In [19]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [20]:
```
df = pd.read_csv('retail_sales_data.csv', parse_dates=['invoice_date'])
df.head()
```

Out[20]:

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invo
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24

In [21]:
```
df.describe()
```

Out[21]:

	age	quantity	price
count	99457.000000	99457.000000	99457.000000
mean	43.427089	3.003429	689.256321
std	14.990054	1.413025	941.184567
min	18.000000	1.000000	5.230000
25%	30.000000	2.000000	45.450000
50%	43.000000	3.000000	203.300000
75%	56.000000	4.000000	1200.320000
max	69.000000	5.000000	5250.000000

In [22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99457 entries, 0 to 99456
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   invoice_no      99457 non-null  object 
 1   customer_id     99457 non-null  object 
 2   gender          99457 non-null  object 
 3   age             99457 non-null  int64  
 4   category        99457 non-null  object 
 5   quantity        99457 non-null  int64  
 6   price           99457 non-null  float64
 7   payment_method  99457 non-null  object 
 8   invoice_date    99457 non-null  object 
 9   shopping_mall   99457 non-null  object 
dtypes: float64(1), int64(2), object(7)
memory usage: 7.6+ MB
```

In [23]:
```
df.isna().sum()
```

Out[23]:
```
invoice_no        0
customer_id       0
gender            0
age               0
category          0
quantity          0
price             0
payment_method    0
invoice_date      0
shopping_mall     0
dtype: int64
```

In [24]:
```
df.drop(['invoice_no', 'customer_id', 'gender', 'age', 'payment_method'], a
df.head()
```

Out[24]:

	category	quantity	price	invoice_date	shopping_mall
0	Clothing	5	1500.40	5/8/2022	Kanyon
1	Shoes	3	1800.51	12/12/2021	Forum Istanbul
2	Clothing	1	300.08	9/11/2021	Metrocity
3	Shoes	5	3000.85	16/05/2021	Metropol AVM
4	Books	4	60.60	24/10/2021	Kanyon

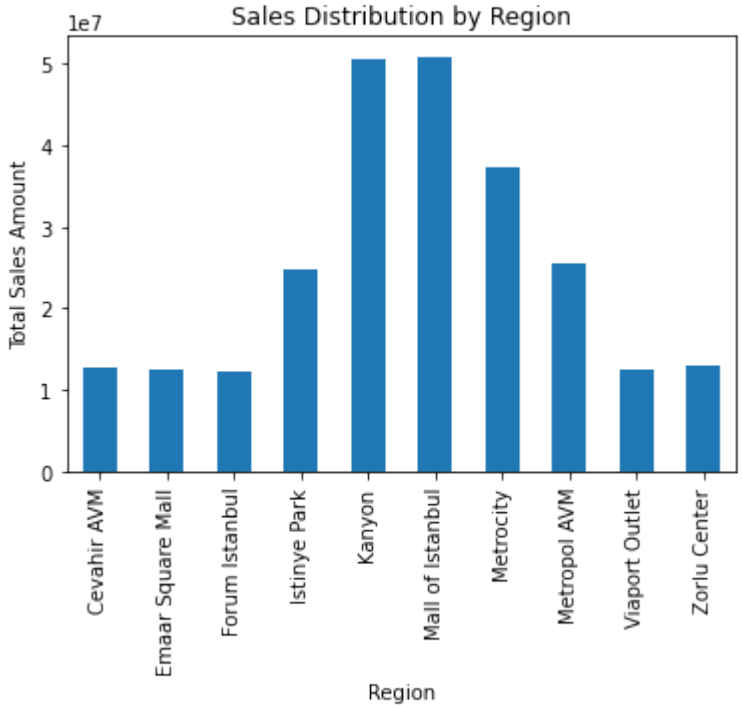
In [29]:
```
df['Sales'] = df['quantity']*df['price']
df.head()
```

Out[29]:

	category	quantity	price	invoice_date	shopping_mall	Sales
0	Clothing	5	1500.40	5/8/2022	Kanyon	7502.00
1	Shoes	3	1800.51	12/12/2021	Forum Istanbul	5401.53
2	Clothing	1	300.08	9/11/2021	Metrocity	300.08
3	Shoes	5	3000.85	16/05/2021	Metropol AVM	15004.25
4	Books	4	60.60	24/10/2021	Kanyon	242.40

In [30]:
```
#Group data by region and calculate total amount

region_sales = df.groupby('shopping_mall')['Sales'].sum()
region_sales.plot(kind='bar')
plt.title('Sales Distribution by Region')
plt.xlabel('Region')
plt.ylabel('Total Sales Amount')
plt.show()
```

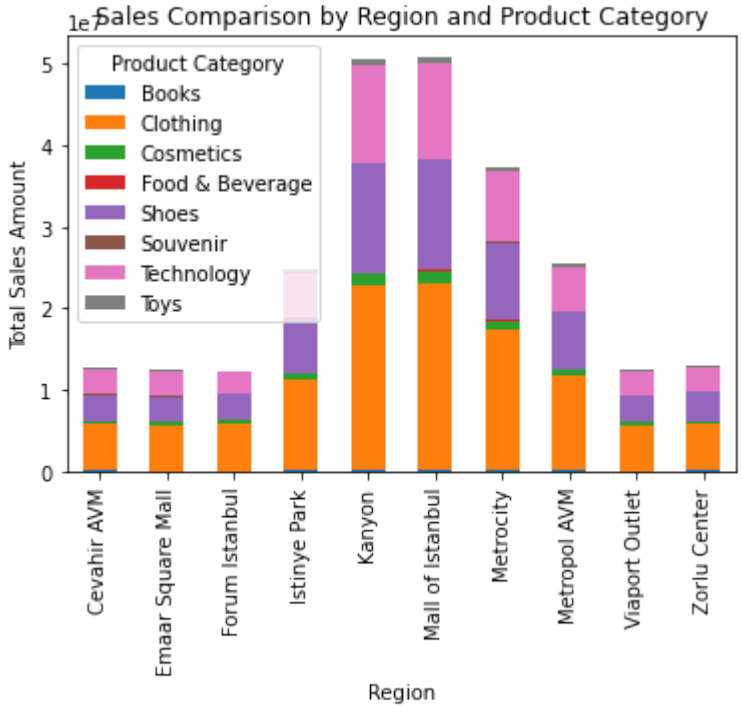


In [36]:
```
print('Top performing region is:', region_sales.idxmax())
```

```
Top performing region is: Mall of Istanbul
```

In [41]:
```
#Stacked bar plot to compare sales amounts across regions and categories

region_category_sales = df.groupby(['shopping_mall', 'category'])['Sales'].
region_category_sales.plot(kind='bar', stacked=True)
plt.title('Sales Comparison by Region and Product Category')
plt.xlabel('Region')
plt.ylabel('Total Sales Amount')
plt.legend(title='Product Category')
plt.show()
```



In [ ]: