Creado por:

Isabel Maniega

In [1]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

In [2]:
```python
df = sns.load_dataset("diamonds")
df.head()
```

Out[2]:

	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

Crearemos un columna de "Volumen" con los valores x, y, z

In [3]:
```python
df["volume"] = df["x"] * df["y"] * df["z"]
df = df.drop(["x", "y", "z"], axis=1)
df = df.drop(df.index[df["volume"] == 0], axis=0)
```

In [5]:
```python
df.head()
```

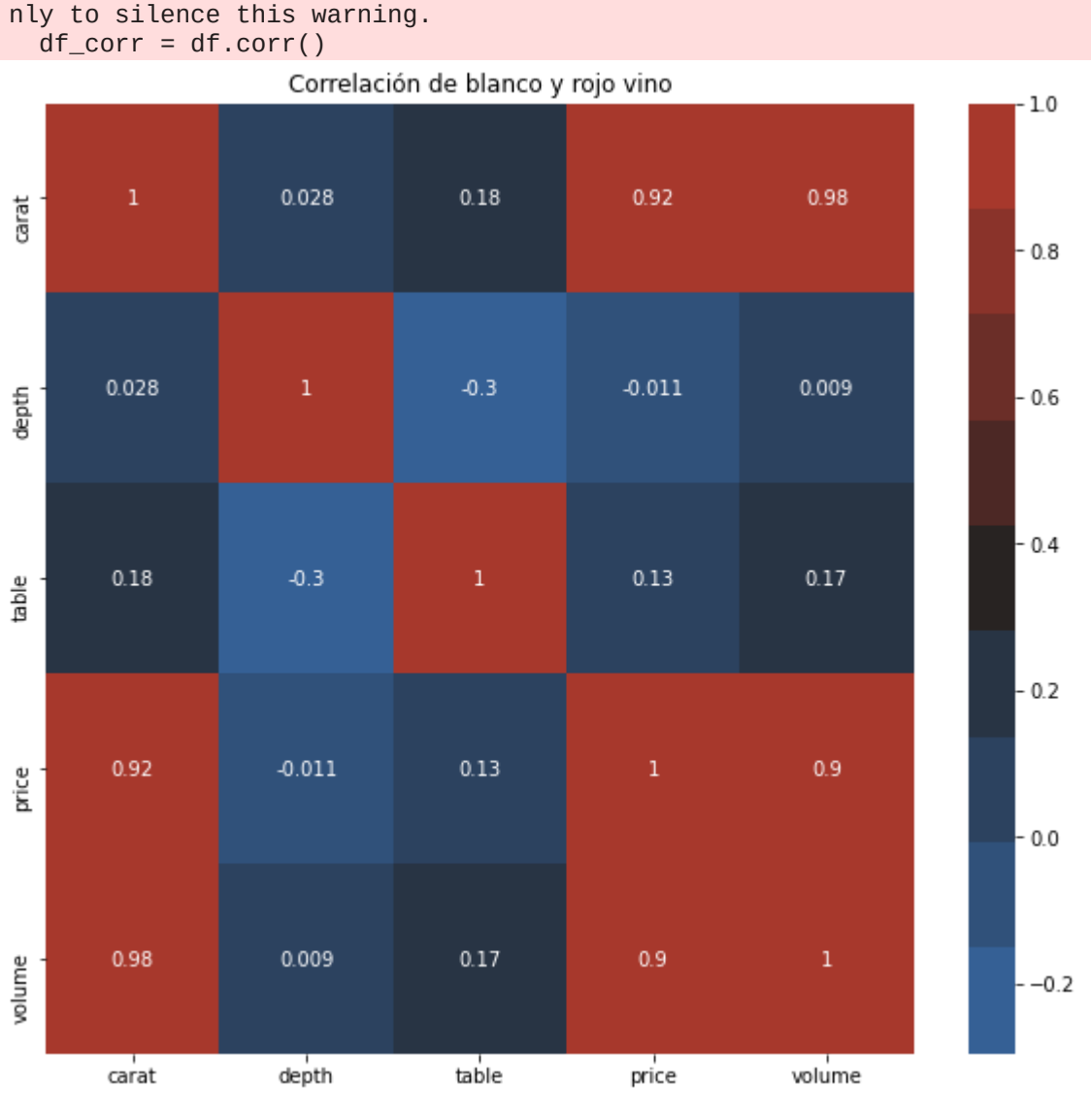
Out[5]:

	carat	cut	color	clarity	depth	table	price	volume
0	0.23	Ideal	E	SI2	61.5	55.0	326	38.202030
1	0.21	Premium	E	SI1	59.8	61.0	326	34.505856
2	0.23	Good	E	VS1	56.9	65.0	327	38.076885
3	0.29	Premium	I	VS2	62.4	58.0	334	46.724580
4	0.31	Good	J	SI2	63.3	58.0	335	51.917250

In [6]:
```python
plt.figure(figsize=(10, 9))
df_corr = df.corr()

sns.heatmap(df_corr, cmap=sns.diverging_palette(250, 15, s=75, l=40, n=9, center="dark"), annot=True)
plt.title("Correlación de blanco y rojo vino")
plt.show()
```

```
<ipython-input-6-6bc136bec0a6>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is depreca
ted. In a future version, it will default to False. Select only valid columns or specify the value of numeric_o
nly to silence this warning.
  df_corr = df.corr()
```



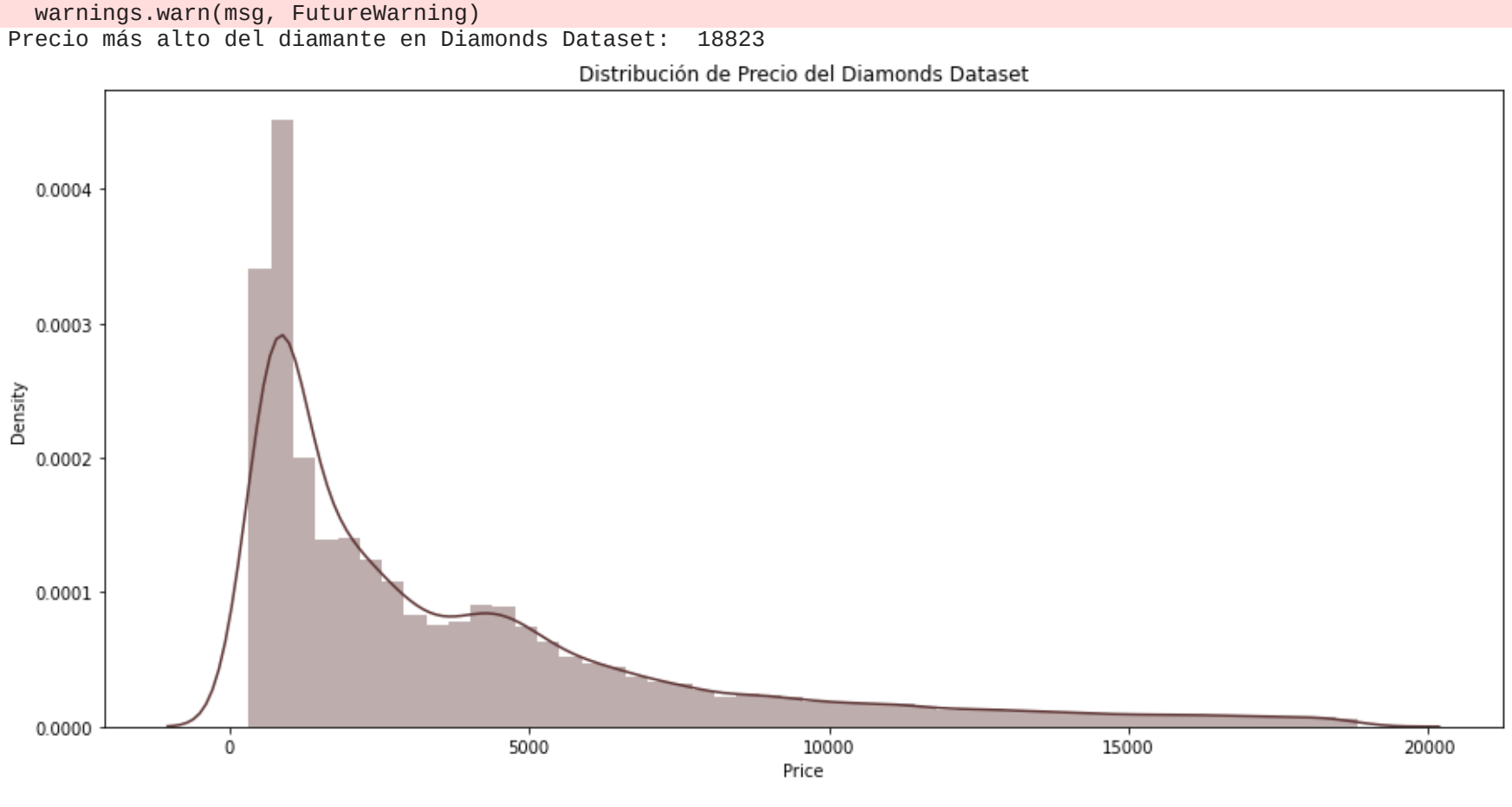
Conclusiones:

- Vemos que existe más correlación entre carat-volume, carat-price, price-volume

Distribución del Precio y Carat

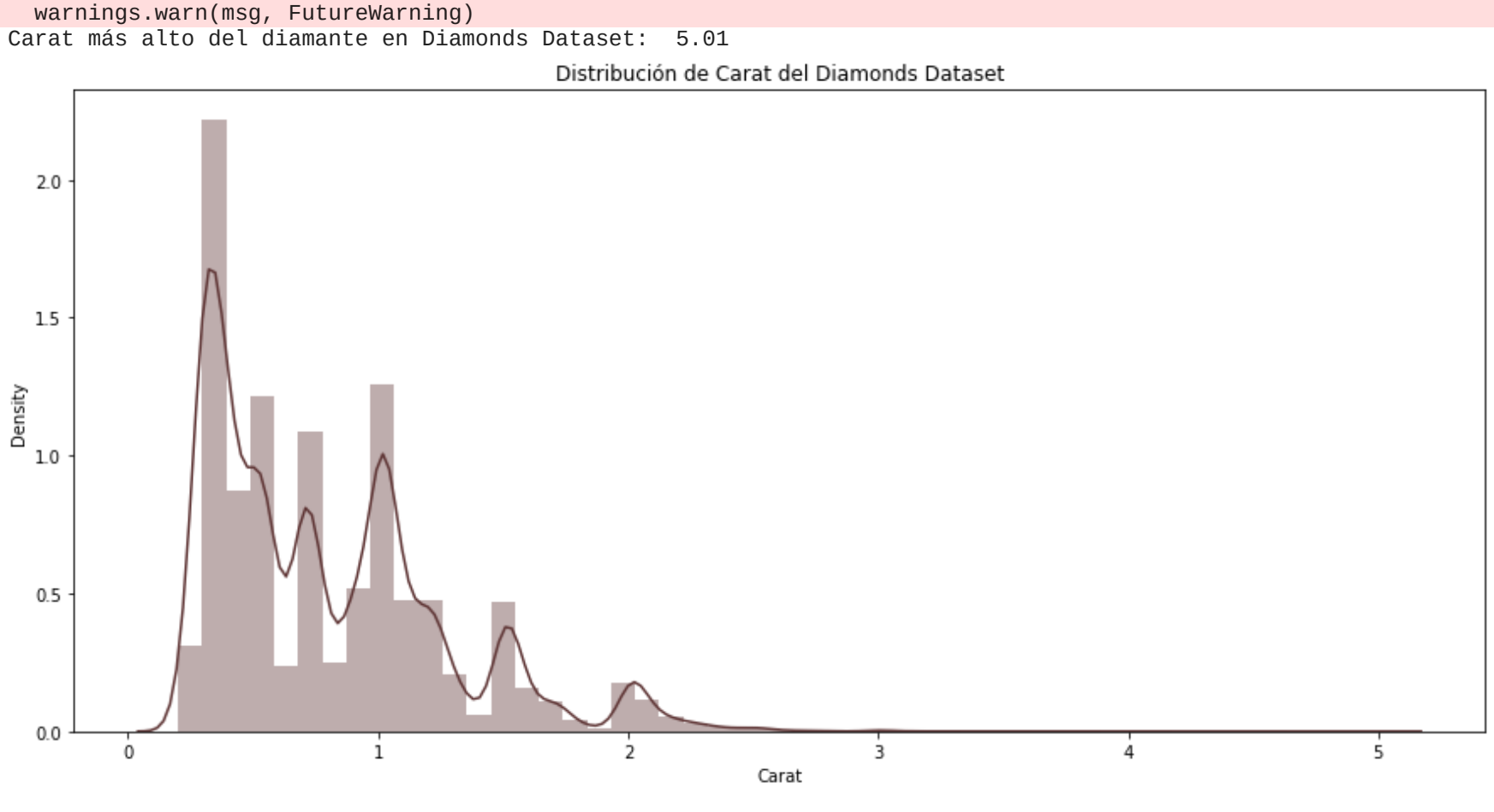
In [8]:
```python
plt.figure(figsize=(15, 7))
sns.distplot(df["price"], color="#5E3434")
plt.xlabel("Price")
plt.title("Distribución de Precio del Diamonds Dataset")
print("Precio más alto del diamante en Diamonds Dataset: ", df["price"].max())
plt.show()
```

```
/home/isabel/.local/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f
igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Precio más alto del diamante en Diamonds Dataset:  18823
```



In [9]:
```python
plt.figure(figsize=(15, 7))
sns.distplot(df["carat"], color="#5E3434")
plt.xlabel("Carat")
plt.title("Distribución de Carat del Diamonds Dataset")
print("Carat más alto del diamante en Diamonds Dataset: ", df["carat"].max())
plt.show()
```

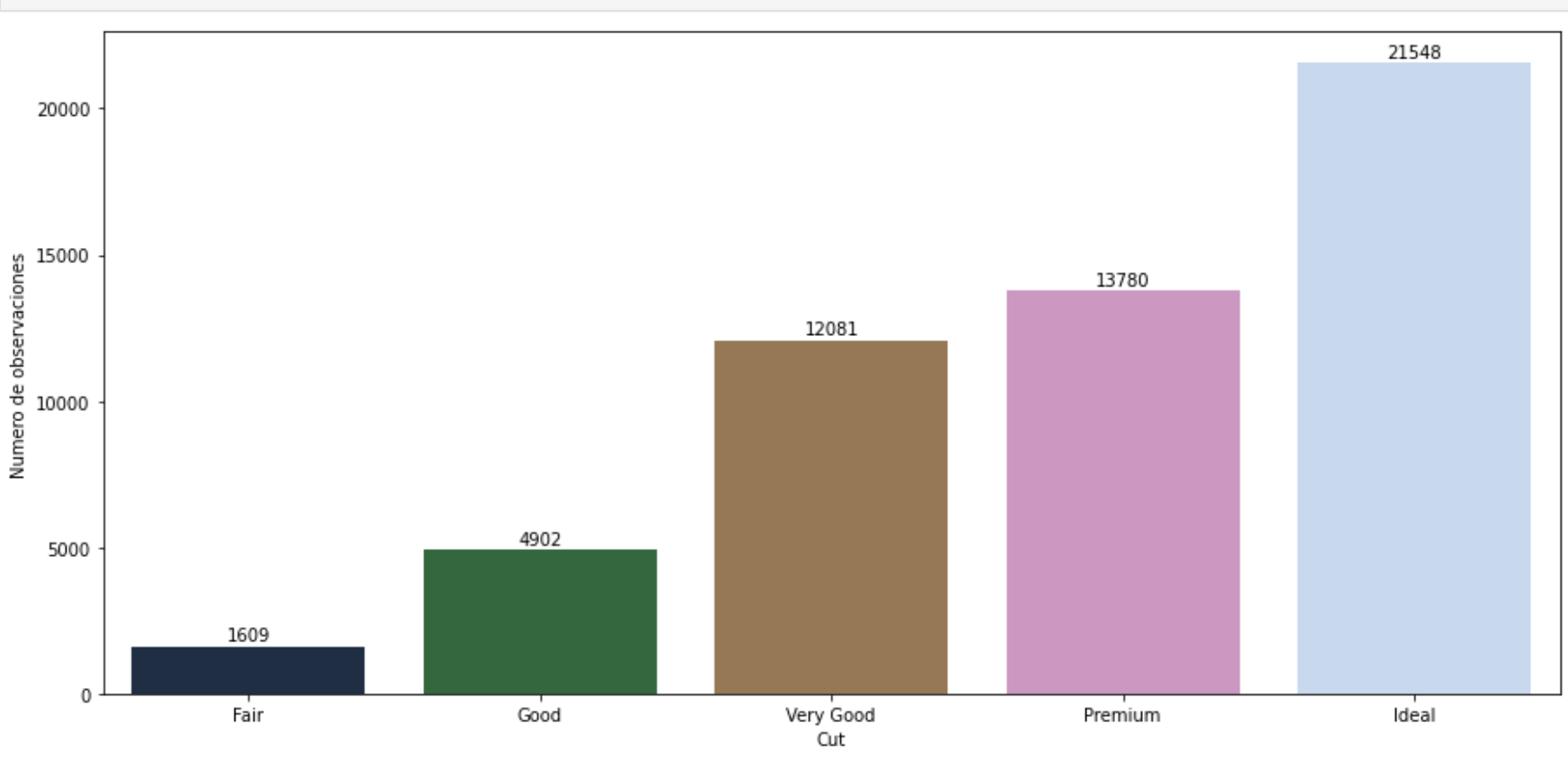
```
/home/isabel/.local/lib/python3.8/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a f
igure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
Carat más alto del diamante en Diamonds Dataset:  5.01
```



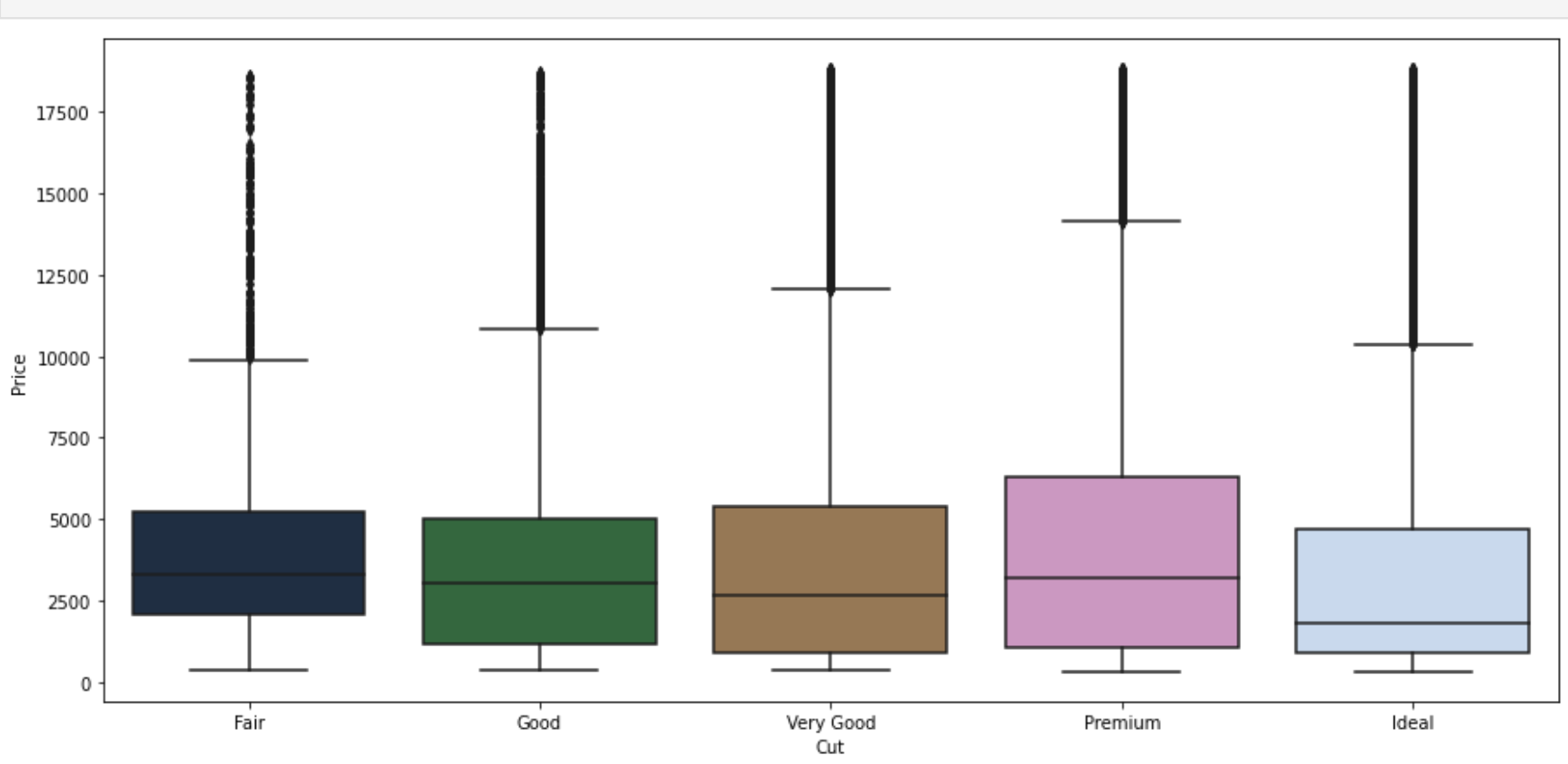
Cur Plots

In [13]:
```python
plt.figure(figsize=(15, 7))
cut= sns.countplot(x="cut", data=df, palette=sns.color_palette("cubehelix", 5),
                   order=["Fair", "Good", "Very Good", "Premium", "Ideal"])
plt.xlabel("Cut")
plt.ylabel("Numero de observaciones")

for p in cut.patches:
    heigth = p.get_height().round(2)
    text = str(heigth)
    cut.text(p.get_x() + p.get_width()/2, heigth + 200,text, ha="center")
plt.show()
```



