Prediction Of Employee Turnover

Importing Library

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

Loading Dataset

```
df = pd.read_csv("/content/HR_Analytics.csv.csv")
```

```
df
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	...
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	

1470 rows × 35 columns

Data Exploration

```
df.head()
```

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

5 rows × 35 columns

```
df.tail()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	

5 rows × 35 columns

```
df.shape
```

```
(1470, 35)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

```
df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000

8 rows × 26 columns

Checking for Missing Values

```
df.isnull()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
1465	False	False	False	False	False	False	False	False	False	
1466	False	False	False	False	False	False	False	False	False	
1467	False	False	False	False	False	False	False	False	False	
1468	False	False	False	False	False	False	False	False	False	
1469	False	False	False	False	False	False	False	False	False	

1470 rows × 35 columns

```
df.isnull().sum()
```

	0
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

```
df.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

```
df['Age'].unique()
```

```
array([41, 49, 37, 33, 27, 32, 59, 30, 38, 36, 35, 29, 31, 34, 28, 22, 53,
       24, 21, 42, 44, 46, 39, 43, 50, 26, 48, 55, 45, 56, 23, 51, 40, 54,
       58, 20, 25, 19, 57, 52, 47, 18, 60])
```

```
df['Age']
```

	Age
0	41
1	49
2	37
3	33
4	27
...	...
1465	36
1466	39
1467	27
1468	49
1469	34

1470 rows × 1 columns

dtype: int64

```
df['Age'].unique().sum()
```

```
1677
```

Feature Engineering

```
age_range = [0, 28, 46, 58, 100]
labels = ['Gen Z', 'Millennials', 'Gen X', 'Baby Boomers']
```

```
df['age_group'] = pd.cut(df['Age'], bins=age_range, labels=labels)
```

```
age_group_counts=df['age_group'].value_counts()
```

```
print(age_group_counts)
```

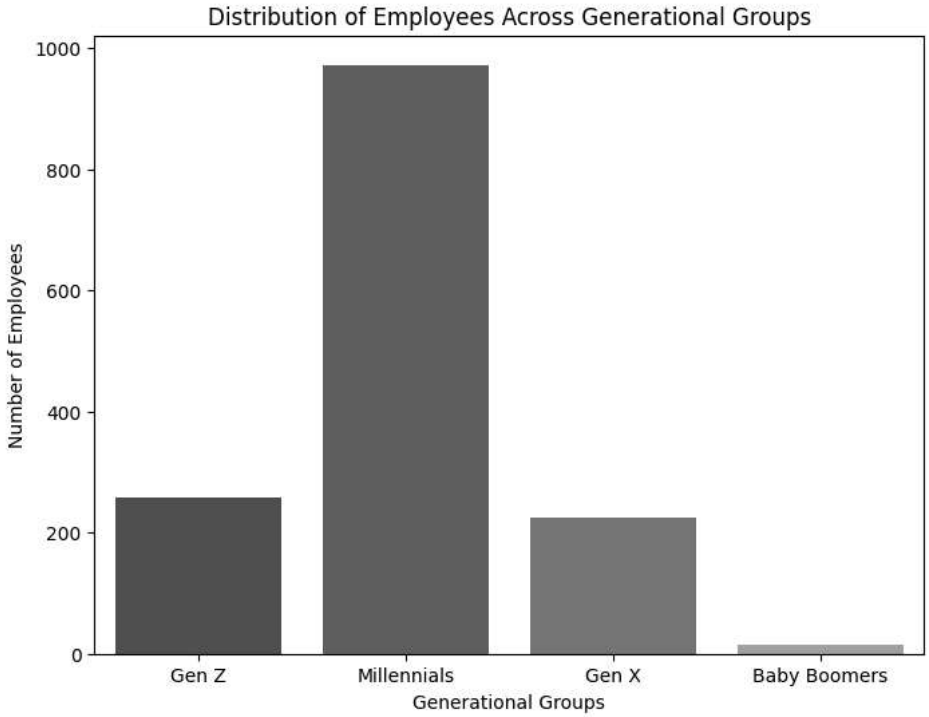
```
age_group
Millennials     972
Gen Z           258
Gen X           225
Baby Boomers     15
Name: count, dtype: int64
```

```
plt.figure(figsize=(8, 6))
sns.barplot(x=age_group_counts.index, y=age_group_counts.values, palette='viridis')
plt.xlabel('Generational Groups')
plt.ylabel('Number of Employees')
plt.title('Distribution of Employees Across Generational Groups')
plt.show()
```

```
<ipython-input-19-50b363d29099>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.barplot(x=age_group_counts.index, y=age_group_counts.values, palette='viridis')
```




```
df.loc[df.Attrition == 'Yes']
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
14	28	Yes	Travel_Rarely	103	Research & Development	24	3	Life Sciences	1	
21	36	Yes	Travel_Rarely	1218	Sales	9	4	Life Sciences	1	
24	34	Yes	Travel_Rarely	699	Research & Development	6	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	...
1438	23	Yes	Travel_Frequently	638	Sales	9	3	Marketing	1	
1442	29	Yes	Travel_Rarely	1092	Research & Development	1	4	Medical	1	
1444	56	Yes	Travel_Rarely	310	Research & Development	7	2	Technical Degree	1	
1452	50	Yes	Travel_Frequently	878	Sales	1	4	Life Sciences	1	
1461	50	Yes	Travel_Rarely	410	Sales	28	3	Marketing	1	

237 rows × 36 columns

```
df_attrition = df.loc[df.Attrition == 'Yes']
df_attrition
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
14	28	Yes	Travel_Rarely	103	Research & Development	24	3	Life Sciences	1	
21	36	Yes	Travel_Rarely	1218	Sales	9	4	Life Sciences	1	
24	34	Yes	Travel_Rarely	699	Research & Development	6	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	...
1438	23	Yes	Travel_Frequently	638	Sales	9	3	Marketing	1	
1442	29	Yes	Travel_Rarely	1092	Research & Development	1	4	Medical	1	
1444	56	Yes	Travel_Rarely	310	Research & Development	7	2	Technical Degree	1	
1452	50	Yes	Travel_Frequently	878	Sales	1	4	Life Sciences	1	
1461	50	Yes	Travel_Rarely	410	Sales	28	3	Marketing	1	

237 rows × 36 columns

Data Cleaning

```
df['Education'].replace([1,2,3,4,5],['High School','Associates Degree', 'Bachelors Degree', 'Masters Degree', 'Doctorate'],inplace = Tru
```

Filtering Attrition Data

```
df_attrition
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNu
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	
14	28	Yes	Travel_Rarely	103	Research & Development	24	3	Life Sciences	1	
21	36	Yes	Travel_Rarely	1218	Sales	9	4	Life Sciences	1	
24	34	Yes	Travel_Rarely	699	Research & Development	6	1	Medical	1	
...	...	...	...	...	...	...	...	...	...	...
1438	23	Yes	Travel_Frequently	638	Sales	9	3	Marketing	1	
1442	29	Yes	Travel_Rarely	1092	Research & Development	1	4	Medical	1	
1444	56	Yes	Travel_Rarely	310	Research & Development	7	2	Technical Degree	1	
1452	50	Yes	Travel_Frequently	878	Sales	1	4	Life Sciences	1	
1461	50	Yes	Travel_Rarely	410	Sales	28	3	Marketing	1	

237 rows × 36 columns

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

Attrition	count
No	1233
Yes	237

dtype: int64

```
df_attrition['EducationField'].unique()
```

```
array(['Life Sciences', 'Other', 'Medical', 'Technical Degree',
       'Marketing', 'Human Resources'], dtype=object)
```

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

Gender	count
Male	150
Female	87

dtype: int64

```
df_attrition['age_group'].value_counts()
```

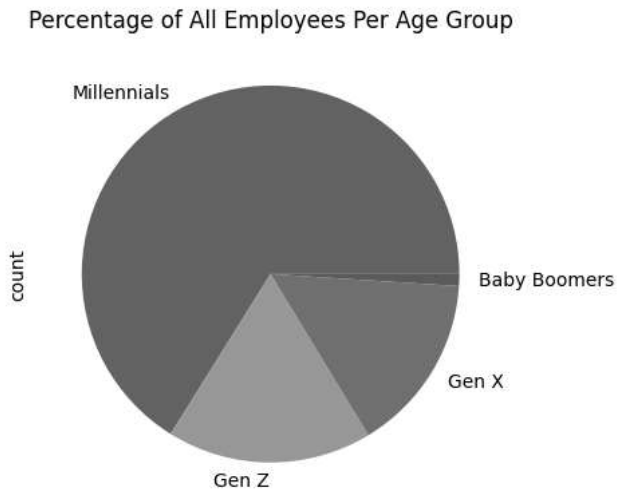
age_group	count
Millennials	134
Gen Z	73
Gen X	30
Baby Boomers	0

dtype: int64

Visualization

```
plt.figure(figsize = (10,10))
plt.subplot(1,2,1)
df['age_group'].value_counts().plot(kind='pie', title = "Percentage of All Employees Per Age Group")
```

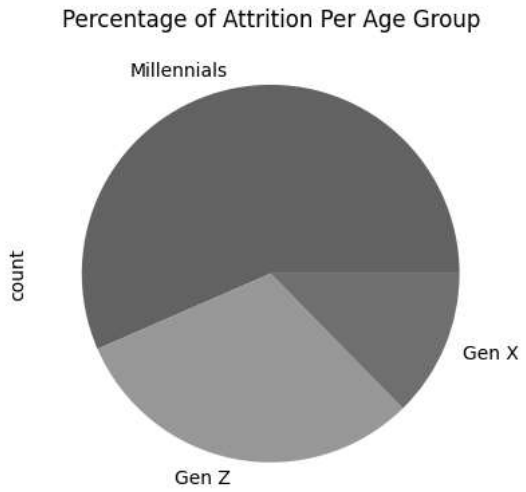
```
<Axes: title={'center': 'Percentage of All Employees Per Age Group'}, ylabel='count'>
```



The pie chart shows that Millennials are the largest age group in the dataset, followed by Gen X and Baby Boomers. This suggests that the company has a relatively younger workforce.

```
plt.figure(figsize = (10,10))
plt.subplot(1,2,2)
df_attrition['age_group'].value_counts().plot(kind='pie', title = "Percentage of Attrition Per Age Group")
```

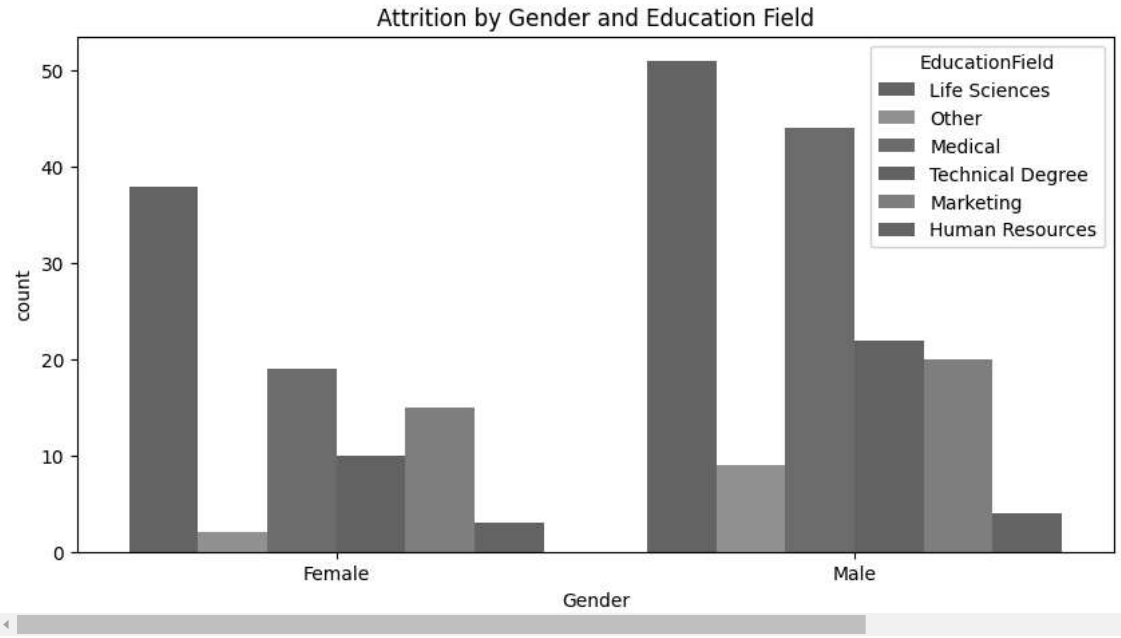
```
<Axes: title={'center': 'Percentage of Attrition Per Age Group'}, ylabel='count'>
```



The pie chart shows that Millennials have the highest attrition rate compared to other age groups.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Gender and Education Field")
sns.countplot(x = 'Gender', hue = 'EducationField', data = df_attrition)
```

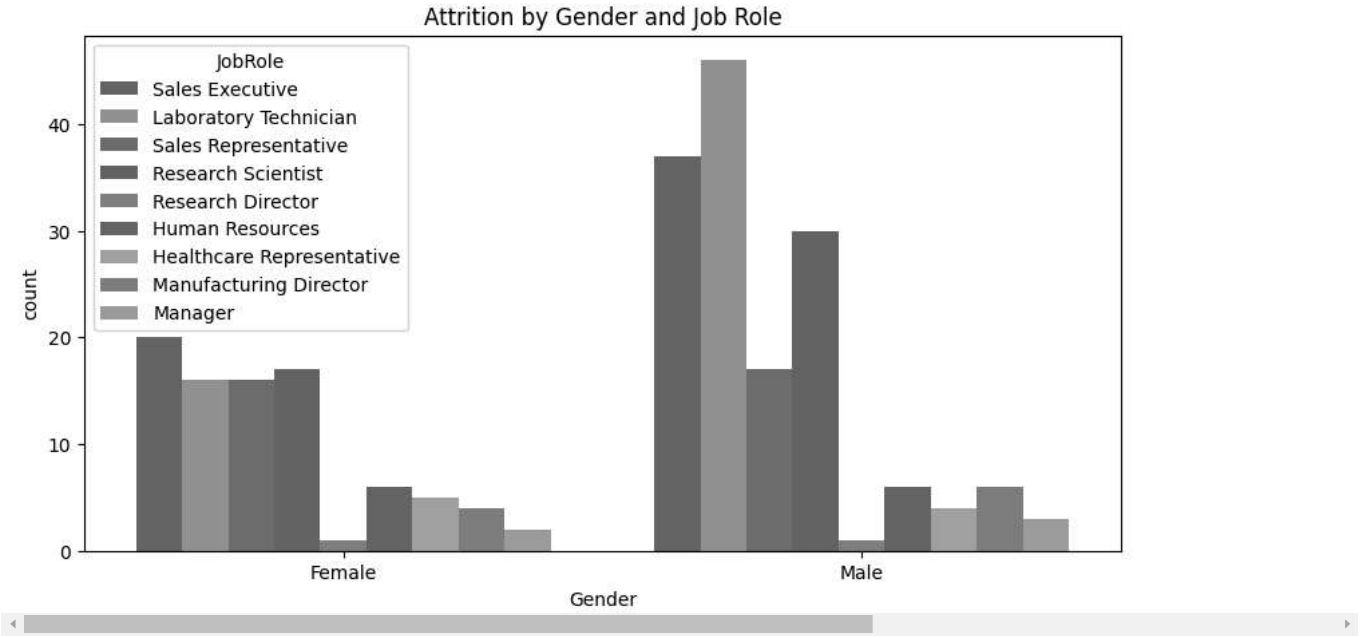
```
<Axes: title={'center': 'Attrition by Gender and Education Field'}, xlabel='Gender', ylabel='count'>
```



The attrition level is higher among male employees. Employees with education in life sciences and medicine leave the company at a higer rate than other professionals

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Gender and Job Role")
sns.countplot(x = 'Gender', hue = 'JobRole', data = df_attrition)
```

```
<Axes: title={'center': 'Attrition by Gender and Job Role'}, xlabel='Gender', ylabel='count'>
```

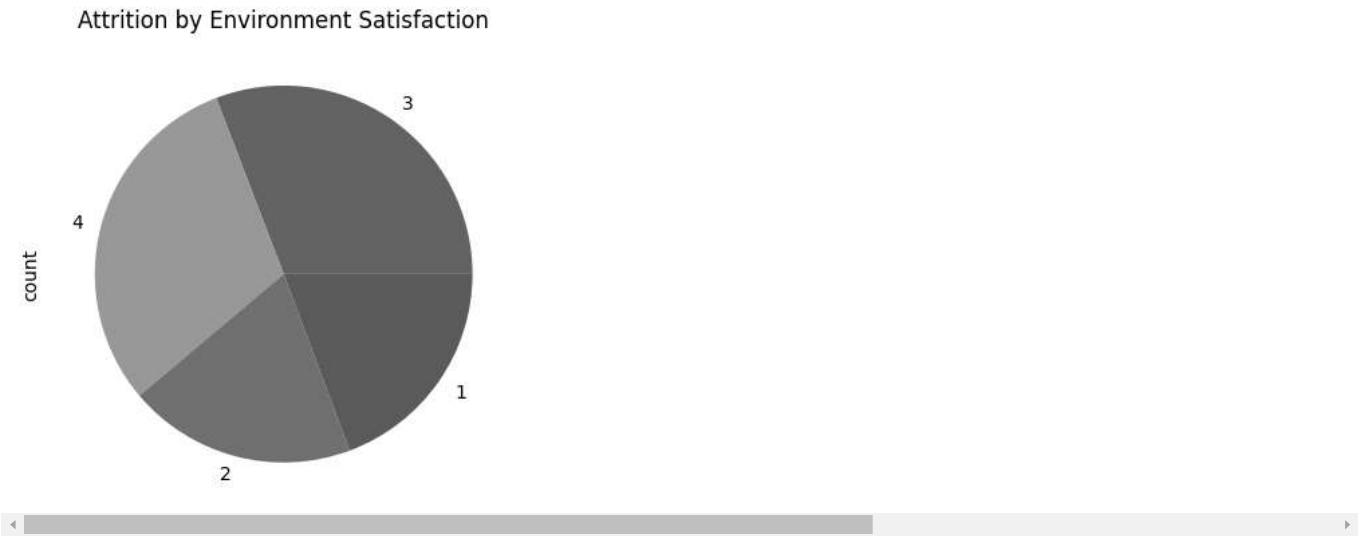


Attrition levels are highest among men performing jobs as Sales Executives, Laboratory Technicians and Research Scientists. For women, attrition is highest at the Sales Executive level. Men and women are less likely to quit the Research Director or Manager job role

```
# Show Percentage of Employees per Age Group

plt.figure(figsize = (10,10))
plt.subplot(1,2,1)
df['EnvironmentSatisfaction'].value_counts().plot(kind='pie', title = "Attrition by Environment Satisfaction")
```

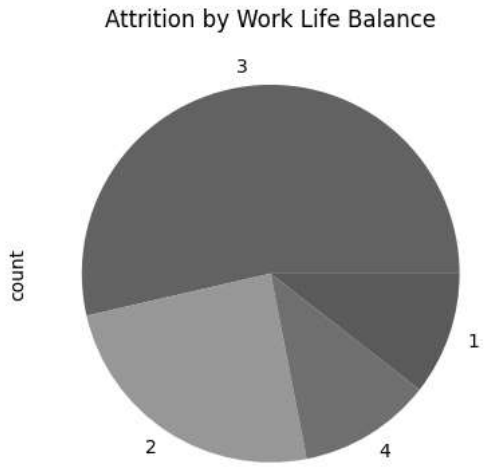
```
<Axes: title={'center': 'Attrition by Environment Satisfaction'}, ylabel='count'>
```



The pie chart shows that the majority of employees have a high level of environment satisfaction. This suggests that the company has a positive work environment.

```
plt.figure(figsize = (10,10))
plt.subplot(1,2,2)
df_attrition['WorkLifeBalance'].value_counts().plot(kind='pie', title = "Attrition by Work Life Balance")
```

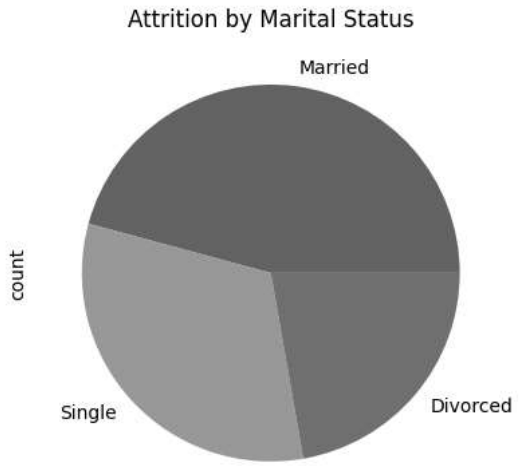
```
<Axes: title={'center': 'Attrition by Work Life Balance'}, ylabel='count'>
```



The pie chart shows that the majority of employees who left the company had a lower level of work-life balance.

```
# Percentage of Employees per Age Group
plt.figure(figsize = (10,10))
plt.subplot(1,2,1)
df['MaritalStatus'].value_counts().plot(kind='pie', title = "Attrition by Marital Status")
```

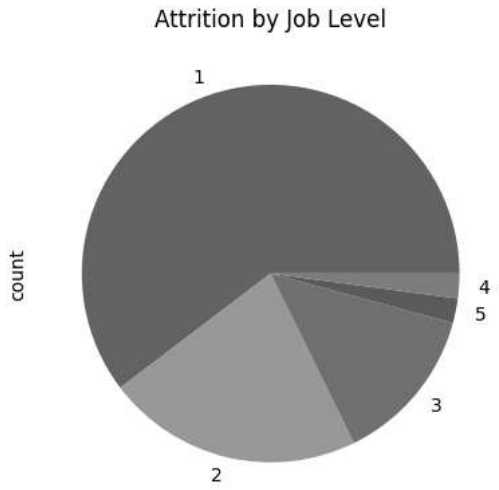
```
<Axes: title={'center': 'Attrition by Marital Status'}, ylabel='count'>
```



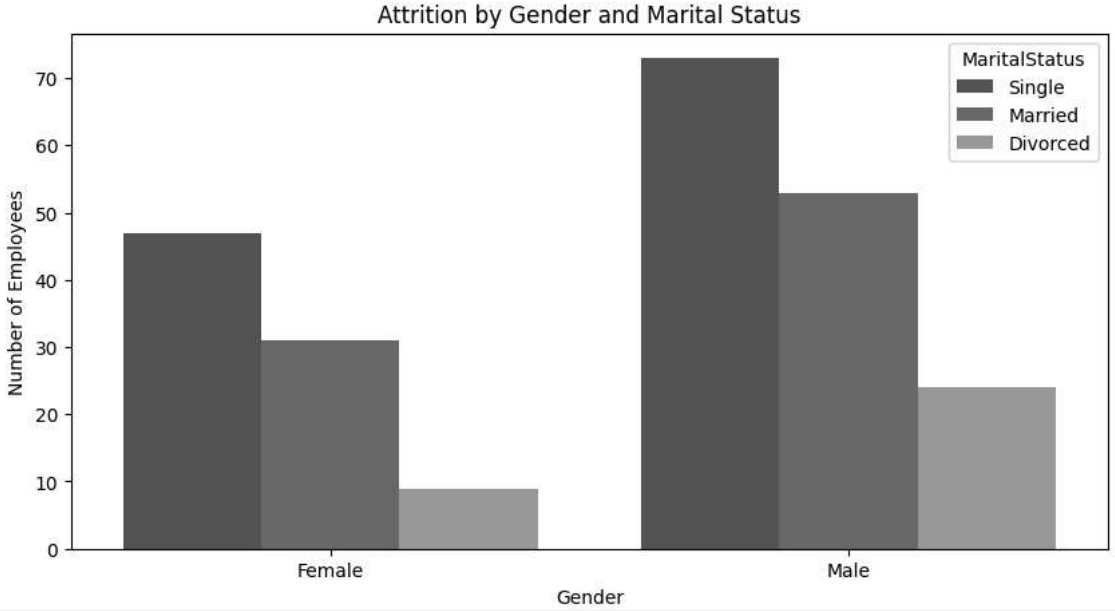
The pie chart shows that the majority of employees are married.

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,2)
df_attrition['JobLevel'].value_counts().plot(kind='pie', title = "Attrition by Job Level")
```

```
<Axes: title={'center': 'Attrition by Job Level'}, ylabel='count'>
```



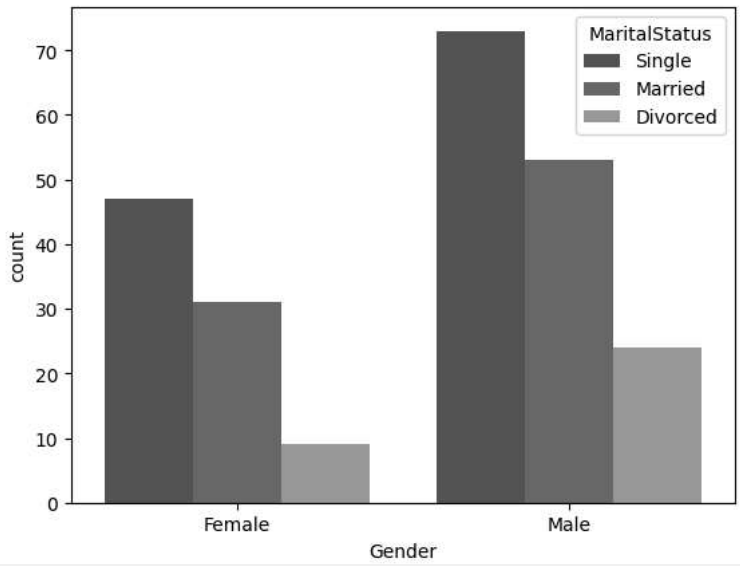

```
plt.figure(figsize=(10, 5))
plt.title("Attrition by Gender and Marital Status")
sns.countplot(x='Gender', hue='MaritalStatus', data=df_attrition, palette='viridis')
plt.xlabel('Gender')
plt.ylabel('Number of Employees')
plt.show()
```



There are higher level of single men employee in the company.

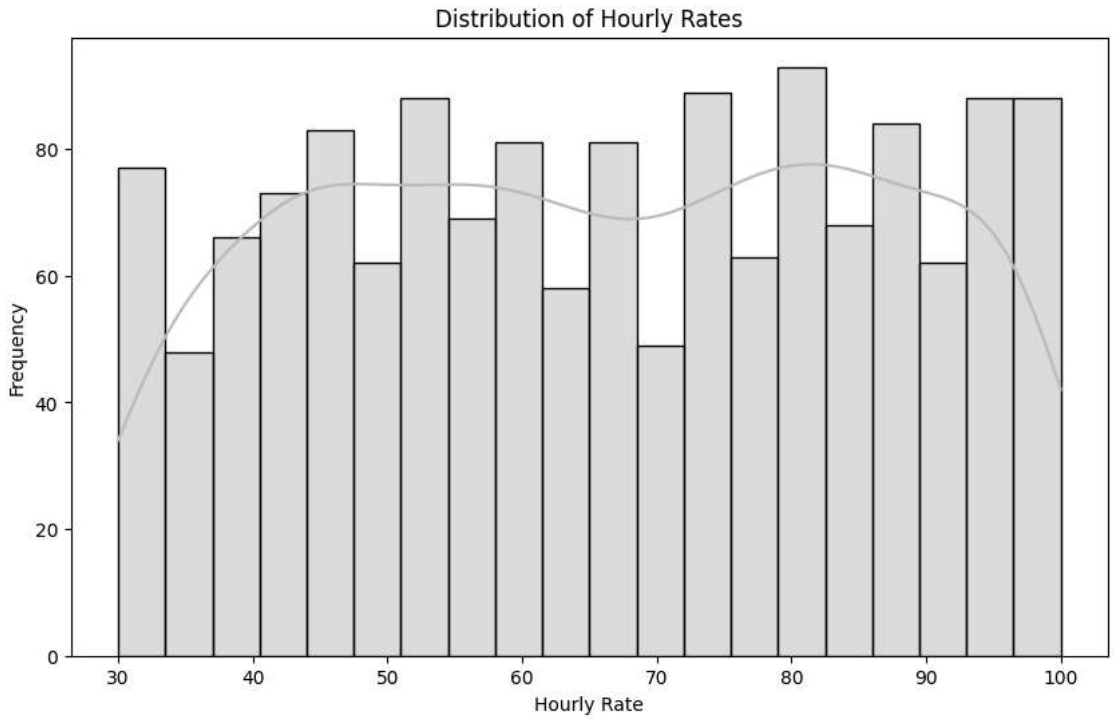
```
df_attrition_yes = df_attrition[df_attrition['Attrition'] == 'Yes']
sns.countplot(x='Gender', hue='MaritalStatus', data=df_attrition_yes, palette='viridis')
```

```
<Axes: xlabel='Gender', ylabel='count'>
```



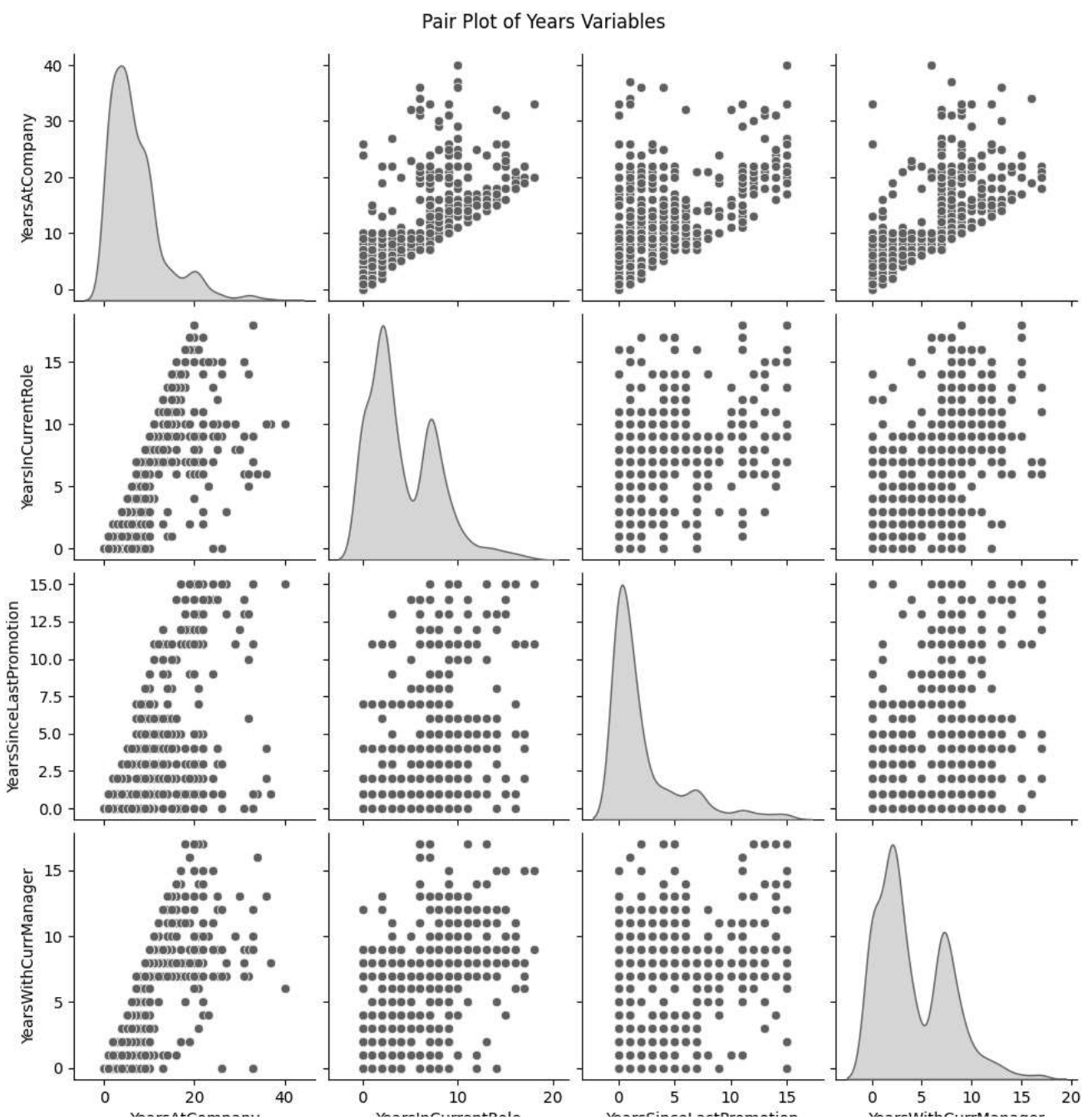
Attrition levels are highest among single men.

```
plt.figure(figsize=(10, 6))
sns.histplot(df['HourlyRate'], bins=20, kde=True, color='skyblue')
plt.title('Distribution of Hourly Rates')
plt.xlabel('Hourly Rate')
plt.ylabel('Frequency')
plt.show()
```



The distribution of hourly rates appears to be roughly normal, with a slight skew towards the lower end. This suggests that the majority of employees are paid within a certain range, with a smaller number of employees

```
df_subset = df[['YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager']]
sns.pairplot(df_subset, diag_kind='kde')
plt.suptitle('Pair Plot of Years Variables', y=1.02)
plt.show()
```



- **Positive Correlation:** There seems to be a positive correlation between 'YearsAtCompany' and 'YearsInCurrentRole', 'YearsSinceLastPromotion', and 'YearsWithCurrManager'. This makes sense as employees tend to spend more time in the company overall, they also tend to stay longer in their current role and with the same manager.
- **Years Since Last Promotion:** The relationship between 'YearsAtCompany' and 'YearsSinceLastPromotion' suggests that some employees stay in the company for a long time without getting promoted. This could be a potential factor contributing to attrition.
- **Years with Current Manager:** There's a noticeable positive correlation between 'YearsAtCompany' and 'YearsWithCurrManager'. This implies that employees tend to stay with the same manager for a significant period.

Years At Company

```
plt.figure(figsize=(16, 12))
plt.subplot(2, 2, 1)
sns.histplot(df['YearsAtCompany'], bins=20, kde=True, color='blue')
plt.title('Distribution of Years At Company')
plt.xlabel('Years At Company')
plt.ylabel('Frequency')
```

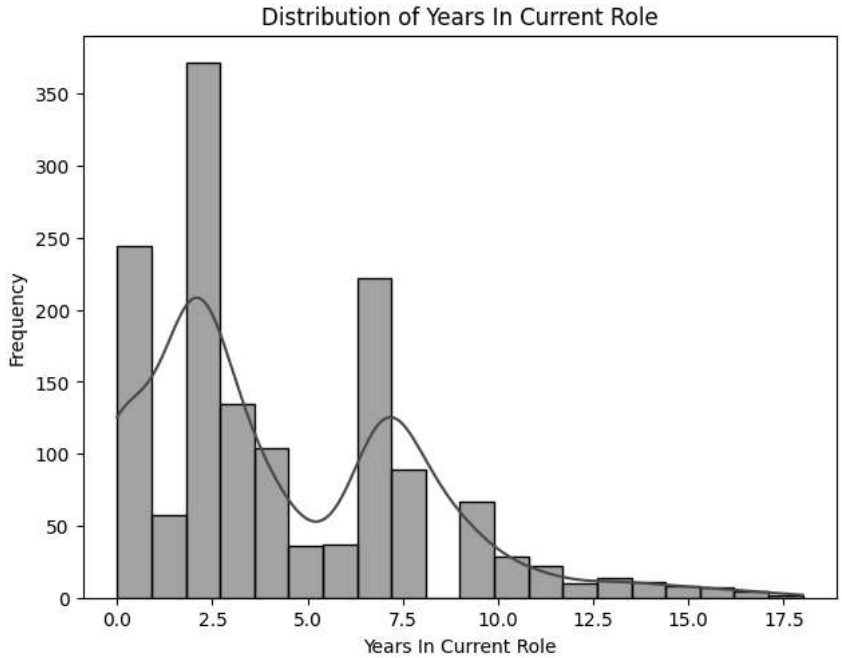
```
Text(0, 0.5, 'Frequency')
```

Many employees are relatively new to the company (0-5 years). A noticeable number of employees have 10-15 years of service.

Years In Current Role

```
plt.figure(figsize=(16, 12))
plt.subplot(2, 2, 2)
sns.histplot(df['YearsInCurrentRole'], bins=20, kde=True, color='green')
plt.title('Distribution of Years In Current Role')
plt.xlabel('Years In Current Role')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```



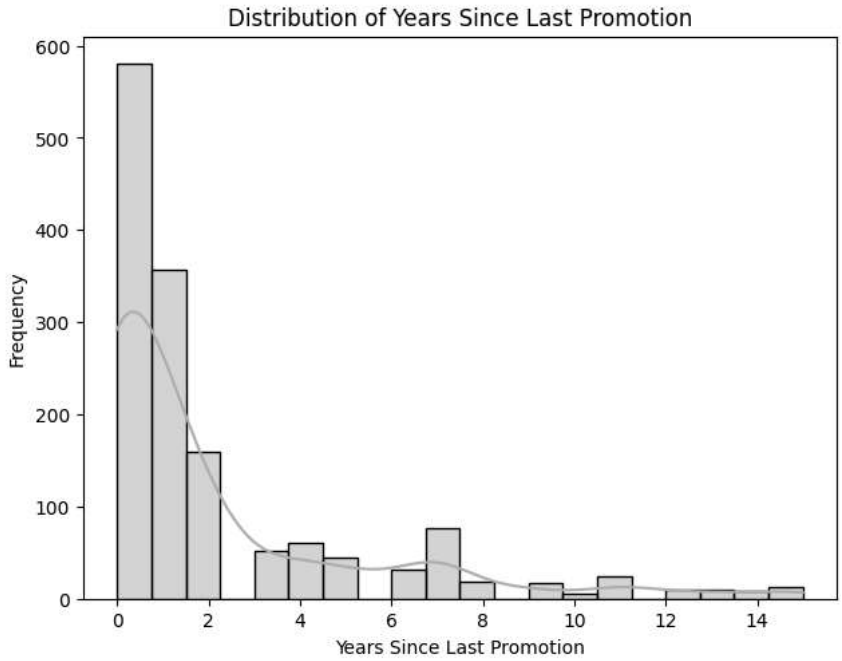
Many employees are relatively new to their current role (0-5 years).

Years Since Last Promotion

```
plt.figure(figsize=(16, 12))
plt.subplot(2, 2, 3)
sns.histplot(df['YearsSinceLastPromotion'], bins=20, kde=True, color='orange')
plt.title('Distribution of Years Since Last Promotion')
```

```
plt.xlabel('Years Since Last Promotion')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```

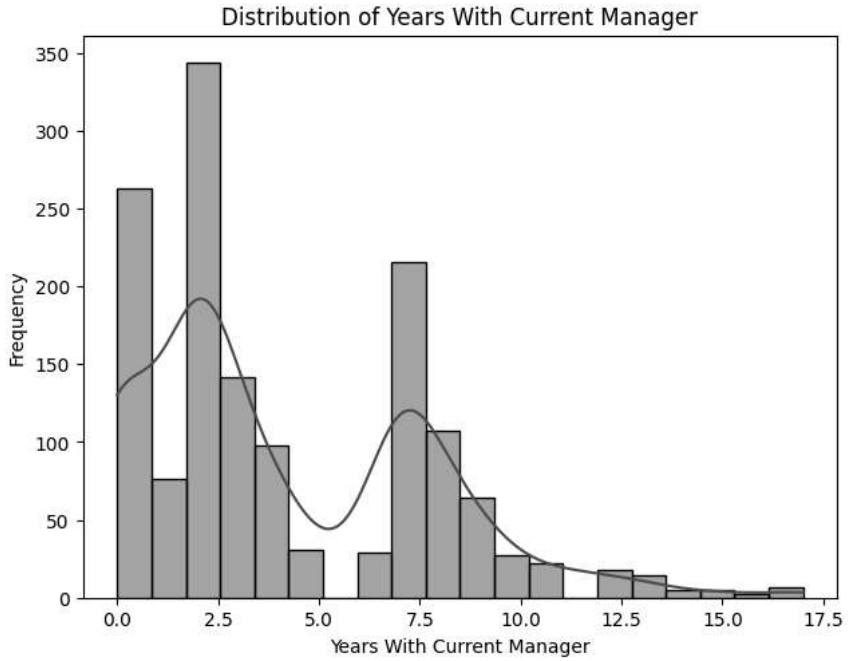


Many employees have not been promoted recently (0-5 years).

Years With Current Manager

```
plt.figure(figsize=(16,12))
plt.subplot(2, 2, 4)
sns.histplot(df['YearsWithCurrManager'], bins=20, kde=True, color='red')
plt.title('Distribution of Years With Current Manager')
plt.xlabel('Years With Current Manager')
plt.ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```



Many employees have been with their current manager for 0-5 years.

```
df['Department'].unique()
```

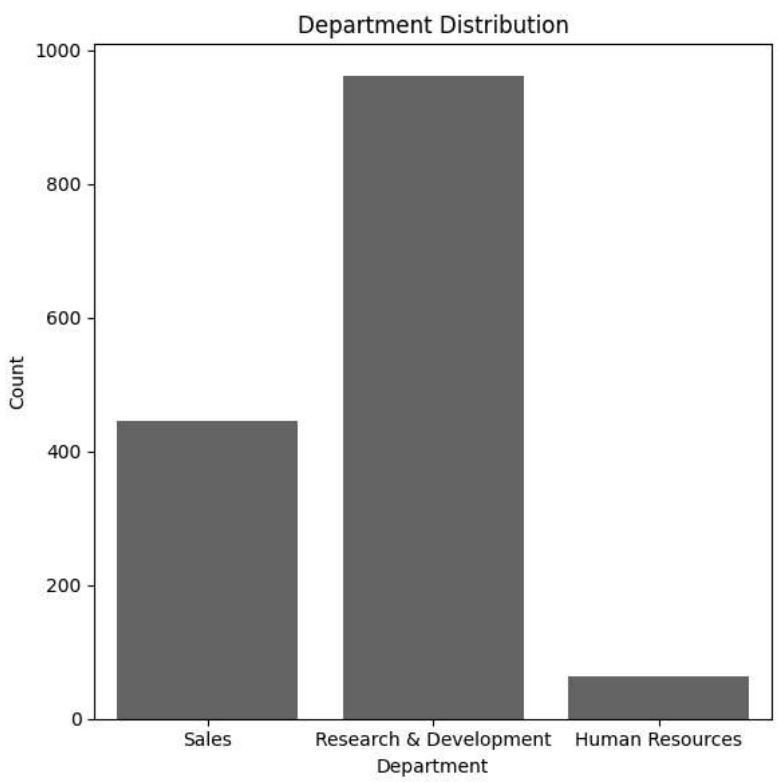
```
array(['Sales', 'Research & Development', 'Human Resources'], dtype=object)
```

```
df['Department'].value_counts()
```

Department	count
Research & Development	961
Sales	446
Human Resources	63

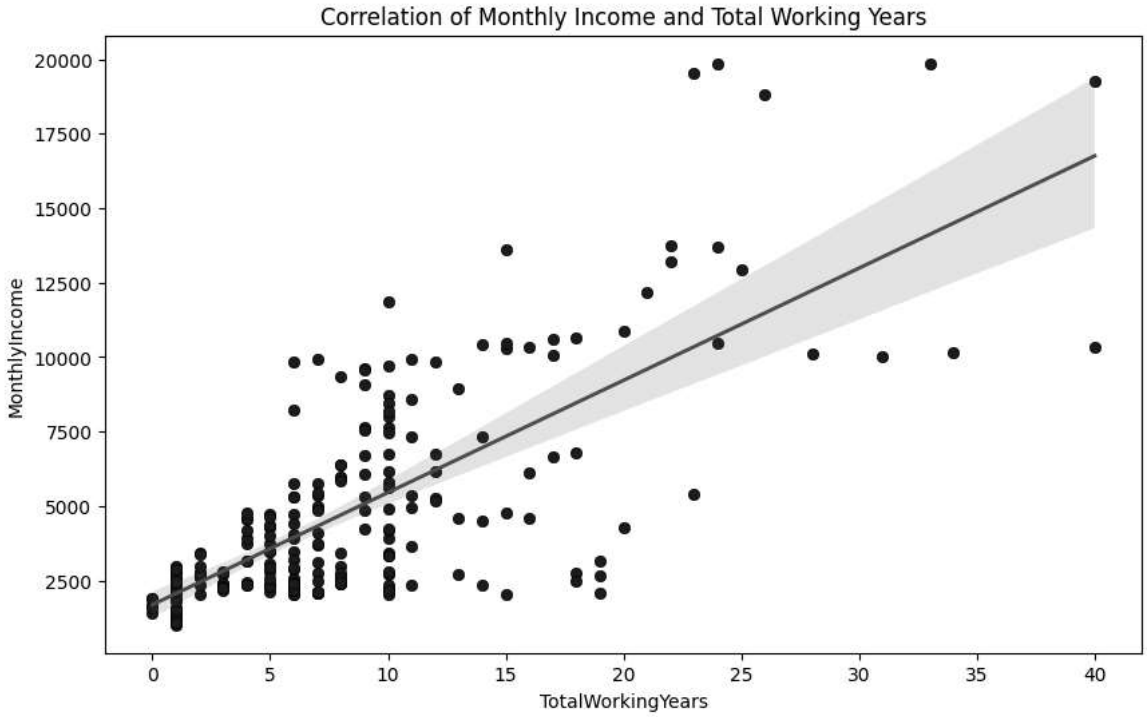
dtype: int64

```
plt.figure(figsize=(6,6))
sns.countplot(x='Department', data=df)
plt.title('Department Distribution')
plt.xlabel('Department')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```



The Research & Development department has the highest number of employees, followed by Sales and Human Resources.

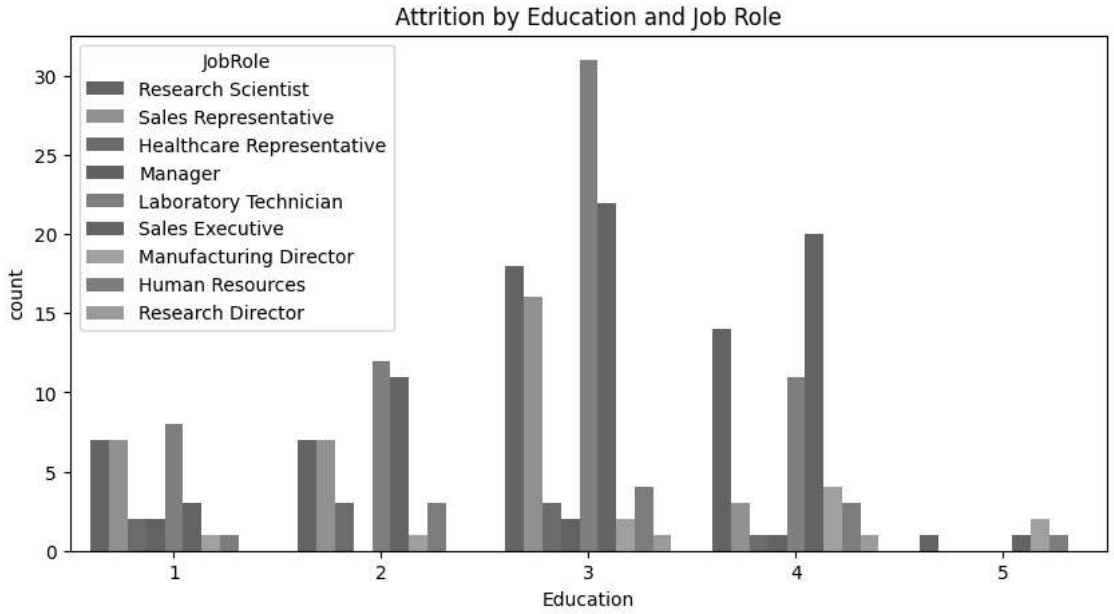
```
plt.figure(figsize=(10, 6))
plt.title("Correlation of Monthly Income and Total Working Years")
sns.scatterplot(x=df_attrition['TotalWorkingYears'], y=df_attrition['MonthlyIncome'], color='blue', edgecolor='black')
plt.xlabel('Total Working Years')
plt.ylabel('Monthly Income')
sns.regplot(x=df_attrition['TotalWorkingYears'], y=df_attrition['MonthlyIncome'], scatter=False, color='red', line_kws={'linewidth': 2})
plt.show()
```



There appears to be a positive correlation between TotalWorkingYears and MonthlyIncome for employees who have left the company. This suggests that employees with more experience tend to have higher monthly incomes. However, the relationship isn't perfectly linear, indicating that other factors might influence income.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Education and Job Role")
sns.countplot(x = 'Education', hue = 'JobRole', data = df_attrition)
```

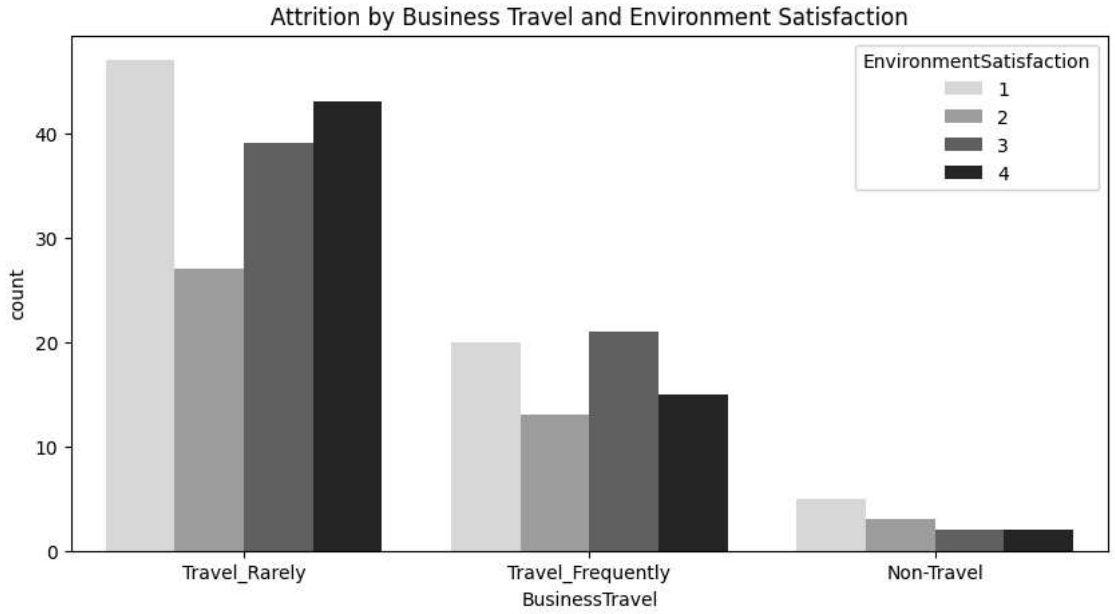
```
<Axes: title={'center': 'Attrition by Education and Job Role'}, xlabel='Education', ylabel='count'>
```



Laboratory Technicians are at the greatest risk of leaving, followed by Sales Executives.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Business Travel and Environment Satisfaction")
sns.countplot(x = 'BusinessTravel', hue = 'EnvironmentSatisfaction', data = df_attrition)
```

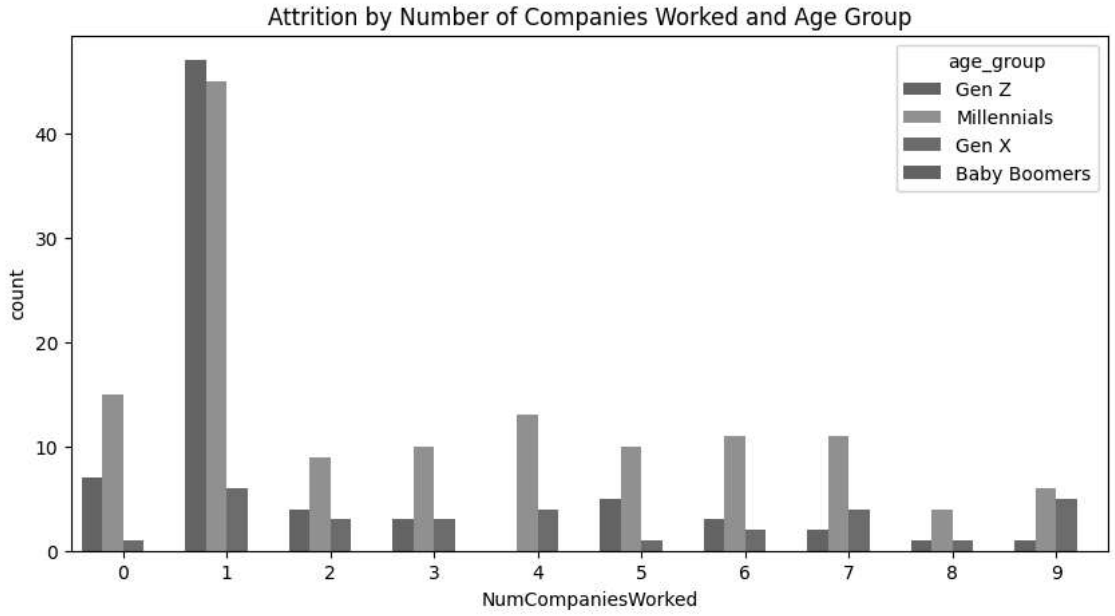
```
<Axes: title={'center': 'Attrition by Business Travel and Environment Satisfaction'}, xlabel='BusinessTravel', ylabel='count'>
```



Employees that travel rarely are less satified with work environment and account for the highest level of attrition.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Number of Companies Worked and Age Group")
sns.countplot(x = 'NumCompaniesWorked', hue = 'age_group', data = df_attrition)
```

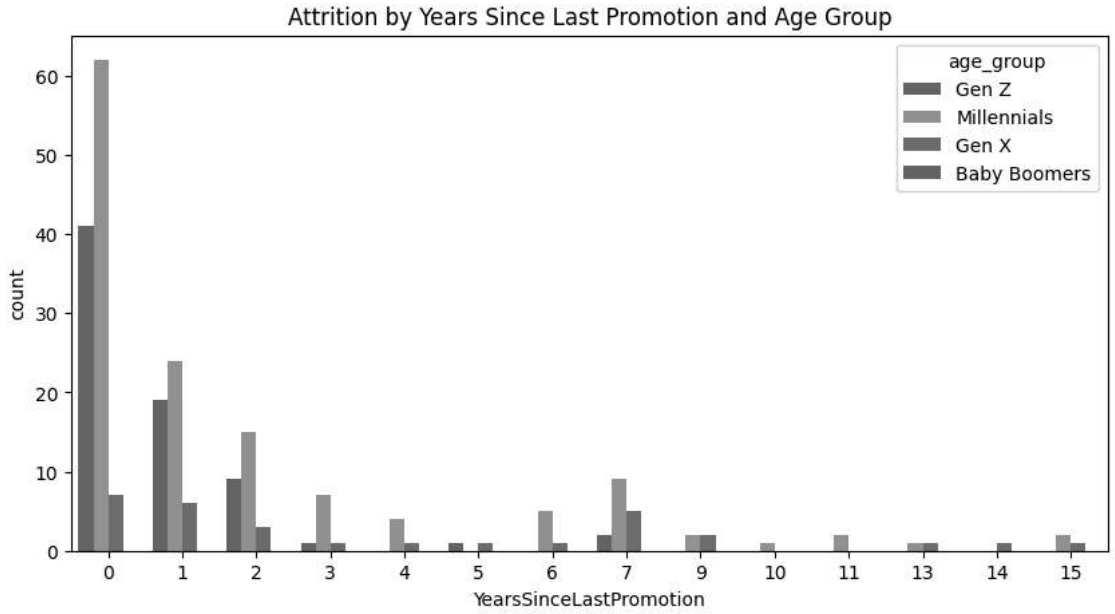
```
<Axes: title={'center': 'Attrition by Number of Companies Worked and Age Group'}, xlabel='NumCompaniesWorked', ylabel='count'>
```



Gen Z and Millennials are most likely to leave if they have only worked for 1 company prior to employment.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Years Since Last Promotion and Age Group")
sns.countplot(x = 'YearsSinceLastPromotion', hue = 'age_group', data = df_attrition)
```

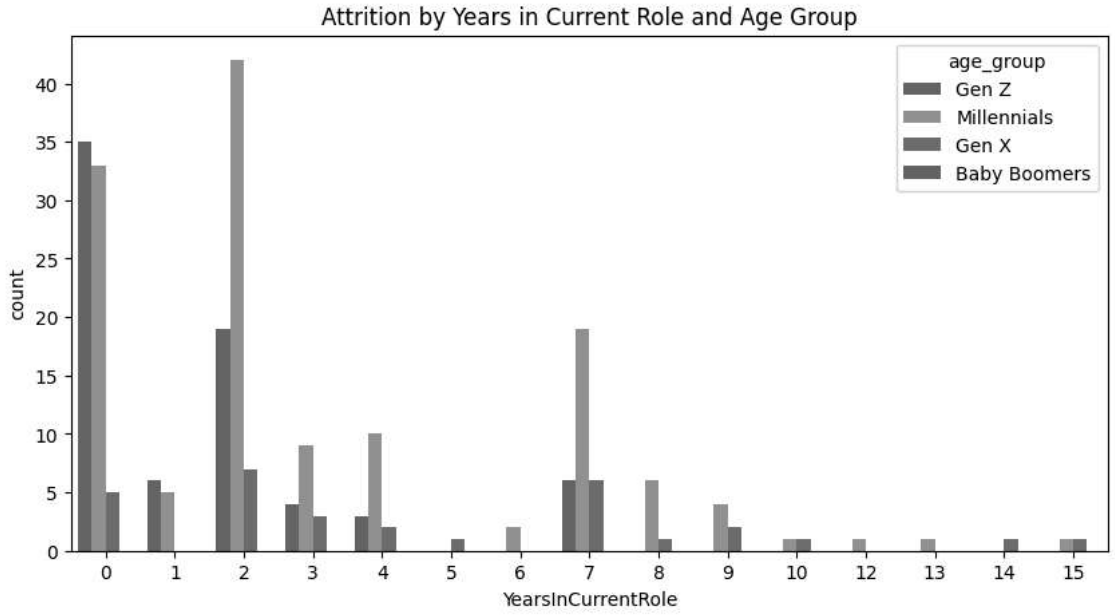
```
<Axes: title={'center': 'Attrition by Years Since Last Promotion and Age Group'}, xlabel='YearsSinceLastPromotion', ylabel='count'>
```



Millennials are most likely to leave within one year of not getting a promotion.

```
plt.figure(figsize = (10,5))
plt.title("Attrition by Years in Current Role and Age Group")
sns.countplot(x = 'YearsInCurrentRole', hue = 'age_group', data = df_attrition)
```

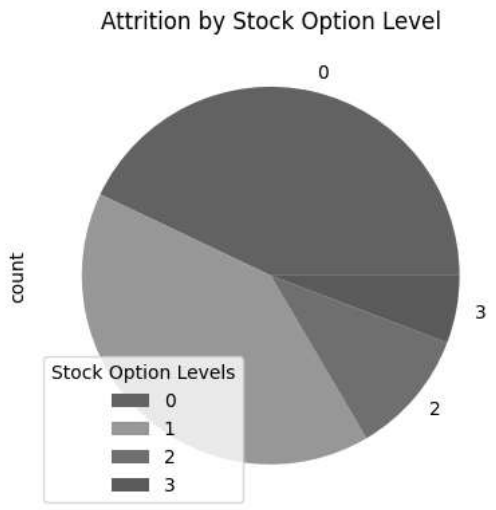
```
<Axes: title={'center': 'Attrition by Years in Current Role and Age Group'}, xlabel='YearsInCurrentRole', ylabel='count'>
```



Gen Z employees and Millennials are most likely to leave before completing an entire year in a job role. However, Millenials are most likely to leave within the second year of employment.

```
# Turnover by Stock Option Level
plt.figure(figsize = (10,10))
plt.subplot(1,2,1)
df['StockOptionLevel'].value_counts().plot(kind='pie', title = "Attrition by Stock Option Level")
plt.legend(title='Stock Option Levels', loc='best')
```

```
<matplotlib.legend.Legend at 0x7d6abff53790>
```



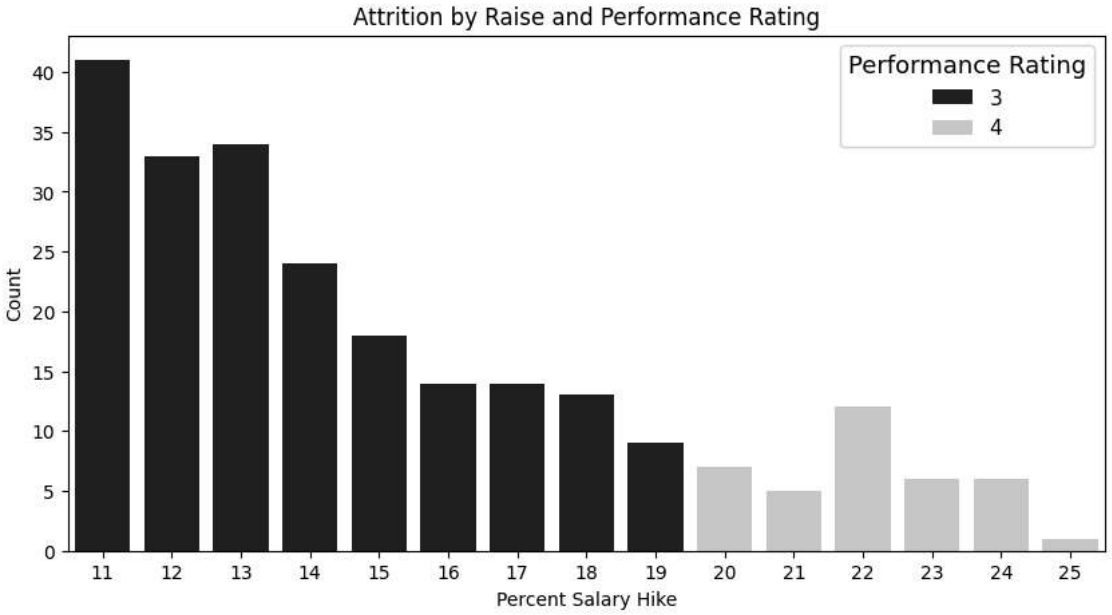
Employees with the highest stock option level are less likely to leave the company.

```
plt.figure(figsize=(10,12))
plt.subplot(1,2,2)
df_attrition['JobLevel'].value_counts().plot(kind='pie', title = "Attrition by Job Level")
plt.legend(title='Job Levels', loc='upper right')
```

```
<matplotlib.legend.Legend at 0x7d6abfeaffa0>
```

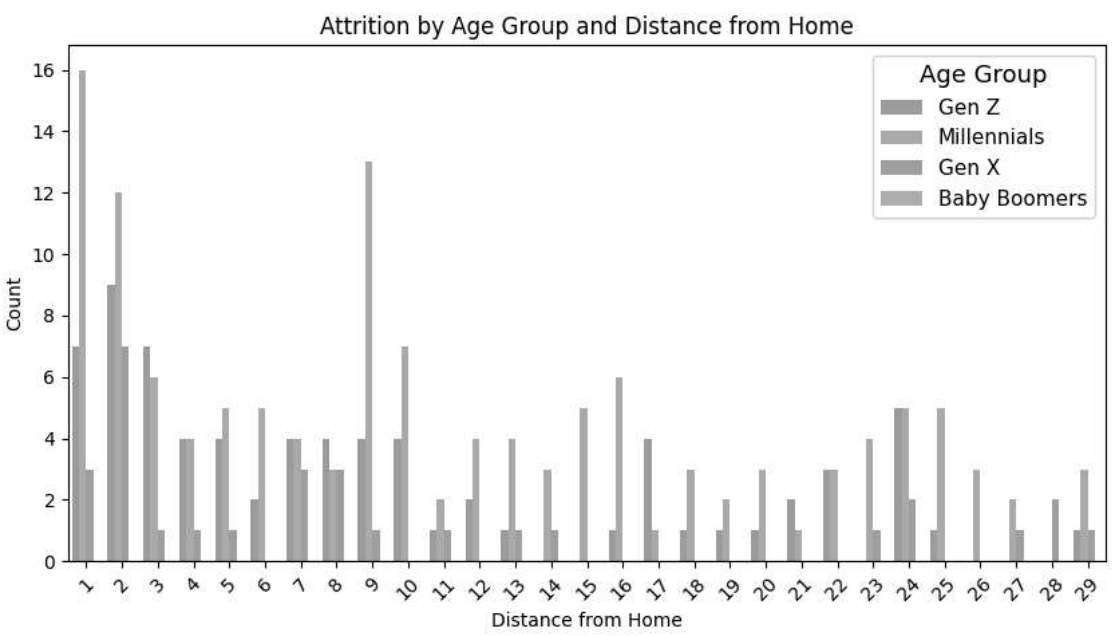
Additionally, employees with the highest job level are less likely to leave the company.

```
# Attrition by Raise and Performance Rating
plt.figure(figsize = (10,5))
plt.title("Attrition by Raise and Performance Rating")
sns.countplot(x='PercentSalaryHike', hue='PerformanceRating', data=df_attrition, palette='viridis')
plt.xlabel('Percent Salary Hike')
plt.ylabel('Count')
plt.legend(title='Performance Rating', title_fontsize='13', fontsize='11')
plt.show()
```



Employees receiving a Performance Rating of 3 are more likely to leave the company.

```
# Attrition by Age and Distance from Home
plt.figure(figsize=(10, 5))
sns.countplot(x='DistanceFromHome', hue='age_group', data=df_attrition, palette='Set2')
plt.title("Attrition by Age Group and Distance from Home")
plt.xlabel('Distance from Home')
plt.ylabel('Count')
plt.legend(title='Age Group', title_fontsize='13', fontsize='11')
plt.xticks(rotation=45)
plt.show()
```



Distance from home to work, when measured from 1 mile up to 29 miles, indicates Millenials are most likely to leave even if the distance from work to home is as low as 1 mile.

Model Building

```
print(df.columns)
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
```

```
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager', 'age_group'],
      dtype='object')
```

```
print(df.dtypes)
```

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

```
categorical_columns = ['Attrition', 'BusinessTravel', 'Department', 'Education', 'EducationField',
                       'Gender', 'JobRole', 'MaritalStatus', 'Over18', 'OverTime', 'age_group']

# encoding categorical features
df_encoded = pd.get_dummies(df, columns=categorical_columns, drop_first=True)
print(df_encoded.head())
```

```
   Age  DailyRate  DistanceFromHome  EmployeeCount  EmployeeNumber  \
0   41       1102                 1              1               1
1   49        279                 8              1               2
2   37       1373                 2              1               4
3   33       1392                 3              1               5
4   27        591                 2              1               7

   EnvironmentSatisfaction  HourlyRate  JobInvolvement  JobLevel  \
0                        2          94               3         2
1                        3          61               2         2
2                        4          92               2         1
3                        4          56               3         1
4                        1          40               3         1

   JobSatisfaction  ...  JobRole_Research Director  \
0                4  ...                      False
1                2  ...                      False
2                3  ...                      False
3                3  ...                      False
4                2  ...                      False

   JobRole_Research Scientist  JobRole_Sales Executive  \
0                       False                     True
1                        True                    False
2                       False                    False
3                        True                    False
4                       False                    False

   JobRole_Sales Representative  MaritalStatus_Married  MaritalStatus_Single  \
0                         False                  False                  True
1                         False                   True                 False
2                         False                  False                  True
3                         False                   True                 False
4                         False                   True                 False
```

```
   OverTime_Yes  age_group_Millennials  age_group_Gen X  \
0          True                   True            False
1         False                  False             True
2          True                   True            False
3          True                   True            False
4         False                  False            False

   age_group_Baby Boomers
0                   False
1                   False
2                   False
3                   False
4                   False

[5 rows x 54 columns]
```