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

Start coding or generate with AI.

Start coding or generate with AI.

```
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']
```

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

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300000	7.10	77.40	2006
1	Radial Velocity	1	874.774000	2.21	56.95	2008
2	Radial Velocity	1	763.000000	2.60	19.84	2011
3	Radial Velocity	1	326.030000	19.40	110.62	2007
4	Radial Velocity	1	516.220000	10.50	119.47	2009
...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006
1031	Transit	1	2.615864	NaN	148.00	2007
1032	Transit	1	3.191524	NaN	174.00	2007
1033	Transit	1	4.125083	NaN	293.00	2008
1034	Transit	1	4.187757	NaN	260.00	2008

1035 rows × 6 columns

Start coding or generate with AI.

Start coding or generate with AI.

```
col_num = df.select_dtypes(exclude='category')
col_num
```

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300000	7.10	77.40	2006
1	Radial Velocity	1	874.774000	2.21	56.95	2008
2	Radial Velocity	1	763.000000	2.60	19.84	2011
3	Radial Velocity	1	326.030000	19.40	110.62	2007
4	Radial Velocity	1	516.220000	10.50	119.47	2009
...	...	...	...	...	...	...
1030	Transit	1	3.941507	NaN	172.00	2006
1031	Transit	1	2.615864	NaN	148.00	2007
1032	Transit	1	3.191524	NaN	174.00	2007
1033	Transit	1	4.125083	NaN	293.00	2008
1034	Transit	1	4.187757	NaN	260.00	2008

1035 rows × 6 columns

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

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
df['species'].unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

```
num_col = df.select_dtypes(exclude = 'object')
num_col
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
num_col.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

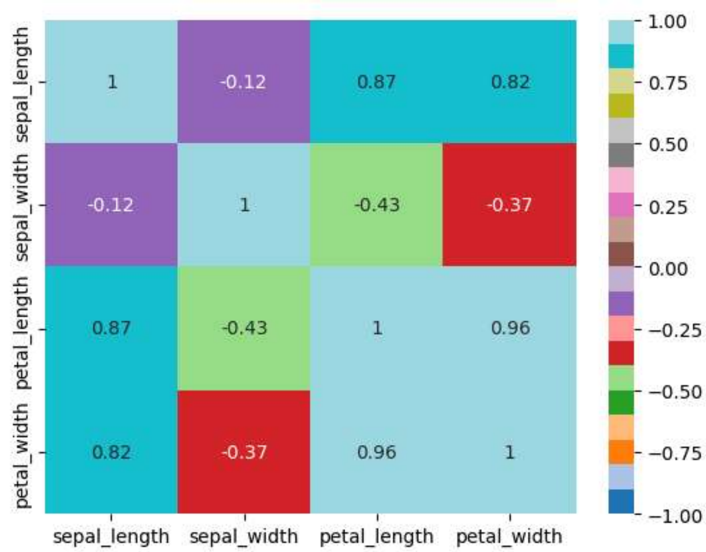
```
num_col.corr(method = 'pearson')
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

Start coding or generate with AI.

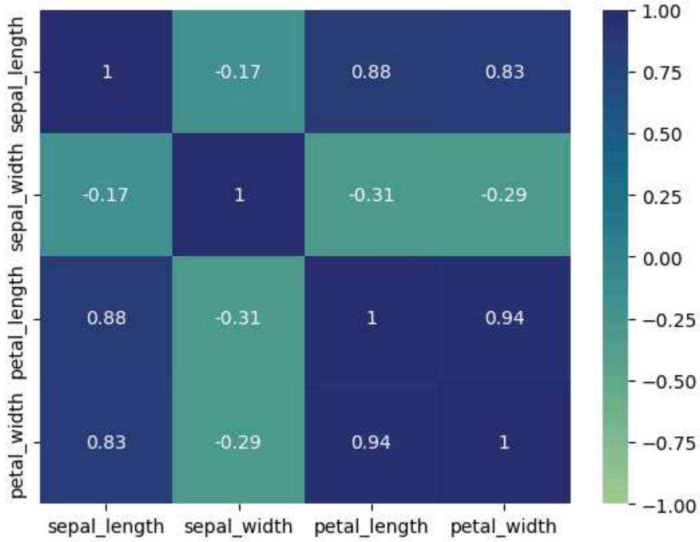
```
sns.heatmap(num_col.corr(method = 'pearson'),vmin = -1,vmax=1,annot = True,cmap = 'tab20')
```

<Axes: >



```
sns.heatmap(num_col.corr(method = 'spearman'),vmin=-1,vmax=1,annot = True,cmap='crest')
```

```
<Axes: >
```



Start coding or generate with AI.

Statistics are 3 types

```
1. univariate
2. bivariate
3. multivariate
```

```
#univariate - single column visualization
import seaborn as sns
df = sns.load_dataset('glue')
df
```

	Model	Year	Encoder	Task	Score
0	ERNIE	2019	Transformer	CoLA	75.5
1	T5	2019	Transformer	CoLA	71.6
2	RoBERTa	2019	Transformer	CoLA	67.8
3	BERT	2018	Transformer	CoLA	60.5
4	BiLSTM+ELMo	2018	LSTM	CoLA	32.1
...	...	...	...	...	...
59	BERT	2018	Transformer	RTE	70.1
60	BiLSTM+ELMo	2018	LSTM	RTE	57.4
61	BiLSTM+CoVe	2017	LSTM	RTE	52.7
62	BiLSTM+Attn	2017	LSTM	RTE	58.4
63	BiLSTM	2017	LSTM	RTE	57.4

64 rows × 5 columns

```
df['Task'].unique()
```

```
array(['CoLA', 'SST-2', 'MRPC', 'STS-B', 'QQP', 'MNLI', 'QNLI', 'RTE'],
      dtype=object)
```

```
df.head(15)
```

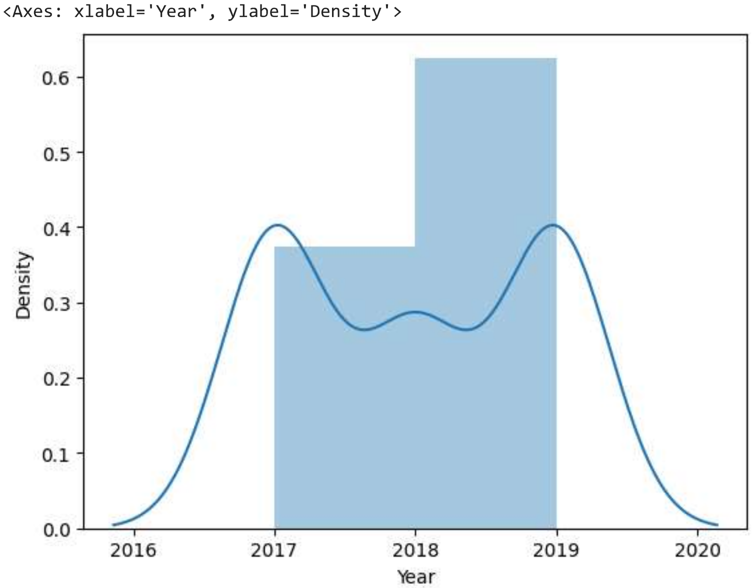
	Model	Year	Encoder	Task	Score
0	ERNIE	2019	Transformer	CoLA	75.5
1	T5	2019	Transformer	CoLA	71.6
2	RoBERTa	2019	Transformer	CoLA	67.8
3	BERT	2018	Transformer	CoLA	60.5
4	BiLSTM+ELMo	2018	LSTM	CoLA	32.1
5	BiLSTM+CoVe	2017	LSTM	CoLA	18.5
6	BiLSTM+Attn	2017	LSTM	CoLA	18.6
7	BiLSTM	2017	LSTM	CoLA	11.6
8	ERNIE	2019	Transformer	SST-2	97.8
9	T5	2019	Transformer	SST-2	97.5
10	RoBERTa	2019	Transformer	SST-2	96.7
11	BERT	2018	Transformer	SST-2	94.9
12	BiLSTM+ELMo	2018	LSTM	SST-2	89.3
13	BiLSTM+CoVe	2017	LSTM	SST-2	81.9
14	BiLSTM+Attn	2017	LSTM	SST-2	83.0

Start coding or generate with AI.

```
import warnings
warnings.filterwarnings('ignore')
```

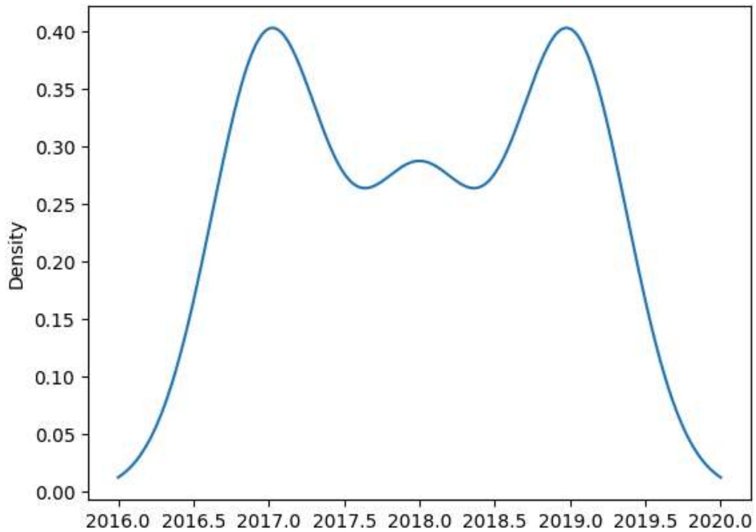
```
sns.distplot(df['Year'])
```

```
<Axes: xlabel='Year', ylabel='Density'>
```



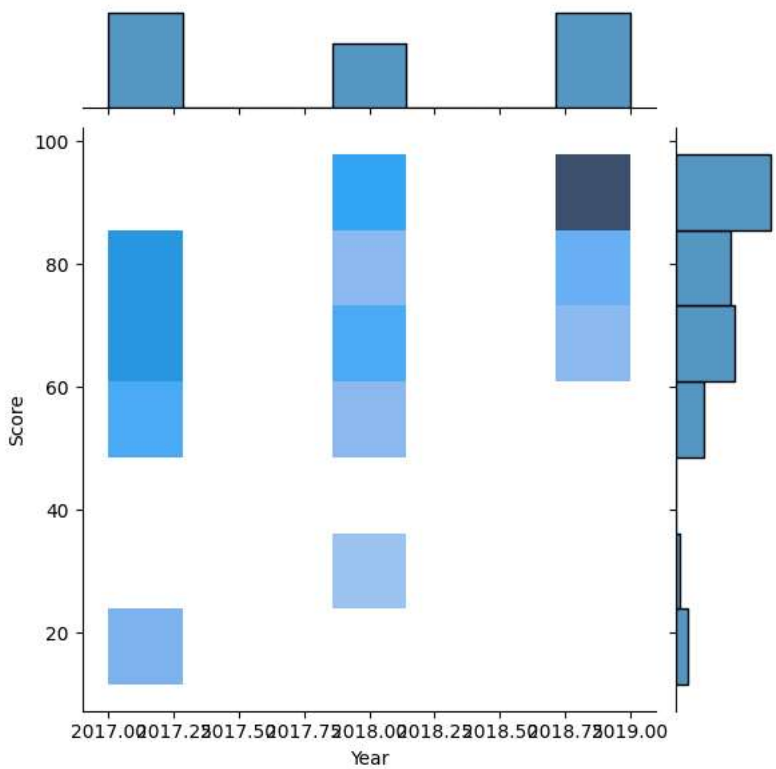

```
df['Year'].plot(kind = 'kde')
```

<Axes: ylabel='Density'>



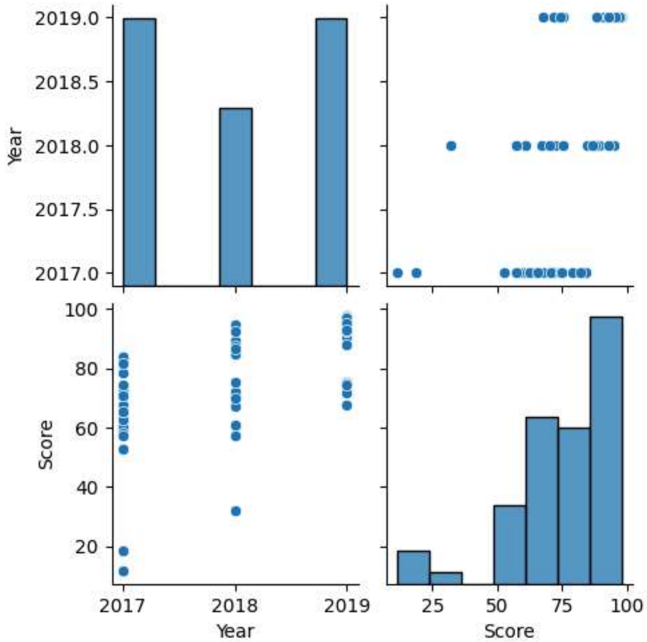

```
sns.jointplot(x=df['Year'],y=df['Score'],kind = 'hist')
```

<seaborn.axisgrid.JointGrid at 0x7b301ff76680>



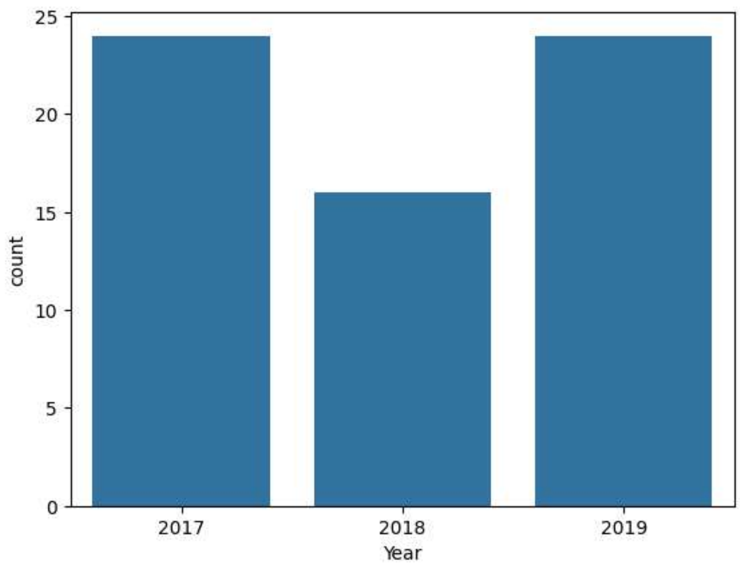

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7b301db3ee00>
```




```
sns.countplot(x=df['Year'])
```

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




```
import seaborn as sns
```

```
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']
```

```
sns.load_dataset('mpg')
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
0	18.0	8	307.0	130.0	3504	12.0	70	usa
1	15.0	8	350.0	165.0	3693	11.5	70	usa
2	18.0	8	318.0	150.0	3436	11.0	70	usa
3	16.0	8	304.0	150.0	3433	12.0	70	usa
4	17.0	8	302.0	140.0	3449	10.5	70	usa
...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	usa

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

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
0	18.0	8	307.0	130.0	3504	12.0	70	usa
1	15.0	8	350.0	165.0	3693	11.5	70	usa
2	18.0	8	318.0	150.0	3436	11.0	70	usa
3	16.0	8	304.0	150.0	3433	12.0	70	usa
4	17.0	8	302.0	140.0	3449	10.5	70	usa
...	...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82	usa

Start coding or generate with AI.

```
kk = df.select_dtypes(exclude = 'object')
kk
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
0	18.0	8	307.0	130.0	3504	12.0	70
1	15.0	8	350.0	165.0	3693	11.5	70
2	18.0	8	318.0	150.0	3436	11.0	70
3	16.0	8	304.0	150.0	3433	12.0	70
4	17.0	8	302.0	140.0	3449	10.5	70
...	...	...	...	...	...	...	...
393	27.0	4	140.0	86.0	2790	15.6	82
394	44.0	4	97.0	52.0	2130	24.6	82
395	32.0	4	135.0	84.0	2295	11.6	82
396	28.0	4	120.0	79.0	2625	18.6	82
397	31.0	4	119.0	82.0	2720	19.4	82

398 rows × 7 columns

```
kk.cov()
```

	mpg	cylinders	displacement	horsepower	weight	accele
mpg	61.089611	-10.308911	-655.402318	-233.857926	-5505.211745	9.
cylinders	-10.308911	2.893415	168.623214	55.348244	1290.695575	-2.
displacement	-655.402318	168.623214	10872.199152	3614.033744	82368.423240	-156.
horsepower	-233.857926	55.348244	3614.033744	1481.569393	28265.620231	-73.
weight	-5505.211745	1290.695575	82368.423240	28265.620231	717140.990526	-974.
acceleration	9.058930	-2.370842	-156.332976	-73.186967	-974.899011	7.
model_year	16.741163	-2.193499	-142.717137	-59.036432	-959.946344	2.

```
kk.corr()
```

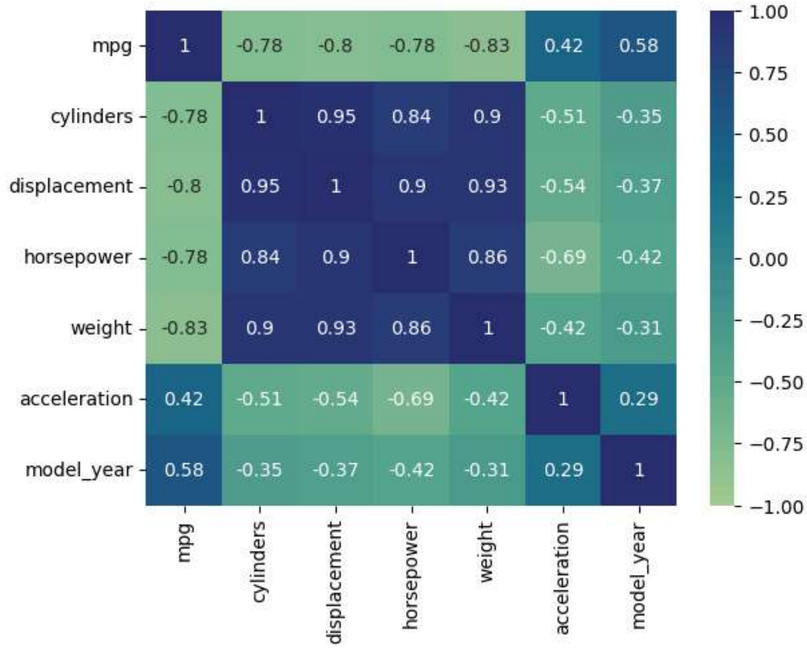
	mpg	cylinders	displacement	horsepower	weight	acceleration	mod
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	(
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-(
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-(
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-(
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-(
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	(
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1

```
kk.corr(method = 'pearson')
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	mod
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	(
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-(
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-(
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-(
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-(
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	(
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1

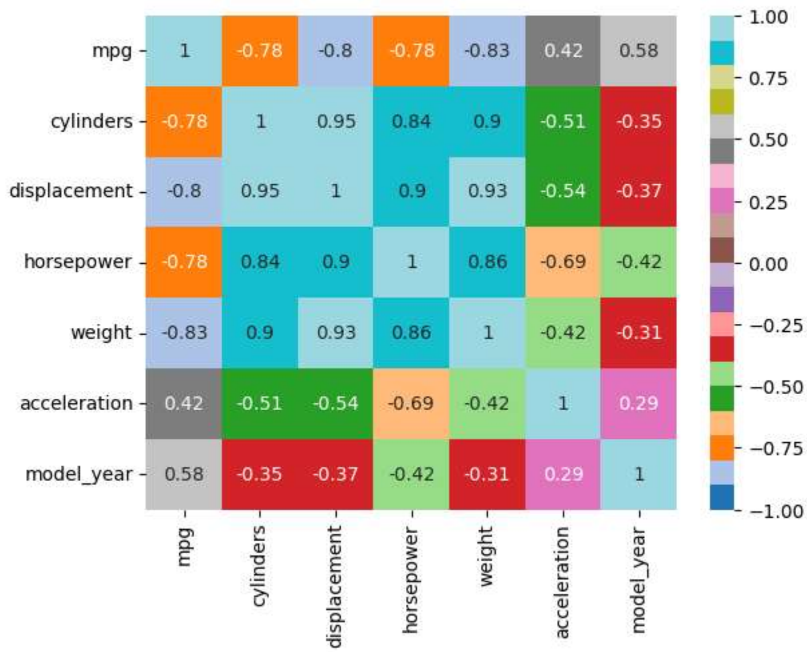
```
sns.heatmap(kk.corr(method = 'pearson'),vmin = -1,vmax = 1,annot = True,cmap='crest')
```

```
<Axes: >
```




```
sns.heatmap(kk.corr(method = 'pearson'),vmin = -1,vmax = 1,annot = True,cmap='tab20')
```

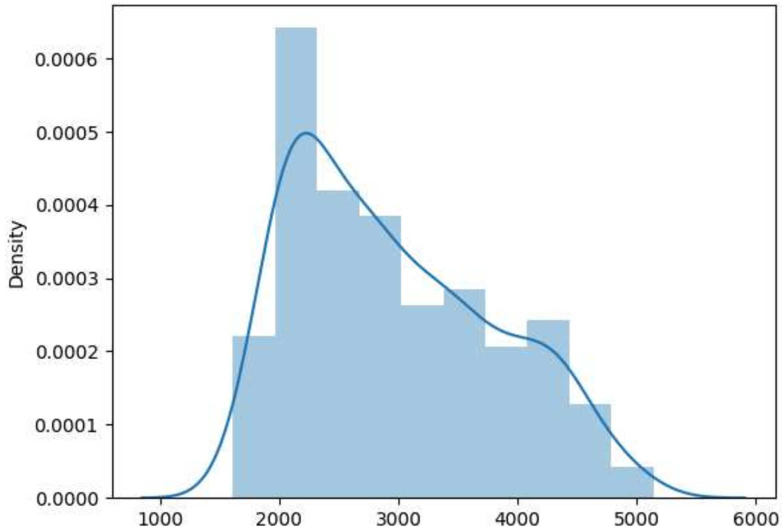
```
<Axes: >
```




```
#univariate analysis
```

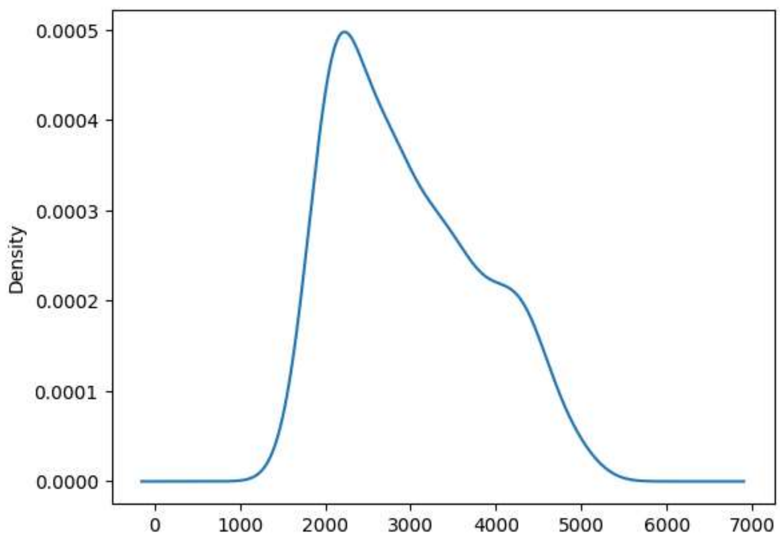
```
sns.distplot(x = df['weight'])
```

```
<ipython-input-17-93c64b9801d3>:4: 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(x = df['weight'])
<Axes: ylabel='Density'>
```



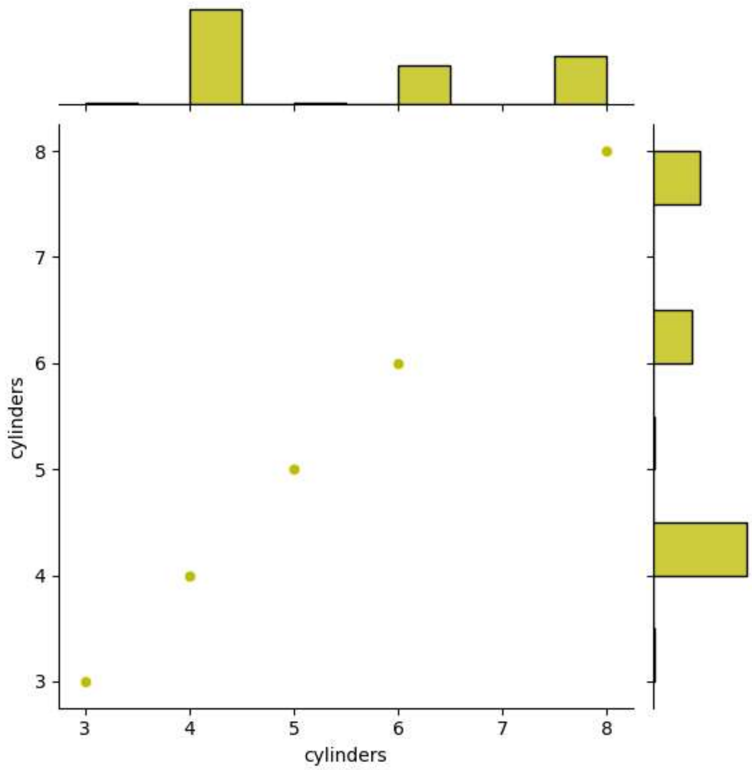

```
df['weight'].plot(kind = 'kde')
```

```
<Axes: ylabel='Density'>
```




```
sns.jointplot(x = kk['cylinders'], y = kk['cylinders'],color = 'y')
```

```
<seaborn.axisgrid.JointGrid at 0x7da434a0bc40>
```




```
sns.pairplot(kk)
```

```
<seaborn.axisgrid.PairGrid at 0x7da4301dfa60>
```