In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn.preprocessing as prs
import seaborn as sns
import plotly.express as px
```

In [129…

```
url='/Users/sumedhajauhari/Downloads/WA_Fn-UseC_-HR-Employee-Attrition.csv'
df=pd.read_csv(url)
df.sample(10)
```

Out[129]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationI
493	44	No	Travel_Rarely	1112	Human Resources	1	4	Life Scie
889	27	No	Travel_Rarely	1103	Research & Development	14	3	Life Scie
470	24	No	Travel_Frequently	535	Sales	24	3	Me
1421	47	No	Non-Travel	1162	Research & Development	1	1	Me
869	46	No	Travel_Rarely	1450	Research & Development	15	2	Life Scie
564	45	No	Travel_Rarely	954	Sales	2	2	Tech De
1057	29	Yes	Travel_Frequently	115	Sales	13	3	Tech De
1444	56	Yes	Travel_Rarely	310	Research & Development	7	2	Tech De
441	42	No	Travel_Frequently	1474	Research & Development	5	2	C
219	54	No	Travel_Rarely	1147	Sales	3	3	Mark

10 rows × 35 columns

In [9]:

```
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
```

In [10]:
```
df.describe()
```

Out[10]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	En
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	

8 rows × 26 columns

In [16]:
```
df.describe(include=['object'])
```

Out[16]:

	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus	Over18
count	1470	1470	1470	1470	1470	1470	1470	1470
unique	2	3	3	6	2	9	3	1
top	No	Travel_Rarely	Research & Development	Life Sciences	Male	Sales Executive	Married	Y
freq	1233	1043	961	606	882	326	673	1470

In [17]:
```
df.describe(include="all")
```

Out[17]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educatior
count	1470.000000	1470	1470	1470.000000	1470	1470.000000	1470.00000
unique	NaN	2	3	NaN	3	NaN	Na
top	NaN	No	Travel_Rarely	NaN	Research & Development	NaN	Na

freq	NaN	1233	1043	NaN	961	NaN	Na
mean	36.923810	NaN	NaN	802.485714	NaN	9.192517	2.91292
std	9.135373	NaN	NaN	403.509100	NaN	8.106864	1.02416
min	18.000000	NaN	NaN	102.000000	NaN	1.000000	1.00000
25%	30.000000	NaN	NaN	465.000000	NaN	2.000000	2.00000
50%	36.000000	NaN	NaN	802.000000	NaN	7.000000	3.00000
75%	43.000000	NaN	NaN	1157.000000	NaN	14.000000	4.00000
max	60.000000	NaN	NaN	1499.000000	NaN	29.000000	5.00000

11 rows × 35 columns

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
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
```

In [20]:
```
df.duplicated().sum()
```

Out[20]:
```
0
```

In [21]:
```
df=df.drop(['EmployeeCount',"Over18","StandardHours"],axis=1)
```

In [121…
```
df.columns
```

```
 Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
Out[121]:         'DistanceFromHome', 'Education', 'EducationField', 'EmployeeNumber',
                  '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 [130…

```
for col in df.columns:
    if df[col].dtype=='object':
        print(col,df[col].unique(),"\n")
```

```
Attrition ['Yes' 'No']

BusinessTravel ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']

Department ['Sales' 'Research & Development' 'Human Resources']

EducationField ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']

Gender ['Female' 'Male']

JobRole ['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']

MaritalStatus ['Single' 'Married' 'Divorced']

Over18 ['Y']

OverTime ['Yes' 'No']
```

In [131…

```
df_Not=df[df['Attrition']=="No"]
df_=df[df['Attrition']=='Yes']
print("Attrition is",df_.shape[0],"Employee")
print("Not Attrition is",df_Not.shape[0],"Employee")
```

```
Attrition is 237 Employee
Not Attrition is 1233 Employee
```

In [115…

```
#check the ratio from Male to female in Not Attrition
df1=df_Not.Gender.value_counts()/df_Not.shape[0]*100
df1
```

Out[115]:

```
Series([], Name: Gender, dtype: float64)
```

In [42]:

```
#check the ratio from Male to female in Attrition
df2=df_.Gender.value_counts()/df_.shape[0]*100
df2
```

Out[42]:

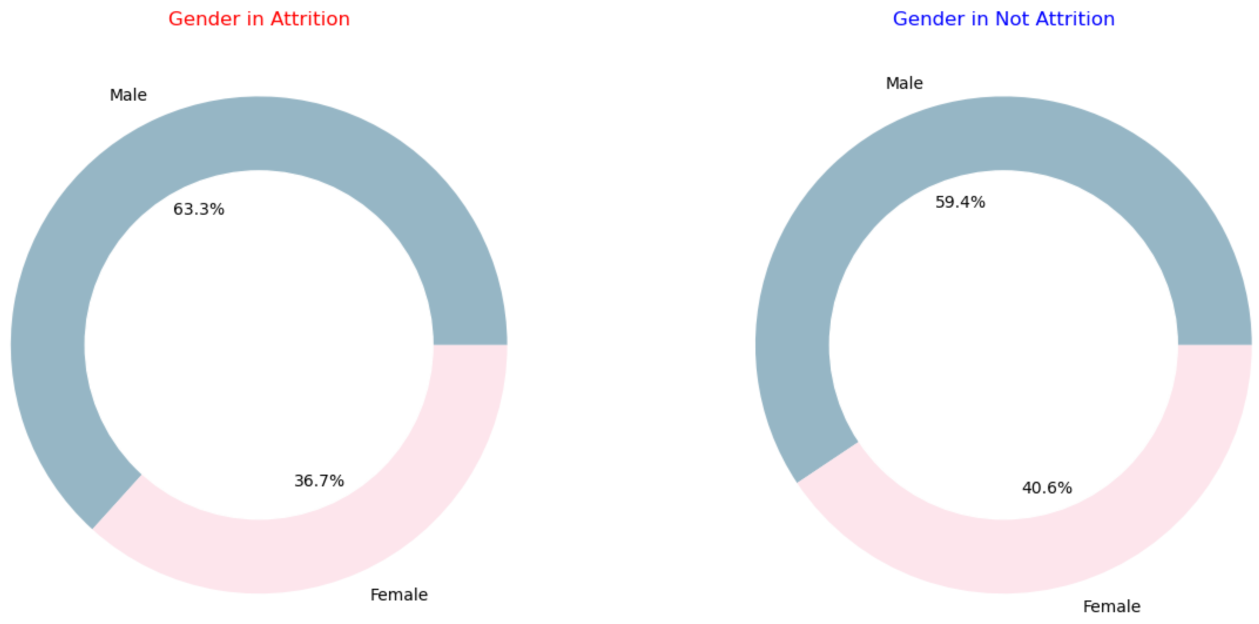
```
Male      63.291139
Female    36.708861
Name: Gender, dtype: float64
```

In [60]:

```
lbl=df_.Gender.value_counts().index.to_list()
plt.figure(figsize=(15,7))
plt.subplot(1,2,1)
plt.pie(df_.Gender.value_counts(),labels=lbl,autopct="%1.1f%%",colors=["#96B6C5","#FDE5E
plt.title("Gender in Attrition",color='red')
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)
```

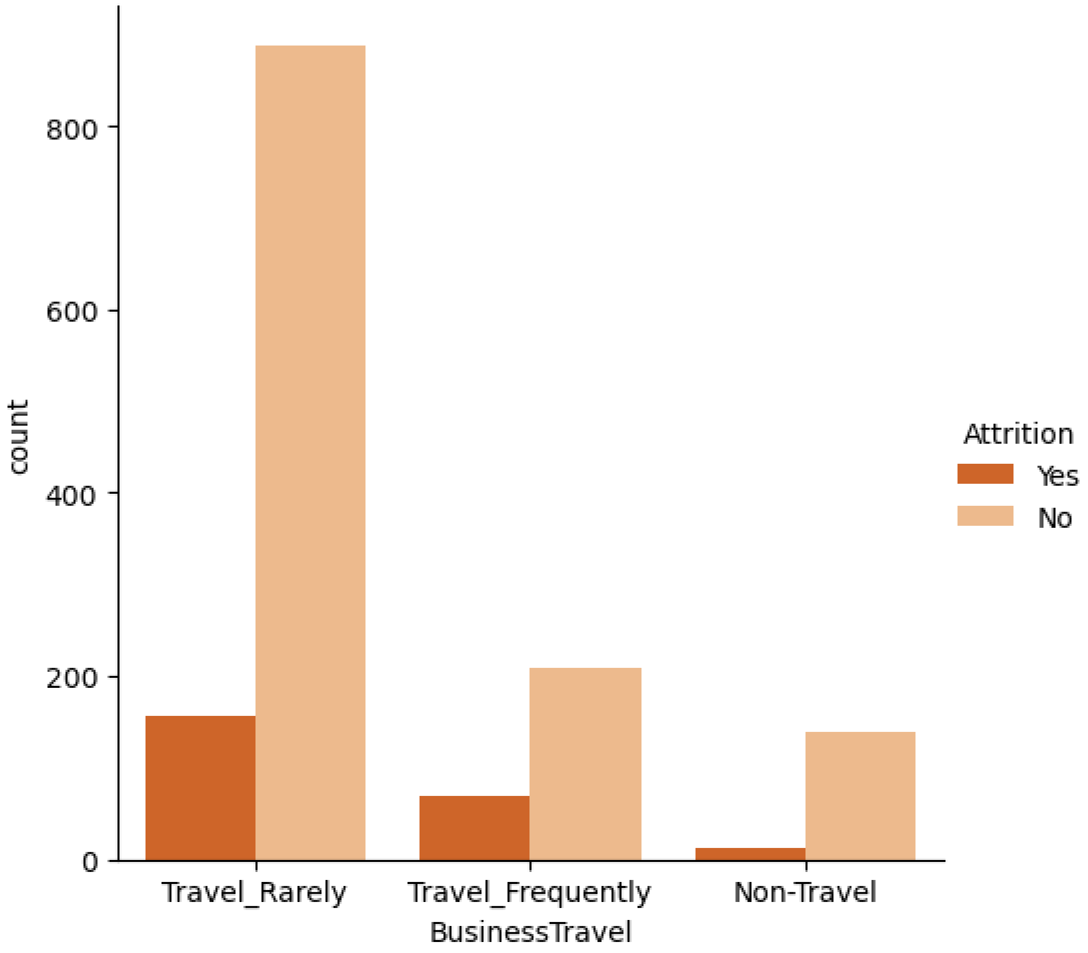
```
plt.subplot(1,2,2)
lbl2=df_Not.Gender.value_counts().index.to_list()
plt.pie(df_Not.Gender.value_counts(),labels=lbl2,autopct="%1.1f%%",colors=['#96B6C5','#F
plt.title("Gender in Not Attrition",color='blue')
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)
```

Out[60]:
```
<matplotlib.patches.Circle at 0x151e4e850>
```



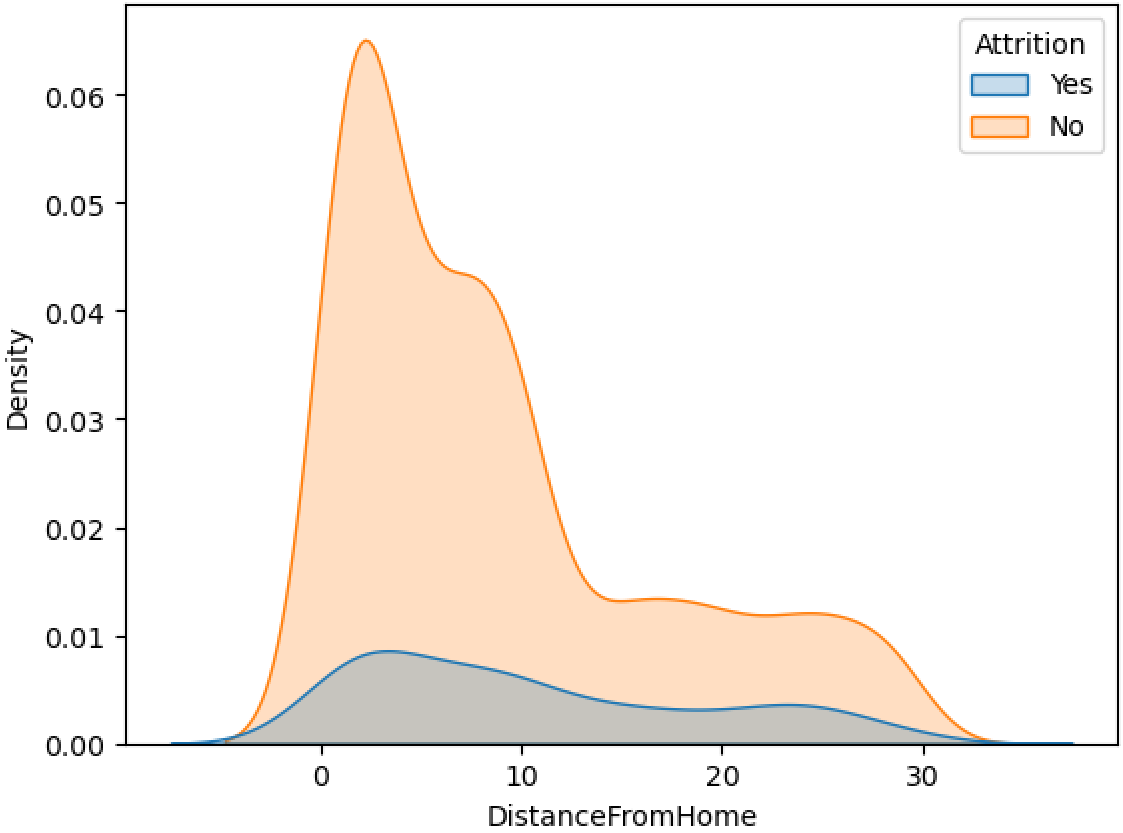
In [61]:
```
sns.catplot(data=df, x="BusinessTravel", kind="count",hue='Attrition', palette="Oranges_
```

Out[61]:
```
<seaborn.axisgrid.FacetGrid at 0x151e5de50>
```



In [63]:
```
sns.kdeplot(data=df,x="DistanceFromHome",hue="Attrition",fill=True)
```

Out[63]:
```
<Axes: xlabel='DistanceFromHome', ylabel='Density'>
```

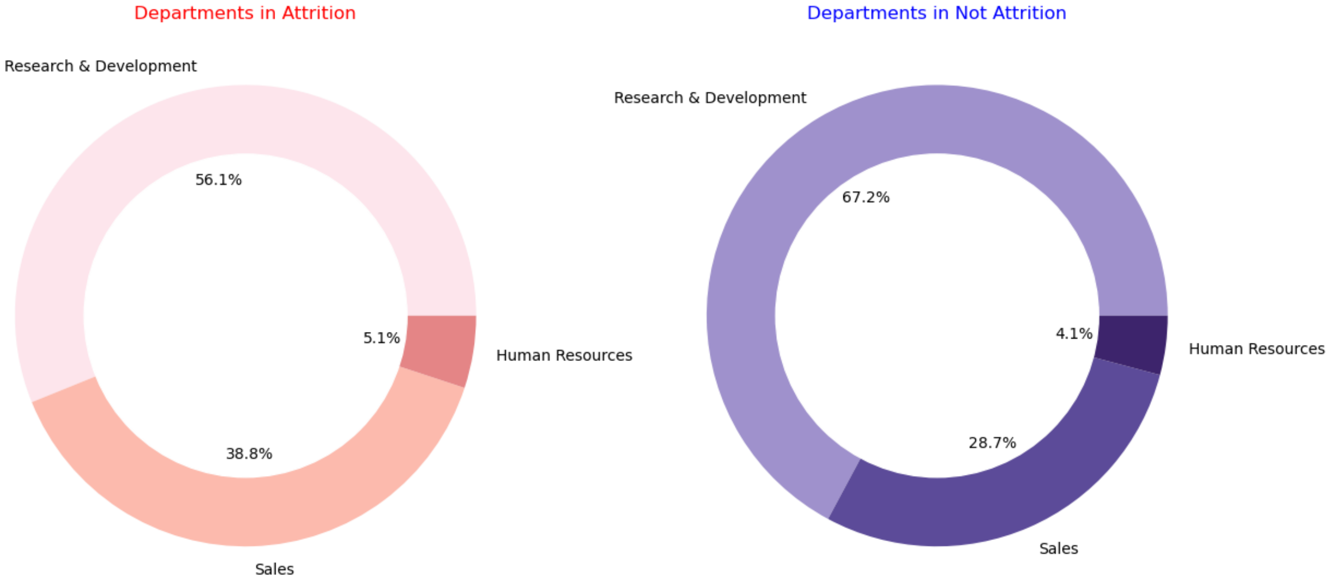


In [92]:
```
l=df_.Department.value_counts().values.to_list()
l
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[92], line 1
----> 1 l=df_.Department.value_counts().values.to_list()
      2 l

AttributeError: 'numpy.ndarray' object has no attribute 'to_list'
```

In [67]:
```
plt.figure(figsize=(15,7))
plt.subplot(1,2,1)
plt.pie(df_.Department.value_counts(),labels=l,autopct="%1.1f%%",colors=["#FDE5EC",'#FCB
plt.title("Departments in Attrition",color='red')
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.subplot(1,2,2)

plt.pie(df_Not.Department.value_counts(),labels=l,autopct="%1.1f%%",colors=["#9F91CC",'#
plt.title("Departments in Not Attrition",color='blue')
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)
```

Out[67]:
```
<matplotlib.patches.Circle at 0x151ec7ad0>
```

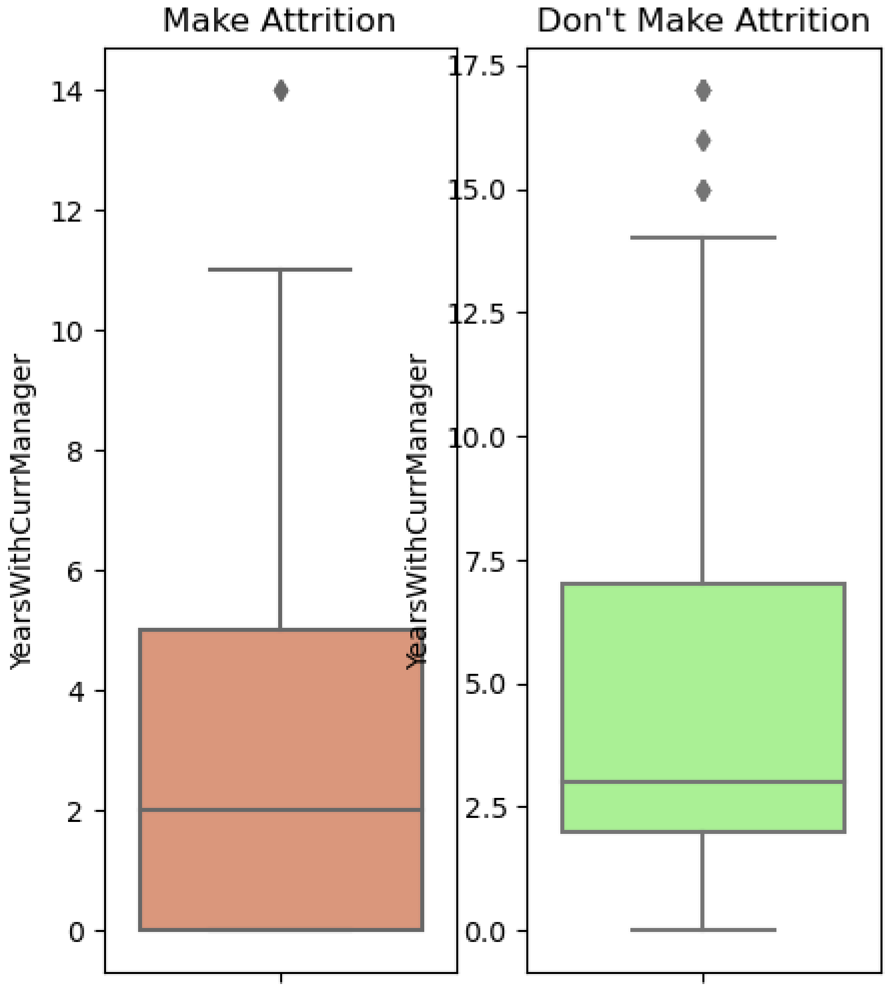


In [76]:

```
plt.figure(figsize=(5,6))
plt.subplot(1,2,1)
sns.boxplot(y=df_.YearsWithCurrManager,palette=['#EA906C'])
plt.title("Make Attrition")
plt.subplot(1,2,2)
sns.boxplot(y=df_Not.YearsWithCurrManager,palette=['#A2FF86'])
plt.title("Don't Make Attrition")
```

Out[76]:

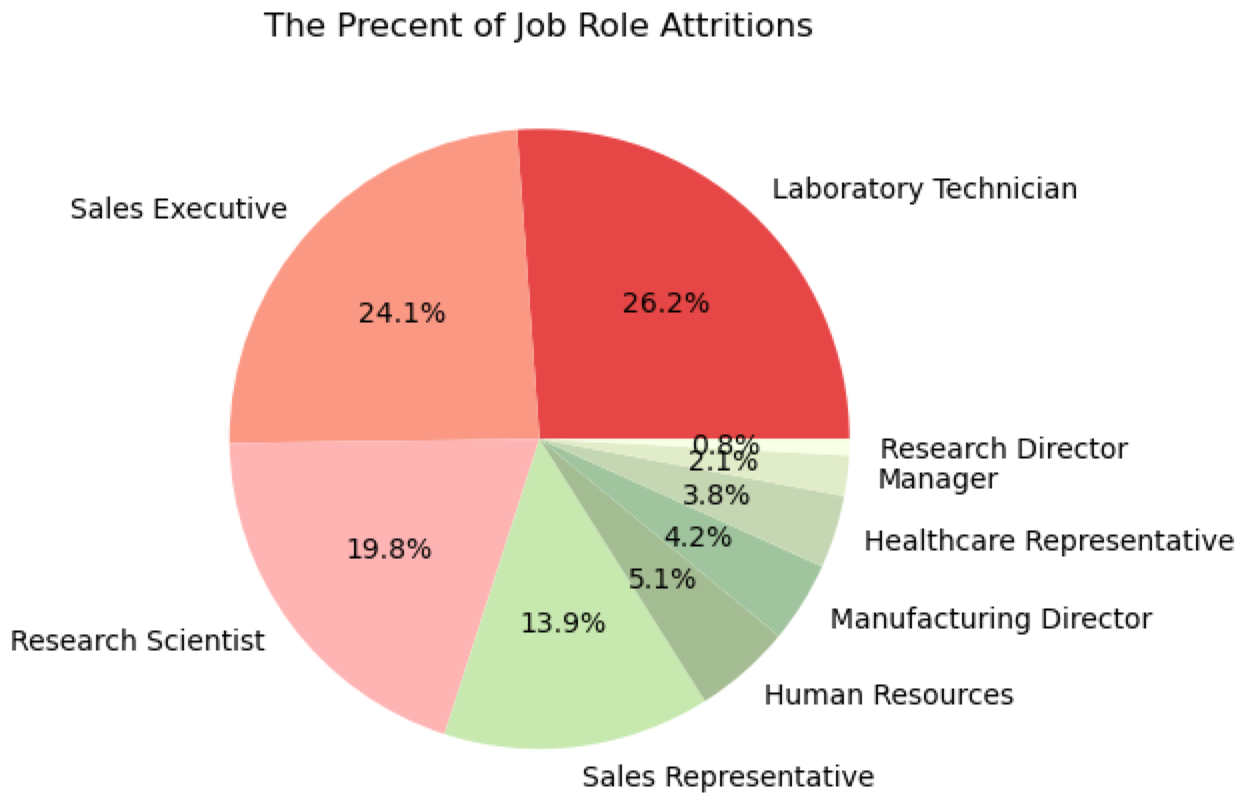
```
Text(0.5, 1.0, "Don't Make Attrition")
```



In [78]:

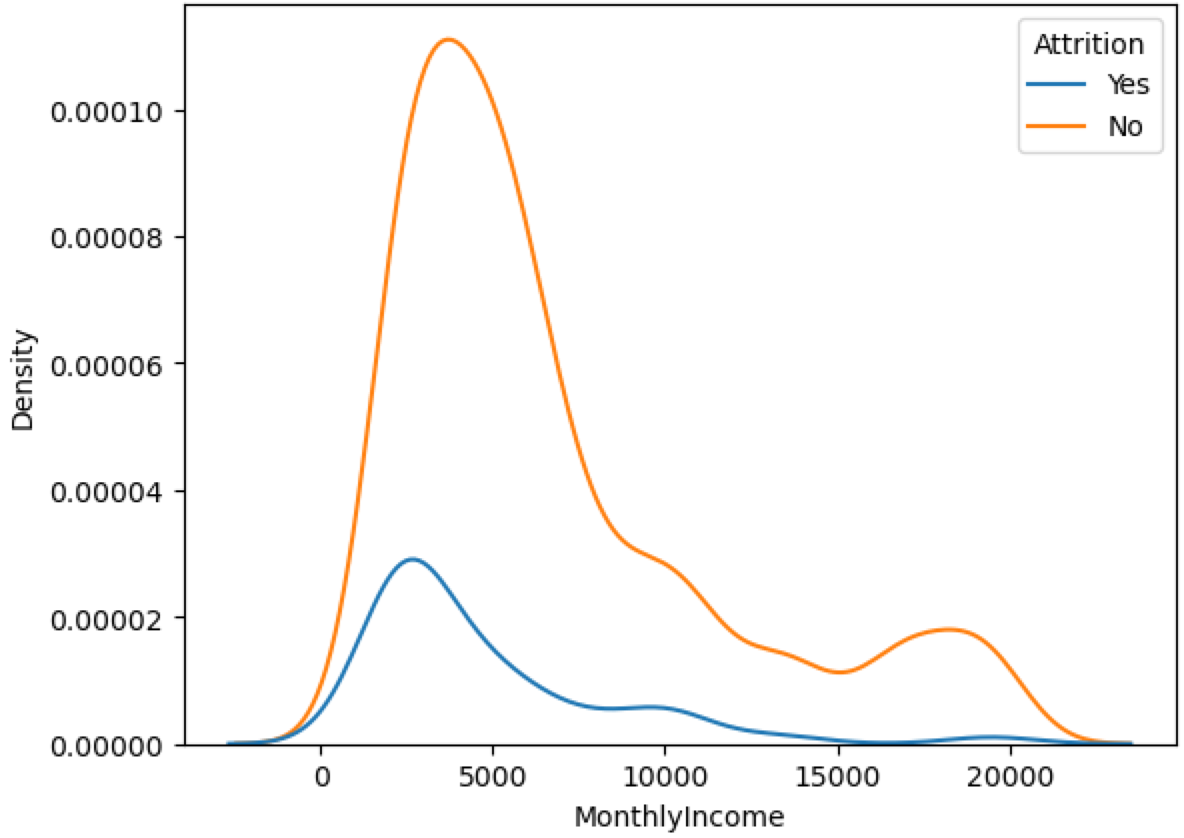
```
plt.figure(figsize=(5,10))
lbl=df_.JobRole.value_counts().index.to_list()
```

```
plt.pie(df_.JobRole.value_counts(),labels=lbl,colors=['#E74646','#FA9884','#FFB4B4','#C7
plt.title("The Precent of Job Role Attritions")
plt.show()
```



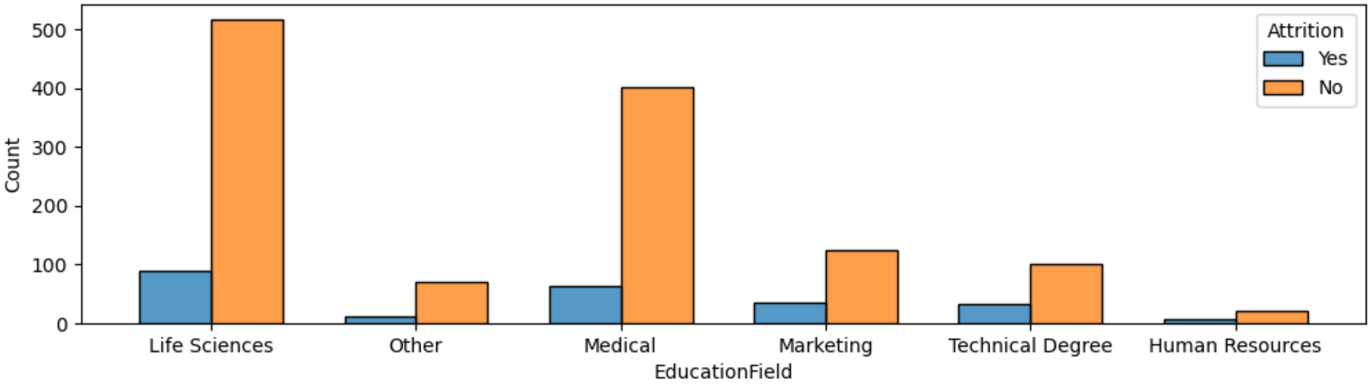
In [80]:
```
sns.kdeplot(x=df.MonthlyIncome,hue=df.Attrition) #palette='viridis'
```

Out[80]:
```
<Axes: xlabel='MonthlyIncome', ylabel='Density'>
```



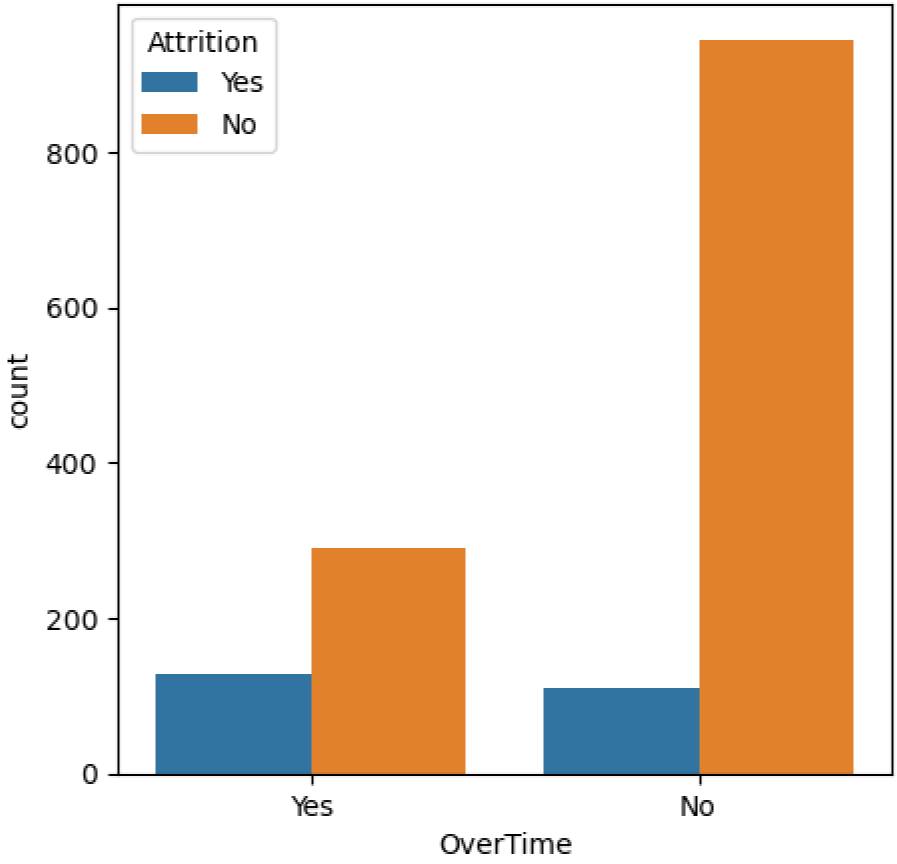
In [91]:
```
plt.figure(figsize=(12,3))
sns.histplot(data=df,x="EducationField",shrink=.7,multiple='dodge',hue='Attrition')#pale
```

```
Out[91]: <Axes: xlabel='EducationField', ylabel='Count'>
```



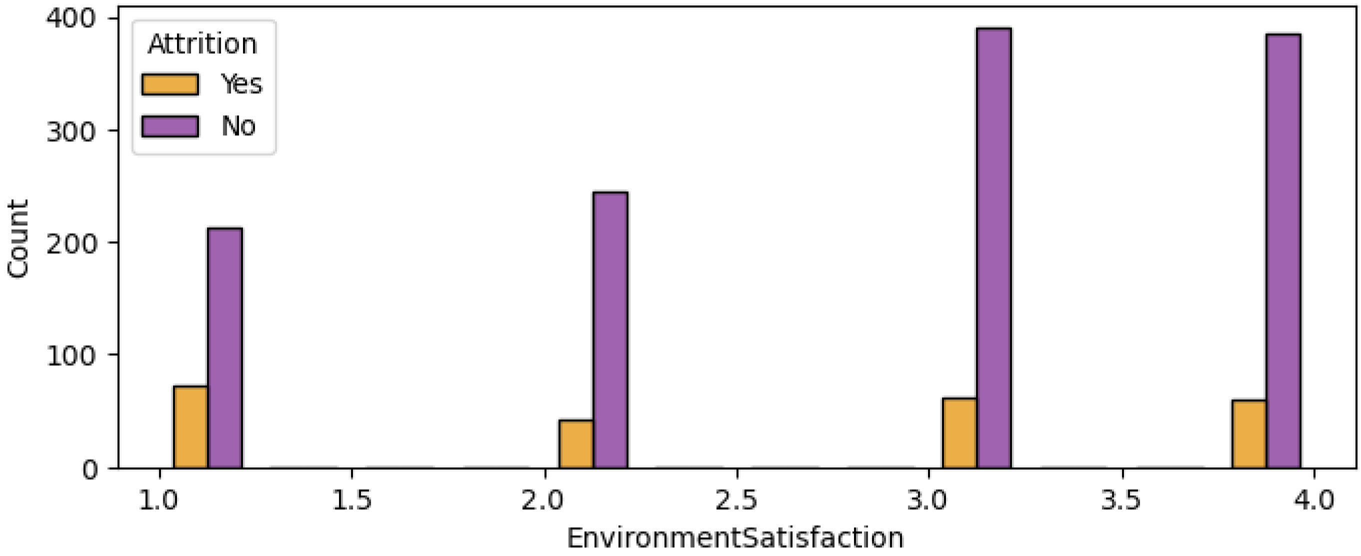
```
In [95]: plt.figure(figsize=(5,5))
         sns.countplot(data=df,x="OverTime",hue='Attrition')#,palette='ocean_r'
```

```
Out[95]: <Axes: xlabel='OverTime', ylabel='count'>
```



```
In [97]: plt.figure(figsize=(8,3))
         sns.histplot(data=df,x="EnvironmentSatisfaction",shrink=.7,multiple='dodge',hue='Attriti
```

```
Out[97]: <Axes: xlabel='EnvironmentSatisfaction', ylabel='Count'>
```

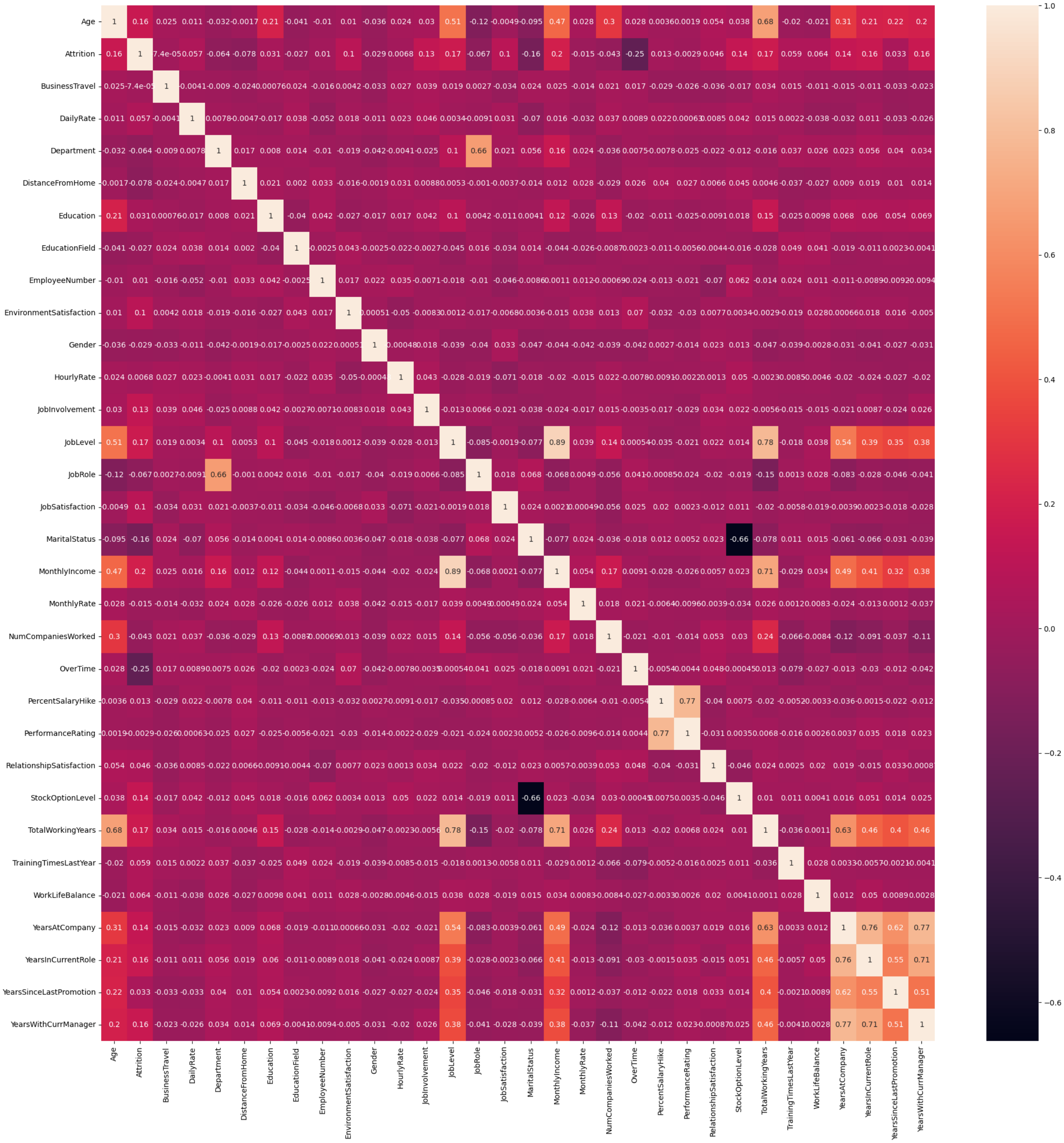


In [99]:

```
 df.Attrition=df_.Attrition.replace({'Yes':1,'No':0})
# print(df_.Attrition.value_counts())
df=df.apply(prs.LabelEncoder().fit_transform)
 plt.figure(figsize=(25,25))
sns.heatmap(df.corr(),annot=True)
```

Out[99]:

```
<Axes: >
```



In [140…

```
plt.figure(figsize=(10,7))

Top_Product = df_.groupby(["Age"]).count().sort_values("Attrition",ascending=False).head
Top_Product = Top_Product[["Attrition"]]#.round(2)
Top_Product.reset_index(inplace=True)
print(Top_Product)

Top_Product2 = df_Not.groupby(["Age"]).count().sort_values("Attrition",ascending=False).
Top_Product2 = Top_Product2[["Attrition"]].round(2)
Top_Product2.reset_index(inplace=True)
#print(Top_Product2)
```

```
   Age  Attrition
0   31         18
1   29         18
2   28         14
3   33         12
4   26         12
```

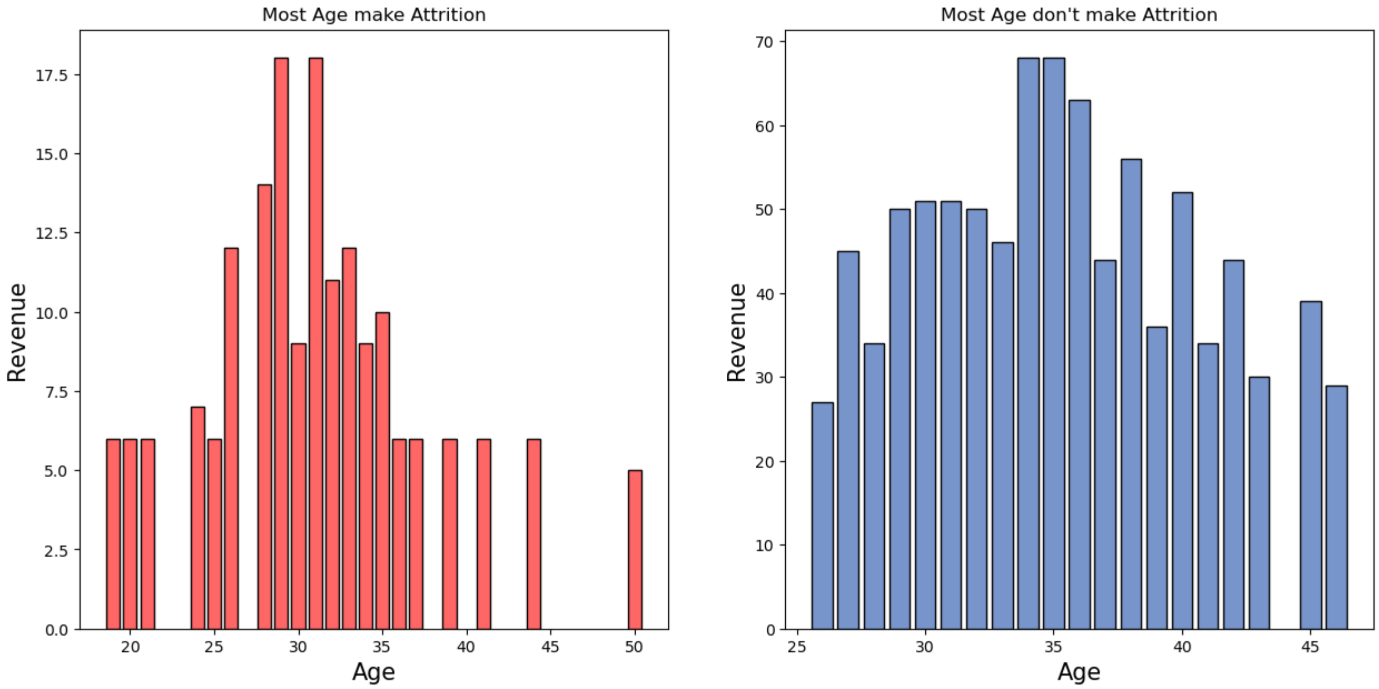
```
5     32          11
6     35          10
7     30           9
8     34           9
9     24           7
10    19           6
11    36           6
12    37           6
13    39           6
14    41           6
15    25           6
16    44           6
17    21           6
18    20           6
19    50           5
<Figure size 1000x700 with 0 Axes>
```

In [141…

```
plt.figure(figsize = (15,7))
plt.subplot(1,2,1)
plt.title("Most Age make Attrition")
plt.bar(Top_Product["Age"], Top_Product["Attrition"],color='#FF6666',edgecolor="k", line
plt.xlabel("Age",fontsize=15) # x axis shows the customers
plt.ylabel("Revenue",fontsize=15) # y axis shows the Revenue
plt.subplot(1,2,2)
plt.title("Most Age don't make Attrition")
plt.bar(Top_Product2["Age"], Top_Product2["Attrition"],color='#7895CB',edgecolor="k", li
plt.xlabel("Age",fontsize=15) # x axis shows the customers
plt.ylabel("Revenue",fontsize=15) # y axis shows the Revenue
```

Out[141]:

```
Text(0, 0.5, 'Revenue')
```



In [143…

```
df_.JobSatisfaction.value_counts()
d=df_.JobSatisfaction.value_counts().index.to_list()
dn=df_Not.JobSatisfaction.value_counts().index.to_list()
d
```

Out[143]:

```
[3, 1, 4, 2]
```

In [145…

```
df_["JobSatisfaction"].head(10)
```

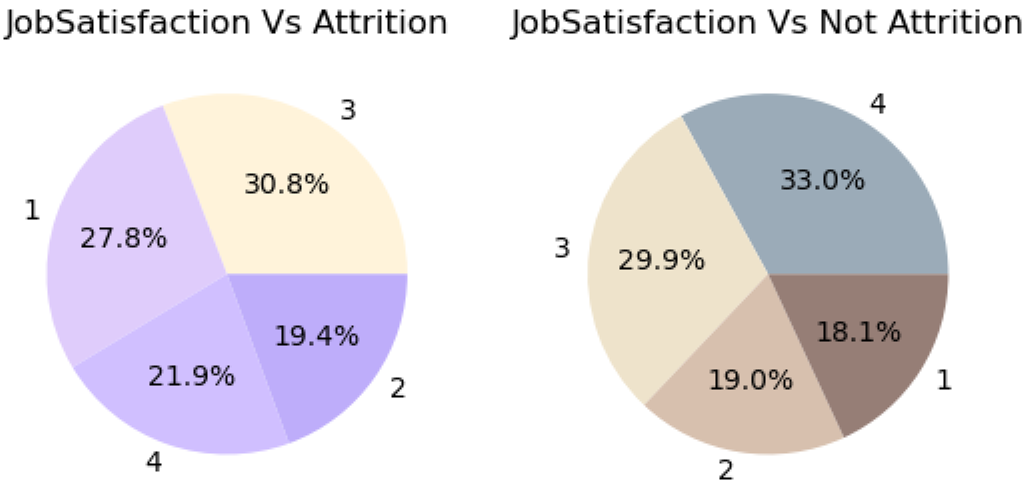
Out[145]:

```
0     4
2     3
14    3
```

```
21     1
24     1
26     1
33     4
34     4
36     3
42     3
Name: JobSatisfaction, dtype: int64
```

In [146…

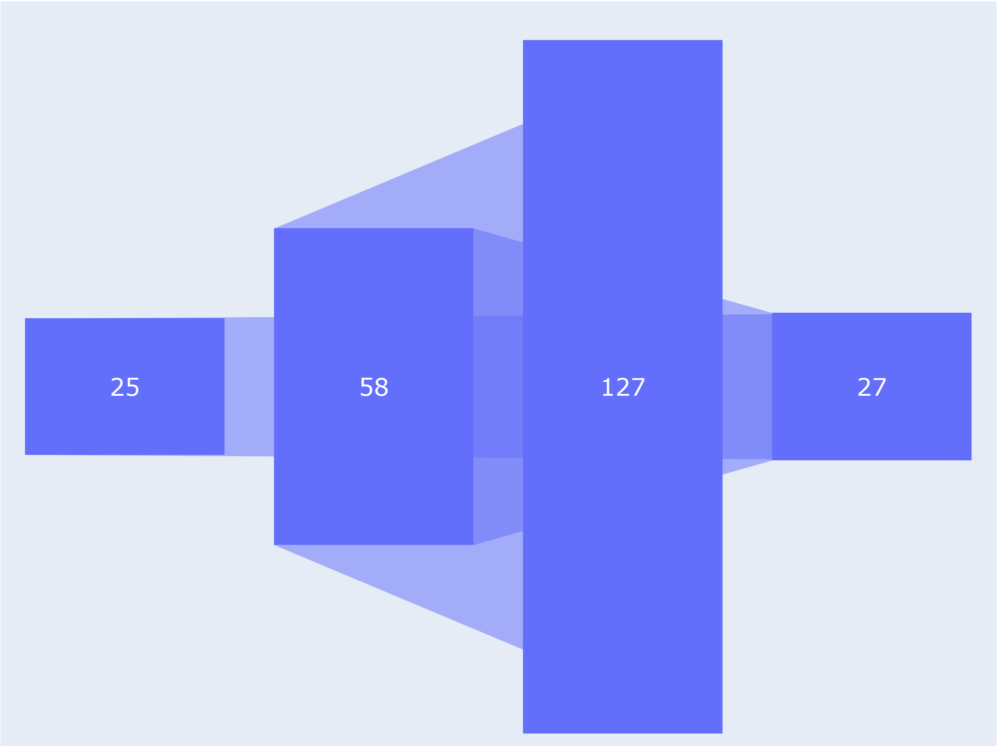
```
plt.subplot(1,2,1)
plt.pie(df_.JobSatisfaction.value_counts(),labels=d,colors=['#FFF3DA','#DFCCFB','#D0BFFF
plt.title("JobSatisfaction Vs Attrition")
plt.subplot(1,2,2)
plt.pie(df_Not.JobSatisfaction.value_counts(),labels=dn,colors=['#9BABB8','#EEE3CB','#D7
plt.title("JobSatisfaction Vs Not Attrition")

plt.show()
```

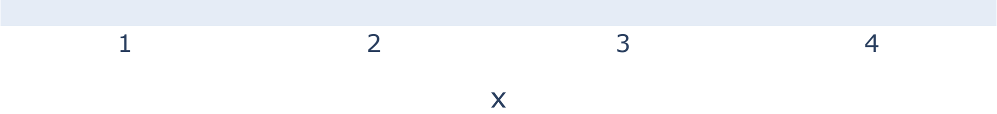

JobSatisfaction Vs Attrition

JobSatisfaction Vs Not Attrition

In [147…

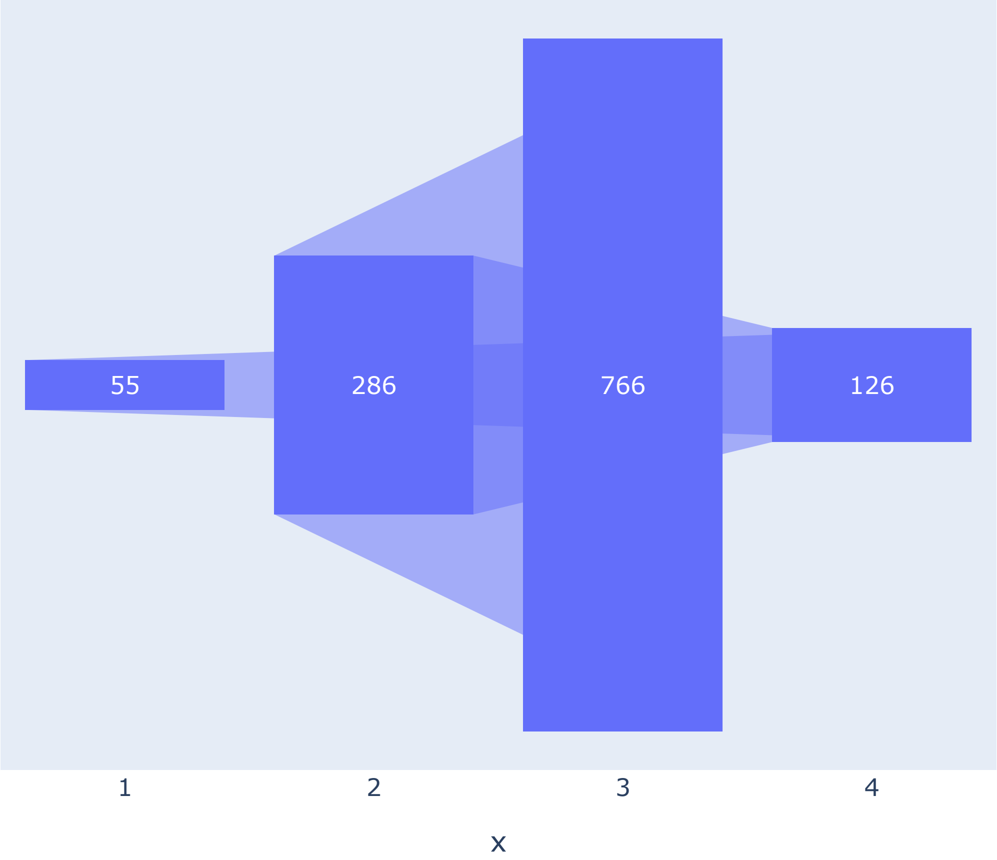
```
fig = px.funnel(df_, x=df_.WorkLifeBalance.value_counts().index.to_list(), y=df_.WorkLif
fig.show()
```





In [148…

```
fig = px.funnel(df_Not, x=df_Not.WorkLifeBalance.value_counts().index.to_list(), y=df_No
fig.show()
```

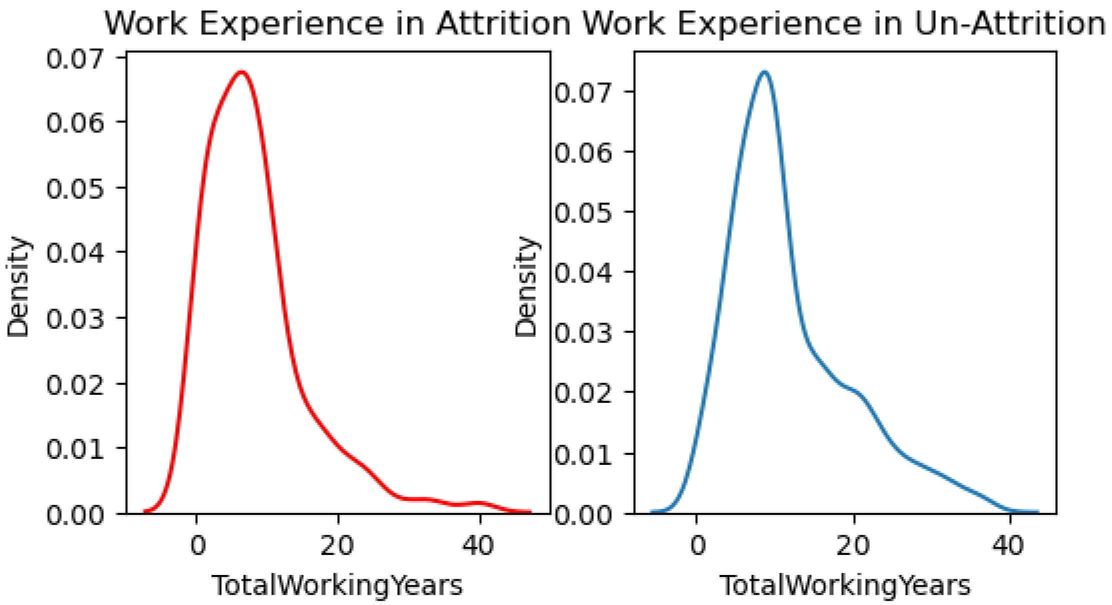


In [152…

```
plt.figure(figsize=(6,3))
plt.subplot(1,2,1)
sns.kdeplot(df_['TotalWorkingYears'],color='red')
plt.title("Work Experience in Attrition")
plt.subplot(1,2,2)
sns.kdeplot(df_Not['TotalWorkingYears'])
plt.title("Work Experience in Un-Attrition")
```

Out[152]:

```
Text(0.5, 1.0, 'Work Experience in Un-Attrition')
```

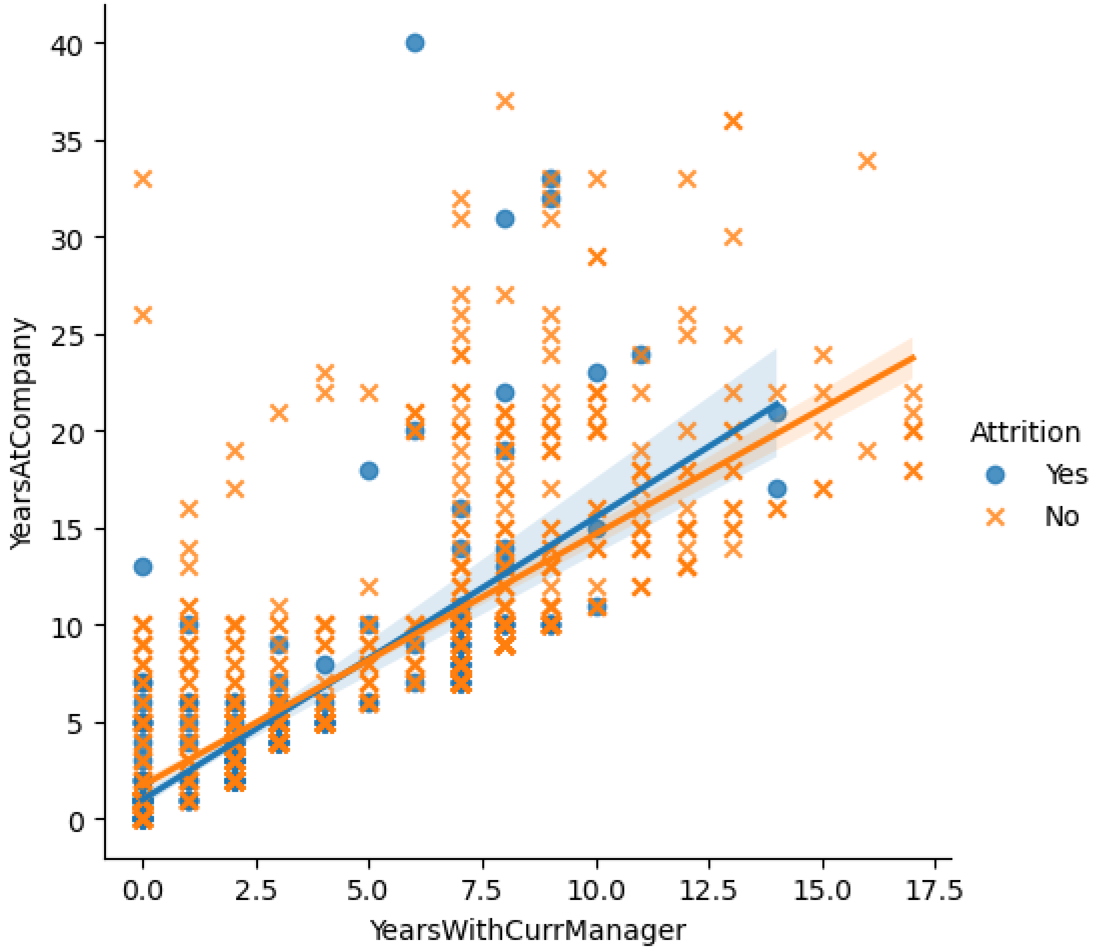

Work Experience in Attrition Work Experience in Un-Attrition

In [153…

```
# plt.figure(figsize=(10,7))
# plt.subplot(1,2,1)
# sns.boxplot(df_['YearsWithCurrManager'],palette=['#FF93AC'])
# plt.title('ManagerInAttrition')
# plt.subplot(1,2,2)
# sns.boxplot(df_Not['YearsWithCurrManager'],palette=['#dfe3ee'])
# plt.title('ManagerInNOTAttrition')
sns.lmplot(x='YearsWithCurrManager',y='YearsAtCompany',data=df,hue='Attrition',markers=[
plt.xlabel('YearsWithCurrManager')
plt.ylabel('YearsAtCompany')
```

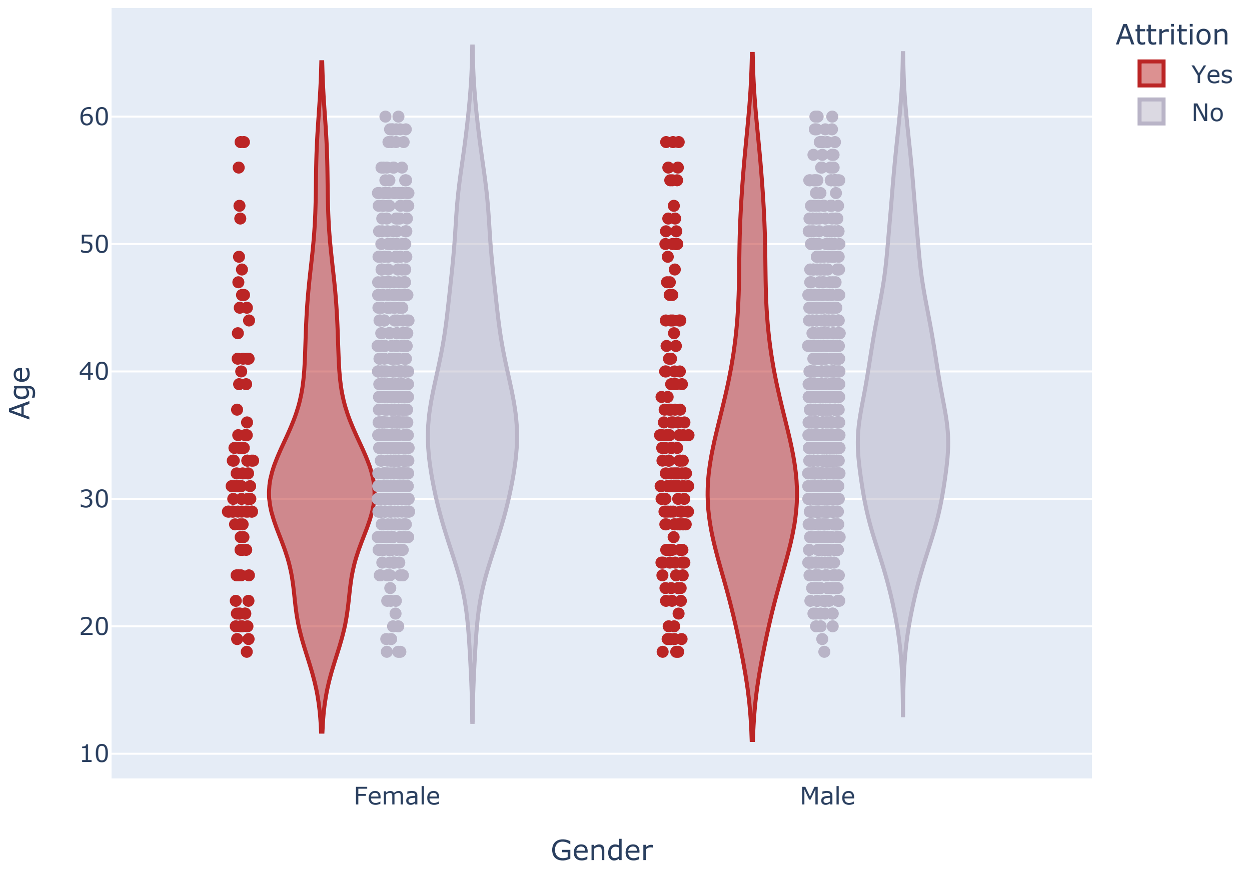
Out[153]:

```
Text(37.94750694444445, 0.5, 'YearsAtCompany')
```



In [156…

```
fig = px.violin(df,x='Gender', y='Age', color='Attrition', points='all', color_discrete_
fig.show()
```



In [162…

```
OldManager=df[df['YearsWithCurrManager']==df['YearsWithCurrManager'].max()]
OldManager
```

Out[162]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationFie
28	44	No	Travel_Rarely	477	Research & Development	7	4	Medi
386	37	No	Travel_Rarely	1107	Research & Development	14	3	Life Scienc
616	51	No	Travel_Rarely	1318	Sales	26	4	Marketi
686	41	No	Travel_Rarely	263	Research & Development	6	3	Medi
875	44	No	Travel_Rarely	200	Research & Development	29	4	Oth
926	43	No	Travel_Rarely	531	Sales	4	4	Marketi
1078	44	No	Travel_Rarely	136	Research & Development	28	3	Life Scienc

7 rows × 35 columns

How many new hires leave in less than a year and why?

In [170…

```
df_leave=df_[df_['YearsAtCompany']<1].count()['YearsAtCompany']
```

```
df_leave
```

Out[170]:

```
16
```

In [171…

```
OneYear=df_[df_['YearsAtCompany']<1]
OneYear
```

Out[171]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationI
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	C
127	19	Yes	Travel_Rarely	528	Sales	22	1	Mark
171	19	Yes	Travel_Frequently	602	Sales	1	1	Tech De
264	28	Yes	Travel_Rarely	529	Research & Development	2	4	Life Scie
296	18	Yes	Travel_Rarely	230	Research & Development	3	3	Life Scie
457	18	Yes	Travel_Frequently	1306	Sales	5	3	Mark
585	23	Yes	Travel_Rarely	1243	Research & Development	6	3	Life Scie
711	29	Yes	Travel_Rarely	906	Research & Development	10	3	Life Scie
801	50	Yes	Travel_Frequently	959	Sales	1	4	C
828	18	Yes	Non-Travel	247	Research & Development	8	1	Me
860	22	Yes	Travel_Frequently	1256	Research & Development	3	4	Life Scie
1060	24	Yes	Travel_Frequently	381	Research & Development	9	3	Me
1068	28	Yes	Travel_Frequently	289	Research & Development	2	2	Me
1153	18	Yes	Travel_Frequently	544	Sales	3	2	Me
1237	32	Yes	Travel_Rarely	964	Sales	1	2	Life Scie
1255	33	Yes	Travel_Rarely	211	Sales	16	3	Life Scie

16 rows × 35 columns

In [174…

```
OneYear.describe(include="object")
```

Out[174]:

	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	MaritalStatus	Over18
count	16	16	16	16	16	16	16	16
unique	1	3	2	5	2	4	2	1
top	Yes	Travel_Rarely	Research & Development	Life Sciences	Male	Laboratory Technician	Single	Y
freq	16	8	9	7	12	7	14	16

In [ ]: