ZENVY AI POWERED PAYROLL

HR INTELLIGENCE DASHBOARD SUMMARY REPORT

First downloaded the dataset from GitHub - https://www.kaggle.com/datasets/karanchunara/hr-analytical-data?select=HR_Analytics.xlsx

Then opened python notebook and carried on these particular operation

ZENVY HR DATA CLEANING,PREPROCESSING AND EDA

IMPORTING LIBRARIES

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

```
df = pd.read_excel(r"C:\Users\Lenovo\Downloads\HR_Analytics.xlsx")
print(df)
```

```
       EmpID  Age AgeGroup Attrition     BusinessTravel  DailyRate  \
0      RM297   18    18-25       Yes      Travel_Rarely        230
1      RM302   18    18-25        No      Travel_Rarely        812
2      RM458   18    18-25       Yes  Travel_Frequently       1306
3      RM728   18    18-25        No         Non-Travel        287
4      RM829   18    18-25       Yes         Non-Travel        247
...      ...  ...      ...       ...                ...        ...
1475   RM412   60      55+        No      Travel_Rarely        422
1476   RM428   60      55+        No  Travel_Frequently       1499
1477   RM537   60      55+        No      Travel_Rarely       1179
1478   RM880   60      55+        No      Travel_Rarely        696
1479  RM1210   60      55+        No      Travel_Rarely        370

                  Department  DistanceFromHome  Education EducationField  ...
\
0     Research & Development                 3          3  Life Sciences  ...
1                      Sales                10          3        Medical  ...
2                      Sales                 5          3      Marketing  ...
3     Research & Development                 5          2  Life Sciences  ...
4     Research & Development                 8          1        Medical  ...
...                      ...               ...        ...            ...  ...
1475  Research & Development                 7          3  Life Sciences  ...
1476                   Sales                28          3      Marketing  ...
1477                   Sales                16          4      Marketing  ...
1478                   Sales                 7          4      Marketing  ...
1479  Research & Development                 1          4        Medical  ...
```

```
      RelationshipSatisfaction  StandardHours  StockOptionLevel  \
0                            3             80                 0
1                            1             80                 0
2                            4             80                 0
3                            4             80                 0
4                            4             80                 0
...                        ...            ...               ...
1475                         4             80                 0
1476                         4             80                 0
1477                         4             80                 0
1478                         2             80                 1
1479                         3             80                 1

     TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
0                    0                      2                3
1                    0                      2                3
2                    0                      3                3
3                    0                      2                3
4                    0                      0                3
...                ...                    ...              ...
1475                33                      5                1
1476                22                      5                4
1477                10                      1                3
1478                12                      3                3
1479                19                      2                4

      YearsAtCompany YearsInCurrentRole  YearsSinceLastPromotion  \
0                  0                  0                        0
1                  0                  0                        0
2                  0                  0                        0
3                  0                  0                        0
4                  0                  0                        0
...              ...                ...                      ...
1475              29                  8                       11
1476              18                 13                       13
1477               2                  2                        2
1478              11                  7                        1
1479               1                  0                        0

     YearsWithCurrManager
0                     0.0
1                     0.0
2                     0.0
3                     0.0
4                     0.0
...                   ...
1475                 10.0
1476                 11.0
```

```
1477                 2.0
1478                 9.0
1479                 0.0

[1480 rows x 38 columns]
```

DATA PROFILING

```
df.head()
```

```
   EmpID  Age AgeGroup Attrition     BusinessTravel  DailyRate  \
0  RM297   18    18-25       Yes      Travel_Rarely        230
1  RM302   18    18-25        No      Travel_Rarely        812
2  RM458   18    18-25       Yes  Travel_Frequently       1306
3  RM728   18    18-25        No         Non-Travel        287
4  RM829   18    18-25       Yes         Non-Travel        247

               Department  DistanceFromHome  Education EducationField  ...  \
0  Research & Development                 3          3  Life Sciences  ...
1                   Sales                10          3        Medical  ...
2                   Sales                 5          3      Marketing  ...
3  Research & Development                 5          2  Life Sciences  ...
4  Research & Development                 8          1        Medical  ...

   RelationshipSatisfaction  StandardHours  StockOptionLevel  \
0                         3             80                 0
1                         1             80                 0
2                         4             80                 0
3                         4             80                 0
4                         4             80                 0

  TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany
\
0                 0                      2                3               0
1                 0                      2                3               0
2                 0                      3                3               0
3                 0                      2                3               0
4                 0                      0                3               0

  YearsInCurrentRole  YearsSinceLastPromotion YearsWithCurrManager
0                  0                        0                 0.0
1                  0                        0                 0.0
2                  0                        0                 0.0
3                  0                        0                 0.0
4                  0                        0                 0.0

[5 rows x 38 columns]
```

```
df.columns
```

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

```
df.shape
```

```
(1480, 38)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1480 entries, 0 to 1479
Data columns (total 38 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   EmpID                     1480 non-null   object
 1   Age                       1480 non-null   int64
 2   AgeGroup                  1480 non-null   object
 3   Attrition                 1480 non-null   object
 4   BusinessTravel            1480 non-null   object
 5   DailyRate                 1480 non-null   int64
 6   Department                1480 non-null   object
 7   DistanceFromHome          1480 non-null   int64
 8   Education                 1480 non-null   int64
 9   EducationField            1480 non-null   object
 10  EmployeeCount             1480 non-null   int64
 11  EmployeeNumber            1480 non-null   int64
 12  EnvironmentSatisfaction   1480 non-null   int64
 13  Gender                    1480 non-null   object
 14  HourlyRate                1480 non-null   int64
 15  JobInvolvement            1480 non-null   int64
 16  JobLevel                  1480 non-null   int64
 17  JobRole                   1480 non-null   object
 18  JobSatisfaction           1480 non-null   int64
 19  MaritalStatus             1480 non-null   object
 20  MonthlyIncome             1480 non-null   int64
 21  SalarySlab                1480 non-null   object
 22  MonthlyRate               1480 non-null   int64
 23  NumCompaniesWorked        1480 non-null   int64
 24  Over18                    1480 non-null   object
```

```
 25  OverTime                  1480 non-null   object
 26  PercentSalaryHike         1480 non-null   int64
 27  PerformanceRating         1480 non-null   int64
 28  RelationshipSatisfaction  1480 non-null   int64
 29  StandardHours             1480 non-null   int64
 30  StockOptionLevel          1480 non-null   int64
 31  TotalWorkingYears         1480 non-null   int64
 32  TrainingTimesLastYear     1480 non-null   int64
 33  WorkLifeBalance           1480 non-null   int64
 34  YearsAtCompany            1480 non-null   int64
 35  YearsInCurrentRole        1480 non-null   int64
 36  YearsSinceLastPromotion   1480 non-null   int64
 37  YearsWithCurrManager      1423 non-null   float64
dtypes: float64(1), int64(25), object(12)
memory usage: 439.5+ KB
```

```
df.describe()
```

```
               Age    DailyRate  DistanceFromHome    Education  EmployeeCount
\
count  1480.000000  1480.000000       1480.000000  1480.000000         1480.0
mean     36.917568   801.384459          9.220270     2.910811            1.0
std       9.128559   403.126988          8.131201     1.023796            0.0
min      18.000000   102.000000          1.000000     1.000000            1.0
25%      30.000000   465.000000          2.000000     2.000000            1.0
50%      36.000000   800.000000          7.000000     3.000000            1.0
75%      43.000000  1157.000000         14.000000     4.000000            1.0
max      60.000000  1499.000000         29.000000     5.000000            1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate  JobInvolvement
\
count     1480.000000              1480.000000  1480.000000     1480.000000
mean      1031.860811                 2.724324    65.845270        2.729730
std        605.955046                 1.092579    20.328266        0.713007
min          1.000000                 1.000000    30.000000        1.000000
25%        493.750000                 2.000000    48.000000        2.000000
50%       1027.500000                 3.000000    66.000000        3.000000
75%       1568.250000                 4.000000    83.000000        3.000000
max       2068.000000                 4.000000   100.000000        4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1480.000000  ...               1480.000000         1480.0
mean      2.064865  ...                  2.708784           80.0
std       1.105574  ...                  1.081995            0.0
min       1.000000  ...                  1.000000           80.0
25%       1.000000  ...                  2.000000           80.0
50%       2.000000  ...                  3.000000           80.0
75%       3.000000  ...                  4.000000           80.0
max       5.000000  ...                  4.000000           80.0
```

```
       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1480.000000        1480.000000            1480.000000
mean           0.791892          11.281757               2.797973
std            0.850527           7.770870               1.288791
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1480.000000     1480.000000         1480.000000
mean          2.760811        7.009459            4.228378
std           0.707024        6.117945            3.616020
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1480.000000           1423.000000
mean                  2.182432              4.118060
std                   3.219357              3.555484
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 26 columns]
```

CHECKING THE UNIQUE VALUE

```
df.nunique()
```

```
EmpID                     1470
Age                         43
AgeGroup                     5
Attrition                    2
BusinessTravel               4
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
SalarySlab                    4
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
```

DATA CLEANING

Identify and handle missing values

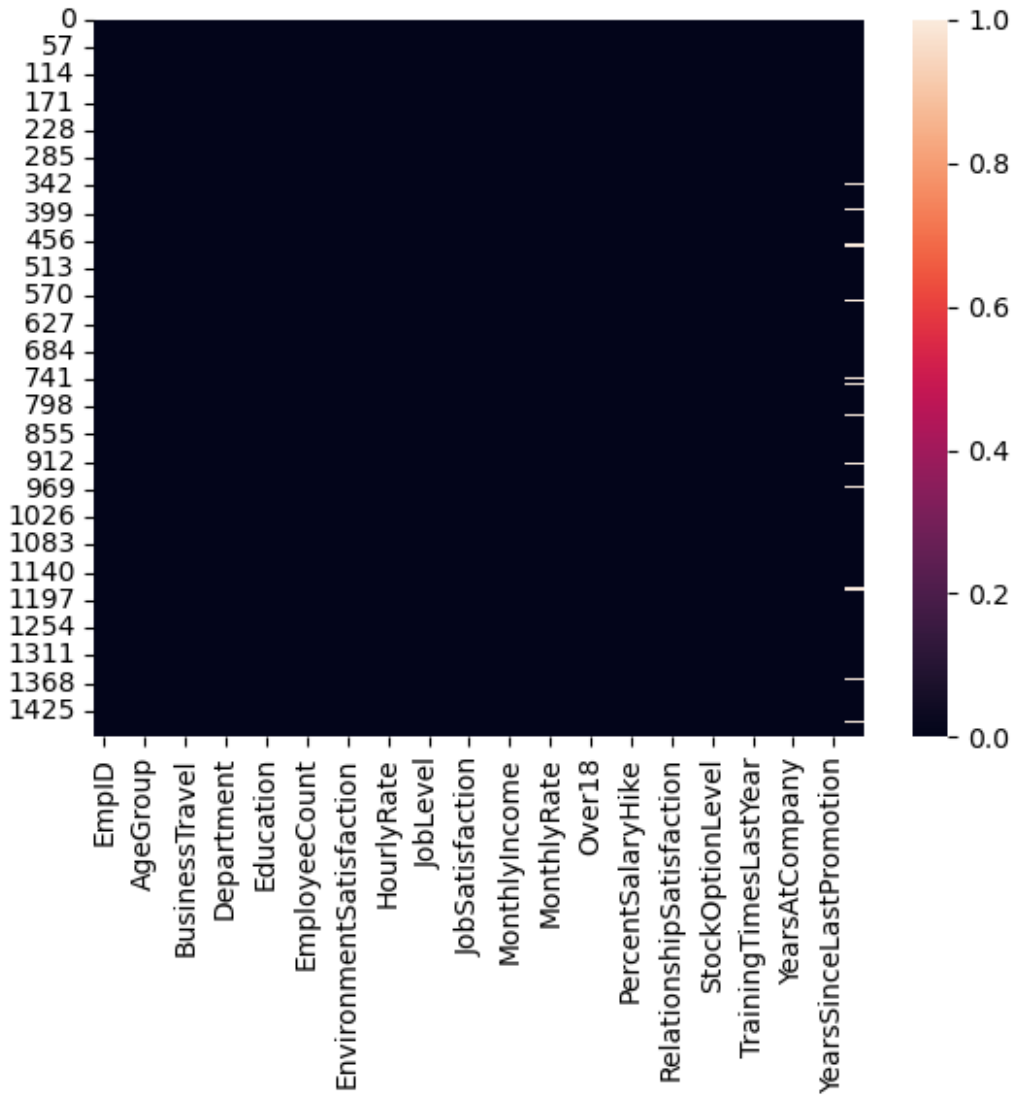
```
df.isnull().sum()
```

```
EmpID                        0
Age                          0
AgeGroup                     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
SalarySlab                   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        57
dtype: int64
```

```
sns.heatmap(df.isnull())
```

```
<Axes: >
```




```
df['YearsWithCurrManager'].fillna(0, inplace=True)
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_3864\3902701200.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.
```

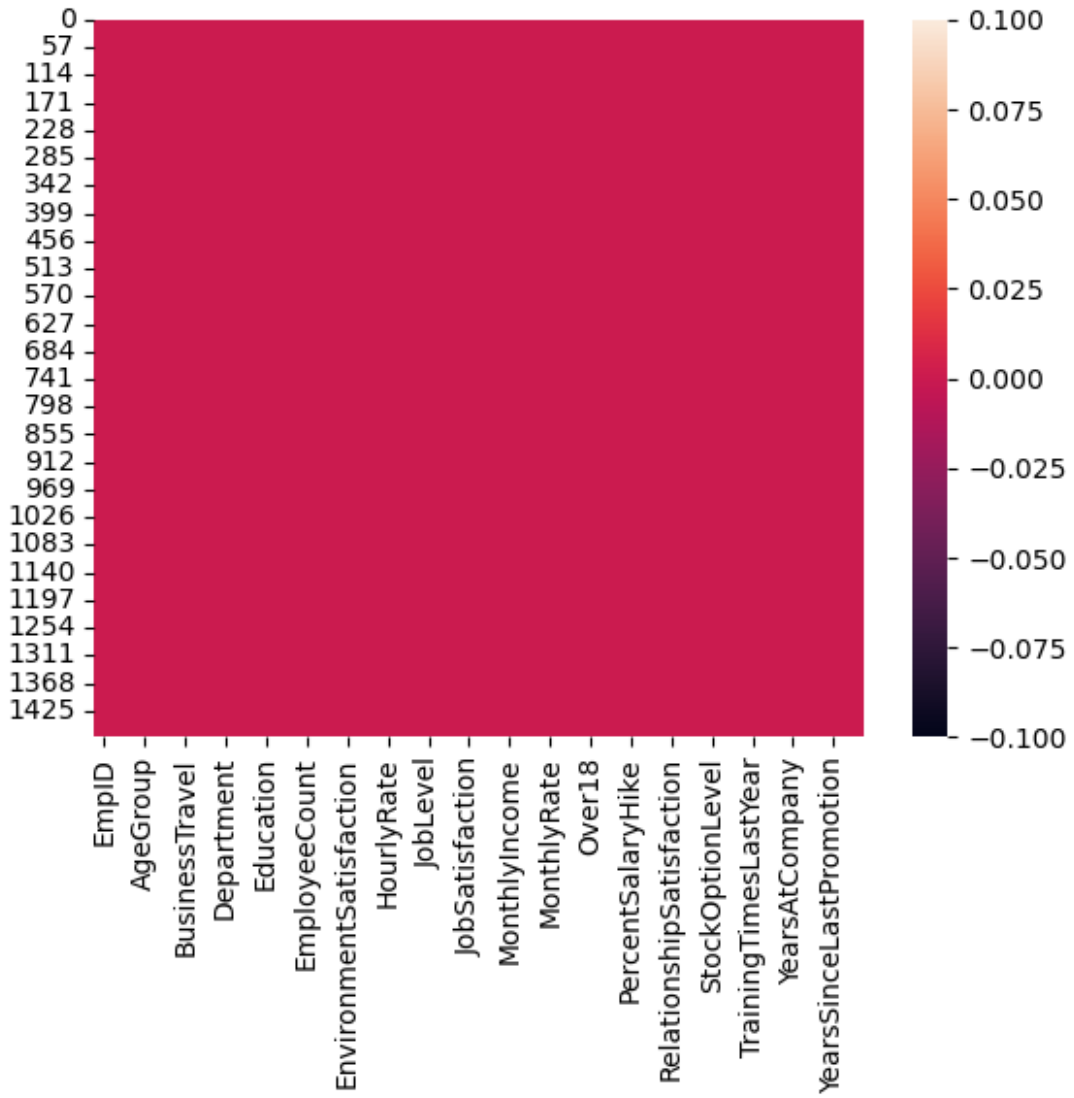
```
  df['YearsWithCurrManager'].fillna(0, inplace=True)
```

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

```
EmpID                     0
Age                       0
AgeGroup                  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
SalarySlab                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
```

```
sns.heatmap(df.isnull())
```

```
<Axes: >
```



Blanks in YearsWithCurrManager likely indicate new hires (0 years)or recent manager changes so it would be better to classiy all the cases as 0 for a smooth estimate

NO MORE NULL VALUE LEFT IN THE DATASET

MADE A COPY OF THE DATASET FOR PREPROCESSING AND EDA

```
df1 = df.copy()
df1.head()
```

```
   EmpID  Age AgeGroup Attrition     BusinessTravel  DailyRate  \
0  RM297   18    18-25       Yes      Travel_Rarely        230
1  RM302   18    18-25        No      Travel_Rarely        812
2  RM458   18    18-25       Yes  Travel_Frequently       1306
3  RM728   18    18-25        No         Non-Travel        287
4  RM829   18    18-25       Yes         Non-Travel        247
```

```
              Department  DistanceFromHome  Education EducationField  ...  \
0  Research & Development                 3          3  Life Sciences  ...
1                   Sales                10          3        Medical  ...
2                   Sales                 5          3      Marketing  ...
3  Research & Development                 5          2  Life Sciences  ...
4  Research & Development                 8          1        Medical  ...

   RelationshipSatisfaction  StandardHours  StockOptionLevel  \
0                         3             80                 0
1                         1             80                 0
2                         4             80                 0
3                         4             80                 0
4                         4             80                 0

  TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany
\
0                 0                      2                3               0
1                 0                      2                3               0
2                 0                      3                3               0
3                 0                      2                3               0
4                 0                      0                3               0

  YearsInCurrentRole  YearsSinceLastPromotion YearsWithCurrManager
0                  0                        0                 0.0
1                  0                        0                 0.0
2                  0                        0                 0.0
3                  0                        0                 0.0
4                  0                        0                 0.0

[5 rows x 38 columns]
```

Preprocessing

```
df1[df1.duplicated()]
```

```
       EmpID  Age AgeGroup Attrition     BusinessTravel  DailyRate  \
211   RM1468   27    26-35        No      Travel_Rarely        155
328   RM1461   29    26-35        No      Travel_Rarely        468
458   RM1464   31    26-35        No         Non-Travel        325
655   RM1470   34    26-35        No       TravelRarely        628
954   RM1463   39    36-45        No      Travel_Rarely        722
1305  RM1469   49    46-55        No  Travel_Frequently       1023
1336  RM1462   50    46-55       Yes      Travel_Rarely        410

                  Department  DistanceFromHome  Education EducationField  ...
\
211   Research & Development                 4          3  Life Sciences  ...
328   Research & Development                28          4        Medical  ...
458   Research & Development                 5          3        Medical  ...
655   Research & Development                 8          3        Medical  ...
```

```
954                    Sales                 24          1      Marketing  ...
1305                   Sales                  2          3        Medical  ...
1336                   Sales                 28          3      Marketing  ...

      RelationshipSatisfaction  StandardHours  StockOptionLevel  \
211                          2             80                 1
328                          2             80                 0
458                          2             80                 0
655                          1             80                 0
954                          1             80                 1
1305                         4             80                 0
1336                         2             80                 1

     TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
211                  6                      0                3
328                  5                      3                1
458                 10                      2                3
655                  6                      3                4
954                 21                      2                2
1305                17                      3                2
1336                20                      3                3

      YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
211                6                   2                        0
328                5                   4                        0
458                9                   4                        1
655                4                   3                        1
954               20                   9                        9
1305               9                   6                        0
1336               3                   2                        2

     YearsWithCurrManager
211                   3.0
328                   4.0
458                   7.0
655                   2.0
954                   6.0
1305                  8.0
1336                  0.0

[7 rows x 38 columns]
```

Searching for Ghost employees

By the defination of it ghost employees represent non-existent workers listed on a company's payroll, enabling fraudsters to collect their wages or benefits illegally. To identify them check for duplicated EmpID as EmpID is unique for each individual employee

```
df1[df1.duplicated(subset=['EmpID'])]
```

```
       EmpID  Age AgeGroup Attrition     BusinessTravel  DailyRate  \
162   RM1465   26    26-35        No      Travel_Rarely       1167
211   RM1468   27    26-35        No      Travel_Rarely        155
328   RM1461   29    26-35        No      Travel_Rarely        468
458   RM1464   31    26-35        No         Non-Travel        325
655   RM1470   34    26-35        No       TravelRarely        628
803   RM1466   36    36-45        No  Travel_Frequently        884
954   RM1463   39    36-45        No      Travel_Rarely        722
955   RM1467   39    36-45        No      Travel_Rarely        613
1305  RM1469   49    46-55        No  Travel_Frequently       1023
1336  RM1462   50    46-55       Yes      Travel_Rarely        410

                  Department  DistanceFromHome  Education EducationField  ...
\
162                     Sales                 5          3          Other  ...
211   Research & Development                 4          3  Life Sciences  ...
328   Research & Development                28          4        Medical  ...
458   Research & Development                 5          3        Medical  ...
655   Research & Development                 8          3        Medical  ...
803   Research & Development                23          2        Medical  ...
954                     Sales                24          1      Marketing  ...
955   Research & Development                 6          1        Medical  ...
1305                    Sales                 2          3        Medical  ...
1336                    Sales                28          3      Marketing  ...

     RelationshipSatisfaction  StandardHours  StockOptionLevel  \
162                         4             80                 0
211                         2             80                 1
328                         2             80                 0
458                         2             80                 0
655                         1             80                 0
803                         3             80                 1
954                         1             80                 1
955                         1             80                 1
1305                        4             80                 0
1336                        2             80                 1

     TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
162                  5                      2                3
211                  6                      0                3
328                  5                      3                1
458                 10                      2                3
655                  6                      3                4
803                 17                      3                3
954                 21                      2                2
955                  9                      5                3
1305                17                      3                2
1336                20                      3                3
```

```
      YearsAtCompany YearsInCurrentRole  YearsSinceLastPromotion  \
162                4                  2                        0
211                6                  2                        0
328                5                  4                        0
458                9                  4                        1
655                4                  3                        1
803                5                  2                        0
954               20                  9                        9
955                7                  7                        1
1305               9                  6                        0
1336               3                  2                        2

     YearsWithCurrManager
162                   5.0
211                   3.0
328                   4.0
458                   7.0
655                   2.0
803                   2.0
954                   6.0
955                   1.0
1305                  8.0
1336                  0.0

[10 rows x 38 columns]
```

There are 10 such Ghost employees. Most of them are from R&D Department. So we meet out objective of finding out the ghost employee.Take note of it and inform the payroll department immediately. So that they can take strict action on it.

FINDING OUT SALARY LEAKAGE

by the definition salary leakage indicates loss in the process of paying employee due to errors, inefficiencies, or fraud, leading to overpayments or excess spending In this case we are investigative for Ghost Employees which falls under fraud category

```
dup_empid_df = df1[df1.duplicated(subset=['EmpID'])].copy()
daily_cost_sum = dup_empid_df['DailyRate'].sum()
print(f"Total daily cost for duplicated EmpIDs: {daily_cost_sum}")
```

```
Total daily cost for duplicated EmpIDs: 6395
```

```
print(dup_empid_df[['EmpID', 'YearsAtCompany']])
```

```
       EmpID  YearsAtCompany
162   RM1465               4
211   RM1468               6
328   RM1461               5
458   RM1464               9
655   RM1470               4
```

```
803   RM1466              5
954   RM1463             20
955   RM1467              7
1305  RM1469              9
1336  RM1462              3
```

```
df[df['EmpID'] == 'RM1463']
```

```
      EmpID  Age AgeGroup Attrition BusinessTravel  DailyRate Department  \
952  RM1463   39    36-45        No  Travel_Rarely        722      Sales
954  RM1463   39    36-45        No  Travel_Rarely        722      Sales

     DistanceFromHome  Education EducationField  ...  \
952                24          1      Marketing  ...
954                24          1      Marketing  ...

     RelationshipSatisfaction  StandardHours  StockOptionLevel  \
952                         1             80                 1
954                         1             80                 1

    TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany
\
952                21                      2                2              20
954                21                      2                2              20

    YearsInCurrentRole  YearsSinceLastPromotion YearsWithCurrManager
952                  9                        9                 6.0
954                  9                        9                 6.0

[2 rows x 38 columns]
```

We are using Monthly rate for obtaining salary leakage info when multiplied by YearAtCompany, as it provides a consistent base pay figure for accurate total lifetime payroll exposure per employee.

```
dup_empid_df['SalaryLeakage'] = dup_empid_df['YearsAtCompany'] * 12 *
dup_empid_df['MonthlyRate']
sum_of_salary_leakage = dup_empid_df['SalaryLeakage'].sum()
print(f"Sum of salary leakage: {sum_of_salary_leakage}")
```

```
Sum of salary leakage: 9493800
```

9493800 this is the amount of money been spend for non existent employees turns out to be a total loss for the company

```
dup_empid_df['SalaryLeakage'] = dup_empid_df['YearsAtCompany'] * 12 *
dup_empid_df['MonthlyRate']
print(dup_empid_df[['EmpID', 'SalaryLeakage']])
```

```
      EmpID  SalaryLeakage
162  RM1465        1026144
```

```
211   RM1468         372528
328   RM1461         509340
458   RM1464         408996
655   RM1470         490944
803   RM1466         737400
954   RM1463        2118720
955   RM1467        1802388
1305  RM1469        1430244
1336  RM1462         597096
```

This is the individual breakdown of Salary Leakage

```
columns_for_csv = ['EmpID', 'Department', 'JobRole', 'MonthlyRate',
'YearsAtCompany', 'SalaryLeakage']
dup_empid_df[columns_for_csv].to_csv('ghost_employees_salary_leakage.csv',
index=False)
```

```
df1.columns
```

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

Investigating Overtime abuse

by the definition it means occurs when employees fraudulently work unauthorized extra hours to receive overtime pay. There are many indicators to find whether the employee is misusing the Overtime for earning the extra payment. But for this dataset we are going to analyse any overtime abuse by considering job involvement column (logic is when the employee does overtime despite of having less the job involvement rating) they are suspected as Overtime Abuser.

```
df1['Abuse_Flag'] = np.where((df1['JobInvolvement']==2) & (df1['OverTime']
=='Yes'), 'SUSPECT', 'NORMAL')
print(df1[df1['Abuse_Flag']=='SUSPECT'][['EmpID', 'Department', 'OverTime',
'JobInvolvement']])
```

```
      EmpID              Department OverTime  JobInvolvement
13    RM689                   Sales      Yes               2
15    RM893  Research & Development      Yes               2
```

```
17    RM103  Research & Development      Yes              2
22    RM732  Research & Development      Yes              2
30    RM358                   Sales      Yes              2
...     ...                     ...      ...            ...
1453  RM158  Research & Development      Yes              2
1456  RM661  Research & Development      Yes              2
1457  RM675  Research & Development      Yes              2
1460  RM967  Research & Development      Yes              2
1471  RM744  Research & Development      Yes              2

[105 rows x 4 columns]
```

```
df1['Abuse_Flag'].value_counts()
```

```
Abuse_Flag
NORMAL     1365
SUSPECT     105
Name: count, dtype: int64
```

106 such overtime abuser has been detected whose job involvement rating is 2 or less than 2 and they are having overtime.

```
df1.drop_duplicates(subset=['EmpID'], inplace=True)
```

```
df1[df1.duplicated()]
```

```
Empty DataFrame
Columns: [EmpID, Age, AgeGroup, Attrition, BusinessTravel, DailyRate,
Department, DistanceFromHome, Education, EducationField, EmployeeCount,
EmployeeNumber, EnvironmentSatisfaction, Gender, HourlyRate, JobInvolvement,
JobLevel, JobRole, JobSatisfaction, MaritalStatus, MonthlyIncome, SalarySlab,
MonthlyRate, NumCompaniesWorked, OverTime, PercentSalaryHike,
PerformanceRating, RelationshipSatisfaction, StandardHours, StockOptionLevel,
TotalWorkingYears, TrainingTimesLastYear, WorkLifeBalance, YearsAtCompany,
YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager,
Abuse_Flag]
Index: []

[0 rows x 38 columns]
```

```
df1.duplicated().sum()
```

```
0
```

```
df.drop('Over18', axis=1, inplace=True)
```

Removed the Over18 column as it only contain a single unique value and obviously everyone in the company is 18 or above

```
df1.shape
```

```
(1470, 37)
```

```
df.columns
```

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

```
df.to_csv('Clean HR data.csv',index = False)
```

EDA

```
df.dtypes
```

```
EmpID                      object
Age                         int64
AgeGroup                   object
Attrition                  object
BusinessTravel             object
DailyRate                   int64
Department                 object
DistanceFromHome            int64
Education                   int64
EducationField             object
EmployeeCount               int64
EmployeeNumber              int64
EnvironmentSatisfaction     int64
Gender                     object
HourlyRate                  int64
JobInvolvement              int64
JobLevel                    int64
JobRole                    object
JobSatisfaction             int64
MaritalStatus              object
MonthlyIncome               int64
SalarySlab                 object
MonthlyRate                 int64
NumCompaniesWorked          int64
OverTime                   object
PercentSalaryHike           int64
```

```
PerformanceRating             int64
RelationshipSatisfaction      int64
StandardHours                 int64
StockOptionLevel              int64
TotalWorkingYears             int64
TrainingTimesLastYear         int64
WorkLifeBalance               int64
YearsAtCompany                int64
YearsInCurrentRole            int64
YearsSinceLastPromotion       int64
YearsWithCurrManager        float64
Abuse_Flag                   object
dtype: object
```

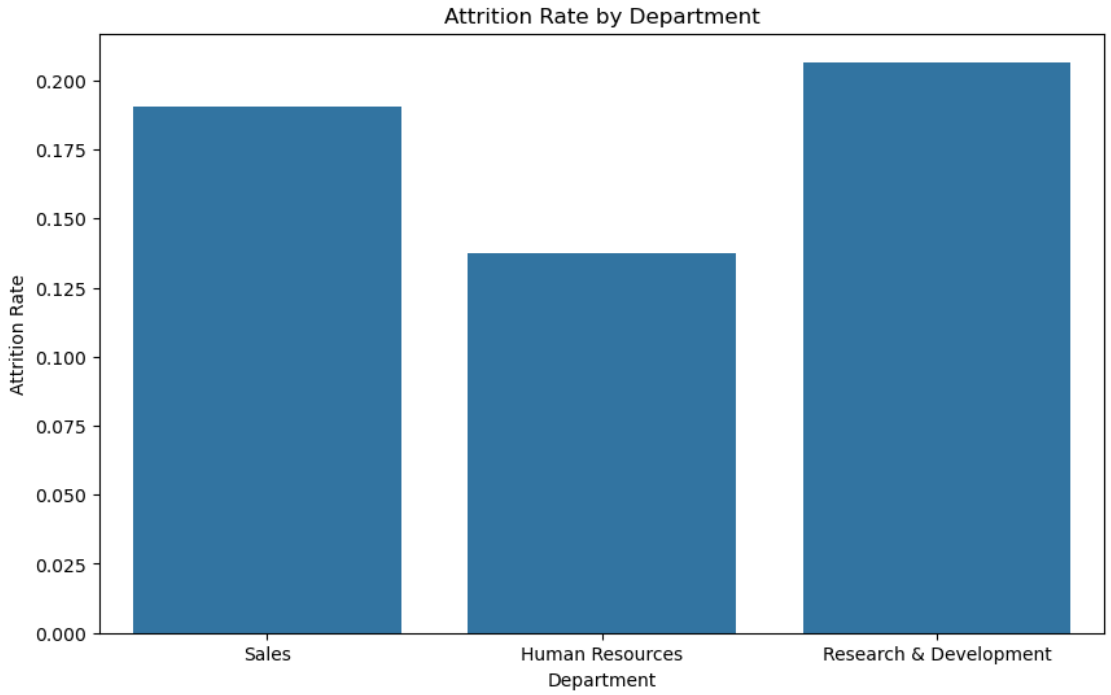
Attrition

Aim to understand “who leaves” using group by columns First need to convert Attrition dtype from string to int())

```
df['Attrition_Flag'] = (df['Attrition'] == 'Yes').astype(int)
```

```
df['Attrition_Flag'] = (df['Attrition'] == 'Yes').astype(int)
```

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df.groupby('Department')['Attrition_Flag'].mean().sort_values(a
scending=False).index,
            y=df.groupby('Department')['Attrition_Flag'].mean())
plt.title('Attrition Rate by Department')
plt.xlabel('Department')
plt.ylabel('Attrition Rate')
plt.show()
```



ATTRITION BY DEPARTMENT

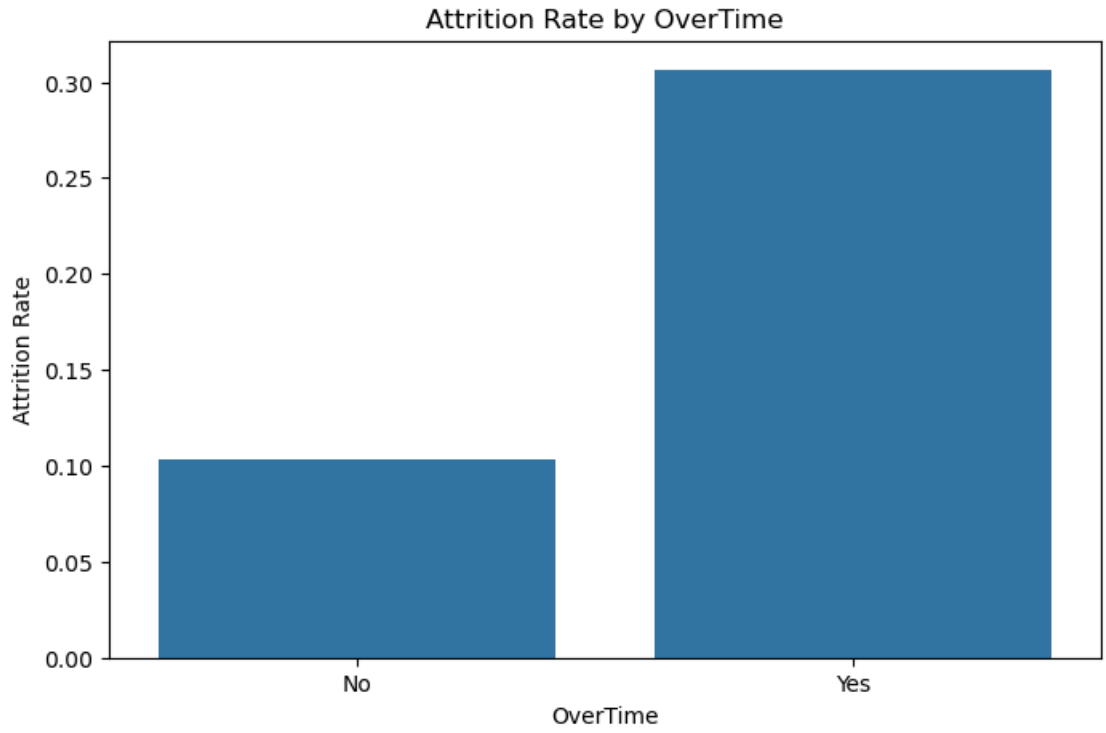
shows that R&D has more rate of attrition signifies more chance of leaving

```
plt.figure(figsize=(10, 6))
sns.barplot(x=df.groupby('SalarySlab')['Attrition_Flag'].mean().index,
            y=df.groupby('SalarySlab')['Attrition_Flag'].mean().values)
plt.title('Attrition Rate by Salary Slab')
plt.xlabel('Salary Slab')
plt.ylabel('Attrition Rate')
plt.show()
```

ATTRITION BY SALARYSLAB

shows that employee with salary slab of upto 5k has more rate of attrition signifies more chance of leaving while interestingly 10k-15k slab ones are more lenient towards leaving than 5k-10k

```
plt.figure(figsize=(8, 5))
sns.barplot(x=df.groupby('OverTime')['Attrition_Flag'].mean().index,
            y=df.groupby('OverTime')['Attrition_Flag'].mean().values)
plt.title('Attrition Rate by OverTime')
plt.xlabel('OverTime')
plt.ylabel('Attrition Rate')
plt.show()
```

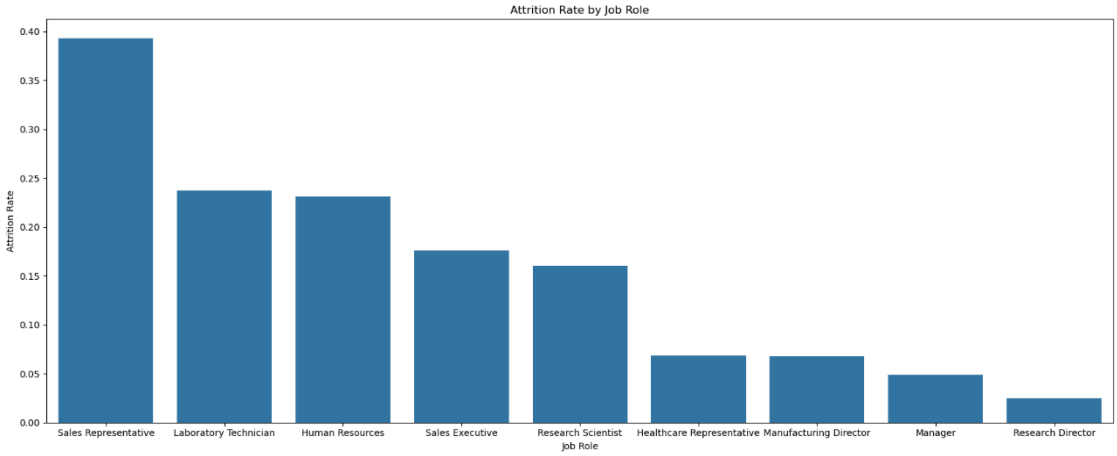


ATTRITION BY OVERTIME

shows a quiet obvious visual overtime employees are most forward in this race of leaving

```
plt.figure(figsize=(17, 7))
sns.barplot(x=df.groupby('JobRole')['Attrition_Flag'].mean().sort_values(asce
nding=False).index,

y=df.groupby('JobRole')['Attrition_Flag'].mean().sort_values(ascending=False)
.values)
plt.title('Attrition Rate by Job Role')
plt.xlabel('Job Role')
plt.ylabel('Attrition Rate')
plt.tight_layout()
plt.show()
```



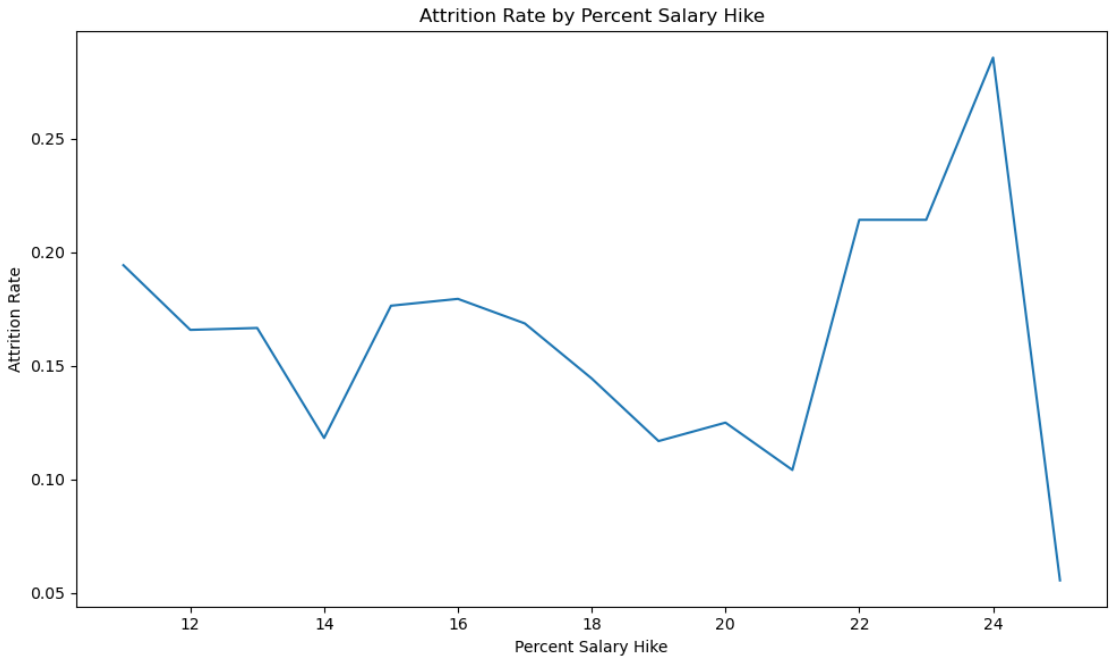
ATTRITION BY JOB ROLE

shows that Sales Representative are most likely to be attrition prone while research director is the least among the role

```
print(df.groupby('PercentSalaryHike')['Attrition_Flag'].mean())
```

```
PercentSalaryHike
11    0.194313
12    0.165829
13    0.166667
14    0.118227
15    0.176471
16    0.179487
17    0.168675
18    0.144444
19    0.116883
20    0.125000
21    0.104167
22    0.214286
23    0.214286
24    0.285714
25    0.055556
Name: Attrition_Flag, dtype: float64
```

```
attrition_by_hike = df.groupby('PercentSalaryHike')['Attrition_Flag'].mean()
plt.figure(figsize=(10, 6))
plt.plot(attrition_by_hike.index, attrition_by_hike.values)
plt.title('Attrition Rate by Percent Salary Hike')
plt.xlabel('Percent Salary Hike')
plt.ylabel('Attrition Rate')
plt.tight_layout()
plt.show()
```



ATTRITION BY % OF SALARY HIKE

shows interesting insight more the % of salary hike more are the chances of leaving exception is 25% of hike as it is nearly not possible for the company to give such a big hike at once.(21-24%) tends to has more attrition chance than other

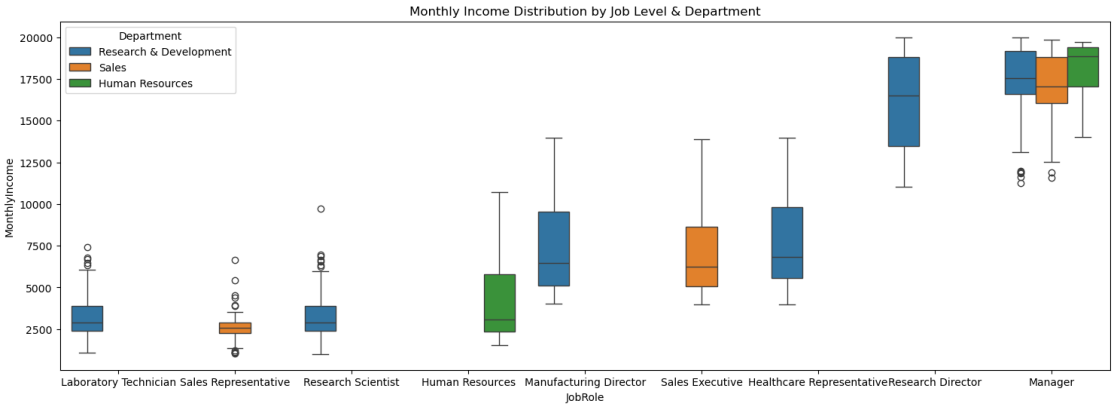
```
print(df.groupby('Gender')['Attrition_Flag'].mean())
```

```
Gender
Female    0.147208
Male      0.169854
Name: Attrition_Flag, dtype: float64
```

ATTRITION BY GENDER

Male has more average of attrition rate than female but still not so significant insight

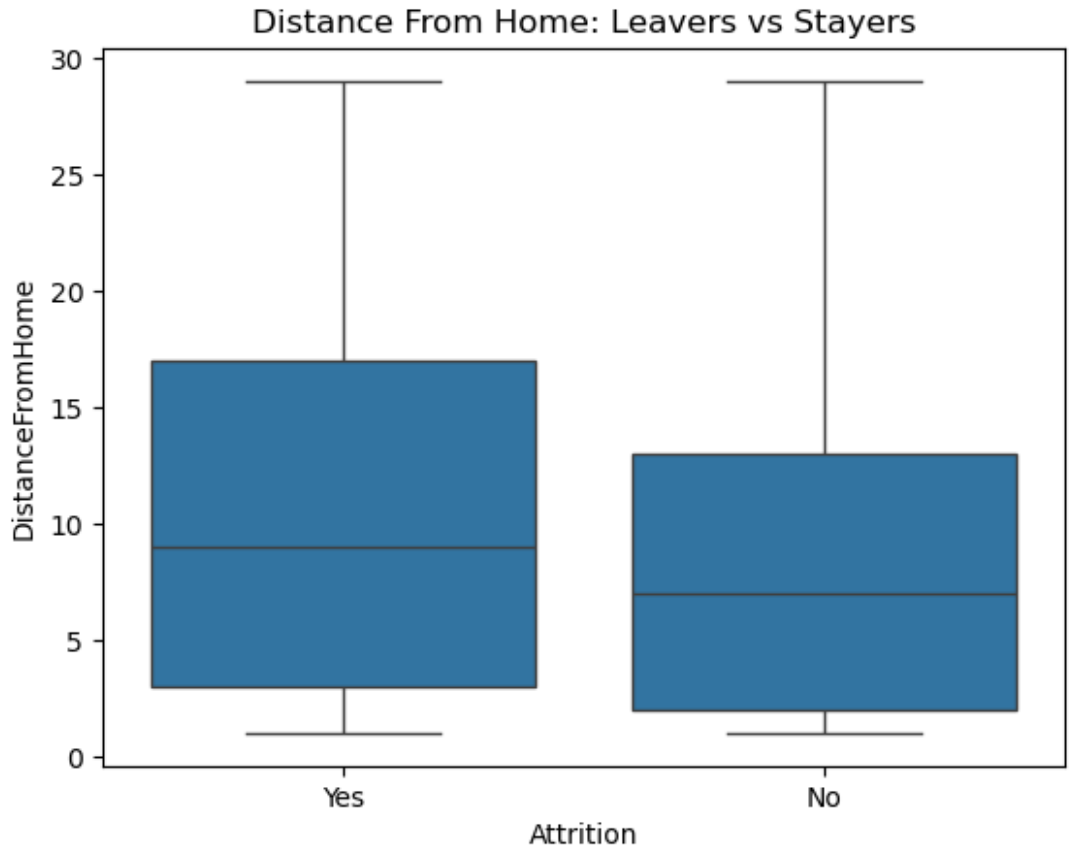
```
plt.figure(figsize=(18, 6))
sns.boxplot(data=df, x='JobRole', y='MonthlyIncome', hue='Department')
plt.title('Monthly Income Distribution by Job Level & Department')
plt.show()
```



Average Monthly Income by JobRole

Sales Executives show the widest income spread indicating inconsistent compensation practices that likely fuel dissatisfaction and attrition, while Lab Technicians across departments earn the lowest medians posing high turnover risk.

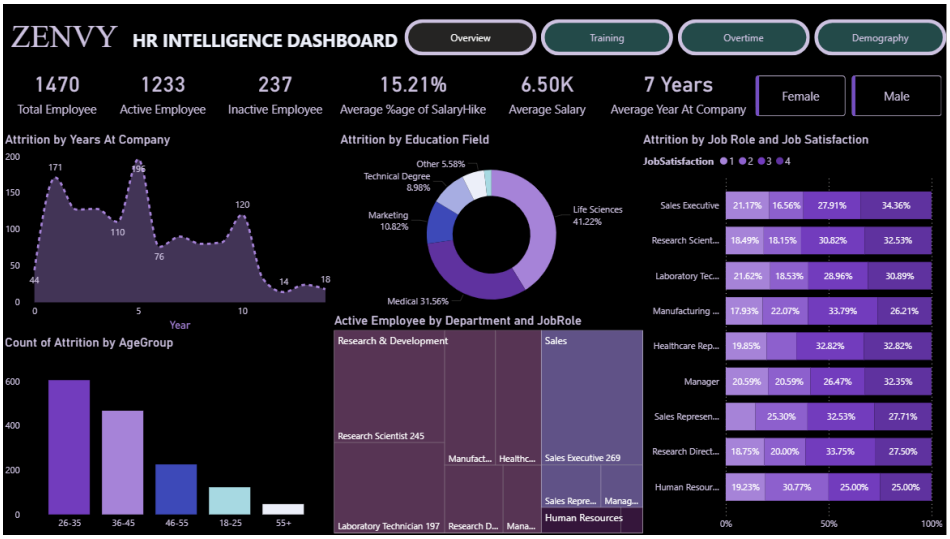
```
sns.boxplot(data=df, x='Attrition', y='DistanceFromHome')
plt.title('Distance From Home: Leavers vs Stayers')
plt.show()
```



Distance From Home vs Attrition

Yes Distance from home does matter in attrition as YES has higher median than NO

After this Analysis we moved on to the dashboarding part in POWERBI.



ZENVY HR Intelligence Dashboard KPIs:

OVERVIEW consist of KPI's like Total Employees, Total Hires, Total Attrition, Attrition Rate

Average Salary, Average Tenure

Attrition Visuals:

Attrition Count by Year at Company (Area chart)

Active Employees by Department and Age Group (Tree map)

Education Breakdown (donut)

Attrition by Job Role and Job Satisfaction (stacked bars)

Attrition by Age Group (Bar)

ZENVY HR Intelligence Dashboard KPIs

Training Total Employees, Turnover Rate, Cost per Trained Employee, Total Salary Leakage, Number of Ghost Employee, Total Training Cost

Turnover and training Visuals:

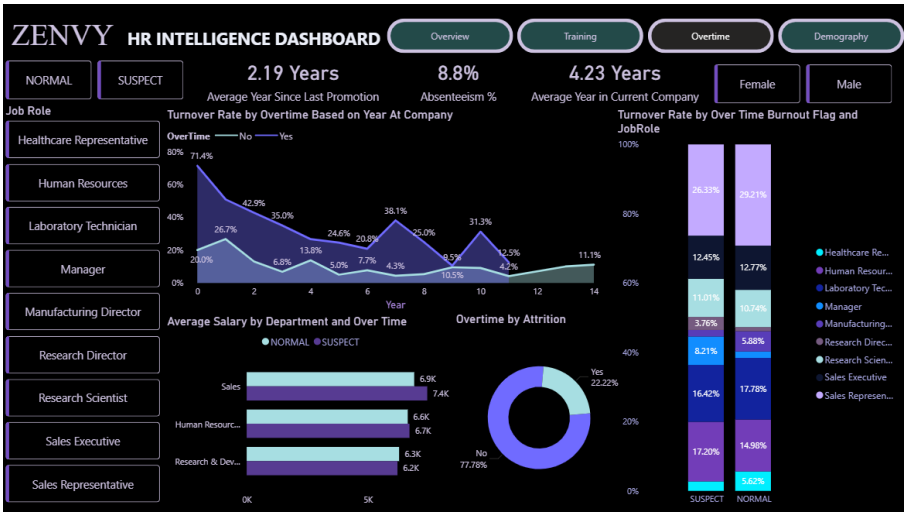
Ghost Employee (Matrix Table)

Cost per Trained Ghost Employees (donut)

By Department and salary slab and department (bar)

Cost per Trained Employee by Years at Company (Area)

Turnover Rate by Department and job role (stacked bars):



ZENVY HR Intelligence Dashboard KPIs

Salary: Average Years Since Last Promotion, Absenteeism Rate, Average Years at Company

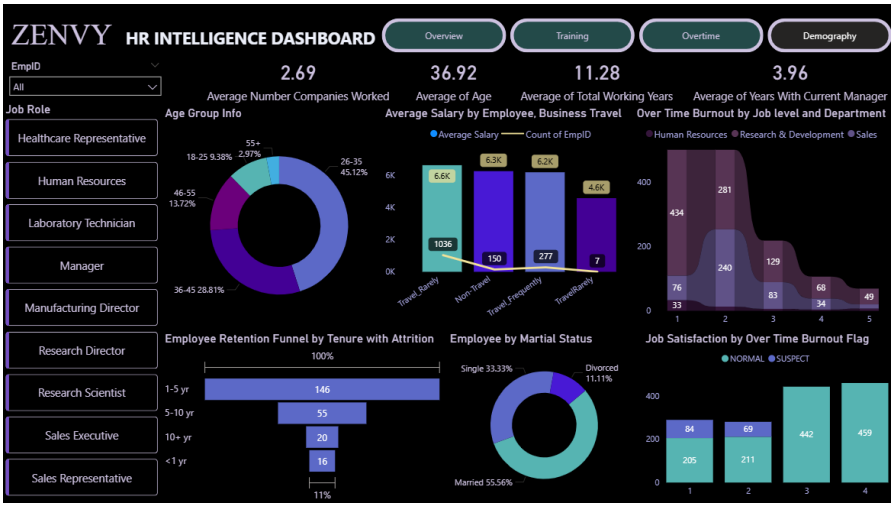
Overtime Visuals:

Turnover Rate by Overtime Based on Year at Company (Area Chart)

Turnover Rate by Overtime Burnout Flag and Job Role (Stacked Bar)

Overtime by Attrition(donut)

Average Salary by Department and Overtime (Bar)



ZENVY HR Intelligence Dashboard KPIs

Demographics Average number company worked, Average of age, Average of total working years, Average of Years with Current Manager

Demographic Visual:

Age group info(donut)

Employee Retention Funnel by tenure with attrition (Funnel chart)

Average salary by employee and business Travel (bar and line chart)

Overtime Burnout by Job level and department (Ribbon chart)

Employee by Martial status (donut)

Job Satisfaction by Overtime Burnout Flag (stacked bar)

THANK YOU