

```
import pandas as pd
diamonds = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/diamonds.csv')
print("Original Dataframe:")
print(diamonds.head())
print("\nSeries sorted by its values:")
print(diamonds.cut.value_counts().sort_values())
print("\nSeries sorted by its index:")
print(diamonds.cut.value_counts().sort_index())
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nMultiply of length, width and depth for each cut:")
print((diamonds.x*diamonds.y*diamonds.z).head())
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nConcatenate the 'diamonds' DataFrame with the 'color' Series:")
print(pd.concat([diamonds, diamonds.color], axis=1).head())
```

```
print("First 7 rows:")
print(diamonds.head(7))
print("\nAll columns:")
print(diamonds.loc[0, :])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows 0, 5, 7 and all columns:")
print(diamonds.loc[[0, 1, 2], :])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows 2 through 5 and all columns :")
print(diamonds.loc[0:2, :])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows 0 through 2 (inclusive), columns 'color' and 'price':")
print(diamonds.loc[0:2, 'color':'price'])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows 0 through 2 (inclusive), columns 'color' and 'price':")
print(diamonds.loc[0:2, 'color':'price'])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows in which the 'cut' is 'Premium', column 'color':")
print(diamonds.loc[diamonds.cut=='Premium', 'color'])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows in positions 0 and 1, columns in positions 0 and 3 :")
print(diamonds.iloc[[0, 1], [0, 3]])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows in positions 0 through 4, columns in positions 1 through 4:")
print(diamonds.iloc[0:4, 1:4])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows in positions 0 through 4 (exclusive) and all columns :")
print(diamonds.iloc[0:5, :])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nRows 2 through 5 (inclusive), columns in positions 0 through 2 (exclusive):")
print(diamonds.iloc[2:5, 0:2])
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nConcise summary of diamonds DataFrame:")
print(diamonds.info())
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nTrue memory usage by diamonds DataFrame:")
print(diamonds.info(memory_usage='deep'))
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nMemory usage for each Series (in bytes) of diamonds DataFrame:")
print(diamonds.memory_usage(deep=True))
```

```
print("Original Dataframe:")
print(diamonds.head())
print("\nSample 5 rows from the DataFrame without replacement:")
print(diamonds.sample(n=3))
```

```
print("Original Dataframe:")
print(diamonds.shape)
print("\nSample 75% of diamonds DataFrame's rows without replacement:")
result = diamonds.sample(frac=0.75, random_state=99)
print(result)
print("\nRemaining 25% of the rows:")
print(diamonds.loc[~diamonds.index.isin(result.index), :])
```

```
print("Original Dataframe:")
print(diamonds.shape)
print("\nCount the duplicate items:")
print(diamonds.clarity.duplicated().sum())
```

```
print("Original Dataframe:")
print(diamonds.shape)
print("\nDuplicate rows of diamonds DataFrame:")
print(diamonds.duplicated().sum())
```