An Analysis of the NOAA Storm Data Set

Coursera - Reproducible Research - July 2014

Mario Ibanez

The purpose of this analysis is to answer two key questions: Which weather events were most harmful to the population health, and which weather events had the greatest economic consequences. Harm to population health is measured in terms of injuries and fatalities, while economic consequences are measured in terms of dollar amounts. In order to come up with reasonable conclusions, a certain amount of cleaning and preparation of the data had to first be performed, which will also be included in this report. The majority of preparation involved restricting our scope to just certain columns, analyzing only those weather events that occured in 1996 or more recently, and creating a few new columns in order to help with analysis later on.

Data Processing

The data was obtained through a link provided within the Coursera course called Reproducible Research (July, 2014). The unzipped file is approximately 500mb and consists of 37 columns and 902297 rows.

```
## Load the package "stringdist" for use later.
library(stringdist)
require(stringdist)
```

```
## Download the file from the link given by the Reproducible Research Coursera course.
## fileURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
## destinationFile <- "/Users/Mario/StormData.csv"
## download.file(fileURL, destinationFile, method="curl")

## Read the .csv file into a data frame.
data <- read.csv("StormData.csv")
```

As mentioned, there are 37 columns and here are their column names:

```
## Returns the names of the columns in the data frame.
names(data)
```

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

In order to keep this analysis focused, only this subset of columns will be kept in the data frame:

1. “STATE___"
2. “BGN_DATE”
3. “BGN_TIME”
4. “STATE”
5. “EVTYPE”
6. “F”
7. “MAG”
8. “FATALITIES”
9. “INJURIES”
10. “PROPDMG”
11. “PROPDMGEXP”
12. “CROPDMG”
13. “CROPDMGEXP”

Please see the Appendix at the end of this document for more information about the meaning of each of these.

So we will next go ahead and subset the data frame to keep just these columns.

```
## Subset the data frame to include only the columns of interest.  The new data frame will be called "da
data_c <- data[c("STATE__","BGN_DATE","BGN_TIME","STATE","EVTYPE","F","MAG","FATALITIES","INJURIES","PRO
```

The next thing to do is create the “dictionary” of the 48 official event types. The 900+ event types in the original data set will be mapped to one of these 48 official event types using the amatch() function in the “stringdist” pachage, using the Levenshtein distance to compare the similarities and differences between strings.

```
## Create the dictionary
## These 48 event types can be found on pages 2, 3, and 4 at this url: http://www.nws.noaa.gov/directive
## To view the original set of event types, use: levels(factor(data$EVTYPE)) .

dictionary <- c("Astronomical Low Tide","Avalanche","Blizzard","Coastal Flood","Cold/Wind Chill","Debris
```

First make the dictionary and the EVTYPE column both lowercase, and then add the replacement column to the data frame.

```
## Make both the dictionary and the EVTYPE column lowercase, to make the mapping more accurate.
dictionary <- tolower(dictionary)
data_c$EVTYPE <- tolower(data_c$EVTYPE)
data_c$EVTYPE_48 <- dictionary[amatch(data_c$EVTYPE,dictionary,method="lv", maxDist=20)]
```

I would however like to take a moment to talk about the validity of using this method to standardize the event types found in the column EVTYPE. There are numerous non-standard entries such as "?", "dust devil waterspout", "freezing drizzle", "gustnado and", "lack of snow", "no severe weather", "none", "summary of june 6", and even "excessive". Under the assumption that these represent a minority of cases, and that events like "no severe weather" and "none" did not lead to casualties or property damage, I decided that it was okay to let the function amatch() decide what to assign to each of these values. I will also not be looking at averages, so this is another reason that this method is sufficient. This would not be a good method to use if, for example, one wanted to know how much damage tornados caused on average, per occurence.

Next, since the full range of 48 event types were not included into the data base until 1996, we'll subset the data frame to include only years greater than or equal to 1996. In reality this step could have been done before running the amatch() function. It would have saved a little time, but the amatch() function excuted rather quickly even on the full data set.

```
## Creates a column of dates for ease of use.
data_c$BGN_DATE_Stan <- as.Date(data_c$BGN_DATE,format="%m/%d/%Y")

## Creates a new data frame.  The "d" added to the name signifies that it has been subsetted according t
data_cd <- data_c[data_c$BGN_DATE_Stan >= "1996-01-01",]
```

Further, the data will be subsetted to include only events that had either casualties or property damage. This is due to the fact that we will only be looking at totals rather than averages.

```
## Removes all rows that have a 0 in each of the 4 columns FATALITIES, INJURIES, PROPDMG, CROPDMG.
## In other words, if a weather event led to no casualites or economic damage, we will ignore it from th
data_cd0 <- data_cd[data_cd$FATALITIES != 0 | data_cd$INJURIES != 0 | data_cd$PROPDMG != 0 | data_cd$CRO
```

Before we begin to look at results, let's do a little sanity check and compare the 8 most common event types in the column EVTYPE to the 8 most common event types in EVTYPE_48. Remember that EVTYPE has many nonstandard and misspelled entries, while the latter only contains the standard 48 event types.

```
## These return the top 8 most common event types in the columns EVTYPE and EVTYPE_48.
head(sort(table(data_cd0$EVTYPE),decreasing=TRUE),n=8)
```

```
##
##         tstm wind thunderstorm wind              hail       flash flood
##             61776             43097             22679             19011
##           tornado         lightning             flood         high wind
##             12366             11152              9513              5402
```

```
head(sort(table(data_cd0$EVTYPE_48),decreasing=TRUE),n=8)
```

```
##
##         high wind thunderstorm wind              hail       flash flood
##             67244             43100             22683             19100
##           tornado         lightning             flood       strong wind
##             12366             11294              9744              3414
```

Pleasently, we see no immediately cause for concern with the way the function amatch() has cleaned up our EVTYPE column. "tstm wind" means "thunderstorm wind", and it is a little unfortunate that "tstm wind" was apparently mapped to "high wind", but the meanings are similar enough that we can accept this. The important thing is, we know that in the original data wind, hail, floods, and tornados were the most common event types, and they are as well in our new column of standard event types.

Another step of processing that will be done is to combine the data on fatalities and injuries into a total number of casualities. Instead of simply adding the two figures however, a weight of 10 will be given to each fatality. This is an arbitrary weight, though more reasonable than considering a fatality and an injury to be of equal importance. This will be added to a column called CASUALTIES.

```
## Adds a column called CASUALITIES to the data frame, which is a weighted combination of fatalties and
data_cd0$CASUALTIES <- 10*data_cd0$FATALITIES + data_cd0$INJURIES
```

The last step of processing the data is to find out what the total dollar amounts of property damage and crop damage are, as well as an overall total. This will require a bit of care because the amount of damage is found in one column as the base value, and in the adjacent column as an exponent. Luckily, at this point, the only values left in the exponents column are "K", "M", and "B", whereas before we had done some subsetting, there were many nonstandard values.

```
## To have a look at what type of exponents we are dealing with, look at: table(data_cd0$PROPDMGEXP) an
## It was seen at the time of this analysis that the only values were blanks, "K"'s, "M"'s, and "B"'s.
## The following are two for loops to find the total amount of property damage per event.
## ******* These for loops take about 3 minutes each, they should be optimized somehow ***************
for (i in 1:length(data_cd0$PROPDMG)) {
  if (data_cd0$PROPDMGEXP[i] == "K") {
    data_cd0$PROPDMG_TOTAL[i] <- data_cd0$PROPDMG[i]*10^3
  }
  else if (data_cd0$PROPDMGEXP[i] == "M") {
    data_cd0$PROPDMG_TOTAL[i] <- data_cd0$PROPDMG[i]*10^6
  }
  else if (data_cd0$PROPDMGEXP[i] == "B") {
    data_cd0$PROPDMG_TOTAL[i] <- data_cd0$PROPDMG[i]*10^9
  }
  else data_cd0$PROPDMG_TOTAL[i] <- 0
}

## For loop to find the total amount of crop damage per event.
for (i in 1:length(data_cd0$CROPDMG)) {
  if (data_cd0$CROPDMGEXP[i] == "K") {
    data_cd0$CROPDMG_TOTAL[i] <- data_cd0$CROPDMG[i]*10^3
  }
  else if (data_cd0$CROPDMGEXP[i] == "M") {
    data_cd0$CROPDMG_TOTAL[i] <- data_cd0$CROPDMG[i]*10^6
  }
  else if (data_cd0$CROPDMGEXP[i] == "B") {
    data_cd0$CROPDMG_TOTAL[i] <- data_cd0$CROPDMG[i]*10^9
  }
  else data_cd0$CROPDMG_TOTAL[i] <- 0
}

## Now lets find the total monetary damage, giving equal weight to property and crop damage.
data_cd0$TOTALDMG <- data_cd0$CROPDMG_TOTAL + data_cd0$PROPDMG_TOTAL
```

```
## One last step, let's create a data table from our data frame.
library(data.table)
datatable <- data.table(data_cd0)
```

Now that we have totals for the amount of casualties that occured in each event, as well as how much

economic damage occured in terms of property, crops, and in total, we can begin to pull some results from the data.

Results

Again, our primary goal here is find out which weather events caused the most damage in terms of casualties, as well as in terms of economic damage.

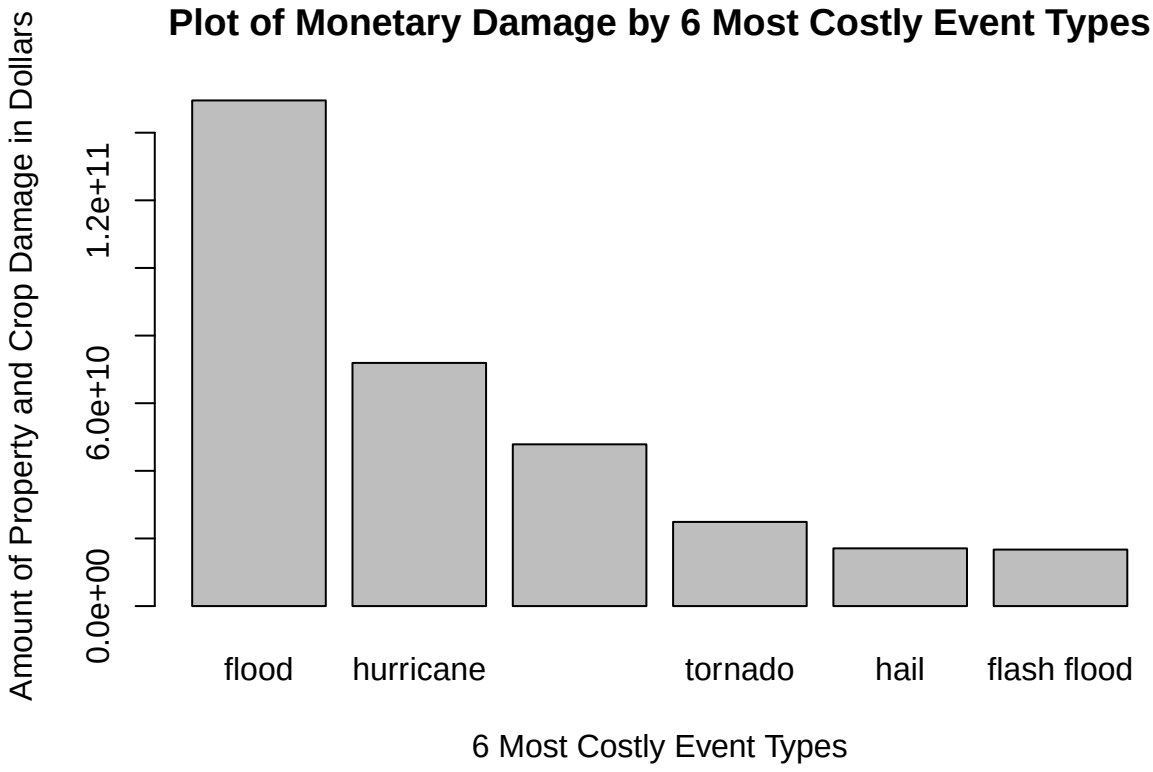
Let us start by totalling the amount of monetary damage according to each event type, and ordering it from greatest to least. Below are the top 10 most costly categories of event types. Floods caused $149,539,674,950, hurricanes caused $71,913,712,800, and storm tides caused $47,835,729,000 in total damage to property and crops.

```
total_by_EV <- datatable[,sum(TOTALDMG),by=EVTYPE_48]
head(total_by_EV[order(V1,decreasing=TRUE)],n=10)
```

```
##             EVTYPE_48          V1
##  1:             flood 149539674950
##  2: hurricane/typhoon  71913712800
##  3:        storm tide  47835729000
##  4:           tornado  24900370720
##  5:              hail  17072292870
##  6:       flash flood  16713502610
##  7:            seiche  14556434010
##  8:           drought  14415436600
##  9:         high wind  10923966450
## 10:          wildfire   8508309630
```

Here we have a bar plot of the top 5 most costly categories, in order to get a visual idea of how the compare to each other in relative terms. Notice that not only does "flood" appear first, but it appears again as "flash flood" at the sixth position on the plot. Floods in all forms are almost surely the most costly form of event type as a whole.

```
plot <- head(total_by_EV[order(V1,decreasing=TRUE)],n=6)
## On the plot, I manually input the event types labels, since "hurricane/typhoon" was too long of a lab
barplot(plot$V1, xlab="6 Most Costly Event Types",names.arg=c("flood","hurricane","storm tide","tornado
```



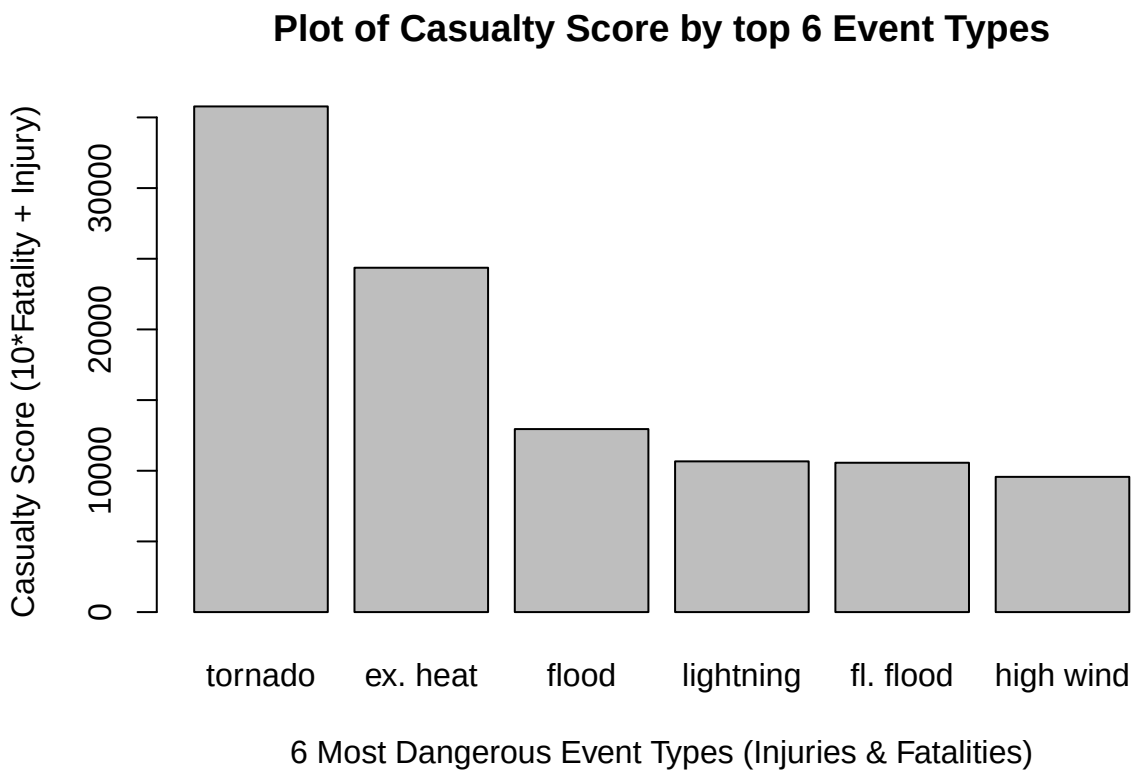
Now we do look at casualties in a similar manner. Remember, the number found in the CASUALTIES column is a score, rather than a number that represents directly the amount of people that were hurt. It is a score calculated by giving 1 point for each injury, and 10 points for each fatality. Here is a ranking of the top 10 most dangerous events in terms of their scores:

```
total2_by_EV <- datatable[,sum(CASUALTIES),by=EVTYPE_48]
head(total2_by_EV[order(V1,decreasing=TRUE)],n=10)
```

```
##          EVTYPE_48    V1
##  1:        tornado 35777
##  2: excessive heat 24363
##  3:          flood 12948
##  4:      lightning 10663
##  5:    flash flood 10565
##  6:      high wind  9567
##  7:    rip current  5923
##  8:           heat  3686
##  9:   winter storm  3289
## 10:       wildfire  2796
```

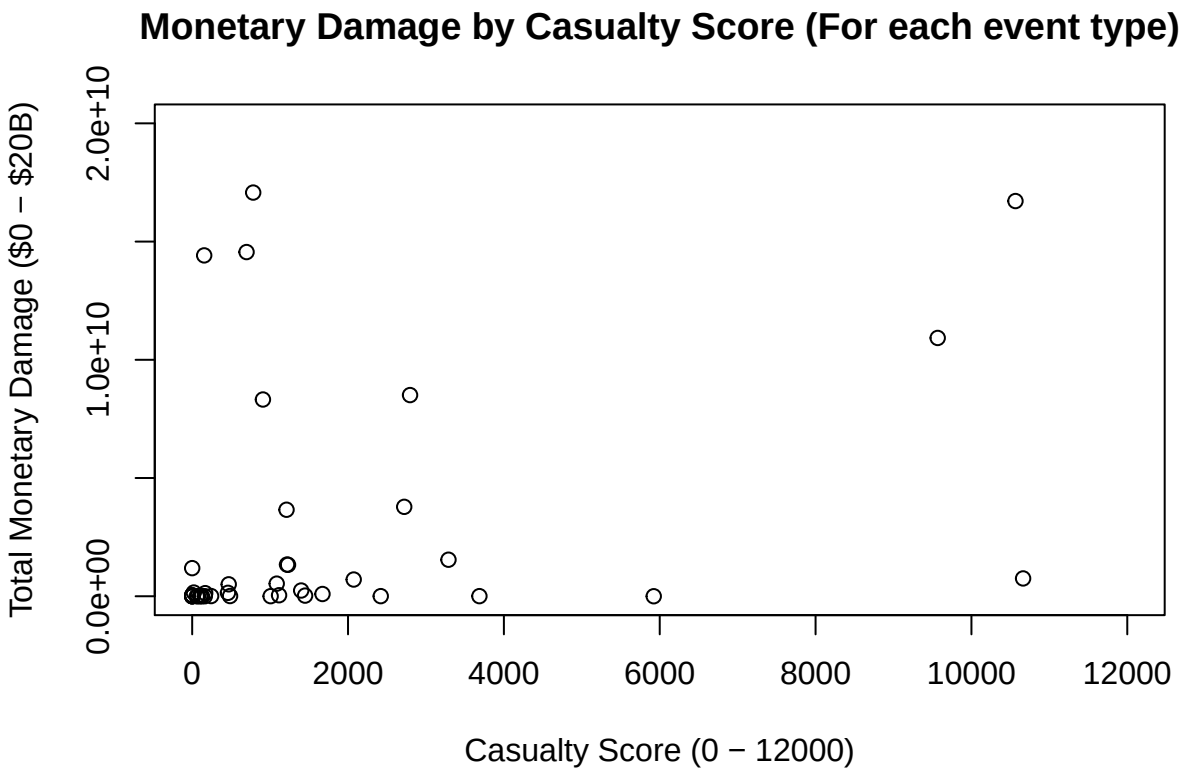
We see that "tornado" is at the top of the list. This is likely do to the fact that they arrive with little to no warning, while in the case of hurricanes, there is generally a lot of time to evacuate. Though as we saw, hurricanes lead to a lot of property damage, more than tornados. Now lets see a bar plot of these results:

```
plot2 <- head(total2_by_EV[order(V1,decreasing=TRUE)],n=6)
## On the plot, I manually input the event types labels, since some labels were too long to appear.
barplot(plot2$V1, xlab="6 Most Dangerous Event Types (Injuries & Fatalities)",names.arg=c("tornado","ex
```



One last interesting thing to look at is what sort of correlation there is between property/crop damage and damage to human life. It would be reasonable to expect at least a weak positive correlation, since severe events do not discrimate in the damage they inflict. Though as noted earlier, some events occur without warning, while some allow time to evacuate.

```
plot(total2_by_EV$V1,total_by_EV$V1,xlab="Casualty Score (0 - 12000)",ylab="Total Monetary Damage ($0 -
```



Within the plot, since there are some extreme outliers in the data and I wanted to focus in on how the majority of the data was behaving, the x-axis and y-axis were adjusted to give a better view, and ignore the few outlying values. As can be seen, there is a slight positive correlation though it is indeed weak. This is likely due to the fact that even in this window, many event types are still being included that were not significant in terms of damage or casualties. One new question that presents itself is what sort of events lead to high economic damage with low casualties, and which lead to high casualties with low economic damage. As can be seen in the plot, there are a few events that hug the x-axis and y-axis quite tightly, even as their casualty scores or monetary damage numbers increase, respectively.

Appendix

Session Information

```
## Returns information about the computing environment.
sessionInfo()
```

```
## R version 3.2.3 (2015-12-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.4 (Yosemite)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
```

```
## [1] data.table_1.9.6   stringdist_0.9.4.1
##
## loaded via a namespace (and not attached):
##  [1] magrittr_1.5     formatR_1.2.1    parallel_3.2.3   tools_3.2.3
##  [5] htmltools_0.3    yaml_2.1.13      codetools_0.2-14 stringi_1.0-1
##  [9] rmarkdown_0.9.2  knitr_1.12       stringr_1.0.0    digest_0.6.9
## [13] chron_2.3-47     evaluate_0.8
```

Data dictionary

This is to provide some idea of the purpose of each of the 37 columns, along with url links for further information.

https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf http://www.ncdc.noaa.gov/stormevents/details.jsp http://www.ncdc.noaa.gov/stormevents/pd01016005curr.pdf

1. "STATE___" State number (1 = Alabama)
2. "BGN_DATE" Begin date
3. "BGN_TIME" Begin time
4. "TIME_ZONE" Time zone
5. "COUNTY" County number
6. "COUNTYNAME" County name
7. "STATE" State, 2 letter abbreviation
8. "EVTYPE" Type of event (storm, flood, etc)
9. "BGN_RANGE" Beginning range
10. "BGN_AZI" Beginning azimuth
11. "BGN_LOCATI" Beginning location
12. "END_DATE" End date
13. "END_TIME" End time
14. "COUNTY_END" County where event ended? (name or number?)
15. "COUNTYENDN" County where event ended? (name or number?)
16. "END_RANGE" Ending range
17. "END_AZI" Ending azimuth
18. "END_LOCATI" Ending location
19. "LENGTH" Length of tornado path (in yards?)
20. "WIDTH" Maximum width of tornado's path in yards
21. "F" Fujita tornado intensity scale
22. "MAG" Hail in inches (implied hundreths)
23. "FATALITIES" Number of fatalities
24. "INJURIES" Number of injuries
25. "PROPDMG" Property damage in dollars
26. "PROPDMGEXP" K=thousands, M=millions, B=billions
27. "CROPDMG" Crop damage in dollars
28. "CROPDMGEXP" H=hundreds, K=thousands, M=millions, B=billions
29. "WFO" Weather Forecast Office
30. "STATEOFFIC" State office?
31. "ZONENAMES" Zone names?
32. "LATITUDE" Latitude
33. "LONGITUDE" Longitude
34. "LATITUDE_E" ?

35. “LONGITUDE_” ?
36. “REMARKS” Remarks
37. “REFNUM” Reference number?