Descriptive Analysis of Attrition Within a Business

In [92]:

```
## import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

## check versions
print('pandas version:', pd.__version__)
print('numpy version:', np.__version__)
```

```
pandas version: 1.4.2
numpy version: 1.21.5
```

Load Files into Database

In [89]:

```
import sqlite3
conn = sqlite3.connect('DSC 540')
c = conn.cursor()
```

In [29]:

```
## Load in flat file
df1 = pd.read_csv('DSC540_DF_Milestone_2.csv')
df1
```

Out[29]:

	Unnamed: 0	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromH
0	0	1	51	No	Travel_Rarely	Sales	
1	1	2	31	Yes	Travel_Frequently	Research & Development	
2	2	3	32	No	Travel_Frequently	Research & Development	
3	3	4	38	No	Non-Travel	Research & Development	
4	4	5	32	No	Travel_Rarely	Research & Development	
...	...	...	...	...	...	...	
4405	4405	4406	42	No	Travel_Rarely	Research & Development	
4406	4406	4407	29	No	Travel_Rarely	Research & Development	
4407	4407	4408	25	No	Travel_Rarely	Research & Development	
4408	4408	4409	42	No	Travel_Rarely	Sales	
4409	4409	4410	40	No	Travel_Rarely	Research & Development	

4410 rows × 29 columns

In [30]:

```python
df1 = df1.drop('Unnamed: 0', axis=1)
df1
```

Out[30]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Educ
0	1	51	No	Travel_Rarely	Sales	6	
1	2	31	Yes	Travel_Frequently	Research & Development	10	
2	3	32	No	Travel_Frequently	Research & Development	17	
3	4	38	No	Non-Travel	Research & Development	2	
4	5	32	No	Travel_Rarely	Research & Development	10	
...	...	...	...	...	...	...	
4405	4406	42	No	Travel_Rarely	Research & Development	5	
4406	4407	29	No	Travel_Rarely	Research & Development	2	
4407	4408	25	No	Travel_Rarely	Research & Development	25	
4408	4409	42	No	Travel_Rarely	Sales	18	
4409	4410	40	No	Travel_Rarely	Research & Development	28	

4410 rows × 28 columns

In [31]:

```
## load in website file
df2 = pd.read_excel('milestone_3.xlsx')
df2
```

Out[31]:

	Unnamed: 0	Rank	Occupation	EducationField	# of Jobs	Median Salary	Unemployment Rate	Educa
0	0	1	Dentist	Medical	27600	142750	0.007	
1	1	2	Registered Nurse	Medical	712900	65790	0.020	
2	2	3	Pharmacist	Medical	69740	113410	0.032	
3	3	4	Computer Systems Analyst	Technology	120440	78670	0.025	Te
4	4	5	Physician	Medical	168330	183270	0.007	
...	...	...	...	...	...	...	...	
95	95	96	Carpenter	Technical Degree	196200	40210	0.160	
96	96	97	Security Guard	Other	195300	23930	0.113	
97	97	98	Construction Worker	Technical Degree	212500	29450	0.212	
98	98	99	Fabricator	Technical Degree	12500	35570	0.143	
99	99	100	Telemarketer	Other	21500	23570	0.313	

100 rows × 8 columns

In [32]:
```python
df2 = df2.drop('Unnamed: 0', axis=1)
df2
```

Out[32]:

	Rank	Occupation	EducationField	# of Jobs	Median Salary	Unemployment Rate	EducationField2
0	1	Dentist	Medical	27600	142750	0.007	Medical
1	2	Registered Nurse	Medical	712900	65790	0.020	Medical
2	3	Pharmacist	Medical	69740	113410	0.032	Medical
3	4	Computer Systems Analyst	Technology	120440	78670	0.025	Technology
4	5	Physician	Medical	168330	183270	0.007	Medical
...	...	...	...	...	...	...	...
95	96	Carpenter	Technical Degree	196200	40210	0.160	Technical Degree
96	97	Security Guard	Other	195300	23930	0.113	Other
97	98	Construction Worker	Technical Degree	212500	29450	0.212	Technical Degree
98	99	Fabricator	Technical Degree	12500	35570	0.143	Technical Degree
99	100	Telemarketer	Other	21500	23570	0.313	Other

100 rows × 7 columns

Realizing similar to milestone 4, df2 will not join successfully to df1 and needs to be aggregated in order to join row to row.

In [49]:
```python
df2['EducationField'].value_counts()
```

Out[49]:
```
Medical                        29
Technical Degree               20
Other                          11
Accounting and Finance          8
Arts and Education              8
Technology                      7
Marketing                       6
Human Resources                 5
Engineering                     4
Legal and Public Policies       2
Name: EducationField, dtype: int64
```

In [56]:

```
df2.pivot_table(index='EducationField', values='Occupation', aggfunc='
```

Out[56]:

	Occupation
EducationField	
Accounting and Finance	8
Arts and Education	8
Engineering	4
Human Resources	5
Legal and Public Policies	2
Marketing	6
Medical	29
Other	11
Technical Degree	20
Technology	7

In [57]:

```
df2.pivot_table(index='EducationField', values='# of Jobs', aggfunc='s
```

Out[57]:

	# of Jobs
EducationField	
Accounting and Finance	843570
Arts and Education	865870
Engineering	172440
Human Resources	795520
Legal and Public Policies	120560
Marketing	801380
Medical	3700950
Other	1166770
Technical Degree	2080580
Technology	559340

In [59]:
```
df2.pivot_table(index='EducationField', values='Median Salary', aggfun
```

Out[59]:

EducationField	Median Salary
Accounting and Finance	69481.250000
Arts and Education	46900.000000
Engineering	69010.000000
Human Resources	37752.000000
Legal and Public Policies	80030.000000
Marketing	69501.666667
Medical	58736.206897
Other	39120.000000
Technical Degree	39850.500000
Technology	83212.857143

In [64]:
```
Data = {'Education_Field': ['Accounting and Finance', 'Human Resources
        'Occupation': [8, 5, 6, 29],
        'Num_of_Jobs': [843570, 795520, 801380, 3700950],
        'Median_Salary': [69481, 37752, 69501, 58736]}

df2_final = pd.DataFrame(Data)

df2_final
```

Out[64]:

	Education_Field	Occupation	Num_of_Jobs	Median_Salary
0	Accounting and Finance	8	843570	69481
1	Human Resources	5	795520	37752
2	Marketing	6	801380	69501
3	Medical	29	3700950	58736

In [33]:
```python
## Load in API fil
df3 = pd.read_excel('milestone_4.xlsx')
df3
```

Out[33]:

	Unnamed: 0	Department	Job_Listings	Applications	Average_minimumSalary	Average_m
0	0	Sales	55	1794	78490	
1	1	Human Resources	10	185	38260	
2	2	Healthcare	5	5	53907	
3	3	Accounting and Finance	3	15	34333	
4	4	Other	1	4	100000	

In [34]:
```python
df3 = df3.drop('Unnamed: 0', axis=1)
df3
```

Out[34]:

	Department	Job_Listings	Applications	Average_minimumSalary	Average_maximumSala
0	Sales	55	1794	78490	1185:
1	Human Resources	10	185	38260	473(
2	Healthcare	5	5	53907	621:
3	Accounting and Finance	3	15	34333	1450(
4	Other	1	4	100000	1200(

In [35]:
```python
## write df1 to SQL
df1.to_sql('df1', conn, if_exists='append', index = False)
```

Out[35]: 4410

In [71]:
```python
## write df2 to SQL
df2_final.to_sql('df2_final', conn, if_exists='append', index = False)
```

Out[71]: 4

In [37]:
```python
## write df3 to SQL
df3.to_sql('df3', conn, if_exists='append', index = False)
```

Out[37]: 5

In [38]:

```python
conn.commit()
```

In [73]:

```python
## join df1 to df2
#Retrieving data
c.execute('''SELECT * FROM df1 LEFT JOIN df2_final ON df1.EducationFie

df_merge = pd.DataFrame(c.fetchall())
df_merge.columns = [x[0] for x in c.description]
df_merge
```

Out[73]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome
0	1	51	No	Travel_Rarely	Sales	6
1	2	31	Yes	Travel_Frequently	Research & Development	10
2	3	32	No	Travel_Frequently	Research & Development	17
3	4	38	No	Non-Travel	Research & Development	2
4	5	32	No	Travel_Rarely	Research & Development	10
...	...	...	...	...	...	...
12715	4408	25	No	Travel_Rarely	Research & Development	25

In [81]:

```python
## write df_merge to SQL
df_merge.to_sql('df_merge', conn, if_exists='append', index = False)
```

Out[81]: 12720

In [80]:
```python
## join merge to df3
c.execute('''SELECT * FROM df_merge LEFT JOIN df3 ON df_merge.Departme

df_merge2 = pd.DataFrame(c.fetchall())
df_merge2.columns = [x[0] for x in c.description]
df_merge2
```

Out[80]:

	EmployeeID	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Edu
0	1	51	No	Travel_Rarely	Sales	6	
1	1	51	No	Travel_Rarely	Sales	6	
2	2	31	Yes	Travel_Frequently	Research & Development	10	
3	3	32	No	Travel_Frequently	Research & Development	17	
4	4	38	No	Non-Travel	Research & Development	2	
...	...	...	...	...	...	...	
17491	4409	42	No	Travel_Rarely	Sales	18	
17492	4409	42	No	Travel_Rarely	Sales	18	
17493	4409	42	No	Travel_Rarely	Sales	18	
17494	4410	40	No	Travel_Rarely	Research & Development	28	
17495	4410	40	No	Travel_Rarely	Research & Development	28	

17496 rows × 37 columns

In [87]:
```python
df_merge2 = df_merge2.loc[:, ~df_merge2.columns.duplicated()]
```

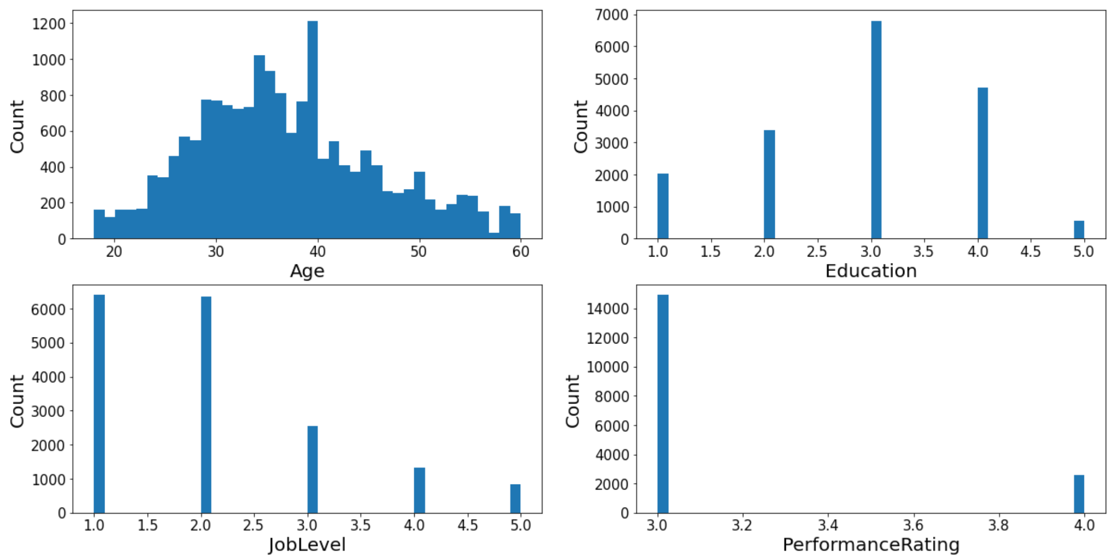
In [88]:
```python
## write df_merge2 to SQL
df_merge2.to_sql('df_merge2', conn, if_exists='append', index = False)
```

Out[88]: 17496

In [111]:
```python
conn.close()
```

Visualizations

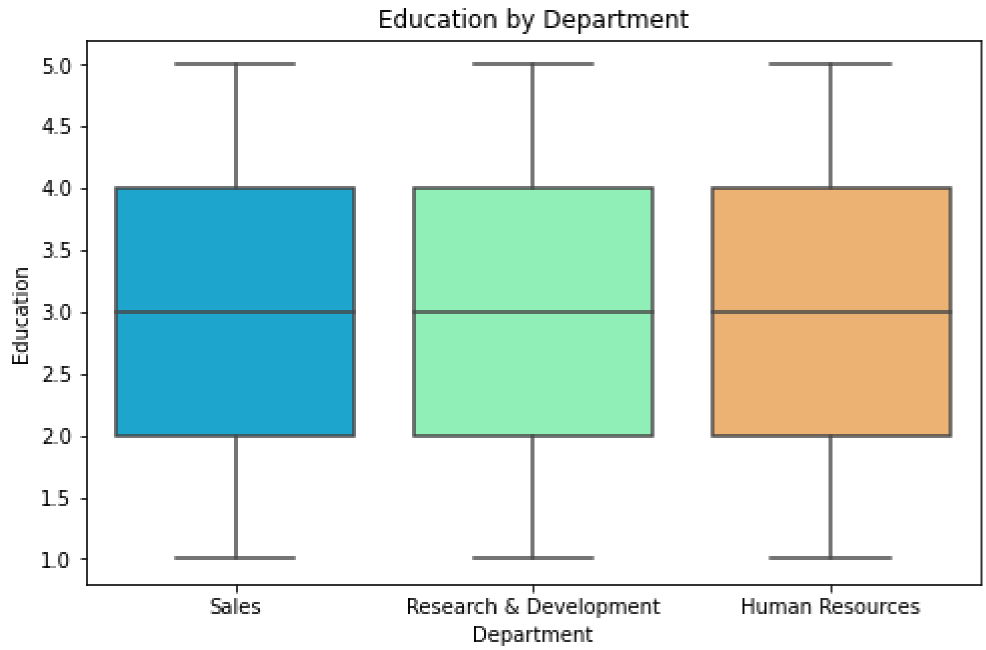
In [91]:
```
## histograms of numerical features
plt.rcParams['figure.figsize'] = (20, 10)
fig, axes = plt.subplots(nrows = 2, ncols = 2)
num_features = ['Age', 'Education', 'JobLevel', 'PerformanceRating']
xaxes = num_features
yaxes = ['Count', 'Count', 'Count', 'Count']

## histogram
axes = axes.ravel()
for idx, ax in enumerate(axes):
    ax.hist(df_merge2[num_features[idx]].dropna(), bins=40)
    ax.set_xlabel(xaxes[idx], fontsize=20)
    ax.set_ylabel(yaxes[idx], fontsize=20)
    ax.tick_params(axis='both', labelsize=15)
plt.show()
```



In [95]:

```
plt.figure(figsize=(8,5))
sns.boxplot(x='Department',y='Education',data=df_merge2, palette='rain
plt.title("Education by Department")
```

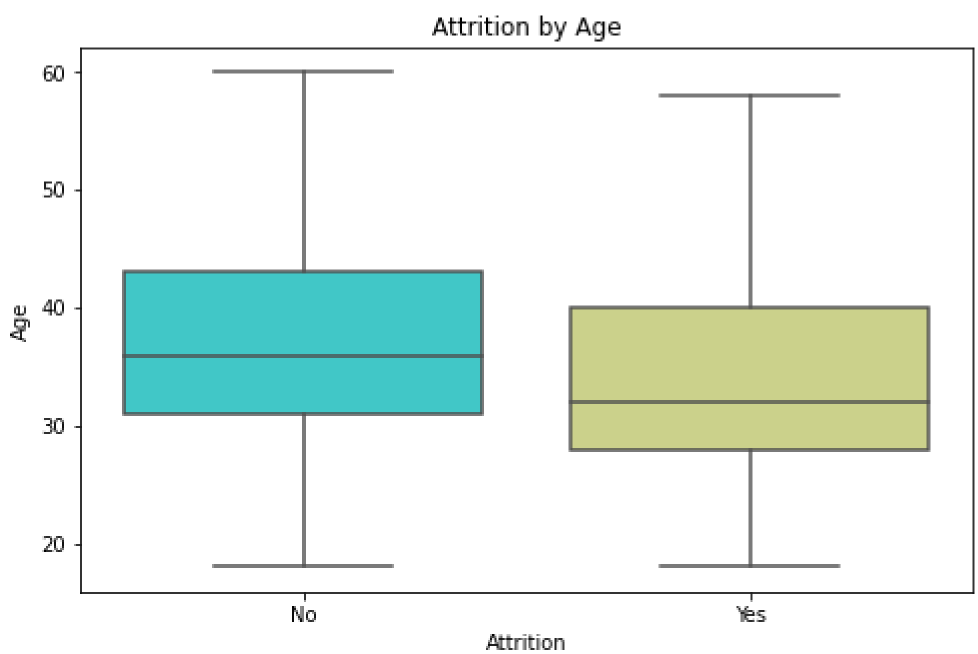
Out[95]: Text(0.5, 1.0, 'Education by Department')



In [96]:

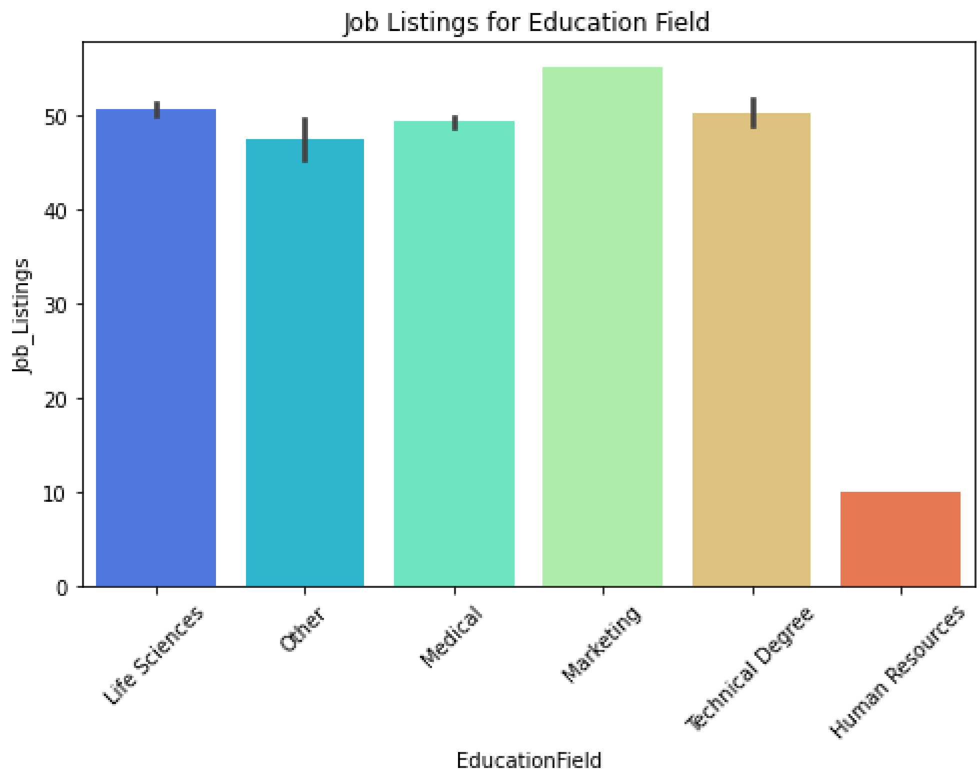
```
plt.figure(figsize=(8,5))
sns.boxplot(x='Attrition',y='Age',data=df_merge2, palette='rainbow')
plt.title("Attrition by Age")
```

Out[96]: Text(0.5, 1.0, 'Attrition by Age')



In [99]:
```python
plt.figure(figsize=(8,5))
sns.barplot(x='EducationField',y='Job_Listings',data=df_merge2, palett
plt.title("Job Listings for Education Field")
plt.xticks(rotation = 45)
```

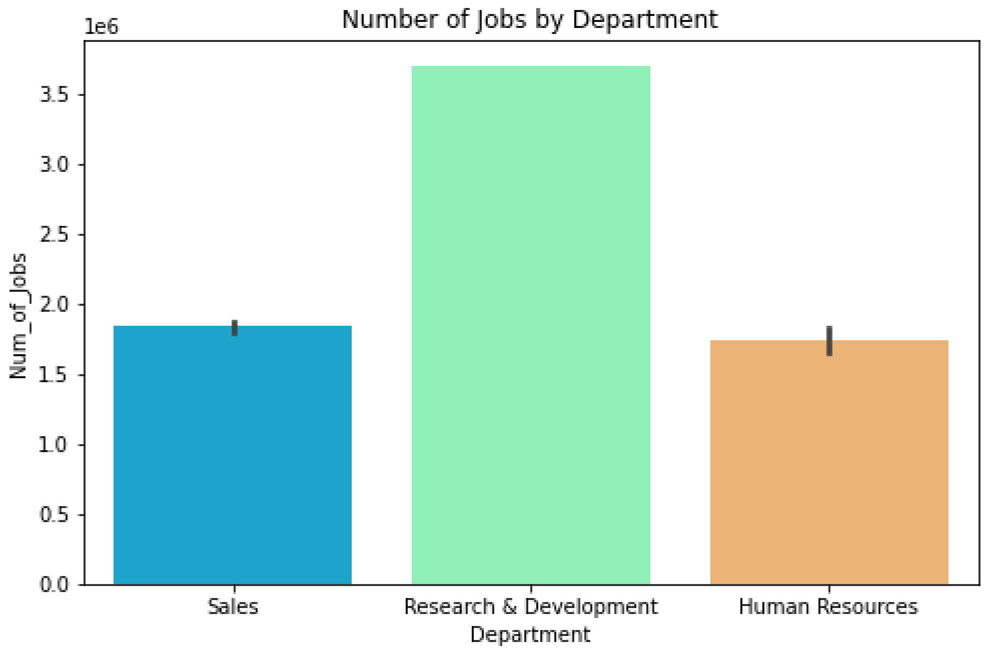
Out[99]:
```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, 'Life Sciences'),
  Text(1, 0, 'Other'),
  Text(2, 0, 'Medical'),
  Text(3, 0, 'Marketing'),
  Text(4, 0, 'Technical Degree'),
  Text(5, 0, 'Human Resources')])
```



In [110]:

```
plt.figure(figsize=(8,5))
sns.barplot(x='Department',y='Num_of_Jobs',data=df_merge2, palette='ra
plt.title("Number of Jobs by Department")
```

Out[110]: Text(0.5, 1.0, 'Number of Jobs by Department')



Summary

Through the course of the project, I realized the challenges of choosing data sources so early into the project. While I read the overall scope of the project at the start, it wasn't until I progressed through the milestones that I began to understand the impacts that the data sources I selected had on the flexibility of the final outcome. By choosing a more unique flat file, I found it difficult to join the API and website data, as those both had to be aggregated in order to be joined. In a similar vein, the variety within the flat file did not always match to the availability in the other sources. While file manipulation and transformations were done in order to create a primary key to join the data on, each source had a variation in sample size that didn't inherently match with the other. This led to a inequal weight of different categorical fields.

Fortunately, this was a fabricated project, so the volatility in available resources is to be expected. In a real-world application, a heavier weight and consideration to the population sizes would be taken into greater analysis before being signed off on as a reliable data source. This could include conversations with key stakeholders, HR business partners, privacy, legal, and even IT (for reliability of outside data sources). Ethical considerations also need to be made to the accuracy and soundness of any online data source. While these data sources could be used to inform analysis and qualitative data, they would not be large enough in sample size or historical record to drive business decisions.

As stated in earlier milestones, working with employee data involves a lot of sentiment and sensitive information. The proposal, data sources, and use case of the final analysis would all need to be approved by key business partners in order to account for PII, sample bias, and assumption bias.