ASSIGNMENT-02

PERECHARLA BHAVANA

ROLL NO:21BCE9184

VITAP

bhavanaperecharla@gmail.com

bhavana.21bce9184@vitapstudent.ac.in

```
import seaborn as sns
import matplotlib.pyplot as plt
```

This code imports Seaborn, loads the 'car_crashes' dataset, and assigns it to the variable 'car_crashes'

```
# Load the car_crashes dataset
car_crashes = sns.load_dataset("car_crashes")
car_crashes
```

```
    total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0    18.8     7.332    5.640          18.048       15.040       784.55

1    18.1     7.421    4.525          16.290       17.014      1053.48

2    18.6     6.510    5.208          15.624       17.856       899.47

3    22.4     4.032    5.824          21.056       21.280       827.34

4    12.0     4.200    3.360          10.920       10.680       878.41

5    13.6     5.032    3.808          10.744       12.920       835.50

6    10.8     4.968    3.888           9.396        8.856      1068.73

7    16.2     6.156    4.860          14.094       16.038      1137.87

8     5.9     2.006    1.593           5.900        5.900      1273.89

9    17.9     3.759    5.191          16.468       16.826      1160.13

10   15.6     2.964    3.900          14.820       14.508       913.15

11   17.5     9.450    7.175          14.350       15.225       861.18

12   15.3     5.508    4.437          13.005       14.994       641.96

13   12.8     4.608    4.352          12.032       12.288       803.11
```

14	14.5	3.625	4.205	13.775	13.775	710.46
15	15.7	2.669	3.925	15.229	13.659	649.06
16	17.8	4.806	4.272	13.706	15.130	780.45
17	21.4	4.066	4.922	16.692	16.264	872.51
18	20.5	7.175	6.765	14.965	20.090	1281.55
19	15.1	5.738	4.530	13.137	12.684	661.88
20	12.5	4.250	4.000	8.875	12.375	1048.78
21	8.2	1.886	2.870	7.134	6.560	1011.14
22	14.1	3.384	3.948	13.395	10.857	1110.61
23	9.6	2.208	2.784	8.448	8.448	777.18
24	17.6	2.640	5.456	1.760	17.600	896.07
25	16.1	6.923	5.474	14.812	13.524	790.32
26	21.4	8.346	9.416	17.976	18.190	816.21
27	14.9	1.937	5.215	13.857	13.410	732.28
28	14.7	5.439	4.704	13.965	14.553	1029.87
29	11.6	4.060	3.480	10.092	9.628	746.54
30	11.2	1.792	3.136	9.632	8.736	1301.52
31	18.4	3.496	4.968	12.328	18.032	869.85
32	12.3	3.936	3.567	10.824	9.840	1234.31
33	16.8	6.552	5.208	15.792	13.608	708.24
34	23.9	5.497	10.038	23.661	20.554	688.75
35	14.1	3.948	4.794	13.959	11.562	697.73
36	19.9	6.368	5.771	18.308	18.706	881.51
37	12.8	4.224	3.328	8.576	11.520	804.71
38	18.2	9.100	5.642	17.472	16.016	905.99
39	11.1	3.774	4.218	10.212	8.769	1148.99

```
40   23.9     9.082    9.799          22.944        19.359        858.97

41   19.4     6.014    6.402          19.012        16.684        669.31

42   19.5     4.095    5.655          15.990        15.795        767.91

43   19.4     7.760    7.372          17.654        16.878       1004.75

44   11.3     4.859    1.808           9.944        10.848        809.38

45   13.6     4.080    4.080          13.056        12.920        716.20

46   12.7     2.413    3.429          11.049        11.176        768.95

47   10.6     4.452    3.498           8.692         9.116        890.03

48   23.8     8.092    6.664          23.086        20.706        992.61

49   13.8     4.968    4.554           5.382        11.592        670.31

50   17.4     7.308    5.568          14.094        15.660        791.14
```

	ins_losses	abbrev
0	145.08	AL
1	133.93	AK
2	110.35	AZ
3	142.39	AR
4	165.63	CA
5	139.91	CO
6	167.02	CT
7	151.48	DE
8	136.05	DC
9	144.18	FL
10	142.80	GA
11	120.92	HI
12	82.75	ID
13	139.15	IL
14	108.92	IN
15	114.47	IA
16	133.80	KS
17	137.13	KY
18	194.78	LA
19	96.57	ME
20	192.70	MD
21	135.63	MA
22	152.26	MI
23	133.35	MN
24	155.77	MS

```
25      144.45     MO
26       85.15     MT
27      114.82     NE
28      138.71     NV
29      120.21     NH
30      159.85     NJ
31      120.75     NM
32      150.01     NY
33      127.82     NC
34      109.72     ND
35      133.52     OH
36      178.86     OK
37      104.61     OR
38      153.86     PA
39      148.58     RI
40      116.29     SC
41       96.87     SD
42      155.57     TN
43      156.83     TX
44      109.48     UT
45      109.61     VT
46      153.72     VA
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY
```

```
car_crashes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   total           51 non-null     float64
 1   speeding        51 non-null     float64
 2   alcohol         51 non-null     float64
 3   not_distracted  51 non-null     float64
 4   no_previous     51 non-null     float64
 5   ins_premium     51 non-null     float64
 6   ins_losses      51 non-null     float64
 7   abbrev          51 non-null     object
dtypes: float64(7), object(1)
memory usage: 3.3+ KB
```

```
car_crashes.head()
```

```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0   18.8     7.332    5.640          18.048       15.040       784.55
```

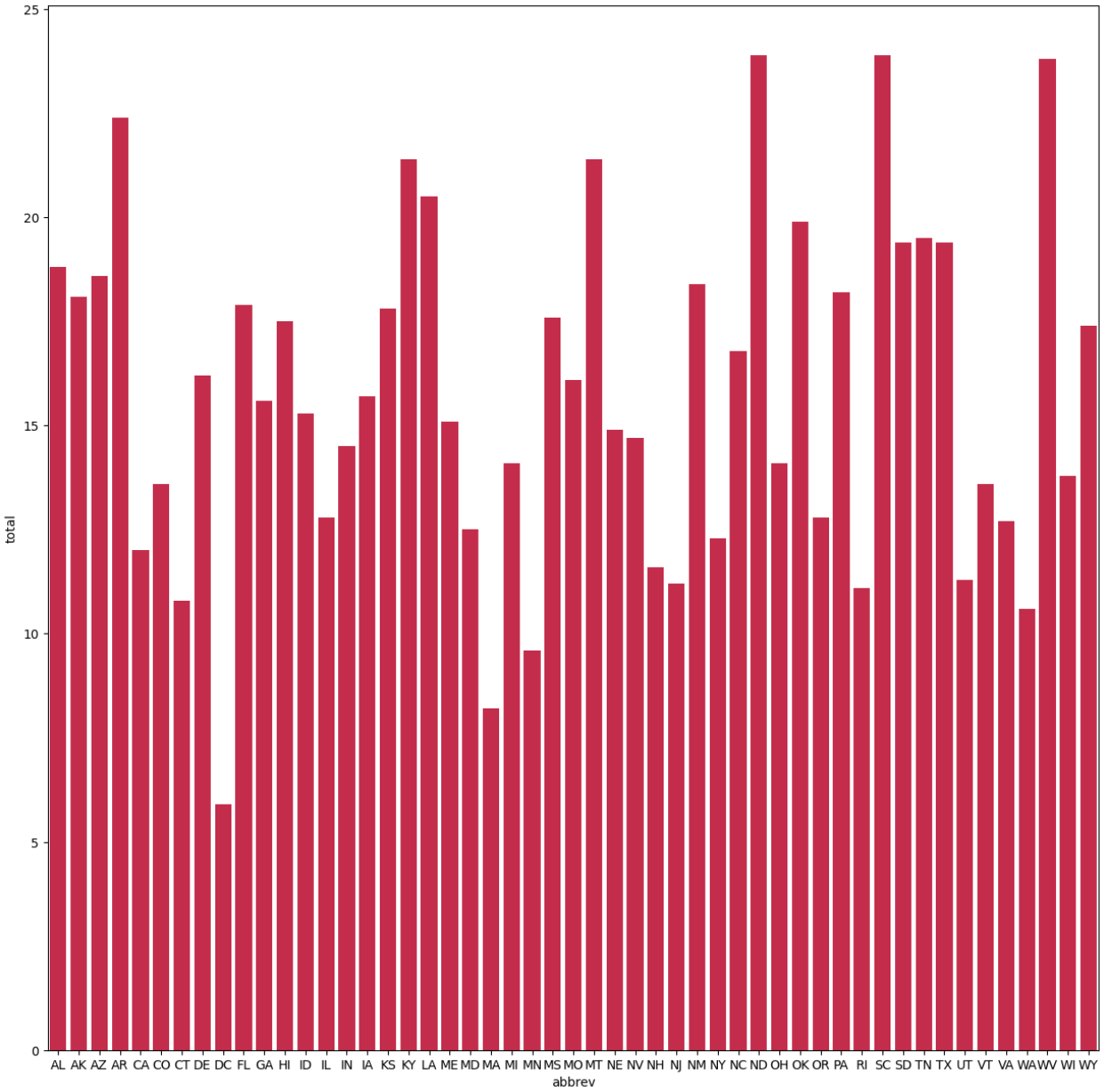
```
1   18.1     7.421    4.525          16.290       17.014       1053.48

2   18.6     6.510    5.208          15.624       17.856        899.47

3   22.4     4.032    5.824          21.056       21.280        827.34

4   12.0     4.200    3.360          10.920       10.680        878.41
```

```
   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
3      142.39     AR
4      165.63     CA
```

