# employee-data

August 23, 2023

[29]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import plotly.express as px
import statsmodels.api as sm
```

[5]:
```
data = pd.read_csv('C:\\Users\\96653\\Desktop\\Test jupyer\\Employee DATA.csv ')
```

[6]:
```
data.head()
```

[6]:

| | Age | Attrition | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationField | EmployeeCount | EmployeeNumber | … | RelationshipSatisfaction | StandardHours | StockOptionLevel | TotalWorkingYears | TrainingTimesLastYear | WorkLifeBalance | YearsAtCompany |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 41 | Yes | Travel_Rarely | 1102 | Sales | 1 | 2 | Life Sciences | 1 | 1 | … | 1 | 80 | 0 | 8 | 0 | 1 | 6 |
| 1 | 49 | No | Travel_Frequently | 279 | Research & Development | 8 | 1 | Life Sciences | 1 | 2 | … | 4 | 80 | 1 | 10 | 3 | 3 | 10 |
| 2 | 37 | Yes | Travel_Rarely | 1373 | Research & Development | 2 | 2 | Other | 1 | 4 | … | 2 | 80 | 0 | 7 | 3 | 3 | 0 |
| 3 | 33 | No | Travel_Frequently | 1392 | Research & Development | 3 | 4 | Life Sciences | 1 | 5 | … | 3 | 80 | 0 | 8 | 3 | 3 | 8 |
| 4 | 27 | No | Travel_Rarely | 591 | Research & Development | 2 | 1 | Medical | 1 | 7 | … | 4 | 80 | 1 | 6 | 3 | 3 | 2 |

```
3                   8                        3                3                8
4                   6                        3                3                2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

[7]:
```
data.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
```

[8]:
```
data.describe()
```

[8]:

| | Age | DailyRate | DistanceFromHome | Education | EmployeeCount | \ |
|---|---|---|---|---|---|---|
| count | 1470.000000 | 1470.000000 | 1470.000000 | 1470.000000 | 1470.0 | |
| mean | 36.923810 | 802.485714 | 9.192517 | 2.912925 | 1.0 | |
| std | 9.135373 | 403.509100 | 8.106864 | 1.024165 | 0.0 | |
| min | 18.000000 | 102.000000 | 1.000000 | 1.000000 | 1.0 | |
| 25% | 30.000000 | 465.000000 | 2.000000 | 2.000000 | 1.0 | |
| 50% | 36.000000 | 802.000000 | 7.000000 | 3.000000 | 1.0 | |
| 75% | 43.000000 | 1157.000000 | 14.000000 | 4.000000 | 1.0 | |
| max | 60.000000 | 1499.000000 | 29.000000 | 5.000000 | 1.0 | |

| | EmployeeNumber | EnvironmentSatisfaction | HourlyRate | JobInvolvement | \ |
|---|---|---|---|---|---|
| count | 1470.000000 | 1470.000000 | 1470.000000 | 1470.000000 | |
| mean | 1024.865306 | 2.721769 | 65.891156 | 2.729932 | |
| std | 602.024335 | 1.093082 | 20.329428 | 0.711561 | |
| min | 1.000000 | 1.000000 | 30.000000 | 1.000000 | |
| 25% | 491.250000 | 2.000000 | 48.000000 | 2.000000 | |
| 50% | 1020.500000 | 3.000000 | 66.000000 | 3.000000 | |
| 75% | 1555.750000 | 4.000000 | 83.750000 | 3.000000 | |
| max | 2068.000000 | 4.000000 | 100.000000 | 4.000000 | |

| | JobLevel | … | RelationshipSatisfaction | StandardHours | \ |
|---|---|---|---|---|---|
| count | 1470.000000 | … | 1470.000000 | 1470.0 | |
| mean | 2.063946 | … | 2.712245 | 80.0 | |
| std | 1.106940 | … | 1.081209 | 0.0 | |
| min | 1.000000 | … | 1.000000 | 80.0 | |
| 25% | 1.000000 | … | 2.000000 | 80.0 | |
| 50% | 2.000000 | … | 3.000000 | 80.0 | |
| 75% | 3.000000 | … | 4.000000 | 80.0 | |
| max | 5.000000 | … | 4.000000 | 80.0 | |

| | StockOptionLevel | TotalWorkingYears | TrainingTimesLastYear | \ |
|---|---|---|---|---|
| count | 1470.000000 | 1470.000000 | 1470.000000 | |
| mean | 0.793878 | 11.279592 | 2.799320 | |
| std | 0.852077 | 7.780782 | 1.289271 | |
| min | 0.000000 | 0.000000 | 0.000000 | |

```
25%            0.000000           6.000000              2.000000
50%            1.000000          10.000000              3.000000
75%            1.000000          15.000000              3.000000
max            3.000000          40.000000              6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 26 columns]
```

