

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pylab
%matplotlib inline
```

```
df=pd.read_csv('C:\\Users\\hp\\Downloads\\WA_Fn-UseC_-HR-Employee-Attrition.csv')
```

```
df.head(2)
```

	Age	Attrition	BusinessTravel	DailyRate	Department
0	41	Yes	Travel_Rarely	1102	Sales
1	49	No	Travel_Frequently	279	Research & Development

	DistanceFromHome	Education	EducationField	EmployeeCount
EmployeeNumber \				
0	1	2	Life Sciences	1
1				
1	8	1	Life Sciences	1
2				

	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	\
0	...		1	80	0
1	...		4	80	1

	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance
YearsAtCompany \			
0	8		0
6			1
1	10		3
10			

	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
0	4		0
1	7		1
			5
			7

```
[2 rows x 35 columns]
```

```
df.shape
```

```
(1470, 35)
```

```
df.isnull().sum()
```

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

There is no null value in dataset

```
df.columns
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. Total Employee

```

total_employee=df.shape[0]
print(f"total_employees : {total_employee}")

total_employees : 1470

```

## 2. Total Attrition

```

df['Attrition'].value_counts()

Attrition
No      1233
Yes      237
Name: count, dtype: int64

df['Attrition'].value_counts(normalize=True)*100

Attrition
No      83.877551
Yes     16.122449
Name: proportion, dtype: float64

```

The attrition rate for the company is 16.12%.

```

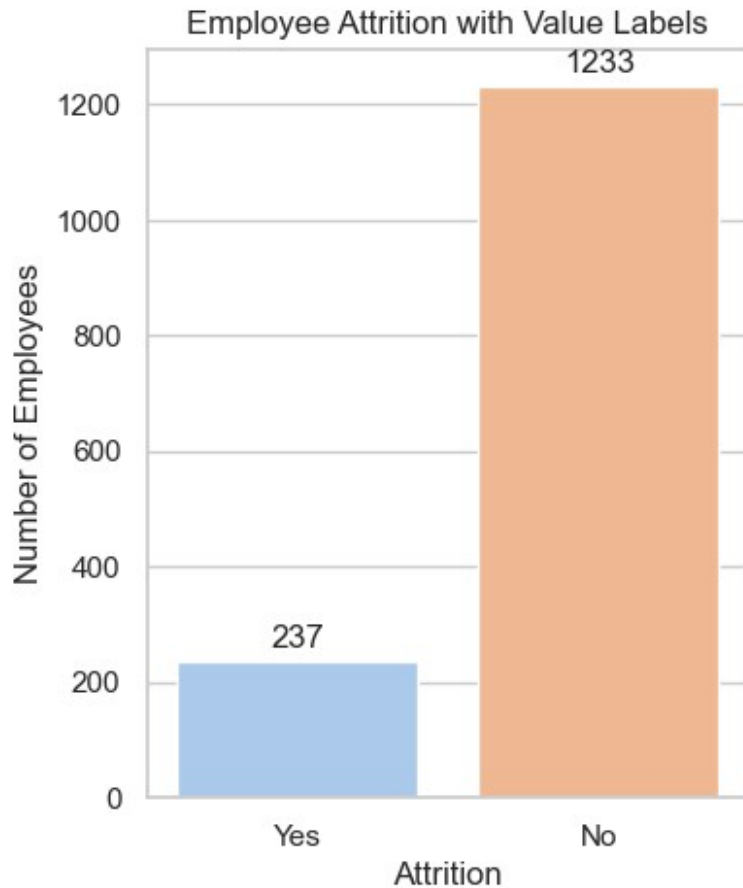
# Set up the visualization style
sns.set(style="whitegrid")

# Create the bar chart
plt.figure(figsize=(4, 5))
ax = sns.countplot(x='Attrition', data=df, palette='pastel')

# Loop through each container and add the value labels
for container in ax.containers:
    ax.bar_label(container, label_type='edge', fmt='%d', padding=3)

# Set chart title and labels
plt.title('Employee Attrition with Value Labels')
plt.xlabel('Attrition')
plt.ylabel('Number of Employees')
plt.show()

```



```
total_attrition=df[df['Attrition']=='Yes'].shape[0]  
print(f"Total Attrition : {total_attrition}")
```

Total Attrition : 237

### 3. Attrition Rate

```
attrition_rate=(total_attrition/total_employee)*100  
print(f"Attrition Rate : {attrition_rate:.2f}%")
```

Attrition Rate : 16.12%

### 4. Avg Age

```
avg_age=df['Age'].mean()  
print(f"Average Age : {avg_age:.2f} years")
```

Average Age : 36.92 years

```
avg_age_attrition_yes=df[df['Attrition']=='Yes']  
['Age'].mean().round(2)  
print(f"Average age where Attrition is 'Yes' :  
{avg_age_attrition_yes}")
```

```
Average age where Attrition is 'Yes' : 33.61
```

The average age of employees is 37, and the average age of attrition employees is 34.

## 5. Avg Salary

```
avg_salary = df['MonthlyIncome'].mean()  
print(f"Average Salary: {avg_salary:2f}")
```

```
Average Salary: 6502.931293
```

The average salary for employees is 6502.931293.

```
# Calculate the average salary grouped by attrition  
avg_salary_by_attrition=df.groupby('Attrition')  
['MonthlyIncome'].mean()  
print("Average Salary by Attrition:")  
print(avg_salary_by_attrition.round(2))
```

```
Average Salary by Attrition:
```

```
Attrition
```

```
No      6832.74
```

```
Yes      4787.09
```

```
Name: MonthlyIncome, dtype: float64
```

## 6. Avg Working Years

```
avg_working_years = df['YearsAtCompany'].mean()  
print(f"Average Working Years : {avg_working_years}")
```

```
Average Working Years : 7.0081632653061225
```

Average years of employee to leave the company is 7.0081632653061225

```
avg_working_year_attrition=df.groupby('Attrition')  
['YearsAtCompany'].mean()  
print("Average working years by Attrition:")  
print(avg_working_year_attrition.round(2))
```

```
Average working years by Attrition:
```

```
Attrition
```

```
No      7.37
```

```
Yes      5.13
```

```
Name: YearsAtCompany, dtype: float64
```

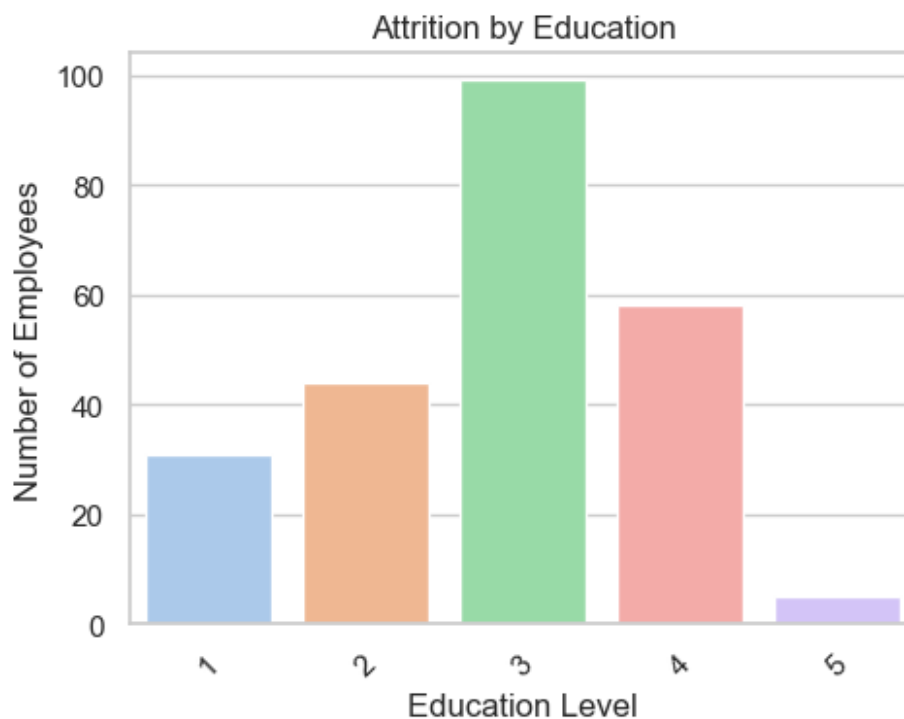
## Plotting Functions

```
def bar_chart_small(data, title, xlabel, ylabel,  
color_palette="pastel", figsize=(5, 4), rotation=45):
```

```
plt.figure(figsize=figsize) # Set figure size
sns.barplot(x=data.index, y=data.values, palette=color_palette)
plt.title(title)
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.xticks(rotation=rotation) # Rotate x-axis labels
plt.tight_layout() # Adjust layout
plt.show()
```

## 7. Attrition by Education

```
attrition_by_education = df[df['Attrition'] == 'Yes']
['Education'].value_counts()
bar_chart_small(attrition_by_education, "Attrition by Education",
"Education Level", "Number of Employees")
```



## 8. Attrition by EducationField

```
df['EducationField'].value_counts()
```

EducationField	
Life Sciences	606
Medical	464
Marketing	159
Technical Degree	132
Other	82

```
Human Resources      27
Name: count, dtype: int64
```

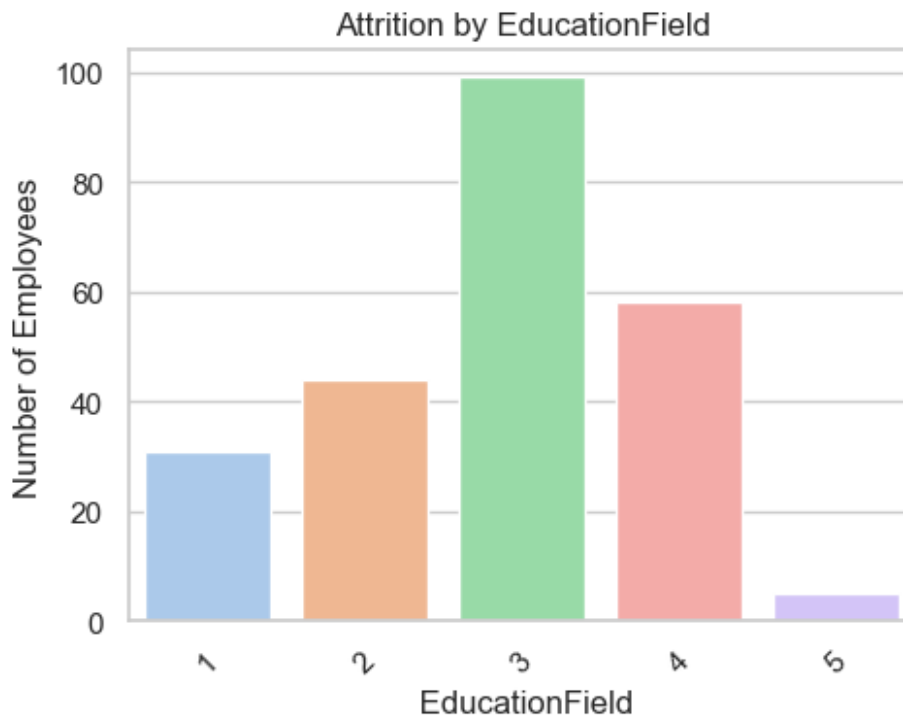
```
# Count the number of employees in each EducationField where Attrition is 'Yes'
```

```
educationfield_attrition_yes = df[df['Attrition'] == 'Yes']
['EducationField'].value_counts()
print("EducationField by Attrition (where Attrition is 'Yes'):")
print(educationfield_attrition_yes)
```

```
EducationField by Attrition (where Attrition is 'Yes'):
```

```
EducationField
Life Sciences      89
Medical           63
Marketing          35
Technical Degree   32
Other             11
Human Resources     7
Name: count, dtype: int64
```

```
attrition_by_educationField = df[df['Attrition'] == 'Yes']
['EducationField'].value_counts()
bar_chart_small(attrition_by_education, "Attrition by EducationField",
" EducationField", "Number of Employees")
```



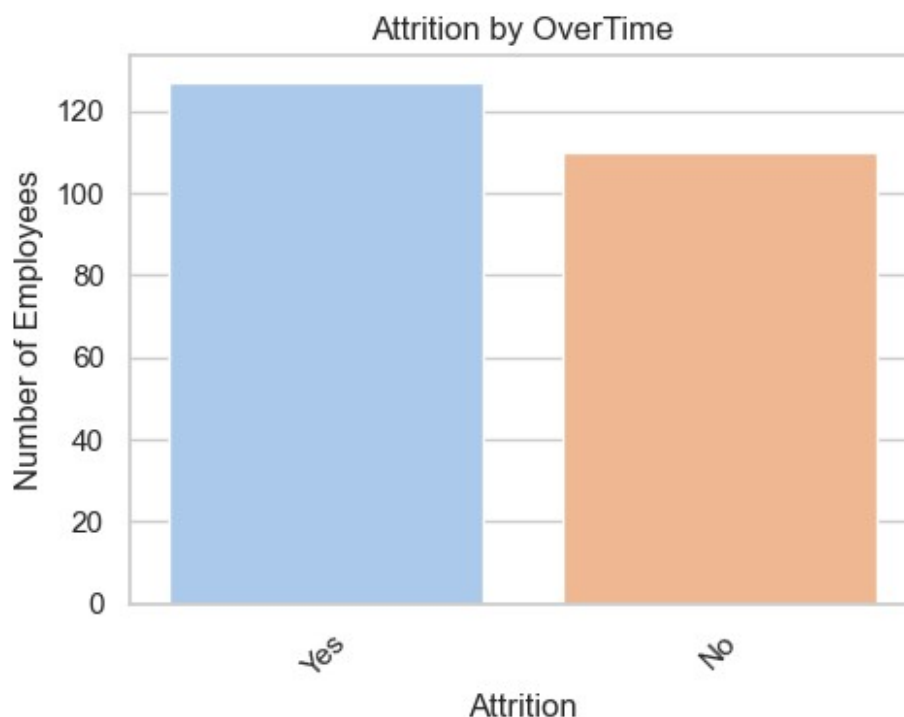
Note: Life Sciences have had significant workforce churn at the educational level.

## 8. Attrition by OverTime

```
attrition_by_OverTime = df[df['Attrition'] == 'Yes']  
['OverTime'].value_counts()  
attrition_by_OverTime
```

```
OverTime  
Yes      127  
No       110  
Name: count, dtype: int64
```

```
attrition_by_OverTime = df[df['Attrition'] == 'Yes']  
['OverTime'].value_counts()  
bar_chart_small(attrition_by_OverTime, "Attrition by OverTime",  
"Attrition", "Number of Employees")  
plt.show()
```



```
attrition_by_OverTime= df.groupby('OverTime')  
['Attrition'].value_counts(normalize=True).unstack().round(2)  
attrition_by_OverTime
```

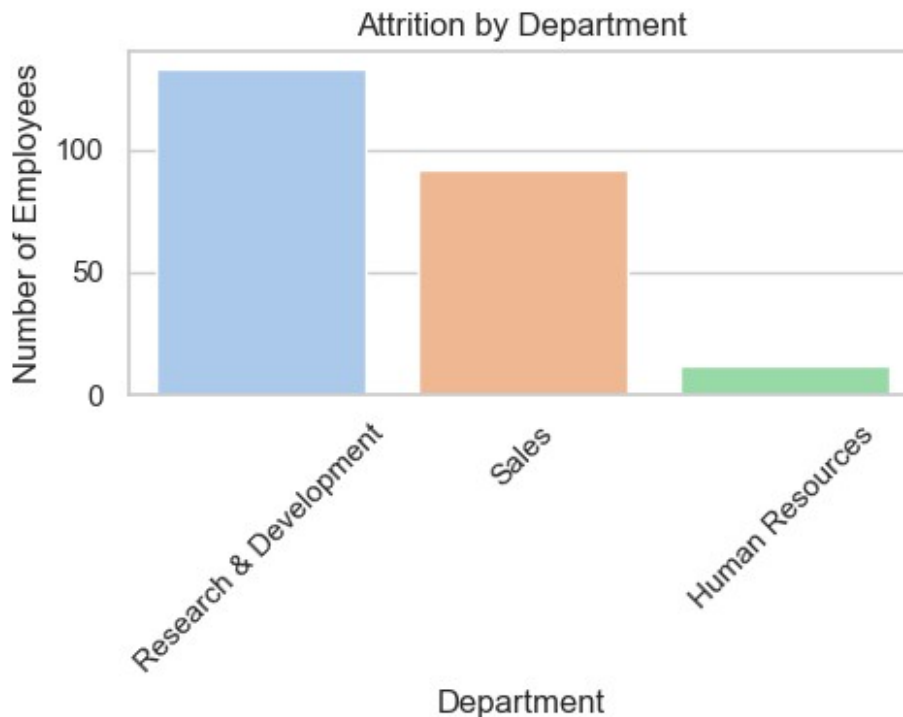
```
Attrition    No    Yes  
OverTime  
No           0.90  0.10  
Yes           0.69  0.31
```



Employee attrition does not show a significant difference based on overtime.

## 9. Attrition by Department

```
attrition_by_Department = df[df['Attrition'] == 'Yes']  
['Department'].value_counts()  
attrition_by_Department  
  
Department  
Research & Development    133  
Sales                     92  
Human Resources           12  
Name: count, dtype: int64  
  
attrition_by_Department = df[df['Attrition'] == 'Yes']  
['Department'].value_counts()  
bar_chart_small(attrition_by_Department, "Attrition by Department",  
"Department", "Number of Employees")
```



Department: Most of the employee attrition in the company has occurred in the Research and Development department. This suggests that the company is heavily focused on product or service research and development activities.

## 10. Attrition by Age

```
bins = [18, 25, 35, 45, 55, 65]  
labels = ['18-25', '26-35', '36-45', '46-55', '56-65']
```

```

df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels,
right=False)

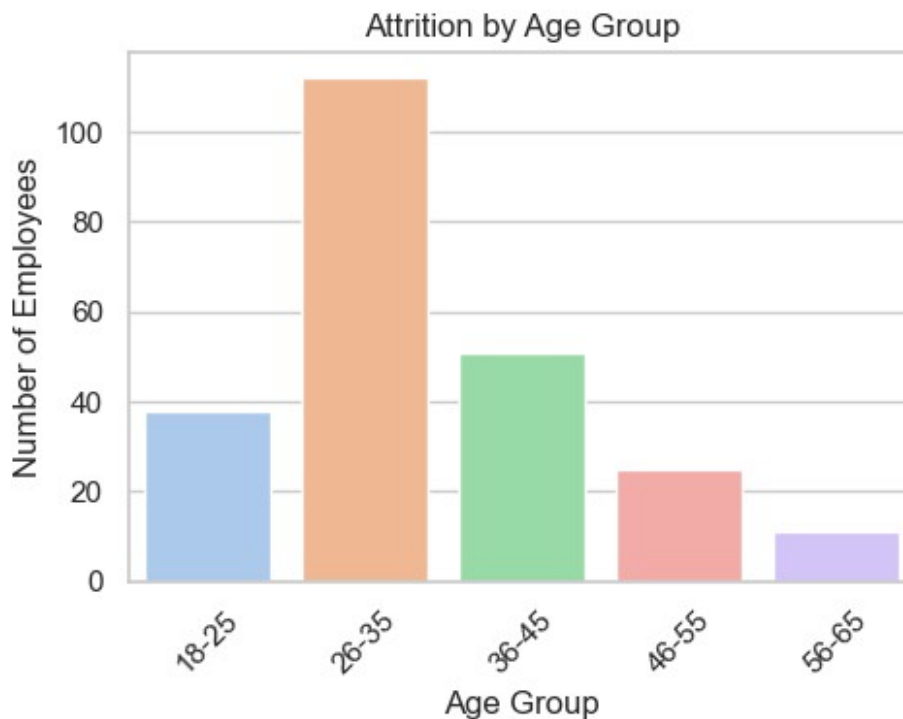
# Calculate attrition by age group
plt.figure(figsize=(4, 4))
attrition_by_age_group = df[df['Attrition'] == 'Yes']
['AgeGroup'].value_counts().sort_index()

# Plot the chart
bar_chart_small(attrition_by_age_group, "Attrition by Age Group", "Age
Group", "Number of Employees")

C:\ProgramData\anaconda3\Lib\site-packages\seaborn\categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be
changed to True in a future version of pandas. Pass observed=False to
retain current behavior or observed=True to adopt the future default
and silence this warning.
    grouped_vals = vals.groupby(grouper)

<Figure size 400x400 with 0 Axes>

```



Age: Most of the company's employees are in the 25-35 age group. This indicates that the company has many employees who are at a productive and experienced age.

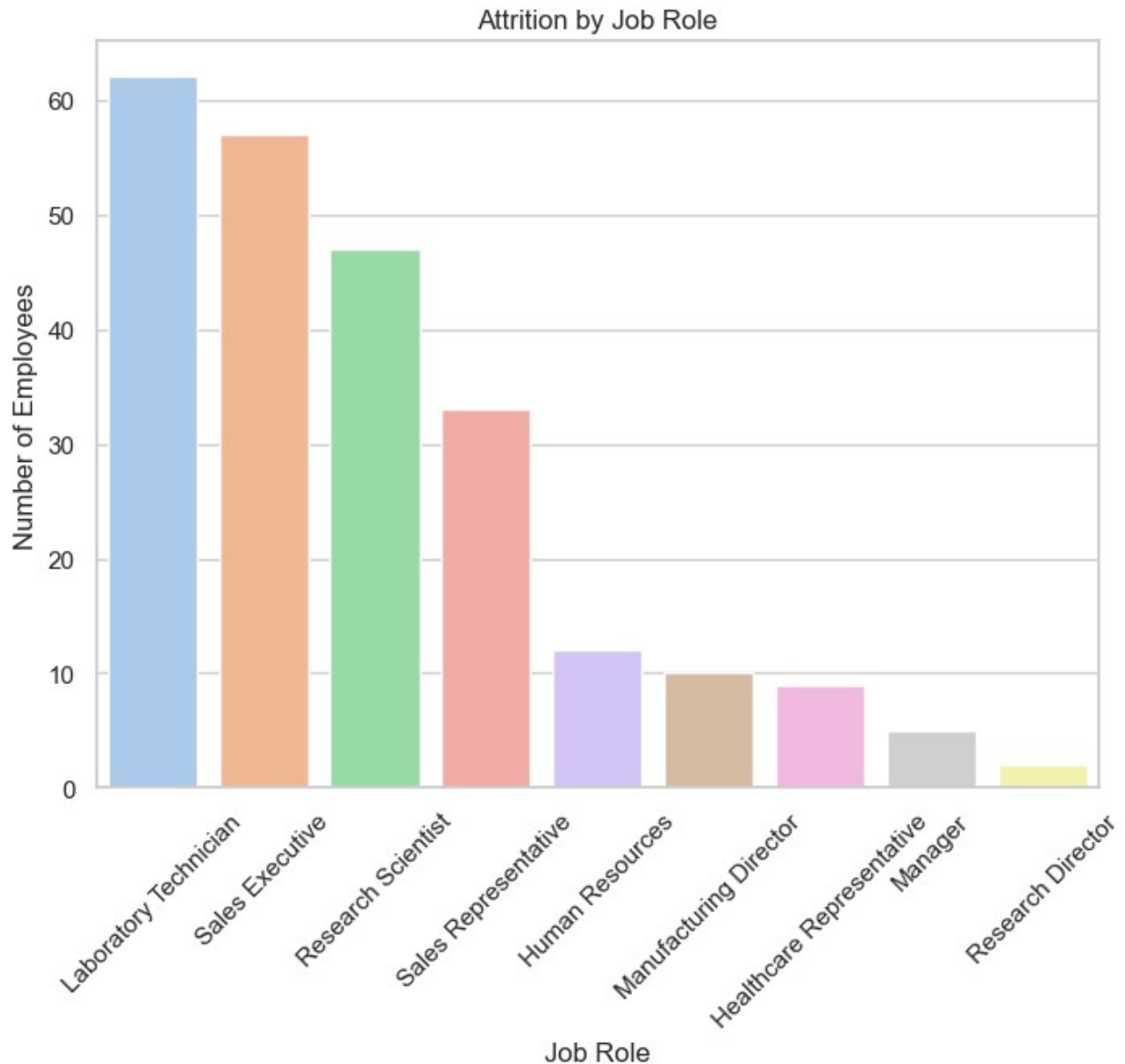
## 11. Attrition by Job Role

```
attrition_by_jobrole = df[df['Attrition'] == 'Yes']  
['JobRole'].value_counts()  
attrition_by_jobrole
```

JobRole	
Laboratory Technician	62
Sales Executive	57
Research Scientist	47
Sales Representative	33
Human Resources	12
Manufacturing Director	10
Healthcare Representative	9
Manager	5
Research Director	2

Name: count, dtype: int64

```
attrition_by_jobrole = df[df['Attrition'] == 'Yes']  
['JobRole'].value_counts()  
bar_chart(attrition_by_jobrole, "Attrition by Job Role", "Job Role",  
"Number of Employees")
```



Most employee attrition has occurred in the Laboratory Technician job role.

## 12. Attrition by Gender

```
attrition_by_gender = df[df['Attrition'] == 'Yes']  
['Gender'].value_counts()  
attrition_by_gender
```

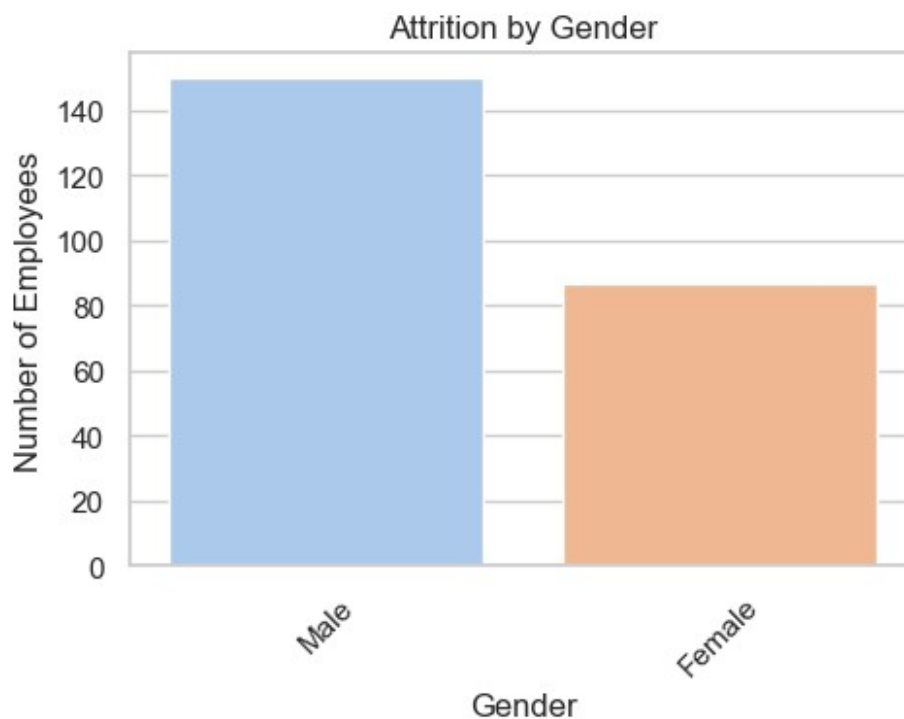
```
Gender  
Male      150  
Female     87  
Name: count, dtype: int64
```

```
attrition_by_gender = df[df['Attrition'] == 'Yes']  
['Gender'].value_counts(normalize=True)  
attrition_by_gender
```

```
Gender  
Male      0.632911  
Female    0.367089  
Name: proportion, dtype: float64
```

Male employees are more likely than female employees to leave the organization.

```
attrition_by_gender = df[df['Attrition'] == 'Yes']  
['Gender'].value_counts()  
bar_chart_small(attrition_by_gender, "Attrition by Gender", "Gender",  
"Number of Employees")
```



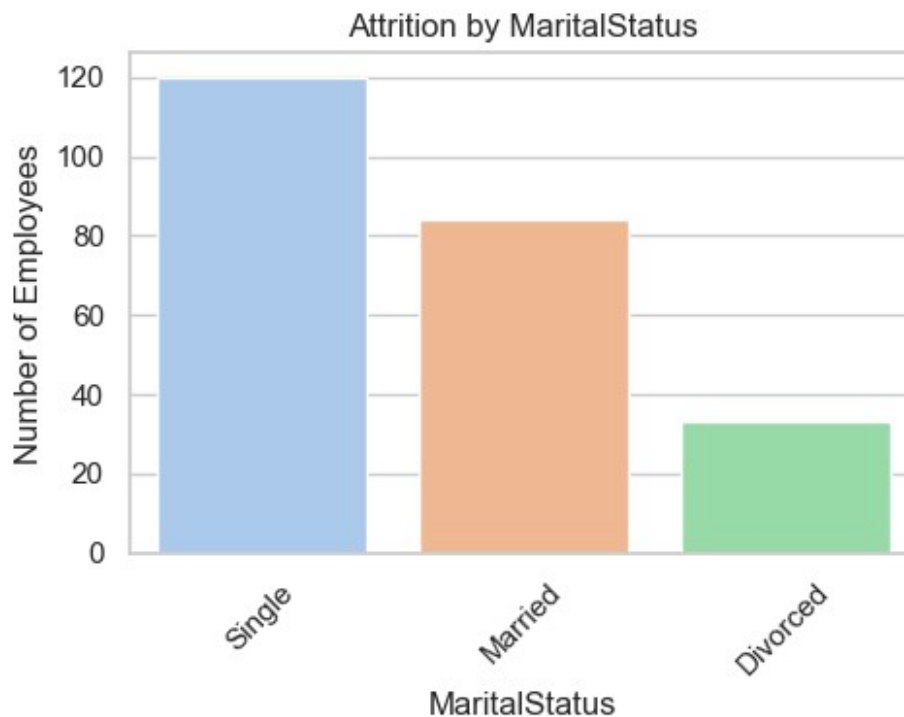
```
df['MaritalStatus'].value_counts()
```

```
MaritalStatus  
Married      673  
Single       470  
Divorced     327  
Name: count, dtype: int64
```

```
df['MaritalStatus'].value_counts(normalize=True)
```

```
MaritalStatus
Married      0.457823
Single       0.319728
Divorced     0.222449
Name: proportion, dtype: float64

attrition_by_MaritalStatus = df[df['Attrition'] == 'Yes']
['MaritalStatus'].value_counts()
bar_chart_small(attrition_by_MaritalStatus, "Attrition by
MaritalStatus", "MaritalStatus", "Number of Employees")
```



Compared to unmarried and divorced employees, married employees have quit the Company.