

Diamonds_url.CSV file containing various attributes like carat, cut, color, clarity, price etc. for over 53,000 diamonds. Visualize the relationship between the carat (size of diamond) and price using a scatter plot. Instead of using the entire dataset for this visualization, just pick the diamonds with a clarity "SI2" and color "E". Use the values of the "cut" column to color the dots in the scatter plot. Make appropriate modifications to the chart title, axis titles, legend, figure size, font size, colors etc. To make the chart readable and visually appealing.

```
"""Diamonds_url.CSV file containing various attributes like carat, cut, color, clarity, price etc. for over 53,000 diamonds.
Visualize the relationship between the carat (size of diamond) and price using a scatter plot.
Instead of using the entire dataset for this visualization, just pick the diamonds with a clarity "SI2" and color "E".
Use the values of the "cut" column to color the dots in the scatter plot.
Make appropriate modifications to the chart title, axis titles, legend, figure size, font size, colors etc.
To make the chart readable and visually appealing."""

df=pd.read_csv(r'D:\itvedant\Vscode\diamonds.csv')
```

[4] ✓ 0.0s

```
df.head()
```

[5] ✓ 0.0s

```
...
```

	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

```
df.shape
```

[6] ✓ 0.0s

Ln 9, Col 49 CRLF Go Live

```
df.shape
✓ 0.0s
(53940, 10)

df.isnull().sum()
✓ 0.0s
carat    0
cut       0
color     0
clarity   0
depth     0
table     0
price     0
x         0
y         0
z         0
dtype: int64

df.duplicated().sum()
✓ 0.1s
146
```

```
data=df.drop_duplicates() #delete all the duplicates from df
✓ 0.0s

data.duplicated().sum()
✓ 0.0s
0

# Instead of using the entire dataset for this visualization, just pick the diamonds with a clarity "SI2" and color "E".
Df= df[(df['clarity'] == "SI2") & (df['color'] == 'E')]
Df
✓ 0.0s
```

	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
14	0.20	Premium	E	SI2	60.2	62.0	345	3.79	3.75	2.27
91	0.86	Fair	E	SI2	55.1	69.0	2757	6.45	6.33	3.52
111	0.74	Ideal	E	SI2	62.2	56.0	2761	5.80	5.84	3.62
298	1.01	Fair	E	SI2	67.4	60.0	2797	6.19	6.05	4.13
...
53785	0.89	Good	E	SI2	64.3	65.0	2728	6.00	5.95	3.84

Ln 1, Col 21 CRLF Go

```
[17] ✓ 0.0s Python
... array(['Ideal', 'Premium', 'Fair', 'Very Good', 'Good'], dtype=object)

cut_colors={'Ideal':'red',
            'Premium':'green',
            'Fair':'blue',
            'Very Good':'black',
            'Good':'silver'}
colors=[cut_colors[cut] for cut in Df['cut']]

plt.figure(figsize=(8,5))
plt.scatter(Df['carat'],Df['price'],c=colors)
plt.xlabel('carat',fontsize=10)
plt.ylabel('price',fontsize=10)
plt.title("carat vs price")
plt.show()

[27] ✓ 0.2s Python
```