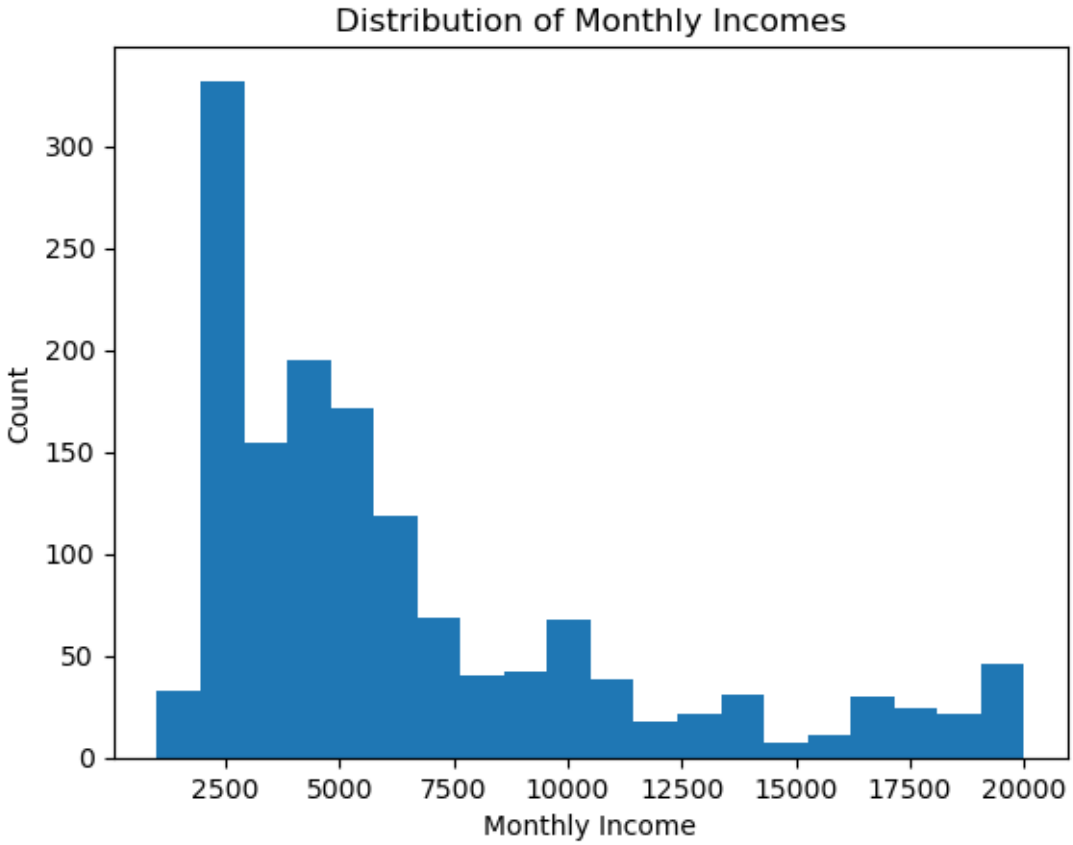
[9]:
```
attrition_rate = data['Attrition'].value_counts() / len(data) * 100
print(attrition_rate)
```

```
No     83.877551
Yes    16.122449
Name: Attrition, dtype: float64
```

[10]:
```
job_satisfaction_by_role = data.groupby('JobRole')['JobSatisfaction'].mean()
print(job_satisfaction_by_role)
```

```
JobRole
Healthcare Representative    2.786260
Human Resources              2.557692
Laboratory Technician        2.691120
Manager                      2.705882
Manufacturing Director       2.682759
Research Director            2.700000
Research Scientist           2.773973
Sales Executive              2.754601
Sales Representative         2.734940
Name: JobSatisfaction, dtype: float64
```

[11]:
```
plt.hist(data['MonthlyIncome'], bins=20)
plt.xlabel('Monthly Income')
plt.ylabel('Count')
plt.title('Distribution of Monthly Incomes')
plt.show()
```



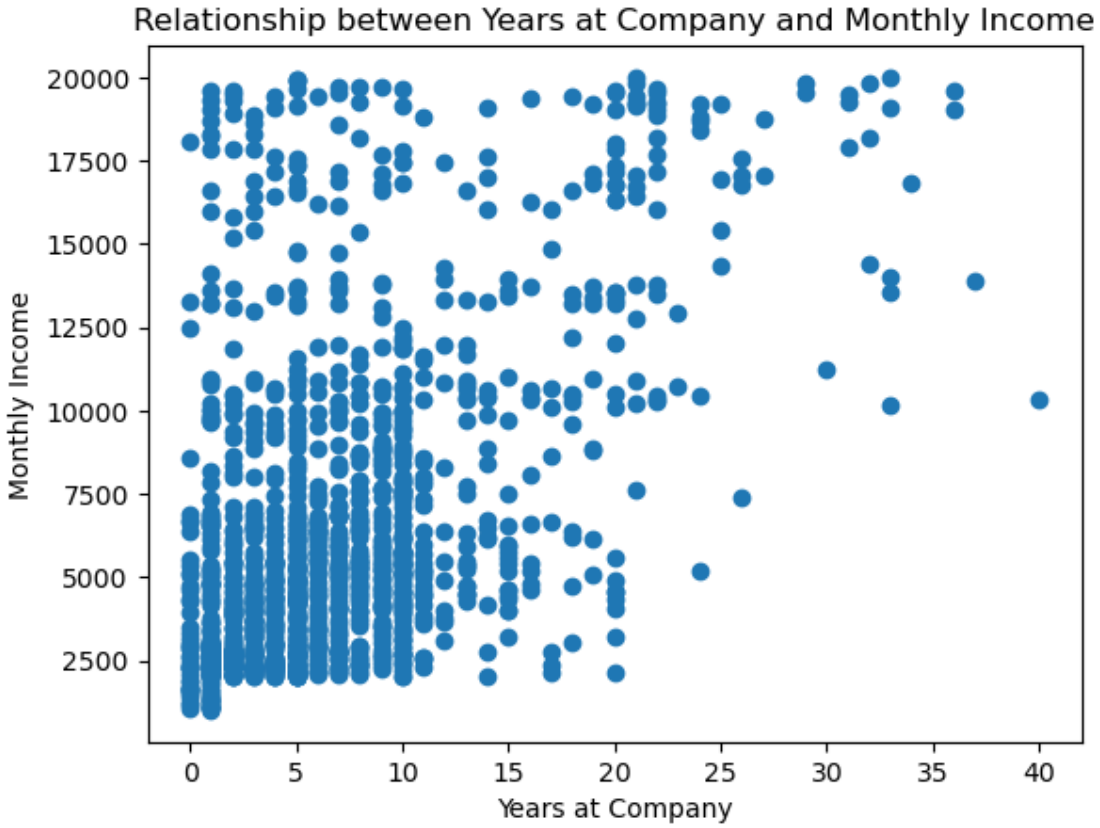

[12]:
```
average_age = data['Age'].mean()
print(average_age)
```

```
36.923809523809524
```

[13]:
```
job_levels = data['JobLevel'].value_counts()
print(job_levels)
```

```
1    543
2    534
3    218
4    106
5     69
Name: JobLevel, dtype: int64
```

[15]:
```
plt.scatter(data['YearsAtCompany'], data['MonthlyIncome'])
plt.xlabel('Years at Company')
plt.ylabel('Monthly Income')
plt.title('Relationship between Years at Company and Monthly Income')
plt.show()
```

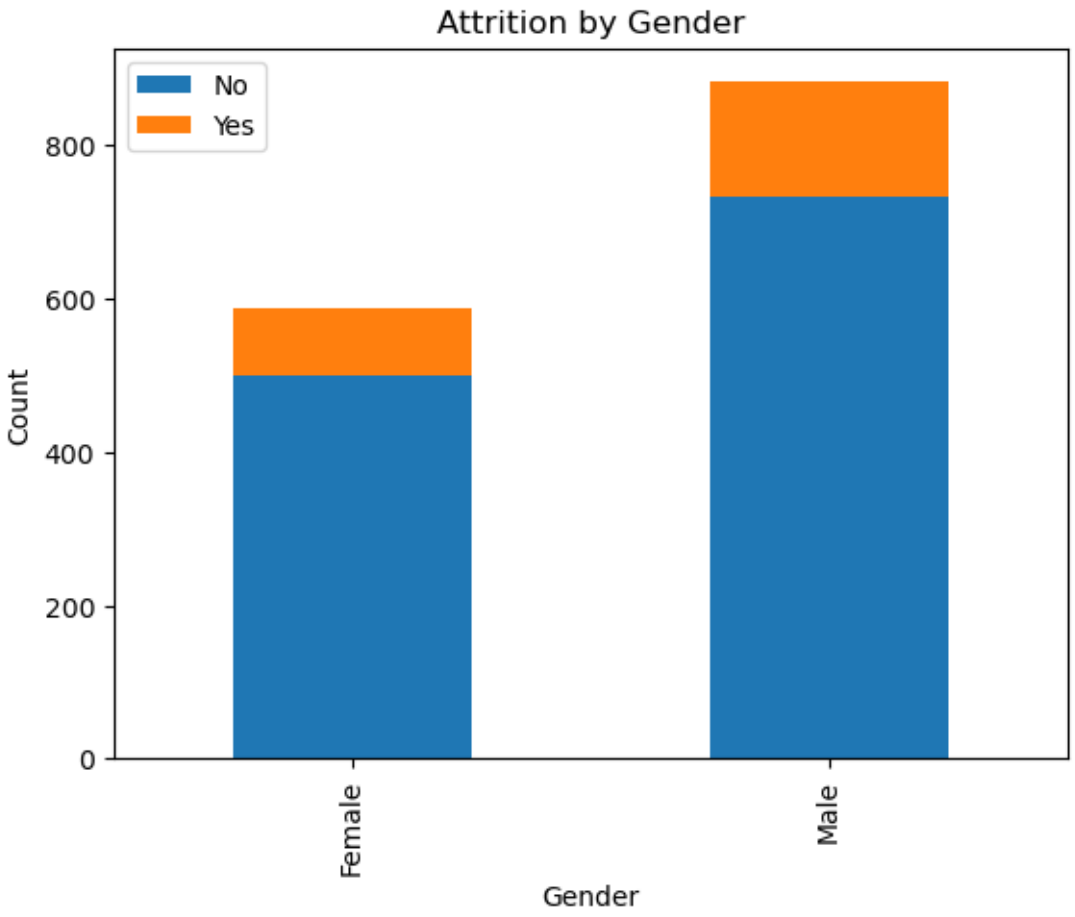



[16]:
```
department_income = data.groupby('Department')['MonthlyIncome'].mean()
print(department_income)
```

```
Department
Human Resources           6654.507937
Research & Development    6281.252862
Sales                     6959.172646
Name: MonthlyIncome, dtype: float64
```

