---
title: Analysis of Health & Economic Consequences of Severe Weather Events Across
  the USA
author: "Amal Haq"
date: "Wednesday, February 25, 2015"
output: html_document

#SYNOPSIS

Weather events have a major impact on both the population and the economy. In the report below, we use data gathered by the US National Weather Service to determine the Weather Events that result in the highest population and economic impact, in death/injury tolls and US dollars respectively. The data has been heavily processed to align with the 48 weather event types recognized by the National Weather Service; additionally, the data is only reflective of weather events from the year 1993 onwards. Please note that this is largely an exploratory analysis of the available data.

#SETTING GLOBAL ENVIRONMENT

```{r environment, echo=TRUE}
library(ggplot2)
library(gridExtra)
library(data.table)
library(Rcpp)
library(plyr)
library(dplyr)
library(lubridate)
```

#DATA PROCESSING

##Data Download and Upload

Using 'If' loops here will first check the working directory and the environment to see if the data already exists. This helps save time in download and upload. As recommended in the instructions, we can also cache this code so it doesn't have to run the code each time.
```{r, cache=TRUE}
if(!file.exists("./project2data/stormData.csv.bz2"))
{dir.create("./project2data")}
fileURL<-
"http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
download.file(fileURL, destfile = "./project2data/stormData.csv.bz2")


if (!"storm.data" %in% ls()) {
storm.data = read.csv(bzfile("project2data/stormData.csv.bz2"), header = TRUE)
}
```

##Data Subsetting and Cleaning

The instructions note that the earlier years represent a lack of good data. Lets evaluate the dataset to determine a good cut-off point for subsetting the data for the analysis. We can do this by calculating the different types of events per year.
The `r class(storm.data[2])` shows us that our date column is actually of class 'factor'; we need to extract the year from each date and put it in a variable column, so that we can split the dataset by year. We can use *lubridate* package functions:
```{r}
storm.data$year<- mdy_hms(storm.data$BGN_DATE)
storm.data$year<- year(storm.data$year)
```
Alternatively, use the base R code. The difference is merely in the class() of the storm.data$year variable.
```{r}
storm.data$year<-as.Date(storm.data$BGN_DATE, "%m/%d/%Y")
storm.data$year<- format(storm.data$year, "%Y")
```
If you check with `r dim(storm.data)` you will see that there are now 902297 observations for 38 variables, which means that a new column "year" has been added to the data frame.
While we are at it, lets also change the `r class(storm.data$EVTYPE)` from 'factor' to 'character'.
```{r}
storm.data$EVTYPE<- as.character(storm.data$EVTYPE)
```
Now, lets split the data by year to calculate different events per year:
```{r, results ='hold'}
no.events<- ddply(
  storm.data,
  .(year),
  summarise,
  count=length(unique(EVTYPE))
  )
```
Using *plyr* is the in vogue way of doing it, but the following uses base R code with traditional split/apply/combine
```{r}
no.events <- split(storm.data, storm.data$year)
no.events <- lapply(no.events, function (x) length(unique(x$EVTYPE)) )
no.events <- do.call(rbind, no.events)
```
The table shows a significant spike in events recorded after 1993; using this as a cut-off point to subset will reduce noise in further calcuations
```{r}
subset.storm<- storm.data[storm.data$year >=1993,]
```
This subset reduced the number of observations from 902297 to 714738.

Let's do some heavy duty cleaning! By checking the unique values in the EVTYPE variable, we see that there are 985 different entries of EVTYPE (i.e. Event

Type). According to the documentation provided in the assignment (the National Weather Service Instruction 10-1605, page 6), there should be only 48 different event types. There are a lot of inconsistencies, typos, different names for the same type of event. Therefore clean up the subset.storm$EVTYPE entries for the 714738 observations is necessary.

First, lets put all the event types in uppercase and remove unnecessary spaces. That in itself will reduce the number of unique EVTYPE:
```{r}
subset.storm$EVTYPE<- toupper(gsub("^\\s+|\\s+$", "", subset.storm$EVTYPE))
```

Let's start with Thunderstorms. There are two distinct categories "THUNDERSTORM WIND" and "MARINE THUNDERSTORM WIND". Let's see what we have:
```{r}
unique(subset.storm[grep("THUNDERSTORM", subset.storm$EVTYPE),8])
unique(subset.storm[grep("MARINE THUNDERSTORM", subset.storm$EVTYPE),8])
unique(subset.storm[grep("(.*)+ THUNDERSTORM", subset.storm$EVTYPE),8])
length(subset.storm[grep("THUNDERSTORM", subset.storm$EVTYPE),8])
length(subset.storm[grep("MARINE THUNDERSTORM", subset.storm$EVTYPE),8])
```

There are 81 unique entry types for 'Thunderstorms', one of which is "MARINE THUNDERSTORM WIND". And there are 9 unqiue entries that include at least one word before 'Thunderstorm'. We want to bring down 80 unique event types to a single event type "THUNDERSTORM WIND" without affecting the "MARINE THUNDERSTORM WIND" eventype.
```{r}
subset.storm$EVTYPE<- gsub("MARINE THUNDERSTORM", "MARINE
TSTM",subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("(.*)+ THUNDERSTORM",
"THUNDERSTORM",subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("THUNDERSTORM(.*)", "THUNDERSTORM",
subset.storm$EVTYPE)
```

But we also have entries for 'Thunderstorms' as "TSTM".
```{r}
unique(subset.storm[grep("TSTM", subset.storm$EVTYPE),8])
```
The code below takes care of any entries that begin with *TSTM WIND* to include 'TSTM WIND/TSTM WIND*S*/TSTM WIND*/HAIL*/TSTM WIND *(insert any other word or words*)'
```{r}
subset.storm$EVTYPE<- gsub("^TSTM WIND(.*)|^TSTM WIND (.*)", "THUNDERSTORM",
subset.storm$EVTYPE)
```
Then, the code below takes care of any remaining entries. Specifically, any entries that begin with *TSTM*, * TSTM* (space before TSTM), and the specific entries *TSTM* and *TSTMW*.
```{r}
subset.storm$EVTYPE<- gsub("^TSTM (.*)|^\\s+TSTM (.*)|^TSTM$|TSTMW",
"THUNDERSTORM", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^THU.*|^TUN.*", "THUNDERSTORM",
subset.storm$EVTYPE)
```

```

We have shaved off over 100 unique entries (from 985 to 881) by recategorizing them as regular or marine thunderstorms. To be concise and to maintain clarity, we have change the "THUNDERSTORM WIND" and "MARINE THUNDERSTORM WIND" categories to "THUNDERSTORM" and "MARINE TSTM" respectively.
**SIMILARLY**, we need to continue our data clean-up process to boil down our unique entries in storm.data$EVTYPE and bring it as close as possible to the 48 categories recognized by the National Weather Service. At the very least, we have to at least clean up the data so the 48 event types are reflected properly for our later computations.

Because we don't have a Subject Matter Expert at hand, we will need to make some educated guesses about categories (e.g., a 'BEACH FLOOD' entry can be recategorized as 'COASTAL FLOOD'). Additionally, anywhere the entry indicates multiple event types (e.g., 'SLEET/SNOW'), we will categorize by the first entry for simplicity. Therefore, SLEET/SNOW is categorized as 'SLEET', and SNOW/SLEET is categorized as 'SNOW'.

Below, with the help of the unique() function to make decisions about appropriate substitutions, and the gsub() function, we can clean up the remainder of the EVTYPE variable.

```{r}
subset.storm$EVTYPE<- gsub("^BLIZZARD(.*)|GROUND BLIZZARD", "BLIZZARD",
subset.storm$EVTYPE)
unique(subset.storm[grep("COASTAL", subset.storm$EVTYPE),8])
subset.storm$EVTYPE<- gsub("(.*)COASTAL(.*)", "COASTAL FLOOD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("TIDAL(.*)|BEACH(.*)|CSTL(.*)", "COASTAL FLOOD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("LAKE(.*) FLOOD", "LAKESHORE FLOOD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^FLASH FLOOD(.*)", "FLASH FLOOD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("FLASH FLOOODING|^ +FLASH FLOOD|LOCAL FLASH FLOOD",
"FLASH FLOOD", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^FLOOD(.*)|LOCAL FLOOD|BREAKUP FLOODING", "FLOOD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^URBAN(.*) FLOOD(.*)|^R(.*) FLOOD(.*)|^S(.*)
FLOOD(.*)|^M(.*) FLOOD(.*)|^H(.*) FLOOD(.*)", "FLOOD", subset.storm$EVTYPE)
unique(subset.storm[grep("COLD", subset.storm$EVTYPE),8])
subset.storm$EVTYPE<- gsub("EXTREME.*COLD.*|RECORD +COLD|SEVERE COLD", "EXTREME
WIND CHILL", subset.storm$EVTYPE)
unique(subset.storm[grep("FOG", subset.storm$EVTYPE),8])
subset.storm$EVTYPE<- gsub("FREEZING FOG", "FF", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*FOG.*", "FOG", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^FF$", "FREEZING FOG", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^WIND CHILL.*", "COLD/WIND CHILL",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^COLD.*|.*[^/\\]COLD.*", "COLD/WIND CHILL",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*SMOKE.*", "DENSE SMOKE", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*[^/]DROUGHT|^DROUGHT.*", "DROUGHT",
subset.storm$EVTYPE)

```
subset.storm$EVTYPE<- gsub("^DUST D(.*)", "DUST DEVIL", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("DUSTSTORM|.* DUST +.*|DUST STORM/.*|.*DUST$", "DUST
STORM", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^BITTER|^LOW TEMPERATURE|.*LOW TEMP|RECORD LOW$",
"EXTREME COLD/WIND CHILL", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("EXTREME.*WIND.*CHILL.*", "EXTREME COLD/WIND CHILL",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*FROST.*", "FROST/FREEZE", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*FUNNEL.*|.*WALL CLOUD.*", "FUNNEL CLOUD",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("MARINE HAIL", "MARINE-H", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^HAIL.*|.*[^/]HAIL.*", "HAIL", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("MARINE-H", "MARINE HAIL", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^HEAT.*", "HEAT", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^EXCESSIVE.*HEAT.*|^RECORD.*HEAT|^EXTREME.*HEAT",
"EXCESSIVE HEAT", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^RECORD.*TEMP.*|^HIGH.*TEMP", "EXCESSIVE HEAT",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^RAIN.*|^HEAVY RAIN.*|LOCALLY HEAVY RAIN", "HEAVY
RAIN", subset.storm$EVTYPE)
subset.storm$EVTYPE<-
gsub("^RECORD.*RAIN.*|^EXCESSIVE.*RAIN.*|^HVY.*|^TORRENTIAL.*|^PROLONGED.*|^HEAVY
 PRECIP.*|^HEAVY SHOWER.*", "HEAVY RAIN", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^SNOW.*|^HEAVY SNOW.*", "HEAVY SNOW",
subset.storm$EVTYPE)
subset.storm$EVTYPE<-
gsub(".*RECORD.*SNOW.*|^EXCESSIVE.*SNOW.*|^BLOWING.*|THUNDERSNOW|.*WET SNOW",
"HEAVY SNOW", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("RIP.*", "RIP CURRENT", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*SURF.*", "HIGH SURF", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^HIGH.*WIND.*", "HIGH WIND", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^HURRICANE.*", "HURRICANE(TYPHOON)",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^SLEET.*|^FREEZING RAIN|FREEZING DRIZZLE.*",
"SLEET", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*ICE.*", "ICE STORM", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*LAKE.*SNOW", "LAKE-EFFECT SNOW",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*LIGHTNING.*", "LIGHTNING", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^STORM SURGE.*", "STORM SURGE/TIDE",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("STRONG WIND.*|^GUSTY.*|.*BURST.*", "STRONG WIND",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^TORNADO.*", "TORNADO", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^TROPICAL STORM.*", "TROPICAL STORM",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^TORNADO.*", "TORNADO", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^VOLCANIC.*", "VOLCANIC ASH", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub(".*WATERSPOUT.*|^WA.*SPOUT", "WATERSPOUT",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^WILD.*FIRE.*", "WILDFIRE", subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^WINTER STORM.*", "WINTER STORM",
subset.storm$EVTYPE)
subset.storm$EVTYPE<- gsub("^WINTER WEATHER.*", "WINTER WEATHER",
subset.storm$EVTYPE)
```

```

We have significantly reduced the number of unique event types (from 985 to 309). This is not the best that it could but we will be able to further reduce our dataset size by cleaningthe 'PROPDMGEXP' 'CROPDMGEXP' variables, and then subsetting by the 'PROPDMG' 'CROPDMG' variables.
```{r}
unique(subset.storm$PROPDMGEXP)
unique(subset.storm$CROPDMGEXP)
```
The PROPDMEXP/CROPDMGEXP should provide the 'exponent' or power to which the PROPDMG/CROPDMG valriables have to be raised. The letters H,K,M,B, represent the exponents 2,3,6,9 or the exponential values 10^2, 10^3, 10^6, 10^9 respectively. However, the PROPDMGEXP and CROPDMGEXP columns have both alphabetic and numeric entries, as well as some random symbols. This mean, for example, anywhere there is a '5' in PROPDMEXP/CROPDMGEXP, the PROPDMG/CROPDMG values have to be multiplied by 10^5 to give the total value of the damage. Whereas, anywhere there is a 'K' in PROPDMEXP/CROPDMGEXP, the PROPDMG/CROPDMG values have to be multiplied by 10^3. Therefore, we first have to standardize the entry format. We can use the gsub() function again and write a loop to make the necessary replacements.
```{r}
subset.storm$PROPDMGEXP<- gsub("\\+|-|\\?", "0", subset.storm$PROPDMGEXP)
subset.storm$CROPDMGEXP<- gsub("\\?", "0", subset.storm$CROPDMGEXP)
subset.storm$PROPDMGEXP<- toupper(subset.storm$PROPDMGEXP)
subset.storm$CROPDMGEXP<- toupper(subset.storm$CROPDMGEXP)

exponent <- function(X) {
        if (X == "H") {
                    X<-2
                    }
            else if (X == "K"){
                    X<-3
                    }
            else if (X == "M"){
                    X<-6
                    }
            else if (X == "B"){
                    X<-9
                    }
            else if (X == ""){
                    X<-0
                    }
        else{
                X<-X
                }
        }

```
As you can see from the code above, unless the entry is a letter or a number, I have set the exponent to be 0. Exponent set to 0 means that the property or crop damage is being raised to 10^0, and 10^0 =1. In these cases, the multiplier is 1, or in other words: *Total Property Damage = PROPDMG x 10^PROPDMXEXP = PROPDMG x 1 = PROPDMG*
```

Then we simply apply the function to the variables:
```{r}
subset.storm$PROPDMGEXP<- as.numeric(sapply(subset.storm$PROPDMGEXP, exponent))
subset.storm$CROPDMGEXP<- as.numeric(sapply(subset.storm$CROPDMGEXP, exponent))
```
Now, we can create new variables that reflect the total property damage and crop damage values.
```{r}
subset.storm<- mutate(subset.storm, TOTALPROPDMG = subset.storm$PROPDMG *
10^subset.storm$PROPDMGEXP)
subset.storm<- mutate(subset.storm, TOTALCROPDMG = subset.storm$CROPDMG *
10^subset.storm$CROPDMGEXP)
```

In our analysis in the following section, we are interested in events that are either harmful or cause economic damage, we can do an additional subset to reduce our dataset to exclude event types that do not cause injuries, fatalities, property damage or crop damage:
```{r}
subset.storm<- subset(subset.storm, INJURIES > 0 | FATALITIES > 0 |
TOTALPROPDMG > 0 | TOTALCROPDMG > 0)
```

#RESULTS

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

Let's process the cleaned and subsetted data we produced in the section above a little more. We calculate the *total fatalities*, *total injuries* as well as the *total population* affected (fatalities plus injuries), for each event type. We can do this easily by grouping the data along the EVTYPE variable and creating new data frames with variables that reflect the totals.
```{r}
popl.impact<- ddply(
  subset.storm,
  c("EVTYPE"),
  summarise,
  TOTAL.FATALITIES=sum(FATALITIES),
  TOTAL.INJURIES=sum(INJURIES),
  TOTAL.POP.DMG=sum(FATALITIES+INJURIES)
  )

popl.deaths<- popl.impact[order(-popl.impact$TOTAL.FATALITIES),]
popl.deaths<- popl.deaths[1:10, 1:2]

popl.injuries<- popl.impact[order(-popl.impact$TOTAL.INJURIES),]
popl.injuries<- popl.injuries[1:10, c("EVTYPE", "TOTAL.INJURIES")]

popl.overall<- popl.impact[order(-popl.impact$TOTAL.POP.DMG),]
popl.overall<- popl.overall[1:10, c("EVTYPE", "TOTAL.POP.DMG")]
```

Now, plot:

````
```{r, results='asis'}
g1<- ggplot(popl.deaths, aes(x= reorder(EVTYPE, TOTAL.FATALITIES),
y=TOTAL.FATALITIES)) +
        geom_bar(stat = "identity") +
        ggtitle("Deaths by Weather Event") +
        xlab("") +
        ylab("Total Deaths") +
        theme(axis.text.x=element_text(angle = 25, hjust = 1), plot.margin =
unit(c(0,1,0,1), "lines"), plot.title=element_text(size=12))

g2<- ggplot(popl.injuries, aes(x= reorder(EVTYPE, TOTAL.INJURIES),
y=TOTAL.INJURIES)) +
        geom_bar(stat = "identity") +
        ggtitle("Injuries by Weather Event") +
        xlab("") +
        ylab("Total Injuries") +
        theme(axis.text.x=element_text(angle = 45, hjust = 1), plot.margin =
unit(c(0,0,0,0), "lines"), plot.title=element_text(size=12))

g3<- ggplot(popl.overall, aes(x= reorder(EVTYPE, TOTAL.POP.DMG),
y=TOTAL.POP.DMG)) +
        geom_bar(stat = "identity") +
        ggtitle("Worst Weather Event for Population") +
        xlab("") +
        ylab("Total Population Affected") +
        theme(axis.text.x=element_text(angle = 45, hjust = 1), plot.margin =
unit(c(0,1,0,1), "lines"))

grid.arrange(g1,g2,g3, nrow=2, ncol=2)
```

**The oveall most harmful event with respect to population health is a TORNADO;
however, the weather event that causes most deaths is EXCESSIVE HEAT. TORNADOES
are a close second for population deaths, however, they are the primary weather
event that for the highest number of injuries.**

## QUESTION 2: Across the United States, which types of events have the
greatest economic consequences?

We can process the cleaned and subsetted data we prodcued in the previous
section for economic impact in exactly the same way we did for population
impact:
```{r}
econ.impact<- ddply(
  subset.storm,
  c("EVTYPE"),
  summarise,
  TOTALPROPDMG=sum(TOTALPROPDMG),
  TOTALCROPDMG=sum(TOTALCROPDMG),
  TOTALDMG=sum(TOTALPROPDMG+TOTALCROPDMG)
  )

econdmgprop<- econ.impact[order(-econ.impact$TOTALPROPDMG),]
econdmgprop<- econdmgprop[1:10, 1:2]
````
````

````
econdmgcrop<- econ.impact[order(-econ.impact$TOTALCROPDMG),]
econdmgcrop<- econdmgcrop[1:10, c("EVTYPE", "TOTALCROPDMG")]

econ.overall<- econ.impact[order(-econ.impact$TOTALDMG),]
econ.overall<- econ.overall[1:10, c("EVTYPE", "TOTALDMG")]
```
````

And, we can plot with a nearly identical code, making sure to use the correct arguments and labels:

````
```{r, results='asis'}
g4<- ggplot(econdmgprop, aes(x= reorder(EVTYPE, TOTALPROPDMG), y=TOTALPROPDMG))
+
        geom_bar(stat = "identity") +
        ggtitle("Property Damage by Weather Event") +
        xlab("") +
        ylab("Total Damage") +
        theme(axis.text.x=element_text(angle = 25, hjust = 1), plot.margin =
unit(c(0,1,1,0), "lines"), plot.title=element_text(size=12))

g5<- ggplot(econdmgcrop, aes(x= reorder(EVTYPE, TOTALCROPDMG), y=TOTALCROPDMG))
+
        geom_bar(stat = "identity") +
        ggtitle("Crop Damage by Weather Event") +
        xlab("") +
        ylab("Total Damage") +
        theme(axis.text.x=element_text(angle = 20, hjust = 1), plot.margin =
unit(c(0,0,0,0), "lines"), plot.title=element_text(size=12))

g6<- ggplot(econ.overall, aes(x= reorder(EVTYPE, TOTALDMG), y=TOTALDMG)) +
        geom_bar(stat = "identity") +
        ggtitle("Worst Weather Event for Economy") +
        xlab("") +
        ylab("Total Econ. Dmg.") +
        theme(axis.text.x=element_text(angle = 25, hjust = 1), plot.margin =
unit(c(0,0,0,0), "lines"))

grid.arrange(g4,g5,g6, nrow=2, ncol=2)
```
````

The oveall most harmful event with respect to economic damage is a FLOOD. The weather event that causes most property damage is also the FLOOD, however, DROUGHTS are responsible for the most crop damage. Interestingly, DROUGHTS are only the fifth highest contributors to total economic damage, succeeded by other events such as hurricanes, tides, tornadoes, hail and flash flood.
````