In [2]:
```
## Import the library
import pandas as pd
import numpy  as np
import matplotlib.pyplot as   plt
import seaborn as sns
import plotly.express as px
from ydata_profiling import ProfileReport
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
sns.set()
pd.options.display.float_format = '{:,.2f}'.format
pd.options.display.max_rows = None
pd.options.display.max_columns = None
```

load Data

In [3]:
```
hr=pd.read_csv("HR-Employee-Attrition.csv")
```

In [4]:
```
hr.sample(10)
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSa
585	23	Yes	Travel_Rarely	1243	Research & Development	6	3	Life Sciences	1	811	
480	30	Yes	Travel_Frequently	448	Sales	12	4	Life Sciences	1	648	
142	38	No	Travel_Rarely	364	Research & Development	3	5	Technical Degree	1	193	
736	48	No	Travel_Rarely	1355	Research & Development	4	4	Life Sciences	1	1024	
441	42	No	Travel_Frequently	1474	Research & Development	5	2	Other	1	591	
1335	39	No	Travel_Rarely	835	Research & Development	19	4	Other	1	1871	
1081	35	No	Travel_Rarely	1029	Research & Development	16	3	Life Sciences	1	1529	
1371	56	No	Travel_Rarely	1443	Sales	11	5	Marketing	1	1935	
971	51	No	Travel_Rarely	1405	Research & Development	11	2	Technical Degree	1	1367	
76	35	No	Travel_Rarely	776	Sales	1	4	Marketing	1	100	

Explortatory Data Analyses

In [5]:
```
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   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
```

In [6]:
```
hr.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
hr.isnull().sum()
```

Out[7]:
```
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
```

In [8]:
```
hr.nunique()
```

Out[8]:
```
Age                          43
Attrition                     2
BusinessTravel                3
DailyRate                   886
Department                    3
DistanceFromHome             29
Education                     5
EducationField                6
EmployeeCount                 1
EmployeeNumber             1470
EnvironmentSatisfaction       4
Gender                        2
HourlyRate                   71
JobInvolvement                4
JobLevel                      5
JobRole                       9
JobSatisfaction               4
MaritalStatus                 3
MonthlyIncome              1349
MonthlyRate                1427
NumCompaniesWorked           10
Over18                        1
OverTime                      2
PercentSalaryHike            15
PerformanceRating             2
RelationshipSatisfaction      4
StandardHours                 1
StockOptionLevel              4
TotalWorkingYears            40
TrainingTimesLastYear         7
WorkLifeBalance               4
YearsAtCompany               37
YearsInCurrentRole           19
YearsSinceLastPromotion      16
YearsWithCurrManager         18
dtype: int64
```

In [9]:
```
for col in hr.columns:

    # Check column dtype
    if hr[col].dtype == 'object':
```

```
        # Column details
        print(f"Column: {col}")

        # Number of unique values
        print(f"Unique values: {hr[col].unique()}")

        # Value counts
        print(hr[col].value_counts())

        # Separator
        print("="*40)
```

```
Column: Attrition
Unique values: ['Yes' 'No']
No     1233
Yes     237
Name: Attrition, dtype: int64
========================================
Column: BusinessTravel
Unique values: ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
Travel_Rarely        1043
Travel_Frequently     277
Non-Travel            150
Name: BusinessTravel, dtype: int64
========================================
Column: Department
Unique values: ['Sales' 'Research & Development' 'Human Resources']
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
========================================
Column: EducationField
Unique values: ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
Life Sciences       606
Medical             464
Marketing           159
Technical Degree    132
Other                82
Human Resources      27
Name: EducationField, dtype: int64
========================================
Column: Gender
Unique values: ['Female' 'Male']
Male      882
Female    588
Name: Gender, dtype: int64
========================================
Column: JobRole
Unique values: ['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']
Sales Executive              326
Research Scientist           292
Laboratory Technician        259
Manufacturing Director       145
Healthcare Representative    131
Manager                      102
Sales Representative          83
Research Director             80
Human Resources               52
Name: JobRole, dtype: int64
========================================
Column: MaritalStatus
Unique values: ['Single' 'Married' 'Divorced']
Married     673
Single      470
Divorced    327
Name: MaritalStatus, dtype: int64
========================================
Column: Over18
Unique values: ['Y']
Y    1470
Name: Over18, dtype: int64
========================================
Column: OverTime
Unique values: ['Yes' 'No']
No     1054
Yes     416
Name: OverTime, dtype: int64
========================================
```

In [10]:
```
for col in hr.columns:

    # Check column numiric
    if hr[col].dtype != 'object':

        # Column details
        print(f"Column: {col}")

        # max & min values
        print(f"max: {hr[col].max()}",f"min: {hr[col].min()}")

        # Separator
        print("="*40)
```

```
Column: Age
max: 60 min: 18
```

```
========================================
Column: DailyRate
max: 1499 min: 102
========================================
Column: DistanceFromHome
max: 29 min: 1
========================================
Column: Education
max: 5 min: 1
========================================
Column: EmployeeCount
max: 1 min: 1
========================================
Column: EmployeeNumber
max: 2068 min: 1
========================================
Column: EnvironmentSatisfaction
max: 4 min: 1
========================================
Column: HourlyRate
max: 100 min: 30
========================================
Column: JobInvolvement
max: 4 min: 1
========================================
Column: JobLevel
max: 5 min: 1
========================================
Column: JobSatisfaction
max: 4 min: 1
========================================
Column: MonthlyIncome
max: 19999 min: 1009
========================================
Column: MonthlyRate
max: 26999 min: 2094
========================================
Column: NumCompaniesWorked
max: 9 min: 0
========================================
Column: PercentSalaryHike
max: 25 min: 11
========================================
Column: PerformanceRating
max: 4 min: 3
========================================
Column: RelationshipSatisfaction
max: 4 min: 1
========================================
Column: StandardHours
max: 80 min: 80
========================================
Column: StockOptionLevel
max: 3 min: 0
========================================
Column: TotalWorkingYears
max: 40 min: 0
========================================
Column: TrainingTimesLastYear
max: 6 min: 0
========================================
Column: WorkLifeBalance
max: 4 min: 1
========================================
Column: YearsAtCompany
max: 40 min: 0
========================================
Column: YearsInCurrentRole
max: 18 min: 0
========================================
Column: YearsSinceLastPromotion
max: 15 min: 0
========================================
Column: YearsWithCurrManager
max: 17 min: 0
========================================
```

In [11]:
```
hr.columns
```

Out[11]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [12]:
```
columns_to_drop = ['EmployeeCount','EmployeeNumber','Over18','StandardHours']

# Drop columns
for col in columns_to_drop:
    hr.drop(col, axis=1, inplace=True)
```

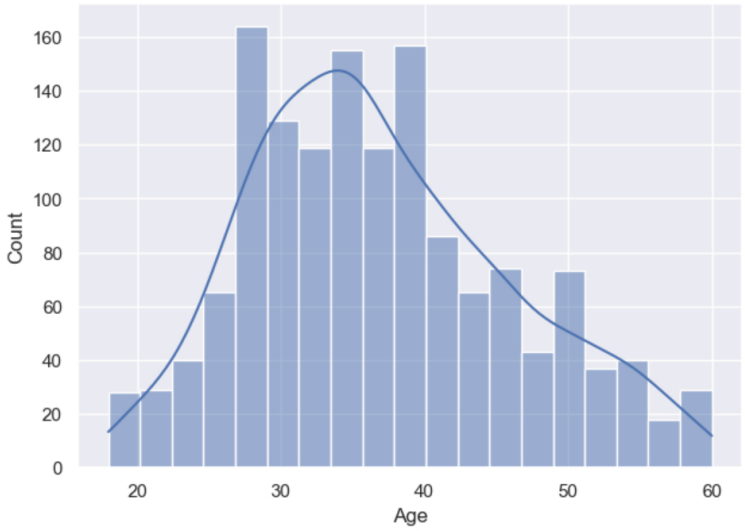
In [ ]:

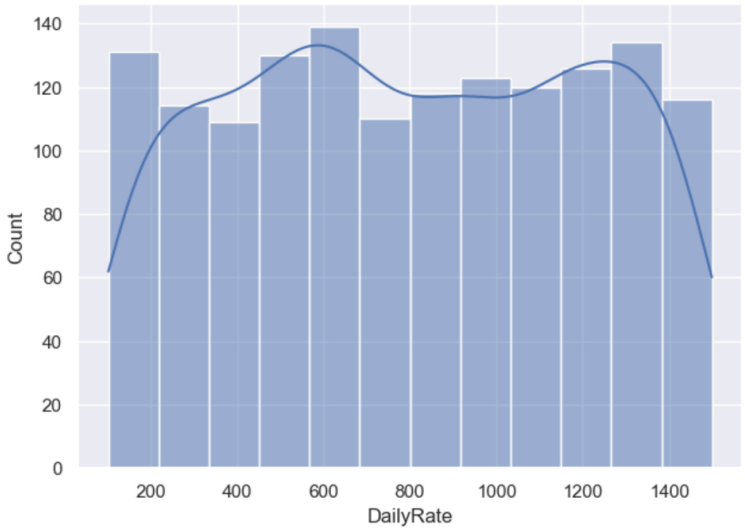
Univariate Analyses

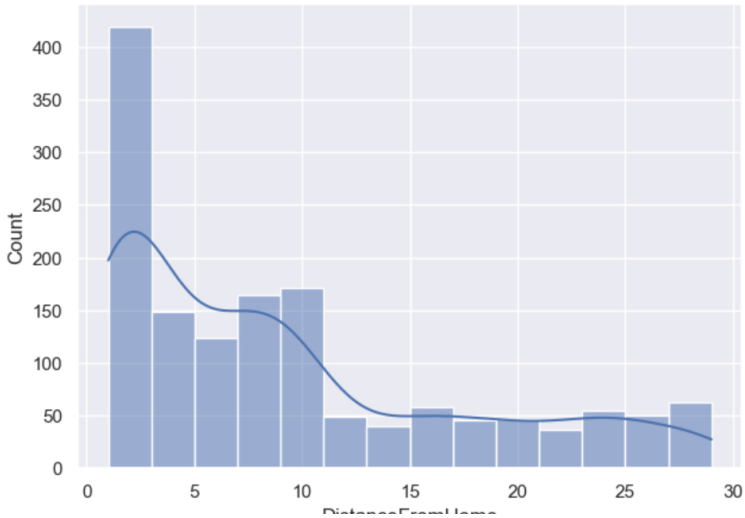
In [13]:
```
obj=hr.select_dtypes(include="object")
num=hr.select_dtypes(exclude="object")
```

In [14]:
```
obj_col=list(hr.select_dtypes(include="object").columns)
num_col=list(hr.select_dtypes(exclude="object").columns)
```

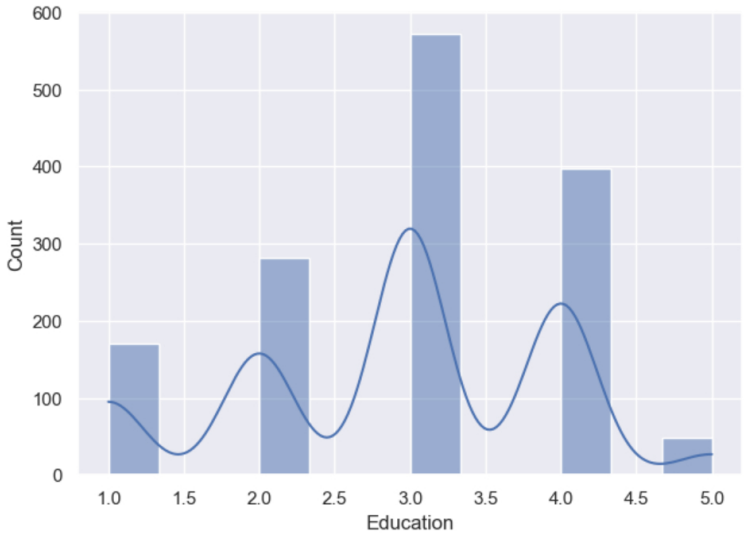
In [15]:
```
## numeric columns
for col in num_col:
    sns.histplot(hr[col],kde=True)
    plt.show()
```

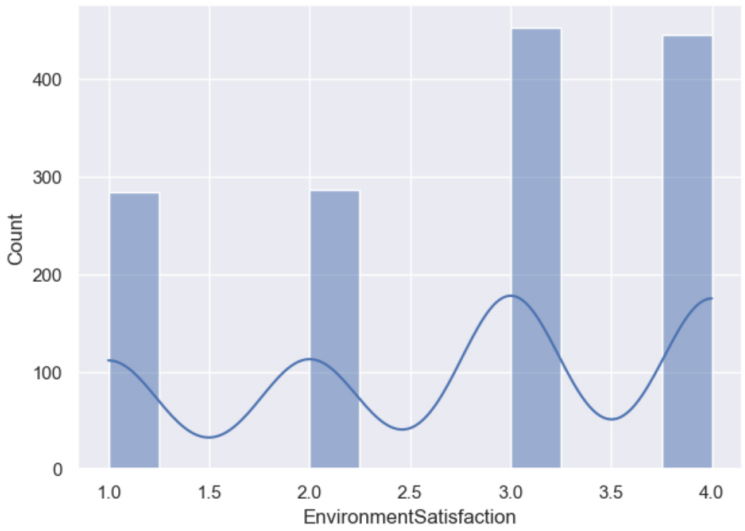


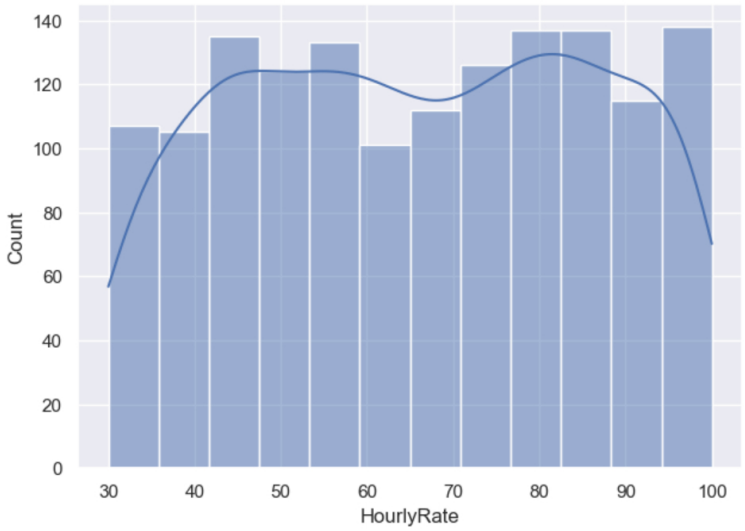


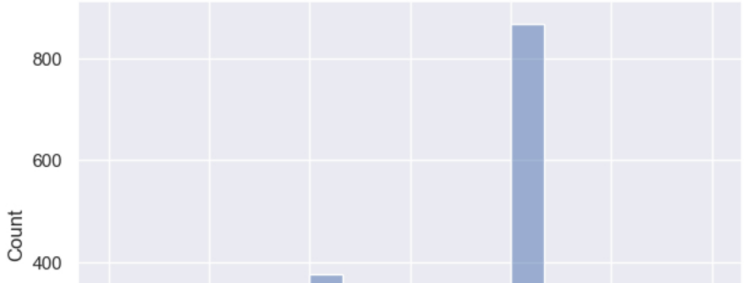


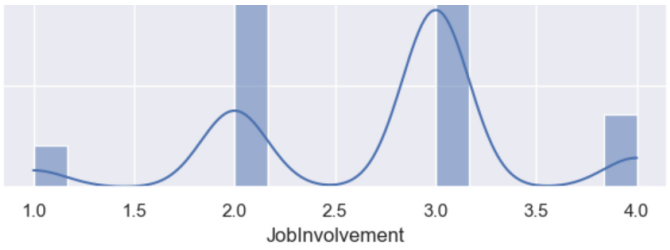
DistanceFromHome

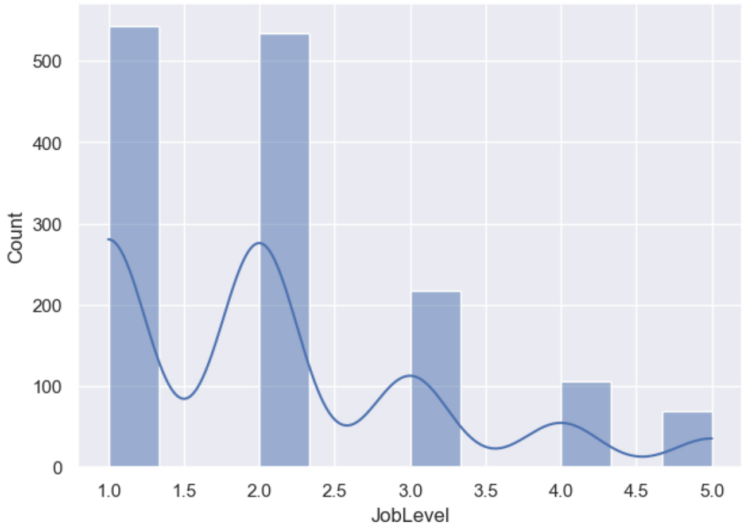


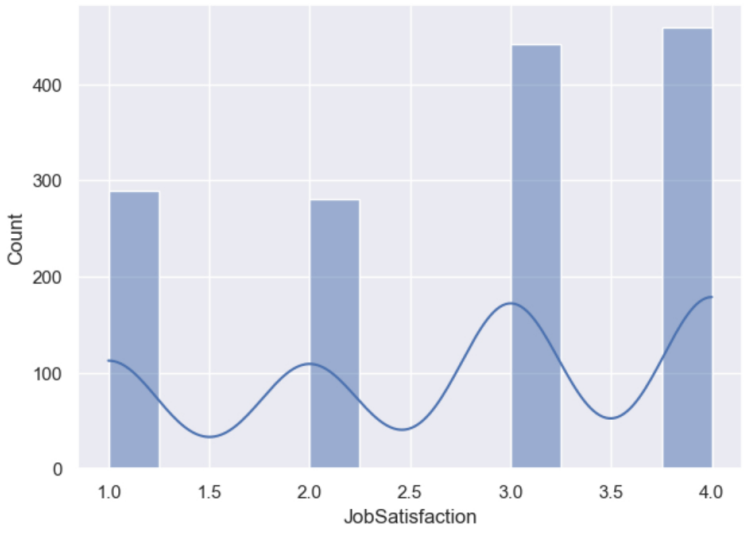


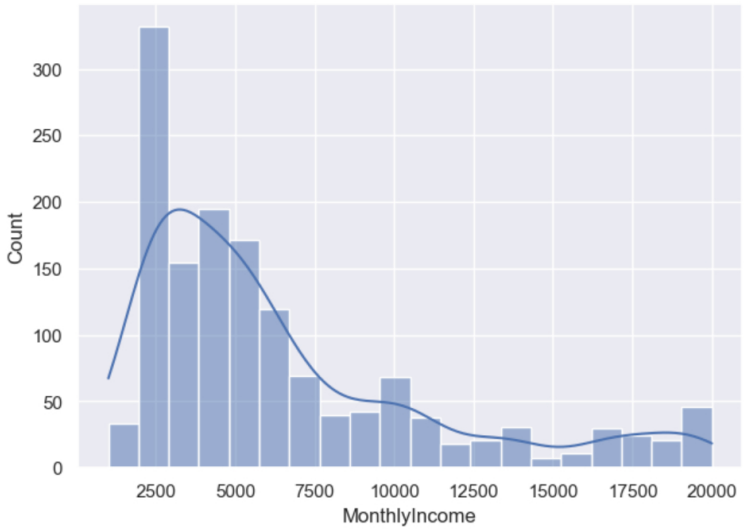


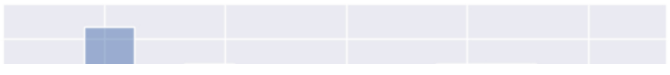


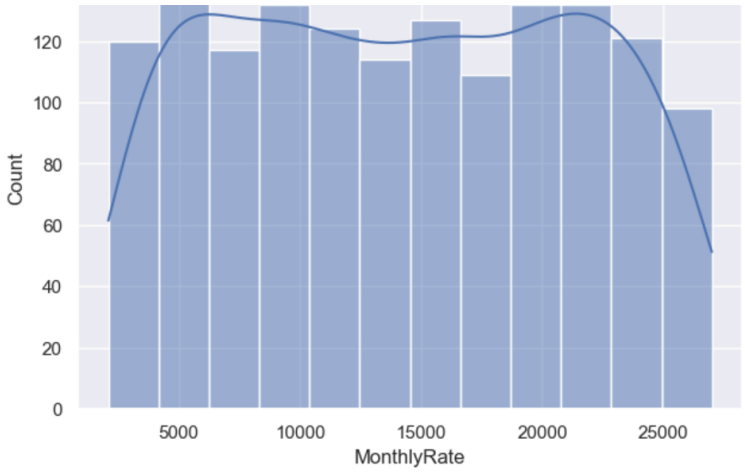

200
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
JobInvolvement

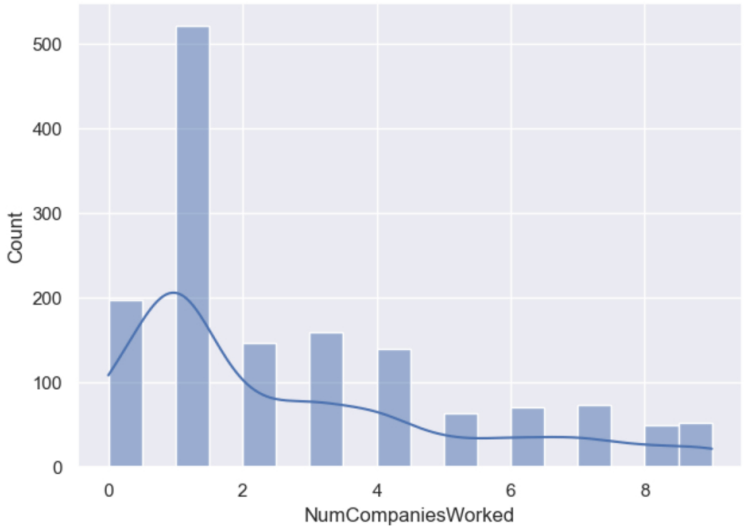

500
400
300
200
100
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
JobLevel

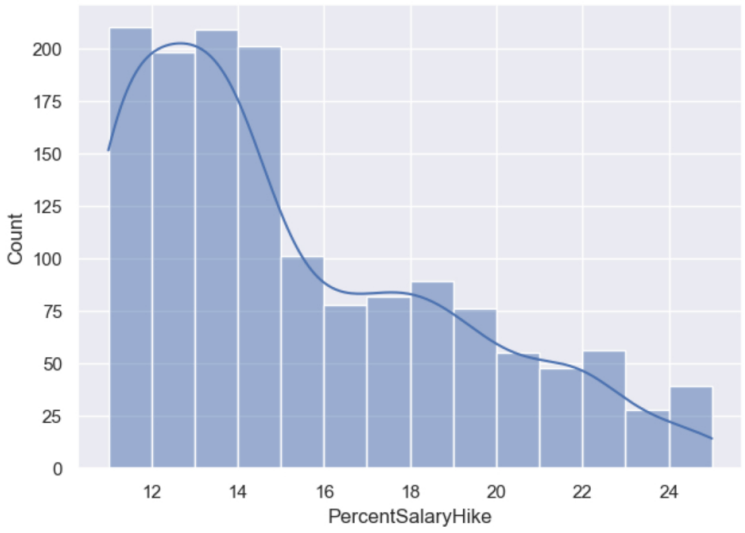

400
300
200
100
0
Count
1.0
1.5
2.0
2.5
3.0
3.5
4.0
JobSatisfaction


300
250
200
150
100
50
0
Count
2500
5000
7500
10000
12500
15000
17500
20000
MonthlyIncome

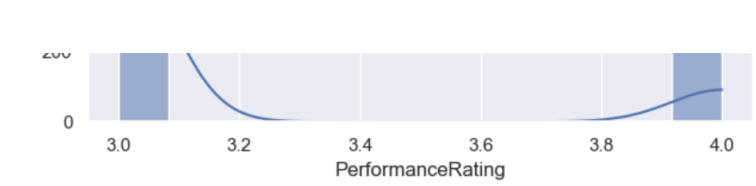

140

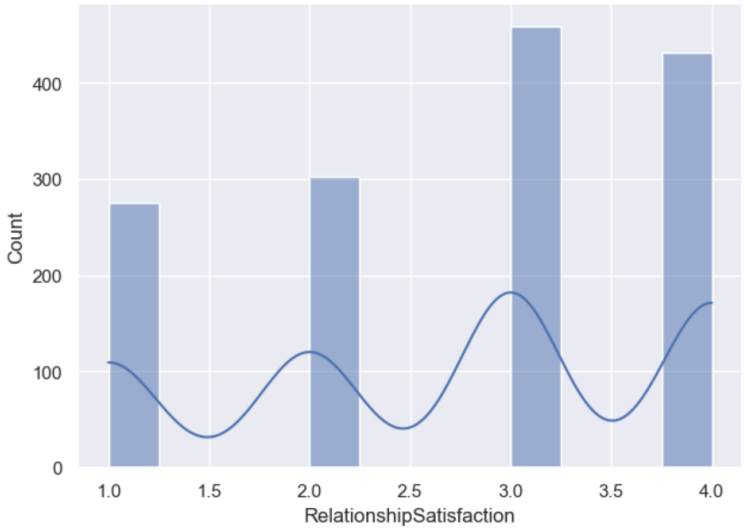

Count
MonthlyRate
0
20
40
60
80
100
120
5000
10000
15000
20000
25000

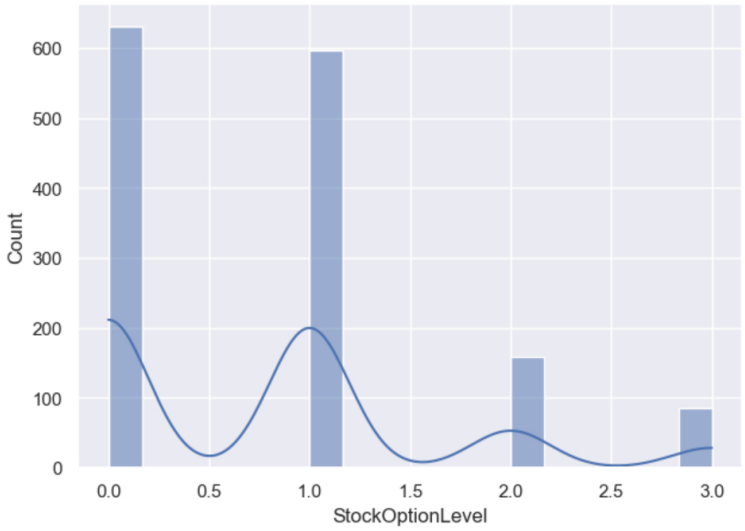

Count
NumCompaniesWorked
0
100
200
300
400
500
0
2
4
6
8

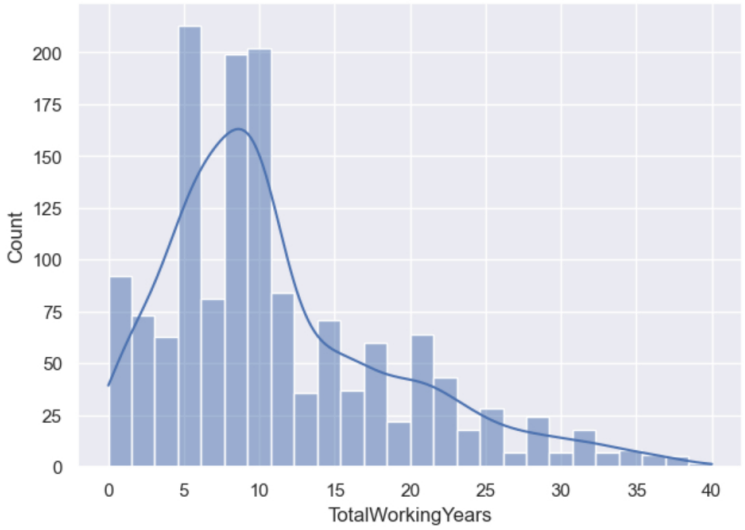

Count
PercentSalaryHike
0
25
50
75
100
125
150
175
200
12
14
16
18
20
22
24

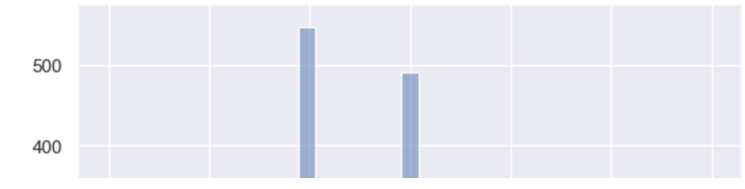

Count
0
200
400
600
800
1000
1200

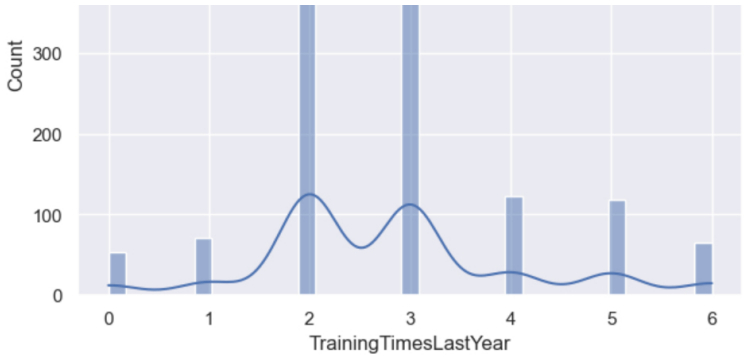

0
3.0
3.2
3.4
3.6
3.8
4.0
PerformanceRating

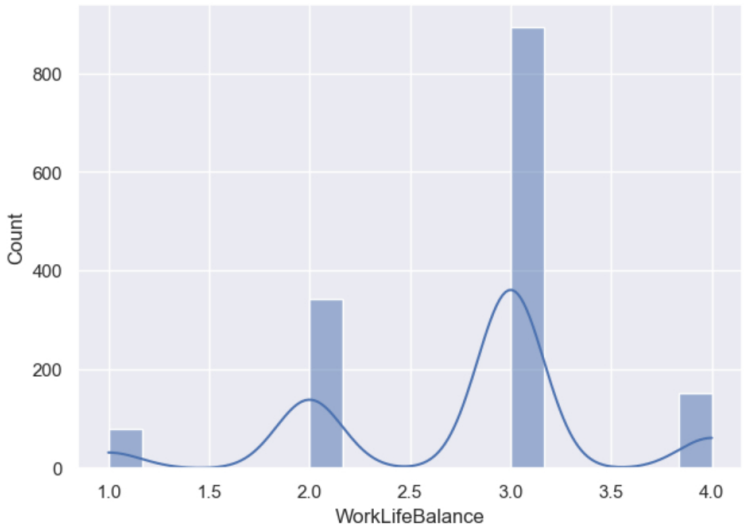

Count
400
300
200
100
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
RelationshipSatisfaction

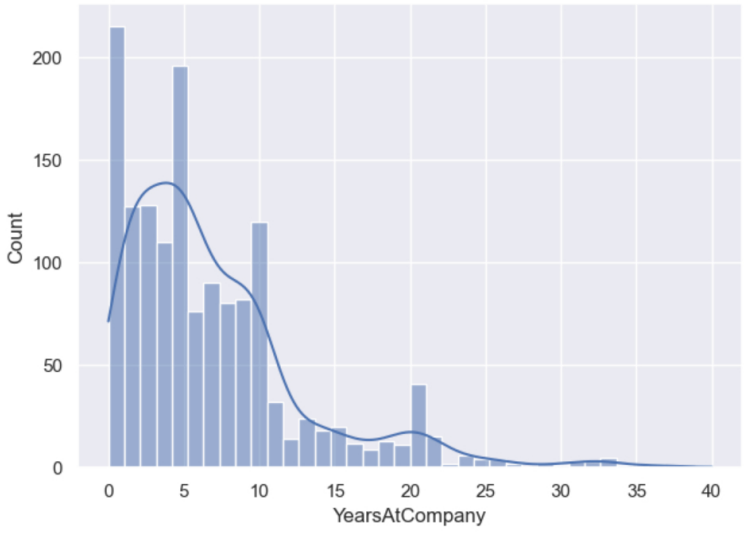

Count
600
500
400
300
200
100
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
StockOptionLevel

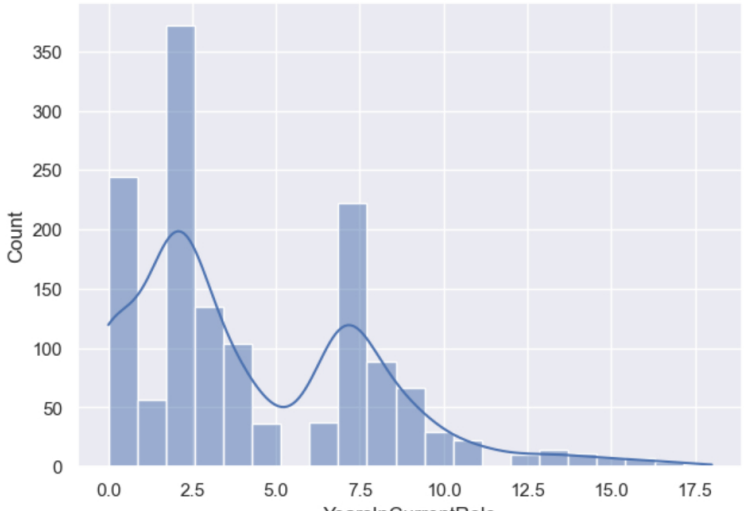

Count
200
175
150
125
100
75
50
25
0
0
5
10
15
20
25
30
35
40
TotalWorkingYears

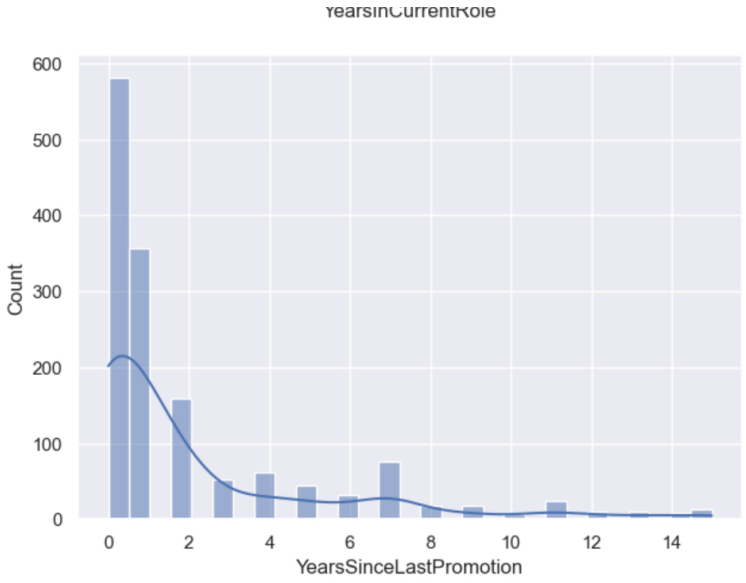

500
400

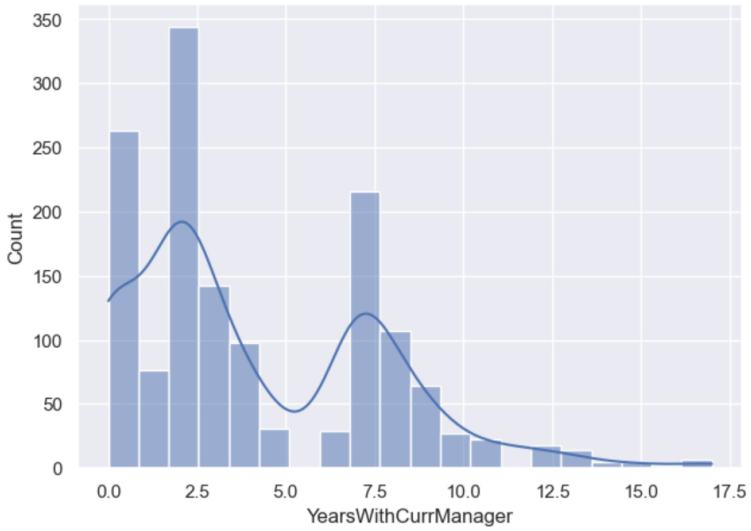

Count
300
200
100
0
0
1
2
3
4
5
6
TrainingTimesLastYear


Count
800
600
400
200
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
WorkLifeBalance

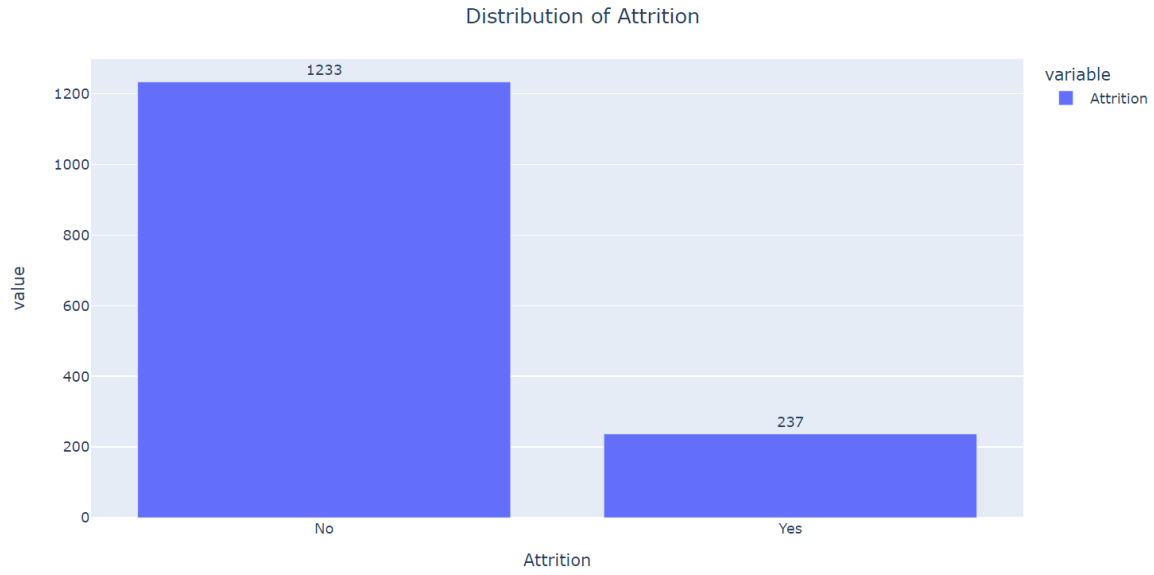

Count
200
150
100
50
0
0
5
10
15
20
25
30
35
40
YearsAtCompany

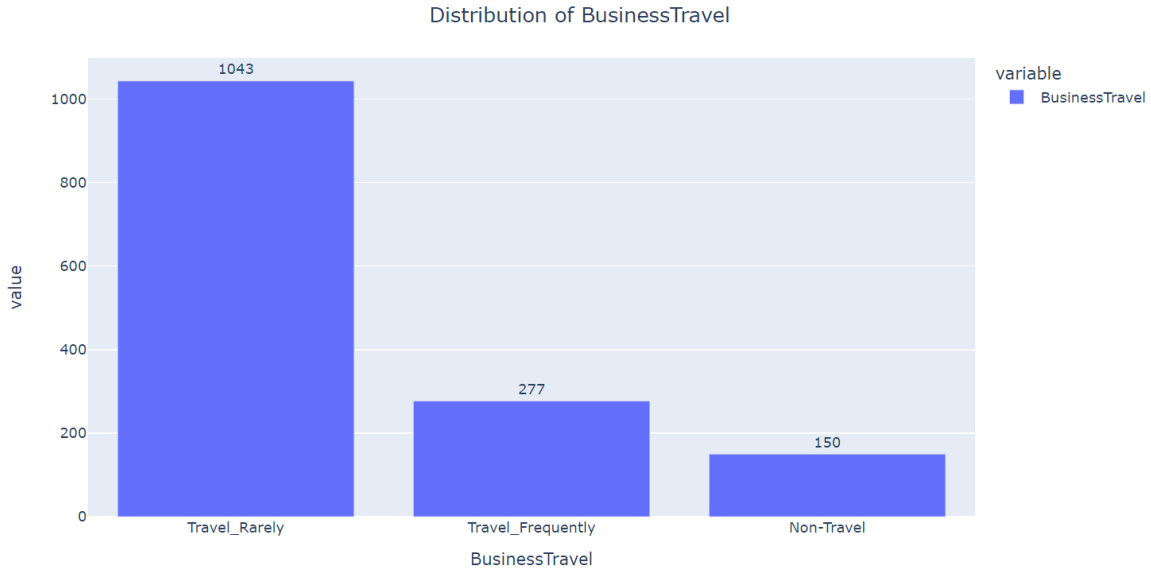

Count
350
300
250
200
150
100
50
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

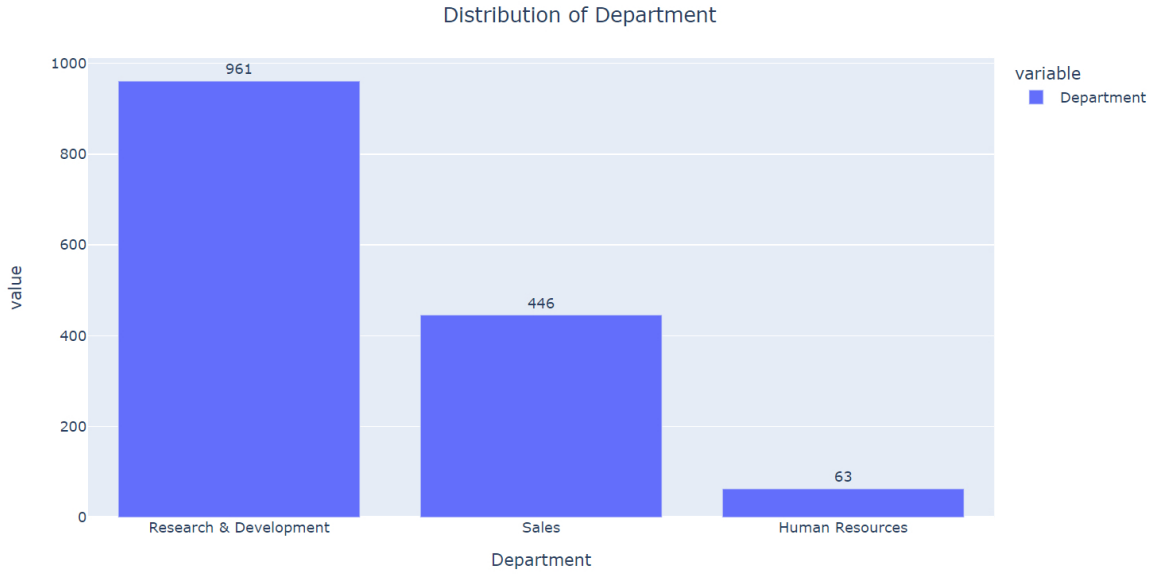


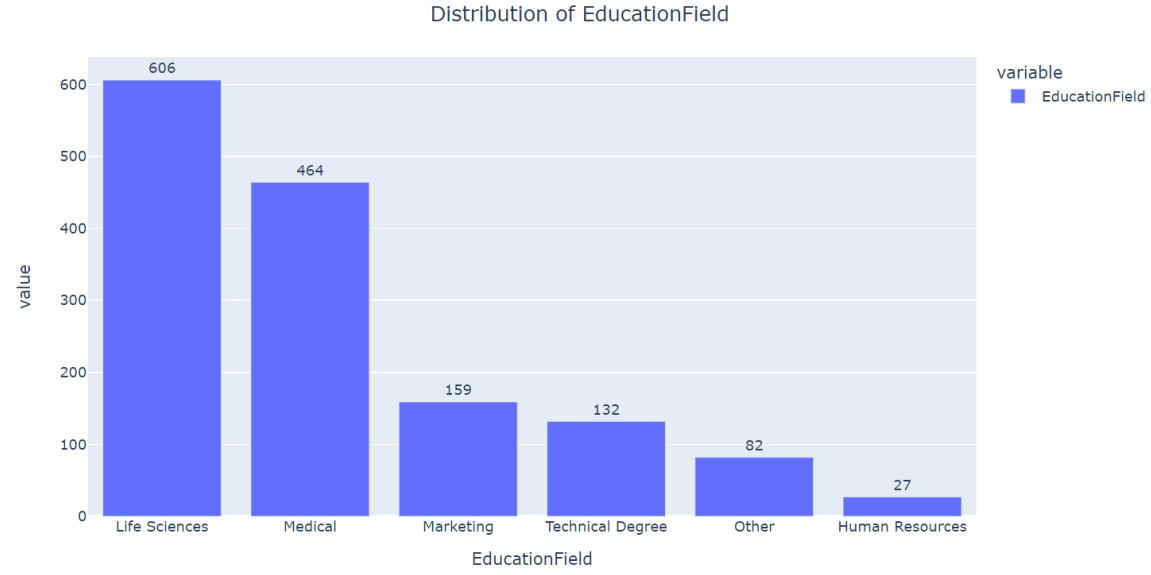


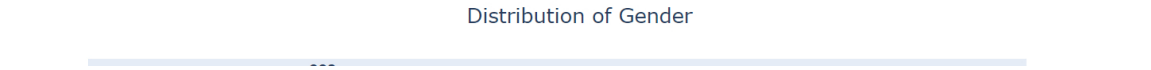
In [16]:
```
## object columns
for i in obj_col:

    fig=px.bar(data_frame=hr[i].value_counts(),text_auto=True)
    fig.update_layout(title="Distribution of " + i)
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

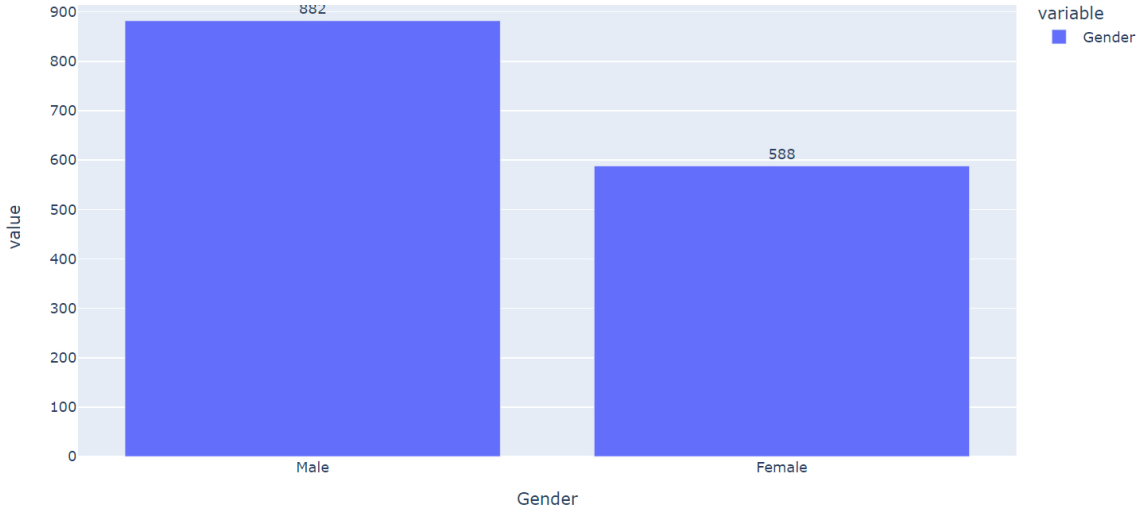



Distribution of BusinessTravel
variable
BusinessTravel
value
1043
277
150
0
200
400
600
800
1000
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel

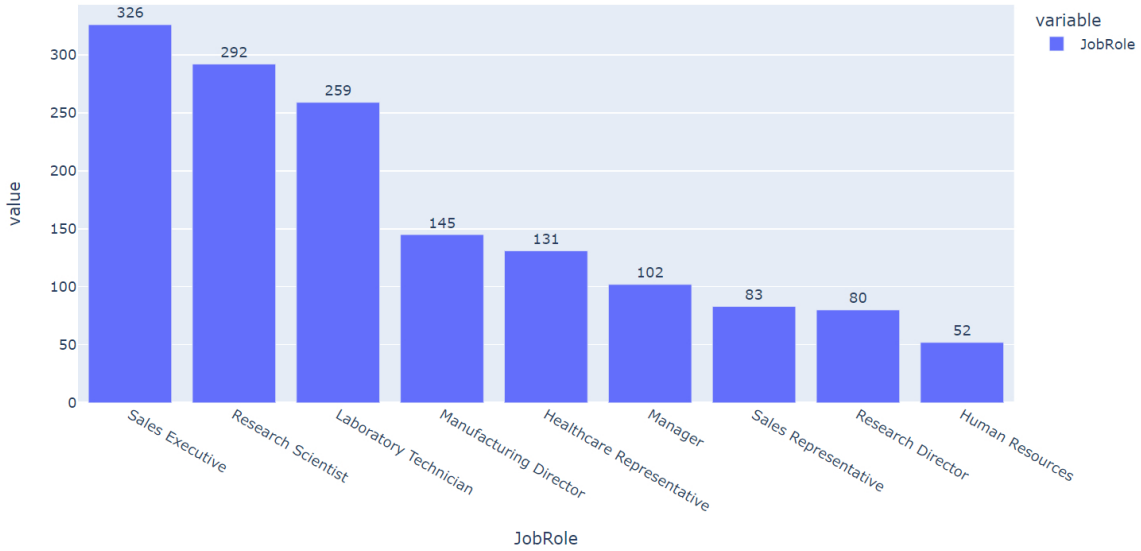

Distribution of Department
variable
Department
value
961
446
63
0
200
400
600
800
1000
Research & Development
Sales
Human Resources
Department


Distribution of EducationField
variable
EducationField
value
606
464
159
132
82
27
0
100
200
300
400
500
600
Life Sciences
Medical
Marketing
Technical Degree
Other
Human Resources
EducationField

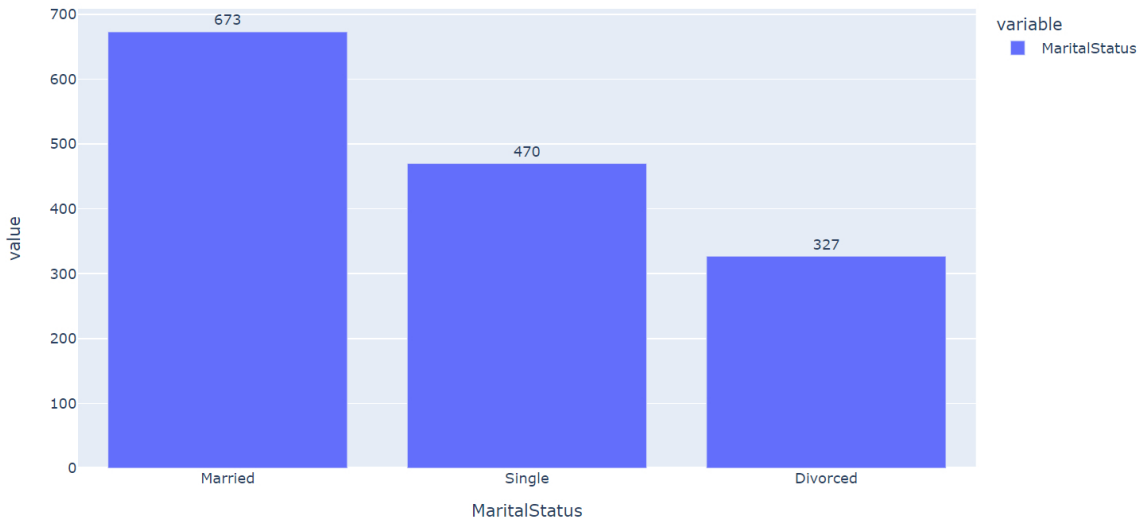

Distribution of Gender



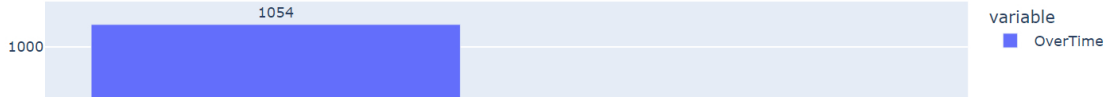
Distribution of JobRole

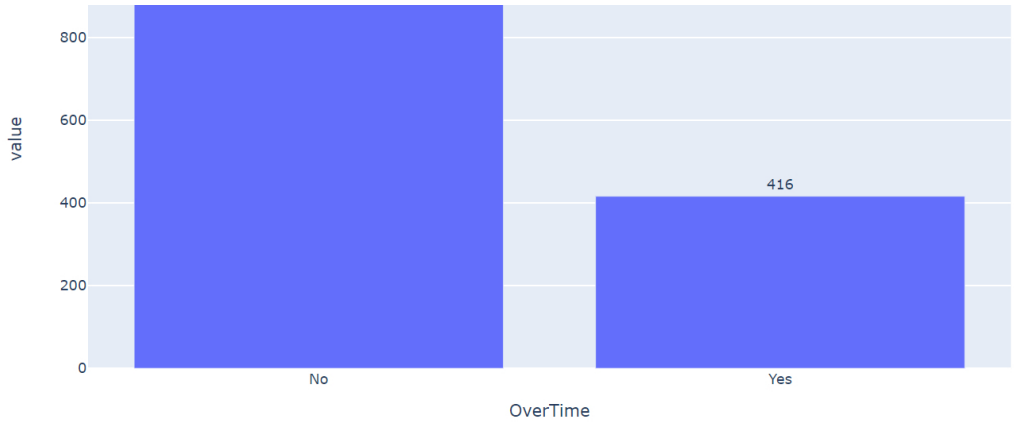


Distribution of MaritalStatus



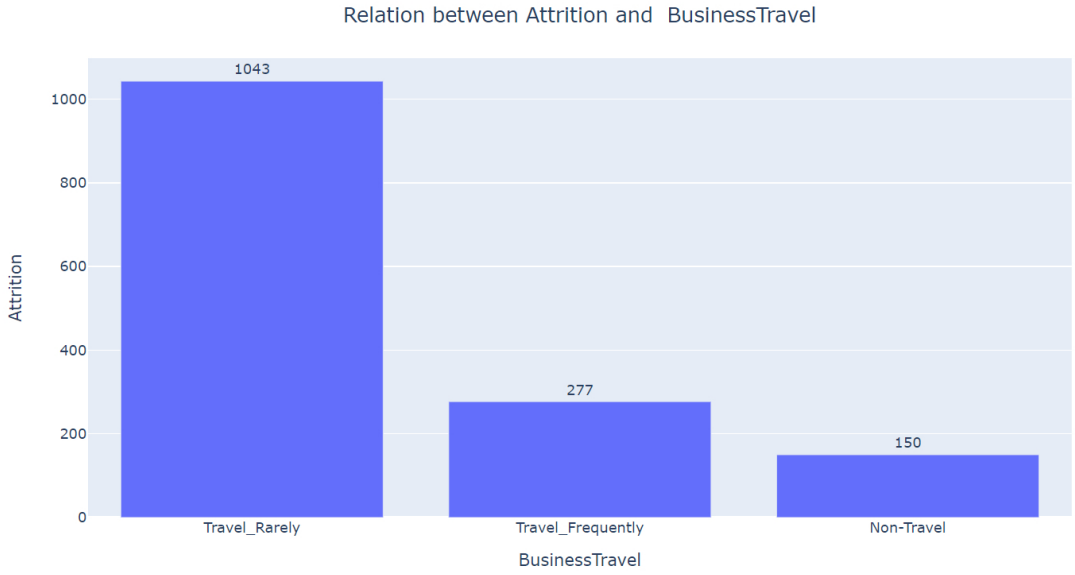
Distribution of OverTime

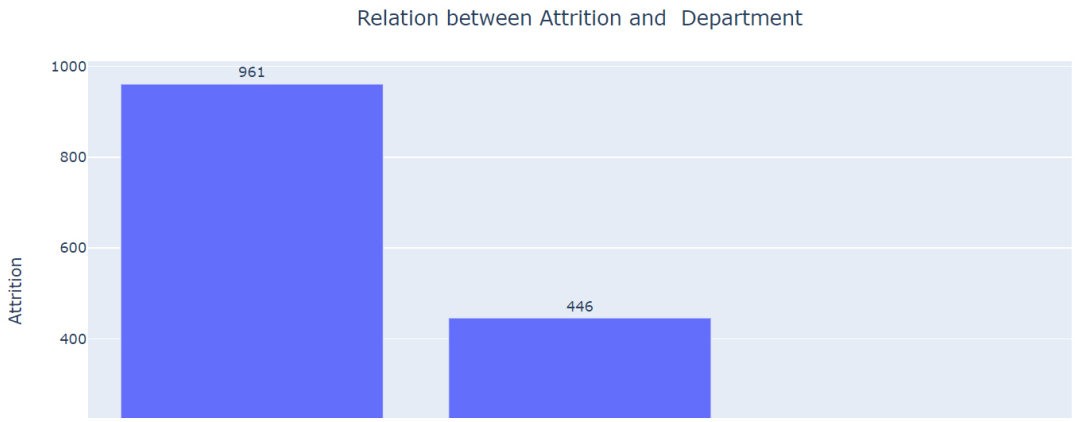




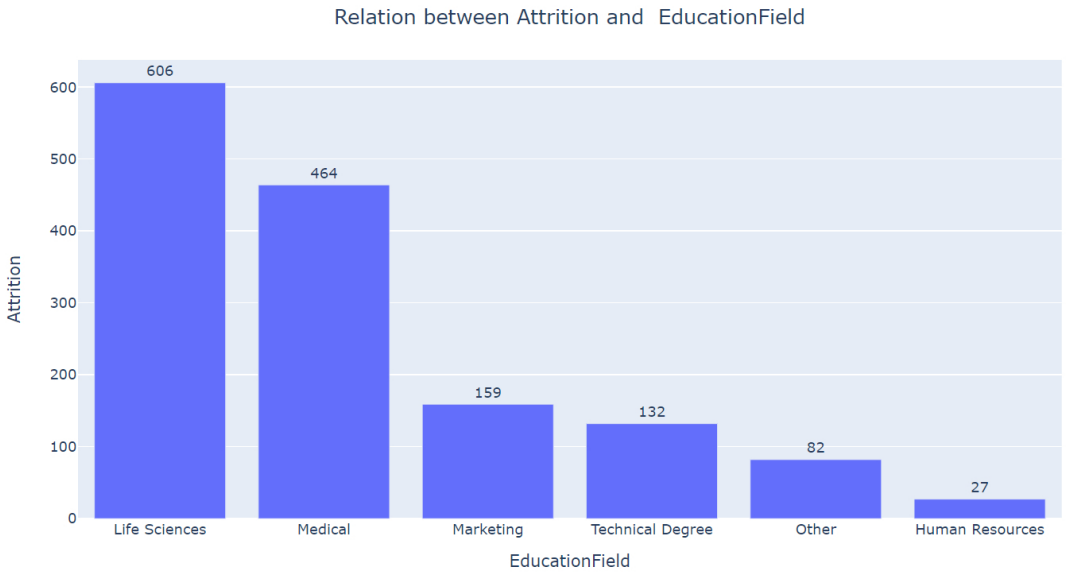
Bivariate Analyses

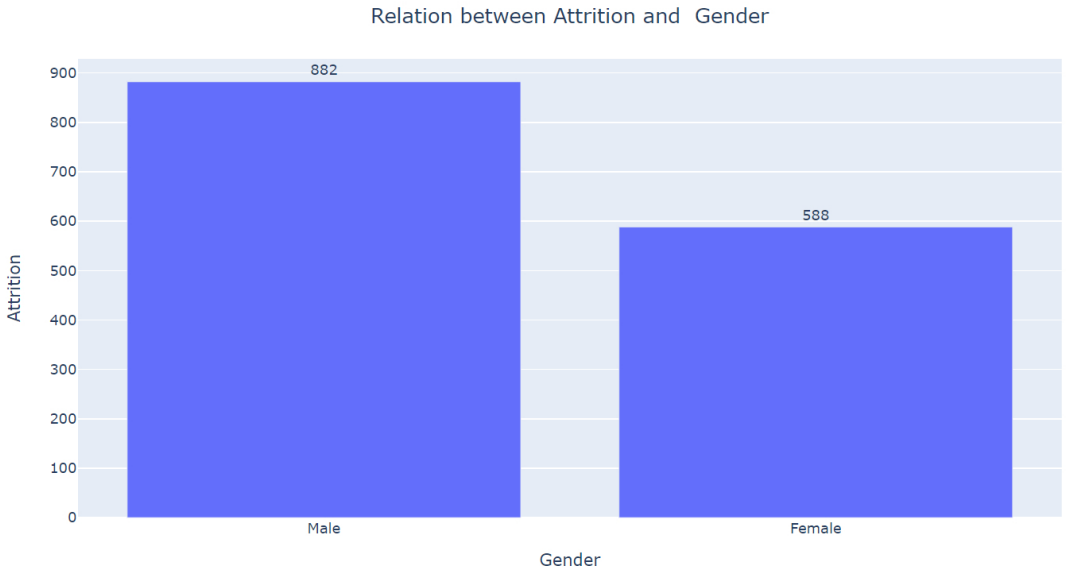
In [17]:
```
## Relation between Attrition and object columns
col=['BusinessTravel',
 'Department',
 'EducationField',
 'Gender',
 'JobRole',
 'MaritalStatus',
 'OverTime']
for i in col:
    ee=hr.groupby(i)['Attrition'].count().reset_index().sort_values('Attrition',ascending=False)
    fig = px.bar(ee, x=i,y="Attrition",text_auto=True)
    fig.update_layout(title="Relation between Attrition and  " + i)
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

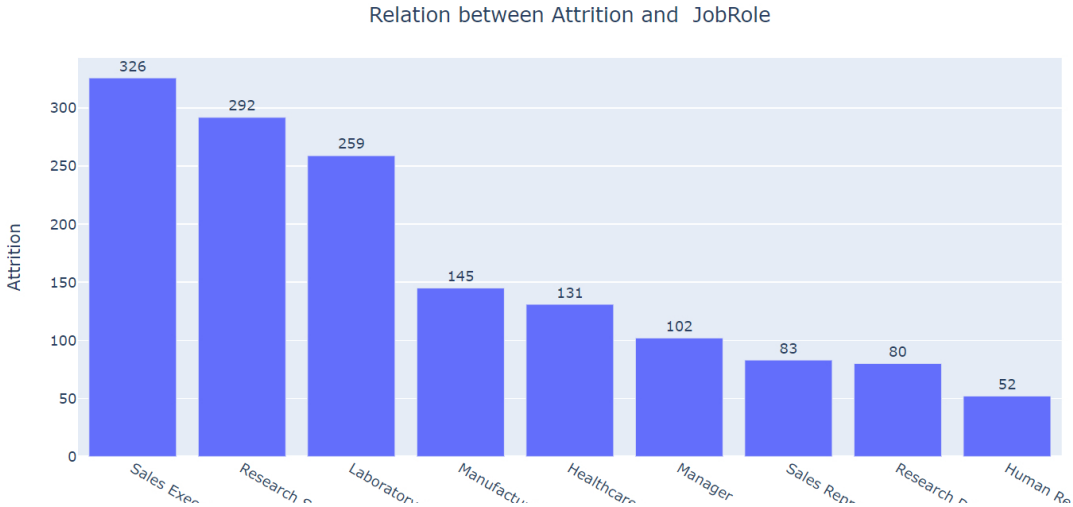


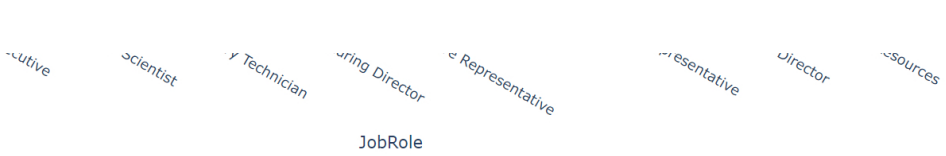


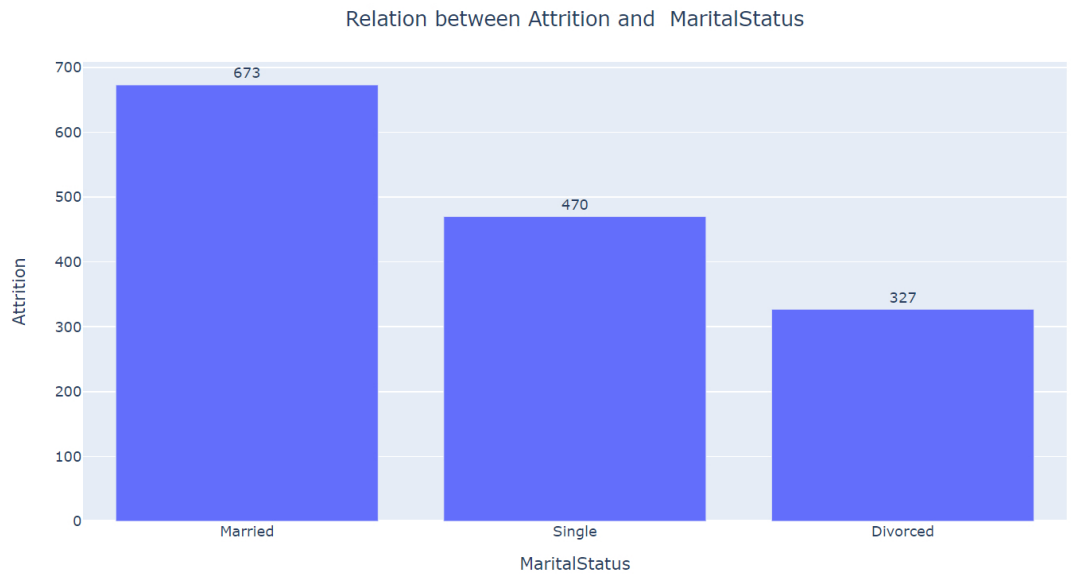

200
0
63
Research & Development
Sales
Human Resources
Department

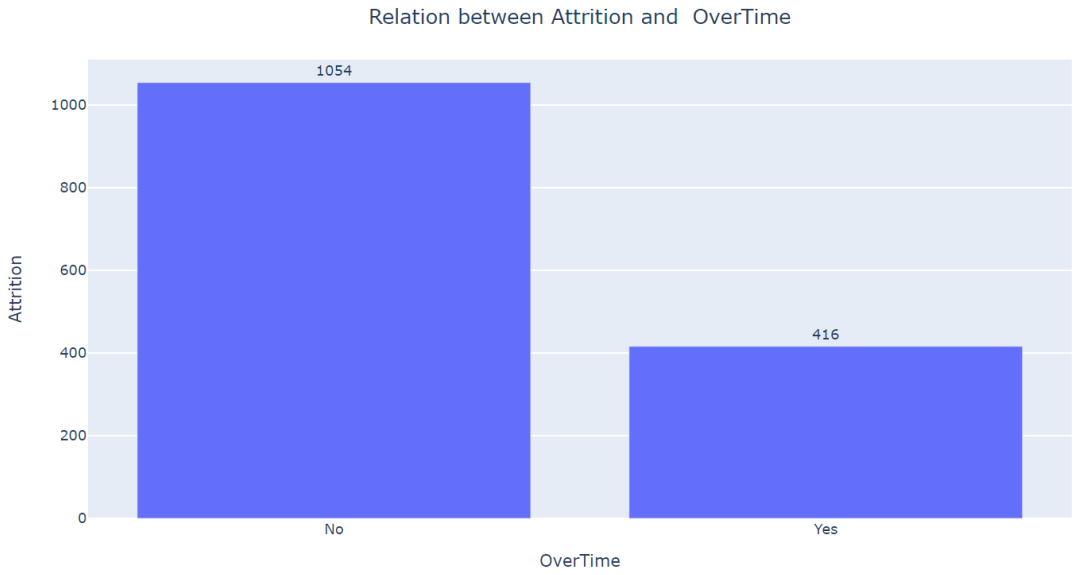

Relation between Attrition and EducationField
606
464
159
132
82
27
600
500
400
300
200
100
0
Attrition
Life Sciences
Medical
Marketing
Technical Degree
Other
Human Resources
EducationField


Relation between Attrition and Gender
882
588
900
800
700
600
500
400
300
200
100
0
Attrition
Male
Female
Gender

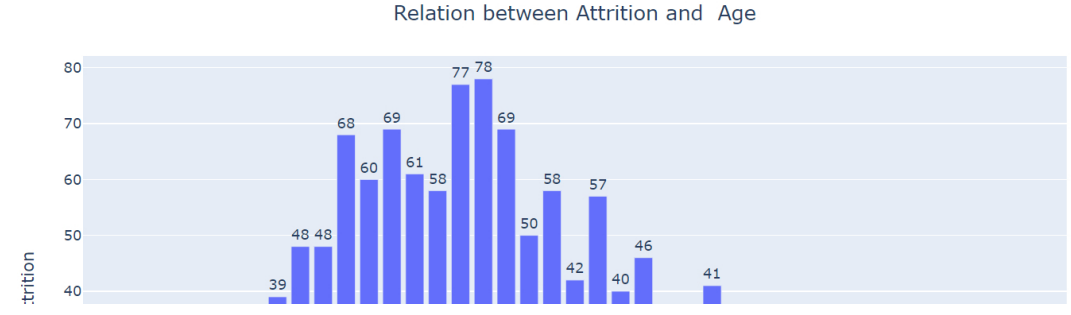

Relation between Attrition and JobRole
326
292
259
145
131
102
83
80
52
300
250
200
150
100
50
0
Attrition

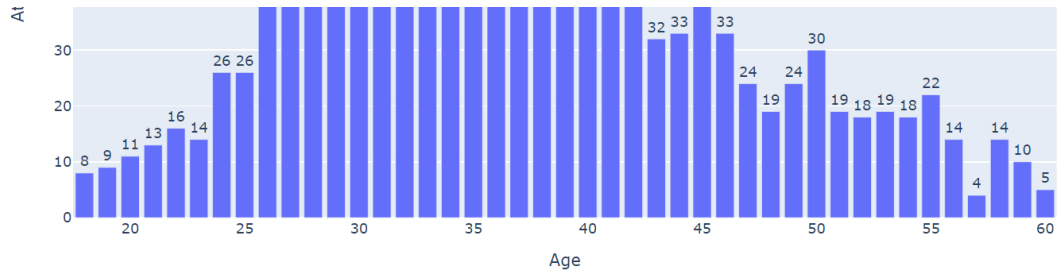




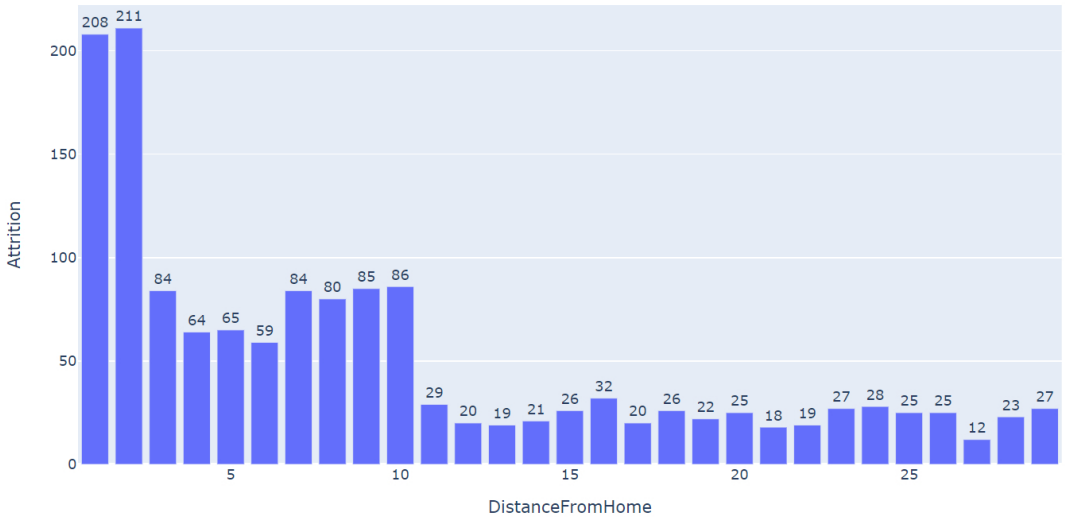


In [18]:
```
## Relation between Attrition and Numiric Columns
col=['Age','DistanceFromHome','Education','EnvironmentSatisfaction','JobInvolvement','JobLevel','JobSatisfaction','NumCompaniesWo
 'PercentSalaryHike','PerformanceRating','RelationshipSatisfaction','StockOptionLevel','TotalWorkingYears','TrainingTimesLastYear
 'WorkLifeBalance','YearsAtCompany','YearsInCurrentRole','YearsSinceLastPromotion','YearsWithCurrManager']
for i in col:
    ee=hr.groupby(i)['Attrition'].count().reset_index().sort_values('Attrition',ascending=False)
    fig = px.bar(ee, x=i,y="Attrition",text_auto=True)
    fig.update_layout(title="Relation between Attrition and  " + i)
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

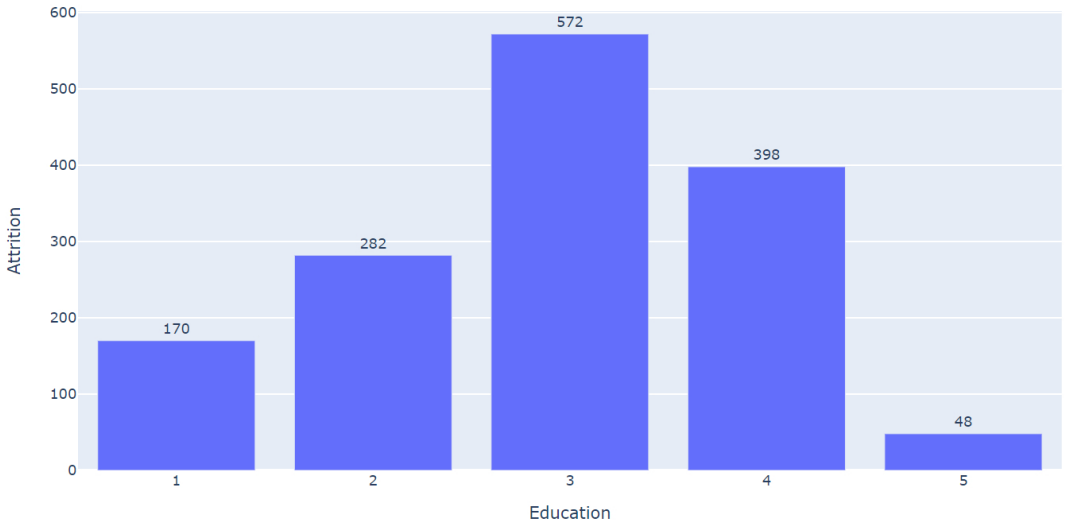




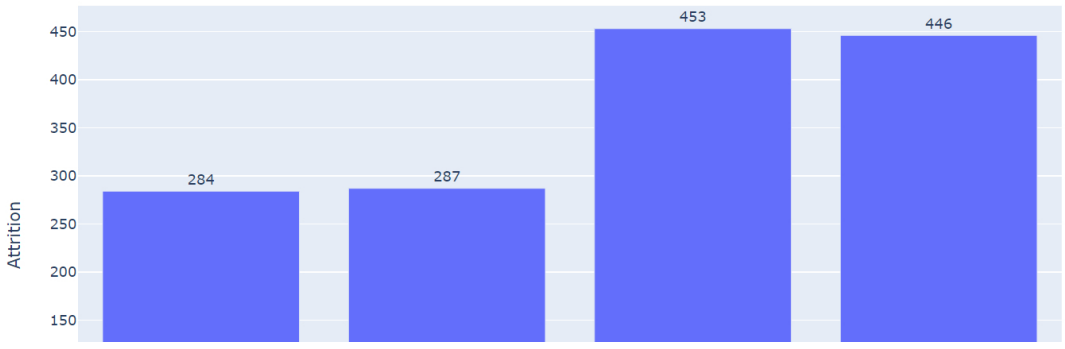
Relation between Attrition and DistanceFromHome

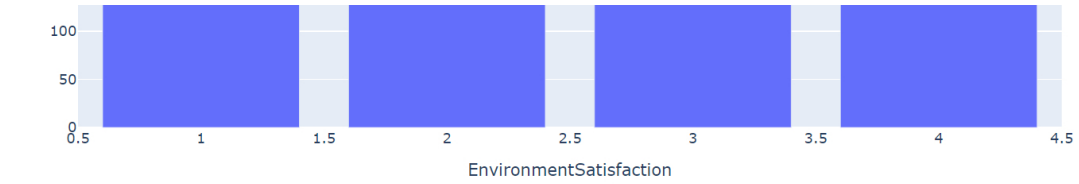


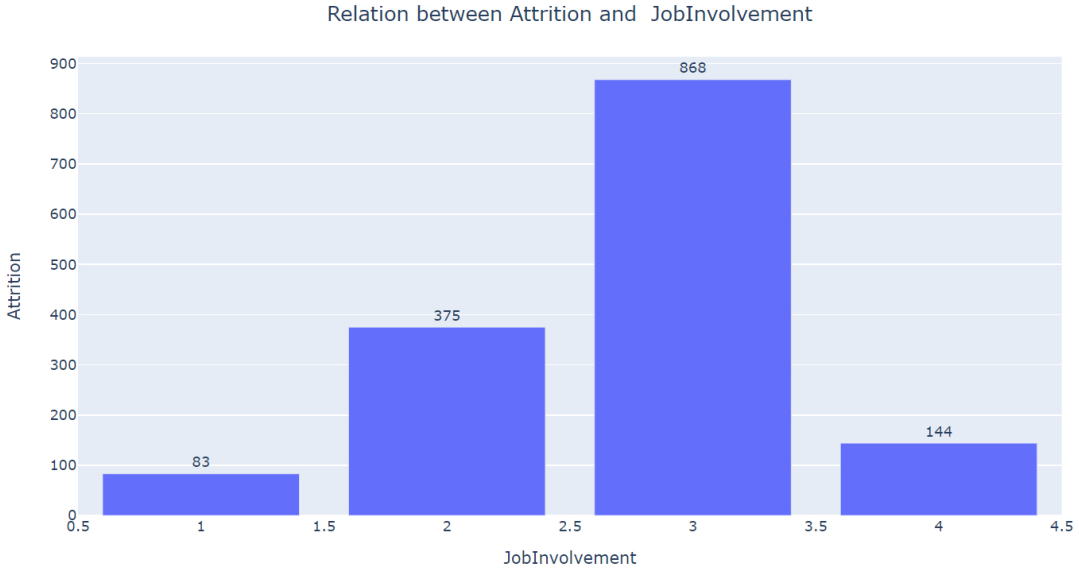
Relation between Attrition and Education

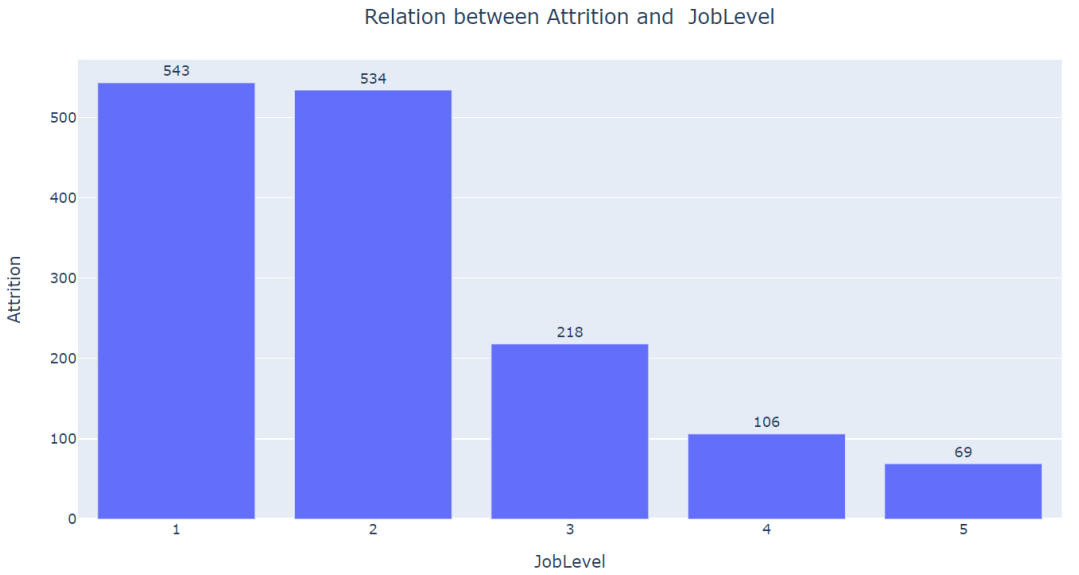


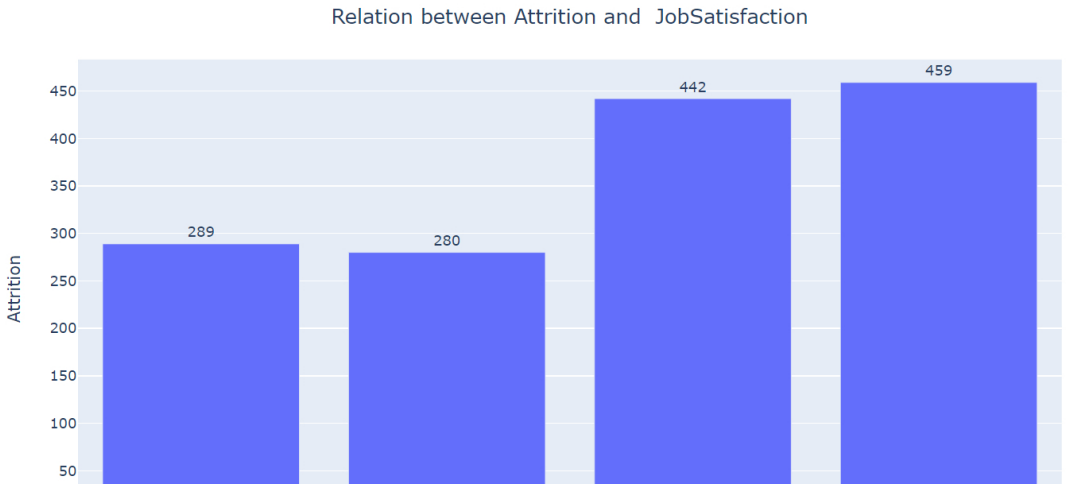
Relation between Attrition and EnvironmentSatisfaction



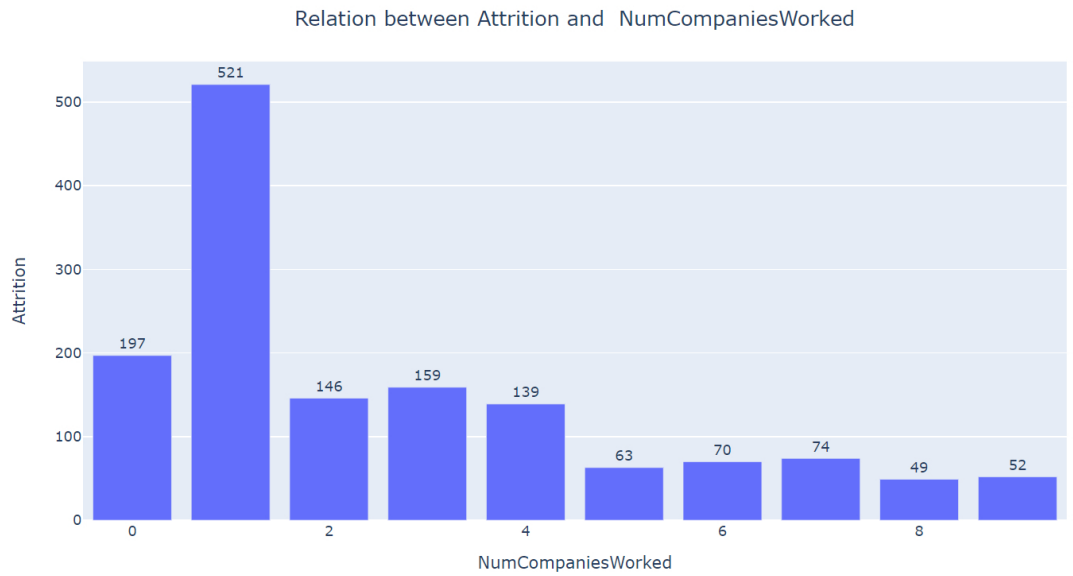

100
50
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
EnvironmentSatisfaction

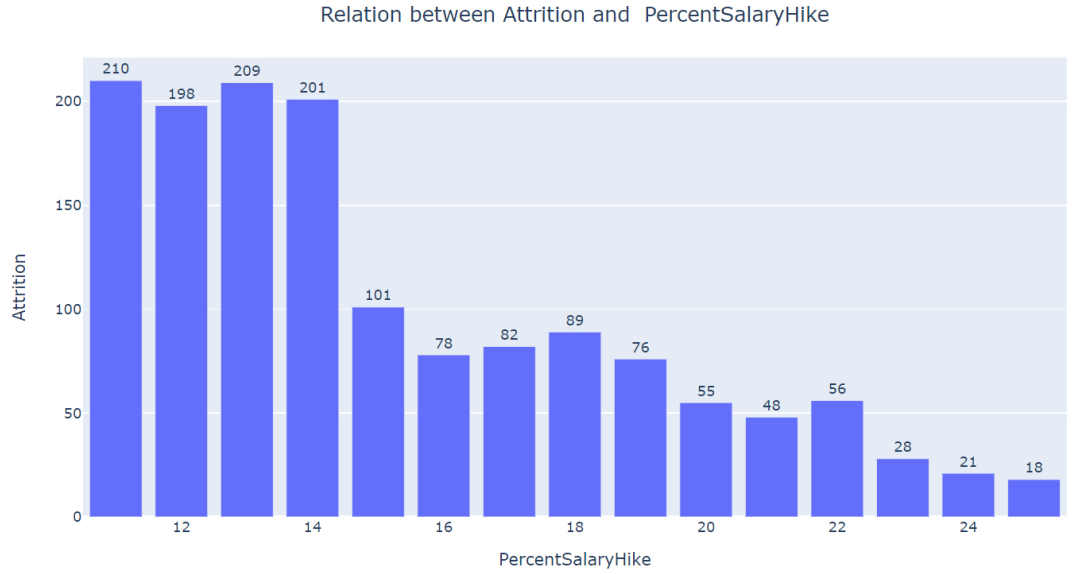

Relation between Attrition and JobInvolvement
900
800
700
600
500
400
300
200
100
0
Attrition
83
375
868
144
0.5
1
1.5
2
2.5
3
3.5
4
4.5
JobInvolvement

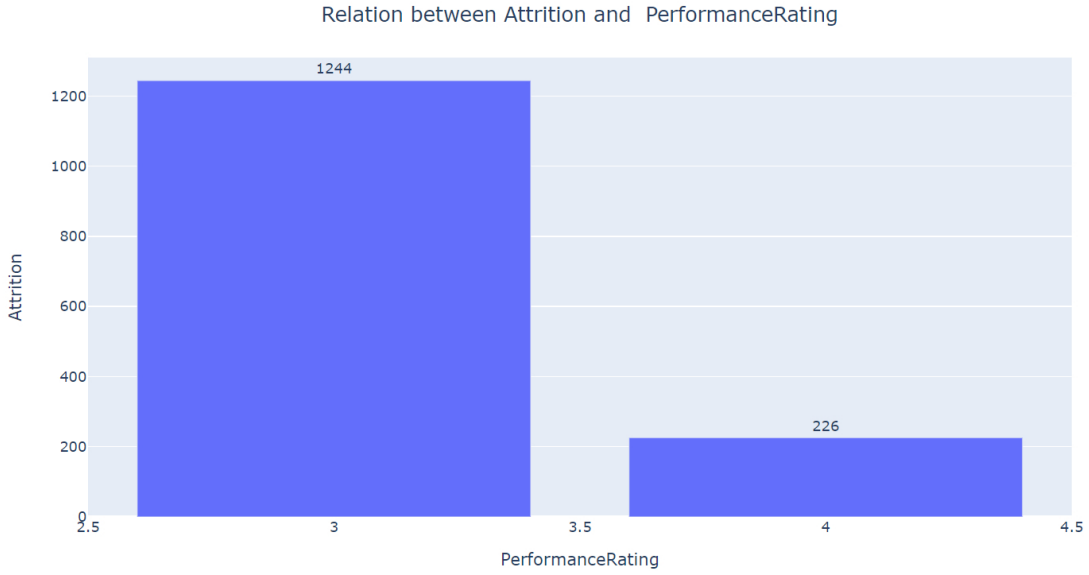

Relation between Attrition and JobLevel
500
400
300
200
100
0
Attrition
543
534
218
106
69
1
2
3
4
5
JobLevel

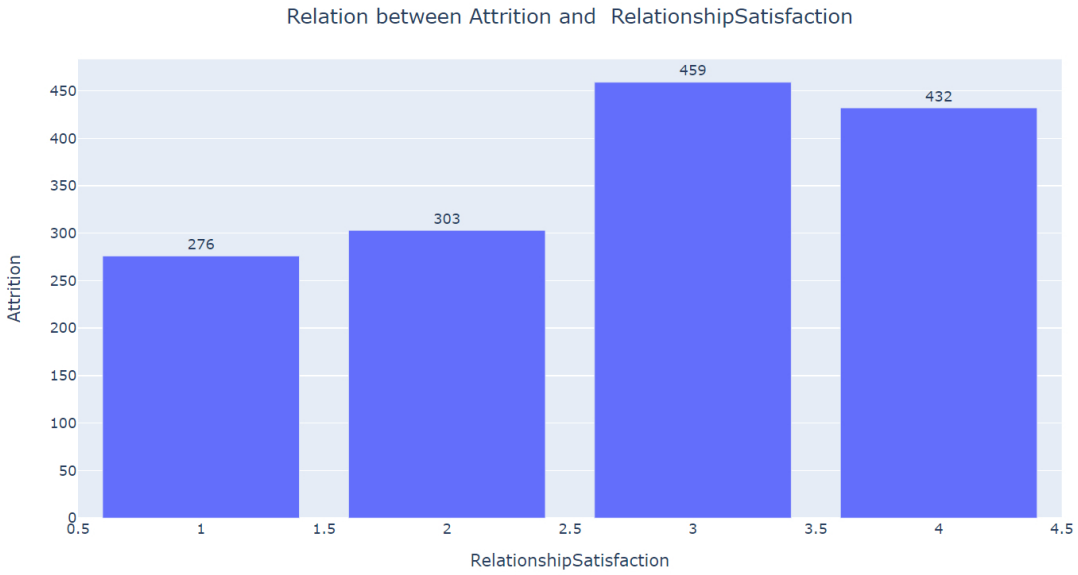

Relation between Attrition and JobSatisfaction
450
400
350
300
250
200
150
100
50
Attrition
289
280
442
459

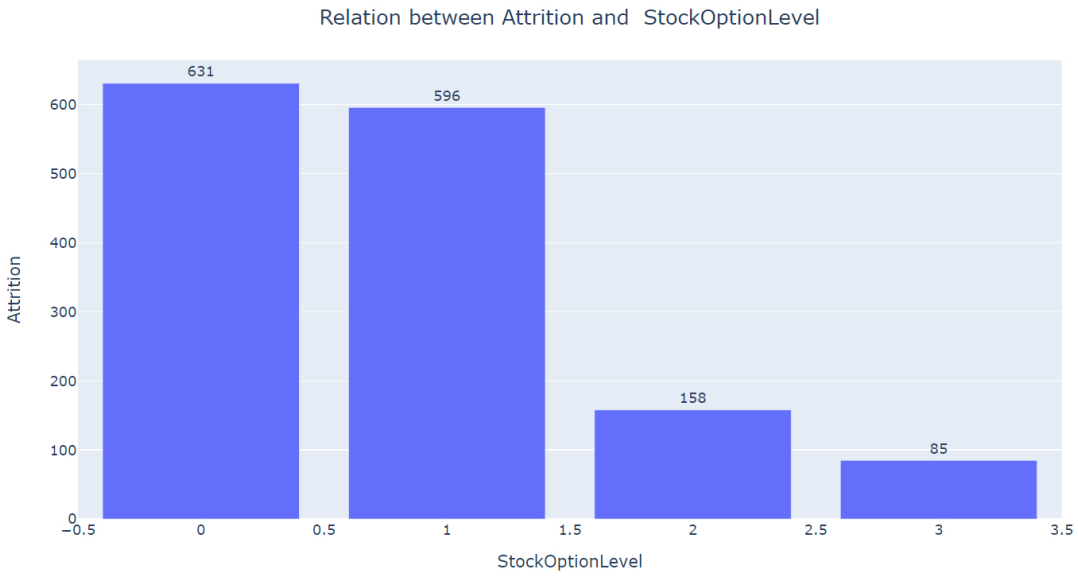

0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
JobSatisfaction

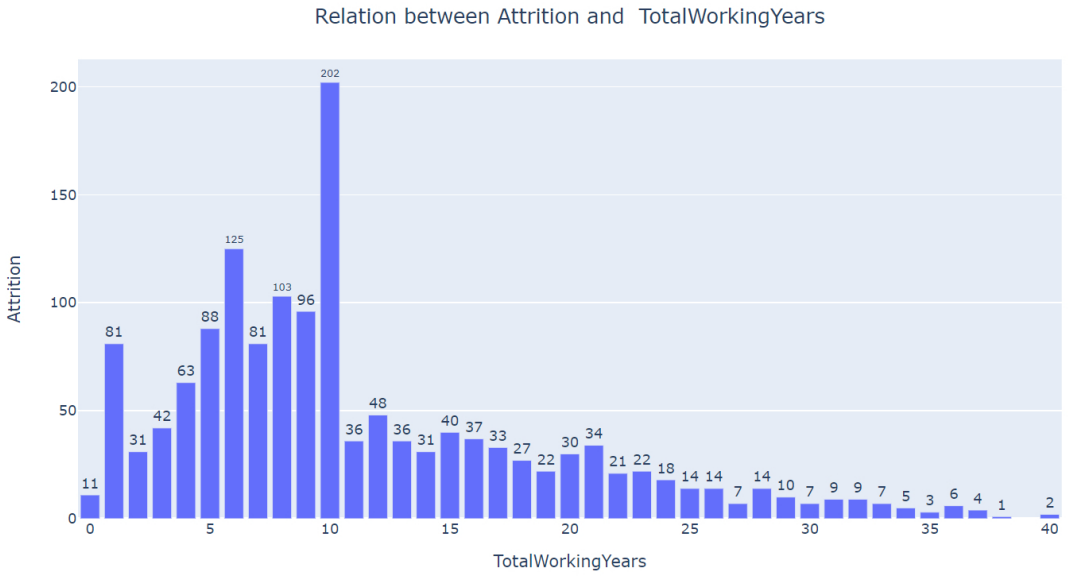

Relation between Attrition and NumCompaniesWorked
Attrition
0
100
200
300
400
500
197
521
146
159
139
63
70
74
49
52
0
2
4
6
8
NumCompaniesWorked


Relation between Attrition and PercentSalaryHike
Attrition
0
50
100
150
200
210
198
209
201
101
78
82
89
76
55
48
56
28
21
18
12
14
16
18
20
22
24
PercentSalaryHike

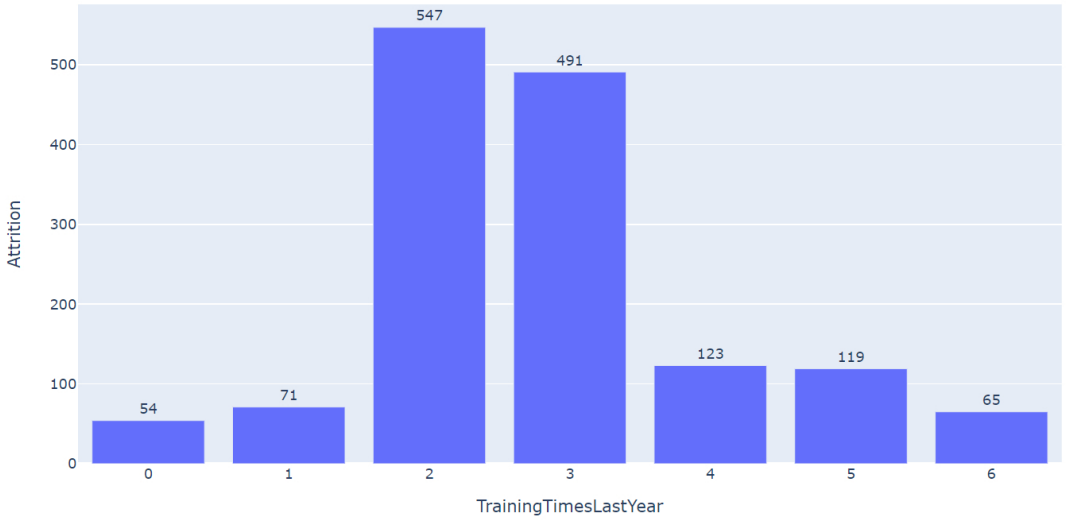

Relation between Attrition and PerformanceRating
Attrition
0
200
400
600
800
1000
1200
1244
226
2.5
3
3.5
4
4.5
PerformanceRating



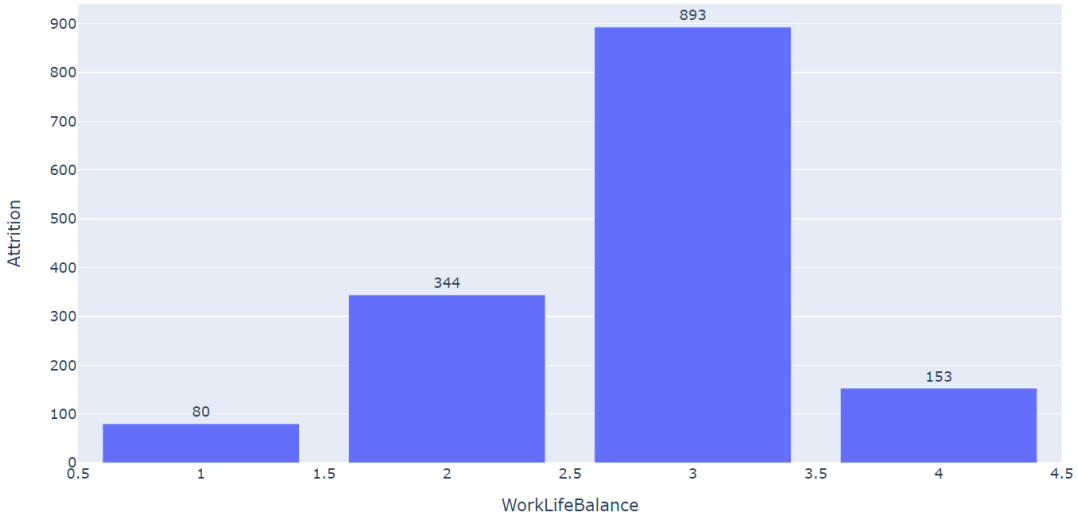




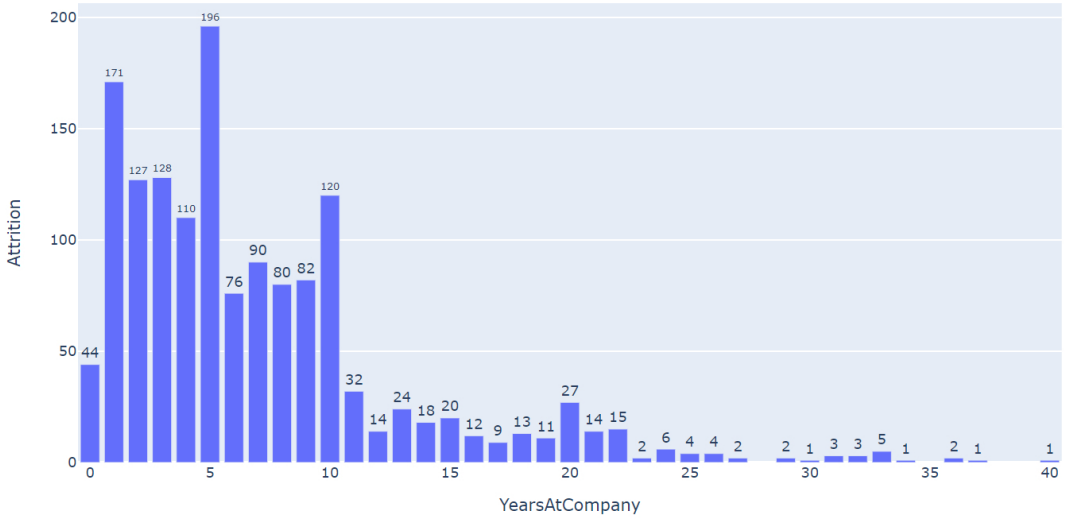
Relation between Attrition and TrainingTimesLastYear



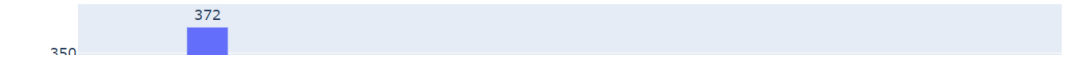
Relation between Attrition and WorkLifeBalance

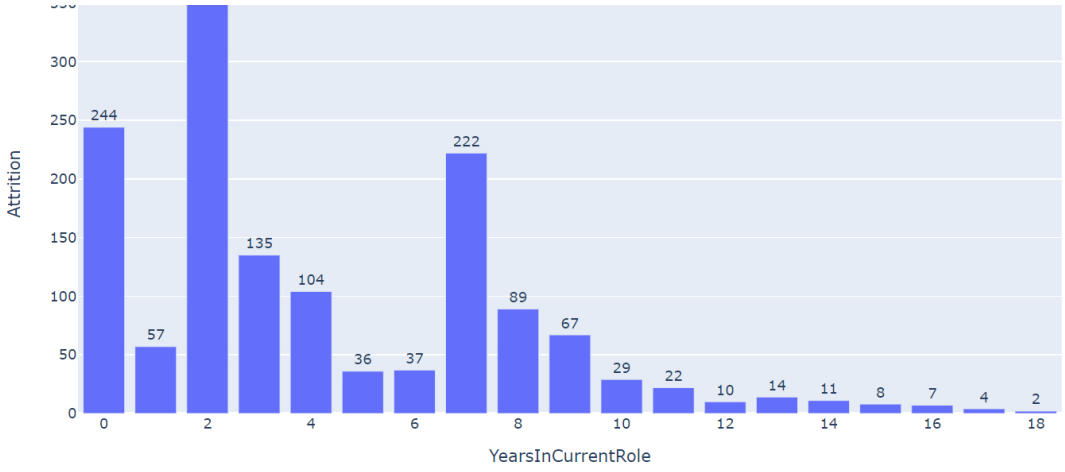


Relation between Attrition and YearsAtCompany

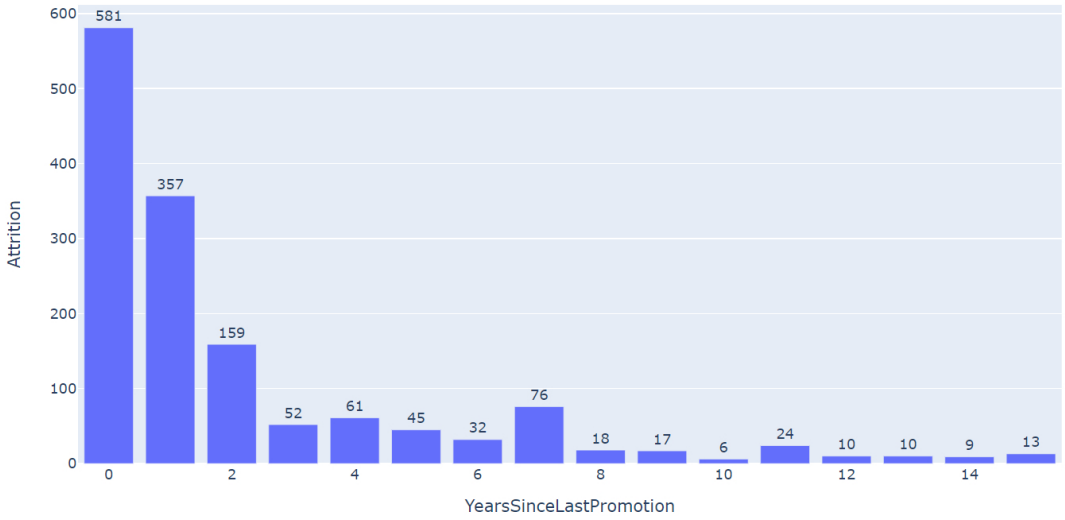


Relation between Attrition and YearsInCurrentRole





Relation between Attrition and YearsSinceLastPromotion



Relation between Attrition and YearsWithCurrManager

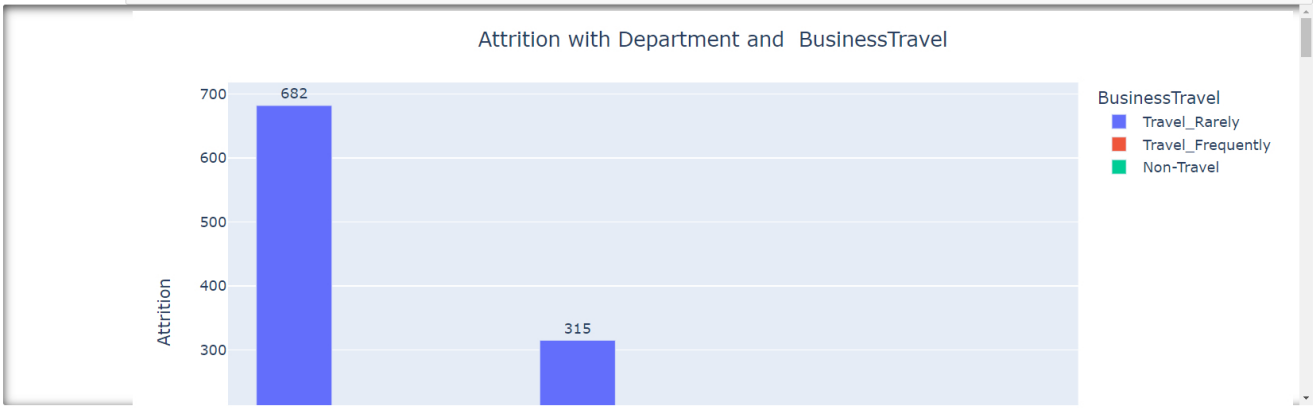
Multivariate Analyses

In [19]:
```
obj_col
```

Out[19]:
```
['Attrition',
 'BusinessTravel',
 'Department',
 'EducationField',
```

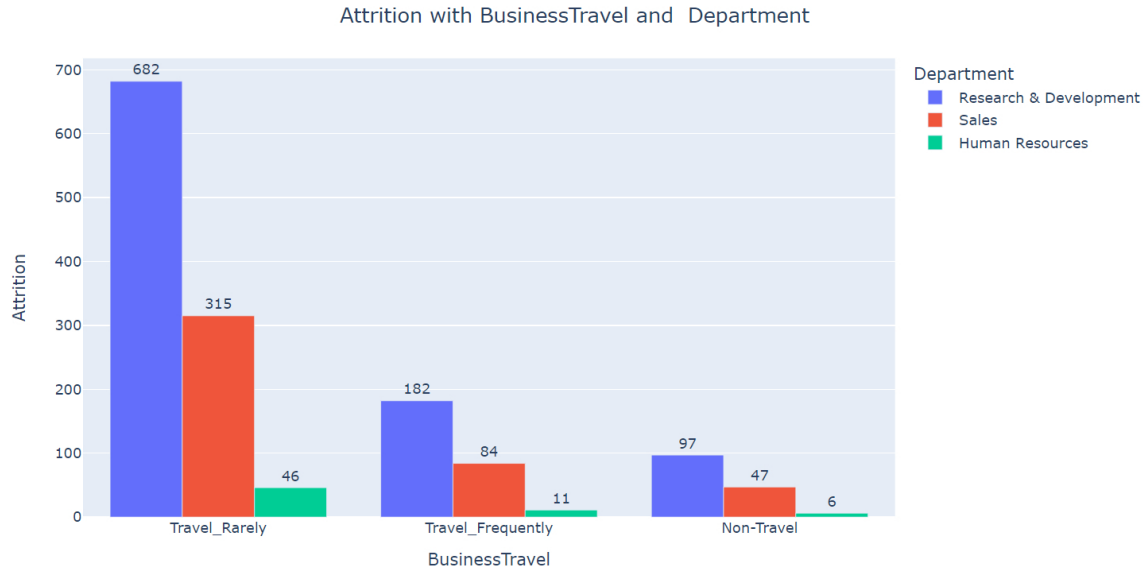
```
'Gender',
'JobRole',
'MaritalStatus',
'OverTime']
```

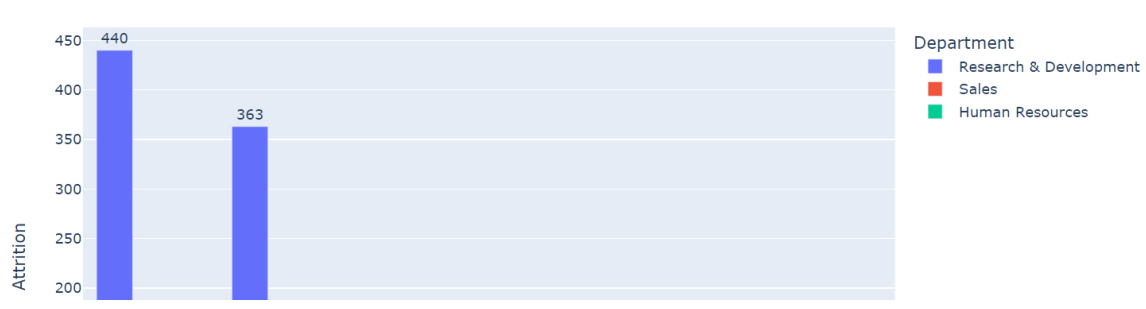
In [20]:
```
col=['Department','EducationField','Gender','JobRole','MaritalStatus','OverTime']

for i in col:
    be=hr.groupby([i,'BusinessTravel'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='BusinessTravel',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  BusinessTravel')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

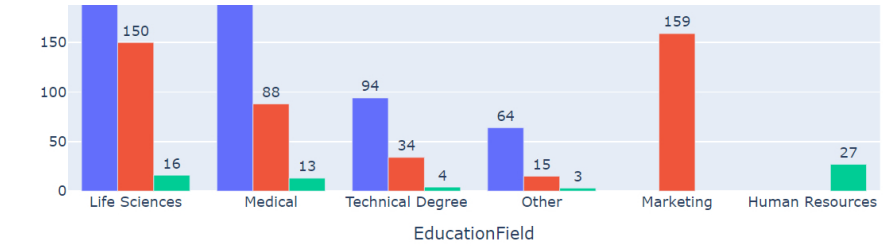


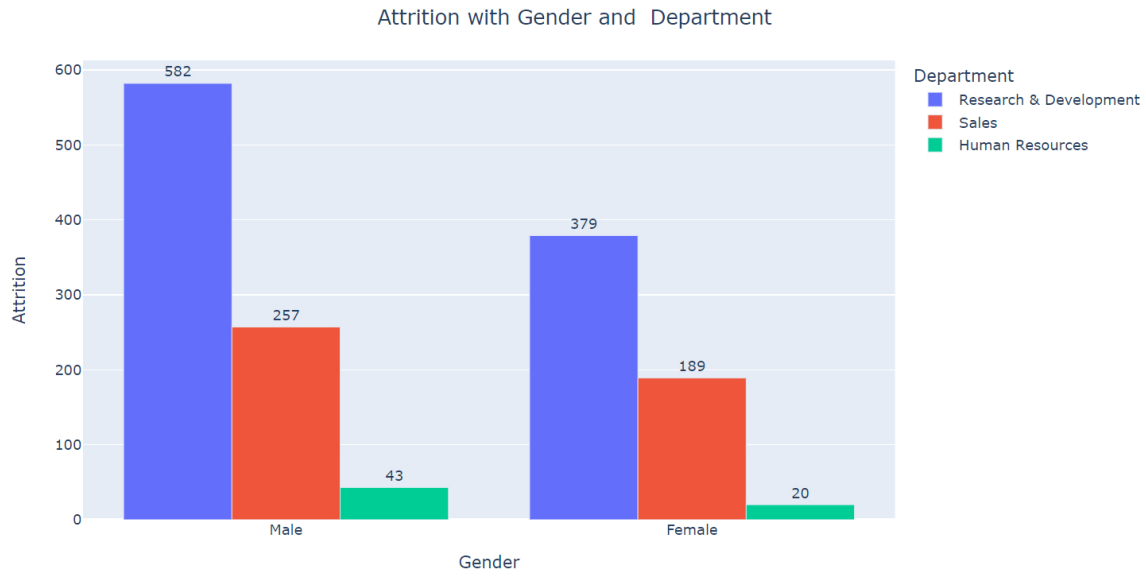
In [ ]:

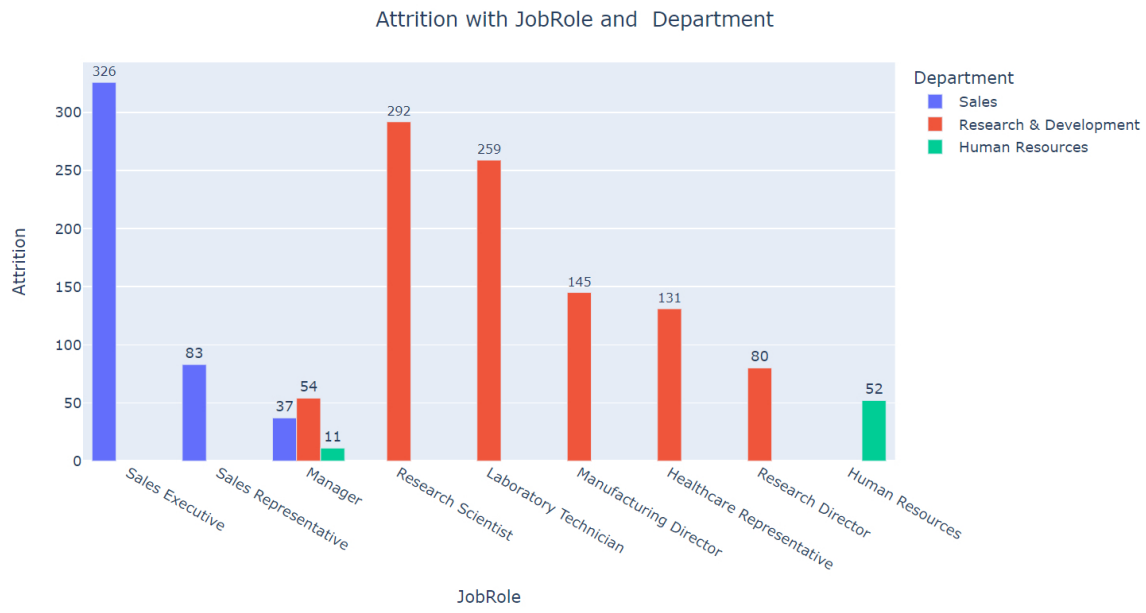
In [21]:
```
col=['BusinessTravel','EducationField','Gender','JobRole','MaritalStatus','OverTime']

for i in col:
    be=hr.groupby([i,'Department'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='Department',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  Department')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

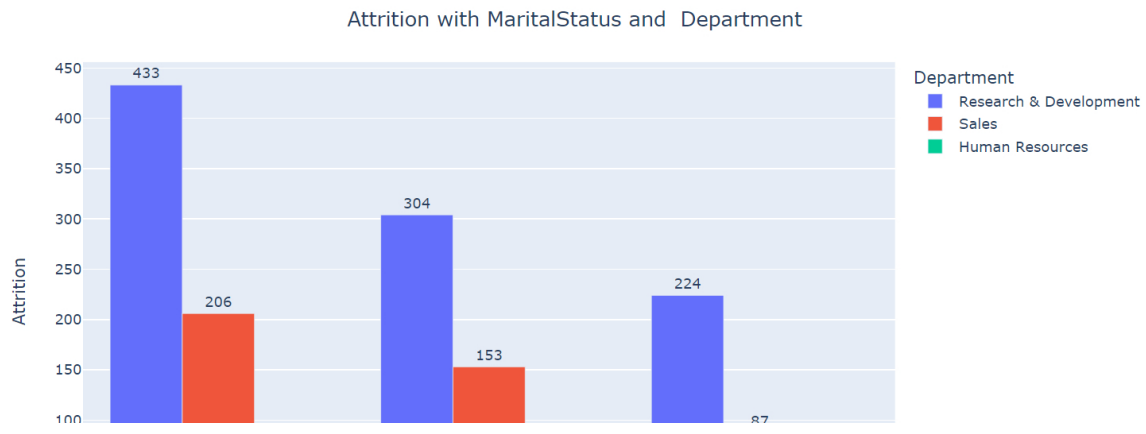


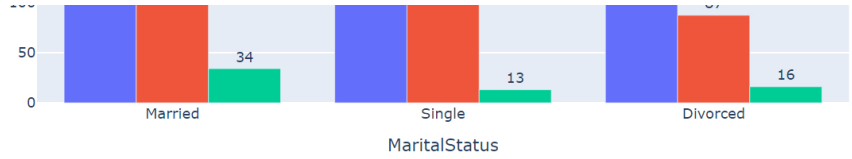


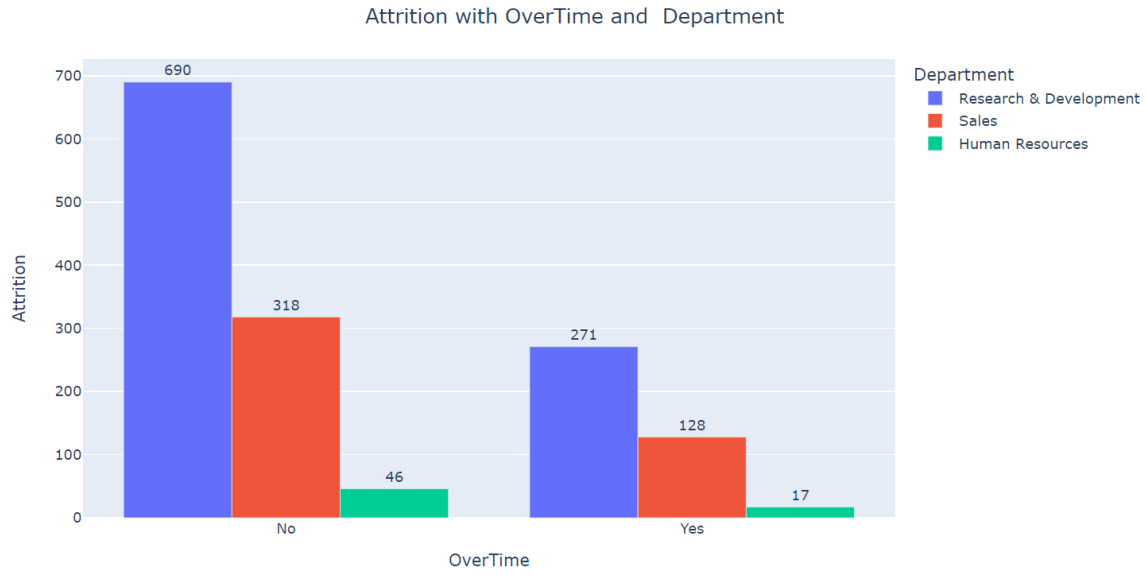

150
88
16
13
94
34
4
64
15
3
159
27
Life Sciences
Medical
Technical Degree
Other
Marketing
Human Resources
EducationField


Attrition with Gender and Department
582
257
43
379
189
20
Male
Female
Gender
Attrition
Department
Research & Development
Sales
Human Resources

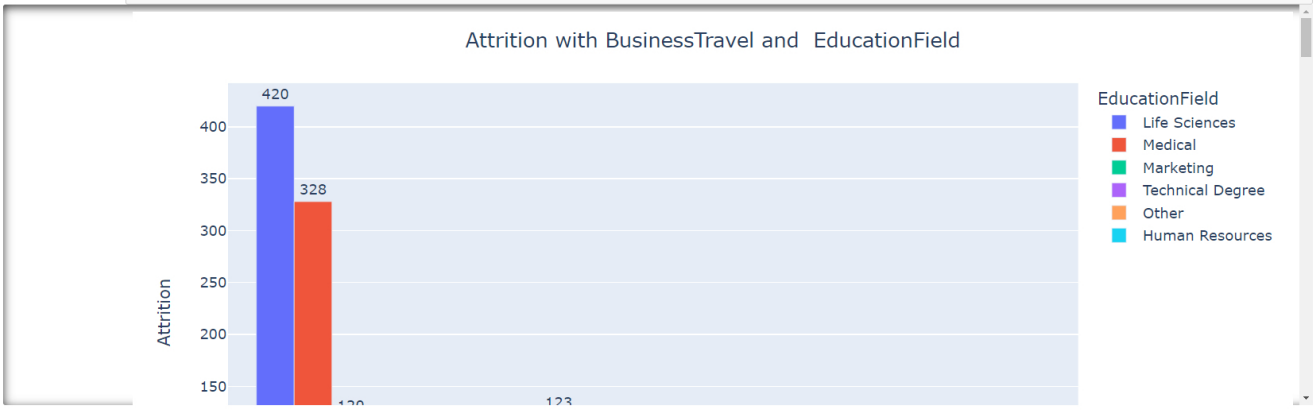

Attrition with JobRole and Department
326
83
37
54
11
292
259
145
131
80
52
Sales Executive
Sales Representative
Manager
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Research Director
Human Resources
JobRole
Attrition
Department
Sales
Research & Development
Human Resources


Attrition with MaritalStatus and Department
433
206
304
153
224
87
Attrition
Department
Research & Development
Sales
Human Resources

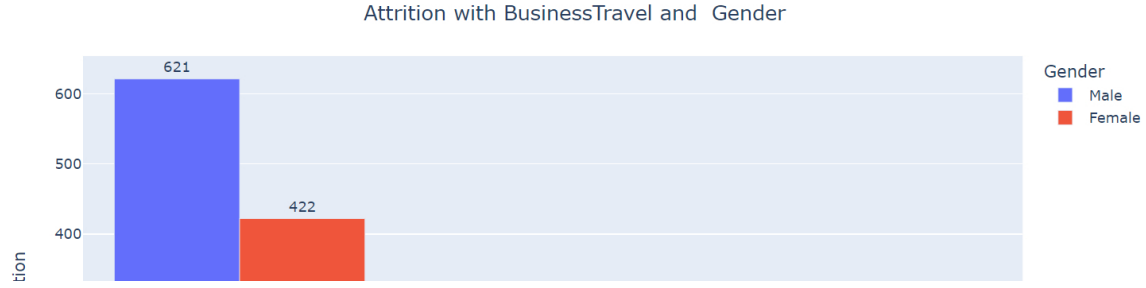


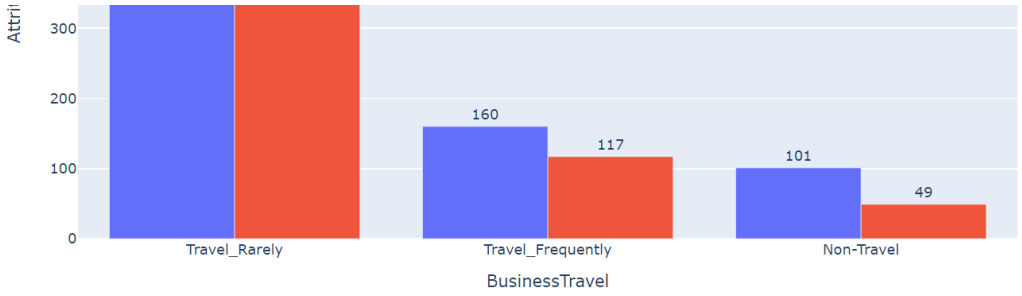


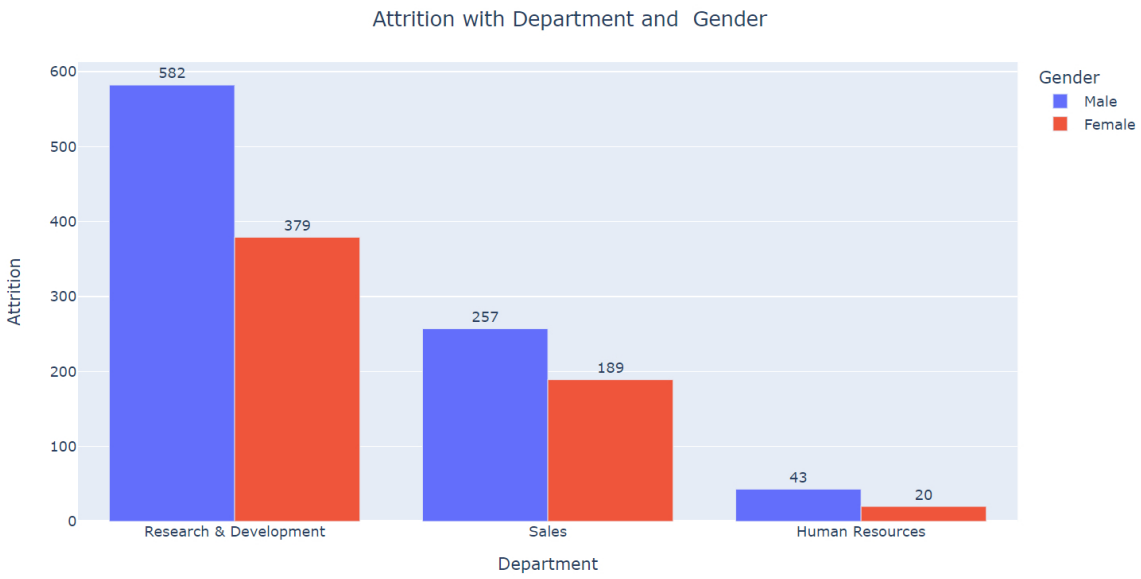
In [22]:
```
col=['BusinessTravel','Department','Gender','JobRole','MaritalStatus','OverTime']

for i in col:
    be=hr.groupby([i,'EducationField'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='EducationField',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  EducationField')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

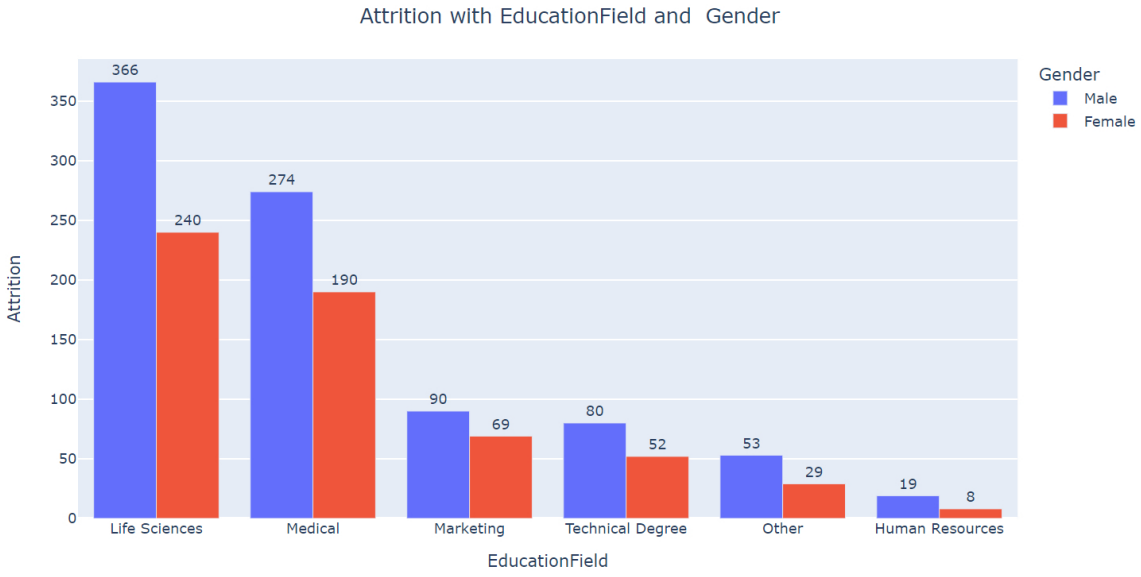


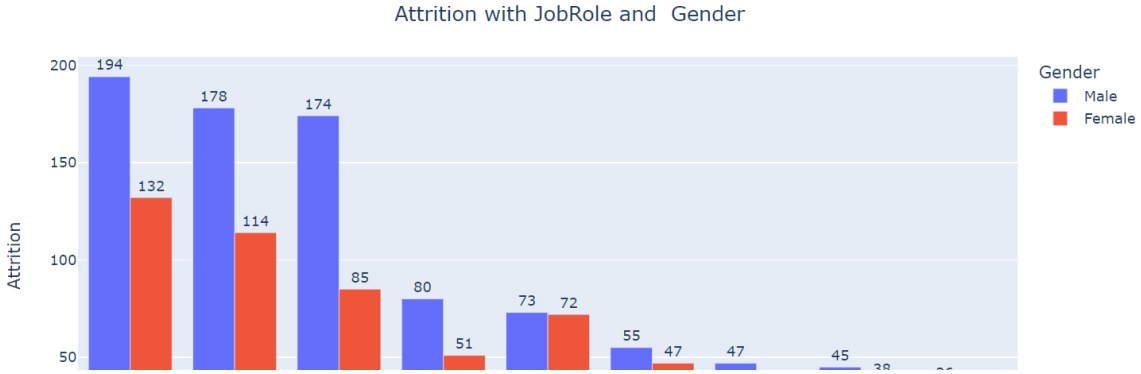
In [23]:
```
col=['BusinessTravel','Department','EducationField','JobRole','MaritalStatus','OverTime']

for i in col:
    be=hr.groupby([i,'Gender'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='Gender',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  Gender')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

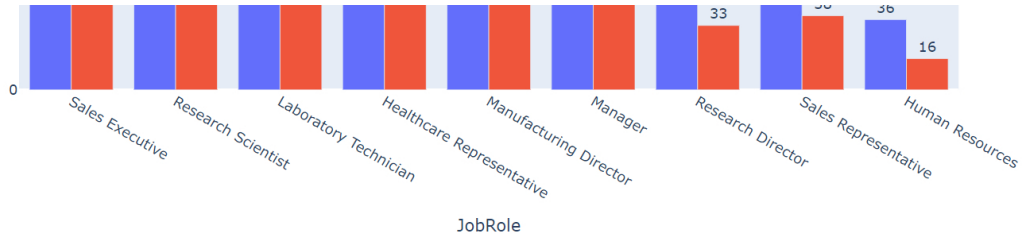


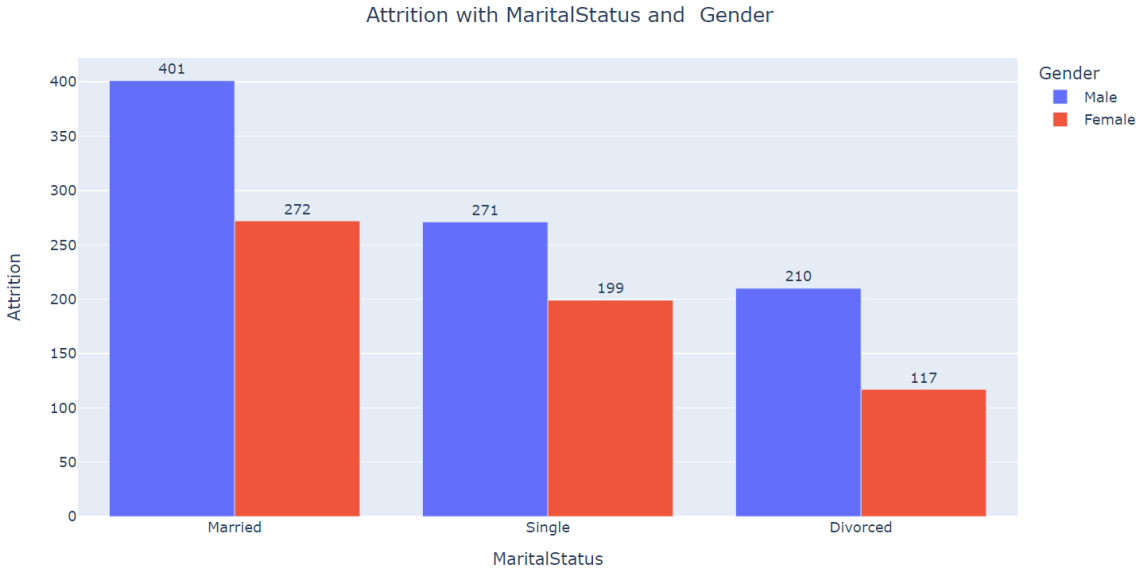

Attri
300
200
100
0
160
117
101
49
Travel_Rarely
Travel_Frequently
Non-Travel
BusinessTravel

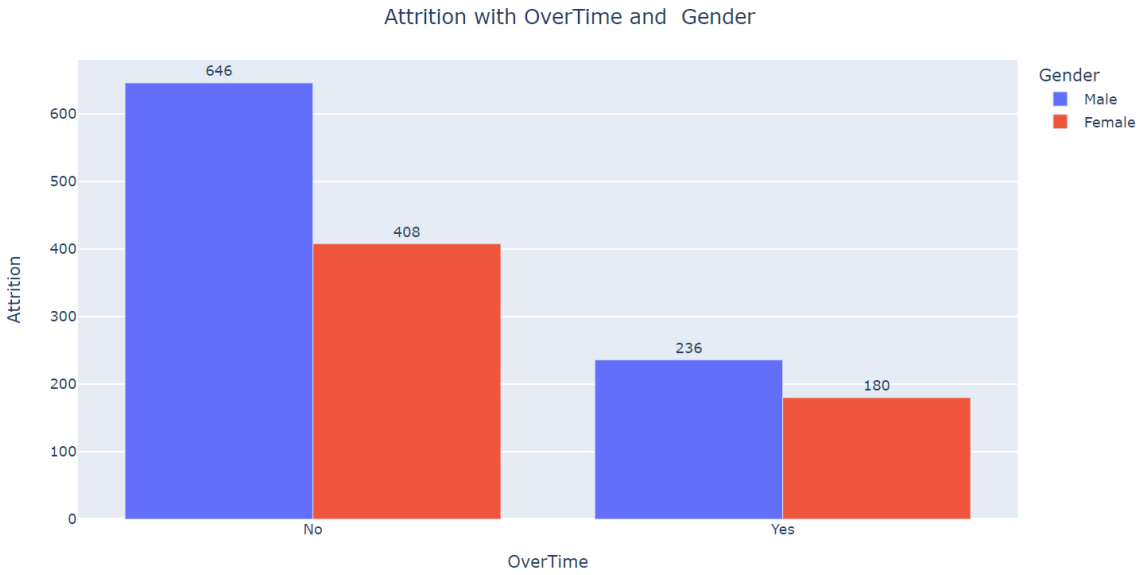

Attrition with Department and Gender
600
500
400
300
200
100
0
Attrition
582
379
257
189
43
20
Research & Development
Sales
Human Resources
Department
Gender
Male
Female


Attrition with EducationField and Gender
350
300
250
200
150
100
50
0
Attrition
366
240
274
190
90
69
80
52
53
29
19
8
Life Sciences
Medical
Marketing
Technical Degree
Other
Human Resources
EducationField
Gender
Male
Female

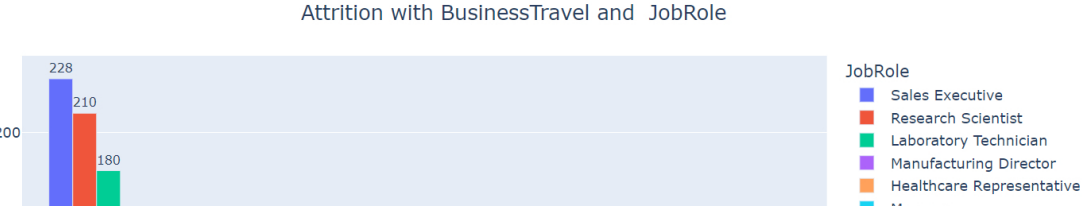

Attrition with JobRole and Gender
200
150
100
50
Attrition
194
132
178
114
174
85
80
51
73
72
55
47
47
45
Gender
Male
Female

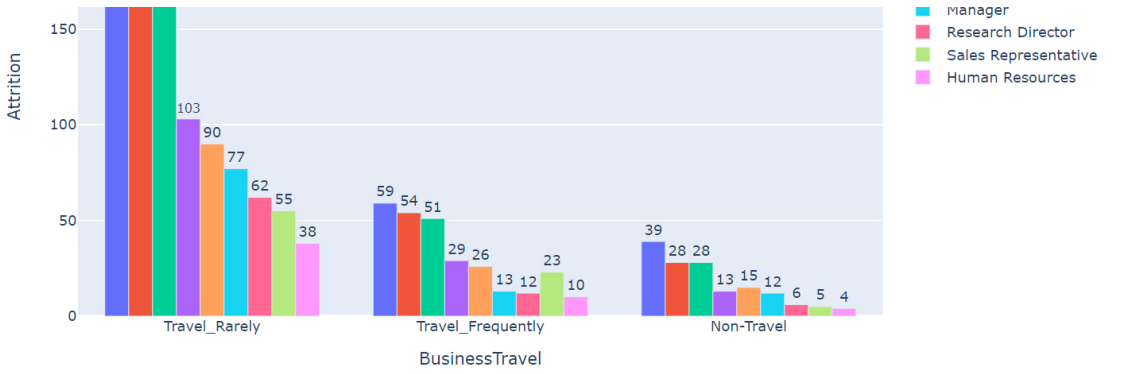




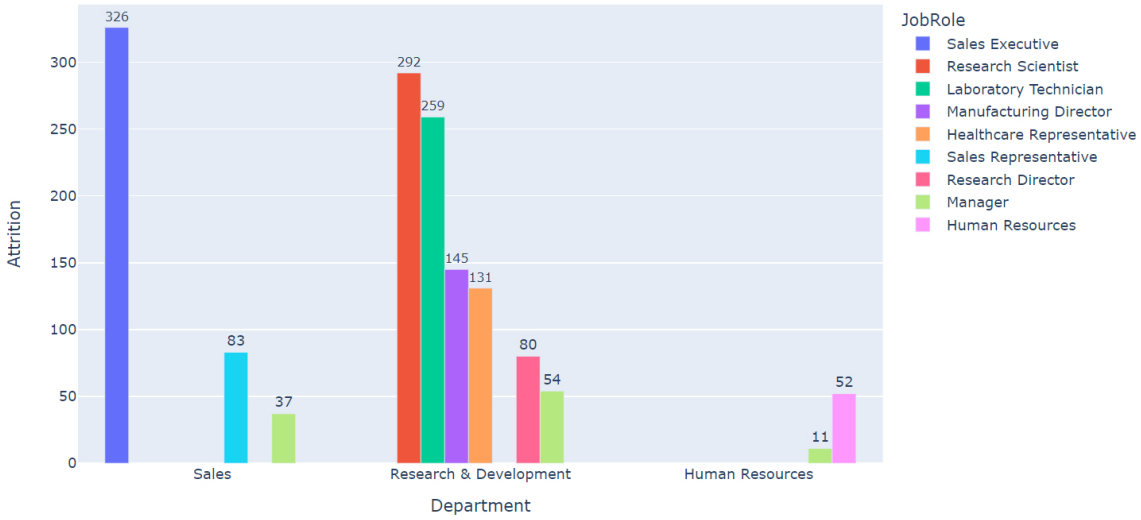


In [24]:
```
col=['BusinessTravel','Department','EducationField','Gender','MaritalStatus','OverTime']

for i in col:
    be=hr.groupby([i,'JobRole'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='JobRole',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  JobRole')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

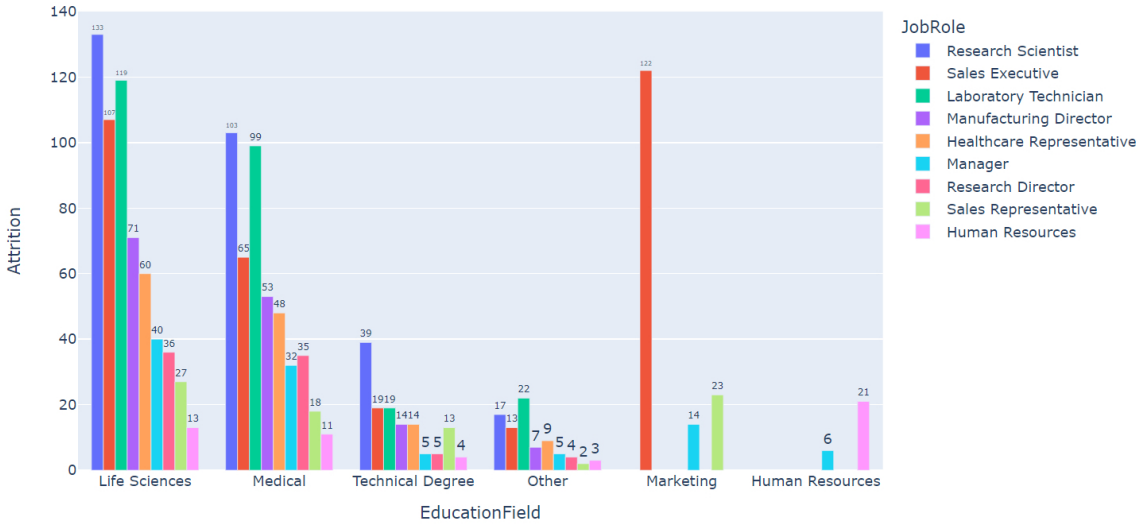




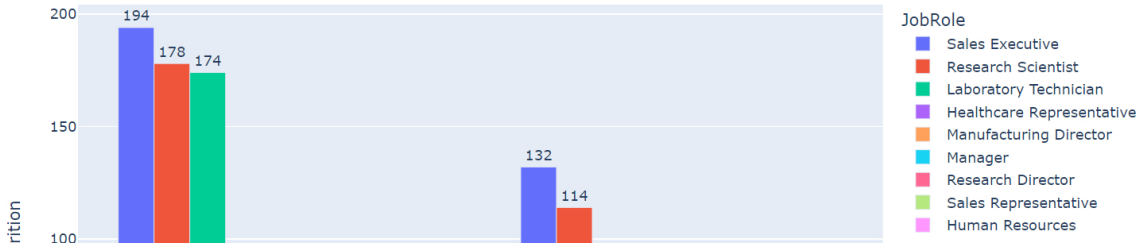
Attrition with Department and JobRole

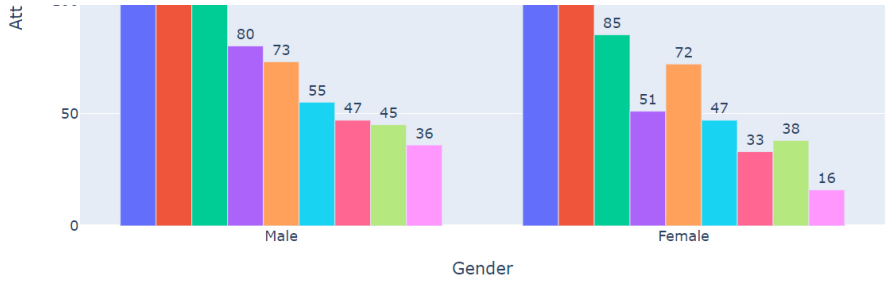


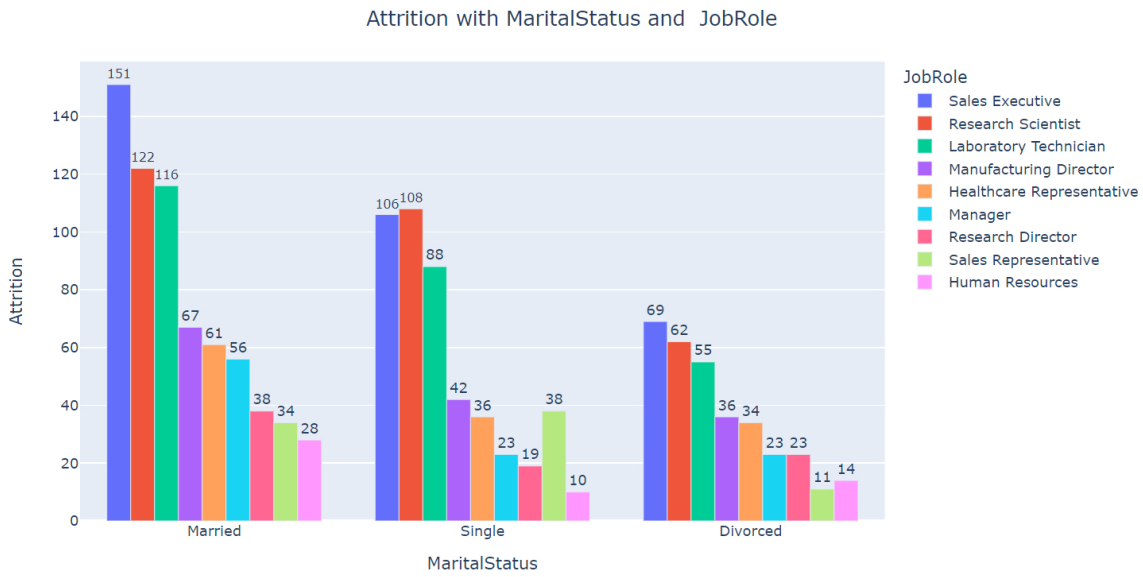
Attrition with EducationField and JobRole

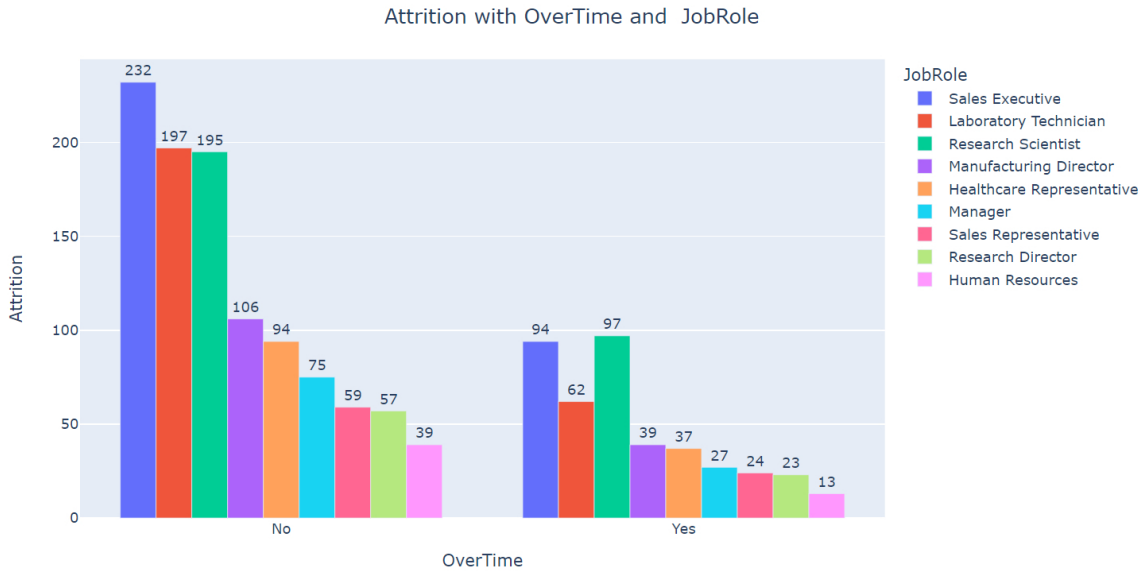


Attrition with Gender and JobRole

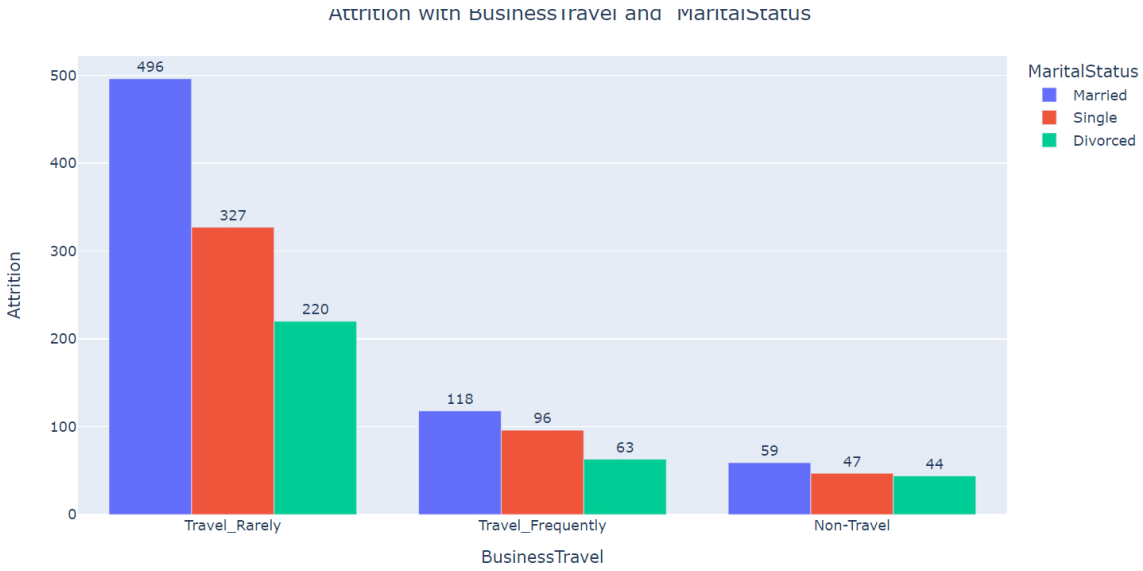


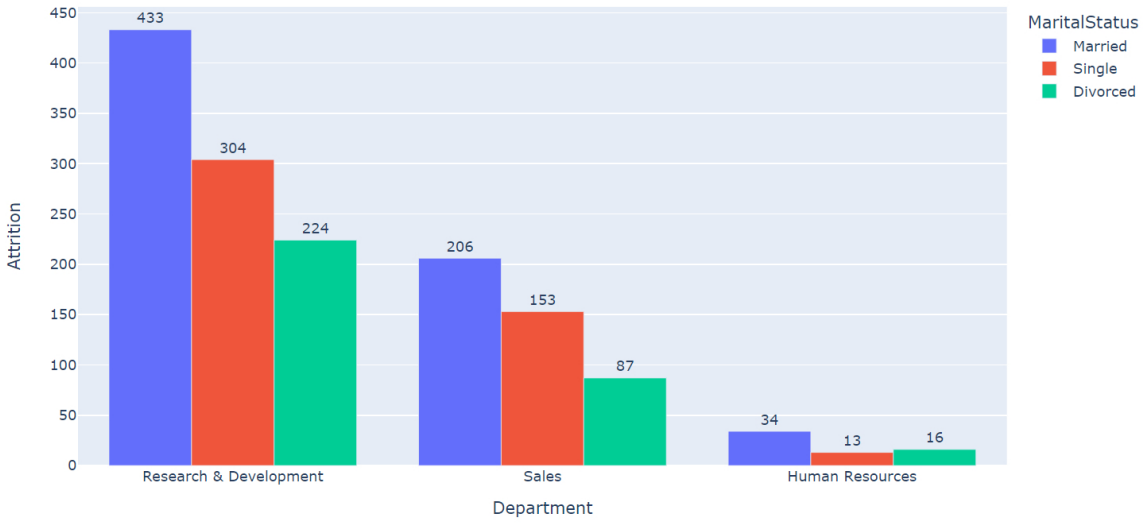


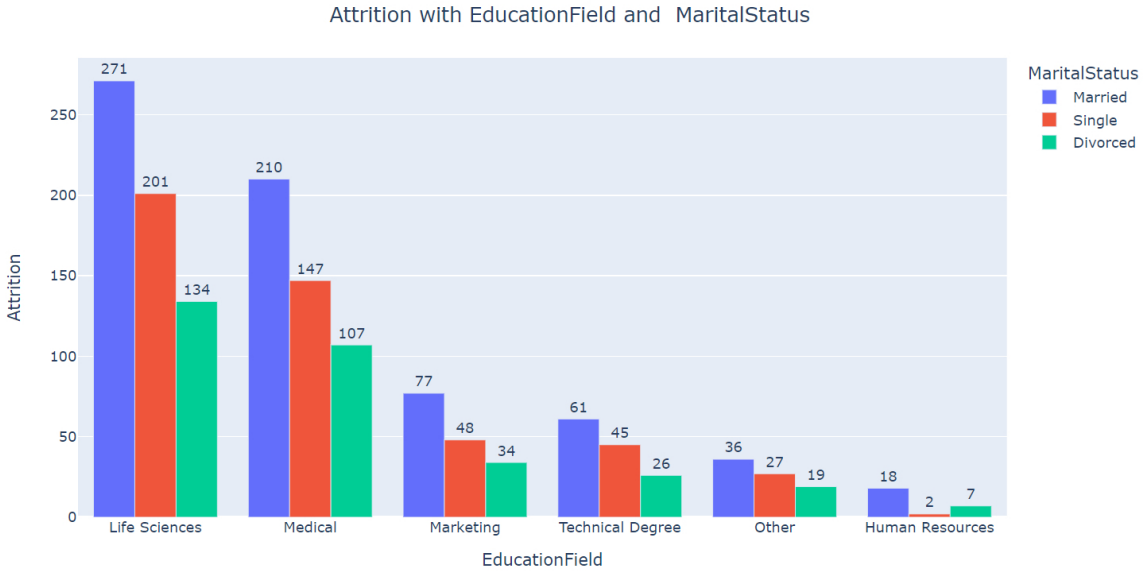




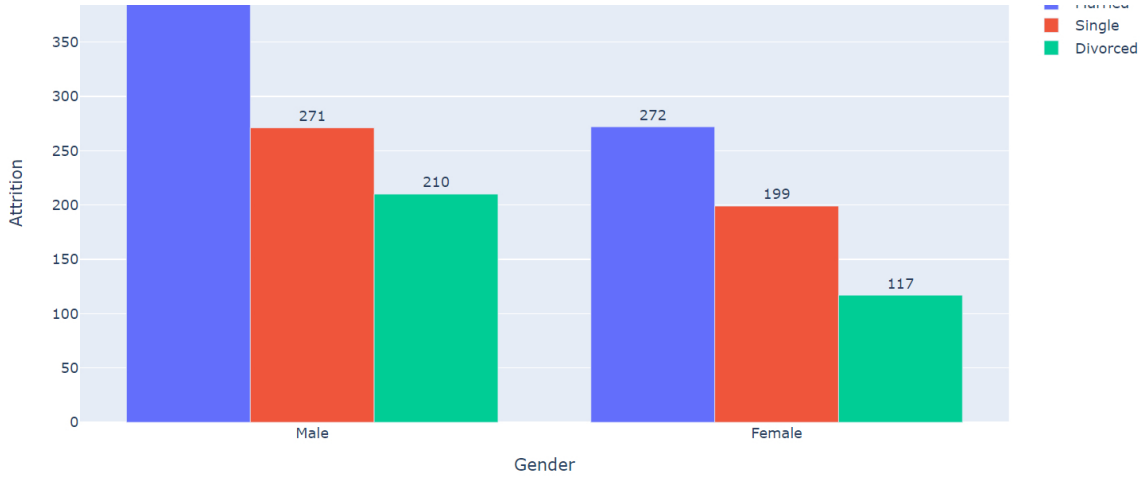
In [25]:
```
col=[
 'BusinessTravel',
 'Department',
 'EducationField',
 'Gender',
 'JobRole',

 'OverTime']

for i in col:
    be=hr.groupby([i,'MaritalStatus'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='MaritalStatus',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  MaritalStatus')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
    fig.show()
```

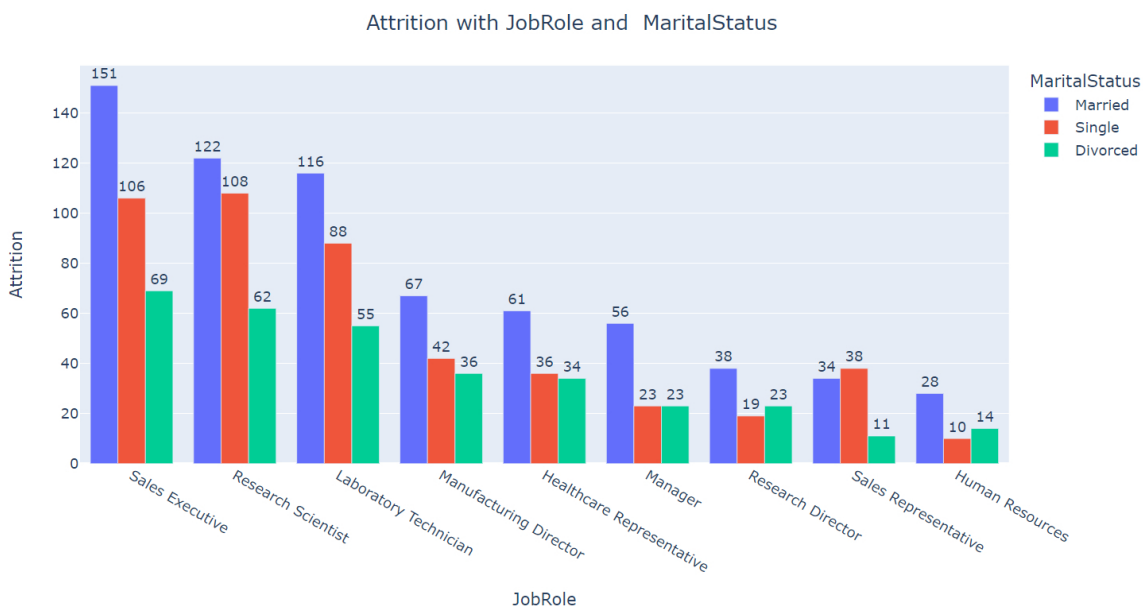

Attrition with BusinessTravel and MaritalStatus
Attrition
BusinessTravel
Travel_Rarely
Travel_Frequently
Non-Travel
496
327
220
118
96
63
59
47
44
MaritalStatus
Married
Single
Divorced

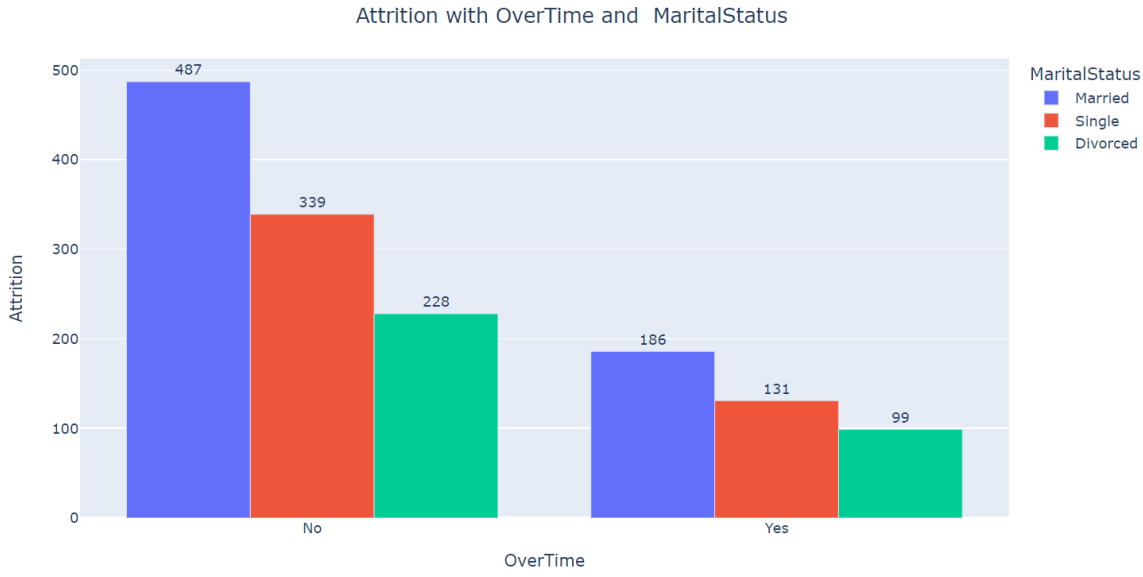

Attrition with Department and MaritalStatus
Attrition
Department
Research & Development
Sales
Human Resources
433
304
224
206
153
87
34
13
16
MaritalStatus
Married
Single
Divorced


Attrition with EducationField and MaritalStatus
Attrition
EducationField
Life Sciences
Medical
Marketing
Technical Degree
Other
Human Resources
271
201
134
210
147
107
77
48
34
61
45
26
36
27
19
18
2
7
MaritalStatus
Married
Single
Divorced


Attrition with Gender and MaritalStatus
401
MaritalStatus
Married

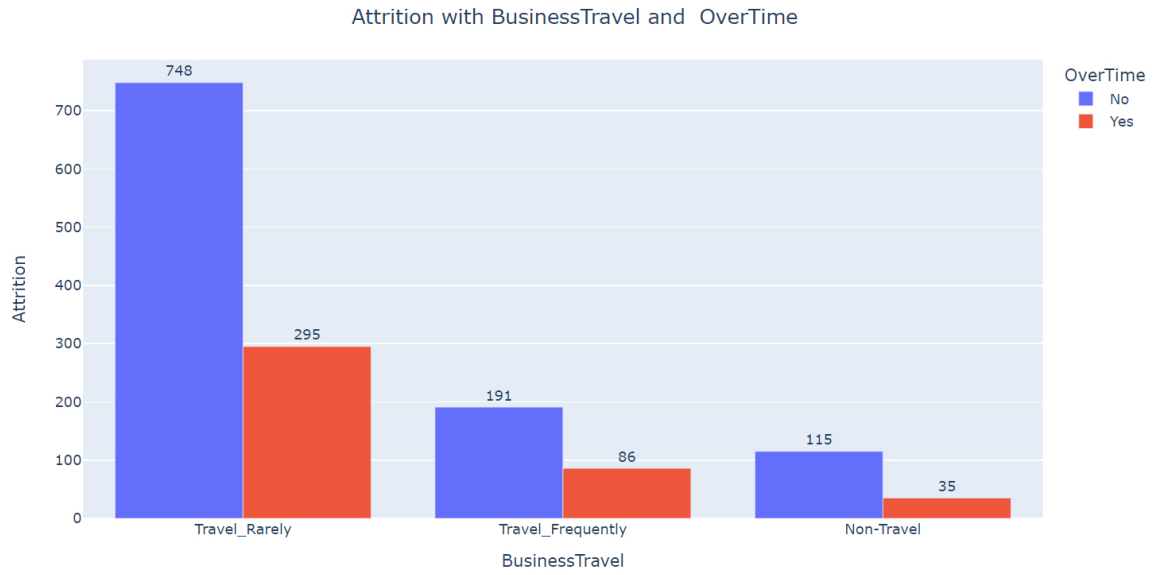


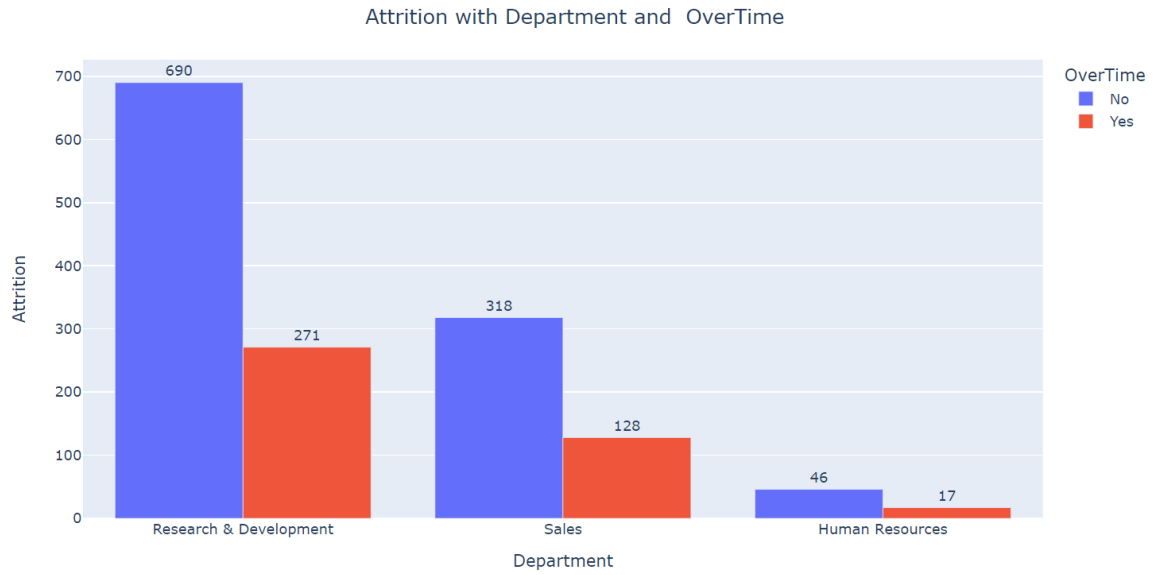


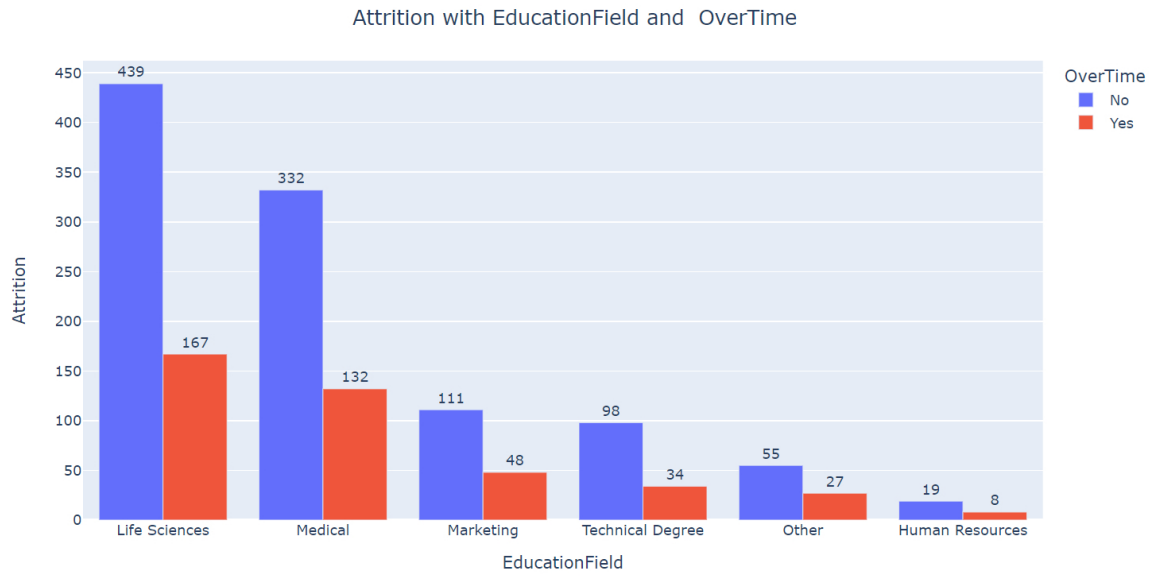


In [26]:
```
col=['BusinessTravel','Department','EducationField','Gender','JobRole','MaritalStatus']

for i in col:
    be=hr.groupby([i,'OverTime'])["Attrition"].count().reset_index().sort_values('Attrition',ascending=False)
    fig=px.bar(data_frame=be,x=i,y="Attrition",color='OverTime',barmode='group',text_auto=True)
    fig.update_layout(title="Attrition with " + i + ' and  OverTime')
    fig.update_layout(xaxis_title= i)
    fig.update_traces(textposition='outside')
    fig.update_layout(title_x=.5)
```

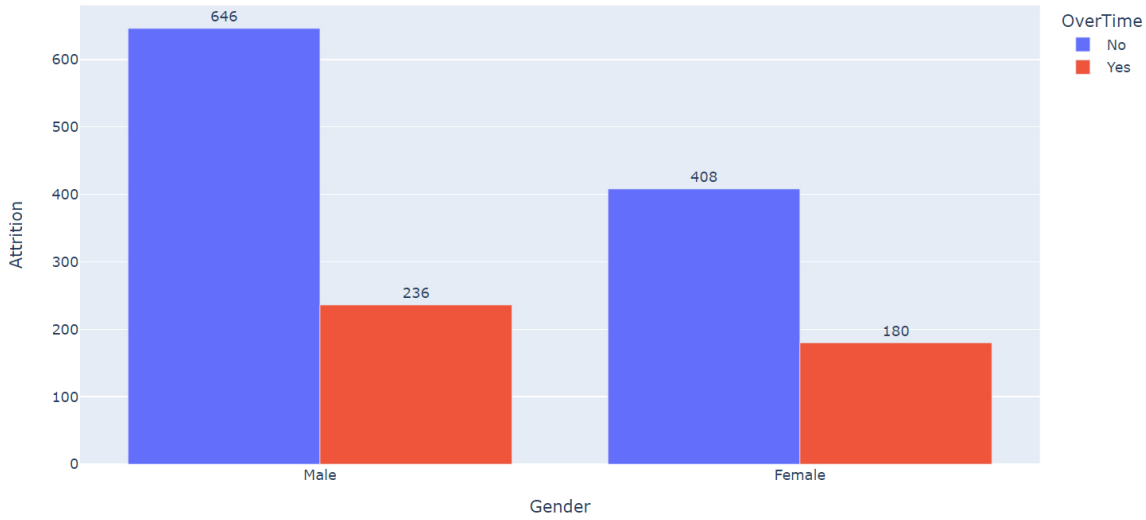
```
fig.show()
```



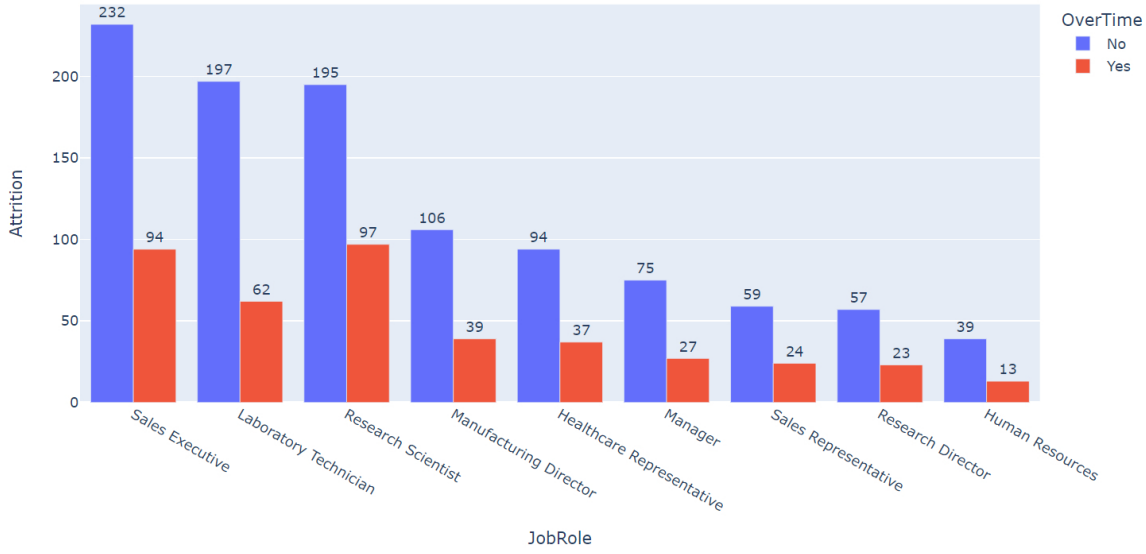




Attrition with Gender and OverTime



Attrition with JobRole and OverTime



Attrition with MaritalStatus and OverTime