Importing Libraries and Data

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
df_raw = pd.read_excel("HR_Data.xlsx")
```

In [4]:
```
df_raw
```

Out[4]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Performance Rating	Relationship Satisfaction
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	3	1
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	4	4
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	3	2
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	3	3
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	3	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	Yes	Non-Travel	25 - 34	Ex-Employees	R&D	Technical Degree	STAFF-1905	1905	Male	Research Scientist	...	4	2
1466	Yes	Travel_Frequently	25 - 34	Ex-Employees	R&D	Life Sciences	STAFF-1868	1868	Male	Research Scientist	...	4	2
1467	Yes	Travel_Frequently	35 - 44	Ex-Employees	Sales	Other	STAFF-1667	1667	Male	Sales Executive	...	4	1
1468	Yes	Travel_Rarely	Under 25	Ex-Employees	R&D	Life Sciences	STAFF-1878	1878	Male	Research Scientist	...	4	1
1469	Yes	Travel_Rarely	Under 25	Ex-Employees	Sales	Life Sciences	STAFF-1702	1702	Male	Sales Representative	...	4	2

1470 rows × 39 columns

In [5]:
```
df_raw.columns
```

Out[5]:
```
Index(['Attrition', 'Business Travel', 'CF_age band', 'CF_attrition label',
       'Department', 'Education Field', 'emp no', 'Employee Number', 'Gender',
       'Job Role', 'Marital Status', 'Over Time', 'Over18',
       'Training Times Last Year', 'Age', 'CF_current Employee', 'Daily Rate',
       'Distance From Home', 'Education', 'Employee Count',
       'Environment Satisfaction', 'Hourly Rate', 'Job Involvement',
       'Job Level', 'Job Satisfaction', 'Monthly Income', 'Monthly Rate',
       'Num Companies Worked', 'Percent Salary Hike', 'Performance Rating',
       'Relationship Satisfaction', 'Standard Hours', 'Stock Option Level',
       'Total Working Years', 'Work Life Balance', 'Years At Company',
       'Years In Current Role', 'Years Since Last Promotion',
       'Years With Curr Manager'],
      dtype='object')
```

Manipulating Data

Creating new column to hold 'Attrition' column values in a numeric form (int data type)

In [12]:
```
df_raw['Attrition_int'] = np.where(df_raw['Attrition'] == 'Yes', 1, 0)
```

In [13]:
```
df_raw['Attrition_int']
```

Out[13]:
```
0       1
1       0
2       1
3       0
4       0
       ..
1465    1
1466    1
1467    1
1468    1
1469    1
Name: Attrition_int, Length: 1470, dtype: int32
```

In [14]:
```
df_raw
```

Out[14]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Relationship Satisfaction	Standard Hours	Sto Opti Le
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	1	80	
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	4	80	
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	2	80	
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	3	80	
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	4	80	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	Yes	Non-Travel	25 - 34	Ex-Employees	R&D	Technical Degree	STAFF-1905	1905	Male	Research Scientist	...	2	80	
1466	Yes	Travel_Frequently	25 - 34	Ex-Employees	R&D	Life Sciences	STAFF-1868	1868	Male	Research Scientist	...	2	80	
1467	Yes	Travel_Frequently	35 - 44	Ex-Employees	Sales	Other	STAFF-1667	1667	Male	Sales Executive	...	1	80	
1468	Yes	Travel_Rarely	Under 25	Ex-Employees	R&D	Life Sciences	STAFF-1878	1878	Male	Research Scientist	...	1	80	
1469	Yes	Travel_Rarely	Under 25	Ex-Employees	Sales	Life Sciences	STAFF-1702	1702	Male	Sales Representative	...	2	80	

1470 rows × 40 columns

First checkpoint (attrition_int column addition)

In [15]:
```
df_check_point1 = df_raw
```

In [16]:
```
df_check_point1
```

Out[16]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Relationship Satisfaction	Standard Hours	Sto Opti Le
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	1	80	
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	4	80	
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	2	80	
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	3	80	
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	4	80	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	Yes	Non-Travel	25 - 34	Ex-Employees	R&D	Technical Degree	STAFF-1905	1905	Male	Research Scientist	...	2	80	
1466	Yes	Travel_Frequently	25 - 34	Ex-Employees	R&D	Life Sciences	STAFF-1868	1868	Male	Research Scientist	...	2	80	
1467	Yes	Travel_Frequently	35 - 44	Ex-Employees	Sales	Other	STAFF-1667	1667	Male	Sales Executive	...	1	80	
1468	Yes	Travel_Rarely	Under 25	Ex-Employees	R&D	Life Sciences	STAFF-1878	1878	Male	Research Scientist	...	1	80	
1469	Yes	Travel_Rarely	Under 25	Ex-Employees	Sales	Life Sciences	STAFF-1702	1702	Male	Sales Representative	...	2	80	

1470 rows × 40 columns

Creating new column to hold 'Business Travel' column values in a numeric form (int data type)

In [17]:
```
df_raw['Business Travel'].values
```

Out[17]:
```
array(['Travel_Rarely', 'Travel_Frequently', 'Travel_Rarely', ...,
       'Travel_Frequently', 'Travel_Rarely', 'Travel_Rarely'],
      dtype=object)
```

In [18]:
```
df_raw['Business Travel'].unique()
```

Out[18]:
```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'], dtype=object)
```

0-non-travel, 1-travel rarely, 2-travel frequently

In [23]:
```
d =  { 'Non-Travel' : 0,  'Travel_Rarely' : 1, 'Travel_Frequently' : 2}
df_raw['Travel_Modes'] = df_raw['Business Travel'].map(d)
```

In [24]:
```
df_raw
```

Out[24]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Standard Hours	Stock Option Level	Total Working Years
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	80	0	8
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	80	1	10
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	80	0	7
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	80	0	8
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	80	1	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	Yes	Non-Travel	25 - 34	Ex-Employees	R&D	Technical Degree	STAFF-1905	1905	Male	Research Scientist	...	80	1	5
1466	Yes	Travel_Frequently	25 - 34	Ex-Employees	R&D	Life Sciences	STAFF-1868	1868	Male	Research Scientist	...	80	0	1
1467	Yes	Travel_Frequently	35 - 44	Ex-Employees	Sales	Other	STAFF-1667	1667	Male	Sales Executive	...	80	0	13
1468	Yes	Travel_Rarely	Under 25	Ex-Employees	R&D	Life Sciences	STAFF-1878	1878	Male	Research Scientist	...	80	0	1
1469	Yes	Travel_Rarely	Under 25	Ex-Employees	Sales	Life Sciences	STAFF-1702	1702	Male	Sales Representative	...	80	1	3

1470 rows × 41 columns

Second checkpoint (Travel_Modes column addition)

In [25]:
```
df_check_point2 = df_raw
```

Creating 'Att_label_boolean' colum to hold 'CF_attrition label' in a boolean form

In [26]:
```
df_raw['CF_attrition label'].unique()
```

Out[26]:
```
array(['Ex-Employees', 'Current Employees'], dtype=object)
```

In [27]:
```
df_raw['Att_label_boolean'] = np.where(df_raw['CF_attrition label'] == 'Ex-Employees', 0, 1)
```

In [28]:
```
df_raw
```

Out[28]:

emp no	Employee Number	Gender	Job Role	...	Stock Option Level	Total Working Years	Work Life Balance	Years At Company	Years In Current Role	Years Since Last Promotion	Years With Curr Manager	Attrition_int	Travel_Modes	Att_label_bo
STAFF-1	1	Female	Sales Executive	...	0	8	1	6	4	0	5	1	1	
STAFF-2	2	Male	Research Scientist	...	1	10	3	10	7	1	7	0	2	
STAFF-4	4	Male	Laboratory Technician	...	0	7	3	0	0	0	0	1	1	
STAFF-5	5	Female	Research Scientist	...	0	8	3	8	7	3	0	0	2	
STAFF-7	7	Male	Laboratory Technician	...	1	6	3	2	2	2	2	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
STAFF-1905	1905	Male	Research Scientist	...	1	5	3	5	2	3	0	1	0	
STAFF-1868	1868	Male	Research Scientist	...	0	1	2	1	0	1	0	1	2	
STAFF-1667	1667	Male	Sales Executive	...	0	13	4	11	9	6	7	1	2	
STAFF-1878	1878	Male	Research Scientist	...	0	1	3	1	0	0	0	1	1	
STAFF-1702	1702	Male	Sales Representative	...	1	3	3	3	2	0	2	1	1	

Third checkpoint ('Att_label_boolean' column addition)

In [29]:
```
df_check_point3 = df_raw
```

Checking for duplicates in the dataset

In [31]:
```
df_raw.duplicated().unique()
```

Out[31]:
```
array([False])
```

Searching for missing values

In [37]:
```
df_raw.isna()
```

Out[37]:

Education Field	emp no	Employee Number	Gender	Job Role	...	Stock Option Level	Total Working Years	Work Life Balance	Years At Company	Years In Current Role	Years Since Last Promotion	Years With Curr Manager	Attrition_int	Travel_Modes	Att_label_bo
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	
False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	

In [38]:
```
df_unique = df_raw.isna().drop_duplicates()
```

In [39]:
```
df_unique
```

Out[39]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Stock Option Level	Total Working Years	Work Life Balance	Years At Company	Years In Current Role	P
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	

1 rows × 42 columns

In [41]:
```
df_unique.values
```

Out[41]:
```
array([[False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False]])
```

Conclusion: There are no missing values

Visualization

importing matplotlib

In [43]:
```
import matplotlib.pyplot as plt #visualization of data
```

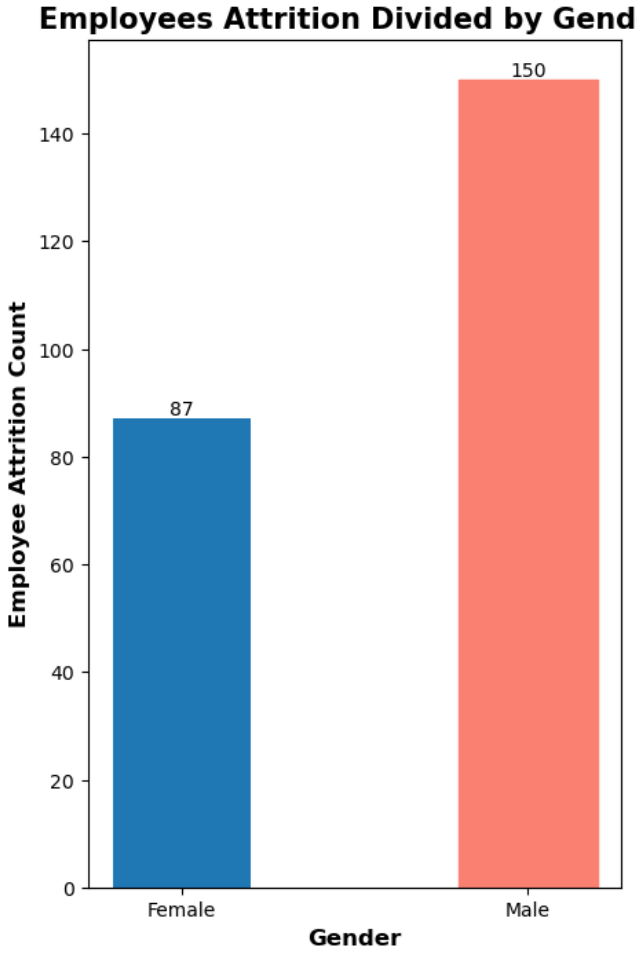
Grouping by gender and calculating the attrition in each group

In [49]:
```
attrition_by_gender = df_raw.groupby('Gender')['Attrition_int'].sum()
```

In [50]:
```
attrition_by_gender
```

Out[50]:
```
Gender
Female     87
Male      150
Name: Attrition_int, dtype: int32
```

In [256]:
```
plt.figure(figsize=(5,8))
x_pos = [0,0.2]
bars = plt.bar(x = x_pos , height = attrition_by_gender, width = 0.08)
plt.bar_label(bars)
bars[1].set_color('salmon')
plt.xticks(x_pos, attrition_by_gender.index)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Gender', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition Divided by Gender", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



Conclusion: The attrition is visibly higher in male employees than in female employees. Meaning, there is a greater chance for a male employee to leave the company.

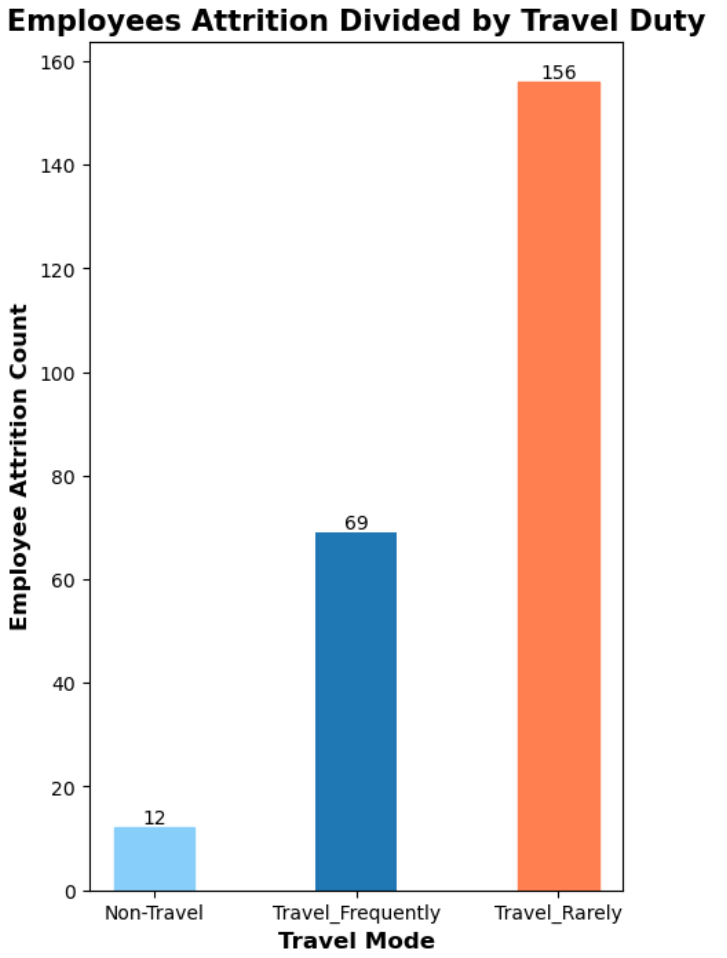
Grouping by 'Business Travel' and calculating attrition in each group

In [79]:
```
attrition_by_travel = df_raw.groupby('Business Travel')['Attrition_int'].sum()
```

In [80]:
```
attrition_by_travel
```

Out[80]:
```
Business Travel
Non-Travel            12
Travel_Frequently     69
Travel_Rarely        156
Name: Attrition_int, dtype: int32
```

In [110]:
```
plt.figure(figsize=(5,8))
x_pos = [0, 0.2, 0.4]
bars = plt.bar(x = x_pos , height = attrition_by_travel, width = 0.08)
plt.bar_label(bars)
bars[2].set_color('coral')
bars[0].set_color('lightskyblue')
plt.xticks(x_pos, attrition_by_travel.index)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Travel Mode', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition Divided by Travel Duty", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



Conclusion: Interesitingly, employees who travel rarely are effected most by attrition.

Grouping by 'Department' and calculating attrition in each group

In [85]:
```
df_raw['Department'].unique()
```

Out[85]:
```
array(['Sales', 'R&D', 'HR'], dtype=object)
```

In [86]:
```
attrition_by_department = df_raw.groupby('Department')['Attrition_int'].sum()
```

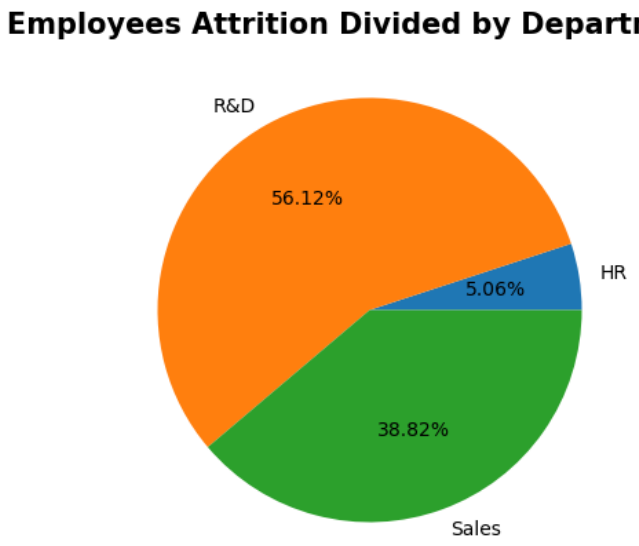
In [87]:
```
attrition_by_department
```

Out[87]:
```
Department
HR        12
R&D      133
Sales     92
Name: Attrition_int, dtype: int32
```

In [89]:
```
attrition_by_department.values
```

Out[89]:
```
array([ 12, 133,  92])
```

In [94]:
```
plt.figure(figsize=(7,5))
plt.pie(attrition_by_department, labels = attrition_by_department.index, autopct = '%.2f%%')
plt.title("Employees Attrition Divided by Departments", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



Conclusion: The highest rate of attrition is spotted in the R&D department. Sales department has also a significant rate of attrition.

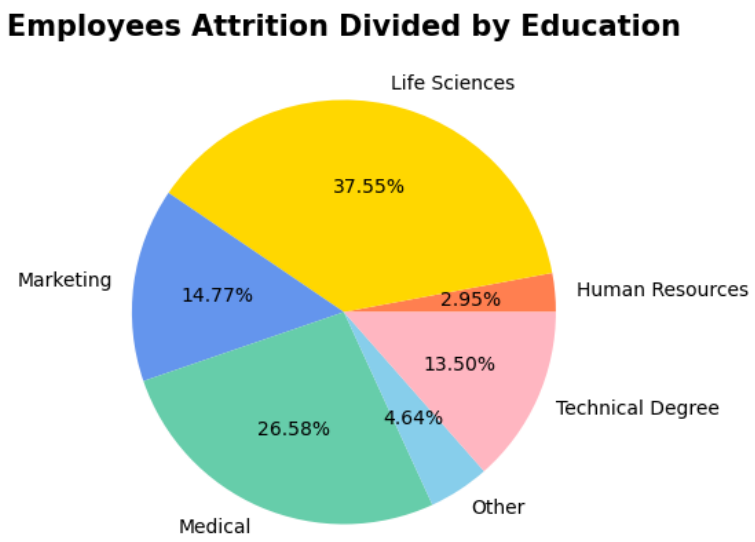
Grouping by 'Education Field' and calculating attrition in each group

In [98]:
```
attrition_by_education = df_raw.groupby('Education Field')['Attrition_int'].sum()
```

In [99]:
```
attrition_by_education
```

Out[99]:
```
Education Field
Human Resources      7
Life Sciences       89
Marketing           35
Medical             63
Other               11
Technical Degree    32
Name: Attrition_int, dtype: int32
```

In [103]:
```
plt.figure(figsize=(7,5))
plt.pie(attrition_by_education, labels = attrition_by_education.index, colors = ['coral', 'gold', 'cornflowerblue', 'mediuma
plt.title("Employees Attrition Divided by Education", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



Conclusion: Attrition presents most in employees from Life Sciences and Medical fields.

Grouping by 'Total Working Years' and calculating attrition in each group

In [111]:
```
df_raw['Total Working Years'].unique()
```

Out[111]:
```
array([ 8, 10,  7,  6, 12,  1, 17,  5,  3, 31, 13,  0, 26, 24, 22,  9, 19,
        2, 23, 14, 15,  4, 29, 28, 21, 25, 20, 11, 16, 37, 38, 30, 40, 18,
       36, 34, 32, 33, 35, 27], dtype=int64)
```

In [114]:
```
attrition_by_work_years = df_raw.groupby('Total Working Years')['Attrition_int'].sum()
```

In [115]:
```
attrition_by_work_years
```

Out[115]:
```
Total Working Years
0      5
1     40
2      9
3      9
4     12
5     16
6     22
7     18
8     16
9     10
10    25
11     7
12     5
13     3
14     4
15     5
16     3
17     3
18     4
19     3
20     2
21     1
22     2
23     2
24     3
25     1
26     1
27     0
28     1
29     0
30     0
31     1
32     0
33     1
34     1
35     0
36     0
37     0
38     0
40     2
Name: Attrition_int, dtype: int32
```

In [129]:
```
attrition_by_work_years.index
```

Out[129]:
```
Int64Index([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
            17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
            34, 35, 36, 37, 38, 40],
           dtype='int64', name='Total Working Years')
```

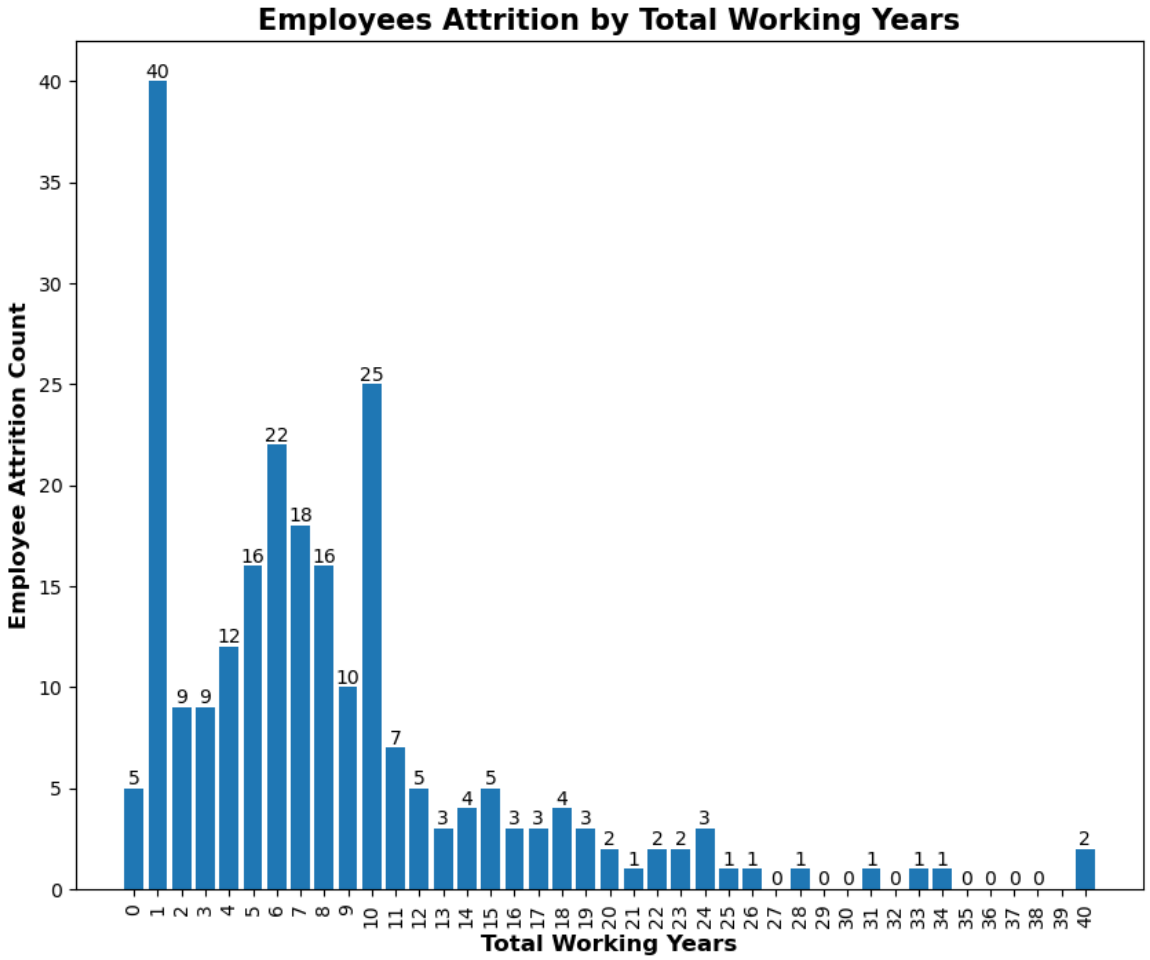
In [137]:
```
type(attrition_by_work_years)
```

Out[137]:
```
pandas.core.series.Series
```

In [165]:
```
type(bars)
```

Out[165]:
```
matplotlib.container.BarContainer
```

In [372]:
```
plt.figure(figsize=(10,8))
bars = plt.bar(x = attrition_by_work_years.index, height = attrition_by_work_years)
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Total Working Years', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Total Working Years", fontdict={'size':15, 'fontweight':'bold'})
plt.xticks(range(0,41,1), rotation = 90)
plt.show()
```

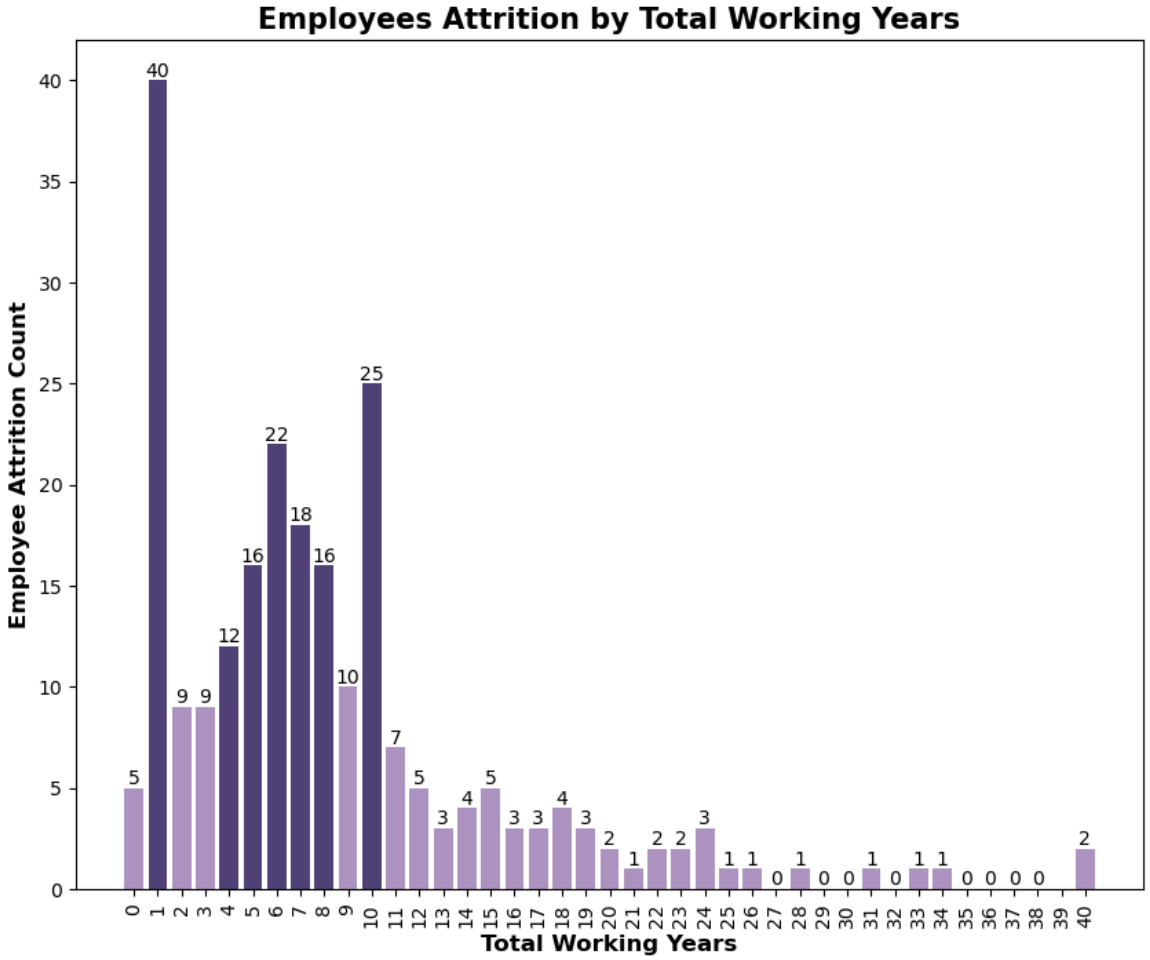


Conclusion: Most employees who affected by attrition are those who worked for 1 year in total.

Function that creates a list with colors based on the attrition number (threshold = 10)

In [178]:
```
color_list = []
def Under10Color(attrition_by_work_years):
    for i in attrition_by_work_years:
        if i > 10:
            color_list.append('#4F4176')
        else:
            color_list.append('#AB92BF')
    return color_list
```

In [371]:
```
plt.figure(figsize=(10,8))
bars = plt.bar(x = attrition_by_work_years.index, height = attrition_by_work_years, color = Under10Color(attrition_by_work_y
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Total Working Years', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Total Working Years", fontdict={'size':15, 'fontweight':'bold'})
plt.xticks(range(0,41,1), rotation = 90)
plt.show()
```



In the chart above, the total working years, which were affected most by attrition, were marked (attrition > 10). These years are 1 > 10 > 6 > 7> 8,5 > 4.

Grouping by total working years and department and calculating the attrition

In [184]:
```
attrition_by_work_years_dep = df_raw.groupby(['Total Working Years', 'Department'])['Attrition_int'].sum()
```

In [185]:
```
attrition_by_work_years_dep
```

Out[185]:
```
Total Working Years  Department
0                    R&D           2
                     Sales         3
1                    HR            3
                     R&D          25
                     Sales        12
                                  ..
36                   Sales         0
37                   R&D           0
                     Sales         0
38                   Sales         0
40                   R&D           2
Name: Attrition_int, Length: 104, dtype: int32
```

In [193]:
```
type(attrition_by_work_years_dep)
```

Out[193]:
```
pandas.core.series.Series
```

Converting the grouped data to DF

In [202]:
```
attrition_by_work_years_dep = attrition_by_work_years_dep.to_frame()
```

In [203]:
```
type(attrition_by_work_years_dep)
```

Out[203]:
```
pandas.core.frame.DataFrame
```

In [211]:
```
attrition_by_work_years_dep.index
```

Out[211]:
```
MultiIndex([( 0,   'R&D'),
            ( 0, 'Sales'),
            ( 1,    'HR'),
            ( 1,   'R&D'),
            ( 1, 'Sales'),
            ( 2,    'HR'),
            ( 2,   'R&D'),
            ( 2, 'Sales'),
            ( 3,    'HR'),
            ( 3,   'R&D'),
            ...
            (35,    'HR'),
            (35,   'R&D'),
            (35, 'Sales'),
            (36,    'HR'),
            (36,   'R&D'),
            (36, 'Sales'),
            (37,   'R&D'),
            (37, 'Sales'),
            (38, 'Sales'),
            (40,   'R&D')],
           names=['Total Working Years', 'Department'], length=104)
```

In [212]:
```
attrition_by_work_years_dep['Attrition_int']
```

Out[212]:
```
Total Working Years  Department
0                    R&D           2
                     Sales         3
1                    HR            3
                     R&D          25
                     Sales        12
                                  ..
36                   Sales         0
37                   R&D           0
                     Sales         0
38                   Sales         0
40                   R&D           2
Name: Attrition_int, Length: 104, dtype: int32
```

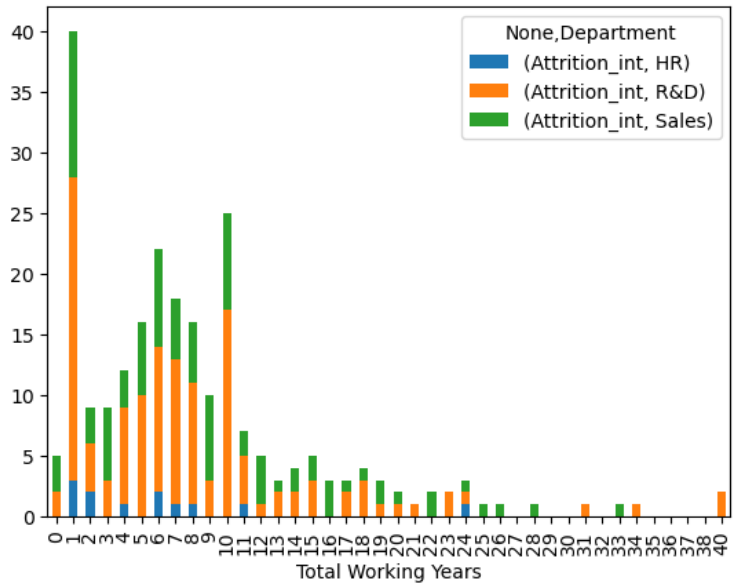
In [214]:
```
type(attrition_by_work_years_dep.index)
```

Out[214]:
```
pandas.core.indexes.multi.MultiIndex
```

Using 'unstack' method to create stacked bar chart (DF method)

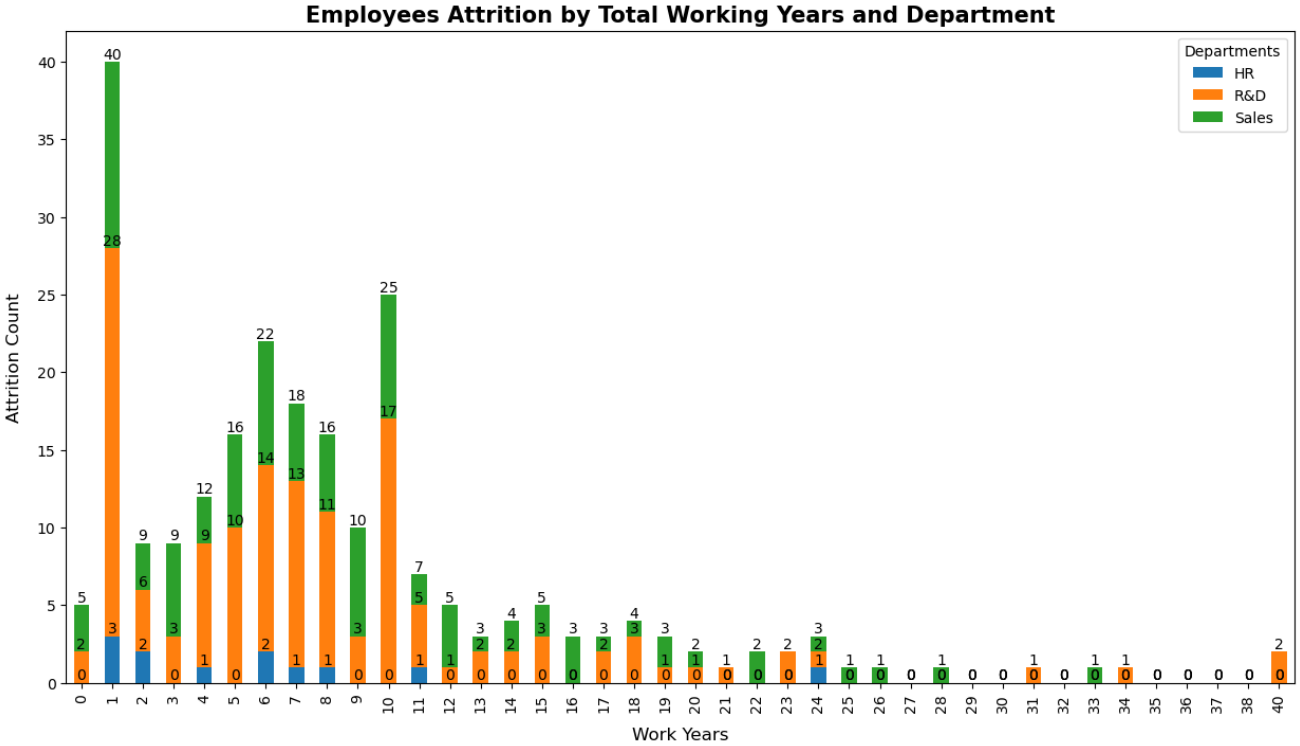
In [215]:
```
attrition_by_work_years_dep.unstack().plot(kind='bar', stacked=True)
```

Out[215]: <AxesSubplot:xlabel='Total Working Years'>



Adding labels to each segment of the stacked bar

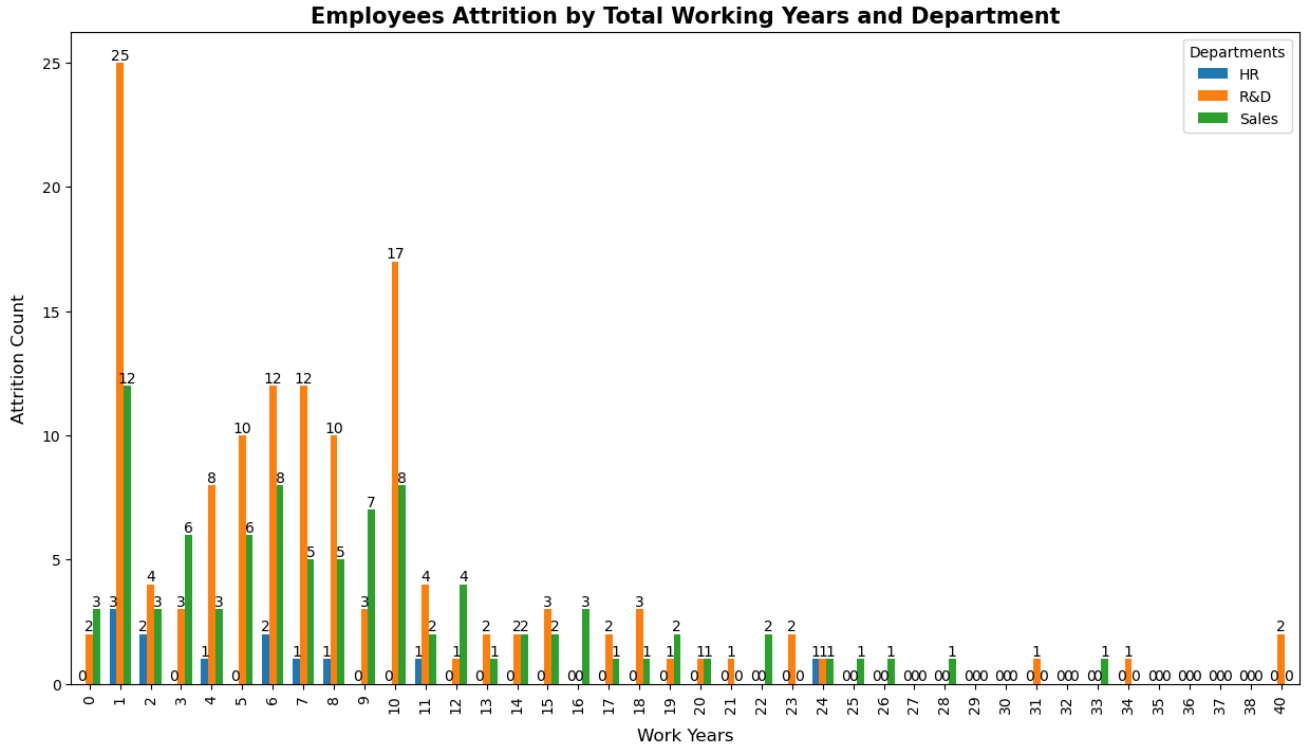
In [239]:
```
ax = attrition_by_work_years_dep.unstack().plot(kind='bar', stacked=True, figsize =(15,8))
ax.legend(["HR", "R&D", "Sales"],title='Departments', fontsize=10, loc='upper right')
ax.set_xlabel('Work Years', fontsize=12, labelpad=10)
ax.set_ylabel('Attrition Count', fontsize=12, labelpad=10)
for container in ax.containers:
    ax.bar_label(container, label_type='edge', fontsize=10)
plt.title("Employees Attrition by Total Working Years and Department", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



Conclusion: The years of interest (1,10,6,7,8,5,4) can be further explored. For example, the employees who leave after their first working year, are mostly leaving from the R&D department.

Creating a multi-bar chart

In [255]:
```
ax = attrition_by_work_years_dep.unstack().plot(kind='bar', figsize =(15,8), width = 0.7)
ax.legend(["HR", "R&D", "Sales"],title='Departments', fontsize=10, loc='upper right')
ax.set_xlabel('Work Years', fontsize=12, labelpad=10)
ax.set_ylabel('Attrition Count', fontsize=12, labelpad=10)
for container in ax.containers:
    ax.bar_label(container, label_type='edge', fontsize=10)
plt.title("Employees Attrition by Total Working Years and Department", fontdict={'size':15, 'fontweight':'bold'})
plt.show()
```



In this way of representation, we can have the accurate numerical values of attrition in each department.

Grouping 'CF_age band' and calculating attrition in relation to age

In [258]:
```
df_raw['CF_age band'].unique()
```

Out[258]:
```
array(['35 - 44', '45 - 54', '25 - 34', 'Over 55', 'Under 25'],
      dtype=object)
```

In [295]:
```
attrition_by_age_group = df_raw.groupby('CF_age band')['Attrition_int'].sum()
```

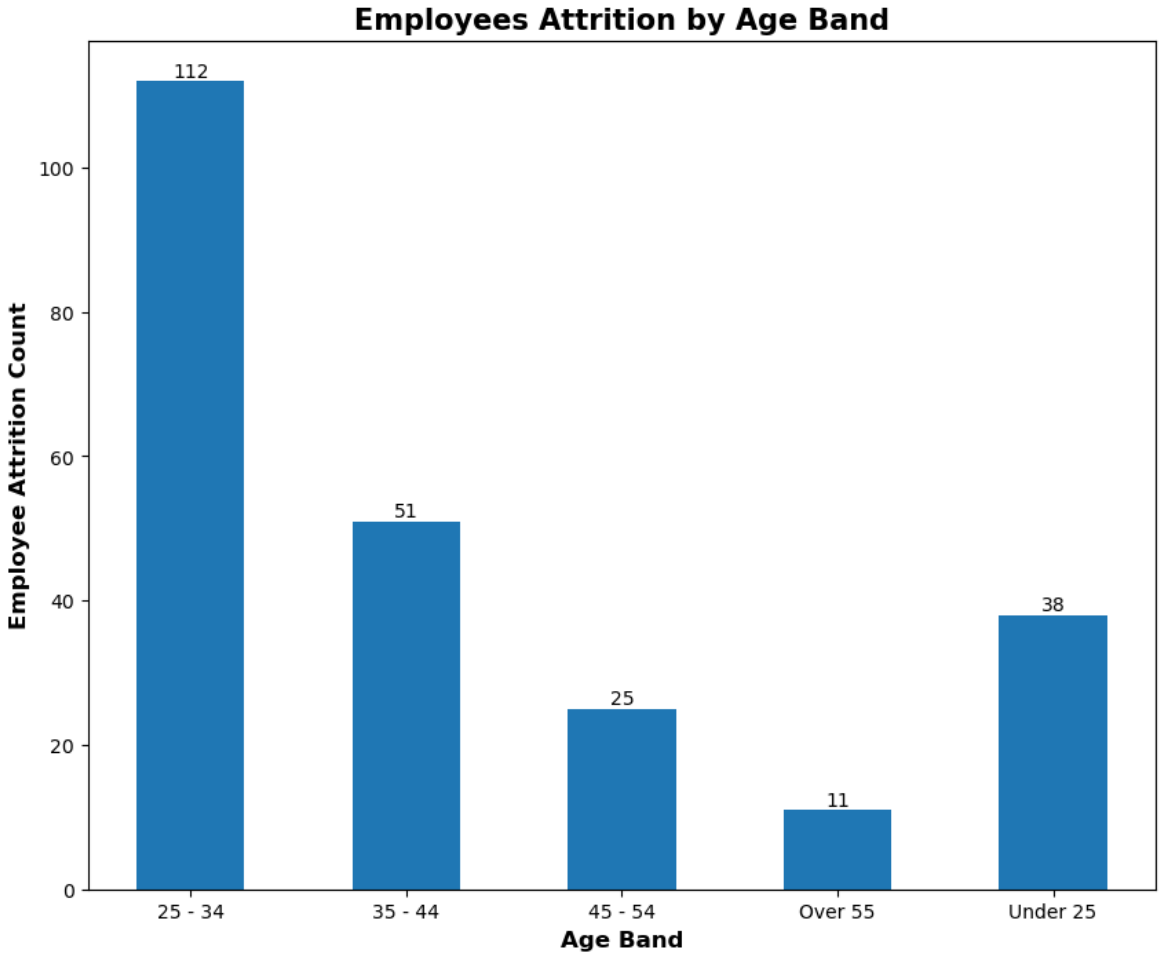
In [296]:
```
attrition_by_age_group
```

Out[296]:
```
CF_age band
25 - 34     112
35 - 44      51
45 - 54      25
Over 55      11
Under 25     38
Name: Attrition_int, dtype: int32
```

In [263]:
```
attrition_by_age_group.index
```

Out[263]:
```
Index(['25 - 34', '35 - 44', '45 - 54', 'Over 55', 'Under 25'], dtype='object', name='CF_age band')
```

In [267]:
```
plt.figure(figsize=(10,8))
list_pos = []
for i in attrition_by_age_group.index:
    list_pos.append(i)
bars = plt.bar(x = list_pos, height = attrition_by_age_group, width = 0.5)
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Age Band', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Age Band", fontdict={'size':15, 'fontweight':'bold'})
x = np.arange(0,attrition_by_age_group.shape[0],1)
plt.xticks(x)
plt.show()
```



Conclusion: the most affected group is of 25-34 years of age.

Changing bars order to appear choronologically

In [268]:
```
type(attrition_by_age_group.index)
```

Out[268]: pandas.core.indexes.base.Index

In [271]:
```
type(attrition_by_age_group)
```

Out[271]: pandas.core.series.Series

In [297]:
```
attrition_by_age_group.values
```

Out[297]: array([112, 51, 25, 11, 38])

Creating a copy of the original series

In [298]:
```
attrition_by_age_group_reorder = attrition_by_age_group
```

In [299]:
```
attrition_by_age_group
```

Out[299]:
```
CF_age band
25 - 34     112
35 - 44      51
45 - 54      25
Over 55      11
Under 25     38
Name: Attrition_int, dtype: int32
```

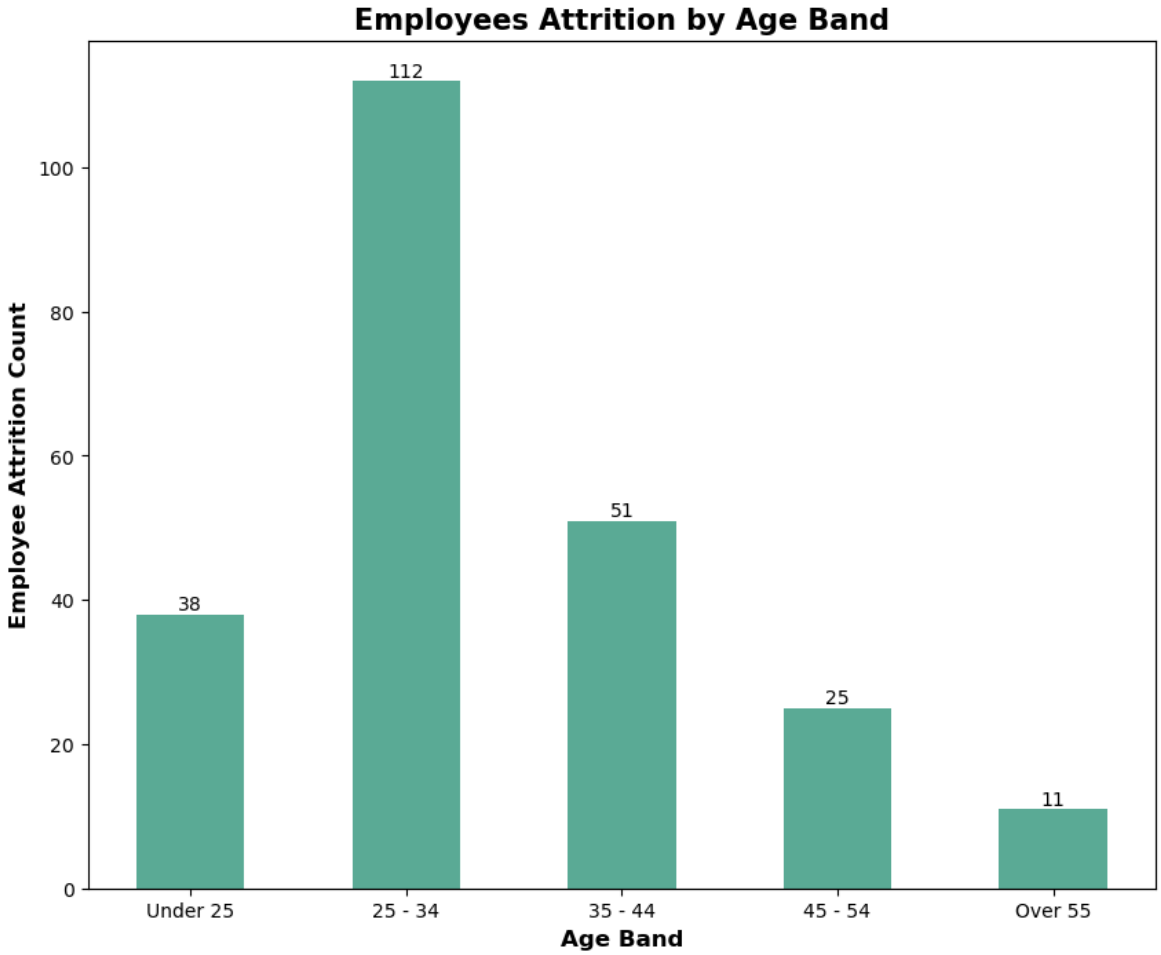
Changing indices and values

In [ ]:
```
attrition_by_age_group_reorder = attrition_by_age_group_reorder.reindex(['Under 25','25 - 34','35 - 44','45 - 54','Over 55']
```

In [309]:
```
attrition_by_age_group_reorder
```

Out[309]:
```
CF_age band
Under 25     38
25 - 34     112
35 - 44      51
45 - 54      25
Over 55      11
Name: Attrition_int, dtype: int32
```

In [313]:
```
plt.figure(figsize=(10,8))
list_pos = []
for i in attrition_by_age_group_reorder.index:
    list_pos.append(i)
bars = plt.bar(x = list_pos, height = attrition_by_age_group_reorder, width = 0.5, color = '#5AAA95')
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Age Band', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Age Band", fontdict={'size':15, 'fontweight':'bold'})
x = np.arange(0,attrition_by_age_group_reorder.shape[0],1)
plt.xticks(x)
plt.show()
```



Grouping by 'Years At Company' and calculating the attrition

In [314]:
```
df_raw['Years At Company'].unique()
```

Out[314]:
```
array([ 6, 10,  0,  8,  2,  7,  1,  9,  5,  4, 25,  3, 12, 14, 22, 15, 27,
       21, 17, 11, 13, 37, 16, 20, 40, 24, 33, 19, 36, 18, 29, 31, 32, 34,
       26, 30, 23], dtype=int64)
```

In [317]:
```
df_raw['Years At Company'].nunique()
```

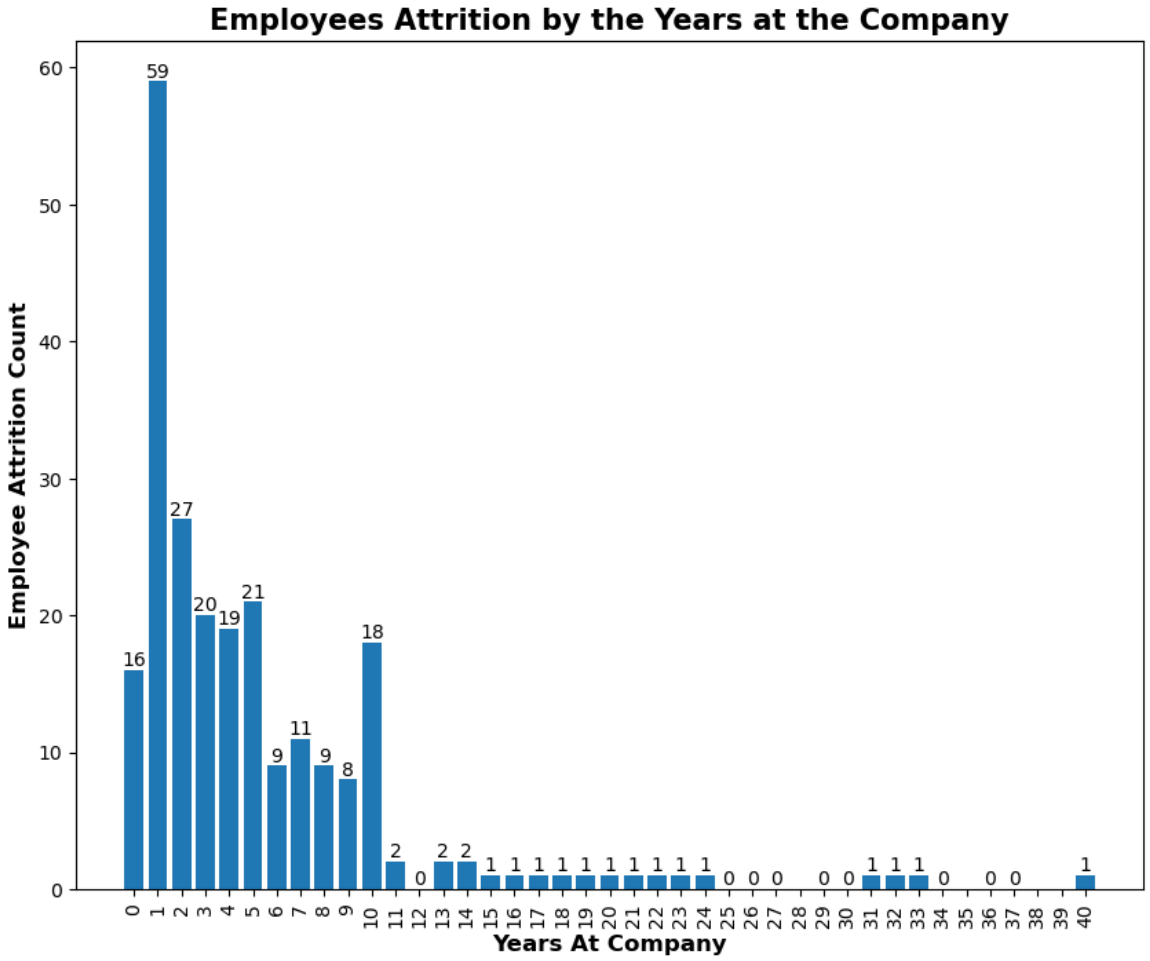
Out[317]:
```
37
```

In [319]:
```
attrition_by_years_comp = df_raw.groupby('Years At Company')['Attrition_int'].sum()
```

In [320]:
```
attrition_by_years_comp
```

Out[320]:
```
Years At Company
0     16
1     59
2     27
3     20
4     19
5     21
6      9
7     11
8      9
9      8
10    18
11     2
12     0
13     2
14     2
15     1
16     1
17     1
18     1
19     1
20     1
21     1
22     1
23     1
24     1
25     0
26     0
27     0
29     0
30     0
31     1
32     1
33     1
34     0
36     0
37     0
40     1
Name: Attrition_int, dtype: int32
```

In [370]:
```
plt.figure(figsize=(10,8))
bars = plt.bar(x = attrition_by_years_comp.index, height = attrition_by_years_comp)
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Years At Company', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by the Years at the Company", fontdict={'size':15, 'fontweight':'bold'})
plt.xticks(range(0,41,1), rotation = 90)
plt.show()
```



Conclusion: As in 'Total working years', most attritions accure after 1 year at the company .

In [327]:
```
df_raw
```

Out[327]:

	Attrition	Business Travel	CF_age band	CF_attrition label	Department	Education Field	emp no	Employee Number	Gender	Job Role	...	Stock Option Level	Total Working Years	Work Life Balance
0	Yes	Travel_Rarely	35 - 44	Ex-Employees	Sales	Life Sciences	STAFF-1	1	Female	Sales Executive	...	0	8	1
1	No	Travel_Frequently	45 - 54	Current Employees	R&D	Life Sciences	STAFF-2	2	Male	Research Scientist	...	1	10	3
2	Yes	Travel_Rarely	35 - 44	Ex-Employees	R&D	Other	STAFF-4	4	Male	Laboratory Technician	...	0	7	3
3	No	Travel_Frequently	25 - 34	Current Employees	R&D	Life Sciences	STAFF-5	5	Female	Research Scientist	...	0	8	3
4	No	Travel_Rarely	25 - 34	Current Employees	R&D	Medical	STAFF-7	7	Male	Laboratory Technician	...	1	6	3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	Yes	Non-Travel	25 - 34	Ex-Employees	R&D	Technical Degree	STAFF-1905	1905	Male	Research Scientist	...	1	5	3
1466	Yes	Travel_Frequently	25 - 34	Ex-Employees	R&D	Life Sciences	STAFF-1868	1868	Male	Research Scientist	...	0	1	2
1467	Yes	Travel_Frequently	35 - 44	Ex-Employees	Sales	Other	STAFF-1667	1667	Male	Sales Executive	...	0	13	4
1468	Yes	Travel_Rarely	Under 25	Ex-Employees	R&D	Life Sciences	STAFF-1878	1878	Male	Research Scientist	...	0	1	3
1469	Yes	Travel_Rarely	Under 25	Ex-Employees	Sales	Life Sciences	STAFF-1702	1702	Male	Sales Representative	...	1	3	3

1470 rows × 42 columns

Grouping by 'Work Life Balance' and calculating the attrition

In [328]:
```
attrition_wl_balance = df_raw.groupby('Work Life Balance')['Attrition_int'].sum()
```

In [329]:
```
attrition_wl_balance
```

Out[329]:
```
Work Life Balance
1     25
2     58
3    127
4     27
Name: Attrition_int, dtype: int32
```

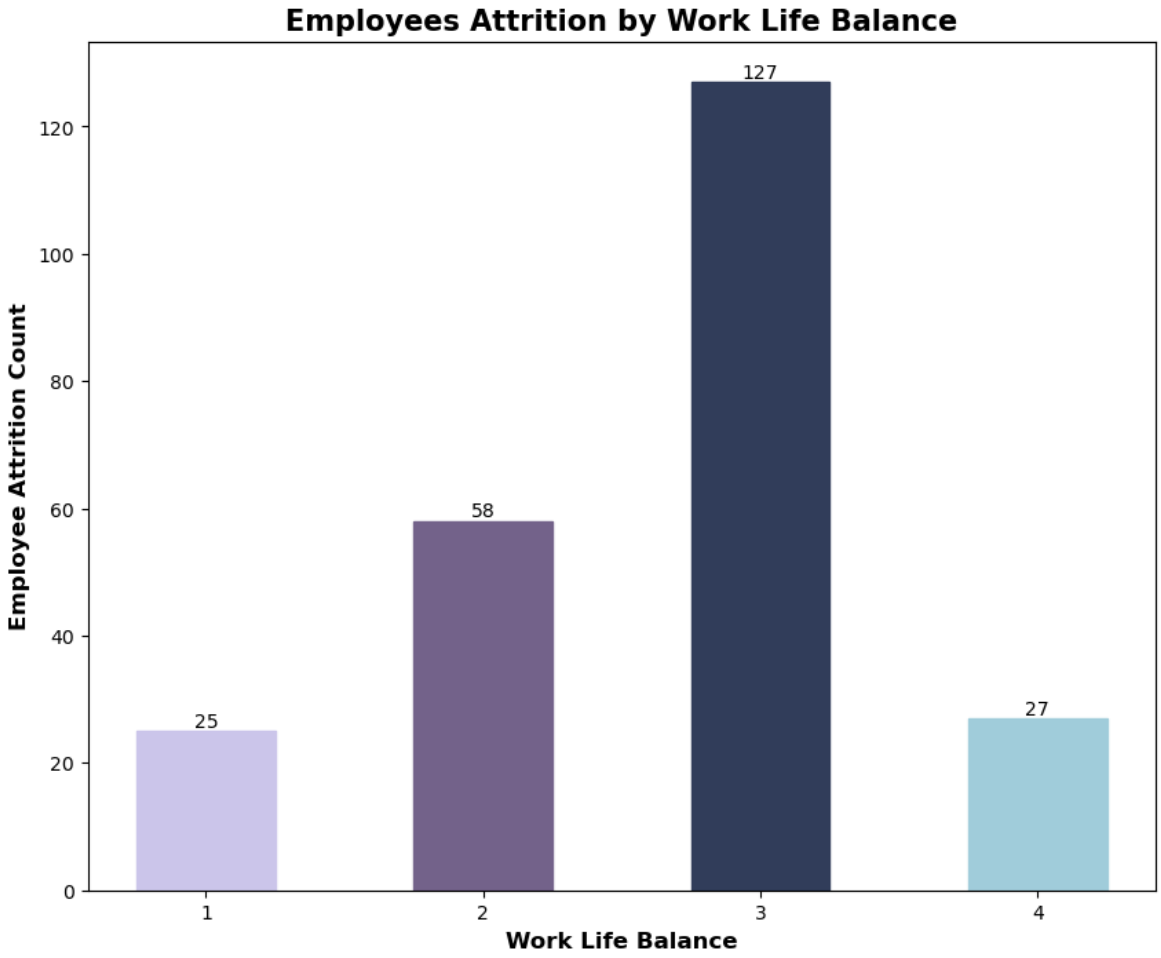
In [333]:
```
attrition_wl_balance.index
```

Out[333]:
```
Int64Index([1, 2, 3, 4], dtype='int64', name='Work Life Balance')
```

In [347]:
```
attrition_wl_balance.shape
```

Out[347]:
```
(4,)
```

In [360]:
```
col_list = ['#CBC5EA', '#73628A', '#313D5A', '#A0CCDA']
plt.figure(figsize=(10,8))
bars = plt.bar(x = attrition_wl_balance.index, height = attrition_wl_balance, width = 0.5)
for i in range(attrition_wl_balance.index.shape[0]):
    bars[i].set_color(col_list[i])
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Work Life Balance', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Work Life Balance", fontdict={'size':15, 'fontweight':'bold'})
plt.xticks(attrition_wl_balance.index)
plt.show()
```



Changing the indices to significant names

In [361]:
```
index_values = ['Low WLB', 'Medium WLB', 'Good WLB', 'Eecellent WLB']
```

In [362]:
```
index_values
```

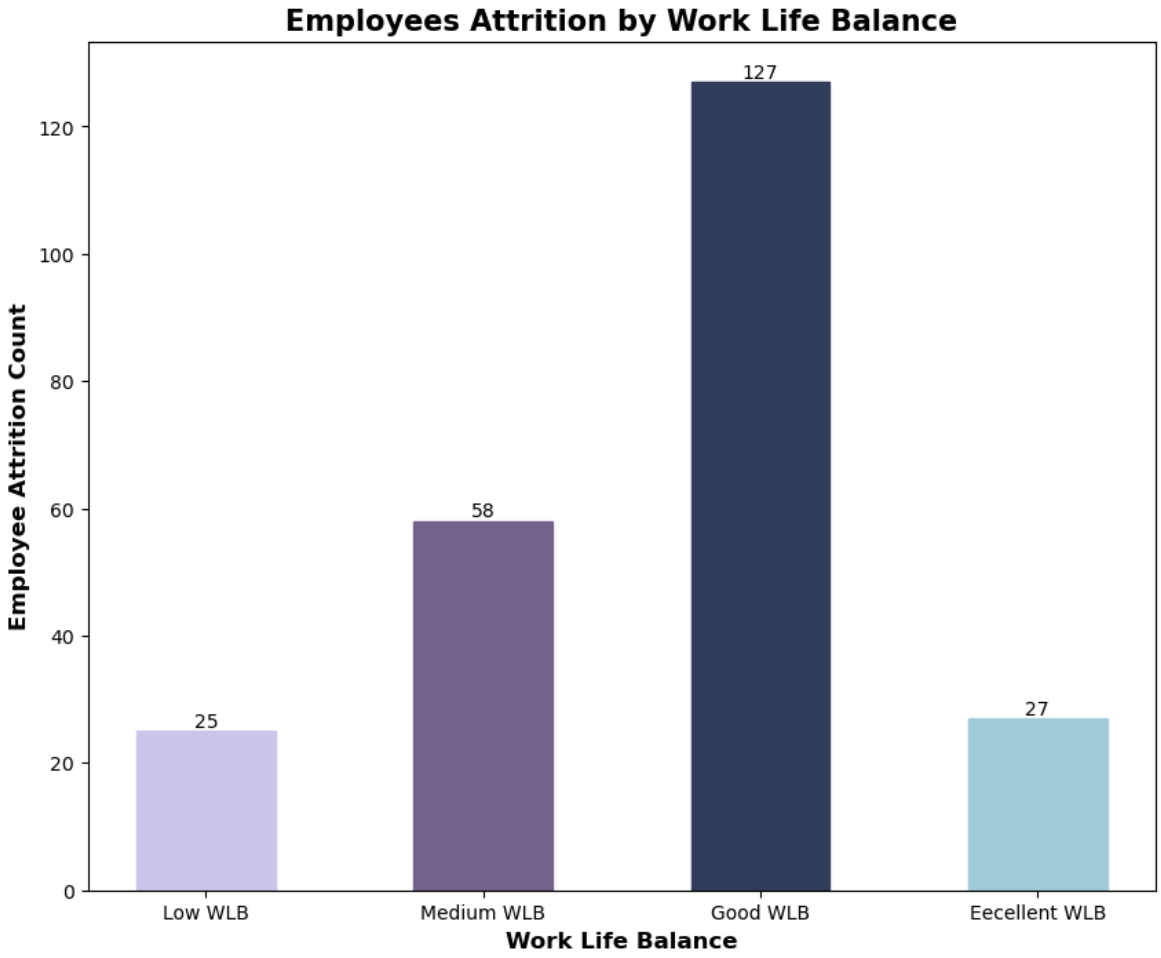
Out[362]:
```
['Low WLB', 'Medium WLB', 'Good WLB', 'Eecellent WLB']
```

In [363]:
```
attrition_wl_balance.set_axis(index_values)
```

Out[363]:
```
Low WLB           25
Medium WLB        58
Good WLB         127
Eecellent WLB     27
Name: Attrition_int, dtype: int32
```

In [364]:
```
attrition_wl_balance = attrition_wl_balance.set_axis(index_values)
```

In [373]:
```
col_list = ['#CBC5EA', '#73628A', '#313D5A', '#A0CCDA']
plt.figure(figsize=(10,8))
bars = plt.bar(x = attrition_wl_balance.index, height = attrition_wl_balance, width = 0.5)
for i in range(attrition_wl_balance.index.shape[0]):
    bars[i].set_color(col_list[i])
plt.bar_label(bars)
plt.ylabel('Employee Attrition Count',fontdict={'size':12, 'fontweight':'bold'})
plt.xlabel('Work Life Balance', fontdict={'size':12, 'fontweight':'bold'})
plt.title("Employees Attrition by Work Life Balance", fontdict={'size':15, 'fontweight':'bold'})
plt.xticks(attrition_wl_balance.index)
plt.show()
```



Conclusion: Interestingly, most attritions are of employees who experience good work life balance.