In [1]:
```
import pandas as pd
import numpy as np
from pandas import Series,DataFrame
```

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
%matplotlib inline
```

In [3]:
```
from __future__ import division
```

In [7]:
```
from io import StringIO
```

In [9]:
```
import requests
```

```

```

In [10]:
```
url="http://elections.huffingtonpost.com/pollster/2012-general-election-romn
source=requests.get(url).text
poll_data=StringIO(source)

```

In [12]:
```
poll_df=pd.read_csv(poll_data)
```

In [13]:
```
poll_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 586 entries, 0 to 585
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Pollster                586 non-null    object 
 1   Start Date              586 non-null    object 
 2   End Date                586 non-null    object 
 3   Entry Date/Time (ET)    586 non-null    object 
 4   Number of Observations  564 non-null    float64
 5   Population              586 non-null    object 
 6   Mode                    586 non-null    object 
 7   Obama                   586 non-null    float64
 8   Romney                  586 non-null    float64
 9   Undecided               423 non-null    float64
 10  Other                   202 non-null    float64
 11  Pollster URL            586 non-null    object 
 12  Source URL              584 non-null    object 
 13  Partisan                586 non-null    object 
 14  Affiliation             586 non-null    object 
 15  Question Text           0 non-null      float64
 16  Question Iteration      586 non-null    int64  
dtypes: float64(6), int64(1), object(10)
memory usage: 78.0+ KB
```

In [14]:
```
1 poll_df.head()
```

Out[14]:

	Pollster	Start Date	End Date	Entry Date/Time (ET)	Number of Observations	Population	Mode	Obam
0	Politico/GWU/Battleground	2012-11-04	2012-11-05	2012-11-06T08:40:26Z	1000.0	Likely Voters	Live Phone	47.
1	YouGov/Economist	2012-11-03	2012-11-05	2012-11-26T15:31:23Z	740.0	Likely Voters	Internet	49.
2	Gravis Marketing	2012-11-03	2012-11-05	2012-11-06T09:22:02Z	872.0	Likely Voters	Automated Phone	48.
3	IBD/TIPP	2012-11-03	2012-11-05	2012-11-06T08:51:48Z	712.0	Likely Voters	Live Phone	50.
4	Rasmussen	2012-11-03	2012-11-05	2012-11-06T08:47:50Z	1500.0	Likely Voters	Automated Phone	48.

In [16]:
```
1 sns.factorplot(x='Affiliation',y='Population',data=poll_df)
```

```
C:\Users\Dell\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n
2kfra8p0\LocalCache\local-packages\Python310\site-packages\seaborn\categorical.
py:3717: UserWarning: The `factorplot` function has been renamed to `catplot`.
The original name will be removed in a future release. Please update your code.
Note that the default `kind` in `factorplot` (`'point'`) has changed `'strip'`
in `catplot`.
  warnings.warn(msg)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [16], in <cell line: 1>()
----> 1 sns.factorplot(x='Affiliation',y='Population',data=poll_df)

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p
0\LocalCache\local-packages\Python310\site-packages\seaborn\categorical.py:3727
, in factorplot(*args, **kwargs)
   3723     warnings.warn(msg, UserWarning)
   3725 kwargs.setdefault("kind", "point")
-> 3727 return catplot(*args, **kwargs)

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p
0\LocalCache\local-packages\Python310\site-packages\seaborn\_decorators.py:46,
 in _deprecate_positional_args.<locals>.inner_f(*args, **kwargs)
     36     warnings.warn(
     37         "Pass the following variable{} as {}keyword arg{}: {}. "
     38         "From version 0.12, the only valid positional argument "
   (...)
     43         FutureWarning
     44     )
     45 kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46 return f(**kwargs)

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p
0\LocalCache\local-packages\Python310\site-packages\seaborn\categorical.py:3792
, in catplot(x, y, hue, data, row, col, col_wrap, estimator, ci, n_boot, units,
seed, order, hue_order, row_order, col_order, kind, height, aspect, orient, col
or, palette, legend, legend_out, sharex, sharey, margin_titles, facet_kws, **kw
args)
   3790 p = _CategoricalPlotter()
   3791 p.require_numeric = plotter_class.require_numeric
-> 3792 p.establish_variables(x_, y_, hue, data, orient, order, hue_order)
   3793 if (
   3794     order is not None
   3795     or (sharex and p.orient == "v")
   3796     or (sharey and p.orient == "h")
   3797 ):
   3798     # Sync categorical axis between facets to have the same categories
   3799     order = p.group_names

File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p
0\LocalCache\local-packages\Python310\site-packages\seaborn\categorical.py:156,
in _CategoricalPlotter.establish_variables(self, x, y, hue, data, orient, orde
r, hue_order, units)
    153         raise ValueError(err)
    155 # Figure out the plotting orientation
```

```
--> 156 orient = infer_orient(
    157     x, y, orient, require_numeric=self.require_numeric
    158 )
    160 # Option 2a:
    161 # We are plotting a single set of data
    162 # ---------------------------------
    163 if x is None or y is None:
    164
    165     # Determine where the data are
```

```
File ~\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p
0\LocalCache\local-packages\Python310\site-packages\seaborn\_core.py:1352, in i
nfer_orient(x, y, orient, require_numeric)
   1350 elif require_numeric and "numeric" not in (x_type, y_type):
   1351     err = "Neither the `x` nor `y` variable appears to be numeric."
-> 1352     raise TypeError(err)
   1354 else:
   1355     return "v"

TypeError: Neither the `x` nor `y` variable appears to be numeric.
```

In [18]:

```
avg=pd.DataFrame(poll_df.mean())
avg.drop('Number of Observations',axis=0,inplace=True)

```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_2344\2144941160.py:1: FutureWarning:
Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only vali
d columns before calling the reduction.
  avg=pd.DataFrame(poll_df.mean())
```

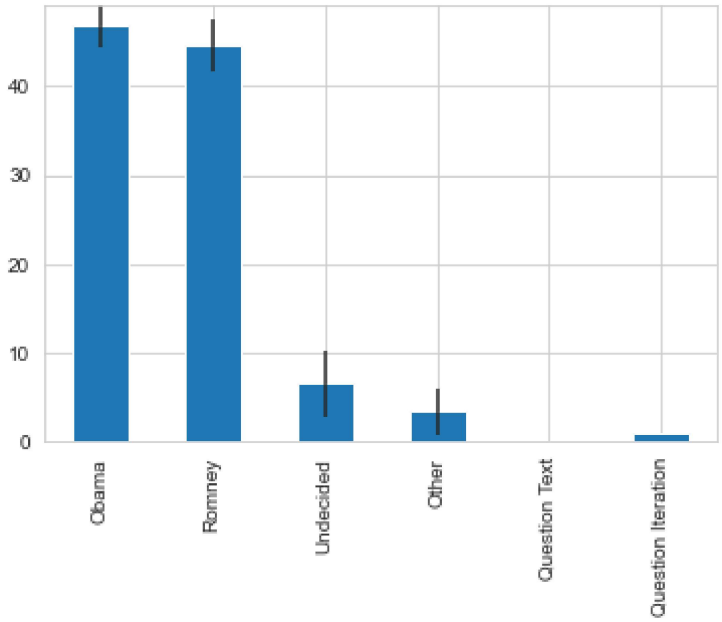
In [19]:

```
std=pd.DataFrame(poll_df.std())
std.drop('Number of Observations',axis=0,inplace=True)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_2344\3034885784.py:1: FutureWarning:
Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None')
is deprecated; in a future version this will raise TypeError.  Select only vali
d columns before calling the reduction.
  std=pd.DataFrame(poll_df.std())
```

In [20]:
```
avg.plot(yerr=std,kind='bar',legend=False)
poll_avg=pd.concat([avg,std],axis=1)
```

```
C:\Users\Dell\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n
2kfra8p0\LocalCache\local-packages\Python310\site-packages\numpy\core\_methods.
py:44: RuntimeWarning: invalid value encountered in reduce
  return umr_minimum(a, axis, None, out, keepdims, initial, where)
C:\Users\Dell\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n
2kfra8p0\LocalCache\local-packages\Python310\site-packages\numpy\core\_methods.
py:40: RuntimeWarning: invalid value encountered in reduce
  return umr_maximum(a, axis, None, out, keepdims, initial, where)
```



In [21]:
```
poll_avg.columns=['Average','STD']
```

In [22]:
```
1 poll_avg
```

Out[22]:

	Average	STD
Obama	46.805461	2.422058
Romney	44.614334	2.906180
Undecided	6.550827	3.701754
Other	3.376238	2.692726
Question Text	NaN	NaN
Question Iteration	1.000000	0.000000

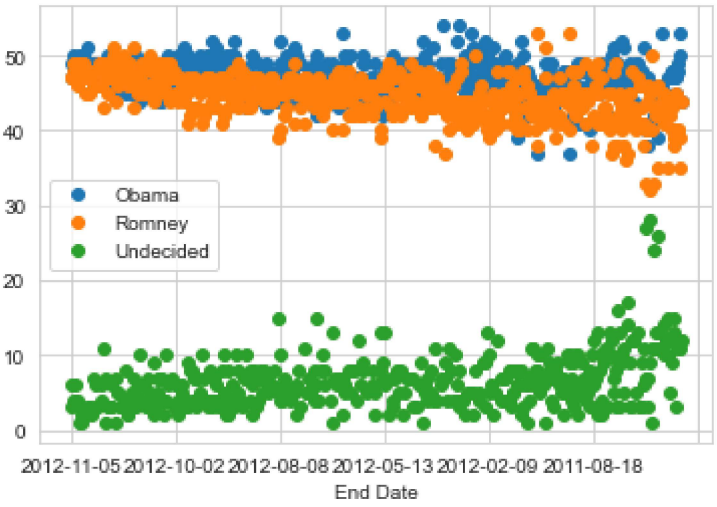
In [23]:
```
1 poll_df.head()
2
```

Out[23]:

	Pollster	Start Date	End Date	Entry Date/Time (ET)	Number of Observations	Population	Mode	Obam
0	Politico/GWU/Battleground	2012-11-04	2012-11-05	2012-11-06T08:40:26Z	1000.0	Likely Voters	Live Phone	47.
1	YouGov/Economist	2012-11-03	2012-11-05	2012-11-26T15:31:23Z	740.0	Likely Voters	Internet	49.
2	Gravis Marketing	2012-11-03	2012-11-05	2012-11-06T09:22:02Z	872.0	Likely Voters	Automated Phone	48.
3	IBD/TIPP	2012-11-03	2012-11-05	2012-11-06T08:51:48Z	712.0	Likely Voters	Live Phone	50.
4	Rasmussen	2012-11-03	2012-11-05	2012-11-06T08:47:50Z	1500.0	Likely Voters	Automated Phone	48.

In [25]:
```
1 poll_df.plot(x='End Date',y=['Obama','Romney','Undecided'],linestyle='',mark
```

Out[25]: <AxesSubplot:xlabel='End Date'>



In [27]:
```
1 from datetime import datetime
```

In [28]:
```
1 poll_df['Difference']=(poll_df.Obama-poll_df.Romney)/100
```

In [29]:
```
1 poll_df.head()
```

Out[29]:

	Pollster	Start Date	End Date	Entry Date/Time (ET)	Number of Observations	Population	Mode	Obam
0	Politico/GWU/Battleground	2012-11-04	2012-11-05	2012-11-06T08:40:26Z	1000.0	Likely Voters	Live Phone	47.
1	YouGov/Economist	2012-11-03	2012-11-05	2012-11-26T15:31:23Z	740.0	Likely Voters	Internet	49.
2	Gravis Marketing	2012-11-03	2012-11-05	2012-11-06T09:22:02Z	872.0	Likely Voters	Automated Phone	48.
3	IBD/TIPP	2012-11-03	2012-11-05	2012-11-06T08:51:48Z	712.0	Likely Voters	Live Phone	50.
4	Rasmussen	2012-11-03	2012-11-05	2012-11-06T08:47:50Z	1500.0	Likely Voters	Automated Phone	48.

In [31]:
```
1 poll_df=poll_df.groupby(['Start Date'],as_index=False).mean()
```

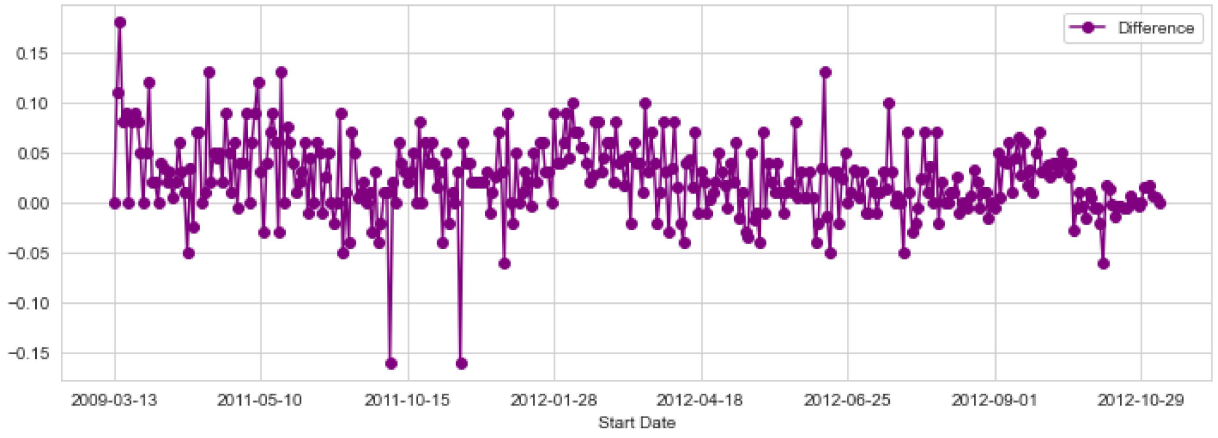
In [32]:
```
1 poll_df.head()
```

Out[32]:

	Start Date	Number of Observations	Obama	Romney	Undecided	Other	Question Text	Question Iteration	Difference
0	2009-03-13	1403.0	44.0	44.0	12.0	NaN	NaN	1.0	0.00
1	2009-04-17	686.0	50.0	39.0	11.0	NaN	NaN	1.0	0.11
2	2009-05-14	1000.0	53.0	35.0	12.0	NaN	NaN	1.0	0.18
3	2009-06-12	638.0	48.0	40.0	12.0	NaN	NaN	1.0	0.08
4	2009-07-15	577.0	49.0	40.0	11.0	NaN	NaN	1.0	0.09

In [33]:
```
1 poll_df.plot('Start Date','Difference',figsize=(12,4),marker='o',linestyle='
```

Out[33]:
```
<AxesSubplot:xlabel='Start Date'>
```

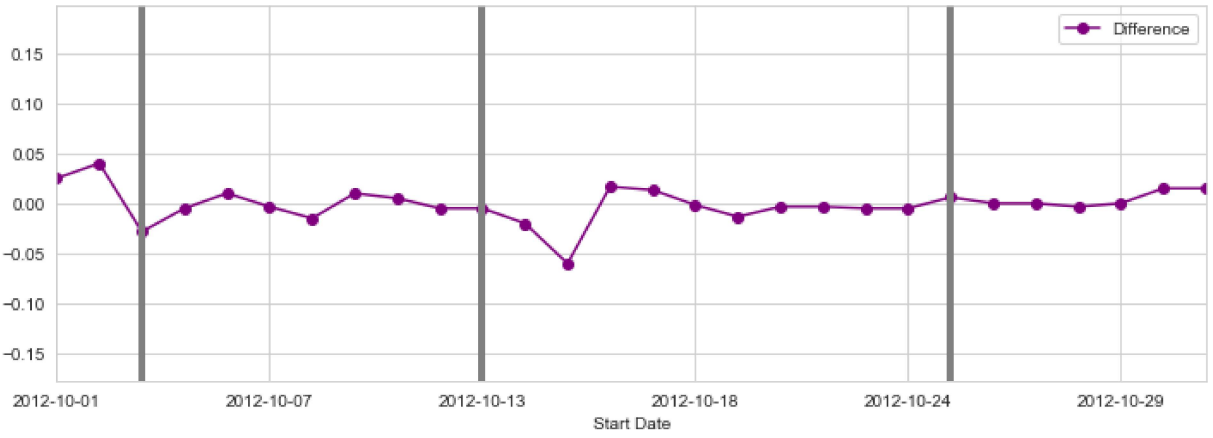


In [34]:
```
1  row_in=0
2  xlimit=[]
3  for date in poll_df['Start Date']:
4      if date[0:7]=='2012-10':
5          xlimit.append(row_in)
6          row_in+=1
7      else:
8          row_in+=1
9
10 print(min(xlimit))
11 print(max(xlimit))
```

```
325
352
```

In [35]:
```
1 poll_df.plot('Start Date','Difference',figsize=(12,4),marker='o',linestyle='
2 plt.axvline(x=325+2,linewidth=4,color='grey')
3 plt.axvline(x=325+10,linewidth=4,color='grey')
4 plt.axvline(x=325+21,linewidth=4,color='grey')
```

Out[35]: <matplotlib.lines.Line2D at 0x1cf205898d0>



In [36]:
```
1 pwd
```

Out[36]: 'C:\\Users\\Dell'

In [38]:
```
1 donor_df=pd.read_csv(r'C:\Users\Dell\Downloads\Election_Donor_Data.csv')
2 donor_df.info()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_2344\4189936121.py:1: DtypeWarning:
Columns (6) have mixed types. Specify dtype option on import or set low_memory=
False.
  donor_df=pd.read_csv(r'C:\Users\Dell\Downloads\Election_Donor_Data.csv')

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001731 entries, 0 to 1001730
Data columns (total 16 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   cmte_id            1001731 non-null  object
 1   cand_id            1001731 non-null  object
 2   cand_nm            1001731 non-null  object
 3   contbr_nm          1001731 non-null  object
 4   contbr_city        1001712 non-null  object
 5   contbr_st          1001727 non-null  object
 6   contbr_zip         1001620 non-null  object
 7   contbr_employer    988002 non-null   object
 8   contbr_occupation  993301 non-null   object
 9   contb_receipt_amt  1001731 non-null  float64
 10  contb_receipt_dt   1001731 non-null  object
 11  receipt_desc       14166 non-null    object
 12  memo_cd            92482 non-null    object
 13  memo_text          97770 non-null    object
 14  form_tp            1001731 non-null  object
 15  file_num           1001731 non-null  int64
dtypes: float64(1), int64(1), object(14)
memory usage: 122.3+ MB
```

In [39]:
```
1 donor_df.head()
```

Out[39]:

	cmte_id	cand_id	cand_nm	contbr_nm	contbr_city	contbr_st	contbr_zip	contbr_
0	C00410118	P20002978	Bachmann, Michelle	HARVEY, WILLIAM	MOBILE	AL	366010290.0	
1	C00410118	P20002978	Bachmann, Michelle	HARVEY, WILLIAM	MOBILE	AL	366010290.0	
2	C00410118	P20002978	Bachmann, Michelle	SMITH, LANIER	LANETT	AL	368633403.0	INFO REC
3	C00410118	P20002978	Bachmann, Michelle	BLEVINS, DARONDA	PIGGOTT	AR	724548253.0	
4	C00410118	P20002978	Bachmann, Michelle	WARDENBURG, HAROLD	HOT SPRINGS NATION	AR	719016467.0	

In [40]:
```
1 donor_df['contb_receipt_amt'].value_counts()
```

Out[40]:
```
 100.00     178188
 50.00      137584
 25.00      110345
 250.00      91182
 500.00      57984
             ...
 386.10          1
-113.40          1
 1385.00         1
 43.98           1
 2408.79         1
Name: contb_receipt_amt, Length: 8079, dtype: int64
```

In [41]:
```
1 donor_mean=donor_df['contb_receipt_amt'].mean()
2 donor_std=donor_df['contb_receipt_amt'].std()
```

In [42]:
```
1 print('the average donation %.2f with a std %.2f'%(donor_mean,donor_std))
```

```
the average donation 298.24 with a std 3749.67
```

In [45]:
```
1 top_donor=donor_df['contb_receipt_amt'].copy()
2 top_donor.sort_values()
3 top_donor
```

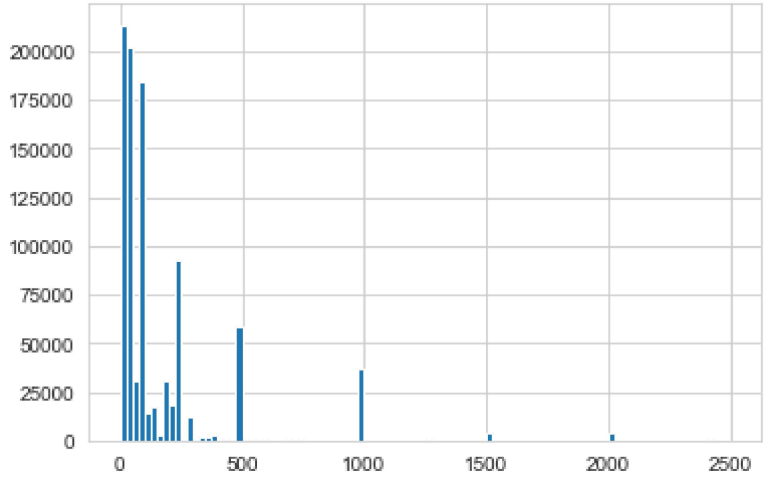
Out[45]:
```
0           250.0
1            50.0
2           250.0
3           250.0
4           300.0
            ...
1001726    5000.0
1001727    2500.0
1001728     500.0
1001729     500.0
1001730    2500.0
Name: contb_receipt_amt, Length: 1001731, dtype: float64
```

In [46]:
```
1 top_donor=top_donor[top_donor>0]
2 top_donor.sort_values()
3 top_donor.value_counts().head()
```

Out[46]:
```
100.0    178188
50.0     137584
25.0     110345
250.0     91182
500.0     57984
Name: contb_receipt_amt, dtype: int64
```

In [50]:
```
com_dom=top_donor[top_donor<2500]
com_dom.hist(bins=100)
```

Out[50]: <AxesSubplot:>



In [51]:
```
candidates=donor_df.cand_nm.unique()
candidates
```

Out[51]:
```
array(['Bachmann, Michelle', 'Romney, Mitt', 'Obama, Barack',
       "Roemer, Charles E. 'Buddy' III", 'Pawlenty, Timothy',
       'Johnson, Gary Earl', 'Paul, Ron', 'Santorum, Rick',
       'Cain, Herman', 'Gingrich, Newt', 'McCotter, Thaddeus G',
       'Huntsman, Jon', 'Perry, Rick'], dtype=object)
```

In [52]:
```
party_map = {'Bachmann, Michelle': 'Republican',
             'Cain, Herman': 'Republican',
             'Gingrich, Newt': 'Republican',
             'Huntsman, Jon': 'Republican',
             'Johnson, Gary Earl': 'Republican',
             'McCotter, Thaddeus G': 'Republican',
             'Obama, Barack': 'Democrat',
             'Paul, Ron': 'Republican',
             'Pawlenty, Timothy': 'Republican',
             'Perry, Rick': 'Republican',
             "Roemer, Charles E. 'Buddy' III": 'Republican',
             'Romney, Mitt': 'Republican',
             'Santorum, Rick': 'Republican'}
donor_df['Party']=donor_df.cand_nm.map(party_map)
```

In [53]:
```
donor_df=donor_df[donor_df.contb_receipt_amt>0]
```

In [54]:
```
1 donor_df.head()
```

Out[54]:

	cmte_id	cand_id	cand_nm	contbr_nm	contbr_city	contbr_st	contbr_zip	contbr_
0	C00410118	P20002978	Bachmann, Michelle	HARVEY, WILLIAM	MOBILE	AL	366010290.0	
1	C00410118	P20002978	Bachmann, Michelle	HARVEY, WILLIAM	MOBILE	AL	366010290.0	
2	C00410118	P20002978	Bachmann, Michelle	SMITH, LANIER	LANETT	AL	368633403.0	INFO REC
3	C00410118	P20002978	Bachmann, Michelle	BLEVINS, DARONDA	PIGGOTT	AR	724548253.0	
4	C00410118	P20002978	Bachmann, Michelle	WARDENBURG, HAROLD	HOT SPRINGS NATION	AR	719016467.0	

In [55]:
```
1 donor_df.groupby('cand_nm')['contb_receipt_amt'].count()
```

Out[55]:
```
cand_nm
Bachmann, Michelle                13082
Cain, Herman                      20052
Gingrich, Newt                    46883
Huntsman, Jon                      4066
Johnson, Gary Earl                 1234
McCotter, Thaddeus G                 73
Obama, Barack                    589127
Paul, Ron                        143161
Pawlenty, Timothy                  3844
Perry, Rick                       12709
Roemer, Charles E. 'Buddy' III     5844
Romney, Mitt                     105155
Santorum, Rick                    46245
Name: contb_receipt_amt, dtype: int64
```

In [57]:
```
1 donor_df.groupby('cand_nm')['contb_receipt_amt'].sum()
```

Out[57]:
```
cand_nm
Bachmann, Michelle                2.711439e+06
Cain, Herman                      7.101082e+06
Gingrich, Newt                    1.283277e+07
Huntsman, Jon                     3.330373e+06
Johnson, Gary Earl                5.669616e+05
McCotter, Thaddeus G              3.903000e+04
Obama, Barack                     1.358774e+08
Paul, Ron                         2.100962e+07
Pawlenty, Timothy                 6.004819e+06
Perry, Rick                       2.030575e+07
Roemer, Charles E. 'Buddy' III    3.730099e+05
Romney, Mitt                      8.833591e+07
Santorum, Rick                    1.104316e+07
Name: contb_receipt_amt, dtype: float64
```

In [58]:
```
cand_amount=donor_df.groupby('cand_nm')['contb_receipt_amt'].sum()
```

In [60]:
```
i=0
for don in cand_amount:
    print('the candidate %s raise %.f of dollars'%(cand_amount.index[i],don))
    print('\n')
    i+=1

```

```
the candidate Bachmann, Michelle raise 2711439 of dollars


the candidate Cain, Herman raise 7101082 of dollars


the candidate Gingrich, Newt raise 12832770 of dollars


the candidate Huntsman, Jon raise 3330373 of dollars


the candidate Johnson, Gary Earl raise 566962 of dollars


the candidate McCotter, Thaddeus G raise 39030 of dollars


the candidate Obama, Barack raise 135877427 of dollars


the candidate Paul, Ron raise 21009620 of dollars


the candidate Pawlenty, Timothy raise 6004819 of dollars


the candidate Perry, Rick raise 20305754 of dollars


the candidate Roemer, Charles E. 'Buddy' III raise 373010 of dollars


the candidate Romney, Mitt raise 88335908 of dollars


the candidate Santorum, Rick raise 11043159 of dollars
```

In [61]:
```
1 cand_amount.plot(kind='bar')
```

Out[61]: <AxesSubplot:xlabel='cand_nm'>