

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
# If using CSV file, uncomment below:
```

```
# df = pd.read_csv("accident_data.csv")
```

```
data = {
```

```
    "AccidentDate": ["04-02-2013", "23-02-2005", "08-10-2014", "14-01-2015", "17-01-2006"],
```

```
    "Timing": ["Morning", "Night", "Afternoon", "Night", "Afternoon"],
```

```
    "State": ["Alaska", "Arizona", "California", "Colorado", "Georgia"],
```

```
    "WeatherCondition": ["Rainy", "Clear", "Clear", "Rainy", "Clear"],
```

```
    "RoadCondition": ["Under Construction", "Under Construction", "Fine", "Rough", "Fine"],
```

```
    "Deaths": [10, 3, 6, 8, 2],
```

```
    "Reason": ["Drunk Driving", "Weather Conditions", "Poor Visibility", "Road Conditions",  
    "Speeding"]
```

```
}
```

```
df = pd.DataFrame(data)
```

```
df["AccidentDate"] = pd.to_datetime(df["AccidentDate"], dayfirst=True)
```

```
df["Year"] = df["AccidentDate"].dt.year
```

```
df["Month"] = df["AccidentDate"].dt.month
```

```
df["Day"] = df["AccidentDate"].dt.day
```

```
df["Quarter"] = df["AccidentDate"].dt.quarter
```

```
def severity_level(deaths):
```

```
if deaths <= 3:
    return "Low"
elif deaths <= 7:
    return "Medium"
else:
    return "High"
```

```
df["Severity"] = df["Deaths"].apply(severity_level)
```

```
sns.set(style="whitegrid")
plt.figure(figsize=(20, 15))
```

```
plt.subplot(3, 2, 1)
state_counts = df["State"].value_counts()
sns.barplot(x=state_counts.index, y=state_counts.values)
plt.title("State by Accident Count")
plt.xticks(rotation=45)
```

```
plt.subplot(3, 2, 2)
daily_counts = df["Day"].value_counts().sort_index()
plt.plot(daily_counts.index, daily_counts.values, marker="o")
plt.title("Daily Accident Trend")
plt.xlabel("Day")
plt.ylabel("Count")
```

```
plt.subplot(3, 2, 3)
monthly_counts = df["Month"].value_counts().sort_index()
```

```
plt.plot(monthly_counts.index, monthly_counts.values, marker="o")  
plt.title("Monthly Accident Trend")  
plt.xlabel("Month")
```

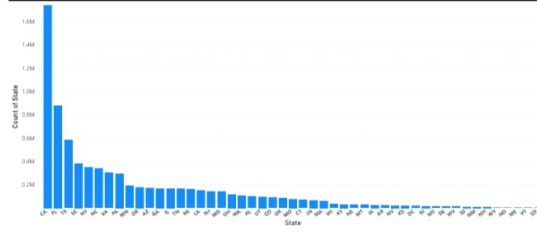
```
plt.subplot(3, 2, 4)  
quarterly_counts = df["Quarter"].value_counts().sort_index()  
plt.plot(quarterly_counts.index, quarterly_counts.values, marker="o")  
plt.title("Quarterly Accident Trend")  
plt.xlabel("Quarter")
```

```
plt.subplot(3, 2, 5)  
weather_counts = df["WeatherCondition"].value_counts()  
plt.pie(weather_counts.values, labels=weather_counts.index, autopct='%1.1f%%')  
centre_circle = plt.Circle((0,0),0.70,fc='white')  
plt.gca().add_artist(centre_circle)  
plt.title("Accident by Weather")
```

```
plt.subplot(3, 2, 6)  
severity_counts = df["Severity"].value_counts()  
plt.pie(severity_counts.values, labels=severity_counts.index, autopct='%1.1f%%')  
centre_circle = plt.Circle((0,0),0.70,fc='white')  
plt.gca().add_artist(centre_circle)  
plt.title("Accident by Severity")
```

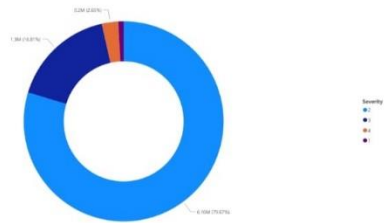
```
plt.tight_layout()  
plt.show()
```

STATE BY ACCIDENT COUNT IN UNITED STATES



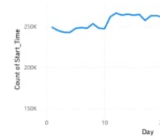
1/5

accident by severity

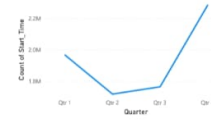


2/5

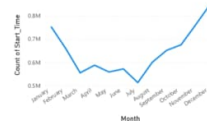
DAILY ACCIDENT TREND



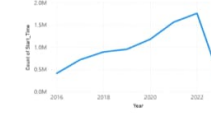
QUARTELY ACCIDENT TREND



MONTHLY ACCIDENT TREND

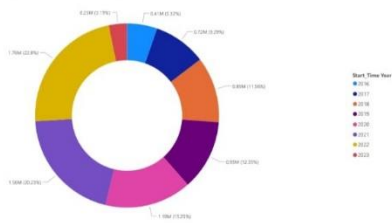


YEARLY ACCIDENT TREND



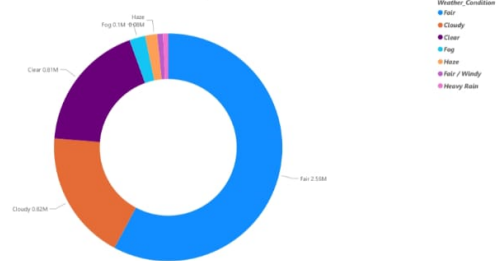
2/5

yearly report of accident



4/5

accident by weather



3/5

accident by severity