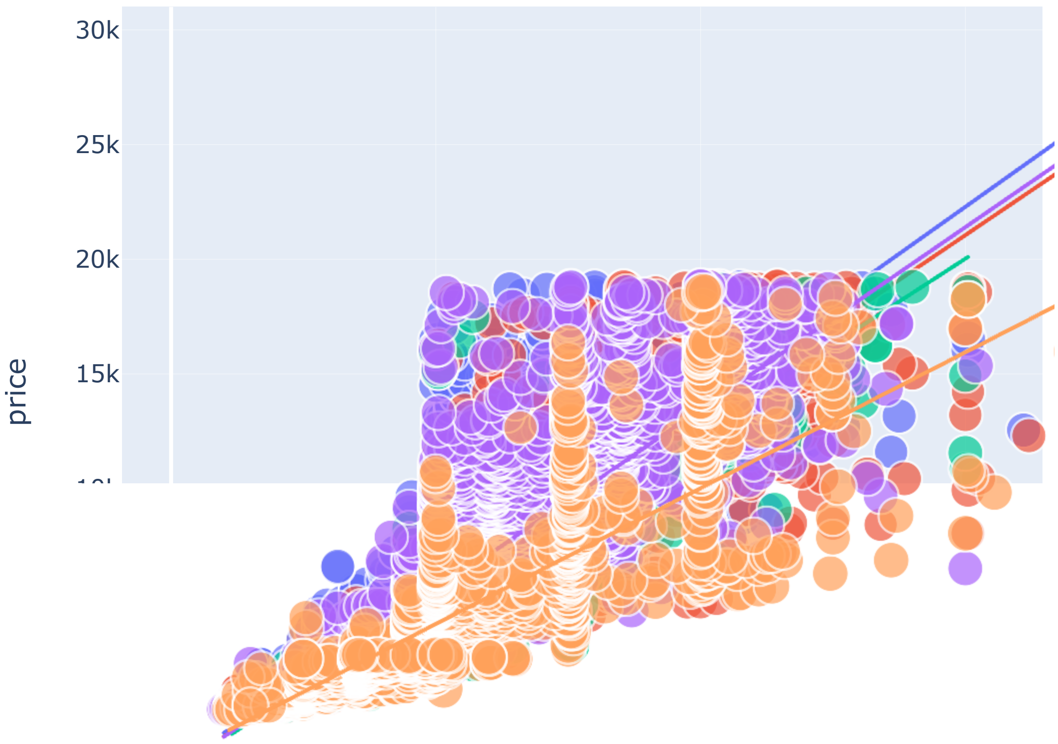
In [2]:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("E:\diamondspricepredection.csv")
print(data.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
```

In [3]:

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

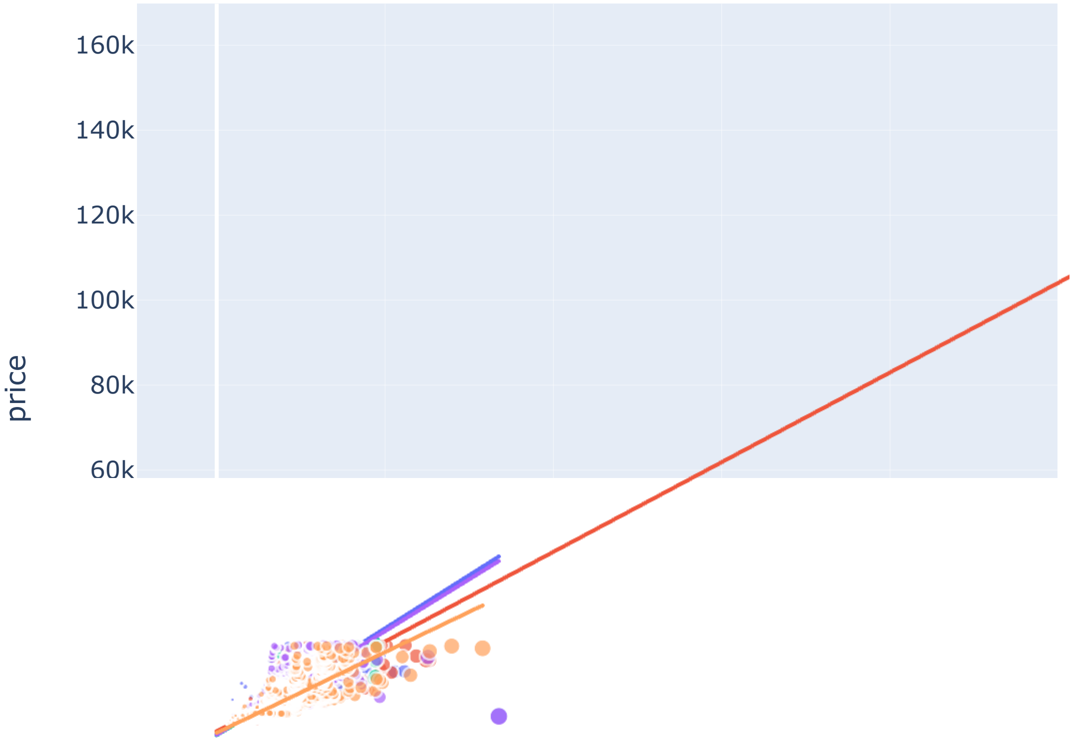


In [4]:

```
data["size"] = data["x"] * data["y"] * data["z"]
print(data)
```

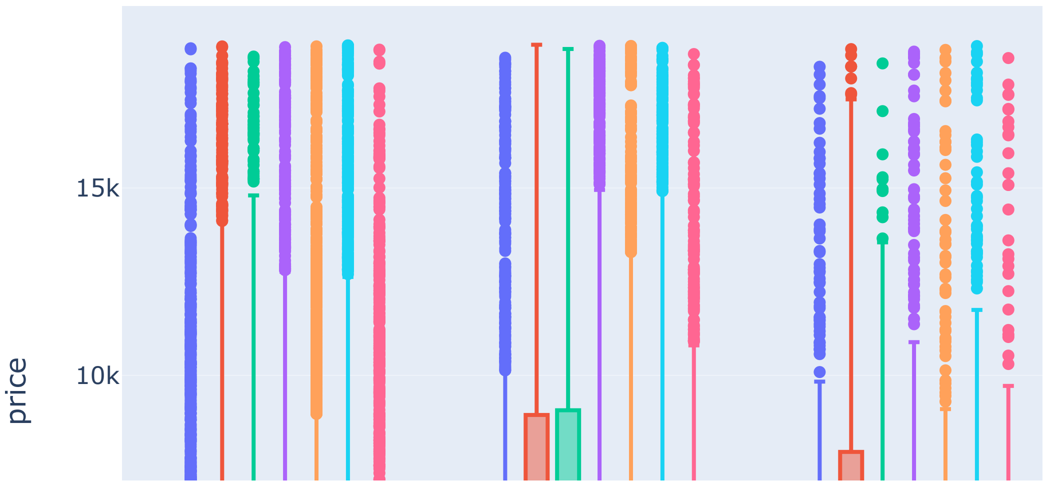
```
       carat        cut color clarity  depth  table  price     x     y  
z  \
0       0.23      Ideal     E     SI2   61.5   55.0    326  3.95  3.98  
2.43   
1       0.21    Premium     E     SI1   59.8   61.0    326  3.89  3.84  
2.31   
2       0.23       Good     E     VS1   56.9   65.0    327  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   

             size  
0       38.202030  
1       34.505856  
2       38.076885  
3       46.724580  
4       51.917250  
...           ...  
53935  115.920000  
53936  118.110175  
53937  114.449728  
53938  140.766120  
53939  124.568444  

[53940 rows x 11 columns]
```

```
In [5]:  figure = px.scatter(data_frame = data, x="size",
                             y="price", size="size",
                             color= "cut", trendline="ols")
         figure.show()
```



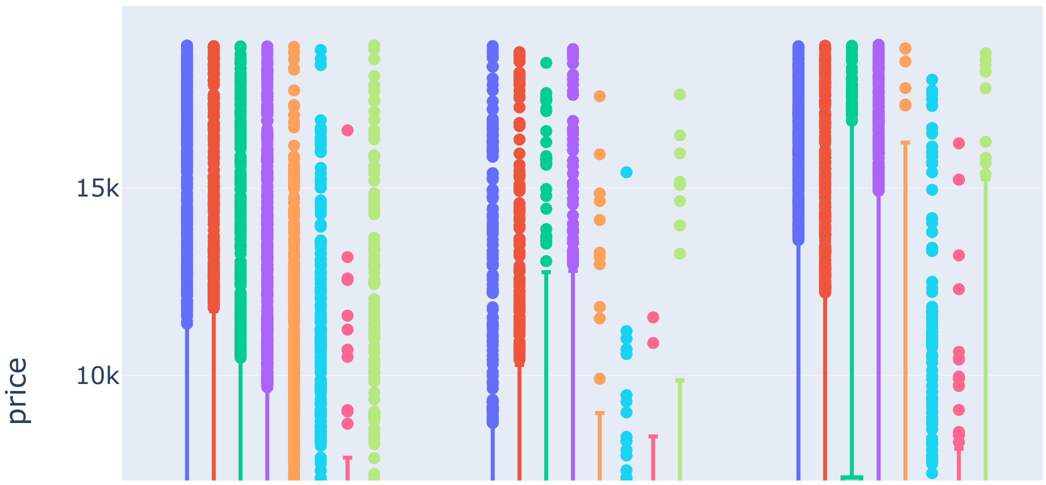
In [6]:

```
fig = px.box(data, x="cut",
             y="price",
             color="color")
fig.show()
```



In [7]:

```
fig = px.box(data,
             x="cut",
             y="price",
             color="clarity")
fig.show()
```



In [8]:

```
correlation = data.corr()
print(correlation["price"].sort_values(ascending=False))
```

```
price     1.000000
carat     0.921591
size      0.902385
x         0.884435
y         0.865421
z         0.861249
table     0.127134
depth    -0.010647
Name: price, dtype: float64
```

In [9]:

```
data["cut"] = data["cut"].map({"Ideal": 1,
                               "Premium": 2,
                               "Good": 3,
                               "Very Good": 4,
                               "Fair": 5})
```

In [10]:

```
#splitting data
from sklearn.model_selection import train_test_split
x = np.array(data[["carat", "cut", "size"]])
y = np.array(data[["price"]])

xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.10,
                                                random_state=42)
```

In [11]:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(xtrain, ytrain)
```

```
C:\Users\bhava\AppData\Local\Temp\ipykernel_9560\2944638855.py:3: Data
ConversionWarning:

A column-vector y was passed when a 1d array was expected. Please chan
ge the shape of y to (n_samples,), for example using ravel().
```

Out[11]:

```
RandomForestRegressor()
```

In [12]:

```
print("Diamond Price Prediction")
a = float(input("Carat Size: "))
b = int(input("Cut Type (Ideal: 1, Premium: 2, Good: 3, Very Good: 4, F
c = float(input("Size: "))
features = np.array([[a, b, c]])
print("Predicted Diamond's Price = ", model.predict(features))
```

```
Diamond Price Prediction
Carat Size: 3
Cut Type (Ideal: 1, Premium: 2, Good: 3, Very Good: 4, Fair: 5): 5
Size: 100
Predicted Diamond's Price =  [16824.0465]
```