Reproducible Research, Project 2

Randall Helms

22 November 2016

Introduction

The US National Oceanic and Atmospheric Administration's (NOAA) storm database provides a rich data set on which to analyze the impact of severe weather events in the United States.

In this analysis, we have taken data covering the years 1950 to 2011, and then analyzed which types of severe weather events have had the biggest impact on human life (both injuries and fatalities) as well as the biggest financial impact on property and crops.

Data loading and first impressions

Let's start this process by loading the data into R using the fread function from `data.table`:

```
library(data.table)
library(plyr)
library(ggplot2)
library(scales)

stormdataALL <- fread("repdata_data_StormData.csv",sep=",",header=TRUE)
```

Let's quickly check the dimensions and structure of this data table:

```
dim(stormdataALL)
```

```
## [1] 902297     37
```

```
str(stormdataALL)
```

```
## Classes 'data.table' and '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         : chr  "3" "2" "2" "2" ...
##  $ 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 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Wow, that's a big and complicated file! There's a ton of information there, much of it interesting, no doubt, but a lot of it that is irrelevant to answering our original questions.

Since working with such a large file will require a lot of processing time and resources, it therefore makes sense to transform the data table into a more manageable size.

Data Transformation

Since we don't need all of those columns to answer the two questions, let's create a new data table using by subsetting only those columns that we actually need:

```
stormdata <- stormdataALL[,.(EVTYPE,FATALITIES, INJURIES, PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP)]
```

Let's check that that has worked:

```
head(stormdata)
```

```
##     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
```

As we can see, there is no neat way to calculate the costs of damage to property and crops with the way this table is currently set up.

Therefore, we need to now create for each row new columns containing the cost of the relevant damage.

Here's how we can do that:

```
#create formula for calculating cost
  #each letter in the *EXP columns represents a multiple of 100, so H is 100, K is 1000, etc

value <- function(x) {
  if (x %in% c("h", "H"))
    return(2)
  else if (x %in% c("k", "K"))
```

```
    return(3)
  else if (x %in% c("m", "M"))
    return(6)
  else if (x %in% c("b", "B"))
    return(9)
  else if (!is.na(as.numeric(x)))
    return(as.numeric(x))
  else if (x %in% c("", "-", "?", "+"))
    return(0)
  else {
    stop("Invalid value.")
  }
}

#apply formula to each cost type

propCost <- sapply(stormdata$PROPDMGEXP,FUN=value)
cropCost <- sapply(stormdata$CROPDMGEXP,FUN=value)

#create new columns with the damage costs in  a numeric format

stormdata$property_damage <- stormdata$PROPDMG * (10 ** propCost)
stormdata$crop_damage <- stormdata$CROPDMG * (10 ** cropCost)

#remove property and crop damage columns now that we have the costs expressed numerically

stormdata <- stormdata[,c("PROPDMG","PROPDMGEXP","CROPDMG","CROPDMGEXP"):=NULL]
```

With this done, let's check the structure of stormdata again:

```
str(stormdata)
```

```
## Classes 'data.table' and 'data.frame':   902297 obs. of  5 variables:
##  $ EVTYPE         : chr  "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
##  $ FATALITIES     : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES       : num  15 0 2 2 2 6 1 0 14 0 ...
##  $ property_damage: num  25000 2500 25000 2500 2500 2500 2500 2500 25000 25000 ...
##  $ crop_damage    : num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Since we have one character vector and the rest are numeric, it looks like now is a good time to sum the values of each by the event type:

```
stormdata <- stormdata[,lapply(.SD,sum),by=EVTYPE]
```

Now that we have a much tidier data set, let's add in additional columns to sum up the total casualties (injuries plus fatalities) and the total damage costs (property damage plus crop damage), and also tidy the names up a little:

```
#add new columns for totals

stormdata$casualties <- stormdata$FATALITIES + stormdata$INJURIES
stormdata$totalcost <- stormdata$property_damage + stormdata$crop_damage

#change column names
```

```r
colnames(stormdata) <- c('event_type','fatalities','injuries','property_damage','crop_damage','casualti
```

Let's check the head of `stormdata` to see how the data table looks now:

```r
head(stormdata)
```

```
##                event_type fatalities injuries property_damage crop_damage
## 1:                TORNADO       5633    91346     56947380677   414953270
## 2:              TSTM WIND        504     6957      4484928495   554007350
## 3:                   HAIL         15     1361     15735267513  3025954473
## 4:          FREEZING RAIN          7       23         8111500           0
## 5:                   SNOW          5       29        14762550       10000
## 6: ICE STORM/FLASH FLOOD           0        2               0           0
##    casualties  total_cost
## 1:      96979 57362333947
## 2:       7461  5038935845
## 3:       1376 18761221986
## 4:         30     8111500
## 5:         34    14772550
## 6:          2           0
```

It looks pretty good now, however in order to do the final analysis let's create two new data tables, one covering harm to people, and the other covering financial impacts.

To make the analysis even more concise, we will also further filter these data tables down to the top 10 most harmful event types, with an additional row for 'other' aggregating the impact of all other event types:

```r
#create summary data tables for each question

stormdata_casualties <- stormdata[,c(1:3,6),with=FALSE]
stormdata_costs <- stormdata[,c(1,4:5,7),with=FALSE]

#reorder each data table in descending order by totals

stormdata_casualties <- stormdata_casualties[order(-casualties)]
stormdata_costs <- stormdata_costs[order(-total_cost)]

#rework the data tables to have a top 10 and an 'other' row, summarizing the values of everything outsi

top10_casualties <- stormdata_casualties[1:10,]
top10_costs <- stormdata_costs[1:10,]

other_casualties <- stormdata_casualties[11:985,]
other_costs <- stormdata_costs[11:985,]

#create the 'other' row for the casualties set

oc1 <- as.data.frame(other_casualties)
oc1 <- colSums(oc1[,2:4],na.rm=TRUE)
oc1 <- transpose(as.data.frame(oc1))
colnames(oc1) <- c('fatalities','injuries','casualties')
oc1$event_type <- 'OTHER'
oc1 <- oc1[c(4,1,2,3)]

#combine the other row with the top 10 for casualties
```

```
stormdata_casualties <- rbind(top10_casualties,oc1)

#create the 'other' row for the costs set

oc2 <- as.data.frame(other_costs)
oc2 <- colSums(oc2[,2:4],na.rm=TRUE)
oc2 <- transpose(as.data.frame(oc2))
colnames(oc2) <- c('property_damage','crop_damage','total_cost')
oc2$event_type <- 'OTHER'
oc2 <- oc2[c(4,1,2,3)]

#combine the other row with the top 10 for costs

stormdata_costs <- rbind(top10_costs,oc2)
```

Now our data tables are down to 11 rows - a big change from the 900,000+ we started with!

Data Analysis Results - Human Health Impacts

Now that we have shrunk our data down to a much more manageable size, we can add a few different ratios as columns in order to help us understand the health impacts of the different types of extreme weather:

```
#which percent of all casualties did the event account for?

stormdata_casualties$pct_casualties <- 100 * (stormdata_casualties$casualties / sum(stormdata_casualtie

#which percent of all fatalities did the event account for?

stormdata_casualties$pct_fatalities <- 100 * (stormdata_casualties$fatalities / sum(stormdata_casualtie

#which percent of all injuries did the event account for?

stormdata_casualties$pct_injuries <- 100 * (stormdata_casualties$injuries / sum(stormdata_casualties$in

#what percent of casualties were fatalities?

stormdata_casualties$ratio_fatalities <- 100 * (stormdata_casualties$fatalities / stormdata_casualties$

#what percent of casualties were injuries?

stormdata_casualties$ratio_injuries <- 100 - stormdata_casualties$ratio_fatalities
```

Now let's have a look at the table overall:

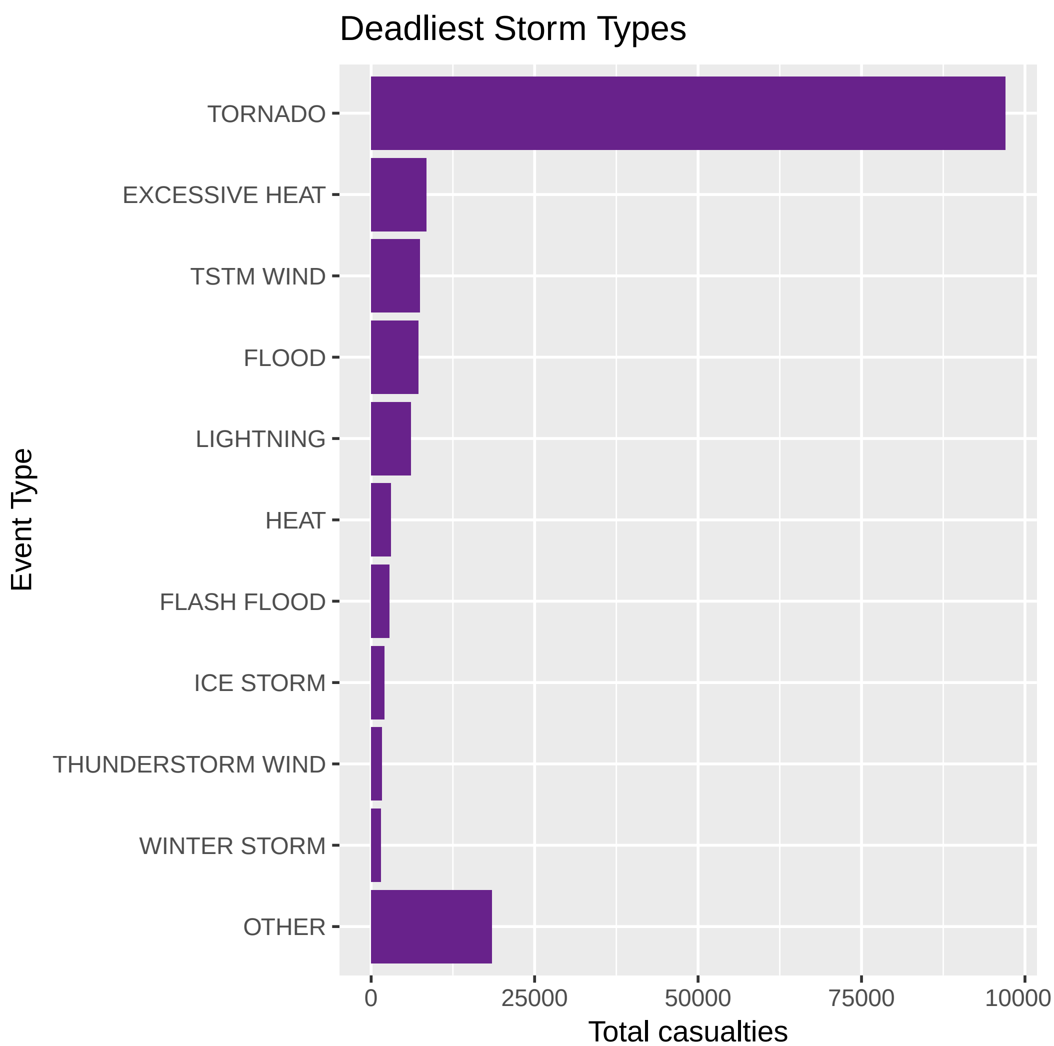
```
##           event_type fatalities injuries casualties pct_casualties
##  1:          TORNADO       5633    91346      96979     62.2966089
##  2:   EXCESSIVE HEAT       1903     6525       8428      5.4139125
##  3:        TSTM WIND        504     6957       7461      4.7927386
##  4:            FLOOD        470     6789       7259      4.6629795
##  5:        LIGHTNING        816     5230       6046      3.8837820
##  6:             HEAT        937     2100       3037      1.9508842
##  7:      FLASH FLOOD        978     1777       2755      1.7697353
##  8:        ICE STORM         89     1975       2064      1.3258561
##  9: THUNDERSTORM WIND        133     1488       1621      1.0412853
```

```
## 10:      WINTER STORM        206     1321       1527      0.9809023
## 11:             OTHER       3476    15020      18496     11.8813153
##     pct_fatalities pct_injuries ratio_fatalities ratio_injuries
##  1:     37.1937933   65.0019925         5.808474       94.19153
##  2:     12.5652030    4.6432028        22.579497       77.42050
##  3:      3.3278310    4.9506148         6.755127       93.24487
##  4:      3.1033344    4.8310657         6.474721       93.52528
##  5:      5.3879168    3.7216782        13.496527       86.50347
##  6:      6.1868603    1.4943641        30.852815       69.14718
##  7:      6.4575768    1.2645167        35.499093       64.50091
##  8:      0.5876527    1.4054139         4.312016       95.68798
##  9:      0.8781776    1.0588637         8.204812       91.79519
## 10:      1.3601849    0.9400262        13.490504       86.50950
## 11:     22.9514691   10.6882614        18.793253       81.20675
```

And now let's plot the casualties on a bar chart using `ggplot2`:

```r
stormdata_casualties$order <- 1:11

ggplot(stormdata_casualties,aes(x=reorder(event_type,-order),y=casualties))+
  geom_bar(fill="darkorchid4",stat="identity")+
  coord_flip()+
  ylab("Total casualties")+
  xlab("Event Type")+
  ggtitle("Deadliest Storm Types")
```



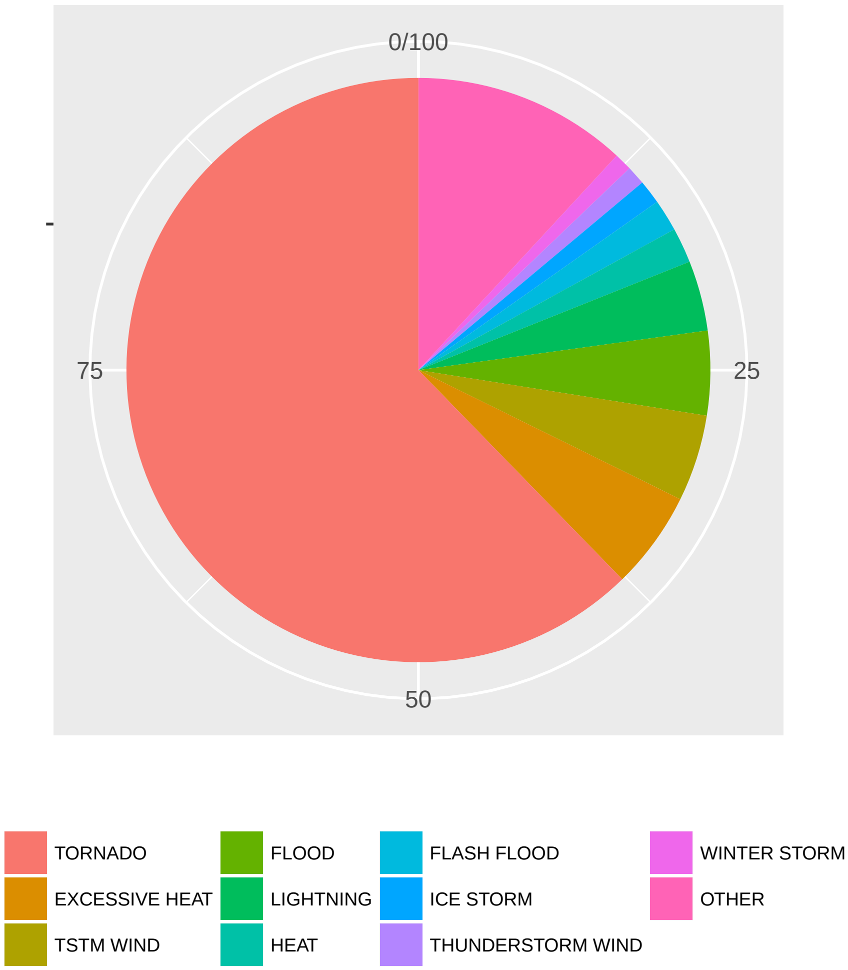
Next we can plot the proportion of casualties on a pie chart:

```
#convert event_type to a factor

stormdata_casualties$event_type <- factor(stormdata_casualties$event_type, levels = stormdata_casualtie

ggplot(stormdata_casualties,aes(x="",y=pct_casualties,fill=event_type))+
geom_bar(width = 1, stat = "identity")+
coord_polar(theta = "y")+
xlab("")+
ylab("")+
ggtitle("Percent of casualties by event type")+
  theme(legend.position = "bottom",legend.title = element_blank(),legend.text=element_text(size=6.5))
```

Percent of casualties by event type



Based on this data, there are a few conclusions that we can draw:

1. Tornados are, by some distance, the deadliest natural disaster in terms of both fatalities and injuries.
2. Flash floods are the most lethal types of severe storm event - 35% of casualties of flash floods are fatalities, a ratio eight times higher than the least lethal event, ice storms, where only 4.3% of people affected died.
3. The top 10 deadliest event types accounted for over 88% of all casualties in this time period - the other 975 only accounted for 12%

Data Analysis Results - Financial Impacts

Now we can perform a similar analysis on the financial impacts of different types of severe weather events:

```
#which percent of all costs did the event account for?

stormdata_costs$pct_costs <- 100 * (stormdata_costs$total_cost / sum(stormdata_costs$total_cost))

#which percent of all property damage did the event account for?

stormdata_costs$pct_prop_damage <- 100 * (stormdata_costs$property_damage / sum(stormdata_costs$property
```

```r
#which percent of all crop damage did the event account for?

stormdata_costs$pct_crop_damage <- 100 * (stormdata_costs$crop_damage / sum(stormdata_costs$crop_damage)

#what percent of costs were property damage?

stormdata_costs$ratio_prop_damage <- 100 * (stormdata_costs$property_damage / stormdata_costs$total_cos

#what percent of costs were injuries?

stormdata_costs$ratio_crop_damage <- 100 - stormdata_costs$ratio_prop_damage
```

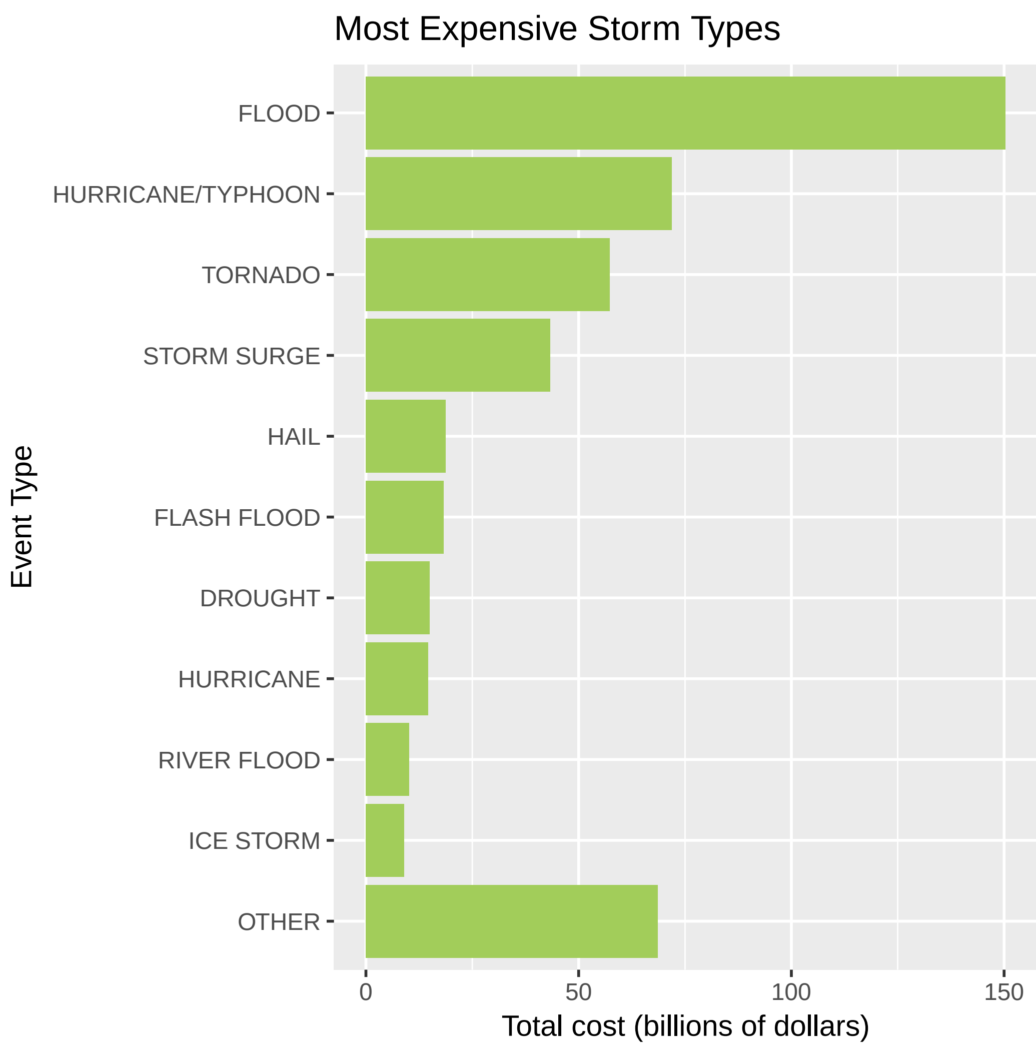
Now let's have a look at the table overall:

```
##            event_type property_damage crop_damage   total_cost pct_costs
##  1:             FLOOD    144657709807  5661968450 150319678257 31.491835
##  2: HURRICANE/TYPHOON     69305840000  2607872800  71913712800 15.065857
##  3:           TORNADO     56947380677   414953270  57362333947 12.017356
##  4:       STORM SURGE     43323536000        5000  43323541000  9.076242
##  5:              HAIL     15735267513  3025954473  18761221986  3.930459
##  6:       FLASH FLOOD     16822673979  1421317100  18243991079  3.822099
##  7:           DROUGHT      1046106000 13972566000  15018672000  3.146398
##  8:         HURRICANE     11868319010  2741910000  14610229010  3.060830
##  9:       RIVER FLOOD      5118945500  5029459000  10148404500  2.126081
## 10:         ICE STORM      3944927860  5022113500   8967041360  1.878587
## 11:             OTHER     59454162423  9206072588  68660235011 14.384256
##     pct_prop_damage pct_crop_damage ratio_prop_damage ratio_crop_damage
##  1:      33.7807821    1.153052e+01          96.23338      3.766618e+00
##  2:      16.1844501    5.310896e+00          96.37361      3.626392e+00
##  3:      13.2984758    8.450465e-01          99.27661      7.233898e-01
##  4:      10.1170061    1.018243e-05          99.99999      1.154107e-05
##  5:       3.6745338    6.162314e+00          83.87123      1.612877e+01
##  6:       3.9284673    2.894492e+00          92.20940      7.790604e+00
##  7:       0.2442889    2.845494e+01           6.96537      9.303463e+01
##  8:       2.7715156    5.583861e+00          81.23294      1.876706e+01
##  9:       1.1953873    1.024242e+01          50.44089      4.955911e+01
## 10:       0.9212281    1.022746e+01          43.99364      5.600636e+01
## 11:      13.8838649    1.874804e+01          86.59184      1.340816e+01
```

And now let's plot the total costs on a bar chart using ggplot2:

```r
stormdata_costs$order <- 1:11

ggplot(stormdata_costs,aes(x=reorder(event_type,-order),y=total_cost / 1000000000))+
  geom_bar(fill="darkolivegreen3",stat="identity")+
  coord_flip()+
  ylab("Total cost (billions of dollars)")+
  xlab("Event Type")+
  ggtitle("Most Expensive Storm Types")
```

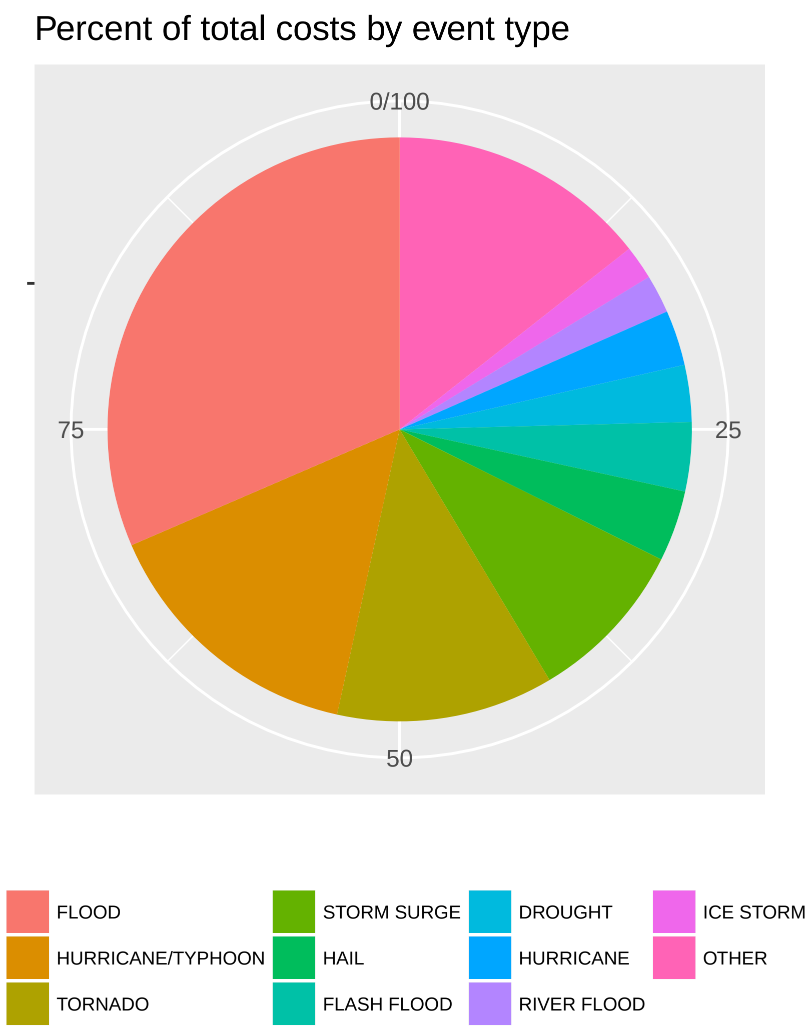


Next we can plot the proportion of costs on a pie chart:

```
#convert event_type to a factor

stormdata_costs$event_type <- factor(stormdata_costs$event_type, levels = stormdata_costs$event_type)

ggplot(stormdata_costs,aes(x="",y=pct_costs,fill=event_type))+
geom_bar(width = 1, stat = "identity")+
coord_polar(theta = "y")+
xlab("")+
ylab("")+
ggtitle("Percent of total costs by event type")+
  theme(legend.position = "bottom",legend.title = element_blank(),legend.text=element_text(size=6.5))
```



Now, let's do some analysis of the types of costs involved in extreme weather events:

1. Floods are the most expensive storm types, but since they only account for 30% of all costs, this is much less lopsided than the deadliness of tornados in terms of impact on people.
2. Property damage is much more expensive than crop damage both overall (it accounts for 90% of total costs), as well as individually for most extreme weather events. The main exception (unsurprisingly) is drought, where property damage accounted for only 6% of total costs.
3. The top 10 most expensive event types accounted for over 85% of all casualties in this time period - the other 975 only accounted for 15%.