**ATTRITION RATE ANALYSIS**

**Step 1**

Import the required libraries and packages

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**Step 2**

Load the csv file (data frame) to the workspace

dataset = pd.read\_csv("general\_data.csv")

**Step 3**

Data Cleaning

1. Display the first five columns of the data frame

dataset.head()

Out[5]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

[5 rows x 24 columns]

1. Display the column headings of the data frame

dataset.columns

Out[6]:

Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'], dtype='object')

1. Total number of records with null values

dataset.isnull().any(axis = 1).sum()

Out[7]: 28

1. Drop duplicates of the dataframe

dataset.drop\_duplicates()

Out[8]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

... ... ... ... ...

4405 42 No ... 0 2

4406 29 No ... 0 2

4407 25 No ... 1 2

4408 42 No ... 7 8

4409 40 No ... 3 9

[4410 rows x 24 columns]

1. Drop null values

dataCleaned= dataset.dropna()

1. After Cleaning Total number of records with null values

dataCleaned.isnull().any(axis = 1).sum()

Out[10]: 0

1. Change the Yes No value of Attrition to 0 and 1

from sklearn.preprocessing import LabelEncoder

lb = LabelEncoder()

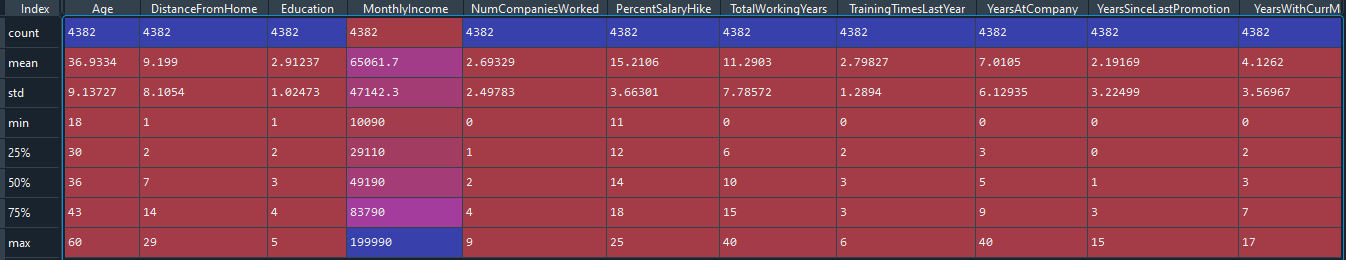
dataCleaned['Attrition'] = lb.fit\_transform(dataCleaned['Attrition'])

**Step 4**

Find the descriptive statistics values of the dataframe

1. Find the count, mean, standard deviation, min, max and quartile values

datDesc = dataCleaned[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()

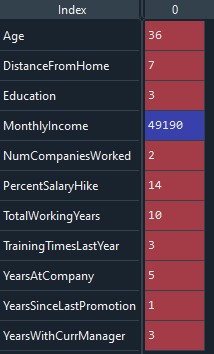


Inference

* All other values except Monthly income shows a minimum standard deviation. Huge SD in monthly income indicates that there are extreme values present
* Standard deviation of age indicates it is normally distributed

1. Find the median

descDatamed = dataCleaned[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].median()



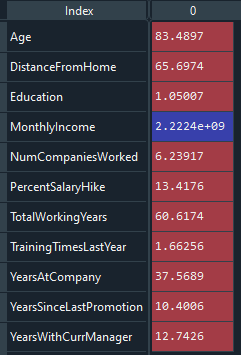
1. Find the mode

descDatamod = dataCleaned [[ 'Age', 'DistanceFromHome', 'Education', 'MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike', 'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager' ]] . mode()



1. Find the variance, skewness, kurtosis

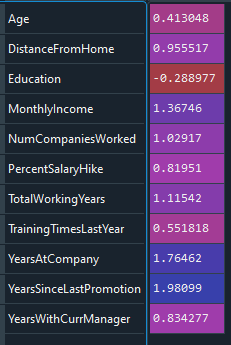
descDatavar = dataCleaned[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].var()



Inference

* High variance in Age, Distance from home and Total working years indicates that the data points are spread out from the mean

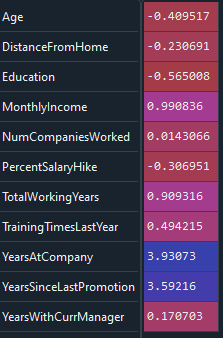
descDataskew = dataCleaned[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].skew()



Inference

* All the data points are positively skewed except Education which shows mild negative skewness

descDatakurt = dataCleaned[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].kurt()



Inference:

* Monthly income, Number of companies worked, Total Working Years, Training , Years at company shows leptokurtosis
* Age , Distance from home, Education and Salary hike shows platyokurtosis
* Years with Manger is normally distributed and shows mesokurtosis

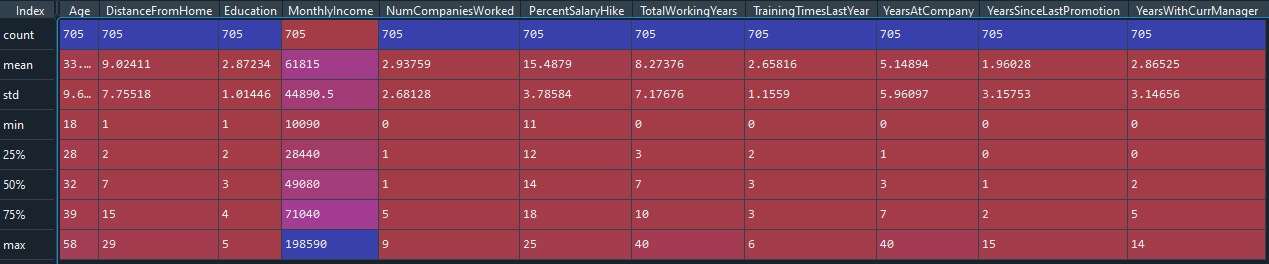
1. **Filter the Yes and No value records separately in two variables**

dataFilterAttrYs = dataCleaned[dataCleaned['Attrition']==1]

dataFilterAttrNo = dataCleaned[dataCleaned['Attrition']==0]

1. **Find the mean, standard deviation, min, max and count of employees with Attrition YES**

datAttrYesDes = dataFilterAttrYs[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()

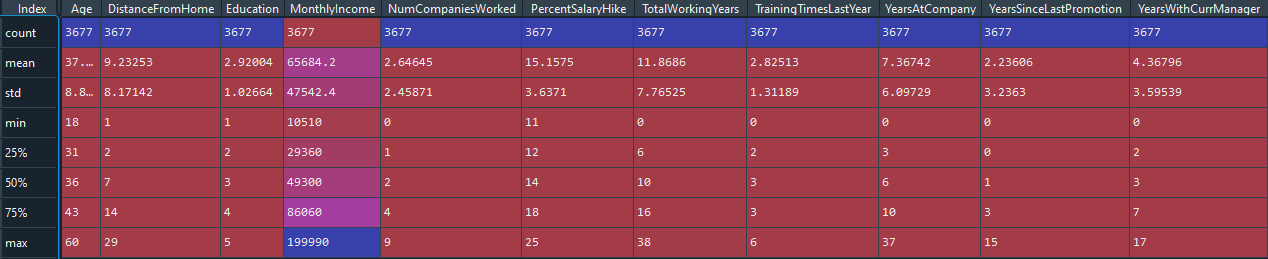


Inference from Attrition Yes Data

* Monthly income shows huge standard deviation

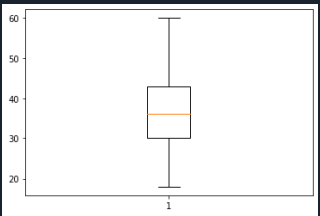
1. **Find the mean, standard deviation, min, max and count of employees with Attrition NO**

datAttrNoDes = dataFilterAttrNo[['Age','DistanceFromHome','Education','MonthlyIncome', 'NumCompaniesWorked', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany','YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()



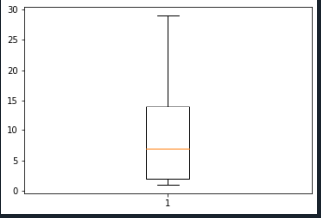
**Outliers**

plt.boxplot(dataCleaned['Age'])



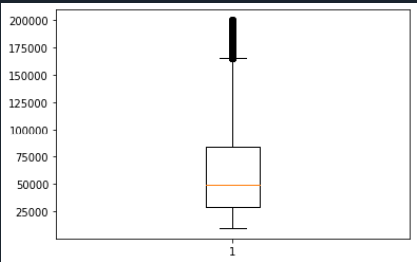
**There is no outliers in Age. The data is normally distributed**

plt.boxplot(dataCleaned['DistanceFromHome'])



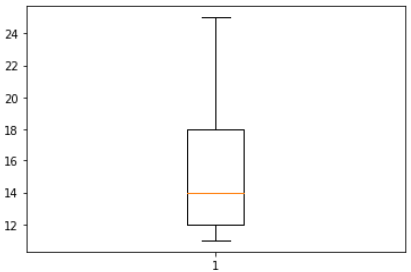
**No outliers and normally distributed**

plt.boxplot(dataCleaned['MonthlyIncome'])



**There are outliers in Monthly income and is skewed to right**

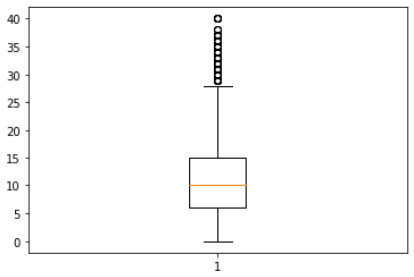
plt.boxplot(dataCleaned['PercentSalaryHike'])



**There are no outliers in 'PercentSalaryHike' and is skewed to right**

plt.boxplot(dataCleaned['TotalWorkingYears'])

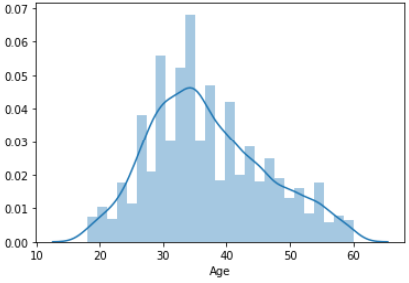
**There are outliers in Total Working Years and is normally distributed**



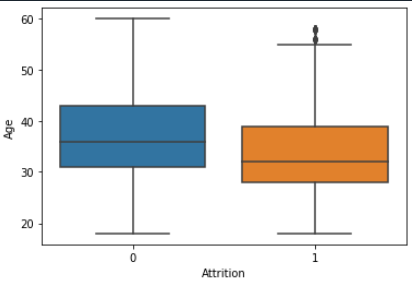
**Data Visualisation**

import seaborn as sns

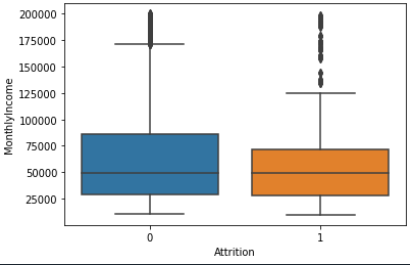
sns.distplot(dataCleaned["Age"])



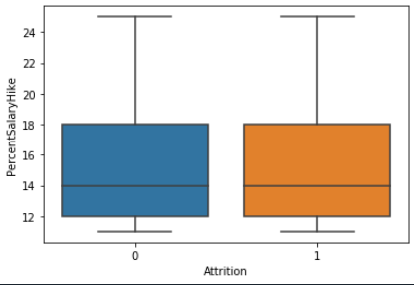
sns.boxplot(x="Attrition",y="Age",data=dataCleaned)



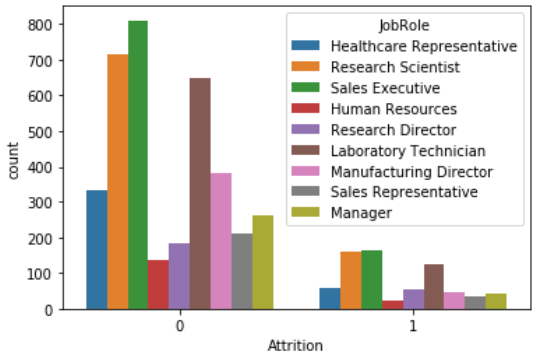
sns.boxplot(x="Attrition",y="MonthlyIncome",data=dataCleaned)



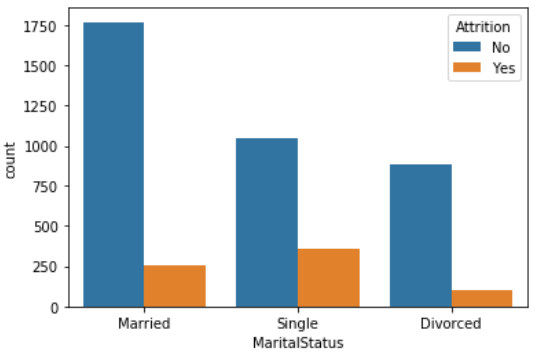
sns.boxplot(x="Attrition",y="PercentSalaryHike",data=dataCleaned)



sns.countplot(x="Attrition",hue="JobRole",data=dataCleaned)



sns.countplot(x="MaritalStatus",hue="Attrition",data=dataset)



**Statistical Test**

**Mann – Whitney Test**

**Attrition Vs Distance from Home**

H0: There is no significant differences in the Distance From Home between attrition (Y) and attirition (N)

Ha: There is significant differences in the Distance From Home between attrition (Y) and attirition (N)

from scipy.stats import mannwhitneyu

stats,p = mannwhitneyu(dataCleaned.Attrition,dataCleaned.DistanceFromHome)

print(stats,p)

219255.0 0.0

As the P value of 0.0 is < 0.05, the H0 is rejected and Ha is accepted.

**Attrition Vs EducationField**

H0: There is no significant differences in the EducationField between attrition (Y) and attirition (N)

Ha: There is significant differences in the EducationField between attrition (Y) and attirition (N)

stats,p = mannwhitneyu(dataset.Attrition,dataset.EducationField)

print(stats,p)

2374758.0 0.0

As the P value of 0.0 is < 0.05, the H0 is rejected and Ha is accepted.

**Attrition Vs MonthlyIncome**

H0: There is no significant differences in the MonthlyIncome between attrition (Y) and attirition (N)

Ha: There is significant differences in the MonthlyIncome between attrition (Y) and attirition (N)

stats,p = mannwhitneyu(dataCleaned.Attrition,dataCleaned.MonthlyIncome)

print(stats,p)

0.0 0.0

As the P value of 0.0 is < 0.05, the H0 is rejected and Ha is accepted.

**Attrition Vs YearsWithCurrManager**

H0: There is no significant differences in the YearsWithCurrManager between attrition (Y) and attirition (N)

Ha: There is significant differences in the YearsWithCurrManager between attrition (Y) and attirition (N)

stats,p = mannwhitneyu(dataCleaned.Attrition,dataCleaned.YearsWithCurrManager)

print(stats,p)

2069034.5 0.0

As the P value of 0.0 is < 0.05, the H0 is rejected and Ha is accepted.

**Statistical Tests (Separate T Test)**

**Attrition Vs Distance From Home**

from scipy.stats import ttest\_ind

stat, p=ttest\_ind(dataCleaned.Attrition,dataCleaned.DistanceFromHome)

print(stats,p)

2069034.5 0.0

As the P value is again 0.0, which is < than 0.05, the H0 is rejected and ha is accepted.

H0: There is no significant differences in the Distance From Home between attrition (Y) and attirition (N)

Ha: There is significant differences in the Distance From Home between attrition (Y) and attirition (N)

**Attrition Vs Yeats At Company**

stat, p=ttest\_ind(dataCleaned.Attrition,dataCleaned.YearsAtCompany)

print(stats,p)

2069034.5 0.0

As the P value is again 0.0, which is < than 0.05, the H0 is rejected and ha is accepted.

H0: There is no significant differences in the Years At Company between attrition (Y) and attirition (N)

Ha: There is significant differences in the Years At Company between attrition (Y) and attirition (N)

**Attrition Vs Total Working Hours**

stat, p=ttest\_ind(dataCleaned.Attrition,dataCleaned.TotalWorkingYears)

print(stats,p)

2069034.5 0.0

As the P value is again 0.0, which is < than 0.05, the H0 is rejected and ha is accepted.

H0: There is no significant differences in the Total Working Hours between attrition (Y) and attirition (N)

Ha: There is significant differences in the Total Working Hours between attrition (Y) and attirition (N)

**Attrition Vs Total Percentage of Salary Hike**

stat, p=ttest\_ind(dataCleaned.Attrition,dataCleaned.PercentSalaryHike)

print(stats,p)

2069034.5 0.0

As the P value is again 0.0, which is < than 0.05, the H0 is rejected and ha is accepted.

H0: There is no significant differences in the Percentage of Salary Hike between attrition (Y) and attirition (N)

Ha: There is significant differences in the Percentage of Salary Hike between attrition (Y) and attirition (N)

**Unsupervised Learning - Correlation Analysis**

In order to find the correlation of the variables DistanceFromHome, MonthlyIncome, TotalWorkingYears, YearsAtCompany, YearsWithCurrManager from that of Attrition, we executed the Correlation Analysis as follows.

**Attrition and DistanceFromHome**

from scipy.stats import pearsonr

stats, p=pearsonr(dataCleaned.Attrition, dataCleaned.DistanceFromHome)

print(stats,p)

-0.009448638515156248 0.5317715668019558

As r = -0.009, there’s low negative correlation between Attrition and DistanceFromHome

As the P value of 0.518 is > 0.05, we are accepting H0 and hence there’s no significant correlation between Attrition & DistanceFromHome

**Attrition and MonthlyIncome**

stats, p=pearsonr(dataCleaned.Attrition, dataCleaned.MonthlyIncome)

print(stats,p)

-0.030160293808460664 0.045890862744719166

As r = -0.031, there’s low negative correlation between Attrition and MonthlyIncome

As the P value of 0.045 is < 0.05, we are accepting Ha and hence there’s significant correlation between Attrition & MonthlyIncome

**Attrition and TotalWorkingYears**

stats, p=pearsonr(dataCleaned.Attrition, dataCleaned.TotalWorkingYears)

print(stats,p)

-0.1696699168472392 1.1645434967091854e-29

As r = -0.169, there’s low negative correlation between Attrition and TotalWorkingYears

As the P value of 1. 1645434967091854e-29 is < 0.05, we are accepting Ha and hence there’s significant correlation between Attrition & TotalWorkingYears

**Attrition and YearsAtCompany**

stats, p=pearsonr(dataCleaned.Attrition, dataCleaned.YearsAtCompany)

print(stats,p)

-0.1330026184252154 9.476118084836507e-19

As r = -0.133, there’s low negative correlation between Attrition and YearsAtCompany

As the P value of 9.476118084836507e-19 is < 0.05, we are accepting Ha and hence there’s significant correlation between Attrition & YearsAtCompany

**Attrition and Years With Current Manager**

stats, p=pearsonr(dataCleaned.Attrition, dataCleaned.YearsWithCurrManager)

print(stats,p)

-0.15469153690287274 7.105369646771178e-25

As r = -0.154, there’s low negative correlation between Attrition and YearsWithCurrManager

As the P value of 7.105369646771178e-25 is < 0.05, we are accepting Ha and hence there’s significant correlation between Attrition & YearsWithCurrManager