Some Health and Economic Impacts of Historic Weather Events

PA2 Assignment
Reproducible Research #2
(Second course project for Coursera Reproducible Research)
December, 2014

Synopsis

The U. S. National Weather Service (NWS) maintains a weather event database that is available to the public through the internet. For this project, data collected in the U.S. and territories, from 1951 to 2007, is used to analyze some impacts of weather events on public health and on the economy. Data is read in from the provided files, then cleaned, tidied, and formatted prior to calculations. Two assignment questions are addressed through data analysis, and separate conclusions are provided for each.

Data Processing: Obtaining and Reading in Data

Data is first conditionally (if not present) downloaded from the internet source

```
library(dplyr, warn.conflicts = FALSE)
library(knitr)
library(stringr)
library(ggplot2)
library(grid)
library(scales)

targFileName <-                              ## Target-download compressed file
  "data/repdata\ data\ StormData.csv.bz2"
if(length(list.dirs("./figures")) == 0){     ## Create a target directory for any
  dir.create("./figures")                    ## plot files that may be produced.
}
downLoadData <- function(){                  ## Function downloads data if needed
  if(length(list.dirs("./data")) == 0){      ## Create data directory if it does
    dir.create("./data")                     ## not exist.
  }
  dataURL <- paste0(                         ## Name of data file
    "https://d396qusza40orc.cloudfront.",
    "net/repdata%2Fdata%2FStormData.csv.bz2")
  if(!file.exists(targFileName)){            ## If not present, download file
    download.file(dataURL,                   ## from web and decompress
```

```
                  destfile = targFileName,
                  method = "curl")
  }
}

# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot
# objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
# Credit: this function obtained from R Cookbook example at
# http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_%28ggplot2%29/
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  require(grid, quietly=TRUE)
  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)
  numPlots = length(plots)
  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }
  if (numPlots==1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
```

```
    }
  }
}
```

Once present on the local system, the file is read into an R data structure "stormDat".

```
loadStormDat <- function(){                 ## Function reads csv file.
  downLoadData()                            ## Call to download data function.
  if(!"stormDat" %in% ls()){                ## If not already loaded, read-in
    stormDat <- read.csv(targFileName)      ## csv data file directly from bzip
  return(stormDat)                          ## archive.
  }
}
stormDat <- loadStormDat()
```

The data set, read into R, is composed of almost 1 million entries with 37 variables, as shown:

```
dim(stormDat)                               ## Show dimensions of the data
```

```
[1] 902297     37
```

Data variable names do not all clearly describe the data contained within the variable, nor do they seem to follow any standard convention in composition:

```
colnames(stormDat)                          ## Show the variable names of 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 addition, some of the data seem to be formatted and composed in a manner not conducive to easy use or analysis.

```
head(stormDat, 3)                           ## First 3 rows of data
```

```
  STATE__           BGN_DATE BGN_TIME TIME_ZONE COUNTY COUNTYNAME STATE
1       1 4/18/1950 0:00:00     0130       CST     97     MOBILE    AL
2       1 4/18/1950 0:00:00     0145       CST      3    BALDWIN    AL
3       1 2/20/1951 0:00:00     1600       CST     57    FAYETTE    AL
   EVTYPE BGN_RANGE BGN_AZI BGN_LOCATI END_DATE END_TIME COUNTY_END
1 TORNADO         0                                               0
2 TORNADO         0                                               0
3 TORNADO         0                                               0
  COUNTYENDN END_RANGE END_AZI END_LOCATI LENGTH WIDTH F MAG FATALITIES
1         NA         0                      14.0   100 3   0          0
2         NA         0                       2.0   150 2   0          0
3         NA         0                       0.1   123 2   0          0
  INJURIES PROPDMG PROPDMGEXP CROPDMG CROPDMGEXP WFO STATEOFFIC ZONENAMES
1       15    25.0          K       0
2        0     2.5          K       0
3        2    25.0          K       0
  LATITUDE LONGITUDE LATITUDE_E LONGITUDE_ REMARKS REFNUM
1     3040      8812       3051       8806              1
2     3042      8755          0          0              2
3     3340      8742          0          0              3
```

```
range(stormDat[, 37])                       ## Max, min of range variable
```

```
[1]      1 902297
```

Initial Cleaning and Formatting Data We define a set of functions to clean up some of the data, including conversion of factors (mostly into text format) replacing some NA elements, formatting date and time entries consistently, and properly formatting latitude and longitude. The last variable, the reference number, is obviously an index for the entire table, and is discarded. See the NWS data format description and the additional data cleaning remarks, below.

```
chainGsub <-function(dat, pat, repl){       ## Subfunction to chain string
  return(gsub(pat, repl, dat))              ## replacements for colnames
}

insertMark <- function(hourMinStr){         ## Subfunction to change time stamp
  return(paste0(strtrim(                    ## data to text, and insert ":".
    hourMinStr, 2), ":",
    substr(hourMinStr, 3, 4)))
}

cleanStormDat <- function(stormDat){        ## Function makes data "tidier."
  stormDat <- stormDat[, 1:36]              ## Discard record number column
```

```
colnames(stormDat) <-                       ## Rename the column headings to
  colnames(stormDat) %>%                    ## lower case using chaining to
  tolower() %>%                             ## modify the strings to "tidier"
  chainGsub("_locati" , "location") %>%     ## form.
  chainGsub("tude_e" , "tudeatend") %>%
  chainGsub("tude_" , "tudeatend") %>%
  chainGsub("_azi" , "heading") %>%
  chainGsub("bgn" , "begin") %>%
  chainGsub("evtype" , "eventtype") %>%
  chainGsub("endn" , "endname") %>%
  chainGsub("state__" , "statenumcode") %>%
  chainGsub("stateoffic" , "stateweatheroffice") %>%
  chainGsub("mag" , "magnitude")  %>%
  chainGsub("dmg" , "damage") %>%
  chainGsub("propd" , "propertyd") %>%
  chainGsub("wfo" , "weatherforecastoffice") %>%
  chainGsub("_" , "")
colnames(stormDat)[21] <- "fujitascale"     ## Name this column name directly.
stormDat$begindate <-                       ## Change begin and end date types
  gsub(" 0:00:00" , "",                     ## to plain character and remove
       as.character(stormDat$begindate))    ## superfluous time stamps from
stormDat$enddate <-                         ## date fields.
  gsub(" 0:00:00" , "", as.character(stormDat$enddate))
stormDat$begindate <-                       ## Change date fields to date type
  as.Date(stormDat$begindate,
            format = "%m/%d/%Y")
stormDat$enddate <-
  as.Date(stormDat$enddate, format = "%m/%d/%Y")
stormDat$begintime <- as.character(stormDat$begintime)
stormDat$endtime <- as.character(stormDat$endtime)
stormDat[str_length(                        ## Add colon to event time stamps
           stormDat$begintime) == 4,        ## with entry as a 4-digit integer.
         3] <- insertMark(
                stormDat[str_length(
                           stormDat$begintime) == 4,
                         3])
stormDat[str_length(stormDat$endtime) == 4, 13] <-
  insertMark(stormDat[str_length(
                        stormDat$endtime) == 4,
                      13])
                                            ## Correcting State codes
stormDat[stormDat$state == "LO", 7] <-      ## New York Lake Ontario events
  "NY"
stormDat[stormDat$state == "PM", 7] <-      ## Guam coastal area events
  "GU"
stormDat[stormDat$state == "PK", 7] <-      ## Pacific + Alaskan coastal areas
```

```
  "AK"
stormDat[stormDat$state == "PZ", 7] <-    ## Indicates pacific coast. All
  "CA"                                    ## references are to coastal CA
stormDat[stormDat$state == "SL", 7] <-    ## New York St. Lawrence Seaway
  "NY"                                    ## events
stormDat[stormDat$state == "ST", 7] <-    ## Correction for Ohio typo
  "OH"
stormDat[stormDat$state == "XX", 7] <-    ## New York coastal events
  "NY"
stormDat$state <-                         ## Recast state abbreviation back to
  as.factor(stormDat$state)               ## factor.
stormDat$timezone <-                      ## Change time zone abbreviation to
  toupper(                                ## upper case, and temporarily to
    as.character(stormDat$timezone))      ## plain text format.
stormDat[stormDat$state == "AK" &         ## Correction of Alaska Daylight
         stormDat$timezone == "ADT",      ## Savings time code to modern
         4] <- "AKDT"                     ## standard code.
stormDat[stormDat$state == "AK" &         ## Correction of Alaskan Standard
         stormDat$timezone %in%           ## time codes to modern standard
         c("AST" , "AKS"), 4] <- "AKST"   ## code.
stormDat$timezone <-                      ## Recast time zone as a factor.
  as.factor(stormDat$timezone)
toCent <- function(num){return(num*.01)}
for(colNum in 32:35){                     ## Fix lat/long in four columns from
  stormDat[,colNum] <-                    ## integer to real number with two
    sapply(stormDat[,colNum], toCent)     ## decimal places
}
stormDat[stormDat$state ==  "AS" &        ## Fix American Samoa Latitude
         is.na(stormDat$latitude),        ## values that are NA to appropriate
         32] <- -14.3                     ## lat for approx middle of island
stormDat[stormDat$state  ==  "AS" &
         is.na(stormDat$latitudeatend),
         34] <- -14.3
stormDat[stormDat$state  ==  "AS",        ## Fix American Samoa Longitude to
         33] <-                           ## negative numbers.
  (stormDat[stormDat$state  ==  "AS",
            33] * -1)
stormDat[stormDat$state  ==  "AS",
        35] <-
  (stormDat[stormDat$state  ==  "AS",
            35] * -1)
stormDat[stormDat$state  ==  "GU" &       ## Fix Guam longitude reported with
         stormDat$longitude > 0,          ## wrong sign.
         33] <-
  (stormDat[stormDat$state  ==  "GU" &
            stormDat$longitude > 0,
```

```
              33] * -1)
  stormDat[stormDat$state  ==  "GU" &
           stormDat$longitudeatend > 0,
           35] <-
    (stormDat[stormDat$state  ==  "GU" &
              stormDat$longitudeatend > 0,
              35] * -1)
  stormDat[524416, 7] <- "MH"              ## Fix single Marshall Islands event
  stormDat$remarks <- as.character(        ## reported by Guam regional weather
    stormDat$remarks)                      ## station
  return(stormDat)                         ## Change freeform text from factor
}                                          ## to plain text.
stormDat <- cleanStormDat(stormDat)        ## Call to clean data function
```

Data cleanup/tidy data notes:
- *State codes sometimes include regional office abbreviations that often affect multiple states, and should be treated specially in state counts.*
– *Code "AN" represents "Atlantic North," data with this label represent multiple states, including at least "DE" , "NJ" , "NY" , "VA" , "MD" , "NC".*
– *State code "LE" is used for Lake Erie multistate events, including "OH", "NY" , "MI" , "PA".*
– *State code "GM" is used for mulitstate, oceanic events occuring in states around the Gulf of Mexico.*
– *Code "LS"" indicates events occurring around Lake Superior, usually affecting both "MI" and "WI".* – *Code "LH" represents Lake Huron, and affects the same states as "LS".*
- *Record collection began before Alaska and Hawaii statehood, and time zone standard codes have changed multiple times since the beginning of record keeping. Several typographical errors are noted in the original data set time zone and state abbreviations.*

From the NWS supplemental information page, we are informed that database information was originally entered on paper forms, but since 1993 have been entered into a computer database; either directly or ingested from older (paper) records.

The data description from the NWS data source is the service operating instruction dated August 17, 2007. The reader should note that the first several entries in the table, shown above, are from the 1950s. Section 7 of the instruction lists 48 standardized event types, yet the data appear to contain 985 distinct category levels as a factor. In addition to the 45 standard event types (with subtypes and definitions), there are apparently mulitple entries combining two or more of the standard types (in varying order), freeform entries that do not match any of the standard types or subtypes, summary entries for which mulitple types are entered in the remarks variable, type entries containing magnitude information intended for entry to other variables, arbitrary punctuation, and mis-spellings.

NWS further defines database content and history at the database details web page. Tornado data were initially entered in 1950 through 1954, and from 1954 to 1992 thunderstorm, hail, and tornado data were also recorded. Only since 1996 have 48 different event types been recorded. We assume the data will be skewed toward tornado and thunderstorm events, as these have been the primary observation to be recorded through the database lifecycle.

In spite of the effort invested in cleaning the variable names, state code, time zone, and geo reference data, above, these data are still not in a form to support confidence in any analyses with a high degree of precision: they have been collected over a very long period of time using widely varying methods and standards. Per the data source, the NWS FAQ :

> "Therefore, when using information from Storm Data, customers should be cautious as the NWS does not guarantee the accuracy or validity of the information."

The reader should consider that these data, without a great deal more effort to screen, clean, and "tidy" the information, are useful only to the extent of providing estimates, and not precise results.

Assignment Questions:

Question 1 Analysis

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

In our preliminary cleaning o data, we have renamed the EVTYPE variable to "eventtype." We initially explore the data stored within the variable:

```
format(stormDat[                              ## Random sample of 10 observations
               sample(1:length(stormDat$eventtype), 10),
               6:8], justify = "right")
```

```
       countyname state   eventtype
460057     TAYLOR    KY FLASH FLOOD
58486    MARSHALL    KY   TSTM WIND
420890    JOHNSON    IN        HAIL
545967    WHEELER    NE        HAIL
85935        CASS    MO   TSTM WIND
554619    DOUGLAS    SD   TSTM WIND
164606      SMITH    TX     TORNADO
862831     COLLIN    TX        HAIL
813757  CHRISTIAN    MO        HAIL
616455    JACKSON    KS   TSTM WIND
```

Additional Data Processing In addition to the state abbreviation and date/time issues at least partially addressed above, the event type field contains much duplicate data in various text formats, different levels of information, duplicate data, incorrect data (e.g. Forest Fires are reported, yet not considered a weather event), and data that should be entered in other columns of the data table.

We extend modifications to a few additional cleaning changes in the event type variable by eliminating punctuation markings, changing text to upper case, and eliminating a few of the most obvious spelling and data entry errors. We also take the opportunity to classify a few of the variable entries to the current standard, but this work is to be considered neither precise nor complete.

Additionally, data entered as summaries of multistate, multiple event types (e.g. TX and OK Tornado and thunderstorm events) are cross-referenced to the remarks variable, where the actual event classification data has been listed. An obvious improvement could be had by developing code to parse the remarks variable for types or combinations of events – however, considering the extensive coding needed for correcting most event type data entries, we consider additional effort out-of-scope unless the "SEE REMARKS" cross-reference is found to corr significant numbers of deaths or injuries.

