Hurricane Analysis and Visulization Using R

Romane Goldmuntz, Vy Tran, and Jianqiong Zhan

2019-11-07

Contents

Chapter 1

Preface

This is a class project written in **Markdown**. We are still working on it.

We are using the **bookdown** package (Xie, 2019) in this project, which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 2

Introduction

As coastal shoreline counties create about 40% of the United States's jobs and account for 46% of its GDP, hurricanes have a trumendous impact on the country's economy.

They are considered as one of the costliest natural disasters in the world : they currently cost the government over $28 billion each year, and that amount is expected to increase to over $39 billion a year due to the increased development of the U.S. coastlines and the global warming. The latter will indeed increase the proportion of cyclones of category 4 and 5, which lead to the most damages and therefore higher costs (Amadeo, 2019).

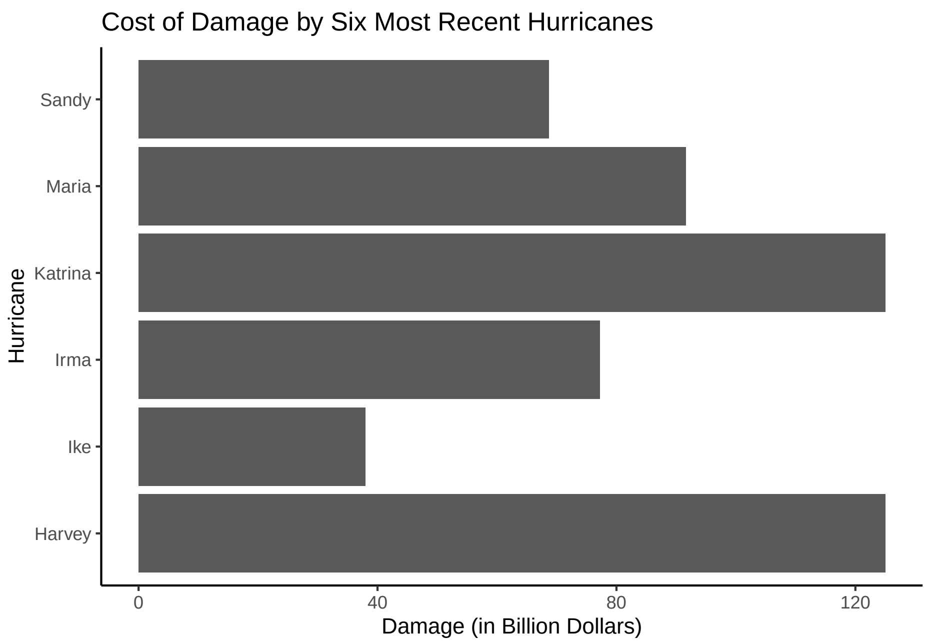
Besides the government, several industries are heavily impacted by hurricanes, including the insurance industry. For example, according to Bloomberg, hurricane Dorian caused the insurance industry losses of up $25 billion, making it the most expensive natural disaster for the industry since 2017's Hurricane Maria (D'Souza, 2019).

```
library(rvest)
library(dplyr)
library(robotstxt)
library(ggplot2)
url <- "https://en.wikipedia.org/wiki/List_of_costliest_Atlantic_hurricanes"
#paths_allowed(url)

df<-as.data.frame(read_html(url) %>% html_table(fill = TRUE))

df_clean <- df %>% mutate(Nominal_Damage = as.factor(gsub("[$><]", "",Nominal.damage.Billions.USD
  select(Name, Season, Storm.classificationat.peak.intensity,Nominal_Damage) %>% rename(Classific
df_clean$Season<-as.factor(df_clean$Season)
as.numeric.factor <- function(x) {as.numeric(levels(x))[x]}
```

```
df_clean[4]<-lapply(df_clean[4],as.numeric.factor)
df_clean[2]<-lapply(df_clean[2],as.numeric.factor)
df_clean <- df_clean %>% arrange(desc(Season)) %>% top_n(6)
ggplot(df_clean,aes(x=Name, y= Nominal_Damage)) + geom_bar(position = "dodge", stat = '
```



In addition, hurricane tracking data can provide Federal Emergency Management Agency (FEMA), local emergency managers, and first responders the information they need to be able to send out appropriate responses and help to the citizens at the affected areas (GOES-r, 2019).

For those reasons, hurricanes data is very interesting to analyze and will constitute the topic of this Exploratory Data Analysis and Vizualisation final project.

Chapter 3

Methods

3.1 Data sources

(We describe our data sources, our methods in this chapter)

Storm tracks data can be downloaded from National Hurricane Center and Central Pacific Hurricane Center. The data using in the project is known as Atlantic hurricane database (HURDAT2) 1851-2018 (5.9MB download). The data has a comma-delimited, text format with six-hourly information on the location, maximum winds, central pressure, and (beginning in 2004) size of all known tropical cyclones and subtropical cyclones.

3.2 Data transformat

Describe the process of getting the data into a form in which you could work with it in R.

```
library(tidyverse)
library(stringr)
# Read in data set
#dfile <- read_lines("https://www.nhc.noaa.gov/data/hurdat/hurdat2-1851-2018-051019.txt")
dfile<- "data/hurdat2-1851-2018-051019.txt"
storm_strings <- read_lines(dfile)
```

```
# Identify the header lines that have three commas
library(stringr)
header_locations <- (1:length(storm_strings))[str_count(storm_strings, "\\,") == 3] # count # of
```

```r
headers <- as.list(storm_strings[header_locations])
headers_df <-
  headers %>%
  purrr::map(stringr::str_sub, start = 1L, end = -2L) %>% # to remove trailing comma
  purrr::map(paste0, "\n") %>%                # and replace final comma with \n
  purrr::map_dfr(read_csv, col_names = c("id", "name", "n_obs")) %>%
  dplyr::mutate(#name = dplyr::recode(name, "UNNAMED" = id),
                skip = header_locations) %>%
  dplyr::select(id, name, skip, n_obs)
```

```r
# Set data frames names
df_names <- c(
  "date", "time", "record_type", "status", "lat", "long", "wind", "pressure",
  "extent_34_NE", "extent_34_SE", "extent_34_SW", "extent_34_NW",
  "extent_50_NE", "extent_50_SE", "extent_50_SW", "extent_50_NW",
  "extent_64_NE", "extent_64_SE", "extent_64_SW", "extent_64_NW", "nas"
)

storm_dfs <- vector("list", nrow(headers_df))
names(storm_dfs) <- headers_df$id
```

```r
# Read in the sub-datasets as data frames
for (i in seq_along(headers_df$name)) {
  storm_dfs[[i]] <- read_csv(dfile,
    skip = headers_df$skip[i],
    n_max = headers_df$n_obs[i],
    col_names = df_names,
    na = c("", "-99", "-999"),
    col_types = list(
      time = col_character(),
      pressure = col_integer(),
      extent_34_NE = col_integer(),
      extent_34_SE = col_integer(),
      extent_34_SW = col_integer(),
      extent_34_NW = col_integer(),
      extent_50_NE = col_integer(),
      extent_50_SE = col_integer(),
      extent_50_SW = col_integer(),
      extent_50_NW = col_integer(),
      extent_64_NE = col_integer(),
      extent_64_SE = col_integer(),
      extent_64_SW = col_integer(),
      extent_64_NW = col_integer()
    )
  )
```

```
}
```

```
# Combine and clean the data sets
library(lubridate)
storms <-
  storm_dfs %>%
  dplyr::bind_rows(.id = "id") %>%
  dplyr::mutate(
    date = lubridate::ymd(date),
    year = lubridate::year(date),
    month = lubridate::month(date),
    day = lubridate::day(date),
    hour = as.numeric(stringr::str_sub(time, 1, 2)),
    datetime = ISOdate(year, month, day, hour, min = 0, sec = 0, tz = "GMT"),
    lat_hemisphere = stringr::str_sub(lat, -1),
    lat_sign = dplyr::if_else(lat_hemisphere == "N", 1, -1),
    lat = as.numeric(stringr::str_sub(lat, 1, -2)) * lat_sign,
    long_hemisphere = stringr::str_sub(long, -1),
    long_sign = dplyr::if_else(long_hemisphere == "E", 1, -1),
    long = as.numeric(stringr::str_sub(long, 1, -2)) * long_sign,
    category = cut(wind, #cut divides the range of x into intervals
      breaks = c(0, 34, 64, 83, 96, 113, 137, 500),
      labels = c(-1, 0, 1, 2, 3, 4, 5),
      include.lowest = TRUE, ordered = TRUE
    ),
    # wind = wind * 1.15078, # transforms knots to mph,
    TSradius1 = extent_34_NE + extent_34_SW,
    TSradius2 = extent_34_NW + extent_34_SE,
    ts_diameter = pmax(TSradius1, TSradius2) * 1.15078, # to convert from nautical miles to miles
    HUradius1 = extent_64_NE + extent_64_SW,
    HUradius2 = extent_64_NW + extent_64_SE,
    hu_diameter = pmax(HUradius1, HUradius2) * 1.15078, # to convert from nautical miles to miles
    status = recode(status,
                    "TD" = "tropical depression",
                    "TS" = "tropical storm",
                    "HU" = "tropical hurricane",
                    "EX" = "Extratropical cyclone", ##
                    "SD" = "subtropical depression",
                    "SS" = "subtropical storm",
                    "HU" = "tropical hurricane",
                    "LO" = "a low",
                    "WV" = "tropical wave",
                    "DB" = "disturbance")
  )
```

```r
# data output
headers_df_selected <- headers_df %>% select(c("id","name"))
# headers_df_selected
storms_with_name <- left_join(headers_df_selected, storms, by=c("id")) %>%
  select(id, name, datetime, year, month, day, hour, lat, long, status, category,
         wind, pressure,
         extent_34_NE, extent_34_SW, extent_34_NW, extent_34_SE,
         extent_50_NE, extent_50_SW, extent_50_NW, extent_50_SE,
         extent_64_NE, extent_64_SW, extent_64_NW, extent_64_SE,
         ts_diameter, hu_diameter)
```

```r
storms_all <- storms_with_name %>%
  mutate(name= dplyr::if_else(grepl("UNNAMED", name), name,
                              stringr::str_to_title(name)))
```

```r
storms_all$status <- factor(storms_all$status)
storms_all$id <- factor(storms_all$id)
storms_all$name <- factor(storms_all$name)
```

```r
levels(storms_all$status)
```

```
##  [1] "a low"                  "disturbance"            "ET"
##  [4] "Extratropical cyclone"  "subtropical depression" "subtropical storm"
##  [7] "tropical depression"    "tropical hurricane"     "tropical storm"
## [10] "tropical wave"
```

```r
storms_all %>% filter(status == "ET")
```

```
## # A tibble: 1 x 27
##   id    name  datetime             year month   day  hour   lat  long status
##   <fct> <fct> <dttm>              <dbl> <dbl> <int> <dbl> <dbl> <dbl> <fct>
## 1 AL09~ Harv~ 1993-09-21 18:00:00  1993     9    21    18    46   -42 ET
## # ... with 17 more variables: category <ord>, wind <dbl>, pressure <int>,
## #   extent_34_NE <int>, extent_34_SW <int>, extent_34_NW <int>,
## #   extent_34_SE <int>, extent_50_NE <int>, extent_50_SW <int>,
## #   extent_50_NW <int>, extent_50_SE <int>, extent_64_NE <int>,
## #   extent_64_SW <int>, extent_64_NW <int>, extent_64_SE <int>,
## #   ts_diameter <dbl>, hu_diameter <dbl>
```

Note: there is an "ET" in Status of system*, however, there is no description in HURDAT2, which is a typo in the dataset,* `recode` *it into 'EX".*

```r
storms_all$status <- dplyr::recode(storms_all$status, ET = "Extratropical cyclone")
#levels(factor(storms_all$status))
```

Narrow to storms that have complete pressure record

```r
# Narrow to storms that have complete pressure record
completeish <- storms_all %>%
  dplyr::group_by(id) %>%
  dplyr::summarise(n_pressure = sum(!is.na(pressure)), p_pressure = mean(!is.na(pressure))) %>%
  dplyr::filter(p_pressure == 1) %>%
  .[["id"]]
#length(completeish)
dim(storms_all)[1]
```

```
## [1] 50911
```

Theare are 562 out of 50911storms that have complete pressure record.

```r
storms_completish_selected <- storms_all %>%
  filter(
    status %in% c("hurricane", "tropical storm", "tropical depression"),
    id %in% completeish)
```

```r
storms_all_add_com <- storms_all %>% mutate(completeish = if_else(status %in% c("hurricane", "tro
storms_all_add_com$completeish <- factor(storms_all_add_com$completeish)
```

Save transformed data

```r
dir <- 'data/'
write_csv(storms_all_add_com, file.path(dir, "storms_all_out.csv"))
```

```r
dir <- 'data/'
write_csv(storms, file.path(dir, "storms_completish_selected.csv"))
```

3.3 Missing values

Describe any patterns you discover in missing values.

Chapter 4

Results

Provide a short nontechnical but *significant* summary of the most revealing findings of our analysis written for a nontechnical audience. Take extra care to clean up our graphs, ensuring that best practices for presentation are followed, as described in the audience ready style section below.

Chapter 5

Interactive Component

Select one (or more) of our key findings to present in an interactive format (D3). Be selective in the choices that we present to the user; the idea is that in 5-10 minutes, users should have a good sense of the question(s) that we are interested in and the trends we've identified in the data. In other words, they should understand the value of the analysis, be it business value, scientific value, general knowledge, etc.

Chapter 6

Conclusion

Discuss limitations and future directions, lessons learned.

Bibliography

Amadeo, K. (2019). *How Florence, Harvey, Maria, and Other Hurricanes Battered the Economy.* The Balance.

D'Souza, D. (2019). *Hurricane Dorian: Measuring the Economic Impact.* Investopedia.

GOES-r (2019). *New Tools for Monitoring Hurricanes in a Changing Climate.*

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2019). *bookdown: Authoring Books and Technical Documents with R Markdown.* R package version 0.14.