Reproducible Research - Assignment 2 - NOAA storm database analysis

Synopsis

This document looks at the National Oceanic and Atmospheric Administration's (NOAA) Storm Data, collected from 1950 to 2011. The analysis calculates the storm event types that cause the most human damage (as measured by fatalities and injuries) and the storm event types that cause the most economic damage (as measured by crop and property damage).

Human damage: 'TORNADO' are by far the leading cause of death and injury, followed by 'HEAT' with less than 1/8th the fatalities and injuries, then 'THUNDERSTORM' and 'FLOOD'

Economic damage: 'FLOOD' causes by far the largest amount of economic damage, followed by 'HURRICANE' with about half as much, then 'TORNADO', 'STORM SURGE' and 'HAIL'

Data Processing

Note: Before running the code the Working directory should be set to the location of "repdata_data_StormData.csv.bz2"

```r
#Import data from current directory, removing leading and trailing spaces from all values
storm_data <- read.csv(bzfile("repdata_data_StormData.csv.bz2"), strip.white=TRUE)
```

Import the data (with caching = True to avoid unnessecessary lengthy repeated imports)

```r
#Make all EVTYPE values upper case
storm_data$EVTYPE <- toupper(storm_data$EVTYPE)

#Remove excess spaces
storm_data$EVTYPE <- gsub("^ *|(?<= ) | *$", "", storm_data$EVTYPE, perl=T)
```

Clean up data in EVTYPE field

Simplify crop and property costs Crop and property costs are currently each split across two fields - a number and a indicated multiplier (h = hundred, k = thousand, m = million, b = billion) Below I create new columns which hold the crop and property damage cost multiplied out by the appropriate multiplier The multiplier column has some messy and a lot of missing data. Where data is not valid I assume the multiplier is 1. I have included some calculations showing the huge number of invalid multiplier values

```r
#convert multiplier column to values for both CROP and PROP
#take copy of multiplier column for conversion to value
storm_data$CROPDMGEXP_as_num <- toupper(as.character(storm_data$CROPDMGEXP))
storm_data$PROPDMGEXP_as_num <- toupper(as.character(storm_data$PROPDMGEXP))

#create dataframe of valid multipliers and their values
```

```
multiplier_value <- data.frame(code = c("H", "K", "M", "B"), value = c(100, 1000, 1000000, 1000000000))

#if multiplier is not valid then use multiplier of 1
storm_data$CROPDMGEXP_as_num [!(storm_data$CROPDMGEXP_as_num %in% multiplier_value$code)] <- 1
storm_data$PROPDMGEXP_as_num [!(storm_data$PROPDMGEXP_as_num %in% multiplier_value$code)] <- 1

#Allocate valid multipliers their true value (e.g K = 1000)
storm_data$CROPDMGEXP_as_num [(storm_data$CROPDMGEXP_as_num %in% multiplier_value$code)] <- multiplier_v

storm_data$PROPDMGEXP_as_num [(storm_data$PROPDMGEXP_as_num %in% multiplier_value$code)] <- multiplier_v

#Now that characters have been replaced by numbers make column numeric in preparation for calculation
storm_data$CROPDMGEXP_as_num <- as.numeric(storm_data$CROPDMGEXP_as_num)
storm_data$PROPDMGEXP_as_num <- as.numeric(storm_data$PROPDMGEXP_as_num)

#NOTE - The calculations below show the extent of the problem of missing or invalid multiplier values -
#Invalid CROP multiplier
invalid_crop_mult <- length(storm_data$CROPDMGEXP[storm_data$CROPDMGEXP_as_num==1])
invalid_crop_mult
```

```
## [1] 618440
```

```
#Valid CROP multiplier
valid_crop_mult <- length(storm_data$CROPDMGEXP[!storm_data$CROPDMGEXP_as_num==1])
valid_crop_mult
```

```
## [1] 283857
```

```
#Invalid PROP multiplier
invalid_prop_mult <- length(storm_data$PROPDMGEXP[storm_data$PROPDMGEXP_as_num==1])
invalid_prop_mult
```

```
## [1] 466248
```

```
#Valid PROP multiplier
valid_prop_mult <- length(storm_data$PROPDMGEXP[!storm_data$PROPDMGEXP_as_num==1])
valid_prop_mult
```

```
## [1] 436049
```

```
#Percentage of CROP multipliers that are valid
valid_crop_mult / (valid_crop_mult + invalid_crop_mult) *100
```

```
## [1] 31.46
```

```
#Percentage of CROP multipliers that are valid
valid_prop_mult / (valid_prop_mult + invalid_prop_mult) *100
```

```
## [1] 48.33
```

```r
#Multiply value * multiplier into new column
storm_data$CROPDMG_cost <- storm_data$CROPDMG * storm_data$CROPDMGEXP_as_num
storm_data$PROPDMG_cost <- storm_data$PROPDMG * storm_data$PROPDMGEXP_as_num
```

```r
event_type_summary <- aggregate(storm_data[,c("FATALITIES", "INJURIES", "CROPDMG_cost", "PROPDMG_cost")]

#Name first column appropriately
names(event_type_summary)[1]<-"EVTYPE"

#Create total columns
event_type_summary$total_human_cost <- event_type_summary$FATALITIES + event_type_summary$INJURIES
event_type_summary$total_cost <- event_type_summary$CROPDMG_cost + event_type_summary$PROPDMG_cost
```

Create summary table of event type (EVTYPE variable), summing the total fatalities, injuries, property cost and crop cost

Results

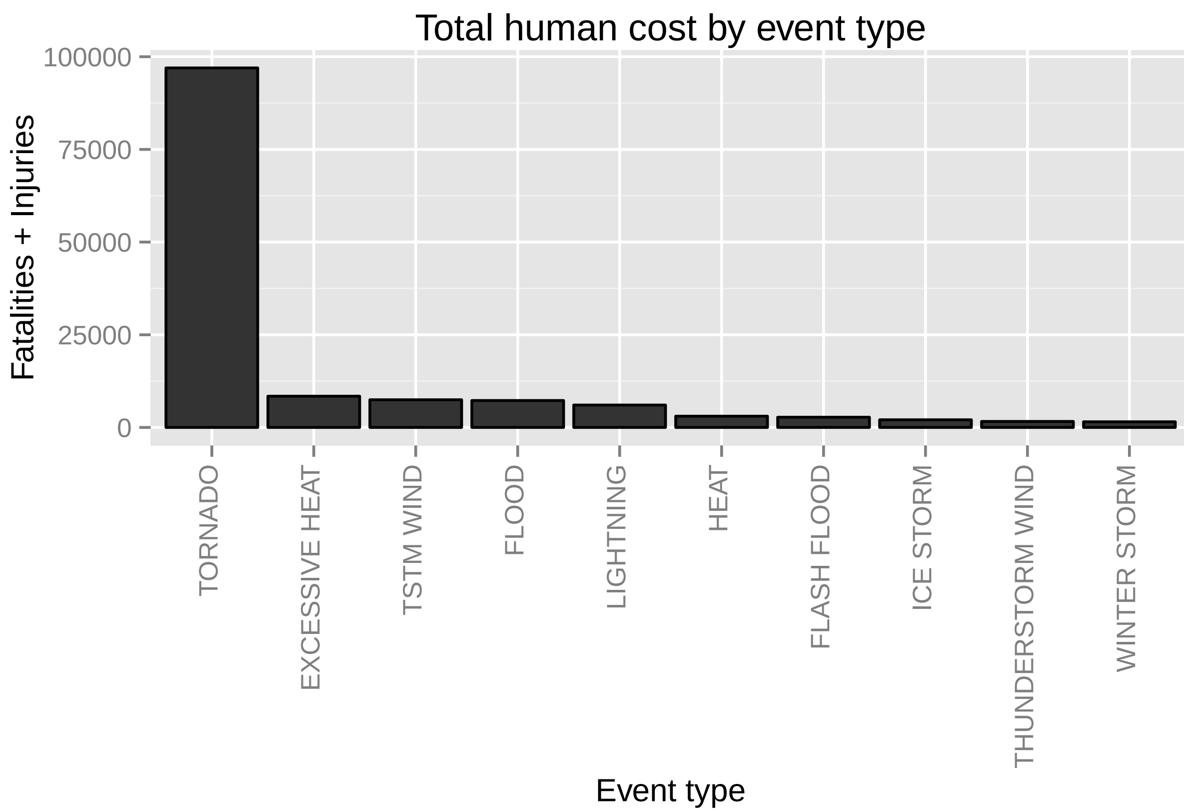
Across the United States, which types of events (as indicated in the EVTYPE variable) are most harmful with respect to population health? This question is easily answered using the "event_type_summary" dataframe. Although EVTYPE data is messy and inconsistently entered there are a number of clear major causes of death and injury. As can be seen in the list of top twenty causes of death and injury below 'TORNADOS' are the clear major cause, with 'EXCESSIVE HEAT', 'TSTM WIND', 'FLOOD' and 'LIGHTNING' having large numbers too. However you can also see the data is messy, with many events being roughly or exactly the same (e.g. 'TSTM Wind', 'THUNDERSTORM WIND', 'HIGH WIND', 'THUNDERSTORM WINDS').

```r
#Return events with highest total human cost (fatalities + injuries)
ordered_human_cost <- event_type_summary[c("EVTYPE", "FATALITIES", "INJURIES", "total_human_cost")] [or

#Reorder factors before plotting
ordered_human_cost$EVTYPE <- factor(ordered_human_cost$EVTYPE, as.character(ordered_human_cost$EVTYPE))

#Print the top 20 events that cause death and injury
ordered_human_cost[1:20,]
```

```
##                EVTYPE FATALITIES INJURIES total_human_cost
## 745           TORNADO       5633    91346            96979
## 107    EXCESSIVE HEAT       1903     6525             8428
## 766         TSTM WIND        504     6957             7461
## 145             FLOOD        470     6789             7259
## 407         LIGHTNING        816     5230             6046
## 234              HEAT        937     2100             3037
## 129       FLASH FLOOD        978     1777             2755
## 376         ICE STORM         89     1975             2064
## 672 THUNDERSTORM WIND        133     1488             1621
## 873      WINTER STORM        206     1321             1527
## 309         HIGH WIND        248     1137             1385
## 203              HAIL         15     1361             1376
```

```
## 361  HURRICANE/TYPHOON         64      1275              1339
## 265          HEAVY SNOW        127      1021              1148
## 860            WILDFIRE         75       911               986
## 698 THUNDERSTORM WINDS         64       918               982
## 20             BLIZZARD        101       805               906
## 162                 FOG         62       734               796
## 512         RIP CURRENT        368       232               600
## 858   WILD/FOREST FIRE         12       545               557
```

```
#and plot the top ten
ordered_human_cost_top <- ordered_human_cost[1:10,]
library(ggplot2)
ggplot(data=ordered_human_cost_top, aes(x=EVTYPE, y=total_human_cost)) +
  geom_bar(colour="black", stat="identity") +
  xlab ("Event type") + ylab ("Fatalities + Injuries") +
  theme(axis.text.x=element_text(angle = 90, hjust = 1, vjust = 0.3)) +
  ggtitle ("Total human cost by event type")
```



To clean up the EVTYPE data a bit I combined a number of categories if they cotained certain keywords (e.g. combining all EVTYPEs that contain the text 'TORNADO'). Note that the order of search and replace matters as there is some overlap between categories (e.g. 'THUNDERSTORM WINDS LIGHTNING'). I have ordered the keywords from most specific to least specific to split these overlaps in the most appropriate way.

```r
#Create copy of EVTYPE field
event_type_summary$EVTYPE_clean <- as.character(event_type_summary$EVTYPE)

keyword_replace <- data.frame (keyword = c("TORNADO", "HURRICANE", "TYPHOON", "LIGHTNING", "FLOOD", "HA

#Show keyword/replace table. An EVTYPE which contins the keyword anywhere within it will be replaed wit
keyword_replace
```

```
##        keyword     replace
## 1      TORNADO     TORNADO
## 2    HURRICANE   HURRICANE
## 3      TYPHOON   HURRICANE
## 4    LIGHTNING   LIGHTNING
## 5        FLOOD       FLOOD
## 6         HAIL        HAIL
## 7    ICE STORM        HAIL
## 8          FOG         FOG
## 9     BLIZZARD    BLIZZARD
## 10 WINTER STORM    BLIZZARD
## 11  RIP CURRENT  RIP CURRENT
## 12        FIRE        FIRE
## 13     THUNDER THUNDERSTORM
## 14        TSTM THUNDERSTORM
## 15        WIND        WIND
## 16        HEAT        HEAT
```

```r
#Search for keywords and replace value in EVTYPE_clean column
for (keyword in keyword_replace$keyword)
{
  event_type_summary$EVTYPE_clean[grep(keyword, event_type_summary$EVTYPE_clean)] <- as.character(keywo
}

#convert EVTYPE_clean back to factor
event_type_summary$EVTYPE_clean <- as.factor(event_type_summary$EVTYPE_clean)
```

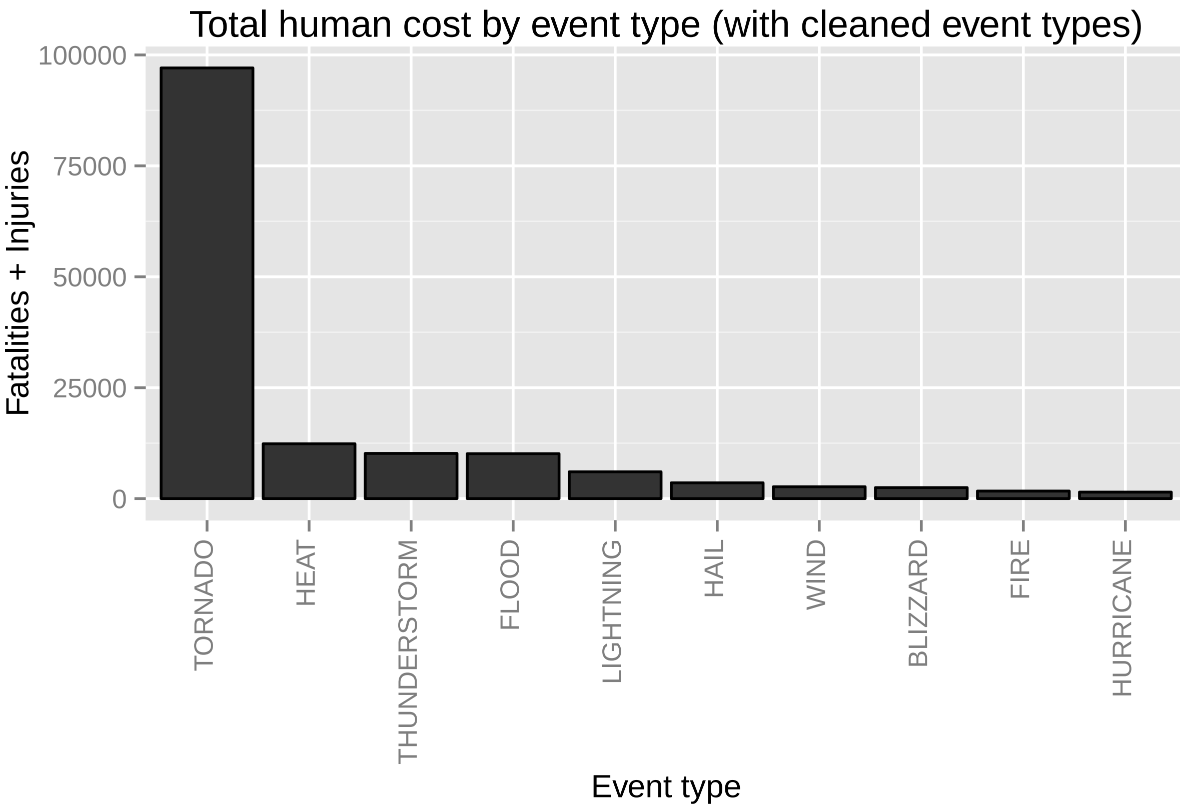
Then we create a new EVTYPE summary table (incluing damage costs for analysis in second question) and rerun the human cost analysis

```r
#Create event type summary based on cleaned EVTYPE data
event_type_summary_clean <- aggregate(event_type_summary[c("FATALITIES", "INJURIES", "total_human_cost"

#Name EVTYPE column
names(event_type_summary_clean)[1]<-"EVTYPE"

#Return events with highest total human cost (fatalities + injuries)
ordered_human_cost <- event_type_summary_clean[c("EVTYPE", "FATALITIES", "INJURIES", "total_human_cost")

#Print the top 20 events that cause death and injury
ordered_human_cost[1:20,]
```

```
##             EVTYPE FATALITIES INJURIES total_human_cost
## 388        TORNADO       5661    91407            97068
```

```
## 114           HEAT       3138     9224            12362
## 387   THUNDERSTORM        726     9448            10174
## 86           FLOOD       1525     8604            10129
## 199      LIGHTNING        817     5232             6049
## 111           HAIL        109     3457             3566
## 450           WIND        694     1979             2673
## 16        BLIZZARD        318     2159             2477
## 82            FIRE         90     1608             1698
## 165      HURRICANE        135     1333             1468
## 87             FOG         81     1077             1158
## 130     HEAVY SNOW        127     1021             1148
## 268    RIP CURRENT        577      529             1106
## 62      DUST STORM         22      440              462
## 452 WINTER WEATHER         33      398              431
## 393 TROPICAL STORM         58      340              398
## 11       AVALANCHE        224      170              394
## 78    EXTREME COLD        162      231              393
## 119     HEAVY RAIN         98      251              349
## 152      HIGH SURF        104      156              260
```

```r
#and plot the top ten
#Reorder factors before plotting
ordered_human_cost$EVTYPE <- factor(ordered_human_cost$EVTYPE, as.character(ordered_human_cost$EVTYPE))

#Get top ten events
ordered_human_cost_top <- ordered_human_cost[1:10,]

ggplot(data=ordered_human_cost_top, aes(x=EVTYPE, y=total_human_cost)) +
  geom_bar(colour="black", stat="identity") +
  xlab ("Event type") + ylab ("Fatalities + Injuries") +
  theme(axis.text.x=element_text(angle = 90, hjust = 1, vjust = 0.3)) +
  ggtitle ("Total human cost by event type (with cleaned event types)")
```



As you can see in our cleaned dataset 'TORNADO' are still by far the leading cause of death and injury, followed by 'HEAT' with less than 1/8th the fatalities and injuries, then 'THUNDERSTORM' and 'FLOOD'

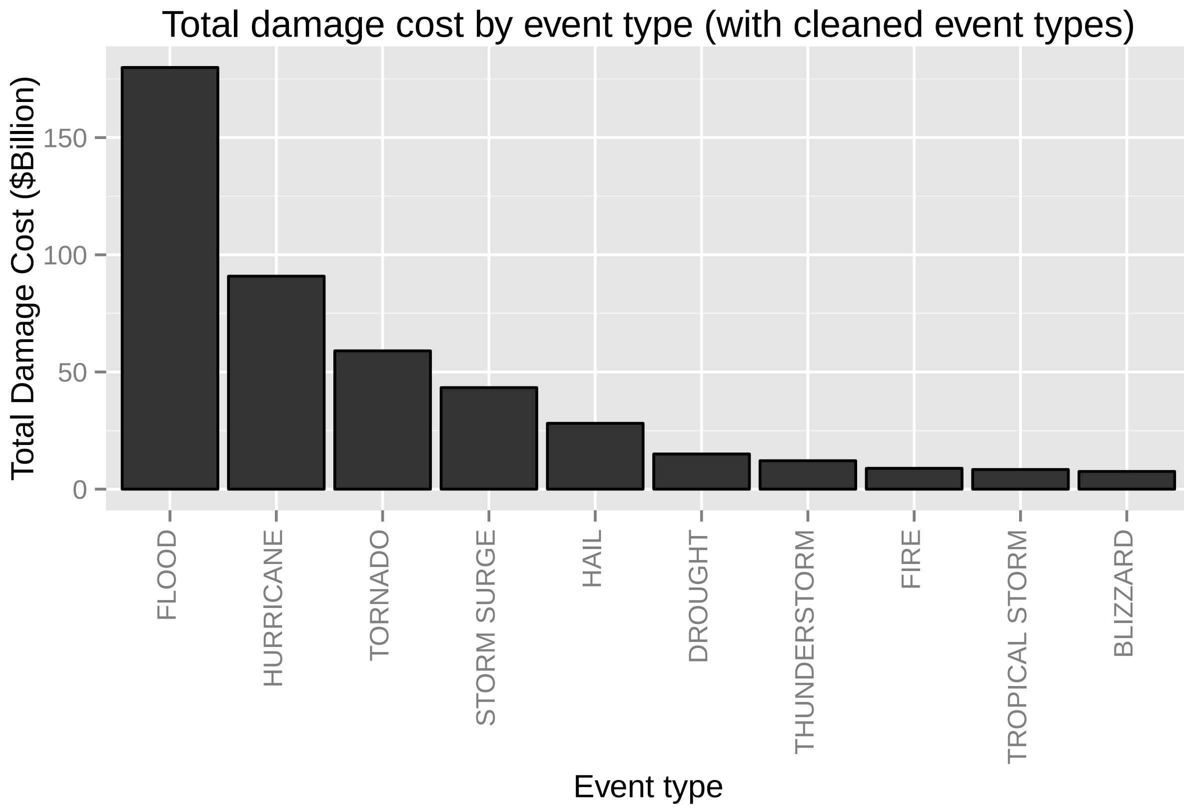
Across the United States, which types of events have the greatest economic consequences? Most of the work to answer this question has already been done. The total property damge cost and crop damage cost have been calculated, the EVTYPE varible has been cleaned up and a summary table has been created. Now it's just a matter of seeing what the table says. NOTE: as noted above when multiplying out the damage cost, many multiplier values are missing and so the data here is very incomplete. There's not much we can do about it on a simple pass though - perhaps the notes columns have useful information, but that's messy and beyond the scope of this work

```
#Return events with highest total damage cost (property damage cost + crop damage cost)
ordered_cost <- event_type_summary_clean[c("EVTYPE", "CROPDMG_cost", "PROPDMG_cost", "total_cost")] [ord

#Create copy of total cost column which gives value in Billions
ordered_cost$total_cost_billions <- ordered_cost$total_cost/1e+09

#Print the top 20 events causing crop and property damage
ordered_cost[1:20,]
```

```
##                         EVTYPE CROPDMG_cost PROPDMG_cost total_cost
## 86                       FLOOD    1.238e+10    1.675e+11  1.799e+11
## 165                  HURRICANE    5.516e+09    8.536e+10  9.087e+10
## 388                    TORNADO    4.175e+08    5.859e+10  5.901e+10
## 318                STORM SURGE    5.000e+03    4.332e+10  4.332e+10
```

```
## 111                     HAIL    8.134e+09    1.997e+10  2.810e+10
## 44                   DROUGHT    1.397e+10    1.046e+09  1.502e+10
## 387             THUNDERSTORM    1.207e+09    1.093e+10  1.214e+10
## 82                      FIRE    4.033e+08    8.497e+09  8.900e+09
## 393           TROPICAL STORM    6.783e+08    7.704e+09  8.382e+09
## 16                  BLIZZARD    1.445e+08    7.414e+09  7.559e+09
## 450                     WIND    7.755e+08    6.208e+09  6.983e+09
## 319         STORM SURGE/TIDE    8.500e+05    4.641e+09  4.642e+09
## 122 HEAVY RAIN/SEVERE WEATHER    0.000e+00    2.500e+09  2.500e+09
## 119               HEAVY RAIN    7.334e+08    6.942e+08  1.428e+09
## 78              EXTREME COLD    1.313e+09    6.774e+07  1.381e+09
## 100             FROST/FREEZE    1.094e+09    1.048e+07  1.105e+09
## 130               HEAVY SNOW    1.347e+08    9.326e+08  1.067e+09
## 199                LIGHTNING    1.210e+07    9.390e+08  9.511e+08
## 114                     HEAT    9.045e+08    2.033e+07  9.248e+08
## 88                    FREEZE    4.567e+08    2.050e+05  4.569e+08
##     total_cost_billions
## 86             179.9099
## 165             90.8725
## 388             59.0106
## 318             43.3235
## 111             28.0998
## 44              15.0187
## 387             12.1374
## 82               8.8999
## 393              8.3822
## 16               7.5589
## 450              6.9832
## 319              4.6420
## 122              2.5000
## 119              1.4276
## 78               1.3807
## 100              1.1047
## 130              1.0672
## 199              0.9511
## 114              0.9248
## 88               0.4569
```

```r
#and plot the top ten
#Reorder factors before plotting
ordered_cost$EVTYPE <- factor(ordered_cost$EVTYPE, as.character(ordered_cost$EVTYPE))

#Get top ten events
ordered_cost_top <- ordered_cost[1:10,]

ggplot(data=ordered_cost_top, aes(x=EVTYPE, y=total_cost_billions)) +
  geom_bar(colour="black", stat="identity") +
  xlab ("Event type") + ylab ("Total Damage Cost ($Billion)") +
  theme(axis.text.x=element_text(angle = 90, hjust = 1, vjust = 0.3)) +
  ggtitle ("Total damage cost by event type (with cleaned event types)")
```



As you can see 'FLOOD' creates by far the largest total damage bill, followed by 'HURRICANE' with about half as much, then 'TORNADO', 'STORM SURGE' and 'HAIL'