Data Collection and Data Management

Network Questions

Let us say our research objectives include studying the trust ties among members of a terrorist group over time. Unfortunately, that may not be possible for a number of reasons.

- can we get the respondents to talk to us?
- what are we allowed/tolerated to ask? To what depth?
- Are our relational questions sensitive (in general or for some specific respondent)?
- Can our questions change the willingness/accurateness/honesty/competence of the answers of the responder?
- Questions are rarely one-dimensional. Can some of these factors influence the answers? Cultural context (e.g., economic relations), time (e.g., season, quarter, first-to-last responder, incomparable time-frames), varied data-collection methods (e.g., face-to-face versus online interviews).

Network Questions

The proper selection of the network questions and formats is critical to the success of any network study.

The structure of network questions greatly influences the validity and reliability of respondent answers due to such things as question clarity, burden, sensitivity, and cognitive demand.

Reminder: network questions are not simply asking about some attribute of the respondent or ego (e.g., age). They concern the web of relations of the responders, who may have an emotional response or tax their abilities to remember or recall aspects of their network relations and/or network behaviours.

Network Questions • Use case: Extreme Friendship

In a 4-year study at polar research stations, the researchers initially investigated the formation of friendships and the ability of individuals to assess potential friendships one day following the initial contact.

During a training break, the crew members, amounting to n people, were given a questionnaire asking them to rank the other members of the crew from 1 to $n-1$, with respect to how likely they were to form a friendship with each of the other members of the crew over the coming winter.

Immediately, several of the crew began to grumble and protest and one crew member threw down his pencil and walked out of the room. This resistance to the administered question was related to two primary problems:

- it was discovered that the initial period of group formation was filled with *great optimism* (i.e., a utopian stage), where there was a general perception that everyone would get along and be friends over the course of the winter.

- the task of having people rank-order one another in terms of potential friendship created a negative emotional response on the part of crew members, since they believed that, at this point in the group formation process, “everyone” would be friends.

Network Questions • Use case: Extreme Friendship

The take-away of the use case is **not** that **the purpose** of the study was unfeasible, but that **the way** it was performed made it unfeasible. Specifically, the question and answer methodology created tension and countered the respondents' beliefs/expectations. The mix of a rank-order collection method (best friend to least friend), alter judgments, and expectations of friendship fostered a "perfect storm" in terms of sensitivity and interviewee burden.

To solve this issue, the researchers asked the crew about "who one interacts with socially" rather than "friendship", measuring that interaction using the 11-point Likert scale (from 0 to 10) anchored with words from never (0) to most often (10).

0 ___ 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10

Never Rarely Sometimes Often Most Often

Questions Format

A fundamental issue in the design of network questions is whether to use an open- or closed-ended format.

Questions Format · Close-ended Questions

Close-ended questions require the definition of the set of nodes of the network beforehand and respondents respond to answers on their relations with those actors.

The main advantages of using rosters are:

- limited recall error from the responders (forgetting someone they are related to);
- the guarantee that the set of respondents matches the set of actors asked about;
- it limits potential biases affecting the probability of an actor being selected by a respondent.

Closed-ended (aided)	**Example**
+ Boundaries are known and actors listed + Fewer concerns about respondent recall and accuracy + Each actor has an equal chance to being selected - Becomes cumbersome as networks grow in size	Who would you converse with if you meet on the street (check as many as apply)? Felicia Hardy ☐ Steve Rogers ☐ Sam Wilson ☐ Patsy Walker ☐ Bruce Banner ☐ Ted Sallis ☐ Kitty Pryde ☐

Questions Format · Close-ended Questions

Close-ended questions require the definition of the set of nodes of the network beforehand and respondents respond to answers on their relations with those actors.

The main <u>disadvantages</u> of using rosters are:

- the need to decide which nodes pertain to the study;
- it can be cumbersome/intimidating when the list of potential alters gets large. That can be mitigated with hierarchically-organised rosters, e.g., letting responders respond only about a subset of nodes selected with respect to organisational unit (s)he is in.

Closed-ended (aided)	**Example**
+ Boundaries are known and actors listed + Fewer concerns about respondent recall and accuracy + Each actor has an equal chance to being selected - Becomes cumbersome as networks grow in size	Who would you converse with if you meet on the street (check as many as apply)? `Felicia Hardy` ☐ `Steve Rogers` ☐ `Sam Wilson` ☐ `Patsy Walker` ☐ `Bruce Banner` ☐ `Ted Sallis` ☐ `Kitty Pryde` ☐

Questions Format • Open-ended Questions

Open-ended question require no prior decisions on who to obtain information about.

The main disadvantages of unaided questions are:

- it is not always clear to identify the actor whom a respondent names;

- free-listing introduce problems on the network of the respondent. E.g., if A lists 30 alters while B lists 15, can we conclude that A has a larger network than B? To mitigate this effect, the interviewer can probe the respondent (e.g., read the names listed and ask “do others come in mind?”)

Closed-ended (aided)	**Example**
+ Boundaries are known and actors listed + Fewer concerns about respondent recall and accuracy + Each actor has an equal chance to being selected - Becomes cumbersome as networks grow in size	Who would you converse with if you meet on the street (check as many as apply)? Felicia Hardy ☐ Steve Rogers ☐ Sam Wilson ☐ Patsy Walker ☐ Bruce Banner ☐ Ted Sallis ☐ Kitty Pryde ☐
Open-ended (unaided)	**Example**
+ Better for face-to-face interviews where probing can be used - Each actor in the network has unequal chances to being selected due to recall and free-listing issues - More subject to recall error • Can use a fixed-choice method to limit the number of actors elicited	If you wanted to learn more about what goes on in the Avengers organisation, who would you talk to? (Please, list as many relevant names as you can) _____ _____ _____ _____ _____ _____

Respondent Burden

The respondent burden represents the **commitment required from the respondent** to participate to the study—including time, attention, and emotions.

For example, in a political study the researchers identified over 400 potential actors. However, the respondents were high-status people who would not grant 3-hour interviews to the researchers.

To deal with the problem, the study began with interviews of *10 politically knowledgeable key informants* to free-list actors who were seen as “important” in the development and passing of a particular piece of legislation. The top 45 names most frequently listed by the key informants were used to confine the network.

This is an *emic/realist/recognition-based* way to find the boundaries of the network.

In addition, respondents were asked to name only three people on the list. This reduced the task to approximately 135 reported dyads, which was much more reasonable, although still daunting.

Respondent Burden

A guiding principle to relieve respondent burden is to *minimise respondent anger and frustration*.

Source of frustration for interviewees are *interview length* (particularly when respondents feel time constraints) and *lack of motivation*, e.g., the respondent feels coerced in participating or thinks the study is not useful (to them, in general, etc.).

A rule of thumb for optimally-sized network surveys is to **include only those questions that are theoretically critical** for the study (those we cannot dispense to answer the research questions). When uncertain about the relevance of a question, one can also conduct a preliminary study (exploratory/ethnographic) to understand how relevant that question is to characterise the studied network.

Also **the ordering of questions** matters. Try helping the respondent into and out of the survey. Questions can be divided into “simple” and “demanding” ones and ordered in a simple-demanding-simple fashion, where the first, simple questions “warm-up” the respondent for the more demanding ones and the last, simple ones relieve the accumulated cognitive tension.

Data Collection and Reliability

Repetita iuvant: whole network approaches are usually sensitive to missing/wrong data, moreover—as in any quantitative study—the smaller the network, the larger the effect of omissions or commissions of actors and/or ties.

The process of collection of network data has a profound impact on actor participation and on the reliability and validity of the collected data.

Data Collection and Reliability

Face-to-face data collection provides the **greatest opportunity for establishing rapport** with respondents and increase response rate. Additionally, it facilitates the use of elicitation interviewing techniques for the collection of network data, such as various probing techniques to improve respondent recall. Unfortunately, its also the most time-consuming and observer-dependent method for data collection.

Type of data collection	Establish Rapport	Issue of sensitivity	Interviewer response effect	Data-handling errors	Cost of administering	Ability to establish a rapport	Ability to maximise elicitation
Face-to-face	▼	▲	▲	~	▲	▲	▲
Phone	~	~	~	~	~	~	~
Self-administered	▲	▼	▼	~	~	▼	▼
Mail-out	▲	▼	▼	~	▼	▼	▼
Electronic Survey	▲	▼	▼	▼	▼	▼	▼

Data Collection and Reliability

Self-administered network surveys, including mail-out and online surveys, may minimise the degree of self-consciousness on the part of respondents. In addition, they do not suffer from reactions to the interviewer, and they are very convenient for the researcher. On the other hand, self-administered surveys that are not hand-delivered typically have much lower response rates.

Type of data collection	Establish Rapport	Issue of sensitivity	Interviewer response effect	Data-handling errors	Cost of administering	Ability to establish a rapport	Ability to maximise elicitation
Face-to-face	▼	▲	▲	~	▲	▲	▲
Phone	~	~	~	~	~	~	~
Self-administered	▲	▼	▼	~	~	▼	▼
Mail-out	▲	▼	▼	~	▼	▼	▼
Electronic Survey	▲	▼	▼	▼	▼	▼	▼

Archival Data Collection

To perform archival data-collection, the archival sources must contain information on social relations that are amenable to either a one-mode or two-mode network format.

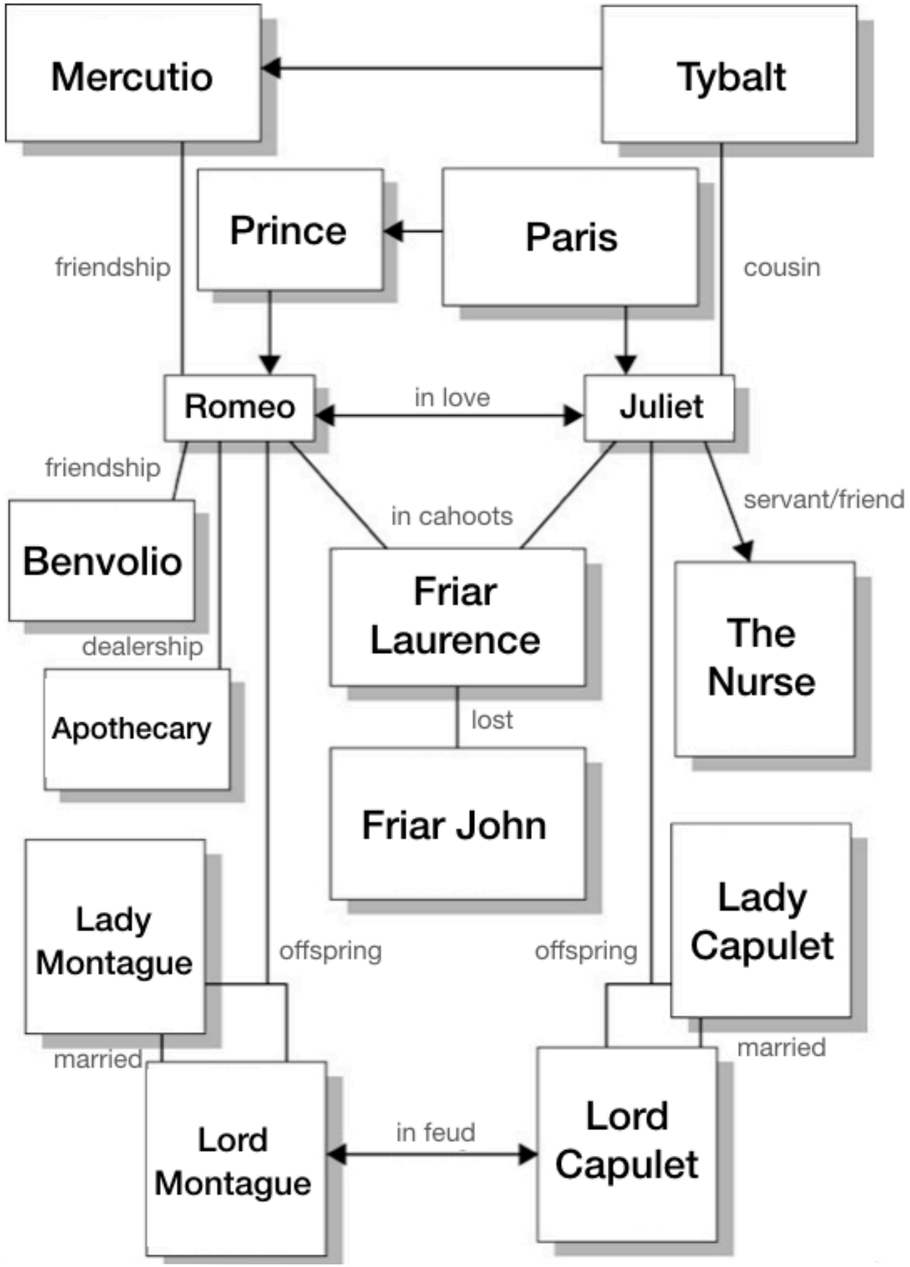
Examples of inherently relational archival sources are *records of marriages*, *business partnerships*, *legislative voting*, and *trades* (one-mode). Alternatively, ties can be inferred through co-occurrences, e.g., *overlaps in voting behaviours*, the *co-occurrence of patrons* of artists in Italy, the *co-attendance at political rallies* or events (two-mode).

However, be aware: **the nature** and structure **of the archival data frames which network relations a study can use**. If we are interested in economic exchange among villagers in Tuscany in the sixteenth century, but all that exists are marriage records, then the relational data available is not suitable for your research problem.

Archival Data Collection

Also less-structured archival data can be a source for relational studies. For example, historical records may not be well structured (as in accounting or marriage records) and use a narrative form. However, when those narratives—such as letters between people of some historical period—mention names, events, locations, etc., it is possible to build a social network by coding the narratives.

This is similar to the unfolding of a prosaic recount.



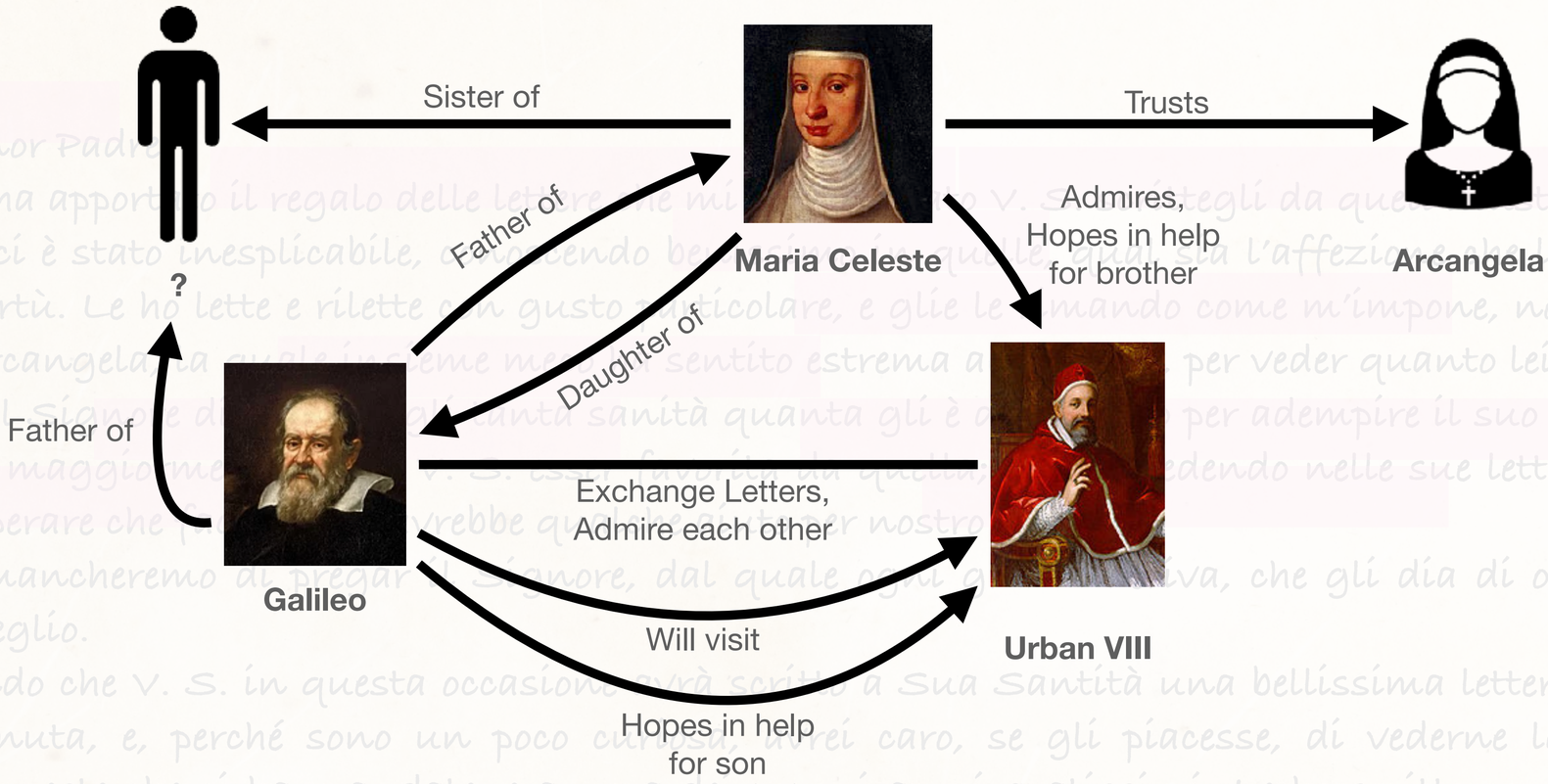
Archival Data Collection • Examples

In Villa
10 agosto [1623]

Molto Illustre Signor Padre.
Il contento che mi ha apportato il regalo delle lettere che mi ha mandato V. S. scrittegli da quell'illustrissimo Cardinale, oggi sommo Pontefice, ci è stato inesplicabile, conoscendo benissimo in quelle, qual sia l'affezione che le porta, e quanta stima faccia della sua virtù. Le ho lette e rilette con gusto particolare, e glie le rimando come m'impone, non l'avendo mostrate ad altri che a Suor Arcangela, la quale insieme meco ha sentito estrema allegrezza, per veder quanto lei sia favorita da persona tale. Piaccia pure al Signore di concedergli tanta sanità quanta gli è di bisogno per adempire il suo desiderio di visitar Sua Santità, acciocchè maggiormente possa V. S. esser favorita da quella; e anco vedendo nelle sue lettere quante promesse gli faccia, possiamo sperare che facilmente avrebbe qualche aiuto per nostro fratello.
Intanto noi non mancheremo di pregar il Signore, dal quale ogni grazia deriva, che gli dia di ottener quanto desidera, purché sia per il meglio.
Mi vo immaginando che V. S. in questa occasione avrà scritto a Sua Santità una bellissima lettera per rallegrarsi con lei della dignità ottenuta, e, perché sono un poco curiosa, avrei caro, se gli piacesse, di vederne la copia, e la ringrazio infinitamente di queste che ci ha mandate, e ancora dei poponi a noi gratissimi. Le ho scritto con molta fretta, imperò la prego a scusarmi se ho scritto così male. La saluto di cuore insieme con l'altre solite.

figliuola Affezionatissima
S. M. C.

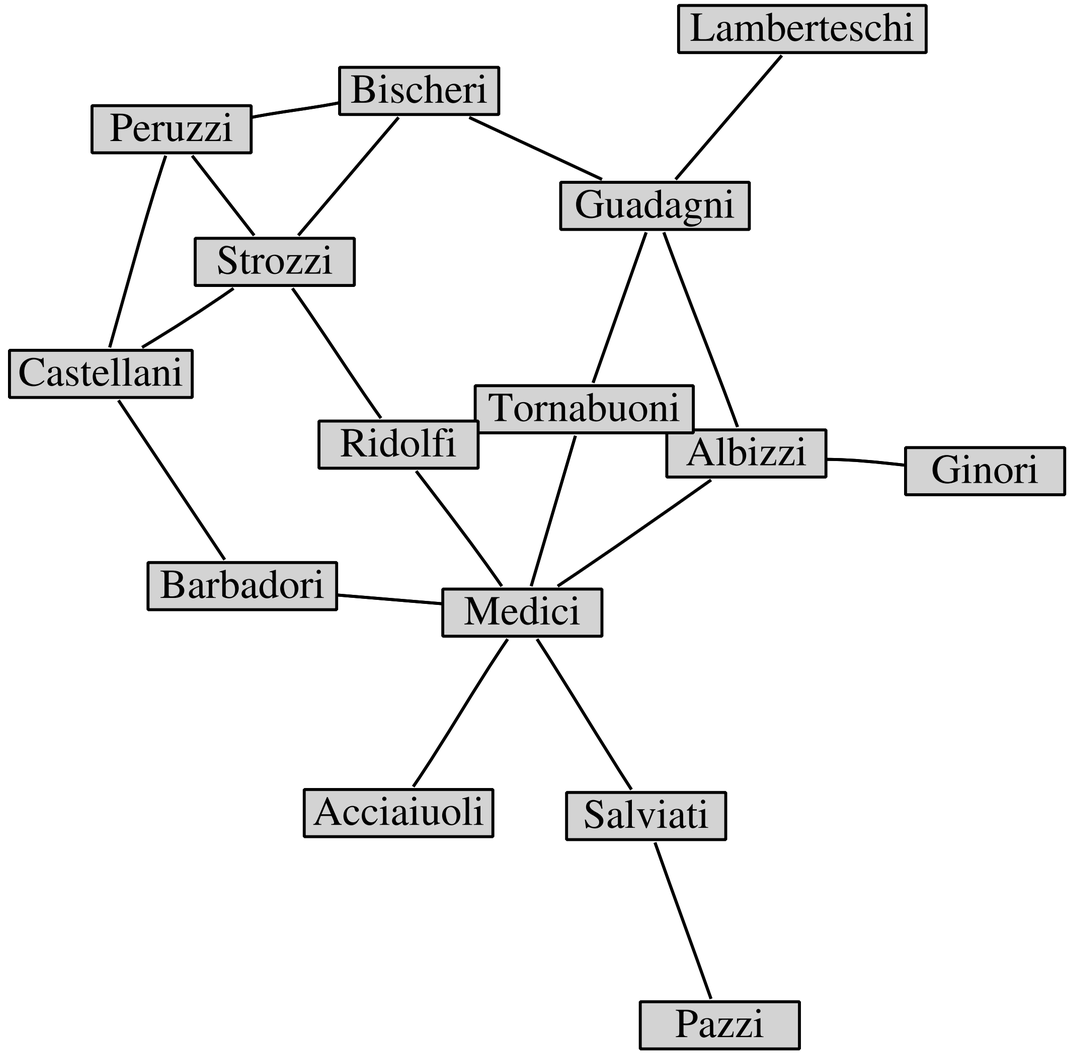
Archival Data Collection • Examples



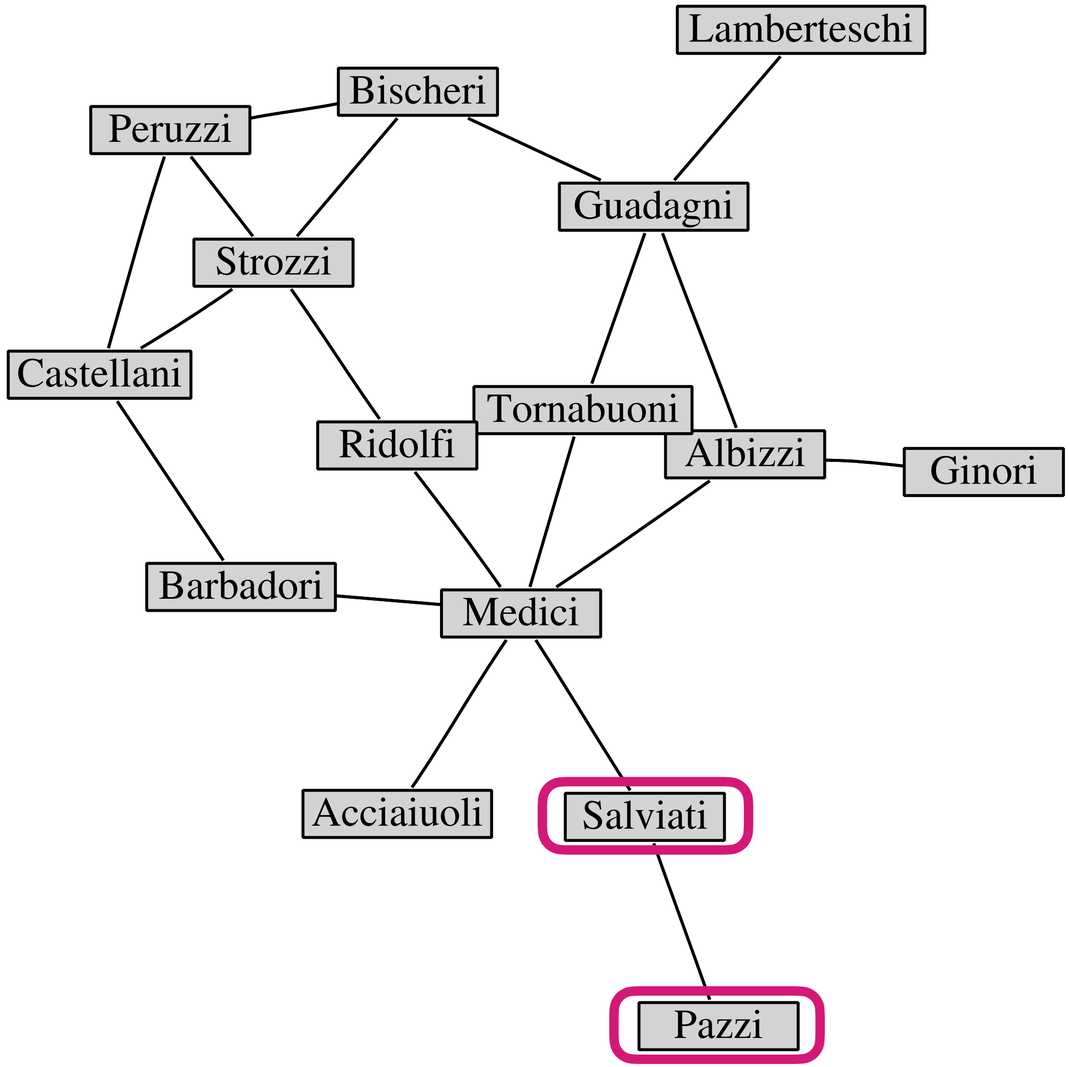
Archival Data Collection • Examples

A study used data from a major historical work on social dynamics in fifteenth-century Florence, with a particular focus on the rise of the Medici, to build a multiplex network dataset of intermarriage ties, business ties, joint ownerships, partnerships, bank employment, real estate ties, patronage, personal loans, friendships, “surety ties”—actors who put up bond for someone in exile—historical accounts, tax records (catasto), neighbourhood residence, and tax assessments.

The authors were able to build datasets that included dynamic networks involving multiple relations and modes (both one- and two-mode) and a variety of attributes that could be used to triangulate data and test hypotheses.



Archival Data Collection • Examples



Data from Electronic Sources

The collection of data from **electronic sources** is similar to the collection of network data in archival or historical research.

Many sites on the Internet contain information that is inherently network-oriented.

There is a **large amount of existing data** on—or data that can be mined from—email communications, social networking sites, movie/music/book databases, scientific citation databases, wikis, Web pages, digital news sources, and so on.

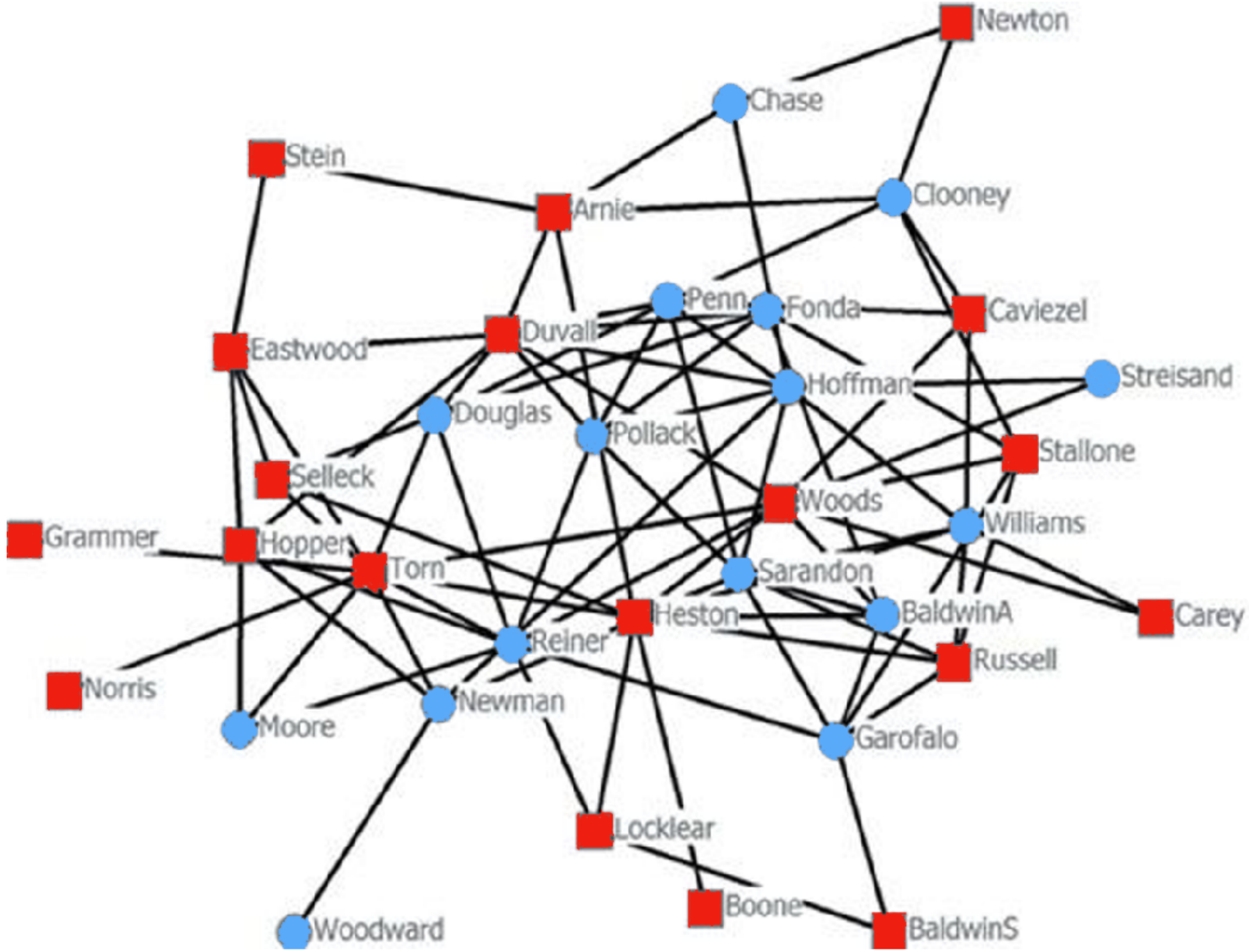
Many of these already have information available in a one-mode or two-mode network format, while others require using/writing data-mining software to put it into data formats that can be more readily analysed.

E.g., Twitter provides network data in the form of follower and followed ties.

Data from Electronic Sources • Examples

The Internet Movie Database (IMDb) has a tremendous amount of data on virtually every movie ever made, in machine-readable form.

Some of this information can be used to construct two-mode data matrices, such as actor-by-movie, movie-by-keyword, movie-by-news article and so on, which can then be converted into one-mode networks.



Storing Network Data, Formats

When storing network data digitally, we need to decide a form of representation in the memory of the computer.

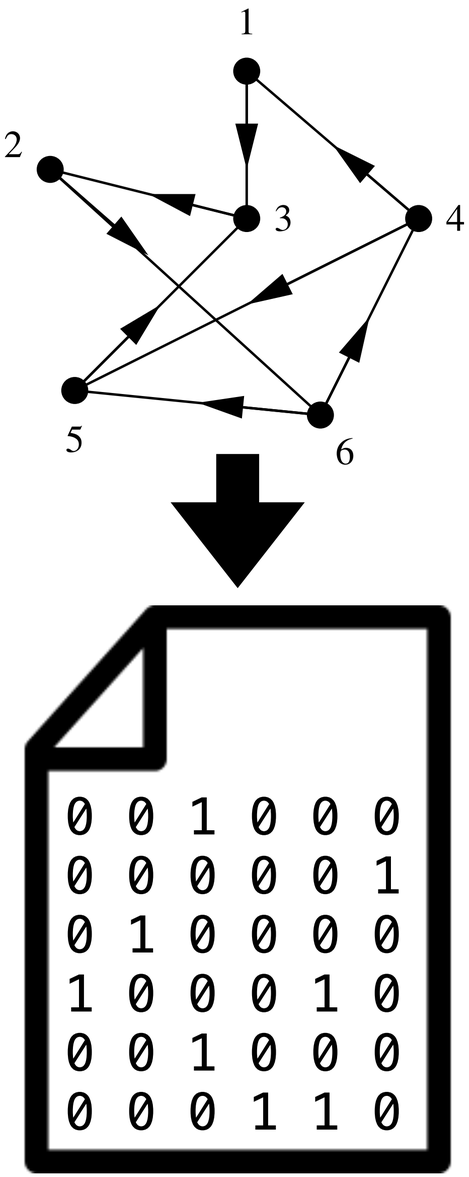
Network data can be stored in files using a large number of different formats, however, they mostly boil down to entries with information either on the edges, the nodes, or both.

Different choices about how to store the data can make a substantial difference to both the speed of a program and the amount of memory it uses.

Storing Network Data, Formats

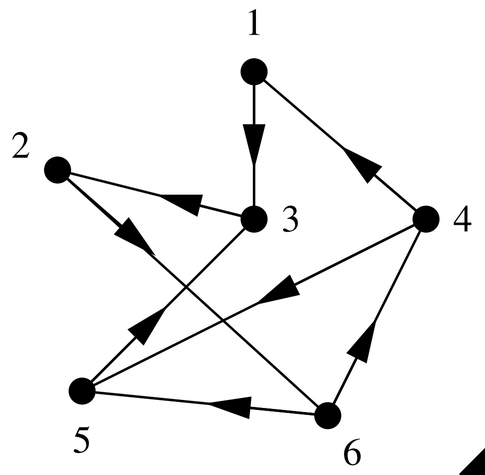
The first step in representing a network in a computer is to label the nodes so that each can be uniquely identified. The most common way of doing this is to give each a numeric label, usually an integer. It usually does not matter which node gets assigned which number—the purpose of the numbers is only to provide unique labels for identifying the nodes.

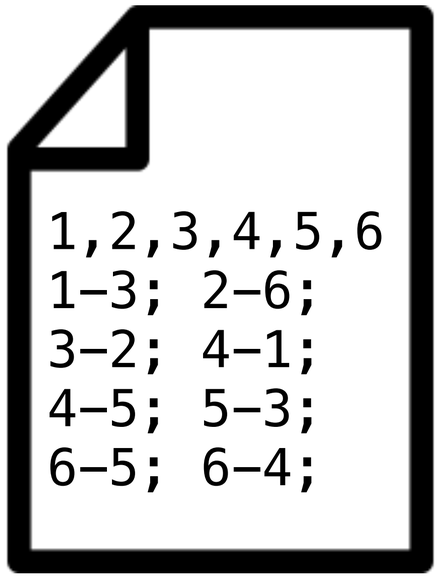
In some programming languages, including C, Python, and Java, it is conventional for numberings to start at zero and go up to $n - 1$. Most, though not all, file formats for storing networks already specify integer labels for nodes. In that case, it is sufficient to just use those labels.



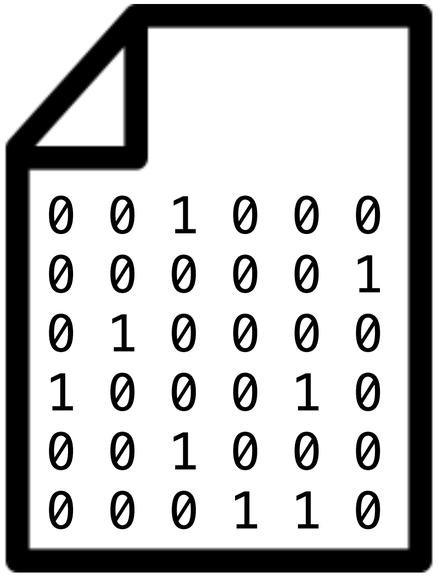
Adjacency matrix

Storing Network Data, Formats





edge list

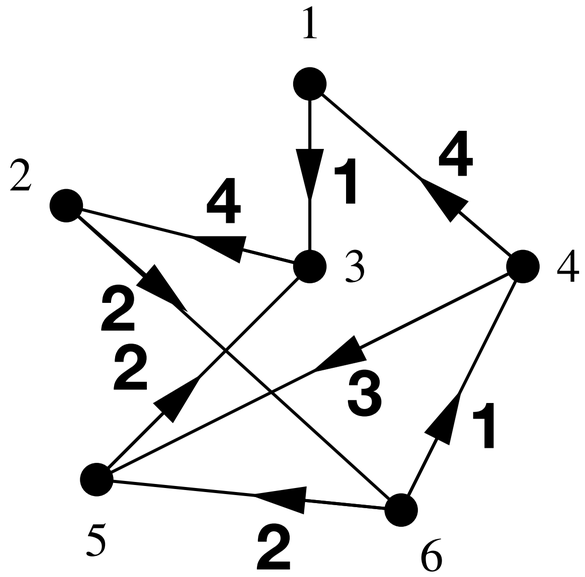


adjacency matrix

Storing Network Data, Formats

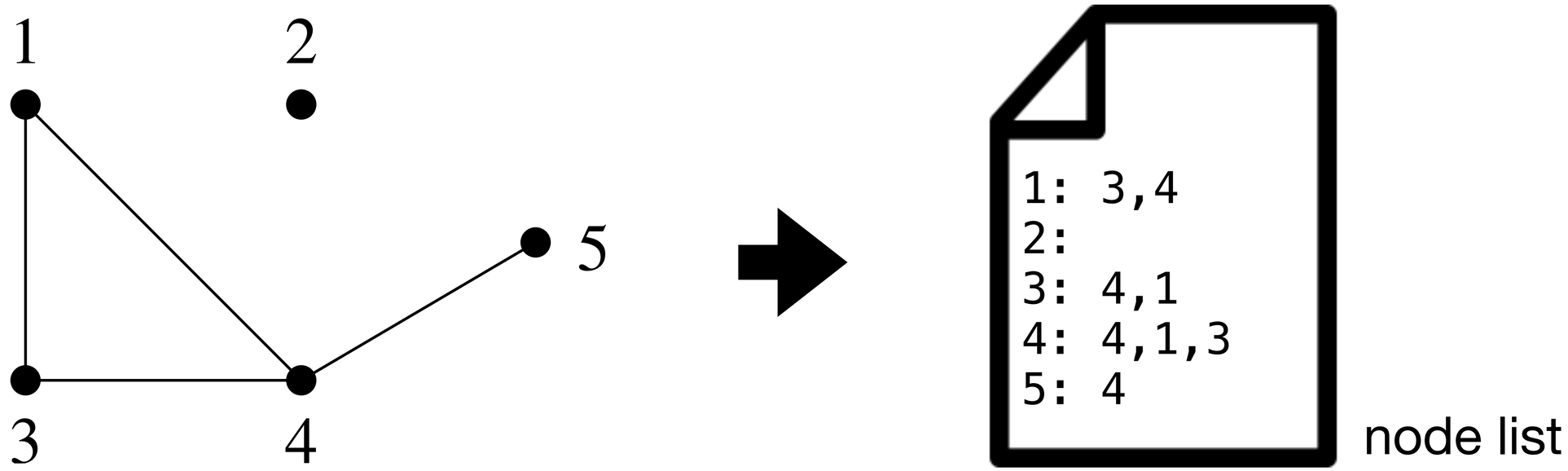
Often, the nodes in a network have annotations or values attached to them, in addition to their labels, e.g., the nodes in a social network might have names; nodes from the Web might have URLs; nodes on the Internet might have IP addresses. Nodes could also have properties like age, capacity, or weight, represented by additional numbers (integers, decimals, etc.). All of these other notations and values can be stored straightforwardly in the memory of the computer by defining an array of a suitable type with n elements, one for each node.

```
1: [Felicia; Hardy; Female; Burglar, Martial Arts ]
2: [Steve; Rogers; Male; Strength, Military Tactics ]
3: [Sam; Wilson; Male; Acrobatics, Military Tactics]
4: [Patsy; Walker; Female; Martial Arts, Gymnast]
5: [Bruce; Banner; Male; Genius intellect]
6: [Kitty; Pryde; Female; Intangibility, Gifted intellect]
1–3:1; 2–6:3; 3–2:4; 4–1:1; 4–5:2; 5–3:3; 6–5:4; 6–4:3;
```

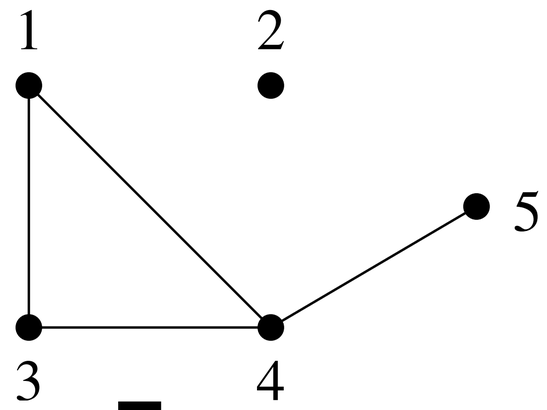


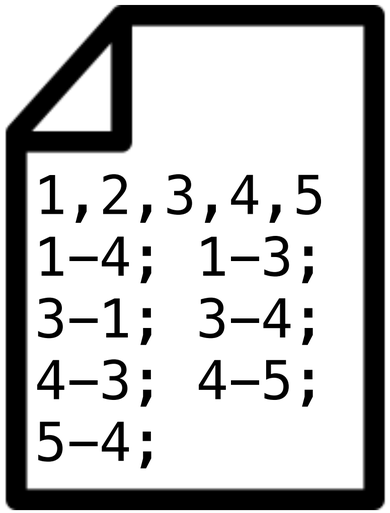
Storing Network Data, Formats

The most common alternative to storing the adjacency matrix of a network is to use an adjacency list, which is actually a set of lists, one for each node. Each list contains the labels of the other nodes to which a given node is connected.



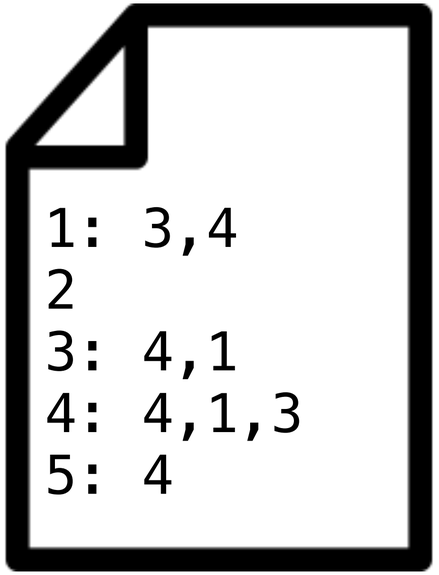
Storing Network Data, Formats





edge list

~ 44 bytes



~ 29 bytes

node list

Data Transformation

There are many transformations applicable to data in the course of an analysis to make some evidence emerge.

In the following, we will briefly overview the main ones: transposing matrices, symmetrising, dichotomising, inputing missing values, combining relations, combining nodes, and extracting subgraphs.

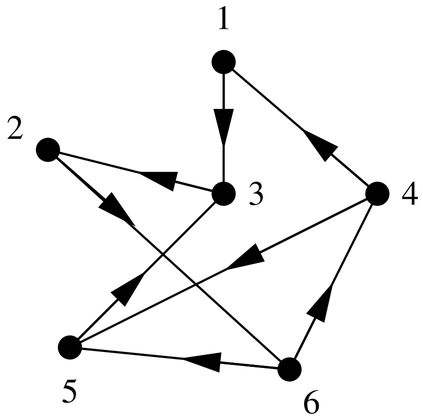
Data Transformation • Transposition

Transposing a matrix means **interchanging its rows with its columns**. Transposition, applied to a non-symmetric adjacency matrix, reverses the direction of arcs and can be helpful in maintaining a consistent interpretation of the ties in the network.

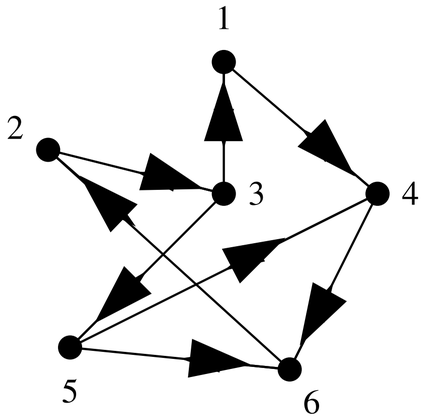
For example, suppose a survey asks “who do you seek advice from?”. However, advice flows from the adviser to the advisee. In this case, it might be useful to transpose the matrix and think of it as “who gives advice to whom”.

A similar situation occurs with food webs, where we have data on which species eat some other species. Ecologists like to reverse the direction of the arrows because they want to think in terms of the direction of energy flow through the ecosystem (flowing from the prey to the predator).

Data Transformation • Transposition



	1	2	3	4	5	6
1	0	0	0	**1**	0	0
2	0	0	**1**	0	0	0
3	**1**	0	0	0	**1**	0
4	0	0	0	0	0	**1**
5	0	0	0	**1**	0	**1**
6	0	**1**	0	0	0	0



	1	2	3	4	5	6
1	0	0	**1**	0	0	0
2	0	0	0	0	0	1
3	0	**1**	0	0	0	0
4	**1**	0	0	0	**1**	0
5	0	0	**1**	0	0	0
6	0	0	0	**1**	**1**	0

Data Transformation • Missing Data

Many graph-theoretic measures are sensible to missing values (and the consequent lack of ties).

A (naive) solution is to eliminate the nodes of which we miss the data (i.e., deleting both their rows and columns in the adjacency matrix). However, node removals (especially from the original/source matrix) should be performed with a reason, accounted for in the analysis of the validity and reliability of the study.

It is also likely that other nodes can have arcs to that node (since the other nodes responded about that node) and thus, removing it, we would waste some useful data. It would seem worthwhile, then, to search for ways to retain the problematic node.

Indeed, for example, if the missing node is one of the most “important” in the network—e.g., people from top-level management, who frequently have little time to fill out surveys or leave interviews—the models, and thus our conclusions drawn from them, can be quite far from the reality.

Data Transformation • Missing Data

In the case of symmetric or undirected relations, a direct solution is to fill-in the missing rows with the data found in the corresponding column.

For non-symmetric relations, such as “who do you seek advice from?”, this technique would not make sense.

However, if we have two non-symmetric relations that can be used to fill each other’s missing data, we can use the transpose of the second matrix to fill-in the missing rows in the first, and vice versa. To make an example, if we have two relations from the questions “who do you seek advice from?” and “who seeks advice from you?”, e.g., we can transpose the matrix of the first question and use that data to fill-in the missing values of the second—and *combine the relations*, as explained later on. Of course, the ties in the resulting network would abstract from the specific “questions” of the two original sources, since now they represent e.g., mentorship relations (where we lost the role of the nodes involved in a tie).

Data Transformation • Symmetrisation

Symmetrising means creating a new dataset in which all ties are reciprocated (and perhaps regarded as undirected). There are many reasons to symmetrise data:

- some analytical techniques assume symmetric data;

- some data-cleaning processing has a symmetrisation step. E.g., in open-ended questionnaires unintended asymmetry comes from respondents who forgot to mention people. When symmetrising, we can either follow a **union (OR) or intersection (AND)** policy. E.g., if we consider friendship to be symmetric, then we can symmetrise using an OR policy (commutative, $0 \vee 0 = 0$ and $1 \vee 0 = 1 \vee 1 = 1$); however, if we suspect name-dropping, we should adopt an AND policy (commutative, $0 \wedge 0 = 1 \wedge 0 = 0$ and $1 \wedge 1 = 1$).

- when studying inherently symmetric relations. E.g., if we asked respondents who they receive advice from, we are tracking information exchange. If the latter is the social relation we are interested in, we can symmetrise using the rule that if one gives or receives advice from the other, there is an exchange.

Data Transformation • Symmetrisation

From the point of view of a matrix A representing a network, when we symmetrise, we are comparing an entry A_{ij} with the corresponding entry A_{ji} and, if needed, we make them the same.

More in general, the **union** policy corresponds to taking the **larger** of the two entries while the **intersection** policy takes the **smaller** of the two.

Besides the above two policies, others are possible. For instance, for valued data, we might consider taking the average of the two entries. E.g, if i estimates having had lunch with j eight times in a month, but j estimates having had lunch with i ten times, we can view these as two measurements of the same underlying quantity, and use the average as the best estimate of that quantity.

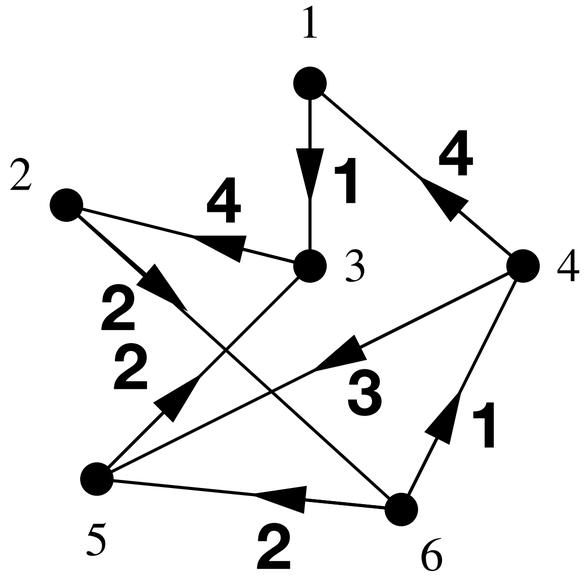
Data Transformation • Dichotomisation

Dichotomising refers to converting valued data to binary data, i.e., we take a valued adjacency matrix and set to 1 all cells with a value greater than (or less than, or exactly equal to) a certain threshold, and set all the remaining cells to 0.

The main reason for doing this is that some measures are only applicable to binary data.

Dichotomising is the first example of a more general concept of edge cut-off, useful to reduce the density of a network and make it more efficient/feasible to handle large networks.

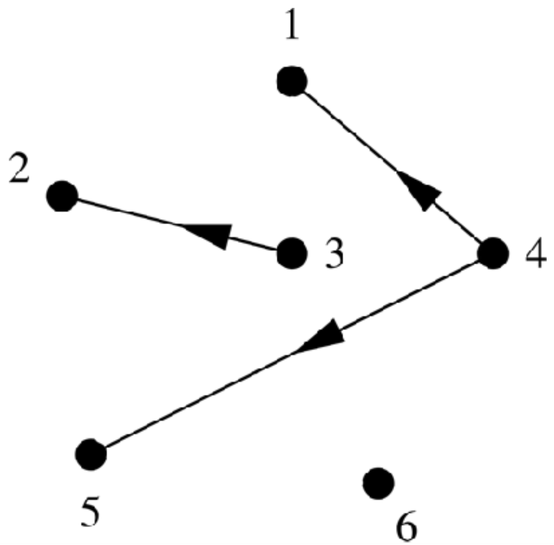
Data Transformation • Dichotomisation



	1	2	3	4	5	6
1	0	0	0	**4**	0	0
2	0	0	**4**	0	0	0
3	**1**	0	0	0	**2**	0
4	0	0	0	0	0	**1**
5	0	0	0	**3**	0	**2**
6	0	**2**	0	0	0	0

Cut-off set to 3

	1	2	3	4	5	6
1	0	0	0	**1**	0	0
2	0	0	**1**	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	**1**	0	0
6	0	0	0	0	0	0



Data Transformation • Combining Relations

Most network studies collect, on the same set of nodes, multiple relations which are then useful to be combined into one.

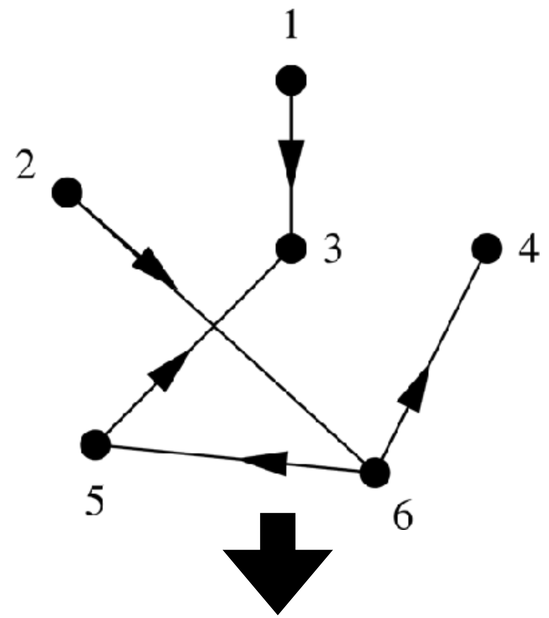
For example, we might take three separate network questions, such as “who do you attend sports events with?”, “who do you go to the theatre with?”, and “who do you go out to dinner with?” and combine them into a more general, analytically defined, relation, such as “who socialised with whom”.

Mathematically, to combine relations, we can sum the separate matrices.

$$\mathbf{A}+\mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

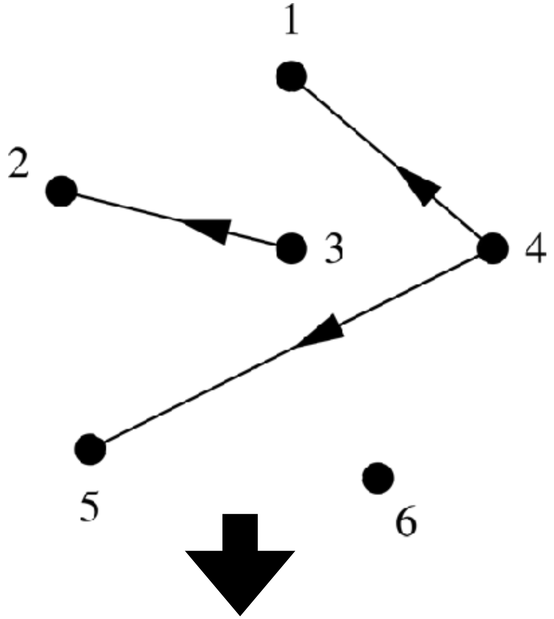
$$= \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$$

Data Transformation • Combining Relations



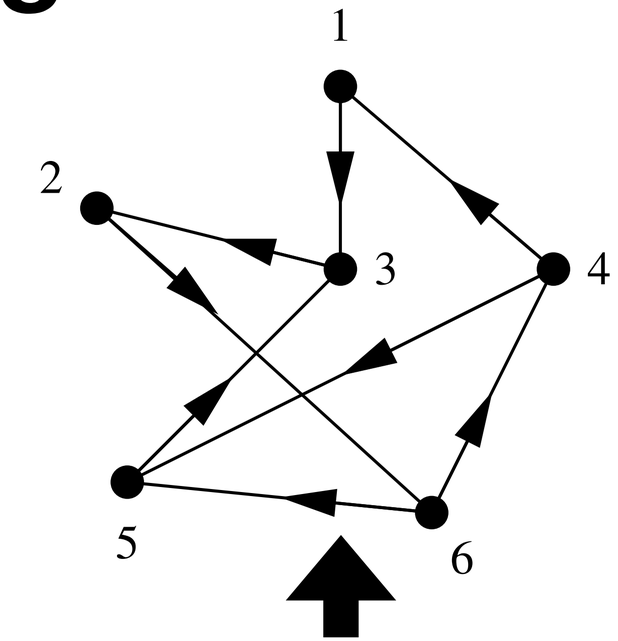
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	**1**	0	0	0	**1**	0
4	0	0	0	0	0	**1**
5	0	0	0	0	0	**1**
6	0	**1**	0	0	0	0

+



	1	2	3	4	5	6
1	0	0	0	**1**	0	0
2	0	0	**1**	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	**1**	0	0
6	0	0	0	0	0	0

=



	1	2	3	4	5	6
1	0	0	0	**1**	0	0
2	0	0	**1**	0	0	0
3	**1**	0	0	0	**1**	0
4	0	0	0	0	0	**1**
5	0	0	0	**1**	0	**1**
6	0	**1**	0	0	0	0

Data Transformation • Subgraphs

It may happen that we either cannot (computationally) or do not want to analyse a whole network and thus we may wish to delete nodes from the network.

When we want to analyse a subset of the nodes of the original network, called a subgraph, it could be because the nodes in the subgraph are outliers in some respect or because we need to match the data to another dataset where not all nodes of the original network are present.

It could also be the case that we want to aggregate similar nodes (wrt some measure) into a single node, preserving all the ties of the aggregated nodes (e.g., aggregating nodes in the same departments, to model department-level relations).

Data Transformation • Normalisation

In some studies it is useful to re-express, standardise or normalise network data to ensure we are making fair comparisons across rows, columns or entire matrices.

For example, when collecting ratings there could be a problem due to respondents' use and interpretations of the scales: some respondents may lean towards the high-end of the scale while others may have assigned lower values, although the described ties are similar.

Normalisation is also a way to uniform measures, e.g., if in interviews respondents assessed some physical distance using different scales (feet, yards, meters).

Data Transformation • Normalisation

To be able to compare (reliably) the values in the above cases, it is necessary to reduce each value to a common denominator.

In the example of the ratings, to “smooth out” the problem of individual-perception issues, we can normalise the data using procedures from statistics. These include methods that use as common denominators means, marginals, standard deviations, means and standard deviations together, Euclidean norms, and maximums. Each type of normalisation can be performed on each row separately, on each column separately, on each row and each column, and on the matrix as a whole.

In the second example, the normalisation is performed by fixing a study-standard metric system and converting the non-standard values.

Software for Network Analysis and Visualisation

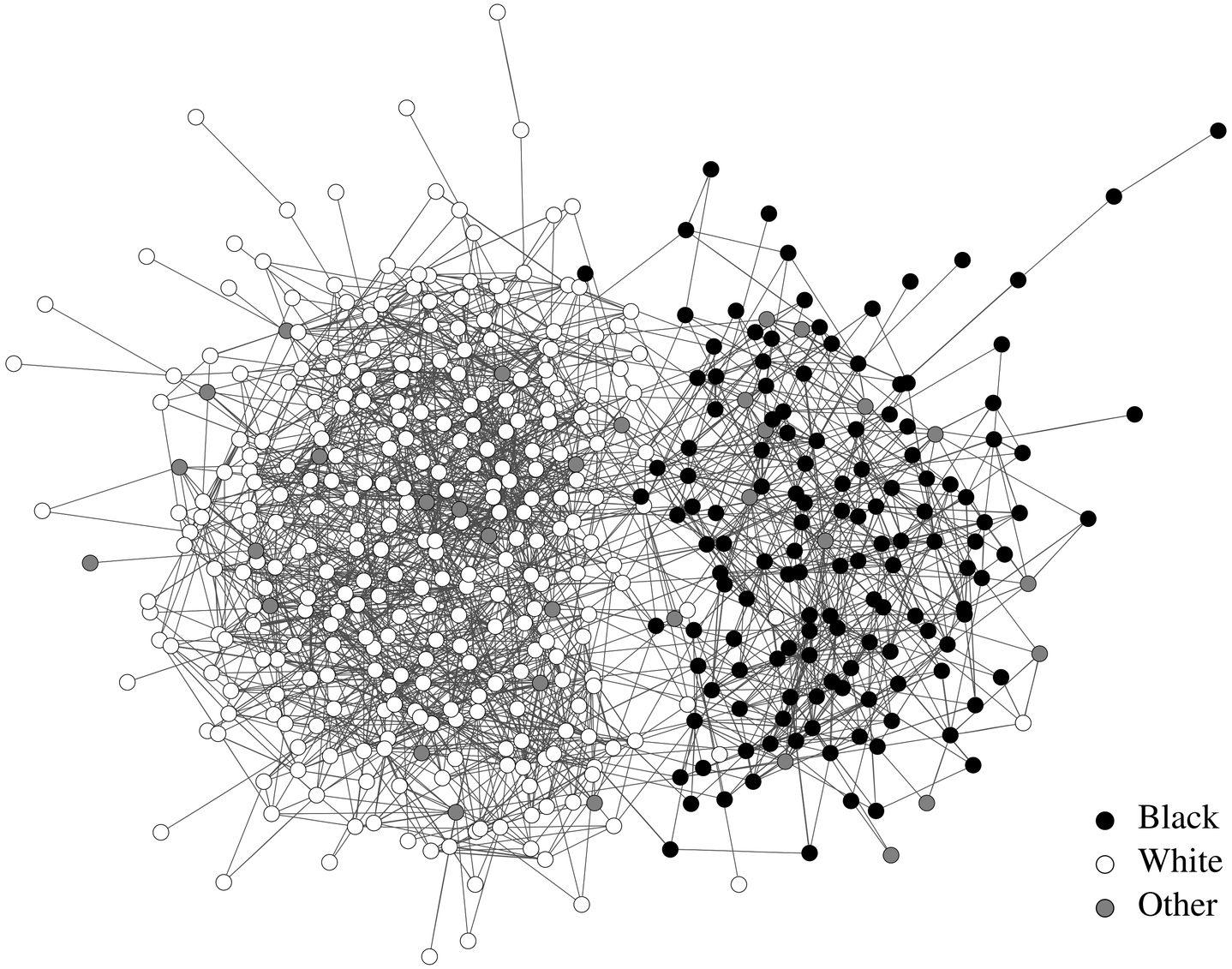
Many of the techniques for studying networks have been automatised and are available for use in the form of network analysis software packages.

These packages are often of very high quality, produced by skilled and knowledgeable programmers, and they are frequently used and tested by network researchers who, in time, provided feedback and requested to fix, optimise, and expand those packages.

Name	Availability	Platform	Description
Gephi	Free	WML	Interactive network analysis and visualization
Pajek	Free	W	Interactive social network analysis and visualization
InFlow	Commercial	W	Interactive social network analysis and visualization
UCINET	Commercial	W	Interactive social network analysis
yEd	Free	WML	Interactive visualization
Visone	Free	WL	Interactive visualization
Graphviz	Free	WML	Visualization
NetworkX	Free	WML	Python library for network analysis and visualization
JUNG	Free	WML	Java library for network analysis and visualization
igraph	Free	WML	C/R/Python libraries for network analysis

Software for Network Analysis and Visualisation

Network analysis software usually also come with network visualisation support, to enable researchers to have a “glimpse” on the network and orient the application of measures—the famous interplay between the human ability to spot visual patterns and the application of measures that test/quantify their presence.



Running Time and Computational Complexity

Computers are much faster than humans, but they too have limits in how quick they can terminate a given computation. How fast a computer can perform a given task can be mathematically determined and falls under the field of “**computational complexity**” and (here) is useful to predict how much time our measures can take to be performed and possibly avoid wasting time on programs that will not finish running in any reasonable amount of time.

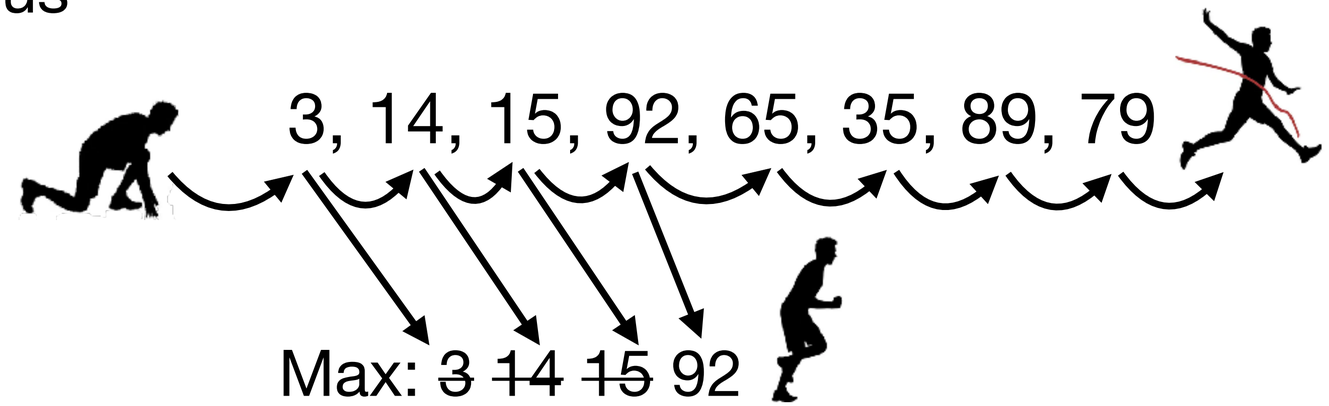
More technically, **computational complexity is a measure of the running time of a computer algorithm, as a function of the size of the problem it is tackling.**

Running Time and Computational Complexity

Consider a simple example:

how long does it take to find the largest number in a list of n numbers?

Assuming the numbers are not given to us in some special order (such as largest first), our algorithm consists of a set of steps: we go through the whole list, number by number, keeping a running record of the largest one we have seen, until we get to the end.



This is a very simple example of a computer algorithm, but still we might use it to find the node in a network that has the highest degree.

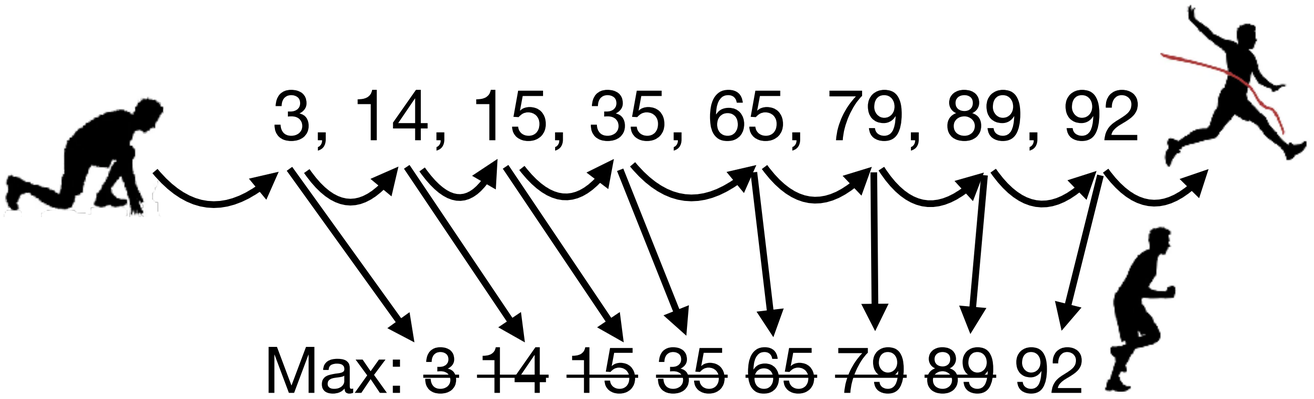
Running Time and Computational Complexity

The worst possible case, in which our algorithm does the most work is when the list is sorted in increasing order, and at each step the algorithm will:

- compare the next number in the list with the previous record-holder;

- replace the previous record-holder with the new number.

The case is called “worse” because at each step we do the maximum amount of work, due to the configuration of the data.



Running Time and Computational Complexity

Identifying and measuring the worst case is important.

Let us say n is the number of steps our worst case takes. Then, we know that, if we are unlucky, the total time taken to complete the algorithm (its running time) will at most be $n \cdot t$, where t is the time taken at each individual step.

Thus, we say that the running time, or time complexity, of this algorithm is of order n, or just $O(n)$ for short, as the descriptor of the upper bound of the growth-rate of the function.