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

```
df = pd.read_excel('HR Data.xlsx')
```

```
df.head()
```

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Performance Rating	Relatic Satisfa
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	3	
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	4	
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	3	
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	3	
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	3	

5 rows × 41 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 41 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   Attrition                  1470 non-null   object
 1   Business Travel            1470 non-null   object
 2   CF_age band                1470 non-null   object
 3   CF_attrition label         1470 non-null   object
 4   Department                 1470 non-null   object
 5   Education Field            1470 non-null   object
 6   emp no                     1470 non-null   object
 7   Employee Number            1470 non-null   int64 
 8   Gender                     1470 non-null   object
 9   Job Role                   1470 non-null   object
 10  Marital Status             1470 non-null   object
 11  Over Time                  1470 non-null   object
 12  Over18                     1470 non-null   object
 13  Training Times Last Year   1470 non-null   int64 
 14  -2                         1470 non-null   int64 
 15  0                          1470 non-null   int64 
 16  Age                        1470 non-null   int64 
 17  CF_current Employee        1470 non-null   int64 
 18  Daily Rate                 1470 non-null   int64 
 19  Distance From Home         1470 non-null   int64 
 20  Education                  1470 non-null   object
 21  Employee Count             1470 non-null   int64 
 22  Environment Satisfaction   1470 non-null   int64 
 23  Hourly Rate                1470 non-null   int64 
 24  Job Involvement            1470 non-null   int64 
 25  Job Level                  1470 non-null   int64 
 26  Job Satisfaction           1470 non-null   int64 
 27  Monthly Income             1470 non-null   int64 
 28  Monthly Rate               1470 non-null   int64 
 29  Num Companies Worked       1470 non-null   int64 
 30  Percent Salary Hike        1470 non-null   int64 
 31  Performance Rating         1470 non-null   int64 
 32  Relationship Satisfaction  1470 non-null   int64 
 33  Standard Hours             1470 non-null   int64 
 34  Stock Option Level         1470 non-null   int64 
 35  Total Working Years        1470 non-null   int64 
 36  Work Life Balance          1470 non-null   int64 
 37  Years At Company           1470 non-null   int64 
 38  Years In Current Role      1470 non-null   int64 
 39  Years Since Last Promotion 1470 non-null   int64 
```

```
 40  Years With Curr Manager     1470 non-null   int64
dtypes: int64(28), object(13)
memory usage: 471.0+ KB
```

```
df.describe()
```

	Employee Number	Training Times Last Year	-2	0	Age	CF_current Employee	Daily Rate	Distance From Home	Employee Count	Environment Satisfaction	...	Performanc Ratin
count	1470.000000	1470.000000	1470.0	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	...	1470.00000
mean	1024.865306	2.799320	-2.0	0.0	36.923810	0.838776	802.485714	9.192517	1.0	2.721769	...	3.15374
std	602.024335	1.289271	0.0	0.0	9.135373	0.367863	403.509100	8.106864	0.0	1.093082	...	0.36082
min	1.000000	0.000000	-2.0	0.0	18.000000	0.000000	102.000000	1.000000	1.0	1.000000	...	3.00000
25%	491.250000	2.000000	-2.0	0.0	30.000000	1.000000	465.000000	2.000000	1.0	2.000000	...	3.00000
50%	1020.500000	3.000000	-2.0	0.0	36.000000	1.000000	802.000000	7.000000	1.0	3.000000	...	3.00000
75%	1555.750000	3.000000	-2.0	0.0	43.000000	1.000000	1157.000000	14.000000	1.0	4.000000	...	3.00000
max	2068.000000	6.000000	-2.0	0.0	60.000000	1.000000	1499.000000	29.000000	1.0	4.000000	...	4.00000

8 rows × 28 columns

```
# Department-wise attrition
dept_attrition = df.groupby('Department')['Attrition'].value_counts(normalize=True).unstack()
print("Department-wise Attrition:\n", dept_attrition)
```

```
Department-wise Attrition:
 Attrition         No       Yes
Department
HR          0.809524  0.190476
R&D         0.861602  0.138398
Sales       0.793722  0.206278
```

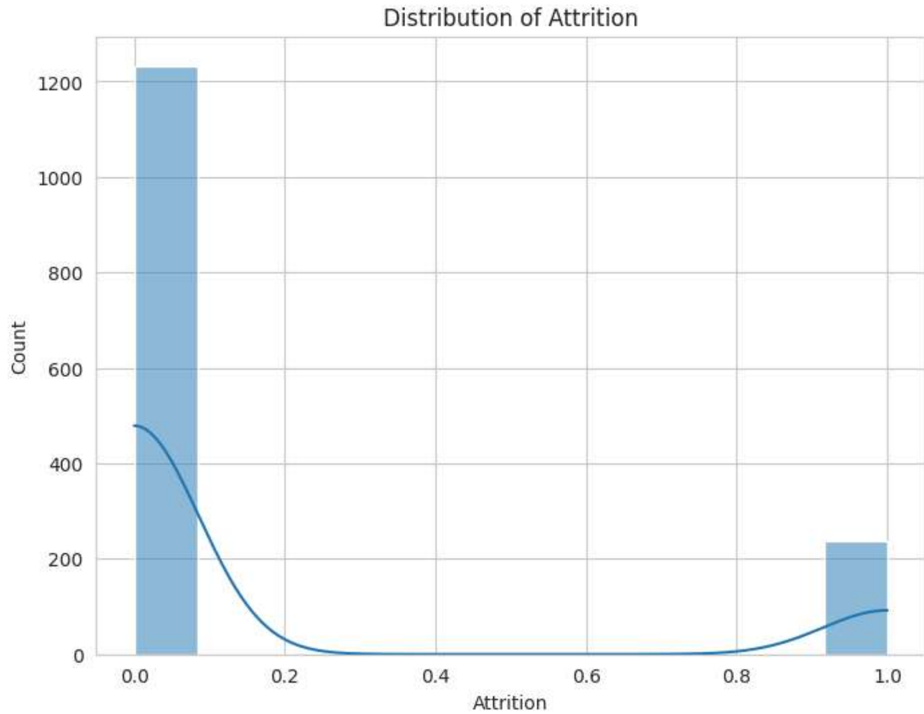
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Assuming 'Attrition' is your target variable and it's categorical (e.g., 'Yes', 'No')
# Convert 'Attrition' to numerical if needed (e.g., 1 for Yes, 0 for No)
if df['Attrition'].dtype == 'object':
    le = LabelEncoder()
    df['Attrition'] = le.fit_transform(df['Attrition'])

# Identify features (X) and target (y)
X = df.drop('Attrition', axis=1)
y = df['Attrition']

# Identify categorical and numerical columns
categorical_features = X.select_dtypes(include=['object']).columns
numerical_features = X.select_dtypes(include=[np.number]).columns

# Create preprocessing pipelines for numerical and categorical features
numerical_transformer = 'passthrough'  # No transformation needed for numerical features for Logistic Regression
categorical_transformer = OneHotEncoder(handle_unknown='ignore')

# Create a column transformer to apply different transformations to different columns
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_features),
        ('cat', categorical_transformer, categorical_features)
    ],
    remainder='passthrough'  # Keep other columns (if any)
)

# Create the Logistic Regression model pipeline
model = Pipeline(steps=[('preprocessor', preprocessor),
                        ('classifier', LogisticRegression(solver='liblinear'))]) # solver='liblinear' is good for small datasets
```

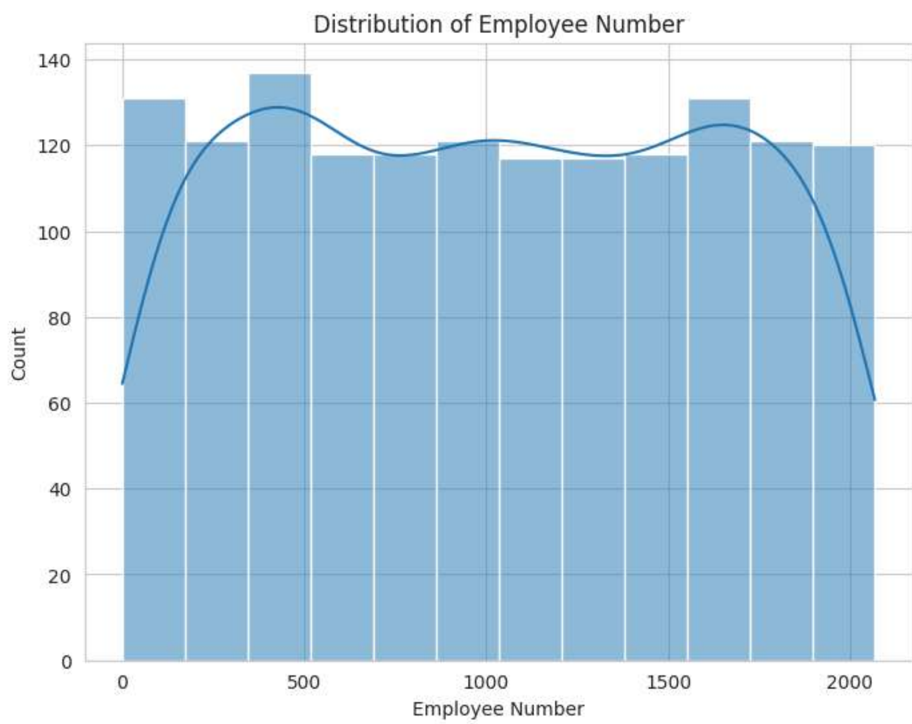
```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("\nModel Evaluation:")
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

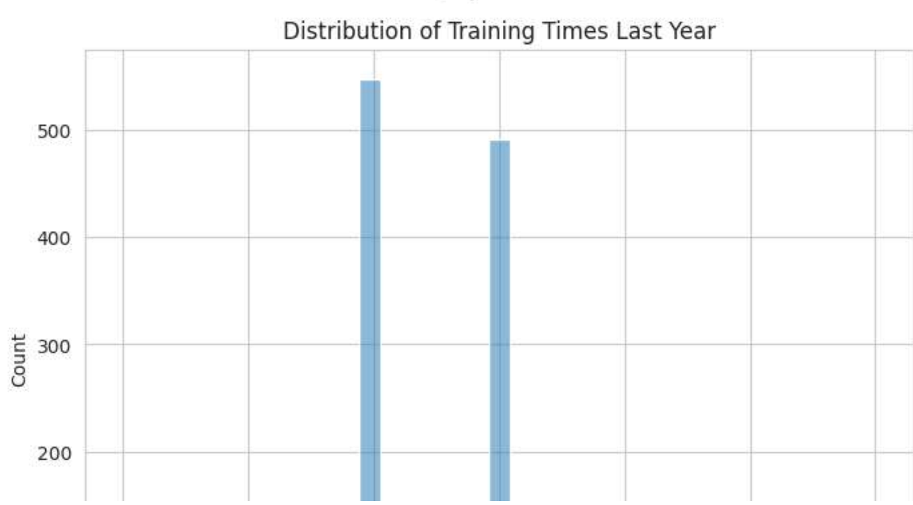
```
Model Evaluation:
Accuracy: 1.0
Confusion Matrix:
 [[247   0]
 [  0  47]]
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00       247
           1       1.00      1.00      1.00        47

    accuracy                           1.00       294
   macro avg       1.00      1.00      1.00       294
weighted avg       1.00      1.00      1.00       294
```

```
# Set a style for the plots
sns.set_style("whitegrid")

# 1. Univariate Analysis

# Histograms for numerical features
numerical_features = df.select_dtypes(include=[np.number]).columns
for col in numerical_features:
    plt.figure(figsize=(8, 6))
    sns.histplot(data=df, x=col, kde=True)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.show()

# Count plots for categorical features
categorical_features = df.select_dtypes(include=['object']).columns
for col in categorical_features:
    plt.figure(figsize=(8, 6))
    sns.countplot(data=df, y=col, order=df[col].value_counts().index)
    plt.title(f'Count of {col}')
    plt.xlabel('Count')
    plt.ylabel(col)
    plt.show()

# 2. Bivariate Analysis

# Scatter plot example (Age vs. MonthlyIncome)
if 'Age' in df.columns and 'MonthlyIncome' in df.columns:
    plt.figure(figsize=(10, 8))
    sns.scatterplot(data=df, x='Age', y='MonthlyIncome', hue='Attrition')
    plt.title('Age vs. Monthly Income (colored by Attrition)')
    plt.xlabel('Age')
    plt.ylabel('Monthly Income')
    plt.show()

# Box plot example (Monthly Income by Department)
if 'MonthlyIncome' in df.columns and 'Department' in df.columns:
    plt.figure(figsize=(12, 8))
    sns.boxplot(data=df, x='Department', y='MonthlyIncome', hue='Attrition')
    plt.title('Monthly Income by Department (colored by Attrition)')
    plt.xlabel('Department')
```

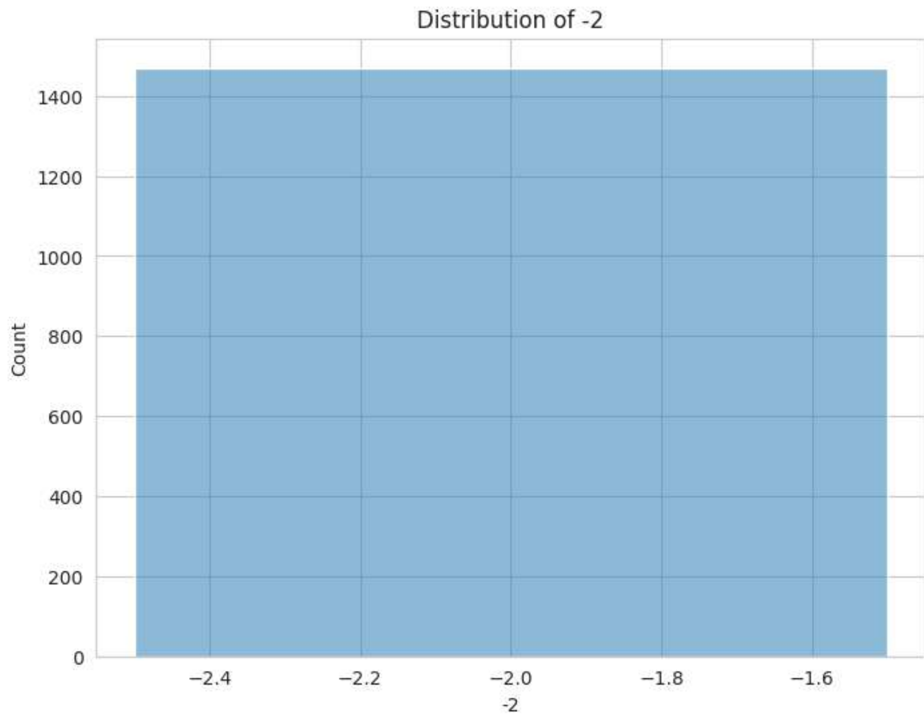
```
    plt.ylabel('Monthly Income')
    plt.xticks(rotation=45, ha='right')
    plt.show()

# Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.select_dtypes(include=[np.number]).corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numerical Features')
plt.show()

# 3. Multivariate Analysis

# Pair plot (example with a subset of numerical features)
numerical_subset = ['Age', 'MonthlyIncome', 'YearsAtCompany', 'Attrition'] # Select relevant numerical features
if all(col in df.columns for col in numerical_subset):
    sns.pairplot(df[numerical_subset], hue='Attrition')
    plt.suptitle('Pair Plot of Selected Numerical Features (colored by Attrition)', y=1.02)
    plt.show()

# 4. Attrition-Specific Analysis

# Attrition rate by various categorical features
categorical_features_for_attrition = [col for col in categorical_features if col != 'Attrition'] # Exclude 'Attrition' itself
for col in categorical_features_for_attrition:
    plt.figure(figsize=(10, 6))
    sns.countplot(data=df, x=col, hue='Attrition', palette='viridis')
    plt.title(f'Attrition Count by {col}')
    plt.xlabel(col)
    plt.ylabel('Count')
    plt.xticks(rotation=45, ha='right')
    plt.show()
```


Distribution of Attrition
Count
Attrition
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0

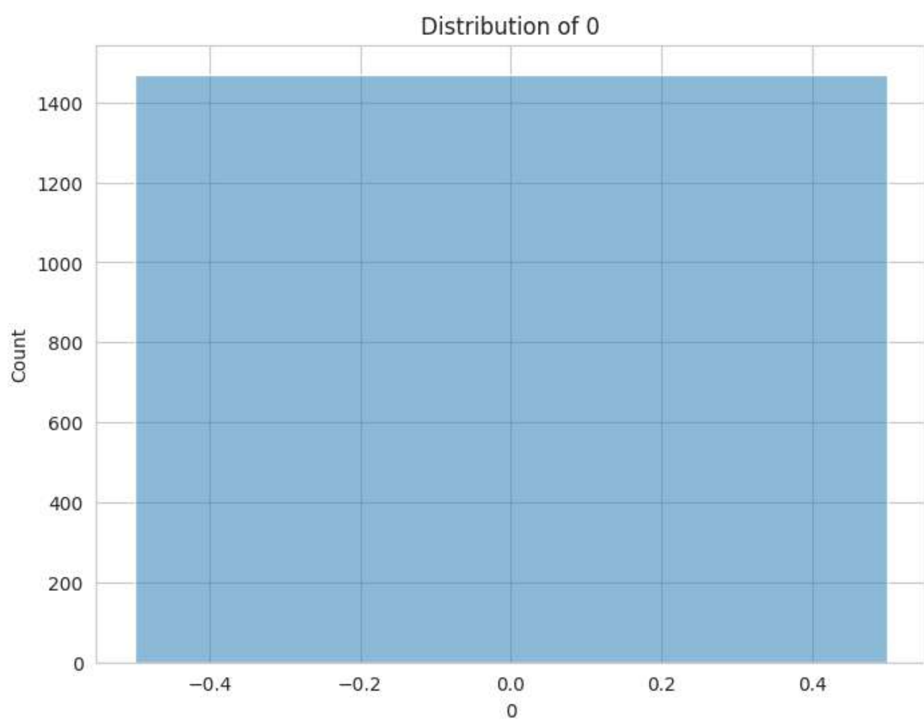

Distribution of Employee Number
Count
Employee Number
140
120
100
80
60
40
20
0
0
500
1000
1500
2000


Distribution of Training Times Last Year
Count
500
400
300
200

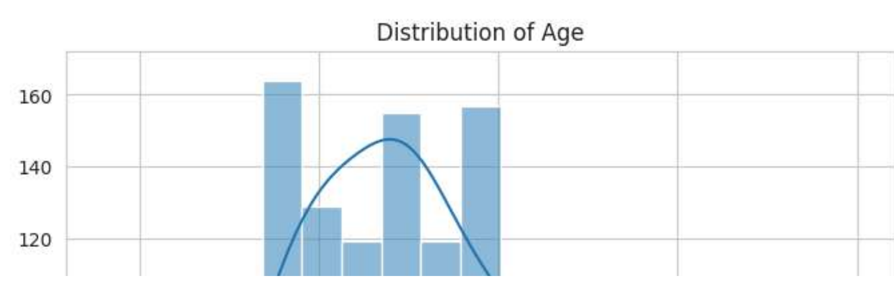
0
100
0
1
2
3
4
5
6
Training Times Last Year



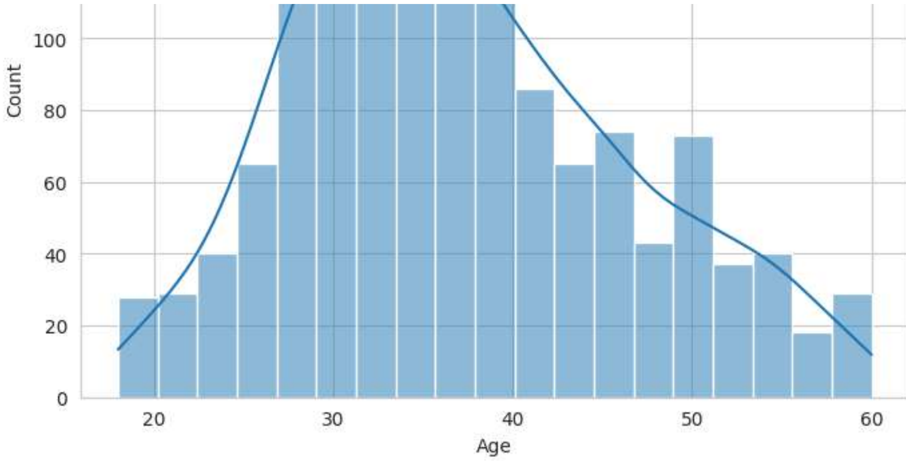
Distribution of -2
Count
0
200
400
600
800
1000
1200
1400
−2.4
−2.2
−2.0
−1.8
−1.6
-2



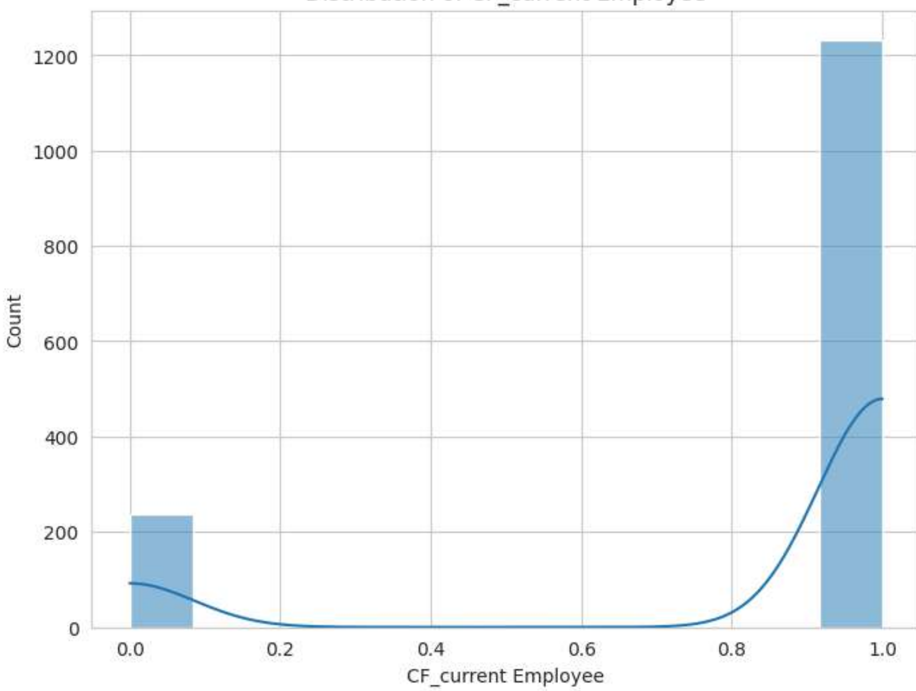
Distribution of 0
Count
0
200
400
600
800
1000
1200
1400
−0.4
−0.2
0.0
0.2
0.4
0



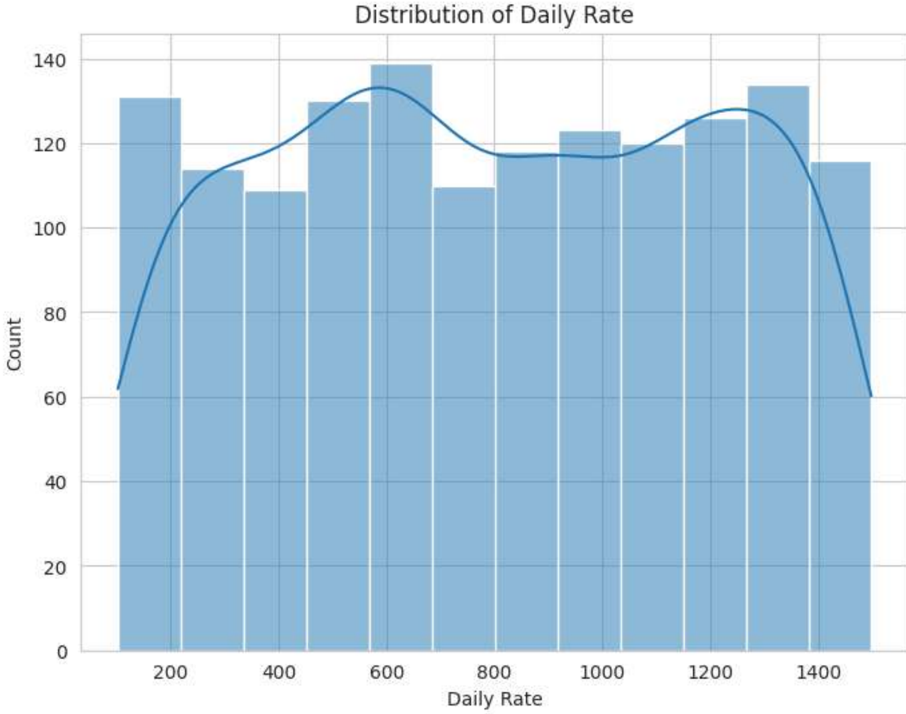
Distribution of Age
120
140
160



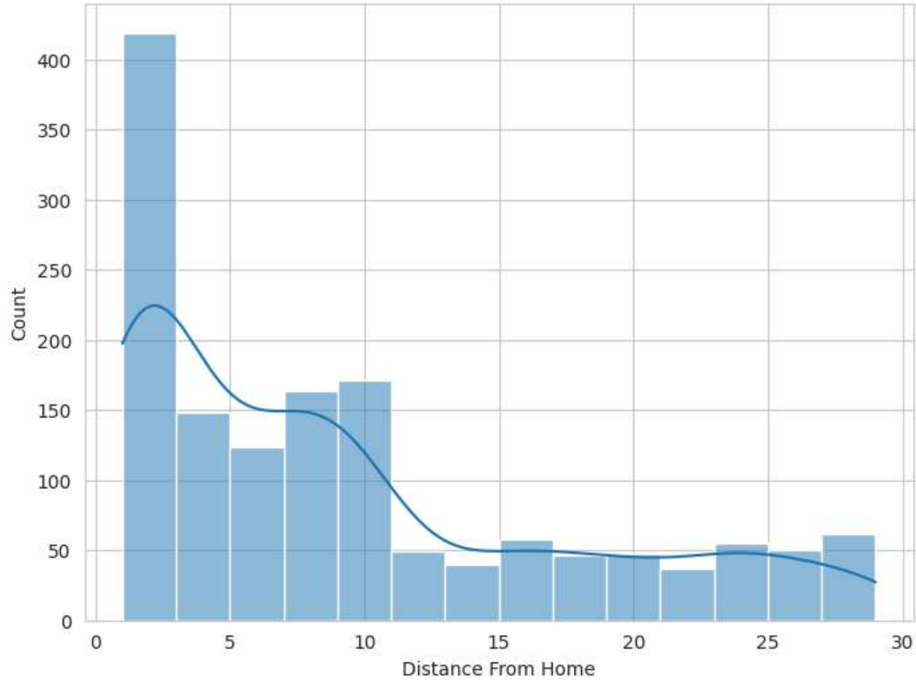
Distribution of CF_current Employee

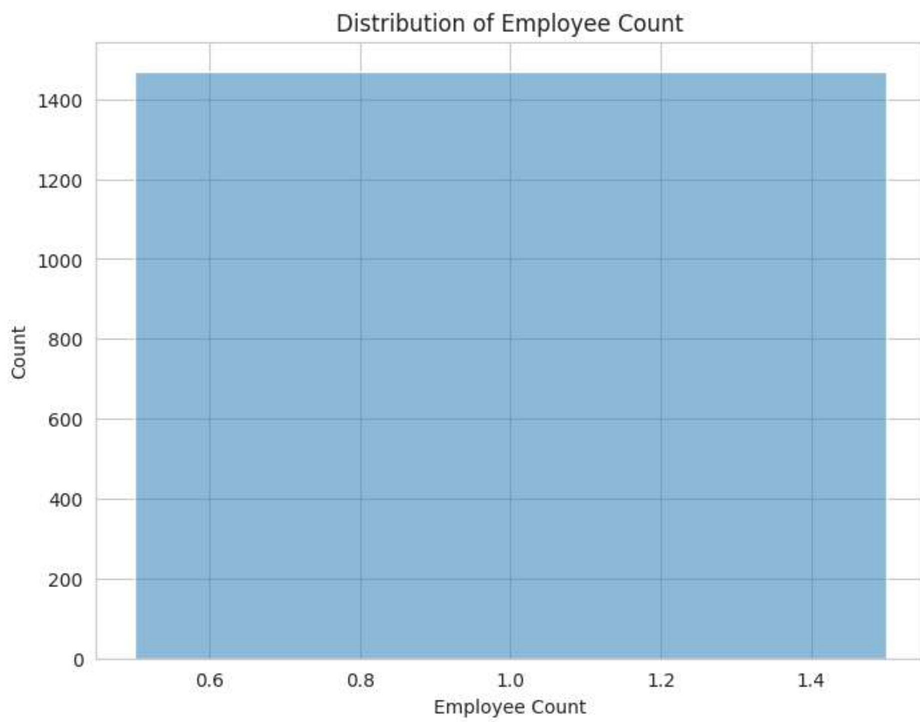


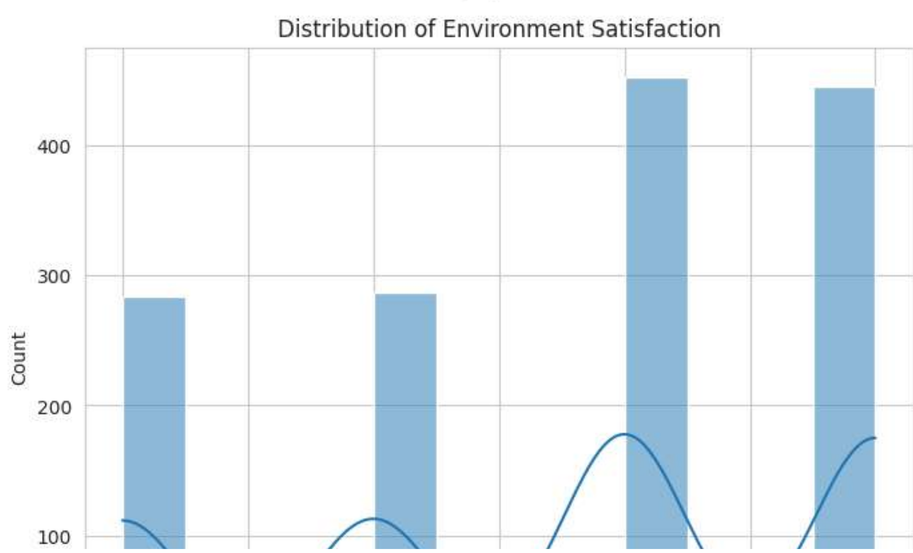
Distribution of Daily Rate

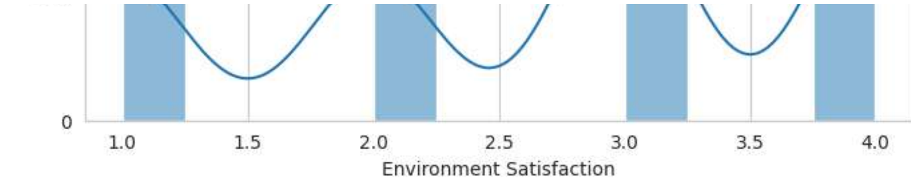


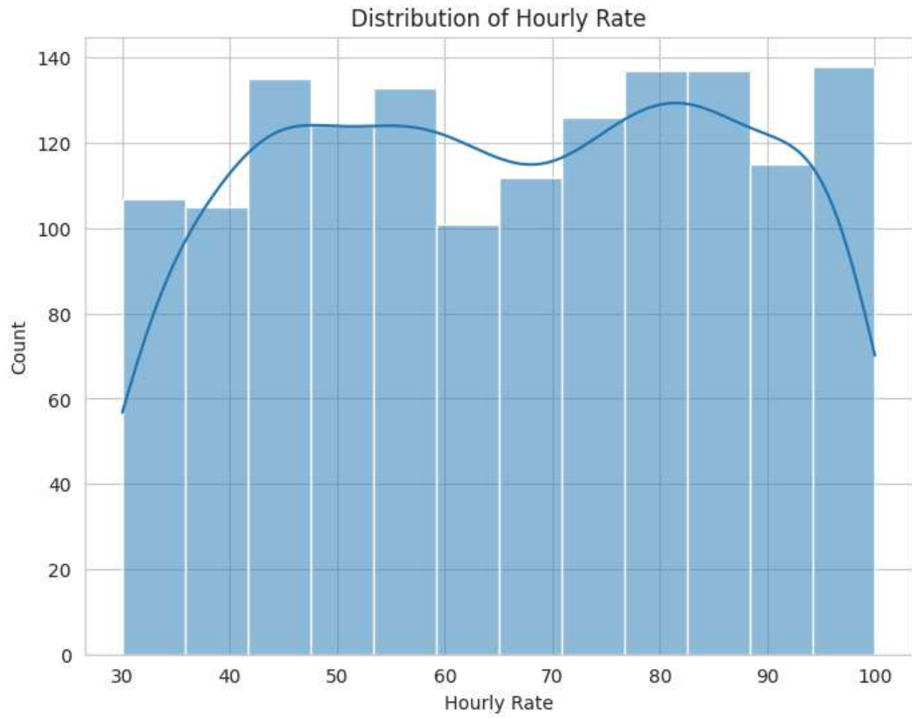
Distribution of Distance From Home

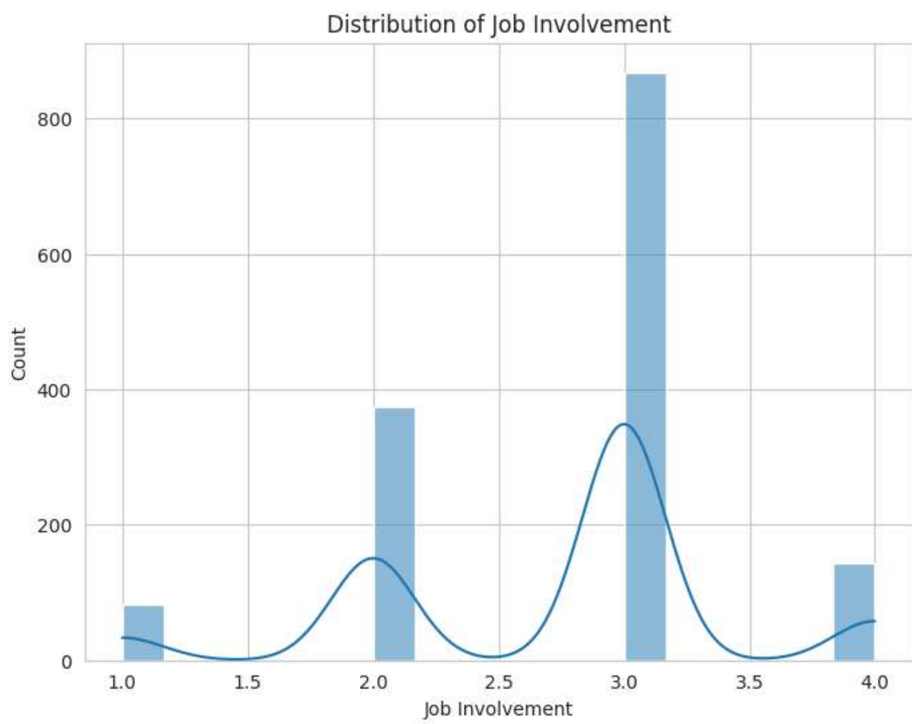

400
350
300
250
200
150
100
50
0
Count
0
5
10
15
20
25
30
Distance From Home

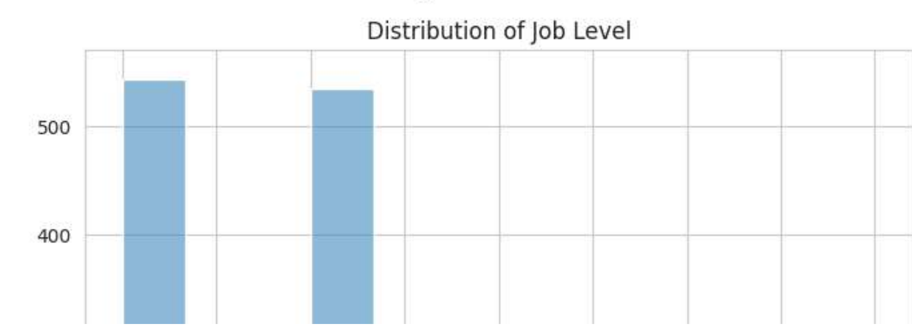

Distribution of Employee Count
1400
1200
1000
800
600
400
200
0
Count
0.6
0.8
1.0
1.2
1.4
Employee Count

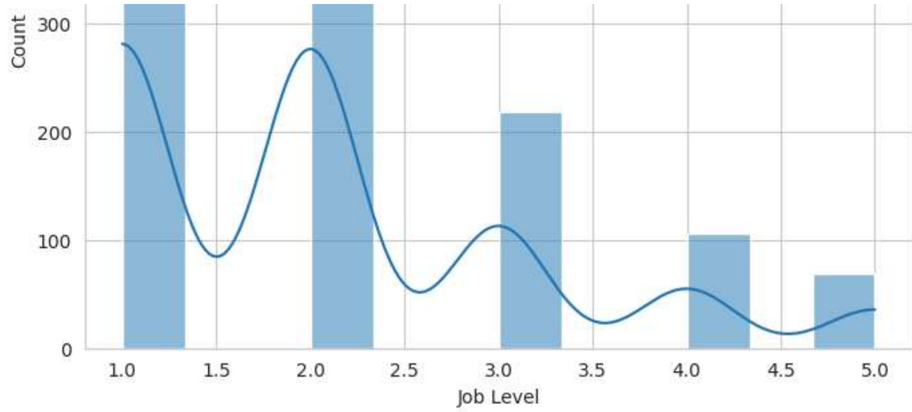

Distribution of Environment Satisfaction
400
300
200
100
Count


0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Environment Satisfaction

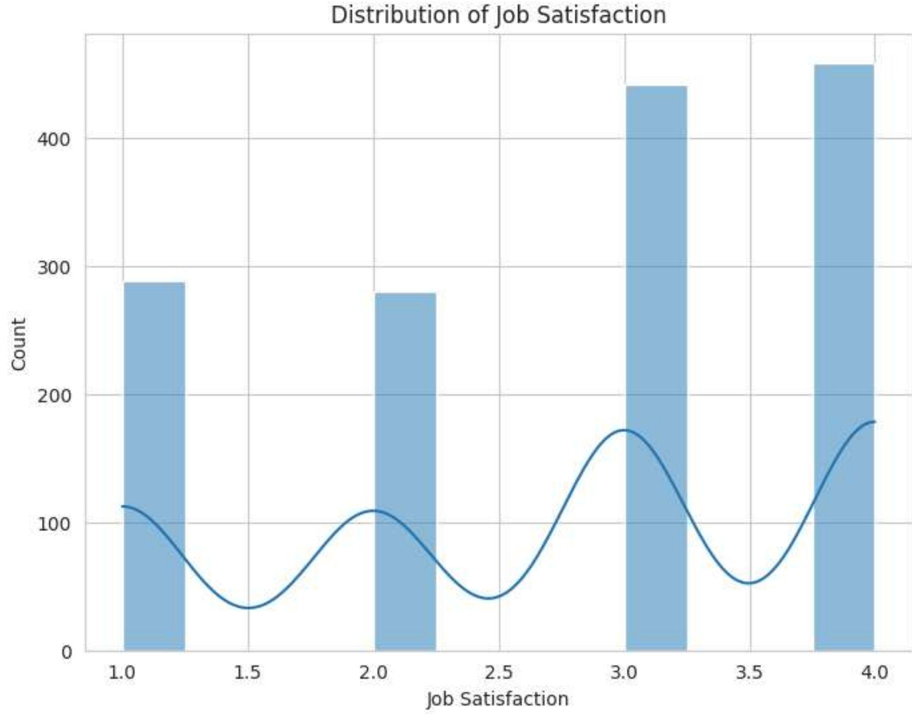

Distribution of Hourly Rate
Count
Hourly Rate
0
20
40
60
80
100
120
140
30
40
50
60
70
80
90
100


Distribution of Job Involvement
Count
Job Involvement
0
200
400
600
800
1.0
1.5
2.0
2.5
3.0
3.5
4.0

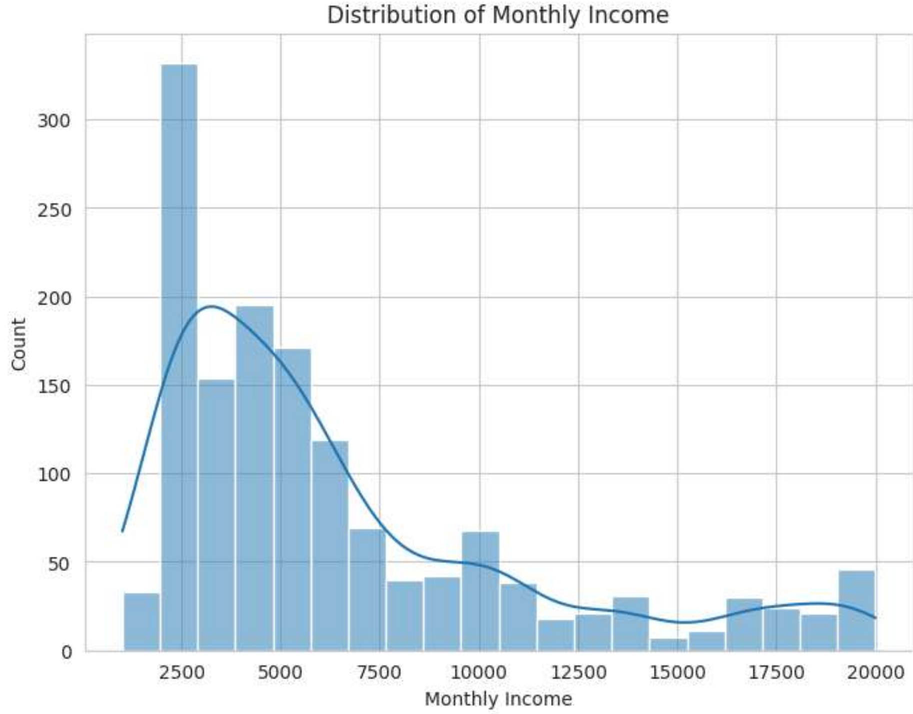

Distribution of Job Level
400
500



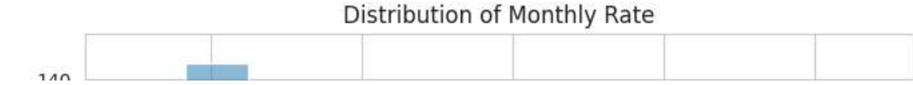
Distribution of Job Satisfaction

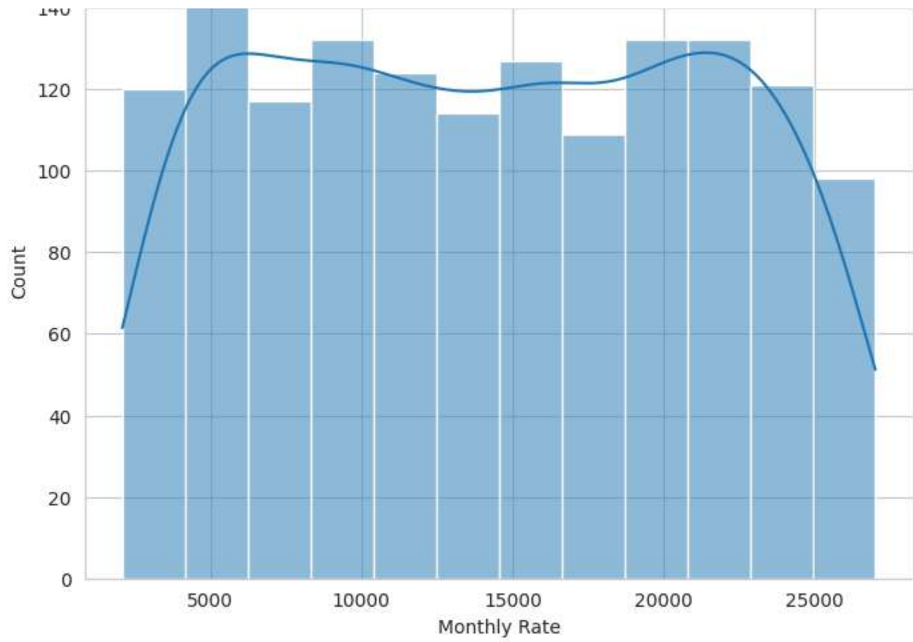


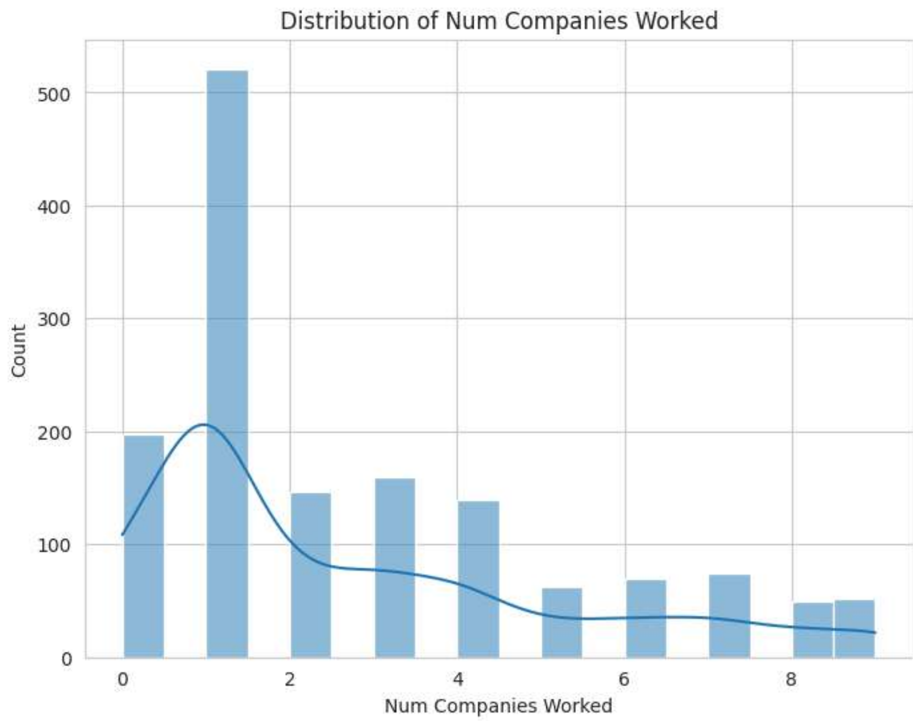
Distribution of Monthly Income



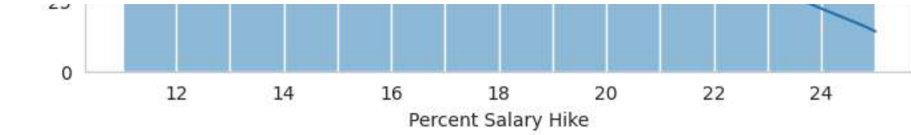
Distribution of Monthly Rate

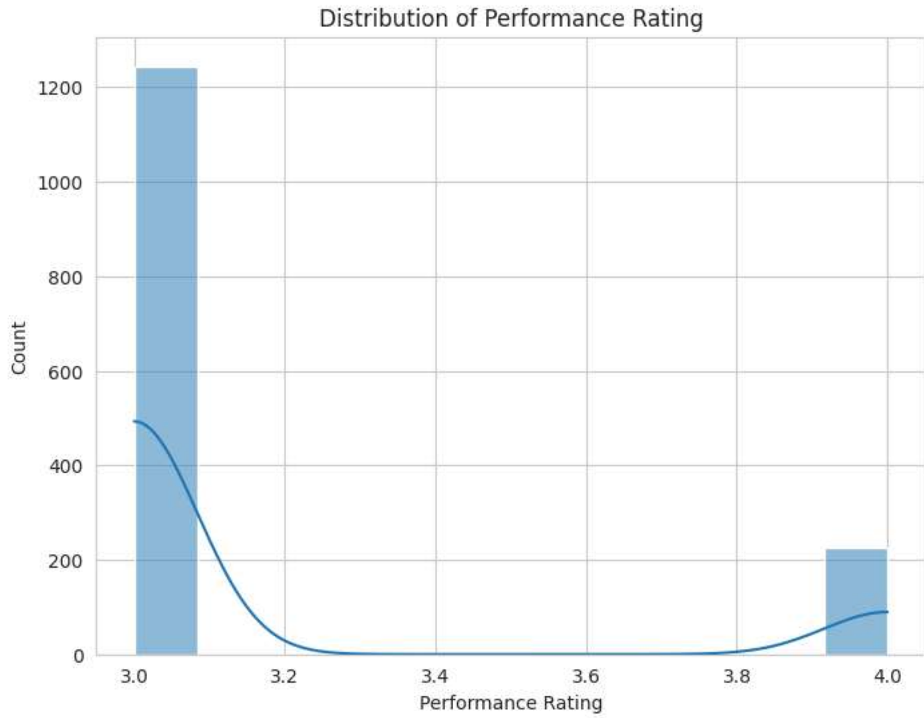


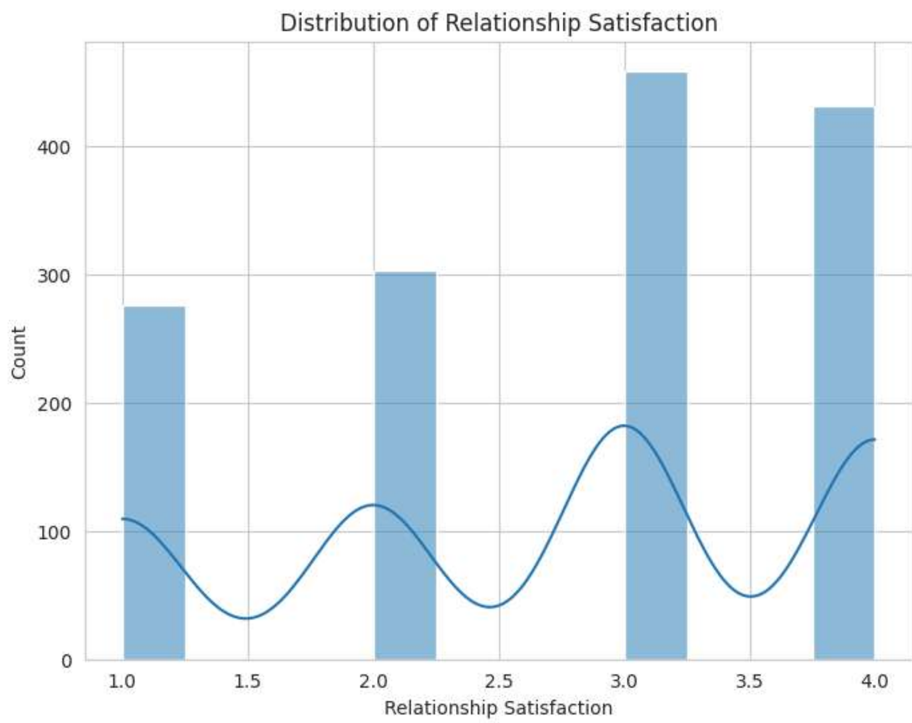

Count
Monthly Rate
0
20
40
60
80
100
120
5000
10000
15000
20000
25000

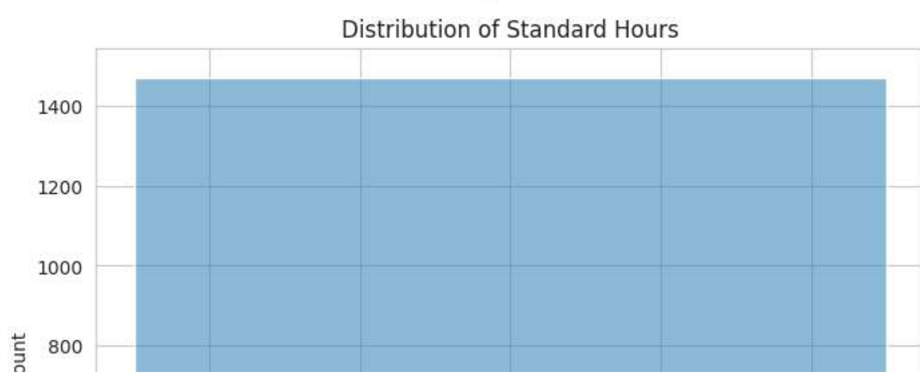

Distribution of Num Companies Worked
Count
Num Companies Worked
0
100
200
300
400
500
0
2
4
6
8

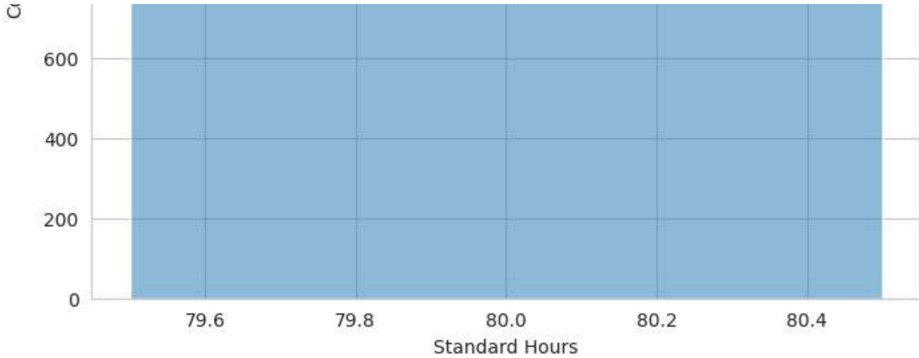

Distribution of Percent Salary Hike
Count
50
75
100
125
150
175
200

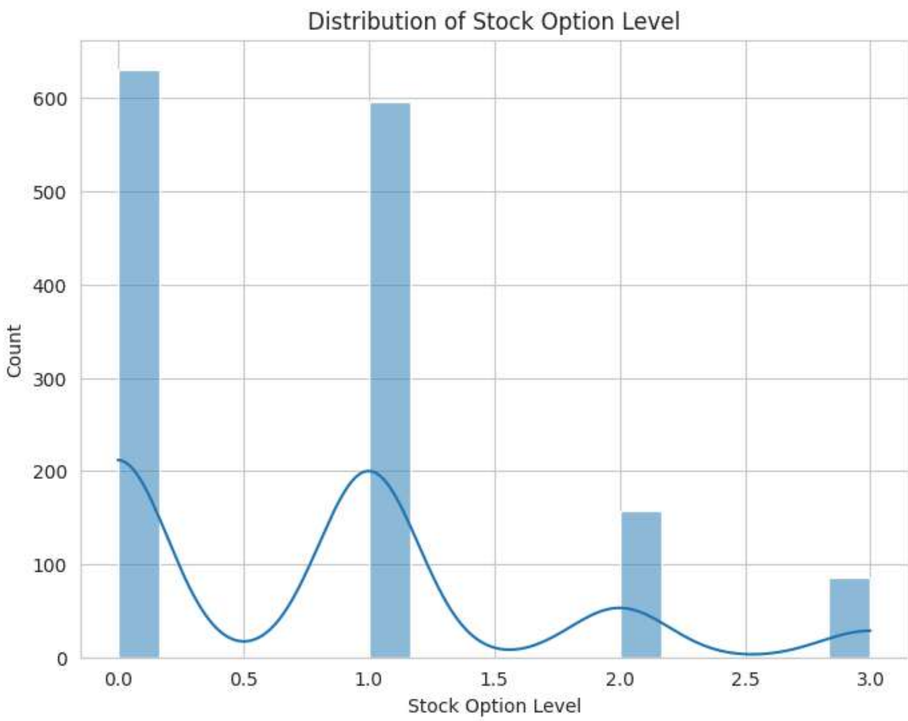

0
12
14
16
18
20
22
24
Percent Salary Hike

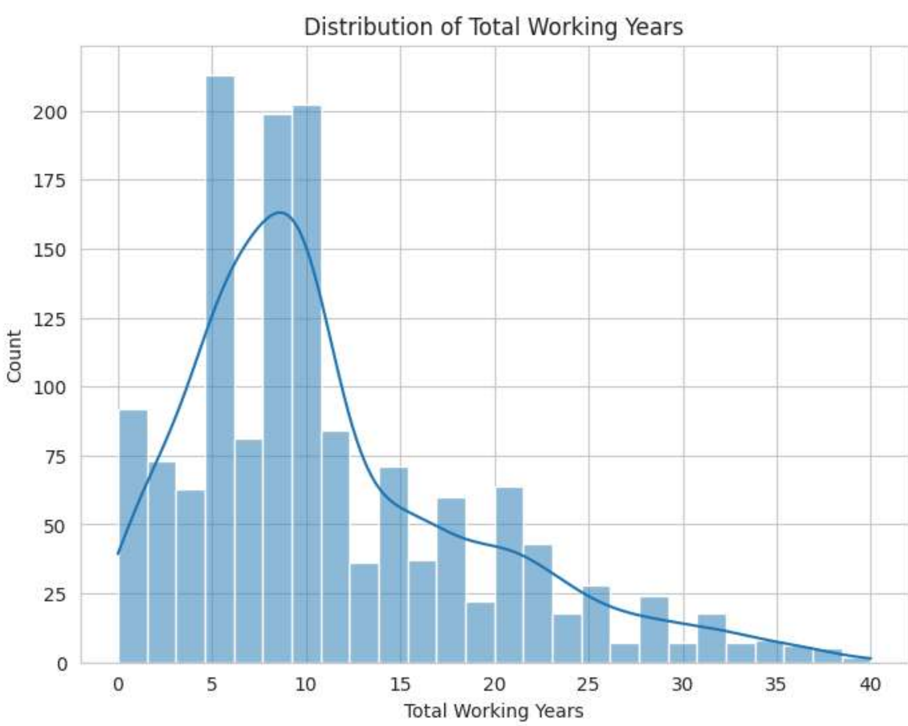

Distribution of Performance Rating
Count
0
200
400
600
800
1000
1200
3.0
3.2
3.4
3.6
3.8
4.0
Performance Rating

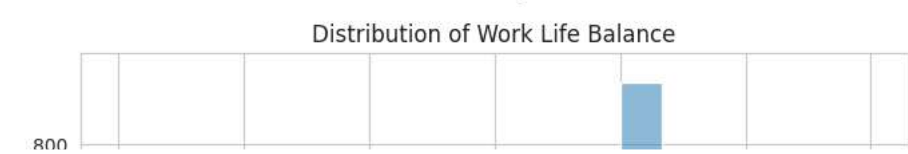

Distribution of Relationship Satisfaction
Count
0
100
200
300
400
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Relationship Satisfaction

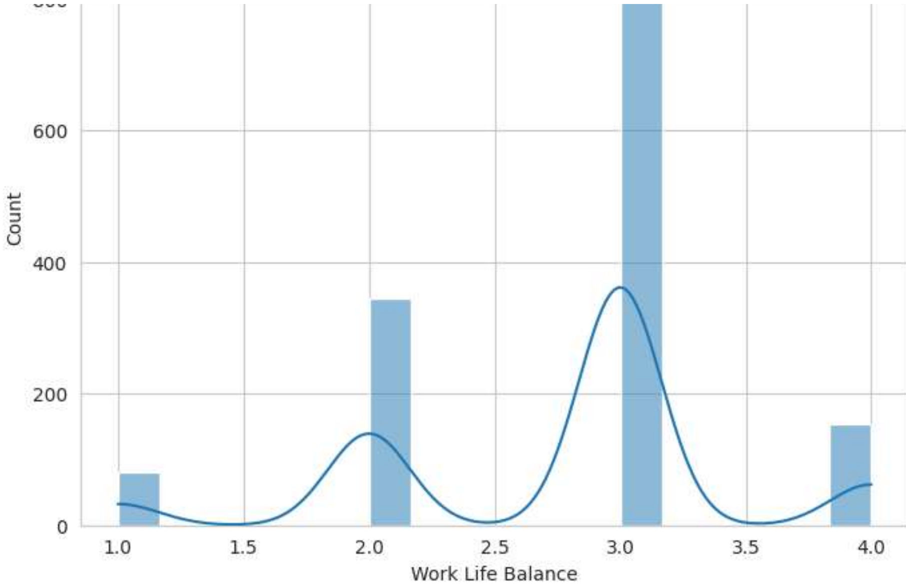

Distribution of Standard Hours
800
1000
1200
1400


600
400
200
0
79.6
79.8
80.0
80.2
80.4
Standard Hours

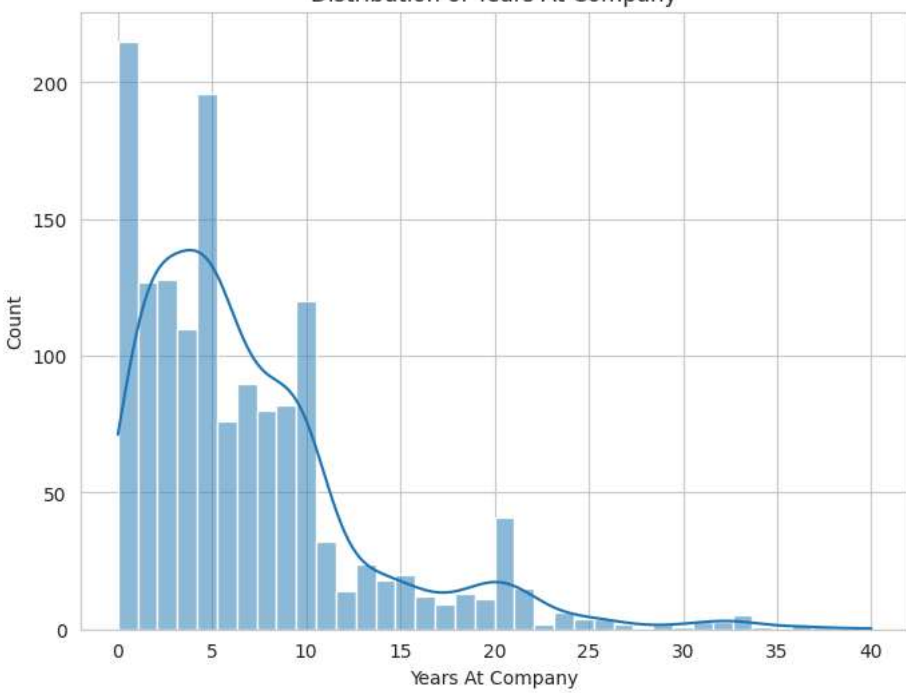

Distribution of Stock Option Level
Count
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Stock Option Level


Distribution of Total Working Years
Count
200
175
150
125
100
75
50
25
0
0
5
10
15
20
25
30
35
40
Total Working Years

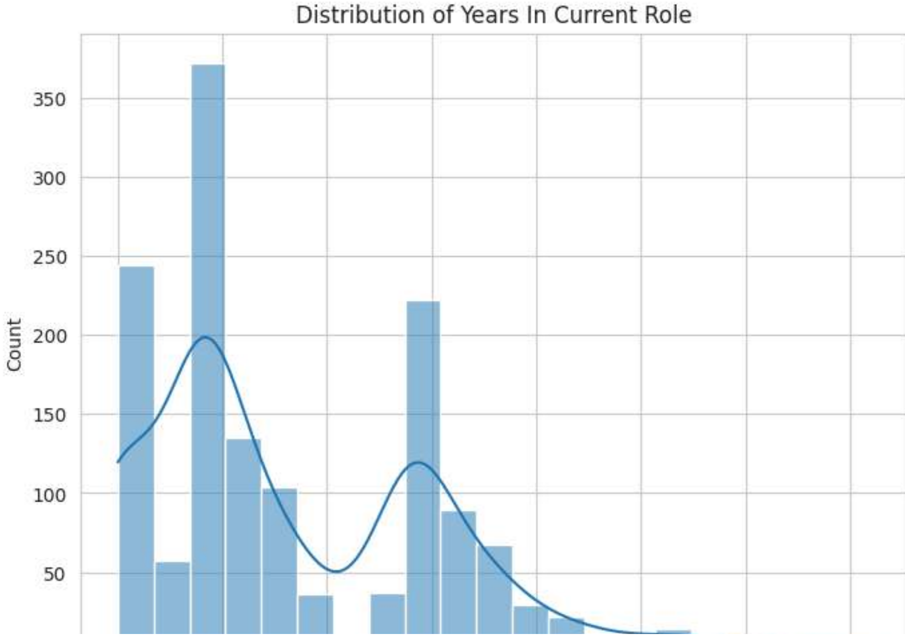

Distribution of Work Life Balance
800



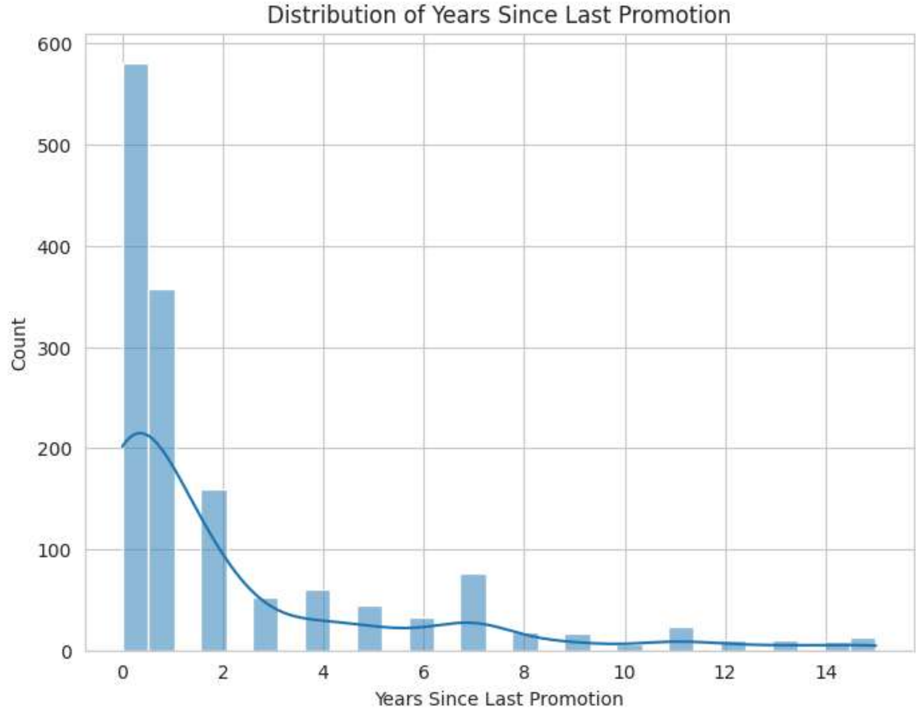
Distribution of Years At Company

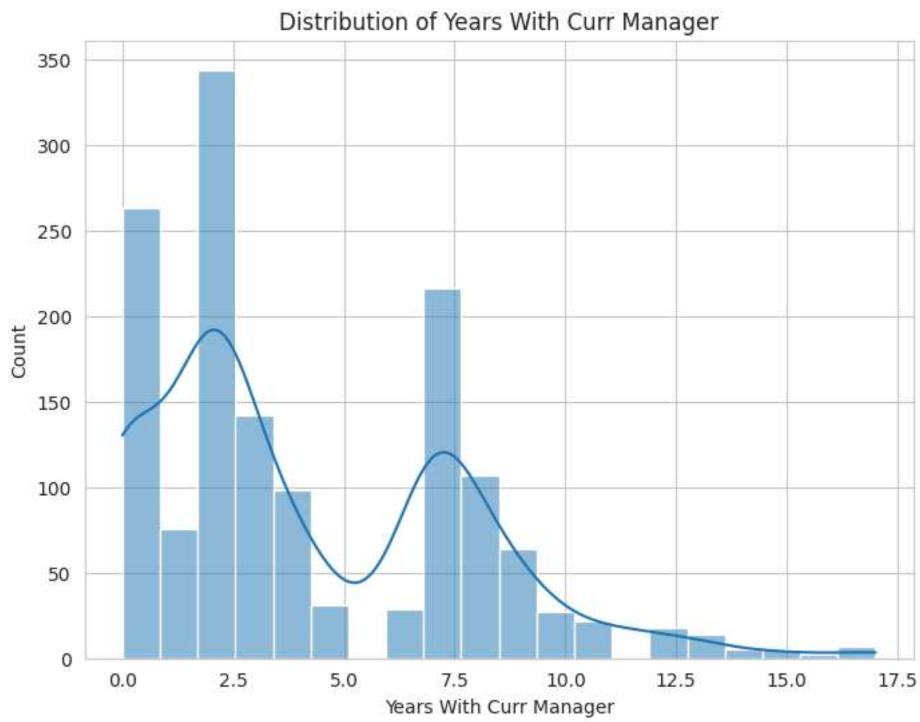


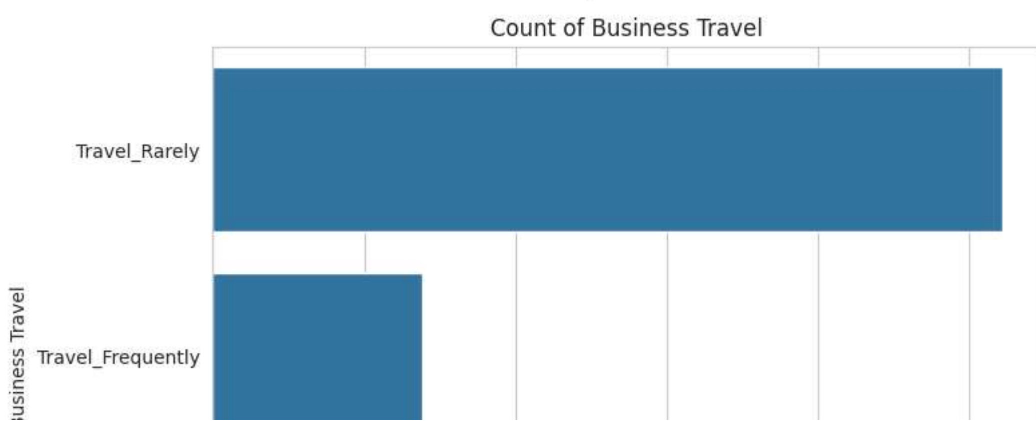
Distribution of Years In Current Role

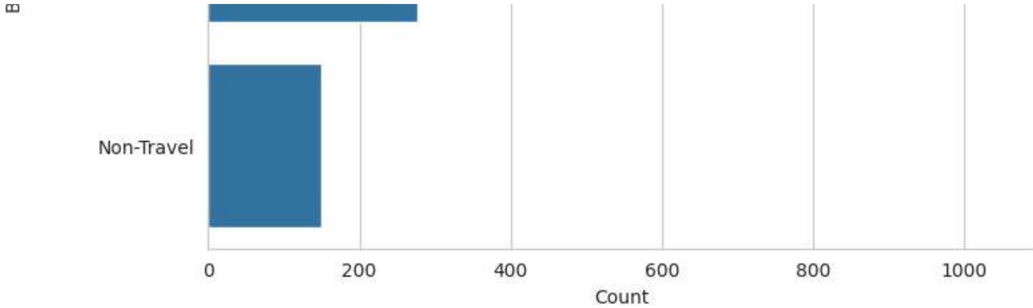


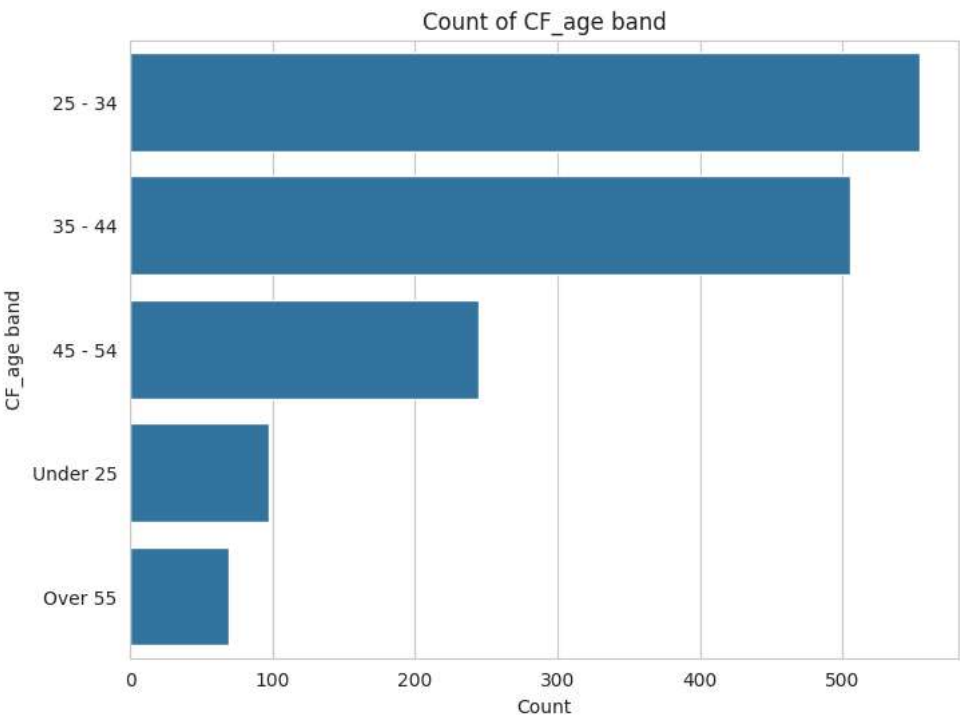

0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Years In Current Role

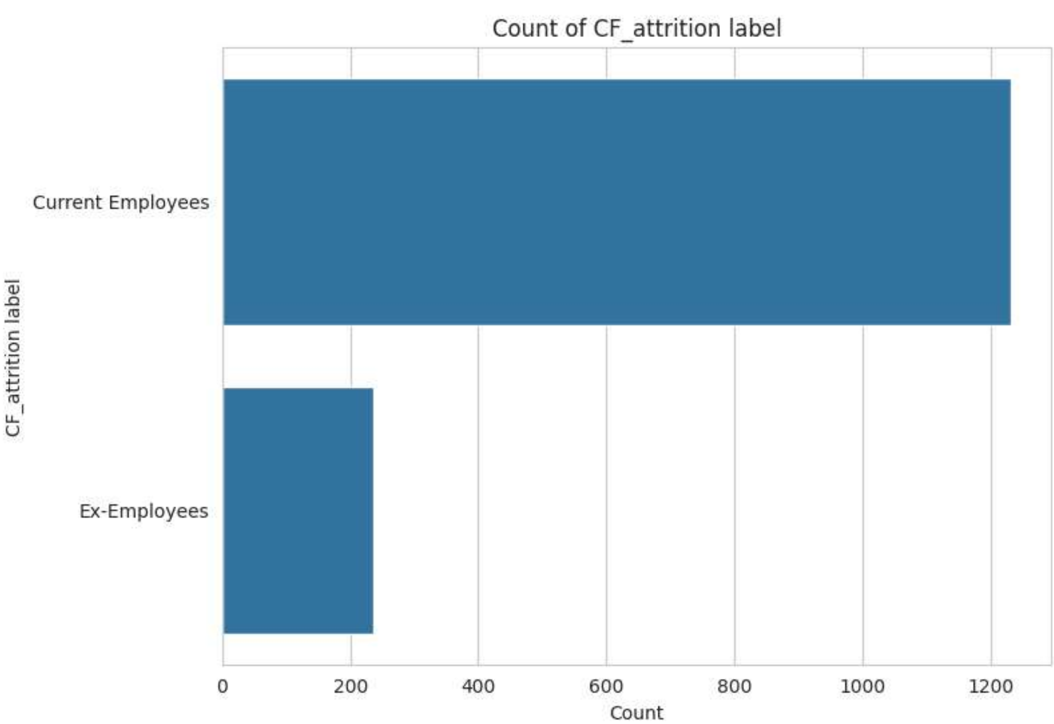

Distribution of Years Since Last Promotion
Count
Years Since Last Promotion

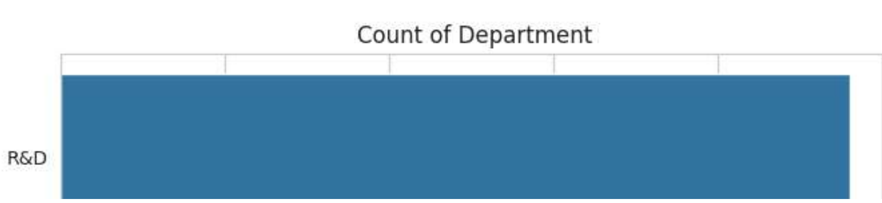

Distribution of Years With Curr Manager
Count
Years With Curr Manager

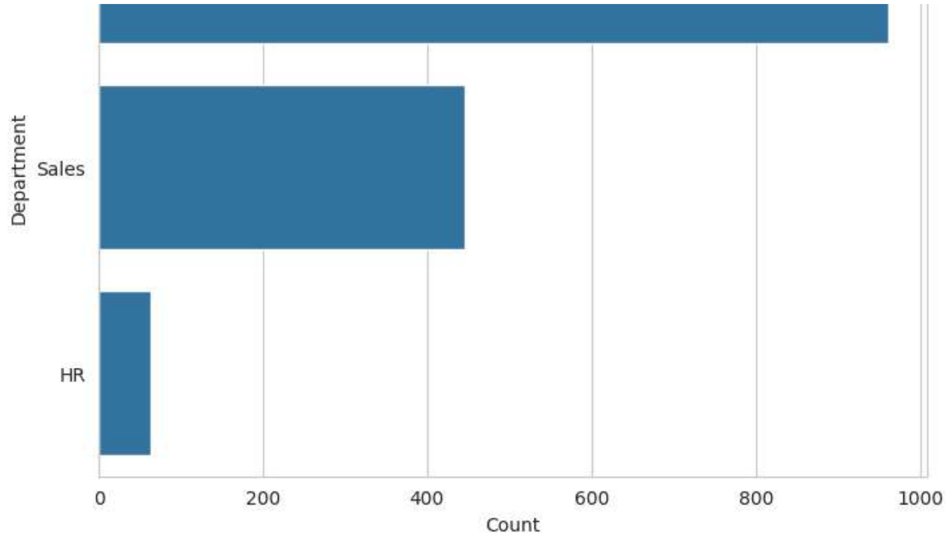

Count of Business Travel
Travel_Rarely
Travel_Frequently
usiness Travel

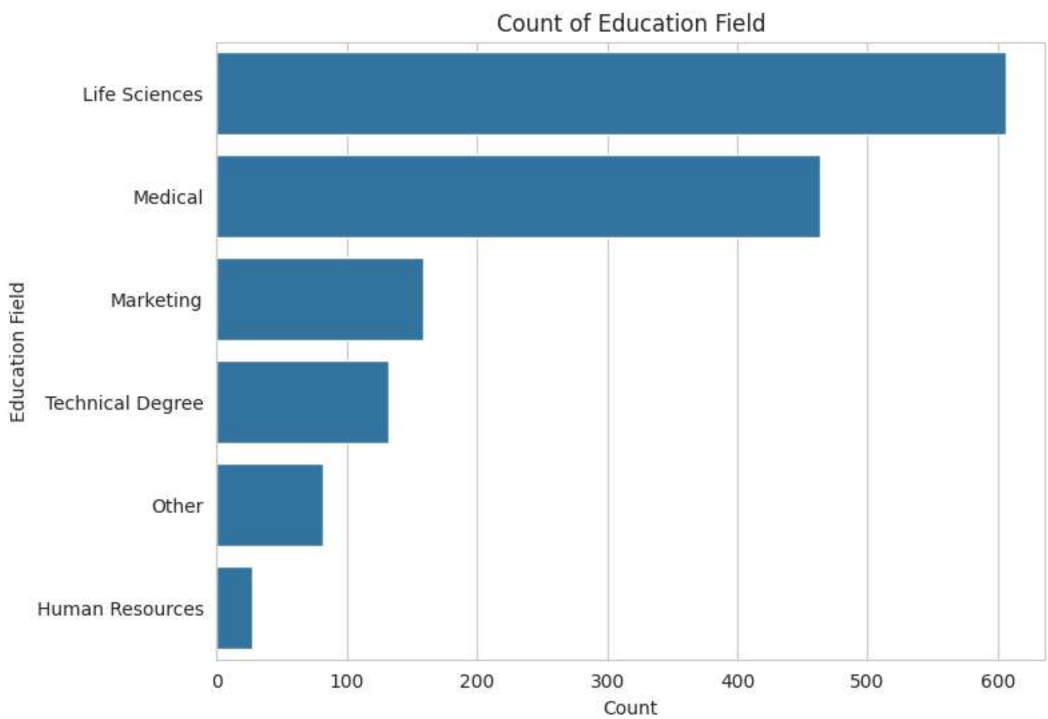

B
Non-Travel
0
200
400
600
800
1000
Count

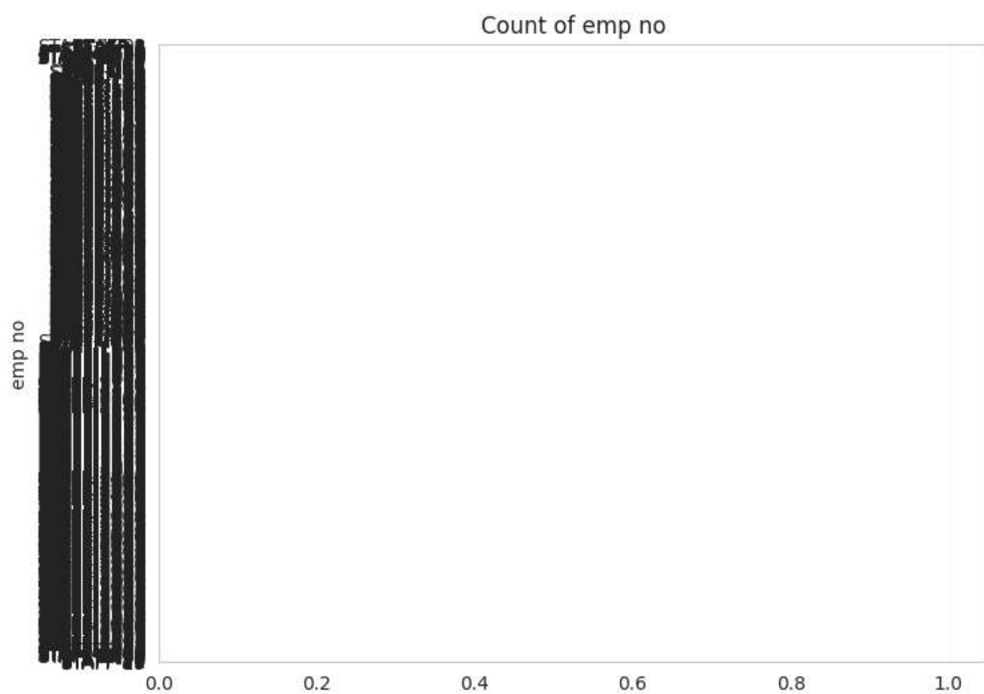

Count of CF_age band
CF_age band
25 - 34
35 - 44
45 - 54
Under 25
Over 55
0
100
200
300
400
500
Count


Count of CF_attrition label
CF_attrition label
Current Employees
Ex-Employees
0
200
400
600
800
1000
1200
Count

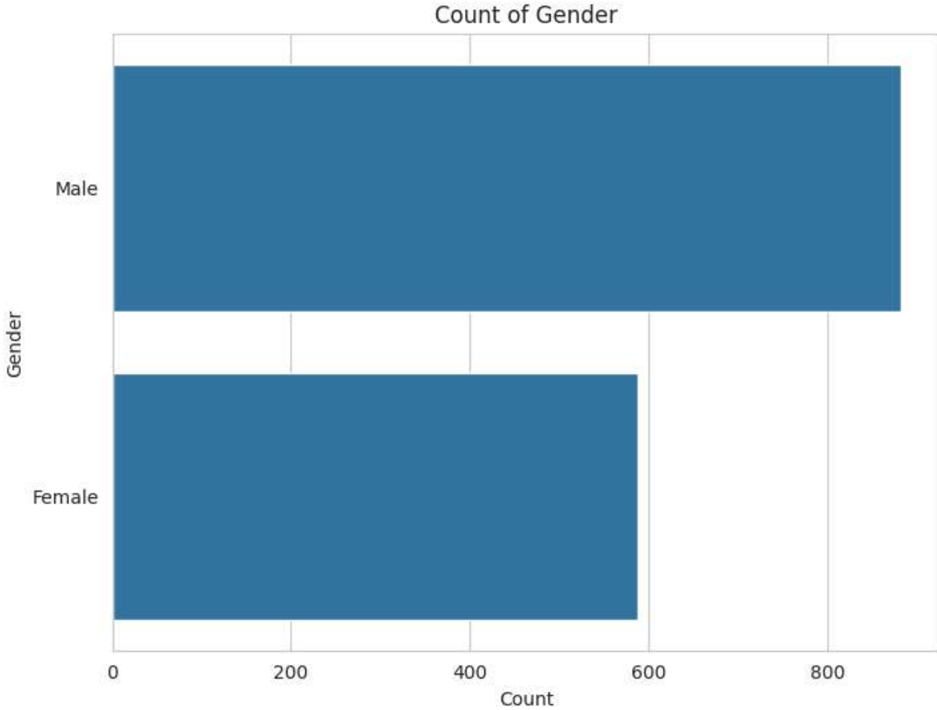

Count of Department
R&D

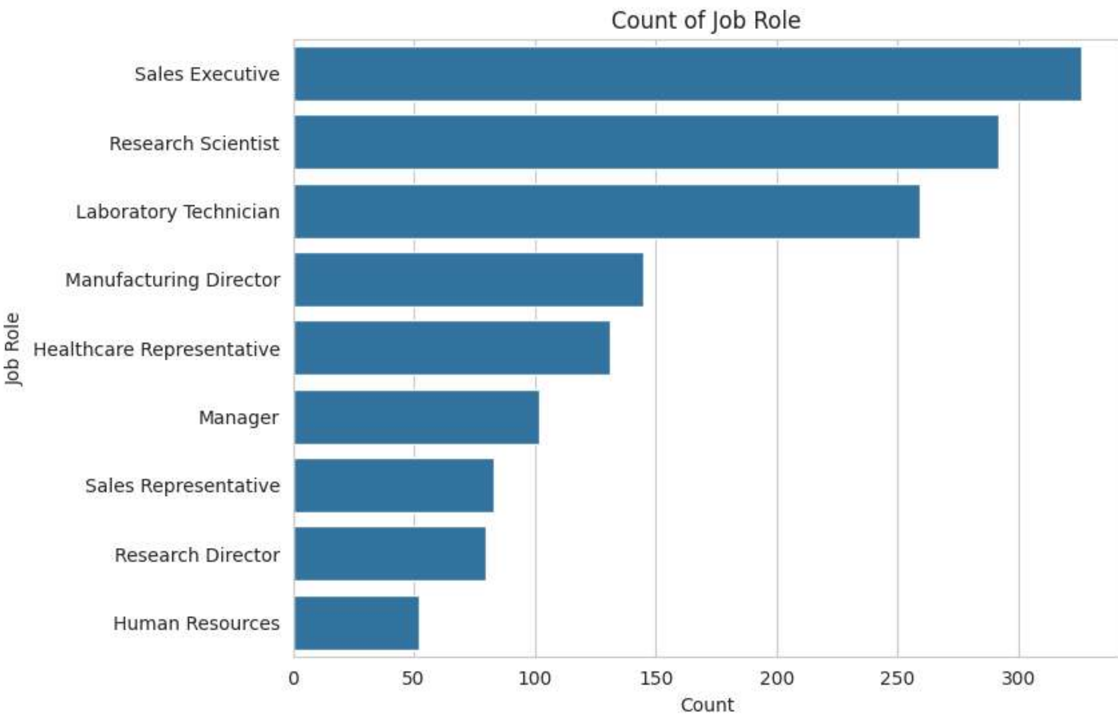

Department
Sales
HR
0
200
400
600
800
1000
Count

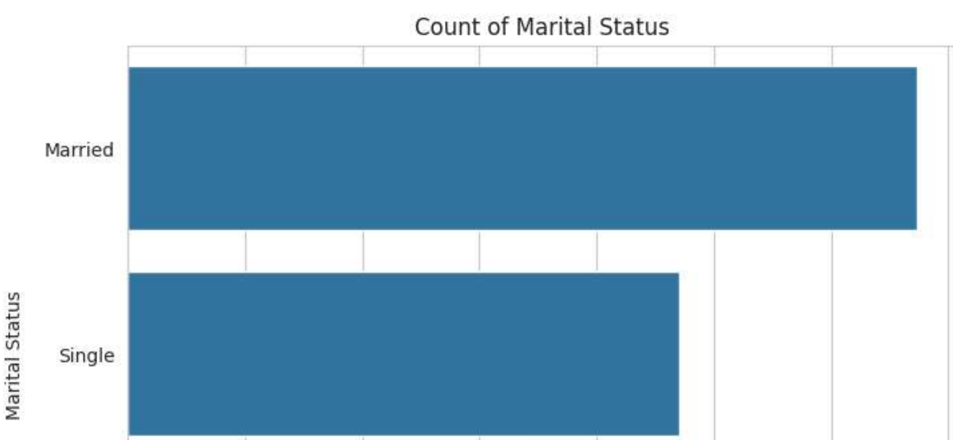

Count of Education Field
Education Field
Life Sciences
Medical
Marketing
Technical Degree
Other
Human Resources
0
100
200
300
400
500
600
Count

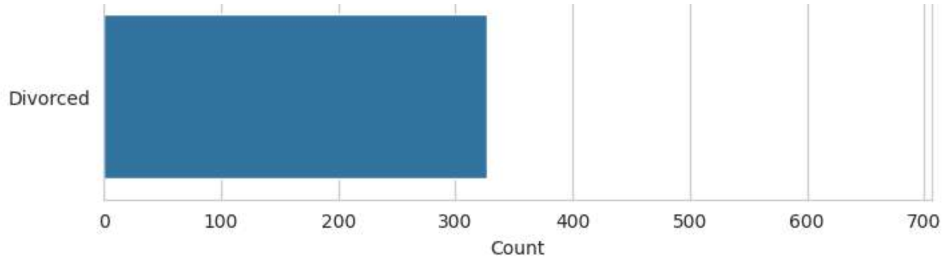

Count of emp no
emp no
0.0
0.2
0.4
0.6
0.8
1.0

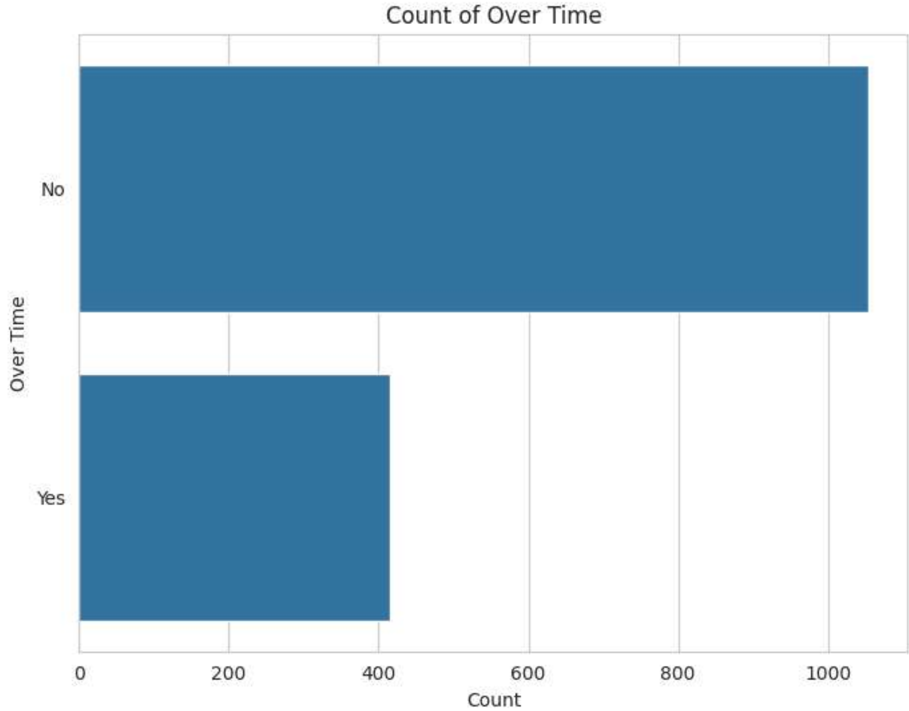
Count

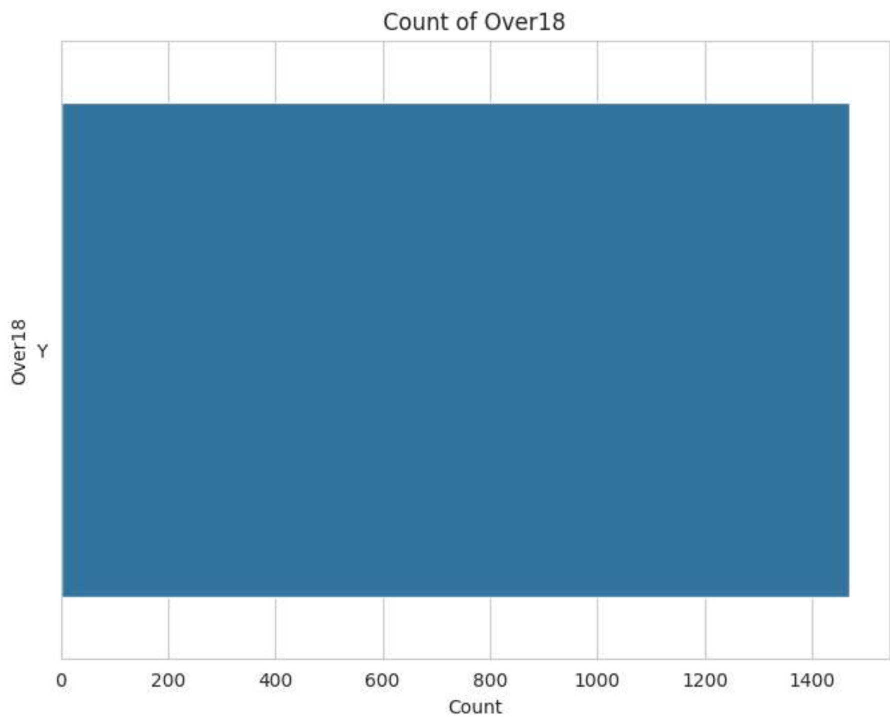

Count of Gender
Gender
Male
Female
0
200
400
600
800
Count

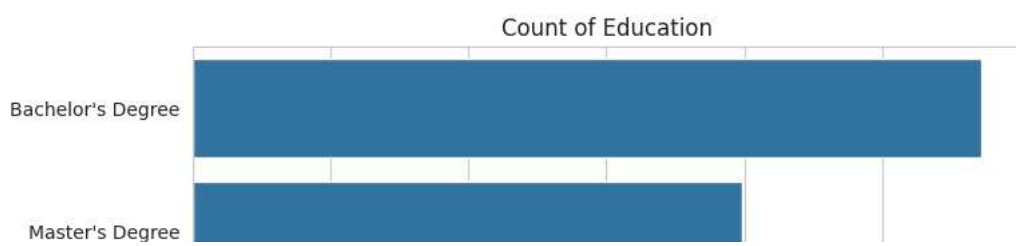

Count of Job Role
Job Role
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources
0
50
100
150
200
250
300
Count

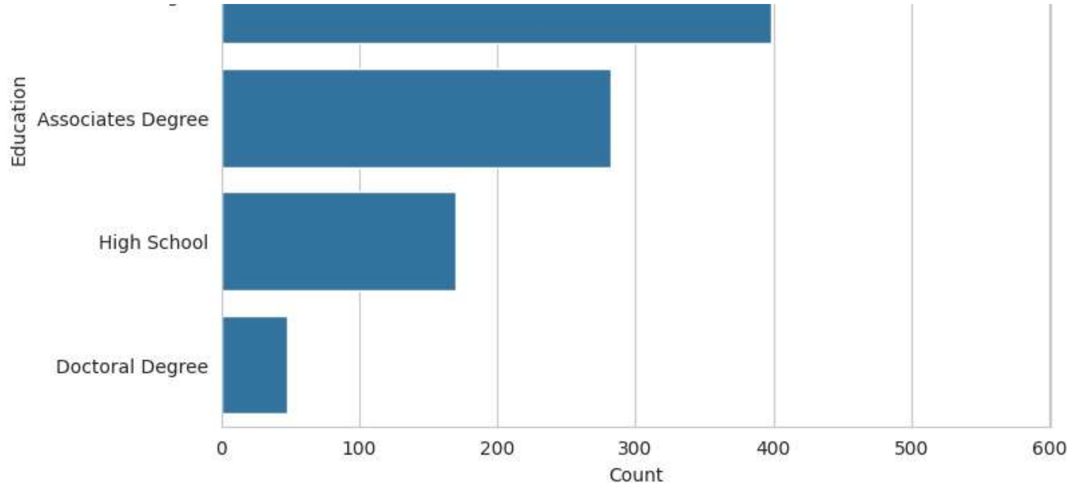

Count of Marital Status
Marital Status
Married
Single

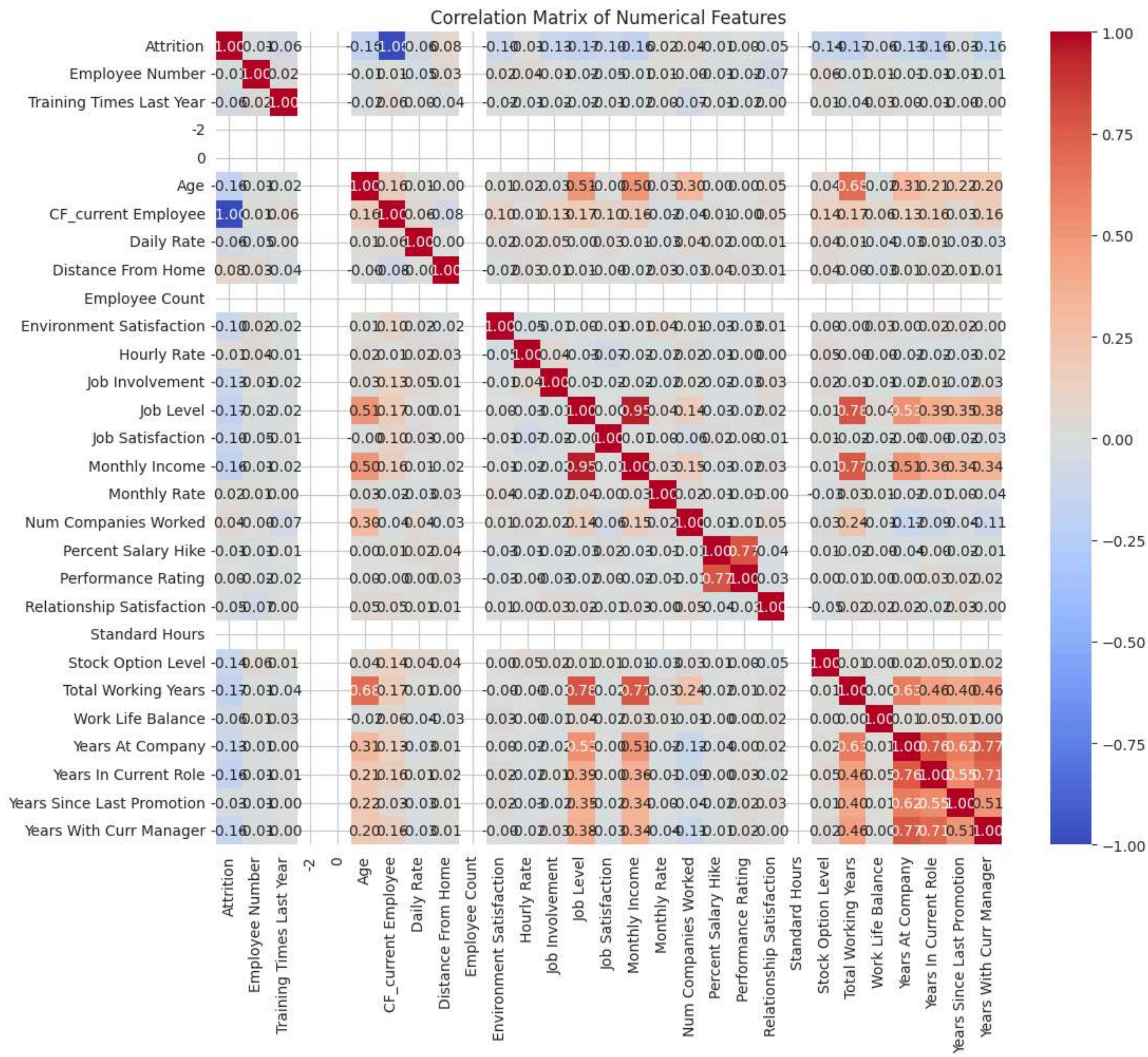

Divorced
0
100
200
300
400
500
600
700
Count


Count of Over Time
No
Yes
Over Time
0
200
400
600
800
1000
Count

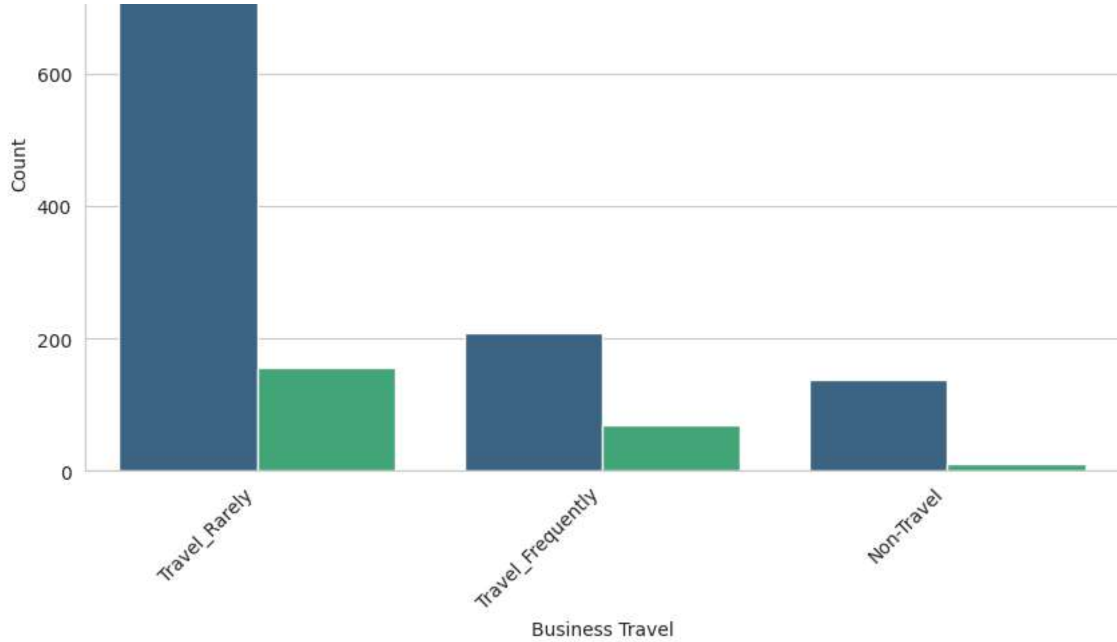

Count of Over18
Y
Over18
0
200
400
600
800
1000
1200
1400
Count


Count of Education
Bachelor's Degree
Master's Degree

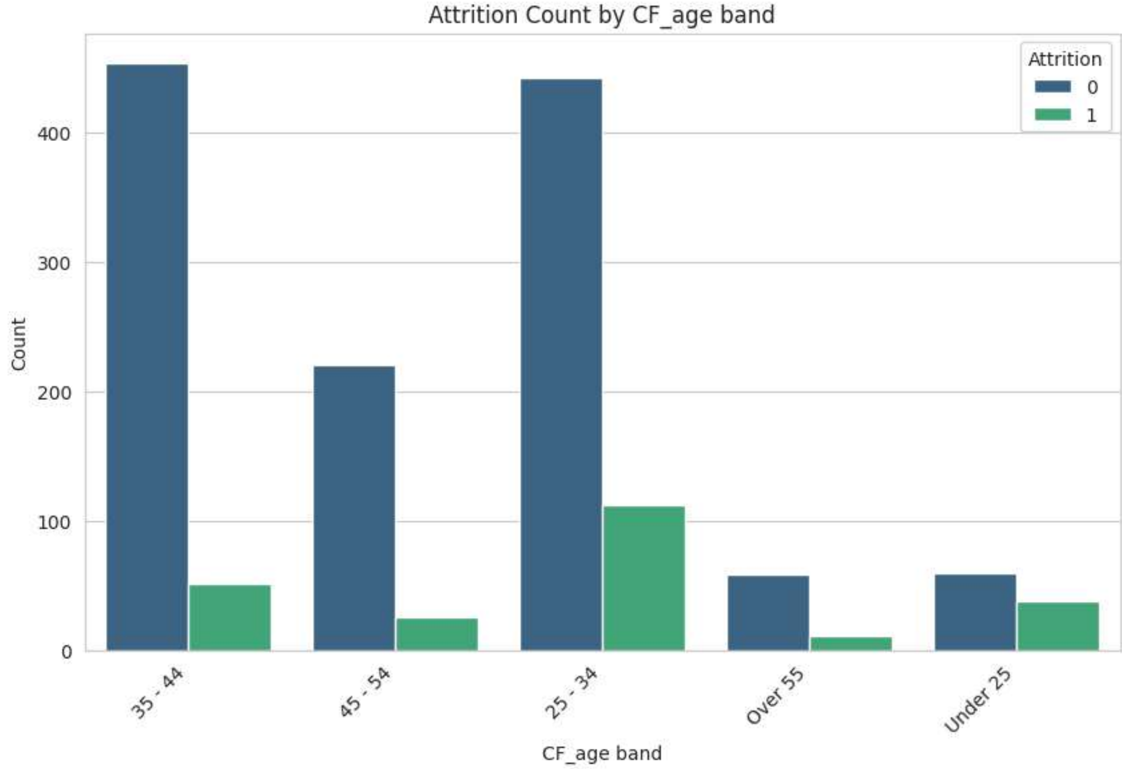

Education
Associates Degree
High School
Doctoral Degree
0
100
200
300
400
500
600
Count


Correlation Matrix of Numerical Features
Attrition
Employee Number
Training Times Last Year
-2
0
Age
CF_current Employee
Daily Rate
Distance From Home
Employee Count
Environment Satisfaction
Hourly Rate
Job Involvement
Job Level
Job Satisfaction
Monthly Income
Monthly Rate
Num Companies Worked
Percent Salary Hike
Performance Rating
Relationship Satisfaction
Standard Hours
Stock Option Level
Total Working Years
Work Life Balance
Years At Company
Years In Current Role
Years Since Last Promotion
Years With Curr Manager
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

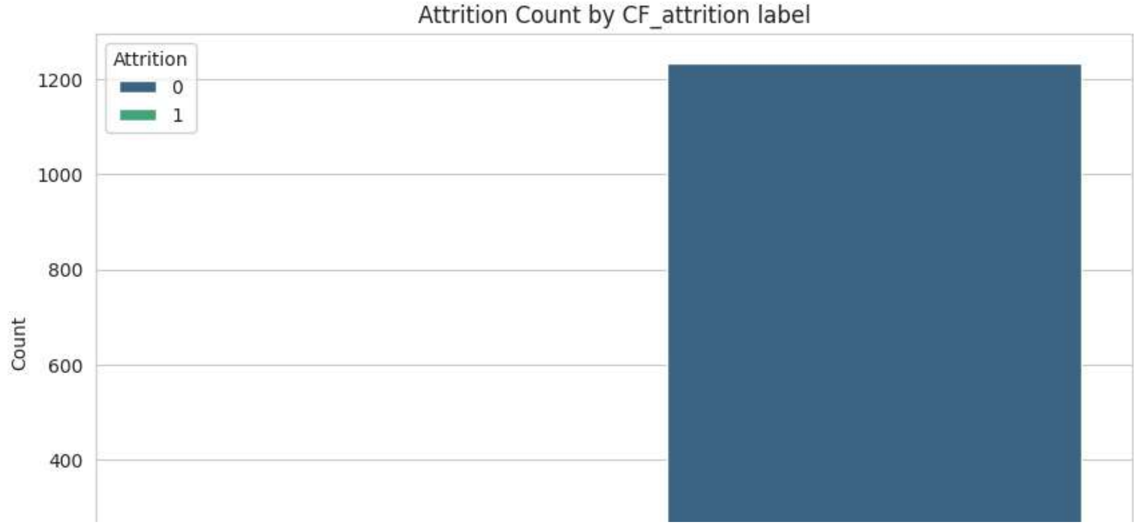

Attrition Count by Business Travel
800
Attrition
0
1

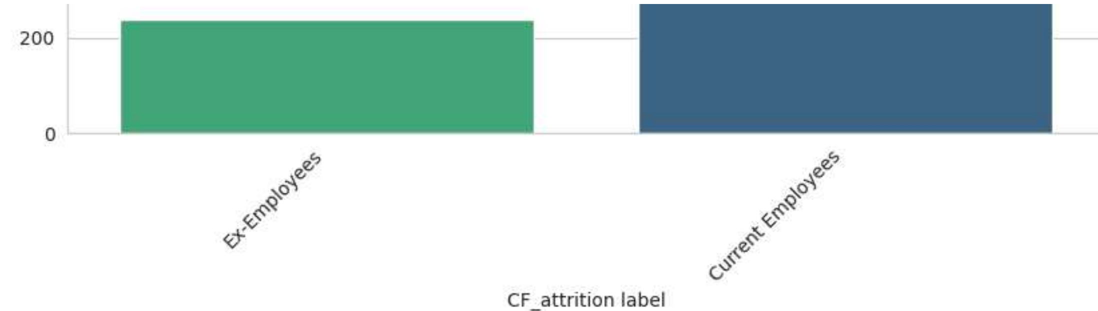


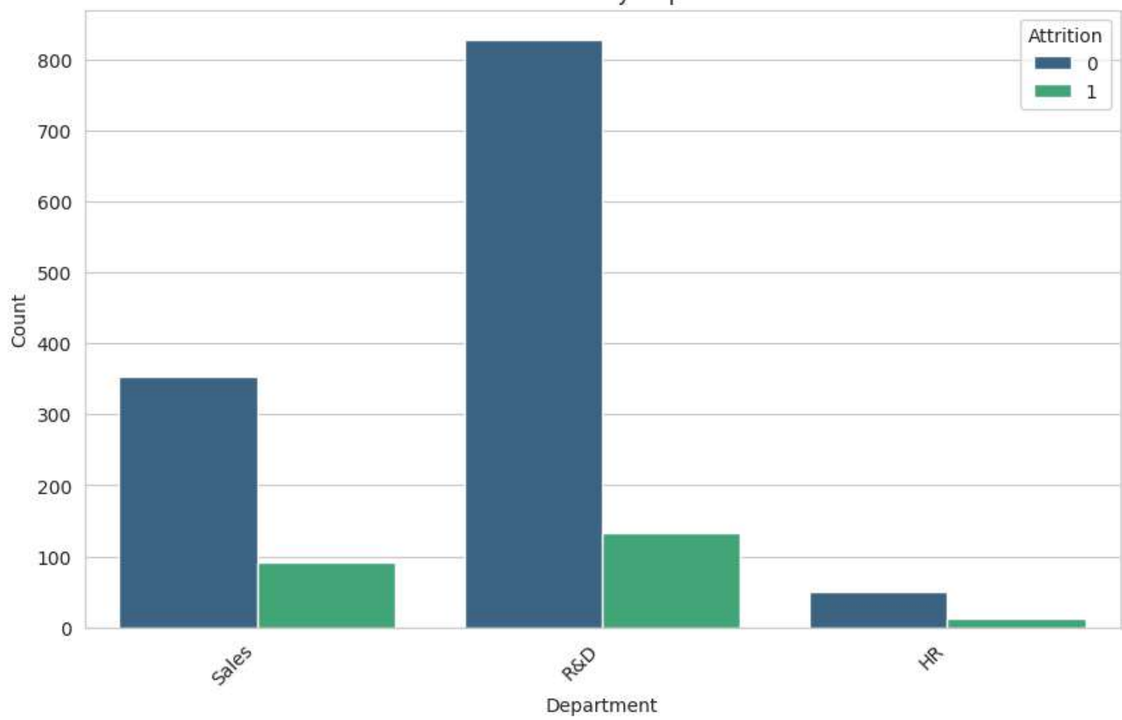
Attrition Count by CF_age band

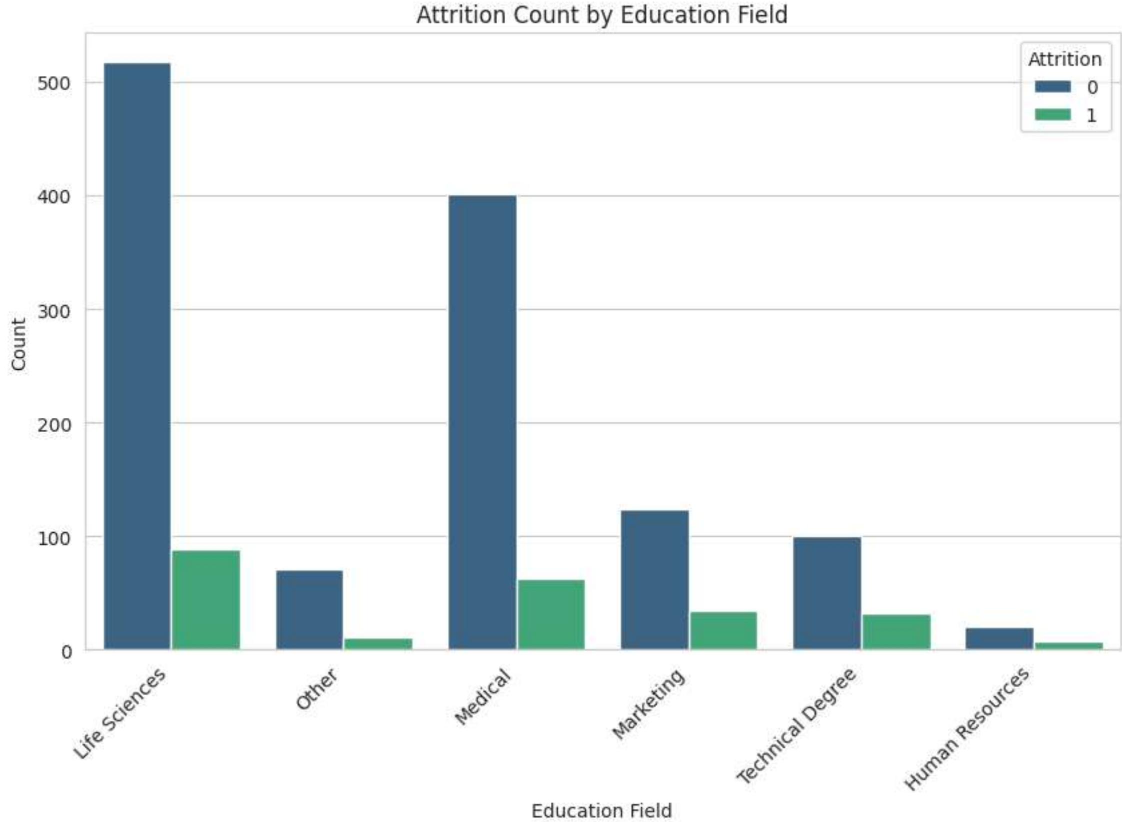


Attrition Count by CF_attrition label

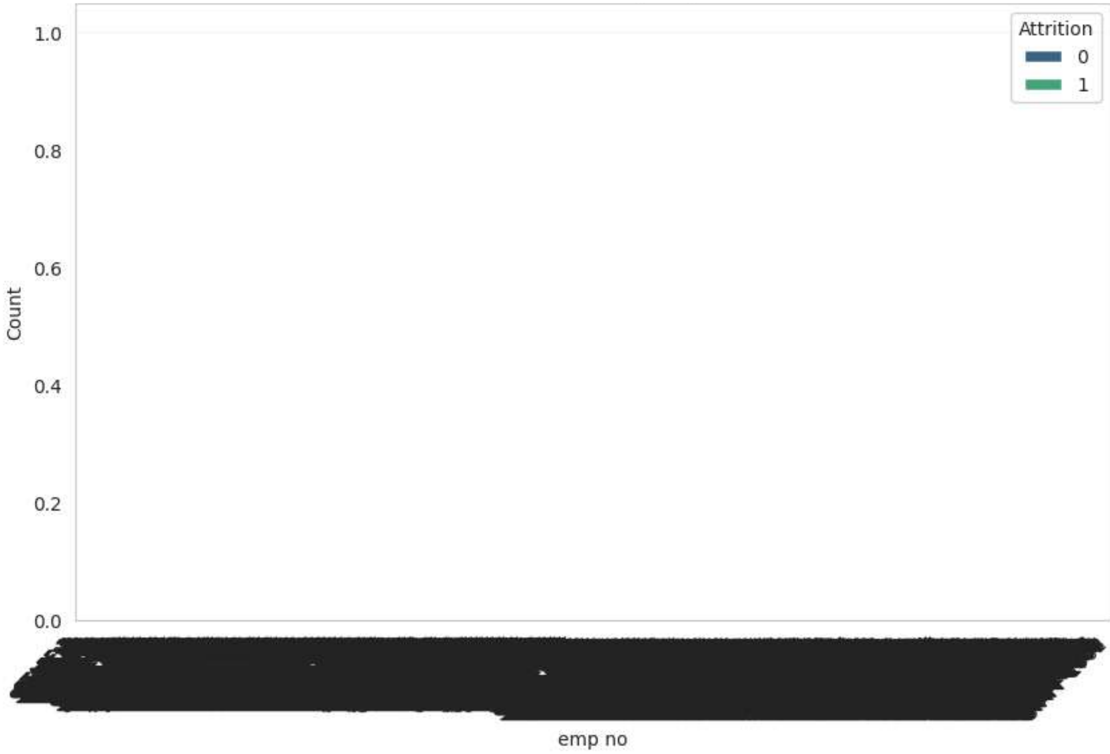


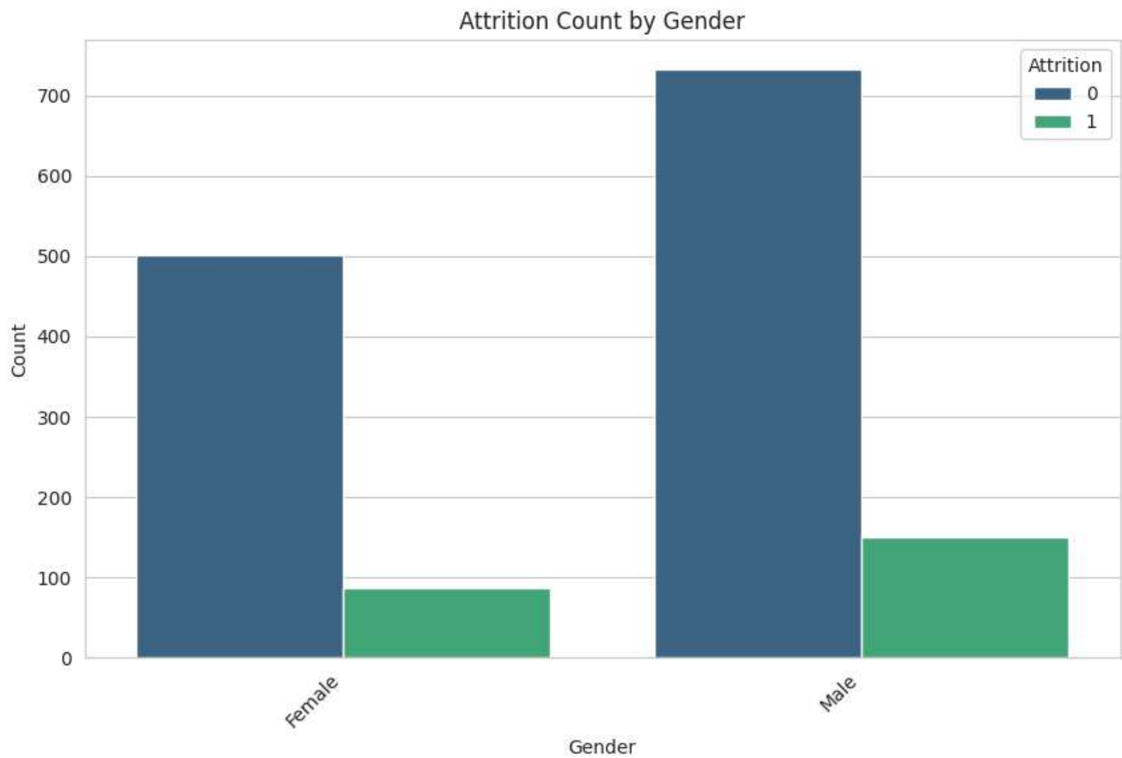


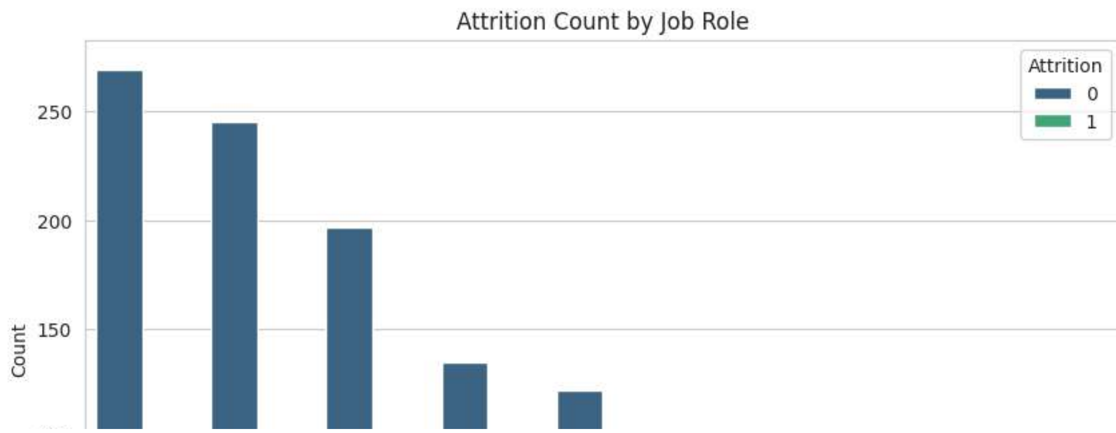


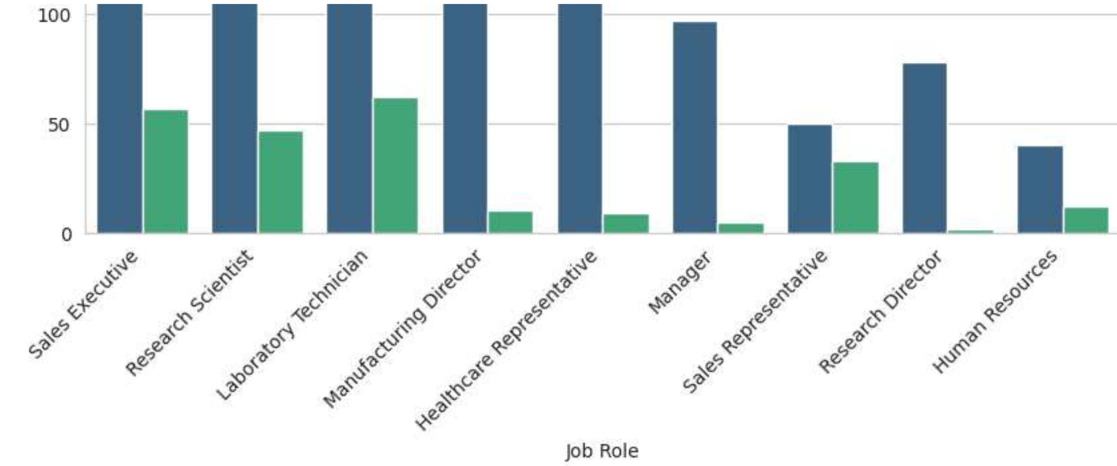


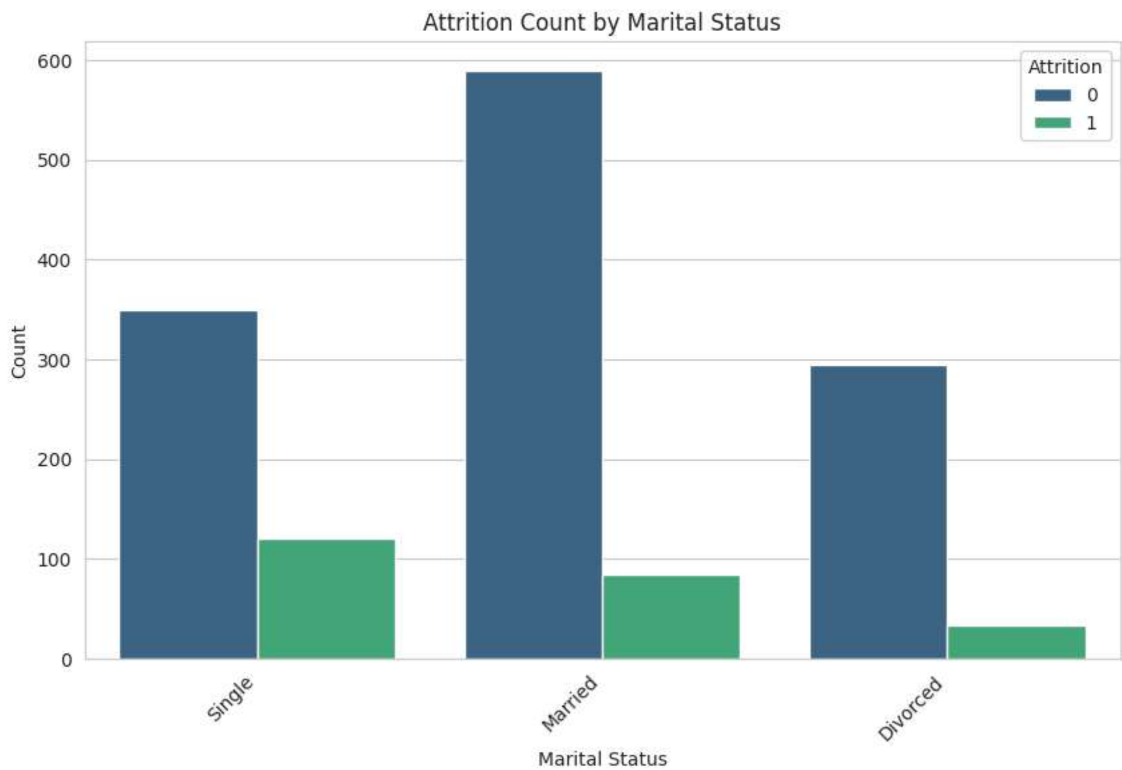
Attrition Count by emp no

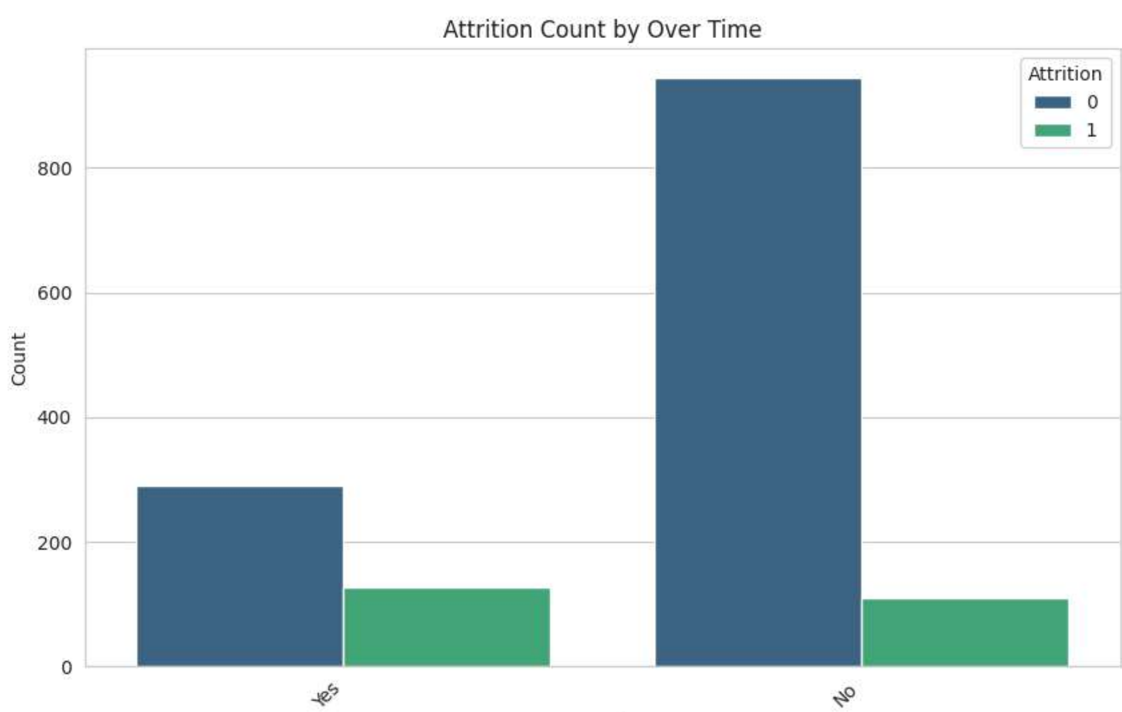

Attrition
0
1
1.0
0.8
0.6
0.4
0.2
0.0
Count
emp no


Attrition Count by Gender
Attrition
0
1
700
600
500
400
300
200
100
0
Count
Female
Male
Gender

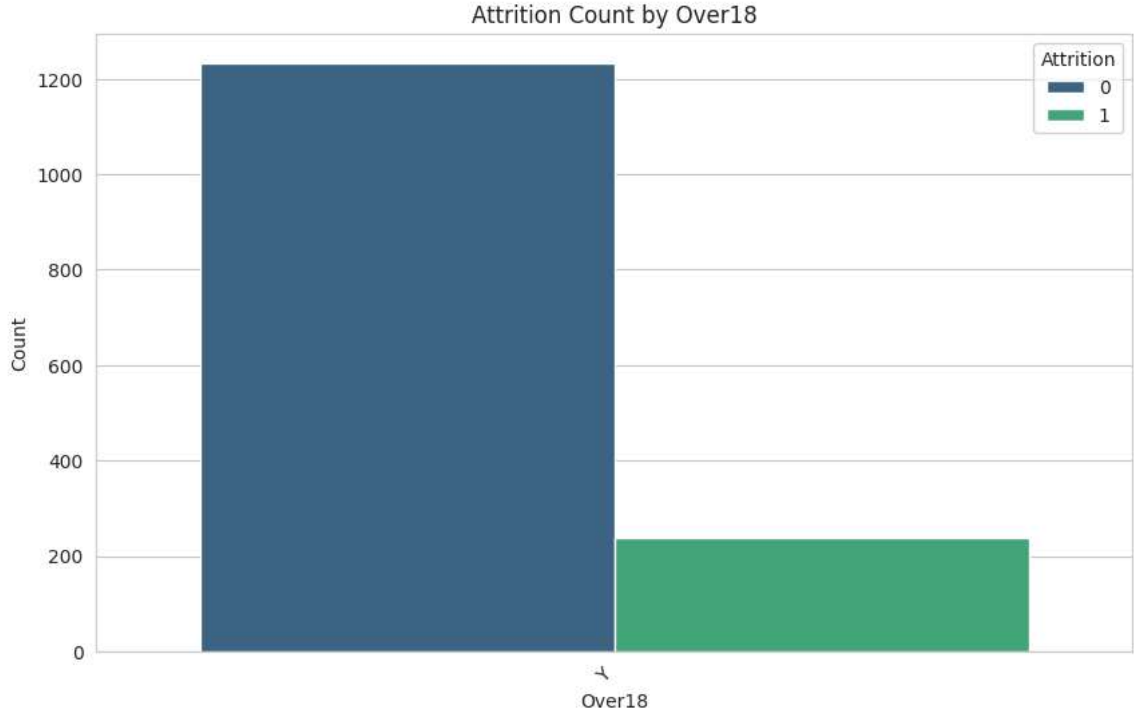

Attrition Count by Job Role
Attrition
0
1
250
200
150
Count

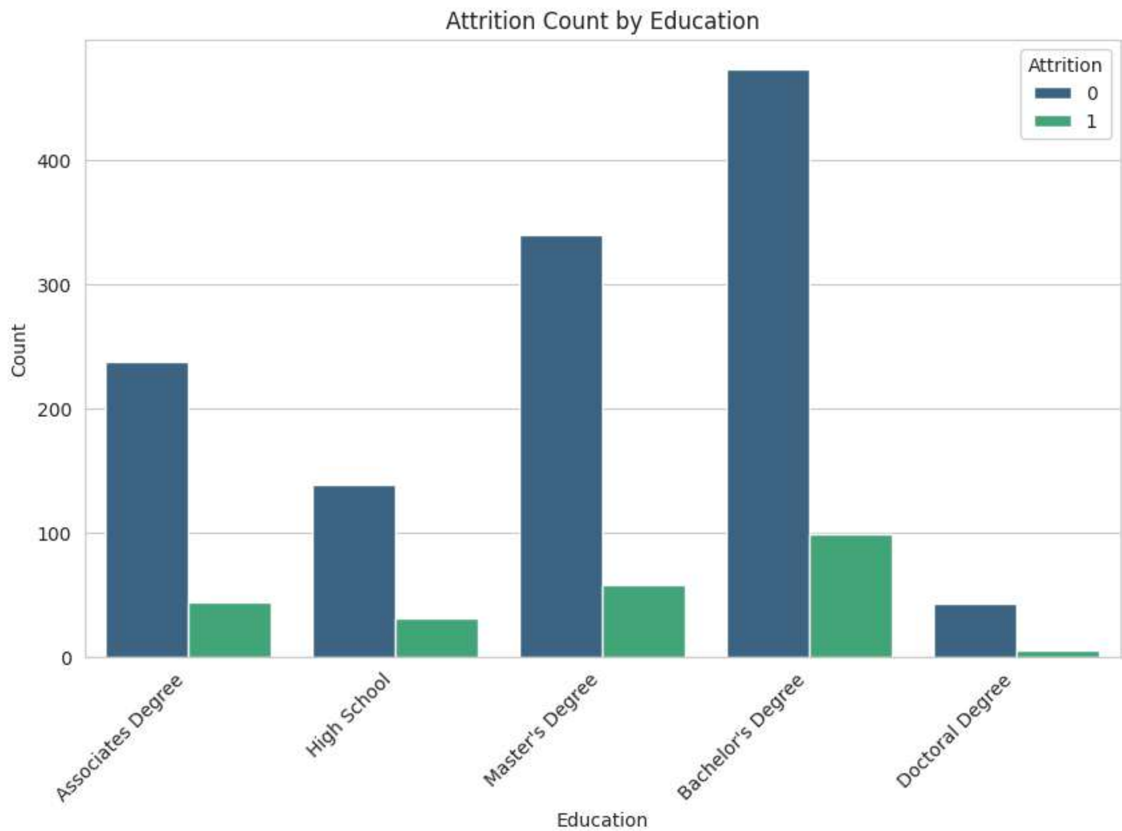

100
50
0
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources
Job Role


Attrition Count by Marital Status
Attrition
0
1
600
500
400
300
200
100
0
Count
Single
Married
Divorced
Marital Status


Attrition Count by Over Time
Attrition
0
1
800
600
400
200
0
Count
Yes
No

Over Time


Attrition Count by Over18
Attrition
0
1
1200
1000
800
600
400
200
0
Count
Y
Over18


Attrition Count by Education
Attrition
0
1
400
300
200
100
0
Count
Associates Degree
High School
Master's Degree
Bachelor's Degree
Doctoral Degree
Education

```
import shap

X_train_processed = model.named_steps['preprocessor'].transform(X_train)

# Get the feature names after preprocessing (including one-hot encoded categories)
# This can be a bit tricky with ColumnTransformer and one-hot encoding
# Here's a way to get feature names for one-hot encoded features:
ohe_feature_names = model.named_steps['preprocessor'].named_transformers_['cat'].get_feature_names_out(categorical_features)
all_feature_names = list(numerical_features) + list(ohe_feature_names)

# Create an explainer object
# For tree-based models (like Decision Tree), you'd use shap.TreeExplainer
# For linear models (like Logistic Regression), shap.LinearExplainer is appropriate
explainer = shap.LinearExplainer(model.named_steps['classifier'], X_train_processed)

# Calculate SHAP values for the test set
shap_values = explainer.shap_values(model.named_steps['preprocessor'].transform(X_test))

# Visualize SHAP values
# Summary plot
shap.summary_plot(shap_values, model.named_steps['preprocessor'].transform(X_test), feature_names=all_feature_names)
```