In [1]:

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

In [2]:

```
emp = pd.read_excel('C:/Users/ADMIN/Employee
Project/Data/INX_Future_Inc_Employee_Performance_CDS_Project2_Data_V1.8.xls')
```

In [3]:

```
emp
```

Out[3]:

	EmpNumber	Age	Gender	EducationBackground	MaritalStatus	EmpDepartment	EmpJobRole	BusinessTravelFrequency	Distan
0	E1001000	32	Male	Marketing	Single	Sales	Sales Executive	Travel_Rarely	
1	E1001006	47	Male	Marketing	Single	Sales	Sales Executive	Travel_Rarely	
2	E1001007	40	Male	Life Sciences	Married	Sales	Sales Executive	Travel_Frequently	
3	E1001009	41	Male	Human Resources	Divorced	Human Resources	Manager	Travel_Rarely	
4	E1001010	60	Male	Marketing	Single	Sales	Sales Executive	Travel_Rarely	
...	...	...	...	...	...	...	...	...	
1195	E100992	27	Female	Medical	Divorced	Sales	Sales Executive	Travel_Frequently	
1196	E100993	37	Male	Life Sciences	Single	Development	Senior Developer	Travel_Rarely	
1197	E100994	50	Male	Medical	Married	Development	Senior Developer	Travel_Rarely	
1198	E100995	34	Female	Medical	Single	Data Science	Data Scientist	Travel_Rarely	
1199	E100998	24	Female	Life Sciences	Single	Sales	Sales Executive	Travel_Rarely	

1200 rows × 28 columns

In [4]:

```
emp.describe()
```

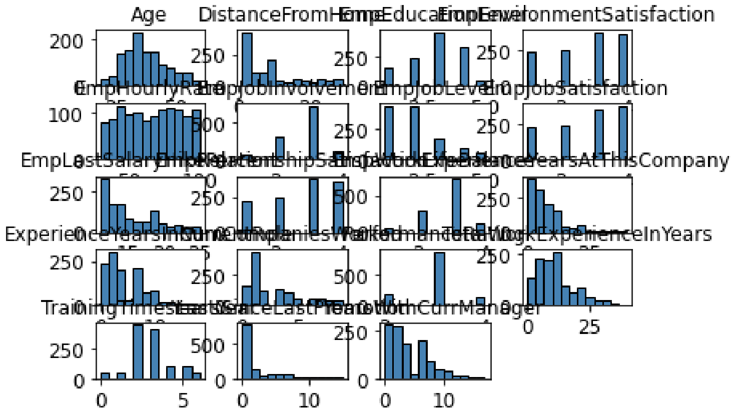
Out[4]:

	Age	DistanceFromHome	EmpEducationLevel	EmpEnvironmentSatisfaction	EmpHourlyRate	EmpJobInvolvement	EmpJo
count	1200.000000	1200.000000	1200.00000	1200.000000	1200.000000	1200.000000	1200.
mean	36.918333	9.165833	2.89250	2.715833	65.981667	2.731667	2.
std	9.087289	8.176636	1.04412	1.090599	20.211302	0.707164	1.
min	18.000000	1.000000	1.00000	1.000000	30.000000	1.000000	1.
25%	30.000000	2.000000	2.00000	2.000000	48.000000	2.000000	1.
50%	36.000000	7.000000	3.00000	3.000000	66.000000	3.000000	2.
75%	43.000000	14.000000	4.00000	4.000000	83.000000	3.000000	3.
max	60.000000	29.000000	5.00000	4.000000	100.000000	4.000000	5.

In [5]:

```
emp.hist(bins=12, color='steelblue', edgecolor='black', linewidth=1.0,
           xlabelsize=12, ylabelsize=12, grid=False)
plt.tight_layout(rect=(0, 0,2.5, 2.5))
```

```
<ipython-input-5-6f44faa63db0>:3: UserWarning: Tight layout not applied. tight_layout cannot make
axes width small enough to accommodate all axes decorations
  plt.tight_layout(rect=(0, 0,2.5, 2.5))
```



In [6]:

```
emp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1200 entries, 0 to 1199
Data columns (total 28 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   EmpNumber                     1200 non-null   object
 1   Age                           1200 non-null   int64 
 2   Gender                        1200 non-null   object
 3   EducationBackground           1200 non-null   object
 4   MaritalStatus                 1200 non-null   object
 5   EmpDepartment                 1200 non-null   object
 6   EmpJobRole                    1200 non-null   object
 7   BusinessTravelFrequency       1200 non-null   object
 8   DistanceFromHome              1200 non-null   int64 
 9   EmpEducationLevel             1200 non-null   int64 
 10  EmpEnvironmentSatisfaction    1200 non-null   int64 
 11  EmpHourlyRate                 1200 non-null   int64 
 12  EmpJobInvolvement             1200 non-null   int64 
 13  EmpJobLevel                   1200 non-null   int64 
 14  EmpJobSatisfaction            1200 non-null   int64 
 15  NumCompaniesWorked            1200 non-null   int64 
 16  OverTime                      1200 non-null   object
 17  EmpLastSalaryHikePercent      1200 non-null   int64 
 18  EmpRelationshipSatisfaction   1200 non-null   int64 
 19  TotalWorkExperienceInYears    1200 non-null   int64 
 20  TrainingTimesLastYear         1200 non-null   int64 
 21  EmpWorkLifeBalance            1200 non-null   int64 
 22  ExperienceYearsAtThisCompany  1200 non-null   int64 
 23  ExperienceYearsInCurrentRole  1200 non-null   int64 
 24  YearsSinceLastPromotion       1200 non-null   int64 
 25  YearsWithCurrManager          1200 non-null   int64 
 26  Attrition                     1200 non-null   object
 27  PerformanceRating             1200 non-null   int64 
dtypes: int64(19), object(9)
memory usage: 262.6+ KB
```

In [7]:

```
emp.shape
```

Out[7]:

```
(1200, 28)
```

In [8]:

```
emp.columns
```

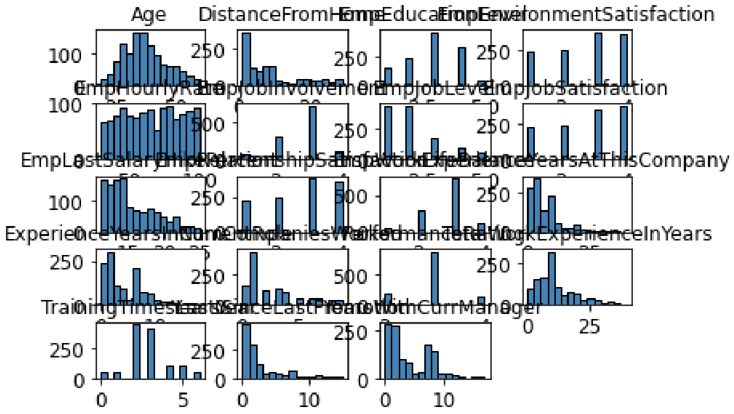
Out[8]:

```
Index(['EmpNumber', 'Age', 'Gender', 'EducationBackground', 'MaritalStatus',
       'EmpDepartment', 'EmpJobRole', 'BusinessTravelFrequency',
       'DistanceFromHome', 'EmpEducationLevel', 'EmpEnvironmentSatisfaction',
       'EmpHourlyRate', 'EmpJobInvolvement', 'EmpJobLevel',
       'EmpJobSatisfaction', 'NumCompaniesWorked', 'OverTime',
       'EmpLastSalaryHikePercent', 'EmpRelationshipSatisfaction',
       'TotalWorkExperienceInYears', 'TrainingTimesLastYear',
       'EmpWorkLifeBalance', 'ExperienceYearsAtThisCompany',
       'ExperienceYearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager', 'Attrition', 'PerformanceRating'],
      dtype='object')
```

In [9]:

```
emp.hist(bins=15, color='steelblue', edgecolor='black', linewidth=1.0,
           xlabelsize=12, ylabelsize=12, grid=False)
plt.tight_layout(rect=(0, 0, 2.5, 2.5))
```

```
<ipython-input-9-acacbf7c0ca5>:3: UserWarning: Tight layout not applied. tight_layout cannot make
axes width small enough to accommodate all axes decorations
  plt.tight_layout(rect=(0, 0, 2.5, 2.5))
```

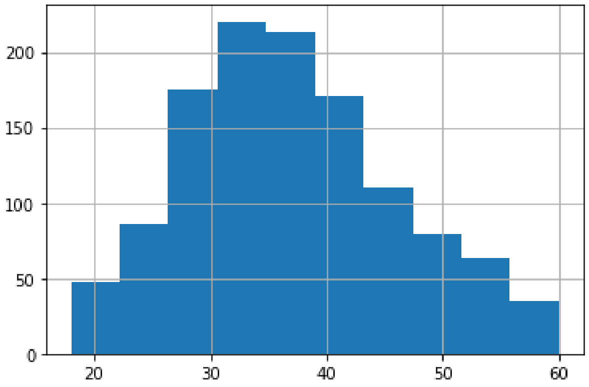


In [10]:

```
#as we can ND of sample and population matches so our sample taken is in ND
emp['Age'].hist()
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124758640>
```

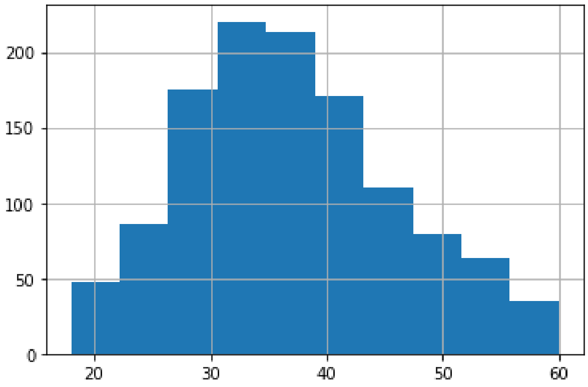


In [11]:

```
emp['Age'].hist()
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124e58160>
```



In [12]:

```
emp['Age'].describe()
```

Out[12]:

```
count    1200.000000
mean       36.918333
std         9.087289
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```

In [13]:

```
emp.describe()
```

Out[13]:

	Age	DistanceFromHome	EmpEducationLevel	EmpEnvironmentSatisfaction	EmpHourlyRate	EmpJobInvolvement	EmpJc
count	1200.000000	1200.000000	1200.00000	1200.000000	1200.000000	1200.000000	1200.
mean	36.918333	9.165833	2.89250	2.715833	65.981667	2.731667	2.
std	9.087289	8.176636	1.04412	1.090599	20.211302	0.707164	1.
min	18.000000	1.000000	1.00000	1.000000	30.000000	1.000000	1.
25%	30.000000	2.000000	2.00000	2.000000	48.000000	2.000000	1.
50%	36.000000	7.000000	3.00000	3.000000	66.000000	3.000000	2.
75%	43.000000	14.000000	4.00000	4.000000	83.000000	3.000000	3.
max	60.000000	29.000000	5.00000	4.000000	100.000000	4.000000	5.

In [14]:

```
emp1 = emp[emp['PerformanceRating']==2]
print(emp1.shape)
```

```
(194, 28)
```

In [15]:

```
emp1['PerformanceRating'].value_counts()
```

```
emp1['PerformanceRating'].value_counts()
```

Out[15]:

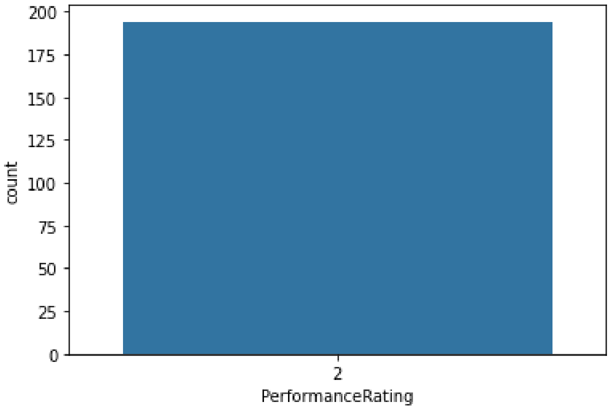
```
2    194
Name: PerformanceRating, dtype: int64
```

In [16]:

```
sns.countplot(emp1['PerformanceRating'])
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23123fdc640>
```



In [17]:

```
emp['Age'].value_counts()
```

Out[17]:

```
34    71
35    64
36    60
31    57
29    51
38    48
32    46
33    46
40    46
27    43
30    42
37    41
28    39
42    36
45    36
26    33
41    32
39    31
44    30
50    28
43    26
46    24
25    24
49    21
24    20
47    20
55    17
54    16
48    16
52    15
53    15
22    15
51    14
56    11
58    11
21    11
23     9
```

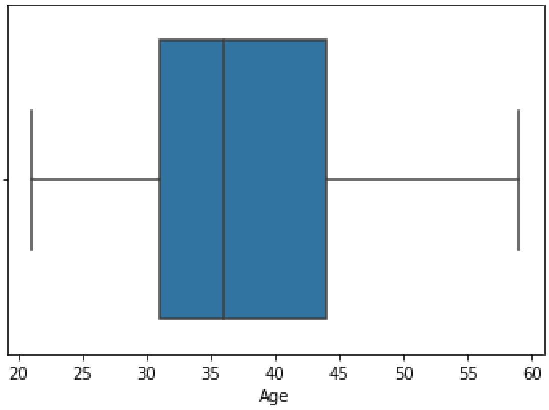
```
19      8
18      8
59      6
20      6
57      4
60      3
Name: Age, dtype: int64
```

In [18]:

```
sns.boxplot(emp1['Age'])
```

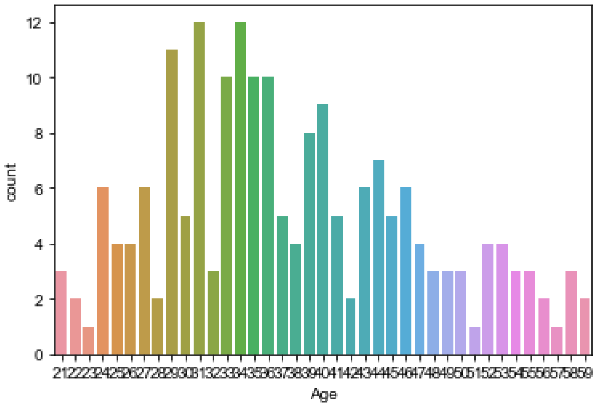
Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124acdc40>
```



In [19]:

```
sns.countplot(emp1['Age'])
sns.set(rc={'figure.figsize':(8.7,12.27)})
```



In [20]:

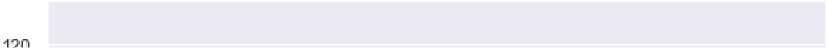
```
emp1['Gender'].value_counts()
```

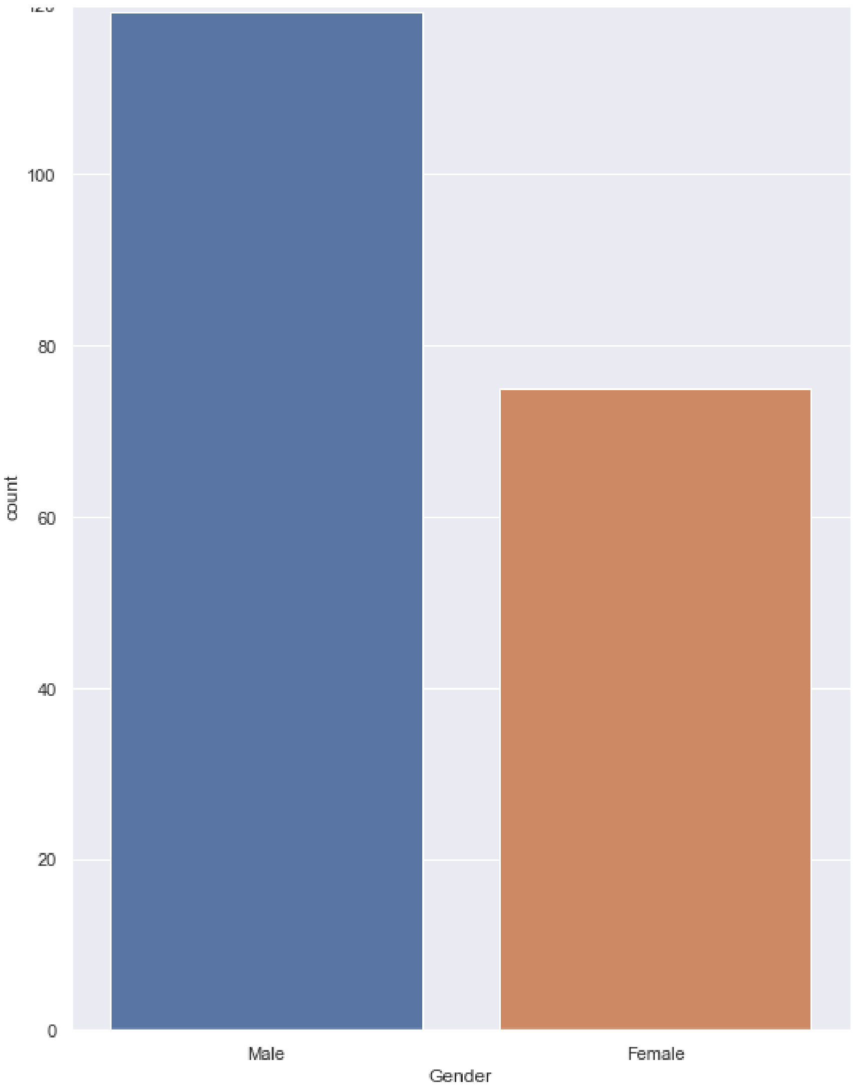
Out[20]:

```
Male      119
Female     75
Name: Gender, dtype: int64
```

In [21]:

```
sns.countplot(emp1['Gender'])
sns.set(rc={'figure.figsize':(5,5.27)})
```





In [22]:

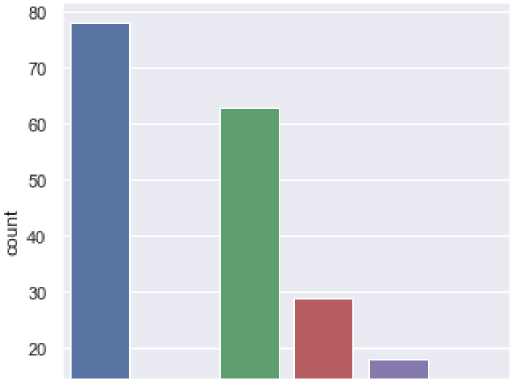
```
emp1['EducationBackground'].value_counts()
```

Out[22]:

```
Life Sciences        78
Medical              63
Marketing            29
Technical Degree     18
Other                 3
Human Resources       3
Name: EducationBackground, dtype: int64
```

In [23]:

```
sns.countplot(emp1['EducationBackground'])
sns.set(rc={'figure.figsize':(8,5.27)})
```



In [24]:

```
emp1['MaritalStatus'].value_counts()
```

Out[24]:

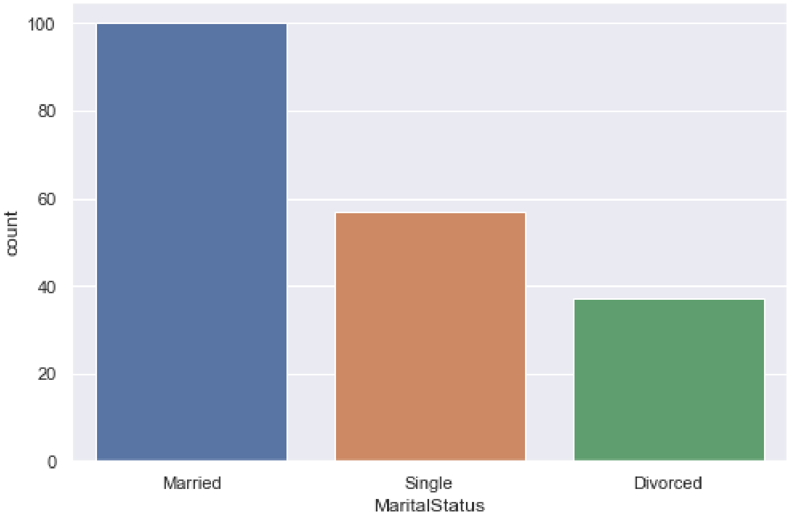
```
Married     100
Single       57
Divorced     37
Name: MaritalStatus, dtype: int64
```

In [25]:

```
sns.countplot(emp1['MaritalStatus'])
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124a2f430>
```



In [26]:

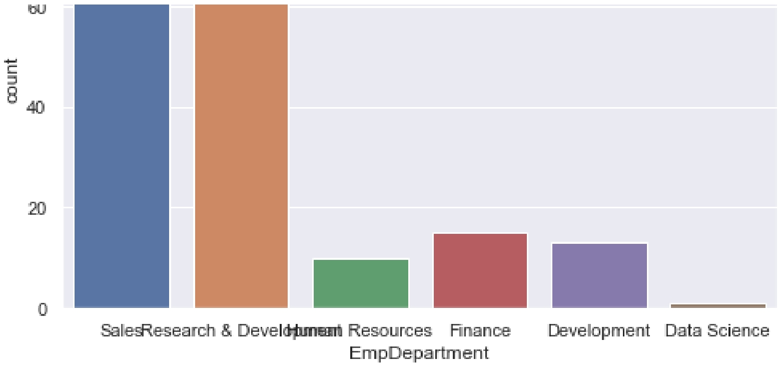
```
emp1['EmpDepartment'].value_counts()
```

Out[26]:

```
Sales                     87
Research & Development    68
Finance                   15
Development               13
Human Resources           10
Data Science               1
Name: EmpDepartment, dtype: int64
```

In [27]:

```
sns.countplot(emp1['EmpDepartment'])
sns.set(rc={'figure.figsize':(12,5.27)})
```



In [28]:

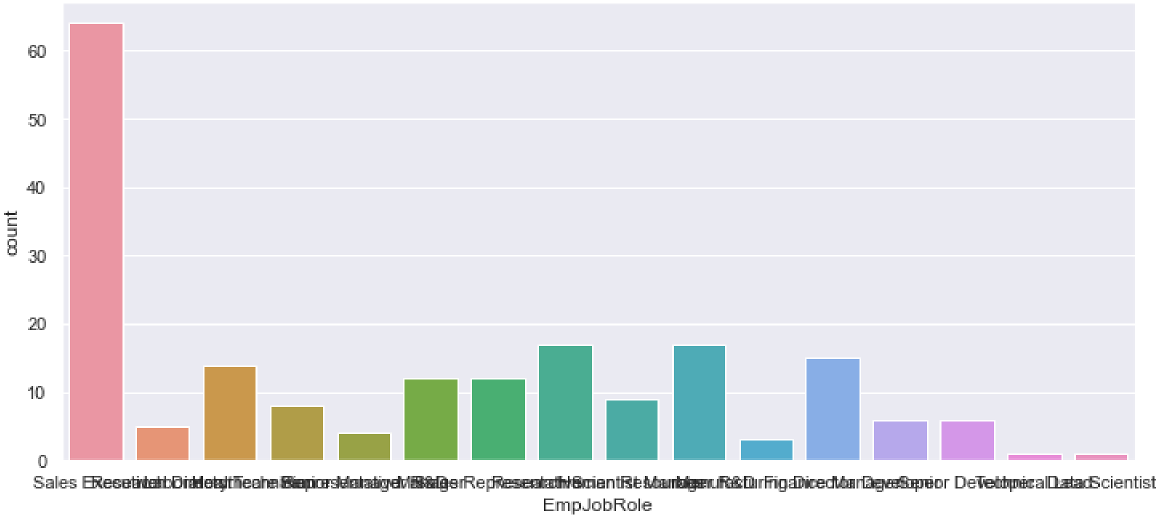
```
emp1['EmpJobRole'].value_counts()
```

Out[28]:

```
Sales Executive              64
Research Scientist           17
Manager R&D                  17
Finance Manager              15
Laboratory Technician        14
Manager                      12
Sales Representative         12
Human Resources               9
Healthcare Representative     8
Senior Developer              6
Developer                     6
Research Director             5
Senior Manager R&D            4
Manufacturing Director        3
Technical Lead                1
Data Scientist                1
Name: EmpJobRole, dtype: int64
```

In [29]:

```
sns.countplot(emp1['EmpJobRole'])
sns.set(rc={'figure.figsize':(35,12.27)})
```



In [30]:

```
import plotly.express as px

fig = px.pie(emp1, values='', names='', title='no of employee performing at 2')
fig.show()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
ValueError                                Traceback (most recent call last)
<ipython-input-30-d693235613d4> in <module>
      1 import plotly.express as px
      2 
----> 3 fig = px.pie(emp1, values='', names='', title='no of employee performing at 2')
      4 fig.show()

~\anaconda3\lib\site-packages\plotly\express\_chart_types.py in pie(data_frame, names, values, color, color_discrete_sequence, color_discrete_map, hover_name, hover_data, custom_data, labels, title, template, width, height, opacity, hole)
   1340     else:
   1341         layout_patch = {}
-> 1342     return make_figure(
   1343         args=locals(),
   1344         constructor=go.Pie,

~\anaconda3\lib\site-packages\plotly\express\_core.py in make_figure(args, constructor, trace_patch, layout_patch)
   1824     apply_default_cascade(args)
   1825 
-> 1826     args = build_dataframe(args, constructor)
   1827     if constructor in [go.Treemap, go.Sunburst] and args["path"] is not None:
   1828         args = process_dataframe_hierarchy(args)

~\anaconda3\lib\site-packages\plotly\express\_core.py in build_dataframe(args, constructor)
   1356     # now that things have been prepped, we do the systematic rewriting of `args`
   1357 
-> 1358     df_output, wide_id_vars = process_args_into_dataframe(
   1359         args, wide_mode, var_name, value_name
   1360     )

~\anaconda3\lib\site-packages\plotly\express\_core.py in process_args_into_dataframe(args, wide_mode, var_name, value_name)
   1162                     if argument == "index":
   1163                         err_msg += "\n To use the index, pass it in directly as `df.index`."
-> 1164                     raise ValueError(err_msg)
   1165             elif length and len(df_input[argument]) != length:
   1166                 raise ValueError(

ValueError: Value of 'names' is not the name of a column in 'data_frame'. Expected one of
['EmpNumber', 'Age', 'Gender', 'EducationBackground', 'MaritalStatus', 'EmpDepartment',
'EmpJobRole', 'BusinessTravelFrequency', 'DistanceFromHome', 'EmpEducationLevel',
'EmpEnvironmentSatisfaction', 'EmpHourlyRate', 'EmpJobInvolvement', 'EmpJobLevel',
'EmpJobSatisfaction', 'NumCompaniesWorked', 'OverTime', 'EmpLastSalaryHikePercent',
'EmpRelationshipSatisfaction', 'TotalWorkExperienceInYears', 'TrainingTimesLastYear',
'EmpWorkLifeBalance', 'ExperienceYearsAtThisCompany', 'ExperienceYearsInCurrentRole',
'YearsSinceLastPromotion', 'YearsWithCurrManager', 'Attrition', 'PerformanceRating'] but received:
```

In [31]:

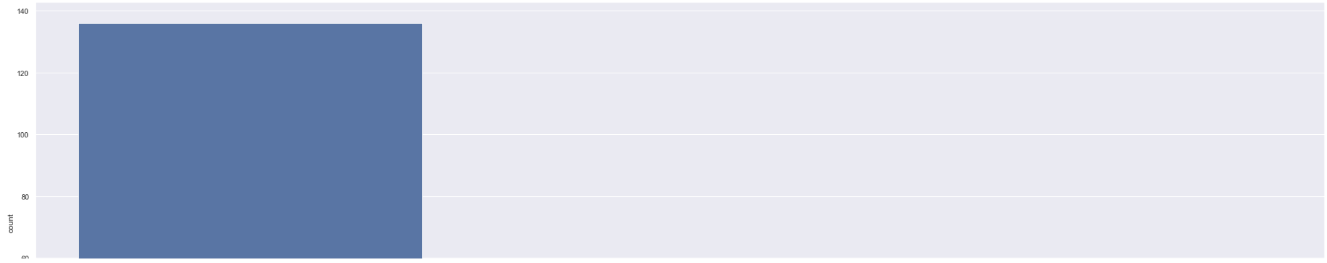
```
emp1['BusinessTravelFrequency'].value_counts()
```

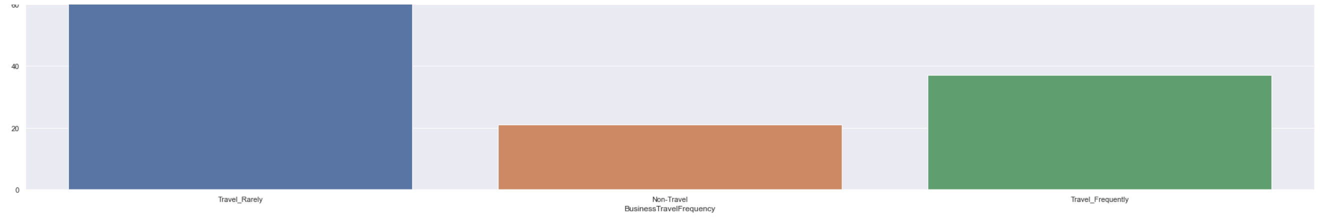
Out[31]:

```
Travel_Rarely        136
Travel_Frequently     37
Non-Travel            21
Name: BusinessTravelFrequency, dtype: int64
```

In [32]:

```
sns.countplot(emp1['BusinessTravelFrequency'])
sns.set(rc={'figure.figsize':(5,5.27)})
```





In [33]:

```
emp1['DistanceFromHome'].value_counts()
```

Out[33]:

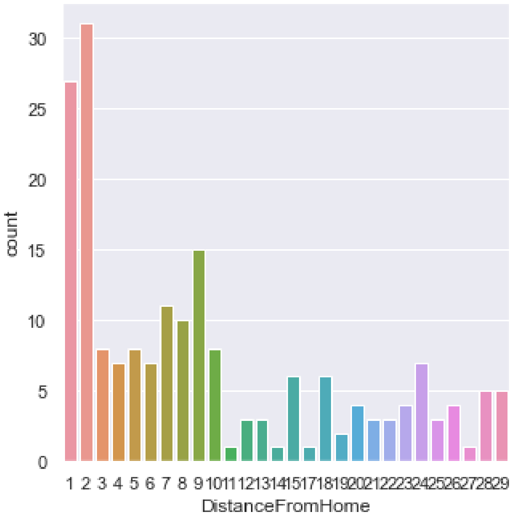
```
2     31
1     27
9     15
7     11
8     10
3      8
5      8
10     8
4      7
24     7
6      7
18     6
15     6
28     5
29     5
20     4
23     4
26     4
13     3
12     3
21     3
22     3
25     3
19     2
11     1
17     1
27     1
14     1
Name: DistanceFromHome, dtype: int64
```

In [34]:

```
sns.countplot(emp1['DistanceFromHome'])
```

Out[34]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231248c0940>
```

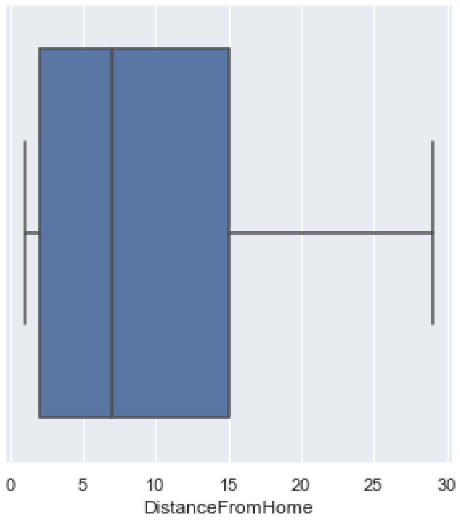


In [35]:

```
sns.boxplot(emp1['DistanceFromHome'])
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23123fe6670>
```



In [36]:

```
emp1['EmpEducationLevel'].value_counts()
```

Out[36]:

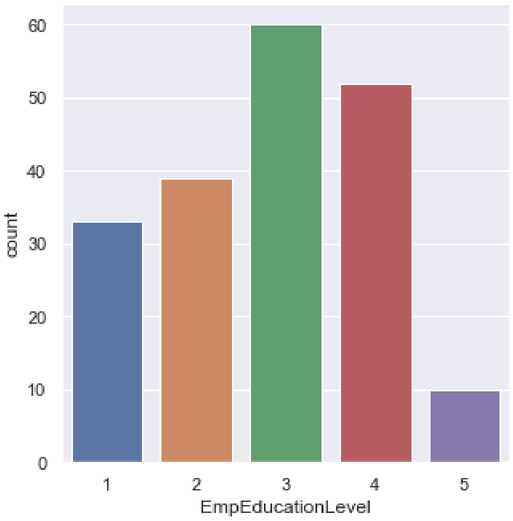
```
3    60
4    52
2    39
1    33
5    10
Name: EmpEducationLevel, dtype: int64
```

In [37]:

```
sns.countplot(emp1['EmpEducationLevel'])
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124a7aca0>
```



In [38]:

```
emp1['EmpEnvironmentSatisfaction'].value_counts()
```

Out[38]:

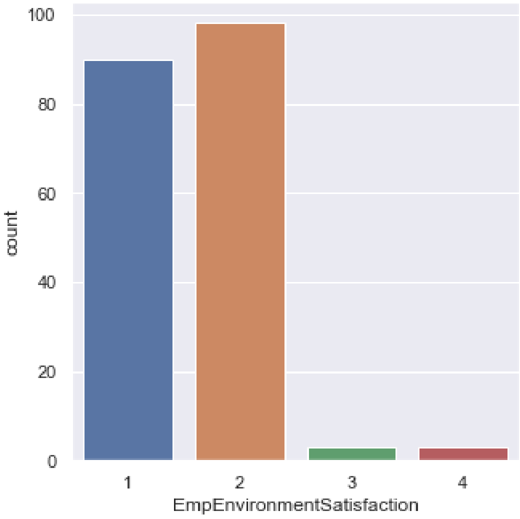
```
2    98
1    90
4     3
3     3
Name: EmpEnvironmentSatisfaction, dtype: int64
```

In [39]:

```
sns.countplot(emp1['EmpEnvironmentSatisfaction'])
```

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124caf280>
```



In [40]:

```
emp1['EmpJobInvolvement'].value_counts()
```

Out[40]:

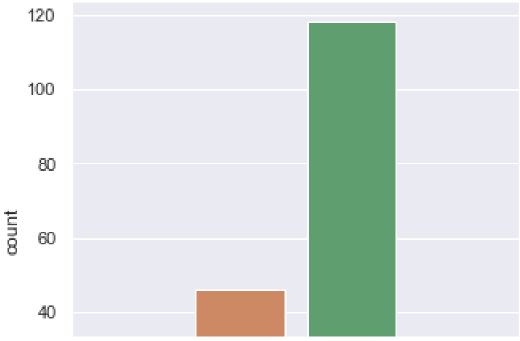
```
3    118
2     46
4     17
1     13
Name: EmpJobInvolvement, dtype: int64
```

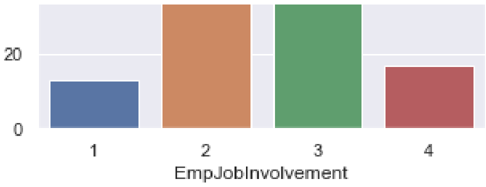
In [41]:

```
sns.countplot(emp1['EmpJobInvolvement'])
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124713280>
```





In [42]:

```
emp1['EmpJobLevel'].value_counts()
```

Out[42]:

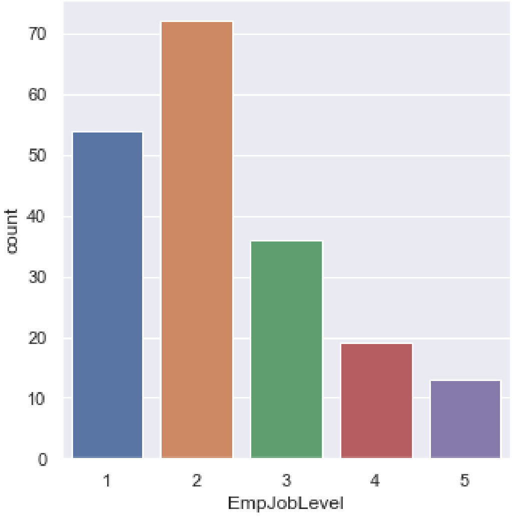
```
2    72
1    54
3    36
4    19
5    13
Name: EmpJobLevel, dtype: int64
```

In [43]:

```
sns.countplot(emp1['EmpJobLevel'])
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124734a60>
```



In [44]:

```
emp1['EmpJobSatisfaction'].value_counts()
```

Out[44]:

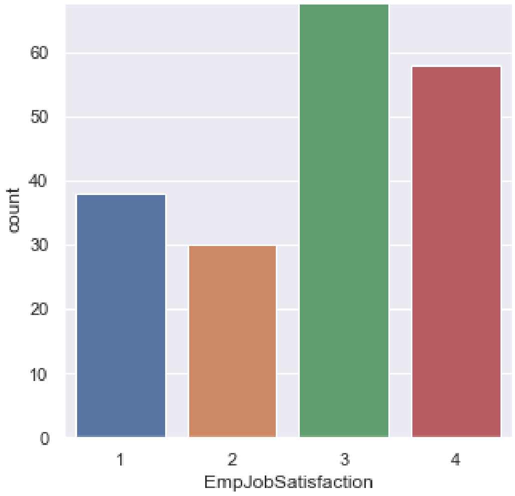
```
3    68
4    58
1    38
2    30
Name: EmpJobSatisfaction, dtype: int64
```

In [45]:

```
sns.countplot(emp1['EmpJobSatisfaction'])
```

Out[45]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231247861c0>
```



In [46]:

```
emp1['NumCompaniesWorked'].value_counts()
```

Out[46]:

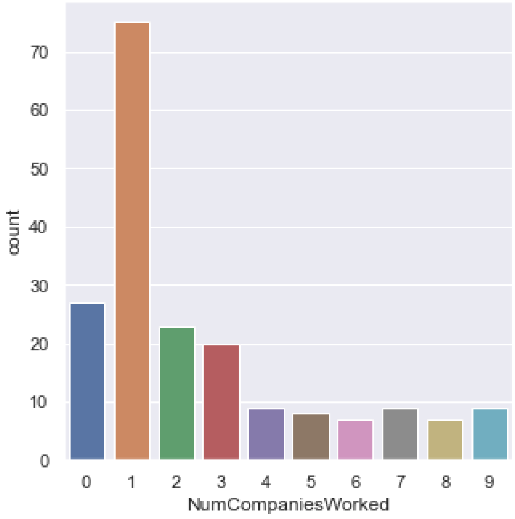
```
1    75
0    27
2    23
3    20
9     9
7     9
4     9
5     8
8     7
6     7
Name: NumCompaniesWorked, dtype: int64
```

In [47]:

```
sns.countplot(emp1['NumCompaniesWorked'])
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312491a1c0>
```

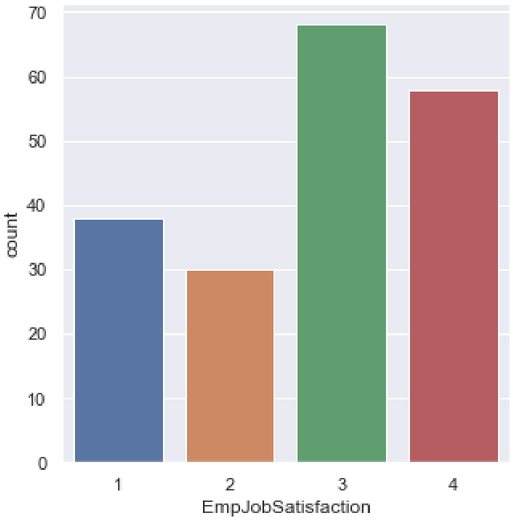


In [48]:

```
sns.countplot(emp1['EmpJobSatisfaction'])
```

Out[48]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124d95760>
```



In [49]:

```
emp1['OverTime'].value_counts()
```

Out[49]:

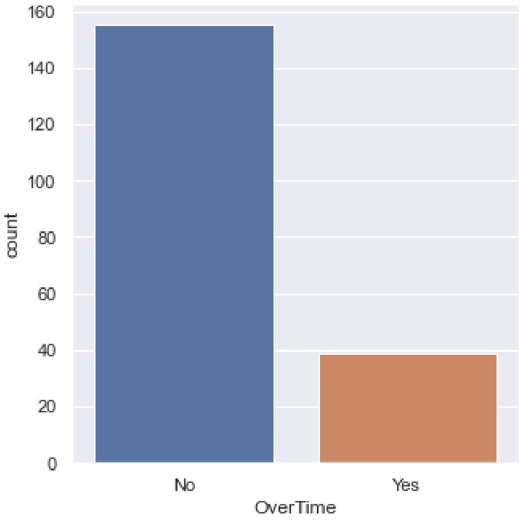
```
No     155
Yes     39
Name: OverTime, dtype: int64
```

In [50]:

```
sns.countplot(emp1['OverTime'])
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23124f0d4f0>
```



In [51]:

```
emp1['EmpLastSalaryHikePercent'].value_counts()
```

Out[51]:

```
12    30
14    28
11    28
13    27
16    12
15    11
```

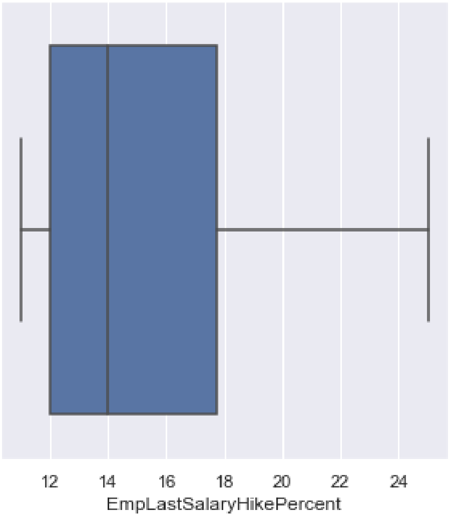
```
19     10
18     10
20      9
17      9
22      7
23      4
21      4
25      3
24      2
Name: EmpLastSalaryHikePercent, dtype: int64
```

In [52]:

```
sns.boxplot(emp1['EmpLastSalaryHikePercent'])
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23125f66f40>
```

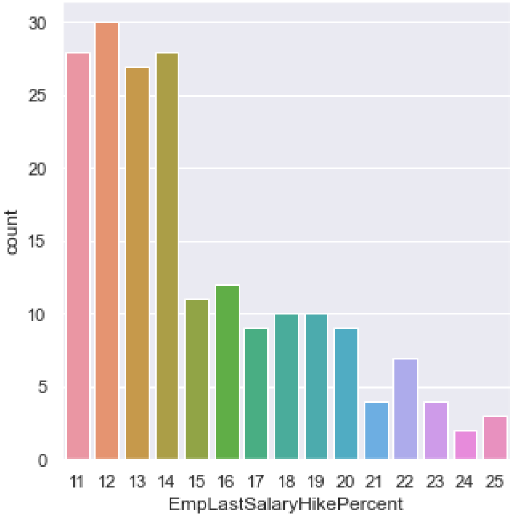


In [53]:

```
sns.countplot(emp1['EmpLastSalaryHikePercent'])
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23125f35160>
```



In [54]:

```
emp1['EmpRelationshipSatisfaction'].value_counts()
```

Out[54]:

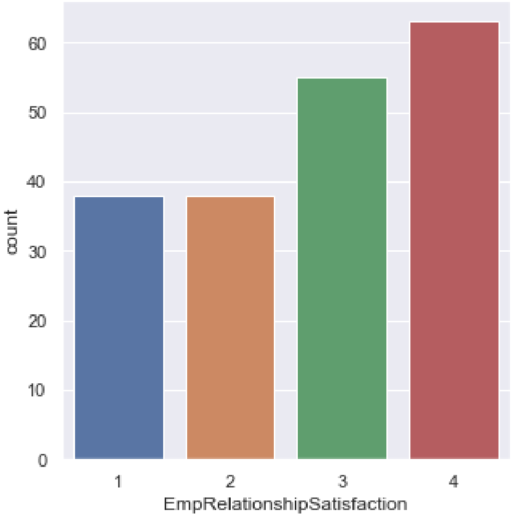
```
4    63
3    55
2    38
1    38
Name: EmpRelationshipSatisfaction, dtype: int64
```

In [55]:

```
sns.countplot(emp1['EmpRelationshipSatisfaction'])
```

Out[55]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23125fb8b80>
```



In [56]:

```
emp['TotalWorkExperienceInYears'].value_counts()
```

Out[56]:

```
10    159
6     105
8      85
9      77
5      71
1      65
7      61
4      51
12     37
15     34
3      34
13     33
11     33
16     32
21     28
14     26
2      26
20     25
17     24
18     21
19     20
22     18
23     17
24     13
28     13
26     13
25     12
0      10
29      8
32      8
31      7
```

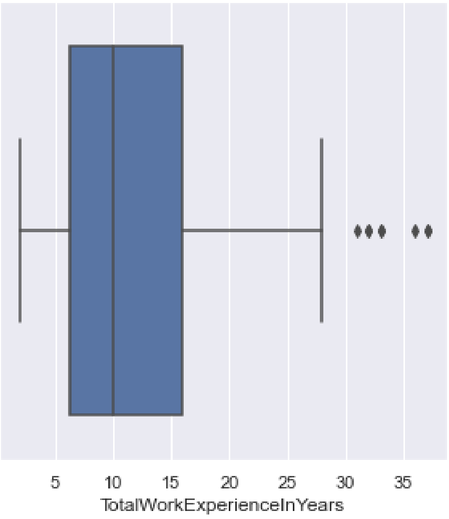
```
33      7
27      6
30      5
34      5
36      4
37      3
35      2
38      1
40      1
Name: TotalWorkExperienceInYears, dtype: int64
```

In [57]:

```
sns.boxplot(emp1['TotalWorkExperienceInYears'])
```

Out[57]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231260b1a00>
```



In [58]:

```
emp1['TrainingTimesLastYear'].value_counts()
```

Out[58]:

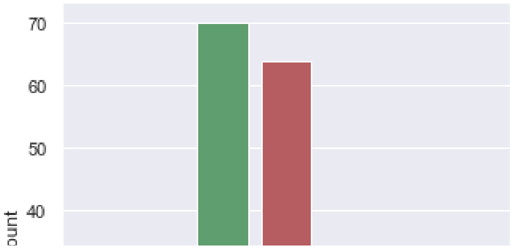
```
2    70
3    64
5    16
4    16
1    11
0     9
6     8
Name: TrainingTimesLastYear, dtype: int64
```

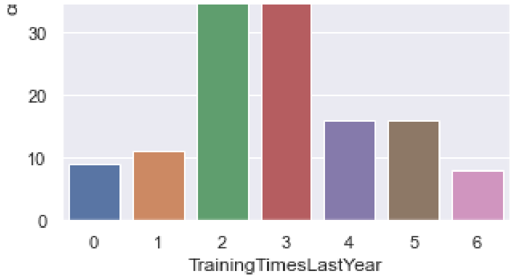
In [59]:

```
sns.countplot(emp1['TrainingTimesLastYear'])
```

Out[59]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23125eea640>
```





In [60]:

```
emp1['EmpWorkLifeBalance'].value_counts()
```

Out[60]:

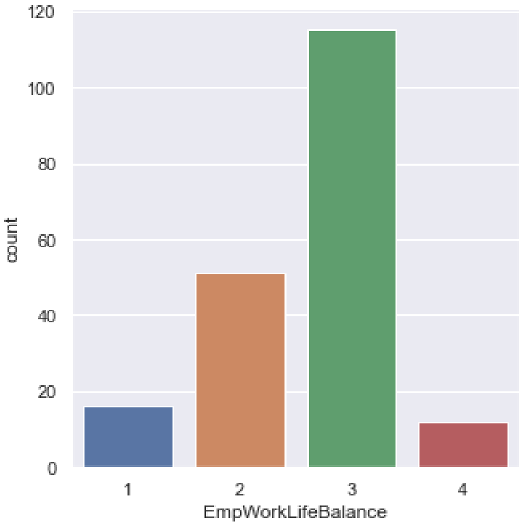
```
3    115
2     51
1     16
4     12
Name: EmpWorkLifeBalance, dtype: int64
```

In [61]:

```
sns.countplot(emp1['EmpWorkLifeBalance'])
```

Out[61]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23125fd6310>
```



In [62]:

```
emp['ExperienceYearsAtThisCompany'].value_counts()
```

Out[62]:

```
5     152
1     138
2     107
3     105
10    100
4      88
7      73
9      66
6      66
8      63
0      36
11     27
20     21
13     18
15     17
```

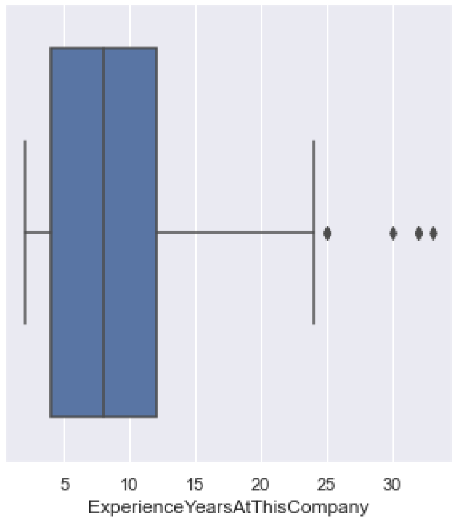
```
12     14
21     14
14     14
22     12
18     11
19     11
16     10
17      7
33      5
25      4
24      3
32      3
31      2
36      2
27      2
29      2
26      2
30      1
23      1
34      1
37      1
40      1
Name: ExperienceYearsAtThisCompany, dtype: int64
```

In [63]:

```
sns.boxplot(emp1['ExperienceYearsAtThisCompany'])
```

Out[63]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23126183490>
```



In [64]:

```
emp1['ExperienceYearsInCurrentRole'].value_counts()
```

Out[64]:

```
2      55
7      33
8      21
9      19
4      19
3      17
10      8
16      5
5       5
14      4
6       3
17      1
15      1
13      1
11      1
```

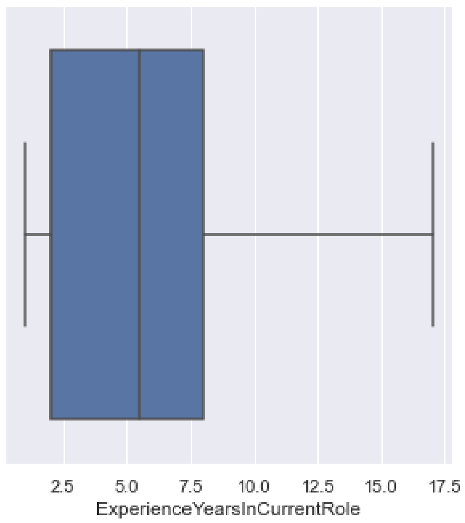
```
1      1
Name: ExperienceYearsInCurrentRole, dtype: int64
```

In [65]:

```
sns.boxplot(emp1['ExperienceYearsInCurrentRole'])
```

Out[65]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312616dfd0>
```



In [66]:

```
emp1['YearsSinceLastPromotion'].value_counts()
```

Out[66]:

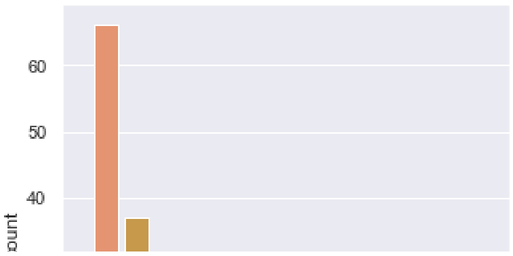
```
1     66
2     37
7     15
3     15
4     14
5     10
11     8
6      8
9      5
8      5
12     3
0      3
15     2
10     2
14     1
Name: YearsSinceLastPromotion, dtype: int64
```

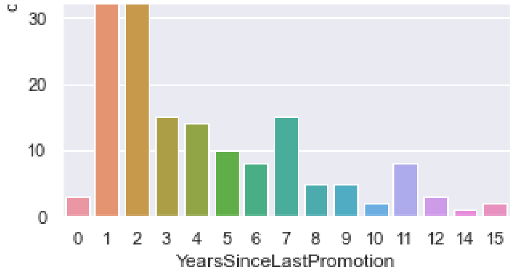
In [67]:

```
sns.countplot(emp1['YearsSinceLastPromotion'])
```

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231262331f0>
```





In [68]:

```
emp1['YearsWithCurrManager'].value_counts()
```

Out[68]:

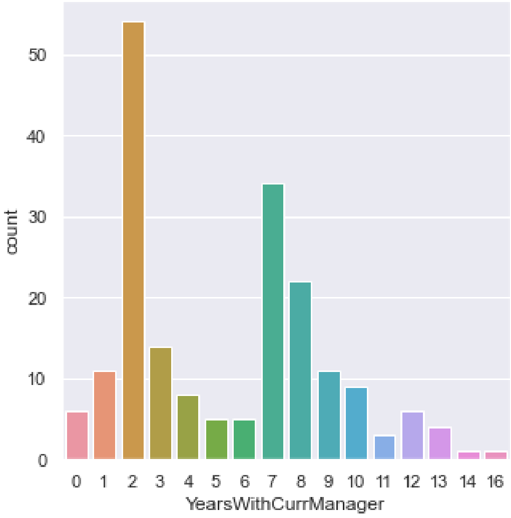
```
2     54
7     34
8     22
3     14
9     11
1     11
10     9
4      8
12     6
0      6
6      5
5      5
13     4
11     3
16     1
14     1
Name: YearsWithCurrManager, dtype: int64
```

In [69]:

```
sns.countplot(emp1['YearsWithCurrManager'])
```

Out[69]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231262bf550>
```

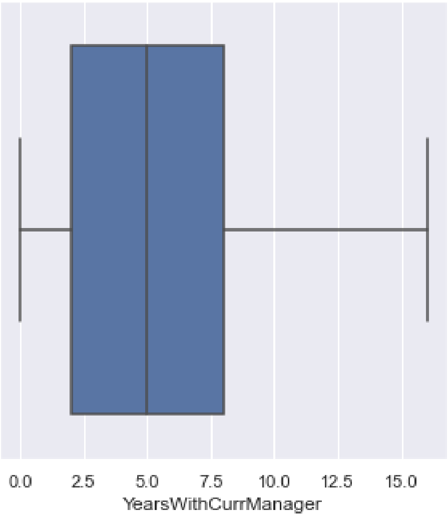


In [70]:

```
sns.boxplot(emp1['YearsWithCurrManager'])
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x23126338940>
```

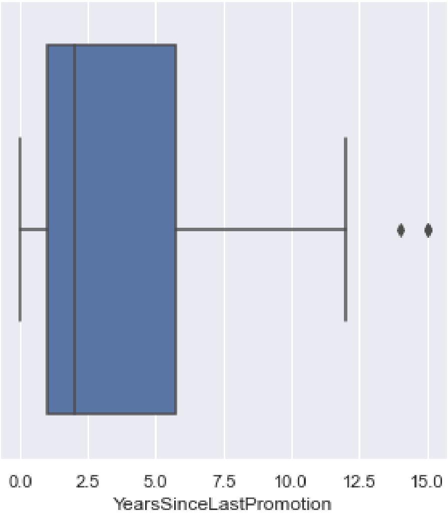


In [71]:

```
sns.boxplot(emp1['YearsSinceLastPromotion'])
```

Out[71]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312629edc0>
```



In [72]:

```
emp1['Attrition'].value_counts()
```

Out[72]:

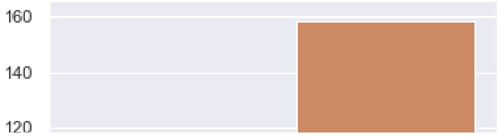
```
No     158
Yes     36
Name: Attrition, dtype: int64
```

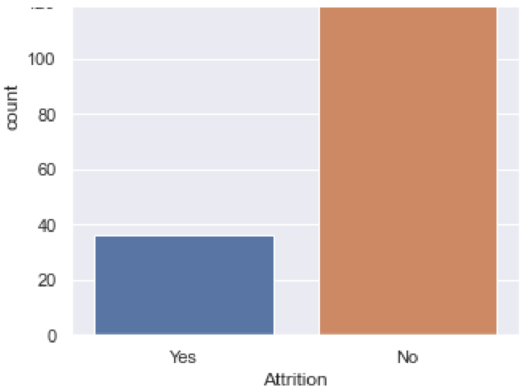
In [73]:

```
sns.countplot(emp1['Attrition'])
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231263f83a0>
```





In [ ]:

```
```

In [74]:

```
emp.pivot_table(values=['Age'], index=['PerformanceRating'],aggfunc = np.mean)
#young age group of people are performing well
```

Out[74]:

	Age
PerformanceRating	
2	37.804124
3	36.784897
4	36.500000

In [75]:

```
emp.groupby(emp1.PerformanceRating).mean()
```

Out[75]:

	Age	DistanceFromHome	EmpEducationLevel	EmpEnvironmentSatisfaction	EmpHourlyRate	EmpJobInvolveme
PerformanceRating						
2.0	37.804124	9.835052	2.829897	1.582474	68.216495	2.7164

In [76]:

```
emp1.loc[:, ['Age', 'PerformanceRating']]
```

Out[76]:

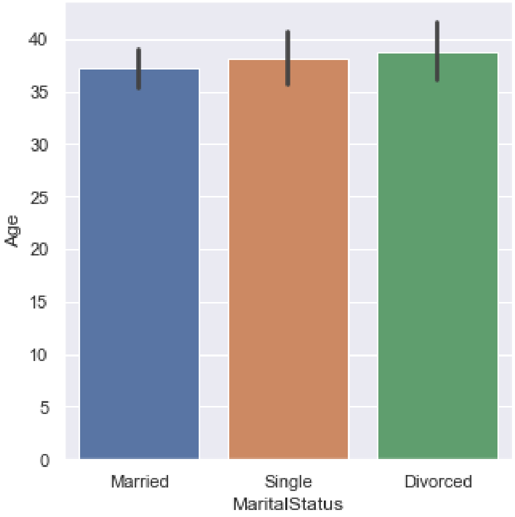
	Age	PerformanceRating
132	37	2
162	30	2
164	22	2
165	48	2
169	27	2
...	...	...
1152	41	2
1160	50	2
1162	24	2
1165	31	2
1199	24	2

1199	24	2
	Age	PerformanceRating

194 rows × 2 columns

In [77]:

```
sns.barplot(x='MaritalStatus',y='Age',data=emp1)
sns.set(rc={'figure.figsize':(5.7,4.27)})
```



In [78]:

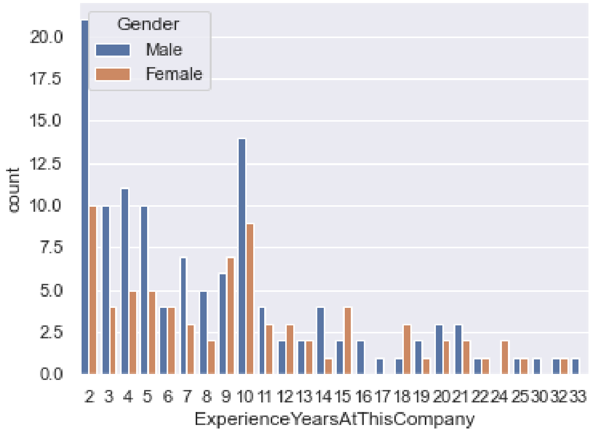
```
emp1.pivot_table(values=['ExperienceYearsAtThisCompany'],
                 index=['Gender'],aggfunc = np.mean)
```

Out[78]:

	ExperienceYearsAtThisCompany
Gender	
Female	9.893333
Male	8.596639

In [79]:

```
sns.countplot(emp1['ExperienceYearsAtThisCompany'],hue=emp1['Gender'])
#in rating 2 womens are performing more
#in rating 3 male are performing more
#in rating 4 both are almost equall
sns.set(rc={'figure.figsize':(9.7,4.27)})
```



In [80]:

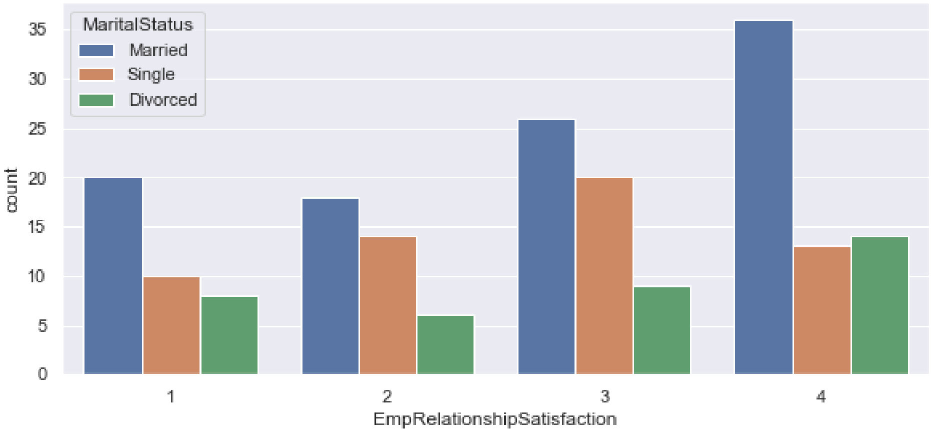
```
emp1.pivot_table(values=['PerformanceRating'],
                              index=['EducationBackground'],aggfunc = np.mean)
#overall marketing team is not performing well compare to others
sns.countplot
```

Out[80]:

```
<function seaborn.categorical.countplot(x=None, y=None, hue=None, data=None, order=None,
hue_order=None, orient=None, color=None, palette=None, saturation=0.75, dodge=True, ax=None, **kwa
rgs)>
```

In [81]:

```
sns.countplot(emp1['EmpRelationshipSatisfaction'],hue=emp1['MaritalStatus'])
sns.set(rc={'figure.figsize':(4,5.27)})
```



In [82]:

```
emp1.pivot_table(values=['EmpRelationshipSatisfaction'],
                              index=['MaritalStatus'],aggfunc = np.mean)
#as we can see their is slight performnce difference between married and single status
```

Out[82]:

	EmpRelationshipSatisfaction
MaritalStatus	
Divorced	2.783784
Married	2.780000
Single	2.631579

In [ ]:

In [83]:

```
a =emp1.pivot_table(values=['ExperienceYearsInCurrentRole'],
                              index=['EmpJobRole'],aggfunc = np.mean)
a
```

Out[83]:

	ExperienceYearsInCurrentRole
EmpJobRole	
Data Scientist	7.000000
Developer	5.666667
Finance Manager	5.066667
Healthcare Representative	4.875000

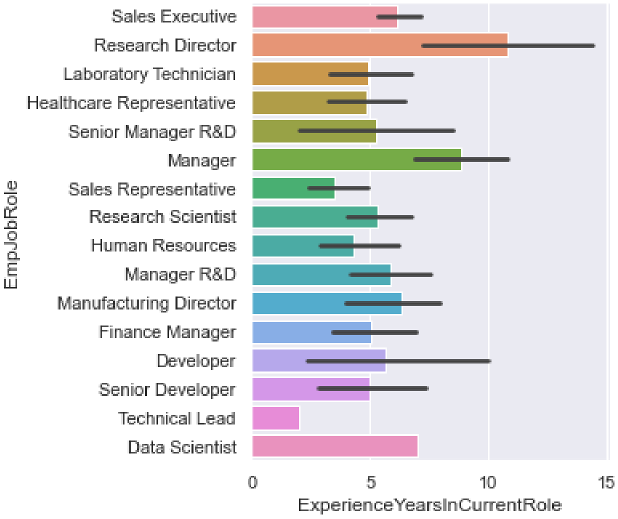
	ExperienceYearsInCurrentRole
EmpJobRole	
Healthcare Representative	4.875000
Human Resources	4.333333
Laboratory Technician	4.928571
Manager	8.833333
Manager R&D	5.882353
Manufacturing Director	6.333333
Research Director	10.800000
Research Scientist	5.352941
Sales Executive	6.156250
Sales Representative	3.500000
Senior Developer	5.000000
Senior Manager R&D	5.250000
Technical Lead	2.000000

In [84]:

```
sns.barplot(x='ExperienceYearsInCurrentRole',y='EmpJobRole',data= emp1)
```

Out[84]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312661a340>
```



In [85]:

```
emp1.pivot_table(values=['EmpJobSatisfaction'],
                         index=['BusinessTravelFrequency'],aggfunc = np.mean)
```

Out[85]:

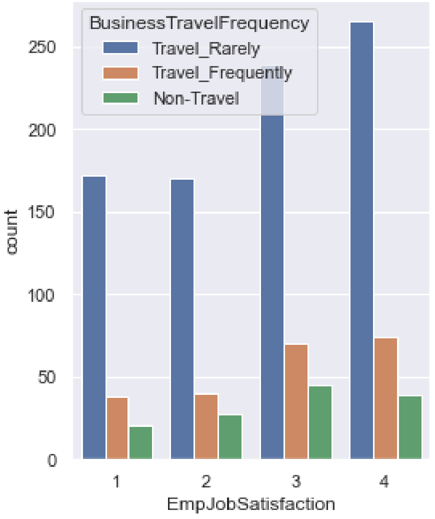
	EmpJobSatisfaction
BusinessTravelFrequency	
Non-Travel	2.666667
Travel_Frequently	3.027027
Travel_Rarely	2.691176

In [86]:

```
sns.countplot(x="EmpJobSatisfaction", hue="BusinessTravelFrequency", data= emp)
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231266bbb20>
```



In [87]:

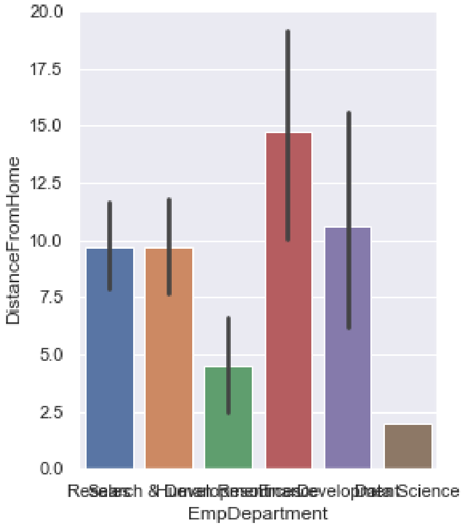
```
emp1.pivot_table(values=['DistanceFromHome'],
                          index=['EmpDepartment'],aggfunc = np.mean)
```

Out[87]:

	DistanceFromHome
EmpDepartment	
Data Science	2.000000
Development	10.615385
Finance	14.733333
Human Resources	4.500000
Research & Development	9.691176
Sales	9.689655

In [88]:

```
sns.barplot(x="EmpDepartment", y="DistanceFromHome", data= emp1)
sns.set(rc={'figure.figsize':(12,5.27)})
```



In [89]:

```
emp1.pivot_table(values=['EmpJobInvolvement'],
                          index=['EmpJobRole'],aggfunc = np.mean)
```

Out[89]:

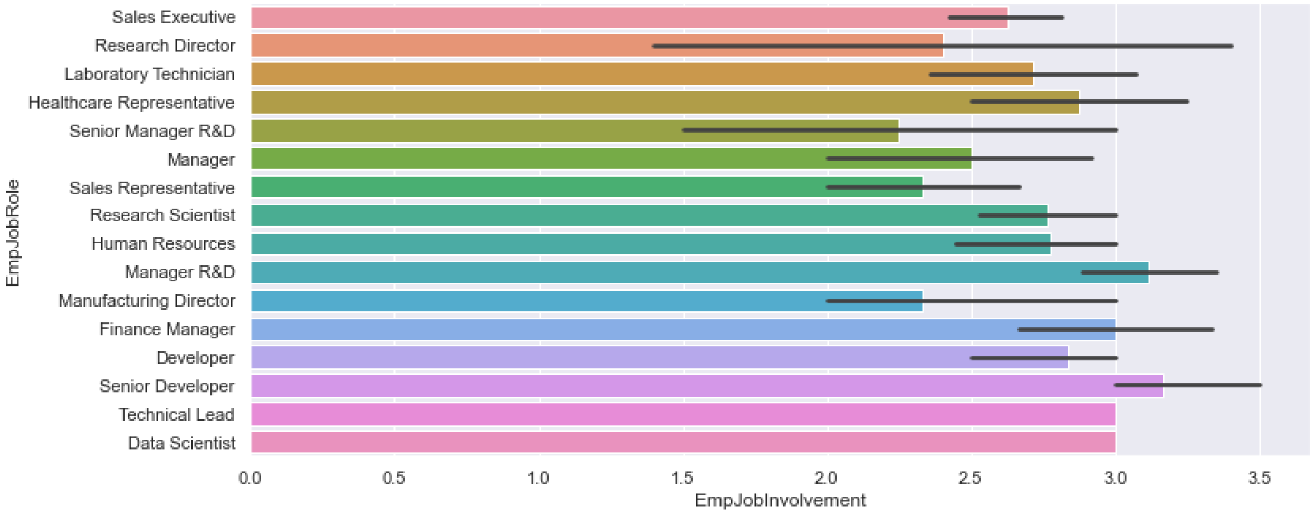
	EmpJobInvolvement
EmpJobRole	
Data Scientist	3.000000
Developer	2.833333
Finance Manager	3.000000
Healthcare Representative	2.875000
Human Resources	2.777778
Laboratory Technician	2.714286
Manager	2.500000
Manager R&D	3.117647
Manufacturing Director	2.333333
Research Director	2.400000
Research Scientist	2.764706
Sales Executive	2.625000
Sales Representative	2.333333
Senior Developer	3.166667
Senior Manager R&D	2.250000
Technical Lead	3.000000

In [90]:

```
sns.barplot(x="EmpJobInvolvement", y="EmpJobRole", data= emp1)
```

Out[90]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x231264fd730>
```



In [91]:

```
emp1.pivot_table(values=['YearsWithCurrManager'],
                              index=['EmpJobRole'],aggfunc = np.mean)
```

Out[91]:

	YearsWithCurrManager
EmpJobRole	
Data Scientist	9.000000
Developer	5.666667

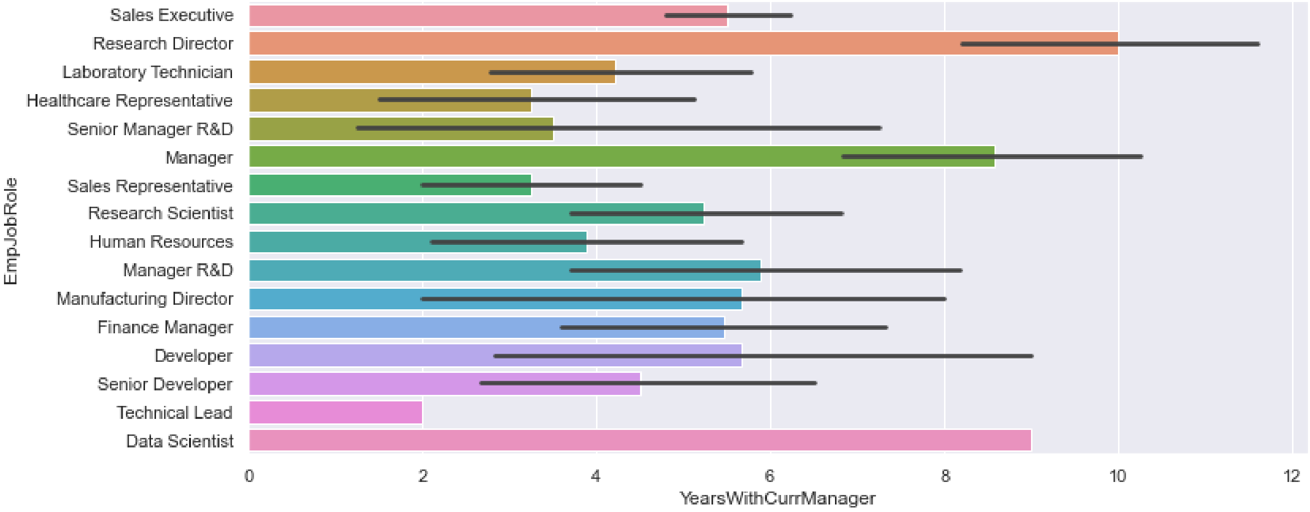
	YearsWithCurrManager
EmpJobRole	
Developer	5.666667
Finance Manager	5.466667
Healthcare Representative	3.250000
Human Resources	3.888889
Laboratory Technician	4.214286
Manager	8.583333
Manager R&D	5.882353
Manufacturing Director	5.666667
Research Director	10.000000
Research Scientist	5.235294
Sales Executive	5.500000
Sales Representative	3.250000
Senior Developer	4.500000
Senior Manager R&D	3.500000
Technical Lead	2.000000

In [145]:

```
sns.barplot(x="YearsWithCurrManager", y="EmpJobRole", data= emp1)
```

Out[145]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312ac18970>
```



In [141]:

```
emp1.pivot_table(values=['YearsSinceLastPromotion'],
                              index=['EmpDepartment'],aggfunc = np.mean)
```

Out[141]:

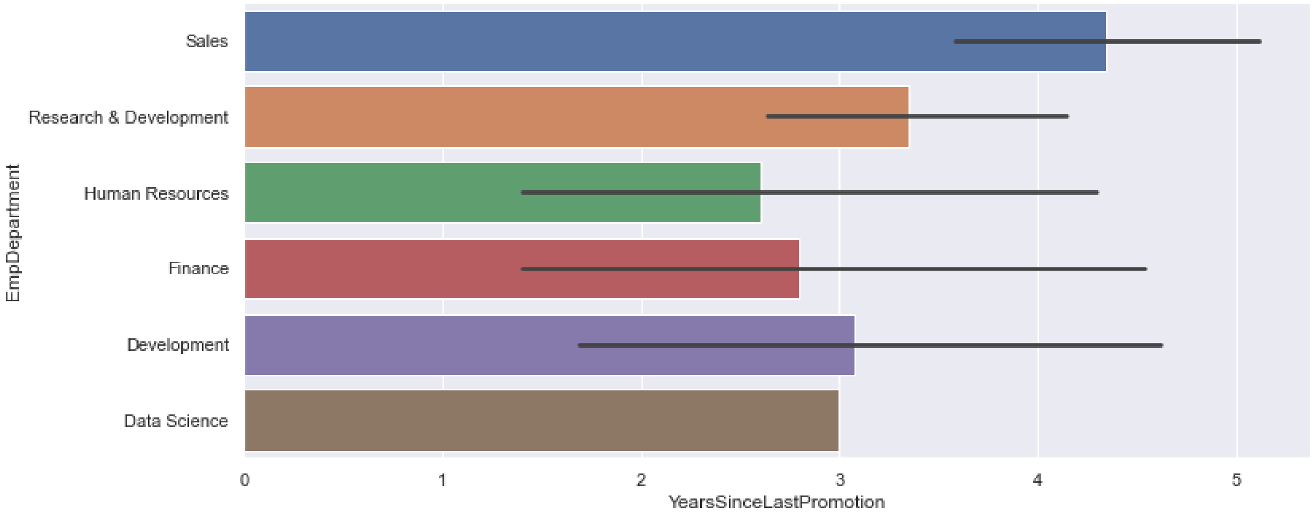
	YearsSinceLastPromotion
EmpDepartment	
Data Science	3.000000
Development	3.076923
Finance	2.800000
Human Resources	2.600000
Research & Development	3.352941
Sales	4.344828

In [143]:

```
sns.barplot(x='YearsSinceLastPromotion', y='EmpDepartment', data= emp1)
```

Out[143]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2312abcc160>
```



In [144]:

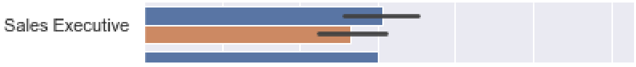
```
emp1.pivot_table(values=['TrainingTimesLastYear'],
                        index=['EmpJobRole'],aggfunc = np.mean)
```

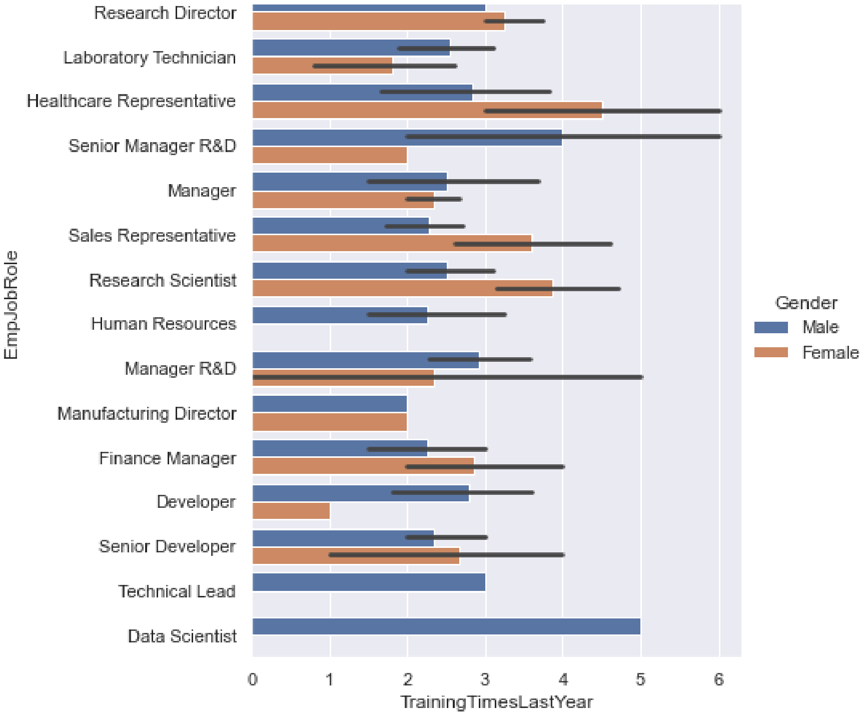
Out[144]:

	TrainingTimesLastYear
EmpJobRole	
Data Scientist	5.000000
Developer	2.500000
Finance Manager	2.533333
Healthcare Representative	3.250000
Human Resources	2.000000
Laboratory Technician	2.285714
Manager	2.416667
Manager R&D	2.823529
Manufacturing Director	2.000000
Research Director	3.200000
Research Scientist	3.058824
Sales Executive	2.875000
Sales Representative	2.833333
Senior Developer	2.500000
Senior Manager R&D	3.500000
Technical Lead	3.000000

In [173]:

```
sns.catplot(data = emp1, x="TrainingTimesLastYear", y="EmpJobRole", hue='Gender', kind='bar',
            height =7,)
plt.show()
```





Multi var analysis

In [180]:

```
d=emp1.pivot_table(values=['ExperienceYearsAtThisCompany','YearsSinceLastPromotion','ExperienceYear
sInCurrentRole','YearsWithCurrManager'],
                        index=['EmpDepartment','EmpJobRole','MaritalStatus'],aggfunc = np.mean)
d
#human resouces not working perfecctly adn even sales dept is slow
#Research & Development dep Senior Manager R&D even working low
```

Out[180]:

			ExperienceYearsAtThisCompany	ExperienceYearsInCurrentRole	YearsSinceLastPromotion
EmpDepartment	**EmpJobRole**	**MaritalStatus**			
Data Science	**Data Scientist**	**Divorced**	21.000000	7.000000	3.000000
Development	**Developer**	**Divorced**	13.333333	5.333333	2.666667
		Married	3.000000	2.000000	1.500000
		Single	15.000000	14.000000	8.000000
	Senior Developer	**Divorced**	11.000000	6.333333	3.333333
		Married	5.000000	3.666667	3.000000
	Technical Lead	**Divorced**	2.000000	2.000000	2.000000
Finance	**Finance Manager**	**Divorced**	10.666667	4.333333	1.000000
		Married	10.222222	6.111111	3.777778
		Single	6.333333	2.666667	1.666667
Human Resources	**Human Resources**	**Divorced**	11.000000	10.000000	3.000000
		Married	5.500000	3.000000	1.833333
		Single	7.000000	5.500000	1.000000
	Manager	**Married**	32.000000	5.000000	10.000000
Research & Development	**Healthcare Representative**	**Divorced**	11.666667	6.333333	8.333333
		Married	9.000000	7.000000	8.000000
		Single	4.250000	3.250000	2.000000
	Laboratory Technician	**Divorced**	4.500000	3.250000	2.750000
		Married	8.400000	7.200000	4.600000
		Single	6.000000	4.000000	2.200000

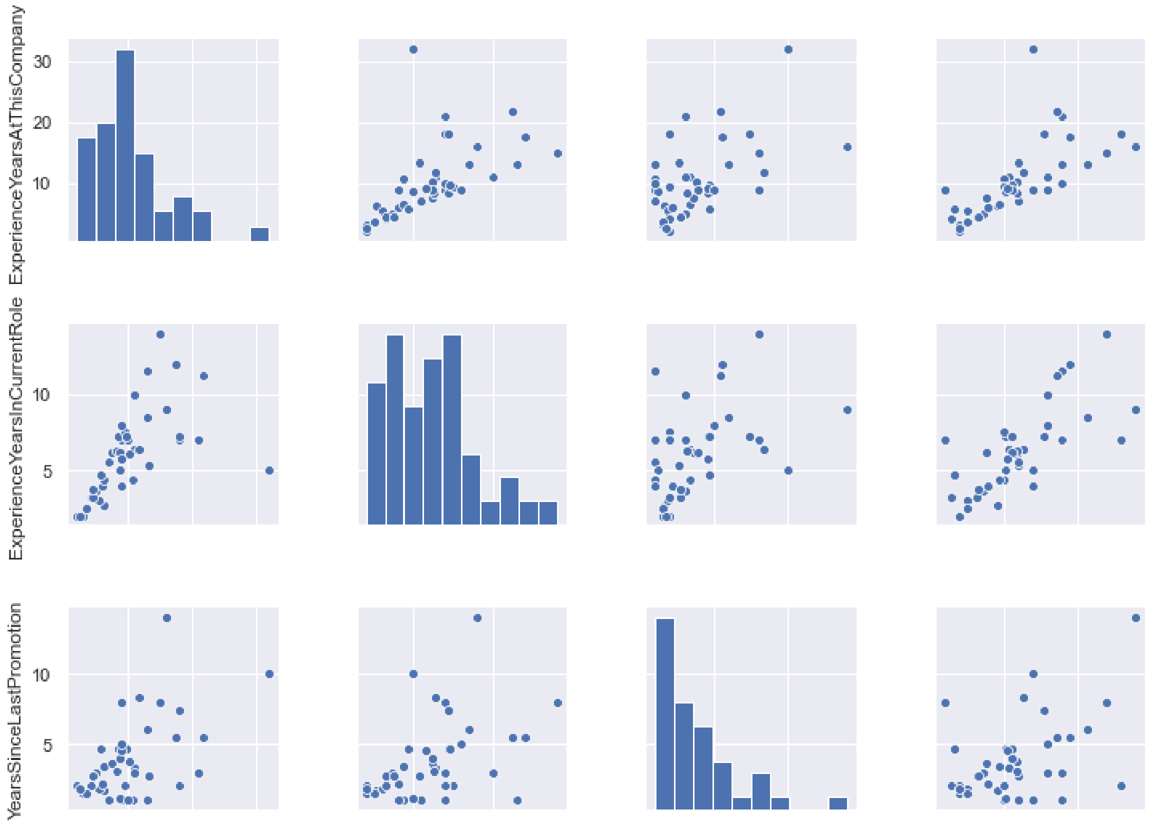
EmpDepartment	EmpJobRole	MaritalStatus	ExperienceYearsAtThisCompany	ExperienceYearsInCurrentRole	YearsSinceLastPromotion
		Single	6.000000	4.000000	2.200000
	Manager R&D	Divorced	13.000000	8.500000	6.000000
		Married	6.375000	4.375000	3.375000
		Single	7.600000	6.200000	3.600000
	Manufacturing Director	Married	9.500000	7.500000	2.000000
		Single	9.000000	4.000000	1.000000
	Research Director	Divorced	17.500000	12.000000	5.500000
		Married	13.000000	11.500000	1.000000
		Single	18.000000	7.000000	2.000000
	Research Scientist	Divorced	3.500000	2.500000	1.500000
		Married	8.222222	6.222222	3.111111
		Single	8.666667	5.000000	1.166667
	Senior Manager R&D	Married	5.666667	4.666667	4.666667
		Single	10.000000	7.000000	1.000000
Sales	Manager	Divorced	16.000000	9.000000	14.000000
		Married	21.800000	11.200000	5.400000
		Single	18.000000	7.200000	7.400000
	Sales Executive	Divorced	9.750000	7.250000	4.625000
		Married	8.810811	6.135135	3.945946
		Single	9.052632	5.736842	4.526316
	Sales Representative	Divorced	9.000000	8.000000	5.000000
		Married	4.428571	3.714286	2.714286
		Single	2.500000	2.000000	1.750000

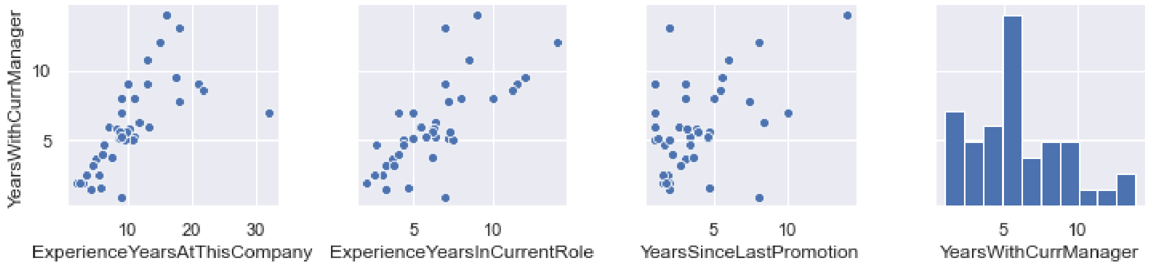
In [181]:

```
sns.pairplot(d)
```

Out[181]:

```
<seaborn.axisgrid.PairGrid at 0x23132781580>
```





In [121]:

```
g =emp1.pivot_table(values=['EmpWorkLifeBalance','EmpLastSalaryHikePercent','PerformanceRating'],
                           index=['EmpDepartment','EmpJobRole'],aggfunc = np.mean)
g
g.sort_values(by='PerformanceRating')
```

Out[121]:

		EmpLastSalaryHikePercent	EmpWorkLifeBalance	PerformanceRating
EmpDepartment	**EmpJobRole**			
Data Science	**Data Scientist**	19.000000	3.000000	2
Sales	**Manager**	15.909091	2.818182	2
Research & Development	**Senior Manager R&D**	17.250000	2.750000	2
	Research Scientist	14.588235	2.470588	2
	Research Director	15.400000	2.400000	2
	Manufacturing Director	15.666667	2.333333	2
	Manager R&D	16.117647	2.705882	2
Sales	**Sales Executive**	15.328125	2.640625	2
Research & Development	**Laboratory Technician**	14.285714	2.500000	2
Human Resources	**Manager**	11.000000	3.000000	2
	Human Resources	14.222222	2.666667	2
Finance	**Finance Manager**	14.533333	2.533333	2
Development	**Technical Lead**	12.000000	3.000000	2
	Senior Developer	12.500000	3.000000	2
	Developer	13.000000	2.500000	2
Research & Development	**Healthcare Representative**	13.000000	2.625000	2
Sales	**Sales Representative**	17.333333	2.750000	2

In [122]:

```
emp1['EmpLastSalaryHikePercent'].mode()
```

Out[122]:

```
0    12
dtype: int64
```

In [123]:

```
import plotly.express as px
fig = px.sunburst(emp, path=['PerformanceRating', 'EmpDepartment',
'EmpJobRole','BusinessTravelFrequency'])
fig.show()
```

In [124]:

```
emp1 = emp[emp['PerformanceRating']==2]
print(emp1.shape)
```

```
(194, 28)
```

In [125]:

```
emp1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 194 entries, 132 to 1199
Data columns (total 28 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   EmpNumber                     194 non-null    object
 1   Age                           194 non-null    int64 
 2   Gender                        194 non-null    object
 3   EducationBackground           194 non-null    object
 4   MaritalStatus                 194 non-null    object
 5   EmpDepartment                 194 non-null    object
 6   EmpJobRole                    194 non-null    object
 7   BusinessTravelFrequency       194 non-null    object
 8   DistanceFromHome              194 non-null    int64 
 9   EmpEducationLevel             194 non-null    int64 
 10  EmpEnvironmentSatisfaction    194 non-null    int64 
 11  EmpHourlyRate                 194 non-null    int64 
 12  EmpJobInvolvement             194 non-null    int64 
 13  EmpJobLevel                   194 non-null    int64 
 14  EmpJobSatisfaction            194 non-null    int64 
 15  NumCompaniesWorked            194 non-null    int64 
 16  OverTime                      194 non-null    object
 17  EmpLastSalaryHikePercent      194 non-null    int64 
 18  EmpRelationshipSatisfaction   194 non-null    int64 
 19  TotalWorkExperienceInYears    194 non-null    int64 
 20  TrainingTimesLastYear         194 non-null    int64 
 21  EmpWorkLifeBalance            194 non-null    int64 
 22  ExperienceYearsAtThisCompany  194 non-null    int64 
 23  ExperienceYearsInCurrentRole  194 non-null    int64 
 24  YearsSinceLastPromotion       194 non-null    int64 
 25  YearsWithCurrManager          194 non-null    int64 
 26  Attrition                     194 non-null    object
 27  PerformanceRating             194 non-null    int64 
dtypes: int64(19), object(9)
memory usage: 44.0+ KB
memory usage: 11.0+ KB
```

In [126]:

```
import plotly.express as px
fig = px.sunburst(emp1, path=['PerformanceRating', 'EmpDepartment', 'EmpJobRole','Gender','MaritalS
tatus'])
fig.show()
```

In [ ]:

In [ ]:

In [ ]:

In [127]:

```
h = emp1.pivot_table(values=['YearsWithCurrManager','YearsSinceLastPromotion'],
                        index=['EmpDepartment','EmpJobRole'],aggfunc = np.mean)
h
```

Out[127]:

		YearsSinceLastPromotion	YearsWithCurrManager
EmpDepartment	**EmpJobRole**		
Data Science	**Data Scientist**	3.000000	9.000000
Development	**Developer**	3.166667	5.666667
	Senior Developer	3.166667	4.500000
	Technical Lead	2.000000	2.000000
Finance	**Finance Manager**	2.800000	5.466667

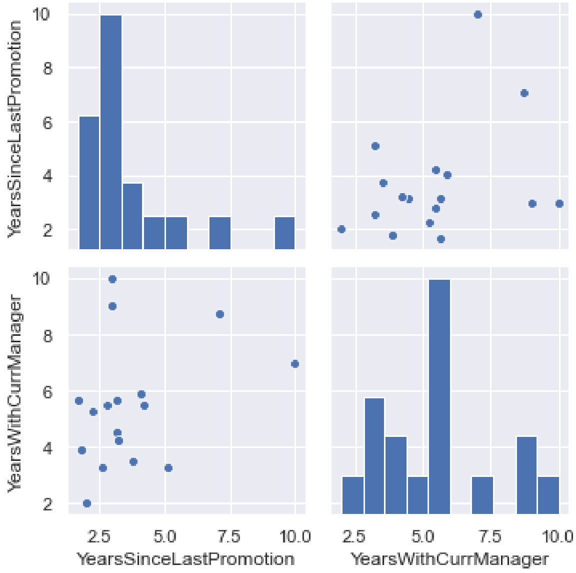
EmpDepartment	EmpJobRole	YearsSinceLastPromotion	YearsWithCurrManager
Human Resources	Human Resources	1.777778	3.888889
	Manager	10.000000	7.000000
Research & Development	Healthcare Representative	5.125000	3.250000
	Laboratory Technician	3.214286	4.214286
	Manager R&D	4.058824	5.882353
	Manufacturing Director	1.666667	5.666667
	Research Director	3.000000	10.000000
	Research Scientist	2.235294	5.235294
	Senior Manager R&D	3.750000	3.500000
Sales	Manager	7.090909	8.727273
	Sales Executive	4.203125	5.500000
	Sales Representative	2.583333	3.250000

In [128]:

```
sns.pairplot(h)
```

Out[128]:

```
<seaborn.axisgrid.PairGrid at 0x231261dfac0>
```



In [129]:

```
emp1.pivot_table(values=['YearsWithCurrManager','YearsSinceLastPromotion','PerformanceRating','EmpH
ourlyRate'],
                       index=['EmpDepartment','EmpJobRole','OverTime'],aggfunc = np.mean)
```

Out[129]:

			EmpHourlyRate	PerformanceRating	YearsSinceLastPromotion	YearsWithCurrManager
EmpDepartment	EmpJobRole	OverTime				
Data Science	Data Scientist	No	49.000000	2	3.000000	9.000000
Development	Developer	No	56.500000	2	2.500000	5.000000
		Yes	75.500000	2	4.500000	7.000000
	Senior Developer	No	76.666667	2	3.333333	4.000000
		Yes	54.333333	2	3.000000	5.000000
	Technical Lead	No	73.000000	2	2.000000	2.000000
Finance	Finance Manager	No	69.363636	2	2.454545	4.818182
		Yes	47.250000	2	3.750000	7.250000

EmpDepartment	EmpJobRole	OverTime	EmpHourlyRate	PerformanceRating	YearsSinceLastPromotion	YearsWithCurrManager
Human Resources	Human Resources	No	70.111111	2	1.777778	3.888889
	Manager	No	99.000000	2	10.000000	7.000000
Research & Development	Healthcare Representative	No	73.000000	2	4.714286	3.571429
		Yes	95.000000	2	8.000000	1.000000
	Laboratory Technician	No	68.833333	2	2.750000	3.750000
		Yes	59.000000	2	6.000000	7.000000
	Manager R&D	No	71.769231	2	4.461538	6.384615
		Yes	75.500000	2	2.750000	4.250000
	Manufacturing Director	No	83.500000	2	2.000000	7.500000
		Yes	79.000000	2	1.000000	2.000000
	Research Director	No	78.250000	2	3.500000	10.750000
		Yes	73.000000	2	1.000000	7.000000
	Research Scientist	No	66.076923	2	2.538462	5.769231
		Yes	52.000000	2	1.250000	3.500000
	Senior Manager R&D	No	66.750000	2	3.750000	3.500000
Sales	Manager	No	74.666667	2	6.888889	8.111111
		Yes	86.500000	2	8.000000	11.500000
	Sales Executive	No	64.352941	2	4.509804	5.686275
		Yes	71.692308	2	3.000000	4.769231
	Sales Representative	No	63.600000	2	2.600000	3.200000
		Yes	66.500000	2	2.500000	3.500000

In [130]:

```
emp1.pivot_table(values=['YearsWithCurrManager','PerformanceRating'],
                 index=['EmpDepartment','EmpJobRole','OverTime'],aggfunc = np.mean)
```

Out[130]:

			PerformanceRating	YearsWithCurrManager
EmpDepartment	EmpJobRole	OverTime		
Data Science	Data Scientist	No	2	9.000000
Development	Developer	No	2	5.000000
		Yes	2	7.000000
	Senior Developer	No	2	4.000000
		Yes	2	5.000000
	Technical Lead	No	2	2.000000
Finance	Finance Manager	No	2	4.818182
		Yes	2	7.250000
Human Resources	Human Resources	No	2	3.888889
	Manager	No	2	7.000000
Research & Development	Healthcare Representative	No	2	3.571429
		Yes	2	1.000000
	Laboratory Technician	No	2	3.750000
		Yes	2	7.000000
	Manager R&D	No	2	6.384615
		Yes	2	4.250000
	Manufacturing Director	No	2	7.500000
		Yes	2	2.000000
	Research Director	No	2	10.750000
		Yes	2	7.000000

EmpDepartment	EmpJobRole	OverTime	PerformanceRating	YearsWithCurrManager
	Research Scientist	No	2	5.769231
		Yes	2	3.500000
	Senior Manager R&D	No	2	3.500000
Sales	Manager	No	2	8.111111
		Yes	2	11.500000
	Sales Executive	No	2	5.686275
		Yes	2	4.769231
	Sales Representative	No	2	3.200000
		Yes	2	3.500000

In [131]:

```
f=emp1.corr()
f
```

Out[131]:

	Age	DistanceFromHome	EmpEducationLevel	EmpEnvironmentSatisfaction	EmpHourlyRate	EmpJ
Age	1.000000	-0.030277	0.204028	-0.021045	0.190302	
DistanceFromHome	-0.030277	1.000000	0.103505	0.095346	-0.006688	
EmpEducationLevel	0.204028	0.103505	1.000000	-0.064854	-0.040018	
EmpEnvironmentSatisfaction	-0.021045	0.095346	-0.064854	1.000000	0.045106	
EmpHourlyRate	0.190302	-0.006688	-0.040018	0.045106	1.000000	
EmpJobInvolvement	0.180085	-0.129792	0.047790	-0.023347	-0.013598	
EmpJobLevel	0.440019	-0.081609	0.092024	-0.002664	0.041173	
EmpJobSatisfaction	0.022414	-0.072503	-0.116233	-0.149427	-0.136038	
NumCompaniesWorked	0.215244	-0.099737	0.066062	0.001145	0.051560	
EmpLastSalaryHikePercent	0.091668	0.065728	0.078990	-0.005027	-0.045864	
EmpRelationshipSatisfaction	0.117896	-0.013526	-0.010773	-0.147741	-0.130197	
TotalWorkExperienceInYears	0.625342	-0.076558	0.149687	-0.070170	0.136028	
TrainingTimesLastYear	-0.015770	0.016301	-0.023799	0.028314	-0.024702	
EmpWorkLifeBalance	-0.084914	-0.016158	0.030529	-0.302425	0.037808	
ExperienceYearsAtThisCompany	0.328717	-0.099152	0.049416	0.002447	0.084253	
ExperienceYearsInCurrentRole	0.231371	-0.030627	0.003714	0.132776	0.086235	
YearsSinceLastPromotion	0.225096	-0.013934	0.113764	0.044216	0.009500	
YearsWithCurrManager	0.232285	-0.011111	0.110223	0.113435	0.062913	
PerformanceRating	NaN	NaN	NaN	NaN	NaN	

In [133]:

```
emp1.pivot_table(values=['EmpJobLevel','TotalWorkExperienceInYears','DistanceFromHome'],
                 index=['EmpDepartment','EmpJobRole','Gender',])
```

Out[133]:

			DistanceFromHome	EmpJobLevel	TotalWorkExperienceInYears
EmpDepartment	EmpJobRole	Gender			
Data Science	Data Scientist	Male	2.000000	5.000000	22.000000
Development	Developer	Female	1.000000	3.000000	15.000000
		Male	10.600000	2.400000	13.600000
	Senior Developer	Female	15.333333	2.333333	14.666667

EmpDepartment	EmpJobRole	Gender	DistanceFromHome	EmpJobLevel	TotalWorkExperienceInYears
		Male	11.666667	3.000000	13.000000
	Technical Lead	Male	3.000000	3.000000	9.000000
Finance	Finance Manager	Female	13.285714	2.571429	16.285714
		Male	16.000000	2.000000	13.000000
Human Resources	Human Resources	Female	9.000000	1.000000	2.000000
		Male	4.125000	1.375000	9.625000
	Manager	Male	3.000000	5.000000	32.000000
Research & Development	Healthcare Representative	Female	13.500000	2.000000	12.500000
		Male	3.666667	2.666667	13.833333
	Laboratory Technician	Female	7.800000	1.000000	7.000000
		Male	15.111111	1.444444	8.111111
	Manager R&D	Female	5.000000	2.333333	17.000000
		Male	7.285714	2.500000	16.428571
	Manufacturing Director	Female	25.000000	2.000000	10.000000
		Male	6.500000	2.000000	10.000000
	Research Director	Female	8.500000	3.500000	17.500000
		Male	15.000000	4.000000	16.000000
	Research Scientist	Female	13.285714	1.714286	11.000000
		Male	8.400000	1.300000	9.800000
	Senior Manager R&D	Female	2.000000	2.000000	4.000000
		Male	17.333333	2.333333	9.000000
Sales	Manager	Female	4.666667	4.500000	28.500000
		Male	3.800000	4.600000	28.000000
	Sales Executive	Female	10.413793	2.448276	12.068966
		Male	11.228571	2.400000	11.400000
	Sales Representative	Female	13.800000	1.200000	4.600000
		Male	4.571429	1.142857	6.571429

In [134]:

```
import plotly.express as px
fig = px.sunburst(emp1, path=['PerformanceRating','EmpWorkLifeBalance','EmpDepartment', 'EmpJobRole'])
fig.show()
import plotly.express as px
fig = px.sunburst(emp1, path=['PerformanceRating', 'EmpDepartment', 'EmpJobRole','Gender','MaritalStatus'])
fig.show()
import plotly.express as px
fig = px.sunburst(emp1, path=['PerformanceRating','EmpEnvironmentSatisfaction','EmpDepartment', 'EmpJobRole'])
fig.show()
```

In [135]:

In [135]:

```
import plotly.express as px
fig = px.sunburst(emp1, path=['PerformanceRating','EmpJobInvolvement','EmpDepartment', 'EmpJobRole'
,'Gender'])
fig.show()
```

In [136]:

```
emp1.pivot_table(values=['EmpEnvironmentSatisfaction','EmpJobInvolvement','EmpJobSatisfaction','Emp
RelationshipSatisfaction','EmpWorkLifeBalance','EmpJobLevel'],
                 index=['EmpDepartment','EmpJobRole',])
```

Out[136]:

		EmpEnvironmentSatisfaction	EmpJobInvolvement	EmpJobLevel	EmpJobSatisfaction	EmpRelationship
EmpDepartment	**EmpJobRole**					
Data Science	**Data Scientist**	1.000000	3.000000	5.000000	3.000000	
Development	**Developer**	1.166667	2.833333	2.500000	2.833333	
	Senior Developer	1.666667	3.166667	2.666667	2.833333	
	Technical Lead	2.000000	3.000000	3.000000	3.000000	
Finance	**Finance Manager**	1.600000	3.000000	2.266667	2.800000	
Human Resources	**Human Resources**	1.666667	2.777778	1.333333	2.666667	
	Manager	1.000000	3.000000	5.000000	2.000000	
Research & Development	**Healthcare Representative**	1.625000	2.875000	2.500000	2.625000	
	Laboratory Technician	1.714286	2.714286	1.285714	2.785714	
	Manager R&D	1.529412	3.117647	2.470588	3.058824	
	Manufacturing Director	2.000000	2.333333	2.000000	3.000000	
	Research Director	2.000000	2.400000	3.600000	2.200000	
	Research Scientist	1.470588	2.764706	1.470588	3.117647	

EmpDepartment	EmpJobRole	EmpEnvironmentSatisfaction	EmpJobInvolvement	EmpJobLevel	EmpJobSatisfaction	EmpRelationship
	Senior Manager R&D	1.250000	2.250000	2.250000	3.000000	
Sales	Manager	1.454545	2.454545	4.545455	2.363636	
	Sales Executive	1.625000	2.625000	2.421875	2.640625	
	Sales Representative	1.500000	2.333333	1.166667	2.833333	

In [137]:

```
import plotly.express as px
fig = px.sunburst(emp1, path=['EmpJobSatisfaction','EmpDepartment', 'EmpJobRole','Gender'])
fig.show()
```

In [138]:

```
fig = px.scatter(emp1,x = 'EmpWorkLifeBalance',y='EmpJobRole',color ='Gender',
                title='Work life Balance')
fig.show()
```

In [139]:

```
emp1.pivot_table(values=['YearsWithCurrManager','ExperienceYearsAtThisCompany','EmpLastSalaryHikePe
rcent'],
                index=['EmpJobInvolvement','EmpDepartment','EmpJobRole'])
```

Out[139]:

			EmpLastSalaryHikePercent	ExperienceYearsAtThisCompany	YearsWithCurrManage
EmpJobInvolvement	**EmpDepartment**	**EmpJobRole**			
1	**Research & Development**	**Laboratory Technician**	15.000000	4.000000	1.00000
		Research Director	17.000000	19.500000	10.50000
		Senior Manager R&D	19.000000	3.000000	2.00000
	Sales	**Manager**	15.500000	28.000000	10.00000
		Sales Executive	15.500000	3.333333	3.00000
		Sales Representative	13.000000	2.000000	2.00000
2	**Development**	**Developer**	11.000000	25.000000	9.00000
	Finance	**Finance Manager**	12.666667	4.666667	2.00000
	Human Resources	**Human Resources**	13.000000	11.000000	3.00000
	Research & Development	**Healthcare Representative**	12.500000	3.000000	2.00000
		Laboratory Technician	17.333333	5.666667	4.33333
		Manager R&D	14.000000	15.000000	13.00000
		Manufacturing Director	14.000000	9.500000	5.00000
		Research Scientist	15.000000	10.600000	6.00000
		Senior Manager R&D	12.000000	10.000000	9.00000
	Sales	**Manager**	17.000000	29.000000	9.50000
		Sales Executive	14.000000	9.388889	5.72222
		Sales Representative	16.666667	4.000000	3.00000
3	**Data Science**	**Data Scientist**	19.000000	21.000000	9.00000
	Development	**Developer**	13.400000	7.200000	5.00000
		Senior Developer	12.800000	7.600000	3.80000
		Technical Lead	12.000000	2.000000	2.00000
	Finance	**Finance Manager**	15.222222	11.000000	6.22222
	Human Resources	**Human Resources**	14.571429	5.142857	4.14285
		Manager	11.000000	32.000000	7.00000
	Research & Development	**Healthcare Representative**	13.400000	10.400000	4.00000

EmpJobInvolvement	EmpDepartment	EmpJobRole	EmpLastSalaryHikePercent	ExperienceYearsAtThisCompany	YearsWithCurrManage
		Laboratory Technician	13.444444	7.444444	4.77777
		Manager R&D	15.769231	7.615385	5.46153
		Manufacturing Director	19.000000	9.000000	7.00000
		Research Director	15.500000	16.000000	11.00000
		Research Scientist	13.545455	6.181818	4.63636
		Senior Manager R&D	19.000000	7.000000	1.50000
	Sales	Manager	15.714286	14.428571	8.14285
		Sales Executive	15.794118	9.235294	5.44117
		Sales Representative	19.000000	4.800000	3.80000
4	Development	Senior Developer	11.000000	10.000000	8.00000
	Finance	Finance Manager	14.333333	10.000000	6.66666
	Research & Development	Healthcare Representative	12.000000	3.000000	2.00000
		Laboratory Technician	12.000000	2.000000	2.00000
		Manager R&D	18.333333	9.000000	5.33333
		Research Director	12.000000	8.000000	7.00000
		Research Scientist	24.000000	12.000000	8.00000
	Sales	Sales Executive	16.500000	12.166667	7.66666

In [140]:

```
emp1.pivot_table(values=['EmpJobLevel','TrainingTimesLastYear',],
                 index=['EmpDepartment','EmpJobRole','Gender'])
```

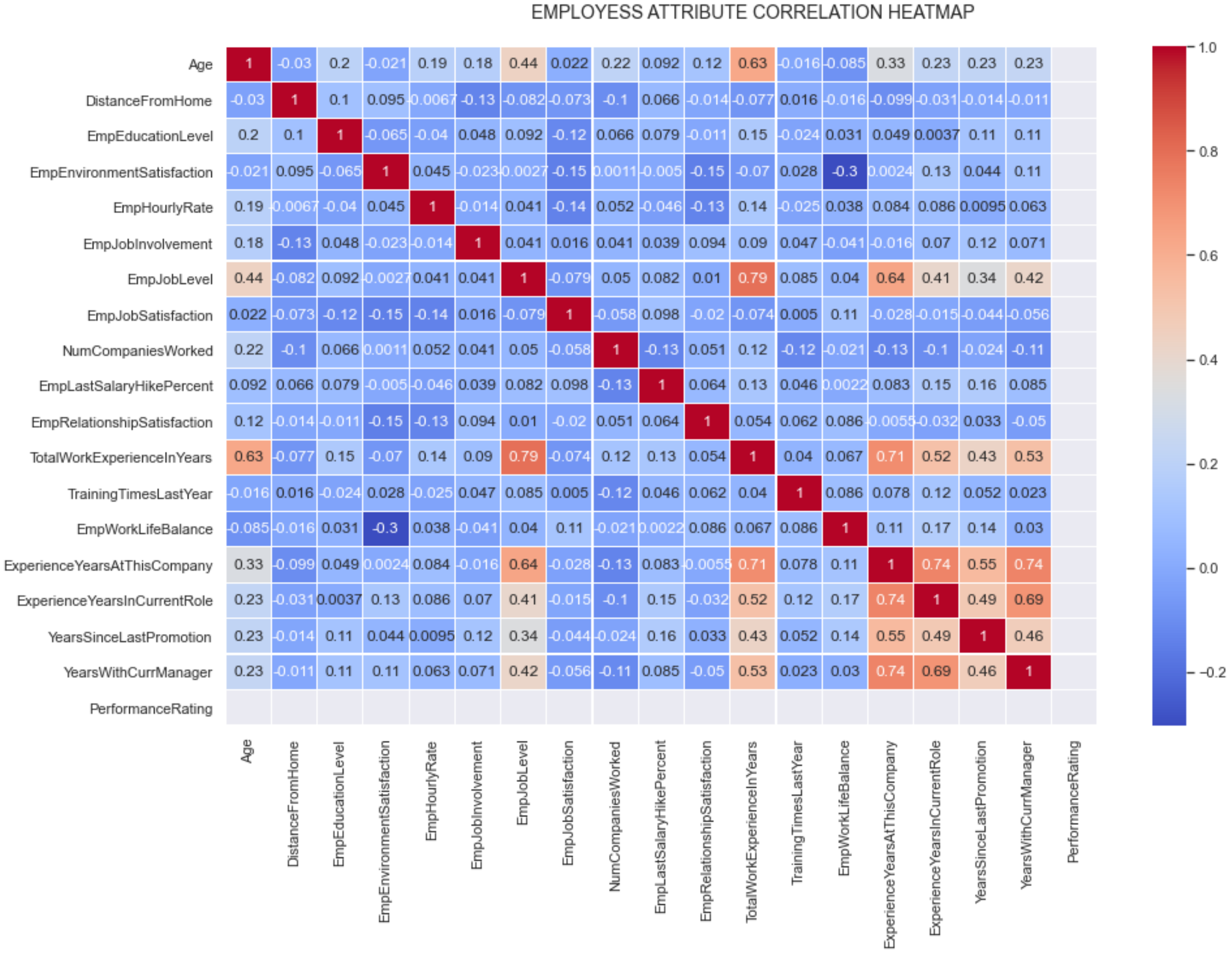
Out[140]:

			EmpJobLevel	TrainingTimesLastYear
EmpDepartment	EmpJobRole	Gender		
Data Science	Data Scientist	Male	5.000000	5.000000
Development	Developer	Female	3.000000	1.000000
		Male	2.400000	2.800000
	Senior Developer	Female	2.333333	2.666667
		Male	3.000000	2.333333
	Technical Lead	Male	3.000000	3.000000
Finance	Finance Manager	Female	2.571429	2.857143
		Male	2.000000	2.250000
Human Resources	Human Resources	Female	1.000000	0.000000
		Male	1.375000	2.250000
	Manager	Male	5.000000	2.000000
Research & Development	Healthcare Representative	Female	2.000000	4.500000
		Male	2.666667	2.833333
	Laboratory Technician	Female	1.000000	1.800000
		Male	1.444444	2.555556
	Manager R&D	Female	2.333333	2.333333
		Male	2.500000	2.928571

EmpDepartment	EmpJobRole	Gender	EmpJobLevel	TrainingTimesLastYear
	Manufacturing Director	Female	2.000000	2.000000
		Male	2.000000	2.000000
	Research Director	Female	3.500000	3.250000
		Male	4.000000	3.000000
	Research Scientist	Female	1.714286	3.857143
		Male	1.300000	2.500000
	Senior Manager R&D	Female	2.000000	2.000000
		Male	2.333333	4.000000
Sales	Manager	Female	4.500000	2.333333
		Male	4.600000	2.600000
	Sales Executive	Female	2.448276	2.655172
		Male	2.400000	3.057143
	Sales Representative	Female	1.200000	3.600000
		Male	1.142857	2.285714

In [182]:

```
f, ax = plt.subplots(figsize=(15, 9))
corr = emp1.corr()
hm = sns.heatmap(corr, annot=True, ax=ax, cmap="coolwarm",
                 linewidths=.05)
f.subplots_adjust(top=0.93)
t= f.suptitle('EMPLOYESS ATTRIBUTE CORRELATION HEATMAP', fontsize=14)
```



In [ ]:

In [ ]:

In [ ]: