PA2

R Markdown

reading the dataset from web and test it

```
library(ggplot2)
setwd("C://Users//pg000//Desktop//Download")
fileUrl <- "http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
destfile <- ".//repdata-data-StormData.csv.bz2"
if(!file.exists(destfile)) {
 download.file(fileUrl, destfile = destfile, quiet = TRUE)
 dateDownload <- date()
}
rawData <- read.csv(bzfile(destfile), stringsAsFactors = FALSE)
names(rawData)
```

```
##  [1] "STATE__"    "BGN_DATE"   "BGN_TIME"   "TIME_ZONE"  "COUNTY"
##  [6] "COUNTYNAME" "STATE"      "EVTYPE"     "BGN_RANGE"  "BGN_AZI"
## [11] "BGN_LOCATI" "END_DATE"   "END_TIME"   "COUNTY_END" "COUNTYENDN"
## [16] "END_RANGE"  "END_AZI"    "END_LOCATI" "LENGTH"     "WIDTH"
## [21] "F"          "MAG"        "FATALITIES" "INJURIES"   "PROPDMG"
## [26] "PROPDMGEXP" "CROPDMG"    "CROPDMGEXP" "WFO"        "STATEOFFIC"
## [31] "ZONENAMES"  "LATITUDE"   "LONGITUDE"  "LATITUDE_E" "LONGITUDE_"
## [36] "REMARKS"    "REFNUM"
```

```
rawData$Total_cas <- rawData$FATALITIES + rawData$INJURIES
value <- function(x) {
 x <- tolower(x)
 if(x=="k")res <- 1000
 if(x == "m")res <- 1e+06
 if(x == "b") res <- 1e+09
 else res <- 1
 res
}
rawData$pd <- rawData$PROPDMG * sapply(rawData$PROPDMGEXP, value)/1000000
rawData$cd <- rawData$CROPDMG * sapply(rawData$CROPDMGEXP, value)/1000000
rawData$td <- rawData$pd + rawData$cd
```

taking relevant variable new dataset is constructed

```
proc_data <- rawData[,c("EVTYPE","FATALITIES","INJURIES",
                        "Total_cas","pd","cd","td")]
proc_data <- aggregate(proc_data[,2:7],
           by =list(proc_data$EVTYPE), FUN = sum, na.rm = TRUE)

colnames(proc_data) <- c("EVETYPE", colnames(proc_data[2:7]))
```

top_data() function takes data frame (df), column number (col) and returns the top results.

```
top_data <- function(df, col,top) {
 df <- df[,c(1,col)]
 df <- df[order(df[,2],decreasing = TRUE),]
 df <- df[1:top,]
 rownames(df) <- NULL
 df
}
```

top 3 events with FATALITIES

```
top_data(proc_data,2,3)
```

```
##          EVETYPE FATALITIES
## 1        TORNADO       5633
## 2 EXCESSIVE HEAT       1903
## 3    FLASH FLOOD        978
```

check the INJURIES

```
top_data(proc_data,3,3)
```

```
##     EVETYPE INJURIES
## 1   TORNADO    91346
## 2 TSTM WIND     6957
## 3     FLOOD     6789
```

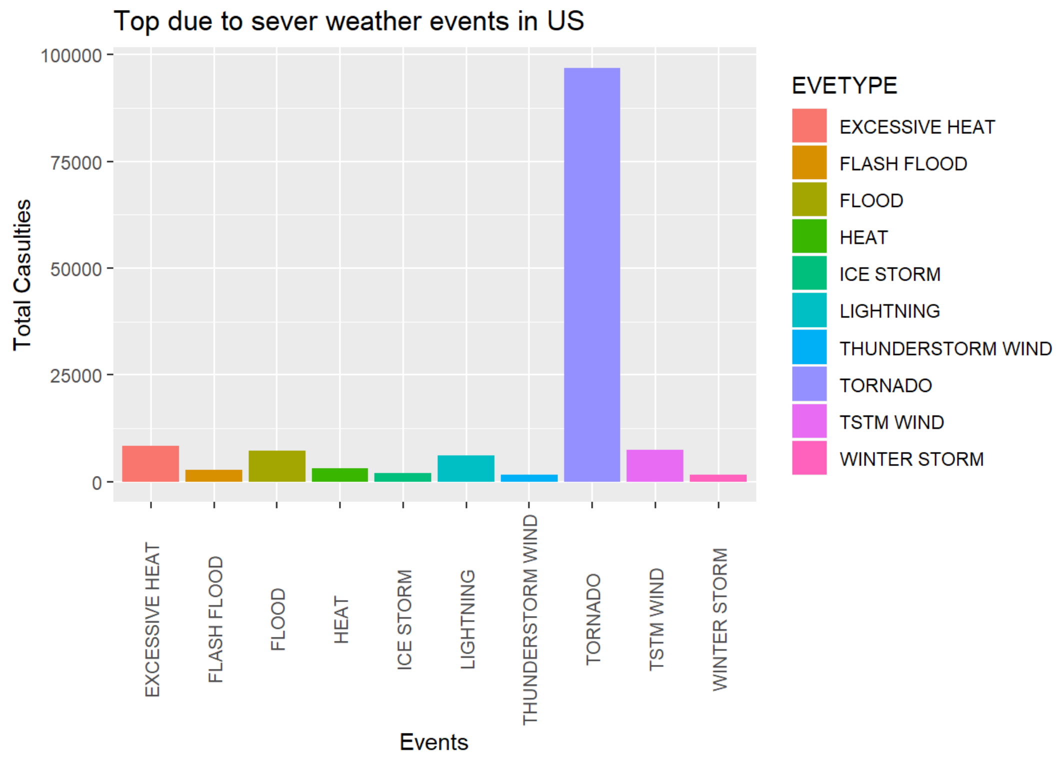
top 10 events with total casualities

```
cu_data <- top_data(proc_data, 4, 10)
cu_data
```

```
##               EVETYPE Total_cas
## 1             TORNADO     96979
## 2      EXCESSIVE HEAT      8428
## 3           TSTM WIND      7461
## 4               FLOOD      7259
## 5           LIGHTNING      6046
## 6                HEAT      3037
## 7         FLASH FLOOD      2755
## 8           ICE STORM      2064
## 9   THUNDERSTORM WIND      1621
## 10       WINTER STORM      1527
```

plot the top 10 events with most total casualities

```
ggplot(cu_data, aes(x=EVETYPE, y=Total_cas, fill = EVETYPE))+
geom_bar(stat = "identity")+
ggtitle("Top due to sever weather events in US")+
 xlab("Events")+
 ylab("Total Casulties")+
 theme(axis.text.x = element_text(angle = 90, vjust =0.5))
```



## show the United States

which the types of events have greatest consequence

```
top_data(proc_data, 5,3)
```

```
##           EVETYPE        pd
## 1           FLOOD 122500.90
## 2 HURRICANE/TYPHOON  65500.01
## 3     STORM SURGE  42560.02
```

the top 10 event with most total_cas

```
top_data(proc_data, 6,3)
```

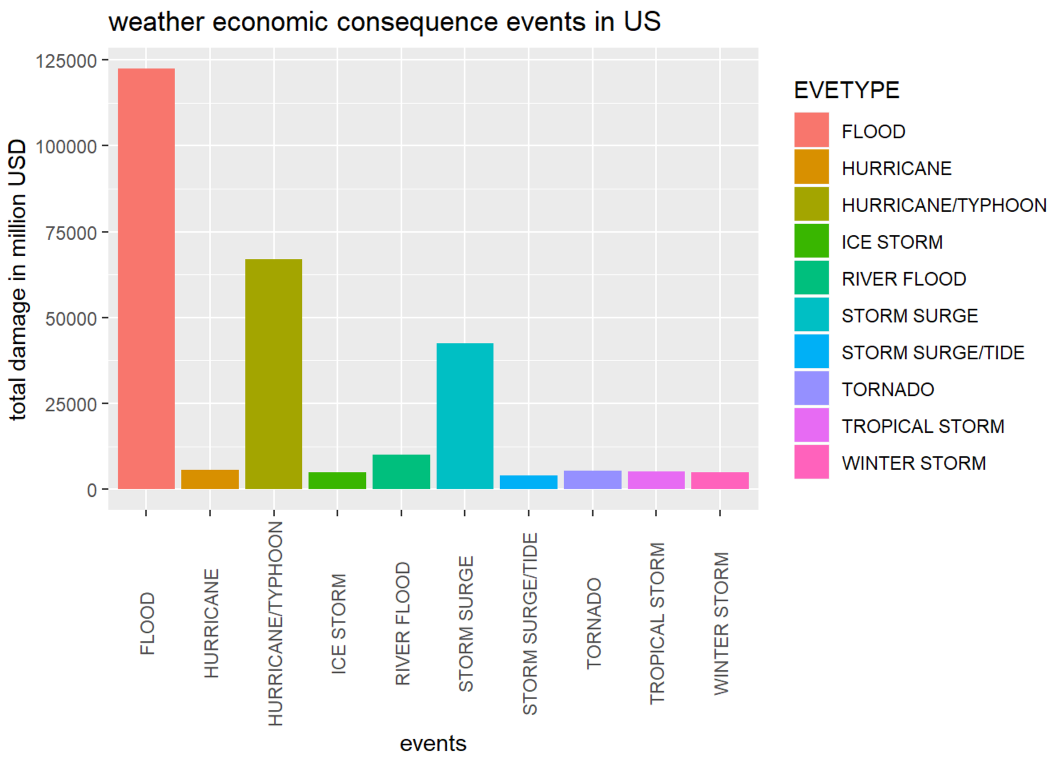
```
##           EVETYPE       cd
## 1     RIVER FLOOD 5000.003
## 2       ICE STORM 5000.002
## 3 HURRICANE/TYPHOON 1510.005
```

```
dmg_data <- top_data(proc_data, 7,10)
dmg_data
```

```
##             EVETYPE         td
## 1             FLOOD 122501.068
## 2  HURRICANE/TYPHOON  67010.011
## 3       STORM SURGE  42560.019
## 4       RIVER FLOOD  10000.017
## 5         HURRICANE   5700.021
## 6           TORNADO   5303.312
## 7    TROPICAL STORM   5150.054
## 8      WINTER STORM   5000.135
## 9         ICE STORM   5000.068
## 10 STORM SURGE/TIDE   4000.008
```

plot the graph

```
ggplot(dmg_data, aes(x = EVETYPE, y = td, fill =EVETYPE))+
 geom_bar(stat ="identity") +
 ggtitle("weather economic consequence events in US") +
 xlab("events")+
 ylab("total damage in million USD")+
 theme(axis.text.x = element_text(angle = 90, vjust =0.5))
```



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