vit-si-assignment-2

September 14, 2023

0.0.1 SmartBridge Assessment -2 RISHIKA SAHOO 21BCB0184

- car_crashes data set imported from seaborn
- Done Visualization for the data set and writtern inference for each graph that has been observed.

[2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

[3]:
```
warnings.filterwarnings('ignore', category=FutureWarning)
# To Ignore Future Warnings

warnings.filterwarnings('ignore', category=UserWarning)
# To ignore user warnings
```

[4]:
```
df = sns.load_dataset('car_crashes')
```

[5]:
```
df
```

[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
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

0.1 About Data set:

- total : No of Drivers involved per billion miles
- speeding : % of drivers involed in car crashes by speeding
- alcohol : % of drivers involved in car cashes by alcohol
- not_distracted : % of drivers involved without distraction

- no_previous : % of drivers involved without previous crashes records
- ins_premium : Car insurence premium range
- ins_loss : Insurence company loss
- abbrev : Abbrevations of States of US (NH : New Hampshire , MD : Maryland)

[6]:
```
df.isnull().sum()
```

[6]:
```
total             0
speeding          0
alcohol           0
not_distracted    0
no_previous       0
ins_premium       0
ins_losses        0
abbrev            0
dtype: int64
```

[7]:
```
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
```

[8]:
```
df.describe()
```

[8]:
```
           total   speeding    alcohol  not_distracted  no_previous  \
count  51.000000  51.000000  51.000000       51.000000    51.000000
mean   15.790196   4.998196   4.886784       13.573176    14.004882
std     4.122002   2.017747   1.729133        4.508977     3.764672
min     5.900000   1.792000   1.593000        1.760000     5.900000
25%    12.750000   3.766500   3.894000       10.478000    11.348000
50%    15.600000   4.608000   4.554000       13.857000    13.775000
75%    18.500000   6.439000   5.604000       16.140000    16.755000
max    23.900000   9.450000  10.038000       23.661000    21.280000
```

```
       ins_premium  ins_losses
count    51.000000   51.000000
mean    886.957647  134.493137
std     178.296285   24.835922
min     641.960000   82.750000
25%     768.430000  114.645000
50%     858.970000  136.050000
75%    1007.945000  151.870000
max    1301.520000  194.780000
```

[9]:
```
df.head()
```

[9]:
```
   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
```

[10]:
```
df.tail()
```

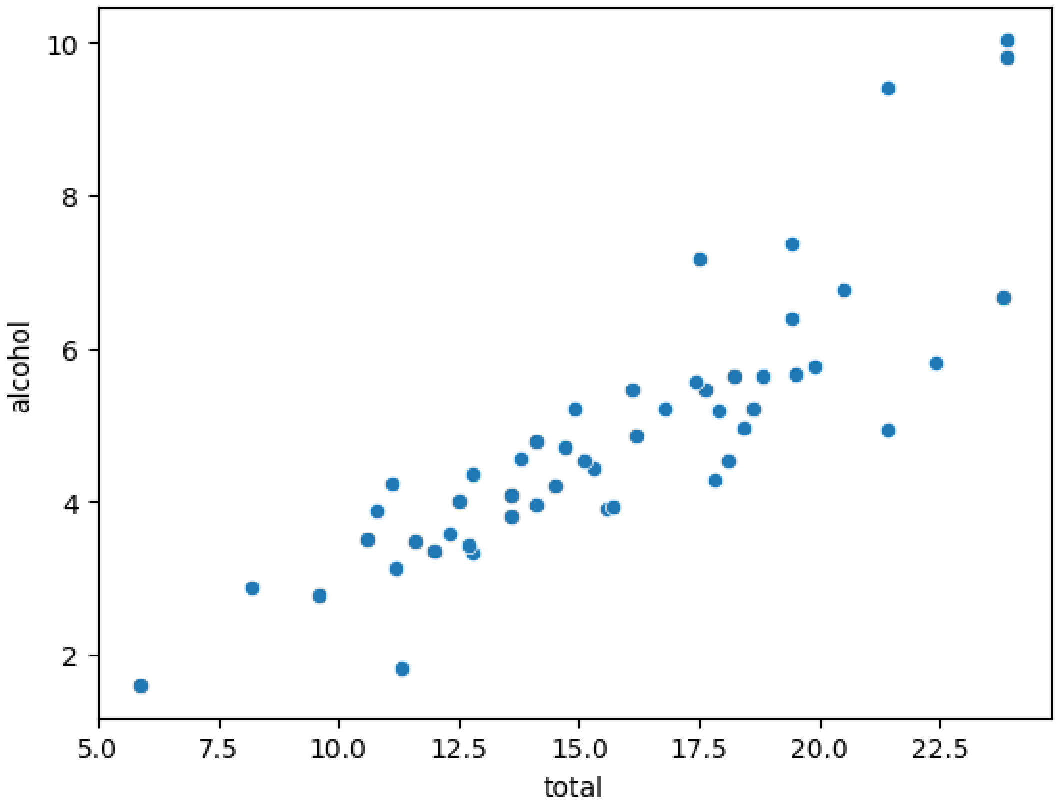
[10]:
```
    total  speeding  alcohol  not_distracted  no_previous  ins_premium  \
46   12.7     2.413    3.429          11.049       11.176       768.95
47   10.6     4.452    3.498           8.692        9.116       890.03
48   23.8     8.092    6.664          23.086       20.706       992.61
49   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
46      153.72     VA
47      111.62     WA
48      152.56     WV
49      106.62     WI
50      122.04     WY
```

