Notebook

Evaluate the dependence of p(win) on differences in eigenvector centrality

In [1]:

```
import os
os.chdir('C:\Users\Scott\Dropbox (Personal)\Frisbee\Weather/frisbee_weather')
import matplotlib.pyplot as plt
%matplotlib inline
import json
import random
import numpy as np
import predict_usau as pu
```

In [2]:

```
def calc_success(s):
    # s is a predict_usau.season object
    success = []
    rankdiff = []
    for g in s.games.itervalues():
        if s.teams[g['teams'][0]]['ranking'] < s.teams[g['teams'][1]]['ranking']:
            predicted_winner_id = g['teams'][0]
            predicted_loser_id = g['teams'][1]
            if g['score'][0] - g['score'][1] > 0:
                was_prediction_correct = 1
            else:
                was_prediction_correct = 0
        else:
            predicted_winner_id = g['teams'][1]
            predicted_loser_id = g['teams'][0]
            if g['score'][1] - g['score'][0] > 0:
                was_prediction_correct = 1
            else:
                was_prediction_correct = 0
        rankdiff.append(s.teams[predicted_winner_id]['eigenvector_centrality'] - s.teams[pre
        success.append(was_prediction_correct)
    return rankdiff,success
```

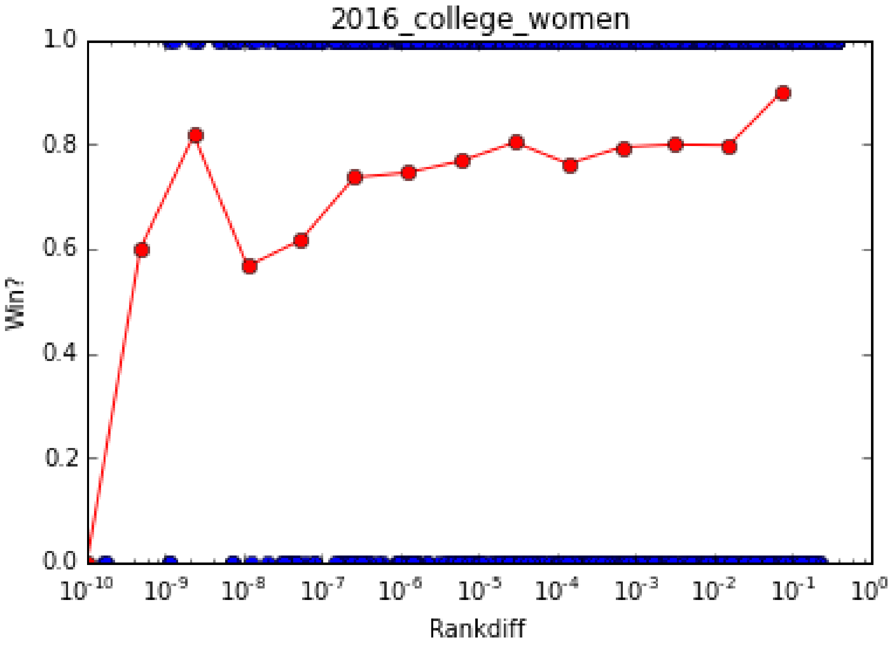
In [108]:

```
def plot_success(fname,Nbins=20,XLim=(10**-10,1.0)):

    s = pu.season(json.load(open(fname + '.json','r')))
    rankdiff,success = calc_success(s)
    for rd,suc in zip(rankdiff,success):
        plt.plot(rd,suc,'bo')


    p_win = []
    bin_centers = []
    bin_min = np.log10(max(min(rankdiff),10**-10))
    bin_max = np.log10(max(rankdiff))
    bin_starts = np.logspace(bin_min, bin_max,Nbins)
    log_binhalfwidth = 0.5 * (np.log10(bin_starts[1]) - np.log10(bin_starts[0]))
    for i in range(len(bin_starts)-1):
        tuples = [(rd,suc) for rd,suc in zip(rankdiff,success) if  bin_starts[i] <= rd < bin
        try:
            rd,suc = zip(*tuples)
        except:
            continue
        p_win.append( float(sum(suc)) / float(len(suc)) )
        bin_centers.append(bin_starts[i] )
    print len(bin_centers)
    plt.plot(bin_centers,p_win,'ro-')
    plt.xlabel('Rankdiff')
    plt.ylabel('Win?')
    plt.xscale('log')
    plt.xlim(XLim)
    plt.title(fname)
    plt.show()
```

In [105]:

```
plot_success('2016_college_women',Nbins=15)
```

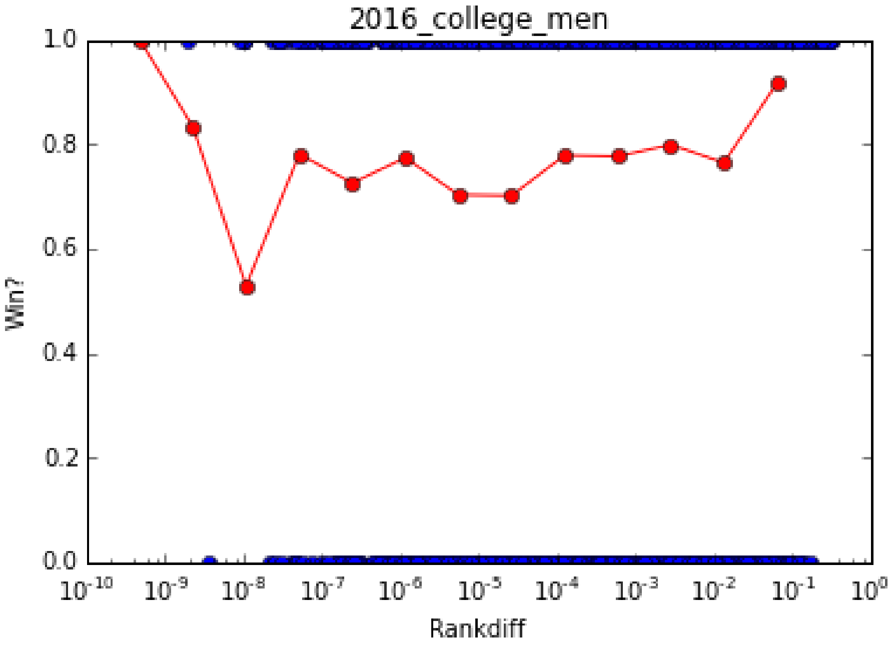
```
14
```



In [109]:

```
plot_success('2016_college_men',Nbins=15)
```

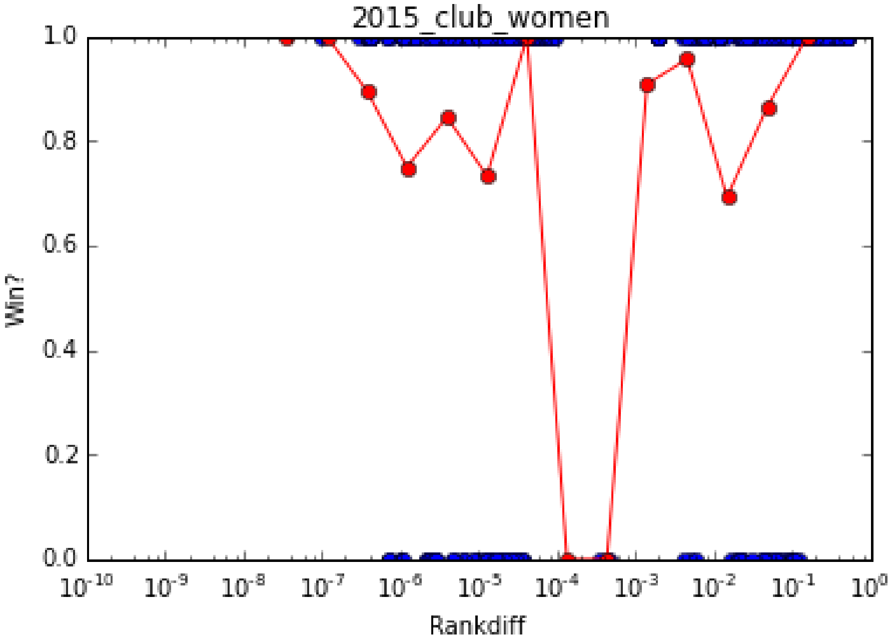
```
13
```



In [111]:

```
plot_success('2015_club_women',Nbins=20)
```

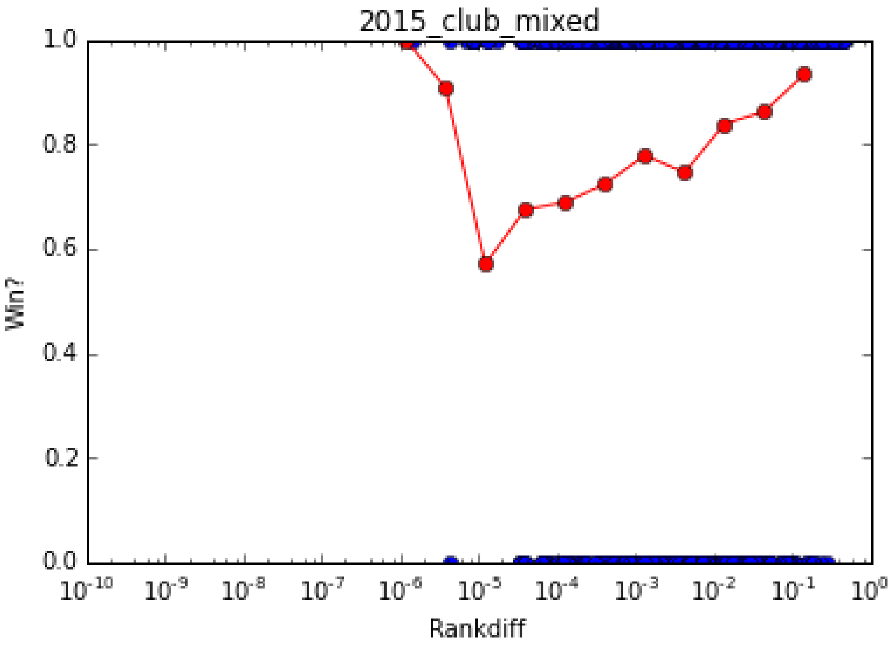
```
14
```



In [112]:

```
plot_success('2015_club_mixed')
```

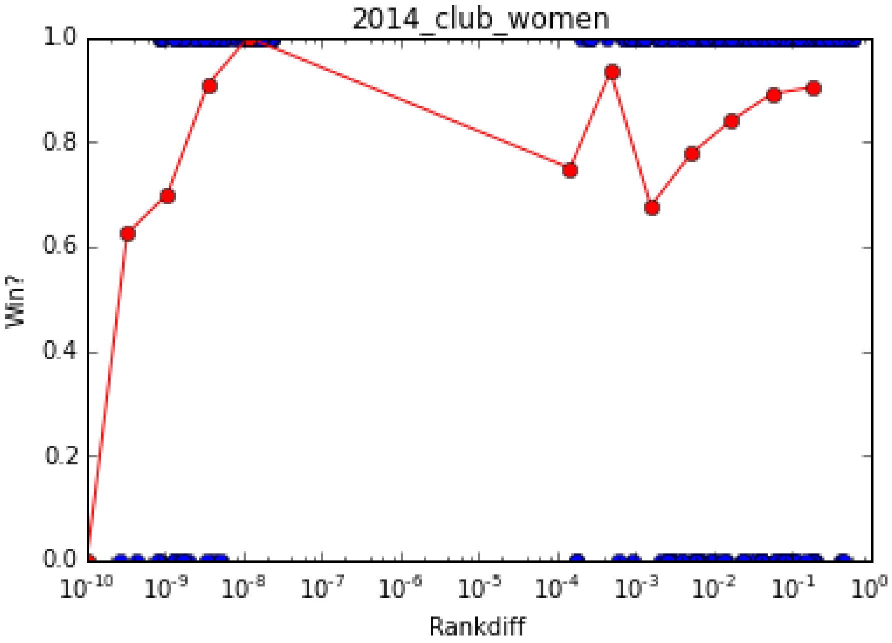
```
11
```



In [113]:

```
plot_success('2014_club_women')
```

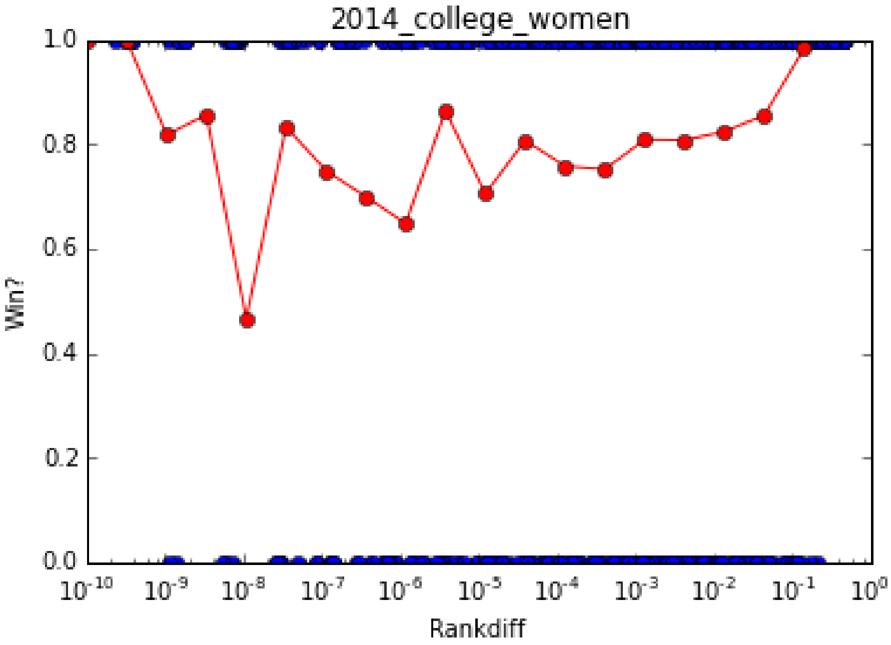
```
12
```



In [114]:

```
plot_success('2014_college_women')
```

```
19
```


2014_college_women
1.0
0.8
0.6
0.4
0.2
0.0
Win?
10^{-10}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
Rankdiff

In []: