Data Wrangling

Storm Events

The storm events data was imported from the database as a DataFrame, querying for yearmonth, day, parish name, event type, latitude, longitude, direct injuries, indirect injuries, direct deaths, indirect deaths, and property damage for the state of Louisiana. FIPS data was also imported into a separate DataFrame for later computations (state FIPS, parish FIPS, parish, latitude, longitude).

	yearmonth	day	parish	event_type	lat	lon	injuries_direct	injuries_indirect	deaths_direct	deaths_indirect	damage_property
0	195703	3	None	Thunderstorm Wind	30.50	-92.18	0	0	0	0	0
1	195906	8	None	Thunderstorm Wind	29.68	-90.18	0	0	0	0	0
2	195909	21	None	Thunderstorm Wind	32.28	-93.40	0	0	0	0	0
3	196005	6	None	Thunderstorm Wind	30.08	-90.18	0	0	0	0	0
4	196103	20	None	Thunderstorm Wind	29.00	-89.40	0	0	0	0	0

Date

Since yearmonth and day are separate columns, I defined a function to separate the year and month into their own respective columns and converted the separate year, month, and day columns into a single datetime column, date.

Parish

The parishes were in all caps, so they were reformatted to title case. There were two odd parish names, ‘Laz038>040 - 056>070’ and ‘Laz067 - 070’, for only two rows, so I dropped them. There was also a typo, ‘Rapdies’ (supposed to be ‘Rapides) and was corrected.

There were several entries for parish that weren’t exact parish names, mainly including a region within the parish (e.g., East Cameron vs Cameron). I used the FIPS DataFrame to check the FIPS code for each of the entries and compared them with the actual parish FIPS codes:

```
East Cameron: [74]
Lower Jefferson: [68]
Lower Lafourche: [67]
Lower Plaquemines: [69]
Lower St. Bernard: [70]
Lower St. Martin: [55]
Lower Terrebonne: [66]
Northern Tangipahoa: [71]
Sabine And Natchitoches: [85]
Southern Tangipahoa: [72]
Upper Jefferson: [61]
Upper Lafourche: [59]
Upper Plaquemines: [63]
Upper St. Bernard: [64]
Upper St. Martin: [45]
Upper Terrebonne: [65]
West Cameron: [73]
```

```
Cameron: [23 51]
Jefferson: [51]
Lafourche: [57]
Plaquemines: [75]
St. Bernard: [87]
St. Martin: [99]
Terrebonne: [109]
Tangipahoa: [ 38 105]
Sabine: [17 85]
Natchitoches: [18 69]
```

Then, I checked for entries with coordinates and only one had them, Sabine and Natchitoches. I used the coordinates2politics API to find the parish based on the coordinates, and Sabine was returned. Some of the entries had invalid FIPS codes (from comparing with the Louisiana parish Wikipedia). I attempted to create a choropleth with the codes to see if anything came up. None of the codes worked, so I changed the names to remove the region and contain only the parish.

I then looked at the remaining null parishes (52 rows) and tried to find the parish using the coordinates2politics API again. 14 rows returned no parish data, and they were dropped.

Finally, I created a dictionary with parishes as keys and the correct FIPS codes as values and mapped it to the DataFrame to create a FIPS column and dropped the latitude and longitude

columns. There were a few rows with null FIPS, but they were counties outside of Louisiana and were dropped.

Damage

The property damage column consisted of strings with nulls, 0, or values appended by K (thousand), M (million), and B (billion). I defined a function to convert the strings into float values.

Monthly Aggregates by Parish

Finally, I needed to resample each parish by month. Event type was aggregated by count and renamed to event count, while the remaining columns were aggregated by sum. The date column was converted to month-year periods.

Final Storm Events DataFrame

	date	parish	fips	event_count	injuries_direct	injuries_indirect	deaths_direct	deaths_indirect	damage_property
0	1952-04	Acadia	22001	1	0	0	0	0	250000.0
1	1952-05	Acadia	22001	0	0	0	0	0	0.0
2	1952-06	Acadia	22001	0	0	0	0	0	0.0
3	1952-07	Acadia	22001	0	0	0	0	0	0.0
4	1952-08	Acadia	22001	0	0	0	0	0	0.0

Meteorological Data

	station	name	latitude	longitude	elevation	date	awnd	cldd	dp1x	emnt	...	evap	htdd	mnpn	mxpn	prcp	tavg	tmax	tmin	wdmv
0	USC00168181	ST MARTINVILLE 3 SW, LA US	30.0858	-91.8694	9.1	1985-03	NaN	NaN	3.0	NaN	...	NaN	NaN	NaN	NaN	5.68	NaN	NaN	NaN	NaN
1	USC00168181	ST MARTINVILLE 3 SW, LA US	30.0858	-91.8694	9.1	1985-04	NaN	NaN	0.0	NaN	...	NaN	NaN	NaN	NaN	1.12	NaN	NaN	NaN	NaN
2	USC00168181	ST MARTINVILLE 3 SW, LA US	30.0858	-91.8694	9.1	1985-05	NaN	NaN	2.0	NaN	...	NaN	NaN	NaN	NaN	3.78	NaN	NaN	NaN	NaN
3	USC00168181	ST MARTINVILLE 3 SW, LA US	30.0858	-91.8694	9.1	1985-06	NaN	NaN	0.0	NaN	...	NaN	NaN	NaN	NaN	1.17	NaN	NaN	NaN	NaN
4	USC00168181	ST MARTINVILLE 3 SW, LA US	30.0858	-91.8694	9.1	1985-07	NaN	NaN	2.0	NaN	...	NaN	NaN	NaN	NaN	4.58	NaN	NaN	NaN	NaN

5 rows × 24 columns

Date

The date column consisted of strings, so it was converted to datetime and then to month-year periods.

Parish/FIPS from Coordinates

I created a new DataFrame with name, latitude, and longitude from the original DataFrame and dropped duplicates. I used multi-processing (pandarallel library) to use the coordinates2poltiics API to query for each parish based on the coordinates of each station. I then added a FIPS code column by mapping with the previous parish-FIPS dictionary. Finally, this DataFrame was merged with the original DataFrame, and the station, latitude, longitude, and elevation columns were dropped.

The stations with null FIPS after this process were primarily out of state with the exception of two that were either an uninhabited parish or a bridge. These rows were dropped.

Missing Values

I wrote a for loop to print the percentage of data missing for each column and added the variable names to a list if more than 20% of the data was missing.

```
name   :   0.0% missing
date   :   0.0% missing
parish :   0.0% missing
fips   :   0.0% missing
awnd   :  96.4% missing
cldd   :  50.2% missing
dp1x   :   2.5% missing
emnt   :  49.9% missing
emsd   :  17.3% missing
emsn   :  16.1% missing
emxp   :   2.5% missing
emxt   :  49.7% missing
evap   :  98.2% missing
htdd   :  50.2% missing
mnpn   :  99.4% missing
mxpn   :  99.4% missing
prcp   :   1.9% missing
tavg   :  50.1% missing
tmax   :  49.7% missing
tmin   :  49.9% missing
wdmv   :  98.0% missing
wsfg   :  97.9% missing

Check: ['awnd', 'cldd', 'emnt', 'emxt', 'evap', 'htdd', 'mnpn', 'mxpn', 'tavg', 'tmax', 'tmin', 'wdmv', 'wsfg']
```

Using this list, I wrote another for loop to count how many parishes had data for each variable and if more than half of the parishes was missing, I dropped the variable.

```
64 total parishes

awnd 16 parishes with values
cldd 62 parishes with values
emnt 62 parishes with values
emxt 62 parishes with values
evap  9 parishes with values
htdd 62 parishes with values
mnpn  7 parishes with values
mxpn  7 parishes with values
tavg 62 parishes with values
tmax 62 parishes with values
tmin 62 parishes with values
wdmv 10 parishes with values
wsfg  9 parishes with values

Drop vars: ['awnd', 'evap', 'mnpn', 'mxpn', 'wdmv', 'wsfg']
```

To further investigate, I checked which parishes had missing data for each variable and interestingly, St. James and Assumption were the only two with missing data for each of the variables.

```
cldd missing data for parishes: ['St. James' 'Assumption']
emnt missing data for parishes: ['St. James' 'Assumption']
emxt missing data for parishes: ['St. James' 'Assumption']
htdd missing data for parishes: ['St. James' 'Assumption']
tavg missing data for parishes: ['St. James' 'Assumption']
tmax missing data for parishes: ['St. James' 'Assumption']
tmin missing data for parishes: ['St. James' 'Assumption']
```

Aggregating Station Data to Parish

I wrote a function to find the means using data from all stations within a parish during each month and again used multi-processing.

Final Meteorological DataFrame

	date	parish	fips	cldd	dp1x	emnt	emsd	emsn	emxp	emxt	htdd	prcp	tavg	tmax	tmin
0	1985-03	St. Martin	22099	NaN	2.5	NaN	0.0	0.0	1.850	NaN	NaN	5.265	NaN	NaN	NaN
1	1985-04	St. Martin	22099	NaN	0.0	NaN	0.0	0.0	0.500	NaN	NaN	1.500	NaN	NaN	NaN
2	1985-05	St. Martin	22099	NaN	2.0	NaN	0.0	0.0	2.050	NaN	NaN	4.370	NaN	NaN	NaN
3	1985-06	St. Martin	22099	NaN	0.5	NaN	0.0	0.0	1.190	NaN	NaN	1.765	NaN	NaN	NaN
4	1985-07	St. Martin	22099	NaN	3.0	NaN	0.0	0.0	2.625	NaN	NaN	8.160	NaN	NaN	NaN

Merged Storm Events and Meteorological DataFrames

	event_count	injuries_direct	injuries_indirect	deaths_direct	deaths_indirect	damage_property	cldd	dp1x	emnt	emsd
count	47282.000000	47282.000000	47282.000000	47282.000000	47282.000000	4.728200e+04	41434.000000	49186.000000	41623.000000	45313.000000
mean	0.495664	0.069286	0.001734	0.024766	0.000508	1.435434e+06	209.530285	1.564516	42.482859	0.025492
std	1.299436	1.849473	0.287639	3.011663	0.027589	1.185391e+08	203.163531	1.266367	16.650116	0.272581
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000	0.000000	-12.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	18.000000	0.666667	28.500000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	127.000000	1.333333	40.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000e+00	409.000000	2.000000	57.666667	0.000000
max	35.000000	266.000000	62.000000	638.000000	3.000000	2.146000e+10	803.400000	10.000000	77.000000	11.000000

emsn	emxp	emxt	htdd	prcp	tavg	tmax	tmin
45645.000000	49186.000000	41699.000000	41434.000000	49278.000000	41502.000000	41699.000000	41623.000000
0.028205	1.853123	87.767291	156.613268	4.868936	66.657962	77.394576	55.905983
0.293811	1.207345	8.177539	195.257073	3.167266	12.353092	12.125166	12.836015
0.000000	0.000000	23.000000	0.000000	0.000000	33.800000	21.700000	23.100000
0.000000	1.080000	81.000000	0.000000	2.600000	56.000000	67.100000	44.766667
0.000000	1.605000	88.000000	55.000000	4.287750	67.361905	78.800000	56.100000
0.000000	2.304750	94.500000	286.500000	6.450000	78.700000	88.600000	68.400000
11.000000	15.000000	112.000000	967.000000	33.400000	90.920000	104.320000	80.300000

I did an outer merge to include meteorological data where there was no storm data and filled NaN storm data values with 0 (assuming that no storm data for a particular month means there was no storm events for that month) and missing meteorological data were handled later. I dropped columns that were likely not useful and/or were missing a significant amount of data (cooling and heating degree days, extreme min and max temperature, highest daily snow depth, highest daily snowfall). Then, to handle the remaining missing data, I created a dictionary with nested dictionaries for each variable. This contained the means for each variable for each parish

and month. NaN values were replaced using this dictionary. After this, there were still some rows with missing data, and as discovered earlier, were from Assumption and St. James parishes. The number of rows for these parishes and the number of rows with missing values were equal, so these parishes were dropped. I also renamed the meteorological columns into a more readable form ('highest_daily_total_precip', 'total_precip', 'avg_temp', 'mean_max_temp', 'mean_min_temp').

Parish Coordinates Data

I added interpolated centroid coordinates for each parish using data from https://en.wikipedia.org/wiki/User:Michael_J/County_table. The coordinates had to be formatted to remove the degree and plus symbols and then convert the string values into floats.

Region Data

I created a new column, region, using data from https://www.lsuagcenter.com/portals/our_offices/Regions. Finally, I converted the parish and region parishes to category data type.

Final Combined DataFrame

	date	parish	lat	lon	fips	event_count	injuries_direct	injuries_indirect	deaths_direct	deaths_indirect	damage_property
0	1952-04	Acadia	30.291497	-92.411037	22001	1	0	0	0	0	250000.0
1	1952-05	Acadia	30.291497	-92.411037	22001	0	0	0	0	0	0.0
2	1952-06	Acadia	30.291497	-92.411037	22001	0	0	0	0	0	0.0
3	1952-07	Acadia	30.291497	-92.411037	22001	0	0	0	0	0	0.0
4	1952-08	Acadia	30.291497	-92.411037	22001	0	0	0	0	0	0.0

highest_daily_total_precip	total_precip	avg_temp	mean_max_temp	mean_min_temp
3.410	8.560	63.9	74.5	53.2
4.425	6.540	73.9	84.6	63.2
0.795	1.820	82.3	92.5	72.1
3.425	10.530	81.9	90.5	73.3
0.565	1.375	82.4	91.4	73.3