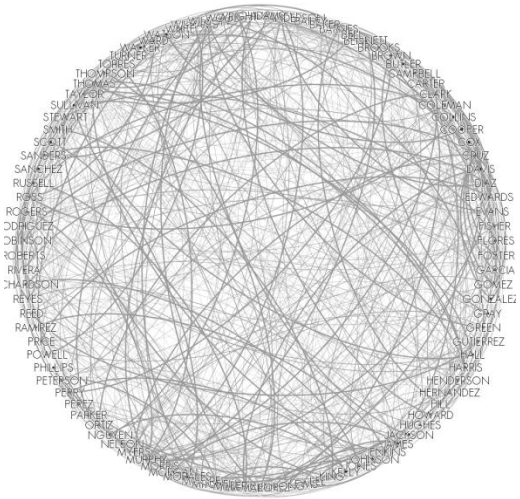
R, IGRAPH, AND HANDSOME GRAPHS

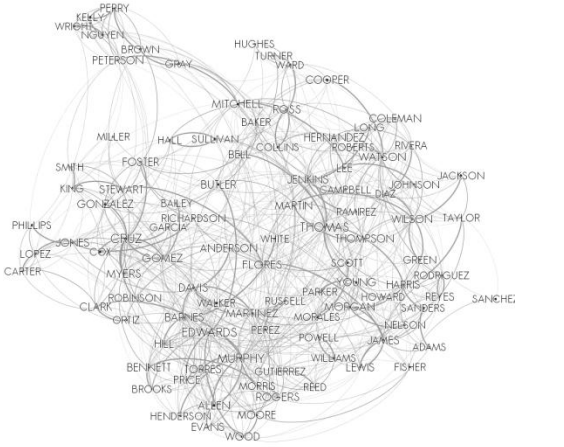
adam hogan
Design & Analytics
Presentation for the Chicago R-Users Group

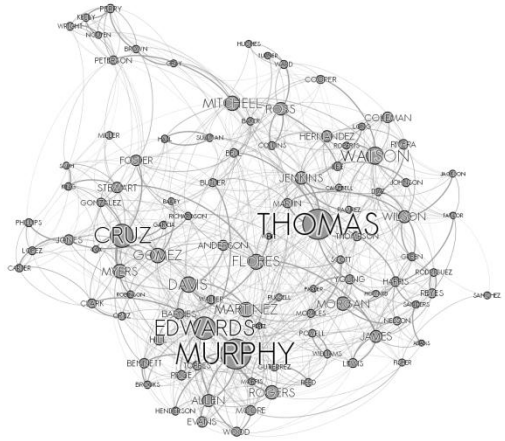
Chicago, IL
October 3, 2012

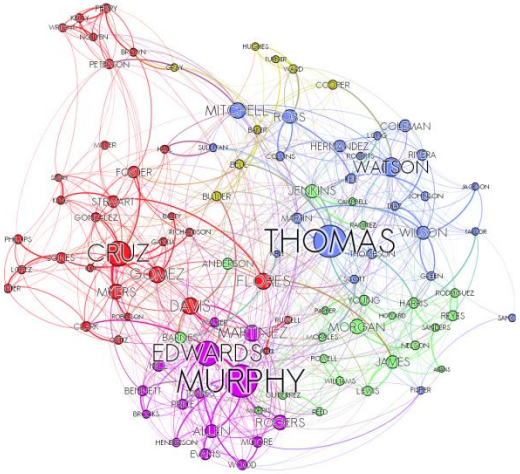
OVERVIEW

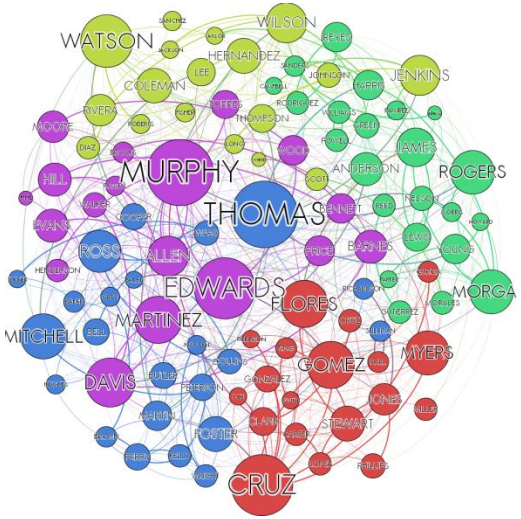
SOURCE	TARGET	WEIGHT
Adams	Allen	1
Adams	Anderson	2
Adams	Bailey	3
Allen	Adams	1
Allen	Bailey	2
Anderson	Adams	2
…	…	…





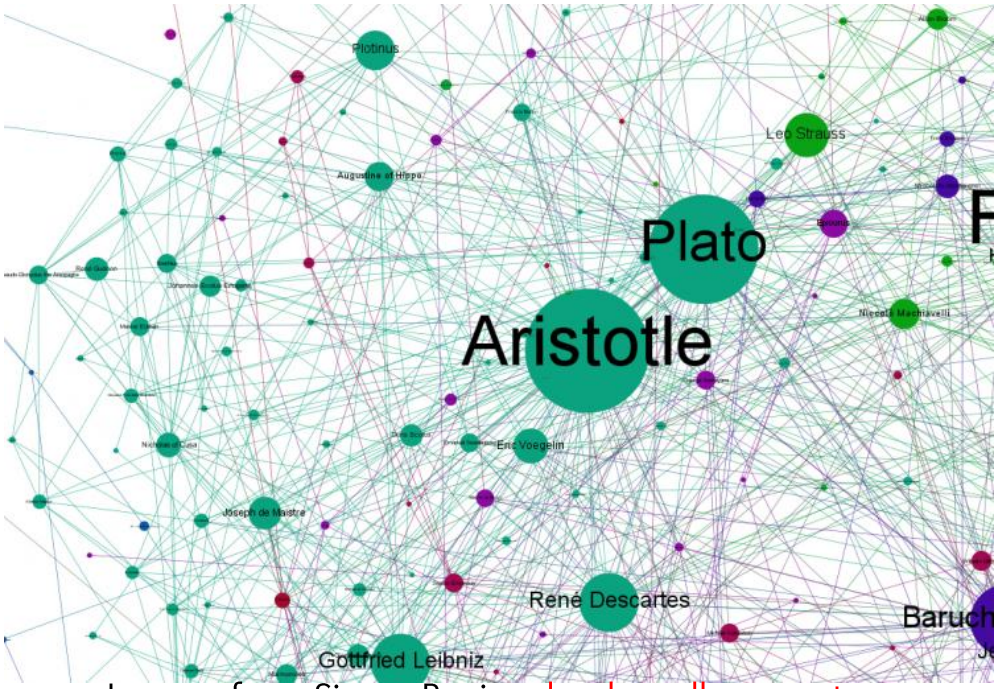




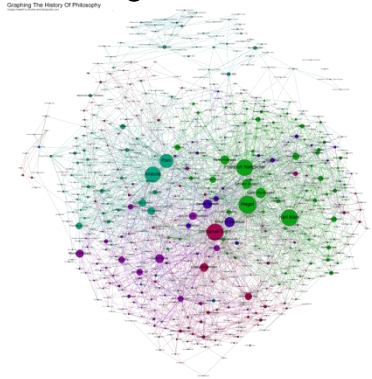


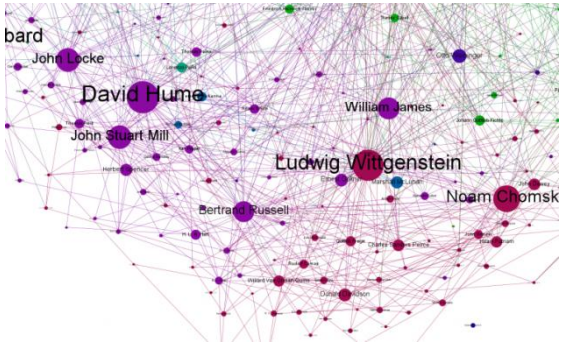
HISTORY OF PHILOSOPHY

- Query + idea from drunksandlampposts Beautiful stuff.
- Wanted to try directed graphs, other SNA algorithms.
- Replicate, automate, and exploRe.



Images from Simon Rapier, drunksandlampposts.com





GET THE DATA: THE QUERY

- There are some quirks:
  - Doesn't just get all philosophers influenced by philosophers
  - Some abstract nouns
  - ...but an excellent demo.
- SPARQL is an RDF query language used by Wikipedia's DB

```
# Create the query
qq <- 'SELECT * WHERE {
?p a
<http://dbpedia.org/ontology/Philosopher> . ?p
<http://dbpedia.org/ontology/influenced>
?influenced. }'


# Use it in SPARQL
data <-
SPARQL(url='http://dbpedia.org/sparql',query=qq)
```

GET THE DATA: CLEANING

- Cleaning and data munging script is available on the Design & Analytics blog.
- Careful!
  - The signs are tricky.
  - Originator of influence is the TARGET of influence-citation, not the SOURCE.

```
# Prep to make it directed
orig <- unlist(data$results[[2]],
use.names=F)
dest <- unlist(data$results[[1]],
use.names=F)

# Turn URLs into readable names
orig <- url2names(orig)
dest <- url2names(dest)

# Format as an edge graph.
edges <-
data.frame(cbind(as.matrix(orig),as.
matrix(dest), rep(1,length(orig))),
stringsAsFactors=F)

# TADA!
```

R TOOLS

- Once in R, we have a few package choices:
  - igraph – Good documentation here, has plenty of algorithms, and is cross-language with Python. Strong export libraries for communicating with other tools.
  - SNA – Tailored for social networks---has p* and some other useful algorithms.
  - network – What's under the hood of SNA.

USING IGRAPH

- We want:
  - Labels
  - Communities
  - Colors
  - Authority
- Careful!
  - Syntax
    - `V(g) <- values`
    - `set.vertex.attribute(g,values)`
  - Sometimes 0-indexed instead of 1-indexed
  - Differences in some algorithm results between igraph and gephi.

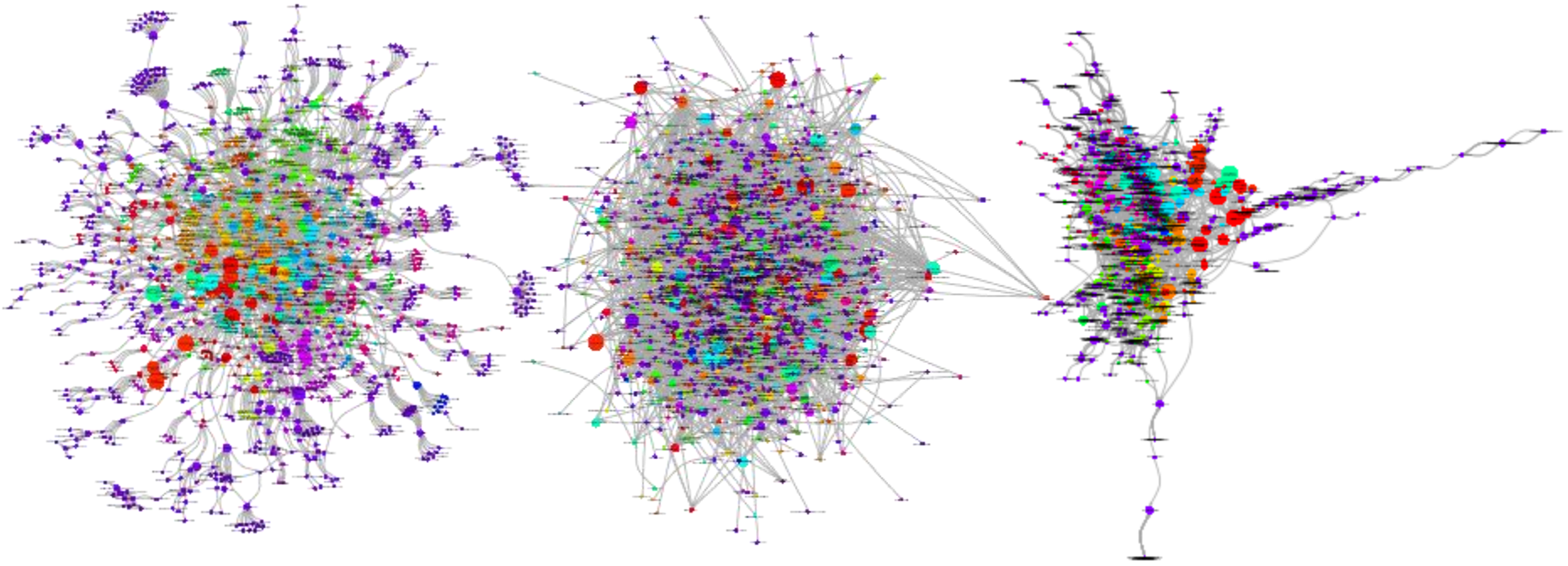
```
g <- graph.data.frame(edges, direct=TRUE, vertices=NULL)

SNAbasics <- function(X) {
V(X)$label <- V(X)$name
comps <- leading.eigenvector.community(X, options=list(naive=F))$membership
colbar <- rainbow(max(comps)+1)
V(X)$color <- colbar[comps+1]
# REMOVE OPACITY
V(X)$color <- substr(V(X)$color,1,7)
RPageRank <- page.rank(X, vids=V(X), directed=TRUE,
options=list(maxiter=10000,eps=0.0001))$vector
V(X)$RPageRank <- RPageRank*1000
V(X)$size <- RPageRank*1000
V(X)$Katz <- evcent(X)$vector*1000
print(length(unique(comps)))
return(X)
}
```

VISUALIZATION LAYOUTS

Fruchterman-Reingold

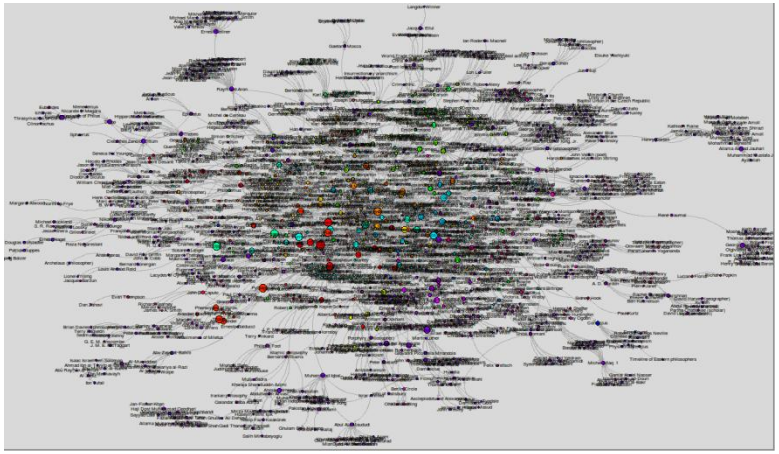
Kamada Kawai

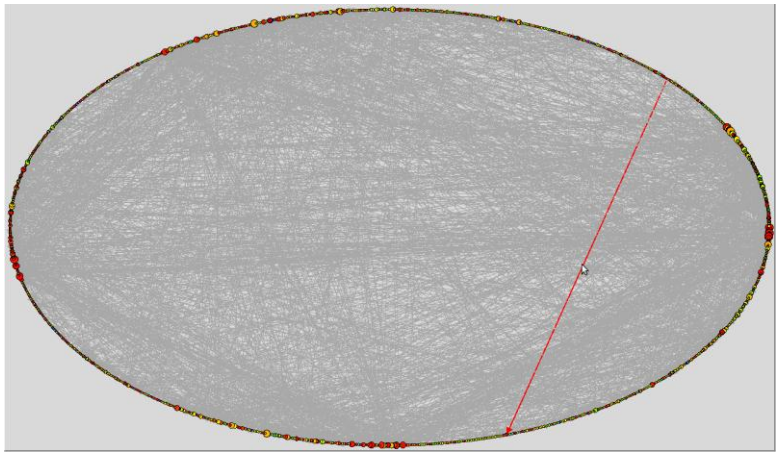
Multi-Dim Scaling



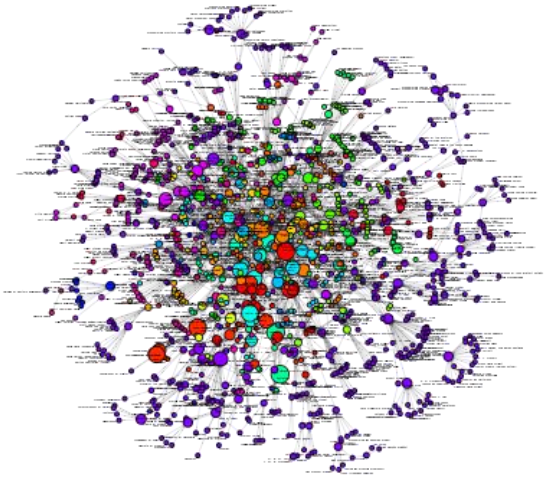
R VISUALIZATION OPTIONS

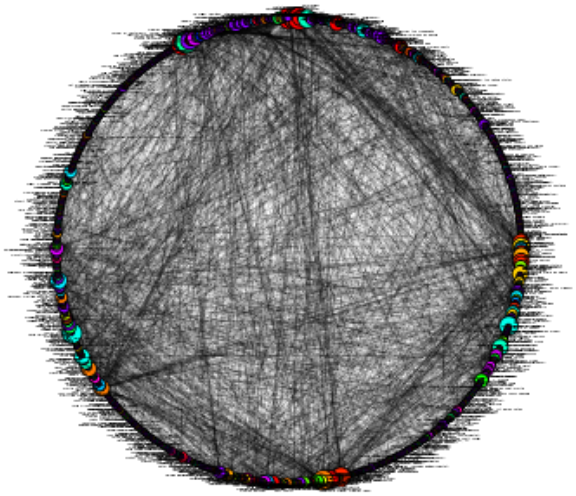
Interactive with TK





Network





R2OTHER

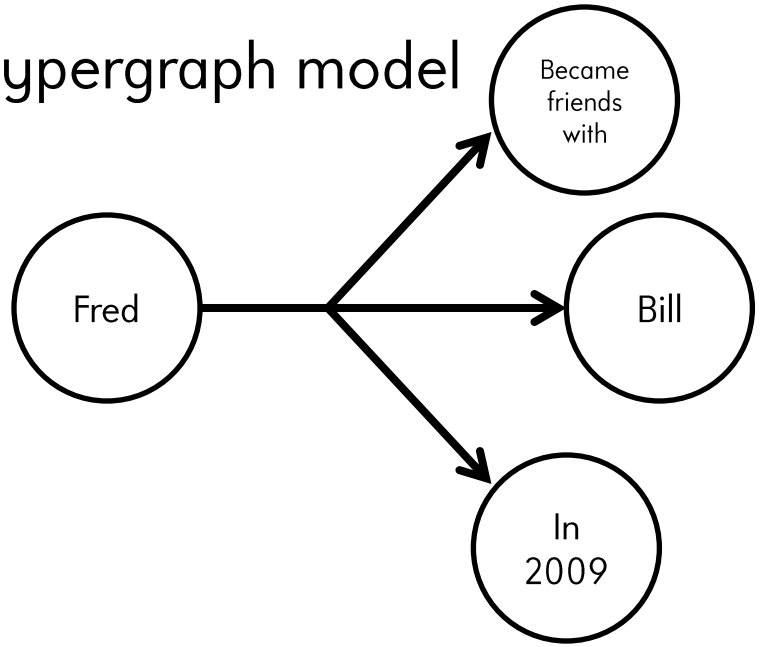
- Need persistence?
  - Output to a graph database (probably through XML)
- Need artistic control over placement?
  - Output to Gephi
- Need hierarchy?
  - Output to dot/graphviz
- Need mass graphical gratification?
  - Output to web with a JSON string
  - Easy: `python -m SimpleHTTPServer 8000`
    - D3.js: more control to make dashboards and UIs with js
    - Sigma.js lets you use gephi's GEXF files immediately

GRAPH DATABASES

- Attribute dbs:
  - Means 1 edge can have multiple properties
  - Example: Neo4j
- Hypergraph dbs:
  - Means edges can have a multi-node range.
  - Examples: HypergraphDB, Microsoft Trinity
  - Careful! A brilliant idea, but good luck re-writing your traversal algorithms. NON-TRIVIAL

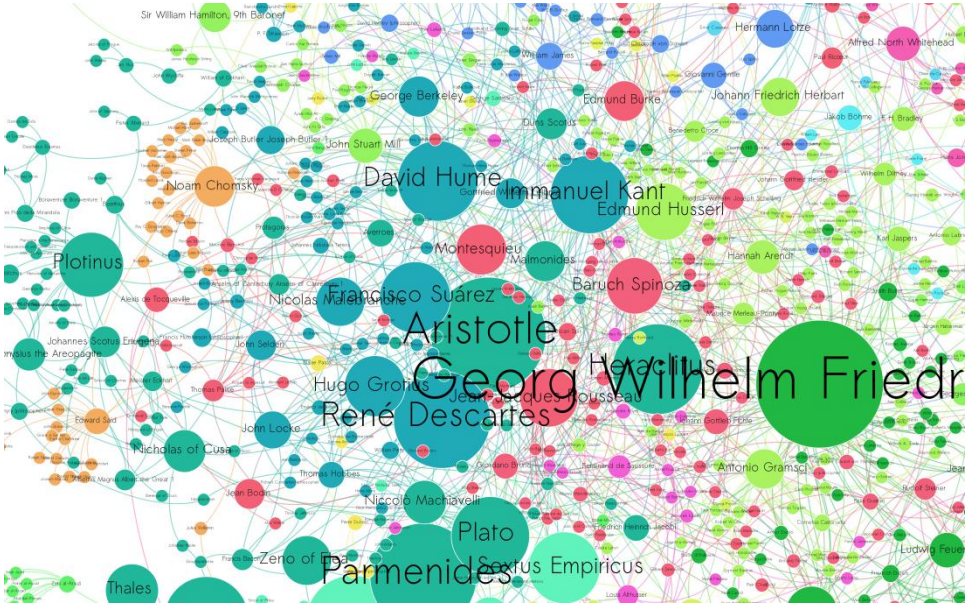
Edge Attribute model

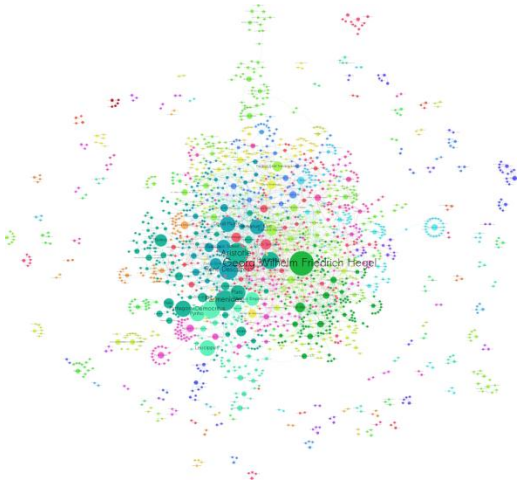
Hypergraph model



TO GEPHI

- Looks like this -->
- Careful!
  - Use GML format to preserve node attributes from R analysis.
    - `write.graph(g, file='gephi.GML', format='GML')`
  - Use just an edge list for easiest export.

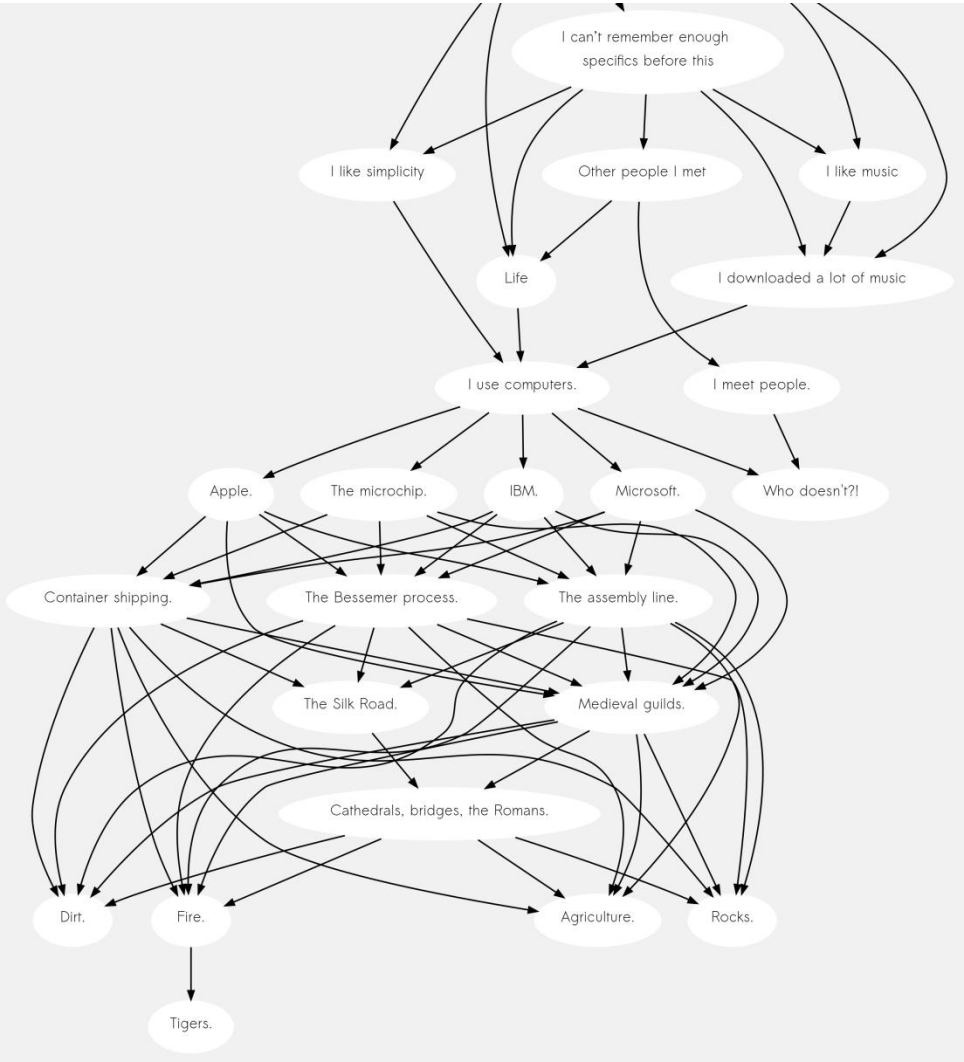




For full version...

TO GRAPHVIZ (WITH DOT)

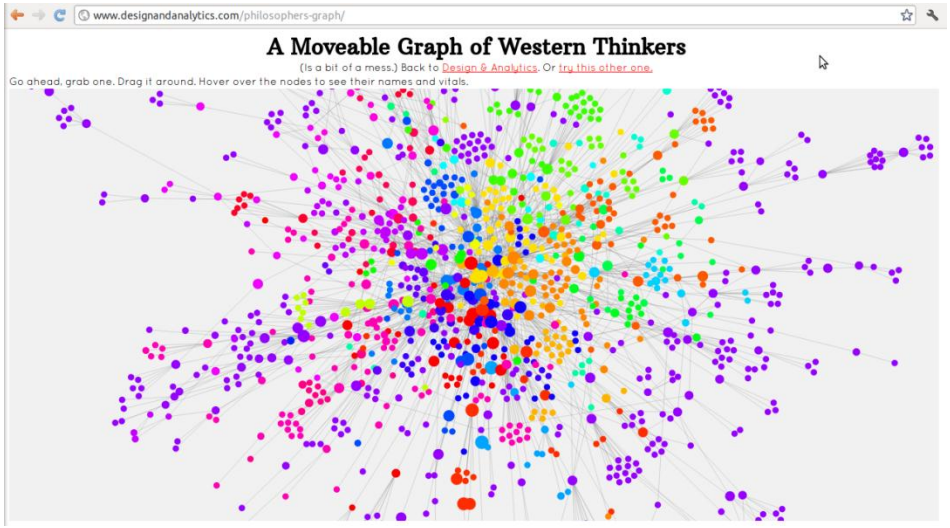
- Looks like this -->
- (Different graph data: the previous wasn't hierarchical.)
- Use graphviz layouts if you want to express flow.
- Format accessible from igraph export and gephi export

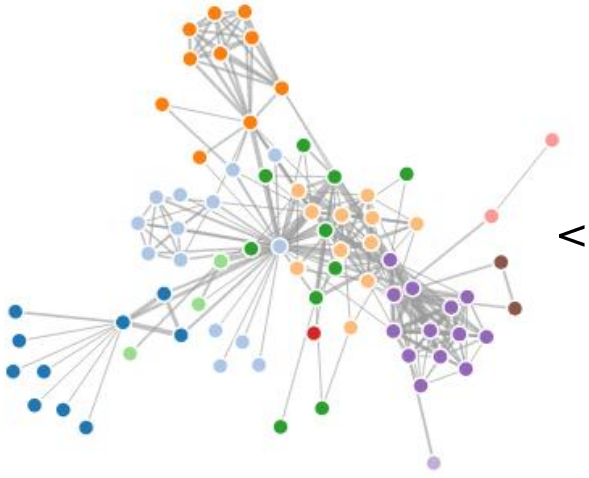


For full version...

JSON -> D3

- Looks like this -->
- Libraries to choose from:
  - "RJSON"
  - "RJSONIO"
  - Write your own script (or use mine)
- Careful!
  - `As.numeric`
  - `stringsAsFactors =FALSE`
  - Exported nodes and edges separately, pasted together by hand.

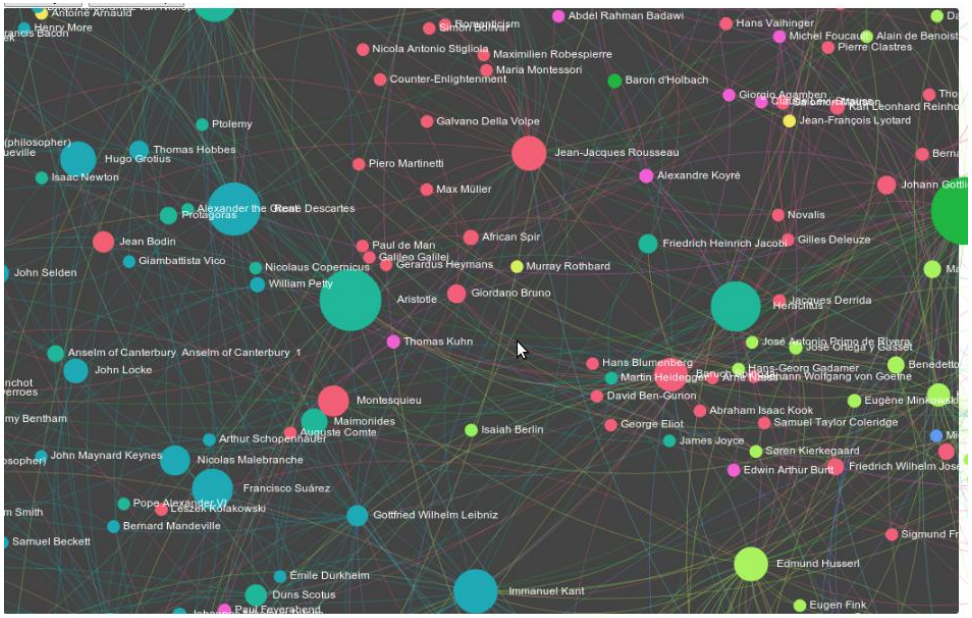


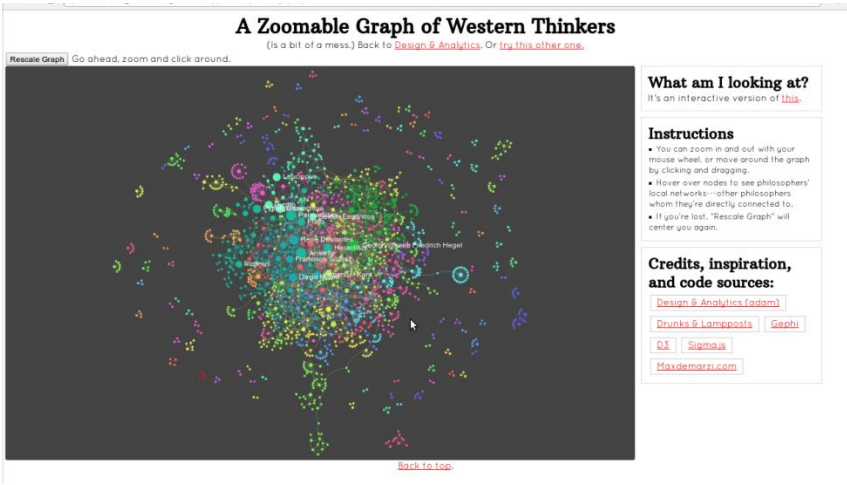


<--Mike Bostock, D3
Tim Dwyer
Thomas Jakobson
(Donald Knuth)
(Victor Hugo!)

GEPHI -> SIGMA.JS

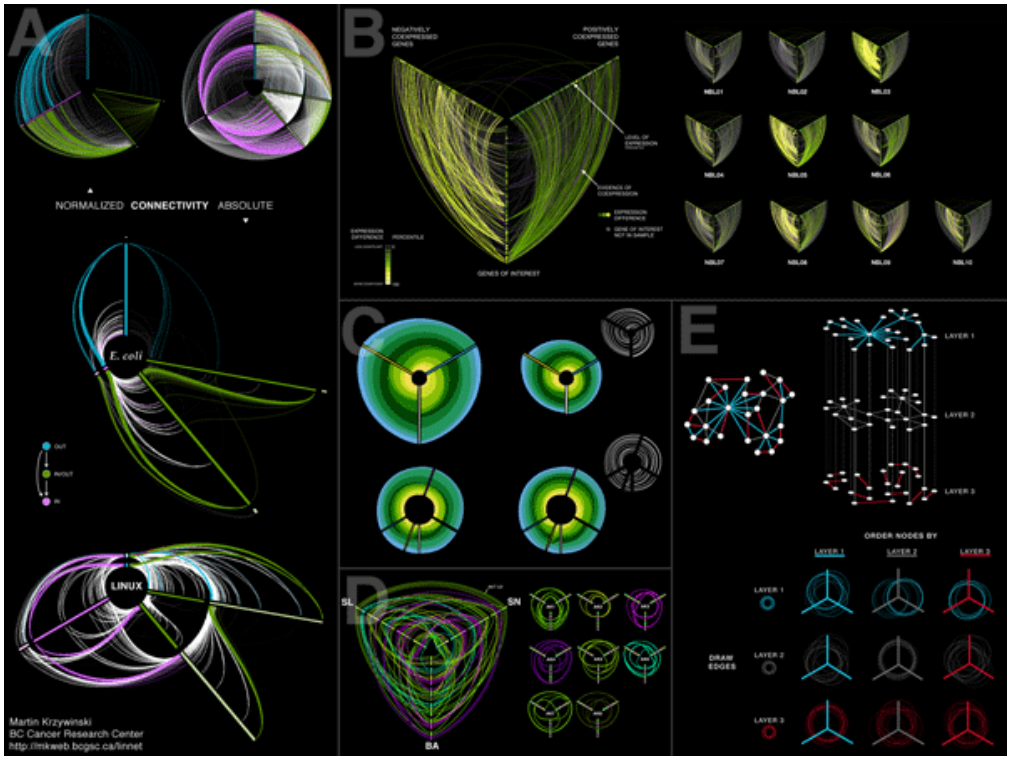
- Looks like this -->
- The advantage of this output is that you can use pre-rendered gephi graphs, and let others explore them interactively.
- Stored in one gexf file + minimal HTML + javascript.





HAIRBALLS AND HIVES

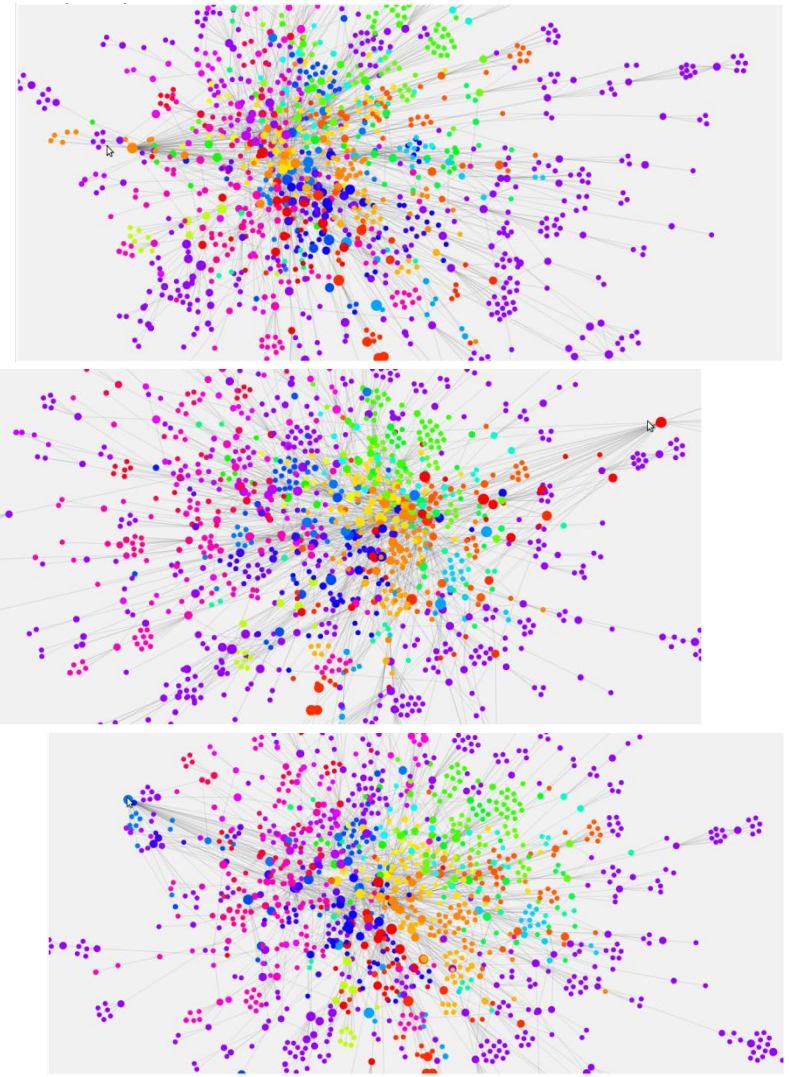
- There's a good argument against "hairball-style" visualizations, best stated by Martin Krzywinski
- He created "hive plots."



Krzywinski M, Birol I, Jones S, Marra M (2011). Hive Plots — Rational Approach to Visualizing Networks. Briefings in Bioinformatics (early access 9 December 2011, doi: 10.1093/bib/bbr069)

INTERACTIVE HAIRBALLS?

- But interactivity is a big deal. You get more than just another dimension.
- Not a graph like a line graph, but a map like a wall map.
  - It is too big to take in all at once---and that can be OK.
- "Tactile" networks
- Hegel/Aristotle demo



CHALLENGES!

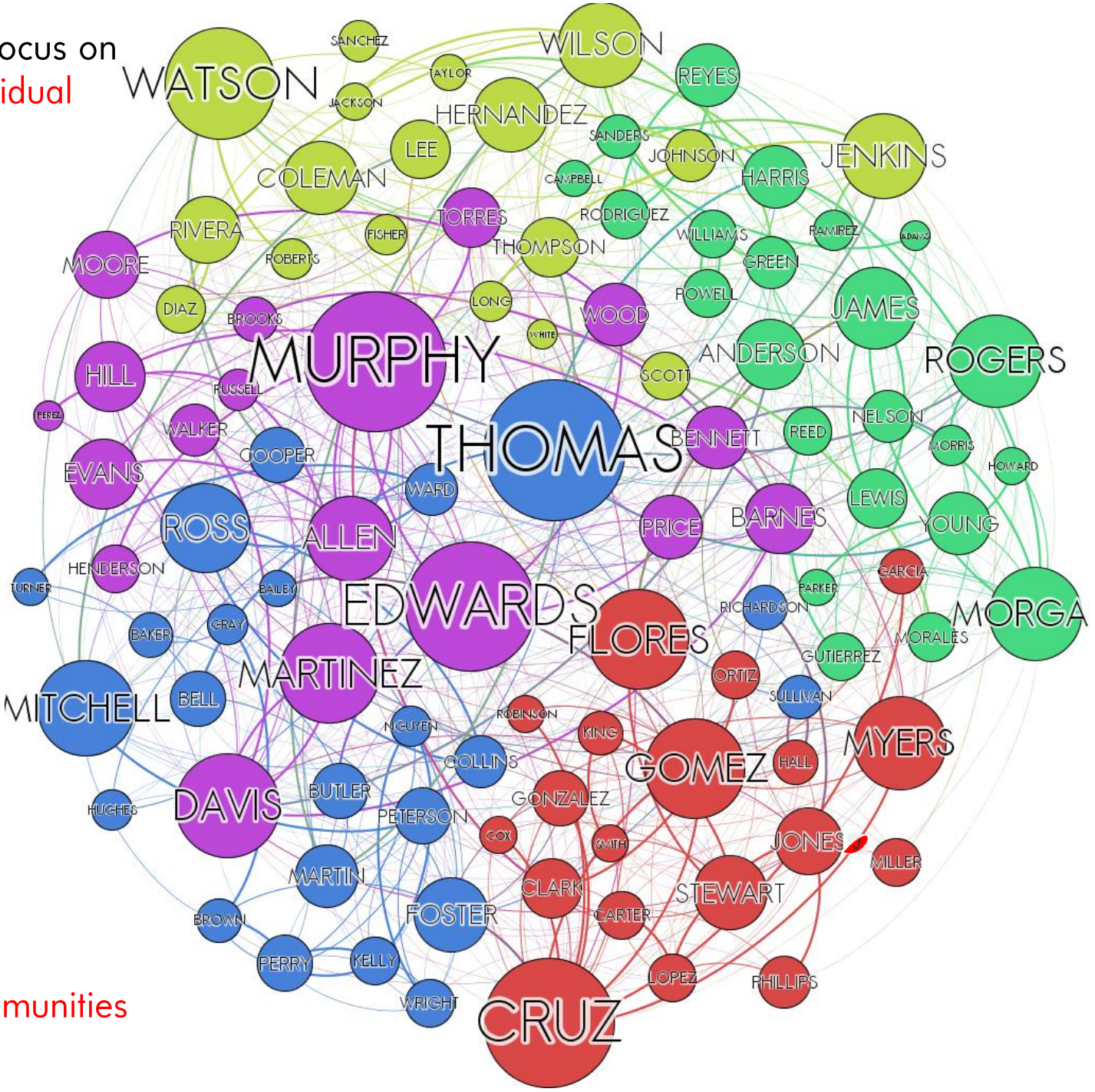
- R challenges: general output `graph2web()`
  - Clean up the JSON export to a D3-acceptable format
  - One-line, easy output to local web server?
    - `PlotInMyBrowser(mydata,myport)`
  - Performance implementation for large graphs with D3?
- Clean up the data of the philosopher database
  - On a hosted, open repo graph database
- Prediction: we should take javascript more seriously. Interactive visualization is a watershed.

Focus on individual members.

and relationships between them.

Discover influence

Detect communities



THANKS

- Data manipulation in R, gephi, python, the wrangler.
- Graph visualization in Gephi, D3, graphviz, sigma.js.
- Data from anonymized and public sources.
- Code snippets from lots of people, sorry if I missed crediting anyone

Isn't this stuff cool? Contact me.

adam hogan

adam.hogan@designandanalytics.com