Data Viz in Plotting

In [ ]:

```
import pandas as pd  # For data manipulation and analysis
import numpy as np  # For numerical computations
import matplotlib.pyplot as plt  # For creating static, animated, and interactiv
import seaborn as sns  # For statistical data visualization based on Matplotlib
import scipy  # For scientific and technical computing (including optimization,
```

In [4]:

```
import pandas as pd
import numpy as np
# Creating realistic data for employees
data = {
    'Employee ID': np.arange(1001, 1011),
    'Employee Name': ['Satender Kumar', 'data 1', 'Jane Smith', 'Robert Brown', 
    'Department': ['Data Analyst', 'IT', 'Finance', 'Marketing', 'Sales', 'Opera
    'Age': [24, np.random.randint(25, 60), np.random.randint(25, 60), np.random.
    'Location': ['London, Canada', 'Toronto', 'London', 'Sydney', 'San Francisco
    'Salary': np.random.randint(50000, 150000, size=10),
    'Years with Company': np.random.randint(1, 15, size=10),
    'Position': ['Data Analyst', 'Developer', 'Analyst', 'Designer', 'Consultant
    'Performance Score': np.random.randint(1, 5, size=10),
    'Bonus': np.random.randint(1000, 10000, size=10),
    'Gender': ['Male', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Ma
    'Marital Status': ['Single', 'Single', 'Married', 'Single', 'Single', 'Marri
    'Education': ['Bachelor', 'Master', 'PhD', 'Bachelor', 'Master', 'PhD', 'Bac
    'Hire Date': pd.to_datetime(['2019-06-12', '2015-07-23', '2012-09-05', '2018
    'Overtime Hours': np.random.randint(0, 20, size=10),
    'Sick Days Taken': np.random.randint(0, 10, size=10),
    'Vacation Days Taken': np.random.randint(5, 20, size=10),
    'Training Hours': np.random.randint(10, 50, size=10),
    'Certifications': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No'
    'Supervisor': ['Anna Smith', 'Brian Adams', 'Clara Jones', 'Daniel Martin', 
}

# Creating the DataFrame
df = pd.DataFrame(data)
```

In [10]:

```
df
```

Out[10]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position	P
0	1001	Satender Kumar	Data Analyst	24	London, Canada	50379	3	Data Analyst	
1	1002	data 1	IT	37	Toronto	98030	3	Developer	
2	1003	Jane Smith	Finance	42	London	54742	13	Analyst	
3	1004	Robert Brown	Marketing	50	Sydney	79468	7	Designer	
4	1005	Emily Davis	Sales	39	San Francisco	140576	5	Consultant	
5	1006	Michael Wilson	Operations	55	Paris	149810	2	Engineer	
6	1007	Sarah Taylor	R&D	46	Berlin	122118	1	Scientist	
7	1008	David Lee	Support	56	Tokyo	121107	11	Support Agent	
8	1009	Laura Johnson	Admin	38	Dubai	143323	8	Admin Assistant	
9	1010	James White	Legal	48	Singapore	51334	12	Lawyer	

In [11]:

```
import pandas as pd
import numpy as np
```

In [13]:

```
# Creating realistic data for a second set of employees
data1 = {
    'Employee ID': np.arange(1011, 1021),
    'Employee Name': ['Satender Kumar', 'data 1', 'Chris Evans', 'Natalie Portma
    'Department': ['Data Analyst', 'HR', 'IT', 'Marketing', 'Finance', 'Sales',
    'Age': [24, np.random.randint(25, 60), np.random.randint(25, 60), np.random.
    'Location': ['London, Canada', 'Los Angeles', 'New York', 'Chicago', 'Housto
    'Salary': np.random.randint(60000, 160000, size=10),
    'Years with Company': np.random.randint(1, 20, size=10),
    'Position': ['Data Analyst', 'HR Manager', 'IT Specialist', 'Marketing Coord
    'Performance Score': np.random.randint(1, 5, size=10),
    'Bonus': np.random.randint(2000, 12000, size=10),
    'Gender': ['Male', 'Male', 'Female', 'Female', 'Male', 'Female', 'Male', 'Fe
    'Marital Status': ['Single', 'Married', 'Single', 'Single', 'Married', 'Sing
    'Education': ['Master', 'Bachelor', 'Master', 'PhD', 'Bachelor', 'Master', '
    'Hire Date': pd.to_datetime(['2018-07-15', '2014-03-22', '2011-10-12', '2017
    'Overtime Hours': np.random.randint(0, 25, size=10),
    'Sick Days Taken': np.random.randint(0, 8, size=10),
    'Vacation Days Taken': np.random.randint(7, 22, size=10),
    'Training Hours': np.random.randint(15, 55, size=10),
    'Certifications': ['Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes
    'Supervisor': ['John Smith', 'Michael Johnson', 'Patricia Williams', 'Linda
}

# Creating the second DataFrame
df1 = pd.DataFrame(data1)
```

In [15]:

```
df1
```

Out[15]:

	Employee ID	Employee Name	Department	Age	Location	Salary	Years with Company	Position
0	1011	Satender Kumar	Data Analyst	24	London, Canada	95392	5	Data Analyst
1	1012	data 1	HR	56	Los Angeles	75471	17	HR Manager
2	1013	Chris Evans	IT	30	New York	129083	12	IT Specialist
3	1014	Natalie Portman	Marketing	27	Chicago	104039	3	Marketing Coordinator
4	1015	Tom Holland	Finance	56	Houston	138405	5	Financial Analyst
5	1016	Emma Watson	Sales	35	Phoenix	145713	19	Sales Manager
6	1017	Daniel Radcliffe	R&D	41	Philadelphia	62592	1	Research Scientist
7	1018	Scarlett Johansson	Operations	44	San Antonio	119897	10	Operations Manager
8	1019	Robert Downey Jr.	Legal	27	San Diego	92816	3	Legal Advisor
9	1020	Mark Ruffalo	Support	46	Dallas	144765	1	Support Specialist

In [17]:

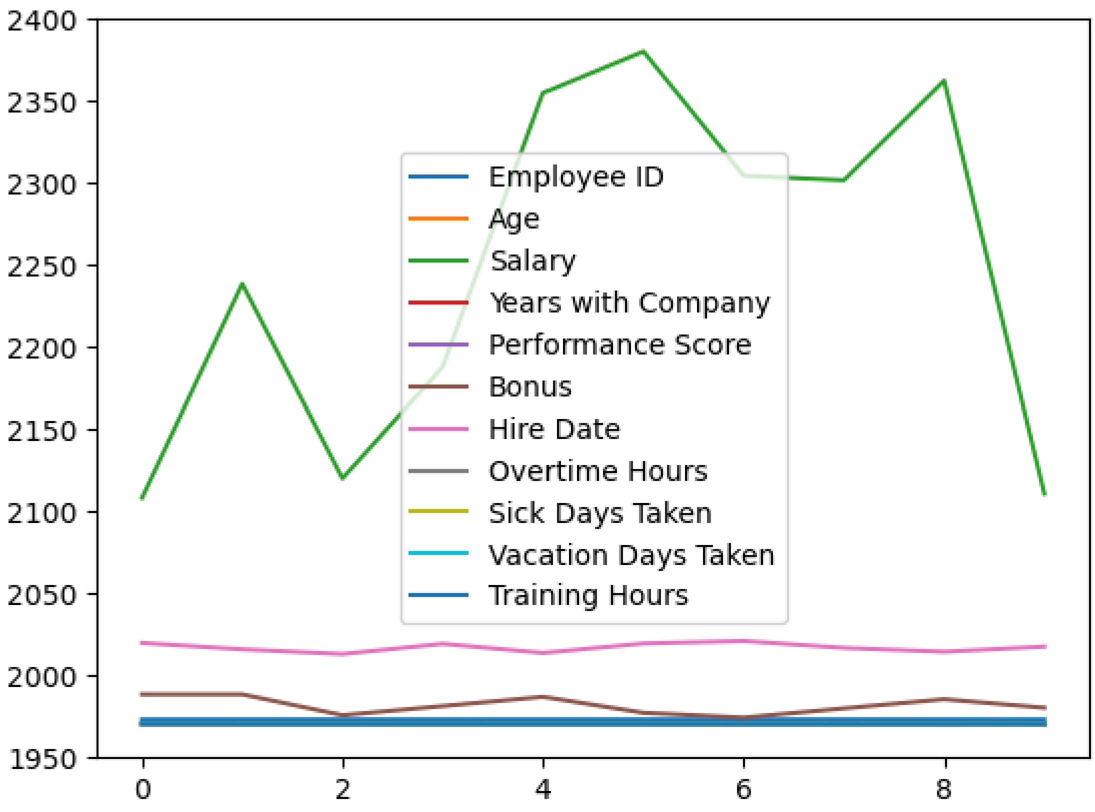
```
merged_df = pd.merge(df, df1, on='Employee ID', suffixes=('_df', '_df1'), how='c
print(merged_df)
```

```
    Employee ID Employee Name_df Department_df  Age_df     Location_df  \
0          1001   Satender Kumar  Data Analyst    24.0  London, Canada   
1          1002           data 1            IT    37.0         Toronto   
2          1003       Jane Smith       Finance    42.0          London   
3          1004     Robert Brown     Marketing    50.0          Sydney   
4          1005      Emily Davis         Sales    39.0   San Francisco   
5          1006   Michael Wilson    Operations    55.0           Paris   
6          1007     Sarah Taylor           R&D    46.0          Berlin   
7          1008        David Lee       Support    56.0           Tokyo   
8          1009    Laura Johnson         Admin    38.0           Dubai   
9          1010      James White         Legal    48.0       Singapore   
10         1011              NaN           NaN     NaN             NaN   
11         1012              NaN           NaN     NaN             NaN   
12         1013              NaN           NaN     NaN             NaN   
13         1014              NaN           NaN     NaN             NaN   
14         1015              NaN           NaN     NaN             NaN   
15         1016              NaN           NaN     NaN             NaN   
16         1017              NaN           NaN     NaN             NaN   
17         1018              NaN           NaN     NaN             NaN   
18         1019              NaN           NaN     NaN             NaN   
19         1020              NaN           NaN     NaN             NaN   

    Salary_df  Years with Company_df      Position_df  Performance Score_df  \
0     50379.0                    3.0     Data Analyst                   2.0   
1     98030.0                    3.0        Developer                   4.0   
2     54742.0                   13.0          Analyst                   3.0   
3     79468.0                    7.0         Designer                   4.0   
4    140576.0                    5.0       Consultant                   1.0   
5    149810.0                    2.0         Engineer                   2.0   
6    122118.0                    1.0        Scientist                   2.0   
7    121107.0                   11.0    Support Agent                   4.0   
8    143323.0                    8.0  Admin Assistant                   1.0   
9     51334.0                   12.0           Lawyer                   1.0   
10        NaN                    NaN              NaN                   NaN   
11        NaN                    NaN              NaN                   NaN   
12        NaN                    NaN              NaN                   NaN   
13        NaN                    NaN              NaN                   NaN   
14        NaN                    NaN              NaN                   NaN   
15        NaN                    NaN              NaN                   NaN   
16        NaN                    NaN              NaN                   NaN   
17        NaN                    NaN              NaN                   NaN   
18        NaN                    NaN              NaN                   NaN   
19        NaN                    NaN              NaN                   NaN   

    Bonus_df  ... Gender_df1 Marital Status_df1 Education_df1 Hire Date_df1  \
0     6598.0  ...        NaN                NaN           NaN           NaT   
1     6601.0  ...        NaN                NaN           NaN           NaT   
2     1951.0  ...        NaN                NaN           NaN           NaT   
3     3987.0  ...        NaN                NaN           NaN           NaT   
4     6045.0  ...        NaN                NaN           NaN           NaT   
5     2484.0  ...        NaN                NaN           NaN           NaT   
6     1421.0  ...        NaN                NaN           NaN           NaT   
7     3473.0  ...        NaN                NaN           NaN           NaT   
8     5515.0  ...        NaN                NaN           NaN           NaT   
9     3612.0  ...        NaN                NaN           NaN           NaT   
10       NaN  ...       Male             Single        Master    2018-07-15   
11       NaN  ...       Male            Married      Bachelor    2014-03-22   
12       NaN  ...     Female             Single        Master    2011-10-12   
13       NaN  ...     Female             Single           PhD    2017-04-17   
14       NaN  ...       Male            Married      Bachelor    2015-09-23   
```

```
15       NaN  ...     Female             Single        Master    2016-11-01
16       NaN  ...       Male             Single           PhD    2019-05-11
17       NaN  ...     Female            Married      Bachelor    2020-07-08
18       NaN  ...       Male             Single        Master    2013-08-19
19       NaN  ...       Male            Married           PhD    2012-01-09

    Overtime Hours_df1  Sick Days Taken_df1  Vacation Days Taken_df1  \
0                  NaN                  NaN                      NaN   
1                  NaN                  NaN                      NaN   
2                  NaN                  NaN                      NaN   
3                  NaN                  NaN                      NaN   
4                  NaN                  NaN                      NaN   
5                  NaN                  NaN                      NaN   
6                  NaN                  NaN                      NaN   
7                  NaN                  NaN                      NaN   
8                  NaN                  NaN                      NaN   
9                  NaN                  NaN                      NaN   
10                 3.0                  0.0                     18.0   
11                18.0                  2.0                     14.0   
12                16.0                  7.0                     16.0   
13                 9.0                  6.0                     11.0   
14                16.0                  1.0                     17.0   
15                 2.0                  7.0                      9.0   
16                17.0                  2.0                     21.0   
17                21.0                  2.0                     13.0   
18                 9.0                  3.0                     10.0   
19                16.0                  7.0                     21.0   

    Training Hours_df1 Certifications_df1     Supervisor_df1  
0                  NaN                NaN                NaN  
1                  NaN                NaN                NaN  
2                  NaN                NaN                NaN  
3                  NaN                NaN                NaN  
4                  NaN                NaN                NaN  
5                  NaN                NaN                NaN  
6                  NaN                NaN                NaN  
7                  NaN                NaN                NaN  
8                  NaN                NaN                NaN  
9                  NaN                NaN                NaN  
10                42.0                Yes         John Smith  
11                41.0                Yes    Michael Johnson  
12                39.0                 No  Patricia Williams  
13                36.0                Yes        Linda Brown  
14                45.0                 No      Barbara Jones  
15                51.0                Yes   Elizabeth Garcia  
16                24.0                 No     Susan Martinez  
17                37.0                Yes  Jessica Hernandez  
18                16.0                Yes        Sarah Lopez  
19                54.0                 No       Karen Wilson  

[20 rows x 39 columns]
```

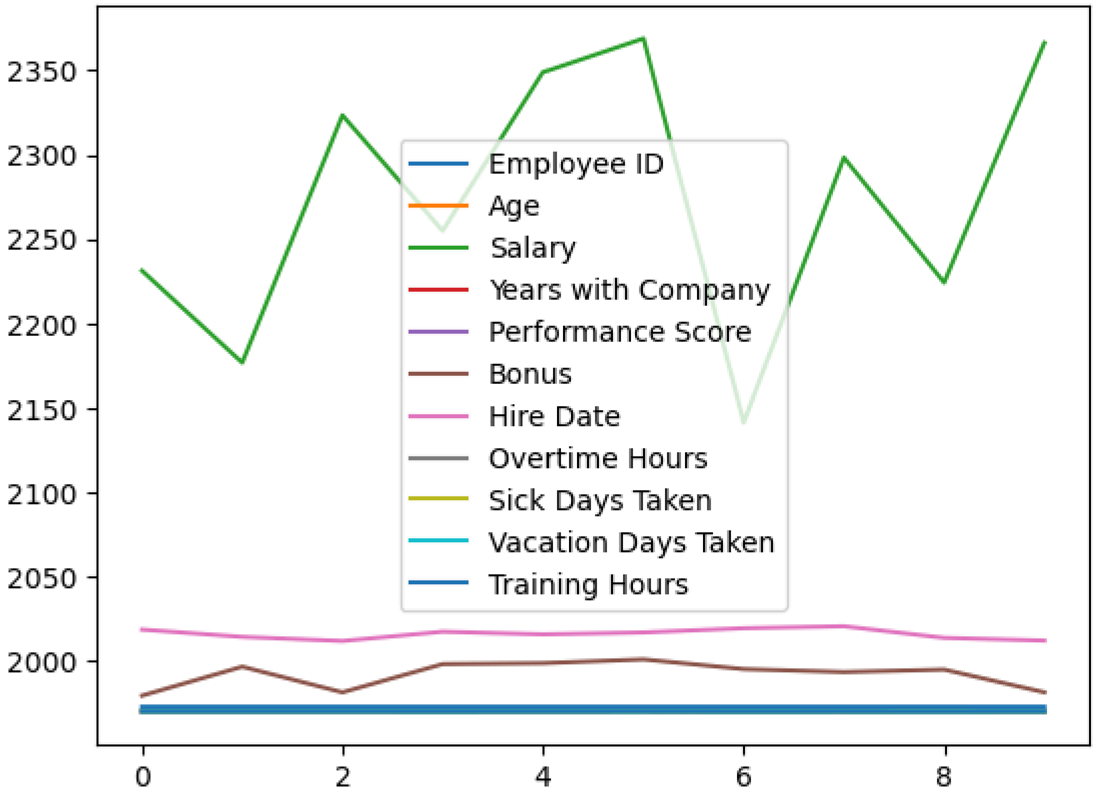
```
In [19]: df.plot()
```

```
Out[19]: <Axes: >
```

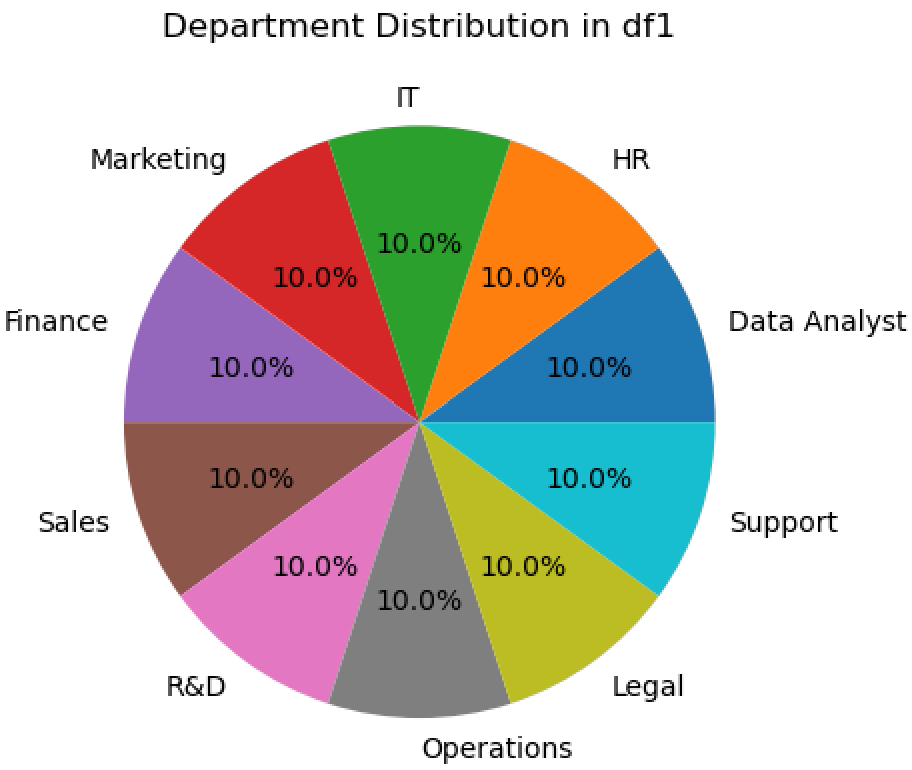


```
In [21]: df1.plot()
```

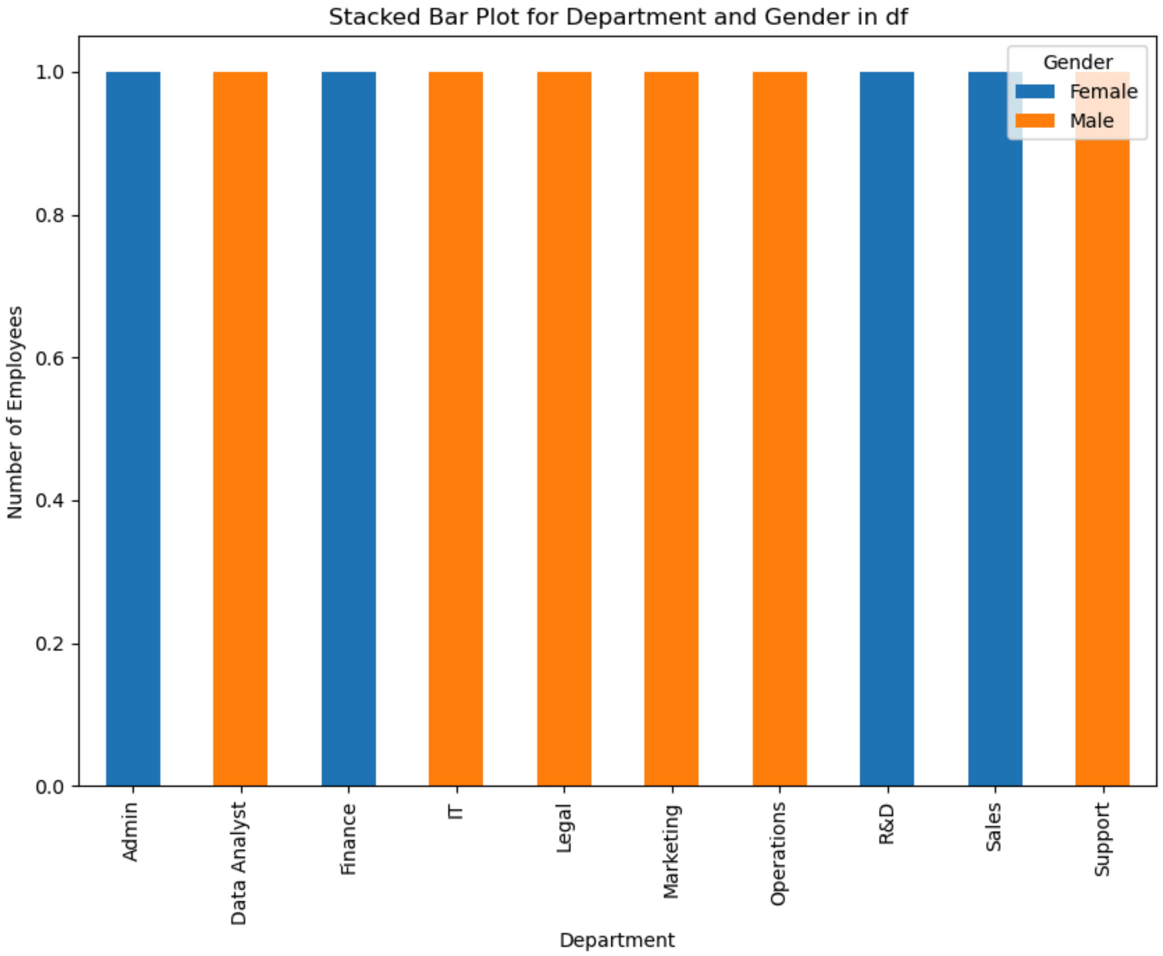
```
Out[21]: <Axes: >
```



```
In [54]: # Pie chart for department distribution in df1
         df1['Department'].value_counts().plot(kind='pie', autopct='%1.1f%%', title='Depa
         plt.ylabel('')  # Remove the y-label for a cleaner look
         plt.show()
```

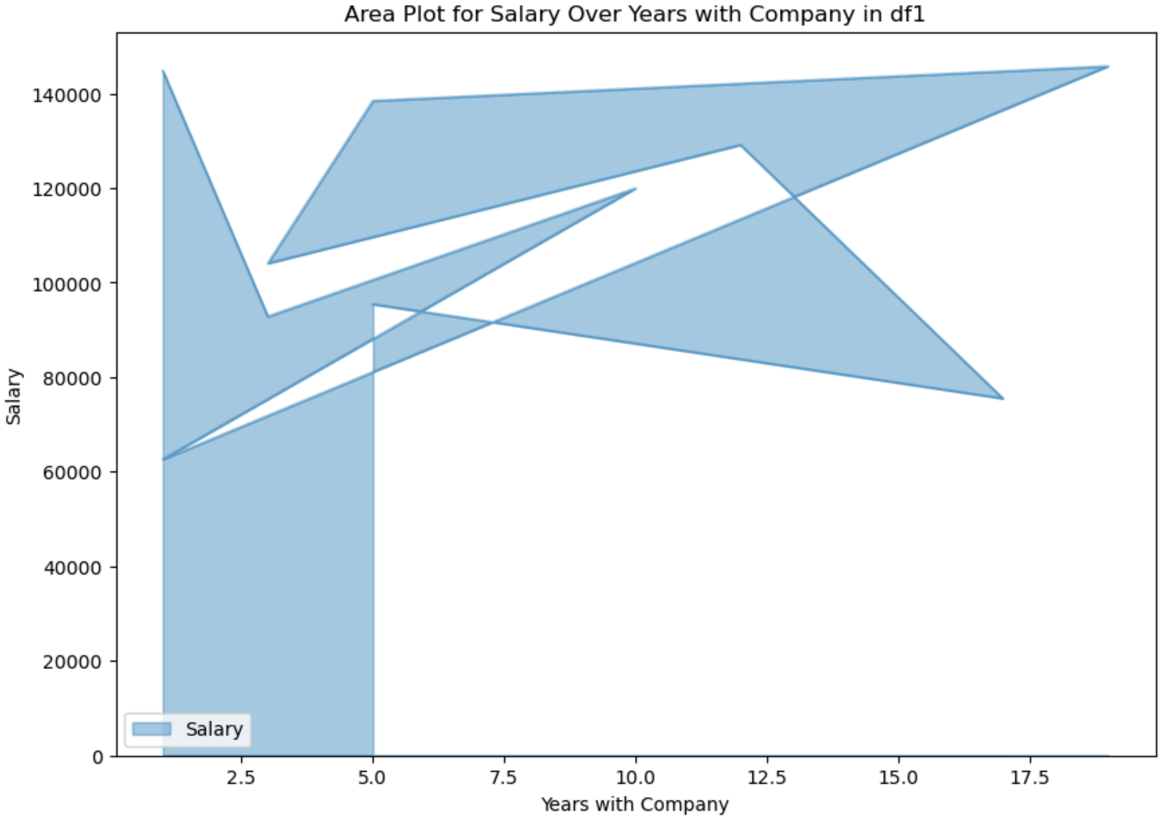


```
In [56]: # Stacked bar plot showing count of employees by Department and Gender in df
         df.groupby(['Department', 'Gender']).size().unstack().plot(kind='bar', stacked=T
         plt.title('Stacked Bar Plot for Department and Gender in df')
         plt.ylabel('Number of Employees')
         plt.show()
```



In [57]:

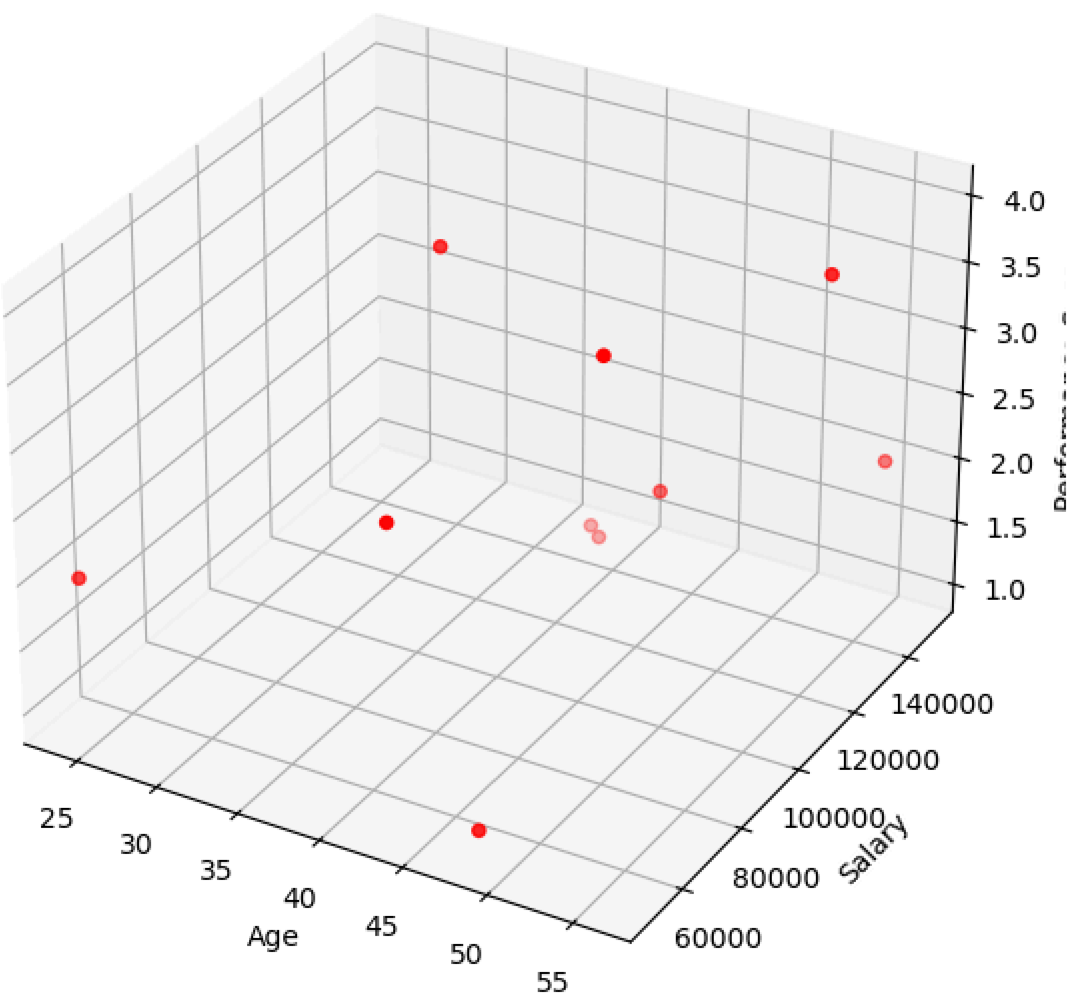
```
# Area plot showing salary over years with company in df1
df1.plot.area(x='Years with Company', y='Salary', figsize=(10, 7), alpha=0.4)
plt.title('Area Plot for Salary Over Years with Company in df1')
plt.ylabel('Salary')
plt.show()
```



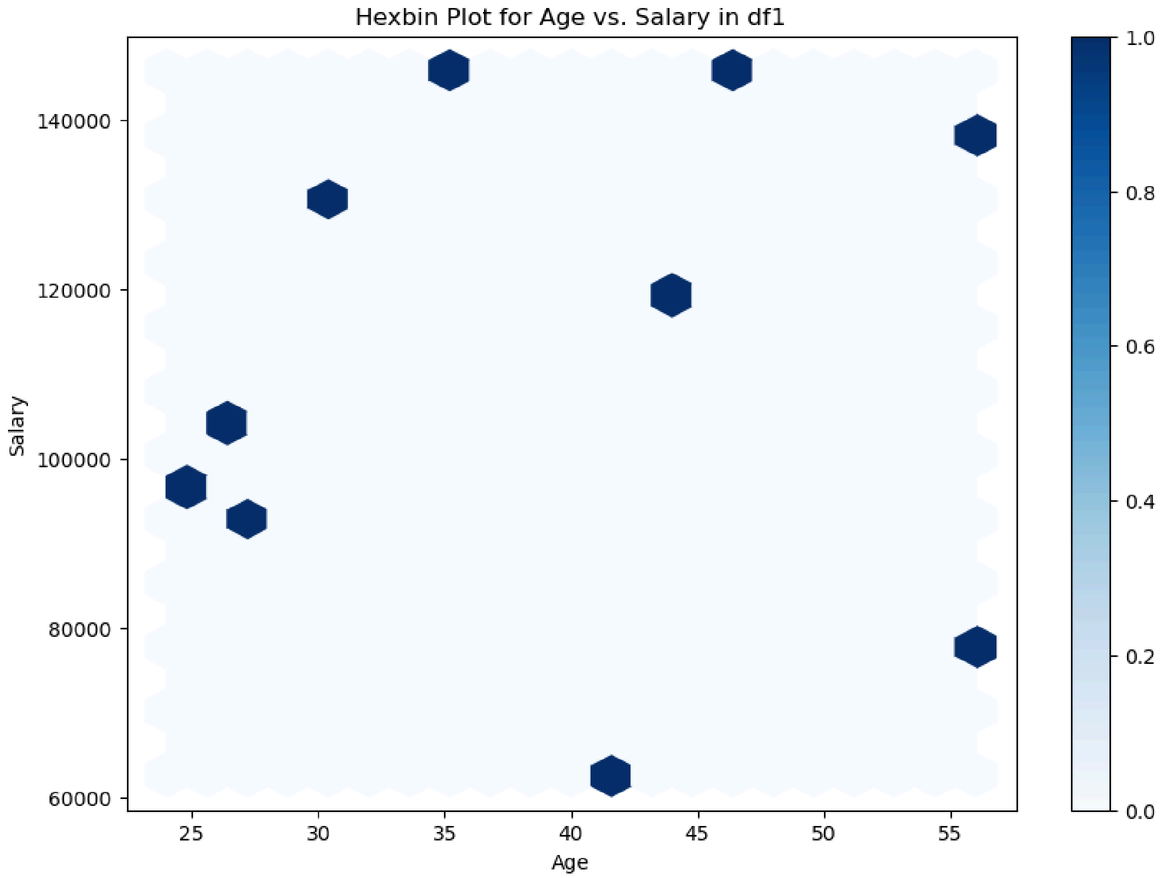
In [58]:

```
from mpl_toolkits.mplot3d import Axes3D

# 3D scatter plot in df
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df['Age'], df['Salary'], df['Performance Score'], c='r', marker='o')
ax.set_xlabel('Age')
ax.set_ylabel('Salary')
ax.set_zlabel('Performance Score')
plt.title('3D Scatter Plot in df')
plt.show()
```

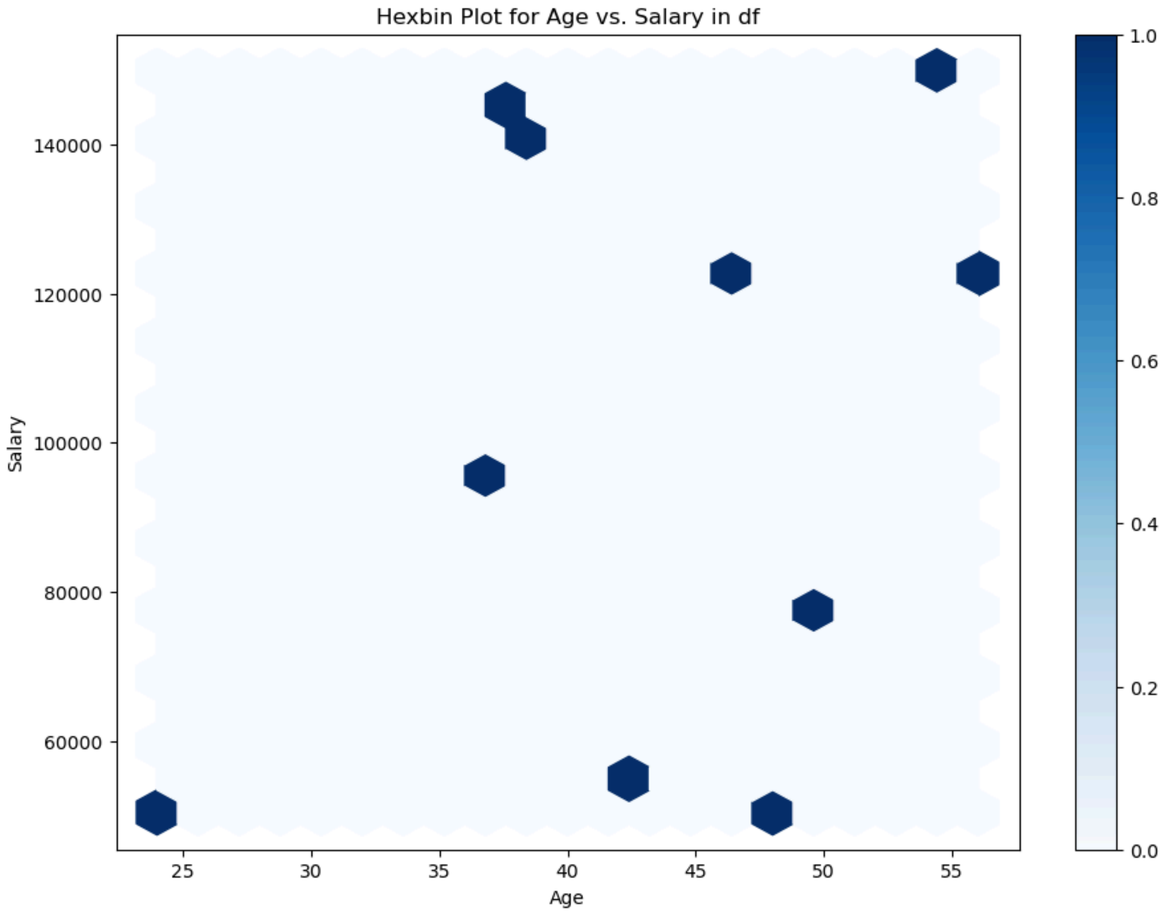
3D Scatter Plot in df



```
In [59]: # Hexbin plot for Age vs. Salary in df1
         df1.plot.hexbin(x='Age', y='Salary', gridsize=20, cmap='Blues', figsize=(10, 7))
         plt.title('Hexbin Plot for Age vs. Salary in df1')
         plt.show()
```

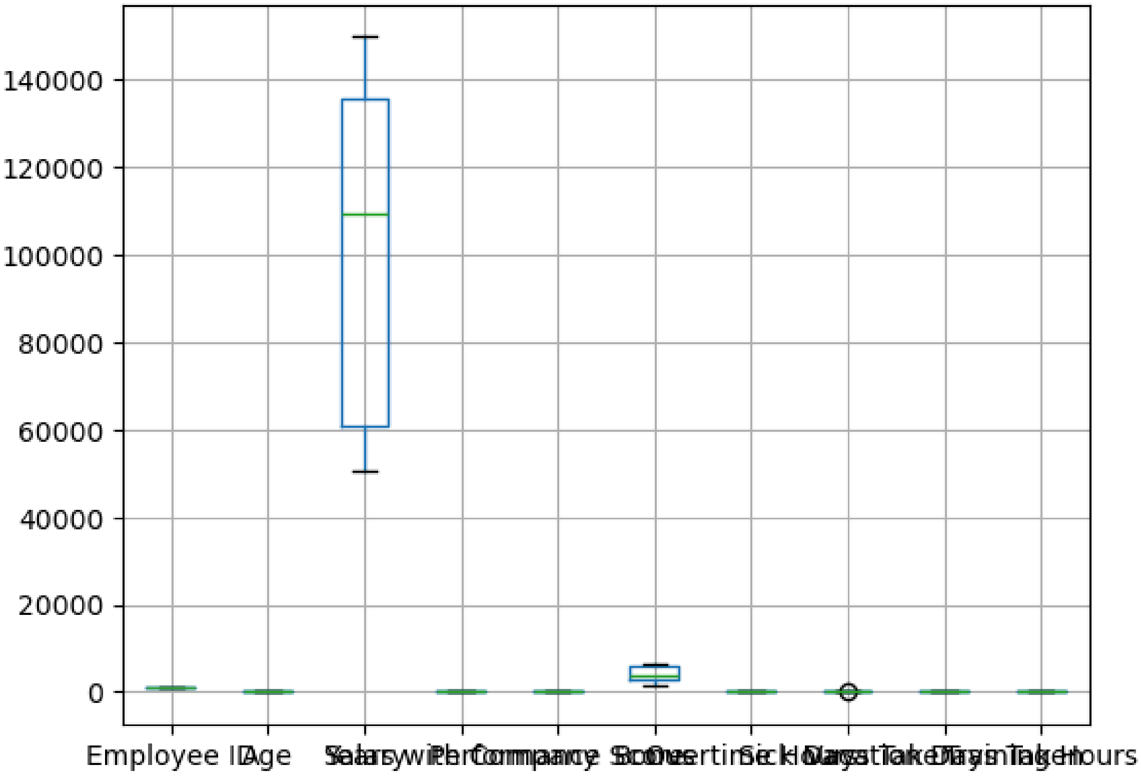


```
In [61]: # Hexbin plot for Age vs. Salary in df
         df.plot.hexbin(x='Age', y='Salary', gridsize=20, cmap='Blues', figsize=(11, 8))
         plt.title('Hexbin Plot for Age vs. Salary in df')
         plt.show()
```



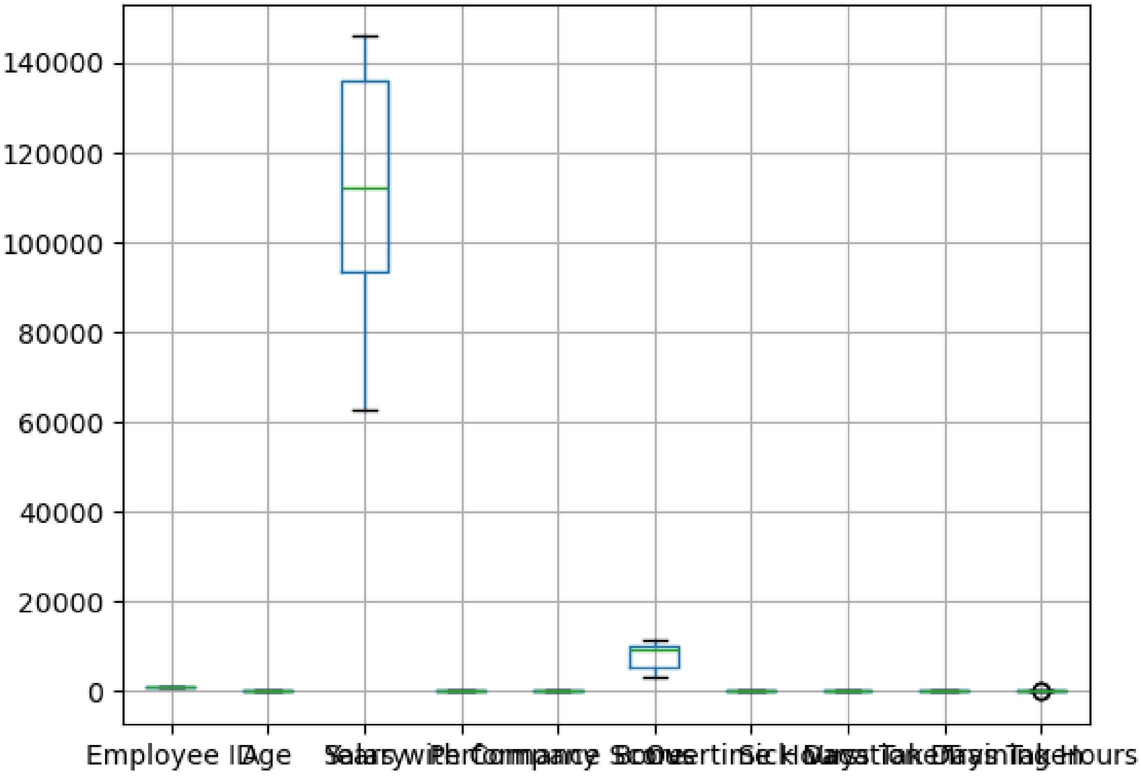
In [63]:
```
df.boxplot()
```

Out[63]:
```
<Axes: >
```



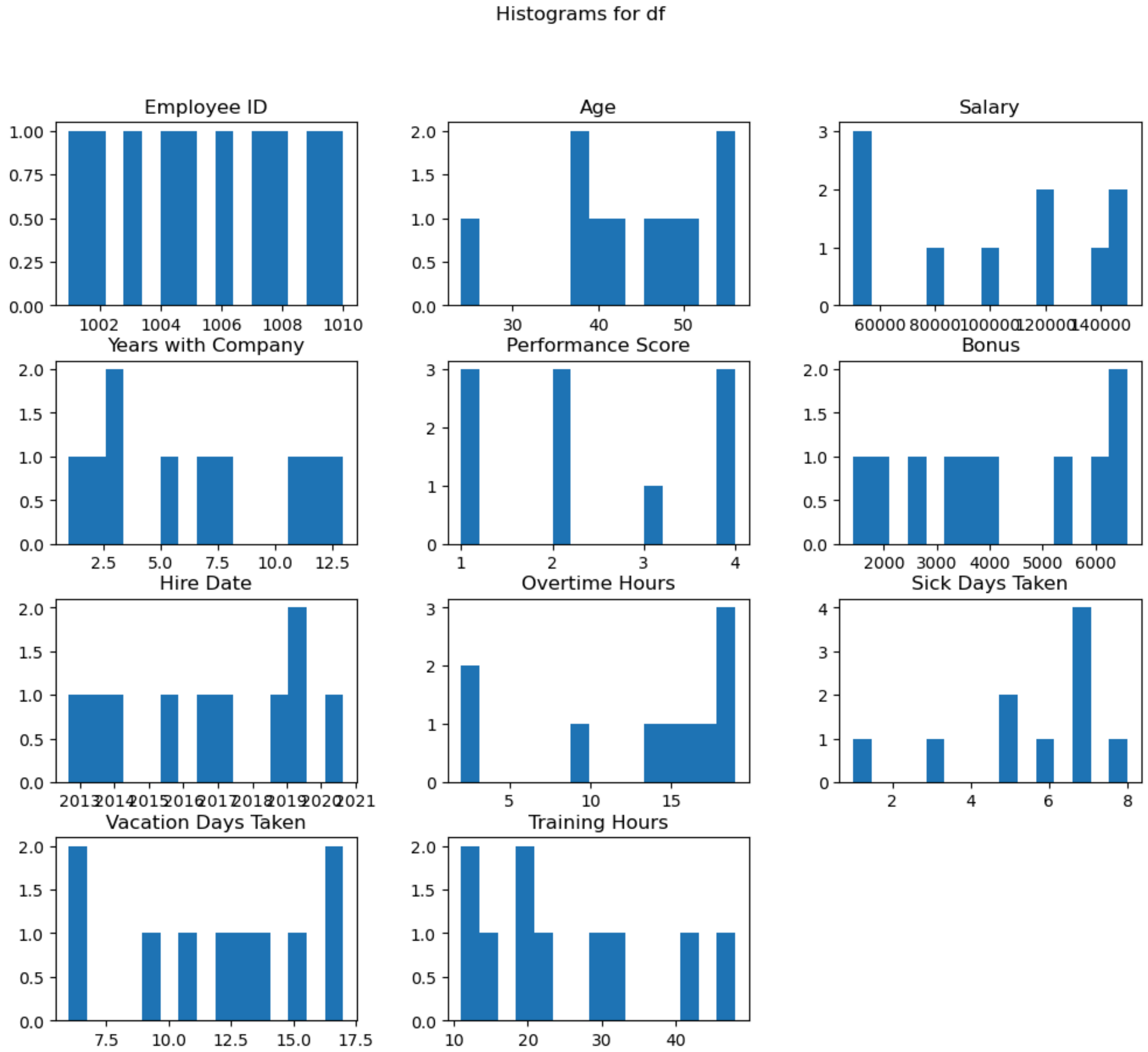
In [30]:
```
df1.boxplot()
```

Out[30]:
```
<Axes: >
```



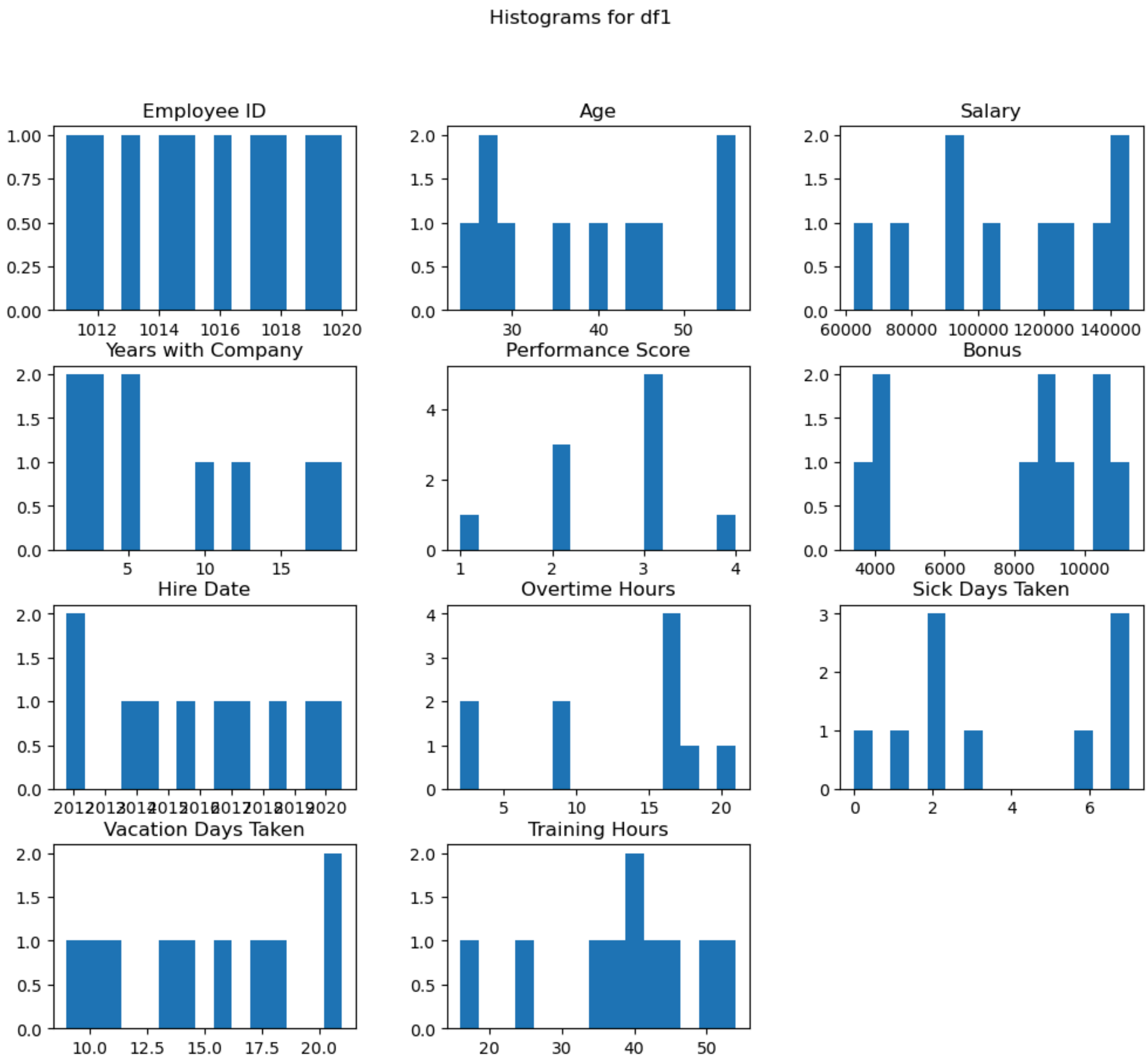
In [32]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```python
# Assuming df is your first DataFrame
df.hist(figsize=(12, 10), bins=15, grid=False)

# Display the plots for df
plt.suptitle('Histograms for df')
plt.show()
```

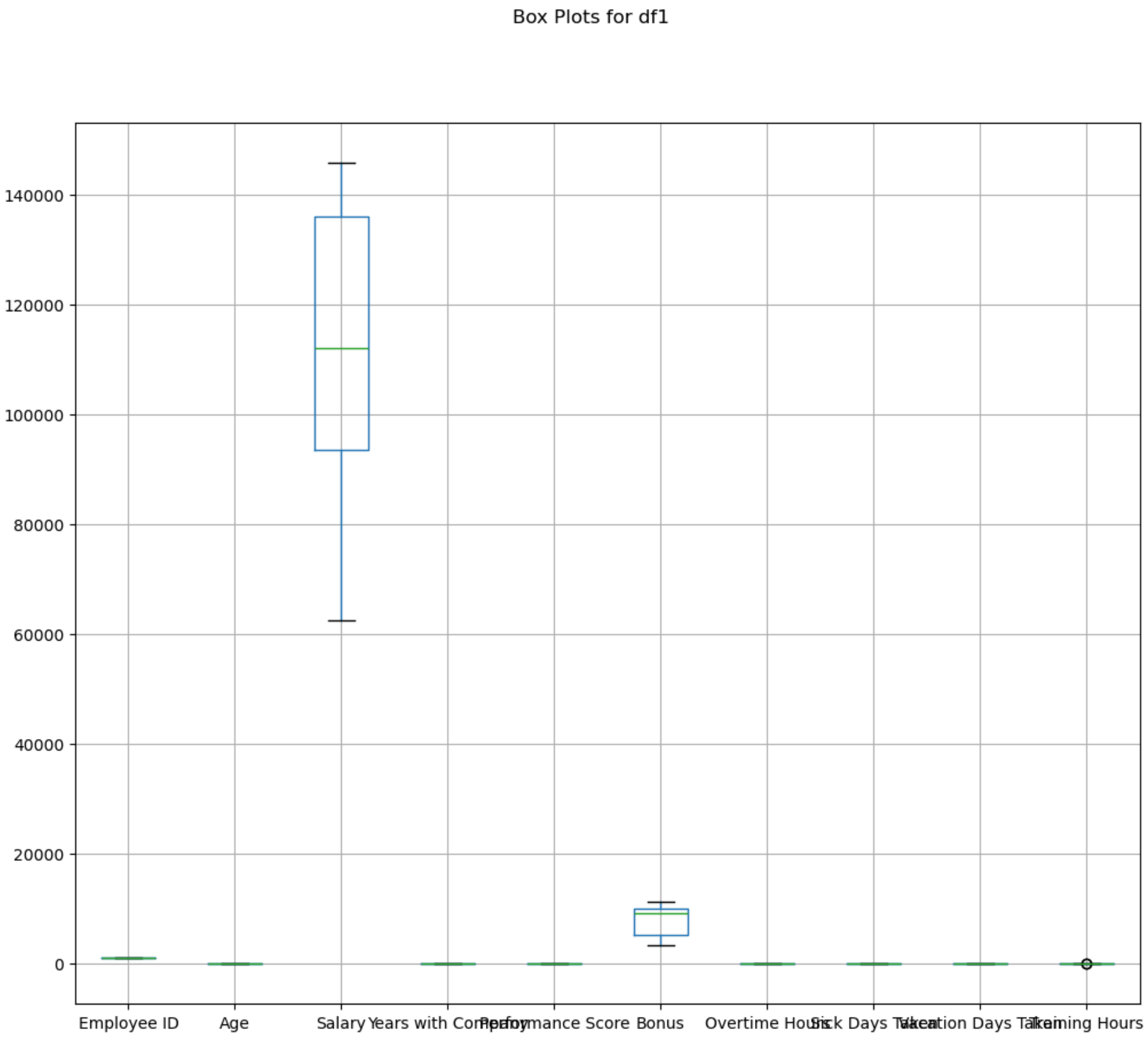


In [35]:

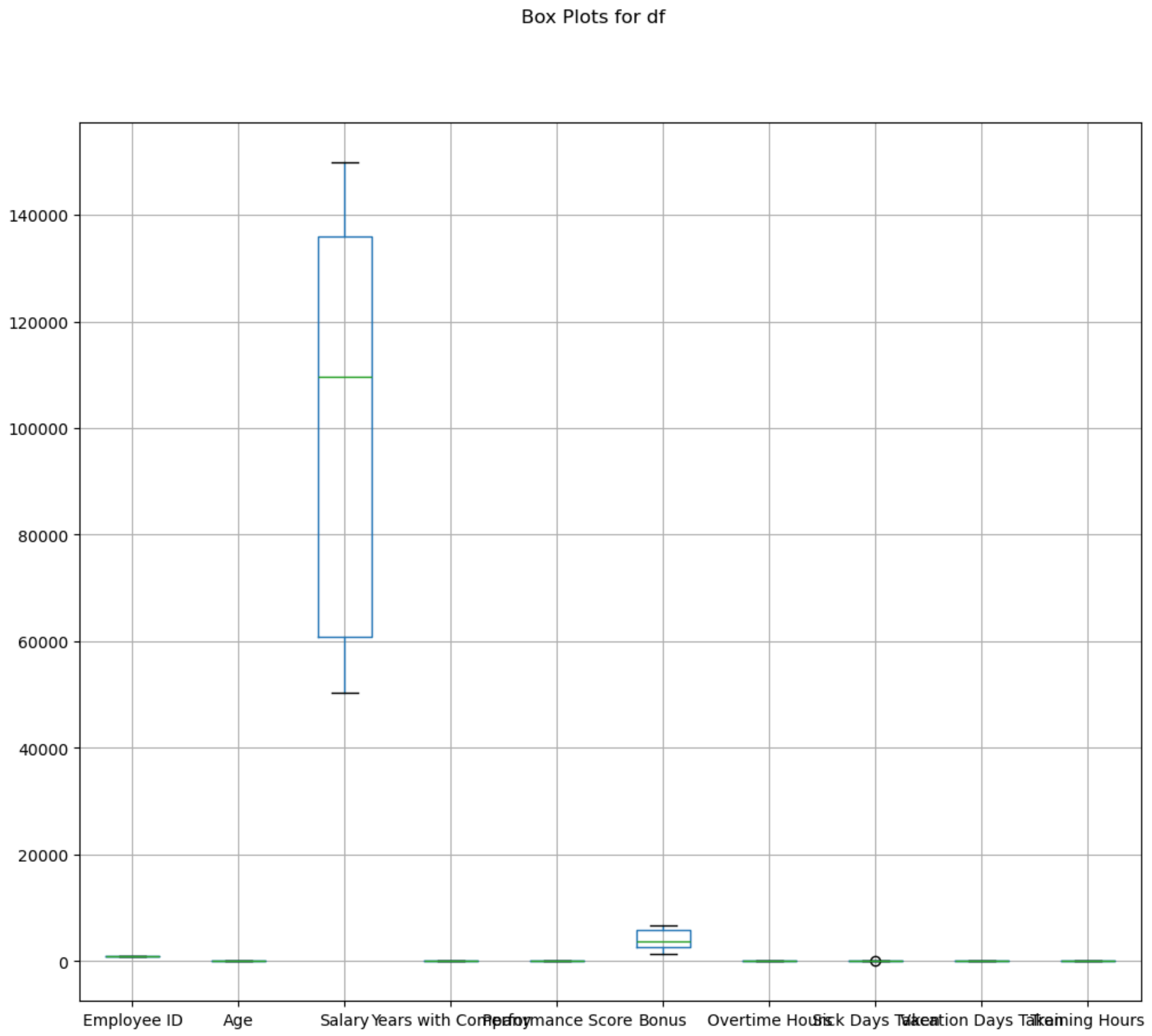
```python
# Assuming df1 is your second DataFrame
df1.hist(figsize=(12, 10), bins=15, grid=False)

# Display the plots for df1
plt.suptitle('Histograms for df1')
plt.show()
```



```
In [36]: # Box plots for df1
         df1.boxplot(figsize=(12, 10))
         plt.suptitle('Box Plots for df1')
         plt.show()
```

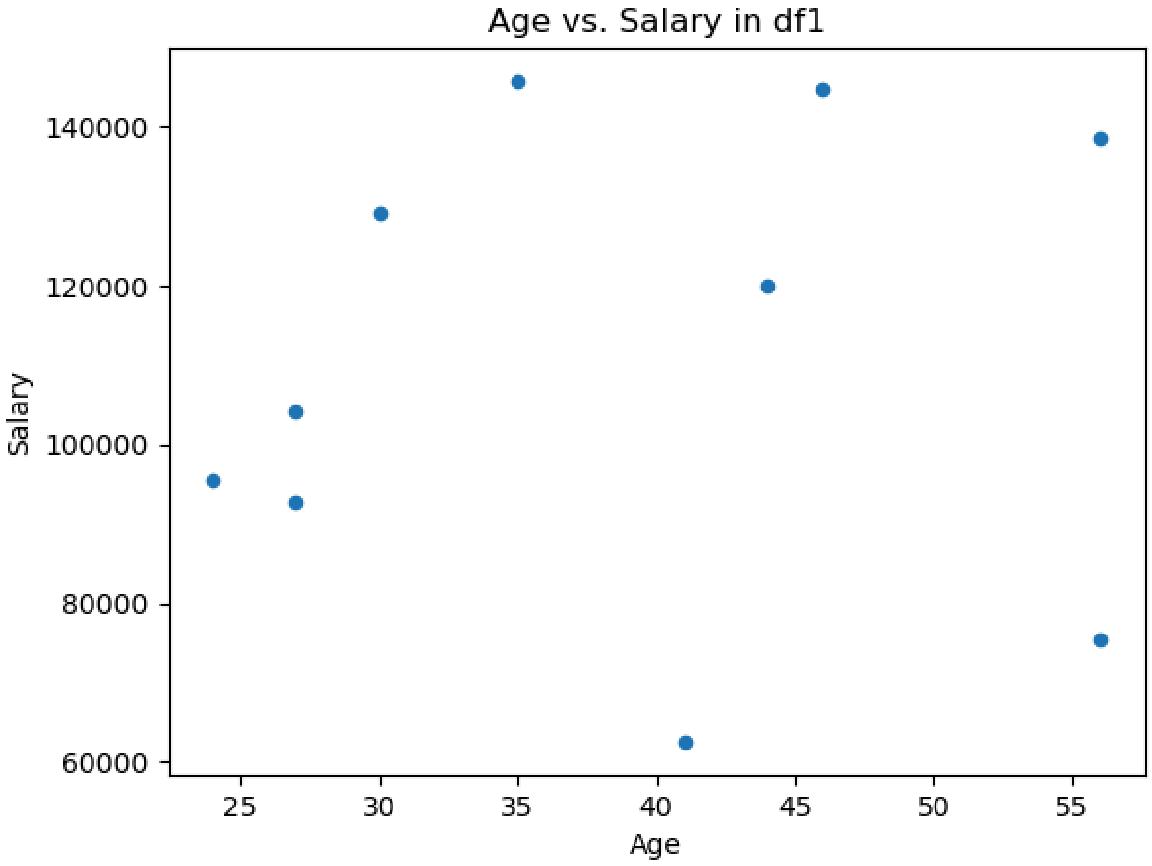


```
In [37]: # Box plots for df1
         df.boxplot(figsize=(12, 10))
         plt.suptitle('Box Plots for df')
         plt.show()
```



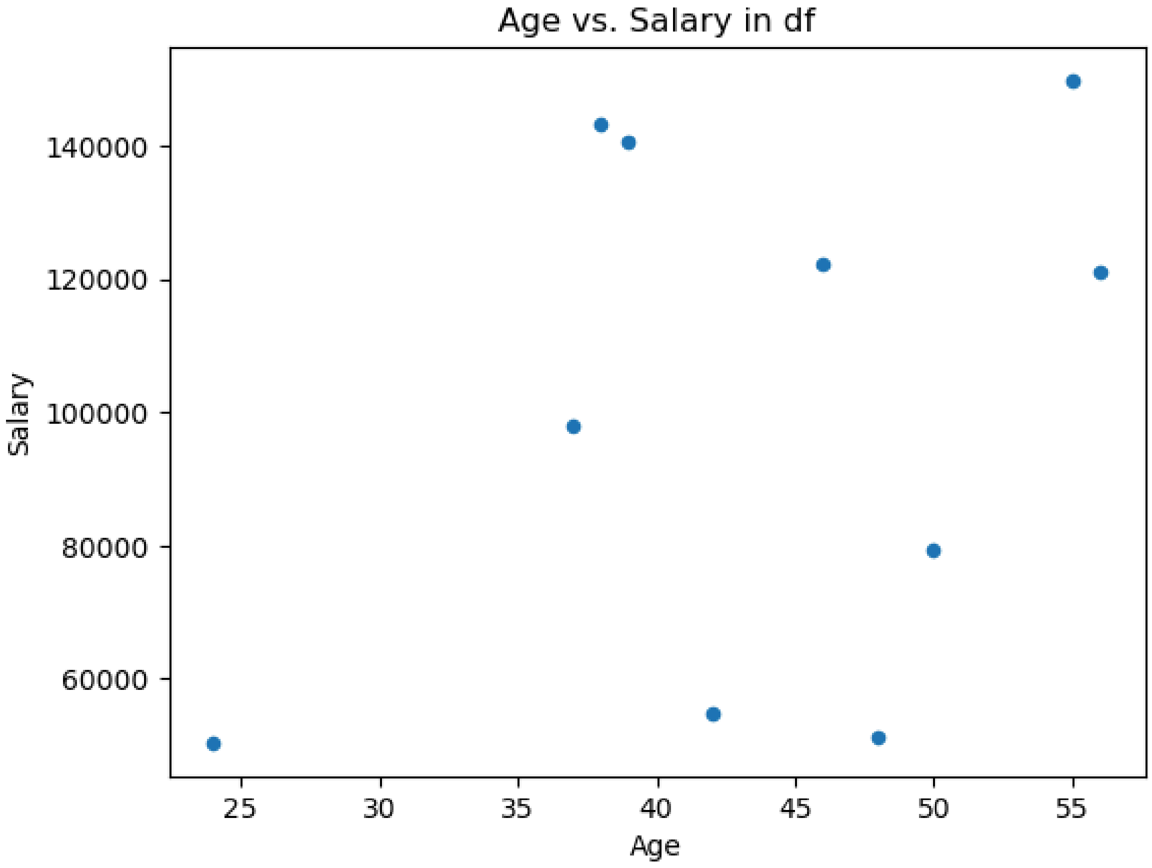
In [44]:

```
# Scatter plot between 'Age' and 'Salary' in df1
df1.plot.scatter(x='Age', y='Salary', title='Age vs. Salary in df1')
plt.show()
```



In [45]:

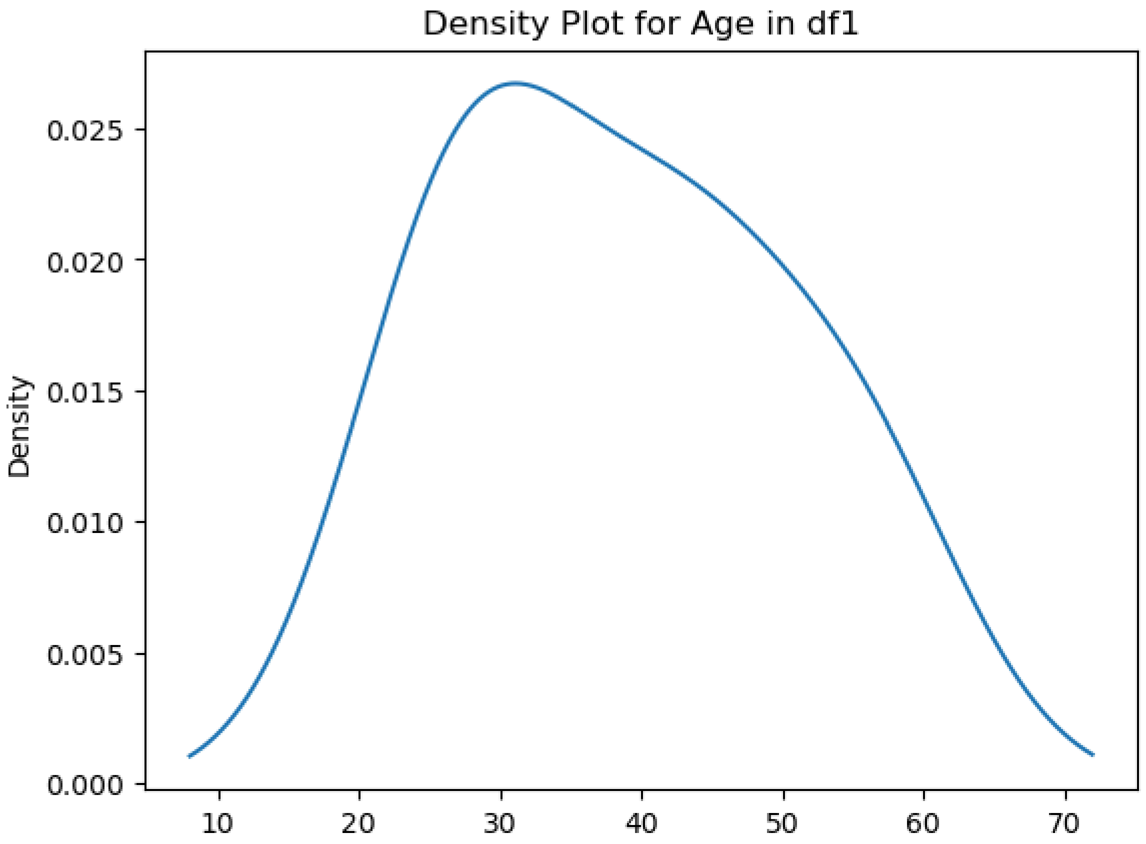
```
# Scatter plot between 'Age' and 'Salary' in df
df.plot.scatter(x='Age', y='Salary', title='Age vs. Salary in df')
plt.show()
```



In [48]:

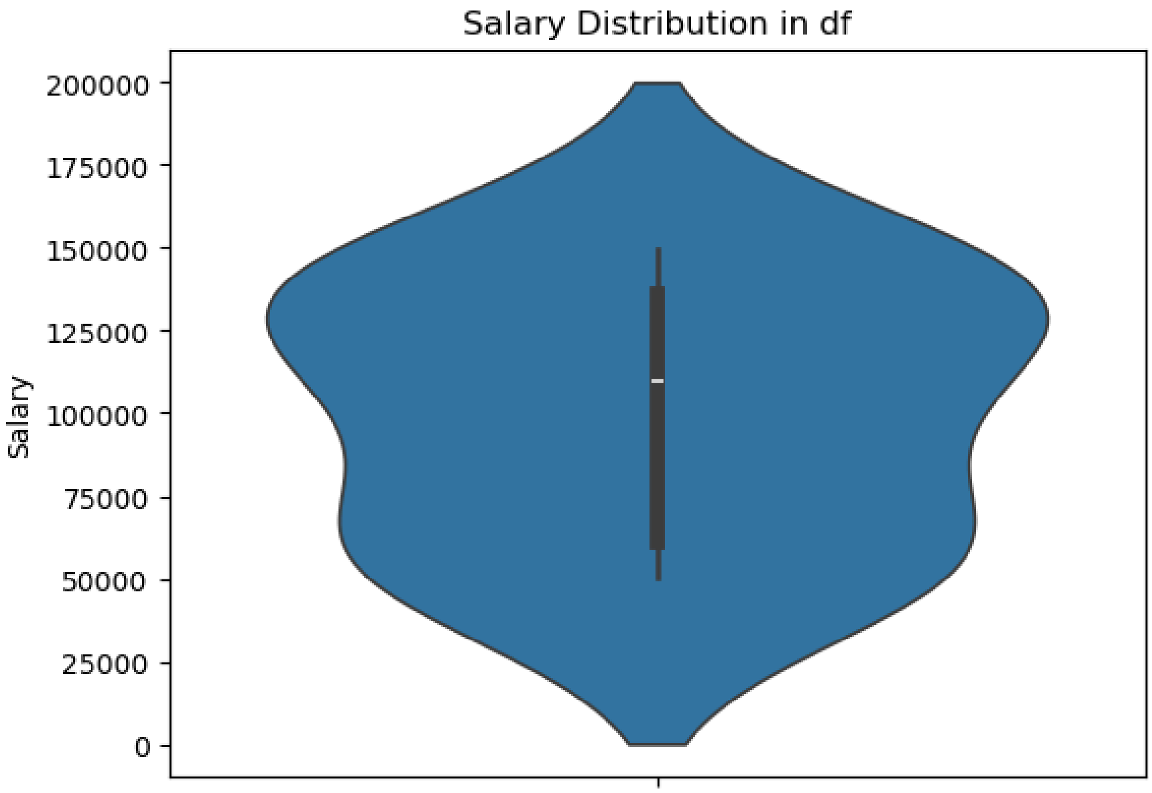
```
# Density plot for Age distribution in df1
df1['Age'].plot(kind='density', title='Density Plot for Age in df1')
```

```
plt.show()
```



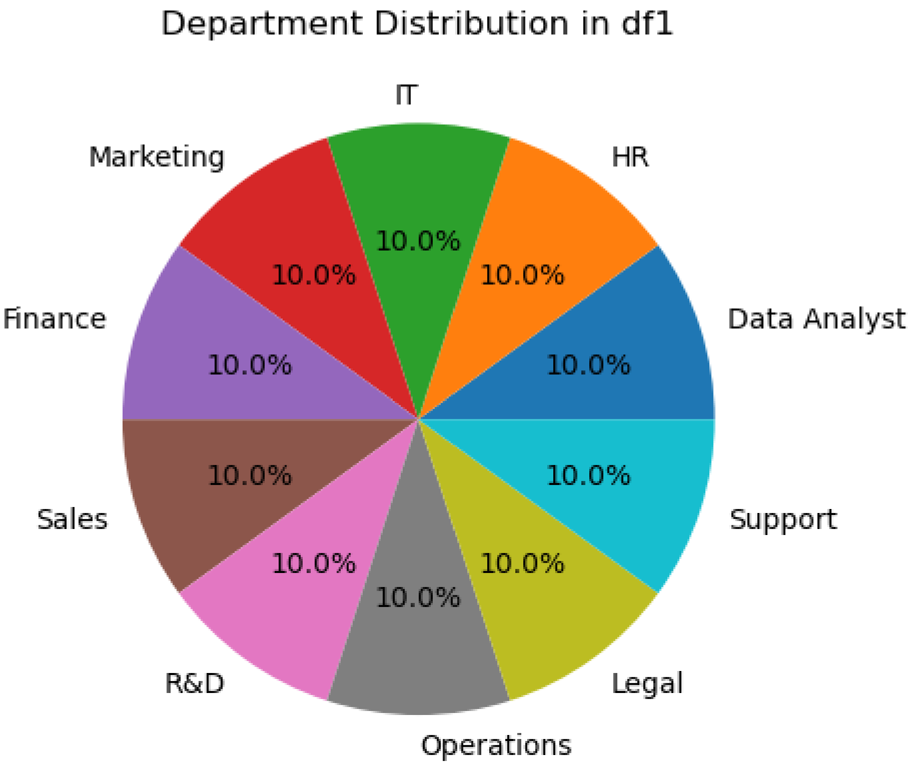
In [51]:

```
import seaborn as sns

# Violin plot for Salary distribution in df
sns.violinplot(y='Salary', data=df)
plt.title('Salary Distribution in df')
plt.show()
```



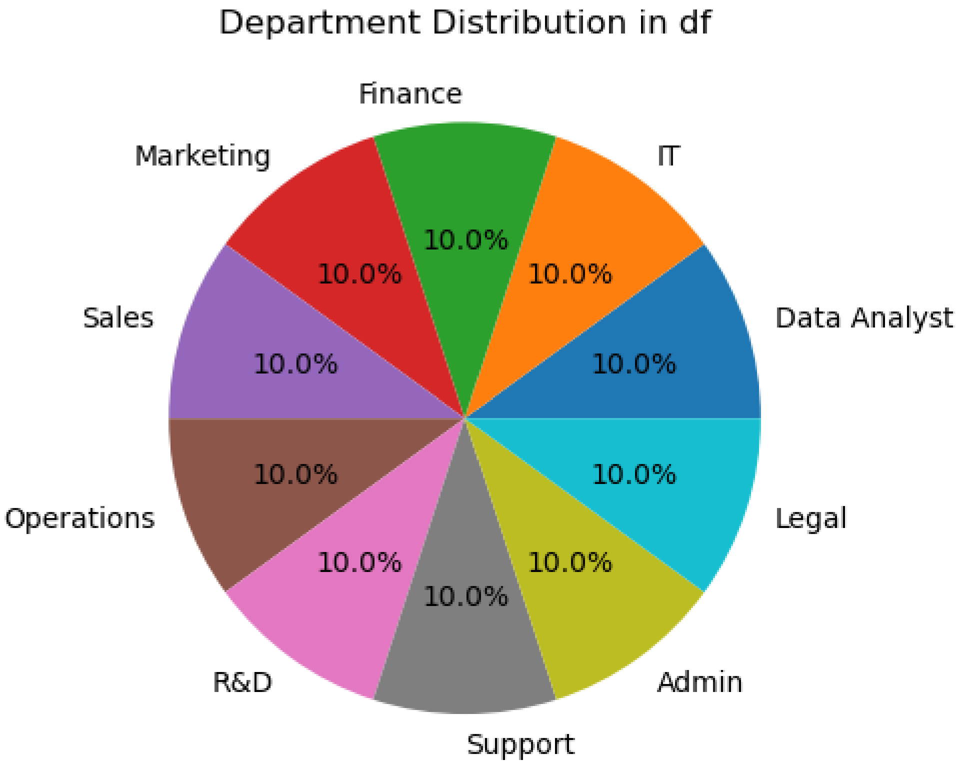
In [52]:

```
# Pie chart for department distribution in df1
df1['Department'].value_counts().plot(kind='pie', autopct='%1.1f%%', title='Depa
plt.ylabel('')  # Remove the y-label for a cleaner look
plt.show()
```



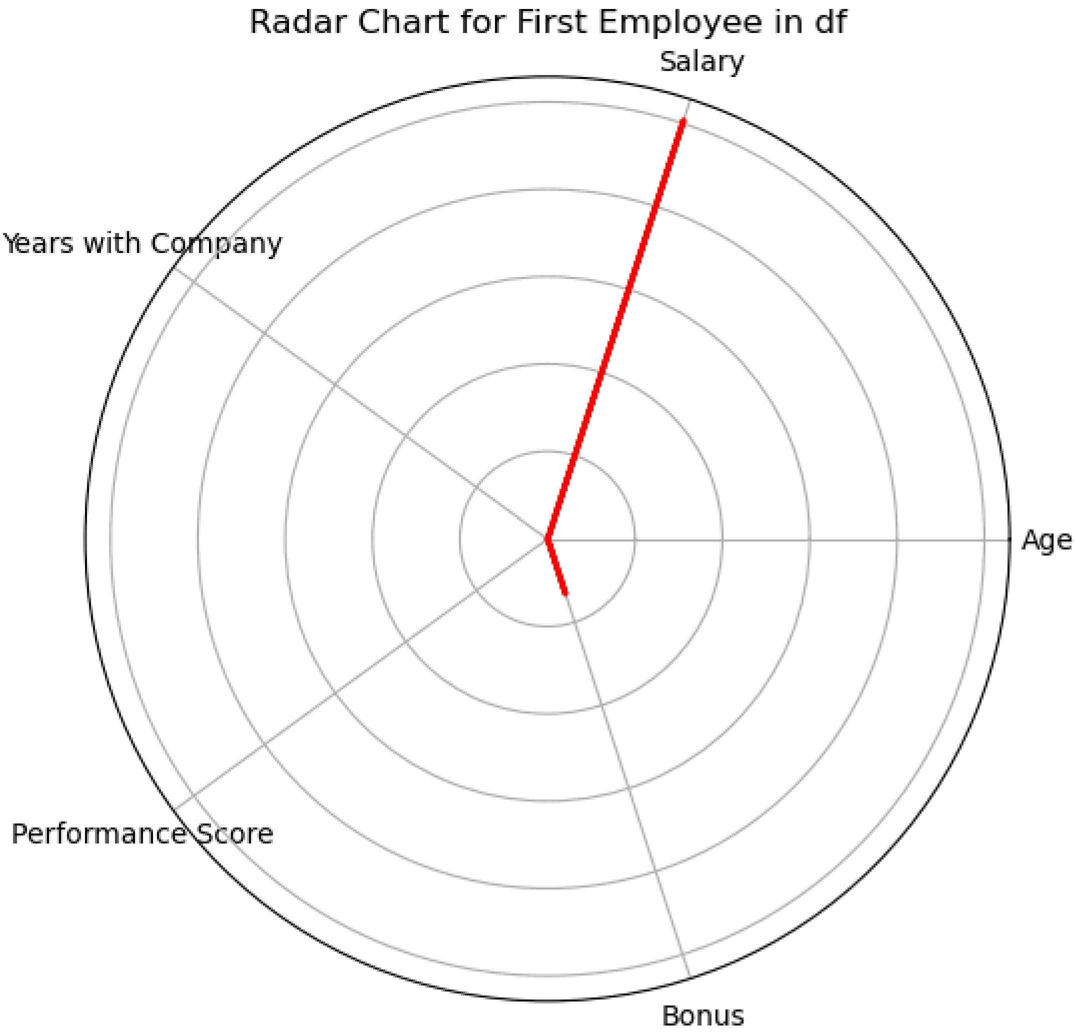
In [53]:

```
# Pie chart for department distribution in df
df['Department'].value_counts().plot(kind='pie', autopct='%1.1f%%', title='Depar
plt.ylabel('')  # Remove the y-label for a cleaner look
plt.show()
```



In [68]:

```python
import numpy as np

# Radar chart for comparing different metrics for the first employee in df
categories = ['Age', 'Salary', 'Years with Company', 'Performance Score', 'Bonus
values = df.loc[0, categories].values.flatten().tolist()

# Adding the first value to the end of the list to close the radar chart
values += values[:1]
angles = np.linspace(0, 2 * np.pi, len(categories), endpoint=False).tolist()
angles += angles[:1]

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
ax.fill(angles, values, color='red', alpha=0.25)
ax.plot(angles, values, color='red', linewidth=2)
ax.set_yticklabels([])
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)
plt.title('Radar Chart for First Employee in df')
plt.show()
```



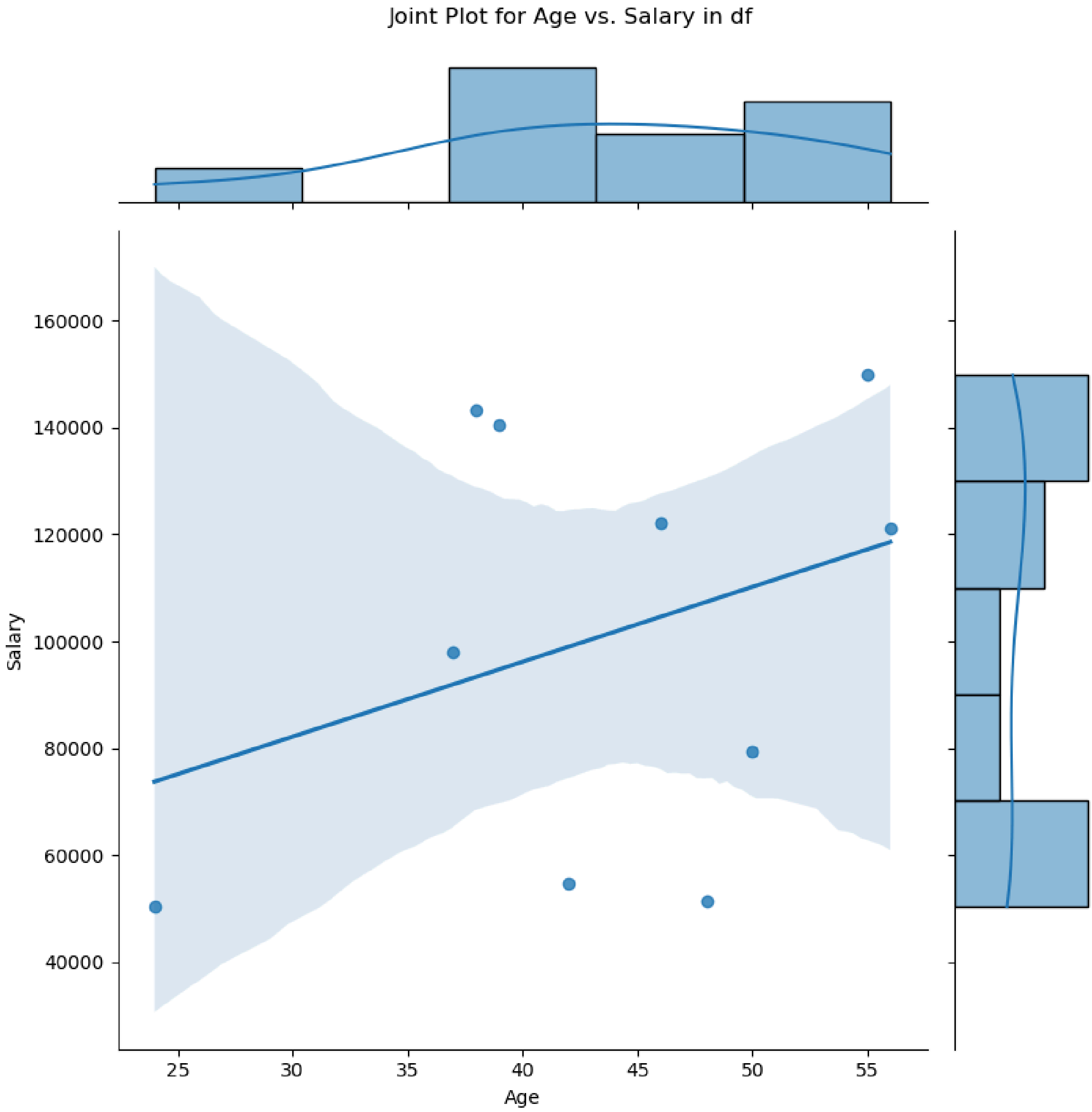
In [69]:

```python
import plotly.express as px

# Sunburst plot showing hierarchy of Department and Gender in df
fig = px.sunburst(df, path=['Department', 'Gender'], values='Salary')
fig.update_layout(title='Sunburst Plot for Department and Gender in df')
fig.show()
```

Sunburst Plot for Department and Gender in df

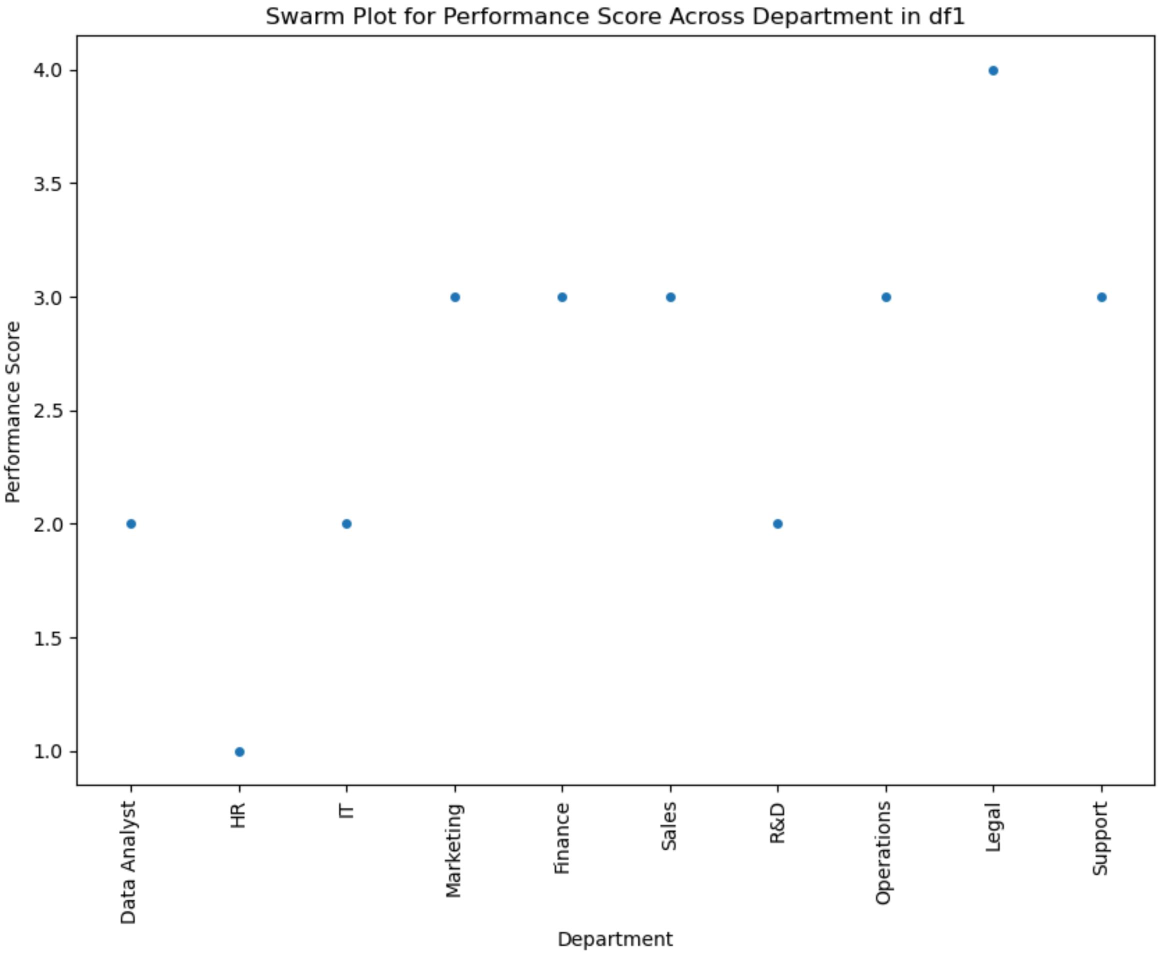
In [72]:

```
# Joint plot for Age vs. Salary in df
sns.jointplot(x='Age', y='Salary', data=df, kind='reg', height=8)
plt.suptitle('Joint Plot for Age vs. Salary in df', y=1.03)
plt.show()
```



In [73]:

```
# Swarm plot for Performance Score across Department in df1
plt.figure(figsize=(10, 7))
sns.swarmplot(x='Department', y='Performance Score', data=df1)
plt.title('Swarm Plot for Performance Score Across Department in df1')
plt.xticks(rotation=90)
plt.show()
```


Swarm Plot for Performance Score Across Department in df1
Performance Score
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Data Analyst
HR
IT
Marketing
Finance
Sales
R&D
Operations
Legal
Support
Department