```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import plotly.graph_objects as go
import plotly.express as px


df=sns.load_dataset("diamonds")



df1=pd.DataFrame(df)



df1
```

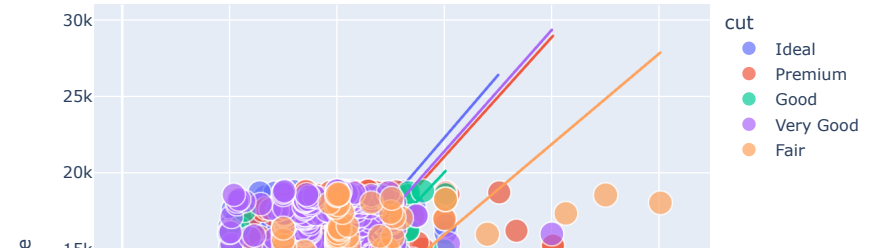
	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

53940 rows × 10 columns

```
df.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

```
#analyze the relationship between the carat and the price
figure=px.scatter(data_frame=df1,x="carat",y="price",size="depth",color="cut",trendline="ols")
figure.show()
```

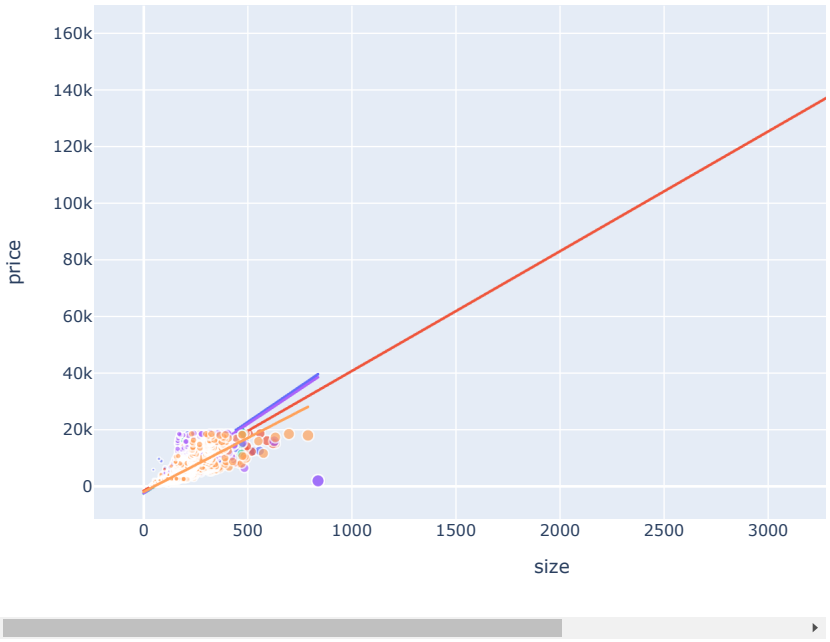



```
#create new column and add new data
df1['size']=df1['x']*df1['y']*df1['z']
df1
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917250
...	...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50	115.920000
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61	118.110175
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56	114.449728
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74	140.766120
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64	124.568444

53940 rows × 11 columns

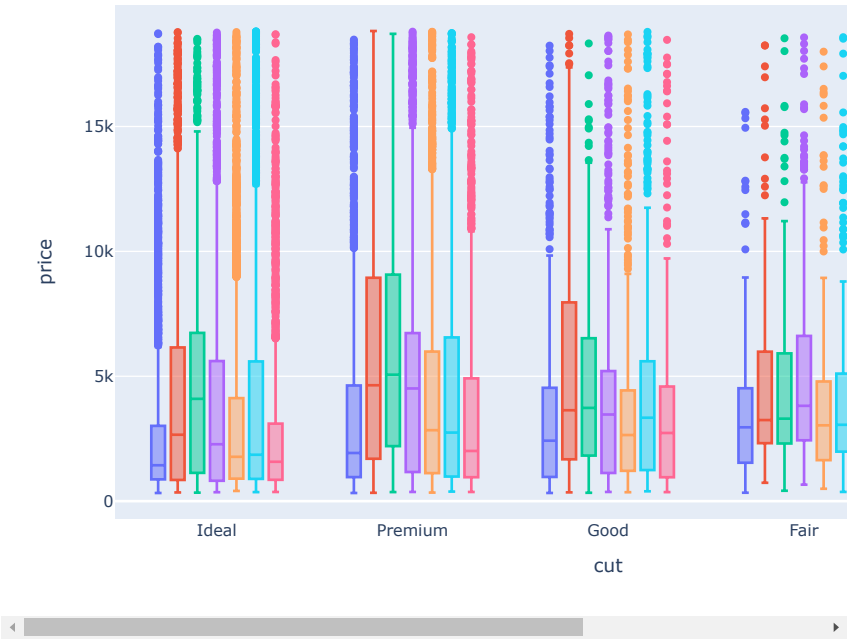
```
#We see the relatonship between the size of a diamond and its price

figure=px.scatter(data_frame=df1,x="size",y="price",size="size",color="cut",trendline="ols")
figure.show()
```



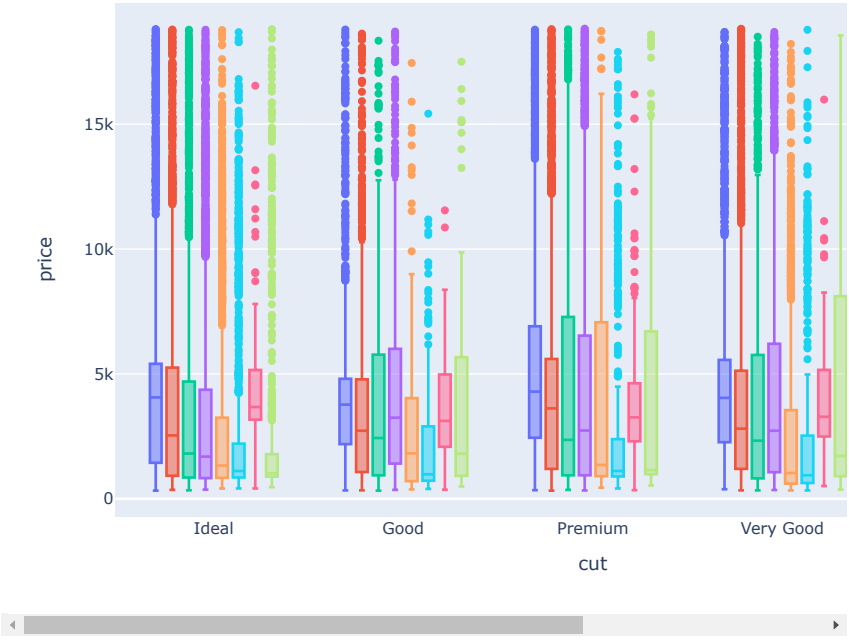

```
df.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917250

```
#prices of all types of diamonds based on their colour:
fig=px.box(df1,x="cut",y="price",color="color")
fig.show()
```




```
#prices of all types of diamonds based on their clarity
fig=px.box(df1,x="cut",y="price",color="clarity")
fig.show()
```



```
df1
```

	carat	cut	color	clarity	depth	table	price	x	y	z	
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917
...	...	...	...	...	...	...	...	...	...	...	
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50	115.920
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61	118.110
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56	114.449
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74	140.760
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64	124.568

```
df1.head(2)
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856

```
df1['price'].unique()
```

```
array([ 326,  327,  334, ..., 2753, 2755, 2756])
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 11 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    53940 non-null  float64
 1   cut      53940 non-null  category
 2   color    53940 non-null  category
 3   clarity  53940 non-null  category
 4   depth    53940 non-null  float64
 5   table    53940 non-null  float64
 6   price    53940 non-null  int64
 7   x        53940 non-null  float64
 8   y        53940 non-null  float64
 9   z        53940 non-null  float64
 10  size     53940 non-null  float64
dtypes: category(3), float64(7), int64(1)
memory usage: 3.4 MB
```

```
df1.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917250

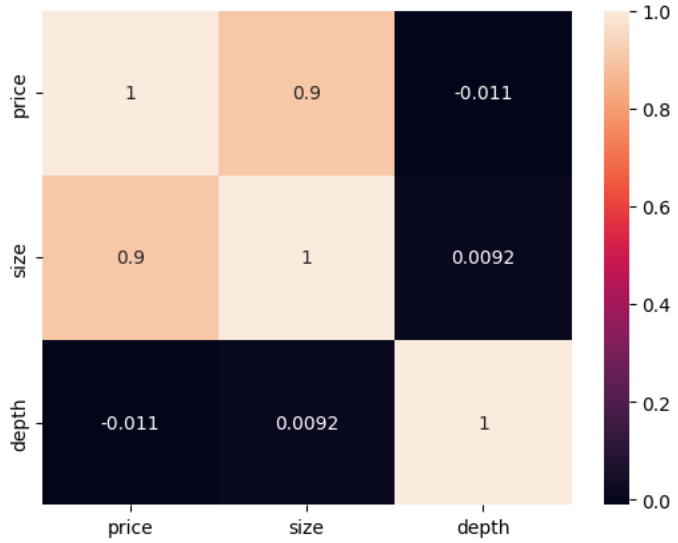
```
df1.head(2)
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030

```
corr=df1[['price','size','depth']].corr()
```

```
sns.heatmap(corr,annot=True)
```

```
<Axes: >
```




```
np.mean(df1['price'])
```

```
3932.799721913237
```

```
np.median(df1['price'])
```

```
2401.0
```

```
import statistics
```

```
statistics.mode(df1['price'])
```

```
605
```

```
#We see the relation between table and price
sns.scatterplot(x=df1['table'],y=df1['price'])
```

```
<Axes: xlabel='table', ylabel='price'>
```

```
# Maximum price of diamond
df1['price'].max()
```

```
18823
```

```
df1['table'].max()
```

```
95.0
```

```
np.std(df1['price'])
```

```
3989.4027576288736
```

```
df1.head(2)
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856

```
df1.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z	size
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75	51.917250