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).

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1.

```
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
```

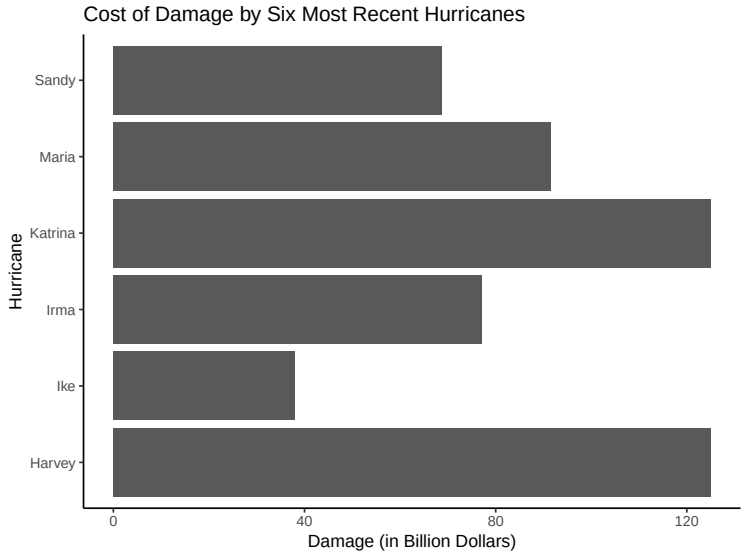

Figure 2.1: Here is a nice figure!

```
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.)

** load libraries **

```
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"
hurdat2_in <- read_lines(dfile)
```

The dataset is a combination of serveral subsets. Eeach subset is for a storm track record which incluedes header information and values.

For instance, the header has the following format:

AL162018, OSCAR, 36, * AL – Basin – Atlantic * 16 – ATCF cyclone number for that year * 2018 – Year * OSCAR – Name, if available, or else "UNNAMED" * 36 – Number of best track entries – rows – to follow

The header is followed by data, which has the following format:

20181026, 1800, , SS, 25.4N, 45.3W, 35, 1006, 80, 80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

- 2018 – Year
- 10 – Month
- 26 – Day
- 09 – Hours in UTC (Universal Time Coordinate) 35 (Spaces 13-14, before 2nd comma) – Minutes ...

please refer this file for detail information.

```
header_locations <- (1:length(hurdat2_in))[str_count(hurdat2_in, "\\,") == 3]
header_df <-readr::read_csv(hurdat2_in[header_locations],
                            col_names = FALSE) %>%
  dplyr::select(-c("X4"))
names(header_df) <- c("id","name","n_entries")
```

```
# read hearder information
library(stringr)
header_locations <- (1:length(hurdat2_in))[stringr::str_count(hurdat2_in, "\\,") == 3]

header_df <- readr::read_csv(hurdat2_in[header_locations],
                            col_names = FALSE) %>%
  dplyr::select(-c("X4"))

names(header_df) <- c("id","name","n_entries")

header_df <-  header_df %>% mutate(header_loc = as.numeric(header_locations))

#tail(header_df)
```

```
# read data value

hurdat2_df <- vector("list", nrow(header_df))
names(hurdat2_df) <- header_df$id
df_names <- c(
  "date", "time", "record_identifier", "status", "latitude", "longitude", "max_wind",
  "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"
)

#
for (i in seq_along(header_df$id)) {
  hurdat2_df[[i]] <- read_csv(dfile,
    skip = header_df$header_loc[i],
    n_max = header_df$n_entries[i],
    col_names = df_names,
    na = c("", "-99", "-999"),
    col_types = list(
      time = col_character(),
      min_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()
    )
  )
}
```

```
#head(hurdat2_df)
```

```
# Combine and clean the data sets
library(lubridate)
hurdat2 <-
  hurdat2_df %>%
  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 = as.Date(ISOdate(year, month, day, hour, min = 0, sec = 0, tz = "GMT")),
    #lat_hemisphere = stringr::str_sub(latitude, -1),
    latitude = dplyr::if_else(stringr::str_sub(latitude, -1) == "N",
```

```
                              as.numeric(stringr::str_sub(latitude, 1, -2))*1,
                              as.numeric(stringr::str_sub(latitude, 1, -2))*(-1)),
    longitude = dplyr::if_else(stringr::str_sub(longitude, -1) == "E",
                              as.numeric(stringr::str_sub(longitude, 1, -2))*1,
                              as.numeric(stringr::str_sub(longitude, 1, -2))*(-1)),
    category = cut(max_wind, # Saffir-Simpson Hurricane Wind Scale
      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 mil
    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 mil
    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")
  )
```

Note: category has been calculated based on Saffir-Simpson Hurricane Wind Scale to indicate "Types of Damage Due to Hurricane Winds".

```
# absorb header information to data values
header_df_selected <- header_df %>% select(c("id","name"))
# headers_df_selected
hurdat2_add_name <- left_join(header_df_selected, hurdat2, by=c("id")) %>%
  select(id, name, datetime, year, month, day, hour, latitude, longitude, status, categ
         max_wind, min_pressure, ts_diameter, hu_diameter)
```

```
hurdat2_out <- hurdat2_add_name %>%
  mutate(name= dplyr::if_else(grepl("UNNAMED", name), name,
                              stringr::str_to_title(name)))
hurdat2_out$status <- factor(hurdat2_out$status)
hurdat2_out$category <- factor(hurdat2_out$category)
```

```
levels(hurdat2_out$status)
```

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

```
hurdat2_out %>% filter(status == "ET")
```

```
## # A tibble: 1 x 15
##   id    name  datetime    year month   day  hour latitude longitude status
##   <chr> <chr> <date>     <dbl> <dbl> <int> <dbl>    <dbl>     <dbl> <fct>
## 1 AL09~ Harv~ 1993-09-21  1993     9    21    18       46       -42 ET
## # ... with 5 more variables: category <ord>, max_wind <dbl>,
## #   min_pressure <int>, ts_diameter <dbl>, hu_diameter <dbl>
```

Note: there is an "ET" in Status of system, *which does not included in the description HURDAT2. This is a typo in the dataset,* `recode` *it into 'EX".*

```
hurdat2_out$status <- dplyr::recode(hurdat2_out$status, ET = "Extratropical cyclone")
```

** Mark storms that have complete pressure record**

```
completeish <- hurdat2_out %>%
  dplyr::group_by(id) %>%
  dplyr::summarise(n_pressure = sum(!is.na(min_pressure)), p_pressure = mean(!is.na(min_pressure)
  dplyr::filter(p_pressure == 1) %>%
  .[["id"]]
#length(completeish)
#dim(hurdat2_out)[1]
```

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

```
storms_completish_selected <- hurdat2_out %>%
  filter(
    status %in% c("hurricane", "tropical storm", "tropical depression"),
    id %in% completeish)
```

```
hurdat2_out_add_com <- hurdat2_out %>% mutate(completeish = if_else(status %in% c("hurricane", "t
hurdat2_out_add_com$completeish <- factor(hurdat2_out_add_com$completeish)
```

Meaning for each variables

```r
names(hurdat2_out_add_com)
```

```
##  [1] "id"           "name"         "datetime"     "year"         "month"
##  [6] "day"          "hour"         "latitude"     "longitude"    "status"
## [11] "category"     "max_wind"     "min_pressure" "ts_diameter"  "hu_diameter"
## [16] "completeish"
```

id

Storm id, which is unique. A id is a combination of 8 characters, for example, 'AL092011', * AL (Spaces 1 and 2) – Basin – Atlantic * 09 (Spaces 3 and 4) – ATCF cyclone number for that year * 2011 (Spaces 5-8, before first comma) – Year for detail information, please see dataformat

name

Storm Name, which is non-unique. There are six lists that are used in rotation and re-cycled every six years, i.e., the 2013 list is used again in 2019. For more information, please see tropical cyclone names.

datetime, year, month, day, hour

Date of report (in Universal Time Coordinate)

latitude,longitude

Location of storm center

status

Storm classification (Tropical Depression, Tropical Storm, or Hurricane)

category

Saffir-Simpson storm category (estimated from wind speed. -1 = Tropical Depression, 0 = Tropical Storm)

max_wind

storm's maximum sustained wind speed (in knots)

min_pressure

Air pressure at the storm's center (in millibars)

ts_diameter

Diameter of the area experiencing tropical storm strength winds (34 knots or above)

hu_diameter

Diameter of the area experiencing hurricane strength winds (64 knots or above)

completeish

whether storms that have complete pressure record, yes or no

Save transformed data for further use.

```
dir <- 'data/'
write_csv(hurdat2_out_add_com, file.path(dir, "hurdat2_out.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.

4.1 Read tranformed data

```
# Read in transformed data
dfile<- "data/hurdat2_out.csv"
#dfile <- "https://raw.githubusercontent.com/jqz300/edav_proj/master/data/storms_all_out.csv"
df <- read_csv(dfile,
                   na = c("NA", "-99", "-999"),
                   col_types = list(
                     id = col_character(),
                   name = col_character(),
                   year =col_integer(),
                   month =col_integer(),
                   day = col_integer(),
                   hour  =col_integer(),
                   latitude  = col_double(),
                   longitude   = col_double(),
                   status  = col_factor(),
                   category = col_factor(),
                   max_wind  = col_double(),
                   min_pressure  = col_integer(),
                   ts_diameter = col_double(),
```

```
                    hu_diameter = col_double(),
                    completeish = col_factor()))
```

```
tail(df)
```

```
## # A tibble: 6 x 16
##   id    name  datetime    year month   day  hour latitude longitude status
##   <chr> <chr> <date>     <int> <int> <int> <int>    <dbl>     <dbl> <fct>
## 1 AL16~ Oscar 2018-11-03  2018    11     3     6     56.6     -23.1 Extra~
## 2 AL16~ Oscar 2018-11-03  2018    11     3    12     57.9     -19.6 Extra~
## 3 AL16~ Oscar 2018-11-03  2018    11     3    18     58.9     -17.1 Extra~
## 4 AL16~ Oscar 2018-11-04  2018    11     4     0     59.8     -14.5 Extra~
## 5 AL16~ Oscar 2018-11-04  2018    11     4     6     60.8     -12.1 Extra~
## 6 AL16~ Oscar 2018-11-04  2018    11     4    12     62.4      -9.1 Extra~
## # ... with 6 more variables: category <fct>, max_wind <dbl>,
## #   min_pressure <int>, ts_diameter <dbl>, hu_diameter <dbl>, completeish <fct>
```

4.2 Figures

```
df %>%
  #filter(completeish == "yes") %>%
  ggplot(aes(x=longitude, y=latitude))+
  geom_hex()+
  facet_wrap(~category, ncol=3)+
  labs(title = "Fig title")
```

```
# need to add map
```

Figure 4.1 shows

```
library(gridExtra)

df %>%
        ggplot()+
        geom_boxplot(aes(x=reorder(category, max_wind, median), y=latitude), varwidth =
        #coord_flip()
        facet_wrap(~status, scale="free")
```

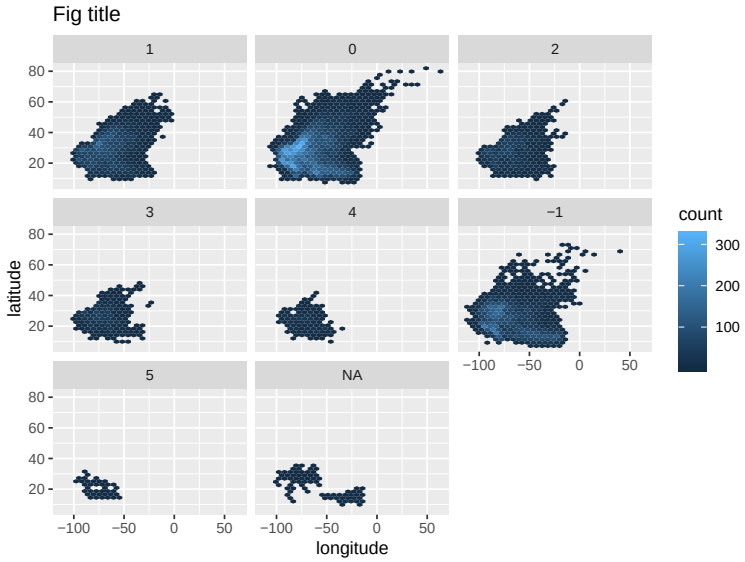

Figure 4.1: Here is a nice figure!

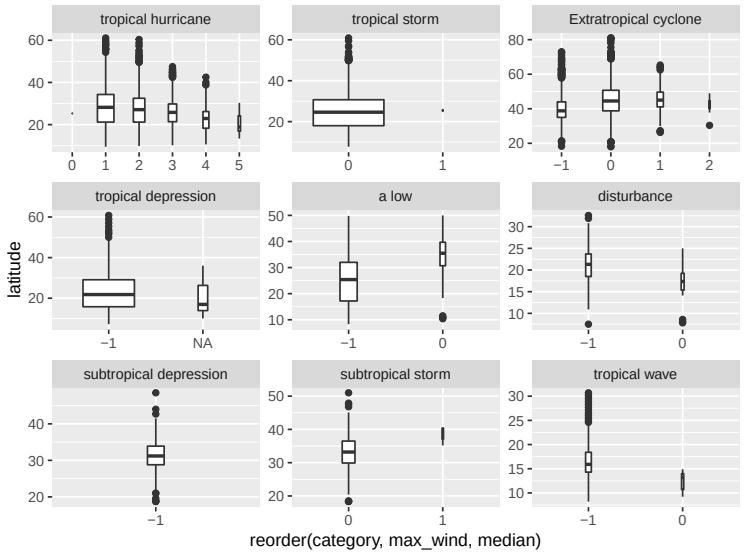

Figure 4.2: Here is a nice figure!

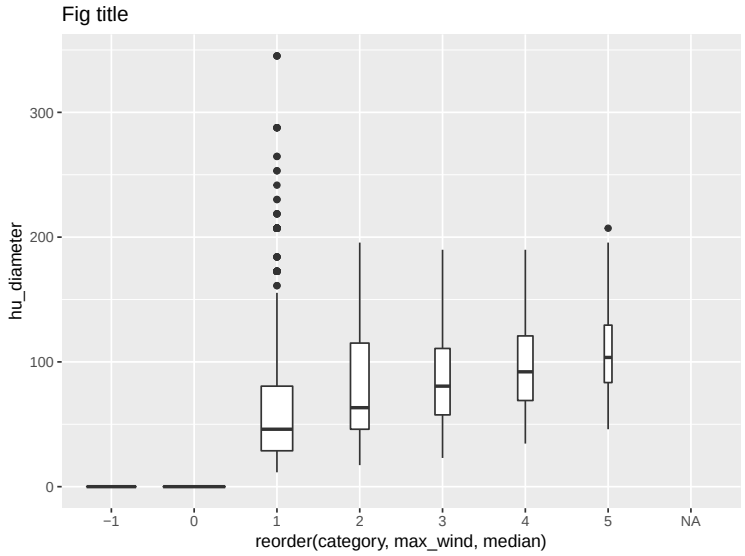

Figure 4.3: Here is a nice figure!

```
library(gridExtra)

df %>%
        ggplot()+
        geom_boxplot(aes(x=reorder(category, max_wind, median), y=hu_diameter), varwid
        #coord_flip()
        #facet_wrap(~status)#, scale="free")+
        labs(title = "Fig title")
```

```
library(ggridges)
library(viridis)
df %>%
        ggplot()+
        geom_density_ridges_gradient(aes(x= year, y= category, group = category, fill =
        scale_fill_viridis()+
        #coord_flip()+
        labs(title = "Fig title")
```

```
library(ggridges)
library(viridis)
df %>%
        ggplot()+
```

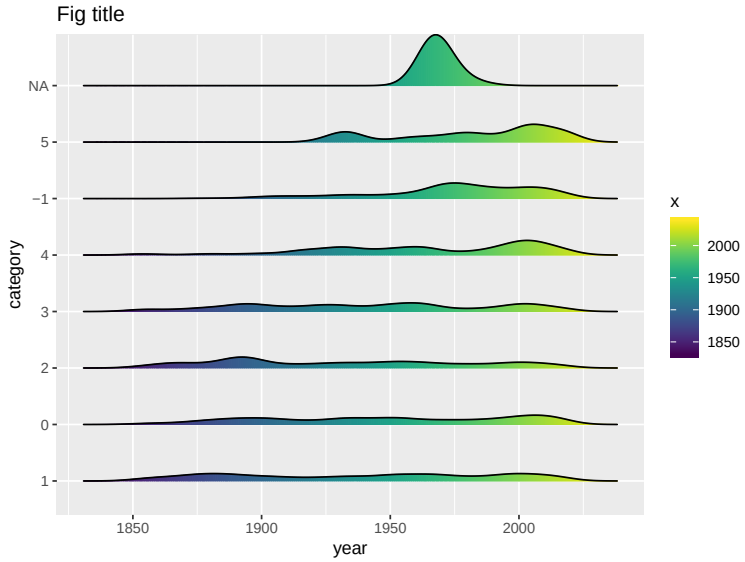

Figure 4.4: Here is a nice figure!

```
        geom_density_ridges_gradient(aes(x= year, y= status, group = status, fill = ..x.., scale
        scale_fill_viridis()+
        #coord_flip()+
        #facet_wrap(~cluster_name_more_short, scale="free", ncol = 1)+
        labs(title = "Fig title")
```

```
# cleveland
```

```
library(parcoords)

#df %>%
#        dplyr::select( ts_diameter, hu_diameter, min_pressure, max_wind, category, status) %>%
#  drop_na() %>%
#        parcoords(alpha = 0.2,
#                  color = list(colorBy = "category"),
#                  withD3 = TRUE,
#                  rownames = FALSE,
#                  reorderable = TRUE,
#                  brushMode = "1d-axes")
```

```
df %>%
  drop_na() %>% filter(name =="Katrina") %>%
  ggplot(aes(x=longitude, y=latitude, color=ts_diameter))+
```

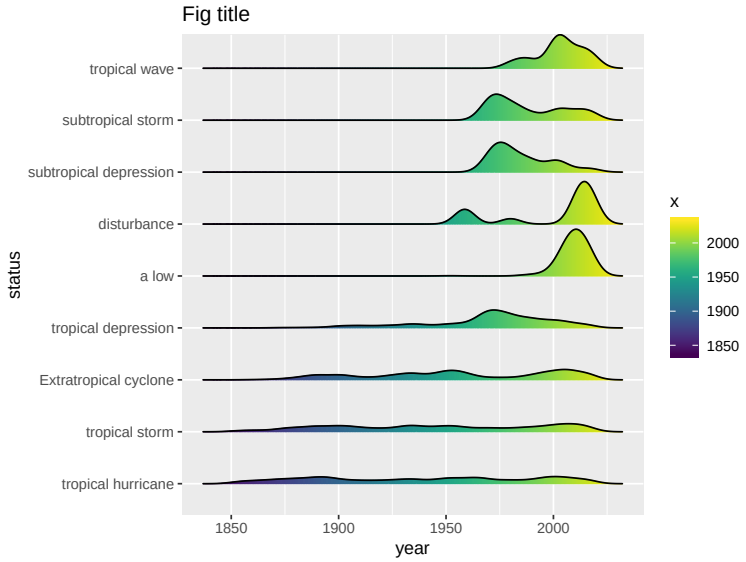

Figure 4.5: Here is a nice figure!

```
  geom_point()
```

```
## need to add map

df %>%
  drop_na() %>% filter(name =="Katrina") %>%
  ggplot(aes(x=datetime, y=max_wind, color=category))+
  geom_point()
```

```
library(plotly)
#df %>%
#  drop_na() %>%
#  filter(name =="Katrina") %>%
#  plot_ly(x=~datetime, y=~max_wind)
```

```
levels(df %>% filter(completeish=="yes") %>% .$status)
```

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

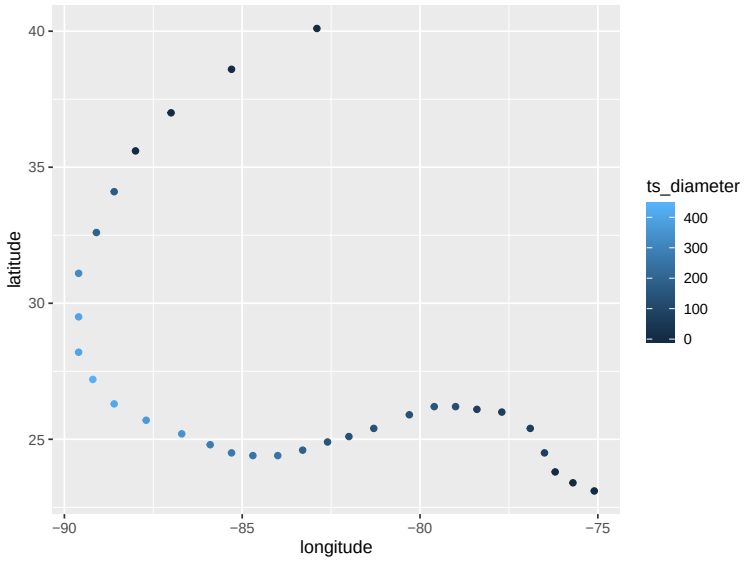

Figure 4.6: Here is a nice figure!

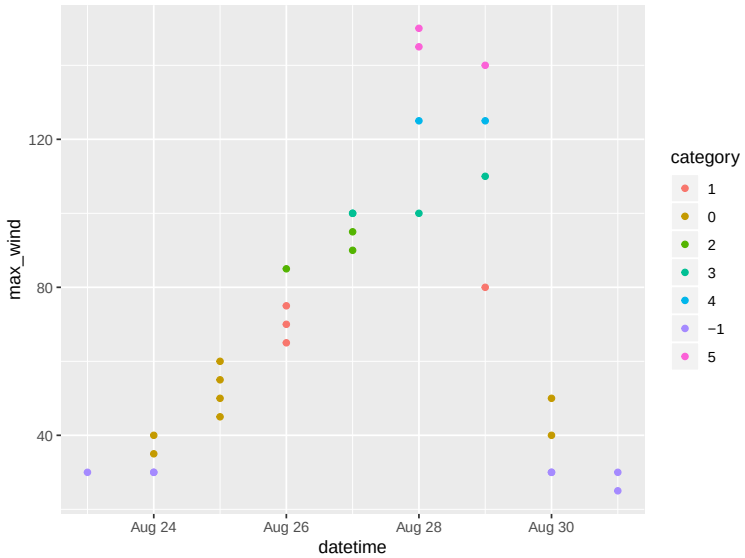

Figure 4.7: Here is a nice figure!

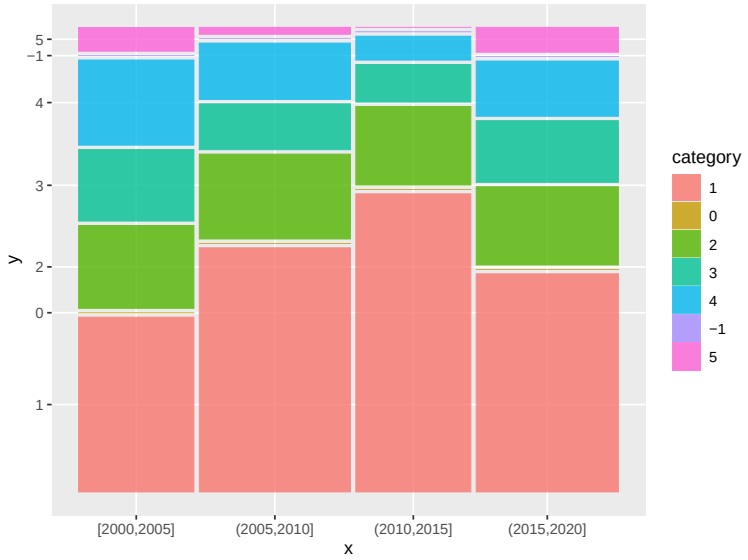

Figure 4.8: Here is a nice figure!

```
#names(df)
df_selected <- df %>%
  #filter(completeish == "yes") %>%
  filter(year>1990) %>%
  #filter (status == c("hurricane","tropical storm","tropical depression")) %>%
  filter(hu_diameter>0) %>%
  drop_na() %>%
  dplyr::mutate(decade=cut(year,
                           breaks = c(2000, 2005, 2010, 2015, 2020),
                           lables = c('00s', '05s','10s','15', '20s'),
                           include.lowest = TRUE, ordered=TRUE))
```

```
library(ggmosaic)
ggplot(data = df_selected) +
   geom_mosaic(aes(x = product(category, decade), fill=category), na.rm=TRUE)
```

```
library(ggmosaic)
ggplot(data = df_selected) +
   geom_mosaic(aes(x = product(category, month), fill=category), na.rm=TRUE)
```

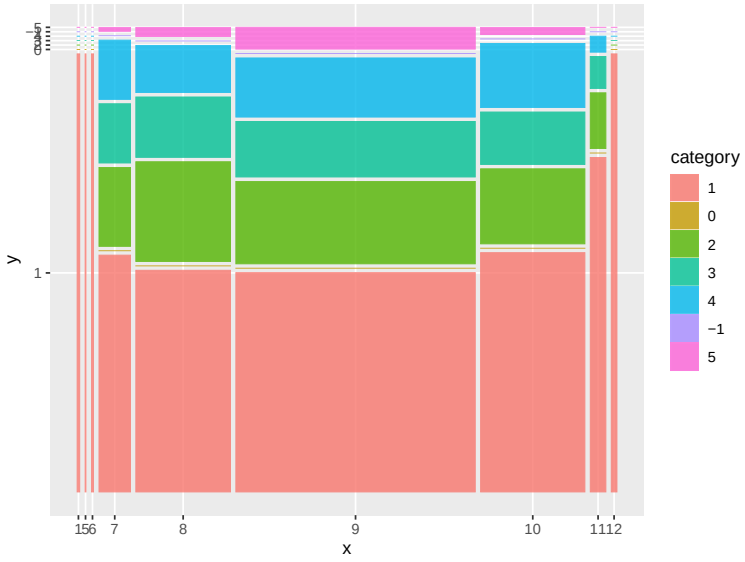

Figure 4.9: Here is a nice figure!

```
df %>%
  select(c("year", "id", "status")) %>%
  unique() %>%
  group_by(year, status) %>%
  drop_na() %>%
  count() %>%
  ungroup() %>%
  ggplot(aes(x=year, y=n))+
  geom_line()+
  facet_wrap(~status, scale="free")
```

4.3 Tables

Summary table if applicable

We can reference tables generated from `knitr::kable()`, e.g., see Table 4.1. And it works even the item is not in this chapter.

```
knitr::kable(
  head(df_selected[2:3], 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

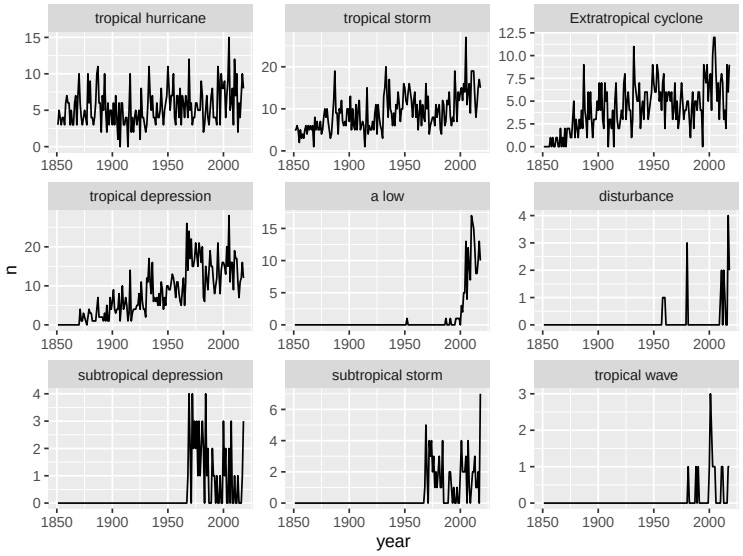

Figure 4.10: Here is a nice figure!

Table 4.1: Here is a nice table!

name	datetime
Alex	2004-08-03
Alex	2004-08-03
Alex	2004-08-03
Alex	2004-08-04
Alex	2004-08-04
Alex	2004-08-04
Alex	2004-08-04
Alex	2004-08-05
Alex	2004-08-05
Alex	2004-08-05

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.