NIHR | Applied Research Collaboration South West Peninsula

@penCHORD_UoE
@peninsula_ARC

Module 4 : Network Analysis
Session 4A : Network Analysis
Elliott Coyne

““Enrich our country by connecting New Zealanders to each other
and New Zealand to the world””

HSMA 5

#hsma5isalive

NIHR | Applied Research Collaboration South West Peninsula

@penCHORD_UoE
@peninsula_ARC

Module 4 : Network Analysis
Session 4A : Network Analysis
Elliott Coyne

"Enrich our country by connecting New Zealanders to each other and New Zealand to the world"

HSMA 5

#hsma5isalive

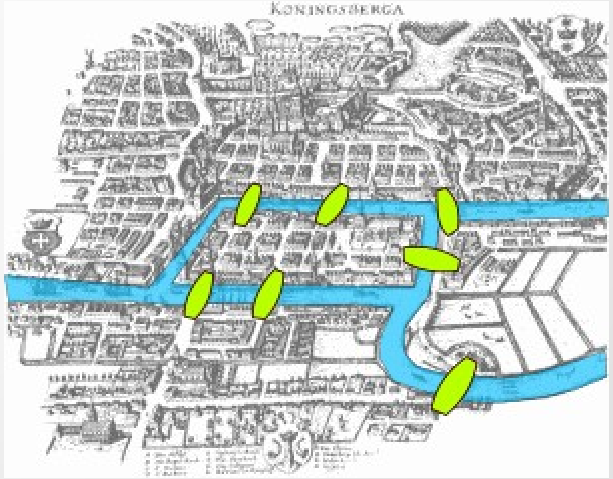
Learning Objectives

- Understand what a network graph is and how it is constructed
- Be able to transform raw data into a format suitable for conducting a network analysis
- Know how to create a simple network graph in Python using NetworkX

What is a network?

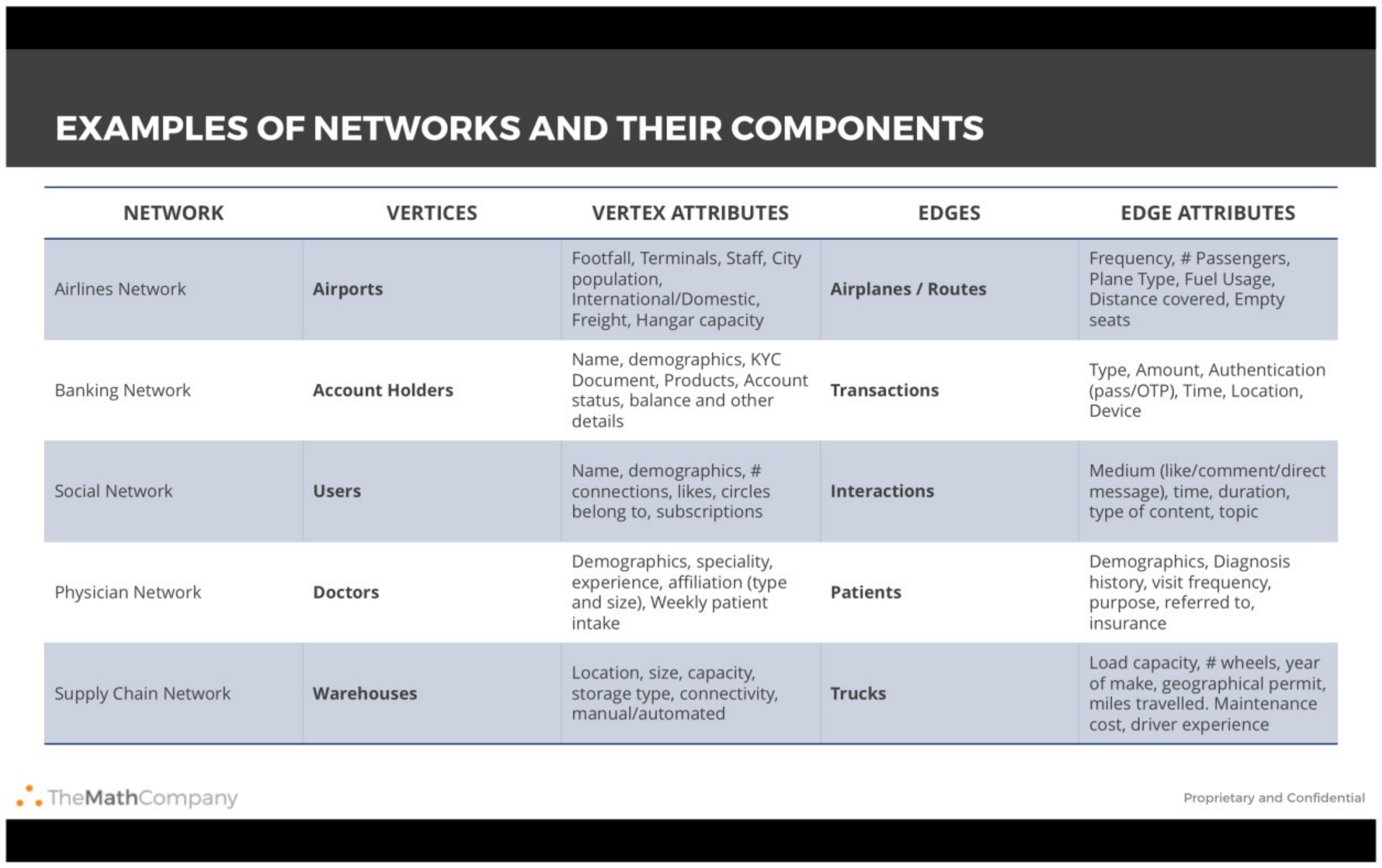
Network graphs are used to:

- Study the function of large and/or complex systems
- Visualise relationships and interactions in a intuitive way
- Represent abstract concepts in a concrete way



https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

Examples



EXAMPLES OF NETWORKS AND THEIR COMPONENTS

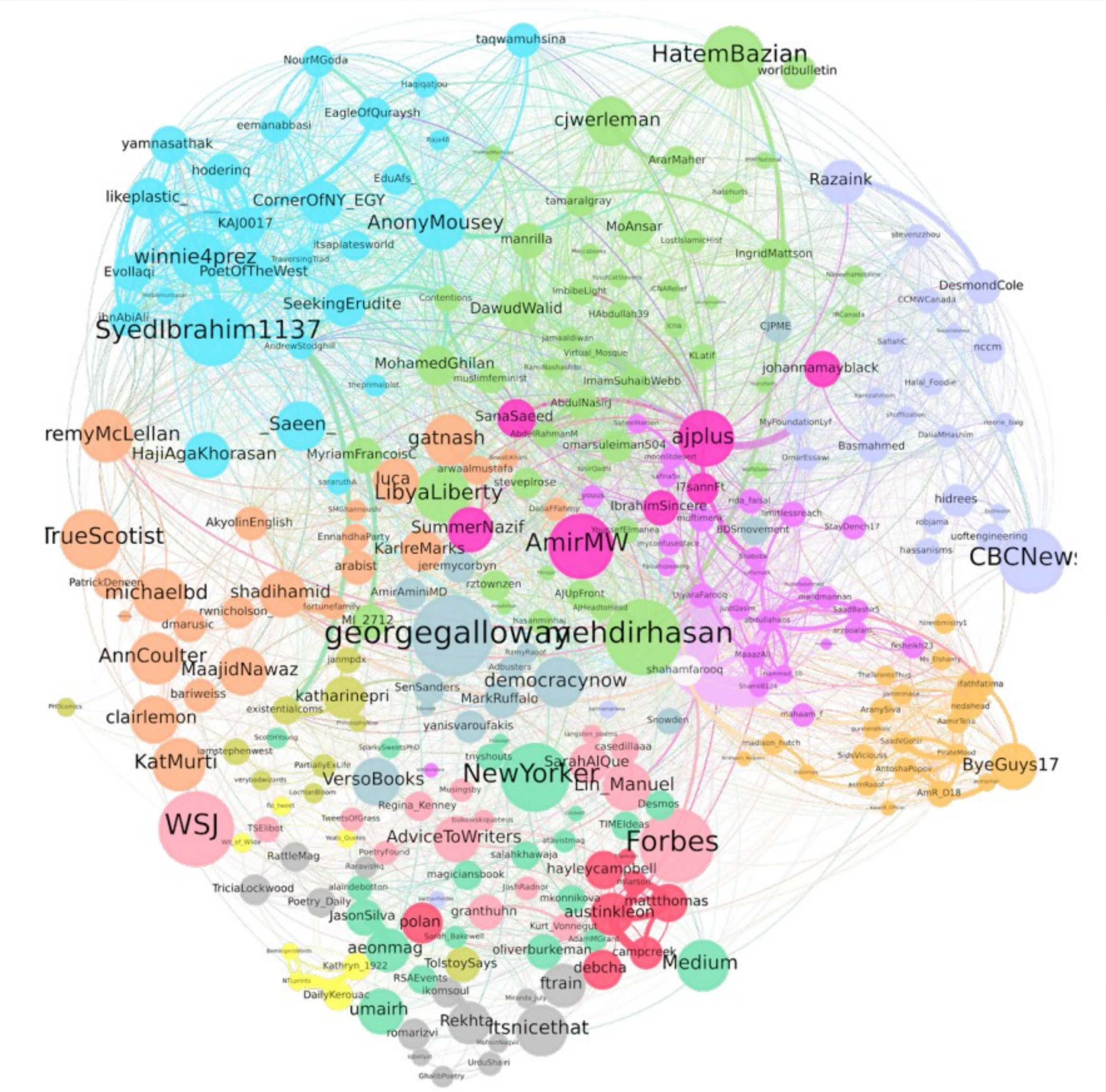
NETWORK	VERTICES	VERTEX ATTRIBUTES	EDGES	EDGE ATTRIBUTES
Airlines Network	**Airports**	Footfall, Terminals, Staff, City population, International/Domestic, Freight, Hangar capacity	**Airplanes / Routes**	Frequency, # Passengers, Plane Type, Fuel Usage, Distance covered, Empty seats
Banking Network	**Account Holders**	Name, demographics, KYC Document, Products, Account status, balance and other details	**Transactions**	Type, Amount, Authentication (pass/OTP), Time, Location, Device
Social Network	**Users**	Name, demographics, # connections, likes, circles belong to, subscriptions	**Interactions**	Medium (like/comment/direct message), time, duration, type of content, topic
Physician Network	**Doctors**	Demographics, speciality, experience, affiliation (type and size), Weekly patient intake	**Patients**	Demographics, Diagnosis history, visit frequency, purpose, referred to, insurance
Supply Chain Network	**Warehouses**	Location, size, capacity, storage type, connectivity, manual/automated	**Trucks**	Load capacity, # wheels, year of make, geographical permit, miles travelled. Maintenance cost, driver experience

