fv_si618finalprojet

December 12, 2018

```
In [1]: MY_UNIQNAME = 'lopezy'
```

1 SI618 Fall 2018 Final Project

Date: December 12, 2018
Name: Yuko Lopez (lopezy)

1.0.1 *Title: "Factors That May Lead To Job Attrition"*

---

2 The Structure of The Report:

The Table of Contents listed in the following section provides the general structure of this repot. Motivation, Exploratory Questions, and Data Source are fully addressed before the report shows any code and/or visualizations. For Methodologies, Analysis, and Results, are found in detail with the main content of this report together with accomopanying code and visualizations. The section of conclusion addresses the summary conclusions before the main content, where coding/visualization begins.

3 Table of Contents:

---

4 Introduction:

4.1 Motivation - The Nature of the project:

Unlike layoffs, the job attritions are voluntary means for leaving the companies such as volutary resignations and retirements.

The voluntary departures -(i.e.) job attrition- can stem from various causes (logistics such as too long commute and/or family circumstances, mismatch agaisnt personal career goals, economics/financial etc) to organizational (compmany structure, management changes, work environments, etc).

From employees' perspectives, departing from their current companies is likely to result in the loss of income unless they have next jobs lined up. From companies' perspectives, too, the attrition can be costly due the the times and the associated costs for advertisement and manpower to screen candidates.

To complicate the matter, we live in a culture, in which the female workforce are at a greater risk of lower wages compared to male counterparts. Even outside the workplace, female population tend to be the ones to bare more household obligations whether it is clearning, cooking, or child rearing. No matter what the cause, job attrtion can have many negative consequences for both employees and emploers. Yet, this is a fact of working professionals that many of us face.

This project aims to explore and hope to find some potential causes and their trends/patterns for this costly yet incredibly common phenomenon.

4.2 Exploratory Questions:

(1) "How does *income* affect job attrition?": Not many people can genuinely say that the money does not matter at all in life. After all, to live is to cost. The ability to finance the lives of ourselves and our loved ones can have an effect in retirement, education, etc. In addition, the income within the workplace is supposed to be the reflection of the employees' values.

Further, job hopping is one of the ways in which how employees can have an increase in income aside from natural career progressions within an organization.

(2) "Can the level of *responsibilities/positions* affect job attrition?": Based on my own professional experiences, the right amount of challenges and responsibilities based on my capabilities at any given times were one of the key factors whether or not I found my jobs to be sataisfying. The nature of the jobs -(e.g.) Some may find leading and managing others to be their callings while some others may prefer to remain as individual contributors- is also critical to individual career success. Or, even if the positions are good fit, if the types of assingments and projects are not challenging enough, the jobs may soon become repetitive, mundane, or even boring.

(3) "Does the *long hours of work engagement* result in job attrition?": One of the few equalities that the life gives us all is that we all have 24 hours a day, no exceptions.

And it is rarely the case that anybody who commute for a full-time work needs to allocate more than eight hours a day as we must take the communiting time into considerations. In addition, some may have frequent over-time, and/or buisness trips.

(4) Is there a *gender and/or age difference* in job attrition? Analyzing potential gender and age differences in any areas of study are one of the basic yet powerful ways to engage with data, and this dataset is no exception.

---

5 Data Source:

5.0.1 Data Set:

Data Source URL: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

About the dataset:

```
(1) The dataset consists of 1470 datapoints with no missing values.
(2) The dataset has 35 columns in total with respecitve pandas' datatypes as desribed below:
    - Age                         (int64)
    - 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)
    - MonthlyRate                 (int64)
    - NumCompaniesWorked          (int64)
    - Over18                      (object)
    - 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        (int64)


(3) The above columns can be further categorized as follows:
(a) Categorical Values (21 columns):
    - Attrition                   object ('Yes'/'No')
    - BusinessTravel              object ('Travel_Rarely', 'Travel_Frequently', 'Non-Travel')
    - Department                  object ('Sales', 'Research & Development', 'Human Resources')
    - Education                   int64 (1:'Below College', 2:'College', 3:'Bachelor',  4:'Mas
    - EducationField              object ('Life Sciences', 'Other', 'Medical', 'Marketing', 'T
                                            'Human Resources')
    - EmployeeCount               int64 (1) (*)
    - EmployeeNumber              int64 (1 ,    2,    4, ..., 2064, 2065, 2068) (*) This colum
    - EnvironmentSatisfaction     int64 (1:'Low', 2:'Medium', 3:'High', 4:'Very High')
    - Gender                      object ('Female', 'Male')
    - JobInvolvement              int64 (1: 'Low', 2:'Medium', 3:'High' 4:'Very High')
    - JobLevel                    int64 (1, 2, 3, 4, 5)
    - JobRole                     object ('Sales Executive', 'Research Scientist', 'Laboratory
                                            'Manufacturing Director', 'Healthcare Representati
                                            'Sales Representative', 'Research Director', 'Huma
    - JobSatisfaction             int64 (1:'Low', 2:'Medium', 3:'High', 4:'Very High')
    - MaritalStatus               object ('Single', 'Married', 'Divorced')
    - Over18                      object ('Y') (*)
    - OverTime                    object ('Yes', 'No')
    - PerformanceRating           int64 (1:'Low', 2:'Good', 3:'Excellent', 4:'Outstanding')
    - RelationshipSatisfaction    int64 (1:'Low', 2:'Medium', 3:'High', 4:'Very High')
    - StandardHours               int64 (80) (*)
    - StockOptionLevel            int64 (1, 2, 3, 4)
    - WorkLifeBalance             int64 (1:'Bad', 2:'Good', 3:'Better' 4:'Best')


(b) Non-Categorical (numeric) Values (14 columns):
    - Age                         (int64) (18... 60)
    - DailyRate                   (int64) (
    - DistanceFromHome            (int64)
    - HourlyRate                  (int64)
    - MonthlyIncome               (int64)
    - MonthlyRate                 (int64)
    - NumCompaniesWorked          (int64)
    - PercentSalaryHike           (int64)
    - TotalWorkingYears           int64 (0... 40)
    - TrainingTimesLastYear       int64 (0, 1, 2, 3, 4, 5, 6)
    - YearsAtCompany              int64 (0... 40)
    - YearsInCurrentRole          int64 (0... 18)
    - YearsSinceLastPromotion     int64 (0... 15)
    - YearsWithCurrManager        int64 (0... 17)
```

```
(4) (*) Columns excluded for the analysis: "EmployeeCount" with the value, 1, for all the datap
"EmployeeNumber", the IDs, which will be converted into Index.  "StandardHours", all of which a

(5) There is no time series/element in this dataset.
```

---

6 Methodologies, Data Manipulation, Workflow, and Challenges:

The following elements are addressed within the main contents (code and visualizations) either directly above or below the code as markdowns:
(a) Data manipulation
(b) Workflow
(c) Challenges

The analysis and interpretations mentnioned below are also addressed in applicable markdowns together with the main contentes:
(a) Summary of interesting observations and results, relationship or insight, if any, that are found.
(b) Negative results, where I failed to answer the questions.
(c) Visualizations

6.1 Income and Attrition:

```
(1) Data manipulation and visualization
(2) Hypothesizing the observations:
    H0: Income has no affect in attritioin.
    H1: Income does have an effect in attrtion (H0 is not true)
(3) Hpothesis testing (T-test)
(4) Results, observations, and interpretations
```

6.2 Position level and attrition:

```
(1) Data manipulation and visualization
(2) Hypothesizing the observations:
    H0: PositonLevel/ JobPositions has no affect in attritioin.
    H1: PositionLevel/ JobPosition does have an effect in attrtion (H0 is not true)
(3) Hpothesis testing (T-test, or one-way ANOVA)
(4) Results, observations, and interpretations
```

6.3 Hours of Engagement for Work:

```
(1) Data manipulation and visualization
(2) Hypothesizing the observations:
    H0: Long hours of work has no affect in attritioin.
    H1: Long hours of work does have an effect in attrtion (H0 is not true)
```

```
(3) Hpothesis testing (T-test, or one-way ANOVA)
(4) Results, observations, and interpretations
```

6.4 Gender and Age Difference in Attrition:

```
(1) Data manipulation and visualization
(2) Hypothesizing the observations:
    * Gender:
    H0: Gender has no affect in attritioin.
    H1: Gender does have an effect in attrtion (H0 is not true)

    * Age:
    H0: Age has no affect in attrition.
    H1: Age does have an effect in attritio (H) is not true).
(3) Hpothesis testing (T-test, or one-way ANOVA)
(4) Results, observations, and interpretations
```

6.5 Investigating Numeric Data Correlation:

The summarized result is that there are multiple highly correlated columns are found:

(1) JobLevel and MonthlyIncome (0.95)

(2) JobLevel and TotalWorkingYears (0.78)

(3) MonthlyIncome and TotalWorkingYears (0.77)

(4) YearsAtCompany and YearsWithCurrManager (0.77)

(5) YearsAtCompany and YearsInCurrentRole (0.76)

(6) YearsWithCurrManager and YearsInCurrentRole (0.71)

The same results with a brief discussion are found with the correlationheatmap.

6.6 Investigating Categorical Data Correlation

A similar heatmap was created without an interesting result. This is due to the fact that categorical data is not suited for correlation mapping.

6.7 K-Means Clustering

The K-Means clustering with PCA are performed experimentally to see if there are visually interesting clusters can be made. However, the resulting 3D graph did not show anything apparetly interesting as it can be found later in the paper.

6.8 Random Forest Classifications

The top five most important features are found to be as follows:
(1) 'MonthlyIncome', 0.09620026433543806,
(2) 'Age', 0.07465100132268015,
(3) 'MonthlyRate', 0.0678728429991805,
(4) 'DailyRate', 0.06395539641367105,
(5) 'HourlyRate', 0.060778294100276326,

7 Conclusions:

By the end of the analysis detailed in the main contents below, I can conclude that the income indeed does affect job attrition and appears to be the most important factor when considering the potential causes of job attritions as found in the analysis and its visualizatiaons, hypothesis testingm, and the Random Forest Classification results. Similarly, the job level and roles do affect job attritition. This is likely to be the fact that the job level and roles are relfective of the income. Hours of engagement for work does affect job attritions. If employees have long hours tied to work, whether over time and/or business traveling, they are more likely to have job attrtions. The commuting distances the dataset has -ranging from 1 to 29-, does not seem to affect attritions. Age affects job attrition while a similar results could not be obtained for a potential gender difference in attrition. The lack of evidence for gender and attrition relationship was further confirmed after conducting the Random Forest Classifications. However, there are some visualizations that can be found in the areas of income, highlighting potentially interesting gender differences although they may be statistically insitnificant.

(The Coding/visualization contents starts here)

In [2]:
```
import pandas as pd # linear algebra
import numpy as np  # data processing, CSV file I/O (e.g. pd.read_csv)
import scipy as sp
import sklearn as sk
import seaborn as sns # visualization
import matplotlib.pyplot as plt # for plotting
from statsmodels.graphics.mosaicplot import mosaic
```

```
from matplotlib import colors as mcolors
from pandas import DataFrame, read_html
%matplotlib inline
from scipy import stats # statistical analysis
from scipy.stats import chisquare

sns.set()

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (accuracy_score, log_loss, classification_report)
#from imblearn.over_sampling import SMOTE
#import xgboost

# Import statements required for Plotly
#import plotly.offline as py
#py.init_notebook_mode(connected=True)
#import plotly.graph_objs as go
#import plotly.tools as tls

# Import and suppress warnings
import warnings
warnings.filterwarnings('ignore')


# For clustering:
from scipy.cluster.hierarchy import dendrogram
import scipy.cluster.hierarchy as shc
from scipy.spatial.distance import cdist

from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering


# For classifications:
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
import sklearn.ensemble as skens
import sklearn.metrics as skmetric
import sklearn.naive_bayes as sknb
import sklearn.tree as sktree
import matplotlib.pyplot as plt
```

```
%matplotlib inline
import seaborn as sns
sns.set(style='white', color_codes=True, font_scale=1.3)
import sklearn.externals.six as sksix
import IPython.display as ipd
from sklearn.model_selection import cross_val_score
from sklearn import metrics
import os

from sklearn.model_selection import GridSearchCV
from sklearn.naive_bayes import GaussianNB

#Do I need these (for decision trees):
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus

# K-Means
from sklearn.cluster import KMeans
import sklearn as sk
from sklearn import metrics
from scipy.spatial.distance import cdist
import numpy as np
import matplotlib.pyplot as plt


# Dendrogram
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt


from sklearn.cluster import AgglomerativeClustering

from sklearn.model_selection import train_test_split

import sklearn.decomposition as skd
import sklearn.preprocessing as skp
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.mplot3d import proj3d
```

```
In [3]: attrition_data = pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
```

```
In [4]: # Convert EmployeeNumber colum to DataFrame index:
        attrition_data.set_index('EmployeeNumber', inplace=True)
```

```
In [5]: attrition_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1470 entries, 1 to 2068
```

```
Data columns (total 34 columns):
Age                         1470 non-null int64
Attrition                   1470 non-null object
BusinessTravel              1470 non-null object
DailyRate                   1470 non-null int64
Department                  1470 non-null object
DistanceFromHome            1470 non-null int64
Education                   1470 non-null int64
EducationField              1470 non-null object
EmployeeCount               1470 non-null int64
EnvironmentSatisfaction     1470 non-null int64
Gender                      1470 non-null object
HourlyRate                  1470 non-null int64
JobInvolvement              1470 non-null int64
JobLevel                    1470 non-null int64
JobRole                     1470 non-null object
JobSatisfaction             1470 non-null int64
MaritalStatus               1470 non-null object
MonthlyIncome               1470 non-null int64
MonthlyRate                 1470 non-null int64
NumCompaniesWorked          1470 non-null int64
Over18                      1470 non-null object
OverTime                    1470 non-null object
PercentSalaryHike           1470 non-null int64
PerformanceRating           1470 non-null int64
RelationshipSatisfaction    1470 non-null int64
StandardHours               1470 non-null int64
StockOptionLevel            1470 non-null int64
TotalWorkingYears           1470 non-null int64
TrainingTimesLastYear       1470 non-null int64
WorkLifeBalance             1470 non-null int64
YearsAtCompany              1470 non-null int64
YearsInCurrentRole          1470 non-null int64
YearsSinceLastPromotion     1470 non-null int64
YearsWithCurrManager        1470 non-null int64
dtypes: int64(25), object(9)
memory usage: 402.0+ KB
```

```
In [6]: attrition_data.head()
```

```
Out[6]:                 Age Attrition     BusinessTravel  DailyRate  \
        EmployeeNumber
        1                41       Yes      Travel_Rarely       1102
        2                49        No  Travel_Frequently        279
        4                37       Yes      Travel_Rarely       1373
        5                33        No  Travel_Frequently       1392
        7                27        No      Travel_Rarely        591
```

```
                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
2               Research & Development                 8          1
4               Research & Development                 2          2
5               Research & Development                 3          4
7               Research & Development                 2          1

               EducationField  EmployeeCount  EnvironmentSatisfaction  \
EmployeeNumber
1               Life Sciences              1                        2
2               Life Sciences              1                        3
4                       Other              1                        4
5               Life Sciences              1                        4
7                     Medical              1                        1

                        ...         RelationshipSatisfaction  StandardHours  \
EmployeeNumber          ...
1                       ...                                1             80
2                       ...                                4             80
4                       ...                                2             80
5                       ...                                3             80
7                       ...                                4             80

                StockOptionLevel  TotalWorkingYears TrainingTimesLastYear  \
EmployeeNumber
1                              0                  8                     0
2                              1                 10                     3
4                              0                  7                     3
5                              0                  8                     3
7                              1                  6                     3

                WorkLifeBalance YearsAtCompany  YearsInCurrentRole  \
EmployeeNumber
1                             1              6                   4
2                             3             10                   7
4                             3              0                   0
5                             3              8                   7
7                             3              2                   2

                YearsSinceLastPromotion  YearsWithCurrManager
EmployeeNumber
1                                     0                     5
2                                     1                     7
4                                     0                     0
5                                     3                     0
7                                     2                     2
```

```
[5 rows x 34 columns]
```

In [7]:
```
# making sure no spaces before/after each column names.
attrition_data.columns
```

Out[7]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [8]:
```
# checking to see if missing values exist:
attrition_data.isnull().sum(axis = 0)
```

Out[8]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
```

```
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [9]:
```
attrition_data.describe()
```

Out[9]:
```
               Age    DailyRate  DistanceFromHome    Education  EmployeeCount  \
count  1470.000000  1470.000000       1470.000000  1470.000000         1470.0
mean     36.923810   802.485714          9.192517     2.912925            1.0
std       9.135373   403.509100          8.106864     1.024165            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   802.000000          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

       EnvironmentSatisfaction   HourlyRate  JobInvolvement     JobLevel  \
count              1470.000000  1470.000000     1470.000000  1470.000000
mean                  2.721769    65.891156        2.729932     2.063946
std                   1.093082    20.329428        0.711561     1.106940
min                   1.000000    30.000000        1.000000     1.000000
25%                   2.000000    48.000000        2.000000     1.000000
50%                   3.000000    66.000000        3.000000     2.000000
75%                   4.000000    83.750000        3.000000     3.000000
max                   4.000000   100.000000        4.000000     5.000000

       JobSatisfaction          ...            RelationshipSatisfaction  \
count      1470.000000          ...                         1470.000000
mean          2.728571          ...                            2.712245
std           1.102846          ...                            1.081209
min           1.000000          ...                            1.000000
25%           2.000000          ...                            2.000000
50%           3.000000          ...                            3.000000
75%           4.000000          ...                            4.000000
max           4.000000          ...                            4.000000

       StandardHours  StockOptionLevel  TotalWorkingYears  \
count         1470.0       1470.000000        1470.000000
mean            80.0          0.793878          11.279592
std              0.0          0.852077           7.780782
min             80.0          0.000000           0.000000
25%             80.0          0.000000           6.000000
50%             80.0          1.000000          10.000000
75%             80.0          1.000000          15.000000
```

```
max              80.0          3.000000          40.000000

       TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
count            1470.000000      1470.000000     1470.000000
mean                2.799320         2.761224        7.008163
std                 1.289271         0.706476        6.126525
min                 0.000000         1.000000        0.000000
25%                 2.000000         2.000000        3.000000
50%                 3.000000         3.000000        5.000000
75%                 3.000000         3.000000        9.000000
max                 6.000000         4.000000       40.000000

       YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
count         1470.000000              1470.000000           1470.000000
mean             4.229252                 2.187755              4.123129
std              3.623137                 3.222430              3.568136
min              0.000000                 0.000000              0.000000
25%              2.000000                 0.000000              2.000000
50%              3.000000                 1.000000              3.000000
75%              7.000000                 3.000000              7.000000
max             18.000000                15.000000             17.000000

[8 rows x 25 columns]
```

.unique() method is applied for the columns that appear to be categorical rather than numeric.

```
In [10]: attrition_data['BusinessTravel'].unique()

Out[10]: array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'], dtype=object)

In [11]: attrition_data['Department'].unique()

Out[11]: array(['Sales', 'Research & Development', 'Human Resources'], dtype=object)

In [12]: attrition_data['EducationField'].unique()

Out[12]: array(['Life Sciences', 'Other', 'Medical', 'Marketing',
                'Technical Degree', 'Human Resources'], dtype=object)

In [13]: attrition_data['EducationField'].unique()

Out[13]: array(['Life Sciences', 'Other', 'Medical', 'Marketing',
                'Technical Degree', 'Human Resources'], dtype=object)

In [14]: attrition_data['JobLevel'].unique()

Out[14]: array([2, 1, 3, 4, 5])
```

In [15]: attrition_data['JobRole'].unique()

```
Out[15]: array(['Sales Executive', 'Research Scientist', 'Laboratory Technician',
               'Manufacturing Director', 'Healthcare Representative', 'Manager',
               'Sales Representative', 'Research Director', 'Human Resources'],
              dtype=object)
```

In [16]: attrition_data['MaritalStatus'].unique()

Out[16]: array(['Single', 'Married', 'Divorced'], dtype=object)

In [17]: attrition_data['Over18'].unique()

Out[17]: array(['Y'], dtype=object)

In [18]: attrition_data['OverTime'].unique()

Out[18]: array(['Yes', 'No'], dtype=object)

In [19]: attrition_data['StandardHours'].unique()

Out[19]: array([80])

In [20]: attrition_data['StockOptionLevel'].unique()

Out[20]: array([0, 1, 3, 2])

In [21]: attrition_data['TotalWorkingYears'].unique().min()

Out[21]: 0

In [22]: attrition_data['TotalWorkingYears'].unique().max()

Out[22]: 40

In [23]: attrition_data['TrainingTimesLastYear'].unique()

Out[23]: array([0, 3, 2, 5, 1, 4, 6])

In [24]: attrition_data['YearsAtCompany'].unique().min()

Out[24]: 0

In [25]: attrition_data['YearsAtCompany'].unique().max()

Out[25]: 40

In [26]: attrition_data['YearsInCurrentRole'].unique().min()

Out[26]: 0

In [27]: attrition_data['YearsInCurrentRole'].unique().max()

```
Out[27]: 18

In [28]: attrition_data['YearsSinceLastPromotion'].unique().min()

Out[28]: 0

In [29]: attrition_data['YearsSinceLastPromotion'].unique().max()

Out[29]: 15

In [30]: attrition_data['YearsWithCurrManager'].unique().min()

Out[30]: 0

In [31]: attrition_data['YearsWithCurrManager'].unique().max()

Out[31]: 17
```

As mentioned Data Source, Data Set in 4., some columns are removed from attrtion_data as deemed unnecessary for the analysis. This is to serve the purposes of: (1) using as clean dataset as possible, leaving out the values otherwise serve no purpose, (2) removing the elements that may negatively affect further analysis. For example, the column, "StandardHours", is removed, as all the 1,470 datapoints have 80 hours. The purpose of having a separate DataFrame from attrtion_data is just in case such DataFrame is needed.

```
In [32]: attrition = attrition_data.drop(['EmployeeCount', 'StandardHours', 'Over18'], axis=1)
         attrition.columns

Out[32]: Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
                'DistanceFromHome', 'Education', 'EducationField',
                'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
                'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
                'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime',
                'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
                'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
                'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
                'YearsSinceLastPromotion', 'YearsWithCurrManager'],
               dtype='object')

In [33]: attrition.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1470 entries, 1 to 2068
Data columns (total 31 columns):
Age                         1470 non-null int64
Attrition                   1470 non-null object
BusinessTravel              1470 non-null object
DailyRate                   1470 non-null int64
```

```
Department                  1470 non-null object
DistanceFromHome            1470 non-null int64
Education                   1470 non-null int64
EducationField              1470 non-null object
EnvironmentSatisfaction     1470 non-null int64
Gender                      1470 non-null object
HourlyRate                  1470 non-null int64
JobInvolvement              1470 non-null int64
JobLevel                    1470 non-null int64
JobRole                     1470 non-null object
JobSatisfaction             1470 non-null int64
MaritalStatus               1470 non-null object
MonthlyIncome               1470 non-null int64
MonthlyRate                 1470 non-null int64
NumCompaniesWorked          1470 non-null int64
OverTime                    1470 non-null object
PercentSalaryHike           1470 non-null int64
PerformanceRating           1470 non-null int64
RelationshipSatisfaction    1470 non-null int64
StockOptionLevel            1470 non-null int64
TotalWorkingYears           1470 non-null int64
TrainingTimesLastYear       1470 non-null int64
WorkLifeBalance             1470 non-null int64
YearsAtCompany              1470 non-null int64
YearsInCurrentRole          1470 non-null int64
YearsSinceLastPromotion     1470 non-null int64
YearsWithCurrManager        1470 non-null int64
dtypes: int64(23), object(8)
memory usage: 367.5+ KB
```

```
In [34]: attrition.head()
```

```
Out[34]:                 Age Attrition     BusinessTravel  DailyRate  \
         EmployeeNumber
         1                41       Yes      Travel_Rarely       1102
         2                49        No  Travel_Frequently        279
         4                37       Yes      Travel_Rarely       1373
         5                33        No  Travel_Frequently       1392
         7                27        No      Travel_Rarely        591

                                     Department  DistanceFromHome  Education  \
         EmployeeNumber
         1                                Sales                 1          2
         2               Research & Development                 8          1
         4               Research & Development                 2          2
         5               Research & Development                 3          4
         7               Research & Development                 2          1
```

```
               EducationField  EnvironmentSatisfaction  Gender  \
EmployeeNumber
1               Life Sciences                        2  Female
2               Life Sciences                        3    Male
4                       Other                        4    Male
5               Life Sciences                        4  Female
7                     Medical                        1    Male

                      ...          PerformanceRating  \
EmployeeNumber        ...
1                     ...                          3
2                     ...                          4
4                     ...                          3
5                     ...                          3
7                     ...                          3

                RelationshipSatisfaction  StockOptionLevel TotalWorkingYears  \
EmployeeNumber
1                                      1                 0                 8
2                                      4                 1                10
4                                      2                 0                 7
5                                      3                 0                 8
7                                      4                 1                 6

                TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
EmployeeNumber
1                                   0               1               6
2                                   3               3              10
4                                   3               3               0
5                                   3               3               8
7                                   3               3               2

                YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                                4                        0
2                                7                        1
4                                0                        0
5                                7                        3
7                                2                        2

               YearsWithCurrManager
EmployeeNumber
1                                 5
2                                 7
4                                 0
5                                 0
7                                 2
```

```
[5 rows x 31 columns]
```

The DataFrame, attrition, has four columns removed from the original DataFrame, attrition_data, leaving out 'EmployeeCount', 'EmployeeNumber', 'StandardHours', 'Over18'.

Below, .describe() is applied to the DataFrame, attrition, except that each column is specified. This way, the result of .describe() will show all the contents.

```
In [35]: attrition.describe()
```

```
Out[35]:                Age    DailyRate  DistanceFromHome    Education  \
         count  1470.000000  1470.000000       1470.000000  1470.000000
         mean     36.923810   802.485714          9.192517     2.912925
         std       9.135373   403.509100          8.106864     1.024165
         min      18.000000   102.000000          1.000000     1.000000
         25%      30.000000   465.000000          2.000000     2.000000
         50%      36.000000   802.000000          7.000000     3.000000
         75%      43.000000  1157.000000         14.000000     4.000000
         max      60.000000  1499.000000         29.000000     5.000000

                EnvironmentSatisfaction   HourlyRate  JobInvolvement     JobLevel  \
         count              1470.000000  1470.000000     1470.000000  1470.000000
         mean                  2.721769    65.891156        2.729932     2.063946
         std                   1.093082    20.329428        0.711561     1.106940
         min                   1.000000    30.000000        1.000000     1.000000
         25%                   2.000000    48.000000        2.000000     1.000000
         50%                   3.000000    66.000000        3.000000     2.000000
         75%                   4.000000    83.750000        3.000000     3.000000
         max                   4.000000   100.000000        4.000000     5.000000

                JobSatisfaction  MonthlyIncome          ...            \
         count      1470.000000    1470.000000          ...
         mean          2.728571    6502.931293          ...
         std           1.102846    4707.956783          ...
         min           1.000000    1009.000000          ...
         25%           2.000000    2911.000000          ...
         50%           3.000000    4919.000000          ...
         75%           4.000000    8379.000000          ...
         max           4.000000   19999.000000          ...

                PerformanceRating  RelationshipSatisfaction  StockOptionLevel  \
         count        1470.000000               1470.000000       1470.000000
         mean            3.153741                  2.712245          0.793878
         std             0.360824                  1.081209          0.852077
         min             3.000000                  1.000000          0.000000
         25%             3.000000                  2.000000          0.000000
```

```
50%              3.000000                    3.000000          1.000000
75%              3.000000                    4.000000          1.000000
max              4.000000                    4.000000          3.000000

       TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
count        1470.000000            1470.000000      1470.000000
mean           11.279592               2.799320         2.761224
std             7.780782               1.289271         0.706476
min             0.000000               0.000000         1.000000
25%             6.000000               2.000000         2.000000
50%            10.000000               3.000000         3.000000
75%            15.000000               3.000000         3.000000
max            40.000000               6.000000         4.000000

       YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
count     1470.000000         1470.000000              1470.000000
mean         7.008163            4.229252                 2.187755
std          6.126525            3.623137                 3.222430
min          0.000000            0.000000                 0.000000
25%          3.000000            2.000000                 0.000000
50%          5.000000            3.000000                 1.000000
75%          9.000000            7.000000                 3.000000
max         40.000000           18.000000                15.000000

       YearsWithCurrManager
count           1470.000000
mean               4.123129
std                3.568136
min                0.000000
25%                2.000000
50%                3.000000
75%                7.000000
max               17.000000

[8 rows x 23 columns]
```

Further, the following two separate DataFrame are created: (1) number_data, which consists of only numerica data, and (2) categorical_data, which consists of only categorical data.

In [36]:
```
number_data = attrition_data[['Age', 'DailyRate', 'DistanceFromHome', 'HourlyRate',
       'JobLevel', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'PercentSala
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompa
       'YearsSinceLastPromotion', 'YearsWithCurrManager']]
number_data
```

Out[36]:
```
                Age  DailyRate  DistanceFromHome  HourlyRate  JobLevel  \
EmployeeNumber
```

1	41	1102	1	94	2
2	49	279	8	61	2
4	37	1373	2	92	1
5	33	1392	3	56	1
7	27	591	2	40	1
8	32	1005	2	79	1
10	59	1324	3	81	1
11	30	1358	24	67	1
12	38	216	23	44	3
13	36	1299	27	94	2
14	35	809	16	84	1
15	29	153	15	49	2
16	31	670	26	31	1
18	34	1346	19	93	1
19	28	103	24	50	1
20	29	1389	21	51	3
21	32	334	5	80	1
22	22	1123	16	96	1
23	53	1219	2	78	4
24	38	371	2	45	1
26	24	673	11	96	2
27	36	1218	9	82	1
28	34	419	7	53	3
30	21	391	15	96	1
31	34	699	6	83	1
32	53	1282	5	58	5
33	32	1125	16	72	1
35	42	691	8	48	2
36	44	477	7	42	3
38	46	705	2	83	5
...	...	...	...	...	...
2025	36	688	4	97	2
2026	56	667	1	57	2
2027	29	1092	1	36	1
2031	42	300	2	56	5
2032	56	310	7	72	1
2034	41	582	28	60	4
2035	34	704	28	95	2
2036	36	301	15	88	2
2037	41	930	3	57	2
2038	32	529	2	78	1
2040	35	1146	26	31	3
2041	38	345	10	100	2
2044	50	878	1	94	2
2045	36	1120	11	100	2
2046	45	374	20	50	2
2048	40	1322	2	52	1
2049	35	1199	18	80	2

2051	40	1194	2	98	1
2052	35	287	1	62	1
2053	29	1378	13	46	2
2054	29	468	28	73	1
2055	50	410	28	39	3
2056	39	722	24	60	4
2057	31	325	5	74	2
2060	26	1167	5	30	1
2061	36	884	23	41	2
2062	39	613	6	42	3
2064	27	155	4	87	2
2065	49	1023	2	63	2
2068	34	628	8	82	2

	MonthlyIncome	MonthlyRate	NumCompaniesWorked	\
EmployeeNumber				
1	5993	19479	8	
2	5130	24907	1	
4	2090	2396	6	
5	2909	23159	1	
7	3468	16632	9	
8	3068	11864	0	
10	2670	9964	4	
11	2693	13335	1	
12	9526	8787	0	
13	5237	16577	6	
14	2426	16479	0	
15	4193	12682	0	
16	2911	15170	1	
18	2661	8758	0	
19	2028	12947	5	
20	9980	10195	1	
21	3298	15053	0	
22	2935	7324	1	
23	15427	22021	2	
24	3944	4306	5	
26	4011	8232	0	
27	3407	6986	7	
28	11994	21293	0	
30	1232	19281	1	
31	2960	17102	2	
32	19094	10735	4	
33	3919	4681	1	
35	6825	21173	0	
36	10248	2094	3	
38	18947	22822	3	
...	...	...	...	
2025	5131	9192	7	

2026	6306	26236	1
2027	4787	26124	9
2031	18880	17312	5
2032	2339	3666	8
2034	13570	5640	0
2035	6712	8978	1
2036	5406	10436	1
2037	8938	12227	2
2038	2439	11288	1
2040	8837	16642	1
2041	5343	5982	1
2044	6728	14255	7
2045	6652	14369	4
2046	4850	23333	8
2048	2809	2725	2
2049	5689	24594	1
2051	2001	12549	2
2052	2977	8952	1
2053	4025	23679	4
2054	3785	8489	1
2055	10854	16586	4
2056	12031	8828	0
2057	9936	3787	0
2060	2966	21378	0
2061	2571	12290	4
2062	9991	21457	4
2064	6142	5174	1
2065	5390	13243	2
2068	4404	10228	2

	PercentSalaryHike	StockOptionLevel	TotalWorkingYears \
EmployeeNumber			
1	11	0	8
2	23	1	10
4	15	0	7
5	11	0	8
7	12	1	6
8	13	0	8
10	20	3	12
11	22	1	1
12	21	0	10
13	13	2	17
14	13	1	6
15	12	0	10
16	17	1	5
18	11	1	3
19	14	0	6
20	11	1	10

21	12	2	7
22	13	2	1
23	16	0	31
24	11	0	6
26	18	1	5
27	23	0	10
28	11	0	13
30	14	0	0
31	11	0	8
32	11	1	26
33	22	0	10
35	11	1	10
36	14	1	24
38	12	0	22
...	...	...	...
2025	13	3	18
2026	21	1	13
2027	14	3	4
2031	11	0	24
2032	11	1	14
2034	23	1	21
2035	21	2	8
2036	24	1	15
2037	11	1	14
2038	14	0	4
2040	16	0	9
2041	11	1	10
2044	12	2	12
2045	13	1	8
2046	15	0	8
2048	14	0	8
2049	14	2	10
2051	14	3	20
2052	12	1	4
2053	13	1	10
2054	14	0	5
2055	13	1	20
2056	11	1	21
2057	19	0	10
2060	18	0	5
2061	17	1	17
2062	15	1	9
2064	20	1	6
2065	14	0	17
2068	12	0	6

```
                TrainingTimesLastYear  YearsAtCompany  YearsInCurrentRole  \
EmployeeNumber
```

1	0	6	4
2	3	10	7
4	3	0	0
5	3	8	7
7	3	2	2
8	2	7	7
10	3	1	0
11	2	1	0
12	2	9	7
13	3	7	7
14	5	5	4
15	3	9	5
16	1	5	2
18	2	2	2
19	4	4	2
20	1	10	9
21	5	6	2
22	2	1	0
23	3	25	8
24	3	3	2
26	5	4	2
27	4	5	3
28	4	12	6
30	6	0	0
31	2	4	2
32	3	14	13
33	5	10	2
35	2	9	7
36	4	22	6
38	2	2	2
...	...	...	...
2025	3	4	2
2026	2	13	12
2027	3	2	2
2031	2	22	6
2032	4	10	9
2034	3	20	7
2035	2	8	7
2036	4	15	12
2037	5	5	4
2038	4	4	2
2040	2	9	0
2041	1	10	7
2044	3	6	3
2045	2	6	3
2046	3	5	3
2048	2	2	2
2049	2	10	2

2051	2	5	3
2052	5	4	3
2053	2	4	3
2054	3	5	4
2055	3	3	2
2056	2	20	9
2057	2	9	4
2060	2	4	2
2061	3	5	2
2062	5	7	7
2064	0	6	2
2065	3	9	6
2068	3	4	3

	YearsSinceLastPromotion	YearsWithCurrManager
EmployeeNumber		
1	0	5
2	1	7
4	0	0
5	3	0
7	2	2
8	3	6
10	0	0
11	0	0
12	1	8
13	7	7
14	0	3
15	0	8
16	4	3
18	1	2
19	0	3
20	8	8
21	0	5
22	0	0
23	3	7
24	1	2
26	1	3
27	0	3
28	2	11
30	0	0
31	1	3
32	4	8
33	6	7
35	4	2
36	5	17
38	2	1
...	...	...
2025	0	2

```
2026                                      1                      9
2027                                      2                      2
2031                                      4                     14
2032                                      9                      8
2034                                      0                     10
2035                                      1                      7
2036                                     11                     11
2037                                      0                      4
2038                                      1                      2
2040                                      1                      7
2041                                      1                      9
2044                                      0                      1
2045                                      0                      0
2046                                      0                      1
2048                                      2                      2
2049                                      0                      2
2051                                      0                      2
2052                                      1                      1
2053                                      0                      3
2054                                      0                      4
2055                                      2                      0
2056                                      9                      6
2057                                      1                      7
2060                                      0                      0
2061                                      0                      3
2062                                      1                      7
2064                                      0                      3
2065                                      0                      8
2068                                      1                      2

[1470 rows x 16 columns]
```

In [37]:
```
categorical_data = attrition_data[['Attrition', 'Gender', 'Education', 'EnvironmentSa
                            'PerformanceRating', 'RelationshipSatisfaction','WorkLifel
categorical_data
```

Out[37]:
```
               Attrition  Gender  Education  EnvironmentSatisfaction  \
EmployeeNumber
1                    Yes  Female          2                        2
2                     No    Male          1                        3
4                    Yes    Male          2                        4
5                     No  Female          4                        4
7                     No    Male          1                        1
8                     No    Male          2                        4
10                    No  Female          3                        3
11                    No    Male          1                        4
12                    No    Male          3                        4
13                    No    Male          3                        3
```

14	No	Male	3	1
15	No	Female	2	4
16	No	Male	1	1
18	No	Male	2	2
19	Yes	Male	3	3
20	No	Female	4	2
21	No	Male	2	1
22	No	Male	2	4
23	No	Female	4	1
24	No	Male	3	4
26	No	Female	2	1
27	Yes	Male	4	3
28	No	Female	4	1
30	No	Male	2	3
31	Yes	Male	1	2
32	No	Female	3	3
33	Yes	Female	1	2
35	No	Male	4	3
36	No	Female	4	1
38	No	Female	4	2
...	...	...	...	...
2025	No	Female	2	4
2026	No	Male	4	3
2027	Yes	Male	4	1
2031	No	Male	3	1
2032	Yes	Male	2	4
2034	No	Female	4	1
2035	No	Female	3	4
2036	No	Male	4	4
2037	No	Male	3	3
2038	No	Male	3	4
2040	No	Female	4	3
2041	No	Female	2	1
2044	Yes	Male	4	2
2045	No	Female	4	2
2046	No	Female	3	4
2048	No	Male	4	3
2049	No	Male	4	3
2051	No	Female	4	3
2052	No	Female	4	3
2053	No	Male	2	4
2054	No	Female	4	4
2055	Yes	Male	3	4
2056	No	Female	1	2
2057	No	Male	3	2
2060	No	Female	3	4
2061	No	Male	2	3
2062	No	Male	1	4

```
2064                 No    Male           3                          2
2065                 No    Male           3                          4
2068                 No    Male           3                          2

                JobInvolvement  JobSatisfaction  PerformanceRating  \
EmployeeNumber
1                            3                4                  3
2                            2                2                  4
4                            2                3                  3
5                            3                3                  3
7                            3                2                  3
8                            3                4                  3
10                           4                1                  4
11                           3                3                  4
12                           2                3                  4
13                           3                3                  3
14                           4                2                  3
15                           2                3                  3
16                           3                3                  3
18                           3                4                  3
19                           2                3                  3
20                           4                1                  3
21                           4                2                  3
22                           4                4                  3
23                           2                4                  3
24                           3                4                  3
26                           4                3                  3
27                           2                1                  4
28                           3                2                  3
30                           3                4                  3
31                           3                1                  3
32                           3                3                  3
33                           1                1                  4
35                           3                2                  3
36                           2                4                  3
38                           3                1                  3
...                        ...              ...                ...
2025                         3                2                  3
2026                         3                3                  4
2027                         3                4                  3
2031                         3                3                  3
2032                         3                3                  3
2034                         2                2                  4
2035                         2                3                  4
2036                         1                4                  4
2037                         2                2                  3
2038                         3                1                  3
2040                         3                4                  3
```

2041	3	4	3
2044	3	3	3
2045	2	4	3
2046	3	3	3
2048	2	3	3
2049	3	3	3
2051	3	3	3
2052	1	4	3
2053	2	2	3
2054	2	1	3
2055	2	1	3
2056	2	4	3
2057	3	1	3
2060	2	3	3
2061	4	4	3
2062	2	1	3
2064	4	2	4
2065	2	2	3
2068	4	3	3

	RelationshipSatisfaction	WorkLifeBalance
EmployeeNumber		
1	1	1
2	4	3
4	2	3
5	3	3
7	4	3
8	3	2
10	1	2
11	2	3
12	2	3
13	2	2
14	3	3
15	4	3
16	4	2
18	3	3
19	2	3
20	3	3
21	4	2
22	2	2
23	3	3
24	3	3
26	4	2
27	2	3
28	3	3
30	4	3
31	3	3
32	4	2

```
33                                                  2              3
35                                                  4              3
36                                                  4              3
38                                                  4              2
...                                               ...            ...
2025                                                2              3
2026                                                1              2
2027                                                2              4
2031                                                1              2
2032                                                4              1
2034                                                3              3
2035                                                4              3
2036                                                1              2
2037                                                3              3
2038                                                4              3
2040                                                3              3
2041                                                3              3
2044                                                4              3
2045                                                1              2
2046                                                3              3
2048                                                4              3
2049                                                4              4
2051                                                2              3
2052                                                4              3
2053                                                1              3
2054                                                2              1
2055                                                2              3
2056                                                1              2
2057                                                2              3
2060                                                4              3
2061                                                3              3
2062                                                1              3
2064                                                2              3
2065                                                4              2
2068                                                1              4

[1470 rows x 9 columns]
```

Applying NumPy's .mean(), .median(), and .mode(), to double-check the results for the selected column, 'MonthlyIncome'.

```
In [38]: mean_monthly_income = np.mean(attrition_data['MonthlyIncome'])
         median_monthly_income = np.median(attrition_data['MonthlyIncome'])
         mode_monthly_income = stats.mode(attrition_data['MonthlyIncome'])

         print(mean_monthly_income, median_monthly_income, mode_monthly_income)
```

```
#Note the skewness of mean_monthly_income compared to median_monincome
```

```
6502.931292517007 4919.0 ModeResult(mode=array([2342]), count=array([4]))
```

Further, the percentile range (95%) for "MonthlyIncome", the variance, and the standard deviation are calculated to get a further sense for the shape of the dataset:

```
In [39]: # Measuring DISPERSION:

         print("Monthly Income Percentile Range, Variance, and Standard Deviation are: ")
         print("95% Percentile Range($): ", np.percentile(attrition_data['MonthlyIncome'],2.5)
         print("Variance($): ", np.var(attrition_data['MonthlyIncome']))
         print("Standard Deviation($): ", np.std(attrition_data['MonthlyIncome']))  # this sho
```

```
Monthly Income Percentile Range, Variance, and Standard Deviation are:
95% Percentile Range($):  2010.175 - 19191.925
Variance($):  22149778.937456165
Standard Deviation($):  4706.355164823004
```

Further notes on the dataset:
* Within in the dataset, there are 588 female and 882 male, the total 1,470 datapoints, with the gender ratio 40% for female, and 60% for male, respectively (see the output for attrition_gender_ratio, below.
* Further, for the Attrition/Yes group, the 37% of those who had attritioin are female while 63% are male.
* For the Attrition/No group, 41% are female while 59% are male. * This appears as though that for Attrition/No group, there is no apparent gender difference as the gender ratio in this group as the dataset's overal gender ratio is 4(female):6(male).
* This will be explored further below.

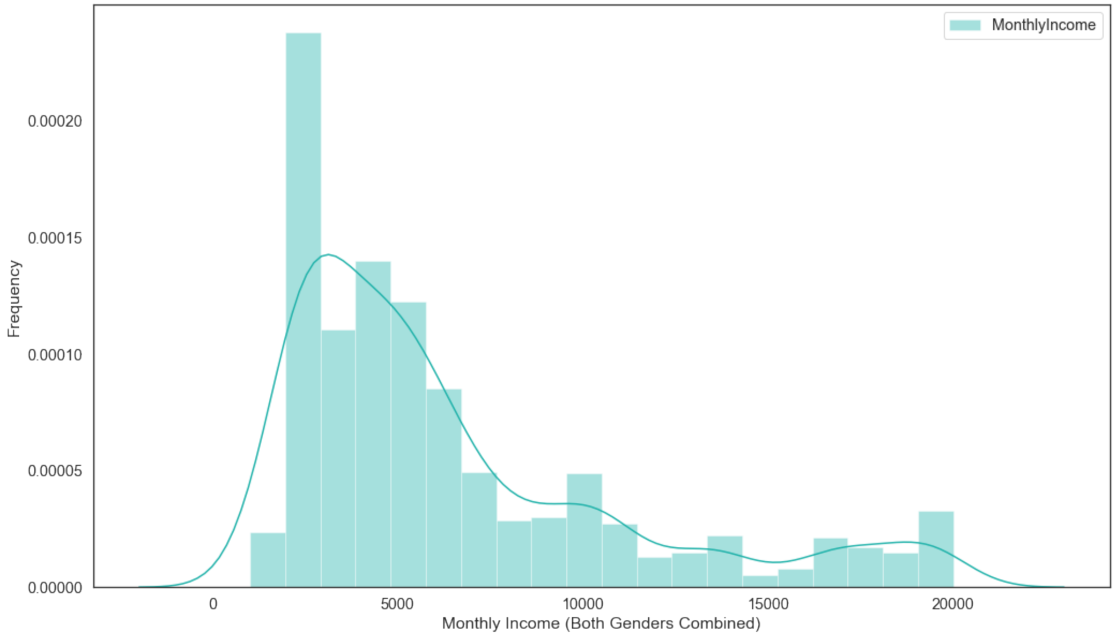
```
In [40]: attrition_gender_ratio = attrition_data.pivot_table(index='Attrition', columns='Gende
                                                               fill_value=0)

         # calculate ratios
         sums = attrition_gender_ratio[['Female', 'Male']].sum(axis=1)
         attrition_gender_ratio['FemaleRatio'] = attrition_gender_ratio['Female'] / sums
         attrition_gender_ratio['MaleRatio'] = attrition_gender_ratio['Male'] / sums
         attrition_gender_ratio
```

```
Out[40]: Gender     Female  Male  FemaleRatio  MaleRatio
         Attrition
         No            501   732     0.406326   0.593674
         Yes            87   150     0.367089   0.632911
```

7.0.1 Exploratory Question: *"How Does Income Income Affect Attrtion?"*:

First, a simple distribution is plotted for MonthlyIncome:

In [41]:
```
# Plot Female and Male in same plot
fig, axes = plt.subplots(1, 1, figsize=(17, 10))
sns.distplot(attrition_data[['MonthlyIncome']], axlabel=None, label='MonthlyIncome', c
plt.legend()
plt.xlabel('Monthly Income (Both Genders Combined)')
plt.ylabel('Frequency')
plt.show()
```

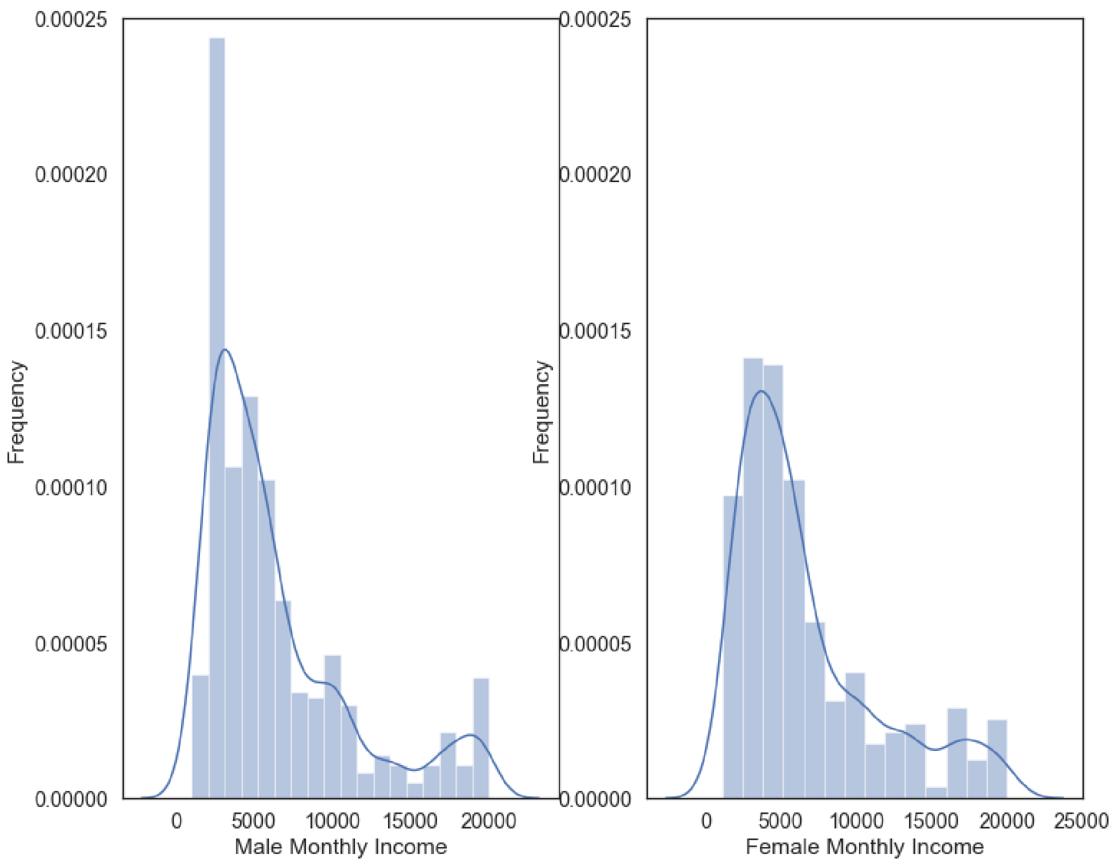


By looking at the above distribution for MonthlyIncome, there is a right-skewness, confirming that the MonthlyIncome means calculated earlier is the result of this skew, as a small portion of the datapoints' monthly income is almost 3-4 times more than the the majority of datapoints.

Next, MonthlyIncome distributions by Gender will be plotted to see if any apparent gender difference exists:

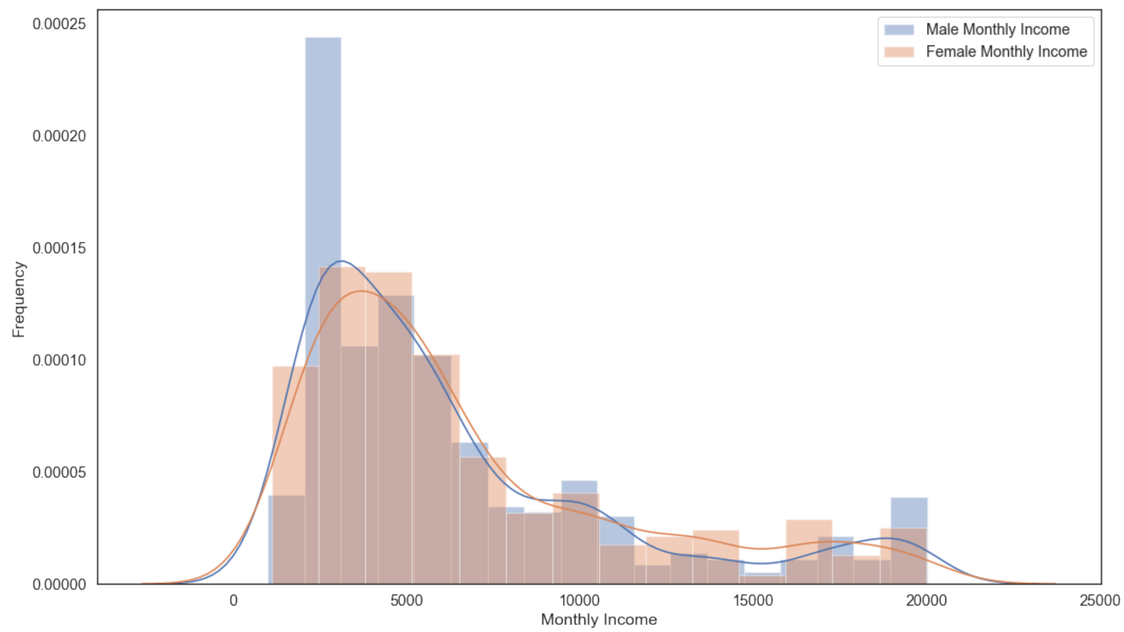
In [42]:
```
# Plot Female and Male in same plot
fig, axes = plt.subplots(1, 2, figsize=(12, 10))
attrition_data_M = attrition_data.loc[attrition_data['Gender'] == 'Male']
attrition_data_F = attrition_data.loc[attrition_data['Gender'] == 'Female']

sns.distplot(attrition_data_M[['MonthlyIncome']], axlabel=None, label='Male Monthly In
sns.distplot(attrition_data_F[['MonthlyIncome']], axlabel=None, label='Female Monthly
axes[0].set_xlabel("Male Monthly Income")
axes[0].set_ylabel("Frequency")
axes[1].set_xlabel("Female Monthly Income")
```

```
axes[1].set_ylabel("Frequency")
ylim = [0, .00025]
axes[0].set_ylim(ylim)
axes[1].set_ylim(ylim)
plt.show()
```



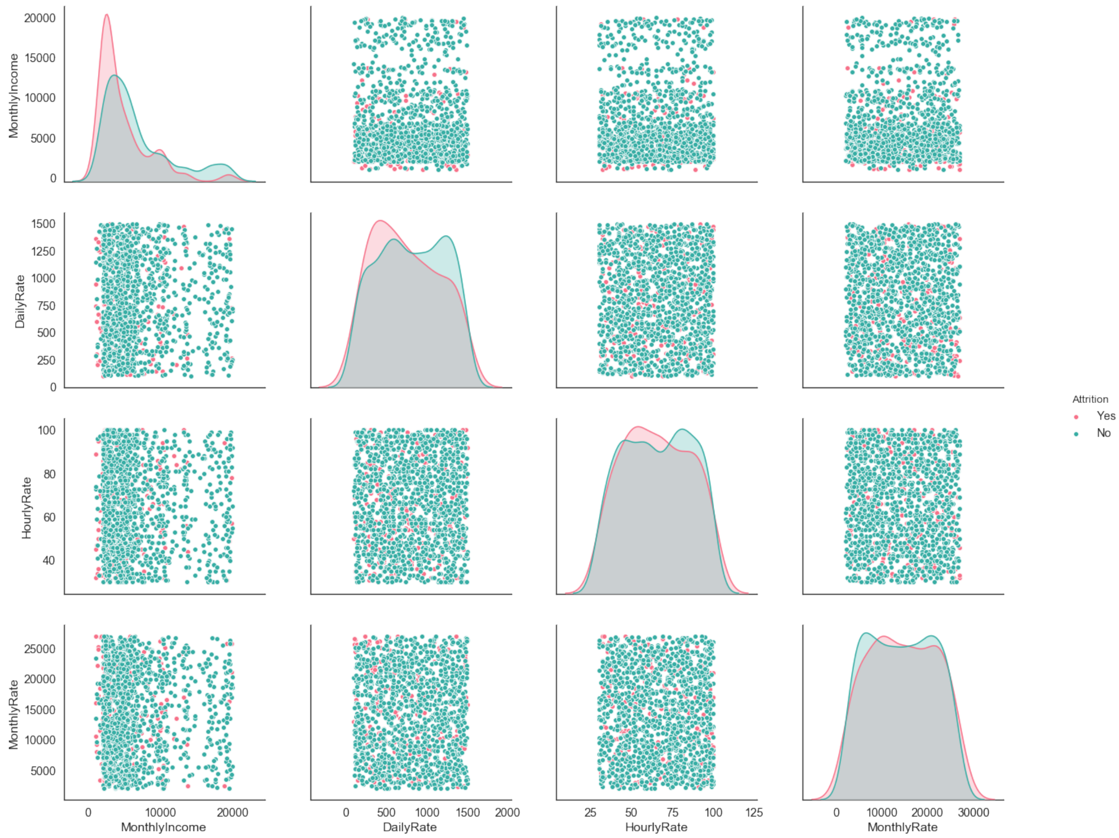
A superimposed version of the above distributions for an easier comparison:

```
In [43]: # Plot Female and Male in same plot
         fig, axes = plt.subplots(1, 1, figsize=(17, 10))
         attrition_data_M = attrition_data.loc[attrition_data['Gender'] == 'Male']
         attrition_data_F = attrition_data.loc[attrition_data['Gender'] == 'Female']

         sns.distplot(attrition_data_M[['MonthlyIncome']], axlabel=None, label='Male Monthly In
         sns.distplot(attrition_data_F[['MonthlyIncome']], axlabel=None, label='Female Monthly
         plt.legend()
         plt.xlabel('Monthly Income')
         plt.ylabel('Frequency')
         plt.show()
```



In terms of the shapes of the distributions, there is no apparent difference in distributions between the genders.
Next, all the income related numeric data -MonthlyIncome, DailyRate, HourlyRate, and MonthlyRate are plotted for both gender combined to see their respective distributions:

```
In [44]: g = sns.pairplot(attrition_data, x_vars=["MonthlyIncome", "DailyRate", "HourlyRate", "
                          y_vars=["MonthlyIncome", "DailyRate", "HourlyRate", "MonthlyRate"],
                          palette="husl")
         g.fig.set_figheight(15)
         g.fig.set_figwidth(20)
```



By default, this function will create a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. The diagonal Axes are treated differently, drawing a plot to show the univariate distribution of the data for the variable in that column.

The above pairplot distributions indicate that MonthlyIncome and DailyRate clearly showing in thier univariate distribiutions that the lower the income, higher the attrition. Similar is true for HourlyRate and MonthlyRate though not as visibly striking as MontlyIncome and DailyRate.

To make the visual analysis easier, the MonthlyIncome values are convered into ranges (brackets), in which each MonthlyIncome values are categorized into:

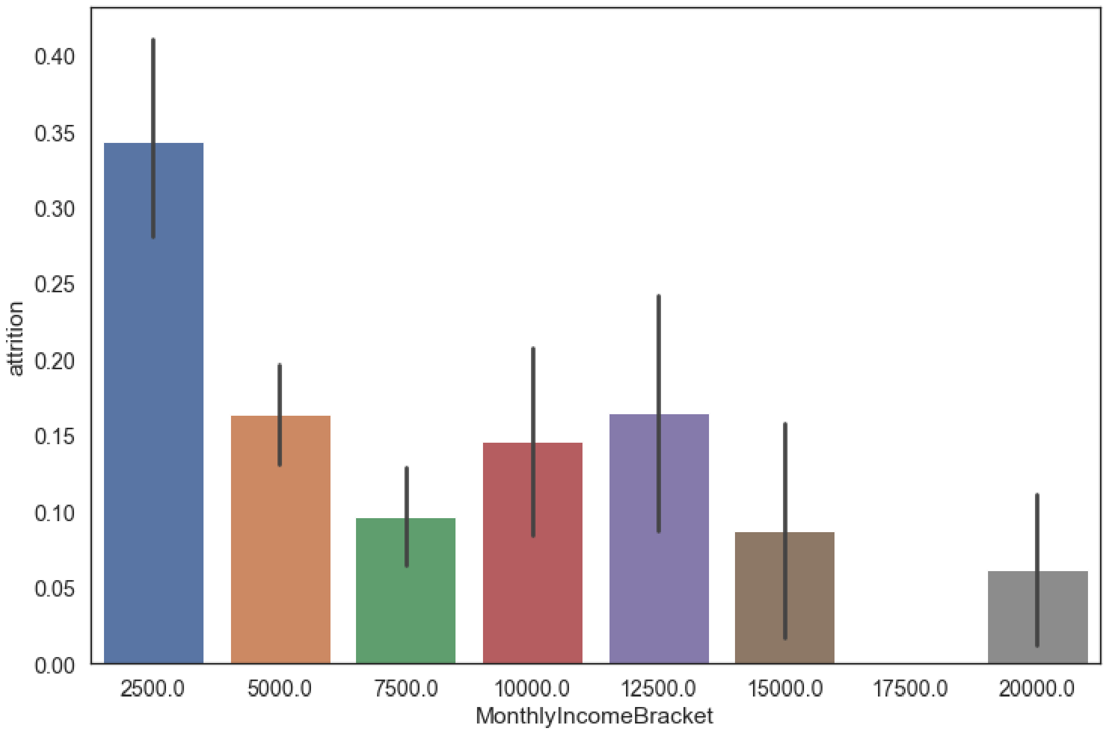
In [45]:
```
step = 2500
for start in range(0, attrition_data['MonthlyIncome'].max(), step):
  # 0 2500 5000
  rows = (attrition_data['MonthlyIncome'] >= start) & (attrition_data['MonthlyIncome']
  attrition_data.loc[rows, 'MonthlyIncomeBracket'] = start + step
attrition_data.head()
```

Out[45]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
```

```
1                    41      Yes      Travel_Rarely      1102
2                    49       No  Travel_Frequently       279
4                    37      Yes      Travel_Rarely      1373
5                    33       No  Travel_Frequently      1392
7                    27       No      Travel_Rarely       591

                           Department  DistanceFromHome  Education  \
EmployeeNumber
1                               Sales                 1          2
2              Research & Development                 8          1
4              Research & Development                 2          2
5              Research & Development                 3          4
7              Research & Development                 2          1

               EducationField  EmployeeCount  EnvironmentSatisfaction  \
EmployeeNumber
1               Life Sciences              1                        2
2               Life Sciences              1                        3
4                       Other              1                        4
5               Life Sciences              1                        4
7                     Medical              1                        1

                        ...         StandardHours  StockOptionLevel  \
EmployeeNumber          ...
1                       ...                    80                 0
2                       ...                    80                 1
4                       ...                    80                 0
5                       ...                    80                 0
7                       ...                    80                 1

                TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  \
EmployeeNumber
1                               8                      0               1
2                              10                      3               3
4                               7                      3               3
5                               8                      3               3
7                               6                      3               3

                YearsAtCompany YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                            6                  4                        0
2                           10                  7                        1
4                            0                  0                        0
5                            8                  7                        3
7                            2                  2                        2

                YearsWithCurrManager  MonthlyIncomeBracket
EmployeeNumber
```

```
1                                   5                7500.0
2                                   7                7500.0
4                                   0                2500.0
5                                   0                5000.0
7                                   2                5000.0

[5 rows x 35 columns]
```

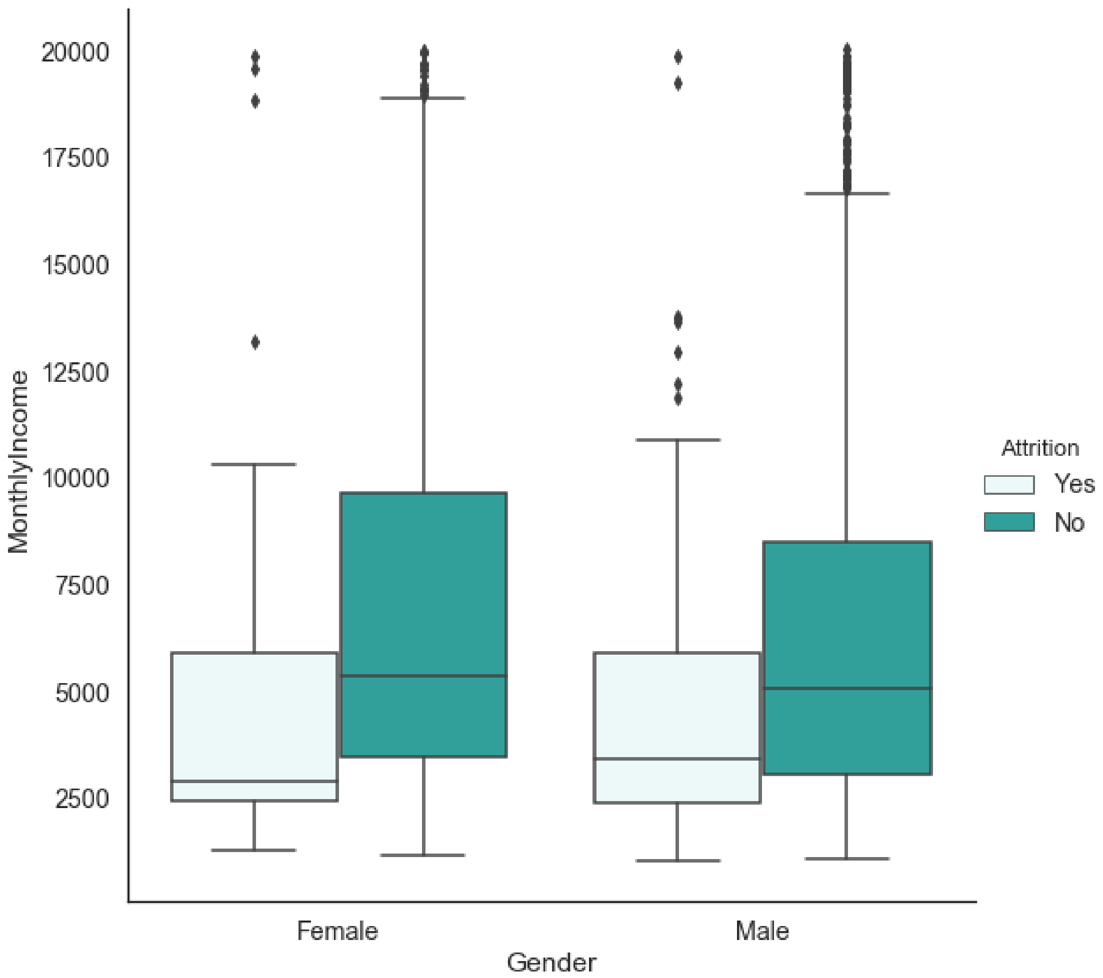
```
In [46]: fig, ax = plt.subplots(1, figsize=(12, 8))
         attrition_data['attrition'] = attrition_data['Attrition'].replace({'Yes': 1, 'No': 0}
         sns.barplot(x='MonthlyIncomeBracket', y='attrition', data=attrition_data)
         plt.show()
```



As seen in the above barplot, the monthly income group with the highest attrition is the lowest monthly income group (2500 dollars) followed by the next lowst group (5000 dollars). This appears to show that the lower the income, higher the attrition. To provide further contexts to this initial exploration, MonthlyIncome, HourlyRate, DailyRate, and MonthlyRate are plotted against Attrtion.

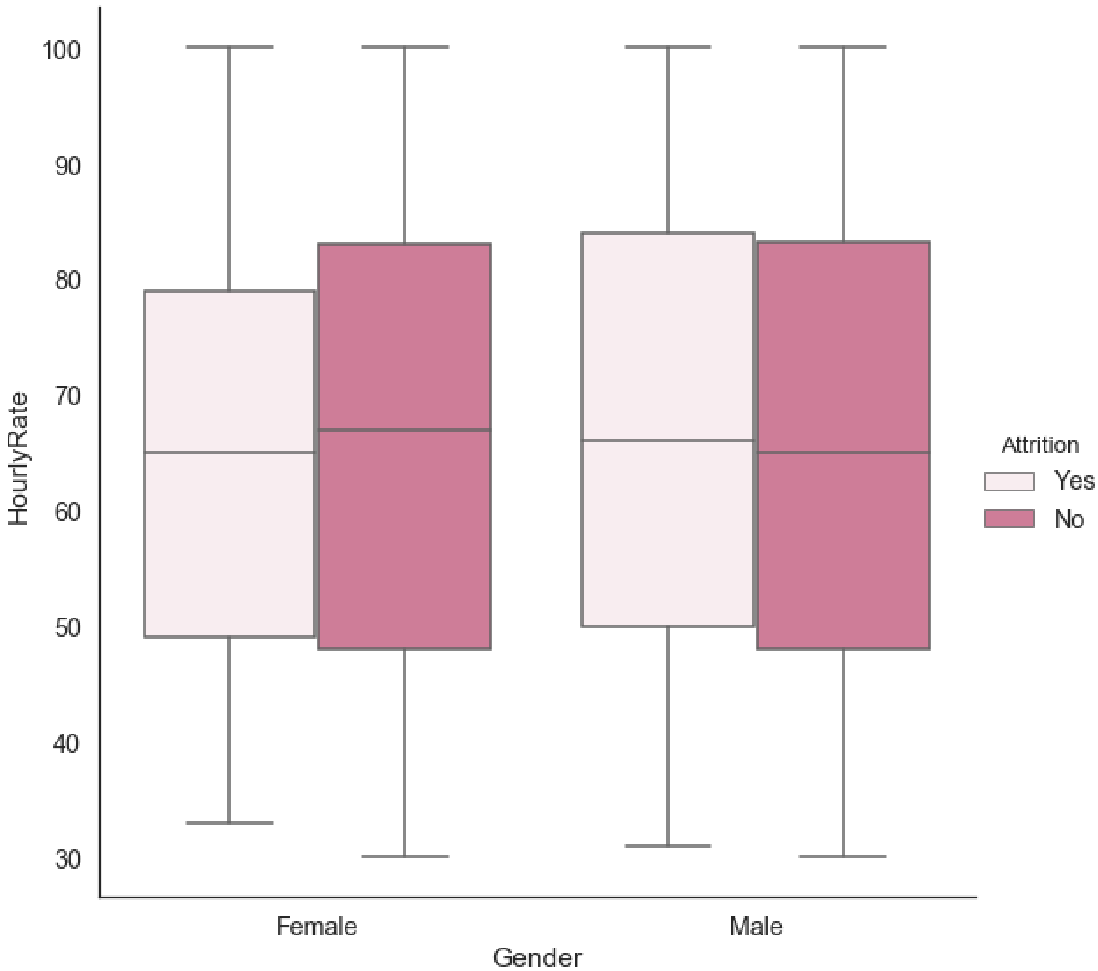
(1) Visualizing the relationship between MonthlyIncome and Attrition:

```
In [47]: #fig,axes = plt.subplots(1, figsize=(20,8))
         g = sns.catplot(x='Gender', y='MonthlyIncome', hue='Attrition', data=attrition_data, k
                         aspect=1, color="lightseagreen")
         plt.show()
```



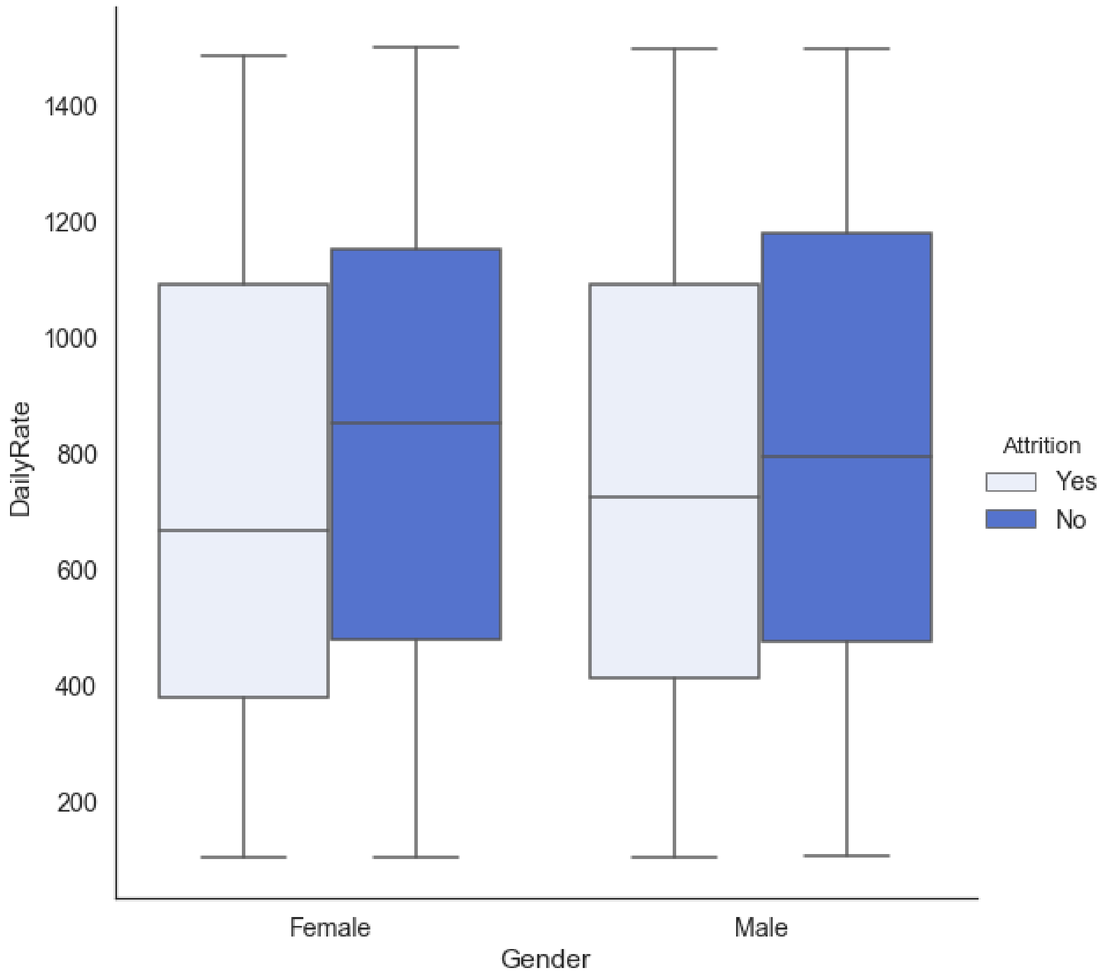
(2) Visualizing the relationship between HourlyRate and Attrition:

```
In [48]: g = sns.catplot(x='Gender', y='HourlyRate', hue='Attrition', data=attrition_data, kinc
                         aspect=1, color='palevioletred')
         plt.show()
```



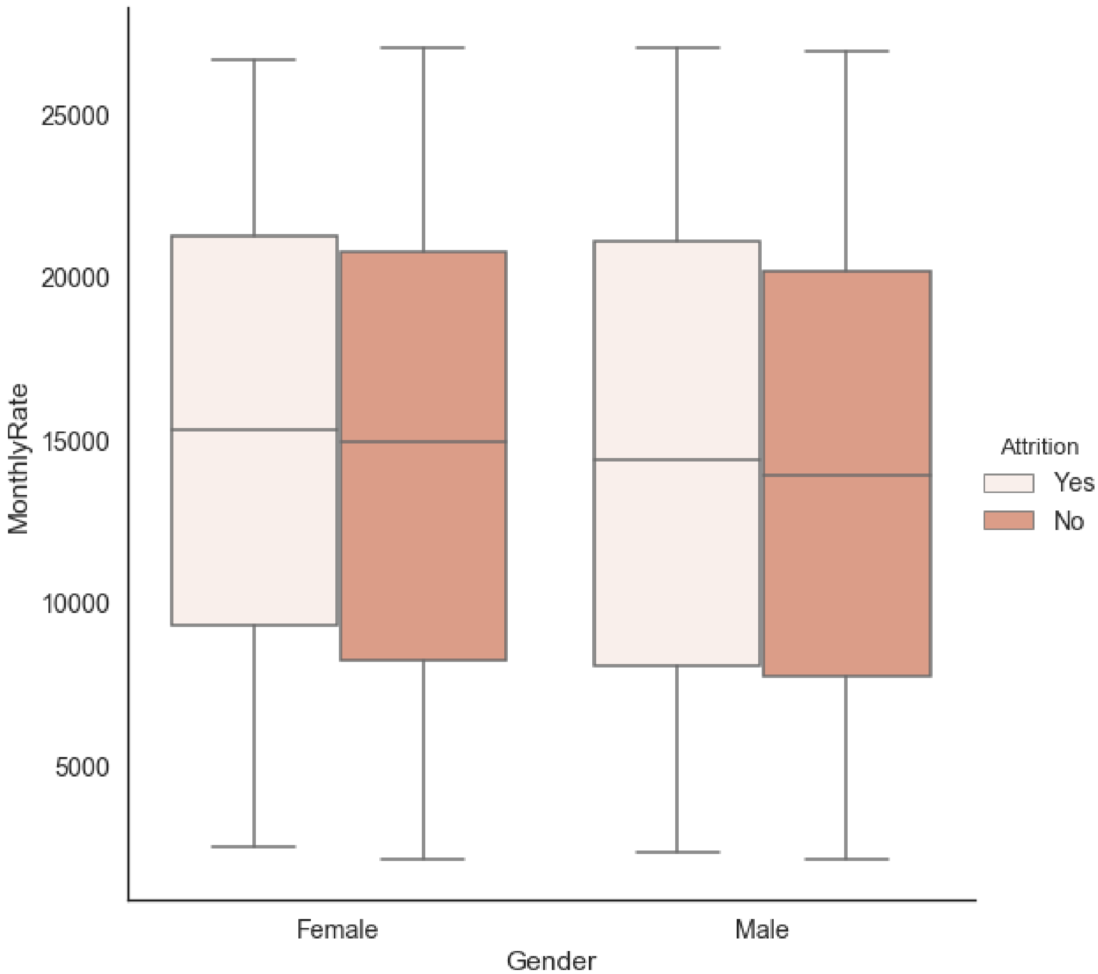
(3) Visualizing the relationship between DailyRate and Attrition:

```
In [49]: g = sns.catplot(x='Gender', y='DailyRate', hue='Attrition', data=attrition_data, kind=
                        aspect=1, color='royalblue')
         plt.show()
```



(4) Visualizing the relationship between MonthlyRate and Attrition:

```
In [50]: g = sns.catplot(x='Gender', y='MonthlyRate', hue='Attrition', data=attrition_data,
                         kind='box', height=8, aspect=1, color='darksalmon')
         plt.show()
```



Observations from the boxplots (1) to (4):

(1) Based on the boxplots above, MonthlyIncome distributions most strikinkly show (see the 50-percentile mark) that the individuals with Attrition/Yes tends to have the lower incomes in both genders.
The means for the both gender shows that the incomes for the 50 percentile are approximately 2,500 dollars for Attrition/Yes group while it is about 5,000 dollars for Attrition/No group. Also notable is how 50% of the income distributions for Attrtion/Yes groups in both genders (the boxes) are much shorter than that of its Attrtion/No groups. For the female Attrition/No group, the 50% of the distribution lies somewhere between about 3200 dollars to 9600 dollars. As for the male Attrtion/No group, the 50% of the distribution lies somewhere between about 3500 dollars to 8500 dollars. This observating is interesting in that just by looking at the 50% distributions alone, it appears as though female are earning more. However, it is important to note that the male Attrtion/No group has more outliers.

(2) The distibutions for DailyRate also show that the individuals with Attrition/Yes tends to have the lower incomes in both genders. The means for the HourlyRate discrepancy for the female shows that the incomes for the 50 percentile are approximately 660 dollars for Attrition/Yes group while it is about 830 dollars for Attrition/No group. The same is true for the male counterparts. While the discrepancy is not as wide as the female one, the differences between the means are approximately between 700 dollars to 750 dollars, which can make a significant difference in their paychecks (Note: No outliers exist).

(3) The distributions for HourlyRate indicates that the female Attrition/Yes earn less than the Attrition/No group while the male Attrition/Yes grouyp earning slightly more than its Attrition/No group.

(4) For the MonthlyRate distributions, the Attrition/No groups for both male and female have slightly higher than that of the ones for the Attrition/Yes groups.

(5) Although not all the distributuions show that the lower rates of income/wages always indicate higher Attrition/Yes, more than half the distributuions above may warrant the following hypothesis:

H0: The lower income/wages have no effect on the job attrition.
H1: The lower income/wages does have an effect on the job attrition.

(6) To test the hypothesis, t-test is performed as follows:

```
In [51]: stats.ttest_ind(attrition_data[attrition_data.Attrition == 'Yes']['MonthlyIncome'],
                         attrition_data[attrition_data.Attrition == 'No']['MonthlyIncome'])
```

```
Out[51]: Ttest_indResult(statistic=-6.203935765608938, pvalue=7.14736398535381e-10)
```

Results:
+ The result of the t-test performed above, the calculated p-value (pvalue=7.14736398535381e-10) is significantly smaller than 0.05. + This indicates the strong evidence against the null hypothesis H0=Income makes no difference in attrition.
+ Therefore, I can safely rejct the null hypothesis, thus conclude that the alternative hypothesis is true, income does make a difference in attrition.

Before concluding the exploration on the question of income and attrtion, the following tabular information is created:

```
In [52]: attrition_pivot = attrition_data.pivot_table(index=['Attrition', 'MonthlyIncomeBracke
                                   values='Age', aggfunc=lambda x: len(x) if len(x) != np.nan
         attrition_pivot
```

```
Out[52]: Gender                            Female  Male   All
         Attrition MonthlyIncomeBracket
         No        2500.0                    53    94   147
                   5000.0                   175   264   439
                   7500.0                   117   163   280
                   10000.0                   40    71   111
```

```
            12500.0                            34    42    76
            15000.0                            30    22    52
            17500.0                            28    24    52
            20000.0                            24    52    76
Yes         2500.0                             25    52    77
            5000.0                             36    50    86
            7500.0                             10    20    30
            10000.0                             9    10    19
            12500.0                             3    12    15
            15000.0                             1     4     5
            20000.0                             3     2     5
All                                           588   882  1470
```

To put the above numbers in perspective, the following pivot table is created to see the gender ratios for each monthly income brackets:

In [53]:
```
attrition_pivot = attrition_pivot.rename(index={'All': 'Total'}, columns={'All': 'Tota
attrition_pivot.loc[:, 'Male'] = (attrition_pivot.loc[:, 'Male'] / attrition_pivot.loc
attrition_pivot.loc[:, 'Female'] = (attrition_pivot.loc[:, 'Female'] / attrition_pivot
attrition_pivot.loc[:, 'total_prcnt'] = (attrition_pivot.loc[:, 'Total'] / attrition_p
attrition_pivot
```

Out[53]:

Gender		Female	Male	Total	total_prcnt
Attrition	MonthlyIncomeBracket				
No	2500.0	36.054422	63.945578	147	100.0
	5000.0	39.863326	60.136674	439	100.0
	7500.0	41.785714	58.214286	280	100.0
	10000.0	36.036036	63.963964	111	100.0
	12500.0	44.736842	55.263158	76	100.0
	15000.0	57.692308	42.307692	52	100.0
	17500.0	53.846154	46.153846	52	100.0
	20000.0	31.578947	68.421053	76	100.0
Yes	2500.0	32.467532	67.532468	77	100.0
	5000.0	41.860465	58.139535	86	100.0
	7500.0	33.333333	66.666667	30	100.0
	10000.0	47.368421	52.631579	19	100.0
	12500.0	20.000000	80.000000	15	100.0
	15000.0	20.000000	80.000000	5	100.0
	20000.0	60.000000	40.000000	5	100.0
Total		40.000000	60.000000	1470	100.0

Observations:
It is interesting to see that some of the income groups has a similar ratio to the overall dataset ratio (female:4, male:6).
For example, looking at the groups for Attrtion/No, the gender ratio (percentage) for female is anywhere approximately between 36% to 45% for the income groups between 2500 and 12500, staying withing the 10% range of the 40%.
The 15000 dollar group for female Attrtion/No is almost 58%, the highest percentage, for the female Attrtion/No group. Together with the 17500 dollar group, female in these income category

show higher percentage than that of the males though the percentage dramatically decreases for 20000 group.

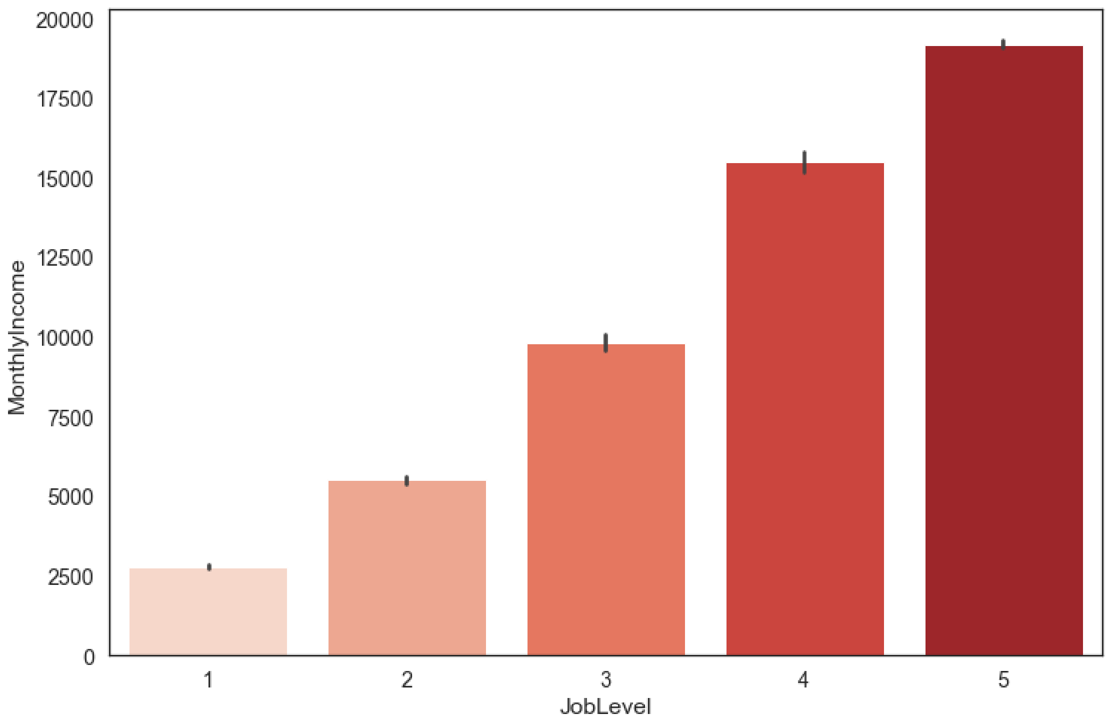
Looking at the Attrition/Yes group, the lower income groups do not seem to have anything striking in terms of gender differences, compared to the Atrtion/No group. However, looking at the income groups 12500, 15000, for the Attrtion/Yes, the gender ratio is 2:8, meaning that only 20% of female are in these categories (only one female in each of these two groups). Even though the 20000 income group indicates the 60% of female is in this category, the row total within this category verifies that 3 out of 5 individuals, who are in this income category, are female, which does not appear to be significant. Overall, it seems more notable to see the gender ratio differences in 12500 and 15000 dollars income categories for Attrtion/Yes as these may be an indication that female tend to reach the ceiling of highest earning potentials compared to male whatever the cause may be.

Next, the topic of the Job Role/Job Level and Attrtion is investigated.

7.0.2 Exploratory Question: *"Can Job Level and/or Job Role affect Attrition?"*:

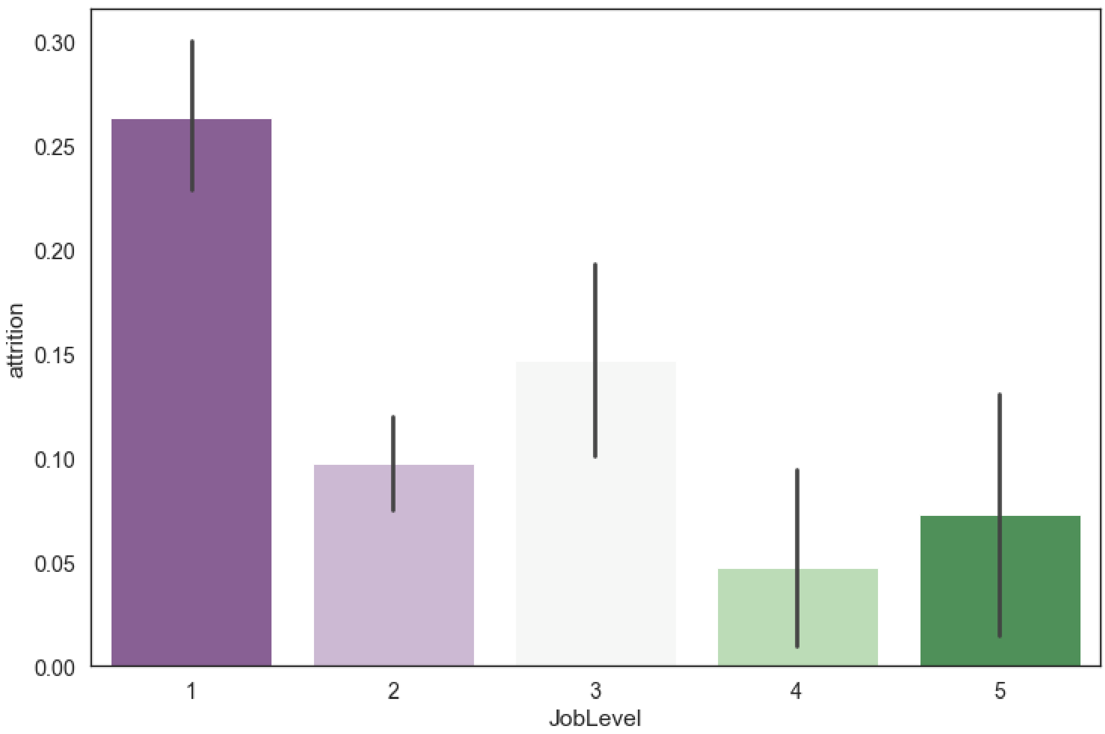
Notes on the relevant columns: This topic involves categorical data mainly, JobLevel and JobRole. As mentioned earlier, JobLevel consists of 1 to 5 categories while JobRole consists of 9 job titles, (e.g.) Sales Representative, etc. There is no clear group, or demarcations between the JobLevel and JobRoles. For instance, JobLevel 1 includes the JobRoles such as Sales Representative, Laboratory Technicians, Research Scientist, etc, while JobLevel 2 can include all the said JobRoles. For example, JobRole, "Manager" appears in JobLevel 3, 4, and 5. That said, it is important to note that the lower number the JobLevel is, the income is lower as shown in the below plot:

```
In [54]: fig, ax = plt.subplots(1, figsize=(12, 8))
         sns.barplot(x='JobLevel', y='MonthlyIncome', data=attrition_data, palette='Reds')
         plt.show()
```



First, a simple bar plot is created against Attrtion to see if a general trend exists.

```
In [55]: fig, ax = plt.subplots(1, figsize=(12, 8))
         sns.barplot(x='JobLevel', y='attrition', data=attrition_data, palette='PRGn')
         plt.show()
```



Although there is no clean and clear trend is seen, the JobLevel 1 clearly indicates the group's vulnerability to job attrtion. The figure for this group, 0.26-0.27 is much higher than the ones for JobLevel 4 and 5 ranging from approximately 0.04 to 0.075. This observation is further confirmed by creating the following crosstab shwoing the percentages for each group.

```
In [56]: joblevel = pd.crosstab(attrition_data['Attrition'], attrition_data['JobLevel'], margi
         joblevel * 100
```

```
Out[56]: JobLevel           1          2          3         4         5         All
         Attrition
         No         27.210884  32.789116  12.653061  6.870748  4.353741   83.877551
         Yes         9.727891   3.537415   2.176871  0.340136  0.340136   16.122449
         All        36.938776  36.326531  14.829932  7.210884  4.693878  100.000000
```

The crosstab above shows that 16% of the dataset indicates job attrtion (Attrition/Yes), almost 10% of which belongs to the JobLevel 1. This is almost 63% of the total Attrition/Yes. This obser-vataion leads to the following hypothesis:

H0: The JobLevel has no effect on Attrition. H1: The JobLevel does have an effect on Attrition (H0 is not True).

One-Way ANOVA will be performed to test this hypothesis. To do so, a list needs to be created first so that the list variable can be passed onto the inside .f_oneway().

```
In [57]: joblevels = []
         for i in range(1, 6):
           jl = attrition_data.loc[attrition_data['JobLevel'] == i, 'attrition']
```

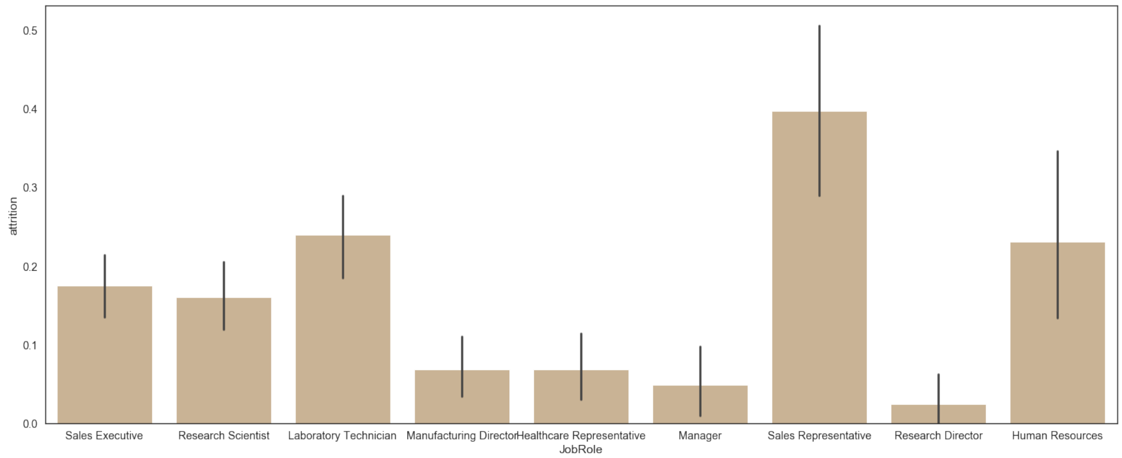
```
        joblevels.append(jl)
    stats.f_oneway(*joblevels)
```

```
Out[57]: F_onewayResult(statistic=19.0084454700361, pvalue=2.975150100310332e-15)
```

The calculated pvalue=2.975150100310332e-15, significantly smaller than 0.05. Therefore, I can safely reject the null hypothesis (H0: The JobLevel has no effect in Attrition).

To investigate further, another relevant column, JobRole will be analyzed in a similar fashion. First, a simple barplot is created for JobRoles agaisnt Attrition as follows:

```
In [58]: fig, ax = plt.subplots(1, figsize=(25, 10))
         sns.barplot(x='JobRole', y='attrition', data=attrition_data, color='tan')
         plt.show()
```



The bargraph above indicates that the Sales Representative, Laboratory Technician, and Human Resources professionals seem more vulnerable to job attrition compared to their counterparts.

The following pivot table and crosstab are created to see the relationship between JobLevel and JobRole.

```
In [59]: levelandrole_pivot = attrition_data.pivot_table(index=['JobRole'], columns=['JobLevel
                                          values='Attrition', aggfunc=lambda x: len(x) if len(x) != 
         levelandrole_pivot
```

```
Out[59]: JobLevel                      1      2      3      4      5   All
         JobRole
         Healthcare Representative   NaN   78.0   44.0    9.0    NaN   131
         Human Resources            33.0   13.0    6.0    NaN    NaN    52
         Laboratory Technician     200.0   56.0    3.0    NaN    NaN   259
```

```
Manager                        NaN    NaN   12.0   47.0  43.0   102
Manufacturing Director         NaN   90.0   45.0   10.0   NaN   145
Research Director              NaN    NaN   28.0   26.0  26.0    80
Research Scientist           234.0   57.0    1.0    NaN   NaN   292
Sales Executive                NaN  233.0   79.0   14.0   NaN   326
Sales Representative          76.0    7.0    NaN    NaN   NaN    83
All                          543.0  534.0  218.0  106.0  69.0  1470
```

```
In [60]: jobrole = pd.crosstab(attrition_data['Attrition'], attrition_data['JobRole'], margins=
                              normalize=True)
         jobrole * 100
```

```
Out[60]: JobRole    Healthcare Representative  Human Resources  Laboratory Technician  \
         Attrition
         No                          8.299320         2.721088              13.401361
         Yes                         0.612245         0.816327               4.217687
         All                         8.911565         3.537415              17.619048

         JobRole     Manager  Manufacturing Director  Research Director  \
         Attrition
         No         6.598639                9.183673           5.306122
         Yes        0.340136                0.680272           0.136054
         All        6.938776                9.863946           5.442177

         JobRole    Research Scientist  Sales Executive  Sales Representative  \
         Attrition
         No                  16.666667        18.299320              3.401361
         Yes                  3.197279         3.877551              2.244898
         All                 19.863946        22.176871              5.646259

         JobRole           All
         Attrition
         No          83.877551
         Yes         16.122449
         All        100.000000
```

As the table above indicates, the JobRoles such as Sales Representative, Laboratory Technician, and Human Resources professionals occupy the lower JobLevel, mostly in 1, followed by 2, and very few in 3. As seen earlier, the JobLevel 1 is the group that is particularly high in job attrtion.

The following hypothesis is formed as the result of this observations: H0: JobRole has no effect on Attrtion H1: JobRole does have an effect on Attrtion (H0 is not True).

```
In [61]: jobroles = []
         for j in attrition_data['JobRole'].unique():
           jr = attrition_data.loc[attrition_data['JobRole'] == j, 'attrition']
           jobroles.append(jr)
           # anova needs a list to be paassed onto
         print(len(jobroles))
         stats.f_oneway(*jobroles)
```

9

```
Out[61]: F_onewayResult(statistic=11.374753732967797, pvalue=9.562555450860023e-16)
```

The calculated pvalue=9.562555450860023e-16, is significantly smaller than 0.05. Therefore, I can safely reject the null hypothesis (H0: The JobRole has no effect in Attrition).

For the analyses and results of JobLevel and JobRole taken taken togher, I conclude that JobLevel and JobRole may lead to job attrition.

7.0.3 3. Hours of Commitment:

To see if the job attrition can be affected by the total time the individuals may have to spend for in and outside the office associated with work, the following columns are analyzed:
(1) 'DistanceFromHome'
(2) 'MaritalStatus',
(3) 'OverTime',
(4) 'BusinessTravel'

Note: MaritalStatus is included in this analysis based on the possibility that whether individual is single, married, or divorced, may play a role in the job attrtion especially if one must not only travel but also have to have over time, and/or commuting distance is long. Or, it could be the case that the single ones are more likely to have job attrtitions based on the assumption that they do not have financial obligations to thier families and children, allowing them to feel more at ease about resining. Either way, this column seems to be an interesting one to analyzed together.

The following code is to categorize commuting distances ranging from 1 to 29 to make analysis and subsequent visualization easier:

```
In [62]: step = 5
         for start in range(0, attrition_data['DistanceFromHome'].max(), step):
           # 0 2500 5000
           rows = (attrition_data['DistanceFromHome'] >= start) & (attrition_data['DistanceFro
           if start + step > 24:
             attrition_data.loc[rows, 'DistanceRange'] = 25
           else:
             attrition_data.loc[rows, 'DistanceRange'] = start + step
         attrition_data.head()
```

```
Out[62]:                 Age Attrition     BusinessTravel  DailyRate  \
         EmployeeNumber
         1                41       Yes      Travel_Rarely       1102
         2                49        No  Travel_Frequently        279
         4                37       Yes      Travel_Rarely       1373
         5                33        No  Travel_Frequently       1392
         7                27        No      Travel_Rarely        591
```

```
                          Department  DistanceFromHome  Education  \
EmployeeNumber
1                              Sales                 1          2
2               Research & Development                 8          1
4               Research & Development                 2          2
5               Research & Development                 3          4
7               Research & Development                 2          1

               EducationField  EmployeeCount  EnvironmentSatisfaction  \
EmployeeNumber
1               Life Sciences              1                        2
2               Life Sciences              1                        3
4                       Other              1                        4
5               Life Sciences              1                        4
7                     Medical              1                        1

                    ...      TotalWorkingYears  TrainingTimesLastYear  \
EmployeeNumber      ...
1                   ...                      8                      0
2                   ...                     10                      3
4                   ...                      7                      3
5                   ...                      8                      3
7                   ...                      6                      3

                WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \
EmployeeNumber
1                             1               6                  4
2                             3              10                  7
4                             3               0                  0
5                             3               8                  7
7                             3               2                  2

                YearsSinceLastPromotion YearsWithCurrManager  \
EmployeeNumber
1                                     0                    5
2                                     1                    7
4                                     0                    0
5                                     3                    0
7                                     2                    2

                MonthlyIncomeBracket  attrition  DistanceRange
EmployeeNumber
1                             7500.0          1            5.0
2                             7500.0          0           10.0
4                             2500.0          1            5.0
5                             5000.0          0            5.0
7                             5000.0          0            5.0
```

```
[5 rows x 37 columns]
```

The following is the summary of the selected columns, in which 'DistanceRange' is a new column as the result of the above code to categorize 'DistanceFromHome'.

```
In [63]: # Hours of Commitment's relevant colums in a pivot table:
         attrition_pivot = attrition_data.pivot_table(index=['OverTime', 'BusinessTravel', 'Di
                                                      columns=['Gender','Attrition'], values='
                                                      aggfunc=lambda x: len(x) if len(x) != np
         attrition_pivot
```

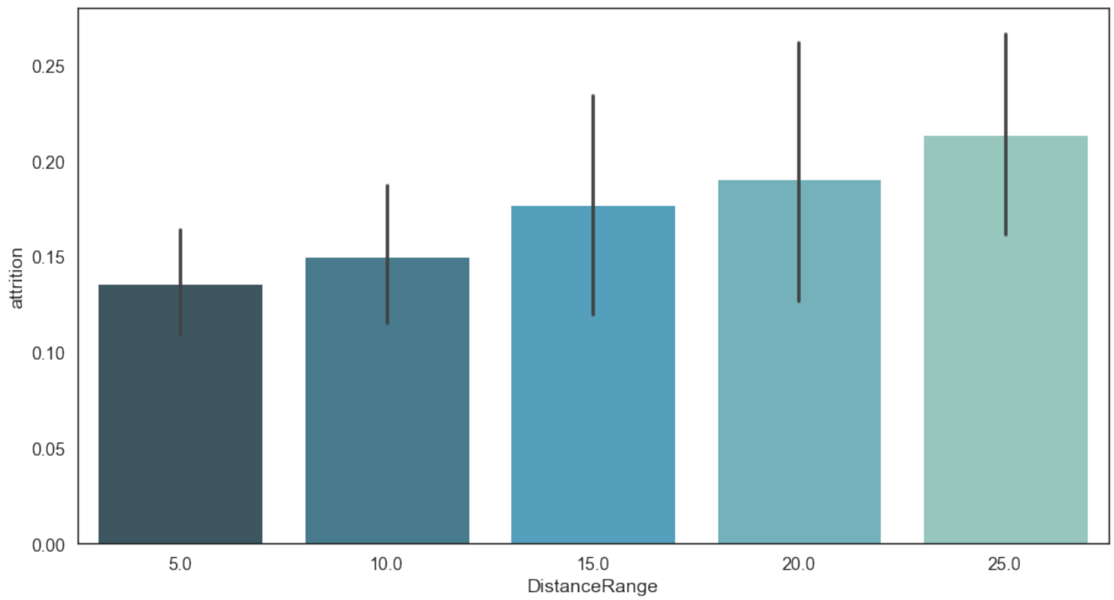
```
Out[63]: Gender                                                    Female         Male  \
         Attrition                                                     No   Yes      No
         OverTime BusinessTravel    DistanceRange MaritalStatus
         No       Non-Travel        5.0           Divorced            2.0   NaN    10.0
                                                  Married             8.0   NaN    17.0
                                                  Single              4.0   NaN     3.0
                                    10.0          Divorced            1.0   NaN    10.0
                                                  Married             4.0   NaN     3.0
                                                  Single              2.0   NaN    11.0
                                    15.0          Divorced            1.0   NaN     NaN
                                                  Married             2.0   NaN     2.0
                                                  Single              1.0   NaN     3.0
                                    20.0          Divorced            1.0   NaN     2.0
                                                  Married             NaN   NaN     1.0
                                                  Single              NaN   NaN     4.0
                                    25.0          Divorced            2.0   NaN     2.0
                                                  Married             4.0   NaN     5.0
                                                  Single              1.0   NaN     4.0
                  Travel_Frequently 5.0           Divorced            7.0   NaN     7.0
                                                  Married            11.0   NaN    18.0
                                                  Single              9.0   5.0    13.0
                                    10.0          Divorced            1.0   2.0     8.0
                                                  Married             8.0   NaN    11.0
                                                  Single              4.0   1.0     2.0
                                    15.0          Divorced            3.0   NaN     5.0
                                                  Married             3.0   NaN     6.0
                                                  Single              2.0   2.0     3.0
                                    20.0          Divorced            1.0   NaN     1.0
                                                  Married             2.0   NaN     3.0
                                                  Single              2.0   NaN     2.0
                                    25.0          Divorced            3.0   NaN     2.0
                                                  Married             8.0   2.0     6.0
                                                  Single              6.0   2.0     1.0
         ...                                                          ...   ...     ...
         Yes      Travel_Frequently 5.0           Married             3.0   NaN     7.0
                                                  Single              4.0   3.0     6.0
                                    10.0          Divorced            1.0   NaN     4.0
```

```
                                      Married          NaN   3.0    3.0
                                      Single           1.0   3.0    1.0
                        15.0          Divorced         NaN   NaN    2.0
                                      Married          NaN   NaN    1.0
                                      Single           1.0   1.0    NaN
                        20.0          Divorced         NaN   1.0    1.0
                                      Married          NaN   1.0    3.0
                                      Single           NaN   1.0    NaN
                        25.0          Divorced         1.0   2.0    NaN
                                      Married          3.0   NaN    3.0
                                      Single           1.0   1.0    1.0
         Travel_Rarely  5.0           Divorced         9.0   3.0    6.0
                                      Married         15.0   3.0   22.0
                                      Single          14.0   5.0   10.0
                        10.0          Divorced         9.0   1.0   14.0
                                      Married         13.0   2.0   12.0
                                      Single           5.0   2.0    6.0
                        15.0          Divorced         7.0   NaN    3.0
                                      Married         10.0   2.0   10.0
                                      Single           2.0   3.0    1.0
                        20.0          Divorced         2.0   NaN    3.0
                                      Married          5.0   NaN    5.0
                                      Single           3.0   2.0    NaN
                        25.0          Divorced         1.0   NaN    2.0
                                      Married          6.0   3.0   12.0
                                      Single           2.0   2.0    2.0
All                                                  501.0  87.0  732.0

Gender                                                       All
Attrition                                              Yes
OverTime BusinessTravel    DistanceRange MaritalStatus
No       Non-Travel        5.0           Divorced      NaN    12
                                         Married       NaN    25
                                         Single        NaN     7
                           10.0          Divorced      1.0    12
                                         Married       NaN     7
                                         Single        1.0    14
                           15.0          Divorced      NaN     1
                                         Married       1.0     5
                                         Single        NaN     4
                           20.0          Divorced      NaN     3
                                         Married       NaN     1
                                         Single        NaN     4
                           25.0          Divorced      NaN     4
                                         Married       NaN     9
                                         Single        2.0     7
         Travel_Frequently 5.0           Divorced      3.0    17
                                         Married       1.0    30
```

```
                                         Single            4.0    31
                           10.0          Divorced          NaN    11
                                         Married           3.0    22
                                         Single            1.0     8
                           15.0          Divorced          1.0     9
                                         Married           1.0    10
                                         Single            NaN     7
                           20.0          Divorced          NaN     2
                                         Married           1.0     6
                                         Single            NaN     4
                           25.0          Divorced          NaN     5
                                         Married           1.0    17
                                         Single            3.0    12
...                                                        ...   ...
Yes      Travel_Frequently 5.0           Married           3.0    13
                                         Single            3.0    16
                           10.0          Divorced          NaN     5
                                         Married           1.0     7
                                         Single            3.0     8
                           15.0          Divorced          NaN     2
                                         Married           NaN     1
                                         Single            2.0     4
                           20.0          Divorced          1.0     3
                                         Married           1.0     5
                                         Single            NaN     1
                           25.0          Divorced          1.0     4
                                         Married           1.0     7
                                         Single            2.0     5
         Travel_Rarely     5.0           Divorced          NaN    18
                                         Married           8.0    48
                                         Single            8.0    37
                           10.0          Divorced          2.0    26
                                         Married           5.0    32
                                         Single            6.0    19
                           15.0          Divorced          3.0    13
                                         Married           1.0    23
                                         Single            3.0     9
                           20.0          Divorced          2.0     7
                                         Married           2.0    12
                                         Single            1.0     6
                           25.0          Divorced          1.0     4
                                         Married           5.0    26
                                         Single            9.0    15
All                                                      150.0  1470

[90 rows x 5 columns]
```

(1) Visualizing the relatipnship between "DistanceFromHome" and "Attrition": First, plotting DistanceRange and Attrtion in a simple bar graph.

```
In [64]: fig, ax = plt.subplots(1, figsize=(15, 8))
         sns.barplot(x='DistanceRange', y='attrition', data=attrition_data, palette="GnBu_d")
         plt.show()
```

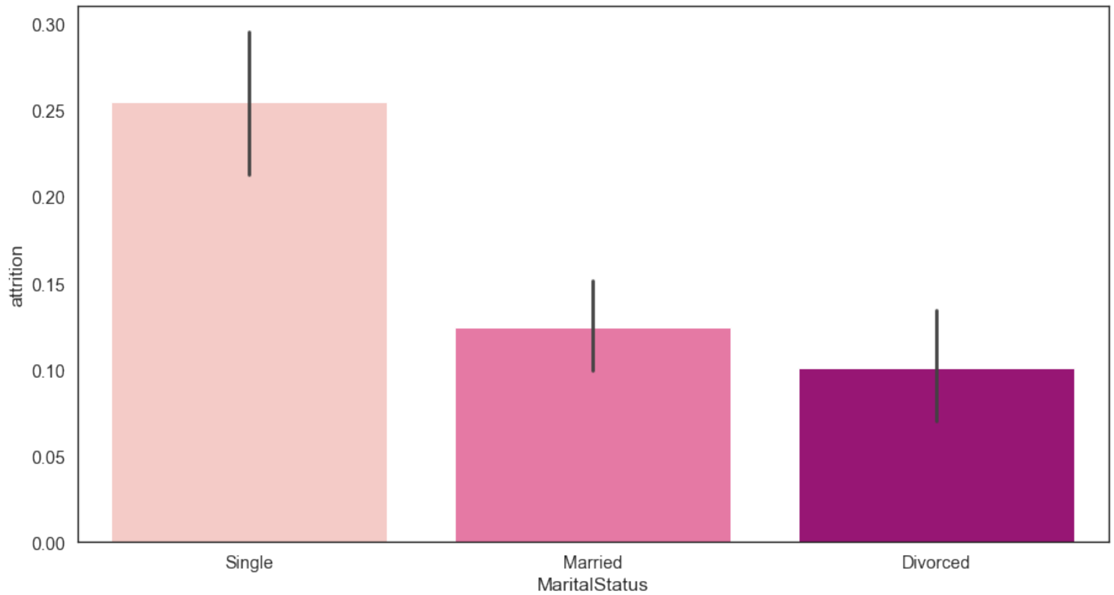


The bargrph above clearly indicates that the longer the distance, the higher the Attrition/Yes.

Nextly, similarly for the MaritalStatus and Attrition, a bargraph is plotted.

(2) Visualizing the relatipnship between "MaritalStatus" and "Attrition":

```
In [65]: fig, ax = plt.subplots(1, figsize=(15, 8))
         sns.barplot(x='MaritalStatus', y='attrition', data=attrition_data, palette="RdPu")
         plt.show()
```

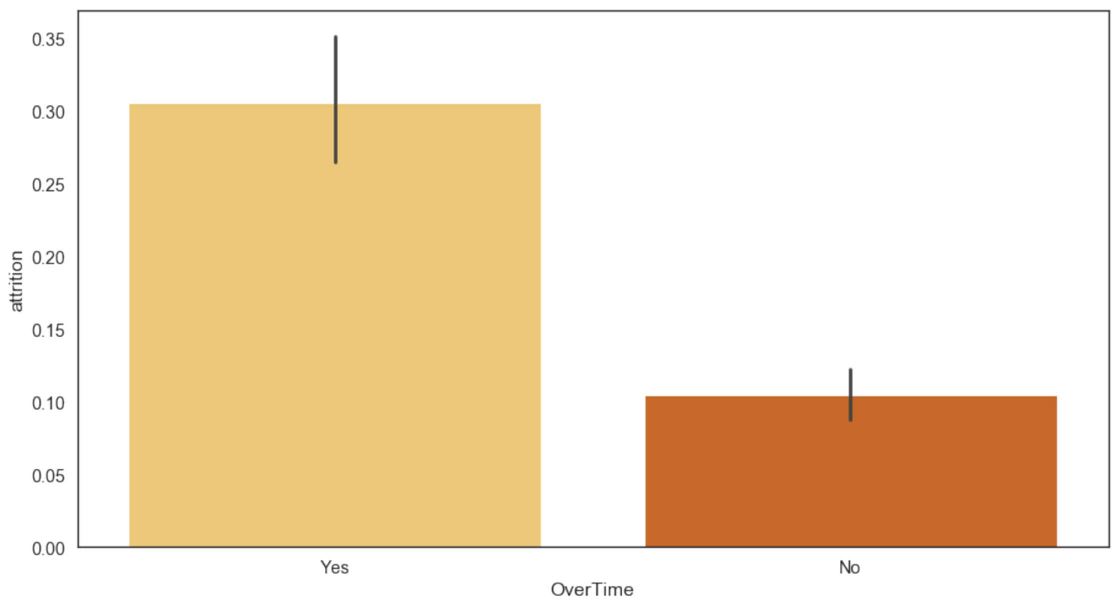


The bargraph above shows that the Single group is more likely to lead to job attrition with much higher number, little over 0.25, than the ones for Married, and Divorced groups. This may be because of the fact that, as brieflymentioned earlier, the singles are likely to have less obligations to family and children.
Comparing the Single and the Divorced, one may think that why they are strikingly so different although divorced means they are single in a general categorical sense. However, the difference between the Single and the Divorced may be large because of the potential financial burdens that the Divorced group may have to their former spourses, and especially to their children. Compared to the Married group, the Divorced group cannot enjoy the tax benefits of filing tax jointly, which is often done by married couples. The Divorced group is also more vulnerable to less healthcare benefits compared to its married counterparts. With the aforesaid potential financial burdens, the tax, the healthcare, and possibly more, it is after costly to live the life alone while being fiancially responsible for others. Whatever the true cauases may be, this is an interesting graph to see.

---

(3) Visualizing the relatipnship between "OverTime" and "Attrition":

```
In [66]: fig, ax = plt.subplots(1, figsize=(15, 8))
         sns.barplot(x='OverTime', y='attrition', data=attrition_data, palette="YlOrBr")
         plt.show()
```



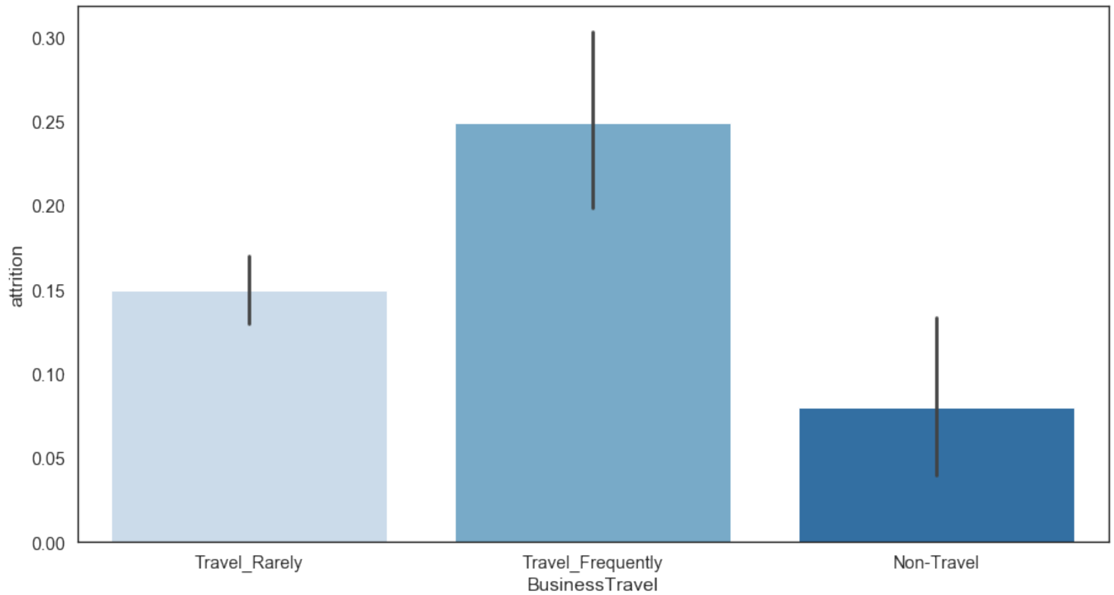
The bar graph above shows a visually striking difference between the Attriton/Yes and Attrtion/No group, where the former, the Attrition/Yes group is approximately three times larger than its counterpart, the Attrtion/No group.

Based on the p-value, 1.0e-21, there is a strong evidence against the nulll hypothesis, meaning that OverTime does affect Attrition.

---

(4) Visualizing the relatipnship between "BusinessTravel" and "Attrition":

```
In [67]: fig, ax = plt.subplots(1, figsize=(15, 8))
         sns.barplot(x='BusinessTravel', y='attrition', data=attrition_data, palette="Blues")
         plt.show()
```



The bar graph above also shows a trend that the more one travels, the more the individual is likely to have Attrtion/Yes.

Finally, One-Way ANOVA will be performd to test the following hypotheses for each column analyzed above:

(1) The DistanceFromHome:
H0: The commuting distance has no effect in job attrition.
H1: The commuting distance does affect job attrition (H0 is not True).

(2) MaritalStatus:
H0: The marital status has no effect in job attrition.
H1: The marital status distance does affect job attrition (H0 is not True).

(3) OverTime:
H0: The over time has no effect in job attrition.
H1: The over time does affect job attrition (H0 is not True).

(4) BusinessTravel:
H0: The business travel frequency has no effect in job attrition.
H1: The business travel frequency does affect job attrition (H0 is not True).

First, testing for (1) The DistanceFromHome:
H0: The commuting distance has no effect in job attrition.
H1: The commuting distance does affect job attrition (H0 is not True).

In [68]:
```
distance = pd.crosstab(attrition_data['Attrition'], attrition_data['DistanceRange'],
                       margins=True, normalize=True)
distance * 100
```

Out[68]:
```
DistanceRange        5.0       10.0       15.0      20.0       25.0  \
Attrition
No             33.333333  21.564626   9.795918  6.938776  12.244898
Yes             5.238095   3.809524   2.108844  1.632653   3.333333
All            38.571429  25.374150  11.904762  8.571429  15.578231

DistanceRange         All
Attrition
No              83.877551
Yes             16.122449
All            100.000000
```

In [69]:
```
distances = []
mind = int(attrition_data['DistanceRange'].min())
maxd = int(attrition_data['DistanceRange'].max())
for d in range(mind, maxd + 1, 5):
  dist = attrition_data.loc[attrition_data['DistanceRange'] == d, 'attrition']
  distances.append(dist)
  # anova needs a list to be paassed onto
print(len(distances))
stats.f_oneway(*distances)
```

```
5
```

Out[69]:
```
F_onewayResult(statistic=2.227412086976853, pvalue=0.0639418609871053)
```

The p-value=0.063, meaning that I cannot reject the null hypothesis.

This was an unintuitive result for me as I would have guessed that the longer the travel, the more attrtion is likely. That said, a potential explation to this is that even the longest 29 in the dataset the one must ravel (assuming it is in miles) may not be considered too much of a commute after all. This is because for the country such as the United States, 29miles are not a long distance if there is a highway access. This also is relevant whether the company is located in a city, where there is a constant heavy traffic, or located in a rural area with no traffic at all.
Either way, it is an interesting result.

Next, testing for (2) MaritalStatus:
H0: The marital status has no effect in job attrition.
H1: The marital status distance does affect job attrition (H0 is not True).

In [70]:
```
mss = []
for val in attrition_data['MaritalStatus'].unique():
  ms = attrition_data.loc[attrition_data['MaritalStatus'] == val, 'attrition']
```

```
        mss.append(ms)

    print(len(mss))
    stats.f_oneway(*mss)
```

3

Out[70]: F_onewayResult(statistic=23.78156546845813, pvalue=6.850067559825624e-11)

Based on the p-value=6.85e-11, and I can safely reject the null hypothesis, meaning that H1 is true: The marital status distance does affect job attrition. For the potential reasons discussed earlier, they may be a variety of causes to why MaritalStatus does affect job attrition. As the Single group is the most likely to have job attrition, this may be related to Age rather than the MaritalStatus in itself.

The next, testing for (3) OverTime:
H0: The over time has no effect in job attrition.
H1: The over time does affect job attrition (H0 is not True).

In [71]:
```
ots = []
for val in attrition_data['OverTime'].unique():
    ot = attrition_data.loc[attrition_data['OverTime'] == val, 'attrition']
    ots.append(ot)
    # anova needs a list to be paassed onto
print(len(ots))
stats.f_oneway(*ots)
```

2

Out[71]: F_onewayResult(statistic=94.65645707175152, pvalue=1.0092540336562444e-21)

Based on the very low pvalue=1.0092540336562444e-21, I can safely reject the null hypothesis (H0: The over time has no effect in job attrition.), meaning that the over time does affect job attrition. This is an intuitive result, as I know based on my own experience that too long of working hours can lead to 'burn out', leading to the workers to resign.

Finally, testing for (4) BusinessTravel:
H0: The business travel frequency has no effect in job attrition.
H1: The business travel frequency does affect job attrition (H0 is not True).

In [72]:
```
bts = []
for val in attrition_data['BusinessTravel'].unique():
    bt = attrition_data.loc[attrition_data['BusinessTravel'] == val, 'attrition']
    bts.append(bt)

print(len(bts))
stats.f_oneway(*bts)
```

3

Out[72]: F_onewayResult(statistic=12.26835294184309, pvalue=5.1998333569549645e-06)

Based on the pvalue=5.1998333569549645e-06, I can safely reject the null hypothesis (the business travel frequency has no effect in job attrition.), meaning that the business travel frequency does affect job attrition. This is also an intuitive result, again based my own experiences. Traveling often can only enjoyable if they are for leisure. Being stuck at a meeting room outside one's own country, is the same as being stuck in the office at home.

7.0.4 4. Gender and Age Differences in Attrition:

(1) Gender and Attrition: First, a pivot table by count and crosstab by percentages are created as a summary view. To do so, I first created the below pivot table, which was later added with additional columns with percentages (no_percent, and yes_percent). The total_prcnt column is also created for more complete view.

```
In [73]: attrition_by_gender = attrition_data.pivot_table(index='Gender', columns='Attrition',
                                                           aggfunc='count', margins=True)
         attrition_by_gender
         # Add precentage so that it is easier to see the ratios --> Do later.
```

```
Out[73]: Attrition    No  Yes   All
         Gender
         Female      501   87   588
         Male        732  150   882
         All        1233  237  1470
```

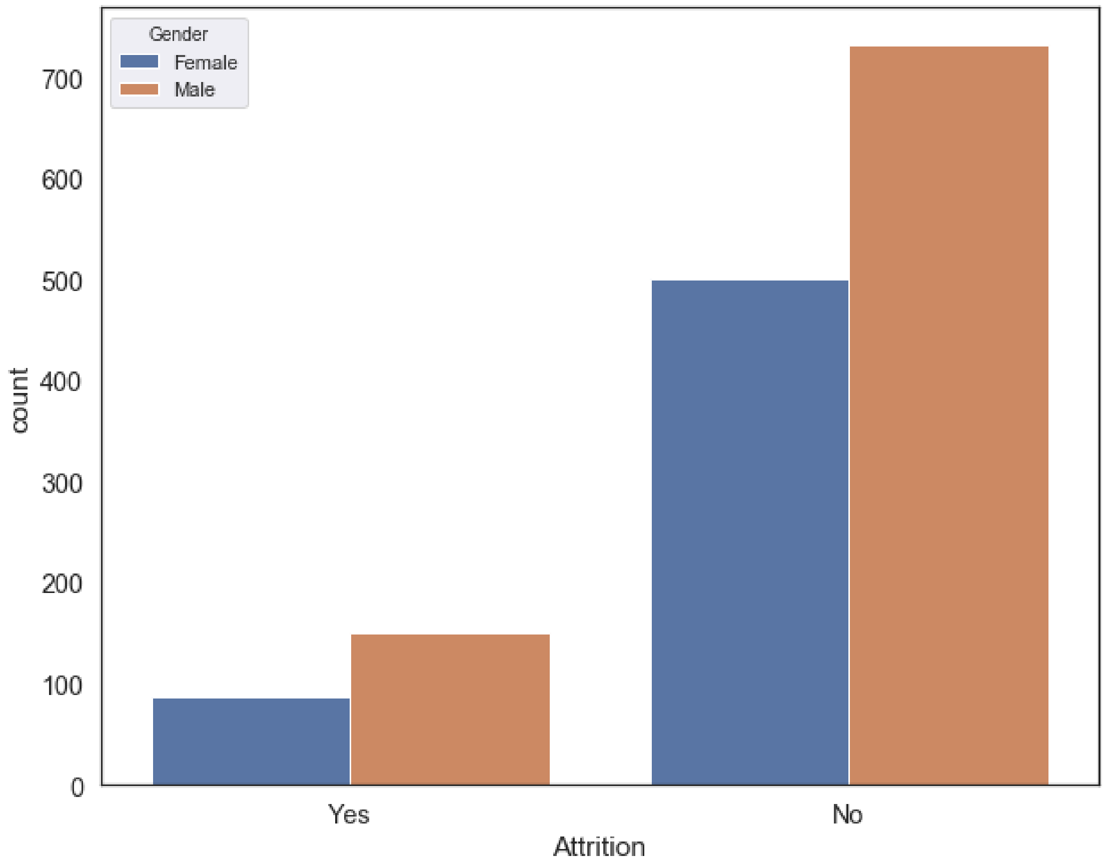
```
In [74]: gender = pd.crosstab(attrition_data['Gender'], attrition_data['Attrition'], margins=T
         gender * 100
```

```
Out[74]: Attrition         No        Yes    All
         Gender
         Female     34.081633   5.918367   40.0
         Male       49.795918  10.204082   60.0
         All        83.877551  16.122449  100.0
```

To visualize Gender and Attrition, the sns.countplot was used. This is because both the columns are for categorical values: Attrtion: Yes/No, and Gender: Female/Male.

```
In [75]: f, ax = plt.subplots(figsize=(10, 8))
         sns.set(style="darkgrid")
         sns.countplot(x='Attrition', hue="Gender", data=attrition_data)
```

```
Out[75]: <matplotlib.axes._subplots.AxesSubplot at 0x129964470>
```



First, as mentioned earlier, the ration in Attrtion (both genders combined) is Attrition/Yes with 16%, and Attrtion/No with 84%.

Taking the results form the crosstab above, in which Looking at the output above, the ratios for the Attrition/Yes and Attrition/No in respective gender are approximately 40% for female and 60% for male.

H0: The two categorical data (Gender and Attrition) 5.9% of Attrition/Yes is female whle 10.2 of Attrtion/Yes is male. This means that the out of 16%, the total Attrtion/Yes, about 37% are female while the rest, 63% are male, similar to the proportion of female:male, 4:6, included in the dataset. This suggests the following hypothesis:

H0: The two categories (Attrtion and Gender) are indepdent of each other. H1: The two categories (Attrtion and Gender) are not indepdent of each other (H0 is not True).

To test these categorical values, the following Chi-Square test is performed:

```
In [76]: # Insert your code here
         from scipy.stats import chi2_contingency
         chi2, p, dof, ex = chi2_contingency(attrition_by_gender)
         print("chi2 = ", chi2)
         print("p-val = ", p)
         print("degree of freedom = ",dof)
         print("Expected:")
         pd.DataFrame(ex)
```

```
### The code needs to be convered from numbers to string values.
```

```
chi2 =  1.2752163602205182
p-val =  0.8655661914618858
degree of freedom =  4
Expected:
```

```
Out[76]:         0      1       2
         0   493.2   94.8   588.0
         1   739.8  142.2   882.0
         2  1233.0  237.0  1470.0
```

Looking at the p-val = 0.8655661914618858, there is a strong evidence for the null hypothesis (H0: the two categorical data (Gender and Attrition) are independent of each other. In other words, I cannot reject the null hypothesis.

This test result indicates that the potential gender difference I thought we may see is a chance, not due to a dependencies that exists between the two.

My prior was that there is a gender difference in job attrtion as I assumed that female working population is more vulnerable to it due to various social challgenges in and outside work women face. However, after the analysis and the test result, I can deduce that there is no apparent gender difference in attrition because of a self-selecting nature of career options women may be making. For instance, if a woman knows the most of the family obligations fall onto her shoulder (if she has a familly with children), she may not choose the type of work that is likely to make her more hectic to the level that she would end up leaving her workplace.
This means that she may not be choosing a type of work that is known to have a log of over time, business travels, etc.
Perhaps, this result is also an indication that this dataset is rick in nature, as many of the factors are hard to separate, interdependent on each other.

---

(2) Age and Attrition: Finally, Age and Attrition will be analyzed. But first, a new colum, AgeRange will be created to categorize the age data in 'Age' column, which ranges between 18 and 60. The ragnes are divided by 5-year interval, except the first age range (18-25).

```
In [77]: step = 5
         for start in range(25, attrition_data['Age'].max(), step):
           age_rows = (attrition_data['Age'] >= start) & (attrition_data['Age'] <= start + step
           if start + step > 59:
             attrition_data.loc[age_rows, 'Age'] = 60
           else:
             attrition_data.loc[age_rows, 'AgeRange'] = start + step
         attrition_data.loc[attrition_data['Age'] <= 25, 'AgeRange'] = 25

         attrition_data.head()
```

```
Out[77]:                 Age Attrition     BusinessTravel  DailyRate  \
         EmployeeNumber
         1                41       Yes      Travel_Rarely       1102
         2                49        No  Travel_Frequently        279
         4                37       Yes      Travel_Rarely       1373
         5                33        No  Travel_Frequently       1392
         7                27        No      Travel_Rarely        591

                                     Department  DistanceFromHome  Education  \
         EmployeeNumber
         1                                Sales                 1          2
         2               Research & Development                 8          1
         4               Research & Development                 2          2
         5               Research & Development                 3          4
         7               Research & Development                 2          1

                        EducationField  EmployeeCount  EnvironmentSatisfaction  \
         EmployeeNumber
         1               Life Sciences              1                        2
         2               Life Sciences              1                        3
         4                       Other              1                        4
         5               Life Sciences              1                        4
         7                     Medical              1                        1

                           ...    TrainingTimesLastYear  WorkLifeBalance  \
         EmployeeNumber    ...
         1                 ...                        0                1
         2                 ...                        3                3
         4                 ...                        3                3
         5                 ...                        3                3
         7                 ...                        3                3

                         YearsAtCompany  YearsInCurrentRole YearsSinceLastPromotion  \
         EmployeeNumber
         1                            6                   4                       0
         2                           10                   7                       1
         4                            0                   0                       0
         5                            8                   7                       3
         7                            2                   2                       2

                         YearsWithCurrManager MonthlyIncomeBracket  attrition  \
         EmployeeNumber
         1                                  5               7500.0          1
         2                                  7               7500.0          0
         4                                  0               2500.0          1
         5                                  0               5000.0          0
         7                                  2               5000.0          0
```

```
                 DistanceRange  AgeRange
EmployeeNumber
1                          5.0      45.0
2                         10.0      50.0
4                          5.0      40.0
5                          5.0      35.0
7                          5.0      30.0

[5 rows x 38 columns]
```

Using the newly created column, 'AgeRange', the following pivot table with percentages is created as an overview.

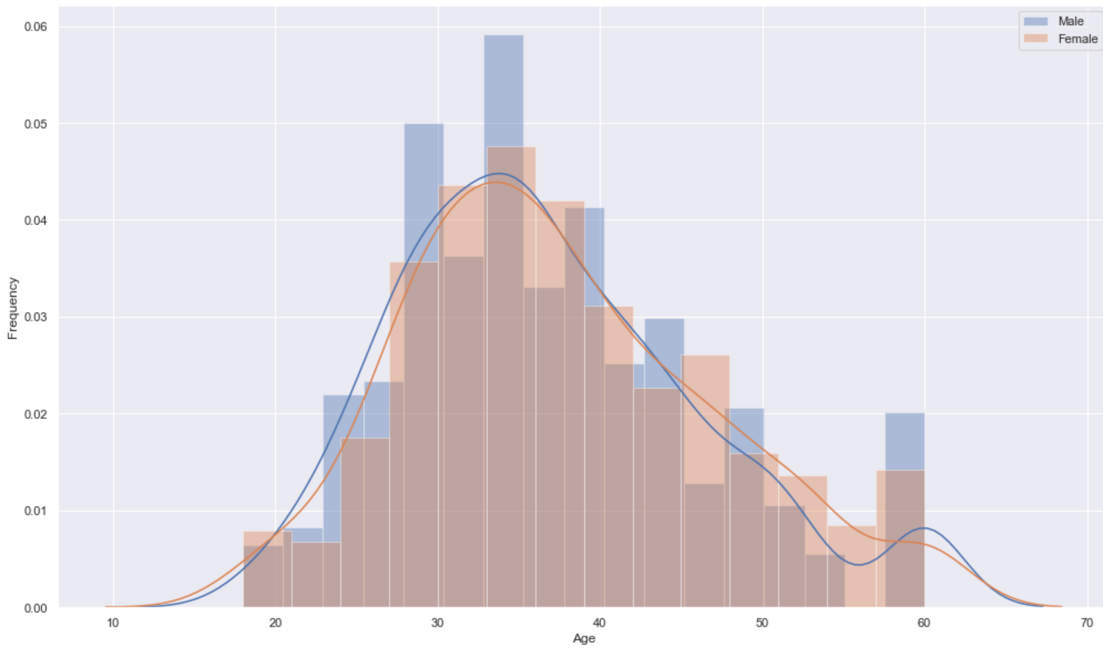
```
In [78]: attrition_by_age = attrition_data.pivot_table(index='AgeRange', columns='Attrition', v
                                                        aggfunc='count', margins=True)
```

```
In [79]: attrition_by_age = attrition_by_age.rename(index={'All': 'Total'}, columns={'All': 'To
         attrition_by_age.loc[:, 'no_prcnt'] = (attrition_by_age.loc[:, 'No'] / attrition_by_ag
         attrition_by_age.loc[:, 'yes_prcnt'] = (attrition_by_age.loc[:, 'Yes'] / attrition_by_
         attrition_by_age.loc[:, 'total_prcnt'] = (attrition_by_age.loc[:, 'Total'] / attritio
         attrition_by_age
```

```
Out[79]: Attrition    No  Yes  Total   no_prcnt  yes_prcnt  total_prcnt
         AgeRange
         25.0         79   44    123  64.227642  35.772358        100.0
         30.0        156   47    203  76.847291  23.152709        100.0
         35.0        266   59    325  81.846154  18.153846        100.0
         40.0        267   30    297  89.898990  10.101010        100.0
         45.0        187   21    208  89.903846  10.096154        100.0
         50.0        128   13    141  90.780142   9.219858        100.0
         55.0        111   15    126  88.095238  11.904762        100.0
         Total      1194  229   1423  83.907238  16.092762        100.0
```

First, a histogram with KDE for age distribution is plotted.

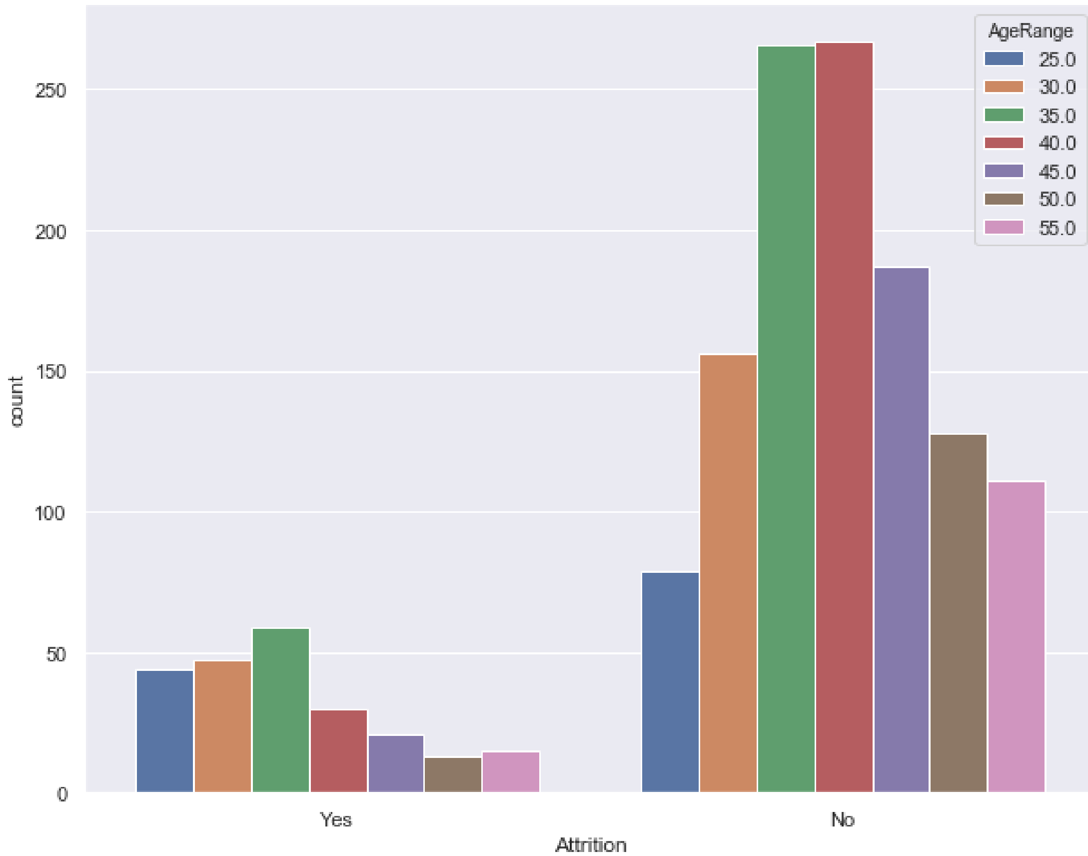
```
In [80]: # Plot Female and Male in same plot
         fig, axes = plt.subplots(1, 1, figsize=(17, 10))
         attrition_data_M = attrition_data.loc[attrition_data['Gender'] == 'Male']
         attrition_data_F = attrition_data.loc[attrition_data['Gender'] == 'Female']

         sns.distplot(attrition_data_M[['Age']], axlabel=None, label='Male')
         sns.distplot(attrition_data_F[['Age']], axlabel=None, label='Female')
         plt.legend()
         plt.xlabel('Age')
         plt.ylabel('Frequency')
         plt.show()
```



The above age distributions shows normal distributions for both genders, where the age range for the highest frequencies occuring are between the 30s and 40s.

The following countplot is plotted for AgeRage and Attrition.

```
In [81]: f, ax = plt.subplots(figsize=(10, 8))
         sns.set(style="darkgrid")
         sns.countplot(x="Attrition", hue="AgeRange", data=attrition_data)

Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x129685f60>
```



Though the y-axis is the total count for the each AgeRange, the graph for Attrition/No appears similar to the KDE normal curve for AgeRange plotted earlier. However, for the Attrition/Yes, the yonger age groups, the 20s and the 30s have higher counts than the older age groups, the 40s and above.

This is a curious result as an earlier discussion on the marital status and attrion mensions the higher attrtion for the Single group may be the result of the relationship between the age and attrition.

Based on this observation, I will hypothesize the following:

H0: Age has not effect in attrition.

H1: Age does have an effecgt in attrition.

To test the hypothesis, the following One-Way ANOVA is performed.

```
In [82]: ages = []
         mina = int(attrition_data['AgeRange'].min())
         maxa = int(attrition_data['AgeRange'].max())
         for a in range(mina, maxa + 1, 5):
           age = attrition_data.loc[attrition_data['AgeRange'] == a, 'attrition']
           ages.append(age)
```

```
  # anova needs a list to be paassed onto
print(len(ages))
stats.f_oneway(*ages)
```

7

Out[82]: F_onewayResult(statistic=11.07747452064328, pvalue=4.226676719467157e-12)

Based on the pvalue=4.226676719467157e-12, there is a strong evidence against the nulll hypothesis (H0: Age has not effect in attrition). Therefore, I can safely reject the null hypothesis and accept the alternative hypothesis, H1: Age does have an effecgt in attrition.

This concludes the analysis of the four exploratory question.

Next, to see if any similar results in be obtianed, a correlation heatmap is created for the numerical data.

7.0.5 Visualizing Correlations for numerical data:

Creating a heatmap to show the correaltions within the number_data:

In [83]:
```
num_data_correlation = number_data.corr()
num_data_correlation
```

Out[83]:

	Age	DailyRate	DistanceFromHome	HourlyRate
Age	1.000000	0.010661	-0.001686	0.024287
DailyRate	0.010661	1.000000	-0.004985	0.023381
DistanceFromHome	-0.001686	-0.004985	1.000000	0.031131
HourlyRate	0.024287	0.023381	0.031131	1.000000
JobLevel	0.509604	0.002966	0.005303	-0.027853
MonthlyIncome	0.497855	0.007707	-0.017014	-0.015794
MonthlyRate	0.028051	-0.032182	0.027473	-0.015297
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.022157
PercentSalaryHike	0.003634	0.022704	0.040235	-0.009062
StockOptionLevel	0.037510	0.042143	0.044872	0.050263
TotalWorkingYears	0.680381	0.014515	0.004628	-0.002334
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.008548
YearsAtCompany	0.311309	-0.034055	0.009508	-0.019582
YearsInCurrentRole	0.212901	0.009932	0.018845	-0.024106
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	-0.026716
YearsWithCurrManager	0.202089	-0.026363	0.014406	-0.020123

	JobLevel	MonthlyIncome	MonthlyRate
Age	0.509604	0.497855	0.028051
DailyRate	0.002966	0.007707	-0.032182

```
DistanceFromHome          0.005303      -0.017014     0.027473
HourlyRate               -0.027853      -0.015794    -0.015297
JobLevel                  1.000000       0.950300     0.039563
MonthlyIncome             0.950300       1.000000     0.034814
MonthlyRate               0.039563       0.034814     1.000000
NumCompaniesWorked        0.142501       0.149515     0.017521
PercentSalaryHike        -0.034730      -0.027269    -0.006429
StockOptionLevel          0.013984       0.005408    -0.034323
TotalWorkingYears         0.782208       0.772893     0.026442
TrainingTimesLastYear    -0.018191      -0.021736     0.001467
YearsAtCompany            0.534739       0.514285    -0.023655
YearsInCurrentRole        0.389447       0.363818    -0.012815
YearsSinceLastPromotion   0.353885       0.344978     0.001567
YearsWithCurrManager      0.375281       0.344079    -0.036746

                         NumCompaniesWorked  PercentSalaryHike  \
Age                                0.299635           0.003634
DailyRate                          0.038153           0.022704
DistanceFromHome                  -0.029251           0.040235
HourlyRate                         0.022157          -0.009062
JobLevel                           0.142501          -0.034730
MonthlyIncome                      0.149515          -0.027269
MonthlyRate                        0.017521          -0.006429
NumCompaniesWorked                 1.000000          -0.010238
PercentSalaryHike                 -0.010238           1.000000
StockOptionLevel                   0.030075           0.007528
TotalWorkingYears                  0.237639          -0.020608
TrainingTimesLastYear             -0.066054          -0.005221
YearsAtCompany                    -0.118421          -0.035991
YearsInCurrentRole                -0.090754          -0.001520
YearsSinceLastPromotion           -0.036814          -0.022154
YearsWithCurrManager              -0.110319          -0.011985

                         StockOptionLevel  TotalWorkingYears  \
Age                              0.037510           0.680381
DailyRate                        0.042143           0.014515
DistanceFromHome                 0.044872           0.004628
HourlyRate                       0.050263          -0.002334
JobLevel                         0.013984           0.782208
MonthlyIncome                    0.005408           0.772893
MonthlyRate                     -0.034323           0.026442
NumCompaniesWorked               0.030075           0.237639
PercentSalaryHike                0.007528          -0.020608
StockOptionLevel                 1.000000           0.010136
TotalWorkingYears                0.010136           1.000000
TrainingTimesLastYear            0.011274          -0.035662
YearsAtCompany                   0.015058           0.628133
YearsInCurrentRole               0.050818           0.460365
```

YearsSinceLastPromotion	0.014352	0.404858
YearsWithCurrManager	0.024698	0.459188

	TrainingTimesLastYear	YearsAtCompany \
Age	-0.019621	0.311309
DailyRate	0.002453	-0.034055
DistanceFromHome	-0.036942	0.009508
HourlyRate	-0.008548	-0.019582
JobLevel	-0.018191	0.534739
MonthlyIncome	-0.021736	0.514285
MonthlyRate	0.001467	-0.023655
NumCompaniesWorked	-0.066054	-0.118421
PercentSalaryHike	-0.005221	-0.035991
StockOptionLevel	0.011274	0.015058
TotalWorkingYears	-0.035662	0.628133
TrainingTimesLastYear	1.000000	0.003569
YearsAtCompany	0.003569	1.000000
YearsInCurrentRole	-0.005738	0.758754
YearsSinceLastPromotion	-0.002067	0.618409
YearsWithCurrManager	-0.004096	0.769212

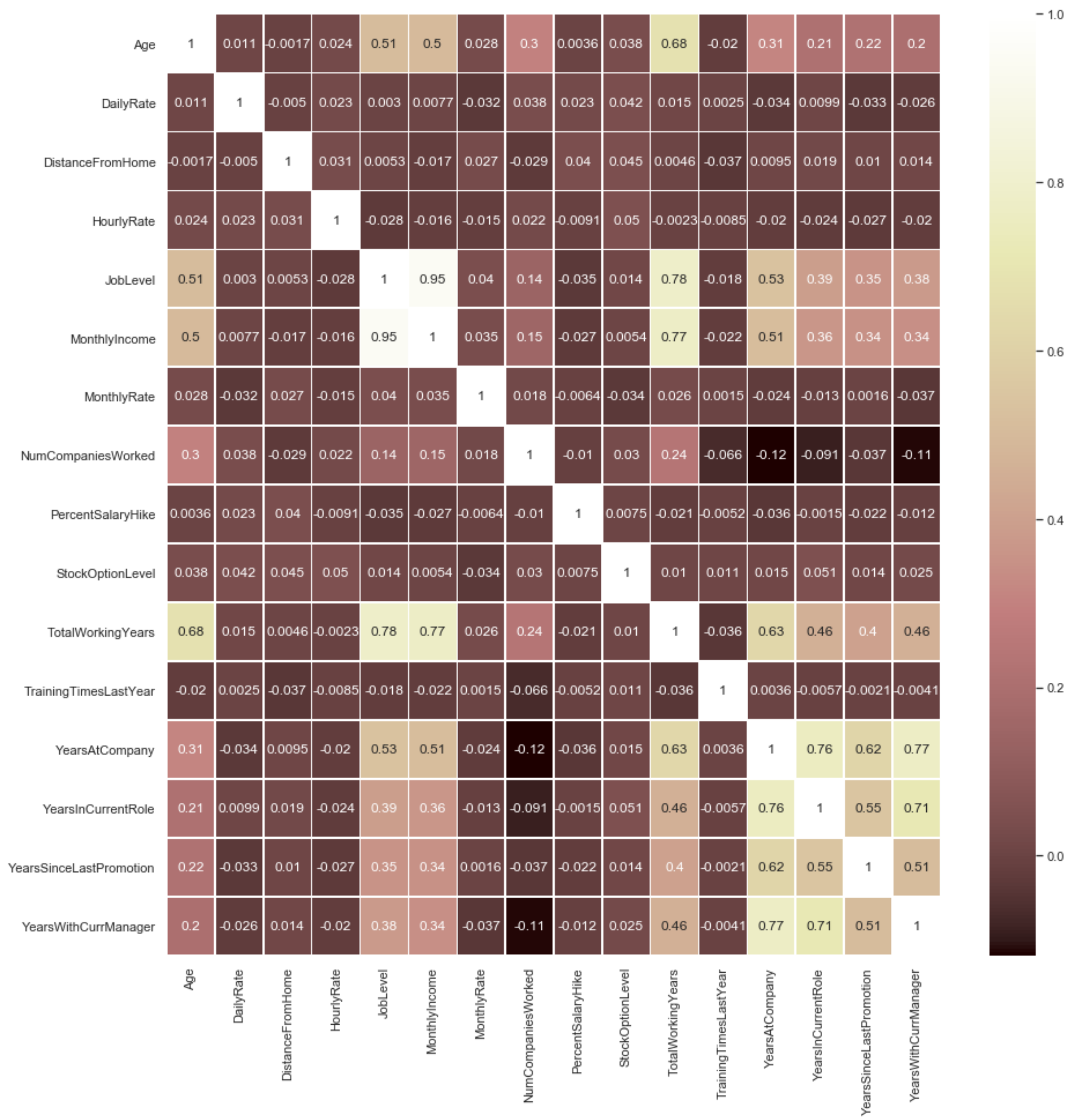
	YearsInCurrentRole	YearsSinceLastPromotion \
Age	0.212901	0.216513
DailyRate	0.009932	-0.033229
DistanceFromHome	0.018845	0.010029
HourlyRate	-0.024106	-0.026716
JobLevel	0.389447	0.353885
MonthlyIncome	0.363818	0.344978
MonthlyRate	-0.012815	0.001567
NumCompaniesWorked	-0.090754	-0.036814
PercentSalaryHike	-0.001520	-0.022154
StockOptionLevel	0.050818	0.014352
TotalWorkingYears	0.460365	0.404858
TrainingTimesLastYear	-0.005738	-0.002067
YearsAtCompany	0.758754	0.618409
YearsInCurrentRole	1.000000	0.548056
YearsSinceLastPromotion	0.548056	1.000000
YearsWithCurrManager	0.714365	0.510224

	YearsWithCurrManager
Age	0.202089
DailyRate	-0.026363
DistanceFromHome	0.014406
HourlyRate	-0.020123
JobLevel	0.375281
MonthlyIncome	0.344079
MonthlyRate	-0.036746
NumCompaniesWorked	-0.110319

```
PercentSalaryHike                   -0.011985
StockOptionLevel                     0.024698
TotalWorkingYears                    0.459188
TrainingTimesLastYear               -0.004096
YearsAtCompany                       0.769212
YearsInCurrentRole                   0.714365
YearsSinceLastPromotion              0.510224
YearsWithCurrManager                 1.000000
```

```
In [84]: f, ax = plt.subplots(figsize=(15, 15))
         num_data_correlation_map = sns.heatmap(num_data_correlation, annot=True, linewidths=.
         num_data_correlation_map      # Should I include 'StandardHours'???
```

```
Out[84]: <matplotlib.axes._subplots.AxesSubplot at 0x11dffe710>
```



The results:
The following is the list of some of the clumns in the order of highest correlations:
(1) JobLevel and MonthlyIncome (0.95)
(2) JobLevel and TotalWorkingYears (0.78)
(3) MonthlyIncome and TotalWorkingYears (0.77)
(4) YearsAtCompany and YearsWithCurrManager (0.77)
(5) YearsAtCompany and YearsInCurrentRole (0.76)
(6) YearsWithCurrManager and YearsInCurrentRole (0.71)
Analysis and interpretations:
* JobLevel is highly correlated to MonthlyIncome and TotalWorkingYears: This result is likely because that the longer you serve, the higher you are likely to be promoted. And higher your positon is, the more you are likely to earn as well. This intuitive result also explains the high correlation in (3) MonthlyIncome and TotalWorkingYears (0.77).

- Some of the "Years" columns are highly correlated with each other. For example, YearsAtCompany are highly correlated to YearsWithCurrentManager and YearsInCurrentRole. These correlations are interesting in that YearsAtCompany can be affected either positively and negatively. For example, if YearsAtCompany is long, it may also be the case that YearsWithCurrManager if you like working for your manager.
  But the opposite can be true and both YearsAtCompany and YearsWithCurrManager may be short if you do not like working for your manager. A similar case can be made the relationshp between YearsWithCurrManager and YearsInCurrentRole.

The important thing to note as result of this correlationmap is that there appears to be many interrelationships at work, which makes the analysis challenging.

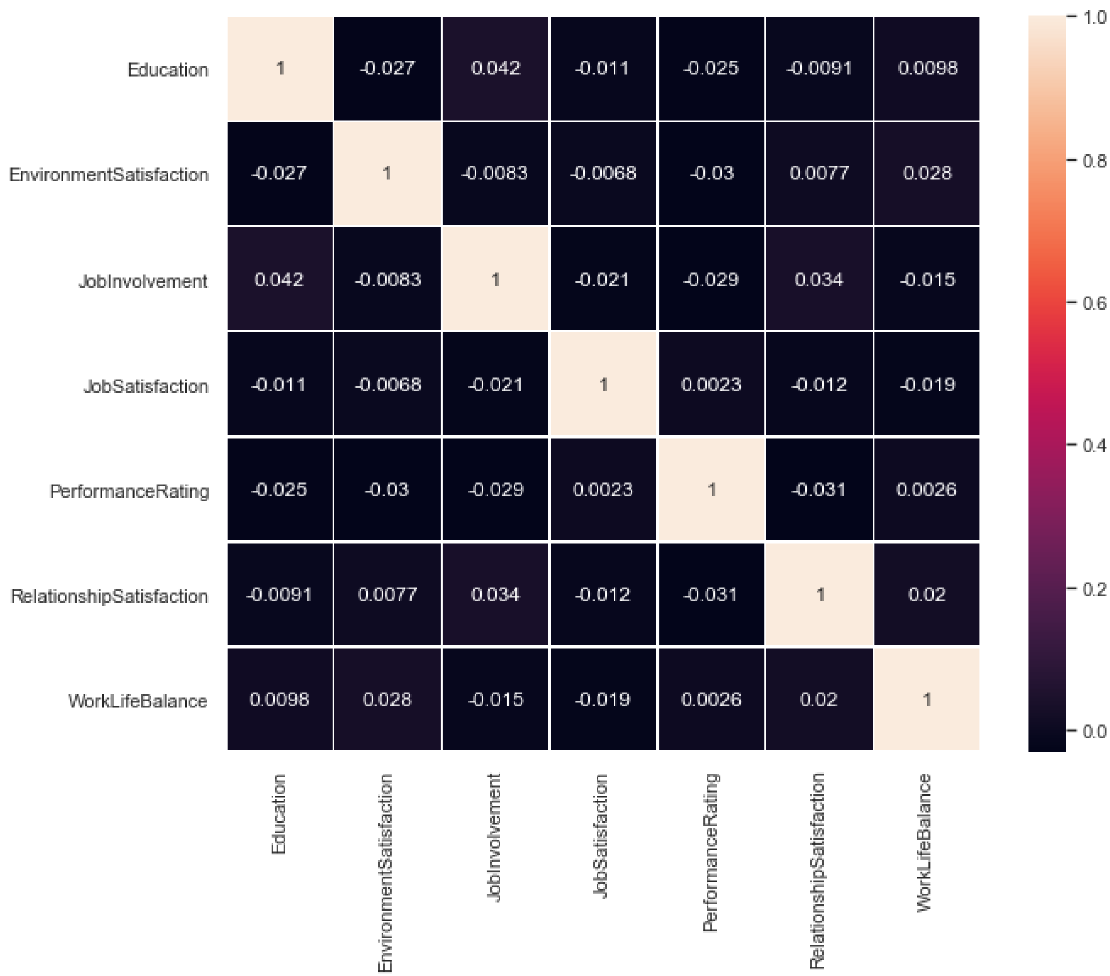
7.0.6 Investigating Correlation for categorical data:

Several columns have categorical data as follows:
Education: 1 'Below College' 2 'College' 3 'Bachelor' 4 'Master' 5 'Doctor'
EnvironmentSatisfaction: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'
JobInvolvement: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'
JobSatisfaction: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'
PerformanceRating: 1 'Low' 2 'Good' 3 'Excellent' 4 'Outstanding'
RelationshipSatisfaction: 1 'Low' 2 'Medium' 3 'High' 4 'Very High'
WorkLifeBalance: 1 'Bad' 2 'Good' 3 'Better' 4 'Best'

```
In [85]: categorical_data_correlation = categorical_data.corr()
```

```
In [86]: f, ax = plt.subplots(figsize=(10, 8))
         sns.heatmap(categorical_data_correlation, annot=True, linewidths=.5, ax=ax)
```

```
Out[86]: <matplotlib.axes._subplots.AxesSubplot at 0x129b410f0>
```



Result: The above map did not show anything interesting in temrs of correlations. This is because that the columns are categorical. The correlation mapping is not suiitable for the categorical data. As an experiment, the following K-Means Clustering is performed to see if it shows some interesting and visible clusters.

7.1 K-Means (Clustering)

This section experimental to see if a quick K-Means Clustering would yield any interesting clusters.

Process:
(1) Use the rule of the thumb to calculate the number of potential clusters.
(2) The elbow plot will be drawn.
(3) To analyze categorical data for K-Means Clustering, the column, 'Gender' Female/Male needs to be convered in 0 and 1 while Attrition column needs to be dropped for elbow plot.
The attrition column, 'Attrtion' column will be later convered to Yes:1 and No:0 and used

```
In [87]: ruleofthumb = np.sqrt(1470)
         ruleofthumb
```

```
Out[87]: 38.34057902536163
```

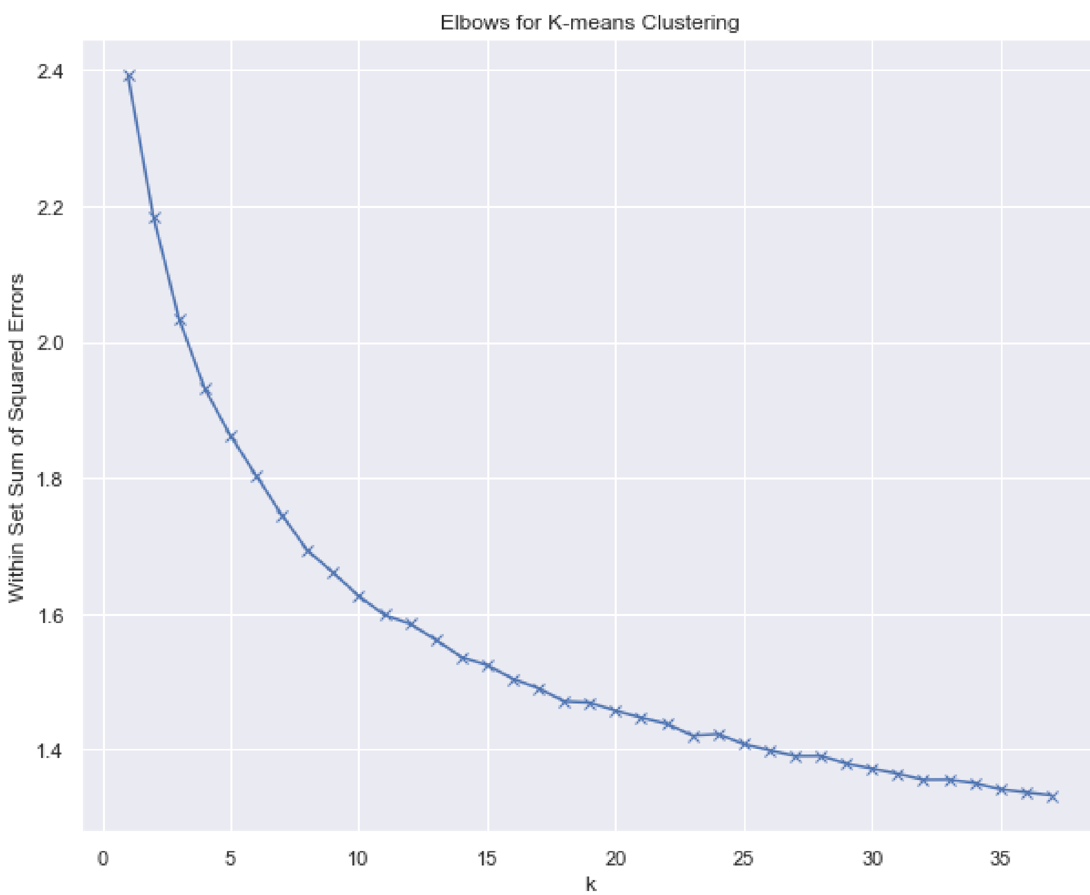
The rule of thum suggests that the number of cluster is 38.

```
In [88]: categorical_data.loc[:, 'Gender'] = categorical_data.replace({'Male': 1, 'Female': 0}
         dropped_attrition = categorical_data.drop('Attrition', axis=1)
         dropped_attrition.head()
```

```
Out[88]:                 Gender  Education  EnvironmentSatisfaction  JobInvolvement  \
         EmployeeNumber
         1                    0          2                        2               3
         2                    1          1                        3               2
         4                    1          2                        4               2
         5                    0          4                        4               3
         7                    1          1                        1               3

                         JobSatisfaction  PerformanceRating  RelationshipSatisfaction  \
         EmployeeNumber
         1                             4                  3                         1
         2                             2                  4                         4
         4                             3                  3                         2
         5                             3                  3                         3
         7                             2                  3                         4

                         WorkLifeBalance
         EmployeeNumber
         1                             1
         2                             3
         4                             3
         5                             3
         7                             3
```

```
In [89]: ### K-means may be sufficient.
         import sklearn as sk
         from sklearn import metrics

         # Insert your code here
         #sklearn.metrics.silhouette_score(X, labels, metric=euclidean
         #, sample_size=None, random_state=None, **kwds)


         from sklearn.cluster import KMeans
         from sklearn import metrics
         from scipy.spatial.distance import cdist
```

```python
import numpy as np
import matplotlib.pyplot as plt


#K-means: Derermine K:
distortions = []
K = range(1,38)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(dropped_attrition)
    distortions.append(sum(np.min(cdist(dropped_attrition, kmeanModel.cluster_centers_

# Elbow:
fig, axix = plt.subplots(figsize=(10, 8))
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Within Set Sum of Squared Errors')
plt.title('Elbows for K-means Clustering')
plt.show()
```



The above elbow plot suggests the 5 cluster, which will be used below. The resulting clusters with respective assigned cluster numbers are added in kmeans_2 column below.

In [90]:
```
dropped_attrition = categorical_data.drop('Attrition', axis=1)
dropped_attrition.head()
```

Out[90]:
```
                Gender  Education  EnvironmentSatisfaction  JobInvolvement  \
EmployeeNumber
1                    0          2                        2               3
2                    1          1                        3               2
4                    1          2                        4               2
5                    0          4                        4               3
7                    1          1                        1               3

                JobSatisfaction  PerformanceRating  RelationshipSatisfaction  \
EmployeeNumber
1                             4                  3                         1
2                             2                  4                         4
4                             3                  3                         2
5                             3                  3                         3
7                             2                  3                         4

                WorkLifeBalance
EmployeeNumber
1                             1
2                             3
4                             3
5                             3
7                             3
```

In [91]:
```
categorical_data.loc[:, 'Attrition'] = categorical_data.replace({'Yes': 1, 'No': 0})
```

In [92]:
```
#K-means: Derermine K:
kmeanModel = KMeans(n_clusters=5)
kmeanModel.fit(categorical_data)
categorical_data.loc[:, 'kmeans_2'] = kmeanModel.predict(categorical_data)
categorical_data.loc[:, 'Attrition'] = categorical_data['Attrition']
categorical_data.head(20)
```

Out[92]:
```
                Attrition  Gender  Education  EnvironmentSatisfaction  \
EmployeeNumber
1                       1       0          2                        2
2                       0       1          1                        3
4                       1       1          2                        4
5                       0       0          4                        4
7                       0       1          1                        1
8                       0       1          2                        4
10                      0       0          3                        3
11                      0       1          1                        4
```

```
12                    0       1           3                         4
13                    0       1           3                         3
14                    0       1           3                         1
15                    0       0           2                         4
16                    0       1           1                         1
18                    0       1           2                         2
19                    1       1           3                         3
20                    0       0           4                         2
21                    0       1           2                         1
22                    0       1           2                         4
23                    0       0           4                         1
24                    0       1           3                         4

                JobInvolvement  JobSatisfaction  PerformanceRating  \
EmployeeNumber
1                            3                4                  3
2                            2                2                  4
4                            2                3                  3
5                            3                3                  3
7                            3                2                  3
8                            3                4                  3
10                           4                1                  4
11                           3                3                  4
12                           2                3                  4
13                           3                3                  3
14                           4                2                  3
15                           2                3                  3
16                           3                3                  3
18                           3                4                  3
19                           2                3                  3
20                           4                1                  3
21                           4                2                  3
22                           4                4                  3
23                           2                4                  3
24                           3                4                  3

                RelationshipSatisfaction  WorkLifeBalance  kmeans_2
EmployeeNumber
1                                      1                1         4
2                                      4                3         1
4                                      2                3         4
5                                      3                3         3
7                                      4                3         0
8                                      3                2         3
10                                     1                2         2
11                                     2                3         4
12                                     2                3         4
13                                     2                2         4
```

14	3	3	0
15	4	3	3
16	4	2	0
18	3	3	0
19	2	3	4
20	3	3	1
21	4	2	0
22	2	2	4
23	3	3	0
24	3	3	3

Together with K-clustering, the following PCA is performed with the 3D visualized result.

```
In [93]: pca_model = skd.PCA().fit(dropped_attrition)
         pca_model.components_.shape
```

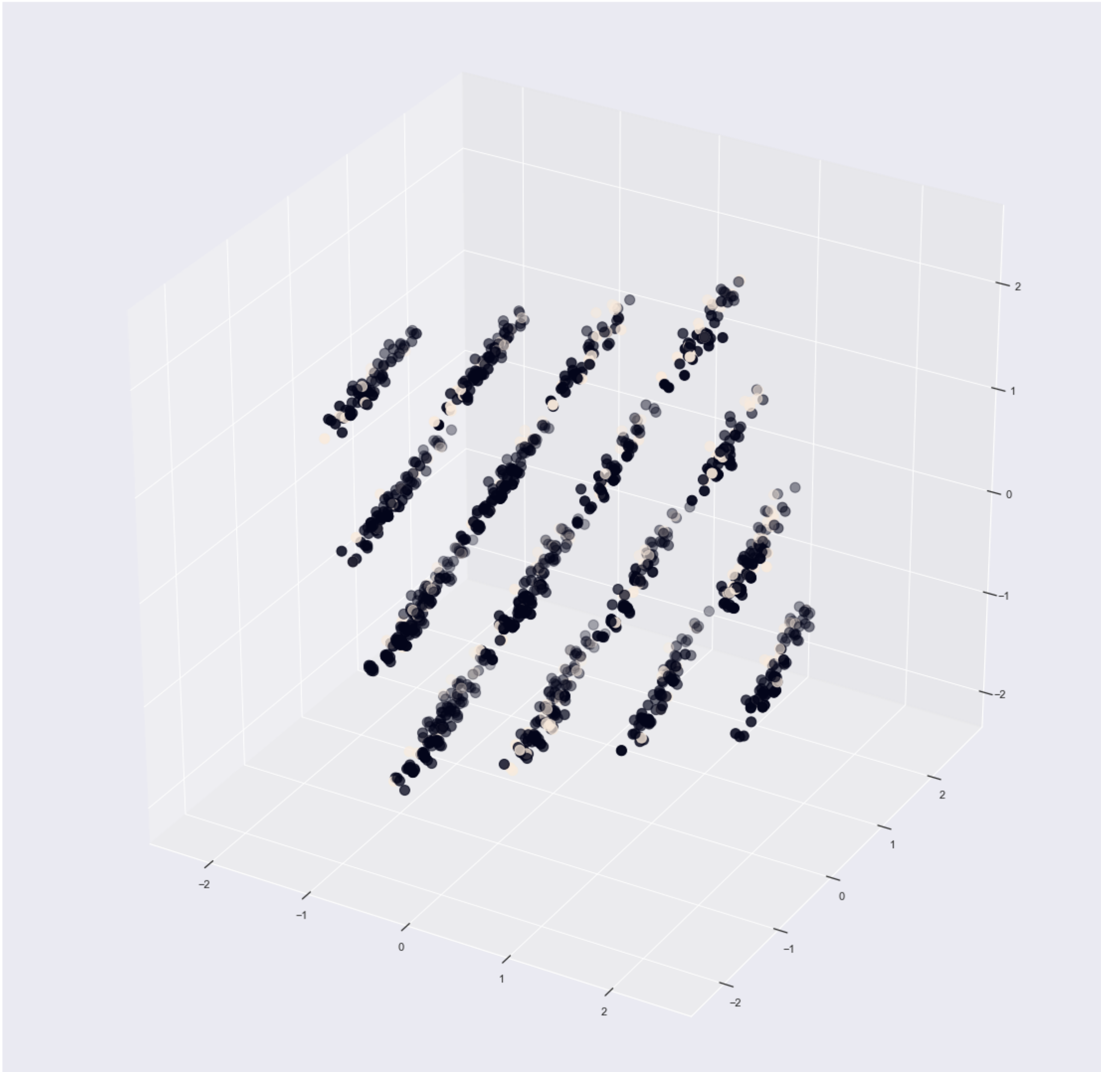
```
Out[93]: (8, 8)
```

```
In [94]: pca_model.explained_variance_
         # the first four variances below are explaiend well by pca.
```

```
Out[94]: array([1.22597269, 1.20092464, 1.16465519, 1.04299838, 0.50954755,
                0.49158964, 0.23947231, 0.12964874])
```

As the above results from .explained_varience_shows, the four variances are qualified (the numbers above 1.0) and explained well by PCA: 1.22597269, 1.20092464, 1.16465519, 1.04299838

```
In [95]: X = pca_model.transform(dropped_attrition)

         fig = plt.figure(figsize=(20, 20))
         ax = fig.add_subplot(111, projection='3d')
         ax.scatter(X[:,0], X[:,1], X[:,2], c=categorical_data['Attrition'].astype('category')

         for i, s in enumerate(categorical_data.index):
             x, y, _ = proj3d.proj_transform(X[i,0],X[i,1],X[i,2],
                                             ax.get_proj())
             #plt.annotate(s, xy=(x-0.005,y+0.002), fontsize=1)
```



Unfortunately, no apparent patterns were found, which may be due to the fact that the dataset has cerain complexies that could not have beene explained within the 3D space.

Next, as the final task, Random Forest Classification is performed to calculate the feature importance for Attrition.

7.2 Classifications

7.2.1 Random Forest to predict most important features 'Attrition':

```
In [96]: # First, perform random forest the way it is, and can perform with pd.get_dummies for
         from sklearn.ensemble import RandomForestClassifier

In [97]: rf_data = pd.concat([number_data, categorical_data], axis=1)
         rf_data.head()

Out[97]:                 Age  DailyRate  DistanceFromHome  HourlyRate  JobLevel  \
         EmployeeNumber
         1                41       1102                 1          94         2
         2                49        279                 8          61         2
         4                37       1373                 2          92         1
         5                33       1392                 3          56         1
         7                27        591                 2          40         1

                         MonthlyIncome  MonthlyRate  NumCompaniesWorked  \
         EmployeeNumber
         1                        5993        19479                   8
         2                        5130        24907                   1
         4                        2090         2396                   6
         5                        2909        23159                   1
         7                        3468        16632                   9

                         PercentSalaryHike  StockOptionLevel    ...     Attrition  \
         EmployeeNumber                                         ...
         1                              11                 0    ...             1
         2                              23                 1    ...             0
         4                              15                 0    ...             1
         5                              11                 0    ...             0
         7                              12                 1    ...             0

                         Gender  Education  EnvironmentSatisfaction  JobInvolvement  \
         EmployeeNumber
         1                    0          2                        2               3
         2                    1          1                        3               2
         4                    1          2                        4               2
         5                    0          4                        4               3
         7                    1          1                        1               3

                         JobSatisfaction  PerformanceRating  RelationshipSatisfaction  \
         EmployeeNumber
         1                             4                  3                         1
         2                             2                  4                         4
         4                             3                  3                         2
         5                             3                  3                         3
         7                             2                  3                         4

                         WorkLifeBalance  kmeans_2
```

```
EmployeeNumber
1                              1         4
2                              3         1
4                              3         4
5                              3         3
7                              3         0

[5 rows x 26 columns]
```

In [98]:
```python
X = rf_data.drop(['Attrition', 'kmeans_2'], axis=1)
y = rf_data['Attrition']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

Out[98]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

In [99]:
```python
feat_importance = sorted(list(zip(X_train.columns, model.feature_importances_)), key=
feat_importance
```

Out[99]:
```
[('MonthlyIncome', 0.09379283008900413),
 ('Age', 0.07683870994510993),
 ('MonthlyRate', 0.06662225205773609),
 ('DailyRate', 0.06650441578321957),
 ('HourlyRate', 0.06038339502709399),
 ('TotalWorkingYears', 0.058804030061166176),
 ('DistanceFromHome', 0.05694604928129479),
 ('StockOptionLevel', 0.04671605989505792),
 ('YearsAtCompany', 0.04550911339400427),
 ('PercentSalaryHike', 0.04478360732430316),
 ('NumCompaniesWorked', 0.043924822978152006),
 ('YearsInCurrentRole', 0.03772494749683493),
 ('YearsWithCurrManager', 0.03762775548739895),
 ('JobSatisfaction', 0.03250515707774317),
 ('TrainingTimesLastYear', 0.032204545361074324),
 ('EnvironmentSatisfaction', 0.03083096818778214),
 ('YearsSinceLastPromotion', 0.02912700440222398),
 ('JobInvolvement', 0.02834863436501242),
 ('RelationshipSatisfaction', 0.025315916283770886),
 ('WorkLifeBalance', 0.02462336240645489),
 ('JobLevel', 0.02176025573458919),
 ('Education', 0.020531780653247397),
```

```
('Gender', 0.01326538472757025),
('PerformanceRating', 0.0053090019801554636)]
```

The results obtained from the calculation (n the order of the importance) is as follows:
(1) 'MonthlyIncome', 0.09620026433543806,
(2) 'Age', 0.07465100132268015,
(3) 'MonthlyRate', 0.0678728429991805,
(4) 'DailyRate', 0.06395539641367105,
(5) 'HourlyRate', 0.060778294100276326,
(6) 'TotalWorkingYears', 0.059386776367443835,
(7) 'DistanceFromHome', 0.05537957826054828,
(8) 'YearsAtCompany', 0.04897909582304177,
(9) 'StockOptionLevel', 0.04782603253763362,
(10) 'PercentSalaryHike', 0.04393563135475805,
(11) 'NumCompaniesWorked', 0.04368879940471908,
(12) 'YearsWithCurrManager', 0.037408724867448895,
(13) 'YearsInCurrentRole', 0.036640647371725926,
(14) 'TrainingTimesLastYear', 0.034674160760093234,
(15) 'JobSatisfaction', 0.0322863585903226,
(16) 'EnvironmentSatisfaction', 0.03193999230375698,
(17) 'YearsSinceLastPromotion', 0.029594830954866907,
(18) 'RelationshipSatisfaction', 0.027932283322458836,
(19) 'JobInvolvement', 0.025080256516368957,
(20) 'Education', 0.025006182546777035,
(21) 'WorkLifeBalance', 0.02088513349853633,
(22) 'JobLevel', 0.020547005053445814,
(23) 'Gender', 0.011329291160433025,
(24) 'PerformanceRating', 0.004021420134374657

Finally, the accuracy score is calculated to see if the results are reliable.

```
In [100]: model.score(X_train, y_train), model.score(X_test, y_test)
```

```
Out[100]: (1.0, 0.8594104308390023)
```

Result: The accuracy score indicates that 0.85 (85%) is classified correctly, which is a high score knowing the score higher than 50% is considered a good score.
(End of Report)