

```
In [1]: # import important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [52]: #Read CSV File
df = pd.read_csv(r"C:\Users\yashi\Downloads\traffic accident data.csv")
print("CSV file loaded successfully!")
```

CSV file loaded successfully!

```
In [5]: # import important libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#Read CSV File
df = pd.read_csv(r"C:\Users\yashi\Downloads\traffic accident data.csv")
print("CSV file loaded successfully!")
#Show Dataset Information
print("Dataset Information:")
print(df.info())
```

CSV file loaded successfully!

Dataset Information:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 209306 entries, 0 to 209305

Data columns (total 24 columns):

#	Column	Non-Null Count	Dtype
0	crash_date	209306 non-null	object
1	traffic_control_device	209306 non-null	object
2	weather_condition	209306 non-null	object
3	lighting_condition	209306 non-null	object
4	first_crash_type	209306 non-null	object
5	trafficway_type	209306 non-null	object
6	alignment	209306 non-null	object
7	roadway_surface_cond	209306 non-null	object
8	road_defect	209306 non-null	object
9	crash_type	209306 non-null	object
10	intersection_related_i	209306 non-null	object
11	damage	209306 non-null	object
12	prim_contributory_cause	209306 non-null	object
13	num_units	209306 non-null	int64
14	most_severe_injury	209306 non-null	object
15	injuries_total	209306 non-null	int64
16	injuries_fatal	209306 non-null	int64
17	injuries_incapacitating	209306 non-null	int64
18	injuries_non_incapacitating	209306 non-null	int64
19	injuries_reported_not_evident	209306 non-null	int64
20	injuries_no_indication	209306 non-null	int64
21	crash_hour	209306 non-null	int64
22	crash_day_of_week	209306 non-null	int64
23	crash_month	209306 non-null	int64

dtypes: int64(10), object(14)

memory usage: 38.3+ MB

None

```
In [5]: # Summary Statistics
print("Summary Statistics:")
print(df.describe())
```

Summary Statistics:

	num_units	injuries_total	injuries_fatal	injuries_incapacitating \
count	209306.000000	209306.000000	209306.000000	209306.000000
mean	2.063300	0.382717	0.001859	0.038102
std	0.396012	0.799720	0.047502	0.233964
min	1.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	0.000000
50%	2.000000	0.000000	0.000000	0.000000
75%	2.000000	1.000000	0.000000	0.000000
max	11.000000	21.000000	3.000000	7.000000

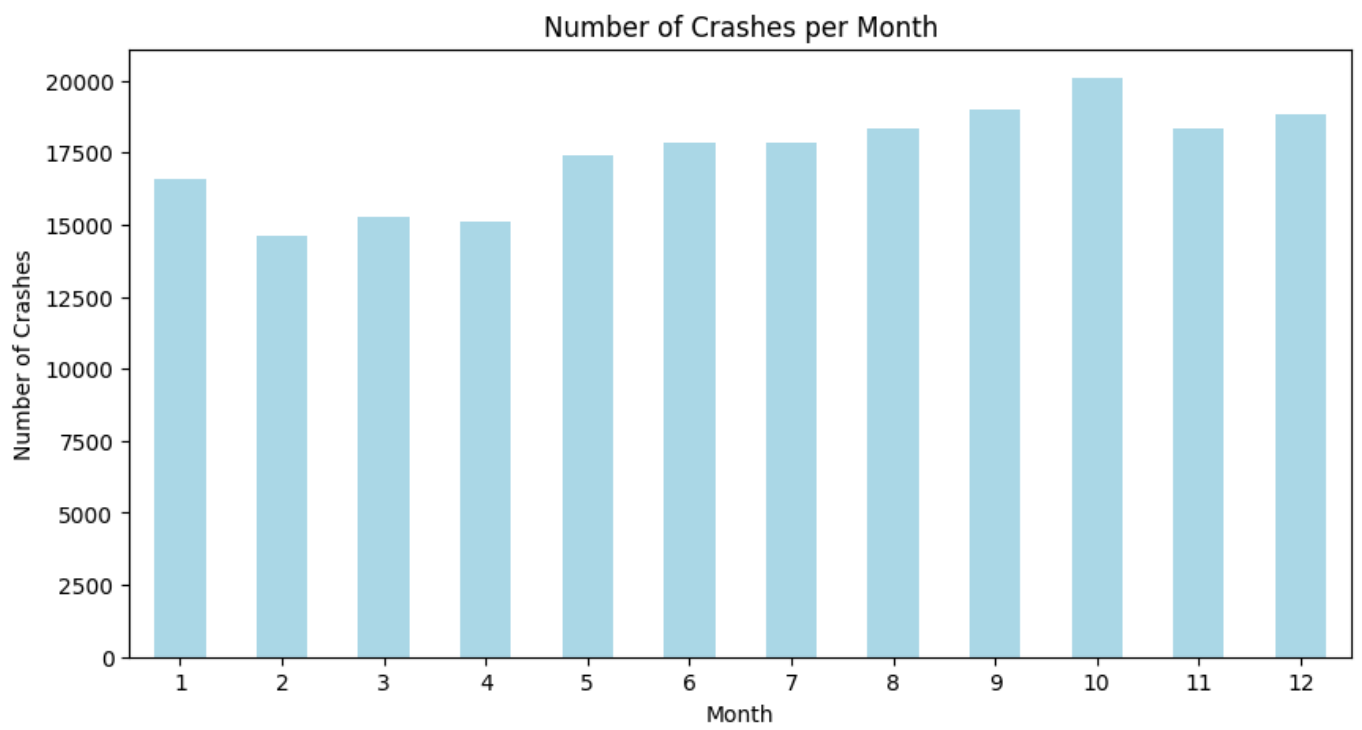
	injuries_non_incapacitating	injuries_reported_not_evident \
count	209306.000000	209306.000000
mean	0.221241	0.121516
std	0.614960	0.450865
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	21.000000	15.000000

	injuries_no_indication	crash_hour	crash_day_of_week	crash_month
count	209306.000000	209306.000000	209306.000000	209306.000000
mean	2.244002	13.373047	4.144024	6.771822
std	1.241175	5.603830	1.966864	3.427593
min	0.000000	0.000000	1.000000	1.000000
25%	2.000000	9.000000	2.000000	4.000000
50%	2.000000	14.000000	4.000000	7.000000
75%	3.000000	17.000000	6.000000	10.000000
max	49.000000	23.000000	7.000000	12.000000

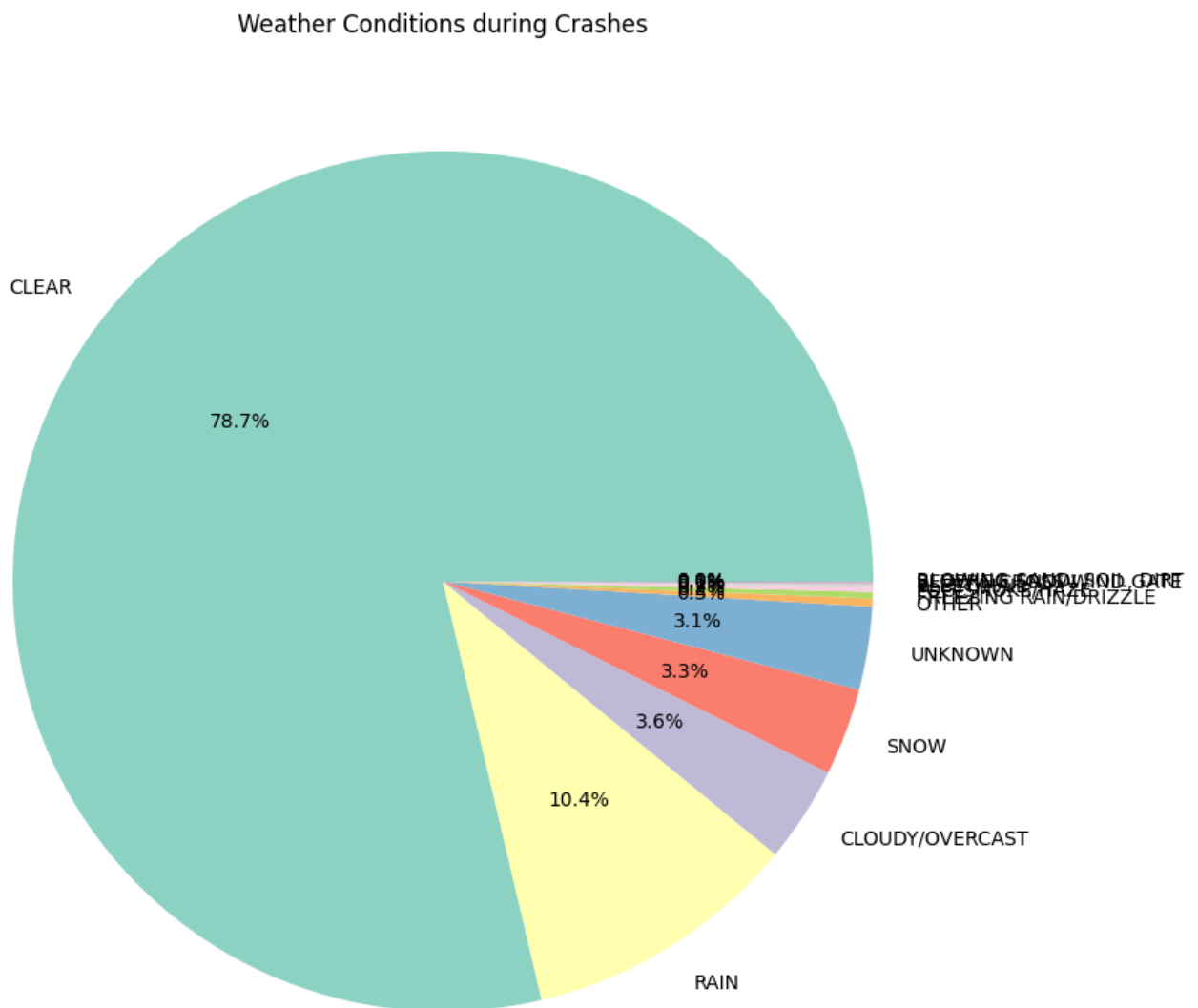
```
In [6]: # Data Cleaning
# Removing duplicates
df.drop_duplicates(inplace=True)
# Checking for missing values
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values[missing_values > 0])
```

Missing Values:
Series([], dtype: int64)

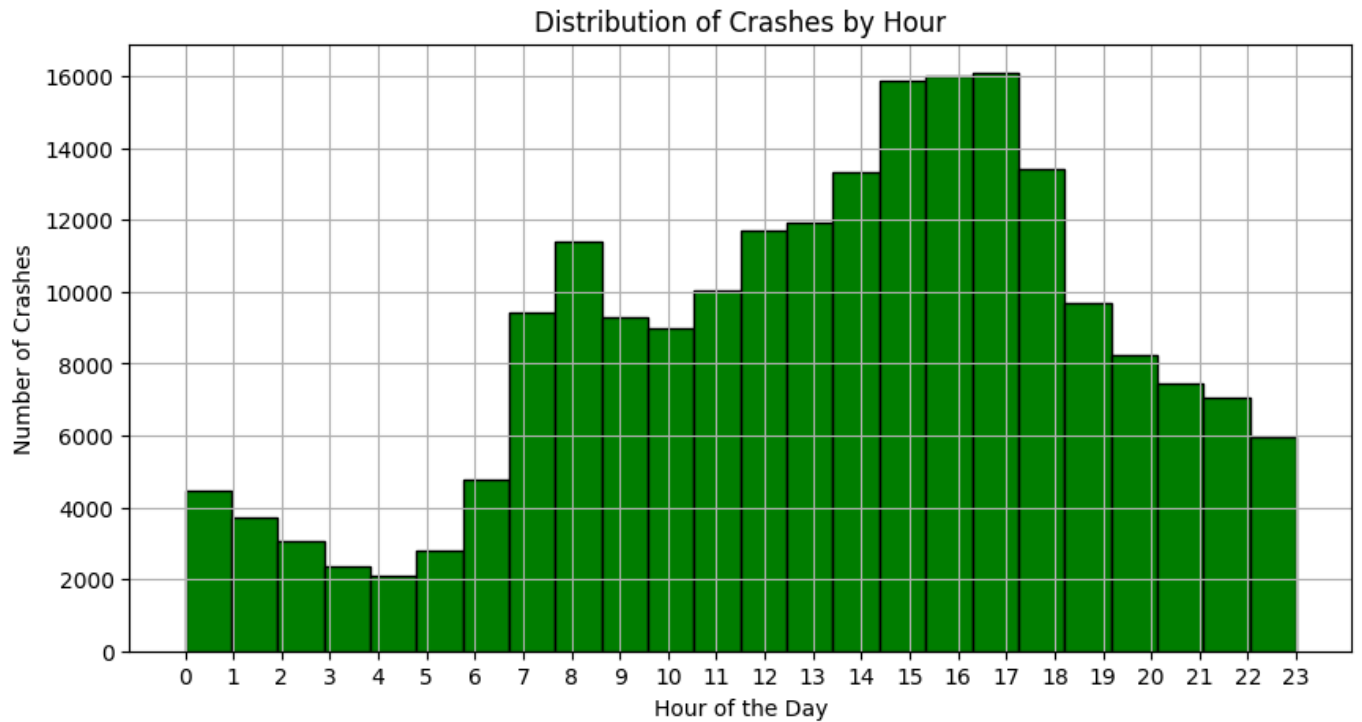
```
In [9]: # analysis the Crash Count by Month
plt.figure(figsize=(10, 5))
df["crash_month"].value_counts().sort_index().plot(kind='bar', color='lightblue')
plt.title("Number of Crashes per Month")
plt.xlabel("Month")
plt.ylabel("Number of Crashes")
plt.xticks(rotation=0)
plt.show()
```



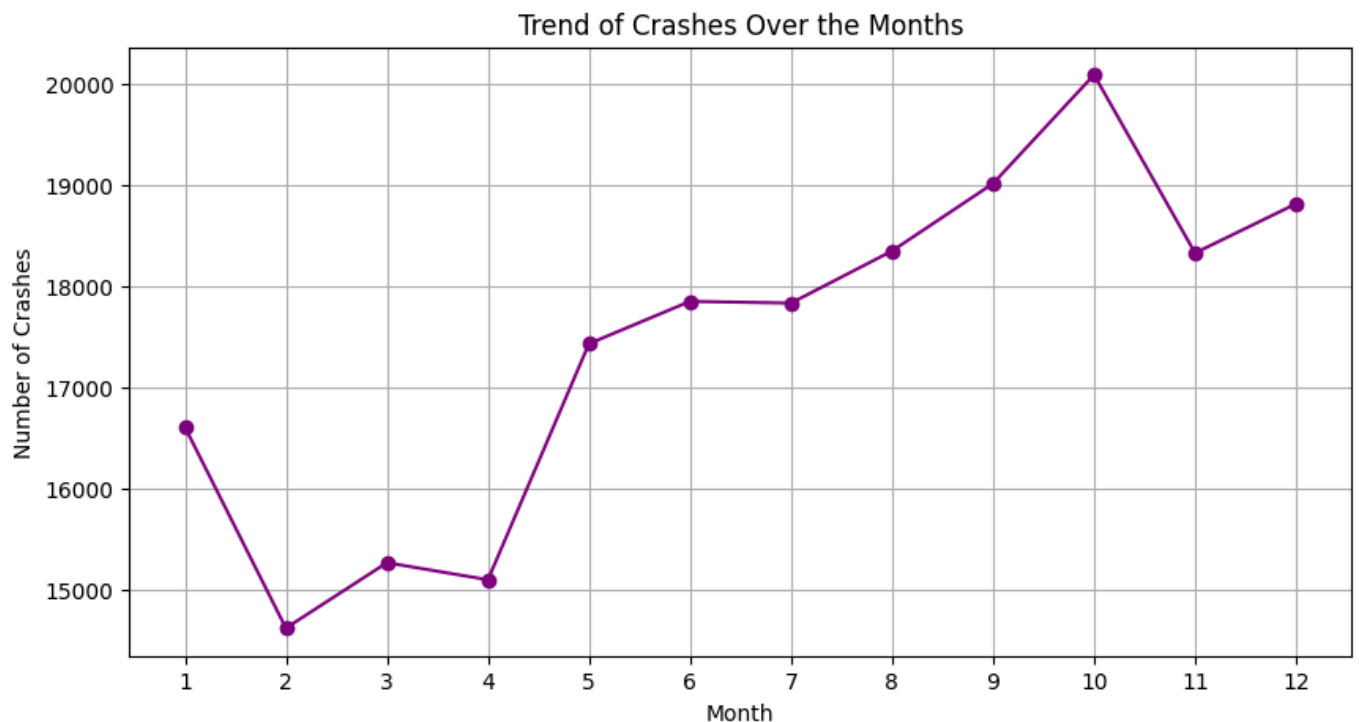
```
In [13]: # Analysis the Weather Conditions during Crashes
plt.figure(figsize=(10, 10))
df["weather_condition"].value_counts().plot(kind='pie', autopct='%1.1f%%', colormap='Set3')
plt.title("Weather Conditions during Crashes")
plt.ylabel("")
plt.show()
```



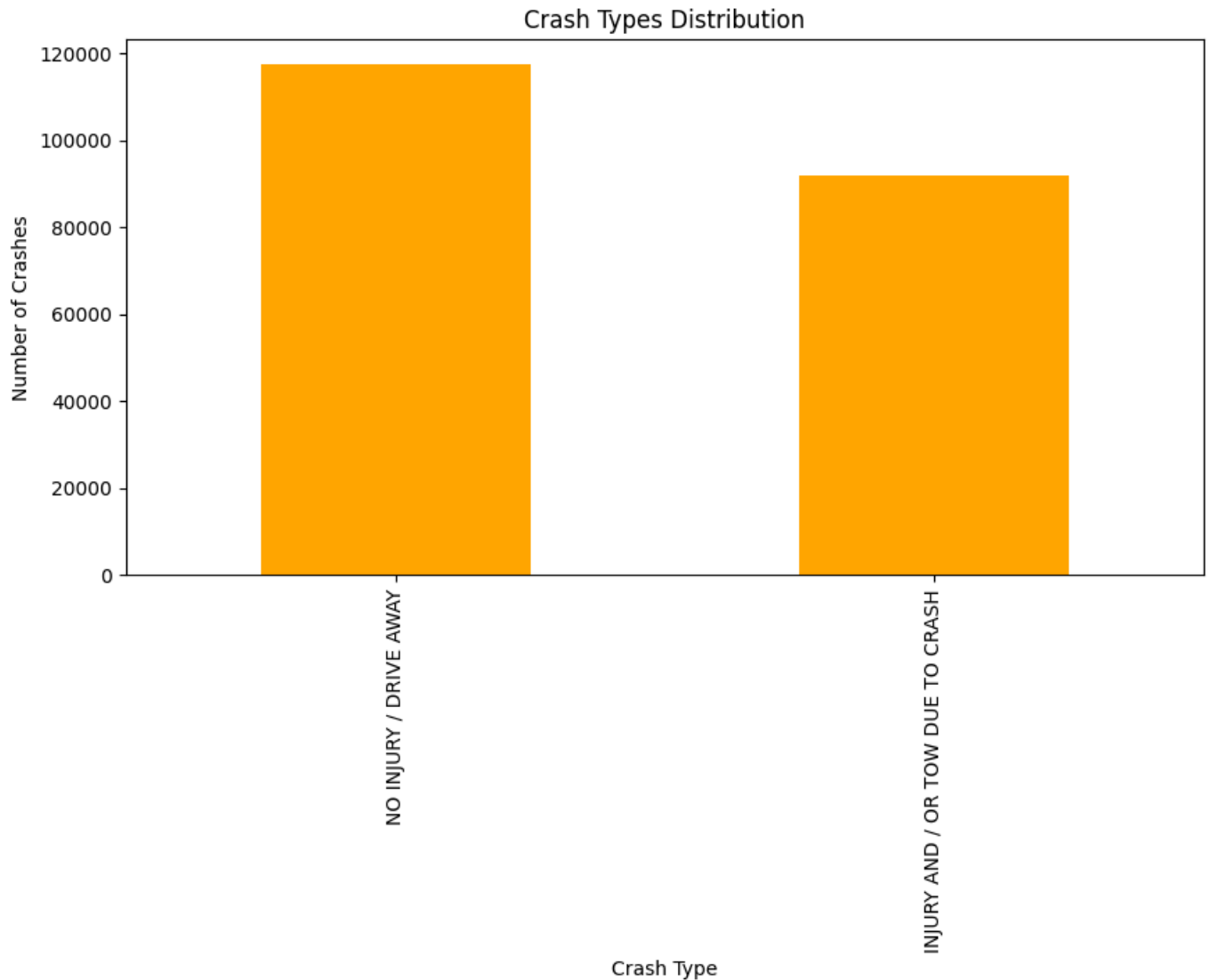
```
In [17]: # analysis the Crash Hour Distribution
plt.figure(figsize=(10, 5))
df["crash_hour"].hist(bins=24, color='green', edgecolor='black')
plt.title("Distribution of Crashes by Hour")
plt.xlabel("Hour of the Day")
plt.ylabel("Number of Crashes")
plt.xticks(range(0, 24))
plt.show()
```



```
In [22]: # Analysis the Trend of Crashes Over the Months
plt.figure(figsize=(10, 5))
df.groupby("crash_month").size().sort_index().plot(kind='line', marker='o', linestyle='-', color='purple')
plt.title("Trend of Crashes Over the Months")
plt.xlabel("Month")
plt.ylabel("Number of Crashes")
plt.grid(True)
plt.xticks(range(1, 13))
plt.show()
```

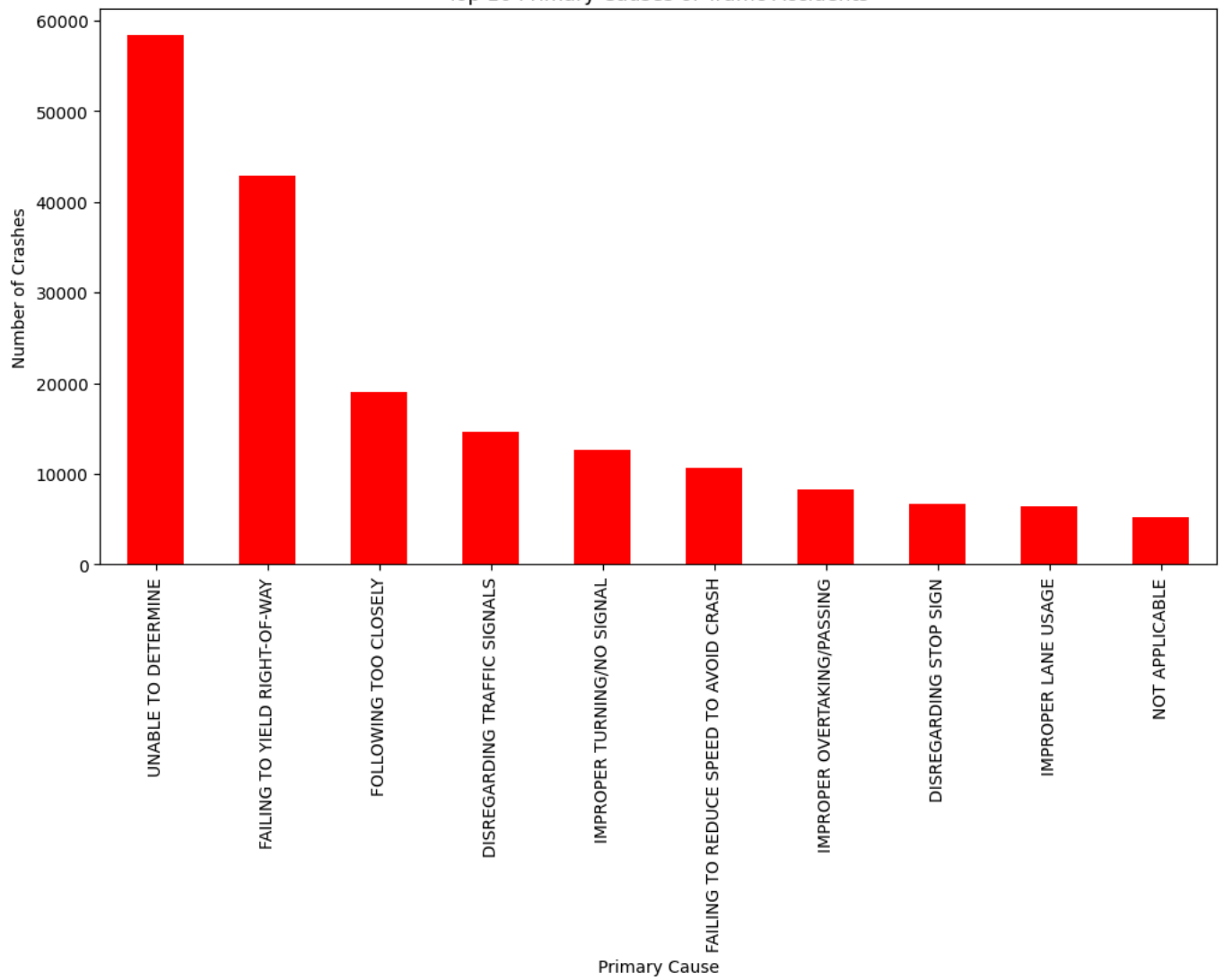


```
In [23]: # Analysis the Crash Types
plt.figure(figsize=(10, 5))
df["crash_type"].value_counts().plot(kind='bar', color='orange')
plt.title("Crash Types Distribution")
plt.xlabel("Crash Type")
plt.ylabel("Number of Crashes")
plt.xticks(rotation=90)
plt.show()
```

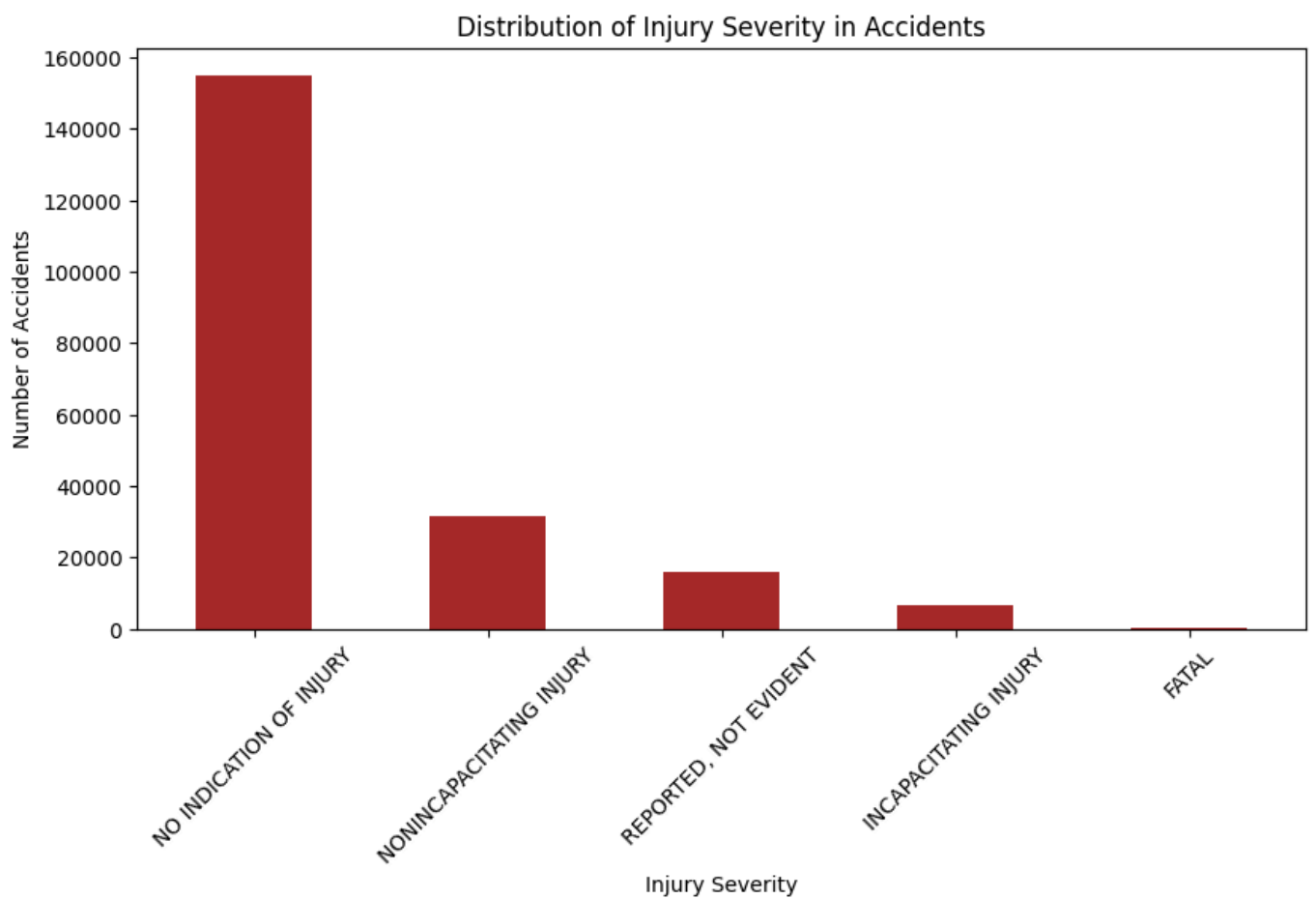


```
In [26]: # Analysis the Primary Contributory Cause of Crashes
plt.figure(figsize=(12, 6))
df["prim_contributory_cause"].value_counts().head(10).plot(kind='bar', color='red')
plt.title("Top 10 Primary Causes of Traffic Accidents")
plt.xlabel("Primary Cause")
plt.ylabel("Number of Crashes")
plt.xticks(rotation=90)
plt.show()
```

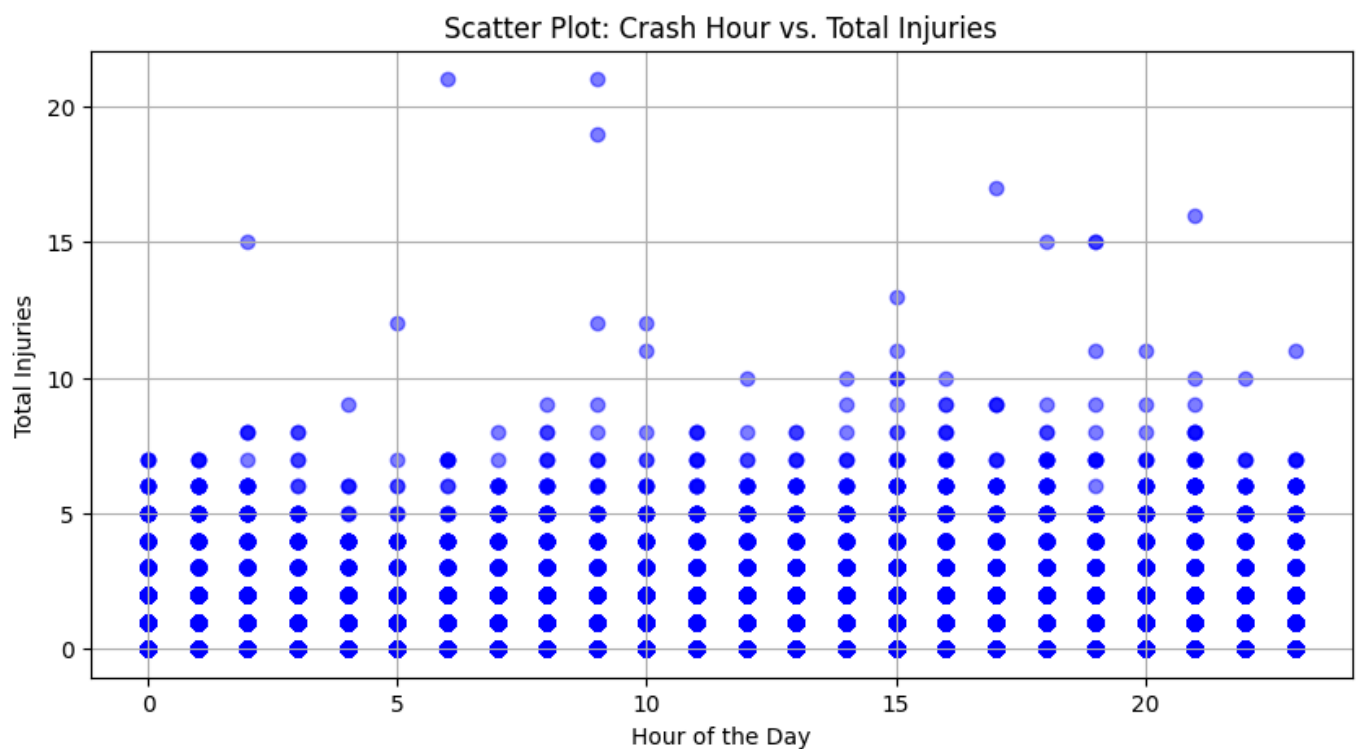
Top 10 Primary Causes of Traffic Accidents



```
In [28]: # Analysis the Severity of Injuries
plt.figure(figsize=(10, 5))
df["most_severe_injury"].value_counts().plot(kind='bar', color='brown')
plt.title("Distribution of Injury Severity in Accidents")
plt.xlabel("Injury Severity")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.show()
```

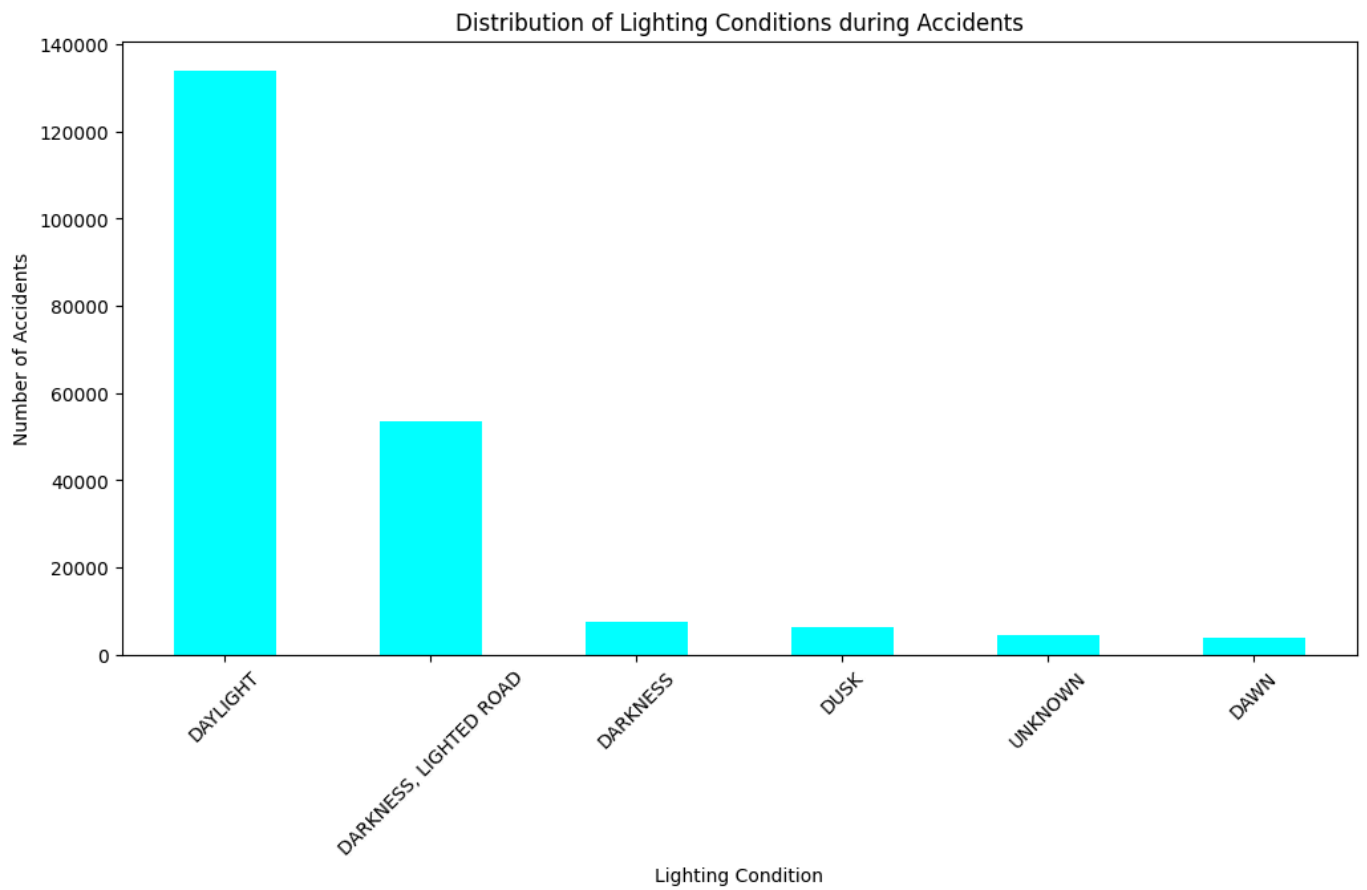


```
In [33]: # Scatter Plot - Crash Hour vs. Number of Injuries
plt.figure(figsize=(10, 5))
plt.scatter(df["crash_hour"], df["injuries_total"], alpha=0.5, color='blue')
plt.title("Scatter Plot: Crash Hour vs. Total Injuries")
plt.xlabel("Hour of the Day")
plt.ylabel("Total Injuries")
plt.grid(True)
plt.show()
```



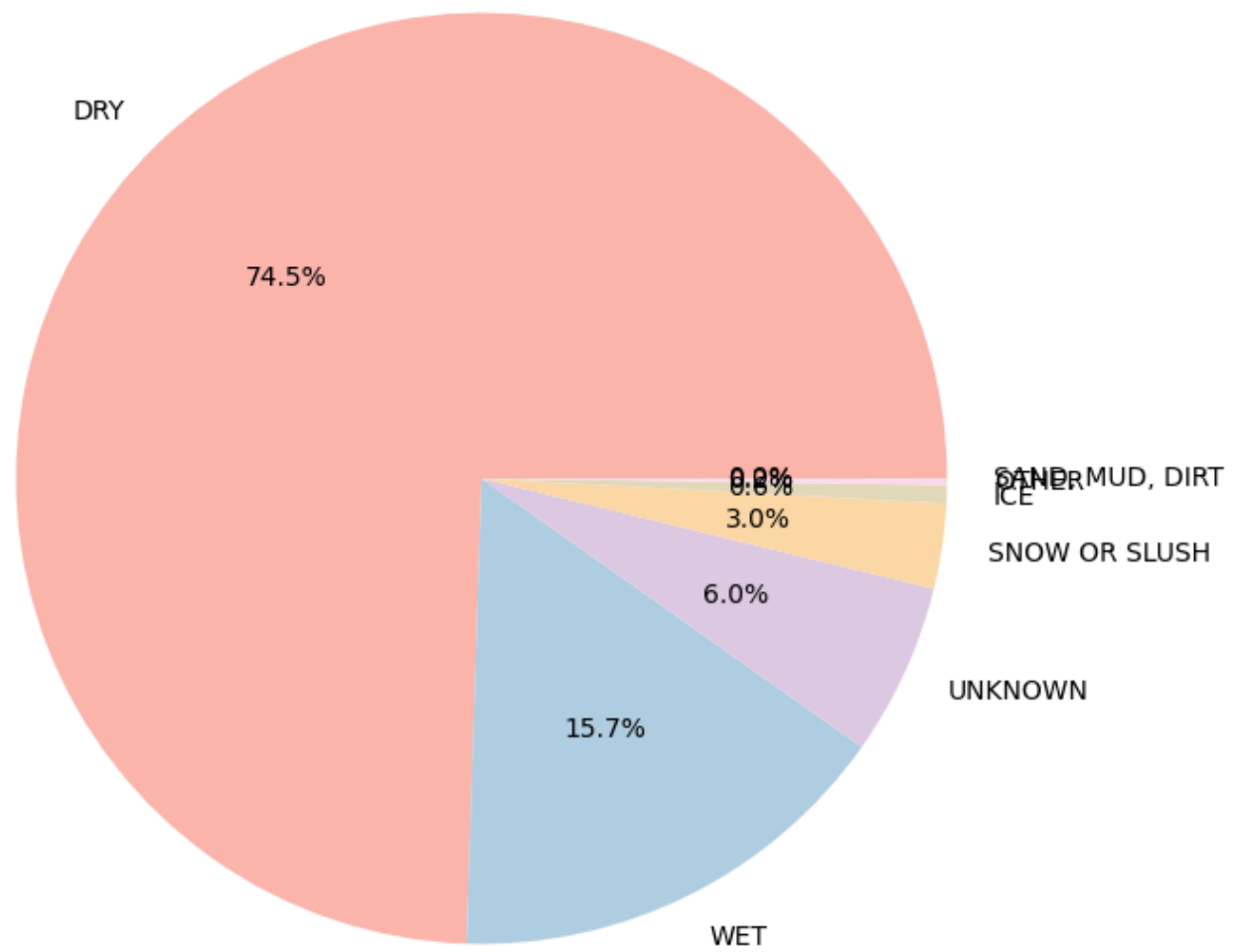
```
In [34]: # Analysis Lighting Conditions during Accidents
plt.figure(figsize=(12, 6))
df["lighting_condition"].value_counts().plot(kind='bar', color='cyan')
plt.title("Distribution of Lighting Conditions during Accidents")
```

```
plt.xlabel("Lighting Condition")
plt.ylabel("Number of Accidents")
plt.xticks(rotation=45)
plt.show()
```

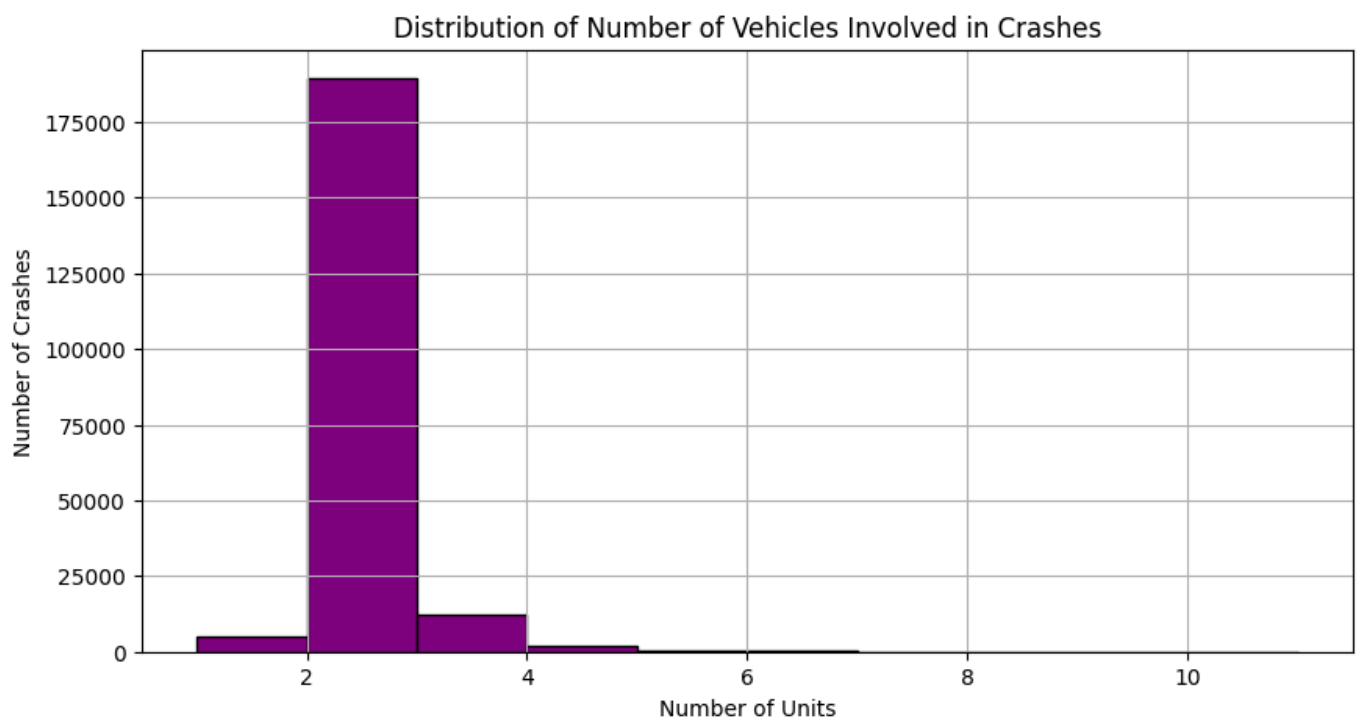


```
In [36]: # Analysis the Roadway Surface Conditions during Accidents
plt.figure(figsize=(8, 8))
df["roadway_surface_cond"].value_counts().plot(kind='pie', autopct='%1.1f%%', colormap='Pastel2')
plt.title("Roadway Surface Conditions during Accidents")
plt.ylabel("")
plt.show()
```

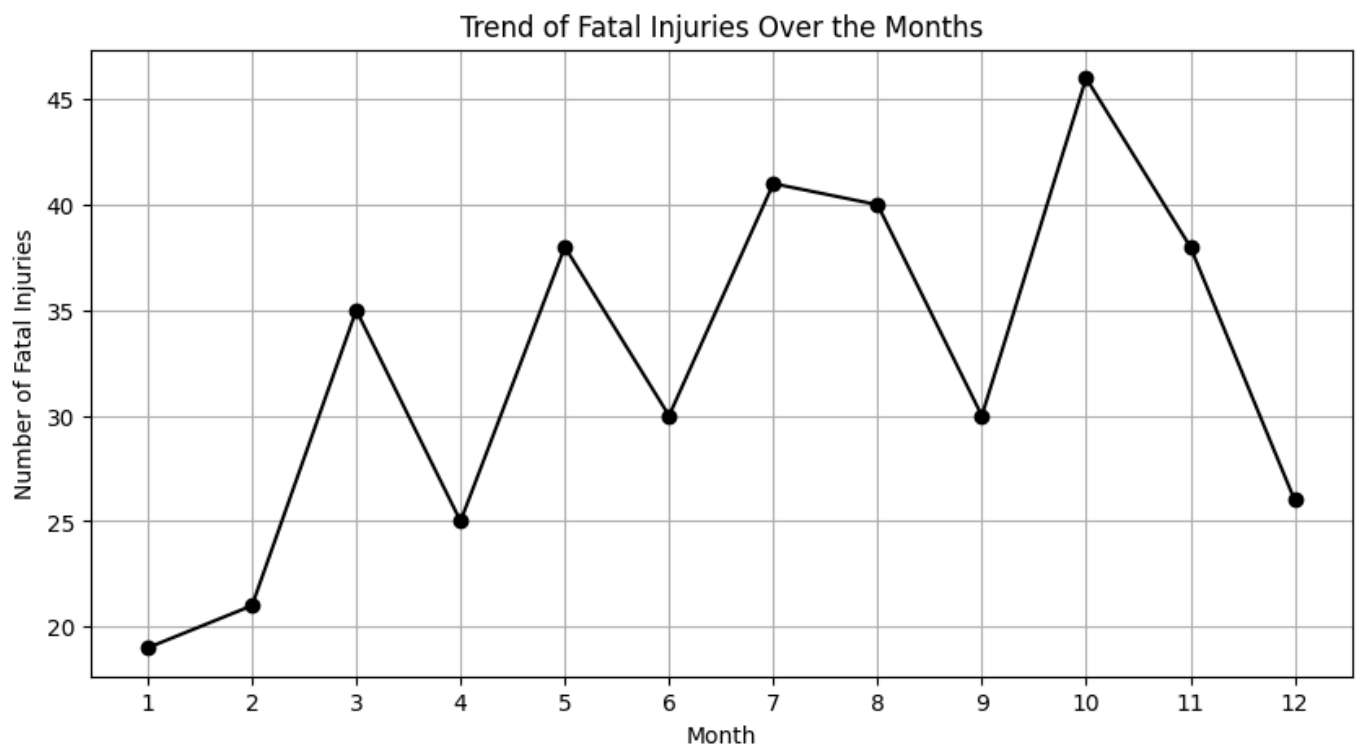

Roadway Surface Conditions during Accidents



```
In [8]: # Analysis the Number of Vehicles Involved in Crashes
plt.figure(figsize=(10, 5))
df["num_units"].hist(bins=10, color='purple', edgecolor='black')
plt.title("Distribution of Number of Vehicles Involved in Crashes")
plt.xlabel("Number of Units")
plt.ylabel("Number of Crashes")
plt.grid(True)
plt.show()
```

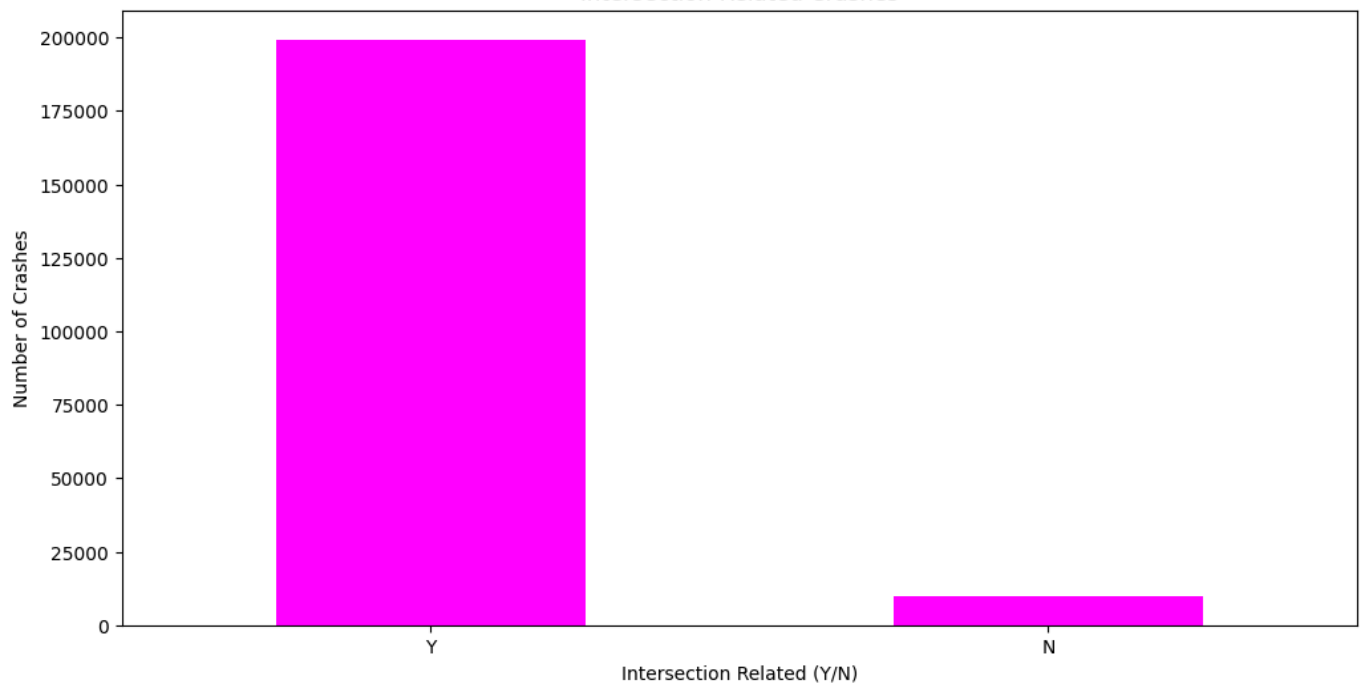


```
In [39]: #See Trend of Fatal Injuries by Month
plt.figure(figsize=(10, 5))
df.groupby("crash_month")["injuries_fatal"].sum().sort_index().plot(kind='line', marker='o',
plt.title("Trend of Fatal Injuries Over the Months")
plt.xlabel("Month")
plt.ylabel("Number of Fatal Injuries")
plt.grid(True)
plt.xticks(range(1, 13))
plt.show()
```



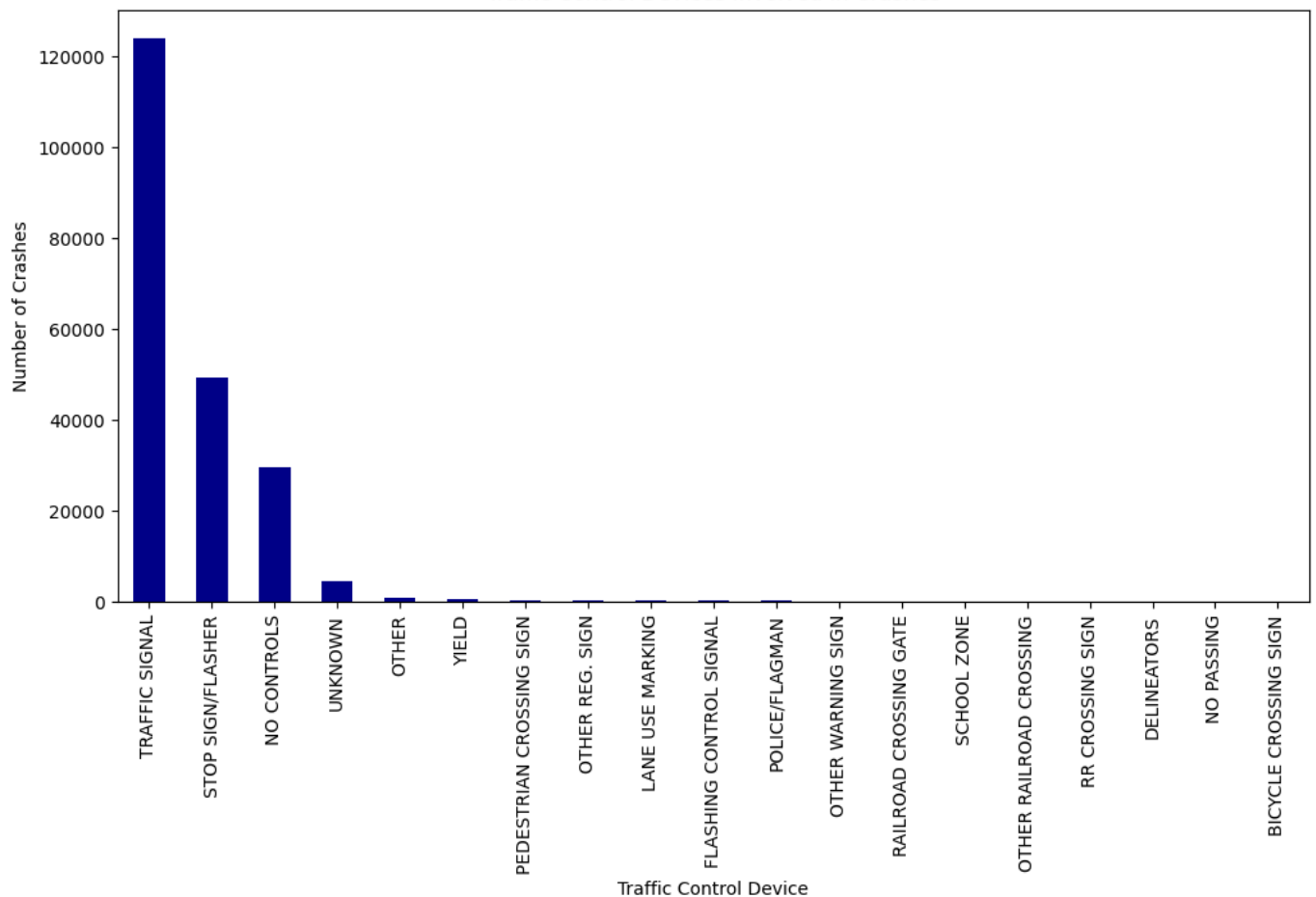
```
In [40]: # See the Intersection-Related Crashes
plt.figure(figsize=(12, 6))
df["intersection_related_i"].value_counts().plot(kind='bar', color='magenta')
plt.title("Intersection-Related Crashes")
plt.xlabel("Intersection Related (Y/N)")
plt.ylabel("Number of Crashes")
plt.xticks(rotation=0)
plt.show()
```

Intersection-Related Crashes



```
In [43]: # Traffic Control Devices in Crashes
plt.figure(figsize=(12, 6))
df["traffic_control_device"].value_counts().plot(kind='bar', color='darkblue')
plt.title("Traffic Control Devices Involved in Crashes")
plt.xlabel("Traffic Control Device")
plt.ylabel("Number of Crashes")
plt.xticks(rotation=90)
plt.show()
```

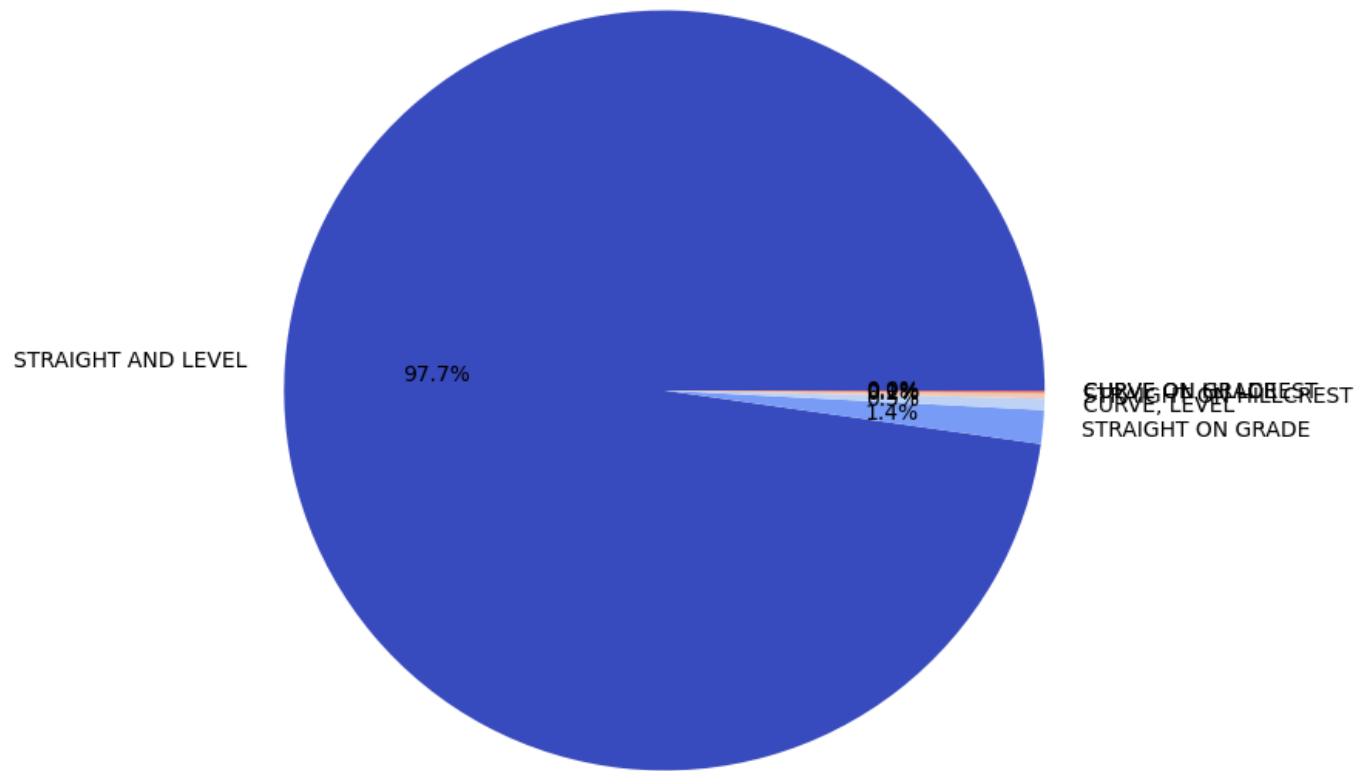
Traffic Control Devices Involved in Crashes



```
In [46]: # Alignment of Road
plt.figure(figsize=(8, 8))
df["alignment"].value_counts().plot(kind='pie', autopct='%1.1f%%', colormap='coolwarm')
plt.title("Alignment of Road during Crashes")
```

```
plt.ylabel("")  
plt.show()
```

Alignment of Road during Crashes



In []: