seaborn

September 14, 2023

[ ]:
```python
import seaborn as sns
print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glue',
'healthexp', 'iris', 'mpg', 'penguins', 'planets', 'seaice', 'taxis', 'tips',
'titanic']
```

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

[ ]:

	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
```

[ ]:
```
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
```

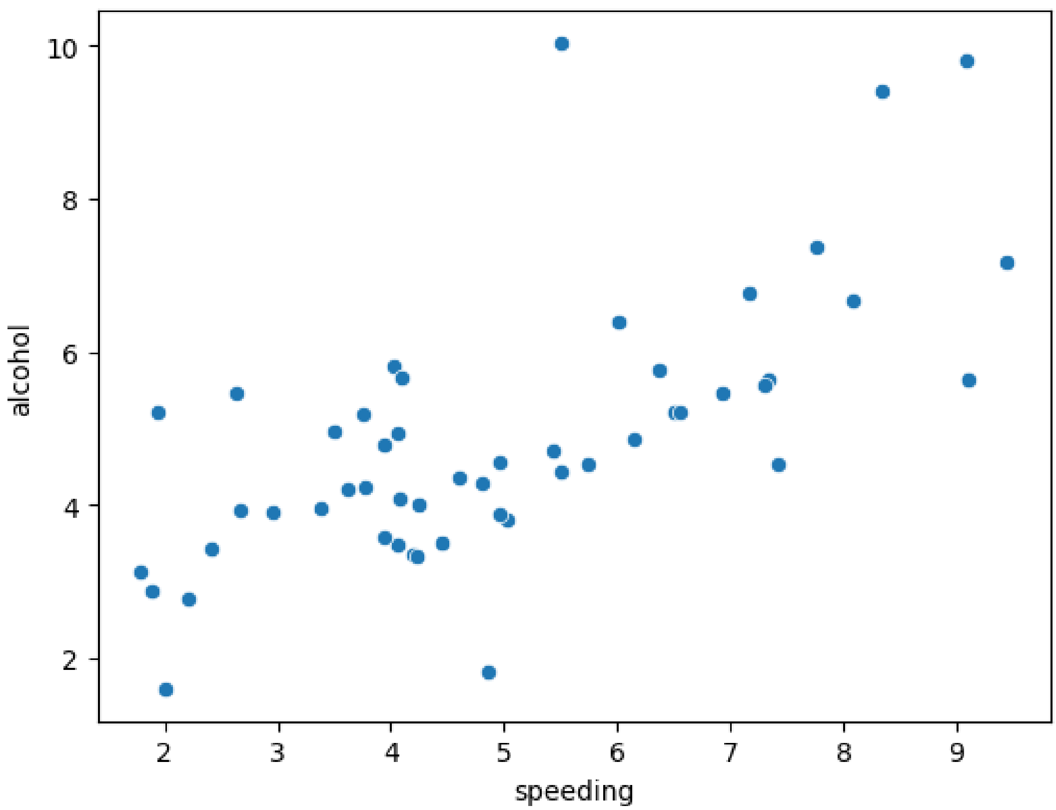
```
dtypes: float64(7), object(1)
memory usage: 3.3+ KB
```

[ ]:
```
df.head(5)
```

[ ]:
```
   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
```

[ ]:
```
sns.scatterplot(x = "speeding" , y = "alcohol" , data = df)
```

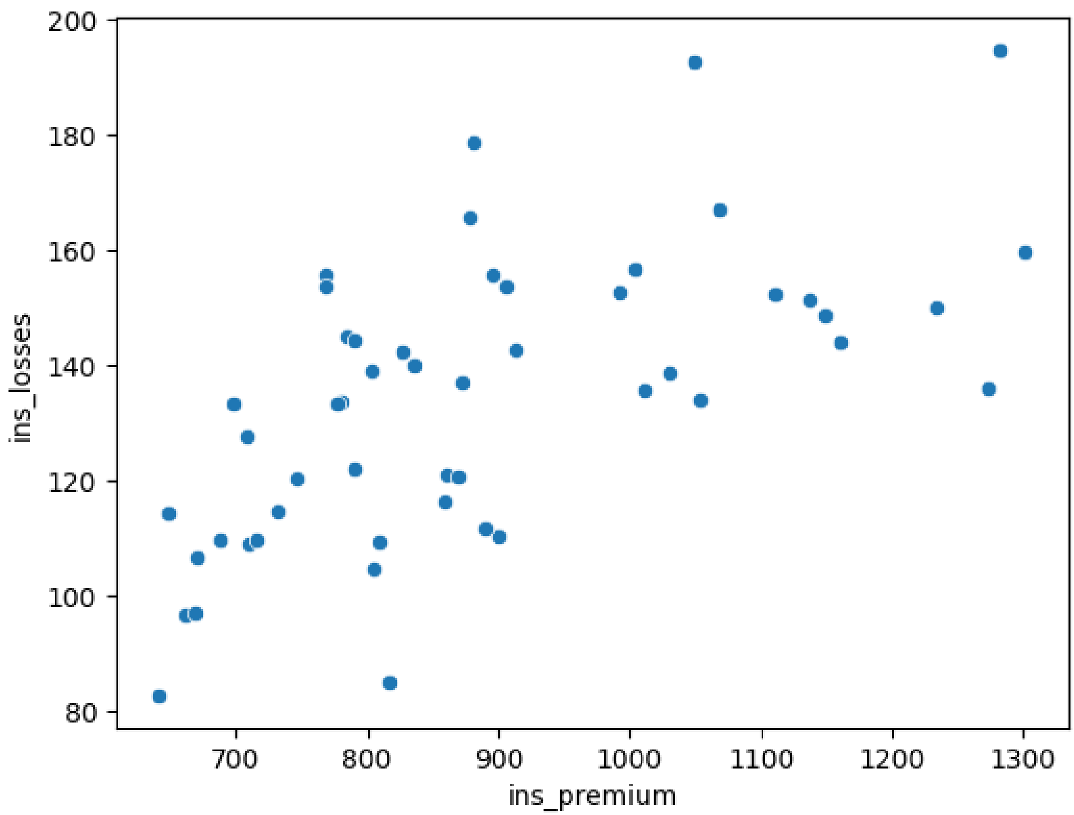
[ ]: <Axes: xlabel='speeding', ylabel='alcohol'>



[ ]:
```
Inference : From the plot we can say that as speeding increases alcohol is also␣
↪increasing . They are directly proportional .
```

[ ]:
```
import seaborn as sns
sns.scatterplot(x = "ins_premium" , y = "ins_losses" , data = df)
#Inference
#The scatter plot between "Insurance Premiums (ins_premium)" and "Insurance␣
↪Losses (ins_losses)" from the given dataset (df) shows a positive linear␣
↪relationship.
#As insurance premiums increase, there is a tendency for insurance losses to␣
↪also increase,
# suggesting that states with higher premiums may experience higher losses,␣
↪possibly due to increased risk or other factors.
```

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

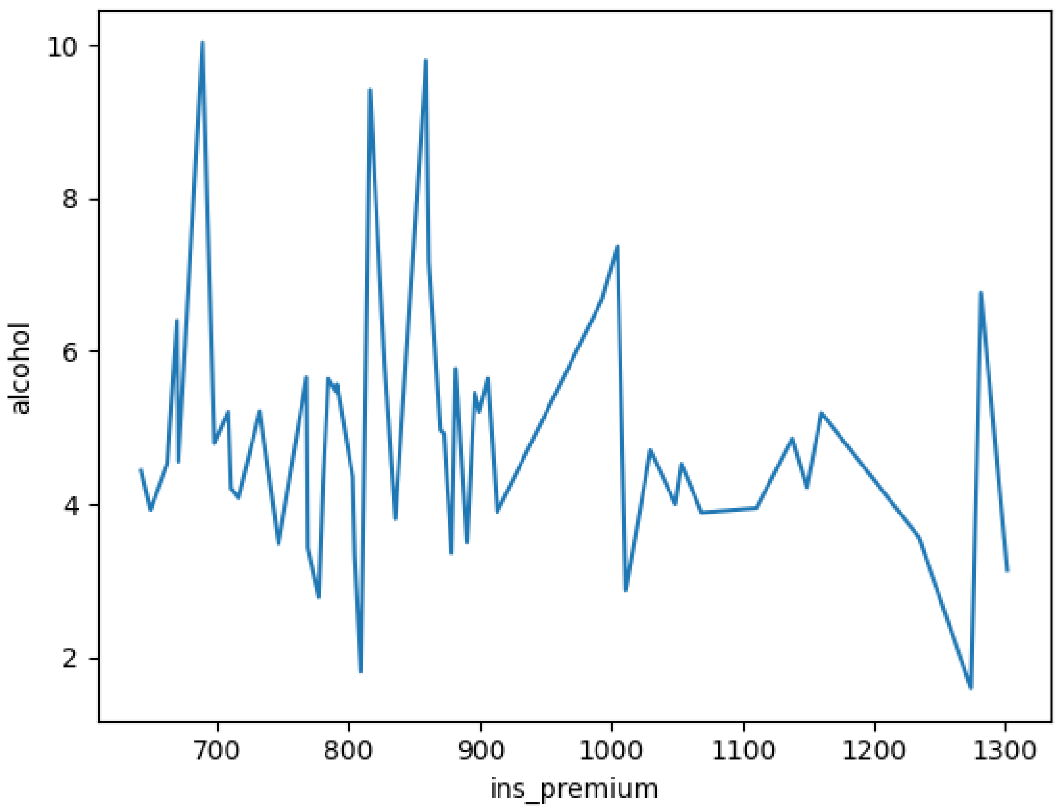


[ ]:
```
sns.lineplot(x = "ins_premium" , y = "alcohol" , data = df , ci = None)
#Inference
```

```
#The lineplot, created using sns.lineplot(x="ins_premium", y="alcohol",␣
↪data=df, ci=None), indicates that there is a general trend of increasing␣
↪alcohol consumption with higher insurance premiums.
#However, the absence of confidence intervals (ci=None) suggests that the plot␣
↪does not account for uncertainty,
#and further statistical analysis may be needed to confirm the significance of␣
↪this relationship.
```

```
<ipython-input-19-e5661d87ab13>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(x = "ins_premium" , y = "alcohol" , data = df , ci = None)
```

[ ]: <Axes: xlabel='ins_premium', ylabel='alcohol'>

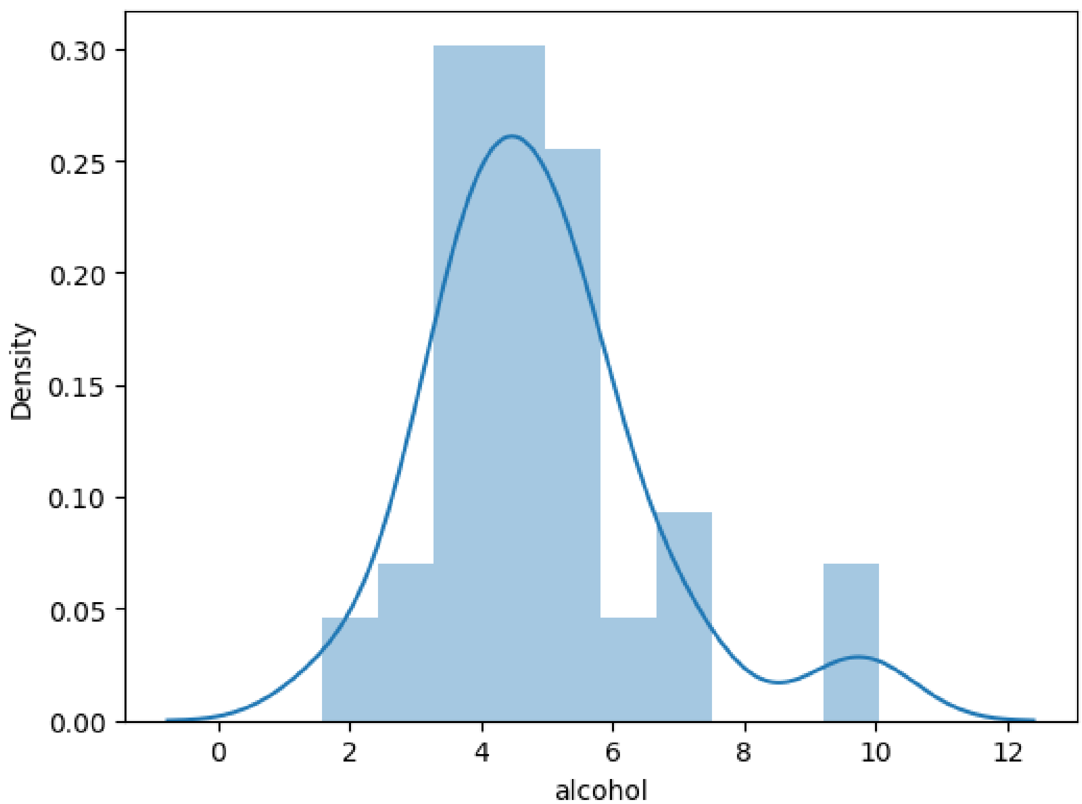


[ ]:
```
sns.distplot(df["alcohol"])
#Inference
#The distribution plot (sns.distplot) of the "alcohol" variable in the dataset␣
↪suggests that
```

```
#the majority of states have a relatively low average alcohol consumption among␣
↪car crash incidents,
#with a peak around the lower values. However, there is a noticeable␣
↪right-skew, indicating a few states with higher alcohol consumption and␣
↪potentially higher crash rates,
#warranting further investigation into the relationship between alcohol␣
↪consumption and car accidents.
```

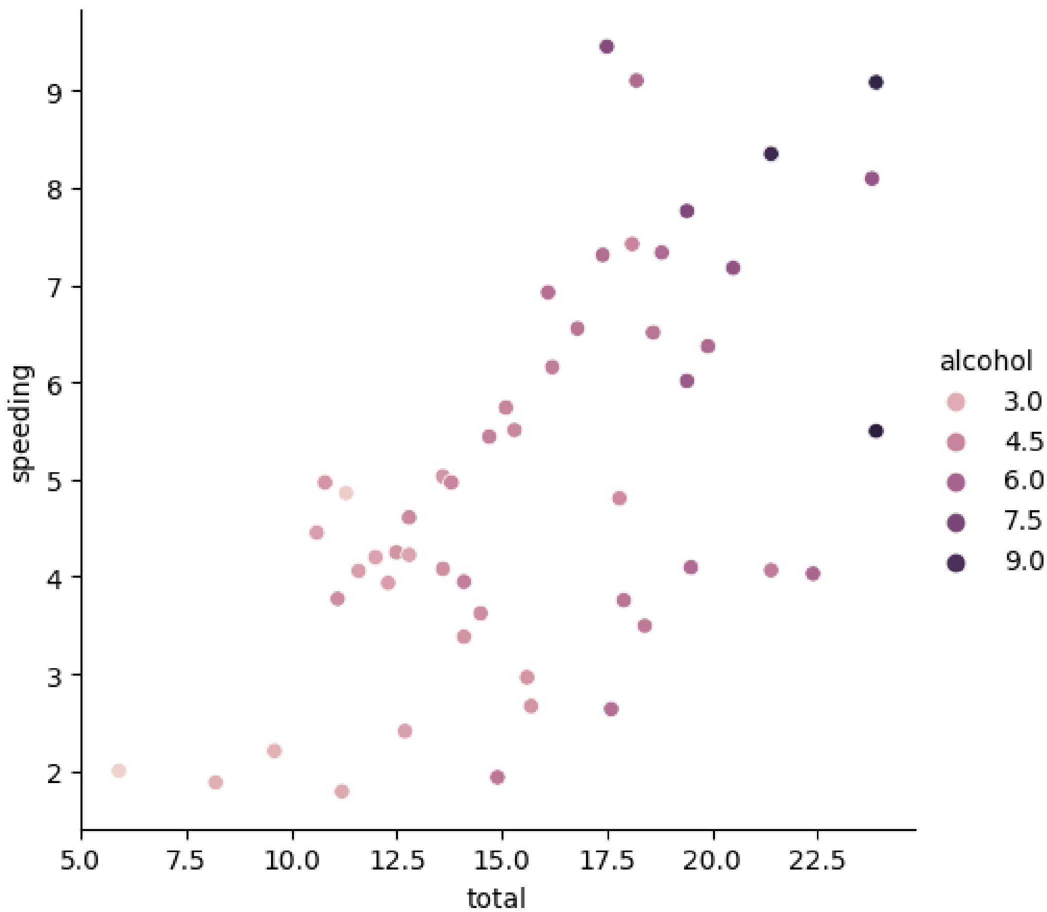
```
<ipython-input-20-281d56044cde>:1: UserWarning:

`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"])
```

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



[ ]:
```
sns.relplot(x = "total" , y = "speeding" , data = df ,hue = "alcohol")
#Inference
#From the plot, it can be inferred that there is a noticeable pattern where␣
↪higher levels of alcohol consumption (alcohol) are associated with an␣
↪increase
#in both the total number of car crashes (total) and crashes involving speeding␣
↪(speeding).
#This suggests a potential correlation between alcohol consumption and unsafe␣
↪driving behavior.
```

[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb3a7135ea0>



[ ]:
```
df["ins_losses"].value_counts()
```

[ ]:
```
145.08    1
153.86    1
138.71    1
120.21    1
```

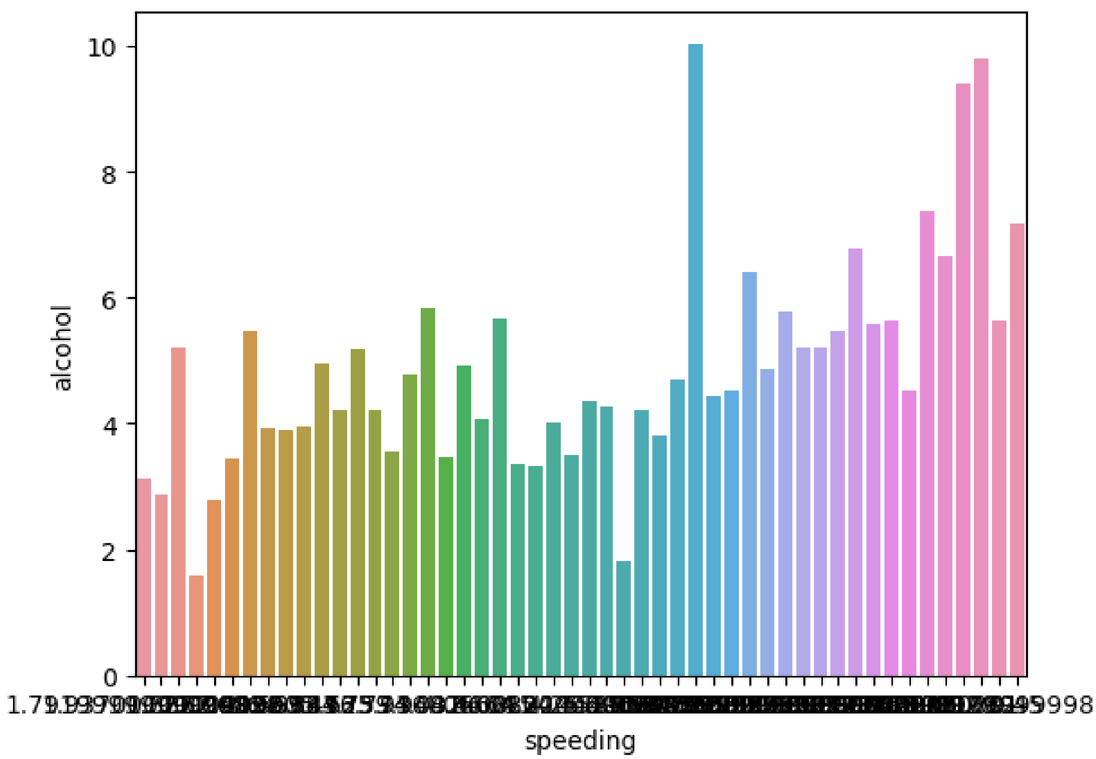
159.85	1
120.75	1
150.01	1
127.82	1
109.72	1
133.52	1
178.86	1
104.61	1
148.58	1
85.15	1
116.29	1
96.87	1
155.57	1
156.83	1
109.48	1
109.61	1
153.72	1
111.62	1
152.56	1
106.62	1
114.82	1
144.45	1
133.93	1
82.75	1
110.35	1
142.39	1
165.63	1
139.91	1
167.02	1
151.48	1
136.05	1
144.18	1
142.80	1
120.92	1
139.15	1
155.77	1
108.92	1
114.47	1
133.80	1
137.13	1
194.78	1
96.57	1
192.70	1
135.63	1
152.26	1
133.35	1
122.04	1

```
Name: ins_losses, dtype: int64
```

[ ]:
```python
df.head()
sns.barplot(data=df, x="speeding" , y = "alcohol" , ci = None)
#Inference
#The sns.barplot with `x="speeding"` and `y="alcohol"` (without confidence intervals) suggests that there might not be a strong linear relationship between the percentage of car crashes
#involving speeding and the percentage involving alcohol consumption across the dataset.
# However, further statistical analysis is needed to determine if there is any significant correlation or pattern in the data.
```

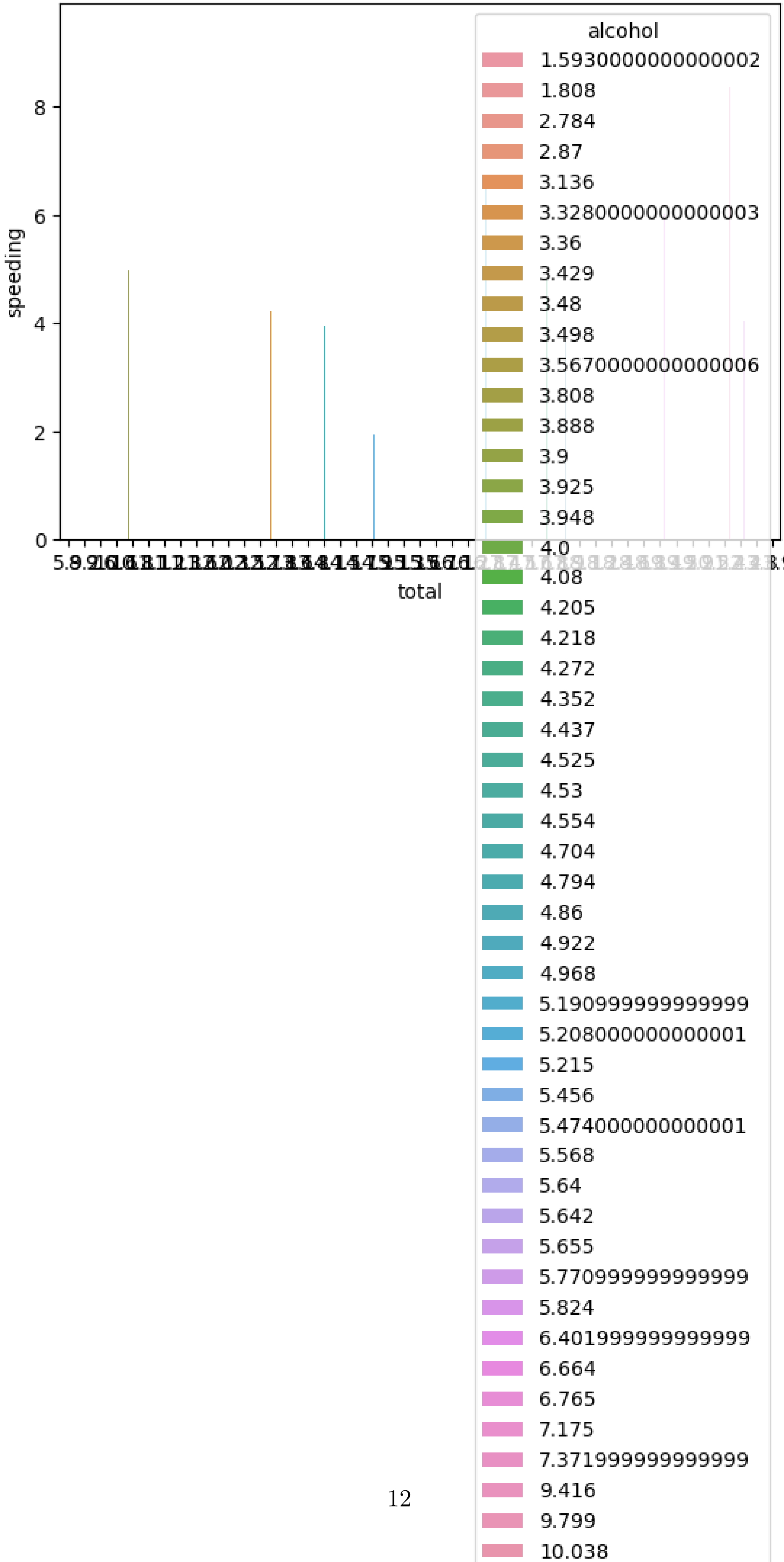
```
<ipython-input-31-8c89cb3a3fef>:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(data=df, x="speeding" , y = "alcohol" , ci = None)
```

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



[ ]:
```
sns.barplot(data=df, x="total" , y = "speeding" , hue = "alcohol")
df.head()
#Inference
#The barplot indicates a positive relationship between the total number of car␣
↪crashes and those involving speeding.
#The hue parameter separates the bars by alcohol involvement, showing the␣
↪influence of alcohol on speeding-related accidents.
```

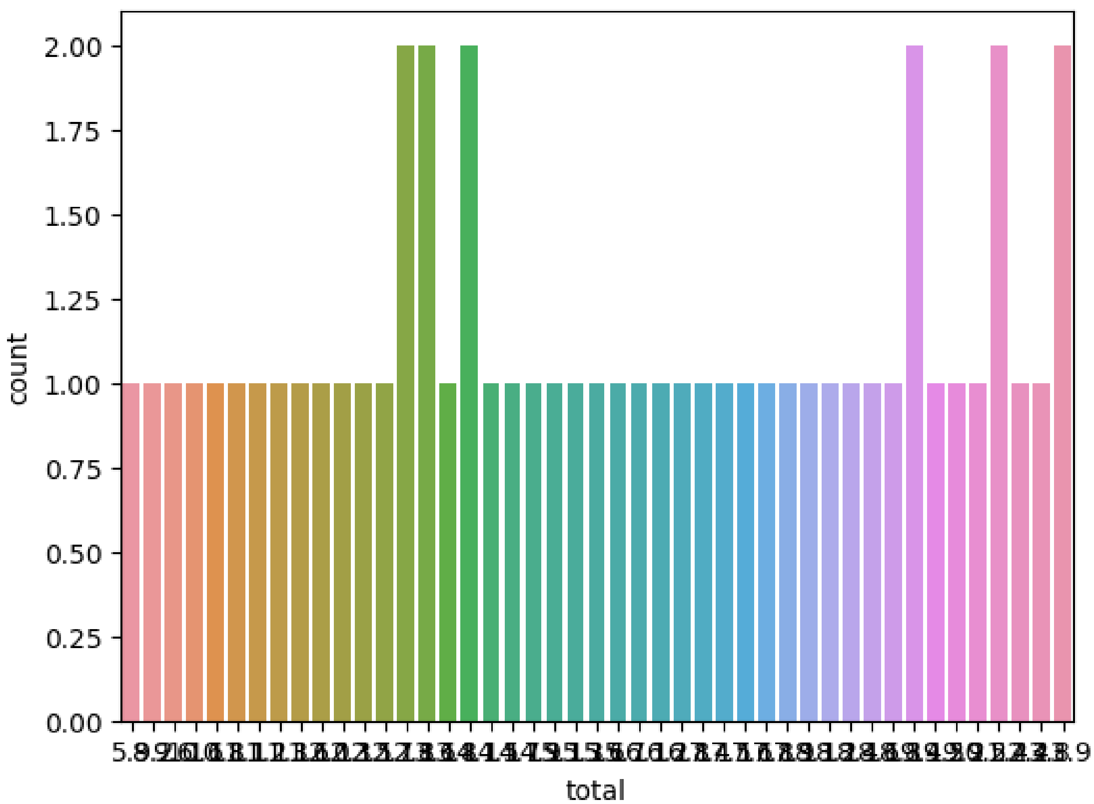
[ ]:
```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium  \
0   18.8     7.332    5.640          18.048       15.040       784.55
1   18.1     7.421    4.525          16.290       17.014      1053.48
2   18.6     6.510    5.208          15.624       17.856       899.47
3   22.4     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
```



alcohol
1.5930000000000002
1.808
2.784
2.87
3.136
3.3280000000000003
3.36
3.429
3.48
3.498
3.5670000000000006
3.808
3.888
3.9
3.925
3.948
4.0
4.08
4.205
4.218
4.272
4.352
4.437
4.525
4.53
4.554
4.704
4.794
4.86
4.922
4.968
5.190999999999999
5.208000000000001
5.215
5.456
5.474000000000001
5.568
5.64
5.642
5.655
5.770999999999999
5.824
6.401999999999999
6.664
6.765
7.175
7.371999999999999
9.416
9.799
10.038
speeding
0
2
4
6
8
total

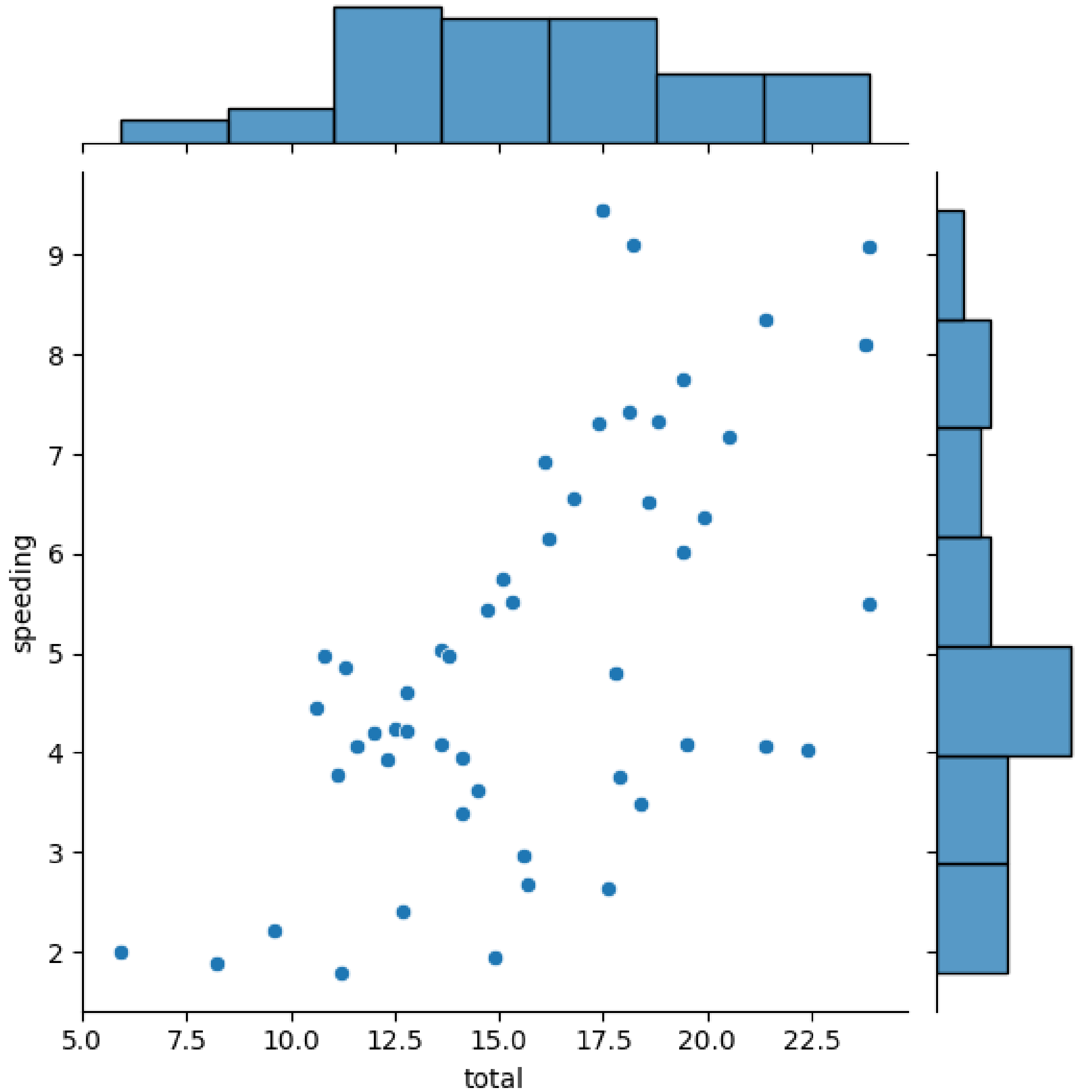
[ ]:
```
sns.countplot(x="total",data=df)
#Inference
# The graph likely represents the distribution of the total number of car crashes across different categories or values of the "total" variable in the dataset.
#This plot can help us see how frequently each value occurs and whether there are any dominant or unusual values in the dataset.
```

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



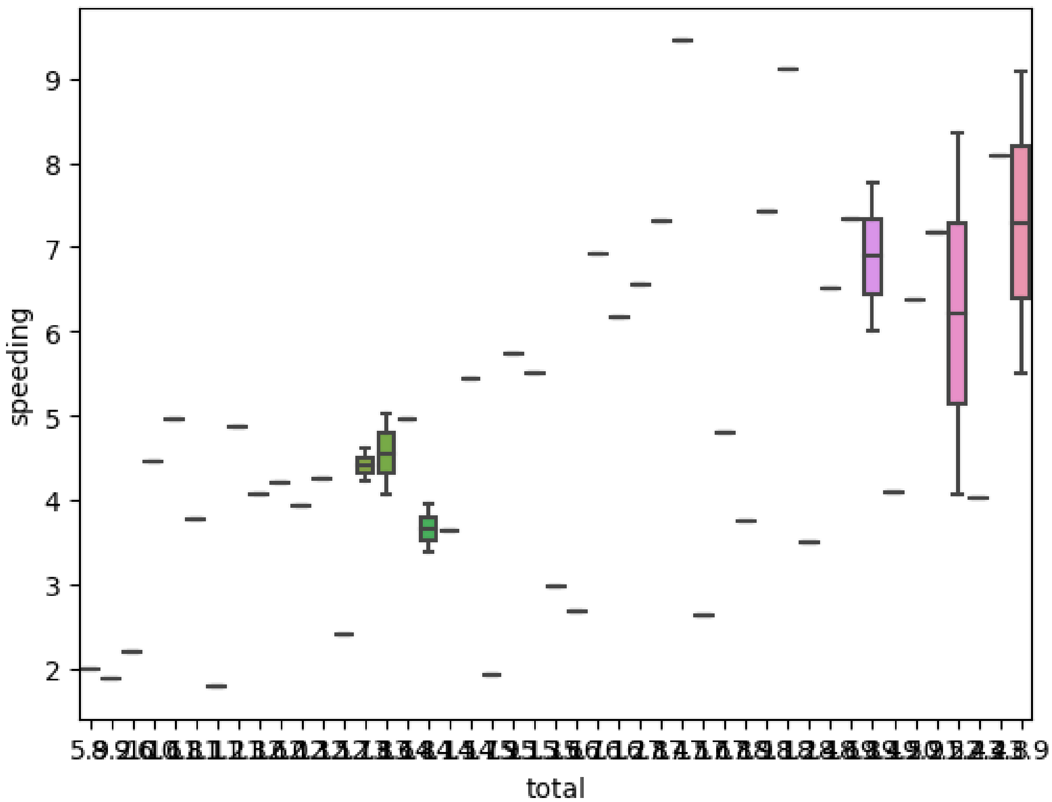
[ ]:
```
sns.jointplot(x="total",y="speeding",data=df)
#Inference
#The jointplot between the "total" number of car crashes and the "speeding" as a contributing factor in the dataset (df) suggests a positive correlation,
# indicating that states with higher total car crashes tend to have a higher number of crashes related to speeding.
#This observation implies that addressing speeding issues may be crucial in reducing overall car accidents in these states.T
```

[ ]: <seaborn.axisgrid.JointGrid at 0x7fb3912059c0>



[ ]:
```
sns.boxplot(x="total",y="speeding",data=df)
# indicates that the median number of speeding-related car crashes doesn't␣
↪significantly change with varying total car crash counts,
#with a few outliers showing high speeding-related incidents in states with␣
↪high overall crash rates.
```

[ ]: <Axes: xlabel='total', ylabel='speeding'>



[ ]:
```
Correlation

positive correlation
negative correlation
neutral correlation
```

[ ]:
```
corr=df.corr()
corr
```

```
<ipython-input-44-7d5195e2bf4d>: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=df.corr()
```

[ ]:

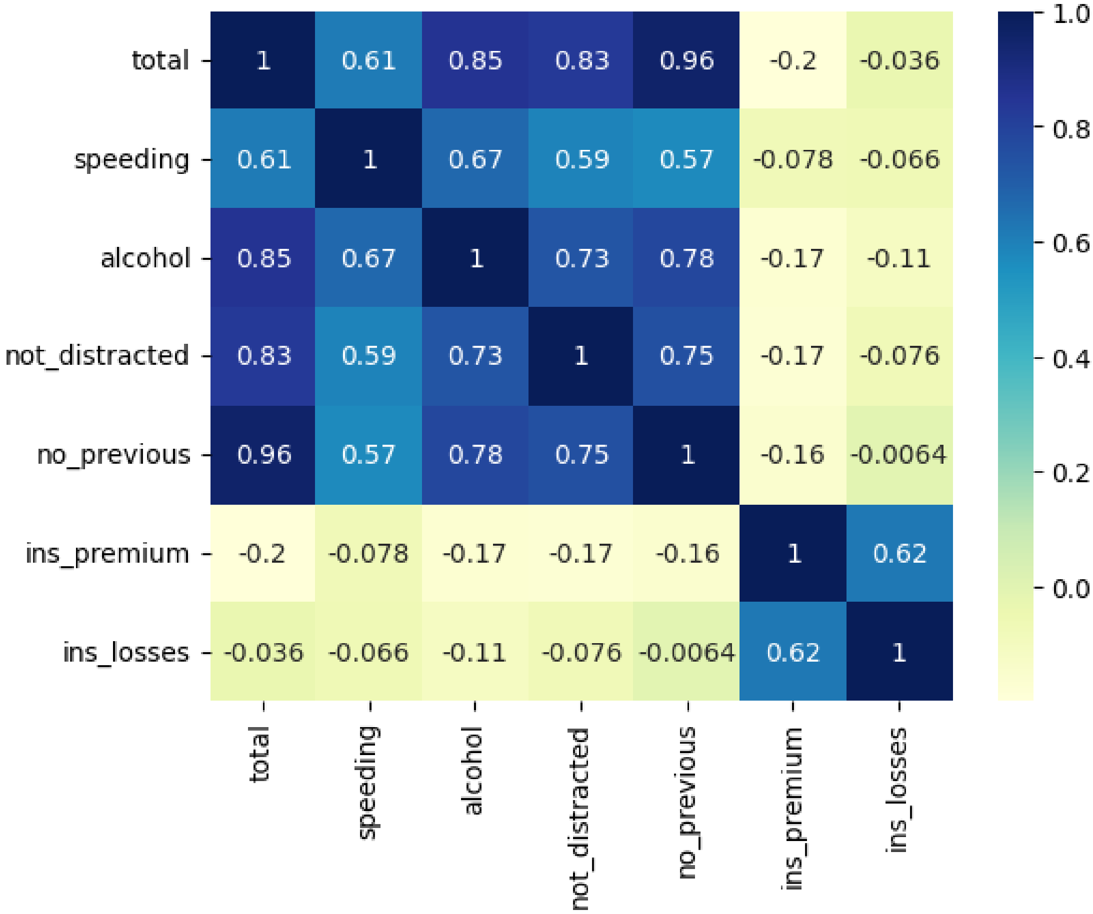
	total	speeding	alcohol	not_distracted	no_previous	\
total	1.000000	0.611548	0.852613	0.827560	0.956179	
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	

```
ins_premium    -0.199702 -0.077675 -0.170612       -0.174856    -0.156895
ins_losses     -0.036011 -0.065928 -0.112547       -0.075970    -0.006359

                ins_premium  ins_losses
total             -0.199702   -0.036011
speeding          -0.077675   -0.065928
alcohol           -0.170612   -0.112547
not_distracted    -0.174856   -0.075970
no_previous       -0.156895   -0.006359
ins_premium        1.000000    0.623116
ins_losses         0.623116    1.000000
```

[ ]:
```
>0.5 is highly correlated
<0.5 is less correlated
```

[ ]:
```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
#The Seaborn heatmap with `annot=True` and the "YlGnBu" colormap is an␣
↪effective visualization for understanding the correlations between variables␣
↪in a dataset.
#The color intensity in the heatmap allows us to quickly identify strong␣
↪positive (dark blue) and negative (light yellow-green) correlations,␣
↪providing valuable insights into which variables are closely related in the␣
↪dataset.
```

[ ]: <Axes: >


total
speeding
alcohol
not_distracted
no_previous
ins_premium
ins_losses
1 0.61 0.85 0.83 0.96 -0.2 -0.036
0.61 1 0.67 0.59 0.57 -0.078 -0.066
0.85 0.67 1 0.73 0.78 -0.17 -0.11
0.83 0.59 0.73 1 0.75 -0.17 -0.076
0.96 0.57 0.78 0.75 1 -0.16 -0.0064
-0.2 -0.078 -0.17 -0.17 -0.16 1 0.62
-0.036 -0.066 -0.11 -0.076 -0.0064 0.62 1
1.0
0.8
0.6
0.4
0.2
0.0