In [14]:
```python
plt.figure(figsize=(15, 7))
sns.boxplot(x="cut", y="price", data=df, palette=sns.color_palette("cubehelix", 5),
            order=["Fair", "Good", "Very Good", "Premium", "Ideal"])
plt.xlabel("Cut")
plt.ylabel("Price")
plt.show()
```



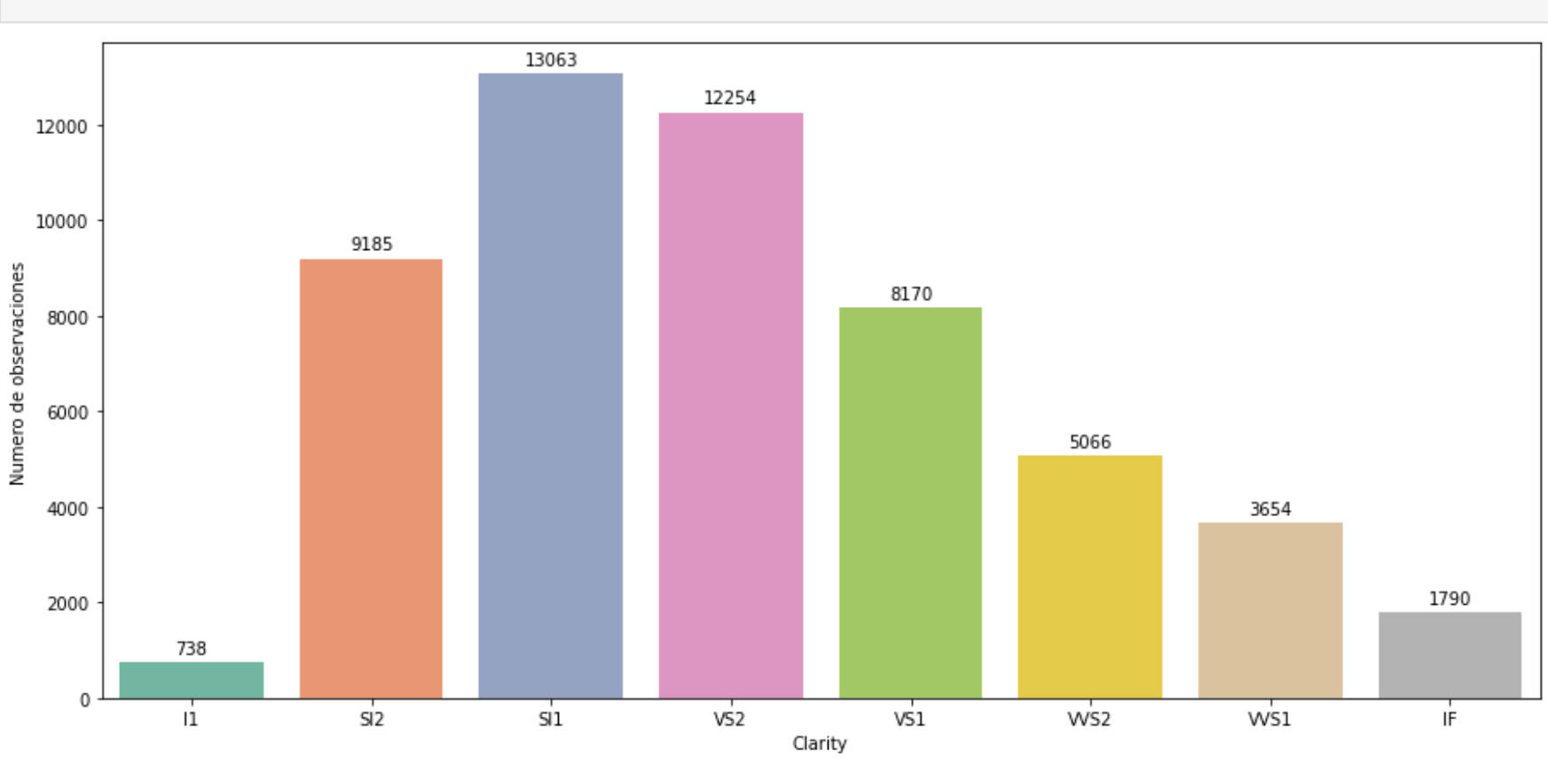
Conclusiones:

- presentan todos outliers en valores máximos
- Numero de observaciones es mayor en "Ideal"

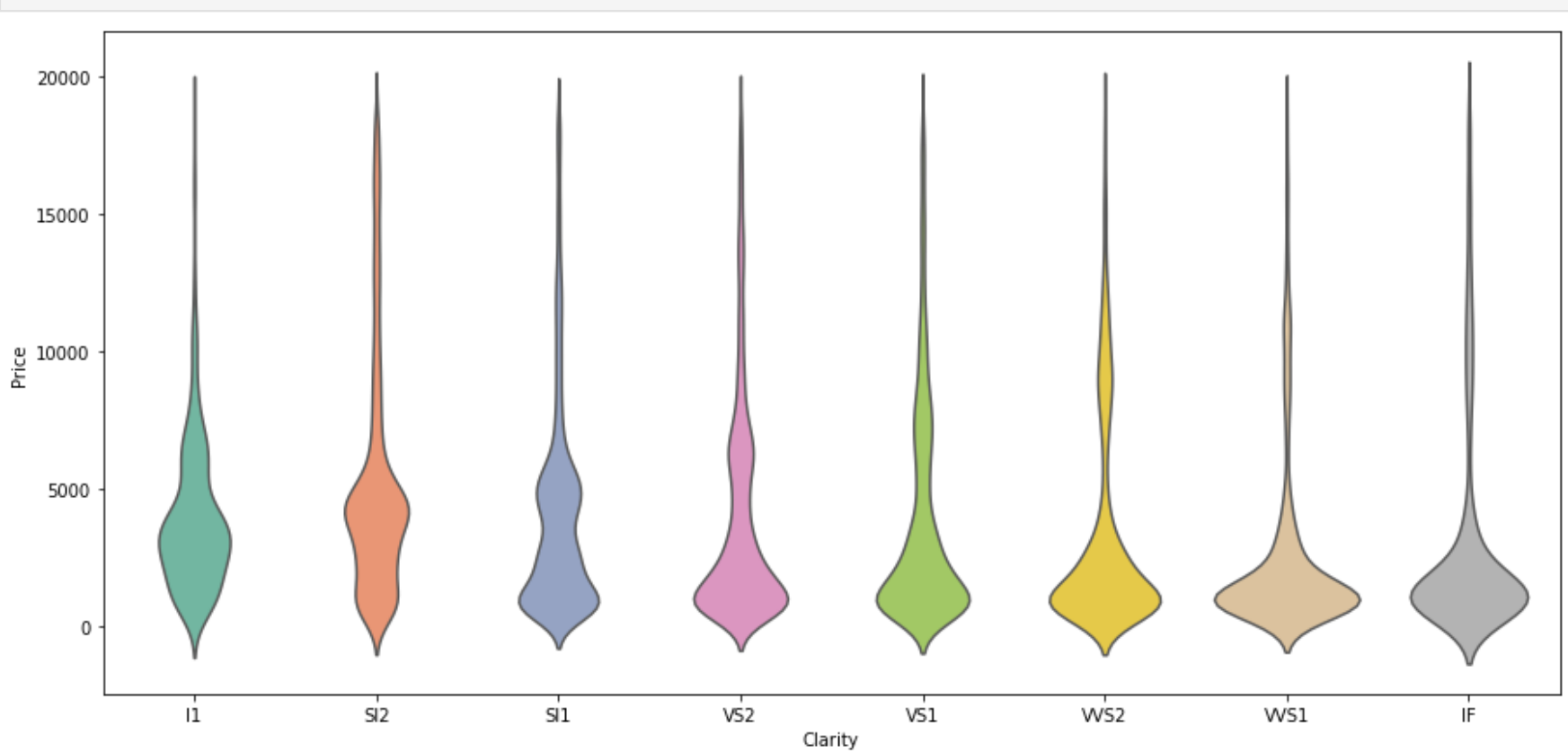
Clarity plots

In [15]:
```python
plt.figure(figsize=(15, 7))
clarity= sns.countplot(x="clarity", data=df, palette="Set2",
                       order=['I1', 'SI2', 'SI1', 'VS2', 'VS1', 'VVS2', 'VVS1', 'IF'])
plt.xlabel("Clarity")
plt.ylabel("Numero de observaciones")

for p in clarity.patches:
    heigth = p.get_height().round(2)
    text = str(heigth)
    clarity.text(p.get_x() + p.get_width()/2, heigth + 200,text, ha="center")
plt.show()
```



In [16]:
```python
plt.figure(figsize=(15, 7))
sns.violinplot(x="clarity", y="price", data=df, inner=None, palette="Set2",
               order=['I1', 'SI2', 'SI1', 'VS2', 'VS1', 'VVS2', 'VVS1', 'IF'])
plt.xlabel("Clarity")
plt.ylabel("Price")
plt.show()
```

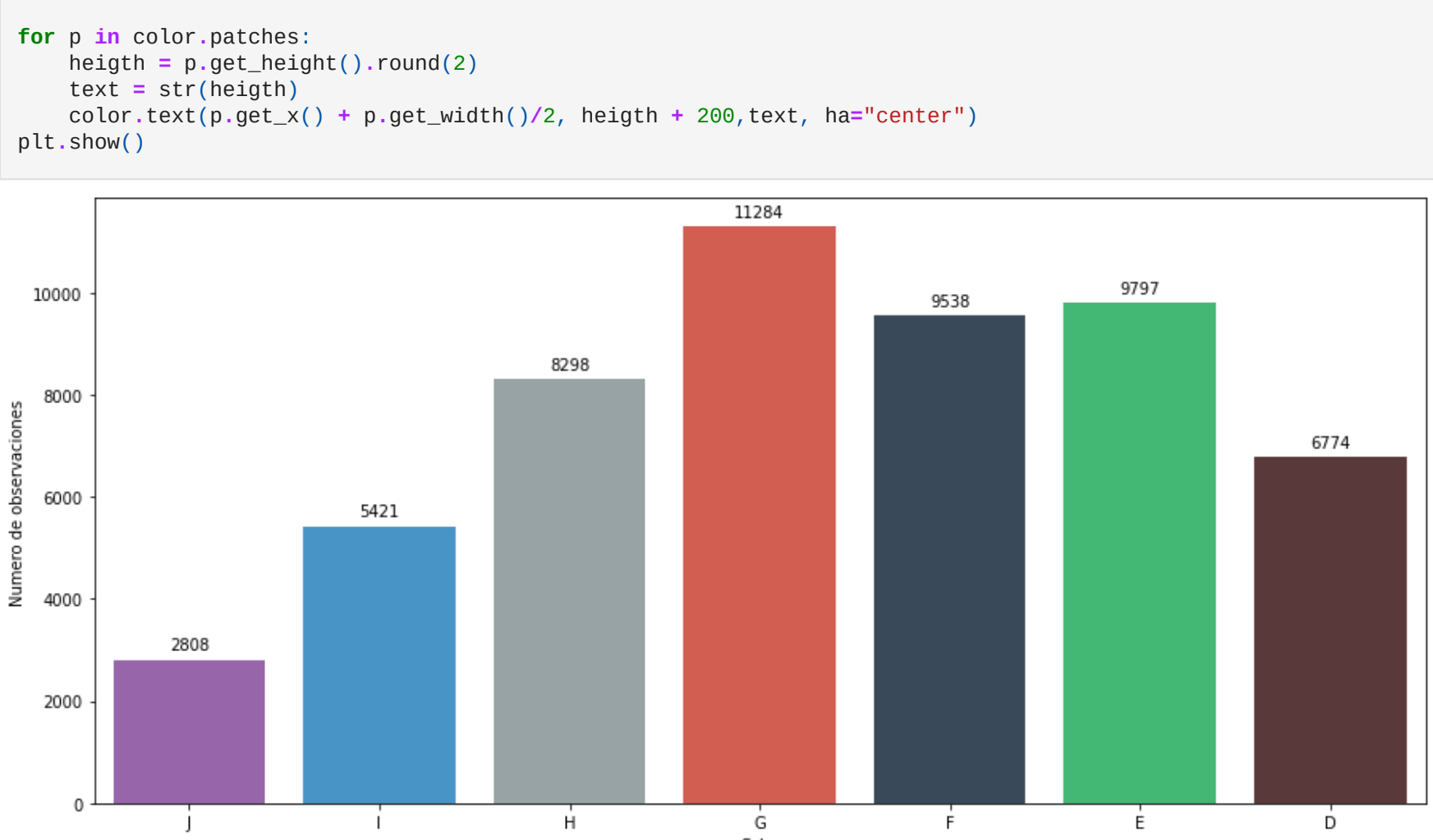


Color Plots

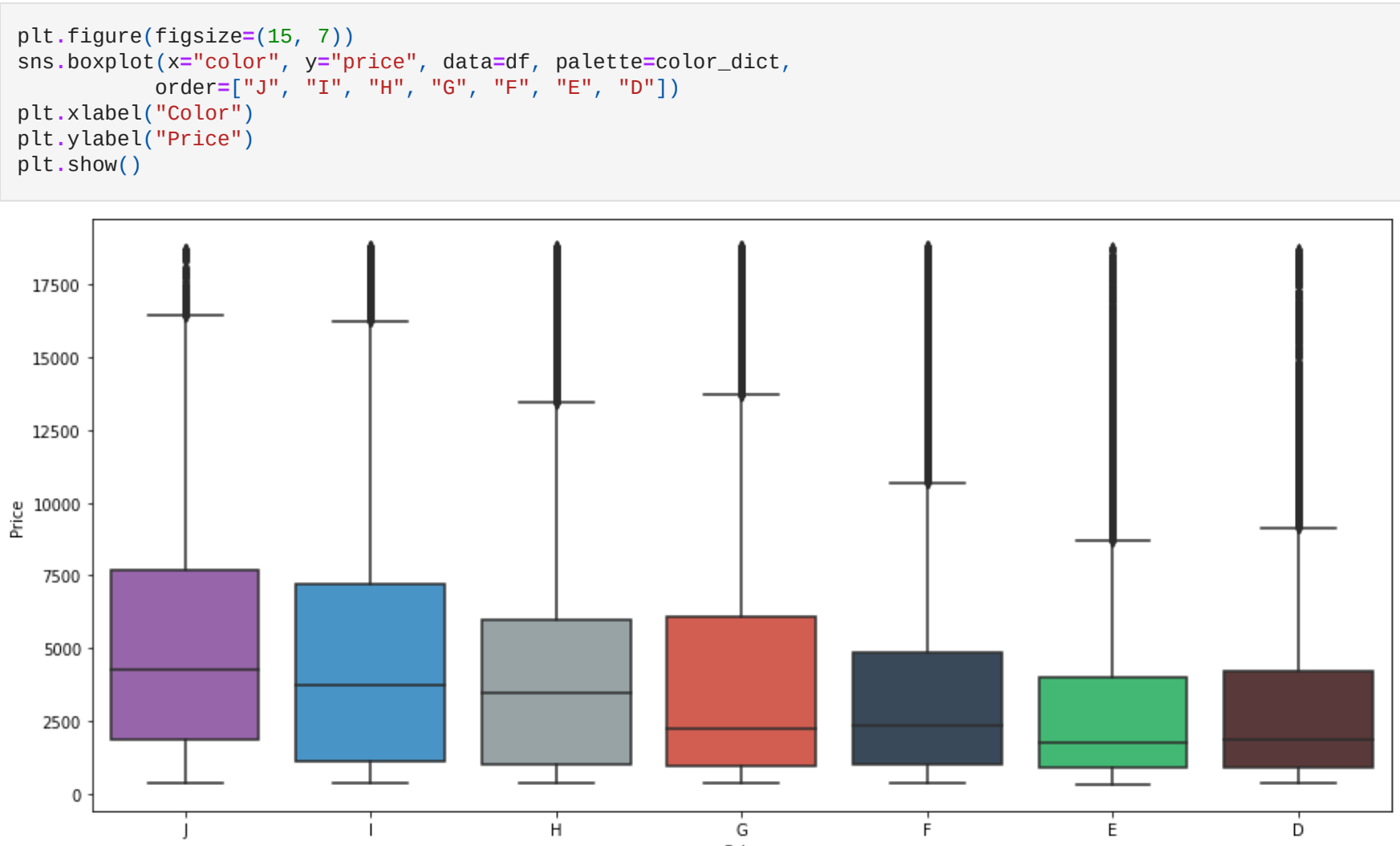
In [17]:
```python
color_dict = ["#9b59b6", "#3498db", "#95a5a6", "#e74c3c", "#34495e", "#2ecc71", "#5E3434"]
```

In [20]:
```python
plt.figure(figsize=(15, 7))
color= sns.countplot(x="color", data=df, palette=color_dict,
                     order=["J", "I", "H", "G", "F", "E", "D"])
plt.xlabel("Color")
plt.ylabel("Numero de observaciones")

for p in color.patches:
    heigth = p.get_height().round(2)
    text = str(heigth)
    color.text(p.get_x() + p.get_width()/2, heigth + 200,text, ha="center")
plt.show()
```



In [21]:
```python
plt.figure(figsize=(15, 7))
sns.boxplot(x="color", y="price", data=df, palette=color_dict,
            order=["J", "I", "H", "G", "F", "E", "D"])
plt.xlabel("Color")
plt.ylabel("Price")
plt.show()
```



Scatter plot de la Table & Depth de los diferentes cortes

In [22]:
```python
plt.figure(figsize=(9, 9))
sns.scatterplot(x="depth", y="table", data=df, y_jitter=True, alpha=0.5, hue="cut")
plt.xlabel("Depth")
plt.ylabel("Table")
plt.legend(ncol=3)
plt.xlim(50, 75)
plt.ylim(48, 72)
print("correlación de Depth & Table en los diamantes: ", round(df["table"].corr(df["depth"]), 2))
plt.show()
```

```
correlación de Depth & Table en los diamantes:  -0.3
```

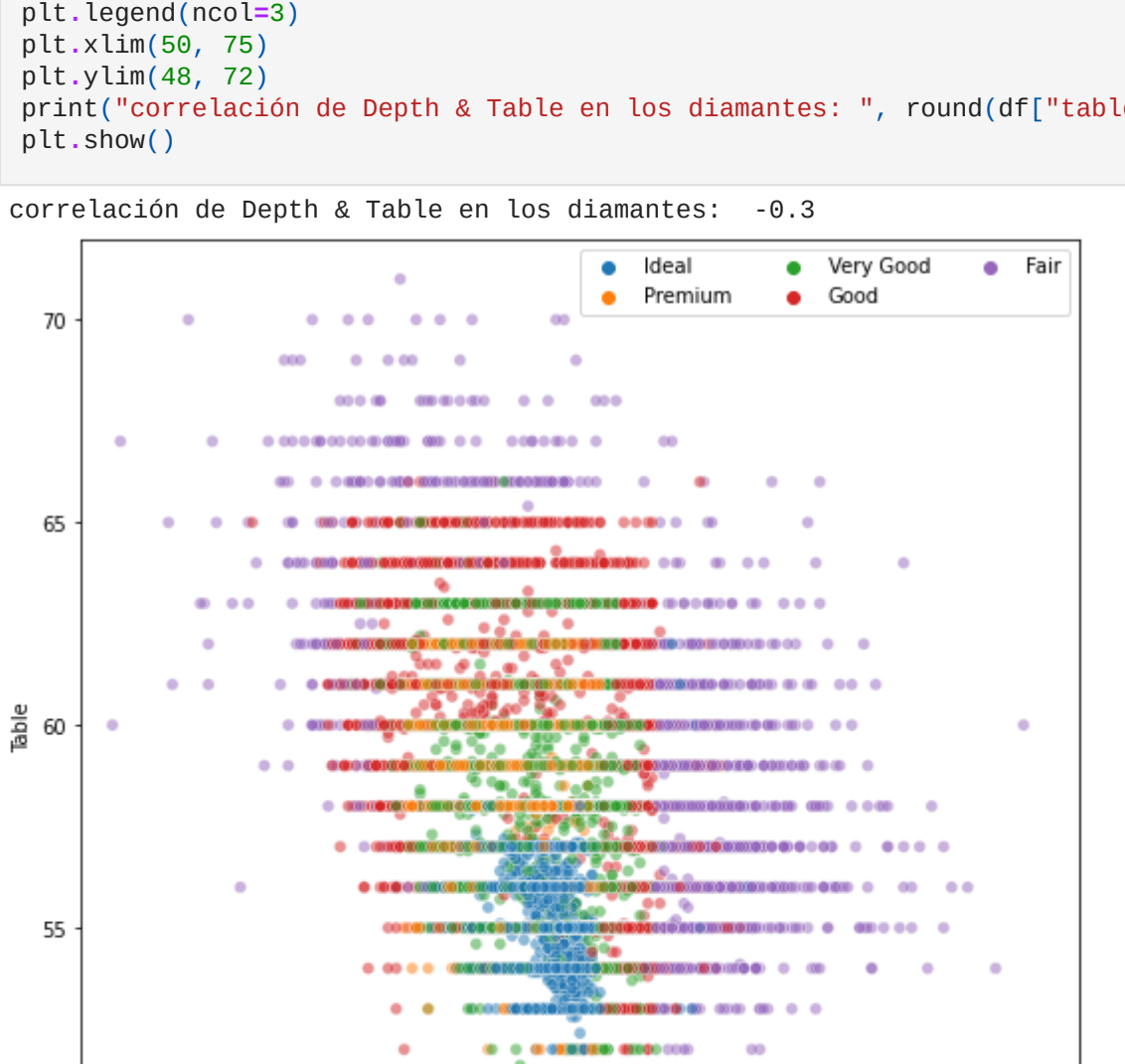


Gráfico 3D de "carat", "volume", & "price" con los diferentes cortes

In [23]:
```python
fig = px.scatter_3d(df, x="carat", y="volume", z="price", color="cut")
fig.show()
```

Creado por:

Isabel Maniega