HR_MeriSkill_P2_final

October 5, 2023

1 Employee Attrition Analysis

Importing Data:

[1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

[2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
from pathlib import Path
from plotly.offline import iplot,init_notebook_mode,plot
# from plotly.subplots import make_subplots
init_notebook_mode(connected=True)


%matplotlib inline
pd.set_option('display.max_columns', None)
```

[3]:
```
hr = pd.read_csv("D:\DataScience\Internship\MeriSkill\Project-2\Project 3 - HR␣
↪Analytics\Clean Data\HR-Employee-Attrition.csv")
hr.head()
```

[3]:
```
   Age Attrition  DailyRate              Department  DistanceFromHome  \
0   41       Yes       1102                   Sales                 1
1   49        No        279  Research & Development                 8
2   37       Yes       1373  Research & Development                 2
3   33        No       1392  Research & Development                 3
4   27        No        591  Research & Development                 2

   Education EducationField  EmployeeCount  EmployeeNumber  \
0          2  Life Sciences              1               1
1          1  Life Sciences              1               2
2          2          Other              1               4
3          4  Life Sciences              1               5
4          1        Medical              1               7

   EnvironmentSatisfaction  Gender  HourlyRate  JobInvolvement  JobLevel  \
```

```
0                         2  Female         94              3         2
1                         3    Male         61              2         2
2                         4    Male         92              2         1
3                         4  Female         56              3         1
4                         1    Male         40              3         1

                 JobRole  JobSatisfaction MaritalStatus  MonthlyIncome  \
0        Sales Executive                4        Single           5993
1     Research Scientist                2       Married           5130
2  Laboratory Technician                3        Single           2090
3     Research Scientist                3       Married           2909
4  Laboratory Technician                2       Married           3468

   MonthlyRate  NumCompaniesWorked Over18 OverTime  PercentSalaryHike  \
0        19479                   8      Y      Yes                 11
1        24907                   1      Y       No                 23
2         2396                   6      Y      Yes                 15
3        23159                   1      Y      Yes                 11
4        16632                   9      Y       No                 12

   PerformanceRating  RelationshipSatisfaction  StandardHours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   YearsSinceLastPromotion  YearsWithCurrManager BusinessTravel
0                        0                     5         Rarely
1                        1                     7     Frequently
2                        0                     0         Rarely
3                        3                     0     Frequently
4                        2                     2         Rarely
```

Data Informations:

```
[4]: hr.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   DailyRate                 1470 non-null   int64
 3   Department                1470 non-null   object
 4   DistanceFromHome          1470 non-null   int64
 5   Education                 1470 non-null   int64
 6   EducationField            1470 non-null   object
 7   EmployeeCount             1470 non-null   int64
 8   EmployeeNumber            1470 non-null   int64
 9   EnvironmentSatisfaction   1470 non-null   int64
 10  Gender                    1470 non-null   object
 11  HourlyRate                1470 non-null   int64
 12  JobInvolvement            1470 non-null   int64
 13  JobLevel                  1470 non-null   int64
 14  JobRole                   1470 non-null   object
 15  JobSatisfaction           1470 non-null   int64
 16  MaritalStatus             1470 non-null   object
 17  MonthlyIncome             1470 non-null   int64
 18  MonthlyRate               1470 non-null   int64
 19  NumCompaniesWorked        1470 non-null   int64
 20  Over18                    1470 non-null   object
 21  OverTime                  1470 non-null   object
 22  PercentSalaryHike         1470 non-null   int64
 23  PerformanceRating         1470 non-null   int64
 24  RelationshipSatisfaction  1470 non-null   int64
 25  StandardHours             1470 non-null   int64
 26  StockOptionLevel          1470 non-null   int64
 27  TotalWorkingYears         1470 non-null   int64
 28  TrainingTimesLastYear     1470 non-null   int64
 29  WorkLifeBalance           1470 non-null   int64
 30  YearsAtCompany            1470 non-null   int64
 31  YearsInCurrentRole        1470 non-null   int64
 32  YearsSinceLastPromotion   1470 non-null   int64
 33  YearsWithCurrManager      1470 non-null   int64
 34  BusinessTravel            1470 non-null   object
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

```
[5]: hr.shape
```

[5]: (1470, 35)

[6]:
```
hr.columns
```

[6]:
```
Index(['Age', 'Attrition', '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', 'BusinessTravel'],
      dtype='object')
```

[7]:
```
hr["Department"].unique()
```

[7]:
```
array(['Sales', 'Research & Development', 'Human Resources'], dtype=object)
```

[8]:
```
hr["EducationField"].unique()
```

[8]:
```
array(['Life Sciences', 'Other', 'Medical', 'Marketing',
       'Technical Degree', 'Human Resources'], dtype=object)
```

[9]:
```
hr["Gender"].unique()
```

[9]:
```
array(['Female', 'Male'], dtype=object)
```

[10]:
```
hr["JobRole"].unique()
```

[10]:
```
array(['Sales Executive', 'Research Scientist', 'Laboratory Technician',
       'Manufacturing Director', 'Healthcare Representative', 'Manager',
       'Sales Representative', 'Research Director', 'Human Resources'],
      dtype=object)
```

[11]:
```
hr["MaritalStatus"].unique()
```

[11]:
```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

[12]:
```
hr["Over18"].unique()
```

[12]:
```
array(['Y'], dtype=object)
```

[13]:
```
hr["OverTime"].unique()
```

[13]:
```
array(['Yes', 'No'], dtype=object)
```

[14]:
```
hr["BusinessTravel"].unique()
```

[14]:
```
array(['Rarely', 'Frequently', 'Non-Travel'], dtype=object)
```

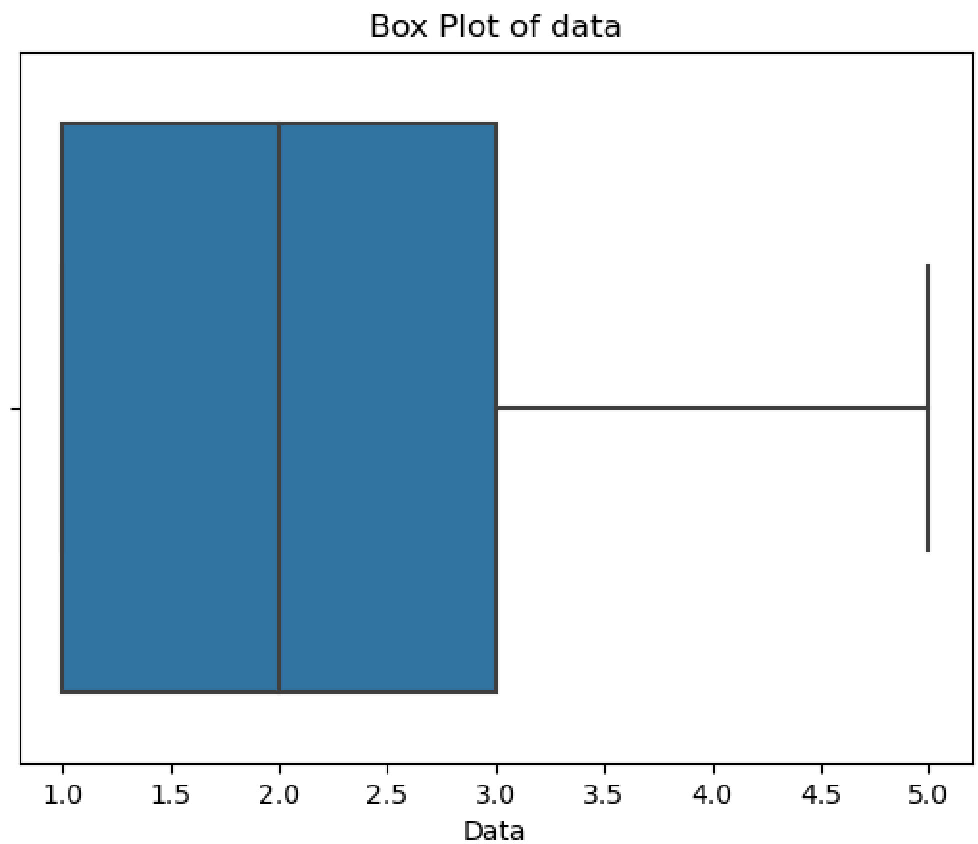
[15]:
```
hr.describe()
```

[15]:
```
               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  JobSatisfaction  MonthlyIncome   MonthlyRate  \
count  1470.000000      1470.000000    1470.000000   1470.000000
mean      2.063946         2.728571    6502.931293  14313.103401
std       1.106940         1.102846    4707.956783   7117.786044
min       1.000000         1.000000    1009.000000   2094.000000
25%       1.000000         2.000000    2911.000000   8047.000000
50%       2.000000         3.000000    4919.000000  14235.500000
75%       3.000000         4.000000    8379.000000  20461.500000
max       5.000000         4.000000   19999.000000  26999.000000

       NumCompaniesWorked  PercentSalaryHike  PerformanceRating  \
count         1470.000000        1470.000000        1470.000000
mean             2.693197          15.209524           3.153741
std              2.498009           3.659938           0.360824
min              0.000000          11.000000           3.000000
25%              1.000000          12.000000           3.000000
50%              2.000000          14.000000           3.000000
75%              4.000000          18.000000           3.000000
max              9.000000          25.000000           4.000000

       RelationshipSatisfaction  StandardHours  StockOptionLevel  \
count               1470.000000         1470.0       1470.000000
mean                   2.712245           80.0          0.793878
```

```
std                      1.081209           0.0          0.852077
min                      1.000000          80.0          0.000000
25%                      2.000000          80.0          0.000000
50%                      3.000000          80.0          1.000000
75%                      4.000000          80.0          1.000000
max                      4.000000          80.0          3.000000

       TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
count        1470.000000            1470.000000      1470.000000
mean           11.279592               2.799320         2.761224
std             7.780782               1.289271         0.706476
min             0.000000               0.000000         1.000000
25%             6.000000               2.000000         2.000000
50%            10.000000               3.000000         3.000000
75%            15.000000               3.000000         3.000000
max            40.000000               6.000000         4.000000

       YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
count     1470.000000         1470.000000              1470.000000
mean         7.008163            4.229252                 2.187755
std          6.126525            3.623137                 3.222430
min          0.000000            0.000000                 0.000000
25%          3.000000            2.000000                 0.000000
50%          5.000000            3.000000                 1.000000
75%          9.000000            7.000000                 3.000000
max         40.000000           18.000000                15.000000

       YearsWithCurrManager
count           1470.000000
mean               4.123129
std                3.568136
min                0.000000
25%                2.000000
50%                3.000000
75%                7.000000
max               17.000000
```

[16]:
```python
# Create the box plot
sns.boxplot(x=hr["JobLevel"])

