Reproducible Research

Aiman D.

Sunday, March 15, 2015

Initial Settings

```
# Show code
echo = TRUE

# Don't Show Scientific Number Formatting
options(scipen = 1)
```

Loading and preprocessing the data

```
# Extract the csv data file
unzip("activity.zip")

filename <- "activity.csv"
data <- read.csv(filename, colClasses = c("integer", "Date", "factor"))

# Convert date to date data type
data$month <- as.numeric(format(data$date, "%m"))
omitNA <- na.omit(data)
rownames(omitNA) <- 1:nrow(omitNA)
head(omitNA)
```

```
##   steps       date interval month
## 1     0 2012-10-02        0    10
## 2     0 2012-10-02        5    10
## 3     0 2012-10-02       10    10
## 4     0 2012-10-02       15    10
## 5     0 2012-10-02       20    10
## 6     0 2012-10-02       25    10
```

```
dim(omitNA)
```

```
## [1] 15264     4
```

```
# Load plotting library
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

What is mean total number of steps taken per day?

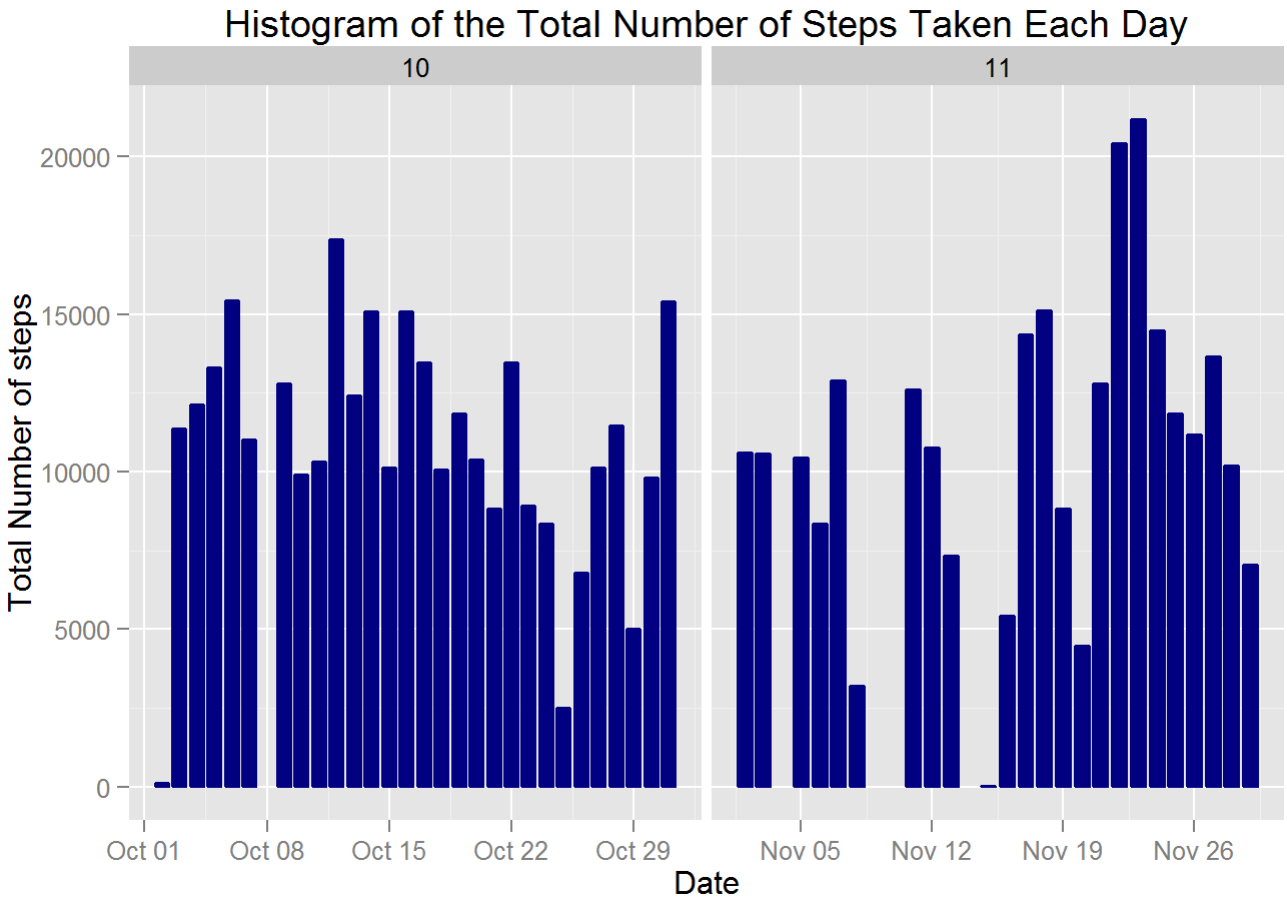
For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day

```
totalDailySteps <- aggregate(omitNA$steps, list(Date = omitNA$date), FUN = "sum")$x
```

2. Make a histogram of the total number of steps taken each day

```
ggplot(omitNA, aes(date, steps)) + geom_bar(stat = "identity", colour = "navy", fill = "nav
y", width = 0.8) + facet_grid(. ~ month, scales = "free") + labs(title = "Histogram of the To
tal Number of Steps Taken Each Day", x = "Date", y = "Total Number of steps")
```



3. Calculate and report the mean and median of the total number of steps taken per day

Mean of the total number of steps taken per day:

```
mean(totalDailySteps)
```

```
## [1] 10766.19
```

Median of the total number of steps taken per day:

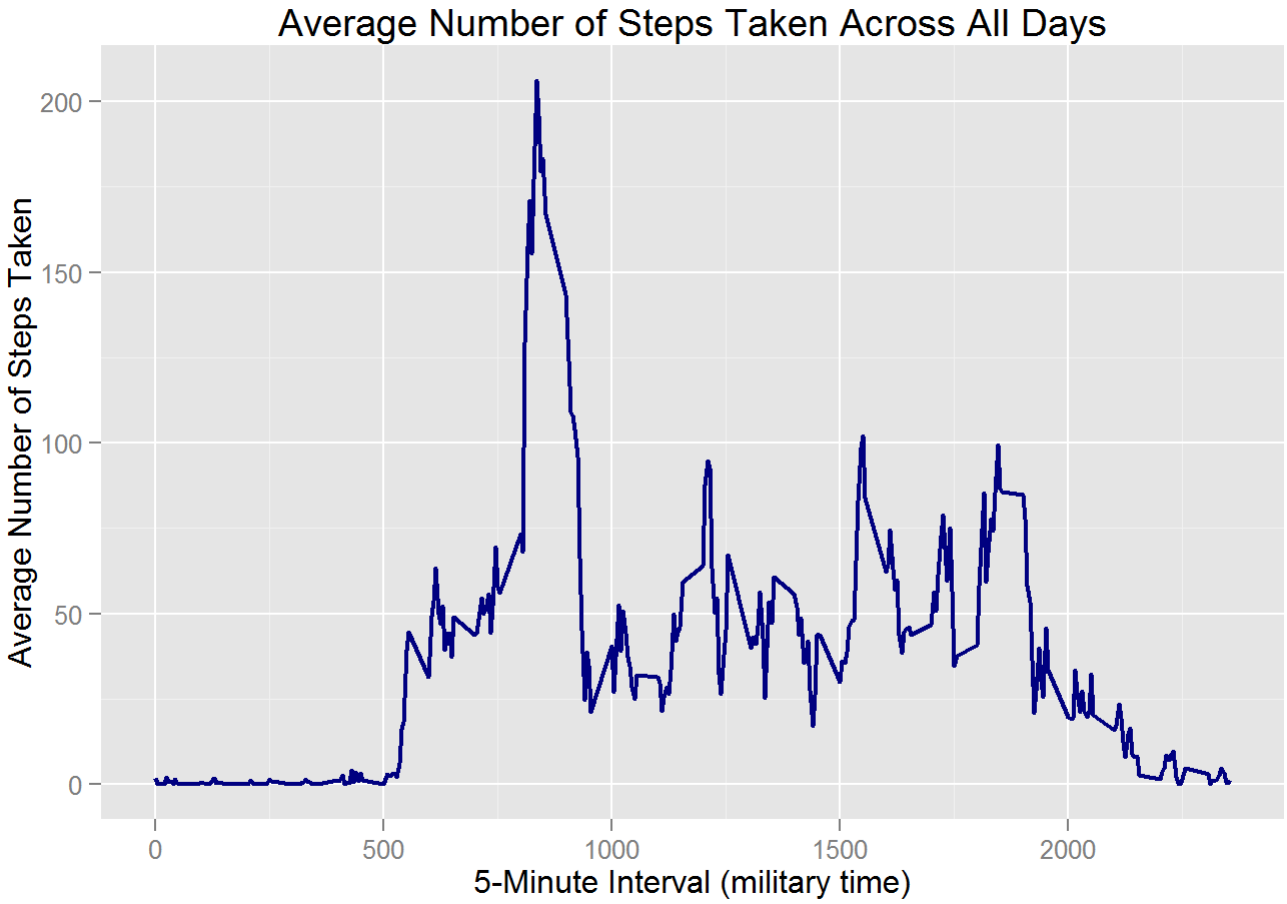
```
median(totalDailySteps)
```

```
## [1] 10765
```

What is the average daily activity pattern?

1. Make a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
averageSteps <- aggregate(omitNA$steps, list(interval = as.numeric(as.character(omitNA$interval))), FUN = "mean")
names(averageSteps)[2] <- "meanOfSteps"

ggplot(averageSteps, aes(interval, meanOfSteps)) + geom_line(color = "navy", size = 0.8) + labs(title = "Average Number of Steps Taken Across All Days", x = "5-Minute Interval (military time)", y = "Average Number of Steps Taken")
```



2.Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
averageSteps[averageSteps$meanOfSteps == max(averageSteps$meanOfSteps), ]
```

```
##     interval meanOfSteps
## 104      835    206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(data))
```

```
## [1] 2304
```

2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

My strategy will be based on the means for the 5-minute intervals to fill for the missing values.

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
newDataset <- data
for (i in 1:nrow(newDataset)) {
    if (is.na(newDataset$steps[i])) {
        newDataset$steps[i] <- averageSteps[which(newDataset$interval[i] == averageSteps$inte
rval), ]$meanOfSteps
    }
}

head(newDataset)
```

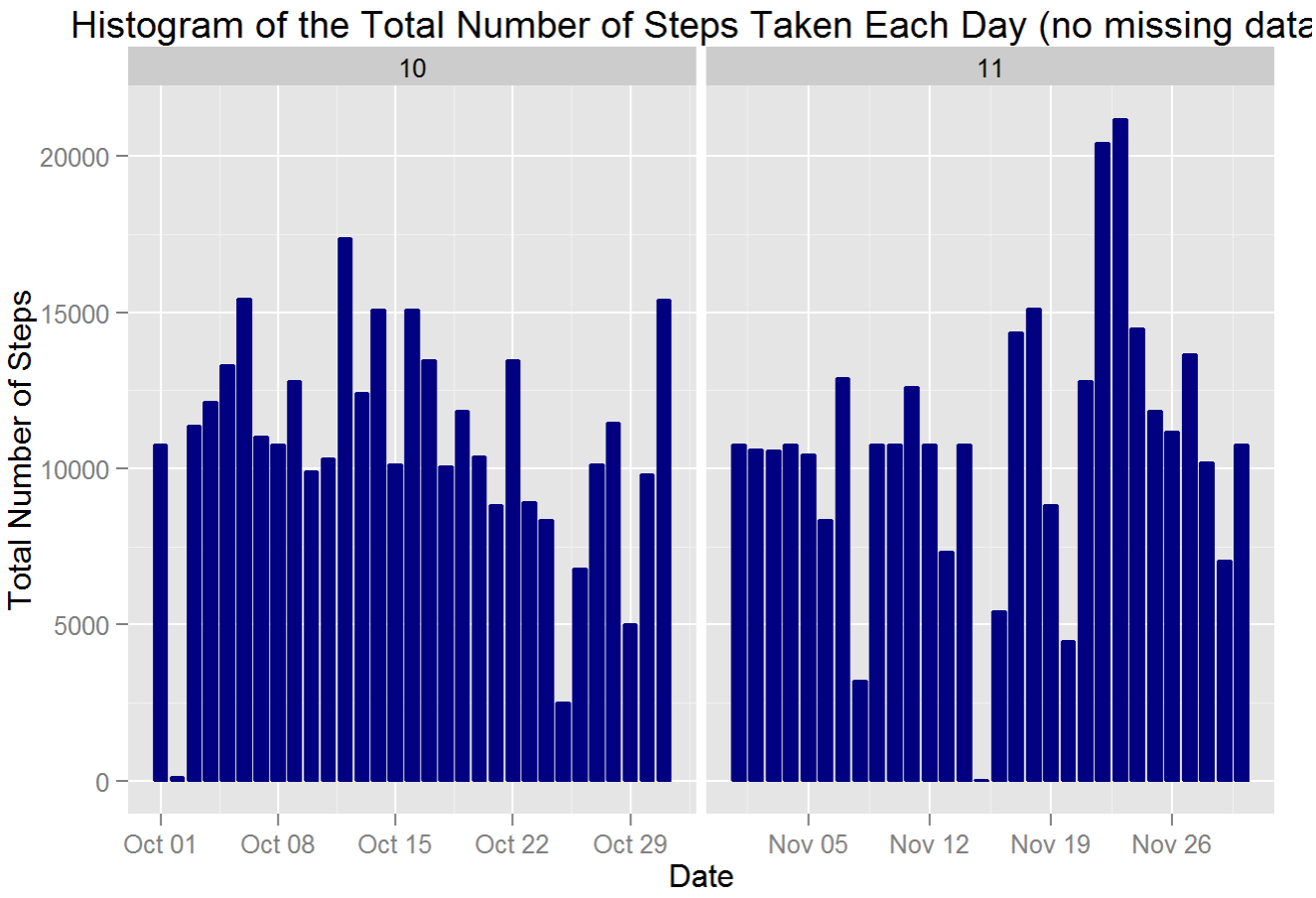
```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

```
sum(is.na(newDataset))
```

```
## [1] 0
```

4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
ggplot(newDataset, aes(date, steps)) + geom_bar(stat = "identity",
                                                colour = "navy",
                                                fill = "navy",
                                                width = 0.8) + facet_grid(. ~ month, scales = "f
ree") + labs(title = "Histogram of the Total Number of Steps Taken Each Day (no missing dat
a)", x = "Date", y = "Total Number of Steps")
```



Mean total number of steps taken per day:

```
newTotalSteps <- aggregate(newDataset$steps, list(Date = newDataset$date), FUN = "sum")$x
newMean <- mean(newTotalSteps)
newMean
```

```
## [1] 10766.19
```

Median total number of steps taken per day:

```
newMedian <- median(newTotalSteps)
newMedian
```

```
## [1] 10766.19
```

Compare them with the two before imputing missing data:

```
oldMean <- mean(totalDailySteps)
oldMedian <- median(totalDailySteps)
newMean - oldMean
```

```
## [1] 0
```

```
newMedian - oldMedian
```

```
## [1] 1.188679
```

After imputing the missing data, both of the old and the new mean of total steps taken per day are equal. Whereas,the new median of total steps taken per day is greater than the old median.

Are there differences in activity patterns between weekdays and weekends?

1. Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
head(newDataset)
```

```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

```
newDataset$weekdays <- factor(format(newDataset$date, "%A"))
levels(newDataset$weekdays)
```

```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
```

```
levels(newDataset$weekdays) <- list(weekday = c("Monday", "Tuesday",
                                                "Wednesday",
                                                "Thursday", "Friday"),
                                    weekend = c("Saturday", "Sunday"))
levels(newDataset$weekdays)
```

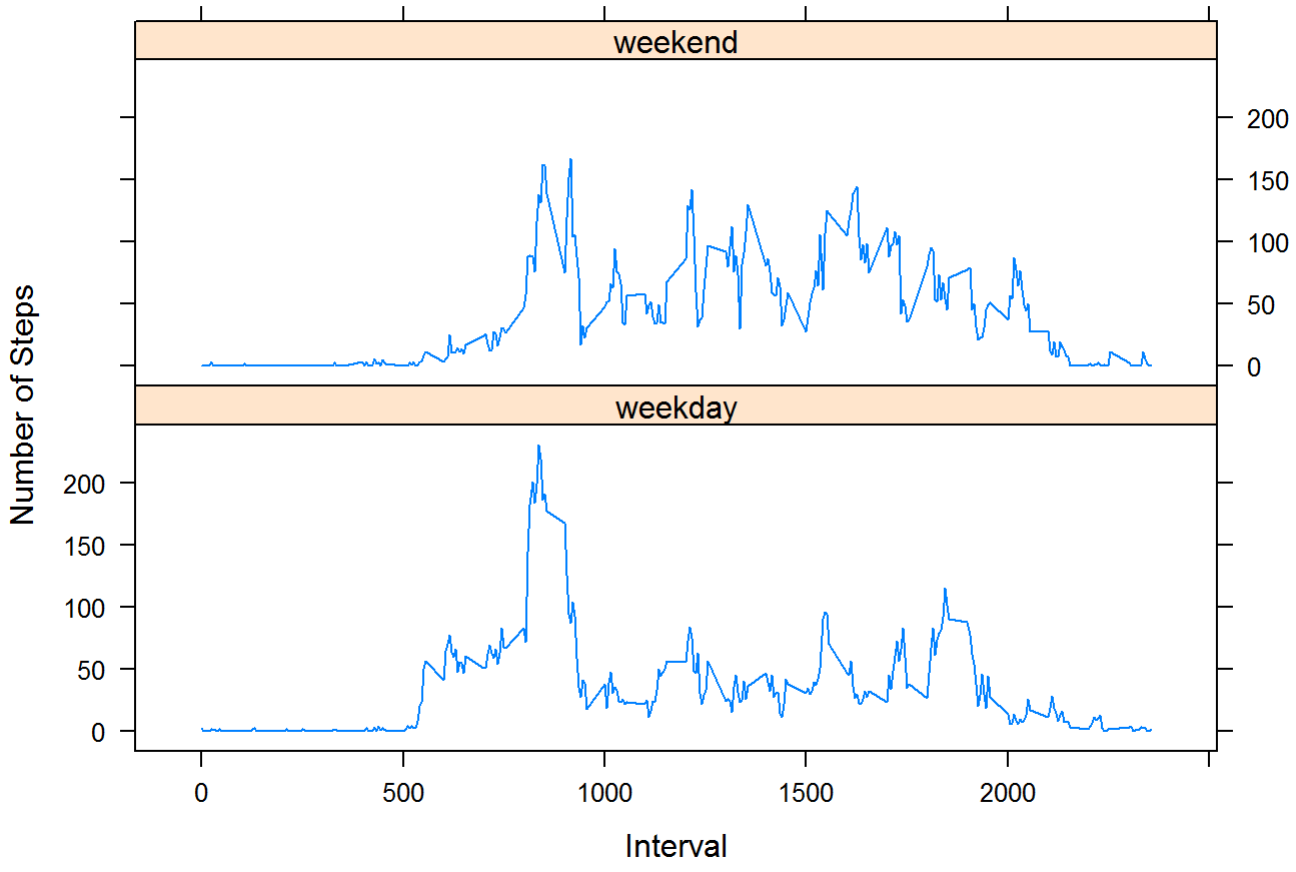
```
## [1] "weekday" "weekend"
```

```
table(newDataset$weekdays)
```

```
##
## weekday weekend
##   12960    4608
```

2. Make a panel plot containing a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
averageSteps <- aggregate(newDataset$steps,
                      list(interval = as.numeric(as.character(newDataset$interval)),
                           weekdays = newDataset$weekdays),
                      FUN = "mean")
names(averageSteps)[3] <- "meanOfSteps"
library(lattice)
xyplot(averageSteps$meanOfSteps ~ averageSteps$interval | averageSteps$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of Steps")
```



Effect of Severe Weather on Population Health and Economy

Synopsis

Severe weather can have very devastating effects on the stricken areas. It can cause loss of life in addition to shattering economic damage. This report analyzes the data of natural disasters from the U.S. National Oceanic and Atmospheric Administration's (NOAA) database that spans the time between 1950 and 2011.

Data Processing

1. Set the directory to where the Code folder of this project resides:

   setwd("C:/Users/Aiman/Box Sync/NSU/DataScience/5.Reproducible Research/Week3/Code/")

2. Extract the data from the downloaded file either programmatically or using an unzipping tool.
3. Read the data:

   stormData <- read.csv("C:/Users/Aiman/Box Sync/NSU/DataScience/5.Reproducible Research/Week3/Data/repdata_data_StormData.csv", sep = ",")

4. Fourth get the structure of the data:

   str(stormData)

   'data.frame': 902297 obs. of 37 variables:

   $ STATE__ : num 1 1 1 1 1 1 1 1 1 1 ...

   $ BGN_DATE : Factor w/ 16335 levels "1/1/1966 0:00:00",..: 6523 6523 4242 11116 2224 2224 2260 383 3980 3980 ...

   $ BGN_TIME : Factor w/ 3608 levels "00:00:00 AM",..: 272 287 2705 1683 2584 3186 242 1683 3186 3186 ...

   $ TIME_ZONE : Factor w/ 22 levels "ADT","AKS","AST",..: 7 7 7 7 7 7 7 7 7 7 ...

   $ COUNTY : num 97 3 57 89 43 77 9 123 125 57 ...

   $ COUNTYNAME: Factor w/ 29601 levels "","5NM E OF MACKINAC BRIDGE TO PRESQUE ISLE LT MI",..: 13513 1873 4598 10592 4372 10094 1973 23873 24418 4598 ...

   $ STATE : Factor w/ 72 levels "AK","AL","AM",..: 2 2 2 2 2 2 2 2 2 2 ...

   $ EVTYPE : Factor w/ 985 levels " HIGH SURF ADVISORY",..: 834 834 834 834 834 834 834 834 834 834 ...

   $ BGN_RANGE : num 0 0 0 0 0 0 0 0 0 0 ...

   $ BGN_AZI : Factor w/ 35 levels ""," N"," NW",..: 1 1 1 1 1 1 1 1 1 1 ...

   $ BGN_LOCATI: Factor w/ 54429 levels "","- 1 N Albion",..: 1 1 1 1 1 1 1 1 1 1 ...

   $ END_DATE : Factor w/ 6663 levels "","1/1/1993 0:00:00",..: 1 1 1 1 1 1 1 1 1 1 ...

```
 $ END_TIME  : Factor w/ 3647 levels ""," 0900CST",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ COUNTY_END: num  0 0 0 0 0 0 0 0 0 0 ...
 $ COUNTYENDN: logi  NA NA NA NA NA NA ...
 $ END_RANGE : num  0 0 0 0 0 0 0 0 0 0 ...
 $ END_AZI   : Factor w/ 24 levels "","E","ENE","ESE",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ END_LOCATI: Factor w/ 34506 levels "","- .5 NNW",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ LENGTH    : num  14 2 0.1 0 0 1.5 1.5 0 3.3 2.3 ...
 $ WIDTH     : num  100 150 123 100 150 177 33 33 100 100 ...
 $ F         : int  3 2 2 2 2 2 2 1 3 3 ...
 $ MAG       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ FATALITIES: num  0 0 0 0 0 0 0 0 1 0 ...
 $ INJURIES  : num  15 0 2 2 2 6 1 0 14 0 ...
 $ PROPDMG   : num  25 2.5 25 2.5 2.5 2.5 2.5 2.5 25 25 ...
 $ PROPDMGEXP: Factor w/ 19 levels "","-","?","+",..: 17 17 17 17 17 17 17 17 17 17 ...
 $ CROPDMG   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ CROPDMGEXP: Factor w/ 9 levels "","?","0","2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ WFO       : Factor w/ 542 levels ""," CI","$AC",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ STATEOFFIC: Factor w/ 250 levels "","ALABAMA, Central",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ ZONENAMES : Factor w/ 25112 levels "","                                                    "| __truncated__,..: 1 1 1 1 1
1 1 1 1 1 ...
 $ LATITUDE  : num  3040 3042 3340 3458 3412 ...
 $ LONGITUDE : num  8812 8755 8742 8626 8642 ...
 $ LATITUDE_E: num  3051 0 0 0 0 ...
 $ LONGITUDE_: num  8806 0 0 0 0 ...
 $ REMARKS   : Factor w/ 436774 levels "","-2 at Deer Park\n",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ REFNUM    : num  1 2 3 4 5 6 7 8 9 10 ...
```

5. Take a subset of the data:

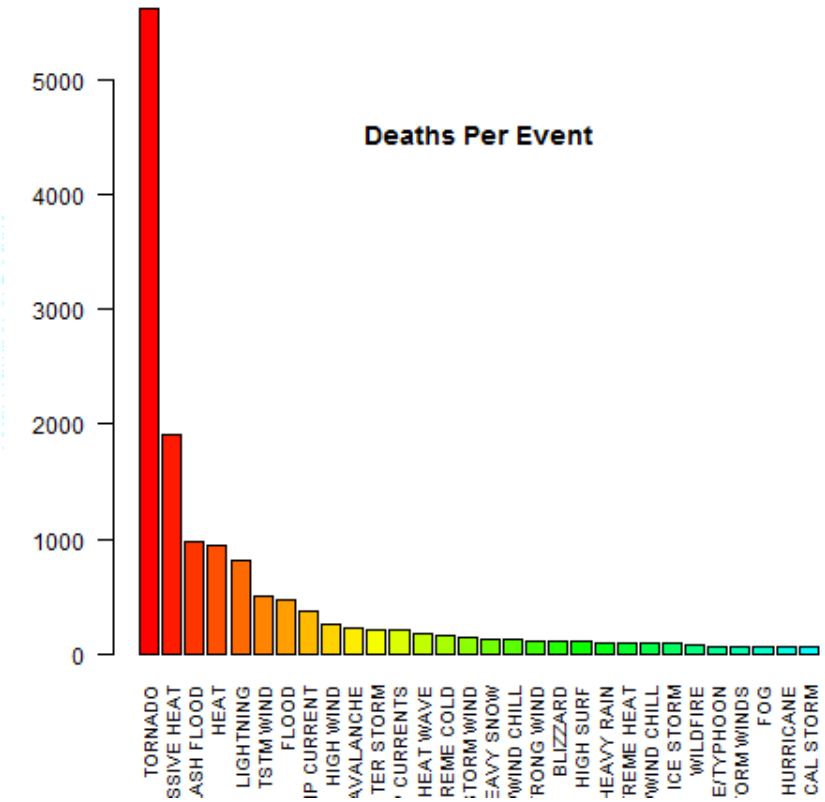
subStorm <- stormData [,c("STATE", "EVTYPE", "FATALITIES", "INJURIES", "PROPDMG", "PROPDMGEXP","CROPDMG", "CROPDMGEXP")]

Results

We will analyze the deaths, injuries, and economic losses per event from 1950 to November 2011.

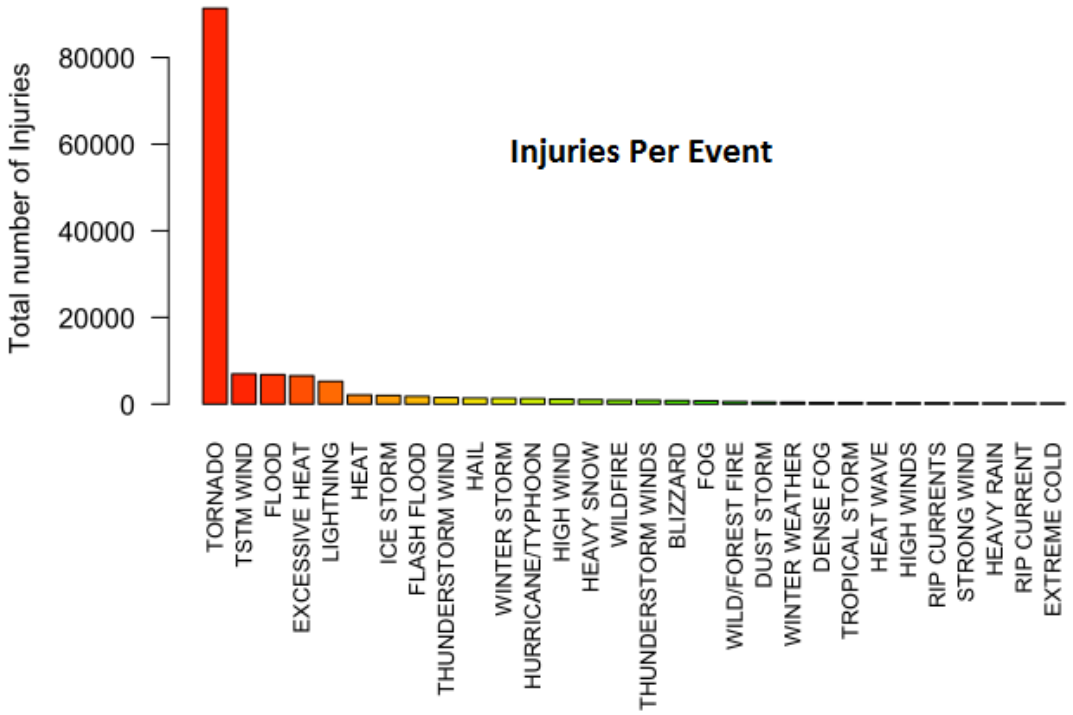
Deaths

```
deathData <- aggregate (FATALITIES~EVTYPE, subStorm, sum)

deathData <- deathData [order(deathData$FATALITIES, decreasing=TRUE),]

png(filename = "../Plots/Deaths.png")

barplot (height = deathData$FATALITIES[1:30], names.arg = deathData$EVTYPE[1:30], las = 2, cex.names= 0.8,
col = rainbow (30, start=0, end=0.5))

title (main = "Deaths Per Event", line=-5)

title (ylab = "Total Number of Deaths", line=4)
```



Injuries

```
injurData <- aggregate (INJURIES~EVTYPE, stormData, sum)

injurData <- injurData [order(injurData$INJURIES, decreasing=TRUE),]

par(mar=c(12, 6, 1, 1))

png(filename = "../Plots/Injuries.png")

barplot (height = injurData$INJURIES[1:30], names.arg = injurData$EVTYPE[1:30], las = 2, cex.names = 0.8,
col = rainbow (30, start=0, end=0.5))

title (main = "Injuries Per Event", line=-5)

title (ylab = "Total number of Injuries", line=4)
```



Damage

```
        symbol <- c("", "+", "-", "?", 0:9, "h", "H", "k", "K", "m", "M", "b", "B");
        factor <- c(rep(0,4), 0:9, 2, 2, 3, 3, 6, 6, 9, 9)
        multiplier <- data.frame (symbol, factor)
        subStorm$damage.prop <-
subStorm$PROPDMG*10^multiplier[match(subStorm$PROPDMGEXP,multiplier$symbol),2]
        subStorm$damage.crop <-
subStorm$CROPDMG*10^multiplier[match(subStorm$CROPDMGEXP,multiplier$symbol),2]
        subStorm$damage <- subStorm$damage.prop + subStorm$damage.crop
        damage <- aggregate (damage~EVTYPE, subStorm, sum);
        damage$bilion <- damage$damage / 1e9;
        damage <- damage [order(damage$bilion, decreasing=TRUE),]
        png(filename = "../Plots/Damages.png")
        barplot (height = damage$bilion[1:30], names.arg = damage$EVTYPE[1:30], las = 2, cex.names = 0.8,
    col = rainbow (30, start=0, end=0.5))
        title ("Damages Per Event", line=-5)
        title (ylab = "Total Damage In Bilion of US$")
```