Total Number of car crash cases in every state of USA Visualized using a barplot

```
plt.subplots(figsize=(15,15))
sns.barplot(x="abbrev",y="total",data=car_crashes,color='crimson')
```

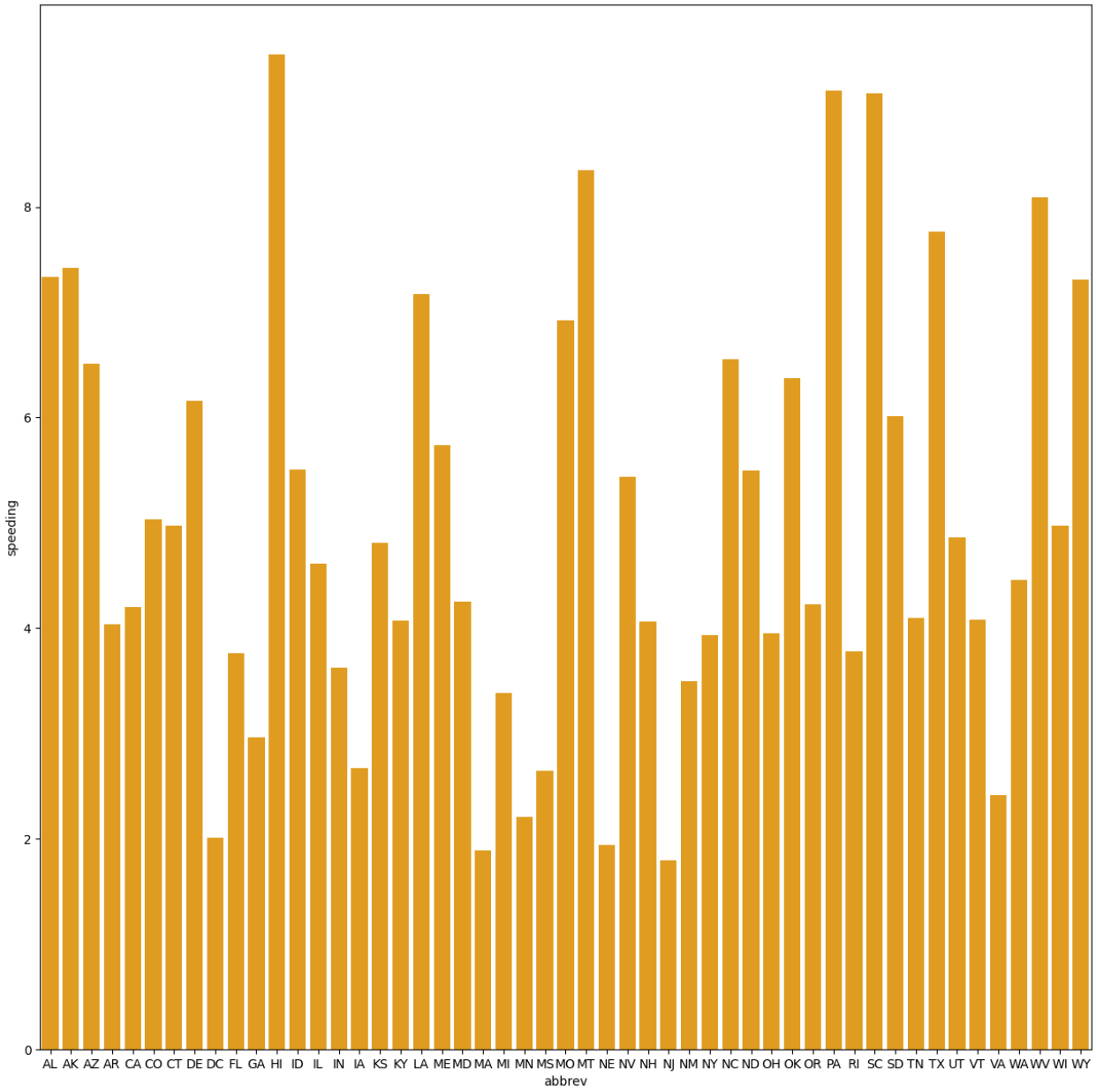
```
<Axes: xlabel='abbrev', ylabel='total'>
```



Car Crashes due to Speeding in every state of USA

```
plt.subplots(figsize=(15,15))
sns.barplot(x="abbrev",y="speeding",data=car_crashes,color='orange')
```

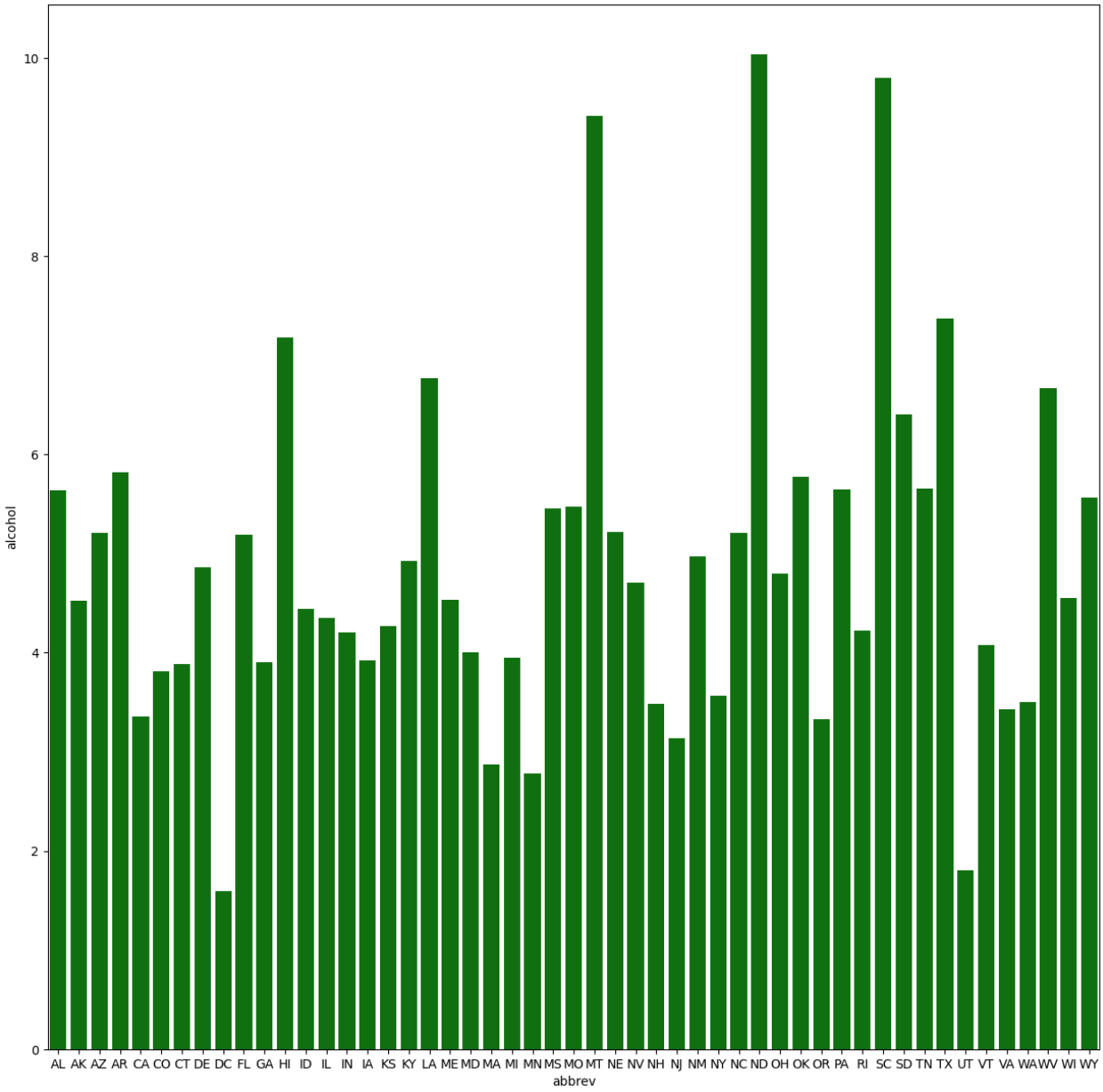
```
<Axes: xlabel='abbrev', ylabel='speeding'>
```



Car Crashes due to alcohol consumption in every state of USA

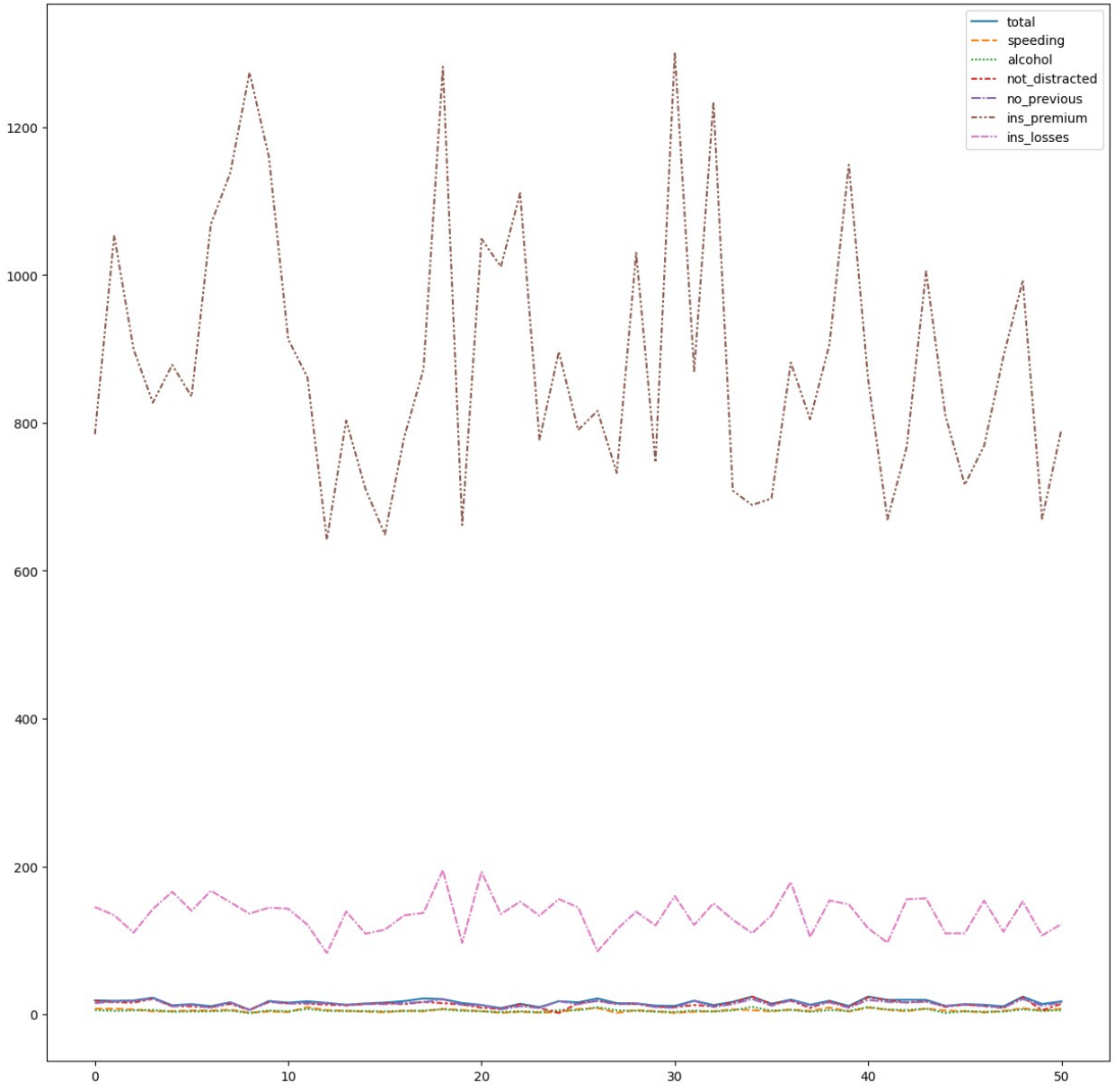
```
plt.subplots(figsize=(15,15))
sns.barplot(x="abbrev",y="alcohol",data=car_crashes,color='GREEN')
```

```
<Axes: xlabel='abbrev', ylabel='alcohol'>
```



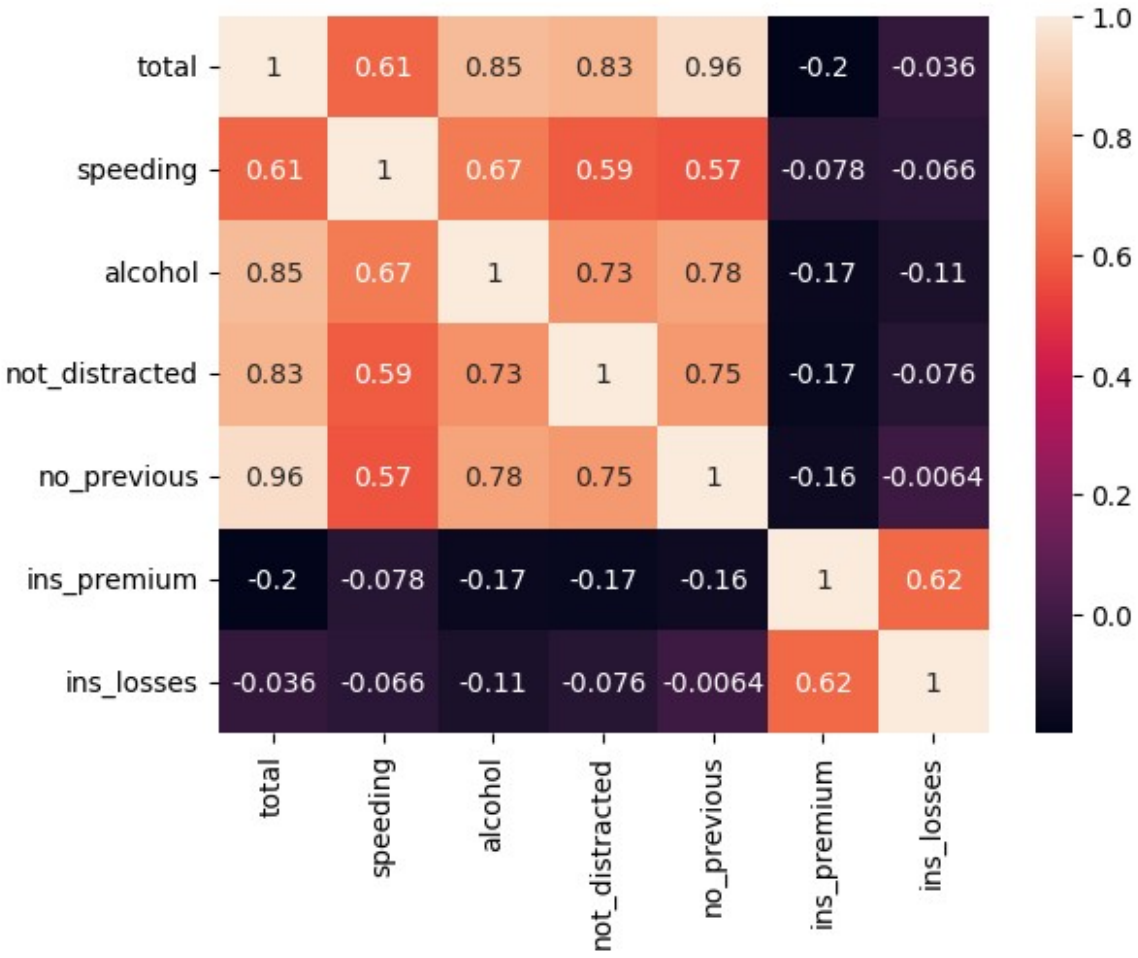

```
plt.subplots(figsize=(15,15))
sns.lineplot(data=car_crashes)
```

```
<Axes: >
```




```
sns.heatmap(corr,annot=True)
```

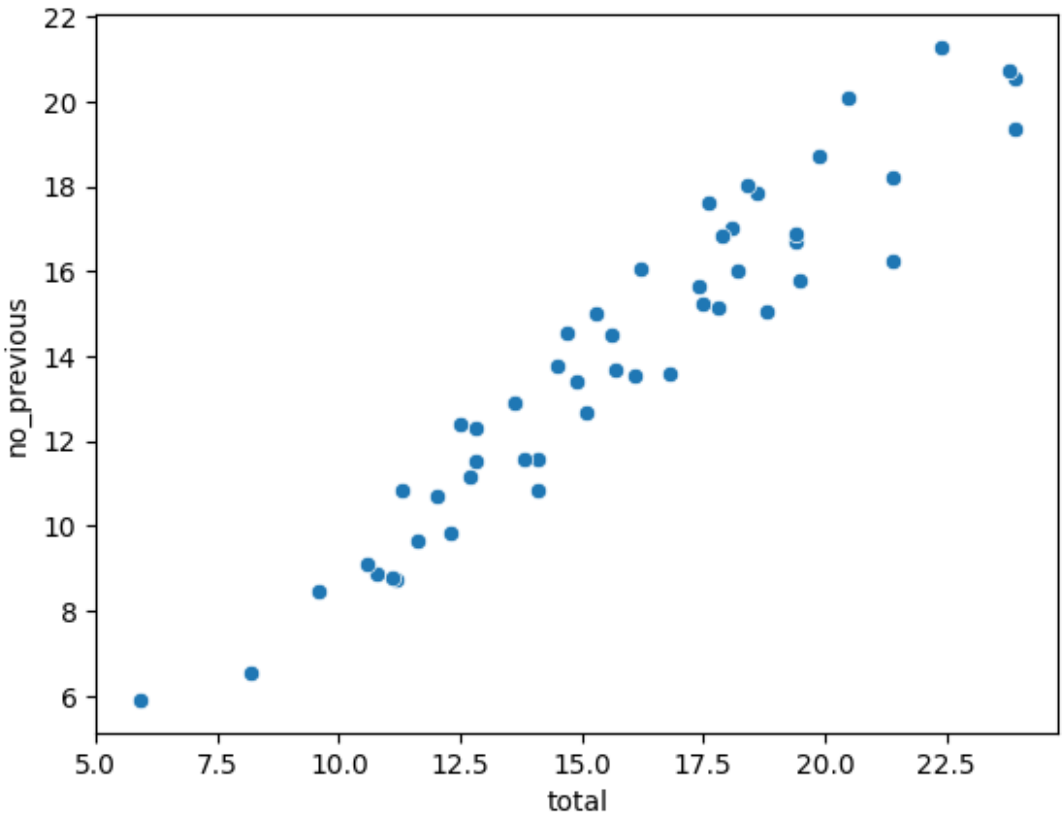
```
<Axes: >
```



Correlation between Total crashes and new crashes Here,we are checking if increase in total number of crashes affects the car crashes with no previous crash record

```
sns.scatterplot(data=car_crashes,y='no_previous',x='total')
```

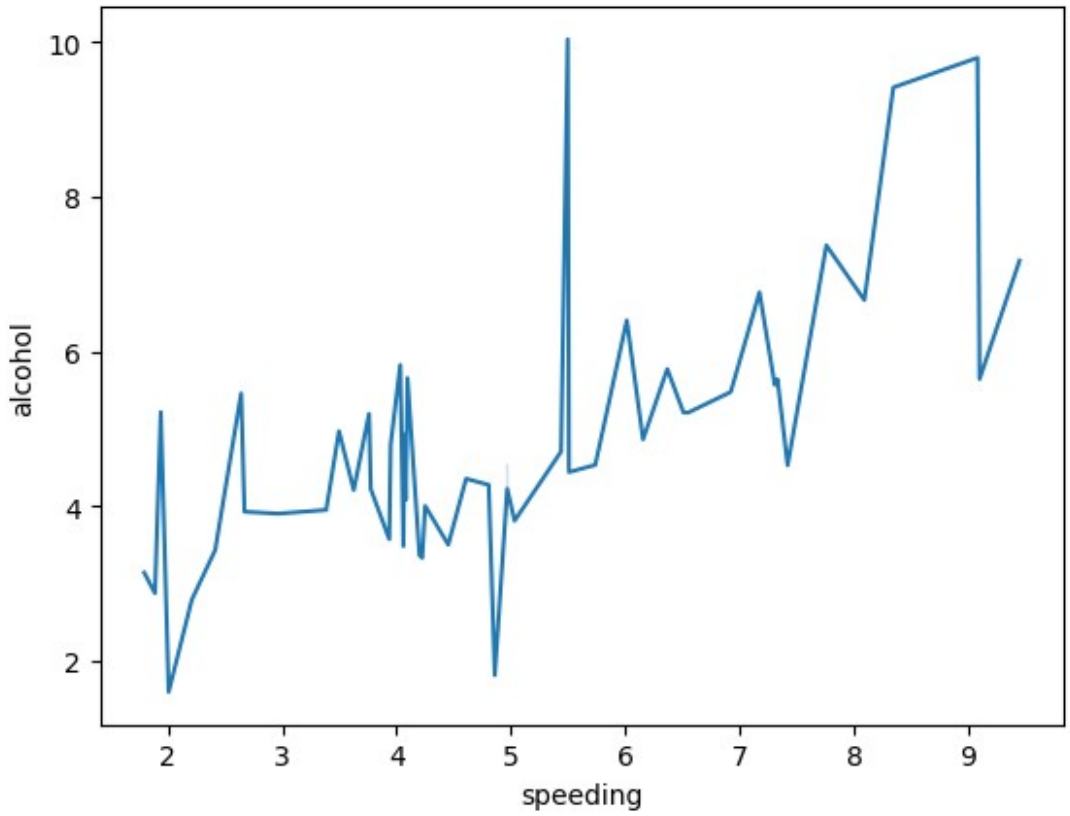
```
<Axes: xlabel='total', ylabel='no_previous'>
```



Scatterplot- Are Speeding cases related to alcohol cases? Checking whether cases of car crashes due to speeding are because of alcohol

```
sns.lineplot(data=car_crashes,x='speeding',y='alcohol')
```

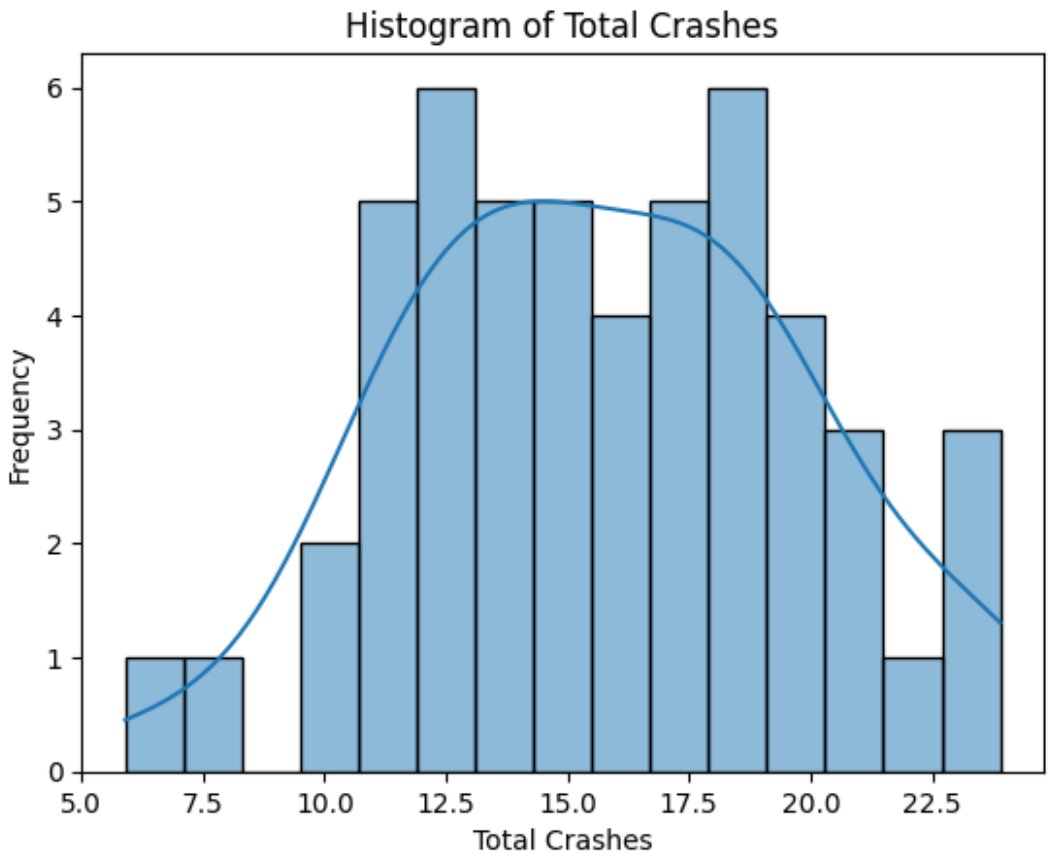
```
<Axes: xlabel='speeding', ylabel='alcohol'>
```



This code snippet uses Seaborn and Matplotlib to create a histogram of the "total crashes" column from the 'car_crashes' dataset with 15 bins and a kernel density estimate (kde), and it prepares the plot for display.

histogram of the number of total crashes in the car_crashes DataFrame, with 15 bins. It also plots a smoothed version of the histogram, which can be used to better visualize the distribution of the data.

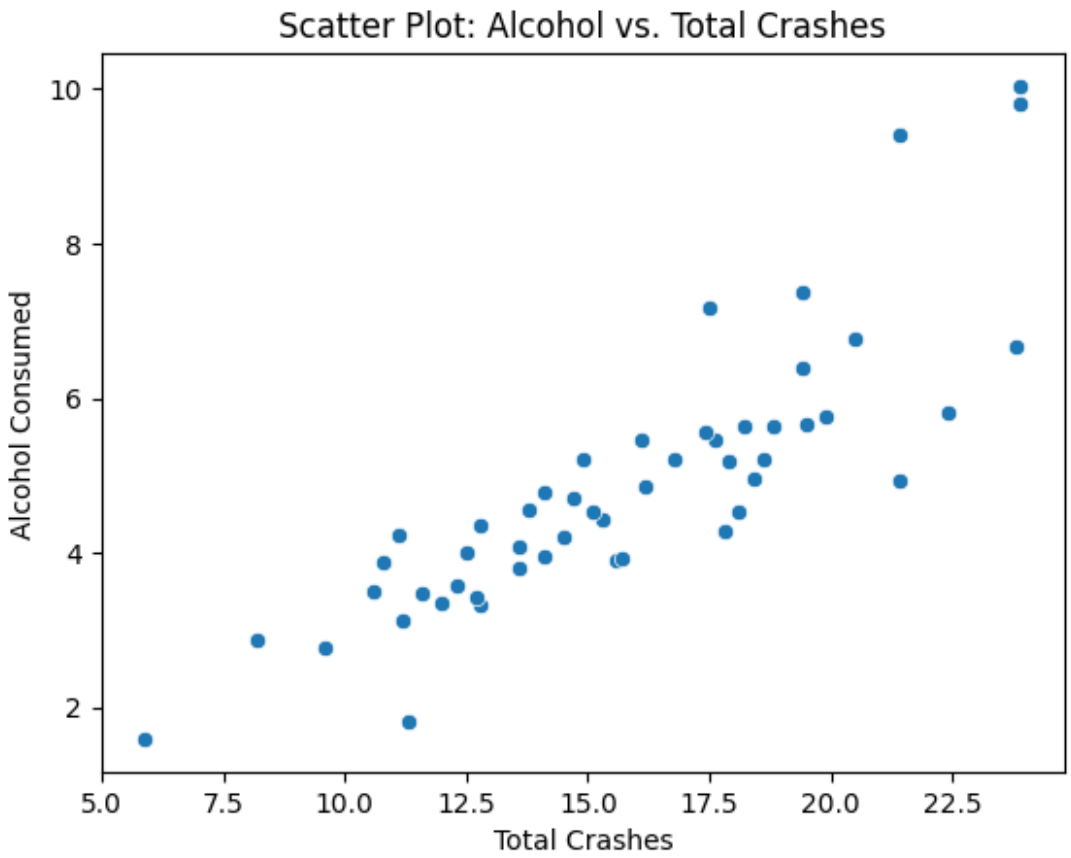
```
#  Histogram of the total crashes
sns.histplot(car_crashes["total"], bins=15, kde=True)
plt.xlabel("Total Crashes")
plt.ylabel("Frequency")
plt.title("Histogram of Total Crashes")
plt.show()
```



This code generates a scatter plot using Seaborn to visualize the relationship between "total crashes" and "alcohol consumed" from the 'car_crashes' dataset. It includes labeled axes and a title for the plot

The plot can be used to visualize the relationship between alcohol consumption and the risk of car crashes..

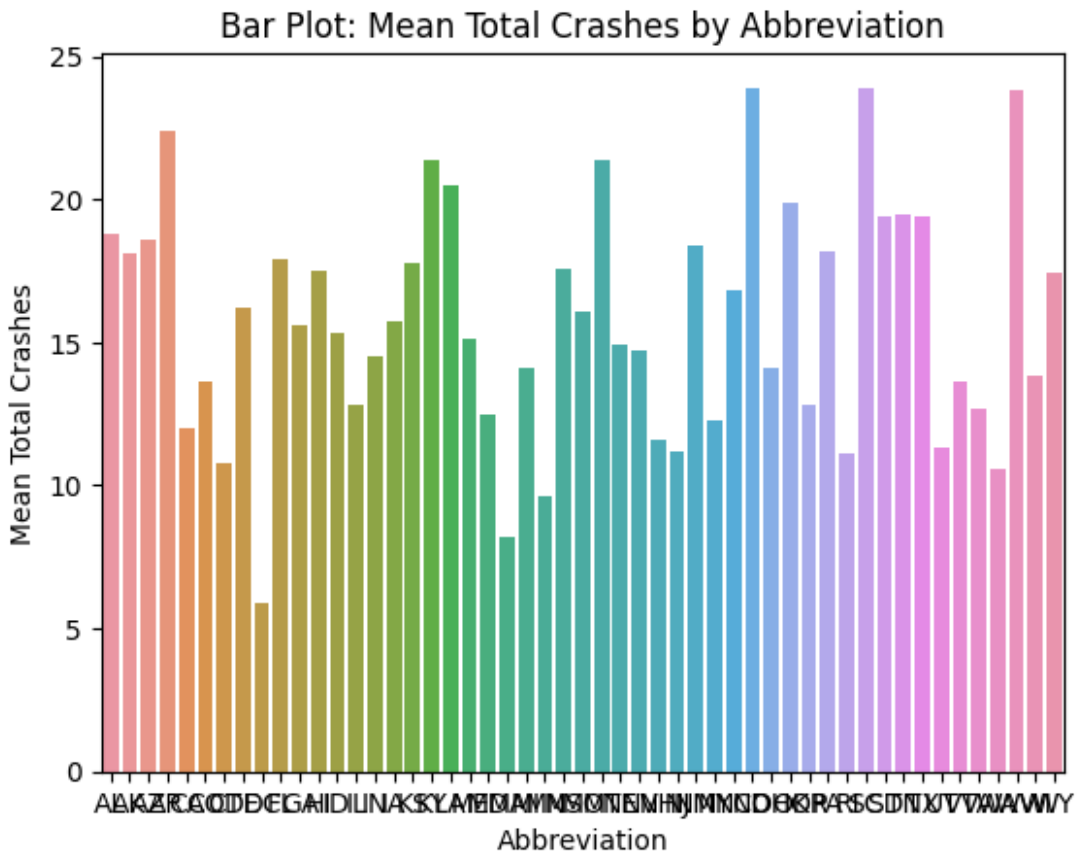
```
sns.scatterplot(x="total", y="alcohol", data=car_crashes)
plt.xlabel("Total Crashes")
plt.ylabel("Alcohol Consumed")
plt.title("Scatter Plot: Alcohol vs. Total Crashes")
plt.show()
```



This code utilizes Seaborn to create a bar plot, displaying the mean total crashes for each abbreviation in the 'car_crashes' dataset. It also adds labels to the axes and a title to the plot for clarity.

bar plot of the mean total crashes for each abbreviation in the car_crashes DataFrame. The plt.show() function displays the plot.

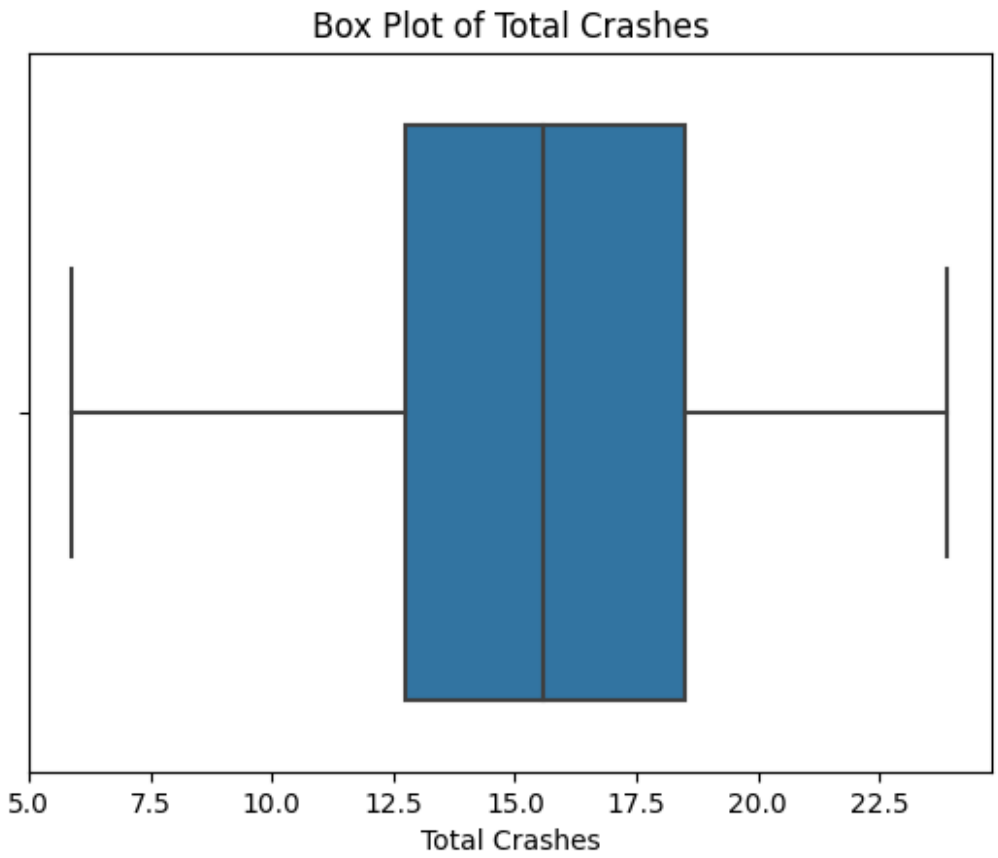
```
sns.barplot(x="abbrev", y="total", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Mean Total Crashes")
plt.title("Bar Plot: Mean Total Crashes by Abbreviation")
plt.show()
```



This code employs Seaborn to generate a box plot that visualizes the distribution of "total crashes" from the 'car_crashes' dataset. It includes a labeled x-axis and a title for the plot.

box plot of the total number of crashes in the car_crashes DataFrame. The plot can be used to visualize the distribution of the data and identify outliers.

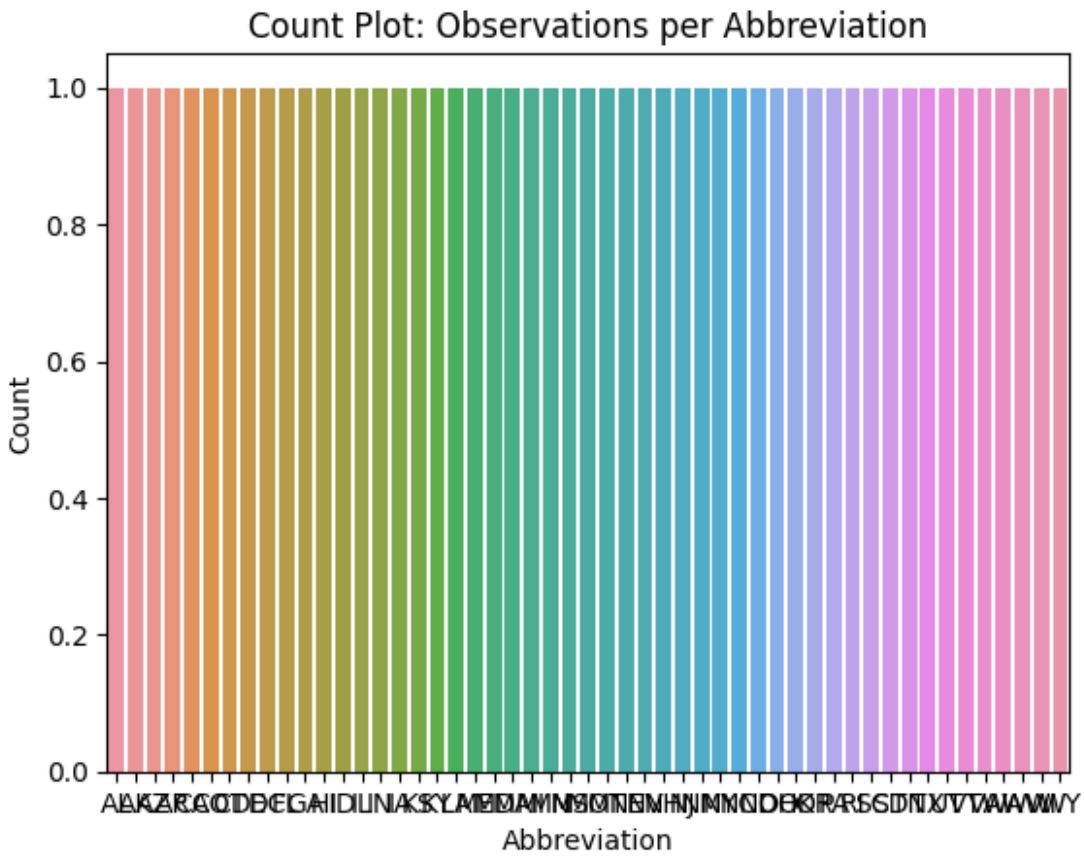
```
sns.boxplot(x="total", data=car_crashes)
plt.xlabel("Total Crashes")
plt.title("Box Plot of Total Crashes")
plt.show()
```



This code utilizes Seaborn to create a count plot, illustrating the number of observations per abbreviation in the 'car_crashes' dataset. It includes labeled axes and a title for the plot.

count plot of the number of observations for each abbreviation in the car_crashes DataFrame. The plot can be used to identify the states or regions with the most and least car crashes.

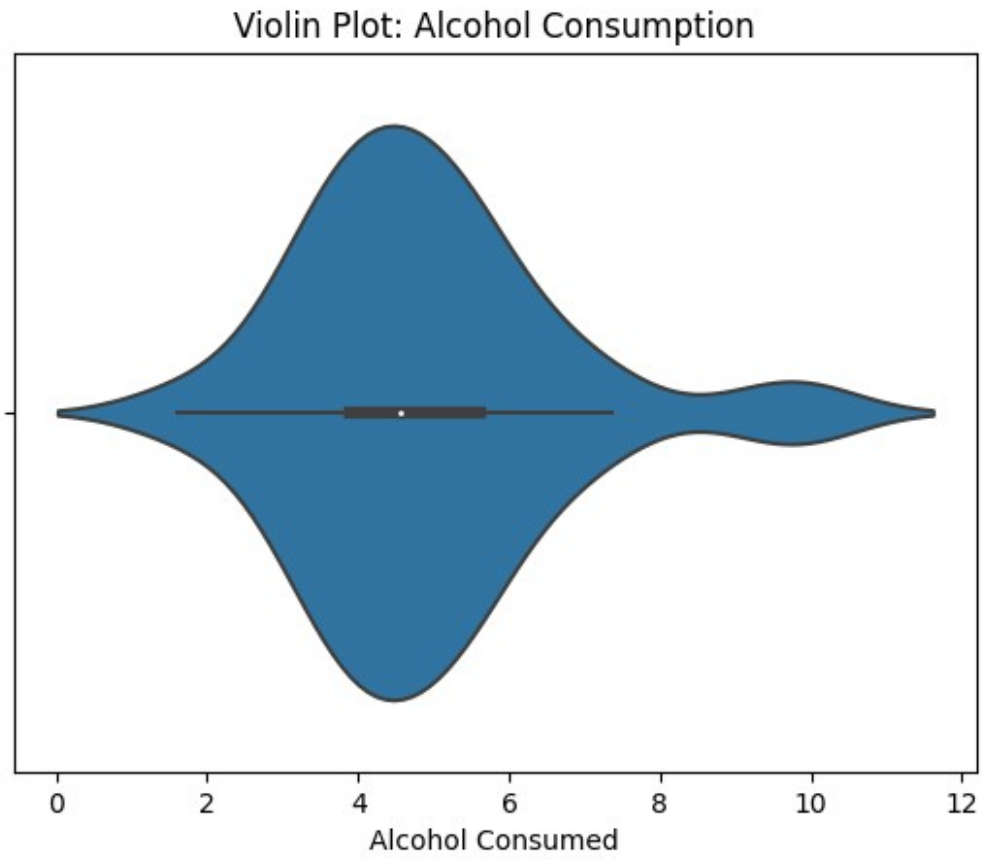
```
sns.countplot(x="abbrev", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Count")
plt.title("Count Plot: Observations per Abbreviation")
plt.show()
```



This code uses Seaborn to generate a violin plot, displaying the distribution of "alcohol consumed" from the 'car_crashes' dataset. It includes a labeled x-axis and a title for the plot.

violin plot of the alcohol consumption for each driver in the car_crashes DataFrame. The plot can be used to visualize the distribution of alcohol consumption and identify outliers.

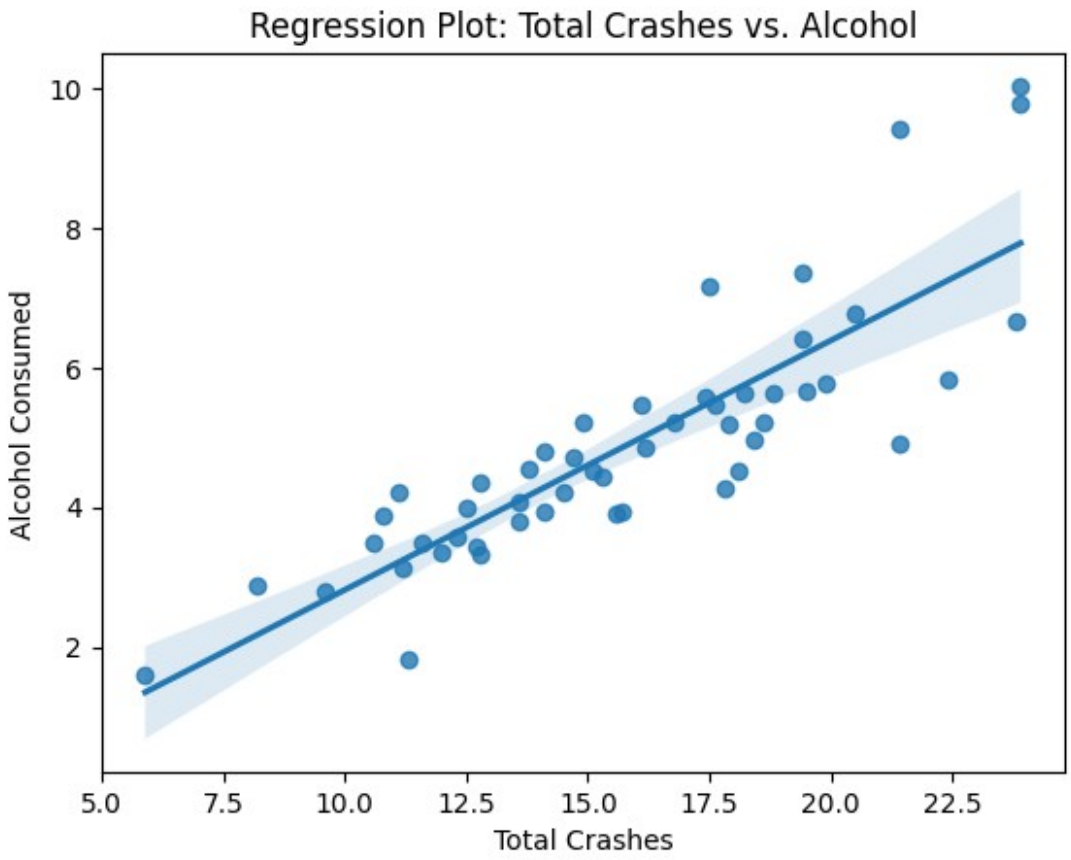
```
sns.violinplot(x="alcohol", data=car_crashes)
plt.xlabel("Alcohol Consumed")
plt.title("Violin Plot: Alcohol Consumption")
plt.show()
```



This code creates a regression plot using Seaborn, illustrating the relationship between "total crashes" and "alcohol consumed" in the 'car_crashes' dataset. It includes labeled axes and a title for the plot.

regression plot of the total number of crashes versus the amount of alcohol consumed by drivers in the car_crashes DataFrame.

```
sns.regplot(x="total", y="alcohol", data=car_crashes)
plt.xlabel("Total Crashes")
plt.ylabel("Alcohol Consumed")
plt.title("Regression Plot: Total Crashes vs. Alcohol")
plt.show()
```

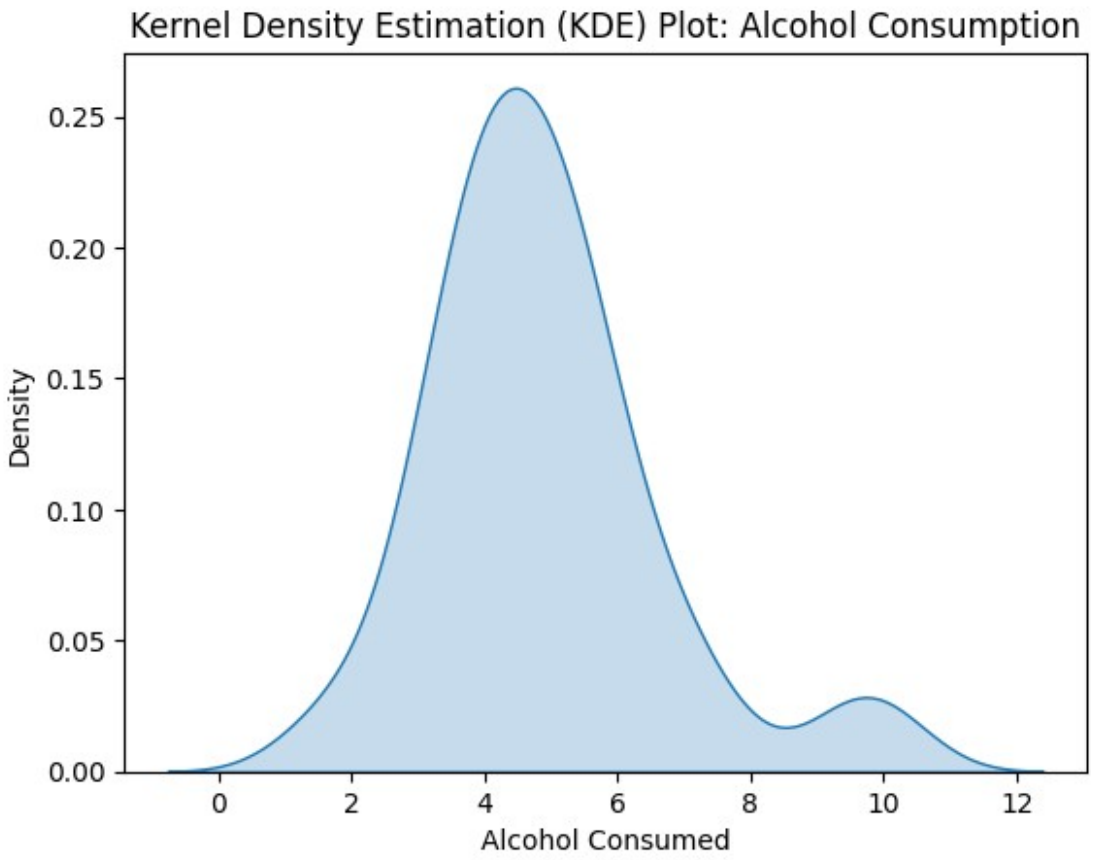


This code generates a Kernel Density Estimation (KDE) plot using Seaborn for the "alcohol consumed" data from the 'car_crashes' dataset. It uses shading to represent the KDE and includes labeled axes and a title for the plot.

kernel density estimation (KDE) plot of the alcohol consumption for each driver in the car_crashes DataFrame. The plot can be used to visualize the distribution of alcohol consumption and identify outliers.

```
sns.kdeplot(car_crashes["alcohol"], shade=True)
plt.xlabel("Alcohol Consumed")
plt.title("Kernel Density Estimation (KDE) Plot: Alcohol Consumption")
plt.show()
```

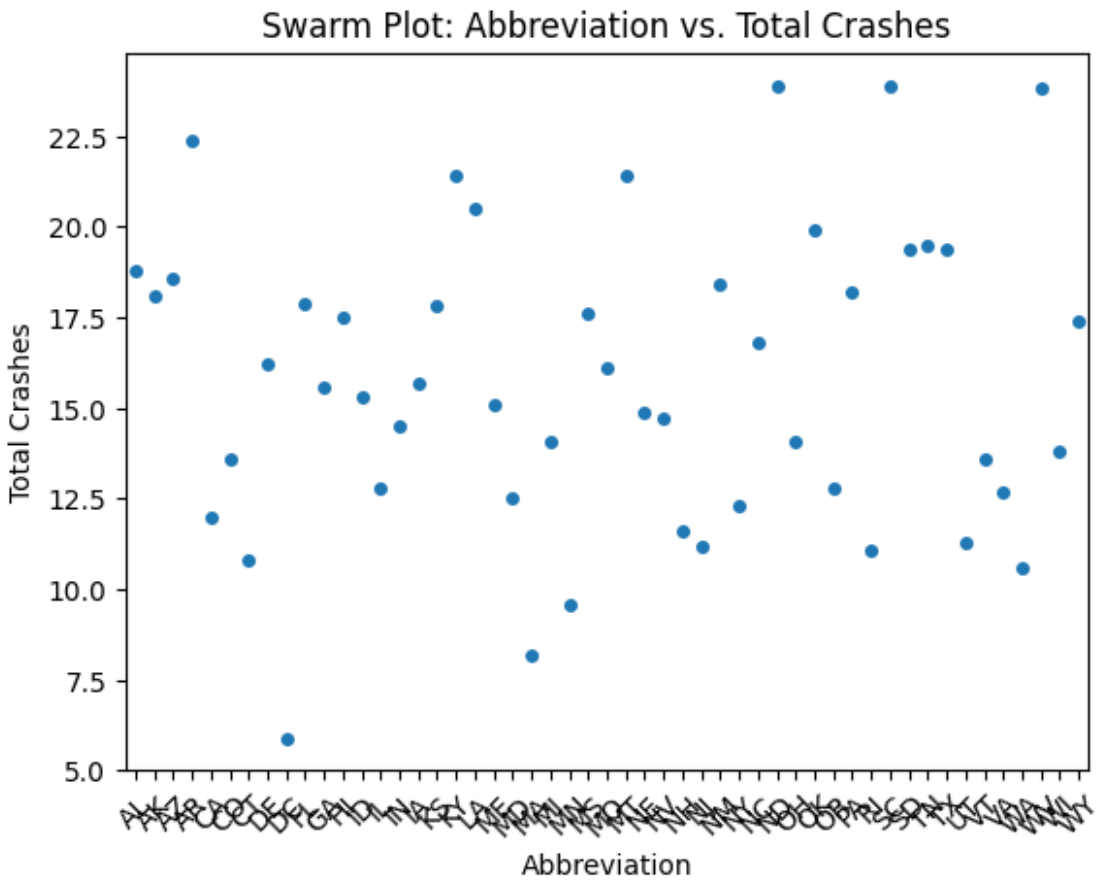
```
<ipython-input-16-d3625963060d>:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(car_crashes["alcohol"], shade=True)
```



This code creates a swarm plot using Seaborn to visualize the relationship between "abbreviation" and "total crashes" in the 'car_crashes' dataset. It includes labeled axes, a title, and rotates the x-axis labels for better readability.

swarm plot of the total number of crashes for each abbreviation in the car_crashes DataFrame. The plot can be used to visualize the distribution of the data and identify outliers.

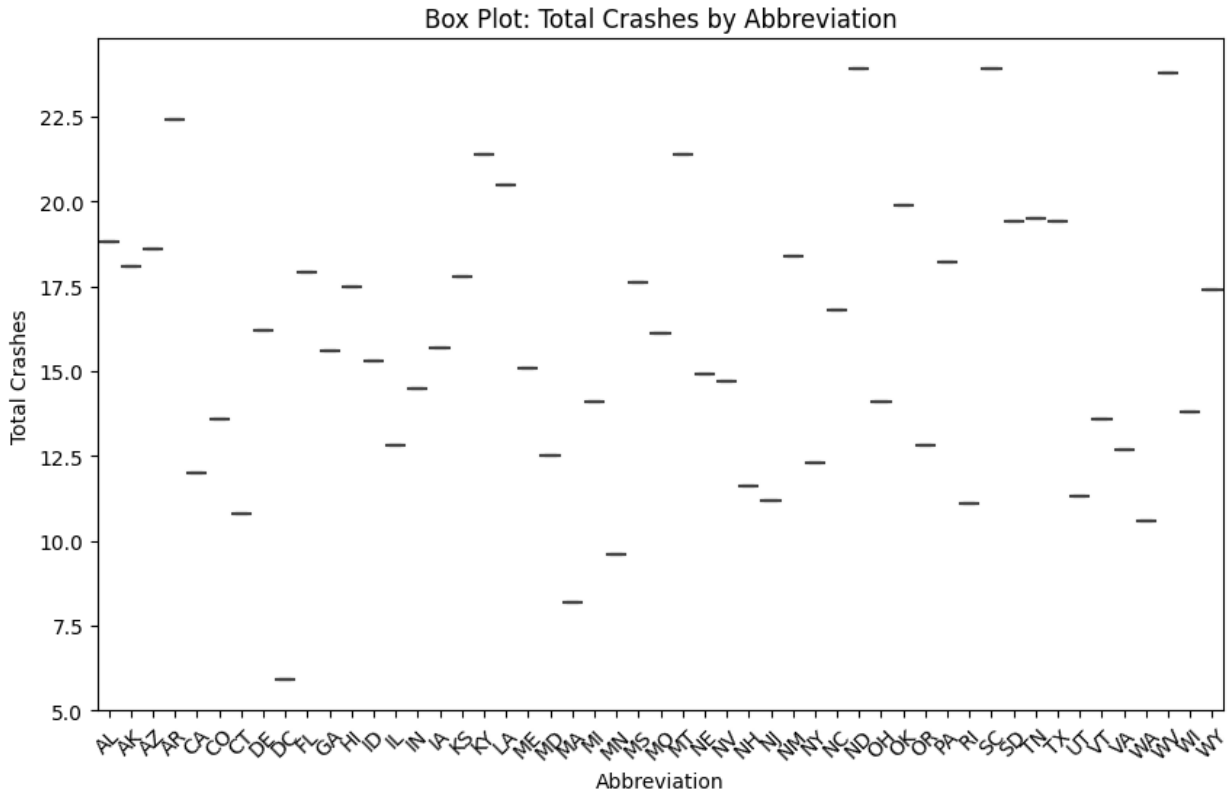
```
sns.swarmplot(x="abbrev", y="total", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Total Crashes")
plt.title("Swarm Plot: Abbreviation vs. Total Crashes")
plt.xticks(rotation=45)
plt.show()
```



This code generates a box plot using Seaborn to display the distribution of "total crashes" by "abbreviation" in the 'car_crashes' dataset. It sets the figure size, includes labeled axes, a title, and rotates the x-axis labels for better visualization.

box plot of the total number of crashes for each abbreviation in the car_crashes DataFrame. The plot can be used to visualize the distribution of the data and identify outliers.

```
plt.figure(figsize=(10, 6))
sns.boxplot(x="abbrev", y="total", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Total Crashes")
plt.title("Box Plot: Total Crashes by Abbreviation")
plt.xticks(rotation=45)
plt.show()
```


Box Plot: Total Crashes by Abbreviation
Total Crashes
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
AL AK AZ AR CA CO CT DE DC FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY
Abbreviation