Project3___EDA-VIS

February 19, 2024

1 HR Data

1.0.1 import requirements

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

1.0.2 import data from csv file

[10]:
```
df = pd.read_csv('HR-Employee-Attrition.csv')
df.head()
```

[10]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	…	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	…	1	80	0	8	0	1	6
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	…	4	80	1	10	3	3	10
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	…	2	80	0	7	3	3	0
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	…	3	80	0	8	3	3	8
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	…	4	80	1	6	3	3	2

```
2                    7                         3                3                 0
3                    8                         3                3                 8
4                    6                         3                3                 2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

1.0.3 informations about the dataframe

[11]:
```
df.shape
```

[11]:
```
(1470, 35)
```

[3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
```

```
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

1.0.4 name of columns

[12]:
```
df.columns
```

[12]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', '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')
```

1.0.5 Number of missing values

[14]:
```
df.isnull().sum()
```

[14]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
```

```
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
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
```

1.0.6 deleting missing and duplicates data

[15]:
```
df = df.dropna()
```

[18]:
```
df = df.drop_duplicates()
```

[19]:
```
df.head()
```

[19]:

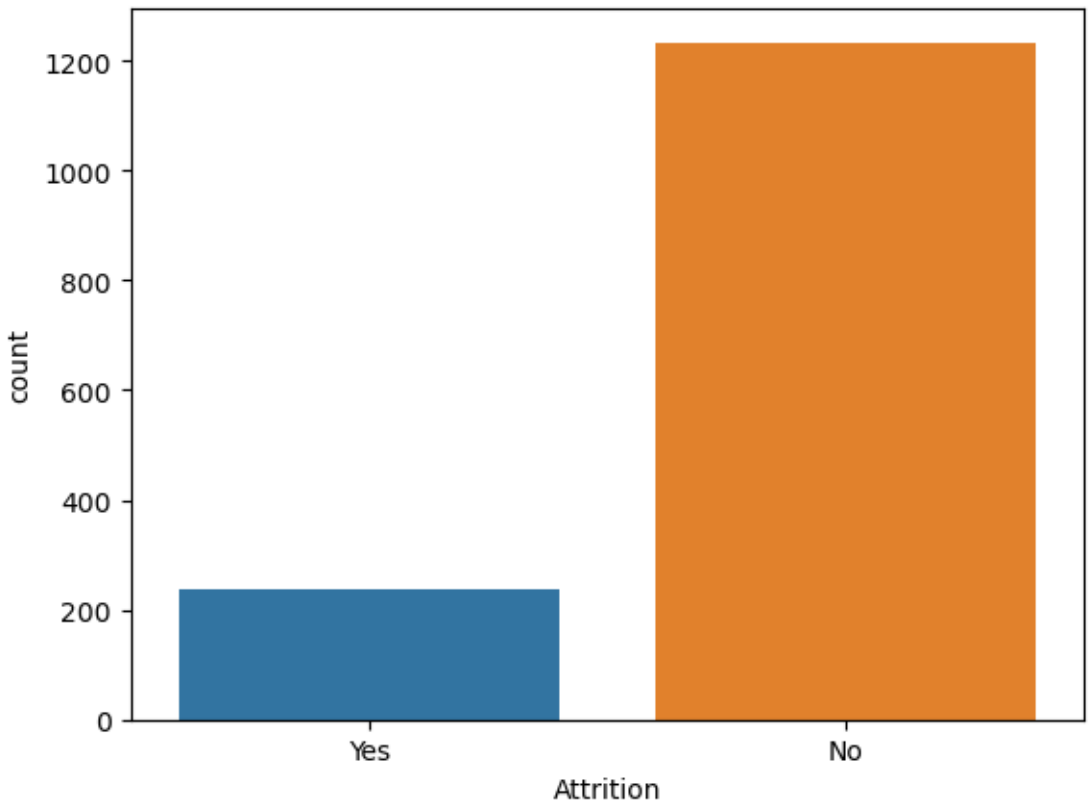
	Age	Attrition	BusinessTravel	DailyRate	Department	\
0	41	Yes	Travel_Rarely	1102	Sales	
1	49	No	Travel_Frequently	279	Research & Development	
2	37	Yes	Travel_Rarely	1373	Research & Development	
3	33	No	Travel_Frequently	1392	Research & Development	
4	27	No	Travel_Rarely	591	Research & Development	

	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	\
0	1	2	Life Sciences	1	1	
1	8	1	Life Sciences	1	2	
2	2	2	Other	1	4	
3	3	4	Life Sciences	1	5	
4	2	1	Medical	1	7	

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

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

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

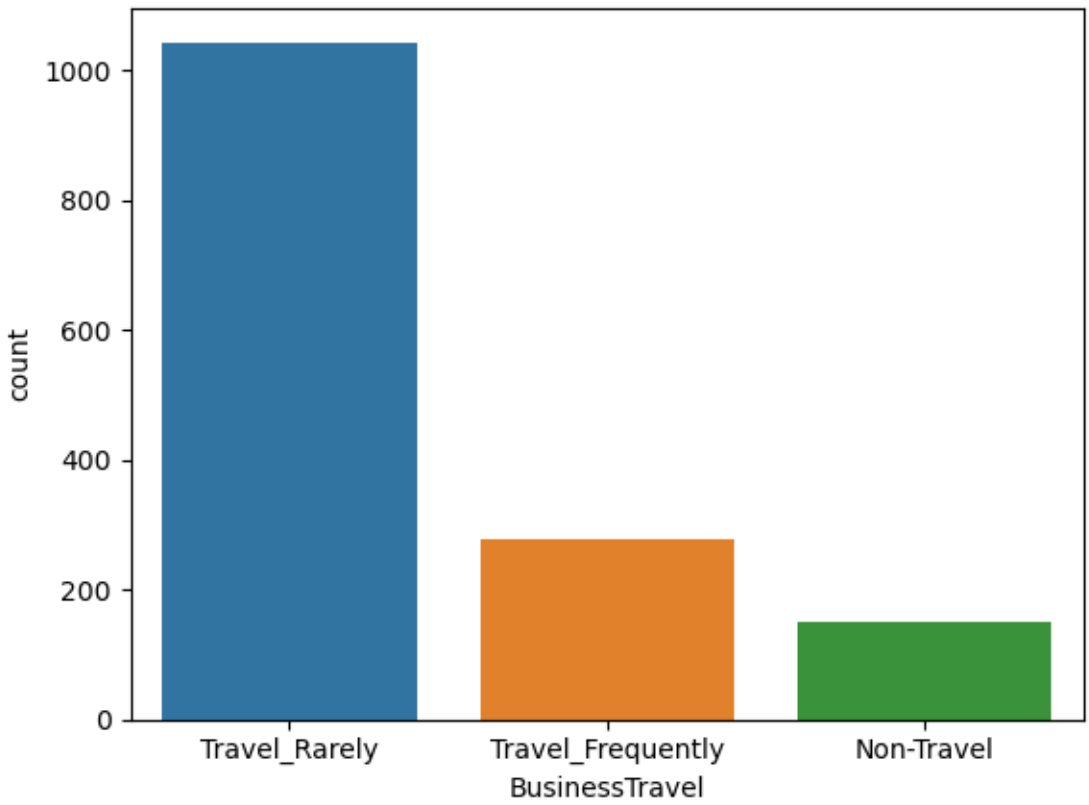
1.0.7 ploting Attrition numbers

[28]:
```
sns.countplot(x='Attrition', data=df) #yes pour les employées qui on quité la␣
↪societe et No pour les restants
plt.show()
```



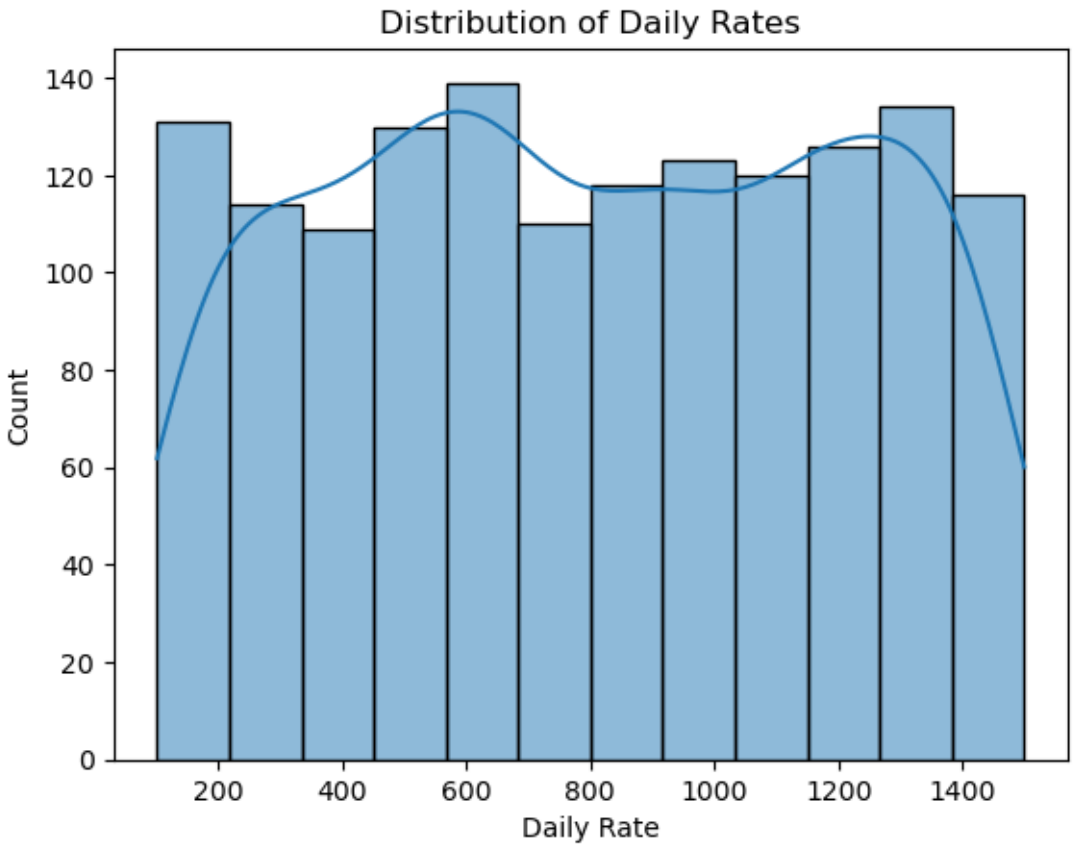
1.0.8 ploting number of business travel

[32]:
```
sns.countplot(x='BusinessTravel', data=df)  #refers to the daily salary that an␣
↪employee receives for their work.
plt.show()
```



1.0.9 ploting distribution of daily rates

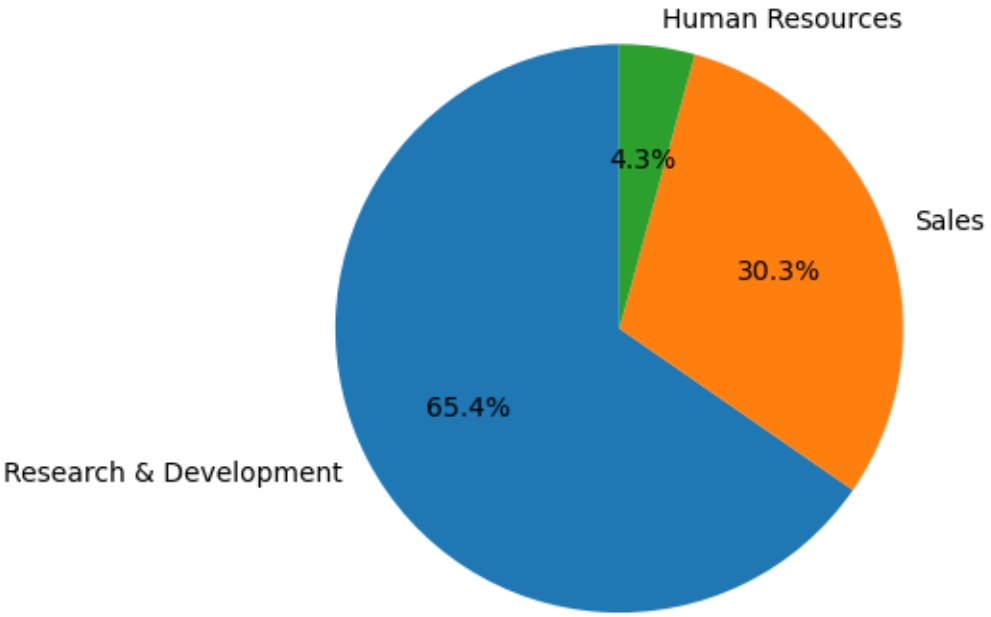
[34]:
```
sns.histplot(df['DailyRate'], kde=True)
plt.xlabel('Daily Rate')
plt.ylabel('Count')
plt.title('Distribution of Daily Rates')
plt.show()
```



1.0.10 ploting number of employees by departement

[40]:
```
plt.pie(df['Department'].value_counts() , labels=df['Department'].
↪value_counts().index , autopct='%1.1f%%', startangle=90)
```

[40]:
```
([<matplotlib.patches.Wedge at 0x23eacd603d0>,
  <matplotlib.patches.Wedge at 0x23eacd61750>,
  <matplotlib.patches.Wedge at 0x23eacd62d90>],
 [Text(-0.9741696801325502, -0.5108751650946098, 'Research & Development'),
  Text(1.0339296864268501, 0.3754855570128838, 'Sales'),
  Text(0.14765677500918445, 1.0900447132085396, 'Human Resources')],
 [Text(-0.5313652800723001, -0.27865918096069625, '65.4%'),
  Text(0.5639616471419181, 0.20481030382520932, '30.3%'),
  Text(0.08054005909591878, 0.5945698435682942, '4.3%')])
```



[41]:
```
df.head()
```

[41]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	…	RelationshipSatisfaction	StandardHours	StockOptionLevel
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	…	1	80	0
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	…	4	80	1
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	…	2	80	0
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	…	3	80	0
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	…	4	80	1

```
   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

1.0.11 Statistics about employees

[42]:
```
df['Age'].mean()
```

[42]:
```
36.923809523809524
```

[47]:
```
df.loc[df['Attrition'] == 'Yes' , 'Attrition'].count()
```

[47]:
```
237
```

[48]:
```
df.loc[df['Attrition'] == 'No' , 'Attrition'].count()
```

[48]:
```
1233
```

[50]:
```
df['Education'].median()
```

[50]:
```
3.0
```

[54]:
```
df.isna().sum()
```

[54]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
```

```
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
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
```

[55]:
```
df.columns
```

[55]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', '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')
```

1.0.12 spliting data into old_employees and actual employees

[64]:
```
old_emp = df.loc[df['Attrition'] == 'Yes' , :]
old_emp.shape
```

[64]:
```
(237, 35)
```

[66]:
```
emp = df.loc[df['Attrition'] == 'No' , :]
emp.shape
```

[66]:
```
(1233, 35)
```

1.0.13 Statistics about old and actual employees

[68]:
```
old_emp['JobSatisfaction'].mean()
```

[68]:
```
2.4683544303797467
```

[67]:
```
emp['JobSatisfaction'].mean()
```

[67]:
```
2.778588807785888
```

[69]:
```
old_emp['MonthlyIncome'].mean()
```

[69]:
```
4787.0928270042195
```

[70]:
```
emp['MonthlyIncome'].mean()
```

[70]:
```
6832.739659367397
```

[74]:
```
old_emp['TotalWorkingYears'].mean()
```

[74]:
```
8.244725738396625
```

[73]:
```
emp['TotalWorkingYears'].mean()
```

[73]:
```
11.862935928629359
```

1.0.14 encode categorial column `BusinessTravel`

[84]:
```
dict = {'Non-Travel':0  , 'Travel_Rarely' : 1 , 'Travel_Frequently':2}
old_emp['BusinessTravel'].replace(dict)
emp['BusinessTravel'].replace(dict)
```

[84]:
```
1       2
3       2
4       1
5       2
6       1
       ..
1465    2
1466    1
1467    1
1468    2
1469    1
```

```
Name: BusinessTravel, Length: 1233, dtype: int64
```

1.0.15 statistic using BusinessTravel column

[105]:
```
old_emp.loc[old_emp['BusinessTravel'] == 0 , 'BusinessTravel'].count() ,␣
↪old_emp.loc[old_emp['BusinessTravel'] == 1 , 'BusinessTravel'].count()  ,␣
↪old_emp.loc[old_emp['BusinessTravel'] == 2 , 'BusinessTravel'].count()
```

[105]:
```
(81, 156, 0)
```

[106]:
```
emp.loc[emp['BusinessTravel'] == 0 , 'BusinessTravel'].count() , emp.
↪loc[emp['BusinessTravel'] == 1 , 'BusinessTravel'].count()  , emp.
↪loc[emp['BusinessTravel'] == 2 , 'BusinessTravel'].count()
```

[106]:
```
(0, 0, 0)
```

[119]:
```
emp.groupby('Gender')['Gender'].count()
```

[119]:
```
Gender
Female    501
Male      732
Name: Gender, dtype: int64
```

[120]:
```
old_emp.groupby('Gender')['Gender'].count()
```

[120]:
```
Gender
Female     87
Male      150
Name: Gender, dtype: int64
```

[124]:
```
df['DistanceFromHome'].max() , df['DistanceFromHome'].min()
```

[124]:
```
(29, 1)
```

[127]:
```
emp['DistanceFromHome'].mean() , emp['DistanceFromHome'].median()
```

[127]:
```
(8.915652879156529, 7.0)
```

[128]:
```
old_emp['DistanceFromHome'].mean() , old_emp['DistanceFromHome'].median()
```

[128]:
```
(10.632911392405063, 9.0)
```

[131]:
```
emp.groupby('Gender')['DistanceFromHome'].mean() , emp.
↪groupby('Gender')['DistanceFromHome'].median()
```

[131]:
```
(Gender
 Female    8.914172
 Male      8.916667
```

```
Name: DistanceFromHome, dtype: float64,
Gender
Female    7.0
Male      7.0
Name: DistanceFromHome, dtype: float64)
```

[132]:
```
old_emp.groupby('Gender')['DistanceFromHome'].mean() , old_emp.
↪groupby('Gender')['DistanceFromHome'].median()
```

[132]:
```
(Gender
Female    10.919540
Male      10.466667
Name: DistanceFromHome, dtype: float64,
Gender
Female    9.0
Male      8.0
Name: DistanceFromHome, dtype: float64)
```

1.0.16 encode categorial column Attrition and drop other column not used

[134]:
```
df.head(1)
```

[134]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	\
0	41	Yes	Travel_Rarely	1102	Sales	1	

	Education	EducationField	EmployeeCount	EmployeeNumber	…	\
0	2	Life Sciences	1	1	…	

	RelationshipSatisfaction	StandardHours	StockOptionLevel	\
0	1	80	0	

	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	\
0	8	0	1	6	

	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
0	4	0	5

[1 rows x 35 columns]

[160]:
```
df_num = df.loc[:,:]
dict_num1={'Yes':1 , 'No':0}
df_num['Attrition']=df_num['Attrition'].replace(dict_num1)
df_num = df_num.drop(['BusinessTravel', 'Department' ,␣
↪'EducationField','Gender','JobRole','MaritalStatus','Over18','OverTime'] ,␣
↪axis=1)
```

[161]:
```
df_num.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 27 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   int64
 2   DailyRate                 1470 non-null   int64
 3   DistanceFromHome          1470 non-null   int64
 4   Education                 1470 non-null   int64
 5   EmployeeCount             1470 non-null   int64
 6   EmployeeNumber            1470 non-null   int64
 7   EnvironmentSatisfaction   1470 non-null   int64
 8   HourlyRate                1470 non-null   int64
 9   JobInvolvement            1470 non-null   int64
 10  JobLevel                  1470 non-null   int64
 11  JobSatisfaction           1470 non-null   int64
 12  MonthlyIncome             1470 non-null   int64
 13  MonthlyRate               1470 non-null   int64
 14  NumCompaniesWorked        1470 non-null   int64
 15  PercentSalaryHike         1470 non-null   int64
 16  PerformanceRating         1470 non-null   int64
 17  RelationshipSatisfaction  1470 non-null   int64
 18  StandardHours             1470 non-null   int64
 19  StockOptionLevel          1470 non-null   int64
 20  TotalWorkingYears         1470 non-null   int64
 21  TrainingTimesLastYear     1470 non-null   int64
 22  WorkLifeBalance           1470 non-null   int64
 23  YearsAtCompany            1470 non-null   int64
 24  YearsInCurrentRole        1470 non-null   int64
 25  YearsSinceLastPromotion   1470 non-null   int64
 26  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(27)
memory usage: 310.2 KB
```

[162]:
```
df_num.head()
```

[162]:

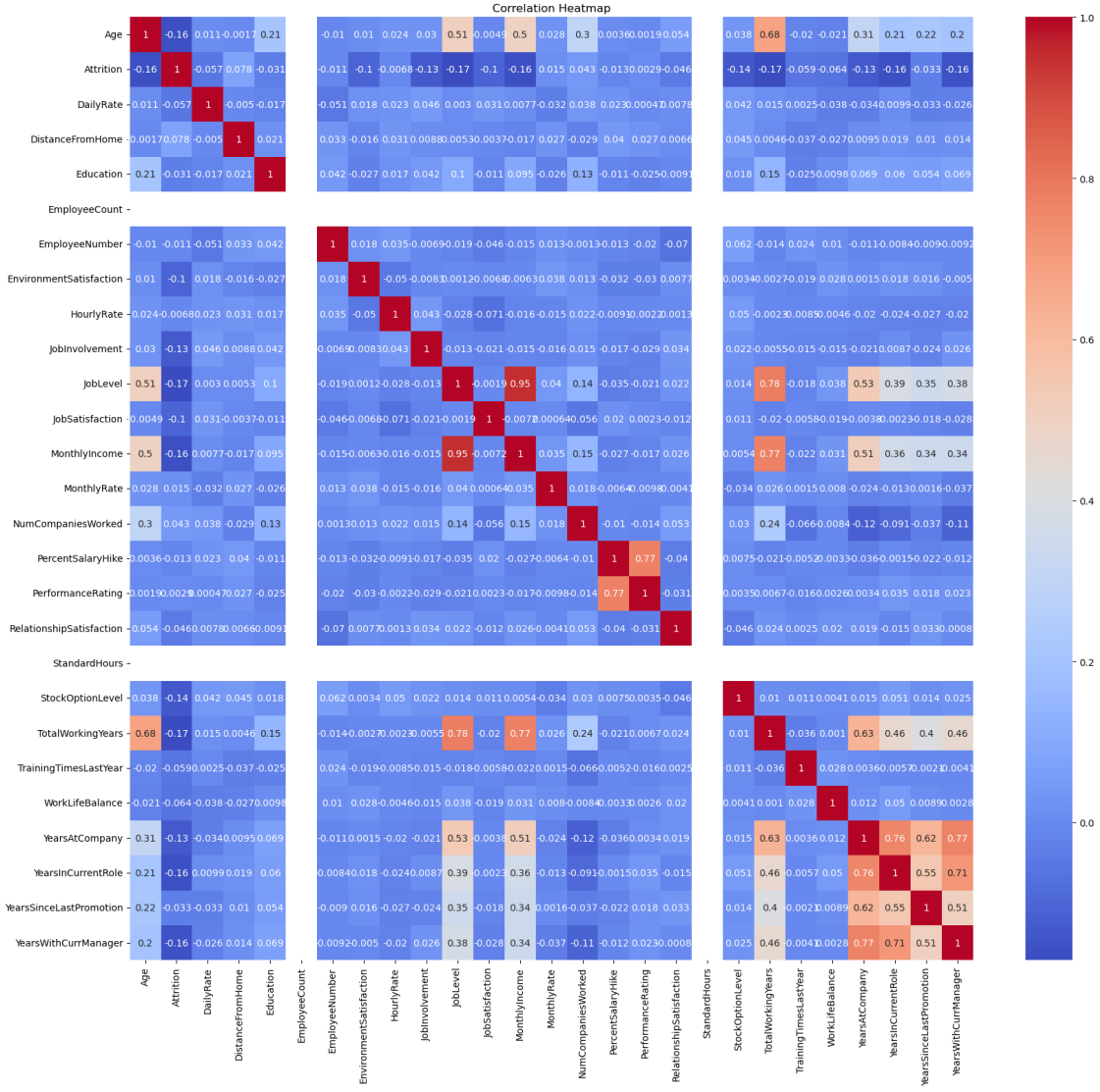
	Age	Attrition	DailyRate	DistanceFromHome	Education	EmployeeCount	\
0	41	1	1102	1	2	1	
1	49	0	279	8	1	1	
2	37	1	1373	2	2	1	
3	33	0	1392	3	4	1	
4	27	0	591	2	1	1	

	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	…	\
0	1	2	94	3	…	
1	2	3	61	2	…	

```
2                4                         4          92                2  …
3                5                         4          56                3  …
4                7                         1          40                3  …

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

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

   YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                   4                        0                     5
1                   7                        1                     7
2                   0                        0                     0
3                   7                        3                     0
4                   2                        2                     2

[5 rows x 27 columns]
```

1.0.17 correlation between numeric values

[169]:
```
# Calculate correlation matrix
correlation_matrix = df_num.corr()

# Plot correlation heatmap
plt.figure(figsize=(20, 18))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```


Correlation Heatmap
Age
Attrition
DailyRate
DistanceFromHome
Education
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
1.0
0.8
0.6
0.4
0.2
0.0