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

In [2]:
```
engine = mysql.connector.connect(
    host = "127.0.0.1",
    user = "root",
    password = "",
    database = "crm_sales",
    use_pure = True
)
cur = engine.cursor()
```

In [3]:
```
cur.execute("SELECT * FROM accounts")
```

In [4]:
```
result = cur.fetchall()
df = pd.DataFrame(result)
```

In [5]:
```
query = "SELECT * FROM accounts"
```

In [6]:
```
df = pd.read_sql(query, engine)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\1589124803.py:1: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  df = pd.read_sql(query, engine)
```

In [7]:
```
df.to_csv('accounts.csv')
```

In [8]:
```
df = pd.read_csv('accounts.csv')
```

In [9]:
```
df
```

Out[9]:

	Unnamed: 0	account	sector	year_established	revenue	employees	office_location
0	0	Acme Corporation	technolgy	1996	1100.04	2822	United States
1	1	Betasoloin	medical	1999	251.41	495	United States
2	2	Betatech	medical	1986	647.18	1185	Kenya
3	3	Bioholding	medical	2012	587.34	1356	Philipines
4	4	Bioplex	medical	1991	326.82	1016	United States
...	...	...	...	...	...	...	...
80	80	Zathunicon	retail	2010	71.12	144	United States
81	81	Zencorporation	technolgy	2011	40.79	142	China
82	82	Zoomit	entertainment	1992	324.19	978	United States
83	83	Zotware	software	1979	4478.47	13809	United States
84	84	Zumgoity	medical	1984	441.08	1210	United States

85 rows × 7 columns

In [10]:
```
cur.execute("SELECT * FROM productss")
```

In [11]:
```
result = cur.fetchall()
df = pd.DataFrame(result)
```

In [12]:
```
query = "SELECT * FROM productss"
```

In [13]:
```
df = pd.read_sql(query, engine)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\1589124803.py:1: UserWarning: pandas only supports SQLAlchemy connect
able (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Pl
ease consider using SQLAlchemy.
  df = pd.read_sql(query, engine)
```

In [14]:
```
df.to_csv('productss.csv')
```

In [15]:
```
df = pd.read_csv('productss.csv')
```

In [16]:
```
df
```

Out[16]:

	Unnamed: 0	product	series	sales_price
0	0	GTX Basic	GTX	550
1	1	GTX Pro	GTX	4821
2	2	MG Special	MG	55
3	3	MG Advanced	MG	3393
4	4	GTX Plus Pro	GTX	5482
5	5	GTX Plus Basic	GTX	1096
6	6	GTK 500	GTK	26768

In [17]:
```
cur.execute("SELECT * FROM sales_pipeline")
```

In [18]:
```
result = cur.fetchall()
df = pd.DataFrame(result)
```

In [19]:
```
query = "SELECT * FROM sales_pipeline"
```

In [20]:
```
df = pd.read_sql(query, engine)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\1589124803.py:1: UserWarning: pandas only supports SQLAlchemy connect
able (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Pl
ease consider using SQLAlchemy.
  df = pd.read_sql(query, engine)
```

In [21]:
```
df.to_csv('sales_pipeline.csv')
```

In [22]:
```
df = pd.read_csv('sales_pipeline.csv')
```

In [23]:
```
df
```

Out[23]:

	Unnamed: 0	sales_agent	product	account	deal_stage	engage_date	close_date	close_value
0	0	Moses Frase	GTX Plus Basic	Cancity	Won	10/20/2016	3/1/2017	1054
1	1	Darcel Schlecht	GTXPro	Isdom	Won	10/25/2016	3/11/2017	4514
2	2	Darcel Schlecht	MG Special	Cancity	Won	10/25/2016	3/7/2017	50
3	3	Moses Frase	GTX Basic	Codehow	Won	10/25/2016	3/9/2017	588
4	4	Zane Levy	GTX Basic	Hatfan	Won	10/25/2016	3/2/2017	517
...	...	...	...	...	...	...	...	...
6706	6706	Lajuana Vencill	GTX Basic	Conecom	Won	12/24/2017	12/26/2017	622
6707	6707	Violet Mclelland	GTX Plus Basic	Bluth Company	Won	12/24/2017	12/30/2017	1093
6708	6708	Maureen Marcano	GTXPro	Hottechi	Won	12/26/2017	12/29/2017	4433
6709	6709	Gladys Colclough	GTX Plus Basic	Inity	Won	12/27/2017	12/30/2017	1052
6710	6710	Gladys Colclough	MG Special	Betatech	Won	12/27/2017	12/29/2017	67

6711 rows × 8 columns

In [24]:
```
cur.execute("SELECT * FROM sales_teams")
```

In [25]:
```
result = cur.fetchall()
df = pd.DataFrame(result)
```

In [26]:
```
query = "SELECT * FROM sales_teams"
```

In [27]:
```
df = pd.read_sql(query, engine)
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\1589124803.py:1: UserWarning: pandas only supports SQLAlchemy connect
able (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Pl
ease consider using SQLAlchemy.
  df = pd.read_sql(query, engine)
```

In [28]:
```
df.to_csv('sales_teams.csv')
```

In [29]:
```
df = pd.read_csv('sales_teams.csv')
```

In [30]:
```
df
```

Out[30]:

	Unnamed: 0	sales_agent	manager	regional_office
0	0	Anna Snelling	Dustin Brinkmann	Central
1	1	Cecily Lampkin	Dustin Brinkmann	Central
2	2	Versie Hillebrand	Dustin Brinkmann	Central
3	3	Lajuana Vencill	Dustin Brinkmann	Central
4	4	Moses Frase	Dustin Brinkmann	Central
5	5	Jonathan Berthelot	Melvin Marxen	Central
6	6	Marty Freudenburg	Melvin Marxen	Central
7	7	Gladys Colclough	Melvin Marxen	Central
8	8	Niesha Huffines	Melvin Marxen	Central
9	9	Darcel Schlecht	Melvin Marxen	Central
10	10	Mei-Mei Johns	Melvin Marxen	Central
11	11	Violet Mclelland	Cara Losch	East
12	12	Corliss Cosme	Cara Losch	East
13	13	Rosie Papadopoulos	Cara Losch	East
14	14	Garret Kinder	Cara Losch	East
15	15	Wilburn Farren	Cara Losch	East
16	16	Elizabeth Anderson	Cara Losch	East
17	17	Daniell Hammack	Rocco Neubert	East
18	18	Cassey Cress	Rocco Neubert	East
19	19	Donn Cantrell	Rocco Neubert	East
20	20	Reed Clapper	Rocco Neubert	East
21	21	Boris Faz	Rocco Neubert	East

	Unnamed: 0	sales_agent	manager	regional_office
22	22	Natalya Ivanova	Rocco Neubert	East
23	23	Vicki Laflamme	Celia Rouche	West
24	24	Rosalina Dieter	Celia Rouche	West
25	25	Hayden Neloms	Celia Rouche	West
26	26	Markita Hansen	Celia Rouche	West
27	27	Elease Gluck	Celia Rouche	West
28	28	Carol Thompson	Celia Rouche	West
29	29	James Ascencio	Summer Sewald	West
30	30	Kary Hendrixson	Summer Sewald	West
31	31	Kami Bicknell	Summer Sewald	West
32	32	Zane Levy	Summer Sewald	West
33	33	Maureen Marcano	Summer Sewald	West
34	34	Carl Lin	Summer Sewald	West

In [31]:
```
df = pd.read_csv('accounts.csv')
```

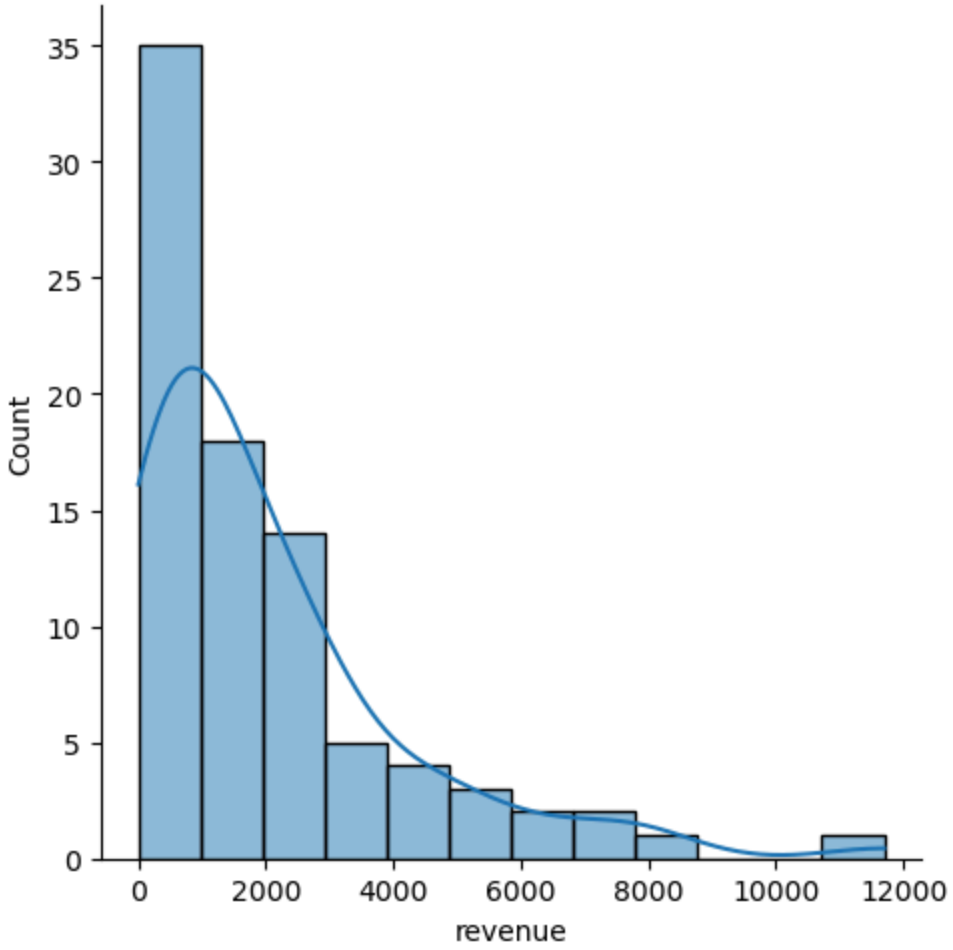
In [32]:
```
df.describe()
```

Out[32]:

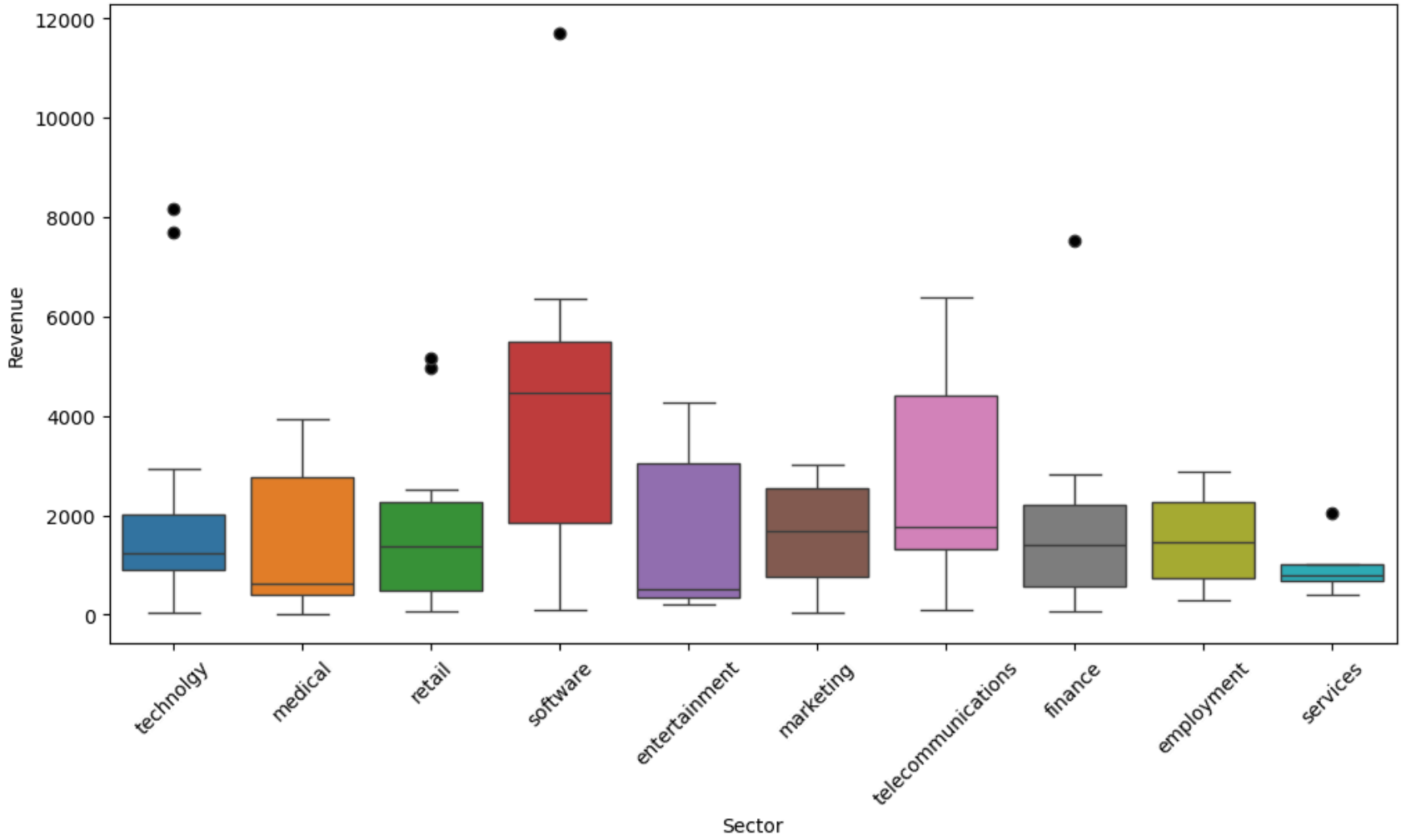
	Unnamed: 0	year_established	revenue	employees
count	85.000000	85.000000	85.000000	85.000000
mean	42.000000	1996.105882	1994.632941	4660.823529
std	24.681302	8.865427	2169.491436	5715.601198
min	0.000000	1979.000000	4.540000	9.000000
25%	21.000000	1989.000000	497.110000	1179.000000
50%	42.000000	1996.000000	1223.720000	2769.000000
75%	63.000000	2002.000000	2741.370000	5595.000000
max	84.000000	2017.000000	11698.030000	34288.000000

In [33]:
```
sns.displot(data = df, x = 'revenue', kind = "hist",
            kde = True)
```

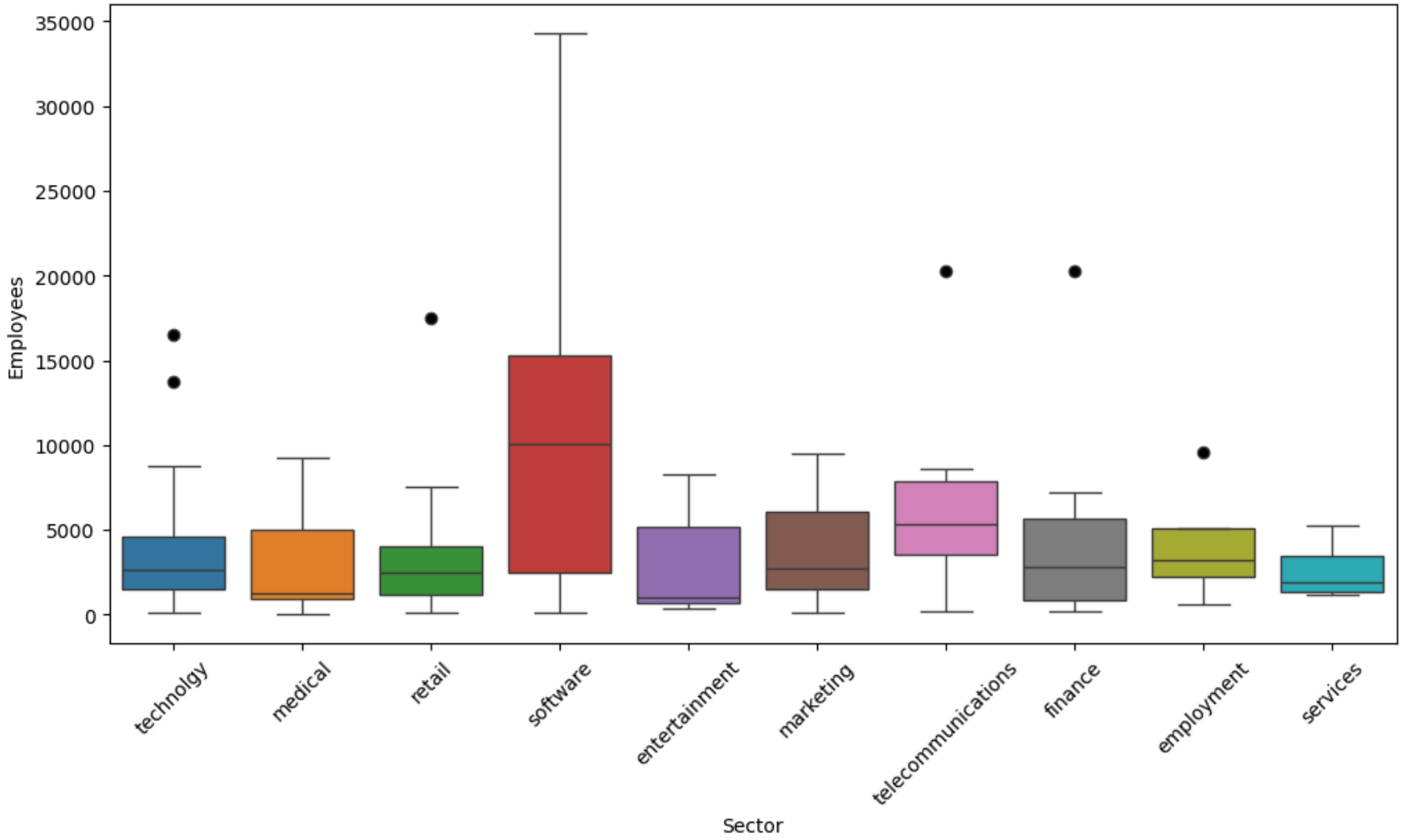
Out[33]:
```
<seaborn.axisgrid.FacetGrid at 0x1dea9046420>
```



In [34]:
```
plt.figure(figsize = (12, 6))
sns.boxplot(x = 'sector', y = 'revenue',
            hue = 'sector', data = df,
           flierprops = dict (markerfacecolor = 'black'))
plt.xlabel('Sector')
plt.ylabel('Revenue')
plt.xticks(rotation = 45)
plt.show()
```

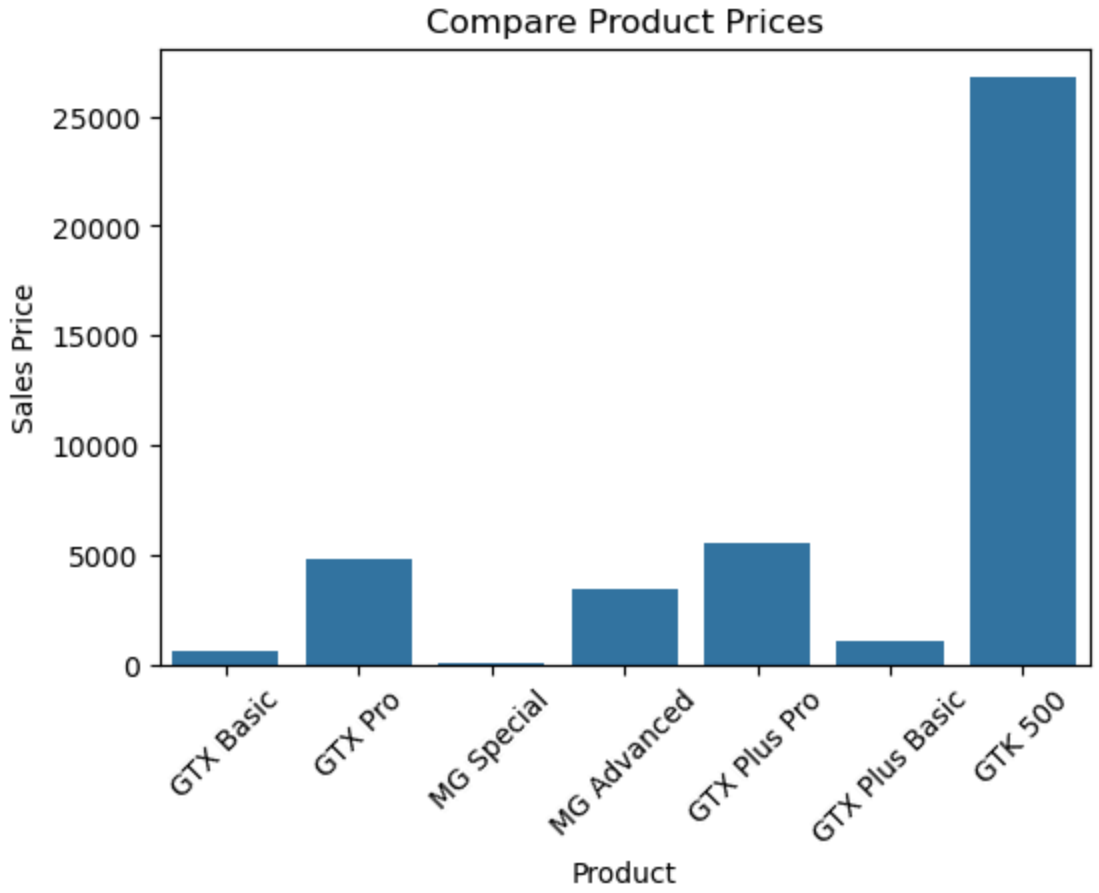


In [35]:
```
plt.figure(figsize = (12, 6))
sns.boxplot(x = 'sector', y = 'employees',
            hue = 'sector', data = df,
           flierprops = dict (markerfacecolor = 'black'))
plt.xlabel('Sector')
plt.ylabel('Employees')
plt.xticks(rotation = 45)
plt.show()
```



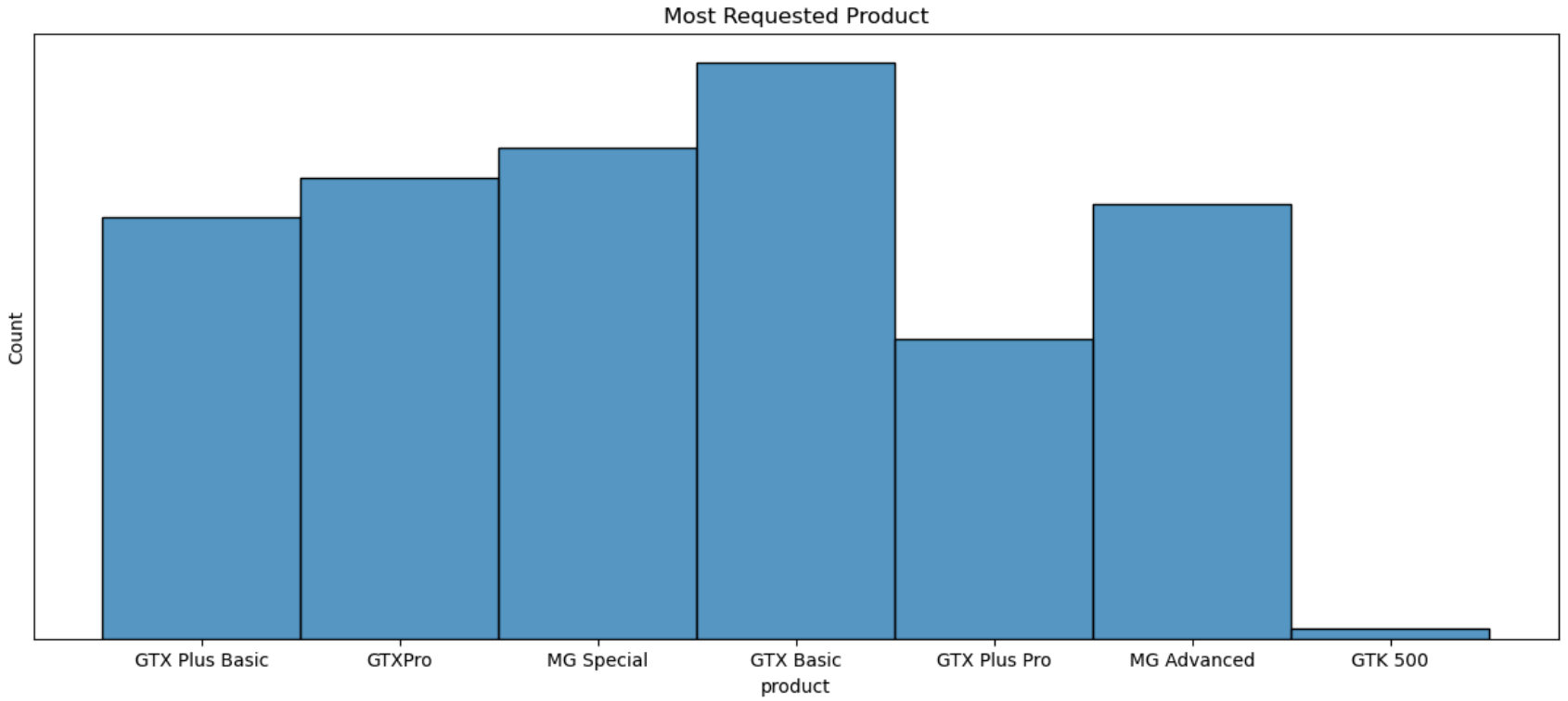
In [36]:
```
df = pd.read_csv('productss.csv')
plt.figure(figsize = (6, 4))
sns.barplot(x='product',y='sales_price', data=df, width = 0.8,
            ci = None)
plt.xlabel('Product')
plt.ylabel('Sales Price')
plt.title('Compare Product Prices')
plt.xticks(rotation = 45)
plt.show()
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\3444211792.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='product',y='sales_price', data=df, width = 0.8,
```



In [37]:
```
df = pd.read_csv('sales_pipeline.csv')
```

In [38]:
```
plt.figure(figsize = (15, 6))
sns.histplot( x = 'product', data = df,
    bins = 36)
plt.yticks([])
plt.title('Most Requested Product')
plt.show()
```



In [39]:
```
df1 = pd.read_csv('sales_pipeline.csv')
df2 = pd.read_csv('sales_teams.csv')
merged_df = pd.merge(df1, df2,
                     on = 'sales_agent')
merged_df.to_csv('sales.csv',
                 index = True)
```

In [40]:
```
df = pd.read_csv('sales.csv')
df.head()
```

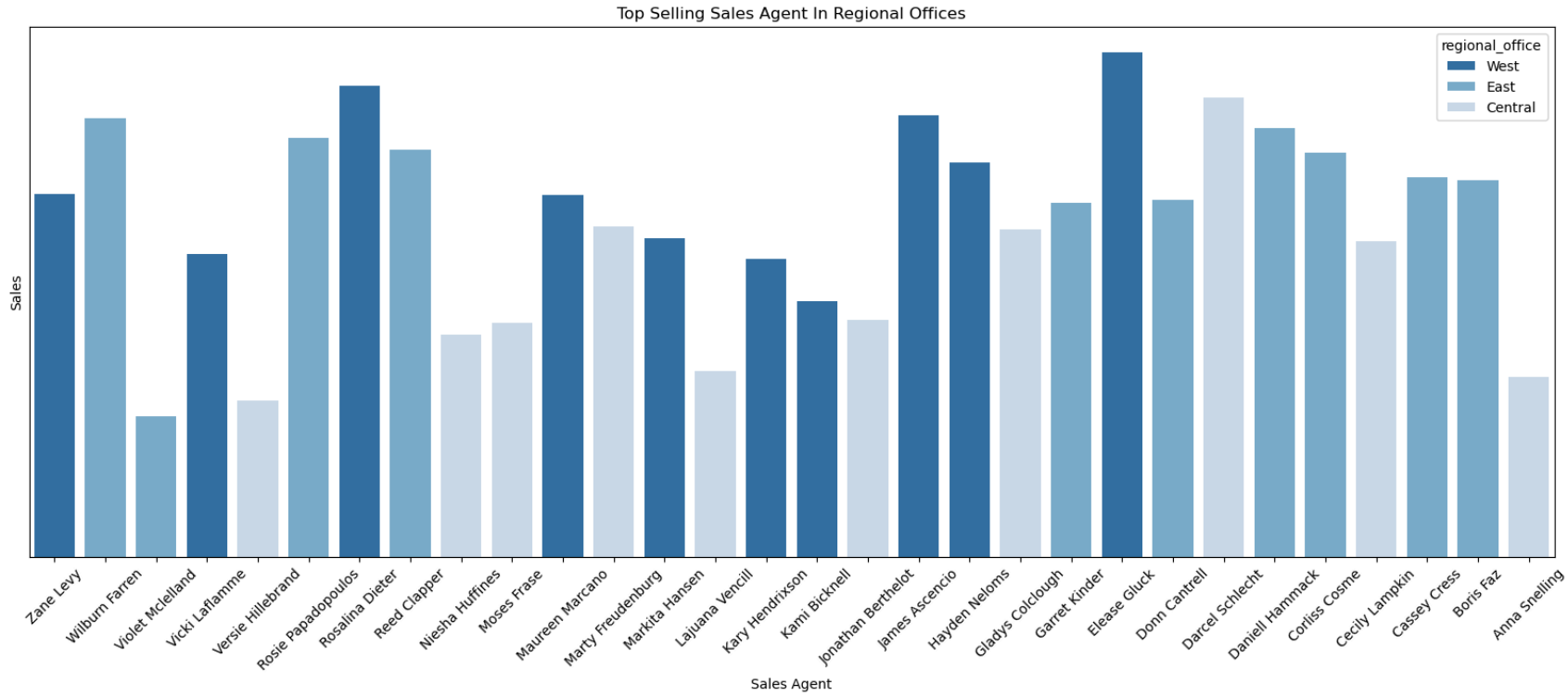
Out[40]:

	Unnamed: 0	Unnamed: 0_x	sales_agent	product	account	deal_stage	engage_date	close_date	close_value	Unnamed: 0_y	manag
0	0	0	Moses Frase	GTX Plus Basic	Cancity	Won	10/20/2016	3/1/2017	1054	4	Dust Brinkma
1	1	1	Darcel Schlecht	GTXPro	Isdom	Won	10/25/2016	3/11/2017	4514	9	Melv Marx
2	2	2	Darcel Schlecht	MG Special	Cancity	Won	10/25/2016	3/7/2017	50	9	Melv Marx
3	3	3	Moses Frase	GTX Basic	Codehow	Won	10/25/2016	3/9/2017	588	4	Dust Brinkma
4	4	4	Zane Levy	GTX Basic	Hatfan	Won	10/25/2016	3/2/2017	517	32	Summ Sewa

In [41]:

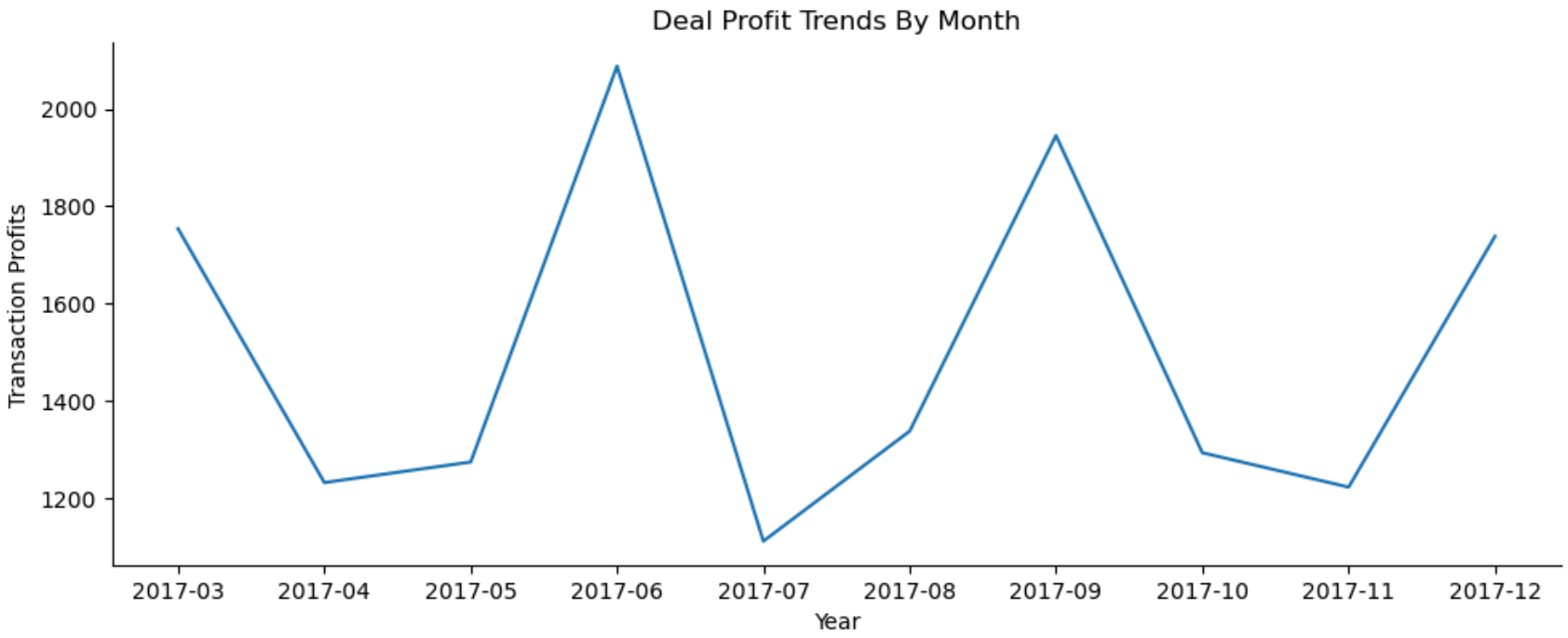
```
plt.figure(figsize = (20, 7))
df = df.sort_values(by = 'sales_agent',
                    ascending = False)
sns.barplot( x = 'sales_agent', y = 'close_value',
            hue = 'regional_office',palette = 'Blues_r', ci = None,
            data = df)
plt.xticks(rotation = 45)
plt.yticks([])
plt.xlabel('Sales Agent')
plt.ylabel('Sales')
plt.title('Top Selling Sales Agent In Regional Offices')
plt.show()
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_9520\599719002.py:4: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot( x = 'sales_agent', y = 'close_value',
```



In [42]:
```
plt.figure(figsize = (10, 4))
df['close_date'] = pd.to_datetime(df['close_date'])
df['Year'] = df['close_date'].dt.strftime('%Y-%m')
df = df.sort_values('close_date')
sns.relplot(x = 'Year' , y = 'close_value', data = df, kind = 'line', errorbar = None)
plt.gcf().set_size_inches((10, 4))
plt.tight_layout()
plt.xlabel('Year')
plt.ylabel('Transaction Profits')
plt.title('Deal Profit Trends By Month')
plt.show()
```

```
<Figure size 1000x400 with 0 Axes>
```


Deal Profit Trends By Month
Transaction Profits
2000
1800
1600
1400
1200
2017-03
2017-04
2017-05
2017-06
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
Year