Employee Attrition | Data Analysis Project

Project Scope

Employee attrition occurs when employees leave an organisation for unpredictable or uncontrollable reasons. Many terms make up attrition, the most common being termination, resignation, planned retirement, structural changes, long-term illness, or layoffs. It often results in a decrease in an organisation's workforce size as employees leave faster than the rate at which their employers hire. Solving the attrition problem within an organisation requires business and human resource leaders to use big-picture, strategic thinking and interventions. (Lucas, S. 2023)

As a data analyst, the main goal of this project was to answer analytical questions about the employee attrition dataset to derive valuable insights and trends. The following analytical questions were the focus of this project.

1. Are certain departments experiencing higher levels of attrition than others? If so, why might that be?
2. What is the average tenure of employees who leave the company, and how does it compare to employees who stay long-term?
3. How does attrition vary based on demographics (e.g., age, gender, education level, etc.)?
4. What are the key indicators or drivers of employee attrition? (e.g. job satisfaction, average monthly hours, salary, promotions, job title, etc.)

Project Approach

The employee attrition dataset used for this project was obtained from Kaggle. The data preparation, cleaning, and exploratory analysis were done using Python. The Python IDE which was used for this project is Spyder. Finally, the dashboard visualisations for the data were created using Tableau.

Part 1. Data Preparation and Cleaning

Before the data could be cleaned up, the dataset had to be loaded into Spyder.

Step 1: Load libraries and the dataset

```python
# Importing libraries
import pandas as pd
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# LOAD DATA
# Save filepath to variable for easier access
employee_attrition_file_path = 'C:/Users/ADMIN/Employee_Attrition/employee_attrition.csv'

# Read and store the data in DataFrame titled employee_data
employee_data = pd.read_csv(employee_attrition_file_path)
```

```
In [22]: print(employee_data.head())
   Employee_ID  Age  Gender  ... Promotion_Last_5Years Salary  Attrition
0            0   27    Male  ...                     0  60132          0
1            1   53  Female  ...                     1  79947          0
2            2   59  Female  ...                     0  46958          1
3            3   42  Female  ...                     0  40662          0
4            4   44  Female  ...                     1  74307          0
```

```
In [23]: print(employee_data.tail())
     Employee_ID  Age  Gender  ... Promotion_Last_5Years Salary  Attrition
995          995   39  Female  ...                     0  71403          0
996          996   50    Male  ...                     0  30181          1
997          997   52    Male  ...                     0  64143          0
998          998   37  Female  ...                     0  74383          1
999          999   59    Male  ...                     0  73220          1
```

Step 2: Preview the data

The dataset comprised **1000** records(rows) and **11** fields(columns).

```
In [10]: print(employee_data.shape)
(1000, 11)
```

The screenshot below shows the names of the columns in the dataset.

```
In [25]: print(employee_data.columns.values)
['Employee_ID' 'Age' 'Gender' 'Department' 'Job_Title' 'Years_at_Company'
 'Satisfaction_Level' 'Average_Monthly_Hours' 'Promotion_Last_5Years'
 'Salary' 'Attrition']
```

The screenshot below shows the column names and their data types.

```
In [24]: print(employee_data.dtypes)
Employee_ID                int64
Age                        int64
Gender                    object
Department                object
Job_Title                 object
Years_at_Company           int64
Satisfaction_Level       float64
Average_Monthly_Hours      int64
Promotion_Last_5Years      int64
Salary                     int64
Attrition                  int64
dtype: object
```

Step 3: Check for missing values

The ‘**False’** value in the screenshot below indicates that the dataset had no missing values.

```
In [29]: print(employee_data.isnull().values.any())
False
```

Step 4: Check for duplicated records/rows

The dataset did not have any duplicated rows/records.

```
In [6]: print(employee_data.duplicated())
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Length: 1000, dtype: bool
```

```
In [7]: print(employee_data.duplicated().value_counts())
False    1000
Name: count, dtype: int64
```

Step 5: Number of unique values in each column

The screenshot below shows the number of unique values in each column.

```
In [20]: print(employee_data.nunique())
Employee_ID               1000
Age                         35
Gender                       2
Department                   5
Job_Title                    5
Years_at_Company            10
Satisfaction_Level        1000
Average_Monthly_Hours      100
Promotion_Last_5Years        2
Salary                     995
Attrition                    2
dtype: int64
```

Step 6: Change data formats

Step 6.1

The data in the '**Satisfaction_Level**' column as the name implies, indicates an employee's level of satisfaction whilst working at the organisation under study. The employee's Satisfaction level is a value (ranging from 0 to 1). For example, 0.586251256, 0.261160889, 0.304381718, etc. These values with so many decimal places are somewhat vague. As such, the data analyst converted the satisfaction level values to percentages (int).

The screenshot below shows the values in the Satisfaction_Level column in the current format.

```
In [43]: print(employee_data['Satisfaction_Level'])
0      0.586251
1      0.261161
2      0.304382
3      0.480779
4      0.636244
         ...
995    0.377435
996    0.431152
997    0.647102
998    0.304813
999    0.940510
Name: Satisfaction_Level, Length: 1000, dtype: float64
```

The screenshot below shows the Satisfaction_Level column with rounded-off values in percentage format.

```
In [56]: print((employee_data['Satisfaction_Level'] *
100).round().astype(int))
0      59
1      26
2      30
3      48
4      64
       ..
995    38
996    43
997    65
998    30
999    94
Name: Satisfaction_Level, Length: 1000, dtype: int32
```

Step 6.2

In the **Attrition** column, the number **one** (**1**) indicates that an employee left the organisation. In contrast, **zero** (**0**) suggests that an employee did not leave the organisation. This format was changed so that '**Yes'** and '**No'** replaced the values **1** and **0** respectively.

```
In [43]:
    ...: print(employee_data['Attrition'].head())
    ...: print(employee_data['Attrition'].tail())
0    0
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int64
995    0
996    1
997    0
998    1
999    1
Name: Attrition, dtype: int64
```

```
In [16]:
    ...: print(employee_data['Attrition'].replace( {0: 'No',
1: 'Yes'}, inplace = True))
    ...:
    ...: # Display attrition coulmn
    ...: print(employee_data['Attrition'].head())
    ...: print(employee_data['Attrition'].tail())
None
0      No
1      No
2     Yes
3      No
4      No
Name: Attrition, dtype: object
995     No
996    Yes
997     No
998    Yes
999    Yes
Name: Attrition, dtype: object
```

Step 7: Identify outlier values in the dataset

An Outlier is a data item/object that deviates significantly from the other (so-called normal) objects. Identifying outliers is important in statistics and data analysis because they can dramatically impact the results of statistical analyses. The analysis for outlier detection is called outlier mining (Geeks for Geeks, 2024).

The outlier values for each column containing numerical data were determined using the column's z-score. The Z-score, also called a standard score, helps to understand how far a data point is from the mean.

The code below was used to calculate the z-score in all the numerical columns of the employee attrition dataset.

```
In [30]: for z_score_cols in employee_data[['Age', 'Years_at_Company',
'Satisfaction_Level',
    ...:             'Average_Monthly_Hours', 'Promotion_Last_5Years', 'Salary']]:
    ...:     column_series = employee_data[z_score_cols]
    ...:     print(np.abs(stats.zscore(employee_data[z_score_cols])))
    ...:     z = np.abs(stats.zscore(employee_data[z_score_cols]))
    ...:
    ...: # Removal of Outliers with Z-Score
    ...: # Let’s remove rows where Z value is greater than 2.
    ...:     threshold_z = 2
    ...:     outlier_indices = np.where(z > threshold_z)[0]
    ...:     no_outliers = employee_data.drop(outlier_indices)
    ...:     print("\nOriginal DataFrame Shape:", employee_data.shape)
    ...:     print("DataFrame Shape after Removing Outliers:", no_outliers.shape)
```

For example, the z-scores for the ‘Age’ column are shown in the screenshot below.

```
0      1.518762
1      1.078266
2      1.677580
3      0.020477
4      0.179295
         ...
995    0.320134
996    0.778609
997    0.978381
998    0.519905
999    1.677580
Name: Age, Length: 1000, dtype: float64
```

The dataset did not have any outlier values as indicated by the screenshot below.

```
Original DataFrame Shape: (1000, 11)
DataFrame Shape after Removing Outliers: (1000, 11)
```

Part 2: Exploratory Data Analysis

The analytical questions which were in the ‘Project Scope’ of the documentation will be answered in this part.

Question 1:

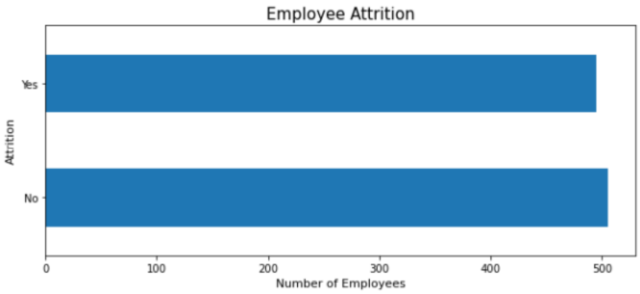
Part A: Are certain departments experiencing higher levels of attrition than others?
Part B: If so, why might that be?

Before question 1 could be answered, the analyst needed to establish the employee attrition statistics. The screenshot below shows that **495** and **505** employees left and stayed respectively.

```
In [18]: attr_count = employee_data['Attrition'].value_counts()
    ...: print(attr_count)
Attrition
No     505
Yes    495
Name: count, dtype: int64
```

Below is the employee attrition bar chart

```
In [34]: employee_data['Attrition'].value_counts().plot(kind =
'barh', figsize = (10,4))
    ...:
    ...: plt.xlabel('Number of Employees', fontsize = 11)
    ...: plt.ylabel('Attrition', fontsize = 11)
    ...: plt.title('Employee Attrition', fontsize = 15)
    ...: plt.show()
```



The organisation under study was made up of five departments namely; **Marketing**, **Sales**, **Engineering**, **HR** and **Finance**.

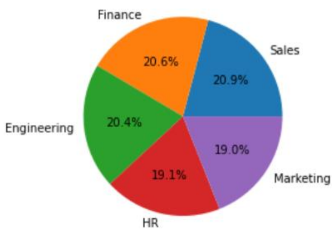
```
In [3]:
   ...: unique_values_dept = employee_data['Department'].unique()
   ...: print('\nThe deparments in the employees dataset are:', unique_values_dept)

The deparments in the employees dataset are: ['Marketing' 'Sales' 'Engineering'
'Finance' 'HR']
```

The distribution of employees across the five departments was as follows;

Marketing:	190	19.0%
Sales:	209	20.9%
Engineering:	204	20.4%
Finance:	206	20.9%
HR:	191	19.1%

```
In [15]: dept_count = employee_data['Department'].value_counts()
    ...: dept_count_pct = (dept_count/sum(dept_count)) * 100
    ...: print('The departmental breakdown is:', dept_count, dept_count_pct)
The departmental breakdown is: Department
Sales          209
Finance        206
Engineering    204
HR             191
Marketing      190
Name: count, dtype: int64 Department
Sales          20.9
Finance        20.6
Engineering    20.4
HR             19.1
Marketing      19.0
Name: count, dtype: float64
```

```
In [18]: dept_labels = ['Sales', 'Finance', 'Engineering', 'HR', 'Marketing']
    ...: plt.pie(dept_count_pct, labels = dept_labels, autopct = '%.1f%%')
    ...: plt.show()
```



The table below shows the attrition levels for each department.

```
In [22]: dept_attr_tbl = pd.crosstab(employee_data.Department, columns = employee_data.Attrition)
    ...: print(dept_attr_tbl)
Attrition     No  Yes
Department
Engineering   95  109
Finance       99  107
HR            95   96
Marketing    110   80
Sales        106  103
```

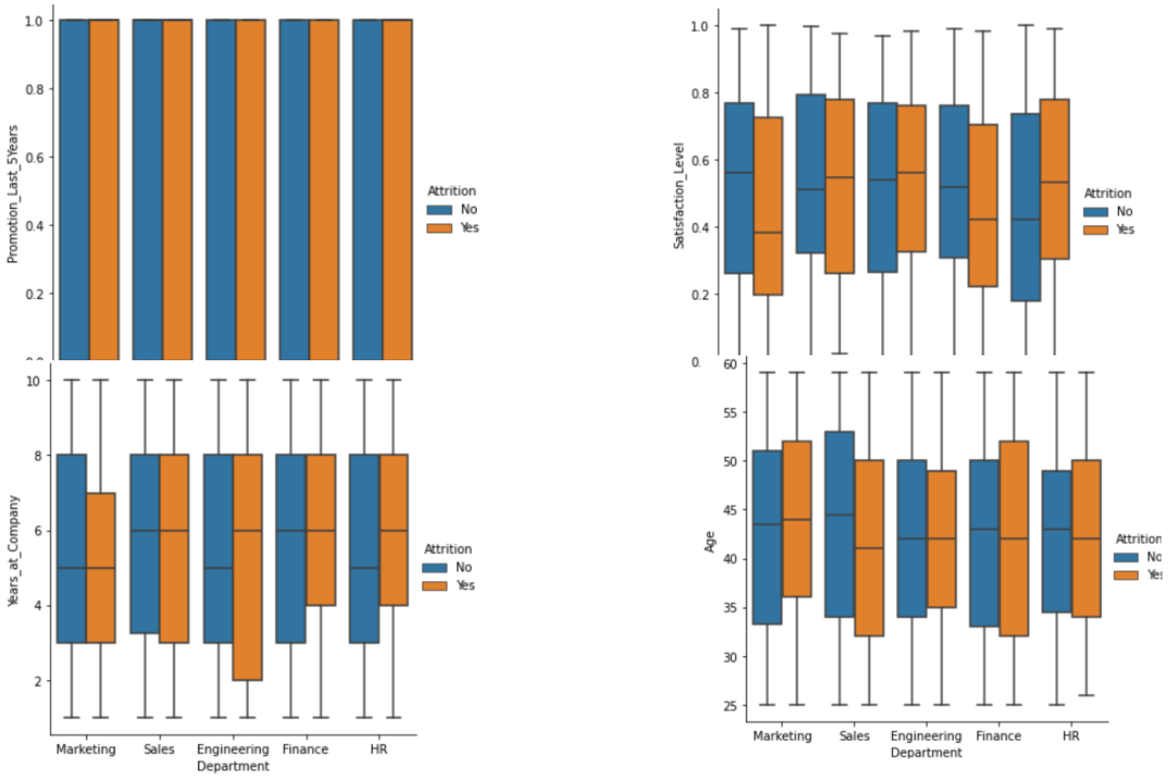
Solution

Part A:

The Engineering department had the highest number of attritions at **109**, whereas the Marketing department had the lowest number at **80**.

Part B:

No definitive attribute resulted in the engineering department having the highest number of attritions as can be seen from the charts below.



Question 2

Part A: What is the average tenure of employees who leave the company?
Part B: How does it compare to employees who stay long-term?

From the screenshot below, the following was established;

- The total number of employees in the dataset is 1000.
- The least number of years served by employees was **1**.
- The mean number of years served by employees was **5** years and 6 months.
- The highest number of years served by employees was **10**.
- 25% (1st quartile) of the employees have served for less than or equal to **3** years.

- 50% (median - 2nd quartile) of the employees have served for less than or equal to **6** years.
- 75% (3rd quartile) of the employees have served for less than or equal to **8** years.

```
In [10]: print(employee_data['Years_at_Company'].describe())
count    1000.000000
mean        5.605000
std         2.822223
min         1.000000
25%         3.000000
50%         6.000000
75%         8.000000
max        10.000000
Name: Years_at_Company, dtype: float64
```

The chart below shows that most of the employees at the company have served for 7 years.

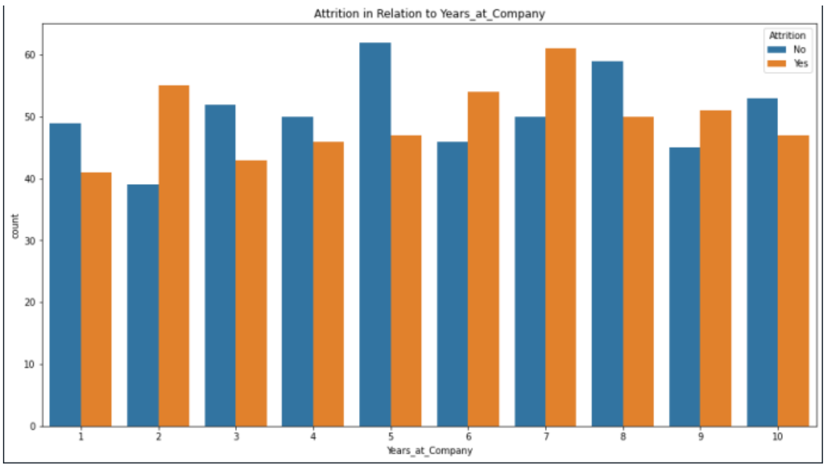
```
In [14]: employee_data['Years_at_Company'].value_counts().plot(kind = 'bar', figsize = (10,4), fontsize = 13)
    ...: plt.xlabel('Years_at_Company', fontsize = 11)
    ...: plt.ylabel('Number of Employees', fontsize = 11)
    ...: plt.title("Employees' Tenure", fontsize = 12)
    ...: plt.show()
```

Solution

The chart below shows that;

- Employees who have served for 7 years have the highest attrition levels.
- **Part A**: The average tenure of employees who leave the organisation is 5.5 years.
- **Part B**: Employees who have served the longest have attrition levels the same as those with an average tenure period.

```
In [21]: plt.subplots(figsize = (15,8))
    ...: sns.countplot(x = 'Years_at_Company', hue = 'Attrition', data = employee_data)
    ...: plt.title('Attrition in Relation to Years_at_Company')
    ...: plt.show()
```



Question 3

How does attrition vary based on demographics (e.g., age, gender, education level, etc.)?

Part A: Age

The number of unique ages of the employees in the dataset was **35.**

```
In [9]: print(employee_data['Age'].nunique())
35
```

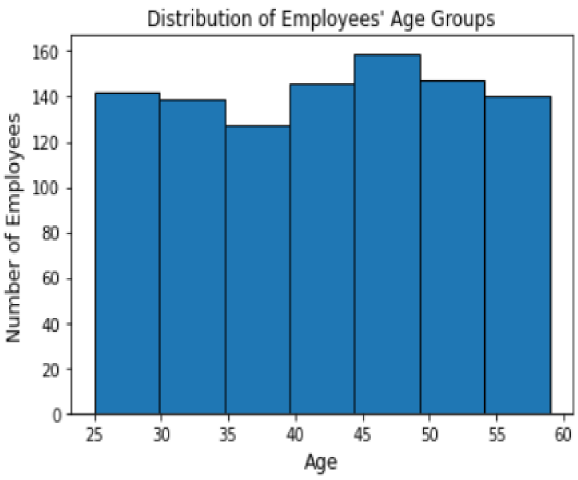
The following was deduced from the screenshot below regarding the ages of the employees;

- The youngest employee was **25** years old.
- The mean age of the employees was **42** years.
- The oldest employee was **59** years old.
- 25% (1st quartile) of the employees were less than or equal to **33** years of age.
- 50% (median – 2nd quartile) of the employees were less than or equal to **43** years of age.
- 75% (3rd quartile) of the employees were less than or equal to **51** years of age.

```
In [10]: print(employee_data['Age'].describe())
count    1000.000000
mean       42.205000
std        10.016452
min        25.000000
25%        33.000000
50%        43.000000
75%        51.000000
max        59.000000
Name: Age, dtype: float64
```

