Employees Attrition of a Company

This project is about the employees of a Company who Leave the company, As a **DATA ANALYST** i will find the factors that cause

the attrition of the employees.

In [ ]:

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

df = pd.read_csv('HR_Analytics.csv')
df.head(3)
```

D	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationFie
7	18	18-25	Yes	Travel_Rarely	230	Research & Development	3	3	Life Scienc
2	18	18-25	No	Travel_Rarely	812	Sales	10	3	Medi
8	18	18-25	Yes	Travel_Frequently	1306	Sales	5	3	Marketi

38 columns

In [45]:
```
df.shape
```

Out[45]:
```
(1423, 38)
```

In [44]:
```
df.columns
```

Out[44]:
```
Index(['EmpID', 'Age', 'AgeGroup', 'Attrition', 'BusinessTravel', 'DailyRate',
       'Department', 'DistanceFromHome', 'Education', 'EducationField',
       'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender',
       'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobRole',
       'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'SalarySlab',
       'MonthlyRate', 'NumCompaniesWorked', 'Over18', 'OverTime',
       'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
       'StandardHours', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany',
       'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

--> Check the missing values in the data set

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
EmpID                       0
Age                         0
AgeGroup                    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
SalarySlab                  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       57
dtype: int64
```

In [4]:
```
# Here you can see that "null values " only exist in "YearsWithCurrManager" coluimn
# as we have a large dataset so, we just remove these missing values that cannot effec
```

In [5]:
```
df = df[df['YearsWithCurrManager'].notnull()]
```

In [6]:
```
df.shape
```

Out[6]:
```
(1423, 38)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
EmpID                     0
Age                       0
AgeGroup                  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
SalarySlab                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 [8]:
```
df.head(3)
```

Out[8]:

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	E
0	RM297	18	18-25	Yes	Travel_Rarely	230	Research & Development	3	
1	RM302	18	18-25	No	Travel_Rarely	812	Sales	10	
2	RM458	18	18-25	Yes	Travel_Frequently	1306	Sales	5	

3 rows × 38 columns

In [ ]:
```
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1423 entries, 0 to 1479
Data columns (total 38 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   EmpID                     1423 non-null   object 
 1   Age                       1423 non-null   int64  
 2   AgeGroup                  1423 non-null   object 
 3   Attrition                 1423 non-null   object 
 4   BusinessTravel            1423 non-null   object 
 5   DailyRate                 1423 non-null   int64  
 6   Department                1423 non-null   object 
 7   DistanceFromHome          1423 non-null   int64  
 8   Education                 1423 non-null   int64  
 9   EducationField            1423 non-null   object 
 10  EmployeeCount             1423 non-null   int64  
 11  EmployeeNumber            1423 non-null   int64  
 12  EnvironmentSatisfaction   1423 non-null   int64  
 13  Gender                    1423 non-null   object 
 14  HourlyRate                1423 non-null   int64  
 15  JobInvolvement            1423 non-null   int64  
 16  JobLevel                  1423 non-null   int64  
 17  JobRole                   1423 non-null   object 
 18  JobSatisfaction           1423 non-null   int64  
 19  MaritalStatus             1423 non-null   object 
 20  MonthlyIncome             1423 non-null   int64  
 21  SalarySlab                1423 non-null   object 
 22  MonthlyRate               1423 non-null   int64  
 23  NumCompaniesWorked        1423 non-null   int64  
 24  Over18                    1423 non-null   object 
 25  OverTime                  1423 non-null   object 
 26  PercentSalaryHike         1423 non-null   int64  
 27  PerformanceRating         1423 non-null   int64  
 28  RelationshipSatisfaction  1423 non-null   int64  
 29  StandardHours             1423 non-null   int64  
 30  StockOptionLevel          1423 non-null   int64  
 31  TotalWorkingYears         1423 non-null   int64  
 32  TrainingTimesLastYear     1423 non-null   int64  
 33  WorkLifeBalance           1423 non-null   int64  
 34  YearsAtCompany            1423 non-null   int64  
 35  YearsInCurrentRole        1423 non-null   int64  
 36  YearsSinceLastPromotion   1423 non-null   int64  
 37  YearsWithCurrManager      1423 non-null   float64
dtypes: float64(1), int64(25), object(12)
memory usage: 433.6+ KB
```

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

Out[10]:

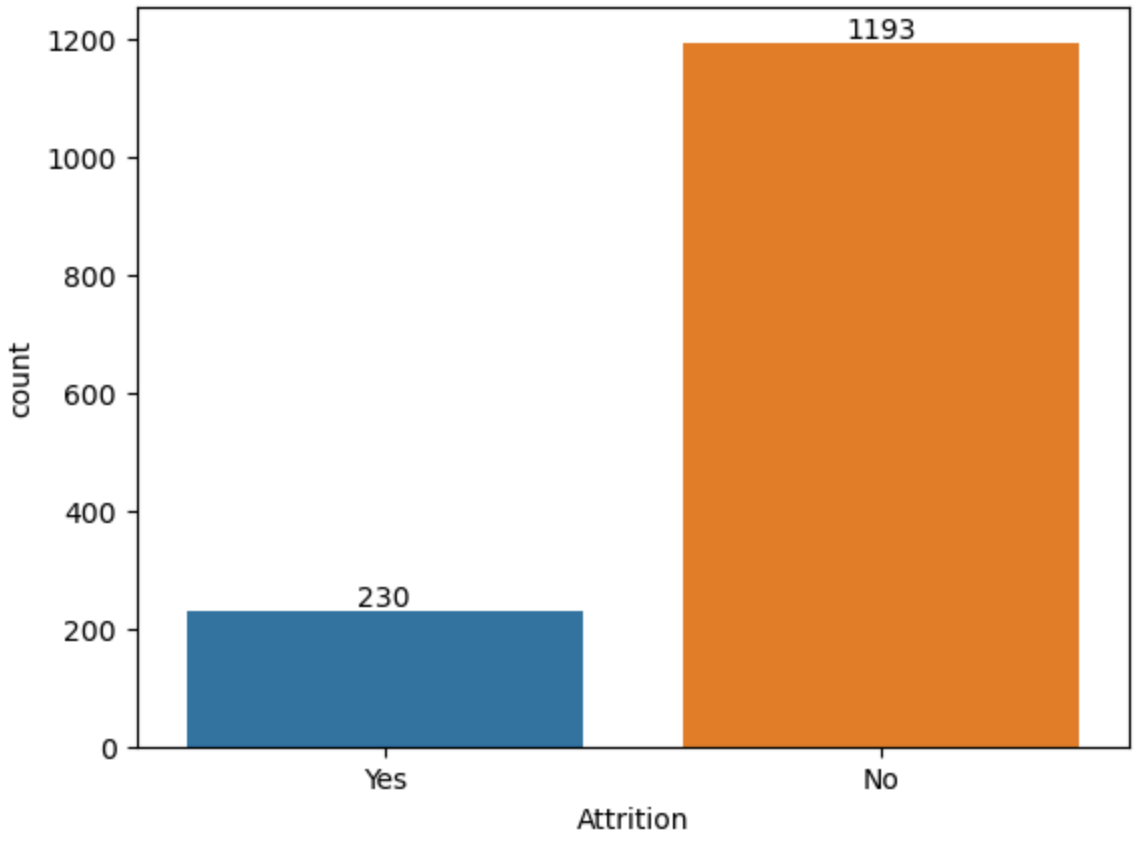
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber
count	1423.000000	1423.000000	1423.000000	1423.000000	1423.0	1423.00000
mean	36.924807	802.000000	9.262825	2.907238	1.0	1063.66409
std	9.133367	404.008071	8.146760	1.023547	0.0	595.37789
min	18.000000	102.000000	1.000000	1.000000	1.0	1.00000
25%	30.000000	465.000000	2.000000	2.000000	1.0	550.50000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1066.00000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1587.50000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.00000

8 rows × 26 columns

Exploratory Data Analysis

FIND ATTRITION STATE OF EMPLOYEES AND COUNT THEM.

