# Corporate Performance and Sales Pipeline Analysis

## Load and Examine the Data

In [2]:
```
import pandas as pd

# Load datasets
accounts = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\
data_dictionary = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New
products = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\
sales_pipeline = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New f
sales_teams = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New fold

# Display basic info
datasets = [accounts, data_dictionary, products, sales_pipeline, sales_teams]
dataset_names = ['Accounts', 'Data Dictionary', 'Products', 'Sales Pipeline', 'Sales Tea

for name, data in zip(dataset_names, datasets):
    print(f"Dataset: {name}")
    print(data.info())
    print(data.head())
    print("\n")
```

```
Dataset: Accounts
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   account           85 non-null     object
 1   sector            85 non-null     object
 2   year_established  85 non-null     int64
 3   revenue           85 non-null     float64
 4   employees         85 non-null     int64
 5   office_location   85 non-null     object
 6   subsidiary_of     15 non-null     object
dtypes: float64(1), int64(2), object(4)
memory usage: 4.8+ KB
None
            account     sector  year_established  revenue  employees  \
0  Acme Corporation  technolgy              1996  1100.04       2822
1        Betasoloin    medical              1999   251.41        495
2          Betatech    medical              1986   647.18       1185
3        Bioholding    medical              2012   587.34       1356
4           Bioplex    medical              1991   326.82       1016

  office_location subsidiary_of
0   United States           NaN
1   United States           NaN
2           Kenya           NaN
3      Philipines           NaN
4   United States           NaN


Dataset: Data Dictionary
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
```

```
 0   Table        21 non-null     object
 1   Field        21 non-null     object
 2   Description  21 non-null     object
dtypes: object(3)
memory usage: 636.0+ bytes
None
      Table             Field                          Description
0  accounts           account                        Company name
1  accounts            sector                            Industry
2  accounts  year_established                    Year Established
3  accounts           revenue  Annual revenue (in millions of USD)
4  accounts         employees                   Number of employees


Dataset: Products
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   product      7 non-null      object
 1   series       7 non-null      object
 2   sales_price  7 non-null      int64
dtypes: int64(1), object(2)
memory usage: 300.0+ bytes
None
        product series  sales_price
0     GTX Basic    GTX          550
1       GTX Pro    GTX         4821
2    MG Special     MG           55
3   MG Advanced     MG         3393
4  GTX Plus Pro    GTX         5482


Dataset: Sales Pipeline
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8800 entries, 0 to 8799
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   opportunity_id  8800 non-null   object
 1   sales_agent     8800 non-null   object
 2   product         8800 non-null   object
 3   account         7375 non-null   object
 4   deal_stage      8800 non-null   object
 5   engage_date     8300 non-null   object
 6   close_date      6711 non-null   object
 7   close_value     6711 non-null   float64
dtypes: float64(1), object(7)
memory usage: 550.1+ KB
None
  opportunity_id      sales_agent         product  account deal_stage  \
0       1C1I7A6R      Moses Frase  GTX Plus Basic  Cancity        Won
1       Z063OYW0  Darcel Schlecht          GTXPro    Isdom        Won
2       EC4QE1BX  Darcel Schlecht      MG Special  Cancity        Won
3       MV1LWRNH      Moses Frase       GTX Basic  Codehow        Won
4       PE84CX4O        Zane Levy       GTX Basic   Hatfan        Won

  engage_date  close_date  close_value
0  2016-10-20  2017-03-01       1054.0
1  2016-10-25  2017-03-11       4514.0
2  2016-10-25  2017-03-07         50.0
3  2016-10-25  2017-03-09        588.0
4  2016-10-25  2017-03-02        517.0
```

```
Dataset: Sales Teams
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   sales_agent      35 non-null     object
 1   manager          35 non-null     object
 2   regional_office  35 non-null     object
dtypes: object(3)
memory usage: 972.0+ bytes
None
         sales_agent           manager regional_office
0      Anna Snelling  Dustin Brinkmann         Central
1     Cecily Lampkin  Dustin Brinkmann         Central
2  Versie Hillebrand  Dustin Brinkmann         Central
3    Lajuana Vencill  Dustin Brinkmann         Central
4        Moses Frase  Dustin Brinkmann         Central
```

# EXPLORATORY DATA ANALYSIS

In [5]:

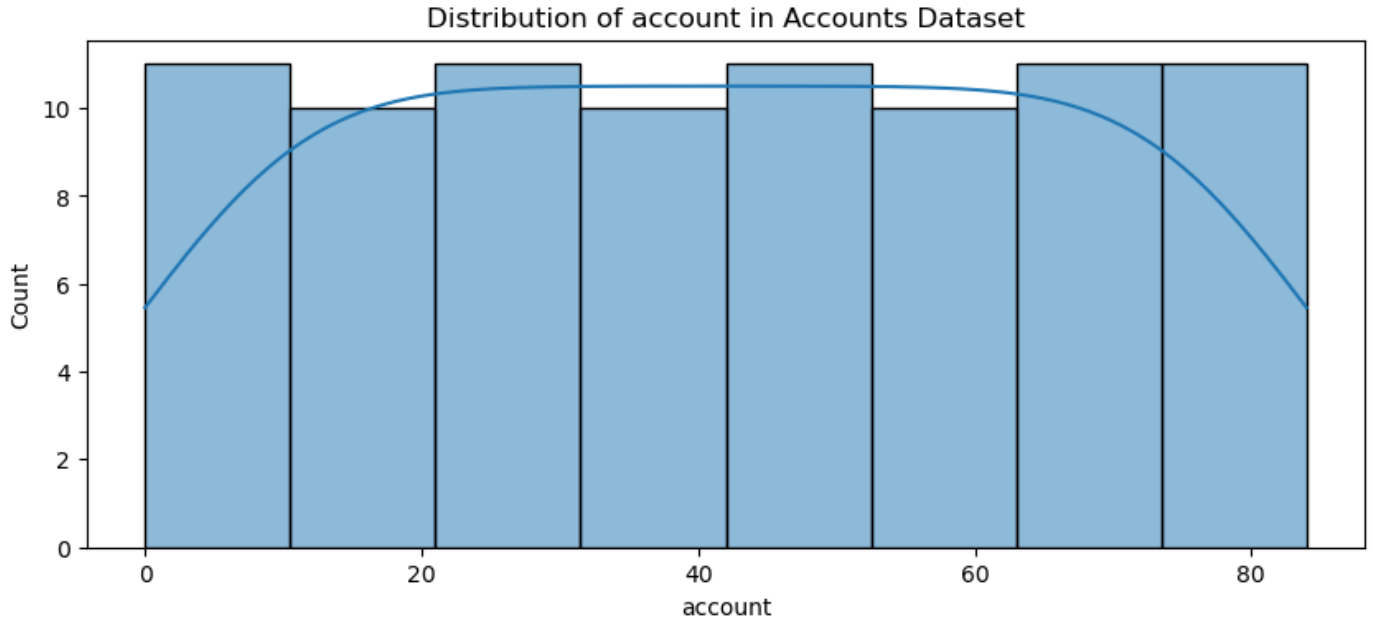
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load datasets
accounts = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\
data_dictionary = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New 
products = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\
sales_pipeline = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New f
sales_teams = pd.read_csv("C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New fold

# Handle missing values in Accounts dataset
accounts['subsidiary_of'].fillna('None', inplace=True)

# Convert categorical columns to numeric using label encoding for correlation analysis
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()

for column in accounts.select_dtypes(include=[object]).columns:
    accounts[column] = label_encoder.fit_transform(accounts[column])

# Display basic info and head of each dataset
datasets = {
    'Accounts': accounts,
    'Data Dictionary': data_dictionary,
    'Products': products,
    'Sales Pipeline': sales_pipeline,
    'Sales Teams': sales_teams
}

for name, data in datasets.items():
    print(f"Dataset: {name}")
    print(data.info())
    print(data.head(), "\n")

# EDA on Accounts dataset
# Summary statistics
print("Summary Statistics - Accounts Dataset")
print(accounts.describe())
```

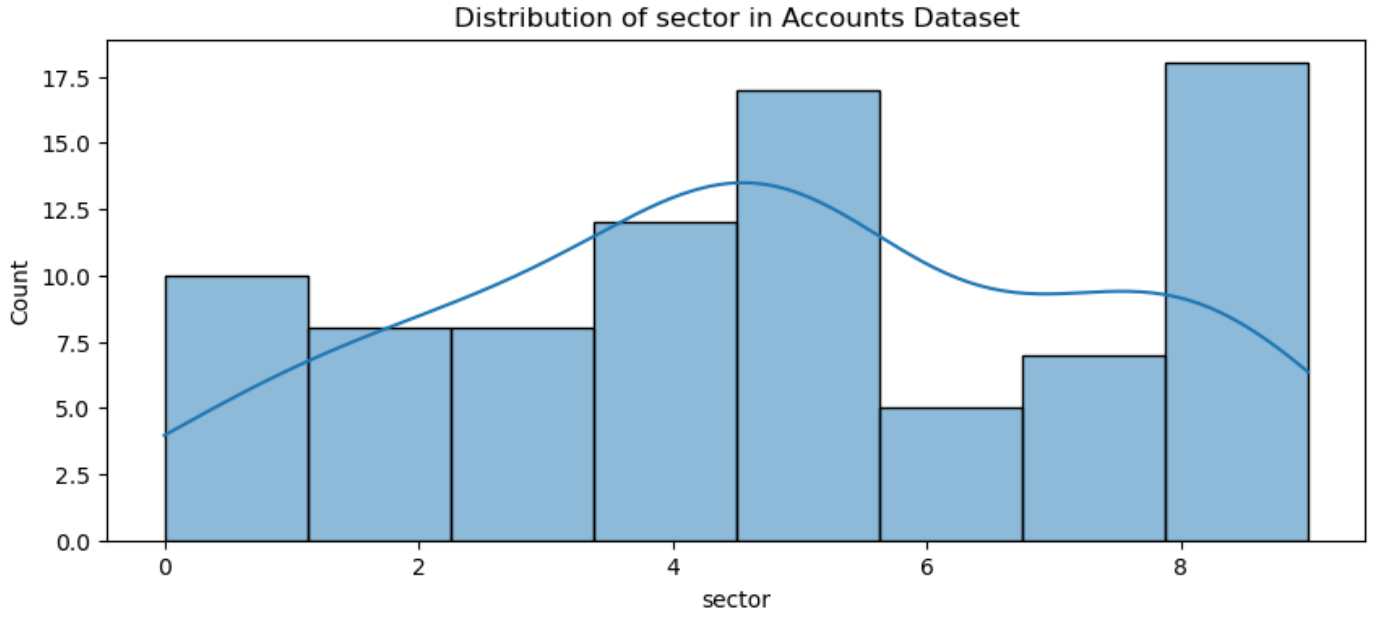
```python
# Data distribution
for column in accounts.select_dtypes(include=[np.number]).columns:
    plt.figure(figsize=(10, 4))
    sns.histplot(accounts[column], kde=True)
    plt.title(f'Distribution of {column} in Accounts Dataset')
    plt.show()

# Correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(accounts.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix - Accounts Dataset')
plt.show()

# Scatter plots for some relationships in Accounts dataset
sns.pairplot(accounts)
plt.show()
```

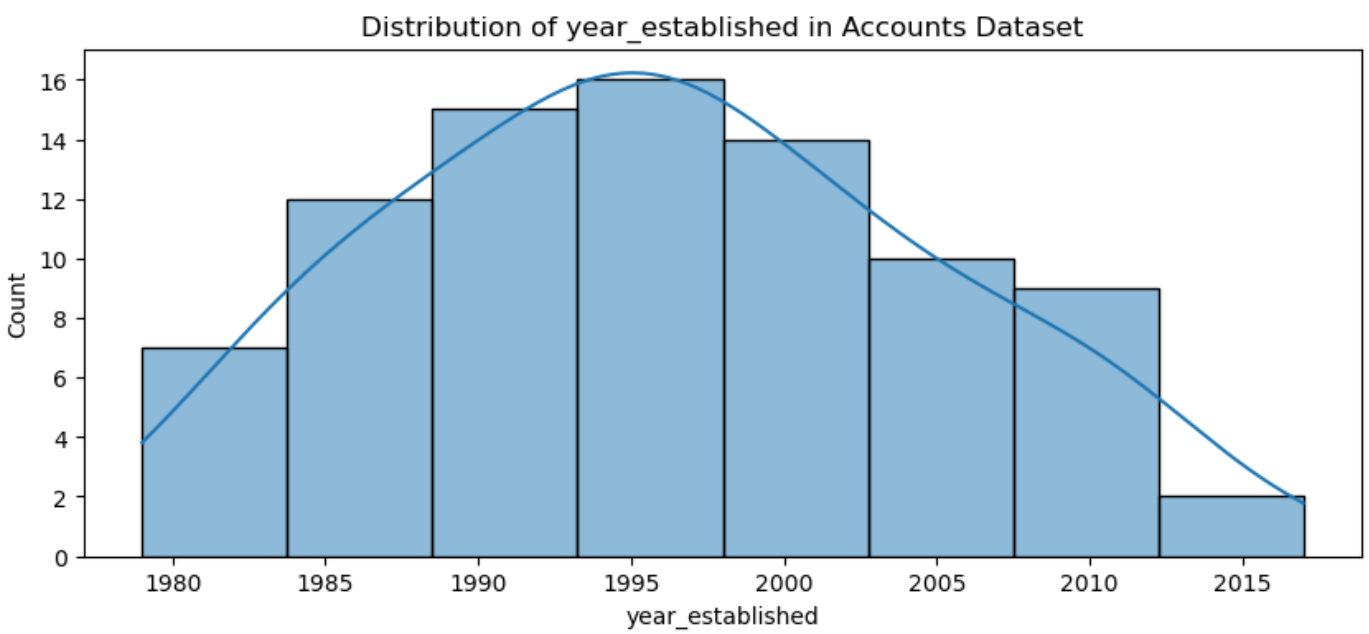
```
Dataset: Accounts
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   account           85 non-null     int32
 1   sector            85 non-null     int32
 2   year_established  85 non-null     int64
 3   revenue           85 non-null     float64
 4   employees         85 non-null     int64
 5   office_location   85 non-null     int32
 6   subsidiary_of     85 non-null     int32
dtypes: float64(1), int32(4), int64(2)
memory usage: 3.4 KB
None
   account  sector  year_established  revenue  employees  office_location  \
0        0       8              1996  1100.04       2822               14
1        1       4              1999   251.41        495               14
2        2       4              1986   647.18       1185                7
3        3       4              2012   587.34       1356               11
4        4       4              1991   326.82       1016               14

   subsidiary_of
0              5
1              5
2              5
3              5
4              5

Dataset: Data Dictionary
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Table        21 non-null     object
 1   Field        21 non-null     object
 2   Description  21 non-null     object
dtypes: object(3)
memory usage: 636.0+ bytes
None
      Table            Field                          Description
0  accounts          account                        Company name
1  accounts           sector                            Industry
2  accounts  year_established                   Year Established
3  accounts          revenue  Annual revenue (in millions of USD)
4  accounts        employees                 Number of employees
```

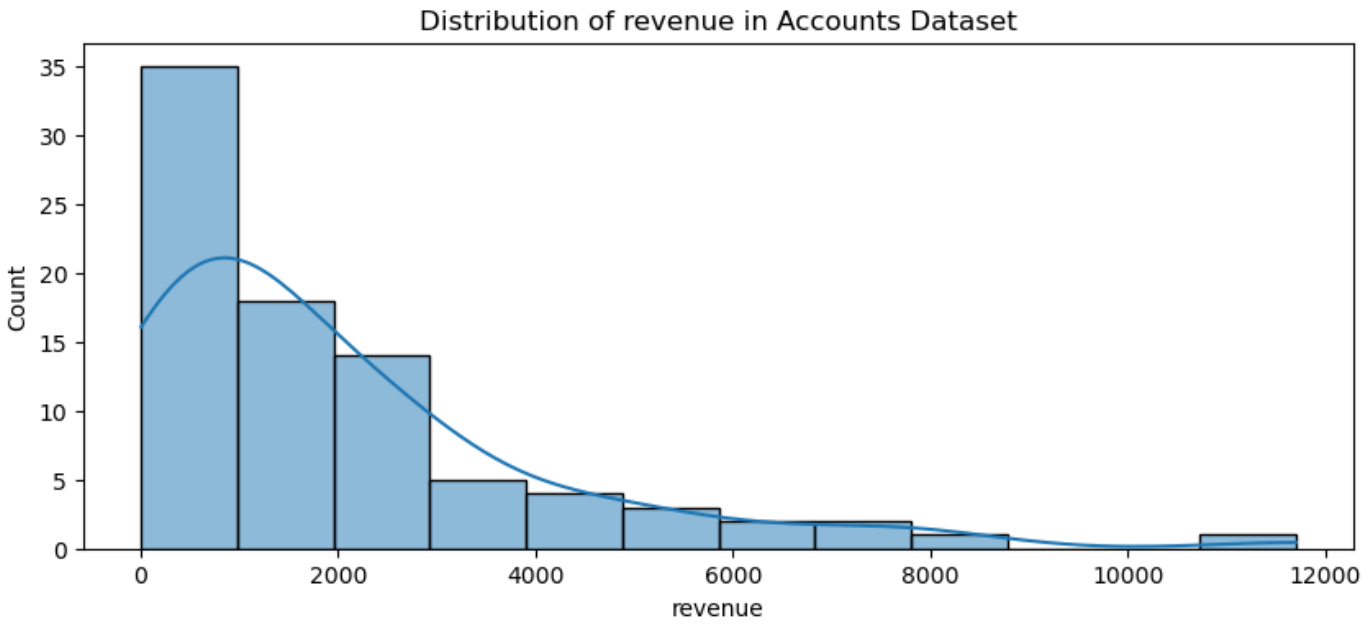
```
Dataset: Products
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   product      7 non-null      object
 1   series       7 non-null      object
 2   sales_price  7 non-null      int64
dtypes: int64(1), object(2)
memory usage: 300.0+ bytes
None
        product series  sales_price
0     GTX Basic    GTX          550
1       GTX Pro    GTX         4821
2    MG Special     MG           55
3   MG Advanced     MG         3393
4  GTX Plus Pro    GTX         5482

Dataset: Sales Pipeline
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8800 entries, 0 to 8799
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   opportunity_id  8800 non-null   object
 1   sales_agent     8800 non-null   object
 2   product         8800 non-null   object
 3   account         7375 non-null   object
 4   deal_stage      8800 non-null   object
 5   engage_date     8300 non-null   object
 6   close_date      6711 non-null   object
 7   close_value     6711 non-null   float64
dtypes: float64(1), object(7)
memory usage: 550.1+ KB
None
  opportunity_id      sales_agent         product  account deal_stage  \
0       1C1I7A6R      Moses Frase  GTX Plus Basic  Cancity        Won
1       Z063OYW0  Darcel Schlecht          GTXPro    Isdom        Won
2       EC4QE1BX  Darcel Schlecht      MG Special  Cancity        Won
3       MV1LWRNH      Moses Frase       GTX Basic  Codehow        Won
4       PE84CX4O        Zane Levy       GTX Basic   Hatfan        Won

  engage_date  close_date  close_value
0  2016-10-20  2017-03-01       1054.0
1  2016-10-25  2017-03-11       4514.0
2  2016-10-25  2017-03-07         50.0
3  2016-10-25  2017-03-09        588.0
4  2016-10-25  2017-03-02        517.0

Dataset: Sales Teams
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   sales_agent      35 non-null     object
 1   manager          35 non-null     object
 2   regional_office  35 non-null     object
dtypes: object(3)
memory usage: 972.0+ bytes
None
         sales_agent           manager regional_office
0      Anna Snelling  Dustin Brinkmann         Central
1     Cecily Lampkin  Dustin Brinkmann         Central
```

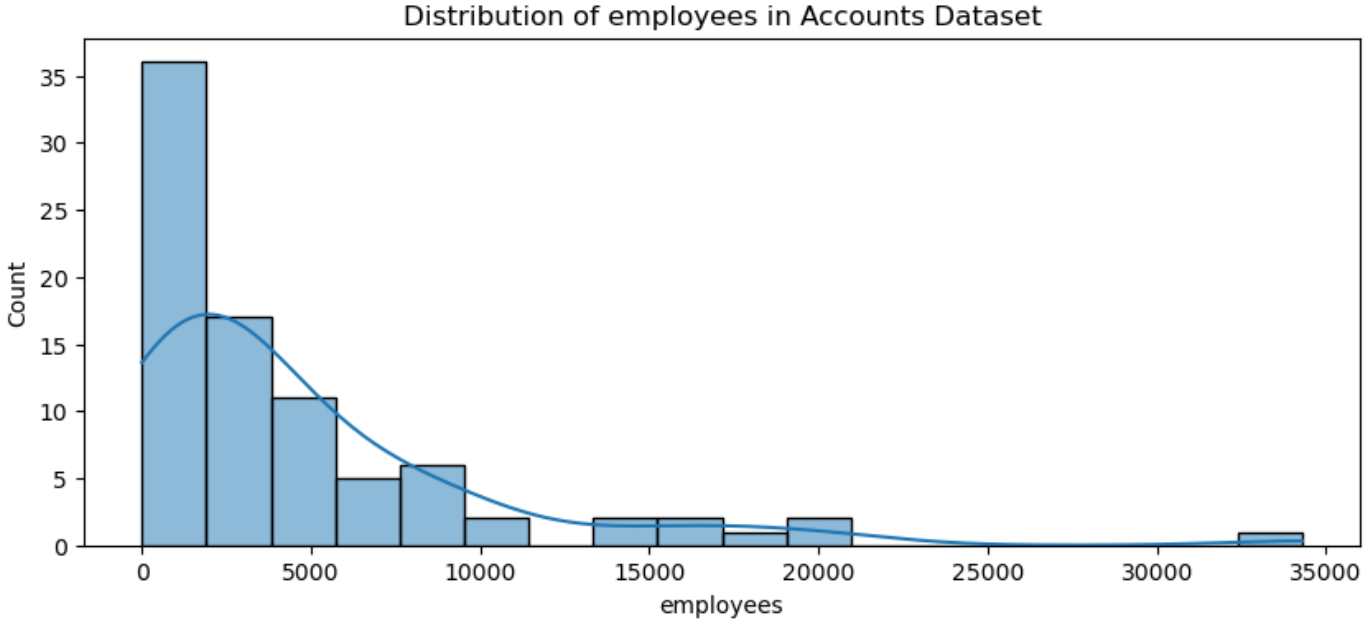
```
2  Versie Hillebrand  Dustin Brinkmann         Central
3    Lajuana Vencill  Dustin Brinkmann         Central
4        Moses Frase  Dustin Brinkmann         Central

Summary Statistics - Accounts Dataset
         account     sector  year_established       revenue     employees  \
count  85.000000  85.000000         85.000000     85.000000     85.000000
mean   42.000000   4.800000       1996.105882   1994.632941   4660.823529
std    24.681302   2.548576          8.865427   2169.491436   5715.601198
min     0.000000   0.000000       1979.000000      4.540000      9.000000
25%    21.000000   3.000000       1989.000000    497.110000   1179.000000
50%    42.000000   5.000000       1996.000000   1223.720000   2769.000000
75%    63.000000   7.000000       2002.000000   2741.370000   5595.000000
max    84.000000   9.000000       2017.000000  11698.030000  34288.000000

       office_location  subsidiary_of
count        85.000000      85.000000
mean         12.764706       4.600000
std           3.246416       1.346954
min           0.000000       0.000000
25%          14.000000       5.000000
50%          14.000000       5.000000
75%          14.000000       5.000000
max          14.000000       7.000000
```

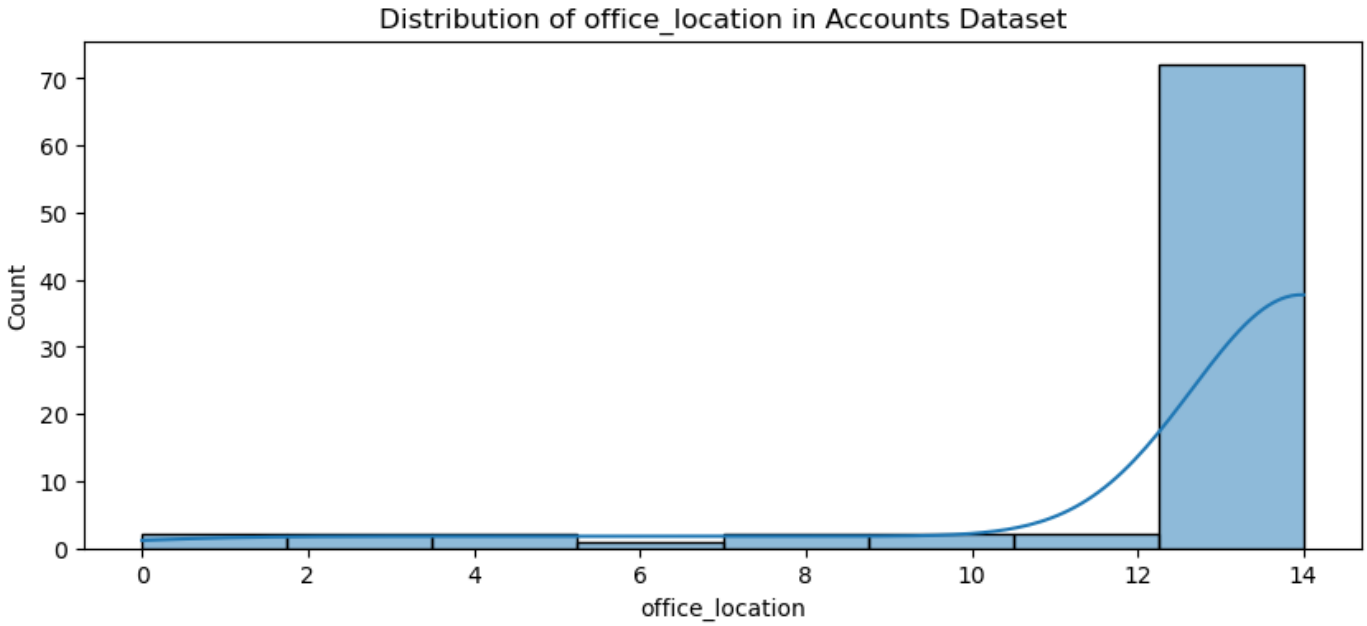


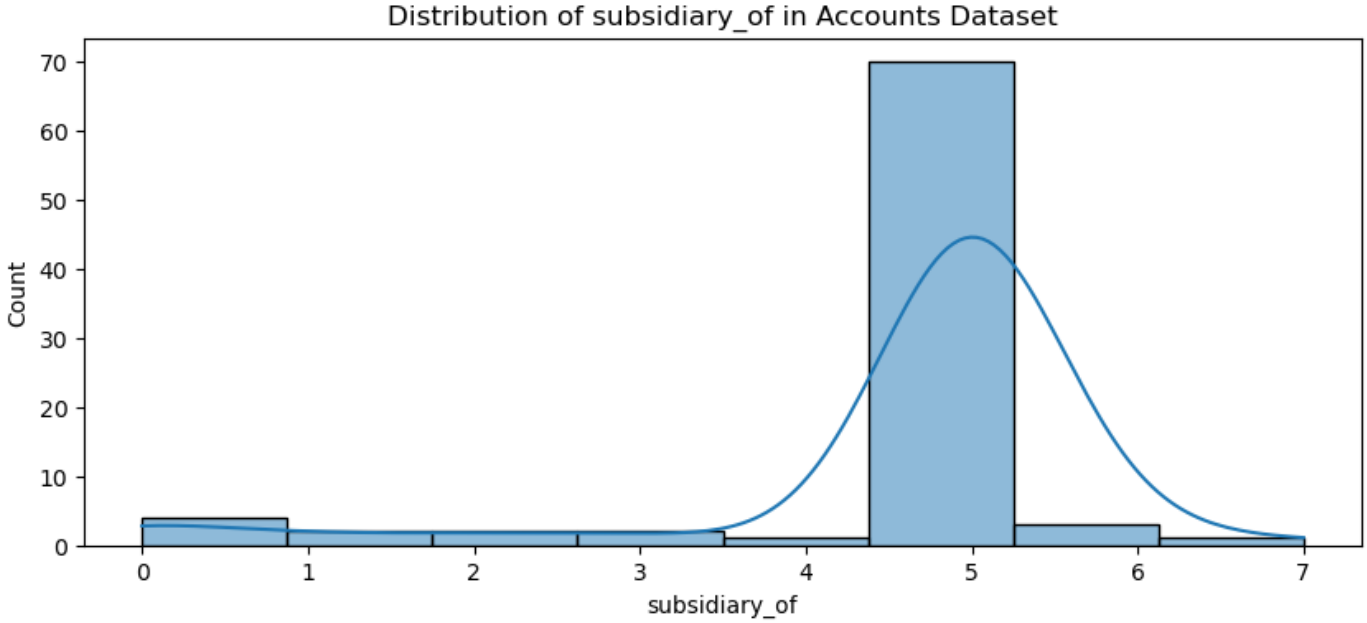



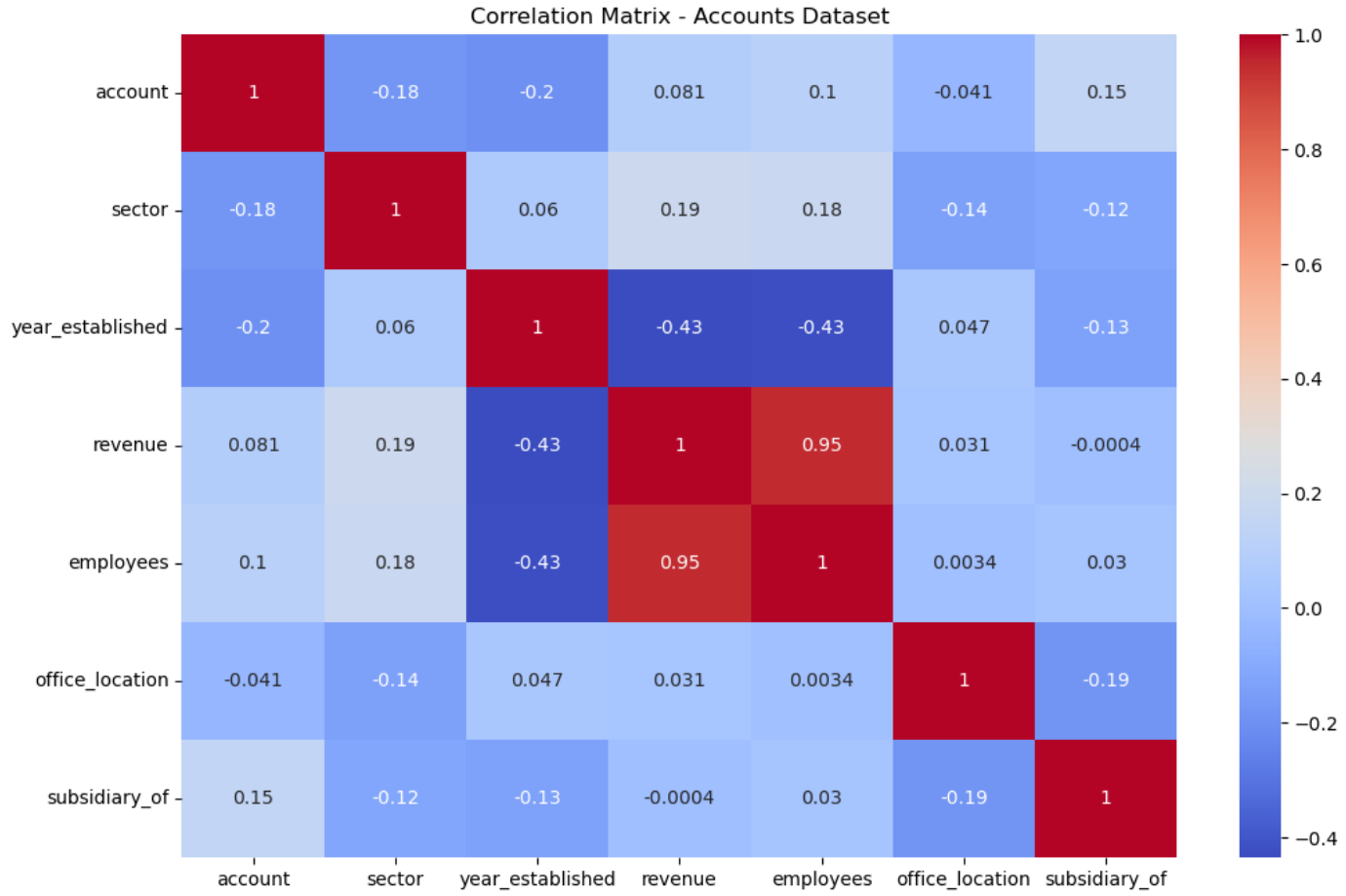

Distribution of year_established in Accounts Dataset
Count
year_established
0
2
4
6
8
10
12
14
16
1980
1985
1990
1995
2000
2005
2010
2015


Distribution of revenue in Accounts Dataset
Count
revenue
0
5
10
15
20
25
30
35
0
2000
4000
6000
8000
10000
12000

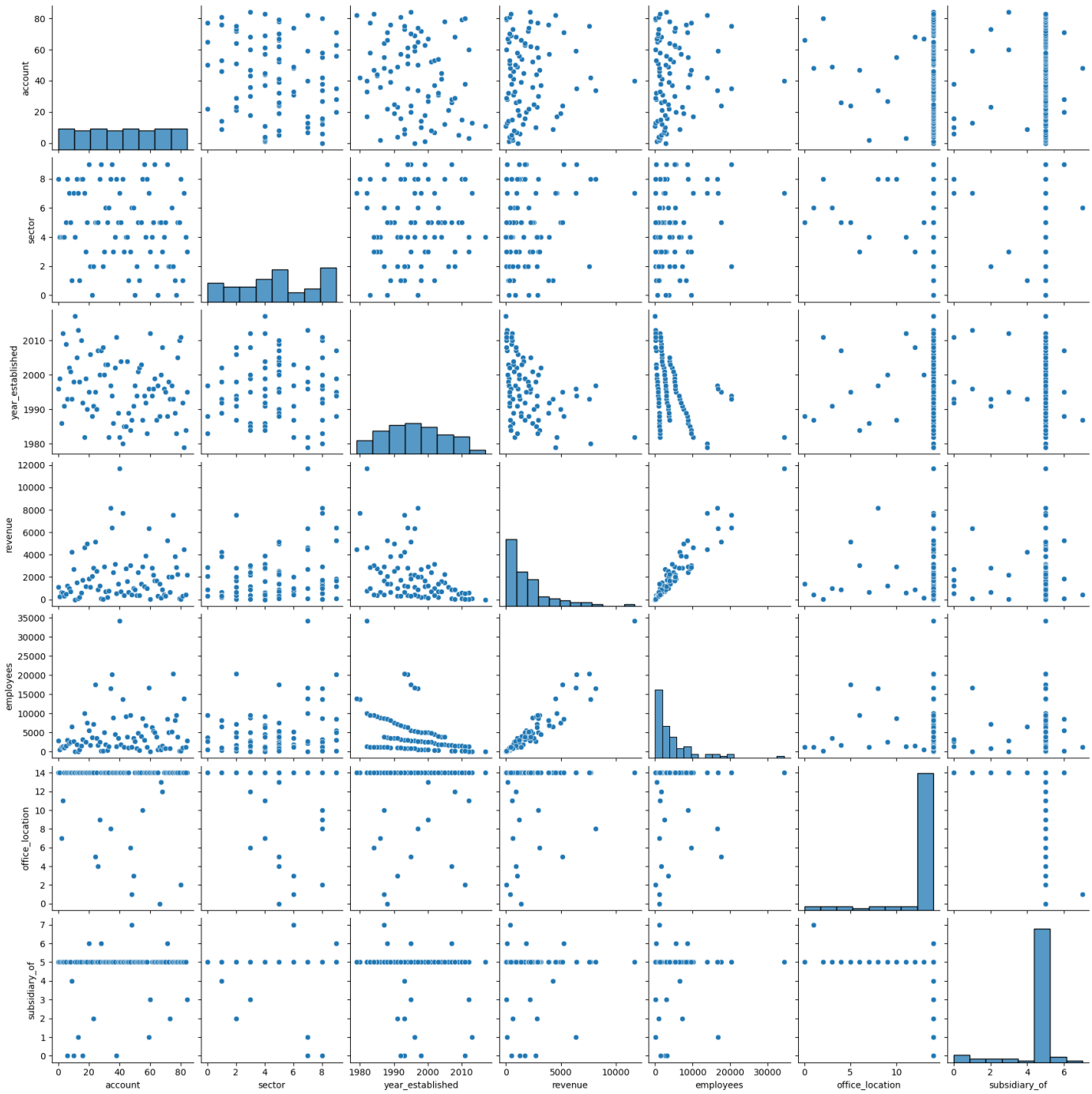

Distribution of employees in Accounts Dataset
Count
employees
0
5
10
15
20
25
30
35
0
5000
10000
15000
20000
25000
30000
35000


Distribution of office_location in Accounts Dataset
Count
office_location
0
10
20
30
40
50
60
70
0
2
4
6
8
10
12
14


Distribution of subsidiary_of in Accounts Dataset
Count
subsidiary_of
0
10
20
30
40
50
60
70
0
1
2
3
4
5
6
7




```
C:\Users\Panchin\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The f
igure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```




# FINDINGS

1. Accounts Dataset Data Summary:

Number of Records: 85 Columns: 7 (account, sector, year_established, revenue, employees, office_location, subsidiary_of) Missing Values: subsidiary_of: 70 missing values (handled by filling with 'None') Summary Statistics:

year_established: Mean: 1996.11 Std Dev: 8.87 Range: 1979 to 2017 revenue: Mean: 1994.63 Std Dev: 2169.49 Range: 4.54 to 11698.03 (in millions of USD) employees: Mean: 4660.82 Std Dev: 5715.60 Range: 9 to 34288 Visual Analysis:

Histograms: Show the distribution of year_established, revenue, and employees. Revenue and employees show right-skewed distributions, indicating a few companies have significantly higher values. Correlation

Matrix: Strong correlation between revenue and employees (high number of employees generally correlates with higher revenue). Scatter Plots:

Pairplot: Provides a visual representation of relationships between numerical variables. Confirms the correlation between revenue and employees.

1. Data Dictionary This dataset provides descriptions for fields in the other datasets. No cleaning was required, and it is primarily used to understand the meaning of each field.

2. Products Dataset Data Summary:

Number of Records: 7 Columns: 3 (product, series, sales_price) Summary Statistics:

sales_price: Mean: 2735.14 Std Dev: 2233.47 Range: 55 to 5482 Visual Analysis:

Bar Chart: Sales price distribution shows that prices vary significantly between products.

1. Sales Pipeline Dataset Data Summary:

Number of Records: 8800 Columns: 8 (opportunity_id, sales_agent, product, account, deal_stage, engage_date, close_date, close_value) Missing Values: account: 1425 missing values engage_date: 500 missing values close_date: 2089 missing values close_value: 2089 missing values Visual Analysis:

Deal Stages Distribution: Majority of deals are either Won or Lost. Close Value Distribution: Shows a significant variance, indicating a mix of high and low-value deals.

1. Sales Teams Dataset Data Summary:

Number of Records: 35 Columns: 3 (sales_agent, manager, regional_office) Visual Analysis:

Bar Chart: Distribution of sales agents across managers and regional offices. Key Findings: Accounts Dataset:

Revenue and Employees: Strong positive correlation. Larger companies (in terms of employees) tend to have higher revenue. Year Established: Most companies were established between 1989 and 2002. Products Dataset:

Wide variance in sales prices, indicating diverse product offerings. Sales Pipeline Dataset:

Significant number of missing values in account, engage_date, close_date, and close_value. Distribution of deal stages shows a clear distinction between Won and Lost deals. Sales Teams Dataset:

Distribution shows how sales agents are managed and their regional assignments.

# CLEANED DATA FOR POWER BI

In [7]:
```
# Save cleaned data for Power BI
accounts.to_csv('C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\\accoun
data_dictionary.to_csv('C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\
products.to_csv('C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\\produc
sales_pipeline.to_csv('C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\\
sales_teams.to_csv('C:\\Users\\Panchin\\Desktop\\Mar\\Personal Projects\\New folder\\sal
```

In [ ]:
```
```