

Data Visualization Resource Sheet

This sheet aims to help you with recalling important functions and methods for data visualization!

However, do remember that you can always look at the documentation of pandas, numpy, seaborn and matplotlib if you need extra help :)

Read csv and convert to dataframe

```
diamonds=pd.read_csv('/kaggle/input/diamonds/diamonds.csv')
```

This function will read the path of the csv file and convert it to a pandas dataframe.

Listing first few rows

```
diamonds.head()
```

Will list the first 5 rows as default, if a number is specified will print the first n rows.

Dimensions of a Dataframe

```
diamonds.shape
```

Will return the shape of the dataset in the format: (row,columns)

Grab a specific column

```
diamonds['price']
```

Outputs all the values in the column price.

Indexing

```
diamonds['cut'][0]
```

Returns the first value in the column cut. Note: this indexing is df[column][row]

iloc

```
diamonds.iloc[0]
```

Returns the first row.

```
diamonds.iloc[:, 1]
```

Returns all values in the second column (indexing starts from 0). Note: iloc is row first, then column.

loc

```
diamonds.loc[(diamonds.cut=='Premium') & (diamonds.price<400)]
```

Returns all rows where cut='Premium' **and** price is less than 400

Describing a column

```
diamonds.cut.describe()
```

Will give important information relating to this column, since this column is categorical will give count, unique, top and frequency. Otherwise will give mean, std_deviation etc.

Groupby

```
diamonds.groupby('carat').carat.count()
```

Will group up rows according to their value in column 'carat' then will sum them up.

Can apply other functions on groupby as well.

NumPy

```
a1 = [10, 20, 30]
```

Normal array

```
a3 = np.array([10, 20, 30])
```

Numpy array

Shape and type

```
a = np.array([1,2,3,4,5,6,7,8,9,10])  
print(a.shape)
```

(10,)

```
print(a.dtype)
```

int64

Random integers

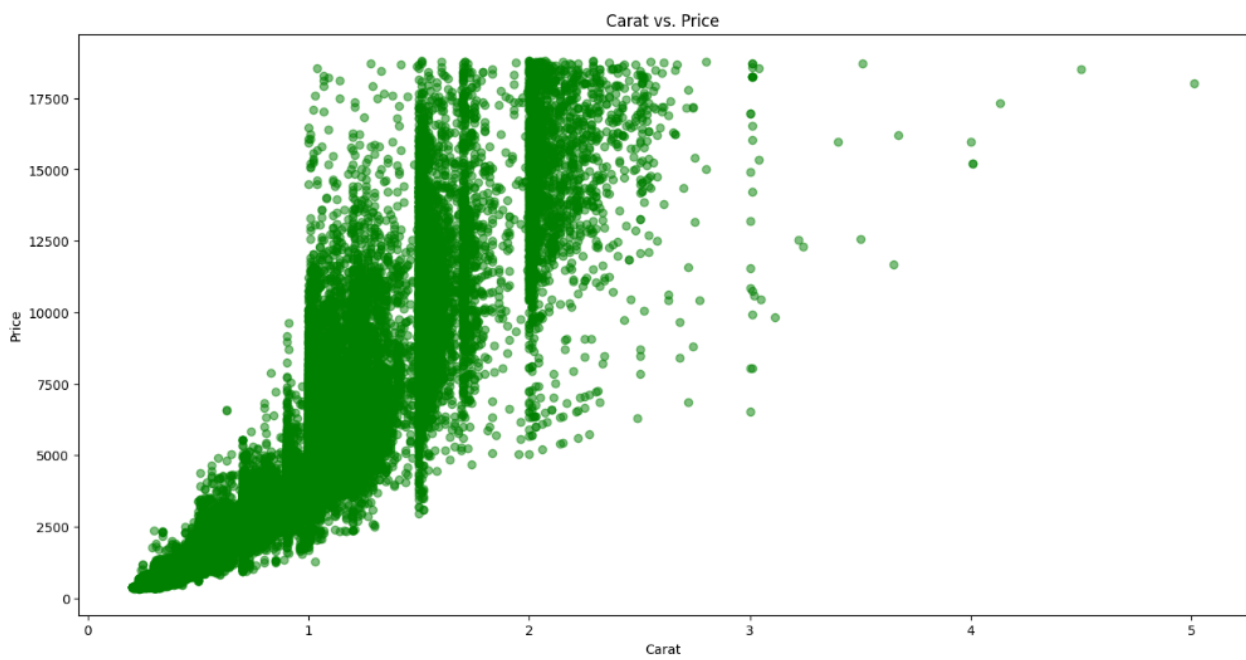
```
np.random.randint(100)
```

generates random integer between 0 and 100

Graphing

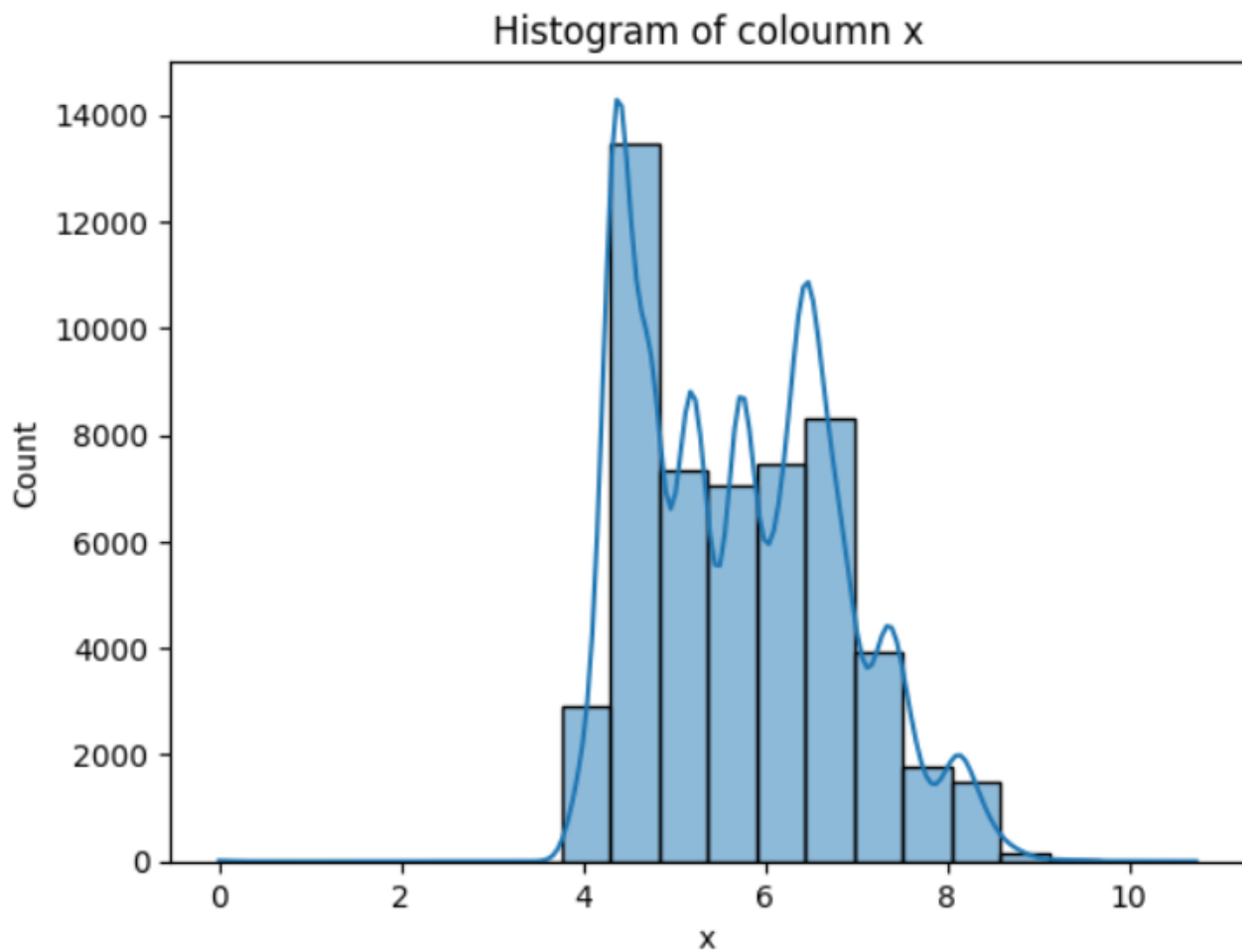
Scatter Plot

```
X= diamonds['carat']  
y= diamonds['price']  
plt.figure(figsize=(16, 8))  
plt.scatter(X, y, c='green', alpha=0.5)  
plt.xlabel('Carat')  
plt.ylabel('Price')  
plt.title('Carat vs. Price')  
plt.show()
```



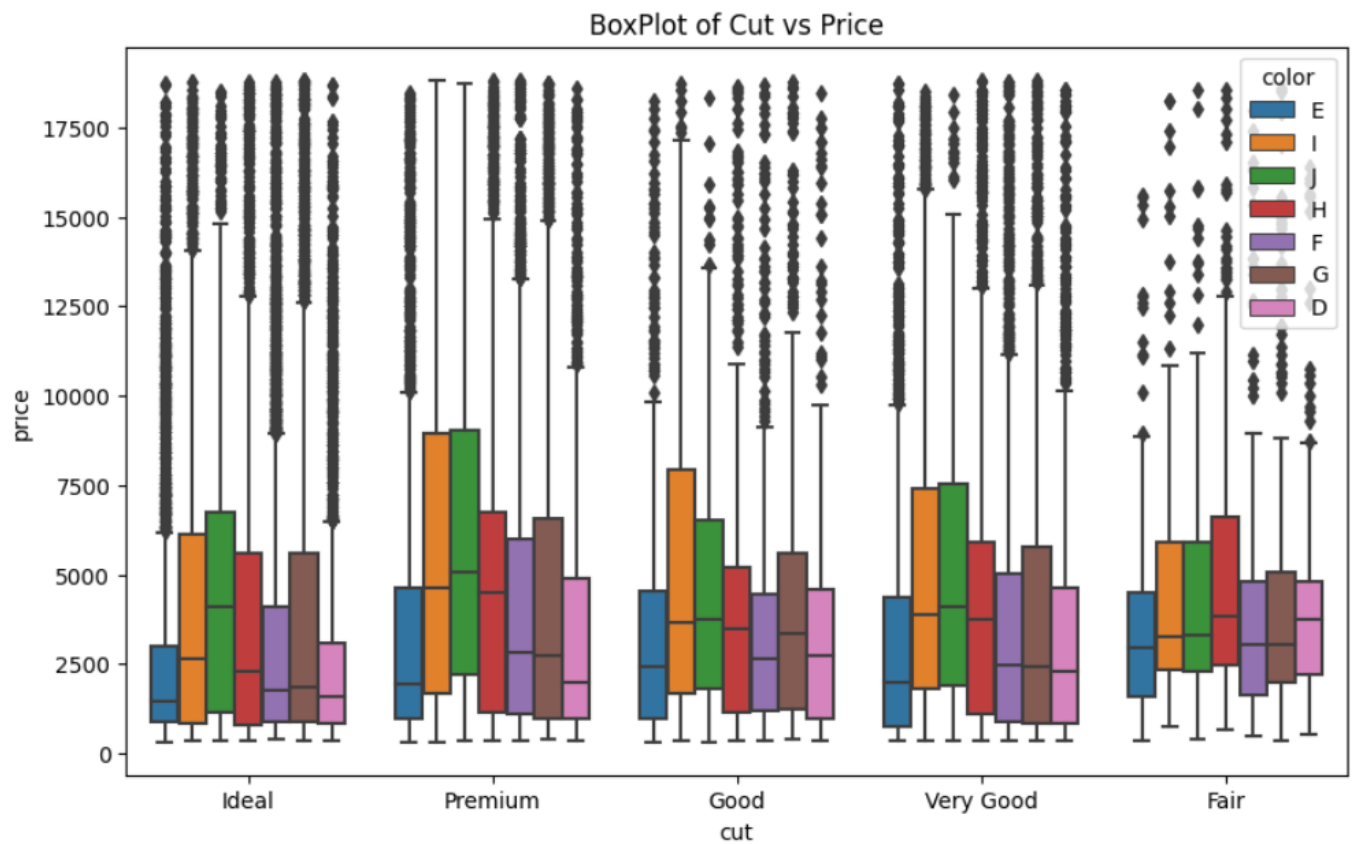
Histograms

```
sns.histplot(diamonds['x'], bins=20, kde=True)
plt.title('Histogram of coloumn x')
plt.show()
```



BoxPlot

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='cut', y='price', data=diamonds, hue='color', dodge=True)
plt.title('BoxPlot of Cut vs Price')
plt.show()
```



HeatMap

```
plt.figure(figsize=(15, 6))
sns.heatmap(diamonds.corr(), annot=True, cmap='coolwarm')
plt.title('Heatmap', fontsize=16)
plt.show()
```

