Health and Economic Consequences of Storm Events

Diane Leigh

May 27, 2017

Synopsis

Data Processing

```
# download and read Storm Data file
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
if(!file.exists("StormData.bz2"))download.file(fileUrl, "StormData.bz2")
if(!exists("StormData"))StormData <- read.csv("StormData.bz2")
```

We have two questions to answer: 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? So we created two subsets with only the necessary columns of data to answer the two (pophealth, economics) questions.

```
# Subset data to include only Reference number, EventData, Fatality, and Injury data to look at populat
pophealth <- as.data.table(StormData[, c("REFNUM", "EVTYPE", "FATALITIES", "INJURIES")])
pophealth <- pophealth[FATALITIES>0|INJURIES>0]

# Subset data to include only Reference number, EventData,  Proper Damage and Crop Damage data to look
economics <- as.data.table(StormData[, c("REFNUM","EVTYPE", "PROPDMG", "PROPDMGEXP", "CROPDMG", "CROPDM
economics <- economics[PROPDMG>0|CROPDMG>0]



# convert damage costs to dollars using multiplyers B = billions = 1E9, M = Millions = 1E6 and K = Thou
x <-levels(economics$PROPDMGEXP)
y <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1e9, 1e2, 1e2, 1e3, 1e6, 1e6)
df1 <- data.table(PROPDMGEXP=x, PRPMULT = y)
w <-levels(economics$CROPDMGEXP)
z <- c(1, 1, 1, 1, 1e9, 1e3, 1e3, 1e6, 1e6)
df2 <- data.table(CROPDMGEXP=w, CRPMULT = z)

setkey(df1, PROPDMGEXP)
setkey(economics, PROPDMGEXP)
economics <- economics[df1]
setkey(df2, CROPDMGEXP)
setkey(economics, CROPDMGEXP)
economics <- economics[df2]

economics <- economics[, ':='(prpcst = PROPDMG * PRPMULT)]
economics <- economics[, ':='(prpcst = CROPDMG * CRPMULT)]
economics <- economics[, ':='(cost = prpcst + prpcst)]
```

The EVTYPE column is not consistent with identifying an event (i.e. "COLD" and "COLD WAVE") There are 48 EVTYPE identifiers in the STORM DATA PREPARATION provided by the National Weather Service (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf), and 985 unique EVTYPE identifiers in the Storm Data. This next block cleans up the EVTYPE column in the subsetted data to match the events in the file supplied by the National Weather Service.

```
pophealth <- mutate_each(pophealth, funs(toupper))
# Get and combine unique event types in the subsetted data
popev <- data.table(unique(pophealth$EVTYPE, names("EVTYPE")))
names(popev) <- "EVTYPE"
popev <- mutate(popev, EVENT = "")

# Identify EVENTs which correspond to multiple EVTYPEs

for(i in 1:nrow(popev)){
       if(grepl(".*TORN.*|.*FUNNEL.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "TORNADO"
       }
     else if(grepl(".*HURRICANE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HURRICANE"
     }
     else if(grepl(".*AVALAN.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "AVALANCHE"
     }
     else if(grepl(".*BLIZ.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "BLIZZARD"
     }
     else if(grepl(".*SNOW.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HEAVY_SNOW"
     }
     else if(grepl(".*COASTAL.*FLOOD.*|.*TIDAL.*FLOOD.*|.*EROSION.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "COASTAL_FLOOD"
     }

     else if(grepl(".*FLOOD.*|.*RISING.*|.*WATER.*|.*FLD.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "FLOOD"
     }
     else if(grepl(".*COLD.*|.*CHILL.*|.*LOW TEMP.*|.*HYPOTHERM.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "COLD/WIND_CHILL"
     }
     else if(grepl(".*DROUGHT.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "DROUGHT"
     }
     else if(grepl(".*HEAT.*|.*HYPERTHERM.*|.*WARM.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HEAT"
     }
     else if(grepl(".*COASTAL.*STORM.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "COASTAL_STORM"
     }
     else if(grepl(".*LIG.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "LIGHTNING"
     }
     else if(grepl(".*MARINE.*THUND.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "MARINE_THUNDERSTORM_WIND"
     }
     else if(grepl(".*THUND.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "THUNDERSTORM WIND"
     }
     else if(grepl(".*DRY.*|.*DUST.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "DUST_STORM"
     }
     else if(grepl(".*SLIDE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "LANDSLIDE"
     }
     else if(grepl(".*RIP.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "RIP_CURRENT"
     }
     else if(grepl(".*FREEZE.*|.*FROST.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "ICE/FREEZE"
     }
     else if(grepl(".*FREEZING.*|.*GLAZE.*|.*ICE STORM.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "ICE_STORM"
     }
     else if(grepl(".*RAIN.*|.*DOWNBURST.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HEAVY_RAIN"
     }
     else if(grepl(".*SEA.*|.*SURF.*|.*SWELLS.*|.*WAVE.*|.*HIGH.*TIDE.*|.*SURGE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HIGH_SURF"
     }
     else if(grepl(".*LOW.*TIDE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "LOW_TIDE"
     }
     else if(grepl(".*ROAD.*|.*ICE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "ICY_ROAD"
     }
     else if(grepl(".*MARINE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "MARINE_HIGH_WIND"
     }
     else if(grepl(".*TROPICAL.*|.*TSTM.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "TROPICAL_STORM"
     }
     else if(grepl(".*WIND.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HIGH_WIND"
     }
     else if(grepl(".*FIRE.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "WILDFIRE"
     }
     else if(grepl(".*WINTER.*|.*WINTRY.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "WINTER_STORM"
     }
     else if(grepl(".*HAIL.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "HAIL"
     }
     else if(grepl(".*FOG.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "DENSE_FOG"
     }
     else if(grepl(".*TSU.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "TSUNAMI"
     }
     else if(grepl(".*TYPH.*", popev$EVTYPE[i])){
            popev$EVENT[i] <- "TYPHOON"
     }
```

```
     else {popev$EVENT[i] <- "OTHER"}
}

#Create column in subsetted data to group EVTYPE by EVENT
pophealth <- as.data.table(pophealth)

pophealth <- merge(pophealth, popev, all.x = TRUE)

pophealth$EVENT <- as.factor(pophealth$EVENT)
pophealth$EVTYPE <- as.factor(pophealth$EVTYPE)
pophealth$FATALITIES <- as.numeric(pophealth$FATALITIES)
pophealth$INJURIES <- as.numeric(pophealth$INJURIES)
```

```
economics <- mutate_each(economics, funs(toupper))
# Get and combine unique event types in the subsetted data
ecoev <- data.table(unique(economics$EVTYPE, names("EVTYPE")))
names(ecoev) <- "EVTYPE"
ecoev <- mutate(ecoev, EVENT = "")

# Identify EVENTs which correspond to multiple EVTYPEs

for(i in 1:nrow(ecoev)){
       if(grepl(".*TORN.*|.*FUNNEL.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "TORNADO"
       }
     else if(grepl(".*HURRICANE.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "HURRICANE"
     }
     else if(grepl(".*AVALAN.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "AVALANCHE"
     }
     else if(grepl(".*BLIZ.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "BLIZZARD"
     }
     else if(grepl(".*SNOW.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "HEAVY_SNOW"
     }
     else if(grepl(".*COASTAL.*FLOOD.*|.*TIDAL.*FLOOD.*|.*EROSION.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "COASTAL_FLOOD"
     }

     else if(grepl(".*FLOOD.*|.*RISING.*|.*WATER.*|.*FLD.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "FLOOD"
     }
     else if(grepl(".*COLD.*|.*CHILL.*|.*LOW TEMP.*|.*HYPOTHERM.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "COLD/WIND_CHILL"
     }
     else if(grepl(".*DROUGHT.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "DROUGHT"
     }
     else if(grepl(".*HEAT.*|.*HYPERTHERM.*|.*WARM.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "HEAT"
     }
     else if(grepl(".*COASTAL.*STORM.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "COASTAL_STORM"
    }
    else if(grepl(".*LIG.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "LIGHTNING"
    }
    else if(grepl(".*MARINE.*THUND.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "MARINE_THUNDERSTORM_WIND"
    }
    else if(grepl(".*THUND.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "THUNDERSTORM WIND"
    }
    else if(grepl(".*DRY.*|.*DUST.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "DUST_STORM"
    }
    else if(grepl(".*SLIDE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "LANDSLIDE"
    }
    else if(grepl(".*RIP.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "RIP_CURRENT"
    }
    else if(grepl(".*FREEZE.*|.*FROST.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "ICE/FREEZE"
    }
    else if(grepl(".*FREEZING.*|.*GLAZE.*|.*ICE STORM.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "ICE_STORM"
    }
    else if(grepl(".*RAIN.*|.*DOWNBURST.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "HEAVY_RAIN"
    }
    else if(grepl(".*SEA.*|.*SURF.*|.*SWELLS.*|.*WAVE.*|.*HIGH.*TIDE.*|.*SURGE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "HIGH_SURF"
    }
    else if(grepl(".*LOW.*TIDE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "LOW_TIDE"
    }
    else if(grepl(".*ROAD.*|.*ICE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "ICY_ROAD"
    }
    else if(grepl(".*MARINE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "MARINE_HIGH_WIND"
    }
    else if(grepl(".*TROPICAL.*|.*TSTM.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "TROPICAL_STORM"
    }
    else if(grepl(".*WIND.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "HIGH_WIND"
    }
    else if(grepl(".*FIRE.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "WILDFIRE"
    }
    else if(grepl(".*WINTER.*|.*WINTRY.*", ecoev$EVTYPE[i])){
           ecoev$EVENT[i] <- "WINTER_STORM"
    }
```

```
     else if(grepl(".*HAIL.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "HAIL"
     }
     else if(grepl(".*FOG.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "DENSE_FOG"
     }
     else if(grepl(".*TSU.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "TSUNAMI"
     }
     else if(grepl(".*TYPH.*", ecoev$EVTYPE[i])){
            ecoev$EVENT[i] <- "TYPHOON"
     }
     else {ecoev$EVENT[i] <- "OTHER"}
}

#Create column in subsetted data to group EVTYPE by EVENT
economics <- as.data.table(economics)

economics <- merge(economics, ecoev, all.x = TRUE)

economics$EVENT <- as.factor(economics$EVENT)
economics$EVTYPE <- as.factor(economics$EVTYPE)
economics$cost <- as.numeric(economics$cost)
```

remove(df1,economics, pophealth, i,x,y, df2, w, z, ecoev, popev, mostavgfat, mostavging, mosttotfat, mosttotinj,sumpophealth, top6avgfat, top6avginj, top6Fat, top6Inj, xt, g)

DATA is TIDY

Results

```
sumpophealth <- pophealth[, .(Total_Fatalities = sum(FATALITIES),
                          Average_Fatalities = mean(FATALITIES),
                          STD_Fat = sd(FATALITIES),
                          Total_Injuries = sum(INJURIES),
                          Average_Injuries = mean(INJURIES),
                          STD_Inj = sd(INJURIES)),
                          by = EVENT]

sumpophealth <- sumpophealth[order(-sumpophealth$Total_Injuries),]
top6Inj <- sumpophealth[1:6, "EVENT"]
setindex(pophealth, EVENT)
mosttotinj <- pophealth[.(top6Inj), on = "EVENT"]
sumpophealth <- sumpophealth[order(-sumpophealth$Average_Fatalities),]
top6avgfat <- sumpophealth[1:6, ]
setindex(pophealth, EVENT)
mostavgfat <- pophealth[.(top6avgfat), on = "EVENT"]
sumpophealth <- sumpophealth[order(-sumpophealth$Average_Injuries),]
```

```r
top6avginj <- sumpophealth[1:6, ]
setindex(pophealth, EVENT)
mostavging <- pophealth[.(top6avginj), on = "EVENT"]
sumpophealth <- sumpophealth[order(-sumpophealth$Total_Fatalities),]
top6Fat <- sumpophealth[1:6, ]
setindex(pophealth, EVENT)
mosttotfat <- pophealth[.(top6Fat), on = "EVENT" ]

xt <- xtable(sumpophealth, caption = "Total and average/incident Fatalities and Injuries - Sorted by to
print(xt, type = "html")
```

Total and average/incident Fatalities and Injuries - Sorted by total Fatalities

EVENT

Total_Fatalities

Average_Fatalities

STD_Fat

Total_Injuries

Average_Injuries

STD_Inj

1

TORNADO

5661.00

0.71

3.86

91410.00

11.52

46.26

2

HEAT

3173.00

3.36

20.06

9228.00

9.77

39.35

3

FLOOD

1554.00

1.07

1.38

8704.00

5.99

46.16

4

LIGHTNING

817.00

0.25

0.48

5232.00

1.58

2.40

5

RIP_CURRENT

577.00

0.90

0.60

529.00

0.83

2.50

6

TROPICAL_STORM

576.00

0.19

0.68

7440.00

2.48

5.61

7

COLD/WIND_CHILL

468.00

1.30

1.01

325.00

0.91

7.28

8

HIGH_WIND

426.00

0.47

0.78

1862.00

2.04

4.71

9

WINTER_STORM

278.00

0.91

1.17

1953.00

6.36

16.10

10

AVALANCHE

225.00

0.93

0.80

171.00

0.71

1.24

11

HIGH_SURF

210.00

1.12

1.52

324.00

1.72

4.98

12

THUNDERSTORM WIND

201.00

0.19

0.55

2452.00

2.33

4.79

13

HEAVY_SNOW

169.00

0.74

1.33

1165.00

5.11

14.90

14

HURRICANE

135.00

1.96

2.86

1328.00

19.25

101.09

15

ICE_STORM

106.00

0.85

1.15

2244.00

18.10

140.69

16

BLIZZARD

101.00

1.17

1.42

805.00

9.36

44.59

17

HEAVY_RAIN

101.00

0.81

1.81

280.00

2.24

4.53

18

WILDFIRE

90.00

0.27

1.22

1608.00

4.83

11.44

19

DENSE_FOG

80.00

0.63

1.26

1076.00

8.54

12.90

20

LANDSLIDE

44.00

1.83

3.37

55.00

2.29

2.97

21

TSUNAMI

33.00

16.50

21.92

129.00

64.50

91.22

22

MARINE_HIGH_WIND

32.00

1.23

1.39

38.00

1.46

2.02

23

DUST_STORM

27.00

0.36

1.26

512.00

6.83

8.43

24

HAIL

15.00

0.05

0.31

1371.00

4.71

11.20

25

ICY_ROAD

13.00

0.46

0.69

193.00

6.89

14.34

26

MARINE_THUNDERSTORM_WIND

10.00

0.91

0.70

26.00

2.36

3.70

27

DROUGHT

6.00

2.00

2.00

19.00

6.33

7.77

28

COASTAL_FLOOD

6.00

0.75

0.89

8.00

1.00

1.69

29

OTHER

5.00

0.50

0.71

31.00

3.10

4.63

30

COASTAL_STORM

4.00

0.80

0.45

2.00

0.40

0.55

31

ICE/FREEZE

2.00

1.00

0.00

3.00

1.50

2.12

32

TYPHOON

0.00

0.00

0.00

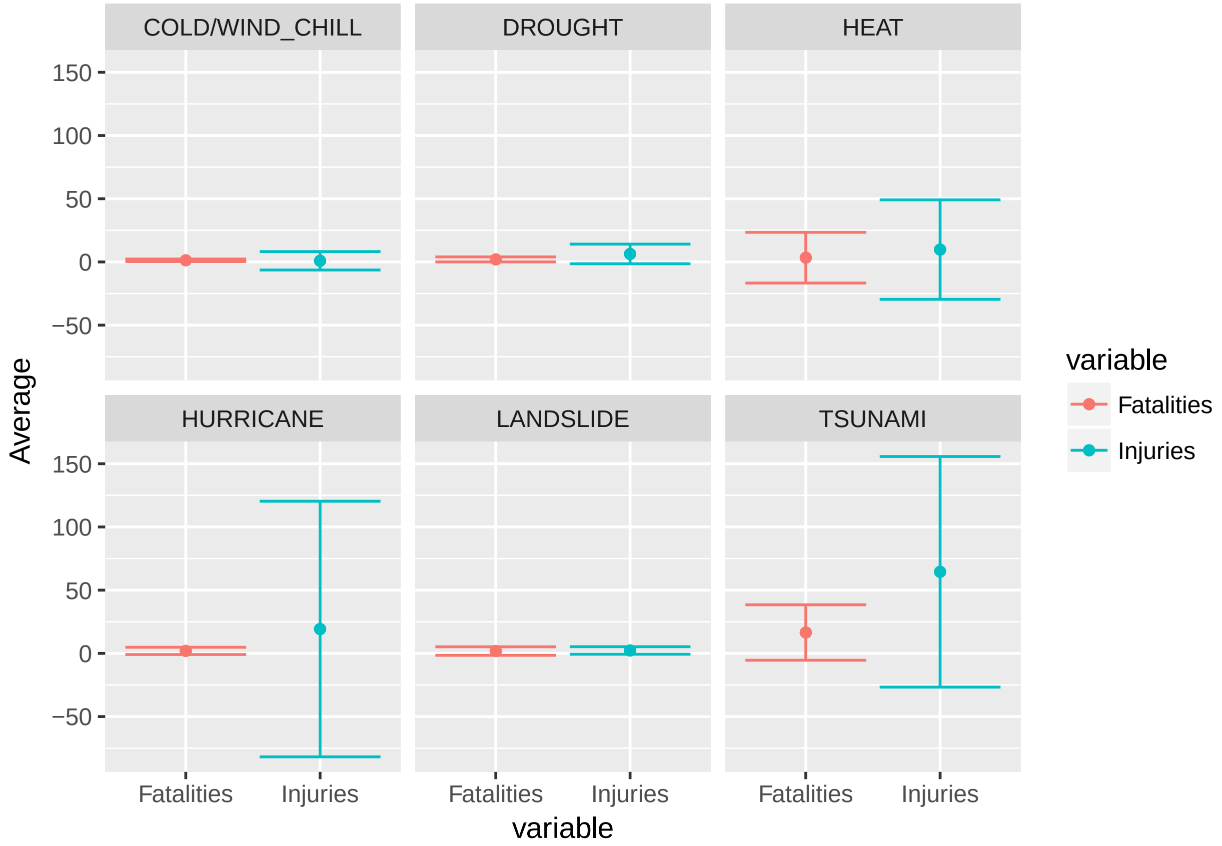
5.00

2.50

0.71

```
top6avgfat.m1 = melt(top6avgfat, id.vars = "EVENT", measure = patterns("^Total_", "^Average_", "^STD_")
top6avgfat.m1 <- top6avgfat.m1[variable == 1, variable := "Fatalities"]
top6avgfat.m1 <- top6avgfat.m1[variable == 2, variable := "Injuries"]


g <- ggplot(top6avgfat.m1, aes(variable, Average, ymin = Average - STD, ymax = Average + STD, color = va
g
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
sumeconomics <- economics[, .(Total_Cost = sum(cost),
                              Average_Cost = mean(cost),
                              STD_Cost = sd(cost)),
                              by = EVENT]


sumeconomics <- sumeconomics[order(-sumeconomics$Average_Cost),]
top6avgcost <- sumeconomics[1:6, ]
setindex(economics, EVENT)
mostavgcost <- economics[.(top6avgcost), on = "EVENT"]

sumeconomics <- sumeconomics[order(-sumeconomics$Total_Cost),]
top6cost <- sumeconomics[1:6, "EVENT"]
setindex(economics, EVENT)
mosttotcost <- economics[.(top6Inj), on = "EVENT"]


xt <- xtable(sumeconomics, caption = "Total and average/incident Cost - Sorted by total Cost")
print(xt, type = "html")
```

Total and average/incident Cost - Sorted by total Cost

EVENT

Total_Cost

Average_Cost

STD_Cost

1

DROUGHT

27945243560.00

101250882.46

232915098.55

2

FLOOD

24777042400.00

761737.71

56439949.14

3

HURRICANE

11030585600.00

51786786.85

244467995.44

4

ICE_STORM

10044228600.00

13815995.32

370878191.01

5

HAIL

6093674946.00

234408.18

2527739.42

6

ICE/FREEZE

3994122000.00

25768529.03

78591033.73

7

COLD/WIND_CHILL

2867531000.00

17810751.55

103104532.32

8

TROPICAL_STORM

2627199200.00

42090.41

2097840.48

9

HEAT

1808847000.00

32300839.29

167960782.79

10

HEAVY_RAIN

1612325600.00

1506846.36

17971350.83

11

HIGH_WIND

1510741100.00

160137.92

4507178.57

12

THUNDERSTORM WIND

1305840776.00

23469.88

996754.70

13

TORNADO

834923040.00

21190.40

1032394.79

14

WILDFIRE

806563260.00

724674.99

9185085.91

15

HEAVY_SNOW

269366200.00

150905.43

3900830.79

16

BLIZZARD

224120000.00

1028073.39

9616038.49

17

WINTER_STORM

84888000.00

44938.06

836393.41

18

LANDSLIDE

40034000.00

198188.12

2814379.29

19

LIGHTNING

24194180.00

2326.81

100299.67

20

DUST_STORM

7230000.00

30765.96

247360.34

21

HIGH_SURF

1710000.00

5104.48

82646.56

22

TYPHOON

1650000.00

183333.33

387298.33

23

COASTAL_FLOOD

112000.00

468.62

6511.63

24

MARINE_THUNDERSTORM_WIND

100000.00

4166.67

20412.41

25

TSUNAMI

40000.00

2857.14

10690.45

26

LOW_TIDE

0.00

0.00

0.00

27

AVALANCHE

0.00

0.00

0.00

28

COASTAL_STORM

0.00

0.00

29

DENSE_FOG

0.00

0.00

0.00

30

ICY_ROAD

0.00

0.00

0.00

31

MARINE_HIGH_WIND

0.00

0.00

0.00

32

RIP_CURRENT

0.00

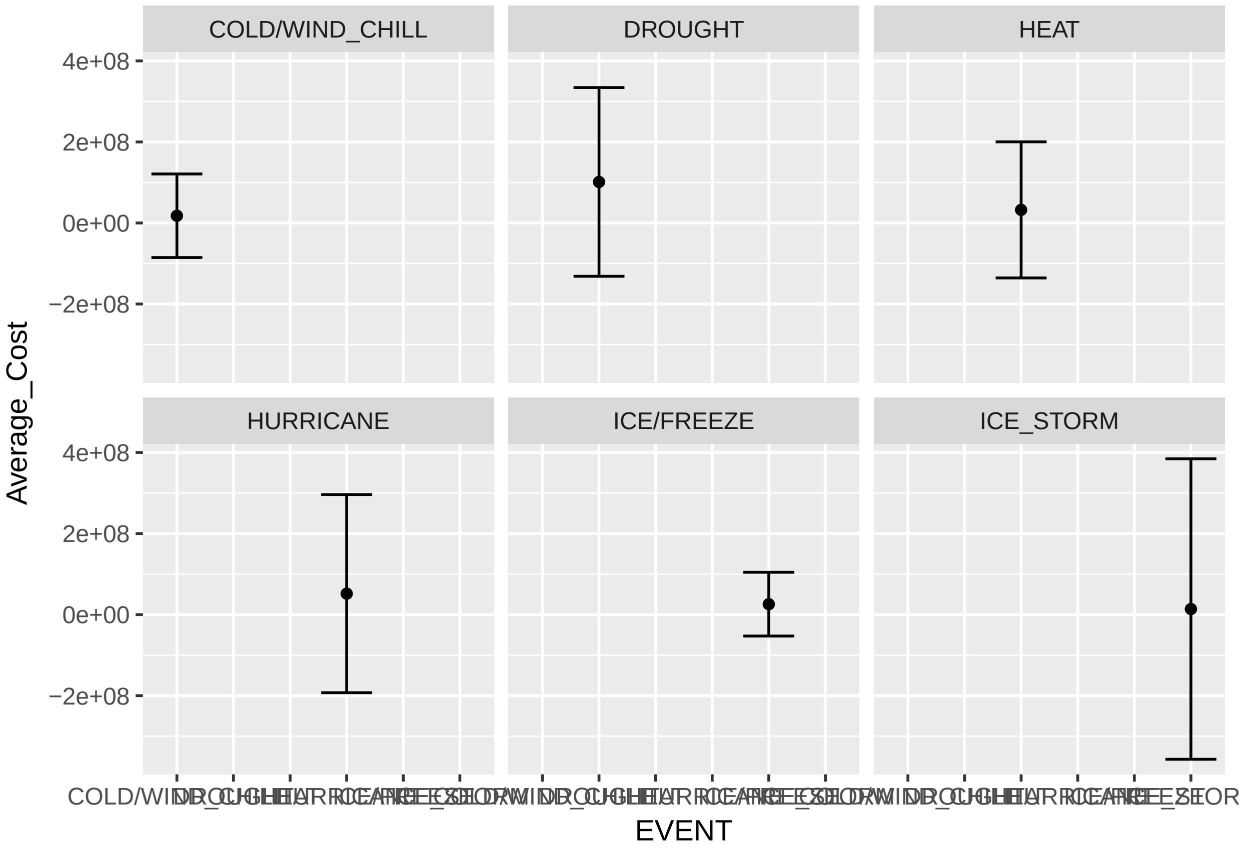
0.00

0.00

33

OTHER

```
g <- ggplot(top6avgcost, aes(EVENT, Average_Cost, ymin = Average_Cost - STD_Cost, ymax = Average_Cost +
g
```


COLD/WIND_CHILL
DROUGHT
HEAT
HURRICANE
ICE/FREEZE
ICE_STORM
4e+08
2e+08
0e+00
−2e+08
Average_Cost
EVENT