In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [3]:

```
df = pd.read_csv("US_Accidents_March23.csv")
```

In [4]:

```
df
```

Out[4]:

	ID	Source	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	En
0	A-1	Source2	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	-84.058723	NaN	
1	A-2	Source2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	-82.831184	NaN	
2	A-3	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	NaN	
3	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	NaN	
4	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	NaN	
...	...	...	...	...	...	...	...	...	...
7728389	A-7777757	Source1	2	2019-08-23 18:03:25	2019-08-23 18:32:01	34.002480	-117.379360	33.99888	-117.
7728390	A-7777758	Source1	2	2019-08-23 19:11:30	2019-08-23 19:38:23	32.766960	-117.148060	32.76555	-117.
7728391	A-7777759	Source1	2	2019-08-23 19:00:21	2019-08-23 19:28:49	33.775450	-117.847790	33.77740	-117.
7728392	A-7777760	Source1	2	2019-08-23 19:00:21	2019-08-23 19:29:42	33.992460	-118.403020	33.98311	-118.
7728393	A-7777761	Source1	2	2019-08-23 18:52:06	2019-08-23 19:21:31	34.133930	-117.230920	34.13736	-117.

7728394 rows × 46 columns

In [5]:

```
print(df.head())
print(df.info())
```

```
   ID   Source  Severity           Start_Time             End_Time  \
0  A-1  Source2         3  2016-02-08 05:46:00  2016-02-08 11:00:00
1  A-2  Source2         2  2016-02-08 06:07:59  2016-02-08 06:37:59
2  A-3  Source2         2  2016-02-08 06:49:27  2016-02-08 07:19:27
3  A-4  Source2         3  2016-02-08 07:23:34  2016-02-08 07:53:34
4  A-5  Source2         2  2016-02-08 07:39:07  2016-02-08 08:09:07
```

```
   Start_Lat   Start_Lng  End_Lat  End_Lng  Distance(mi)  ... Roundabout  \
0  39.865147 -84.058723      NaN      NaN          0.01  ...      False   
1  39.928059 -82.831184      NaN      NaN          0.01  ...      False   
2  39.063148 -84.032608      NaN      NaN          0.01  ...      False   
3  39.747753 -84.205582      NaN      NaN          0.01  ...      False   
4  39.627781 -84.188354      NaN      NaN          0.01  ...      False   

  Station   Stop Traffic_Calming Traffic_Signal Turning_Loop Sunrise_Sunset  \
0   False  False           False          False        False          Night   
1   False  False           False          False        False          Night   
2   False  False           False           True        False          Night   
3   False  False           False          False        False          Night   
4   False  False           False           True        False            Day   

  Civil_Twilight Nautical_Twilight Astronomical_Twilight  
0          Night             Night                 Night  
1          Night             Night                   Day  
2          Night               Day                   Day  
3            Day               Day                   Day  
4            Day               Day                   Day  

[5 rows x 46 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7728394 entries, 0 to 7728393
Data columns (total 46 columns):
 #   Column                 Dtype  
---  ------                 -----  
 0   ID                     object 
 1   Source                 object 
 2   Severity               int64  
 3   Start_Time             object 
 4   End_Time               object 
 5   Start_Lat              float64
 6   Start_Lng              float64
 7   End_Lat                float64
 8   End_Lng                float64
 9   Distance(mi)           float64
 10  Description            object 
 11  Street                 object 
 12  City                   object 
 13  County                 object 
 14  State                  object 
 15  Zipcode                object 
 16  Country                object 
 17  Timezone               object 
 18  Airport_Code           object 
 19  Weather_Timestamp      object 
 20  Temperature(F)         float64
 21  Wind_Chill(F)          float64
 22  Humidity(%)            float64
 23  Pressure(in)           float64
 24  Visibility(mi)         float64
 25  Wind_Direction         object 
 26  Wind_Speed(mph)        float64
 27  Precipitation(in)      float64
 28  Weather_Condition      object 
 29  Amenity                bool   
 30  Bump                   bool   
 31  Crossing               bool   
 32  Give_Way               bool   
 33  Junction               bool   
 34  No_Exit                bool   
 35  Railway                bool
 36  Roundabout             bool
 37  Station                bool
 38  Stop                   bool
 39  Traffic_Calming        bool
 40  Traffic_Signal         bool
 41  Turning_Loop           bool
 42  Sunrise_Sunset         object
 43  Civil_Twilight         object
 44  Nautical_Twilight      object
 45  Astronomical_Twilight  object
dtypes: bool(13), float64(12), int64(1), object(20)
memory usage: 2.0+ GB
None
```

In [6]:

```
print(df['Start_Time'].head())
```

```
0    2016-02-08 05:46:00
1    2016-02-08 06:07:59
2    2016-02-08 06:49:27
3    2016-02-08 07:23:34
4    2016-02-08 07:39:07
Name: Start_Time, dtype: object
```

In [7]:

```
df['Start_Time'] = pd.to_datetime(df['Start_Time'], errors='coerce')
```

In [8]:

```
print(df[df['Start_Time'].isna()])
```

```
                ID   Source  Severity Start_Time  \
3639775  A-3649658  Source1         2        NaT   
3639776  A-3649659  Source1         4        NaT   
3639777  A-3649660  Source1         3        NaT   
3639778  A-3649661  Source1         3        NaT   
3639779  A-3649662  Source1         3        NaT   
...            ...      ...       ...        ...   
6834080  A-6883265  Source1         2        NaT   
6834081  A-6883266  Source1         2        NaT   
6834082  A-6883267  Source1         2        NaT   
6834083  A-6883268  Source1         2        NaT   
6834084  A-6883269  Source1         2        NaT   

                              End_Time  Start_Lat   Start_Lng    End_Lat  \
3639775  2017-07-23 10:21:01.000000000  34.062650 -118.000680  34.061940   
3639776  2017-07-23 11:18:46.000000000  33.931460 -118.390730  33.931530   
3639777  2017-07-23 11:36:01.000000000  33.617890 -117.711160  33.611590   
3639778  2017-07-23 14:36:01.000000000  33.697590 -117.940060  33.695730   
3639779  2017-07-23 17:02:16.000000000  34.035340 -118.329820  34.034960   
...                                ...        ...         ...        ...   
6834080  2020-11-24 05:07:29.000000000  25.943633  -80.205311  25.942485   
6834081  2020-11-19 19:58:52.000000000  39.974916  -75.341137  40.036633   
6834082  2020-11-03 18:47:44.000000000  29.831816  -95.567133  29.831840   
6834083  2020-11-17 02:44:13.000000000  46.556711  -92.603733  46.562231   
6834084  2020-11-25 04:07:32.000000000  38.566700 -121.504014  38.567670   

            End_Lng  Distance(mi)  ... Roundabout Station   Stop  \
3639775 -118.000700         0.049  ...      False   False  False   
3639776 -118.401020         0.590  ...      False   False  False   
3639777 -117.704200         0.591  ...      False   False  False   
3639778 -117.935590         0.287  ...      False   False  False   
3639779 -118.337590         0.446  ...      False   False  False   
...             ...           ...  ...        ...     ...    ...   
6834080  -80.204455         0.095  ...      False   False  False   
```

```
6834081  -75.355487         4.331  ...      False   False  False
6834082  -95.565826         0.078  ...      False   False  False
6834083  -92.595371         0.551  ...      False   False  False
6834084 -121.506734         0.162  ...      False   False  False

        Traffic_Calming Traffic_Signal Turning_Loop Sunrise_Sunset  \
3639775           False          False        False          Night
3639776           False          False        False          Night
3639777           False          False        False          Night
3639778           False          False        False            Day
3639779           False          False        False            Day
...                 ...            ...          ...            ...
6834080           False          False        False            Day
6834081           False          False        False          Night
6834082           False          False        False            Day
6834083           False          False        False          Night
6834084           False          False        False          Night

        Civil_Twilight Nautical_Twilight Astronomical_Twilight
3639775          Night             Night                   Day
3639776          Night               Day                   Day
3639777            Day               Day                   Day
3639778            Day               Day                   Day
3639779            Day               Day                   Day
...                ...               ...                   ...
6834080            Day               Day                   Day
6834081          Night             Night                   Day
6834082            Day               Day                   Day
6834083          Night             Night                 Night
6834084          Night             Night                 Night

[743166 rows x 46 columns]
```

```
In [9]: df = df.dropna(subset=['Start_Time'])
```

```
In [10]: df['Start_Time'] = pd.to_datetime(df['Start_Time'])
```

```
C:\Users\vivek\AppData\Local\Temp\ipykernel_4296\2394841556.py:1: SettingWithC
opyWarning:
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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Start_Time'] = pd.to_datetime(df['Start_Time'])
```

```
In [11]: df['Hour'] = df['Start_Time'].dt.hour
         df['Weekday'] = df['Start_Time'].dt.day_name()
```