In [11]:
```
ax = sns.countplot(x='Attrition',data=df)
ax.bar_label(ax.containers[0])
plt.show()
```

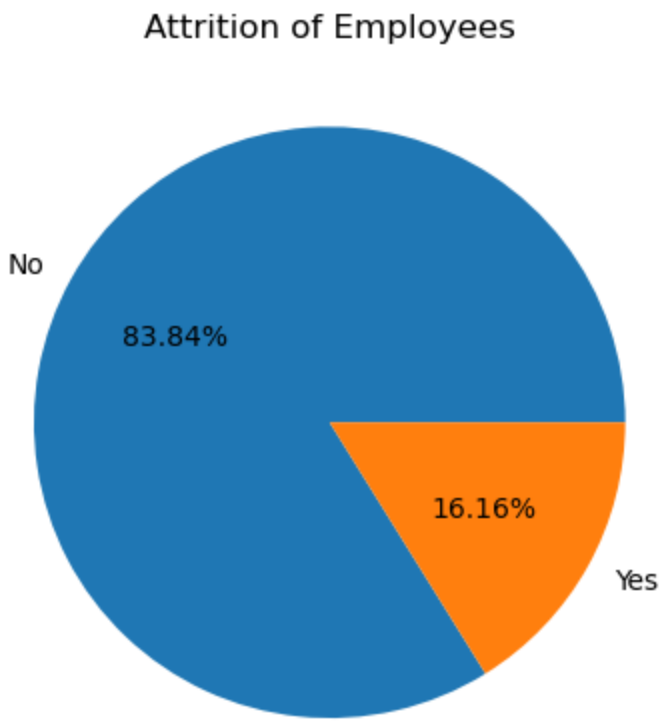


In [12]:
```
# If we want to see this in Count form

grp = df.groupby(df['Attrition']).agg({'Attrition':'count'})
grp
```

Out[12]:

Attrition	Attrition
No	1193
Yes	230

In [13]:
```
# TO CHECK THE SAME THING IN PERCENTAGE FORM I WROTE THIS CODE

plt.pie(grp['Attrition'],labels = grp['Attrition'].index,autopct='%.2f%%')
plt.title('Attrition of Employees')
plt.show()
```



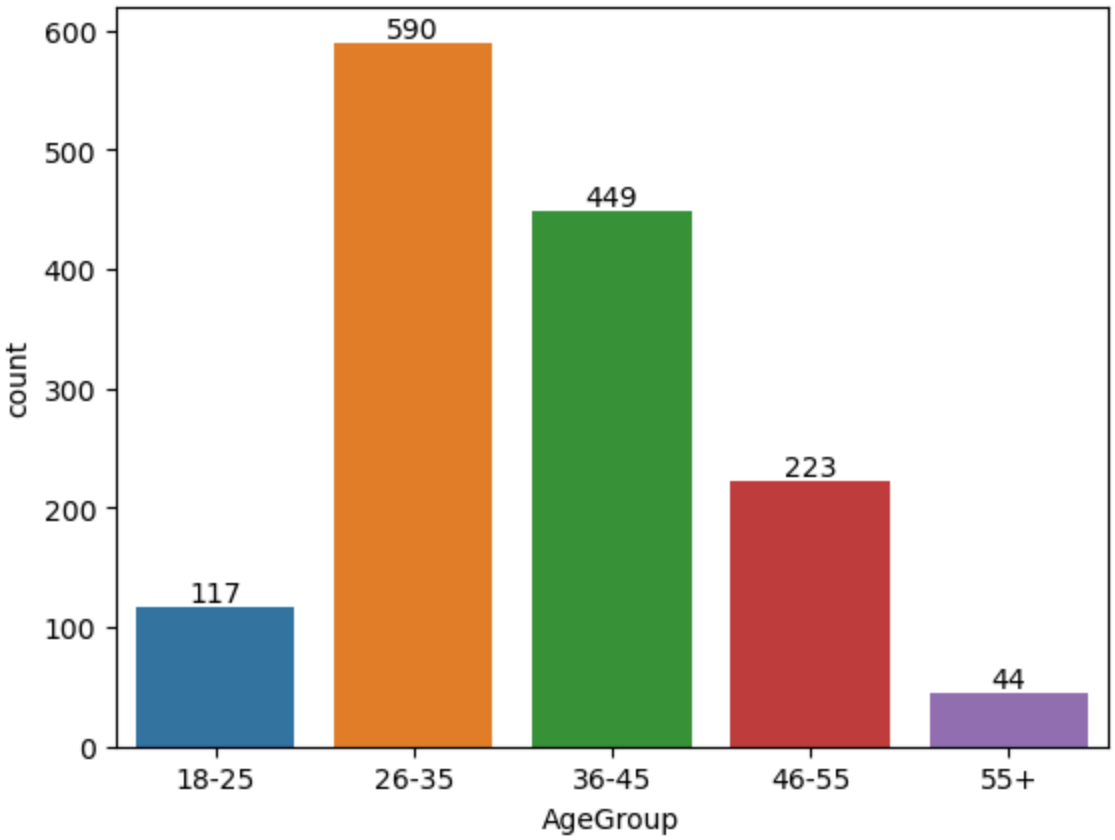
Here you can see that 16.16 Percent Employees had leaved the Company up till now.

In [14]:
```
# NOW CHECKING THE COUNT OF EMPLOYEES IN EACH CATEOGRY OF AGE.
```

In [15]:
```
df['AgeGroup'].value_counts()
```

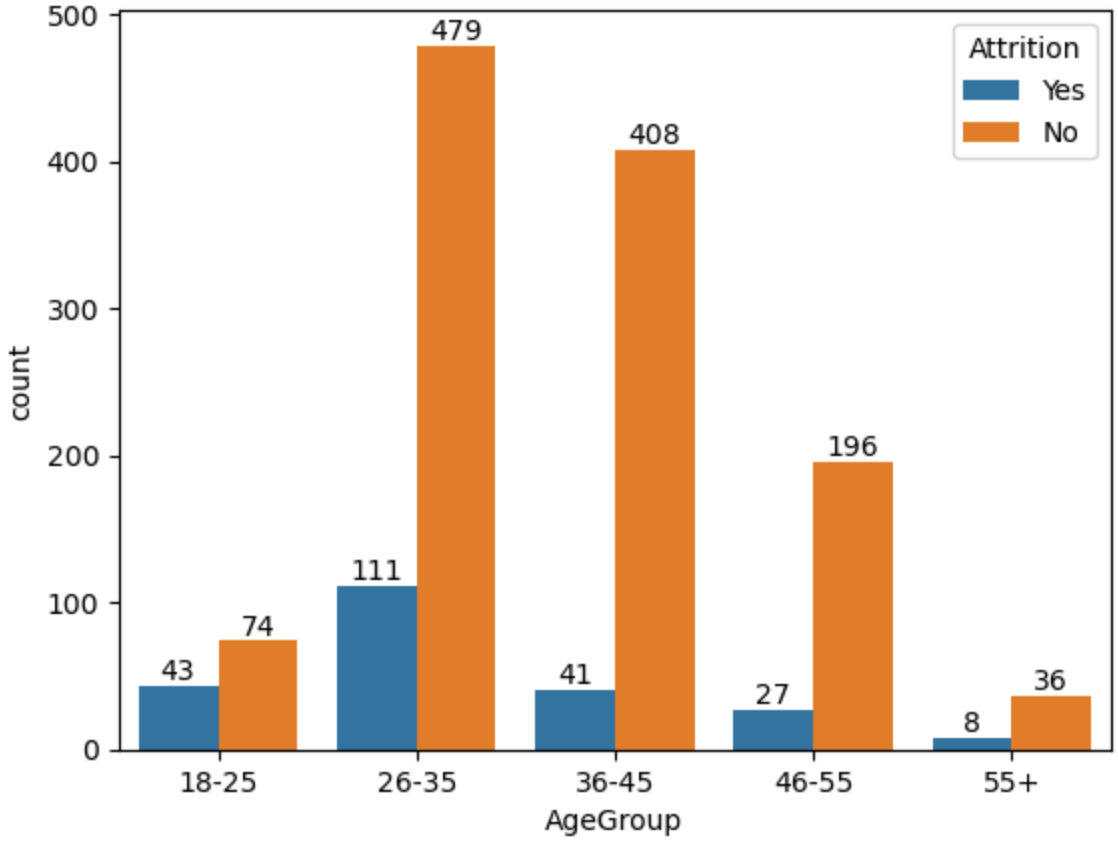
Out[15]:
```
26-35    590
36-45    449
46-55    223
18-25    117
55+       44
Name: AgeGroup, dtype: int64
```

In [16]:
```
# THIS IS THE GRAPHICAL REPRESENTATION OF THE SAME THING

ax = sns.countplot(x='AgeGroup',data=df)
ax.bar_label(ax.containers[0])
plt.show()
```



