# HR_ver2

June 3, 2025

## 1 HR Analytics Employee Attrition

Source: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset/data

This portfolio will focus on analyzing various factors in relation to employee attrition.

## 2 Data Import & Initial Exoplore

[25]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from statsmodels.stats.proportion import proportions_ztest as pro_z
from scipy.stats import ttest_ind, chi2_contingency
```

[2]:
```
#Read file
df = pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')

df.head(3)
```

[2]:

| | Age | Attrition | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationField | EmployeeCount | EmployeeNumber | … | RelationshipSatisfaction | StandardHours | StockOptionLevel |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 41 | Yes | Travel_Rarely | 1102 | Sales | 1 | 2 | Life Sciences | 1 | 1 | … | 1 | 80 | 0 |
| 1 | 49 | No | Travel_Frequently | 279 | Research & Development | 8 | 1 | Life Sciences | 1 | 2 | … | 4 | 80 | 1 |
| 2 | 37 | Yes | Travel_Rarely | 1373 | Research & Development | 2 | 2 | Other | 1 | 4 | … | 2 | 80 | 0 |

```
   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                       0                     5
1                  7                       1                     7
2                  0                       0                     0

[3 rows x 35 columns]
```

We noticed there were too many columns, so we took an additional look.

[3]:
```
df.columns
```

[3]:
```
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')
```

### 2.1 Remove Unimportant Columns

[4]:
```
df.drop(['EmployeeCount', 'Over18', 'StandardHours'], axis = 1, inplace = True)
```

### 2.2 Check Null

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 32 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   EmployeeNumber            1470 non-null   int64
 9   EnvironmentSatisfaction   1470 non-null   int64
 10  Gender                    1470 non-null   object
 11  HourlyRate                1470 non-null   int64
 12  JobInvolvement            1470 non-null   int64
 13  JobLevel                  1470 non-null   int64
 14  JobRole                   1470 non-null   object
 15  JobSatisfaction           1470 non-null   int64
 16  MaritalStatus             1470 non-null   object
 17  MonthlyIncome             1470 non-null   int64
 18  MonthlyRate               1470 non-null   int64
 19  NumCompaniesWorked        1470 non-null   int64
 20  OverTime                  1470 non-null   object
 21  PercentSalaryHike         1470 non-null   int64
 22  PerformanceRating         1470 non-null   int64
 23  RelationshipSatisfaction  1470 non-null   int64
 24  StockOptionLevel          1470 non-null   int64
 25  TotalWorkingYears         1470 non-null   int64
 26  TrainingTimesLastYear     1470 non-null   int64
 27  WorkLifeBalance           1470 non-null   int64
 28  YearsAtCompany            1470 non-null   int64
 29  YearsInCurrentRole        1470 non-null   int64
 30  YearsSinceLastPromotion   1470 non-null   int64
 31  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(24), object(8)
memory usage: 367.6+ KB
```

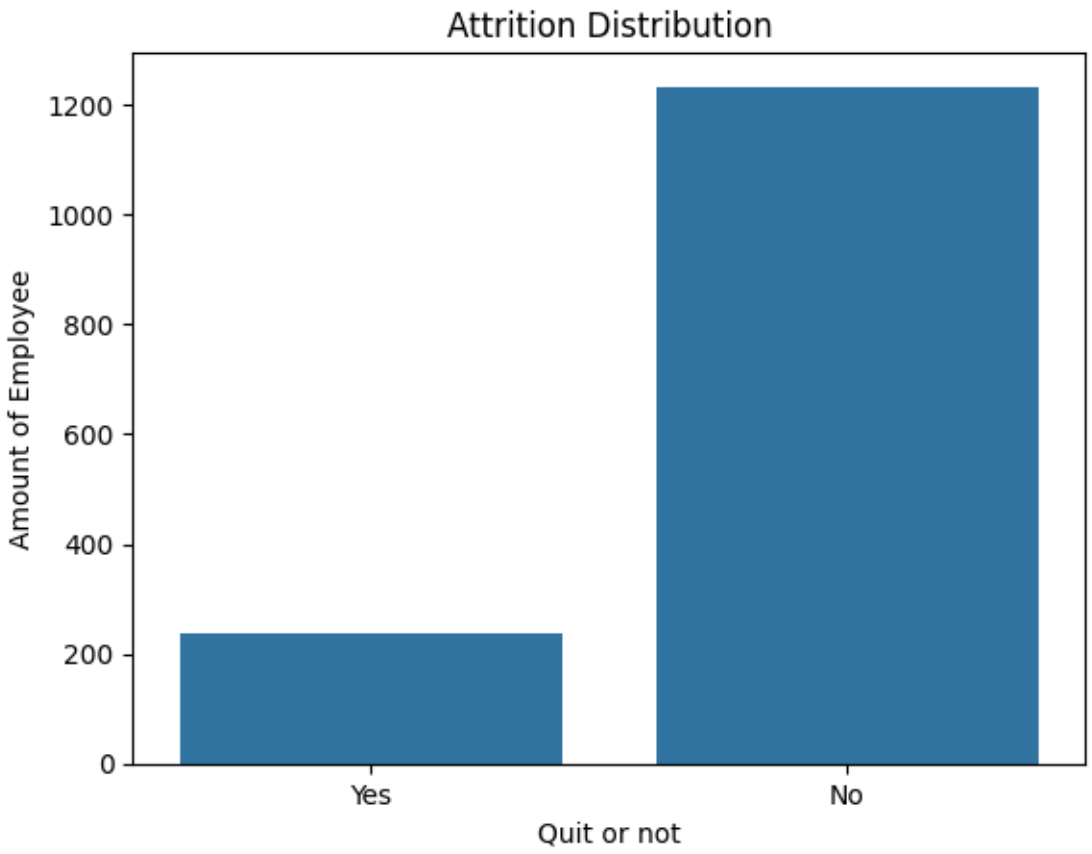
No missing values found!

# 3 EDA

## 3.1 Employee Attrition Distribution

[15]:
```
sns.countplot( x= 'Attrition', data = df)
plt.title('Attrition Distribution')
plt.xlabel('Quit or not')
plt.ylabel('Amount of Employee')

plt.show()
```




Checking whether the attrition rate is significantly higher than the average in the U.S.

https://datatrack.trendforce.com.tw/Chart/content/2381/united-states-resignment-rate-total

[24]:
```
count = df["Attrition"].value_counts()['Yes']

nobs = len(df)

value = 0.032  #   2025/5,    3.2%

z_stat, p_value = pro_z(count, nobs, value, alternative = 'larger')

print(f"Z Statistics: {z_stat:.3f}")
print(f"p-value: {p_value:.4f}")
```

```
Z Statistics: 13.473
p-value: 0.0000
```

Z  1.96,  p-value  0.05,
With a Z-statistic of 13.47 and a p-value $< 0.001$, we reject the null hypothesis at the 5% significance

level and conclude that the employee attrition rate in this company is significantly higher than the national average of 3.2%.

### 3.2 Determine Whether the Variables have Significant Correlation with Attrition

[40]:
```
#
categorical_cols = df.select_dtypes(include = 'object').columns.to_list()
numerical_cols = df.select_dtypes(exclude = 'object').columns.to_list()

#
categorical_cols = [col for col in categorical_cols if col != 'Attrition' and␣
 ↪df[col].nunique() > 1]  #
numerical_cols = [col for col in numerical_cols if col != 'EmployeeNumber']␣
 ↪#

#
results_cat = []
results_num = []

#
for col in categorical_cols:
    contingency_table = pd.crosstab(df[col], df['Attrition'])
    if contingency_table.shape[0] > 1:
        chi2, p, _, _ = chi2_contingency(contingency_table)
        results_cat.append((col, "Chi-Square", p))

#   T test
for col in numerical_cols:
    group_yes = df[ df['Attrition'] == 'Yes'][col]
    group_no = df[ df['Attrition'] == 'No'][col]

    t_stat, p, = ttest_ind(group_yes, group_no, equal_var = False)
    results_num.append((col, "T-test", p))

#
results_cat_df = pd.DataFrame(results_cat, columns = ['Variable','Test Type',␣
 ↪'P-value'])
results_cat_df['Significant(p < 0.05)'] = results_cat_df["P-value"] < 0.05
results_cat_df.sort_values("P-value", inplace = True)

results_num_df = pd.DataFrame(results_num, columns = ['Variable','Test Type',␣
 ↪'P-value'])
results_num_df['Significant(p < 0.05)'] = results_num_df["P-value"] < 0.05
results_num_df.sort_values("P-value", inplace = True)
```

```
results_num_df
```

[40]:

| | Variable | Test Type | P-value | Significant(p < 0.05) |
|---|---|---|---|---|
| 9 | MonthlyIncome | T-test | 4.433589e-13 | True |
| 7 | JobLevel | T-test | 9.844803e-13 | True |
| 16 | TotalWorkingYears | T-test | 1.159817e-11 | True |
| 20 | YearsInCurrentRole | T-test | 3.187390e-11 | True |
| 22 | YearsWithCurrManager | T-test | 1.185022e-10 | True |
| 0 | Age | T-test | 1.379760e-08 | True |
| 19 | YearsAtCompany | T-test | 2.285905e-07 | True |
| 15 | StockOptionLevel | T-test | 2.811541e-07 | True |
| 6 | JobInvolvement | T-test | 4.681195e-06 | True |
| 8 | JobSatisfaction | T-test | 1.052049e-04 | True |
| 4 | EnvironmentSatisfaction | T-test | 2.092053e-04 | True |
| 2 | DistanceFromHome | T-test | 4.136512e-03 | True |
| 17 | TrainingTimesLastYear | T-test | 2.036379e-02 | True |
| 1 | DailyRate | T-test | 3.003954e-02 | True |
| 18 | WorkLifeBalance | T-test | 3.046567e-02 | True |
| 14 | RelationshipSatisfaction | T-test | 8.972776e-02 | False |
| 11 | NumCompaniesWorked | T-test | 1.163340e-01 | False |
| 21 | YearsSinceLastPromotion | T-test | 1.986513e-01 | False |
| 3 | Education | T-test | 2.241713e-01 | False |
| 10 | MonthlyRate | T-test | 5.653438e-01 | False |
| 12 | PercentSalaryHike | T-test | 6.144301e-01 | False |
| 5 | HourlyRate | T-test | 7.913501e-01 | False |
| 13 | PerformanceRating | T-test | 9.124808e-01 | False |

T

According to the results of the Chi-square test and T-test shown in the table above, we can determine whether each factor is significantly correlated with attrition under a univariate condition. A corresponding plot is provided below for your reference.

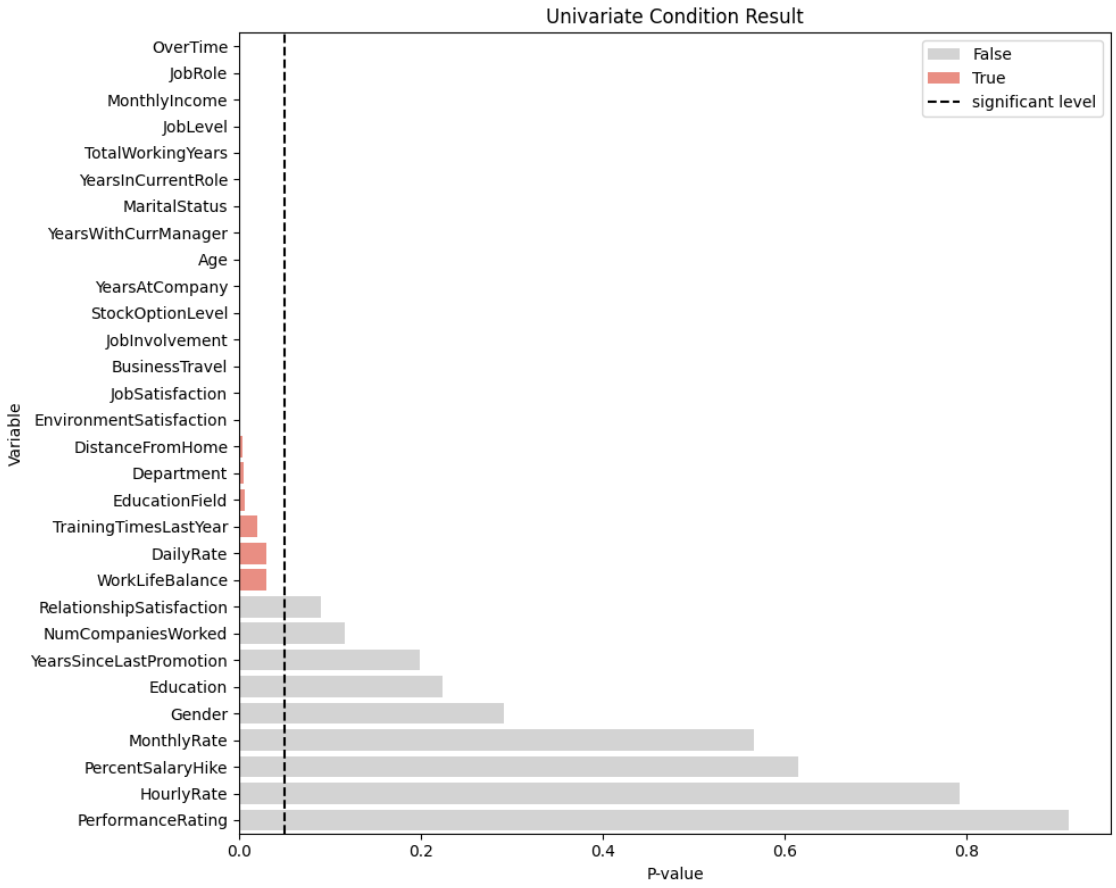
[43]:
```
summary_df = pd.concat([results_cat_df, results_num_df])

plt.figure(figsize = (10, 8))
sns.barplot(
    y = 'Variable',
    x = 'P-value',
    data = summary_df.sort_values("P-value"),
    hue = "Significant(p < 0.05)",
    dodge = False,
    palette = {True: "Salmon", False: "lightgray"}
)

plt.axvline(0.05, color = 'black', linestyle = "--", label = "significant␣
 ↪level")
plt.title("Univariate Condition Result")
```

```
plt.xlabel("P-value")
plt.ylabel('Variable')
plt.legend()
plt.tight_layout()

plt.show()
```




## 4 JobLevel with Overtime Analysis

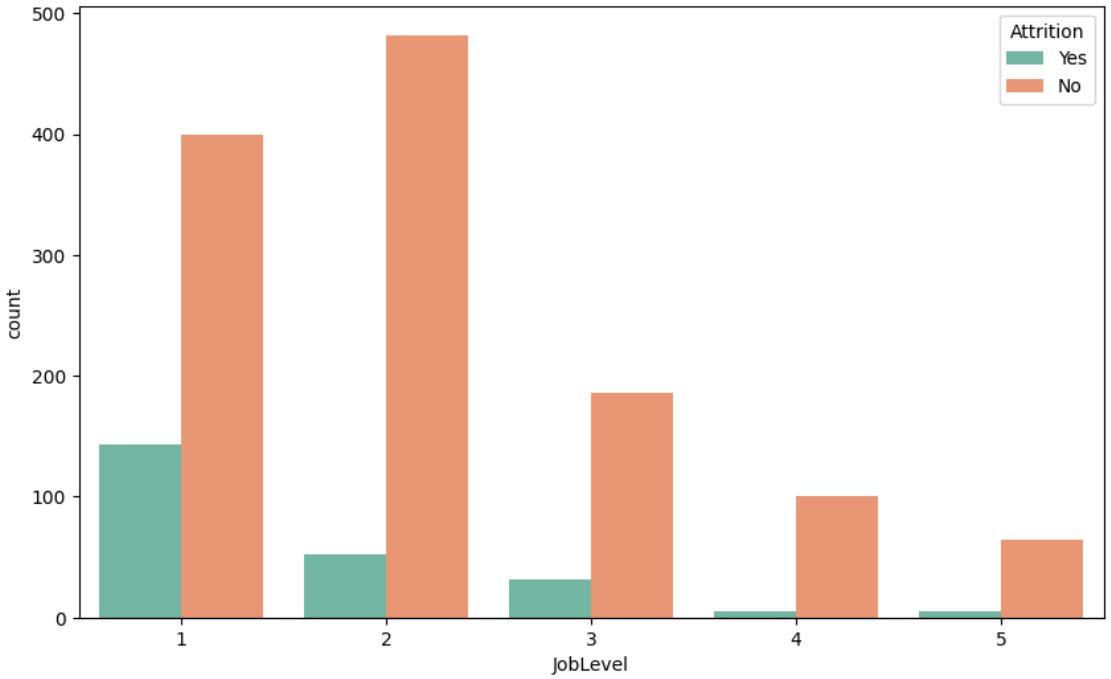
Among the factors that show a significant correlation with attrition, I believe the relationship between job level and overtime work may reveal something interesting. Therefore, I will conduct further analysis focusing on these two variables.

[48]:
```
plt.figure(figsize = (10, 6))
sns.countplot(
    data = df,
    x = 'JobLevel',
```

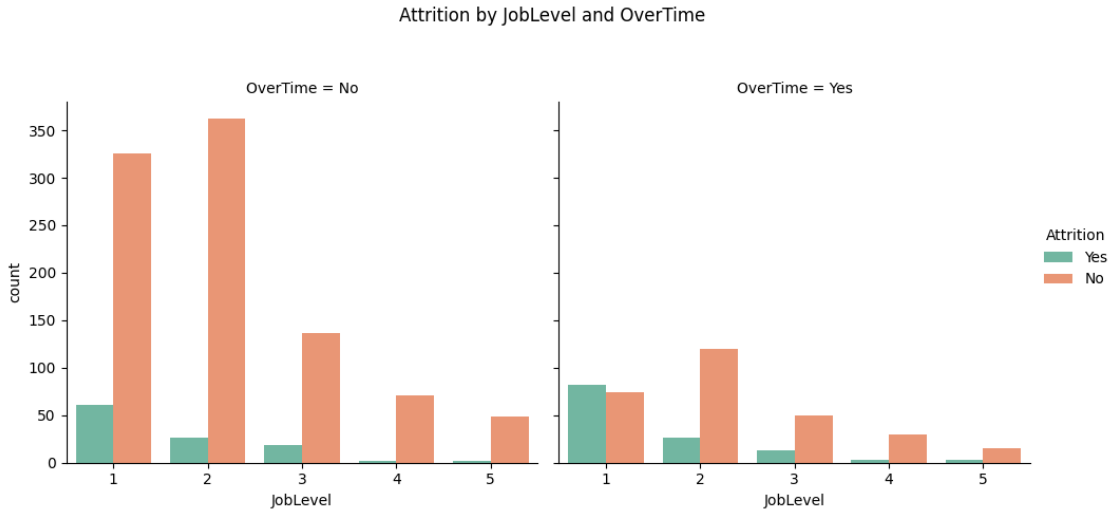
```
    hue = 'Attrition',
    palette = 'Set2',
    hue_order = ["Yes", "No"],
    order = sorted(df["JobLevel"].unique()),
    dodge = True
)

plt.figure(figsize = (12, 6))
sns.catplot(
    data = df,
    x = 'JobLevel',
    hue = 'Attrition',
    col = 'OverTime',
    kind = 'count',
    palette = 'Set2',
    col_order = ["No", "Yes"],
    order = sorted(df["JobLevel"].unique())
)

plt.subplots_adjust(top = 0.8)
plt.suptitle("Attrition by JobLevel and OverTime")
plt.show()
```




```
<Figure size 1200x600 with 0 Axes>
```




According to the charts above, I observed that lower-level employees tend to have a higher attrition rate. In cases involving overtime work, the number of employees who left even exceeds those who stayed. This may lead to increased training costs for entry-level staff and could potentially trigger a broken window effect. Therefore, I consider improving retention among junior employees as a key issue to address.