

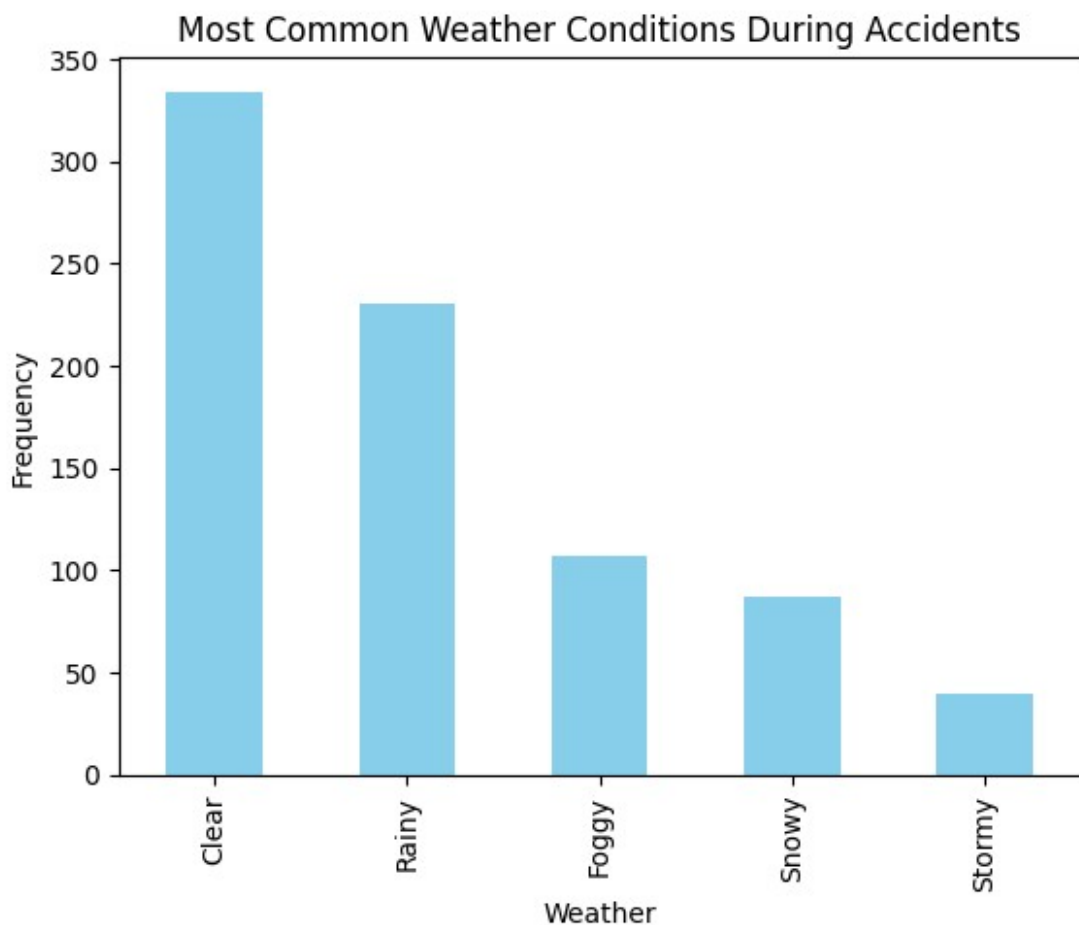
```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
data =
pd.read_csv('C:/Users/Viraj/Documents/TF/dataset_traffic_accident_prediction1.csv')

# 1. Most common weather conditions during accidents
weather_counts = data['Weather'].value_counts()
weather_counts.plot(kind='bar', color='skyblue')
plt.title("Most Common Weather Conditions During Accidents")
plt.xlabel("Weather")
plt.ylabel("Frequency")
plt.show()

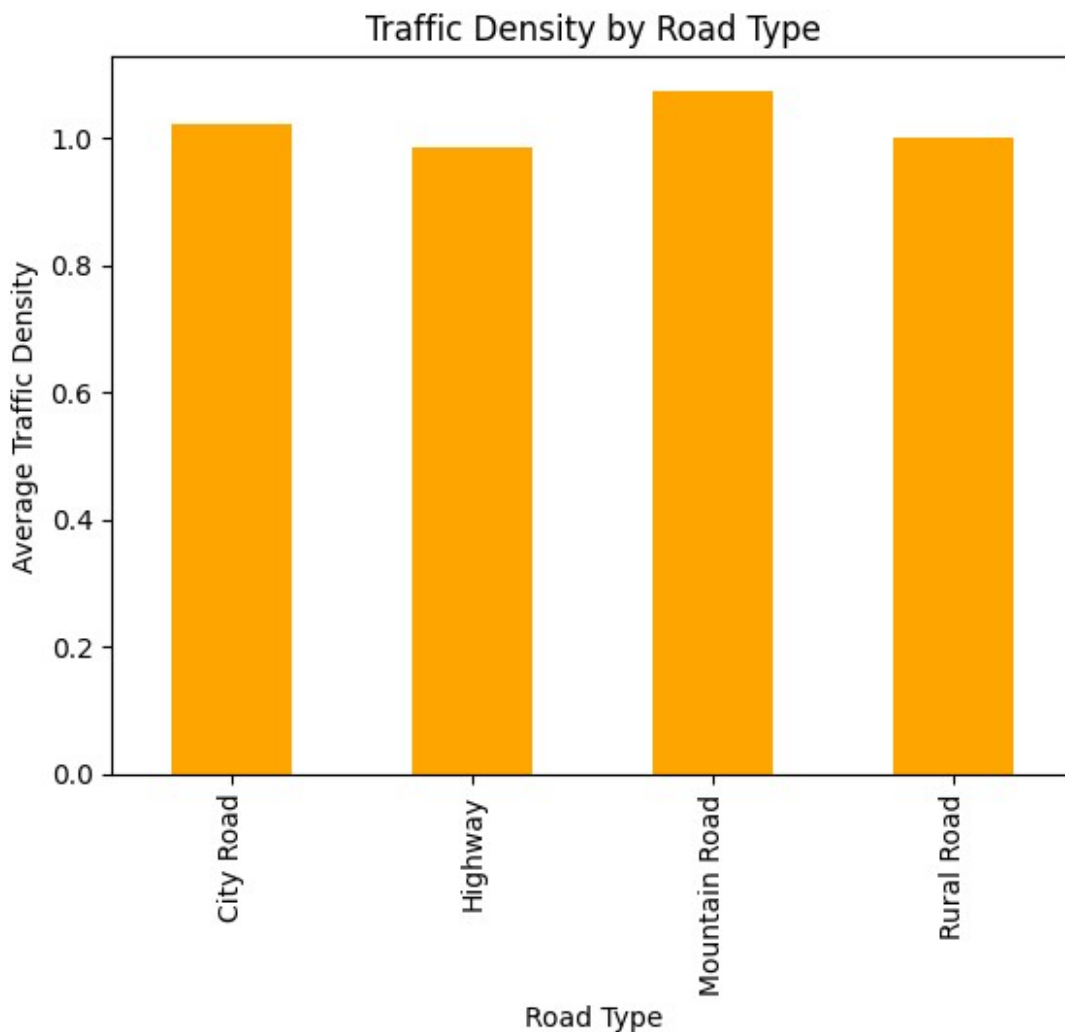
```



```
# 2. Average speed limit where accidents occur
average_speed_limit = data[data['Accident'] == 1]
['Speed_Limit'].mean()
print("Average speed limit during accidents:", average_speed_limit)
```

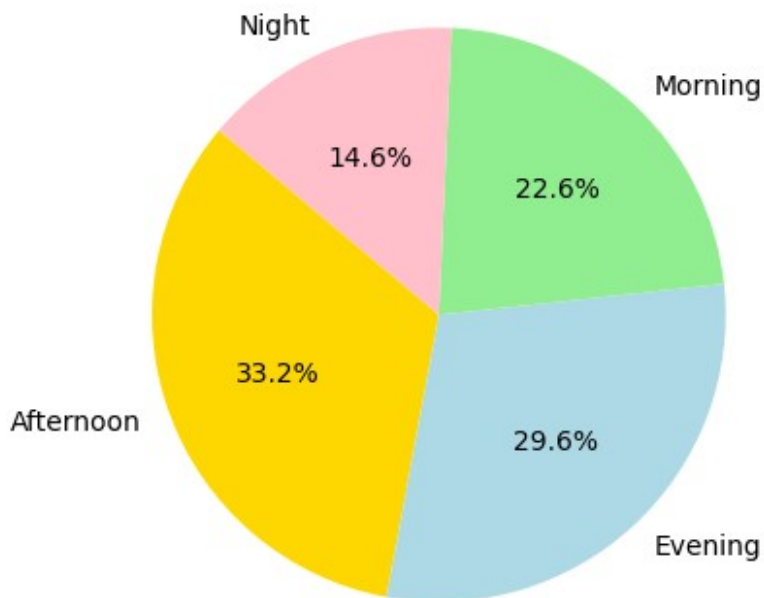
Average speed limit during accidents: 69.77533039647577

```
# 3. Traffic density by road type
traffic_density_road = data.groupby('Road_Type')
['Traffic_Density'].mean()
traffic_density_road.plot(kind='bar', color='orange')
plt.title("Traffic Density by Road Type")
plt.xlabel("Road Type")
plt.ylabel("Average Traffic Density")
plt.show()
```



```
# 4. Time of day with highest number of accidents
time_of_day_accidents = data[data['Accident'] == 1]
['Time_of_Day'].value_counts()
time_of_day_accidents.plot(kind='pie', autopct='%1.1f%%',
startangle=140, colors=['gold', 'lightblue', 'lightgreen', 'pink'])
plt.title("Time of Day with Highest Number of Accidents")
plt.ylabel("")
plt.show()
```

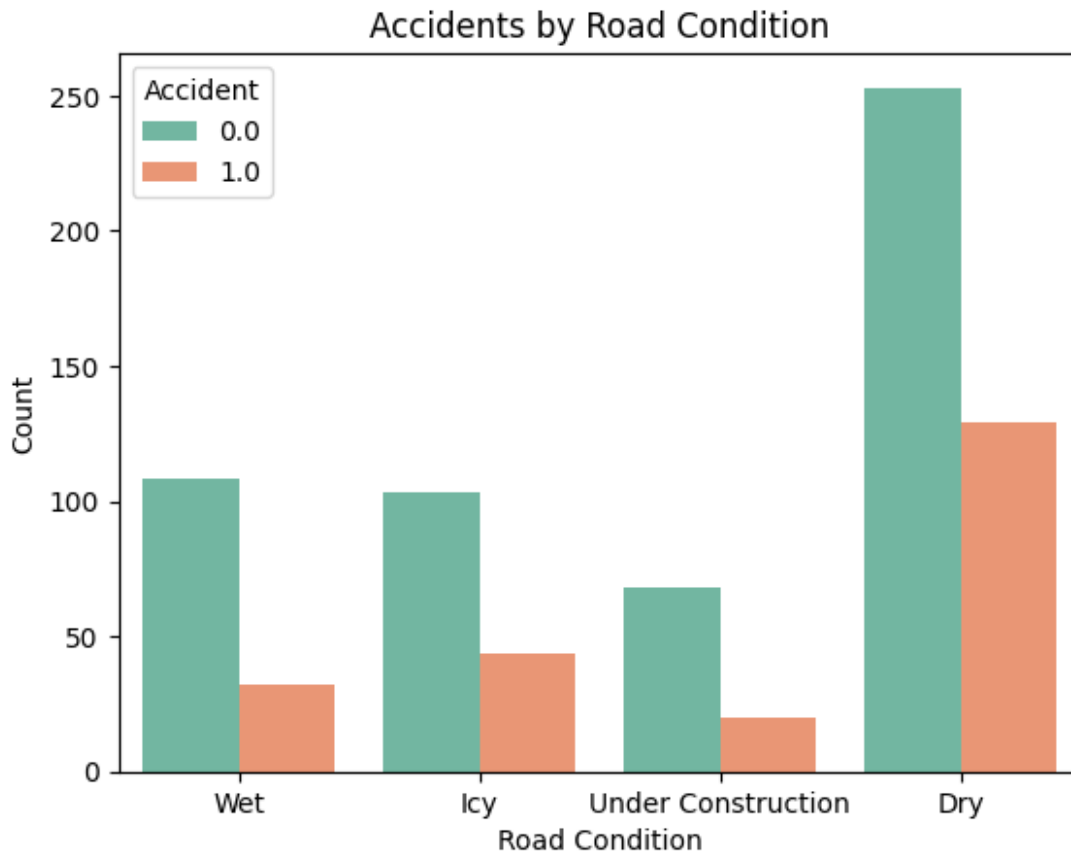
Time of Day with Highest Number of Accidents



```
# 5. Correlation between driver experience and accident severity
correlation_driver_experience =
data['Driver_Experience'].corr(data['Accident'])
print("Correlation between driver experience and accidents:",
correlation_driver_experience)
```

Correlation between driver experience and accidents:
0.019893896280492557

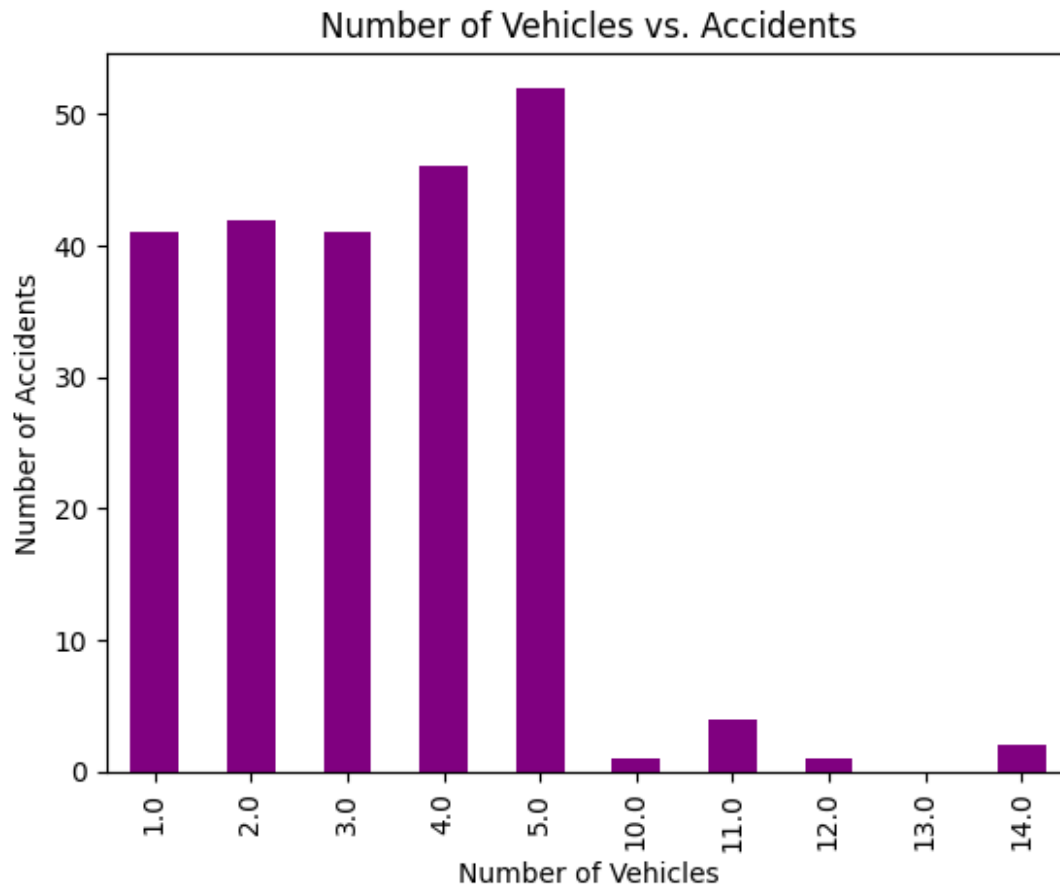
```
# 6. Distribution of accidents by road condition
sns.countplot(x='Road_Condition', hue='Accident', data=data,
palette='Set2')
plt.title("Accidents by Road Condition")
plt.xlabel("Road Condition")
plt.ylabel("Count")
plt.show()
```



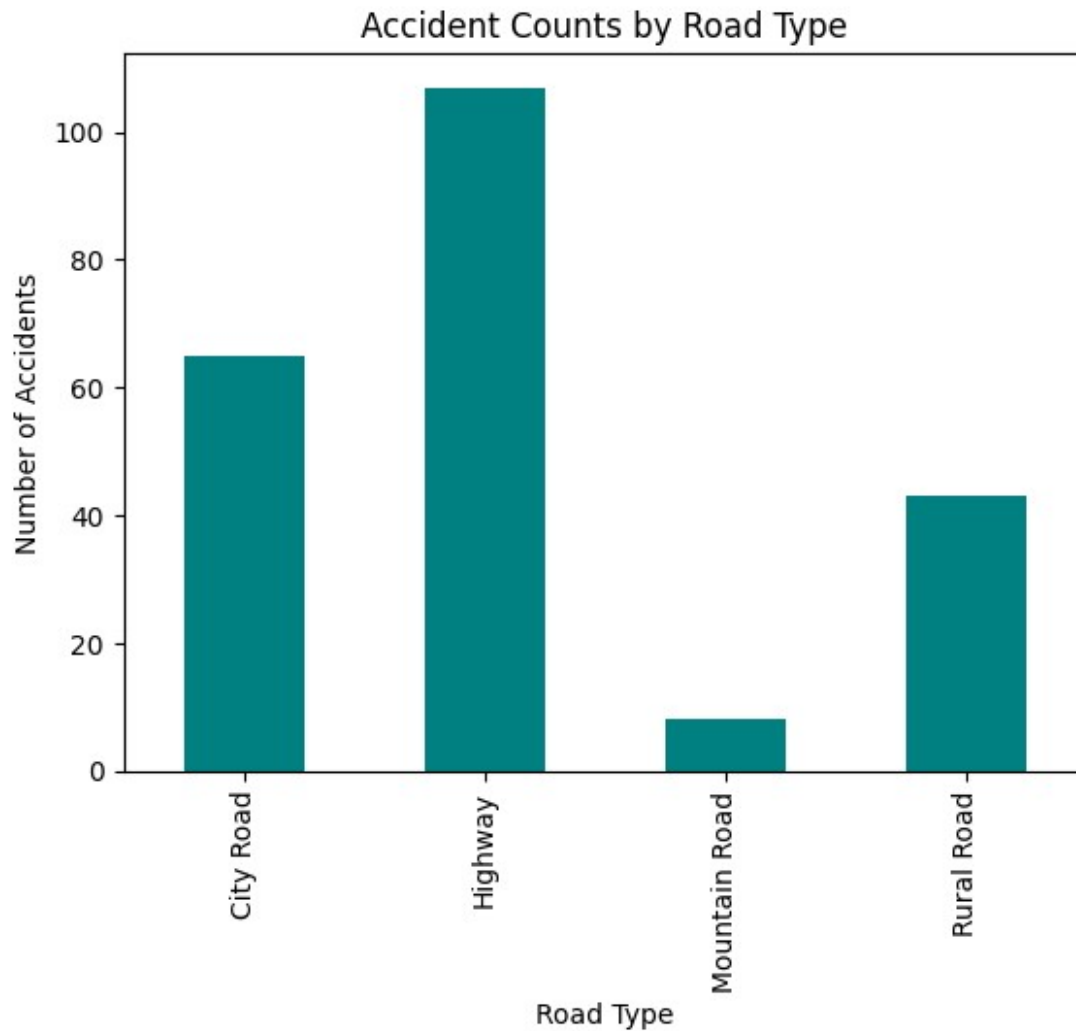
```
# 7. Percentage of accidents involving alcohol
alcohol_accidents = data[(data['Driver_Alcohol'] > 0) &
(data['Accident'] == 1)].shape[0]
total_accidents = data[data['Accident'] == 1].shape[0]
alcohol_percentage = (alcohol_accidents / total_accidents) * 100
print(f"Percentage of accidents involving alcohol:
{alcohol_percentage:.2f}%")
```

Percentage of accidents involving alcohol: 15.90%

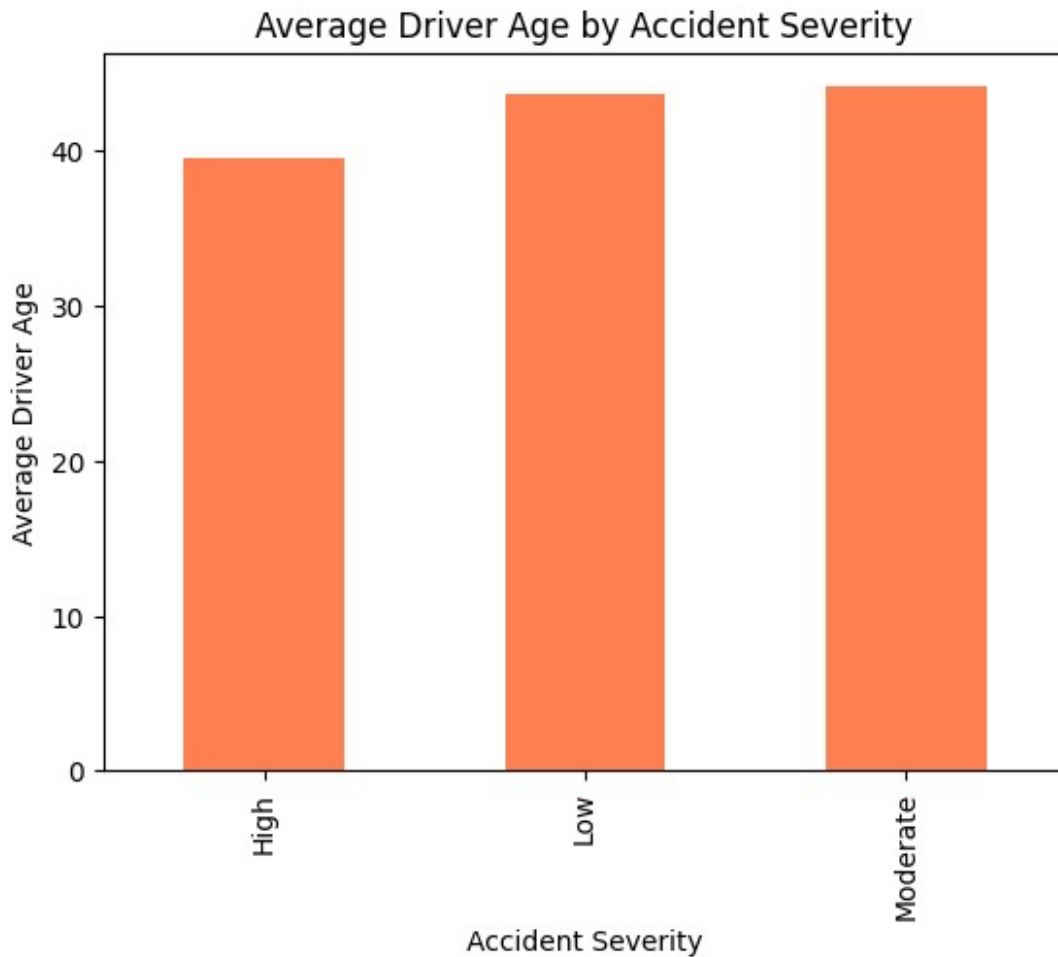
```
# 8. Relationship between number of vehicles and accidents
accident_counts = data.groupby('Number_of_Vehicles')['Accident'].sum()
accident_counts.plot(kind='bar', color='purple')
plt.title("Number of Vehicles vs. Accidents")
plt.xlabel("Number of Vehicles")
plt.ylabel("Number of Accidents")
plt.show()
```



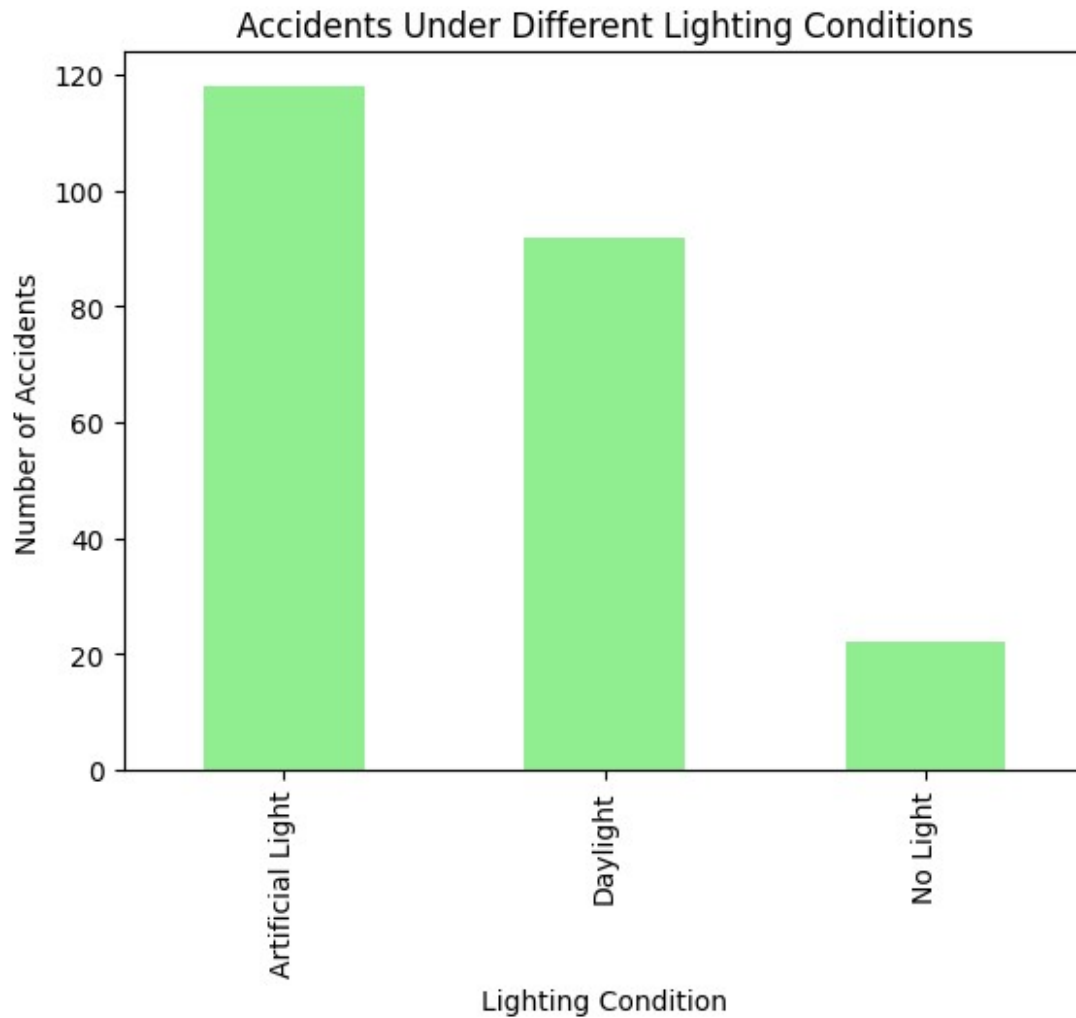
```
# 9. Accident counts between rural and urban roads
rural_vs_urban = data.groupby('Road_Type')['Accident'].sum()
rural_vs_urban.plot(kind='bar', color='teal')
plt.title("Accident Counts by Road Type")
plt.xlabel("Road Type")
plt.ylabel("Number of Accidents")
plt.show()
```



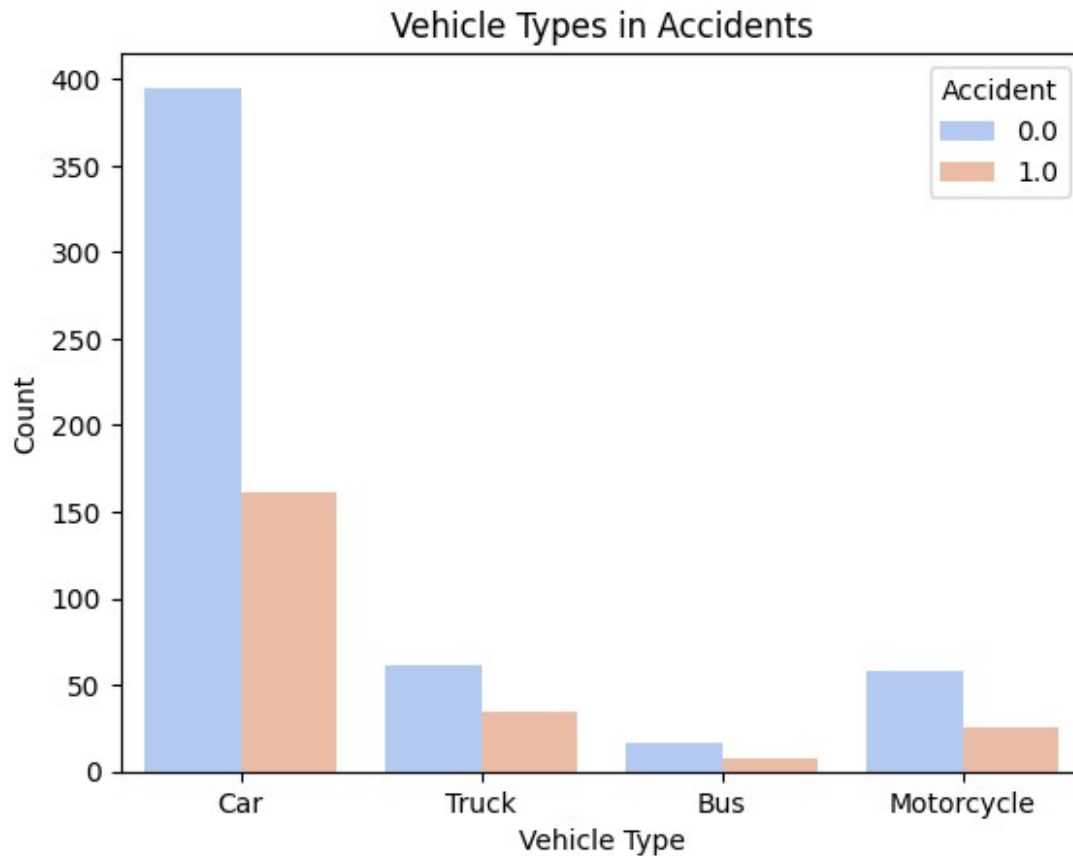
```
# 10. Average driver age by accident severity
avg_age_severity = data.groupby('Accident_Severity')
['Driver_Age'].mean()
avg_age_severity.plot(kind='bar', color='coral')
plt.title("Average Driver Age by Accident Severity")
plt.xlabel("Accident Severity")
plt.ylabel("Average Driver Age")
plt.show()
```



```
# 11. Accidents under different lighting conditions
lighting_accidents = data.groupby('Road_Light_Condition')
['Accident'].sum()
lighting_accidents.plot(kind='bar', color='lightgreen')
plt.title("Accidents Under Different Lighting Conditions")
plt.xlabel("Lighting Condition")
plt.ylabel("Number of Accidents")
plt.show()
```

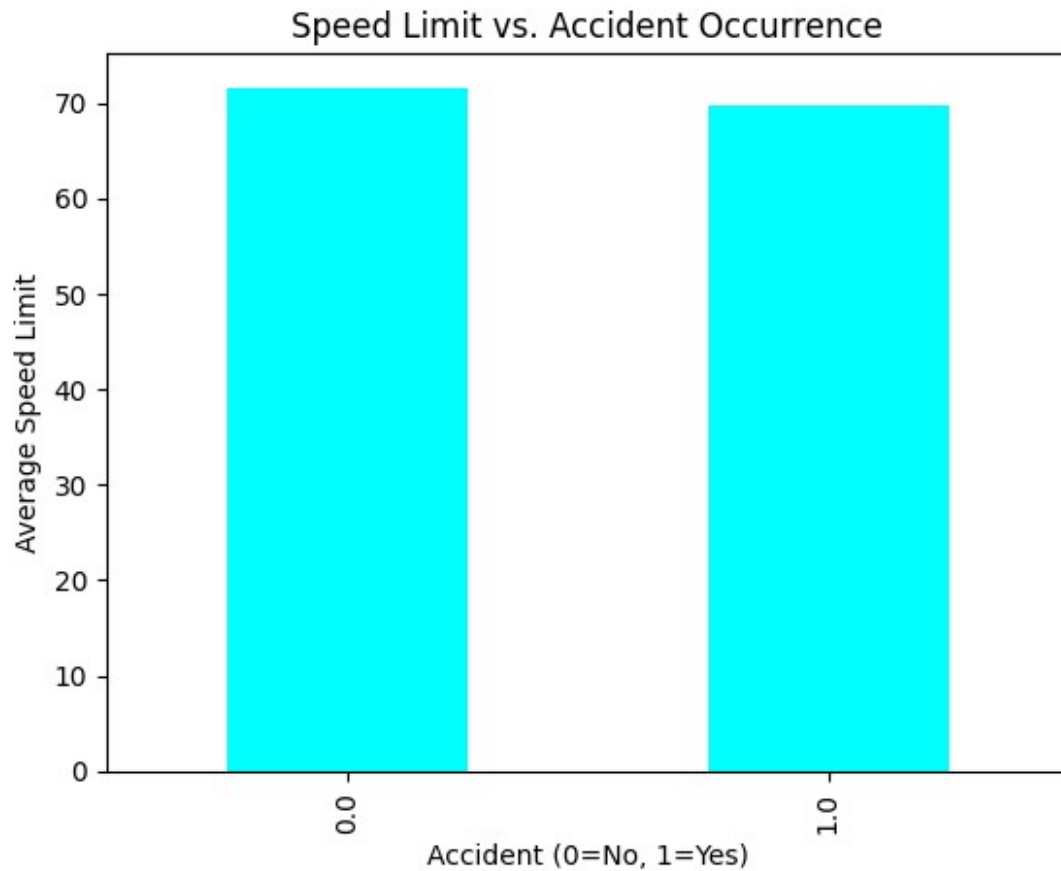


```
# 12. Distribution of vehicle types involved in accidents
sns.countplot(x='Vehicle_Type', hue='Accident', data=data,
palette='coolwarm')
plt.title("Vehicle Types in Accidents")
plt.xlabel("Vehicle Type")
plt.ylabel("Count")
plt.show()
```

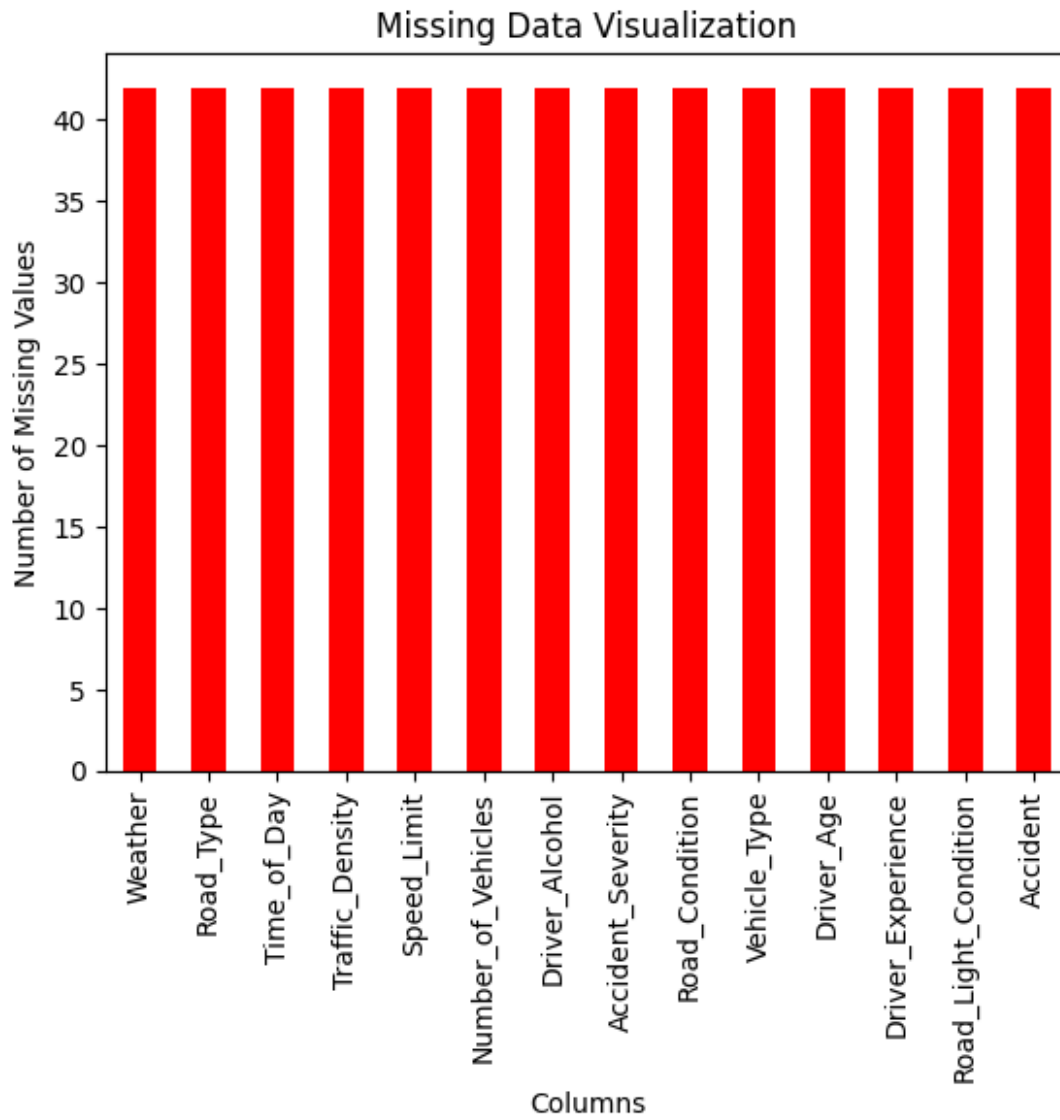



13. Impact of speed limit on accident occurrence

```
speed_limit_impact = data.groupby('Accident')['Speed_Limit'].mean()
speed_limit_impact.plot(kind='bar', color='cyan')
plt.title("Speed Limit vs. Accident Occurrence")
plt.xlabel("Accident (0=No, 1=Yes)")
plt.ylabel("Average Speed Limit")
plt.show()
```



```
# 14. Visualize missing data
missing_data = data.isnull().sum()
missing_data.plot(kind='bar', color='red')
plt.title("Missing Data Visualization")
plt.xlabel("Columns")
plt.ylabel("Number of Missing Values")
plt.show()
```



```
# 15. Trend of accident occurrences across different traffic densities
sns.lineplot(x='Traffic_Density', y='Accident', data=data,
color='blue')
plt.title("Traffic Density vs. Accidents")
plt.xlabel("Traffic Density")
plt.ylabel("Accident (0=No, 1=Yes)")
plt.show()
```

