

```
In [16]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

/Users/michaellee/opt/anaconda3/lib/python3.9/site-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.25.0

```
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
```

```
In [2]: storm_df = pd.read_csv('StormEventDetails.csv')
```

/Users/michaellee/opt/anaconda3/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3444: DtypeWarning: Columns (16,25,26,28,29,34,35,37,39,40,42,43,48,49) have mixed types. Specify dtype option on import or set low_memory=False.

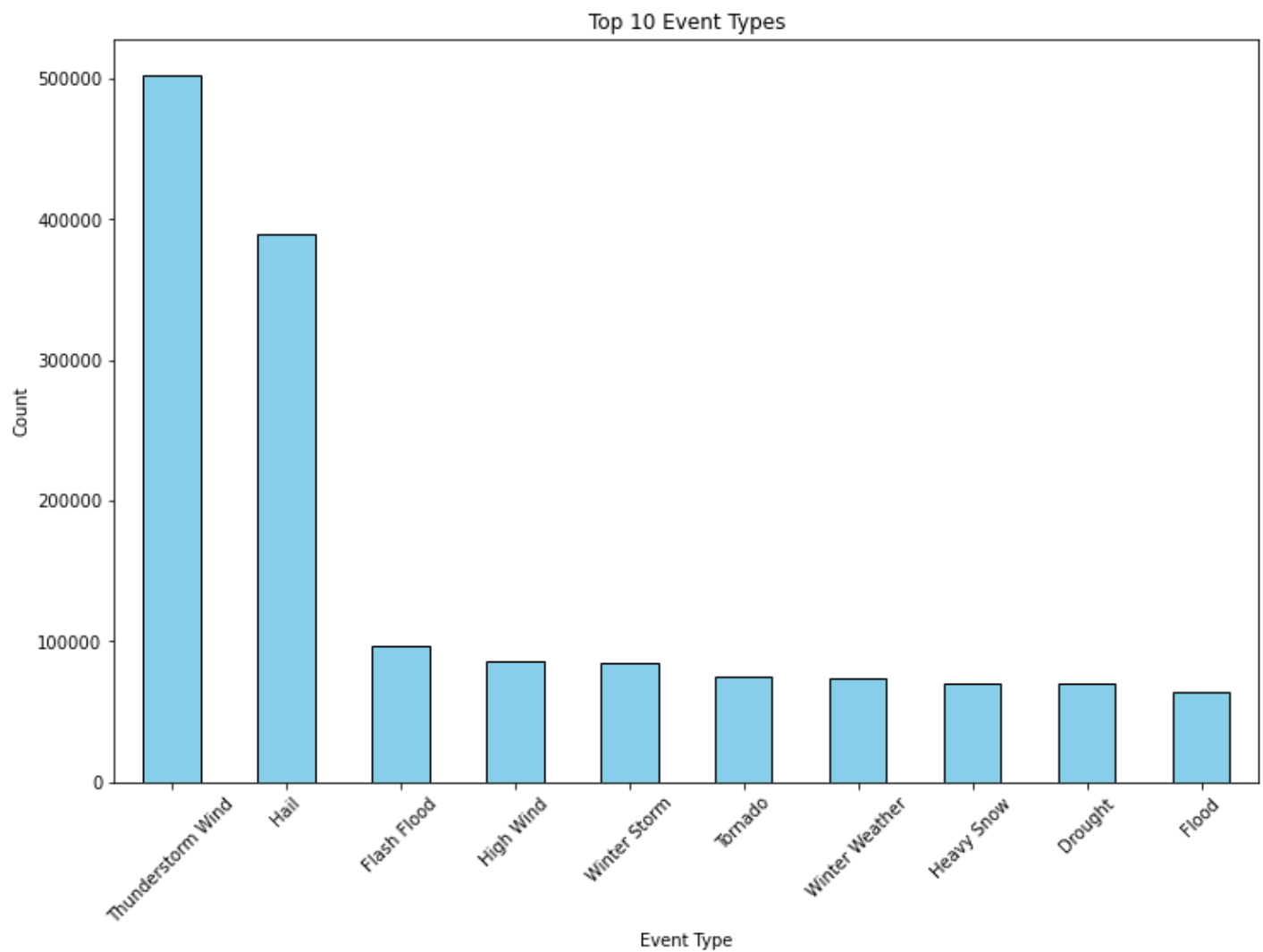
```
exec(code_obj, self.user_global_ns, self.user_ns)
```

