

NAME: P.NAGA BHAVANI

SLOT: MORNING SLOT

CAMPUS: VIT AP

REGISTRATION NO.: 21BCE9043

ASSIGNMENT NO.: 2

"Assignment 8 th september,

"1.Take car crashes dataset from seaborn library",

"2.load the dataset ",

"3.data visualiation ",

"4.Inference is must for each and every graph",

"5.Submit it by wednesday in html format",

```
In [19]: import seaborn as sns  
import matplotlib.pyplot as plt
```

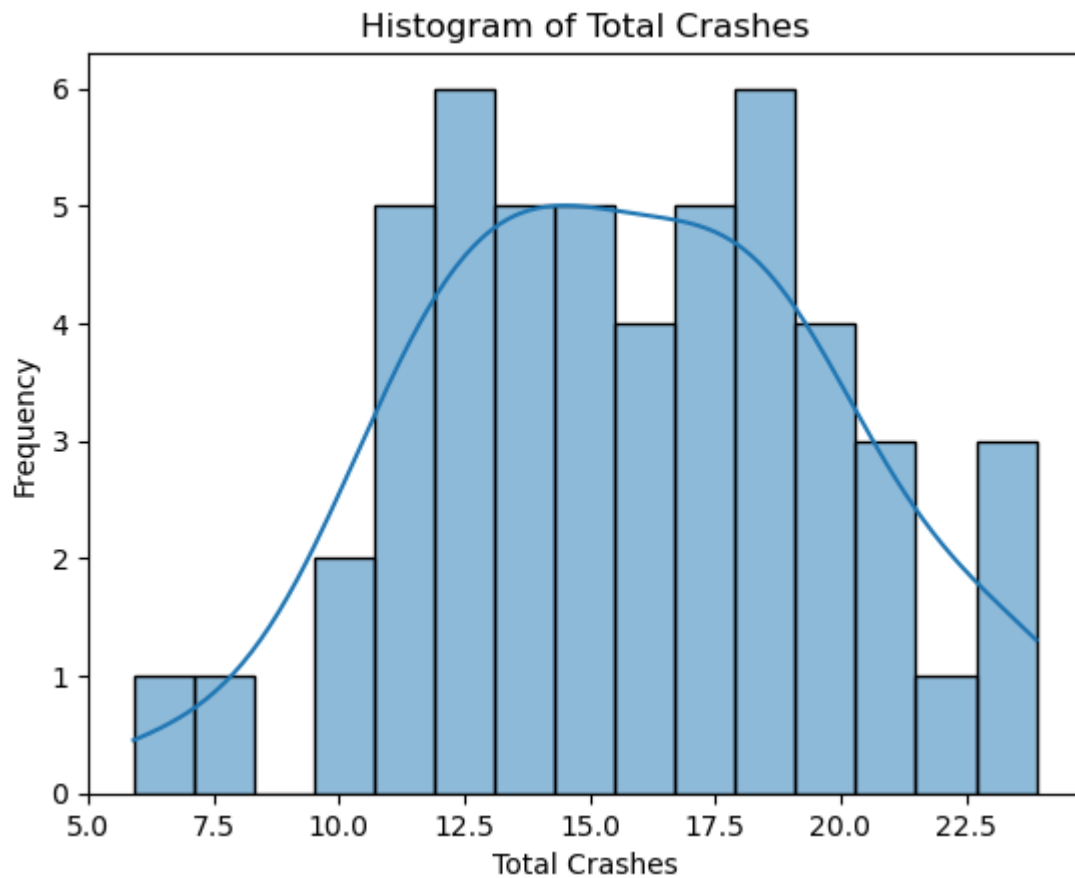
```
In [20]: # Load the car_crashes dataset  
car_crashes = sns.load_dataset("car_crashes")  
car_crashes
```

Out[20]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_lo
<b>0</b>	18.8	7.332	5.640	18.048	15.040	784.55	14
<b>1</b>	18.1	7.421	4.525	16.290	17.014	1053.48	13
<b>2</b>	18.6	6.510	5.208	15.624	17.856	899.47	11
<b>3</b>	22.4	4.032	5.824	21.056	21.280	827.34	14
<b>4</b>	12.0	4.200	3.360	10.920	10.680	878.41	16
<b>5</b>	13.6	5.032	3.808	10.744	12.920	835.50	13
<b>6</b>	10.8	4.968	3.888	9.396	8.856	1068.73	16
<b>7</b>	16.2	6.156	4.860	14.094	16.038	1137.87	15
<b>8</b>	5.9	2.006	1.593	5.900	5.900	1273.89	13
<b>9</b>	17.9	3.759	5.191	16.468	16.826	1160.13	14
<b>10</b>	15.6	2.964	3.900	14.820	14.508	913.15	14
<b>11</b>	17.5	9.450	7.175	14.350	15.225	861.18	12
<b>12</b>	15.3	5.508	4.437	13.005	14.994	641.96	8
<b>13</b>	12.8	4.608	4.352	12.032	12.288	803.11	13
<b>14</b>	14.5	3.625	4.205	13.775	13.775	710.46	10
<b>15</b>	15.7	2.669	3.925	15.229	13.659	649.06	11
<b>16</b>	17.8	4.806	4.272	13.706	15.130	780.45	13
<b>17</b>	21.4	4.066	4.922	16.692	16.264	872.51	13
<b>18</b>	20.5	7.175	6.765	14.965	20.090	1281.55	19
<b>19</b>	15.1	5.738	4.530	13.137	12.684	661.88	9
<b>20</b>	12.5	4.250	4.000	8.875	12.375	1048.78	19
<b>21</b>	8.2	1.886	2.870	7.134	6.560	1011.14	13
<b>22</b>	14.1	3.384	3.948	13.395	10.857	1110.61	15
<b>23</b>	9.6	2.208	2.784	8.448	8.448	777.18	13
<b>24</b>	17.6	2.640	5.456	1.760	17.600	896.07	15
<b>25</b>	16.1	6.923	5.474	14.812	13.524	790.32	14
<b>26</b>	21.4	8.346	9.416	17.976	18.190	816.21	8
<b>27</b>	14.9	1.937	5.215	13.857	13.410	732.28	11
<b>28</b>	14.7	5.439	4.704	13.965	14.553	1029.87	13
<b>29</b>	11.6	4.060	3.480	10.092	9.628	746.54	12
<b>30</b>	11.2	1.792	3.136	9.632	8.736	1301.52	15
<b>31</b>	18.4	3.496	4.968	12.328	18.032	869.85	12
<b>32</b>	12.3	3.936	3.567	10.824	9.840	1234.31	15

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_lo
<b>33</b>	16.8	6.552	5.208	15.792	13.608	708.24	12
<b>34</b>	23.9	5.497	10.038	23.661	20.554	688.75	10
<b>35</b>	14.1	3.948	4.794	13.959	11.562	697.73	13
<b>36</b>	19.9	6.368	5.771	18.308	18.706	881.51	17
<b>37</b>	12.8	4.224	3.328	8.576	11.520	804.71	10
<b>38</b>	18.2	9.100	5.642	17.472	16.016	905.99	15
<b>39</b>	11.1	3.774	4.218	10.212	8.769	1148.99	14
<b>40</b>	23.9	9.082	9.799	22.944	19.359	858.97	11
<b>41</b>	19.4	6.014	6.402	19.012	16.684	669.31	9
<b>42</b>	19.5	4.095	5.655	15.990	15.795	767.91	15
<b>43</b>	19.4	7.760	7.372	17.654	16.878	1004.75	15
<b>44</b>	11.3	4.859	1.808	9.944	10.848	809.38	10
<b>45</b>	13.6	4.080	4.080	13.056	12.920	716.20	10
<b>46</b>	12.7	2.413	3.429	11.049	11.176	768.95	15
<b>47</b>	10.6	4.452	3.498	8.692	9.116	890.03	11
<b>48</b>	23.8	8.092	6.664	23.086	20.706	992.61	15
<b>49</b>	13.8	4.968	4.554	5.382	11.592	670.31	10
<b>50</b>	17.4	7.308	5.568	14.094	15.660	791.14	12

```
In [21]: # 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()
```

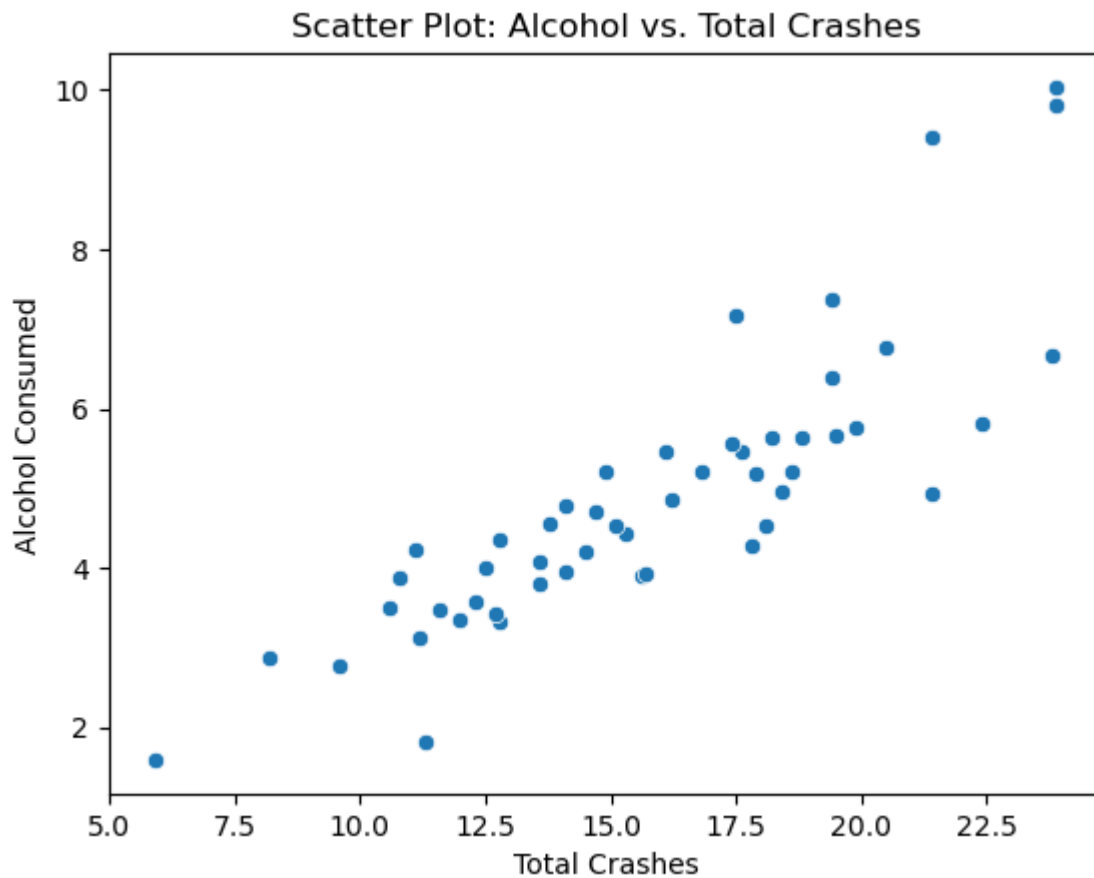


histogram of the number of total crashes in the `car_crashes` DataFrame, with 15 bins. It also

plots a smoothed version of the histogram, which can be used to better visualize the

distribution of the data.

```
In [22]: # Scatter plot of alcohol versus total 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()
```

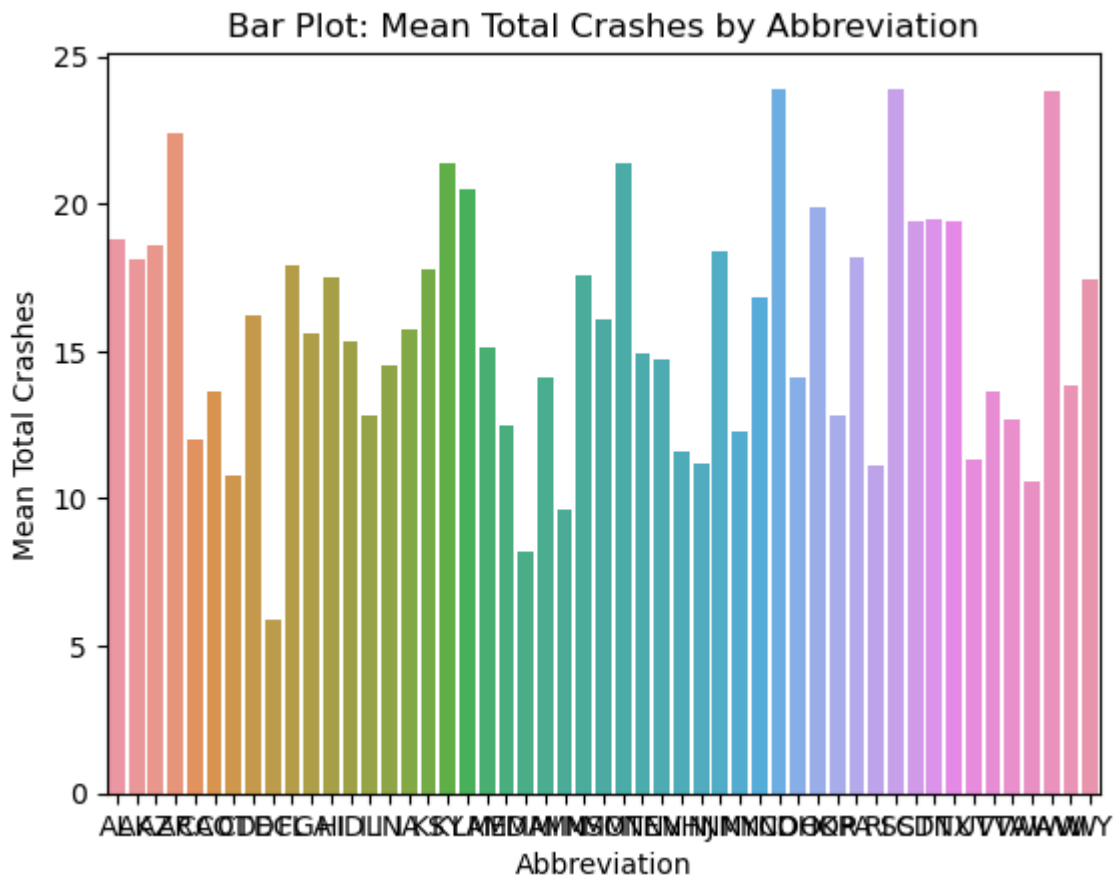


scatter plot of the total number of crashes versus the amount of alcohol consumed by drivers

in the car\_crashes DataFrame. The plot can be used to visualize the relationship between

alcohol consumption and the risk of car crashes..#

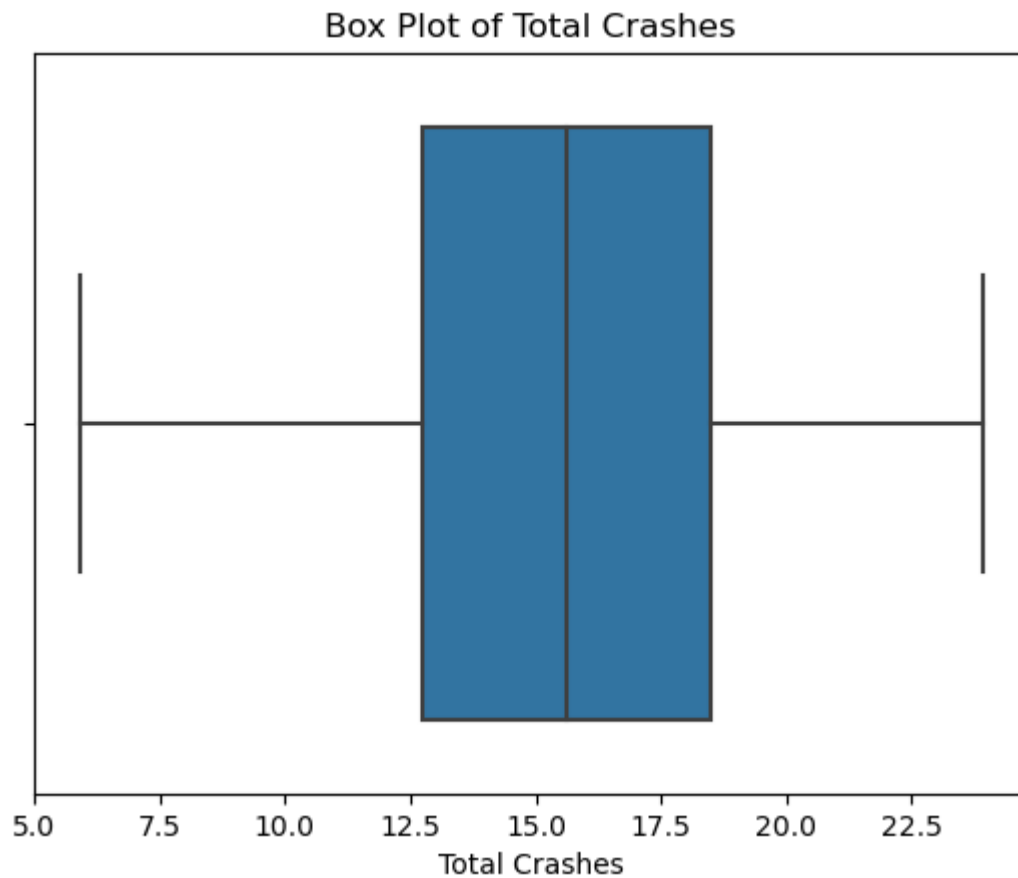
```
In [23]: sns.barplot(x="abbrev", y="total", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Mean Total Crashes")
plt.title("Bar Plot: Mean Total Crashes by Abbreviation")
plt.show()
```



bar plot of the mean total crashes for each abbreviation in the car\_crashes DataFrame. The

plt.show() function displays the plot.

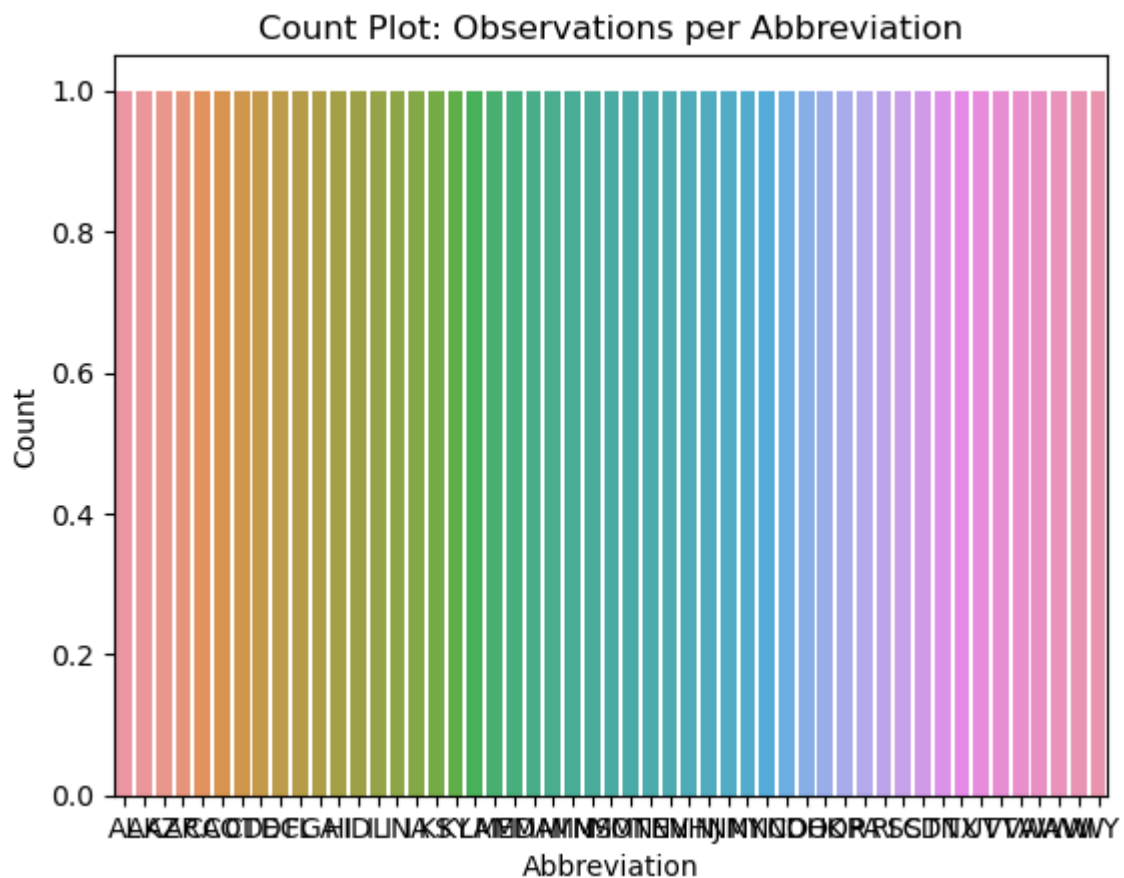
```
In [24]: sns.boxplot(x="total", data=car_crashes)
plt.xlabel("Total Crashes")
plt.title("Box Plot of Total Crashes")
plt.show()
```



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.

```
In [25]: sns.countplot(x="abbrev", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Count")
plt.title("Count Plot: Observations per Abbreviation")
plt.show()
```

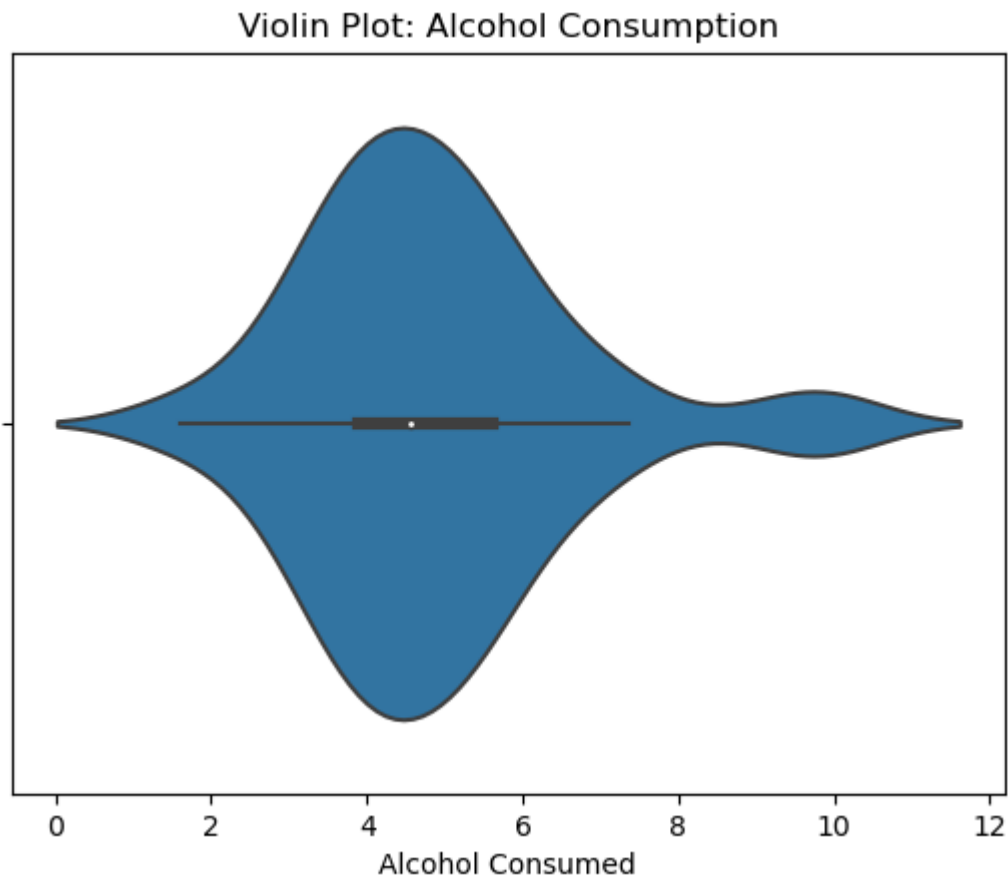


count plot of the number of observations for each abbreviation in the car\_crashes DataFrame.

The plot can be used to identify the states or regions with the most and least car crashes.

```
In [26]: sns.violinplot(x="alcohol", data=car_crashes)
plt.xlabel("Alcohol Consumed")
plt.title("Violin Plot: Alcohol Consumption")
plt.show()
```

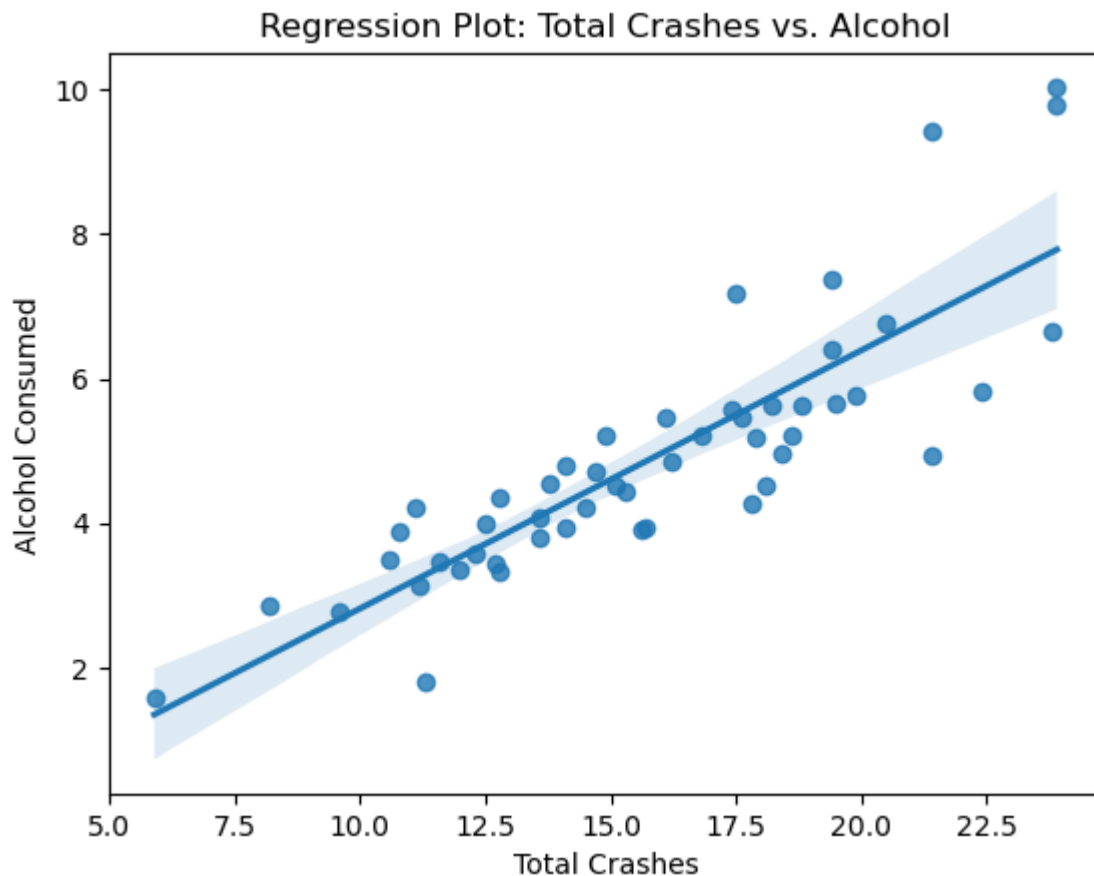




violin plot of the alcohol consumption for each driver in the `car_crashes` DataFrame. The plot

can be used to visualize the distribution of alcohol consumption and identify outliers.

```
In [27]: 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()
```



regression plot of the total number of crashes versus the amount of alcohol consumed by

drivers in the car\_crashes DataFrame. The plot can be used to visualize the relationship

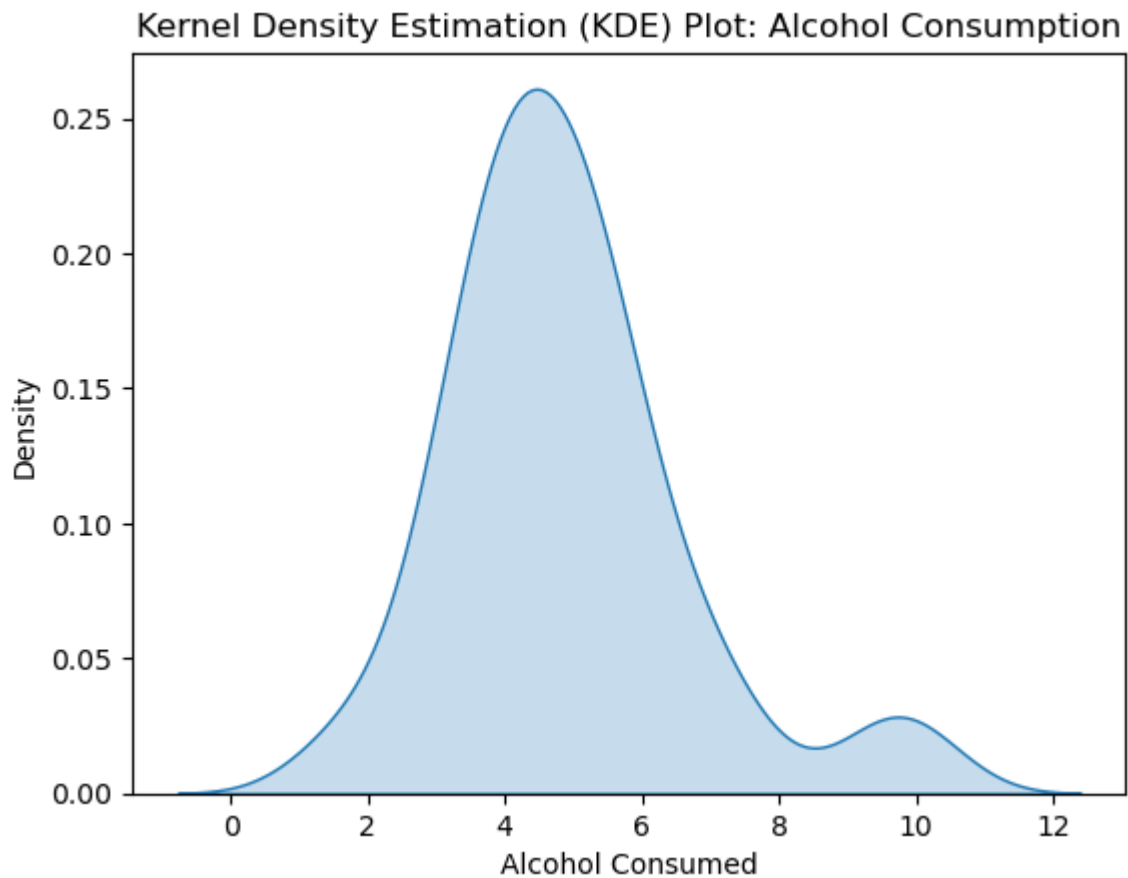
between alcohol consumption and the risk of car crashes.

```
In [28]: sns.kdeplot(car_crashes["alcohol"], shade=True)
plt.xlabel("Alcohol Consumed")
plt.title("Kernel Density Estimation (KDE) Plot: Alcohol Consumption")
plt.show()
```

C:\Users\NAGA BHAVANI\AppData\Local\Temp\ipykernel\_6932\1084857995.py: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)
```

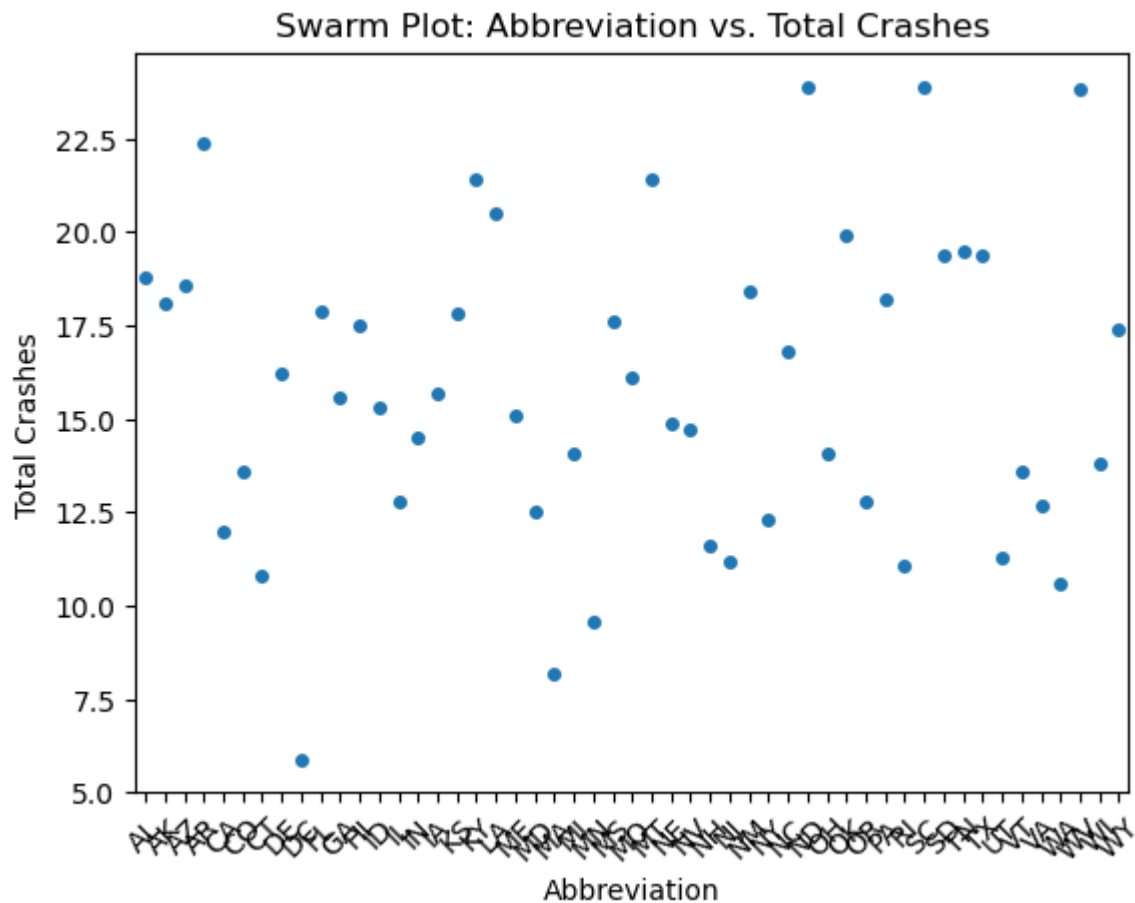


kernel density estimation (KDE) plot of the alcohol consumption for each driver in the

car\_crashes DataFrame. The plot can be used to visualize the distribution of alcohol

consumption and identify outliers.

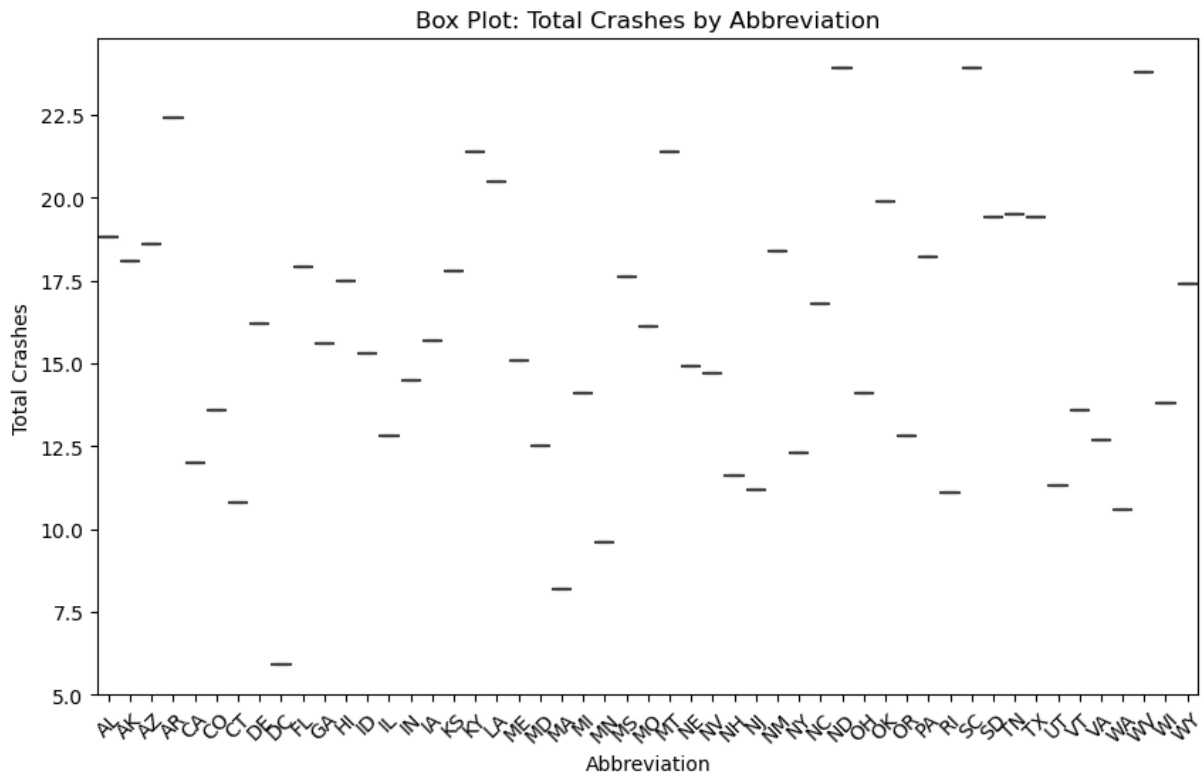
```
In [29]: sns.swarmplot(x="abbrev", y="total", data=car_crashes)
plt.xlabel("Abbreviation")
plt.ylabel("Total Crashes")
plt.title("Swarm Plot: Abbreviation vs. Total Crashes")
plt.xticks(rotation=45)
plt.show()
```



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.

```
In [30]: 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 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.

In [ ]: