We will do the Data visualization of the dataset to find more insights from the data

In [135]:

```
## Importing the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Import data

In [136]:

```
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

Exploratory Data Analysis

In [4]:

```
df.head()
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	E
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	

5 rows × 35 columns

In [5]:

```
df.iloc[0]
```

Out[5]:

```
Age                                      41
Attrition                               Yes
BusinessTravel                Travel_Rarely
DailyRate                              1102
Department                            Sales
DistanceFromHome                          1
Education                                 2
EducationField                Life Sciences
EmployeeCount                             1
EmployeeNumber                            1
EnvironmentSatisfaction                   2
Gender                               Female
HourlyRate                               94
JobInvolvement                            3
JobLevel                                  2
JobRole                     Sales Executive
JobSatisfaction                           4
MaritalStatus                        Single
MonthlyIncome                          5993
MonthlyRate                           19479
NumCompaniesWorked                        8
Over18                                    Y
OverTime                                Yes
PercentSalaryHike                        11
PerformanceRating                         3
RelationshipSatisfaction                  1
StandardHours                            80
StockOptionLevel                          0
TotalWorkingYears                         8
TrainingTimesLastYear                     0
WorkLifeBalance                           1
YearsAtCompany                            6
YearsInCurrentRole                        4
YearsSinceLastPromotion                   0
YearsWithCurrManager                      5
Name: 0, dtype: object
```

There are 35 columns and 1470 rows in this dataset. The columns refer to the attributes such as Age, Attrition, Department, Education, etc. For several attributes such as Education, each datapoint is a representative for description as follows:

- Education
  - 'Below College'
  - 'College'
  - 'Bachelor'
  - 'Master'
  - 'Doctor'
- EnvironmentSatisfaction
  - 'Low'
  - 'Medium'
  - 'High'
  - 'Very High'
- JobInvolvement
  - 'Low'
  - 'Medium'
  - 'High'
  - 'Very High'
- JobSatisfaction
  - 'Low'
  - 'Medium'
  - 'High'
  - 'Very High'
- PerformanceRating
  - 'Low'
  - 'Good'
  - 'Excellent'
  - 'Outstanding'
- RelationshipSatisfaction
  - 'Low'
  - 'Medium'
  - 'High'
  - 'Very High'
- WorkLifeBalance
  - 'Bad'
  - 'Good'
  - 'Better'
  - 'Best' ## The detail of the data types of those attributes can be viewed below.

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
Age                         1470 non-null int64
Attrition                   1470 non-null object
BusinessTravel              1470 non-null object
DailyRate                   1470 non-null int64
Department                  1470 non-null object
DistanceFromHome            1470 non-null int64
Education                   1470 non-null int64
EducationField              1470 non-null object
EmployeeCount               1470 non-null int64
EmployeeNumber              1470 non-null int64
EnvironmentSatisfaction     1470 non-null int64
Gender                      1470 non-null object
HourlyRate                  1470 non-null int64
JobInvolvement              1470 non-null int64
JobLevel                    1470 non-null int64
JobRole                     1470 non-null object
JobSatisfaction             1470 non-null int64
MaritalStatus               1470 non-null object
MonthlyIncome               1470 non-null int64
MonthlyRate                 1470 non-null int64
NumCompaniesWorked          1470 non-null int64
Over18                      1470 non-null object
OverTime                    1470 non-null object
PercentSalaryHike           1470 non-null int64
PerformanceRating           1470 non-null int64
RelationshipSatisfaction    1470 non-null int64
StandardHours               1470 non-null int64
StockOptionLevel            1470 non-null int64
TotalWorkingYears           1470 non-null int64
TrainingTimesLastYear       1470 non-null int64
WorkLifeBalance             1470 non-null int64
YearsAtCompany              1470 non-null int64
YearsInCurrentRole          1470 non-null int64
YearsSinceLastPromotion     1470 non-null int64
YearsWithCurrManager        1470 non-null int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [103]:

```
df.isna().sum()
```

Out[103]:

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

Manipulating data

In [7]:

```
def Educa_numttocat(number):
    if number is 1:
        return 'Below College'
    elif number is 2:
        return 'College'
    elif number is 3:
        return 'Bachelor'
    elif number is 4:
        return 'Master'
    elif number is 5:
        return 'Doctor'
```

In [8]:

```
def numttocat(number):
    if number is 1:
        return 'Low'
    elif number is 2:
        return 'Medium'
    elif number is 3:
        return 'High'
    elif number is 4:
        return 'Very High'
```

In [9]:

```
def PR_numttocat(number):
    if number is 1:
        return 'Low'
    elif number is 2:
        return 'Good'
    elif number is 3:
        return 'Excellent'
    elif number is 4:
        return 'Outstanding'
```

In [10]:

```
def wb_numttocat(number):
    if number is 1:
        return 'Bad'
    elif number is 2:
        return 'Good'
    elif number is 3:
        return 'Better'
    elif number is 4:
        return 'Best'
```

In [11]:

```
df["Education"] = df["Education"].map(Educa_numttocat)
df["EnvironmentSatisfaction"] = df["EnvironmentSatisfaction"].map(numttocat)
df["JobInvolvement"] = df["JobInvolvement"].map(numttocat)
df["JobSatisfaction"] = df["JobSatisfaction"].map(numttocat)
df["RelationshipSatisfaction"] = df["RelationshipSatisfaction"].map(numttocat)
df["PerformanceRating"] = df["PerformanceRating"].map(PR_numttocat)
df["WorkLifeBalance"] = df["WorkLifeBalance"].map(PR_numttocat)
```

In [12]:

```
df.head()
```

Out[12]:

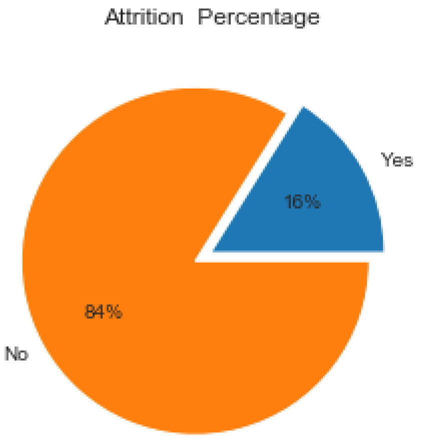
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	E
0	41	Yes	Travel_Rarely	1102	Sales	1	College	Life Sciences	
1	49	No	Travel_Frequently	279	Research & Development	8	Below College	Life Sciences	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	College	Other	
3	33	No	Travel_Frequently	1392	Research & Development	3	Master	Life Sciences	
4	27	No	Travel_Rarely	591	Research & Development	2	Below College	Medical	

5 rows × 35 columns

In [44]:
```
df.iloc[0]
```

Out[44]:
```
Age                                      41
Attrition                               Yes
BusinessTravel                Travel_Rarely
DailyRate                              1102
Department                            Sales
DistanceFromHome                          1
Education                           College
EducationField                Life Sciences
EmployeeCount                             1
EmployeeNumber                            1
EnvironmentSatisfaction              Medium
Gender                               Female
HourlyRate                               94
JobInvolvement                         High
JobLevel                                  2
JobRole                     Sales Executive
JobSatisfaction                   Very High
MaritalStatus                        Single
MonthlyIncome                          5993
MonthlyRate                           19479
NumCompaniesWorked                        8
Over18                                    Y
OverTime                                Yes
PercentSalaryHike                        11
PerformanceRating                 Excellent
RelationshipSatisfaction                Low
StandardHours                            80
StockOptionLevel                          0
TotalWorkingYears                         8
TrainingTimesLastYear                     0
WorkLifeBalance                         Low
YearsAtCompany                            6
YearsInCurrentRole                        4
YearsSinceLastPromotion                   0
YearsWithCurrManager                      5
Name: 0, dtype: object
```

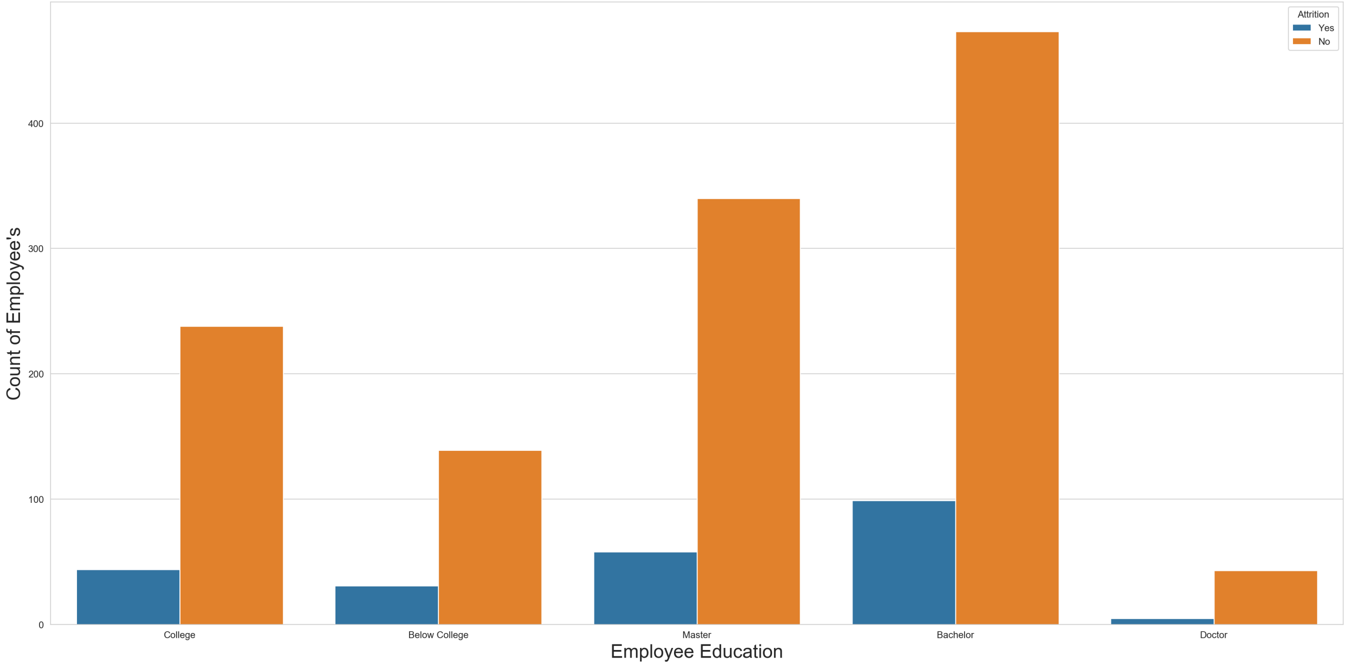
In [71]:
```
vals = [df.Attrition[df.Attrition=='Yes'].count() , df.Attrition[df.Attrition=='No'].cou
nt()]
label = ["Yes" , "No"]
plt.pie(vals , labels=label , autopct = '%1.0f%%' , explode=(0 , 0.1));
plt.title("Attrition  Percentage");
plt.savefig("attir.png")
```



```
In [15]: df.groupby(by='Education')['Attrition'].value_counts()
```

```
Out[15]: Education      Attrition
         Bachelor       No           473
                        Yes           99
         Below College  No           139
                        Yes           31
         College        No           238
                        Yes           44
         Doctor         No            43
                        Yes            5
         Master         No           340
                        Yes           58
         Name: Attrition, dtype: int64
```

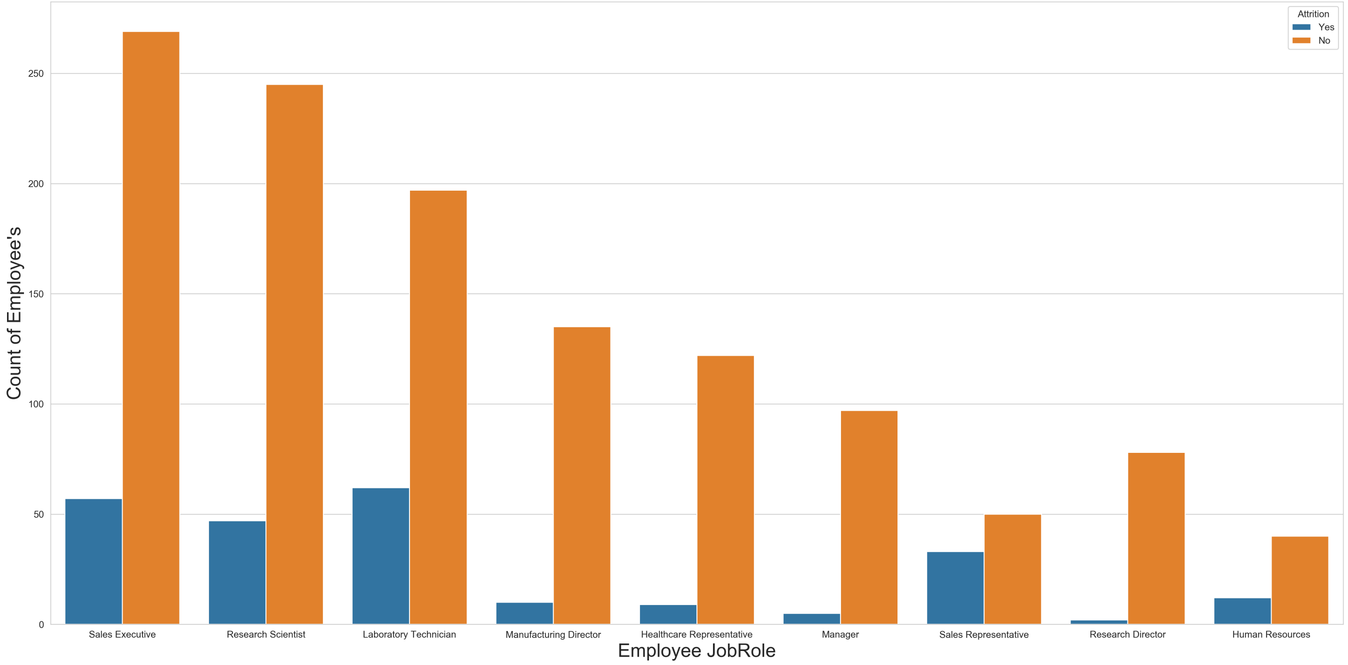
```
In [92]: plt.figure(figsize=(20,10),dpi = 200)
         sns.set_style("whitegrid")
         sns.countplot(x = "Education",hue="Attrition",data = df)
         plt.xlabel("Employee Education", size=20)
         plt.ylabel("Count of Employee's", size=20)
         plt.tight_layout()
         plt.savefig("Education.png")
```



In [17]:
```
df.groupby(by='JobRole')['Attrition'].value_counts()
```

Out[17]:
```
JobRole                    Attrition
Healthcare Representative  No           122
                           Yes            9
Human Resources            No            40
                           Yes           12
Laboratory Technician      No           197
                           Yes           62
Manager                    No            97
                           Yes            5
Manufacturing Director     No           135
                           Yes           10
Research Director          No            78
                           Yes            2
Research Scientist         No           245
                           Yes           47
Sales Executive            No           269
                           Yes           57
Sales Representative       No            50
                           Yes           33
Name: Attrition, dtype: int64
```

In [93]:
```
plt.figure(figsize=(20,10),dpi = 300)
sns.set_style("whitegrid")
sns.countplot(x = "JobRole",hue="Attrition",data  = df)
plt.xlabel("Employee JobRole", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("JobRole.png")
```

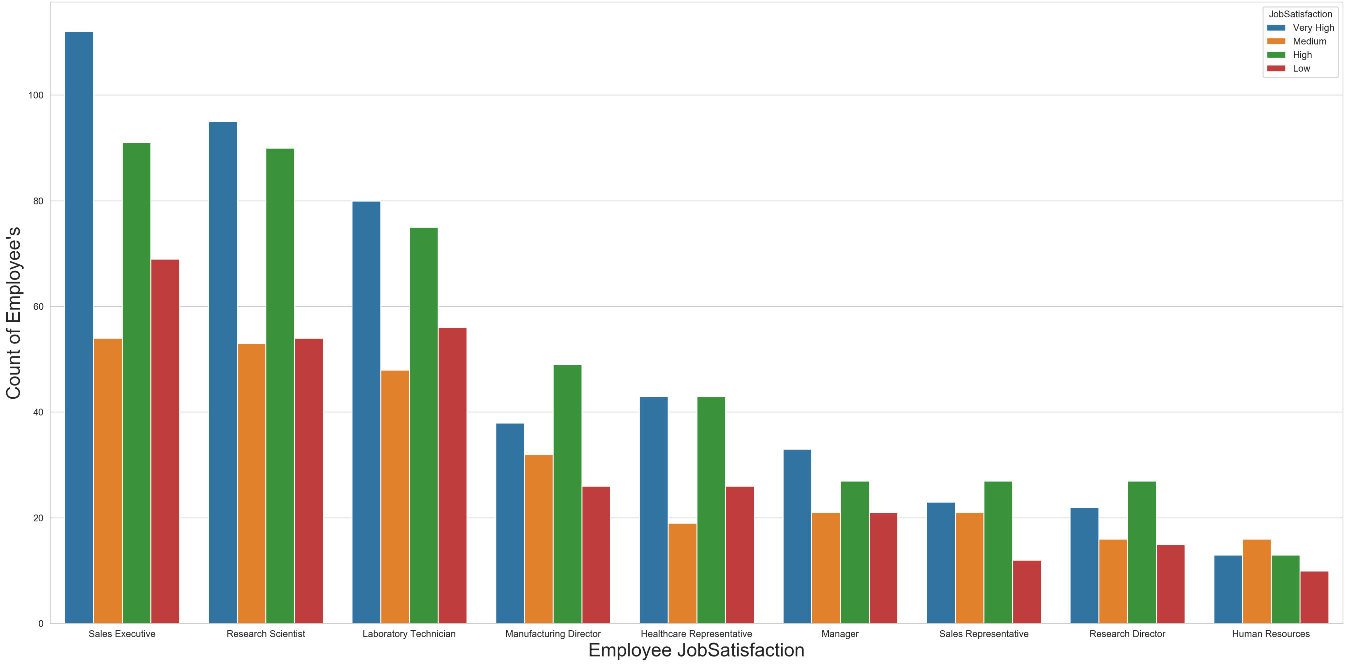


In [19]:
```
df.groupby(by='JobRole')['JobSatisfaction'].value_counts()
```

Out[19]:
```
JobRole                    JobSatisfaction
Healthcare Representative  High                43
                           Very High           43
                           Low                 26
                           Medium              19
Human Resources            Medium              16
                           High                13
                           Very High           13
                           Low                 10
Laboratory Technician      Very High           80
                           High                75
                           Low                 56
                           Medium              48
Manager                    Very High           33
                           High                27
                           Low                 21
                           Medium              21
Manufacturing Director     High                49
                           Very High           38
                           Medium              32
                           Low                 26
Research Director          High                27
                           Very High           22
                           Medium              16
                           Low                 15
Research Scientist         Very High           95
                           High                90
                           Low                 54
                           Medium              53
Sales Executive            Very High          112
                           High                91
                           Low                 69
                           Medium              54
Sales Representative       High                27
                           Very High           23
                           Medium              21
                           Low                 12
Name: JobSatisfaction, dtype: int64
```

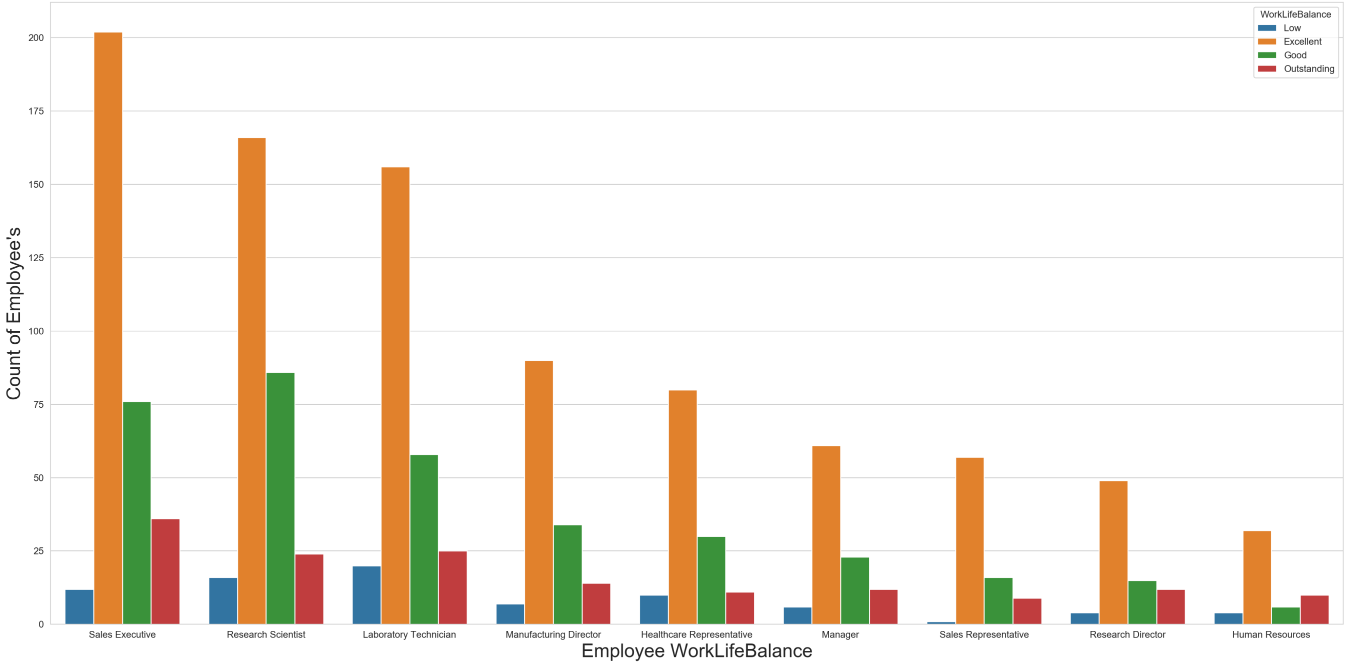
In [97]:

```
plt.figure(figsize=(20,10),dpi = 200)
sns.set_style("whitegrid")
sns.countplot(x = "JobRole",hue= "JobSatisfaction",data = df)
plt.xlabel("Employee JobSatisfaction", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("JobSatisfaction.png")
```

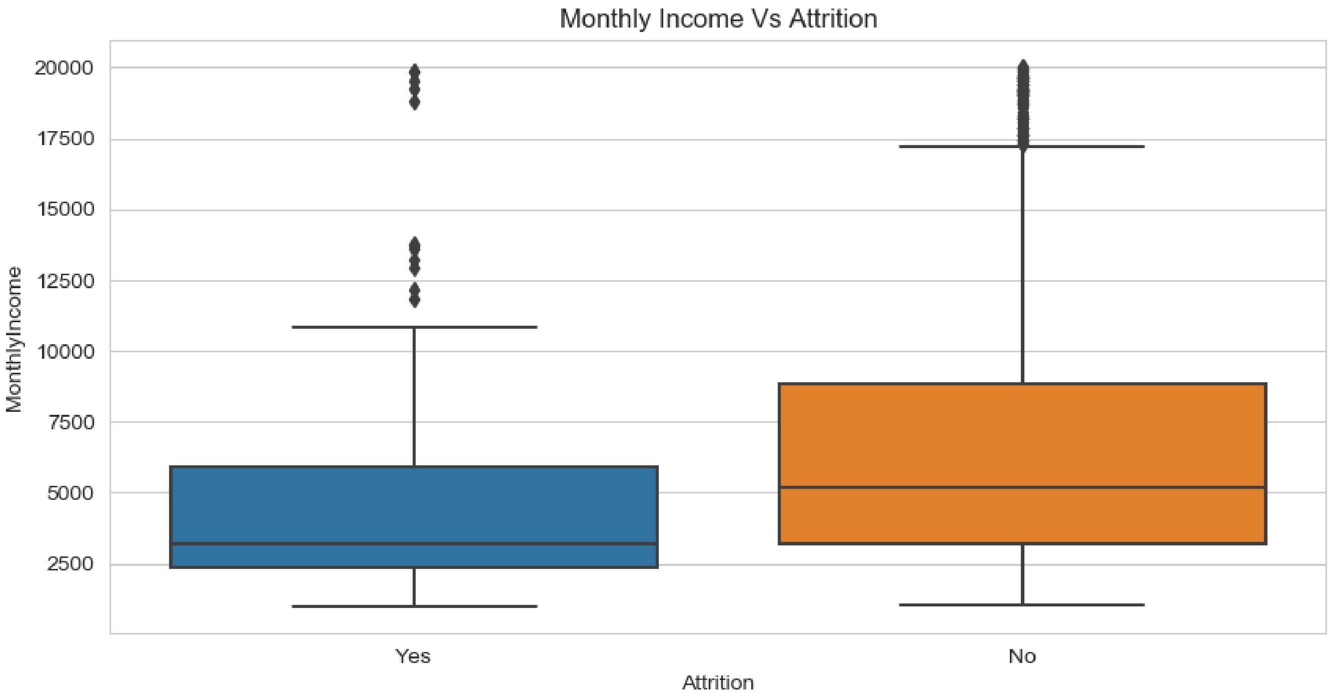


In [98]:

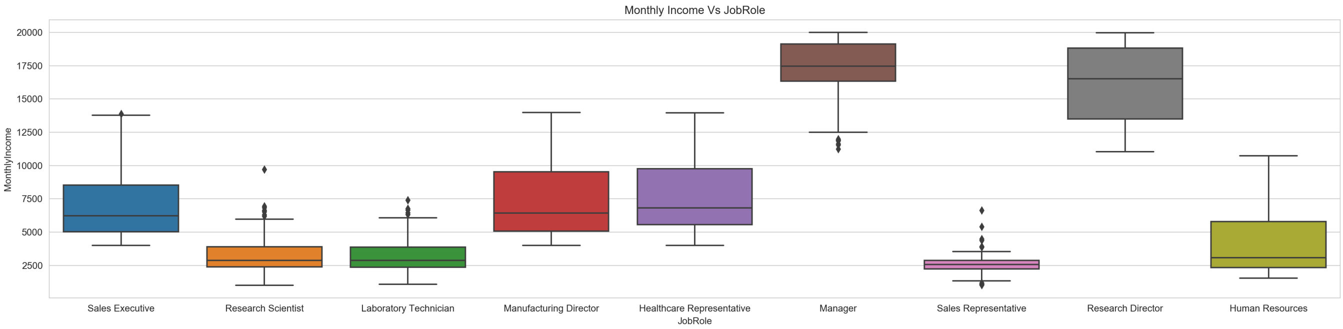
```
plt.figure(figsize=(20,10),dpi = 200)
sns.set_style("whitegrid")
sns.countplot(x ="JobRole",hue="WorkLifeBalance",data = df)
plt.xlabel("Employee WorkLifeBalance", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("WorkLifeBalance.png")
```



```
In [76]: plt.figure(figsize=(10,5),dpi = 100)
         sns.boxplot(y = 'MonthlyIncome' , x='Attrition' , data=df)
         plt.title("Monthly Income Vs Attrition");
         plt.savefig("MonthlyIncome.png")
```



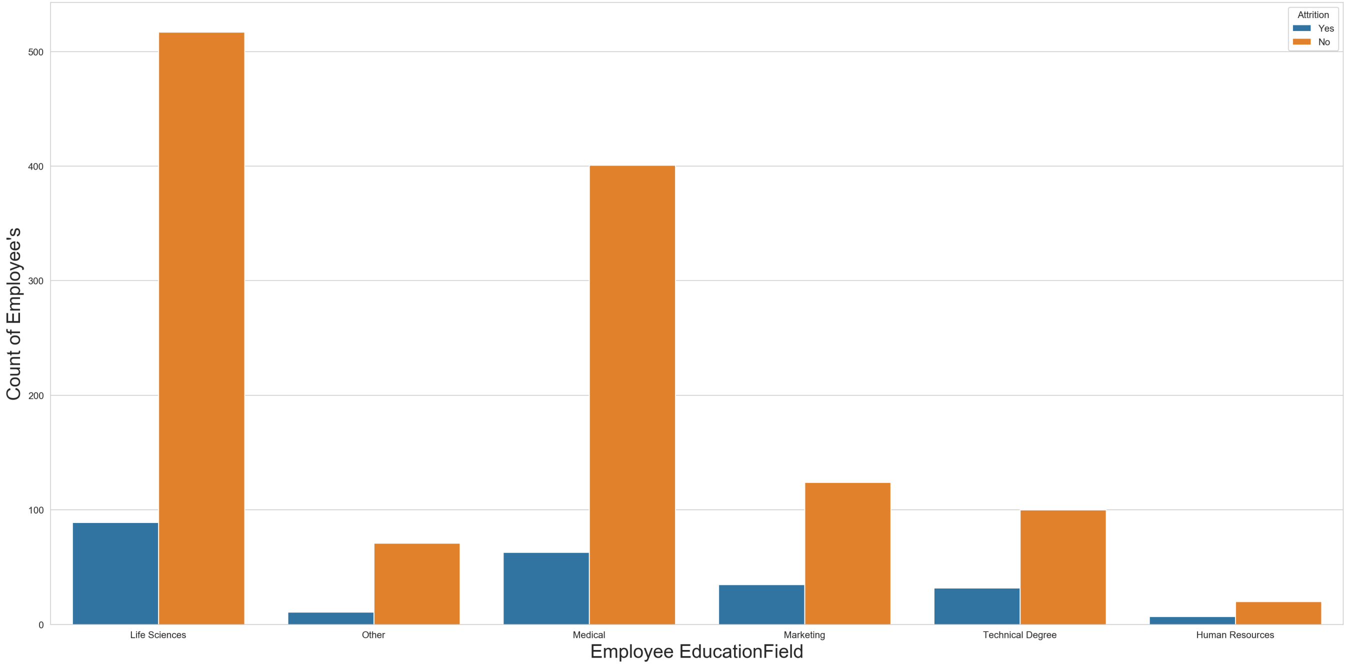
```
In [77]: plt.figure(figsize=(20,5),dpi = 200)
         sns.boxplot(y = 'MonthlyIncome' , x='JobRole' , data=df)
         plt.title("Monthly Income Vs JobRole");
         plt.tight_layout()
         plt.savefig("Monthly_JobRole.png")
```



```
In [68]: df.groupby(by='EducationField')['Attrition'].value_counts()
```

```
Out[68]: EducationField    Attrition
         Human Resources   No            20
                           Yes            7
         Life Sciences     No           517
                           Yes           89
         Marketing         No           124
                           Yes           35
         Medical           No           401
                           Yes           63
         Other             No            71
                           Yes           11
         Technical Degree  No           100
                           Yes           32
         Name: Attrition, dtype: int64
```

```
In [78]: plt.figure(figsize=(20,10),dpi = 200)
         sns.set_style("whitegrid")
         sns.countplot(x = "EducationField",hue= "Attrition", data = df)
         plt.xlabel("Employee EducationField", size=20)
         plt.ylabel("Count of Employee's", size=20)
         plt.tight_layout()
         plt.savefig("EducationField.png")
```

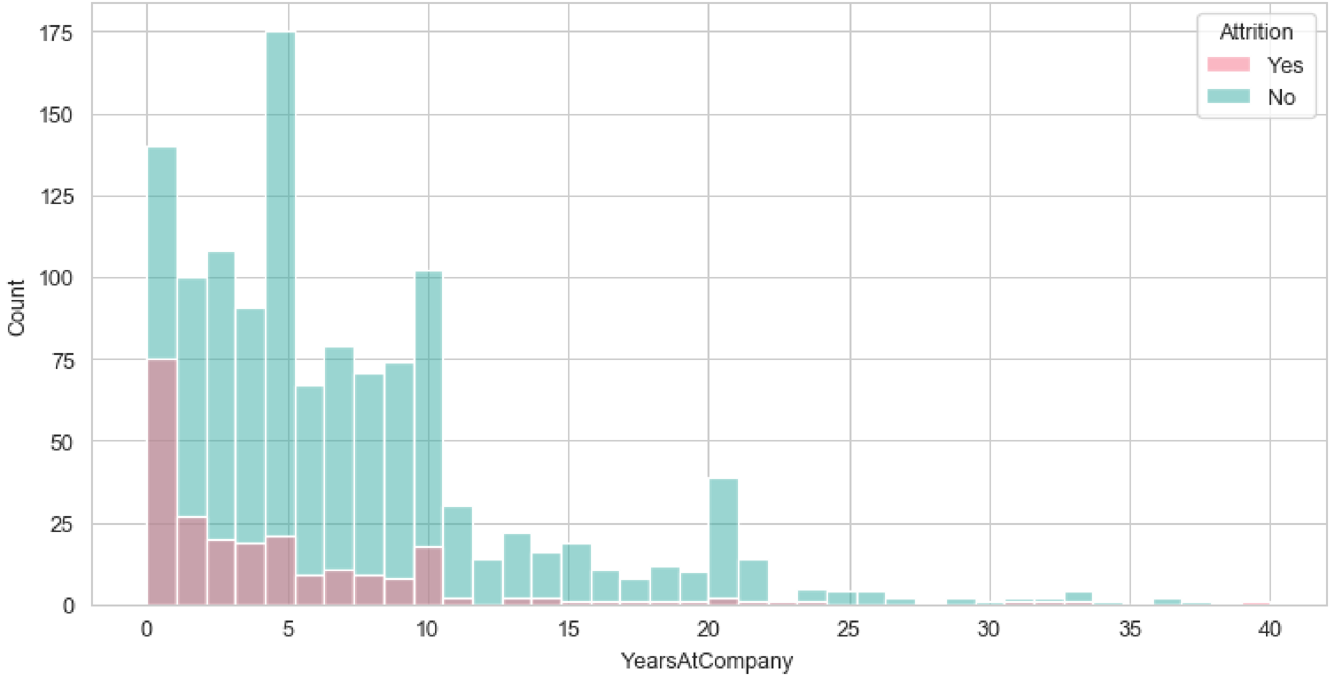


```
In [28]: df.groupby(by='Attrition')['PerformanceRating'].value_counts()
```

```
Out[28]: Attrition  PerformanceRating
         No         Excellent            1044
                    Outstanding           189
         Yes        Excellent             200
                    Outstanding            37
         Name: PerformanceRating, dtype: int64
```

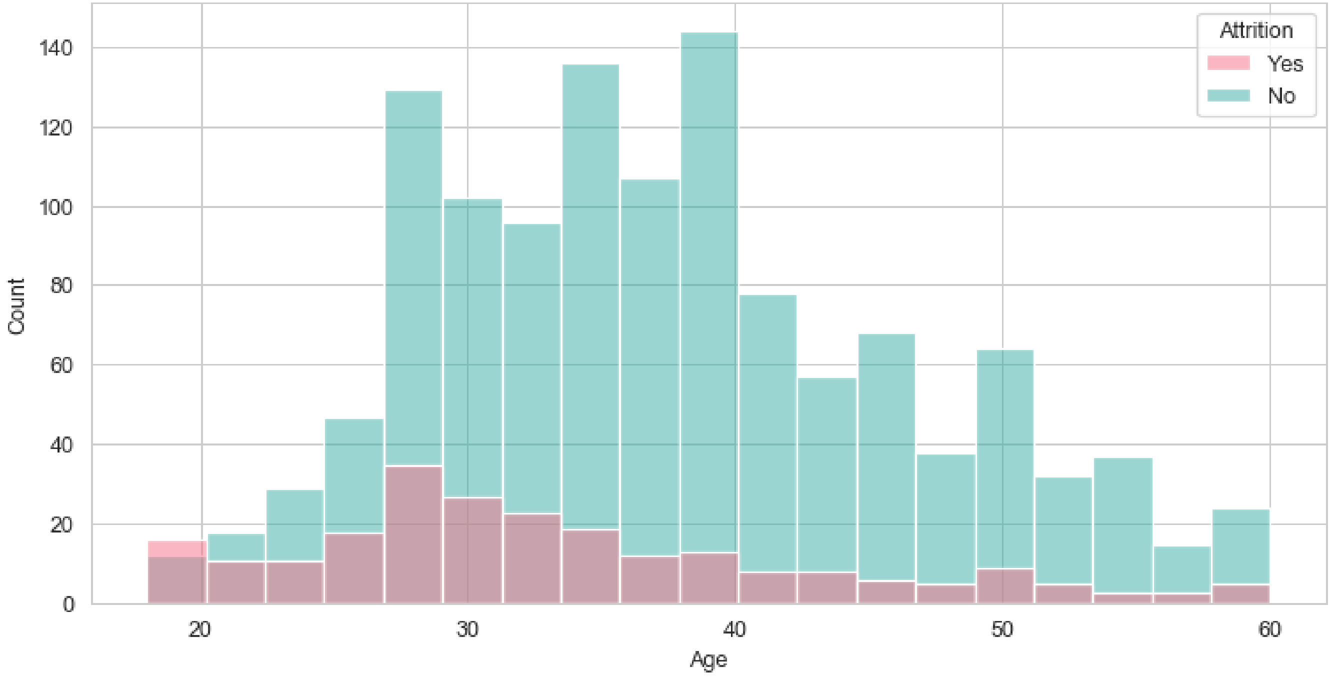
In [79]:

```
plt.figure(figsize=(10 , 5),dpi = 90)
sns.histplot(x='YearsAtCompany' , hue='Attrition' ,data=df  ,palette="husl" , edgecolor=
'white');
plt.savefig("YearsAtCompany.png")
```



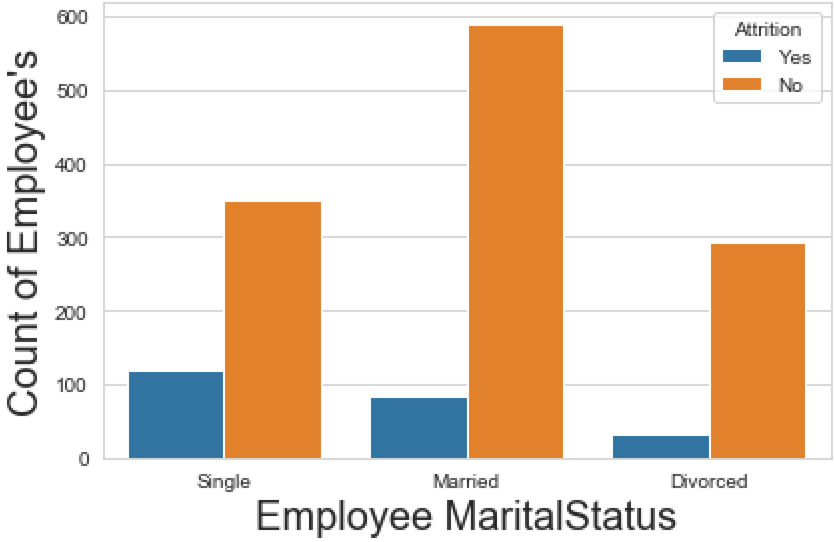
In [82]:

```
plt.figure(figsize=(10 , 5),dpi = 90)
sns.histplot(data = df, x="Age", hue="Attrition",palette="husl");
plt.savefig("age_Attrition.png")
```



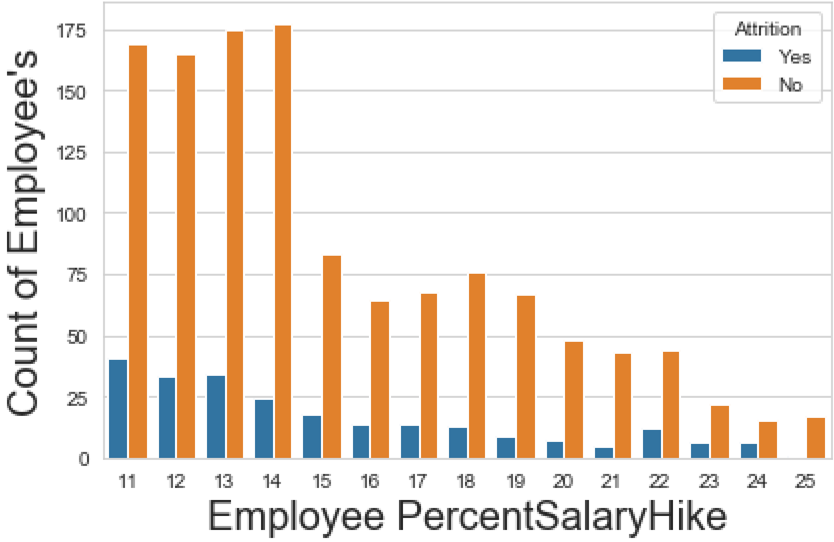
In [99]:

```
sns.set_style("whitegrid")
sns.countplot(x = "MaritalStatus",hue="Attrition",data = df)
plt.xlabel("Employee MaritalStatus", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("MaritalStatus.png")
```



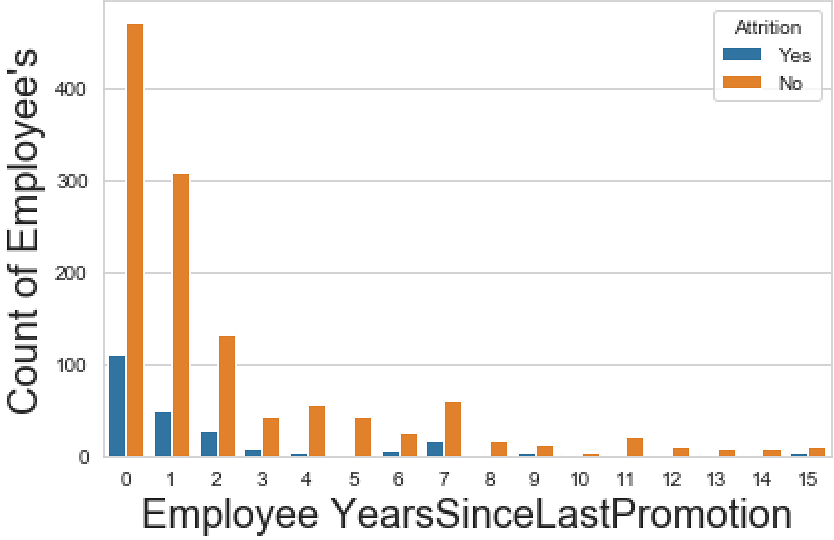
In [89]:

```
sns.set_style("whitegrid")
sns.countplot(x = "PercentSalaryHike",hue="Attrition",data = df)
plt.xlabel("Employee PercentSalaryHike", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("PercentSalaryHike.png")
```



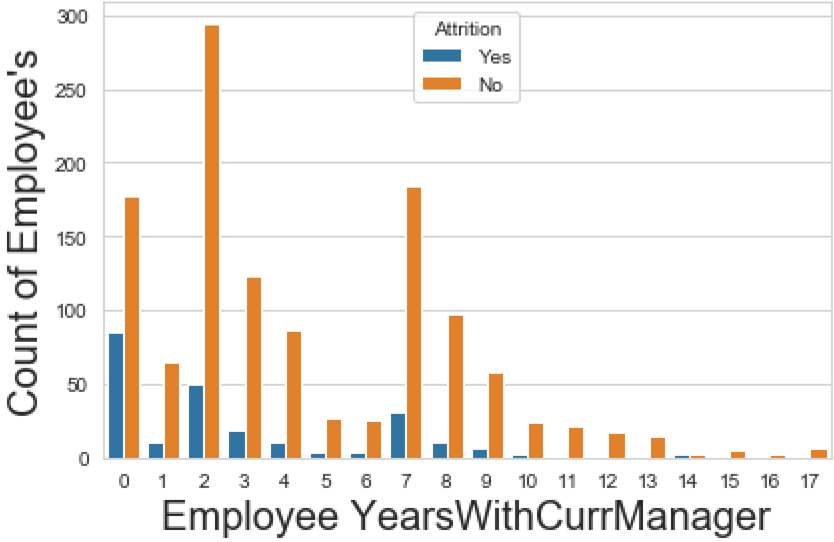
In [90]:

```python
sns.set_style("whitegrid")
sns.countplot(x = "YearsSinceLastPromotion",hue="Attrition",data = df)
plt.xlabel("Employee YearsSinceLastPromotion", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("YearsSinceLastPromotion.png")
```



In [91]:

```python
sns.set_style("whitegrid")
sns.countplot(x = "YearsWithCurrManager",hue="Attrition",data = df)
plt.xlabel("Employee YearsWithCurrManager", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("YearsWithCurrManager.png")
```



In [146]:

```python
df_temp = pd.DataFrame()
```

In [153]:

```python
df_temp["MonthlyIncome_mean"] = df.MonthlyIncome.mean()
```

In [154]:

```python
df_temp["Laboratory_Technician"] = df[(df["MonthlyIncome"]<=6502)&(df.JobRole == "Labora
tory Technician") \
                                      &(df.Attrition== "Yes")]["Attrition"].value_counts
()
```

In [155]:

```
df_temp["Sales_Executive"] = df[(df["MonthlyIncome"]<=6502)&(df.JobRole == "Sales Executive")\
                                &(df.Attrition== "Yes")]["Attrition"].value_counts()
```

In [156]:

```
df_temp["Research_Scientist"]  = df[(df["MonthlyIncome"]<=6502)&(df.JobRole == "Research Scientist")&\
                                    (df.Attrition== "Yes")]["Attrition"].value_counts()
```

In [157]:

```
df_temp["Sales_Representative"]  = df[(df["MonthlyIncome"]<=6502)&(df.JobRole == "Sales Representative")\
                                      &(df.Attrition== "Yes")]["Attrition"].value_counts()
```

In [158]:

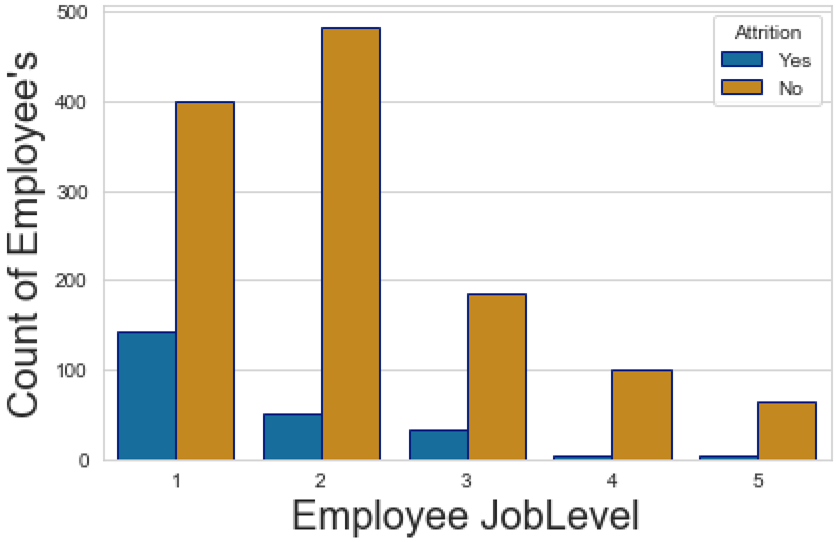
```
df_temp
```

Out[158]:

	MonthlyIncome_mean	Laboratory_Technician	Sales_Executive	Research_Scientist	Sales_Representative
Yes	6502.931293	62	27	47	33

In [142]:

```
sns.set_style("whitegrid")
sns.countplot(x='JobLevel', hue='Attrition', data = df, palette="colorblind",  edgecolor=sns.color_palette("dark", n_colors = 1))
plt.xlabel("Employee JobLevel", size=20)
plt.ylabel("Count of Employee's", size=20)
plt.tight_layout()
plt.savefig("JobLevel.png")
```


Count of Employee's
Attrition
Yes
No
Employee JobLevel