EDA on Sales Export 2019-2020

1. Importing Pandas and Numpy for Implimenting EDA

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns
%matplotlib inline
```

2. Dataset Loading Procedures

In [2]:

```
Sales_Export = pd.read_csv(r"C:\Users\sande\OneDrive\Desktop\Datasets\sales dataset 2023\Sales-Export_2019-2020(new).c
In [3]: Sales_Export
```

Out[3]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id
0	Sweden	17524	14123	02-12-2020	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466
1	Finland	116563	92808	9/26/2019	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323
2	Portugal	296466	257480	07-11-2019	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341
3	Portugal	74532	59752	04-02-2020	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183
4	Spain	178763	146622	12/22/2019	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160
...	...	...	...	...	...	...	...	...	...	...
995	France	46296	40319	5/15/2020	Games	Wisoky Inc	Othello Bowes	Amelina Piscopiello	Tablet	77-3489084
996	Belgium	118061	101131	07-02-2020	Appliances	Johns and Sons	Lambert Norheny	Collin Mackness	Mobile	59-2117058
997	Finland	74481	60024	03-06-2020	Clothing	Homenick-Marvin	Hube Corey	Wat Bowkley	PC	31-1849120
998	Spain	87205	69171	6/18/2020	Games	Johns and Sons	Emalia Dinse	Manuel Goudie	Tablet	45-3085595
999	Portugal	107717	86680	1/18/2020	Accessories	Jacobson, Marvin and Brown	Celine Tumasian	Brynn Dempster	Mobile	61-3294149

1000 rows × 10 columns

In [5]:

```
Sales_Export.head(10)                # Showing first 10 Rows
```

Out[5]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id
0	Sweden	17524.02	14122.61	2020-12-02 00:00:00	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466
1	Finland	116563.40	92807.78	9/26/2019	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323
2	Portugal	296465.56	257480.34	2019-11-07 00:00:00	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341
3	Portugal	74532.02	59752.32	2020-02-04 00:00:00	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183
4	Spain	178763.42	146621.76	12/22/2019	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160
5	Spain	84900.24	73701.90	7/14/2020	Clothing	Farrell, Swaniawski and Crist	Emalia Dinse	Perri Aldersley	PC	60-6998932
6	Portugal	71620.08	62245.01	2019-05-02 00:00:00	Books	Schoen-Keeling	Celine Tumasian	Smitty Culverhouse	PC	69-6259390
7	UK	156585.22	126599.15	8/30/2020	Accessories	Hermiston, Simonis and Wisoky	Jessamine Apark	Winny Agnolo	PC	64-5761908
8	Portugal	78461.13	63537.82	2020-10-05 00:00:00	Appliances	Hessel-Stiedemann	Celine Tumasian	Smitty Culverhouse	Mobile	91-4126746
9	France	64827.80	56043.63	1/20/2019	Appliances	Gislason-Stanton	Othello Bowes	Maighdiln Upcraft	PC	62-3312495

2.1 Understanding the Dataset

In [52]:

```
Sales_Export.shape                 # There are 1000 number of Rows and 10 Columns Present in the Dataset
```

Out[52]:

```
(1000, 10)
```

In [82]:

```
Sales_Export.columns
```

Out[82]:

```
Index(['country', 'order_value_EUR', 'cost', 'date', 'category',
       'customer_name', 'sales_manager', 'sales_rep', 'device_type',
       'order_id'],
      dtype='object')
```

In [4]:

```
Sales_Export['date'] = pd.to_datetime(Sales_Export['date'])
```

In [5]:

```
Sales_Export['day'] = Sales_Export['date'].dt.day
Sales_Export['month'] = Sales_Export['date'].dt.month
Sales_Export['year'] = Sales_Export['date'].dt.year
```

In [6]:
```
Sales_Export
```

Out[6]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	day	month	yea
0	Sweden	17524	14123	2020-02-12	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466	12	2	202
1	Finland	116563	92808	2019-09-26	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323	26	9	201
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	11	7	201
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	2	4	202
4	Spain	178763	146622	2019-12-22	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160	22	12	201
...	...	...	...	...	...	...	...	...	...	...	...	...	..
995	France	46296	40319	2020-05-15	Games	Wisoky Inc	Othello Bowes	Amelina Piscopiello	Tablet	77-3489084	15	5	202
996	Belgium	118061	101131	2020-07-02	Appliances	Johns and Sons	Lambert Norheny	Collin Mackness	Mobile	59-2117058	2	7	202
997	Finland	74481	60024	2020-03-06	Clothing	Homenick-Marvin	Hube Corey	Wat Bowkley	PC	31-1849120	6	3	202
998	Spain	87205	69171	2020-06-18	Games	Johns and Sons	Emalia Dinse	Manuel Goudie	Tablet	45-3085595	18	6	202
999	Portugal	107717	86680	2020-01-18	Accessories	Jacobson, Marvin and Brown	Celine Tumasian	Brynn Dempster	Mobile	61-3294149	18	1	202

1000 rows × 13 columns

In [7]:
```
Sales_Export.drop('day', inplace=True, axis=1)
```

In [8]:

```
Sales_Export
```

Out[8]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
0	Sweden	17524	14123	2020-02-12	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466	2	2020
1	Finland	116563	92808	2019-09-26	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323	9	2019
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	7	2019
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	4	2020
4	Spain	178763	146622	2019-12-22	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160	12	2019
...	...	...	...	...	...	...	...	...	...	...	...	...
995	France	46296	40319	2020-05-15	Games	Wisoky Inc	Othello Bowes	Amelina Piscopiello	Tablet	77-3489084	5	2020
996	Belgium	118061	101131	2020-07-02	Appliances	Johns and Sons	Lambert Norheny	Collin Mackness	Mobile	59-2117058	7	2020
997	Finland	74481	60024	2020-03-06	Clothing	Homenick-Marvin	Hube Corey	Wat Bowkley	PC	31-1849120	3	2020
998	Spain	87205	69171	2020-06-18	Games	Johns and Sons	Emalia Dinse	Manuel Goudie	Tablet	45-3085595	6	2020
999	Portugal	107717	86680	2020-01-18	Accessories	Jacobson, Marvin and Brown	Celine Tumasian	Brynn Dempster	Mobile	61-3294149	1	2020

1000 rows × 12 columns

In [9]:

```
import calendar
```

In [10]:

```
Sales_Export['month'] = Sales_Export['month'].apply(lambda x: calendar.month_abbr[x])
```

In [11]:
```
Sales_Export
```

Out[11]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
0	Sweden	17524	14123	2020-02-12	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466	Feb	2020
1	Finland	116563	92808	2019-09-26	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323	Sep	2019
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	Jul	2019
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	Apr	2020
4	Spain	178763	146622	2019-12-22	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160	Dec	2019
...	...	...	...	...	...	...	...	...	...	...	...	...
995	France	46296	40319	2020-05-15	Games	Wisoky Inc	Othello Bowes	Amelina Piscopiello	Tablet	77-3489084	May	2020
996	Belgium	118061	101131	2020-07-02	Appliances	Johns and Sons	Lambert Norheny	Collin Mackness	Mobile	59-2117058	Jul	2020
997	Finland	74481	60024	2020-03-06	Clothing	Homenick-Marvin	Hube Corey	Wat Bowkley	PC	31-1849120	Mar	2020
998	Spain	87205	69171	2020-06-18	Games	Johns and Sons	Emalia Dinse	Manuel Goudie	Tablet	45-3085595	Jun	2020
999	Portugal	107717	86680	2020-01-18	Accessories	Jacobson, Marvin and Brown	Celine Tumasian	Brynn Dempster	Mobile	61-3294149	Jan	2020

1000 rows × 12 columns

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

Out[21]:

	order_value_EUR	cost	year
count	1000.000000	1000.000000	1000.00000
mean	113361.747000	94369.313000	2019.51000
std	61775.319881	51540.004501	0.50015
min	15101.000000	12114.000000	2019.00000
25%	65311.250000	54248.000000	2019.00000
50%	105419.000000	87094.500000	2020.00000
75%	151192.750000	125570.750000	2020.00000
max	383997.000000	304701.000000	2020.00000

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype         
---  ------           --------------  -----         
 0   country          1000 non-null   object        
 1   order_value_EUR  1000 non-null   int64         
 2   cost             1000 non-null   int64         
 3   date             1000 non-null   datetime64[ns]
 4   category         1000 non-null   object        
 5   customer_name    1000 non-null   object        
 6   sales_manager    1000 non-null   object        
 7   sales_rep        1000 non-null   object        
 8   device_type      1000 non-null   object        
 9   order_id         1000 non-null   object        
 10  month            1000 non-null   object        
 11  year             1000 non-null   int64         
dtypes: datetime64[ns](1), int64(3), object(8)
memory usage: 93.9+ KB
```

In [23]:
```
Sales_Export.isnull().sum()
```

Out[23]:
```
country            0
order_value_EUR    0
cost               0
date               0
category           0
customer_name      0
sales_manager      0
sales_rep          0
device_type        0
order_id           0
month              0
year               0
dtype: int64
```

In [24]:
```
Sales_Export.dtypes
```

Out[24]:
```
country                    object
order_value_EUR             int64
cost                        int64
date               datetime64[ns]
category                   object
customer_name              object
sales_manager              object
sales_rep                  object
device_type                object
order_id                   object
month                      object
year                        int64
dtype: object
```

3. Possible Insights that Can be Extracted From Dataset

3.1 Number of Countries are paid for Beauty Category.

In [25]:
```
Sales_Export[['country','category']]
```

Out[25]:

	country	category
0	Sweden	Books
1	Finland	Games
2	Portugal	Clothing
3	Portugal	Beauty
4	Spain	Games
...	...	...
995	France	Games
996	Belgium	Appliances
997	Finland	Clothing
998	Spain	Games
999	Portugal	Accessories

1000 rows × 2 columns

In [26]:
```
group = Sales_Export.groupby('category')['country']
```

In [27]:
```
group.head(2)
```

Out[27]:
```
0       Sweden
1      Finland
2     Portugal
3     Portugal
4        Spain
5        Spain
6     Portugal
7           UK
8     Portugal
9       France
11          UK
12    Portugal
13          UK
14          UK
16      France
20       Spain
26    Portugal
37          UK
46       Italy
51      France
Name: country, dtype: object
```

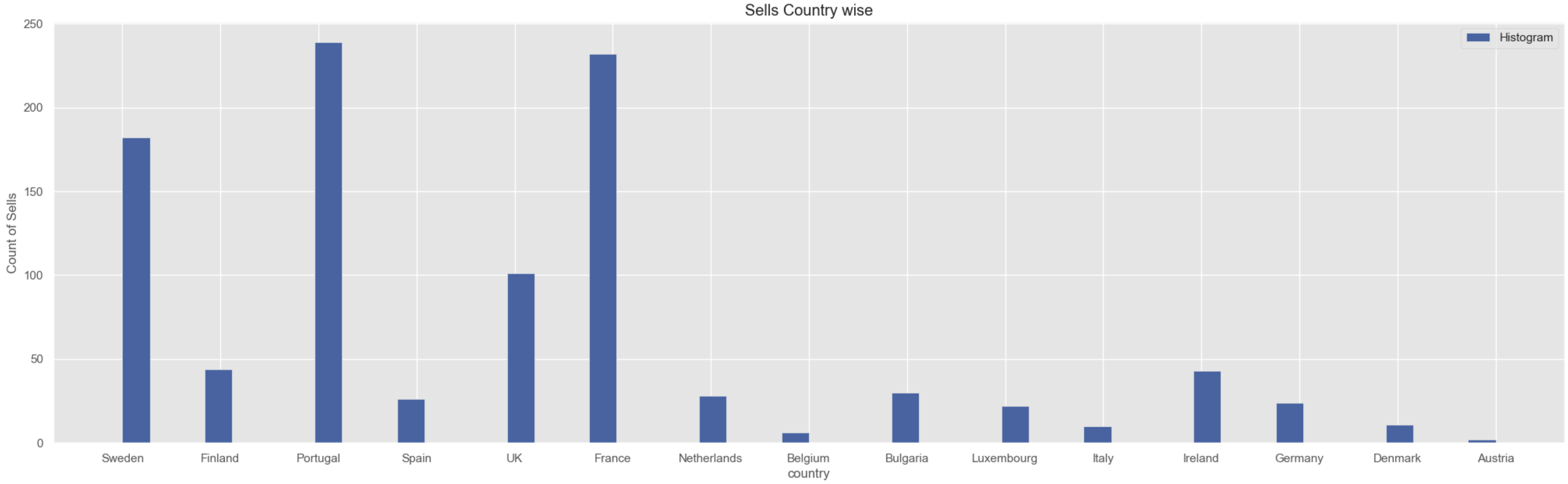
In [28]:
```
country_group = group.count().head(10)
```

In [29]:
```
country_group
```

Out[29]:
```
category
Accessories     35
Appliances     131
Beauty         115
Books          117
Clothing       155
Electronics    134
Games          139
Other           29
Outdoors        50
Smartphones     95
Name: country, dtype: int64
```

In [146]:
```
x_axis = Sales_Export[['country']]
```

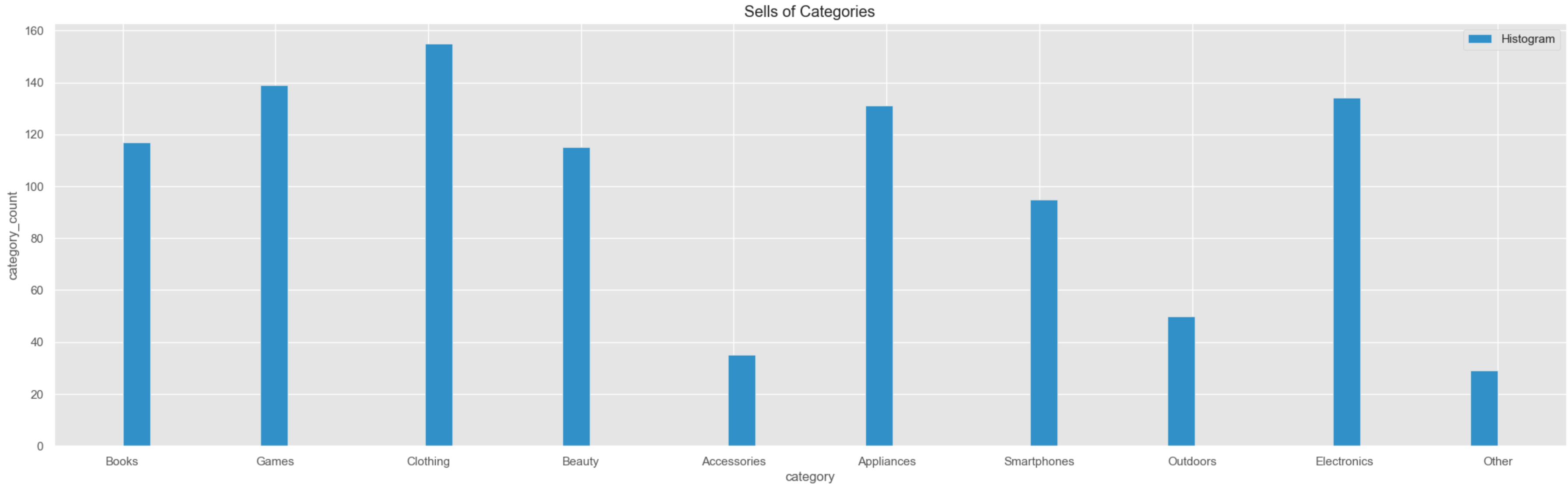
In [147]:
```
style.use('ggplot')

plt.figure(figsize = (25,7))

plt.hist(x_axis,bins = 50,histtype='stepfilled',align='mid',orientation='vertical' , color = '#4863A0' ,label='Histogr
plt.xlabel('country')
plt.ylabel('Count of Sells')
plt.title('Sells Country wise')
plt.legend()
plt.show()
```



In [142]:
```
y_axis = Sales_Export[['category']]
```

In [148]:

```
plt.figure(figsize = (25,7))
plt.hist(y_axis,bins = 50,histtype='stepfilled',align='mid', color = '#3090C7',orientation='vertical' ,label='Histogra
plt.xlabel('category')
plt.ylabel('category_count')
plt.title('Sells of Categories')
plt.legend()
plt.show()
```



In [98]:

```
Country_catagories = Sales_Export[Sales_Export['category'] == 'Beauty'].count().head(1)
```

In [31]:

```
Country_catagories
```

Out[31]:

```
country    115
dtype: int64
```

3.2 The Best known Sales manager in the sales field.

In [32]:
```
Sales_Export.head(5)
```

Out[32]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
0	Sweden	17524	14123	2020-02-12	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466	Feb	2020
1	Finland	116563	92808	2019-09-26	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323	Sep	2019
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	Jul	2019
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	Apr	2020
4	Spain	178763	146622	2019-12-22	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160	Dec	2019

In [33]:
```
Sales_Export[['category','sales_manager']].head(5)
```

Out[33]:

	category	sales_manager
0	Books	Maxie Marrow
1	Games	Hube Corey
2	Clothing	Celine Tumasian
3	Beauty	Celine Tumasian
4	Games	Emalia Dinse

In [19]:
```
best_manager = Sales_Export.groupby('sales_manager')['category']
```

In [44]:
```
List_maneger = Sales_Export[['sales_manager']]
```

In [45]:
```
List_maneger
```

Out[45]:

	sales_manager
0	Maxie Marrow
1	Hube Corey
2	Celine Tumasian
3	Celine Tumasian
4	Emalia Dinse
...	...
995	Othello Bowes
996	Lambert Norheny
997	Hube Corey
998	Emalia Dinse
999	Celine Tumasian

1000 rows × 1 columns

In [35]:
```
best_manager.count().head(5)
```

Out[35]:
```
sales_manager
Celine Tumasian    239
Charil Alpe         30
Denice Amberg       28
Emalia Dinse        26
Glenine Suttaby     43
Name: category, dtype: int64
```

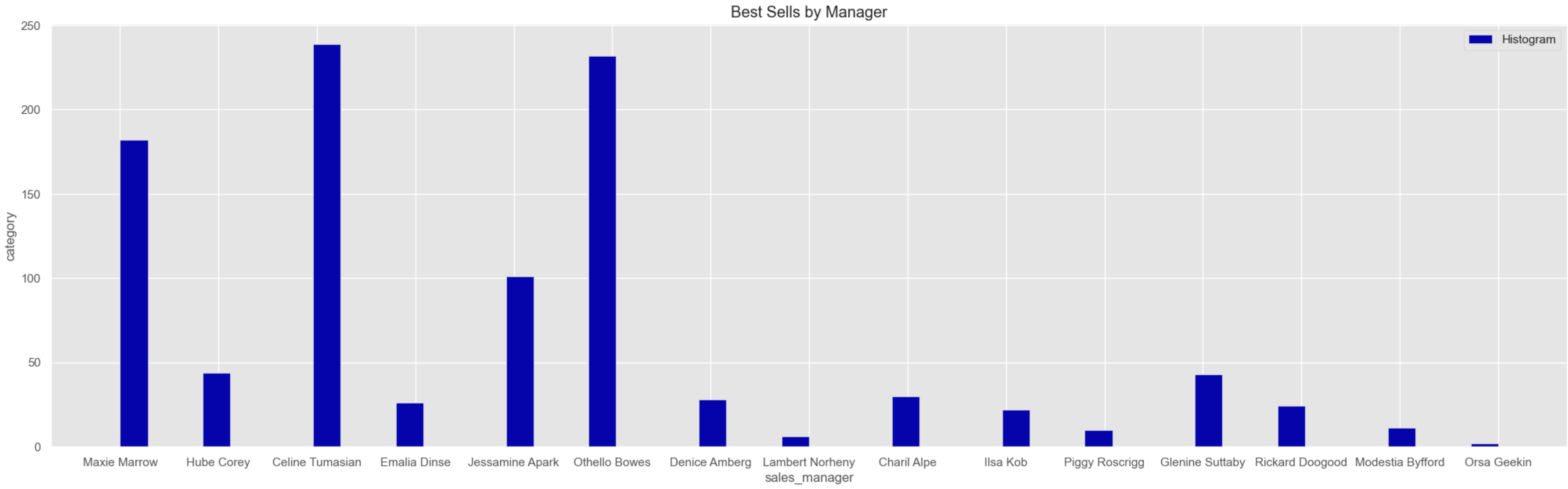
In [36]:
```
best_manager.count().sort_values(ascending = False).head(1)
```

Out[36]:
```
sales_manager
Celine Tumasian    239
Name: category, dtype: int64
```

In [150]:
```
y_axis = best_manager.count().head(5)
```

In [151]:
```
x_axis = List_maneger
```

In [152]:
```
style.use('ggplot')

plt.figure(figsize = (25,7))
plt.hist(x_axis,bins = 50,histtype='stepfilled',align='mid',orientation='vertical',color='#0504aa',label='Histogram',s
plt.xlabel('sales_manager')
plt.ylabel('category')
plt.title('Best Sells by Manager')
plt.legend()
plt.show()
```



3.3 The order_id creates a highest cost in the saling market.

In [37]:

```
Sales_Export.head(5)
```

Out[37]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
0	Sweden	17524	14123	2020-02-12	Books	Goldner-Dibbert	Maxie Marrow	Madelon Bront	Mobile	70-0511466	Feb	2020
1	Finland	116563	92808	2019-09-26	Games	Hilll-Vandervort	Hube Corey	Wat Bowkley	Mobile	28-6585323	Sep	2019
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	Jul	2019
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	Apr	2020
4	Spain	178763	146622	2019-12-22	Games	Johns and Sons	Emalia Dinse	Bertha Walbrook	Tablet	15-8765160	Dec	2019

In [38]:
```
Sales_Export[['order_id' , 'cost']]
```

Out[38]:

	order_id	cost
0	70-0511466	14123
1	28-6585323	92808
2	58-7703341	257480
3	14-6700183	59752
4	15-8765160	146622
...	...	...
995	77-3489084	40319
996	59-2117058	101131
997	31-1849120	60024
998	45-3085595	69171
999	61-3294149	86680

1000 rows × 2 columns

In [39]:
```
Max_cost = Sales_Export['cost'].max()
```

In [40]:
```
Max_cost
```

Out[40]:
```
304701
```

In [41]:
```
Max_values = Sales_Export.max()
```

```
In [42]: Max_values
```

```
Out[42]: country                                   UK
         order_value_EUR                       383997
         cost                                  304701
         date                     2020-12-30 00:00:00
         category                         Smartphones
         customer_name      Zieme, Bailey and Herzog
         sales_manager               Rickard Doogood
         sales_rep                      Winny Agnolo
         device_type                          Tablet
         order_id                         99-9599677
         month                                   Sep
         year                                   2020
         dtype: object
```

```
In [43]: Max_values[['order_id' , 'cost']]
```

```
Out[43]: order_id    99-9599677
         cost            304701
         dtype: object
```

3.4 The Most Tablet Selling Country.

In [44]:
```
Sales_Export[['country' , 'device_type']]
```

Out[44]:
```
       country  device_type
  0     Sweden       Mobile
  1    Finland       Mobile
  2   Portugal           PC
  3   Portugal           PC
  4      Spain       Tablet
...        ...          ...
995     France       Tablet
996    Belgium       Mobile
997    Finland           PC
998      Spain       Tablet
999   Portugal       Mobile

1000 rows × 2 columns
```

In [45]:
```
Tablet_Selling = Sales_Export.groupby('country')['device_type']
```

In [46]:
```
Tablet_Selling.head(5)
```

Out[46]:
```
0      Mobile
1      Mobile
2          PC
3          PC
4      Tablet
        ...  
523        PC
537        PC
625        PC
910        PC
942    Mobile
Name: device_type, Length: 72, dtype: object
```

In [47]:
```
Tablet_Selling.count().sort_values(ascending = False).head(1)
```

Out[47]:
```
country
Portugal    239
Name: device_type, dtype: int64
```

3.5 Category of Products Sold by 'Jessamine Apank'

In [48]:
```
Sales_Export[['category' , 'sales_manager']]
```

Out[48]:

	category	sales_manager
0	Books	Maxie Marrow
1	Games	Hube Corey
2	Clothing	Celine Tumasian
3	Beauty	Celine Tumasian
4	Games	Emalia Dinse
...	...	...
995	Games	Othello Bowes
996	Appliances	Lambert Norheny
997	Clothing	Hube Corey
998	Games	Emalia Dinse
999	Accessories	Celine Tumasian

1000 rows × 2 columns

In [49]:
```
Sold_Categories = Sales_Export[Sales_Export['sales_manager'] == 'Jessamine Apark'].head(5)
```

In [50]:
```
Sold_Categories
```

Out[50]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
7	UK	156585	126599	2020-08-30	Accessories	Hermiston, Simonis and Wisoky	Jessamine Apark	Winny Agnolo	PC	64-5761908	Aug	2020
11	UK	66673	52812	2019-02-23	Smartphones	Gislason-Stanton	Jessamine Apark	Winny Agnolo	PC	25-6368157	Feb	2019
13	UK	164972	132687	2019-07-20	Accessories	Wisoky Inc	Jessamine Apark	Winny Agnolo	PC	32-3534634	Jul	2019
14	UK	149486	118662	2019-08-06	Beauty	Johns and Sons	Jessamine Apark	Genevra Charrisson	PC	02-3972649	Aug	2019
17	UK	29494	24285	2019-11-22	Games	Tillman and Sons	Jessamine Apark	Jay Morefield	PC	53-7769693	Nov	2019

In [51]:
```
Sold_Categories[['sales_manager' , 'category']]
```

Out[51]:

	sales_manager	category
7	Jessamine Apark	Accessories
11	Jessamine Apark	Smartphones
13	Jessamine Apark	Accessories
14	Jessamine Apark	Beauty
17	Jessamine Apark	Games

3.6 Number of Counties has sold PC Devices Coming Under Appliances Category

In [52]:

```
Sales_Export[['country' , 'category' , 'device_type']]
```

Out[52]:

```
        country    category  device_type
  0      Sweden       Books       Mobile
  1     Finland       Games       Mobile
  2    Portugal    Clothing           PC
  3    Portugal      Beauty           PC
  4       Spain       Games       Tablet
...         ...         ...          ...
995      France       Games       Tablet
996     Belgium  Appliances       Mobile
997     Finland    Clothing           PC
998       Spain       Games       Tablet
999    Portugal Accessories       Mobile

1000 rows × 3 columns
```

In [53]:

```
Appliances_Category = Sales_Export[Sales_Export['device_type'] == 'PC'].head()
```

In [54]:
```
Appliances_Category.head()
```

Out[54]:

	country	order_value_EUR	cost	date	category	customer_name	sales_manager	sales_rep	device_type	order_id	month	year
2	Portugal	296466	257480	2019-07-11	Clothing	Larkin-Collier	Celine Tumasian	Smitty Culverhouse	PC	58-7703341	Jul	2019
3	Portugal	74532	59752	2020-04-02	Beauty	Hessel-Stiedemann	Celine Tumasian	Aurelie Wren	PC	14-6700183	Apr	2020
5	Spain	84900	73702	2020-07-14	Clothing	Farrell, Swaniawski and Crist	Emalia Dinse	Perri Aldersley	PC	60-6998932	Jul	2020
6	Portugal	71620	62245	2019-02-05	Books	Schoen-Keeling	Celine Tumasian	Smitty Culverhouse	PC	69-6259390	Feb	2019
7	UK	156585	126599	2020-08-30	Accessories	Hermiston, Simonis and Wisoky	Jessamine Apark	Winny Agnolo	PC	64-5761908	Aug	2020

In [55]:
```
Appliances = Appliances_Category[['device_type' , 'category' , 'country']]
```

In [56]:
```
Appliances
```

Out[56]:

	device_type	category	country
2	PC	Clothing	Portugal
3	PC	Beauty	Portugal
5	PC	Clothing	Spain
6	PC	Books	Portugal
7	PC	Accessories	UK

In [57]:
```
Appliances_category = Appliances.groupby('country')['device_type']
```

```
In [58]: Appliances_category.count().head()
```

```
Out[58]: country
         Portugal    3
         Spain       1
         UK          1
         Name: device_type, dtype: int64
```

There is not a single country which sold PC in Appliances Category

3.7 The Average Cost spend by the Report from the year 2019-2020

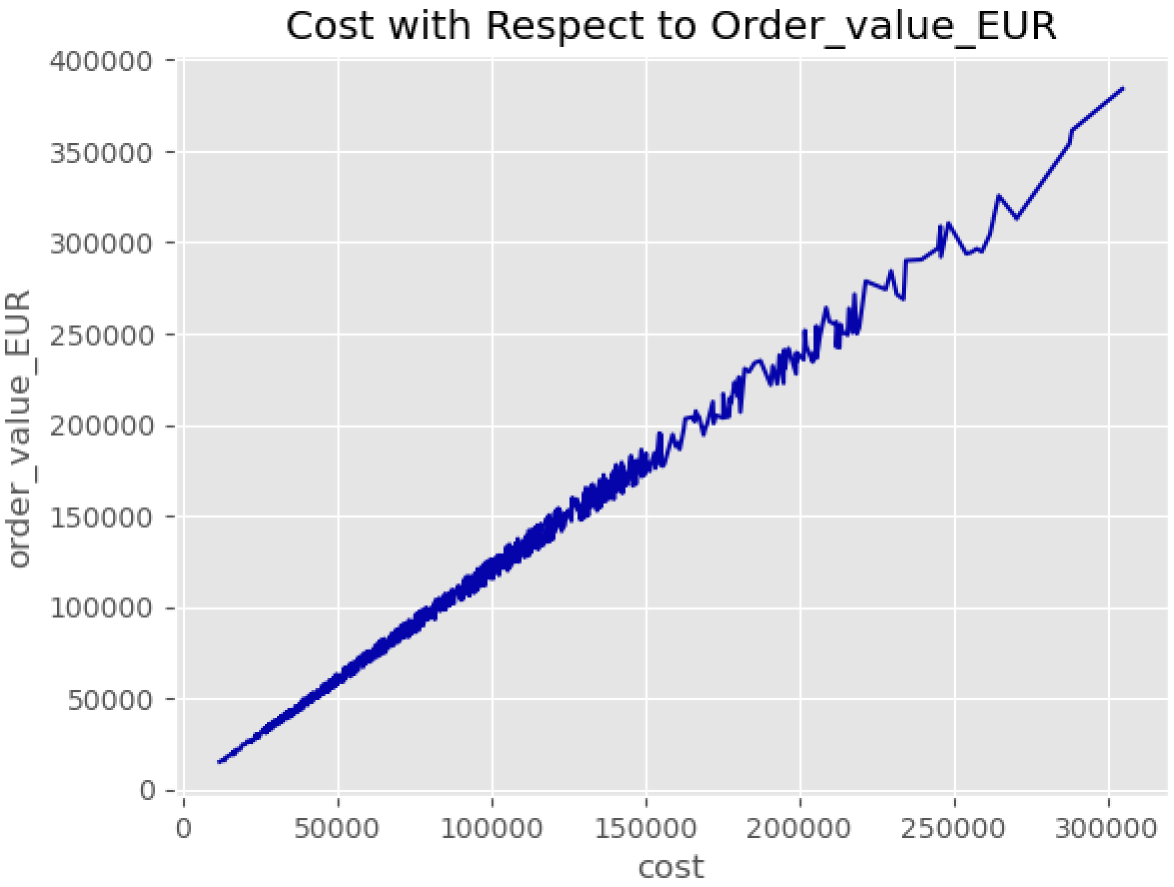
```
In [90]: mean = Sales_Export[['cost', 'order_value_EUR']].mean().head()
```

```
In [91]: mean
```

```
Out[91]: cost                  94369.313
         order_value_EUR    113361.747
         dtype: float64
```

```
In [89]: sns.lineplot(data = Sales_Export, x = 'cost' , y = 'order_value_EUR' , color='#0504aa' )
         plt.title("Cost with Respect to Order_value_EUR")
```

```
Out[89]: Text(0.5, 1.0, 'Cost with Respect to Order_value_EUR')
```



3.8 Maximum Selling Month wise

```
In [60]: month_group = Sales_Export.groupby('month')['order_id']
```

In [63]:

```
monthly_selling = month_group.count().sort_values(ascending = False ).head(12)
```

In [64]:

```
monthly_selling
```

Out[64]:

```
month
Jan    95
Mar    93
May    93
Oct    93
Jun    90
Sep    83
Nov    78
Dec    77
Aug    76
Jul    76
Apr    74
Feb    72
Name: order_id, dtype: int64
```

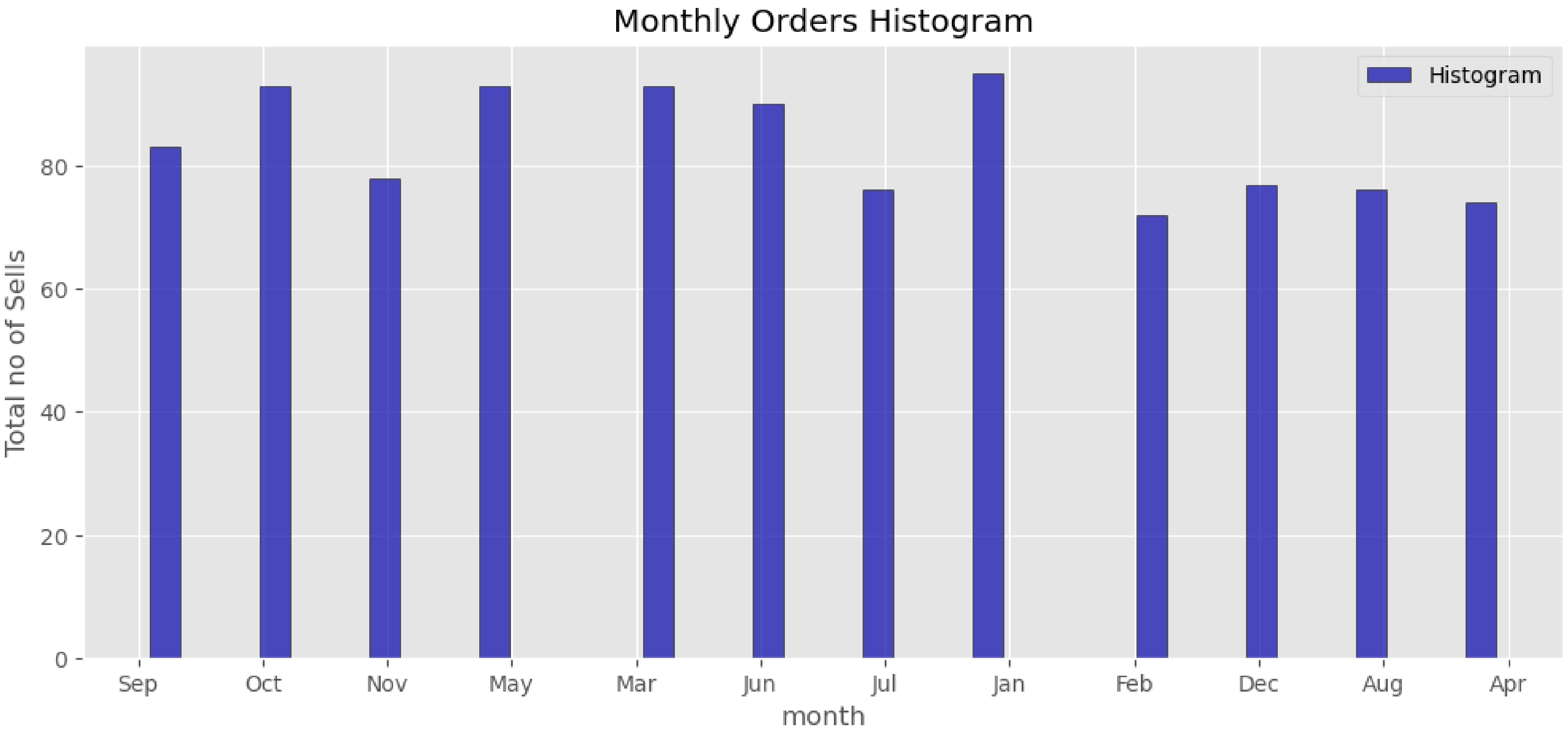
In [84]:

```
x_axis = Sales_Export[['month']].sort_values(by = 'month' , ascending = False)
```

*** Graph Between Categoral Count With Respect to Month

In [85]:

```
style.use('ggplot')

plt.figure(figsize = (12,5))
plt.hist(x_axis,bins = 25,rwidth=0.55,align='mid',alpha=0.7, histtype='bar',orientation='vertical', edgecolor='black',
plt.xlabel('month')
plt.grid(axis='y', alpha=0.75)
plt.ylabel('Total no of Sells')
plt.title('Monthly Orders Histogram')
max_numbers = y_axis.max()
plt.legend()
plt.show()
```



4.1 Correlation

In [12]:

```
Sales_Export.corr()
```

```
C:\Users\sande\AppData\Local\Temp\ipykernel_12260\608406433.py:1: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  Sales_Export.corr()
```

Out[12]:

	order_value_EUR	cost	year
order_value_EUR	1.000000	0.997733	-0.034675
cost	0.997733	1.000000	-0.034533
year	-0.034675	-0.034533	1.000000

4.2 Co-varience

In [134]:

```
Sales_Export.cov()
```

```
C:\Users\sande\AppData\Local\Temp\ipykernel_12260\4291517802.py:1: FutureWarning: The default value of numeric_only
in DataFrame.cov is deprecated. In a future version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  Sales_Export.cov()
```

Out[134]:

	order_value_EUR	cost	year
order_value_EUR	3.816190e+09	3.176684e+09	-1071.346316
cost	3.176684e+09	2.656372e+09	-890.179810
year	-1.071346e+03	-8.901798e+02	0.250150

5.0 Heatmap

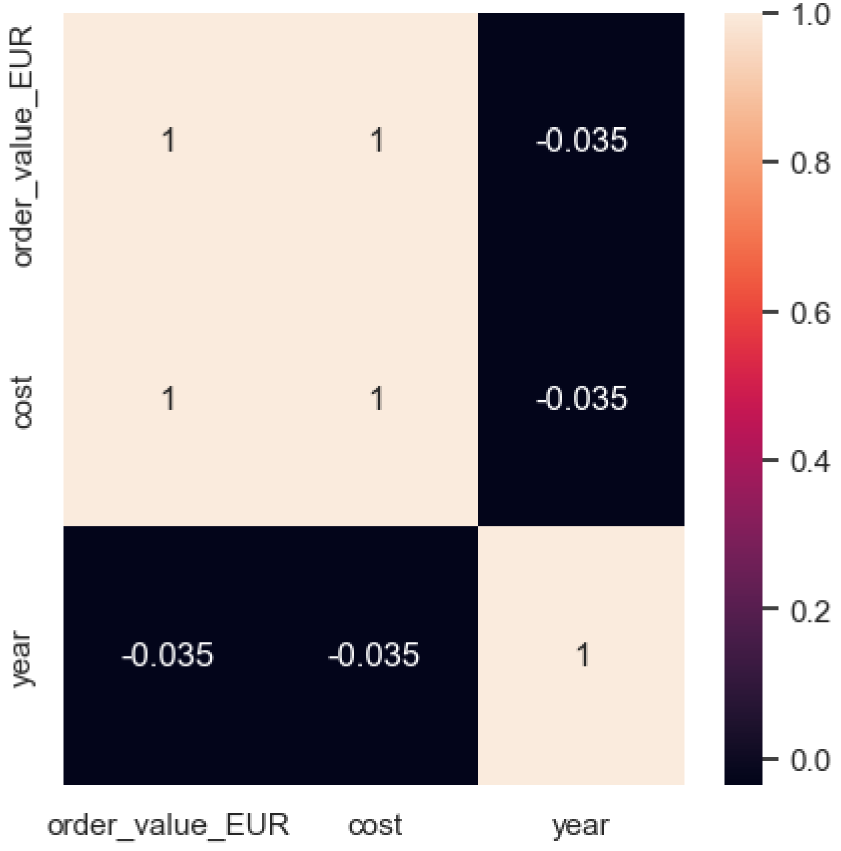
In [133]:

```
sns.set()
plt.subplots(figsize = (5,5))
sns.heatmap(Sales_Export.corr(),annot = True)
```

```
C:\Users\sande\AppData\Local\Temp\ipykernel_12260\2403753365.py:3: FutureWarning: The default value of numeric_only
in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify
the value of numeric_only to silence this warning.
  sns.heatmap(Sales_Export.corr(),annot = True)
```

Out[133]:

```
<AxesSubplot:>
```


order_value_EUR
cost
year
order_value_EUR
cost
year