##Synopsis In this report we aim to explore the U.S. National Oceanic and Atmospheric Administration’s (NOAA) Storm Database and answer some basic questions about severe weather events, in particulary these two:

* Which types of events are most harmful with respect to population health?
* Which types of events have the greatest economic consequences?

The NOAA database tracks characteristics of major storms and weather events in the United States, including when and where they occur, as well as estimates of any fatalities, injuries, and property damage and the events in the database start in the year 1950.

## Library loading and setting parameters

library(knitr)  
library(ggplot2)  
library(data.table)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(R.utils)

## Loading required package: R.oo

## Loading required package: R.methodsS3

## R.methodsS3 v1.8.2 (2022-06-13 22:00:14 UTC) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.25.0 (2022-06-12 02:20:02 UTC) successfully loaded. See ?R.oo for help.

##   
## Attaching package: 'R.oo'

## The following object is masked from 'package:R.methodsS3':  
##   
## throw

## The following objects are masked from 'package:methods':  
##   
## getClasses, getMethods

## The following objects are masked from 'package:base':  
##   
## attach, detach, load, save

## R.utils v2.12.2 (2022-11-11 22:00:03 UTC) successfully loaded. See ?R.utils for help.

##   
## Attaching package: 'R.utils'

## The following object is masked from 'package:utils':  
##   
## timestamp

## The following objects are masked from 'package:base':  
##   
## cat, commandArgs, getOption, isOpen, nullfile, parse, warnings

library(stringdist)

##   
## Attaching package: 'stringdist'

## The following object is masked from 'package:R.utils':  
##   
## extract

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(grid)  
library(ggtext)  
library(ggforce)  
  
opts\_chunk$set(echo = TRUE,cache =TRUE)  
options("scipen"=5, "digits"=4)

## Load database

# Check if the file exist or downloading the file  
url = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"  
dest\_file = "storms\_data.csv.bz2"  
csv\_file = "storms\_data.csv"  
if (!file.exists(dest\_file)){  
 download.file(url, dest\_file)  
 bunzip2(dest\_file, csv\_file,remove = F)  
}  
# Read the data  
storms = fread(csv\_file,  
 sep = ",",  
 header = T,  
 showProgress = F)

storms$BGN\_DATE = as.POSIXct(storms$BGN\_DATE, format="%m/%d/%Y %H:%M:%S")  
storms = storms %>%   
 filter(BGN\_DATE >= as.POSIXct('1/1/1996', format="%m/%d/%Y"))

storms = storms %>%  
 select(COUNTY,  
 COUNTYNAME,   
 STATE,   
 BGN\_DATE,  
 EVTYPE,   
 FATALITIES,   
 INJURIES,   
 PROPDMG,   
 PROPDMGEXP,   
 CROPDMG,   
 CROPDMGEXP,   
 REFNUM)

storms[storms$EVTYPE %like% "HURRICANE"] %>%  
 select(EVTYPE) %>%  
 table()

## EVTYPE  
## HURRICANE HURRICANE/TYPHOON   
## 170 88

# Set toupper all the events  
storms = storms %>% mutate(EVTYPE = toupper(EVTYPE))  
  
events <- storms$EVTYPE  
# Replace special char with space as initial clean up  
events <- gsub("( ){1,}"," ", gsub("[^A-Z0-9 ]"," ", events))  
  
# Divide the cleaning by the events  
## FLOOD related events  
events[grepl("COASTAL|STORM SURGE", events)] <- "COASTAL FLOOD"  
events[grepl("FLASH", events)] <- "FLASH FLOOD"  
events[!grepl("FLASH|COASTAL", events) & grepl("FLOOD", events)] <- "FLOOD"  
events[grepl("STREAM|URBAN", events)] <- "FLOOD"  
  
## HEAT related events  
events[grepl("HEAT|DRY", events)] <- "EXCESSIVE HEAT"  
events[grepl("HOT|WARM", events)] <- "EXCESSIVE HEAT"  
events[grepl("RECORD (HIGH|.\*TEMP)|HIGH TEMPERA", events)] <- "EXCESSIVE HEAT"  
  
## COLD related events  
events[grepl("SLEET", events)] <- "SLEET"  
events[grepl("BLIZZARD", events)] <- "BLIZZARD"  
events[grepl("EXTREME", events) & grepl("CHILL|COLD", events)] <- "EXTREME COLD/WIND CHILL"  
events[!grepl("EXTREME", events) & grepl("CHILL|COLD", events)] <- "COLD/WIND CHILL"  
events[grepl("LAKE", events) & grepl("SNOW", events)] <- "LAKE-EFFECT SNOW"  
events[!grepl("LAKE", events) & grepl("SNOW", events)] <- "HEAVY SNOW"  
events[grepl("FROST|FREEZE", events)] <- "FROST/FREEZE"  
events[!grepl("FROST", events) & grepl("FREEZE", events)] <- "SLEET"  
events[grepl("FREEZ", events) & grepl("RAIN", events)] <- "SLEET"  
events[grepl("DRIZZLE", events)] <- "SLEET"  
events[grepl("(RECORD LOW|LOW TEMP)", events)] <- "EXTREME COLD/WIND CHILL"  
events[grepl("GLAZE", events)] <- "EXTREME COLD/WIND CHILL"  
events[grepl("ICE", events)] <- "ICE STORM"  
events[grepl("WINT", events)] <- "WINTER STORM"  
events[grepl("HAIL", events)] <- "HAIL"  
  
## WIND, RAIN and LIGHTING related events  
events <- gsub("WINDS", "WIND", events)  
events[!grepl("DERSTORM WIND", events) & grepl("THUN|TSTM", events)] <- "LIGHTNING"  
events[grepl("LIGHT|LIGN", events)] <- "LIGHTNING"  
events[grepl("DERSTORM WIND", events)] <- "THUNDERSTORM WIND"  
events[grepl("TORN", events)] <- "TORNADO"  
events[grepl("SPOUT", events)] <- "WATERSPOUT"  
events[grepl("HURRICANE|TYPHOON", events)] <- "HURRICANE (TYPHOON)"  
events[grepl("FIRE", events)] <- "WILDFIRE"  
events[!grepl("MARINE", events) & grepl("HIGH WIND", events)] <- "HIGH WIND"  
events[grepl("GUST", events)] <- "STRONG WIND"  
events[!grepl("COLD|MARINE|THUNDER|STRONG|HIGH", events) & grepl("WIND", events)] <- "STRONG WIND"  
events[grepl("FUNNEL", events)] <- "FUNNEL CLOUD"  
events[grepl("TROPICAL STORM", events)] <- "TROPICAL STORM"  
events[!grepl("FREEZIN", events) & grepl("FOG|VOG", events)] <- "DENSE FOG"  
events[grepl("WET|RAIN|SHOWER|PRECIP", events)] <- "HEAVY RAIN"  
  
## DUST related events  
events[grepl("DUST DEVEL", events)] <- "DUST DEVIL"  
events[!grepl("DEVIL", events) & grepl("DUST", events)] <- "DUST STORM"  
  
## MARINE EVENTS  
events[grepl("RIP CURRENT", events)] <- "RIP CURRENT"  
events[!grepl("LOW", events) & grepl("TIDE|WAVE|SWELL", events)] <- "STORM SURGE/TIDE"  
events[grepl("SURF", events)] <- "HIGH SURF"  
  
## MISC events  
events[grepl("VOLCAN", events)] <- "VOLCANIC ASH"  
  
# Not a storm, but is there, so we will classify it  
events[grepl("(MUD|LAND|ROCK).\*SLIDE", events)] <- "LANDSLIDE"  
  
# All the SUMMARY  
events[grepl("SUMMARY", events)] <- "OTHER/UNKOWN"  
  
# re-assign the cleaned up column values  
storms$EVTYPE <- events

symbol\_to\_num = function(symbol){  
 if(symbol=="K"){  
 num=1000  
 }  
 if(symbol=="" | symbol=="0"){  
 num=1  
 }  
 if(symbol=="M"){  
 num=1000000  
 }  
 if(symbol=="B"){  
 num=1000000000  
 }  
 num  
}  
  
# CROP exp  
storms[, CROPDMGEXPNUM:=symbol\_to\_num(CROPDMGEXP), by = 1:nrow(storms)]  
# PROP exp  
storms[, PROPDMGEXPNUM:=symbol\_to\_num(PROPDMGEXP), by = 1:nrow(storms)]

# CROP  
storms[, CROPDMG:=CROPDMG\*CROPDMGEXPNUM ]  
#PROP  
storms[, PROPDMG:=PROPDMG\*PROPDMGEXPNUM ]

storms

## COUNTY COUNTYNAME STATE BGN\_DATE EVTYPE FATALITIES INJURIES  
## 1: 1 ALZ001>038 AL 1996-01-06 WINTER STORM 0 0  
## 2: 31 COFFEE AL 1996-01-11 TORNADO 0 0  
## 3: 31 COFFEE AL 1996-01-11 LIGHTNING 0 0  
## 4: 45 DALE AL 1996-01-11 LIGHTNING 0 0  
## 5: 67 HENRY AL 1996-01-11 LIGHTNING 0 0  
## ---   
## 653526: 7 WYZ007 - 017 WY 2011-11-30 HIGH WIND 0 0  
## 653527: 9 MTZ009 - 010 MT 2011-11-10 HIGH WIND 0 0  
## 653528: 213 AKZ213 AK 2011-11-08 HIGH WIND 0 0  
## 653529: 202 AKZ202 AK 2011-11-09 BLIZZARD 0 0  
## 653530: 6 ALZ006 AL 2011-11-28 HEAVY SNOW 0 0  
## PROPDMG PROPDMGEXP CROPDMG CROPDMGEXP REFNUM CROPDMGEXPNUM  
## 1: 380000 K 38000 K 248768 1000  
## 2: 100000 K 0 248769 1  
## 3: 3000 K 0 248770 1  
## 4: 5000 K 0 248771 1  
## 5: 2000 K 0 248772 1  
## ---   
## 653526: 0 K 0 K 902293 1000  
## 653527: 0 K 0 K 902294 1000  
## 653528: 0 K 0 K 902295 1000  
## 653529: 0 K 0 K 902296 1000  
## 653530: 0 K 0 K 902297 1000  
## PROPDMGEXPNUM  
## 1: 1000  
## 2: 1000  
## 3: 1000  
## 4: 1000  
## 5: 1000  
## ---   
## 653526: 1000  
## 653527: 1000  
## 653528: 1000  
## 653529: 1000  
## 653530: 1000

sum\_storms\_injuries = storms %>%  
 group\_by(EVTYPE) %>%  
 summarise(TOTINJURIES = sum(INJURIES),  
 TOTFATALITIES = sum(FATALITIES)) %>%  
 arrange(-TOTINJURIES, -TOTFATALITIES) %>%  
 head(10)  
sum\_storms\_injuries

## # A tibble: 10 × 3  
## EVTYPE TOTINJURIES TOTFATALITIES  
## <chr> <dbl> <dbl>  
## 1 TORNADO 20667 1511  
## 2 LIGHTNING 7784 903  
## 3 EXCESSIVE HEAT 7727 2039  
## 4 FLOOD 6839 444  
## 5 WINTER STORM 1852 253  
## 6 FLASH FLOOD 1674 887  
## 7 WILDFIRE 1458 87  
## 8 THUNDERSTORM WIND 1426 141  
## 9 HURRICANE (TYPHOON) 1328 125  
## 10 HIGH WIND 1083 235

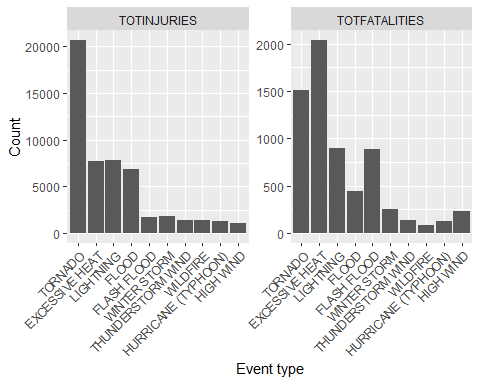
storms\_melt\_injuries = melt(as.data.table(sum\_storms\_injuries))

## Warning in melt.data.table(as.data.table(sum\_storms\_injuries)): id.vars  
## and measure.vars are internally guessed when both are 'NULL'. All  
## non-numeric/integer/logical type columns are considered id.vars, which in this  
## case are columns [EVTYPE]. Consider providing at least one of 'id' or 'measure'  
## vars in future.

