In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [13]:
```
df=pd.read_csv(r'C:\Users\Hello\Downloads\HRDataset_v15.csv',encoding= 'unicode_escape')
```

In [16]:
```
df.shape
```

Out[16]:
```
(311, 36)
```

In [17]:
```
df.head()
```

Out[17]:

	ï»¿Employee_Name	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversityJobFairI
0	Adinolfi, Wilson K	10026	0	0	1	1	5	4	
1	Ait Sidi, Karthikeyan	10084	1	1	1	5	3	3	
2	Akinkuolie, Sarah	10196	1	1	0	5	5	3	
3	Alagbe,Trina	10088	1	1	0	1	5	3	
4	Anderson, Carol	10069	0	2	0	5	5	3	

5 rows × 36 columns

In [18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 311 entries, 0 to 310
Data columns (total 36 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ï»¿Employee_Name           311 non-null    object
 1   EmpID                       311 non-null    int64
 2   MarriedID                   311 non-null    int64
 3   MaritalStatusID             311 non-null    int64
 4   GenderID                    311 non-null    int64
 5   EmpStatusID                 311 non-null    int64
 6   DeptID                      311 non-null    int64
 7   PerfScoreID                 311 non-null    int64
 8   FromDiversityJobFairID      311 non-null    int64
 9   Salary                      311 non-null    int64
 10  Termd                       311 non-null    int64
 11  PositionID                  311 non-null    int64
 12  Position                    311 non-null    object
 13  State                       311 non-null    object
 14  Zip                         311 non-null    int64
 15  DOB                         311 non-null    object
 16  Sex                         311 non-null    object
 17  MaritalDesc                 311 non-null    object
 18  CitizenDesc                 311 non-null    object
 19  HispanicLatino              311 non-null    object
 20  RaceDesc                    311 non-null    object
 21  DateofHire                  311 non-null    object
 22  DateofTermination           104 non-null    object
 23  TermReason                  311 non-null    object
 24  EmploymentStatus            311 non-null    object
 25  Department                  311 non-null    object
 26  ManagerName                 311 non-null    object
 27  ManagerID                   303 non-null    float64
 28  RecruitmentSource           311 non-null    object
 29  PerformanceScore            311 non-null    object
 30  EngagementSurvey            311 non-null    float64
 31  EmpSatisfaction             311 non-null    int64
 32  SpecialProjectsCount        311 non-null    int64
 33  LastPerformanceReview_Date  311 non-null    object
 34  DaysLateLast30              311 non-null    int64
 35  Absences                    311 non-null    int64
```

```
dtypes: float64(2), int64(16), object(18)
memory usage: 87.6+ KB
```

Data Cleaning

In [19]:
```
pd.isnull(df).sum()
```

Out[19]:

ï»¿Employee_Name	0
EmpID	0
MarriedID	0
MaritalStatusID	0
GenderID	0
EmpStatusID	0
DeptID	0
PerfScoreID	0
FromDiversityJobFairID	0
Salary	0
Termd	0
PositionID	0
Position	0
State	0
Zip	0
DOB	0
Sex	0
MaritalDesc	0
CitizenDesc	0
HispanicLatino	0
RaceDesc	0
DateofHire	0
DateofTermination	207
TermReason	0
EmploymentStatus	0
Department	0
ManagerName	0
ManagerID	8
RecruitmentSource	0
PerformanceScore	0
EngagementSurvey	0
EmpSatisfaction	0
SpecialProjectsCount	0
LastPerformanceReview_Date	0
DaysLateLast30	0
Absences	0

dtype: int64

In [20]:
```
df.dropna(inplace=True)
```

In [21]:
```
df.shape
```

Out[21]: (104, 36)

In [22]:
```
pd.isnull(df).sum()
```

Out[22]:
```
ï»¿Employee_Name              0
EmpID                         0
MarriedID                     0
MaritalStatusID               0
GenderID                      0
EmpStatusID                   0
DeptID                        0
PerfScoreID                   0
FromDiversityJobFairID        0
Salary                        0
Termd                         0
PositionID                    0
Position                      0
State                         0
Zip                           0
DOB                           0
