Supplement Sales Analysis

This dataset contains weeklly sales data for a variety of health and wellness supplements from january 2020 to April 2025. The data includes products in categories like Protein, Vitamins, Omega, and Amino Acids, among others, and covers multiple e-commerces such as Amazone, Walmart, and IHerb. The dataset also tracks sales in serveral location including the USA, UK, and Canada.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv('Supplement_Sales_Weekly_Expanded.csv')
df.head()
```

Out[2]:

	Date	Product Name	Category	Units Sold	Price	Revenue	Discount	Units Returned	Location
0	2020-01-06	Whey Protein	Protein	143	31.98	4573.14	0.03	2	Canada
1	2020-01-06	Vitamin C	Vitamin	139	42.51	5908.89	0.04	0	UK
2	2020-01-06	Fish Oil	Omega	161	12.91	2078.51	0.25	0	Canada
3	2020-01-06	Multivitamin	Vitamin	140	16.07	2249.80	0.08	0	Canada
4	2020-01-06	Pre-Workout	Performance	157	35.47	5568.79	0.25	3	Canada

In [3]:
```
df.shape
```

Out[3]:
```
(4384, 10)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4384 entries, 0 to 4383
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Date            4384 non-null   object 
 1   Product Name    4384 non-null   object 
 2   Category        4384 non-null   object 
 3   Units Sold      4384 non-null   int64  
 4   Price           4384 non-null   float64
 5   Revenue         4384 non-null   float64
 6   Discount        4384 non-null   float64
 7   Units Returned  4384 non-null   int64  
 8   Location        4384 non-null   object 
 9   Platform        4384 non-null   object 
dtypes: float64(3), int64(2), object(5)
memory usage: 342.6+ KB
```

In [5]:
```
df['Date'] = pd.to_datetime(df['Date'])
```

In [6]:
```
df.dtypes
```

Out[6]:
```
Date              datetime64[ns]
Product Name              object
Category                  object
Units Sold                 int64
Price                    float64
Revenue                  float64
Discount                 float64
Units Returned             int64
Location                  object
Platform                  object
dtype: object
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
Date              0
Product Name      0
Category          0
Units Sold        0
Price             0
Revenue           0
Discount          0
Units Returned    0
Location          0
Platform          0
dtype: int64
```

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
np.int64(0)
```

In [9]:
```
df.describe()
```

Out[9]:

	Date	Units Sold	Price	Revenue	Discount	Units Returned
count	4384	4384.000000	4384.000000	4384.000000	4384.000000	4384.000000
mean	2022-08-18 12:00:00	150.200274	34.781229	5226.569446	0.124398	1.531478
min	2020-01-06 00:00:00	103.000000	10.000000	1284.000000	0.000000	0.000000
25%	2021-04-26 00:00:00	142.000000	22.597500	3349.372500	0.060000	1.000000
50%	2022-08-18 12:00:00	150.000000	34.720000	5173.140000	0.120000	1.000000
75%	2023-12-11 00:00:00	158.000000	46.712500	7009.960000	0.190000	2.000000
max	2025-03-31 00:00:00	194.000000	59.970000	10761.850000	0.250000	8.000000
std	NaN	12.396099	14.198309	2192.491946	0.071792	1.258479

In [10]:
```
df.describe(include=['object'])
```

Out[10]:

	Product Name	Category	Location	Platform
count	4384	4384	4384	4384
unique	16	10	3	3
top	Whey Protein	Vitamin	Canada	iHerb
freq	274	822	1507	1499

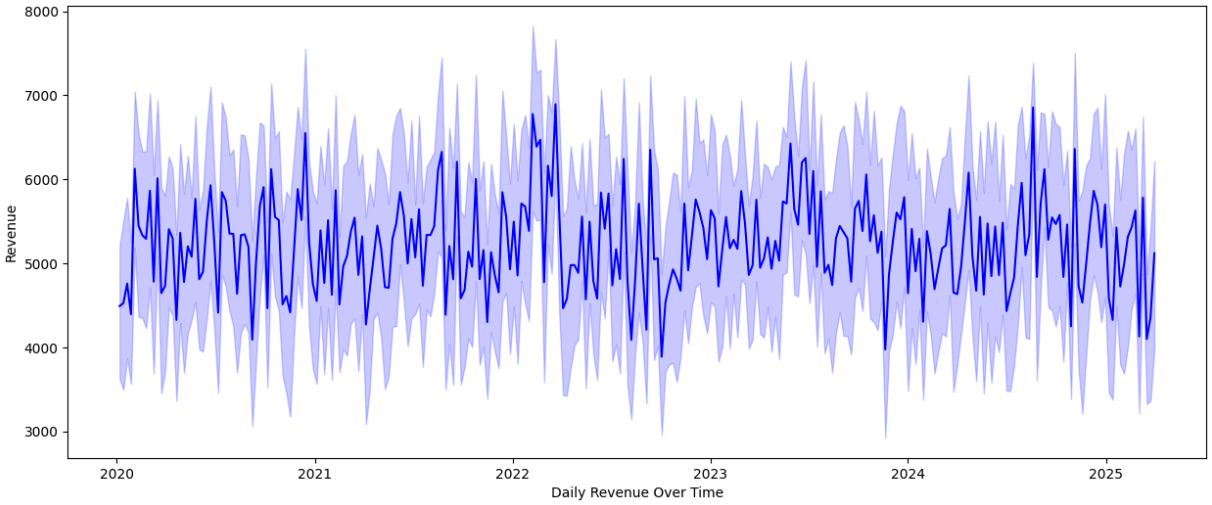
In [11]:
```
d = {
    "Revenue":"sum",
    "Units Sold":"sum",
    "Units Returned":"sum"
}
daily_sales = df.groupby('Date').agg(d)
daily_sales
```

Out[11]:

Date	Revenue	Units Sold	Units Returned
2020-01-06	71848.56	2406	19
2020-01-13	72416.18	2374	27
2020-01-20	76152.42	2370	26
2020-01-27	70306.73	2397	29
2020-02-03	98011.64	2384	34
...	...	...	...
2025-03-03	66065.44	2431	36
2025-03-10	92509.57	2411	30
2025-03-17	65590.53	2381	22
2025-03-24	69778.44	2416	27
2025-03-31	81915.03	2410	20

274 rows × 3 columns

In [12]:
```
plt.figure(figsize=(15,6))
sns.lineplot(x='Date',y='Revenue',data=df,color='b')
plt.xlabel("Daily Revenue Over Time")
plt.show()
```



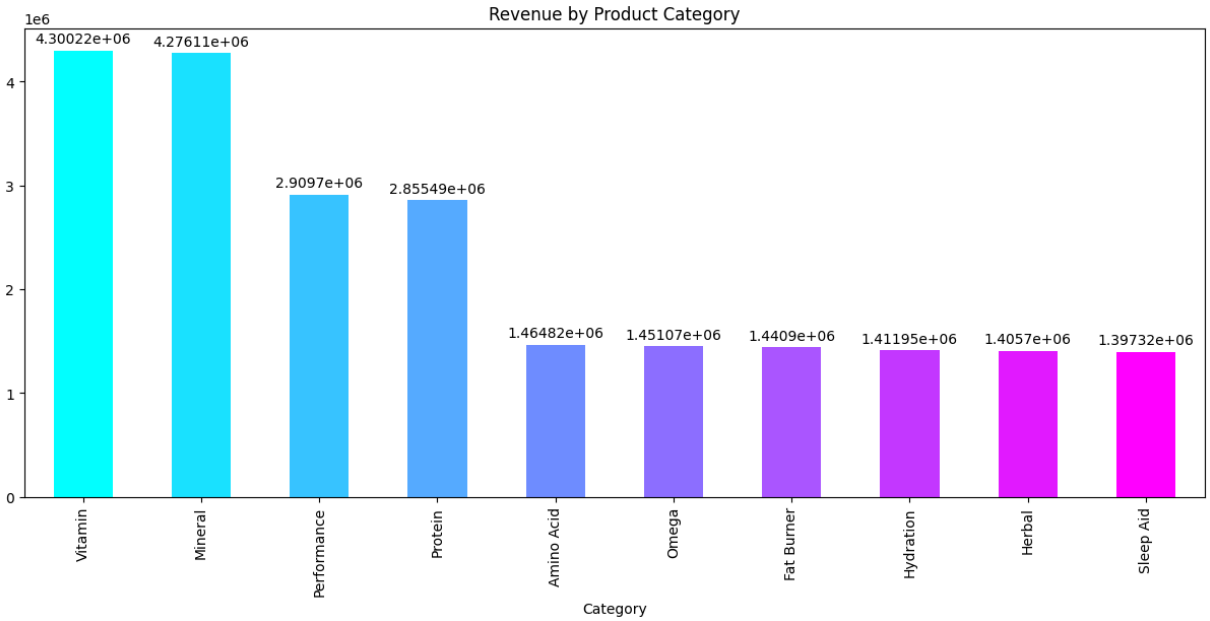
In [13]:
```
df['Category'].value_counts()
```

Out[13]:
```
Category
Vitamin        822
Mineral        822
Protein        548
Performance    548
Omega          274
Amino Acid     274
Herbal         274
Sleep Aid      274
Fat Burner     274
Hydration      274
Name: count, dtype: int64
```

In [14]:
```
category_revenue = df.groupby('Category')['Revenue'].sum().sort_values(ascending=Fa
category_revenue
```

Out[14]:
```
Category
Vitamin        4300224.68
Mineral        4276107.99
Performance    2909702.18
Protein        2855492.09
Amino Acid     1464819.63
Omega          1451065.87
Fat Burner     1440900.05
Hydration      1411951.38
Herbal         1405700.79
Sleep Aid      1397315.79
Name: Revenue, dtype: float64
```

In [15]:
```
colors = cm.get_cmap('cool',len(category_revenue))(np.arange(len(category_revenue))
ax = category_revenue.plot(kind='bar',color=colors,figsize=(15,6))
for i in ax.containers:
    ax.bar_label(i,padding=3)
ax.set_title("Revenue by Product Category")
plt.show()
```



In [16]:
```
df['Location'].value_counts()
```

Out[16]:
```
Location
Canada    1507
UK        1475
USA       1402
Name: count, dtype: int64
```

In [17]:
```
df['Platform'].value_counts()
```

Out[17]:
```
Platform
iHerb      1499
Amazon     1473
Walmart    1412
Name: count, dtype: int64
```

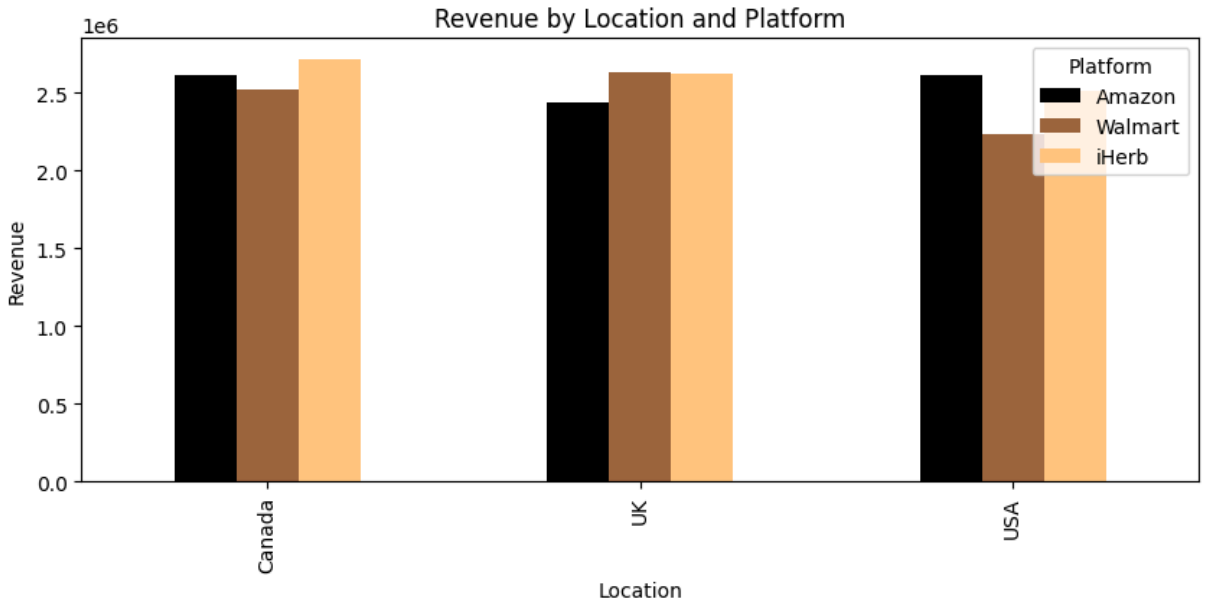
Location of platform based or revenue

In [18]:
```
loc_platform = df.groupby(['Location','Platform'])['Revenue'].sum().unstack().filln
loc_platform
```

Out[18]:

Platform	Amazon	Walmart	iHerb
Location			
Canada	2613844.28	2518639.07	2716096.38
UK	2442671.23	2637066.25	2624222.86
USA	2612936.27	2232862.30	2514941.81

In [19]:
```
colors = cm.get_cmap('copper',len(loc_platform))(np.arange(len(loc_platform)))
ax = loc_platform.plot(kind='bar',color=colors,figsize=(10,4))
ax.set_title("Revenue by Location and Platform")
ax.set_ylabel("Revenue")
plt.show()
```



In [20]:
```
df['Product Name'].value_counts()
```

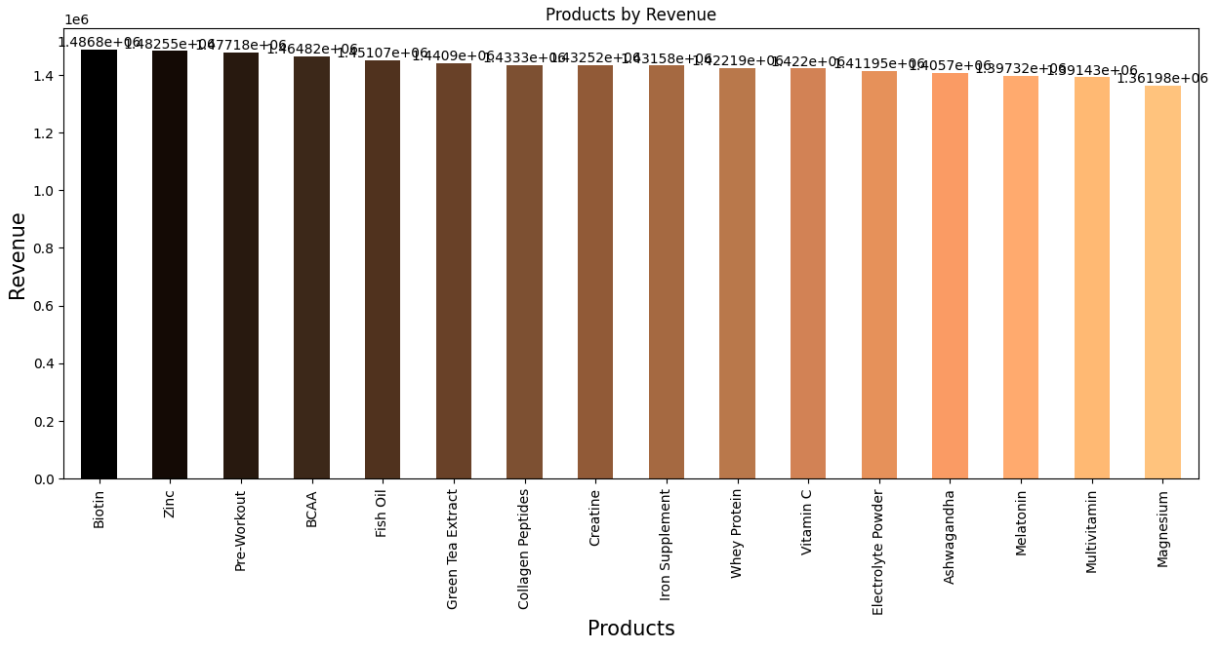
Out[20]:
```
Product Name
Whey Protein          274
Vitamin C             274
Fish Oil              274
Multivitamin          274
Pre-Workout           274
BCAA                  274
Creatine              274
Zinc                  274
Collagen Peptides     274
Magnesium             274
Ashwagandha           274
Melatonin             274
Biotin                274
Green Tea Extract     274
Iron Supplement       274
Electrolyte Powder    274
Name: count, dtype: int64
```

Products and its total revenue

In [21]:
```
top_product = df.groupby(['Product Name'])['Revenue'].sum().sort_values(ascending=F
top_product
```

Out[21]:
```
Product Name
Biotin                1486798.62
Zinc                  1482546.95
Pre-Workout           1477183.78
BCAA                  1464819.63
Fish Oil              1451065.87
Green Tea Extract     1440900.05
Collagen Peptides     1433297.24
Creatine              1432518.40
Iron Supplement       1431582.41
Whey Protein          1422194.85
Vitamin C             1421998.07
Electrolyte Powder    1411951.38
Ashwagandha           1405700.79
Melatonin             1397315.79
Multivitamin          1391427.99
Magnesium             1361978.63
Name: Revenue, dtype: float64
```

In [22]:
```
colors = cm.get_cmap('copper',len(top_product))(np.arange(len(top_product)))
ax = top_product.plot(kind='bar',color=colors,figsize=(15,6))
for i in ax.containers:
    ax.bar_label(i)
ax.set_title("Products by Revenue")
ax.set_xlabel("Products",fontsize=15)
ax.set_ylabel("Revenue",fontsize=15)
plt.show()
```



In [23]:
```
df['Return Rate'] = df['Units Returned'] / df['Units Sold']
df.head()
```

Out[23]:

	Date	Product Name	Category	Units Sold	Price	Revenue	Discount	Units Returned	Location
0	2020-01-06	Whey Protein	Protein	143	31.98	4573.14	0.03	2	Canada
1	2020-01-06	Vitamin C	Vitamin	139	42.51	5908.89	0.04	0	UK
2	2020-01-06	Fish Oil	Omega	161	12.91	2078.51	0.25	0	Canada
3	2020-01-06	Multivitamin	Vitamin	140	16.07	2249.80	0.08	0	Canada
4	2020-01-06	Pre-Workout	Performance	157	35.47	5568.79	0.25	3	Canada

In [24]:
```
df['Category'].value_counts()
```

Out[24]:
```
Category
Vitamin        822
Mineral        822
Protein        548
Performance    548
Omega          274
Amino Acid     274
Herbal         274
Sleep Aid      274
Fat Burner     274
Hydration      274
Name: count, dtype: int64
```

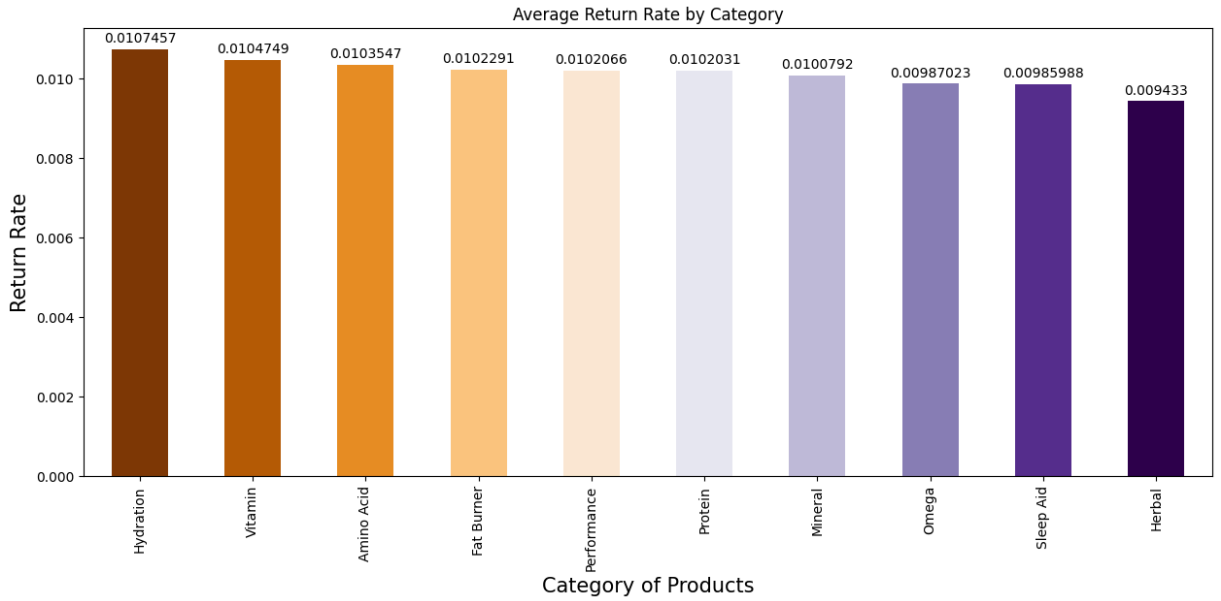
Average return rate based on categories

In [25]:
```
category_return = df.groupby('Category')['Return Rate'].mean().sort_values(ascendin
category_return
```

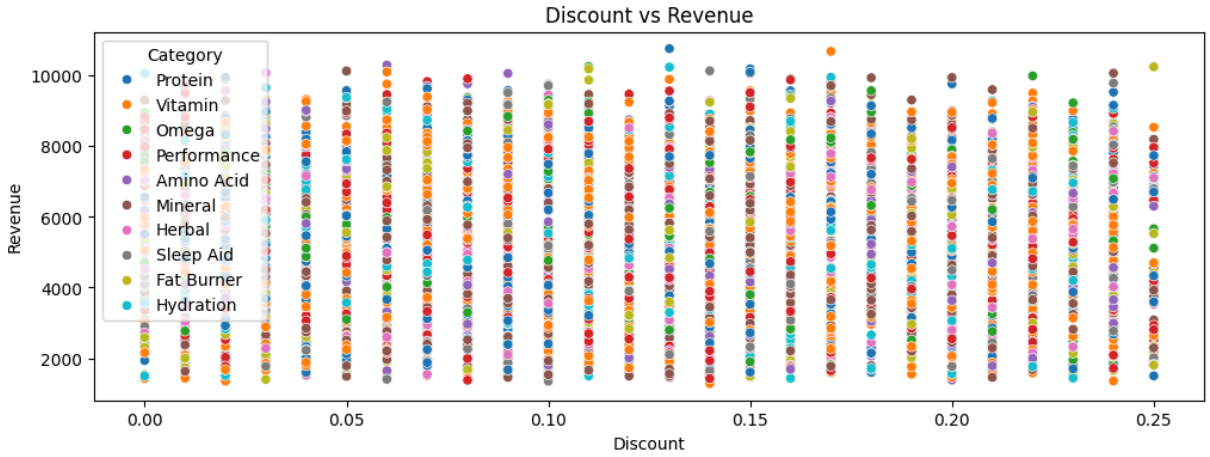
Out[25]:
```
Category
Hydration      0.010746
Vitamin        0.010475
Amino Acid     0.010355
Fat Burner     0.010229
Performance    0.010207
Protein        0.010203
Mineral        0.010079
Omega          0.009870
Sleep Aid      0.009860
Herbal         0.009433
Name: Return Rate, dtype: float64
```

In [26]:
```
colors = cm.get_cmap('PuOr',len(category_return))(np.arange(len(category_return)))
ax = category_return.plot(kind='bar',color=colors,figsize=(15,6))
for i in ax.containers:
```

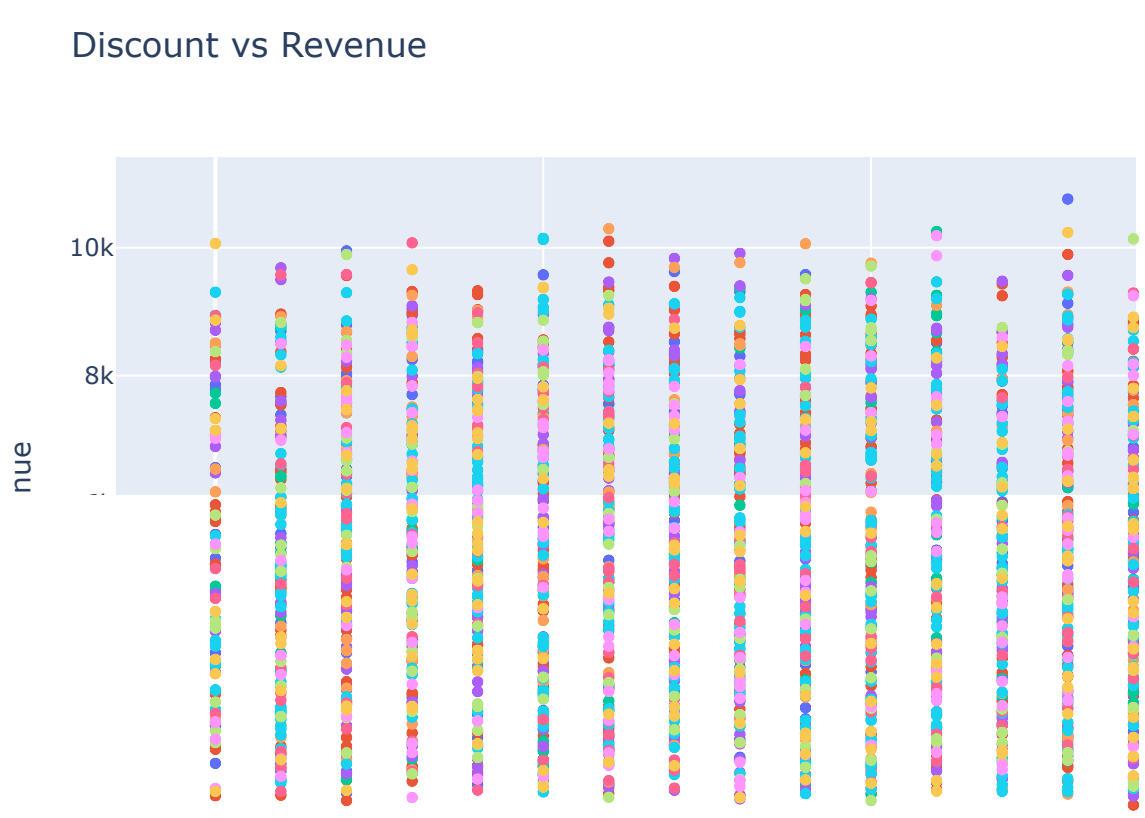
```
    ax.bar_label(i,padding=3)
ax.set_title("Average Return Rate by Category")
ax.set_xlabel("Category of Products",fontsize=15)
ax.set_ylabel("Return Rate",fontsize=15)
plt.show()
```



In [27]:
```
plt.figure(figsize=(12,4))
sns.scatterplot(data=df,x='Discount',y='Revenue',hue='Category')
plt.title("Discount vs Revenue")
plt.show()
```



In [28]:
```
fig = px.scatter(df,x='Discount',y='Revenue',color='Category',title='Discount vs Re
fig.show()
```

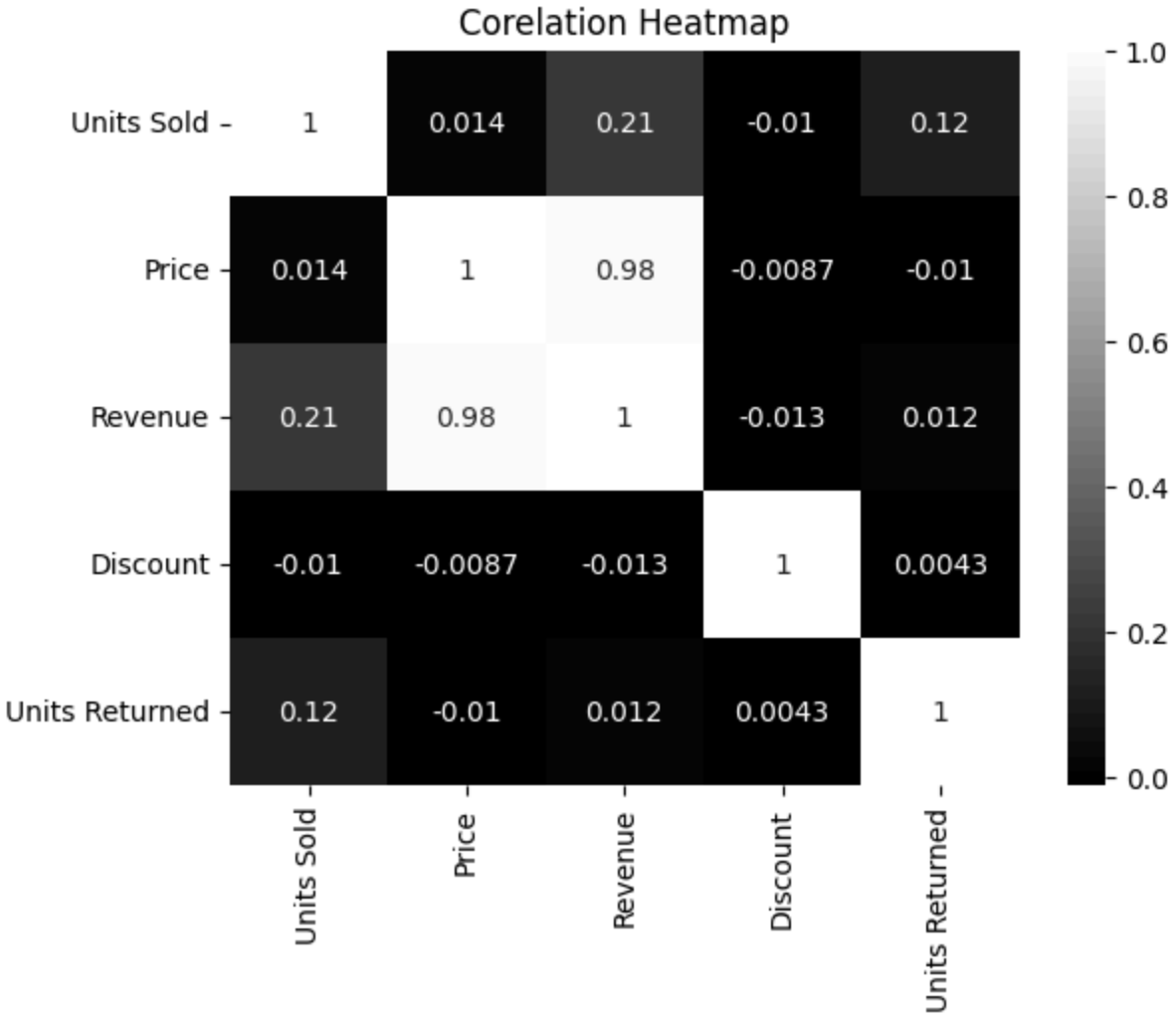


In [29]:
```
corr = df[['Units Sold','Price', 'Revenue','Discount', 'Units Returned']].corr()
corr
```

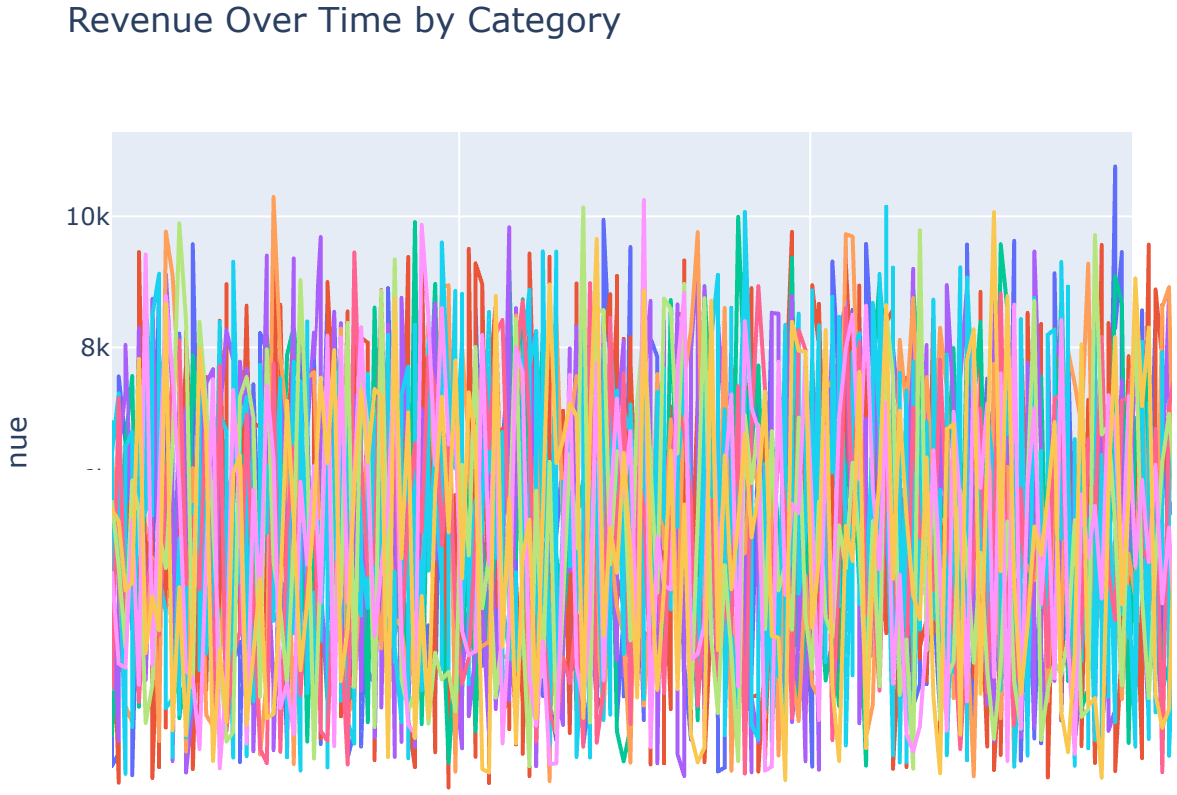
Out[29]:

	Units Sold	Price	Revenue	Discount	Units Returned
Units Sold	1.000000	0.013749	0.210462	-0.010435	0.116523
Price	0.013749	1.000000	0.977198	-0.008668	-0.010410
Revenue	0.210462	0.977198	1.000000	-0.012531	0.012432
Discount	-0.010435	-0.008668	-0.012531	1.000000	0.004276
Units Returned	0.116523	-0.010410	0.012432	0.004276	1.000000

In [30]:
```
sns.heatmap(corr,annot=True,cmap='gray')
plt.title("Corelation Heatmap")
plt.show()
```



In [31]:
```
fig = px.line(df,x='Date',y='Revenue',color='Category',title='Revenue Over Time by
fig.show()
```



In [32]:
```
df['Monthly'] = df['Date'].dt.to_period("M")
df['Monthly']
```

Out[32]:
```
0       2020-01
1       2020-01
2       2020-01
3       2020-01
4       2020-01
         ...
4379    2025-03
4380    2025-03
4381    2025-03
4382    2025-03
4383    2025-03
Name: Monthly, Length: 4384, dtype: period[M]
```

In [33]:
```
d = {
    "Revenue":"sum",
    "Units Sold":"sum",
    "Discount":"mean",
    "Units Returned":"sum"
}
monthly = df.groupby('Monthly').agg(d).reset_index(False)
monthly
```

Out[33]:

	Monthly	Revenue	Units Sold	Discount	Units Returned
0	2020-01	290723.89	9547	0.120313	101
1	2020-02	355213.26	9493	0.128125	91
2	2020-03	416547.17	12145	0.111375	123
3	2020-04	326287.92	9605	0.138281	91
4	2020-05	333210.99	9557	0.119375	90
...	...	...	...	...	...
58	2024-11	329894.33	9838	0.117813	106
59	2024-12	446728.99	12042	0.127125	99
60	2025-01	304965.15	9617	0.125156	92
61	2025-02	341768.25	9542	0.128125	103
62	2025-03	375859.01	12049	0.124625	135

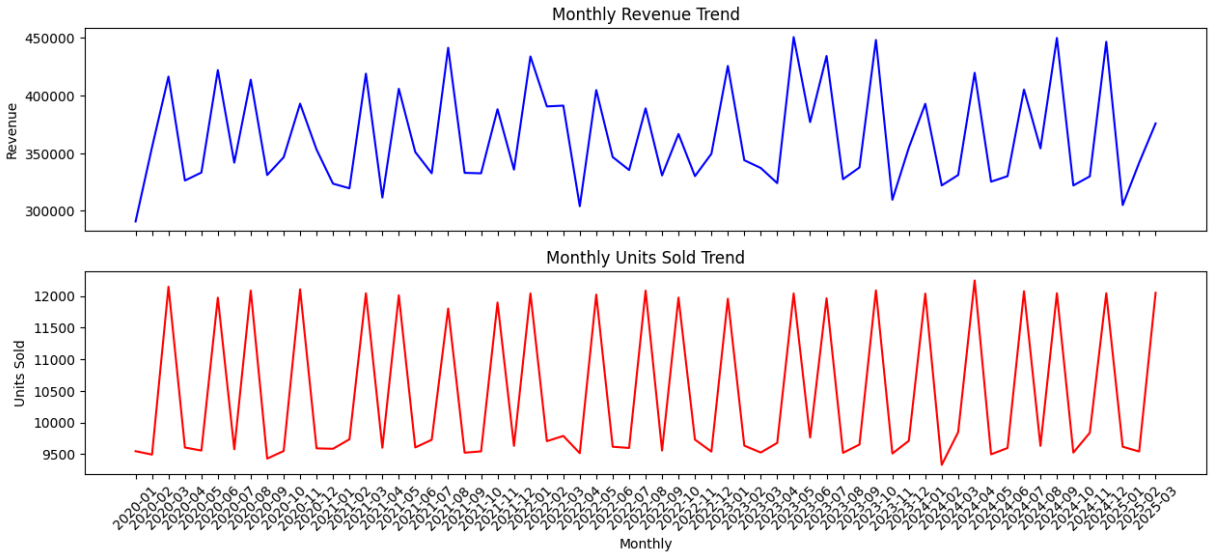
63 rows × 5 columns

In [34]:
```
monthly['Monthly'] = monthly['Monthly'].astype(str)
```

In [35]:
```
fig, axes = plt.subplots(2,1,sharex=True,figsize=(15,6))

sns.lineplot(data=monthly,x='Monthly',y='Revenue',ax=axes[0],color='b')
axes[0].set_title("Monthly Revenue Trend")
axes[0].tick_params(axis='x',rotation=45)

sns.lineplot(data=monthly,x='Monthly',y='Units Sold',ax=axes[1],color='r')
axes[1].set_title("Monthly Units Sold Trend")
axes[1].tick_params(axis='x',rotation=45)
plt.show()
```



In [36]:
```
df['Net Revenue'] = df['Revenue'] - (df['Units Returned'] * df['Price'])
```

In [37]:
```
df.head()
```

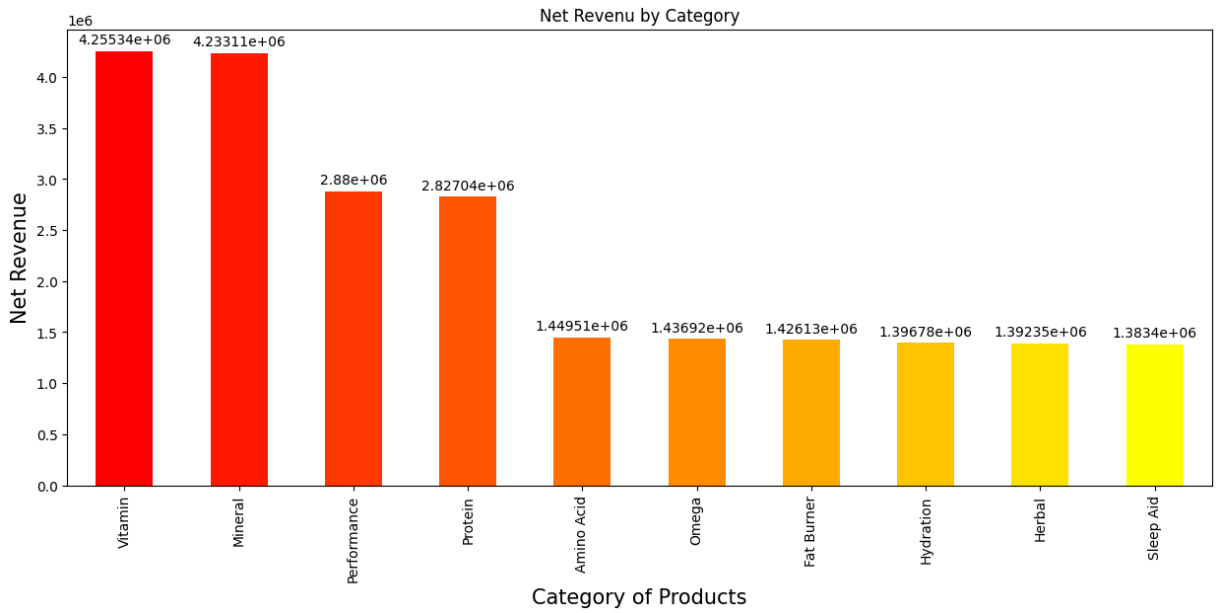
Out[37]:

	Date	Product Name	Category	Units Sold	Price	Revenue	Discount	Units Returned	Location
0	2020-01-06	Whey Protein	Protein	143	31.98	4573.14	0.03	2	Canada
1	2020-01-06	Vitamin C	Vitamin	139	42.51	5908.89	0.04	0	UK
2	2020-01-06	Fish Oil	Omega	161	12.91	2078.51	0.25	0	Canada
3	2020-01-06	Multivitamin	Vitamin	140	16.07	2249.80	0.08	0	Canada
4	2020-01-06	Pre-Workout	Performance	157	35.47	5568.79	0.25	3	Canada

In [38]:
```
category_profit = df.groupby('Category')['Net Revenue'].sum().sort_values(ascending
category_profit
```

Out[38]:
```
Category
Vitamin        4255337.22
Mineral        4233108.50
Performance    2879997.43
Protein        2827040.48
Amino Acid     1449514.18
Omega          1436916.36
Fat Burner     1426130.43
Hydration      1396778.95
Herbal         1392349.37
Sleep Aid      1383401.64
Name: Net Revenue, dtype: float64
```

In [39]:
```
colors = cm.get_cmap('autumn',len(category_profit))(np.arange(len(category_profit))
ax = category_profit.plot(kind='bar',color=colors,figsize=(15,6))
for i in ax.containers:
    ax.bar_label(i,padding=3)
ax.set_title("Net Revenu by Category")
ax.set_xlabel("Category of Products",fontsize=15)
ax.set_ylabel("Net Revenue",fontsize=15)
plt.show()
```



Colclusion

From this analysis, Vitamine, Mineral category consume highest revenue from other categories. Hydration, Vitamin these are
Return rates is high in comparion other categories. Biotin, Zinc these product more revenue consume. However, A future step could involve
applying machine learning models to predict medal counts or analyzing total revenue across product and categories .