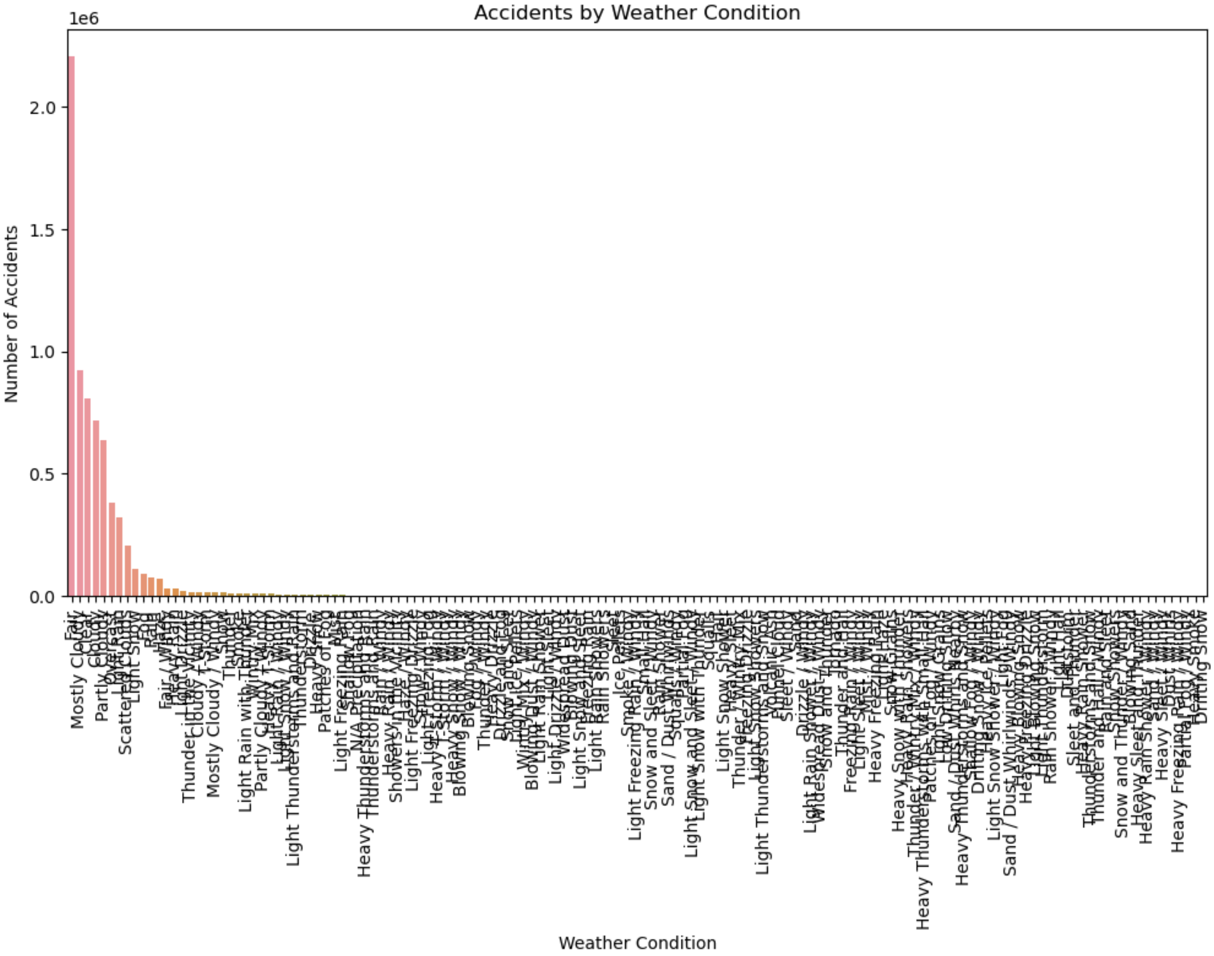
```
C:\Users\vivek\AppData\Local\Temp\ipykernel_4296\3270295720.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Hour'] = df['Start_Time'].dt.hour
C:\Users\vivek\AppData\Local\Temp\ipykernel_4296\3270295720.py:2: SettingWithC
opyWarning:
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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Weekday'] = df['Start_Time'].dt.day_name()
```

In [12]:
```
df['Weekday'] = df['Start_Time'].dt.day_name()
```

```
C:\Users\vivek\AppData\Local\Temp\ipykernel_4296\2272440486.py:1: SettingWithC
opyWarning:
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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Weekday'] = df['Start_Time'].dt.day_name()
```

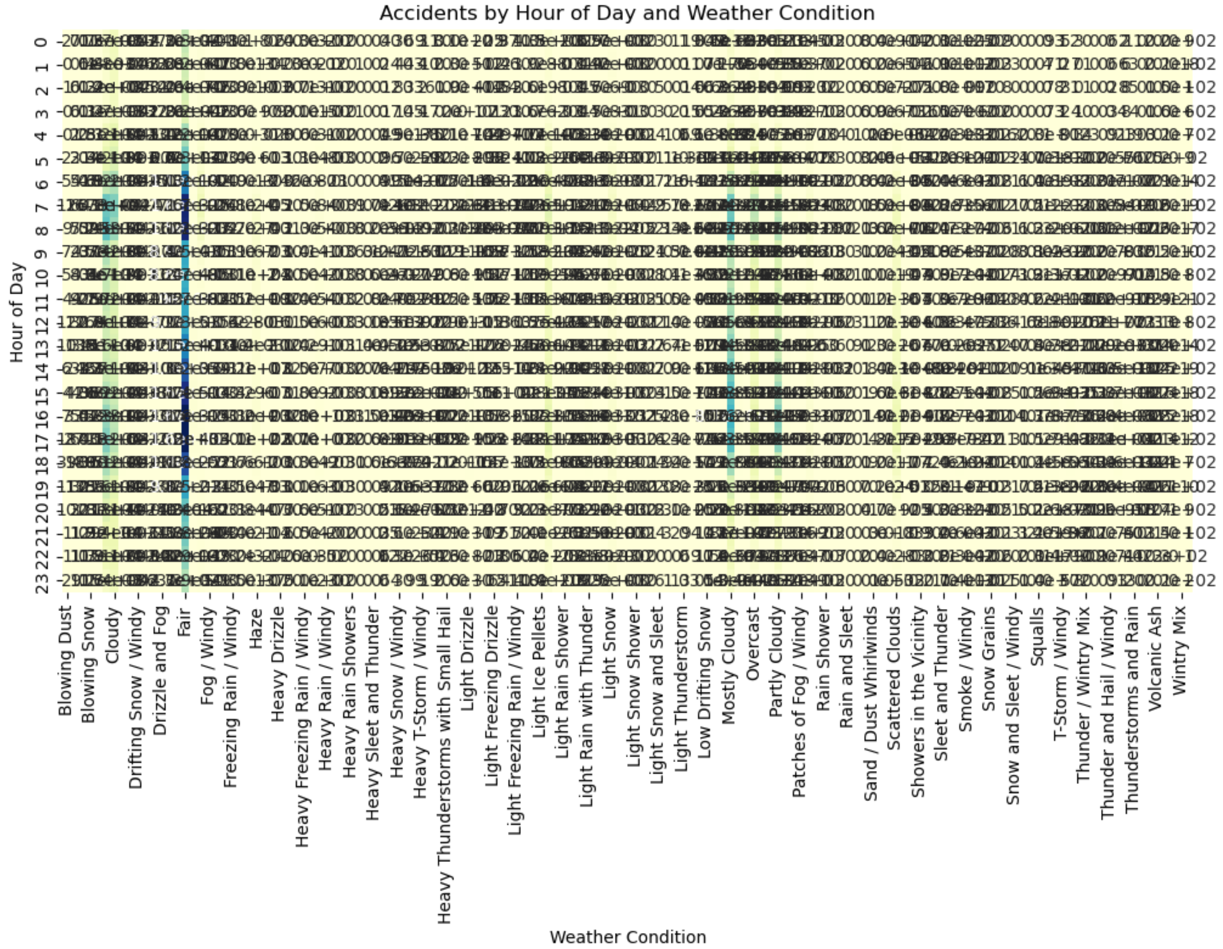
In [13]:
```
weather_accidents = df['Weather_Condition'].value_counts().sort_values(ascendi
plt.figure(figsize=(12,6))
sns.barplot(x=weather_accidents.index, y=weather_accidents.values)
plt.xticks(rotation=90)
plt.title('Accidents by Weather Condition')
plt.xlabel('Weather Condition')
plt.ylabel('Number of Accidents')
plt.show()
```



In [14]:
```
print(df.columns)
```

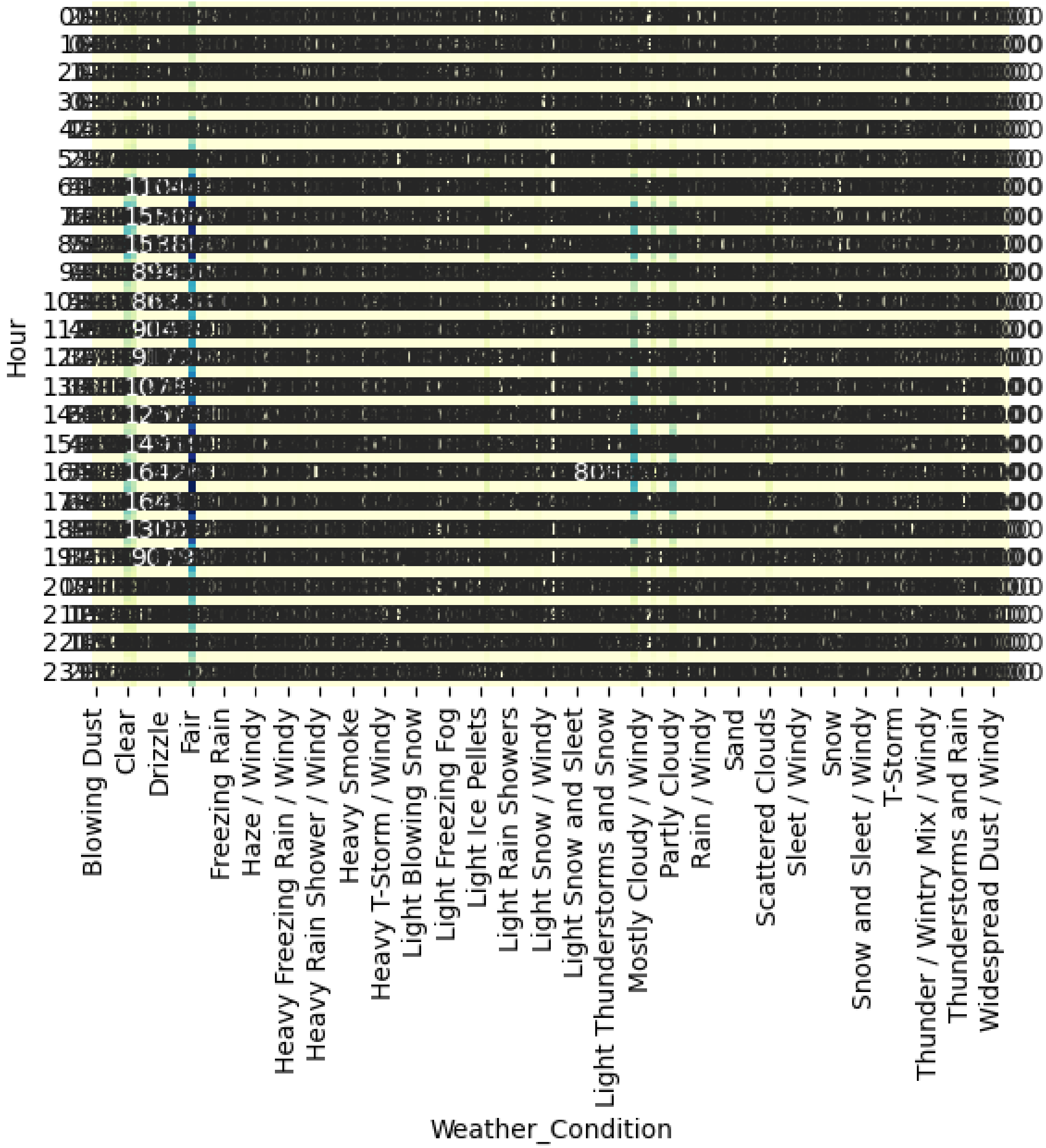
```
Index(['ID', 'Source', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat',
       'Start_Lng', 'End_Lat', 'End_Lng', 'Distance(mi)', 'Description',
       'Street', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone
       'Airport_Code', 'Weather_Timestamp', 'Temperature(F)', 'Wind_Chill(F
       'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction',
       'Wind_Speed(mph)', 'Precipitation(in)', 'Weather_Condition', 'Amenity',
       'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
       'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal',
       'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twiligh
t',
       'Astronomical_Twilight', 'Hour', 'Weekday'],
      dtype='object')
```

In [15]:
```
hour_weather_accidents = df.groupby(['Hour', 'Weather_Condition']).size().unst

plt.figure(figsize=(12, 6))
sns.heatmap(hour_weather_accidents, cmap="YlGnBu", annot=True, cbar=False)
plt.title('Accidents by Hour of Day and Weather Condition')
plt.xlabel('Weather Condition')
plt.ylabel('Hour of Day')
plt.show()
```



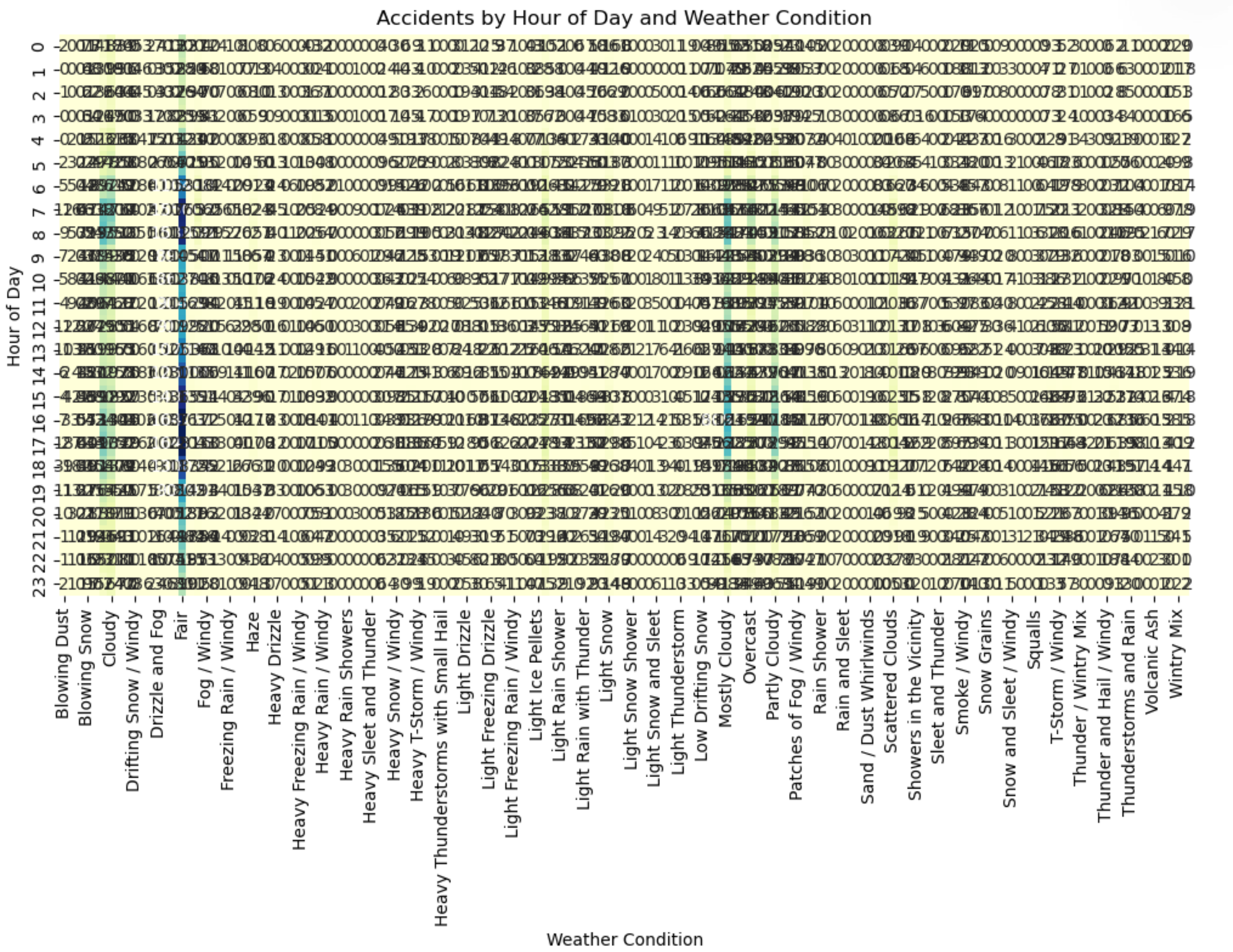
In [16]:
```
sns.heatmap(hour_weather_accidents, cmap="YlGnBu", annot=True, fmt=".2f", cbar
```

Out[16]:
```
<Axes: xlabel='Weather_Condition', ylabel='Hour'>
```



In [17]:

```
hour_weather_accidents = df.groupby(['Hour', 'Weather_Condition']).size().unst

# Convert the grouped data to integers
hour_weather_accidents = hour_weather_accidents.astype(int)

# Plot the heatmap
plt.figure(figsize=(12, 6))
sns.heatmap(hour_weather_accidents, cmap="YlGnBu", annot=True, fmt="d", cbar=F
plt.title('Accidents by Hour of Day and Weather Condition')
plt.xlabel('Weather Condition')
plt.ylabel('Hour of Day')
plt.show()
```



In [18]:

```
condition_columns = [col for col in df.columns if 'Condition' in col]
print(condition_columns)
```

```
['Weather_Condition']
```

