**Attrition Assignment Solution**

**Step1 - Launching**

import pandas as pd

import matplotlib.pyplot as plt

dbDataset = pd.read\_csv(‘general\_data.csv’)

dbDataset.head()

Out[1]:

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]

dbDataset.columns

Out[2]:

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')

**Step 2 - Data Treatment:**

dbDataset.isnull()

Out[3]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 False False ... False False

1 False False ... False False

2 False False ... False False

3 False False ... False False

4 False False ... False False

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

4405 False False ... False False

4406 False False ... False False

4407 False False ... False False

4408 False False ... False False

4409 False False ... False False

[4410 rows x 24 columns]

dbDataset.duplicated()

Out[4]:

0 False

1 False

2 False

3 False

4 False

4405 False

4406 False

4407 False

4408 False

4409 False

Length: 4410, dtype: bool

dbDataset.drop\_duplicates()

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

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

4405 42 No ... 0 2

4406 29 No ... 0 2

4407 25 No ... 1 2

4408 42 No ... 7 8

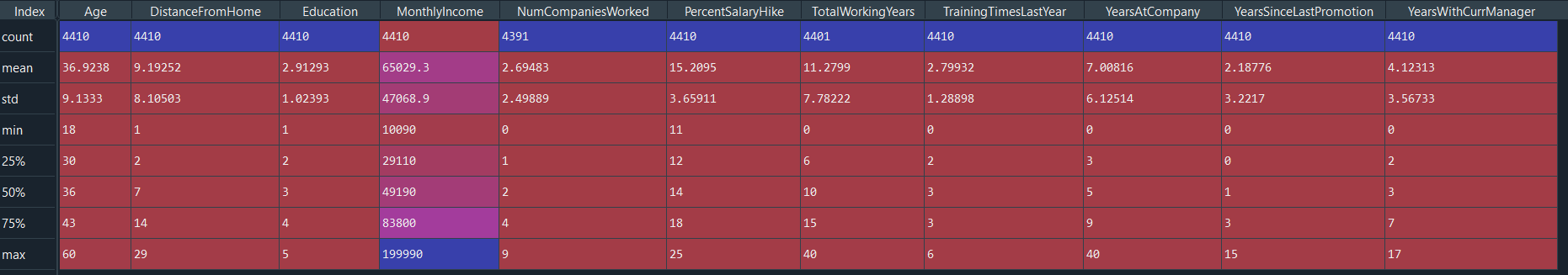
4409 40 No ... 3 9

[4410 rows x 24 columns]

**Step 3 – Univariate Analysis:**

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

dbSecondDataset



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

dbSecondDataset

Out[7]:

Age 36.0

DistanceFromHome 7.0

Education 3.0

MonthlyIncome 49190.0

NumCompaniesWorked 2.0

PercentSalaryHike 14.0

TotalWorkingYears 10.0

TrainingTimesLastYear 3.0

YearsAtCompany 5.0

YearsSinceLastPromotion 1.0

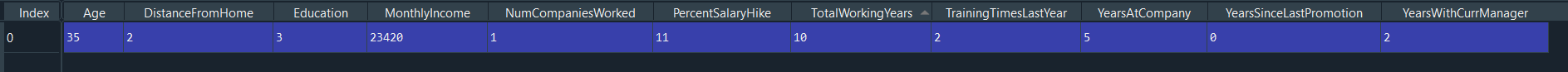
YearsWithCurrManager 3.0

dtype: float64

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

dbSecondDataset

Out[8]:



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

dbSecondDataset

Out[9]:

Age 8.341719e+01

DistanceFromHome 6.569144e+01

Education 1.048438e+00

MonthlyIncome 2.215480e+09

NumCompaniesWorked 6.244436e+00

PercentSalaryHike 1.338907e+01

TotalWorkingYears 6.056298e+01

TrainingTimesLastYear 1.661465e+00

YearsAtCompany 3.751728e+01

YearsSinceLastPromotion 1.037935e+01

YearsWithCurrManager 1.272582e+01

dtype: float64

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

dbSecondDataset

Out[9]:

Age 0.413005

DistanceFromHome 0.957466

Education -0.289484

MonthlyIncome 1.368884

NumCompaniesWorked 1.026767

PercentSalaryHike 0.820569

TotalWorkingYears 1.116832

TrainingTimesLastYear 0.552748

YearsAtCompany 1.763328

YearsSinceLastPromotion 1.982939

YearsWithCurrManager 0.832884

dtype: float64

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

dbSecondDataset

Out[10]:

Age -0.405951

DistanceFromHome -0.227045

Education -0.560569

MonthlyIncome 1.000232

NumCompaniesWorked 0.007287

PercentSalaryHike -0.302638

TotalWorkingYears 0.912936

TrainingTimesLastYear 0.491149

YearsAtCompany 3.923864

YearsSinceLastPromotion 3.601761

YearsWithCurrManager 0.167949

dtype: float64

**Inference from the analysis:**

• All the above variables show positive skewness except education; while Age, Distancefromhome, Education & Percentsalaryhike are platykurtic and all other variables are leptokurtic.

• The Mean MonthlyIncome’s IQR is at 54K suggesting company-wide attrition across all income bands

• Mean age forms a near normal distribution with 13 years of IQR

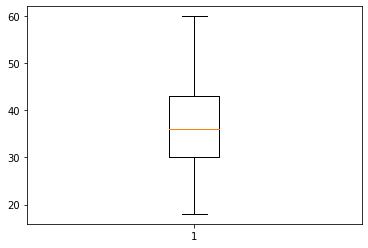
**Outliers:**

There’s no regression found while plotting Age, MonthlyIncome, TotalWorkingYears, YearsAtCompany, etc., on a scatter plot

box\_plot = dbDataset.Age

plt.boxplot(box\_plot)

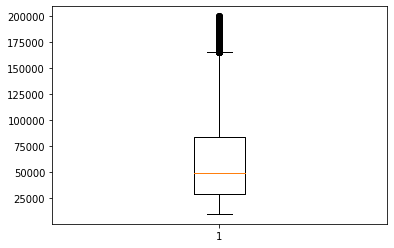
Out[11]:

Age is normally distributed without any outliers

box\_plot = dbDataset.MonthlyIncome

plt.boxplot(box\_plot)

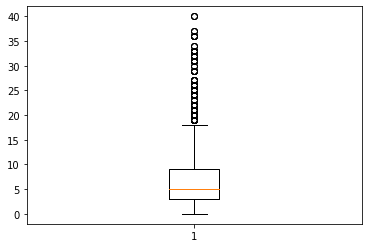
Out[12]:

Monthly Income is Right skewed with several outliers

box\_plot = dbDataset.YearsAtCompany

plt.boxplot(box\_plot)

Out[13]:

Years at company is also Right Skewed with several outliers observed.

**Step 4 – Statistical test (Mann-Whitney test):**

1. Mann-Whitney test on YearsAtCompany between Attrition and Non-Attrition employees.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition YearsAtCompany)

**Hypothesis:**

H0 -> There is no significant difference between YearsAtCompany for Attrition and NonAttrition

H1 -> These is significant difference between YearsAtCompany for Attrition and NonAttrition

*import pandas as pd*

*from scipy.stats import mannwhitneyu*

*# loading dataset*

*dataset = pd.* *read\_excel("general\_data.xlsx", sheet\_name=1)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = mannwhitneyu(dataset.* Attrition*, dataset.* NonAttrition*)*

*print(stats,p)*

*177883.5 1.274723628781325e-22*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *1.274723628781325e-22* | *Less than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

1. Mann-Whitney test on YearsWithCurrManager between Attrition and Non-Attrition employees.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition YearsWithCurrManager)

**Hypothesis:**

H0 -> There is no significant difference between YearsWithCurrManager for Attrition and NonAttrition

H1 -> These is significant difference between YearsWithCurrManager for Attrition and NonAttrition

*import pandas as pd*

*from scipy.stats import mannwhitneyu*

*# loading dataset*

*dataset = pd.* *read\_excel("general\_data.xlsx", sheet\_name=5)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = mannwhitneyu(dataset.* Attrition*, dataset.* NonAttrition*)*

*print(stats,p)*

*187252.5 4.069546197615495e-18*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *4.069546197615495e-18* | *Less than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

1. Mann-Whitney test on MonthlyIncome between Attrition and Non-Attrition employees.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition MonthlyIncome)

**Hypothesis:**

H0 -> There is no significant difference between MonthlyIncome for Attrition and NonAttrition

H1 -> These is significant difference between MonthlyIncome for Attrition and NonAttrition

*import pandas as pd*

*from scipy.stats import mannwhitneyu*

*# loading dataset*

*dataset = pd.* *read\_excel("general\_data.xlsx", sheet\_name=1)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = mannwhitneyu(dataset.*Attrition*, dataset.*NonAttrition*)*

*print(stats,p)*

*244623.0 0.14663998329920325*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *0.14663998329920325* | *Greater than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

1. Mann-Whitney test on DistanceFromHome between Attrition and Non-Attrition employees.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition DistanceFromHome)

**Hypothesis:**

H0 -> There is no significant difference between DistanceFromHome for Attrition and NonAttrition

H1 -> These is significant difference between DistanceFromHome for Attrition and NonAttrition

*import pandas as pd*

*from scipy.stats import mannwhitneyu*

*# loading dataset*

*dataset = pd.* *read\_excel("general\_data.xlsx", sheet\_name=1)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = mannwhitneyu(dataset.*Attrition*, dataset.*NonAttrition*)*

*print(stats,p)*

*250899.0 0.4046713347996649*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *0.4046713347996649* | *Greater than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

**Step 5 – Statistical test (Separate T-Test):**

1. Two-Sample independent t test on mean of PercentSalaryHike for Attrition and NonAttrition.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition PercentSalaryHike)

**Hypothesis:**

H0 -> There is no significant difference in the mean of PercentSalaryHike between Attrition and NonAttrition employees

H1 -> There is significant difference in the mean of PercentSalaryHike between Attrition and NonAttrition employees

*import pandas as pd*

*from scipy.stats import ttest\_ind*

*# loading dataset*

*dataset = pd.read\_excel("general\_data.xlsx",sheet\_name=2)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = ttest\_ind(dataset.Attrition, dataset.NonAttrition)*

*print(stats,p)*

*1.2154526422906522 0.22439553878006924*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *0.22439553878006924* | *Greater than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

1. Two-Sample independent t test on mean of MonthlyIncome for Attrition and NonAttrition.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition MonthlyIncome)

**Hypothesis:**

H0 -> There is no significant difference in the mean of MonthlyIncome between Attrition and NonAttrition employees

H1 -> There is significant difference in the mean of MonthlyIncome between Attrition and NonAttrition employees

*import pandas as pd*

*from scipy.stats import ttest\_ind*

*# loading dataset*

*dataset = pd.read\_excel("general\_data.xlsx",sheet\_name=1)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = ttest\_ind(dataset.Attrition, dataset.NonAttrition)*

*print(stats,p)*

*-7.058064846165972 2.632171469019036e-12*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *2.632171469019036e-12* | *Lesser than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

1. Two-Sample independent t test on mean of DistanceFromHome for Attrition and NonAttrition.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition DistanceFromHome)

**Hypothesis:**

H0 -> There is no significant difference in the mean of DistanceFromHome between Attrition and NonAttrition employees

H1 -> There is significant difference in the mean of DistanceFromHome between Attrition and NonAttrition employees

*import pandas as pd*

*from scipy.stats import ttest\_ind*

*# loading dataset*

*dataset = pd.read\_excel("general\_data.xlsx",sheet\_name=3)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = ttest\_ind(dataset.Attrition, dataset.NonAttrition)*

*print(stats,p)*

*-0.12432886017155614 0.9010724941219667*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *0.9010724941219667* | *Greater than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

1. Two-Sample independent t test on mean of YearsAtCompany for Attrition and NonAttrition.

(For this analysis, the given .csv file was converted into .xlsx file and filtering was done to segregate between Attrition and Non-Attrition YearsAtCompany)

**Hypothesis:**

H0 -> There is no significant difference in the mean of YearsAtCompany between Attrition and NonAttrition employees

H1 -> There is significant difference in the mean of YearsAtCompany between Attrition and NonAttrition employees

*import pandas as pd*

*from scipy.stats import ttest\_ind*

*# loading dataset*

*dataset = pd.read\_excel("general\_data.xlsx",sheet\_name=4)*

*# data treatment*

*dataset.isnull()*

*dataset.dropna(inplace=True)*

*# statistical test*

*stats,p = ttest\_ind(dataset.Attrition, dataset.NonAttrition)*

*print(stats,p)*

*-7.058064846165972 2.632171469019036e-12*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *p value* | *2.632171469019036e-12* | *Lesser than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

**Step 6 – Unsupervised Learning - Correlation Analysis:**

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

1. Correlation between **Attrition** and **DistanceFromHome**

**Hypothesis:**

H0 -> There is no significant correlation between Attrition and DistanceFromHome

H1 -> These is significant correlation between Attrition and DistanceFromHome

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from sklearn.preprocessing import LabelEncoder*

*from scipy.stats import pearsonr*

*# loading dataset*

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

*# data treatment*

*dataset.isnull()*

*dataset.dropna()*

*# converting categorical string data into integers*

*number = LabelEncoder()*

*dataset["Attrition"] = number.fit\_transform(dataset["Attrition"].astype("str"))*

*# determining correlation*

*stats,p = pearsonr(dataset.Attrition, dataset.DistanceFromHome)*

*print(stats,p)*

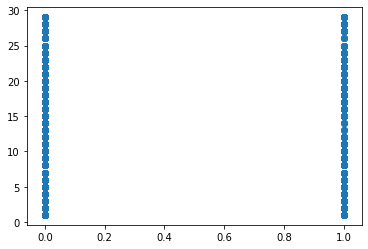
*-0.009730141010179438 0.5182860428049617*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *Correlation coefficient, r* | *-0.009730141010179438* | *Negative correlation* |
| *p value* | *0.5182860428049617* | *Greater than 0.05, thus NULL HYPOTHESIS IS ACCEPTED* |

*# graphically displaying correlation*

*plt.scatter(dataset.Attrition, dataset.DistanceFromHome)*



1. Correlation between **Attrition** and **MonthlyIncome**

**Hypothesis:**

H0 -> There is no significant correlation between Attrition and MonthlyIncome

H1 -> These is significant correlation between Attrition and MonthlyIncome

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from sklearn.preprocessing import LabelEncoder*

*from scipy.stats import pearsonr*

*# loading dataset*

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

*# data treatment*

*dataset.isnull()*

*dataset.dropna()*

*# converting categorical string data into integers*

*number = LabelEncoder()*

*dataset["Attrition"] = number.fit\_transform(dataset["Attrition"].astype("str"))*

*# determining correlation*

*stats,p = pearsonr(dataset.Attrition, dataset.MonthlyIncome)*

*print(stats,p)*

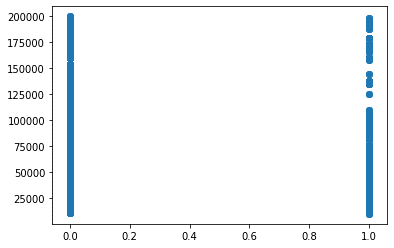
*-0.03016029380845958 0.04589086274474114*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *Correlation coefficient, r* | *-0.03016029380845958* | *Negative correlation* |
| *p value* | *0.04589086274474114* | *Less than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

*#graphically displaying correlation*

*plt.scatter(dataset.Attrition, dataset.MonthlyIncome)*



1. Correlation between **Attrition** and **YearsAtCompany**

**Hypothesis:**

H0 -> There is no significant correlation between Attrition and DistanceFromHome

H1 -> These is significant correlation between Attrition and DistanceFromHome

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from sklearn.preprocessing import LabelEncoder*

*from scipy.stats import pearsonr*

*# loading dataset*

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

*# data treatment*

*dataset.isnull()*

*dataset.dropna()*

*# converting categorical string data into integers*

*number = LabelEncoder()*

*dataset["Attrition"] = number.fit\_transform(dataset["Attrition"].astype("str"))*

*# determining correlation*

*stats,p = pearsonr(dataset.Attrition, dataset.YearsAtCompany)*

*print(stats,p)*

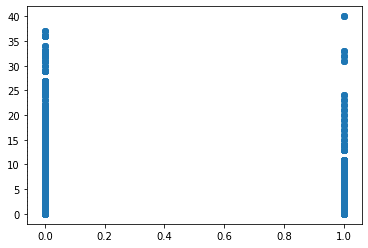
*-0.13300261842521083 9.476118084864852e-19*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *Correlation coefficient, r* | *-0.13300261842521083* | *Negative correlation* |
| *p value* | *9.476118084864852e-19* | *Less than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

*#graphically displaying correlation*

*plt.scatter(dataset.Attrition, dataset.YearsAtCompany)*



1. Correlation between **Attrition** and **YearsWithCurrManager**

**Hypothesis:**

H0 -> There is no significant correlation between Attrition and YearsWithCurrManager

H1 -> These is significant correlation between Attrition and YearsWithCurrManager

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from sklearn.preprocessing import LabelEncoder*

*from scipy.stats import pearsonr*

*# loading dataset*

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

*# data treatment*

*dataset.isnull()*

*dataset.dropna()*

*# converting categorical string data into integers*

*number = LabelEncoder()*

*dataset["Attrition"] = number.fit\_transform(dataset["Attrition"].astype("str"))*

*# determining correlation*

*stats,p = pearsonr(dataset.Attrition, dataset.YearsWithCurrManager)*

*print(stats,p)*

*-0.154691536902868 7.1053696467956645e-25*

*Analysis:*

|  |  |  |
| --- | --- | --- |
| *Parameter* | *Value* | *Inference* |
| *Correlation coefficient, r* | *-0.154691536902868* | *Negative correlation* |
| *p value* | *7.1053696467956645e-25* | *Less than 0.05, thus NULL HYPOTHESIS IS REJECTED* |

*#graphically displaying correlation*

*plt.scatter(dataset.Attrition, dataset.YearsWithCurrManager)*