```
In [3]: pd.set_option('display.max_columns', None)
```

```
In [4]: storm_df['EVENT_TYPE'].value_counts()
```

```
Out[4]: Thunderstorm Wind      502986
Hail                        390036
Flash Flood                 97078
High Wind                   85790
Winter Storm                84574
...
HAIL FLOODING                1
THUNDERSTORM WIND/ TREE      1
Marine Lightning             1
THUNDERSTORM WINDS/FLASH FLOOD 1
THUNDERSTORM WINDS/FLOODING  1
Name: EVENT_TYPE, Length: 70, dtype: int64
```

```
In [5]: top_10_event_types = storm_df['EVENT_TYPE'].value_counts().head(10)
top_10_event_types.plot(kind='bar', color='skyblue', edgecolor='black', figsize=(12, 8))
plt.title('Top 10 Event Types')
plt.xlabel('Event Type')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



```
In [6]: storm_df.columns
```

```
Out[6]: Index(['BEGIN_YEARMONTH', 'BEGIN_DAY', 'BEGIN_TIME', 'END_YEARMONTH',
            'END_DAY', 'END_TIME', 'EPISODE_ID', 'EVENT_ID', 'STATE', 'STATE_FIPS',
            'YEAR', 'MONTH_NAME', 'EVENT_TYPE', 'CZ_TYPE', 'CZ_FIPS', 'CZ_NAME',
            'WFO', 'BEGIN_DATE_TIME', 'CZ_TIMEZONE', 'END_DATE_TIME',
            'INJURIES_DIRECT', 'INJURIES_INDIRECT', 'DEATHS_DIRECT',
            'DEATHS_INDIRECT', 'DAMAGE_PROPERTY', 'DAMAGE_CROPS', 'SOURCE',
            'MAGNITUDE', 'MAGNITUDE_TYPE', 'FLOOD_CAUSE', 'CATEGORY', 'TOR_F_SCALE',
            'TOR_LENGTH', 'TOR_WIDTH', 'TOR_OTHER_WFO', 'TOR_OTHER_CZ_STATE',
            'TOR_OTHER_CZ_FIPS', 'TOR_OTHER_CZ_NAME', 'BEGIN_RANGE',
            'BEGIN_AZIMUTH', 'BEGIN_LOCATION', 'END_RANGE', 'END_AZIMUTH',
            'END_LOCATION', 'BEGIN_LAT', 'BEGIN_LON', 'END_LAT', 'END_LON',
            'EPISODE_NARRATIVE', 'EVENT_NARRATIVE', 'DATA_SOURCE'],
            dtype='object')
```

```
In [7]: top_10_events_index = storm_df['EVENT_TYPE'].value_counts().head(10).index
        filtered_df = storm_df[storm_df['EVENT_TYPE'].isin(top_10_events_index)]
```

```
In [53]: (filtered_df.head())
```

```
Out[53]:
```

	BEGIN_YEARMONTH	BEGIN_DAY	BEGIN_TIME	END_YEARMONTH	END_DAY	END_TIME	EPISODE_ID	EVE
0	195004	28	1445	195004	28	1445	NaN	100
1	195004	29	1530	195004	29	1530	NaN	10

	BEGIN_YEARMONTH	BEGIN_DAY	BEGIN_TIME	END_YEARMONTH	END_DAY	END_TIME	EPISODE_ID	EVE
2	195007	5	1800	195007	5	1800	NaN	10'
3	195007	5	1830	195007	5	1830	NaN	101
4	195007	24	1440	195007	24	1440	NaN	101

In [54]:

```
print(filtered_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1514338 entries, 0 to 1811209
Data columns (total 51 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   BEGIN_YEARMONTH                      1514338 non-null int64
1   BEGIN_DAY                            1514338 non-null int64
2   BEGIN_TIME                           1514338 non-null int64
3   END_YEARMONTH                        1514338 non-null int64
4   END_DAY                              1514338 non-null int64
5   END_TIME                             1514338 non-null int64
6   EPISODE_ID                           1282110 non-null float64
7   EVENT_ID                             1514338 non-null int64
8   STATE                                1514338 non-null object
9   STATE_FIPS                           1514338 non-null float64
10  YEAR                                  1514338 non-null int64
11  MONTH_NAME                           1514338 non-null object
12  EVENT_TYPE                           1514338 non-null object
13  CZ_TYPE                              1514338 non-null object
14  CZ_FIPS                              1514338 non-null int64
15  CZ_NAME                              1512781 non-null object
16  WFO                                   1388776 non-null object
17  BEGIN_DATE_TIME                      1514338 non-null object
18  CZ_TIMEZONE                          1514338 non-null object
19  END_DATE_TIME                        1514338 non-null object
20  INJURIES_DIRECT                      1514338 non-null int64
21  INJURIES_INDIRECT                   1514338 non-null int64
22  DEATHS_DIRECT                       1514338 non-null int64
23  DEATHS_INDIRECT                     1514338 non-null int64
24  DAMAGE_PROPERTY                     1052217 non-null float64
25  DAMAGE_CROPS                        867108 non-null float64
26  SOURCE                              1187873 non-null object
27  MAGNITUDE                           980194 non-null float64
28  MAGNITUDE_TYPE                       404752 non-null object
29  FLOOD_CAUSE                          104697 non-null object
30  CATEGORY                             0 non-null float64
31  TOR_F_SCALE                          73555 non-null object
32  TOR_LENGTH                           270183 non-null float64
33  TOR_WIDTH                            270183 non-null float64
34  TOR_OTHER_WFO                        3012 non-null object
35  TOR_OTHER_CZ_STATE                   3012 non-null object
36  TOR_OTHER_CZ_FIPS                    3012 non-null float64
37  TOR_OTHER_CZ_NAME                    3012 non-null object
38  BEGIN_RANGE                          911469 non-null float64
39  BEGIN_AZIMUTH                        692771 non-null object
40  BEGIN_LOCATION                       921518 non-null object
41  END_RANGE                            911193 non-null float64
42  END_AZIMUTH                          679793 non-null object
43  END_LOCATION                         882158 non-null object
44  BEGIN_LAT                            1033043 non-null float64
45  BEGIN_LON                           1033035 non-null float64
```

```
46  END_LAT      855554 non-null float64
47  END_LON      855547 non-null float64
48  EPISODE_NARRATIVE 1055447 non-null object
49  EVENT_NARRATIVE 794596 non-null object
50  DATA_SOURCE 1514323 non-null object
dtypes: float64(15), int64(13), object(23)
memory usage: 600.8+ MB
None
```

In [55]:

```
missing_values = filtered_df.isnull().sum()
print(missing_values)
```

```
BEGIN_YEARMONTH      0
BEGIN_DAY             0
BEGIN_TIME            0
END_YEARMONTH        0
END_DAY              0
END_TIME             0
EPISODE_ID           232228
EVENT_ID             0
STATE                0
STATE_FIPS           0
YEAR                 0
MONTH_NAME           0
EVENT_TYPE           0
CZ_TYPE              0
CZ_FIPS              0
CZ_NAME              1557
WFO                  125562
BEGIN_DATE_TIME      0
CZ_TIMEZONE          0
END_DATE_TIME        0
INJURIES_DIRECT      0
INJURIES_INDIRECT    0
DEATHS_DIRECT        0
DEATHS_INDIRECT      0
DAMAGE_PROPERTY      462121
DAMAGE_CROPS         647230
SOURCE               326465
MAGNITUDE            534144
MAGNITUDE_TYPE       1109586
FLOOD_CAUSE          1409641
CATEGORY             1514338
TOR_F_SCALE          1440783
TOR_LENGTH           1244155
TOR_WIDTH            1244155
TOR_OTHER_WFO        1511326
TOR_OTHER_CZ_STATE   1511326
TOR_OTHER_CZ_FIPS    1511326
TOR_OTHER_CZ_NAME    1511326
BEGIN_RANGE          602869
BEGIN_AZIMUTH        821567
BEGIN_LOCATION       592820
END_RANGE            603145
END_AZIMUTH          834545
END_LOCATION         632180
BEGIN_LAT            481295
BEGIN_LON            481303
END_LAT              658784
END_LON              658791
EPISODE_NARRATIVE    458891
EVENT_NARRATIVE      719742
DATA_SOURCE          15
dtype: int64
```

```
In [62]: missing_percentage = (filtered_df.isnull().sum() / len(filtered_df)) * 100
         print(missing_percentage)
```

```
BEGIN_YEARMONTH      0.000000
BEGIN_DAY             0.000000
BEGIN_TIME            0.000000
END_YEARMONTH         0.000000
END_DAY               0.000000
END_TIME              0.000000
EPISODE_ID           15.335282
EVENT_ID              0.000000
STATE                 0.000000
STATE_FIPS            0.000000
YEAR                  0.000000
MONTH_NAME            0.000000
EVENT_TYPE            0.000000
CZ_TYPE               0.000000
CZ_FIPS               0.000000
CZ_NAME               0.102817
WFO                   8.291544
BEGIN_DATE_TIME       0.000000
CZ_TIMEZONE           0.000000
END_DATE_TIME         0.000000
INJURIES_DIRECT       0.000000
INJURIES_INDIRECT     0.000000
DEATHS_DIRECT         0.000000
DEATHS_INDIRECT       0.000000
DAMAGE_PROPERTY       30.516371
DAMAGE_CROPS          42.740128
SOURCE                21.558265
MAGNITUDE             35.272442
BEGIN_RANGE           39.810729
BEGIN_LOCATION        39.147139
END_RANGE             39.828955
END_LOCATION          41.746294
BEGIN_LAT             31.782535
BEGIN_LON             31.783063
END_LAT               43.503102
END_LON               43.503564
EPISODE_NARRATIVE     30.303076
EVENT_NARRATIVE       47.528491
DATA_SOURCE           0.000991
dtype: float64
```

```
In [57]: threshold = 50
         columns_to_drop = missing_percentage[missing_percentage > threshold].index

         filtered_df.drop(columns=columns_to_drop, inplace=True)
```

/Users/michaellee/opt/anaconda3/lib/python3.9/site-packages/pandas/core/frame.py:4906: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
        return super().drop()
```

```
In [58]: print(filtered_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1514338 entries, 0 to 1811209
Data columns (total 39 columns):
 #   Column                Non-Null Count  Dtype
---  -

```

```

0 BEGIN_YEARMONTH      1514338 non-null int64
1 BEGIN_DAY            1514338 non-null int64
2 BEGIN_TIME           1514338 non-null int64
3 END_YEARMONTH        1514338 non-null int64
4 END_DAY              1514338 non-null int64
5 END_TIME             1514338 non-null int64
6 EPISODE_ID          1282110 non-null float64
7 EVENT_ID             1514338 non-null int64
8 STATE               1514338 non-null object
9 STATE_FIPS          1514338 non-null float64
10 YEAR               1514338 non-null int64
11 MONTH_NAME         1514338 non-null object
12 EVENT_TYPE         1514338 non-null object
13 CZ_TYPE            1514338 non-null object
14 CZ_FIPS            1514338 non-null int64
15 CZ_NAME            1512781 non-null object
16 WFO               1388776 non-null object
17 BEGIN_DATE_TIME     1514338 non-null object
18 CZ_TIMEZONE         1514338 non-null object
19 END_DATE_TIME       1514338 non-null object
20 INJURIES_DIRECT     1514338 non-null int64
21 INJURIES_INDIRECT  1514338 non-null int64
22 DEATHS_DIRECT       1514338 non-null int64
23 DEATHS_INDIRECT     1514338 non-null int64
24 DAMAGE_PROPERTY     1052217 non-null float64
25 DAMAGE_CROPS        867108 non-null float64
26 SOURCE              1187873 non-null object
27 MAGNITUDE           980194 non-null float64
28 BEGIN_RANGE         911469 non-null float64
29 BEGIN_LOCATION      921518 non-null object
30 END_RANGE           911193 non-null float64
31 END_LOCATION        882158 non-null object
32 BEGIN_LAT           1033043 non-null float64
33 BEGIN_LON           1033035 non-null float64
34 END_LAT             855554 non-null float64
35 END_LON             855547 non-null float64
36 EPISODE_NARRATIVE   1055447 non-null object
37 EVENT_NARRATIVE     794596 non-null object
38 DATA_SOURCE        1514323 non-null object
dtypes: float64(11), int64(13), object(15)
memory usage: 462.1+ MB
None

```

In [50]: `print(filtered_df[['INJURIES_DIRECT', 'INJURIES_INDIRECT', 'DEATHS_DIRECT', 'DEATHS_INDIRECT', 'DAMAGE_PROPERTY', 'DAMAGE_CROPS', 'SOURCE', 'MAGNITUDE', 'BEGIN_RANGE', 'BEGIN_LOCATION', 'END_RANGE', 'END_LOCATION', 'BEGIN_LAT', 'BEGIN_LON', 'END_LAT', 'END_LON', 'EPISODE_NARRATIVE', 'EVENT_NARRATIVE', 'DATA_SOURCE'])`

```

INJURIES_DIRECT      int64
INJURIES_INDIRECT    int64
DEATHS_DIRECT        int64
DEATHS_INDIRECT      int64
DAMAGE_PROPERTY      float64
DAMAGE_CROPS         float64
dtype: object

```

In [38]: `def convert_to_numeric(value):
 if pd.isnull(value) or value == '':
 return np.nan
 if isinstance(value, (int, float)):
 return value
 value = value.upper()
 if value == 'K':
 return 1e3
 elif value == 'M':
 return 1e6
 elif value == 'B':`

```

    return 1e9
elif value == 'H': # Assuming 'h' or 'H' represents hundreds
    return 1e2
elif 'K' in value:
    return float(value.replace('K', '')) * 1e3
elif 'M' in value:
    return float(value.replace('M', '')) * 1e6
elif 'B' in value:
    return float(value.replace('B', '')) * 1e9
elif 'H' in value: # Assuming 'h' or 'H' represents hundreds
    return float(value.replace('H', '')) * 1e2
else:
    return np.nan
return float(value)

```

In [32]:

```

non_numeric_property = filtered_df['DAMAGE_PROPERTY'].str.extract(r'([a-zA-Z]+)')[0].dropna()
non_numeric_crops = filtered_df['DAMAGE_CROPS'].str.extract(r'([a-zA-Z]+)')[0].dropna().ur

print("Non-numeric notations in DAMAGE_PROPERTY:", non_numeric_property)
print("Non-numeric notations in DAMAGE_CROPS:", non_numeric_crops)

```

```

Non-numeric notations in DAMAGE_PROPERTY: ['K' 'M' 'h' 'H' 'B']
Non-numeric notations in DAMAGE_CROPS: ['K' 'M' 'k' 'B']

```

In [39]:

```

filtered_df['DAMAGE_PROPERTY'] = filtered_df['DAMAGE_PROPERTY'].apply(convert_to_numeric)
filtered_df['DAMAGE_CROPS'] = filtered_df['DAMAGE_CROPS'].apply(convert_to_numeric)

```

```

/var/folders/wh/j6yhw1z15k31vm0ycrq37c_c0000gn/T/ipykernel_91950/3977358392.py:1: SettingW
ithCopyWarning:

```

```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

```

```

    filtered_df['DAMAGE_PROPERTY'] = filtered_df['DAMAGE_PROPERTY'].apply(convert_to_numeri
c)

```

```

/var/folders/wh/j6yhw1z15k31vm0ycrq37c_c0000gn/T/ipykernel_91950/3977358392.py:2: SettingW
ithCopyWarning:

```

```

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_gu
ide/indexing.html#returning-a-view-versus-a-copy

```

```

    filtered_df['DAMAGE_CROPS'] = filtered_df['DAMAGE_CROPS'].apply(convert_to_numeric)

```

In [40]:

```

print(filtered_df[['DAMAGE_PROPERTY', 'DAMAGE_CROPS']].dtypes)

```

```

DAMAGE_PROPERTY    float64
DAMAGE_CROPS       float64
dtype: object

```

In [65]:

```

damage_data = filtered_df.groupby('EVENT_TYPE')[['INJURIES_DIRECT', 'INJURIES_INDIRECT', '

```

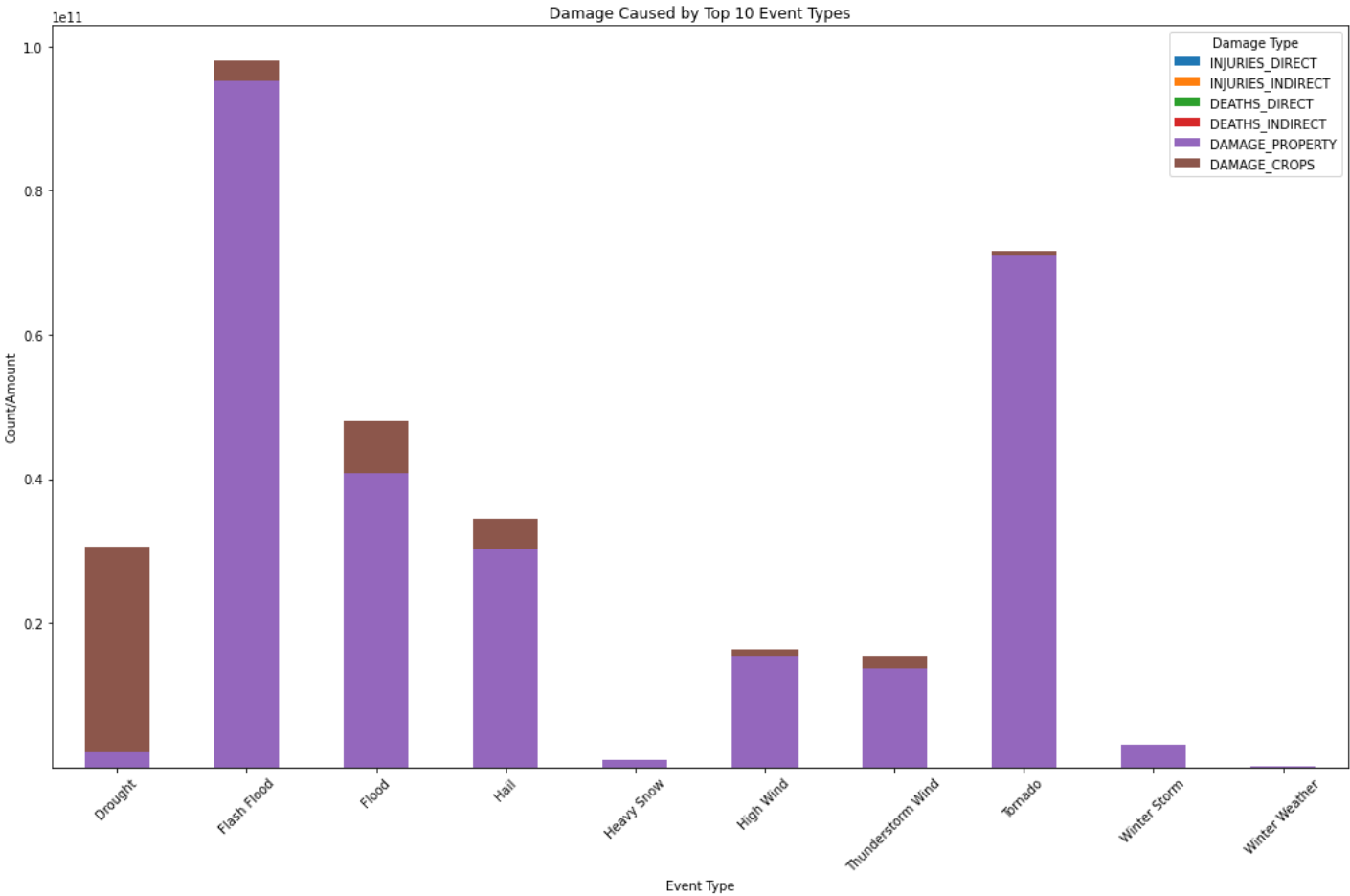
In [66]:

```

damage_data.plot(kind='bar', stacked=True, figsize=(15, 10))
plt.title('Damage Caused by Top 10 Event Types')
plt.xlabel('Event Type')
plt.ylabel('Count/Amount')
plt.xticks(rotation=45)
plt.legend(title='Damage Type')

```

```
plt.tight_layout()
plt.show()
```



```
In [44]: damage_stats = filtered_df.groupby('EVENT_TYPE')[['INJURIES_DIRECT', 'INJURIES_INDIRECT',
              (damage_stats)]
```

Out [44]:

EVENT_TYPE	INJURIES_DIRECT											
	count	mean	std	min	25%	50%	75%	max	count	mean	std	
Drought	69851.0	0.005068	1.324372	0.0	0.0	0.0	0.0	350.0	69851.0	0.000057	0.011965	
Flash Flood	97078.0	0.066771	4.789093	0.0	0.0	0.0	0.0	800.0	97078.0	0.000649	0.033502	
Flood	63689.0	0.037573	2.568986	0.0	0.0	0.0	0.0	500.0	63689.0	0.000816	0.065586	
Hail	390036.0	0.003902	0.337783	0.0	0.0	0.0	0.0	109.0	390036.0	0.000128	0.029956	
Heavy Snow	70384.0	0.010329	0.509817	0.0	0.0	0.0	0.0	100.0	70384.0	0.008596	0.378210	
High Wind	85790.0	0.018184	0.422331	0.0	0.0	0.0	0.0	65.0	85790.0	0.005152	0.946282	
Thunderstorm Wind	502986.0	0.022529	0.500259	0.0	0.0	0.0	0.0	100.0	502986.0	0.000775	0.152475	
Tornado	75522.0	1.295967	15.618754	0.0	0.0	0.0	0.0	1700.0	75522.0	0.003615	0.373831	
Winter Storm	84574.0	0.017440	0.595520	0.0	0.0	0.0	0.0	86.0	84574.0	0.021945	1.192875	
Winter Weather	74428.0	0.031950	1.528260	0.0	0.0	0.0	0.0	177.0	74428.0	0.052856	0.898866	

```
In [45]: merged_df = pd.merge(damage_data, top_10_event_types, left_index=True, right_index=True)
merged_df.rename(columns={'EVENT_TYPE': 'FREQUENCY'}, inplace=True)
```


In [49]:

```
correlation_matrix = merged_df.corr()  
frequency_correlation = correlation_matrix['FREQUENCY']  
print(frequency_correlation)
```

```
INJURIES_DIRECT      -0.096003  
INJURIES_INDIRECT    -0.218308  
DEATHS_DIRECT        -0.115590  
DEATHS_INDIRECT      -0.223603  
DAMAGE_PROPERTY      -0.077397  
DAMAGE_CROPS         -0.128727  
FREQUENCY            1.000000  
Name: FREQUENCY, dtype: float64
```

In []: