Eigen-ranking project

For this second programming project, you'll explore some real world data (probably sports data) and rank it using the eigen-ranking procedure outlined in our eigen-ranking demo in HTML and Notebook formats. Your mission: find some network data with at least seven nodes and directed or undirected connections between them. Your network should be *strongly connected* (or close) so that our theorem on eigen-ranking is applicable. Enter the resulting matrix (manual entry is fine, if your network isn't too big), compute the dominant eigenvector and display the resulting ranking.

Note: This technique is broadly applicable. I suggest sports data simply because there's lots of it that's easy to come by and it's not too hard to find data of manageable size. For example, you might Google something like "`2014 Big Ten football season`". You'll find quite a few pages from Wikipedia to ESPN that describe the results of that season. You'll learn many fun facts along the way - like Ohio State was the national champion that year! Among the more curious facts that you'll learn is that the Big Ten has fourteen teams that are divided into East and West divisions - each with seven teams. Thus, the Big Ten East by itself would form a nice network to analyze - particularly since each team plays every other team.

If you absolutely hate sports, well these ideas are applicable to other topics. Google's original pagerank algorithm is based on the exact same idea. You can evaluate the importance of research papers by analyzing the references between them. The importance of airline hubs can be ranked in this way (with an undirected graph). Let me know if you have an alternative topic that you *really* want to explore and if you have a decent clue where to get the data.

In any event, you should work this out in a well formatted Jupyter notebook and email that to me by next Friday, March 4.