21bce9028-ai-ml-assignment-2

AI and ML Assignment 2

Name: J.S.Rithwik

Reg No: 21BCE9028

[14]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
```

[3]:
```python
df=sns.load_dataset("car_crashes")
```

[4]:
```python
df.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
```

[5]:
```python
df.describe
```

[5]:
```
<bound method NDFrame.describe of     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
```

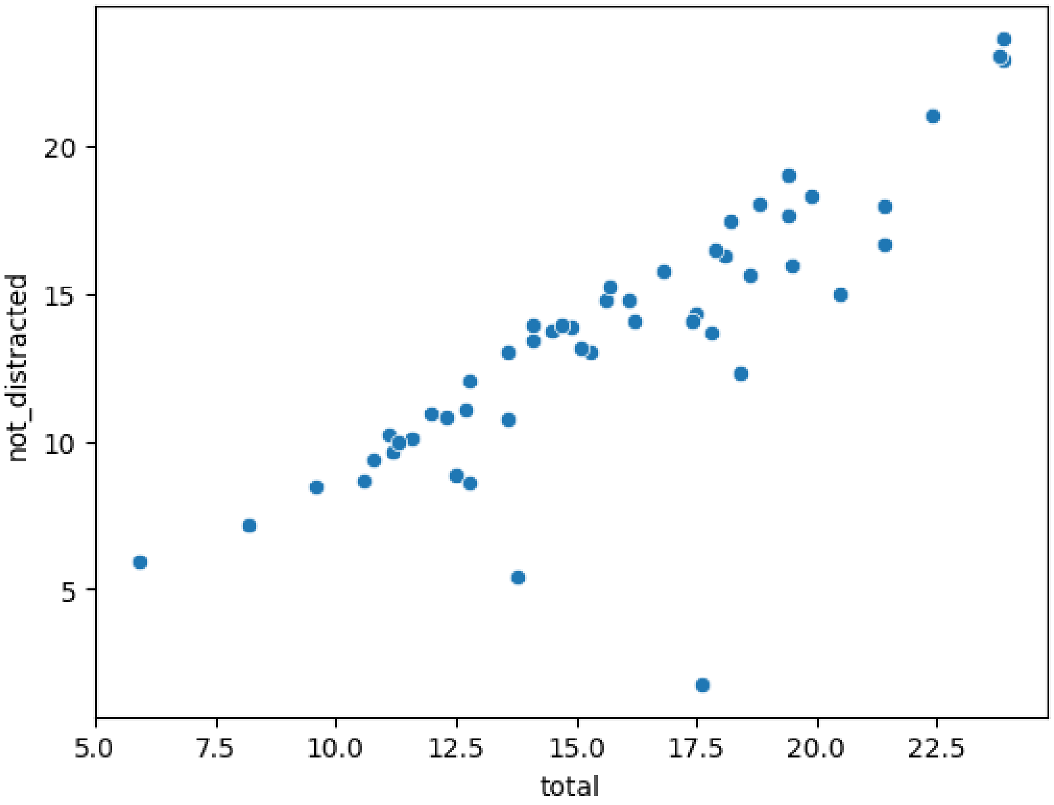
```
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY  >
```

```
[6]: df.head()
```

```
[6]:    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
```

```
[7]: sns.scatterplot(x="total",y="not_distracted",data=df)
```

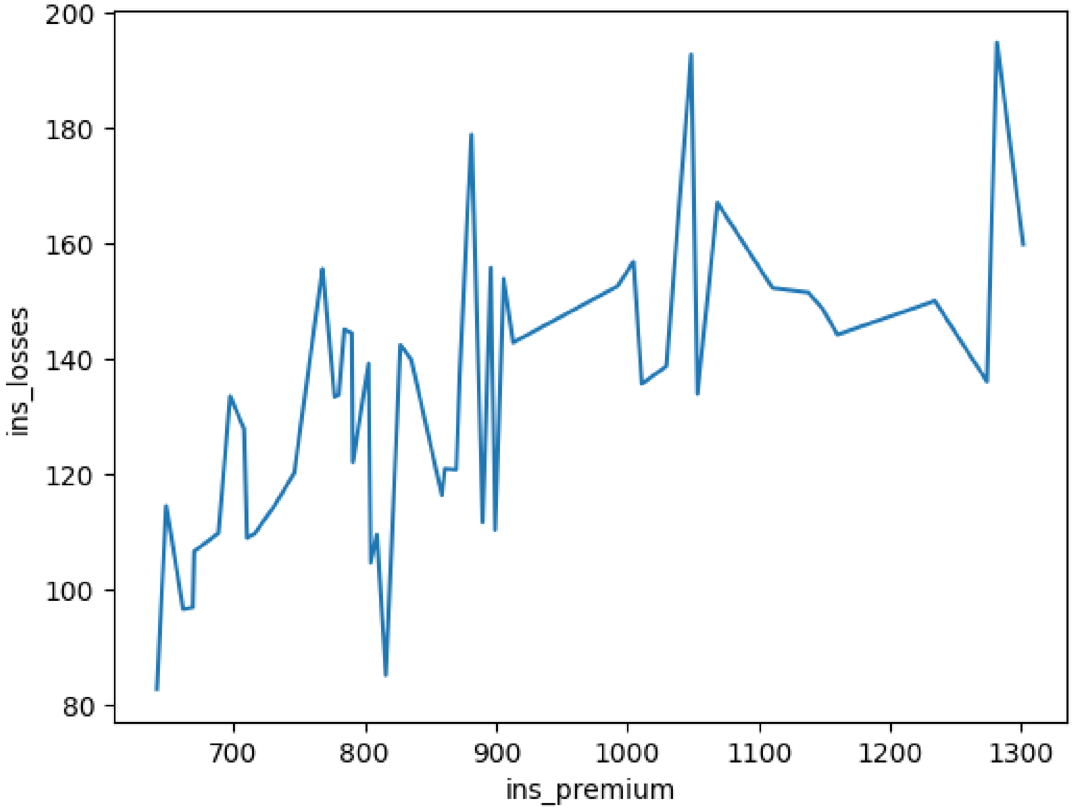
```
[7]: <Axes: xlabel='total', ylabel='not_distracted'>
```



Inference: not_distracted is directly proportional and linearly related to not_distracted

```
[9]: sns.lineplot(x="ins_premium",y="ins_losses",data=df)
```

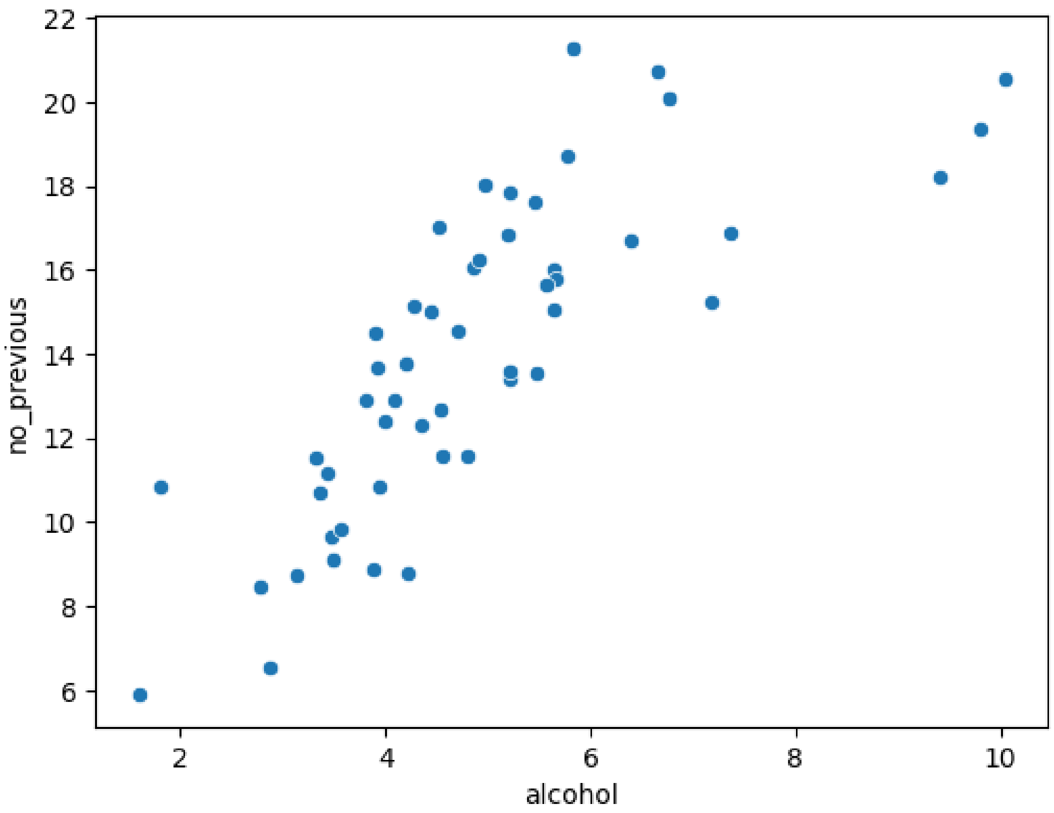
```
[9]: <Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



Inference: As ins_prmium is increasing ins_losses are increasing (not strictly directly proportional) , there are highs and lows in between .

```
[8]: sns.scatterplot(x='alcohol', y='no_previous',data=df)
```

```
[8]: <Axes: xlabel='alcohol', ylabel='no_previous'>
```

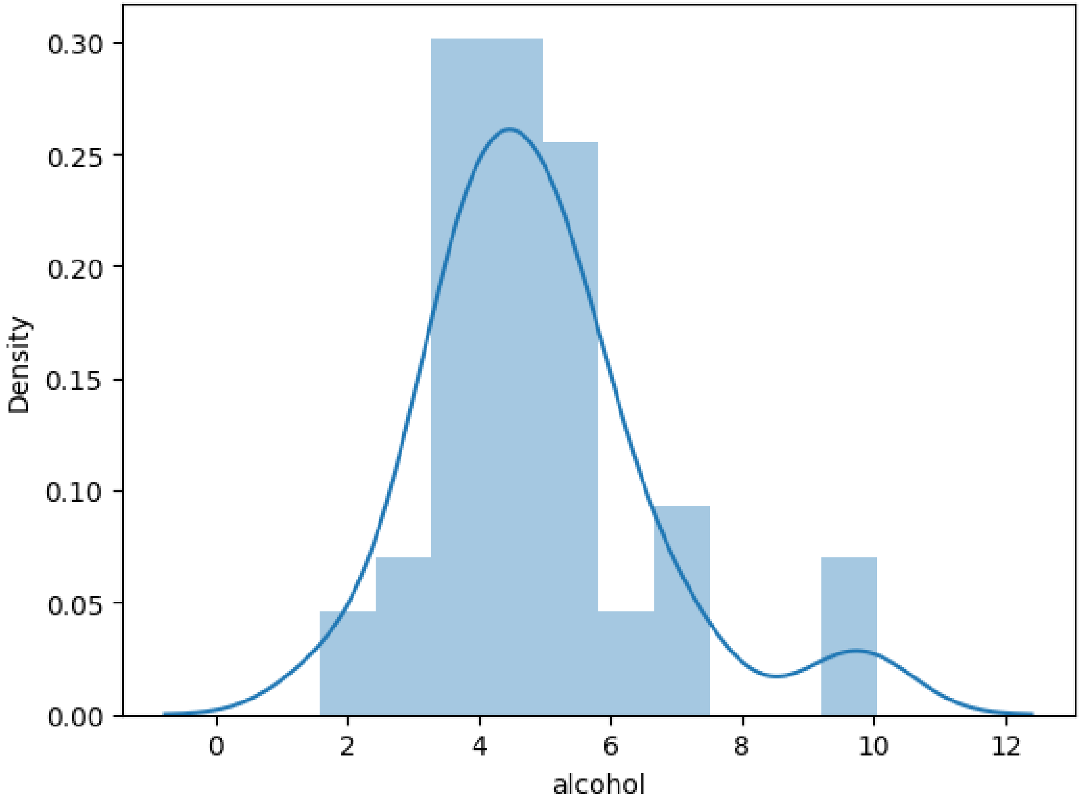


Inference: From above scatterplot no_previous is directly proportional to alcohol

[10]:
```
sns.distplot(df['alcohol'])
```

```
<ipython-input-10-570de8ff0310>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['alcohol'])
```

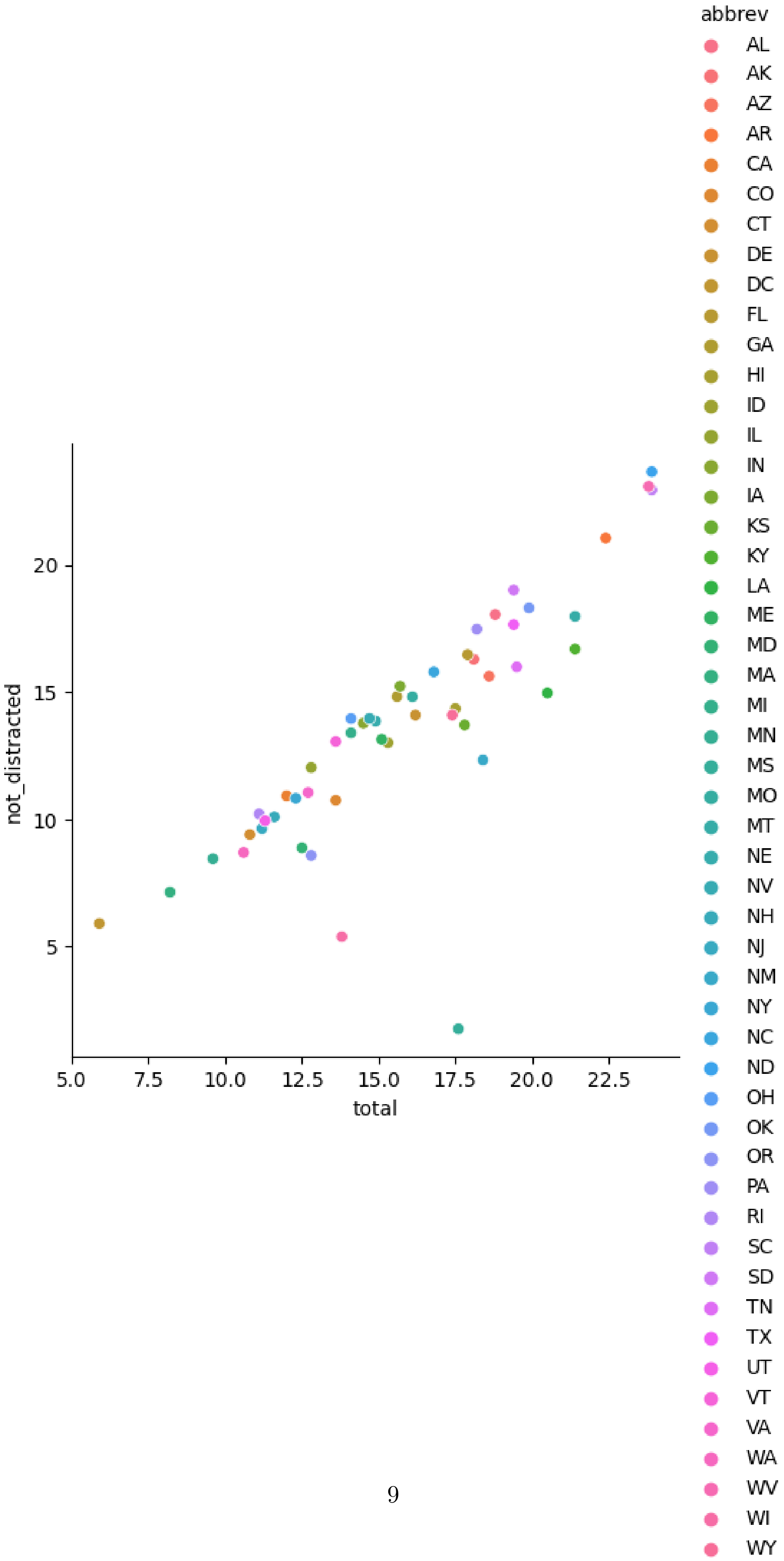
[10]:
```
<Axes: xlabel='alcohol', ylabel='Density'>
```



Inference: Maximum density 0.30 is present at alcohol level 4.

[11]:
```
sns.relplot(x='total',y='not_distracted',data=df,hue='abbrev')
```

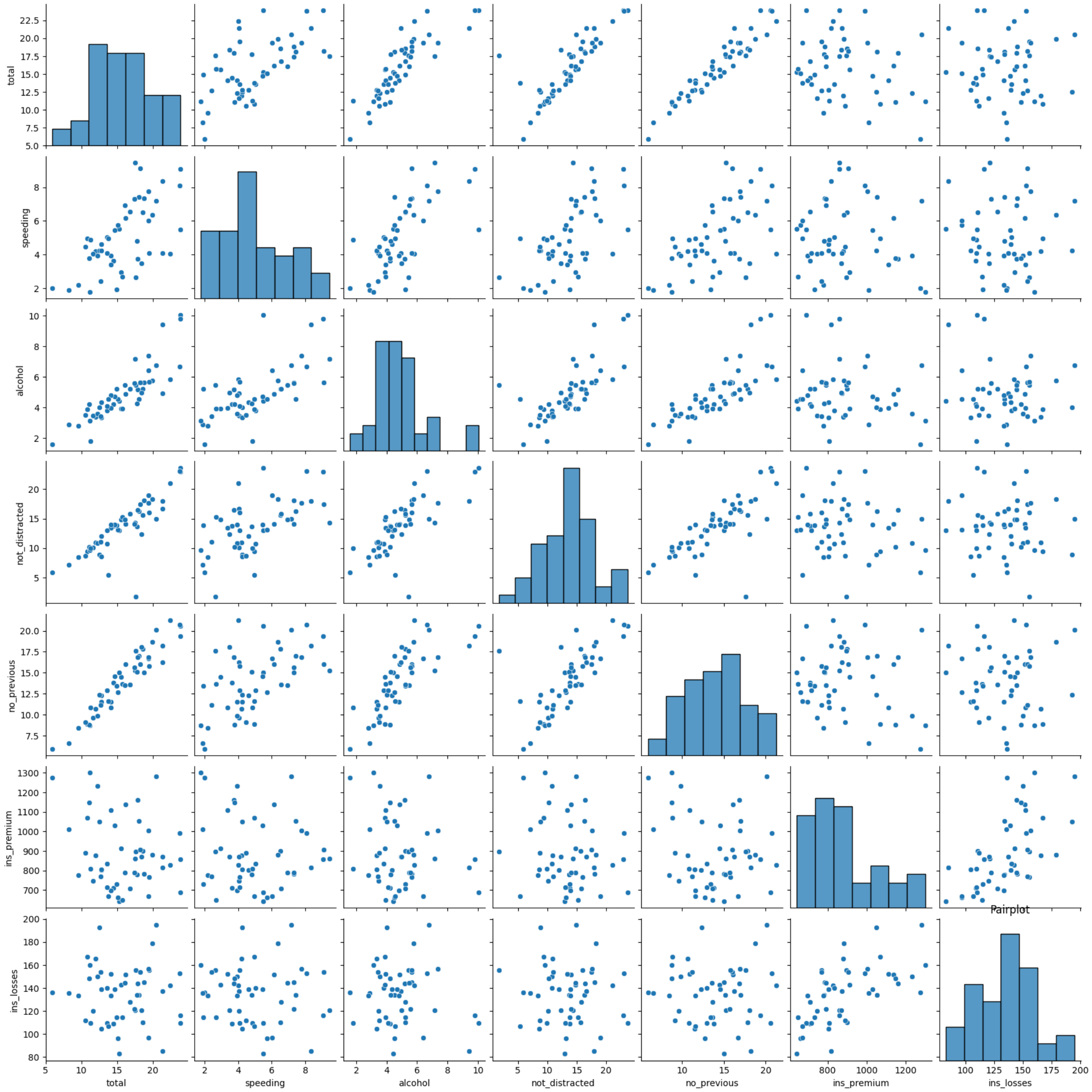
[11]:
```
<seaborn.axisgrid.FacetGrid at 0x7c99bc47be80>
```


abbrev
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
not_distracted
20
15
10
5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
total

Inference: From the above graph we can conclude that as total drivers are incresing not distracting drivers are also increasing which is positively co-related.

[15]:

```
sns.pairplot(df)
plt.title("Pairplot")
plt.show()
```

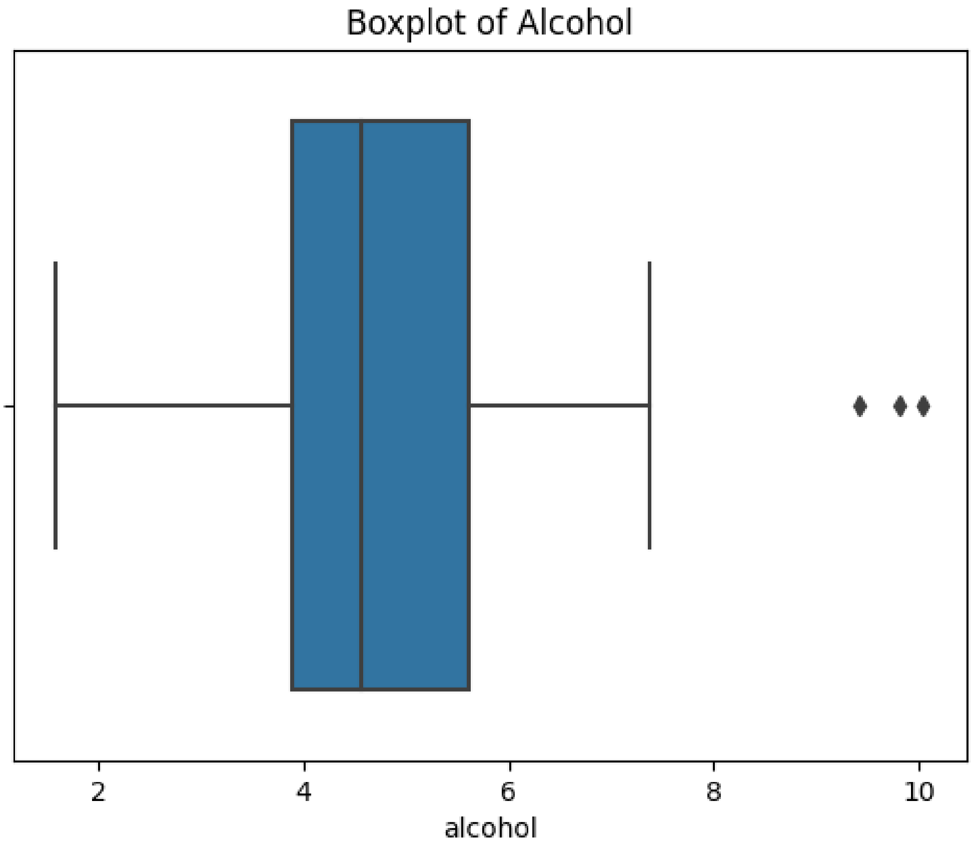


Inference:

- The pairplot provides a quick overview of relationships between all pairs of numerical variables.

- Some variables show positive correlations (e.g., total vs. not_distracted), while others show weaker or negative correlations.
- Diagonal plots represent the distribution of each variable.

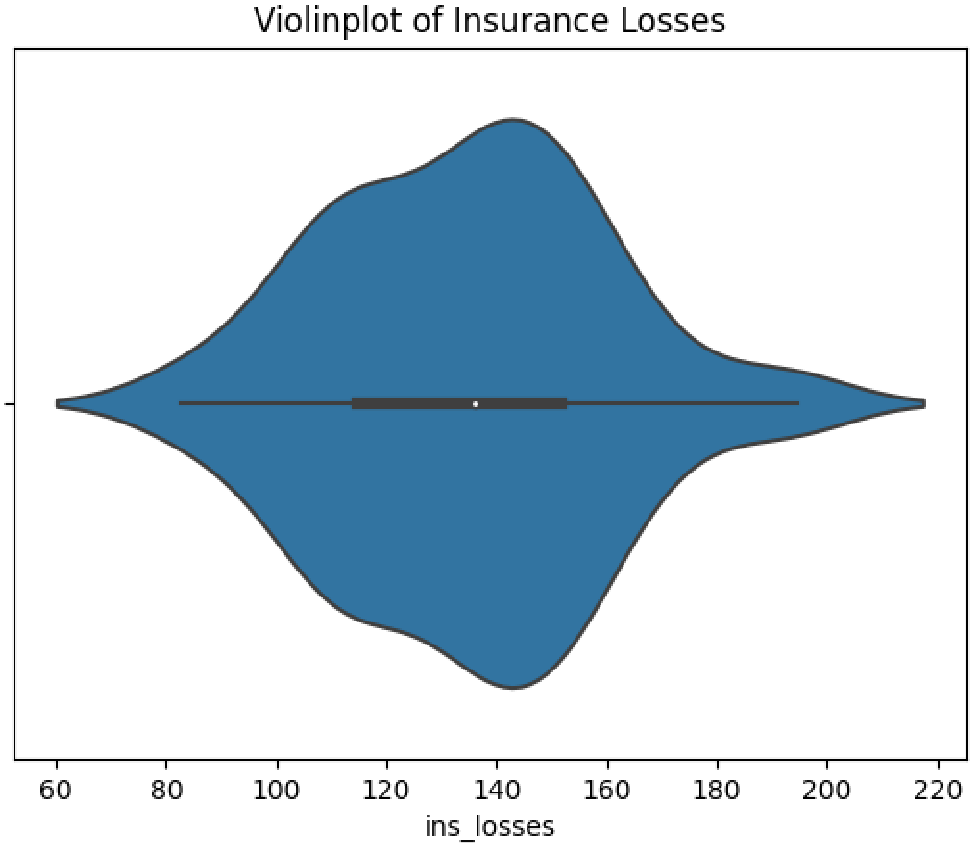
[16]:
```
sns.boxplot(x="alcohol", data=df)
plt.title("Boxplot of Alcohol")
plt.show()
```



Inference:

- This boxplot shows the distribution of alcohol involvement in car crashes.
- It identifies potential outliers and shows that alcohol involvement tends to be higher in some cases.

[19]:
```
sns.violinplot(x="ins_losses", data=df)
plt.title("Violinplot of Insurance Losses")
plt.show()
```

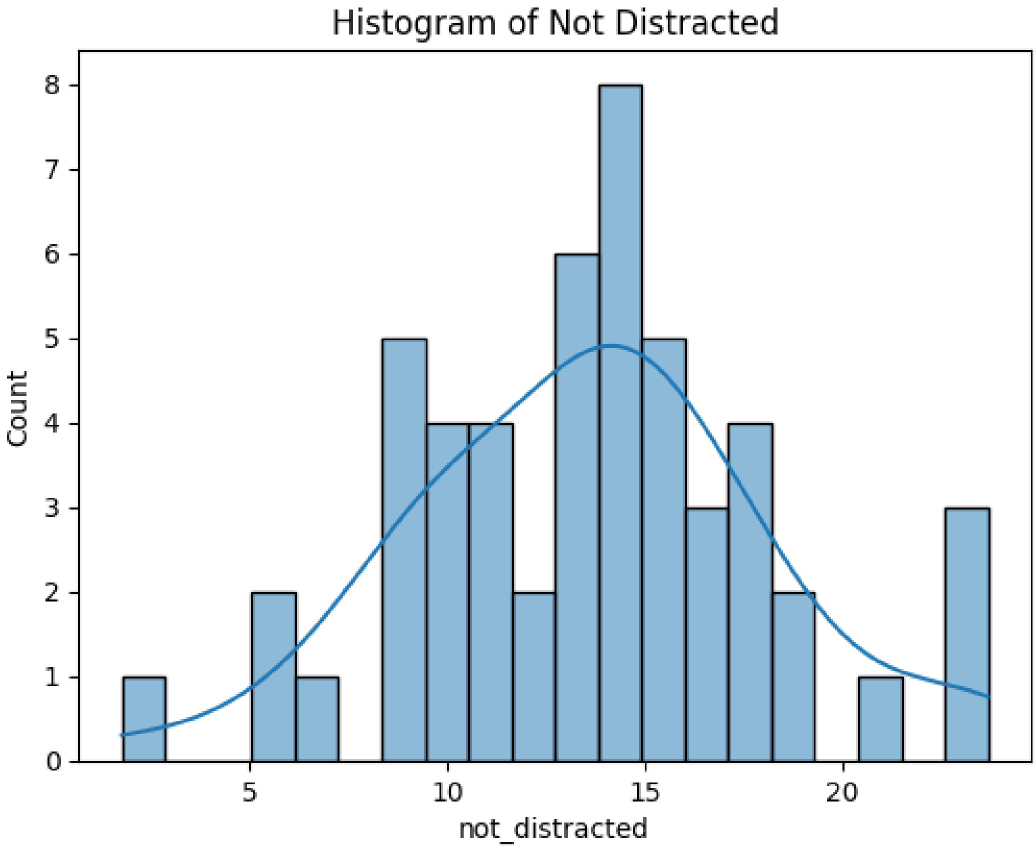


Inference:

- The violinplot reveals the distribution of insurance losses.
- It indicates that most insurance losses are concentrated around lower values, with some variability.

[20]:

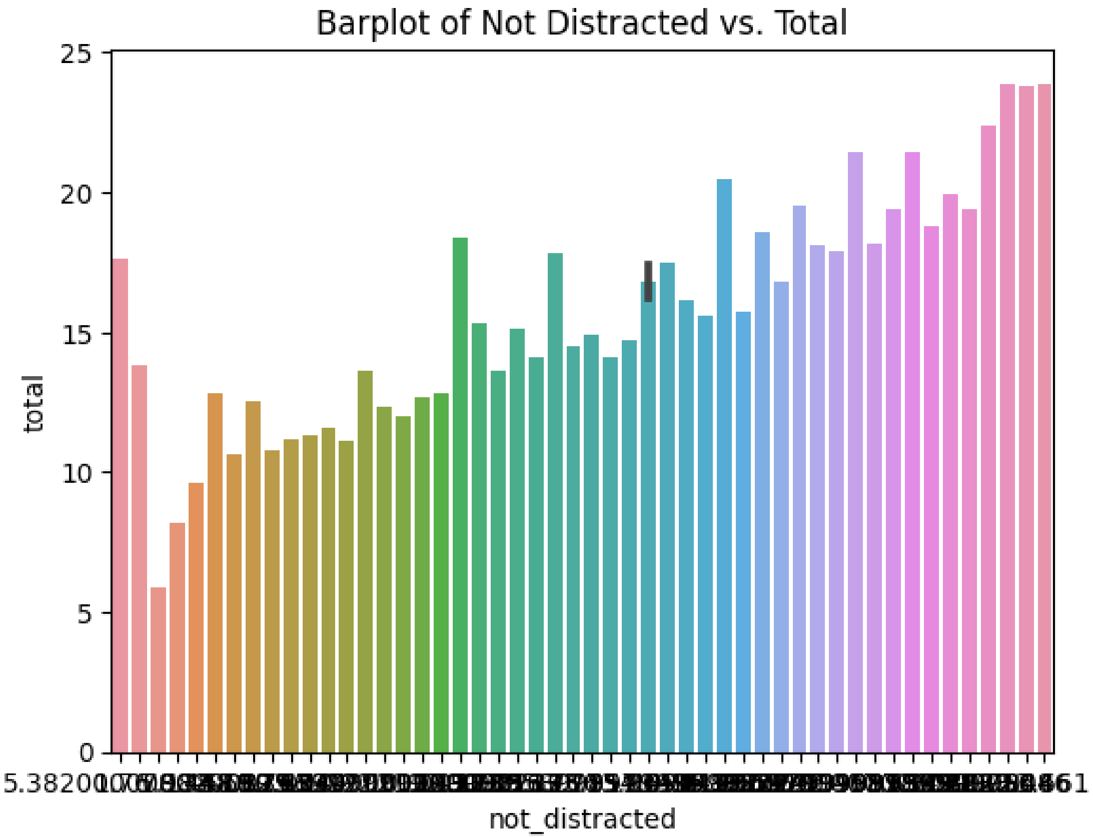
```
sns.histplot(data=df, x="not_distracted", bins=20, kde=True)
plt.title("Histogram of Not Distracted")
plt.show()
```



Inference:

- The histogram shows the distribution of the “not_distracted” variable.
- It suggests that a majority of car crashes involve a relatively low percentage of drivers not being distracted.

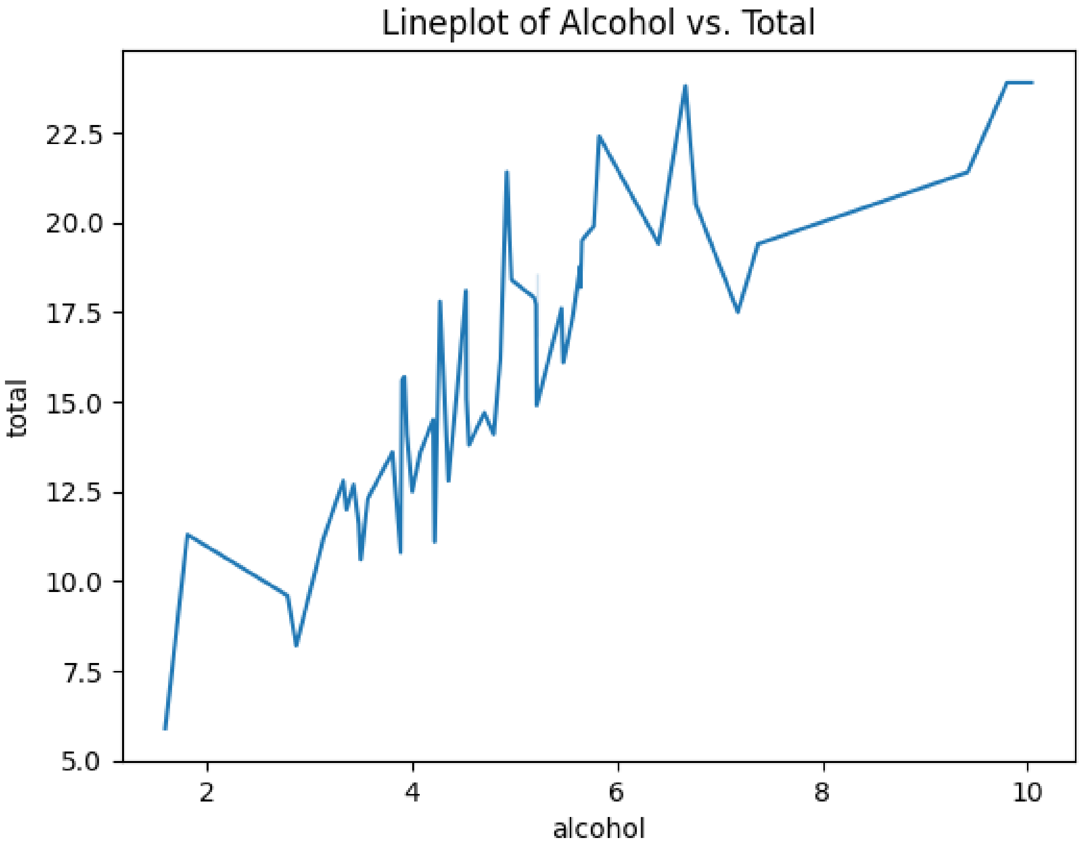
[22]:
```
sns.barplot(x="not_distracted", y="total", data=df)
plt.title("Barplot of Not Distracted vs. Total")
plt.show()
```



Inference:

- This barplot compares the average total crashes for different levels of not distraction.
- It suggests that, on average, there isn't a significant difference in total crashes based on the not distraction level.

[24]:
```
sns.lineplot(x="alcohol", y="total", data=df)
plt.title("Lineplot of Alcohol vs. Total")
plt.show()
```

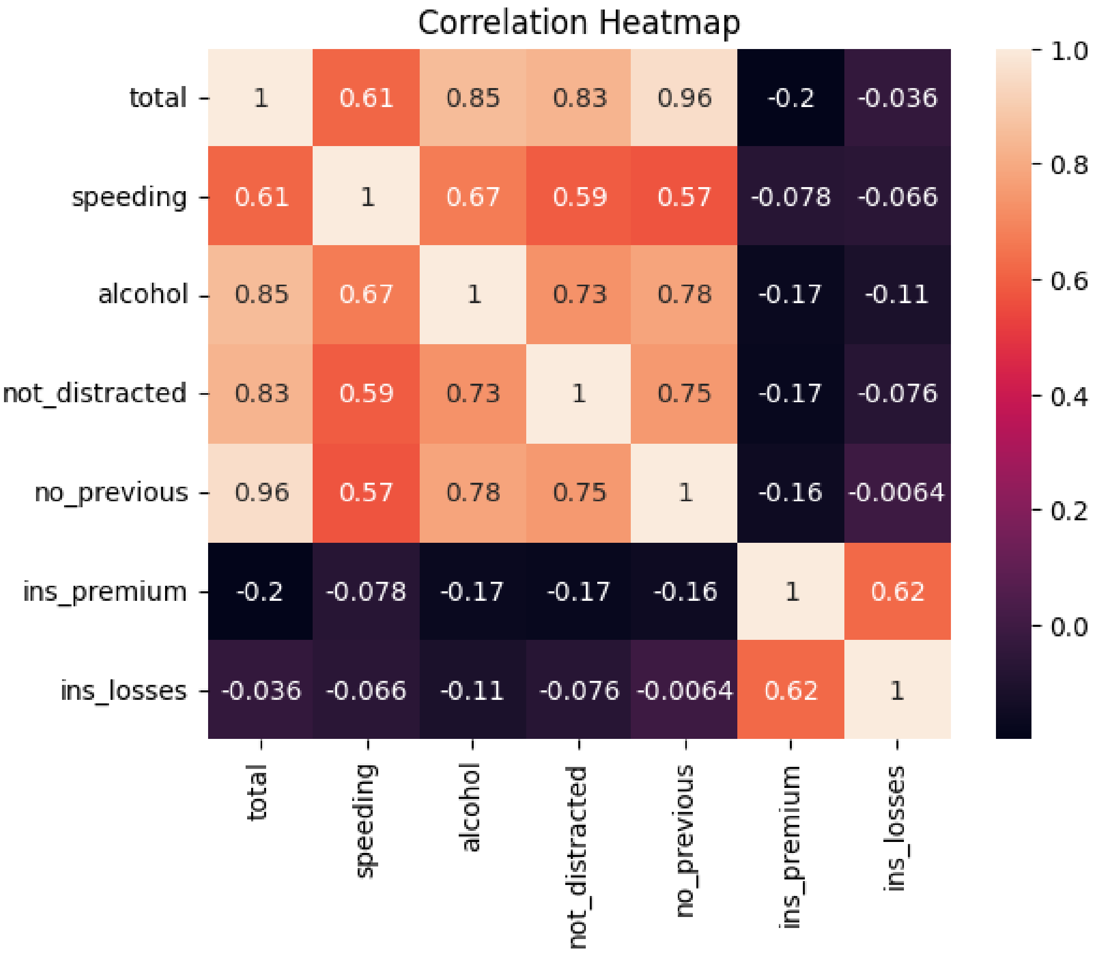


Inference

- The lineplot shows how the total number of crashes changes concerning alcohol involvement. crashes.
- It implies a potential positive relationship between alcohol involvement and total

[26]:
```
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True)
plt.title("Correlation Heatmap")
plt.show()
```

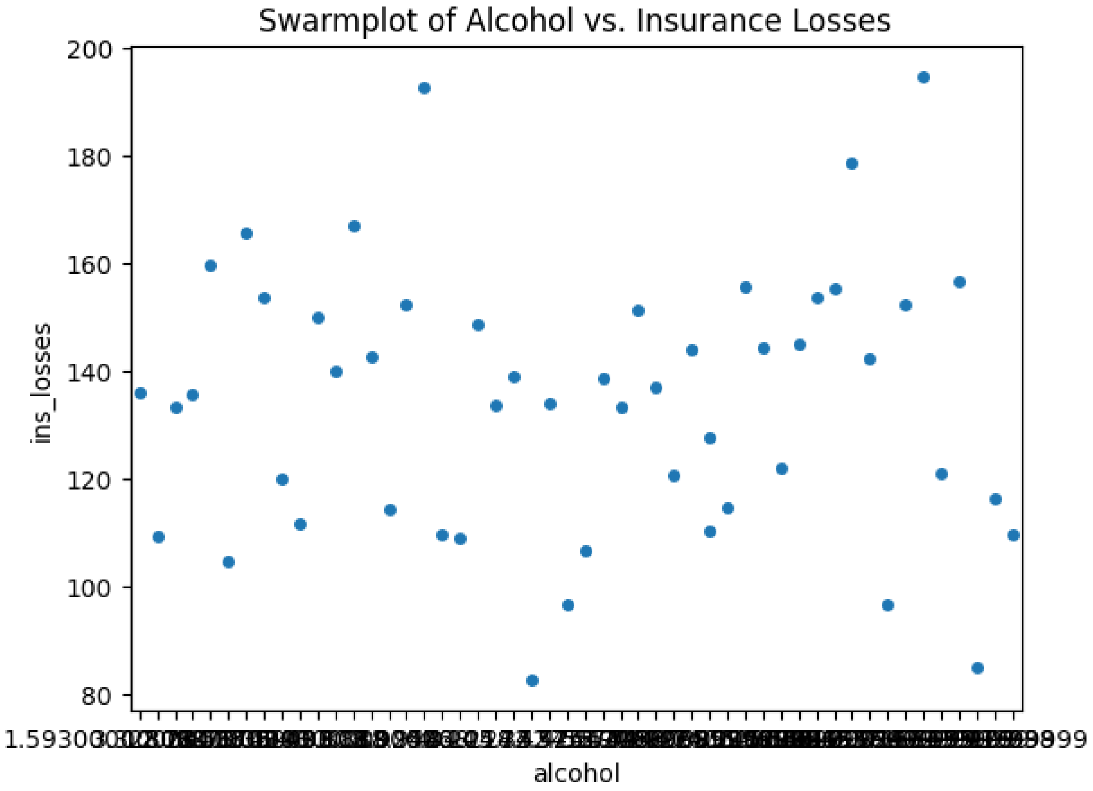
```
<ipython-input-26-525fba809d44>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr_matrix = df.corr()
```



Inference:

- The heatmap displays the correlation matrix between numerical variables.
- It provides a visual representation of the relationships between variables, where darker colors indicate stronger correlations.
- some pairs like total vs no_previous total vs alcohol which are in light color are highly correlated.

[31]:
```
sns.swarmplot(x="alcohol", y="ins_losses", data=df)
plt.title("Swarmplot of Alcohol vs. Insurance Losses")
plt.show()
```

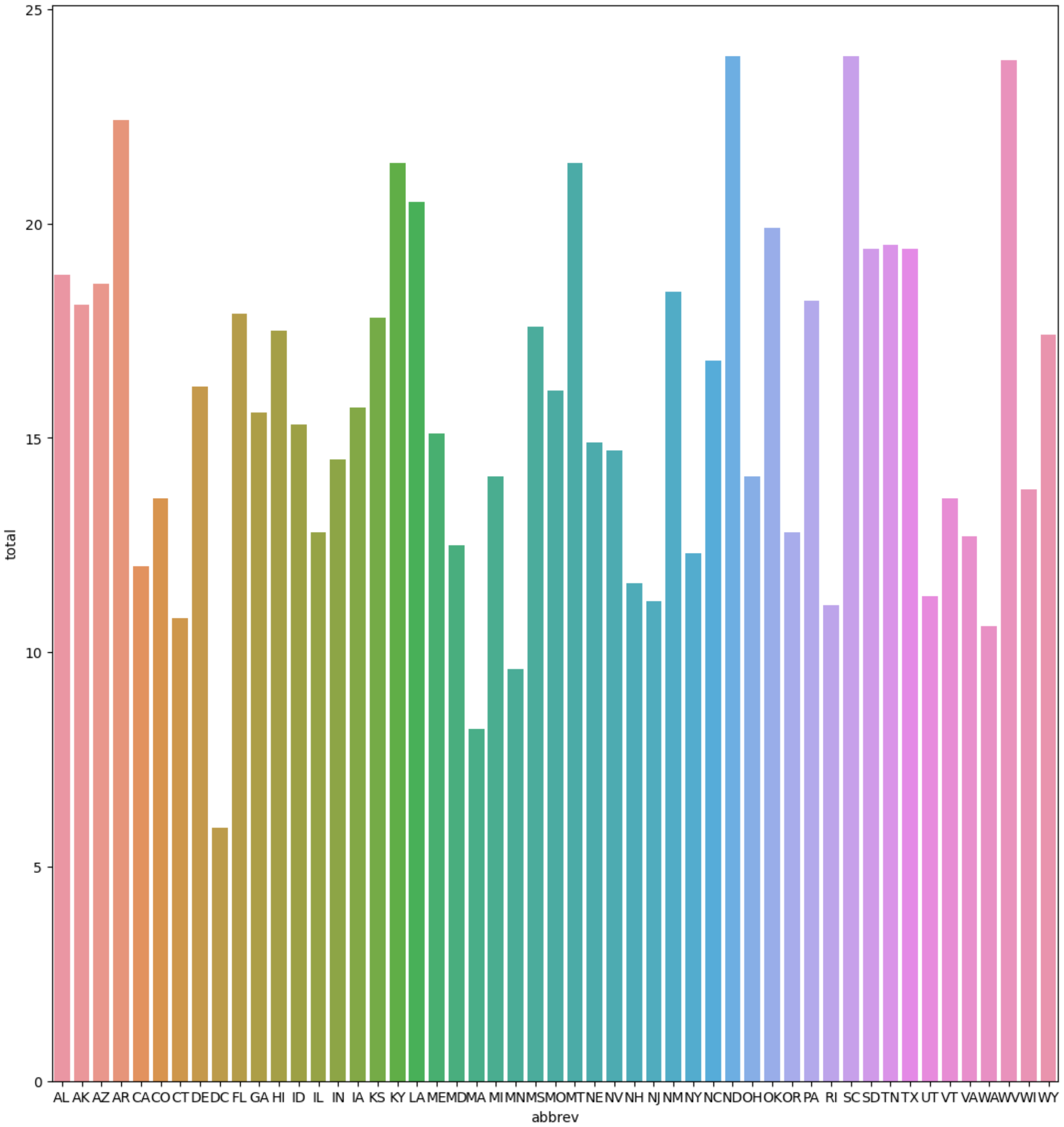


Inference:

- The swarmplot showcases the distribution of insurance losses concerning alcohol involvement.
- It reveals that insurance losses are lower when alcohol is not involved.

[39]:

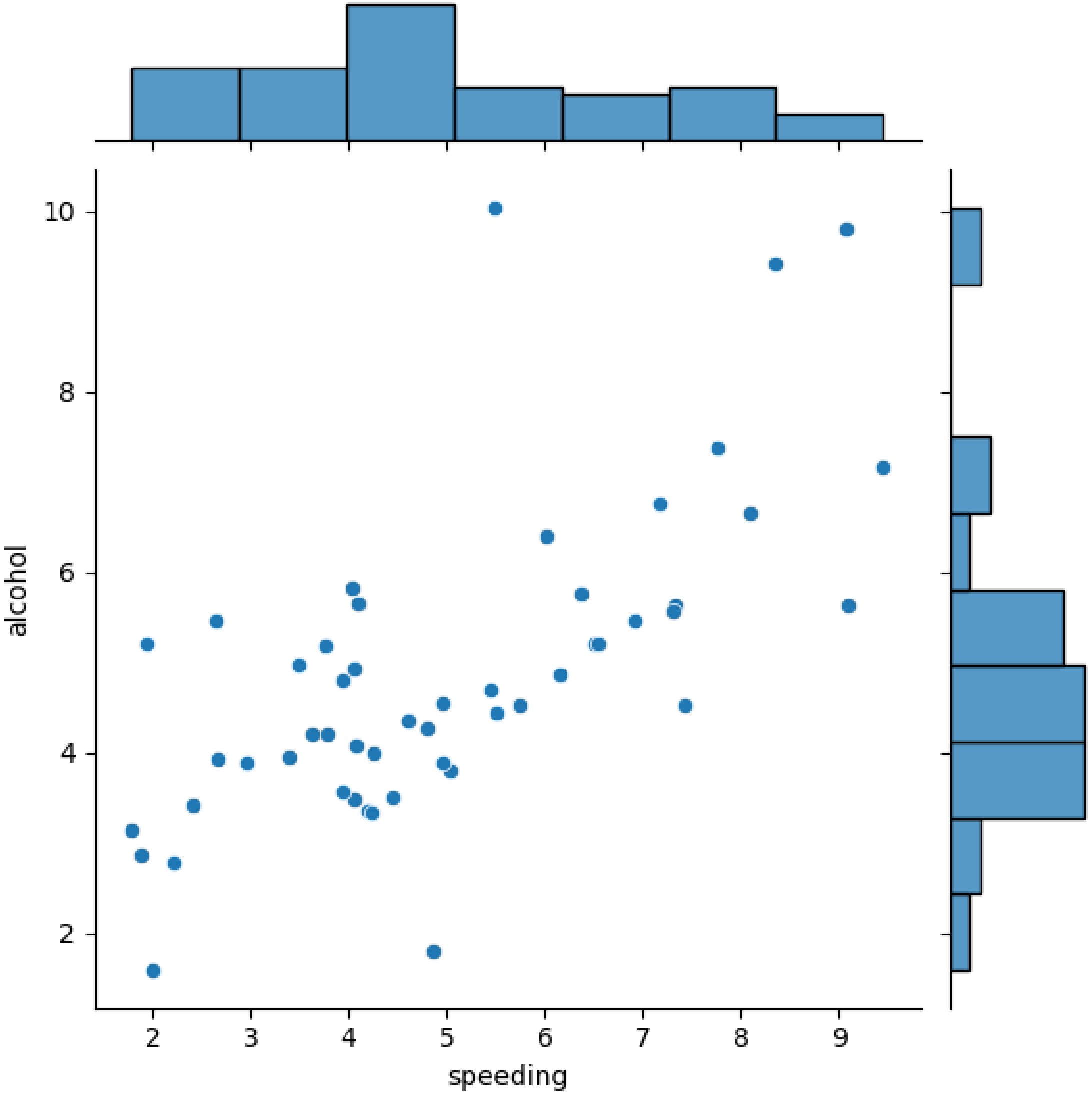
```
plt.figure(figsize=(13,14))
sns.barplot(x="abbrev",y="total",data=df)
plt.show()
```



Inference:

- The barplot shows the total of each abbreviation in the dataset.
- From the barplot we can infer that ND,SC and WV states have hightest total accidents.

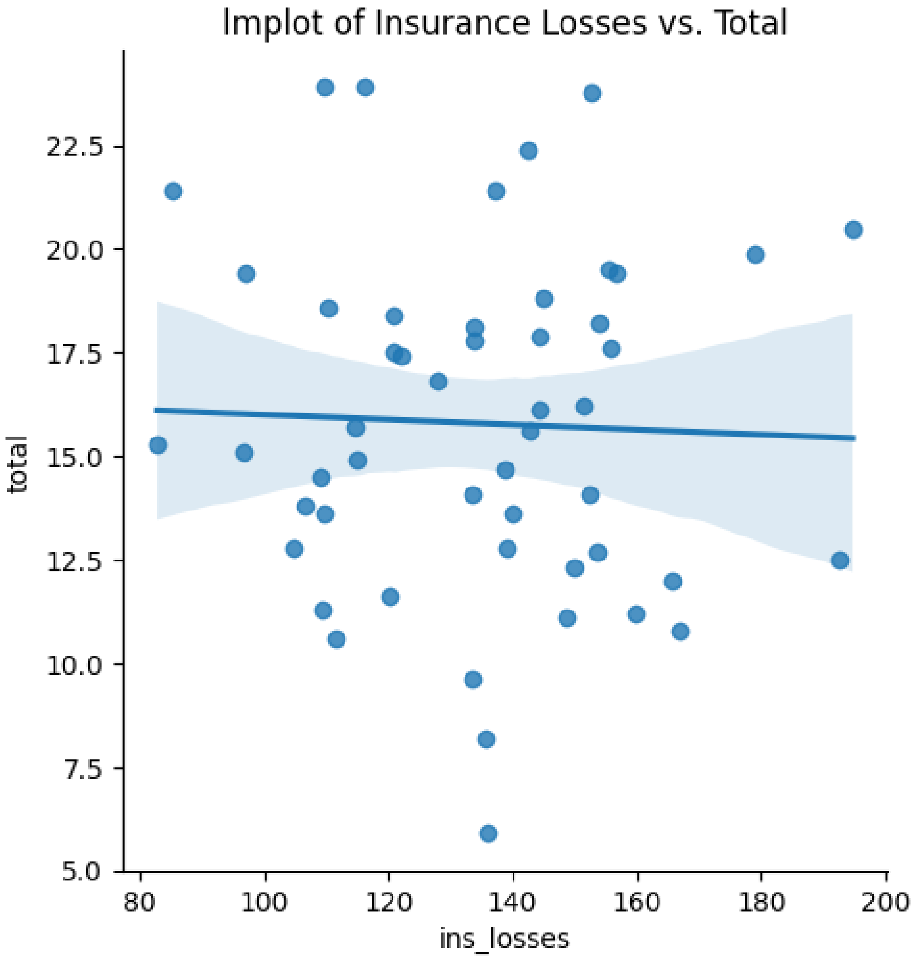
[46]:
```
sns.jointplot(x="speeding", y="alcohol", data=df)
plt.show()
```



Inference:

- The jointplot illustrates the relationship between speeding and alcohol involvement.
- It uses scatterplot to display the density of data points.

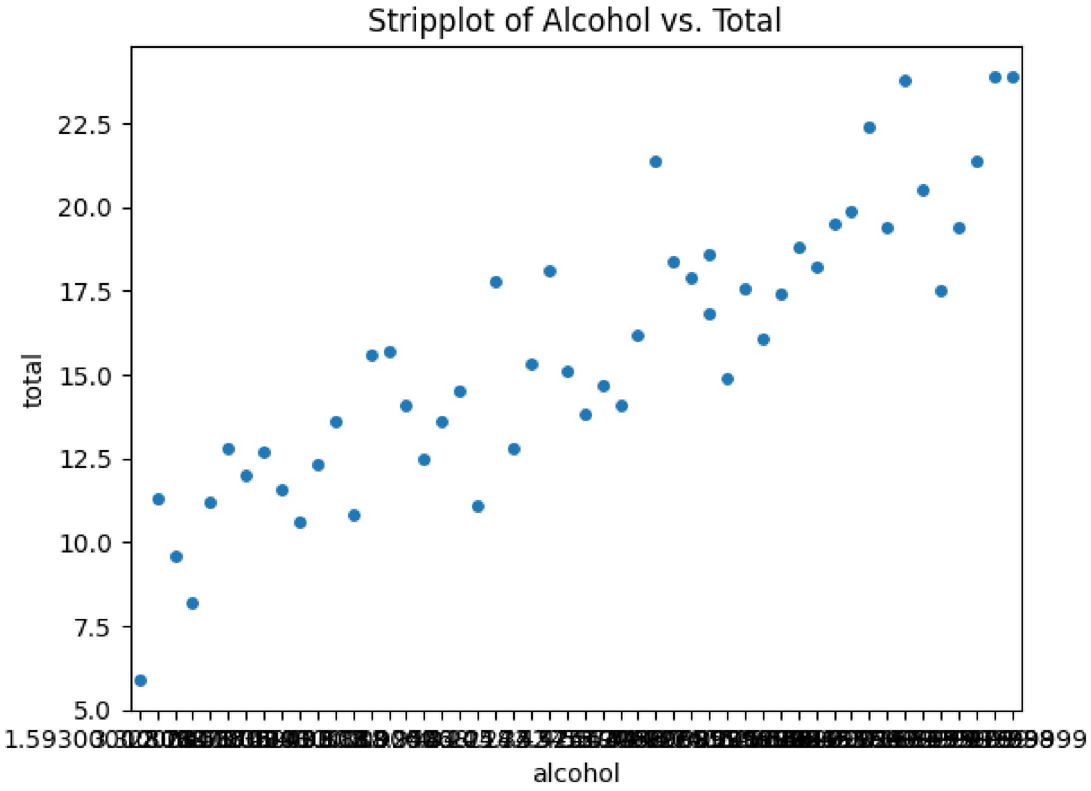
[35]:
```
sns.lmplot(x="ins_losses", y="total", data=df)
plt.title("lmplot of Insurance Losses vs. Total")
plt.show()
```



Inference:

- The lmplot adds a linear regression line to explore the relationship between insurance losses and total crashes.
- It suggests a potential positive correlation between the two variables.

[45]:
```
sns.stripplot(x="alcohol", y="total", data=df, jitter=True)
plt.title("Stripplot of Alcohol vs. Total")
plt.show()
```



Inference:

- The stripplot displays individual data points for alcohol involvement and total crashes.
- It highlights the distribution of data and potential trends.