Coursera #Duration 4 weeks / July-Aug2015 Reproducible Research

Assignment2 - Storms https://class.coursera.org/repdata-031/human_grading/view/courses/975144/assessments/4/submissions (https://class.coursera.org/repdata-031/human_grading/view/courses/975144/assessments/4/submissions) Storm-events in the database start in the year 1950 and end in November 2011 NOAA Storm Database: verylarge CSV bzip2 [47Mb] #https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2 (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2) Instructions / Variable Name: #https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2Fpd01016005curr.pdf) FAQ #https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf (https://d396qusza40orc.cloudfront.net/repdata%2Fpeer2_doc%2FNCDC%20Storm%20Events-FAQ%20Page.pdf)

0.1 Instructions for Assignment

Document Layout Title: Your document should have a title that briefly summarizes your data analysis Synopsis: describes and summarizes your analysis in at most 10 complete sentences.

section titled Data Processing which describes (in words and code) how the data were loaded into R and processed for analysis. In particular, your analysis must start from the raw CSV file containing the data. You cannot do any preprocessing outside the document. If preprocessing is time-consuming you may consider using the cache = TRUE option for certain code chunks.

section titled Results in which your results are presented. At least one figure containing a plot/ no more than three figures. You must show all your code for the work in your analysis document.echo = TRUE

Submission of Output/ Code / Analysis Results: Final rendered HTML output at Rpubs: Reproducible study available at Github:

0.2 Pre-setting

```
setwd("D:/VIVEK/DataAnalysis/Coursera/5 RepRes/RepRes_Assignment2_Storms"); getwd()
```

```
## [1] "D:/VIVEK/DataAnalysis/Coursera/5 RepRes/RepRes_Assignment2_Storms"
```

```
require(stringr,  quietly = TRUE)
```

```
## Warning: package 'stringr' was built under R version 3.2.1
```

```
require(lubridate,  quietly = TRUE)
```

```
## Warning: package 'lubridate' was built under R version 3.2.1
```

```
require(reshape2,  quietly = TRUE)
```

```
## Warning: package 'reshape2' was built under R version 3.2.1
```

```
require(grid,  quietly = TRUE)
require(gridExtra,  quietly = TRUE)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.1
```

```
require(ggplot2,  quietly = TRUE)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.1
```

```
require(scales, quietly = TRUE)
```

```
## Warning: package 'scales' was built under R version 3.2.1
```

```
require(knitr,  quietly = TRUE)
```

```
## Warning: package 'knitr' was built under R version 3.2.1
```

```
require(dplyr,  quietly = TRUE)
```

```
## Warning: package 'dplyr' was built under R version 3.2.1
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
sessionInfo()
```

```
## R version 3.2.0 (2015-04-16)
## Platform: i386-w64-mingw32/i386 (32-bit)
## Running under: Windows XP (build 2600) Service Pack 2
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] dplyr_0.4.2     knitr_1.10.5    scales_0.2.5    ggplot2_1.0.1
## [5] gridExtra_2.0.0 reshape2_1.4.1  lubridate_1.3.3 stringr_1.0.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.11.6      magrittr_1.5     MASS_7.3-40      munsell_0.4.2
##  [5] colorspace_1.2-6 R6_2.1.0         plyr_1.8.3       tools_3.2.0
##  [9] parallel_3.2.0   gtable_0.1.2     DBI_0.3.1        htmltools_0.2.6
## [13] yaml_2.1.13      digest_0.6.8     assertthat_0.1   formatR_1.2
## [17] memoise_0.2.1    evaluate_0.7     rmarkdown_0.7    stringi_0.5-5
## [21] proto_0.3-10
```

```
#Version 0.99.442 - © 2009-2015 RStudio, Inc.
```

0.3 Knitr Global Options

```
opts_chunk$set(echo=TRUE, eval=TRUE, results='as.is', cache= FALSE, strip.white= TRUE, tidy = T
RUE)
```

0.4 Data Processing

0.4.1 Loading the data

Sets the paths to the dataset/ files

```
webURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
zipfilepath <- "repdata_data_StormData.csv.bz2"
rdsfilepath <- "StormData.RDS"
```

-

Checks if the file is avalable in local folder / If not downloads the zipfile

```
if (!file.exists(zipfilepath)) {
    message("wait for file to be downloaded from website-url")
    download.file(url = webURL, destfile = zipfilepath)  # not used method = 'curl'
}
```

-

Checks if the data has been uploaded/ If not extracts data from the zipfile directly If data was already uploaded/ look for RDS file [As filesize is large, cache is necessary] If available, reads file from RDS (much faster, and is the option used to knit this report)

```
RDSloaded <- FALSE
message("main data set is being loaded")
```

```
## main data set is being loaded
```

```
if (!file.exists("rdsfilepath")) {
    message("large file...do patiently wait...will take many minutes")
    mdata <- read.csv(file = bzfile(zipfilepath), strip.white = TRUE)
    saveRDS(mdata, file = "rdsfilepath")
} else {
    message("extracting from compressed file and reading dataset into environment")
    mdata <- readRDS("rdsfilepath")
    RDSloaded <- TRUE
}
```

```
## extracting from compressed file and reading dataset into environment
```

0.4.2 Explore the Dataset

Knowing the dataset and understanding the variable provides scope Build a stratergy sequence to process the dataset

```
dim(mdata)  #ncol= records #nrow = variables
```

```
## [1] 902297     37
```

```
str(mdata)
```

```
## '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/ 436781 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 ...
```

```
names(mdata)  #head(mdata, n=3)
```

```
##  [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"
```

```
# issues: large dataset, not tidy, manual record entry, arbitarery naming.
```

-

Pictoral Exploration: provides a quick visual of the large datatable. Graphical plots invisible for report/ used for understanding only.

```
invisible(plot(table(mdata$STATE__)))  #
invisible(plot(log(table(mdata$TIME_ZONE))))  #
invisible(plot(table(mdata$STATE)))  #
invisible(plot(log(table(mdata$EVTYPE))))  #
invisible(plot(log(table(mdata$FATALITIES))))  #
invisible(plot(log(table(mdata$INJURIES))))  #
invisible(table(mdata$PROPDMGEXP))  #
invisible(table(mdata$CROPDMGEXP))  #
```

0.4.3 Cleaning the data

The datset is not tidy / List_levels not defined Lot of manual entry / spelling mistakes Many fields of of Blank data Wrong / irrelevant / incomprehensible entries

0.4.4 Data Processing 1

The USD value is in two seperate columns; one with the numerical value; second the multiplier

```
message("these code-lines take a few minutes to calculate")
```

```
## these code-lines take a few minutes to calculate
```

```
multifunction <- function(unit, multiplier) unit * switch(toupper(multiplier),
    H = 100, K = 1000, M = 1e+06, B = 1e+09, 1)
mdata$propDamage <- with(mdata, mapply(multifunction, PROPDMG, PROPDMGEXP))
mdata$cropDamage <- with(mdata, mapply(multifunction, CROPDMG, CROPDMGEXP))
message("two columns added to maindata for PropertyDamage and CropDamage in numerical_USD_Amoun
t")
```

```
## two columns added to maindata for PropertyDamage and CropDamage in numerical_USD_Amount
```

-

0.4.5 Data Processing 2

convert dates to POSIXct format

```
str(mdata$BGN_DATE)
```

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

```
mdata$dateUTC <- mdy(str_extract(mdata$BGN_DATE, "[^ ]+"))
message("column added for Date as POSIXct")
```

```
## column added for Date as POSIXct
```

-

0.4.6 Data Processing 3

cleaning the EVTYPE and making Category EVTYPE column retained; new EVTYPEcat column added

```
str(mdata$EVTYPE)  #levels(mdata$EVTYPE)
```

```
##  Factor w/ 985 levels "   HIGH SURF ADVISORY",..: 834 834 834 834 834 834 834 834 834 834 ..
.
```

```
levels(mdata$EVTYPE) <- tolower(levels(mdata$EVTYPE))
mdata$EVTYPEcat <- mdata$EVTYPE
```

-

0.4.7 Data Processing 4

Important: EVTYPE categories are defined

1. “lightning” 2.“rain_shower” 3.“tornado_landwinds” 4.“typhoon_seawinds”

5.“fire & smoke” 6.“low_visibility” 7.“tsunami_oceansurge” 8.“heat_temparature”

9.“volcanic_activity” 10.“flood, erosion & avalanche” 11.“unknown condition” 12.“others”

note: there is no documentation available from the dataset authorities for this categorisation step Data Processing

```
levels(mdata$EVTYPEcat)[grepl("lightning |lighting|lightning|lignting| lightning| lignting|li
ghtning.|lighting",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "lightning"
levels(mdata$EVTYPEcat)[grepl("thunder|rain|hail|wet|downburst|precip|precipatation|shower|micr
oburst",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "rain_shower"
levels(mdata$EVTYPEcat)[grep("snow|winter|wintry|blizzard|glaze|hail|spout|sleet|cold|ice|freez
|icy|frost",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "winter_conditions"
levels(mdata$EVTYPEcat)[grepl("tornado|torndao|wnd|wind|gustnado|funnel", levels(mdata$EVTYPEca
t),
    ignore.case = T)] <- "tornado_landwinds"
levels(mdata$EVTYPEcat)[grepl("typhoon|swells|storm|hurricane|tropical +storm|turbulence",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "typhoon_seawinds"
levels(mdata$EVTYPEcat)[grepl("fire|smoke", levels(mdata$EVTYPEcat), ignore.case = T)] <- "fire
 & smoke"
levels(mdata$EVTYPEcat)[grepl("fog|visibility|dark|dust", levels(mdata$EVTYPEcat),
    ignore.case = T)] <- "low_visibility"
levels(mdata$EVTYPEcat)[grepl("marine| surf|surge|tide|tstm|tsunami|current|rough + seas|wave|d
epression| rapidly rising water| seas",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "tsunami_oceansurge"
levels(mdata$EVTYPEcat)[grepl("heat|high +temp|temperature|record +temp|warm|dry|hot",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "heat_temparature"
levels(mdata$EVTYPEcat)[grepl("volcan", levels(mdata$EVTYPEcat), ignore.case = T)] <- "volcanic
_activity"
levels(mdata$EVTYPEcat)[grepl("avalance|avalanche|flooding|fld|stream|+flood|slide|mud|dam|flas
h|landslump|erosion|erosin|rapidly rising water",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "flood, erosion & avalanche"
levels(mdata$EVTYPEcat)[grepl("summary|southeast|vog|none|northern|\\?|other|urban|small|criter
ia|apache|floyd",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "unknown condition"
levels(mdata$EVTYPEcat)[grepl("wallcloud|county|record+low|excessive|high|seiche|heavy mix|exce
ssive|no severe weather",
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "unknown condition"
levels(mdata$EVTYPEcat)[grepl("hyperthermia/exposure|drowning|unseasonal low temp|driest|record
 low|unseasonably cool|cool spell|drought|large wall cloud|record cool|mild pattern|wall cloud"
,
    levels(mdata$EVTYPEcat), ignore.case = T)] <- "others"
```

-

Important: The inital dataset had 985 factors for EVTYPE; it is now reduced to 13 categories as stored in EVTYPEcat

```
str(mdata$EVTYPEcat)
```

```
##  Factor w/ 13 levels "tsunami_oceansurge",..: 4 4 4 4 4 4 4 4 4 4 ...
```

```
levels(mdata$EVTYPEcat)
```

```
##  [1] "tsunami_oceansurge"        "flood, erosion & avalanche"
##  [3] "lightning"                 "tornado_landwinds"
##  [5] "winter_conditions"         "unknown condition"
##  [7] "heat_temparature"          "rain_shower"
##  [9] "low_visibility"            "fire & smoke"
## [11] "typhoon_seawinds"          "others"
## [13] "volcanic_activity"
```

```
table(mdata$EVTYPEcat)
```

```
##
##         tsunami_oceansurge flood, erosion & avalanche
##                       2302                      87130
##                  lightning          tornado_landwinds
##                      15780                     320425
##           winter_conditions          unknown condition
##                      50234                        195
##            heat_temparature               rain_shower
##                       3085                     412486
##              low_visibility              fire & smoke
##                       1992                       4260
##            typhoon_seawinds                     others
##                       1852                       2527
##           volcanic_activity
##                         29
```

-

0.4.8 Data Processing 4

Make dataset lean, improve speed of system: Purge many columns from the dataset, which will not be used in analysis.

```
mdata <- select(mdata, STATE__, STATE, EVTYPE, FATALITIES, INJURIES, propDamage,
    cropDamage, dateUTC, EVTYPEcat)
names(mdata)
```

```
## [1] "STATE__"    "STATE"      "EVTYPE"     "FATALITIES" "INJURIES"
## [6] "propDamage" "cropDamage" "dateUTC"    "EVTYPEcat"
```

0.5 Exploratory Analysis

Graphical plots invisible for report/ used for understanding only.

```
invisible(hist(mdata$dateUTC, breaks = 61))
invisible(plot(mdata$dateUTC, mdata$FATALITIES))
invisible(plot(mdata$dateUTC, mdata$INJURIES))
invisible(pairs(EVTYPE ~ FATALITIES + INJURIES + propDamage + cropDamage, data = mdata,
    main = "pairs plot", subset = FATALITIES > 0 & INJURIES > 0 & propDamage >
        0 & propDamage > 0))
invisible(plot(mdata$EVTYPE, mdata$propDamage))
invisible(plot(mdata$EVTYPE, mdata$cropDamage))
```

-

Understanding the Dataset; the number of events/ ratios

```
t <- nrow(mdata)  #Ans: total recorded incidents\t\t[1] 902,297

dl <- sum(mdata$FATALITIES)  #Ans: death_count \t   \t\t\t\t[1] 15,145

icd <- sum(mdata$FATALITIES > 0, na.rm = TRUE)  #Ans: fatality_incidents\t\t\t[1] 6,974

jl <- sum(mdata$INJURIES)  #Ans: injury_count\t\t\t\t\t[1] 140,528

ici <- sum(mdata$INJURIES > 0, na.rm = TRUE)  #Ans: injury_incidents \t\t\t\t[1] 17,604

pl <- sum(mdata$propDamage)  #Ans: Total Dollar Value \t\t\t[1] 427,318,652,972

icp <- sum(mdata$propDamage > 0, na.rm = TRUE)  #Ans: propertyloss_incidents \t\t[1] 239,174

cl <- sum(mdata$cropDamage)  #Ans: Total Dollar Value\t\t\t[1] 49,104,192,181

icc <- sum(mdata$cropDamage > 0, na.rm = TRUE)  #Ans: croploss_incidents  \t\t\t[1] 22,099
```

-

Ratio of number of Human_Incident to total recorderd Incidents

```
t/icd  #1 in every 130, recorded incidents has led to atleast 1 fatality
```

```
## [1] 129.3801
```

```
t/ici  #1 in every 51, recorded incidents has led to atleast 1 injury
```

```
## [1] 51.25523
```

-

Ratio of number of Damage_Incident to total recorderd Incidents

```
t/icp  #1 in every 3, recorded incidents has caused property damage
```

```
## [1] 3.772555
```

```
t/icc  #1 in every 40, recorded incidents has caused crop damage
```

```
## [1] 40.82977
```

Average per incident

```
# fatality_incident
dl/icd  #Ans: 2.17 average deaths per incident
```

```
## [1] 2.171638
```

```
# injury_incident
jl/ici  #Ans: 7.98 average persons injured per incident
```

```
## [1] 7.982731
```

```
# propertyloss_incident
pl/icp  #Ans: $1,786,643 average loss per incident
```

```
## [1] 1786643
```

```
# croploss_incident
cl/icc  #Ans: $2,222,010 average loss per incident
```

```
## [1] 2222010
```

caution note: for below code block, group_by does not work if 'plyr'is loaded after 'dplyr search() #display the packages detach("package:dplyr", unload=TRUE) #unload the package

0.6 Analysis:

0.6.1 Which categories are most harmful?

Impact on Human Life (Fatalities and Injuries)

0.6.1.1 Analysis Code 1

Analysing the effect of Event-type-categories on human_incidents (deaths and injuries)

```
harmful <- filter(mdata, FATALITIES > 0 & INJURIES > 0)
harmful <- dplyr::group_by(harmful, EVTYPEcat)
harmful <- dplyr::summarise(harmful, FATALITIES = sum(FATALITIES, na.rm = T),
    INJURIES = sum(INJURIES, na.rm = T))
harmful <- dplyr::arrange(harmful, desc(FATALITIES + INJURIES))
names(harmful) <- c("EVTYPEcat", "human_deaths", "human_injuries")
meltharmful <- melt(harmful[1:4, ], id.vars = "EVTYPEcat", measure.vars = c("human_deaths",
    "human_injuries"))
```

-

0.6.1.2 Presentation Code 1

The ggplot grapplots only the top four EVTYPEcat / as the rest comparitively have less impact

```
plot1 <- ggplot(data = meltharmful, aes(x = reorder(EVTYPEcat, desc(value)),
    y = value/10^3, fill = variable)) + geom_bar(stat = "identity", width = 0.8,
    position = "dodge") + theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
    ylab("Number of Person  ( in Thousand)") + xlab("Major Event Category") +
    labs(fill = "Incidents") + theme(legend.position = c(0.8, 0.8), legend.background = element
_rect(fill = "transparent")) +
    ggtitle("Top 4 Weather Events | Dangerous for Human Life") + theme(plot.title = element_tex
t(size = 12,
    lineheight = 0.6, face = "bold")) + geom_text(aes(label = value), size = 3,
    hjust = 1.2, vjust = -0.8, alpha = 0.4)
```

- ### 0.6.1.2 Record the year of events, for the top categories - harmful incident.

0.6.1.3 Analysis Code 2

Subset the primary cleandataset for the four category variables

```
harmloc <- rbind(filter(mdata, EVTYPEcat == "tornado_landwinds"), filter(mdata,
    EVTYPEcat == "heat_temparature"), filter(mdata, EVTYPEcat == "winter_conditions"),
    filter(mdata, EVTYPEcat == "flood, erosion & avalanche")) %>% droplevels()
```

-

0.6.1.4 Presentation Code 2

The ggplot plots both deaths and injuries

```
plot2 <- ggplot(harmloc, aes(x = dateUTC, y = INJURIES, group = EVTYPEcat)) +
    geom_line(aes(stat = "identity", colour = "human_injuries"), alpha = 0.2) +
    geom_line(data = harmloc, aes(y = FATALITIES, stat = "identity", colour = "human_deaths"),
        alpha = 0.5) + scale_y_continuous(limits = c(0, 400), oob = rescale_none) +
    facet_wrap(~EVTYPEcat, nrow = 1) + coord_flip() + xlab("time-line.years 1950-2011") +
    ylab("quantity of casuality") + ggtitle("Timeline: Annual Casuality Incidents. EVTYPE by Pa
nel - for the Top 4 Weather Events - dangerous for humanlife") +
    theme(plot.title = element_text(size = 10, lineheight = 0.6, face = "bold")) +
    theme(legend.position = c(0.9, 0.3), legend.background = element_rect(fill = "transparent")
) +
    theme(legend.title = element_text(colour = "darkgrey", size = 10, face = "bold")) +
    scale_color_discrete(name = "Incidents")
```

-

0.6.2 Which categories are most destructive?

Economic Consequences (Property and Crop Damage)

0.6.2.1 Analysis Code 3

```
damage <- filter(mdata, propDamage > 0 & propDamage > 0)
damage <- dplyr::group_by(damage, EVTYPEcat)
damage <- dplyr::summarise(damage, propDamage = sum(propDamage, na.rm = T),
    cropDamage = sum(cropDamage, na.rm = T))
damage <- dplyr::arrange(damage, desc(propDamage + cropDamage))
meltdamage <- melt(damage[1:5, ], id.vars = "EVTYPEcat", measure.vars = c("propDamage",
    "cropDamage"))
```

-

0.6.2.2 Presentation Code 3

The ggplot grapplots only the top five EVTYPEcat / as the rest comparitively have less impact

```
plot3 <- ggplot(data = meltdamage, aes(x = reorder(EVTYPEcat, desc(value)),
    y = value/10^9, fill = variable)) + geom_bar(stat = "identity", width = 0.8,
    position = "dodge") + ylab("Financial Measure: Loss in $ (Billion)") + xlab("Major Event Ca
tegory") +
    labs(fill = "Legend") + theme(legend.position = c(0.8, 0.8), legend.background = element_re
ct(fill = "transparent")) +
    ggtitle("Top 5 Weather Events | causing Economic Damage") + theme(plot.title = element_text
(size = 12,
    lineheight = 0.6, face = "bold")) + geom_text(aes(label = round(value/10^9)),
    size = 3, hjust = 1.1, vjust = -0.8, alpha = 0.4) + theme(axis.text.x = element_text(angle
= 60,
    hjust = 1))
```

0.7 Results

0.7.1 Combined Plots in single Graphical Output

Setting up the multiplot function
http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_%28ggplot2%29/ (http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_%28ggplot2%29/)

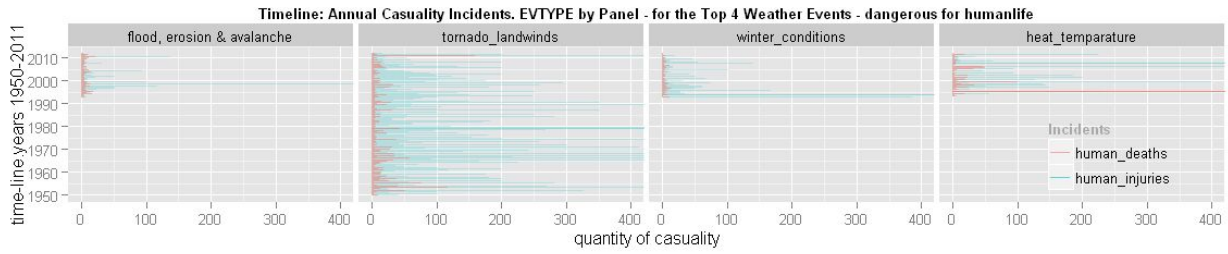
```
# Multiple plot function ggplot objects can be passed in ..., or to plotlist
# (as a list of ggplot objects) - cols: Number of columns in layout -
# layout: A matrix specifying the layout. If present, 'cols' is ignored.  If
# the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then
# 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) {
    library(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$row, layout.pos.col = mat
chidx$col))
        }
    }
}
```

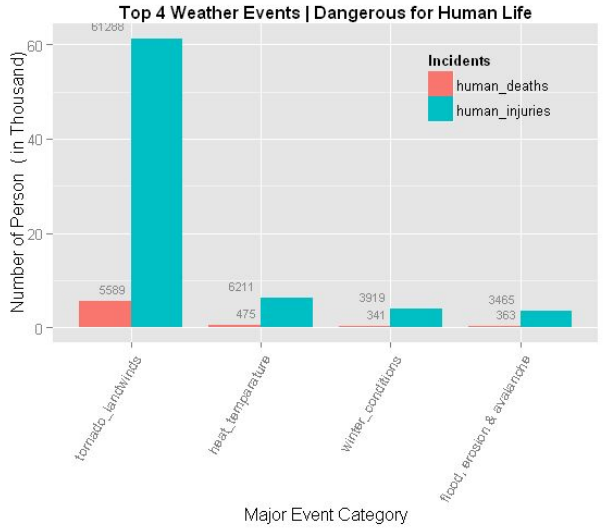
-

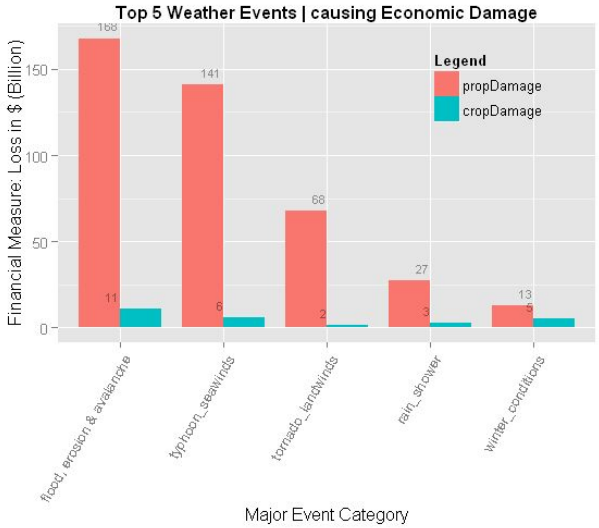
0.7.1.1 Presentation Code 3

Analysis of data in visible form

```
grid.newpage()
pushViewport(viewport(layout = grid.layout(3, 2)))
vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)
print(plot2, vp = vplayout(1, 1:2))  # key is to define vplayout
print(plot1, vp = vplayout(2:3, 1))
print(plot3, vp = vplayout(2:3, 2))
```








```
message("Graphical Output: Single figure with Three plots and multi-pane")
```

```
## Graphical Output: Single figure with Three plots and multi-pane
```

```
message("This meets the assignemtnt instructions to have less than three plots in the report")
```

```
## This meets the assignemtnt instructions to have less than three plots in the report
```

-

0.7.2 Q1. and answer

Which types of events (as indicated in the EVTYPE variable) are most harmful with respect to population health?[Across the United States]

Ans1: From the output, we find the harmful for humans (fatalities + injuries) 'Top Five Categories' are 1. tornado_landwinds 2. heat_temparature 3. winter_conditions 4. flood, erosion & avalanche The harm-impact of the other 9 categories is comparitively negligible, hence not plotted. -

0.7.3 Q2. and answer

Which types of events have the greatest economic consequences?[Across the United States]

Ans2: From the output, we find the destructive (propDamage + cropDamage) 'Top Five Categories' are 1.tornado_landwinds 2.flood, 3.erosion & rain_shower 4.winter_conditions 5. fire&smoke The harm-impact of the other 10 categories is comparitively negligible, hence not plotted. -

Note: Analysing for the Categories, there is a key-obervation to be made. Records for tornado_landwinds available since 1950 | for the others there is minimal records and mostly not available till 1996

0.8 End of Report.