In [1]:
```
#Importing the libraries

import numpy as np #np is shortcut
import matplotlib.pyplot as plt #plt is shortcut
import seaborn as sns #sns is a alias taken
import pandas as pd #pd is shortcut panads is used for working with dataset
```

In [2]:
```
#Importing the dataset 1

dataset1=pd.read_csv('employees.csv')
X1 = dataset1.iloc[:,:-1].values  # All columns except last
Y1 = dataset1.iloc[:, -1].values   # The last column (target)
```

In [3]:
```
#Importing the dataset 2

dataset2=pd.read_csv('Employee_noName.csv')
X2 = dataset2.iloc[:,:-1].values  # All columns except last
Y2 = dataset2.iloc[: , -1].values  # The last column (target)
```

In [4]:
```
# printing both datasets

print(X1)
print("\n-----------------------------------------------------------------------------------------------------------
print(X2)
```

```
[['Douglas' 'Male' '8/6/1993' ... 97308 6.945 True]
 ['Thomas' 'Male' '3/31/1996' ... 61933 4.17 True]
 ['Maria' 'Female' '4/23/1993' ... 130590 11.858 False]
 ...
 ['Russell' 'Male' '5/20/2013' ... 96914 1.421 False]
 ['Larry' 'Male' '4/20/2013' ... 60500 11.985 False]
 ['Albert' 'Male' '5/15/2012' ... 129949 10.169 True]]

-----------------------------------------------------------------------------------------------------------------
--------------

[['Bachelors' 2017 'Bangalore' ... 'Male' 'No' 0]
 ['Bachelors' 2013 'Pune' ... 'Female' 'No' 3]
 ['Bachelors' 2014 'New Delhi' ... 'Female' 'No' 2]
 ...
 ['Masters' 2018 'New Delhi' ... 'Male' 'No' 5]
 ['Bachelors' 2012 'Bangalore' ... 'Male' 'Yes' 2]
 ['Bachelors' 2015 'Bangalore' ... 'Male' 'Yes' 4]]
```

In [5]:
```
print(Y1)
print("\n-----------------------------------------------------------------------------------------------------------
print(Y2)
```

```
['Marketing' nan 'Finance' 'Finance' 'Client Services' 'Legal' 'Product'
 'Finance' 'Engineering' 'Business Development' nan 'Legal'
 'Human Resources' 'Sales' 'Finance' 'Product' 'Human Resources' 'Product'
 'Client Services' 'Product' 'Legal' 'Marketing' 'Client Services' nan
 'Client Services' 'Client Services' 'Marketing' 'Legal' 'Client Services'
 'Legal' 'Engineering' 'Product' nan 'Business Development'
 'Client Services' 'Sales' 'Business Development' 'Client Services'
 'Business Development' 'Client Services' 'Distribution'
 'Business Development' 'Legal' 'Marketing' 'Product' 'Sales' 'Finance'
 'Client Services' 'Business Development' 'Sales' 'Engineering' 'Sales'
 'Human Resources' 'Finance' 'Engineering' 'Product' 'Finance'
 'Human Resources' 'Engineering' 'Engineering' 'Distribution'
 'Business Development' 'Marketing' 'Human Resources'
 'Business Development' 'Distribution' 'Business Development' 'Finance'
 'Finance' 'Finance' 'Client Services' 'Sales' 'Product' 'Sales'
 'Marketing' 'Human Resources' 'Distribution' 'Marketing' 'Sales'
 'Product' 'Sales' 'Legal' 'Client Services' 'Finance' 'Finance'
 'Client Services' 'Business Development' 'Sales' 'Legal' 'Legal' 'Legal'
 nan 'Business Development' 'Legal' 'Legal' 'Client Services' 'Finance'
 'Marketing' 'Marketing' 'Business Development' 'Finance' 'Marketing'
 'Client Services' 'Finance' 'Marketing' 'Finance' 'Legal' 'Legal' 'Legal'
 nan 'Legal' 'Business Development' 'Marketing' 'Engineering'
 'Business Development' 'Product' 'Legal' 'Finance' 'Business Development'
 'Marketing' 'Business Development' 'Product' 'Engineering' 'Product'
 'Product' 'Human Resources' 'Human Resources' 'Human Resources'
 'Client Services' 'Business Development' 'Human Resources' 'Product'
 'Human Resources' 'Client Services' 'Business Development' 'Legal'
 'Legal' 'Distribution' 'Engineering' nan 'Marketing' 'Product' 'Finance'
 'Engineering' 'Sales' 'Client Services' 'Product' 'Legal' 'Sales'
 'Distribution' 'Marketing' 'Business Development' 'Client Services'
 'Finance' 'Product' 'Business Development' 'Human Resources' 'Product'
 'Marketing' 'Marketing' 'Finance' 'Distribution' 'Legal'
 'Client Services' 'Business Development' 'Legal' 'Sales' 'Sales'
 'Marketing' 'Product' 'Sales' 'Engineering' 'Finance' 'Engineering'
 'Client Services' 'Engineering' 'Product' 'Distribution' 'Product'
 'Finance' 'Business Development' 'Distribution' 'Business Development'
 'Distribution' 'Client Services' 'Legal' 'Sales' 'Marketing' 'Legal'
 'Sales' 'Finance' 'Engineering' 'Legal' 'Legal' 'Distribution' 'Product'
 'Client Services' 'Client Services' 'Product' nan 'Finance' 'Marketing'
 'Sales' 'Business Development' 'Marketing' 'Finance' 'Client Services'
 'Client Services' 'Human Resources' 'Engineering' 'Legal'
 'Human Resources' 'Client Services' 'Engineering' 'Engineering'
 'Client Services' 'Marketing' 'Client Services' 'Finance' 'Finance'
 'Marketing' 'Legal' 'Finance' 'Legal' 'Distribution' 'Sales' 'Finance'
 'Client Services' 'Engineering' 'Distribution' 'Legal' 'Product'
 'Human Resources' 'Sales' 'Client Services' 'Engineering' 'Product'
 'Legal' 'Legal' 'Human Resources' 'Distribution' 'Finance' 'Engineering'
 'Product' 'Client Services' 'Engineering' 'Human Resources' 'Product'
 'Distribution' 'Business Development' 'Sales' 'Business Development'
 'Marketing' 'Sales' 'Client Services' 'Human Resources' 'Legal' 'Sales'
 nan 'Human Resources' 'Distribution' 'Product' 'Engineering'
 'Engineering' 'Human Resources' 'Client Services' 'Distribution'
 'Distribution' 'Finance' 'Human Resources' 'Human Resources' 'Marketing'
 'Product' 'Product' 'Marketing' 'Business Development' 'Finance' 'Sales'
 'Distribution' 'Business Development' 'Business Development'
 'Human Resources' 'Client Services' 'Engineering' 'Client Services'
 'Human Resources' 'Finance' 'Client Services' 'Distribution' 'Legal' nan
 'Client Services' 'Client Services' 'Marketing' 'Distribution' 'Legal'
 'Sales' 'Human Resources' 'Marketing' 'Human Resources' 'Engineering'
 'Engineering' 'Human Resources' 'Client Services' 'Finance' 'Marketing'
 'Business Development' 'Distribution' 'Legal' 'Marketing' 'Legal'
 'Finance' 'Sales' 'Legal' nan 'Client Services' 'Product'
 'Business Development' 'Finance' 'Marketing' 'Sales' 'Sales' 'Product'
 'Sales' 'Business Development' 'Client Services' 'Product' 'Marketing'
 'Finance' 'Engineering' 'Client Services' 'Marketing' 'Product'
 'Client Services' 'Client Services' 'Finance' 'Legal' 'Sales' 'Product'
 'Human Resources' 'Sales' 'Finance' 'Product' 'Engineering'
 'Business Development' 'Human Resources' 'Human Resources' 'Sales'
 'Finance' 'Sales' 'Sales' 'Sales' 'Engineering' 'Marketing' 'Legal'
 'Legal' 'Distribution' 'Engineering' 'Product' 'Client Services' 'Sales'
 'Sales' 'Distribution' 'Finance' 'Product' 'Human Resources'
 'Client Services' nan 'Business Development' 'Product'
 'Business Development' 'Sales' 'Engineering' 'Sales' 'Distribution'
 'Sales' 'Engineering' 'Human Resources' 'Product' 'Marketing' 'Sales'
 'Engineering' nan 'Product' 'Product' 'Client Services' 'Sales'
 'Client Services' 'Product' 'Client Services' 'Sales' 'Sales'
 'Client Services' 'Engineering' 'Product' 'Sales' 'Human Resources'
 'Distribution' 'Human Resources' 'Finance' 'Marketing'
 'Business Development' 'Engineering' 'Marketing' 'Sales' 'Finance'
 'Business Development' 'Distribution' 'Client Services' 'Human Resources'
 'Sales' 'Business Development' 'Legal' 'Marketing' 'Business Development'
 'Finance' 'Distribution' 'Human Resources' 'Finance'
 'Business Development' 'Finance' 'Sales' 'Product' 'Client Services'
 'Human Resources' 'Finance' 'Human Resources' 'Sales' 'Finance'
 'Human Resources' 'Distribution' 'Legal' 'Client Services' 'Legal' nan
 'Distribution' 'Finance' 'Sales' nan 'Human Resources' 'Client Services'
 'Business Development' 'Finance' 'Sales' 'Distribution' nan 'Marketing'
 'Engineering' 'Client Services' 'Human Resources' 'Legal' 'Marketing'
 'Marketing' 'Human Resources' 'Marketing' 'Human Resources' 'Engineering'
 'Legal' 'Finance' 'Marketing' 'Legal' 'Marketing' 'Engineering' 'Product'
```

```
'Marketing' 'Legal' 'Sales' 'Engineering' 'Marketing' 'Legal'
'Distribution' 'Human Resources' 'Client Services' 'Business Development'
'Engineering' 'Engineering' 'Human Resources' 'Business Development'
'Business Development' nan 'Sales' 'Business Development' 'Product'
'Human Resources' 'Marketing' 'Finance' 'Distribution'
'Business Development' 'Legal' 'Client Services' 'Marketing'
'Distribution' 'Business Development' 'Finance' 'Sales' 'Sales'
'Marketing' 'Client Services' 'Business Development' 'Distribution'
'Legal' 'Distribution' 'Client Services' 'Marketing' 'Distribution'
'Engineering' 'Client Services' 'Engineering' 'Human Resources'
'Business Development' 'Legal' 'Business Development' nan nan 'Product'
'Sales' 'Legal' 'Human Resources' 'Product' 'Human Resources' nan
'Engineering' 'Distribution' 'Sales' 'Client Services' 'Human Resources'
'Finance' 'Product' 'Product' 'Client Services' 'Distribution'
'Marketing' 'Product' 'Product' 'Legal' 'Marketing'
'Business Development' 'Business Development' 'Human Resources' 'Sales'
'Finance' 'Engineering' 'Distribution' 'Client Services' 'Marketing'
'Product' 'Product' 'Finance' 'Sales' 'Finance' 'Marketing' 'Engineering'
'Product' 'Engineering' 'Client Services' 'Product' 'Marketing'
'Distribution' 'Engineering' 'Human Resources' 'Client Services'
'Engineering' 'Legal' 'Engineering' 'Client Services' 'Human Resources'
'Distribution' nan 'Marketing' 'Client Services' 'Product' 'Marketing'
'Human Resources' nan 'Client Services' 'Product' 'Product'
'Client Services' 'Product' 'Legal' nan 'Marketing' 'Finance'
'Business Development' 'Legal' 'Marketing' 'Marketing' 'Client Services'
'Human Resources' 'Human Resources' 'Business Development' 'Distribution'
'Sales' 'Sales' 'Business Development' 'Finance' 'Business Development'
'Distribution' 'Product' 'Human Resources' 'Distribution' 'Marketing'
'Engineering' 'Business Development' 'Engineering' 'Client Services'
'Client Services' 'Sales' 'Engineering' 'Sales' 'Product' 'Marketing'
'Distribution' 'Finance' 'Distribution' 'Engineering' 'Distribution'
'Marketing' 'Finance' 'Engineering' 'Finance' 'Client Services' 'Sales'
'Legal' 'Sales' 'Marketing' nan 'Marketing' 'Distribution' 'Marketing'
'Legal' 'Client Services' 'Engineering' 'Engineering' nan nan
'Engineering' 'Human Resources' 'Business Development' 'Client Services'
'Distribution' 'Sales' 'Finance' 'Human Resources' 'Finance' 'Marketing'
'Finance' nan 'Business Development' 'Finance' 'Finance'
'Client Services' 'Engineering' 'Product' 'Legal' 'Client Services'
'Marketing' 'Sales' 'Client Services' 'Marketing' 'Distribution'
'Engineering' 'Distribution' 'Distribution' 'Legal' 'Distribution'
'Business Development' 'Marketing' 'Legal' nan 'Client Services'
'Distribution' 'Human Resources' 'Business Development' 'Human Resources'
'Marketing' 'Marketing' 'Client Services' 'Product' 'Client Services'
'Engineering' 'Product' 'Product' 'Distribution' nan 'Finance' 'Finance'
'Distribution' 'Legal' 'Finance' 'Sales' 'Finance' 'Sales' 'Sales'
'Client Services' 'Human Resources' 'Marketing' 'Distribution' 'Legal'
'Distribution' 'Distribution' 'Legal' 'Finance' 'Human Resources'
'Distribution' 'Engineering' nan 'Engineering' 'Legal' 'Human Resources'
'Finance' 'Engineering' 'Engineering' 'Distribution' 'Distribution'
'Engineering' 'Business Development' 'Human Resources' 'Engineering'
'Engineering' 'Human Resources' 'Business Development' 'Marketing'
'Legal' 'Engineering' 'Finance' nan 'Sales' 'Client Services'
'Client Services' 'Marketing' 'Finance' 'Finance' 'Business Development'
'Human Resources' 'Business Development' 'Product' 'Product'
'Business Development' 'Sales' 'Marketing' 'Legal' 'Client Services'
'Human Resources' 'Finance' 'Business Development' 'Business Development'
'Business Development' 'Business Development' 'Legal' 'Product'
'Client Services' 'Business Development' nan 'Legal' 'Client Services'
'Distribution' 'Product' 'Legal' 'Distribution' 'Human Resources'
'Engineering' 'Distribution' 'Legal' 'Sales' 'Finance' 'Human Resources'
'Client Services' 'Sales' 'Marketing' 'Product' 'Product'
'Business Development' 'Finance' nan 'Finance' 'Human Resources' 'Sales'
'Distribution' 'Business Development' 'Human Resources' nan
'Client Services' 'Product' 'Sales' 'Marketing' 'Product' 'Sales'
'Business Development' 'Product' 'Finance' 'Legal' 'Distribution'
'Distribution' nan 'Human Resources' 'Client Services' 'Engineering'
'Marketing' 'Product' 'Product' 'Human Resources' 'Business Development'
'Product' 'Distribution' 'Engineering' 'Sales' 'Finance' 'Engineering'
'Finance' 'Business Development' 'Marketing' 'Product' 'Marketing'
'Distribution' 'Human Resources' 'Engineering' 'Marketing' 'Distribution'
'Legal' 'Human Resources' 'Distribution' 'Business Development'
'Engineering' 'Marketing' 'Sales' nan 'Legal' 'Product' 'Human Resources'
'Distribution' 'Finance' 'Legal' 'Human Resources' 'Business Development'
'Engineering' 'Finance' 'Finance' 'Distribution' 'Human Resources'
'Distribution' 'Business Development' 'Product' 'Sales' 'Legal'
'Client Services' 'Human Resources' 'Finance' 'Product' 'Product' nan nan
'Business Development' nan 'Finance' nan 'Finance' 'Product'
'Human Resources' 'Business Development' 'Marketing' 'Finance' 'Sales'
'Human Resources' nan 'Legal' 'Client Services' 'Marketing' 'Product'
'Business Development' 'Marketing' 'Engineering' 'Business Development'
'Sales' 'Human Resources' 'Engineering' 'Marketing' 'Client Services'
'Distribution' 'Client Services' 'Finance' 'Marketing' 'Distribution'
'Marketing' 'Finance' 'Product' 'Sales' 'Legal' 'Legal' 'Finance'
'Finance' 'Finance' 'Sales' 'Business Development' 'Marketing'
'Business Development' 'Sales' 'Finance' 'Business Development'
'Marketing' 'Human Resources' 'Distribution' 'Distribution'
'Human Resources' 'Human Resources' 'Finance' 'Sales' 'Finance'
'Engineering' 'Product' 'Legal' 'Business Development' nan 'Marketing'
'Distribution' 'Client Services' 'Business Development' 'Product'
'Client Services' 'Distribution' 'Client Services' 'Sales'
```

```
 'Business Development' 'Business Development' 'Client Services' 'Sales'
 'Distribution' 'Business Development' 'Business Development' 'Legal'
 'Marketing' 'Distribution' 'Client Services' 'Business Development'
 'Engineering' 'Engineering' 'Business Development' 'Client Services'
 'Client Services' 'Client Services' 'Distribution' 'Engineering'
 'Marketing' 'Engineering' 'Distribution' 'Distribution' 'Distribution'
 'Marketing' 'Engineering' 'Business Development' 'Business Development'
 nan 'Human Resources' 'Product' 'Finance' 'Legal' 'Engineering' 'Sales'
 'Engineering' 'Business Development' 'Business Development' 'Legal'
 'Product' 'Sales' 'Sales' 'Client Services' 'Human Resources'
 'Engineering' 'Distribution' 'Engineering' 'Product'
 'Business Development' 'Sales' 'Business Development' 'Client Services'
 'Sales' 'Legal' 'Product' 'Human Resources' 'Product' 'Engineering'
 'Legal' 'Human Resources' 'Engineering' 'Engineering' 'Legal' 'Marketing'
 'Finance' 'Human Resources' 'Legal' 'Client Services' 'Marketing'
 'Finance' 'Engineering' 'Marketing' 'Distribution' 'Finance' 'Product'
 'Business Development' 'Sales']

--------------------------------------------------------------------------------------------------------------
--------------

[0 1 0 ... 1 0 0]
```

In [6]:
```
# step - 3 : to understand data
#to display concise summary of columns
dataset1.info()
print("\n----------------------------------------------------------------------------------------------------
dataset2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   First Name         933 non-null    object
 1   Gender             855 non-null    object
 2   Start Date         1000 non-null   object
 3   Last Login Time    1000 non-null   object
 4   Salary             1000 non-null   int64
 5   Bonus %            1000 non-null   float64
 6   Senior Management  933 non-null    object
 7   Team               957 non-null    object
dtypes: float64(1), int64(1), object(6)
memory usage: 62.6+ KB

--------------------------------------------------------------------------------------------------------------
--------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4653 entries, 0 to 4652
Data columns (total 9 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Education                  4653 non-null   object
 1   JoiningYear                4653 non-null   int64
 2   City                       4653 non-null   object
 3   PaymentTier                4653 non-null   int64
 4   Age                        4653 non-null   int64
 5   Gender                     4653 non-null   object
 6   EverBenched                4653 non-null   object
 7   ExperienceInCurrentDomain  4653 non-null   int64
 8   LeaveOrNot                 4653 non-null   int64
dtypes: int64(5), object(4)
memory usage: 327.3+ KB
```

In [7]:
```
# to display top 20 records dataset1
dataset1.head(20)
```

Out[7]:

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	8/6/1993	12:42 PM	97308	6.945	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	61933	4.170	True	NaN
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	False	Finance
3	Jerry	Male	3/4/2005	1:00 PM	138705	9.340	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	True	Client Services
5	Dennis	Male	4/18/1987	1:35 AM	115163	10.125	False	Legal
6	Ruby	Female	8/17/1987	4:20 PM	65476	10.012	True	Product
7	NaN	Female	7/20/2015	10:43 AM	45906	11.598	NaN	Finance
8	Angela	Female	11/22/2005	6:29 AM	95570	18.523	True	Engineering
9	Frances	Female	8/8/2002	6:51 AM	139852	7.524	True	Business Development
10	Louise	Female	8/12/1980	9:01 AM	63241	15.132	True	NaN
11	Julie	Female	10/26/1997	3:19 PM	102508	12.637	True	Legal
12	Brandon	Male	12/1/1980	1:08 AM	112807	17.492	True	Human Resources
13	Gary	Male	1/27/2008	11:40 PM	109831	5.831	False	Sales
14	Kimberly	Female	1/14/1999	7:13 AM	41426	14.543	True	Finance
15	Lillian	Female	6/5/2016	6:09 AM	59414	1.256	False	Product
16	Jeremy	Male	9/21/2010	5:56 AM	90370	7.369	False	Human Resources
17	Shawn	Male	12/7/1986	7:45 PM	111737	6.414	False	Product
18	Diana	Female	10/23/1981	10:27 AM	132940	19.082	False	Client Services
19	Donna	Female	7/22/2010	3:48 AM	81014	1.894	False	Product

In [8]:
```
# to display top 20 records dataset2
dataset2.head(20)
```

Out[8]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1
5	Bachelors	2016	Bangalore	3	22	Male	No	0	0
6	Bachelors	2015	New Delhi	3	38	Male	No	0	0
7	Bachelors	2016	Bangalore	3	34	Female	No	2	1
8	Bachelors	2016	Pune	3	23	Male	No	1	0
9	Masters	2017	New Delhi	2	37	Male	No	2	0
10	Masters	2012	Bangalore	3	27	Male	No	5	1
11	Bachelors	2016	Pune	3	34	Male	No	3	0
12	Bachelors	2018	Pune	3	32	Male	Yes	5	1
13	Bachelors	2016	Bangalore	3	39	Male	No	2	0
14	Bachelors	2012	Bangalore	3	37	Male	No	4	0
15	Bachelors	2017	Bangalore	1	29	Male	No	3	0
16	Bachelors	2014	Bangalore	3	34	Female	No	2	0
17	Bachelors	2014	Pune	3	34	Male	No	4	0
18	Bachelors	2015	Pune	2	30	Female	No	0	1
19	Bachelors	2016	New Delhi	2	22	Female	No	0	1

In [9]:
```
#to display last 10 records dataset2
dataset2.tail(10)
```

Out[9]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
4643	Bachelors	2013	Bangalore	3	31	Female	No	5	0
4644	Bachelors	2015	Pune	3	32	Female	Yes	1	1
4645	Masters	2017	Pune	2	31	Female	No	2	0
4646	Bachelors	2013	Bangalore	3	25	Female	No	3	0
4647	Bachelors	2016	Pune	3	30	Male	No	2	0
4648	Bachelors	2013	Bangalore	3	26	Female	No	4	0
4649	Masters	2013	Pune	2	37	Male	No	2	1
4650	Masters	2018	New Delhi	3	27	Male	No	5	1
4651	Bachelors	2012	Bangalore	3	30	Male	Yes	2	0
4652	Bachelors	2015	Bangalore	3	33	Male	Yes	4	0

In [10]:
```
#to display any 10 random records dataset2
dataset2.sample(10)
```

Out[10]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
3588	Masters	2013	New Delhi	3	35	Male	No	4	0
241	Bachelors	2017	New Delhi	3	24	Male	No	2	1
2140	Bachelors	2013	New Delhi	3	29	Female	No	1	0
1844	Bachelors	2014	Pune	3	26	Male	No	4	0
1498	Bachelors	2017	Bangalore	3	25	Male	No	3	0
370	Bachelors	2017	Bangalore	3	28	Male	No	3	0
416	Masters	2017	New Delhi	3	28	Male	Yes	2	0
2903	Masters	2017	New Delhi	2	30	Male	No	2	0
245	Bachelors	2014	Pune	2	24	Female	No	2	1
3080	Bachelors	2018	Bangalore	3	36	Male	No	1	1

In [11]:
```
#to display shape of data dataset2
print("Shape of data : ",dataset2.shape)
print("\n----------------------------------------------------------------------------------------------------------------------------")
#to display shape of data dataset1
print("Shape of data : ",dataset1.shape)
```

```
Shape of data :  (4653, 9)

------------------------------------------------------------------------------------------------------------------------------
--------------
Shape of data :  (1000, 8)
```

In [12]:
```
# calculating the no. of unique values in dataset2
dataset2.nunique()
```

Out[12]:
```
Education                    3
JoiningYear                  7
City                         3
PaymentTier                  3
Age                         20
Gender                       2
EverBenched                  2
ExperienceInCurrentDomain    8
LeaveOrNot                   2
dtype: int64
```

In [13]:
```
# calculating the no. of unique values in dataset1
dataset1.nunique()
```

Out[13]:
```
First Name           200
Gender                 2
Start Date           972
Last Login Time      720
Salary               995
Bonus %              971
Senior Management      2
Team                  10
dtype: int64
```

In [14]:
```
#to view entire data dataset2
dataset2.head(dataset2.shape[0])
```

Out[14]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1
...	...	...	...	...	...	...	...	...	...
4648	Bachelors	2013	Bangalore	3	26	Female	No	4	0
4649	Masters	2013	Pune	2	37	Male	No	2	1
4650	Masters	2018	New Delhi	3	27	Male	No	5	1
4651	Bachelors	2012	Bangalore	3	30	Male	Yes	2	0
4652	Bachelors	2015	Bangalore	3	33	Male	Yes	4	0

4653 rows × 9 columns

In [15]:
```
#to view entire data dataset1
dataset1.head(dataset1.shape[0])
```

Out[15]:

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	8/6/1993	12:42 PM	97308	6.945	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	61933	4.170	True	NaN
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	False	Finance
3	Jerry	Male	3/4/2005	1:00 PM	138705	9.340	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	True	Client Services
...	...	...	...	...	...	...	...	...
995	Henry	NaN	11/23/2014	6:09 AM	132483	16.655	False	Distribution
996	Phillip	Male	1/31/1984	6:30 AM	42392	19.675	False	Finance
997	Russell	Male	5/20/2013	12:39 PM	96914	1.421	False	Product
998	Larry	Male	4/20/2013	4:45 PM	60500	11.985	False	Business Development
999	Albert	Male	5/15/2012	6:24 PM	129949	10.169	True	Sales

1000 rows × 8 columns

In [16]:
```
#to display datatype of columns dataset2
print("Column datatypes:")
dataset2.dtypes
```

Column datatypes:

Out[16]:
```
Education                   object
JoiningYear                  int64
City                        object
PaymentTier                  int64
Age                          int64
Gender                      object
EverBenched                 object
ExperienceInCurrentDomain    int64
LeaveOrNot                   int64
dtype: object
```

In [17]:
```
#to display datatype of columns dataset1
print("Column datatypes:")
dataset1.dtypes
```

Column datatypes:

Out[17]:
```
First Name            object
Gender                object
Start Date            object
Last Login Time       object
Salary                 int64
Bonus %              float64
Senior Management     object
Team                  object
dtype: object
```

In [18]:
```
#to fetch duplicate records dataset2
duplicate_record2 = dataset2[dataset2.duplicated()]
print("Duplicate records : ")
print(duplicate_record2)
```

```
Duplicate records :
      Education  JoiningYear       City  PaymentTier  Age  Gender EverBenched  \
111   Bachelors         2017       Pune            2   27  Female          No
130   Bachelors         2017  Bangalore            3   26  Female          No
138   Bachelors         2017  New Delhi            3   28    Male          No
160   Bachelors         2014  Bangalore            3   28  Female          No
167   Bachelors         2014  Bangalore            3   25    Male          No
...         ...          ...        ...          ...  ...     ...         ...
4640  Bachelors         2015  Bangalore            3   35    Male          No
4642  Bachelors         2012  Bangalore            3   36  Female          No
4646  Bachelors         2013  Bangalore            3   25  Female          No
4648  Bachelors         2013  Bangalore            3   26  Female          No
4652  Bachelors         2015  Bangalore            3   33    Male         Yes

      ExperienceInCurrentDomain  LeaveOrNot
111                           5           1
130                           4           0
138                           2           0
160                           3           0
167                           3           0
...                         ...         ...
4640                          0           0
4642                          4           0
4646                          3           0
4648                          4           0
4652                          4           0

[1889 rows x 9 columns]
```

In [19]:
```
#to fetch duplicate records dataset1
duplicate_record1 = dataset1[dataset1.duplicated()]
print("Duplicate records : ")
print(duplicate_record1)
```

```
Duplicate records :
Empty DataFrame
Columns: [First Name, Gender, Start Date, Last Login Time, Salary, Bonus %, Senior Management, Team]
Index: []
```

In [20]:
```
#to drop duplicate record and obtain unique record
dataset2_unique = dataset2.drop_duplicates()
print("Uinque records : ")
print(dataset2_unique)
```

```
Uinque records :
      Education  JoiningYear       City  PaymentTier  Age  Gender EverBenched  \
0     Bachelors         2017  Bangalore            3   34    Male          No
1     Bachelors         2013       Pune            1   28  Female          No
2     Bachelors         2014  New Delhi            3   38  Female          No
3       Masters         2016  Bangalore            3   27    Male          No
4       Masters         2017       Pune            3   24    Male         Yes
...         ...          ...        ...          ...  ...     ...         ...
4645    Masters         2017       Pune            2   31  Female          No
4647  Bachelors         2016       Pune            3   30    Male          No
4649    Masters         2013       Pune            2   37    Male          No
4650    Masters         2018  New Delhi            3   27    Male          No
4651  Bachelors         2012  Bangalore            3   30    Male         Yes

      ExperienceInCurrentDomain  LeaveOrNot
0                             0           0
1                             3           1
2                             2           0
3                             5           1
4                             2           1
...                         ...         ...
4645                          2           0
4647                          2           0
4649                          2           1
4650                          5           1
4651                          2           0

[2764 rows x 9 columns]
```

In [21]:
```
#to drop duplicate record and obtain unique record dataset1
dataset1_unique = dataset1.drop_duplicates()
print("Uinque records : ")
print(dataset1_unique)
```

```
Uinque records :
    First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male    8/6/1993        12:42 PM   97308    6.945
1       Thomas    Male   3/31/1996         6:53 AM   61933    4.170
2        Maria  Female   4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male    3/4/2005         1:00 PM  138705    9.340
4        Larry    Male   1/24/1998         4:47 PM  101004    1.389
..         ...     ...         ...             ...     ...      ...
995      Henry     NaN  11/23/2014         6:09 AM  132483   16.655
996    Phillip    Male   1/31/1984         6:30 AM   42392   19.675
997    Russell    Male   5/20/2013        12:39 PM   96914    1.421
998      Larry    Male   4/20/2013         4:45 PM   60500   11.985
999     Albert    Male   5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
1                True                  NaN
2               False              Finance
3                True              Finance
4                True       Client Services
..                ...                  ...
995             False         Distribution
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[1000 rows x 8 columns]
```

In [22]:
```
# as we just identified that our one data set have null values and  We need to handle these null values

# Now at first we need to identify how many columns are there which I have missing values

# 1. Identify Columns with Missing Values (in the COPY):

cols_with_missing = dataset1.columns[dataset1.isnull().any()]
print("Columns with missing values (in empoyees.csv):\n", cols_with_missing)
print("Number of missing values per column (in empoyees.csv):\n", dataset1[cols_with_missing].isnull().sum())
```

```
Columns with missing values (in empoyees.csv):
 Index(['First Name', 'Gender', 'Senior Management', 'Team'], dtype='object')
Number of missing values per column (in empoyees.csv):
 First Name            67
Gender                145
Senior Management      67
Team                   43
dtype: int64
```

In [23]:
```
# --- Handle Missing Values (NULLs) on copy dataset ---
#Strategies for Handling Missing Values (in the COPY):

# a) Remove Rows with ANY Missing Values (it is the Simplest, but can lose data):
dataset1_no_rows = dataset1.dropna()
print("dataset after nul data removal:\n", dataset1_no_rows)
```

```
dataset after nul data removal:
     First Name  Gender Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male   8/6/1993        12:42 PM   97308    6.945
2        Maria  Female  4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male   3/4/2005         1:00 PM  138705    9.340
4        Larry    Male  1/24/1998         4:47 PM  101004    1.389
5       Dennis    Male  4/18/1987         1:35 AM  115163   10.125
..         ...     ...         ...             ...     ...      ...
994     George    Male  6/21/2013         5:47 PM   98874    4.479
996    Phillip    Male  1/31/1984         6:30 AM   42392   19.675
997    Russell    Male  5/20/2013        12:39 PM   96914    1.421
998      Larry    Male  4/20/2013         4:45 PM   60500   11.985
999     Albert    Male  5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
2               False              Finance
3                True              Finance
4                True       Client Services
5               False                Legal
..                ...                  ...
994              True            Marketing
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[764 rows x 8 columns]
```

In [24]:
```
# 2. Remove columns with ALL missing values:
# Using this method we are losing a lot of data
dataset1_no_cols = dataset1.dropna(axis=1, how='all')
print("\ndataset after nul data removal\n", dataset1_no_cols)
```

```
dataset after nul data removal
     First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male    8/6/1993        12:42 PM   97308    6.945
1       Thomas    Male   3/31/1996         6:53 AM   61933    4.170
2        Maria  Female   4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male    3/4/2005         1:00 PM  138705    9.340
4        Larry    Male   1/24/1998         4:47 PM  101004    1.389
..         ...     ...         ...             ...     ...      ...
995      Henry     NaN  11/23/2014         6:09 AM  132483   16.655
996    Phillip    Male   1/31/1984         6:30 AM   42392   19.675
997    Russell    Male   5/20/2013        12:39 PM   96914    1.421
998      Larry    Male   4/20/2013         4:45 PM   60500   11.985
999     Albert    Male   5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
1                True                  NaN
2               False              Finance
3                True              Finance
4                True       Client Services
..                ...                  ...
995             False         Distribution
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[1000 rows x 8 columns]
```

In [25]:
```python
# Impute with mode for string/object type columns
pd.set_option('future.no_silent_downcasting', True)
for col in dataset1.columns:
    if dataset1[col].dtype == 'object':  # Check if the column is of object type (string)
        mode_value = dataset1[col].mode()[0]  # Get the first mode (in case of ties)
        dataset1_cleaned_mode = dataset1[col].fillna(mode_value)

print("Original Dataset:\n",dataset1)
print("\n-----------------------------------------------------------------------------------------------------------------------")
print("\nDataset after Mode Imputation:\n", dataset1_cleaned_mode)

#Demonstrate mode_value
for col in dataset1.columns:
    if dataset1[col].dtype == 'object':
        mode_value = dataset1[col].mode()[0]

        print(f"Mode for column {col} is : {mode_value}")
```

```
Original Dataset:
     First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
0      Douglas    Male    8/6/1993        12:42 PM   97308    6.945
1       Thomas    Male   3/31/1996         6:53 AM   61933    4.170
2        Maria  Female   4/23/1993        11:17 AM  130590   11.858
3        Jerry    Male    3/4/2005         1:00 PM  138705    9.340
4        Larry    Male   1/24/1998         4:47 PM  101004    1.389
..         ...     ...         ...             ...     ...      ...
995      Henry     NaN  11/23/2014         6:09 AM  132483   16.655
996    Phillip    Male   1/31/1984         6:30 AM   42392   19.675
997    Russell    Male   5/20/2013        12:39 PM   96914    1.421
998      Larry    Male   4/20/2013         4:45 PM   60500   11.985
999     Albert    Male   5/15/2012         6:24 PM  129949   10.169

    Senior Management                 Team
0                True            Marketing
1                True                  NaN
2               False              Finance
3                True              Finance
4                True       Client Services
..                ...                  ...
995             False         Distribution
996             False              Finance
997             False              Product
998             False  Business Development
999              True                Sales

[1000 rows x 8 columns]
--------------------------------------------------------------------------------------------------------------------------------
--------------


Dataset after Mode Imputation:
 0                 Marketing
1           Client Services
2                   Finance
3                   Finance
4           Client Services
               ...
995            Distribution
996                 Finance
997                 Product
998    Business Development
999                   Sales
Name: Team, Length: 1000, dtype: object
Mode for column First Name is : Marilyn
Mode for column Gender is : Female
Mode for column Start Date is : 1/26/2005
Mode for column Last Login Time is : 1:35 PM
Mode for column Senior Management is : True
Mode for column Team is : Client Services
```

In [26]:
```
# now we need to do basic statistical analysis for numeric columns
# for all statistical analysis we are going to use dataset1_no_rows As our data set
# because in this one we have removed all the row which contained null values

dataset1_no_rows.describe()
```

Out[26]:

	Salary	Bonus %
count	764.000000	764.000000
mean	90433.196335	10.148041
std	32864.665282	5.608733
min	35013.000000	1.015000
25%	62071.750000	5.193250
50%	90428.000000	9.658500
75%	118075.250000	14.965000
max	149908.000000	19.944000

In [27]:
```
# And we are also going to use dataset2 because this one contains more numerical values

dataset2.describe()
```

Out[27]:

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.000000	4653.000000	4653.000000	4653.000000	4653.000000
mean	2015.062970	2.698259	29.393295	2.905652	0.343864
std	1.863377	0.561435	4.826087	1.558240	0.475047
min	2012.000000	1.000000	22.000000	0.000000	0.000000
25%	2013.000000	3.000000	26.000000	2.000000	0.000000
50%	2015.000000	3.000000	28.000000	3.000000	0.000000
75%	2017.000000	3.000000	32.000000	4.000000	1.000000
max	2018.000000	3.000000	41.000000	7.000000	1.000000

In [28]:
```
#basic statistical analysis for categorial columns
dataset2.describe(include='all')
```

Out[28]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	Leave
count	4653	4653.000000	4653	4653.000000	4653.000000	4653	4653	4653.000000	4653.0
unique	3	NaN	3	NaN	NaN	2	2	NaN	
top	Bachelors	NaN	Bangalore	NaN	NaN	Male	No	NaN	
freq	3601	NaN	2228	NaN	NaN	2778	4175	NaN	
mean	NaN	2015.062970	NaN	2.698259	29.393295	NaN	NaN	2.905652	0.3
std	NaN	1.863377	NaN	0.561435	4.826087	NaN	NaN	1.558240	0.4
min	NaN	2012.000000	NaN	1.000000	22.000000	NaN	NaN	0.000000	0.0
25%	NaN	2013.000000	NaN	3.000000	26.000000	NaN	NaN	2.000000	0.0
50%	NaN	2015.000000	NaN	3.000000	28.000000	NaN	NaN	3.000000	0.0
75%	NaN	2017.000000	NaN	3.000000	32.000000	NaN	NaN	4.000000	1.0
max	NaN	2018.000000	NaN	3.000000	41.000000	NaN	NaN	7.000000	1.0

In [29]:
```
#basic statistical analysis for categorial columns
dataset1_no_rows.describe() .describe(include='all')
```

Out[29]:

	Salary	Bonus %
count	8.000000	8.000000
mean	72444.732702	103.816565
std	49198.035932	266.820006
min	764.000000	1.015000
25%	34475.916321	5.504862
50%	76249.875000	9.903270
75%	97343.709751	16.209750
max	149908.000000	764.000000

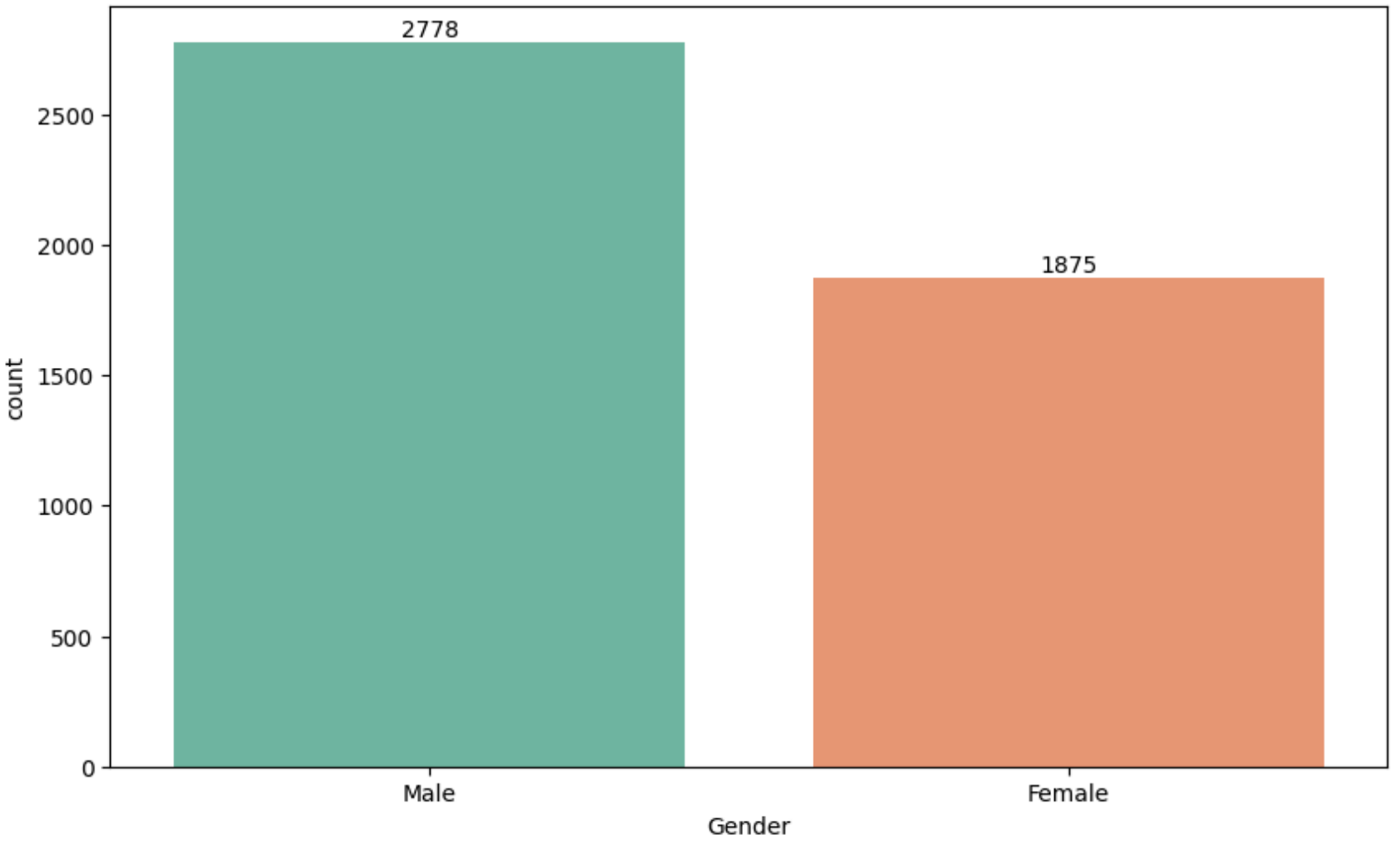
In [30]:
```
#finding missing values
dataset2.isnull().sum()
```

Out[30]:
```
Education                   0
JoiningYear                 0
City                        0
PaymentTier                 0
Age                         0
Gender                      0
EverBenched                 0
ExperienceInCurrentDomain   0
LeaveOrNot                  0
dtype: int64
```

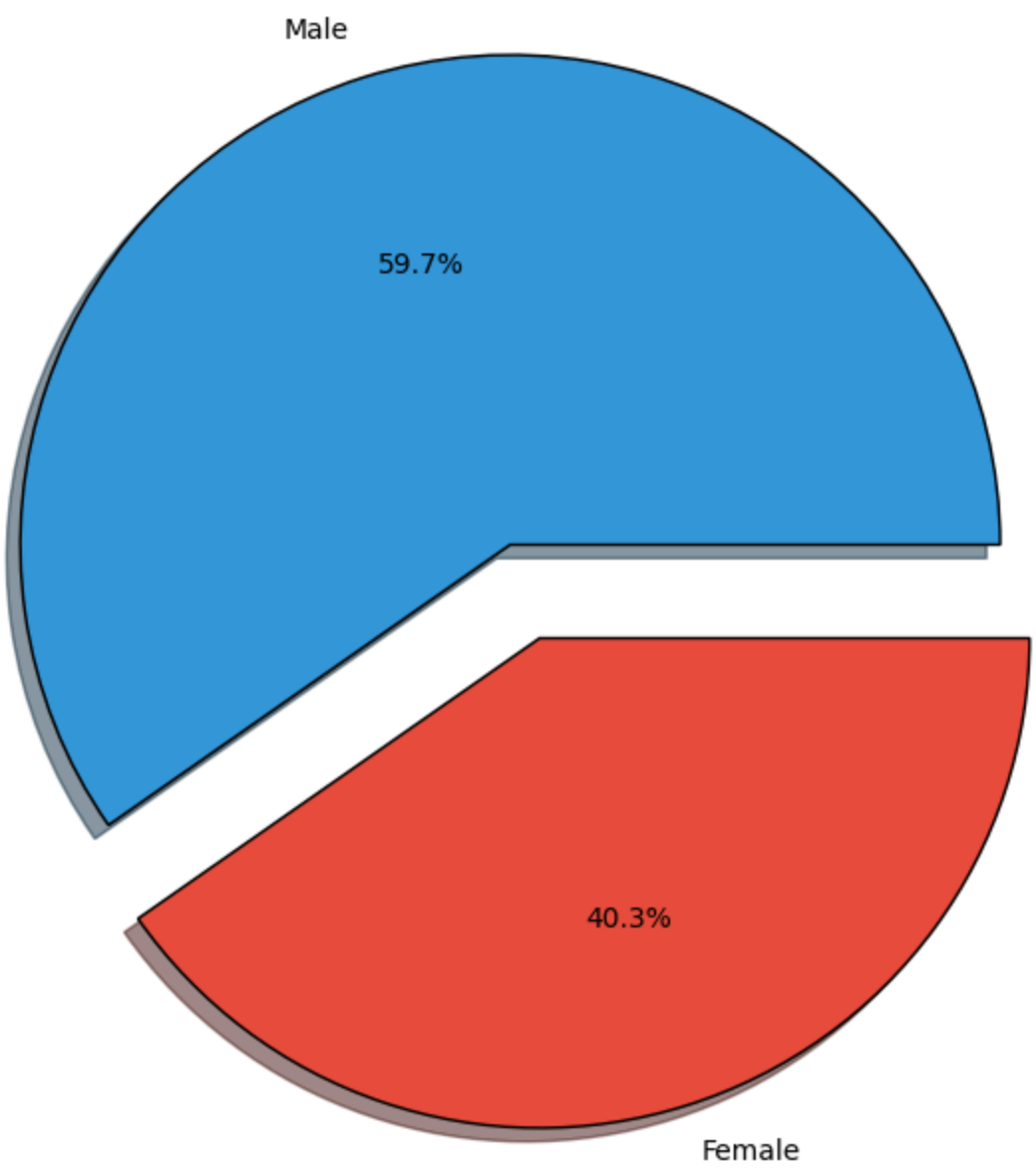
In [31]:
```
#finding missing values
dataset1_no_rows.isnull().sum()
```

Out[31]:
```
First Name           0
Gender               0
Start Date           0
Last Login Time      0
Salary               0
Bonus %              0
Senior Management    0
Team                 0
dtype: int64
```

In [34]:
```
# starting data visualization
# from here we are going to use dataset2 and dataset1_no_rows becouse dataset2 did not required any refinement
# and dataset1 required it hence the refined versoin of dataset1 named as dataset1_no_rows wiil be used
plt.figure(figsize=(10, 6))

ax = sns.countplot(data=dataset2, x='Gender', hue='Gender', palette='Set2', legend=False)

# Add labels to ALL bars the containers take 0 value as default and provides only first container no.
for container in ax.containers:
    ax.bar_label(container)

plt.title('Distribusi Gender Karyawan')
plt.show()
```



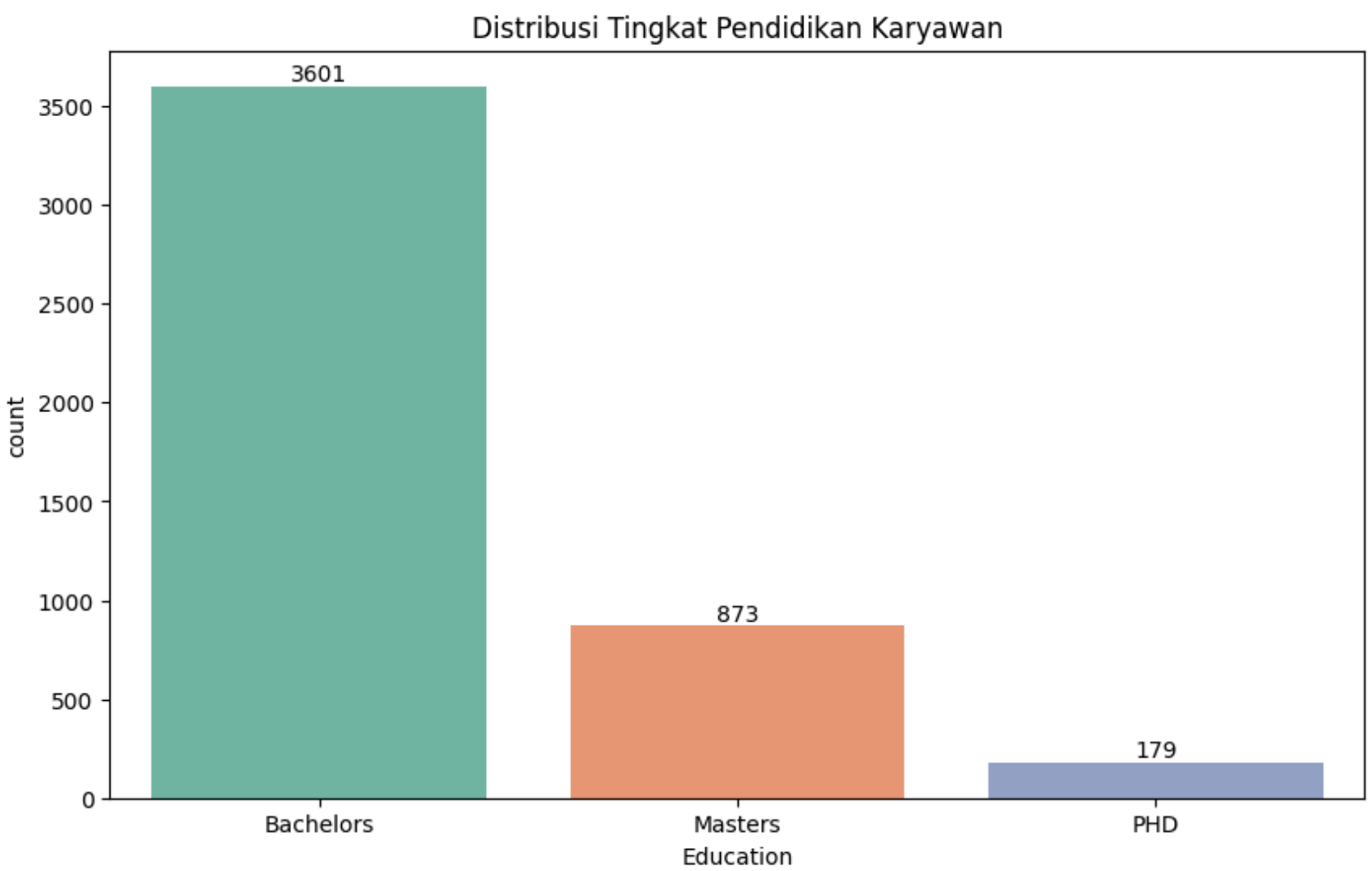
In [35]:
```
#making a circuler chart
edu_count = dataset2['Gender'].value_counts()

explode = [0.1] * len(edu_count)

plt.figure(figsize=(8, 8))
plt.pie(edu_count, labels=edu_count.index, shadow=True, autopct='%1.1f%%', explode=explode, colors=['#3498db', '#e74
plt.title('Distribusi Gender Karyawan')

plt.show()
```



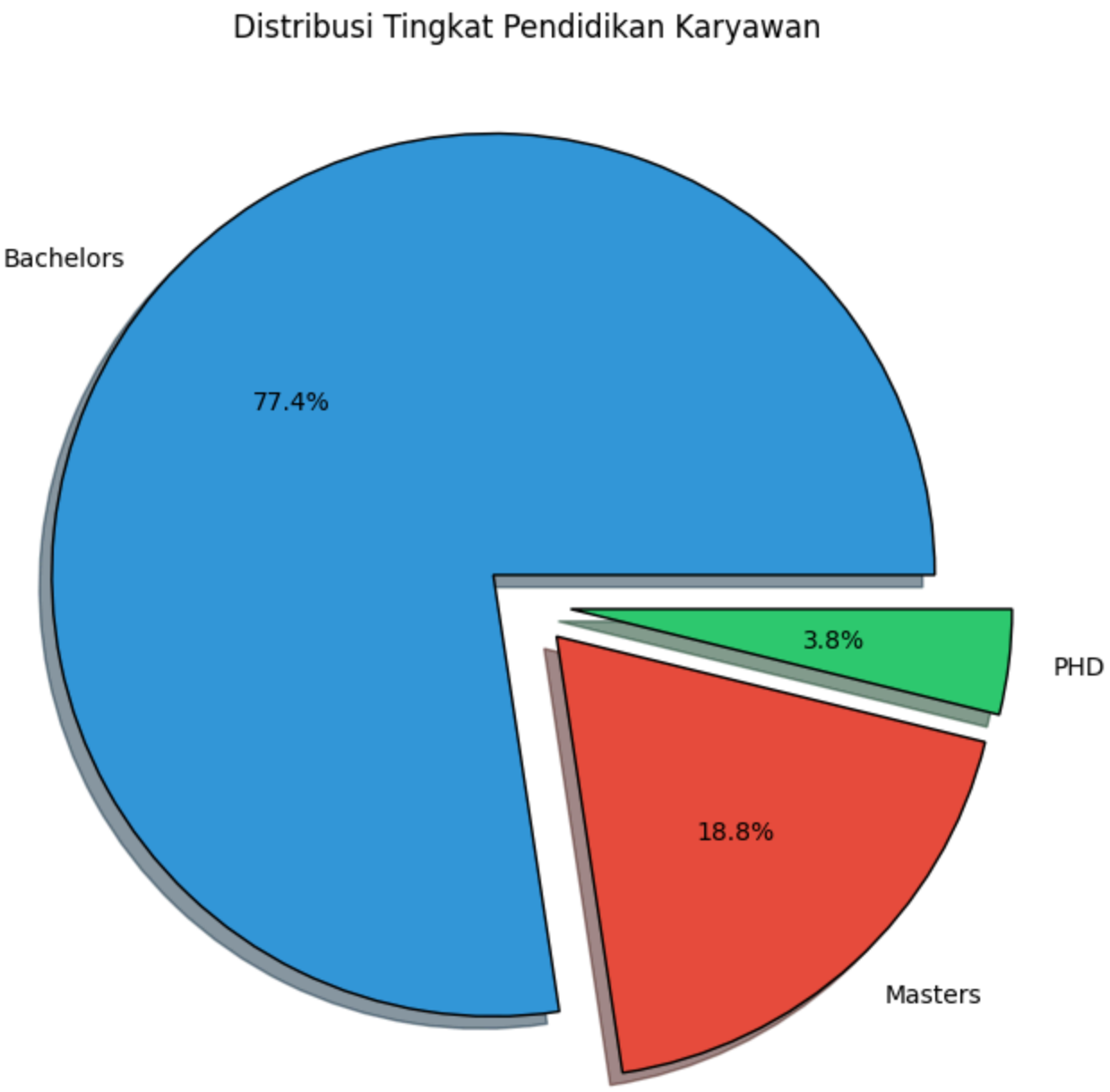
In [37]:
```
dataset2.Education.value_counts()
```

Out[37]:
```
Education
Bachelors    3601
Masters       873
PHD           179
Name: count, dtype: int64
```

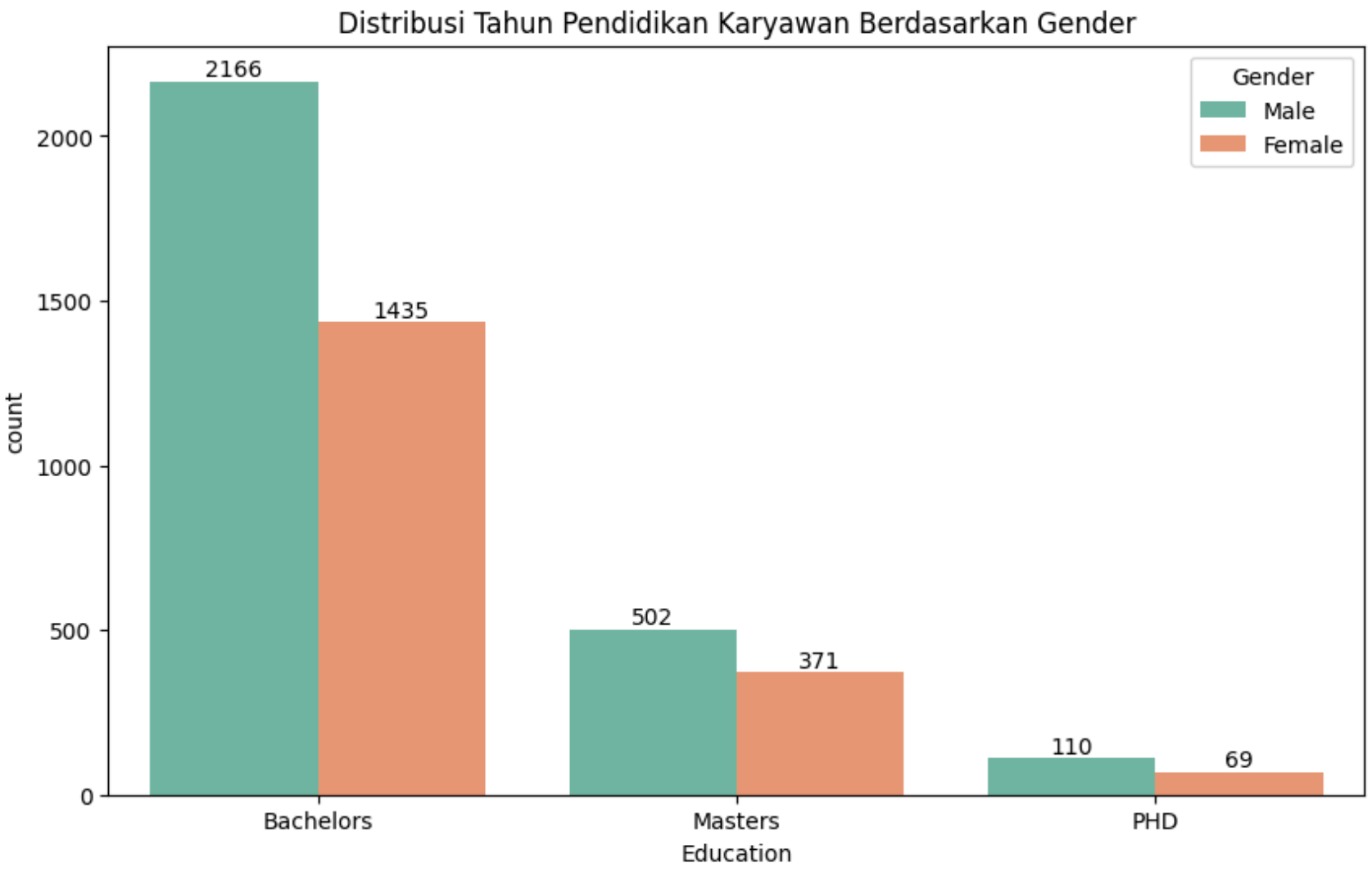
In [42]:
```
plt.figure(figsize=(10, 6))
ax = sns.countplot(data=dataset2, x='Education', hue='Education', palette='Set2', legend=False)

# Add labels to ALL bars the containers take 0 value as default and provides only first container no.
for container in ax.containers:
    ax.bar_label(container)

plt.title('Distribusi Tingkat Pendidikan Karyawan')
plt.show()
```



In [43]:
```
edu_count = dataset2['Education'].value_counts()

explode = [0.1] * len(edu_count)

plt.figure(figsize=(8, 8))
plt.pie(edu_count, labels=edu_count.index, shadow=True, autopct='%1.1f%%', explode=explode, colors=['#3498db', '#e74c
plt.title('Distribusi Tingkat Pendidikan Karyawan')

plt.show()
```



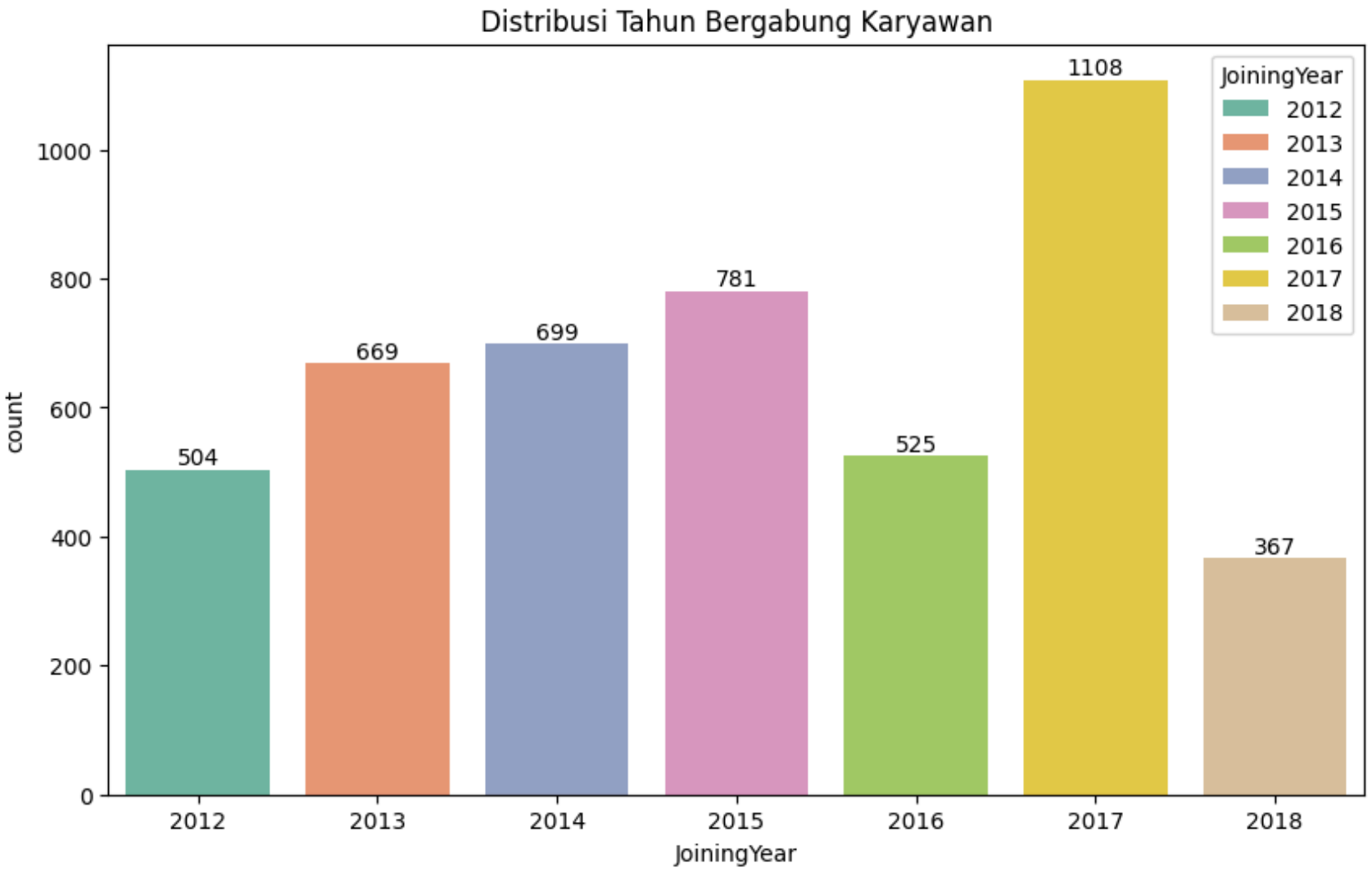
In [44]:
```
plt.figure(figsize=(10, 6))
ax = sns.countplot(data=dataset2,x=dataset2.Education,hue='Gender', palette='Set2')

ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])

plt.title('Distribusi Tahun Pendidikan Karyawan Berdasarkan Gender')
plt.show()
```



In [46]:
```
dataset2.JoiningYear.value_counts()
```

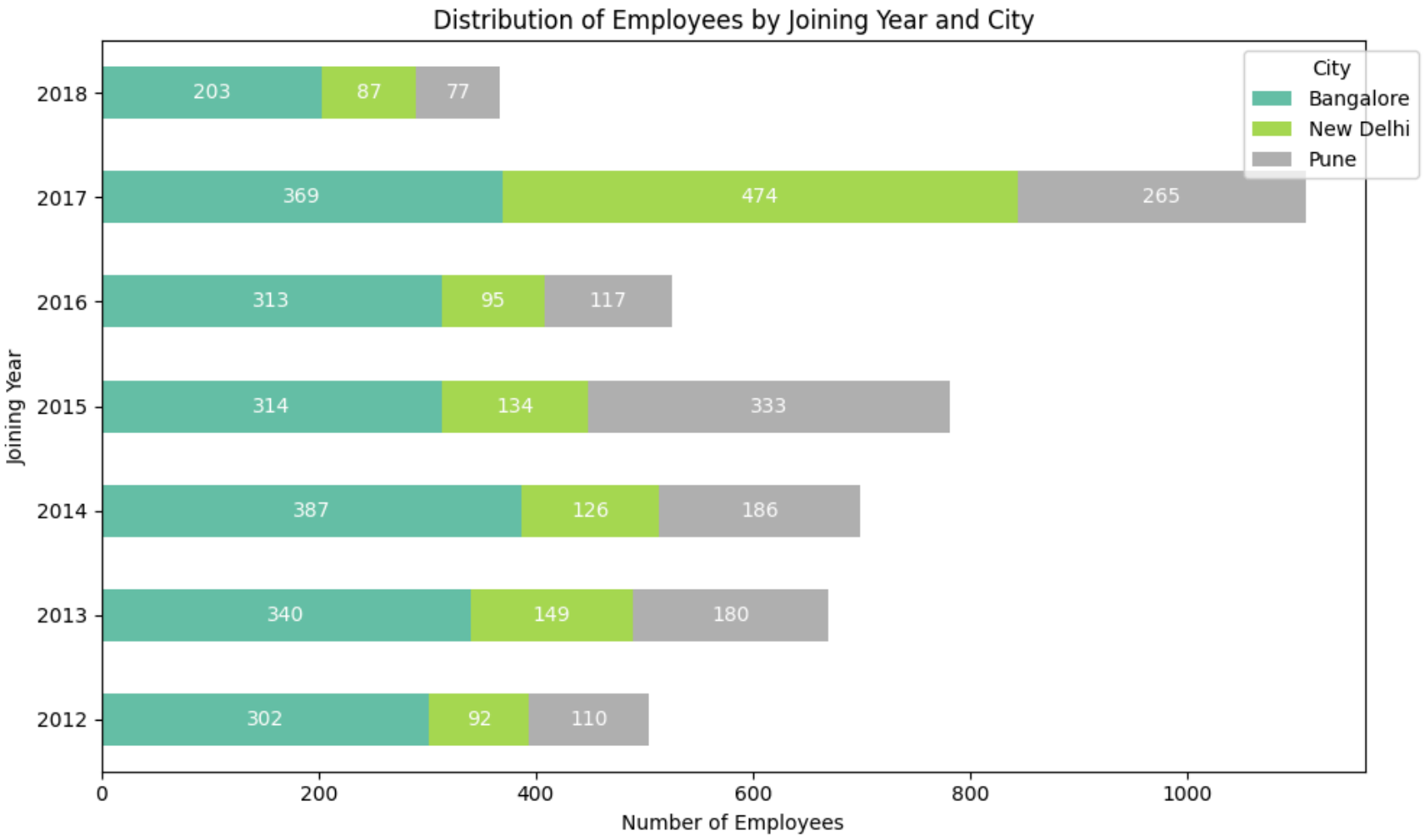
Out[46]:
```
JoiningYear
2017    1108
2015     781
2014     699
2013     669
2016     525
2012     504
2018     367
Name: count, dtype: int64
```

In [48]:
```
plt.figure(figsize=(10, 6))
ax = sns.countplot(data=dataset2, x='JoiningYear' , hue="JoiningYear", palette="Set2")

# Add labels to ALL bars the containers take 0 value as default and provides only first container no.
for container in ax.containers:
    ax.bar_label(container)

plt.title('Distribusi Tahun Bergabung Karyawan')
plt.show()
```



In [49]:
```
city_joinyear = dataset2.groupby(['JoiningYear', 'City']).size().unstack()
```

```
ax = city_joinyear.plot(kind='barh', stacked=True, figsize=(10, 6), colormap='Set2')

for container in ax.containers:
    ax.bar_label(container, label_type='center', fontsize=10, color='white')

plt.xlabel('Number of Employees')
plt.ylabel('Joining Year')
plt.title('Distribution of Employees by Joining Year and City')

plt.legend(title='City', loc='upper right', bbox_to_anchor=(1.05, 1))

plt.tight_layout()
plt.show()
```

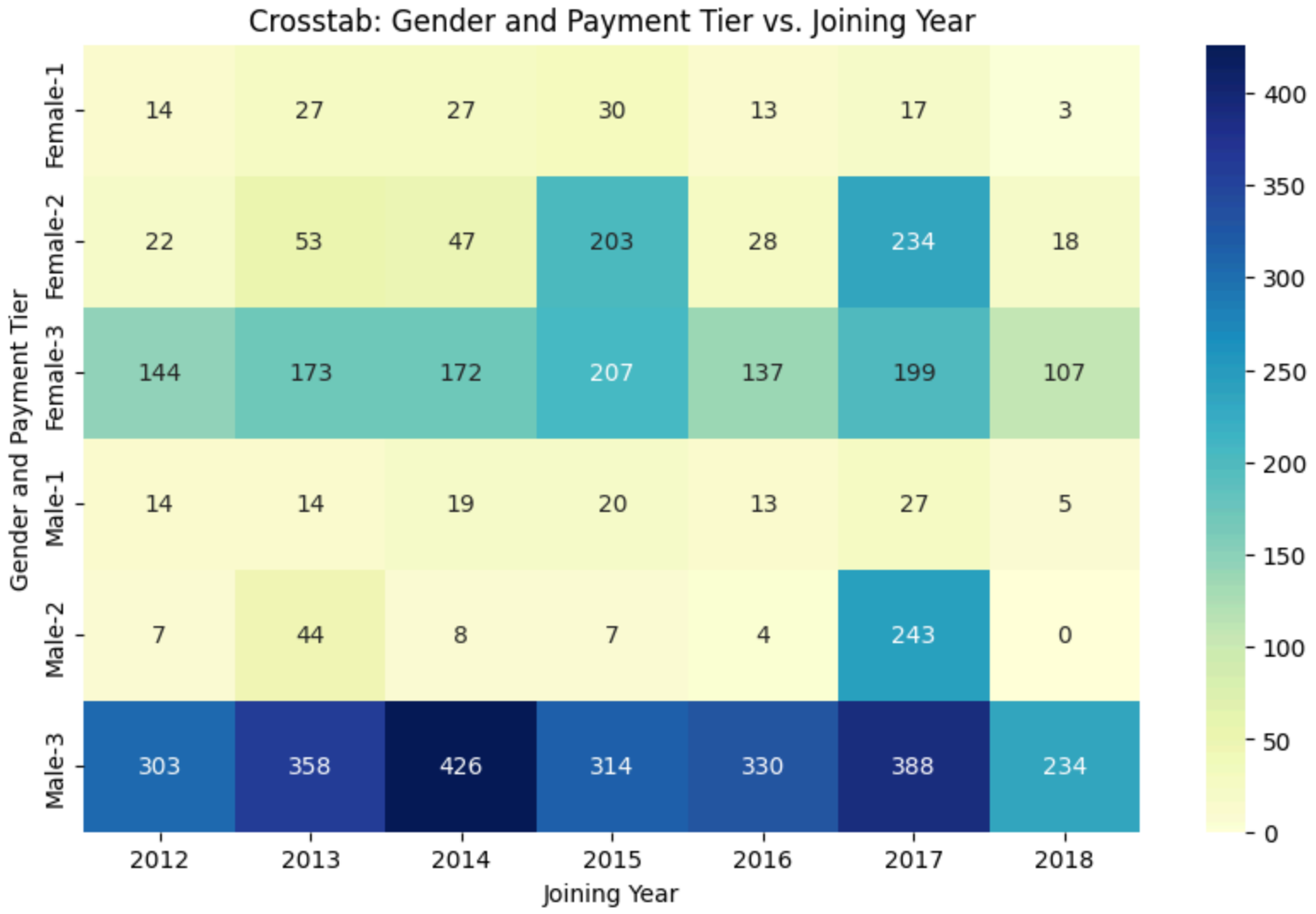


In [53]:
```
pd.crosstab(dataset2.PaymentTier,dataset2.Gender)
```

Out[53]:

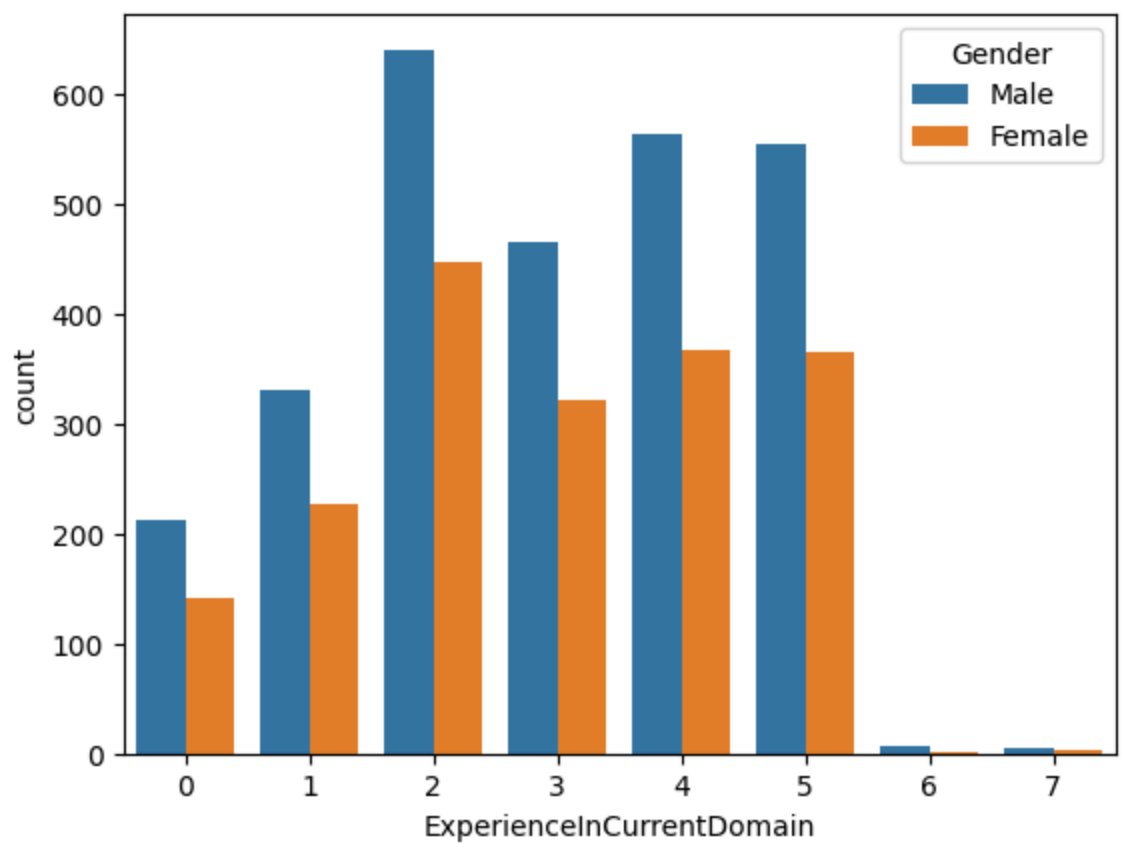
Gender	Female	Male
PaymentTier		
1	131	112
2	605	313
3	1139	2353

In [54]:
```
x=pd.crosstab([dataset2.Gender,dataset2.PaymentTier],dataset2.JoiningYear)
plt.figure(figsize=(10, 6))
sns.heatmap(x, annot=True, fmt='d', cmap='YlGnBu')
plt.xlabel('Joining Year')
plt.ylabel('Gender and Payment Tier')
plt.title('Crosstab: Gender and Payment Tier vs. Joining Year')
plt.show()
```

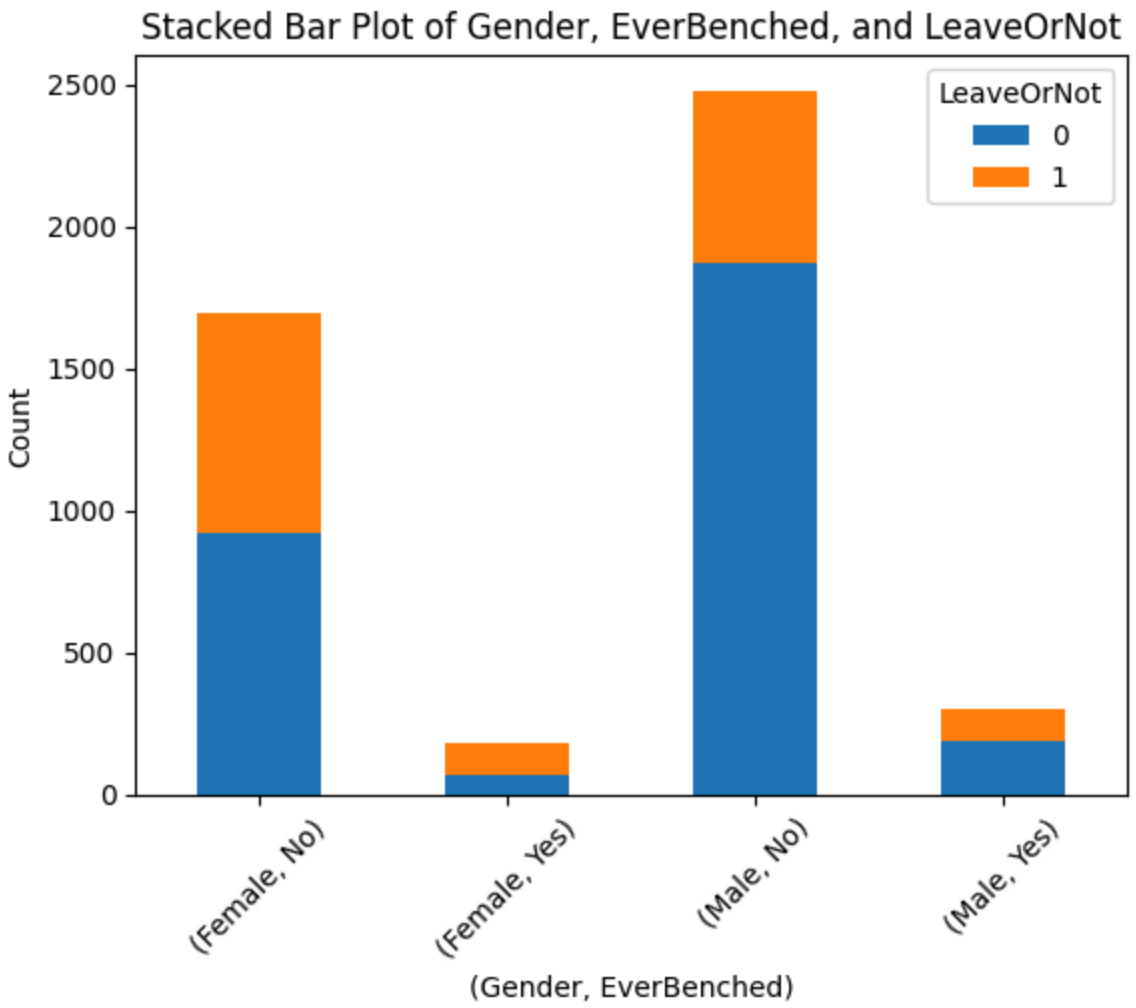


In [61]:
```
sns.countplot(data=dataset2,x='ExperienceInCurrentDomain',hue='Gender')
```

Out[61]: <Axes: xlabel='ExperienceInCurrentDomain', ylabel='count'>



In [58]:
```
x=pd.crosstab([dataset2.Gender,dataset2.EverBenched],dataset2.LeaveOrNot)
x
x.plot(kind='bar', stacked=True)
plt.title('Stacked Bar Plot of Gender, EverBenched, and LeaveOrNot')
plt.xlabel('(Gender, EverBenched)')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

# Show the plot
plt.show()
```


Stacked Bar Plot of Gender, EverBenched, and LeaveOrNot
LeaveOrNot
0
1
Count
2500
2000
1500
1000
500
0
(Female, No)
(Female, Yes)
(Male, No)
(Male, Yes)
(Gender, EverBenched)

In [ ]:

In [ ]: