Check stability of eigenvector centrality wrt fewer games played. Randomly select a certain percentage of games to have been played, then see how robust rankings are.

Future idea: remove all games from a certain tournament, see how specific tournaments impact rankings.

In [63]:
```
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
```

In [51]:
```
import predict_usau as pu
reload(pu)
```

Out[51]:
```
<module 'predict_usau' from 'predict_usau.pyc'>
```

In [47]:
```
d = json.load(open('2016_college_women.json','r'))
```

In [60]:
```
Npoints = 10
Ngames_total = len(d['games'].keys())
season_100 = pu.season(d)
rank_vector_100 = np.array([team['ranking'] for team in season_100.teams.itervalues()]
dist = []
for percent in np.linspace(0.95,0.5):
    Ngames = int(np.floor(percent * Ngames_total))
    distance_from_100 = []
    for i in range(0,Npoints):
        tmp_keys = random.sample(d['games'].keys(),Ngames)
        new_games = {key:d['games'][key] for key in tmp_keys}
        tmp_season = pu.season({'games':new_games,
                    'tournaments':d['tournaments'],
                    'teams':d['teams']})
        rank_vector = np.array([team['ranking'] for team in tmp_season.teams.itervalue
        distance_from_100.append(np.linalg.norm(rank_vector_100 - rank_vector))
    dist.append(np.mean(distance_from_100))
```

In [66]:
```
plt.plot(np.linspace(0.95,0.5), dist)
plt.xlabel('Percent Games remaining')
plt.ylabel('Avg. ranking euclidian distance')
plt.show()
```