Generate print hello world using rot13 Close

```
', 'Product Description': 'Product_Des
```

Generate print hello world using rot13 Close

```
# Drop columns that are not needed for analysis
df = df.drop(columns=['Currency', 'Feature', 'Product Description'])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-17-9eae2c8dbfe5> in <cell line: 2>()
      1 # Drop columns that are not needed for analysis
----> 2 df = df.drop(columns=['Currency', 'Feature', 'Product Description'])
Next steps:
                              3 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self,
labels, errors)
   6998         if mask.any():
   6999             if errors != "ignore":
-> 7000                 raise KeyError(f"{labels[mask].tolist()} not found in
axis")
   7001             indexer = indexer[~mask]
   7002         return self.delete(indexer)

KeyError: "['Product Description'] not found in axis"
```

Next steps: Explain error

```
# Drop columns that are not needed for analysis
df = df.drop(columns=['Currency', 'Feature', 'Product_Description']) # Changed 'Product

# Display the cleaned data
print(df.head())
```

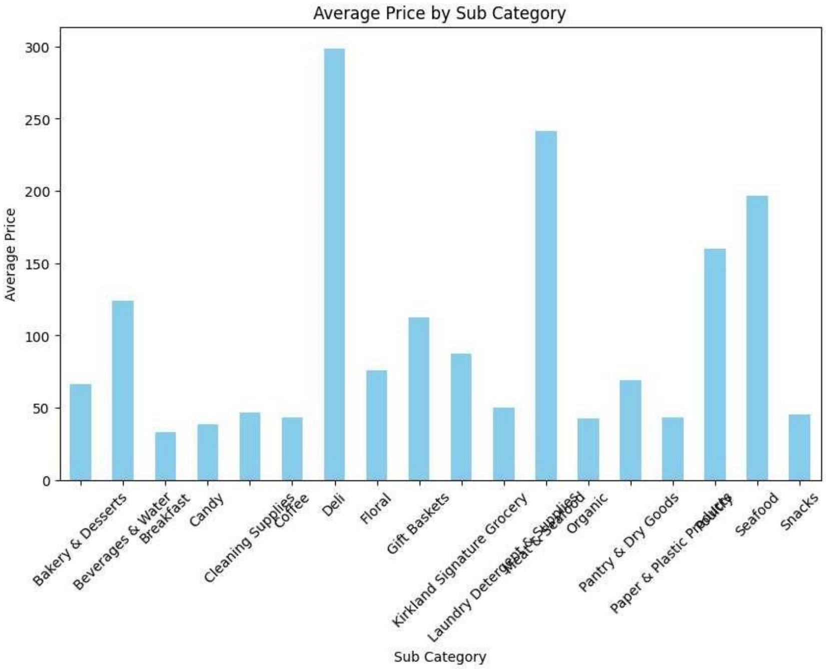
```
          Sub_Category  Price     Discount  Rating  \
0  Bakery & Desserts  56.99  No Discount     4.3
2  Bakery & Desserts  44.99  No Discount     4.1
3  Bakery & Desserts  39.99  No Discount     4.7
4  Bakery & Desserts  59.99  No Discount     4.5
5  Bakery & Desserts  59.99  No Discount     4.4

                                              Title
0  David's Cookies Mile High Peanut Butter Cake, ...
2  St Michel Madeleine, Classic French Sponge Cak...
3  David's Cookies Butter Pecan Meltaways 32 oz, ...
4  David's Cookies Premier Chocolate Cake, 7.2 lb...
5  David's Cookies Mango & Strawberry Cheesecake ...
```

```
# Calculate average price
average_price = df['Price'].mean()
print(f'Average Price: ${average_price:.2f}')

# Calculate average rating
average_rating = df['Rating'].mean()
print(f'Average Rating: {average_rating:.2f}')

# Most expensive product
most_expensive = df[df['Price']
print(f'Most Expensive Product:                                    ')

# Products with the highest rating
top_rated = df[df['Rating'] == df['Rating'].max()]
print(f'Highest Rated Product: {top_rated["Title"].values[0]}')
```

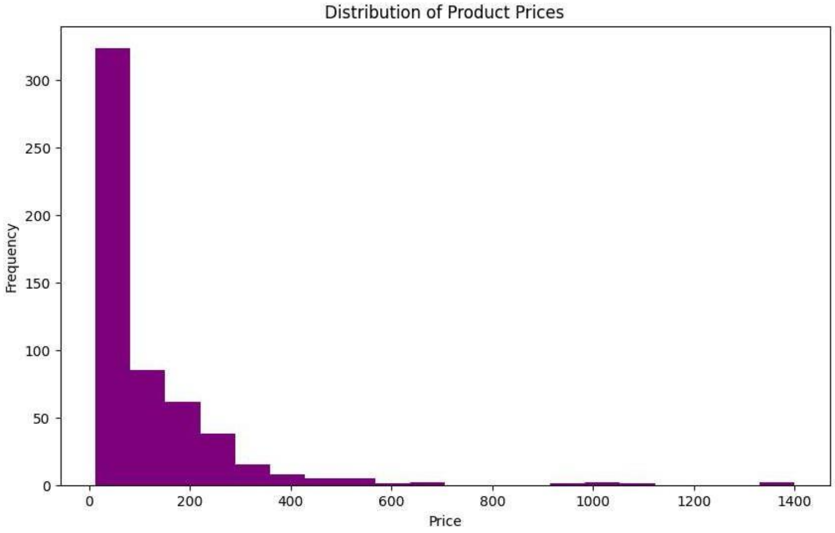
```
Average Price: $123.43
Average Rating: 4.37
Most Expensive Product: Plaza Osetra Kilo Caviar Pack
Highest Rated Product: Tiesta Tea Loose Leaf Tea Variety 3-Pack (16 oz each)
```

```
import matplotlib.pyplot as plt
```

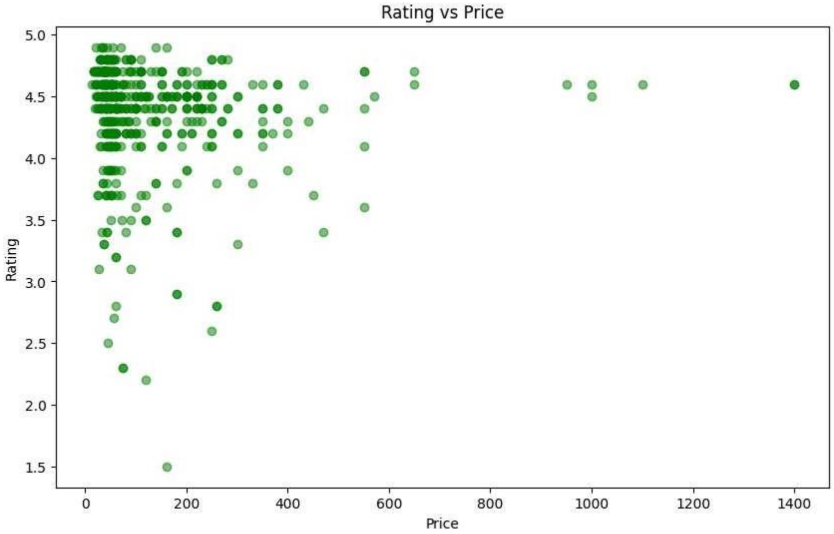
```
# Group by Sub_Category and calculate the average price
avg_price_by_category = df.groupby('Sub_Category')['Price'].mean()

# Plotting the bar chart
avg_price_by_category.plot(kind='bar', color='skyblue', figsize=(10,6))
plt.title('Average Price by Sub Category')
plt.xlabel('Sub Category')
plt.ylabel('Average Price')
plt.xticks(rotation=45)
plt.show()
```



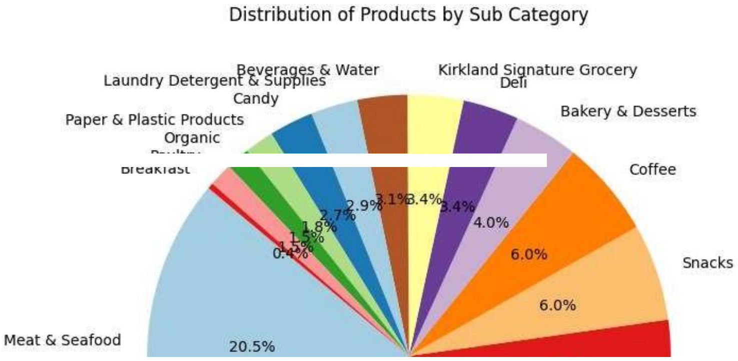
```
# Plotting the histogram
plt.figure(figsize=(10,6))
plt.hist(df['Price'], bins=20, color='purple')
plt.title('Distribution of Product Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```



```
# Scatter plot of Rating vs Price
plt.figure(figsize=(10,6))
plt.scatter(df['Price'], df['Rating'], alpha=0.5, color='green')
plt.title('Rating vs Price')
plt.xlabel('Price')
plt.ylabel('Rating')
plt.show()
```



```
# Pie chart for product distribution by sub-category
sub_category_counts = df['Sub_Category'].value_counts()

plt.figure(figsize=(8,8))
plt.pie(sub_category_counts, labels=sub_category_counts.index, autopct='%1.1f%%', star
plt.title('Distribution of Products by Sub Category')
plt.show()
```



Enter a prompt here

0 / 400

Responses may display inaccurate or offensive information that doesn't represent Google's views. Learn more