Q5:Analyze traffic accident data to identify patterns related to road conditions, weather, and time of day. Visualize accident hotspots and contributing factors.

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
data = pd.read_csv("accident_data.csv")
data
```

Out[4]:

	Date	Time	Latitude	Longitude	Location Description	Weather Conditions	Road Conditions	Visibility	Acci
0	05-01-2022	15:30:00	40.7128	-74.0060	Highway	Foggy	Wet	Poor	Coll
1	10-02-2022	18:45:00	41.8781	-87.6298	Street	Rainy	Wet	Poor	Rol
2	15-03-2022	12:00:00	37.7749	-122.4194	Intersection	Clear	Dry	Clear	Coll
3	20-04-2022	09:20:00	34.0522	-118.2437	Highway	Clear	Dry	Clear	Coll
4	25-05-2022	17:10:00	40.7128	-74.0060	Street	Rainy	Wet	Poor	Coll
...	...	...	...	...	...	...	...	...	
142	10-08-2024	12:45:00	40.7128	-74.0060	Street	Rainy	Wet	Poor	Rol
143	15-09-2024	08:30:00	37.7749	-122.4194	Highway	Clear	Dry	Clear	Coll
144	20-10-2024	16:20:00	34.0522	-118.2437	Street	Clear	Dry	Clear	Coll
145	25-11-2024	14:10:00	40.7128	-74.0060	Intersection	Rainy	Wet	Poor	Coll
146	30-12-2024	10:00:00	41.8781	-87.6298	Highway	Clear	Dry	Clear	Rol

147 rows × 17 columns

In [5]:
```
data.head(10)
```

Out[5]:

	Date	Time	Latitude	Longitude	Location Description	Weather Conditions	Road Conditions	Visibility	Accide Ty
0	05-01-2022	15:30:00	40.7128	-74.0060	Highway	Foggy	Wet	Poor	Collisi
1	10-02-2022	18:45:00	41.8781	-87.6298	Street	Rainy	Wet	Poor	Rollov
2	15-03-2022	12:00:00	37.7749	-122.4194	Intersection	Clear	Dry	Clear	Collisi
3	20-04-2022	09:20:00	34.0522	-118.2437	Highway	Clear	Dry	Clear	Collisi
4	25-05-2022	17:10:00	40.7128	-74.0060	Street	Rainy	Wet	Poor	Collisi
5	30-06-2022	14:30:00	41.8781	-87.6298	Intersection	Clear	Dry	Clear	Rollov
6	05-07-2022	20:15:00	34.0522	-118.2437	Street	Clear	Dry	Clear	Collisi
7	10-08-2022	12:45:00	40.7128	-74.0060	Highway	Rainy	Wet	Poor	Rollov
8	15-09-2022	08:30:00	37.7749	-122.4194	Intersection	Clear	Dry	Clear	Collisi
9	20-10-2022	16:20:00	34.0522	-118.2437	Highway	Clear	Dry	Clear	Collisi

In [6]:
```
data.describe()
```

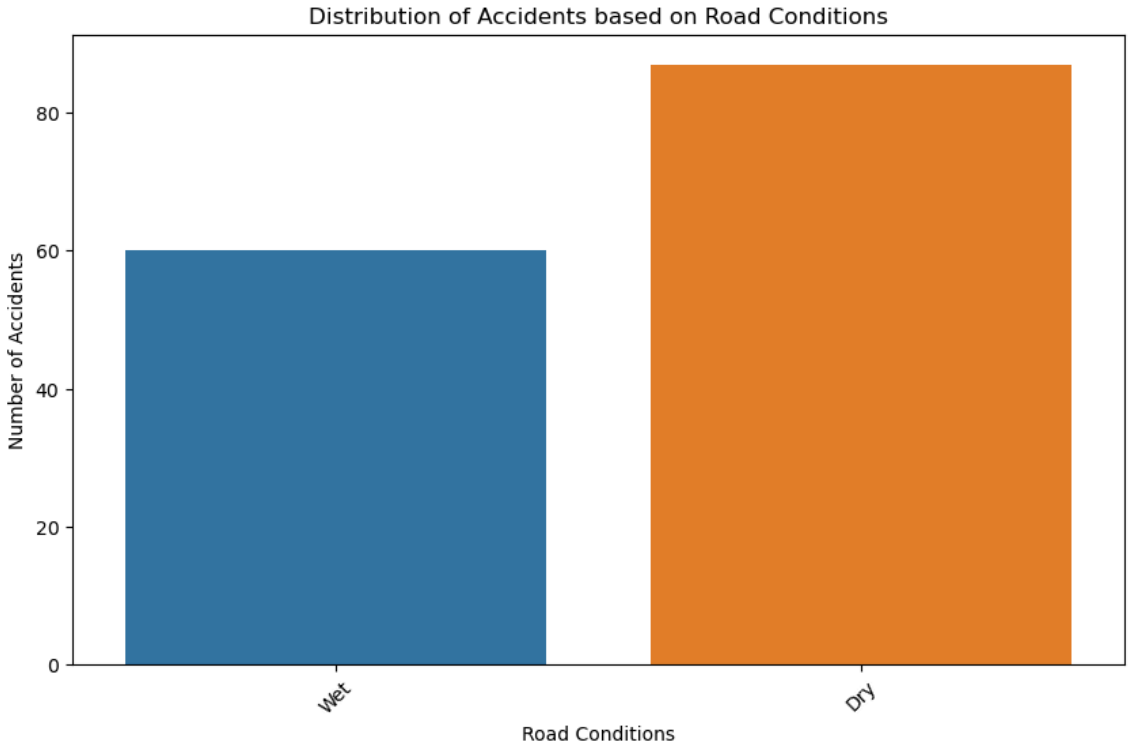
Out[6]:

	Latitude	Longitude	No of Vehichles	Vehicle Speed	Year
count	147.000000	147.000000	147.000000	147.000000	147.000000
mean	38.325525	-98.649265	1.428571	38.843537	2022.979592
std	3.197276	20.962406	0.496564	8.110285	0.823203
min	34.052200	-122.419400	1.000000	25.000000	2022.000000
25%	34.052200	-118.243700	1.000000	30.000000	2022.000000
50%	40.712800	-87.629800	1.000000	35.000000	2023.000000
75%	40.712800	-74.006000	2.000000	45.000000	2024.000000
max	41.878100	-74.006000	2.000000	55.000000	2024.000000

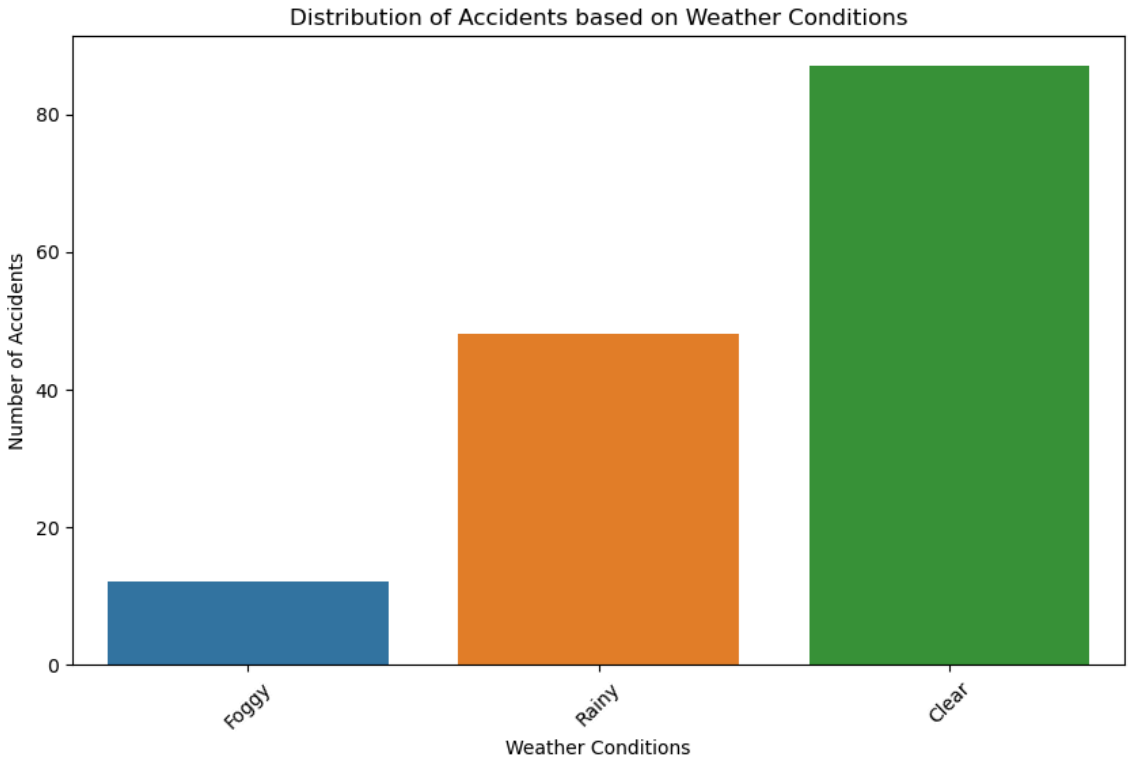
In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Date                    0
Time                    0
Latitude                0
Longitude               0
Location Description    0
Weather Conditions      0
Road Conditions         0
Visibility              0
Accident Type           0
Injury Severity         0
Contributing Factors    0
Vehicle Type            0
No of Vehichles         0
Vehicle Speed           0
Day of Week             0
Month                   0
Year                    0
dtype: int64
```

In [12]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Road Conditions', data=data)
plt.title('Distribution of Accidents based on Road Conditions')
plt.xlabel('Road Conditions')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```



In [13]:
```
plt.figure(figsize=(10, 6))
sns.countplot(x='Weather Conditions', data=data)
plt.title('Distribution of Accidents based on Weather Conditions')
plt.xlabel('Weather Conditions')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```

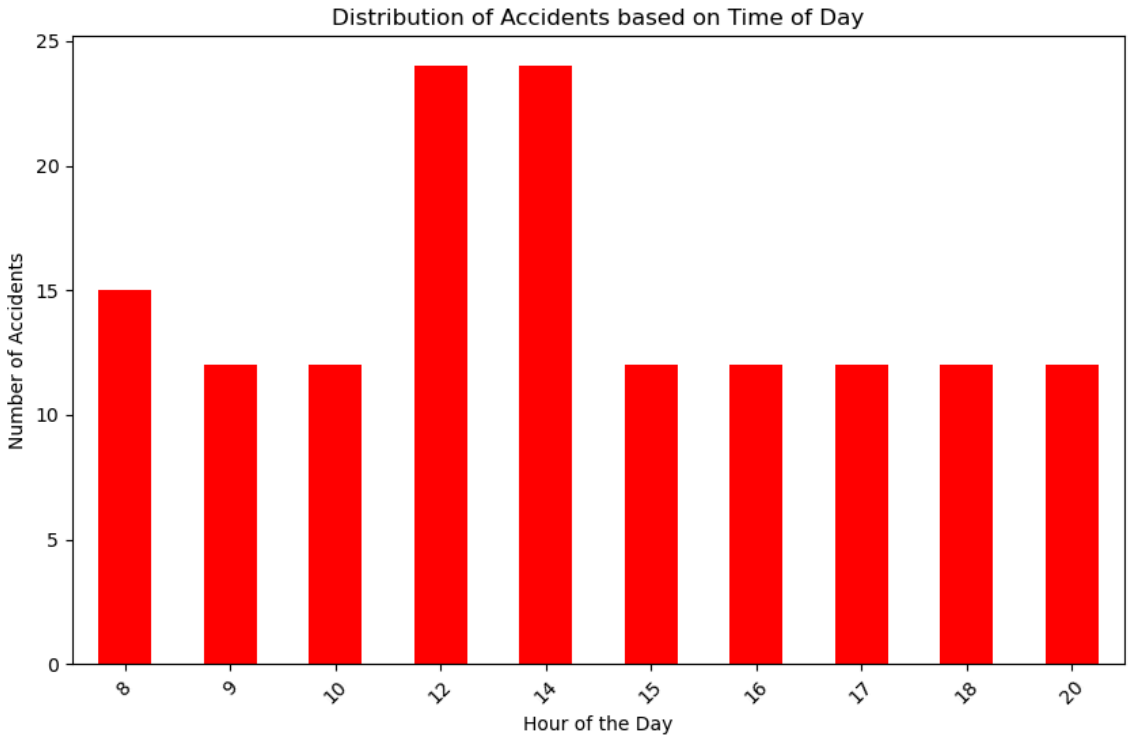


In [26]:
```
data['DateTime']=pd.to_datetime(data['Date']+' '+ data['Time'],format="mixe
data.set_index('DateTime', inplace=True)
```

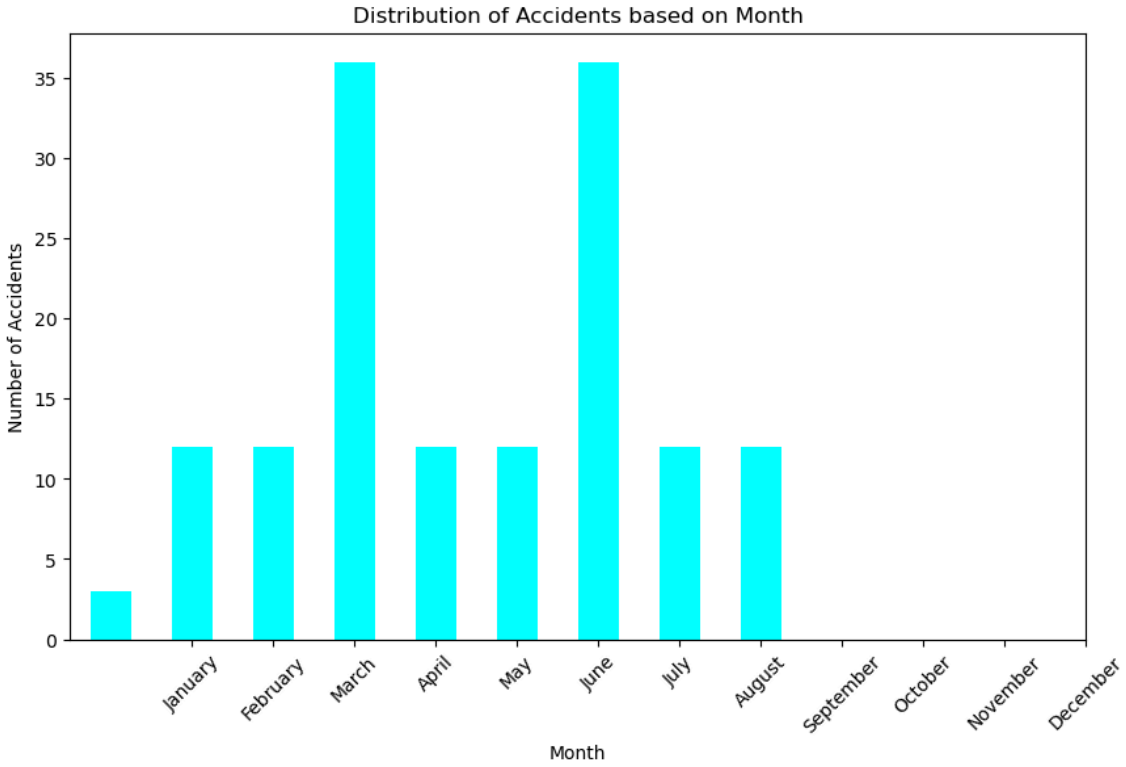
In [27]:
```
data.index
```

Out[27]:
```
DatetimeIndex(['2022-05-01 15:30:00', '2022-10-02 18:45:00',
               '2022-03-15 12:00:00', '2022-04-20 09:20:00',
               '2022-05-25 17:10:00', '2022-06-30 14:30:00',
               '2022-05-07 20:15:00', '2022-10-08 12:45:00',
               '2022-09-15 08:30:00', '2022-10-20 16:20:00',
               ...
               '2024-03-15 12:00:00', '2024-04-20 09:20:00',
               '2024-05-25 17:10:00', '2024-06-30 14:30:00',
               '2024-05-07 20:15:00', '2024-10-08 12:45:00',
               '2024-09-15 08:30:00', '2024-10-20 16:20:00',
               '2024-11-25 14:10:00', '2024-12-30 10:00:00'],
              dtype='datetime64[ns]', name='DateTime', length=147, freq=No
ne)
```

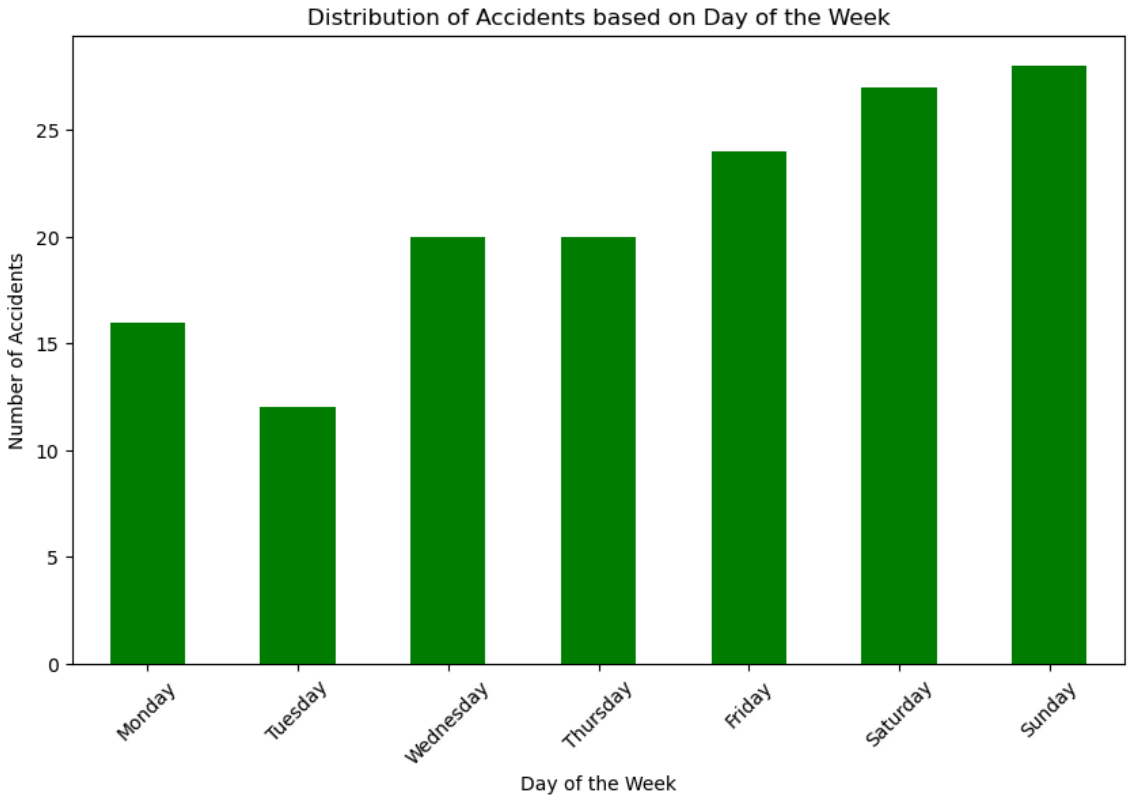
In [28]:
```
plt.figure(figsize=(10, 6))
data.index.hour.value_counts().sort_index().plot(kind='bar',color='red')
plt.title('Distribution of Accidents based on Time of Day')
plt.xlabel('Hour of the Day')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```



In [29]:
```
plt.figure(figsize=(10, 6))
data.index.month.value_counts().sort_index().plot(kind='bar',color="cyan")
plt.title('Distribution of Accidents based on Month')
plt.xlabel('Month')
plt.ylabel('Number of Accidents')
plt.xticks(range(1, 13), ['January', 'February', 'March', 'April', 'May', '
plt.show()
```



In [30]:
```
plt.figure(figsize=(10, 6))
data.index.dayofweek.value_counts().sort_index().plot(kind='bar',color = "g
plt.title('Distribution of Accidents based on Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Accidents')
plt.xticks(range(7), ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday
plt.show()
```



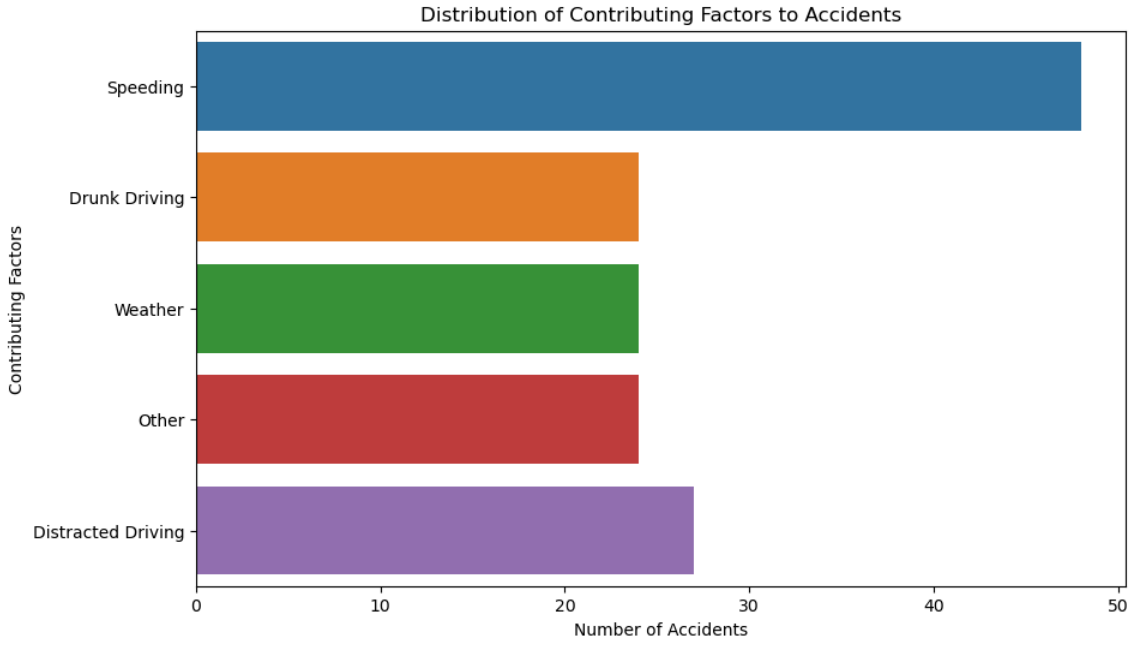
In [32]:
```
!pip install folium
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Collecting folium
  Obtaining dependency information for folium from https://files.pythonhos
ted.org/packages/b9/98/9ba4b9d2d07dd32765ddb4e4c189dcbdd7dca4d5a735e2e4ea7
56f40c36b/folium-0.16.0-py2.py3-none-any.whl.metadata (https://files.pytho
nhosted.org/packages/b9/98/9ba4b9d2d07dd32765ddb4e4c189dcbdd7dca4d5a735e2e
4ea756f40c36b/folium-0.16.0-py2.py3-none-any.whl.metadata)
  Downloading folium-0.16.0-py2.py3-none-any.whl.metadata (3.6 kB)
Collecting branca>=0.6.0 (from folium)
  Obtaining dependency information for branca>=0.6.0 from https://files.py
thonhosted.org/packages/17/ce/14166d0e273d12065516625fb02426350298e7b4ba59
198b5fe454b46202/branca-0.7.1-py3-none-any.whl.metadata (https://files.pyt
honhosted.org/packages/17/ce/14166d0e273d12065516625fb02426350298e7b4ba591
98b5fe454b46202/branca-0.7.1-py3-none-any.whl.metadata)
  Downloading branca-0.7.1-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: jinja2>=2.9 in c:\programdata\anaconda3\lib
\site-packages (from folium) (3.1.2)
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-
packages (from folium) (1.24.3)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\si
te-packages (from folium) (2.31.0)
Requirement already satisfied: xyzservices in c:\programdata\anaconda3\lib
\site-packages (from folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3
\lib\site-packages (from jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata
\anaconda3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\li
b\site-packages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anacon
da3\lib\site-packages (from requests->folium) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anacon
da3\lib\site-packages (from requests->folium) (2023.7.22)
Downloading folium-0.16.0-py2.py3-none-any.whl (100 kB)
   ---------------------------------------- 0.0/100.0 kB ? eta -:--:--
   ---- ----------------------------------- 10.2/100.0 kB ? eta -:--:--
   ----------- ---------------------------- 30.7/100.0 kB 435.7 kB/s eta 0:
00:01
   ----------------------- ---------------- 61.4/100.0 kB 544.7 kB/s eta 0:
00:01
   ---------------------------------------- 100.0/100.0 kB 641.2 kB/s eta 0:
00:00
Downloading branca-0.7.1-py3-none-any.whl (25 kB)
Installing collected packages: branca, folium
Successfully installed branca-0.7.1 folium-0.16.0
```

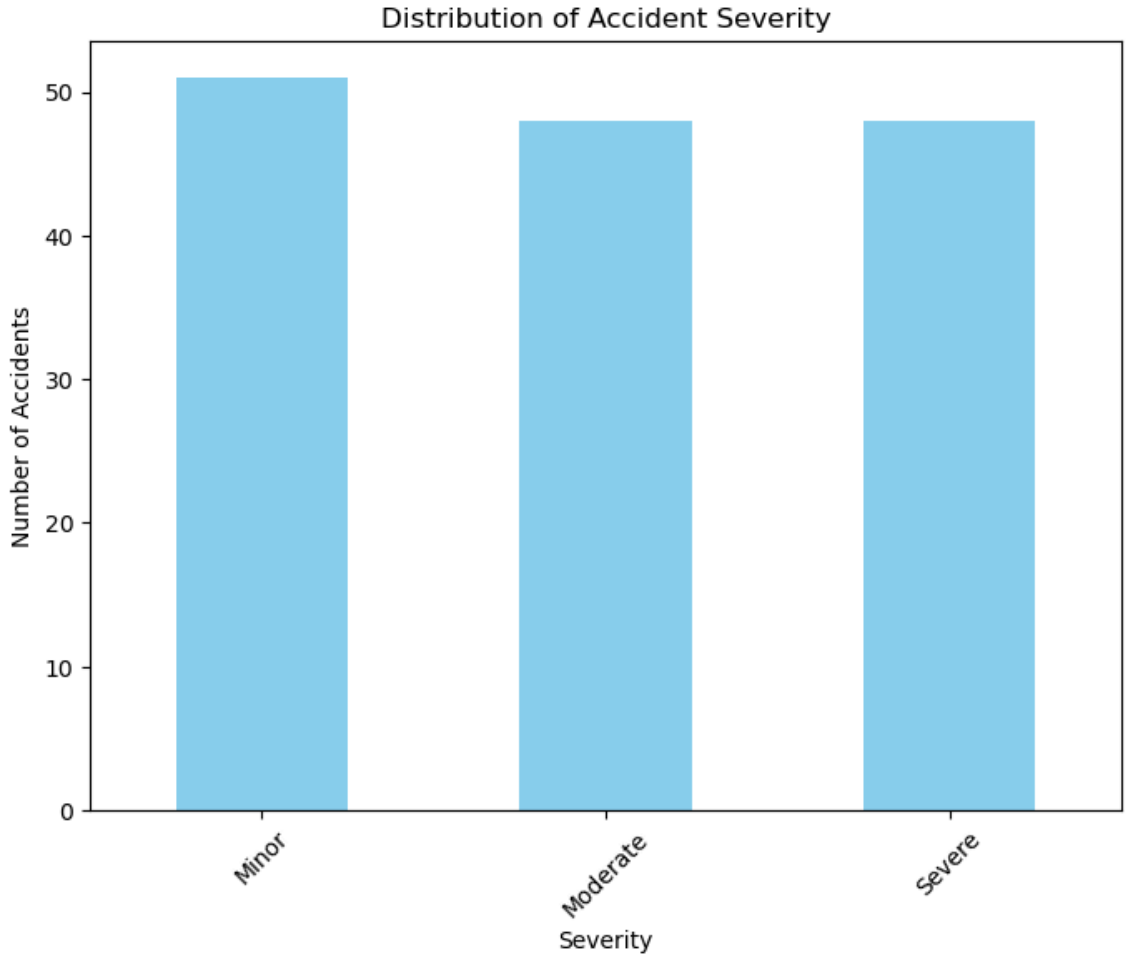
In [33]:
```
import folium
map_accidents=folium.Map(location=[data['Latitude'].mean(), data['Longitude
for index, row in data.iterrows():
    folium.CircleMarker(location=[row['Latitude'], row['Longitude']],
                        radius=3,
                        color='red',
                        fill=True,
                        fill_color='red',
                        fill_opacity=0.6,
                        popup=f"Latitude: {row['Latitude']}, Longitude: {ro
map_accidents
```

Out[33]: Make this Notebook Trusted to load map: File -> Trust Notebook

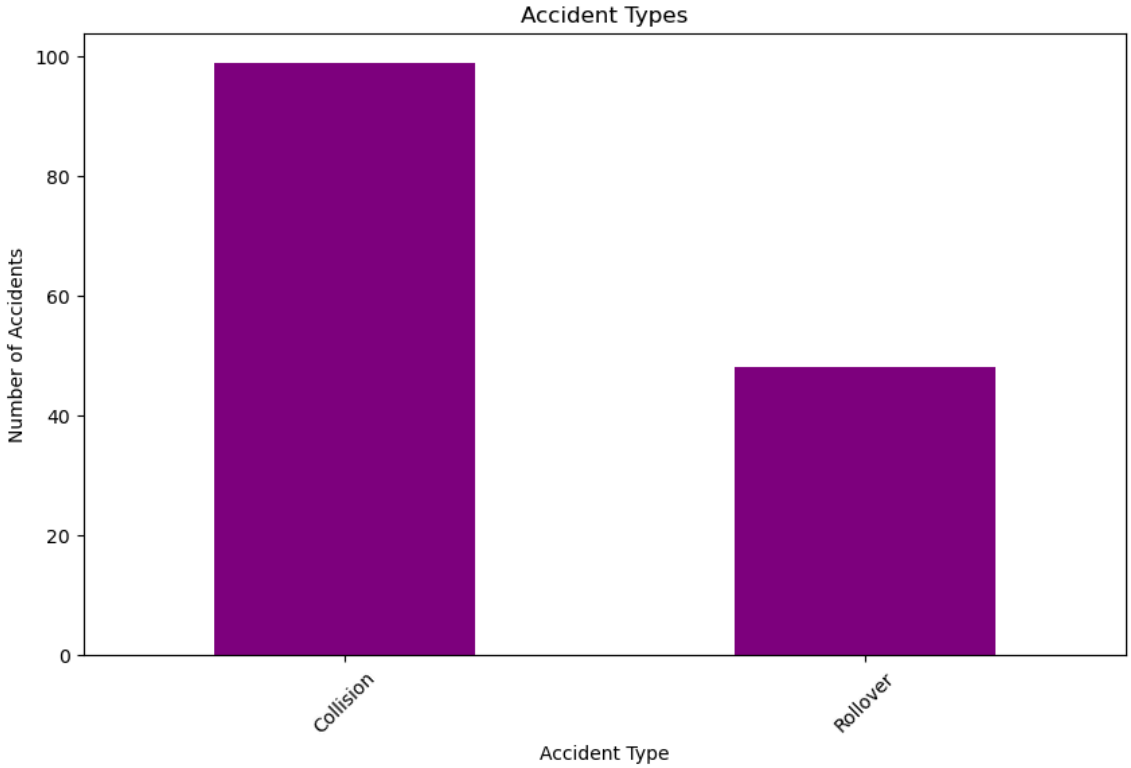
In [34]:
```
plt.figure(figsize=(10, 6))
sns.countplot(y='Contributing Factors', data=data)
plt.title('Distribution of Contributing Factors to Accidents')
plt.xlabel('Number of Accidents')
plt.ylabel('Contributing Factors')
plt.show()
```



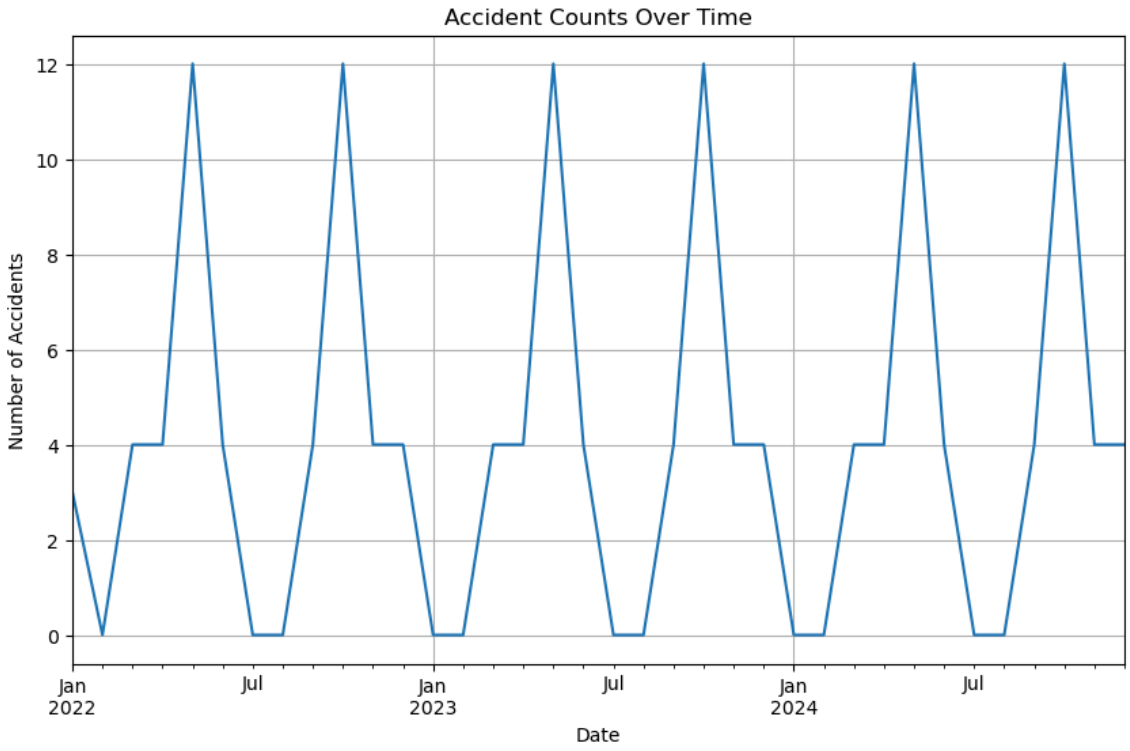
In [35]:
```
plt.figure(figsize=(8, 6))
data['Injury Severity'].value_counts().plot(kind='bar', color='skyblue')
plt.title('Distribution of Accident Severity')
plt.xlabel('Severity')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```



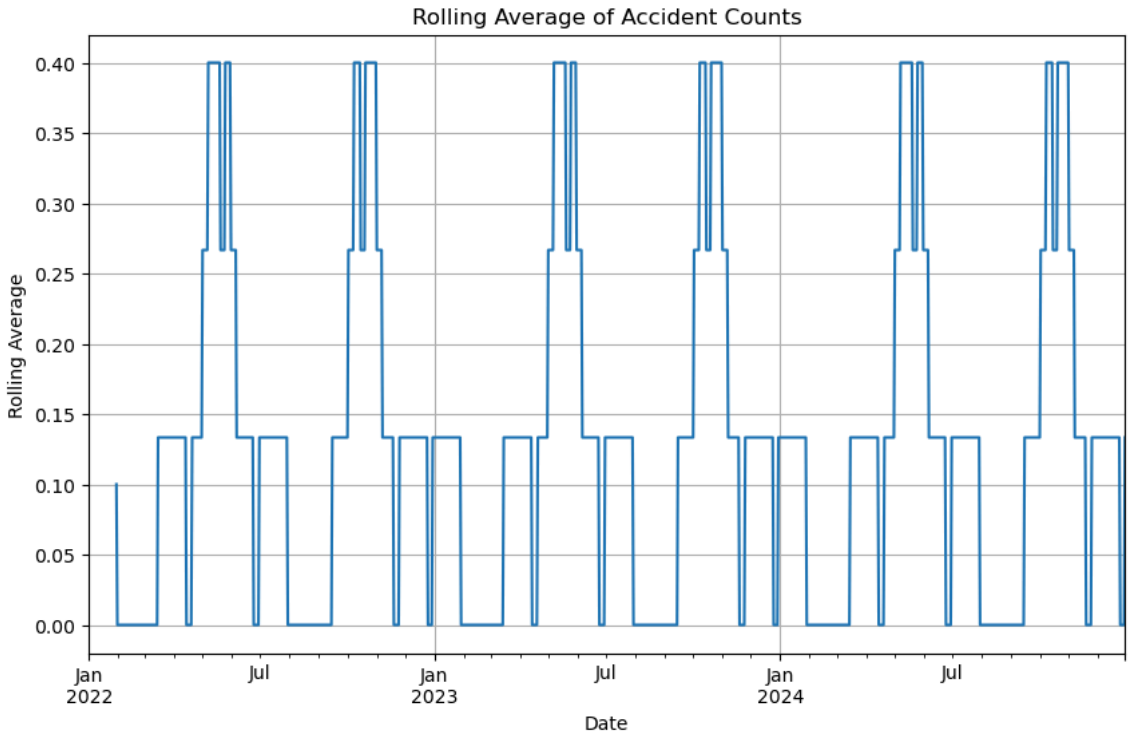
In [36]:
```
plt.figure(figsize=(10, 6))
data['Accident Type'].value_counts().plot(kind='bar', color='purple')
plt.title('Accident Types')
plt.xlabel('Accident Type')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.show()
```



In [37]:
```
plt.figure(figsize=(10, 6))
data.resample('M').size().plot()
plt.title('Accident Counts Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```



In [38]:
```
plt.figure(figsize=(10, 6))
data.resample('D').size().rolling(window=30).mean().plot()
plt.title('Rolling Average of Accident Counts')
plt.xlabel('Date')
plt.ylabel('Rolling Average')
plt.grid(True)
plt.show()
```

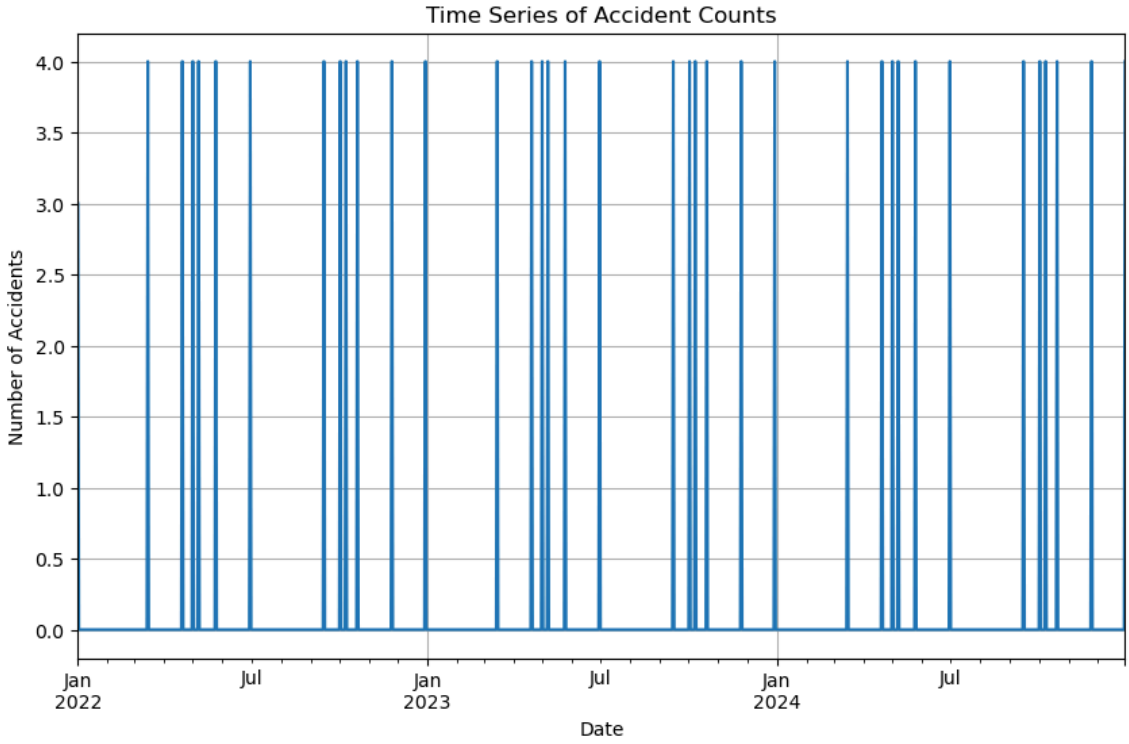


In [39]:
```
import plotly.express as px
```

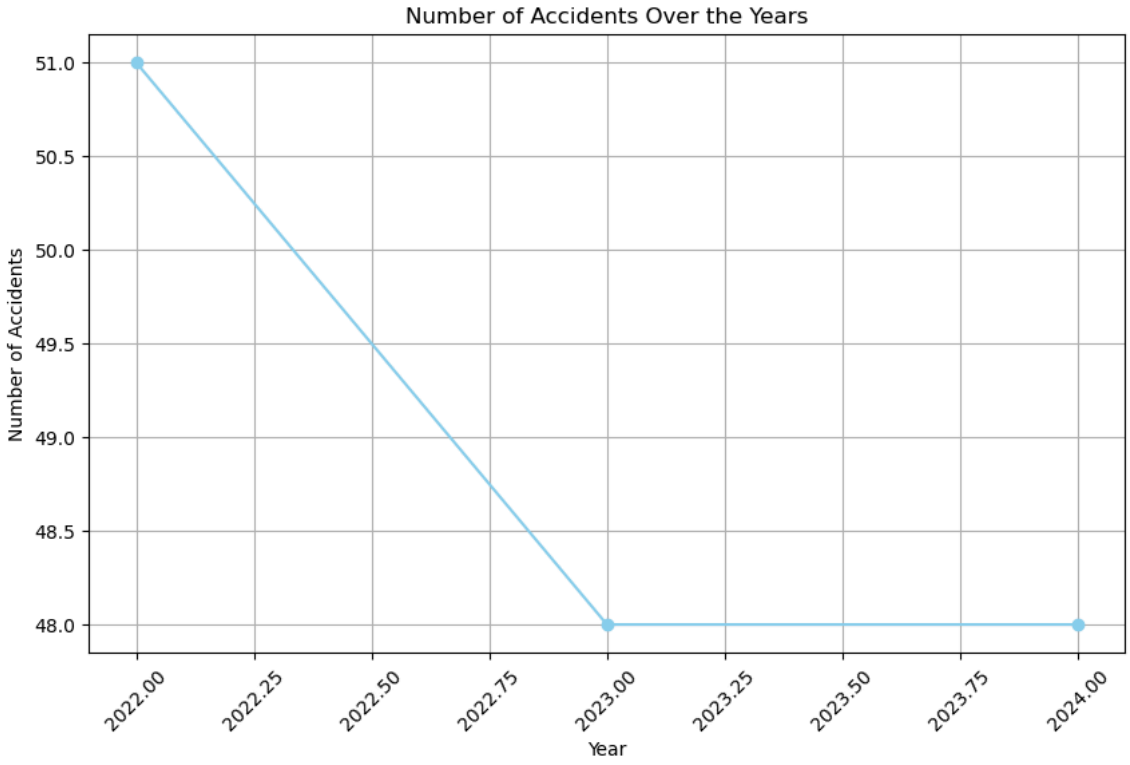
In [40]:
```
fig = px.scatter(data, x='Longitude', y='Latitude', color='Injury Severity'
fig.update_layout(title='Interactive Scatter Plot of Accident Locations',
                  xaxis_title='Longitude',
                  yaxis_title='Latitude')
fig.show()
```

In [41]:
```
import matplotlib.pyplot as plt
```

In [42]:
```
data.index=pd.to_datetime(data.index)
plt.figure(figsize=(10, 6))
data.resample('D').size().plot()
plt.title('Time Series of Accident Counts')
plt.xlabel('Date')
plt.ylabel('Number of Accidents')
plt.grid(True)
plt.show()
```



In [43]:
```
data['Year'] = data.index.year
accidents_by_year = data['Year'].value_counts().sort_index()
plt.figure(figsize=(10, 6))
accidents_by_year.plot(color='skyblue', marker='o', linestyle='-')
plt.title('Number of Accidents Over the Years')
plt.xlabel('Year')
plt.ylabel('Number of Accidents')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



In [ ]: