HR Employee Attrition

Import libraries

```
import numpy as np
import pandas as pd
#import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

df = pd.read_csv('HR-Employee-Attrition.csv')
```

```
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
```

Data Validation and Cleaning

```
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.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	J
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

```
df.duplicated().sum()
```

```
0
```

No duplicates

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

```
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
```

No null values.

```
#Unique values in object type column
object_columns = df.select_dtypes(include='object').columns
unique_values = {}
for column in object_columns:
    unique_values[column] = df[column].unique()
pd.Series(unique_values)
```

```
Attrition                                                [Yes, No]
BusinessTravel       [Travel_Rarely, Travel_Frequently, Non-Travel]
Department         [Sales, Research & Development, Human Resources]
EducationField    [Life Sciences, Other, Medical, Marketing, Tec...
Gender                                                 [Female, Male]
JobRole           [Sales Executive, Research Scientist, Laborato...
MaritalStatus                          [Single, Married, Divorced]
Over18                                                         [Y]
OverTime                                                 [Yes, No]
dtype: object
```

```
#Deleting redundant columns

df.drop(['EmployeeCount', 'Over18', 'StandardHours'], axis=1, inplace=True, errors='ignore')
```

```
#Correlation of numeric variables

df.corr()
```

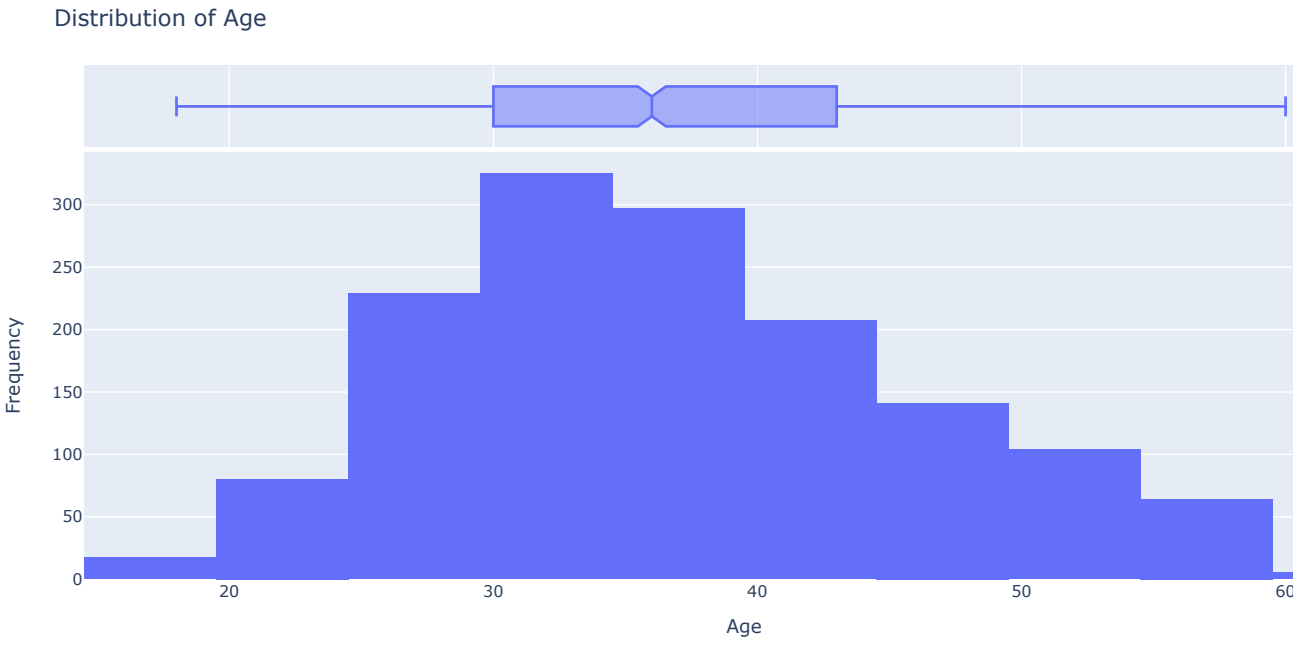
```
<ipython-input-27-02306b056f67>:3: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
```

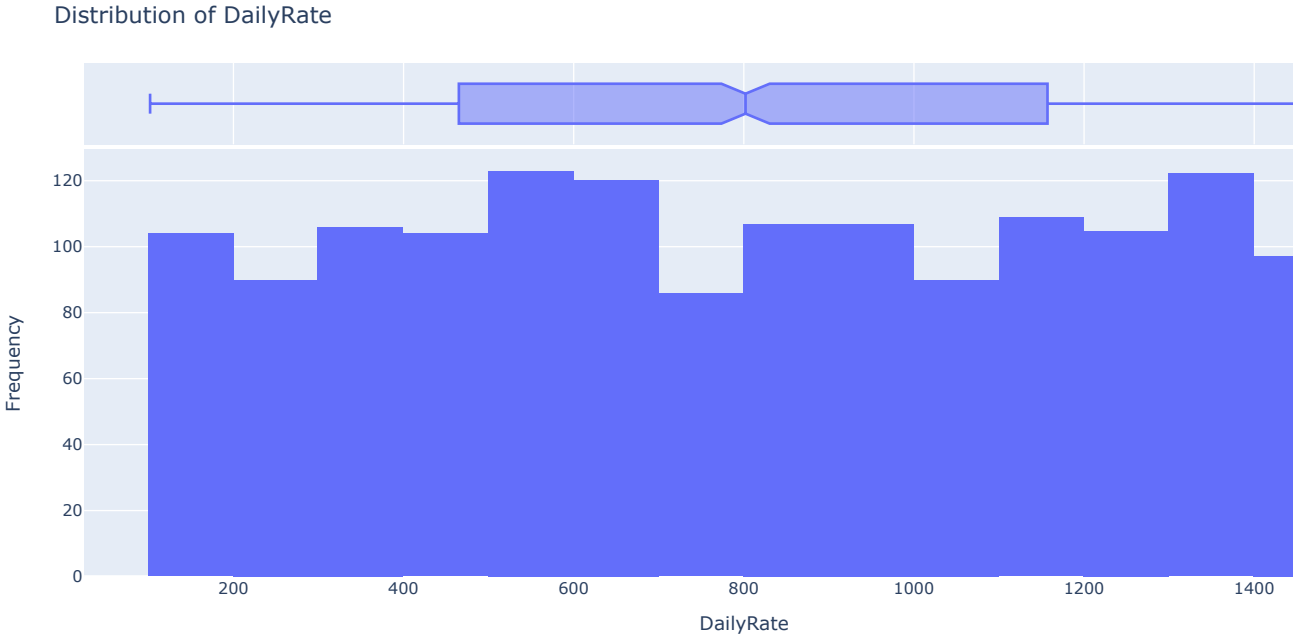
	Age	DailyRate	DistanceFromHome	Education	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobI
Age	1.000000	0.010661	-0.001686	0.208034	-0.010145	0.010146	0.024287	

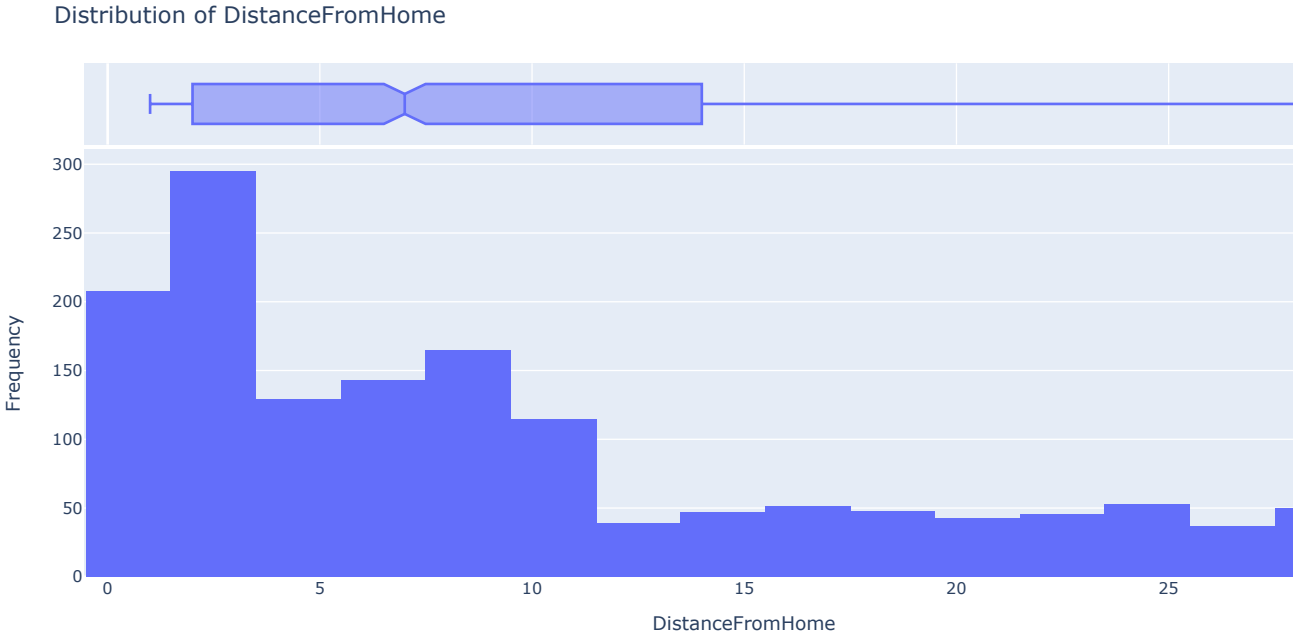
Data Visualization

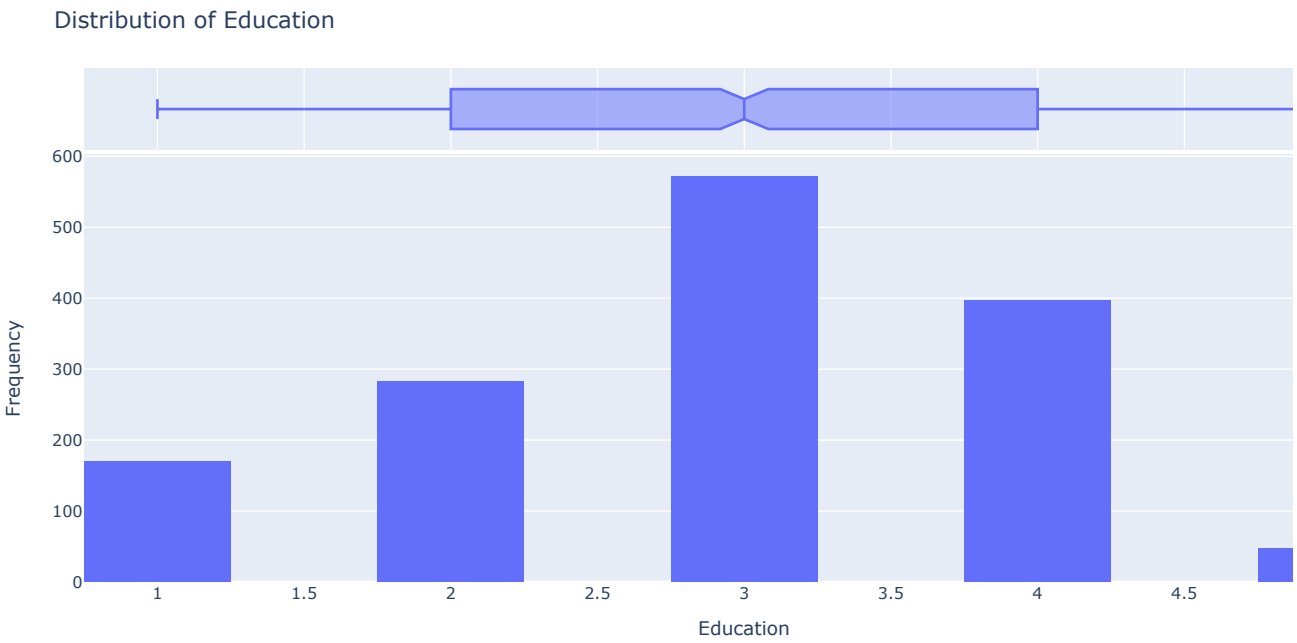
```
# Selecting only the numeric columns
numeric_columns = df.select_dtypes(include='number')

# Creating the heatmap
fig = px.imshow(numeric_columns.corr(), text_auto='.2f',  color_continuous_scale='sunsetdark')
#fig = px.imshow(numeric_columns.corr(), text_auto='.2f',  color_continuous_scale='YlOrRd')
# Updating the layout
fig.update_layout(
    title="Correlation Heatmap",
    width=1000,
    height=1000
)


fig.show()
```

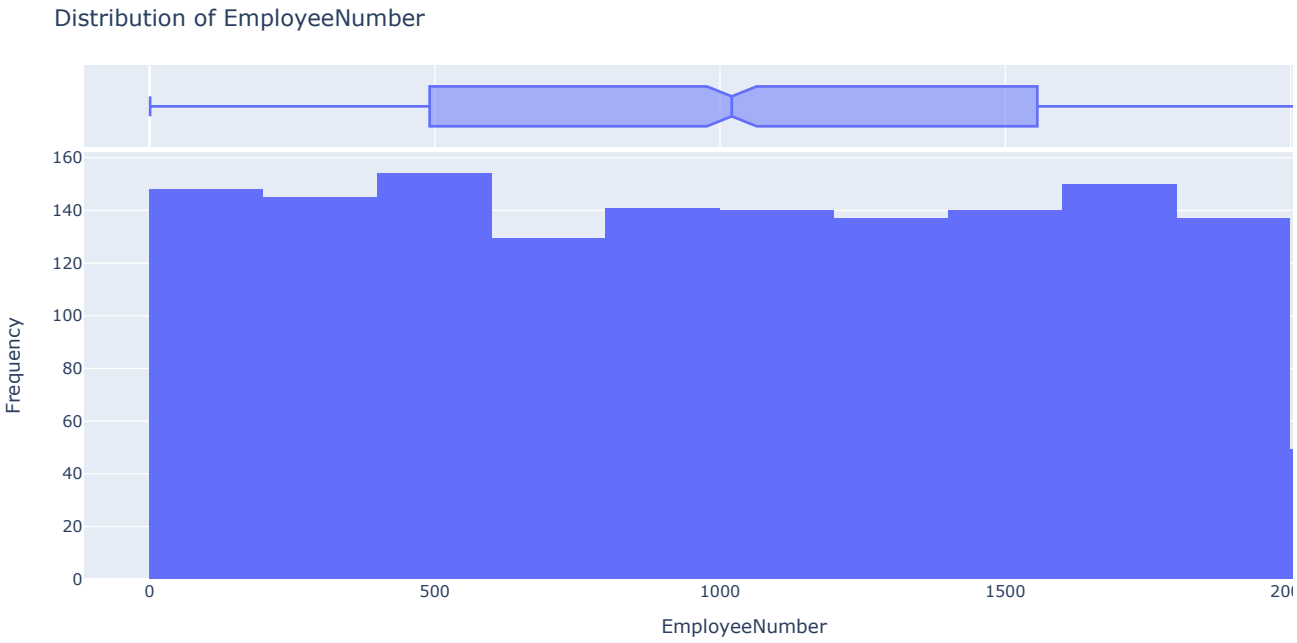
```
# Selecting only the numeric columns
numeric_columns = df.select_dtypes(include='number')

# Plotting the distribution of each numeric column
for column in numeric_columns.columns:
    fig = px.histogram(numeric_columns, x=column, nbins=20, marginal="box")
    fig.update_layout(
        title=f"Distribution of {column}",
        xaxis_title=column,
        yaxis_title="Frequency",
        # width=600,
        # height=400,
        showlegend=False
    )
    fig.show()
```

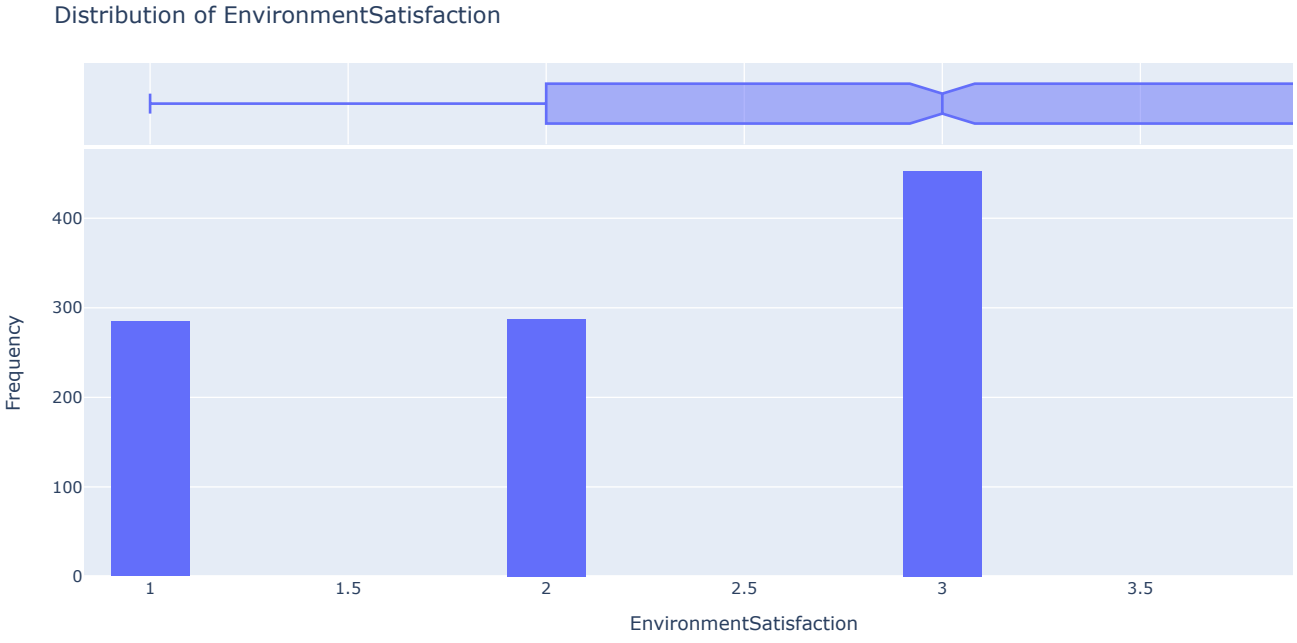

Distribution of Age
Frequency
0
50
100
150
200
250
300
20
30
40
50
60
Age


Distribution of DailyRate
Frequency
0
20
40
60
80
100
120
200
400
600
800
1000
1200
1400
DailyRate

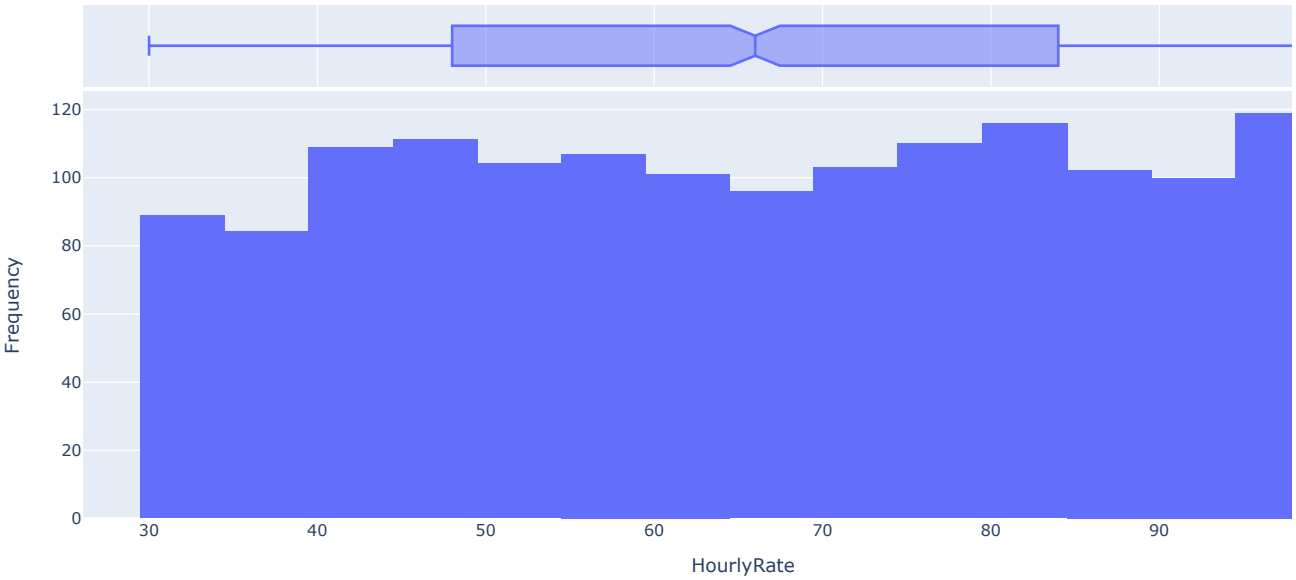

Distribution of DistanceFromHome
Frequency
0
50
100
150
200
250
300
0
5
10
15
20
25
DistanceFromHome


Distribution of Education
Frequency
600
500
400
300
200
100
0
1
1.5
2
2.5
3
3.5
4
4.5
Education

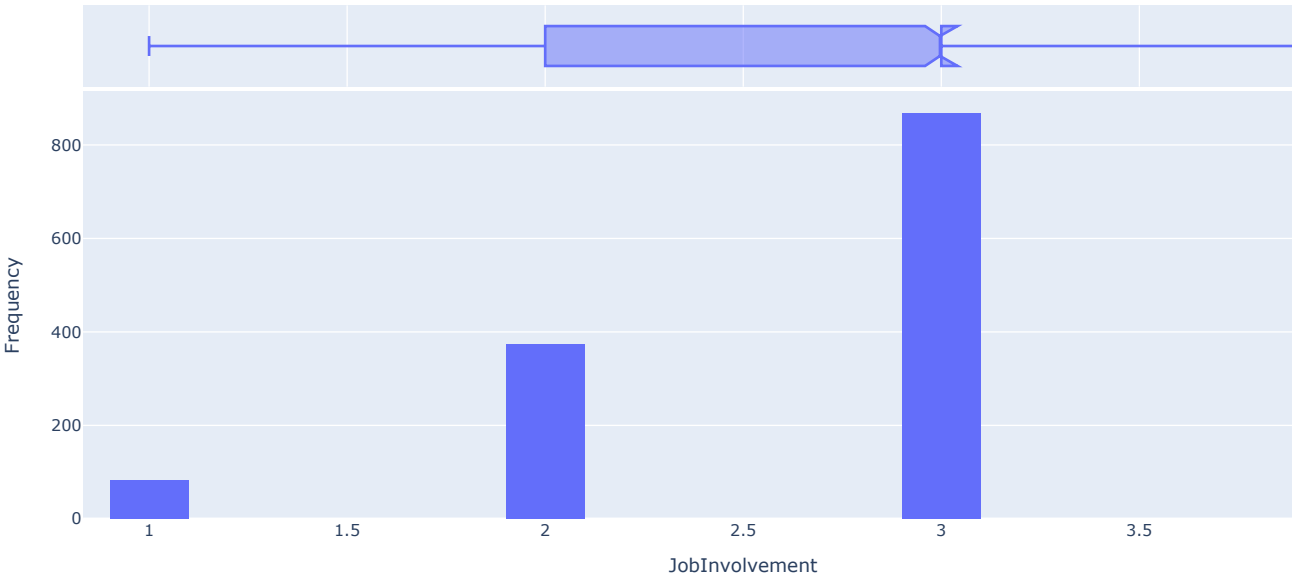

Distribution of EmployeeNumber
Frequency
160
140
120
100
80
60
40
20
0
0
500
1000
1500
EmployeeNumber


Distribution of EnvironmentSatisfaction
Frequency
400
300
200
100
0
1
1.5
2
2.5
3
3.5
EnvironmentSatisfaction

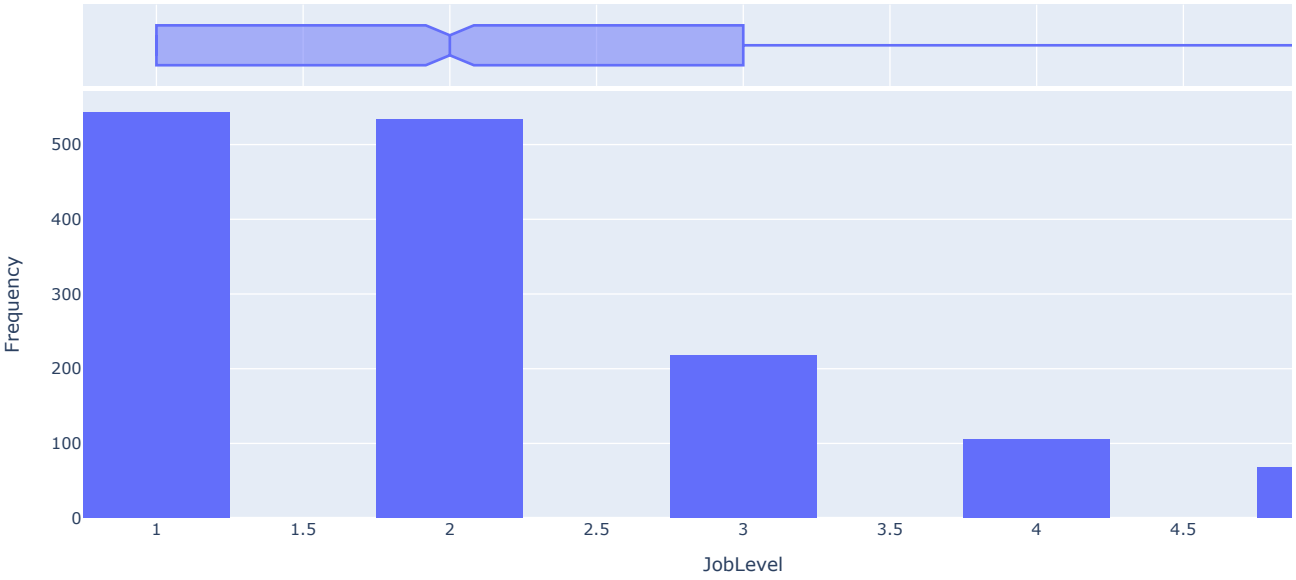
Distribution of HourlyRate



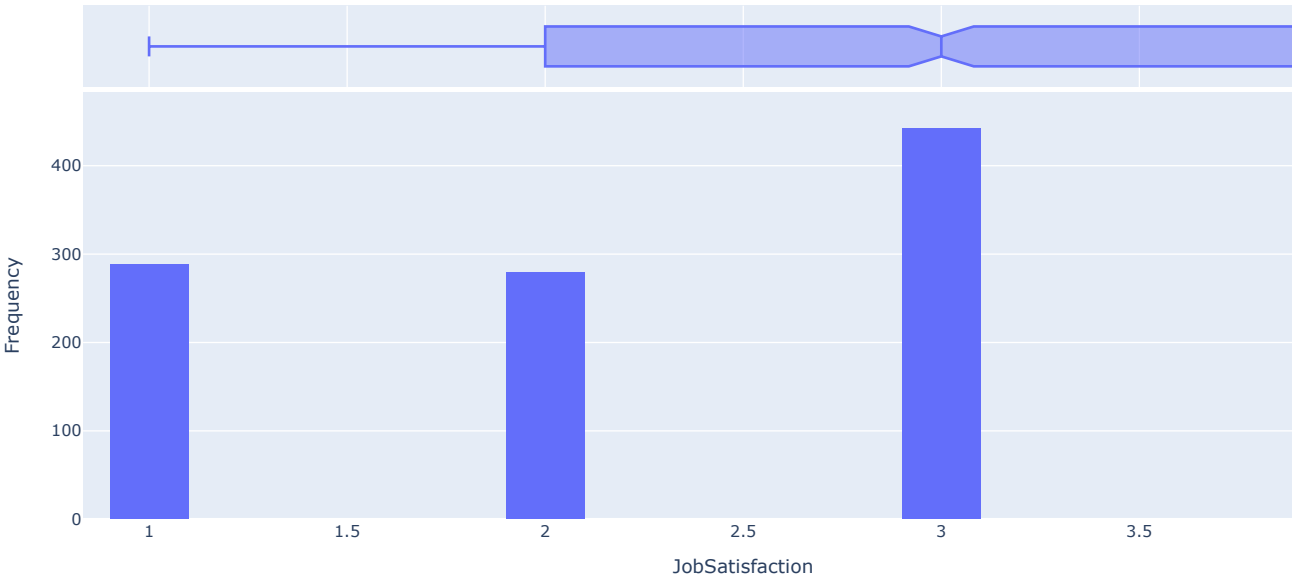
Distribution of JobInvolvement



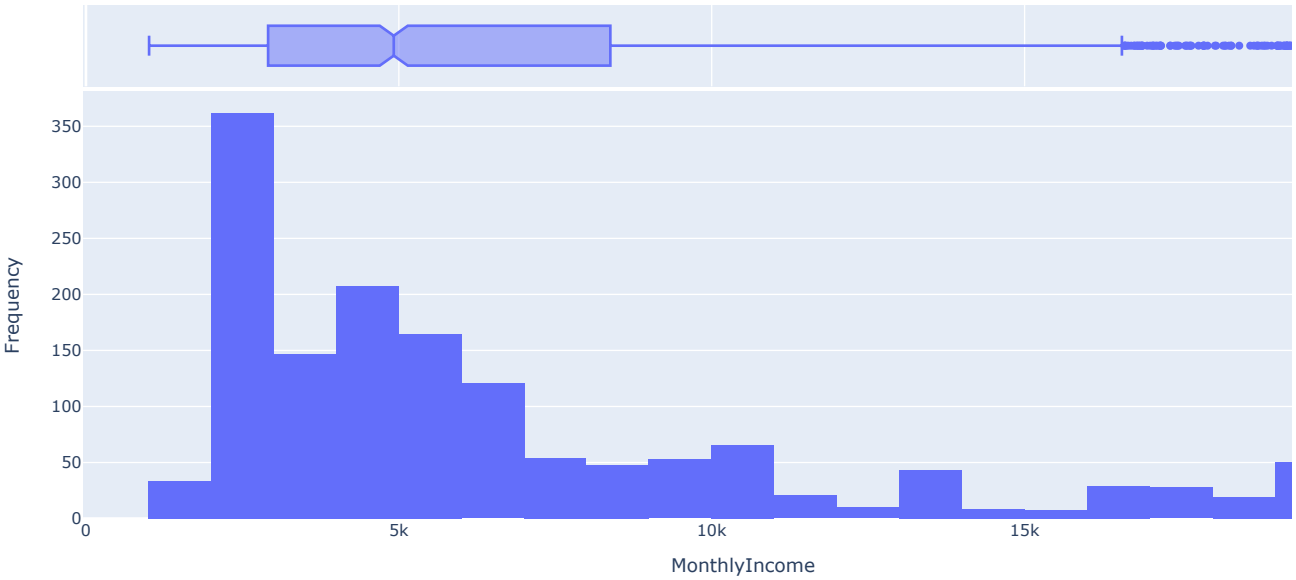
Distribution of JobLevel



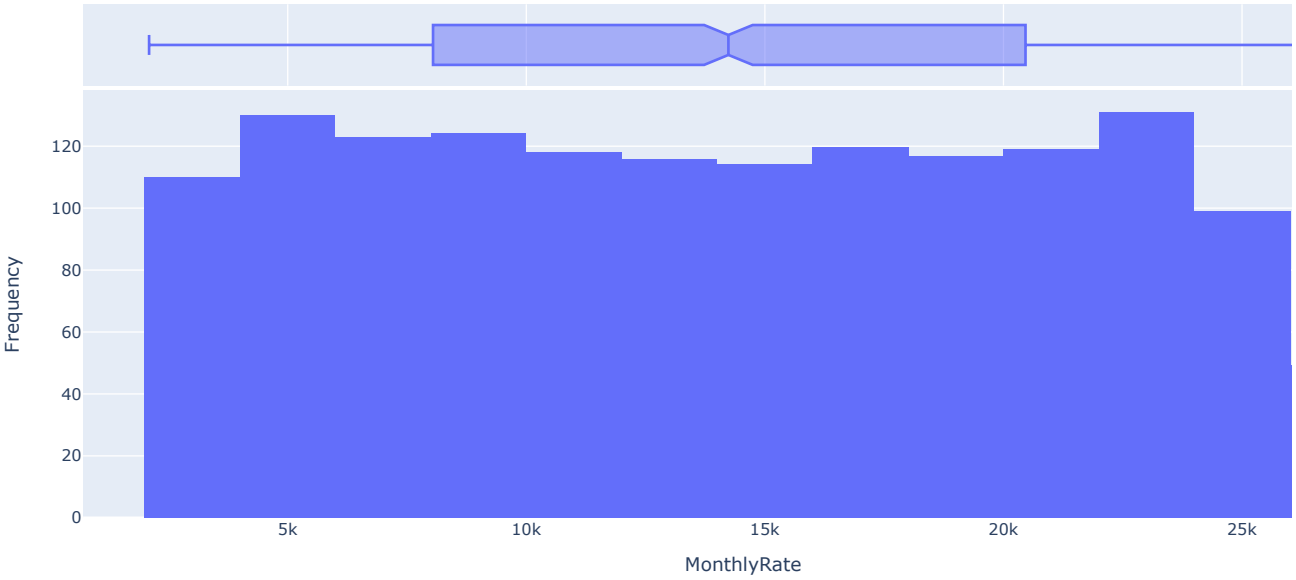
Distribution of JobSatisfaction

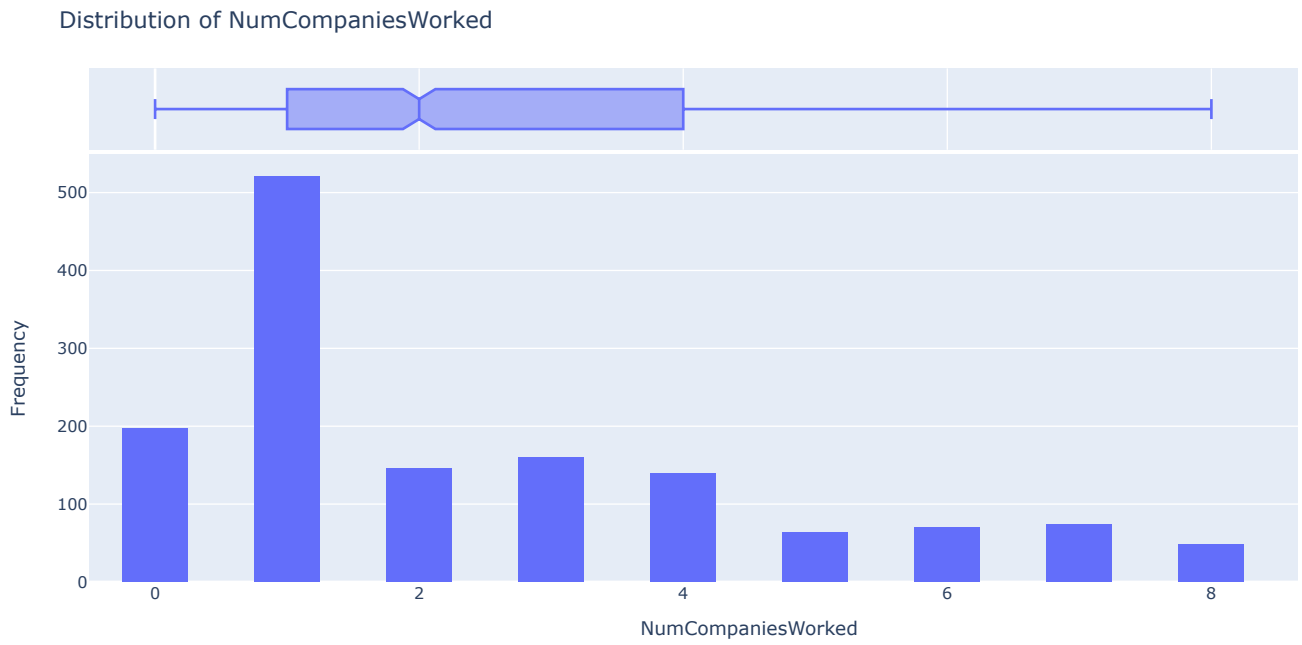


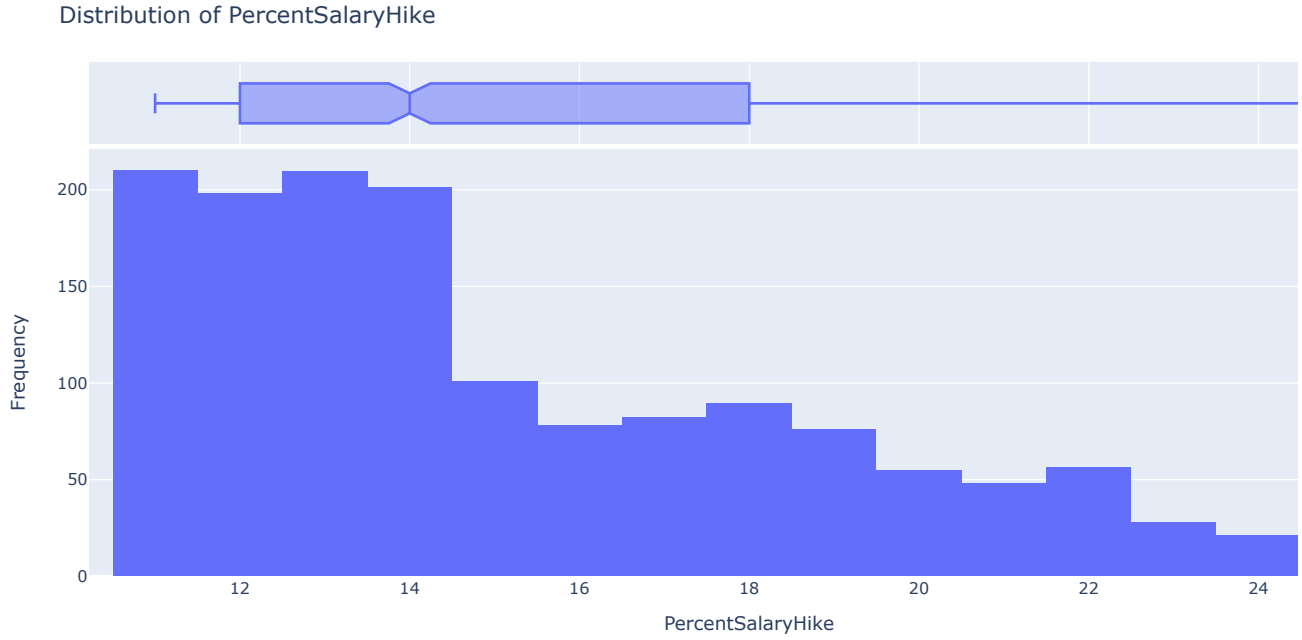
Distribution of MonthlyIncome

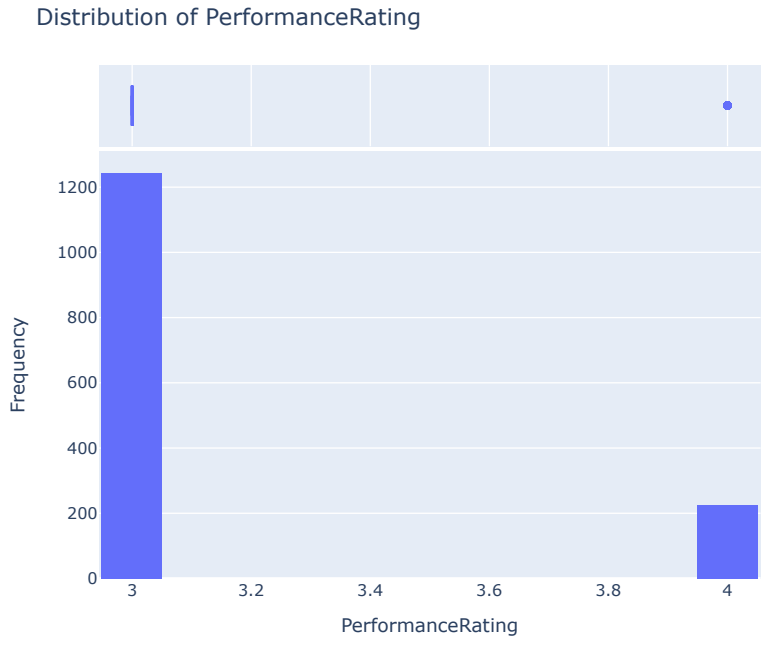


Distribution of MonthlyRate

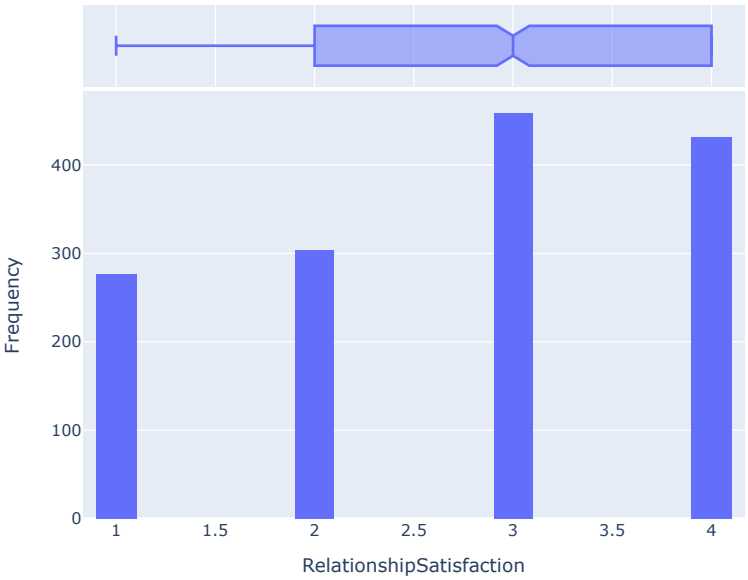



Distribution of NumCompaniesWorked
Frequency
0
100
200
300
400
500
0
2
4
6
8
NumCompaniesWorked

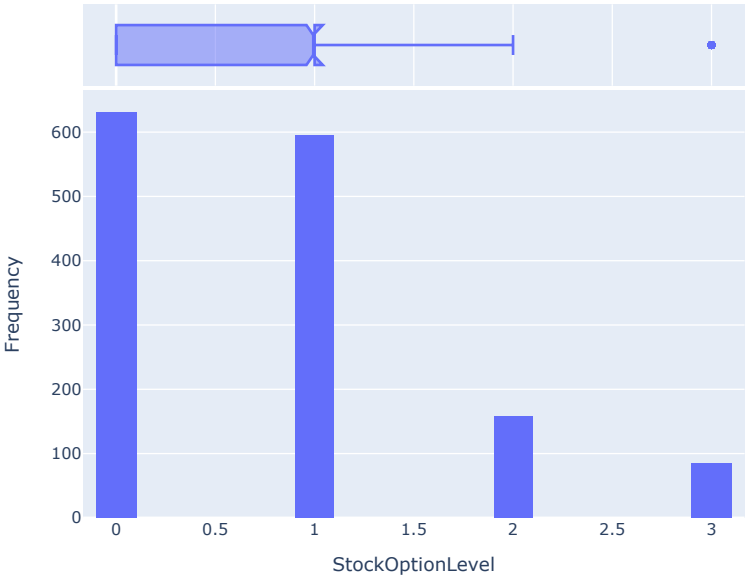

Distribution of PercentSalaryHike
Frequency
0
50
100
150
200
12
14
16
18
20
22
24
PercentSalaryHike


Distribution of PerformanceRating
Frequency
0
200
400
600
800
1000
1200
3
3.2
3.4
3.6
3.8
4
PerformanceRating

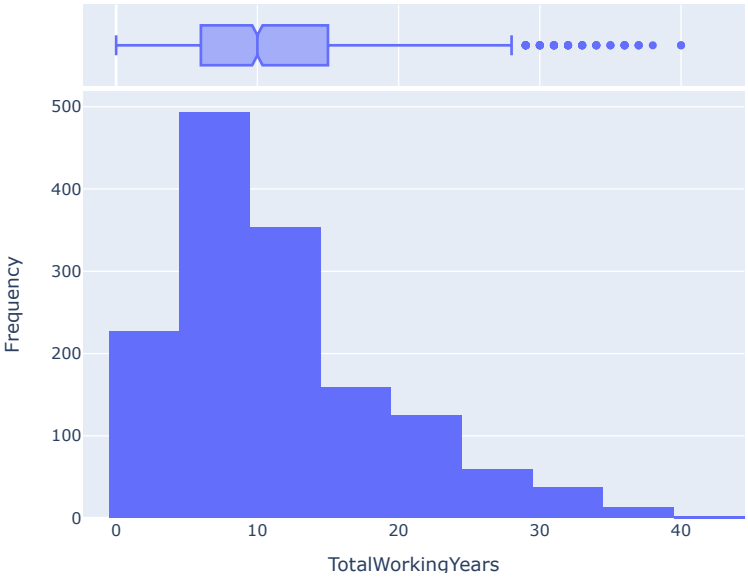
Distribution of RelationshipSatisfaction



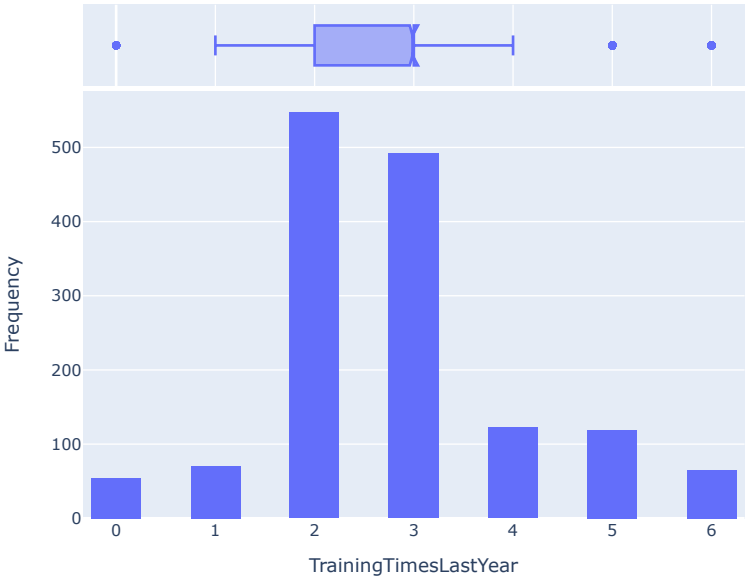
Distribution of StockOptionLevel



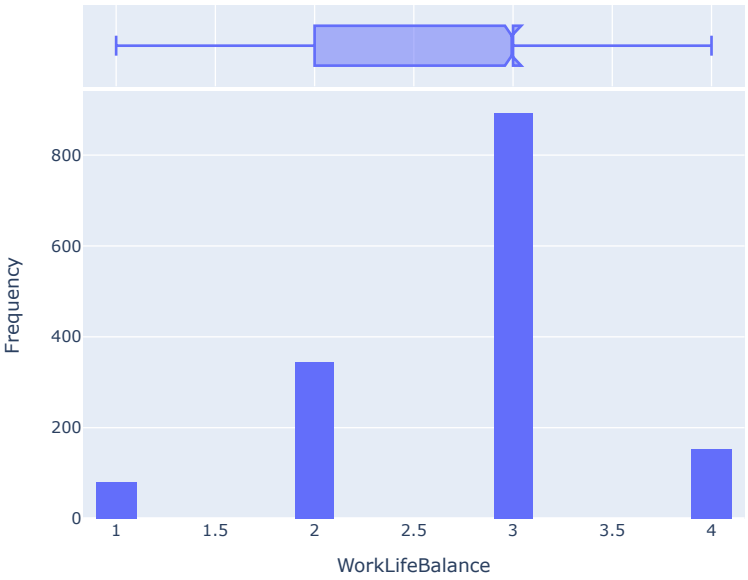
Distribution of TotalWorkingYears



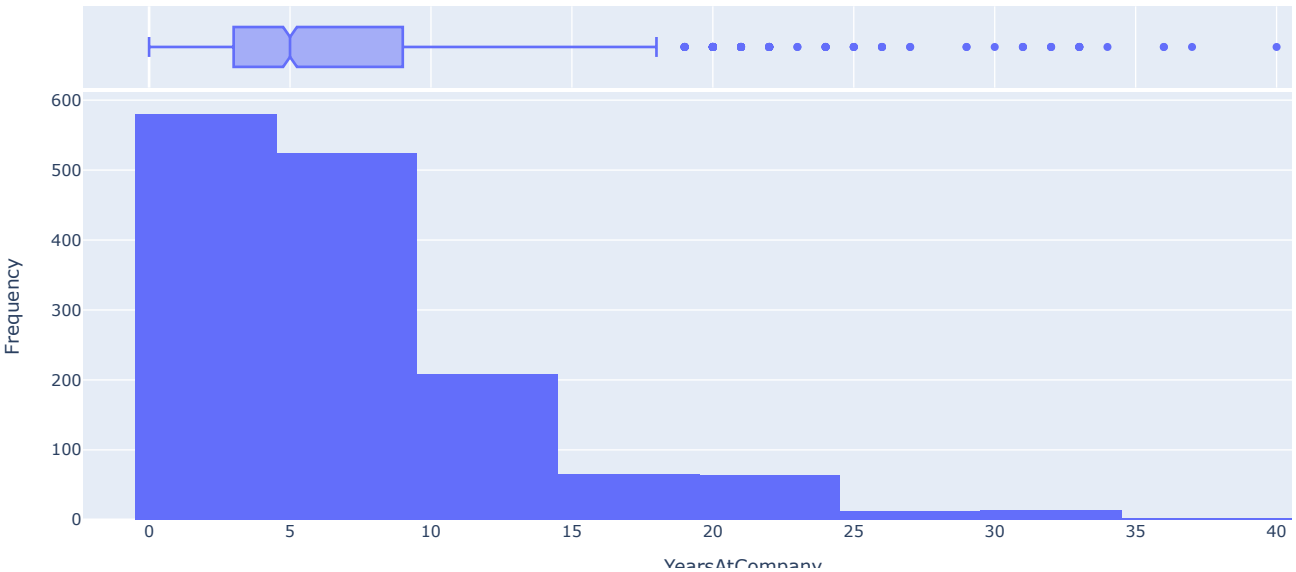
Distribution of TrainingTimesLastYear

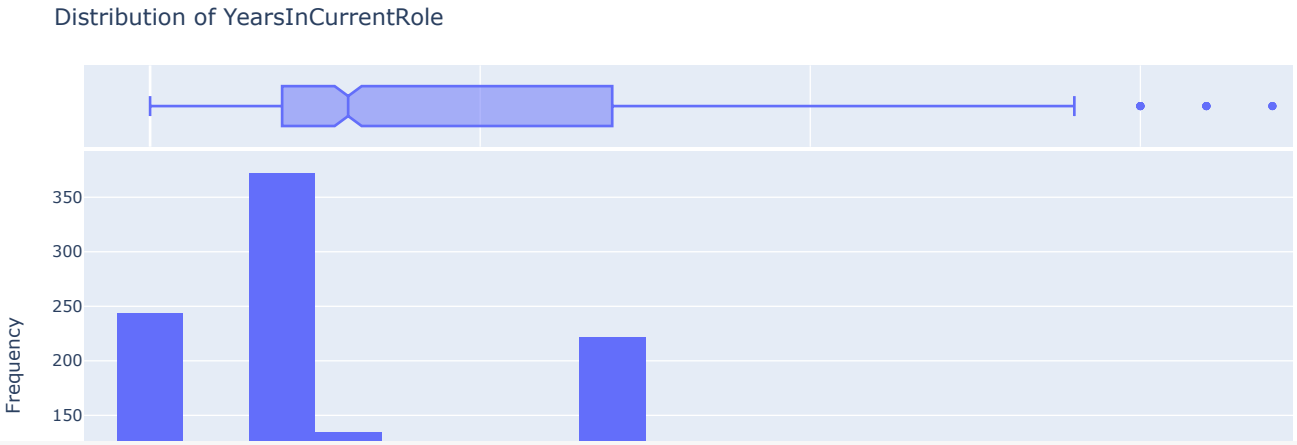


Distribution of WorkLifeBalance



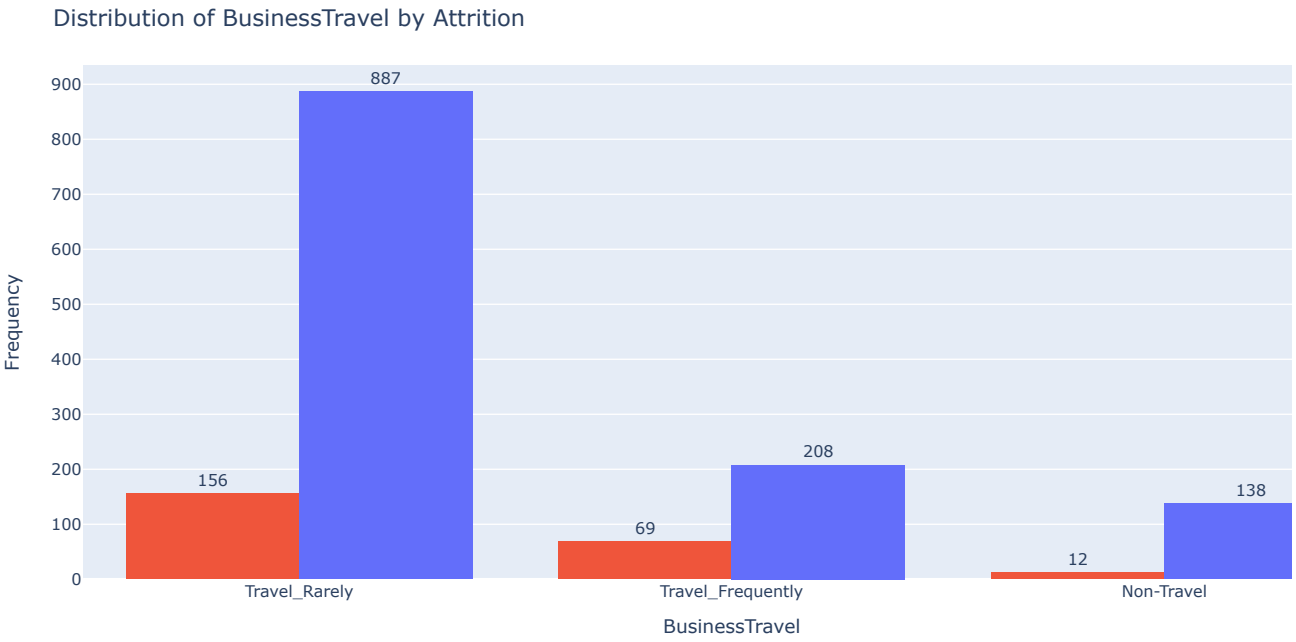
Distribution of YearsAtCompany

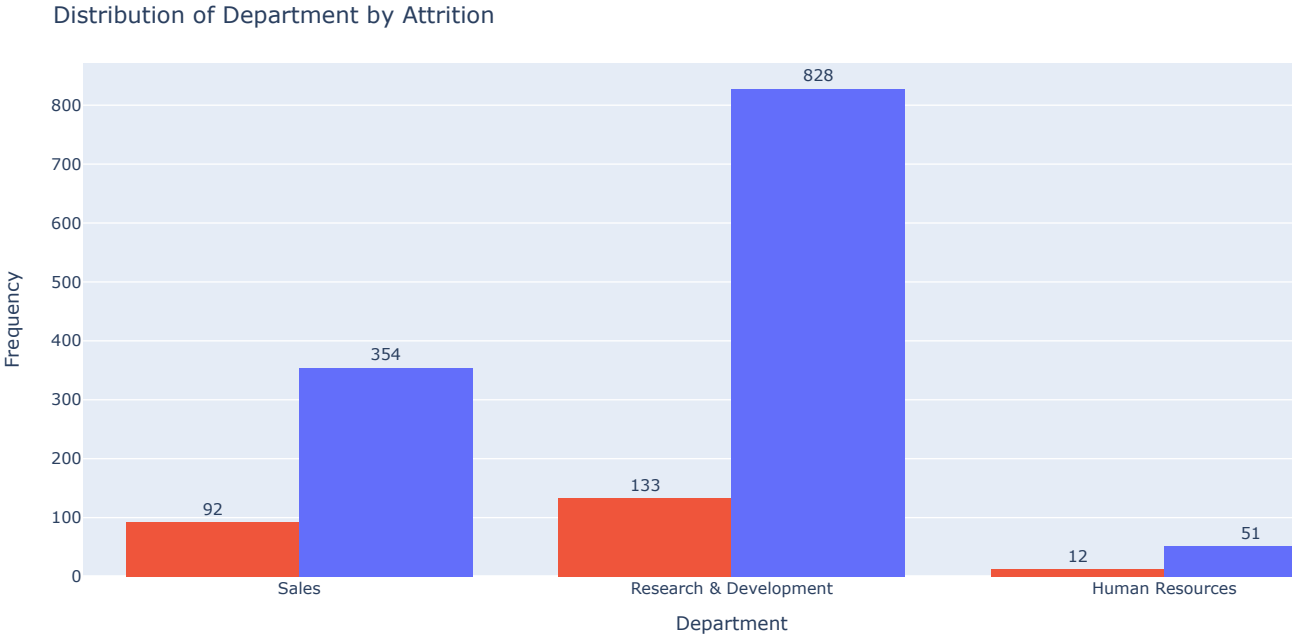


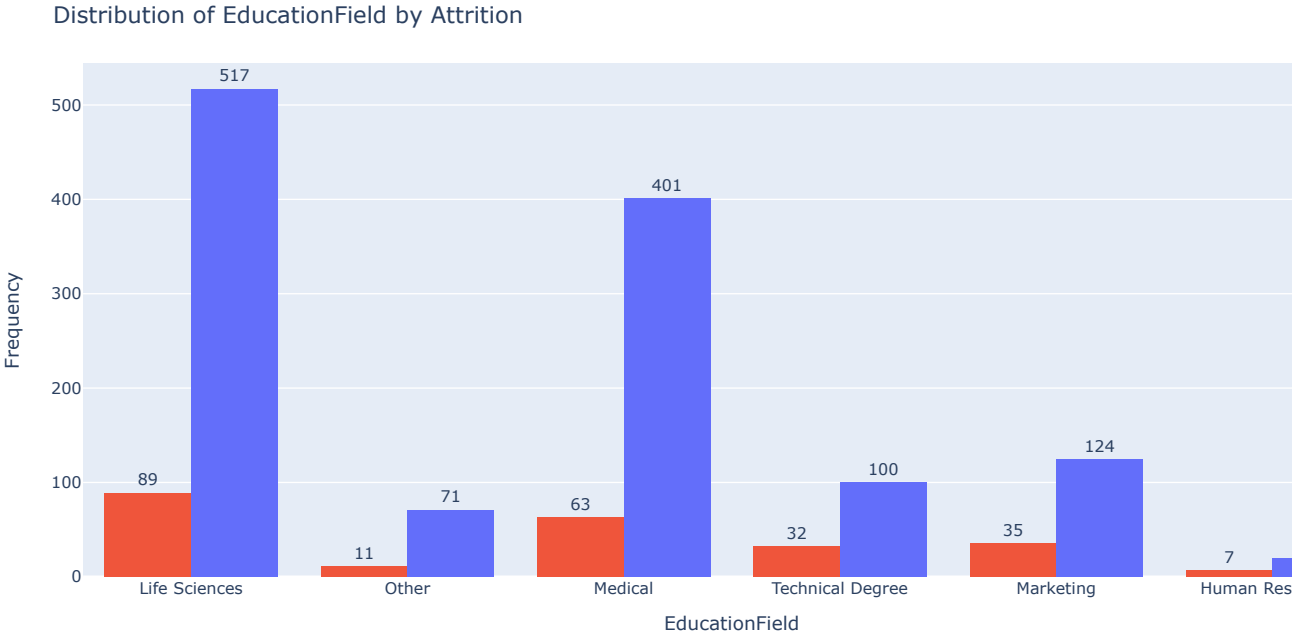


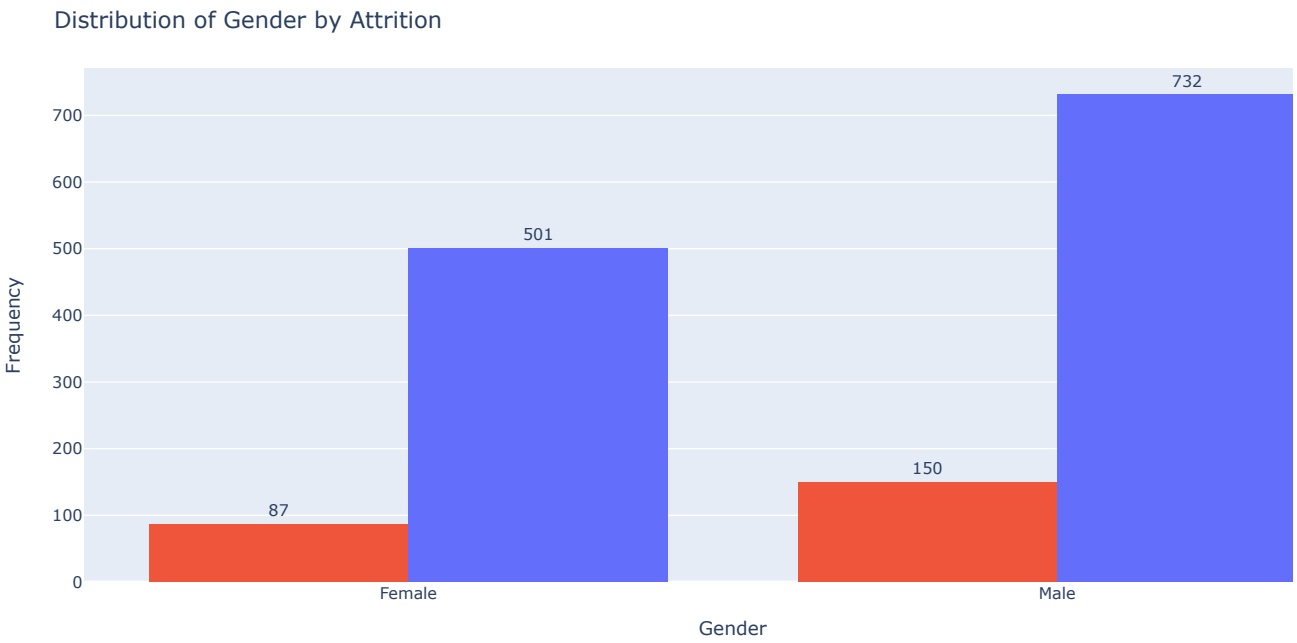

```
#Filter only object type columns without Attrition
obj_columns=df.select_dtypes(include='object').drop(['Attrition'], axis=1)
```

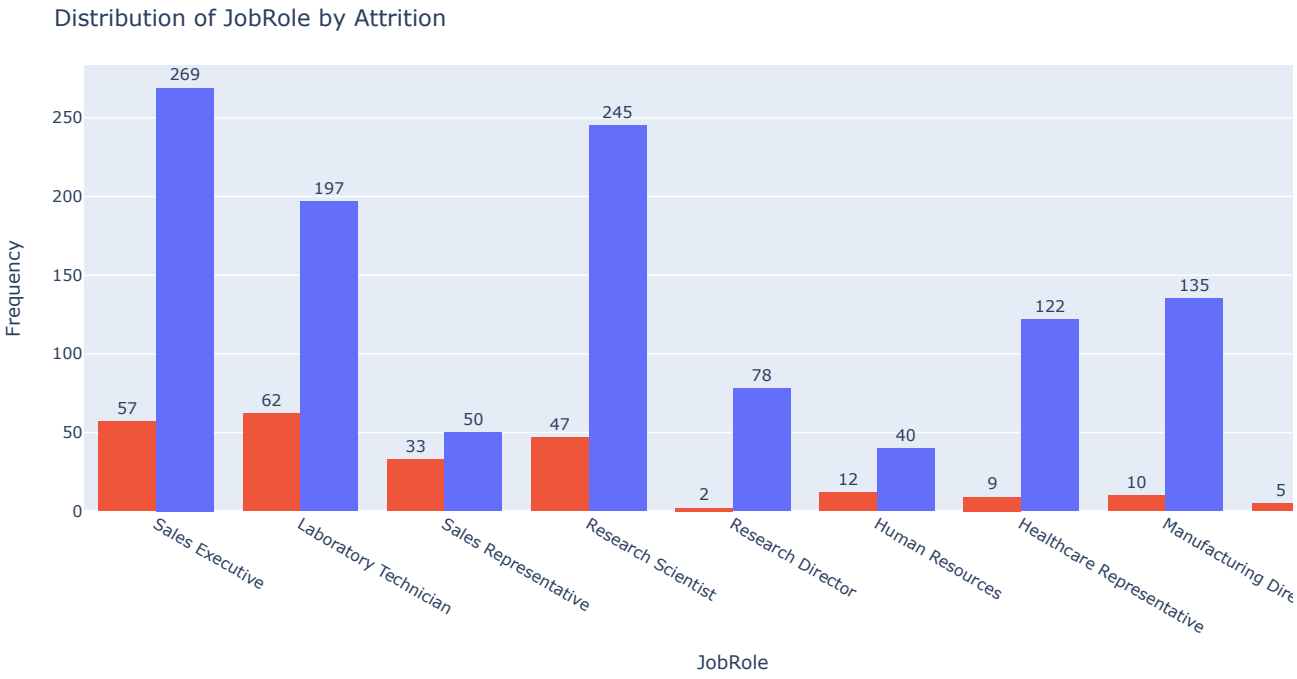
```
# Plotting the distribution of each object column grouped by Attrition
for column in obj_columns.columns:
    fig = px.histogram(df, x=column, color='Attrition', barmode='group', text_auto=True, color_discrete_sequence=[px.colors.qualitative.P
    fig.update_layout(
        title=f"Distribution of {column} by Attrition",
        xaxis_title=column,
        yaxis_title="Frequency",
        # width=600,
        # height=400,
        showlegend=True
    )
    fig.update_traces(textposition='outside')
    fig.show()
```

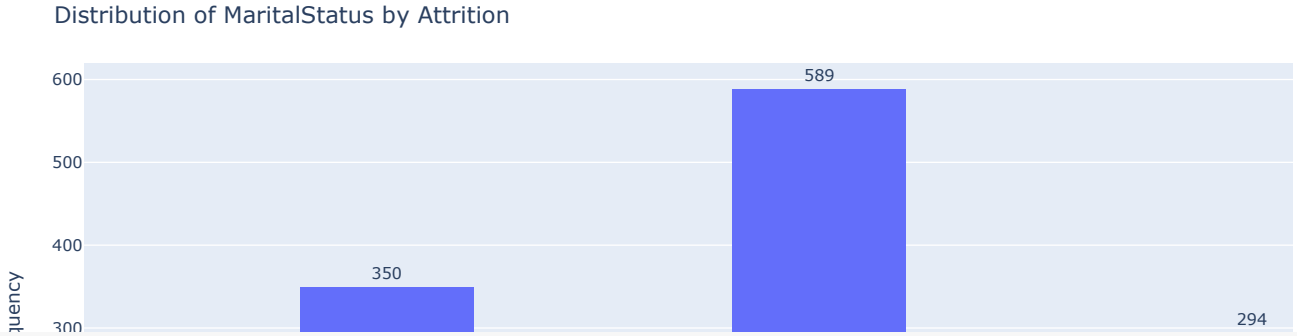

Distribution of BusinessTravel by Attrition
Frequency
900
800
700
600
500
400
300
200
100
0
887
156
208
69
138
12
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel


Distribution of Department by Attrition
Frequency
800
700
600
500
400
300
200
100
0
828
354
92
133
51
12
Sales
Research & Development
Human Resources
Department

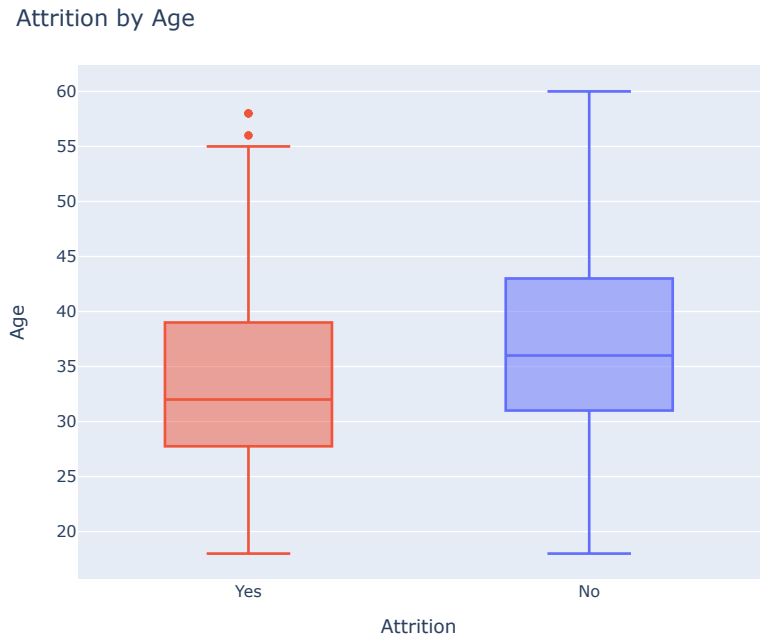

Distribution of EducationField by Attrition
Frequency
500
400
300
200
100
0
517
401
89
71
63
11
124
100
32
35
7
Life Sciences
Other
Medical
Technical Degree
Marketing
Human Res
EducationField



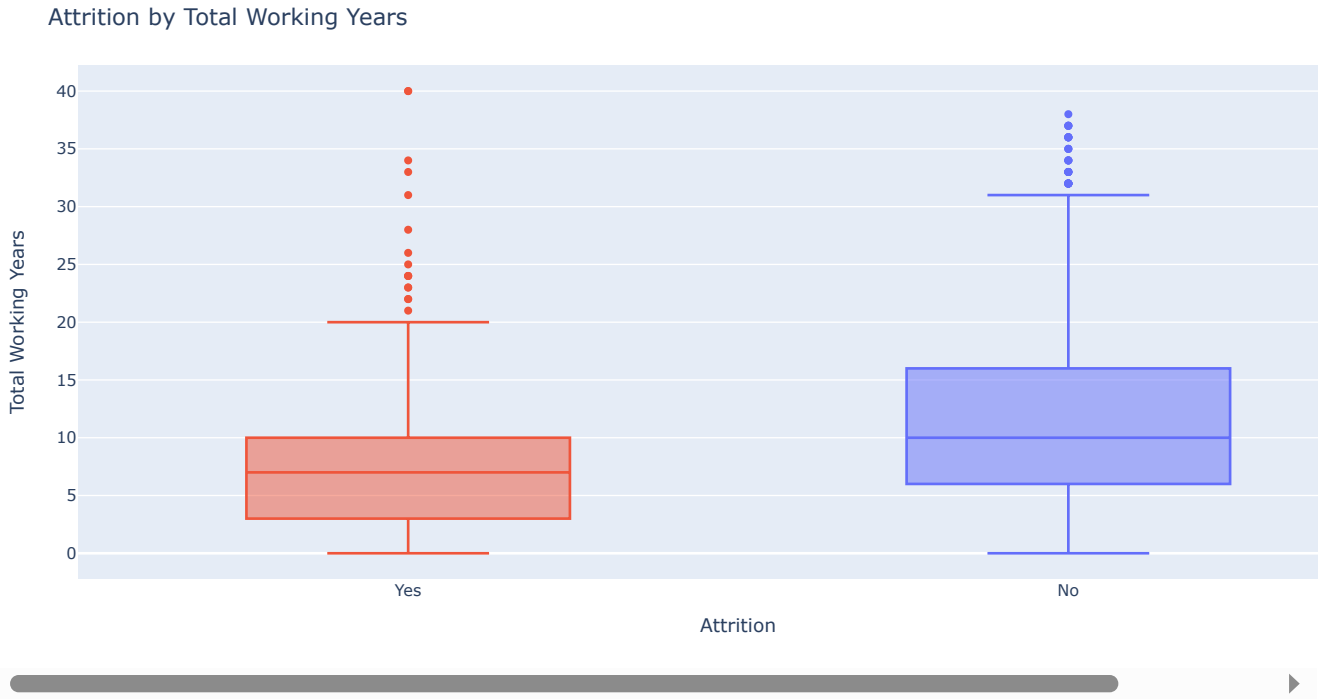




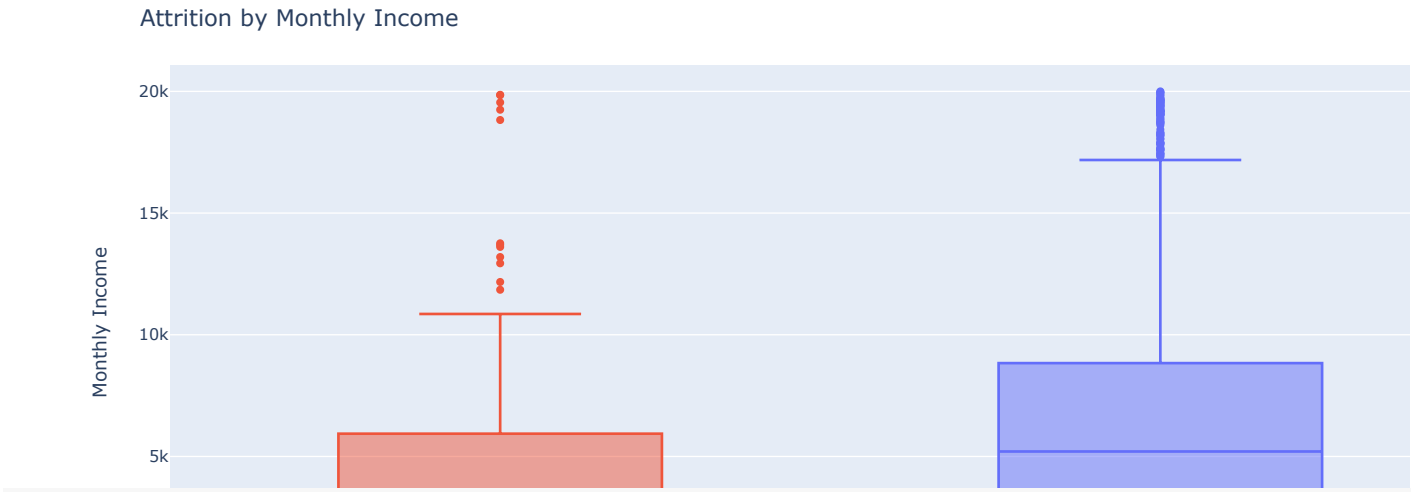

```
fig = px.box(df, x="Attrition", y="Age", color="Attrition", color_discrete_sequence=[px.colors.qualitative.Plotly[1],px.colors.qualitativ
fig.update_layout(
    title="Attrition by Age",
    xaxis_title="Attrition",
    yaxis_title="Age",
    showlegend=False
)
fig.show()
```




```
fig = px.box(df, x="Attrition", y="TotalWorkingYears", color="Attrition", color_discrete_sequence=[px.colors.qualitative.Plotly[1],px.col
fig.update_layout(
    title="Attrition by Total Working Years",
    xaxis_title="Attrition",
    yaxis_title="Total Working Years",
    showlegend=False
)
fig.show()
```




```
fig = px.box(df, x="Attrition", y="MonthlyIncome", color="Attrition", color_discrete_sequence=[px.colors.qualitative.Plotly[1],px.colors.
fig.update_layout(
    title="Attrition by Monthly Income",
    xaxis_title="Attrition",
    yaxis_title="Monthly Income",
    showlegend=False
)
fig.show()
```




```
import seaborn as sns
sns.lineplot(data=df, y='JobLevel' , x="YearsInCurrentRole")
plt.title('Career growth')
```

```
Text(0.5, 1.0, 'Career growth')
```