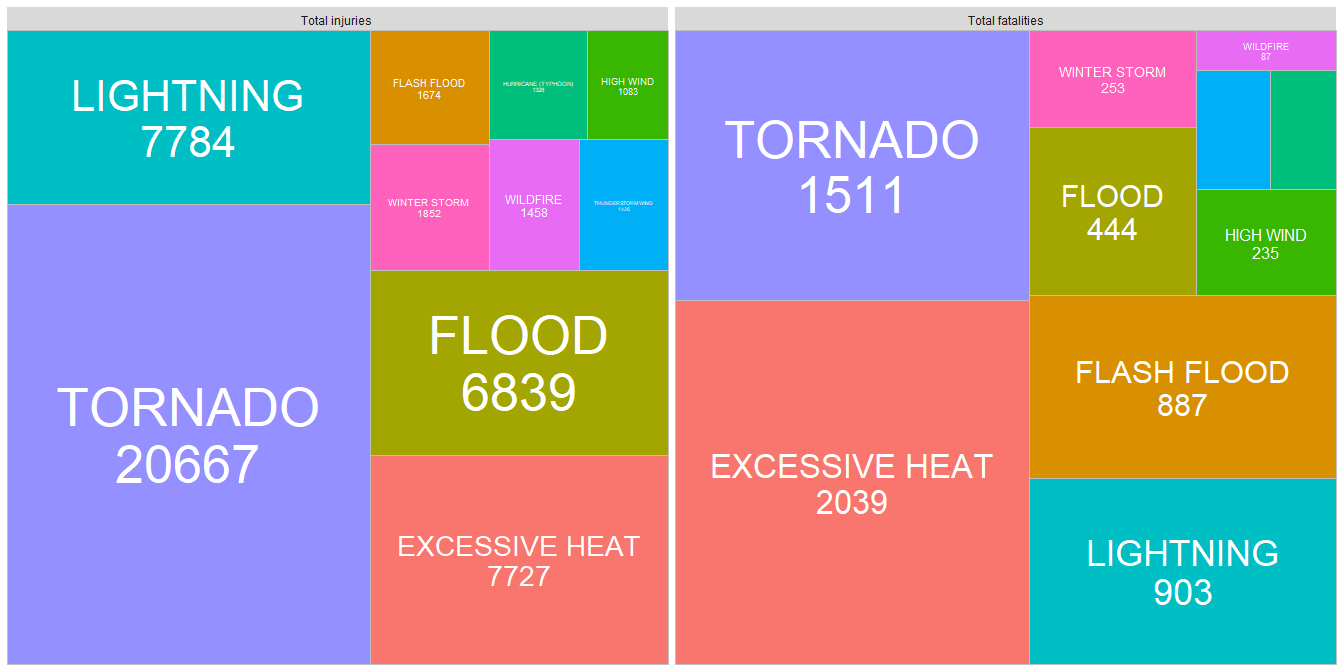
storms\_melt\_injuries

## EVTYPE variable value  
## 1: TORNADO TOTINJURIES 20667  
## 2: LIGHTNING TOTINJURIES 7784  
## 3: EXCESSIVE HEAT TOTINJURIES 7727  
## 4: FLOOD TOTINJURIES 6839  
## 5: WINTER STORM TOTINJURIES 1852  
## 6: FLASH FLOOD TOTINJURIES 1674  
## 7: WILDFIRE TOTINJURIES 1458  
## 8: THUNDERSTORM WIND TOTINJURIES 1426  
## 9: HURRICANE (TYPHOON) TOTINJURIES 1328  
## 10: HIGH WIND TOTINJURIES 1083  
## 11: TORNADO TOTFATALITIES 1511  
## 12: LIGHTNING TOTFATALITIES 903  
## 13: EXCESSIVE HEAT TOTFATALITIES 2039  
## 14: FLOOD TOTFATALITIES 444  
## 15: WINTER STORM TOTFATALITIES 253  
## 16: FLASH FLOOD TOTFATALITIES 887  
## 17: WILDFIRE TOTFATALITIES 87  
## 18: THUNDERSTORM WIND TOTFATALITIES 141  
## 19: HURRICANE (TYPHOON) TOTFATALITIES 125  
## 20: HIGH WIND TOTFATALITIES 235