TheMathCompany

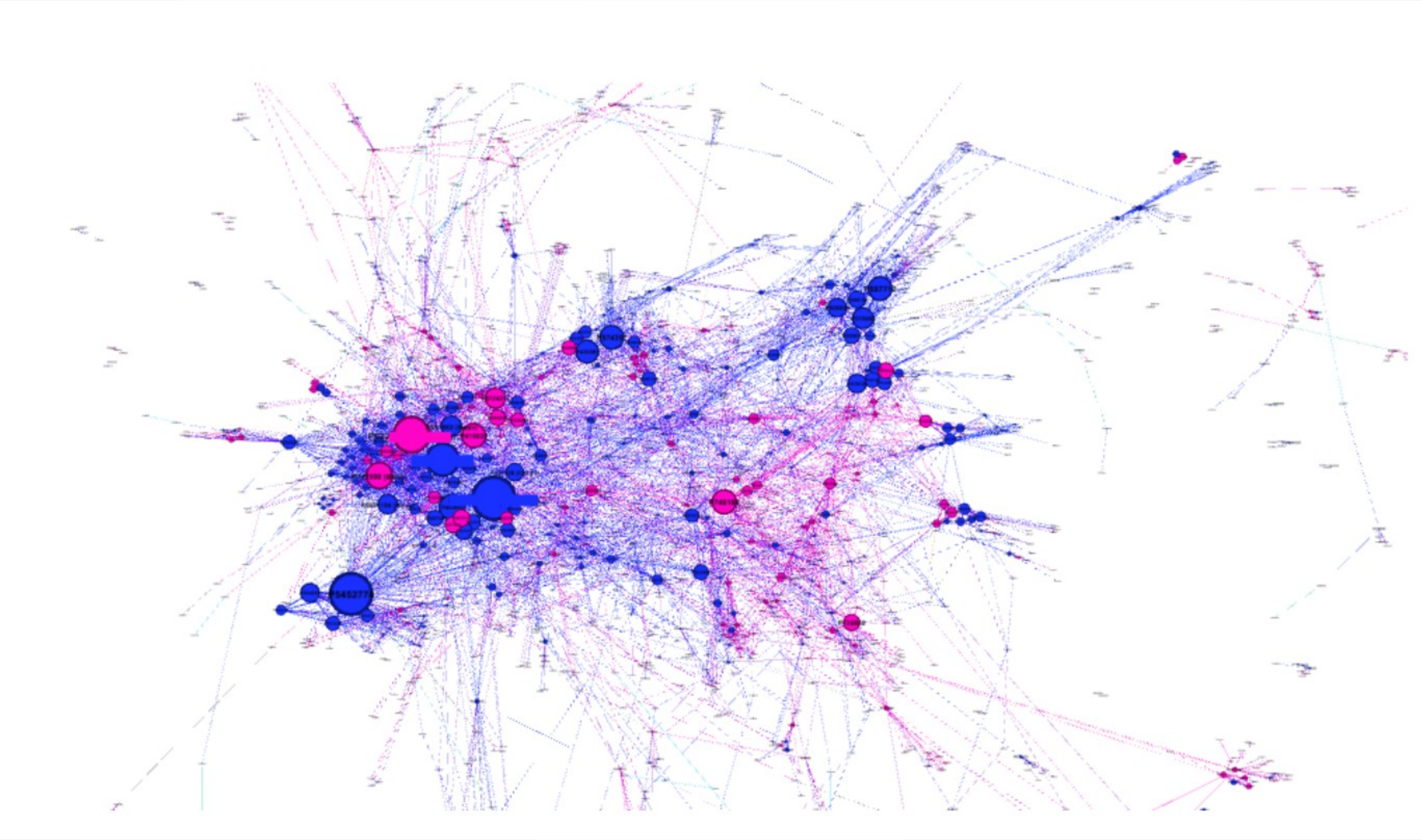
Proprietary and Confidential

Credit: https://www.analyticsvidhya.com/blog/2018/04/introduction-to-graph-theory-network-analysis-python-codes/

Example 1 – Twitter Handles

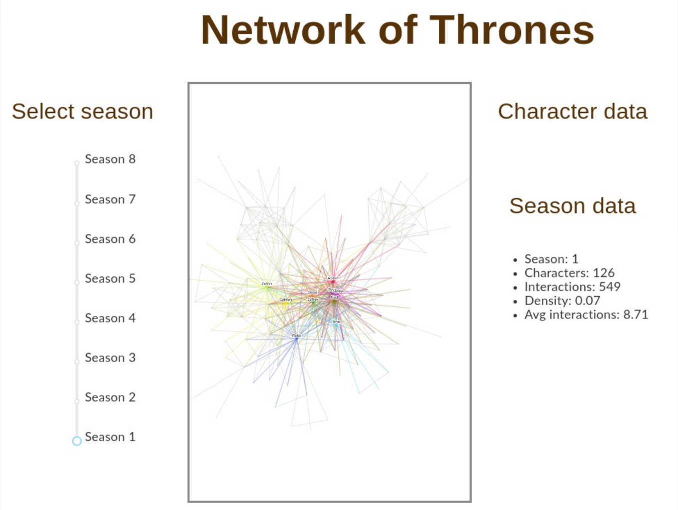


Example 2 – HSMA: Offenders and Victims



Credit: Jenna Thomas and Charley Bartlett, Devon and Cornwall Police

Interactive Examples



Network of Thrones

Select season

Season 8

Season 7

Season 6

Season 5

Season 4

Season 3

Season 2

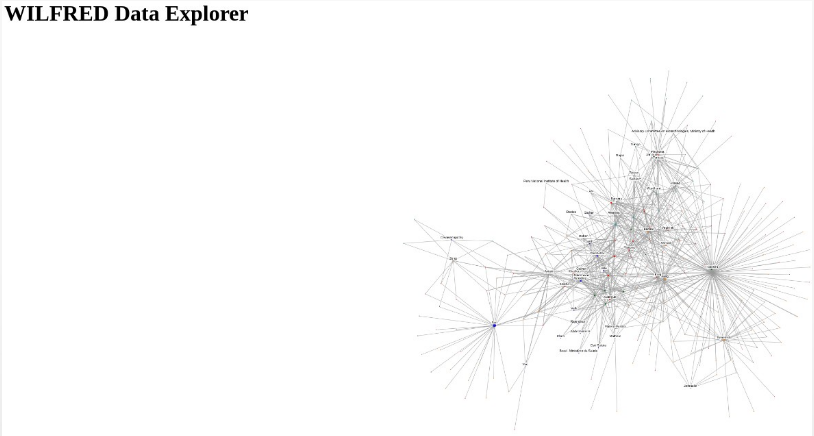
Season 1

Character data

Season data

- Season: 1
- Characters: 126
- Interactions: 549
- Density: 0.07
- Avg interactions: 8.71

https://got-network-app.herokuapp.com/

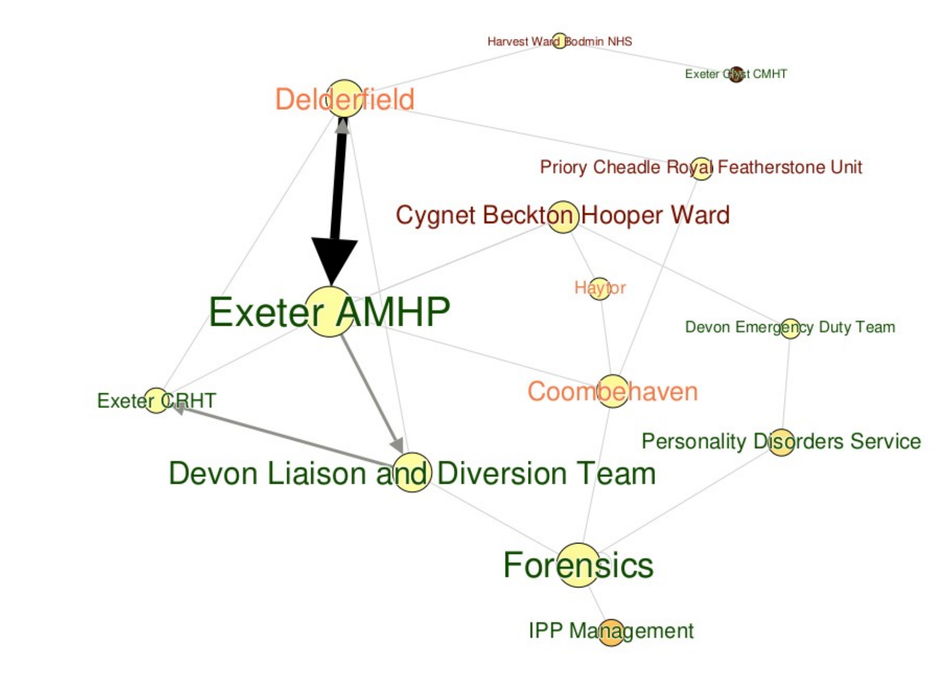


WILFRED Data Explorer

https://wilfred-data-explorer.herokuapp.com/

Network-based operational modelling for healthcare

- Graph as service structure
- Node as service
- Edge as patient movement
- Edge weight as number of patients
- Node attributes as improvement measures


Harvest Ward Bodmin NHS
Exeter Clyst CMHT
Delderfield
Priory Cheadle Royal Featherstone Unit
Cygnet Beckton Hooper Ward
Haytor
Exeter AMHP
Devon Emergency Duty Team
Exeter CRHT
Coombehaven
Personality Disorders Service
Devon Liaison and Diversion Team
Forensics
IPP Management

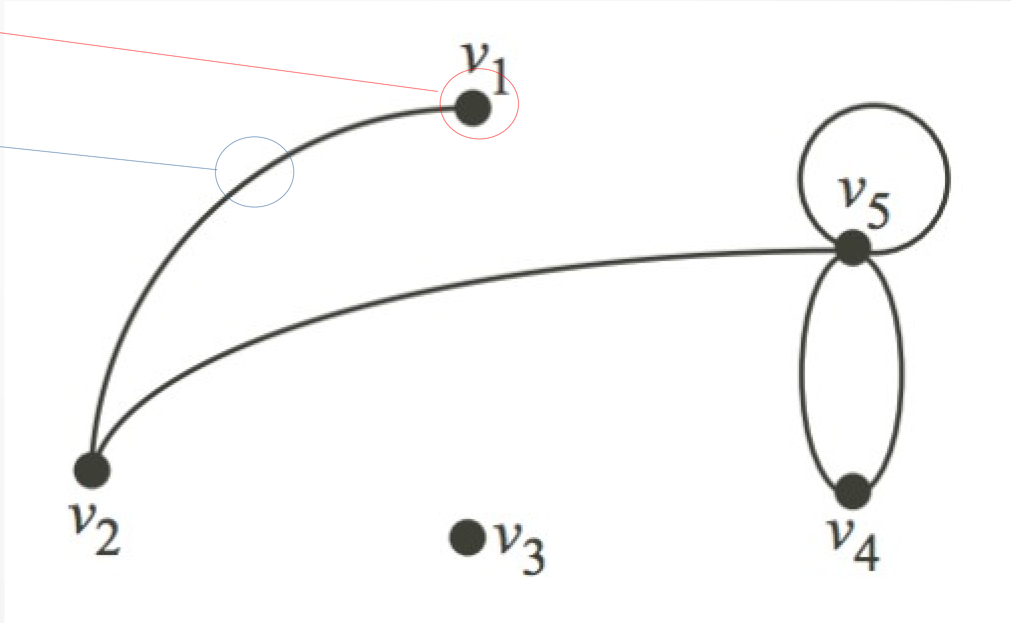
What is a network graph?

A network graph is:

- A graph based representation of a system
- Based on pairwise relationships between entities
- Quantitatively encodes the properties of the system within the network structure

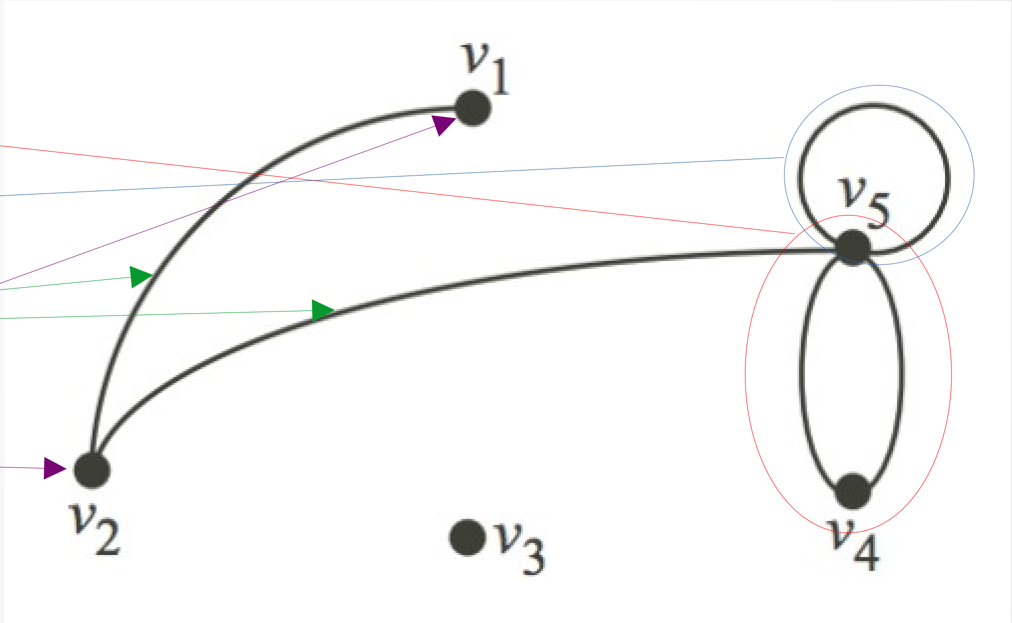
Terminology 1

- Node/ vertex (v1 → v4)
- Edge
- Graph - unordered pairs of nodes *i.e., order does not matter*
- Directed Graph (DiGraph) - ordered pairs of nodes *i.e., ED → Orthopaedics*
- Edge weight *i.e., line weight, or 'traffic'*
- Node attributes



Terminology 2

- Parallel edges
- Loop i.e., *re-referral into a service*
- Adjacent edges
- Adjacent nodes



Required input data

Data required to build a network graph

- **Node list**: ID, Label, n-attributes
- **Edge list**: Source, Target, n-attributes (e.g. Weight)

Task 1 – Exploratory Data Analysis (30 mins)

- Open the ***01_data_analysis.ipyb*** file from the ***exercises*** folder in Jupyter

- This exercise is what you make it! The notebook contains the code to import some data - it's then up to you and your PSG to interrogate and come up with some interesting facts about it. You should also think about how it could be used with respect to network analysis.

Data Transformation

Data transformation process

1) Start with the raw data table - patient level data

2) Clean the data - category consistency, negative dates, erroneous white space

3) Order the data - by patient then by date

4) Create an adjacency matrix - *more on this shortly*

5) Iterate over the interaction matrix to create edge list - source, target, weight

6) Use unique instances or services or entities as nodes - depends on the resolution of the data and how you want to subset your data

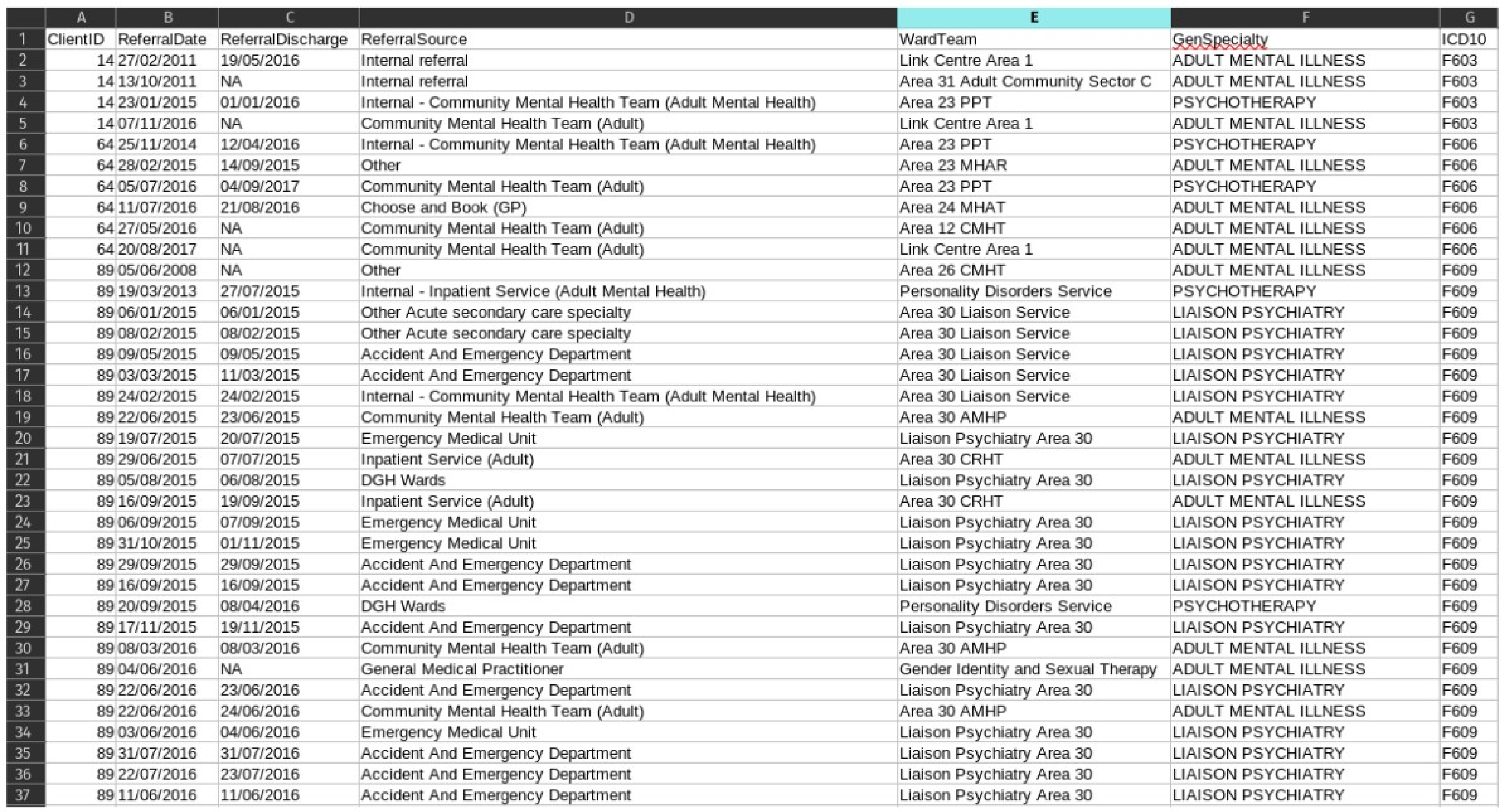
Data cleaning

Some things to consider...

- Capitalisation/ case inconsistencies *(esp if using names as IDs – suggest making all lower case)*
- Spelling mistakes *(to avoid inadvertent replications i.e., Elliot Vs Elliott)*
- Erroneous white space - leading, trailing, multiples
- Incorrect date entries - reversed, missing
- Missing data

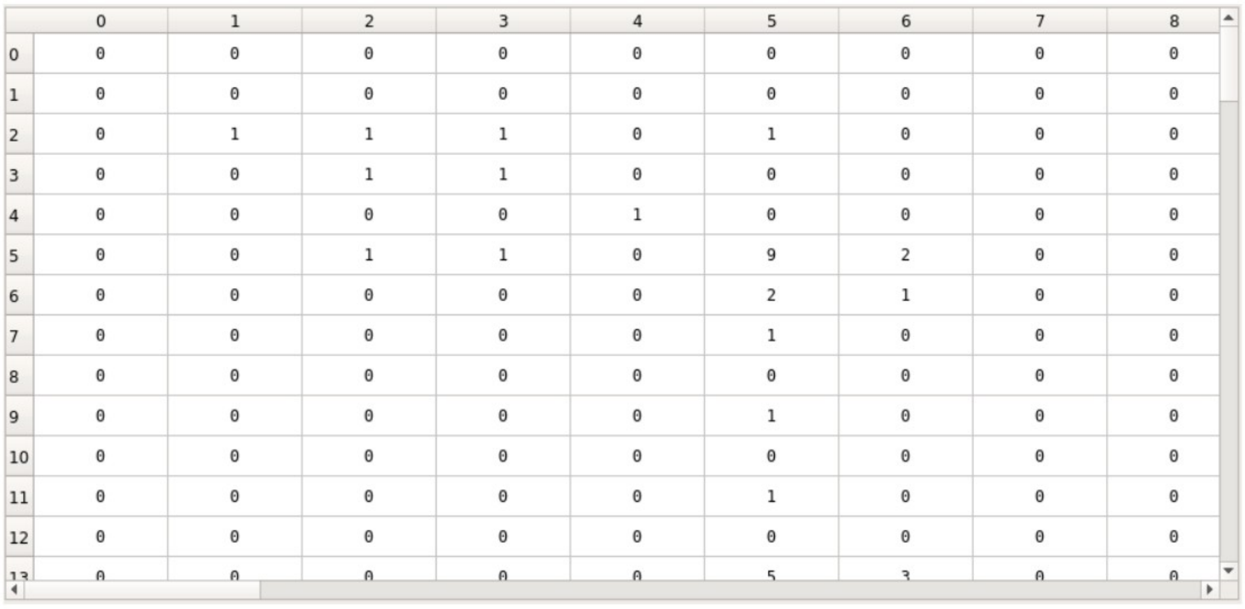
Order the Data

Order the data by patient then by date. This allows you to iterate over the data row by row to create the *adjacency matrix*. NA = current user



	A	B	C	D	E	F	G
1	ClientID	ReferralDate	ReferralDischarge	ReferralSource	WardTeam	GenSpecialty	ICD10
2	14	27/02/2011	19/05/2016	Internal referral	Link Centre Area 1	ADULT MENTAL ILLNESS	F603
3	14	13/10/2011	NA	Internal referral	Area 31 Adult Community Sector C	ADULT MENTAL ILLNESS	F603
4	14	23/01/2015	01/01/2016	Internal - Community Mental Health Team (Adult Mental Health)	Area 23 PPT	PSYCHOTHERAPY	F603
5	14	07/11/2016	NA	Community Mental Health Team (Adult)	Link Centre Area 1	ADULT MENTAL ILLNESS	F603
6	64	25/11/2014	12/04/2016	Internal - Community Mental Health Team (Adult Mental Health)	Area 23 PPT	PSYCHOTHERAPY	F606
7	64	28/02/2015	14/09/2015	Other	Area 23 MHAR	ADULT MENTAL ILLNESS	F606
8	64	05/07/2016	04/09/2017	Community Mental Health Team (Adult)	Area 23 PPT	PSYCHOTHERAPY	F606
9	64	11/07/2016	21/08/2016	Choose and Book (GP)	Area 24 MHAT	ADULT MENTAL ILLNESS	F606
10	64	27/05/2016	NA	Community Mental Health Team (Adult)	Area 12 CMHT	ADULT MENTAL ILLNESS	F606
11	64	20/08/2017	NA	Community Mental Health Team (Adult)	Link Centre Area 1	ADULT MENTAL ILLNESS	F606
12	89	05/06/2008	NA	Other	Area 26 CMHT	ADULT MENTAL ILLNESS	F609
13	89	19/03/2013	27/07/2015	Internal - Inpatient Service (Adult Mental Health)	Personality Disorders Service	PSYCHOTHERAPY	F609
14	89	06/01/2015	06/01/2015	Other Acute secondary care specialty	Area 30 Liaison Service	LIAISON PSYCHIATRY	F609
15	89	08/02/2015	08/02/2015	Other Acute secondary care specialty	Area 30 Liaison Service	LIAISON PSYCHIATRY	F609
16	89	09/05/2015	09/05/2015	Accident And Emergency Department	Area 30 Liaison Service	LIAISON PSYCHIATRY	F609
17	89	03/03/2015	11/03/2015	Accident And Emergency Department	Area 30 Liaison Service	LIAISON PSYCHIATRY	F609
18	89	24/02/2015	24/02/2015	Internal - Community Mental Health Team (Adult Mental Health)	Area 30 Liaison Service	LIAISON PSYCHIATRY	F609
19	89	22/06/2015	23/06/2015	Community Mental Health Team (Adult)	Area 30 AMHP	ADULT MENTAL ILLNESS	F609
20	89	19/07/2015	20/07/2015	Emergency Medical Unit	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
21	89	29/06/2015	07/07/2015	Inpatient Service (Adult)	Area 30 CRHT	ADULT MENTAL ILLNESS	F609
22	89	05/08/2015	06/08/2015	DGH Wards	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
23	89	16/09/2015	19/09/2015	Inpatient Service (Adult)	Area 30 CRHT	ADULT MENTAL ILLNESS	F609
24	89	06/09/2015	07/09/2015	Emergency Medical Unit	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
25	89	31/10/2015	01/11/2015	Emergency Medical Unit	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
26	89	29/09/2015	29/09/2015	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
27	89	16/09/2015	16/09/2015	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
28	89	20/09/2015	08/04/2016	DGH Wards	Personality Disorders Service	PSYCHOTHERAPY	F609
29	89	17/11/2015	19/11/2015	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
30	89	08/03/2016	08/03/2016	Community Mental Health Team (Adult)	Area 30 AMHP	ADULT MENTAL ILLNESS	F609
31	89	04/06/2016	NA	General Medical Practitioner	Gender Identity and Sexual Therapy	ADULT MENTAL ILLNESS	F609
32	89	22/06/2016	23/06/2016	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
33	89	22/06/2016	24/06/2016	Community Mental Health Team (Adult)	Area 30 AMHP	ADULT MENTAL ILLNESS	F609
34	89	03/06/2016	04/06/2016	Emergency Medical Unit	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
35	89	31/07/2016	31/07/2016	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
36	89	22/07/2016	23/07/2016	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609
37	89	11/06/2016	11/06/2016	Accident And Emergency Department	Liaison Psychiatry Area 30	LIAISON PSYCHIATRY	F609

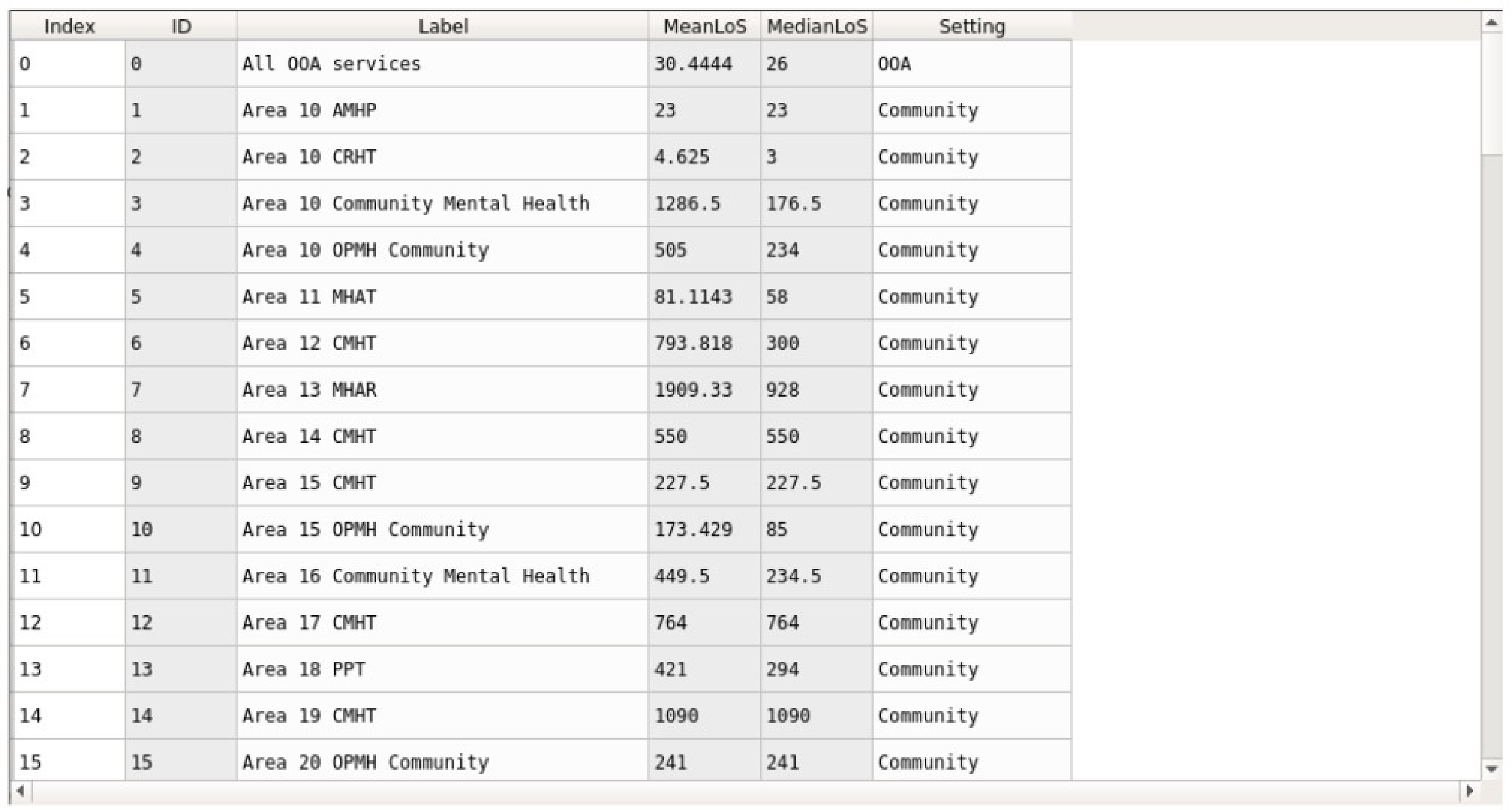
Adjacency Matrix (a.k.a. Sparse Matrix)



	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	1	1	1	0	1	0	0	0
3	0	0	1	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0
5	0	0	1	1	0	9	2	0	0
6	0	0	0	0	0	2	1	0	0
7	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	5	3	0	0

- Create a matrix of zeros n-services by n-services in size
- For each patient iterate over their data selecting the first service then the next and increment the location in the matrix referring to those services
- Source in row, target in column, frequency count in cell
- Can require a lot of computational storage (Adjacency List better)

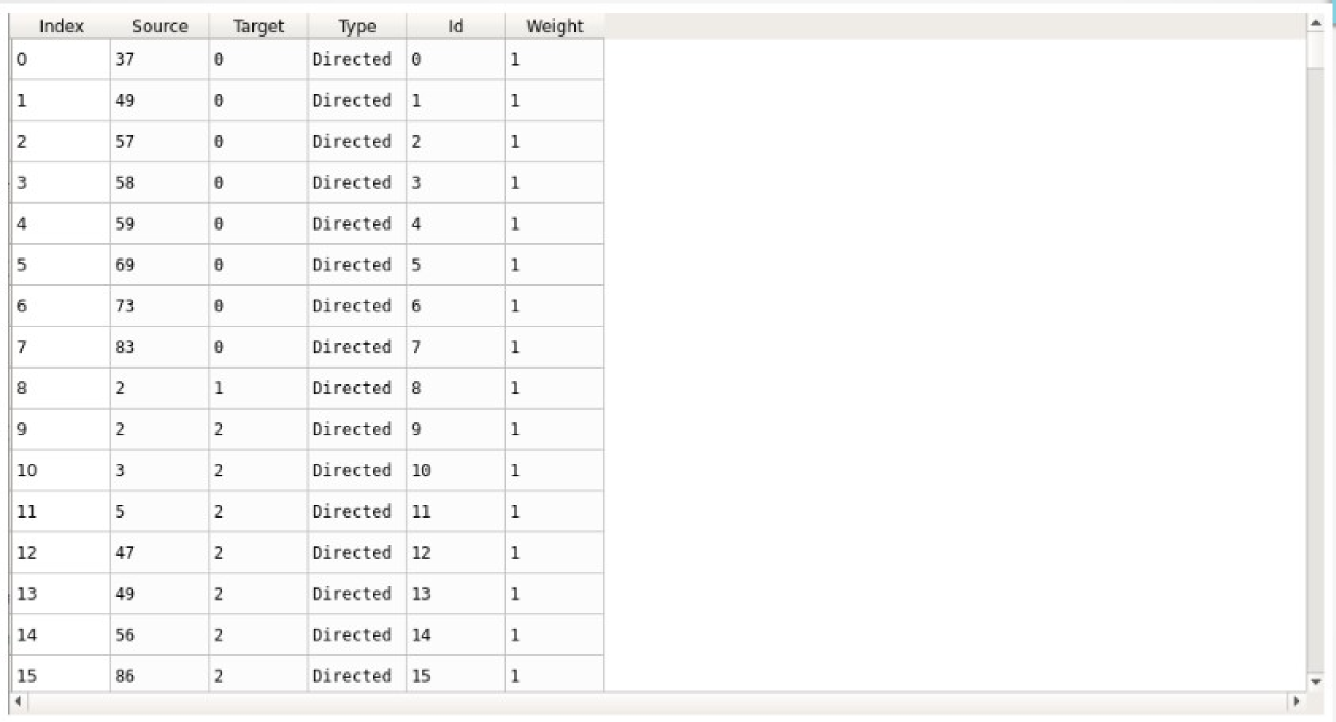
Node List



Index	ID	Label	MeanLoS	MedianLoS	Setting
0	0	All OOA services	30.4444	26	OOA
1	1	Area 10 AMHP	23	23	Community
2	2	Area 10 CRHT	4.625	3	Community
3	3	Area 10 Community Mental Health	1286.5	176.5	Community
4	4	Area 10 OPMH Community	505	234	Community
5	5	Area 11 MHAT	81.1143	58	Community
6	6	Area 12 CMHT	793.818	300	Community
7	7	Area 13 MHAR	1909.33	928	Community
8	8	Area 14 CMHT	550	550	Community
9	9	Area 15 CMHT	227.5	227.5	Community
10	10	Area 15 OPMH Community	173.429	85	Community
11	11	Area 16 Community Mental Health	449.5	234.5	Community
12	12	Area 17 CMHT	764	764	Community
13	13	Area 18 PPT	421	294	Community
14	14	Area 19 CMHT	1090	1090	Community
15	15	Area 20 OPMH Community	241	241	Community

- Required: ID - a unique numerical identifier *(required to determine source and target)*
- Optional: Label - a text string descriptor (i.e., service), continuous and discrete attributes of any type

Edge List



Index	Source	Target	Type	Id	Weight
0	37	0	Directed	0	1
1	49	0	Directed	1	1
2	57	0	Directed	2	1
3	58	0	Directed	3	1
4	59	0	Directed	4	1
5	69	0	Directed	5	1
6	73	0	Directed	6	1
7	83	0	Directed	7	1
8	2	1	Directed	8	1
9	2	2	Directed	9	1
10	3	2	Directed	10	1
11	5	2	Directed	11	1
12	47	2	Directed	12	1
13	49	2	Directed	13	1
14	56	2	Directed	14	1
15	86	2	Directed	15	1

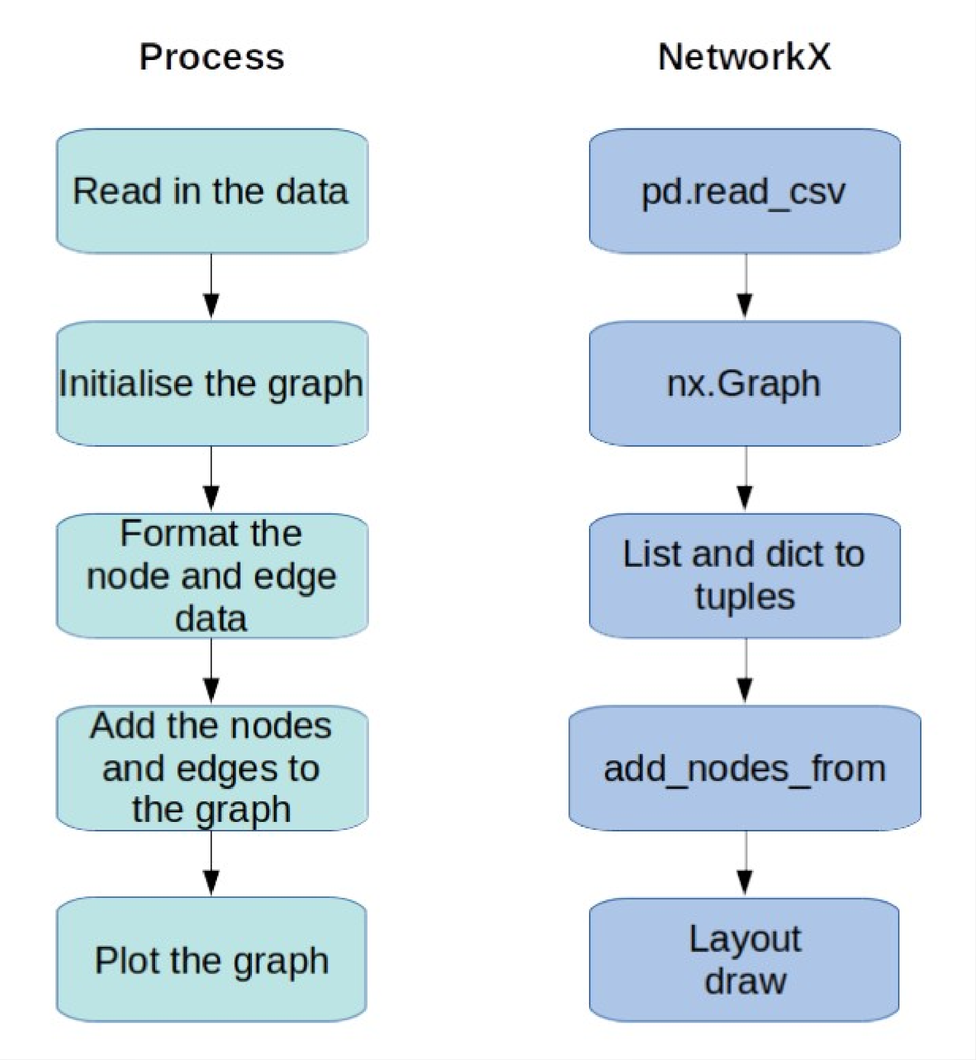
- Required: **Source** - unique node identifier, **Target** - unique node identifier, ID - unique edge identifier. *Picked out from Adjacency Matrix where cell > 0*
- Optional: Type - directed or undirected, weight - number from adjacency matrix, continuous and discrete attributes

Task 2 – Data Transformation (25 mins)

- Open the ***02_data_transformation.ipynb*** file from the ***exercises*** folder in Jupyter
- The code below provides an example of how patient data would be transformed to conduct network analysis. Everything you need is below - **however** the code has been split into **EIGHT** blocks and then jumbled up.
- Your task is to re-sort the code blocks so that they run in the correct and sequential order. Note that each of the **FOUR** main steps have further been split in to two sub-steps. This means main and subsets need to be sorted accordingly.
- As we're nice on the HSMA Programme, the library and data imports have been done for you!

Building a Network Graph

Building a Network Graph



Task 3 – Fill in The Gaps (35 mins)

- Open the ***03_build_a_network_graph.ipynb*** file from the ***exercises*** folder in Jupyter
- Update the code in the subsequent slides which contains missing information (identified with ******) (3a - 3g)
- This task requires the node and edge lists created in Task 2

Task 3a – Library & File Imports

Import the required libraries and the data we just created

```
import networkx as nx

import pandas as pd

nodes = pd.read_csv('./data/node_list.csv'
,low_memory=False)

edges = pd.read_csv(**, low_memory=False)
```

Task 3b – Instantiate Graph Object

Initialise a graph object which we will assign to the variable **G**

```
    ** = nx.Graph()
```

Task 3c – Prepare Nodes

Node inputs are tuples of the node ID and a dictionary of attributes

Node tuples (list of ID, dict of attributes)

```
idList = nodes[**].tolist()

labels = pd.DataFrame(nodes['Label'])

labelDicts = **.to_dict(orient='records')

nodeTuples = [tuple(r) for r in zip(idList, labelDicts)]
```

Task 3d – Prepare Edges and Weights

Edge inputs are tuples of source ID, target ID and a dictionary of attributes

Edge tuples (Source, Target, dict of attributes)

```
sourceList = edges[**].tolist()

targetList = edges['Target'].tolist()

weights = pd.DataFrame(edges[**])

weightDicts = **.to_dict(orient='records')

edgeTuples =[tuple(r) for r in zip(sourceList,**,weightDicts)]
```

Task 3e – Add Nodes & Edges to Graph Object

We add the nodes and edges to the graph object **G**

```
G.add_nodes_from(**)

G.add_edges_from(**)
```

Task 3f – Determine Node Coordinates

We pass the graph object to a layout algorithm to determine the node coordinates and hence the edge start and end points.

Various ways to do this… Some layouts are geometric (i.e., cyclical or hierarchical), some fare force based layouts (working on physics engines i.e. proximity and gravity)

```
    pos = nx.kamada_kawai_layout(**)
```

Task 3g – Plot the Graph

Finally we draw the graph passing in the graph object G, the position object ‘pos’ and adjust the node size. Don’t forget the labels!

```
plot = nx.draw(**, pos=**, node_size=100,

    with_labels=True, font_size=10)
```

Next Time (4C and 4D)

Subsequent sessions will cover…

- Different types of graph
- Graph metrics and their interpretation
- How graph visualisations work
- Graph algorithms and custom layouts
- Interactive graphs

Further Info & Acknowledgement

Slides adapted from Sean Manzi

Checkout https://www.project-nom.com for more information and training on the use of network-based operational modelling for whole system modelling in healthcare