Finding the Attrition Rate and State With AgeGroup category

In [17]:
```
# NOW HERE I AM  FINDING THE TOTAL ATTRITION IN EACH CATEGORY

ax = sns.countplot(x='AgeGroup',data=df,hue='Attrition')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



In [18]:
```
# Showing this Information in Simple Table Form

group = df.groupby(["AgeGroup",'Attrition']).agg({'Attrition':"count"})
group
```

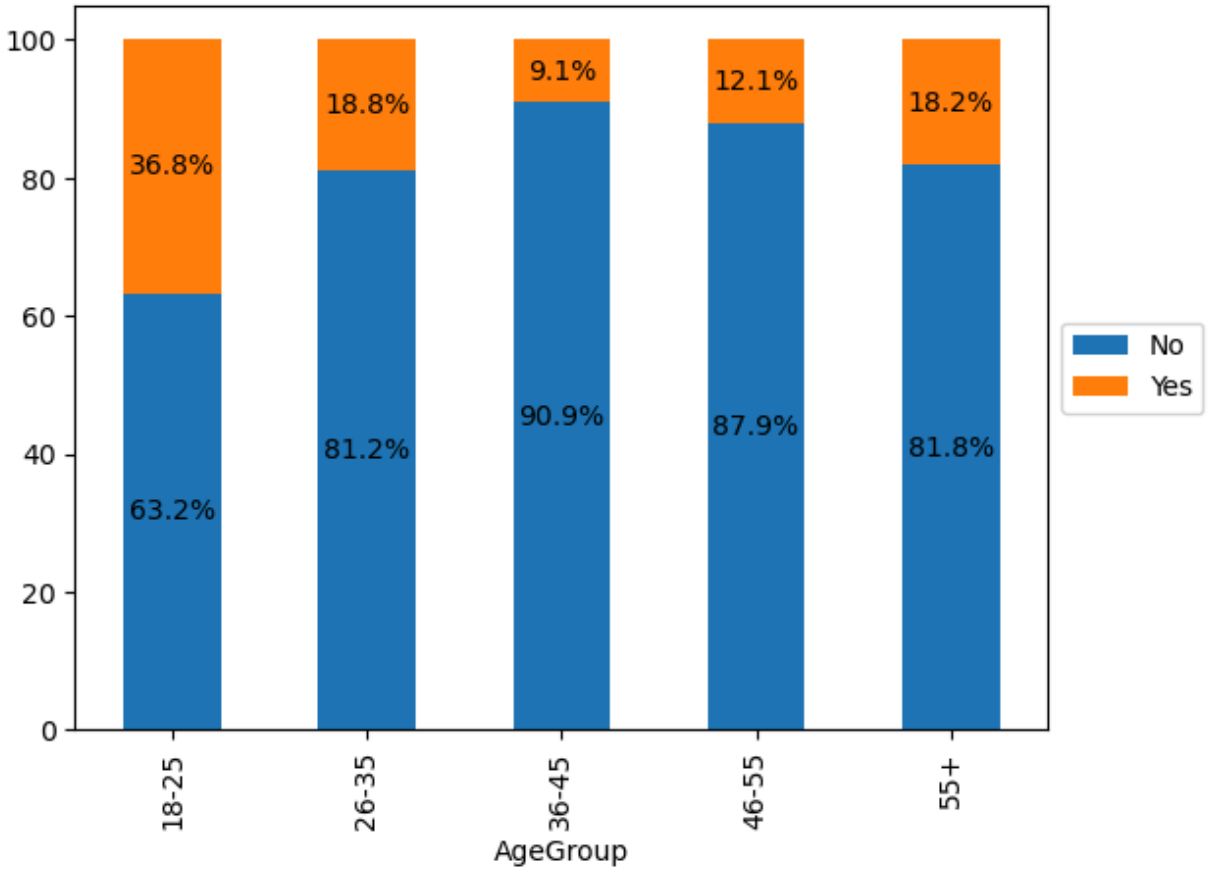
Out[18]:

		Attrition
AgeGroup	**Attrition**	
18-25	**No**	74
	Yes	43
26-35	**No**	479
	Yes	111
36-45	**No**	408
	Yes	41
46-55	**No**	196
	Yes	27
55+	**No**	36
	Yes	8

In [19]:
```
plt.figure(figsize=(9,4))
# Create a pivot table for stacking
stacked_data = df.pivot_table(index='AgeGroup', columns='Attrition', aggfunc='size', f

# Normalize the data to get percentages
```

```
stacked_data_percent = stacked_data.div(stacked_data.sum(axis=1), axis=0) * 100

# Plot the stacked bar plot with percentages
ax = stacked_data_percent.plot(kind='bar', stacked=True)

# this is only for to push the legend on side
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
# Annotate bars with percentages
for c in ax.containers:
    ax.bar_label(c, fmt='%.1f%%', label_type='center')

# Show the plot
plt.show()
```

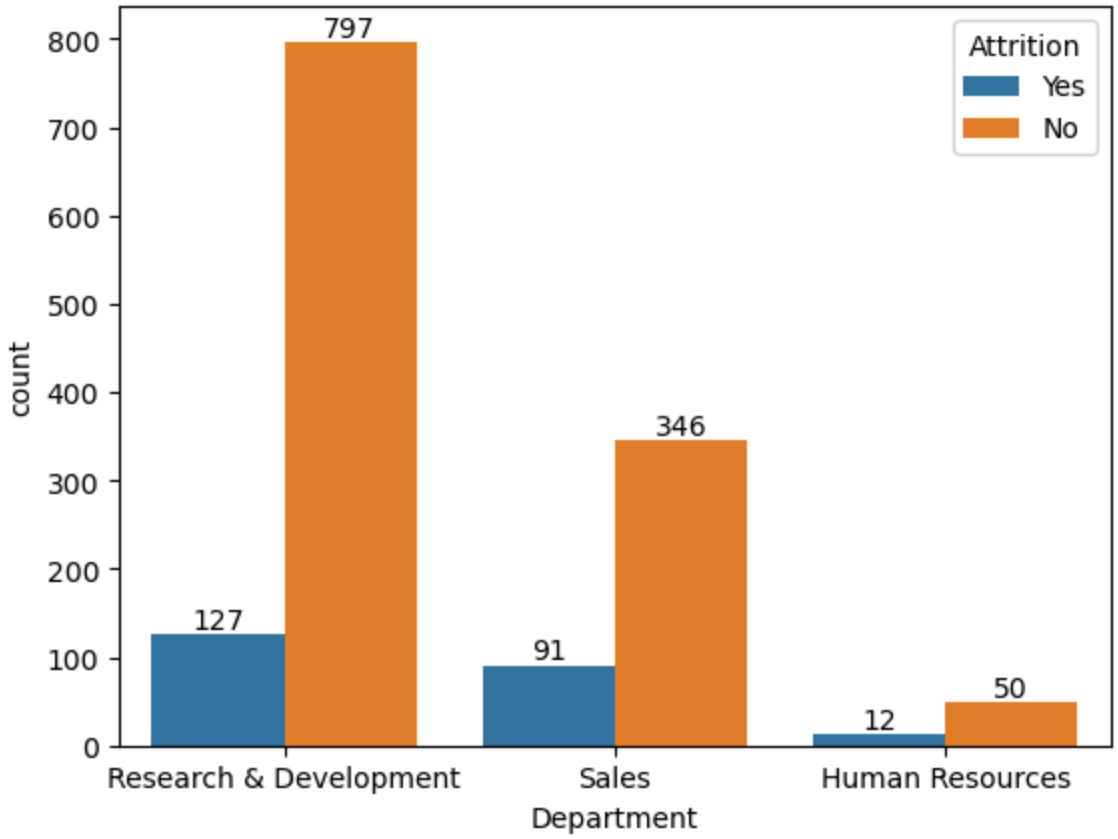
```
<Figure size 900x400 with 0 Axes>
```



This graph clarify that if we check on the basis of AGEGROUP then 36.8% of employees are leaving on the agegroup of 18-25

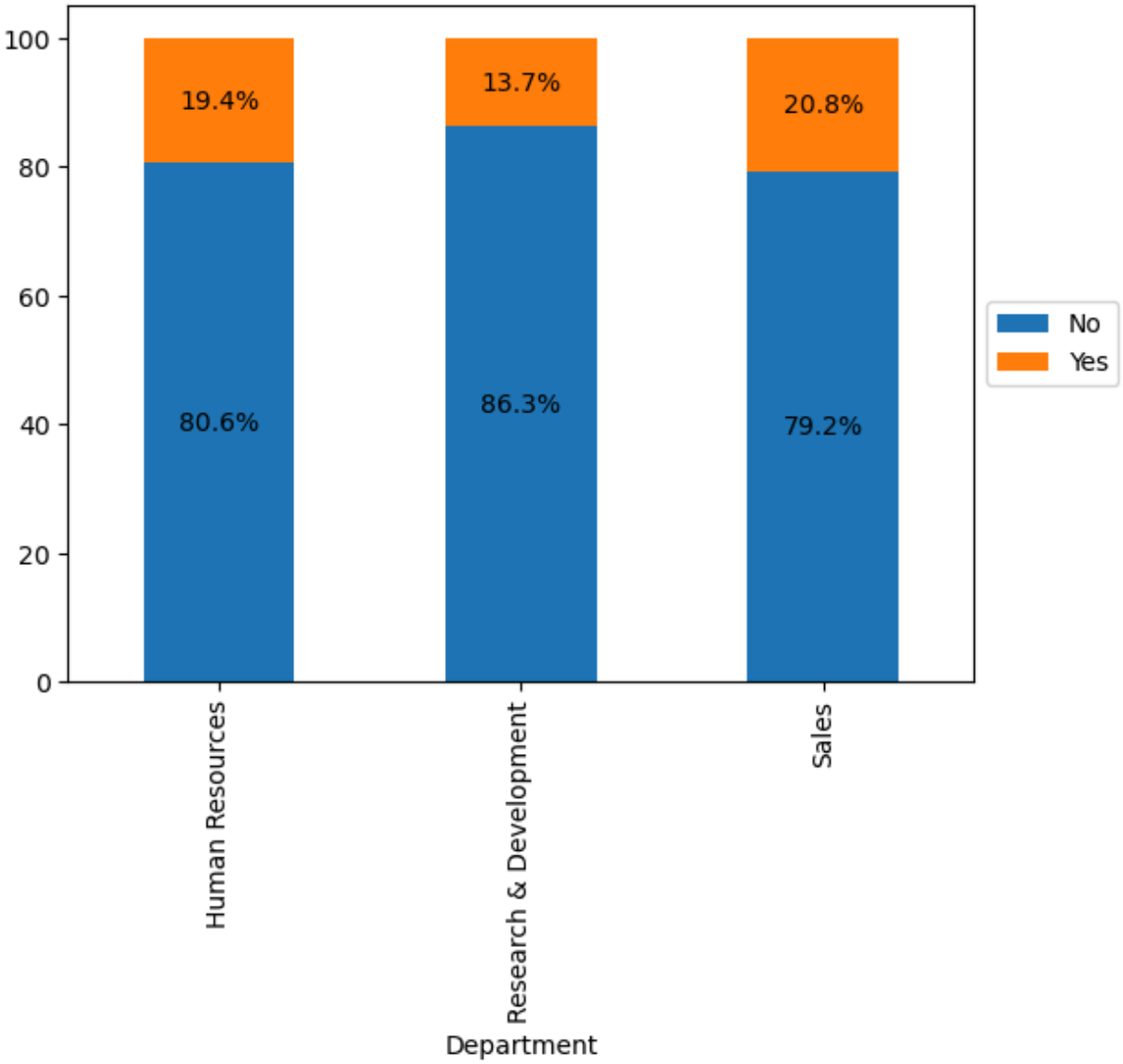
There are different Department in the Company Checking the attrition rate on the basis of Department

In [20]:
```
ax = sns.countplot(x='Department',data=df,hue='Attrition')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



