Project 4 Network Visualizations

Table of Contents

Graph Theory Definitions

Vertex

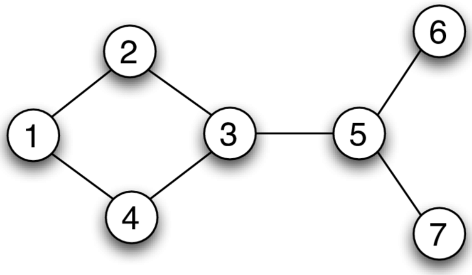
A vertex is a node of a graph. A vertex usually contains functions, and other information that gets moved around the graph. An example would be the three points of a triangle if we had a triangle graph.

Edge

An edge is a connection between two vertexes. Edges may indicate transformations that happen to information being transformed as it moves around the graph. An example of an edge would be the sides of a triangle.

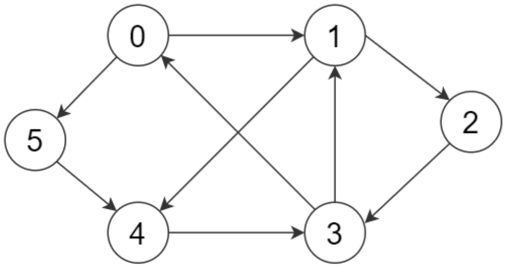
Undirected Graph

An undirected graph is a graph in which the flow of information between vertices is unrestricted via direction. An example of an undirected graph would be a walking path. Individuals can walk either direction on the path.



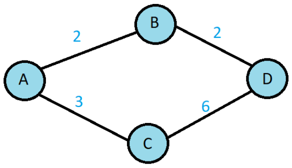
Directed Graph

A directed graph is a graph in which the flow of information between vertices is restricted via direction. An example of a directed graph would be a sequence of one-way roads.



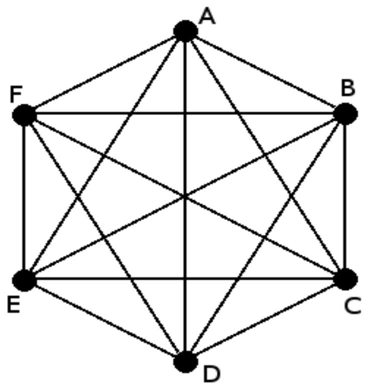
Weighted Graph

A weighted graph is a graph in which the flow of information between vertices is restricted via capacity. Edges are labelled with number that indicate the edges capacity. An example of a weighted graph is our road system. Each road can fit a certain number of vehicles on them.



Complete Graph

A complete graph is a graph in which all vertices are connected to each other via an edge.



In-degree

The in-degree is the number of edges that are directed into a given vertex.

Out-degree

The out-degree is number of edges that are directed out of a given vertex.

Adjacency Matrix

An adjacency matrix is a square matrix used to represent a finite graph. The rows and columns represent a vertex in the graph and connects are indicated via the value of the intersection of a row and a column.

0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

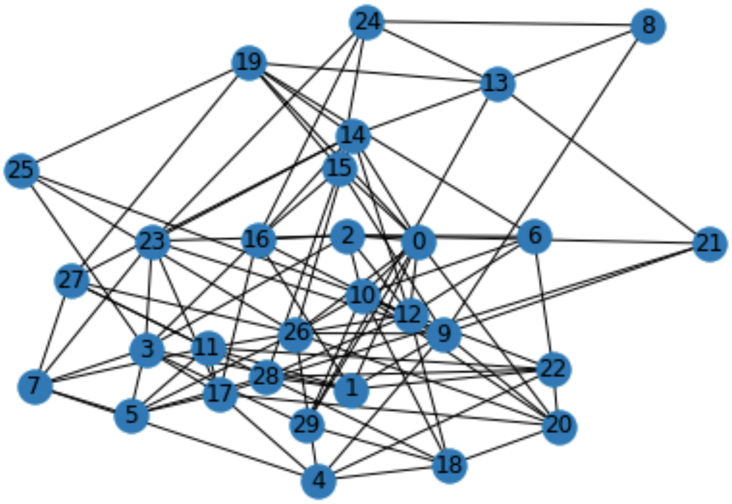
Edge-set Representation

An edge set representation is a list of all the edges in the graph.

Edge List
1-6
2-1
2-5

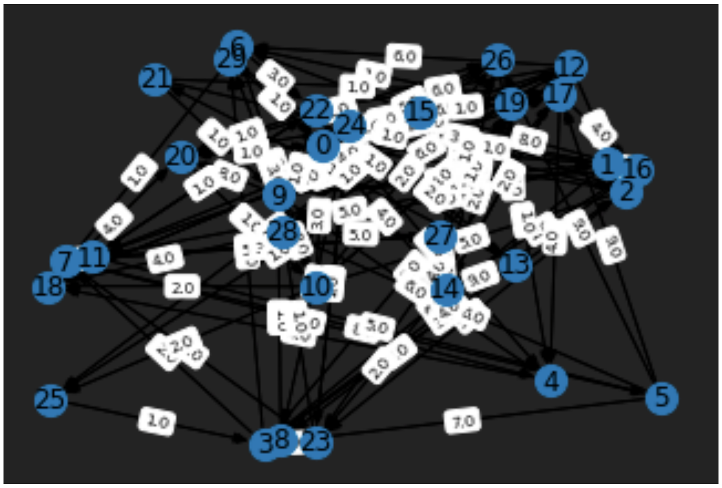
Graphs Visualizations

Graph 1



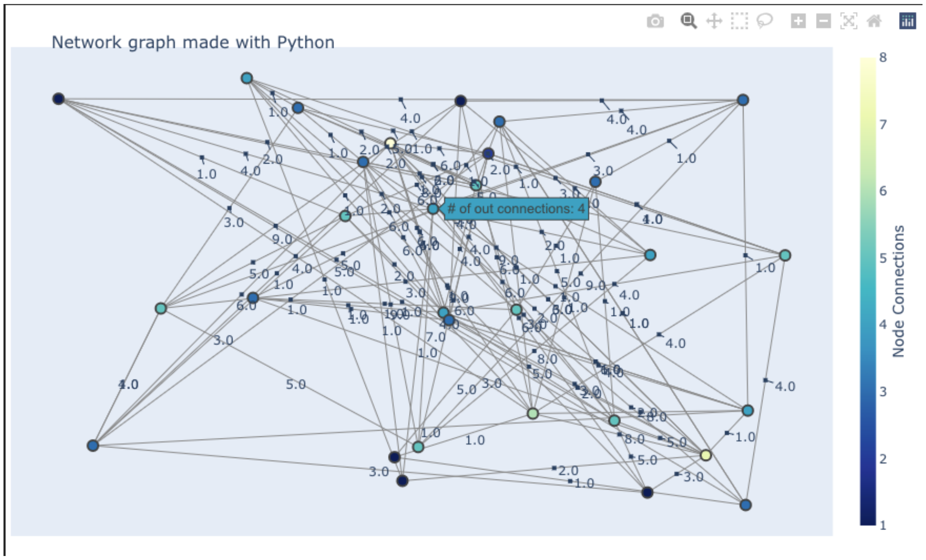
This is the most basic visualization that can be made with a network graph. I simply drew the graph from the adjacency matrix. We can see the node numbers and the connections between. Them. There are weights to the edges that are not seen here. Let's visualize that as well.

Graph 2



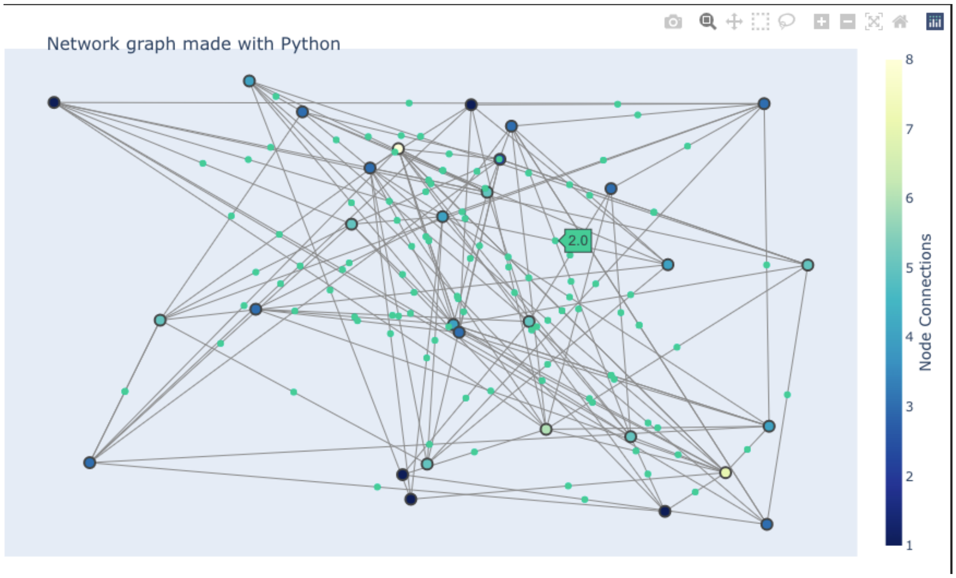
As we can see, this is a nightmare to look at, but we can see the weights, we also drew the directions of the edges here. However, it is hard to see. Here I wanted to move away from networkx's draw and try plotly.

Graph 3



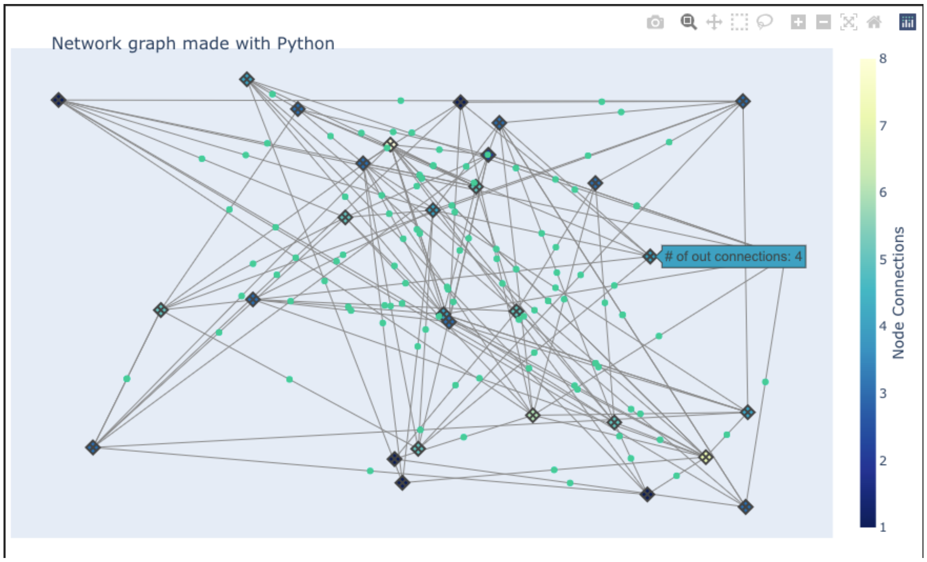
Now the graph is interactable and slightly less cluttered. Hovering over the nodes will show the number of outbound connections. The color bar on the side gives the user some idea as to what the number of outbound connections is for a given node. Weights are still hard to distinguish. Let's see if there is anything we can do about it.

Graph 4



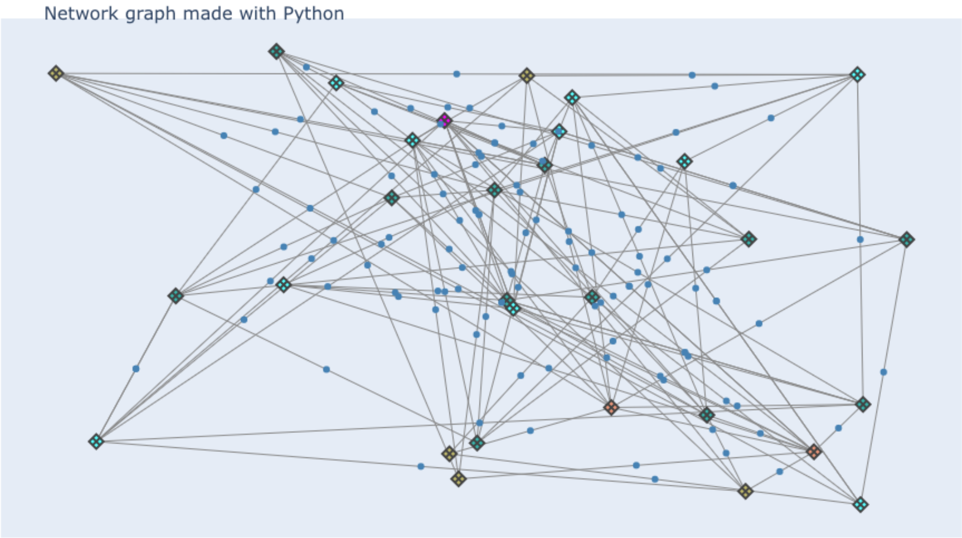
Here we successfully decluttered the weight. It is now interactable and the user can hover over the symbol on the line to get those lines weight. However, now the markers look like nodes, and it is hard to distinguish. Let's play around with the marker symbols.

Graph 5



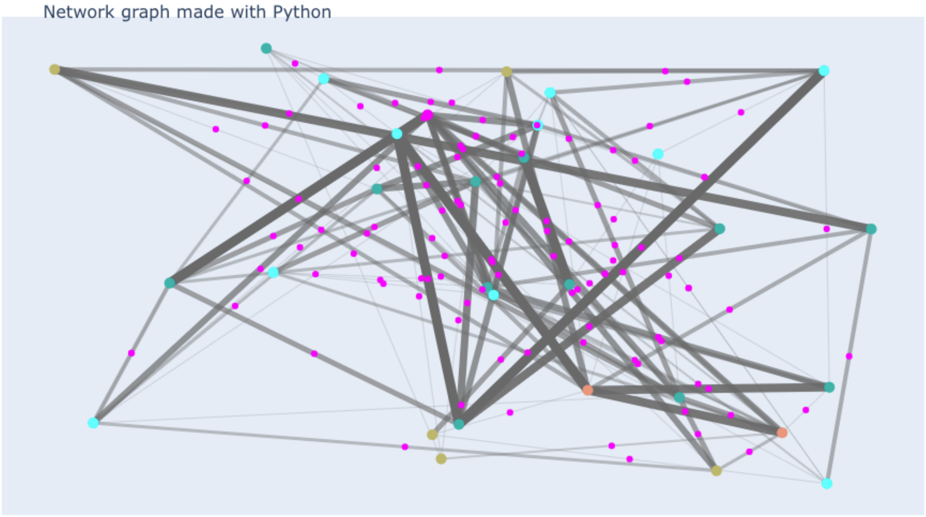
Now we have different symbols to differentiate between nodes and weights. There are still a few things I want to try out here, specifically with color.

Graph 6



Here we removed the color spectrum in place of different colors. This is the final graph for our set of improvements. Additional improvements would be the inclusion of axis titles. There is one more graph that I want to make. It will look a bit ridiculous.

Graph 7



Here we adjusted the width of the edges based on weight, the same is true for the edge opacity. Setting up this graph was a bit tricky, so we lost the marker updates for now. This graph is not that readable, but I wanted to try out additional plotly features.