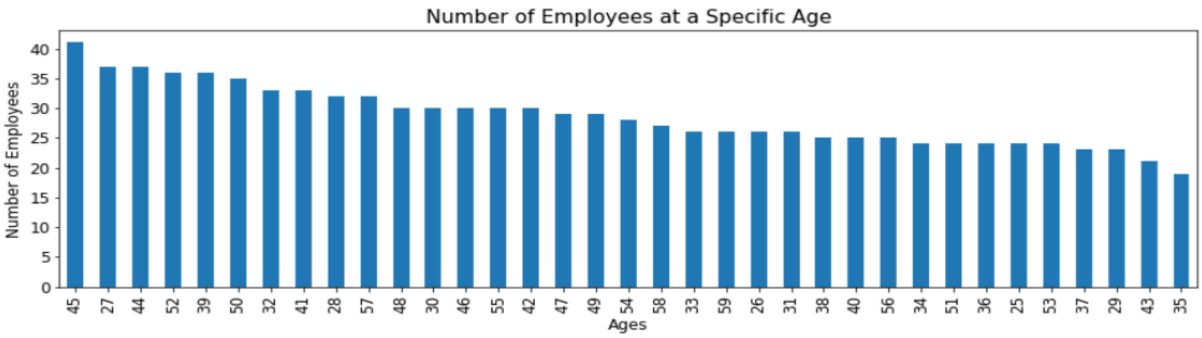
The histogram below shows that the most of the employees are in the age range of **44** to **49** years.

```
In [19]: employee_data.hist(column = 'Age', grid = False, figsize = (6,4), edgecolor = 'black',
    ...:                   bins = 7)
    ...: plt.xlabel('Age', fontsize = 12)
    ...: plt.ylabel('Number of Employees', fontsize = 12)
    ...: plt.title("Distribution of Employees' Ages")
    ...: plt.show()
```



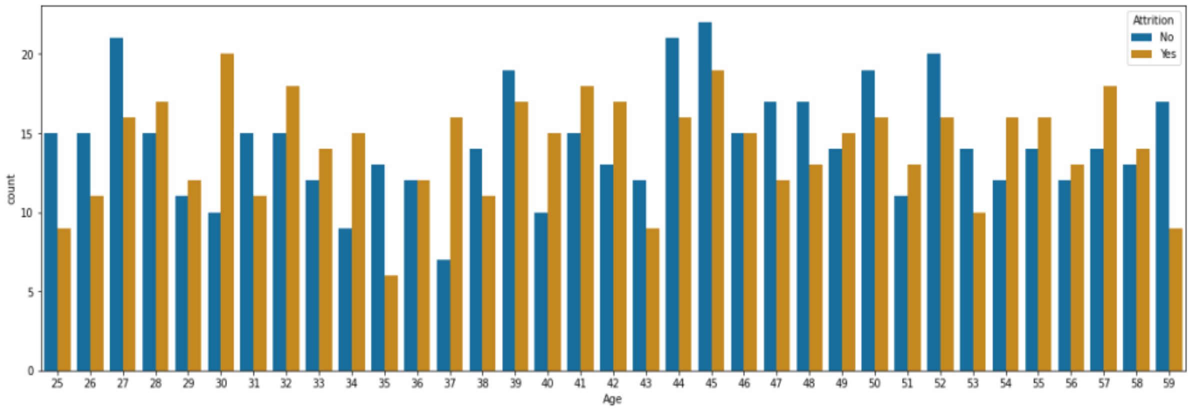
The bar chart below shows the number of employees at a specific age.

```
In [23]: employee_data['Age'].value_counts().plot(kind = 'bar', figsize =(15,4), fontsize = 12)
    ...: plt.xlabel('Ages', fontsize = 12)
    ...: plt.ylabel('Number of Employees', fontsize = 12)
    ...: plt.title('Number of Employees at a Specific Age', fontsize = 15)
    ...: plt.show()
```



The employees with the highest levels of attrition were those aged **30**. The bar chart below shows that there wasn't any one age group which had an outright higher attrition. As such, it was established that the age of an employee didn't play a pivotal role in influencing attrition.

```
In [27]: plt.subplots(figsize = (20,6))
    ...: sns.countplot(x = 'Age', hue = 'Attrition', data = employee_data, palette = 'colorblind')
    ...: plt.show()
```



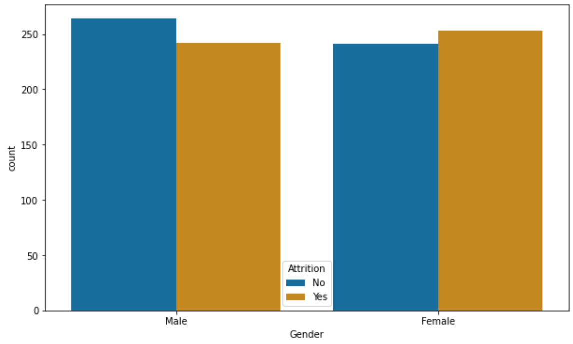
The number of males and females in the dataset was **509** and **494** respectively.

```
In [18]: print(employee_data['Gender'].value_counts())
Gender
Male      506
Female    494
Name: count, dtype: int64
```

The number of male and female attritions stood at **242** and **253** respectively.

```
In [19]: mal_vs_fem_tabl = pd.crosstab(employee_data.Gender, columns = employee_data.Attrition)
    ...: print(mal_vs_fem_tabl)
Attrition   No  Yes
Gender
Female     241  253
Male       264  242
```

```
In [23]: plt.subplots(figsize = (10, 6))
    ...: sns.countplot(x = 'Gender', hue = 'Attrition', data = employee_data, palette = 'colorblind')
    ...: plt.show()
```



Question 4

What are the key indicators or drivers of employee attrition? (e.g. job satisfaction, average monthly hours, salary, promotions, job title, etc.)

Part A: Job Satisfaction

The job satisfaction level in the dataset was measured on a scale of **0** to **1**. Basically, higher values indicated that an employee was satisfied with their job and vice – versa.

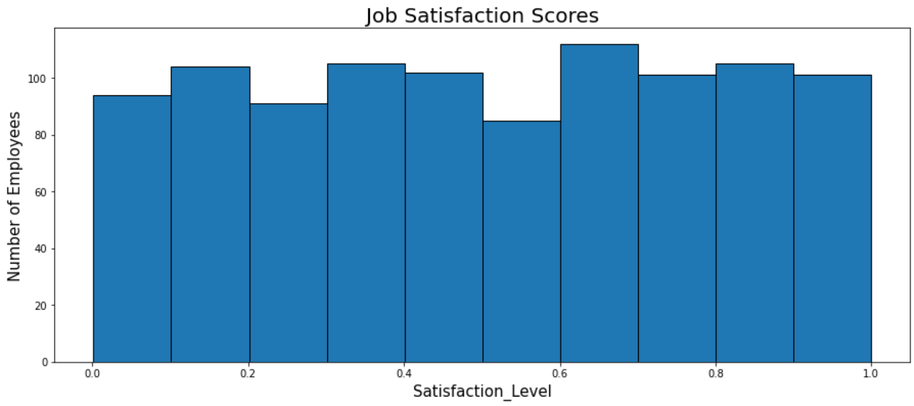
The following information was derived from the screenshot below;

- The minimum satisfaction score was **0.00**.
- The mean satisfaction score was **0.51**.
- The highest satisfaction score was **0.99**.
- 25% (1st quartile) of the employees gave a satisfaction score less than or equal to **0.26**.
- 50% (median – 2nd quartile) of the employees gave a satisfaction score less than or equal to **0.51**.
- 75% (3rd quartile) of the employees gave a satisfaction score less than or equal to **0.76**.

```
In [24]: print(employee_data['Satisfaction_Level'].describe())
count    1000.000000
mean        0.505995
std         0.289797
min         0.001376
25%         0.258866
50%         0.505675
75%         0.761135
max         0.999979
Name: Satisfaction_Level, dtype: float64
```

The histogram below shows that the most of the employees gave a satisfaction score ranging from **0.6** to **0.7**.

```
In [27]: employee_data.hist(column = 'Satisfaction_Level', grid = False, figsize = (15, 6),
    ...:                    edgecolor = 'black', bins = 10)
    ...: plt.xlabel('Satisfaction_Level', fontsize = 15)
    ...: plt.ylabel('Number of Employees', fontsize = 15)
    ...: plt.title('Job Satisfaction Scores', fontsize = 20)
    ...: plt.show()
```



The job satisfaction score values had to be divided into four groups based on the rating given by an employee. This was done to make the comparison process less complicated as there were a total of **1000** unique ratings in the dataset;

```
In [62]: print(employee_data['Satisfaction_Level'].nunique())
1000
```

The four groups were as follows;

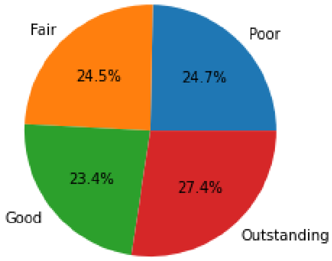
- 0.00 to 0.25 - Poor
- 0.26 to 0.50 - Fair
- 0.51 to 0.75 - Good
- 0.76 to 1.00 - Outstanding

```
In [52]: job_rating = []
    ...: for rating in employee_data['Satisfaction_Level']:
    ...:     if rating > 0.00 and rating <= 0.25:
    ...:         job_rating.append('Poor')
    ...:     elif rating >= 0.26 and rating <= 0.50:
    ...:         job_rating.append('Fair')
    ...:     elif rating >= 0.51 and rating <= 0.75:
    ...:         job_rating.append('Good')
    ...:     else:
    ...:         job_rating.append('Outstanding')
    ...: print(job_rating)
['Good', 'Fair', 'Fair', 'Fair', 'Good', 'Outstanding', 'Good', 'Outstanding', 'Poor', 'Fair',
'Fair', 'Poor', 'Poor', 'Poor', 'Outstanding', 'Good', 'Fair', 'Good', 'Outstanding', 'Good',
```

The job satisfaction level with the highest score was the '**Outstanding'** rating. A total of **274** employees in the dataset gave this rating.

```
In [53]: print('Poor:', '\t\t', job_rating.count('Poor'))
    ...: print('Fair:', '\t\t', job_rating.count('Fair'))
    ...: print('Good:', '\t\t', job_rating.count('Good'))
    ...: print('Outstanding:', job_rating.count('Outstanding'))
Poor:         247
Fair:         245
Good:         234
Outstanding:  274
```

The pie chart below shows a diagrammatic break down of the job satisfaction ratings.



Surprisingly, employees who gave a job satisfaction rating of '**Outstanding**' had the highest number of attritions, amounting to **132**.

```
In [61]: job_rating_tab = pd.crosstab(job_rating, columns = employee_data.Attrition)
    ...: print(job_rating_tab)
Attrition     No  Yes
row_0
Fair         126  119
Good         117  117
Outstanding  142  132
Poor         120  127
```

Part B: Average Monthly Hours

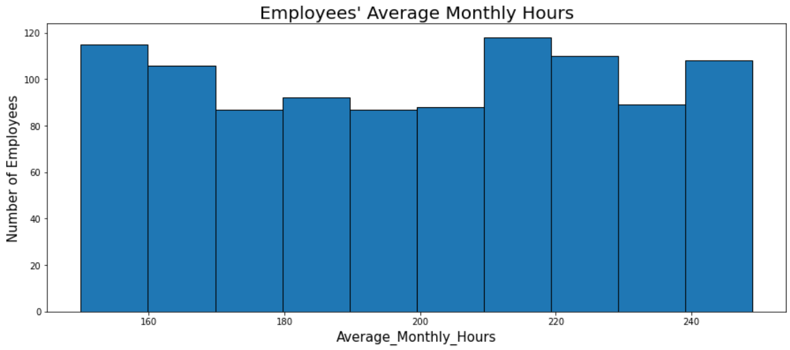
The following information was derived from the screenshot below.

- The lowest monthly hours were **150** hours.
- The mean monthly hours were **199** hours.
- The highest monthly hours were **249** hours.
- 25% (1st quartile) of the employees spent less than or equal to **173** hours at work per month.
- 50% (median – 2nd quartile) of the employees spent less than or equal to **201** hours at work per month.
- 75% (3rd quartile) of the employees spent less than or equal to **225** hours at work per month.

```
print(employee_data['Average_Monthly_Hours'].describe())
count    1000.000000
mean      199.493000
std        29.631908
min       150.000000
25%       173.000000
50%       201.000000
75%       225.000000
max       249.000000
Name: Average_Monthly_Hours, dtype: float64
```

The histogram below shows the employees' average monthly hours.

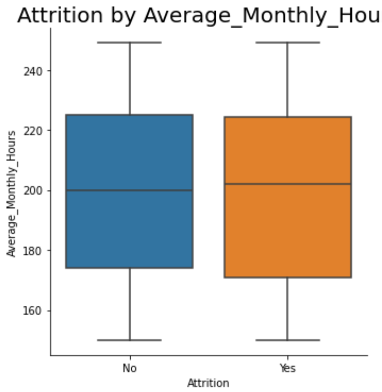
```
In [11]: employee_data.hist(column = 'Average_Monthly_Hours', grid = False, figsize = (15, 6),
    ...:                    edgecolor = 'black', bins = 10)
    ...: plt.xlabel('Average_Monthly_Hours', fontsize = 15)
    ...: plt.ylabel('Number of Employees', fontsize = 15)
    ...: plt.title("Employees' Average Monthly Hours", fontsize = 20)
    ...: plt.show()
```



The Box plot below shows the number of attritions in relation to the average monthly hours. Attritions were comprised of employees who worked for **175** to **225** hours per month. However, the number of hours for the employees who did not leave the organisation was between **177** to **230** hours per month. This clearly showed that the monthly hours spent at work did not necessarily influence employees to leave the organisation.

One factor which could have contributed to employee attritions, was that employees who left had an average of approximately **204** monthly hours. On the other hand, employees who not left the organisation had an average of approximately **200** monthly hours.

```
In [15]: sns.catplot(x = 'Attrition', y = 'Average_Monthly_Hours', kind = 'box', data = employee_data)
    ...: plt.title('Attrition by Average_Monthly_Hours', fontsize = 20)
    ...: plt.show()
```



Part C: Salary

The following information relating to employees' salaries can be derived from the screenshot below.

- The lowest employee salary was **30,009**.
- The mean employee salary was **64,624**.
- The highest employee salary was **99,991**.
- 25% (1st quartile) of the employees' salaries were less than or equal to **47,615**.
- 50% (median – 2nd quartile) of the employees' salaries were less than or equal **64,525**.
- 75% (3rd quartile) of the employees' salaries were less than or equal **81,921**.

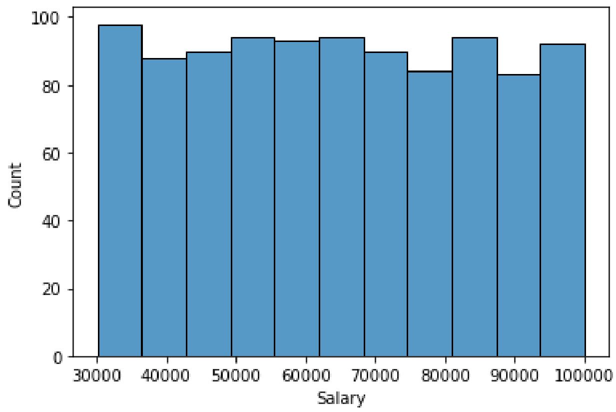
```
In [16]: print(employee_data['Salary'].describe())
count     1000.000000
mean     64624.980000
std      20262.984333
min      30099.000000
25%      47613.500000
50%      64525.000000
75%      81921.000000
max      99991.000000
Name: Salary, dtype: float64
```

There were **995** distinct salary values in the dataset.

```
In [17]: print(employee_data['Salary'].nunique())
995
```

The histplot chart below showed that the salary range of **30,000** to **36,000** had a higher count.

```
sns.histplot(employee_data['Salary'])
plt.show()
```



The box plot chart below showed that an employee's salary did not significantly influence attritions in the dataset. This was because employees who left and those who stayed, fell in more or less a similar salary range of **48,000** to **82,000** (1st to 3rd quartile).

However, one aspect which could have contributed to attritions with regards to salary, was the average salary. The average (mean) salary for employees who left was approximately **63,000**. On the other hand, the average (mean) salary for employees who stayed stood at approximately **66,000**.

```
sns.catplot(x = 'Attrition', y = 'Salary', kind = 'box', data = employee_data)
plt.title('Attrition in Relation to Employee Salary')
plt.show()
```

Part D: Promotion in the last Five (5) Years

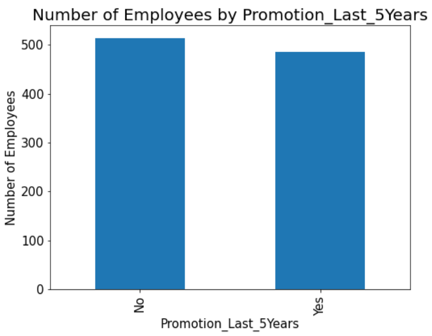
In the 'Promotion_Last_5Years' column of the dataset, the number **one** (**1**) showed that an employee had been promoted. In contrast, the number **zero** (**0**) meant that an employee had not been promoted. These two values 0 and 1, had to converted into string (text) data. Meaning that, 0 and 1 were replaced with '**No**' and '**Yes**' respectively. This was done to improve the clarity and comprehension of the data.

```
print(employee_data['Promotion_Last_5Years'].replace({0: 'No', 1: 'Yes'}, inplace = True))
print(employee_data['Promotion_Last_5Years'].head())
print(employee_data['Promotion_Last_5Years'].tail())
```

```
0      No
1     Yes
2      No
3      No
4     Yes
Name: Promotion_Last_5Years, dtype: object
995     No
996     No
997     No
998     No
999     No
Name: Promotion_Last_5Years, dtype: object
```

The bar chart below shows the number of employees who have been promoted and those who have not. The number of employees who haven't been promoted is more that those who have been promoted.

```
In [36]: employee_data['Promotion_Last_5Years'].value_counts().plot(kind = 'bar', figsize = (8, 6), fontsize = 15)
    ...: plt.xlabel('Promotion_Last_5Years', fontsize = 15)
    ...: plt.ylabel('Number of Employees', fontsize = 15)
    ...: plt.title('Number of Employees by Promotion_Last_5Years', fontsize = 20)
    ...: plt.show()
```



Among the 1000 employees in the dataset, **486** employees were promoted where as, **514** employees were not promoted in the last five years.

```
In [37]: print(employee_data['Promotion_Last_5Years'].value_counts())
Promotion_Last_5Years
No     514
Yes    486
Name: count, dtype: int64
```

Based on the screenshot below, attritions stood at **245** and **250** for employees who were promoted and those who were not promoted respectively. This implies that promotion in the last five (5) years influenced the attritions in the dataset. However, it was a minor factor as the difference between employees who left and those who stayed was only **five** (**5**).

```
In [39]: promo_tab = pd.crosstab(employee_data.Promotion_Last_5Years, columns = employee_data.Attrition)
    ...: print(promo_tab)
Attrition               No  Yes
Promotion_Last_5Years
No                     264  250
Yes                    241  245
```

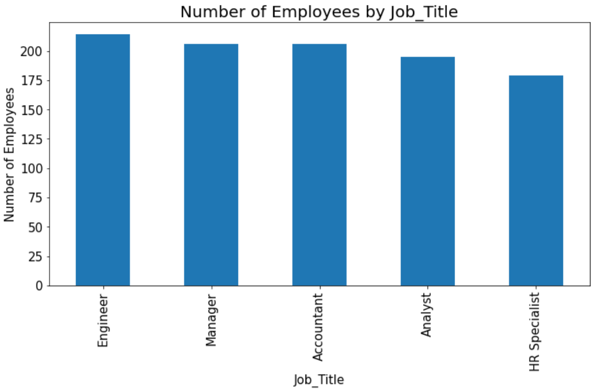
Part E: Job Title

The screenshot below showed that there were more engineers in the dataset as compared to the other professionals.

```
In [41]: print(employee_data['Job_Title'].value_counts())
Job_Title
Engineer        214
Manager         206
Accountant      206
Analyst         195
HR Specialist   179
Name: count, dtype: int64
```

Below is a bar chart showing the composition of the Job Title column in the dataset.

```
In [42]: employee_data['Job_Title'].value_counts().plot(kind = 'bar', figsize = (12, 6), fontsize = 15)
    ...: plt.xlabel('Job_Title', fontsize = 15)
    ...: plt.ylabel('Number of Employees', fontsize = 15)
    ...: plt.title('Number of Employees by Job_Title', fontsize = 20)
    ...: plt.show()
```

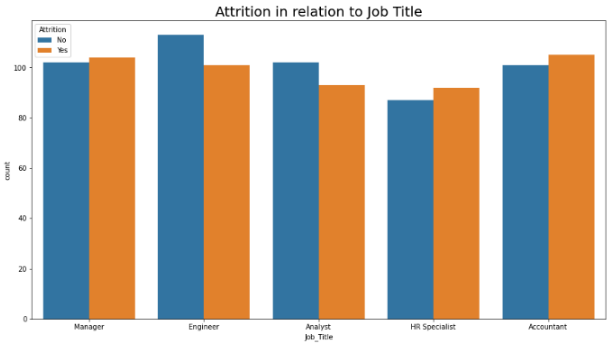


The professionals who had the highest number of attritions were Accountants. The number of attritions stood at **105. Accountants**, **Managers** and **HR Specialists** had attritions which were more than half of the total number of employees in their respective professions. With this in mind, the job title played a role in the number of attritions.

```
In [43]: job_tab = pd.crosstab(employee_data.Job_Title, columns = employee_data.Attrition)
    ...: print(job_tab)
Attrition       No  Yes
Job_Title
Accountant     101  105
Analyst        102   93
Engineer       113  101
HR Specialist   87   92
Manager        102  104
```

Below is a count plot chart showing the attrition levels in relation to job titles.

```
In [49]: plt.subplots(figsize = (15, 8))
    ...: sns.countplot(x = 'Job_Title', hue = 'Attrition', data = employee_data)
    ...: plt.title('Attrition in relation to Job Title', fontsize = 20)
    ...: plt.show()
```

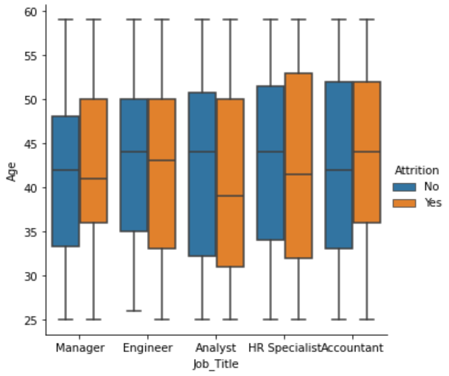


Job Title and Age in Relation to Attrition

The following information was derived from the box plot below;

- **Managers**: Attritions, in relation to age, occurred in an interquartile range of **36** to **50** with a median of **41**.
- **Engineers**: Attritions, in relation to age, occurred in an interquartile range of **33** to **50** with a median of **43**.
- **Analysts**: Attritions, in relation to age, occurred in an interquartile range of **31** to **50** with a median of **38**.
- **HR Specialists**: Attritions, in relation to age, occurred in an interquartile range of **32** to **53** with a median of **42**.
- **Accountants**: Attritions, in relation to age, occurred in an interquartile range of **37** to **52** with a median of **45**.

```
sns.catplot(x = 'Job_Title', y = 'Age', hue = 'Attrition', kind = 'box', data = employee_data)
plt.show()
```

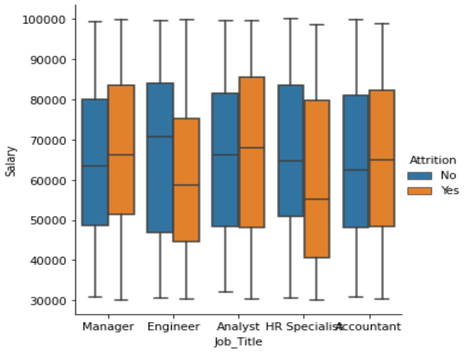


Job Title and Salary in Relation to Attrition

The following information was derived from the box plot below;

- **Managers**: Attritions, in relation to salary, occurred in an interquartile range of **52,000** to **84,000** with a median of **67,000**.
- **Engineers**: Attritions, in relation to salary, occurred in an interquartile range of **45,000** to **75,000** with a median of **60,000**.
- **Analysts**: Attritions, in relation to salary, occurred in an interquartile range of **49,000** to **86,000** with a median of **68,000**.
- **HR Specialists**: Attritions, in relation to salary, occurred in an interquartile range of **42,000** and **80,000** with a median of **55,000**.
- **Managers**: Attritions, in relation to salary, occurred in an interquartile range of **49,000** to **82,000** with a median of **65,000**.

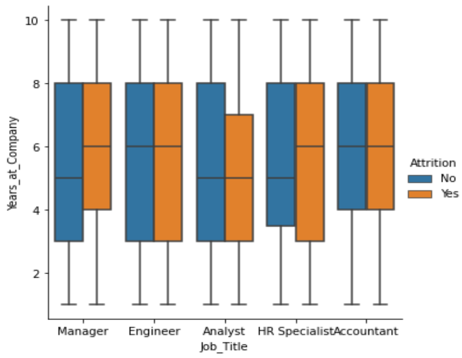
```
sns.catplot(x = 'Job_Title', y = 'Salary', hue = 'Attrition', kind = 'box', data = employee_data)
plt.show()
```



Job Title and Years at Company in Relation to Attrition
The following information was derived from the box plot below;

- **Managers**: Attritions, in relation to years at company, occurred in an interquartile range of **4** to **8** with a median of **5**.
- **Engineers**: Attritions, in relation to years at company, occurred in an interquartile range of **3** to **8** with a median of **6**.
- **Analysts**: Attritions, in relation to years at company, occurred in an interquartile range of **3** to **7** with a median of **5**.
- **HR Specialists**: Attritions, in relation to years at company, occurred in an interquartile range of **3** to **8** with a median of **6**.
- **Accountants**: Attritions, in relation to years at company, occurred in an interquartile range of **4** to **8** with a median of **6**.

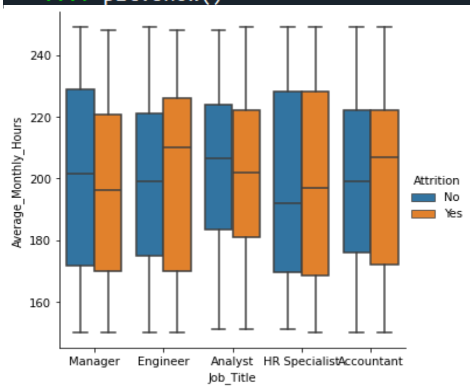
```
In [9]: sns.catplot(x = 'Job_Title', y = 'Years_at_Company', hue = 'Attrition', kind = 'box', data =
employee_data)
   ...: plt.show()
```



Job Title and Average Monthly Hours in Relation to Attrition
The following information was derived from the box plot below;

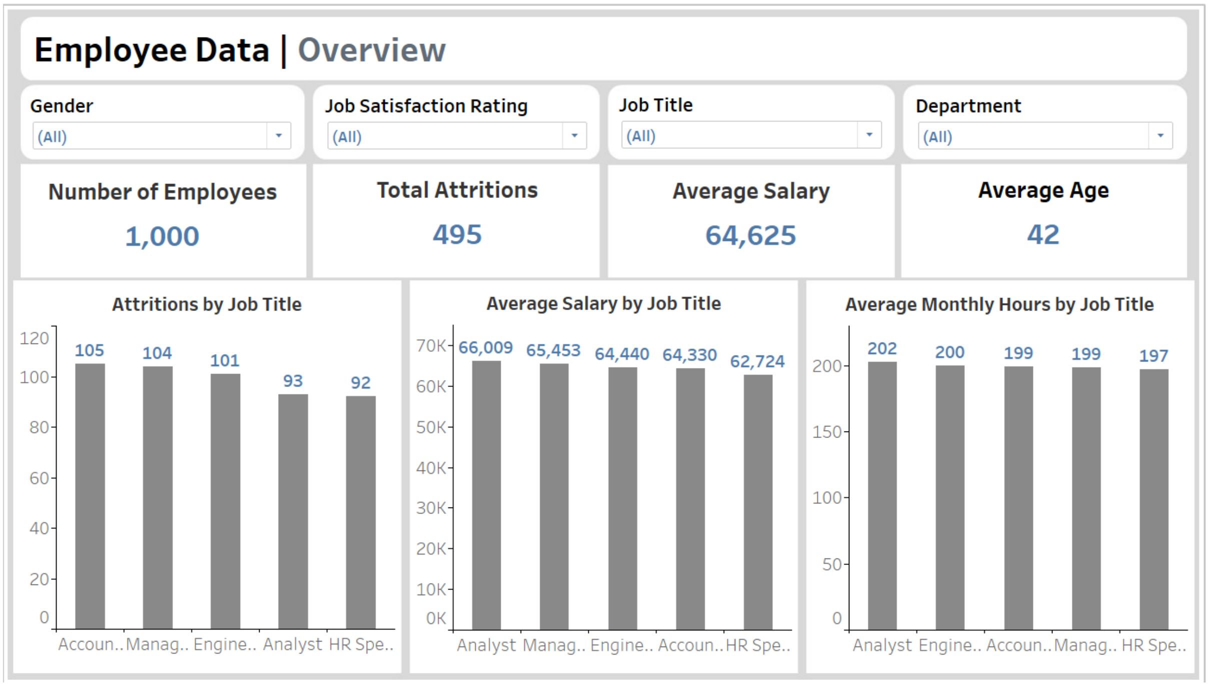
- **Managers**: Attritions, in relation to years at company, occurred in an interquartile range of **170** to **220** with a median of **195**.
- **Engineers**: Attritions, in relation to years at company, occurred in an interquartile range of **170** to **228** with a median of **210**.
- **Analysts**: Attritions, in relation to years at company, occurred in an interquartile range of **185** to **223** with a median of **203**.
- **HR Specialists**: Attritions, in relation to years at company, occurred in an interquartile range of **168** to **228** with a median of **195**.
- **Accountants**: Attritions, in relation to years at company, occurred in an interquartile range of **173** to **222** with a median of **207**.

```
In [2]: sns.catplot(x = 'Job_Title', y = 'Average_Monthly_Hours', hue = 'Attrition', kind = 'box',
data = employee_data)
   ...: plt.show()
```



Part 3: Dashboard Visualisations

The dashboard visualisations for this project were created using Tableau. Below is a screenshot of the dashboard. The dashboard can be accessed using this link.



Part 4: Summary of findings after performing analysis

The following are some of the standout aspects discovered during the analysis.

- The total number of employees in the dataset was **1000**. The number of attritions stood at **495** in contrast to **505** employees who did not leave the organisation.
- From the total employee workforce of 1000, the breakdown according to gender was as follows, Female – **494** and Male – **506**.
- The age of the employees ranged from **25** to **59**. The average age of the employees was 42. With regards to attrition, employees aged **39** had the highest number of attritions.
- The employees' annual salaries ranged from **30,009** to **99,991**. Employees with the highest attritions had a mean salary of **63,000**.
- The tenure of employment ranged from **1** year to **10** years. Employees who had served for **7** years had the highest number of attritions.
- The dataset had five job titles, among these five, **Accountants** had their highest number of attritions which stood at **105**.
- The dataset was comprised of five departments, among these five, the **Engineering** had the highest number of attritions at **109**.
- The average monthly work hours ranged from **150** to **249** hours. Employees who left the organisation had an average workload of **204** monthly hours.
- Among the 1000 employees in the dataset, **486** employees were promoted where as, **514** employees were not promoted. Attritions stood at **245** and **250** for employees who were promoted and those who were not promoted respectively.

- During the analysis, the job satisfaction rating given by each employee was classified in the following manner for more clarity.
    - 0.00 to 0.25 - Poor
    - 0.26 to 0.50 - Fair
    - 0.51 to 0.75 - Good
    - 0.76 to 1.00 - Outstanding

  Surprising, employees who gave a rating of '**Outstanding'** had the highest number of attritions at **132**.

Part 5: Recommendations for stakeholders

The following are some recommendations to reduce employee attrition.

- Encourage career growth: Provide career advancement opportunities, training programs, and mentorship to support employee progression to higher job levels.
- Offer competitive compensation: Offer competitive salaries and benefits that align with the market standards and recognize and reward long-serving employees for their commitment to the organization.
- Foster a positive work environment: Provide a positive and inclusive work environment that encourages employee engagement and job satisfaction.
- Gather employee feedback: Conduct regular employee engagement surveys to understand the underlying reasons for employee turnover and take corrective actions accordingly.
- Focusing on well-being and mental health: Employers should provide support to employees during personal emergencies and give them the flexibility to focus on their mental health.

The data analysis was done by:
Fabiano Chela
chelafabiano@gmail.com
GitHub link
LinkedIn link
0978411822