[11]:
```
df.shape
```

[11]:
```
(51, 8)
```

[12]:
```
sns.scatterplot(x="total",y="alcohol",data=df)
```

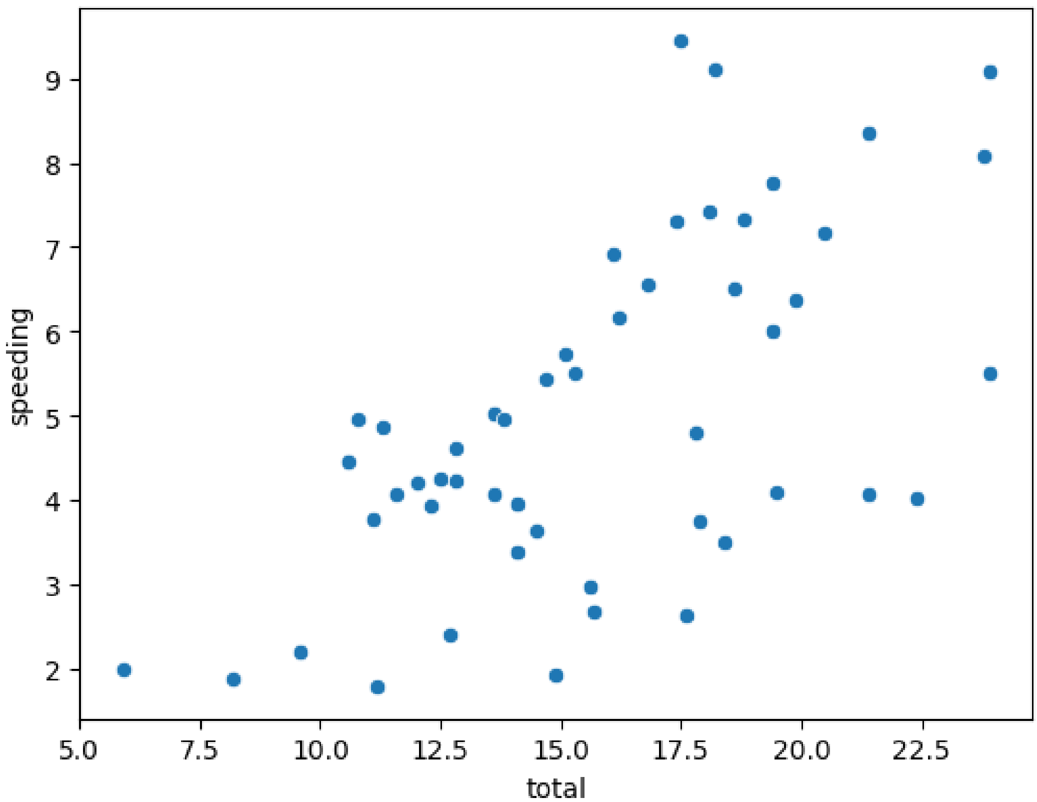
[12]: <Axes: xlabel='total', ylabel='alcohol'>



- Inference : As total drivers are increasing , car crashes due to alcohol are also increasing so directly proportional.

[13]:
```
sns.scatterplot(x="total",y="speeding",data=df)
```

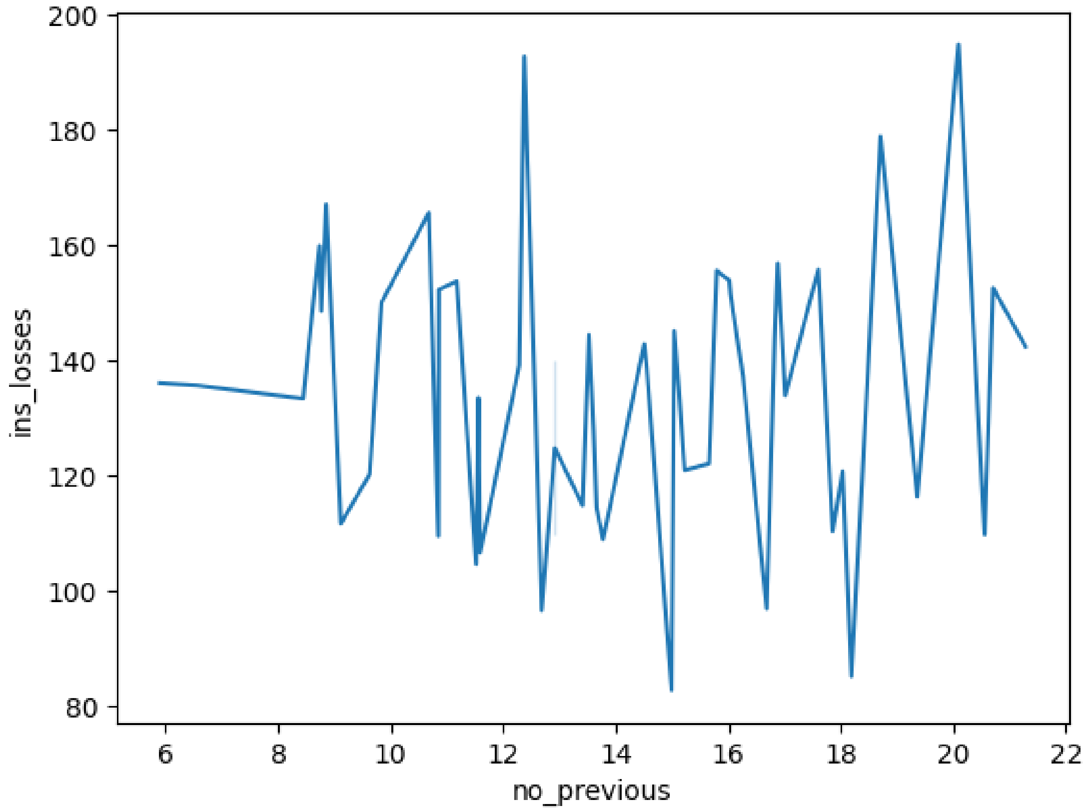
[13]: <Axes: xlabel='total', ylabel='speeding'>



- Inference : As total drivers increasing car crashes due to speeding also increases but not directly proportional

[14]:
```
sns.lineplot(x="no_previous",y="ins_losses",data=df)
```

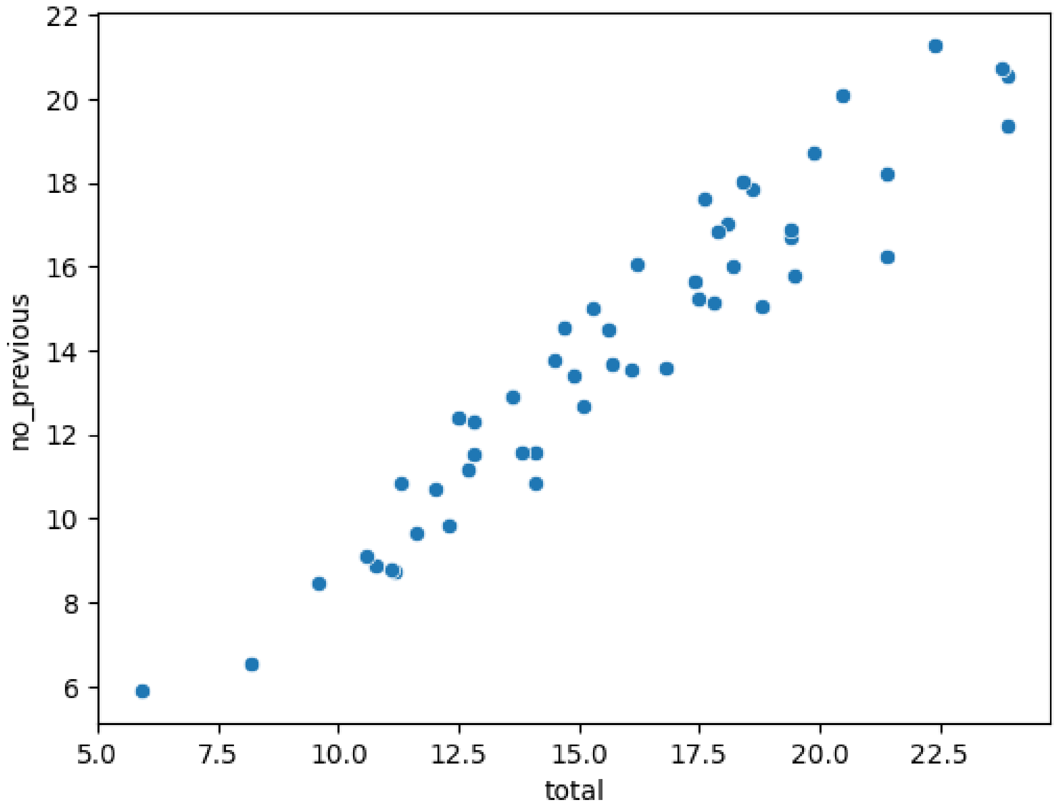
[14]: <Axes: xlabel='no_previous', ylabel='ins_losses'>



- Inference : It was increasing and decreasing and the lowerst point is occured at 15 (no_previous) and highest at 190 (no_previous) [Approx]

[15]:
```
sns.scatterplot(x="total",y="no_previous",data=df)
```

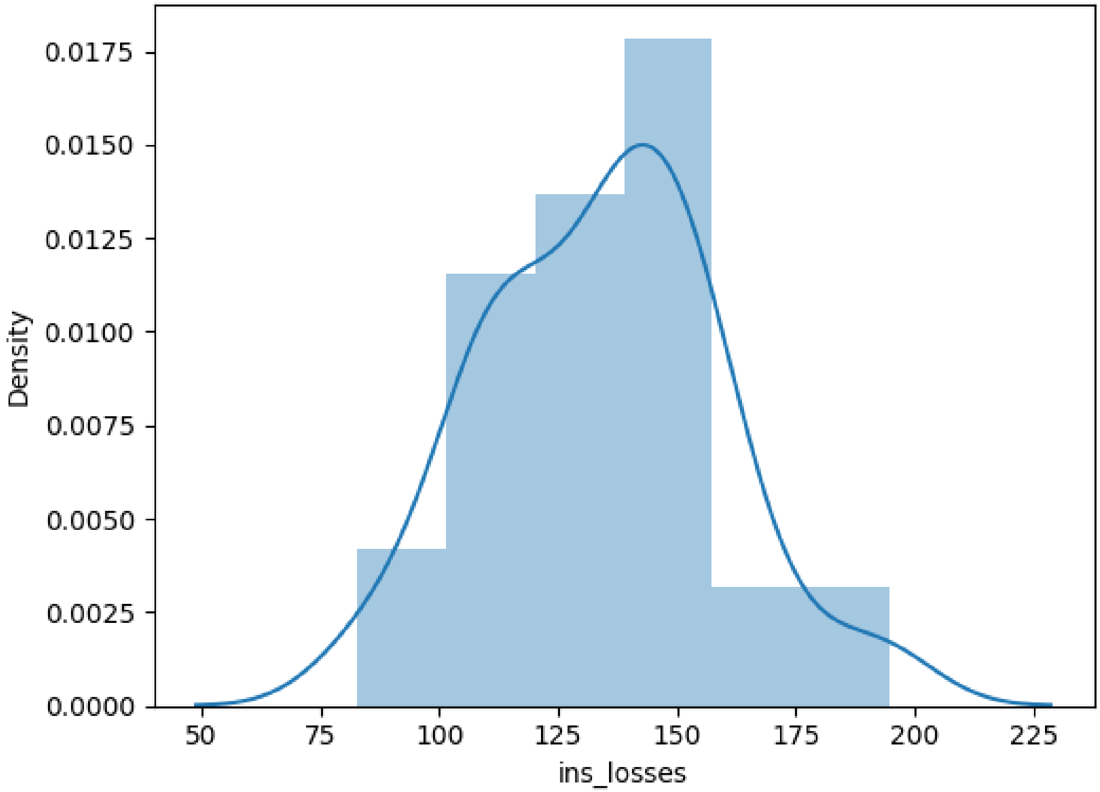
[15]: <Axes: xlabel='total', ylabel='no_previous'>



- Inference : Alcohol and no_previous are directly proportional

[17]:
```
sns.distplot(df["ins_losses"])
```

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



- Inference : Cars whose insurance losses is around 150 are going to crash more(approx)

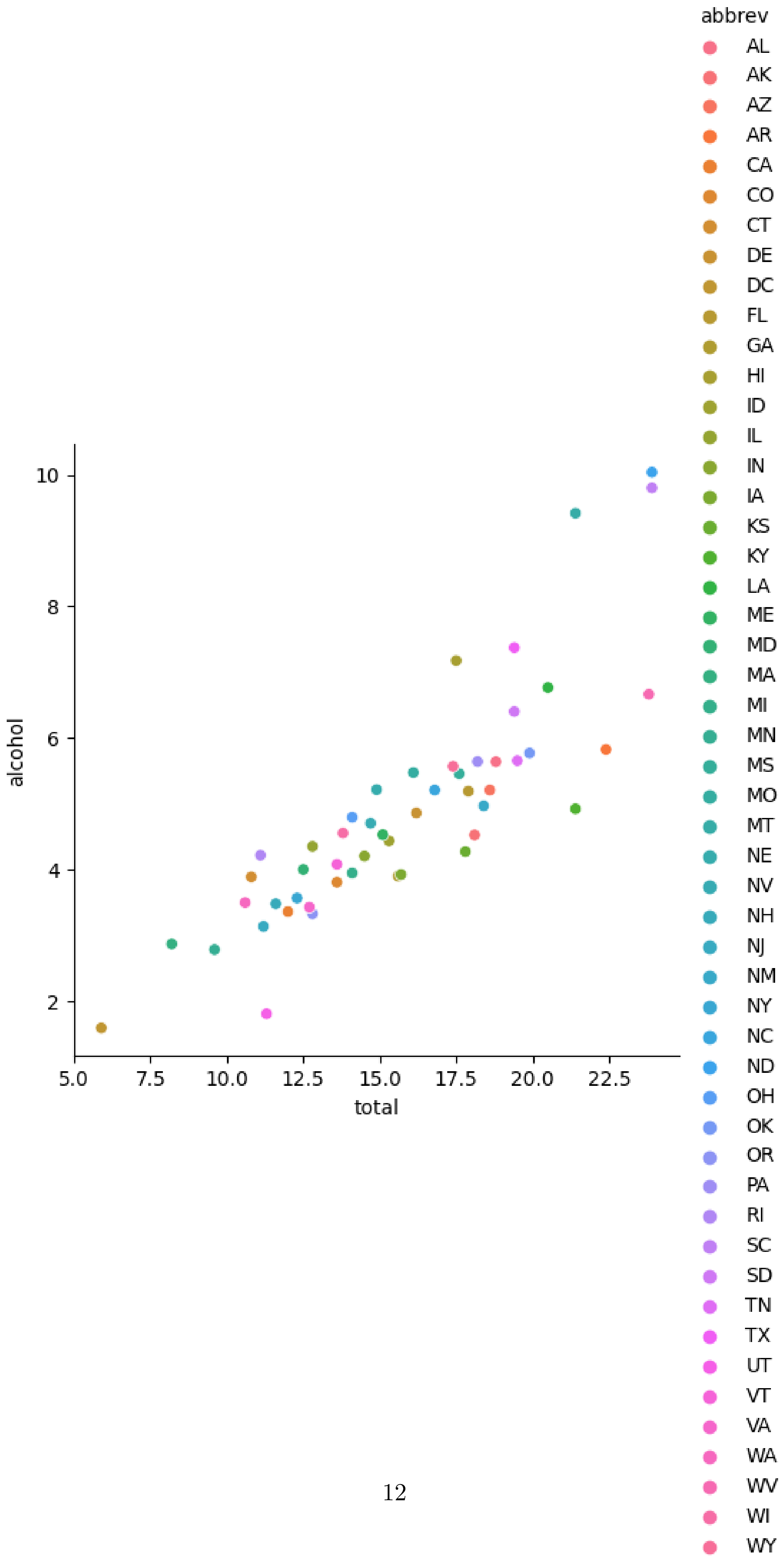
[22]:
```
df["abbrev"].value_counts()
```

[22]:
```
AL    1
PA    1
NV    1
NH    1
NJ    1
NM    1
NY    1
NC    1
ND    1
OH    1
OK    1
OR    1
RI    1
MT    1
SC    1
SD    1
TN    1
TX    1
```

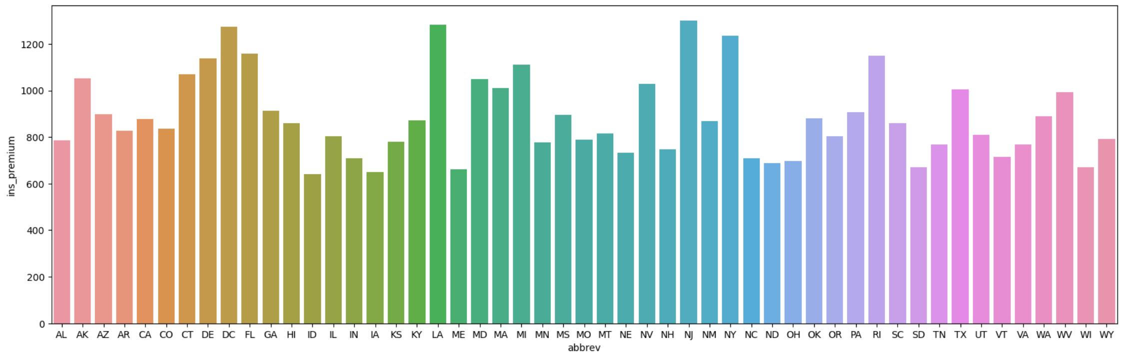
```
UT    1
VT    1
VA    1
WA    1
WV    1
WI    1
NE    1
MO    1
AK    1
ID    1
AZ    1
AR    1
CA    1
CO    1
CT    1
DE    1
DC    1
FL    1
GA    1
HI    1
IL    1
MS    1
IN    1
IA    1
KS    1
KY    1
LA    1
ME    1
MD    1
MA    1
MI    1
MN    1
WY    1
Name: abbrev, dtype: int64
```

[23]:
```
sns.relplot(x="total",y="alcohol",data=df,hue="abbrev")
```

[23]:
```
<seaborn.axisgrid.FacetGrid at 0x7c31a54c94e0>
```

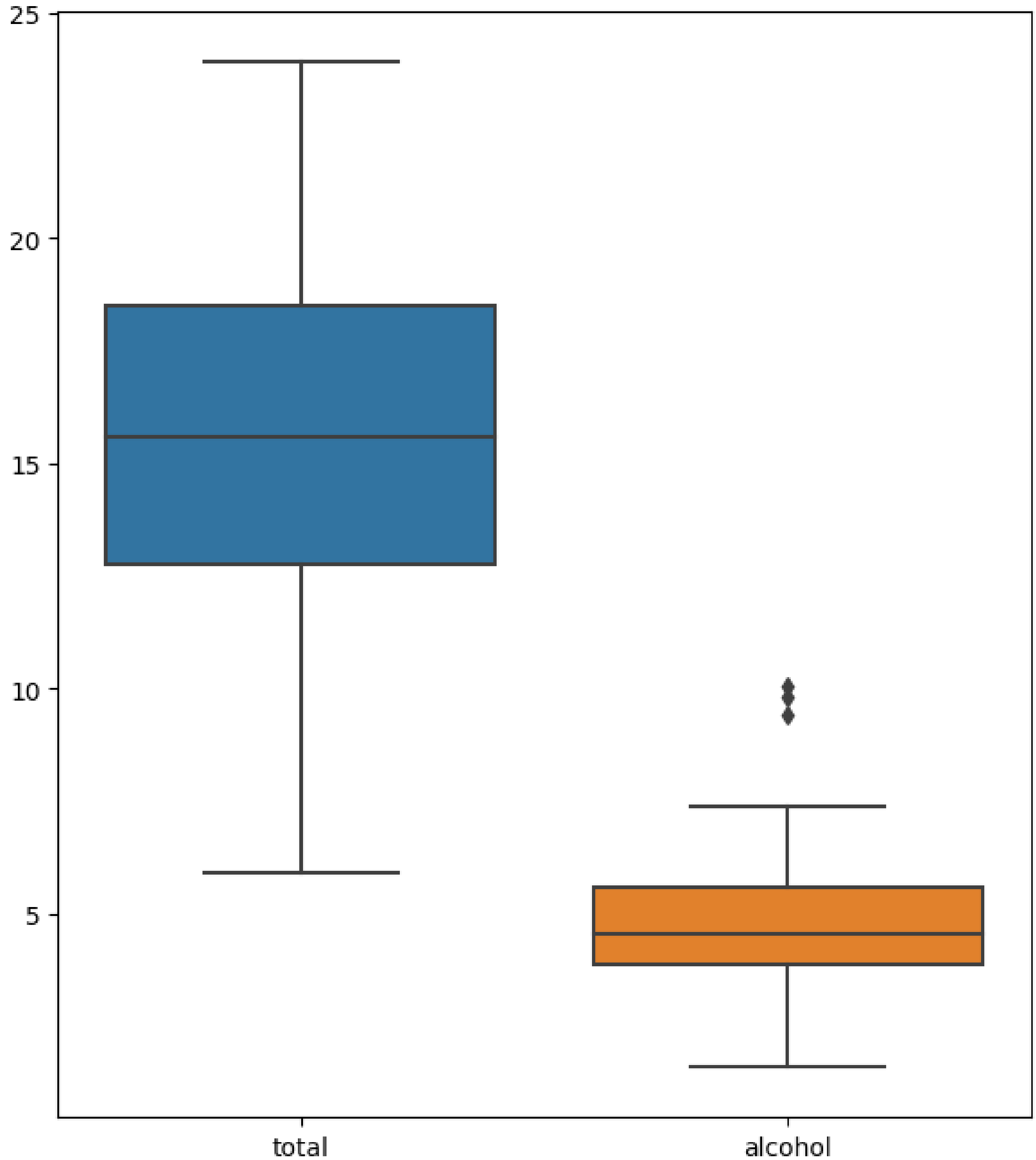

abbrev
AL
AK
AZ
AR
CA
CO
CT
DE
DC
FL
GA
HI
ID
IL
IN
IA
KS
KY
LA
ME
MD
MA
MI
MN
MS
MO
MT
NE
NV
NH
NJ
NM
NY
NC
ND
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VT
VA
WA
WV
WI
WY
alcohol
total
10
8
6
4
2
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5

[19]:
```
plt.figure(figsize=(20, 6))
sns.barplot(x="abbrev",y="ins_premium",data=df)
plt.show()
```



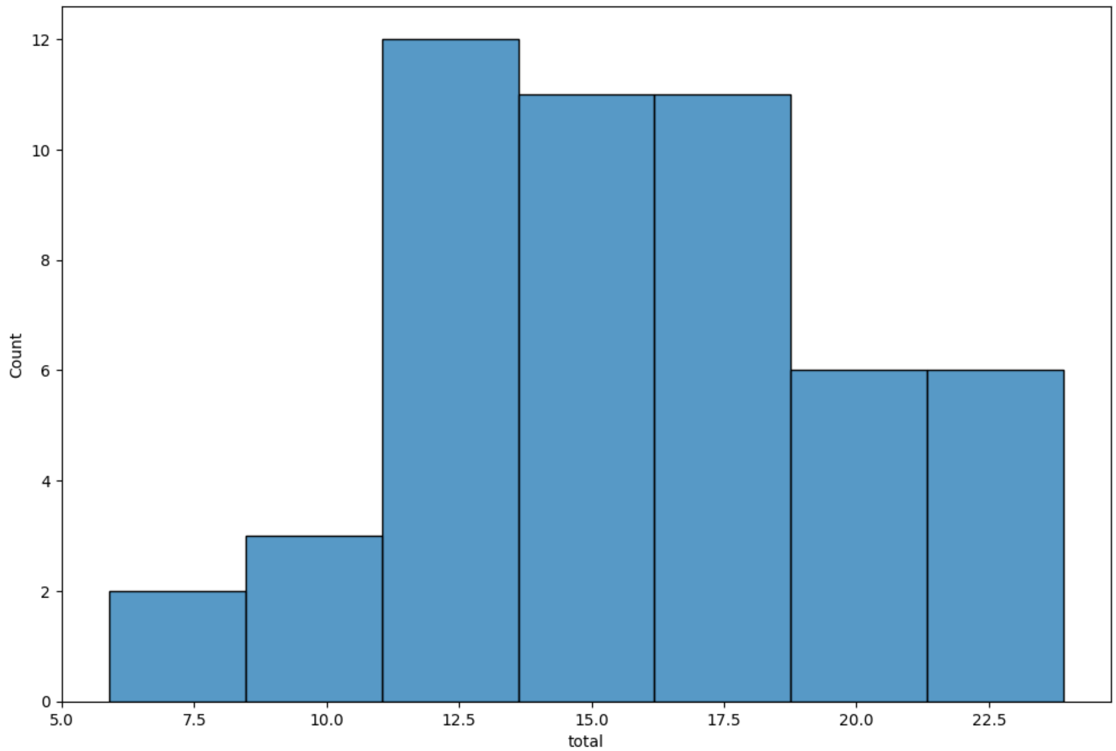
- Inference : In the LA and NJ there are more % of insurance premium

[21]:
```
boxplot_for = df[['total', 'alcohol']]
plt.figure(figsize=(7, 8))
sns.boxplot(data=boxplot_for)
plt.show()
```



- Inference : From the above boxplot , we can see a outliner between 9 and 11 (approximately)

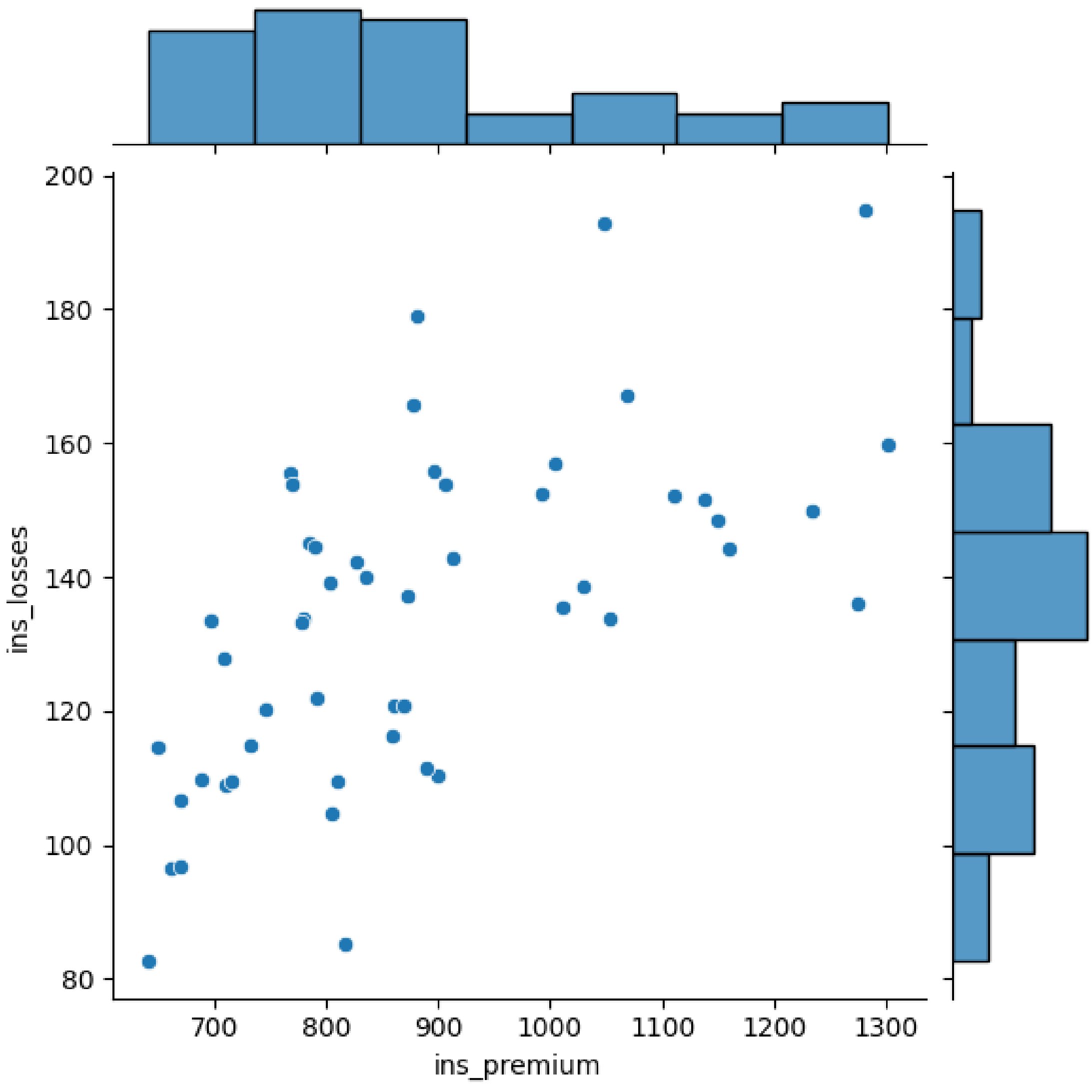
[24]:
```
plt.figure(figsize=(12, 8))
sns.histplot(x="total",data=df)
plt.show()
```



- Inference: At 12.5 the count reached highest than others in data set

[25]:
```
plt.figure(figsize=(17, 12))
sns.jointplot(x="ins_premium",y="ins_losses",data=df)
plt.show()
```

```
<Figure size 1700x1200 with 0 Axes>
```



- Inference: As the ins_premiums increases the ins_losses are also increasing (Nearly Directly proportional). This is a graph of combination of bivariate and univariate

[26]:
```
correlation_value = df.corr(numeric_only=True)
correlation_value
```

[26]:
```
                   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
```

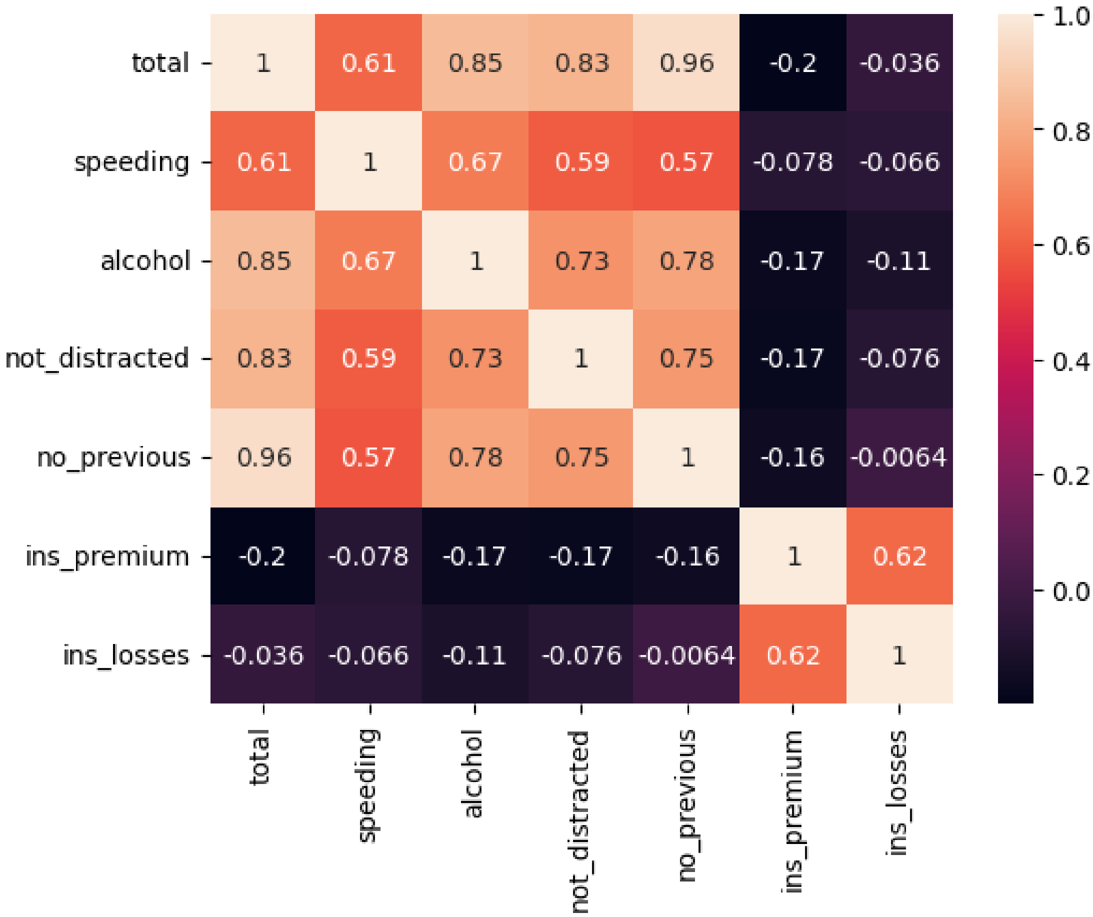
- Inference : From the corr() we can find all corellations values for each with other parameter how it was related

[27]:
```python
df[['total','alcohol']].corr()
```

[27]:
```
            total   alcohol
total    1.000000  0.852613
alcohol  0.852613  1.000000
```

[28]:
```python
sns.heatmap(correlation_value,annot=True)
```

[28]: <Axes: >



Inference : * Highly correlated : Total and no_previous * Neutrally correlated : None * Less correlated : Total and ins_premium