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

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

```
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']
```

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

```
dataset.head(10)
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	ab
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	
5	13.6	5.032	3.808	10.744	12.920	835.50	139.91	
6	10.8	4.968	3.888	9.396	8.856	1068.73	167.02	
7	16.2	6.156	4.860	14.094	16.038	1137.87	151.48	
8	5.9	2.006	1.593	5.900	5.900	1273.89	136.05	
9	17.9	3.759	5.191	16.468	16.826	1160.13	144.18	

```
dataset.tail()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	a
46	12.7	2.413	3.429	11.049	11.176	768.95	153.72	

```
dataset.shape
```

```
(51, 8)
```

49	13.8	4.968	4.554	5.382	11.592	670.31	106.62

```
dataset.info()
```

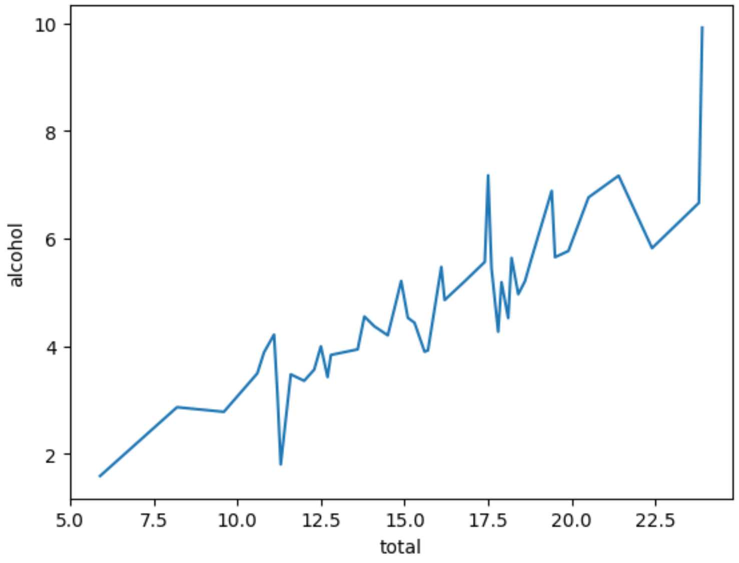
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   total           51 non-null     float64
 1   speeding        51 non-null     float64
 2   alcohol         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
```

```
dataset.describe()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_
count	51.000000	51.000000	51.000000	51.000000	51.000000	51.000000	51.0
mean	15.790196	4.998196	4.886784	13.573176	14.004882	886.957647	134.4
std	4.122002	2.017747	1.729133	4.508977	3.764672	178.296285	24.8
min	5.900000	1.792000	1.593000	1.760000	5.900000	641.960000	82.7
25%	12.750000	3.766500	3.894000	10.478000	11.348000	768.430000	114.6
50%	15.600000	4.608000	4.554000	13.857000	13.775000	858.970000	136.0
75%	18.500000	6.439000	5.604000	16.140000	16.755000	1007.945000	151.8
max	23.900000	9.450000	10.038000	23.661000	21.280000	1301.520000	194.7

```
sns.lineplot(x="total", y="alcohol", data=dataset, ci=None)

# INFERENCE:
# Describes the relation between total crashes and amount of alcohol consumed by driver
```

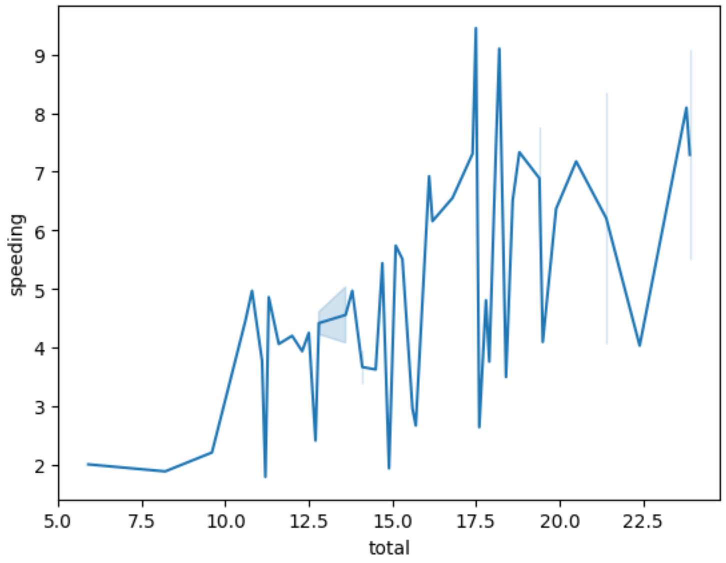
```
<ipython-input-15-af97fdff5ed0>:1: FutureWarning:

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

  sns.lineplot(x="total", y="alcohol", data=dataset, ci=None)
<Axes: xlabel='total', ylabel='alcohol'>
```



```
sns.lineplot(x="total", y="speeding", data=dataset,)

# INFERENCE:
```

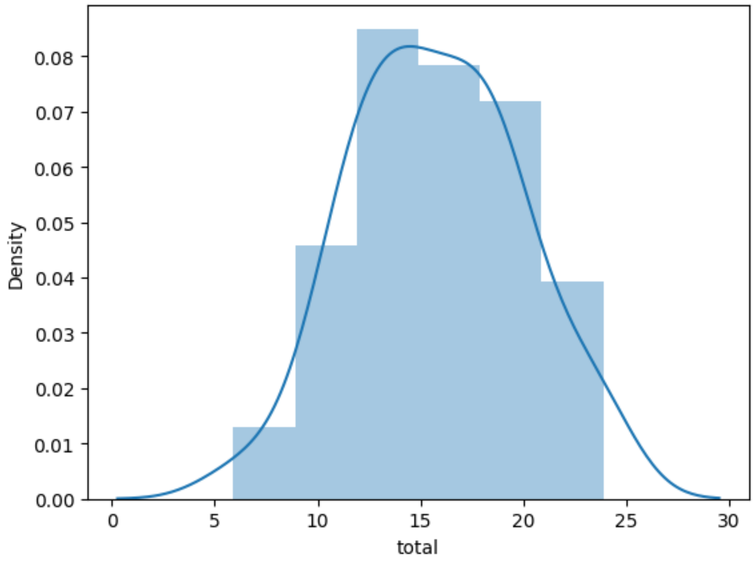
```
# Describes a non-uniform unevenly directly proportional relation between total crashes and speed of vehicle
```

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



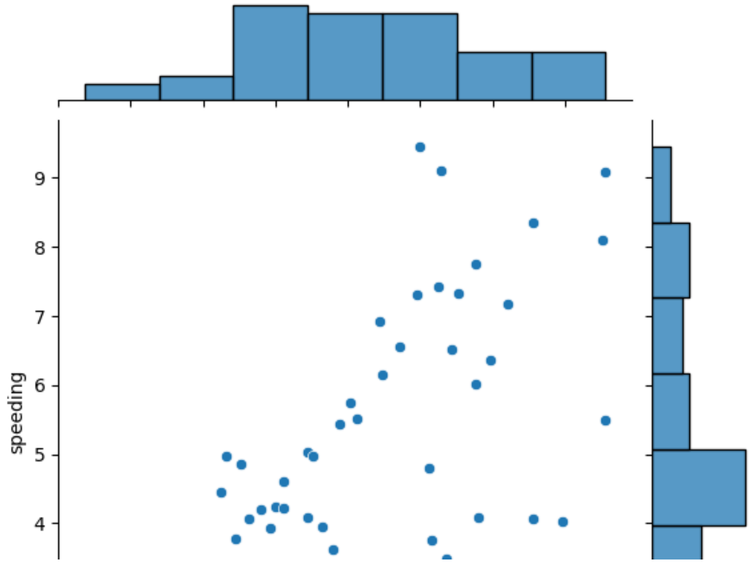
```
sns.distplot(dataset["total"])

# INFERENCE
# distplot = histogram + kernal density function of total crashes
```

```
<ipython-input-17-9194453daee3>: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(dataset["total"])
<Axes: xlabel='total', ylabel='Density'>
```



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

# INFERENCE
# Univariate analysis of total crashes and speeding in form of histograms
# And Bivariate analysis of total crashes with respect to speeding in form of scatter plot
```

```
<seaborn.axisgrid.JointGrid at 0x794a19fabc70>
```



HEATMAP

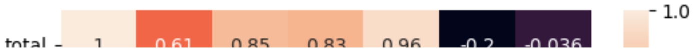
```
corr=dataset.corr()
corr
```

```
<ipython-input-9-f22ca9e9dc13>:1: FutureWarning: The default value of numeric_only ir
  corr=dataset.corr()
```

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

```
#plt.subplots(figsize=(10,10))
sns.heatmap(corr,annot=True)

# INFERENCE
# to understand the degree of correlation between all the variables
# Most correlated variable to total number of crashes : alcohol, speeding, not_distracted, no_previous
# Least correlated variable to total number of crashes : ins_premium, ins_losses
```

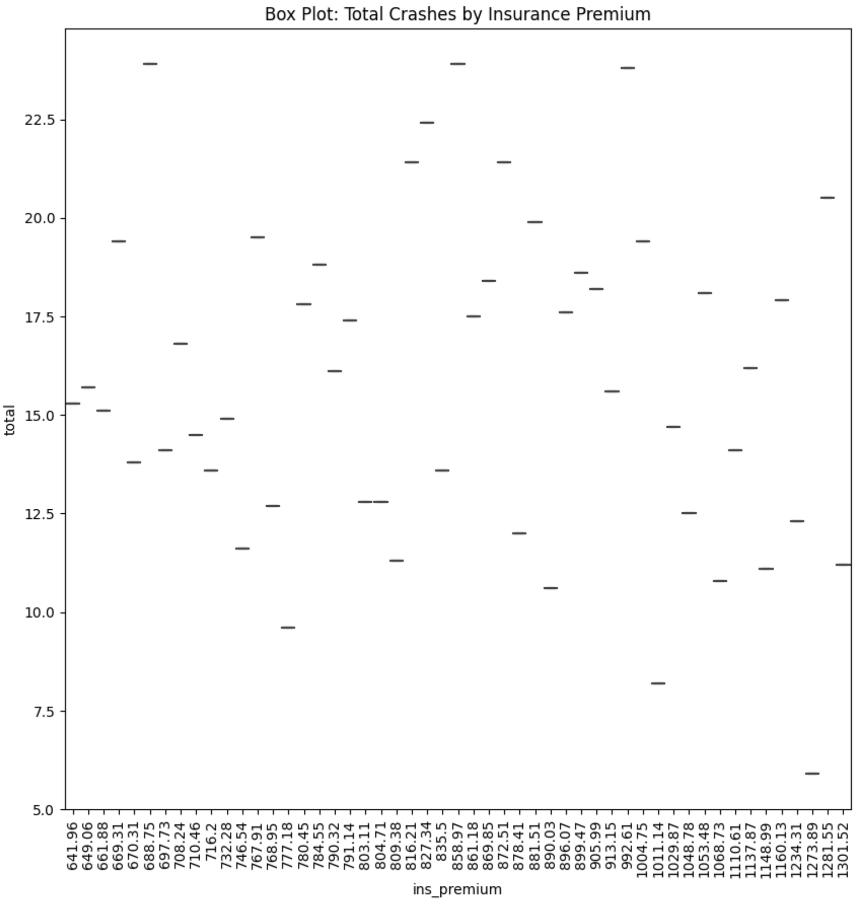
```
<Axes: >
```



.

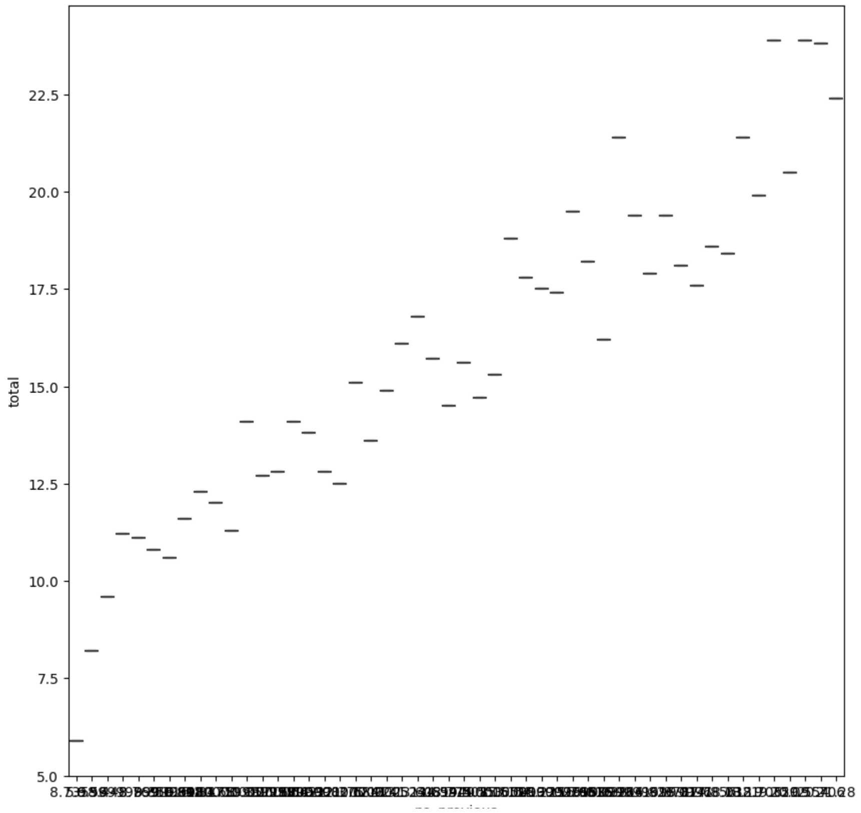
BOX-PLOT

```
# Box plot of "total" crashes by "ins_premium"
plt.subplots(figsize=(10,10))
sns.boxplot(data=dataset, x="ins_premium", y="total")
plt.title("Box Plot: Total Crashes by Insurance Premium")
plt.xticks(rotation=90)
plt.show()
```



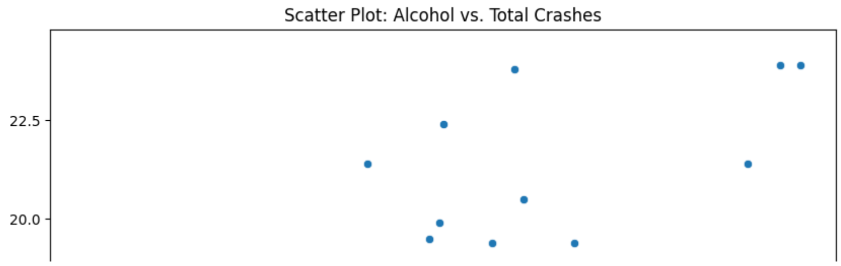
```
plt.subplots(figsize=(10,10))
sns.boxplot(data=dataset, x="no_previous", y="total")
```

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



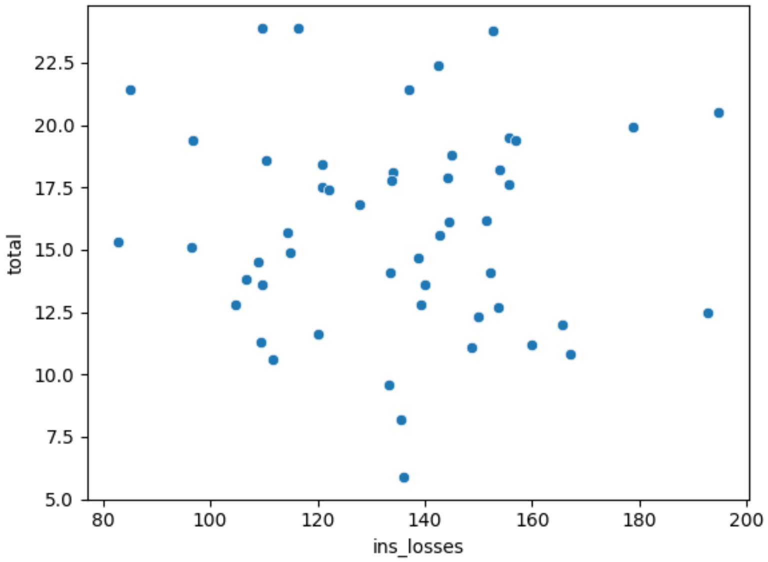
SCATTER PLOT

```
plt.subplots(figsize=(10, 10))
sns.scatterplot(data=dataset, x="alcohol", y="total")
plt.title("Scatter Plot: Alcohol vs. Total Crashes")
plt.show()

#INFERENCE
# To understand relation between "alcohol" and "total" with the help of a scatter plot
# As the amount of alcohol increases, the severity of car crashes also increases.
```



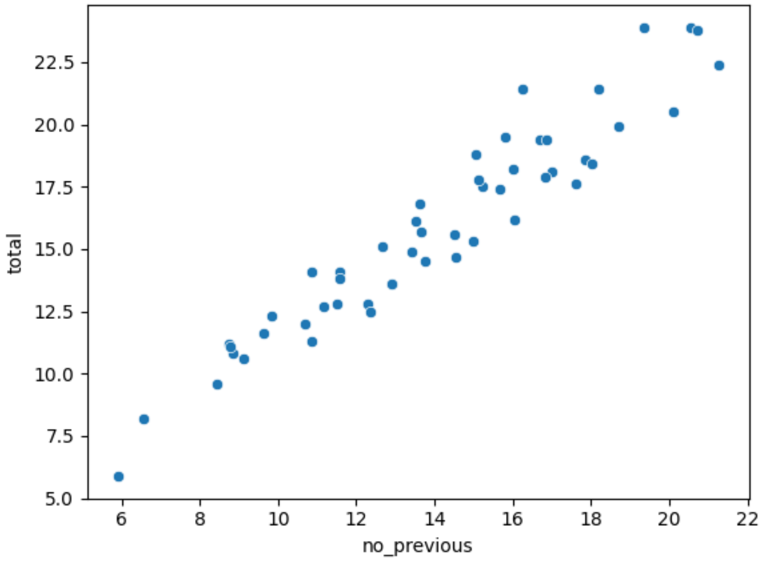
```
sns.scatterplot(data=dataset, x="ins_losses", y="total")

# INFERENCE
# No linear relation observed between total crashes and losses covered by insurance company.
```

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



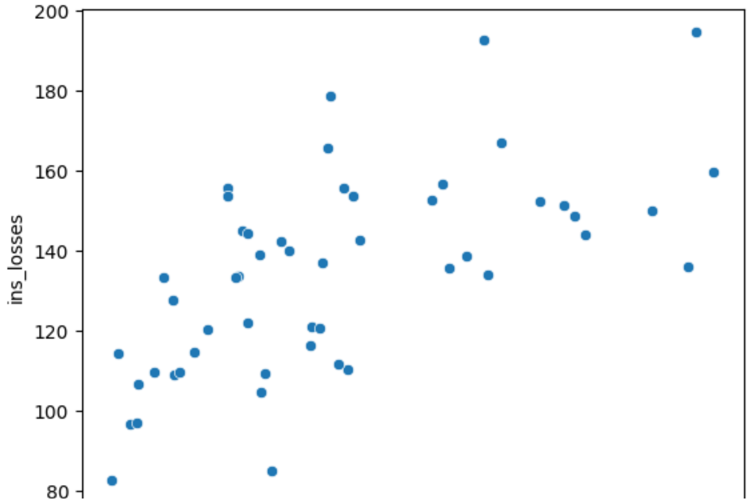
```
sns.scatterplot(data=dataset, x="no_previous", y="total")

# INFERENCE
# No linear relation observed between drivers with no previous record of car crash with total number of crashes.
# This implies that new car drivers are more prone to crashes.
```

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



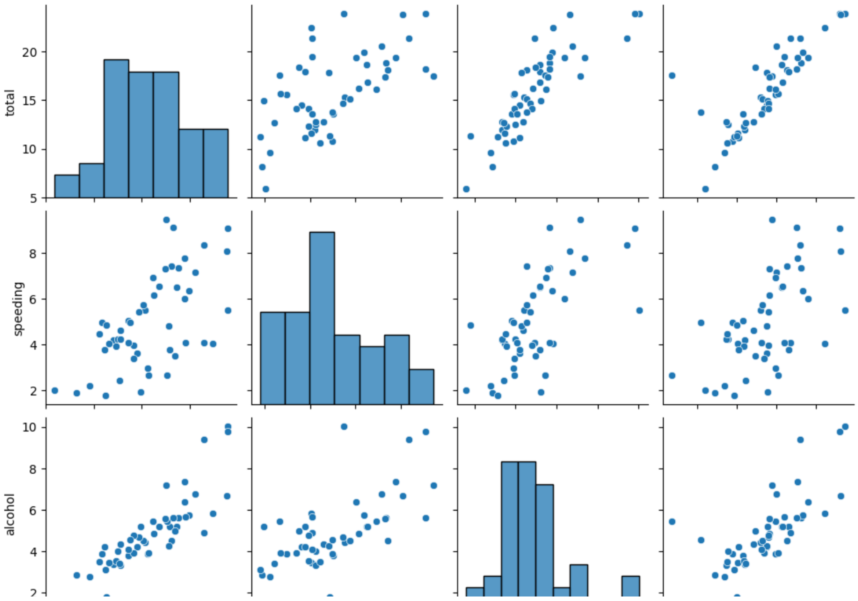
```
sns.scatterplot(data=dataset, x="ins_premium", y="ins_losses")

# INFERENCE
# Comparing the degree of losses covered by insurance company in comparision to premium paid by customer
# More is the premium paid by customer, more are the losses covered by insurance company.
```

```
<Axes: xlabel='ins_premium', ylabel='ins_losses'>
```



PAIR PLOT

```
# Pair plot for selected columns
sns.pairplot(dataset[['total', 'speeding', 'alcohol', 'not_distracted']])
plt.title("Pair Plot")
plt.show()
```


total
20
15
10
5
speeding
8
6
4
2
alcohol
10
8
6
4
2