ggplot(storms\_melt\_injuries, aes(reorder(EVTYPE, -value), value)) +  
 geom\_bar(stat = "identity") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 xlab("Event type") +  
 ylab("Count") +  
 facet\_wrap( ~ variable,  
 scales = "free")



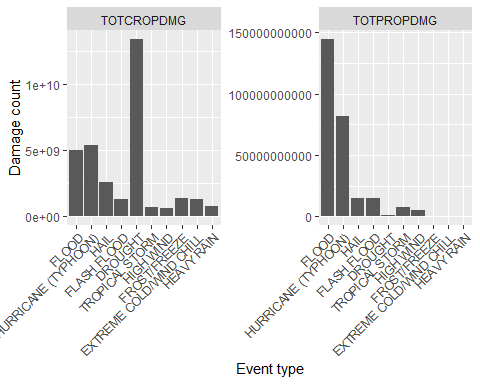
options(repr.plot.width=18, repr.plot.height=10)  
library(treemapify)  
variable.labs <- c("Total injuries", "Total fatalities")  
names(variable.labs) <- c("TOTINJURIES", "TOTFATALITIES")  
ggplot(storms\_melt\_injuries, aes(area = value, fill=EVTYPE, label=paste(EVTYPE, value, sep="\n")), leg) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 1, grow = TRUE) +  
 theme(legend.position = "none") +  
 facet\_grid(~variable,  
 scales = "free",   
 labeller = labeller(variable = variable.labs))



dmg\_storms\_crop = storms %>%  
 group\_by(EVTYPE) %>%  
 summarise(TOTCROPDMG = sum(CROPDMG),  
 TOTPROPDMG = sum(PROPDMG)) %>%  
 arrange(-TOTCROPDMG, -TOTPROPDMG) %>%  
 head(10)  
  
# Plot  
storms\_melt\_crop = melt(as.data.table(dmg\_storms\_crop))

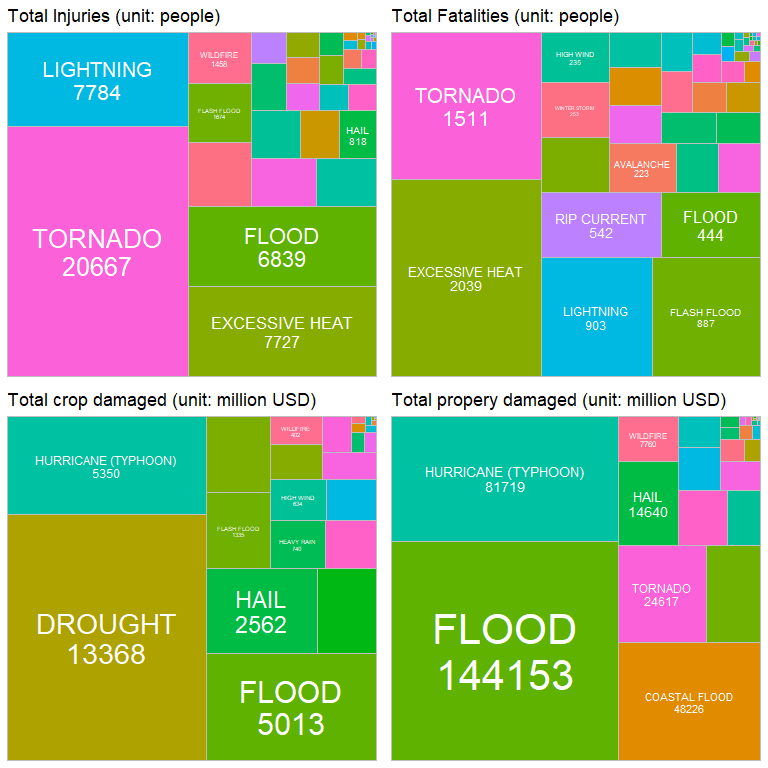
## Warning in melt.data.table(as.data.table(dmg\_storms\_crop)): id.vars  
## and measure.vars are internally guessed when both are 'NULL'. All  
## non-numeric/integer/logical type columns are considered id.vars, which in this  
## case are columns [EVTYPE]. Consider providing at least one of 'id' or 'measure'  
## vars in future.

ggplot(storms\_melt\_crop, aes(reorder(EVTYPE, -value), value)) +  
 geom\_bar(stat = "identity") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 xlab("Event type") +  
 ylab("Damage count") +  
 facet\_wrap( ~ variable,  
 scales = "free")



sum\_storms = storms %>%  
 group\_by(EVTYPE) %>%  
 summarise(TOTINJURIES = sum(INJURIES),  
 TOTFATALITIES = sum(FATALITIES),  
 TOTCROPDMG = round(sum(CROPDMG)/1000000),  
 TOTPROPDMG = round(sum(PROPDMG)/1000000))

# Plot  
  
variable.labs <- c("Total injuries (unit: people)", "Total fatalities (unit: people)", "Total crop damaged", "Total propery damaged")  
names(variable.labs) <- c("TOTINJURIES", "TOTFATALITIES", "TOTCROPDMG", "TOTPROPDMG")  
  
p1 = ggplot(sum\_storms, aes(area = TOTINJURIES, fill=EVTYPE, label=paste(EVTYPE, TOTINJURIES, sep="\n"))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 1, grow = TRUE) +  
   
 theme(legend.position = "none") +   
 ggtitle("Total Injuries (unit: people)")  
   
  
p2 = ggplot(sum\_storms, aes(area = TOTFATALITIES, fill=EVTYPE, label=paste(EVTYPE, TOTFATALITIES, sep="\n"))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 1, grow = TRUE) +  
 theme(legend.position = "none") +   
 ggtitle("Total Fatalities (unit: people)")   
   
  
p3 = ggplot(sum\_storms, aes(area = TOTCROPDMG, fill=EVTYPE, label=paste(EVTYPE, TOTCROPDMG, sep="\n"))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 1, grow = TRUE) +  
 theme(legend.position = "none") +   
 ggtitle("Total crop damaged (unit: million USD)")  
  
p4 = ggplot(sum\_storms, aes(area = TOTPROPDMG, fill=EVTYPE, label=paste(EVTYPE, TOTPROPDMG, sep="\n"))) +  
 geom\_treemap() +  
 geom\_treemap\_text(colour = "white", place = "centre",  
 size = 1, grow = TRUE) +  
 theme(legend.position = "none") +  
 ggtitle("Total propery damaged (unit: million USD)")  
  
grid.arrange(p1, p2, p3, p4, nrow=2)

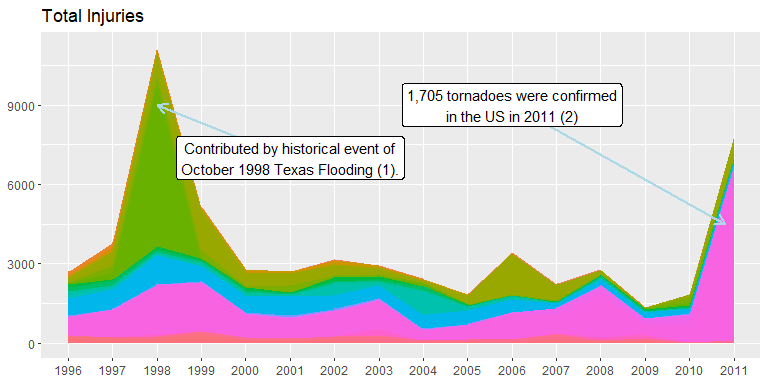


sum\_yearly\_storms = storms %>%  
 group\_by(EVTYPE, YEAR = format(BGN\_DATE, "%Y")) %>%  
 summarise(TOTINJURIES = sum(INJURIES),  
 TOTFATALITIES = sum(FATALITIES),  
 TOTCROPDMG = round(sum(CROPDMG)/1000000),  
 TOTPROPDMG = round(sum(PROPDMG)/1000000))

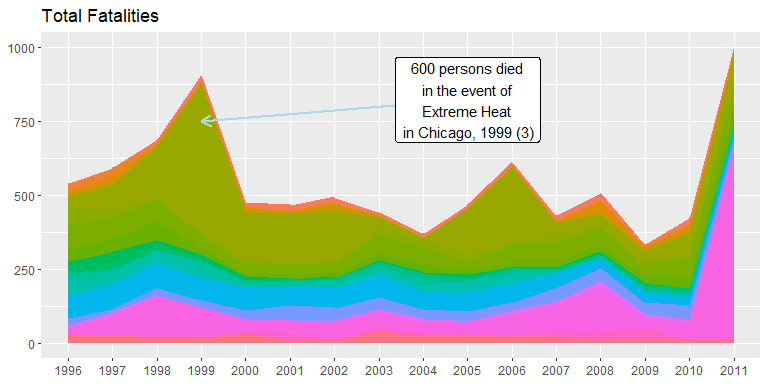
## `summarise()` has grouped output by 'EVTYPE'. You can override using the  
## `.groups` argument.

segment\_data = data.frame(x = c(6, 11), xend=c(3, 15.8), y=c(7000, 9000), yend=c(9000, 4500))  
annotate\_point\_x = c(6, 11)  
annotate\_point\_y = c(7000, 9000)  
annotate\_label = c("Contributed by historical event of\nOctober 1998 Texas Flooding (1).", "1,705 tornadoes were confirmed\nin the US in 2011 (2)")  
  
ggplot(sum\_yearly\_storms, aes(x = YEAR, y=TOTINJURIES, group=EVTYPE, fill=EVTYPE)) +   
 geom\_area() +  
 theme(legend.position = "none", axis.title.y=element\_blank(), axis.title.x=element\_blank()) +   
 ggtitle("Total Injuries") +   
 geom\_segment(  
 data=segment\_data,  
 aes(x = x, y = y, xend = xend, yend = yend),  
 inherit.aes = F,  
 lineend = "round", # See available arrow types in example above  
 linejoin = "round",  
 size = 1,   
 arrow = arrow(length = unit(0.3, "cm")),  
 colour = "lightblue") + # Also accepts "red", "blue' etc   
   
 annotate("label", x=annotate\_point\_x, y=annotate\_point\_y, label=annotate\_label, )

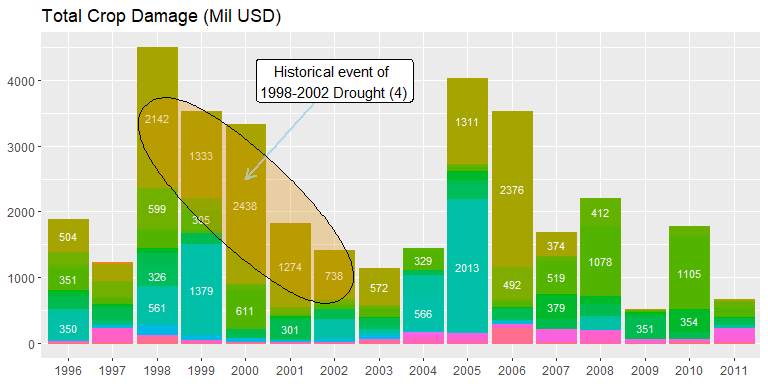
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.



segment\_data = data.frame(x = c(10), xend=c(4), y=c(825), yend=c(750))  
annotate\_point\_x = c(10)  
annotate\_point\_y = c(825)  
annotate\_label = c("600 persons died\nin the event of\nExtreme Heat\n in Chicago, 1999 (3)")  
  
ggplot(sum\_yearly\_storms, aes(x = YEAR, y=TOTFATALITIES, group=EVTYPE, fill=EVTYPE)) +   
 geom\_area()+  
 theme(legend.position = "none", axis.title.y=element\_blank(), axis.title.x=element\_blank()) +   
 ggtitle("Total Fatalities") +  
 geom\_segment(  
 data=segment\_data,  
 aes(x = x, y = y, xend = xend, yend = yend),  
 inherit.aes = F,  
 lineend = "round", # See available arrow types in example above  
 linejoin = "round",  
 size = 1,   
 arrow = arrow(length = unit(0.3, "cm")),  
 colour = "lightblue") + # Also accepts "red", "blue' etc   
   
 annotate("label", x=annotate\_point\_x, y=annotate\_point\_y, label=annotate\_label, )



ellipse\_data = data.frame(x = c(3,4,5,5,6,7), y=c(3500,3000,2500,1500,1500,1200))  
segment\_data = data.frame(x = c(7), xend=c(5), y=c(4000), yend=c(2500))  
annotate\_point\_x = c(7)  
annotate\_point\_y = c(4000)  
annotate\_label = c("Historical event of \n1998-2002 Drought (4)")  
sum\_yearly\_storms %>%  
 mutate(  
 perc = "",  
 perc = if\_else(TOTCROPDMG >= 300, paste0(sprintf("%d", TOTCROPDMG)), perc)  
 ) %>%   
 ggplot(aes(x = YEAR, y=TOTCROPDMG, group=EVTYPE, fill=EVTYPE, color=EVTYPE)) +   
 geom\_bar(stat = "identity")+  
 geom\_text(aes(label=perc),  
 position=position\_stack(vjust=0.5), colour="white", size = 3) +  
 theme(legend.position = "none", axis.title.y=element\_blank(), axis.title.x=element\_blank()) +   
 ggtitle("Total Crop Damage (Mil USD)") +   
 geom\_segment(  
 data=segment\_data,  
 aes(x = x, y = y, xend = xend, yend = yend),  
 inherit.aes = F,  
 lineend = "round", # See available arrow types in example above  
 linejoin = "round",  
 size = 1,   
 arrow = arrow(length = unit(0.3, "cm")),  
 colour = "lightblue") + # Also accepts "red", "blue' etc   
 annotate("label", x=annotate\_point\_x, y=annotate\_point\_y, label=annotate\_label, ) +  
 geom\_mark\_ellipse(data=ellipse\_data, aes(x=x, y=y, fill="blue"), inherit.aes = F, tol=1)



segment\_data = data.frame(x = c(14,5,5), xend=c(11,10,10), y=c(100000,60000,60000), yend=c(75000,85000,35000))  
annotate\_point\_x = c(14,5)  
annotate\_point\_y = c(100000,60000)  
annotate\_label = c("Contributed by \nhistorical event of\n Mid-Atlantic \nUnited States flood (5)", "Distributed by\n2005 Atlantic hurricane season-\nthe most active Atlantic hurricane season\n in history until 2020 (6)")  
  
sum\_yearly\_storms %>%  
mutate(  
 perc = "",  
 perc = if\_else(TOTPROPDMG >= 10000, paste0(sprintf("%d", TOTPROPDMG)), perc)  
 ) %>%   
ggplot(aes(x = YEAR, y=TOTPROPDMG, group=EVTYPE, fill=EVTYPE)) +   
 geom\_bar(stat = "identity")+  
 geom\_text(aes(label=perc),  
 position=position\_stack(vjust=0.5), colour="white", size = 2.5) +  
 theme(legend.position = "none", axis.title.y=element\_blank(), axis.title.x=element\_blank()) +   
 ggtitle("Total property damage (Mil USD)") +  
 geom\_segment(  
 data=segment\_data,  
 aes(x = x, y = y, xend = xend, yend = yend),  
 inherit.aes = F,  
 lineend = "round", # See available arrow types in example above  
 linejoin = "round",  
 size = 1,   
 arrow = arrow(length = unit(0.3, "cm")),  
 colour = "lightblue") + # Also accepts "red", "blue' etc   
 annotate("label", x=annotate\_point\_x, y=annotate\_point\_y, label=annotate\_label, )

