In [1]:

```
import pandas as pd
all_contributions = pd.read_csv('processedPoliceContributions.csv')
all_misconducts = pd.read_csv('processedBostonPoliceInternalAffairs.csv')
final_df = pd.read_csv('filtered_final_dataset.csv')
```

In [2]:

```
all_contributions = all_contributions.drop(['Unnamed: 0'],axis=1)
all_contributions['Employer'] = all_contributions['Employer'].str.lower()
all_contributions['Occupation'] = all_contributions['Occupation'].str.lower()
bos_cont = all_contributions[all_contributions['Employer'].str.contains("boston"
```

Finding Similarity Between Our Datasets

In [3]:

```
all_disciplined_n = all_misconducts['Name'].nunique()
```

In [4]:

```
all_contributors_n = bos_cont['Contributor'].nunique()
```

In [5]:

```
unique_both = final_df['Name'].nunique()
```

In [6]:

```
ers that contributed money out of the disciplined officers (minimum of one payment
ers that contributed money in disciplined officers set out of all boston area poli
```

```
Percentage of officers that contributed money out of the disciplined officers
(minimum of one payment) = 0.27098976109215017
Percentage of officers that contributed money in disciplined officers set out o
f all boston area police donations = 0.32946058091286307
```

Takeaway: We now know that about 27% of all disciplined officers contributed at least once to political campaigns, and that about 33% of all Boston Police officers who made donations were also accused of misconduct.

Let's explore the demographics and positions of ALL officers under investigation

In [7]:

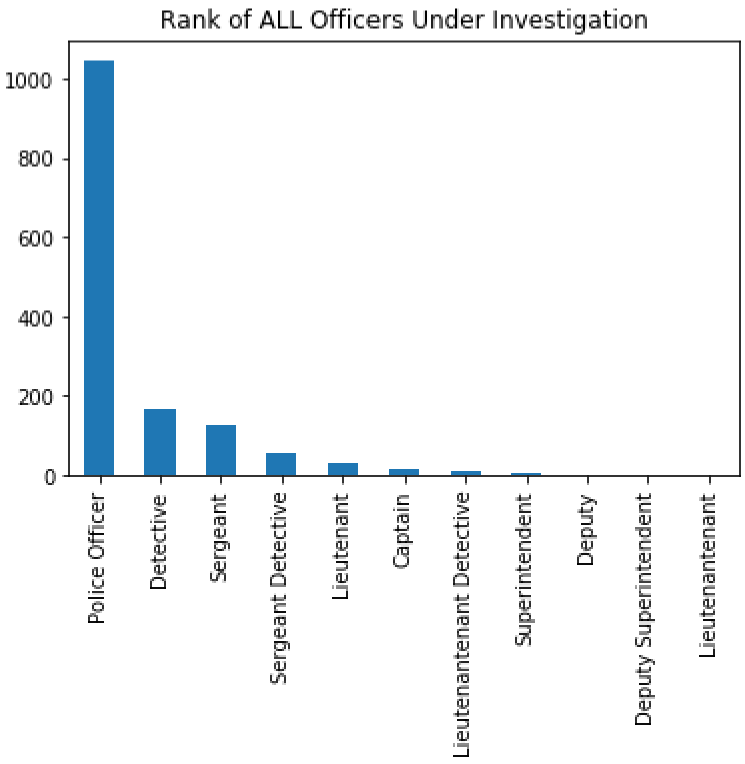
```
import matplotlib.pyplot as plt

all_misconducts.drop_duplicates(subset = ["Name"])['Race'].value_counts().plot(k
# plt.legend(bbox_to_anchor=(1,0), loc="best")
plt.title("All Officers Under Investigation")
plt.show()
```

```
<Figure size 640x480 with 1 Axes>
```

In [8]:
```
all_misconducts['Rank'].value_counts()
```

Out[8]:
```
Police Officer                 4017
Detective                       581
Sergeant                        501
Sergeant Detective              243
Lieutenant                      114
Captain                          82
Lieutenantenant Detective        73
Deputy Superintendent            30
Superintendent                   12
Lieutenantenant                   4
Deputy                            3
Name: Rank, dtype: int64
```

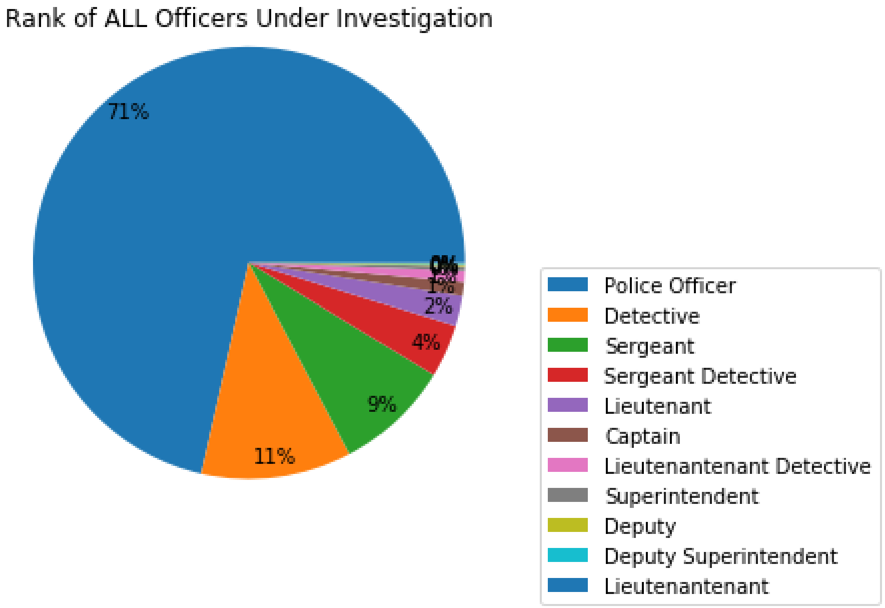
In [9]:
```
# all_misconducts['Rank'].value_counts().plot(kind='pie',autopct='%1.1f%%',radius
all_misconducts.drop_duplicates(subset = ["Name"])['Rank'].value_counts().plot(k
plt.title("Rank of ALL Officers Under Investigation")
# plt.legend(bbox_to_anchor=(1,0), loc="best")
plt.show()
```



In [10]:
```
vals = all_misconducts.drop_duplicates(subset = ["Name"])['Rank'].value_counts()
labels = vals.index.tolist()
x = vals.tolist()
```

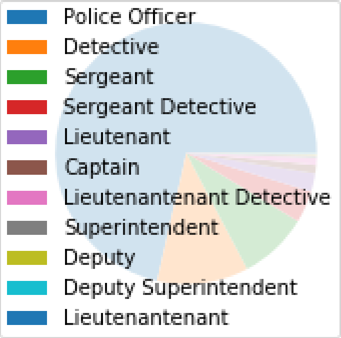
In [38]:

```
plt.figure()

plt.gca().axis("equal")
pie = plt.pie(x, startangle=0, autopct='%1.0f%%', pctdistance=0.9, radius=1.2)
plt.title('Rank of ALL Officers Under Investigation')
plt.legend(pie[0],labels, bbox_to_anchor=(0.75,0.5), loc="best",
           bbox_transform=plt.gcf().transFigure)
plt.subplots_adjust(left=0.0, bottom=0.1, right=0.85)

plt.show()
plt.clf()
plt.close()
```



In [12]:

```
plt.figure(figsize=(3, 3), dpi=72)
plt.pie(x)
plt.legend(labels=labels)
plt.show()
# .plot(kind='pie',autopct='%1.1f%%',radius=1)
# # all_misconducts['Rank'].value_counts().plot(kind='bar')
# plt.title("Rank of ALL Officers Under Investigation")
# # plt.legend(bbox_to_anchor=(1,0), loc="best")
# plt.show()
```



In [13]:

```
all_misconducts['Allegation'].value_counts()
```

Out[13]:

```
Neg.Duty/Unreasonable Judge
1454
Respectful Treatment
1305
Force
536
Conduct Unbecoming
362
Conformance to Laws
338

...
Prisoner
1
Police Officer Assigned to Dept. M/V
1
alcohol
1
Special Order 97-35 Motor Vehicle Accidents Involving Dept. Vehicles and/or Swo
rn Personnel       1
Sick & Injured Persons
1
Name: Allegation, Length: 95, dtype: int64
```

In [14]:
```
df2 = pd.DataFrame({'Name':all_misconducts.Name.unique()})
df2['CaseID'] = [list(set(all_misconducts['CaseID'].loc[all_misconducts['Name']
df2
```

Out[14]:

	Name	CaseID
0	joseph abasciano	[IAD2013-0019, IAD2011-0182, IAD2019-0085]
1	ramadani abdul-aziz	[IAD2017-0431, IAD2011-0258]
2	patrick olaf abrahamson	[IAD2015-0109, IAD2015-0642, IAD2017-0257]
3	cesar abreu	[IAD2015-0056, IAD2016-0253, IAD2015-0491, IAD...
4	moises j abreu	[IAD2015-0680]
...	...	...
1460	tommy t yung	[IAD2012-0196, IAD2019-0294]
1461	joseph m zanoli	[IAD2013-0348, IAD2013-0309, IAD2016-0024]
1462	kevin zarnoch	[IAD2016-0203, IAD2019-0157]
1463	robert m zingg	[IAD2011-0553, IAD2012-0039]
1464	peter a zographos	[IAD2012-0336]

1465 rows × 2 columns

In [15]:
```
individual_misconducts = {}
case_lst = {}
for n in all_misconducts['Name'].unique():
    id_lst = []
    individual_misconducts[n] = []
    for d in all_misconducts.loc[all_misconducts['Name'] == n, 'CaseID']:
        if d not in id_lst:
            id_lst.append(d)
            s = ((all_misconducts.loc[(all_misconducts['CaseID'] == d )& (all_mi
            individual_misconducts[n].append(s)
    case_lst[n] = id
```

In [16]:
```
individual_allegations = {}
case_lst = {}
for n in all_misconducts['Name'].unique():
    id_lst = []
    individual_allegations[n] = []
    for d in all_misconducts.loc[all_misconducts['Name'] == n, 'CaseID']:
        if d not in id_lst:
            id_lst.append(d)
            s = ((all_misconducts.loc[(all_misconducts['CaseID'] == d )& (all_mi
            individual_allegations[n].append(s)
    case_lst[n] = id
```

In [17]:

```
individual_findings = {}
case_lst = {}
for n in all_misconducts['Name'].unique():
    id_lst = []
    individual_findings[n] = []
    for d in all_misconducts.loc[all_misconducts['Name'] == n, 'CaseID']:
        if d not in id_lst:
            id_lst.append(d)
            s = ((all_misconducts.loc[(all_misconducts['CaseID'] == d )& (all_mi
            individual_findings[n].append(s)
    case_lst[n] = id
```

In [18]:

```
individual_misconducts
```

```
  'Citizen complaint',
  'Internal investigation'],
 'gilbert alicea': ['Internal investigation'],
 'hector r alicea': ['Citizen complaint', 'Citizen complaint'],
 'frederick r allen': ['Citizen complaint', 'Internal investigation'],
 'ana c almeida': ['Citizen complaint', 'Citizen complaint'],
 'ismael lopes almeida': ['Citizen complaint'],
 'hugo r alvarez': ['Citizen complaint', 'Citizen complaint'],
 'john f alves': ['Citizen complaint',
  'Citizen complaint',
  'Citizen complaint',
  'Citizen complaint',
  'Citizen complaint'],
 'jose d amado': ['Citizen complaint'],
 'jessica c anderson': ['Citizen complaint',
  'Citizen complaint',
  'Citizen complaint',
  'Citizen complaint'],
 'lamont anderson': ['Internal investigation',
  'Citizen complaint',
```

In [19]:
```
individual_allegations
```

Out[19]:
```
{'joseph abasciano': ['Neg.Duty/Unreasonable Judge',
  'Neg.Duty/Unreasonable Judge',
  'Abuse of Process'],
 'ramadani abdul-aziz': ['Neg.Duty/Unreasonable Judge', 'Reporting for Dut
y'],
 'patrick olaf abrahamson': ['Neg.Duty/Unreasonable Judge',
  'BIAS-Free Policing Policy',
  'Self Identification'],
 'cesar abreu': ['Neg.Duty/Unreasonable Judge',
  'Reporting for Duty',
  'Conduct Unbecoming',
  'CORI Access',
  'Neg.Duty/Unreasonable Judge',
  'Conduct Unbecoming',
  'Respectful Treatment',
  'Conduct Unbecoming',
  'Conformance to Laws'],
 'moises j abreu': ['Respectful Treatment'],
 'rafael w acevedo': ['Evidence'],
```

In [20]:
```
individual_findings
```

Out[20]:
```
{'joseph abasciano': ['Exonerated', 'Not Sustained', 'Not Sustained'],
 'ramadani abdul-aziz': ['Sustained', 'Sustained'],
 'patrick olaf abrahamson': ['Unfounded', 'Not Sustained', 'Not Sustained'],
 'cesar abreu': ['Sustained',
  'Sustained',
  'Unfounded',
  'Unfounded',
  'Unfounded',
  'Not Sustained',
  'Unfounded',
  'Sustained',
  'Sustained'],
 'moises j abreu': ['Unfounded'],
 'rafael w acevedo': ['Sustained'],
 'jean moise acloque': ['Unfounded', 'Unfounded'],
 'jose l acosta': ['Not Sustained', 'Sustained', 'Unfounded'],
 'christopher adams': ['Not Sustained'],
 'christopher p adams': ['Unfounded',
  'Not Sustained',
```

In [21]:

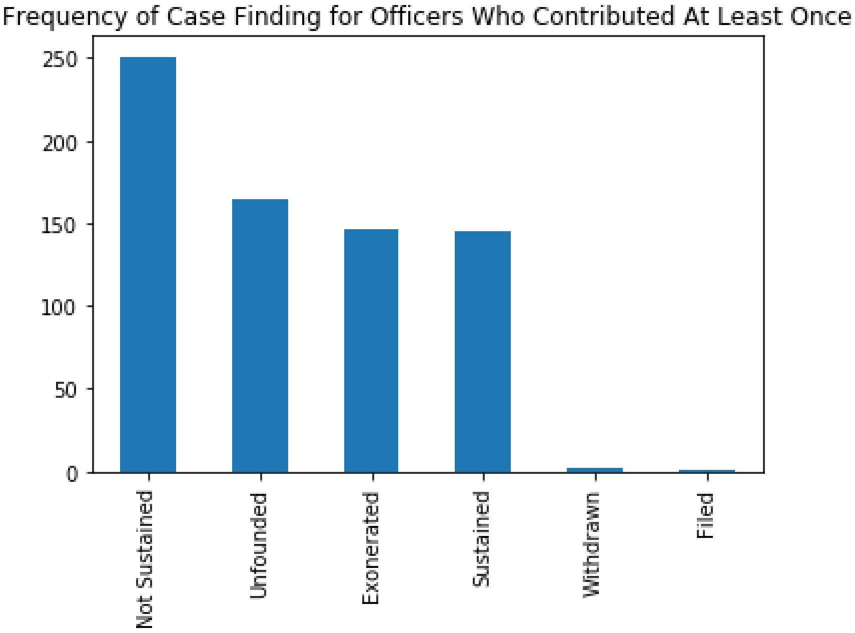
```
unique_case_df = final_df.drop_duplicates(subset = ["CaseID"])
unique_case_df = unique_case_df.drop(['Unnamed: 0'],axis=1)
unique_case_df = unique_case_df.drop(['Unnamed: 0.1'],axis=1)
unique_case_df
```

Out[21]:

	Name	Rank	Race	Year	CaseID	TypeOfMisconduct	Allegation
0	joseph abasciano	Police Officer	White	2011.0	IAD2011-0182	Citizen complaint	Neg.Duty/Unreasonable Judge
4	joseph abasciano	Police Officer	White	2013.0	IAD2013-0019	Citizen complaint	Neg.Duty/Unreasonable Judge
6	joseph abasciano	Police Officer	White	2019.0	IAD2019-0085	Citizen complaint	Abuse of Process
56	cesar abreu	Police Officer	Hispanic	2011.0	IAD2011-0548	Citizen complaint	Neg.Duty/Unreasonable Judge
58	cesar abreu	Police Officer	Hispanic	2014.0	IAD2014-0612	Internal investigation	Reporting for Duty
...	...	...	...	...	...	...	...
	vladimir	Police			IAD2012		

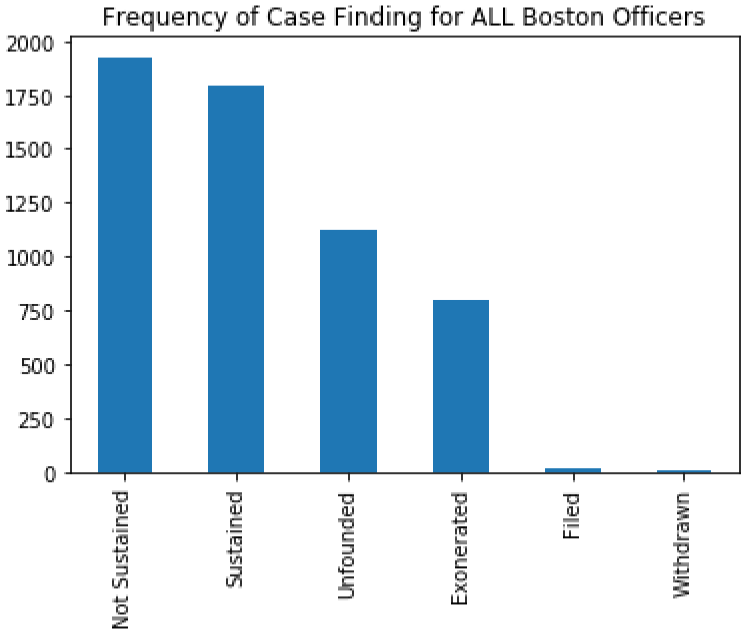
In [22]:

```
unique_case_df['Finding'].value_counts().plot(kind='bar')
plt.title("Frequency of Case Finding for Officers Who Contributed At Least Once"
plt.show()
```



In [23]:

```
all_misconducts['Finding'].value_counts().plot(kind='bar')
plt.title("Frequency of Case Finding for ALL Boston Officers")
plt.show()
```



In [ ]: