In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import re,os,glob,pickle
import datetime
import matplotlib
import warnings
warnings.filterwarnings('ignore')
import yaml,logging
from datetime import timedelta

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [2]:

```
start = datetime.datetime.now()
path = 'C:/Users/ebhavar/OneDrive - Ericsson AB/Desktop/DeleteMe/Python/CaseStudy
mgr_leaves = pd.read_excel(path+'Leave Records Report.xlsx', sheet_name = 'Sheet
mgr_active_emp = pd.read_excel(path+'Managers- Active Employees.xlsx', sheet_name
mgr_certs = pd.read_excel(path+'Managers- Active Employees.xlsx', sheet_name = 'C
mgr_perfTier = pd.read_excel(path+'Managers- Active Employees.xlsx', sheet_name =
mgr_edu = pd.read_excel(path+'Managers- Active Employees.xlsx', sheet_name = 'Edu
mgr_workex = pd.read_excel(path+'Managers- Active Employees.xlsx', sheet_name = '
end_time = datetime.datetime.now()
print("Time Consumed in reading Files--> ",(end_time-start))
```

```
Time Consumed in reading Files-->  0:00:02.983558
```

In [3]:

```
## Counting the total certification for Employee
mgr_certs = mgr_certs.groupby('LPN').count()[['Employee Status','Designation']]
mgr_certs = mgr_certs.rename(columns={'Employee Status':'Count_Certifications'}).
mgr_certs.sample(5)
```

Out[3]:

	LPN	Count_Certifications	Designation
23	5323131	2	2
9	3322079	1	1
21	5316058	1	1
1	2322764	1	1
5	2328879	1	1

In [4]:
```python
df = pd.merge(left=mgr_active_emp,right=mgr_certs[['LPN','Count_Certifications']]
print("Shape of mgr_active_emp :",mgr_active_emp.shape)
print("Shape after Merge certs :",df.shape)
df.sample(5)
```

```
Shape of mgr_active_emp : (156, 14)
Shape after Merge certs : (156, 15)
```

Out[4]:

	LPN	Employee Status	Designation	Band	Service Line	Sub Service Line	Work Location	Work Location City	Country/Region
59	4323881	Active	Manager	3	Service Line 1	Sub Service Line 1	India	NaN	India
68	1326863	Active	Manager	1	Service Line 1	Sub Service Line 1	India	NaN	India
54	5329170	Active	Manager	2	Service Line 5	Sub Service Line 10	India	NaN	India
14	4326931	Active	Manager	1	Service Line 4	Sub Service Line 8	India	NaN	India
109	5322691	Active	Manager	1	Service Line 3	Sub Service Line 6	India	NaN	India

In [5]:

```
# Merging the Performance Data:
df2 = pd.merge(left=df,right=mgr_perfTier[['LPN','Performance Rating 2020','Perfo
print("Shape of mgr_active_emp :",df.shape)
print("Shape after PerformaceRating :",df2.shape)
df2.sample(5)
```

```
Shape of mgr_active_emp : (156, 15)
Shape after PerformaceRating : (156, 17)
```

Out[5]:

	LPN	Employee Status	Designation	Band	Service Line	Sub Service Line	Work Location	Work Location City	Country/Region
90	5325319	Active	Manager	2	Service Line 3	Sub Service Line 6	India	NaN	India
20	3318681	Active	Manager	2	Service Line 1	Sub Service Line 1	India	NaN	India
18	4327636	Active	Manager	1	Service Line 3	Sub Service Line 7	India	NaN	India
13	3324307	Active	Manager	3	Service Line 1	Sub Service Line 2	India	NaN	India
131	4320825	Active	Manager	1	Service Line 2	Sub Service Line 4	India	NaN	India

In [6]:

```
### WOrking on the Education Section. Few entries are duplicated like where peopl
# Concat Degree if duplicate
mgr_edu_temp = mgr_edu.groupby(['LPN'])['Degree'].transform(lambda x: ','.join(x)
# Creating DF from Series
mgr_edu_temp = pd.DataFrame({'Concat_Degree':mgr_edu_temp.values})
mgr_edu=pd.concat([mgr_edu,mgr_edu_temp],axis=1)
def find_highest_education(Concat_Degree):
    if 'doctorate' in Concat_Degree.lower():
        return "Doctorate's Degree"
    if 'master' in Concat_Degree.lower():
        return "Master's Degree"
    elif 'bachelor' in Concat_Degree.lower():
        return "Bachelor's Degree"
    elif 'associate' in Concat_Degree.lower():
        return "Associate's Degree"
    elif 'technical' in Concat_Degree.lower():
        return "Technical Diploma"
    else:
        return Concat_Degree
mgr_edu['Education'] = mgr_edu.apply(lambda x: find_highest_education(x.Concat_De
##Dropping Duplicates
mgr_edu = mgr_edu.drop_duplicates(subset= ['LPN'],keep='first').reset_index(drop=

### Merging the Education Data with Raw Data:
df3 = pd.merge(left=df2,right=mgr_edu[['LPN','Education']],on='LPN',how='left')
print("Shape of mgr_active_emp :",df.shape)
print("Shape after PerformaceRating :",df3.shape)
df3.sample(3)
```

```
Shape of mgr_active_emp : (156, 15)
Shape after PerformaceRating : (156, 18)
```

Out[6]:

	LPN	Employee Status	Designation	Band	Service Line	Sub Service Line	Work Location	Work Location City	Country/Region
81	1323331	Active	Manager	1	Service Line 4	Sub Service Line 9	India	NaN	India
71	3318287	Active	Manager	2	Service Line 1	Sub Service Line 2	India	NaN	India
110	1327336	Active	Manager	1	Service Line 4	Sub Service Line 8	India	NaN	India

In [7]:

```
## WOrking on Manager Experience:
# We have given start dates and end dates of comparines where Managers have peri
# Now since there are duplicate LPNs so first checking unique Employee Ids
print("Total # of Employees in Workex Sheet :",len(mgr_workex.LPN.unique().tolist

#Creating a new Dataframe which will hold experience information of Employe
df_work_ex = pd.DataFrame(columns=['LPN','Experience_In_Months','Companies_Switch
def calculate_exp(index,df_temp):
    #print("LPN :",df_temp[df_temp['LPN']])
    df_temp = df_temp.sort_values(by='Previous Employment Start Date',ascending=T
    shape = df_temp.shape
    #print(df_temp.shape)
    if df_temp.shape[0]==1:
        start_date = df_temp.iloc[0]['Previous Employment Start Date']
        #start_date = datetime.strptime(df_temp.iloc[0]['Previous Employment Star
        #end_date = datetime.strptime(df_temp.iloc[0]['Previous Employment End Da
        end_date = df_temp.iloc[0]['Previous Employment End Date']
        diff_in_months = (end_date-start_date)/np.timedelta64(1, 'M')
        #df['nb_months'] = ((df.dates1 - df.dates2)/np.timedelta64(1, 'M'))
        #print("Months: ",diff_in_months)
        companies_switched = 1
        relevant_work_ex = df_temp.iloc[0]['Previous Relevant Work Experience (Y/
    else:
        start_date = df_temp.iloc[0]['Previous Employment Start Date']
        end_date = df_temp.iloc[(df_temp.shape[0]-1)]['Previous Employment End Da
        diff_in_months = (end_date-start_date)/np.timedelta64(1, 'M')
        companies_switched = df_temp['Previous Company  Name'].nunique()
        relevant_work_ex = df_temp.iloc[(df_temp.shape[0]-1)]['Previous Relevant 
    df_work_ex.loc[index,['LPN']] = df_temp.iloc[0]['LPN']
    df_work_ex.loc[index,['Experience_In_Months']] = (round(diff_in_months, 0))
    df_work_ex.loc[index,['Companies_Switched']] = companies_switched
    df_work_ex.loc[index,['Previous Employment Start Date']] = start_date
    df_work_ex.loc[index,['Previous Employment End Date']] = end_date
    df_work_ex.loc[index,['Previous Relevant Work Experience (Y/N)']] = relevant_
index=0
for each_LPN in mgr_workex.LPN.unique().tolist():
    df_temp = mgr_workex[mgr_workex['LPN']==each_LPN]
    calculate_exp(index,df_temp)
    index+=1
df_work_ex.sample(3)
```

```
Total # of Employees in Workex Sheet : 161
```

Out[7]:

	LPN	Experience_In_Months	Companies_Switched	Previous Employment Start Date	Previous Employment End Date	Previous Relevant Work Experience (Y/N)
16	2328040	NaN	2	2016-06-01 00:00:00	NaT	Yes
54	2322764	38.0	1	2015-07-16 00:00:00	2018-09-12 00:00:00	Yes
129	2329324	NaN	1	NaT	NaT	NaN

In [8]:

```
### Merging the WOrkEx information in the Master Data:
# Merging the Performance Data:
df4 = pd.merge(left=df3,right=df_work_ex[['LPN','Experience_In_Months','Companies
print("Shape of mgr_active_emp :",df3.shape)
print("Shape after PerformaceRating :",df4.shape)
df4.sample(5)
```

```
Shape of mgr_active_emp : (156, 18)
Shape after PerformaceRating : (156, 21)
```

Out[8]:

	LPN	Employee Status	Designation	Band	Service Line	Sub Service Line	Work Location	Work Location City	Country/Region
149	2328511	Active	Manager	3	Service Line 3	Sub Service Line 7	India	NaN	India
88	3328462	Active	Manager	3	Service Line 3	Sub Service Line 7	India	NaN	India
122	4328126	Active	Manager	1	Service Line 5	Sub Service Line 11	India	NaN	India
1	2329008	Active	Manager	3	Service Line 5	Sub Service Line 11	India	NaN	India
137	4326668	Active	Manager	3	Service Line 5	Sub Service Line 10	India	NaN	India

In [9]:

```
### Working with the leave record Data:
mgr_leaves
```

Out[9]:

	LPN	Designation	Grade - Code (Sort By)	Start Date	End Date	Leave Date	Leave Type	Days	Total Days	Conse\ncutive
0	320230	Manager	1	2020-11-16	2020-11-16	2020-11-16	Full Day	1.0	1.0	1.0
1	320230	Manager	1	2020-12-29	2020-12-29	2020-12-29	Full Day	1.0	1.0	1.0
2	320230	Manager	1	2020-12-30	2020-12-30	2020-12-30	Full Day	1.0	1.0	1.0
3	320230	Manager	1	2020-12-31	2020-12-31	2020-12-31	Full Day	1.0	1.0	1.0
4	320230	Manager	1	2020-12-24	2020-12-24	2020-12-24	Full Day	1.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...
9597	323928	Manager	3	2021-04-19	2021-04-22	2021-04-21	Full Day	1.0	4.0	4.0
9598	323928	Manager	3	2021-04-19	2021-04-22	2021-04-22	Full Day	1.0	4.0	4.0
9599	323928	Manager	3	2021-03-18	2021-03-18	2021-03-18	Full Day	1.0	1.0	1.0
9600	323928	Manager	3	2021-03-26	2021-03-26	2021-03-26	1st Half	0.5	0.5	0.5
9601	323928	Manager	3	2021-04-09	2021-04-09	2021-04-09	2nd Half	0.5	0.5	0.5

9602 rows × 10 columns

In [10]:

```
lpn_leave_records = mgr_leaves.LPN.unique().tolist()
active_mgr_records = df4.LPN.unique().tolist()

def list_diff(list1, list2):
    return (list(list(set(list1)-set(list2)) + list(set(list2)-set(list1))))
diff_values = list_diff(active_mgr_records, lpn_leave_records)
```

In [11]:

```
mgr_leaves.columns
```

Out[11]:

```
Index(['LPN', 'Designation', 'Grade - Code (Sort By)', 'Start Date', 'End Dat
e', 'Leave Date', 'Leave Type', 'Days', 'Total Days', 'Conse\ncutive'], dtype
='object')
```

In [12]:

```
## Working on Leave Data:
print("Total # of Employees in Workex Sheet :",len(mgr_leaves.LPN.unique().tolist

#Creating a new Dataframe which will hold experience information of Employe
df_leaves = pd.DataFrame(columns=['LPN','Max_Leaves_InaYear','Max_ConsecutiveLea
def calculate_leaves(index,df_temp):
    df_temp.set_index('Leave Date',inplace=True)
    ## Resmaple for a year
    df_temp2 = df_temp.resample(rule='A').sum()
    df_temp3 = df_temp.resample(rule='A').max()
    ## Now taking max no of leaves taken in a year
    maxLeaves = df_temp2['Days'].max()
    maxconsecutive = df_temp3['Conse\ncutive'].max()
    #print (df_temp2)
    df_leaves.loc[index,['LPN']] = df_temp.iloc[0]['LPN']
    df_leaves.loc[index,['Max_Leaves_InaYear']] = maxLeaves
    df_leaves.loc[index,['Max_ConsecutiveLeaves_InaYear']] = maxconsecutive

index=0
for each_LPN in mgr_leaves.LPN.unique().tolist():
    df_temp = mgr_leaves[mgr_leaves['LPN']==each_LPN]
    calculate_leaves(index,df_temp)
    index+=1
df_leaves.sample(3)
```

```
Total # of Employees in Workex Sheet : 263
```

Out[12]:

	LPN	Max_Leaves_InaYear	Max_ConsecutiveLeaves_InaYear
198	5323421	48.5	39.0
3	1317533	28.0	7.0
88	320102	1.0	1.0

In [13]:

```
### Merging the Leave information in the Master Data:
df5 = pd.merge(left=df4,right=df_leaves[['LPN','Max_Leaves_InaYear','Max_Consecut
print("Shape of mgr_active_emp :",df4.shape)
print("Shape after PerformaceRating :",df5.shape)
df5.sample(5)
```

```
Shape of mgr_active_emp : (156, 21)
Shape after PerformaceRating : (156, 23)
```

Out[13]:

	LPN	Employee Status	Designation	Band	Service Line	Sub Service Line	Work Location	Work Location City	Country/Region
24	5320011	Active	Manager	1	Service Line 5	Sub Service Line 10	India	NaN	India
151	1329044	Active	Manager	2	Service Line 2	Sub Service Line 5	India	NaN	India
102	4328803	Active	Manager	2	Service Line 2	Sub Service Line 4	India	NaN	India
79	4326666	Active	Manager	1	Service Line 1	Sub Service Line 1	India	NaN	India
6	2327207	Active	Manager	2	Service Line 1	Sub Service Line 1	India	NaN	India

Working on Exited Employees List

In [14]:

```
start = datetime.datetime.now()
exited_emp_list = pd.read_excel(path+'Managers_Exit Employees.xlsx', sheet_name =
exit_emp_edu = pd.read_excel(path+'Managers_Exit Employees.xlsx', sheet_name = 'E
exit_emp_workex = pd.read_excel(path+'Managers_Exit Employees.xlsx', sheet_name =
exit_emp_certs = pd.read_excel(path+'Managers_Exit Employees.xlsx', sheet_name =
end_time = datetime.datetime.now()
print("Time Consumed in reading Files--> ",(end_time-start))
```

```
Time Consumed in reading Files-->  0:00:00.344996
```

In [15]:

```
### WOrking on the Education Section. Few entries are duplicated like where peopl
# Concat Degree if duplicate
exit_emp_edu_temp = exit_emp_edu.groupby(['LPN'])['Degree'].transform(lambda x: '
# Creating DF from Series
exit_emp_edu_temp = pd.DataFrame({'Concat_Degree':exit_emp_edu_temp.values})
exit_emp_edu=pd.concat([exit_emp_edu,exit_emp_edu_temp],axis=1)
exit_emp_edu['Education'] = exit_emp_edu.apply(lambda x: find_highest_education(x
##Dropping Duplicates
exit_emp_edu = exit_emp_edu.drop_duplicates(subset= ['LPN'],keep='first').reset_i

### Merging the Education Data with Raw Data:
print("Shape of exited_emp_list :",exited_emp_list.shape)
exited_emp_list = pd.merge(left=exited_emp_list,right=exit_emp_edu[['LPN','Educat
print("Shape after Education    :",exited_emp_list.shape)
exited_emp_list.sample(3)
```

```
Shape of exited_emp_list : (160, 18)
Shape after Education    : (160, 19)
```

Out[15]:

	LPN	Employee Status	Grade	Rank Name	Service Line	Sub Service Line	Region	Country/Region	Event	
33	1616536	Terminated	7	Manager	Service Line 2	Sub Service Line 3	Region 3	India	Termination	S
1	1603404	Terminated	7	Manager	Service Line 1	Sub Service Line 2	Region 5	India	Termination	S
29	1601423	Terminated	7	Manager	Service Line 1	Sub Service Line 2	Region 4	India	Termination	S

In [16]:

```
## WOrking on Manager Experience:
# We have given start dates and end dates of comparines where Managers have peric
# Now since there are duplicate LPNs so first checking unique Employee Ids
print("Total # of Employees in Workex Sheet :",len(exit_emp_workex.LPN.unique().t

#Creating a new Dataframe which will hold experience information of Employe
df_work_ex = pd.DataFrame(columns=['LPN','Experience_In_Months','Companies_Switch
def calculate_exp(index,df_temp):
    #print("LPN :",df_temp[df_temp['LPN']])
    df_temp = df_temp.sort_values(by='Previous Employment Start Date',ascending=T
    shape = df_temp.shape
    #print(df_temp.shape)
    if df_temp.shape[0]==1:
        start_date = df_temp.iloc[0]['Previous Employment Start Date']
        #start_date = datetime.strptime(df_temp.iloc[0]['Previous Employment Star
        #end_date = datetime.strptime(df_temp.iloc[0]['Previous Employment End Da
        end_date = df_temp.iloc[0]['Previous Employment End Date']
        diff_in_months = (end_date-start_date)/np.timedelta64(1, 'M')
        #df['nb_months'] = ((df.dates1 - df.dates2)/np.timedelta64(1, 'M'))
        #print("Months: ",diff_in_months)
        companies_switched = 1
        relevant_work_ex = df_temp.iloc[0]['Previous Relevant Work Experience (Y/
    else:
        start_date = df_temp.iloc[0]['Previous Employment Start Date']
        end_date = df_temp.iloc[(df_temp.shape[0]-1)]['Previous Employment End Da
        diff_in_months = (end_date-start_date)/np.timedelta64(1, 'M')
        companies_switched = df_temp['Previous Company  Name'].nunique()
        relevant_work_ex = df_temp.iloc[(df_temp.shape[0]-1)]['Previous Relevant
    df_work_ex.loc[index,['LPN']] = df_temp.iloc[0]['LPN']
    df_work_ex.loc[index,['Experience_In_Months']] = (round(diff_in_months, 0))
    df_work_ex.loc[index,['Companies_Switched']] = companies_switched
    df_work_ex.loc[index,['Previous Employment Start Date']] = start_date
    df_work_ex.loc[index,['Previous Employment End Date']] = end_date
    df_work_ex.loc[index,['Previous Relevant Work Experience (Y/N)']] = relevant_

index=0
for each_LPN in exit_emp_workex.LPN.unique().tolist():
    df_temp = exit_emp_workex[exit_emp_workex['LPN']==each_LPN]
    calculate_exp(index,df_temp)
    index+=1
df_work_ex.sample(3)
```

```
Total # of Employees in Workex Sheet : 71
```

Out[16]:

	LPN	Experience_In_Months	Companies_Switched	Previous Employment Start Date	Previous Employment End Date	Previous Relevant Work Experience (Y/N)
64	328236	NaN	1	2014-03-01 00:00:00	NaT	NaN
58	325632	10.0	1	2015-03-01 00:00:00	2016-01-01 00:00:00	No

	LPN	Experience_In_Months	Companies_Switched	Previous Employment Start Date	Previous Employment End Date	Previous Relevant Work Experience (Y/N)
31	327258	96.0	4	2013-01-01 00:00:00	2021-01-01 00:00:00	NaN

In [17]:

```
### Merging the WOrkEx information in the Master Data:
# Merging the Performance Data:
print("Shape of mgr_active_emp :",exited_emp_list.shape)
exited_emp_list = pd.merge(left=exited_emp_list,right=df_work_ex[['LPN','Experier
print("Shape after PerformaceRating :",exited_emp_list.shape)
exited_emp_list.sample(5)
```

```
Shape of mgr_active_emp : (160, 19)
Shape after PerformaceRating : (160, 22)
```

Out[17]:

	LPN	Employee Status	Grade	Rank Name	Service Line	Sub Service Line	Region	Country/Region	Event
141	324172	Terminated	7	Manager	Service Line 4	Sub Service Line 8	Region 5	India	Termination
82	323270	Terminated	7	Manager	Service Line 1	Sub Service Line 2	Region 2	India	Termination
25	1609384	Terminated	7	Manager	Service Line 5	Sub Service Line 11	Region 4	India	Termination
81	326750	Terminated	7	Manager	Service Line 1	Sub Service Line 1	Region 5	India	Termination
10	1608417	Terminated	7	Manager	Service Line 2	Sub Service Line 4	Region 1	India	Termination

In [18]:

```
## WOrking on Exited Employee Certifications
## Counting the total certification for Employee
exit_emp_certs = exit_emp_certs.groupby('LPN').count()[['Grade','Designation']]
exit_emp_certs = exit_emp_certs.rename(columns={'Grade':'Count_Certifications'}).
exit_emp_certs.sample(5)
print("Shape of mgr_active_emp :",exited_emp_list.shape)
exited_emp_list = pd.merge(left=exited_emp_list,right=exit_emp_certs[['LPN','Cour
print("Shape after Certification Info Addition :",exited_emp_list.shape)
exited_emp_list.sample(5)
```

```
Shape of mgr_active_emp : (160, 22)
Shape after Certification Info Addition : (160, 23)
```

Out[18]:

	LPN	Employee Status	Grade	Rank Name	Service Line	Sub Service Line	Region	Country/Region	Event
64	318606	Terminated	7	Manager	Service Line 1	Sub Service Line 1	Region 1	India	Termination
26	1602122	Terminated	7	Manager	Service Line 3	Sub Service Line 7	Region 1	India	Termination
13	1605764	Terminated	7	Manager	Service Line 2	Sub Service Line 3	Region 3	India	Termination
57	320033	Terminated	7	Manager	Service Line 5	Sub Service Line 11	Region 3	India	Termination
132	319149	Terminated	7	Manager	Service Line 3	Sub Service Line 6	Region 5	India	Termination

In [19]:

```
exited_emp_list = pd.merge(left=exited_emp_list,right=df_work_ex[['LPN','Experien
print("Shape after PerformaceRating :",exited_emp_list.shape)
exited_emp_list.sample(5)
```

```
Shape after PerformaceRating : (160, 26)
```

Out[19]:

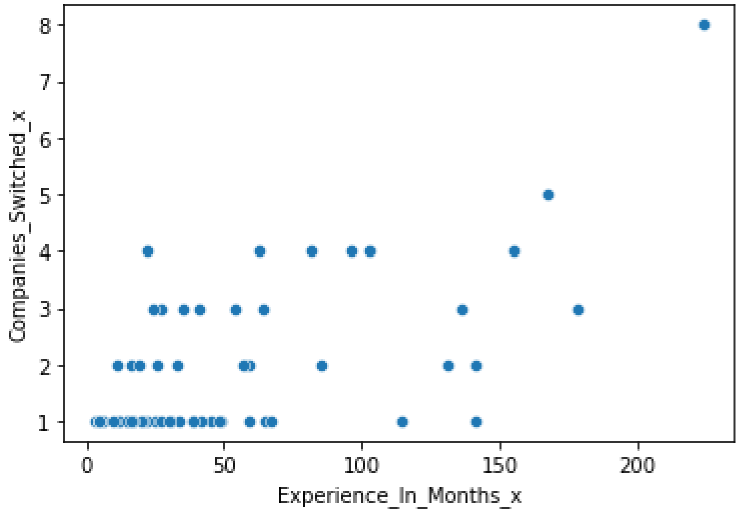
	LPN	Employee Status	Grade	Rank Name	Service Line	Sub Service Line	Region	Country/Region	Event
125	325026	Terminated	7	Manager	Service Line 5	Sub Service Line 11	Region 5	India	Termination
80	322115	Terminated	7	Manager	Service Line 2	Sub Service Line 5	Region 4	India	Termination
91	325178	Terminated	7	Manager	Service Line 5	Sub Service Line 11	Region 4	India	Termination
14	1617310	Terminated	7	Manager	Service Line 4	Sub Service Line 9	Region 1	India	Termination
77	323408	Terminated	7	Manager	Service Line 5	Sub Service Line 10	Region 5	India	Termination

In [20]:

```
#################################################################################
####################### FINAL DATAFRAMES FOR ANALYSIS ###########################
exited_emp_list.to_csv('Exited_Employee_Data.csv',index=False)
df5.to_csv('Active_Employee_Data.csv',index=False)
```

```
In [22]: sns.scatterplot(x="Experience_In_Months_x", y="Companies_Switched_x", data=exited
```

```
Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x298917bcf48>
```



```
In [25]: employee_Data = pd.concat([exited_emp_list, df5],ignore_index = True)
```

```
In [26]: employee_Data.to_csv('Total_Employee.csv',index=False)
```