```
convertEvent <- function(dataVect){          ## Function parses input string (in-
  as.character(dataVect) %>%                 ## tended for stormDat$eventtype)
  toupper() %>%                              ## and makes specific changes to
  chainGsub("&" , " AND ") %>%               ## partly normalize the values to
  chainGsub("/" , " ") %>%                   ## the NWS standard strings.
  chainGsub("-" , " ") %>%
  chainGsub("\\?" , "NONE") %>%
  chainGsub("\\; " , " ") %>%
  chainGsub("\\;$" , "") %>%
  chainGsub("\\\\" , " ") %>%
  chainGsub(" $" , "") %>%
  chainGsub("^  " , "") %>%
  chainGsub("^ " , "") %>%
  chainGsub("ABNORMAL WARMTH" , "EXTREME HEAT") %>%
  chainGsub("ABNORMALLY" , "UNSEASONABLY") %>%
  chainGsub(" AND$" , "") %>%
  chainGsub("APACHE COUNTY" , "THUNDERSTORM WIND") %>%
  chainGsub("ASHFALL" , "VOLCANIC ASH") %>%
  chainGsub("AVALANCE" , "AVALANCHE") %>%
  chainGsub("BITTER WIND CHILL TEMPERATURES" , "EXTREME COLD WIND CHILL") %>%
  chainGsub("BLIZZARD HEAVY" , "BLIZZARD AND HEAVY") %>%
  chainGsub("BLIZZARD SUMMARY" , "BLIZZARD") %>%
  chainGsub("BLIZZARD WEATHER" , "BLIZZARD") %>%
  chainGsub("CHILL TEMPERATURE$" , "CHILL") %>%
  chainGsub("CHILLS" , "CHILL") %>%
```

```
chainGsub("CHIL$" , "CHILL") %>%
chainGsub("CHI$" , "CHILL") %>%
chainGsub("CH$" , "CHILL") %>%
chainGsub("^COL$" , "COLD") %>%
chainGsub("COASTAL FLOODING" , "COASTAL FLOOD") %>%
chainGsub("COASTALF" , "COASTAL F") %>%
chainGsub("COASTALS" , "COASTAL S") %>%
chainGsub(" CLOUDS$" , " CLOUD" ) %>%
chainGsub(" CLOU$" , " CLOUD" ) %>%
chainGsub("CSTL " , "COASTAL ") %>%
chainGsub("DAMAGE TO" , "DAMAGE") %>%
chainGsub("DEVEL" , "DEVIL") %>%
chainGsub("DUST DEVIL" , "TORNADO" ) %>%
chainGsub("DUSTSTORM" , "DUST STORM") %>%
chainGsub("DRIZZLE AND FREEZING" , "DRIZZLE") %>%
chainGsub("DRYNESS" , "DROUGHT") %>%
chainGsub("DRY CONDITIONS" , "DROUGHT") %>%
chainGsub("DRY PATTERN" , "DROUGHT") %>%
chainGsub("DRY SPELL" , "DROUGHT") %>%
chainGsub("DRY WEATHER" , "DROUGHT") %>%
chainGsub("^DRY$" , "DROUGHT") %>%
chainGsub(" DR$" , " DROUGHT") %>%
chainGsub("DUS$" , "DUST") %>%
chainGsub("EROSIN" , "EROSION") %>%
chainGsub(" EROSIO$" , " EROSION") %>%
chainGsub("EROSION COASTAL FLOOD" , "COASTAL FLOOD EROSION") %>%
chainGsub("EXCESSIVE HEAT DROUGHT" , "DROUGHT EXCESSIVE HEAT") %>%
chainGsub("EXCESSIVELY DRY" , "DRY SPELL") %>%
chainGsub("EXCESSIVE" , "EXTREME") %>%
chainGsub("EXTREME RECORD" , "EXTREME") %>%
chainGsub("EXTREME WETNESS" , "EXTREME PRECIPITATION") %>%
chainGsub("SEVERE COLD" , "EXTREME COLD") %>%
chainGsub("FIR$" , "FIRE") %>%
chainGsub("FIRES" , "FIRE") %>%
chainGsub("FREEZ$" , "FREEZE") %>%
chainGsub("FREEZING RA$" , "FREEZING RAIN") %>%
chainGsub("FROS$" , "FROST") %>%
chainGsub("FROSTFREEZE" , "FROST FREEZE") %>%
chainGsub("FLASHFLOOD" , "FLASH FLOOD") %>%
chainGsub("FLD$" , "FLOOD") %>%
chainGsub("FLOODING FLOOD$" , "FLOOD") %>%
chainGsub(" FLOOODING$" , " FLOOD") %>%
chainGsub("FLOOD FLOODING" , "FLOOD") %>%
chainGsub("FLOOD FLASH" , "FLASH FLOOD") %>%
chainGsub("FLOOD RIVER" , "FLOOD") %>%
chainGsub("FLOODIN" , "FLOOD") %>%
```

```
chainGsub("FLOODINGG" , "FLOOD") %>%
chainGsub("FLDG" , "FLOOD") %>%
chainGsub("FLOODG" , "FLOOD") %>%
chainGsub("FLOODS" , "FLOOD") %>%
chainGsub("FLOODING EROSION" , "FLOOD EROSION") %>%
chainGsub("FLOOD FLOOD" , "FLOOD") %>%
chainGsub("FUNNE$" , "FUNNEL") %>%
chainGsub("FUNNELS$" , "FUNNEL") %>%
chainGsub("SNOW ICESTORM" , "SNOW AND ICE STORM") %>%
chainGsub("SNOW ICE$" , "SNOW AND ICE STORM") %>%
chainGsub("SNOW AND ICE$" , "SNOW AND ICE STORM") %>%
chainGsub("ICE ON ROAD" , "ICY ROADS") %>%
chainGsub("ICE ROADS" , "ICY ROADS") %>%
chainGsub("GUSTNADO" , "THUNDERSTORM WIND") %>%
chainGsub("HAIL[0-9]" , "HAIL") %>%
chainGsub("HAIL [0-9]{1,5}$" , "HAIL") %>%
chainGsub("HAIL STORM" , "HAIL") %>%
chainGsub("SMALL HAIL" , "HAIL") %>%
chainGsub("HEAT WAVES{0,1}" , "EXCESSIVE HEAT") %>%
chainGsub("HEATBURST" , "EXCESSIVE HEAT") %>%
chainGsub("HURRICANE$" , "HURRICANE TYPHOON") %>%
chainGsub("HURRICANE [EFGO][A-Z]{1,25}" , "HURRICANE TYPHOON") %>%
chainGsub("HURRICANE [A-Z ]{1,25}WIND" , "HURRICANE TYPHOON") %>%
chainGsub("HURRICANE [A-Z ]{1,25}SWELLS" , "STORM SURGE") %>%
chainGsub("HVY" , "HEAVY") %>%
chainGsub("LIGHTING" , "LIGHTNING") %>%
chainGsub("LIGNTING" , "LIGHTNING") %>%
chainGsub("LIGNTNING" , "LIGHTNING") %>%
chainGsub("LIGHTNINGNONE" , "LIGHTNING") %>%
chainGsub("LOW TEMPERATURE RECORD" , "EXTREME COLD") %>%
chainGsub("RECORD LOW$" , "EXTREME COLD") %>%
chainGsub("RECORD COLD$" , "EXTREME COLD") %>%
chainGsub("LOW WIND CHILL" , "WIND CHILL") %>%
chainGsub(", MAY 26$" , "") %>%
chainGsub("MICO" , "MICRO") %>%
chainGsub("MIRCO" , "MICRO") %>%
chainGsub("MUDSLIDES" , "MUD SLIDE") %>%
chainGsub("MUDSLIDE" , "MUD SLIDE") %>%
chainGsub("MUD SLIDE LANDSLIDE" , "MUD SLIDE") %>%
chainGsub("NO SEVERE WEATHER" , "NONE") %>%
chainGsub("NON SEVERE " , "") %>%
chainGsub("NON THUNDERSTORM WIND" , "STRONG WIND") %>%
chainGsub("^OTHER$" , "NONE") %>%
chainGsub(" PLUME$" , "") %>%
chainGsub(" PRECIPATATION$" , " PRECIPITATION") %>%
chainGsub(" PRECIPITATIO$" , " PRECIPITATION") %>%
```

```
chainGsub(" PRECIP$" , " PRECIPITATION") %>%
chainGsub("^PROLONGED " , "EXTENDED ") %>%
chainGsub("^PROLONG " , "EXTENDED ") %>%
chainGsub("RSPOUT" , "R SPOUT") %>%
chainGsub("RAIN AND SNOW" , "RAIN SNOW") %>%
chainGsub("RAIN SLEET AND LIGHT$" , "RAIN AND SLEET") %>%
chainGsub("RAIN SLEET$" , "RAIN AND SLEET") %>%
chainGsub(" RAINFALL$" , " RAIN") %>%
chainGsub("RAINSTORM$" , "RAIN STORM") %>%
chainGsub("RAIN HEAVY" , "HEAVY RAIN") %>%
chainGsub("RECORD EXTREME" , "EXTREME") %>%
chainGsub("RECORD HIGH TEMPERATURE" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD HIGH$" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD HEAT$" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD HEAT WAVE" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD WARM$" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD WARM TEMPS$" , "EXCESSIVE HEAT") %>%
chainGsub("RECORD PRECIPITATION" , "EXTREME PRECIPITATION") %>%
chainGsub("RIP CURRENTS" , "RIP CURRENT") %>%
chainGsub("RIVER FLOOD$" , "FLOOD") %>%
chainGsub("SLEET FREEZING RAIN" , "WINTER STORM") %>%
chainGsub("SLEET RAIN SNOW" , "WINTER STORM") %>%
chainGsub("SLEET SNOW" , "WINTER STORM") %>%
chainGsub("SLEET STORM" , "SLEET") %>%
chainGsub("SLEET AND FREEZING RAIN" , "WINTER STORM") %>%
chainGsub("SLIDES" , "SLIDE") %>%
chainGsub("SML" , "SMALL") %>%
chainGsub("SMALL STREAM AND URBAN" , "") %>%
chainGsub("SMALL STREAM URBAN" , "") %>%
chainGsub(" SMALL$" , " FLOOD") %>%
chainGsub(" SMALL STREAM FLOOD$" , "FLOOD") %>%
chainGsub(" SNOWFALL$" , " SNOW") %>%
chainGsub("SNOW AND EXTREME" , "SNOW EXTREME") %>%
chainGsub("SNOW ANDBLOWING" , "SNOW AND BLOWING") %>%
chainGsub("SNOW BLOWING" , "SNOW AND BLOWING") %>%
chainGsub("SNOW HEAVY SNOW" , "HEAVY SNOW") %>%
chainGsub("SNOW SLEET RAIN" , "WINTER STORM") %>%
chainGsub("SQUALL$" , "SQUALLS") %>%
chainGsub("NEAR RECORD SNOW$" , "HEAVY SNOW") %>%
chainGsub(" SNO$" , " SNOW") %>%
chainGsub(" SNOWS" , " SNOW") %>%
chainGsub("^SNOW ICE$" , "HEAVY SNOW ICE STORM") %>%
chainGsub("SPOUTS" , "SPOUT") %>%
chainGsub("SPOUT " , "SPOUT") %>%
chainGsub("SPOUTT" , "SPOUT T") %>%
chainGsub("SPOUTF" , "SPOUT F") %>%
```

```
chainGsub("STRM" , "STREAM") %>%
chainGsub("STREA$" , "STREAM") %>%
chainGsub("STREAM FLOOD$" , "FLOOD") %>%
chainGsub("STREET FLOOD$" , "FLOOD") %>%
chainGsub("STREA$" , "STREAM") %>%
chainGsub("STROM" , "STORM" ) %>%
chainGsub("STORMS$" , "STORM") %>%
chainGsub("STORMS WIND" , "STORM WIND") %>%
chainGsub("STORMIND" , "STORM WIND") %>%
chainGsub("STORMSS$" , "STORMS") %>%
chainGsub("STORMSS" , "STORMS") %>%
chainGsub("STORMWIND" , "STORM WIND") %>%
chainGsub("STORMW$" , "STORM") %>%
chainGsub("STORMW " , "STORM ") %>%
chainGsub("STORMS W" , "STORM W") %>%
chainGsub("STORMSW" , "STORM") %>%
chainGsub("SURG$" , "SURGE") %>%
chainGsub("TIDES$" , "TIDE") %>%
chainGsub("TEMPERATURES$" , "TEMPERATURE") %>%
chainGsub("LOW TEMPERATURE" , "COLD") %>%
chainGsub("HIGH TEMPERATUE" , "HEAT") %>%
chainGsub("TREE$" , "TREES") %>%
chainGsub("TROPICAL STORMS " , "TROPICAL STORM ") %>%
chainGsub("TORNADOS" , "TORNADOES") %>%
chainGsub("TORNDAO" , "TORNADO") %>%
chainGsub("TSTM" , "THUNDERSTORM") %>%
chainGsub("TSORM" , "STORM" ) %>%
chainGsub("TORRENTIAL RAINFALL" , "HEAVY RAIN") %>%
chainGsub("THUNDERSTORMS DAMAGE TO" , "THUNDERSTORM WIND") %>%
chainGsub("THUNDERSTORMW" , "THUNDERSTORM") %>%
chainGsub("TUNDERSTORM" , "THUNDERSTORM") %>%
chainGsub("THUNDERTORM" , "THUNDERSTORM") %>%
chainGsub("THUNDEERSTORM" , "THUNDERSTORM") %>%
chainGsub("THUDERSTORM" , "THUNDERSTORM") %>%
chainGsub("THUNDERESTORM" , "THUNDERSTORM") %>%
chainGsub("THUNERSTORM" , "THUNDERSTORM") %>%
chainGsub("THUNDESTORM" , "THUNDERSTORM") %>%
chainGsub("UNSEASONABLE" , "UNSEASONABLY") %>%
chainGsub("UNSEASONAL LOW TEMP" , "COLD") %>%
chainGsub("UNUSUAL RECORD WARMTH" , "UNSEASONABLY WARM") %>%
chainGsub("UNUSUAL WARMTH" , "HEAT") %>%
chainGsub("UNUSUALLY WARM" , "HEAT") %>%
chainGsub("UNUSUALLY COLD" , "COLD") %>%
chainGsub("URBAN FLOODING" , "FLOOD") %>%
chainGsub("URBAN AND SMALL" , "URBAN SMALL") %>%
chainGsub("URBAN SMALL FLOOD$" , "FLOOD") %>%
```

```
chainGsub("^URBAN AND$" , "") %>%
chainGsub("URBAN SMALL STREAM$" , "FLOOD") %>%
chainGsub("URBAN FLOOD" , "FLOOD") %>%
chainGsub("VERY DRY" , "DROUGHT" ) %>%
chainGsub("VERY WARM" , "HEAT" ) %>%
chainGsub("VOG" , "VOLCANIC ASH" ) %>%
chainGsub("VOLCANIC ERUPTION" , "VOLCANIC ASH" ) %>%
chainGsub("VOLCANIC VOLCANIC" , "VOLCANIC" ) %>%
chainGsub("WARMT$" , "WARM") %>%
chainGsub("WARM WET" , "WARM AND WET") %>%
chainGsub("WARM YEAR" , "WARM") %>%
chainGsub("WARM TEMPS$" , "WARM") %>%
chainGsub("WARMTH" , "WARM") %>%
chainGsub("WATCHILL" , "WATCH") %>%
chainGsub("WAV$" , "WAVE") %>%
chainGsub("WAYTER" , "WATER") %>%
chainGsub("^ATER" , "WATER") %>%
chainGsub(" WAUSEON$" , "") %>%
chainGsub("WEATHE$" , "WEATHER") %>%
chainGsub("WHIRLWIND" , "TORNADO" ) %>%
chainGsub("WILD FIRE" , "WILDFIRE" ) %>%
chainGsub("WILD FOREST FIRE" , "WILDFIRE" ) %>%
chainGsub("WINDCHILL" , "WIND CHILL") %>%
chainGsub("WINDTER" , "WINTER") %>%
chainGsub("WI$" , "WIND") %>%
chainGsub("WINDS" , "WIND") %>%
chainGsub("WIN$" , "WIND") %>%
chainGsub("WINDHAIL" , "WIND HAIL") %>%
chainGsub("WINS$" , "WIND") %>%
chainGsub("WND" , "WIND") %>%
chainGsub("W INDS" , "WIND") %>%
chainGsub("?MPH" , "") %>%
chainGsub("[0-9]{1,5}" , "") %>%
chainGsub("WINTRY" , "WINTERY") %>%
chainGsub("WIND CHILL TEMPERATURE" , "WIND CHILL") %>%
chainGsub(" WIND WIND " , " WIND AND WIND ") %>%
chainGsub("WINTER WEATHER MIX" , "WINTER STORM") %>%
chainGsub("WINTER MIX" , "WINTER STORM") %>%
chainGsub("WINTERY MIX" , "WINTER STORM") %>%
chainGsub("WX" , "WEATHER") %>%
chainGsub("WINTER WEATHER$" , "WINTER STORM") %>%
chainGsub("\\." , "") %>%
chainGsub("   " , " ") %>%
chainGsub("  " , " ") %>%
chainGsub("\\)" , "") %>%
chainGsub("\\(" , "") %>%
```

```
  chainGsub(" $" , "") %>%
  chainGsub("^$" , "NONE") %>%
  chainGsub("FLOOD FLOOD$" , "FLOOD") %>%
  chainGsub("FLOOD STREET" , "FLOOD") %>%
  chainGsub("URBAN" , "") %>%
  chainGsub("COLD WAVE" , "COLD") %>%
  chainGsub("COLD WEATHER" , "COLD") %>%
  chainGsub("UNSEASONABLY COLD" , "COLD") %>%
  chainGsub("HIGH WAVES" , "HEAVY SURF") %>%
  chainGsub("HIGH SURF" , "HEAVY SURF") %>%
  chainGsub("HYPERTHERMIA EXPOSURE" , "HYPOTHERMIA") %>%
  chainGsub(" [A-Z]$" , "") %>%
  chainGsub("EXTREME RAIN" , "HEAVY RAIN") %>%
  chainGsub(":" , "") %>%
  chainGsub("\\," , "") %>%
  chainGsub("SUMMARY [ A-Z]{1,20}$" , "SEE REMARKS") %>%
  chainGsub("^ {0,3}" , "") %>%
  chainGsub("AND FLOOD" , "FLOOD") %>%
  chainGsub("ANDFLOOD" , "FLOOD") %>%
  return()
}
eventVect <- convertEvent(stormDat$eventtype)
print(                                      ## Reduction in event types through
  paste0("Index reduced by ",               ## consolidation.
         length(sort(unique(stormDat$eventtype))) -
         length(sort(unique(eventVect))),
         " event type/names" ))
```

```
[1] "Index reduced by 539 event type/names"
```

This code (~265 lines) results in consolidation of 'eventtype' unique entry types by roughly half. While this is too large an index (an order of magnitude greater) than the standard data definition, it should cover the majority of the observations in this data set well enough to estimate results.

Rather than discarding the original event type variable, the vector of modified eventtype strings is inserted into the stormDat table as new variable 'eventclass' which will be used for these analyses where applicable.

```
stormDat <-
  (mutate(                                  ## Inserts new event classification
          stormDat,                         ## vector as variable in position 9
          eventclass = eventVect)
          )[, c(1:8, 37, 9:36)]
print(stormDat[sample(1:length(eventVect),  ## Print some of new data showing
```

```
                    20),                  ## substitution/interpretation
                c(6:9)])
```

```
                countyname state         eventtype        eventclass
541589           FARIBAULT    MN         TSTM WIND THUNDERSTORM WIND
816116              VALLEY    NE       FLASH FLOOD       FLASH FLOOD
189590             YAVAPAI    AZ              HAIL              HAIL
870301            CHRISTIAN    MO              HAIL              HAIL
696601            ST. JOHNS    FL              HAIL              HAIL
766728              HOUSTON    AL THUNDERSTORM WIND THUNDERSTORM WIND
584530                SARPY    NE         TSTM WIND THUNDERSTORM WIND
629126           LIVINGSTON    NY         TSTM WIND THUNDERSTORM WIND
585488               CUMING    NE              HAIL              HAIL
312871              TRINITY    CA              HAIL              HAIL
599562 CHARLOTTESVILLE (C)    VA         TSTM WIND THUNDERSTORM WIND
59092                 BOYLE    KY         TSTM WIND THUNDERSTORM WIND
367720                LUCAS    OH              HAIL              HAIL
453299            CHATTOOGA    GA         TSTM WIND THUNDERSTORM WIND
573563             BUCHANAN    IA         TSTM WIND THUNDERSTORM WIND
725188               ORANGE    NY              HAIL              HAIL
133352            MCCURTAIN    OK         TSTM WIND THUNDERSTORM WIND
692283             MARICOPA    AZ THUNDERSTORM WIND THUNDERSTORM WIND
502082              GRENADA    MS       FLASH FLOOD       FLASH FLOOD
574178                CLOUD    KS              HAIL              HAIL
```

The assignment question asks which types of events are most harmful with respect to public health. The data include the number of fatalaties and injuries caused by many weather events. Since the data set contains many death and injury entries that are blank or zero, it can be assumed that most events cause few or no injuries or death. We will report the three event types that cause the most harm in both categories.

We use the "fatalities" and "injuries" variables to plot effects of weather on public health, discarding observations where the number of fatalities or injuries is less than the mean.

As stated above, we suspect the focus of collections on tornadoes, prior to 1992, has skewed the results to favor these events. We also recompute to determine the effect of any skew by looking only at data recorded after that date.

```
sumInj <-                             ## Small data frame contains events
  data.frame(eventclass = unique(     ##  by injuries
                            stormDat[stormDat$injuries >= 1,
                                     ]$eventclass),
             injuries = 0)
for(evCl in sumInj$eventclass){       ## Originally written to use the
```

```
  sumInj[sumInj$eventclass == evCl,         ## dpyr::summarise(group_by())
         ]$injuries <-                      ## phrasing, software update
    sum(stormDat[stormDat$eventclass == evCl,  ## dependencies disabled this
                 ]$injuries)                ## function during development. The
}                                           ## for() loop is much slower, and
sumInj <- sumInj[order(sumInj$injuries,     ## should be replaced, but produces
                 decreasing = TRUE), ][1:6, ]  ## the same results.
P1 <- ggplot(sumInj,                        ## ggplot2 graphics, simple box
             aes(x = eventclass,            ## plot.
                 y = injuries)) +
      geom_boxplot() +
      ggtitle("Injuries by Weather Event") +
      xlab("Weather Event Type") +
      ylab("Injuries") +
      scale_y_continuous(limits=range(sumInj$injuries)) +
      scale_x_discrete(limits=sumInj$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 25,
                                       hjust = 1))
sumFatal <-                                 ## Small data frame containing
  data.frame(eventclass = unique(           ## deaths by event type.
                            stormDat[stormDat$fatalities >= 1,
                                     ]$eventclass),
             fatalities = 0)
  for(evCl in sumFatal$eventclass){
    sumFatal[sumFatal$eventclass == evCl,
             ]$fatalities <-
      sum(stormDat[stormDat$eventclass == evCl,
                   ]$fatalities)
}
sumFatal <- sumFatal[order(sumFatal$fatalities,
                     decreasing = TRUE), ][1:6, ]
P2 <- ggplot(sumFatal,
             aes(x = eventclass,
                 y = fatalities)) +
      geom_boxplot() +
      ggtitle("Fatalities by Weather Event") +
      xlab("Weather Event Type") +
      ylab("Fatalities") +
      scale_x_discrete(limits=sumFatal$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 25,
                                       hjust = 1))
sumInj92 <-                                 ## Similar to injury dataframe.
  data.frame(                               ## above, but discards all data
    eventclass=unique(                      ## observations before 1992.
```

```
        stormDat[stormDat$injuries >= 1 &
        stormDat$begindate >=
          as.Date("1992-01-01"),
               ]$eventclass),
    injuries92 = 0)
for(evCl in sumInj92$eventclass) {
  sumInj92[sumInj92$eventclass == evCl,
           ]$injuries92 <-
    sum(stormDat[stormDat$eventclass == evCl &
                 stormDat$begindate >= as.Date("1992-01-01"),
                 ]$injuries)
}
sumInj92 <-
  sumInj92[order(sumInj92$injuries92,
           decreasing = TRUE), ][1:6, ]

P3 <- ggplot(sumInj92,
             aes(x = eventclass,
                 y = injuries92)) +
      geom_boxplot() +
      ggtitle("Injuries by Weather Event since 1992") +
      xlab("Weather Event Type") +
      ylab("Injuries") +
      scale_x_discrete(limits=sumInj92$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 25,
                                       hjust = 1))
sumFatal92 <-                              ## Data frame contains deaths since
  data.frame(                              ## 1992 by event type.
    eventclass = unique(
      stormDat[stormDat$fatalities >= 1 &
      stormDat$begindate >=
        as.Date("1992-01-01"), ]$eventclass),
    fatalities92 = 0)
for(evCl in sumFatal92$eventclass) {
  sumFatal92[sumFatal92$eventclass == evCl, ]$fatalities92 <-
    sum(stormDat[stormDat$eventclass == evCl &
                 stormDat$begindate >= as.Date("1992-01-01"),
                 ]$fatalities)
}
sumFatal92 <-
  sumFatal92[order(sumFatal92$fatalities92,
                   decreasing = TRUE), ][1:6, ]
P4 <- ggplot(sumFatal92, aes(x = eventclass,
                             y = fatalities92)) +
      geom_boxplot() +
```

```
      ggtitle("Fatalities by Weather Event since 1992") +
      xlab("Weather Event Type") +
      ylab("Fatalities") +
      scale_x_discrete(limits=sumFatal92$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 25,
                                       hjust = 1))
```

Top 3 number of injuries caused by weather events for all database entries are shown with the data gathered since 1992:

```
mutate(sumInj[1:3, ],                        ## Report top 3 causes of injury
       eventSince92 = sumInj92[1:3, ]$eventclass,
       injuriesSince92 = sumInj92[1:3, ]$injuries)
```

```
          eventclass injuries eventSince92 injuriesSince92
1            TORNADO    91407      TORNADO           24694
2  THUNDERSTORM WIND     9369        FLOOD            6873
12             FLOOD     6873 EXTREME HEAT            6680
```

The three greatest wether event causes of deaths are listed with the data gathered since 1992.

```
mutate(sumFatal[1:3, ],
       eventsSince92 = sumFatal92[1:3, ]$eventclass,
       fatalitiesSince92 = sumFatal92[1:3, ]$fatalities)
```

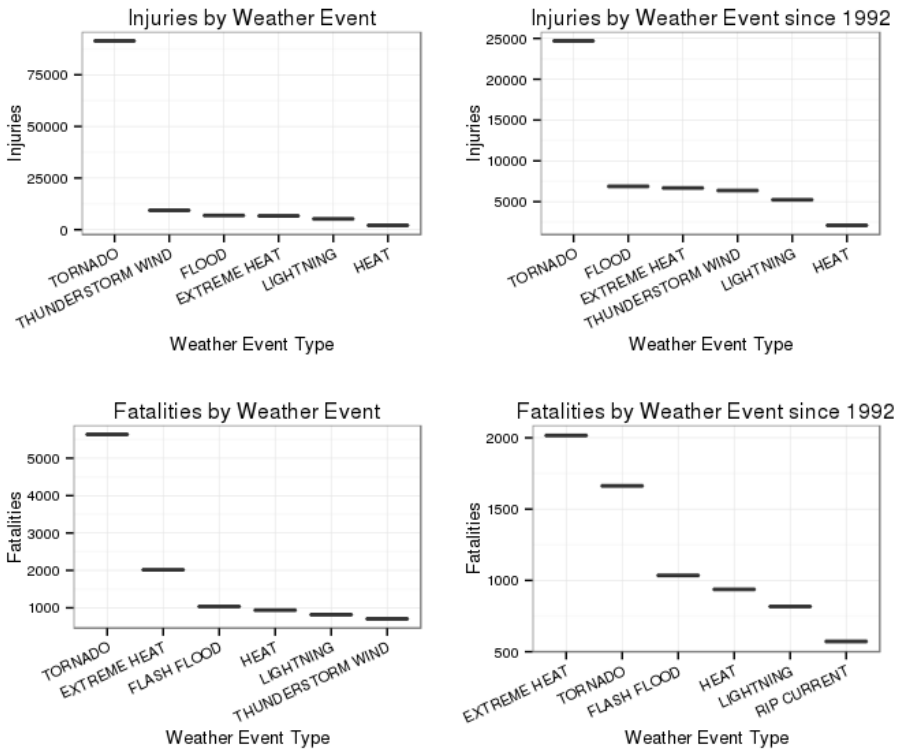
```
     eventclass fatalities eventsSince92 fatalitiesSince92
1       TORNADO       5636  EXTREME HEAT              2016
20 EXTREME HEAT       2016       TORNADO              1663
13  FLASH FLOOD       1035   FLASH FLOOD              1035
```

Question 1 Conclusions

```
multiplot(P1, P2, P3, P4, cols = 2)           ## multipane plots to screen
```

Figure 1 Injuries and Deaths from Weather Events Plots `(r plot1Write, echo = FALSE, cache = TRUE } dev.copy(png, width = 720, height = 720, file = "multiPlot-1.png") dev.off()`

Prior to 1992, with the greater period of reporting concentrated only of reports only on tornado events, it is perhaps not surprising that tornadoes lead the list


Injuries by Weather Event
Injuries
75000
50000
25000
0
TORNADO
THUNDERSTORM WIND
FLOOD
EXTREME HEAT
LIGHTNING
HEAT
Weather Event Type
Injuries by Weather Event since 1992
Injuries
25000
20000
15000
10000
5000
TORNADO
FLOOD
EXTREME HEAT
THUNDERSTORM WIND
LIGHTNING
HEAT
Weather Event Type
Fatalities by Weather Event
Fatalities
5000
4000
3000
2000
1000
TORNADO
EXTREME HEAT
FLASH FLOOD
HEAT
LIGHTNING
THUNDERSTORM WIND
Weather Event Type
Fatalities by Weather Event since 1992
Fatalities
2000
1500
1000
500
EXTREME HEAT
TORNADO
FLASH FLOOD
HEAT
LIGHTNING
RIP CURRENT
Weather Event Type

in injuries and fatalities when all years of data are included.
Thunderstorm Wind events, often reported with or in the same general seasons and conditions as tornadoes, rate as the second greatest causes of injuries.

The influence of tornadoes is so great that, even after removal of more than 40 years of observations, tornado injuries lead the category. The number of flood injury victims (almost unchanged from the information derived from all data sources) replaces the former second-place event: thunderstorm winds. Injuries from extreme heat are now rated third.

Compared to the full-range of observations, since 1992, the order of the first and second weather event causes of death are reversed: extreme heat-related fatalities replaced tornadoes as the leading cause. Flash flood deaths remain the third greatest after pre-1992 data are removed.

Question 2 Analysis

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

The data set contains property damage estimates in four variables as modified here: "propertydamage," "propertydamageexp," "cropdamage," and "cropdamageexp." The two variables containing the terminal substring "exp" are intended to be exponents/multiplying factors to apply to the corresponding damage estimate, which, when not blank or 0, is usually a real number with no more than two decimal places of precision, and no more than 5 digits. Encoding of the exponent variable is not straight forward:

```
unique(stormDat$propertydamageexp)          ## Show property exponent variable
```

```
 [1] K M   B m + 0 5 6 ? 4 2 3 h 7 H - 1 8
Levels:  - ? + 0 1 2 3 4 5 6 7 8 B h H K m M
```

```
unique(stormDat$cropdamageexp)              ## Show the crop exponent variable
```

```
[1]   M K m B ? 0 k 2
Levels:  ? 0 2 B k K m M
```

Additional Data Processing: Per the documentation from the NWS web site, letters h, k, m, b (upper and lower case) represent hundreds, thousands, millions, and billions, respectively. Factors "-" and "+" are interpreted to mean "less than" and "greater than," respectively, and for estimation purposes will be ignored. The "?,"0," "1," and empty values will be considered equal to NA. Useful values for exponents represented as characters are 2, 3, 6, 9 (equivalent to H, K, M, B). Numeric values greater than 1 will be evaluated as powers of ten (10^exp) * the respective damage variable data.

```
cleanExpnt <- function(expVect) {           ## Function replaces exponent values
  as.character(expVect) %>%                 ## with useful numeric equivalents
  toupper() %>%                             ## and strips out unused symbols and
  chainGsub("H" , "2") %>%                  ## values
  chainGsub("K" , "3") %>%
  chainGsub("M" , "6") %>%
  chainGsub("B" , "9") %>%
  chainGsub("\\?" , "") %>%
  chainGsub("\\+" , "") %>%
  chainGsub("\\-" , "") %>%
  chainGsub("0" , "") %>%
  chainGsub("1" , "") %>%
  return()
 }
stormDat$cropdamageexp <-
  as.factor(cleanExpnt(stormDat$cropdamageexp))
stormDat$propertydamageexp <-
  as.factor(cleanExpnt(stormDat$propertydamageexp))
unique(stormDat$propertydamageexp)          ## Show property exponent variable
```

```
[1] 3 6   9 5 4 2 7 8
Levels:  2 3 4 5 6 7 8 9
```

```
unique(stormDat$cropdamageexp)              ## Show the crop exponent variable
```

```
[1]   6 3 9 2
Levels:  2 3 6 9
```

We compute property and crop damage against the top event types, and report the three highest.

```
costDat <-                                  ## Temp data frame contains some
  data.frame(eventclass =                   ## filtered data. Multiplies dollar
               stormDat$eventclass,         ## value by the modified exponent.
             propDamage =
               (as.numeric(stormDat$propertydamage) *
              (10^as.numeric(stormDat$propertydamageexp))))
costDat <-
  costDat[costDat$propDamage >= 1, ]        ## Removes 0 value observations.
propDat <-                                  ## Data frame will store averages
  data.frame(eventclass =                   ## of property damage by event type.
               unique(costDat$eventclass),
             propDamage = 0)
for(dmgN in propDat$eventclass){            ## Originally written for dplyr::
```

```
  propDat[propDat$eventclass ==             ## summarise(group_by()), this
            dmgN, ]$propDamage <-           ## sectoion was rewritten with slow
    formatC(                                ## for() loop to provide the same
      mean(                                 ## output.  See explanation at P1,
        costDat[costDat$eventclass ==       ## comments, above.
                  dmgN, ]$propDamage)/(10^9),  ## Crop out to the $ billions
            width = 8,                      ## for reporting (reverse exponent
            digits = 2,                     ## to common magnitude).
            format = "f")
}
propDat <- propDat[order(                   ## Sort then discard unused data
                    propDat$propDamage,
                    decreasing = TRUE), ][1:6, ]
P5 <- ggplot(propDat, aes(x = eventclass,   ## ggplot 2
                          y = propDamage)) +
      geom_boxplot() +
      ggtitle("Property Damage by Weather Event") +
      xlab("Weather Event Type") +
      ylab("Property Damage ($ billions)") +
      scale_x_discrete(limits=propDat$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 60,
                                       hjust = 1))
costDat <-                                  ## Reuse temp data frame for crop
  data.frame(eventclass =                   ## damage losses. Multiply by expo-
               stormDat$eventclass,         ## nent.
             cropDamage =
               (as.numeric(stormDat$cropdamage) *
               (10^as.numeric(stormDat$cropdamageexp))))
costDat <-                                  ## Strip out 0 value observations.
  costDat[costDat$cropDamage >= 1, ]
cropDat <-                                  ## Build data frame to hold average
  data.frame(eventclass =                   ## damage by event.
               unique(costDat$eventclass),
             cropDamage = 0)
for(dmgN in cropDat$eventclass){            ## For loop to replace summarise(
  cropDat[cropDat$eventclass ==             ## group_by()), as above.
          dmgN, ]$cropDamage <-
                    formatC(
                      mean(                 ## Crop down to $ millions
                        costDat[costDat$eventclass ==
                                  dmgN, ]$cropDamage)/(10^6),
                      width = 8,
                      digits = 2,
                      format = "f")
}
```

```
cropDat <- cropDat[order(cropDat$cropDamage, ## discard unused observations.
                         decreasing = TRUE), ][1:6, ]
P6 <- ggplot(cropDat, aes(x = eventclass,
                          y = cropDamage)) +
      geom_boxplot() +
      ggtitle("Crop Damage by Weather Event") +
      xlab("Weather Event Type") +
      ylab("CropDamage ($ millions)") +
      scale_x_discrete(limits=cropDat$eventclass) +
      theme_bw(base_size = 10) +
      theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

The top 3 causes of property damage listed:

```
propDat[1:3, ]
```

```
                          eventclass propDamage
91        HEAVY RAIN SEVERE WEATHER       2.50
58 TORNADOES THUNDERSTORM WIND HAIL       1.60
5                 HURRICANE TYPHOON       0.41
```

The top 3 causes of crop damage are:

```
cropDat[1:3, ]
```

```
                eventclass cropDamage
53            EXTREME HEAT       1.66
35   EXTREME PRECIPITATION       1.42
34 COLD AND WET CONDITIONS       0.66
```

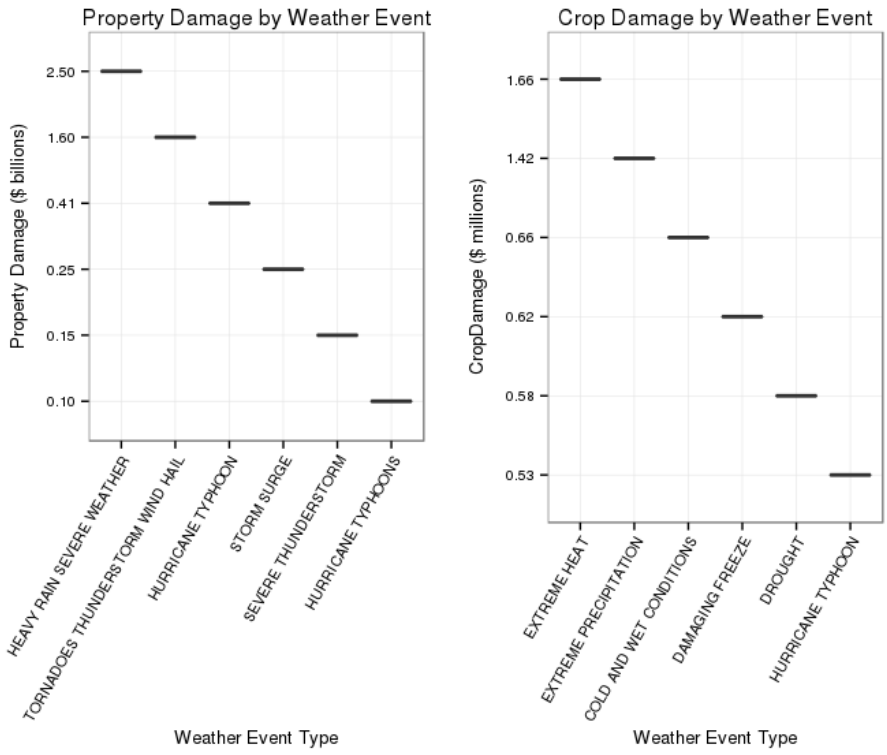
Question 2 Conclusions

Panel plot of the top property and crop damage-causing weather events.

```
multiplot(P5, P6, cols = 2)
```

Figure 2 Crop and Property Damage Plots `(r plot2Write, echo = FALSE, cache = TRUE } dev.copy(png, width = 720, height = 720, file = "plotPropCropDmg-1.png" ) dev.off()`

Since the average per type of event is taken, unlike the analysis of question #1, we do not recompute for observations since 1992. In initial analysis, we


Property Damage by Weather Event
Property Damage ($ billions)
2.50
1.60
0.41
0.25
0.15
0.10
HEAVY RAIN SEVERE WEATHER
TORNADOES THUNDERSTORM WIND HAIL
HURRICANE TYPHOON
STORM SURGE
SEVERE THUNDERSTORM
HURRICANE TYPHOONS
Weather Event Type
Crop Damage by Weather Event
CropDamage ($ millions)
1.66
1.42
0.66
0.62
0.58
0.53
EXTREME HEAT
EXTREME PRECIPITATION
COLD AND WET CONDITIONS
DAMAGING FREEZE
DROUGHT
HURRICANE TYPHOON
Weather Event Type

determined that changing the number and skew of the sample set does not alter the final results.

Per event, using all data collections since 1951, the greatest property damage is caused by events that combine heavy rain with severe weather. This multiple-event classification does not follow current NWS guidelines or definitions for identification of an event type. It is possible that this event could be combined with others, or split into several observations, if modern standard reporting methods were applied.

The event causing the second greatest property damage is also a combined event: tornadoes, thunderstorms, wind and hail. Readers should again note that following the NWS 2007 operating instruction, these should not be classified as a single event, but reported separately, which would no doubt modify the result.

It is possible that further analysis would cause analysts to combine damages for this second classification, the first classification, and those multiple-event/ thunderstorm systems that we have said should be classified by their remarks. Additionally, as combination events many of the summary reports for Oklahoma and Texas thunderstorm systems (event class modified to "see remarks," in our data processing, above) could be added to these observations. We estimate that these additions would increase the dollar value for the event, making the first event type even more distinctly the greatest cause.

Hurricanes/typhoons accounted for the type of events that cause the third greatest amount of property damage.

The events causing the greatest crop damage maximum are also combination types of events, "extreme heat" causing the greatest at an average of $1.6 Million. Second greatest cause of damage is "extreme precipitation" causing an average pf $1.4 Million. "Cold and wet conditions" rate third, causing an average of $660 thousand.

For crop damage, further study and classification of the NWS weather type/ classification might allow for modification of the results above. For instance, we suspect that "extreme heat" and "drought" (rated 4th at $618K) might be combined, and that "extreme precipitation" and various kinds of flood events, each of which were rated outside the top 6 margin, might be combined under the NWS standards.

Results

Since the database was originally oriented to collections for tornado events, it is perhaps not surprising that tornadoes and tornadic weather systems are among the leading causes of deaths, injuries, and property damage throughout the database history. In fatalities and injuries, after a limited data cleaning is performed, the disparity between the first and second causes is very large, and remains very large even when data collected for events prior to 1992 is not

considered. Likewise, the difference between the first and subsequent causes of property and crop damage are very high, although most of the top causes could be classified as tornadic weather events.

Despite expending a great deal of effort (approx 30% of all lines, ~ 50% of R sloc) to normalize weather event classification to a common set of standardized terms, the events leading to the greatest crop and property damage were combined types that do not apparently meet the NWS 2007 reporting guidelines. While refinement and expansion of the data to common standards would provide more specific and precise results, it is clearly not needed to determine these results. The significantly greater damage and injuries from tornadoes and tornado-like weather systems clearly indicate that policies and programs for detection, reporting, protection from, and mitigation of these events would make the greatest contribution to public safety.