# Set the title and labels
plt.title("Box Plot of data")
plt.xlabel("Data")
# plt.xlim(0, 1000)
```

[16]: Text(0.5, 0, 'Data')



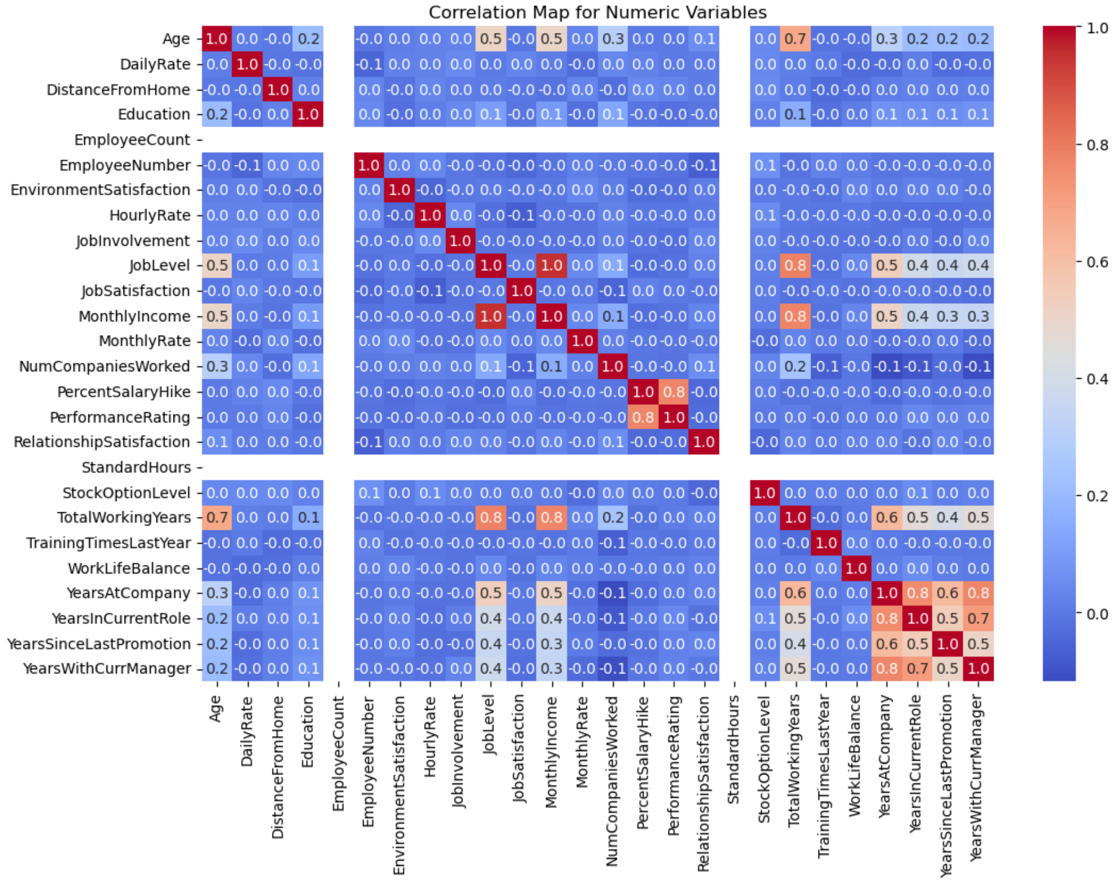
EDA:

Correlation Map for Numeric Variables:

[17]:
```
correlation_matrix = hr.corr()
```

[18]:
```
filepath = Path('D:/DataScience/Internship/MeriSkill/Project-2/Project 3 - HR␣
↪Analytics/Docs/correlation_matrix.csv')
filepath.parent.mkdir(parents=True, exist_ok=True)
correlation_matrix.to_csv(filepath)
```

[19]:
```
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".1f")
plt.title("Correlation Map for Numeric Variables")
plt.show()
```

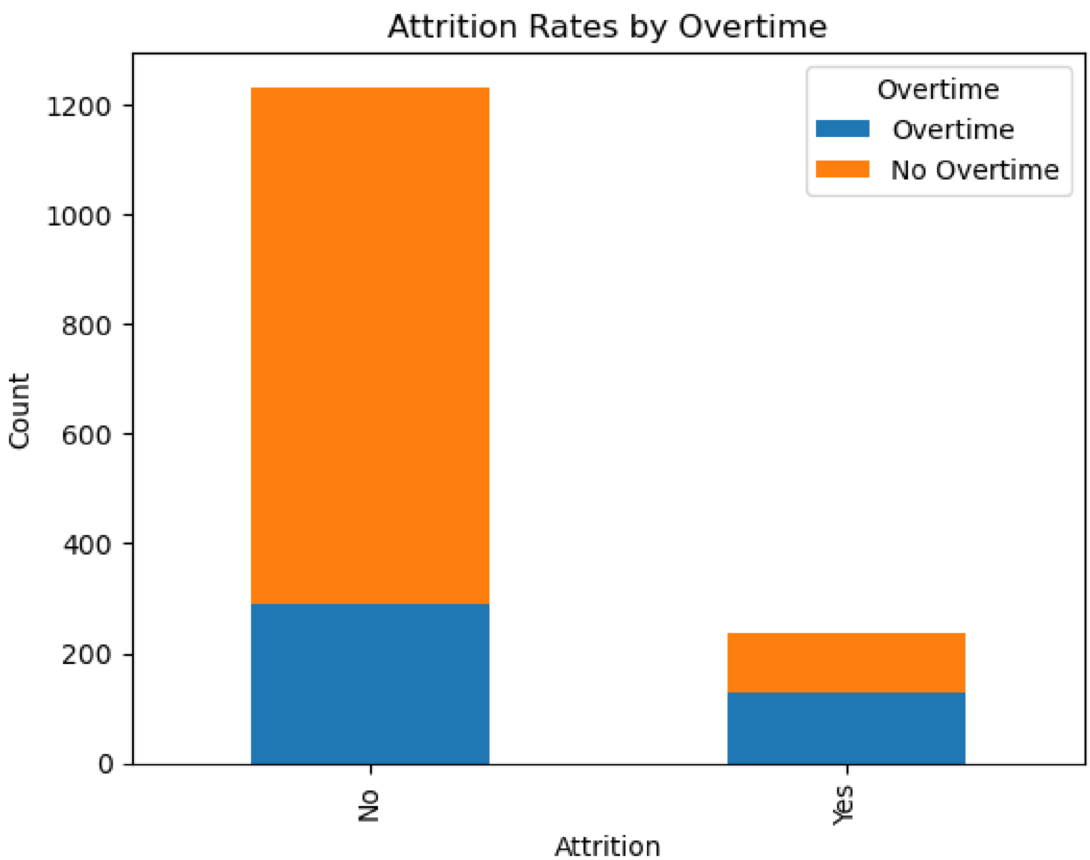


Overtime Status:

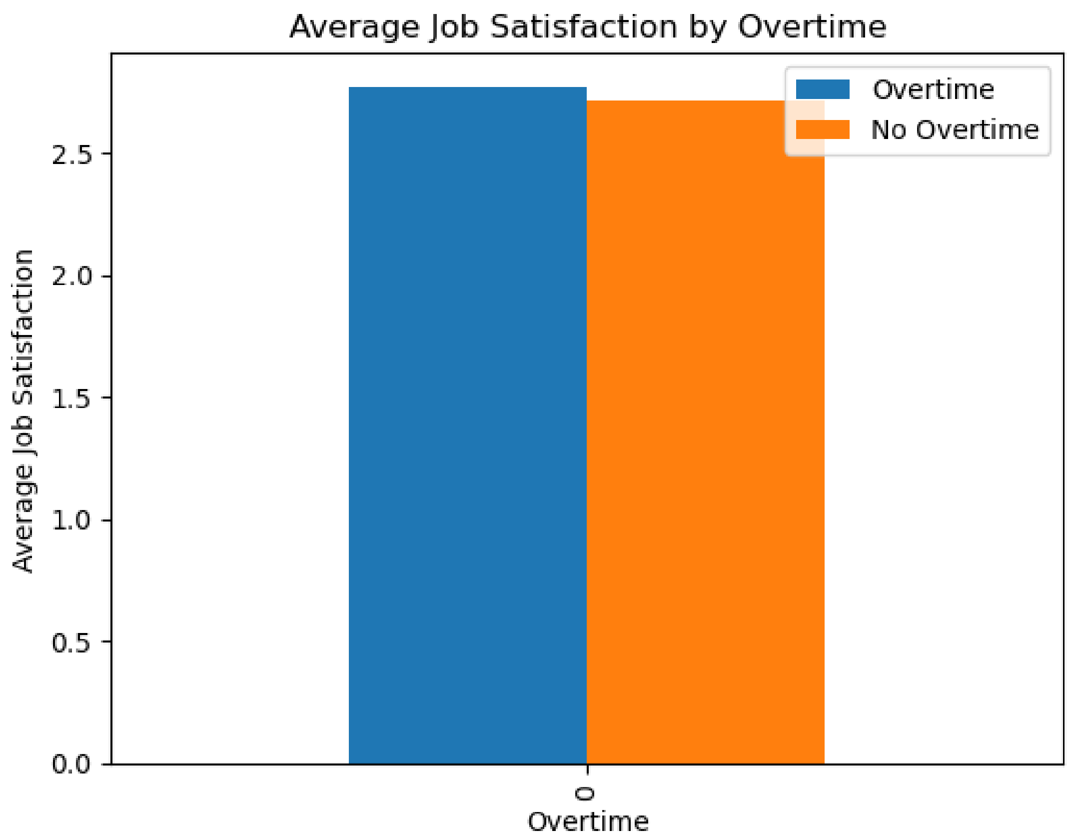
Attrition

Job satisfaction

[21]:
```python
# Calculate attrition rates for overtime and non-overtime employees
attrition_overtime = hr[hr['OverTime'] == 'Yes']['Attrition'].value_counts()
attrition_no_overtime = hr[hr['OverTime'] == 'No']['Attrition'].value_counts()

# Create a bar chart to visualize attrition rates
attrition_rates = pd.DataFrame({'Overtime': attrition_overtime, 'No Overtime': attrition_no_overtime})
attrition_rates.plot(kind='bar', stacked=True)
plt.title('Attrition Rates by Overtime')
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.legend(title='Overtime')
plt.show()
```



[23]:
```
# Calculate average job satisfaction for overtime and non-overtime employees
avg_job_satisfaction_overtime = hr[hr['OverTime'] == 'Yes']['JobSatisfaction'].
    ↪mean()
avg_job_satisfaction_no_overtime = hr[hr['OverTime'] ==␣
    ↪'No']['JobSatisfaction'].mean()

# Create a bar chart to visualize average job satisfaction
avg_job_satisfaction = pd.DataFrame({'Overtime': avg_job_satisfaction_overtime,␣
    ↪'No Overtime': avg_job_satisfaction_no_overtime}, index=[0])
avg_job_satisfaction.plot(kind='bar')
plt.title('Average Job Satisfaction by Overtime')
plt.ylabel('Average Job Satisfaction')
plt.xlabel('Overtime')
plt.show()
```



Marital Status:

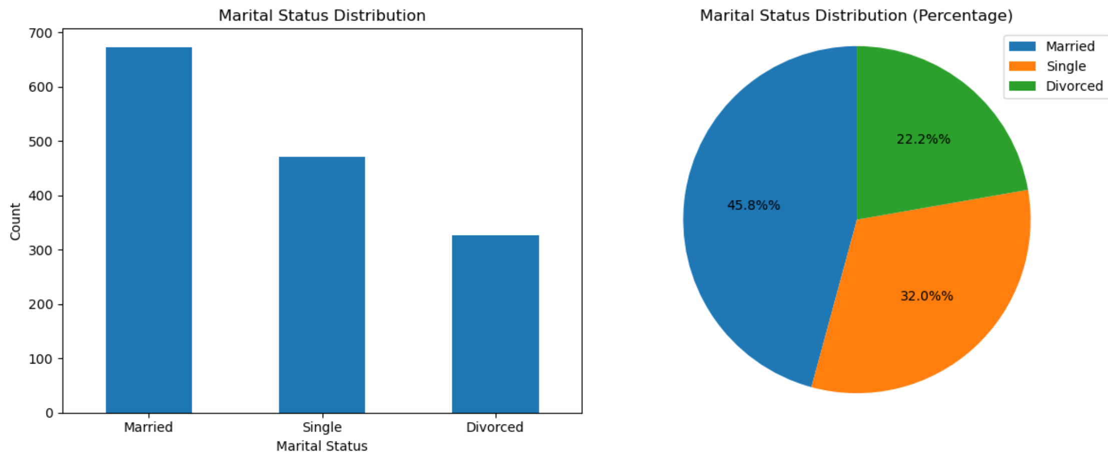
% of Employees

Attrition

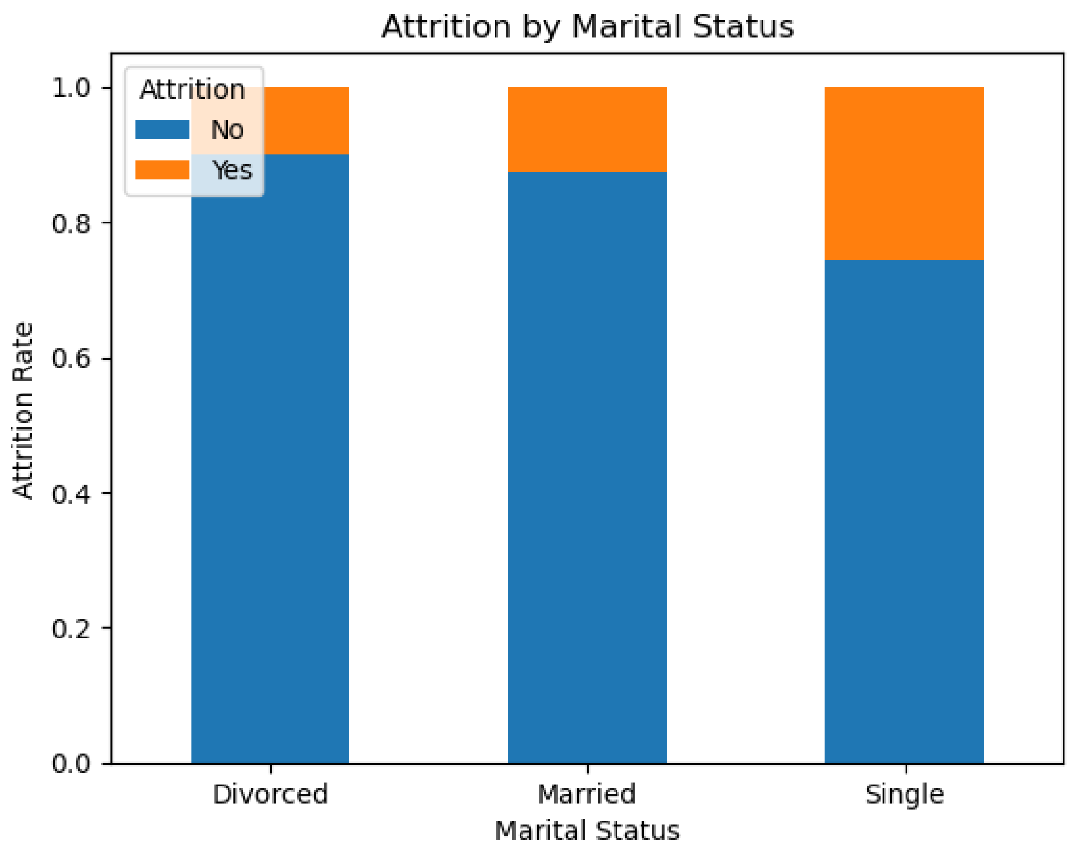
Average Monthly

[24]:
```
# Calculate the count and percentage of employees in each marital status␣
↪category
marital_status_counts = hr['MaritalStatus'].value_counts()
marital_status_percentage = hr['MaritalStatus'].value_counts(normalize=True) *␣
↪100

# Create bar plots to visualize the distribution
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
marital_status_counts.plot(kind='bar', rot=0)
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Count')
```

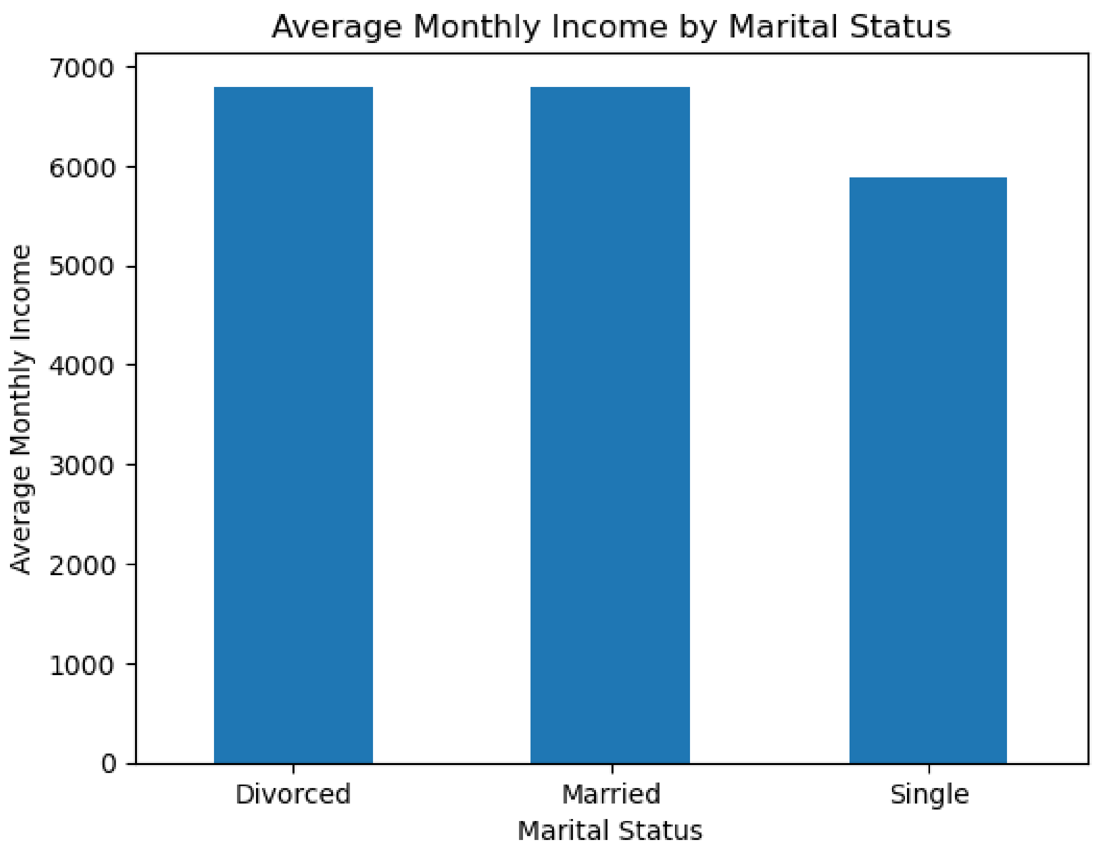
```
# Create a subplot for the pie chart
plt.subplot(1, 2, 2)
# Plot a pie chart
wedges, _, autotexts = plt.pie(marital_status_percentage, autopct='%1.1f%%',␣
↪startangle=90)
# Add legends
plt.legend(marital_status_percentage.index, loc='best')
# Set title and aspect ratio
plt.title('Marital Status Distribution (Percentage)')
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a␣
↪circle
# Display the pie chart
plt.tight_layout()
# Show only percentage values and add '%' symbol
for autotext in autotexts:
    autotext.set_text(f"{autotext.get_text()}%")

plt.show()
```



[26]:
```
# Calculate attrition rates for each marital status
attrition_by_marital_status = hr.groupby('MaritalStatus')['Attrition'].
↪value_counts(normalize=True).unstack()
attrition_by_marital_status.plot(kind='bar', stacked=True, rot=0)
plt.title('Attrition by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Attrition Rate')
plt.legend(title='Attrition', loc='upper left')
plt.show()
```



[28]:
```
# Calculate average monthly income by marital status
average_income_by_marital_status = hr.groupby('MaritalStatus')['MonthlyIncome'].
    ↪mean()
average_income_by_marital_status.plot(kind='bar', rot=0)
plt.title('Average Monthly Income by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Average Monthly Income')
plt.show()
```



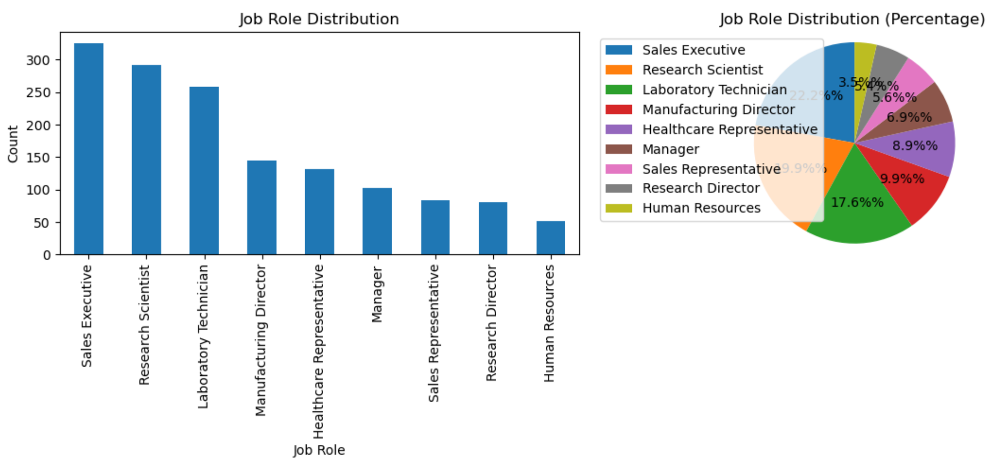
Job Role Status:

% of Employees

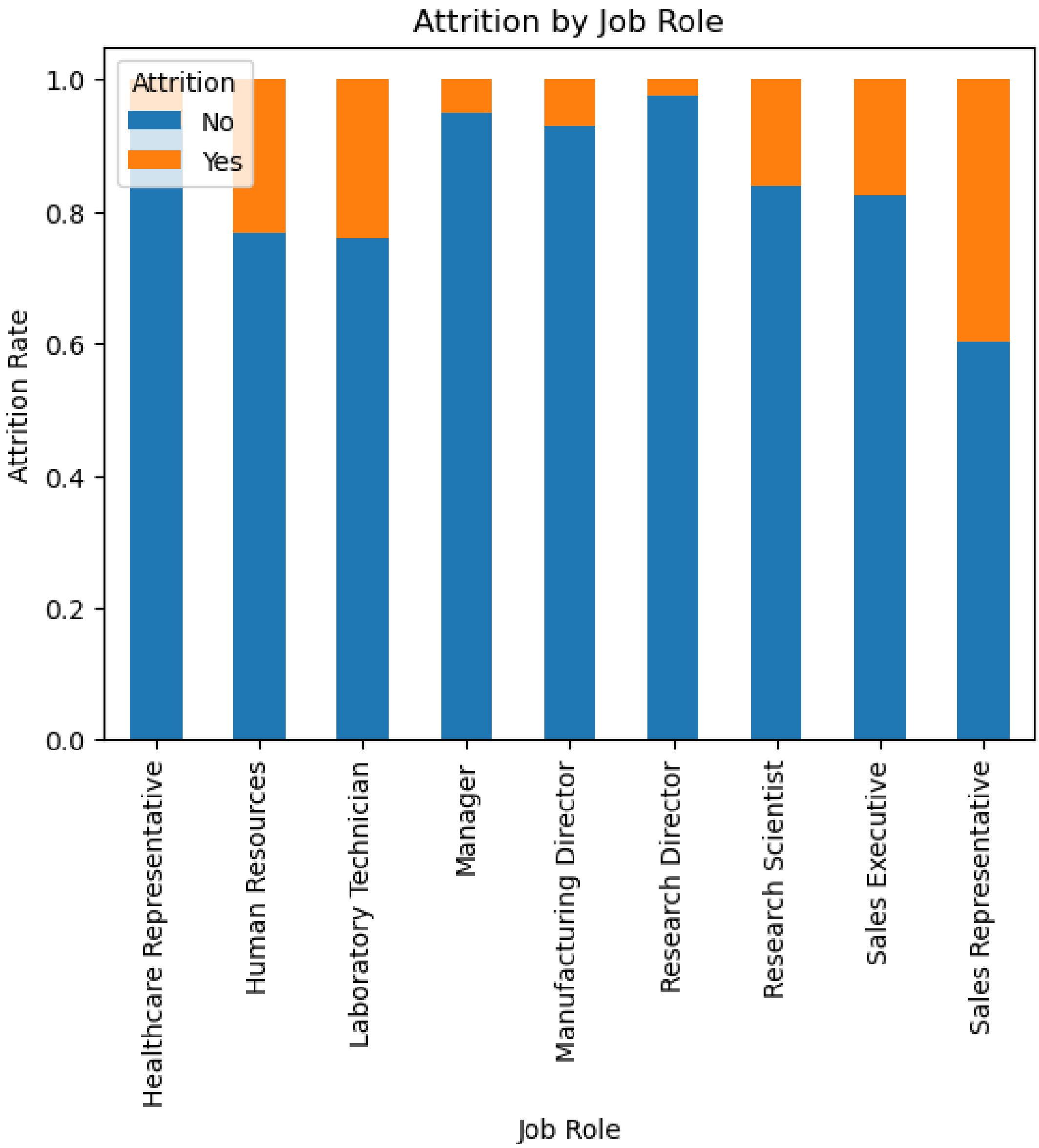
Attrition

[30]:
```
# Calculate the count and percentage of employees in each job role category
job_role_counts = hr['JobRole'].value_counts()
job_role_percentage = hr['JobRole'].value_counts(normalize=True) * 100

# Create bar plots to visualize the distribution
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
job_role_counts.plot(kind='bar', rot=90)
plt.title('Job Role Distribution')
plt.xlabel('Job Role')
plt.ylabel('Count')

plt.subplot(1, 2, 2)
# Plot a pie chart with legends
```

```
wedges, _, autotexts = plt.pie(job_role_percentage, autopct='%1.1f%%',
    startangle=90)
# Add legends
plt.legend(job_role_percentage.index, loc='best')
# Set title and aspect ratio
plt.title('Job Role Distribution (Percentage)')
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a
    circle
# Display the pie chart
plt.tight_layout()
# Show only percentage values and add '%' symbol
for autotext in autotexts:
    autotext.set_text(f"{autotext.get_text()}%")

plt.show()
```



[32]:
```
# Calculate attrition rates for each job role
attrition_by_job_role = hr.groupby('JobRole')['Attrition'].
    value_counts(normalize=True).unstack()
attrition_by_job_role.plot(kind='bar', stacked=True, rot=90)
plt.title('Attrition by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Attrition Rate')
plt.legend(title='Attrition', loc='upper left')
plt.show()
```



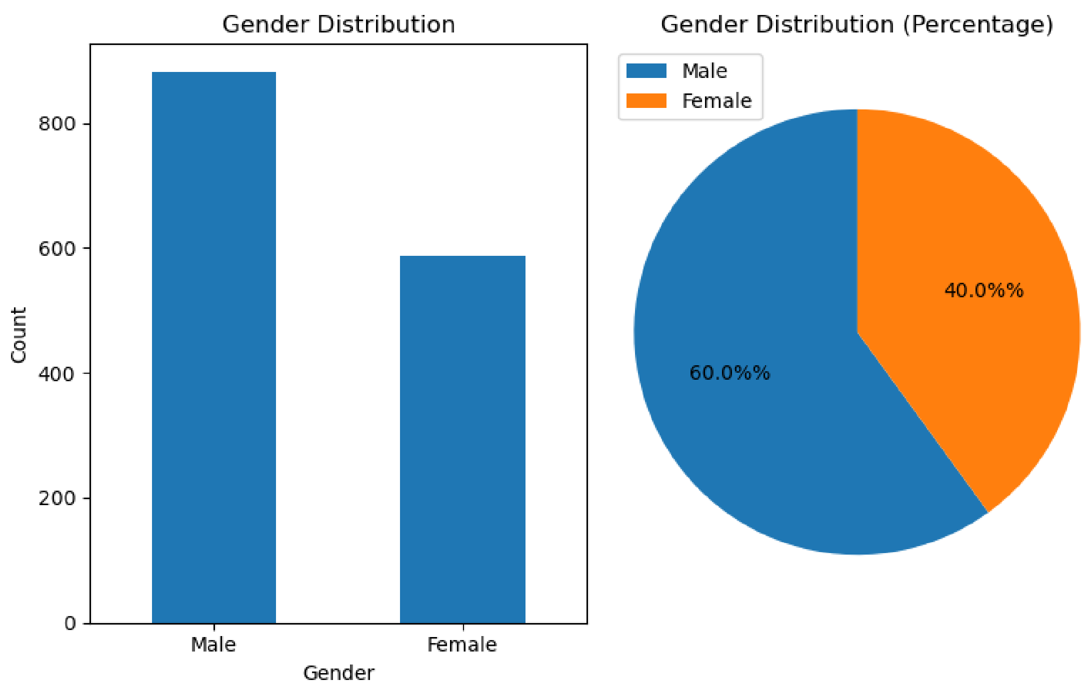
Gender Status:

% of Employees

Attrition

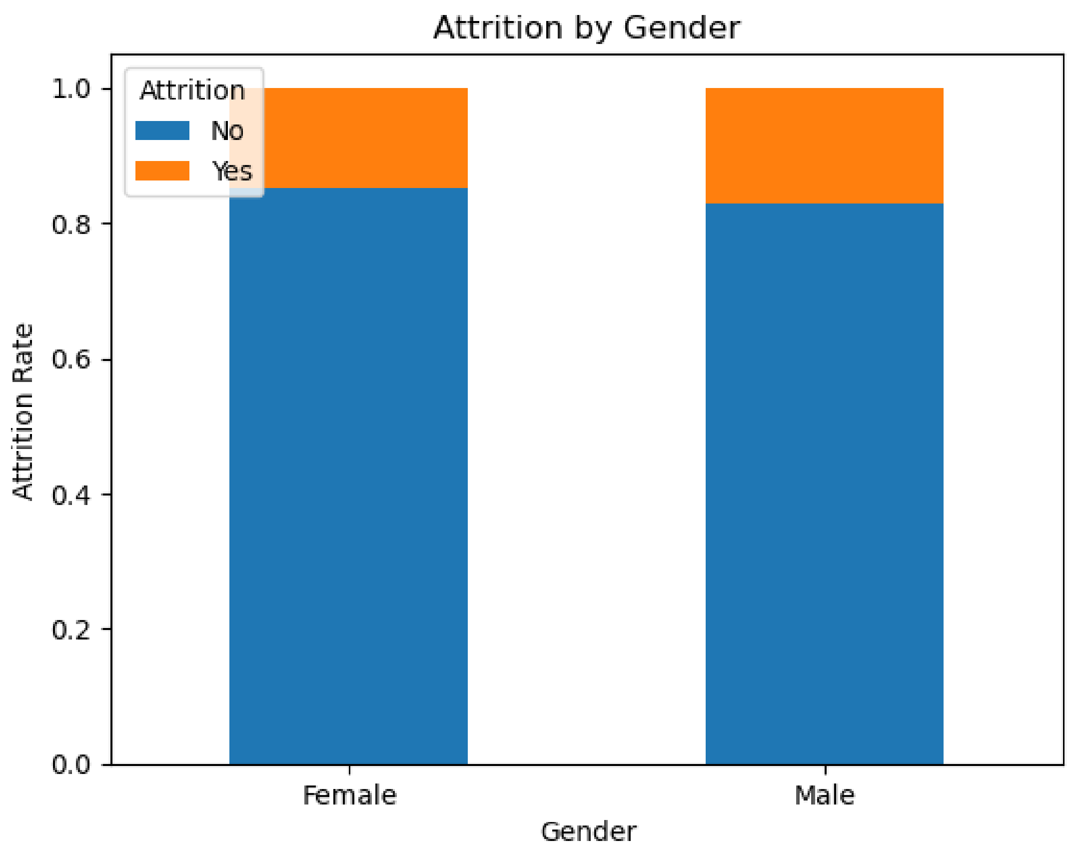
[34]:
```
# Calculate the count and percentage of employees in each gender category
gender_counts = hr['Gender'].value_counts()
gender_percentage = hr['Gender'].value_counts(normalize=True) * 100

# Create a pie chart to visualize the distribution
plt.figure(figsize=(8, 5))
```

```
plt.subplot(1, 2, 1)
gender_counts.plot(kind='bar', rot=0)
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')

plt.subplot(1, 2, 2)
# Plot a pie chart with legends
wedges, _, autotexts = plt.pie(gender_percentage, autopct='%1.1f%%',␣
↪startangle=90)
# Add legends
plt.legend(gender_percentage.index, loc='best')
# Set title and aspect ratio
plt.title('Gender Distribution (Percentage)')
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a␣
↪circle
# Display the pie chart
plt.tight_layout()

# Remove labels and show only percentage values with '%' symbol
for autotext in autotexts:
    autotext.set_text(f"{autotext.get_text()}%")

plt.show()
```



[36]:

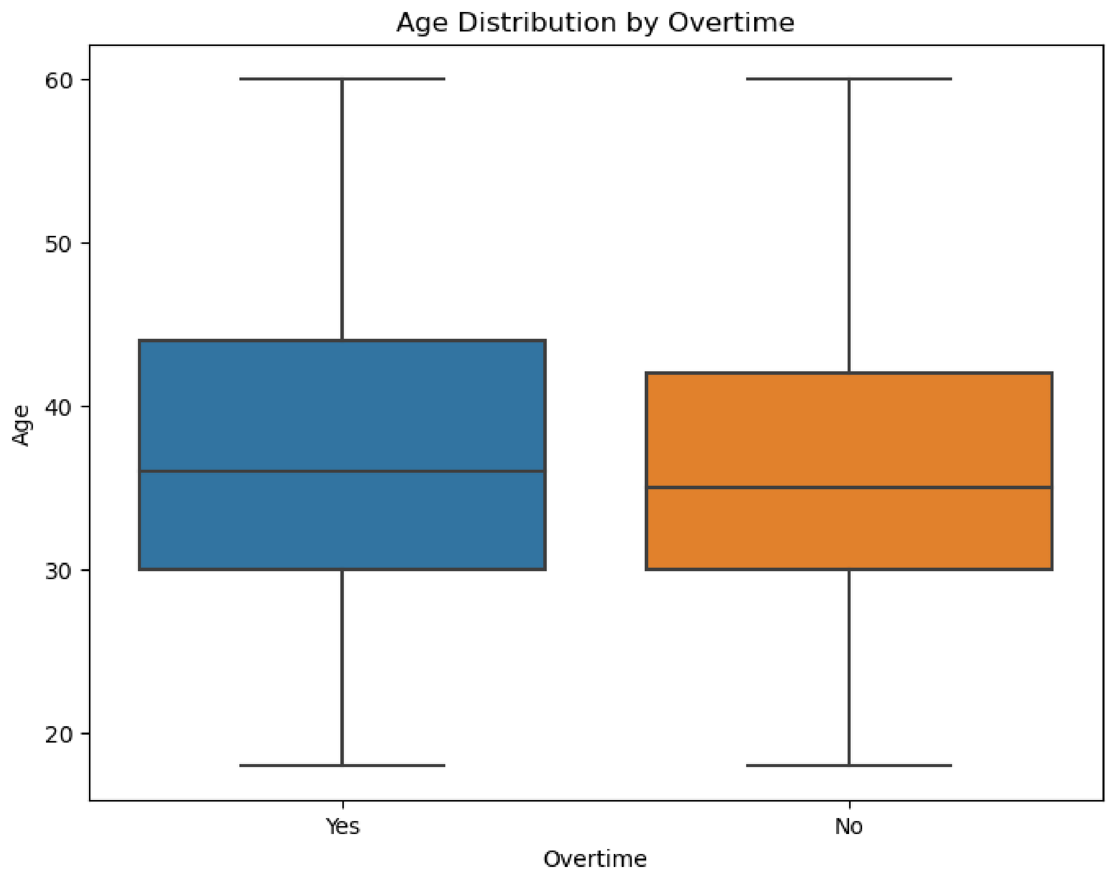
```
# Calculate attrition rates for each gender
attrition_by_gender = hr.groupby('Gender')['Attrition'].
 ↪value_counts(normalize=True).unstack()
attrition_by_gender.plot(kind='bar', stacked=True, rot=0)
plt.title('Attrition by Gender')
plt.xlabel('Gender')
plt.ylabel('Attrition Rate')
plt.legend(title='Attrition', loc='upper left')
plt.show()
```



Relation between Overtime and Age:

[38]:

```
# Create a box plot to compare age distributions
plt.figure(figsize=(8, 6))
sns.boxplot(x='OverTime', y='Age', data=hr)
plt.title('Age Distribution by Overtime')
plt.xlabel('Overtime')
plt.ylabel('Age')
plt.show()
```



[39]:
```
# Calculate summary statistics for age by overtime
age_summary_by_overtime = hr.groupby('OverTime')['Age'].describe()
print(age_summary_by_overtime)
```

```
           count       mean       std   min   25%   50%   75%   max
OverTime
No        1054.0  36.762808  8.975894  18.0  30.0  35.0  42.0  60.0
Yes        416.0  37.331731  9.526402  18.0  30.0  36.0  44.0  60.0
```

[41]:
```
from scipy import stats

# Separate the data into two groups: overtime and no overtime
age_overtime = hr[hr['OverTime'] == 'Yes']['Age']
age_no_overtime = hr[hr['OverTime'] == 'No']['Age']

# Perform a two-sample t-test
t_stat, p_value = stats.ttest_ind(age_overtime, age_no_overtime)

# Print the results
print(f'T-Statistic: {t_stat}')
```

```
print(f'P-Value: {p_value}')
```

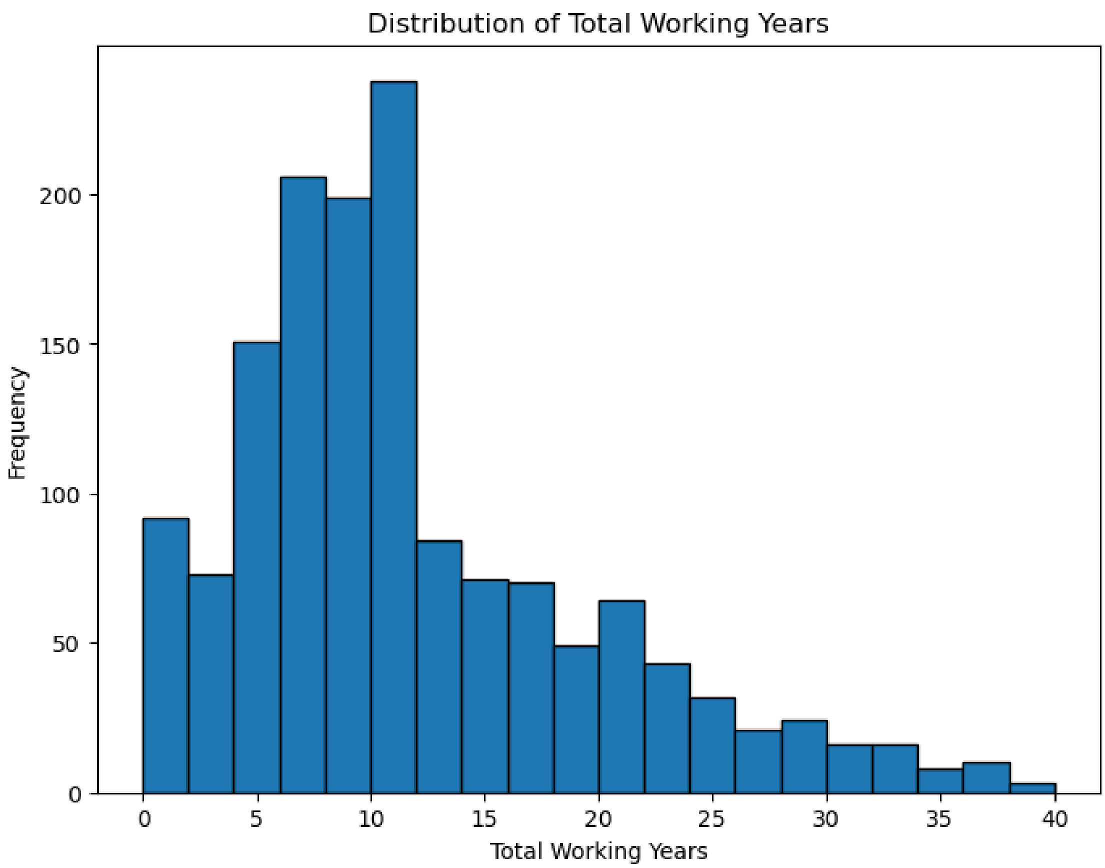
```
T-Statistic: 1.0756184531226642
P-Value: 0.28227467589630123
```

Total Working Years Status:

Distribution

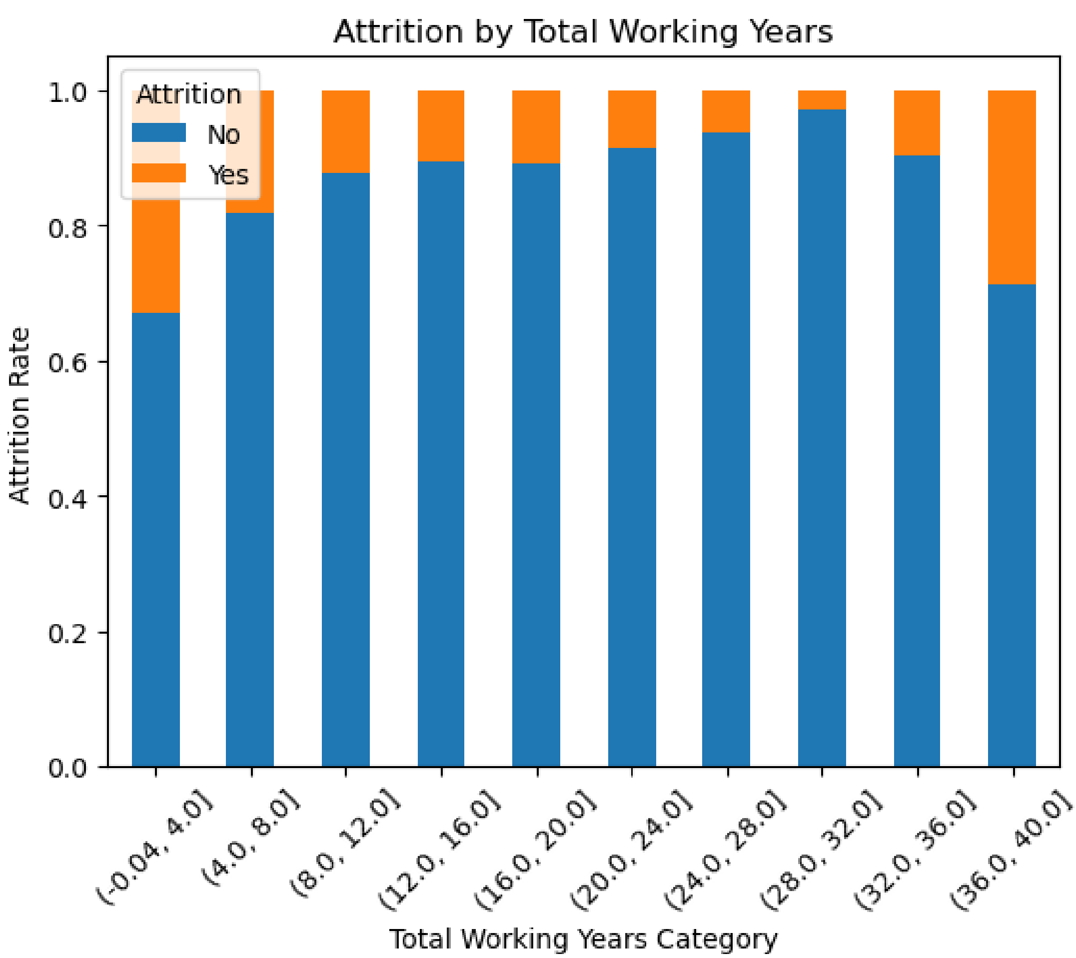
Attrition

[42]:
```
# Create a histogram to visualize the distribution of total working years
plt.figure(figsize=(8, 6))
plt.hist(hr['TotalWorkingYears'], bins=20, edgecolor='k')
plt.title('Distribution of Total Working Years')
plt.xlabel('Total Working Years')
plt.ylabel('Frequency')
plt.show()
```



[43]:
```
# Define the number of bins
num_bins = 10
```

```
# Create a new column for total working years category using a fixed number of␣
↪bins
hr['TotalWorkingYearsCategory'] = pd.cut(hr['TotalWorkingYears'], bins=num_bins)

# Calculate attrition rates for each total working years category
attrition_by_working_years = hr.
↪groupby('TotalWorkingYearsCategory')['Attrition'].
↪value_counts(normalize=True).unstack()
attrition_by_working_years.plot(kind='bar', stacked=True, rot=45)
plt.title('Attrition by Total Working Years')
plt.xlabel('Total Working Years Category')
plt.ylabel('Attrition Rate')
plt.legend(title='Attrition', loc='upper left')
plt.show()

# Remove the added column to avoid confusion
hr.drop(columns=['TotalWorkingYearsCategory'], inplace=True)
```



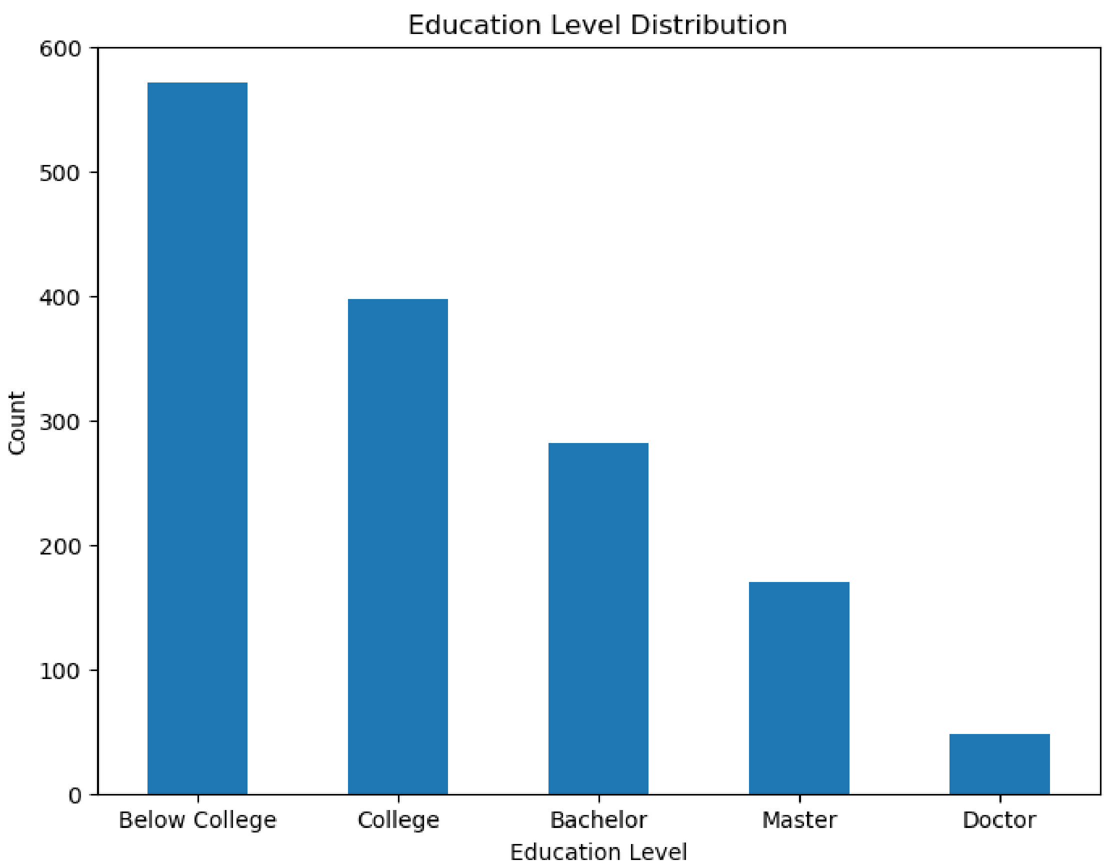
Education Status:

Distribution

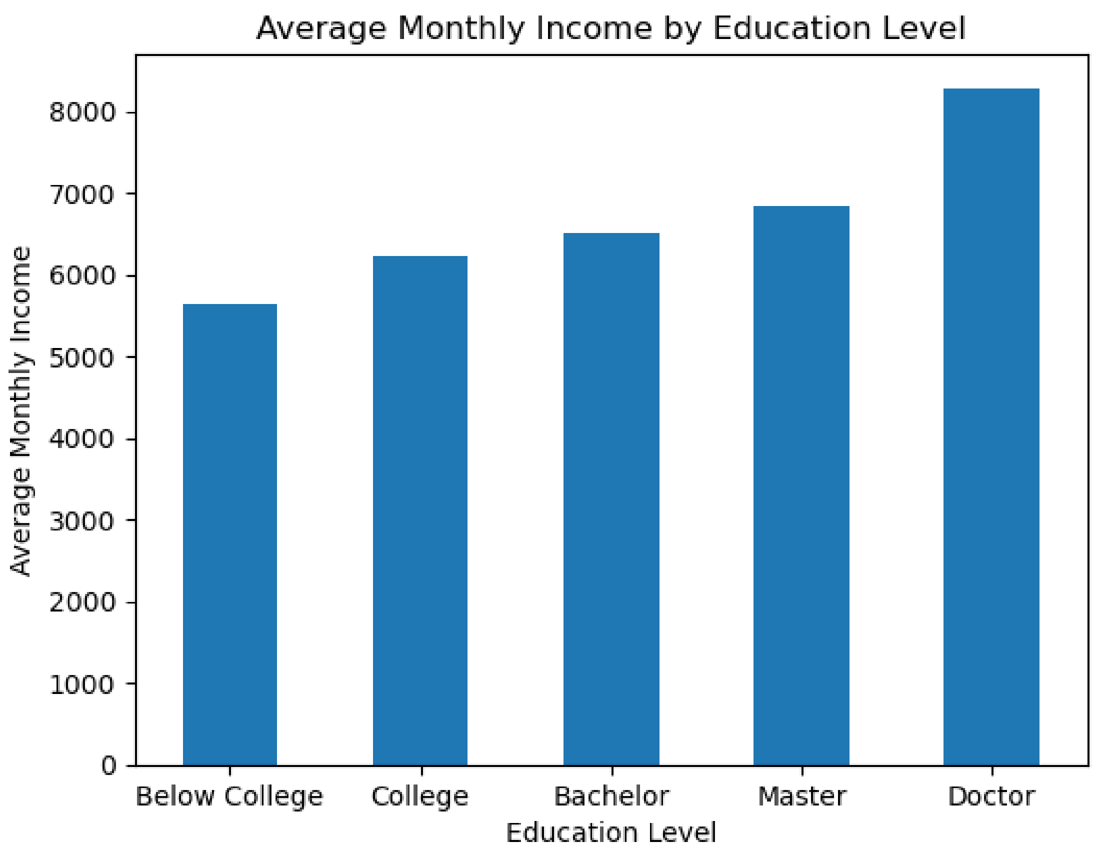
Average Monthly Income

Job Role

[44]:
```
# Calculate the count and percentage of employees at each education level
education_counts = hr['Education'].value_counts()
education_percentage = hr['Education'].value_counts(normalize=True) * 100

# Create a bar chart to visualize the distribution
plt.figure(figsize=(8, 6))
education_counts.plot(kind='bar', rot=0)
plt.title('Education Level Distribution')
plt.xlabel('Education Level')
plt.ylabel('Count')
plt.xticks(range(0, 5), ['Below College', 'College', 'Bachelor', 'Master',␣
↪'Doctor'])
plt.show()
```

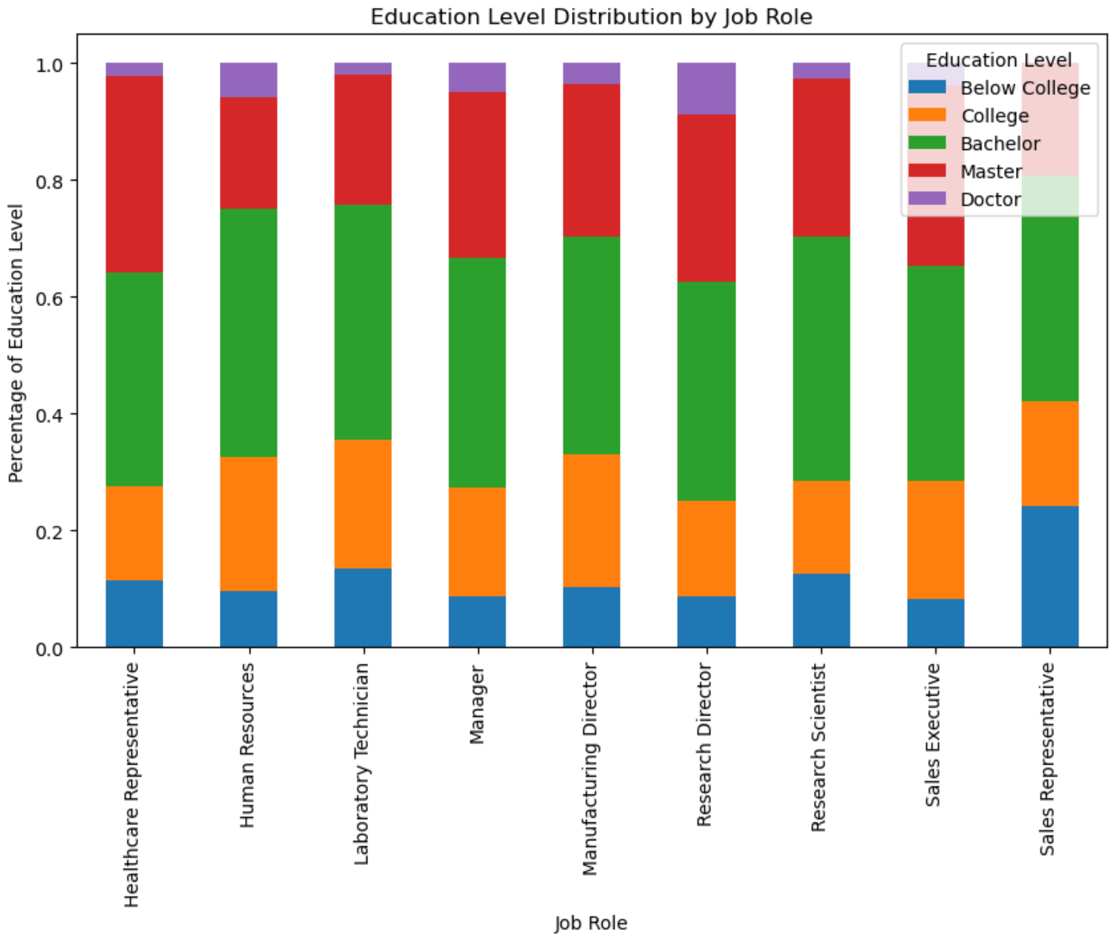


[45]:
```
# Calculate average monthly income by education level
average_income_by_education = hr.groupby('Education')['MonthlyIncome'].mean()
average_income_by_education.plot(kind='bar', rot=0)
plt.title('Average Monthly Income by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Average Monthly Income')
plt.xticks(range(0, 5), ['Below College', 'College', 'Bachelor', 'Master',␣
↪'Doctor'])
plt.show()
```



[46]:
```
# Create a crosstab to analyze education levels by job role
education_job_crosstab = pd.crosstab(hr['JobRole'], hr['Education'],␣
↪normalize='index')
education_job_crosstab.plot(kind='bar', stacked=True, figsize=(10, 6))
plt.title('Education Level Distribution by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Percentage of Education Level')
```

```
plt.legend(title='Education Level', loc='upper right', labels=['Below College', 'College', 'Bachelor', 'Master', 'Doctor'])
plt.xticks(rotation=90)
plt.show()
```



Number of Companies Worked Status:

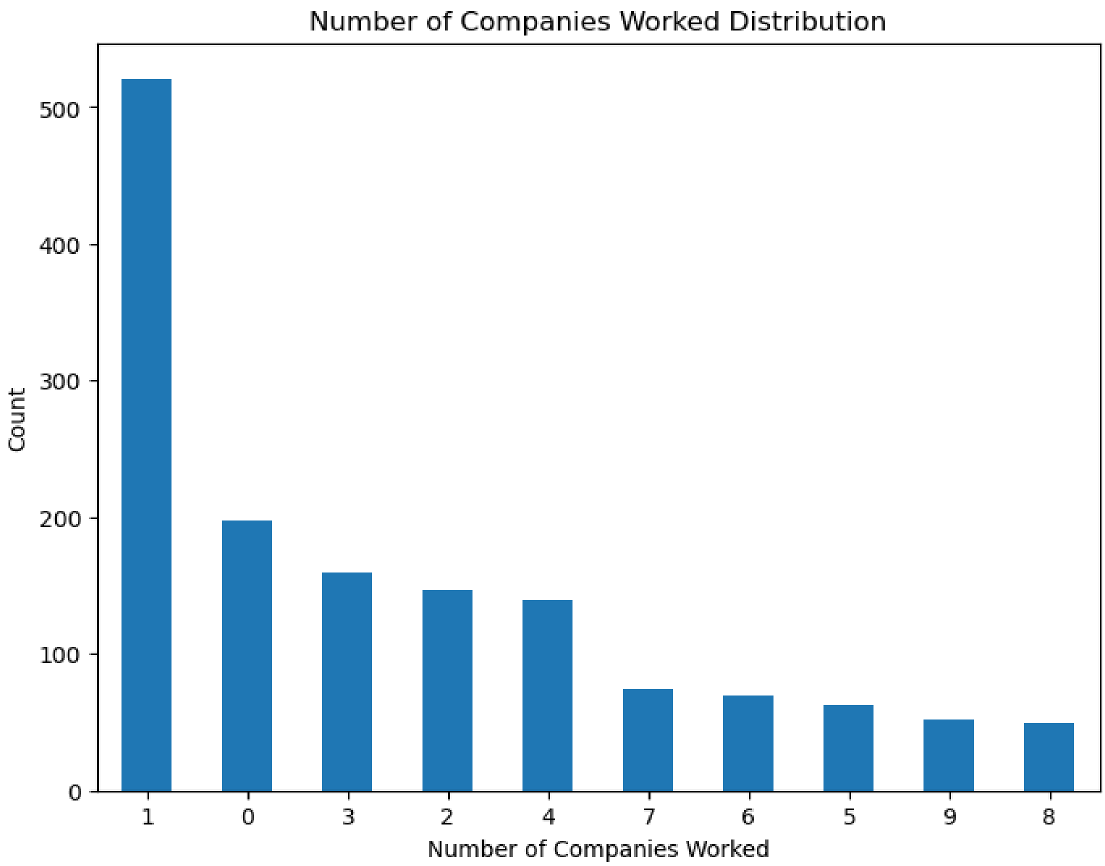
Distribution

Average Monthly Income

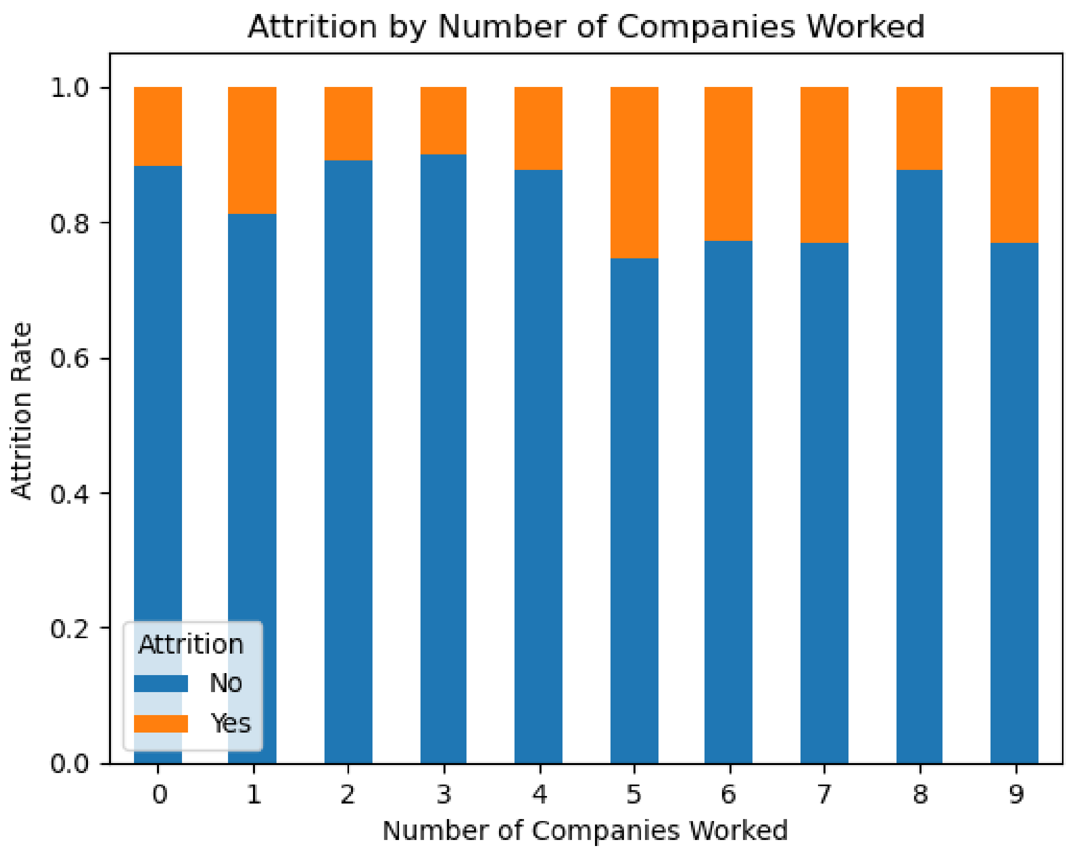
Attrition

[47]:
```
# Calculate the count and percentage of employees for each number of companies worked
companies_worked_counts = hr['NumCompaniesWorked'].value_counts()
companies_worked_percentage = hr['NumCompaniesWorked'].value_counts(normalize=True) * 100
```

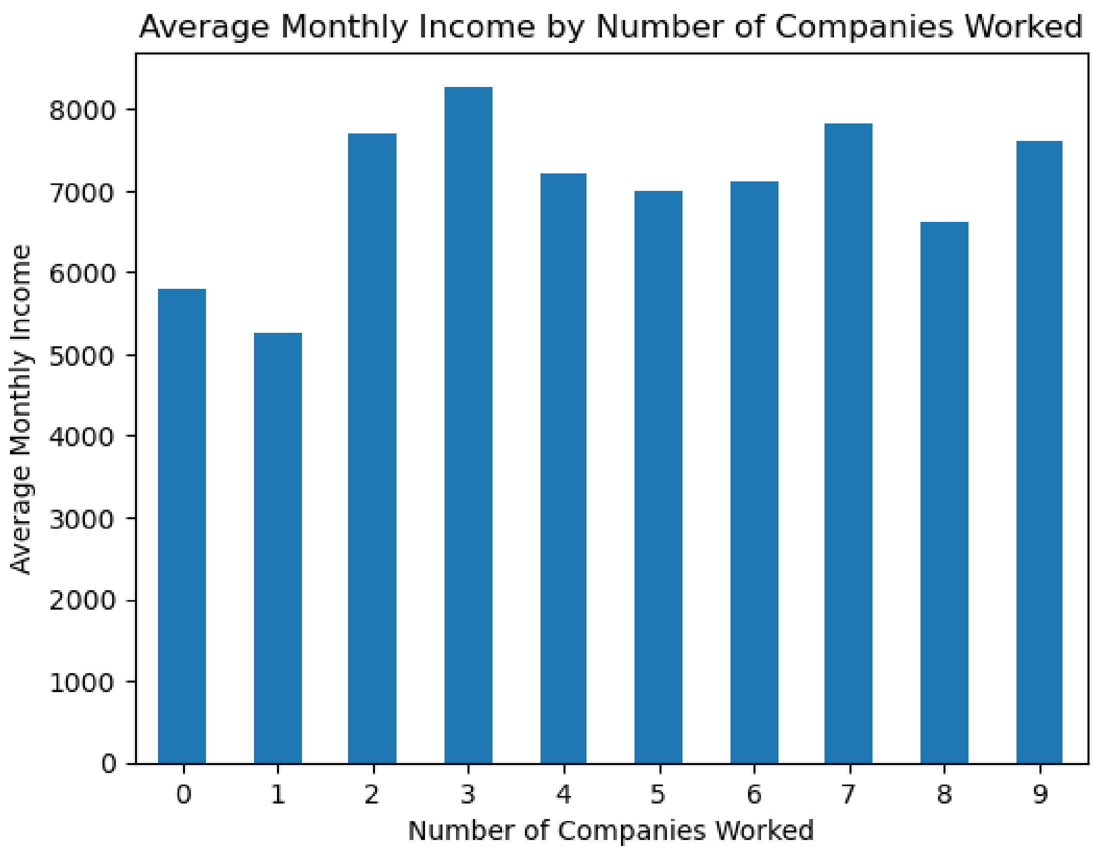
```
# Create a bar chart to visualize the distribution
plt.figure(figsize=(8, 6))
companies_worked_counts.plot(kind='bar', rot=0)
plt.title('Number of Companies Worked Distribution')
plt.xlabel('Number of Companies Worked')
plt.ylabel('Count')
plt.show()
```



[48]:
```
# Calculate attrition rates by the number of companies worked
attrition_by_companies_worked = hr.groupby('NumCompaniesWorked')['Attrition'].
  ↪value_counts(normalize=True).unstack()
attrition_by_companies_worked.plot(kind='bar', stacked=True, rot=0)
plt.title('Attrition by Number of Companies Worked')
plt.xlabel('Number of Companies Worked')
plt.ylabel('Attrition Rate')
plt.show()
```



[49]:
```
# Calculate average monthly income by the number of companies worked
average_income_by_companies_worked = hr.
 ↪groupby('NumCompaniesWorked')['MonthlyIncome'].mean()
average_income_by_companies_worked.plot(kind='bar', rot=0)
plt.title('Average Monthly Income by Number of Companies Worked')
plt.xlabel('Number of Companies Worked')
plt.ylabel('Average Monthly Income')
plt.show()
```



Distance From Home Status:

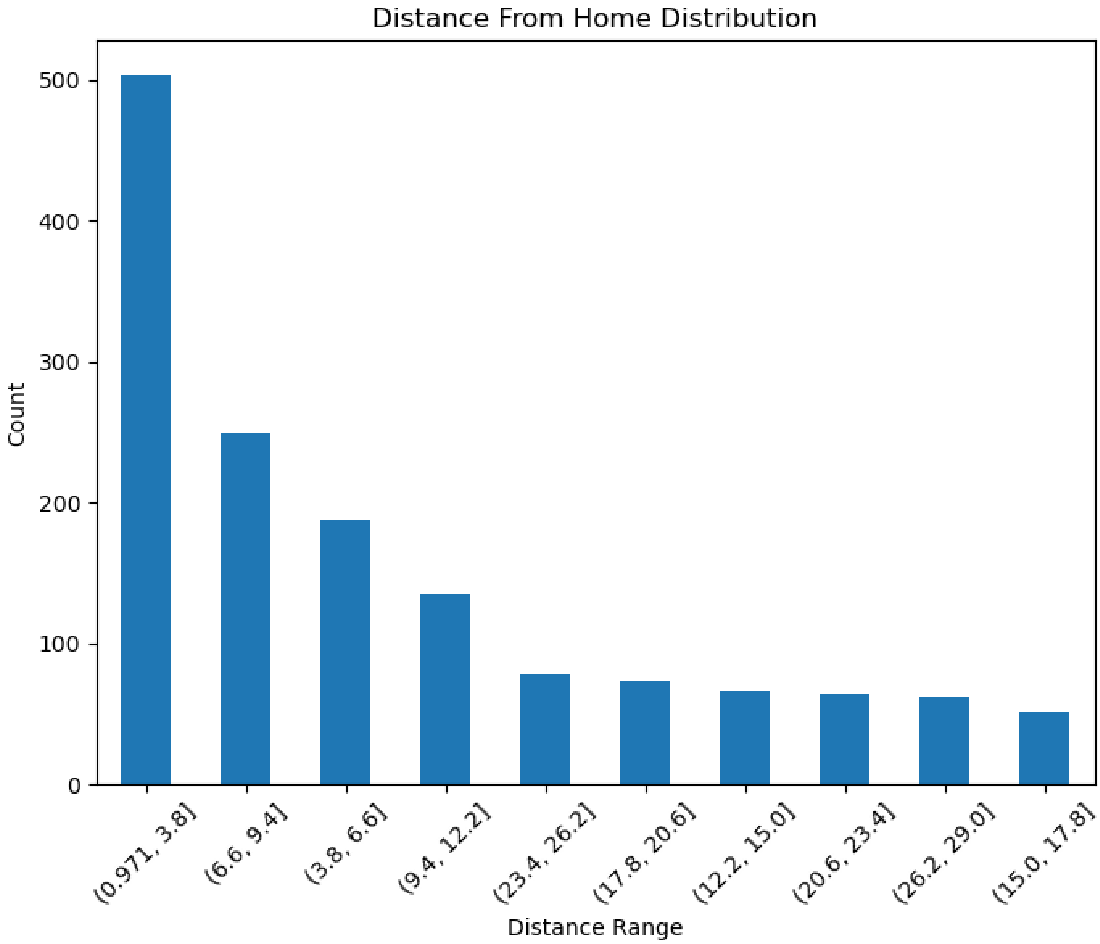
Distribution

Job Role

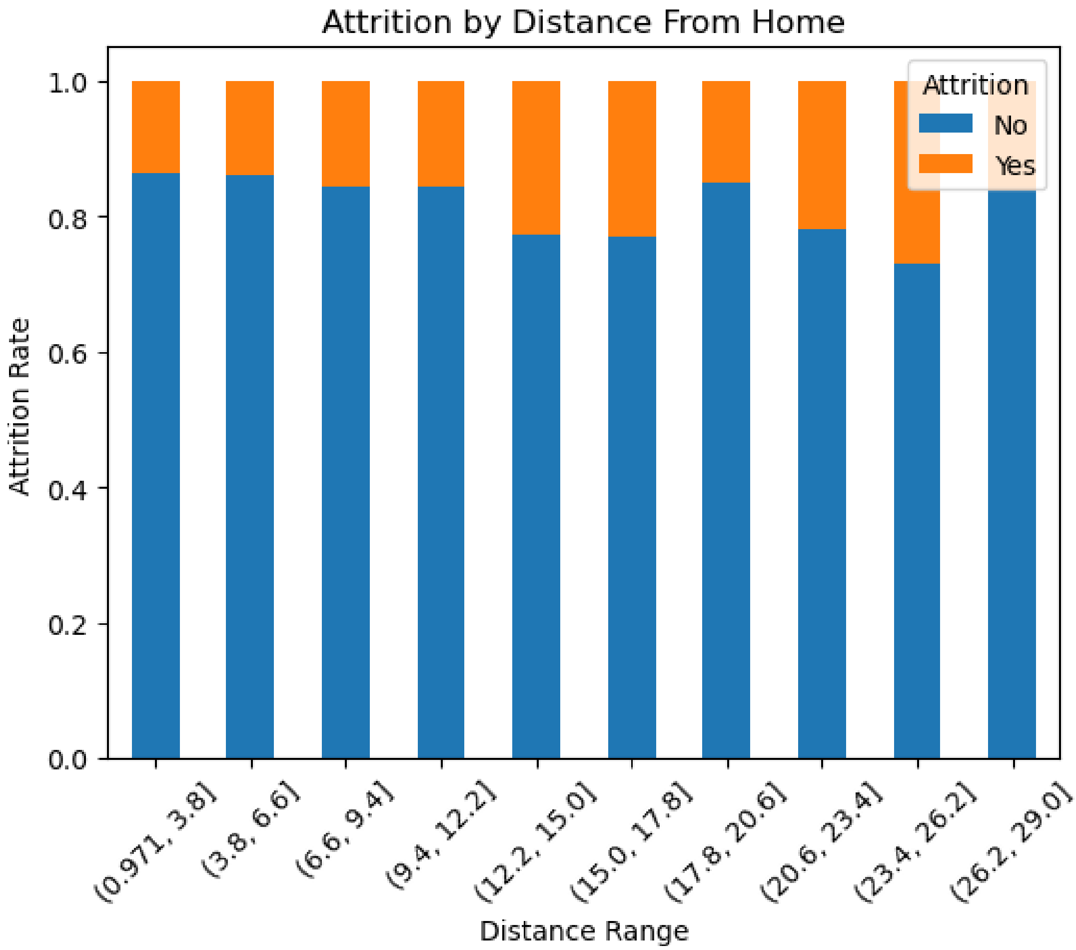
Attrition

[50]:
```
# Define the number of bins for distance ranges (you can adjust this based on
↪your data)
num_bins = 10

# Calculate the count and percentage of employees for each distance range
distance_counts = pd.cut(hr['DistanceFromHome'], bins=num_bins,
↪include_lowest=True).value_counts()
distance_percentage = distance_counts / len(hr) * 100

# Create a histogram to visualize the distribution
plt.figure(figsize=(8, 6))
distance_counts.plot(kind='bar', rot=0)
plt.title('Distance From Home Distribution')
```

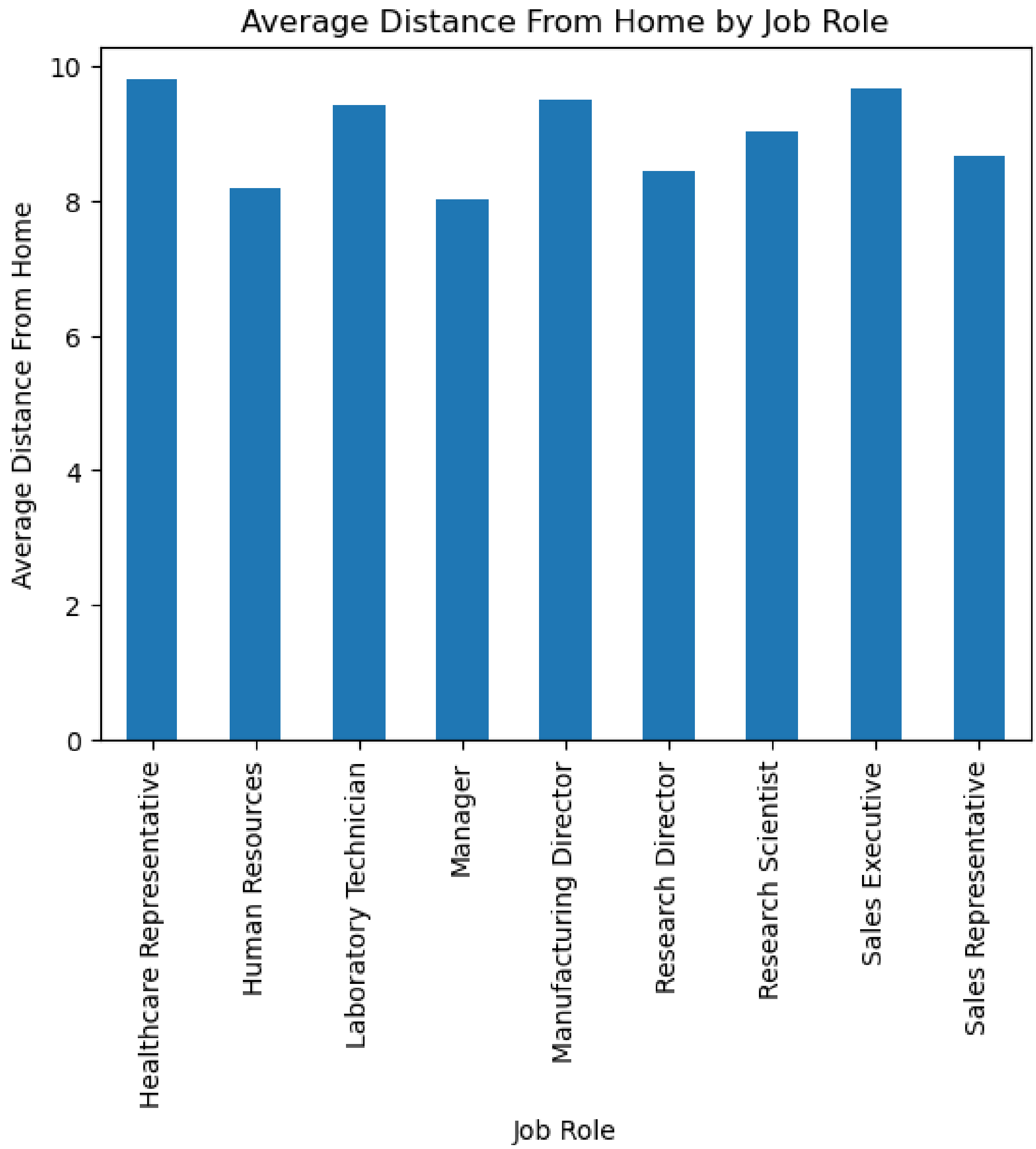
```
plt.xlabel('Distance Range')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



[51]:
```
# Calculate attrition rates by distance range
attrition_by_distance = hr.groupby(pd.cut(hr['DistanceFromHome'],␣
  ↪bins=num_bins, include_lowest=True))['Attrition'].
  ↪value_counts(normalize=True).unstack()
attrition_by_distance.plot(kind='bar', stacked=True, rot=0)
plt.title('Attrition by Distance From Home')
plt.xlabel('Distance Range')
plt.ylabel('Attrition Rate')
plt.xticks(rotation=45)
plt.show()
```



[52]:
```
# Calculate average distance from home by job role
average_distance_by_role = hr.groupby('JobRole')['DistanceFromHome'].mean()
average_distance_by_role.plot(kind='bar', rot=45)
plt.title('Average Distance From Home by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Average Distance From Home')
plt.xticks(rotation=90)
plt.show()
```



2 Employee Attrition Analysis Report

2.1 Introduction:

This data analysis report aims to provide insights into employee attrition within the organization. By examining various factors, including demographics, job-related variables, and overall work environment, the report seeks to assist stakeholders in understanding patterns and areas for improvement.

2.2 Correlation Map:

- A correlation map for all numeric variables was generated, revealing significant relationships among various factors affecting employee attrition. Key correlated factors include Total Working Years, Monthly Income, and Job Level.

2.3 Overtime:

2.3.1 Attrition Rates by Overtime:

- Employees working overtime tend to have a higher attrition rate.
- 17% of employees working overtime experience attrition, compared to 12% for those not working overtime. ### Average Job Satisfaction by Overtime:
- Employees working overtime (2.77) have slightly higher average job satisfaction compared to those without overtime (2.71).

2.4 Marital Status:

2.4.1 Marital Status Distribution:

- Married: 45.78%
- Single: 31.97%
- Divorced: 22.24% ### Attrition by Marital Status:
- Single employees have the highest attrition rate (14.80%) compared to married employees (10.60%) and divorced employees (11.46%).

2.5 Job Role:

2.5.1 Attrition by Job Role:

- Sales Representatives (39.76%) and Laboratory Technicians (23.94%) exhibit higher attrition rates. ### Percentage of Workforce by Job Role:
- Sales Executives (22.18%) and Research Scientists (19.86%) constitute the largest portions of the workforce.

2.6 Gender:

2.6.1 Gender Distribution:

- Male: 60%
- Female: 40% ### Attrition by Gender:
- Attrition rates are comparable between genders, with males at 17.01% and females at 14.80%.

2.7 Education Level:

2.7.1 Education Level Distribution:

- Education Level 3 (38.91%) and Level 4 (27.07%) are most prevalent. ### Average Monthly Income by Education Level:
- Higher education levels correspond to higher average monthly incomes.

2.8 Department:

2.8.1 Attrition by Department:

- Research & Development has the highest attrition rate (15.47%).

2.9 Business Travel:

2.9.1 Attrition by Business Travel:

- Employees who travel frequently have a higher attrition rate (24.24%).

2.10 Relation between Overtime and Age:

- Both groups, with and without overtime, have similar age distributions.
- No significant difference in age is observed between those working overtime and those who are not.

2.11 Total Working Years:

2.11.1 Attrition by Total Working Years:

- Employees with fewer working years tend to have higher attrition rates.

2.12 Number of Companies Worked:

2.12.1 Number of Companies Worked Distribution:

- Most employees have worked for one or fewer companies. ### Average Monthly Income by Number of Companies Worked:
- Employees who have worked for more companies tend to have higher average monthly incomes.

2.13 Distance from Home:

2.13.1 Distance from Home Distribution:

- Majority of employees have a distance from home between 0.971 and 3.8 miles. ### Attrition by Distance from Home:
- Employees with longer commutes (e.g., 23.4 to 26.2 miles) tend to have higher attrition rates.

2.14 Conclusion:

This comprehensive analysis provides valuable insights into employee attrition patterns. Understanding these factors allows us to make informed decisions and implement targeted strategies to improve employee retention and satisfaction. The correlation map and specific insights for Overtime, Marital Status, Job Role, Gender, Education, Department, Business Travel, Age, Total Working Years, Number of Companies Worked, and Distance from Home are crucial for developing effective HR policies and addressing attrition challenges.