[17]:
```
attrition_by_gender = data.groupby('Gender')['Attrition'].value_counts().
  ↪unstack()
attrition_by_gender.plot(kind='bar', stacked=True)
plt.xlabel('Gender')
plt.ylabel('Count')
```

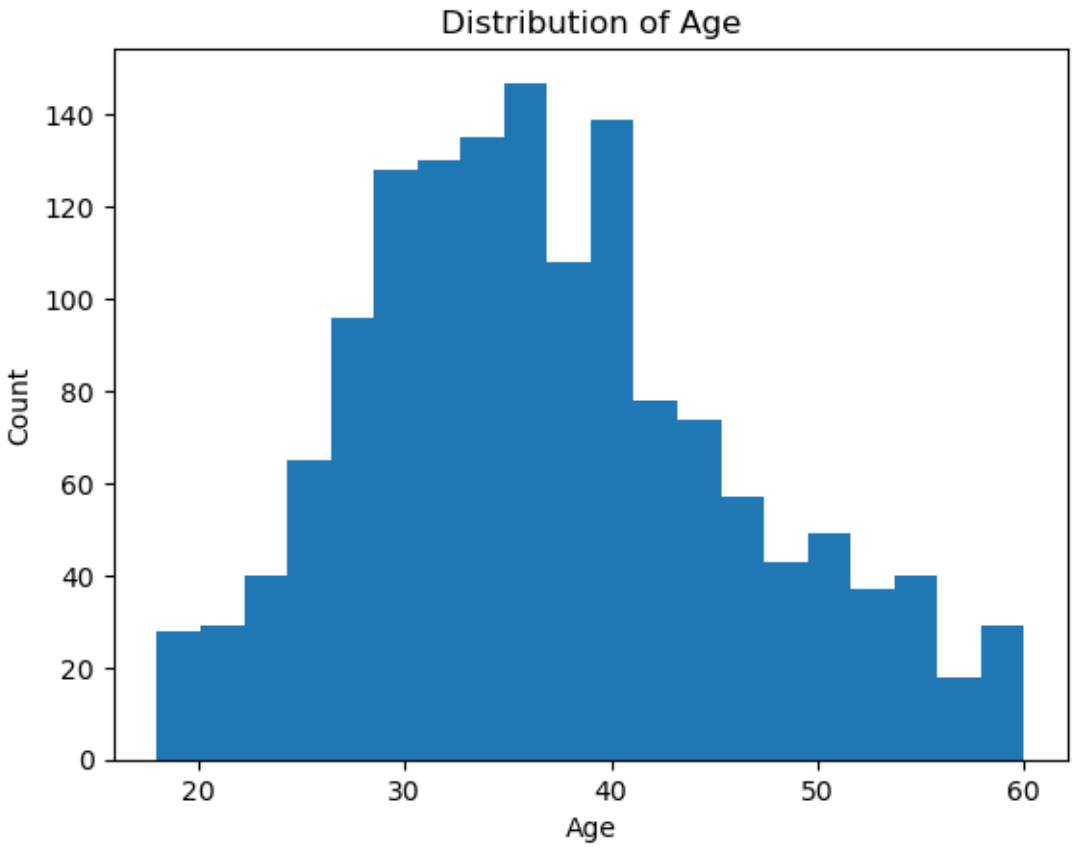
```
plt.title('Attrition by Gender')
plt.legend()
plt.show()
```



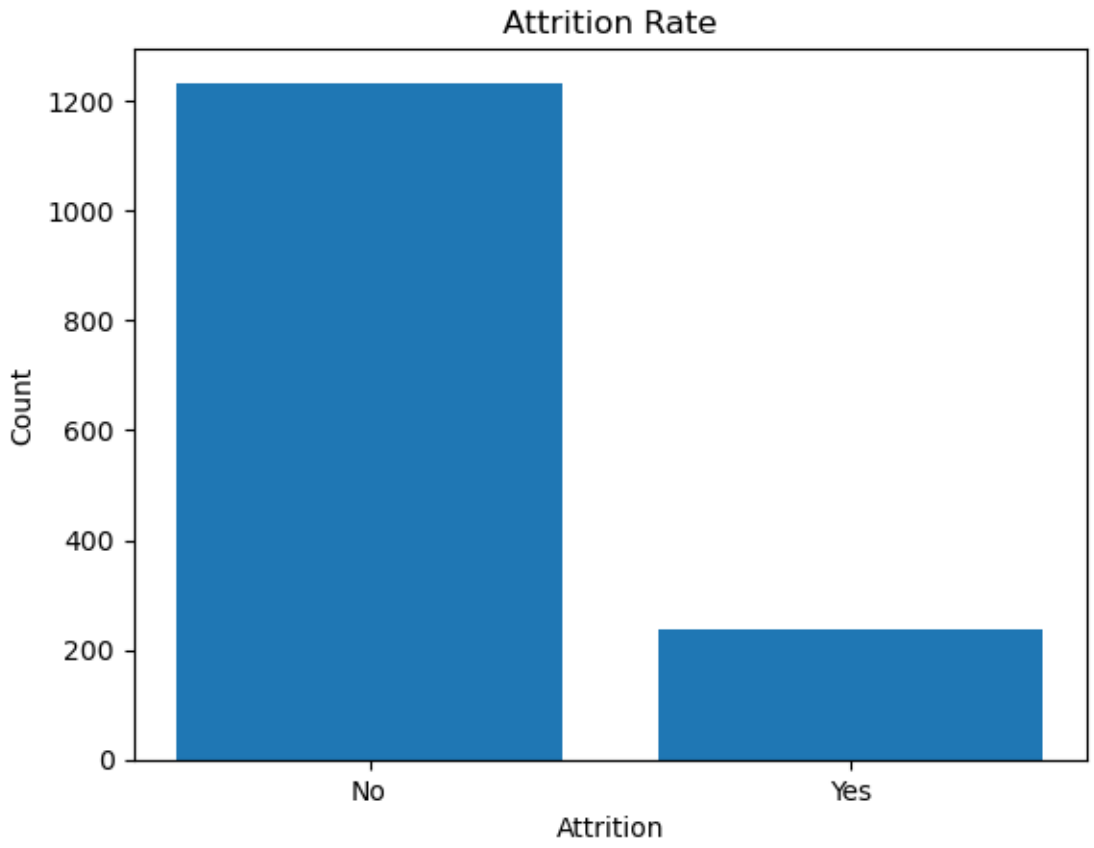

[18]:
```
distance_home_by_role = data.groupby('JobRole')['DistanceFromHome'].mean()
print(distance_home_by_role)
```

```
JobRole
Healthcare Representative    9.786260
Human Resources              8.173077
Laboratory Technician        9.409266
Manager                      8.029412
Manufacturing Director       9.482759
Research Director            8.437500
Research Scientist           9.013699
Sales Executive              9.659509
Sales Representative         8.662651
Name: DistanceFromHome, dtype: float64
```

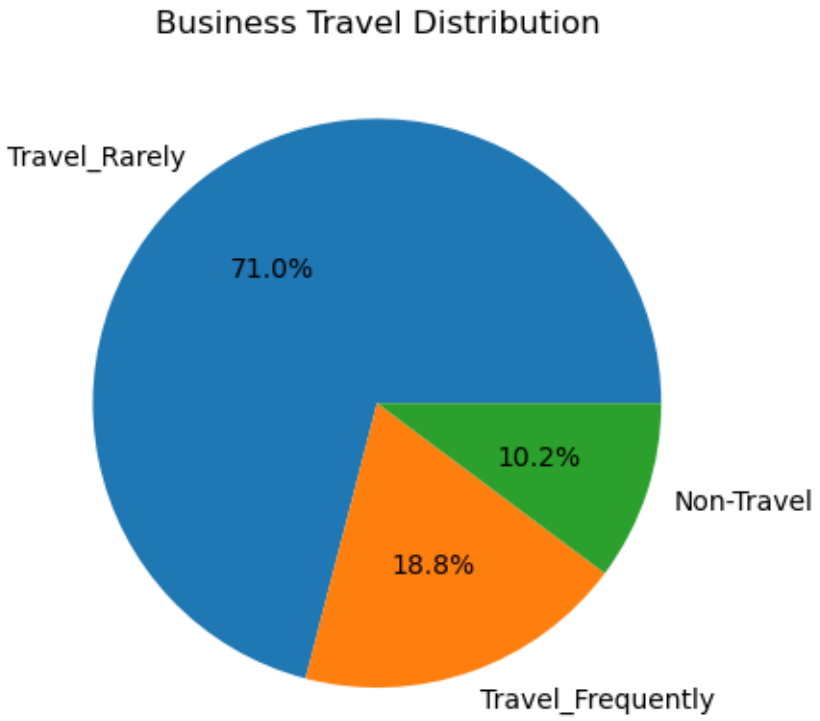
[19]:
```
plt.hist(data['Age'], bins=20)
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Distribution of Age')
plt.show()
```



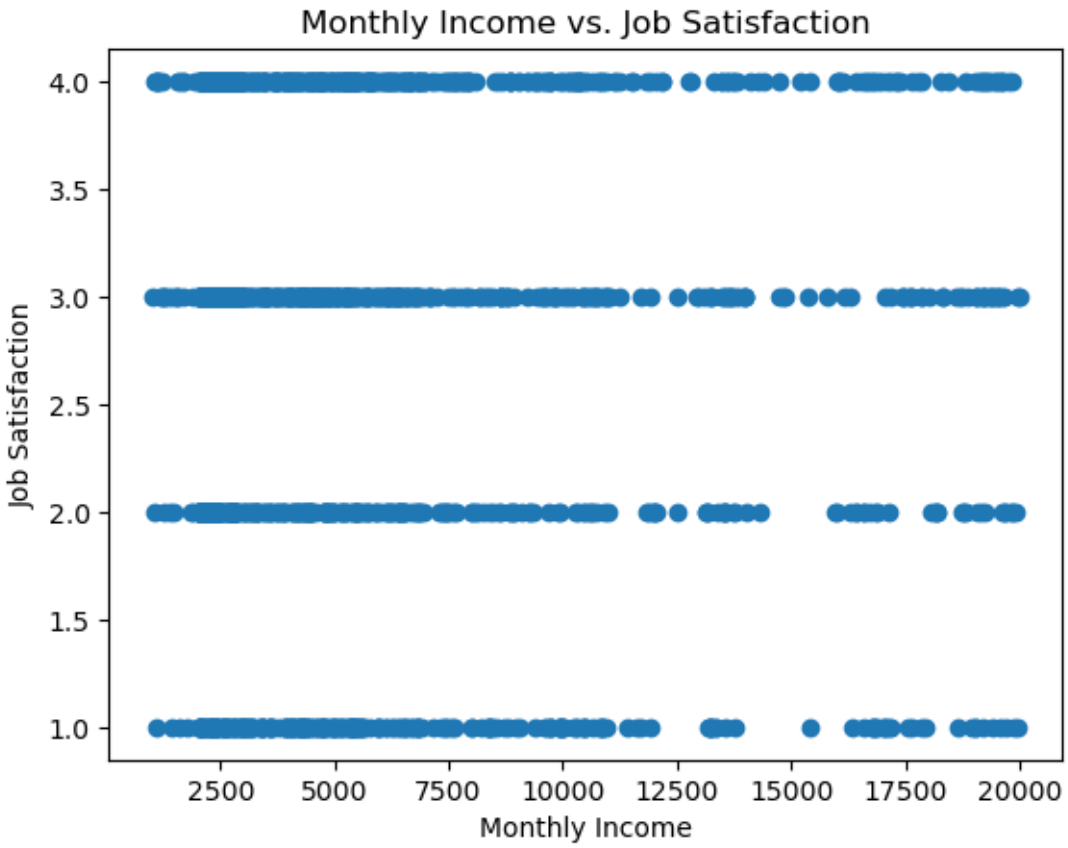

[20]:
```
attrition_counts = data['Attrition'].value_counts()
plt.bar(attrition_counts.index, attrition_counts.values)
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.title('Attrition Rate')
plt.show()
```



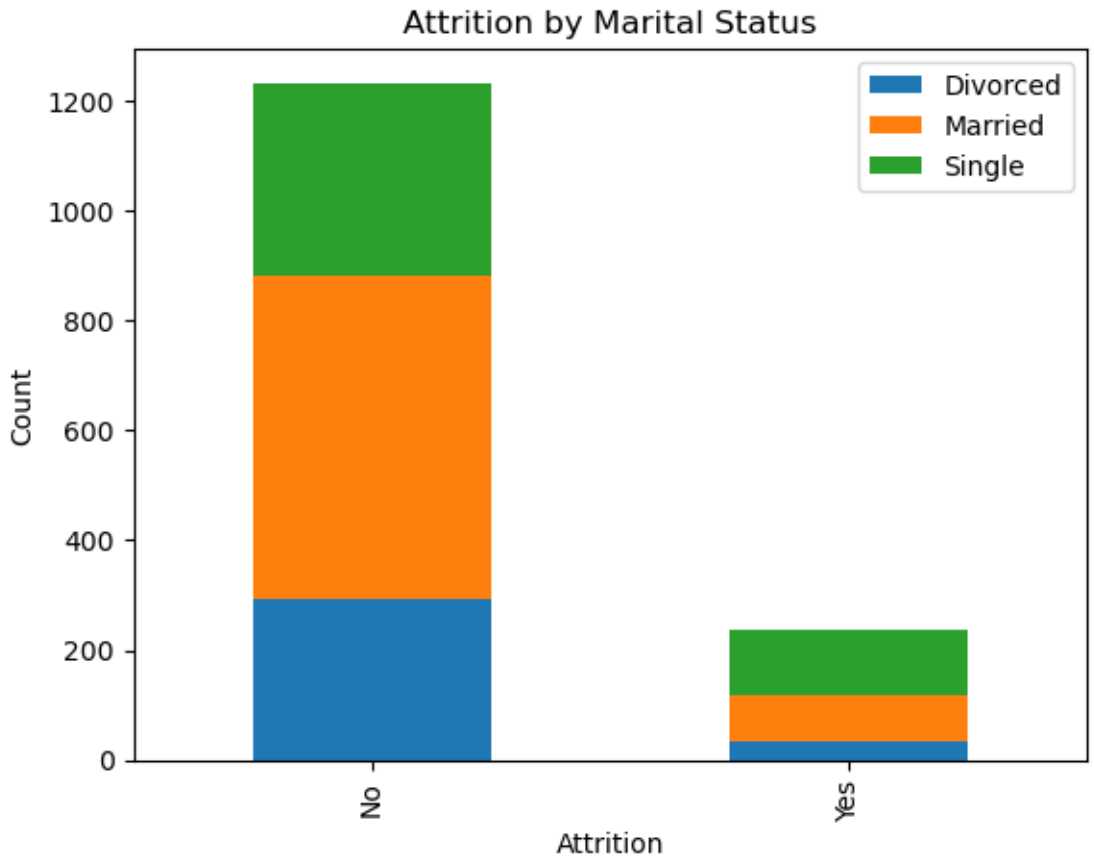

[21]:
```
business_travel_counts = data['BusinessTravel'].value_counts()
plt.pie(business_travel_counts, labels=business_travel_counts.index,␣
 ↪autopct='%1.1f%%')
plt.title('Business Travel Distribution')
plt.show()
```



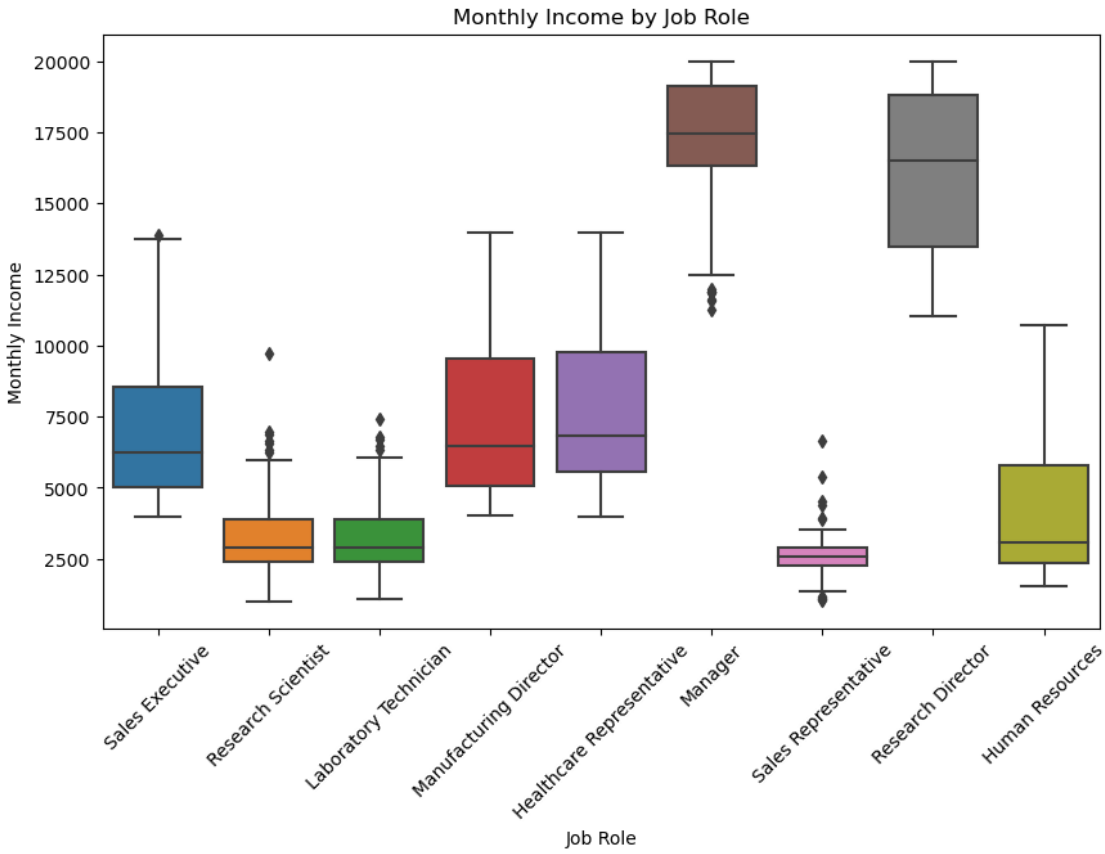

[22]:
```
plt.scatter(data['MonthlyIncome'], data['JobSatisfaction'])
plt.xlabel('Monthly Income')
plt.ylabel('Job Satisfaction')
plt.title('Monthly Income vs. Job Satisfaction')
plt.show()
```




[23]:
```
attrition_by_marital = data.groupby(['Attrition', 'MaritalStatus']).size().
  ↪unstack()
attrition_by_marital.plot(kind='bar', stacked=True)
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.title('Attrition by Marital Status')
plt.legend()
plt.show()
```

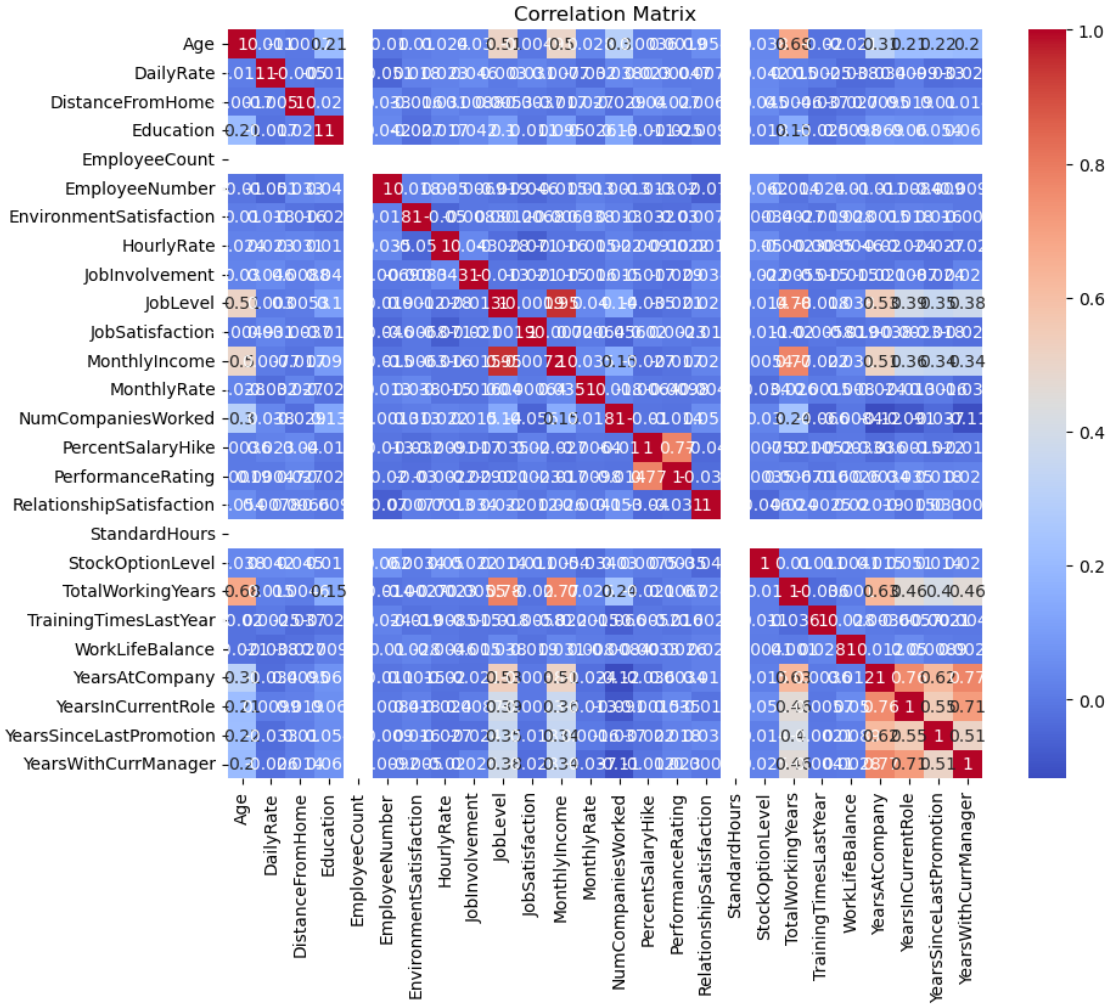



[25]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='JobRole', y='MonthlyIncome', data=data)
plt.xlabel('Job Role')
plt.ylabel('Monthly Income')
plt.title('Monthly Income by Job Role')
plt.xticks(rotation=45)
plt.show()
```




[27]:
```
numeric_cols = data.select_dtypes(include=np.number).columns
correlation = data[numeric_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```




[30]:
```
fig = px.violin(data, x='EducationField', y='JobSatisfaction', title='Job␣
↪Satisfaction by Education Field')
fig.show()
```

[31]:
```
import statsmodels.api as sm

# Define the predictor and response variables
X = data['MonthlyIncome']
y = data['JobSatisfaction']

# Add constant term to predictor variable
X = sm.add_constant(X)

# Fit the linear regression model
model = sm.OLS(y, X)
results = model.fit()
```

```
# Print the model summary
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        JobSatisfaction   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.001
Method:                 Least Squares   F-statistic:                    0.07519
Date:                Wed, 23 Aug 2023   Prob (F-statistic):              0.784
Time:                        09:05:33   Log-Likelihood:                -2229.2
No. Observations:                1470   AIC:                             4462.
Df Residuals:                    1468   BIC:                             4473.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-
const             2.7395      0.049     55.820      0.000       2.643
2.836
MonthlyIncome -1.676e-06   6.11e-06     -0.274      0.784   -1.37e-05
1.03e-05
==============================================================================
Omnibus:                     1411.411   Durbin-Watson:                   2.022
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              118.029
Skew:                          -0.329   Prob(JB):                     2.35e-26
Kurtosis:                       1.778   Cond. No.                     1.37e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.37e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[ ]:

[ ]:

[ ]:

[ ]: