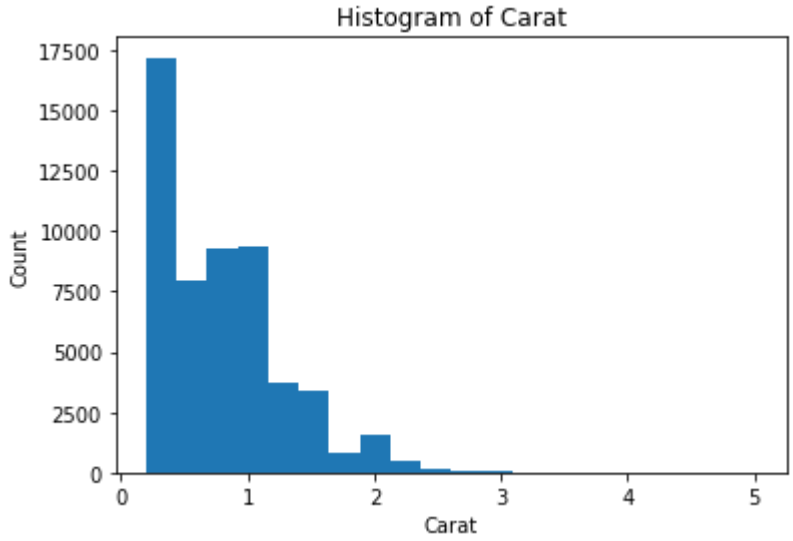
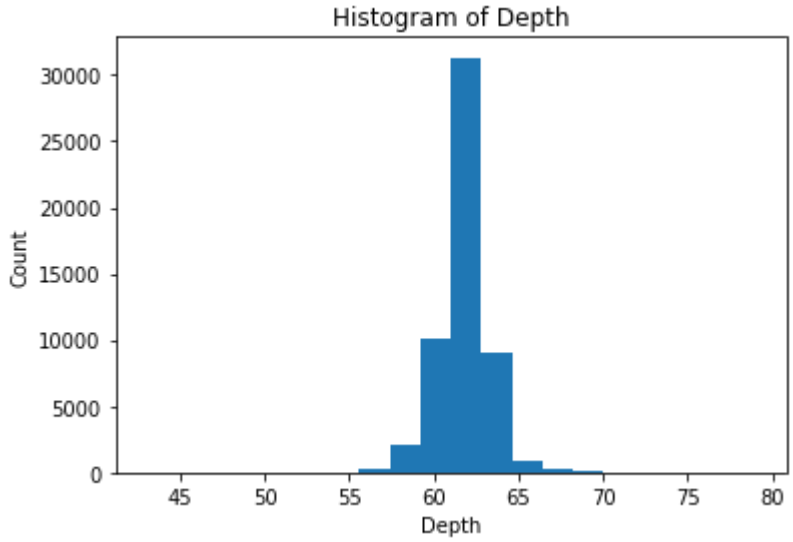
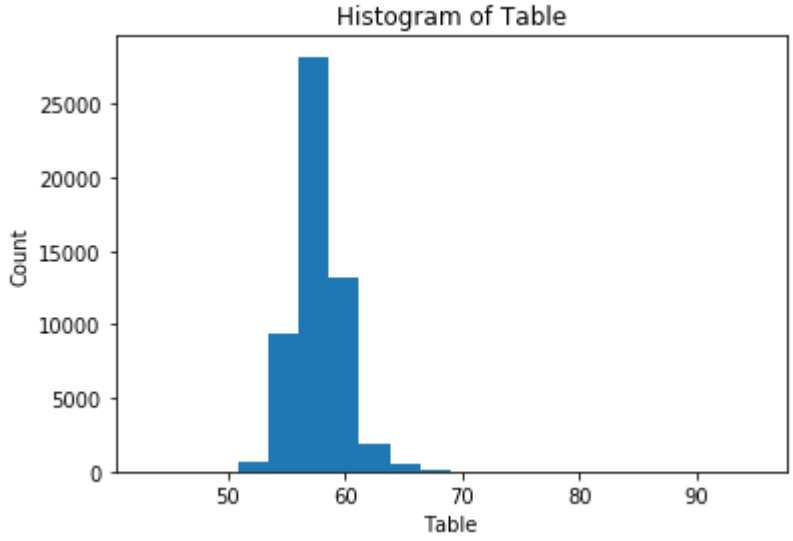
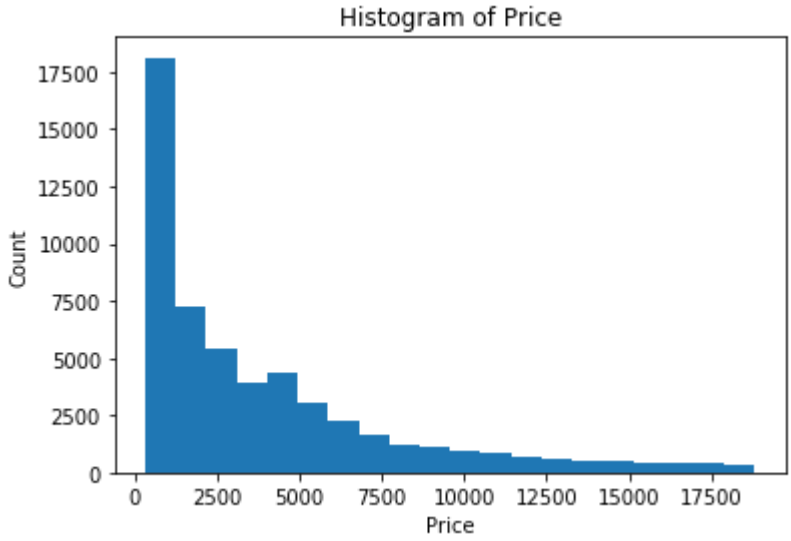
In [1]:
```
#Histograms of numeric variables: Create histograms to visualize the distribut
ion of numeric variables like carat, depth, table, and price.

import pandas as pd
import matplotlib.pyplot as plt

# Load the diamond dataset
diamonds = pd.read_csv('diamonds.csv')

# Histogram of carat
plt.hist(diamonds['carat'], bins=20)
plt.xlabel('Carat')
plt.ylabel('Count')
plt.title('Histogram of Carat')
plt.show()

# Histogram of depth
plt.hist(diamonds['depth'], bins=20)
plt.xlabel('Depth')
plt.ylabel('Count')
plt.title('Histogram of Depth')
plt.show()

# Histogram of table
plt.hist(diamonds['table'], bins=20)
plt.xlabel('Table')
plt.ylabel('Count')
plt.title('Histogram of Table')
plt.show()

# Histogram of price
plt.hist(diamonds['price'], bins=20)
plt.xlabel('Price')
plt.ylabel('Count')
plt.title('Histogram of Price')
plt.show()
```


Histogram of Carat
Count
Carat
0
2500
5000
7500
10000
12500
15000
17500
0
1
2
3
4
5

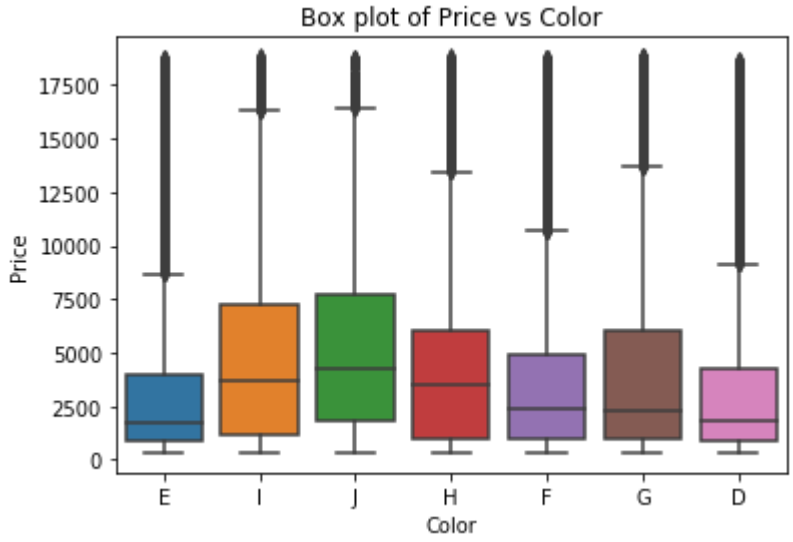

Histogram of Depth
Count
Depth
0
5000
10000
15000
20000
25000
30000
45
50
55
60
65
70
75
80

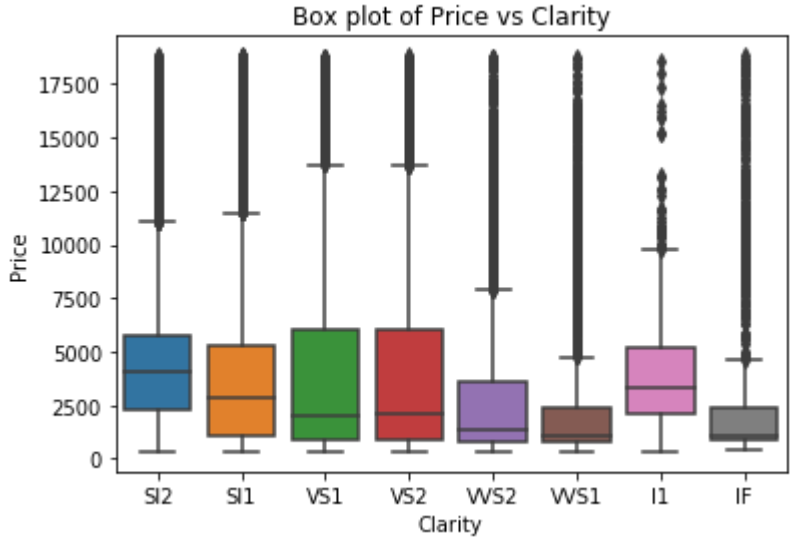

Histogram of Table
Count
Table
0
5000
10000
15000
20000
25000
50
60
70
80
90


Histogram of Price
Count
Price
0
2500
5000
7500
10000
12500
15000
17500

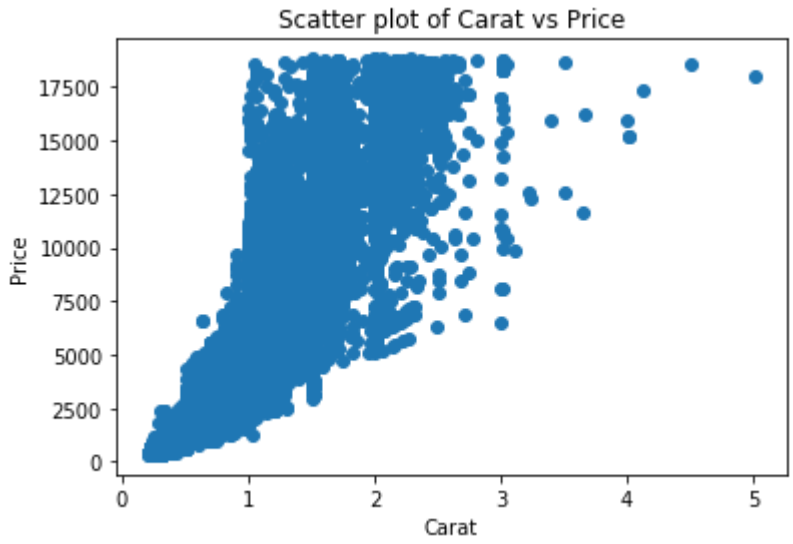
In [2]:
```
#Box plots: Create box plots to compare the distribution of price across diffe
rent categories like cut, color, and clarity.

import seaborn as sns

# Box plot of price vs cut
sns.boxplot(x='cut', y='price', data=diamonds)
plt.xlabel('Cut')
plt.ylabel('Price')
plt.title('Box plot of Price vs Cut')
plt.show()

# Box plot of price vs color
sns.boxplot(x='color', y='price', data=diamonds)
plt.xlabel('Color')
plt.ylabel('Price')
plt.title('Box plot of Price vs Color')
plt.show()

# Box plot of price vs clarity
sns.boxplot(x='clarity', y='price', data=diamonds)
plt.xlabel('Clarity')
plt.ylabel('Price')
plt.title('Box plot of Price vs Clarity')
plt.show()
```

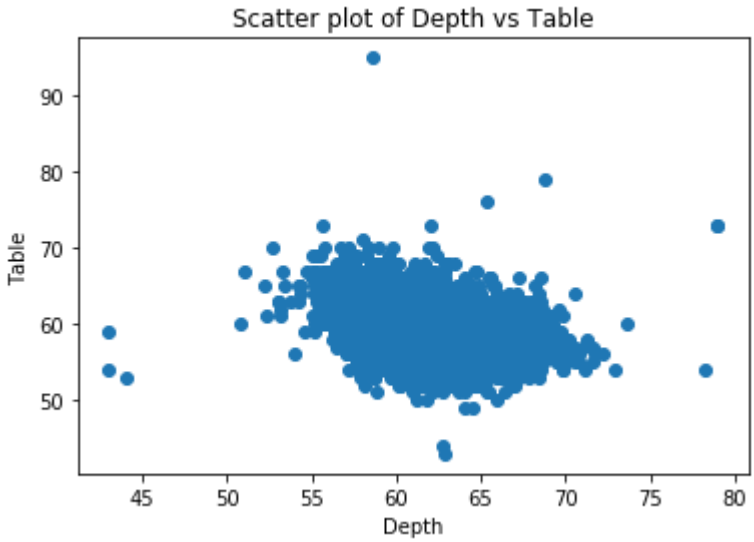

Box plot of Price vs Cut
Price
17500
15000
12500
10000
7500
5000
2500
0
Ideal
Premium
Good
Very Good
Fair
Cut


Box plot of Price vs Color
Price
17500
15000
12500
10000
7500
5000
2500
0
E
I
J
H
F
G
D
Color

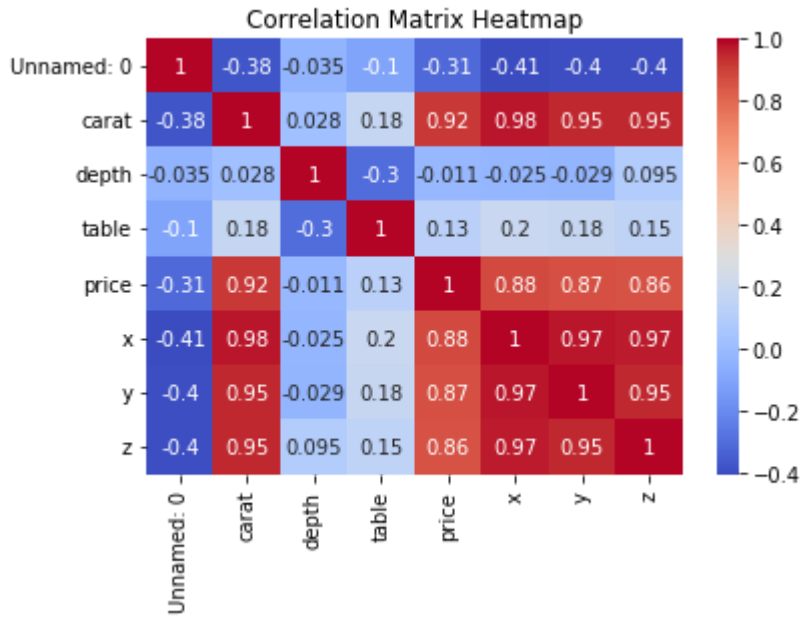

Box plot of Price vs Clarity
Price
17500
15000
12500
10000
7500
5000
2500
0
SI2
SI1
VS1
VS2
VVS2
VVS1
I1
IF
Clarity

In [3]:
```
#Scatter plots: Create scatter plots to visualize the relationship between car
at and price, and between depth and table.


# Scatter plot of carat vs price
plt.scatter(x=diamonds['carat'], y=diamonds['price'])
plt.xlabel('Carat')
plt.ylabel('Price')
plt.title('Scatter plot of Carat vs Price')
plt.show()

# Scatter plot of depth vs table
plt.scatter(x=diamonds['depth'], y=diamonds['table'])
plt.xlabel('Depth')
plt.ylabel('Table')
plt.title('Scatter plot of Depth vs Table')
plt.show()
```

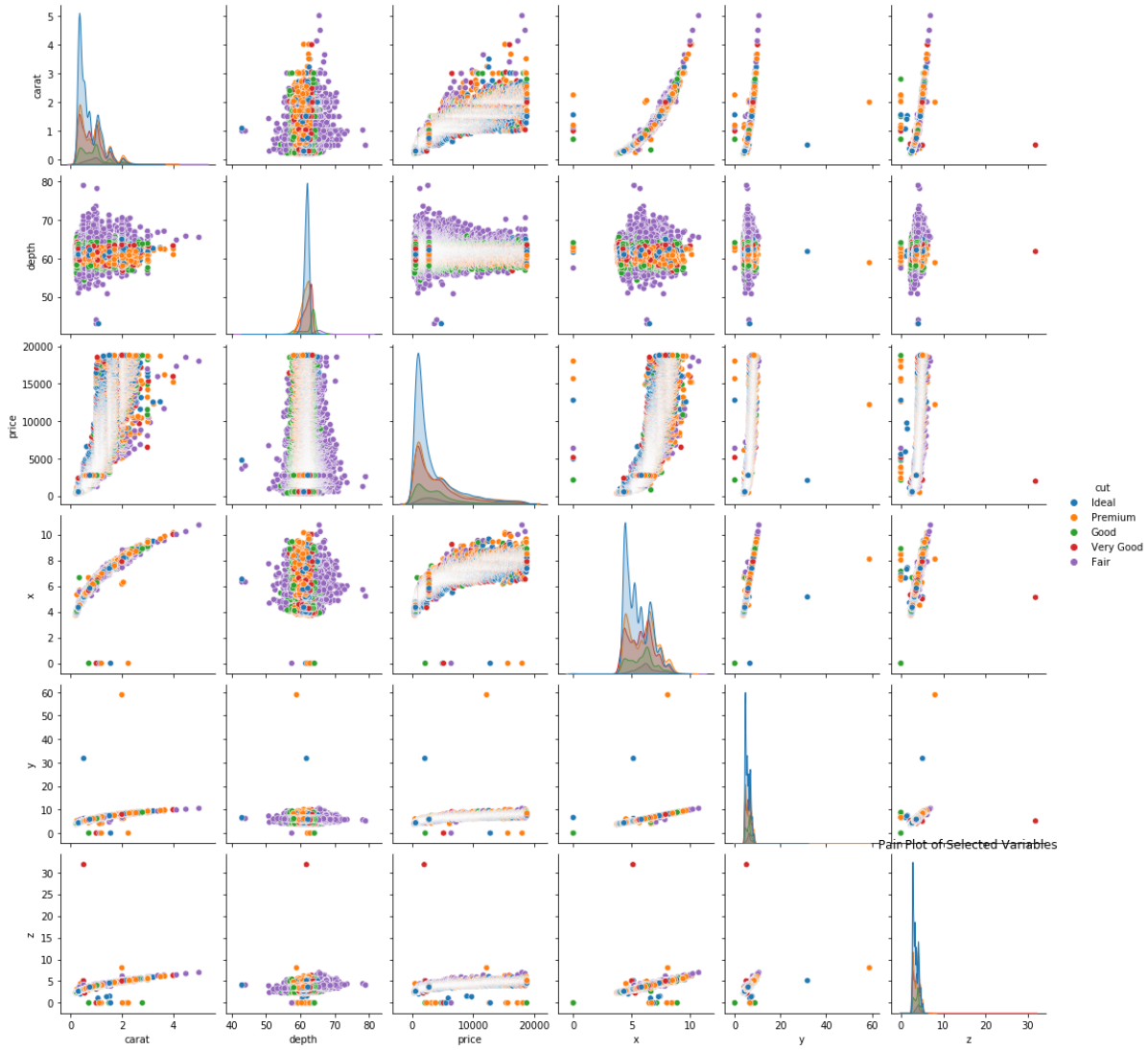


In [4]:
```
#This plot is a heatmap of the correlation matrix of the diamond dataset. It s
hows the pairwise correlations between the variables in the dataset. The annot
=True argument displays the correlation coefficients on the heatmap.

# Create a heatmap of the correlation matrix
corr_matrix = diamonds.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix Heatmap')
plt.show()
```



In [5]:
```
#This plot is a pair plot of selected variables (carat, depth, price, x, y, z) colored by the cut variable. It shows scatterplots of all pairwise combinations of the selected variables, along with histograms of each variable. The hue argument specifies that the cut variable should be used to color the points.

# Create a pair plot of selected variables
sns.pairplot(diamonds, vars=['carat', 'depth', 'price', 'x', 'y', 'z'], hue='cut')
plt.title('Pair Plot of Selected Variables')
plt.show()
```




In [ ]:
```

```