In [ ]:
```
# Project Description:

# Title: Exploratory Data Analysis of Professor Salaries Dataset

# This project involves analyzing a dataset containing information about professors
# Attributes include rank, discipline, years since PhD completion, years of service
# The goal is to explore factors influencing salaries and identify trends.
# Steps include data cleaning, exploration, and analysis of factors like rank, disc
# Insights gained can inform salary policies and recruitment strategies for academi
```

In [4]:
```
import pandas as pd
df = pd.read_csv('Salaries.csv')
```

In [5]:
```
print(df.iloc[[45]]) # to print particular ROW only
```

```
    RANK DISCIPLINE   PHd  SERVICE     SEX    SALARY
45  Prof          B  25.0       25  Female  140096.0
```

In [6]:
```
print(df.to_string()) #use to_string() to print the entire DataFrame.
```

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
6	AssocProf	A	20.0	17	Male	81285.0
7	Prof	A	18.0	18	Male	NaN
8	Prof	A	29.0	19	Male	94350.0
9	Prof	A	51.0	51	Male	57800.0
10	Prof	B	39.0	33	Male	128250.0
11	Prof	B	23.0	23	Male	134778.0
12	AsstProf	B	1.0	0	Male	88000.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
16	AsstProf	B	8.0	3	Male	75044.0
17	AsstProf	B	4.0	0	Male	92000.0
18	Prof	A	19.0	7	Male	107300.0
19	Prof	A	29.0	27	Male	150500.0
20	AsstProf	B	4.0	4	Male	92000.0
21	Prof	A	33.0	30	Male	103106.0
22	AsstProf	A	4.0	2	Male	73000.0
23	AsstProf	A	2.0	0	Male	85000.0
24	Prof	A	30.0	23	Male	91100.0
25	Prof	B	35.0	31	Male	99418.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
28	AsstProf	B	7.0	2	Male	NaN
29	Prof	B	21.0	20	Male	123683.0
30	AssocProf	B	9.0	7	Male	107008.0
31	Prof	B	22.0	21	Male	155750.0
32	Prof	A	27.0	19	Male	103275.0
33	Prof	B	18.0	18	Male	120000.0
34	AssocProf	B	NaN	8	Male	119800.0
35	Prof	B	28.0	23	Male	126933.0
36	Prof	B	45.0	45	Male	146856.0
37	Prof	A	20.0	8	Male	102000.0
38	AsstProf	B	4.0	3	Male	91000.0
39	Prof	B	18.0	18	Female	129000.0
40	Prof	A	39.0	36	Female	137000.0
41	AssocProf	A	13.0	8	Female	74830.0
42	AsstProf	B	4.0	2	Female	80225.0
43	AsstProf	B	5.0	0	Female	77000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
46	AsstProf	B	11.0	3	Female	74692.0
47	AssocProf	B	11.0	11	Female	103613.0
48	Prof	B	17.0	17	Female	111512.0
49	Prof	B	17.0	18	Female	122960.0
50	AsstProf	B	10.0	5	Female	97032.0
51	Prof	B	20.0	14	Female	127512.0
52	Prof	A	12.0	0	Female	105000.0
53	AsstProf	A	5.0	3	Female	73500.0
54	AssocProf	A	25.0	22	Female	62884.0
55	AsstProf	A	2.0	0	Female	72500.0
56	AssocProf	A	10.0	8	Female	77500.0
57	AsstProf	A	3.0	1	Female	72500.0
58	Prof	B	36.0	26	Female	144651.0
59	AssocProf	B	12.0	10	Female	103994.0
60	AsstProf	B	3.0	3	Female	92000.0
61		B	13.0	10	Female	103750.0
62	AssocProf	B	14.0	7	Female	109650.0

```
63       Prof          A  29.0       27  Female   91000.0
64  AssocProf          A  26.0       24  Female   73300.0
65       Prof          A  36.0       19  Female  117555.0
66   AsstProf          A   7.0        6  Female   63100.0
67       Prof          A  17.0       11  Female   90450.0
68   AsstProf          A   4.0        2  Female   77500.0
69       Prof          A  28.0        7  Female  116450.0
70   AsstProf          A   8.0        3  Female   78500.0
71  AssocProf          B  12.0        9  Female   71065.0
72       Prof          B  24.0       15  Female  161101.0
73       Prof          B  18.0       10  Female  105450.0
74  AssocProf          B  19.0        6  Female  104542.0
75       Prof          B  17.0       17  Female  124312.0
76       Prof          A  28.0       14  Female  109954.0
77       Prof          A  23.0       15  Female  109646.0
```

In [7]:
```
import pandas as pd
df = pd.read_csv('Salaries.csv')
print(df) #If you have a large DataFrame with many rows, Pandas will only return th
```

```
        RANK DISCIPLINE   PHd  SERVICE     SEX    SALARY
0       Prof          B  56.0       49    Male  186960.0
1       Prof          A  12.0        6    Male   93000.0
2       Prof          A  23.0       20    Male  110515.0
3       Prof          A  40.0       31    Male  131205.0
4       Prof          B  20.0       18    Male  104800.0
..       ...        ...   ...      ...     ...       ...
73      Prof          B  18.0       10  Female  105450.0
74 AssocProf          B  19.0        6  Female  104542.0
75      Prof          B  17.0       17  Female  124312.0
76      Prof          A  28.0       14  Female  109954.0
77      Prof          A  23.0       15  Female  109646.0

[78 rows x 6 columns]
```

In [8]:
```
#The number of rows returned is defined in Pandas option settings. (by default it i
#To check our system's maximum rows with the pd.options.display.max_rows statement.

import pandas as pd
print(pd.options.display.max_rows)
```

```
60
```

In [9]:
```
#To increase/change the maximum number of rows to display the entire DataFrame:

import pandas as pd
pd.options.display.max_rows = 50
df = pd.read_csv('Salaries.csv')

print(df)
```

```
        RANK DISCIPLINE   PHd  SERVICE     SEX    SALARY
0       Prof          B  56.0       49    Male  186960.0
1       Prof          A  12.0        6    Male   93000.0
2       Prof          A  23.0       20    Male  110515.0
3       Prof          A  40.0       31    Male  131205.0
4       Prof          B  20.0       18    Male  104800.0
..       ...        ...   ...      ...     ...       ...
73      Prof          B  18.0       10  Female  105450.0
74 AssocProf          B  19.0        6  Female  104542.0
75      Prof          B  17.0       17  Female  124312.0
76      Prof          A  28.0       14  Female  109954.0
77      Prof          A  23.0       15  Female  109646.0
```



```
[78 rows x 6 columns]
```

In [10]:
```
import pandas as pd
print(pd.options.display.max_rows)
```

```
50
```

In [11]:
```
df.shape #To show rows, cols
```

Out[11]:
```
(78, 6)
```

In [12]:
```
df.ndim #To show dimension (2D or 3D or ... ) here row and column i.e. 2D
```

Out[12]:
```
2
```

In [13]:
```
df.columns #To show all the columns names.
```

Out[13]:
```
Index(['RANK', 'DISCIPLINE', 'PHd', 'SERVICE', 'SEX', 'SALARY'], dtype='object')
```

In [14]:
```
df.columns.tolist() #To make a list of all the columns names.
```

Out[14]:
```
['RANK', 'DISCIPLINE', 'PHd', 'SERVICE', 'SEX', 'SALARY']
```

In [15]:
```
df.head() #To show top 5 values/rows/tuples in table format.
```

Out[15]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0

In [16]:
```
df.head(10)
```

Out[16]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
5	Prof	A	20.0	20	Male	122400.0
6	AssocProf	A	20.0	17	Male	81285.0
7	Prof	A	18.0	18	Male	NaN
8	Prof	A	29.0	19	Male	94350.0
9	Prof	A	51.0	51	Male	57800.0

In [17]:
```
df.tail() #To show bottom 5 values/rows/tuples in table format.
```

Out[17]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

In [18]:
```
df.tail(10)
```

Out[18]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
68	AsstProf	A	4.0	2	Female	77500.0
69	Prof	A	28.0	7	Female	116450.0
70	AsstProf	A	8.0	3	Female	78500.0
71	AssocProf	B	12.0	9	Female	71065.0
72	Prof	B	24.0	15	Female	161101.0
73	Prof	B	18.0	10	Female	105450.0
74	AssocProf	B	19.0	6	Female	104542.0
75	Prof	B	17.0	17	Female	124312.0
76	Prof	A	28.0	14	Female	109954.0
77	Prof	A	23.0	15	Female	109646.0

In [19]:
```
df.sample(4) #To show random (row every time changes)
```

Out[19]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
1	Prof	A	12.0	6	Male	93000.0
62	AssocProf	B	14.0	7	Female	109650.0
69	Prof	A	28.0	7	Female	116450.0
3	Prof	A	40.0	31	Male	131205.0

In [20]:
```
df.RANK #To display particular column only.
```

Out[20]:
```
0          Prof
1          Prof
2          Prof
3          Prof
4          Prof
         ...
73         Prof
74    AssocProf
75         Prof
76         Prof
77         Prof
Name: RANK, Length: 78, dtype: object
```

In [21]:
```
df[['RANK','SEX']] #To display particular column only.(maybe more than 1 columns)
```

Loading [MathJax]/extensions/Safe.js

Out[21]:

	RANK	SEX
0	Prof	Male
1	Prof	Male
2	Prof	Male
3	Prof	Male
4	Prof	Male
...	...	...
73	Prof	Female
74	AssocProf	Female
75	Prof	Female
76	Prof	Female
77	Prof	Female

78 rows × 2 columns

In [22]:
```
df['RANK'].unique() #To show unique data-values of particular columns/attributes
```

Out[22]:
```
array(['Prof', 'AssocProf', 'AsstProf'], dtype=object)
```

In [23]:
```
df['SEX'].unique()
```

Out[23]:
```
array(['Male', 'Female'], dtype=object)
```

In [24]:
```
df['RANK'].value_counts() #To show count  no. of unique data-values of particular c
```

Out[24]:
```
RANK
Prof         46
AsstProf     19
AssocProf    13
Name: count, dtype: int64
```

In [25]:
```
df['SEX'].value_counts()
```

Out[25]:
```
SEX
Male      39
Female    39
Name: count, dtype: int64
```

In [26]:
```
df['SEX'].value_counts(normalize=True) #To show in PERCENTAGE (%)
```

Out[26]:
```
SEX
Male      0.5
Female    0.5
Name: proportion, dtype: float64
```

In [27]:
```
df['SALARY'].max() #To show in MAX
```

Out[27]:
```
186960.0
```

In [28]:
```
df['SALARY'].min() #To show in MIN
```

Out[28]:
```
57800.0
```

In [29]:
```
df['SALARY'].mean() #To show in AVG
```

Out[29]:
```
108003.3552631579
```

In [30]:
```
df['SALARY']>100000 #To show only salary with condition
```

Out[30]:
```
0      True
1     False
2      True
3      True
4      True
      ...
73     True
74     True
75     True
76     True
77     True
Name: SALARY, Length: 78, dtype: bool
```

In [31]:
```
df[df['SALARY']>130000] #To show only whole data set columns with condition
```

Out[31]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
3	Prof	A	40.0	31	Male	131205.0
11	Prof	B	23.0	23	Male	134778.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
19	Prof	A	29.0	27	Male	150500.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
31	Prof	B	22.0	21	Male	155750.0
36	Prof	B	45.0	45	Male	146856.0
40	Prof	A	39.0	36	Female	137000.0
44	Prof	B	23.0	19	Female	151768.0
45	Prof	B	25.0	25	Female	140096.0
58	Prof	B	36.0	26	Female	144651.0
72	Prof	B	24.0	15	Female	161101.0

In [32]:
```
df[(df['SALARY']>130000) & (df['SEX']=='Male')] #salary>130k and Male
```

Out[32]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
3	Prof	A	40.0	31	Male	131205.0
11	Prof	B	23.0	23	Male	134778.0
13	Prof	B	NaN	33	Male	162200.0
14	Prof	B	25.0	19	Male	153750.0
15	Prof	B	17.0	3	Male	150480.0
19	Prof	A	29.0	27	Male	150500.0
26	Prof	A	38.0	19	Male	148750.0
27	Prof	A	45.0	43	Male	155865.0
31	Prof	B	22.0	21	Male	155750.0
36	Prof	B	45.0	45	Male	146856.0

In [33]:
```
df[(df['SALARY']>130000) & (df['SEX']=='Male') & (df['PHd']>39.0)]
```

Out[33]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
3	Prof	A	40.0	31	Male	131205.0
27	Prof	A	45.0	43	Male	155865.0
36	Prof	B	45.0	45	Male	146856.0

In [34]:
```
df.isnull().any(axis=0) #To show if there is any NULL value in particular attribut
```

Out[34]:
```
RANK          False
DISCIPLINE    False
PHd            True
SERVICE       False
SEX           False
SALARY         True
dtype: bool
```

In [35]:
```
df[df.isnull().any(axis=1)] #To show which particular row have NULL value.
```

Out[35]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
7	Prof	A	18.0	18	Male	NaN
13	Prof	B	NaN	33	Male	162200.0
28	AsstProf	B	7.0	2	Male	NaN
34	AssocProf	B	NaN	8	Male	119800.0

In [36]:
```
# HANDLING MISSINF VALUES (By Removing NULL value containing Rows or By replacing w
```

In [37]:
```
# 1.  By replacing with AVG value
```

In [38]:
```
df['PHd'].mean()
```

Out[38]:
```
19.605263157894736
```

In [39]:
```
df['PHd'].fillna(df['PHd'].mean())
```

Out[39]:
```
0     56.0
1     12.0
2     23.0
3     40.0
4     20.0
      ...
73    18.0
74    19.0
75    17.0
76    28.0
77    23.0
Name: PHd, Length: 78, dtype: float64
```

In [40]:
```
df[df.isnull().any(axis=1)] #PHd ko null wala row has been removed
```

Out[40]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
7	Prof	A	18.0	18	Male	NaN
13	Prof	B	NaN	33	Male	162200.0
28	AsstProf	B	7.0	2	Male	NaN
34	AssocProf	B	NaN	8	Male	119800.0

In [41]:
```
print(df.iloc[[13]]) #replaced PHd value with mean of PHd (See in row 13)
```

```
    RANK DISCIPLINE  PHd  SERVICE   SEX    SALARY
13  Prof          B  NaN       33  Male  162200.0
```

In [42]:
```
# 1.  By deleting NULL Value rows
```

In [43]:
```
df.dropna(inplace=True)
```

In [44]:
```
df[df.isnull().any(axis=1)]
```

Out[44]:

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY

In [45]:
```
df.shape #See 2 rows has been removes and total row becomes 76 from 78.
```

Out[45]:
```
(74, 6)
```

In [ ]:
```

```

In [46]:
```
# HOW TO ADD ROW IN DATA-FRAME.
```

In [47]:
```
df2={'RANK':'Prof', 'DISCIPLINE':'B','PHd':99,'SERVICE':77, 'SEX':'Male', 'SALARY':
#df = pd.DataFrame(df).append(df2, ignore_index = True)  #NOT WORKING
df = pd.concat([df, pd.DataFrame([df2])], ignore_index=True)
display(df)
```

	RANK	DISCIPLINE	PHd	SERVICE	SEX	SALARY
0	Prof	B	56.0	49	Male	186960.0
1	Prof	A	12.0	6	Male	93000.0
2	Prof	A	23.0	20	Male	110515.0
3	Prof	A	40.0	31	Male	131205.0
4	Prof	B	20.0	18	Male	104800.0
...	...	...	...	...	...	...
70	AssocProf	B	19.0	6	Female	104542.0
71	Prof	B	17.0	17	Female	124312.0
72	Prof	A	28.0	14	Female	109954.0
73	Prof	A	23.0	15	Female	109646.0
74	Prof	B	99.0	77	Male	9999999.0

75 rows × 6 columns

In [59]:
```
print(df.iloc[[73,74]])
```

```
    RANK DISCIPLINE   PHd  SERVICE     SEX     SALARY
73  Prof          A  23.0       15  Female   109646.0
74  Prof          B  99.0       77    Male  9999999.0
```

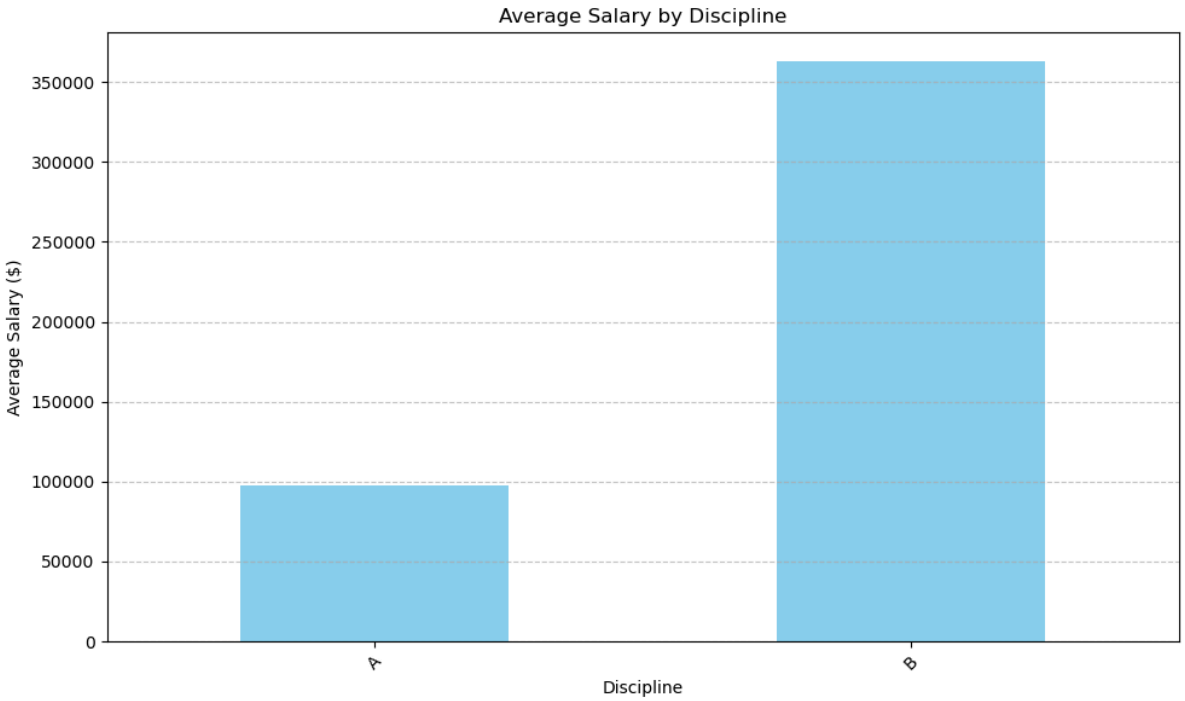
In [ ]:
```
# HOW TO DELETE COLUMN
```

In [ ]:
```
# Import pandas package
import pandas as pd

# create a dictionary with five fields each
data = {
'A': ['A1', 'A2', 'A3', 'A4', 'A5'],
'B': ['B1', 'B2', 'B3', 'B4', 'B5'],
'C': ['C1', 'C2', 'C3', 'C4', 'C5'],
'D': ['D1', 'D2', 'D3', 'D4', 'D5'],
'E': ['E1', 'E2', 'E3', 'E4', 'E5']}

print(data)
```

In [ ]:
```
# Convert the dictionary into DataFrame
df = pd.DataFrame(data)

# Remove two columns name is 'C' and 'D'
df.drop(['C', 'D'], axis=1)

# df.drop(columns =['C', 'D'])
```

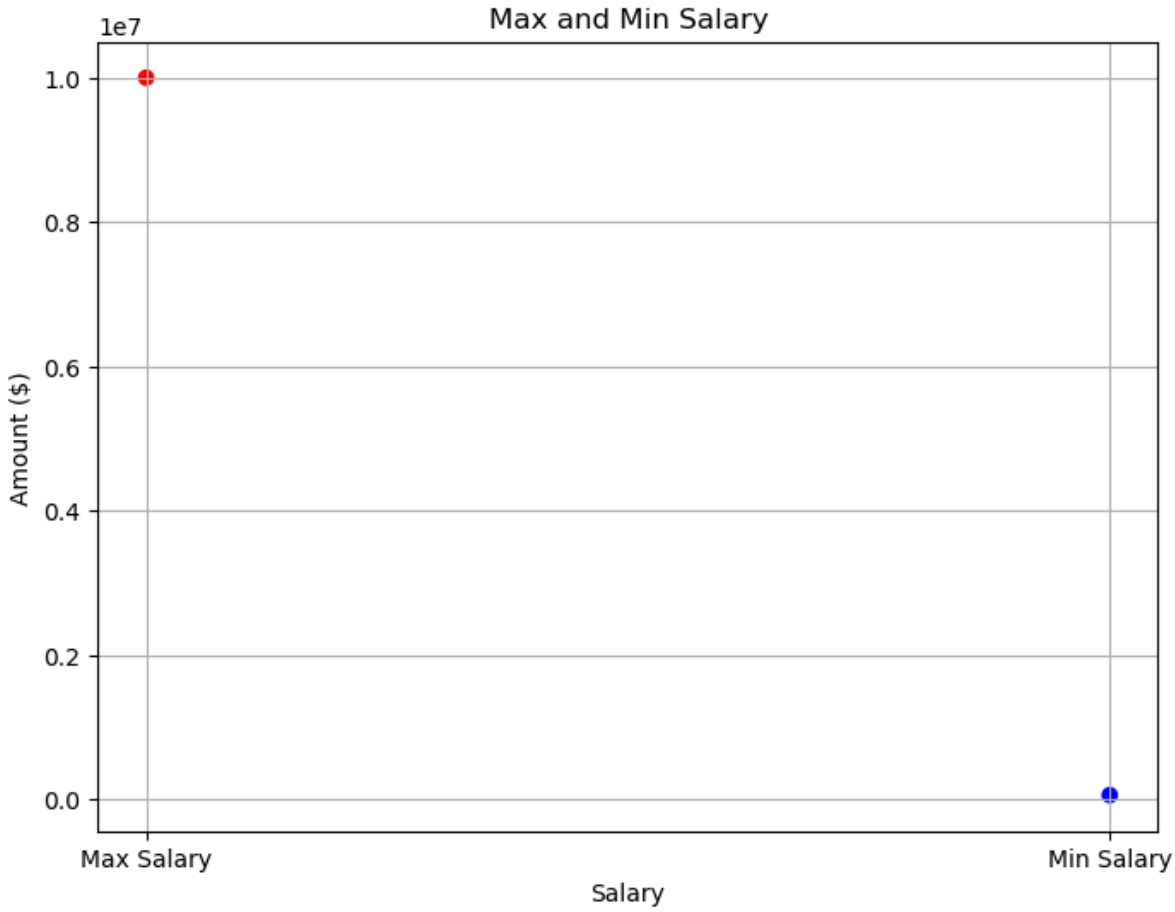
In [49]:
```
import matplotlib.pyplot as plt

# Calculate average salary by discipline
avg_salary_by_discipline = df.groupby('DISCIPLINE')['SALARY'].mean()

# Plotting
plt.figure(figsize=(10, 6))
avg_salary_by_discipline.plot(kind='bar', color='skyblue')
age Salary by Discipline')
```

```
plt.xlabel('Discipline')
plt.ylabel('Average Salary ($)')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

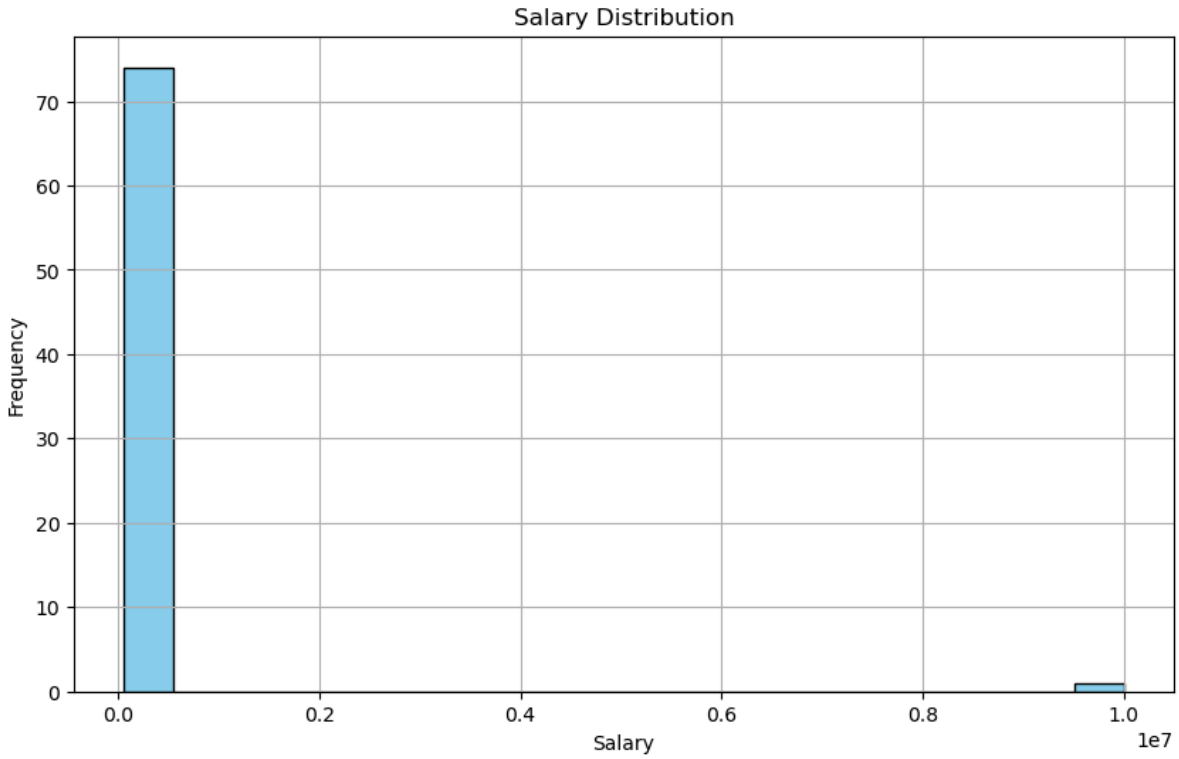


In [50]:

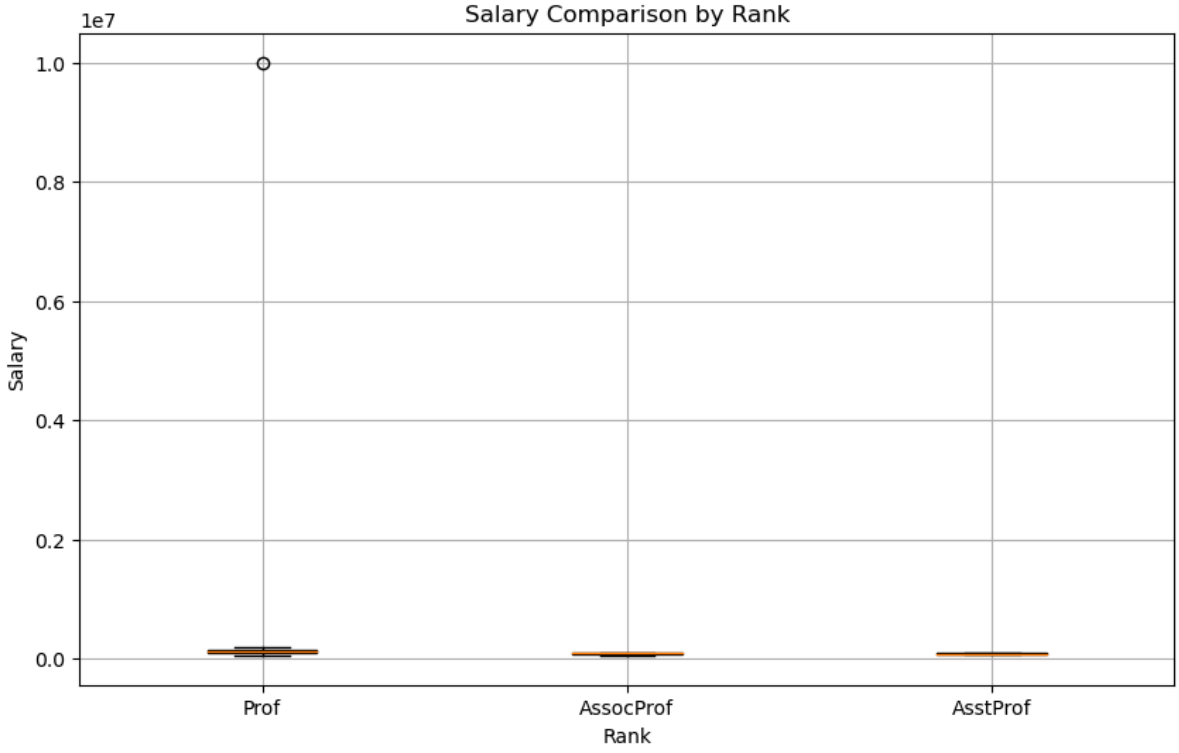
```
import matplotlib.pyplot as plt

# Find max and min salary
max_salary = df['SALARY'].max()
min_salary = df['SALARY'].min()

# Plotting
plt.figure(figsize=(8, 6))
plt.scatter(['Max Salary', 'Min Salary'], [max_salary, min_salary], color=['red', '
plt.title('Max and Min Salary')
plt.xlabel('Salary')
plt.ylabel('Amount ($)')
plt.grid(True)
plt.show()
```



In [51]:
```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 6))
plt.hist(df['SALARY'], bins=20, color='skyblue', edgecolor='black')
plt.title('Salary Distribution')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



In [52]:
```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 6))
plt.boxplot([df[df['RANK'] == 'Prof']['SALARY'], df[df['RANK'] == 'AssocProf']['SAL
            labels=['Prof', 'AssocProf', 'AsstProf'], patch_artist=True)
plt.title('Salary Comparison by Rank')
plt.xlabel('Rank')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```



In [53]:
```
import matplotlib.pyplot as  plt
                          as sns
```

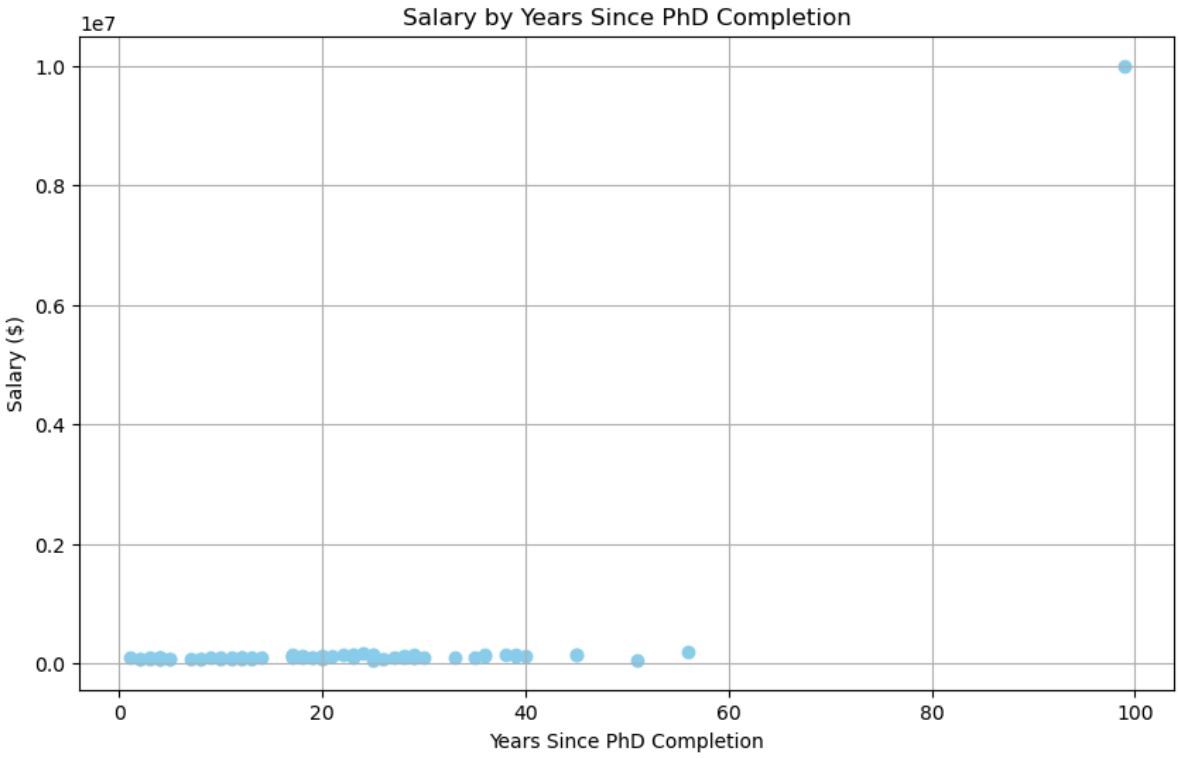
```
# Group by Discipline and Rank./
avg_salary_by_discipline_rank = df.groupby(['DISCIPLINE', 'RANK'])['SALARY'].mean()

# Plotting
plt.figure(figsize=(10, 6))
avg_salary_by_discipline_rank.plot(kind='bar', color=['skyblue', 'orange', 'green']
plt.title('Salary by Discipline and Rank')
plt.xlabel('Discipline')
plt.ylabel('Average Salary ($)')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(title='Rank')
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

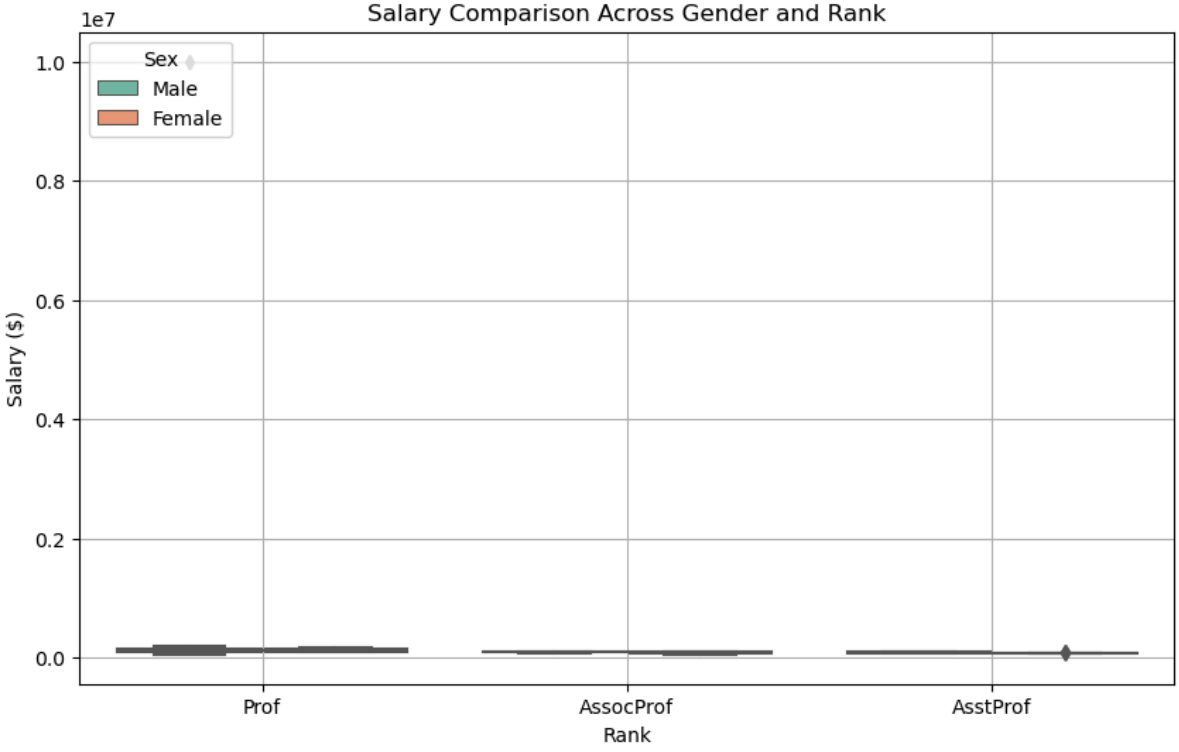
In [54]:

```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(df['PHd'], df['SALARY'], color='skyblue')
plt.title('Salary by Years Since PhD Completion')
plt.xlabel('Years Since PhD Completion')
plt.ylabel('Salary ($)')
plt.grid(True)
plt.show()
```

Loading [MathJax]/extensions/Safe.js



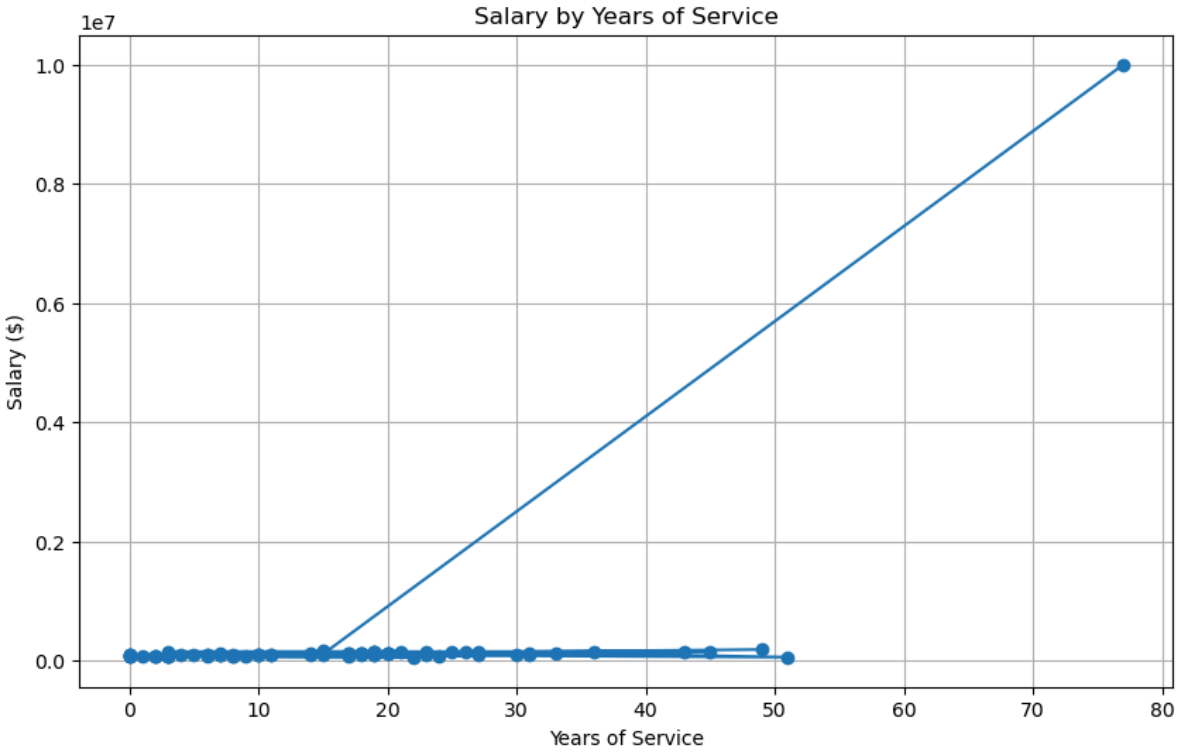
In [55]:
```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 6))
sns.boxplot(x='RANK', y='SALARY', hue='SEX', data=df, palette='Set2')
plt.title('Salary Comparison Across Gender and Rank')
plt.xlabel('Rank')
plt.ylabel('Salary ($)')
plt.grid(True)
plt.legend(title='Sex', loc='upper left')
plt.show()
```



In [57]:
```
import matplotlib.pyplot as plt
```

Loading [MathJax]/extensions/Safe.js

```
# Plotting
plt.figure(figsize=(10, 6))
plt.plot(df['SERVICE'], df['SALARY'], marker='o', linestyle='-')
plt.title('Salary by Years of Service')
plt.xlabel('Years of Service')
plt.ylabel('Salary ($)')
plt.grid(True)
plt.show()
```



In [ ]:

In [ ]: