Severe Weather Events in the United States: Decisive Reasons

Eduardo B. Díez

July 23, 2014

Synopsis

Using the Severe Weather Data (N=902297) from the National Oceanic and Atmospheric Administration, we examined which types of events are most harmful to population health and which have the greatest economic outcomes during the years 1950 to 2011.

Significant differences were observed between the events with greatest consequences to the population healt and to the economy.

Visually inspecting the tabulated data and graphs leads us to affirm that floods are the major causes of damage to property and crops, and tornadoes are responsible for large numbers of fatalities and injuries.

1 Introduction

This report relates to explore the NOAA [3] Storm Database looking for those weather events that are detrimental to population's health as well as those who have the greatest economic consequences.

> Few variables from the dataset were used and transformed to obtain a wide table, with which to display a couple of plots giving response to the issues this report deals. So, the output of the R-code developed and included in § Data Processing, is conformed by one table and two plots that the interested reader could find at § Results

To complete the Peer Assessment 2 as requested for the course **Reproducible Research** by Roger D. Peng, Jeff Leek and Brian Caffo was used the RStudio

markdown facilities to generate PDFs files, like this one you are reading now, directly from

RStudio with Rmarkdown with Knitr and the additional installation of MiKTEX a LaTeX flavor for Windows systems that RStudio use to create a .tex fil e LaTeX starting from a .Rmd file[1] that knit convert into a .md file, and then, pandoc convert the .md into a .tex file and finally, MiKTEX construc a PDFs output file from the .tex. All this process from inside RStudio.

Additionally, and knowing that it is not necessary, the Appendix B has relevant information about the repository from where to download the Rmarkdown file used to obtain this PDF file the reader is reading now, as well as relates to the latex template used for its construction. Thus, not only the results but also the report itself is reproducible.

2 Data

This report makes use of the Storm Data cumulative from 1950 until 2011, an official publication of the National Oceanic and Atmospheric Administration (NOAA) which documents the occurrence of storms and other significant weather phenomena having sufficient intensity to cause loss of life, injuries, significant property damage, and/or disruption to commerce. Really, we'll use a data set that has been modified slightly, with respect the official one, to make easier to work with it [1, 4, 5]

[1] See § Appendix B to get the full listing of the Rmarkdown file used to obtain this pdf. Readers should be able to reproduce this report.

In addition, it is a partial record of other significant meteorological events, such as record maximum or minimum temperatures or precipitation that occurs in connection with another event. Some information appearing in Storm Data may be provided by or gathered from sources outside the National Weather Service (NWS), such as the media, law enforcement and/or other government agencies, private companies, individuals, etc.

An effort is made to use the best available information but because of time and resource constraints, information from these sources may be unverified by the NWS. Therefore, when using information from Storm Data, customers should be cautious as the NWS does not guarantee the accuracy or validity of the information. Further, when it is apparent information appearing in Storm Data originated from a source outside the NWS (frequently credit is provided)

In order to collect the data, we have not been involved neither in the design of the survey nor in the blocking of confounders. We limited our acting ambit to use the data as they are facilitated. That's the reason we are involved in a **observational study**, based on data already collected and compiled in the NOAA database.

Any finding derived from the present study could be a *general relation* but not a *causal relation*, even being possible the existence of such causal relation, we cannot conclude it; because we are dealing with an observational study rather than a experimental one.

We summarize all the variables we will use from the original dataset:

	OriginalVariables	Type
1	BGN_DATE	Factor
2	EVTYPE	Factor
3	FATALITIES	int
4	INJURIES	int
5	PROPDMG	num
6	PROPDMGEXP	Factor
7	CROPDMG	num
8	CROPDMGEXP	Factor

Additionally, new variables are going to be created and we'll use them to construct the images and data to be displayed in this report and we summarize as follow:

	NewVariables	Type
1	BGN_YEAR	int
2	PROPEXP	num
3	PROP	num
4	CROPEXP	num
5	CROP	num

3 Data Processing

The software tool used is RStudio. The processing of the database begins with the establishment of the working environment, where the whole data was saved after being downloaded from the specified location, so it's read into the work environment R. Then, a function called multiplot [2] was defined from the begining to be prepared to construct and show a figure with a custom panel.

The transformation of the data started with the BGN_DATE original variable from which the year was extracted and added into the database as a new variable called BGN_YEAR. The process continues with the variable PROPDMGEXP, where the original values were changed by numerical values in order to be used in future as a multiplicative factor, so was saved as a new variable called

PROPEXP. We applied the same process to the variable CROPDMGEXP, and so was obtained and saved a new variable called CROPEXP. After the multiplication of both new variables by respectively PROPDMG and CROPDMG, the results were saved as PROP and CROP with values in billion of dollars and million of dollars respectively.

These last two new variables, are going to be used by summing their values grouping them by the diverse of events —all of them represented in the variable EVTYPE— therefore, the resulting data set is conformed by the new variables denominated fatalEvent{int}, injurEvent{int}, propEvent{num}, cropEvent{num} and evidently, EVTYPE{factor} too. The new data set was called **FullData** and it will be the starting point from now on.

Continuing to build a wide table that contains the most relevant, the former for health-related and the latter to economic events. In the first case by grouping the events with at least 100 victims and the last with losses of at least $ 1 million in crops. Thus, we are now ready to obtain and display the Table 1, Figure 1 and Figure 2

In order to allow a more fluent reading of this report, the complete code was moved to § Appendix A, where readers can easily copy/paste and get the table and figures mentioned above. Worth warn that the process of loading and manipulation of data is a time consuming process, leading to need several minutes to complete.

4 Results

In Table 1 we obtained a summary of the four totals grouped by events, the latter arranged for a number of victims of at least 100 individuals resulting in an understandable table without oversized.

	Event	Fatalities	Injuries	Property(B$)	Crop(M$)
1	TORNADO	5633	91346	56.95	414.95
2	EXCESSIVE HEAT	1903	6525	0.01	492.40
3	FLASH FLOOD	978	1777	16.82	1421.32
4	HEAT	937	2100	0.00	401.46
5	LIGHTNING	816	5230	0.93	12.09
6	TSTM WIND	504	6957	4.48	554.01
7	FLOOD	470	6789	144.66	5661.97
8	RIP CURRENT	368	232	0.00	0.00
9	HIGH WIND	248	1137	5.27	638.57
10	AVALANCHE	224	170	0.00	0.00
11	WINTER STORM	206	1321	6.69	26.94
12	RIP CURRENTS	204	297	0.00	0.00
13	HEAT WAVE	172	309	0.01	5.55
14	EXTREME COLD	160	231	0.07	1292.97
15	THUNDERSTORM WIND	133	1488	3.48	414.84
16	HEAVY SNOW	127	1021	0.93	134.65
17	EXTREME COLD/WIND CHILL	125	24	0.01	0.05
18	STRONG WIND	103	280	0.18	64.95
19	BLIZZARD	101	805	0.66	112.06
20	HIGH SURF	101	152	0.09	0.00

Table 1: Rearrangement, respect to the type of event, of the four values we are interested; cases of fatalities and injuries, and also, damages to properties and crop.

We note that the leading cause of health risk are tornadoes while floods are causing major damage to the economy. Prominent, among all figures, the huge amount of lost caused by floods to the property over the years that we have studied, with a total sum of roughly 145 billion dollars. We accept this amount because comes from a trusted source such as NOAA, but still, we have found it surprising.

The types of events that are most harmful to population health are show in the above figure and seems that the amount of fatalities and injuries are related —as a anecdotical case— roughly 1:10.

The last figure shows the events with the greatest economic consequences where floods are clearly the most devastating of all events both the damage they cause to crops and properties

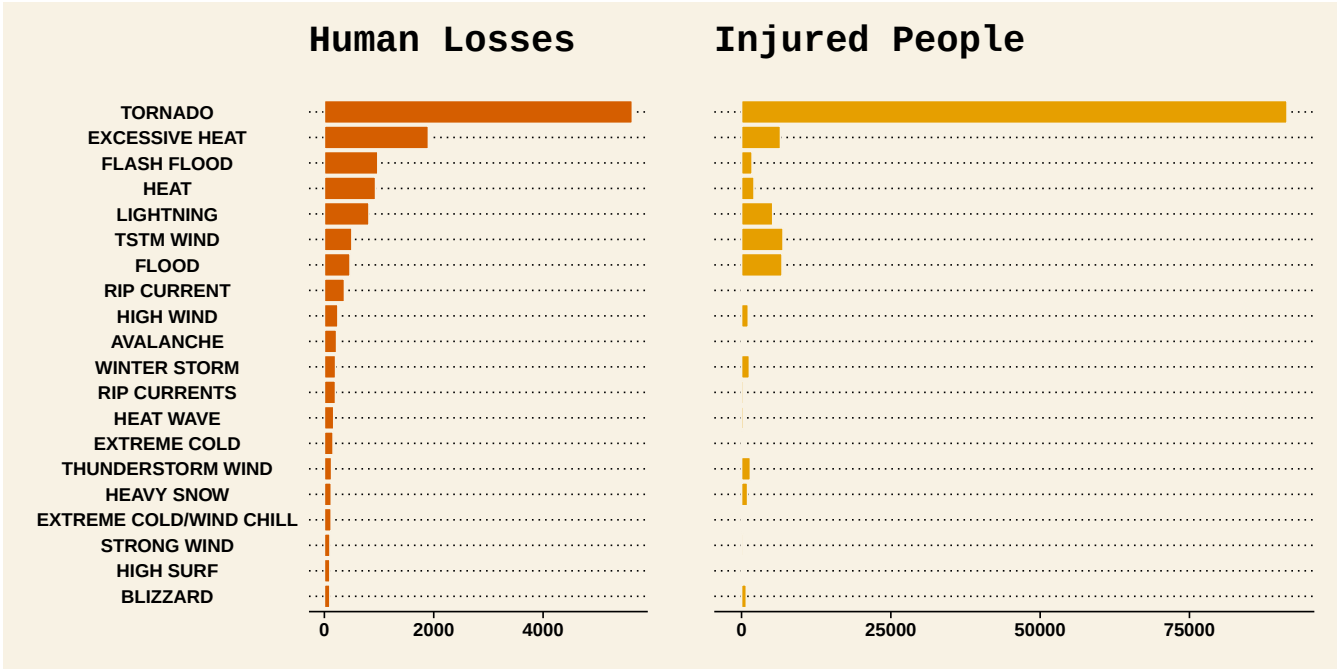

Figure 1: Events are sorted by the number of fatal cases with a minimum of 100 fatalities, showing the total number of fatalities and injuries from 1950 to 2011.

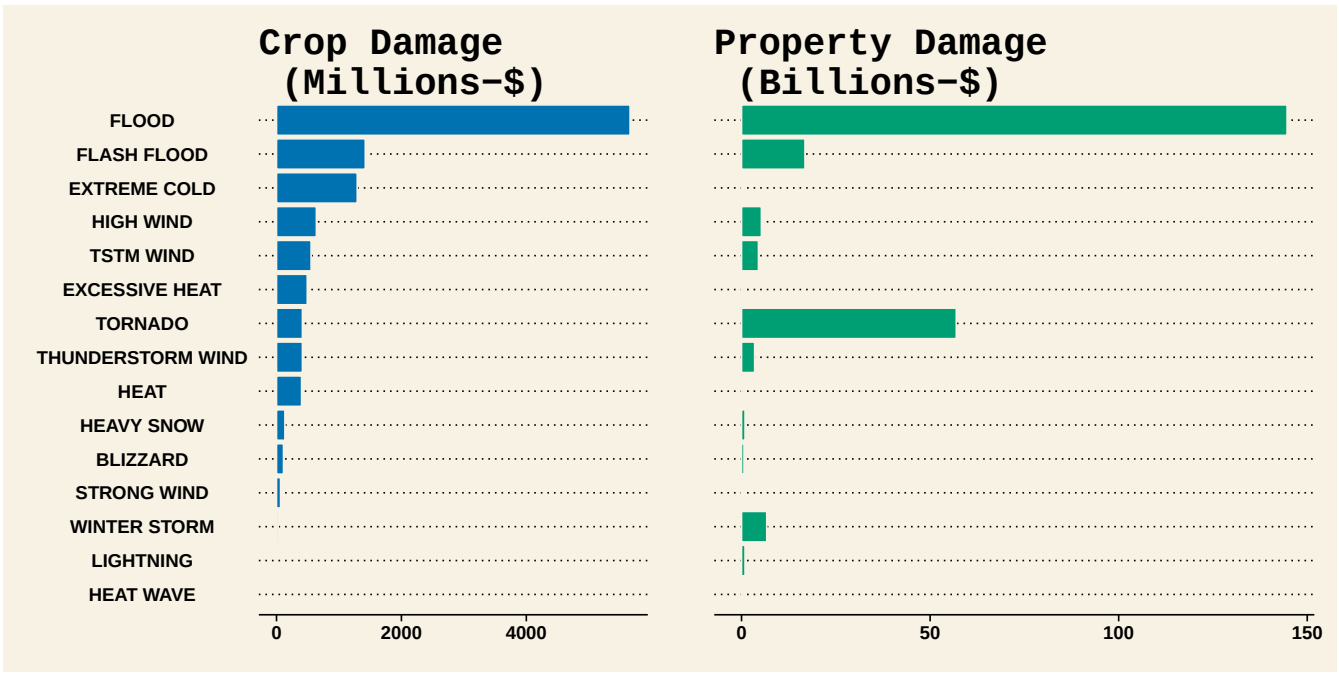

Figure 2: Events are sorted by the $ (million or billion) amount of damages in crops, with a minimum of one million dollars, showing the total sum of damage in properties and crops, from 1950 to 2011.

5 Works cited

(ACM citation style)

[1]CODEBOOK. NATIONAL WEATHER SERVICE INSTRUCTION 10-1605: 2007. *http://www.ncdc.noaa.gov/stormevents/pd01016005curr.pdf*. Accessed: 2014-07-10.

[2]Multiple Graphs On One Page (ggplot2). FUNCTION CODE OF MULTIPLOT: 2014. *http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/*. Accessed: 2014-07-17.

[3]Severe Weather Data: 2012. *http://www.ncdc.noaa.gov/data-access/severe-weather*. Accessed: 2014-07-15.

[4]Storm data for peer assessment 2 - Course: Reproducible Research. (restricted access, only for COURSERA students): 2014. *https://class.coursera.org/repdata-004/human_grading/view/courses/972143/assessments/4/submissions*. Accessed: 2014-07-10.

[5]Storm database.: 2012. *https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2*. Accessed: 2014-07-10.

6 Appendix A

The code that is reproduced below should fully play the manipulation of data, calculations and presentation of the table and the two images used in this report. The code is not intended neither to be nice nor efficient, just get the desired results by giving answer the following two questions; across the United States, which types of events are most harmful with respect to population health, and which have the greatest economic consequences.

```
# the full listing R code


# Getting & loading the data
# get the actual working directory
curdir <- getwd()

# set the pointer to the working directory where the original
# dataset is allocated. Change it to fit your particular setting
workingdirectory <- "D:/Cursos/Hopkin/5-Reproducible Research/Project 2"

# set the new working directory
setwd(workingdirectory)

# loading the dataset that is already in the working diractory
myNOAA <- read.csv("./repdatadataStormData.csv.bz2")

# add a variable with only the year, becuase maybe we need it
myNOAA <- subset(myNOAA, select=c("BGN_DATE",
                                  "EVTYPE",
                                  "FATALITIES",
                                  "INJURIES",
                                  "PROPDMG",
                                  "PROPDMGEXP",
                                  "CROPDMG",
                                  "CROPDMGEXP"))

####################
##### function multiplot
####################

# Multiple plot function
# from http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_%28ggplot2%29/
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  require(grid)

  # Make a list from the ... arguments and plotlist
```

```
plots <- c(list(...), plotlist)

numPlots = length(plots)

# If layout is NULL, then use 'cols' to determine layout
if (is.null(layout)) {
  # Make the panel
  # ncol: Number of columns of plots
  # nrow: Number of rows needed, calculated from # of cols
  layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                   ncol = cols, nrow = ceiling(numPlots/cols))
}

if (numPlots==1) {
  print(plots[[1]])

} else {
  # Set up the page
  grid.newpage()
  pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

  # Make each plot, in the correct location
  for (i in 1:numPlots) {
    # Get the i,j matrix positions of the regions that contain this subplot
    matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

    print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                    layout.pos.col = matchidx$col))
    }
  }
}

####################
##### End of function multiplot
####################

# attach the dataset so we don't need to write large names
attach(myNOAA)

# add a variable to the dataset with only the year
myNOAA$BGN_YEAR <- format(as.Date(BGN_DATE, format = "%m/%d/%Y"), "%Y")

# create a new variable same as PROPDMGEXP
myNOAA$PROPEXP <- PROPDMGEXP

# actual levels of PROPDMGEXP =
# ""  "-" "?" "+" "0" "1" "2" "3" "4" "5" "6" "7" "8" "B" "h" "H" "K" "m" "M"

# but we want to be =
niveles <-c("0", "0", "0", "0",
            "1", "10", "100", "1000", "10000",
            "100000", "1000000", "10000000", "100000000",
            "1000000000", "100", "100", "1000", "1000000", "1000000")

# so, do it
```

```r
levels(myNOAA$PROPEXP) <- niveles

# change from factor to char ...
myNOAA$PROPEXP <- as.character(myNOAA$PROPEXP)

# change from char to numeric
myNOAA$PROPEXP <- as.numeric(myNOAA$PROPEXP)

# calculate the value in billions of dollars and save it as a new variable
million=as.numeric(1000000)
billion=as.numeric(1000000000)


myNOAA$PROP <- PROPDMG * myNOAA$PROPEXP
myNOAA$PROP <- myNOAA$PROP/billion

#
# now, we do the same as the above, but now for CROPDMGEXP
#
myNOAA$CROPEXP <- CROPDMGEXP

# actual levels of PROPDMGEXP =
# ""  "?" "0" "2" "B" "k" "K" "m" "M"
# but we want  =
niveles <-c("0", "0",
            "1", "100",
            "1000000000", "1000", "1000","1000000","1000000")

# so, do it, as we did it above ...
levels(myNOAA$CROPEXP) <- niveles
myNOAA$CROPEXP <- as.character(myNOAA$CROPEXP)
myNOAA$CROPEXP <- as.numeric(myNOAA$CROPEXP)

# calculate the value in millions of dollars and save it as a new variable
myNOAA$CROP <- CROPDMG * myNOAA$CROPEXP
myNOAA$CROP <- myNOAA$CROP / million
# no more attachment
detach(myNOAA)

# goupe the data by EVTYPE summming the fatal cases
fatalEvent <- aggregate(FATALITIES ~ EVTYPE, myNOAA, sum)

# goupe the data by EVTYPE summming the injuried cases
injurEvent <- aggregate(INJURIES ~ EVTYPE, myNOAA, sum)

# merge those two data frames into health one
healthEvent <- merge(fatalEvent, injurEvent, by="EVTYPE", sort = FALSE)

# goupe the data by EVTYPE summming the property cases
propEvent <- aggregate(PROP ~ EVTYPE, myNOAA, sum)

# goupe the data by EVTYPE summming the crop cases
cropEvent <- aggregate(CROP ~ EVTYPE, myNOAA, sum)

# merge those two data frames into economic one
```

```r
economicEvent <- merge(propEvent, cropEvent, by="EVTYPE", sort = FALSE)

# merge health and economic data frames into a fulldata one
FullData <- merge(healthEvent, economicEvent, by="EVTYPE", sort = FALSE)

# save the data in the local disc
write.table(myNOAA, "./myNOAA.txt", sep="\t")
write.table(FullData, "./FullData.txt", sep="\t")


# Load the required libraries to plot and make the latex/HTML table
require(ggthemes)
require(grid)
require(gtable)
require(xtable)

# and the libraries to format the outputs from this own Rmarkdown file
# to incorporate the table printout as a latex commands to be printed as
# part of this file aoutput.
# Note, that the output could be also as a HTML format.
require(plyr)
require(knitr)

# in order to show data in table and plots, we choose the minimum cases
# for the healt affected as the sum of fatal and injuries cases.
# for this analysis the value will be 100 persons as minimum.

myDataEvent <- FullData
MinimumPeopleAfected <- 100
myDataEvent <- myDataEvent[myDataEvent$FATALITIES > MinimumPeopleAfected, ]

myDataEvent <- droplevels(myDataEvent)

# prepare to table
names(myDataEvent) <- c("Event","Fatalities", "Injuries", "Property(B$)", "Crop(M$)")

# re-order the dataframe to be tabulated and convert num into integers
# to show them in table

tmp <- arrange(myDataEvent, -Fatalities)
tmp$Fatalities <- as.integer(tmp$Fatalities)
tmp$Injuries <- as.integer(tmp$Injuries)

myCaption = 'Sed nec mi tincidunt, feugiat arcu ac, malesuada mauris. Suspendisse et scelerisque e

tmpTABLE <- tmp

myLatexTable <- print(xtable(tmpTABLE, caption=myCaption),
                      type='latex',
                      comment = FALSE)


# Load the required libraries to plot and make the latex table
require(ggthemes)
require(grid)
require(gtable)
```

```
require(xtable)

# and the libraries to format the outputs from this own Rmarkdown file
# to incorporate the table printout as a latex commands to be printed as
# part of this file aoutput.
# Note, that the output could be also as a HTML format.
require(plyr)
require(knitr)

# in order to show data in table and plots, we choose the minimum cases
# for the healt affected as the sum of fatal and injuries cases.
# for this analysis the value will be 100 persons as minimum.

myDataEvent <- FullData
MinimumPeopleAfected <- 100
myDataEvent <- myDataEvent[myDataEvent$FATALITIES > MinimumPeopleAfected, ]

myDataEvent <- droplevels(myDataEvent)

# prepare to table
names(myDataEvent) <- c("Event",
                        "Fatalities",
                        "Injuries",
                        "Property(B$)",
                        "Crop(M$)")

# re-order the dataframe to be tabulated and convert num into integers
# to show them in table

tmp <- arrange(myDataEvent, -Fatalities)
tmp$Fatalities <- as.integer(tmp$Fatalities)
tmp$Injuries <- as.integer(tmp$Injuries)

# to reproduce enterely the long sentence for the caption let's do a trick ...
myCaption <- paste("Rearrangement, respect to the type of event,",
                   "of the four values we are interested; cases of",
                   "fatalities and injuries, and also, damages",
                   "to properties and crop.",  sep = " ")

tmpTABLE <- tmp

myLatexTable <- print(xtable(tmpTABLE, caption=myCaption),
                      type='latex',
                      comment = FALSE)

# Load the required libraries to plot and make the latex/HTML table
require(ggthemes)
require(grid)
require(gtable)

# construct a friendly palette to Colorblind People as recommended by
# Color Universal Design (CUD), and add the the default gray from ggplot
# and the salmon from Wall Street Journal theme  background.
# We could also add Black = "#000000", but better not to use it.
```

```
Gray = "#999999"
Orange = "#E69F00"
SkyBlue = "#56B4E9"
BluishGreen = "#009E73"
Yellow = "#F0E442"
Blue = "#0072B2"
Vermillon = "#D55E00"
RedddishPurple = "#CC79A7"
GrayGGplot ="#E5E5E5"
SalmonWSJ = "#F8F2E4"

cbPalette <- c(Gray,
               Orange,
               SkyBlue,
               BluishGreen,
               Yellow,
               Blue,
               Vermillon,
               RedddishPurple,
               GrayGGplot,
               SalmonWSJ)

p1 <- ggplot(tmp, height=480, width=480) +
  scale_fill_manual(values=cbPalette[7]) +
  scale_colour_manual(values=cbPalette[10]) +
  aes(x=Event, y= Fatalities, fill= "manual", colour = "manual") +
  geom_bar(stat="identity") +
  coord_flip() +
  theme(legend.position="none") +
  xlab("") +
  ylab("") +
  ggtitle(paste("Human Losses","\n")) +
  theme(legend.position="none") +
  theme(axis.text.y=element_blank())

# give a look&fell like The Wall Street Journal use ...
g1 <- p1 +
  theme_wsj() +
  theme(legend.position="none")

# save the plot
png(file = "./myproject2_files/figure-latex/plot1.png",
    width = 480,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

plot(g1)

dev.off() -> trashcan

p2 <- ggplot(tmp, height=480, width=480) +
  scale_fill_manual(values=cbPalette[2]) +
```

```
  scale_colour_manual(values=cbPalette[10]) +
  aes(x=Event, y= Injuries, fill= "manual", colour = "manual") +
  geom_bar(stat="identity") +
  coord_flip() +
  theme(legend.position="none") +
  xlab("") +
  ylab("") +
  ggtitle(paste("Injured People","\n")) +
  theme(legend.position="none")

# give a look&fell like The Wall Street Journal use ...
g2 <- p2 +
  theme_wsj() +
  theme(legend.position="none", axis.text.y=element_blank())

# save the plot
png(file = "./myproject2_files/figure-latex/plot2.png",
    width = 480,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

plot(g2)

dev.off() -> trashcan
```

```
# show the plot of injured and losses human
# as figure 1.
# Note, the figure caption is in the heading of the correspondent chunk
multiplot(g1, g2, cols=2)

# save it
png(file = "./myproject2_files/figure-latex/PlotOne.png",
    width = 960,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

multiplot(g1, g2, cols=2)

dev.off() -> trashcan

MinimumCropDamage <- 1
tmp <- myDataEvent[tmp$Crop > MinimumCropDamage, ]

tmp <- droplevels(tmp)

# re-order the data with crop damages

require(plyr)
```

```
tmp$Event <- factor(tmp$Event,
                    levels= tmp[order(tmp$Crop), ]$Event)

myorder <- levels(tmp$Event)

tmp$Event <- factor(tmp$Event, levels=myorder, ordered=TRUE)

p3 <- ggplot(tmp, height=480, width=480) +
  scale_fill_manual(values=cbPalette[4]) +
  scale_colour_manual(values=cbPalette[10]) +
  aes(x=Event, y= Property, fill= "manual", colour = "manual") +
  geom_bar(stat="identity") +
  coord_flip() +
  theme(legend.position="none") +
  xlab("") +
  ylab("") +
  ggtitle(paste("Property Damage\n", "(Billions-$)")) +
  theme(legend.position="none") +
  theme(axis.text.y=element_blank())

# give a look&fell like The Wall Street Journal use ...
g3 <- p3 +
  theme_wsj() +
  theme(legend.position="none", axis.text.y=element_blank())

# save the plot
png(file = "./myproject2_files/figure-latex/plot3.png",
    width = 480,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

plot(g3)

dev.off() -> trashcan

p4 <- ggplot(tmp, height=480, width=480) +
  scale_fill_manual(values=cbPalette[6]) +
  scale_colour_manual(values=cbPalette[10]) +
  aes(x=Event, y= Crop, fill= "manual", colour = "manual") +
  geom_bar(stat="identity") +
  coord_flip() +
  theme(legend.position="none") +
  xlab("") +
  ylab("") +
  ggtitle(paste("Crop Damage\n", "(Millions-$)")) +
  theme(legend.position="none")

# give a look&fell like The Wall Street Journal use ...
g4 <- p4 +
  theme_wsj() +
  theme(legend.position="none")

# save the plot
```

```
png(file = "./myproject2_files/figure-latex/plot4.png",
    width = 480,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

plot(g4)

dev.off() -> trashcan


# show the plot of injured and losses human
# as figure 2.
# Note, the figure caption is in the heading of the correspondent chunk
multiplot(g4, g3, cols=2)

# save it
png(file = "./myproject2_files/figure-latex/PlotTwo.png",
    width = 960,
    height = 480,
    units = "px",
    pointsize = 12,
    bg = "transparent")

multiplot(g4, g3, cols=2)

dev.off() -> trashcan
```

7 Appendix B

The purpose of this addition is to provide an easy way to completely replicate the project, not only the calculations, tables and images, but also to get the final PDF file.

We start by indicating the environment and the software versions we used.

- R version 3.1.1 (2014-07-10), `i386-w64-mingw32`
- Locale: `LC_COLLATE=English_United States.1252`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: ggplot2 1.0.0, ggthemes 1.7.0, gtable 0.1.2, knitr 1.6, plyr 1.8.1, xtable 1.7-3
- Loaded via a namespace (and not attached): colorspace 1.2-4, digest 0.6.4, evaluate 0.5.5, formatR 0.10, MASS 7.3-33, munsell 0.4.2, proto 0.3-10, Rcpp 0.11.2, reshape2 1.4, rmarkdown 0.1.23, scales 0.2.4, stringr 0.6.2, tools 3.1.1, yaml 2.1.13
- RStudio Version 0.98.726

Some times there are troubles to get access to RPubs.com —as it was the case for this submission, being impossible to get upload a HTML file— so to produce a PDF that can be upload basically anyway was the decision for the submission of this project. We was alerted, already, in the instruction for the project:

> **NOTE**[4]**:** If you are having trouble connecting with RPubs due to proxy-related or other issues, you can upload your final analysis document file as a PDF to Coursera instead.

There are several files in the repository where to download all of them to inspect, run or even used discretionally.

- *default.tex* is the LaTeX template to use. It has two minor changes with respect to the original that RStudio + Knit + Rmarkdown install at origin. The first is an additional library in the preamble `\usepackage{booktabs}` to use if needed —as in our case— for the construction of LaTeX tables with the function xtable(). If reader do not desire to make this change in the default.tex can use the R code as showed for the table: `myLatexTable <- print(xtable(tmpTABLE, caption=myCaption), type='latex', comment = FALSE)` with a minimum change with respect the one showed in this report that made use of `booktabs=TRUE` as an additional option in the xtable() aformentioned and also was scaled to give a better look, resulting finally as: `myLatexTable <- print(xtable(tmpTABLE, caption=myCaption), type='latex', booktabs=TRUE, comment = FALSE, scalebox=0.75)`. The second change is relate to have the possibility to change the name of the 'abstract' to any other tittle as in our case to 'Synopsis'. That was done by a new variable for the heading of the Yaml style in the beggining of the Rmarkdown document called `abstract-title` giving the final code where the lines 162, 163 and 164 were added to the original file —again, if reader do not want to touch the original template can remove or comment the line `abstract-title: Synopsis` in the Rmarkdown heading:

```
. . .
 161 $if(abstract)$
*162 $if(abstract-title)$
```

```
*163 \renewcommand{\abstractname}{$abstract-title$}
*164 $endif$
 165 \begin{abstract}
 166 $abstract$
 167 \end{abstract}
 168 $endif$
. . .
```

- *proj2.Rmd* is the productor of this project and this file, is a long one due, basically, to the incorparation of the full list of the R-code in § Appendix A. The latter was made by copy/paste the whole code after the generation of the final .tex file from the original chunks inside the document with the `echo=TRUE` option, then, the option was changed to `echo=FALSE` in order to make a fluent writing without any chunk code in between.
  Only one manual edition was done. It's related to the references or works cited; any tool —bibliatex, natbib or the one used in this file pandoc-citedpro— usually insert the citations at the end of the documents. Latex normally need several —normally three— manual instructed compilations to get everything in place the document. With the pandoc-citepro, incorporated in the Rstudio installation with the knit and Rmarkdown libraries, the citations are edited inside the .tex file, so, we can copy and move it in any other place, for example above the first appendix, without any trouble in the compilation time of the final PDF. The resulting bibliography is not as good looking as the one obtained directly from the latex packages bibliatex or natbib but it is easy, clean and sufficiently good with the fantastic value-add to be move very easy any place inside the document.
  The bibliography items are also edited inside the own .Rmarkdown file with a pure Yaml heading style.

Those files are located in this GitHub repository and also the PDF submited and maybe some other extra of last time.