Reproducible Research: Final Project

Alejandro Osorio

September 2018

Synopsis

Based on the proposed methodology (Appendix 1: Methodology), the obtained datasets (Appendix 2: Data Processing) and the preliminary visualization-based analysis (Appendix 3: Results), the following suggestions can be made with regards to the following questions:

1. Across the United States, which types of events (as indicated in the 'EVTYPE' variable) are most harmful with respect to population health?
2. Across the United States, which types of events have the greatest economic consequences?

1. Most harmful with respect to population health

a. Most harmful standalone events

'Tsunamies' have the highest average fatalities (6.5) and injuries (1.7), by event type, followed by 'heat', with 1.3 and 3 respectively (more results in Appendix 3.1).

b. Most overall harmful events

'Tornados' have the highest total acumulated fatalities (almost 6,000) and total acumulated injuries (almost 10,000), followed by 'excessive heat', with almost 2,000 total acumulated fatalities and 600 total acumulated injuries (more results in Appendix 3.1).

2. Greatest economic consequences

a. Most expensive standalone events

'Huricane typhoon' has the highest average 'total damage' per event, costing an average of around 320,000,000 USD per event, followed by 'storm surge', costing an average of around 120,000,000 USD per event.

b. Most overall expensive events

The most overall expensive event was of the type 'flood', with an accumulated 'total damage' for the whole recorded period of around 150,000,000,000 USD, and followed by 'hurricane typhoon' with an accumulated 'total damage' of around 85,000,000,000.

Appendix 1: Methodology

1. Measuring harm to population health

Harm to population health was quantified using the following variables (detailed calculations in Appendix 2.e):

1. Average injuries, by event type.
2. Average fatalities, by event type.
3. Total number of recorded events, by event type.

4. Total acumulated injuries, by total number of recorded events, by event type.
5. Total acumulated fatalities, by total number of recorded events, by event type.

Most harmful events were identified considering the following two criteria:

1. Most harmful standalone events: those with the highest averages of injuries and fatalities.
2. Most overall harmful events: those with the highest acumulated injuries and fatalities, for the whole recorded period.

2. Measuring economic damage

Economic damage was quantified using the following variables (detailed calculations in Appendix 2.e):

1. Average 'property damage' (in USD), by event type.
2. Average 'crop damage' (in USD), by event type.
3. Average 'total damage' (in USD, as a sum of previous 1 and 2 variables).
4. Total number of recorded events, by event type.
5. Total acumulated 'total damage' (in USD), by total number of recorded events, by event type.

Most economically damaging events were identified considering the following two criteria:

1. Most expensive standalone events: those with the highest averages of 'total damage'.
2. Most overall expensive events: those with the highest acumulated 'total damage', for the whole recorded period.

Appendix 2: Data Processing

1. Reading Data

```
dataoriginal <- read_csv("repdata%2Fdata%2FStormData.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   STATE__ = col_double(),
##   COUNTY = col_double(),
##   BGN_RANGE = col_double(),
##   COUNTY_END = col_double(),
##   END_RANGE = col_double(),
##   LENGTH = col_double(),
##   WIDTH = col_double(),
##   F = col_integer(),
##   MAG = col_double(),
##   FATALITIES = col_double(),
##   INJURIES = col_double(),
##   PROPDMG = col_double(),
##   CROPDMG = col_double(),
##   LATITUDE = col_double(),
##   LONGITUDE = col_double(),
##   LATITUDE_E = col_double(),
##   LONGITUDE_ = col_double(),
##   REFNUM = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

2. Preparing Dataset

a. Subsetting and identifying variables without NAs

The subset that was passed on for further cleaning (variables defined in Apendix 1), was:

```
datafinal <- select(dataoriginal, c(2,5,7,8, 23:28))
```

Variables without NAs:

```
which(colSums(is.na(datafinal)) == 0)
```

```
##   BGN_DATE     COUNTY      STATE     EVTYPE FATALITIES   INJURIES
##          1          2          3          4          5          6
##    PROPDMG    CROPDMG
##          7          9
```

Fortunately, almost all variables required by the methodology (besides 'PROPDMGEXP' and 'CROPDMG-EXP') were complete.

b. 'PROPDMGEXP' and 'CROPDMGEXP' cleaning

First cleaning

These two variables contained NAs. Now, if NAs in these variables were asociated with a zero value on their corresponding 'PROPDMG' and 'CROPDMG' variables, they were considered complete (zero times any magnitude is zero). Therefore:

```
incompletedmg <- with(datafinal, which((is.na(PROPDMGEXP) & PROPDMG != 0) | (is.na(CROPDMGEXP) & CROPDM
incompletedmg
```

```
##  [1] 192467 196196 196687 196961 199598 199706 199850 200018 200270 202691
## [11] 202709 202710 204733 205444 205613 206515 209415 211429 212172 218470
## [21] 218919 221857 221940 222022 222148 222150 222151 222182 222200 222214
## [31] 222388 222396 222447 222479 222486 222510 222572 222576 222581 222595
## [41] 222651 222709 222773 222775 222817 222882 222911 222946 222972 223089
## [51] 223209 223214 223227 223237 223435 223447 223515 223531 223571 223590
## [61] 223639 223642 223683 223689 223690 223735 223738 223907 223999 228295
## [71] 228472 232662 235049 238757 239654 240397 242967 244614 246030
```

Only 79 rows had either a 'PROPDMG' or 'CROPDMG' variable with a non zero value and an NA magnitude. These rows were therefore eliminated from the dataset, as follows:

```
datafinal <- datafinal[-incompletedmg,]
```

Final check:

```
with(datafinal, which((is.na(PROPDMGEXP) & PROPDMG != 0)|is.na(CROPDMGEXP) & CROPDMG != 0))
```

```
## integer(0)
```

So no values with NA magnitudes were left.

Further cleaning

When preparing to create unique numeric variables for property and crop damage (ie: one '1.000' variable instead of '1' and 'K' variables), the following situation was identified for the magnitudes' values:

PROPDMGEXP:

```
table(datafinal$PROPDMGEXP, useNA = "always")
```

```
##
##      -      ?      +      0      1      2      3      4      5      6
##      1      8      5    216     25     13      4      4     28      4
##      7      8      B      h      H      K      m      M   <NA>
##      5      1     40      1      6 424663      7  11329 465858
```

CROPDMGEXP:

```
table(datafinal$CROPDMGEXP, useNA = "always")
```

```
##
##      ?      0      2      B      k      K      m      M   <NA>
##      6     19      1      9     21 281827      1   1994 618340
```

So further cleaning was required in order to end up with only "K", "M" or "B" values.

First, lowercase "k" and "m" values were replaced by uppercase values in both variables, as follows:

```
datafinal$PROPDMGEXP[which(datafinal$PROPDMGEXP == "m")] <- "M"
datafinal$CROPDMGEXP[which(datafinal$CROPDMGEXP == "k")] <- "K"
datafinal$CROPDMGEXP[which(datafinal$CROPDMGEXP == "m")] <- "M"
```

Finally, rows in which either 'PROPDMGEXP' or 'CROPDMGEXP' had values other than "K", "M", "B" or NA, were eliminated, as follows:

```
wrongdmgexp <- with(datafinal, which(!(PROPDMGEXP %in% c("K", "M", "B", NA)) | !(CROPDMGEXP %in% c("K",
datafinal <- datafinal[-wrongdmgexp,]
length(wrongdmgexp)
```

```
## [1] 347
```

With only the above number of eliminated rows.

c. 'EVTYPE' cleaning

Preliminary analysis

Dimensioning number of unique values for this variable:

```
glimpse(unique(datafinal$EVTYPE))
```

```
##  chr [1:973] "TORNADO" "TSTM WIND" "HAIL" "FREEZING RAIN" "SNOW" ...
```

Given the humongous diversity of values (973), a table view was applied in order to get a glipmse of the situation, as follows:

```
head(data.frame(table(datafinal$EVTYPE)), 15)
```

```
##                          Var1 Freq
## 1                           ?    1
## 2              ABNORMAL WARMTH    4
## 3               ABNORMALLY DRY    2
## 4               ABNORMALLY WET    1
## 5         ACCUMULATED SNOWFALL    4
## 6          AGRICULTURAL FREEZE    6
## 7                APACHE COUNTY    1
## 8       ASTRONOMICAL HIGH TIDE  103
```

```
## 9        ASTRONOMICAL LOW TIDE  174
## 10                    AVALANCE    1
## 11                   AVALANCHE  386
## 12                BEACH EROSIN    1
## 13               Beach Erosion    1
## 14               BEACH EROSION    3
## 15 BEACH EROSION/COASTAL FLOOD    1
```

As suspected, just out of the first 15 results it could be concluded that no standardized values were contained within that variable (for example, 'beach erosion' was written three different ways, including one misspelled version).

In order to simplify the unavoidable required value-standartization, unique values with a frecuency equal or less than 10 were identified and arranged as follows:

```
lowfreqevtype <- datafinal$EVTYPE %>%
        table() %>%
        data.frame() %>%
        filter(., Freq <= 10) %>%
        arrange(., desc(Freq)) %>%
        glimpse()
```

```
## Observations: 814
## Variables: 2
## $ .    <fct> Cold, DENSE SMOKE, Gusty Winds, MIXED PRECIP, SNOW/SLEET,...
## $ Freq <int> 10, 10, 10, 10, 10, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, ...
```

So 814 low frequency value types out of 973, reduced the total number down to 159, involving the following number of rows:

```
sum(lowfreqevtype$Freq)
```

```
## [1] 1738
```

Data cleaning

Eliminating said rows:

```
rowslowevtype <- which(datafinal$EVTYPE %in% lowfreqevtype$.)
datafinal <- datafinal[-rowslowevtype,]
```

At this point, in order to execute a quick standardization with a more suited app, the remaining unique 'EVTYPE' values were inputed into a dataset that was exported as csv file, as follows:

```
evtypevalues <- datafinal$EVTYPE %>%
        unique() %>%
        data.frame() %>%
        arrange(., desc(.))
write_csv(evtypevalues, "evtypevalues.csv")
```

Said file was added a variable with standardized names, which was imported as follows:

```
evtypestd <- read_csv2("evtypestd.csv")
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Parsed with column specification:
## cols(
##   EVTYPE = col_character(),
```

```
##   EVTYPESTD = col_character()
## )
```

Next, an extra variable was created in 'datafinal', with standardized 'EVTYPE' values, as follows:

```
datafinal <- left_join(datafinal, evtypestd, by = "EVTYPE")
```

Finally, checking for NAs:

```
sum(is.na(datafinal$EVTYPESTD))
```

```
## [1] 0
```

And a final glimpse to the obtained unique values:

```
glimpse(unique(datafinal$EVTYPESTD))
```

```
##  chr [1:69] "TORNADO" "THUNDERSTORM WIND" "HAIL" "FREEZING RAIN" ...
```

d. Creating variables with numeric values for property and crop damage

In order to quantify costs, as mentioned in Appendix 1, unique numeric variables ('PROPDMGVAL' and 'CROPDMGVAL') for property and crop damage (ie: one '1.000' variable instead of '1' and 'K' variables), were created as follows:

```
datafinal <- mutate(datafinal, PROPDMGVAL = if_else(PROPDMGEXP == "K", PROPDMG * 1000, if_else(PROPDMGE
```

Finally, variable 'TOTALDMGVAL' which added both 'PRPODMGVAL' and 'CROPDMGVAL' values, was created as follows:

```
datafinal <- mutate(datafinal, TOTALDMGVAL = PROPDMGVAL + CROPDMGVAL)
```

e. Creating final datasets for harmful and economically expensive events

Harmful events' dataset

As said in methodology (Appendix 1), the required final dataset to identify the most harmful events, was built as follows:

```
dataharm <- datafinal %>%
        group_by(EVTYPESTD) %>%
        summarise(avgfatal = mean(FATALITIES), avginjury = mean(INJURIES), eventnum = length(EVTYPESTD)
        gather('avgfatal', 'avginjury', 'eventnum', 'totalfatal', 'totalinjury', key = "measure", value
```

Economic damage's dataset

As said in methodology (Appendix 1), the required final dataset to identify the most economically expensive events, was built as follows:

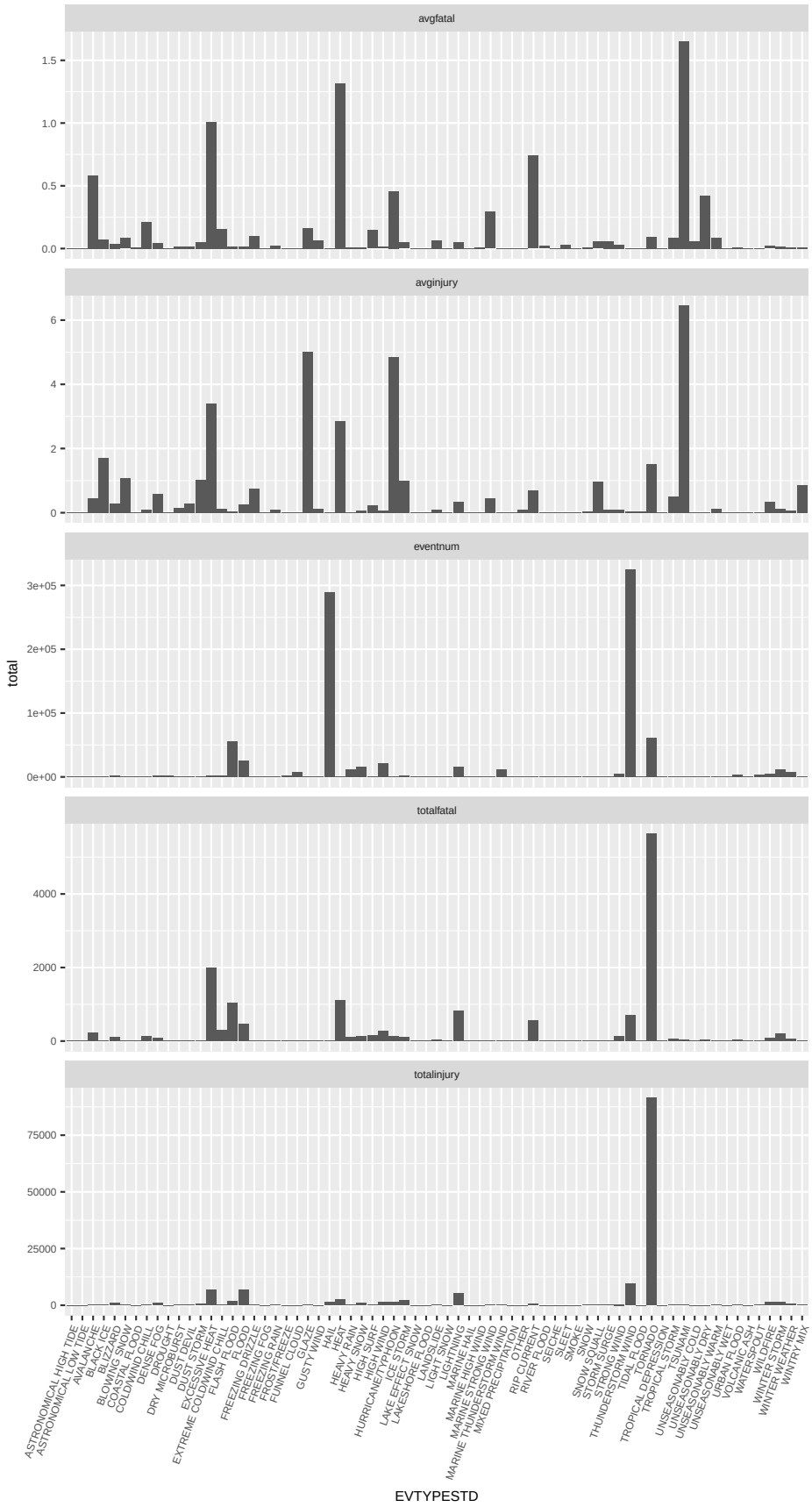
```
dataecon <- datafinal %>%
        group_by(EVTYPESTD) %>%
        summarise(avgprop = mean(PROPDMGVAL), avgcrop = mean(CROPDMGVAL), avgtotal = mean(TOTALDMGVAL),
        gather('avgprop', 'avgcrop', 'avgtotal', 'eventnum', 'totalecon', key = "measure", value = "tota
```

Appendix 3: Results

1. Most Harmful Events

The following plots show the behaviour of each of the five variables (y axis) required by the methodology (Appendix 1), for each type of event (x axis):

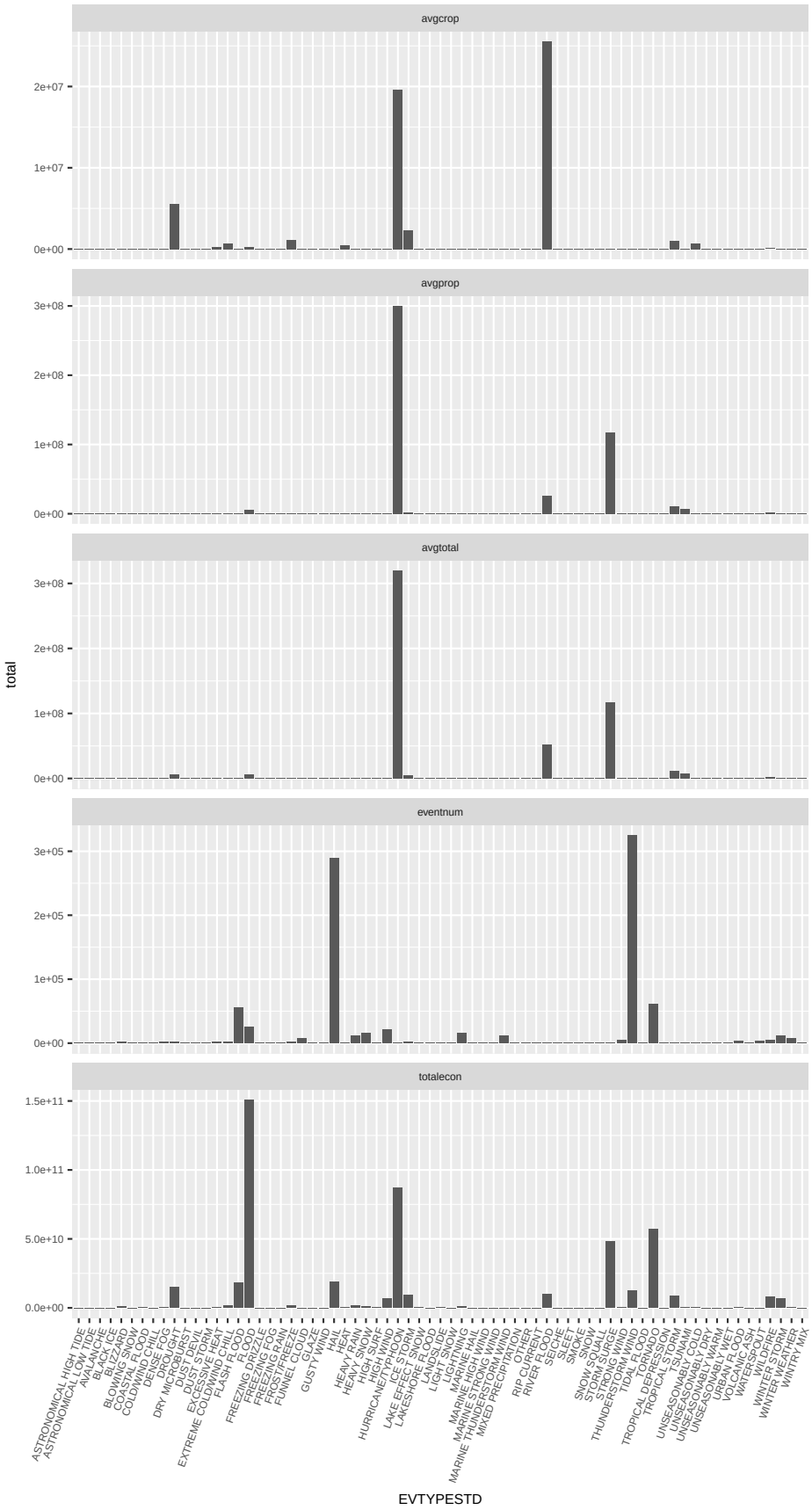
```
ggplot (data = dataharm, mapping = aes(x = EVTYPESTD, y = total)) +
        geom_col() +
        facet_wrap(~ measure, nrow = 5, scales = "free_y") +
        theme(text = element_text(size=8), axis.text.x = element_text(angle = 70, vjust = 1, hjust=1))
```


avgfatal
avginjury
eventnum
totalfatal
totalinjury
total
EVTYPESTD

2. Events with Greatest Economic Damage

The following plots show the behaviour of each of the five variables (y axis) required by the methodology (Appendix 1), for each type of event (x axis):

```
ggplot (data = dataecon, mapping = aes(x = EVTYPESTD, y = total)) +
        geom_col() +
        facet_wrap(~ measure, nrow = 5, scales = "free_y") +
        theme(text = element_text(size=8), axis.text.x = element_text(angle = 70, vjust = 1, hjust=1))
```


avgcrop
avgprop
avgtotal
eventnum
totalecon
total
EVTYPESTD
ASTRONOMICAL HIGH TIDE
ASTRONOMICAL LOW TIDE
AVALANCHE
BLACK ICE
BLIZZARD
BLOWING SNOW
COASTAL FLOOD
COLD/WIND CHILL
DENSE FOG
DROUGHT
DRY MICROBURST
DUST DEVIL
DUST STORM
EXCESSIVE HEAT
EXTREME COLD/WIND CHILL
FLASH FLOOD
FLOOD
FREEZING DRIZZLE
FREEZING FOG
FROST/FREEZE
FUNNEL CLOUD
GUSTY WIND
HAIL
HEAT
HEAVY RAIN
HEAVY SNOW
HIGH SURF
HIGH WIND
HURRICANE/TYPHOON
LAKE-EFFECT SNOW
LAKESHORE FLOOD
LANDSLIDE
LIGHT SNOW
LIGHTNING
MARINE HAIL
MARINE HIGH WIND
MARINE STRONG WIND
MARINE THUNDERSTORM WIND
MIXED PRECIPITATION
OTHER
RIP CURRENT
RIVER FLOOD
SEICHE
SLEET
SMOKE
SNOW
SNOW SQUALL
STORM SURGE
STRONG WIND
THUNDERSTORM WIND
TIDAL FLOOD
TORNADO
TROPICAL DEPRESSION
TROPICAL STORM
TSUNAMI
UNSEASONABLY COLD
UNSEASONABLY DRY
UNSEASONABLY WARM
URBAN FLOOD
VOLCANIC ASH
WATERSPOUT
WILDFIRE
WINTER STORM
WINTER WEATHER
WINTRY MIX