

Visualizing U.S. Natural Disaster Declarations

Milestone – 1: Exploratory Data Analysis

Name: Anirudh Mamilla
Domain: Data Visualization
Dataset: FEMA Disaster Declarations Summary
Tools: Python, Pandas, Matplotlib, Seaborn

```
In [15]: import pandas as pd
df = pd.read_csv(
    "data/DisasterDeclarationsSummaries.csv",
    low_memory=False
)
df.head()
```

Out[15]:

	femaDeclarationString	disasterNumber	state	declarationType	declarationDate	fyD
0	FM-5529-OR	5529	OR	FM	2024-08-09T00:00:00.000Z	
1	FM-5528-OR	5528	OR	FM	2024-08-06T00:00:00.000Z	
2	FM-5527-OR	5527	OR	FM	2024-08-02T00:00:00.000Z	
3	DR-4312-CA	4312	CA	DR	2017-05-02T00:00:00.000Z	
4	DR-4251-AL	4251	AL	DR	2016-01-21T00:00:00.000Z	

5 rows × 28 columns

```
In [2]: df.shape

Out[2]: (68542, 28)
```

```
In [3]: df.columns
```

```
Out[3]: Index(['femaDeclarationString', 'disasterNumber', 'state', 'declarationType',  
              'declarationDate', 'fyDeclared', 'incidentType', 'declarationTitle',  
              'ihProgramDeclared', 'iaProgramDeclared', 'paProgramDeclared',  
              'hmProgramDeclared', 'incidentBeginDate', 'incidentEndDate',  
              'disasterCloseoutDate', 'tribalRequest', 'fipsStateCode',  
              'fipsCountyCode', 'placeCode', 'designatedArea',  
              'declarationRequestNumber', 'lastIAFilingDate', 'incidentId', 'region',  
              'designatedIncidentTypes', 'lastRefresh', 'hash', 'id'],  
              dtype='object')
```

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 68542 entries, 0 to 68541  
Data columns (total 28 columns):  
#   Column                                Non-Null Count  Dtype  
---  ---                                -  
0   femaDeclarationString                 68542 non-null  object  
1   disasterNumber                       68542 non-null  int64  
2   state                                68542 non-null  object  
3   declarationType                      68542 non-null  object  
4   declarationDate                      68542 non-null  object  
5   fyDeclared                           68542 non-null  int64  
6   incidentType                         68542 non-null  object  
7   declarationTitle                    68542 non-null  object  
8   ihProgramDeclared                   68542 non-null  int64  
9   iaProgramDeclared                   68542 non-null  int64  
10  paProgramDeclared                   68542 non-null  int64  
11  hmProgramDeclared                   68542 non-null  int64  
12  incidentBeginDate                   68542 non-null  object  
13  incidentEndDate                     68018 non-null  object  
14  disasterCloseoutDate                52939 non-null  object  
15  tribalRequest                       68542 non-null  int64  
16  fipsStateCode                       68542 non-null  int64  
17  fipsCountyCode                     68542 non-null  int64  
18  placeCode                           68542 non-null  int64  
19  designatedArea                      68542 non-null  object  
20  declarationRequestNumber             68542 non-null  int64  
21  lastIAFilingDate                    19426 non-null  object  
22  incidentId                          68542 non-null  int64  
23  region                              68542 non-null  int64  
24  designatedIncidentTypes              20730 non-null  object  
25  lastRefresh                         68542 non-null  object  
26  hash                                68542 non-null  object  
27  id                                  68542 non-null  object  
dtypes: int64(13), object(15)  
memory usage: 14.6+ MB
```

```
In [5]: df['declarationDate'] = pd.to_datetime(df['declarationDate'], errors='coerce')
```

```
In [6]: df['incidentType'] = df['incidentType'].fillna("Unknown")
```

```
In [7]: df = df.dropna(subset=['state'])
```

```
In [8]: df['year'] = df['declarationDate'].dt.year
```

```
In [9]: df['year'].value_counts().sort_index()
```

```
Out[9]: year
1953      13
1954      17
1955      18
1956      16
1957      16
...
2021     1962
2022     1141
2023     1119
2024     2364
2025     1196
Name: count, Length: 73, dtype: int64
```

```
In [10]: df['state'].value_counts().head(10)
```

```
Out[10]: state
TX      5386
KY      3235
MO      2830
FL      2791
GA      2653
VA      2621
LA      2589
OK      2582
NC      2329
PR      2116
Name: count, dtype: int64
```

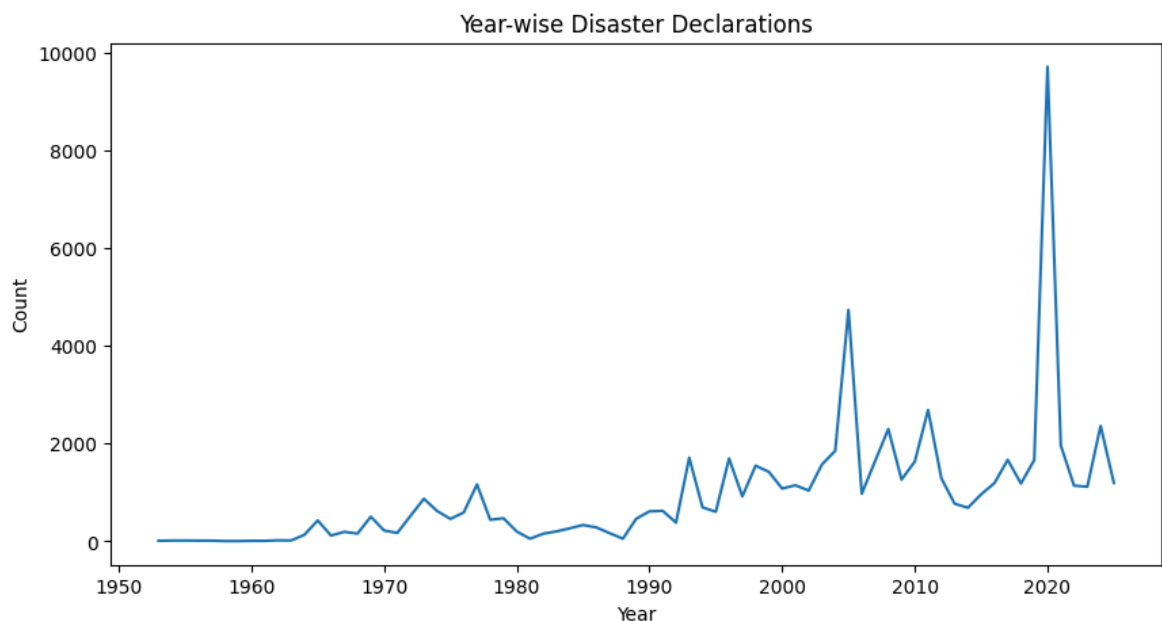
```
In [11]: df['incidentType'].value_counts()
```

```
Out[11]: incidentType
Severe Storm      19299
Hurricane         13721
Flood            11227
Biological        7857
Fire             3844
Snowstorm        3707
Severe Ice Storm  2956
Tornado          1623
Drought          1292
Tropical Storm   1059
Coastal Storm    637
Other            313
Freezing         301
Earthquake       228
Winter Storm     149
Typhoon          130
Volcanic Eruption  51
Mud/Landslide    44
Fishing Losses   42
Dam/Levee Break  13
Tsunami          9
Toxic Substances  9
Chemical         9
Human Cause      7
Tropical Depression 7
Terrorist        5
Straight-Line Winds 3
Name: count, dtype: int64
```

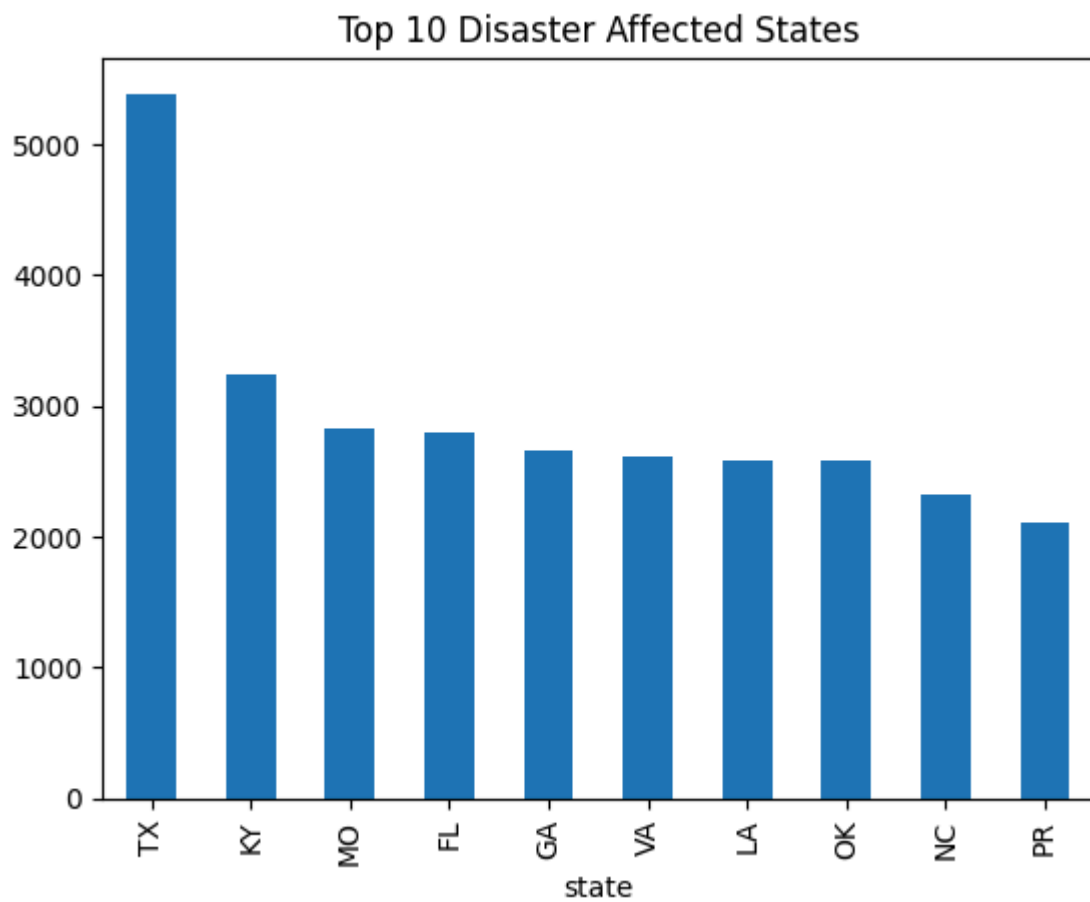
```
In [12]: import matplotlib.pyplot as plt

yearly = df['year'].value_counts().sort_index()

plt.figure(figsize=(10,5))
plt.plot(yearly)
plt.title("Year-wise Disaster Declarations")
plt.xlabel("Year")
plt.ylabel("Count")
plt.show()
```



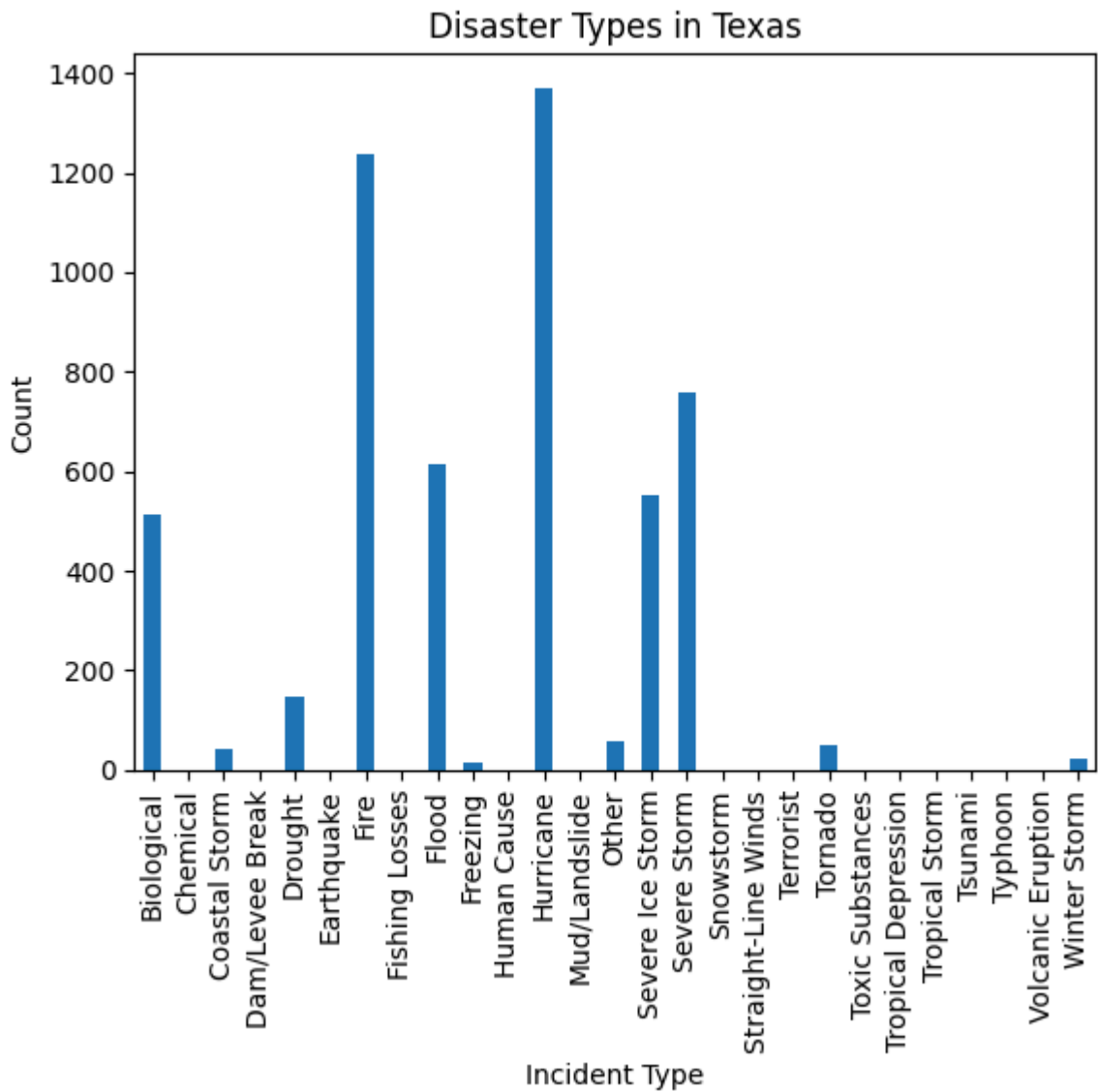
```
In [13]: df['state'].value_counts().head(10).plot(kind='bar')
plt.title("Top 10 Disaster Affected States")
plt.show()
```



```
In [14]: pd.crosstab(df['state'], df['incidentType'])

pd.crosstab(df['state'], df['incidentType']).loc['TX'].plot(kind='bar')

plt.title("Disaster Types in Texas")
plt.xlabel("Incident Type")
plt.ylabel("Count")
plt.show()
```



State vs Disaster Type Analysis (Texas)

- Severe storms dominate disaster occurrences in Texas.
- Floods and hurricanes are also significant contributors.
- The distribution highlights Texas's vulnerability to multiple disaster types.

Insight: State-level disaster profiling helps in targeted disaster preparedness and response planning.

In []: