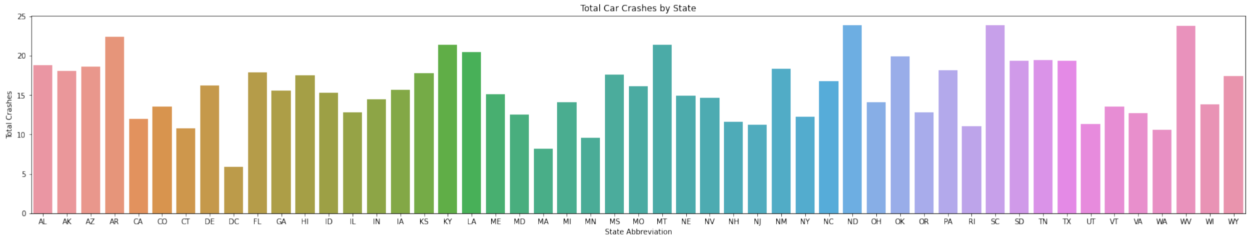
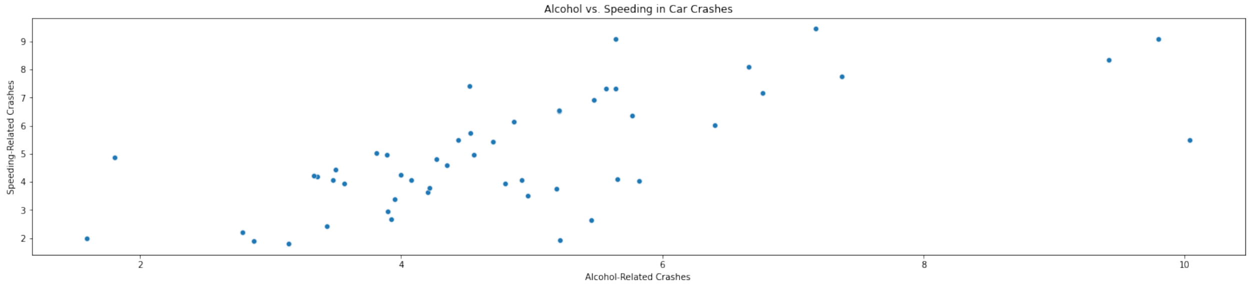
```
pip install seaborn pandas
```

```
Requirement already satisfied: seaborn in c:\users\kanik\anaconda3\
lib\site-packages (0.11.2)Note: you may need to restart the kernel to
use updated packages.
Requirement already satisfied: pandas in c:\users\kanik\anaconda3\lib\
site-packages (1.3.4)
Requirement already satisfied: numpy>=1.15 in c:\users\kanik\
anaconda3\lib\site-packages (from seaborn) (1.20.3)
Requirement already satisfied: matplotlib>=2.2 in c:\users\kanik\
anaconda3\lib\site-packages (from seaborn) (3.4.3)
Requirement already satisfied: scipy>=1.0 in c:\users\kanik\anaconda3\
lib\site-packages (from seaborn) (1.7.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\kanik\
anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\
kanik\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\kanik\
anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (8.4.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\kanik\
anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\kanik\
anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\kanik\
anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: six in c:\users\kanik\anaconda3\lib\
site-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.16.0)
```

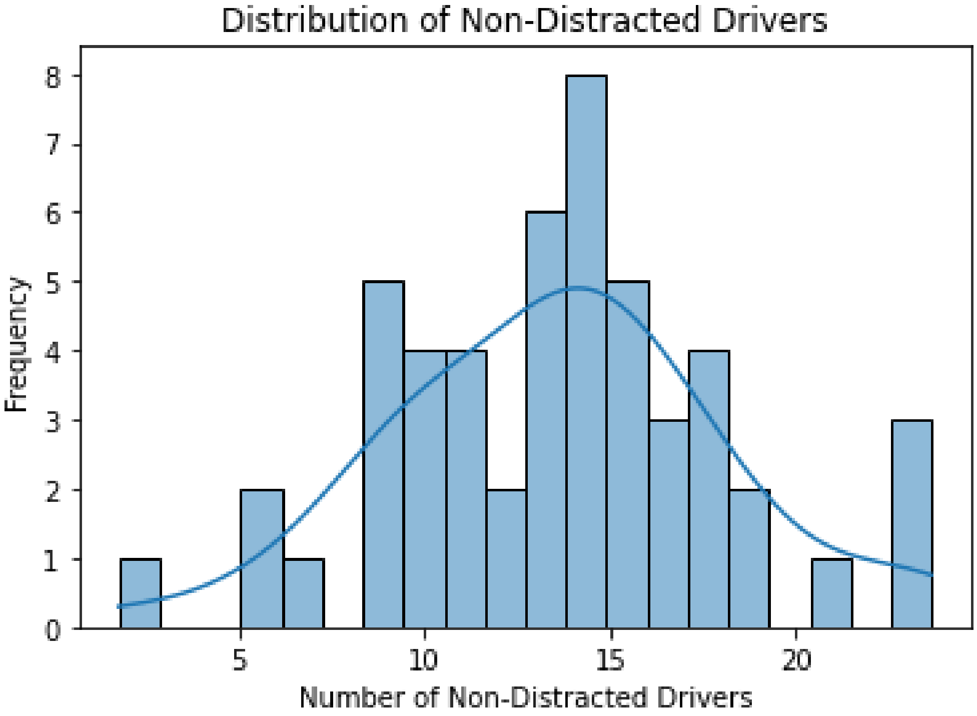
```python
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

car_crashes = sns.load_dataset("car_crashes")

#Bar Plot
plt.figure(figsize=(30,5))
sns.barplot(x='abbrev', y='total', data=car_crashes)
plt.title('Total Car Crashes by State')
plt.xlabel('State Abbreviation')
plt.ylabel('Total Crashes')
plt.show()
#Inference: This bar plot shows the total car crashes in each state.
State 'ND' has the highest number of car crashes, while 'DC' has the
lowest.
```



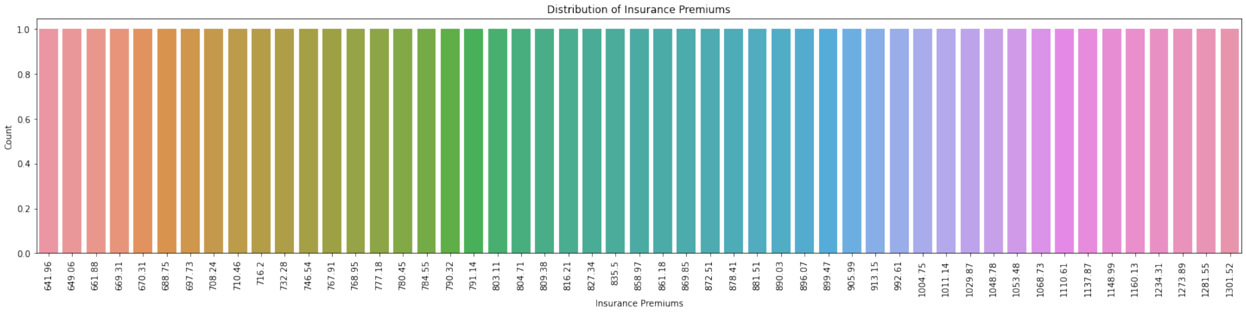
```
#Scatter Plot
plt.figure(figsize=(25,5))
sns.scatterplot(x='alcohol', y='speeding', data=car_crashes)
plt.title('Alcohol vs. Speeding in Car Crashes')
plt.xlabel('Alcohol-Related Crashes')
plt.ylabel('Speeding-Related Crashes')
plt.show()
#Inference: This scatter plot shows the relationship between alcohol-
related car crashes and speeding-related car crashes. It appears that
there is a positive correlation between the two variables.
```



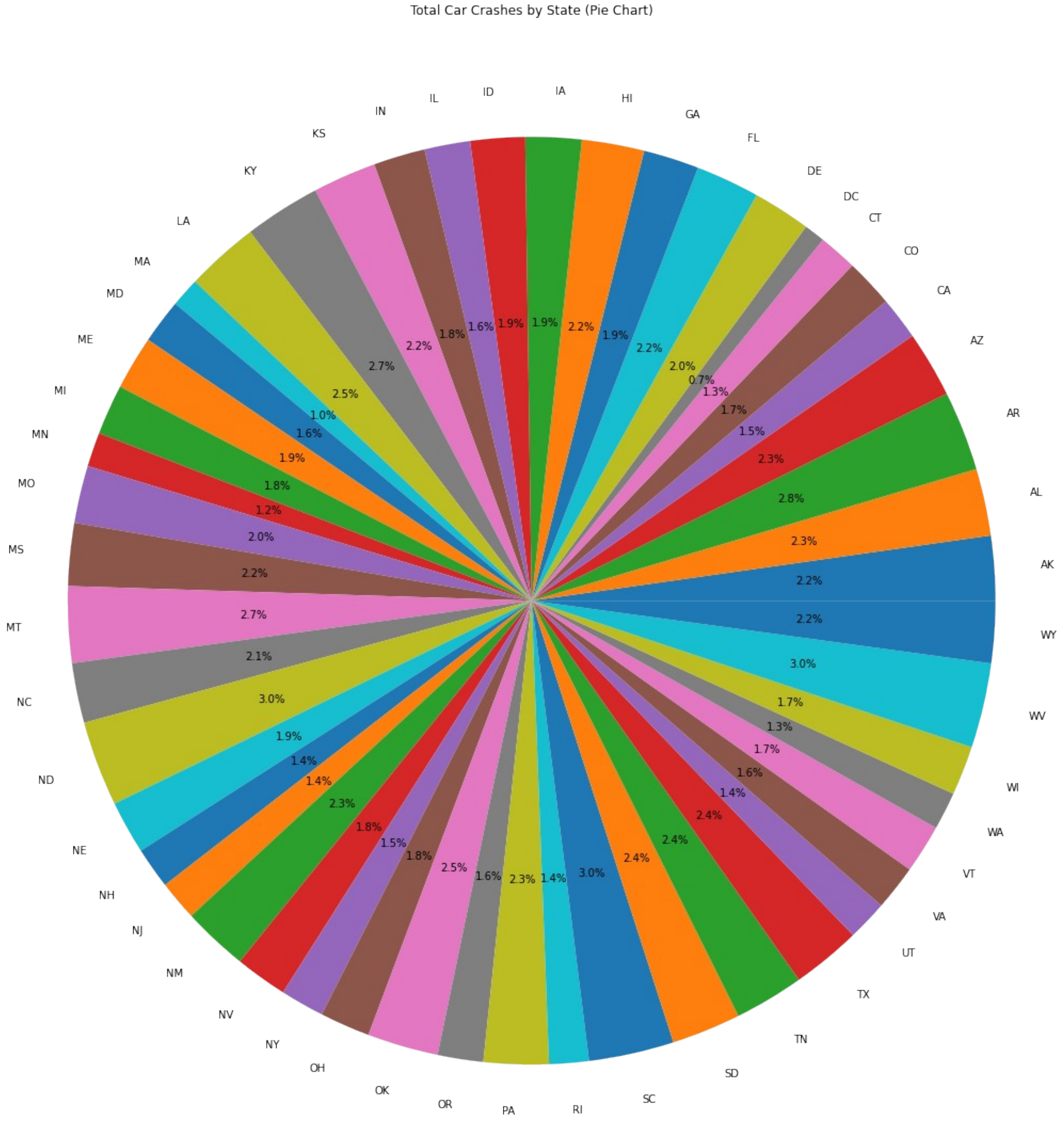
```
#Histogram
sns.histplot(car_crashes['not_distracted'], bins=20, kde=True)
plt.title('Distribution of Non-Distracted Drivers')
plt.xlabel('Number of Non-Distracted Drivers')
plt.ylabel('Frequency')
plt.show()
#Inference: This histogram displays the distribution of non-distracted
drivers involved in car crashes. The majority of car crashes involve a
relatively low number of non-distracted drivers.
```



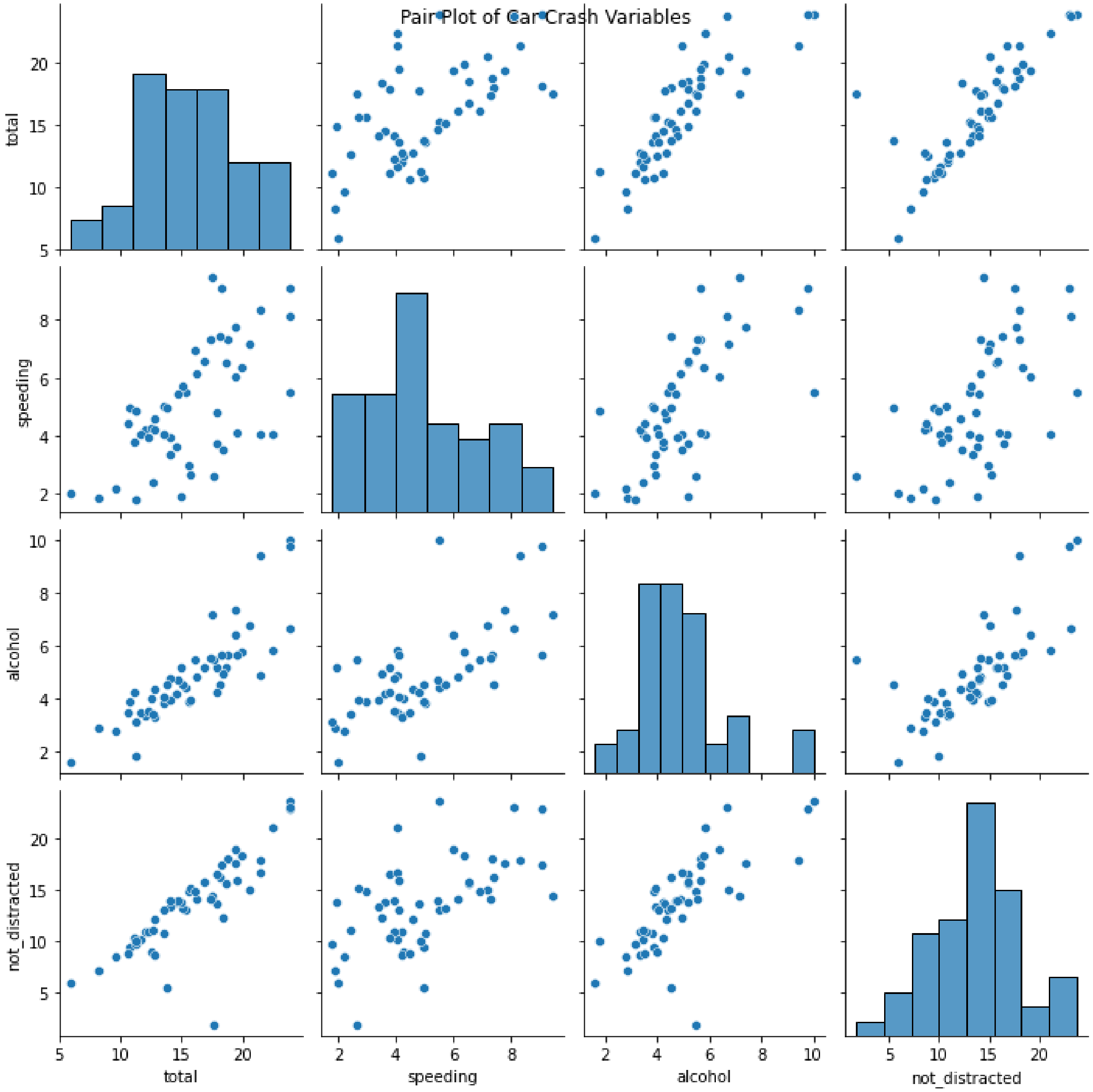
```
#Count Plot
plt.figure(figsize=(25,5))
sns.countplot(x='ins_premium', data=car_crashes)
plt.title('Distribution of Insurance Premiums')
plt.xlabel('Insurance Premiums')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
#Inference: The count plot shows the distribution of insurance
premiums. It indicates the frequency of each premium level in the
dataset.
```



```
#Pie Chart
plt.figure(figsize=(20,20))
total_by_state = car_crashes.groupby('abbrev')['total'].sum()
plt.pie(total_by_state, labels=total_by_state.index, autopct='%1.1f%
%')
plt.title('Total Car Crashes by State (Pie Chart)')
plt.show()
#Inference: The pie chart illustrates the distribution of total car
crashes by state as a percentage of the whole. It helps visualize the
contribution of each state to the overall total.
```

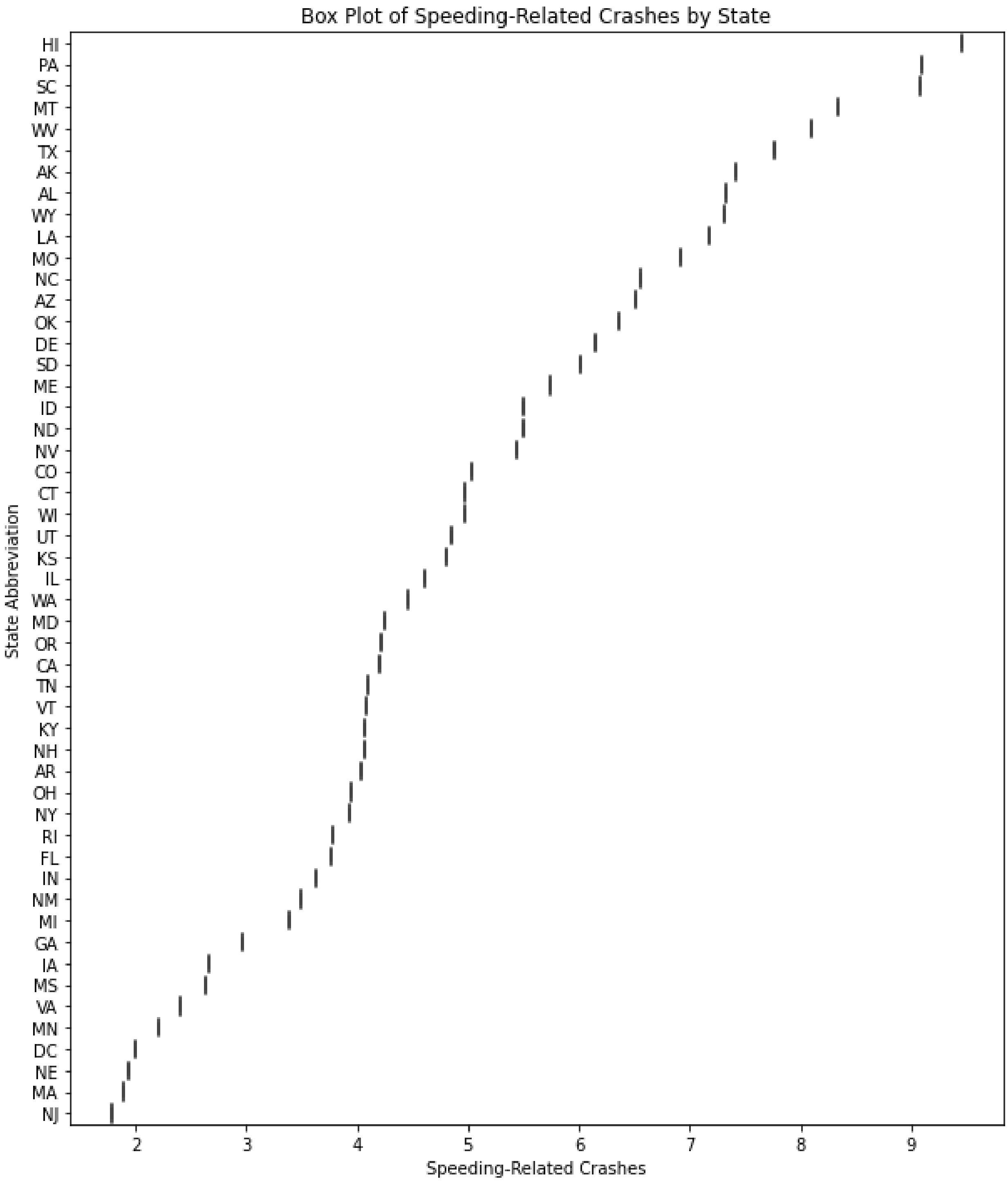


```
#Pair Plot
sns.pairplot(car_crashes[['total', 'speeding', 'alcohol',
'not_distracted']])
plt.suptitle('Pair Plot of Car Crash Variables')
plt.show()
#Inference: This pair plot allows you to visualize the relationships
between multiple variables at once. It shows scatterplots for 'total,'
'speeding,' 'alcohol,' and 'not_distracted' variables, as well as
histograms for each variable along the diagonal. It can help identify
potential correlations and patterns.
```

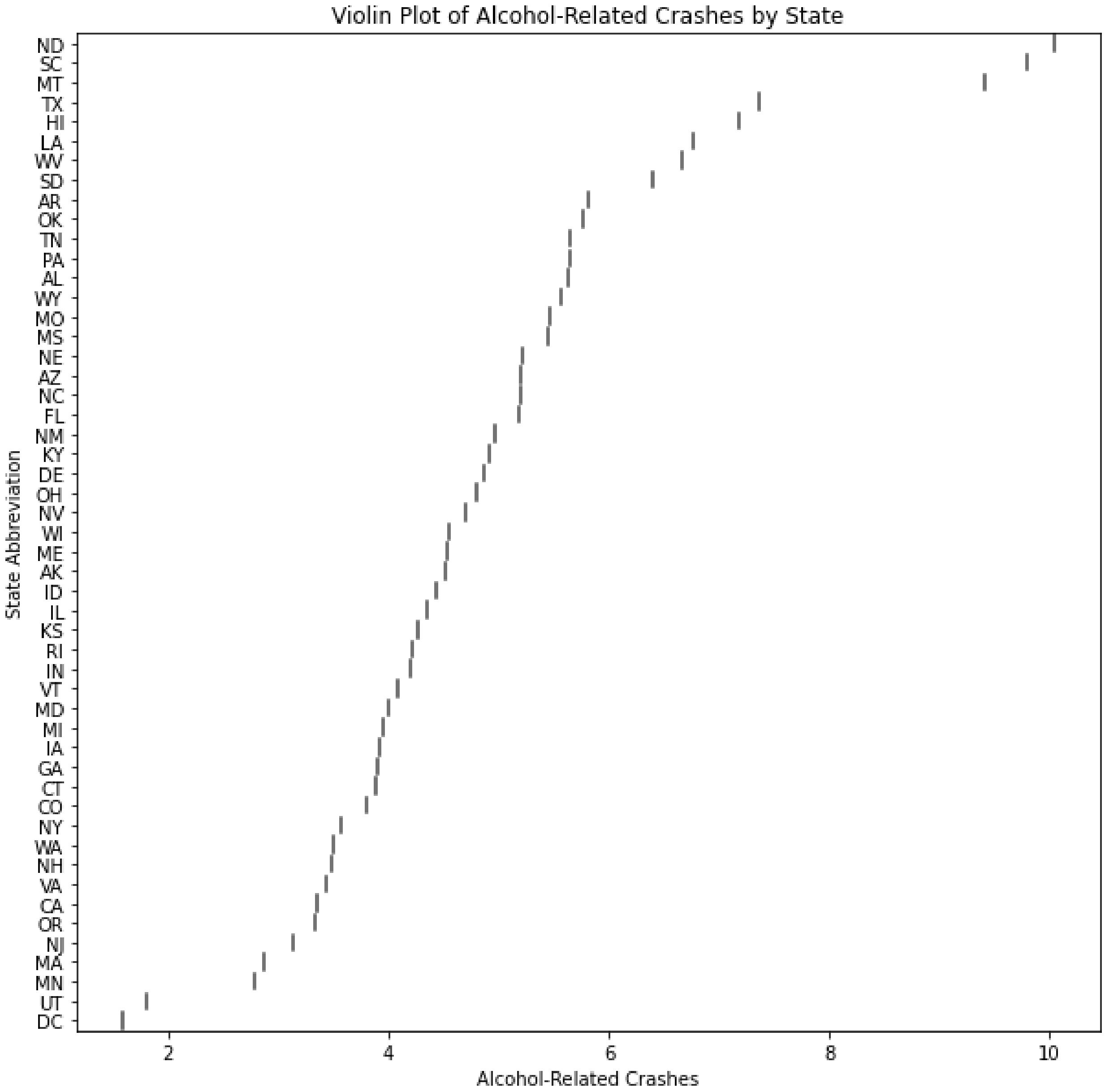


```
#Box Plot
plt.figure(figsize=(10,12))
sns.boxplot(x='speeding', y='abbrev',
data=car_crashes.sort_values('speeding', ascending=False))
```

```
plt.title('Box Plot of Speeding-Related Crashes by State')
plt.xlabel('Speeding-Related Crashes')
plt.ylabel('State Abbreviation')
plt.show()
#Inference: This box plot shows the distribution of speeding-related
crashes across different states. It provides information about the
median, quartiles, and potential outliers in the data.
```

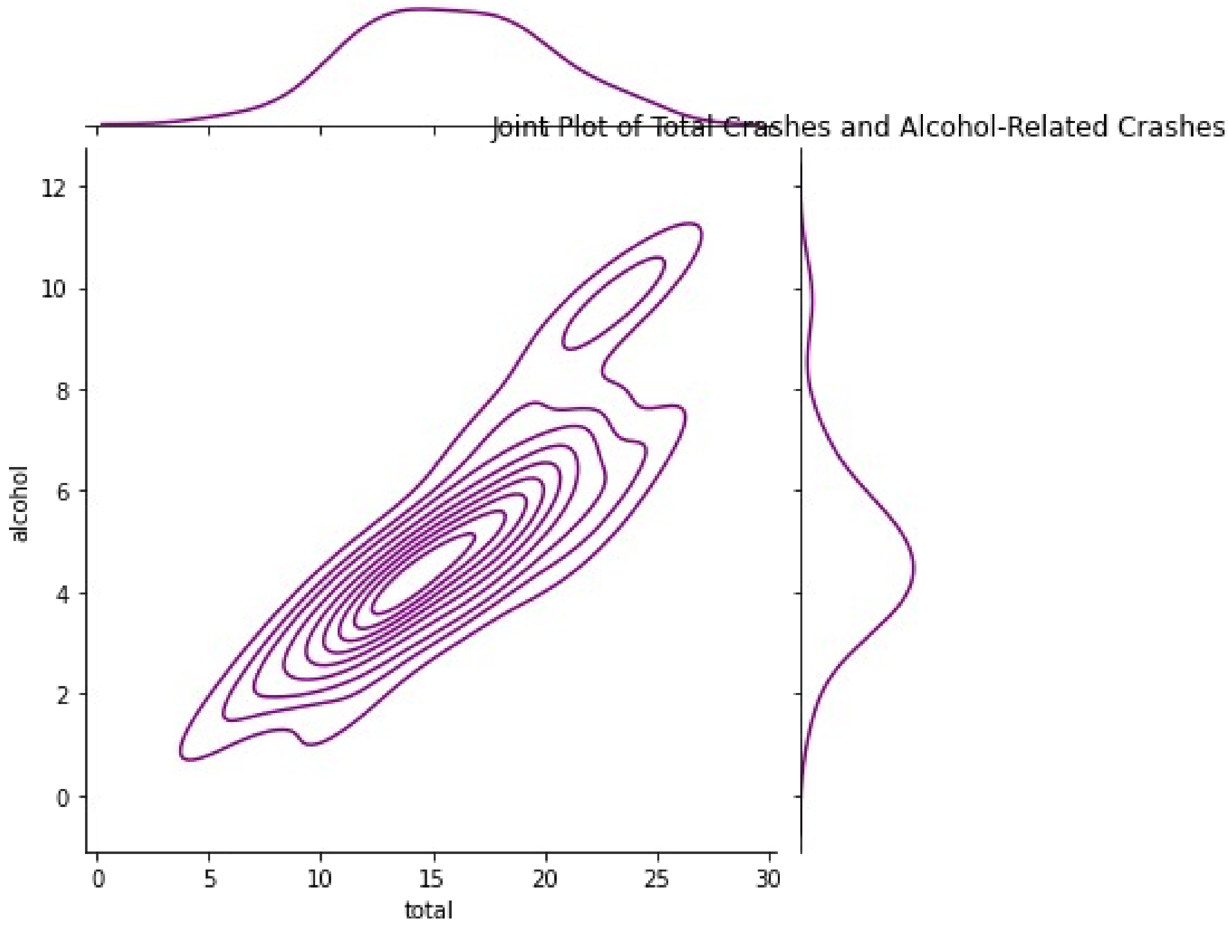


```
#Violin Plot
plt.figure(figsize=(10, 10))
sns.violinplot(x='alcohol', y='abbrev',
data=car_crashes.sort_values('alcohol', ascending=False))
plt.title('Violin Plot of Alcohol-Related Crashes by State')
plt.xlabel('Alcohol-Related Crashes')
plt.ylabel('State Abbreviation')
plt.show()
#Inference: The violin plot displays the distribution of alcohol-
related crashes across states. It combines a box plot with a kernel
density estimation, providing insights into both the central tendency
and spread of the data.
```

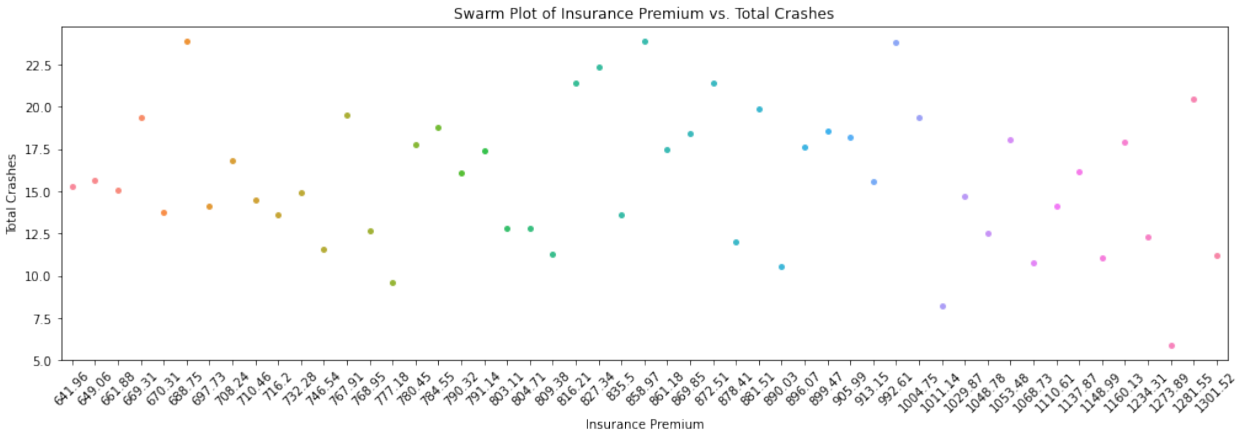

Violin Plot of Alcohol-Related Crashes by State

```
#Joint Plot
plt.figure(figsize=(15,15))
sns.jointplot(x='total', y='alcohol', data=car_crashes, kind='kde',
color='purple')
plt.title('Joint Plot of Total Crashes and Alcohol-Related Crashes')
plt.show()
#Inference: This joint plot displays a kernel density estimate (KDE)
of the relationship between 'total' crashes and 'alcohol' related
crashes. It shows the joint distribution and the marginal
distributions of both variables.
```

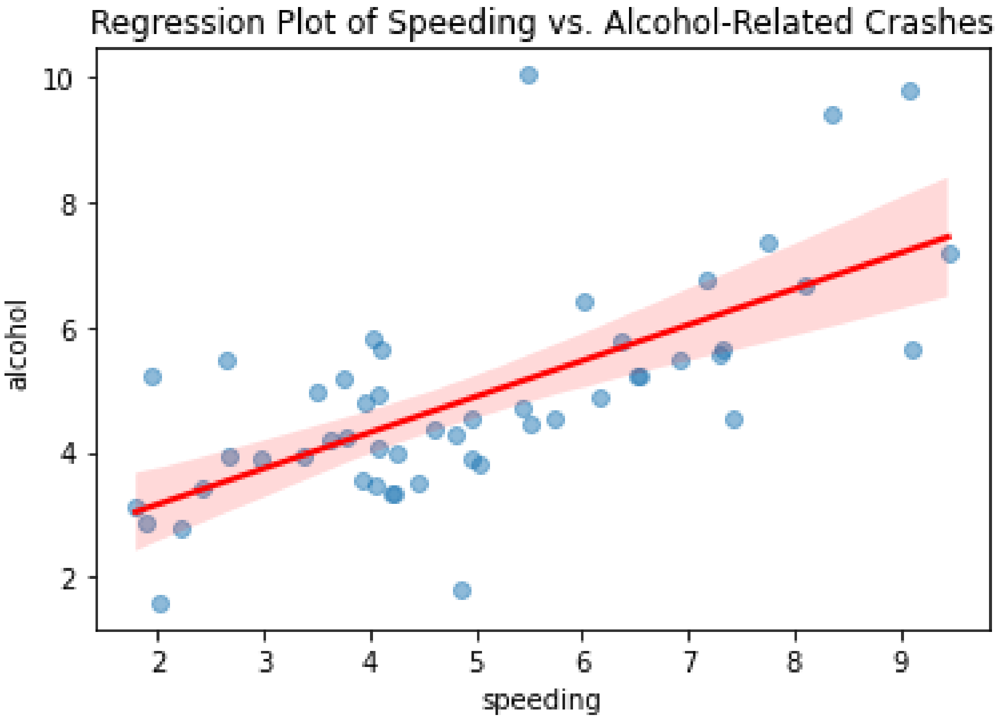
```
<Figure size 1080x1080 with 0 Axes>
```



```
#Swarm Plot
plt.figure(figsize=(17,5))
sns.swarmplot(x='ins_premium', y='total', data=car_crashes)
plt.title('Swarm Plot of Insurance Premium vs. Total Crashes')
plt.xlabel('Insurance Premium')
plt.ylabel('Total Crashes')
plt.xticks(rotation=45)
plt.show()
#Inference: The swarm plot shows how 'total' crashes vary across
different insurance premium categories. It provides a visual
representation of the distribution and density of data points.
```

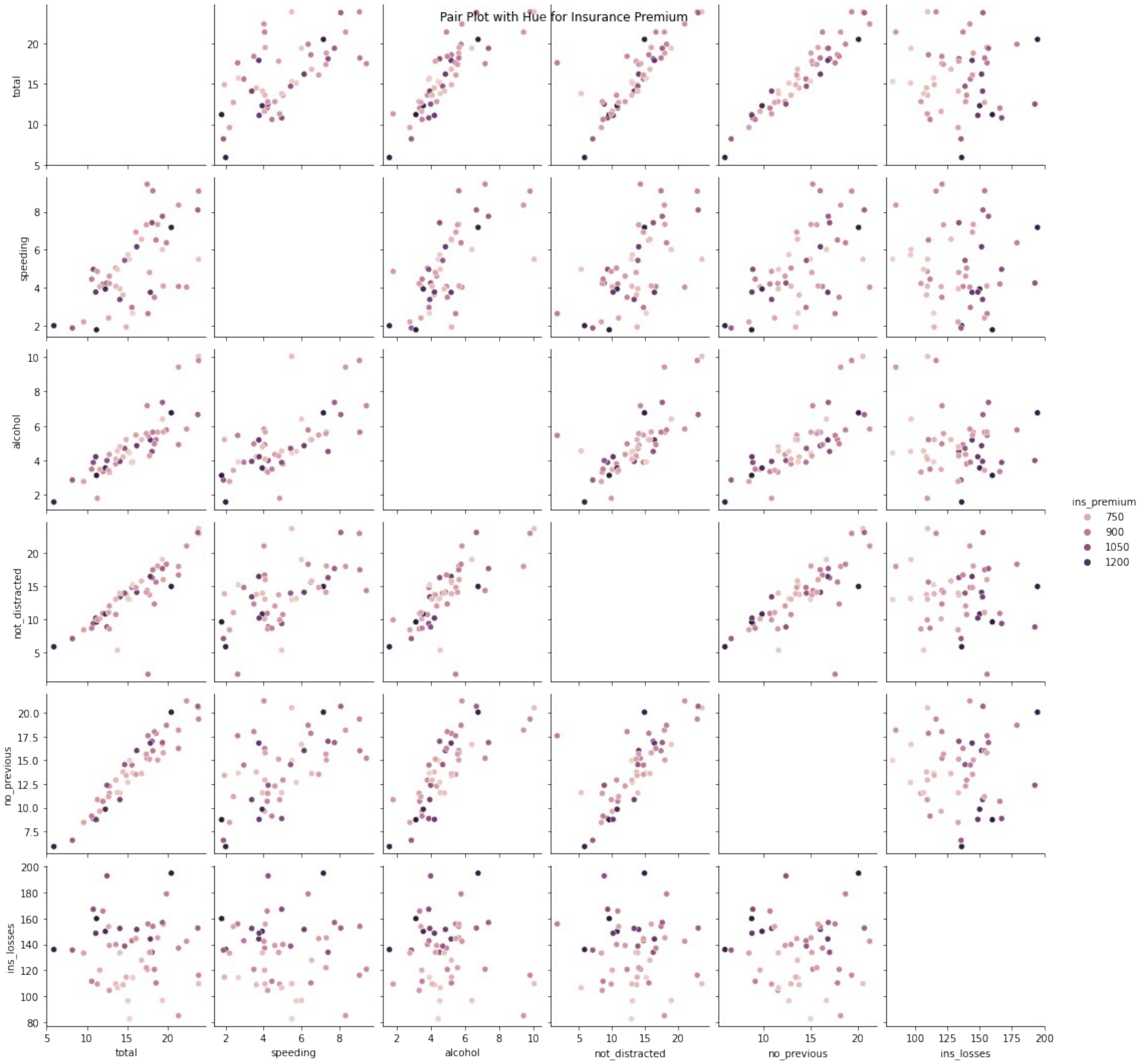


```
#Reg Plot (Regression Plot)
sns.regplot(x='speeding', y='alcohol', data=car_crashes,
scatter_kws={'alpha':0.5}, line_kws={'color':'red'})
plt.title('Regression Plot of Speeding vs. Alcohol-Related Crashes')
plt.show()
#Inference: This regression plot displays a linear regression line
that shows the relationship between 'speeding' and 'alcohol' related
crashes. The red line represents the linear fit to the data points.
```



```
#Pair Plot with Hue
sns.pairplot(car_crashes, hue='ins_premium')
plt.suptitle('Pair Plot with Hue for Insurance Premium')
plt.show()
```

```
#Inference: This pair plot adds a hue based on insurance premium
values. It allows you to visualize how different car crash variables
relate to each other while considering the insurance premium category.
It can help identify patterns specific to premium categories
```


Pair Plot with Hue for Insurance Premium
total
speeding
alcohol
not_distracted
no_previous
ins_losses
ins_premium