Sex                           0
MaritalDesc                   0
CitizenDesc                   0
HispanicLatino                0
RaceDesc                      0
DateofHire                    0
DateofTermination             0
TermReason                    0
EmploymentStatus              0
Department                    0
ManagerName                   0
ManagerID                     0
RecruitmentSource             0
PerformanceScore              0
EngagementSurvey              0
EmpSatisfaction               0
SpecialProjectsCount          0
LastPerformanceReview_Date    0
DaysLateLast30                0
Absences                      0
dtype: int64
```

In [23]:
```
df['ManagerID'] = df['ManagerID'].astype('int')
```

In [24]:
```
df['ManagerID'].dtypes
```

Out[24]:
```
dtype('int64')
```

In [25]:
```
df['EngagementSurvey'] = df['EngagementSurvey'].astype('int')
```

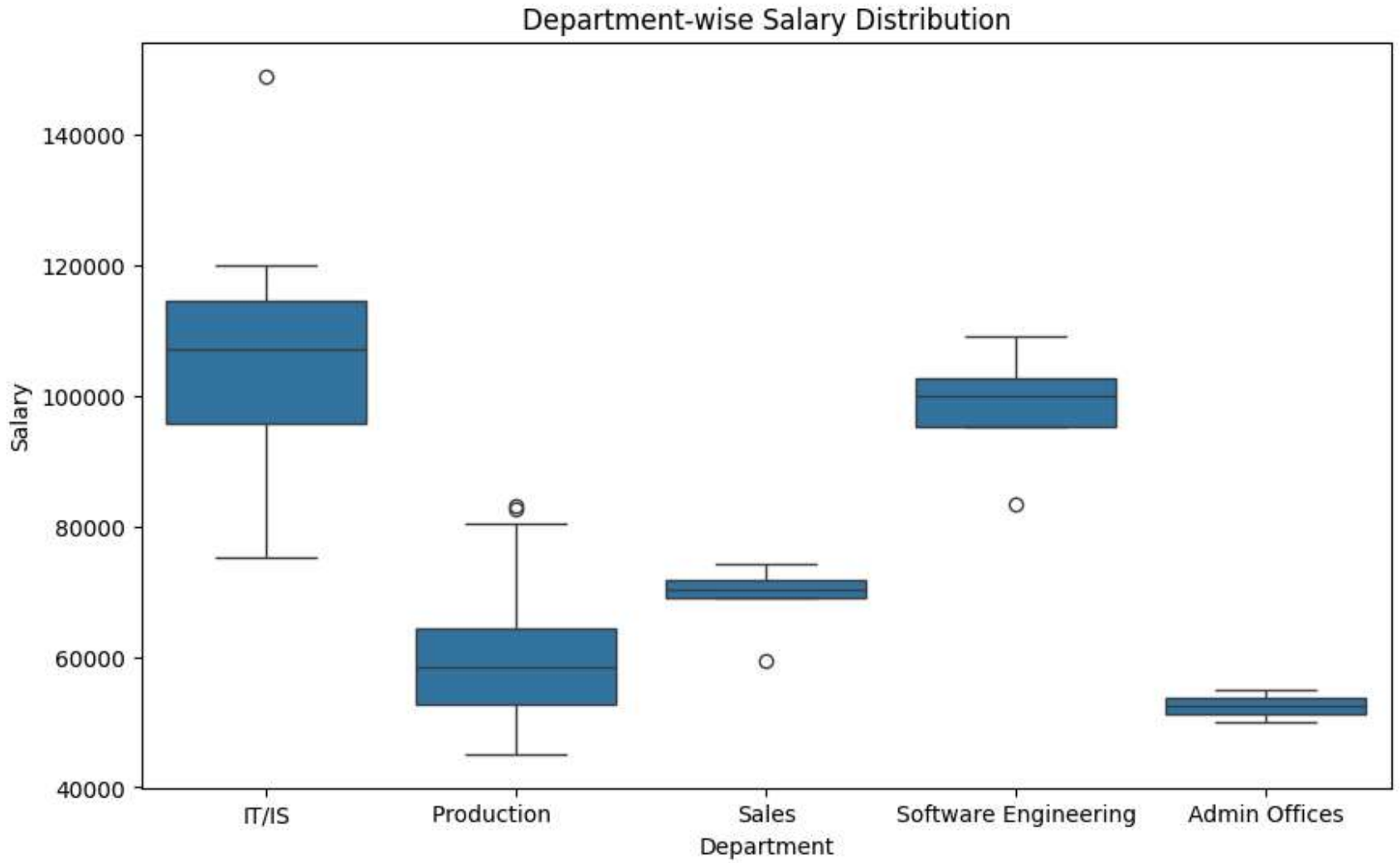
In [26]:
```
df['EngagementSurvey'].dtypes
```

Out[26]:
```
dtype('int64')
```

Explorartory Data Analysis

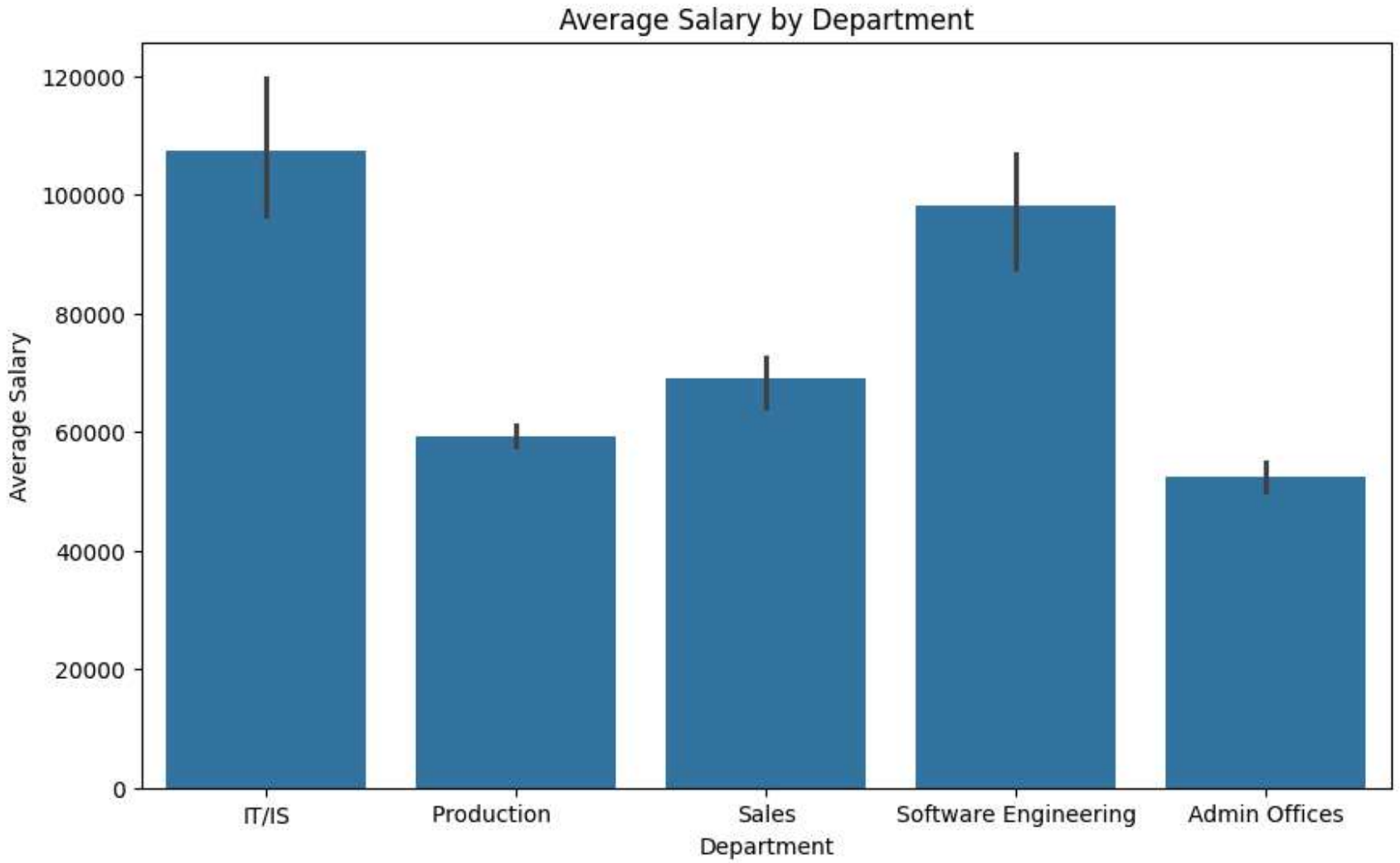
Department Wise Salary Distribution

In [27]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x="Department", y="Salary", data=df)
plt.title("Department-wise Salary Distribution")
plt.xlabel("Department")
plt.ylabel("Salary")
plt.show()
```



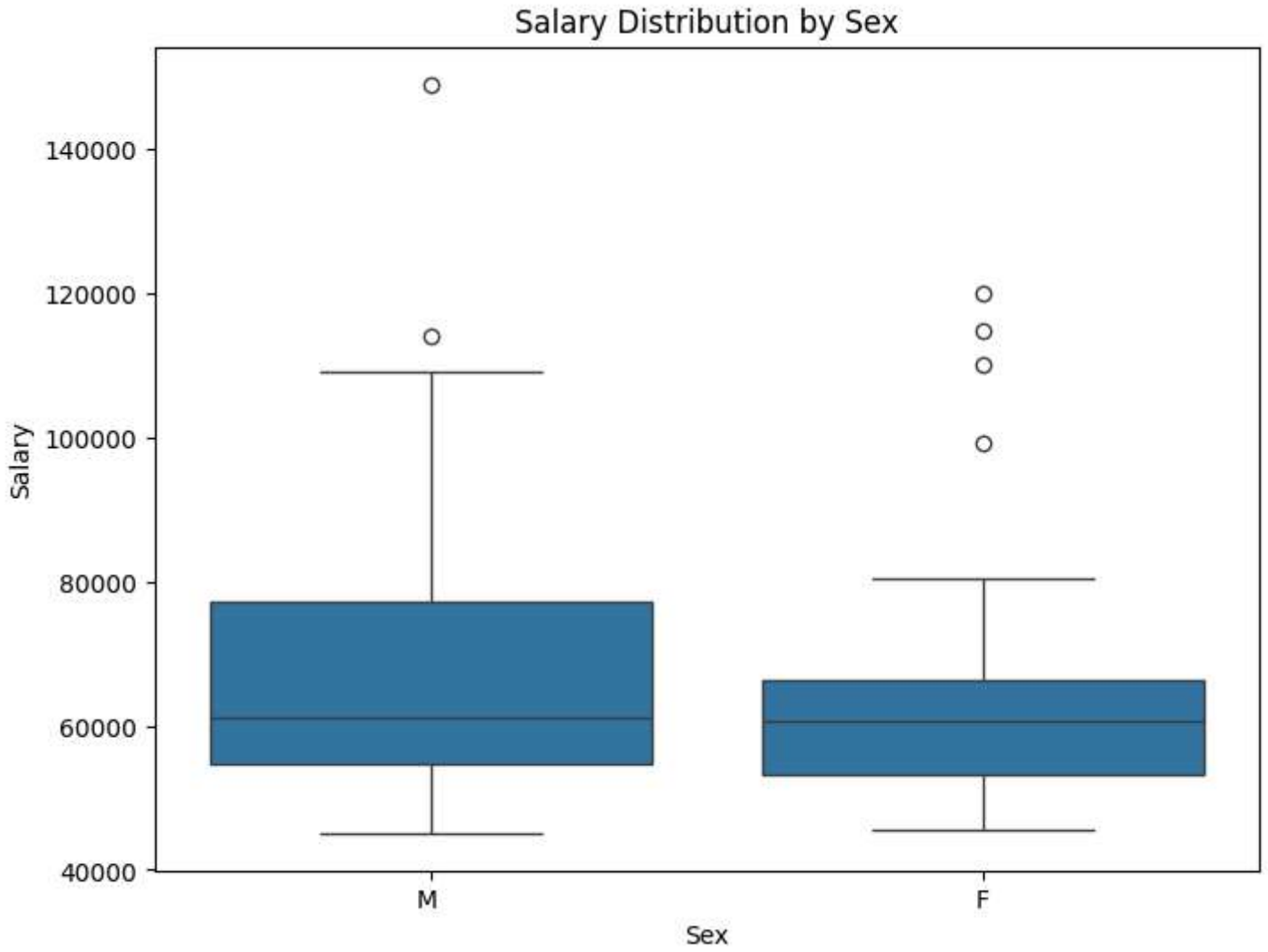
Average Salary By Department

In [28]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x="Department", y="Salary", data=df[df["Department"]!= "Executive Office"], estimator=np.mean)
plt.title("Average Salary by Department")
plt.xlabel("Department")
plt.ylabel("Average Salary")
plt.show()
```



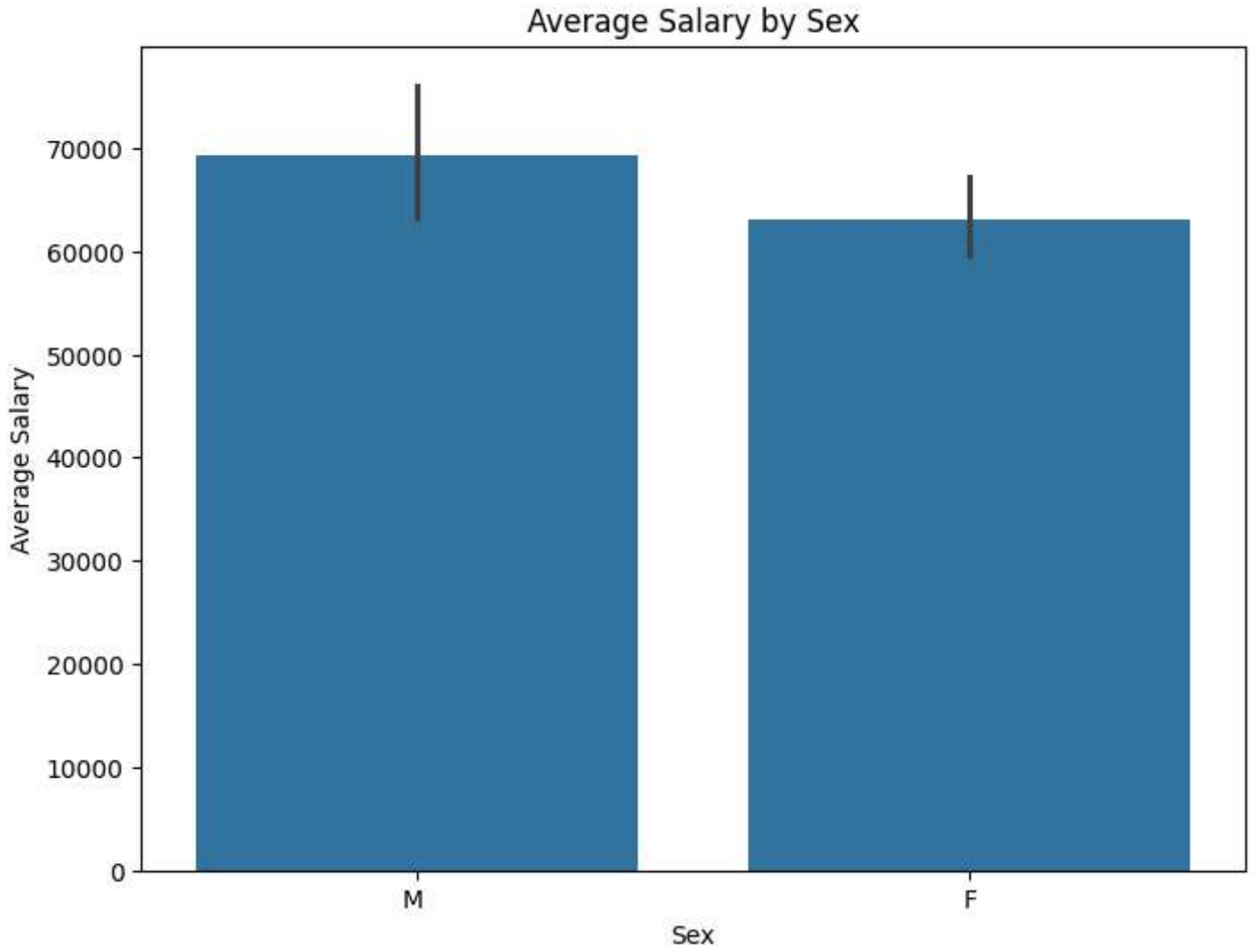
Salary Distribution By Sex

In [30]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x="Sex", y="Salary", data=df[df["Department"]!= "Executive Office"])
plt.title("Salary Distribution by Sex")
plt.xlabel("Sex")
plt.ylabel("Salary")
plt.show()
```



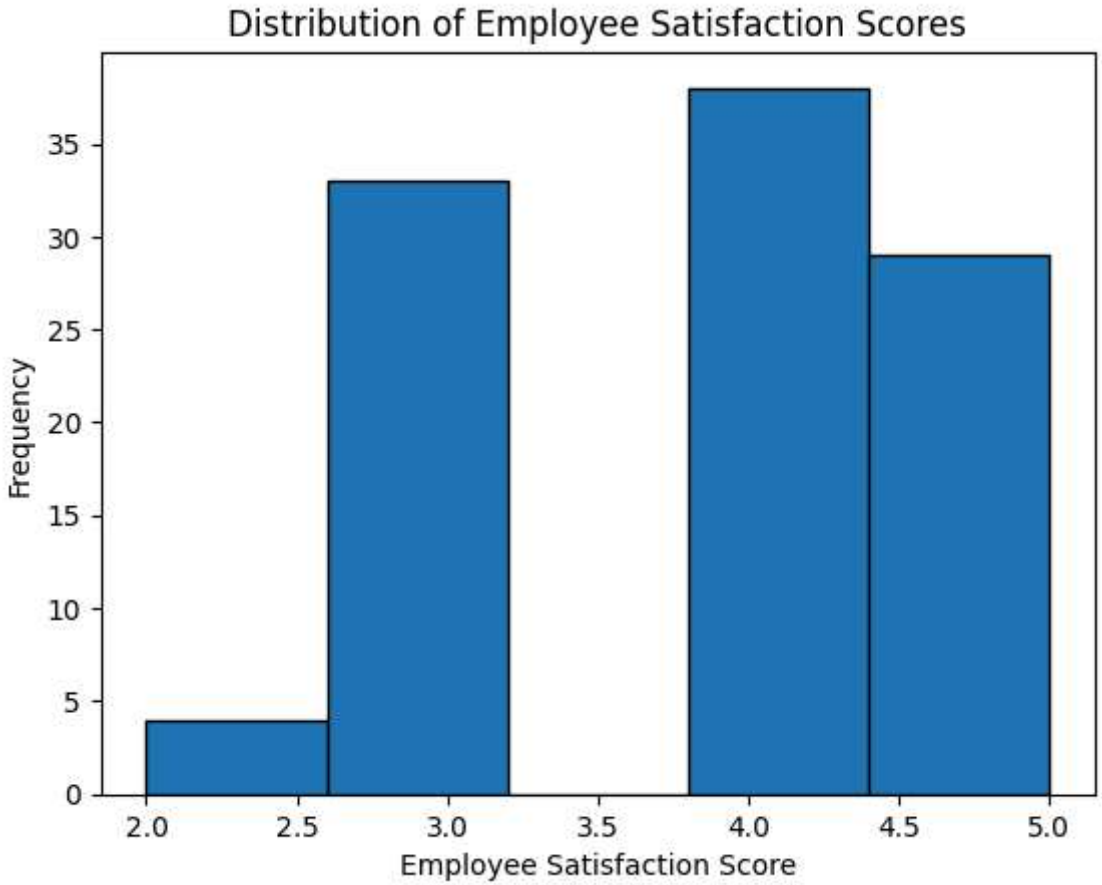
Average Salary By Sex

In [31]:
```
plt.figure(figsize=(8, 6))
sns.barplot(x="Sex", y="Salary", data=df[df["Department"]!= "Executive Office"], estimator=np.mean)
plt.title("Average Salary by Sex")
plt.xlabel("Sex")
plt.ylabel("Average Salary")
plt.show()
```

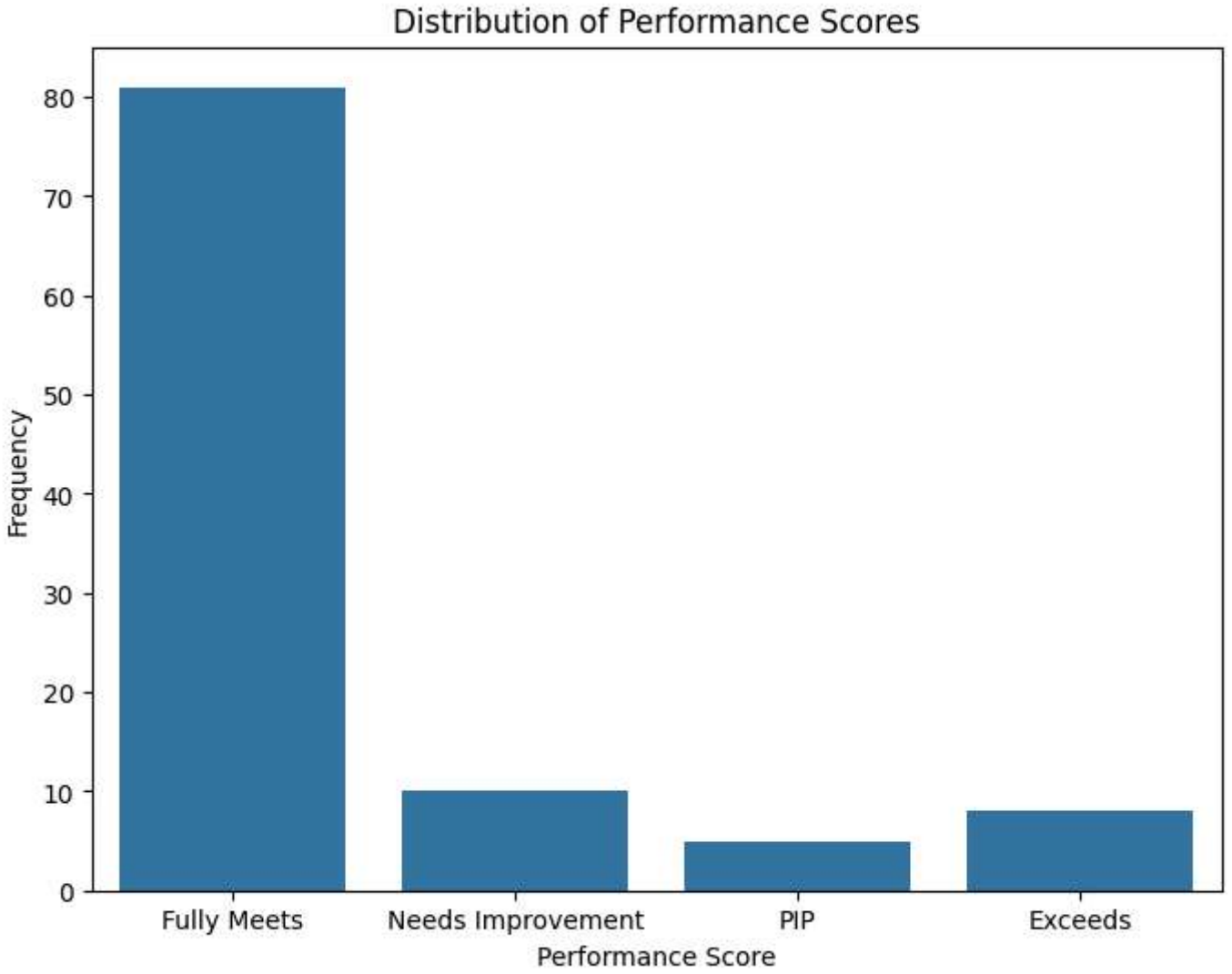


Distribution of Employee Satisfaction Scores

In [33]:
```
plt.hist(df['EmpSatisfaction'], bins=5, edgecolor='black')
plt.xlabel('Employee Satisfaction Score')
plt.ylabel('Frequency')
plt.title('Distribution of Employee Satisfaction Scores')
plt.show()
```



In [36]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='PerformanceScore', data=df)
plt.xlabel('Performance Score')
plt.ylabel('Frequency')
plt.title('Distribution of Performance Scores')
plt.show()
```

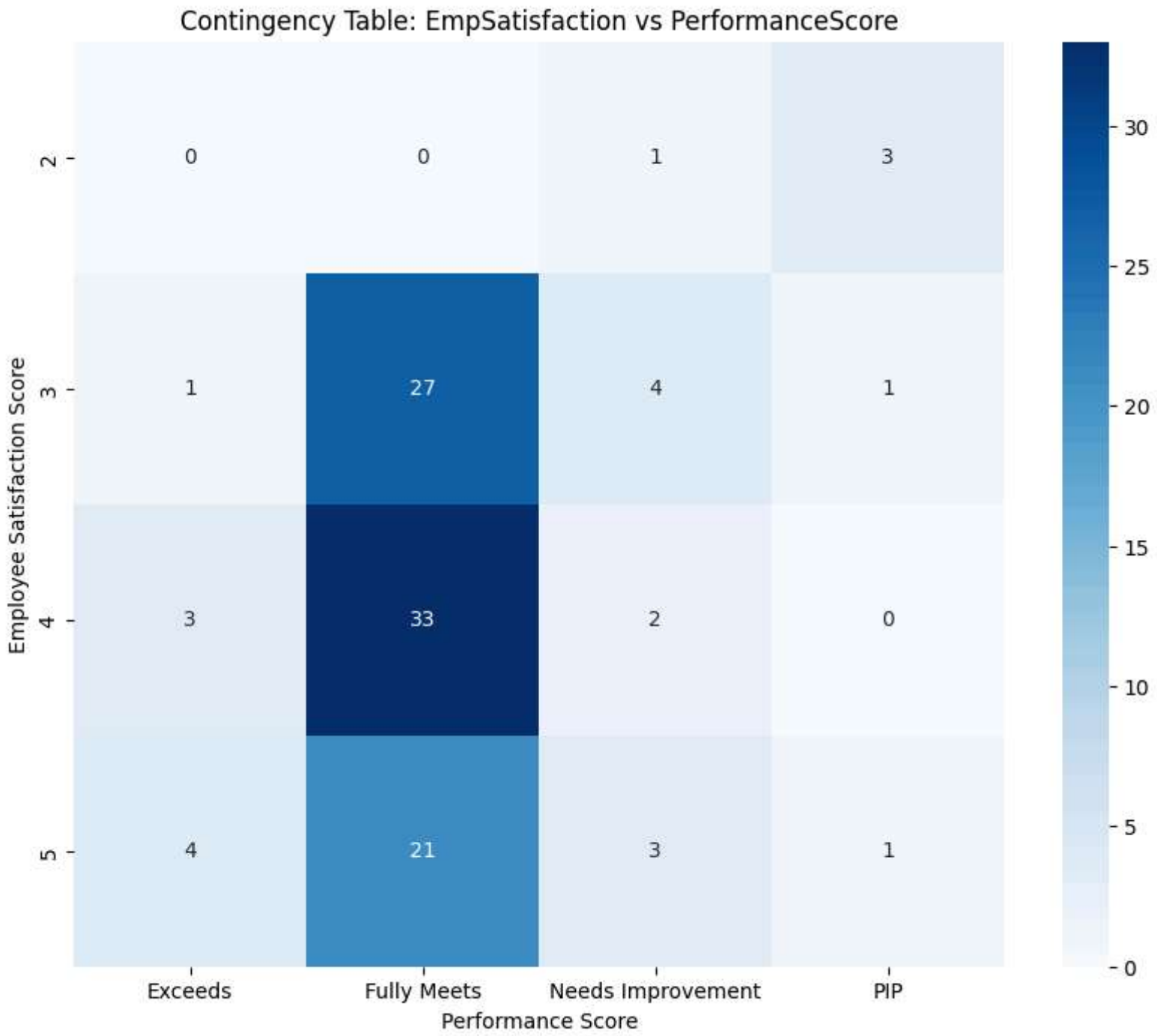


Create a contingency table to analyze the relationship between EmpSatisfaction and PerformanceScore

In [37]:
```
contingency_table = pd.crosstab(df['EmpSatisfaction'], df['PerformanceScore'])
```

Visualize the contingency table using a heatmap

In [38]:
```
plt.figure(figsize=(10, 8))
sns.heatmap(contingency_table, annot=True, cmap='Blues', fmt='d')
plt.xlabel('Performance Score')
plt.ylabel('Employee Satisfaction Score')
plt.title('Contingency Table: EmpSatisfaction vs PerformanceScore')
plt.show()
```


Contingency Table: EmpSatisfaction vs PerformanceScore
Employee Satisfaction Score
2
3
4
5
0
0
1
3
1
27
4
1
3
33
2
0
4
21
3
1
Exceeds
Fully Meets
Needs Improvement
PIP
Performance Score
30
25
20
15
10
5
0

In [ ]: