

```
import numpy as np
import pandas as pd
```

```
pd.set_option('display.max_rows', 2000)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

```
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv('Attrition data.csv')
data.head()
```

EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount
Gender	JobLevel		JobRole	MaritalStatus				
MonthlyIncome	NumCompaniesWorked	Over18	PercentSalaryHike					
StandardHours	StockOptionLevel	TotalWorkingYears						
TrainingTimesLastYear	YearsAtCompany	YearsSinceLastPromotion						
YearsWithCurrManager	EnvironmentSatisfaction	JobSatisfaction						
WorkLifeBalance	JobInvolvement	PerformanceRating						
0	1	51	No	Travel_Rarely				
Sales		6	2	Life Sciences				1
Female	1	Healthcare	Representative	Married				
131160		1.0	Y	11				8
0		1.0		6			1	
0		0		3.0				4.0
2.0		3	3					
1	2	31	Yes	Travel_Frequently	Research &			
Development		10	1	Life Sciences				1
Female	1		Research Scientist	Single				
41890		0.0	Y	23				8
1		6.0		3			5	
1		4		3.0				2.0
4.0		2	4					
2	3	32	No	Travel_Frequently	Research &			
Development		17	4	Other				1
Male	4		Sales Executive	Married				193280
1.0	Y	15		8			3	
5.0		2		5				0
3		2.0		2.0			1.0	
3		3						
3	4	38	No	Non-Travel	Research &			
Development		2	5	Life Sciences				1
Male	3		Human Resources	Married				83210
3.0	Y	11		8			3	
13.0		5		8				7
5		4.0		4.0			3.0	

```

2
4      5      32      No      Travel_Rarely      Research &
Development      10      1      Medical      1
Male      1      Sales Executive      Single      23420
4.0      Y      12      8      2
9.0      2      6      0
4      4.0      1.0      3.0
3      3

```

```
data.shape
```

```
(4410, 29)
```

```
data.Attrition.value_counts()
```

```

Attrition
No      3699
Yes     711
Name: count, dtype: int64

```

```
data.describe()
```

```

      EmployeeID      Age  DistanceFromHome  Education
EmployeeCount  JobLevel  MonthlyIncome  NumCompaniesWorked
PercentSalaryHike  StandardHours  StockOptionLevel  TotalWorkingYears
TrainingTimesLastYear  YearsAtCompany  YearsSinceLastPromotion
YearsWithCurrManager  EnvironmentSatisfaction  JobSatisfaction
WorkLifeBalance  JobInvolvement  PerformanceRating
count  4410.000000  4410.000000      4410.000000  4410.000000
4410.0  4410.000000      4410.000000      4391.000000
4410.000000      4410.0      4410.000000      4401.000000
4410.000000      4410.000000      4410.000000
4410.000000      4385.000000      4390.000000      4372.000000
4410.000000      4410.000000
mean    2205.500000      36.923810      9.192517      2.912925
1.0      2.063946      65029.312925      2.694830      15.209524
8.0      0.793878      11.279936      2.799320
7.008163      2.187755      4.123129
2.723603      2.728246      2.761436      2.729932
3.153741
std     1273.201673      9.133301      8.105026      1.023933
0.0      1.106689      47068.888559      2.498887      3.659108
0.0      0.851883      7.782222      1.288978
6.125135      3.221699      3.567327
1.092756      1.101253      0.706245      0.711400
0.360742
min      1.000000      18.000000      1.000000      1.000000
1.0      1.000000      10090.000000      0.000000      11.000000
8.0      0.000000      0.000000      0.000000      0.000000
0.000000      0.000000      0.000000
1.000000      1.000000      1.000000      1.000000

```

3.000000				
25%	1103.250000	30.000000	2.000000	2.000000
1.0	1.000000	29110.000000	1.000000	12.000000
8.0	0.000000	6.000000	2.000000	
3.000000		0.000000	2.000000	
2.000000	2.000000	2.000000	2.000000	
3.000000				
50%	2205.500000	36.000000	7.000000	3.000000
1.0	2.000000	49190.000000	2.000000	14.000000
8.0	1.000000	10.000000	3.000000	
5.000000		1.000000	3.000000	
3.000000	3.000000	3.000000	3.000000	
3.000000				
75%	3307.750000	43.000000	14.000000	4.000000
1.0	3.000000	83800.000000	4.000000	18.000000
8.0	1.000000	15.000000	3.000000	
9.000000		3.000000	7.000000	
4.000000	4.000000	3.000000	3.000000	
3.000000				
max	4410.000000	60.000000	29.000000	5.000000
1.0	5.000000	199990.000000	9.000000	25.000000
8.0	3.000000	40.000000	6.000000	
40.000000		15.000000	17.000000	
4.000000	4.000000	4.000000	4.000000	
4.000000				

DISTRIBUTION OF EMPLOYEE ATTRITION IN THE COMPANY

```

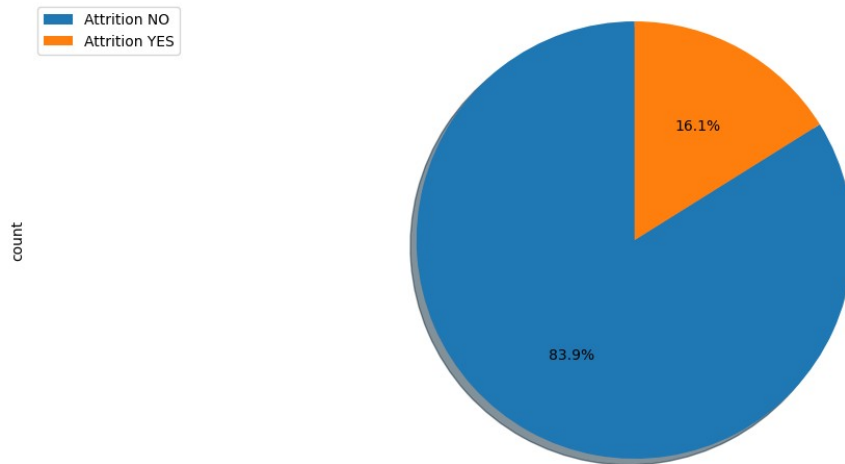
labels = 'Attrition NO', 'Attrition YES'
data['Attrition'].astype(str).value_counts().plot(kind='pie',
                                                    figsize=(15, 6),
                                                    autopct='%1.1f%%',
                                                    startangle=90,
                                                    shadow=True,
                                                    labels=None,

                                                    )

plt.title('Distribution of Employee Attrition in the Company ',
y=1.12)
plt.axis('equal')
# add legend
plt.legend(labels=labels, loc='upper left')
# show plot
plt.show()

```

Distribution of Employee Attrition in the Company



Age distribution of Employees

```
plt.figure(figsize=(10,8))
data['Age'].hist(bins=70)
plt.title("Age distribution of Employees")
plt.xlabel("Age")
plt.ylabel("# of Employees")
plt.show()
```



JOB SATISFACTION

```
data['JobSatisfaction'].value_counts()
```

```
JobSatisfaction
```

```
4.0    1367
```

```
3.0    1323
```

```
1.0     860
```

```
2.0     840
```

```
Name: count, dtype: int64
```

WORK LIFE BALANCE

```
fig2 = plt.figure()
```

```
ax5 = fig2.add_subplot(121)
```

```

ax6 = fig2.add_subplot(122)

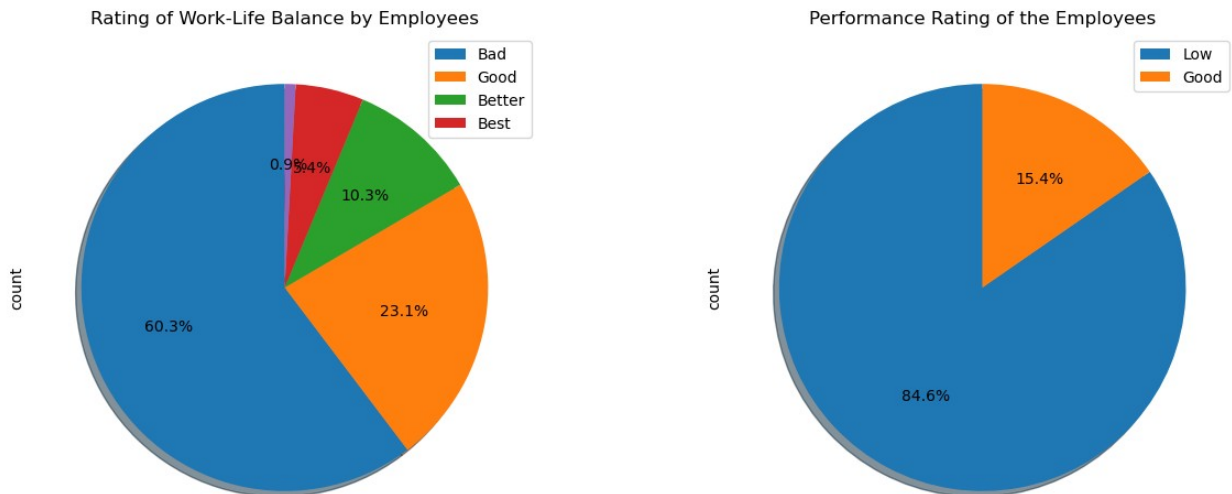
labels_list1 = 'Bad', 'Good', 'Better', 'Best'
labels_list2 = 'Low', 'Good', 'Excellent', 'Outstanding'

data['WorkLifeBalance'].astype(str).value_counts().plot(kind='pie',
                                                         figsize=(15, 6),
                                                         autopct='%1.1f%%',
                                                         startangle=90,
                                                         shadow=True,
                                                         labels=None, ax=ax5) # add to subplot 2
ax5.set_title('Rating of Work-Life Balance by Employees')
ax5.legend(labels=labels_list1, loc='upper right')

data['PerformanceRating'].astype(str).value_counts().plot(kind='pie',
                                                         figsize=(15, 6),
                                                         autopct='%1.1f%%',
                                                         startangle=90,
                                                         shadow=True,
                                                         labels=None, ax=ax6)
ax6.set_title('Performance Rating of the Employees')
ax6.legend(labels=labels_list2, loc='upper right')

plt.show()

```



BUSINESS TRAVEL VS ATTRITION

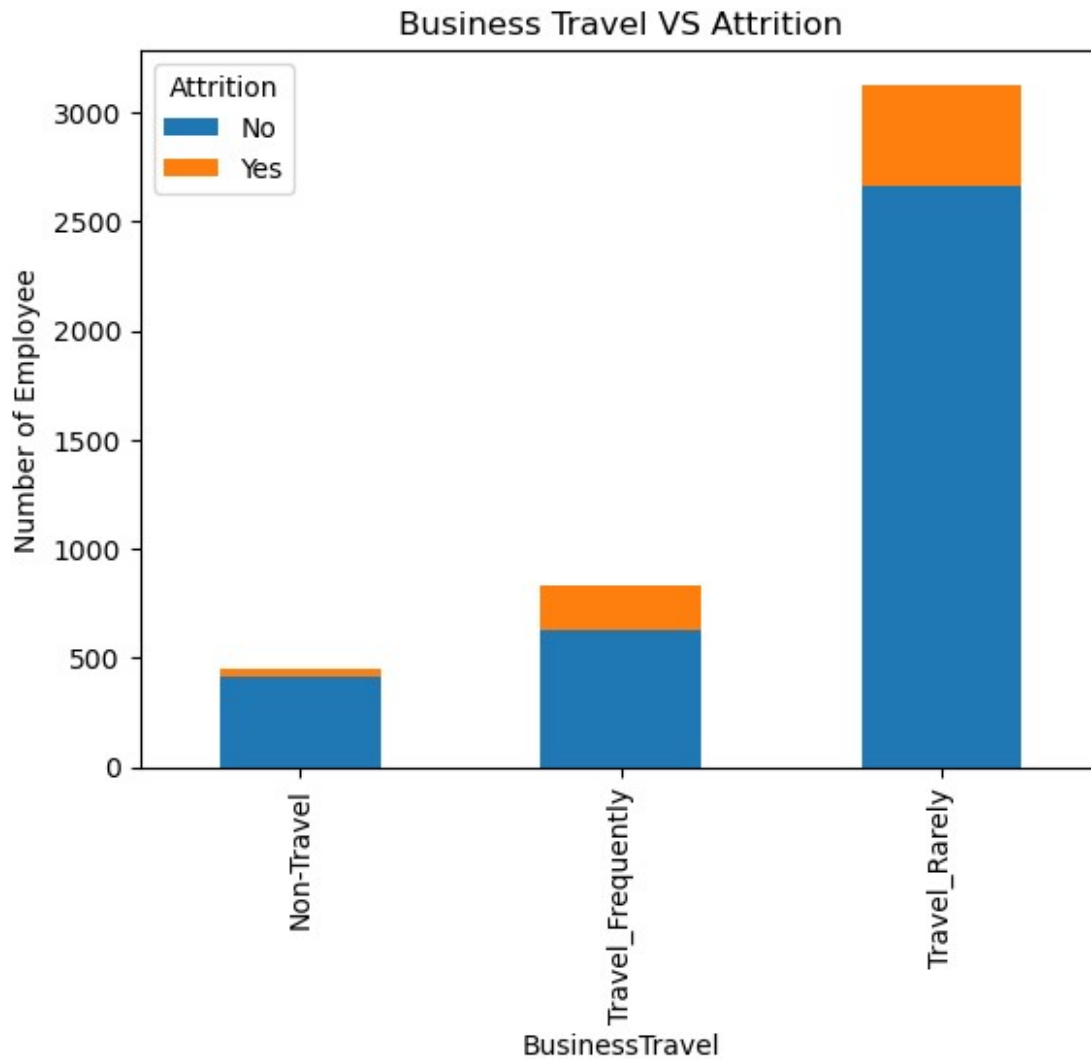
```

props = data.groupby("BusinessTravel")
['Attrition'].value_counts(normalize=False).unstack()

props.plot(kind='bar', alpha=1, stacked=False)

```

```
plt.title('Business Travel VS Attrition')
plt.ylabel('Number of Employee')
plt.show()
```

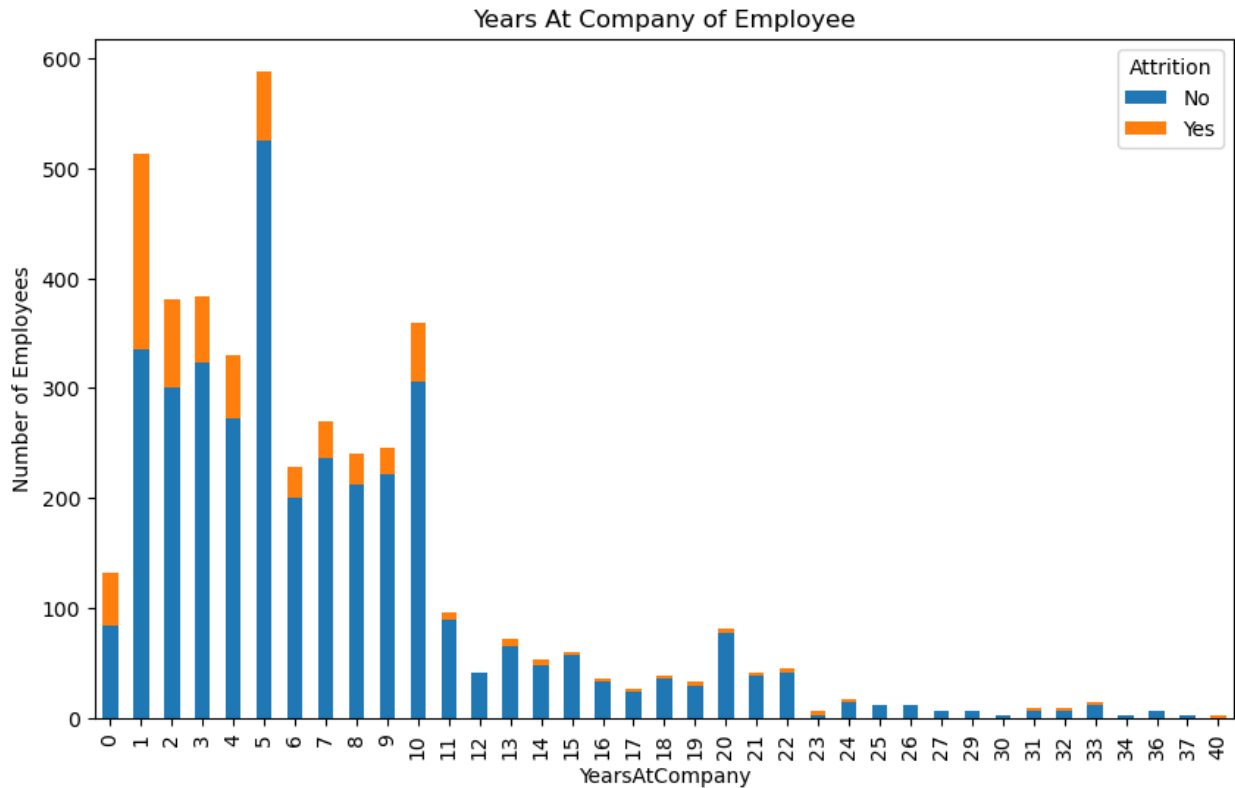


YEARS AT COMPANY OF EMPLOYEE

```
yac = data.groupby("YearsAtCompany")
['Attrition'].value_counts(normalize=False).unstack()

yac.plot(kind='bar', stacked=False, figsize=(10,6))

plt.title('Years At Company of Employee')
plt.ylabel('Number of Employees')
plt.show()
```

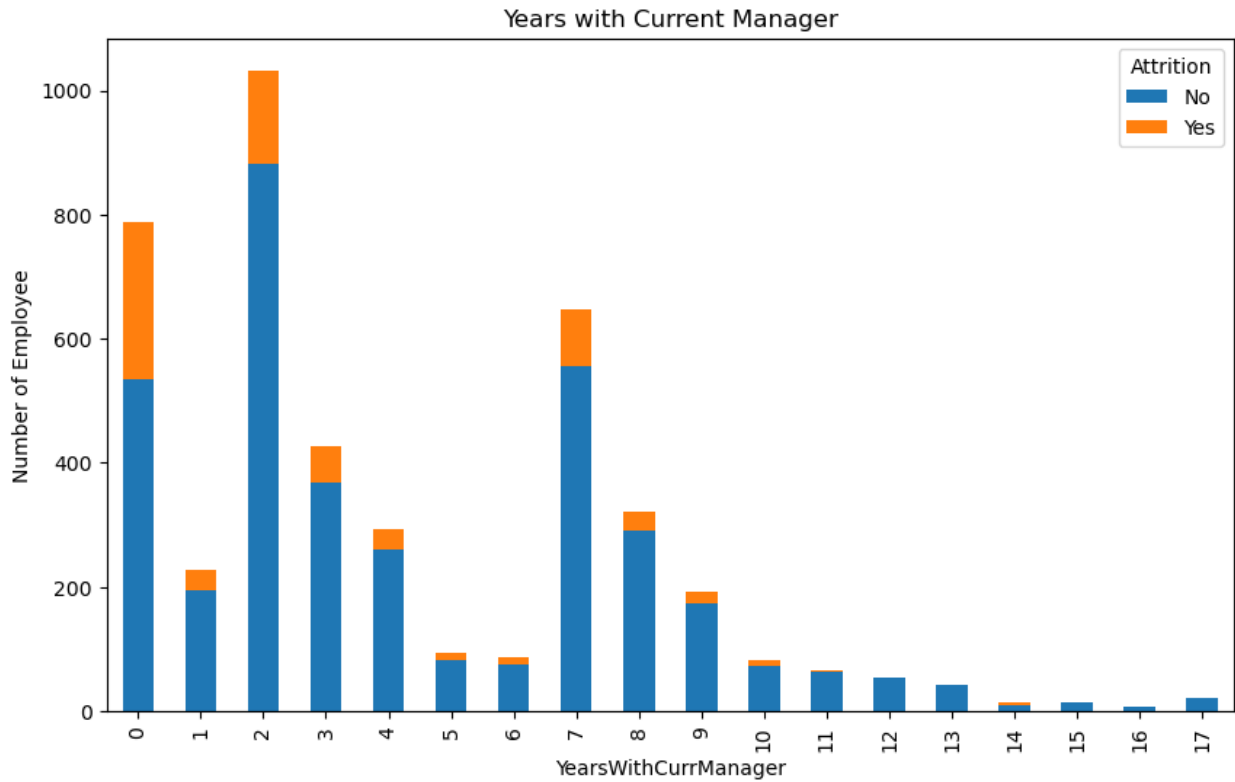


YEARS WITH CURRENT MANAGER

```
ycm = data.groupby("YearsWithCurrManager")
['Attrition'].value_counts(normalize=False).unstack()

ycm.plot(kind='bar', stacked=False, figsize=(10,6))

plt.title('Years with Current Manager')
plt.ylabel('Number of Employee')
plt.show()
```

TOTAL WORKING EXPERIENCE

```
twy = data.groupby("TotalWorkingYears")
['Attrition'].value_counts(normalize=False).unstack()

twy.plot(kind='bar', stacked=False, figsize=(8,5))

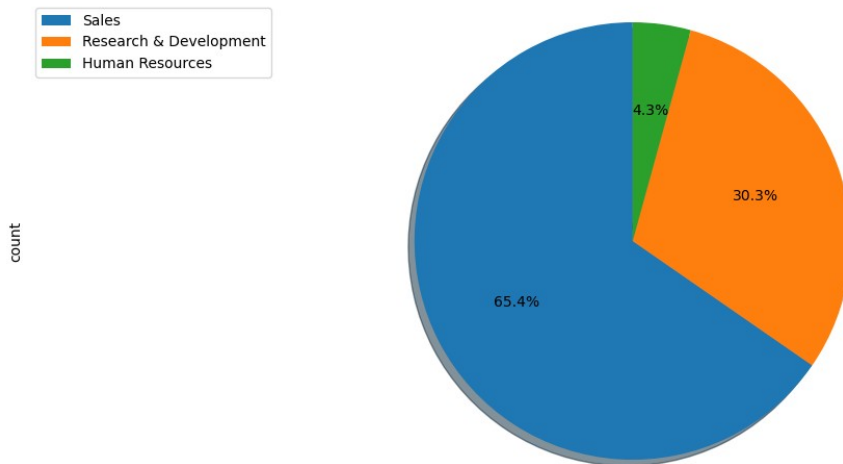
plt.title('Total Working Years of Experience')
plt.ylabel('Number of Employee')
plt.show()
```



```

startangle=90,
shadow=True,
labels=None)
plt.axis('equal')
plt.legend(labels=dpt['Department'].unique(), loc='upper left')
<matplotlib.legend.Legend at 0x19f50a31c50>

```



```

dpm = data.groupby("Department")
['Attrition'].value_counts(normalize=False).unstack()
dpm = dpm.transpose()
dpm

```

Department	Human Resources	Research & Development	Sales
Attrition			
No	132	2430	1137
Yes	57	453	201

```

labels = ['Human Resources', 'Research & Development', 'Sales',]
sizes = [63, 961, 446]
labels_attrition = ['Yes', 'No', 'Yes', 'No', 'Yes', 'No']
sizes_attrition = [12, 51, 133, 828, 92, 354]
colors = ['#ff6666', '#ffcc99', '#99ff99']

```

```

colors_attrition = ['#0a0e77', '#9e0723', '#0a0e77', '#9e0723',
'#0a0e77', '#9e0723', '#0a0e77', '#9e0723']

```

```
# Plot
```

```

plt.pie(sizes, autopct='%1.1f%%', pctdistance=.87, labels=labels,
colors=colors, startangle=90, frame=True)
plt.pie(sizes_attrition, colors=colors_attrition, radius=0.75, startangle=90)
centre_circle = plt.Circle((0,0),0.5,color='black',
fc='white',linewidth=0.5)

```

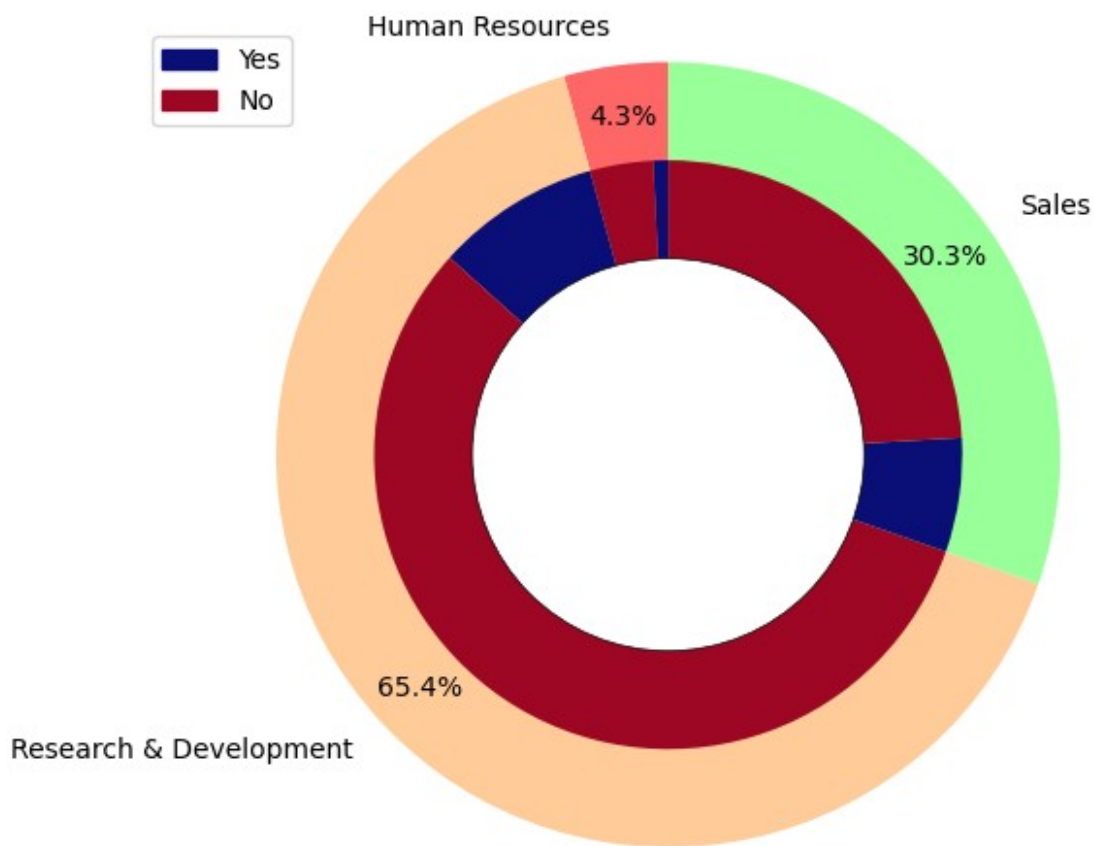
```

fig6 = plt.gcf()
fig6.gca().add_artist(centre_circle)

#legend
import matplotlib.patches as mpatches
pur = mpatches.Patch(color='#0a0e77', label='Yes')
pin = mpatches.Patch(color='#9e0723', label='No')
plt.legend(handles=[pur, pin], loc='upper left')

plt.axis('equal')
plt.tight_layout()
plt.show()

```



GENDER AND DISTANCE FROM HOME VS ATTRITION

```

gda = data[['Gender', 'DistanceFromHome', 'Attrition']]
gda.head()

```

	Gender	DistanceFromHome	Attrition
0	Female	6	No
1	Female	10	Yes
2	Male	17	No
3	Male	2	No
4	Male	10	No

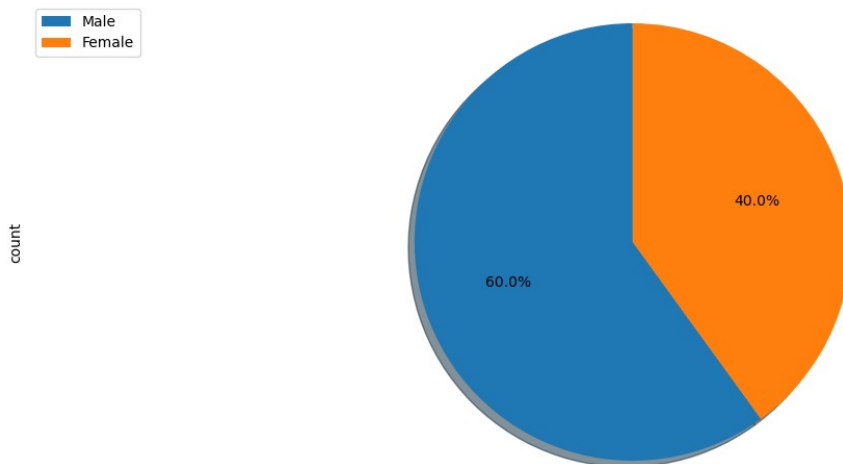
```
gda['Gender'].value_counts()
```

```
Gender
Male      2646
Female    1764
Name: count, dtype: int64
```

```
gda['Gender'].value_counts().plot(kind='pie',
                                  figsize=(15, 6),
                                  autopct='%1.1f%%',
                                  startangle=90,
                                  shadow=True,
                                  labels=None)

plt.axis('equal')
plt.legend(labels=['Male', 'Female'], loc='upper left')

<matplotlib.legend.Legend at 0x19f4f23a990>
```



```
fma = gda.groupby("Gender")
['Attrition'].value_counts(normalize=False).unstack()
fma = fma.transpose()
fma
```

Gender	Female	Male
Attrition		
No	1494	2205
Yes	270	441

```

labels = ['Male', 'Female']
sizes = [882,588]
labels_attrition = ['Yes','No','Yes','No']
sizes_attrition = [150,732,87,501]
colors = ['#ff6666', '#ffcc99']

colors_attrition = ['#c2c2f0', '#ffb3e6', '#c2c2f0', '#ffb3e6']

# Plot
plt.pie(sizes, labels=labels, colors=colors, startangle=90, frame=True)
plt.pie(sizes_attrition, colors=colors_attrition, radius=0.75, startangle=90)
centre_circle = plt.Circle((0,0),0.5,color='black',
fc='white',linewidth=0.5)
fig6 = plt.gcf()
fig6.gca().add_artist(centre_circle)

#legend
import matplotlib.patches as mpatches
pur = mpatches.Patch(color='#c2c2f0', label='Yes')
pin = mpatches.Patch(color='#ffb3e6', label='No')
plt.legend(handles=[pur, pin], loc='center')

plt.axis('equal')
plt.tight_layout()
plt.show()

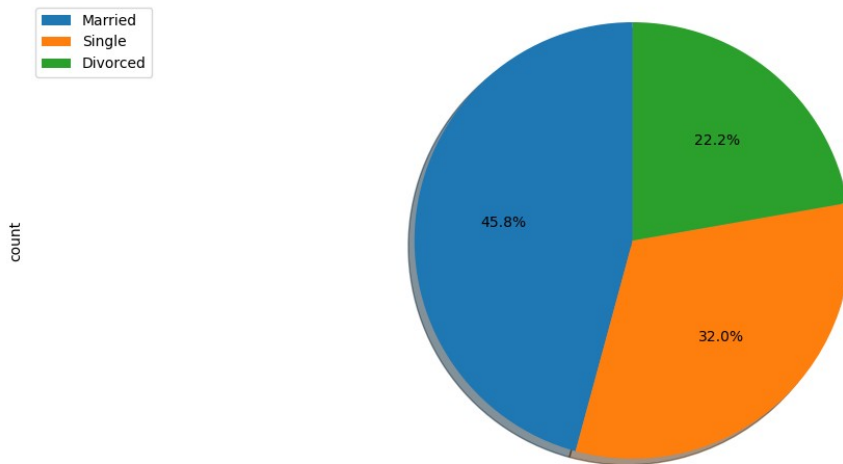
```



```

        autopct='%1.1f%%',
        startangle=90,
        shadow=True,
        labels=None)
plt.axis('equal')
plt.legend(labels=['Married', 'Single', 'Divorced'], loc='upper left')
<matplotlib.legend.Legend at 0x19f50a6e410>

```



```

msa = ms.groupby("MaritalStatus")
['Attrition'].value_counts(normalize=False).unstack()
msa = msa.transpose()
msa

MaritalStatus  Divorced  Married  Single
Attrition
No             882      1767     1050
Yes            99       252      360

labels = ['Married', 'Single', 'Divorced']
sizes = [673, 470, 327]
labels_attrition = ['Yes', 'No', 'Yes', 'No', 'Yes', 'No']
sizes_attrition = [84, 589, 120, 350, 33, 294]
colors = ['#ff6666', '#ffcc99', '#99ff99']

colors_attrition = ['#c2c2f0', '#ffb3e6', '#c2c2f0', '#ffb3e6',
                    '#c2c2f0', '#ffb3e6']

# Plot
plt.pie(sizes, labels=labels, colors=colors, startangle=90, frame=True)
plt.pie(sizes_attrition, colors=colors_attrition, radius=0.75, startangle=90)
centre_circle = plt.Circle((0,0),0.5,color='black',
fc='white',linewidth=0.5)

```



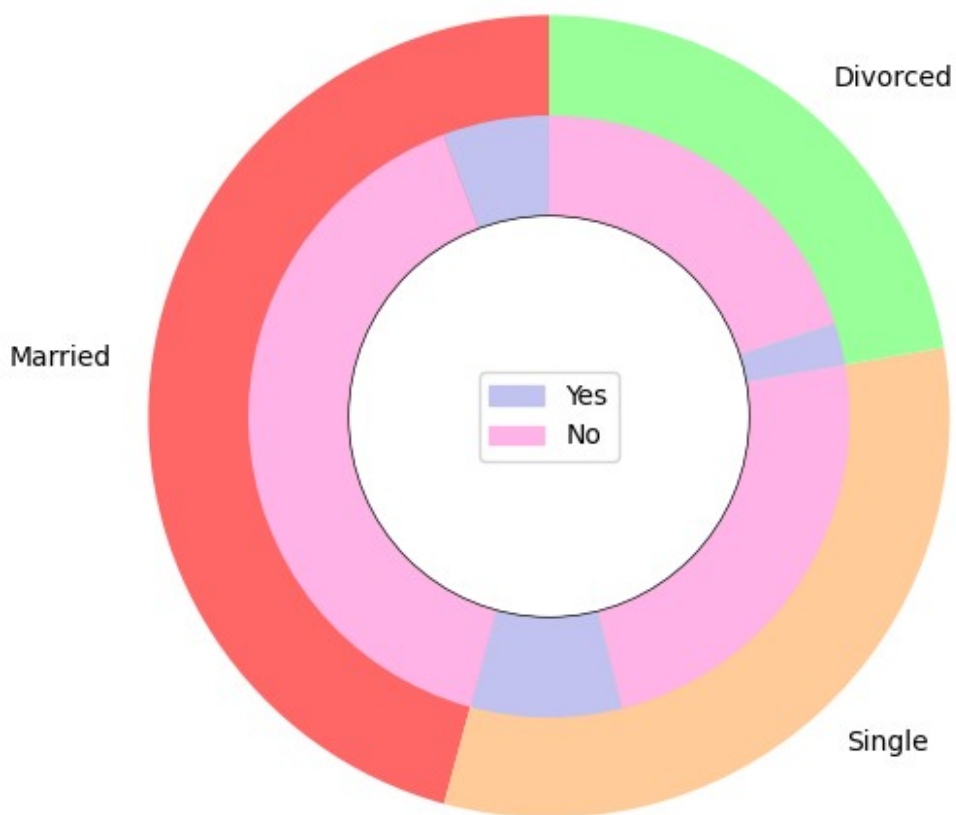
```

fig6 = plt.gcf()
fig6.gca().add_artist(centre_circle)

#legend
import matplotlib.patches as mpatches
pur = mpatches.Patch(color='#c2c2f0', label='Yes')
pin = mpatches.Patch(color='#ffb3e6', label='No')
plt.legend(handles=[pur, pin], loc='center')

plt.axis('equal')
plt.tight_layout()
plt.show()

```



EDUCATION FIELD DISTRIBUTION

```

plt.figure(figsize=(10,8))
data.EducationField.value_counts().plot(kind='barh',color='g',alpha=.65)
plt.title("Education Field Distribution")
plt.show()

```

