Do More People Die of Extreme Heat or Excessive Cold?

Jim Callahan

September 25, 2015

Synopsis: The question of "Do More People Die of Extreme Heat or Excessive Cold?" is addressed using the **"R"** statistical language and data from the **National Oceanic and Atmospheric Administration (NOAA) "Storm Data" database**.

Data Processing The **R** language is an open source version of the the **S** language developed at Bell Labs during the "golden age" that also produced the **Unix** operating system and the **C** programming language. See: https://www.R-project.org/

Unlike **C**, **R** is an interpretive command language where the user types commands at the command line and gets an immediate response:

```
2+2
```

```
## [1] 4
```

```
sqrt(25)
```

```
## [1] 5
```

```
GaussDidThisInHisHeadInElemetarySchool <- sum(1:100)
print(GaussDidThisInHisHeadInElemetarySchool)
```

```
## [1] 5050
```

The last of these three examples is a problem solved by famous mathematician **Carl Friedrich Gauss (1777-1855)** while he was an eight year old child math prodigy in elementary school. He solved it in his head, amazing his teacher. For more of the story see: "Clever Carl" http://nrich.maths.org/2478/index?nomenu=1'

For those of us who are not (child or adult) math prodigies we can solve the problem with **R** by typing **sum(1:100)** at the command line. The **"<-"** assigns the result of the function to the variable name on the left.

The command line commands can be combined in simple text files "scripts" or combined with compliled programs, data and documentation to form complete "packages".

Many open source statisical "packages"" have been written in **R** making the complete system, the base language plus the optional downloadable statistical packages more powerful than proprietary statistical systems such as SAS or SPSS which, depending on the license, whose base price can be as much as $9,000 a seat. See: http://www.sas.com/store/products-solutions/cSoftware-p1.html

When we load the **NOAA "Storm Data" data file** into the **R** statistical system, we will also be using the **"<-"** to assign the result to the variable name on the left.

```
filename <- "~/GitHub/RepData_PeerAssessment2/data/repdata%2Fdata%2FStormData.csv.bz2"
NOAA <- read.csv(filename,
                 stringsAsFactors = FALSE )
```

```r
# Select Columns of interest:
#        Primary key:    REFNUM
#        Date and Time:  BGN_DATE, BGN_TIME, TIME_ZONE,
#        Location:       STATE, COUNTY, COUNTY_END, LATITUDE, LONGITUDE,
#        Type of Storm:  EVTYPE,
#        Damage:         PROPDMG, PROPDMGEXP, CROPDMG, CROPDMGEXP,
#        Casualties:     FATALITIES, INJURIES

columnsubset <- c("REFNUM", "BGN_DATE", "BGN_TIME", "TIME_ZONE",
                  "STATE", "COUNTY", "COUNTY_END", "LATITUDE", "LONGITUDE",
                  "EVTYPE", "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDMGEXP",
                  "FATALITIES", "INJURIES")

storms <- NOAA[, columnsubset]
str(storms)
```

```
## 'data.frame':    902297 obs. of  16 variables:
##  $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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" ...
##  $ STATE     : chr  "AL" "AL" "AL" "AL" ...
##  $ COUNTY    : num  97 3 57 89 43 77 9 123 125 57 ...
##  $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
##  $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
##  $ EVTYPE    : chr  "TORNADO" "TORNADO" "TORNADO" "TORNADO" ...
##  $ 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  "" "" "" "" ...
##  $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
##  $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
```

```r
#### Select Rows of interest
#### Select years of interest (Since 2001 "21st Century Storms" or "last 25 years")

#### Step 1: Create a year variable from date
#### Read in the date.
#### NOAA's Time and time zone are in separate variables that we will ignore.
datetime = as.POSIXct(storms$BGN_DATE, "%m/%d/%Y %H:%M:%S", tz = "")
storms$year <- format(datetime, "%Y")

#### Step 2: Filter by year
storms <- storms[storms$year >= "2001", ]

#### Remove the original NOAA database
rm(NOAA)
```

```r
#### Scale Property Damage and Crop Damage by thouands and millions
storms$PropertyDamage <- storms$PROPDMG
storms$CropDamage     <- storms$CROPDMG
```

```r
with(storms,
     {
         PropertyDamage <- ifelse(PROPDMGEXP == "K", PROPDMG * 1000, PROPDMG)
         PropertyDamage <- ifelse(PROPDMGEXP == "M", PROPDMG * 1000 * 1000, PROPDMG)
         # Round off total to nearest dollar.
         PropertyDamage <- round(PropertyDamage, digits = 0)

         CropDamage     <- ifelse(CROPDMGEXP == "K", CROPDMG * 1000, CROPDMG)
         CropDamage     <- ifelse(CROPDMGEXP == "M", CROPDMG * 1000 * 1000, CROPDMG)
         # Round off total to nearest dollar.
         CropDamage     <- round(CropDamage, digits = 0)
     }
)

#### Rename and put variables in logical order.
```

Results

```r
#### Tabulate Fatalities and Damage by Storm Type and Year
#### Userful R commands include table(), xtabs(), ftable()  or aggregate()
#### This use of aggregate() is based on Jared Lander's "R for Everyone" page 123
#### where he uses aggregate() on the diamonds data set from the ggplot2 package.
stormtab <- aggregate(cbind(FATALITIES, INJURIES, PropertyDamage, CropDamage)
                      ~ EVTYPE + year, storms, sum)
```

Tabulate Fatalities and Damage by Storm Type and Year

```r
# What type of storms caused the most fatalities in the most recent year (2011)?
mostrecentyear <- max(stormtab$year)
stormsrankyear <- stormtab[stormtab$year == mostrecentyear, ]

# This is the sort -- rank by fatalities in the stormrankyear
deadlystorms <- stormsrankyear[order(-stormsrankyear$FATALITIES, na.last = NA), ]
# head(deadlystorms, 10)
print(deadlystorms[1:10, c("EVTYPE", "FATALITIES", "INJURIES", "PropertyDamage", "CropDamage")],
      justify = "left")
```

Health Impact

```
##                EVTYPE FATALITIES INJURIES PropertyDamage CropDamage
## 630           TORNADO        587     6163      155464.51      18374
## 603       FLASH FLOOD         68       30       82067.98       5530
## 609              HEAT         63      611           0.00          0
## 604             FLOOD         58       10      133748.97      25002
## 629 THUNDERSTORM WIND         54      373      243100.34      16955
## 601    EXCESSIVE HEAT         36      138        1143.20          0
```

```
## 624       RIP CURRENT         29        27          0.00          0
## 619         LIGHTNING         26       194      34491.42        112
## 595   COLD/WIND CHILL         21         1         70.00          0
## 612         HIGH SURF         11        11        222.00          0
```

```r
# What type of storms caused the most fatalities in the most recent year (2011)?
damagestorms <- stormsrankyear[order(-stormsrankyear$PropertyDamage, na.last = NA), ]
head(damagestorms, 10)
```

Property Damage

```
##                EVTYPE year FATALITIES INJURIES PropertyDamage CropDamage
## 629 THUNDERSTORM WIND 2011         54      373      243100.34      16955
## 630           TORNADO 2011        587     6163      155464.51      18374
## 604             FLOOD 2011         58       10      133748.97      25002
## 603       FLASH FLOOD 2011         68       30       82067.98       5530
## 608              HAIL 2011          0       31       57214.06      26390
## 619         LIGHTNING 2011         26      194       34491.42        112
## 634          WILDFIRE 2011          6      116       33094.24       1805
## 613         HIGH WIND 2011          4       11       14728.25       1336
## 631    TROPICAL STORM 2011          4        1       11769.30        525
## 635      WINTER STORM 2011          1        0       11663.50         70
```

Event Descriptions indicating Hot or Cold Weather

```
COLD
COLD/WIND CHILL
COLD WIND CHILL TEMPERATURES
EXCESSIVE HEAT
EXTREME COLD
EXTREME COLD/WIND CHILL
EXTREME WINDCHILL
FREEZE
FREEZING DRIZZLE
FREEZING RAIN
FROST/FREEZE
HARD FREEZE
HEAT
RECORD WARMTH
UNSEASONABLY COLD
UNSEASONABLY COOL
UNSEASONABLY WARM
UNUSUALLY COLD
WINTER STORM
WINTER WEATHER
WINTER WEATHER/MIX
```

END