In [19]:

```
print(df.columns)
```

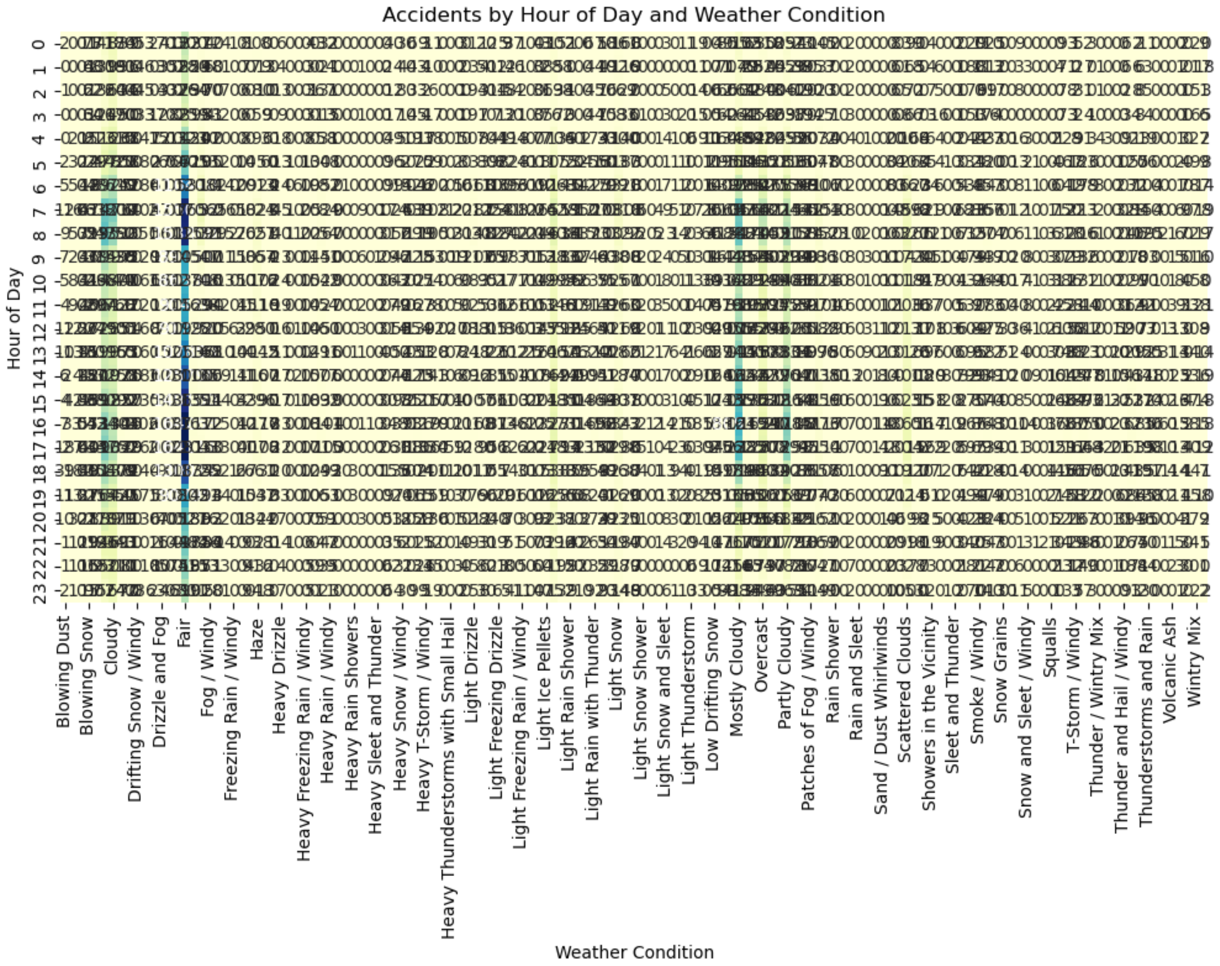
```
Index(['ID', 'Source', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat',
       'Start_Lng', 'End_Lat', 'End_Lng', 'Distance(mi)', 'Description',
       'Street', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone',
       'Airport_Code', 'Weather_Timestamp', 'Temperature(F)', 'Wind_Chill(F)',
       'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction',
       'Wind_Speed(mph)', 'Precipitation(in)', 'Weather_Condition', 'Amenity',
       'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
       'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal',
       'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twiligh
t',
       'Astronomical_Twilight', 'Hour', 'Weekday'],
      dtype='object')
```

In [20]:

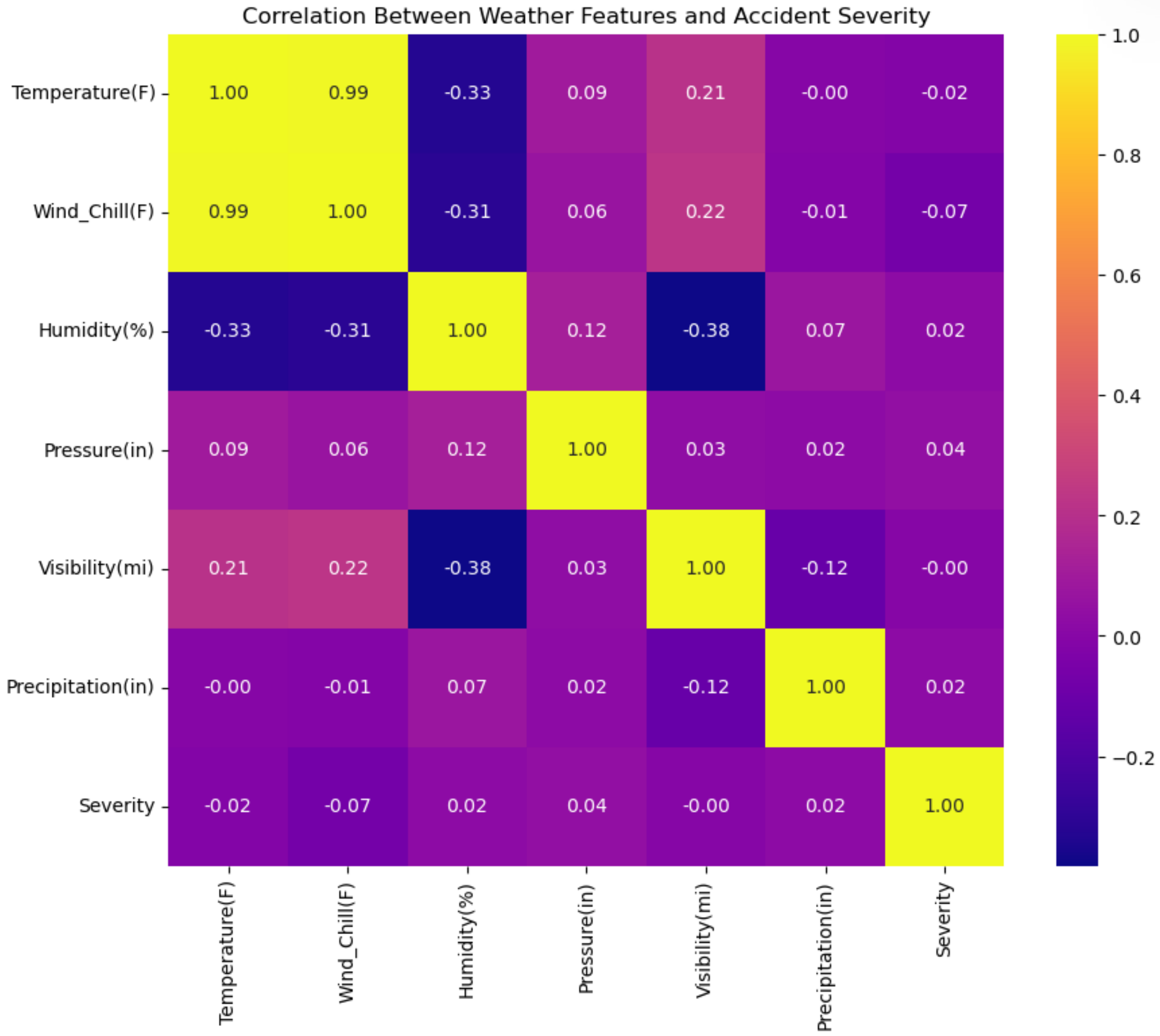
```
print(df.columns)
```

```
Index(['ID', 'Source', 'Severity', 'Start_Time', 'End_Time', 'Start_Lat',
       'Start_Lng', 'End_Lat', 'End_Lng', 'Distance(mi)', 'Description',
       'Street', 'City', 'County', 'State', 'Zipcode', 'Country', 'Timezone',
       'Airport_Code', 'Weather_Timestamp', 'Temperature(F)', 'Wind_Chill(F)',
       'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction',
       'Wind_Speed(mph)', 'Precipitation(in)', 'Weather_Condition', 'Amenity',
       'Bump', 'Crossing', 'Give_Way', 'Junction', 'No_Exit', 'Railway',
       'Roundabout', 'Station', 'Stop', 'Traffic_Calming', 'Traffic_Signal'
       'Turning_Loop', 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilig
t',
       'Astronomical_Twilight', 'Hour', 'Weekday'],
      dtype='object')
```

In [21]:
```
hour_weather_accidents = hour_weather_accidents.astype(int)

plt.figure(figsize=(12, 6))
sns.heatmap(hour_weather_accidents, cmap="YlGnBu", annot=True, fmt="d", cbar=F
plt.title('Accidents by Hour of Day and Weather Condition')
plt.xlabel('Weather Condition')
plt.ylabel('Hour of Day')
plt.show()
```



In [22]:
```
weather_features = ['Temperature(F)', 'Wind_Chill(F)', 'Humidity(%)', 'Pressur
correlation = df[weather_features + ['Severity']].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation, annot=True, cmap='plasma', fmt='.2f')
plt.title('Correlation Between Weather Features and Accident Severity')
plt.show()
```



```
In [25]: fig = px.scatter_mapbox(df, lat="Start_Lat", lon="Start_Lng", color="Weather_C
                                 size="Severity", hover_name="ID", color_continuous_sca
                                 size_max=15, zoom=3, mapbox_style="carto-positron")
         fig.show()
```

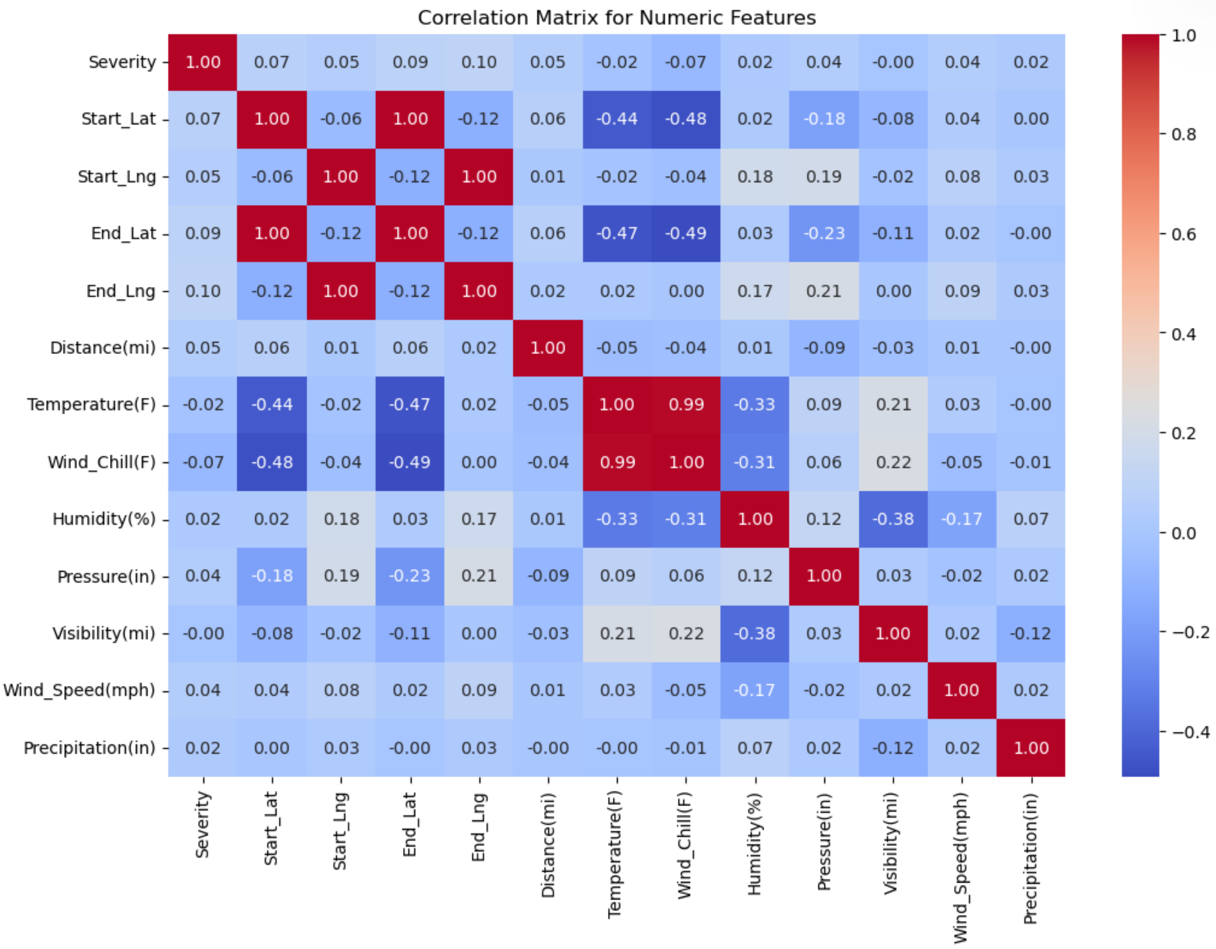
```
In [24]: print(df.dtypes)
```

```
ID                                object
Source                            object
Severity                           int64
Start_Time                datetime64[ns]
End_Time                          object
Start_Lat                        float64
Start_Lng                        float64
End_Lat                          float64
End_Lng                          float64
Distance(mi)                     float64
Description                       object
Street                            object
City                              object
County                            object
State                             object
Zipcode                           object
Country                           object
Timezone                          object
Airport_Code                      object
Weather_Timestamp                 object
```

```
Temperature(F)                float64
Wind_Chill(F)                 float64
Humidity(%)                   float64
Pressure(in)                  float64
Visibility(mi)                float64
Wind_Direction                 object
Wind_Speed(mph)               float64
Precipitation(in)             float64
Weather_Condition              object
Amenity                          bool
Bump                             bool
Crossing                         bool
Give_Way                         bool
Junction                         bool
No_Exit                          bool
Railway                          bool
Roundabout                       bool
Station                          bool
Stop                             bool
Traffic_Calming                  bool
Traffic_Signal                   bool
Turning_Loop                     bool
Sunrise_Sunset                 object
Civil_Twilight                 object
Nautical_Twilight              object
Astronomical_Twilight          object
Hour                            int32
Weekday                        object
dtype: object
```

In [26]:

```
numeric_df = df.select_dtypes(include=['float64', 'int64'])

correlation_matrix = numeric_df.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', cbar=T
plt.title('Correlation Matrix for Numeric Features')
plt.show()
```

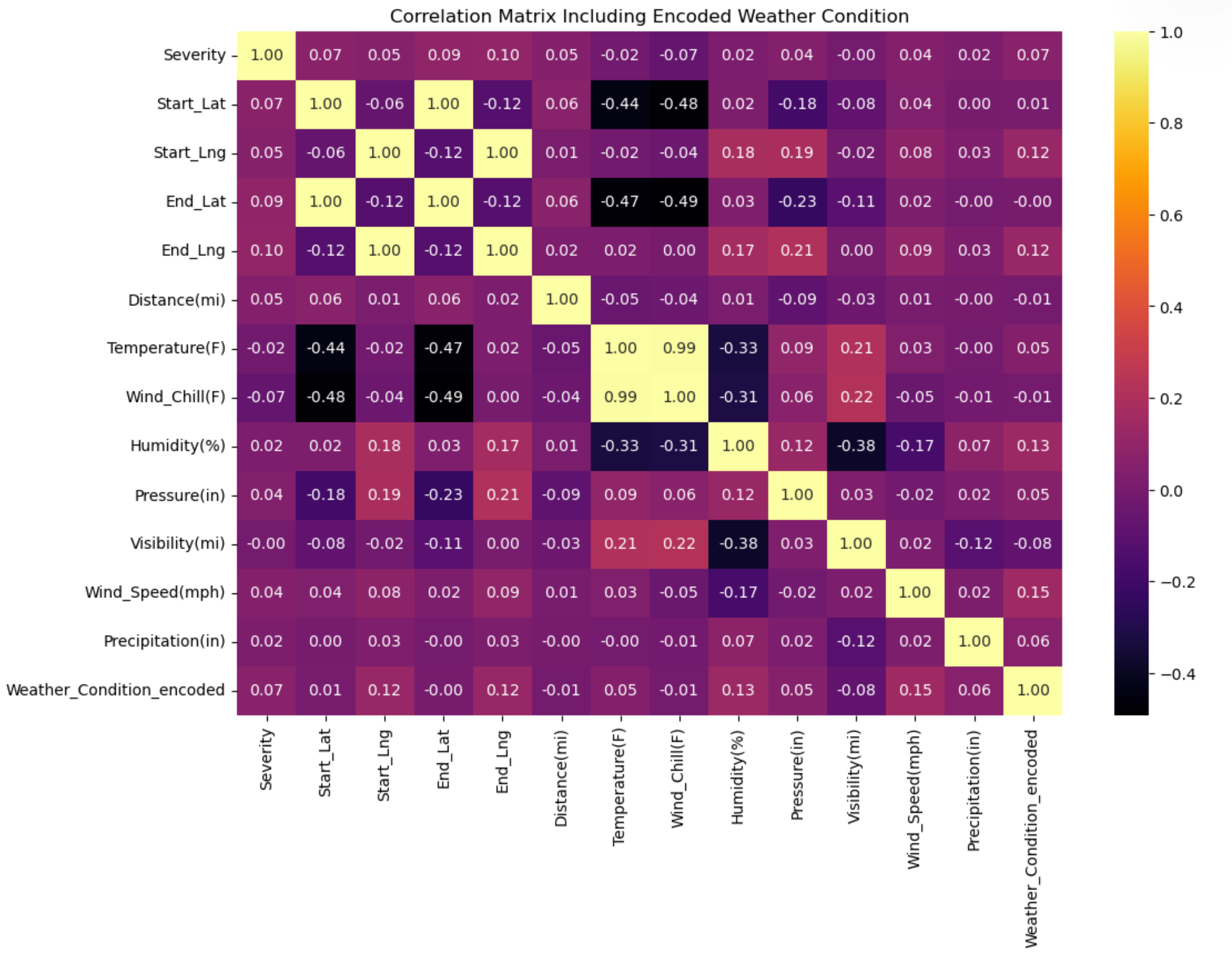


	Severity	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	Temperature(F)	Wind_Chill(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Precipitation(in)
Severity	1.00	0.07	0.05	0.09	0.10	0.05	-0.02	-0.07	0.02	0.04	-0.00	0.04	0.02
Start_Lat	0.07	1.00	-0.06	1.00	-0.12	0.06	-0.44	-0.48	0.02	-0.18	-0.08	0.04	0.00
Start_Lng	0.05	-0.06	1.00	-0.12	1.00	0.01	-0.02	-0.04	0.18	0.19	-0.02	0.08	0.03
End_Lat	0.09	1.00	-0.12	1.00	-0.12	0.06	-0.47	-0.49	0.03	-0.23	-0.11	0.02	-0.00
End_Lng	0.10	-0.12	1.00	-0.12	1.00	0.02	0.02	0.00	0.17	0.21	0.00	0.09	0.03
Distance(mi)	0.05	0.06	0.01	0.06	0.02	1.00	-0.05	-0.04	0.01	-0.09	-0.03	0.01	-0.00
Temperature(F)	-0.02	-0.44	-0.02	-0.47	0.02	-0.05	1.00	0.99	-0.33	0.09	0.21	0.03	-0.00
Wind_Chill(F)	-0.07	-0.48	-0.04	-0.49	0.00	-0.04	0.99	1.00	-0.31	0.06	0.22	-0.05	-0.01
Humidity(%)	0.02	0.02	0.18	0.03	0.17	0.01	-0.33	-0.31	1.00	0.12	-0.38	-0.17	0.07
Pressure(in)	0.04	-0.18	0.19	-0.23	0.21	-0.09	0.09	0.06	0.12	1.00	0.03	-0.02	0.02
Visibility(mi)	-0.00	-0.08	-0.02	-0.11	0.00	-0.03	0.21	0.22	-0.38	0.03	1.00	0.02	-0.12
Wind_Speed(mph)	0.04	0.04	0.08	0.02	0.09	0.01	0.03	-0.05	-0.17	-0.02	0.02	1.00	0.02
Precipitation(in)	0.02	0.00	0.03	-0.00	0.03	-0.00	-0.00	-0.01	0.07	0.02	-0.12	0.02	1.00

In [27]:

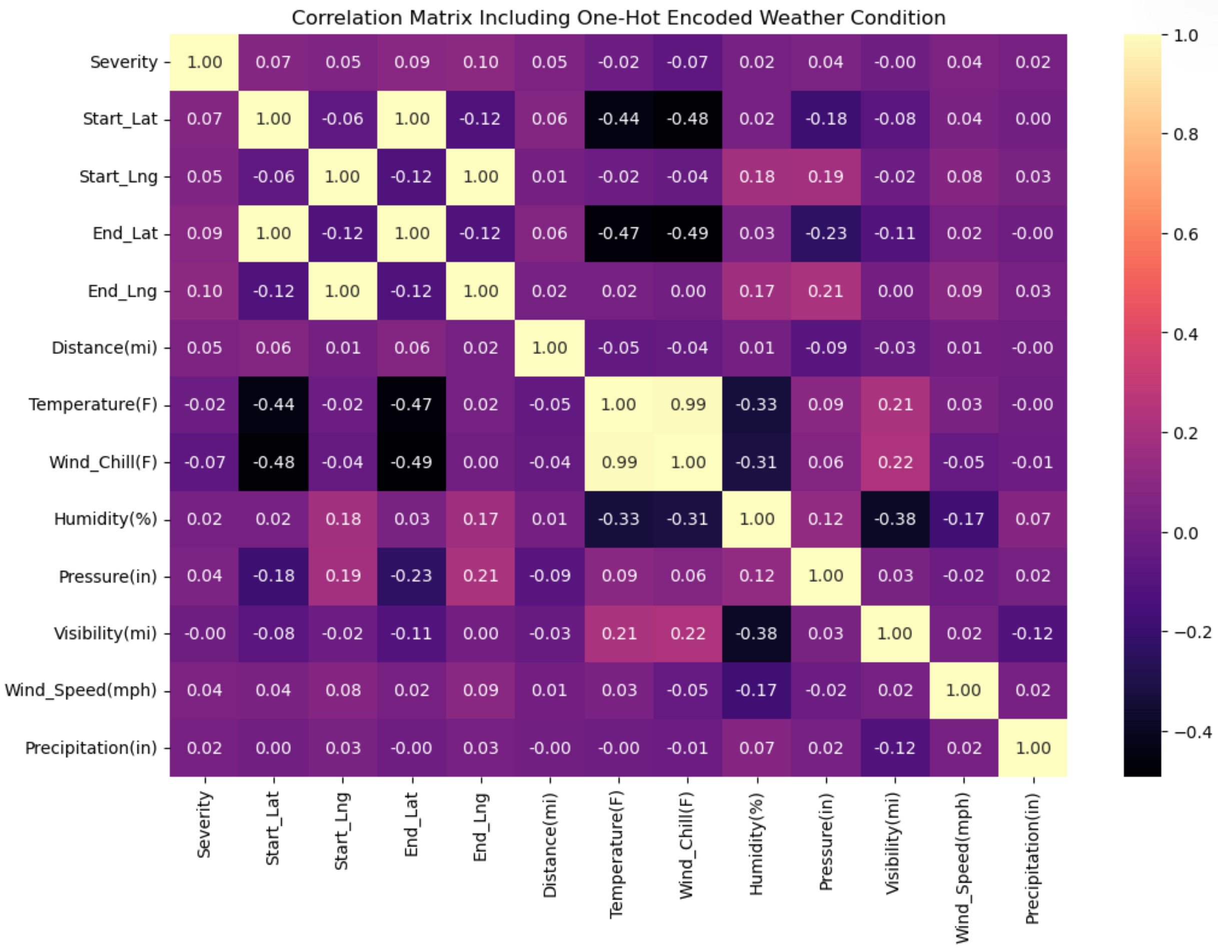
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
df['Weather_Condition_encoded'] = label_encoder.fit_transform(df['Weather_Cond

numeric_df = df.select_dtypes(include=['float64', 'int64']).copy()
numeric_df['Weather_Condition_encoded'] = df['Weather_Condition_encoded']

correlation_matrix = numeric_df.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='inferno', fmt='.2f', cbar=Tr
plt.title('Correlation Matrix Including Encoded Weather Condition')
plt.show()
```

```
C:\Users\vivek\AppData\Local\Temp\ipykernel_4296\3411963840.py:3: SettingWithC
opyWarning:


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/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
```



In [28]:

```
df = pd.get_dummies(df, columns=['Weather_Condition'], drop_first=True)

numeric_df = df.select_dtypes(include=['float64', 'int64'])

correlation_matrix = numeric_df.corr()

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='magma', fmt='.2f', cbar=True
plt.title('Correlation Matrix Including One-Hot Encoded Weather Condition')
plt.show()
```



	Severity	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	Temperature(F)	Wind_Chill(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Precipitation(in)
Severity	1.00	0.07	0.05	0.09	0.10	0.05	-0.02	-0.07	0.02	0.04	-0.00	0.04	0.02
Start_Lat	0.07	1.00	-0.06	1.00	-0.12	0.06	-0.44	-0.48	0.02	-0.18	-0.08	0.04	0.00
Start_Lng	0.05	-0.06	1.00	-0.12	1.00	0.01	-0.02	-0.04	0.18	0.19	-0.02	0.08	0.03
End_Lat	0.09	1.00	-0.12	1.00	-0.12	0.06	-0.47	-0.49	0.03	-0.23	-0.11	0.02	-0.00
End_Lng	0.10	-0.12	1.00	-0.12	1.00	0.02	0.02	0.00	0.17	0.21	0.00	0.09	0.03
Distance(mi)	0.05	0.06	0.01	0.06	0.02	1.00	-0.05	-0.04	0.01	-0.09	-0.03	0.01	-0.00
Temperature(F)	-0.02	-0.44	-0.02	-0.47	0.02	-0.05	1.00	0.99	-0.33	0.09	0.21	0.03	-0.00
Wind_Chill(F)	-0.07	-0.48	-0.04	-0.49	0.00	-0.04	0.99	1.00	-0.31	0.06	0.22	-0.05	-0.01
Humidity(%)	0.02	0.02	0.18	0.03	0.17	0.01	-0.33	-0.31	1.00	0.12	-0.38	-0.17	0.07
Pressure(in)	0.04	-0.18	0.19	-0.23	0.21	-0.09	0.09	0.06	0.12	1.00	0.03	-0.02	0.02
Visibility(mi)	-0.00	-0.08	-0.02	-0.11	0.00	-0.03	0.21	0.22	-0.38	0.03	1.00	0.02	-0.12
Wind_Speed(mph)	0.04	0.04	0.08	0.02	0.09	0.01	0.03	-0.05	-0.17	-0.02	0.02	1.00	0.02
Precipitation(in)	0.02	0.00	0.03	-0.00	0.03	-0.00	-0.00	-0.01	0.07	0.02	-0.12	0.02	1.00

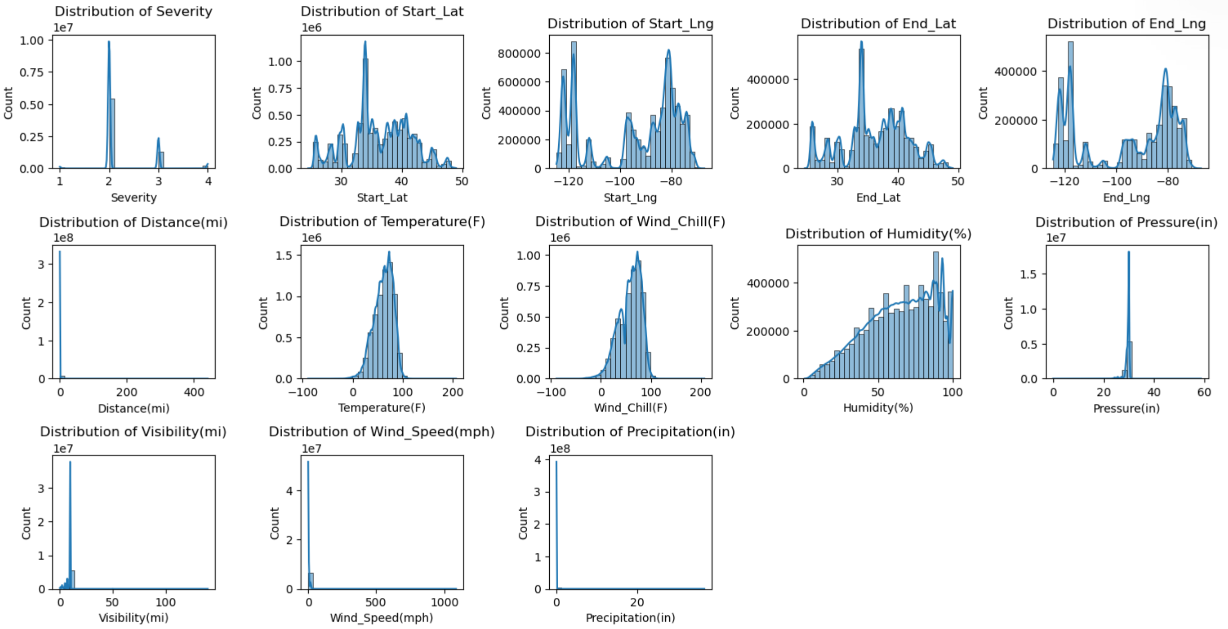
In [29]:
```
numeric_summary = df.describe()
print(numeric_summary)
```

```
           Severity                     Start_Time     Start_Lat  \
count  6.985228e+06                        6985228  6.985228e+06
mean   2.229248e+00  2020-03-21 21:49:59.056442112  3.622647e+01
min    1.000000e+00            2016-01-14 20:18:33  2.455480e+01
25%    2.000000e+00  2018-09-17 02:20:31.750000128  3.342209e+01
50%    2.000000e+00     2020-06-23 13:37:23.500000  3.581521e+01
75%    2.000000e+00            2021-10-28 15:44:07  4.009647e+01
max    4.000000e+00            2023-03-31 23:30:00  4.900220e+01
std    4.988711e-01                            NaN  5.055433e+00

          Start_Lng       End_Lat       End_Lng  Distance(mi)  Temperature(F)
\
count  6.985228e+06  3.582466e+06  3.582466e+06  6.985228e+06    6.839140e+06
mean  -9.476577e+01  3.632363e+01 -9.606114e+01  5.168789e-01    6.168508e+01
min   -1.246238e+02  2.456601e+01 -1.245457e+02  0.000000e+00   -8.900000e+01
25%   -1.172176e+02  3.350703e+01 -1.178403e+02  0.000000e+00    4.900000e+01
50%   -8.789415e+01  3.627421e+01 -8.963065e+01  1.000000e-02    6.400000e+01
75%   -8.038722e+01  4.023982e+01 -8.029351e+01  3.880000e-01    7.600000e+01
max   -6.711317e+01  4.907500e+01 -6.710924e+01  4.417500e+02    2.070000e+02
std    1.735545e+01  5.273354e+00  1.816974e+01  1.746987e+00    1.892246e+01

       Wind_Chill(F)   Humidity(%)  Pressure(in)  Visibility(mi)  \
count   5.011143e+06  6.829884e+06  6.859985e+06    6.827127e+06
mean    5.799474e+01  6.501947e+01  2.955900e+01    9.093596e+00
min    -8.900000e+01  1.000000e+00  0.000000e+00    0.000000e+00
25%     4.200000e+01  4.800000e+01  2.939000e+01    1.000000e+01
50%     6.200000e+01  6.700000e+01  2.987000e+01    1.000000e+01
75%     7.500000e+01  8.400000e+01  3.004000e+01    1.000000e+01
```

```
max     2.070000e+02  1.000000e+02  5.863000e+01    1.400000e+02
std     2.239818e+01  2.279870e+01  9.864551e-01    2.706562e+00

       Wind_Speed(mph)  Precipitation(in)          Hour  \
count     6.435855e+06       4.812697e+06  6.985228e+06
mean      7.696474e+00       8.824113e-03  1.226779e+01
min       0.000000e+00       0.000000e+00  0.000000e+00
25%       4.600000e+00       0.000000e+00  8.000000e+00
50%       7.000000e+00       0.000000e+00  1.300000e+01
75%       1.040000e+01       0.000000e+00  1.700000e+01
max       1.087000e+03       3.647000e+01  2.300000e+01
std       5.411353e+00       1.170015e-01  5.455518e+00

       Weather_Condition_encoded
count               6.985228e+06
mean                4.456084e+01
min                 0.000000e+00
25%                 1.500000e+01
50%                 1.500000e+01
75%                 8.400000e+01
max                 1.430000e+02
std                 3.991002e+01
```

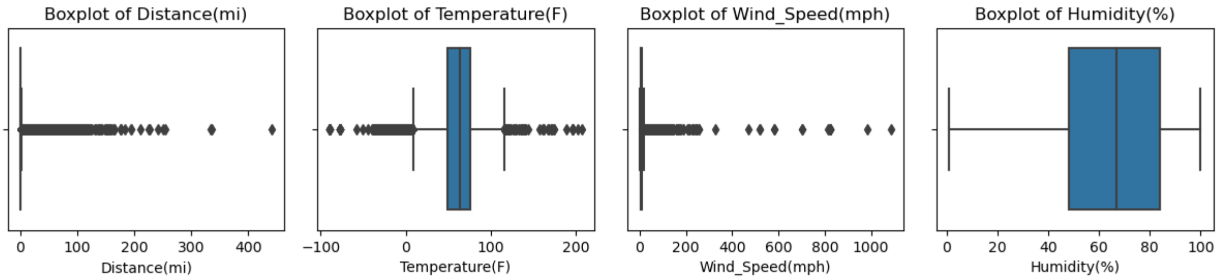
In [30]:

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Select numeric columns from the DataFrame
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns

# Set the figure size
plt.figure(figsize=(15, 10))

# Loop through numeric columns and create subplots
for i, column in enumerate(numeric_columns, 1):
    plt.subplot(4, 5, i)  # Adjust rows/columns as needed
    sns.histplot(df[column], kde=True, bins=30)  # Plot histogram with KDE
    plt.title(f'Distribution of {column}')

# Adjust layout after all subplots have been created
plt.tight_layout()

# Display the plots
plt.show()
```



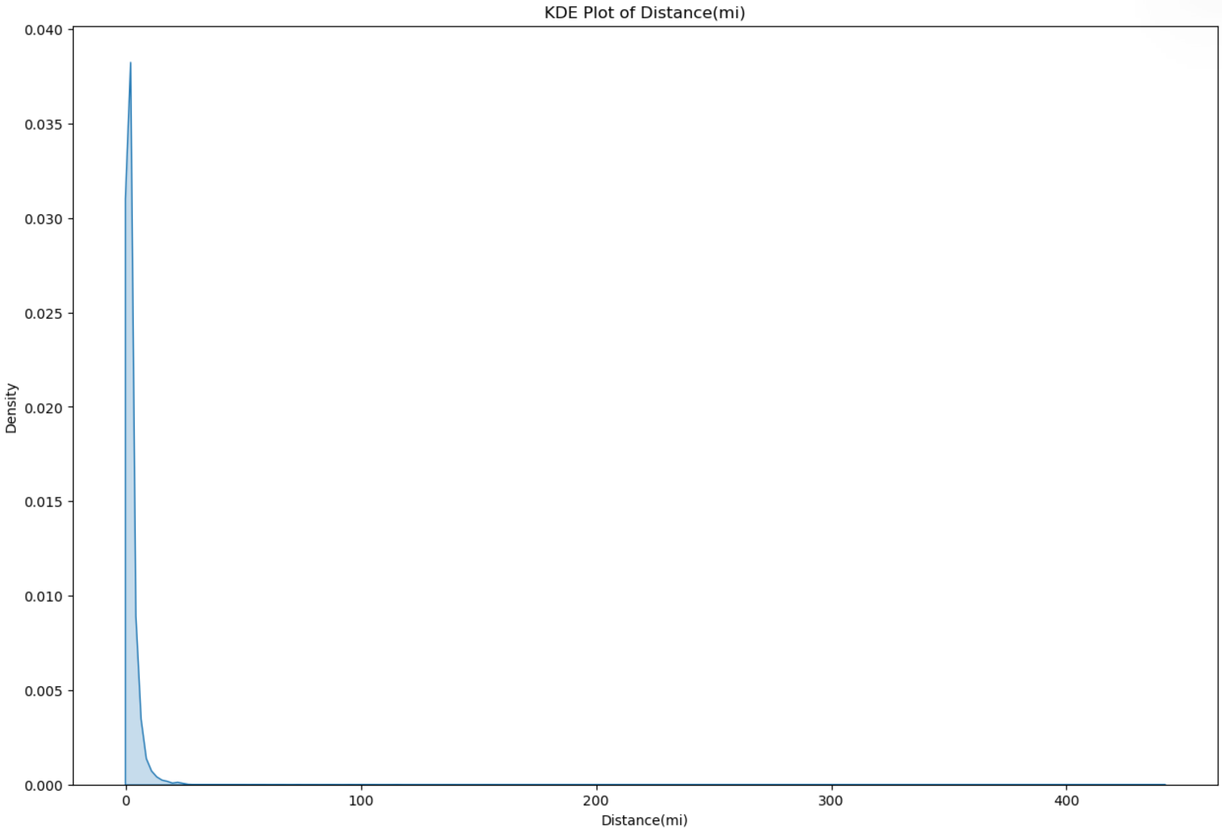
In [35]:

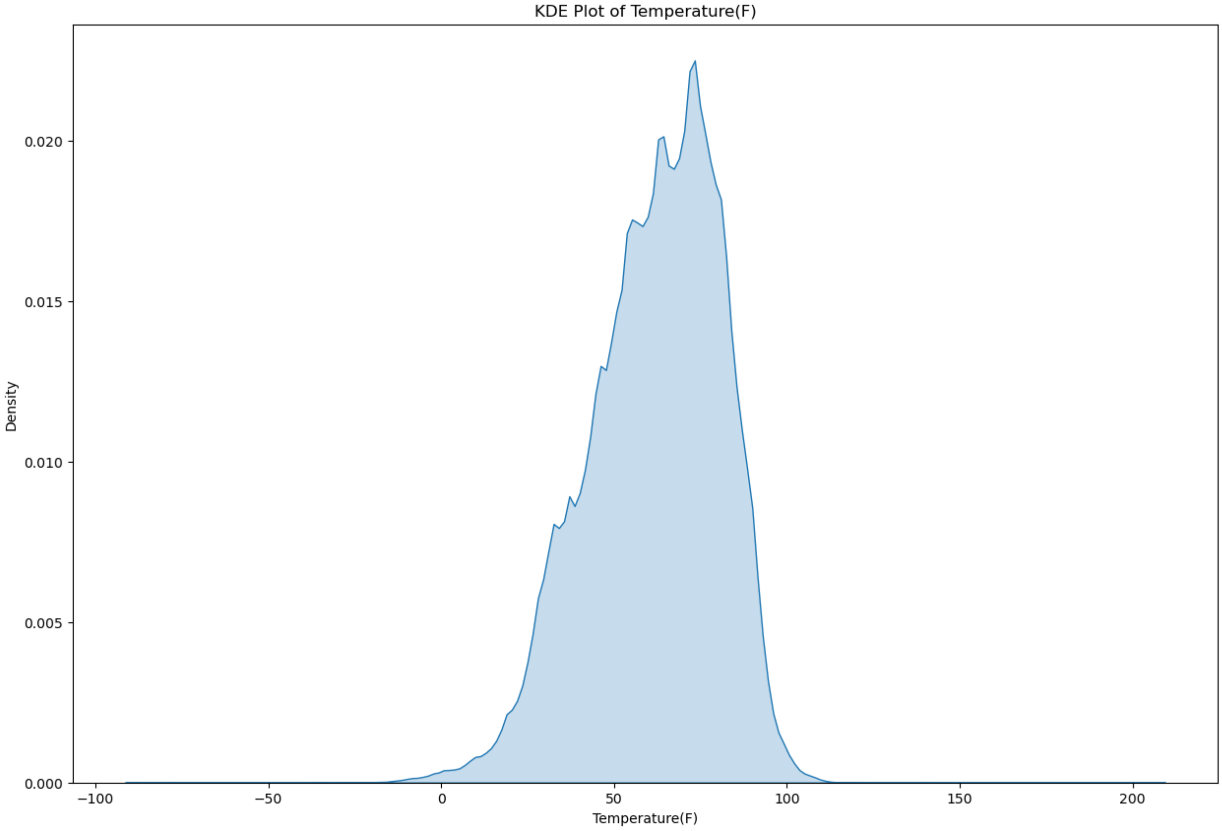
```
plt.figure(figsize=(15, 10))

for i, column in enumerate(numeric_columns, 1):
    plt.subplot(4, 5, i)
    sns.boxplot(x=df[column])
    plt.title(f'Boxplot of {column}')

# Call tight_layout once after the loop
plt.tight_layout()

plt.show()
```

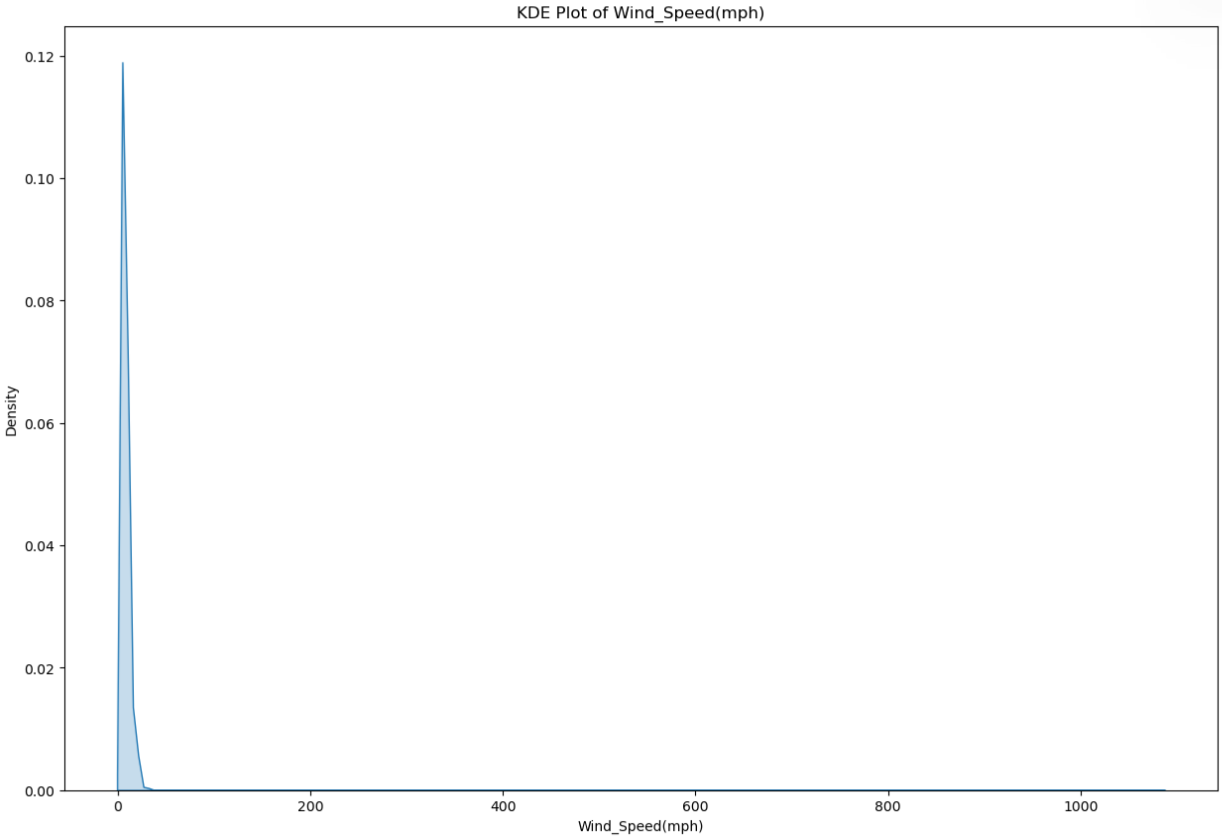


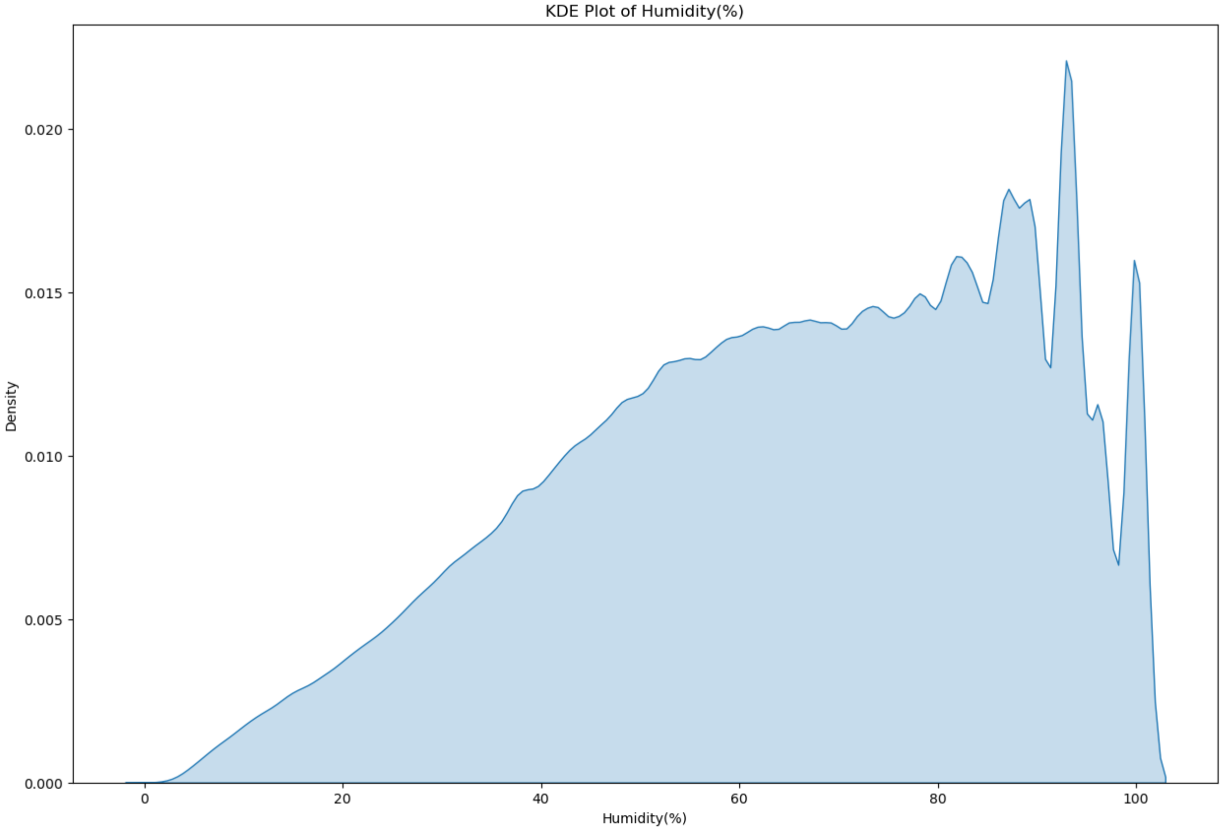
In [32]:

```
numeric_columns = ['Distance(mi)', 'Temperature(F)', 'Wind_Speed(mph)', 'Humid
for column in numeric_columns:
    plt.figure(figsize=(15, 10))
    sns.kdeplot(df[column], fill=True)
    plt.title(f'KDE Plot of {column}')
    plt.show()
```


KDE Plot of Distance(mi)
Density
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
100
200
300
400
Distance(mi)


KDE Plot of Temperature(F)
Density
0.020
0.015
0.010
0.005
0.000
−100
−50
0
50
100
150
200
Temperature(F)


KDE Plot of Wind_Speed(mph)
Density
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
200
400
600
800
1000
Wind_Speed(mph)


KDE Plot of Humidity(%)
Density
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
Humidity(%)

```
In [ ]:
```