The People, Places, and Storms of New Orleans

Sean Connin, Daniel Moscoe, Ethan Haley

3/14/2021

Overview

The goal of this project was to transform three untidy datasets and process them in ways that were useful for downstream analysis. The untidy data sources were selected from topics discussed by DATA607 students. Our project team (Daniel Moscoe, Ethan Haley, and Sean Connin) identified the following datasets for tidying, cleaning, and analyzing, each pertaining to aspects of hurricane activity in the Atlantic and related impacts on New Orleans, LA throughout its history:

1. Wikipedia HTML Tables -— measures of hurricane activity, intensity, and damages since 1900
2. Excel Spreadsheets -— demographic changes in New Orleans attributed to Hurricane Katrina (2005)
3. GeoJSON Files – for mapping surviving landmarks and homes of jazz musicians

The datasets identified by our team afforded us the opportunity to construct a body-of-work with thematic coherence while challenging us to acquire and prepare data from different sources and in different forms, as a group.

Methods

Our team met via videoconference four times over two weeks to ensure agreement on project goals, development, and results. Regular asynchronous exchanges via Slack also enabled us to provide each other assistance and updates during interim periods. Data and associated graphics were prepared in the RStudio work environment - with raw data, R scripts, and related materials hosted on a shared Github repository. The repository can be accessed on Github (https://github.com/ebhtra/gumbo-jazz).

This report is organized into one section for each of the three datasets. Each section includes an Overview, R Script, and Summary. Documentation related to data source(s) and description are included in the section overviews. Results, references, and recommendations are discussed in the section summaries.

Our project assessment (observations, lessons learned, etc) are included as final Conclusions.

Part 1) Wikipedia HTML Tables

In this section we explored patterns of recorded hurricane activity from 1900 -to- present in the Atlantic Ocean, Gulf of Mexico and Caribbean Sea. Our goal was to address the following questions:

1. Has seasonal hurricane activity intensified over the past century across regions represented in this dataset?
2. Do composite measures of storm intensity (e.g., annual metrics) correlate well with storm related deaths and/or property damage?
3. Was the 2005 hurricane season, which included Hurricane Katrina, unusual in terms of its intensity?

Data used in our analysis are idencluded in NOAA's "North Atlantic hurricane Database (HURDAT)". However, we scraped our information from 20 html tables listed on the Wikipedia page, "Atlantic hurricane season" - https://en.wikipedia.org/wiki/Atlantic_hurricane_season (https://en.wikipedia.org/wiki/Atlantic_hurricane_season).

Steps taken to format the data for analysis in R (i.e., Tidy format) included: 1) importing the data into RStudio as a list of dataframes (each representing different periods of time); 2) iterating over this list to subset, filter, rename, and update column types; 3) addressing inconsistencies in variable entry, time-span, and title, and 4) combining these elements (20 total) into a final dataframe.

Additional cleaning was required to address uncertainties inherent to values recorded for select variables. For example, storm-related deaths included values input as "Unknown", single numbers, approximate values, and/or a range of values. We converted this information to integer type and limited values to the minimum number of known deaths; values included as "Unknown" were treated as missing (NA) data. We applied similar steps to update a variable for storm-related damages.

Once in Tidy form, we exported the dataset to an external .csv file and proceeded with our analyses.

The following code can be used to import, clean, analyze, and export our data as described above. We conclude this section with a summary of our findings.

```
# Import Wikipedia data tables into a list of dataframes
temp <- read_html('https://en.wikipedia.org/wiki/Atlantic_hurricane_season')
storms <- temp %>%
  html_nodes("table") %>%
  html_table(fill = TRUE)
# Export Wikipedia data (list of dataframes) to create record of raw data.
capture.output(storms, file = "untidy_wikipedia_tables.txt")
# Iterate over list and subset dataframe columns
storms<-storms[c(-1,-2,-3,-4,-5,-6,-7)]
storms<-lapply(storms, function(x) filter(x, x$Year != "Total"))
storms<-lapply(storms, function(x) x[!(names(x) %in% c("Retired names", "Major landfall hurricanes","Number oftropical cyclo
nes", "Notes", "Strongeststorm"))])
# Convert Year column to character type and rename columns as appropriate
storms <- lapply(storms, function(x) {x$Year <- as.character(x$Year);x})
storms<- lapply(storms, function(x) {colnames(x)[2] <- 'Number_Tropical_Storms'; x})
storms<- lapply(storms, function(x) {colnames(x)[3] <- 'Number_Hurricanes'; x})
storms<- lapply(storms, function(x) {colnames(x)[4] <- 'Number_Major_Hurricanes'; x})
storms<- lapply(storms, function(x) {colnames(x)[5] <- 'Accumulated_Cyclone_Energy'; x})
storms<- lapply(storms, function(x) {colnames(x)[7] <- 'Damage_USD'; x})
# Combine list elements to create a single dataframe
storms<-purrr::map_dfr(storms[], dplyr::bind_rows)
# Clean/rename Deaths col and change type to integer
storms<-storms%>%
  mutate(Deaths, Deaths=sub("None", "0", Deaths))%>%
  mutate(Deaths, Deaths=sub("Unknown", "", Deaths))%>%
  mutate(Deaths, Deaths=sub("\\+", "", Deaths))%>%
  mutate(Deaths, Deaths=sub(",", "", Deaths))%>%
  mutate(Deaths, Deaths=sub(">", "", Deaths))%>%
  mutate(Deaths, Deaths=sub("~", "", Deaths))%>%
  rename(Min_Known_Deaths = Deaths)
storms$Min_Known_Deaths <- as.integer(storms$Min_Known_Deaths)
# Clean Damage_USD column and convert to type integer
storms<-storms%>%
  mutate(Damage_USD, Damage_USD=sub("\\$", "",  Damage_USD))%>%
  mutate(Damage_USD, Damage_USD=sub("\\.", "", Damage_USD))%>%
  mutate(Damage_USD, Damage_USD=sub("\\smillion", "000000", Damage_USD))%>%
  mutate(Damage_USD, Damage_USD=sub("\\sbillion", "000000000",  Damage_USD))%>%
  mutate(Damage_USD, Damage_USD=sub(">|\\+|,", "",  Damage_USD))%>%
  mutate(Damage_USD = str_extract_all(Damage_USD, "\\d+"))
storms$Damage_USD<-as.integer(storms$Damage_USD)
# Review final dataframe: first five rows
head(storms, 5)%>%kbl%>%kable_material(c("striped"))
```

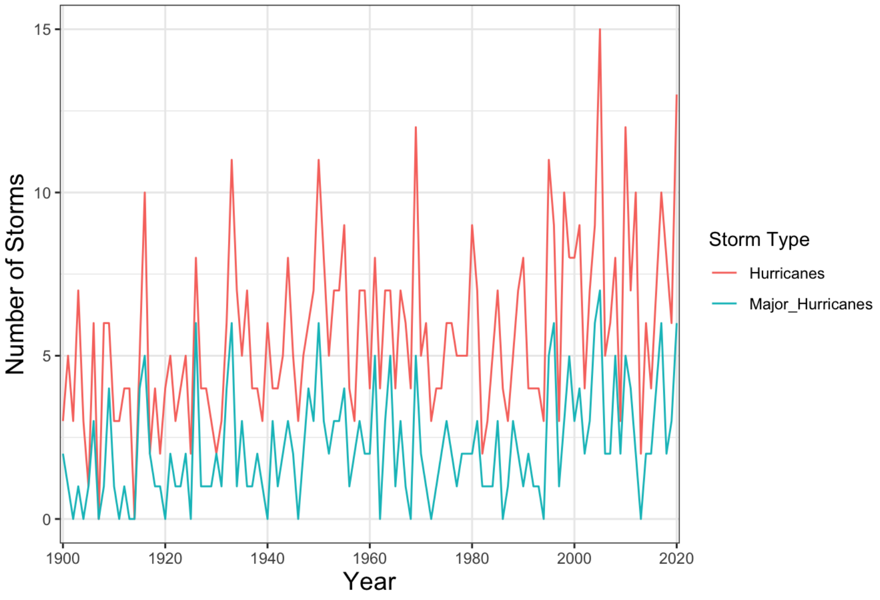
Year	Number_Tropical_Storms	Number_Hurricanes	Number_Major_Hurricanes	Accumulated_Cyclone_Energy	Min_Known_Dea
1900	7	3	2	83.35	8
1901	12	5	1	98.98	
1902	5	3	0	32.65	
1903	10	7	1	102.07	
1904	5	3	0	30.35	

```
# Save final dataframe as .csv file
write.csv(storms, "Proj2_Atlantic_Hurricanes.csv")
# Calculate Summary Statistics and obtain average percentage of hurricanes and major hurricanes over the period of record (f
eature engineering).
summary(storms)
```

```
##      Year           Number_Tropical_Storms Number_Hurricanes
##  Length:121         Min.   : 1.00          Min.   : 0.000
##  Class :character   1st Qu.: 7.00          1st Qu.: 4.000
##  Mode  :character   Median :11.00          Median : 5.000
##                     Mean   :10.74          Mean   : 5.612
##                     3rd Qu.:13.00          3rd Qu.: 7.000
##                     Max.   :30.00          Max.   :15.000
##
##  Number_Major_Hurricanes Accumulated_Cyclone_Energy Min_Known_Deaths
##  Min.   :0.000           Min.   :  2.53             Min.   :    0.00
##  1st Qu.:1.000           1st Qu.: 49.77             1st Qu.:   25.75
##  Median :2.000           Median : 83.48             Median :  100.50
##  Mean   :2.223           Mean   : 93.35             Mean   :  818.32
##  3rd Qu.:3.000           3rd Qu.:126.30             3rd Qu.:  486.75
##  Max.   :7.000           Max.   :258.57             Max.   :12000.00
##                                                     NA's   :1
##    Damage_USD
##  Min.   :6.700e+04
##  1st Qu.:4.500e+07
##  Median :1.000e+08
##  Mean   :2.496e+08
##  3rd Qu.:3.090e+08
##  Max.   :1.575e+09
##  NA's   :52
```

```
p_hurricane<-storms%>%select(Number_Tropical_Storms, Number_Hurricanes)%>%mutate(n_h = (Number_Hurricanes/Number_Tropical_St
orms)*100)%>%summarize(mean_h = mean(n_h, na.rm=TRUE))
p_major_hurricane<-storms%>%select(Number_Major_Hurricanes, Number_Hurricanes)%>%mutate(percent = (Number_Major_Hurricanes/N
umber_Hurricanes)*100)%>%summarize(mean_major = mean(percent, na.rm=TRUE))
# Plot storms counts over time
num <- storms%>%select(Year, Number_Hurricanes, Number_Major_Hurricanes)%>%
  rename(Hurricanes = Number_Hurricanes, Major_Hurricanes=
    Number_Major_Hurricanes)%>%
    pivot_longer(cols=-c(Year), names_to = "Storm_Type", values_to=
    "Storm_Number")
num%>%ggplot(aes(x = Year, y = Storm_Number, group = Storm_Type, color=
    Storm_Type))+
    geom_line()+
    scale_x_discrete(breaks=c("1900","1920", "1940", "1960",
    "1980","2000","2020"))+
    theme_bw()+
    theme(axis.title.x = element_text(size=14))+
    theme(axis.title.y = element_text(size=14))+
    labs(y="Number of Storms", x = "Year")+
    labs(color = "Storm Type")+
    ggtitle("Figure. Annual Count of Atlantic Storms Since 1900")+
    theme(plot.title = element_text(hjust = 0.5))
```

Figure. Annual Count of Atlantic Storms Since 1900



```
# Compare accumulated storm energy from 1900-Present
ace <- storms%>%select(Year, Accumulated_Cyclone_Energy)

ace%>%ggplot(aes(x = Year, y = Accumulated_Cyclone_Energy))+
    geom_point(color="light blue")+
    scale_x_discrete(breaks=c("1900","1920", "1940", "1960", "1980",
    "2000","2020"))+
    geom_hline(yintercept = 152.5, linetype="dotted", color = "dark green", size=0.750)+
    geom_vline(xintercept = "2005", linetype="dotted", color = "red",
    size=0.75)+
    theme_bw()+
    theme(axis.title.x = element_text(size=11))+
    theme(axis.title.y = element_text(size=11))+
    labs(y=TeX("ACE ($kn^{2}$)"), x = "Year", title = "Figure 2. Annual Accumulated Energy of
    Atlantic Storms Since 1900", subtitle = TeX("Reported in Squared Knots ($kn^{2}$)"))+
    theme(plot.title = element_text(hjust = 0.5))+
    theme(plot.subtitle=element_text(size=11, hjust=0.5,
    color="black"))
```

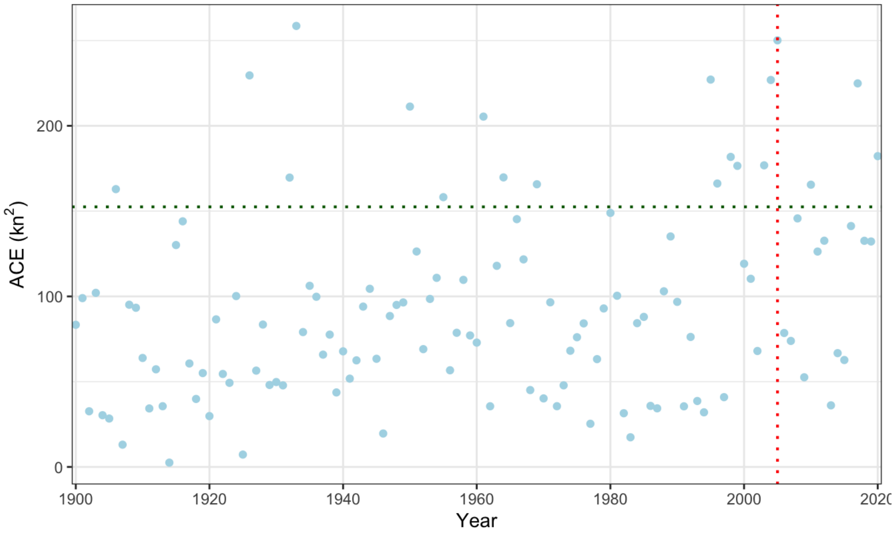
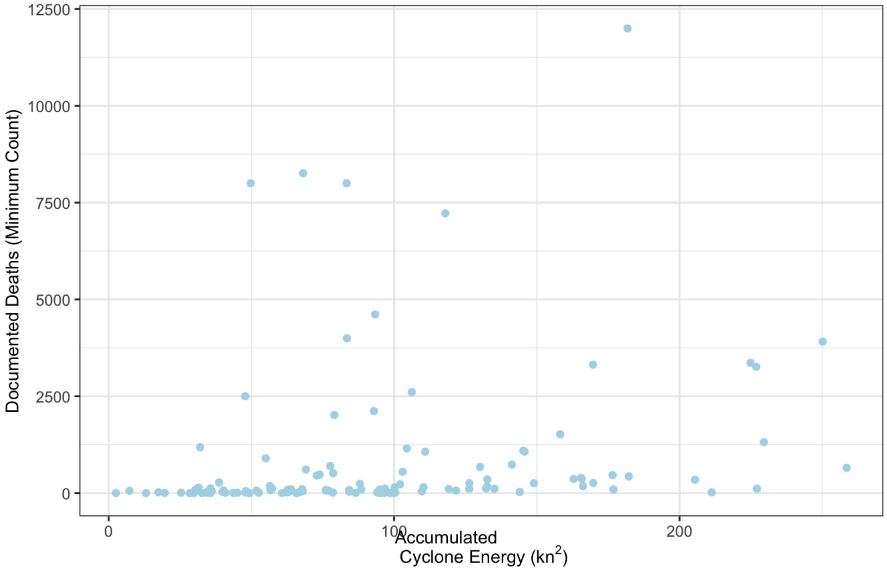


Figure 2. Annual Accumulated Energy of
Atlantic Storms Since 1900

Reported in Squared Knots (kn^2)

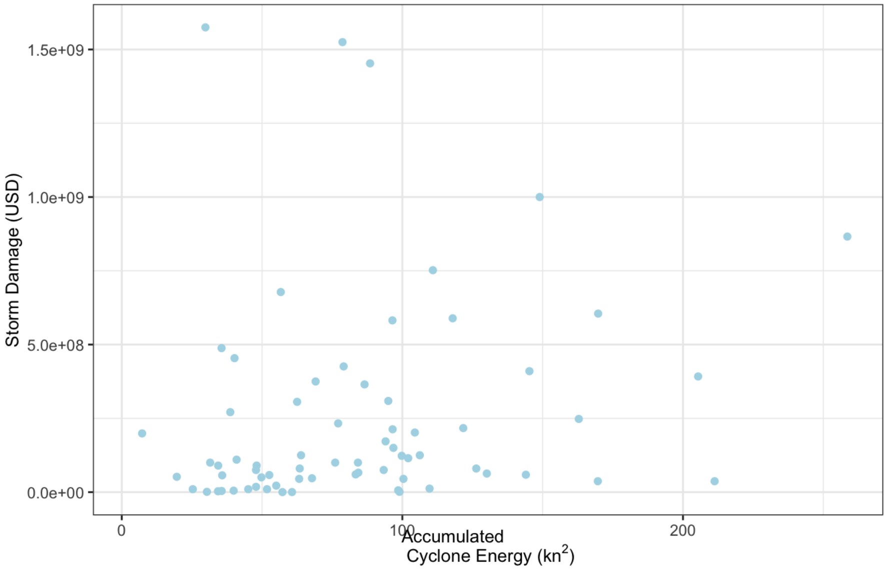
```
# Compare storm related deaths vs. accumulated storm energy by year
ace_death<-storms
ace_death%>%ggplot(aes(Accumulated_Cyclone_Energy, Min_Known_Deaths))+
    geom_point(color="light blue")+
    theme_bw()+
    theme(axis.title.x = element_text(size=10))+
    theme(axis.title.y = element_text(size=10))+
    labs(y="Documented Deaths (Minimum Count)", x = TeX("Accumulated
    Cyclone Energy ($kn^{2}$)"), title = "Figure 2. Storm Related Deaths vs. Cumulative Storm
    Intensity: 1900 to Present")+
    theme(plot.title = element_text(hjust = 0.5))
```

Figure 2. Storm Related Deaths vs. Cumulative Storm Intensity: 1900 to Present



```
# Compare total storm damage vs. accumulated storm energy by year
ace_damage<-storms
ace_damage%>%ggplot(aes(Accumulated_Cyclone_Energy, Damage_USD))+
    geom_point(color="light blue")+
    theme_bw()+
    theme(axis.title.x = element_text(size=10))+
    theme(axis.title.y = element_text(size=10))+
    labs(y="Storm Damage (USD)", x = TeX("Accumulated
    Cyclone Energy ($kn^{2}$)"), title = "Figure 3. Total Storm Damage vs. Cumulative Storm
    Intensity: 1900 to Present")+
    theme(plot.title = element_text(hjust = 0.5))
```

Figure 3. Total Storm Damage vs. Cumulative Storm Intensity: 1900 to Present



Wikipedia: Results

The total number of tropical storms recorded in the Atlantic ranged from 0 to 15 from years 1900-2020. Approximately 52% of these storms were documented hurricanes. And approximately 38% of these hurricanes were identified as Category 3 or higher on the Saffir–Simpson hurricane wind scale (i.e., major hurricanes). While inter-annual variation in the number of Atlantic hurricanes was high, there appears to be an upward trend in these numbers since 1900 (Figure 1). In contrast, there were no apparent trends in Accumulated Cyclone Energy (ACE: an aggregate index of storm energy) over the same period (Figure 2).

Similarly, there was no obvious relationship between annual ACE estimates and either storm related deaths or damage. These results were unexpected and may owe to several factors: 1) hurricanes that made landfall were not distinguished in the data; and 2) inter-site variations in population and infrastructure may mask the impact of hurricanes, particularly when data are aggregated annually and over large spatial scales. Both were the case for this dataset.

It is interesting to note that there were only 19 “extremely active” hurricane seasons in the past 120 years(i.e., ACE >152.5; see points above dashed green line in Figure 2). And that only year 1930 exceeded the 2005 hurricane season in terms of total storm activity (ACE 258.6 vs 250.1, respectively). Figure 2 includes a dashed red line to indicate year 2005 - which included Hurricane Katrina. While 2005 ranked 8th highest in storm-related mortality (3,912 deaths), the 1998 storm season was far deadlier (~12,000 deaths) due to the affects of Hurricane Mitch. Unfortunately, our dataset did not include estimates of storm-related damages for the 2005 hurricane season.

Our analyses can be refined/improved through additional data collection and documentation. We offer the following recommendations for future study: 1) limit the scope of analysis to the Gulf of Mexico, compile data for individual storm events rather than annual totals; 2) distinguish hurricanes that made landfall from annual counts; 3) acquire and verify additional estimates of storm damage, 4) employ time series models to evaluate any trends in annual hurricane patterns over time.

For information regarding ACE measurements and categories, please refer to the following websites:

1. National Weather Service Climate Prediction Center: https://www.cpc.ncep.noaa.gov/products/outlooks/Background.html (https://www.cpc.ncep.noaa.gov/products/outlooks/Background.html)
2. Saffir Simpson Scale: https://bit.ly/3cvRIep (https://bit.ly/3cvRIep)

Part 2) Excel Spreadsheets

Procedure Overview

In this section of the project, we examine demographic data from the New Orleans Metropolitan Area for the period 2000-2019. The data are contained in an Excel spreadsheet. Some of the features of the data in the spreadsheet enhance human readability. For example, geographic regions are listed as column headers, even though each of these regions represents a level of a variable. Variables, like different age groups, are listed as row headers. While this arrangement makes the table easier to read at a glance, it means the table does not have a “tidy” structure. In this section, we import the data from the Excel sheet, transform it into a tidy form, and conduct some exploratory analysis.

Each of the tables from the Excel sheet required similar transformations:
* Transpose the table so that observations reside in rows and variables in columns.
* Repair row names.
* Repair column names.
* Drop empty rows and columns.
* Convert data to a numeric form.

While the transformation of each table conformed to this rough outline, each table also required some special manipulation.

The product of the transformation of these individual tables is one large table, `combined` , that contains all the information from all the imported tables.

Procedure Detail

Import the raw data from the Excel sheet. The path to the local copy of the Excel file is stored as `xlsx_src` . (Note that the `readxl` package requires a local file and will not import data directly from a URL.) Each table is named for the position of its upper-left cell.

```
S1A14 <- read_excel(xlsx_src, range = "A14:C18")
S1A22 <- read_excel(xlsx_src, range = "A22:U48")
S1A53 <- read_excel(xlsx_src, range = "A53:B68")
S1A72 <- read_excel(xlsx_src, range = "A72:J88")
S1A91 <- read_excel(xlsx_src, range = "A91:I104")
S1A108 <- read_excel(xlsx_src, range = "A108:I110")
```

We begin by transforming the most complex table, `S1A22` , in order to demonstrate completely all the elements of the general transformation procedure described above.

```
###S1A22###
#TRANSPOSE
S1A22 <- S1A22 %>%
  t() %>%
  data.frame()
#FIX ROW NAMES
S1A22 <- rownames_to_column(S1A22)
S1A22$rowname <- gsub('.[0-9]+', NA, S1A22$rowname)
S1A22 <- fill(S1A22, rowname)
#FIX COLNAMES
S1A22 <- rename(S1A22, "parish" = rowname,
                "year" = X1,
                "f_wht_not_hisp" = X3, #f_ indicates fraction, not count
                "f_blk_not_hisp" = X4,
                "f_hisp_any" = X5,
                "f_asn_not_hisp" = X6,
                "age_btw_0_4" = X9,
                "age_btw_5_9" = X10,
                "age_btw_10_14" = X11,
                "age_btw_15_19" = X12,
                "age_btw_20_24" = X13,
                "age_btw_25_29" = X14,
                "age_btw_30_34" = X15,
                "age_btw_35_39" = X16,
                "age_btw_40_44" = X17,
                "age_btw_45_49" = X18,
                "age_btw_50_54" = X19,
                "age_btw_55_59" = X20,
                "age_btw_60_64" = X21,
                "age_btw_65_69" = X22,
                "age_btw_70_74" = X23,
                "age_btw_75_79" = X24,
                "age_btw_80_84" = X25,
                "age_geq_85" = X26)
#DROP EMPTY ROWS AND COLUMNS
S1A22 <- S1A22[-c(1),-c(3,8,9)]
S1A22 <- remove_rownames(S1A22)
#CONVERT TO NUMERIC
S1A22[2:length(S1A22[1,])] <- sapply(S1A22[2:length(S1A22[1,])], as.numeric)
```

Choosing variable names carefully will be important here, since using these names consistently across tables will allow us to more easily construct `combined`.

```
###S1A14###
#TRANSPOSE
S1A14 <- S1A14 %>%
  t() %>%
  data.frame()
#FIX ROW NAMES
S1A14 <- rownames_to_column(S1A14)
S1A14 <- cbind(parish = "Orleans", S1A14)
#FIX COLNAMES
S1A14 <- rename(S1A14, "year" = rowname,
                "wht_not_hisp" = X2,
                "blk_not_hisp" = X1,
                "hisp_any" = X3,
                "asn_not_hisp" = X4)
#DROP EMPTY ROWS AND COLUMNS
S1A14 <- S1A14[-c(1),-c(3,5)]
S1A14 <- remove_rownames(S1A14)
#CONVERT TO NUMERIC
S1A14[2:length(S1A14[1,])] <- sapply(S1A14[2:length(S1A14[1,])], as.numeric)
```

```
###S1A53###
#FIX ROW NAMES
S1A53 <- cbind(parish = "Orleans", S1A53)
#FIX COLNAMES
S1A53 <- rename(S1A53,
                "year" = Year,
                "blk_not_hisp" = ...2)
#CONVERT TO NUMERIC
S1A53[2:length(S1A53[1,])] <- sapply(S1A53[2:length(S1A53[1,])], as.numeric)
```

```
###S1A72###
S1A72 <- pivot_longer(S1A72, `Orleans`:`New Orleans Metro`)
#FIX COLNAMES
S1A72 <- rename(S1A72, "parish" = name,
                "year" = ...1,
                "hisp_any" = value)
#IMPROVE CONSISTENCY ACROSS TABLES
S1A72$parish[S1A72$parish == "St. John the Baptist"] <-
  "St. John"
#DROP EMPTY ROWS AND COLUMNS AND REARRANGE
S1A72 <- S1A72[-c(1:9),]
S1A72 <- S1A72[,c(2,1,3)]
#CONVERT TO NUMERIC
S1A72[2:length(S1A72[1,])] <- sapply(S1A72[2:length(S1A72[1,])], as.numeric)
```

```
###S1A91###
#TRANSPOSE
S1A91 <- S1A91 %>%
  t() %>%
  data.frame()
#FIX ROW NAMES
S1A91 <- rownames_to_column(S1A91)
S1A91$rowname <- gsub('.[0-9]+', NA, S1A91$rowname)
S1A91 <- fill(S1A91, rowname)
#FIX COLNAMES
S1A91 <- rename(S1A91, "parish" = rowname,
                "year" = X1,
                "f_hisp_cub" = X3,
                "f_hisp_dom" = X4,
                "f_hisp_mex" = X5,
                "f_hisp_pr" = X6,
                "f_hisp_hon" = X7,
                "f_hisp_gua" = X8,
                "f_hisp_nic" = X9,
                "f_hisp_sal" = X10,
                "f_hisp_ca" = X11,
                "f_hisp_sa" = X12,
                "f_hisp_oth" = X13)
#DROP EMPTY ROWS AND COLUMNS
S1A91 <- S1A91[-c(1,3,5,7,9),-3]
S1A91 <- remove_rownames(S1A91)
#CONVERT TO NUMERIC
S1A91[2:length(S1A91[1,])] <- sapply(S1A91[2:length(S1A91[1,])], as.numeric)
```

```
###S1A108###
#TRANSPOSE
S1A108 <- S1A108 %>%
  t() %>%
  data.frame()
#FIX ROW NAMES
S1A108 <- rownames_to_column(S1A108)
S1A108$rowname <- gsub('.[0-9]+', NA, S1A108$rowname)
S1A108 <- fill(S1A108, rowname)
#FIX COLNAMES
S1A108 <- rename(S1A108, "parish" = rowname,
                "year" = X1,
                "age_btw_0_18" = X2)
#DROP EMPTY ROWS AND COLUMNS
S1A108 <- S1A108[-c(1),]
S1A108 <- remove_rownames(S1A108)
#CONVERT TO NUMERIC
S1A108[2:length(S1A108[1,])] <- sapply(S1A108[2:length(S1A108[1,])], as.numeric)
```

Now that each table conforms to a tidy format, each variable name is used consistently whenever it appears across tables, and each observation is led by a parish and a year in the first two columns, we are ready to join the tables. Since we don't want to lose any information, we use `full_join` on `parish` and `year` on one table after the other. The result is a large table containing all the information, called `combined`. The dataframe `combined` is then exported to a `csv` file (https://raw.githubusercontent.com/ebhtra/gumbo-jazz/parishesCombined.csv). The path for this file is stored as `csv_dest`.

```
combined <-
  full_join(S1A14, S1A22, by = c("parish", "year")) %>%
  full_join(S1A53, by = c("parish", "year")) %>%
  full_join(S1A72, by = c("parish", "year")) %>%
  full_join(S1A91, by = c("parish", "year")) %>%
  full_join(S1A108, by = c("parish", "year"))
write_csv(combined, csv_dest, na = "")
```

A first glance at `combined` shows a significant proportion of missing values. It's interesting to compare `combined` to the original Excel tables, which contained no empty cells. Combining the data from each table together helps highlight gaps in the dataset that were hidden by the "full" appearance of the Excel tables. For example, the dataset is missing year-by-year counts of the white, Asian, and African American populations in most or all parishes.

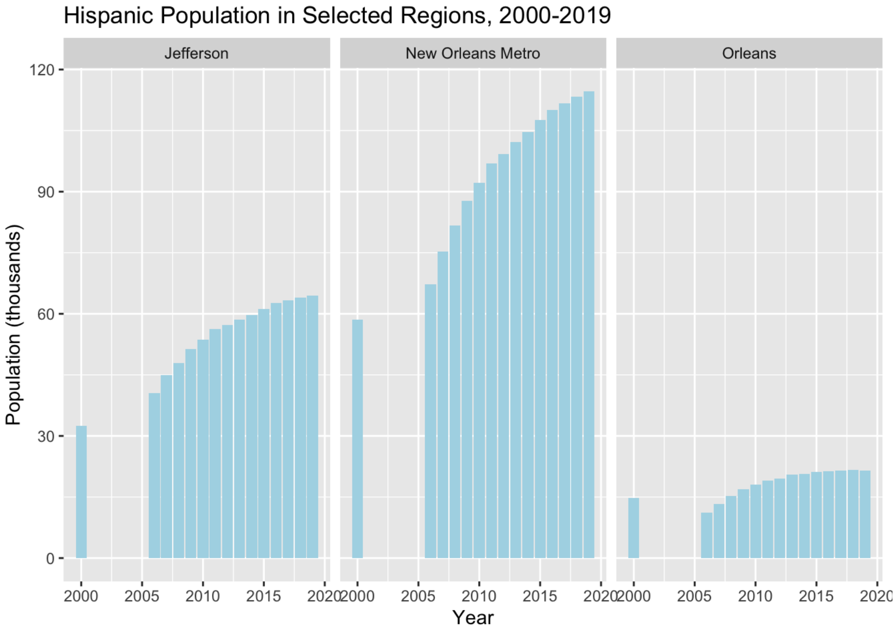
With a tidy table containing all the data, we can now proceed to some exploratory analysis.

Analysis

How did the number of African American and Hispanic residents in select parishes change during the period 2000-2019?

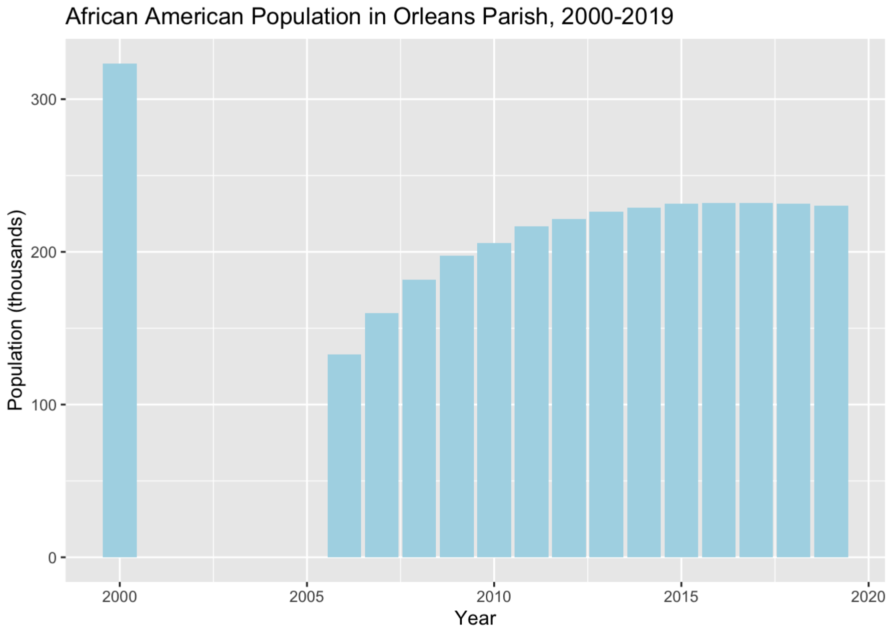
This question allows us to explore some of the more complete regions of this dataset. While we might have asked this same question about the other demographic groups described in the dataset, complete year-by-year data is only available for Hispanic residents (in all parishes) and African American residents (in Orleans parish).

```
combined %>%
  filter(parish %in% c("Orleans", "Jefferson", "New Orleans Metro")) %>%
  ggplot() +
  geom_col(mapping = aes(x = year, y = hisp_any/1000), fill = "light blue") +
  facet_wrap(~parish) +
  labs(title = "Hispanic Population in Selected Regions, 2000-2019",
       x = "Year",
       y = "Population (thousands)")
```



The plot shows that in the three parishes with the greatest numbers of Hispanic residents, the number of Hispanic residents has increased in the period from 2000 to 2019. In the New Orleans Metro area broadly, the number of Hispanic residents has nearly doubled. In Jefferson and Orleans, increases have been more modest. Measurements are missing for the period from 2001-2005.

```
combined %>%
  filter(parish == "Orleans") %>%
  ggplot() +
  geom_col(mapping = aes(x = year, y = blk_not_hisp/1000), fill = "light blue") +
  labs(title = "African American Population in Orleans Parish, 2000-2019",
       x = "Year",
       y = "Population (thousands)")
```



By contrast, the number of African Americans in Orleans Parish has declined significantly during the period from 2000 to 2019. The decline is sharpest during the period for which data is missing, which includes Hurricane Katrina. The graph above is consistent with the claim that Hurricane Katrina caused a very large fraction–well over half–of African American residents to leave Orleans Parish. While some have returned, the total African American population remains about 30% smaller than it was in 2000.

Source for Part 2:

The Excel file analyzed in this section is available here (https://github.com/ebhtra/gumbo-jazz/blob/main/New%20Orleans%20Who%20Lives%20data%20tables.xlsx?raw=true). This file was provided by The Data Center (https://www.datacenterresearch.org/data-resources/who-lives-in-new-orleans-now/) in response to a request.

Part 3) GeoJSON Files

Procedure Overview

In this section, the goal is to build an interactive map of Orleans Parish, showing historic and geographic features to help us better understand the area.

To do so, we'll use GeoJSON data showing historic landmarks and jazz musician homes, downloaded from a New Orleans government website. The jazz homes data will be loaded without transformation into a map layer using kepler.gl, but the landmarks data will need to undergo a process of flattening and processing, in order to extract dates from it. With properly formatted dates, the data can then be used in the kepler.gl application to add a dimension of time to the map.
This procedure will also allow us to do some basic analysis of the data, and to draw some interesting conclusions about Orleans Parish and its history.

Procedure Detail

Use the geojsonR and jsonlite packages to load the geoJSON files.

```
# source <- "https://data.nola.gov/Geographic-Base-Layers/Local-Landmarks/srrj-xwma"
f <- './Local Landmarks.geojson'
landmarks <- FROM_GeoJson(f)
class(landmarks)
```

```
## [1] "list"
```

```
names(landmarks)
```

```
## [1] "features" "type"
```

What do these hold?

```
c(length(landmarks$features), length(landmarks$type))
```

```
## [1] 1242    1
```

With JSON objects, or any objects, for that matter, it's often best to inspect 1242 things cautiously:

```
str(landmarks$features[[1]])
```

```
## List of 3
##  $ geometry  :List of 2
##   ..$ type       : chr "Point"
##   ..$ coordinates: num [1:2] -90.1 29.9
##  $ properties:List of 14
##   ..$ addl_addr : chr ""
##   ..$ architect : chr "James Freret"
##   ..$ const_date: chr "1868"
##   ..$ geo_addr  : chr "1641 Amelia Street"
##   ..$ name      : chr "Hernandez-Davis House"
##   ..$ no_cbd    : chr "New Orleans"
##   ..$ nom_des   : chr "Designated"
##   ..$ num1_edit : chr "1641.0"
##   ..$ num2_edit : chr "0.0"
##   ..$ num_orig  : chr "1641"
##   ..$ objectid  : chr "1"
##   ..$ staff     : chr "BDB"
##   ..$ str_orig  : chr "Amelia Street"
##   ..$ street    : chr "Amelia Street"
##  $ type      : chr "Feature"
```

So each item in the feature list (1242 of them) is one landmark. Each of these comprises a `type` field ("Feature", self-referentially), another nested structure `geometry`, describing its own `type` as `Point` or `MultiPolygon` for some items, and listing its two `coordinates`, and then a bunch of `properties` like architect, address, year of construction, etc.

Build a utility function in hopes of saving some time further down the line when processing other GeoJSON files.

```
# this function takes a geojson file as input and returns a list of 2 data.frames,
## one for the Points, and one for the MultiPolygons
geojson2table <- function(gjfile) {
  geo <- FROM_GeoJson(gjfile)
  pointframe <- 0  # accumulate Points here
  polyframe  <- 0  # and MultiPolygons here
  for (feat in geo$features){
    if (feat$geometry$type == 'Point') {
      if (!is.data.frame(pointframe)) {
        pointframe <- data.frame(feat)
      } else {
          pointframe <- rbind(pointframe, data.frame(feat))
      }
    } else if (feat$geometry$type == 'MultiPolygon') {  # store these in case
        if (!is.data.frame(polyframe)) {
         polyframe <- data.frame(feat)
        } else {
          polyframe <- rbind(polyframe, data.frame(feat))
        }
      }
  }
  # Transforming the nested structure using data.frame() makes two rows for each
  # observation: 1 for the latitude and one for longitude.  Pivoting these
  # wider doesn't work though, since each lat/lon is an actual value.

  # Plan B: Combine every 2 rows into lat/lon pairs using lead(), and then
  # remove the redundant rows.
  pointframe$lat <- lead(pointframe$geometry.coordinates)
  pointframe$lon <- pointframe$geometry.coordinates
  # now drop every second row, which is a dupe
  pointframe <- pointframe %>%
    filter(row_number() %% 2 == 1)
  list(points = pointframe, polys = polyframe)
}
```

Make 2 frames for Landmarks, using the above function

```
landmarklist <- geojson2table('./Local Landmarks.geojson')
names(landmarklist)
```

```
## [1] "points" "polys"
```

And write them to csv

```
write.csv(landmarklist$points, 'landmarkPoints.csv')
write.csv(landmarklist$polys, 'landmarkPolygons.csv')
```

Map construction years to datetimes so that they can be read by kepler.gl

```
landpoints <- read.csv('./landmarkPoints.csv')
head(landpoints, n = 2)
```

```
##   X geometry.type geometry.coordinates  properties.addl_addr
## 1 1         Point            -90.09479
## 2 2         Point            -90.08886 802 Delachaise Street
##   properties.architect properties.const_date      properties.geo_addr
## 1         James Freret                  1868       1641 Amelia Street
## 2              Unknown                       3445 Annunciation Street
##           properties.name properties.no_cbd properties.nom_des
## 1   Hernandez-Davis House       New Orleans         Designated
## 2 Mystical's boyhood home       New Orleans          Nominated
##   properties.num1_edit properties.num2_edit properties.num_orig
## 1                 1641                    0                1641
## 2                 3445                    0                3445
##   properties.objectid properties.staff properties.str_orig   properties.street
## 1                   1              BDB       Amelia Street       Amelia Street
## 2                   2              BDB Annunciation Street Annunciation Street
##      type      lat       lon
## 1 Feature 29.92820 -90.09479
## 2 Feature 29.91984 -90.08886
```

```
# The first 3 columns aren't going to be needed
landpoints <- landpoints %>%
  select(c(4:ncol(landpoints)))
```

```
# We need datetimes for kepler.gl
#install.packages('datetime')
library(datetime)
library(purrr)
# We need to deal with "c.1850", "1879-1881", etc., and add times
dates <- str_match(landpoints$properties.const_date, '[0-9]{4}')
# helper func
datify <- function(year) {
  y <- year
  if (!is.na(y)) {
    y <- paste(y, '/01/01 01:01', sep='')
    y <- as.datetime(y, format = '%Y/%m/%d %H:%M')
  }
  y
}
dates <- as.datetime(as.numeric(map(dates, datify)))
# Add processed dates to the d.f
landpoints$datetime <- dates
write.csv(landpoints, 'landmarkPoints.csv')
# kepler.gl won't allow NA's in date column, so need to filter those out.
dated <- landpoints %>%
  filter(!is.na(landpoints$datetime))

write.csv(dated, 'datedLandmarks.csv')
```

```
geo <- read_json('./Jazz Houses.geojson')
str(geo$features[[1]])
```

```
## List of 3
##  $ type      : chr "Feature"
##  $ properties:List of 4
##   ..$ address          : chr "811 N LIBERTY ST"
##   ..$ musicianfirstname: chr "James"
##   ..$ musiciansurname  : chr "Brown"
##   ..$ objectid         : chr "355"
##  $ geometry  : NULL
```

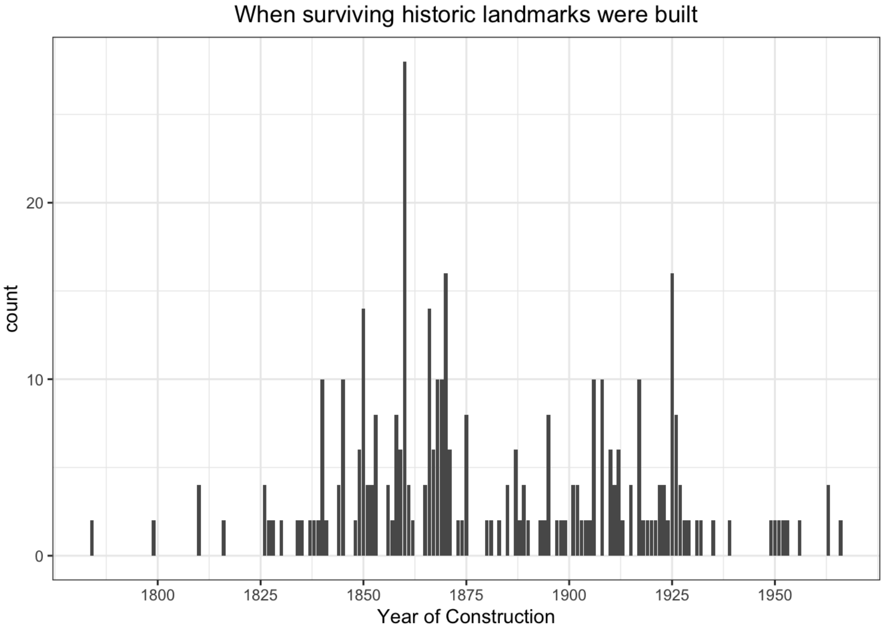
```
## Future todo -- link musicians' birthdays to time-lapse on kepler map.  (Not easy)
```

The rest of the work in building the map takes place over on kepler.gl, and consists of adding layers to the map. The main part for our purposes is to add a filter layer so that the map only shows landmarks constructed within the filtered date range, which can then be part of a time-loop showing the viewer when different areas of the parish were being developed over the last 225 years.

Map with landmarks, jazz houses, Mississippi river, Orleans water features, and parish boundary (https://kepler.gl/demo/map?mapUrl=https://dl.dropboxusercontent.com/s/zzyxejqwsl7tta8/keplergl_iz7oex.json)

Here’s a quick look at the distribution of construction dates for surviving landmarks, which is shown at the bottom of the kepler.gl map in the moving timeline.

```
# focus on known dates (about 2/3 of the landmarks)
# Some other day, use KNN (based on lat/lon) to estimate dates for the other 1/3
known_dates <- landpoints %>%
  filter(!is.na(datetime))
# Need to convert long datetimes back to just integer years, to plot
intYears <- as.integer(format(known_dates$datetime, format = '%Y'))
ggplot(data = NULL, aes(intYears)) +
  geom_bar() +
  xlab("Year of Construction") +
  scale_x_continuous(breaks = seq(1800, 1950, by = 25)) +
  ggtitle('When surviving historic landmarks were built') +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Since these are historic landmarks, they should tend to be mostly older (which they are), but the 19th century seems to have reached its peak pre-Civil War.
The fact that development drops off sharply after that war suggests an outflow of population from New Orleans when slavery ended.

The next, smaller wave of development looks like it crested in the middle of the 1920's, when New Orleans Jazz was at its peak of popularity, and during an economic boom.

Tools and sources used for Part 3:

- https://mygeodata.cloud (https://mygeodata.cloud), to convert shapefiles (.shp, amongst other suffixes) to geoJSON MultiPolygons. Those polygons are for mapping anything more complicated than a geo-point (lat/lon), and don't translate well to csv's. Mygeodata is a really useful tool, although they are only free for 3 conversions each month.
- https://Data.NOLA.gov (https://Data.NOLA.gov), linked to earlier, has a lot of valuable data, free. It also has a REST API, if you need to make requests for an app, e.g.
- https://kepler.gl (https://kepler.gl) is a great way to visualize geo-data.

Project Conclusions

In this project we gathered data on the history of New Orleans from a variety of sources and formats. By transforming this data into tidy R dataframes, we were able to provide some meaningful context to the city and the story of how it has changed due to Hurricane Katrina. Gathering this data together in a single format made it more accessible for analysis. This work also serves as an example for future work in R with GeoJSON, HTML, and Excel datasets.

We drew several key observations from working together on this project. They include the following:

1. Well framed research questions enabled us to effectively locate and extract data from relevant sources. Even so, variations in data quality and provenance remain an issue, particularly when information is curated by the public (e.g., Wikipedia).
2. Regular synchronous and asynchronous meetings among team members allowed for efficient workflows and benchmarks.
3. Further efficiencies may be achieved by establishing a report template/format and file nomenclature at the start of a project.

We expect that investigating complex systems, like a changing city, will almost always require gathering together datasets initially designed with other purposes in mind. We look forward to continuing to explore best practices for transforming and visualizing data from a wide variety of sources.