In [2]:
```
import seaborn as sns
```

In [6]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
df = sns.load_dataset('car_crashes')
```

In [8]:
```
df.head(6)
```

Out[8]:

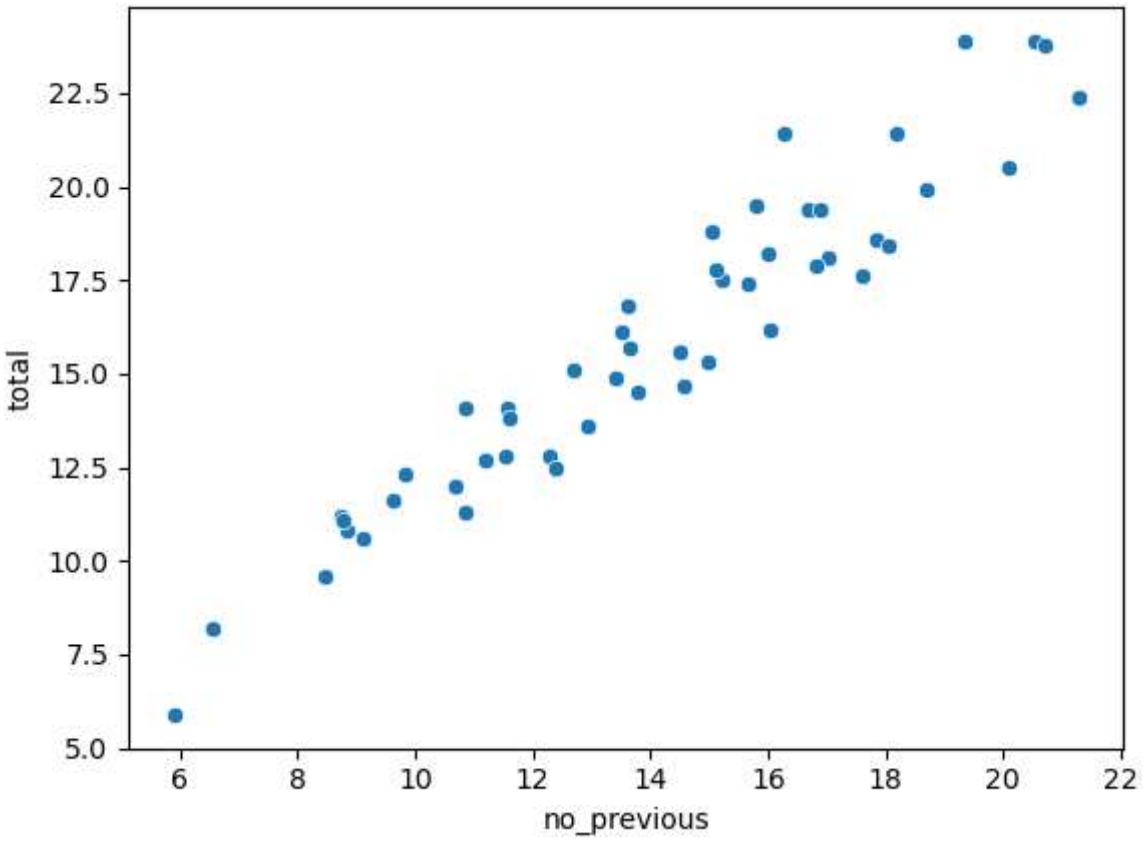
	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	CO

In [9]:
```
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
```

In [94]:
```
sns.scatterplot(x="no_previous",y="total",data=df)
```

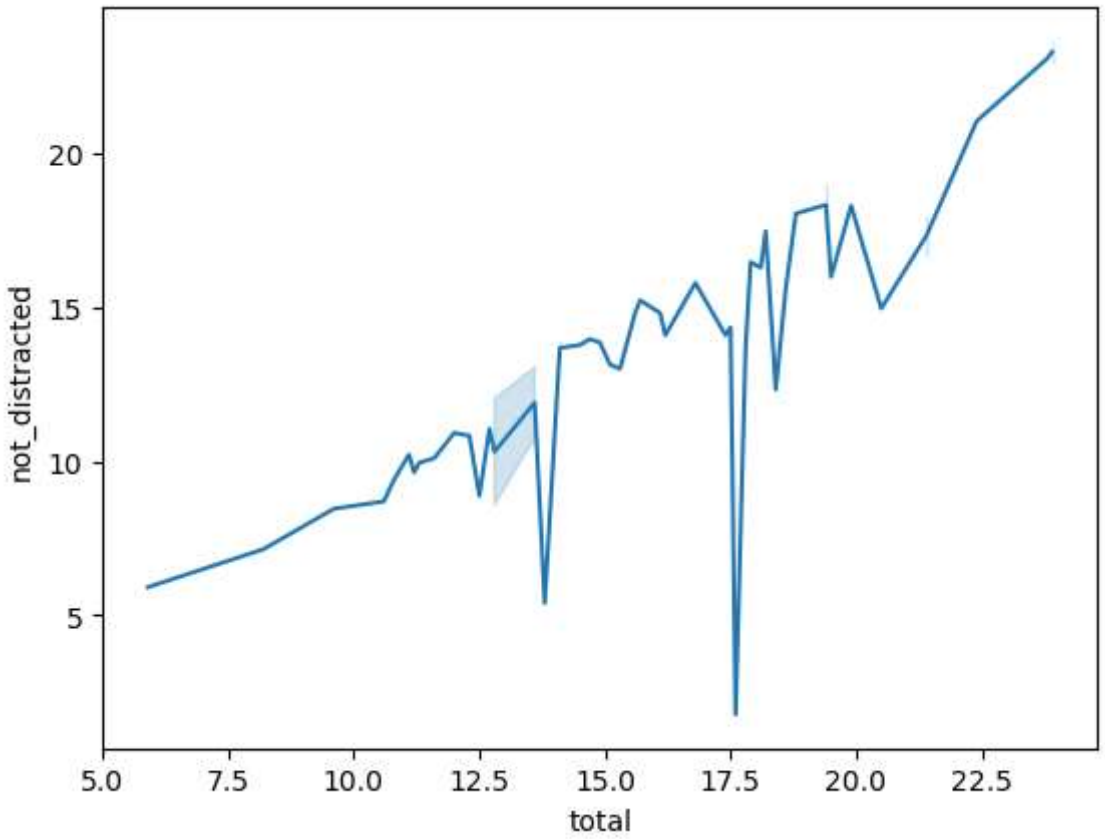
Out[94]:
```
<Axes: xlabel='no_previous', ylabel='total'>
```



In [ ]:
```
# Inference from above scatterplot:
"""
from the plot we can say that as
number of drivers who have never been in accident before increases,
number of car crashes is also increasing.

"""
```

In [108…
```
sns.lineplot(x="total",y="not_distracted",data=df)
```

Out[108]:
```
<Axes: xlabel='total', ylabel='not_distracted'>
```

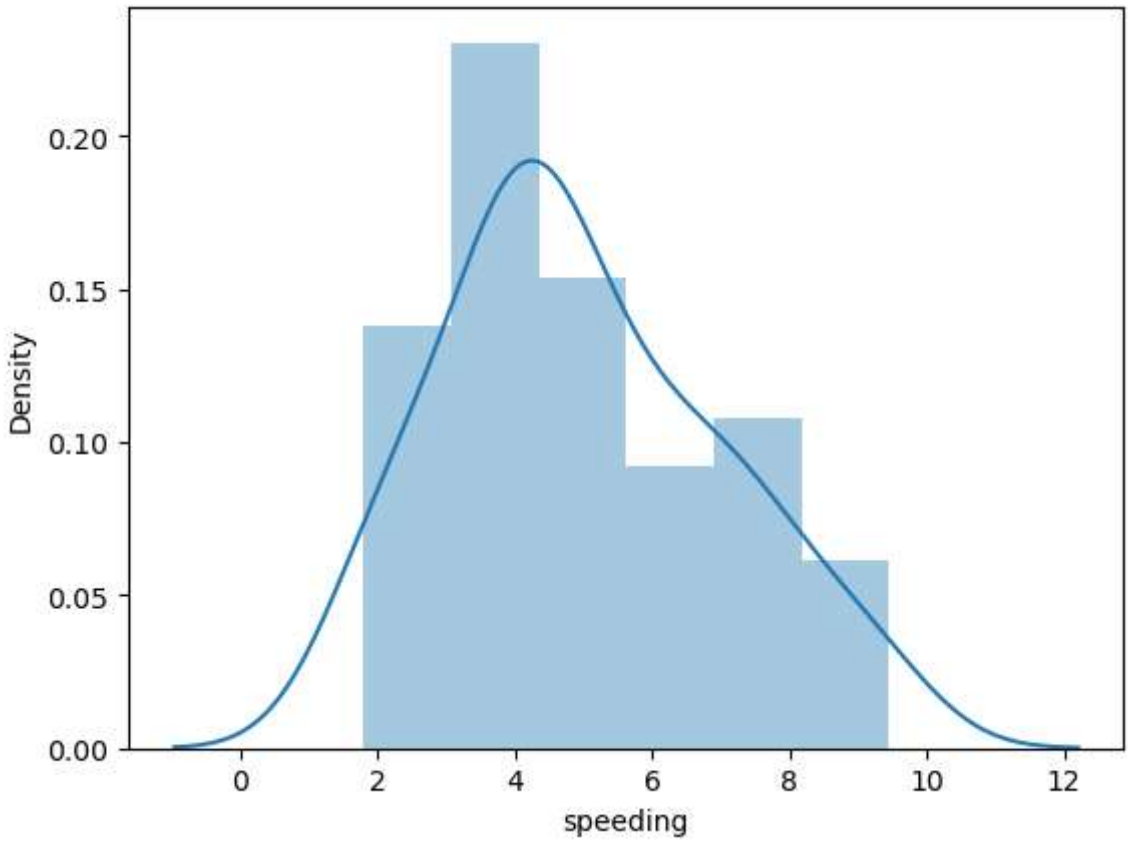


In [ ]:
```
# Inference:
"""
it can be inferenced that even though it has ups and downs,
total no. of crashes happened vs
crashes happened even with drivers not distracted graph is,
on the whole, an increasing graph.

"""
```

In [35]:
```
sns.distplot(df["speeding"])
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25784\2127910581.py: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["speeding"])
```

Out[35]:
```
<Axes: xlabel='speeding', ylabel='Density'>
```

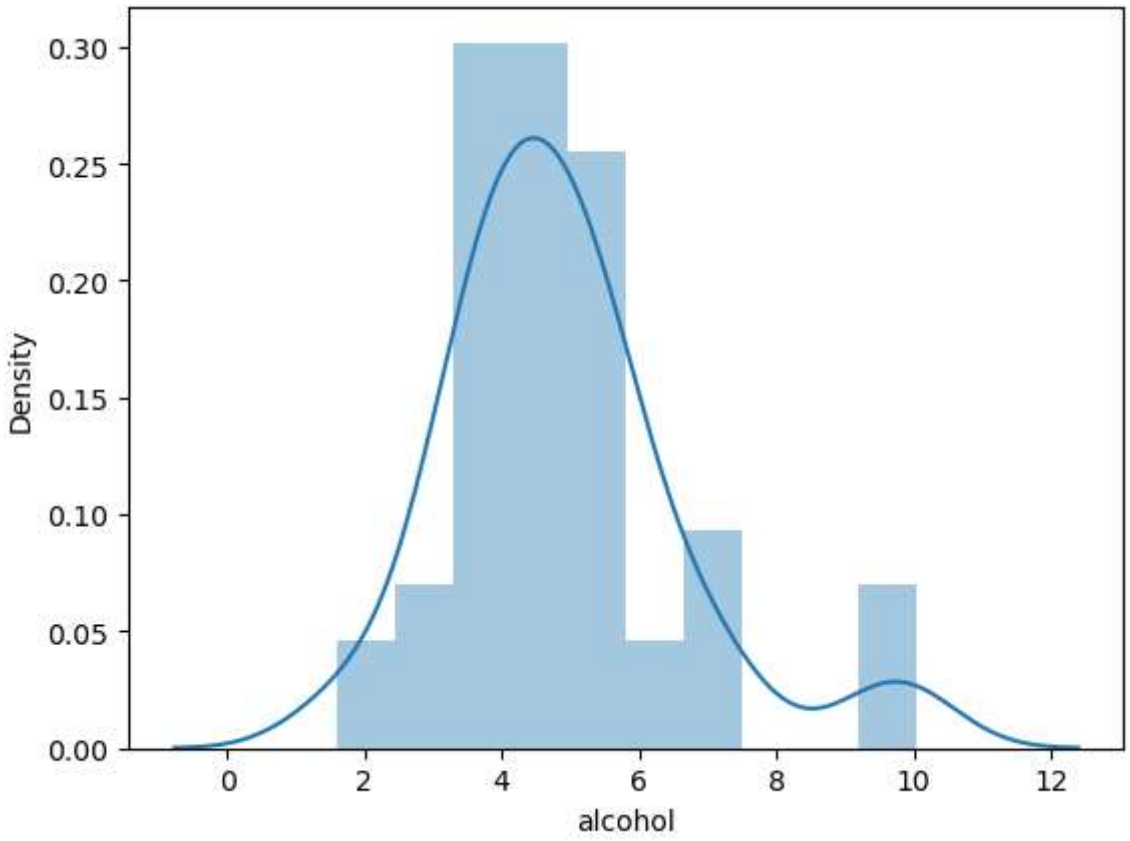


In [ ]:
```
# Inference: the Percentage Of Drivers Involved In Fatal Collisions Who Were Speeding
# are mostly ranging between 3 to 5
```

In [76]:
```
sns.distplot(df["alcohol"])
```

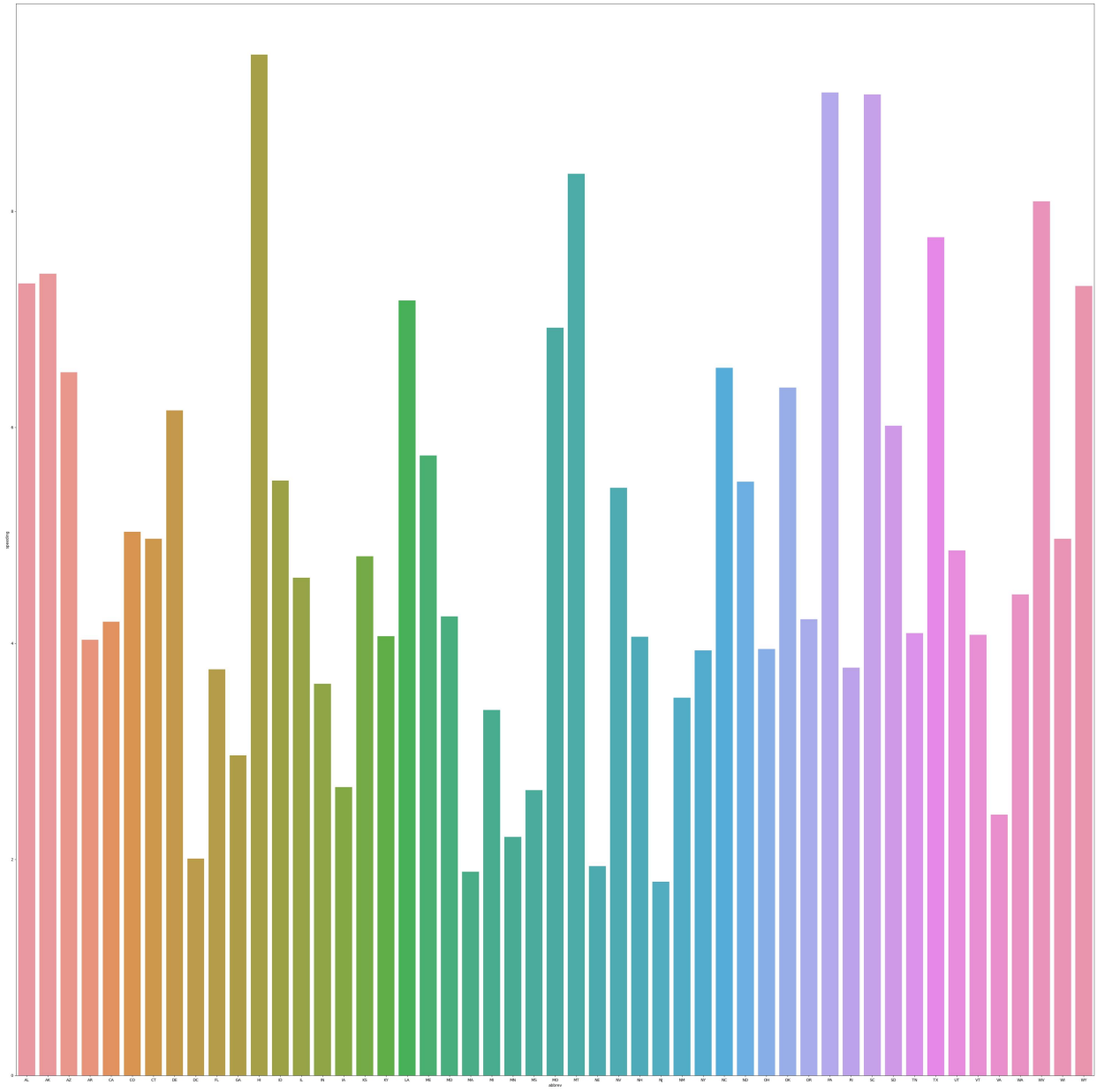
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25784\3201832786.py:1: UserWarning:

`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"])
```

Out[76]:
```
<Axes: xlabel='alcohol', ylabel='Density'>
```



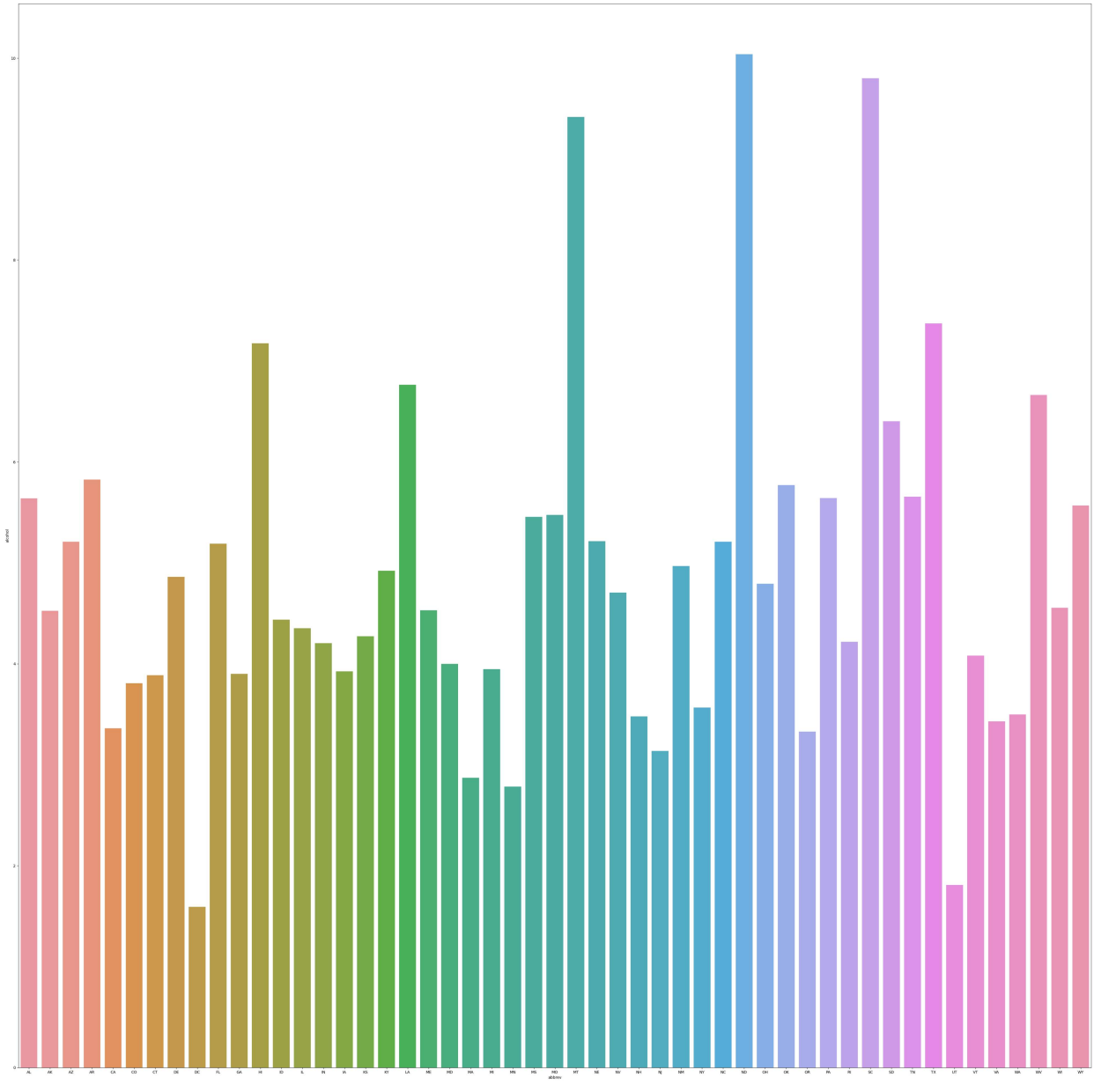
In [ ]:
```
# Inference:
"""the Percentage Of Drivers Involved In Fatal Collisions Who Were DRUNK
is mostly ranging between 4 to 5
"""
```

In [60]:
```
fig, ax = plt.subplots(figsize=(50, 50))

# drawing the plot
sns.barplot(data=df,x="abbrev",y="speeding", ax=ax)
plt.show()
```



In [ ]:
```
#Inference:
"""
After zooming in, It can be said that,
from the above bar chart,
number of crashes that happened due to speeding,
is the highest in the country HI (Hawaii)

"""
```

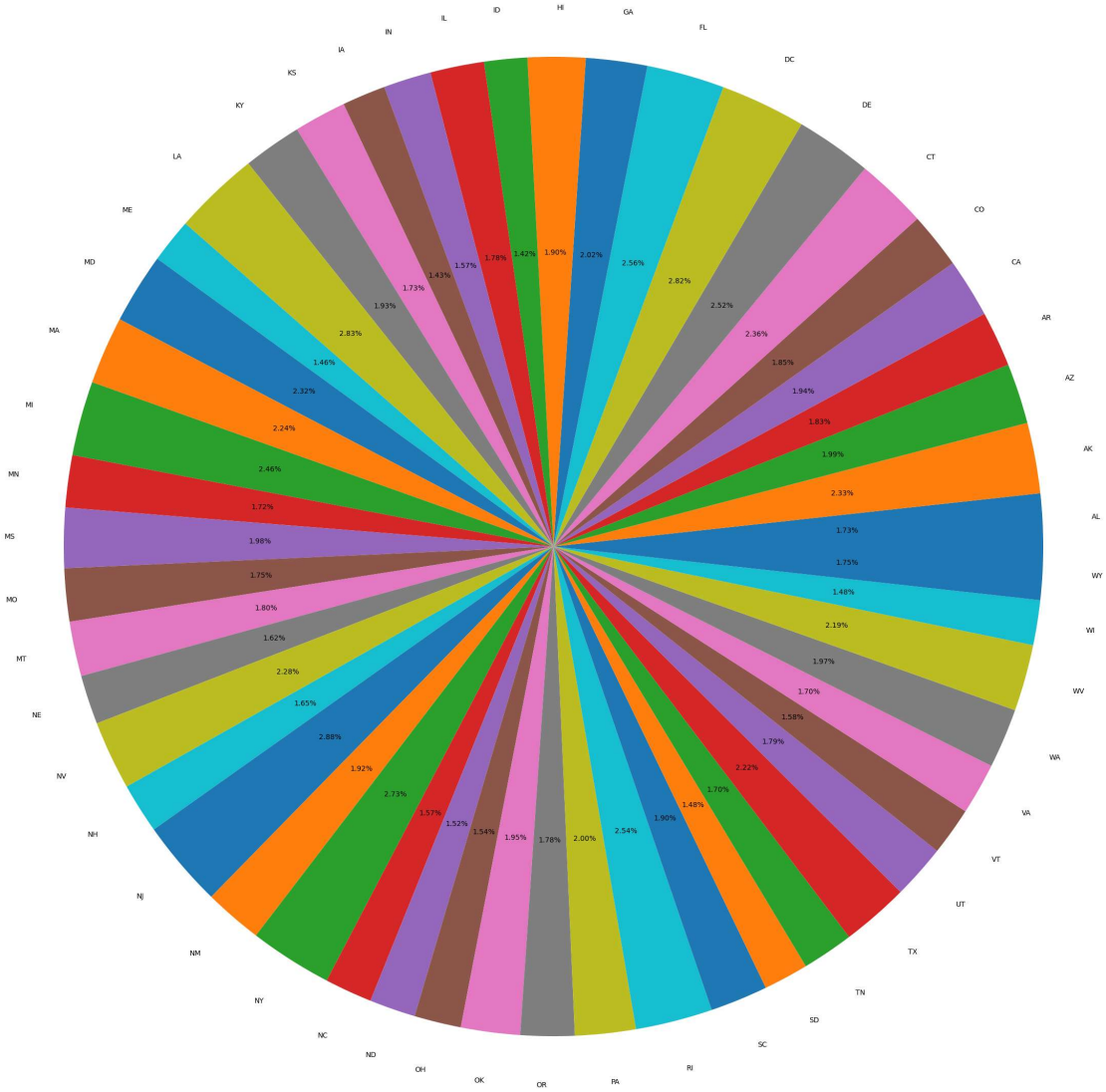
In [61]:
```
fig, ax1 = plt.subplots(figsize=(50, 50))

# drawing the plot
sns.barplot(data=df,x="abbrev",y="alcohol", ax=ax1)
plt.show()
```



In [ ]:
```
#Inference:
"""
After zooming in, It can be said that,
from the above bar chart,
number of crashes that happened due to Alcohol intake
is the highest in the country ND (North Dakota)

"""
```

In [101…
```
import matplotlib.pyplot as plt
number =  df["ins_premium"]
labels = df["abbrev"]
fig=plt.figure(figsize=(30, 30))
axes1=fig.add_axes([0.2,0.2,0.8,0.8])  #[left,bottom,width,height]
axes1.pie(number,labels=labels,autopct="%0.2f%%")
axes1.show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[101], line 7
      5 axes1=fig.add_axes([0.2,0.2,0.8,0.8])  #[left,bottom,width,height]
      6 axes1.pie(number,labels=labels,autopct="%0.2f%%")
----> 7 axes1.show()

AttributeError: 'Axes' object has no attribute 'show'
```

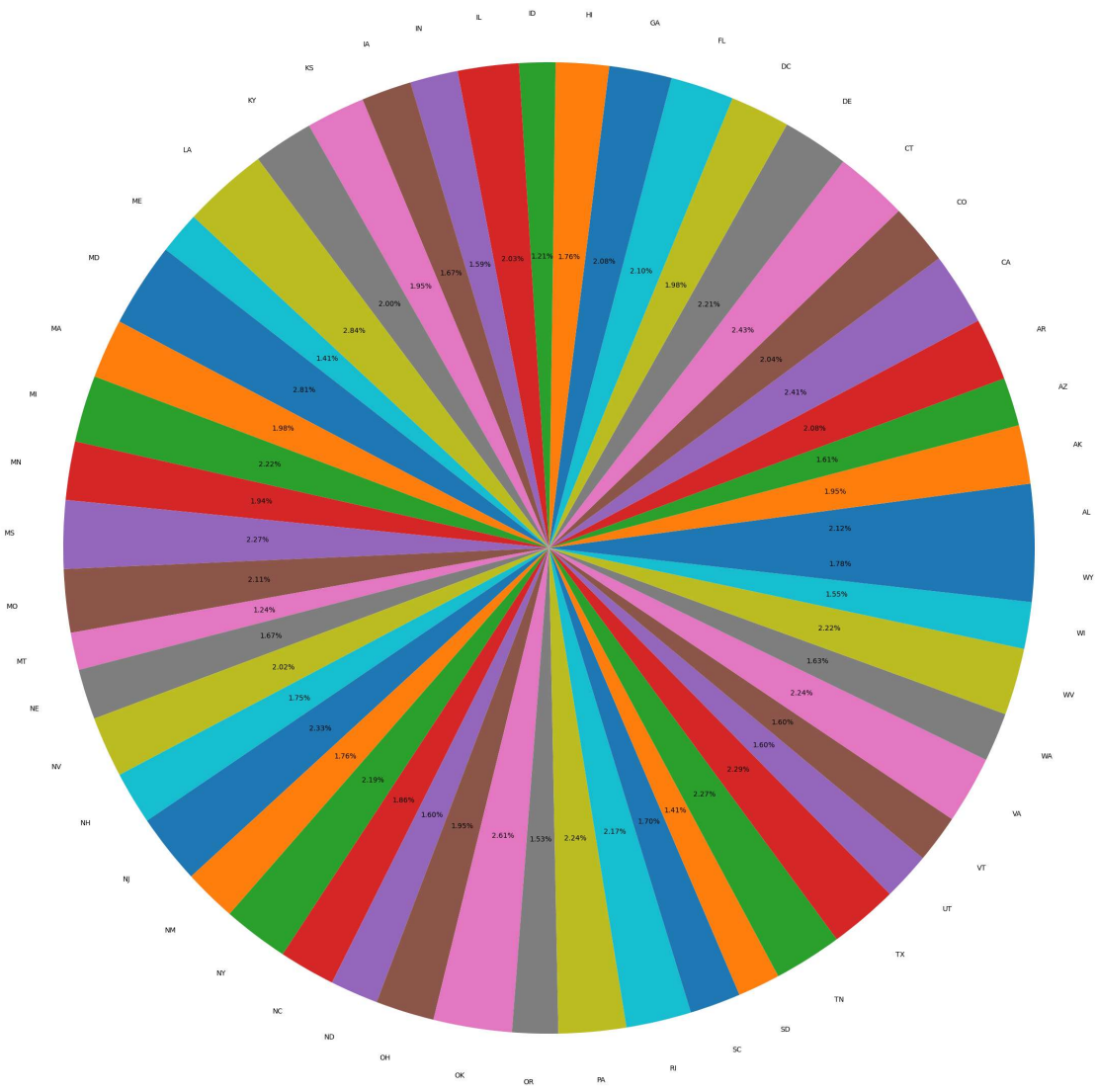


In [ ]:
```
#Inference:
"""
From the above pie chart,
it can be said that the Country NJ
(New Jersey - 2.88%) has highest number of drivers
who have a Car Insurance Premiums
"""
```

In [109…
```
import matplotlib.pyplot as plt
number =  df["ins_losses"]
labels = df["abbrev"]
fig=plt.figure(figsize=(30, 30))
```

```
axes1=fig.add_axes([0.2,0.2,0.8,0.8])  #[left,bottom,width,height]
axes1.pie(number,labels=labels,autopct="%0.2f%%")
axes1.show()
```

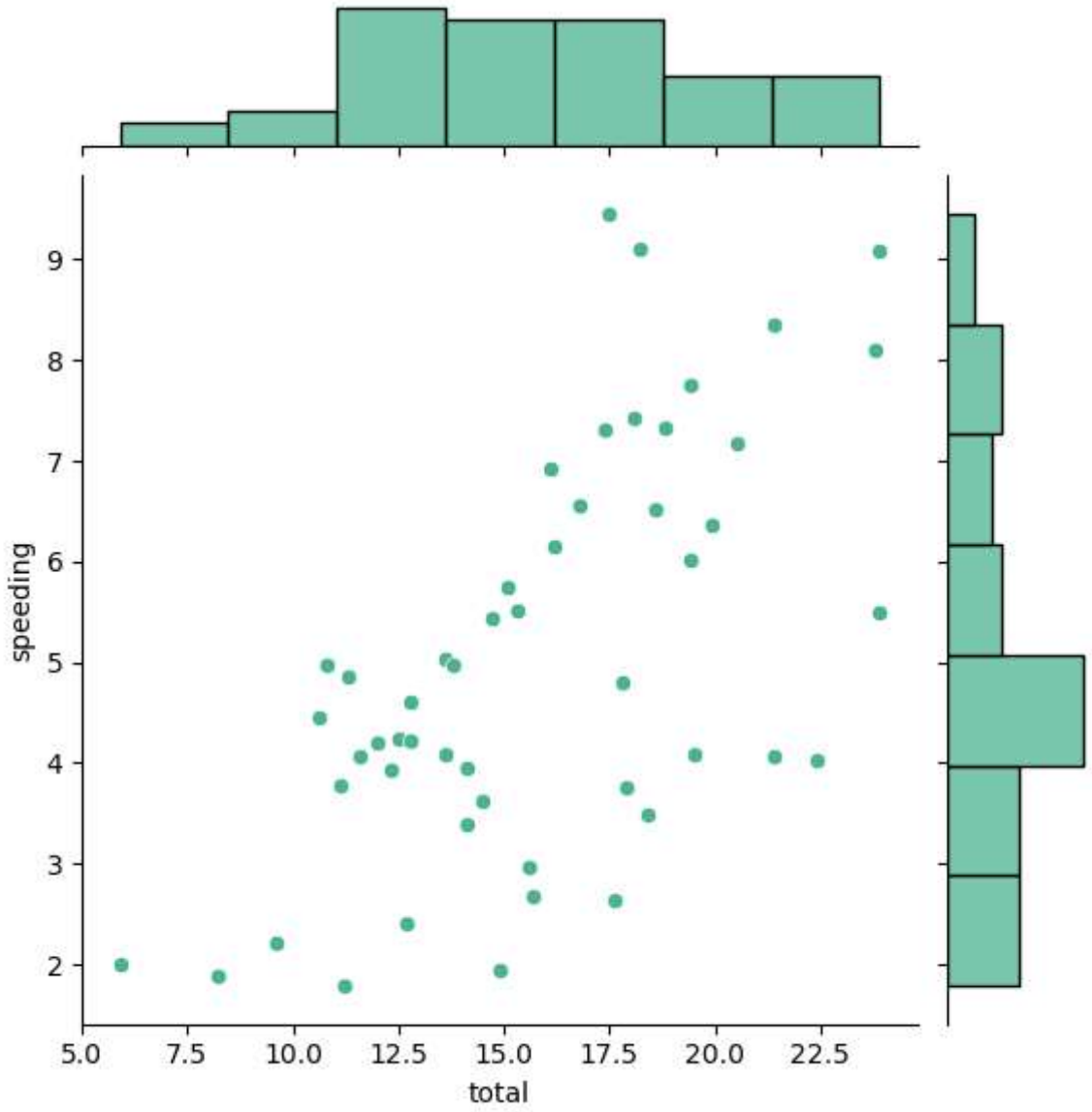
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[109], line 7
      5 axes1=fig.add_axes([0.2,0.2,0.8,0.8])  #[left,bottom,width,height]
      6 axes1.pie(number,labels=labels,autopct="%0.2f%%")
----> 7 axes1.show()

AttributeError: 'Axes' object has no attribute 'show'
```



In [ ]:
```
#Inference:
"""
From the above pie chart,
it can be said that the country LA (Los Angeles - 2.84%)
has the highest number of companies
that bore insurance losses

"""
```

In [115…
```
number =  df["total"]
labels = df["abbrev"]
ll = df["speeding"]
sns.jointplot(x=number, y=ll, color="#4CB391")
```

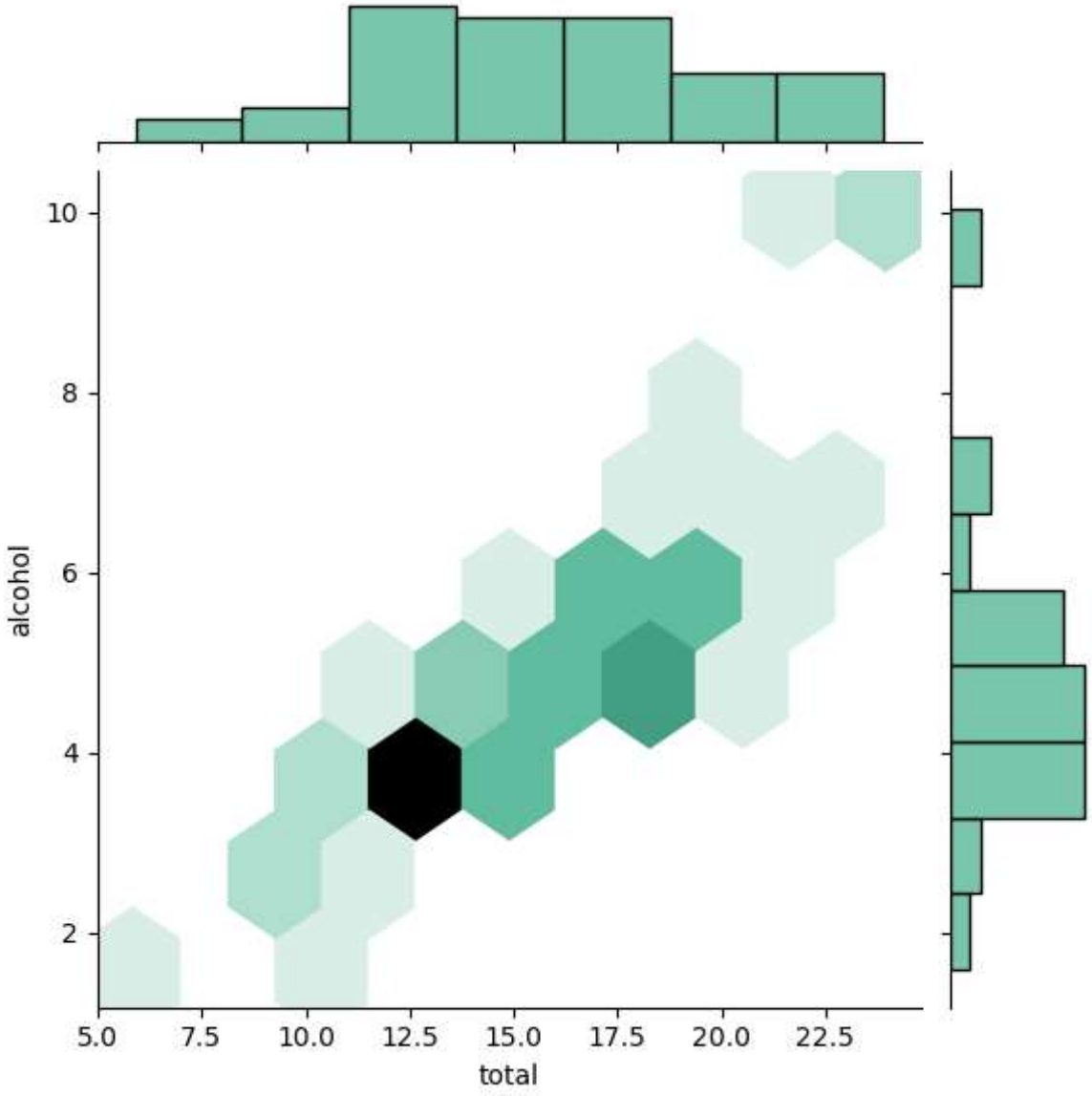
Out[115]:
```
<seaborn.axisgrid.JointGrid at 0x1c30ec0e810>
```



In [ ]:
```
#Inference:
"""
The most concentrated region in the above plot
(total - from 10 to 16 and speeding - from 3 to 6)

Generally, most concentrated regions represents
the areas of high density, that is, it shows us,
where the most data points are located.

"""
```

In [112…
```
number =  df["total"]
labels = df["abbrev"]
ll = df["alcohol"]
sns.jointplot(x=number, y=ll,kind = "hex", color="#4CB391")
```

Out[112]: <seaborn.axisgrid.JointGrid at 0x1c30e6b8e10>



In [ ]:
```
#Inference:
"""
The darkest or the black region in the above plot
(total - from 11 to 14 and alcohol - from 3 to 4.5) can be used to say that,
there is a strong correlation between the two variables.
(Positive correlation as seen from above graphs and correlation value)
This means that as the alcohol level increases,
the total number of car crashes also increases.
Generally, colour at a point depends on number of datapoints
i.e it will be light color when very few points had that value
and will be darker with more points.

"""
```

In [69]:
```
corr = df.corr()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_25784\658818363.py: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()
```

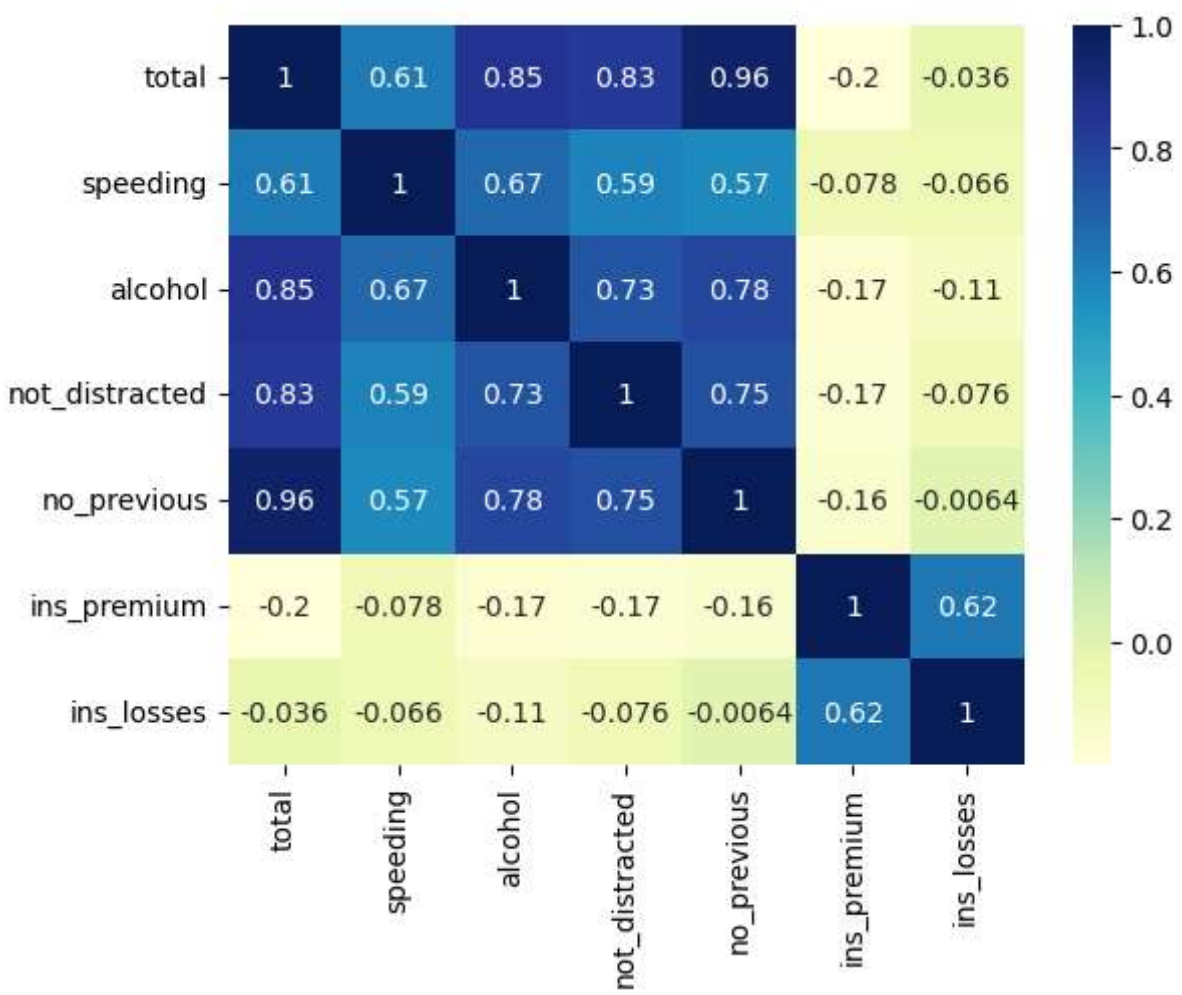
In [70]:
```
corr
```

Out[70]:

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.199702	-0.036011
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.077675	-0.065928
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.170612	-0.112547
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.174856	-0.075970
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.156895	-0.006359
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.000000	0.623116
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.623116	1.000000

In [75]:
```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

Out[75]:
```
<Axes: >
```



In [ ]:
```
#Inference:
"""
From the heatmap, it can be said that, strongest positive correlation
is found between the features "total" and "no_previous".
That is, total number of car crashes per billion miles increase when,
number of car crashes done by drivers with no previous accident case increases.

The weakest correlation is found between:
Percentage Of Drivers Involved In Fatal Collisions
Who Had Not Been Involved In Any Previous Accidents and
Losses incurred by insurance companies for collisions per insured driver.
hence we can say, there is almost no relation between these two features.
That is, cause of accident - no previous accident histroy does not affect
the losses incurred by the insurance companies.

Heatmaps generally rovides a visual summary of the data,
where each cell in the matrix corresponds to a data point or
a combination of variables,
and its color represents the magnitude or intensity of the data

"""
```