Learning Module - Key concepts for graph analysis

The first part of this training will focus on understanding some of the key concepts required to undertake analysis on graphs. This module doesn't require Constellation to be used; the focus is on understanding concepts rather than methods.

There are four components:

- Why use a graph for analysis?
- Understanding the core graph elements: Nodes, Relationships, and Attributes, and what they look like on a graph.
- Using a Histogram to count Attributes of elements on the graph.
- Understanding Graph Properties, and counting them using a Histogram.

By the end of this module you should be able to explain the key elements listed above, and have an initial understanding of some questions that can be asked of a graph.

Why use a graph for analysis?

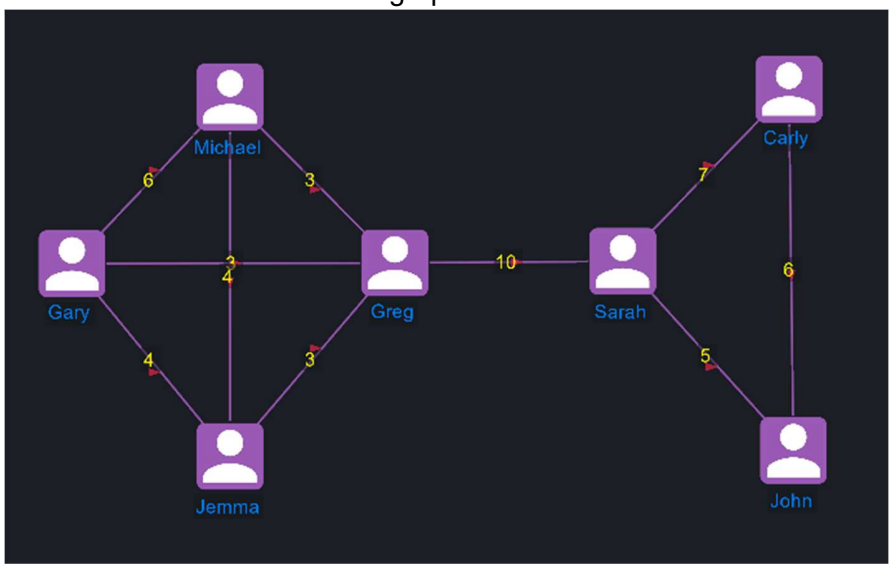
Looking at data displayed on a graph is a great way to visualise and intuitively understand relationships between different entities. While tables offer the value of being able to see a lot of detail in the data, when looking at networks of connected entities it is difficult to understand how various elements are related to one another.

Take the following table as an example, showing lunch meetings between different groups.

	A	B	C
1	Person 1	Meeting Type	Person 2
2	Greg	Lunch	Sarah
3	Greg	Lunch	Michael
4	Greg	Lunch	Gary
5	Gary	Lunch	Michael
6	Gary	Lunch	Jemma
7	Greg	Lunch	Jemma
8	Michael	Lunch	Jemma
9	Sarah	Lunch	John
10	Sarah	Lunch	Carly
11	Carly	Lunch	John
12	Greg	Lunch	Jemma
13	Greg	Lunch	Michael
14	Gary	Lunch	Michael
15	Greg	Lunch	Gary
16	Gary	Lunch	Jemma
17	Michael	Lunch	Jemma
18	Sarah	Lunch	Carly
19	Carly	Lunch	John
20	Sarah	Lunch	John
21	Greg	Lunch	Sarah
22	Sarah	Lunch	Carly
23	Carly	Lunch	John

Answering simple questions like 'has Gary had a lunch meeting with Carly?' already require additional filtering to answer. More complicated questions like 'which person has had lunch meetings with the most people?' are very difficult to determine without more complex tools like pivot tables.

Let's view that same data in a graph.



As you can see, the structure of the lunch groups is now very clear. This helps us to structure our later questions we may have about the network. There are also simple techniques available when analysing data on a graph that allow us to easily answer more complex questions of the networks like this one.

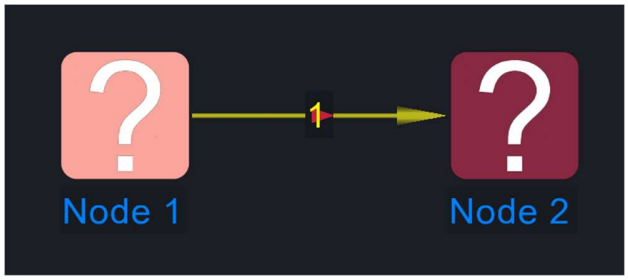
We'll use this lunch group network for the remainder of this Module.

Understanding Nodes, Relationships, and Attributes

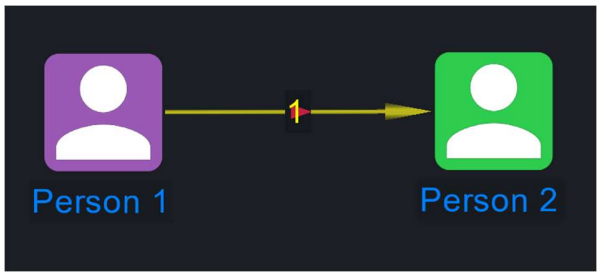
Looking at data displayed on a graph is a great way to visualise and intuitively understand relationships between different entities. There are three critical elements of a graph:

- Nodes: An entity. It's easy to think of entities as nouns; people, places, and things will all appear on a graph as a Node. In our table, these were the people in the 'person 1' and 'person 2' columns.
- Relationships: Essentially anything that links two nodes together. Relationships can be unidirectional, bidirectional, or directionless - it all depends on the type of relationship, which is usually described by the Attributes of the relationship. In many cases, it's easy to think of relationships as a verb (something a noun does). In our table, these were the 'lunch' values in the 'meeting type' column.
- Attributes: Attributes are the properties or descriptions that provide details about what makes up the Nodes and Relationships. Without Attributes, Nodes and Relationships are just nondescript blobs connected on a graph. It's easy to think of Attributes as adjectives (or adverbs) describing the nouns (and verbs).

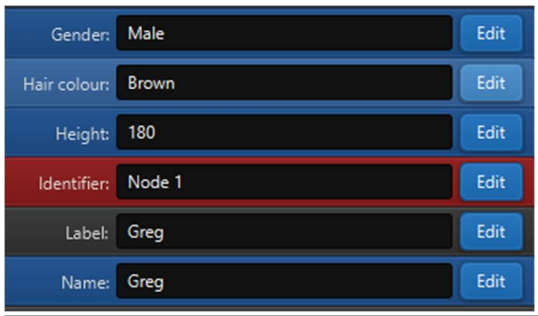
Below is an example of a graph with two nodes, with a relationship between them. The relationship is unidirectional, ie. it runs from Node 1 *to* Node 2.

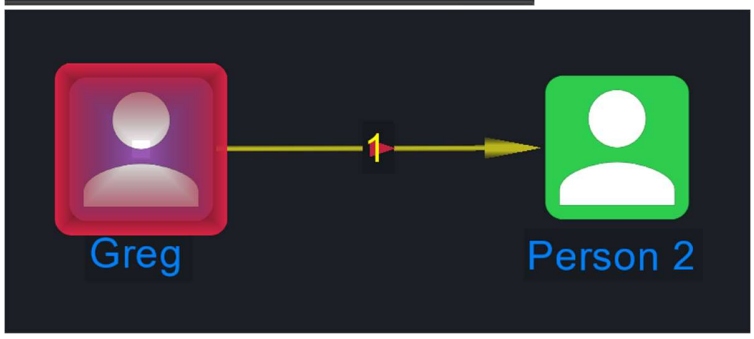


Without Attributes, we have no way to tell what the Nodes are, or what the nature of the Relationship between them is (other than the direction). Let's add some Attributes to turn our Nodes into people, by adjusting the 'Type' attribute.



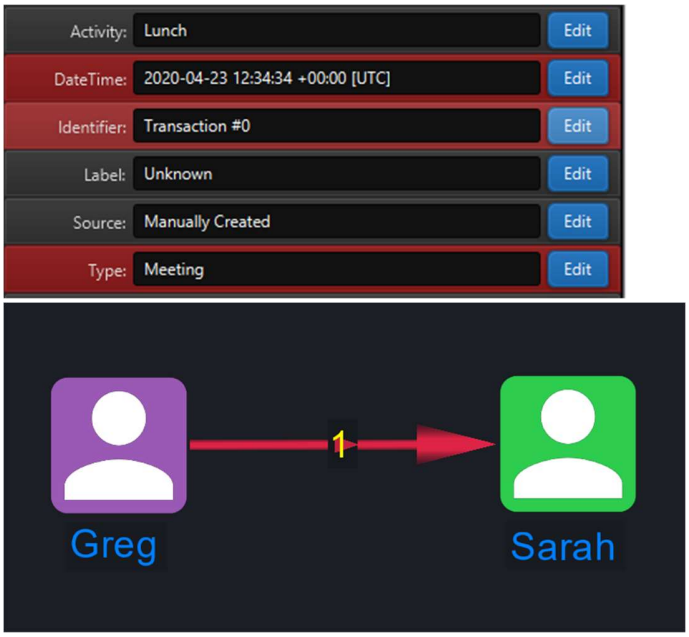
Great, now we know that the graph is displaying some relationship between two people. Nodes and Relationships can have many attributes, but if we started visualising them all on the graph it would become very confusing. To avoid this, we keep Attributes stored in tables that are attached to each Node and Transaction. Let's fill in some Attributes for Person 1, and update the graph.





You can ignore the colours on the Attributes in the table for now, but you can see the Attributes for Node 1, now known as Greg, have been stored and are visible when selecting the relevant Node (highlighted in red).

What about Attributes for Relationships? There are many ways to describe Relationships between Nodes. The important ones are often related to the ‘Type’ of the Relationship, and information around the ‘DateTime’ those relationships occur in, which gives us some temporal (ie. time-related) context. In this example, let’s say that Greg meets with Person 2 two for lunch on 23/4/2020. I’ll also update the details for Person 2.

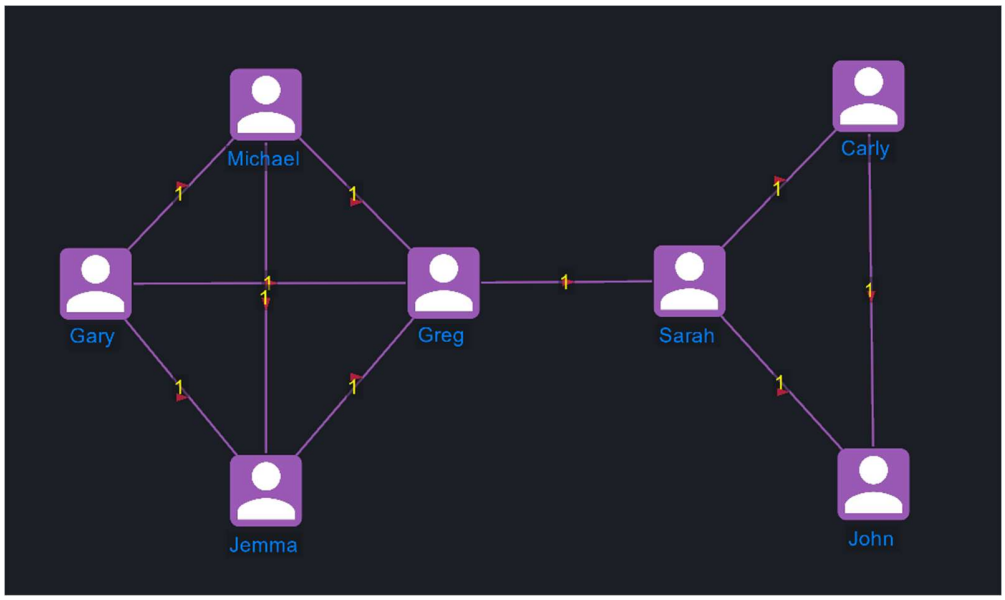


Now by selecting the Relationship, we can see that on 23/4/12 at around 1230, Greg had a meeting with Sarah for lunch.

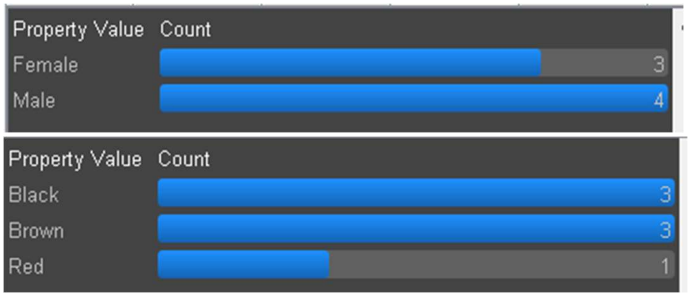
Of course, there are many other Relationship types that could exist between people - we could describe the nature of their relationship (eg. friends, colleagues, siblings) as well as the duration of their relationship (the first time and the last time we recorded them having that relationship with each other). We could be looking at a communication between them, such as an email invitation for lunch. Likewise, additional Attributes could describe both Greg and Sarah in greater detail, depending on what data we had available.

Using a Histogram to count Attributes

Now we'll look at a network of Nodes, rather than just two. In addition to meeting with each other for lunch, Greg and Sarah also have their own separate lunch groups that they meet with. This can easily be visualised on the graph.

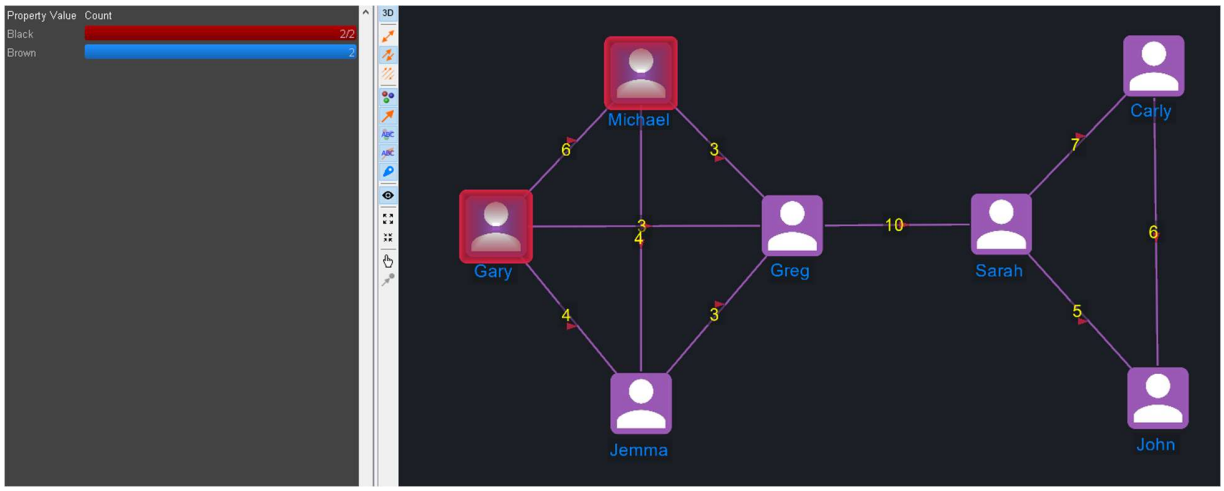


We might want to ask some questions about the frequency of different attributes within our network. In our case, we want to know how many people of each gender and hair colour are in the group. We could look at the attribute table for each individual Node on the graph, but this would be time consuming. Using a histogram, we can count the total number of each of those Attributes.



By 'Histogramming' on the Node Attributes of Gender and Hair Colour respectively, we can count them and see that there are three females and four males on the graph. Likewise there are three people with black hair, three with brown, and one with red.

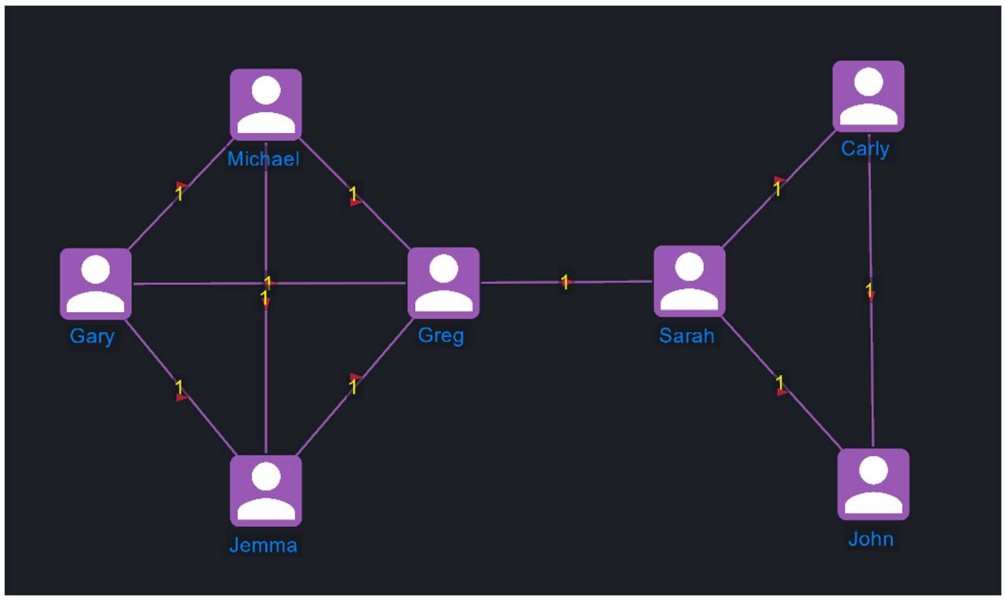
We might have a multi-layered question. For example, how many males on the graph have black hair? To answer this question, we need to use a Histogram technique called Filtering, where we take all Nodes with the first Attribute that we care about: gender = male, and filter on these results so that only the Attributes of Nodes with gender = male will be shown in the Histogram. We then use the Histogram to look at hair colour, and see the hair colour Attributes for only the male Nodes on the graph.



As we can see in the filtered Histogram, there are two males on the graph with black hair; the remaining two have brown hair.

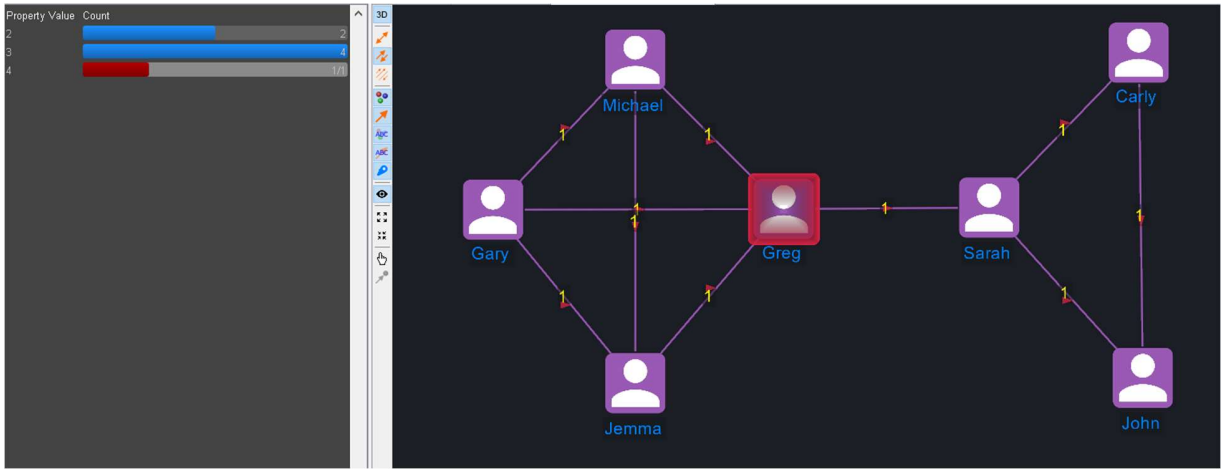
Graph Properties

While it would be complicated to understand this network intuitively from looking at the recorded data in a table, viewing the data on a graph lets us easily grasp the nature of the two distinct groups, that are connected through the relationship between Greg and Sarah.



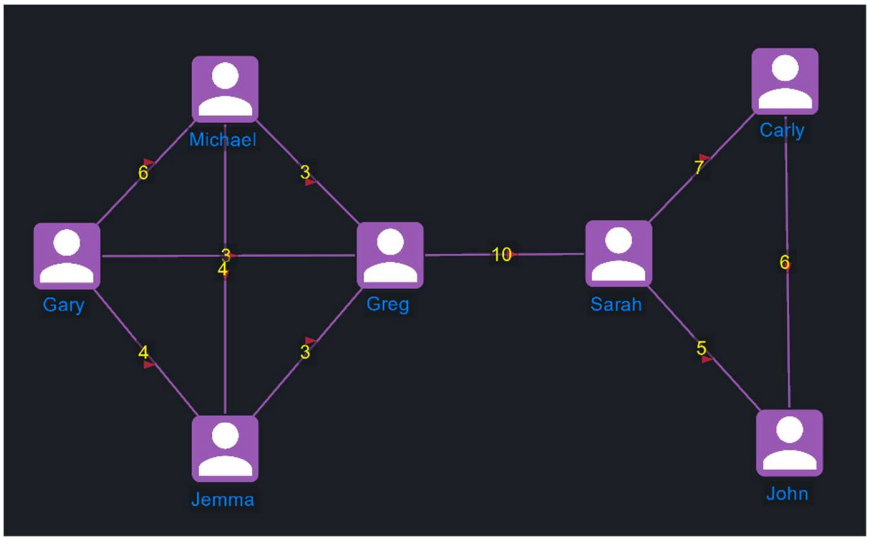
At this stage, we might want to ask some questions about our network. For example, which person meets with the most other people? In this example, we could simply use our eyes to

figure out the answer; in larger networks, this method quickly becomes untenable. Instead, we need to rely on Graph Properties. Similar to how Attributes describe Nodes and Relationships, Graph Properties describe the underlying structure of the graph: how many nodes there are, how many relationships are connected to different nodes, and how many nodes connect with other nodes. It may sound confusing, but we'll use our lunch groups as an example. When asking which person meets with the most other people for lunch, we are really asking the question: which Node is connected to the highest number of other nodes (in graph terminology: Neighbours)?

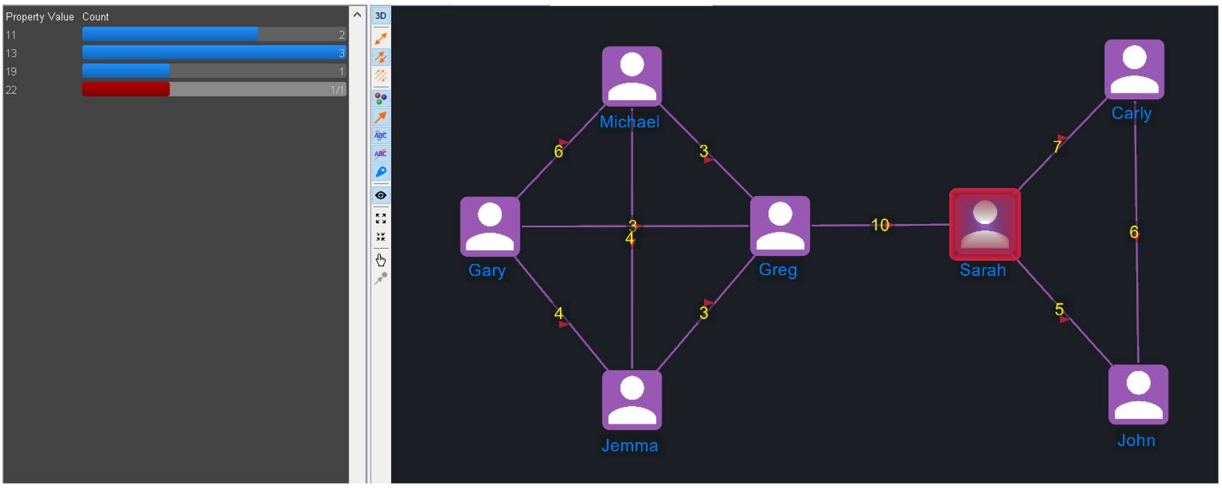


We've visualised the number of neighbours for each node on the graph using a Histogram to count the Graph Property of 'Neighbour Count'. Here, we can see that there are two Nodes that have two Neighbours (Carly and John), four Nodes with three Neighbours (Sarah, Michael, Gary, and Jemma), and one Node with four Neighbours (Greg, highlighted in red). By looking at the Graph Attributes, we can easily answer the question of who meets the most other people for lunch: Greg.

Lunch groups meet more often than a single time though, otherwise they wouldn't be very good lunch groups. Of course, not everyone in the group can attend every lunch meeting, so we've only recorded the dates that each person was at a lunch with someone else.



As the number of nodes increases, so too does the difficulty of answering who has attended the most lunches.We can use the Graph Properties to ask the question: which Node has the highest number of Relationships (or Transactions)?



In the screenshot, we've used the Histogram to count the Graph Property of 'Transaction Count'. In our example, there are two Nodes with 11 Relationships/Transactions (John and Jemma), three Nodes with 13 Relationships/Transactions (Carly, Michael, and Gary), one Node with 19 Relationships/Transactions(Greg), and one Node with 22 Relationships/Transactions (Sarah). The answer to the question of which person has had the most lunch meetings is: Sarah.

As we have seen, the type of question you want to ask will determine if you need to use the Histogram to count the Attributes of Nodes and Relationships, or to count the elements of the underlying Graph Properties.