US Presidential Campaign Data Exploration

Dr. Bruns

8 October 2019

Student Name: Raymond Shum

Class: CST383-30_SP22

Assignment: W5 - HW5a

Due Date: 8-Feb-2022

Instructions:

- Problems 1-12 are shown in code cells below
- Each problem begins with #@
- Insert your code below the problem line
- Do not make changes outside the problem cells, except to change the name and date above
- Be sure to include plot titles, labels, etc. as shown

An exploration of California campaign contribution data for the 2016 US presidential election.

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
sns.set()
rcParams['figure.figsize'] = 8,6
sns.set_context('talk')   # 'talk' for slightly larger
```

In [2]:
```
# code in this cell from:
# https://stackoverflow.com/questions/27934885/how-to-hide-code-from-cells-in-ipython-notebook-visualized-with-
from IPython.display import HTML

HTML('''<script>
code_show=true;
function code_toggle() {
 if (code_show){
 $('div.input').hide();
 } else {
 $('div.input').show();
 }
 code_show = !code_show
}
$( document ).ready(code_toggle);
</script>
<form action="javascript:code_toggle()"><input type="submit" value="Click here to display/hide the code."></for
```

Out[2]: Click here to display/hide the code.

In [3]:
```
# This is a randomly-sample subset of the full data set.
df = pd.read_csv("https://raw.githubusercontent.com/grbruns/cst383/master/campaign-ca-2016-sample.csv")
```

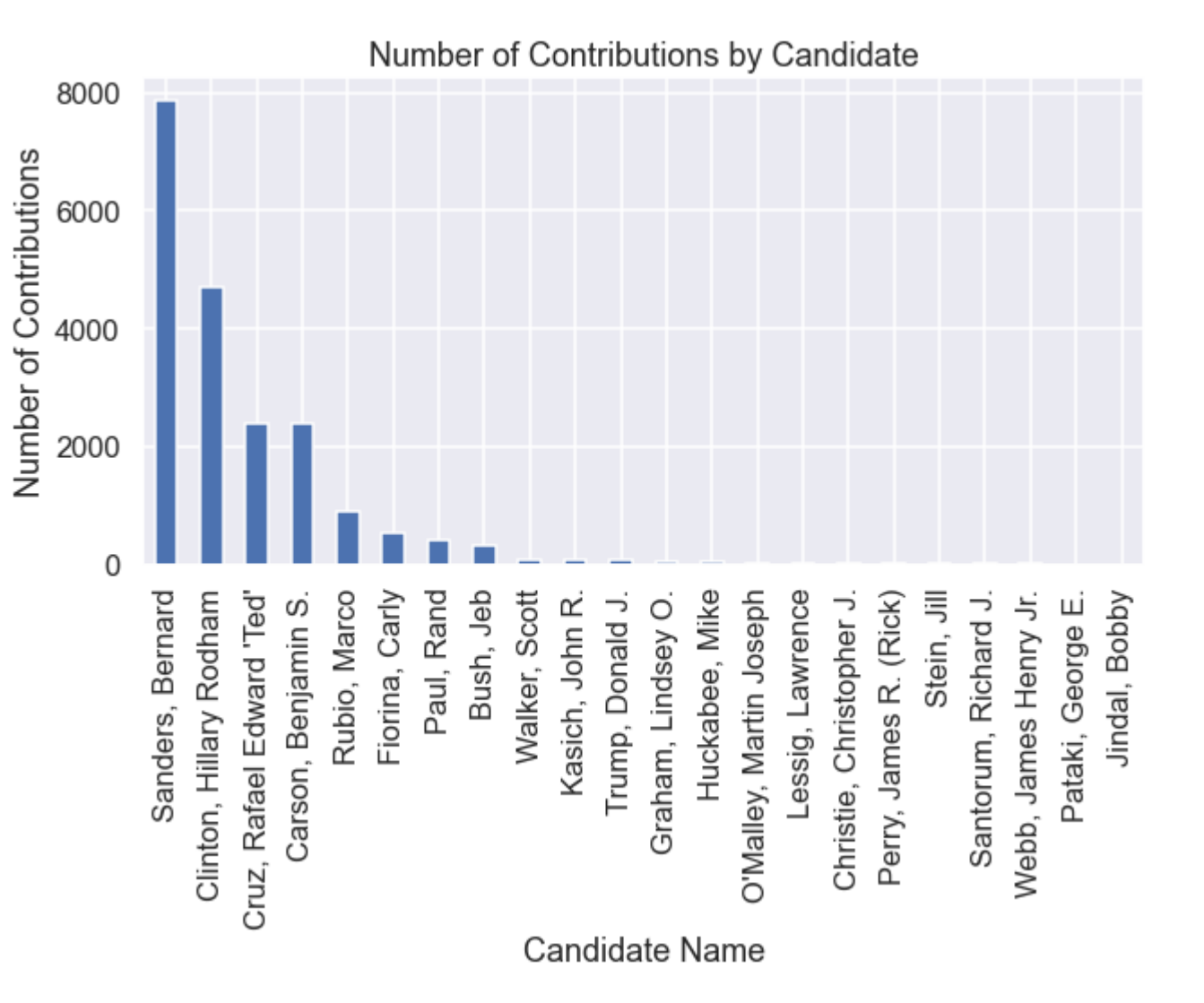
Which candidates received the most contributions?

In [4]:
```
#@ 1 Show the number of contributions by candidate using a barplot.
# Hint: use value_counts(), and use a pandas bar plot.

num_contb = df[[
                'cand_nm',
                'contb_receipt_amt'
                ]] \
                .groupby('cand_nm') \
                .count() \
                .sort_values(
                    by='contb_receipt_amt',
                    ascending=False)

chart = num_contb.plot(kind='bar', figsize=(10,5), legend=False)
chart.set_title('Number of Contributions by Candidate')
chart.set_xlabel('Candidate Name')
chart.set_ylabel('Number of Contributions')
```

Out[4]:
```
Text(0, 0.5, 'Number of Contributions')
```



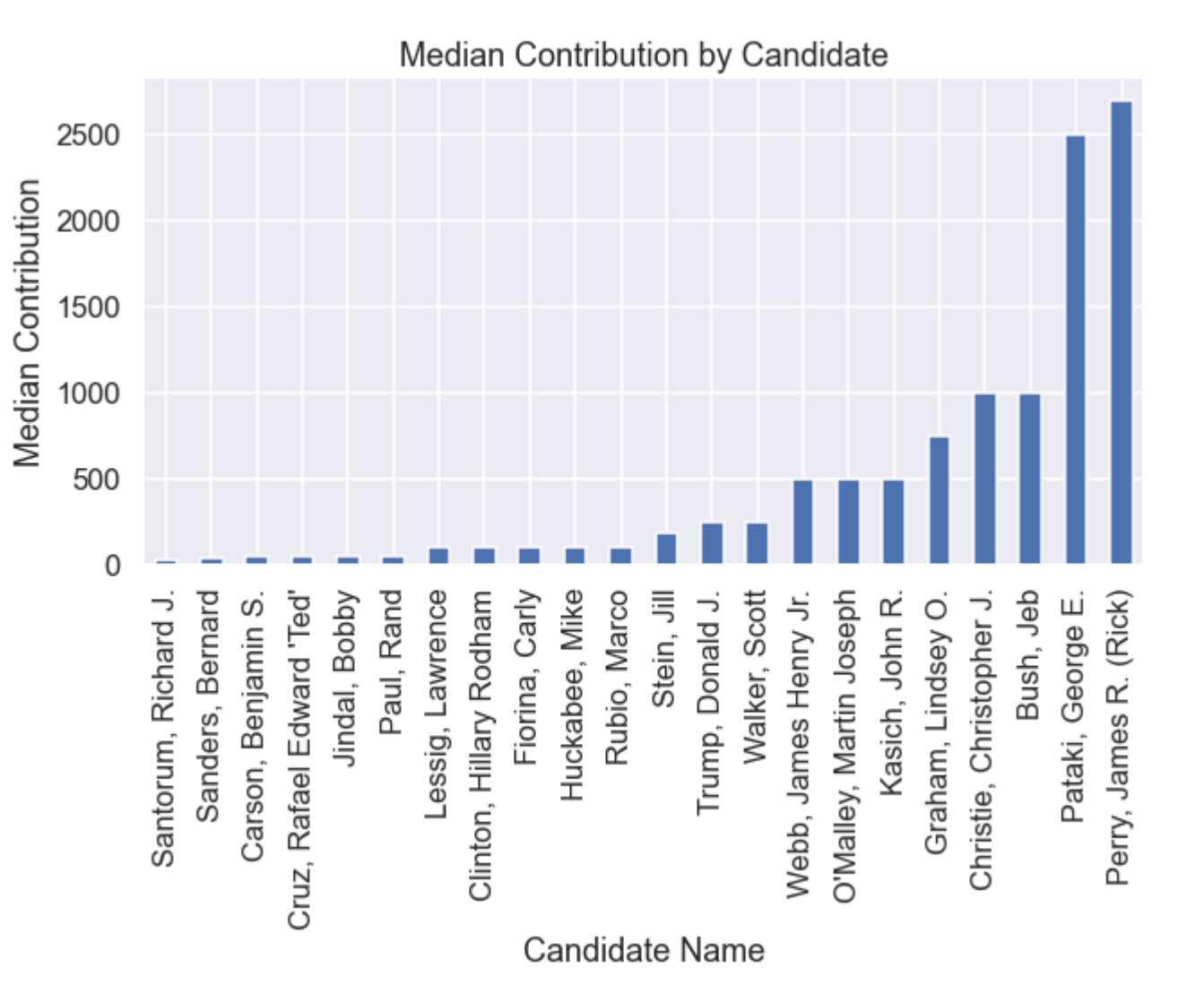
Let's look at the amount of the contributions, instead of the number of contributions. Which candidates had the highest median contribution amounts?

In [5]:
```
#@ 2 Show the median contribution amount by candidate.

med_contb = df[[
                'cand_nm',
                'contb_receipt_amt'
                ]] \
                .groupby('cand_nm') \
                .median() \
                .sort_values(
                    by='contb_receipt_amt')

chart = med_contb.plot(kind='bar', figsize=(10,5), legend=False)
chart.set_title('Median Contribution by Candidate')
chart.set_xlabel('Candidate Name')
chart.set_ylabel('Median Contribution')
```

Out[5]:
```
Text(0, 0.5, 'Median Contribution')
```

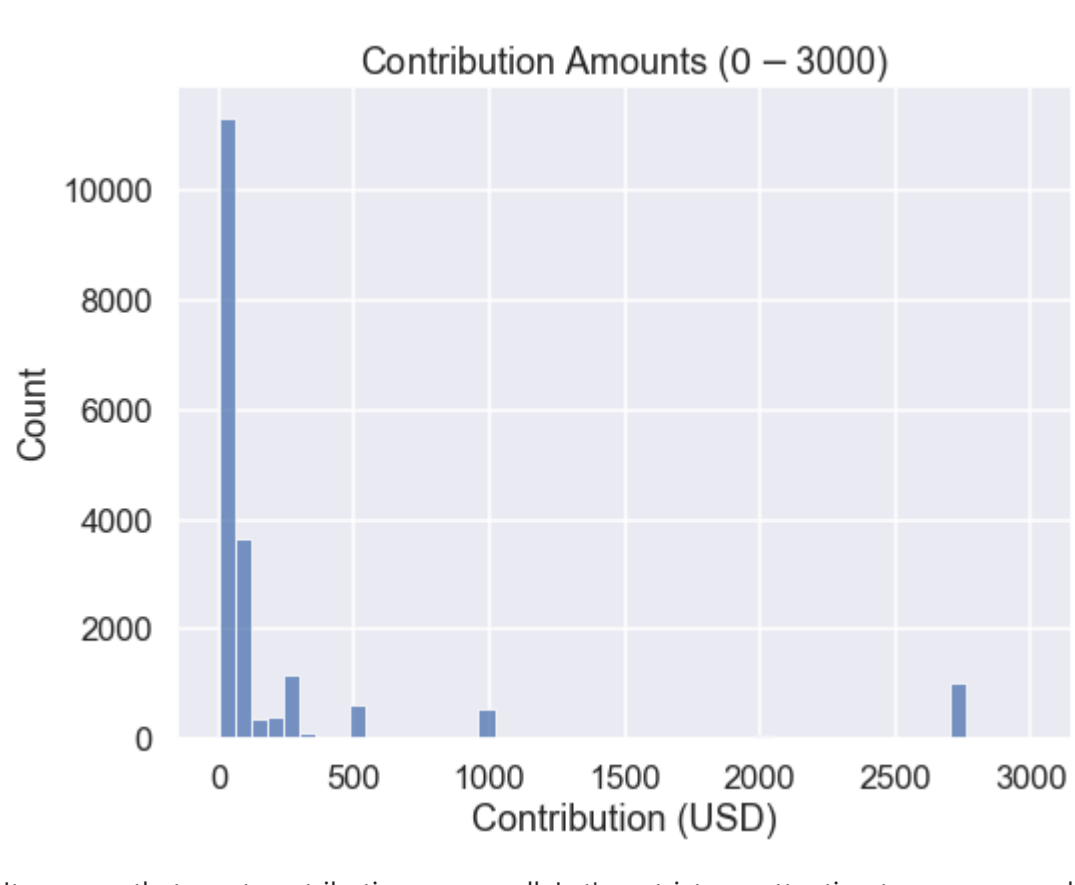


What is the distribution of the contribution amounts? There are a small number of very large amounts, which make it hard to display the distribution. Also, there are some negative contribution amounts that seem to reflect returned contributions. Therefore, let's focus on contributions ranging from 0 to 3,000 dollars.

In [6]:
```
#@ 3 Create a histogram showing contribution amounts.  Show
# contributions from 0 - 3000 dollars only.  Create the
# histogram with Seaborn.

mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 3000)
hist = sns.histplot(data=df.loc[mask, 'contb_receipt_amt'],bins=50)
hist.set_title("Contribution Amounts ($0-$3000)")
hist.set_ylabel("Count")
hist.set_xlabel("Contribution (USD)")
```

Out[6]:
```
Text(0.5, 0, 'Contribution (USD)')
```

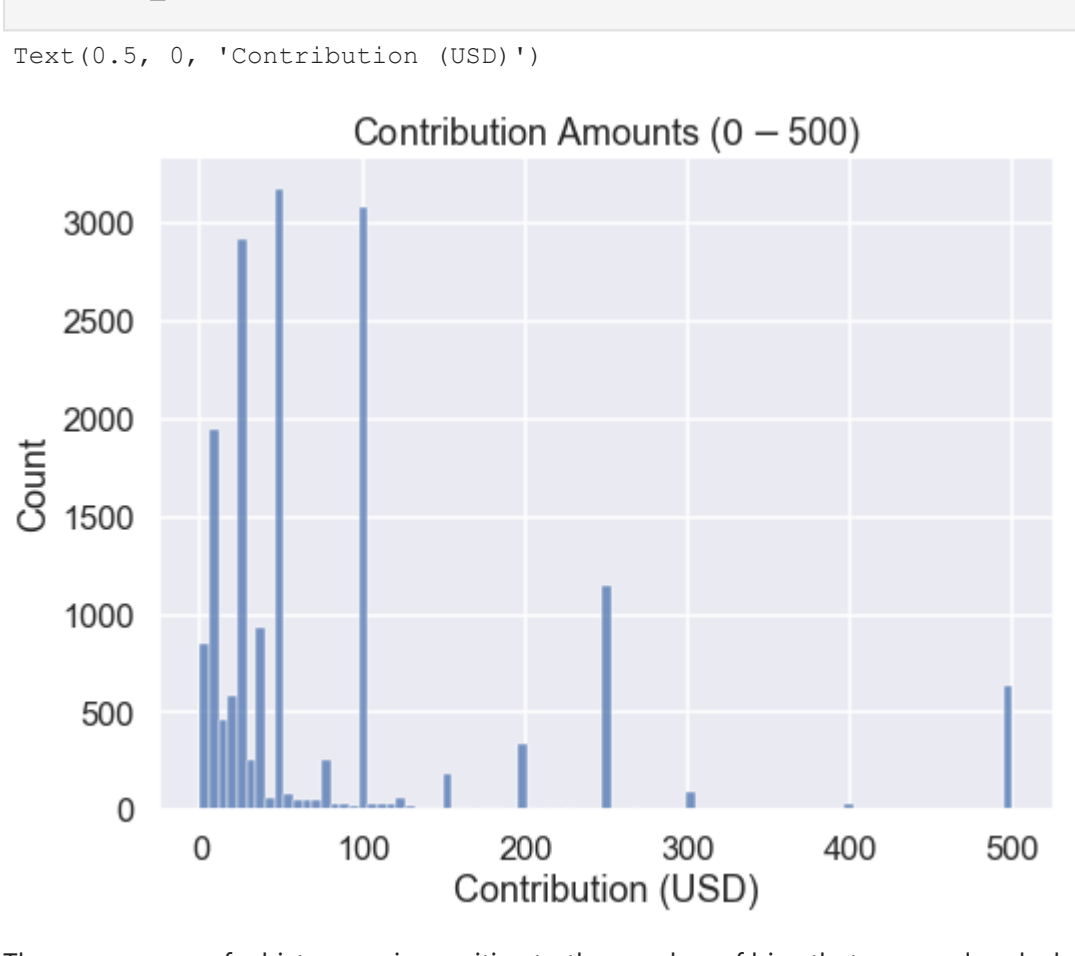


It appears that most contributions are small. Let's restrict our attention to an even smaller range of contributions to get a better idea of how small contributions are distributed.

In [7]:
```
#@ 4 Create a histogram showing contribution amounts.  Show
# contributions from 0 - 500 dollars only.  Create the
# histogram with Seaborn.


mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 500)
hist = sns.histplot(data=df.loc[mask, 'contb_receipt_amt'])
hist.set_title("Contribution Amounts ($0-$500)")
hist.set_ylabel("Count")
hist.set_xlabel("Contribution (USD)")
```

Out[7]:
```
Text(0.5, 0, 'Contribution (USD)')
```

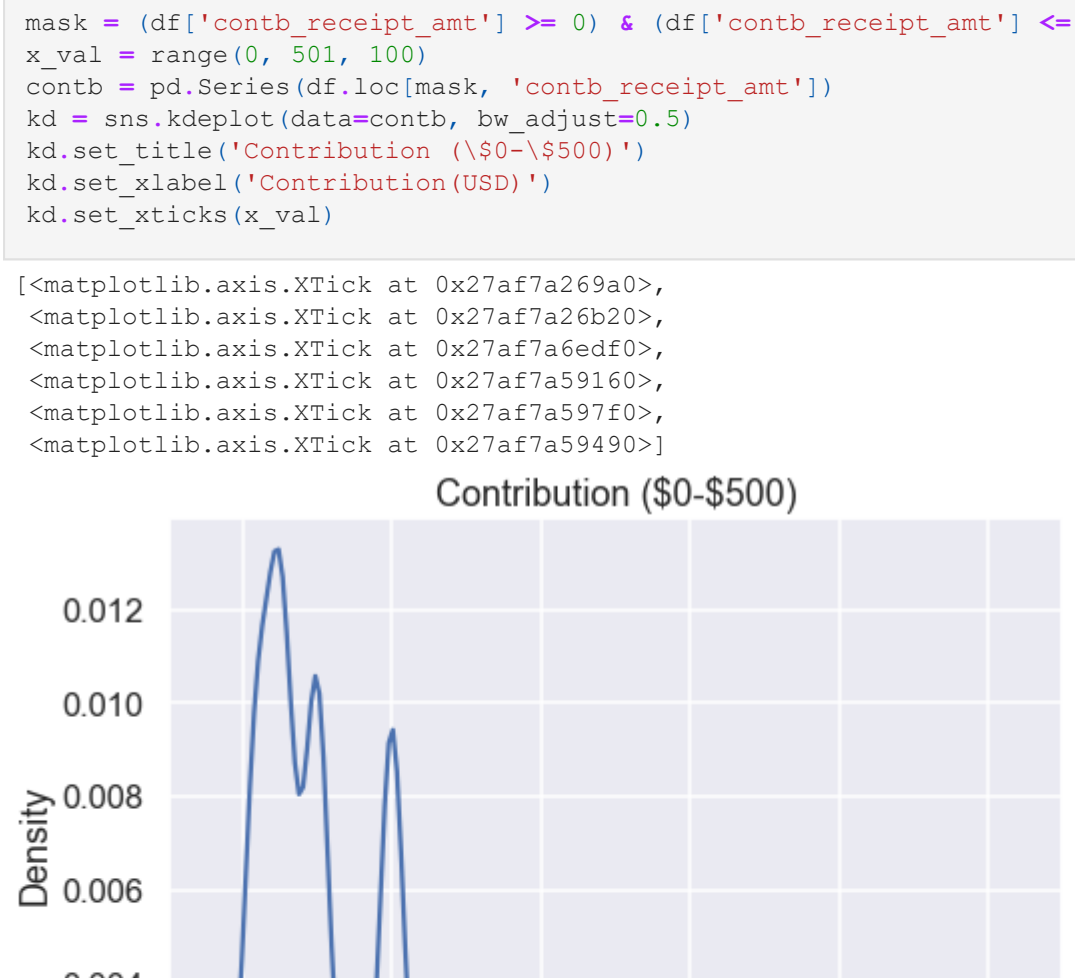


The appearance of a histogram is sensitive to the number of bins that are used and where the bin edges lie. Let's look at the contribution amounts again using a density plot.

In [8]:
```
#@ 5 Create a density plot (sometimes called a kernel density
# plot) showing contribution amounts.  Show contributions from
# 0 - 500 dollars only.  Create the density plot with Seaborn.
# histogram use Seaborn.
# Hint: you may want to start by creating a series containing
# the contb_receipt_amt values from 0-500.

mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 500)
x_val = range(0, 501, 100)
contb = pd.Series(df.loc[mask, 'contb_receipt_amt'])
kd = sns.kdeplot(data=contb, bw_adjust=0.5)
kd.set_title('Contribution (\$0-\$500)')
kd.set_xlabel('Contribution(USD)')
kd.set_xticks(x_val)
```

Out[8]:
```
[<matplotlib.axis.XTick at 0x27af7a269a0>,
 <matplotlib.axis.XTick at 0x27af7a26b20>,
 <matplotlib.axis.XTick at 0x27af7a6edf0>,
 <matplotlib.axis.XTick at 0x27af7a59160>,
 <matplotlib.axis.XTick at 0x27af7a597f0>,
 <matplotlib.axis.XTick at 0x27af7a59490>]
```



Let's compare the size of contributions between candidates Rubio and Cruz. Did one of them tend to get larger-sized contributions?

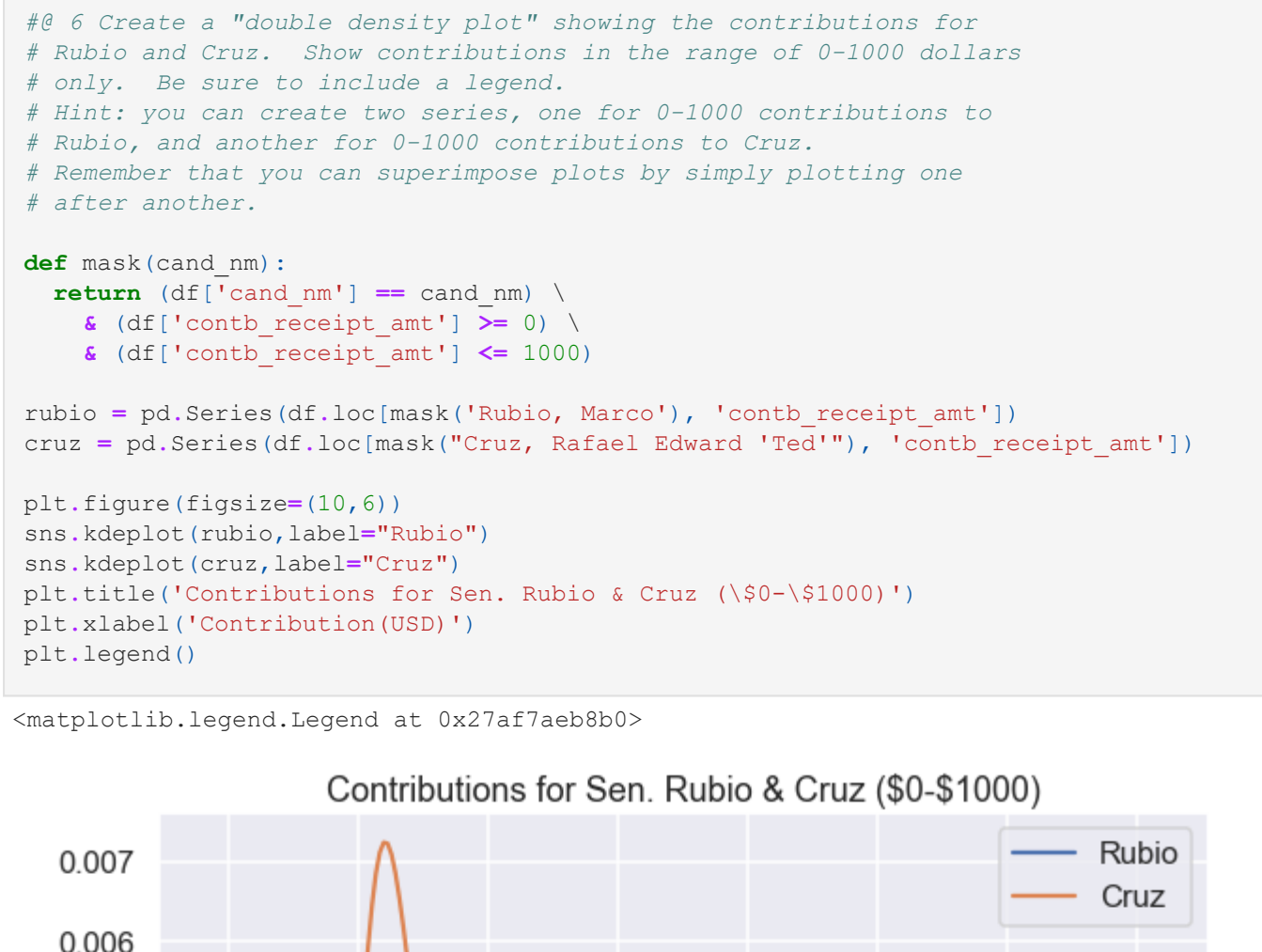
In [9]:
```
#@ 6 Create a "double density plot" showing the contributions for
# Rubio and Cruz.  Show contributions in the range of 0-1000 dollars
# only.  Be sure to include a legend.
# Hint: you can create two series, one for 0-1000 contributions to
# Rubio, and another for 0-1000 contributions to Cruz.
# Remember that you can superimpose plots by simply plotting one
# after another.

def mask(cand_nm):
  return (df['cand_nm'] == cand_nm) \
    & (df['contb_receipt_amt'] >= 0) \
    & (df['contb_receipt_amt'] <= 1000)

rubio = pd.Series(df.loc[mask('Rubio, Marco'), 'contb_receipt_amt'])
cruz = pd.Series(df.loc[mask("Cruz, Rafael Edward 'Ted'"), 'contb_receipt_amt'])

plt.figure(figsize=(10,6))
sns.kdeplot(rubio,label="Rubio")
sns.kdeplot(cruz,label="Cruz")
plt.title('Contributions for Sen. Rubio & Cruz (\$0-\$1000)')
plt.xlabel('Contribution(USD)')
plt.legend()
```

Out[9]:
```
<matplotlib.legend.Legend at 0x27af7aeb8b0>
```



Rubio and Cruz were Republican candidates. Let's look at a pair of Democratic candidates.

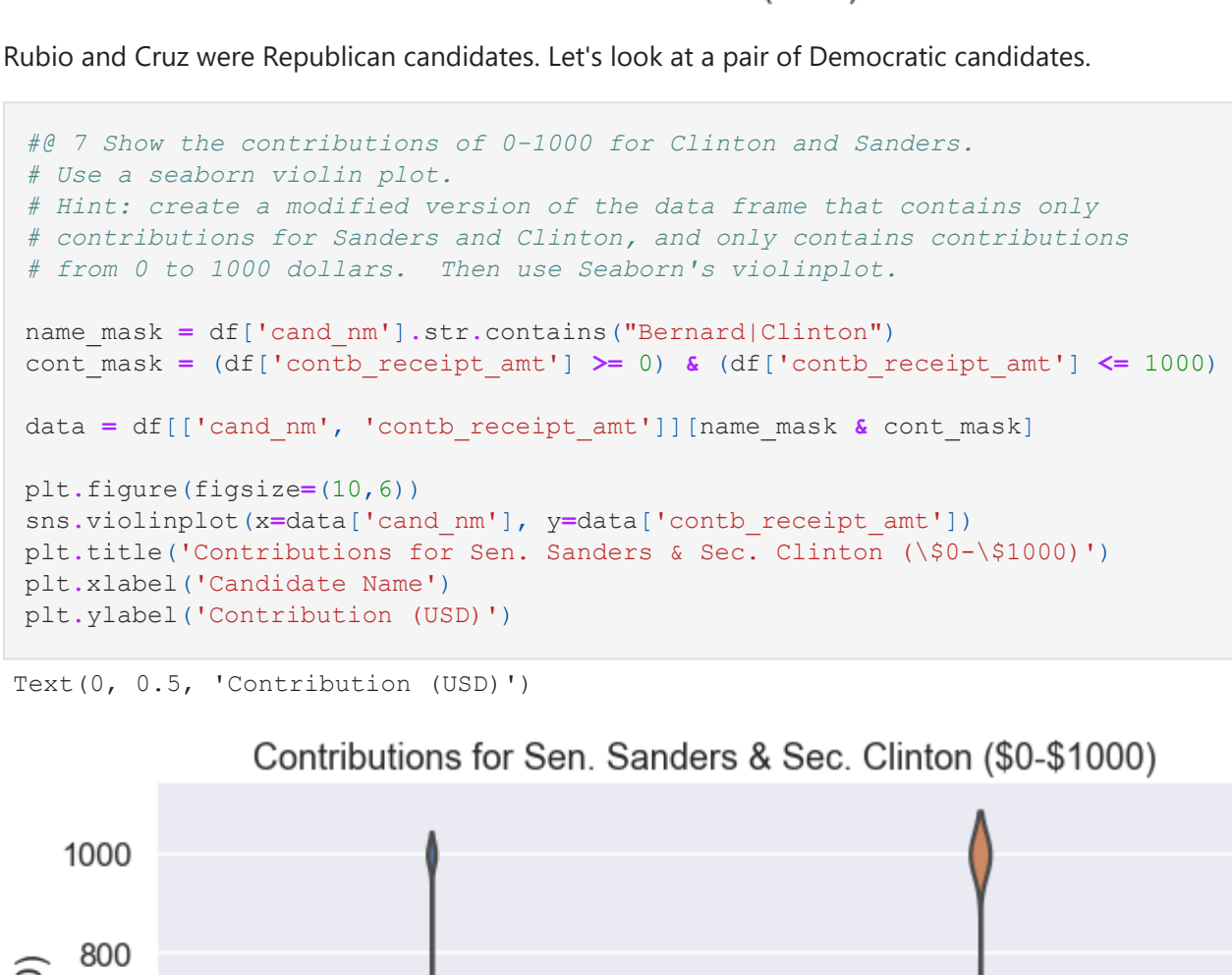
In [10]:
```
#@ 7 Show the contributions of 0-1000 for Clinton and Sanders.
# Use a seaborn violin plot.
# Hint: create a modified version of the data frame that contains only
# contributions for Sanders and Clinton, and only contains contributions
# from 0 to 1000 dollars.  Then use Seaborn's violinplot.

name_mask = df['cand_nm'].str.contains("Bernard|Clinton")
cont_mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 1000)

data = df[['cand_nm', 'contb_receipt_amt']][name_mask & cont_mask]

plt.figure(figsize=(10,6))
sns.violinplot(x=data['cand_nm'], y=data['contb_receipt_amt'])
plt.title('Contributions for Sen. Sanders & Sec. Clinton (\$0-\$1000)')
plt.xlabel('Candidate Name')
plt.ylabel('Contribution (USD)')
```

Out[10]:
```
Text(0, 0.5, 'Contribution (USD)')
```



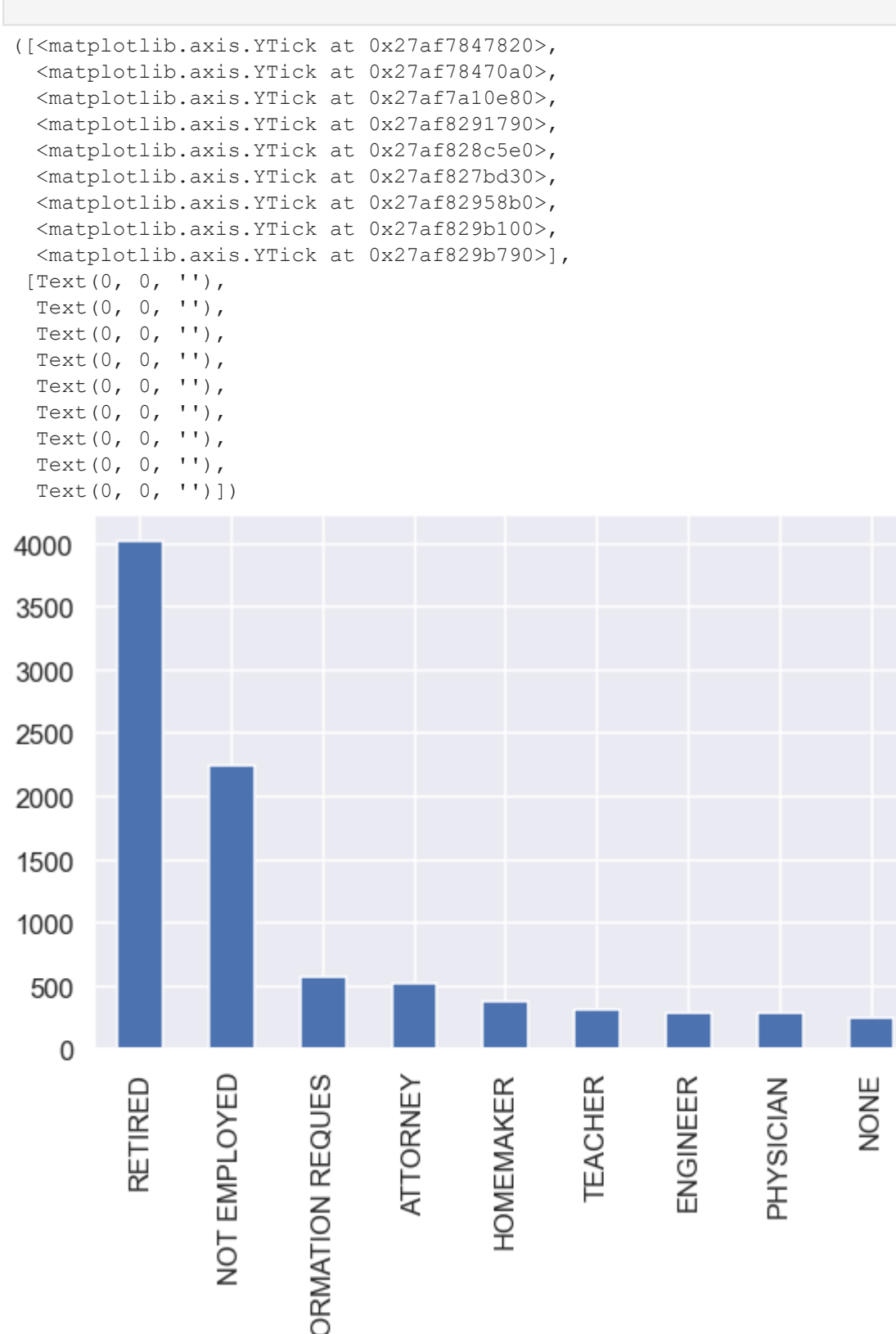
Which occupations are associated with the greatest number of contributions? This will be interesting, but we need to keep in mind that the occupation with the greatest number of contributions might just be the most common occupation.

In [11]:
```
#@ 8 Create a bar plot showing th total number of contributions by occupation,
# for the 10 occupations with the largest number of contributions.  Use
# Pandas for the bar plot.  Limit the occupation names to 18 characters.
# Hint: to limit the occupation names to 18 characters, you can create a
# new column 'short_occ' by using pd.Series.str.slice on the
# 'contbr_occupation' column.

df['short_occ'] = df['contbr_occupation'].str.slice(0,18)

plt.figure(figsize=(10,6))
occ_cont = df['short_occ'].value_counts().nlargest(10)
occ_cont.plot(kind='bar')
plt.yticks(range(0,4001,500))
```

Out[11]:
```
([<matplotlib.axis.YTick at 0x27af7847820>,
  <matplotlib.axis.YTick at 0x27af78470a0>,
  <matplotlib.axis.YTick at 0x27af7a10e80>,
  <matplotlib.axis.YTick at 0x27af8291790>,
  <matplotlib.axis.YTick at 0x27af828c5e0>,
  <matplotlib.axis.YTick at 0x27af827bd30>,
  <matplotlib.axis.YTick at 0x27af82958b0>,
  <matplotlib.axis.YTick at 0x27af829b100>,
  <matplotlib.axis.YTick at 0x27af829b790>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



We can classify contributors as either employed, unemployed, or retired. Among these groups, which makes the most contributions?

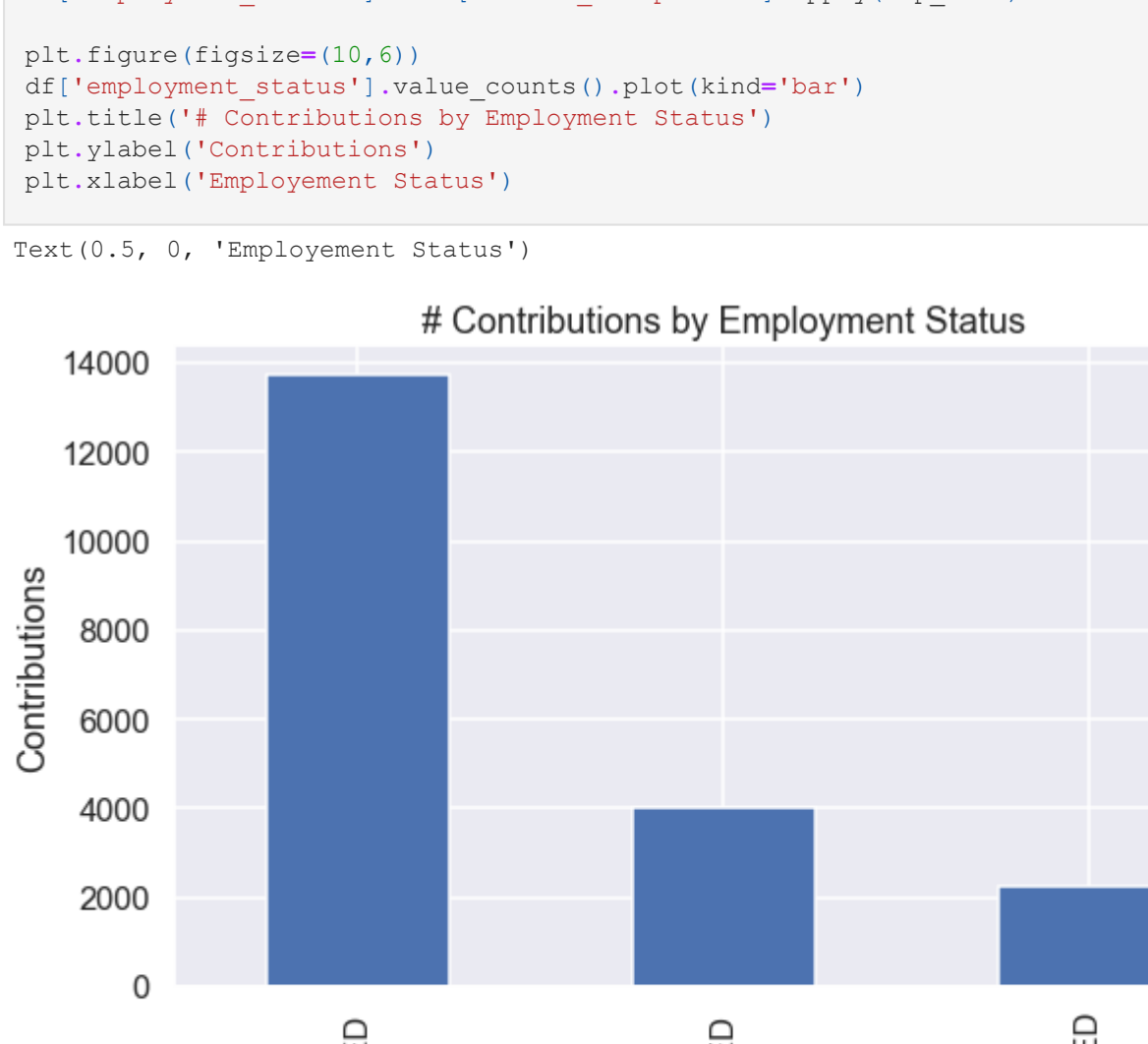
In [12]:
```
#@ 9 Create a new column "employment_status", derived from the
# contbr_occupation column.  The value of employment_status should
# be "EMPLOYED" if contbr_occupation is not "RETIRED" or "NOT EMPLOYED",
# and should be the original contbr_occupation otherwise.  Show the
# number of contributions by employment status as a bar plot.
# Hint: to create the new column, consider creating a function that
# takes as input a contbr_occupation value and returns an employement
# status value.  Then use this function with 'apply'.

def emp_stat(x):
  return x if (x == 'RETIRED' or x == 'NOT EMPLOYED') else 'EMPLOYED'

df['employment_status'] = df['contbr_occupation'].apply(emp_stat)

plt.figure(figsize=(10,6))
df['employment_status'].value_counts().plot(kind='bar')
plt.title('# Contributions by Employment Status')
plt.ylabel('Contributions')
plt.xlabel('Employement Status')
```

Out[12]:
```
Text(0.5, 0, 'Employement Status')
```



Do retired contributors tend to make smaller contributions than employed contributors? It seems likely, but what does the data say?

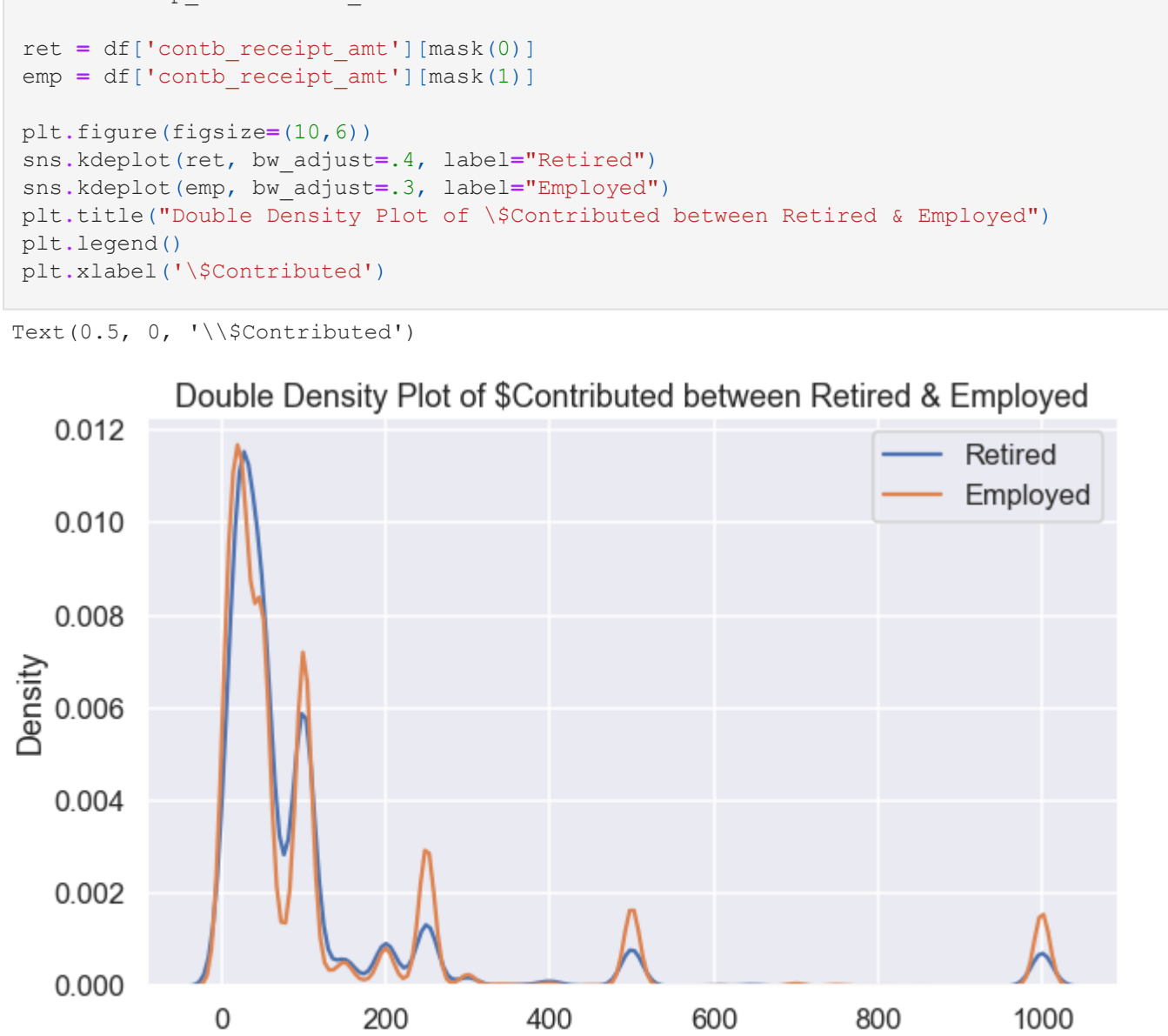
In [13]:
```
#@ 10 Create a double density plot showing the distribution of
# contribution amounts from those with employment_status values
# of RETIRED and EMPLOYED.  Include only contributions of $0-1000.
# Use Seaborn, and make sure to include
# a legend.
# Hint: consider creating two series, one for the contributions
# from retired contributors, and one for the contributions from
# employed contributors.

def mask(switch):
  cont_mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 1000)
  name = "RETIRED" if switch == 0 else "EMPLOYED"
  emp_mask = df['employment_status'] == name
  return emp_mask & cont_mask

ret = df['contb_receipt_amt'][mask(0)]
emp = df['contb_receipt_amt'][mask(1)]

plt.figure(figsize=(10,6))
sns.kdeplot(ret, bw_adjust=.4, label="Retired")
sns.kdeplot(emp, bw_adjust=.3, label="Employed")
plt.title("Double Density Plot of \$Contributed between Retired & Employed")
plt.legend()
plt.xlabel('\$Contributed')
```

Out[13]:
```
Text(0.5, 0, '\\$Contributed')
```



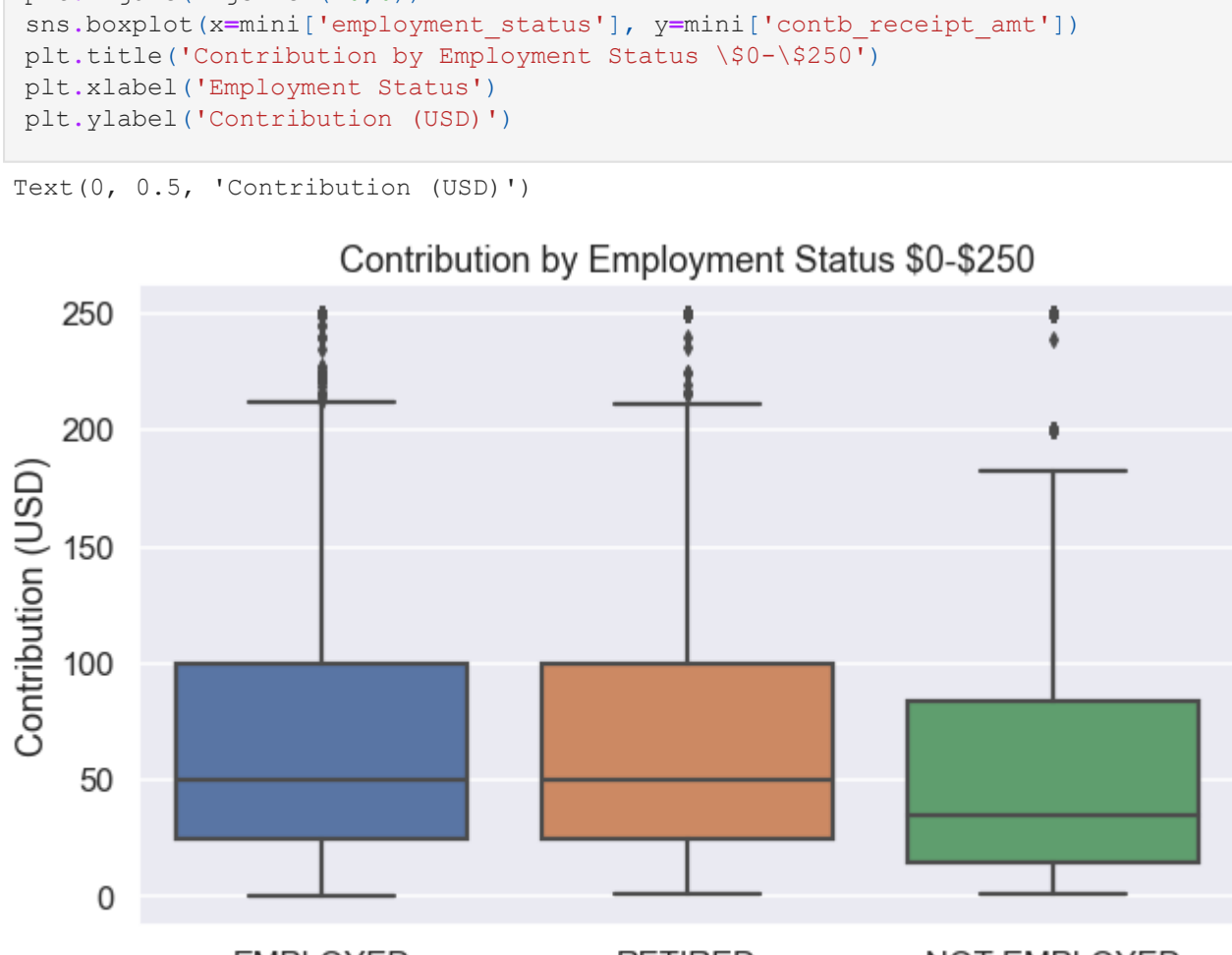
It appears that contributions from the retired and the employed are pretty similar, although there is a significant difference when you focus on larger contributions. Let's look more into the size of contributions from those who are employed, retired, or unemployed.

In [14]:
```
#@ 11 Create a box plot of contribution amounts for each employment
# status category.  Use Seaborn to create the bar plot, and show
# only contributions in the range of $0-250.
# Hint: you may want to create a version of df that contains only
# contributions in the 0-250 range.

mask = (df['contb_receipt_amt'] >= 0) & (df['contb_receipt_amt'] <= 250)
mini = df[['employment_status', 'contb_receipt_amt']][mask]

plt.figure(figsize=(10,6))
sns.boxplot(x=mini['employment_status'], y=mini['contb_receipt_amt'])
plt.title('Contribution by Employment Status \$0-\$250')
plt.xlabel('Employment Status')
plt.ylabel('Contribution (USD)')
```

Out[14]:
```
Text(0, 0.5, 'Contribution (USD)')
```



Previously we looked at the number of contributions from different occupations. What about the size of contributions from different occupations? Let's focus on a few occupations that contribute a lot.

In [15]:
```
#@ 12 Create a bar plot showing the average contribution amount
# for the occupations 'ATTORNEY', 'TEACHER', 'ENGINEER' and 'PHYSICIAN'.
# Include contributions of any amount.  Use Pandas to create the bar plot.
# Show the occupations in decreasing order of mean contribution amount.
# Hint: you may want to create a new data frame that is like df except
# that it only includes data associated with the four occupations.
# To do this, consider the Pandas method pd.Series.isin

mask = df['contbr_occupation'] \
  .isin(['ATTORNEY',
         'TEACHER',
         'ENGINEER',
         'PHYSICIAN'])

data = df[['contbr_occupation','contb_receipt_amt']][mask] \
  .groupby('contbr_occupation') \
  .mean() \
  .sort_values(by='contb_receipt_amt', ascending=False)

plt.figure(figsize=(10,6))
data.plot(kind='bar', legend=False)
plt.yticks(range(0, 601, 100))
plt.title('Average contribution by occupation')
plt.ylabel('Average Contribution (USD)')
plt.xlabel('Occupation')
```

Out[15]:
```
Text(0.5, 0, 'Occupation')
<Figure size 720x432 with 0 Axes>
```