

Project 1: North Atlantic Tropical Cyclones

Stat 133, Spring 2022

Motivation

In this assignment, you are going to perform an exploratory analysis of **tropical cyclones in the North Atlantic from years 1970 to 2020**, focusing on a handful of claims to be further investigated. You will have to write a report containing your code, analysis and narrative. Also, you will have to record a short 3-minute video about some parts of your analysis.

Auxiliary Resources

Here are some optional resources that may help you while working on this project.

- If you haven't studied the material of chapters 8, 9, 10, 11 of the textbook, then you should: <https://www.gastonsanchez.com/intro2cwd/>
- How do hurricanes form? <https://spaceplace.nasa.gov/hurricanes/en/>
- How does a hurricane form? <https://scijinks.gov/hurricane/>
- National Oceanic and Atmospheric Administration (NOAA) <https://www.noaa.gov/>

A) Data IBTrACS

The data for this project comes from the *International Best Track Archive for Climate Stewardship* (IBTrACS) website:

<https://www.ncdc.noaa.gov/ibtracs/index.php?name=ib-v4-access>

The specific dataset is part of **IBTrACS v04**, and you will be working with its Comma Separated Values (CSV) file `ibtracs.NA.list.v04r00.csv`. The link to download the CSV file is available in the following page:

<https://www.ncei.noaa.gov/data/international-best-track-archive-for-climate-stewardship-ibtracs/v04r00/access/csv/>

The associated **data dictionary** is in the following pdf file. This is the document that provides detailed descriptions about the fields (i.e. columns) of the CSV data file:

https://www.ncdc.noaa.gov/ibtracs/pdf/IBTrACS_v04_column_documentation.pdf

A.1) Recommended R Packages

The first code chunk in your Rmd should contain commands to load any R packages. We recommend that you use "tidyverse" and "lubridate" (you can use other packages if you want to).

```
# your first code chunk should load any used packages!!!  
library(tidyverse) # includes dplyr, ggplot2 and other pkgs  
library(lubridate) # for working with dates  
library(spData)    # contains "world" data for maps  
library(gganimate) # for animated graphs (and maps)
```

A.2) Importing Data in R

You will have to download the CSV file to your computer.

You are allowed to use any data-table importing function, either with base R functions e.g. `read.table()`, or with functions from other packages: e.g. `read_table()` from "readr".

Regardless of the importing approach you decide to use, you have to import the data following these specifications:

- Import only the first 16 columns (from SID to LANDFALL)
- Specify data types for the 16 imported columns as:
 - SEASON, NUMBER, WMO_WIND, WMO_PRES, DIST2LAND, and LANDFALL must be of type "integer"
 - LAT and LON must be of type "double" or "real"
 - the rest of the columns must be of type "character"

- Look at the **IBTrACS version 4** website and find how missing values are encoded. Not encoding missing values properly while importing the data in R might produce nonsensical results.

We want to make emphasis on the fact that the imported data frame (or tibble) produced by the reading table function must return a table with 16 columns. In other words, you are NOT allowed to import the data, obtain a table, and then select *a posteriori* the first 16 columns.

At the end of this PDF, you can find an appendix with some sample code that may help you import the IBTrACS data table to R.

A.3) Adding a MONTH column

After importing the data, use the following code to add a new column MONTH by extracting the month number from column ISO_TIME. You may need this MONTH column to perform various analysis taking months into account. This code assumes that your imported data is called `dat`; if this is not the case then modify the code according to your own preferences:

```
# adding column month
# (you may need to change the name of data.frame "dat")
dat$ISO_TIME = as.POSIXct(dat$ISO_TIME)
dat$MONTH <- lubridate::month(dat$ISO_TIME)
```

Include the following code chunk to display the structure of your imported data table. Again, this code assumes that the data frame is called `dat`, feel free to change it according to your needs:

```
# display structure of your data with this command
# (your data object may have a different name)
str(dat, vec.len = 1)
```

B) Univariate Exploratory Data Analysis (*Not to be reported*)

This part does not have to be included in your report, but you must carry out this exploratory analysis, which is typical of almost any data analysis project.

Following the descriptions and details of the **data dictionary** document, perform an exploratory data analysis (EDA) to check that data/values in the columns make sense. For example, the fourth column is `BASIN`, and it could include one or more of the following 7 categories: `NA`, `EP`, `WP`, `NI`, `SI`, `SP`, `SA`. Based on this information, you would need to explore what categories are in `BASIN`, and see if everything makes sense.

We recommend exploring the following columns:

- SEASON
- BASIN
- SUBBASIN
- ISO_TIME (explore a handful of values along the rows of the data table and make sure they all have the adequate format)
- NATURE
- LAT
- LON
- WMO_WIND
- WMO_PRES
- DIST2LAND
- LANDFALL

The type of analysis will depend on the nature of each column (i.e. variable). For those categorical variables, perhaps you may want to identify the unique categories, and obtain their frequencies (or also relative frequencies). For those variables that have a more quantitative flavor (e.g. WMO_WIND), it would be good if you look at their summary statistics, and visualize their distribution (e.g. boxplots, histograms, density curves). Keep in mind that this univariate EDA is intended to “get to know the data, and have a sanity check of the available values”.

C) Main Analysis (*To be reported*)

You will have to analyze the data in order to provide both numeric and visual outputs to the subsections listed below.

You are allowed to use summary tables, statistical charts, and of course maps. Keep in mind that this analysis is decisively exploratory. We are not expecting that you apply predictive models, or perform hypothesis tests, or other type of inferential task. Having said that, your analysis, interpretation, and conclusions should be sound.

C1) Atlantic Hurricane Seasons

Consider the following claims listed below (C1.a - C1.f). For each claim, write R code that provides output that allows you to directly address each claim, determining which parts are true or false. In addition to your R code, include a sound description and interpretation.

- **C1.a)** The 2020 Atlantic hurricane season featured a total of 31 tropical cyclones, all but one of which became a named storm. As expected, none of the tropical cyclones formed pre-season.

- **C1.b)** Of the named storms in the 2020 Atlantic hurricane season, seven of the hurricanes intensified into major hurricanes although none of them reached Category 5 status.
- **C1.c)** The 2010 Atlantic hurricane season had 19 named storms. Despite this above average activity, not one hurricane hit the United States.
- **C1.d)** The 2005 Atlantic hurricane season featured a total of 27 named storms, seven of which became major hurricanes, making this the season with the most number of major hurricanes during the period 1970-2020.
- **C1.e)** In the period from 1970 to 2020, the 2020 Atlantic hurricane season was the most active on record. By “active” we mean the season with the most tropical cyclones.
- **C1.f)** In the 2020 Atlantic hurricane season, 14 storms intensified into hurricanes, making this season the one with the most number of hurricanes during the period 1970 to 2020.

C.2) Animated Map

Make an animated map of storms in 2020, to display their paths (or trajectories). And don't forget to provide a sound description and interpretation for this visual display.

To do this, you will have to plot the locations of the storms with their longitude and latitude on a map. Start by making a static (i.e. non-animated) map following one of the approaches described in Prof. Sanchez's book

<http://www.gastonsanchez.com/intro2cwg/eda-maps.html>

To learn how to animate static ggplots, take a look at the tutorial *Getting Started with the package "gganimate"* by Thomas Lin Pederson and David Robison:

<https://gganimate.com/articles/gganimate.html>

You can see an example of various animated graphics in *An Intro to Animating Charts with "gganimate"* by Joe Drigo Enriquez.

<https://rpubs.com/jedoenriquez/animatingchartsintro>

Look at example 2 *Australia vs New Zealand: life expectancy and GDP per capita* especially the section titled **Animate the scatterplot**, in which Joe Drigo uses the time variable `year` and the function `transition_time()` to make the year-based animated transitions.

D) Submission

You will have to submit your source Rmd file, the generated html document, and the link of your 3-min video.

D.1) Report (Rmd + html)

Your report should include all your code, analysis, results, interpretations, descriptions, etc., as well as your narrative. Above all, do not simply include code chunks, with minimal descriptions, and a boring list of conclusions for each of the analyzed claims. We want to see your “thinking process” and how you organize your workflow, all of this by reading and looking at your report, without you there to explain it to us. Therefore, your submission must “speak for itself”.

Some aspects to keep in mind:

- What was your methodology/approach towards addressing the research questions?
- Describe your data manipulation and exploration process, as well as your analytical steps.
- Do not be afraid to use as many code chunks as necessary. Your code should be easy to read and understand, using descriptive names, well commented, and well organized.
- For visualizations, describe motivations behind the particular ones built and what they illuminate. Make sure any visualizations are functionally labeled (e.g. title, possibly a subtitle, axis labels, units of measurement when applicable).
- You must communicate your key findings and insights clearly.
- Keep the content length of your report to no more than 2500 words. To give you a rough idea: single spaced, 2500 words yields about 5 pages (excluding images). Obviously there are no pages in an html document, but use this guideline to keep track of your code and narrative.

D.2) Video

In addition to the report (Rmd and html files), you will also have to record a video (3-minute max length) in which you use your results to address any three of the claims C1.a - C1.f.

We recommend recording the video using your Zoom Berkeley account, and then sharing with us the public link. Optionally, you can also record a video using other tools (e.g. youtube, vimeo), upload it to your Berkeley Box, or your google drive account.

Regardless of where you decide to host the file of your video, you must share the link in bCourses—in the **comments section of the submission**—and also include the link in your submitted report. (Do NOT upload the video to bCourses).

Make sure that the video does not exceed 3-minutes, that its resolution is okay, without too much background noise, avoiding very low volume or inaudible audio. Above all, record a video in which **both your screen and your face are captured**. We want to see your analysis (numerical and graphical evidence), your results, interpretations, conclusions, and we also want to see your face.

Appendix with sample code for importing data in the next page.

Appendix: Data Importing (Sample Code)

Below we provide auxiliary code that you can use as an optional template to help you import the data into R. Keep in mind that there are multiple ways in which you can perform the importing operation. You are allowed to use other approaches if you want.

```
# -----  
# Optional code to guide your importing data steps.  
# (BTW: there are other ways to achieve the same result)  
# Notice that some lines are incomplete. If you decide to  
# use these commands, you will have to fill-in the blanks.  
# -----  
  
# vector of names for first 16 columns  
col_names <- c(  
  "SID",  
  ... # fill-in with the rest of names!  
  "LANDFALL"  
)  
  
# vector of data-types for first 16 columns  
col_types <- c(  
  "character", # data-type of SID  
  ... # fill-in with the rest of data-types!  
  "integer"    # data-type of LANDFALL  
)  
  
# suggestion for importing CSV file with "read.csv()"  
dat <- read.csv(  
  file = ..., # specify name of file!  
  colClasses = c(col_types, rep("NULL", 147)),  
  stringsAsFactors = FALSE,  
  skip = 77876, # we're not interested in hurricanes before 1970  
  na.strings = ... # specify how missing values are encoded!  
)  
  
# renaming columns using vector col_names  
colnames(dat) <- col_names
```