HR Analytics Report

In [1]:
```
from IPython.display import Markdown
```

Table of Contents

"*Project's goals and objectives*"

- The primary goal of this project is to analyze employee attrition within the organization and uncover the underlying factors contributing to attrition rates. The investigation will cover several critical aspects, including employee satisfaction, career progression, work-life balance, and more. By doing this, I want to understand why employees leave the company and how I can recommend ways to make them stay.
- By accomplishing these tasks, my aim is to provide actionable insights and recommendations to help the organization better understand and manage employee attrition, ultimately enhancing workplace satisfaction and retention.

"Import the Libraries"

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

"Import the dataset"

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

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Education
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sci
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sci
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	(
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sci
4	27	No	Travel_Rarely	591	Research & Development	2	1	Me

5 rows × 35 columns

"General overview of the data"

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [5]:
```
data.shape
```
Out[5]:
```
(1470, 35)
```

Data Cleaning

"Counting Missing Values in Each Column of the Dataset"

In [6]:
```
print(data.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
```

"Detecting and Handling Duplicates"

In [7]:
```
if data.duplicated().sum() > 0:
    print("Duplicates are present")
else:
    print("No Duplicates Exists")
```

```
No Duplicates Exists
```

"Retrieving Column Names from the DataFrame"

In [8]:
```
data.columns
```

Out[8]:
```
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')
```

"Column Name Renaming for Enhanced Clarity"

In [9]:
```
new_data= data.rename(columns={'Education':'EducationLevel',
                               'NumCompaniesWorked':'NumofCompaniesWorked'})
```

"Exploring Unique Values for Data Analysis"

In [10]:
```
new_data.nunique()
```

Out[10]:
```
Age                           43
Attrition                      2
BusinessTravel                 3
DailyRate                    886
Department                     3
DistanceFromHome              29
EducationLevel                 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
NumofCompaniesWorked          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
```

"Eliminating Irrelevant Columns: Enhancing Data Clarity and Focus"

In [11]:
```
new_data.drop(["EmployeeCount","Over18","StandardHours"], axis= 1, inplace=True)
new_data.head()
```

Out[11]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	EducationLevel	Educ
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Li
1	49	No	Travel_Frequently	279	Research & Development	8	1	Li
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Li
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 32 columns

EDA

"Exploring Data Statistics: A Summary Overview"

In [12]:
```
data.describe()
```

Out[12]:

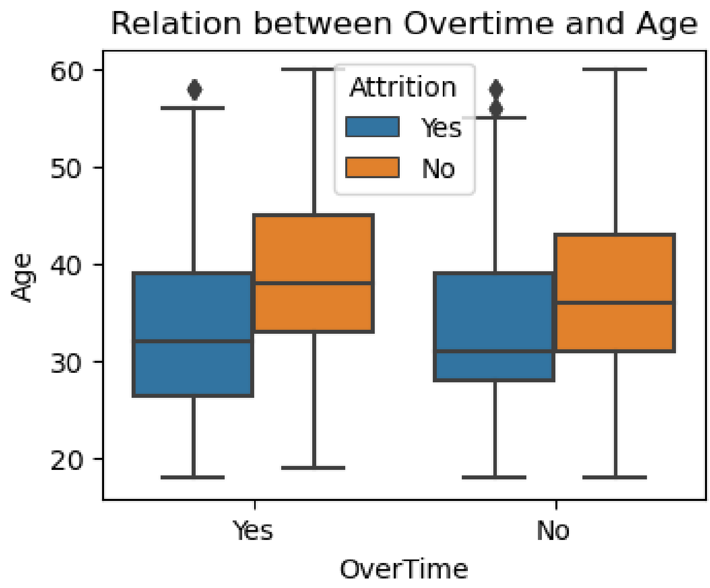
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000

8 rows × 26 columns

"*Analyzing Employee Attrition: Age vs. Overtime*"

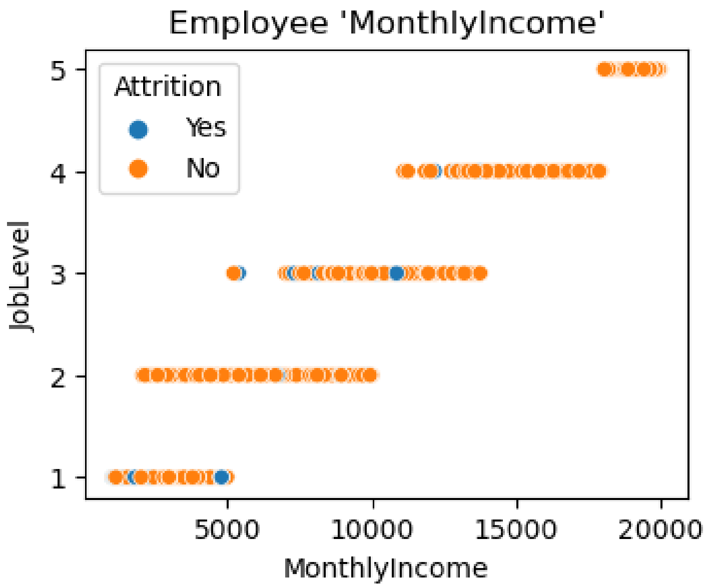
In [13]:
```
result= new_data[(new_data['Attrition'] == 'Yes')].groupby("OverTime")['Attrition'].cc
print(result)
print("------------------------------------------")

plt.figure(figsize=(4,3))
sns.boxplot(x='OverTime', y="Age", hue="Attrition", data=new_data)
plt.title("Relation between Overtime and Age")
plt.show()
```

```
  OverTime  Attrition
0       No        110
1      Yes        127
------------------------------------------
```

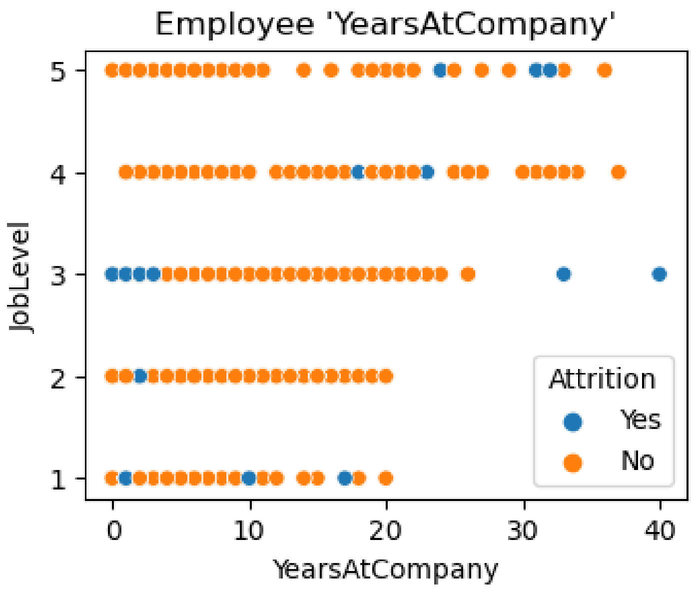


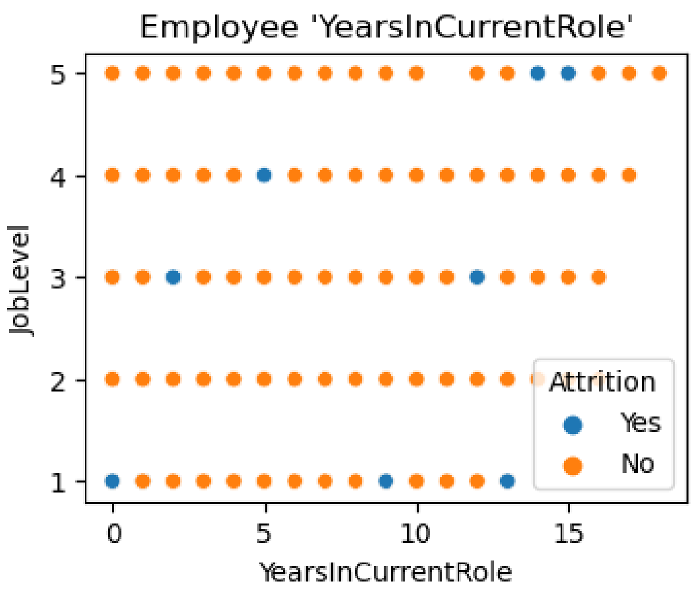
"Analyzing Employee Career Progression and Attrition Trends"

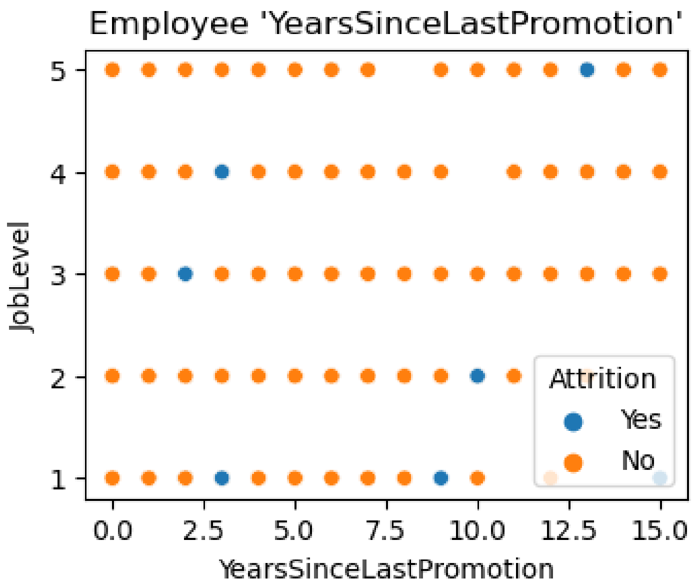
In [14]:
```
columns= ['MonthlyIncome','TotalWorkingYears','YearsAtCompany','YearsInCurrentRole',
          'YearsSinceLastPromotion']

for i in columns:
    plt.figure(figsize=(4,3))
    sns.scatterplot(x=i, y='JobLevel', hue="Attrition", data=new_data)
    plt.title(f"Employee '{i}'")
    plt.show()
```




Employee 'TotalWorkingYears'
JobLevel
TotalWorkingYears
Attrition
Yes
No


Employee 'YearsAtCompany'
JobLevel
YearsAtCompany
Attrition
Yes
No


Employee 'YearsInCurrentRole'
JobLevel
YearsInCurrentRole
Attrition
Yes
No



"Displaying Value Counts for Each Columns in a Dataset":

In [15]:
```
columns =['OverTime','Attrition','MaritalStatus','JobRole','Gender','EducationField','
'BusinessTravel','TotalWorkingYears','EducationLevel','NumofCompaniesWorked','Distance

for i in columns:
    value_counts = new_data[i].value_counts().sort_index()
    print(f'# Value Counts for "{i}":\n{value_counts}\n')
    print("---------------------------------------")
```

```
# Value Counts for "OverTime":
No     1054
Yes     416
Name: OverTime, dtype: int64

---------------------------------------
# Value Counts for "Attrition":
No     1233
Yes     237
Name: Attrition, dtype: int64

---------------------------------------
# Value Counts for "MaritalStatus":
Divorced    327
Married     673
Single      470
Name: MaritalStatus, dtype: int64

---------------------------------------
# Value Counts for "JobRole":
Healthcare Representative    131
Human Resources               52
Laboratory Technician        259
Manager                      102
Manufacturing Director       145
Research Director             80
Research Scientist           292
Sales Executive              326
Sales Representative          83
Name: JobRole, dtype: int64

---------------------------------------
# Value Counts for "Gender":
Female    588
Male      882
Name: Gender, dtype: int64

---------------------------------------
# Value Counts for "EducationField":
Human Resources      27
Life Sciences       606
Marketing           159
Medical             464
Other                82
Technical Degree    132
Name: EducationField, dtype: int64

---------------------------------------
# Value Counts for "Department":
Human Resources            63
Research & Development    961
Sales                     446
Name: Department, dtype: int64

---------------------------------------
# Value Counts for "BusinessTravel":
Non-Travel            150
Travel_Frequently     277
Travel_Rarely        1043
Name: BusinessTravel, dtype: int64

---------------------------------------
# Value Counts for "TotalWorkingYears":
0      11
1      81
2      31
```

```
3      42
4      63
5      88
6     125
7      81
8     103
9      96
10    202
11     36
12     48
13     36
14     31
15     40
16     37
17     33
18     27
19     22
20     30
21     34
22     21
23     22
24     18
25     14
26     14
27      7
28     14
29     10
30      7
31      9
32      9
33      7
34      5
35      3
36      6
37      4
38      1
40      2
Name: TotalWorkingYears, dtype: int64

---------------------------------------
# Value Counts for "EducationLevel":
1    170
2    282
3    572
4    398
5     48
Name: EducationLevel, dtype: int64

---------------------------------------
# Value Counts for "NumofCompaniesWorked":
0    197
1    521
2    146
3    159
4    139
5     63
6     70
7     74
8     49
9     52
Name: NumofCompaniesWorked, dtype: int64

---------------------------------------
# Value Counts for "DistanceFromHome":
1     208
2     211
```

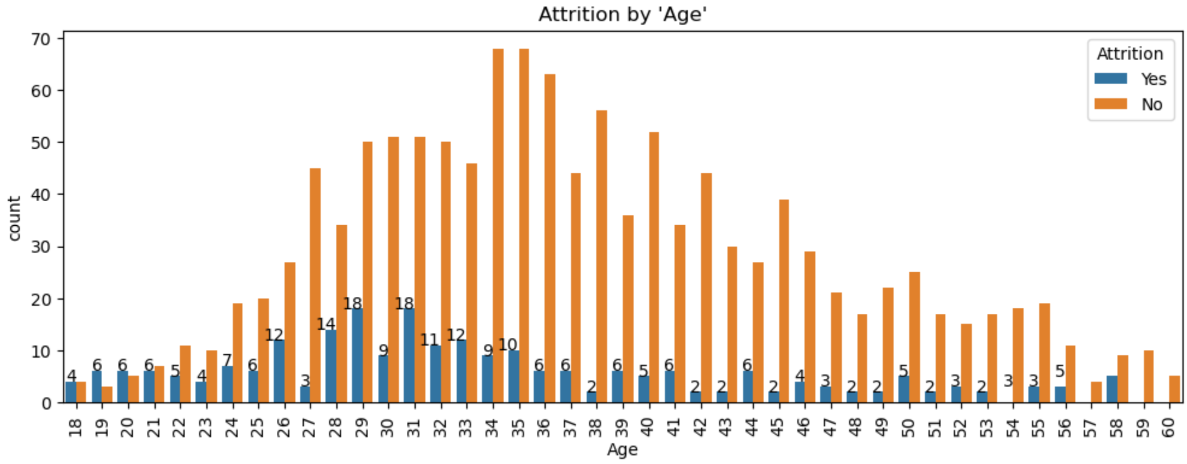
```
3       84
4       64
5       65
6       59
7       84
8       80
9       85
10      86
11      29
12      20
13      19
14      21
15      26
16      32
17      20
18      26
19      22
20      25
21      18
22      19
23      27
24      28
25      25
26      25
27      12
28      23
29      27
Name: DistanceFromHome, dtype: int64

---------------------------------------
```

Analyzing Employee Attrition with Data Visualization

- Employee's attrition count using **groupby()**

"*Analyzing Attrition by Age: A Visual Breakdown*"

In [16]:
```
plt.figure(figsize=(12,4))
sns.countplot(data=new_data, x="Age", hue= "Attrition")
plt.title("Attrition by 'Age'")
counts=new_data[new_data['Attrition'] == 'Yes'].groupby("Age")['Attrition'].count().re
plt.xticks(rotation= 90)
ax = plt.gca()
for index, row in counts.iterrows():
    ax.text(row.name, row['Attrition'], str(row['Attrition']), color='black', ha="righ
plt.show()
```



"Analyzing Attrition by Age groups"

- Divide age ranges into three categories and calculate the attrition rate for each category :
  - <30: Age less than 30 (young_employees)
  - 30-49: Age ranges between 30 and 49 (middle_aged_employees)
  - 50+: Age 50 and above (older_employees)

In [17]:

```
young_employees = data[data['Age'] < 30]
middle_aged_employees = data[(data['Age'] >= 30) & (data['Age'] <= 49)]
older_employees = data[data['Age'] >= 50]

attrition_count_young = (young_employees['Attrition'] == 'Yes').sum()
attrition_count_middle_aged = (middle_aged_employees['Attrition'] == 'Yes').sum()
attrition_count_older = (older_employees['Attrition'] == 'Yes').sum()

total_count_young = len(young_employees)
total_count_middle_aged = len(middle_aged_employees)
total_count_older = len(older_employees)

# Calculate the attrition rate for each category
attrition_rate_young = (attrition_count_young / total_count_young) * 100
attrition_rate_middle_aged = (attrition_count_middle_aged / total_count_middle_aged) *
attrition_rate_older = (attrition_count_older / total_count_older) * 100

print("Attrition Rate for Young Employees <30: {:.1f}%".format(attrition_rate_young))
print("Attrition Rate for middle_aged Employees 30-49: {:.1f}%".format(attrition_rate_
print("Attrition Rate for Older Employees 50+: {:.1f}%".format(attrition_rate_older))
```

```
Attrition Rate for Young Employees <30: 27.9%
Attrition Rate for middle_aged Employees 30-49: 12.7%
Attrition Rate for Older Employees 50+: 13.3%
```

"Analyzing Attrition by Department: Visualizing Percentage Breakdown"

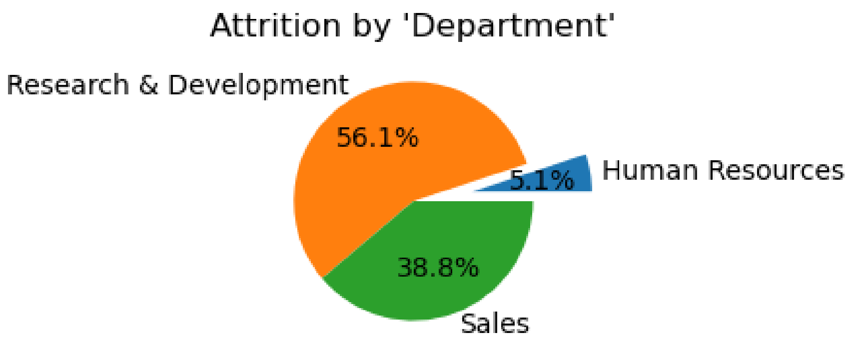
In [18]:

```
result = new_data[new_data['Attrition']=='Yes'].groupby('Department')['Attrition'].siz
print(result)
print("-----------------------------------------")

plt.figure(figsize=(2,2))
plt.pie(result['Attrition'], labels=result['Department'], autopct='%1.1f%%', explode=[
plt.title("Attrition by 'Department'")
plt.show
```

```
                Department  Attrition
0          Human Resources         12
1   Research & Development        133
2                    Sales         92
-----------------------------------------
```

Out[18]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

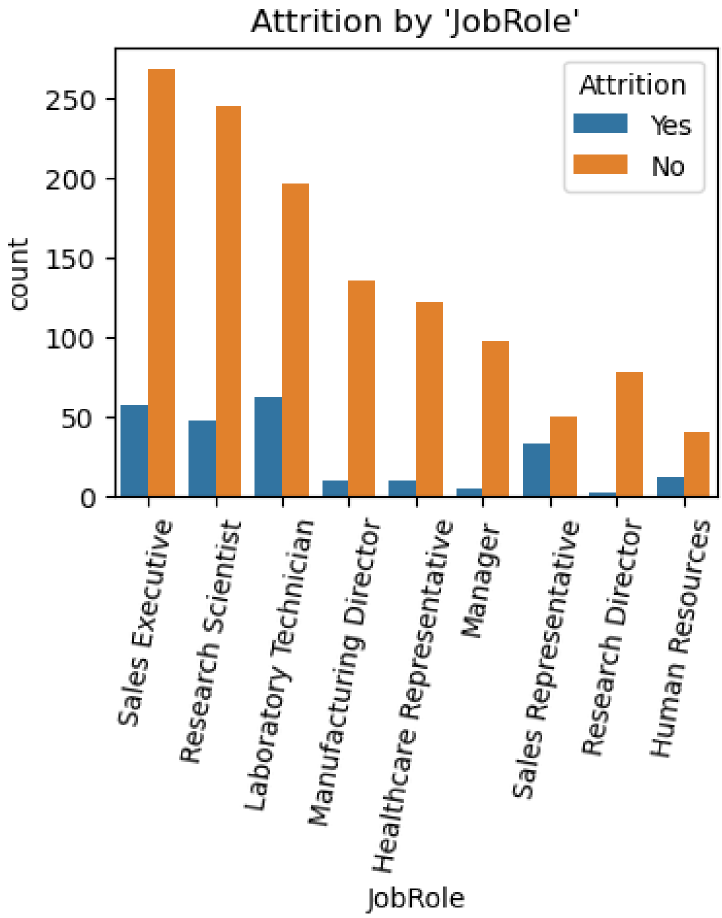


"*Analyzing Attrition and Monthly_Income by Job Role: A Visual Breakdown*"

In [19]:

```
result= new_data[new_data['Attrition'] == 'Yes'].groupby(['Department', 'JobRole']).ag
       Average_Monthly_Income=('MonthlyIncome', 'mean'),
       Attrition_Count=('Attrition', 'size')).reset_index()
print(result)
print("-----------------------------------------")

plt.figure(figsize=(4,3))
sns.countplot(data= new_data, x='JobRole', hue= "Attrition")
plt.title("Attrition by 'JobRole'")
plt.xticks(rotation= 80)
plt.show()
```

```
                Department                    JobRole  Average_Monthly_Income  \
0          Human Resources            Human Resources             3715.750000
1   Research & Development  Healthcare Representative             8548.222222
2   Research & Development      Laboratory Technician             2919.258065
3   Research & Development                    Manager            15106.000000
4   Research & Development     Manufacturing Director             7365.500000
5   Research & Development          Research Director            19395.500000
6   Research & Development         Research Scientist             2780.468085
7                    Sales                    Manager            19334.500000
8                    Sales            Sales Executive             7489.000000
9                    Sales       Sales Representative             2364.727273

   Attrition_Count
0               12
1                9
2               62
3                3
4               10
5                2
6               47
7                2
8               57
9               33
-----------------------------------------
```

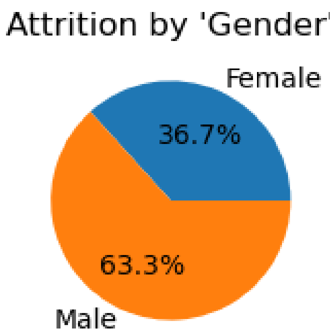


"Analyzing Gender-based Attrition: Visualizing Percentage Breakdown"

In [20]:
```
result = new_data[(new_data['Attrition'] == 'Yes')].groupby('Gender').agg(
     AttritionCount=('Attrition', 'size'),
     AverageAge=('Age', 'median'),
     AverageDistanceFromHome=('DistanceFromHome', 'median')).reset_index()
print(result)
print("------------------------------------------")

plt.figure(figsize=(2,2))
plt.pie(result['AttritionCount'], labels=result['Gender'], autopct='%1.1f%%')
plt.title("Attrition by 'Gender'")
plt.show
```

```
   Gender  AttritionCount  AverageAge  AverageDistanceFromHome
0  Female              87        31.0                      9.0
1    Male             150        32.0                      8.0
------------------------------------------
```

Out[20]: `<function matplotlib.pyplot.show(close=None, block=None)>`

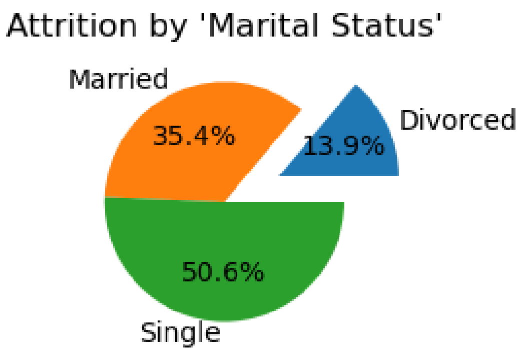


"Analyzing Attrition by Marital Status: Visualizing Percentage Breakdown"

In [21]:
```
result= new_data[(new_data['Attrition'] == 'Yes')].groupby('MaritalStatus')['Attritior
print(result)
print("------------------------------------------")

plt.figure(figsize=(2,2))
plt.pie(result['Attrition'], labels=result['MaritalStatus'],autopct='%1.1f%%',explode=
plt.title("Attrition by 'Marital Status'")
plt.show
```

```
  MaritalStatus  Attrition
0      Divorced         33
1       Married         84
2        Single        120
------------------------------------------
```

Out[21]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

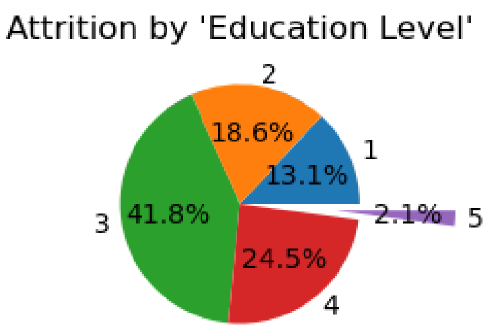


"Analyzing Attrition by their Education Level: Visualizing Percentage Breakdown"

In [22]:
```
result= new_data[(new_data['Attrition'] == 'Yes')].groupby('EducationLevel')['Attritic
print(result)
print("------------------------------------------")

plt.figure(figsize=(2,2))
plt.pie(result['Attrition'], labels=result['EducationLevel'],autopct='%1.1f%%',explode
plt.title("Attrition by 'Education Level'")
plt.show
```

```
   EducationLevel  Attrition
0               1         31
1               2         44
2               3         99
3               4         58
4               5          5
------------------------------------------
```

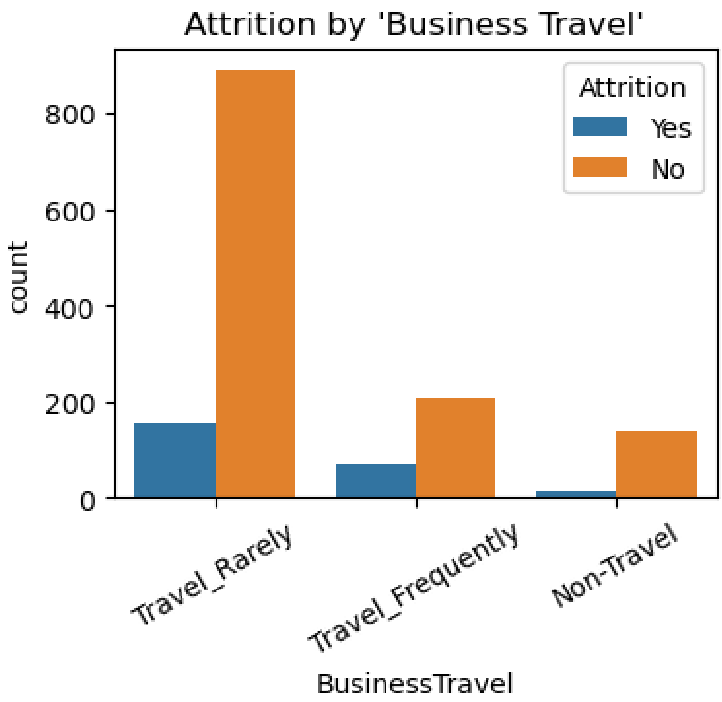
Out[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



"Analyzing Attrition by BusinessTravel: A Visual Breakdown"

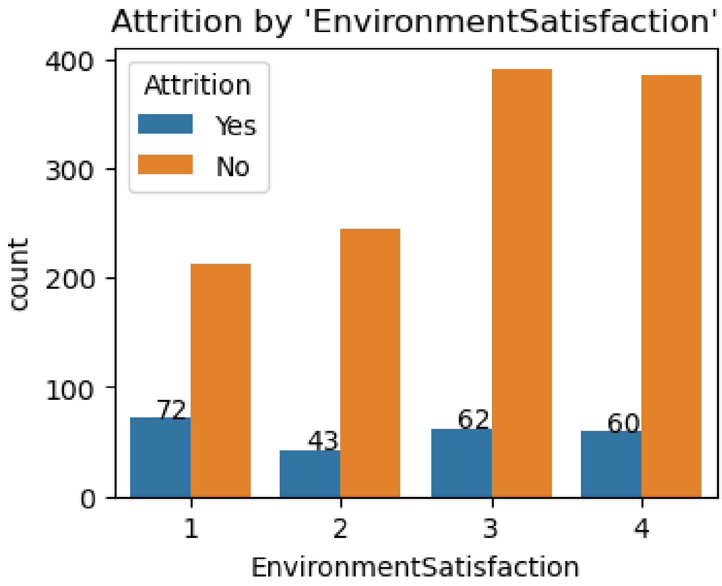
In [23]:
```
result= new_data[new_data['Attrition'] == 'Yes'].groupby("BusinessTravel")['Attrition'
print(result)
print("------------------------------------------")

plt.figure(figsize=(4,3))
sns.countplot(data=new_data, x="BusinessTravel", hue= "Attrition")
plt.title("Attrition by 'Business Travel'")
plt.xticks(rotation= 30)
plt.show()
```

```
      BusinessTravel  Attrition
0         Non-Travel         12
1  Travel_Frequently         69
2      Travel_Rarely        156
------------------------------------------
```

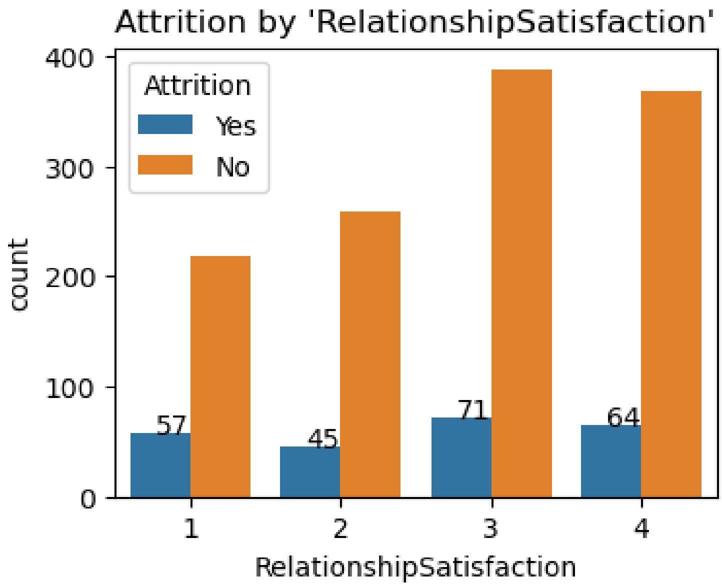


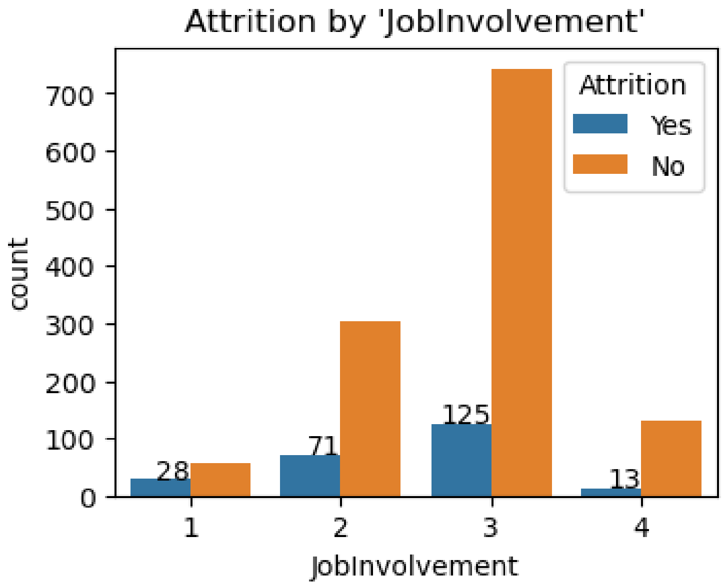
"Analyzing Employee's Satisfaction and Involvement: Attrition Insights"

In [24]:
```
columns= ['EnvironmentSatisfaction','JobSatisfaction','RelationshipSatisfaction','JobI
for i in columns:
    plt.figure(figsize=(4,3))
    sns.countplot(data=new_data, x=i, hue= "Attrition")
    plt.title(f"Attrition by '{i}'")
    counts = new_data[new_data['Attrition'] == 'Yes'].groupby(i)['Attrition'].count().
    ax = plt.gca()
    for index, row in counts.iterrows():
        ax.text(row.name, row['Attrition'], str(row['Attrition']), color='black', ha='
    plt.show()
```


Attrition by 'EnvironmentSatisfaction'
Attrition
Yes
No
count
0
100
200
300
400
72
43
62
60
1
2
3
4
EnvironmentSatisfaction

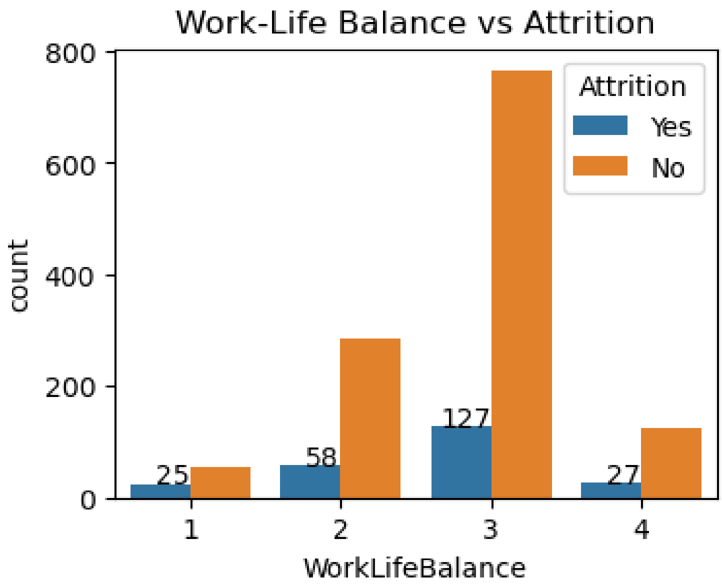

Attrition by 'JobSatisfaction'
Attrition
Yes
No
count
0
100
200
300
400
66
46
73
52
1
2
3
4
JobSatisfaction


Attrition by 'RelationshipSatisfaction'
Attrition
Yes
No
count
0
100
200
300
400
57
45
71
64
1
2
3
4
RelationshipSatisfaction



"Analyzing the Impact of Work-Life Balance on Employee Attrition"

In [25]:
```
plt.figure(figsize=(4,3))
sns.countplot(data=new_data, x='WorkLifeBalance', hue= "Attrition")
plt.title("Work-Life Balance vs Attrition")
counts = new_data[new_data['Attrition'] == 'Yes'].groupby('WorkLifeBalance')['Attritic
ax = plt.gca()
for index, row in counts.iterrows():
    ax.text(row.name, row['Attrition'], str(row['Attrition']), color='black', ha="righ
plt.show()
```

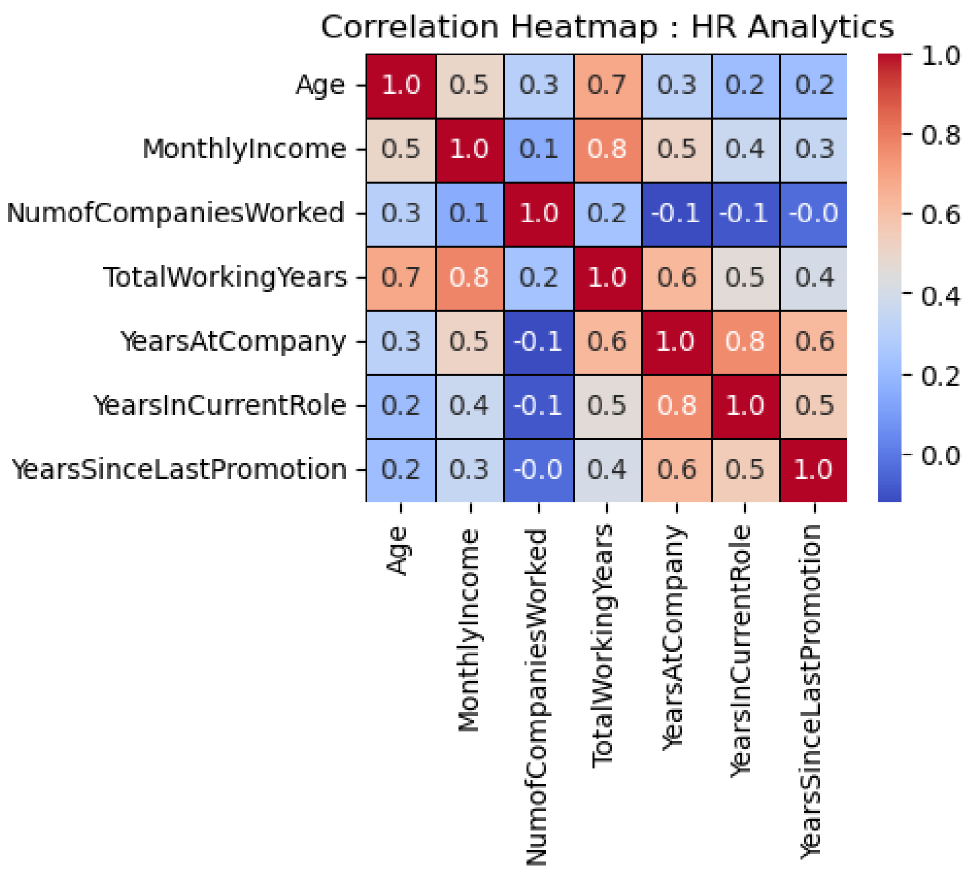


"Calculate and Visualize the Correlation between Numeric Variables"

- Correlation coefficients range from -1 to 1, where:
    - -1 indicates a perfect negative correlation.
    - 1 indicates a perfect positive correlation.
    - 0 indicates no linear correlation.
- Numeric values provide the exact correlation coefficient.

- Darker colors suggest stronger correlations. The color (blue or orange) indicates the direction of the correlation.

In [26]:
```
columns= ['Age','MonthlyIncome','NumofCompaniesWorked','TotalWorkingYears','YearsAtCom
         'YearsInCurrentRole','YearsSinceLastPromotion']
corr_matrix= new_data[columns].corr()

plt.figure(figsize=(4,3))
sns.heatmap(corr_matrix, annot=True, fmt=".1f", linewidths=0.5,cmap="coolwarm",lineco]
plt.title("Correlation Heatmap : HR Analytics")
plt.show()
```



"Summary"

In [27]:
```
print("Summary of Findings-")
print("Total number of Employees were given:",data["EmployeeCount"].count())
print("Total number of Attritions:",(data["Attrition"] == 'Yes').sum())
print("Percentage of Attrition: {:.1f}%".format(((data["Attrition"] == 'Yes').sum()/le
```

```
Summary of Findings-
Total number of Employees were given: 1470
Total number of Attritions: 237
Percentage of Attrition: 16.1%
```

Recommendations and Conclusion:

Based on the Analysis, we have identified several factors that are related to attrition within the organization. It's important for the company to address these factors to improve employee retention and job satisfaction. By implementing the recommended actions, the company can work towards reducing attrition and creating a more positive work environment. These actions include:

- Implement the identified retention strategies.
- Monitor attrition rates regularly to assess the impact of the strategies.
- Maintain open communication channels for employee feedback and concerns.
- Consider improving work-life balance to reduce attrition among employees.
- Monitor the impact of overtime work on attrition and take necessary actions to manage workload.
- Focus on career development opportunities, such as promotions and skill development training, to enhance job satisfaction.
- Conduct exit interviews with departing employees to gather more insights into attrition reasons.

Additionally, the ANALYSIS reveals specific INSIGHTS:

- Attrition rate is higher in employees within the age range of <30 and lower in older employees. Older employees often have stability in their careers and may be less likely to make major career changes.
- Younger employees, especially within the range of 18-30, are often in the phase of career exploration. Their experimentation can lead to higher turnover as they seek the right fit.
- The highest count of attrition is from the Research and Development department, with 56.1%. This is due to the larger size of the department's staff, the percentage of attrition turns out to be lesser than the others.
- Attrition rates are higher in the 'Laboratory Technician' and 'Sales Executive' job roles.
- Attrition rate is higher in male employees with average age (32) and Average Distance From Home (8 km).
- Attrition rate is higher in employees whose marital status is single.
- Attrition rate is higher in employees with an education level of 3.
- Employees who travel rarely for business purposes have a higher attrition count.
- Employees with Job Involvement level 3 have the highest attrition count.
- Employees with Job Satisfaction level 3 have the highest attrition count.
- Employees working overtime have a higher attrition count compare to others.

To address these insights, the company should focus on work-life balance initiatives, conduct surveys and gather feedback, and consider strategies to reduce attrition among employees working overtime.

--**ThankYou**--