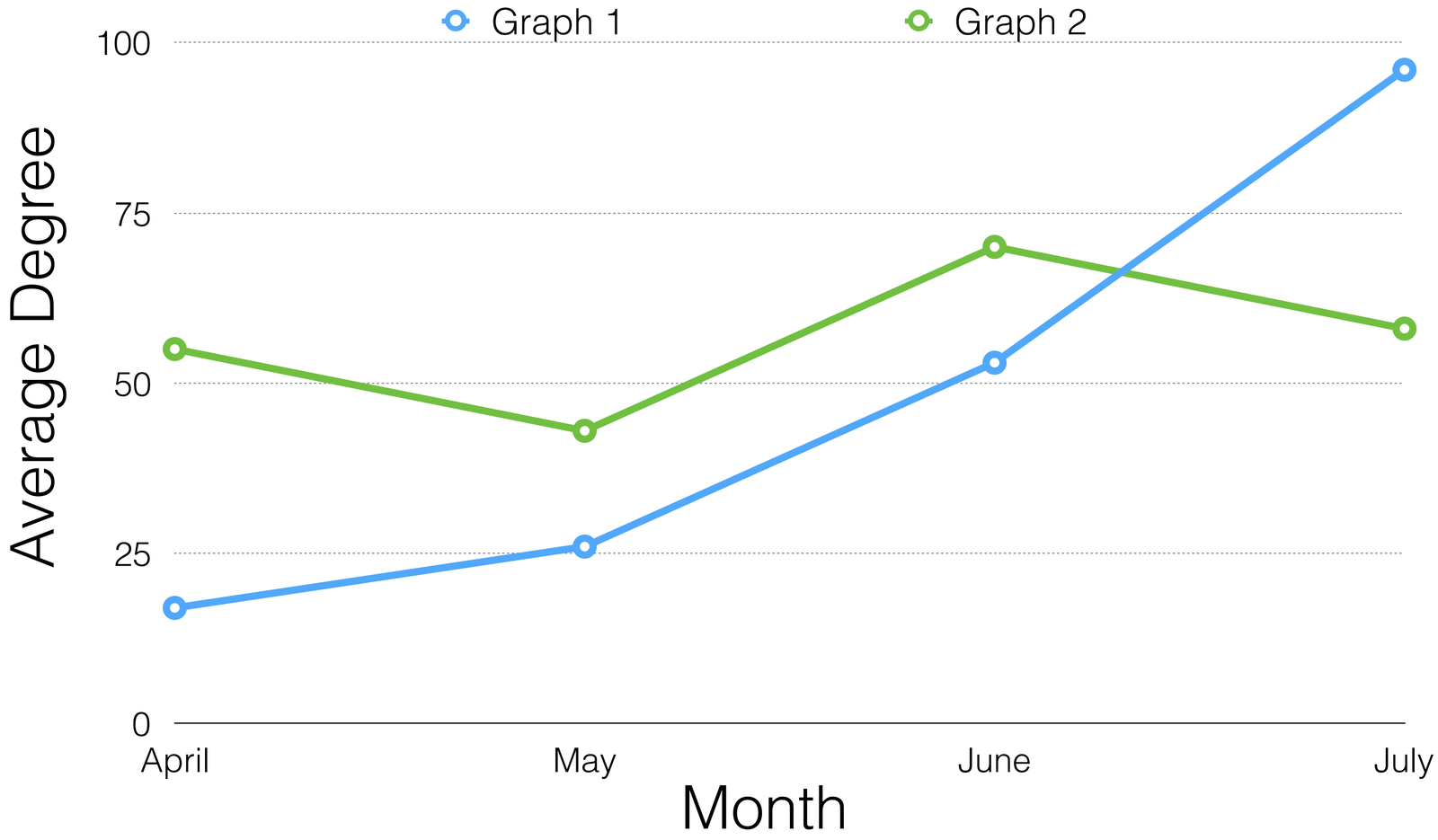
NETWORK ANALYSIS IN PYTHON II

Introduction to graph differences

Time series analysis



Time series analysis

- How some number changes as a function of time
  - Is there an upward or downward trend?
- Rate of change of things over a sliding window of time
- Examples:
  - Tracking weight over time
  - Tracking stock investment portfolio value over time

Evolving graphs

- Graphs that change over time: communication networks
- Assumptions:
  - Edge changes over time; assume nodes stay constant
  - Both edges and nodes change over time

Graph differences

- Graphs are comprised of:
  - A node set
  - An edge set
- If a node set doesn't change:
  - Changing only the edge set will result in a change in the graph

Graph differences

- Analogy: set differences

```
set(c1, c2, c3).difference(set(c2, c3, c4)) = set(c1)

set(c2, c3, c4).difference(set(c1, c2, c3)) = set(c4)
```

- In NetworkX: .difference(G1, G2) function
  - Assumes G1 and G2 have equal node sets

Graph differences in Python

```
In [1]: G1.edges()
Out[1]: [('cust1', 'cust2'), ('cust3', 'cust2')]

In [2]: G2.edges()
Out[2]: [('cust1', 'cust3'), ('cust3', 'cust2')]

In [3]: G2minusG1 = nx.difference(G2, G1)

In [4]: G1minusG2 = nx.difference(G1, G2)
```

NETWORK ANALYSIS IN PYTHON II

Let’s practice!

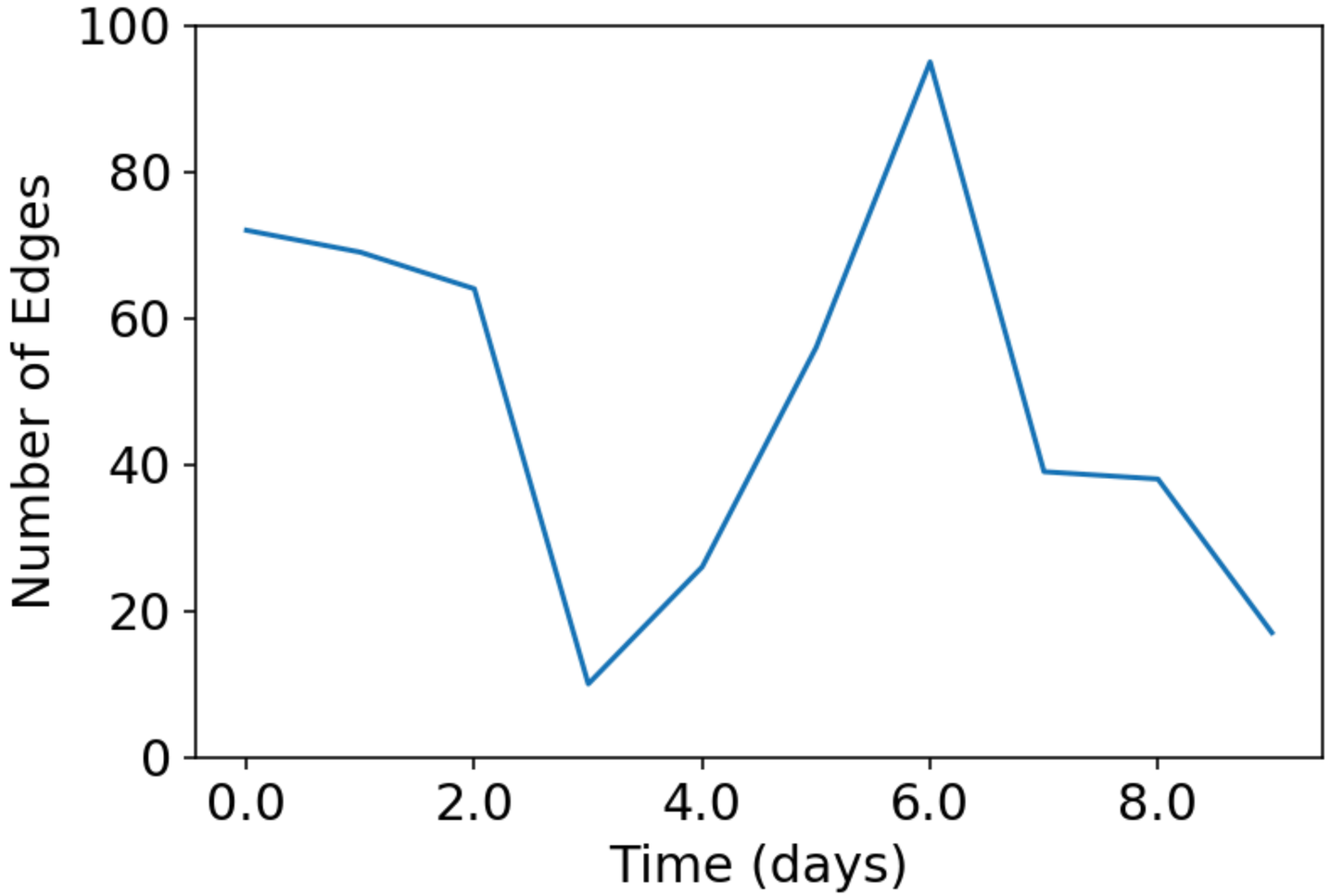
NETWORK ANALYSIS IN PYTHON II

Evolving graph statistics

Evolving graph statistics

- Graph summary statistics:
  - Number of nodes
  - Number of edges
  - Degree distribution
  - Centrality distributions

Evolving graph statistics



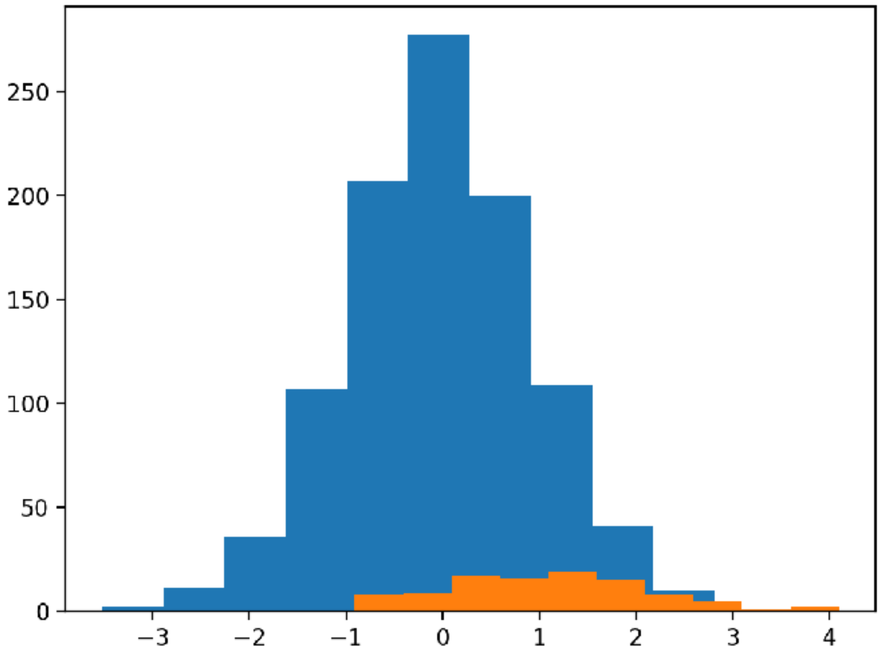
Evolving graph statistics

- For simple metrics, use edgelist data
- For graph theoretic metrics, use graph object

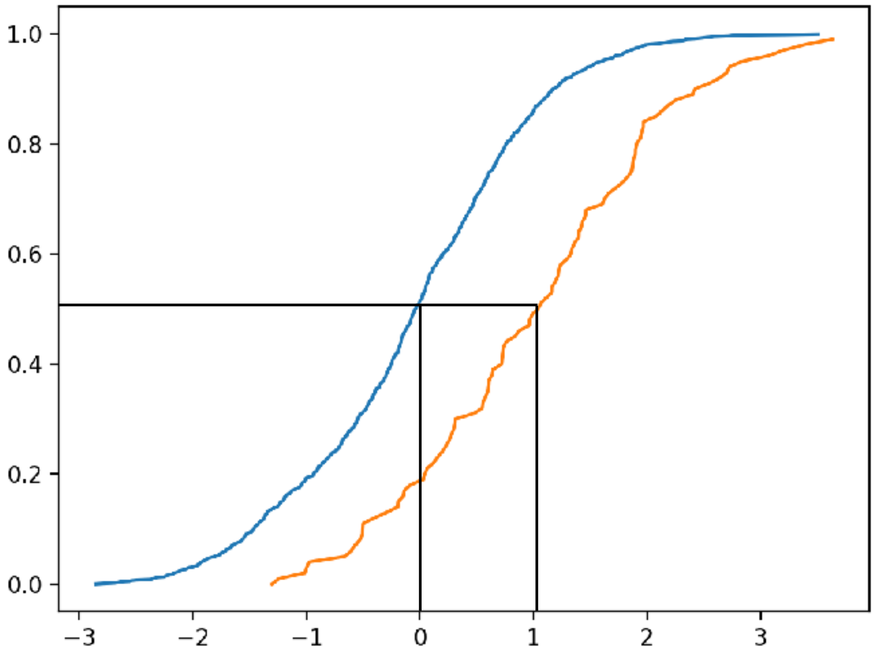
Cumulative distribution

- Compact way of representing the distribution of values

Histogram



Cumulative Dist.



NETWORK ANALYSIS IN PYTHON II

Let’s practice!

NETWORK ANALYSIS IN PYTHON II

Zooming in & zooming out: Overall graph summary

Graph exploration at scales

- Exploration at global and local scales
- Global: Centrality distributions
- Local: Connectivity and structures

Zooming on nodes

- Isolate a given node or set of nodes
- Plot node statistic over time

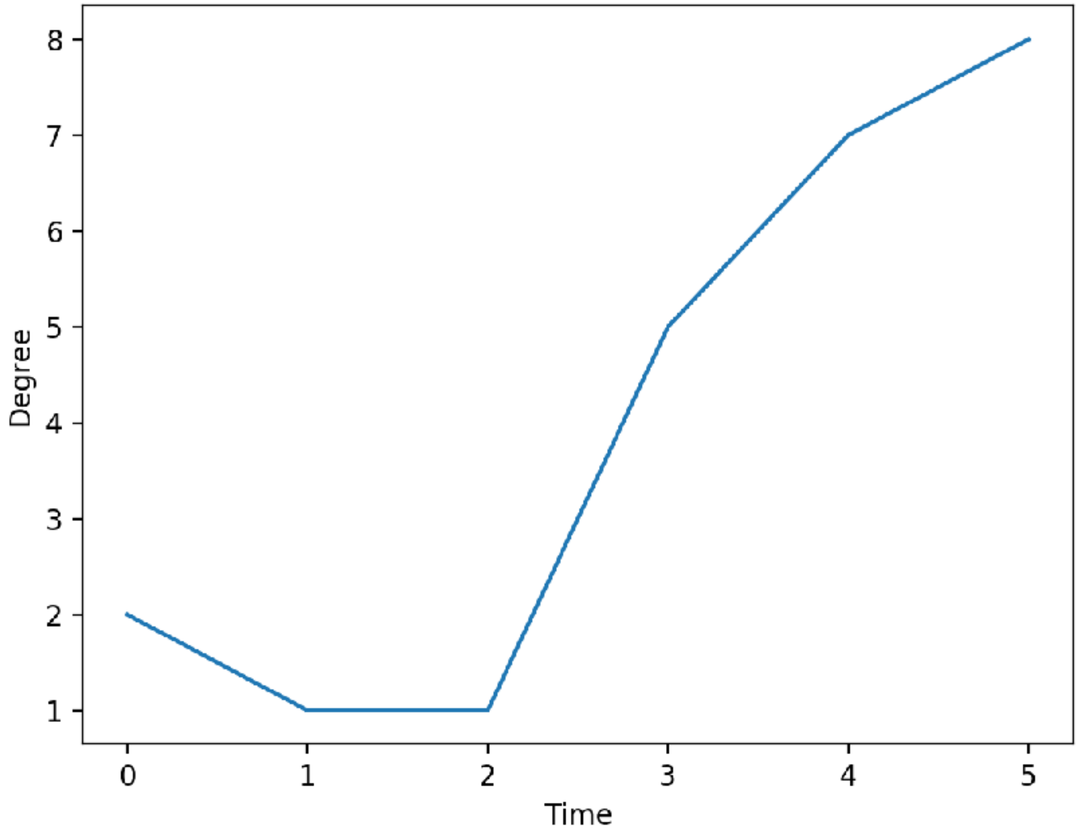
Summarizing evolving node statistics

- Customer-product dataset
  - Investigate how purchasing patterns have changed over time
- 'customer1' - node of interest

Summarizing evolving node statistics

```
In [1]: Gs = [....]

In [2]: noi = 'customer1'

In [3]: degs = []

In [4]: for g in Gs:
...         # Get the degree of the node
...         degs.append(len(g.neighbors(noi)))

In [5]: plt.plot(degs)

In [6]: plt.show()
```

Summarizing evolving node statistics



Default dictionaries

```
In [7]: from collections import defaultdict

In [8]: d = defaultdict(list)

In [9]: d['heathrow'].append(0.31)

In [10]: d['heathrow'].append(0.84)

In [11]: d
Out[11]: defaultdict(list, {'heathrow': [0.31, 0.84]})
```

Default dictionaries

```
In [12]: d2 = dict()

In [13]: d2['heathrow'].append(0.31)
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-19-291c74368a8f> in <module>()
----> 1 d2['heathrow'].append(0.31)

KeyError: 'heathrow'
```

NETWORK ANALYSIS IN PYTHON II

Let’s practice!