8TH_SEPT_ASSIGNMENT

NAME:- M.SUHAS

REG NO:- 21BCE7100

```
Assignment 8 th september
1.Take car crashes dataset from seaborn library
2.load the dataset
3.data visualiation
4.Inference is must for each and every graph
5.Submit it by wednesday in html format

Feedback - https://forms.gle/7vFfvANDVfvDxxW28
```

Steps:

1.import the necessary libraries 2.import the dataset 3.Handling null values 4.Seperate Dependent and independent variables 5.Encoding 6.splitting into training and testing set 7.Feature scaling

1.import the necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2.import the dataset

```
print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes', 'diamond
```

```
df=sns.load_dataset('car_crashes')
```

```
df
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losse
0	18.8	7.332	5.640	18.048	15.040	784.55	145.(
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.9
2	18.6	6.510	5.208	15.624	17.856	899.47	110.3
3	22.4	4.032	5.824	21.056	21.280	827.34	142.3
4	12.0	4.200	3.360	10.920	10.680	878.41	165.(
5	13.6	5.032	3.808	10.744	12.920	835.50	139.9
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.(
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.4
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.(
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.1
10	15.6	2.964	3.900	14.820	14.508	913.15	142.8
11	17.5	9.450	7.175	14.350	15.225	861.18	120.9
12	15.3	5.508	4.437	13.005	14.994	641.96	82.7
13	12.8	4.608	4.352	12.032	12.288	803.11	139.1
14	14.5	3.625	4.205	13.775	13.775	710.46	108.9
15	15.7	2.669	3.925	15.229	13.659	649.06	114.4
16	17.8	4.806	4.272	13.706	15.130	780.45	133.8
17	21.4	4.066	4.922	16.692	16.264	872.51	137.1
18	20.5	7.175	6.765	14.965	20.090	1281.55	194.7
19	15.1	5.738	4.530	13.137	12.684	661.88	96.5
20	12.5	4.250	4.000	8.875	12.375	1048.78	192.7
21	8.2	1.886	2.870	7.134	6.560	1011.14	135.(
22	14.1	3.384	3.948	13.395	10.857	1110.61	152.2
23	9.6	2.208	2.784	8.448	8.448	777.18	133.3
24	17.6	2.640	5.456	1.760	17.600	896.07	155.7
25	16.1	6.923	5.474	14.812	13.524	790.32	144.4
26	21.4	8.346	9.416	17.976	18.190	816.21	85.1
27	14.9	1.937	5.215	13.857	13.410	732.28	114.8
28	14.7	5.439	4.704	13.965	14.553	1029.87	138.7
29	11.6	4.060	3.480	10.092	9.628	746.54	120.2

30	11.2	1.792	3.136	9.632	8.736	1301.52	159.8
31	18.4	3.496	4.968	12.328	18.032	869.85	120.7
32	12.3	3.936	3.567	10.824	9.840	1234.31	150.0
33	16.8	6.552	5.208	15.792	13.608	708.24	127.8
34	23.9	5.497	10.038	23.661	20.554	688.75	109.7
35	14.1	3.948	4.794	13.959	11.562	697.73	133.5
36	19.9	6.368	5.771	18.308	18.706	881.51	178.8
37	12.8	4.224	3.328	8.576	11.520	804.71	104.6
38	18.2	9.100	5.642	17.472	16.016	905.99	153.8
39	11.1	3.774	4.218	10.212	8.769	1148.99	148.5
40	23.9	9.082	9.799	22.944	19.359	858.97	116.2
41	19.4	6.014	6.402	19.012	16.684	669.31	96.8
42	19.5	4.095	5.655	15.990	15.795	767.91	155.5
43	19.4	7.760	7.372	17.654	16.878	1004.75	156.8
44	11.3	4.859	1.808	9.944	10.848	809.38	109.4
45	13.6	4.080	4.080	13.056	12.920	716.20	109.6
46	12.7	2.413	3.429	11.049	11.176	768.95	153.7
47	10.6	4.452	3.498	8.692	9.116	890.03	111.6
48	23.8	8.092	6.664	23.086	20.706	992.61	152.5
49	13.8	4.968	4.554	5.382	11.592	670.31	106.6

```
sns.__version__
```

```
'0.12.2'
```

```
dataset.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()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63

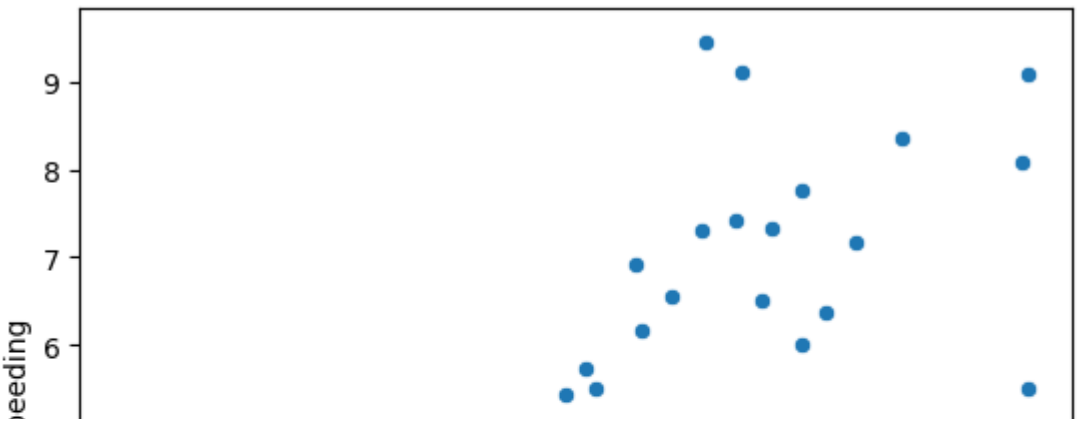
```
df.shape
```

```
(51, 8)
```

```
sns.scatterplot(x="total",y="speeding",data=df)
```

```
print("as speed increases total also increases")
```

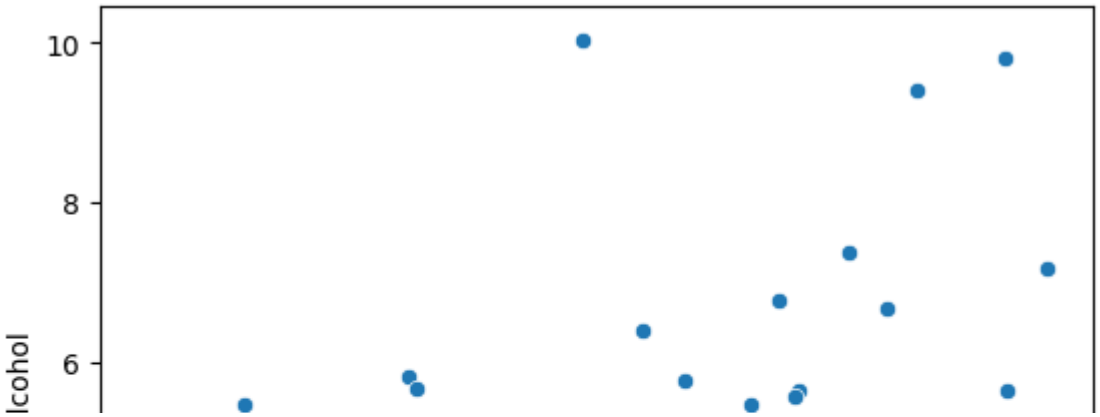
```
as speed increases total also increases
```




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

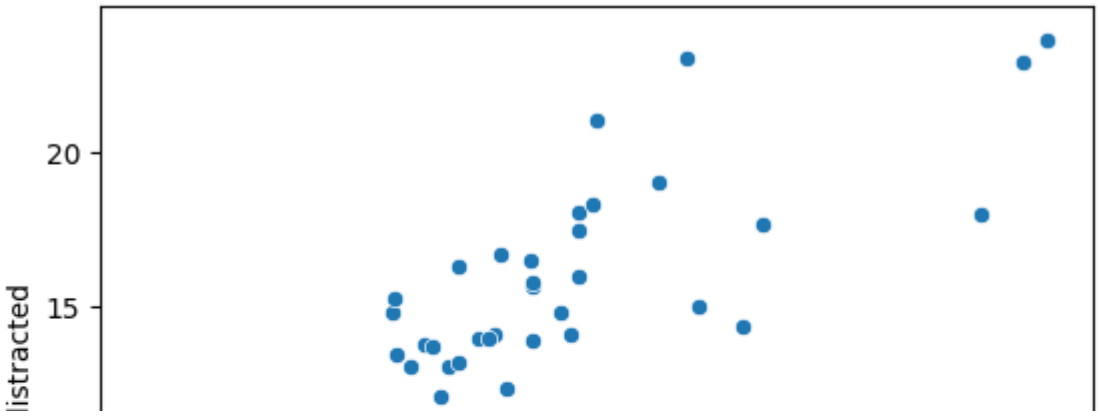
```
print("as alcohol increases speeding also increases")
```

```
as alcohol increases speeding also increases
```



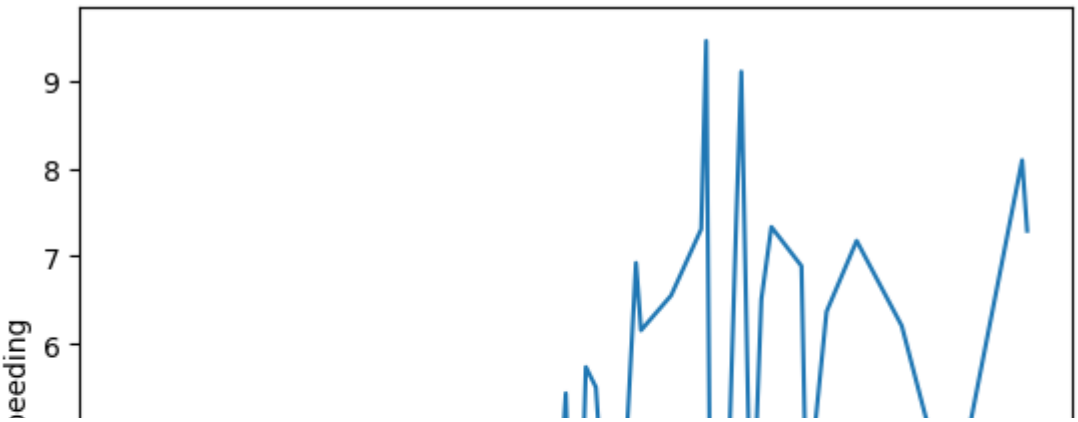

```
sns.scatterplot(x="alcohol",y="not_distracted",data=df)
print("as alcohol increases not_distracted also increases")
```

```
as alcohol increases not_distracted also increases
```



```
sns.lineplot(x="total",y="speeding",data=df,ci=None)
print("here we can see fulcutations but still its they are related")
```

```
<ipython-input-107-7884c730efd4>:1: FutureWarning:

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

  sns.lineplot(x="total",y="speeding",data=df,ci=None)
here we can see fulcutations but still its they are related
```

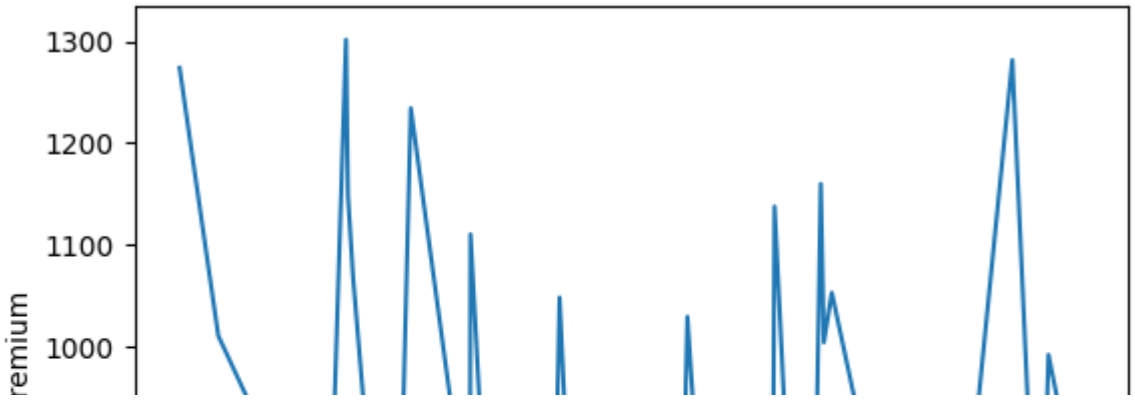



```
sns.lineplot(x="alcohol",y="not_distracted",data=df,ci=None)
print("here we can see fulcutations but still its they are related")
```

```
<ipython-input-109-ee6c7a76b6a6>:1: FutureWarning:

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

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

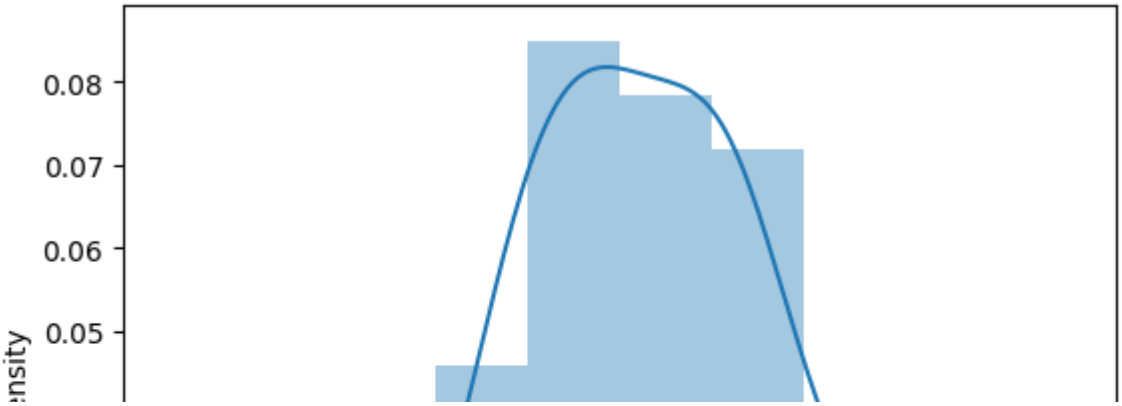
```
sns.lineplot(x="no_previous",y="ins_premium",data=df,ci=None)
print("here we can see fulcutations but still its they are related")
```

```
<ipython-input-110-0c2ff91edc84>:1: FutureWarning:

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

  sns.lineplot(x="no_previous",y="ins_premium",data=df,ci=None)
here we can see fulcutations but still its they are related
```



```
sns.distplot(df["total"])
print("data distribution of a total against the density distribution")
```

```
<ipython-input-115-1eedb4648a09>: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"])
data distribution of a total against the density distribution
```

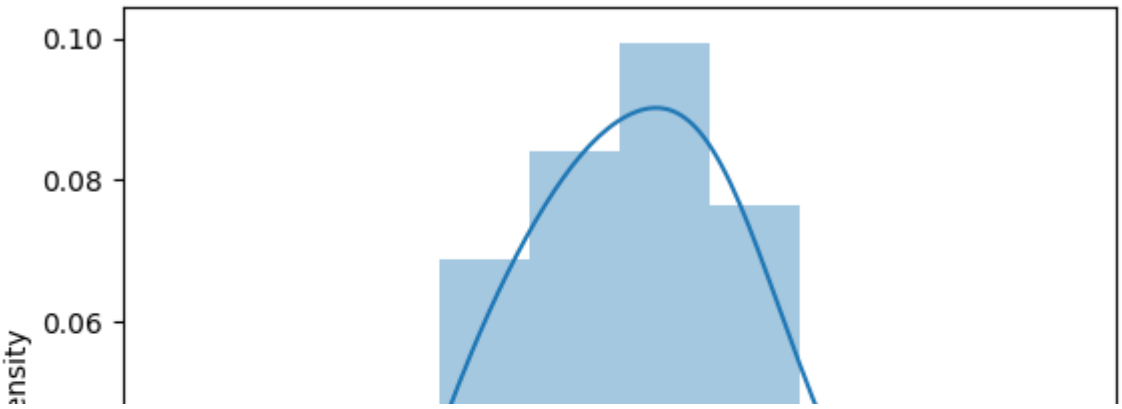



```
sns.distplot(df["alcohol"])
print("data distribution of alcohol against the density distribution")
```

```
<ipython-input-117-6f9e972c38d5>: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"])
data distribution of a alcohol against the density distribution
```

```
sns.distplot(df["no_previous"])
print("data distribution of a no_previous against the density distribution")
```

```
<ipython-input-118-84f63674cc34>: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["no_previous"])
data distribution of a no_previous against the density distribution
```



```
sns.relplot(x="total",y="speeding",data=df,hue="alcohol")
print("using hue we differentiated different categories with colors")
```

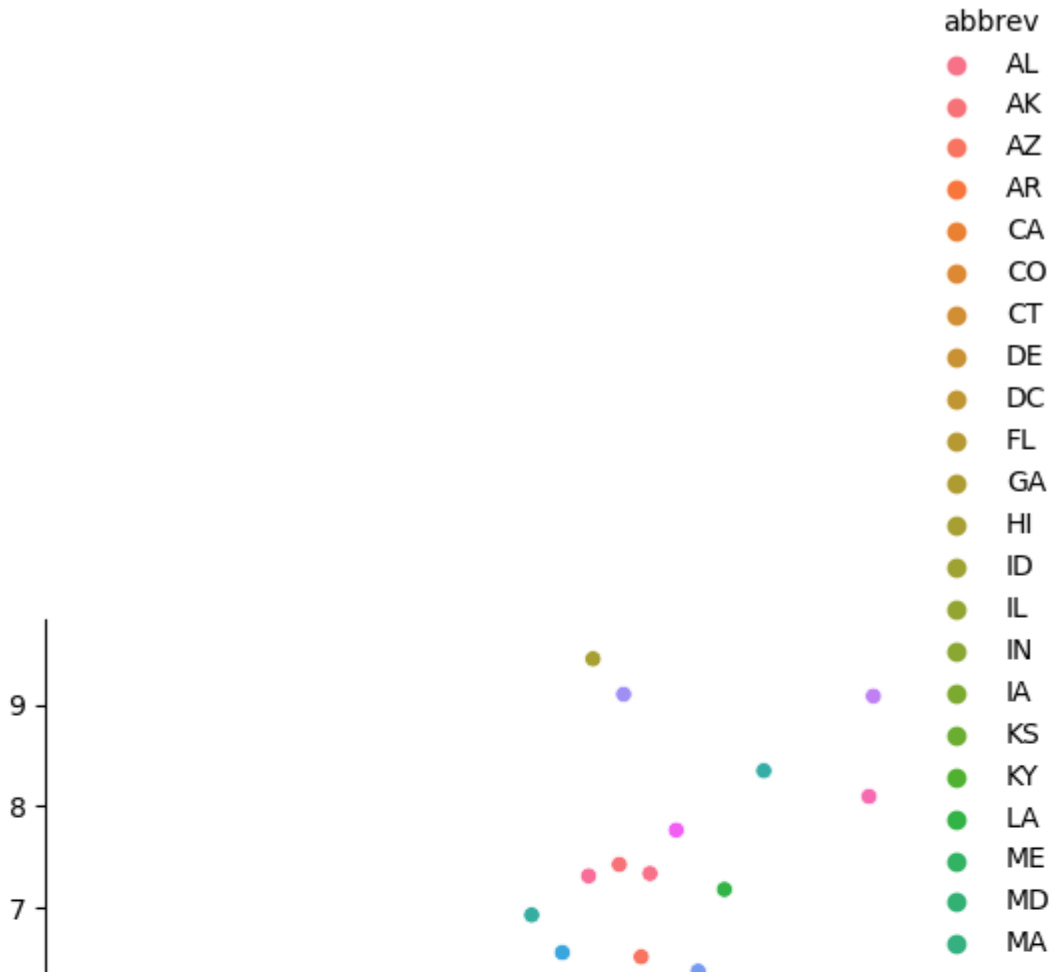
```
using hue we differentiated different categories with colors
```


```
sns.relplot(x="not_distracted",y="no_previous",data=df,hue="ins_premium")
print("using hue we differentiated different categories with colors")
```

using hue we differentiated different categories with colors


```
sns.relplot(x="total",y="speeding",data=df,hue="abbrev")
print("using hue we differentiated different categories with colors")
```

using hue we differentiated different categories with colors



```
print("here we get count of all categories. has everything repeated ")
```

```
      here we get count of all categories. has everything repeated
```

```
df["speeding"].value_counts()
```

```
      4.968    2
      7.332    1
      9.100    1
      5.439    1
      4.060    1
      1.792    1
      3.496    1
      3.936    1
      6.552    1
      5.497    1
      3.948    1
      6.368    1
      4.224    1
      3.774    1
      8.346    1
      9.082    1
      6.014    1
      4.095    1
```

```
7.760    1
4.859    1
4.080    1
2.413    1
4.452    1
8.092    1
1.937    1
6.923    1
7.421    1
2.640    1
6.510    1
4.032    1
4.200    1
5.032    1
6.156    1
2.006    1
3.759    1
2.964    1
9.450    1
5.508    1
4.608    1
3.625    1
2.669    1
4.806    1
4.066    1
7.175    1
5.738    1
4.250    1
1.886    1
3.384    1
2.208    1
7.308    1
Name: speeding, dtype: int64
```

```
df["not_distracted"].value_counts()
```

```
14.094    2
18.048    1
17.472    1
13.965    1
10.092    1
9.632     1
12.328    1
10.824    1
15.792    1
23.661    1
13.959    1
18.308    1
8.576     1
10.212    1
17.976    1
22.944    1
19.012    1
```

```
15.990    1
17.654    1
9.944     1
13.056    1
11.049    1
8.692     1
23.086    1
13.857    1
14.812    1
16.290    1
1.760     1
15.624    1
21.056    1
10.920    1
10.744    1
9.396     1
5.900     1
16.468    1
14.820    1
14.350    1
13.005    1
12.032    1
13.775    1
15.229    1
13.706    1
16.692    1
14.965    1
13.137    1
8.875     1
7.134     1
13.395    1
8.448     1
5.382     1
Name: not_distracted, dtype: int64
```

```
df["no_previous"].value_counts()
```

```
12.920    2
15.040    1
16.016    1
14.553    1
9.628     1
8.736     1
18.032    1
9.840     1
13.608    1
20.554    1
11.562    1
18.706    1
11.520    1
8.769     1
18.190    1
19.359    1
16.684    1
15.795    1
```

```
16.878    1
10.848    1
11.176    1
9.116     1
20.706    1
11.592    1
13.410    1
13.524    1
17.014    1
17.600    1
17.856    1
21.280    1
10.680    1
8.856     1
16.038    1
5.900     1
16.826    1
14.508    1
15.225    1
14.994    1
12.288    1
13.775    1
13.659    1
15.130    1
16.264    1
20.090    1
12.684    1
12.375    1
6.560     1
10.857    1
8.448     1
15.660    1
Name: no_previous, dtype: int64
```

```
print("Bargraph :total vs sppeding")
print("Bargraph : alcohol vs not_distracted ")
print("Bargraph :no_previous vs ins_premium ")
```

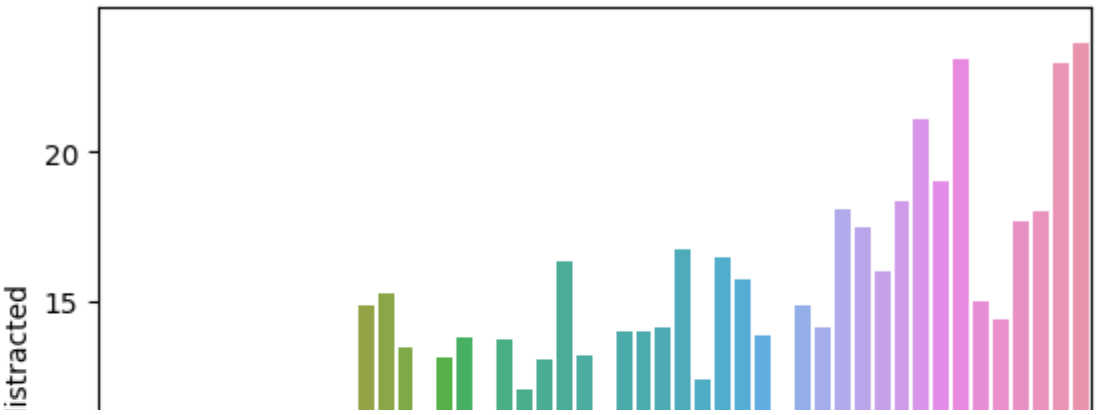
```
Bargraph :total vs sppeding
Bargraph : alcohol vs not_distracted
Bargraph :no_previous vs ins_premium
```

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

```
<ipython-input-84-45580ba4c45b>:1: FutureWarning:

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

  sns.barplot(data=df,x="total",y="speeding",ci=None)
<Axes: xlabel='total', ylabel='speeding'>
```

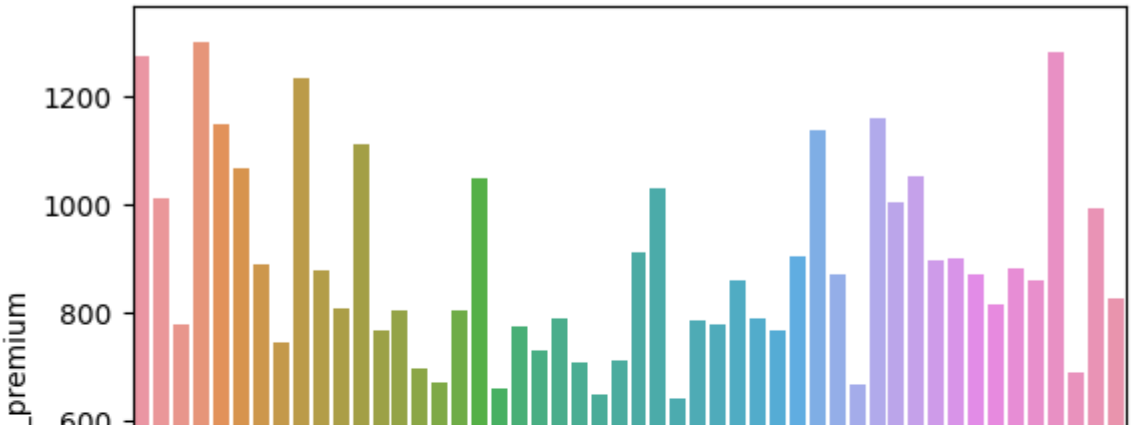
```
sns.barplot(data=df,x="alcohol",y="not_distracted",ci=None)
```

```
<ipython-input-85-c836539ef2b1>:1: FutureWarning:

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

  sns.barplot(data=df,x="alcohol",y="not_distracted",ci=None)
<Axes: xlabel='alcohol', ylabel='not_distracted'>
```



```
sns.barplot(data=df,x="no_previous",y="ins_premium",ci=None)
```

```
<ipython-input-86-560e315f829c>:1: FutureWarning:

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

  sns.barplot(data=df,x="no_previous",y="ins_premium",ci=None)
<Axes: xlabel='no_previous', ylabel='ins_premium'>
```




```
sns.barplot(data=df,x="alcohol",y="not_distracted",hue="no_previous")
```

```
<Axes: xlabel='alcohol', ylabel='not_distracted'>
```


```
sns.barplot(data=df,x="alcohol",y="not_distracted",hue="ins_premium")
```

```
<Axes: xlabel='alcohol', ylabel='not_distracted'>
```


```
print("countplot for : total ")
print("countplot for : alcohol ")
print("countplot for : not_distracted ")
```

```
      countplot for : total
      countplot for : alcohol
      countplot for : not_distracted
```

```
sns.countplot(x="total",data=df)
```

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


```
sns.countplot(x="alcohol",data=df)
```

```
<Axes: xlabel='alcohol', ylabel='count'>
```


```
sns.countplot(x="not_distracted",data=df)
```

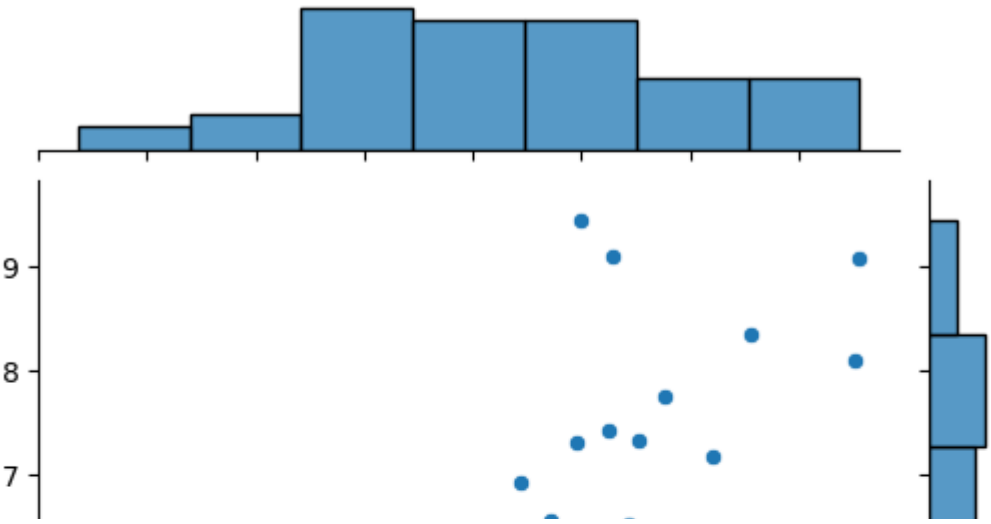
```
<Axes: xlabel='not_distracted', ylabel='count'>
```


```
print("jointplot for : total vs speeding ")
print("jointplot for : alcohol  vs no previous  ")
print("jointplot for : not_distracted  vs no previous ")
```

```
jointplot for : total vs speeding
jointplot for : alcohol  vs no previous
jointplot for : not_distracted  vs no previous
```

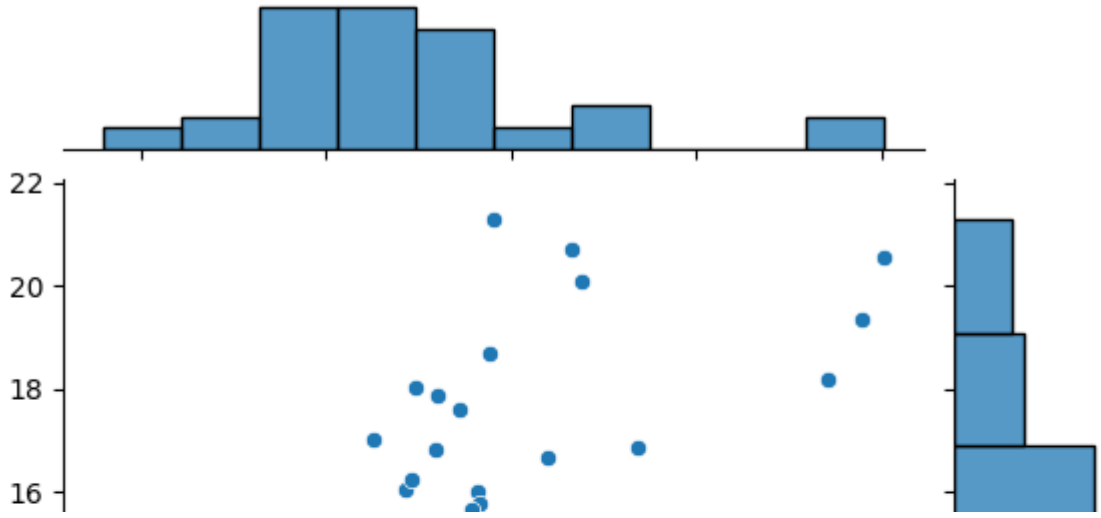
```
sns.jointplot(x="total",y="speeding",data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7d15d2a4d5d0>
```



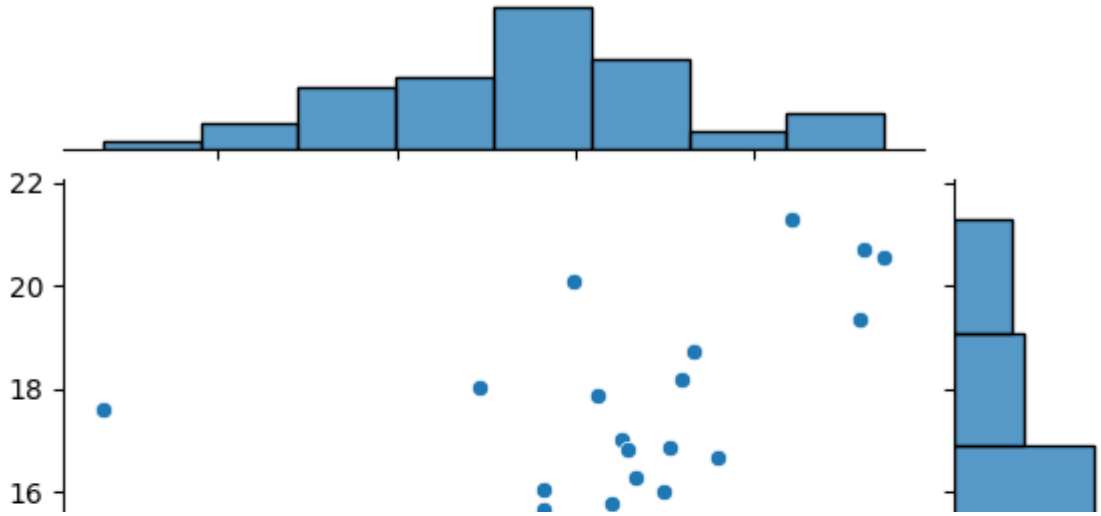
```
sns.jointplot(x="alcohol",y="no_previous",data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7d15befca4a0>
```



```
sns.jointplot(x="not_distracted",y="no_previous",data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7d15bef095a0>
```

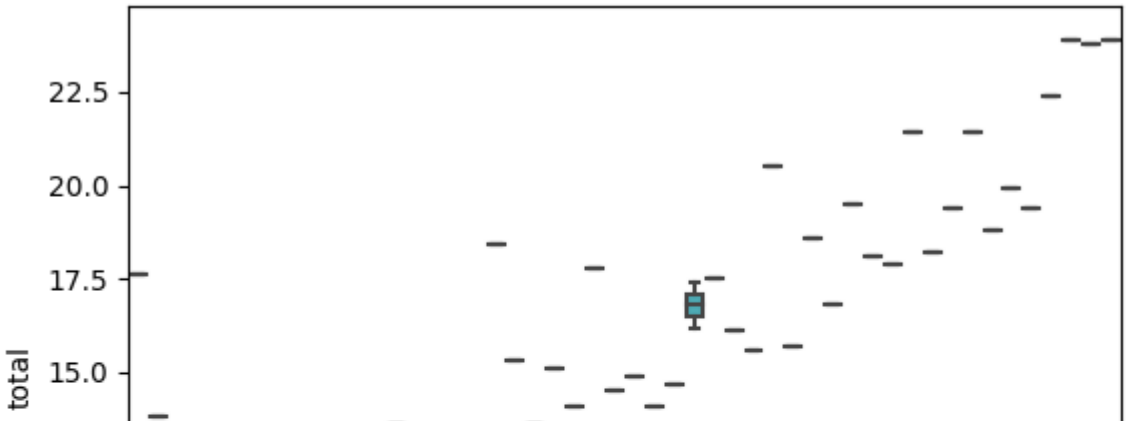



```
sns.boxplot(x="total",y="speeding",data=df)
```

```
<Axes: xlabel='total', ylabel='speeding'>
```

```
sns.boxplot(x="not_distracted",y="total",data=df)
```

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




```
sns.boxplot(x="not_distracted",y="ins_premium",data=df)
```

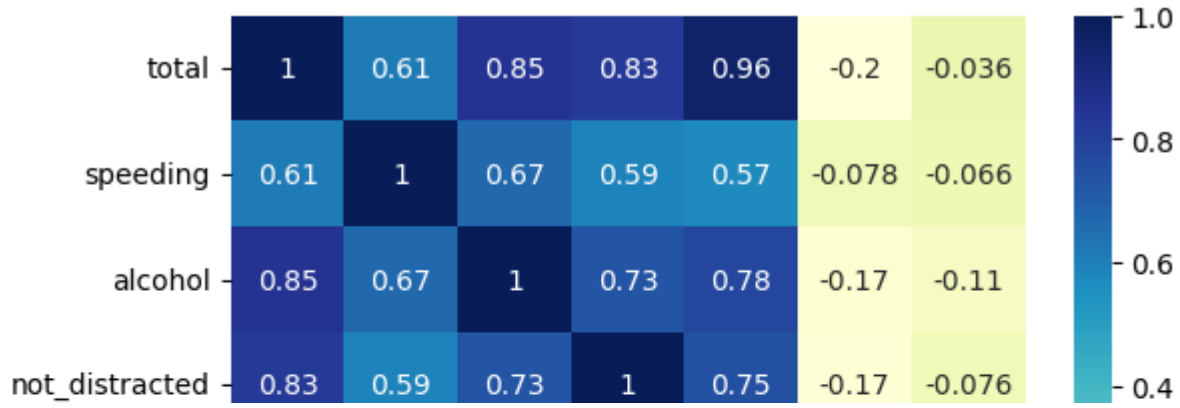
```
corr=df.corr()
corr
```

```
<ipython-input-98-7d5195e2bf4d>:1: FutureWarning: The default value of numeric_
  corr=df.corr()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premiu
total	1.000000	0.611548	0.852613	0.827560	0.956179	-0.1997(
speeding	0.611548	1.000000	0.669719	0.588010	0.571976	-0.07767
alcohol	0.852613	0.669719	1.000000	0.732816	0.783520	-0.17061
not_distracted	0.827560	0.588010	0.732816	1.000000	0.747307	-0.17485
no_previous	0.956179	0.571976	0.783520	0.747307	1.000000	-0.1568(
ins_premium	-0.199702	-0.077675	-0.170612	-0.174856	-0.156895	1.00000(
ins_losses	-0.036011	-0.065928	-0.112547	-0.075970	-0.006359	0.62311

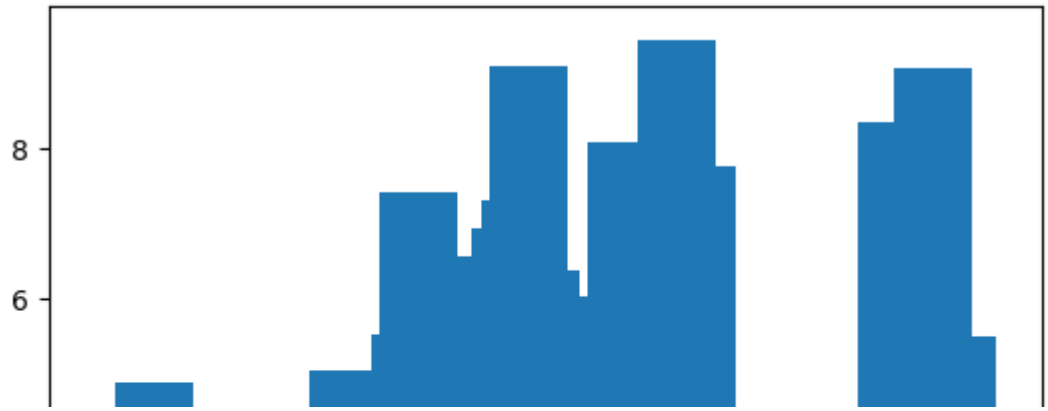
```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

```
<Axes: >
```



```
x=df["alcohol"]
y = df["speeding"]
plt.bar(x,y)
```

```
<BarContainer object of 51 artists>
```




```
plt.barh(x,y)
```

```
<BarContainer object of 51 artists>
```

```
plt.barh(x,y,color = 'black')
```

```
<BarContainer object of 51 artists>
```


```
plt.bar(x,y,width = 0.25)
```

```
<BarContainer object of 51 artists>
```

```
plt.hist(x)
```

```
(array([ 2.,  3., 13., 13., 11.,  2.,  4.,  0.,  0.,  3.]),
 array([ 1.593 ,  2.4375,  3.282 ,  4.1265,  4.971 ,  5.8155,  6.66  ,
         7.5045,  8.349 ,  9.1935, 10.038 ]),
 <BarContainer object of 10 artists>)
```


```
x1 =(df["alcohol"])
fig = plt.figure()
axes1 = fig.add_axes([0.1,0.1,0.8,0.8])
axes1.pie(x1,y,autopct="%0.2f%%",colors=["red","green"])
```

```
([<matplotlib.patches.Wedge at 0x7d15bcce7730>,
  <matplotlib.patches.Wedge at 0x7d15bcce76a0>,
  <matplotlib.patches.Wedge at 0x7d15bcce7670>,
  <matplotlib.patches.Wedge at 0x7d15bcd10b20>,
  <matplotlib.patches.Wedge at 0x7d15bcd111b0>,
  <matplotlib.patches.Wedge at 0x7d15bcd11840>,
  <matplotlib.patches.Wedge at 0x7d15bcd11ea0>,
  <matplotlib.patches.Wedge at 0x7d15bcd12530>,
  <matplotlib.patches.Wedge at 0x7d15bcd12bc0>,
  <matplotlib.patches.Wedge at 0x7d15bcd13250>,
  <matplotlib.patches.Wedge at 0x7d15bcd138e0>,
  <matplotlib.patches.Wedge at 0x7d15bcd13f70>,
  <matplotlib.patches.Wedge at 0x7d15bcb58640>,
  <matplotlib.patches.Wedge at 0x7d15bcb58cd0>,
  <matplotlib.patches.Wedge at 0x7d15bcb591b0>,
  <matplotlib.patches.Wedge at 0x7d15bcb59840>,
  <matplotlib.patches.Wedge at 0x7d15bcb59ed0>,
  <matplotlib.patches.Wedge at 0x7d15bcb5a560>,
  <matplotlib.patches.Wedge at 0x7d15bcb5abf0>,
  <matplotlib.patches.Wedge at 0x7d15bcb5b280>,
  <matplotlib.patches.Wedge at 0x7d15bcb5b910>,
  <matplotlib.patches.Wedge at 0x7d15bcb5bfa0>,
  <matplotlib.patches.Wedge at 0x7d15bcb90670>,
  <matplotlib.patches.Wedge at 0x7d15bcb90d00>,
  <matplotlib.patches.Wedge at 0x7d15bcb91390>,
  <matplotlib.patches.Wedge at 0x7d15bcb91a20>,
  <matplotlib.patches.Wedge at 0x7d15bcb920b0>,
  <matplotlib.patches.Wedge at 0x7d15bcb92740>,
  <matplotlib.patches.Wedge at 0x7d15bcb92dd0>,
  <matplotlib.patches.Wedge at 0x7d15bcb93460>,
  <matplotlib.patches.Wedge at 0x7d15bcb93af0>,
  <matplotlib.patches.Wedge at 0x7d15bcbd41c0>,
  <matplotlib.patches.Wedge at 0x7d15bcbd4850>,
  <matplotlib.patches.Wedge at 0x7d15bcbd4ee0>,
  <matplotlib.patches.Wedge at 0x7d15bcbd5570>,
  <matplotlib.patches.Wedge at 0x7d15bcbd5c00>,
  <matplotlib.patches.Wedge at 0x7d15bcbd6290>,
  <matplotlib.patches.Wedge at 0x7d15bcbd6920>,
  <matplotlib.patches.Wedge at 0x7d15bcbd6fb0>,
  <matplotlib.patches.Wedge at 0x7d15bcbd7640>,
  <matplotlib.patches.Wedge at 0x7d15bcbd7cd0>,
  <matplotlib.patches.Wedge at 0x7d15bcc0c3a0>,
  <matplotlib.patches.Wedge at 0x7d15bcc0ca30>,
  <matplotlib.patches.Wedge at 0x7d15bcc0d0c0>,
  <matplotlib.patches.Wedge at 0x7d15bcc0d750>,
  <matplotlib.patches.Wedge at 0x7d15bcc0dde0>,
  <matplotlib.patches.Wedge at 0x7d15bcc0e470>,
  <matplotlib.patches.Wedge at 0x7d15bcc0eb00>,
  <matplotlib.patches.Wedge at 0x7d15bcc0f190>,
  <matplotlib.patches.Wedge at 0x7d15bcc0f820>,
  <matplotlib.patches.Wedge at 0x7d15bcc0feb0>],
 [Text(8.410699544034177, 0.5989634212398267, ''),
  Text(8.352451213487306, 1.686416237562493, ''),
  Text(7.219080145466262, 2.4076922256249467, ''),
  Text(4.596307879757315, 2.282844251034401, ''),
  Text(4.442676050348083, 2.8900916095625004, ''),
```

```
 Text(4.8174064272344586, 3.7939450859020254, ''),
 Text(4.381069891848447, 4.198434303730266, ''),
 Text(4.654492093706887, 5.566438659468016, ''),
 Text(1.7922050148895288, 2.5367769284280444, ''),
 Text(2.4545111102703068, 4.193477817940573, ''),
 Text(1.638404163601255, 3.7191030903558016, ''),
 Text(2.868396865351445, 10.15257599936302, ''),
 Text(0.8499282227709963, 6.553112696737128, ''),
 Text(0.1038203367441988, 5.7070557503565995, ''),
 Text(-0.423145844593834, 4.706014512748869, ''),
 Text(-0.7197898051735776, 3.699630202651095, ''),
 Text(-1.7198394510077382, 5.650043208928353, ''),
 Text(-2.065738552191629, 4.73500583251903, ''),
 Text(-4.386422626721849, 7.016760045760621, ''),
 Text(-4.410777001228907, 5.225255041185084, ''),
 Text(-3.8697624080585697, 3.6942440235015255, ''),
 Text(-2.3300731564046595, 1.8673390387935518, ''),
 Text(-3.7267978811514566, 2.493438098899793, ''),
 Text(-2.895404069340398, 1.5998435158624316, ''),
 Text(-3.4434161930670757, 1.4596180737862385, ''),
 Text(-7.746810771566733, 2.086971937889355, ''),
 Text(-9.419121928902934, 0.7120801137932684, ''),
 Text(-3.0189852501569154, -0.33029843980102497, ''),
 Text(-6.360780624724265, -1.516242343470325, ''),
 Text(-4.869465826212241, -1.7070156904232423, ''),
 Text(-2.639984763311166, -1.1807389421395764, ''),
 Text(-3.982269204473158, -2.2945038642601414, ''),
 Text(-4.068313434138797, -2.968184933862071, ''),
 Text(-5.646011854385671, -5.1648479300107635, ''),
 Text(-3.927487844397131, -5.30049509311279, ''),
 Text(-2.1990287284597403, -4.543850421328671, ''),
 Text(-2.331844511439544, -7.0946124048089665, ''),
 Text(-1.072627015483695, -5.214829554803737, ''),
 Text(-0.9146094804645407, -10.15891182648242, ''),
 Text(0.16812463279483905, -4.871099476283316, ''),
 Text(2.134396256831493, -9.95577604302265, ''),
 Text(2.8709679729381836, -6.508958357399686, ''),
 Text(2.7919819704048128, -4.380965838366519, ''),
 Text(5.919051915713311, -6.592755449665229, ''),
 Text(4.466332693027959, -3.94481346519574, ''),
 Text(4.126053075319439, -3.131786394319861, ''),
 Text(2.9864412366694113, -1.8499561454046631, ''),
 Text(4.956803039195598, -2.5009613412888423, ''),
 Text(8.668293375132498, -3.0583580501036915, ''),
 Text(5.949707247499431, -1.1923119009104735, ''),
 Text(8.387298811658315, -0.5896462023578266, '')],
[Text(7.911962616612795, 0.5634461405685846, '2.26%'),
 Text(7.862341413376562, 1.5874597631133383, '1.82%'),
 Text(6.744764761401462, 2.249499569539208, '2.09%'),
 Text(4.14849923987449, 2.0604315219780487, '2.34%'),
 Text(4.023555668239773, 2.6174414577169816, '1.35%'),
 Text(4.424597683983117, 3.4845888329745933, '1.53%'),
 Text(4.020072043146366, 3.8524855311750357, '1.56%'),
 Text(4.333758074019257, 5.182863779405446, '1.95%'),
 Text(1.5036980904063462, 2.1284097474190222, '0.64%'),
 Text(2.2019374212118272, 3.761961269479925, '2.08%'),
 Text(1.4368288481975573, 3.26153627313683, '1.56%'),
```

```
Text(2.7324538859509024, 9.671411260056715, '2.88%'),
Text(0.785617673227186, 6.057265791717672, '1.78%'),
Text(0.09472605356758712, 5.207138463184507, '1.75%'),
Text(-0.3783685065415764, 4.208023559018829, '1.69%'),
Text(-0.6243016378647984, 3.208832882055301, '1.57%'),
Text(-1.5742384138414887, 5.171712425917148, '1.71%'),
Text(-1.865802571530418, 4.276720328016607, '1.97%'),
Text(-4.121381984623852, 6.592786629098348, '2.71%'),
Text(-4.088257477886636, 4.843180235599746, '1.82%'),
Text(-3.5081023699222547, 3.3489875727069904, '1.60%'),
Text(-1.9399068542605438, 1.5546566813264464, '1.15%'),
Text(-3.311231658899956, 2.2154008443391557, '1.58%'),
Text(-2.4577674204074476, 1.358029199680081, '1.12%'),
Text(-2.9830664346356484, 1.2644819676650836, '2.19%'),
Text(-7.264023112862587, 1.9569101194991425, '2.20%'),
Text(-8.920544651277329, 0.6743879629467054, '3.78%'),
Text(-2.5219511292881442, -0.27591937496713875, '2.09%'),
Text(-5.8744080429285574, -1.400303947425798, '1.89%'),
Text(-4.3976183624319845, -1.541607193289207, '1.40%'),
Text(-2.1835558623237583, -0.976600120884463, '1.26%'),
Text(-3.549037132620116, -2.044884209749682, '1.99%'),
Text(-3.664390337024143, -2.6734882565524933, '1.43%'),
Text(-5.277087922447246, -4.827364400867353, '2.09%'),
Text(-3.629815580913947, -4.8987598276047715, '4.03%'),
Text(-1.9812168496503364, -4.093785997662994, '1.92%'),
Text(-2.1757220883383424, -6.619611574278104, '2.32%'),
Text(-0.9718919464112217, -4.725082226215857, '1.34%'),
Text(-0.8697756824025534, -9.660925952635242, '2.26%'),
Text(0.15087754284871277, -4.371397026931314, '1.69%'),
Text(2.029584026580487, -9.466885056820399, '3.93%'),
Text(2.6691850116689833, -6.051483072229621, '2.57%'),
Text(2.523263782685389, -3.9593136883793667, '2.27%'),
Text(5.58501964056019, -6.220703787720239, '2.96%'),
Text(4.091577474616484, -3.6138172019640113, '0.73%'),
Text(3.7277854039565588, -2.8294904103121525, '1.64%'),
Text(2.561385552543392, -1.5866546729587956, '1.38%'),
Text(4.5104050709683285, -2.2757306729451066, '1.40%'),
Text(8.19678046308221, -2.89199827801363, '2.67%'),
Text(5.459454507923011, -1.0940660290490303, '1.83%'),
Text(7.888529852829919, -0.5545816089730843, '2.23%')])
```