

# Reproducible Research - Week 4 Peer Project

Vignesh C Iyer

7/12/2020

---

## Synopsis

---

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.

This project involves exploring the U.S. National Oceanic and Atmospheric Administration's (NOAA) storm database. This database tracks characteristics of major storms and weather events in the United States, including when and where they occur, as well as estimates of any fatalities, injuries, and property damage.

---

## Assignment

---

The basic goal of this assignment is to explore the NOAA Storm Database and answer some basic questions about severe weather events. You must use the database to answer the questions below and show the code for your entire analysis. Your analysis can consist of tables, figures, or other summaries. You may use any R package you want to support your analysis.

---

## Data

---

The data for this assignment come in the form of a comma-separated-value file compressed via the bzip2 algorithm to reduce its size. You can download the file from the course web site:

### Storm Data

There is also some documentation of the database available. Here you will find how some of the variables are constructed/defined.

- [National Weather Service Storm Data Documentation](#)
- [National Climatic Data Center Storm Events FAQ](#)

The events in the database start in the year 1950 and end in November 2011. In the earlier years of the database there are generally fewer events recorded, most likely due to a lack of good records. More recent years should be considered more complete.

## Data Pre-processing

The Storm Data is fetched, downloaded to the local system and then its contents are read based on the code given below

```
# This section deals with the downloading the compressed file and
# extracting its contents.

stormData <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"

# The file is downloaded using the download.file function.
download.file(stormData, destfile = "../StormData.csv.bz2")

# reading data from the file
readStormData <- read.csv("../StormData.csv.bz2")

# Fetching column names of Storm Data using the colNames function
colnames(readStormData)

## [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" "PROPDGMG"
## [26] "PROPDMGEXP" "CROPDMG" "CROPDMGEXP" "WFO" "STATEOFFIC"
## [31] "ZONENAMES" "LATITUDE" "LONGITUDE" "LATITUDE_E" "LONGITUDE_"
## [36] "REMARKS" "REFNUM"

str(readStormData)
```

```
## 'data.frame': 902297 obs. of 37 variables:
## $ STATE_ : num 1 1 1 1 1 1 1 1 1 1 ...
## $ BGN_DATE : chr "4/18/1950 0:00:00" "4/18/1950 0:00:00" "2/20/1951 0:00:00" "6/8/1951 0:00:00" .
## $ BGN_TIME : chr "0130" "0145" "1600" "0900" ...
## $ TIME_ZONE : chr "CST" "CST" "CST" "CST" ...
## $ COUNTY : num 97 3 57 89 43 77 9 123 125 57 ...
## $ COUNTYNAME: chr "MOBILE" "BALDWIN" "FAYETTE" "MADISON" ...
## $ STATE : chr "AL" "AL" "AL" "AL" ...
## $ EVTYPE : chr "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
## $ BGN_RANGE : num 0 0 0 0 0 0 0 0 0 0 ...
## $ BGN_AZI : chr "" "" "" "" ...
## $ BGN_LOCATI: chr "" "" "" "" ...
## $ END_DATE : chr "" "" "" "" ...
## $ END_TIME : chr "" "" "" "" ...
## $ 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 : chr "" "" "" "" ...
## $ END_LOCATI: chr "" "" "" "" ...
## $ 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: chr "K" "K" "K" "K" ...
## $ CROPDMG : num 0 0 0 0 0 0 0 0 0 0 ...
## $ CROPDMGEXP: chr "" "" "" "" ...
## $ WFO : chr "" "" "" "" ...
## $ STATEOFFIC: chr "" "" "" "" ...
## $ ZONENAMES : chr "" "" "" "" ...
## $ 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 : chr "" "" "" "" ...
## $ REFNUM : num 1 2 3 4 5 6 7 8 9 10 ...
```

```
# Fetching first few rows of Storm Data
```

```
head(readStormData)
```

```
## STATE__ BGN_DATE BGN_TIME TIME_ZONE COUNTY COUNTYNAME STATE EVTYPE
## 1 1 4/18/1950 0:00:00 0130 CST 97 MOBILE AL TORNADO
## 2 1 4/18/1950 0:00:00 0145 CST 3 BALDWIN AL TORNADO
## 3 1 2/20/1951 0:00:00 1600 CST 57 FAYETTE AL TORNADO
## 4 1 6/8/1951 0:00:00 0900 CST 89 MADISON AL TORNADO
## 5 1 11/15/1951 0:00:00 1500 CST 43 CULLMAN AL TORNADO
## 6 1 11/15/1951 0:00:00 2000 CST 77 LAUDERDALE AL TORNADO
## BGN_RANGE BGN_AZI BGN_LOCATI END_DATE END_TIME COUNTY_END COUNTYENDN
## 1 0 0 0 NA
## 2 0 0 0 NA
## 3 0 0 0 NA
## 4 0 0 0 NA
## 5 0 0 0 NA
## 6 0 0 0 NA
## END_RANGE END_AZI END_LOCATI LENGTH WIDTH F MAG FATALITIES INJURIES PROPDMG
## 1 0 0 14.0 100 3 0 0 15 25.0
## 2 0 0 2.0 150 2 0 0 0 2.5
## 3 0 0 0.1 123 2 0 0 2 25.0
## 4 0 0 0.0 100 2 0 0 2 2.5
## 5 0 0 0.0 150 2 0 0 2 2.5
## 6 0 0 1.5 177 2 0 0 6 2.5
## PROPDMGEXP CROPDMG CROPDMGEXP WFO STATEOFFIC ZONENAMES LATITUDE LONGITUDE
## 1 K 0 3040 8812
## 2 K 0 3042 8755
## 3 K 0 3340 8742
## 4 K 0 3458 8626
## 5 K 0 3412 8642
## 6 K 0 3450 8748
## LATITUDE_E LONGITUDE_ REMARKS REFNUM
## 1 3051 8806 1
## 2 0 0 2
## 3 0 0 3
## 4 0 0 4
## 5 0 0 5
## 6 0 0 6
```

```
# fetching the unique event type in the Storm Data
```

```
head(unique(readStormData$EVTYPE))
```

```
## [1] "TORNADO" "TSTM WIND" "HAIL"
```

```
## [4] "FREEZING RAIN"          "SNOW"          "ICE STORM/FLASH FLOOD"
```

We notice that the Date format is that of a Character from the below code

```
class(readStormData$BGN_DATE)
```

```
## [1] "character"
```

We will convert it to Date format using the as.Date function and assign it to a new variable stormDate

```
readStormData$BGN_DATE <- as.Date(readStormData$BGN_DATE, format = "%m%d%Y %H:%m:%s")
class(readStormData$BGN_DATE)
```

```
## [1] "Date"
```

Getting the events type as a Data Frame

```
# subsetting the Storm Data
readStormData <- subset(readStormData,
                        select = c(EVTYPE, FATALITIES,
                                  INJURIES, PROPDGMG, PROPDMGEXP, CROPDMG,
                                  CROPDMGEXP))
```

---

**1. Across the United States, which types of events (as indicated in the EVTYPE variable) are most harmful with respect to population health?**

---

Since we have already subset the original data based on the EVTYPE, FATALITIES, INJURIES, PROPDGMG, PROPDMGEXP, CROPDMG and CROPDMGEXP we now need to process the data further in such a way that for each “EVTYPE” we need to find the FATALITIES and INJURIES.

Doing the above process would give us an insight as to which event type caused maximum fatalities and injuries.

```
library(dplyr)
```

```
##
## 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
```

```
# Aggregating and arranging the Fatalities and Injuries
```

```
stormDataFatalities <- arrange(
  aggregate(FATALITIES ~ EVTYPE, data = readStormData, sum),
  desc(FATALITIES), EVTYPE)[1:10,]
```

```
# Aggregated data of the Storm Fatalities based on the event type
stormDataFatalities
```

```
##           EVTYPE FATALITIES
## 1      TORNADO      5633
## 2 EXCESSIVE HEAT      1903
```

```
## 3    FLASH FLOOD    978
## 4        HEAT    937
## 5    LIGHTNING    816
## 6    TSTM WIND    504
## 7        FLOOD    470
## 8    RIP CURRENT    368
## 9    HIGH WIND    248
## 10   AVALANCHE    224
```

```
stormDataInjuries <- arrange(
  aggregate(INJURIES ~ EVTYPE, data = readStormData, sum),
  desc(INJURIES), EVTYPE)[1:10,]
```

```
# Aggregated data of the Storm Injuries based on the event type
stormDataInjuries
```

```
##          EVTYPE INJURIES
## 1    TORNADO    91346
## 2    TSTM WIND    6957
## 3     FLOOD    6789
## 4 EXCESSIVE HEAT    6525
## 5    LIGHTNING    5230
## 6        HEAT    2100
## 7    ICE STORM    1975
## 8    FLASH FLOOD    1777
## 9 THUNDERSTORM WIND    1488
## 10         HAIL    1361
```

From both the “stormDataFatalities” and “stormDataInjuries” we can see that event type “TORNADO” has registered the highest number of Fatalities and Injuries, now let is plot the same on the graph.

```
library(lattice)
# plotting the graphs for the Fatalities and Injuries
```

```
par(mfrow=c(1,2),mar=c(10,3,3,2))
```

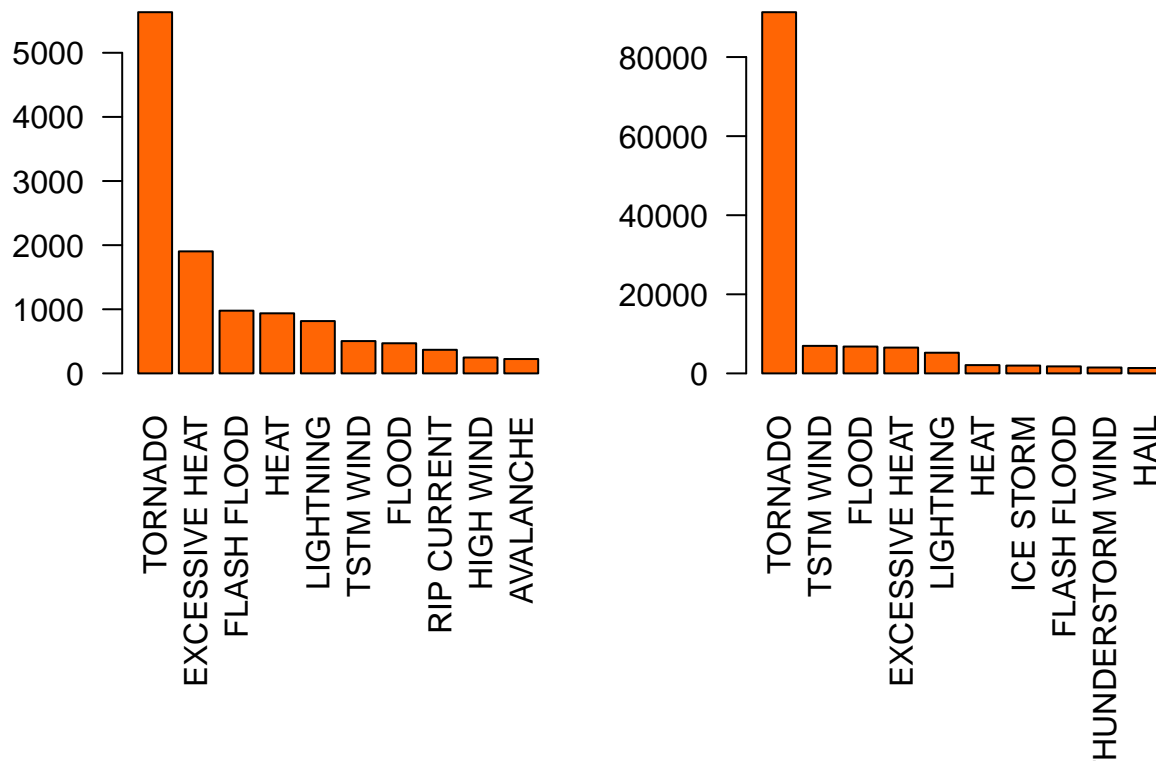
```
# Fatalities by event type
```

```
barplot(stormDataFatalities$FATALITIES,
  names.arg=stormDataFatalities$EVTYPE,
  las=2,
  col="#FF6504",
  ylab="Fatalities",
  main="Top 10 fatalities by weather event")
```

```
# Injuries by event type
```

```
barplot(stormDataInjuries$INJURIES,
  names.arg=stormDataInjuries$EVTYPE,
  las=2,
  col="#FF6504",
  ylab="Injuries",
  main="Top 10 Injuries by weather event")
```

## Top 10 fatalities by weather event      Top 10 Injuries by weather event



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

The greatest economic consequences can be measured by the columns PROPDMG, PROPDGMGEXP, CROPDMG, CROPDMGEXP.

The columns names denote the following

- PROPDMG → Property Damage
- CROPGMG → Crop Damage
- PROPDGMGEXP → Property Damage Exponent
- CROPDMGEXP → Crop Damage Exponent

We need to first associate the Damage caused to the Event type.

To do that we need to convert the notations "K", "M", "B", "m", "+", "0", "5", "6", "4", "2", "3", "h", "7", "H", "-", "1", "8" to their corresponding powers of 10 or exponential values.

For example:

- H represents 100
- K represents 1,000
- M represents 1,000,000
- B represents 1,000,000,000
- '6' can be converted as a million or  $10^6$
- '5' can be converted as a ten thousand or  $10^5$

and so on.. We do that by the following operation

```
unique(readStormData$PROPDMGEXP)

## [1] "K" "M" "" "B" "m" "+" "0" "5" "6" "?" "4" "2" "3" "h" "7" "H" "-" "1" "8"
# convert the notations "K","M","", "B","m","+","0","5","6","?","4","2","3","h","7",
# "H","-" "1","8" to their corresponding powers of 10 or exponential values.

readStormData$PROPEXP[readStormData$PROPDMGEXP == "K"] <- 1000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "M"] <- 1000000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "" ] <- 1
readStormData$PROPEXP[readStormData$PROPDMGEXP == "B"] <- 1000000000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "m"] <- 1000000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "0"] <- 1
readStormData$PROPEXP[readStormData$PROPDMGEXP == "5"] <- 100000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "6"] <- 1000000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "4"] <- 10000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "2"] <- 100
readStormData$PROPEXP[readStormData$PROPDMGEXP == "3"] <- 1000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "h"] <- 100
readStormData$PROPEXP[readStormData$PROPDMGEXP == "7"] <- 10000000
readStormData$PROPEXP[readStormData$PROPDMGEXP == "H"] <- 100
readStormData$PROPEXP[readStormData$PROPDMGEXP == "1"] <- 10
readStormData$PROPEXP[readStormData$PROPDMGEXP == "8"] <- 100000000
# Assigning '0' to invalid exponent data
readStormData$PROPEXP[readStormData$PROPDMGEXP == "+"] <- 0
readStormData$PROPEXP[readStormData$PROPDMGEXP == "-"] <- 0
readStormData$PROPEXP[readStormData$PROPDMGEXP == "?"] <- 0

class(readStormData$PROPEXP)

## [1] "numeric"
# Calculating the property damage value

stormPropertyDamage <- readStormData$PROPDMG * readStormData$PROPEXP
```

After having converted the notations for property damage we now need to do the same for the crop damage which will be achieved by the following code

```
unique(readStormData$CROPDMGEXP)

## [1] "" "M" "K" "m" "B" "?" "0" "k" "2"
# Assigning values for the crop exponent data
readStormData$CROPEXP[readStormData$CROPDMGEXP == "M"] <- 1000000
readStormData$CROPEXP[readStormData$CROPDMGEXP == "K"] <- 1000
readStormData$CROPEXP[readStormData$CROPDMGEXP == "m"] <- 1000000
readStormData$CROPEXP[readStormData$CROPDMGEXP == "B"] <- 1000000000
readStormData$CROPEXP[readStormData$CROPDMGEXP == "0"] <- 1
readStormData$CROPEXP[readStormData$CROPDMGEXP == "k"] <- 1000
readStormData$CROPEXP[readStormData$CROPDMGEXP == "2"] <- 100
readStormData$CROPEXP[readStormData$CROPDMGEXP == "" ] <- 1
# Assigning '0' to invalid exponent data
readStormData$CROPEXP[readStormData$CROPDMGEXP == "?"] <- 0

stormCropDamage <- readStormData$CROPDMG * readStormData$CROPEXP
```

Printing out the Column names

```
colnames(readStormData)
```

```
## [1] "EVTYPE"      "FATALITIES" "INJURIES"   "PROPDMG"    "PROPDMGEXP"  
## [6] "CROPDMG"     "CROPDMGEXP" "PROPEXP"    "CROPEXP"
```

```
# Calculating the total damage
```

```
readStormData$stormTotalDamage <- stormPropertyDamage + stormCropDamage
```

```
colnames(readStormData)
```

```
## [1] "EVTYPE"      "FATALITIES"   "INJURIES"     "PROPDMG"  
## [5] "PROPDMGEXP"  "CROPDMG"      "CROPDMGEXP"   "PROPEXP"  
## [9] "CROPEXP"     "stormTotalDamage"
```

```
# Finding the top 10 events based on which the maximum economic destruction has occurred
```

```
propertydamage <- arrange(  
  aggregate(  
    stormPropertyDamage ~ EVTYPE,  
    data=readStormData, sum),  
  desc(stormPropertyDamage), EVTYPE)[1:10,]
```

```
propertydamage
```

```
##           EVTYPE stormPropertyDamage  
## 1           FLOOD      144657709807  
## 2 HURRICANE/TYPHOON      69305840000  
## 3           TORNADO      56947380617  
## 4       STORM SURGE      43323536000  
## 5       FLASH FLOOD      16822673979  
## 6             HAIL      15735267513  
## 7       HURRICANE      11868319010  
## 8   TROPICAL STORM      7703890550  
## 9       WINTER STORM      6688497251  
## 10          HIGH WIND      5270046260
```

```
cropdamage <- arrange(  
  aggregate(  
    stormCropDamage ~ EVTYPE,  
    data=readStormData, sum),  
  desc(stormCropDamage), EVTYPE)[1:10,]
```

```
cropdamage
```

```
##           EVTYPE stormCropDamage  
## 1          DROUGHT      13972566000  
## 2           FLOOD      5661968450  
## 3    RIVER FLOOD      5029459000  
## 4        ICE STORM      5022113500  
## 5           HAIL      3025954473  
## 6       HURRICANE      2741910000  
## 7 HURRICANE/TYPHOON      2607872800  
## 8       FLASH FLOOD      1421317100  
## 9    EXTREME COLD      1292973000
```



```
## 10      FROST/FREEZE      1094086000
```

```
totaldamage <- arrange(  
  aggregate(  
    stormTotalDamage ~ EVTYPE,  
    data=readStormData, sum),  
  desc(stormTotalDamage),EVTYPE)[1:10,]
```

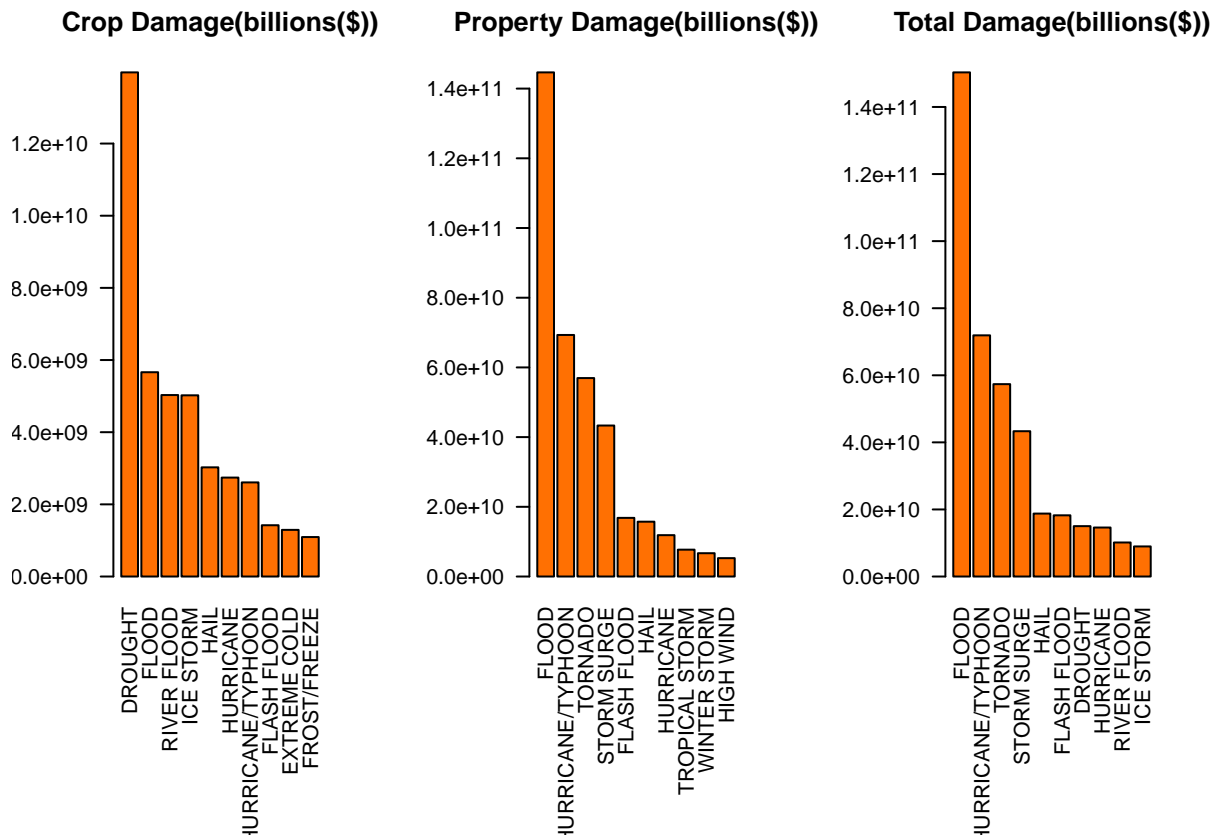
```
totaldamage
```

```
##           EVTYPE stormTotalDamage  
## 1           FLOOD      150319678257  
## 2 HURRICANE/TYPHOON      71913712800  
## 3           TORNADO      57362333887  
## 4       STORM SURGE      43323541000  
## 5           HAIL       18761221986  
## 6     FLASH FLOOD      18243991079  
## 7         DROUGHT      15018672000  
## 8         HURRICANE      14610229010  
## 9       RIVER FLOOD      10148404500  
## 10          ICE STORM       8967041360
```

There is a certain level of damage and destruction that occurs during any sort of natural calamity which amounts to certain economical losses.

Plotting the graphs for Property, Crop and total damage

```
library(lattice)  
library(dplyr)  
par(mfrow=c(1,3),mar=c(10,4,4,4))  
  
# Plotting CROP Damage in billions($) based on the top ten event types  
barplot(cropdamage$stormCropDamage,  
  names.arg = cropdamage$EVTYPE,  
  las = 2,  
  col="#FF7002",  
  main="Crop Damage(billions($))")  
  
# Plotting PROPERTY Damage in billions($) based on the top ten event types  
barplot(propertydamage$stormPropertyDamage,  
  names.arg = propertydamage$EVTYPE,  
  las = 2,  
  col = "#FF7002",  
  main="Property Damage(billions($))")  
  
# Plotting TOTAL damage in billions($) based on the top ten event types  
barplot(totaldamage$stormTotalDamage,  
  names.arg = totaldamage$EVTYPE,  
  las = 2,  
  col = "#FF7002",  
  main = "Total Damage(billions($))")
```



## Results

From the above plots we can conclude the following:

- The maximum number of fatalities reported was 5633 and injuries was 91346 all mainly due to tornadoes
- The crops suffered maximum damage during the drought season wherein the losses were close to \$14 billion.
- The damage to property was maximum during floods amounting to \$14.4 billion
- On the whole damage to both Crops and property was maximum during times when there were floods which amounted to \$15 billion