What are the most serious meteorological disasters in the United States?

Introduction

Business Context. While natural *events* often cannot be avoided, the risks they present can be managed, either by mitigation, avoidance, or insurance, in order to prevent them from becoming natural *disasters*. The consultancy firm you work for has been hired by an independent advocacy group that wants to conduct an analysis of the US emergency management system of preparedness, protection, mitigation, response, and recovery, with the purpose of proposing legislative reforms to make it more effective and financially efficient. Their ultimate goal is to help increase the government's ability to prevent disasters from happening and reduce the negative impact of those that cannot be completely avoided.

Business Problem. Your client would like to know which storm event types are more likely to become disasters, and in which locations, as measured by the number of deaths, injuries, and economic damage they cause. Additionally, they would like to conduct a preliminary assessment of whether the Post-Katrina Emergency Management Reform Act of 2006 had any impact on the severity of the disasters that occurred after the bill was signed. This Act centralized the US emergency management under the coordination of the Federal Emergency Management Agency (FEMA) as a response to the enormous human and material losses that were caused by Hurricane Katrina in August 2005.

Analytical Context. You have been provided with a compressed GZIP file of storm events from 1970 to 2020 as recorded by the US National

In [1]:
```
# Importing relevant libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import folium
import pingouin as pg
```

In [2]:
```
import plotly.graph_objects as go
import numpy as np
```

In [3]:
```
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('retina')
```

In [4]:
```
%%javascript
IPython.OutputArea.prototype._should_scroll = function(lines) {
    return false;
}
```

In [5]:
```
pd.set_option('display.max_columns', None)
```

Loading in the dataset

Let's load in the dataset. We add the `parse_dates` argument to tell pandas which columns should be interpreted as dates.

In [6]:

```python
df = pd.read_csv("data/dataset.csv.gz", parse_dates=["BEGIN_DATE_TIME", "END_DATE_TIME"])
df.head()
```

Out[6]:

	EPISODE_ID	EVENT_ID	STATE	EVENT_TYPE	BEGIN_DATE_TIME	BEGIN_YEAR	CZ_TIMEZONE	END_DATE_TIME	TOR_F_SCALE	BEGIN_LOC
0	NaN	9987739	COLORADO	hail	1983-07-22 16:40:00	1983.0	CST	1983-07-22 16:40:00	NaN	
1	NaN	9987740	COLORADO	hail	1983-07-22 16:45:00	1983.0	CST	1983-07-22 16:45:00	NaN	
2	NaN	9987741	COLORADO	hail	1983-07-22 16:45:00	1983.0	CST	1983-07-22 16:45:00	NaN	
3	NaN	9987735	COLORADO	hail	1983-07-22 16:20:00	1983.0	CST	1983-07-22 16:20:00	NaN	
4	NaN	9987736	COLORADO	hail	1983-07-22 16:25:00	1983.0	CST	1983-07-22 16:25:00	NaN	

In [7]:

```python
len(df)
```

Out[7]:

```
2483191
```

Here is a description of the imported columns:

1. **EPISODE_ID**: The storm episode ID. A single episode can contain multiple events
2. **EVENT_ID**: This is the ID of the actual storm as such. Several storms can be grouped into an episode
3. **STATE**: The state or region where the event occurred
4. **EVENT_TYPE**: The type of the event
5. **BEGIN_DATE_TIME**: The date and time when the event started. Times and dates are in LST (Local Solar Time), which means that they reflect the local time, not a coordinated time
6. **BEGIN_YEAR**: The year in which the event begun
7. **CZ_TIMEZONE**: The timezone of the place where the event occurred
8. **END_DATE_TIME**: The date and time when the event ended. Times and dates are in LST (Local Solar Time), which means that they reflect the local time, not a coordinated time
9. **TOR_F_SCALE**: The enhanced Fujita scale (highest recorded value). This scale measures the strength of a tornado based on the amount of damage that it caused. A level of `EF0` means "light damage" (wind speeds of 40 - 72 mph), and a level of `EF5` means "incredible damage" (261 - 318 mph). `EFU` means "Unknown"
10. **BEGIN_LOCATION**: The name of the city or village where the event started
11. **END_LOCATION**: The name of the city or village where the event ended
12. **BEGIN_LAT**: The latitude of the place where the event begun

```
visualization_choice()
```

```
Out[9]: {'histogram': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16],
         'barplot': [1, 2, 4, 5, 8, 9],
         'boxplot': [1, 2, 4, 5, 8, 9],
         'geoheatmap': [10, 11, 12, 13],
         'lineplot': [7],
         'densityplot': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]}
```

Exercise 2

2.1

As discussed in the regular cases, it is important to compute summary statistics for each feature. We know that the damage caused by natural disasters usually has a distribution that is very skewed to the right. Which TWO of the following summary statistics would be MOST useful to confirm that this is the case for the `TOTAL_DEATHS` , `TOTAL_INJURIES` , and `TOTAL_DAMAGE_DEFLATED` columns?

A. Count
B. Mean
C. Minimum
D. Maximum
E. 25th percentile
F. Median
G. 75th percentile

In [10]:
```
def summary_choice():
    """
    Returns the two options you chose.

    Uncomment the lines that correspond to your choice.
    """

    A = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].count()
    B = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].mean()
    C = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].min()
    D = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].max()
    E = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].quantile(0.25)
    F = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].quantile(0.5)
    G = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].quantile(0.75)


    answer = {}
    #answer["A"] = A
    answer["B"] = B
    #answer["C"] = C
    #answer["D"] = D
    #answer["E"] = E
    answer["F"] = F
    #answer["G"] = G

    # YOUR CODE HERE

    return answer
```

2.2

Inspect the tables of the two summary statistics you chose. What can you interpret from your results?

Note: In this and all similar subsequent exercises, please answer in the cell that is immediately below this one (if you write in the same cell as the question, your answer will not be recorded).

Your answer here.

In [11]:
```
B = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].mean()
F = df.groupby(["EVENT_TYPE"])[["TOTAL_DEATHS", "TOTAL_INJURIES", "TOTAL_DAMAGE_DEFLATED"]].quantile(0.5)
```

In [12]:
```
print(B["TOTAL_DEATHS"].mean(), B["TOTAL_INJURIES"].mean(), B["TOTAL_DAMAGE_DEFLATED"].mean())
```
```
0.5078558350667536 0.2975035000686944 18885141.86965571
```

In [13]:

```
print(F["TOTAL_DEATHS"].mean(), F["TOTAL_INJURIES"].mean(), F["TOTAL_DAMAGE_DEFLATED"].mean())
```

```
0.43548387096774194 0.06451612903225806 17887551.396892674
```

- The means of TOTAL_DEATHS, TOTAL_INJURIES, and TOTAL_DAMAGE_DEFLATED in the groupby mean (B) are more than the means of the groupby medians (F); indicating that the overall data is skewed to the right. "If you have a distribution that is skewed to the right like this one, the mean will be more to the right than the median" (case_1.2_lecture).

Exercise 3

Let's now look at some bivariate distributions. This is a pair plot of some of the numeric variables of the dataset:

In [22]:
```python
# YOUR CODE HERE
# Using #1 as an example
# Create BEGIN_MONTH and BEGIN_YEAR column (adding BEGIN_YEAR to explore more, if needed)

df['BEGIN_MONTH'] = df['BEGIN_DATE_TIME'].dt.strftime('%m')
df['BEGIN_YEAR'] = df['BEGIN_DATE_TIME'].dt.strftime('%Y')

month_dict = {'01':'JANUARY', '02':'FEBRUARY', '03':'MARCH', '04':'APRIL', '05':'MAY', '06':'JUNE',
            '07':'JULY', '08':'AUGUST', '09':'SEPTEMBER', '10':'OCTOBER', '11':'NOVEMBER', '12':'DECEMBER'}

df["BEGIN_MONTH"] = df["BEGIN_MONTH"].replace(month_dict)
```

In [23]:
```python
# Groupby EVENT_TYPE and MONTH

type_by_month = df.groupby("BEGIN_MONTH")["EVENT_TYPE"].value_counts(ascending = False)
month_by_type = df.groupby("EVENT_TYPE")["BEGIN_MONTH"].value_counts(ascending = False)
```

In [24]:
```python
month_by_type.head()
```

Out[24]:
```
EVENT_TYPE              BEGIN_MONTH
astronomical low tide   FEBRUARY       196
                        JANUARY        159
                        DECEMBER       100
                        MARCH           50
                        NOVEMBER        18
Name: BEGIN_MONTH, dtype: int64
```

In [25]:
```python
type_by_month.head()
```

Out[25]:
```
BEGIN_MONTH  EVENT_TYPE
APRIL        hail                  52870
             thunderstorm wind     36460
             high wind              9170
             tornado                8839
             flood                  6518
Name: EVENT_TYPE, dtype: int64
```

In [26]:
```python
# Crosstab BEGIN_MONTH and EVENT_TYPE and normalize BEGIN_MONTH

type_month_cross = pd.crosstab(index=df["EVENT_TYPE"], columns=df["BEGIN_MONTH"], normalize="columns")*100
type_month_cross.head()
```

Out[26]:

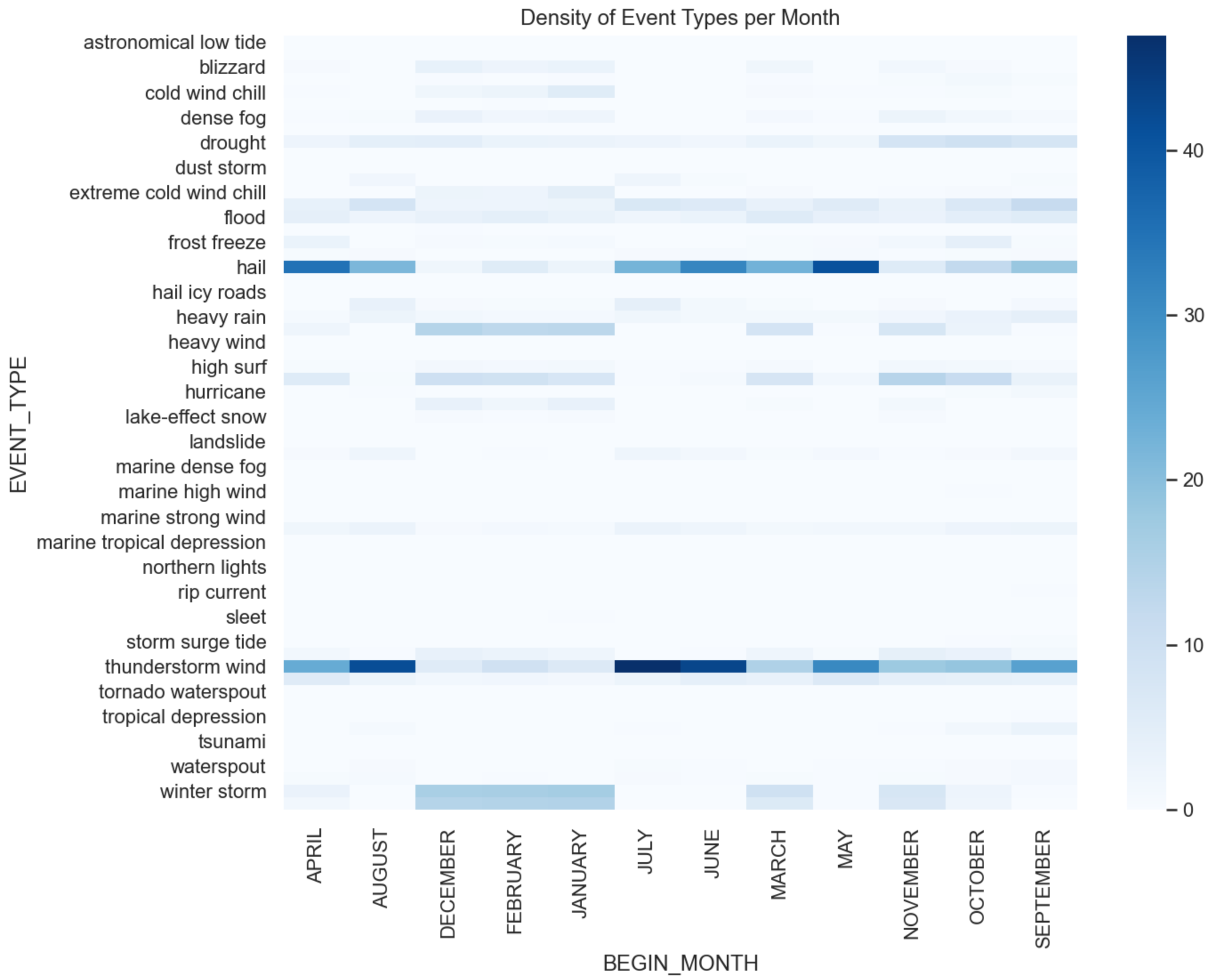
BEGIN_MONTH	APRIL	AUGUST	DECEMBER	FEBRUARY	JANUARY	JULY	JUNE	MARCH	MAY	NOVEMBER	OCTOBER	SEPTEMI

EVENT_TYPE											
astronomical low tide	0.009230	0.002761	0.110003	0.182906	0.137653	0.000971	0.000000	0.044584	0.003974	0.029525	0.004946	0.006
avalanche	0.032304	0.000690	0.112203	0.150244	0.137653	0.000000	0.000419	0.096302	0.007949	0.039367	0.011542	0.003
blizzard	0.671123	0.000000	3.666384	2.469228	3.067320	0.002427	0.000000	1.954577	0.021363	1.594357	0.557305	0.006
coastal flood	0.166133	0.065565	0.326708	0.155843	0.219898	0.016507	0.084623	0.264831	0.068559	0.485524	1.188808	0.437
cold wind chill	0.294688	0.051071	2.013046	2.598009	5.668871	0.075250	0.097191	0.675899	0.236975	0.365784	0.547412	0.178

In [27]:

```python
# Seaborn Heatmap

sns.set(rc = {'figure.figsize':(10,8)})
ax = sns.heatmap(type_month_cross, cmap = "Blues")
ax.set_title("Density of Event Types per Month");
```



The Post-Katrina Emergency Management Reform Act of 2006

Exercise 8 (hard)

8.1

Conduct a hypothesis test for each event type to assess whether there is a difference in average total damage when comparing disasters that happened before the reform to those that happened after. Keep only the event types for which you found a significant difference (using a significance threshold of $\alpha = 0.01$). Since it is likely that not all events that have happened in the US are present in this dataset, we can interpret the data as being a sample (conducting hypothesis tests on population data would not make sense).

Attribution

"Storm Events Database", 20 Nov 2020, National Oceanic and Atmospheric Administration, Licensed under the Freedom of Information Act, https://www.ncdc.noaa.gov/stormevents/ftp.jsp

"F5 tornado Elie Manitoba 2007", 22 Jun 2017, Justin Hobson, Creative Commons Attribution-Share Alike 3.0 Unported license, https://commons.wikimedia.org/wiki/File:F5_tornado_Elie_Manitoba_2007.jpg

"Historical Consumer Price Index for All Urban Consumers (CPI-U): U.S. city average, all items, index averages", March, 2021, Licensed under the Freedom of Information Act, https://www.bls.gov/cpi/tables/supplemental-files/historical-cpi-u-202103.pdf (the value for 2021 corresponds to the three-month average between January and March).