two vertices. Usually this occurs because the edges are defining different kinds of relationships. Travel routes are common examples of multigraphs, where each edge represents a different carrier. For example, Figure 2.3 is a graph of flights between the San Francisco (SFO), Philadelphia (PHL) and Tucson (TUS) airports based on a data set from December 2010. The graph is layered onto a map of the United States. Philadelphia to Tucson is not a common route and is only offered by one carrier in one direction, while there are multiple carriers operating in both directions between Philadelphia and San Francisco and between San Francisco and Tucson.

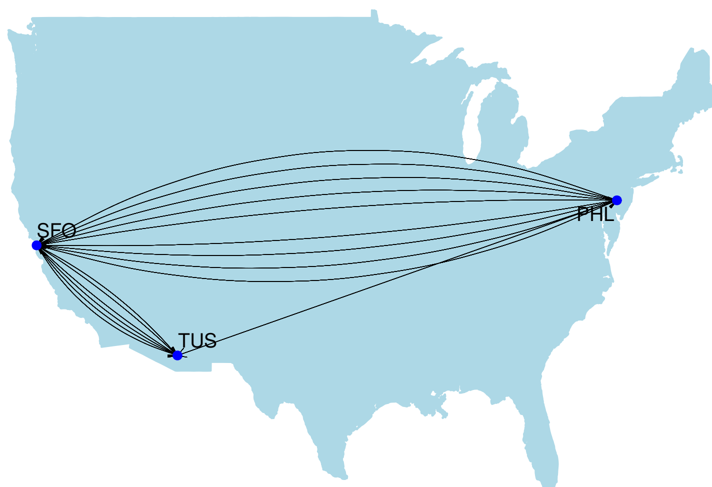

FIGURE 2.3: Carrier routes operating between three US airports in December 2010

Multigraphs are also commonly used when individuals or entities can be related to each other in different ways. For example, imagine if we were to combine our G_{work} and G_{manage} graphs from Section 2.1.1 into one single directed graph depicting both 'worked with' and 'manages' relationships. It might look like Figure 2.4.

Many large graphs used in practice are multigraphs, as they are built to capture many different types of relationships between many different types of entities. For example, a graph of an organizational network might contain vertices which represent individuals, organizational units and knowledge areas. Multiple different types of relationships could exist between individuals (such as 'worked with', 'manages', 'published with'), between individuals and organizational units (such as 'member of' or 'leader of'), between individuals and knowledge areas (such as 'affiliated with' or 'expert in') and all sorts of other possibilities.

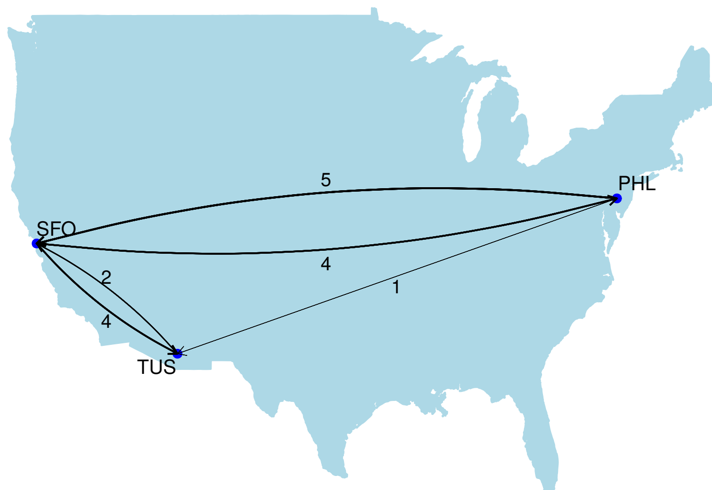

FIGURE 2.10: Simplifying the flights graph using weighted edges to represent the number of carriers on each route. Edge thickness represents weight.

An **edgelist** is the edge set E in our graph $G = (V, E)$. If we don't care about *isolates*[1], then our vertex set V can be derived directly from E. This means that the edgelist is all that is needed to build a graph provided you are happy to ignore isolates. It is common that an analyst is happy to ignore isolates because we are often only interested in the connections or relationships in the data. Let's look at an example.

Recall our edge set E in the graph $G_{\mathrm{work}} = (V, E)$ from Section 2.1.1:

$$\begin{gathered}E = \{\mathrm{David} \longleftrightarrow \mathrm{Zubin}, \mathrm{David} \longleftrightarrow \mathrm{Suraya}, \mathrm{Suraya} \longleftrightarrow \mathrm{Jane},\\ \mathrm{Jane} \longleftrightarrow \mathrm{Zubin}, \mathrm{Jane} \longleftrightarrow \mathrm{Suraya}\}\end{gathered}$$

Since by definition each edge in E must be a pair of vertices from V, and since we are not concerned about isolates (in fact, we know they don't exist in this case), we can obtain the vertex set V by simply listing the unique vertices from the pairs in E. Therefore, we can construct V to be

$$V = \{\mathrm{David}, \mathrm{Suraya}, \mathrm{Jane}, \mathrm{Zubin}\}$$

[1] Isolates, also known as singletons, are vertices which are not connected to any other vertices. Many real-life graphs contain isolates which will often be removed in order to better focus on connected components.