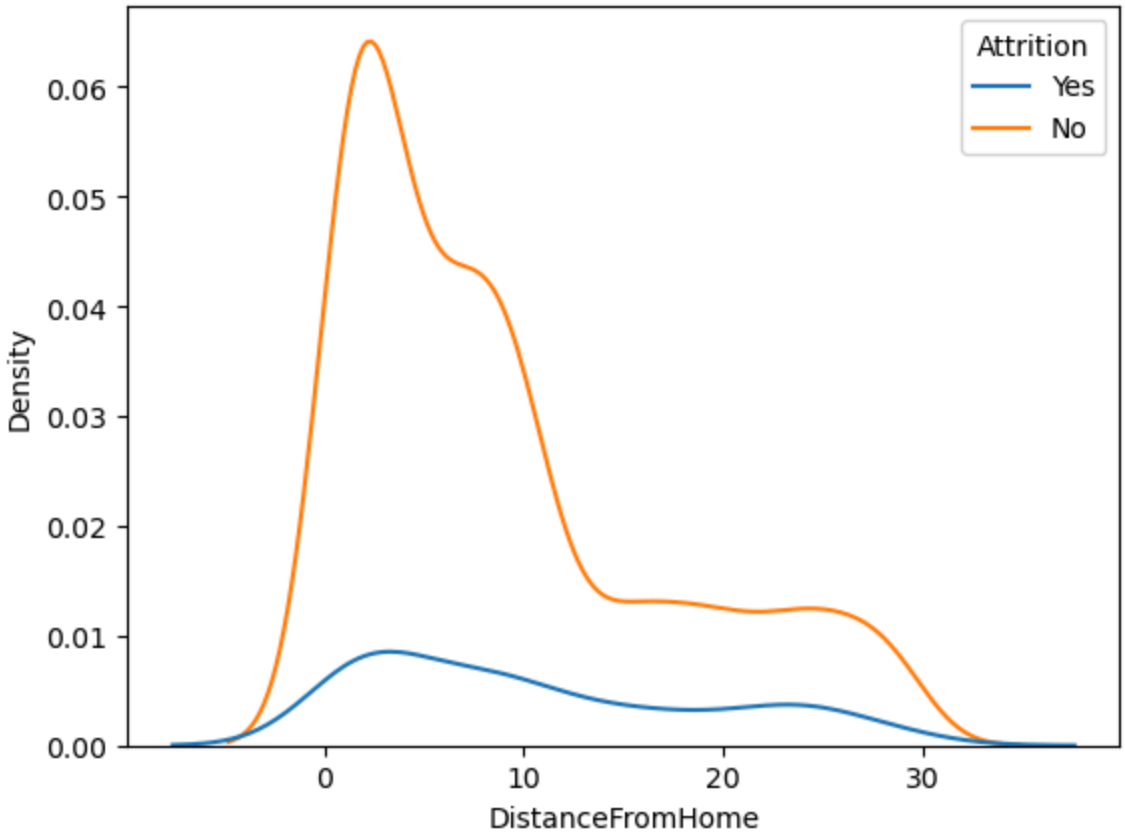
In [21]:
```
plt.figure(figsize=(9,4))
# Create a pivot table for stacking
stacked_data = df.pivot_table(index='Department', columns='Attrition', aggfunc='size',

# Normalize the data to get percentages
stacked_data_percent = stacked_data.div(stacked_data.sum(axis=1), axis=0) * 100

# Plot the stacked bar plot with percentages
ax = stacked_data_percent.plot(kind='bar', stacked=True)

# this is only for to push the legend on side
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
# Annotate bars with percentages
for c in ax.containers:
    ax.bar_label(c, fmt='%.1f%%', label_type='center')

# Show the plot
plt.show()
```

```
<Figure size 900x400 with 0 Axes>
```



In [22]:
```
sns.kdeplot(x='DistanceFromHome',data=df,hue='Attrition')
```

Out[22]:
```
<AxesSubplot:xlabel='DistanceFromHome', ylabel='Density'>
```



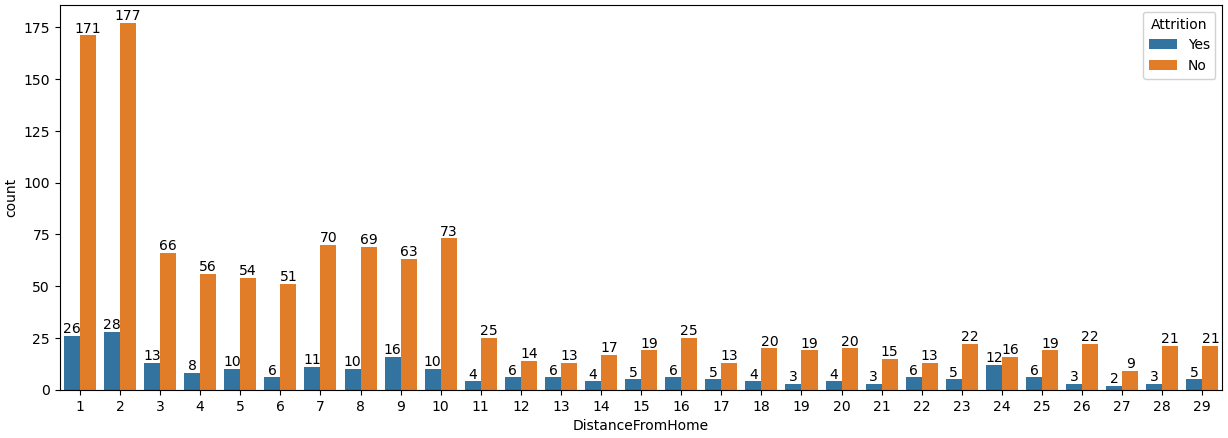
Counting the number of Employees on the basis of Attrition over DistanceFromHome

In [23]:
```
stacked_data = df.pivot_table(index='DistanceFromHome', columns='Attrition', aggfunc='
stacked_data
```

Out[23]:

Attrition	No	Yes
DistanceFromHome		
1	171	26
2	177	28
3	66	13
4	56	8
5	54	10
6	51	6
7	70	11
8	69	10
9	63	16
10	73	10
11	25	4
12	14	6
13	13	6
14	17	4
15	19	5
16	25	6
17	13	5
18	20	4
19	19	3
20	20	4
21	15	3
22	13	6
23	22	5
24	16	12
25	19	6
26	22	3
27	9	2
28	21	3
29	21	5

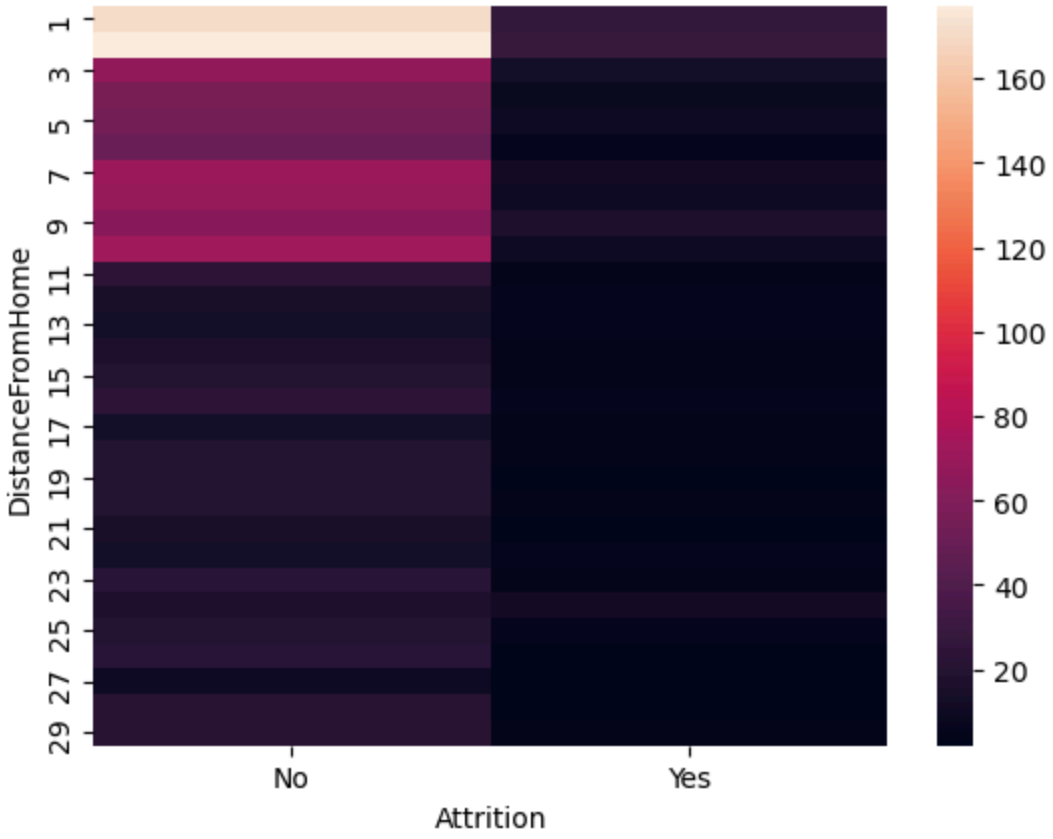
In [24]:
```
plt.figure(figsize=(15,5))
ax = sns.countplot(x='DistanceFromHome',data=df,hue='Attrition')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



If you carefully see this graph then you can see that, Whose People they came from far areas they leave the company

In [25]:
```
sns.heatmap(stacked_data)
```

Out[25]:
```
<AxesSubplot:xlabel='Attrition', ylabel='DistanceFromHome'>
```



In [26]:
```
# On the basis of Size finding a tabel of Department over DistanceFromHome

pd.pivot_table(data=df,index='Department',columns ='DistanceFromHome',aggfunc='size')
```

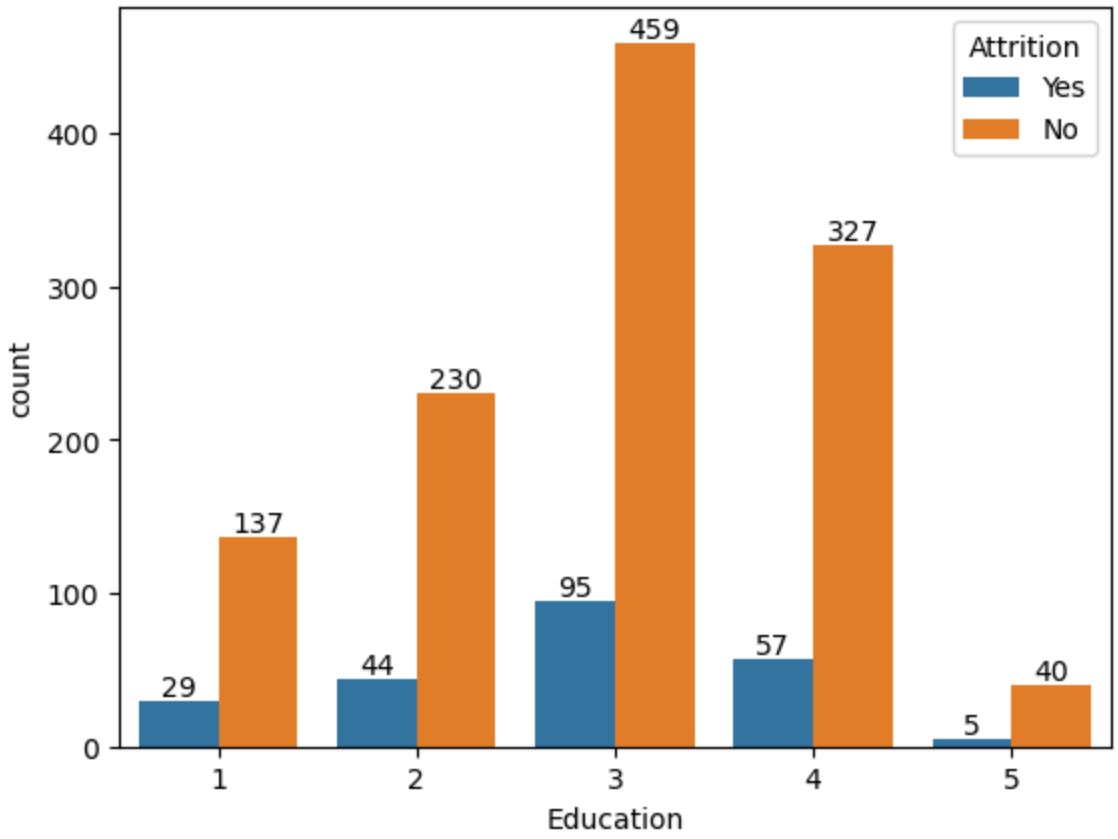
Out[26]:

DistanceFromHome	1	2	3	4	5	6	7	8	9	10	...	20	21	22	
Department															
Human Resources	10.0	11.0	4.0	2.0	2.0	4.0	NaN	7.0	1.0	4.0	...	1.0	NaN	2.0	1
Research & Development	140.0	127.0	48.0	41.0	40.0	42.0	56.0	47.0	54.0	46.0	...	15.0	10.0	12.0	2
Sales	47.0	67.0	27.0	21.0	22.0	11.0	25.0	25.0	24.0	33.0	...	8.0	8.0	5.0	1

3 rows × 29 columns

On the basis of Education find the Count the employees Attrtion

In [27]:
```
ax = sns.countplot(x='Education',data=df,hue='Attrition')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```

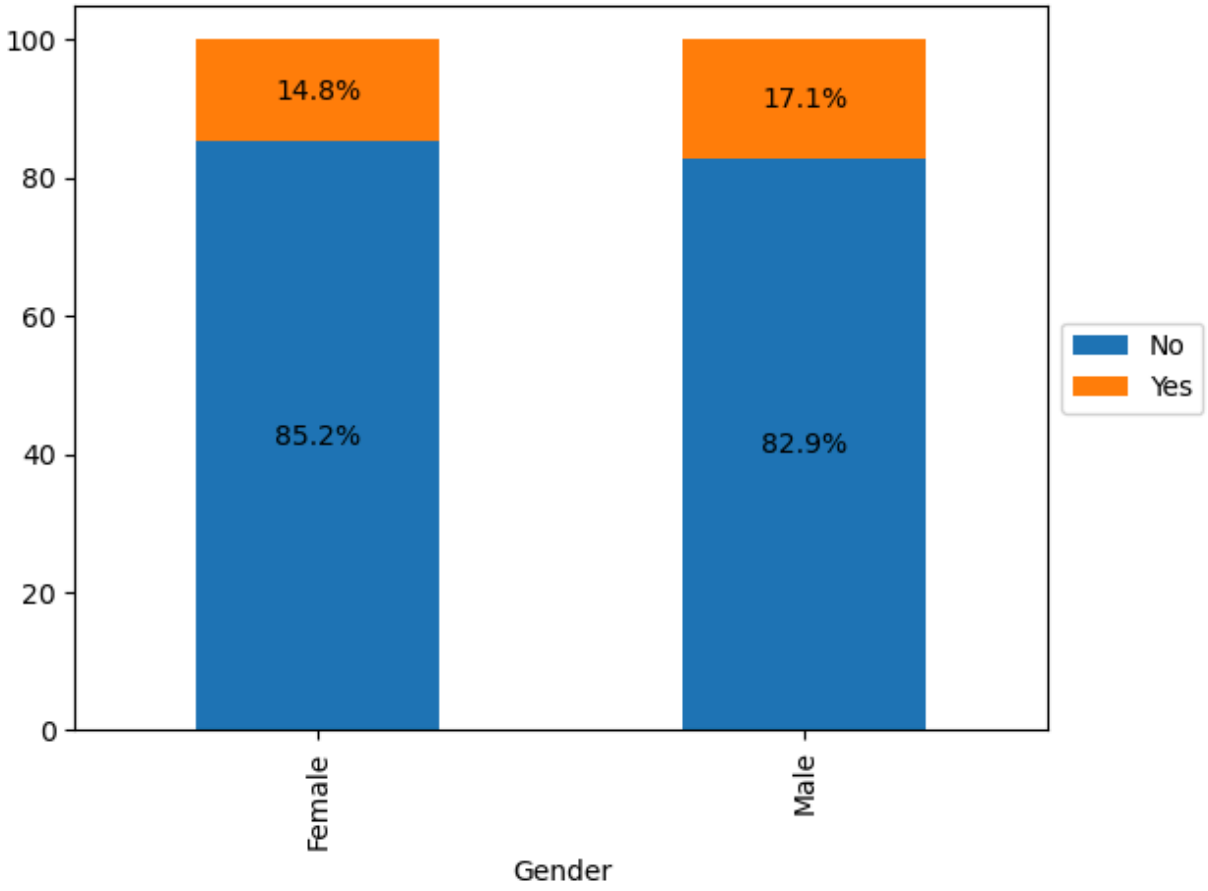


This show that Alot of Employees on your company are at "Undergraduate" Degree

In [28]:
```
plt.figure(figsize=(9,4))
# Create a pivot table for stacking
stacked_data = df.pivot_table(index='Gender', columns='Attrition', aggfunc='size', fil

# Normalize the data to get percentages
```

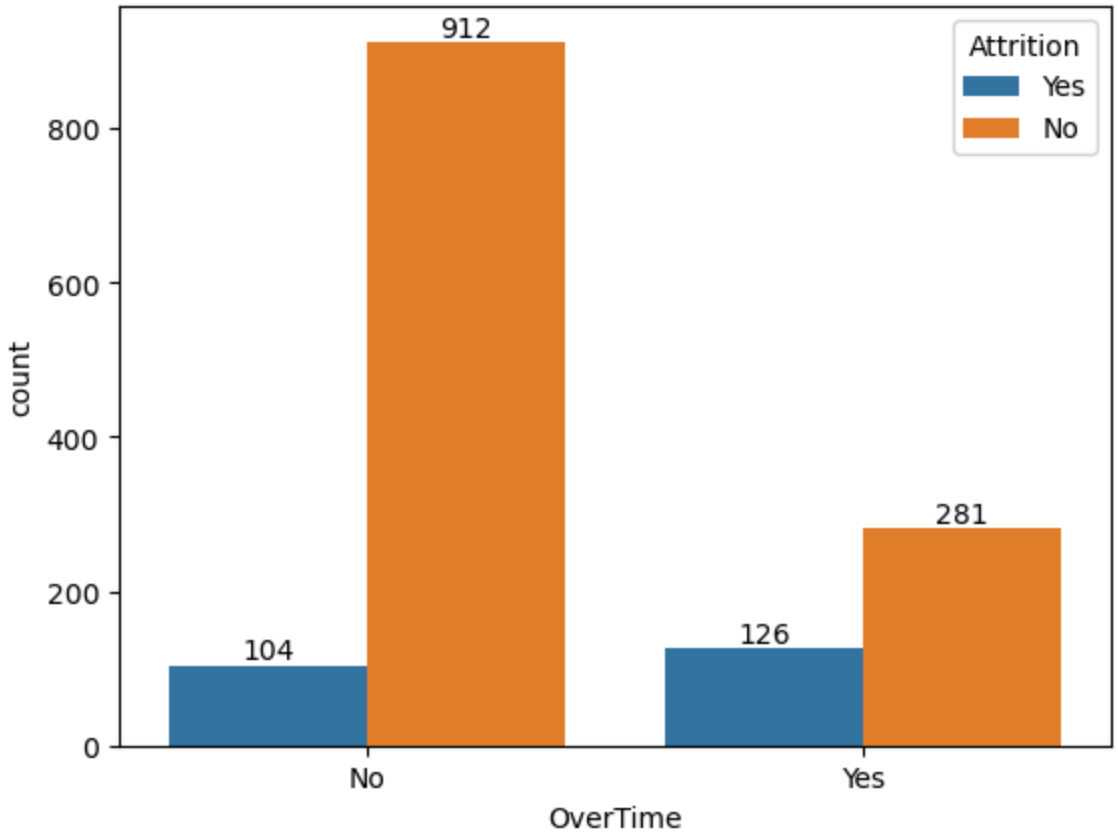
```
stacked_data_percent = stacked_data.div(stacked_data.sum(axis=1), axis=0) * 100

# Plot the stacked bar plot with percentages
ax = stacked_data_percent.plot(kind='bar', stacked=True)

# this is only for to push the legend on side
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
# Annotate bars with percentages
for c in ax.containers:
    ax.bar_label(c, fmt='%.1f%%', label_type='center')

# Show the plot
plt.show()
```

```
<Figure size 900x400 with 0 Axes>
```



CHECK THE ATTRITION OF EMPLOYEES ON BASIS OF OVERTIME THEY DO OR NOT

In [29]:
```
ax = sns.countplot(x='OverTime',data=df,hue='Attrition')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



This show that those people they are doing overtime are leaving the Companay more as compared to those are not did overtime

In [ ]:

In [30]:
```
df.pivot_table(index='Attrition',columns='RelationshipSatisfaction',values='DailyRate'
```

Out[30]:

RelationshipSatisfaction	1	2	3	4
Attrition				
No	791.412037	820.404000	809.226913	821.235632
Yes	806.490909	754.545455	691.602941	763.920635

In [31]:
```
df.columns.values
```

Out[31]:
```
array(['EmpID', 'Age', 'AgeGroup', 'Attrition', 'BusinessTravel',
       'DailyRate', 'Department', 'DistanceFromHome', 'Education',
       'EducationField', 'EmployeeCount', 'EmployeeNumber',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'SalarySlab', 'MonthlyRate',
       'NumCompaniesWorked', 'Over18', 'OverTime', 'PercentSalaryHike',
       'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager'], dtype=object)
```

In [32]:
```
gender = df.pivot_table(index='Attrition',columns='Gender',values='DailyRate',aggfunc=
```

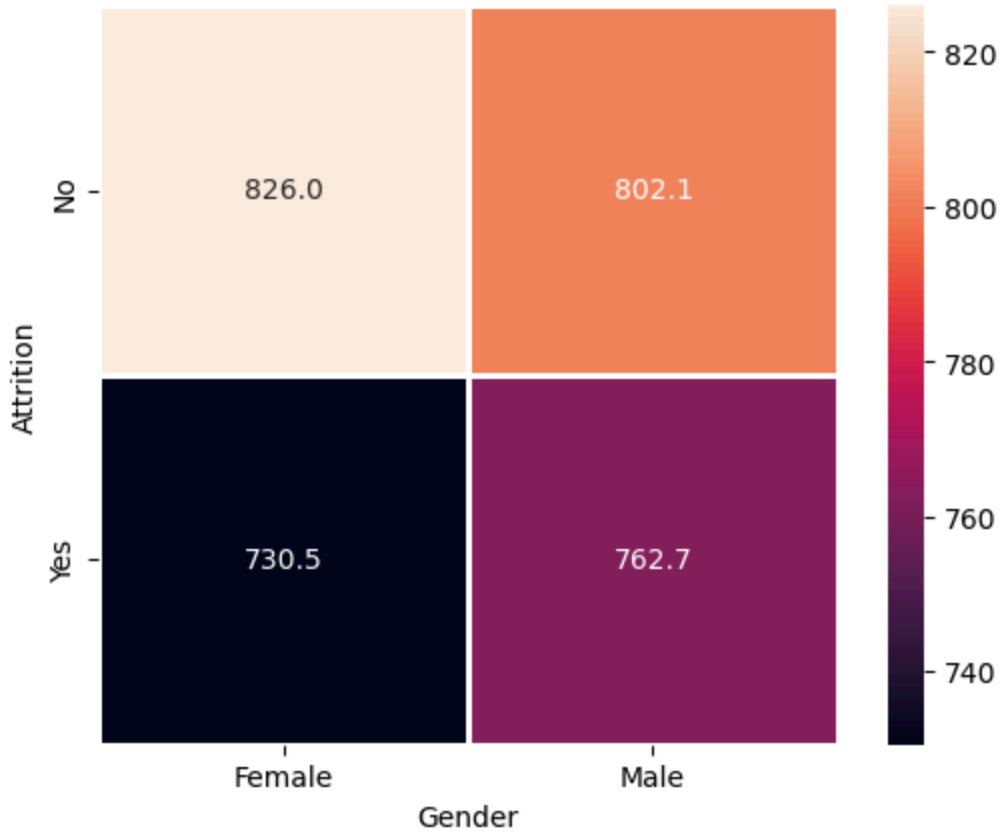
In [33]:
```
gender
```

Out[33]:

Gender	Female	Male
Attrition		
No	826.035052	802.127119
Yes	730.476190	762.691781

In [34]:
```
sns.heatmap(gender,annot=True,fmt='1.1f',linewidths = 1, square= True)
```

Out[34]:
```
<AxesSubplot:xlabel='Gender', ylabel='Attrition'>
```



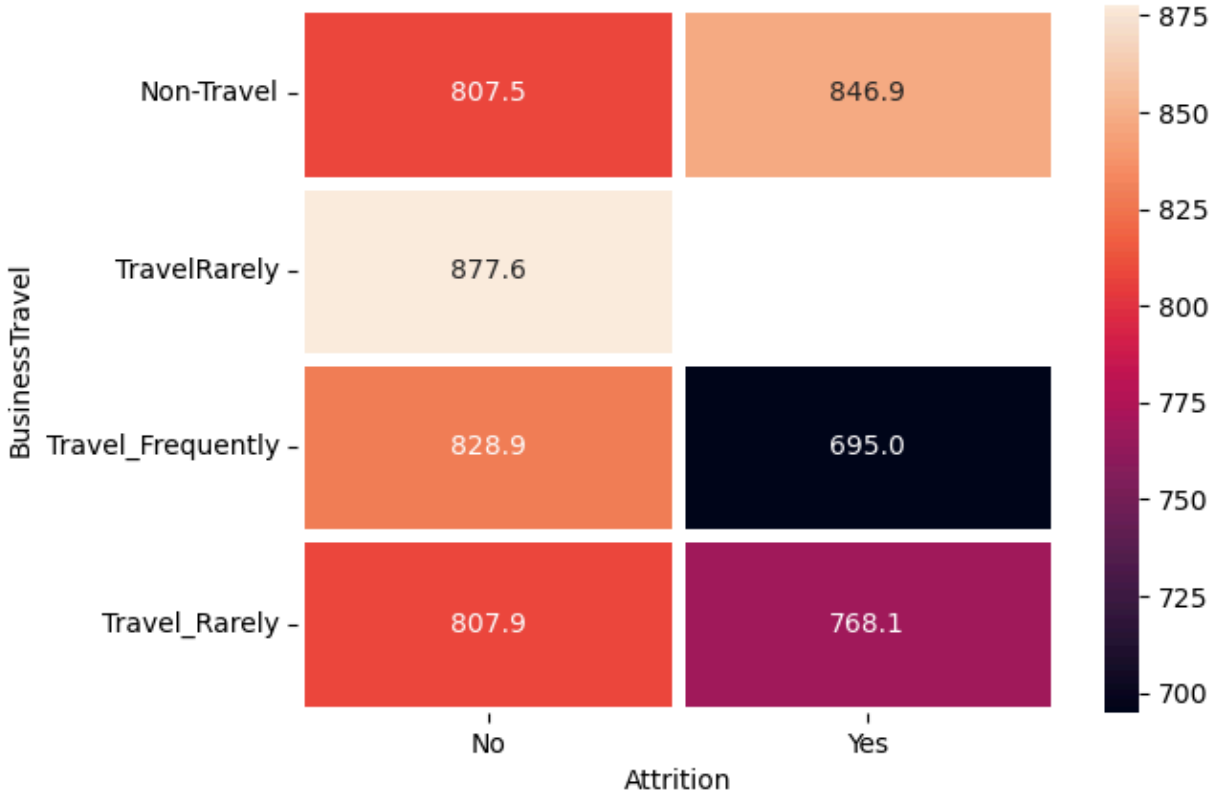
In [35]:
```
g = df.pivot_table(index='BusinessTravel',columns='Attrition',values='DailyRate').rese
g
```

Out[35]:

Attrition	BusinessTravel	No	Yes
0	Non-Travel	807.463235	846.916667
1	TravelRarely	877.571429	NaN
2	Travel_Frequently	828.871921	695.000000
3	Travel_Rarely	807.926800	768.112583

In [36]:
```
g = df.pivot_table(index='BusinessTravel',columns='Attrition',values='DailyRate')
sns.heatmap(g,annot=True,linewidth=5,fmt='.1f')
```

Out[36]:
```
<AxesSubplot:xlabel='Attrition', ylabel='BusinessTravel'>
```



In [ ]:

In [37]:
```
a =pd.pivot_table(index='Department',columns = 'AgeGroup',values='DailyRate',aggfunc={
```

In [38]:
```
a
```

Out[38]:

Department	18-25	26-35	36-45	46-55	55+
Human Resources	483.750000	717.363636	803.600000	732.000000	814.666667
Research & Development	781.837838	789.312169	811.614618	864.873239	757.034483
Sales	782.974359	823.315789	791.357724	755.315068	868.333333

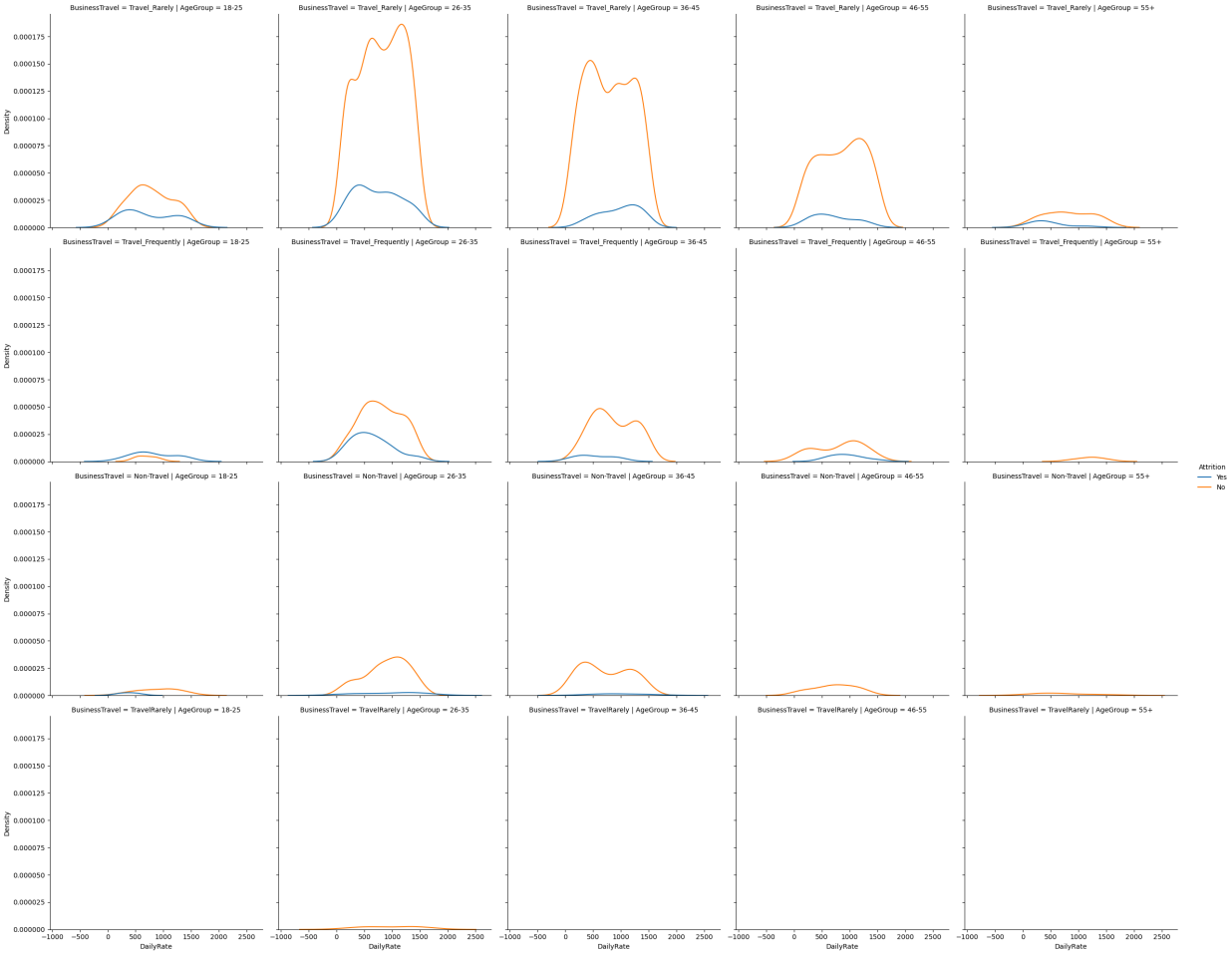
In [39]:
```
sns.heatmap(a,linewidth=5,fmt='.1f',annot=True)
```

Out[39]:
```
<AxesSubplot:xlabel='AgeGroup', ylabel='Department'>
```

In [40]:
```
# IMPRTANT INSIGHT

# the above two graph show that more thatn 50% of employee leave the company because t
# been given very low Daily rate


# WHAT ACTION NEED TO TAKE:

# need to improve DAILY RATE that can stop the attrition of empoyees they are in
# AGEGROUP OF 18-25
```

In [41]:
```
plt.figure(figsize=(14,14))
sns.displot(data = df ,x = 'DailyRate',hue = 'Attrition',kind='kde',col='AgeGroup',row
plt.show()
```

```
C:\Users\PMYLS\anaconda3\lib\site-packages\seaborn\distributions.py:316: UserWarning:
Dataset has 0 variance; skipping density estimate. Pass `warn_singular=False` to disa
ble this warning.
  warnings.warn(msg, UserWarning)
C:\Users\PMYLS\anaconda3\lib\site-packages\seaborn\distributions.py:316: UserWarning:
Dataset has 0 variance; skipping density estimate. Pass `warn_singular=False` to disa
ble this warning.
  warnings.warn(msg, UserWarning)
C:\Users\PMYLS\anaconda3\lib\site-packages\seaborn\distributions.py:316: UserWarning:
Dataset has 0 variance; skipping density estimate. Pass `warn_singular=False` to disa
ble this warning.
  warnings.warn(msg, UserWarning)
<Figure size 1400x1400 with 0 Axes>
```

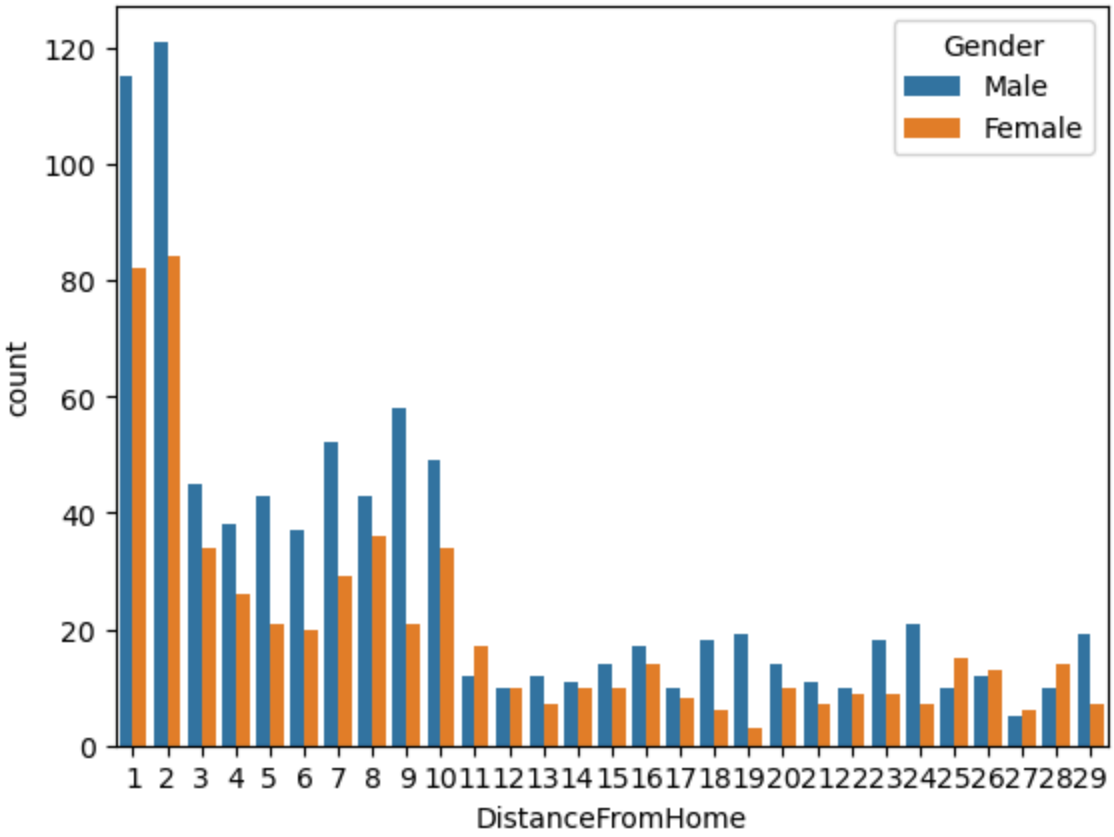


In [42]:
```
# this graph show information that those people who "RarelyTravel" and in ageGroup of
# 18-25 they are alot in number in case of Attrition
```

In [ ]:
```

```

In [43]:
```
sns.countplot(x='DistanceFromHome',hue='Gender',data=df)
```

Out[43]:
```
<AxesSubplot:xlabel='DistanceFromHome', ylabel='count'>
```



In [ ]:

In [ ]:

In [ ]:

Main Finding of the Whole project

EMPLOYEEES ATTION OF A COMPANY REPORT

Find the Attrition State of Employees.

==> From Total of 1423 Employees 230 leave the Company and 1193 employees are remaining.

==> If we see this in Percentages form, then 16.16 percent had leave and 83.84 are remaining.

Attrition rate of Employees on basis of AGEGROUP.

==> This graph clarifies that if we check on the basis of AGEGROUP then 36.8% of employees are leaving on the AGEGROUP of 18-25. To solve this problem provide them good offer in this AGEGROUP so the they can't leave

There are different Department in the Company Check the attrition rate on the basis of Department

==> 20% and 19% Employees Attrition rate in Sales department is 20% and in Humain & Resources Department 19% as compare to Research and Development in which there is only 13% Attrition rate. Try to stop these department employees from attrition.

Count the number of Employees on the basis of Attrition over DistanceFromHome.

==> This represent that employees are less in number those who come from far area and their rate of leaving is also high.

Check on the basis of Education in my Company alot of employees belong to what?

==>This show that Alot of Employees on your company are at "Undergraduate" Degree.

Find and count the Attrition of Employees on basis of Gender.

==> As compared to female male are leaving the Company more.

Is overtime play a crucial role

==> This show that those people they are doing overtime are leaving the Company more as compared to those are not do overtime. this information tell us that those people who "RarelyTravel" and in ageGroup of 18-25 they are alot in number in case o Attrition.

In [ ]: