alexiaprincecheenath-assignment2

September 25, 2023

#Import the libraries

[1]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

#Import the dataset

[2]:
```
df=sns.load_dataset('car_crashes')
df.head()
```

[2]:
```
   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
```

[3]:
```
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
```

#Correlation

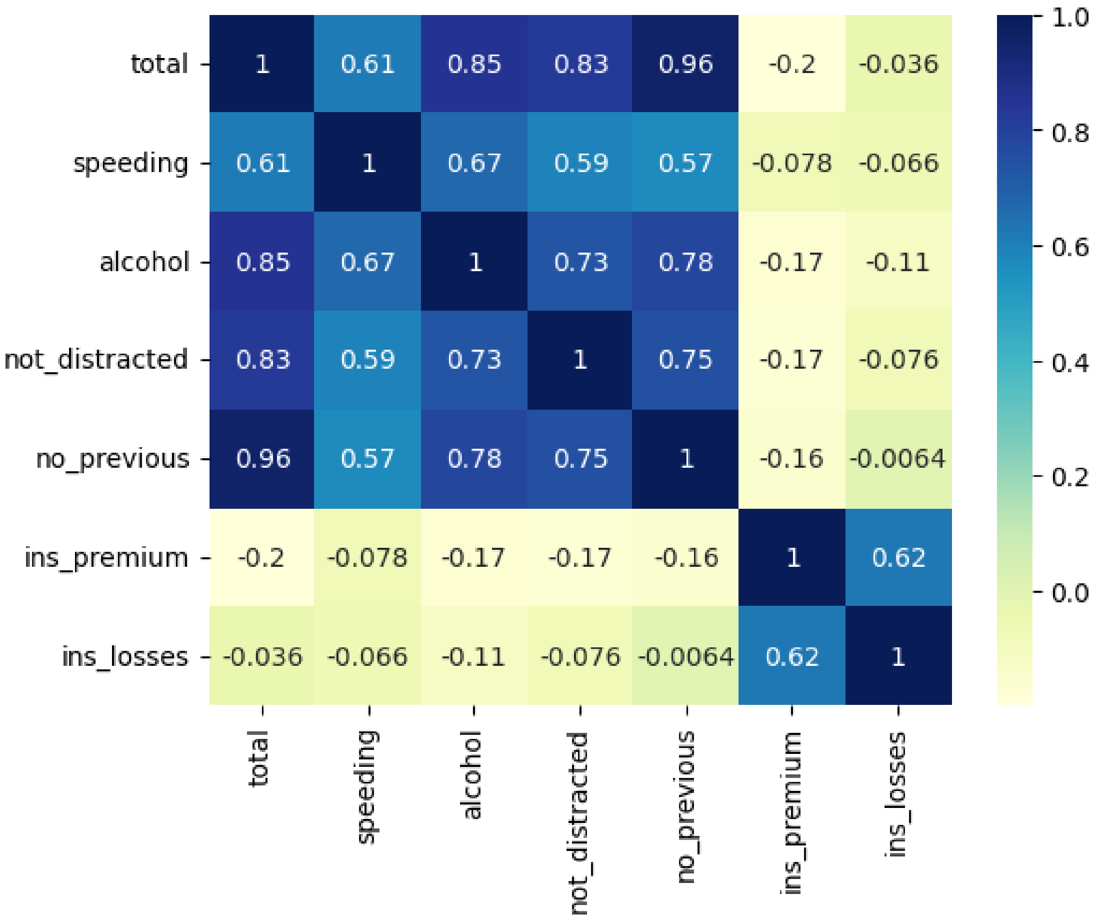
```
[4]: cor=df.corr()
     cor
```

```
<ipython-input-4-7a446f931109>: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.
  cor=df.corr()
```

```
[4]:                     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
```

```
[5]: sns.heatmap(cor,annot=True,cmap="YlGnBu")
     plt.show()
```

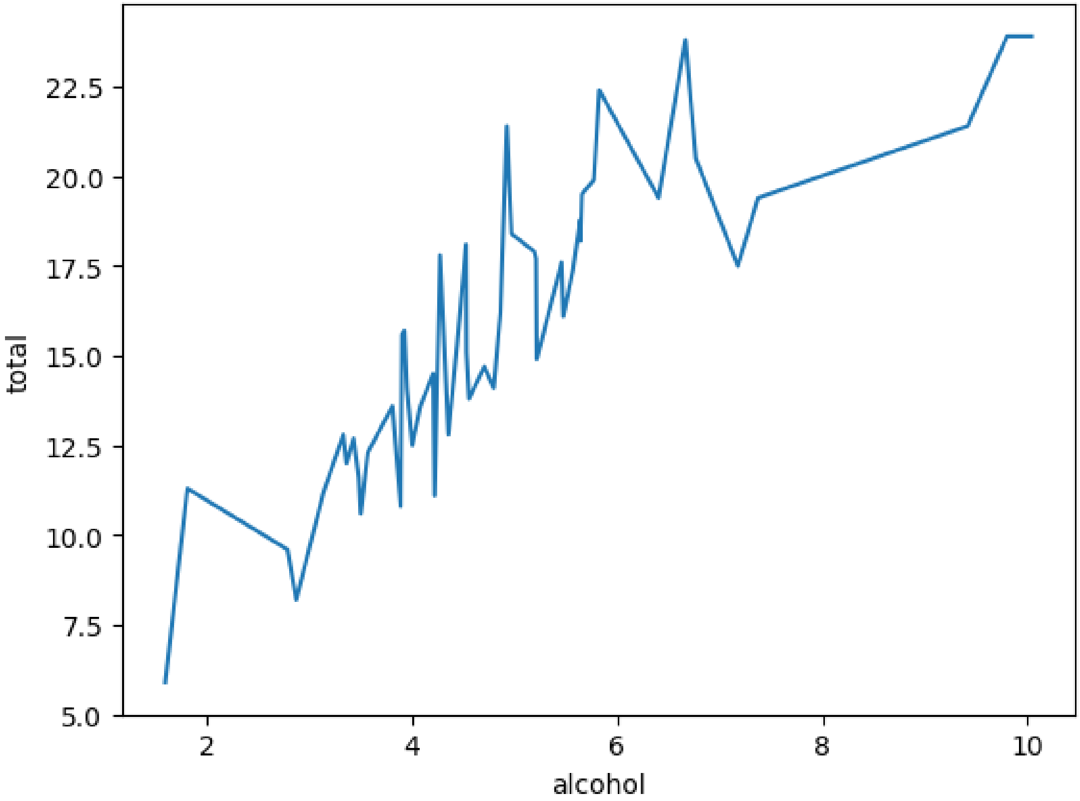


Inference: The above heat map depicts the correlation between each and every column. It is colour coded to easily differentiate between highly correlated columns and less correlated columns

#line graph

[6]:

```
sns.lineplot(x="alcohol",y="total",data=df,errorbar=None)
plt.show()
```

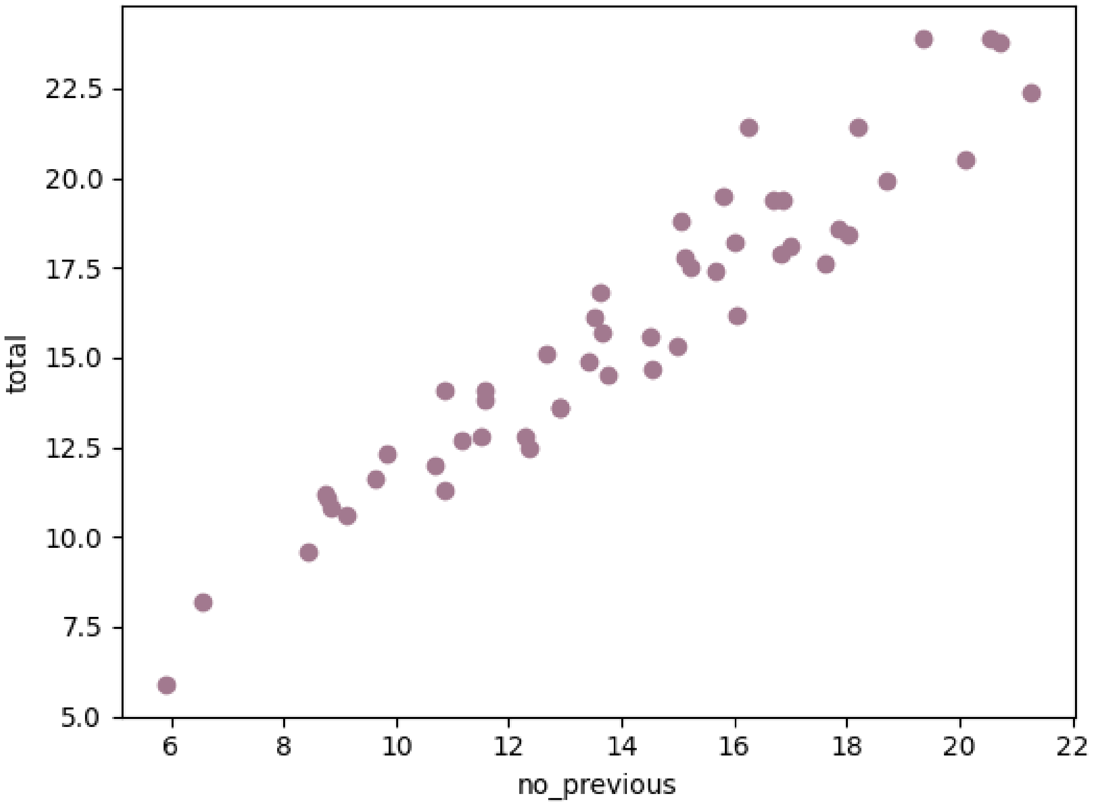


Inference: From the plot we understand that as the consumption of alcohol increases the total amount of car crashes increases.

#**Scatter Plot**

[7]:
```
plt.scatter(x="no_previous",y="total",data=df,color='#a2798f')
plt.xlabel("no_previous")
plt.ylabel("total")
```

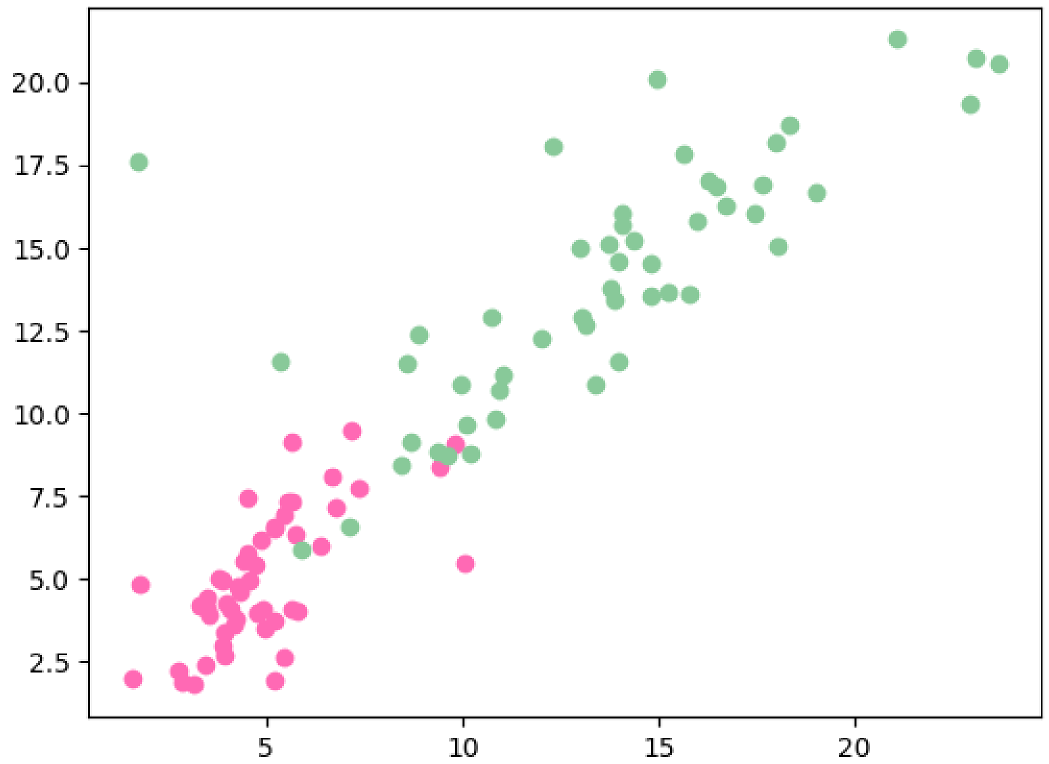
[7]:
```
Text(0, 0.5, 'total')
```



Inference: From the scatterplot we understand that the two columns are highly correlated and have a linear relationship.The people who have not experienced a car crash before is more likely to end up in an accident.

#**Compare two plots**

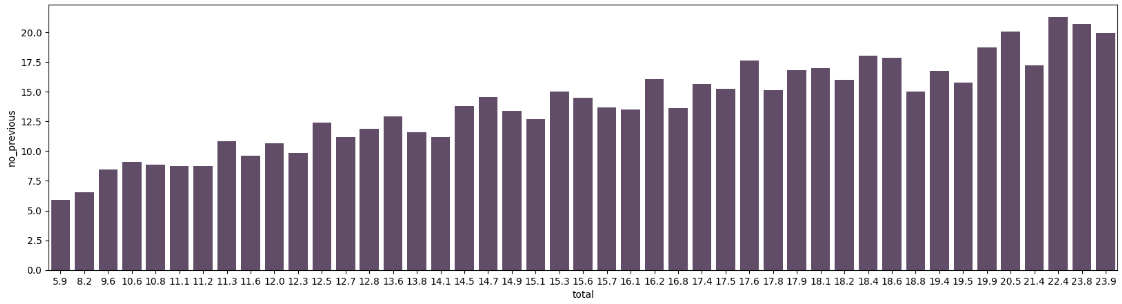
[8]:
```
plt.scatter(x="alcohol",y="speeding",data=df,color='hotpink')
plt.scatter(x="not_distracted",y="no_previous",data=df,color='#88c999')
plt.show()
```



Inference: From the graph it is visible that the relationship between alcohol and speeding is more clustered than the relationship between not distracted and no previous.

#**Bar Plot**

```
[9]: plt.subplots(figsize=(20,5))
     sns.barplot(x="total",y="no_previous",data=df,color='#63486c',width=0.
       ↪8,errorbar=None)
     plt.show()
```

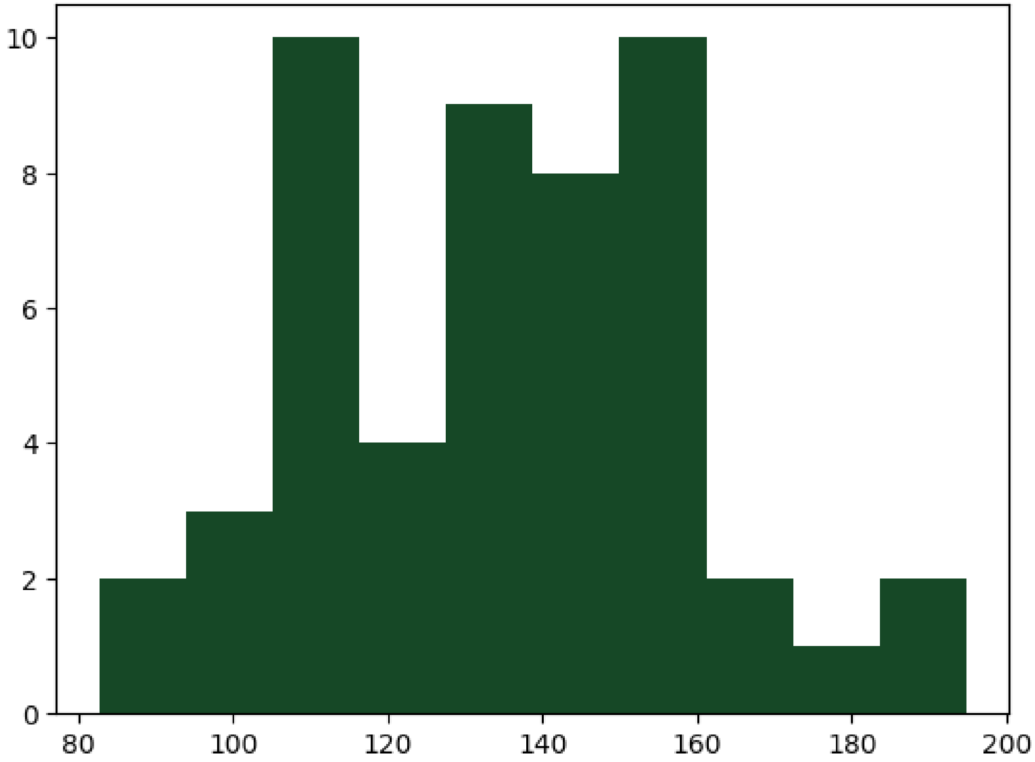


Inference: The bar plot gives us the relation between the total and no_previous

#Histogram

[10]:

```
plt.hist(x=df["ins_losses"],color='#164826')
plt.show()
```



Inference: The histogram shows us the insurance losses at different times.

#Distribution Plot

[11]:

```
sns.distplot(df["ins_premium"])
plt.show()
```

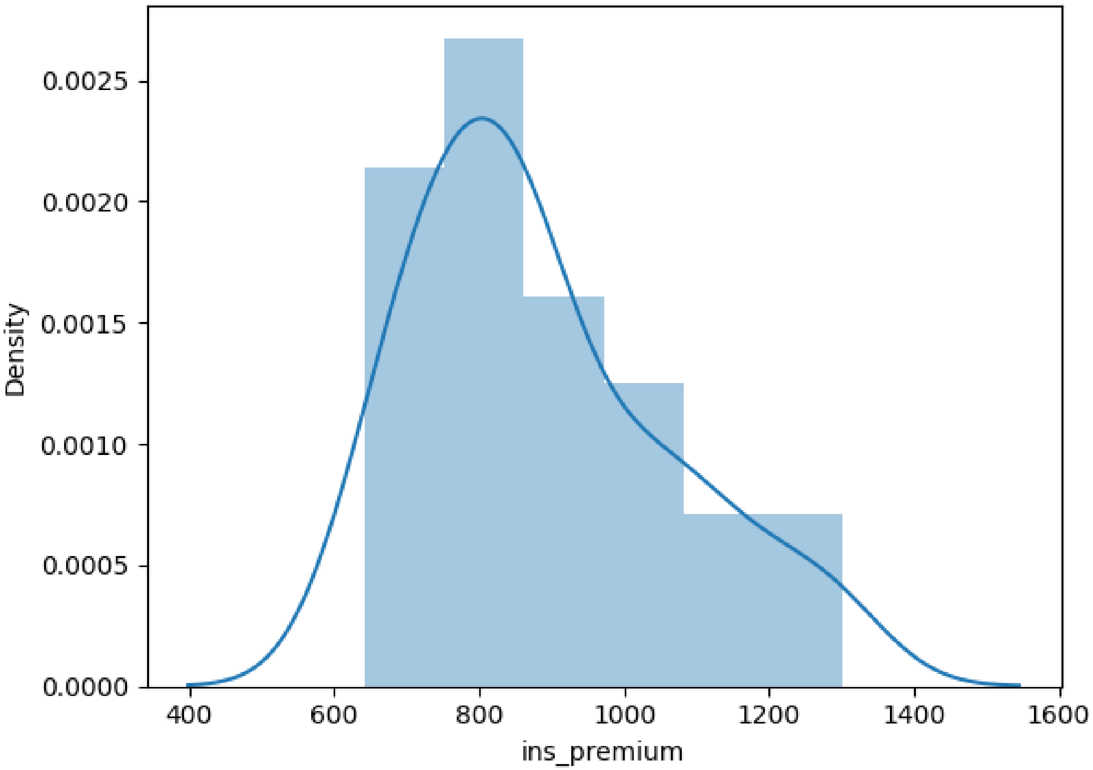
```
<ipython-input-11-39e8f71360ee>:1: UserWarning:

`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["ins_premium"])
```



Inference: This distribution plot of the insurance premium tells us where the majority of the data lies and also draws a density line for better understanding

[12]:
```
sns.distplot(df["total"])
plt.show()
```

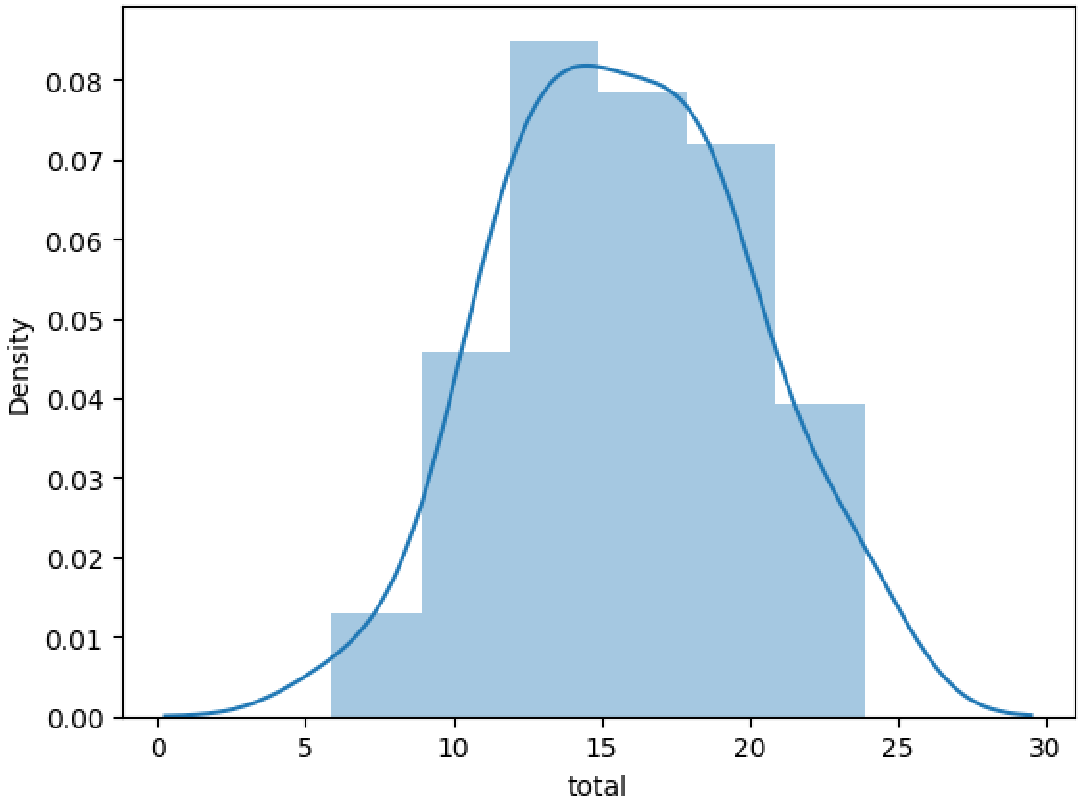
```
<ipython-input-12-e30bd477160a>:1: UserWarning:

`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["total"])
```

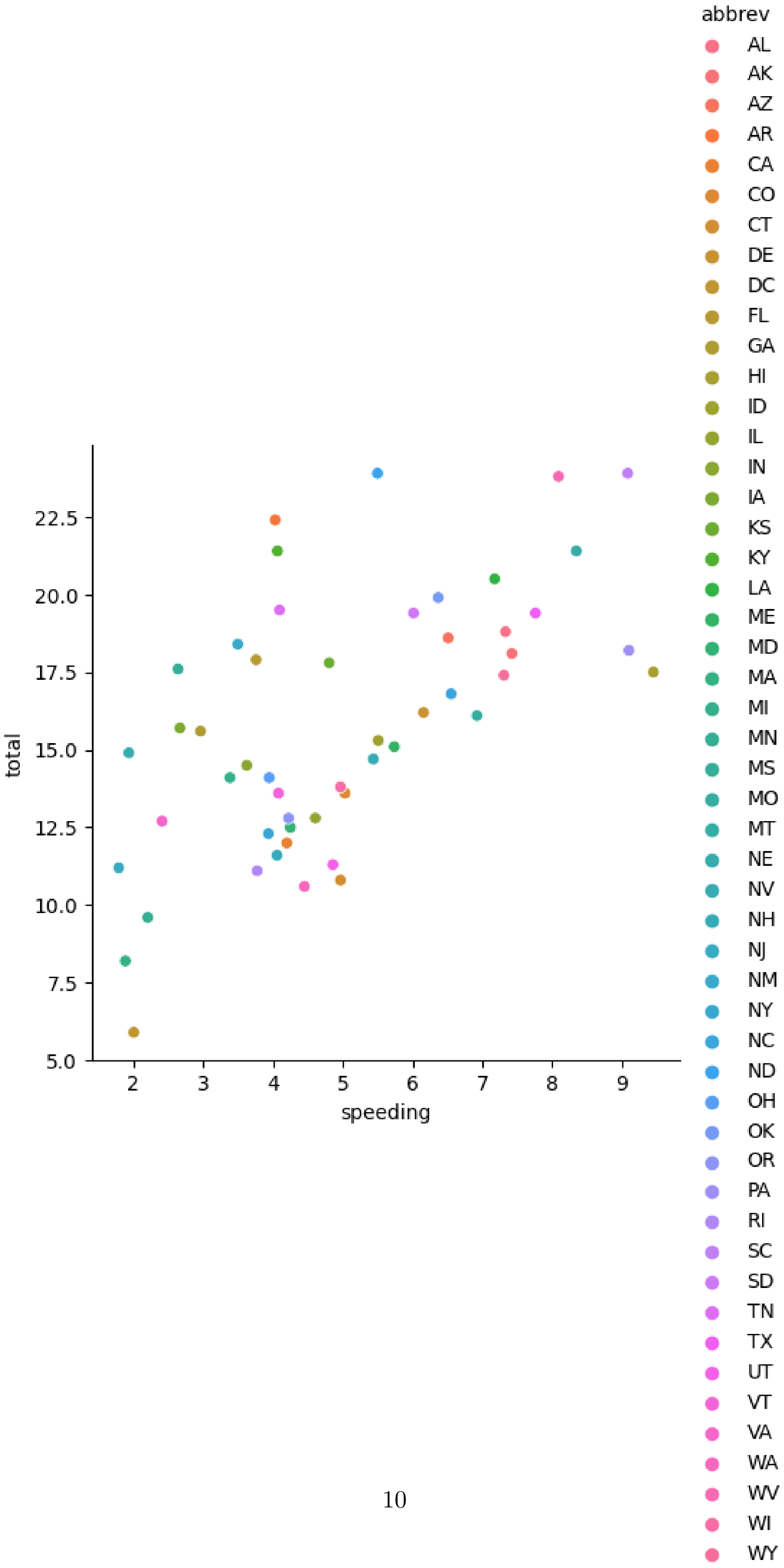


Inference: This distribution plot of the total tells us where the majority of the data lies and also draws a density line for better understanding.

#**Relational Plot**

[13]:

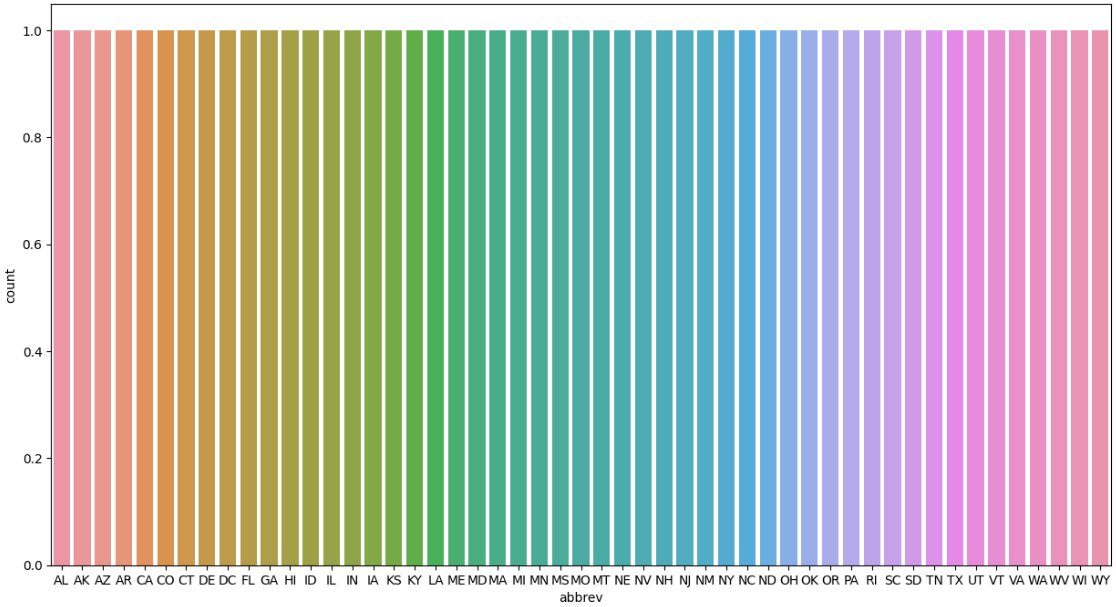
```
sns.relplot(x="speeding",y="total",data=df,hue="abbrev")
plt.show()
```


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
total
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2
3
4
5
6
7
8
9
speeding

Inference: This relational plot shows the relation between speeding and total in different states of USA by color coding them.

[14]:

```
plt.subplots(figsize=(15,8))
sns.countplot(x="abbrev",data=df)
plt.show()
```

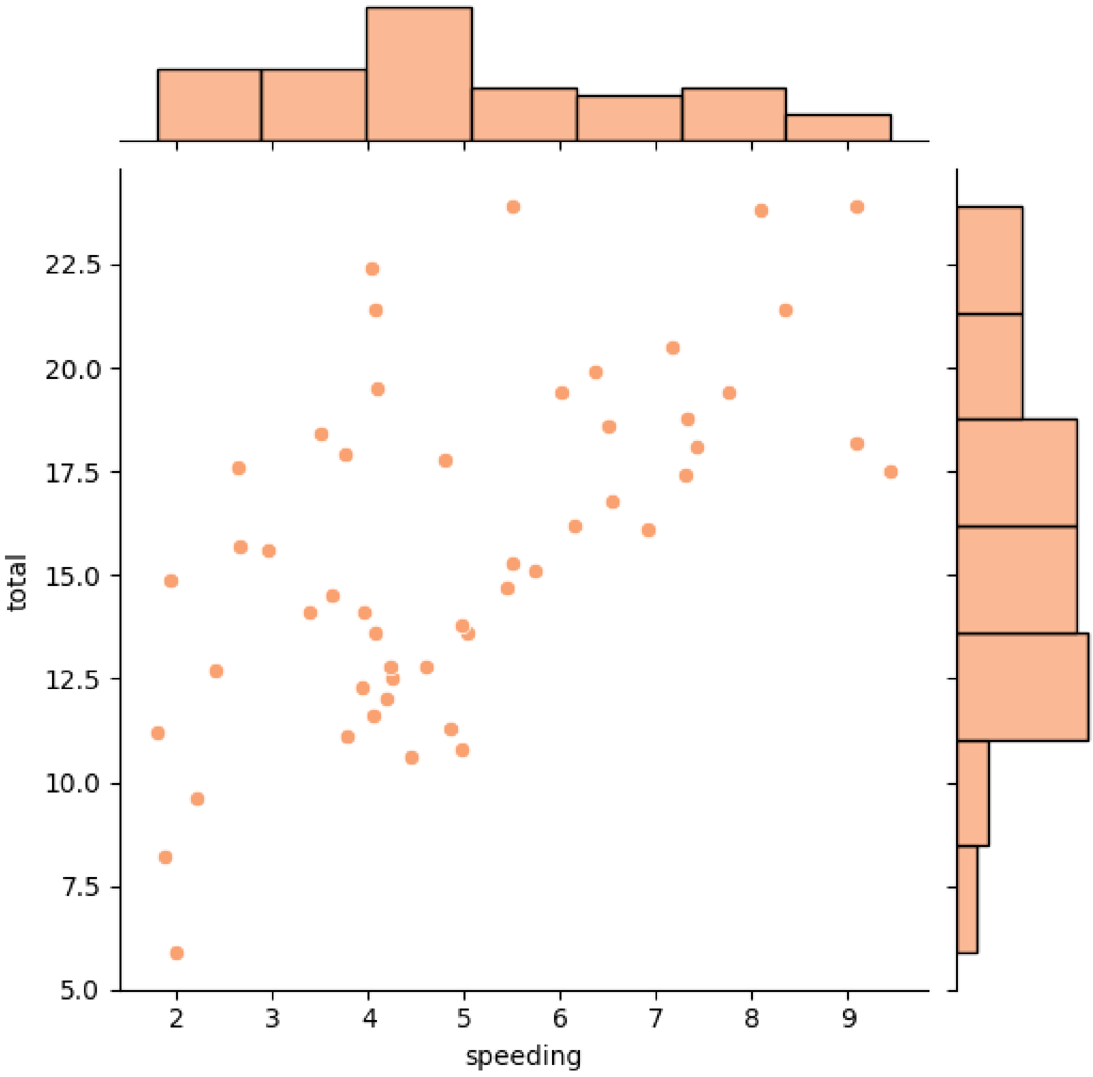


Inference: From the count plot it is evident that each state had exactly one entry.

#**Joint Plot**

[15]:

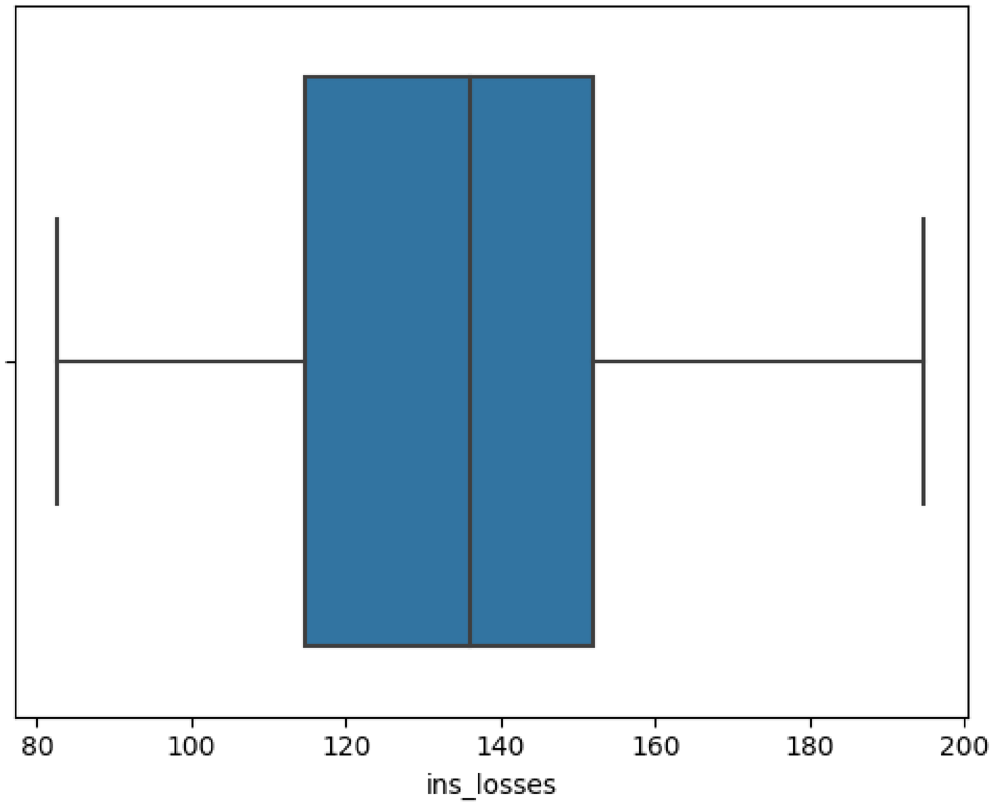
```
sns.jointplot(x="speeding",y="total",data=df,color='#f9a170')
plt.show()
```



Inference: We can observe the scatter plot between speeding and total along with their respective histograms to get a better idea about the relationship between them. We can notice from the graph that they are not highly correlated.

#**Box Plot**

[16]:
```
sns.boxplot(x="ins_losses",data=df)
plt.show()
```



Inference: From the graph we can notice that this graph is negatively skewed as the median is more towards the right. There are no outliers.