    }
}
```

```r
prop_dmg_exp <- sapply(storm$PROPDMGEXP, FUN=exp_transform)
storm$prop_dmg <- storm$PROPDMG * (10 ** prop_dmg_exp)
crop_dmg_exp <- sapply(storm$CROPDMGEXP, FUN=exp_transform)
storm$crop_dmg <- storm$CROPDMG * (10 ** crop_dmg_exp)
```

- Calculating loss by event type

```r
library(plyr)
econ_loss <- ddply(storm, .(EVTYPE), summarize,
                   prop_dmg = sum(prop_dmg),
                   crop_dmg = sum(crop_dmg))
```

- Removing events with no loss

```r
econ_loss <- econ_loss[(econ_loss$prop_dmg > 0 | econ_loss$crop_dmg > 0), ]
prop_dmg_events <- head(econ_loss[order(econ_loss$prop_dmg, decreasing = T), ], 10)
crop_dmg_events <- head(econ_loss[order(econ_loss$crop_dmg, decreasing = T), ], 10)
```

- Top 10 Property damage List

```r
prop_dmg_events[, c("EVTYPE", "prop_dmg")]
```

```
##                 EVTYPE     prop_dmg
## 153        FLASH FLOOD 6.820237e+13
## 786 THUNDERSTORM WINDS 2.086532e+13
## 834            TORNADO 1.078951e+12
## 244               HAIL 3.157558e+11
## 464          LIGHTNING 1.729433e+11
## 170              FLOOD 1.446577e+11
## 411  HURRICANE/TYPHOON 6.930584e+10
## 185           FLOODING 5.920825e+10
## 670        STORM SURGE 4.332354e+10
## 310         HEAVY SNOW 1.793259e+10
```

- Top 10 Crop damage List

```r
crop_dmg_events[, c("EVTYPE", "crop_dmg")]
```

```
##                EVTYPE    crop_dmg
## 95            DROUGHT 13972566000
## 170             FLOOD  5661968450
## 590       RIVER FLOOD  5029459000
## 427         ICE STORM  5022113500
## 244              HAIL  3025974480
## 402         HURRICANE  2741910000
## 411 HURRICANE/TYPHOON  2607872800
## 153       FLASH FLOOD  1421317100
## 140      EXTREME COLD  1292973000
## 212      FROST/FREEZE  1094086000
```

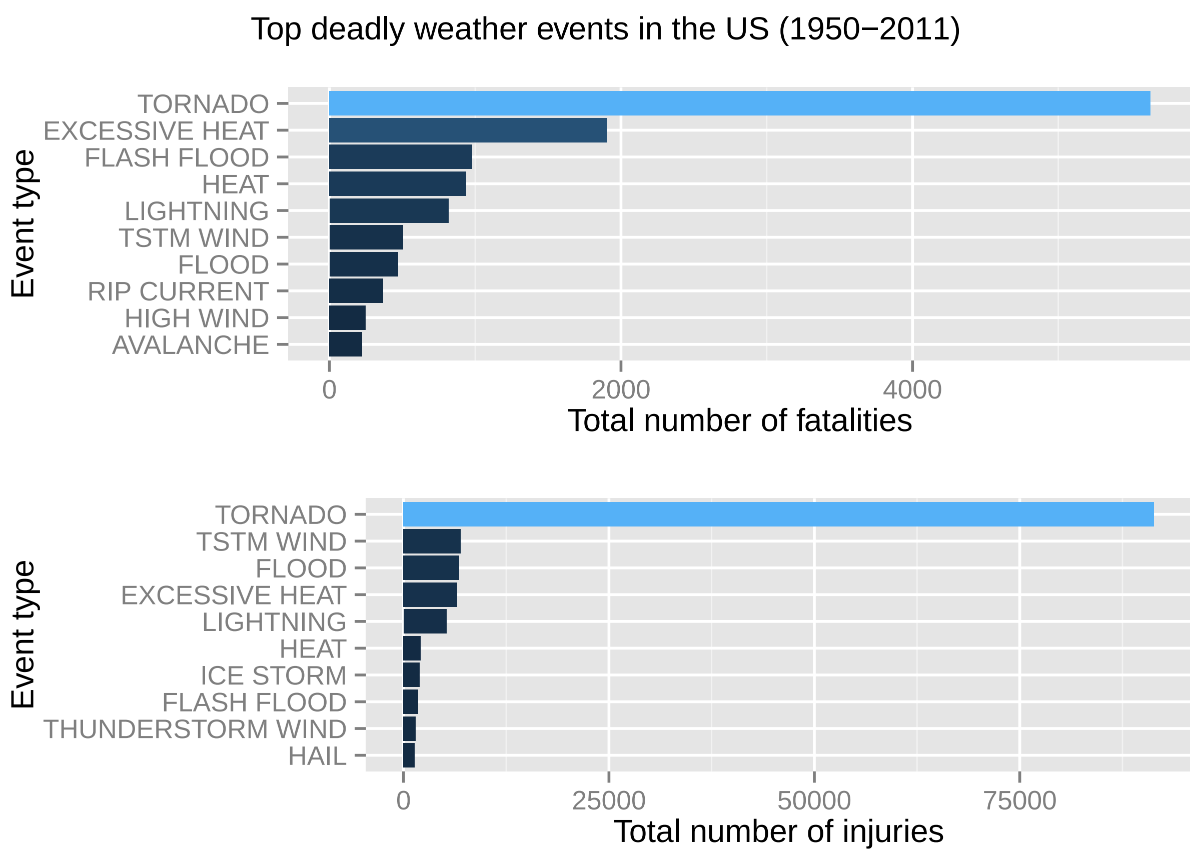
Results

- Set the levels in order and produce 2 types of graph and combined as Top deadly weather

```
library(ggplot2)
library(gridExtra)

p1 <- ggplot(data=fatal_events,
             aes(x=reorder(EVTYPE, fatalities), y=fatalities, fill=fatalities)) +
    geom_bar(stat="identity") +
    coord_flip() +
    ylab("Total number of fatalities") +
    xlab("Event type") +
    theme(legend.position="none")

p2 <- ggplot(data=injury_events,
             aes(x=reorder(EVTYPE, injuries), y=injuries, fill=injuries)) +
    geom_bar(stat="identity") +
    coord_flip() +
    ylab("Total number of injuries") +
    xlab("Event type") +
    theme(legend.position="none")
grid.arrange(p1, p2 , ncol=1, nrow=2, top = "Top deadly weather events in the US (1950-2011)")
```



- Set the levels in order and produce 2 types of graph and combined as Weather costs

```
library(ggplot2)
library(gridExtra)
# Set the levels in order
p1 <- ggplot(data=prop_dmg_events,
             aes(x=reorder(EVTYPE, prop_dmg), y=log10(prop_dmg), fill=prop_dmg )) +
    geom_bar(stat="identity") +
    coord_flip() +
    xlab("Event type") +
    ylab("Property damage in dollars (log-scale)") +
    theme(legend.position="none")

p2 <- ggplot(data=crop_dmg_events,
             aes(x=reorder(EVTYPE, crop_dmg), y=crop_dmg, fill=crop_dmg)) +
    geom_bar(stat="identity") +
    coord_flip() +
    xlab("Event type") +
    ylab("Crop damage in dollars") +
    theme(legend.position="none")

grid.arrange(p1, p2 , ncol=1, nrow=2, top = "Weather costs to the US economy (1950-2011)")
```