



Tiva Clothing Brand EDA

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: camp = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_sto
channel = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_
customer = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion
product = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_
sales = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_st
items = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_st
stock = pd.read_csv("C://Users//hp//Downloads//archive (1)//dataset_fashion_st
```

```
In [4]: camp.head()
```

```
Out[4]:
```

	campaign_id	campaign_name	start_date	end_date	channel	discount_type
0	1	Spring Flash Sale	2025-04-01	2025-04-07	Email	Percentage
1	2	Easter Promotion	2025-04-08	2025-04-15	Social Media	Fixed
2	3	Mother's Day Campaign	2025-05-01	2025-05-09	Email	Percentage
3	4	Mid-Season Clearance	2025-05-10	2025-05-19	App Mobile	Percentage
4	5	TIVA Week	2025-05-20	2025-05-31	Social Media	Percentage

```
In [5]: customer
```

Out[5]:

	customer_id	country	age_range	signup_date
0	1	France	56-65	2025-04-24
1	2	France	36-45	2025-02-24
2	3	Netherlands	46-55	2024-04-12
3	4	Italy	36-45	2025-03-11
4	5	Spain	26-35	2025-04-26
...
995	996	Germany	16-25	2025-05-08
996	997	Italy	56-65	2025-03-29
997	998	France	36-45	2025-01-07
998	999	Germany	56-65	2025-02-21
999	1000	Spain	56-65	2024-01-06

1000 rows × 4 columns

In [6]: product

Out[6]:

	product_id	product_name	category	brand	color	size	catalog_price	cost
0	1	Soft Wrap Dress	Dresses	Tiva	Green	S	40.41	
1	2	Soft Wrap Tee	T-Shirts	Tiva	White	S	78.45	
2	3	Soft Linen Tee	T-Shirts	Tiva	Green	XL	23.90	
3	4	Soft Ribbed Tee	T-Shirts	Tiva	White	S	60.00	
4	5	Soft Wrap Trousers	Pants	Tiva	Blue	M	36.84	
...
495	496	Tailored High-Waist Trousers	Pants	Tiva	Black	S	54.61	
496	497	Dresses Drop 8	Dresses	Tiva	Black	L	36.05	
497	498	T-Shirts Drop 8	T-Shirts	Tiva	White	L	38.33	
498	499	Sleepwear Drop 4	Sleepwear	Tiva	Green	M	30.07	
499	500	Dresses Drop 9	Dresses	Tiva	White	XS	38.03	

500 rows × 9 columns

In [7]: channel

Out[7]:

	channel	description
0	E-commerce	Official online store
1	App Mobile	Brand mobile app

In [8]:

```
sales
```

Out[8]:

	sale_id	channel	discounted	total_amount	sale_date	customer_id	country
0	10	E-commerce	0	299.70	2025-05-21	195	France
1	100	App Mobile	0	681.05	2025-04-21	518	Gerron
2	1000	E-commerce	0	324.50	2025-05-20	439	Gerron
3	1001	E-commerce	0	287.85	2025-04-05	349	Gerron
4	1003	App Mobile	0	430.64	2025-06-06	727	Portugal
...
900	992	App Mobile	1	214.08	2025-05-13	375	Gerron
901	993	E-commerce	0	311.37	2025-04-21	99	Spain
902	994	App Mobile	1	477.09	2025-05-15	798	Portugal
903	995	E-commerce	0	489.02	2025-05-19	565	Spain
904	999	E-commerce	0	109.62	2025-06-16	416	France

905 rows × 7 columns

In [9]:

```
items
```

Out[9]:

	item_id	sale_id	product_id	quantity	original_price	unit_price	discount_
0	2270	658	403	1	81.80	81.80	
1	1170	336	284	1	81.79	81.79	
2	2496	1255	71	1	80.76	80.76	
3	1273	331	98	1	78.52	78.52	
4	1829	1079	98	1	78.52	78.52	
...	
2248	516	1300	334	5	20.82	20.82	
2249	1240	727	334	5	20.82	20.82	
2250	2394	172	246	5	28.98	20.29	
2251	1631	986	195	5	16.69	16.69	
2252	685	724	346	5	15.31	15.31	

2253 rows × 13 columns

In [10]:

stock

Out[10]:

	country	product_id	stock_quantity
0	France	1	61
1	France	2	24
2	France	3	81
3	France	4	70
4	France	5	30
...
995	Germany	496	1
996	Germany	497	1
997	Germany	498	1
998	Germany	499	1
999	Germany	500	1

	country	product_id	stock_quantity
0	France	1	61
1	France	2	24
2	France	3	81
3	France	4	70
4	France	5	30
...
995	Germany	496	1
996	Germany	497	1
997	Germany	498	1
998	Germany	499	1
999	Germany	500	1

...
995	Germany	496	1
996	Germany	497	1
997	Germany	498	1
998	Germany	499	1
999	Germany	500	1

1000 rows × 3 columns

In [11]: `camp.columns`

Out[11]: Index(['campaign_id', 'campaign_name', 'start_date', 'end_date', 'channel',
'discount_type', 'discount_value'],
dtype='object')

In [12]: `channel.columns`

Out[12]: Index(['channel', 'description'], dtype='object')

In [13]: `customer.columns`

Out[13]: Index(['customer_id', 'country', 'age_range', 'signup_date'], dtype='object')

In [14]: `product.columns`

Out[14]: Index(['product_id', 'product_name', 'category', 'brand', 'color', 'size',
'catalog_price', 'cost_price', 'gender'],
dtype='object')

In [15]: `sales.columns`

Out[15]: Index(['sale_id', 'channel', 'discounted', 'total_amount', 'sale_date',
'customer_id', 'country'],
dtype='object')

In [16]: `items.columns`

```
Out[16]: Index(['item_id', 'sale_id', 'product_id', 'quantity', 'original_price',
               'unit_price', 'discount_applied', 'discount_percent', 'discounted',
               'item_total', 'sale_date', 'channel', 'channel_campaigns'],
              dtype='object')
```

```
In [17]: stock.columns
```

```
Out[17]: Index(['country', 'product_id', 'stock_quantity'], dtype='object')
```

```
In [18]: camp
```

```
Out[18]:
```

	campaign_id	campaign_name	start_date	end_date	channel	discount_type
0	1	Spring Flash Sale	2025-04-01	2025-04-07	Email	Percentage
1	2	Easter Promotion	2025-04-08	2025-04-15	Social Media	Fixed
2	3	Mother's Day Campaign	2025-05-01	2025-05-09	Email	Percentage
3	4	Mid-Season Clearance	2025-05-10	2025-05-19	App Mobile	Percentage
4	5	TIVA Week	2025-05-20	2025-05-31	Social Media	Percentage
5	6	June Price Drop	2025-06-01	2025-06-09	Website Banner	Percentage
6	7	Early Summer Deals	2025-06-10	2025-06-17	Email	Fixed

```
In [19]: camp.isnull().sum()
```

```
Out[19]: campaign_id      0
campaign_name      0
start_date        0
end_date          0
channel           0
discount_type     0
discount_value    0
dtype: int64
```

```
In [20]: camp.dtypes
```

```
Out[20]: campaign_id      int64
campaign_name      object
start_date        object
end_date          object
channel           object
discount_type     object
discount_value    object
dtype: object
```

```
In [21]: cmp = camp.copy()
```

```
In [22]: cmp
```

```
Out[22]:
```

	campaign_id	campaign_name	start_date	end_date	channel	discount_type
0	1	Spring Flash Sale	2025-04-01	2025-04-07	Email	Percentage
1	2	Easter Promotion	2025-04-08	2025-04-15	Social Media	Fixed
2	3	Mother's Day Campaign	2025-05-01	2025-05-09	Email	Percentage
3	4	Mid-Season Clearance	2025-05-10	2025-05-19	App Mobile	Percentage
4	5	TIVA Week	2025-05-20	2025-05-31	Social Media	Percentage
5	6	June Price Drop	2025-06-01	2025-06-09	Website Banner	Percentage
6	7	Early Summer Deals	2025-06-10	2025-06-17	Email	Fixed

```
In [23]: cmp.dtypes
```

```
Out[23]: campaign_id      int64
campaign_name    object
start_date       object
end_date         object
channel          object
discount_type    object
discount_value   object
dtype: object
```

```
In [24]: cmp['start_date']=pd.to_datetime(cmp['start_date'])
cmp['end_date']=pd.to_datetime(cmp['end_date'])
```

```
In [25]: cmp.dtypes
```

```
Out[25]: campaign_id      int64
campaign_name    object
start_date       datetime64[ns]
end_date         datetime64[ns]
channel          object
discount_type    object
discount_value   object
dtype: object
```

```
In [ ]:
```

```
In [26]: def clean_discount(x):
try:
```

```

        return float(str(x).replace('%','').strip())
    except Exception as e:
        return np.nan

```

```
In [27]: cmp['discount_value'] = cmp['discount_value'].apply(clean_discount)
```

```
In [28]: cmp
```

```
Out[28]:
```

	campaign_id	campaign_name	start_date	end_date	channel	discount_type
0	1	Spring Flash Sale	2025-04-01	2025-04-07	Email	Percentage
1	2	Easter Promotion	2025-04-08	2025-04-15	Social Media	Fixed
2	3	Mother's Day Campaign	2025-05-01	2025-05-09	Email	Percentage
3	4	Mid-Season Clearance	2025-05-10	2025-05-19	App Mobile	Percentage
4	5	TIVA Week	2025-05-20	2025-05-31	Social Media	Percentage
5	6	June Price Drop	2025-06-01	2025-06-09	Website Banner	Percentage
6	7	Early Summer Deals	2025-06-10	2025-06-17	Email	Fixed

```
In [29]: cmp['campaign_name'].unique()
```

```
Out[29]: array(['Spring Flash Sale', 'Easter Promotion', 'Mother's Day Campaign',
               'Mid-Season Clearance', 'TIVA Week', 'June Price Drop',
               'Early Summer Deals'], dtype=object)
```

```
In [30]: cmp['channel'].value_counts()
```

```
Out[30]: channel
Email          3
Social Media   2
App Mobile     1
Website Banner 1
Name: count, dtype: int64
```

```
In [31]: customer.dtypes
```

```
Out[31]: customer_id    int64
country              object
age_range            object
signup_date          object
dtype: object
```

```
In [32]: customer_c = customer.copy()
```



```
In [33]: customer_c['signup_date']=pd.to_datetime(customer_c['signup_date'])
```

```
In [34]: customer_c.dtypes
```

```
Out[34]: customer_id      int64  
country      object  
age_range    object  
signup_date  datetime64[ns]  
dtype: object
```

```
In [35]: customer_c.isna().sum()
```

```
Out[35]: customer_id    0  
country      0  
age_range    0  
signup_date  0  
dtype: int64
```

```
In [36]: customer_c.duplicated().sum()
```

```
Out[36]: 0
```

```
In [37]: customer_c['country'].unique()
```

```
Out[37]: array(['France', 'Netherlands', 'Italy', 'Spain', 'Germany', 'Portugal'],  
              dtype=object)
```

```
In [38]: product_c=product.copy()
```

```
In [39]: product_c
```

```
Out[39]:
```

	product_id	product_name	category	brand	color	size	catalog_price	cost_price
0	1	Soft Wrap Dress	Dresses	Tiva	Green	S	40.41	30.00
1	2	Soft Wrap Tee	T-Shirts	Tiva	White	S	78.45	50.00
2	3	Soft Linen Tee	T-Shirts	Tiva	Green	XL	23.90	15.00
3	4	Soft Ribbed Tee	T-Shirts	Tiva	White	S	60.00	40.00
4	5	Soft Wrap Trousers	Pants	Tiva	Blue	M	36.84	25.00
...
495	496	Tailored High-Waist Trousers	Pants	Tiva	Black	S	54.61	35.00
496	497	Dresses Drop 8	Dresses	Tiva	Black	L	36.05	25.00
497	498	T-Shirts Drop 8	T-Shirts	Tiva	White	L	38.33	25.00
498	499	Sleepwear Drop 4	Sleepwear	Tiva	Green	M	30.07	20.00
499	500	Dresses Drop 9	Dresses	Tiva	White	XS	38.03	25.00

500 rows × 9 columns

```
In [40]: product_c['brand'].value_counts()
```

```
Out[40]: brand
Tiva      500
Name: count, dtype: int64
```

```
In [41]: product_c.isna().sum()
```

```
Out[41]: product_id      0
product_name    0
category        0
brand           0
color           0
size            0
catalog_price   0
cost_price      0
gender          0
dtype: int64
```

```
In [42]: product_c.duplicated().sum()
```

```
Out[42]: 0
```

```
In [43]: sales_c=sales.copy()
```

```
In [44]: sales_c
```

```
Out[44]:
```

	sale_id	channel	discounted	total_amount	sale_date	customer_id	country
0	10	E-commerce	0	299.70	2025-05-21	195	France
1	100	App Mobile	0	681.05	2025-04-21	518	Gerron
2	1000	E-commerce	0	324.50	2025-05-20	439	Gerron
3	1001	E-commerce	0	287.85	2025-04-05	349	Gerron
4	1003	App Mobile	0	430.64	2025-06-06	727	Portugal
...
900	992	App Mobile	1	214.08	2025-05-13	375	Gerron
901	993	E-commerce	0	311.37	2025-04-21	99	Spain
902	994	App Mobile	1	477.09	2025-05-15	798	Portugal
903	995	E-commerce	0	489.02	2025-05-19	565	Spain
904	999	E-commerce	0	109.62	2025-06-16	416	France

905 rows × 7 columns

```
In [45]: sales_c.dtypes
```

```
Out[45]: sale_id      int64
channel      object
discounted    int64
total_amount  float64
sale_date     object
customer_id   int64
country       object
dtype: object
```

```
In [46]: sales_c['sale_date']=pd.to_datetime(sales_c['sale_date'])
```

```
In [47]: sales_c.dtypes
```

```
Out[47]: sale_id          int64
channel          object
discounted       int64
total_amount     float64
sale_date        datetime64[ns]
customer_id      int64
country          object
dtype: object
```

```
In [48]: sales_c.isna().sum()
```

```
Out[48]: sale_id          0
channel          0
discounted       0
total_amount     0
sale_date        0
customer_id      0
country          0
dtype: int64
```

```
In [49]: sales_c['channel'].value_counts()
```

```
Out[49]: channel
E-commerce    473
App Mobile    432
Name: count, dtype: int64
```

```
In [50]: sales_c.duplicated().sum()
```

```
Out[50]: 0
```

```
In [51]: items_c=items.copy()
```

```
In [52]: items_c.dtypes
```

```
Out[52]: item_id          int64
sale_id          int64
product_id       int64
quantity         int64
original_price    float64
unit_price        float64
discount_applied  float64
discount_percent  object
discounted        int64
item_total        float64
sale_date         object
channel           object
channel_campaigns object
dtype: object
```

```
In [53]: items_c.head()
```

```
Out[53]:
```

	item_id	sale_id	product_id	quantity	original_price	unit_price	discount_app
0	2270	658	403	1	81.80	81.80	
1	1170	336	284	1	81.79	81.79	
2	2496	1255	71	1	80.76	80.76	
3	1273	331	98	1	78.52	78.52	
4	1829	1079	98	1	78.52	78.52	

```
In [54]: items_c['discount_percent']=items_c['discount_percent'].apply(clean_discount)
```

```
In [55]: items_c.head()
```

```
Out[55]:
```

	item_id	sale_id	product_id	quantity	original_price	unit_price	discount_app
0	2270	658	403	1	81.80	81.80	
1	1170	336	284	1	81.79	81.79	
2	2496	1255	71	1	80.76	80.76	
3	1273	331	98	1	78.52	78.52	
4	1829	1079	98	1	78.52	78.52	

```
In [56]: items_c['sale_date']=pd.to_datetime(items_c['sale_date'])
```

```
In [57]: items_c.dtypes
```

```
Out[57]: item_id          int64
        sale_id         int64
        product_id      int64
        quantity        int64
        original_price   float64
        unit_price       float64
        discount_applied float64
        discount_percent float64
        discounted       int64
        item_total       float64
        sale_date        datetime64[ns]
        channel          object
        channel_campaigns object
        dtype: object
```

```
In [58]: stock_c=stock.copy()
```

```
In [59]: stock_c
```

```
Out[59]:
```

	country	product_id	stock_quantity
0	France	1	61
1	France	2	24
2	France	3	81
3	France	4	70
4	France	5	30
...
995	Germany	496	1
996	Germany	497	1
997	Germany	498	1
998	Germany	499	1
999	Germany	500	1

1000 rows × 3 columns

Perfoming EDA

Q1. Who are our most valueable customers?

Goal : Identify to customer segments that derive most revenue

```
In [63]: # analyze revenue by a customer
merge_df = pd.merge(sales, customer, on="customer_id", how="inner")
```

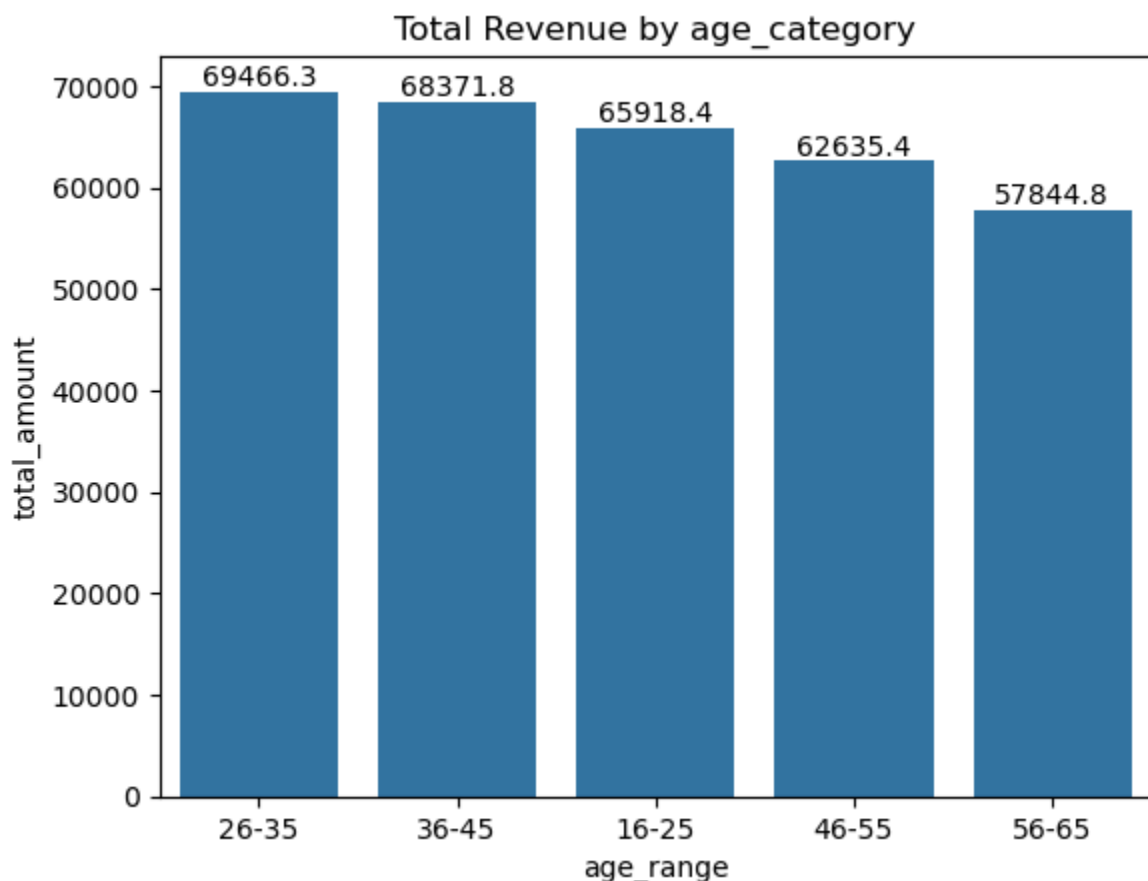
```
Revenue_by_age= merge_df.groupby('age_range')['total_amount'].sum().reset_index()
Revenue_by_age=Revenue_by_age.sort_values(by='total_amount',ascending=False)
```

In [64]: Revenue_by_age

```
Out[64]:
```

	age_range	total_amount
1	26-35	69466.31
2	36-45	68371.81
0	16-25	65918.37
3	46-55	62635.41
4	56-65	57844.76

```
In [65]: br=sns.barplot(data=Revenue_by_age,x='age_range',y='total_amount')
br.bar_label(br.containers[0])
plt.title('Total Revenue by age_category')
plt.figure(figsize=(10,6))
plt.tight_layout()
plt.show()
```



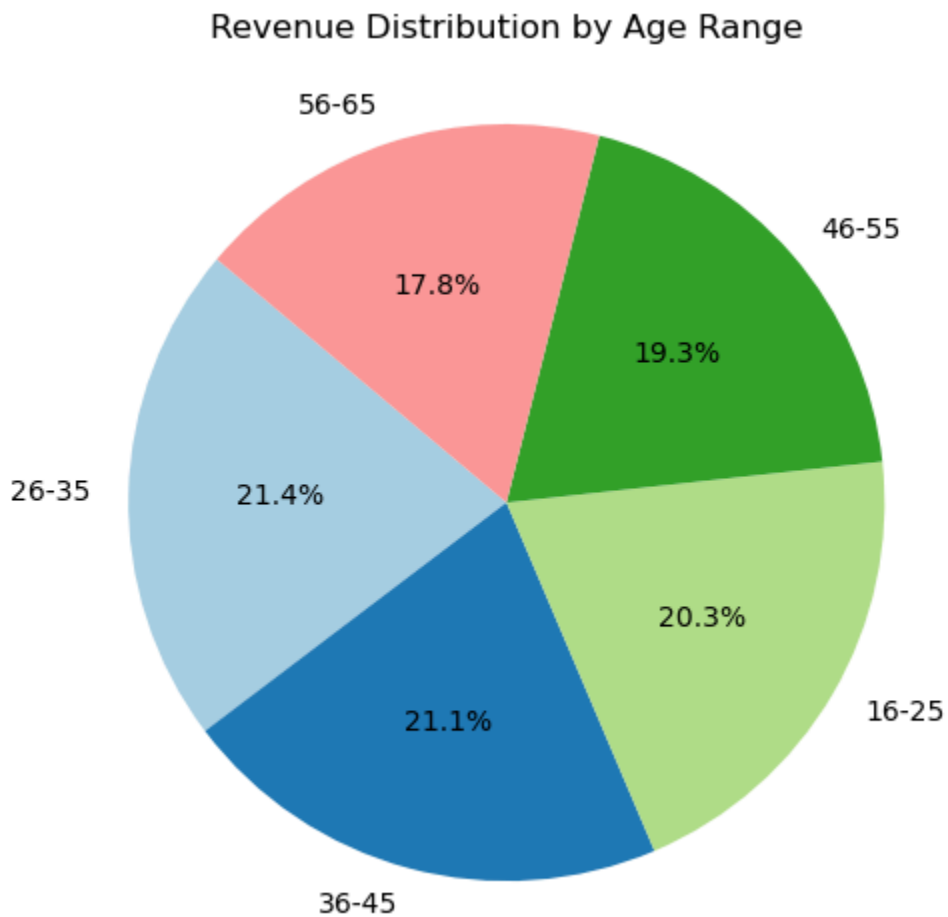
<Figure size 1000x600 with 0 Axes>

Between 26-35 age category is top most and 56-65 is lowest as compare

to other age-categories

```
In [67]: plt.figure(figsize=(5,5))
plt.pie(Revenue_by_age["total_amount"],
        labels=Revenue_by_age["age_range"],
        autopct='%1.1f%%', # Show percentage
        startangle=140,
        colors=plt.cm.Paired.colors)

plt.title("Revenue Distribution by Age Range")
plt.axis("equal")
plt.tight_layout()
plt.show()
```



By above chart we clear that our most of revenue genrate by younger and adults age categories(16-25 & 26-35) and(46-55)

```
In [69]: Revenue_by_age
```


Out[69]:

	age_range	total_amount
--	-----------	--------------

1	26-35	69466.31
2	36-45	68371.81
0	16-25	65918.37
3	46-55	62635.41
4	56-65	57844.76

In [70]: *#Revenue by country*
revenue_by_country = sales_c.groupby('country')['total_amount'].sum().reset_index()
revenue_by_country = revenue_by_country.sort_values(by='total_amount', ascending=False)

In [71]: sales_c

Out[71]:

	sale_id	channel	discounted	total_amount	sale_date	customer_id	country
--	---------	---------	------------	--------------	-----------	-------------	---------

0	10	E-commerce	0	299.70	2025-05-21	195	France
1	100	App Mobile	0	681.05	2025-04-21	518	Gerron
2	1000	E-commerce	0	324.50	2025-05-20	439	Gerron
3	1001	E-commerce	0	287.85	2025-04-05	349	Gerron
4	1003	App Mobile	0	430.64	2025-06-06	727	Portugal
...
900	992	App Mobile	1	214.08	2025-05-13	375	Gerron
901	993	E-commerce	0	311.37	2025-04-21	99	Spain
902	994	App Mobile	1	477.09	2025-05-15	798	Portugal
903	995	E-commerce	0	489.02	2025-05-19	565	Spain
904	999	E-commerce	0	109.62	2025-06-16	416	France

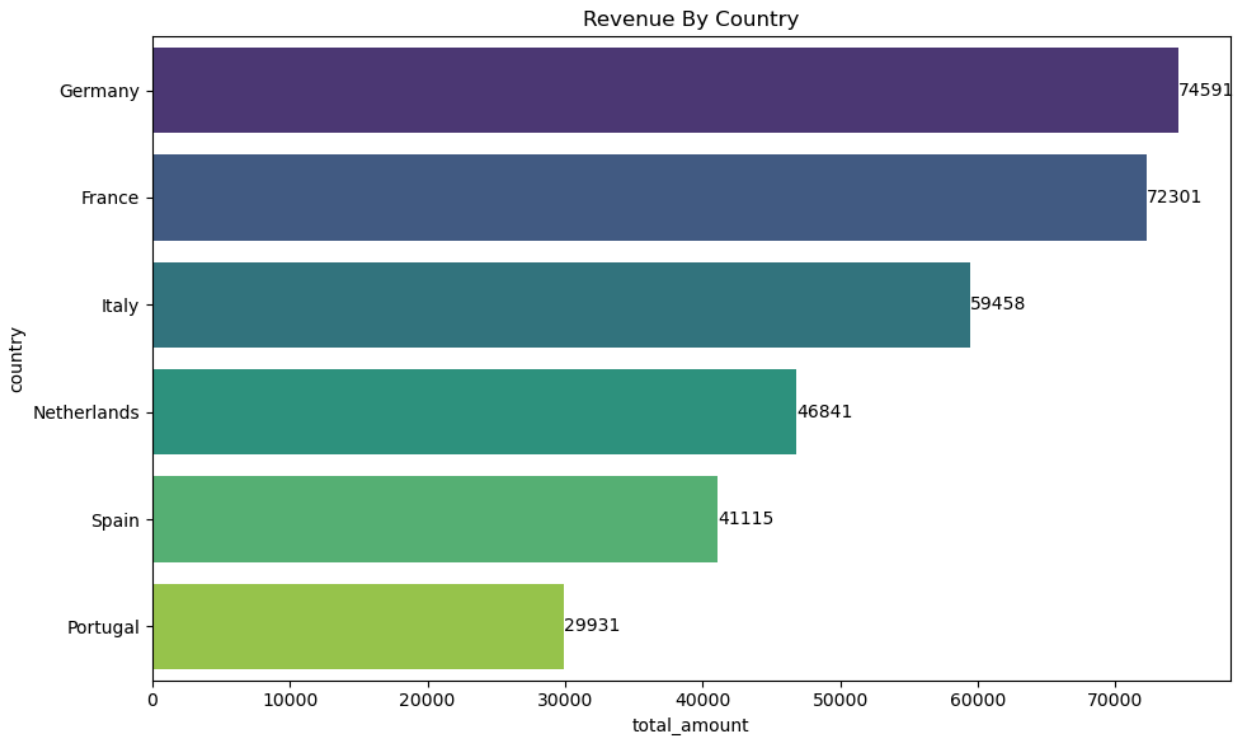
905 rows × 7 columns

In [72]: plt.figure(figsize=(10,6))
c=sns.barplot(data=revenue_by_country,y='country',x='total_amount',hue='country')
c.bar_label(c.containers[0],fmt='%.0f')

```

c.bar_label(c.containers[1],fmt='%.0f')
c.bar_label(c.containers[2],fmt='%.0f')
c.bar_label(c.containers[3],fmt='%.0f')
c.bar_label(c.containers[4],fmt='%.0f')
c.bar_label(c.containers[5],fmt='%.0f')
plt.title('Revenue By Country')
plt.tight_layout()
plt.show()

```



Germany is top country by generating highest revenue after Germany, France is 2nd country by generating highest revenue, Italy and Netherlands give average performance of generating revenue while Portugal is bottom country as compared to other countries.

```

In [74]: customer_by_country = customer_c['country'].value_counts().reset_index()
customer_by_country.columns=['country','customer_count']

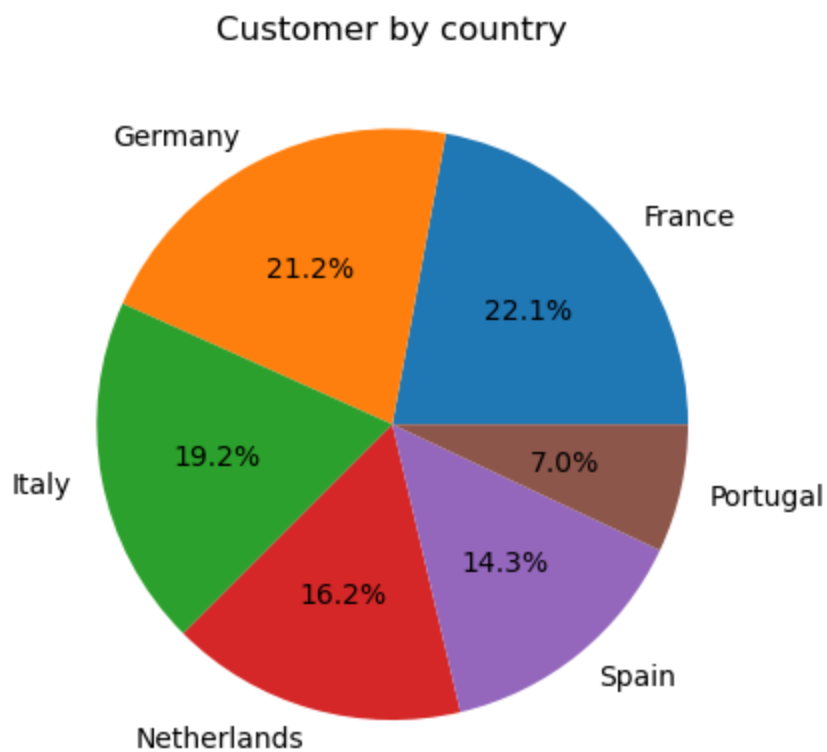
```

```

In [75]: plt.pie(customer_by_country['customer_count'],
                labels=customer_by_country['country'],
                autopct='%1.1f%%')

plt.title('Customer by country')
plt.figure(figsize=(4,4))
plt.show()

```



<Figure size 400x400 with 0 Axes>

most of our customers from Germany and France 21.2% and 22.1% respectively and only 7% of customer are from portugal

Both Revenue and customer are less in portugal compared to other countries

In [78]: customer_by_country

Out[78]:

	country	customer_count
0	France	221
1	Germany	212
2	Italy	192
3	Netherlands	162
4	Spain	143
5	Portugal	70

How sales Performing Over time

In []:

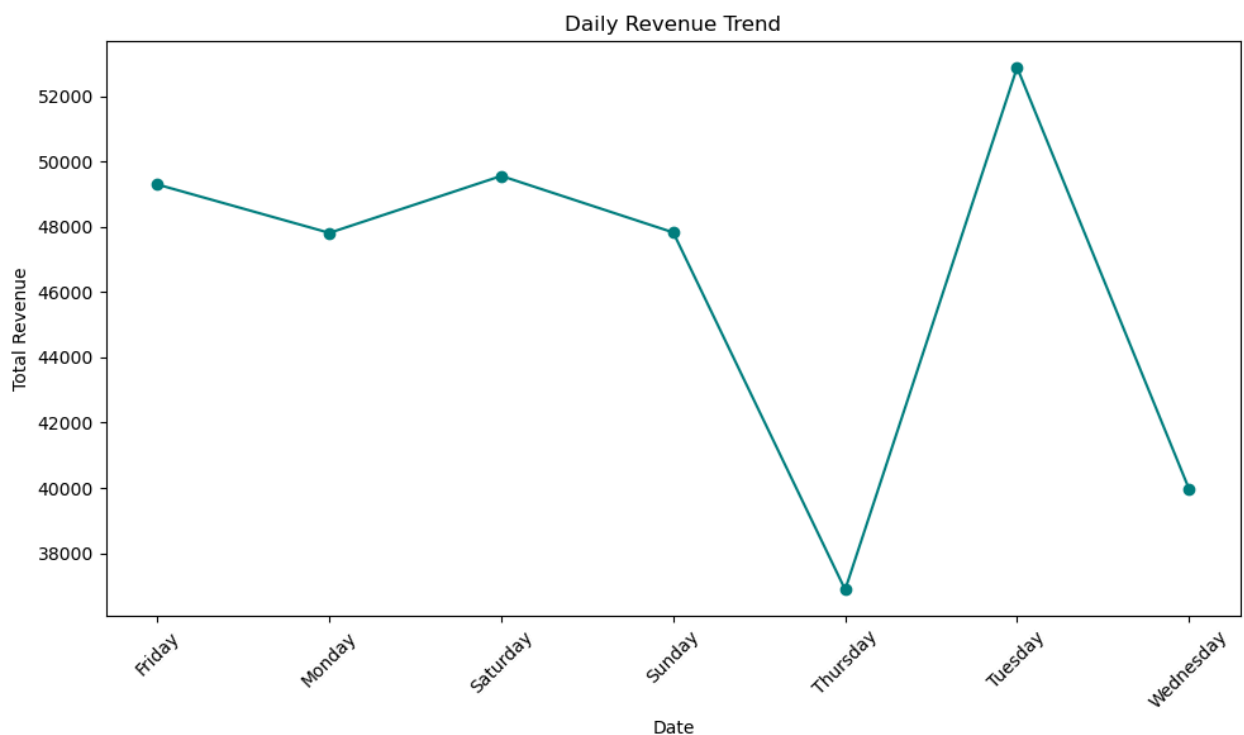
In []:

```
In [80]: sales_c['sale_date'] = pd.to_datetime(sales['sale_date'])
```

```
# Extract just the date part (no time)
sales_c['Day'] = sales_c['sale_date'].dt.day_name()
```

```
In [81]: day_revenue = sales_c.groupby('Day')['total_amount'].sum().reset_index()
```

```
In [82]: plt.figure(figsize=(10, 6))
plt.plot(day_revenue['Day'], day_revenue['total_amount'], marker='o', color='teal')
plt.title("Daily Revenue Trend")
plt.xlabel("Date")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



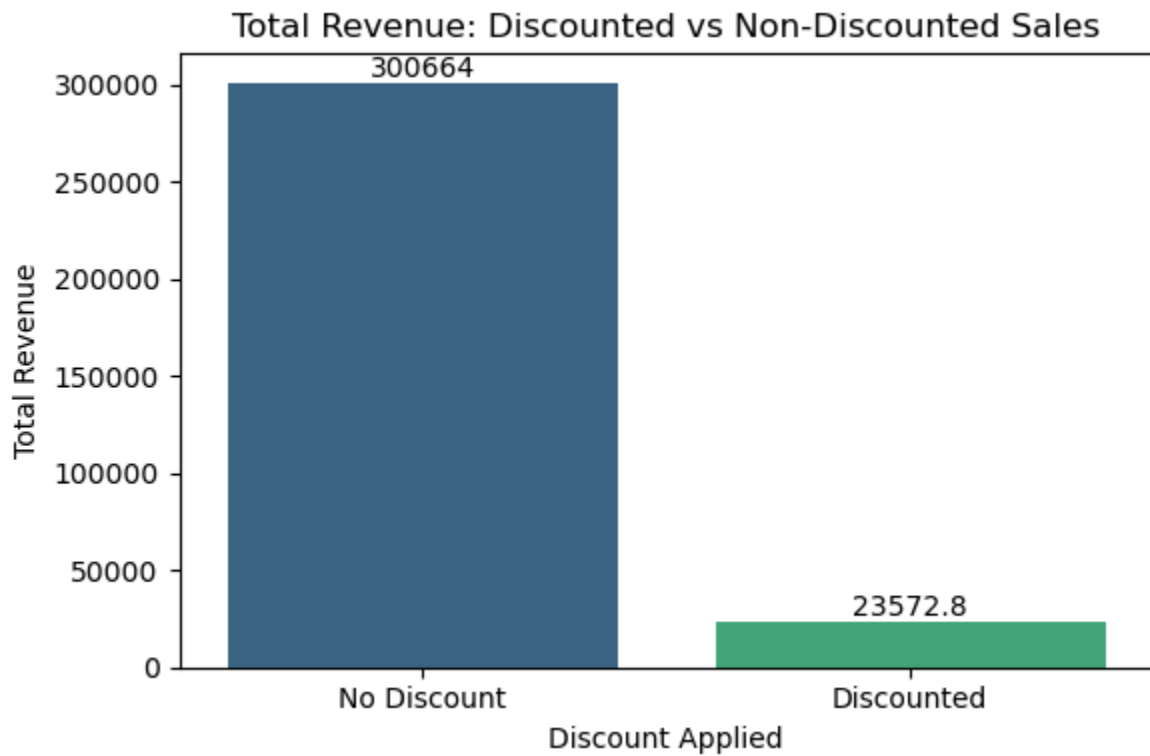
Tuesday is peakday of our sales and wednesday & Thursday our sale is low

Do discount helps increase sales?

```
In [85]: total_by_discount = sales_c.groupby('discounted')['total_amount'].sum().reset_index()
total_by_discount['discounted'] = total_by_discount['discounted'].map({0: 'No', 1: 'Yes'})

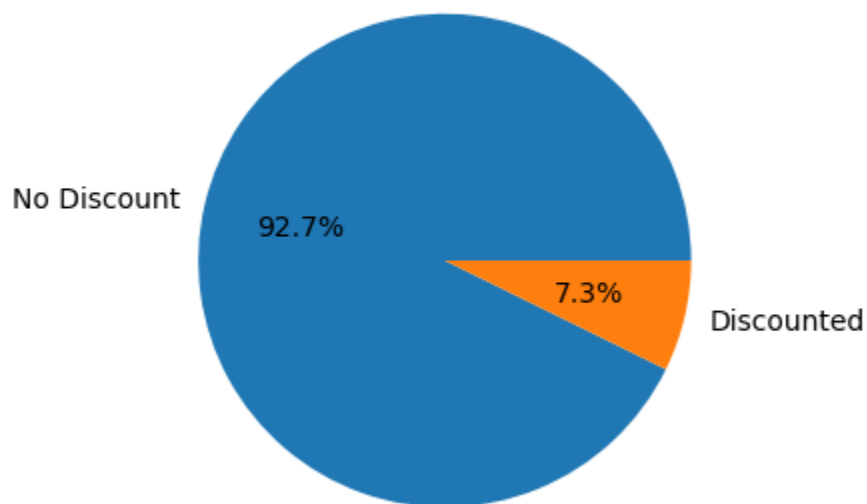
plt.figure(figsize=(6, 4))
ch=sns.barplot(data=total_by_discount, x='discounted', y='total_amount', hue='discounted')
ch.bar_label(ch.containers[0])
ch.bar_label(ch.containers[1])
plt.title("Total Revenue: Discounted vs Non-Discounted Sales")
```

```
plt.ylabel("Total Revenue")
plt.xlabel("Discount Applied")
plt.tight_layout()
plt.show()
```



```
In [86]: plt.figure(figsize=(4,4))
plt.pie(total_by_discount['total_amount'], labels=total_by_discount['discounted'],
plt.title('Share of Total Revenue: With and Without Discounts')
plt.show()
```

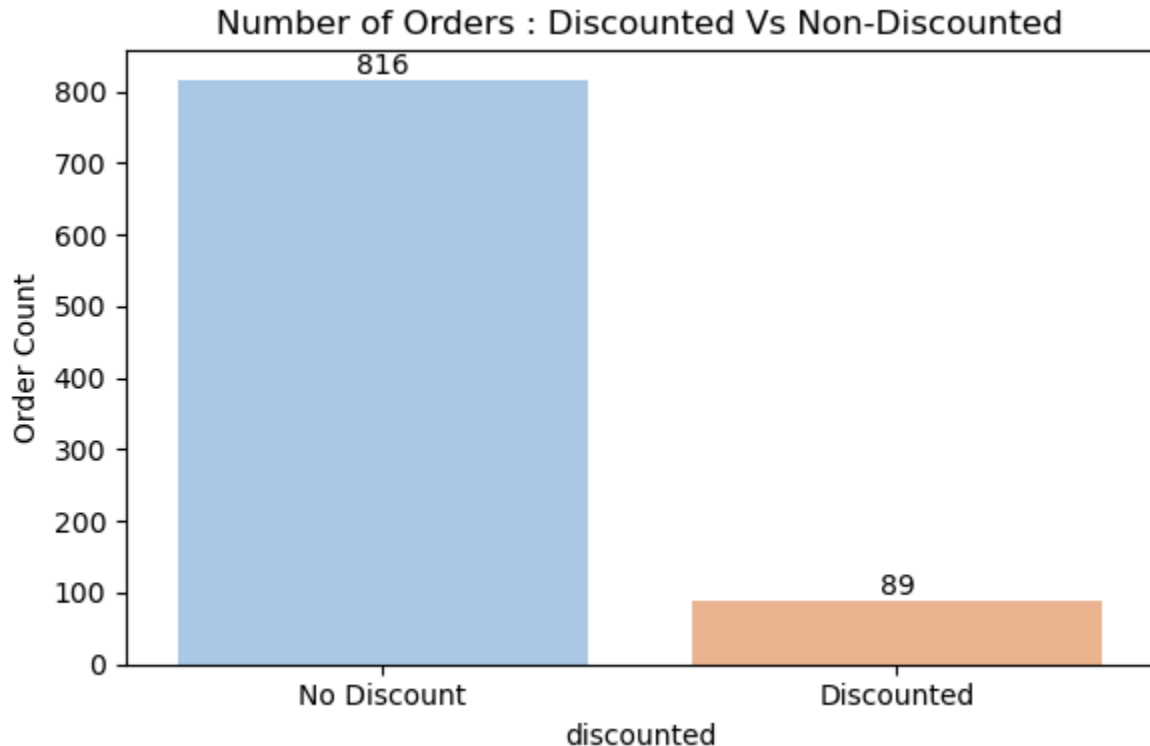
Share of Total Revenue: With and Without Discounts



from above chart we get clear idea about People spent most when we doesn't give any discount only 7.3% (23572) revenue genrated when we give discount and around 92% (300664)revenue genrated when doesn't give any discount

```
In [88]: count_by_discount=sales_c['discounted'].value_counts().reset_index()
count_by_discount.columns=['discounted','count']
plt.figure(figsize=(6,4))
count_by_discount['discounted']=count_by_discount['discounted'].map({0:'No Discount',1:'Discounted'})
k=sns.barplot(data=count_by_discount,x='discounted',y='count',hue='discounted')
k.bar_label(k.containers[0],fmt='%.0f')
k.bar_label(k.containers[1],fmt='%.0f')

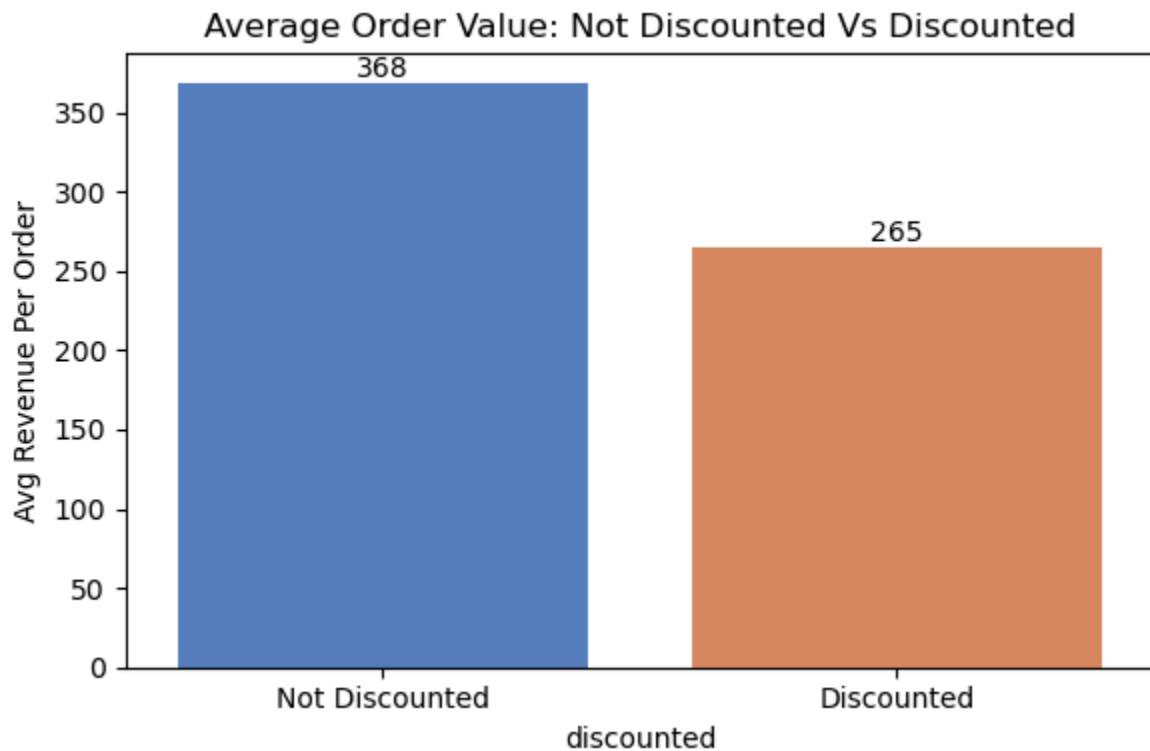
plt.title('Number of Orders : Discounted Vs Non-Discounted')
plt.ylabel('Order Count')
plt.tight_layout()
plt.show()
```



No.of non discount order is high compare to discount orders

```
In [90]: #Avg Order value of discounted vs non-discount
aov= sales_c.groupby('discounted')['total_amount'].mean().reset_index()
aov['discounted']=aov['discounted'].map({0:'Not Discounted',1:'Discounted'})
plt.figure(figsize=(6,4))
sas=sns.barplot(data=aov,x='discounted',y='total_amount',hue='discounted',pale
sas.bar_label(sas.containers[0],fmt='%.0f')
sas.bar_label(sas.containers[1],fmt='%.0f')
plt.title('Average Order Value: Not Discounted Vs Discounted')
```

```
plt.ylabel('Avg Revenue Per Order')
plt.tight_layout()
plt.show()
```



Our avg revenue of not discounted order is better than discounted

conclusion about discount vs no-discount: when we give discount we don't get more orders while on other side without giving discount we get more orders and our sale also good

Which product categories perform the best? Goal: Identify top-performing categories to guide marketing and inventory.

EDA Tasks: Total revenue per category

Top-selling products by count or revenue

Compare revenue vs. cost to find most profitable items

Which product categories perform the best? Goal: Identify top-performing categories to guide marketing and inventory.

```
In [95]: #Total revenue per category
product_c.columns
```

```
Out[95]: Index(['product_id', 'product_name', 'category', 'brand', 'color', 'size',  
              'catalog_price', 'cost_price', 'gender'],  
              dtype='object')
```

```
In [96]: sales_c.columns
```

```
Out[96]: Index(['sale_id', 'channel', 'discounted', 'total_amount', 'sale_date',  
              'customer_id', 'country', 'Day'],  
              dtype='object')
```

5. Which countries bring the most revenue? Goal: Localize marketing and expand high-performing regions.

EDA Tasks: Total sales by country

Sales trends per country

Average order value per country

What's our average order value (AOV) and profit margin? Goal: Evaluate efficiency of sales and product pricing.

EDA Tasks: $AOV = \text{total_amount} / \text{number of orders}$

Calculate profit = catalog_price - cost_price (from products)

Merge sales and products for per-sale profitability

7. Is our stock aligned with customer demand? Goal: Identify understocked or overstocked items.

EDA Tasks: Compare stock_quantity vs. product sales count

Products with high stock but low sales

Countries with stock shortage (low quantity but high orders)

Which sales channels perform best? Goal: Optimize channel marketing and tech investment.

EDA Tasks: Total sales and revenue by channel

Customer acquisition by channel

Discount usage across channels

```
In [101]: total_category=product.groupby('category')['catalog_price'].sum()
```


In []:

```
In [102... product_c['category'].value_counts()
```

```
Out[102... category
Dresses      109
T-Shirts     108
Sleepwear    104
Shoes        100
Pants        79
Name: count, dtype: int64
```

```
In [103... #Average order value per country
avg_od = sales_c.groupby('country')['total_amount'].mean().reset_index()
```

```
In [104... avg_od
```

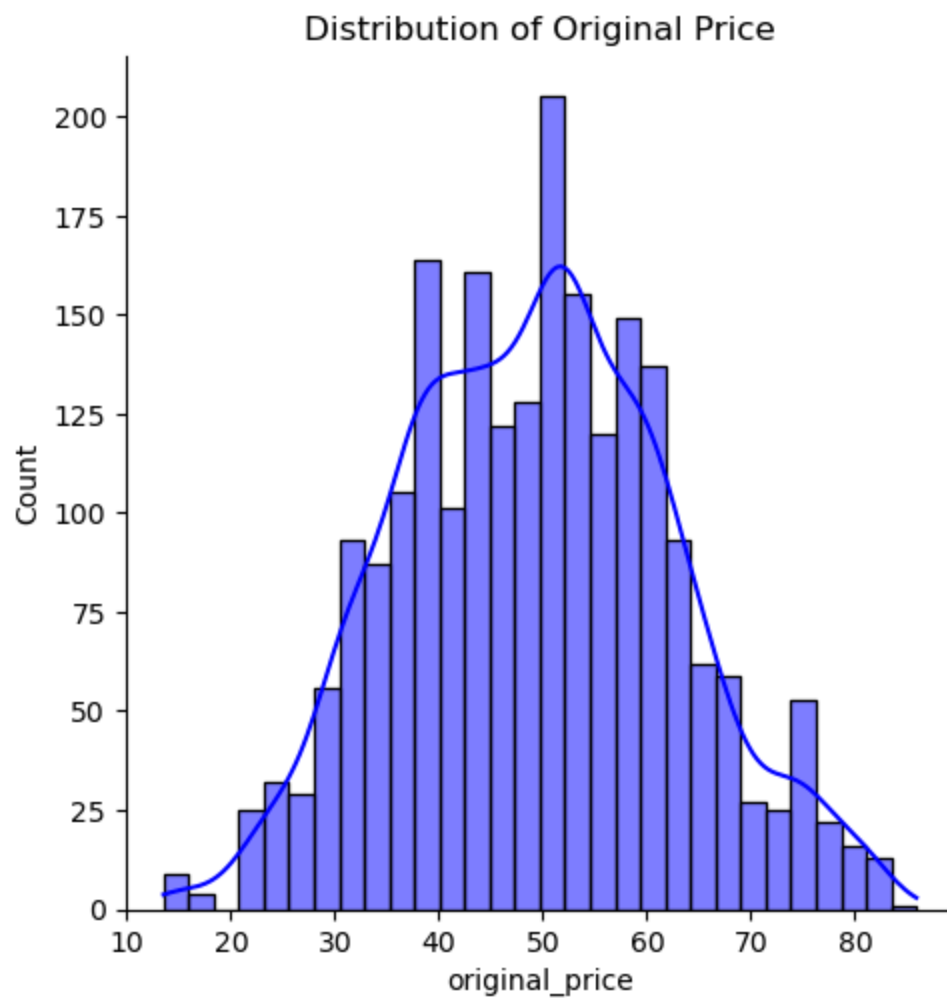
```
Out[104...      country  total_amount
0      France    363.319899
1    Germany    351.842877
2       Italy    349.753588
3 Netherlands    357.568397
4    Portugal    374.136875
5       Spain    363.847699
```

```
In [105... items_c.head(5)
```

```
Out[105...      item_id  sale_id  product_id  quantity  original_price  unit_price  discount_appr
0      2270      658         403          1           81.80         81.80
1      1170      336         284          1           81.79         81.79
2      2496     1255          71          1           80.76         80.76
3      1273      331          98          1           78.52         78.52
4      1829     1079          98          1           78.52         78.52
```

```
In [106... sns.displot(items_c['original_price'],bins=30,color='blue',kde=True)
plt.title('Distribution of Original Price')
plt.figure(figsize=(10,6))

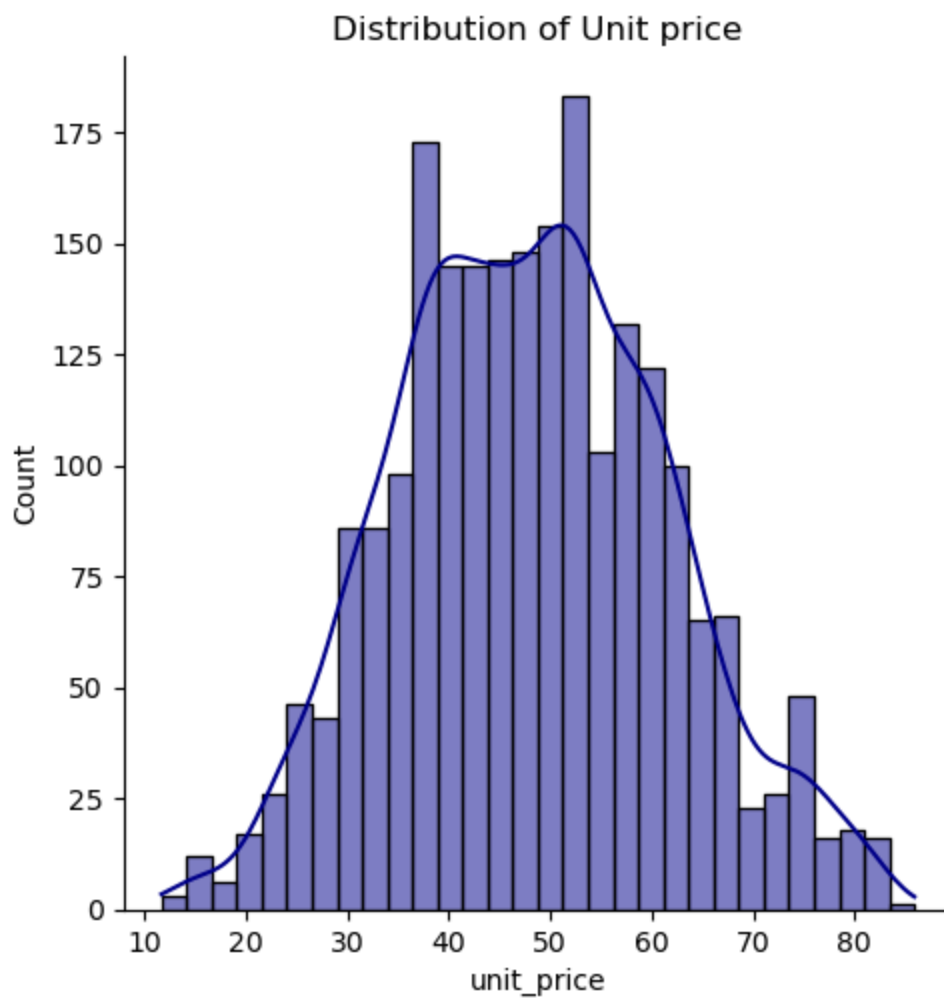
plt.show()
```



<Figure size 1000x600 with 0 Axes>

```
In [107... sns.displot(items_c['unit_price'],bins=30,color='darkblue',kde=True)
plt.title('Distribution of Unit price')

plt.figure(figsize=(10,6))
plt.show()
```



<Figure size 640x480 with 0 Axes>

```
In [108... items_c['channel_campaigns'].value_counts()
```

```
Out[108... channel_campaigns
Website Banner    1151
App Mobile        963
Social Media      120
Email             19
Name: count, dtype: int64
```

```
In [109... product_c
```

Out[109...

	product_id	product_name	category	brand	color	size	catalog_price	co
0	1	Soft Wrap Dress	Dresses	Tiva	Green	S	40.41	
1	2	Soft Wrap Tee	T-Shirts	Tiva	White	S	78.45	
2	3	Soft Linen Tee	T-Shirts	Tiva	Green	XL	23.90	
3	4	Soft Ribbed Tee	T-Shirts	Tiva	White	S	60.00	
4	5	Soft Wrap Trousers	Pants	Tiva	Blue	M	36.84	
...	
495	496	Tailored High-Waist Trousers	Pants	Tiva	Black	S	54.61	
496	497	Dresses Drop 8	Dresses	Tiva	Black	L	36.05	
497	498	T-Shirts Drop 8	T-Shirts	Tiva	White	L	38.33	
498	499	Sleepwear Drop 4	Sleepwear	Tiva	Green	M	30.07	
499	500	Dresses Drop 9	Dresses	Tiva	White	XS	38.03	

500 rows × 9 columns

In [110...

```
items_c
```

Out[110...

	item_id	sale_id	product_id	quantity	original_price	unit_price	discount_
0	2270	658	403	1	81.80	81.80	
1	1170	336	284	1	81.79	81.79	
2	2496	1255	71	1	80.76	80.76	
3	1273	331	98	1	78.52	78.52	
4	1829	1079	98	1	78.52	78.52	
...	
2248	516	1300	334	5	20.82	20.82	
2249	1240	727	334	5	20.82	20.82	
2250	2394	172	246	5	28.98	20.29	
2251	1631	986	195	5	16.69	16.69	
2252	685	724	346	5	15.31	15.31	

2253 rows × 13 columns

In []:

In [111... items_c

Out[111...

	item_id	sale_id	product_id	quantity	original_price	unit_price	discount_
0	2270	658	403	1	81.80	81.80	
1	1170	336	284	1	81.79	81.79	
2	2496	1255	71	1	80.76	80.76	
3	1273	331	98	1	78.52	78.52	
4	1829	1079	98	1	78.52	78.52	
...	
2248	516	1300	334	5	20.82	20.82	
2249	1240	727	334	5	20.82	20.82	
2250	2394	172	246	5	28.98	20.29	
2251	1631	986	195	5	16.69	16.69	
2252	685	724	346	5	15.31	15.31	

2253 rows × 13 columns

In [112...

```
full_sales = sales_c.merge(items_c, on='sale_id', how='left') \
               .merge(product_c, on='product_id', how='left')
```

In [113...

```
full_sales['total_amount'].sum()
```

Out[113...

840339.76

In [114...

```
size_by_revenue=full_sales.groupby('size')['total_amount'].sum().reset_index()
```

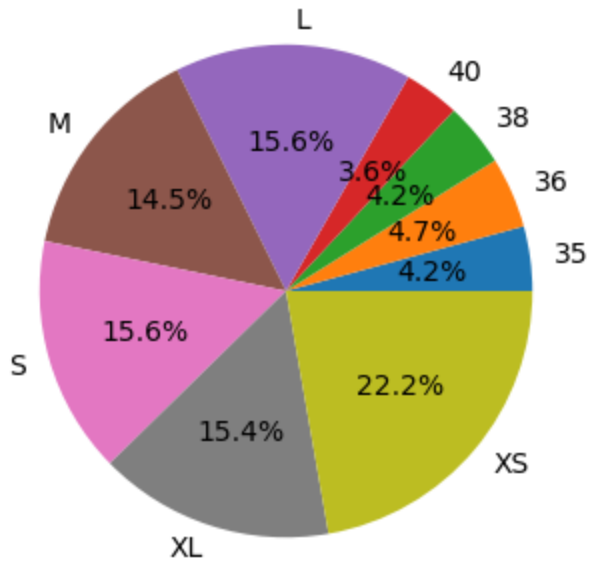
In [115...

```
plt.figure(figsize=(5,4))
plt.pie(size_by_revenue['total_amount'],labels=size_by_revenue['size'],autopct=
plt.title('Size distribution by revenue')
```

Out[115...

Text(0.5, 1.0, 'Size distribution by revenue')

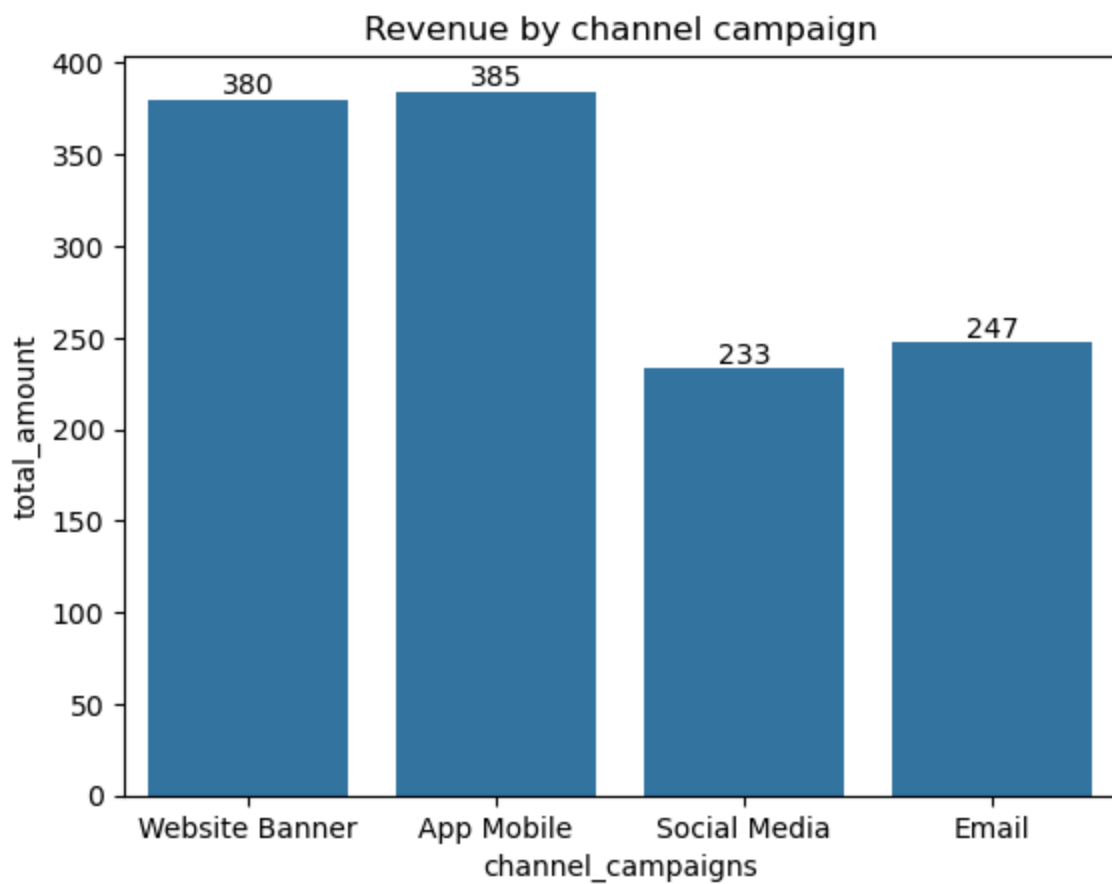
Size distribution by revenue



```
In [117...] full_sales['product_name'].value_counts()
```

```
Out[117...] product_name
Modern Satin Set      11
Polished Boxy Dress  11
Polished Silk Shoes  10
Elegant Satin Set    10
Elegant Cotton Dress 10
..
Soft Crew Shoes      1
Modern High-Waist Dress 1
Vintage Ribbed Shoes  1
Modern Crew Tee       1
Vintage Cotton Trousers 1
Name: count, Length: 499, dtype: int64
```

```
In [118...] p=sns.barplot(data=full_sales,x='channel_campaigns',y='total_amount',errorbar=
p.bar_label(p.containers[0],fmt='%.0f')
plt.title('Revenue by channel campaign')
plt.show()
```



In [119... cmp

	campaign_id	campaign_name	start_date	end_date	channel	discount_type
0	1	Spring Flash Sale	2025-04-01	2025-04-07	Email	Percentage
1	2	Easter Promotion	2025-04-08	2025-04-15	Social Media	Fixed
2	3	Mother's Day Campaign	2025-05-01	2025-05-09	Email	Percentage
3	4	Mid-Season Clearance	2025-05-10	2025-05-19	App Mobile	Percentage
4	5	TIVA Week	2025-05-20	2025-05-31	Social Media	Percentage
5	6	June Price Drop	2025-06-01	2025-06-09	Website Banner	Percentage
6	7	Early Summer Deals	2025-06-10	2025-06-17	Email	Fixed

In [120... product_c

Out[120...

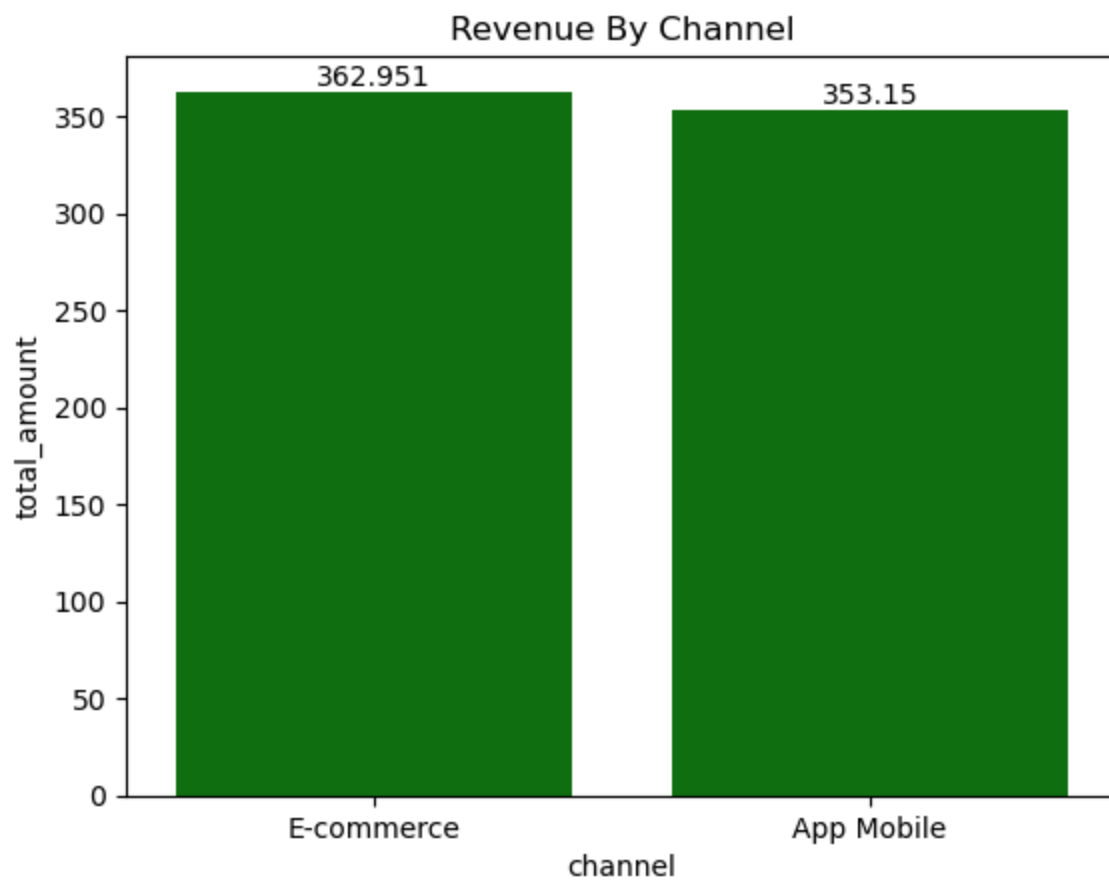
	product_id	product_name	category	brand	color	size	catalog_price	co
0	1	Soft Wrap Dress	Dresses	Tiva	Green	S	40.41	
1	2	Soft Wrap Tee	T-Shirts	Tiva	White	S	78.45	
2	3	Soft Linen Tee	T-Shirts	Tiva	Green	XL	23.90	
3	4	Soft Ribbed Tee	T-Shirts	Tiva	White	S	60.00	
4	5	Soft Wrap Trousers	Pants	Tiva	Blue	M	36.84	
...	
495	496	Tailored High-Waist Trousers	Pants	Tiva	Black	S	54.61	
496	497	Dresses Drop 8	Dresses	Tiva	Black	L	36.05	
497	498	T-Shirts Drop 8	T-Shirts	Tiva	White	L	38.33	
498	499	Sleepwear Drop 4	Sleepwear	Tiva	Green	M	30.07	
499	500	Dresses Drop 9	Dresses	Tiva	White	XS	38.03	

500 rows × 9 columns

In []:

In [121...

```
er=sns.barplot(data=sales_c,x='channel',y='total_amount',color='green',errorba
plt.title('Revenue By Channel')
er.bar_label(er.containers[0])
plt.figure(figsize=(10,6))
plt.show()
```



<Figure size 1000x600 with 0 Axes>

In []:

```
In [122... sales_c['total_amount'].sum()
```

```
Out[122... 324236.660000000003
```

In []:

```
In [124... product_stock= stock_c.merge(product_c,on='product_id',how='inner')
```

```
In [125... product_stock
```

Out[125...

	country	product_id	stock_quantity	product_name	category	brand	color
0	France	1	61	Soft Wrap Dress	Dresses	Tiva	Green
1	France	2	24	Soft Wrap Tee	T-Shirts	Tiva	White
2	France	3	81	Soft Linen Tee	T-Shirts	Tiva	Green
3	France	4	70	Soft Ribbed Tee	T-Shirts	Tiva	White
4	France	5	30	Soft Wrap Trousers	Pants	Tiva	Blue
...
995	Germany	496	1	Tailored High-Waist Trousers	Pants	Tiva	Black
996	Germany	497	1	Dresses Drop 8	Dresses	Tiva	Black
997	Germany	498	1	T-Shirts Drop 8	T-Shirts	Tiva	White
998	Germany	499	1	Sleepwear Drop 4	Sleepwear	Tiva	Green
999	Germany	500	1	Dresses Drop 9	Dresses	Tiva	White

1000 rows × 11 columns

products who has maximum stock

In [127...

```
product_gp = product_stock.groupby('category')['stock_quantity'].sum().reset_index()
product_gp = product_gp.sort_values(by='stock_quantity', ascending=False)
```

In [128...

```
product_stock['product_name'].value_counts()
```

Out[128...

```
product_name
Soft Wrap Dress      2
Essential Wrap Set   2
Tailored Satin Tee   2
Essential Satin Set   2
Essential Cotton Set  2
..
Classic Boxy Set     2
Vintage Linen Tee     2
Classic High-Waist Set 2
Classic High-Waist Trousers 2
Dresses Drop 9        2
Name: count, Length: 500, dtype: int64
```

In [129...

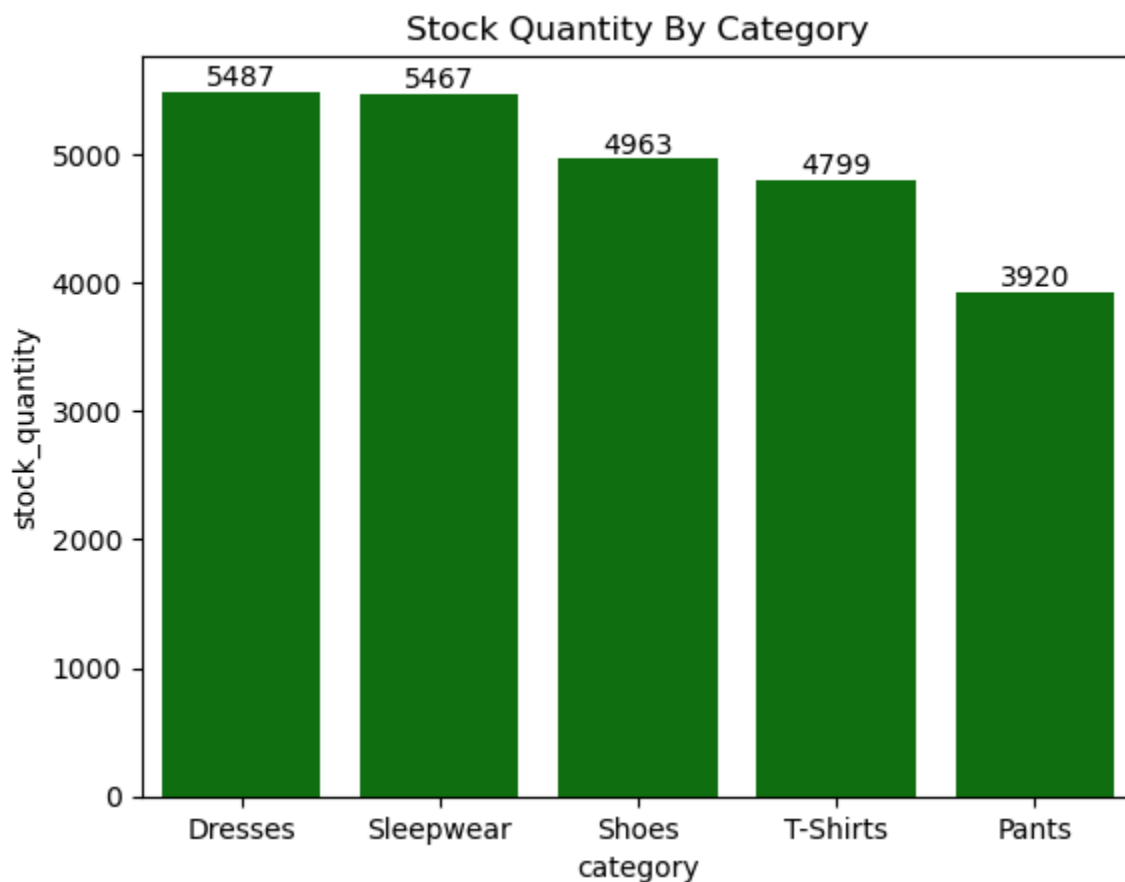
```
product_gp
```

Out[129...

	category	stock_quantity
0	Dresses	5487
3	Sleepwear	5467
2	Shoes	4963
4	T-Shirts	4799
1	Pants	3920

```
In [130... product_gp = product_stock.groupby('category')['stock_quantity'].sum().reset_i
product_gp =product_gp.sort_values(by='stock_quantity',ascending=False)
# Plot
prst = sns.barplot(data=product_gp, x='category', y='stock_quantity', color='g
prst.bar_label(prst.containers[0])

plt.title('Stock Quantity By Category')
plt.show()
```



pants stock is less compare to other categories

```
In [132... productsize= product_stock.groupby('size')['stock_quantity'].sum().reset_index
productsize=productsize.sort_values(by='stock_quantity',ascending=False).reset
```

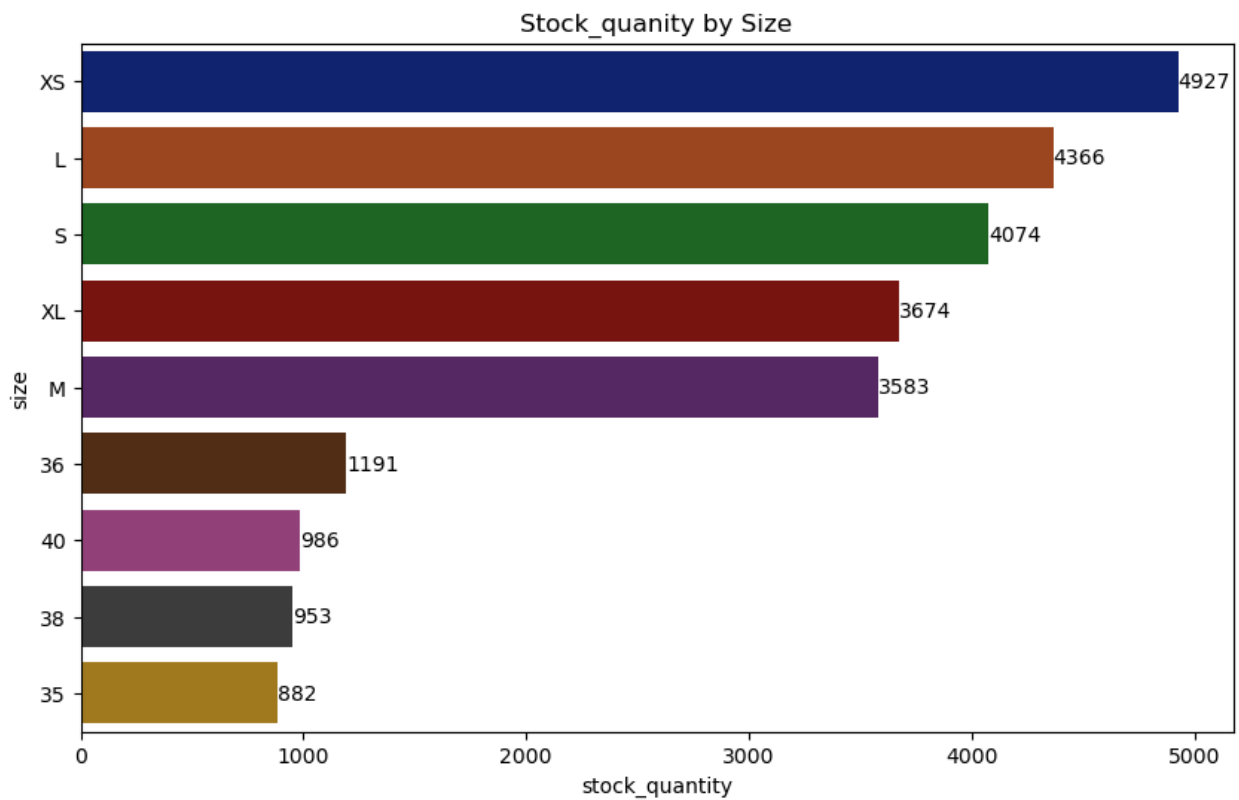
In [133... productsize

Out[133...

	index	size	stock_quantity
0	8	XS	4927
1	4	L	4366
2	6	S	4074
3	7	XL	3674
4	5	M	3583
5	1	36	1191
6	3	40	986
7	2	38	953
8	0	35	882

```
In [134... plt.figure(figsize=(10,6))
sz=sns.barplot(data=productsize,x='stock_quantity',y='size',hue='size',palette
sz.bar_label(sz.containers[0])
sz.bar_label(sz.containers[1])
sz.bar_label(sz.containers[2])
sz.bar_label(sz.containers[3])
sz.bar_label(sz.containers[4])
sz.bar_label(sz.containers[5])
sz.bar_label(sz.containers[6])
sz.bar_label(sz.containers[7])
sz.bar_label(sz.containers[8])

plt.title('Stock_quantity by Size')
plt.show()
```



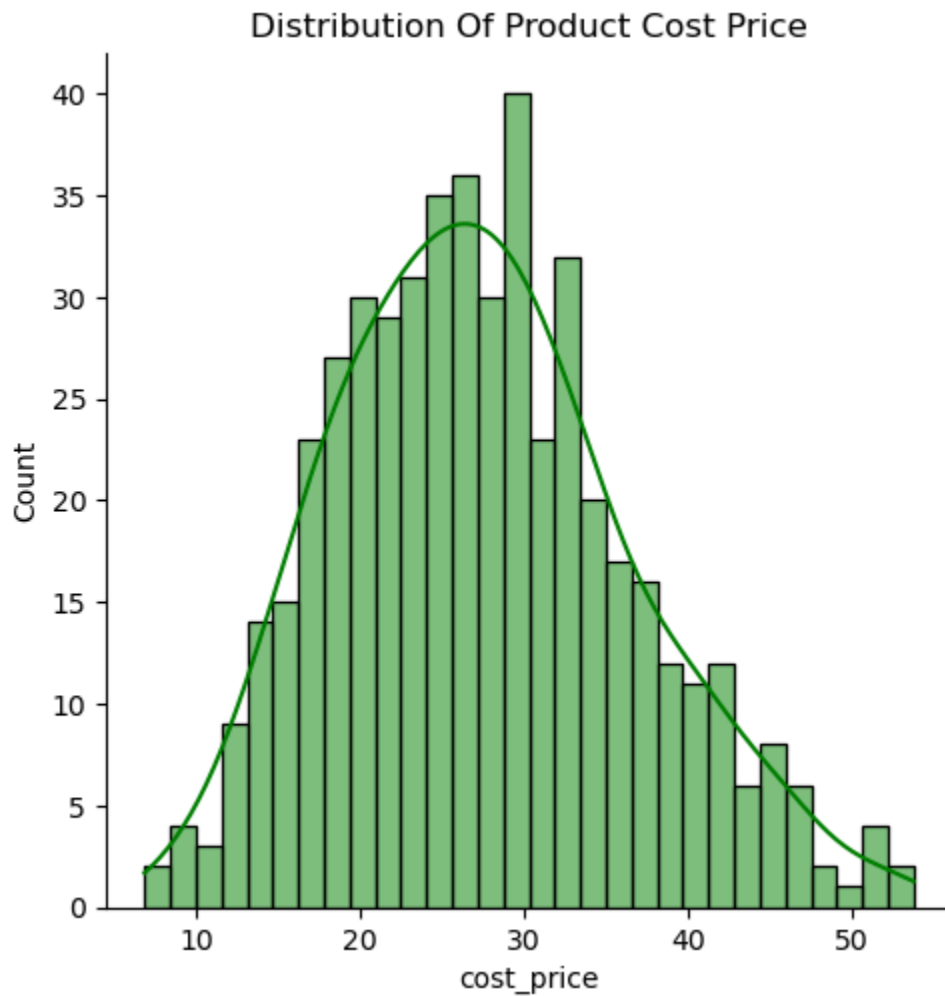
In [135... product_stock

	country	product_id	stock_quantity	product_name	category	brand	color
0	France	1	61	Soft Wrap Dress	Dresses	Tiva	Green
1	France	2	24	Soft Wrap Tee	T-Shirts	Tiva	White
2	France	3	81	Soft Linen Tee	T-Shirts	Tiva	Green
3	France	4	70	Soft Ribbed Tee	T-Shirts	Tiva	White
4	France	5	30	Soft Wrap Trousers	Pants	Tiva	Blue
...
995	Germany	496	1	Tailored High-Waist Trousers	Pants	Tiva	Black
996	Germany	497	1	Dresses Drop 8	Dresses	Tiva	Black
997	Germany	498	1	T-Shirts Drop 8	T-Shirts	Tiva	White
998	Germany	499	1	Sleepwear Drop 4	Sleepwear	Tiva	Green
999	Germany	500	1	Dresses Drop 9	Dresses	Tiva	White

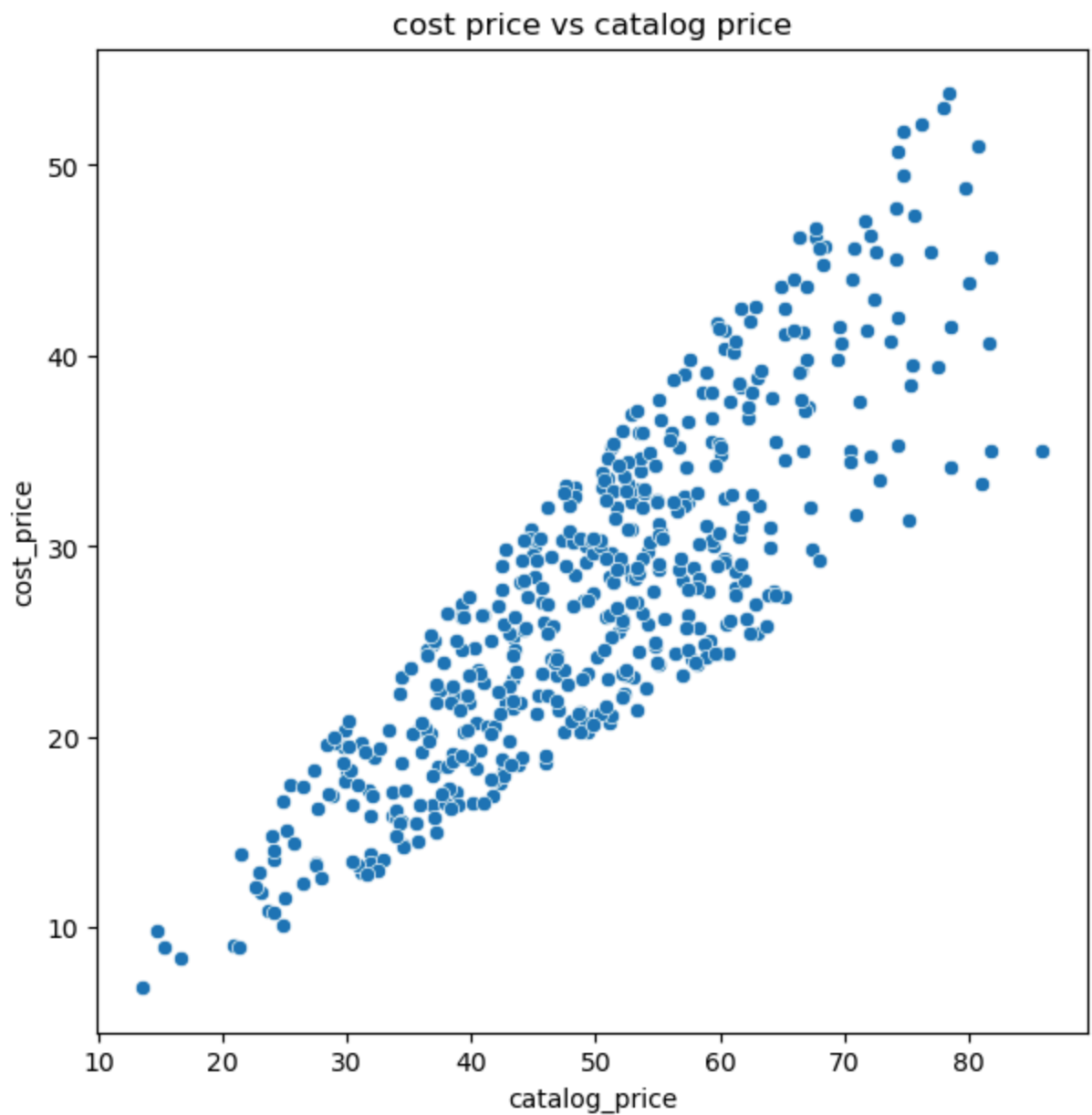
1000 rows × 11 columns

```
In [136... plt.figure(figsize=(10,6))
sns.displot(product_c['cost_price'],bins=30,kde=30,color='green')
plt.title('Distribution Of Product Cost Price')
plt.show()
```

<Figure size 1000x600 with 0 Axes>



```
In [137... plt.figure(figsize=(7,7))
plt.title('cost price vs catalog price')
sns.scatterplot(data=product_c,x='catalog_price',y='cost_price')
plt.show()
```



as we see that cost_price increases catalog_price also increases

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In []:

In [139... `cmp.to_csv('clean_campaign.csv',index=False)`

In [140... `sales_c.to_csv('clean_sales.csv',index=False)`

In [141... `product_c.to_csv('clean_product.csv',index=False)`

In [142... `customer_c.to_csv('clean_customer.csv',index=False)`

In [143... `stock_c.to_csv('clean_stock.csv',index=False)`

In [144... `items_c.to_csv('clean_items.csv',index=False)`

In []:

In []:

In []: