Project 2

1: Synopsis

The goal of the assignment is to explore the NOAA Storm Database and explore the effects of severe weather events on both population and economy. The time period for the database is between 1950 and November 2011.

Following analysis investigates which types of severe weather events are most harmful on:

1. Health (injuries and fatalities)
2. Property and crops (economic consequences)

2: Data Processing

2.1: Data Loading

Download the raw data file and extract the data into a dataframe.Then convert to a data.table

```
## -- Attaching packages ---------------------------------------------------------------------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

2.2: Examining Column Names

```
colnames(storm)
```

```
##  [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"
```

2.3: Data Subsetting

Subset the dataset on the parameters of interest. Basically, we remove the columns we don't need for clarity.

```
# Only use data where fatalities or injuries occurred.
storm <- storm %>%
    filter(EVTYPE != "?" &
               (INJURIES > 0 | FATALITIES > 0 |
                    PROPDMG > 0 | CROPDMG > 0)) %>%
    select( c("EVTYPE", "FATALITIES", "INJURIES", "PROPDMG",
              "PROPDMGEXP","CROPDMG", "CROPDMGEXP"))
```

2.4: Converting Exponent Columns into Actual Exponents instead of (-,+, H, K, etc)

Making the PROPDMGEXP and CROPDMGEXP columns cleaner so they can be used to calculate property and crop cost.

```
# Map property damage alphanumeric exponents to numeric values.
storm$PROPDMGEXP <-plyr::mapvalues(storm$PROPDMGEXP,
                                   from = c("K", "M","", "B", "m", "+", "0", "5",
                                            "6", "?", "4", "2", "3", "h", "7",
                                            "H", "-", "1", "8"),
                                   to = c(10^3, 10^6, 1, 10^9, 10^6, 0,1,10^5, 10^6,
                                          0, 10^4, 10^2, 10^3, 10^2, 10^7, 10^2,
                                          0, 10, 10^8))
```

```
## The following `from` values were not present in `x`: ?, 1, 8
```

```
storm$PROPDMGEXP <- as.numeric(as.character(storm$PROPDMGEXP))
storm$PROPDMGTOTAL <- (storm$PROPDMG * storm$PROPDMGEXP)/1000000000

# Map crop damage alphanumeric exponents to numeric values
storm$CROPDMGEXP <- plyr::mapvalues(storm$CROPDMGEXP,
                                    from = c("","M", "K", "m", "B",
                                             "?", "0", "k","2"),
                                    to = c(1,10^6, 10^3, 10^6, 10^9,
                                           0, 1, 10^3, 10^2))
```

```
## The following `from` values were not present in `x`: 2
```

```
storm$CROPDMGEXP <- as.numeric(as.character(storm$CROPDMGEXP))
storm$CROPDMGTOTAL <- (storm$CROPDMG * storm$CROPDMGEXP)/1000000000
```

2.5: Making Economic Cost Columns

```
storm <- storm %>%
    mutate(propCost = PROPDMG * PROPDMGEXP,
           cropCost = CROPDMG * CROPDMGEXP)
```

2.6: Calcuating Total Property and Crop Cost

```
totalCost <- storm %>%
    group_by(EVTYPE) %>%
    summarize(propCost = sum(propCost),
              cropCost = sum(cropCost))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(totalCost)
```

```
## # A tibble: 6 x 3
##   EVTYPE                  propCost cropCost
##   <chr>                      <dbl>    <dbl>
## 1 "   HIGH SURF ADVISORY"   200000        0
## 2 " FLASH FLOOD"             50000        0
## 3 " TSTM WIND"             8100000        0
## 4 " TSTM WIND (G45)"          8000        0
## 5 "AGRICULTURAL FREEZE"          0 28820000
## 6 "APACHE COUNTY"             5000        0
```

```
totalCost <- storm %>%
    group_by(EVTYPE) %>%
    summarize(propCost = sum(propCost),
           cropCost = sum(cropCost)) %>%
    mutate(Total_Cost = propCost + cropCost) %>%
    select(EVTYPE, Total_Cost) %>%
    arrange(-Total_Cost)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
totalCost <- totalCost[1:10, ]
head(totalCost, 5)
```

```
## # A tibble: 5 x 2
##   EVTYPE               Total_Cost
##   <chr>                     <dbl>
## 1 FLOOD             150319678257
```

```
## 2 HURRICANE/TYPHOON  71913712800
## 3 TORNADO            57362333886.
## 4 STORM SURGE        43323541000
## 5 HAIL               18761221986.
```

2.7: Calcuating Total Fatalities and Injuries

```
totalInjuries <- storm %>%
    group_by(EVTYPE) %>%
    summarize( FATALITIES = sum(FATALITIES),
            INJURIES = sum(INJURIES)) %>%
    mutate(total = FATALITIES + INJURIES) %>%
    select(EVTYPE, total ) %>%
    arrange(-total)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
totalInjuries <- totalInjuries[1:10, ]
head(totalInjuries, 5)
```

```
## # A tibble: 5 x 2
##   EVTYPE         total
##   <chr>          <dbl>
## 1 TORNADO        96979
## 2 EXCESSIVE HEAT  8428
## 3 TSTM WIND       7461
## 4 FLOOD           7259
## 5 LIGHTNING       6046
```

3: Results

3.1: Events that are Most Harmful to Population Health

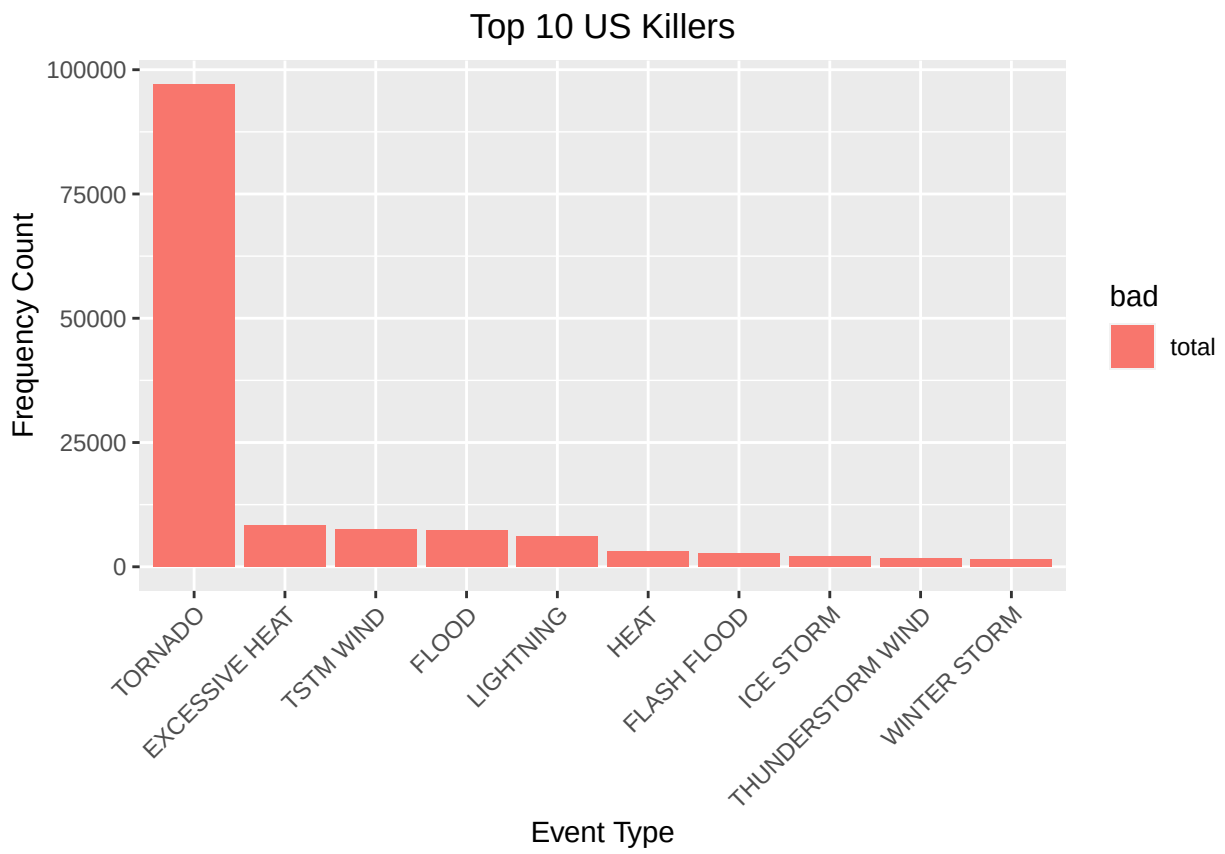
Melting data.table so that it is easier to put in bar graph format

```
bad_stuff <- melt(totalInjuries,
                  id.vars="EVTYPE",
                  variable.name = "bad")
head(bad_stuff, 5)
```

```
##           EVTYPE   bad value
## 1        TORNADO total 96979
## 2 EXCESSIVE HEAT total  8428
## 3      TSTM WIND total  7461
## 4          FLOOD total  7259
## 5      LIGHTNING total  6046
```

```
# Create chart
healthChart <- ggplot(bad_stuff,
                      aes(reorder(EVTYPE, -value), value)) +
    geom_bar(stat="identity",
             aes(fill=bad),
             position="dodge")  +
    ylab("Frequency Count") +
    xlab("Event Type") +
    theme(axis.text.x = element_text(angle=45, hjust=1)) +
    ggtitle("Top 10 US Killers") +
    theme(plot.title = element_text(hjust = 0.5))

healthChart
```



3.2: Events that have the Greatest Economic Consequences

Melting data.table so that it is easier to put in bar graph format

```
econ_consequences <- melt(totalCost,
                          id.vars="EVTYPE",
                          variable.name = "Damage_Type")
head(econ_consequences, 5)
```

```
##              EVTYPE Damage_Type        value
```

```
## 1             FLOOD  Total_Cost 150319678257
## 2 HURRICANE/TYPHOON  Total_Cost  71913712800
## 3           TORNADO  Total_Cost  57362333887
## 4       STORM SURGE  Total_Cost  43323541000
## 5              HAIL  Total_Cost  18761221986
```

```r
# Create chart
econChart <- ggplot(econ_consequences,
                    aes(reorder(EVTYPE, -value),
                        value)) +
    geom_bar(stat="identity",
             aes(fill=Damage_Type),
             position="dodge") +
    ylab("Cost (dollars)") +
    xlab("Event Type") +
    theme(axis.text.x = element_text(angle=45, hjust=1)) +
    ggtitle("Top 10 US Storm Events causing Economic Consequences") +
    theme(plot.title = element_text(hjust = 0.5))

econChart
```