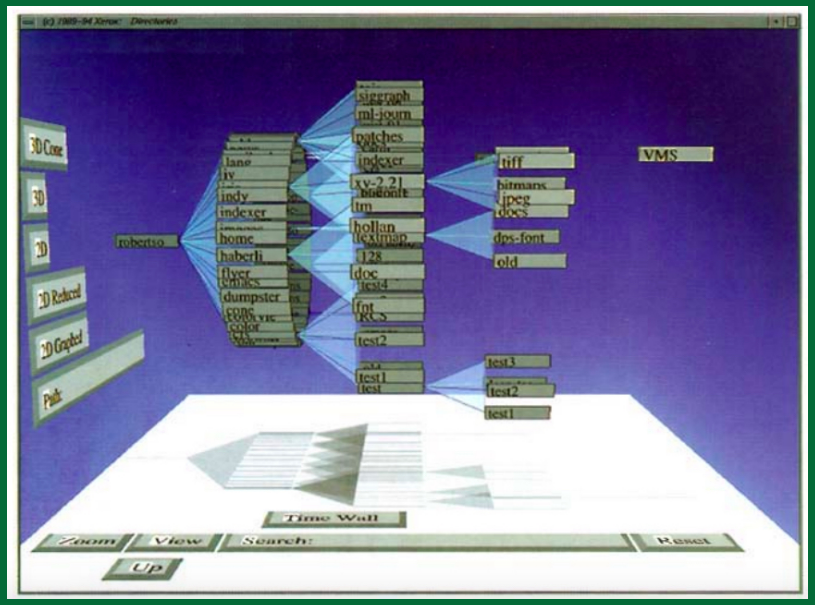
DSBA 5122: Visual Analytics

Class 8: Networks and Trees

Ryan Wesslen

October 21, 2019

Networks and Trees



Read Chapter 5: Networks and Hiearchies of Mazza

Networks (a.k.a. graphs)

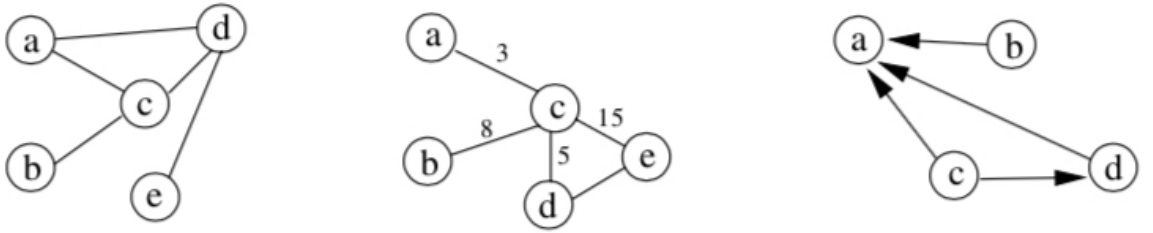
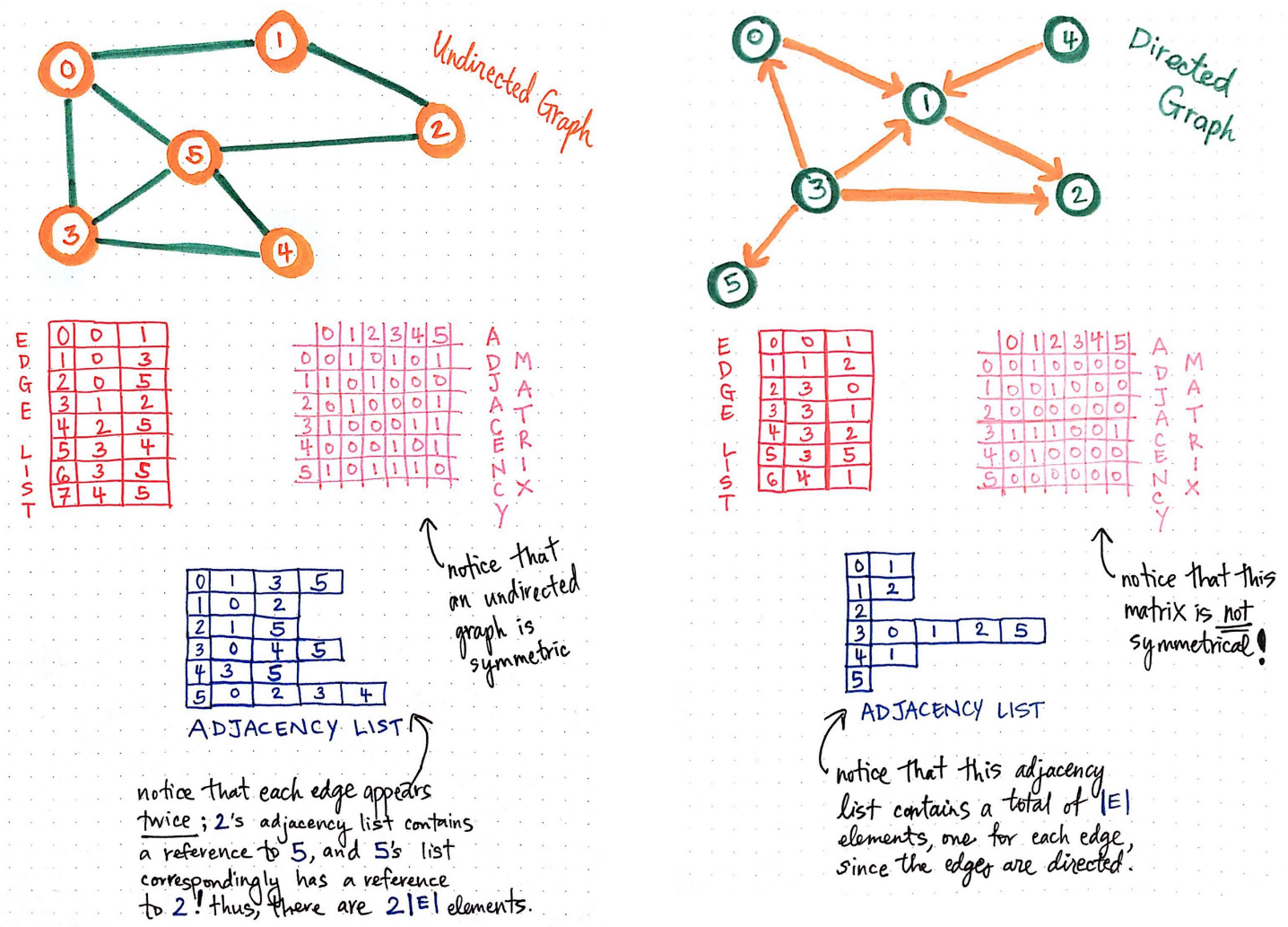


Fig. 5.1 Examples of graphs. On the left is a normal graph, in the center is a graph in which each edge is given a numerical value, and to the right is a directed graph.

- Graphs are visual representations in which the points, called nodes or vertices, represent instances of the data.
- Nodes are correlated by connections, called edges, which represent relationships between the instances.
- Possible features of a network: weights, direction, labels.



Vaidehi Joshi Medium post

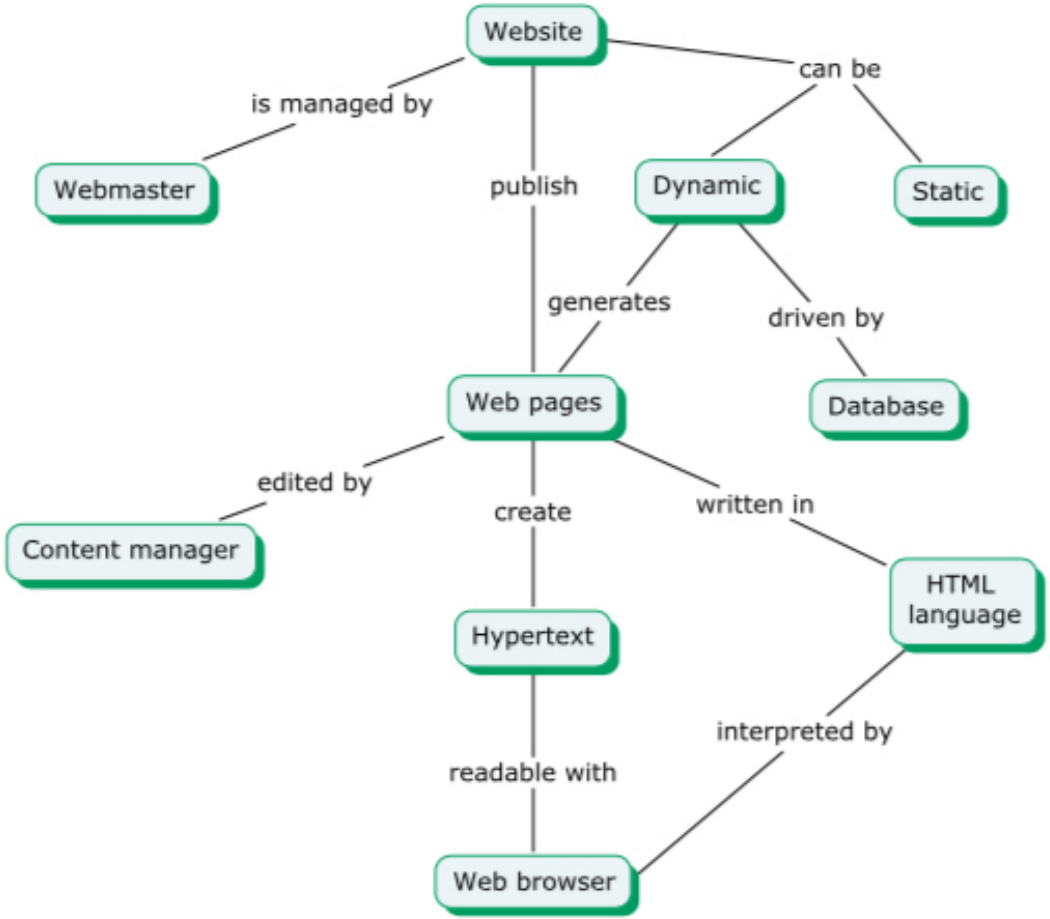


Fig. 5.2 Example of a concept map that describes concepts regarding a website.

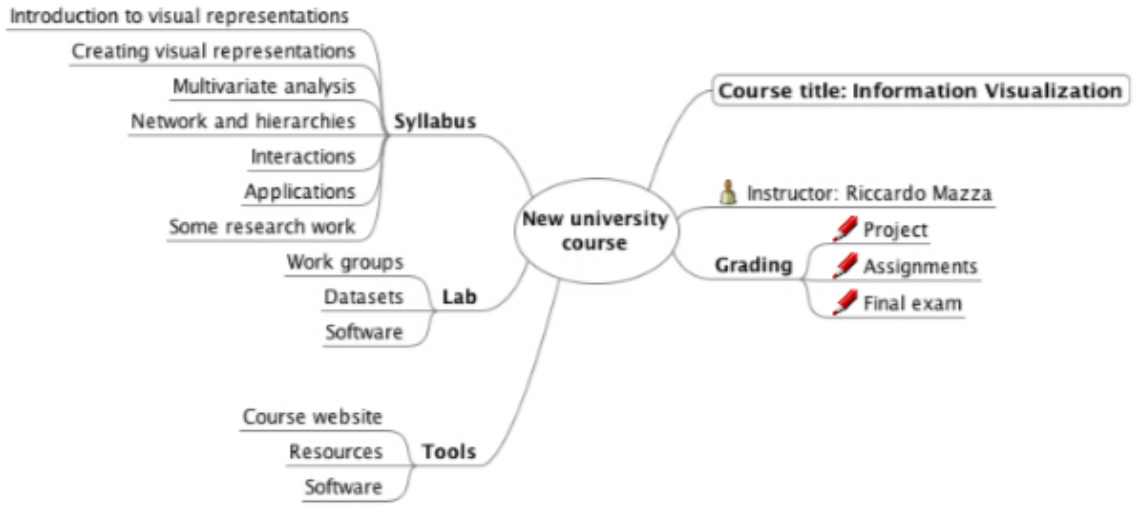


Fig. 5.3 Example of mind map for the planning of a university course.

BNOSAC: R NLP packages ecosystem

Fig. 5.4 An example of a complex graph.

Layouts

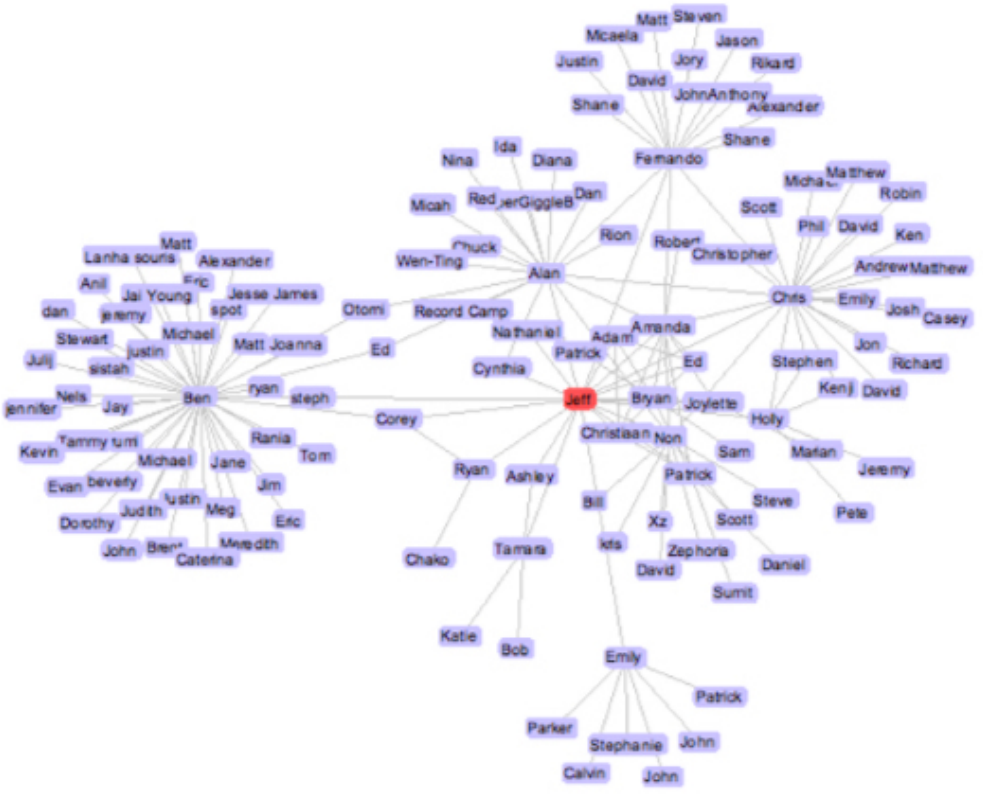


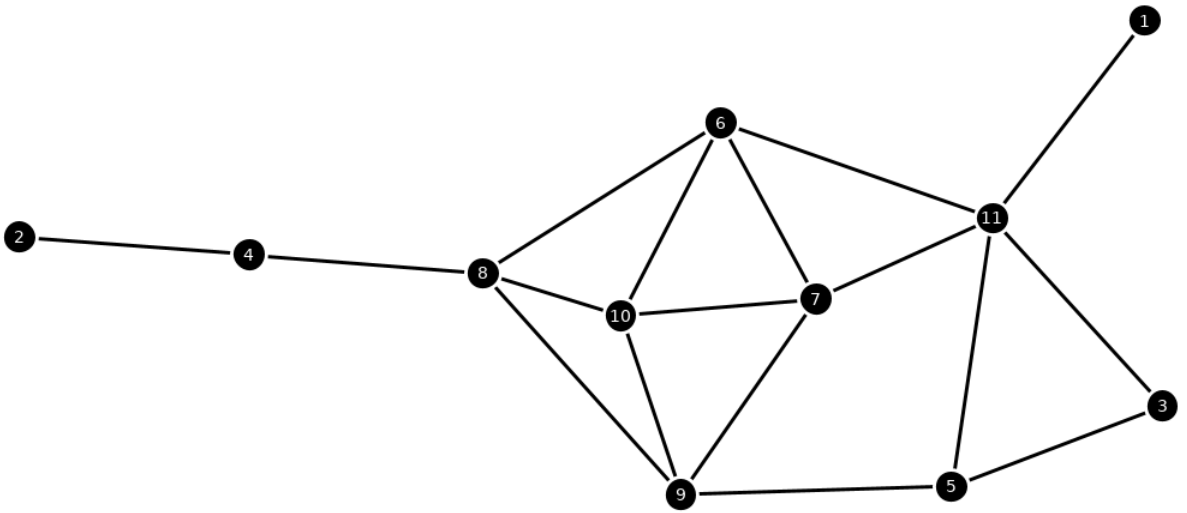
Fig. 5.5 An example of a graph that uses a force-directed algorithm to represent a social network. Image created with the prefuse tool and reproduced with the permission of Jeffrey Heer, University of California, Berkeley.

See R-Graph-Gallery Network Layouts for more.

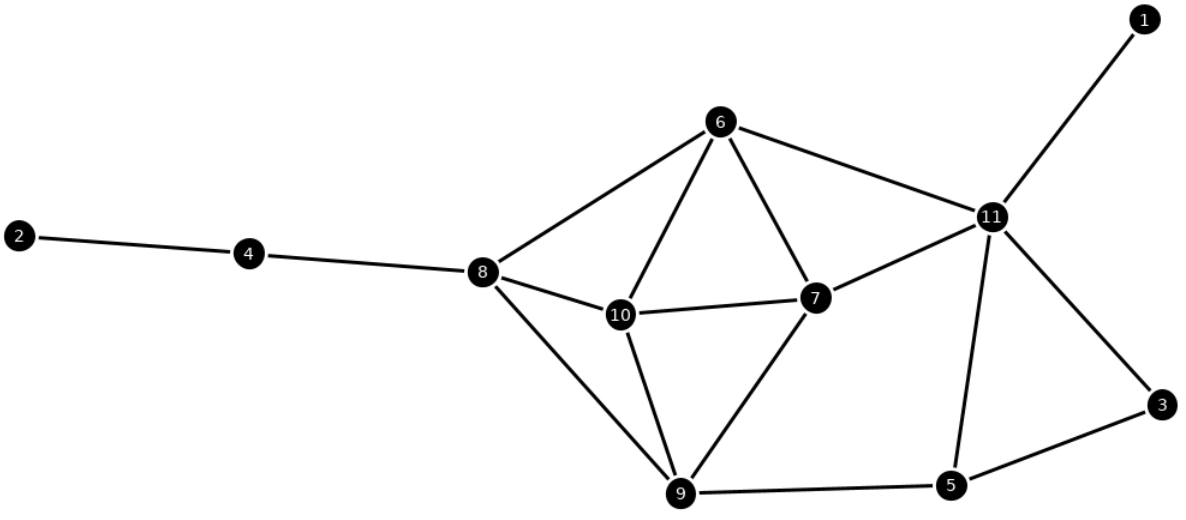
Which is the most important node?

```
library(igraph)
library(netrankr)

g <- graph.empty(n = 11,directed = FALSE)
g <- add_edges(g,c(1,11,2,4,3,5,3,11,4,8,5,9,5,11,6,7,6,8,
                   6,10,6,11,7,9,7,10,7,11,8,9,8,10,9,10))
```



Centrality Measures

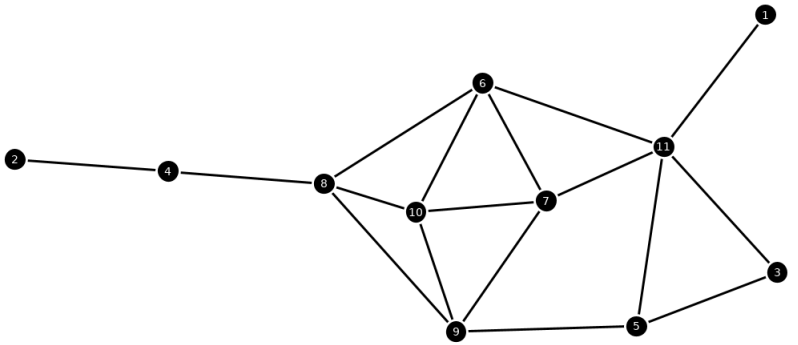



```
# from igraph
cent_scores <- tibble(
   node = 1:11,
   degree = degree(g),
   betweenness = round(betweenness(g),4),
   closeness = round(closeness(g),4),
   eigenvector = round(eigen_centrality(g)$vector,
   subgraph = round(subgraph_centrality(g),4))

cent_scores %>%
  arrange(desc(betweenness))
```

```
## # A tibble: 11 x 6
##     node degree betweenness closeness eigenvector subgraph
##    <int>  <dbl>       <dbl>     <dbl>       <dbl>    <dbl>
##  1     8      4        16.3     0.0556       0.839     6.67
##  2    11      5        14.7     0.0556       0.845     7.39
##  3     6      4         9.83    0.0588       0.985     7.81
##  4     4      2         9       0.04         0.242     2.42
##  5     9      4         7.33    0.0556       0.911     7.03
##  6     5      3         3.83    0.05         0.571     4.39
##  7     7      4         2.67    0.0526       1         7.94
##  8    10      4         1.33    0.0526       0.999     8.24
##  9     1      1         0       0.037        0.226     1.83
## 10     2      1         0       0.0294       0.0646    1.60
## 11     3      2         0       0.04         0.379     3.15
```

Centrality Measures using DT DataTable



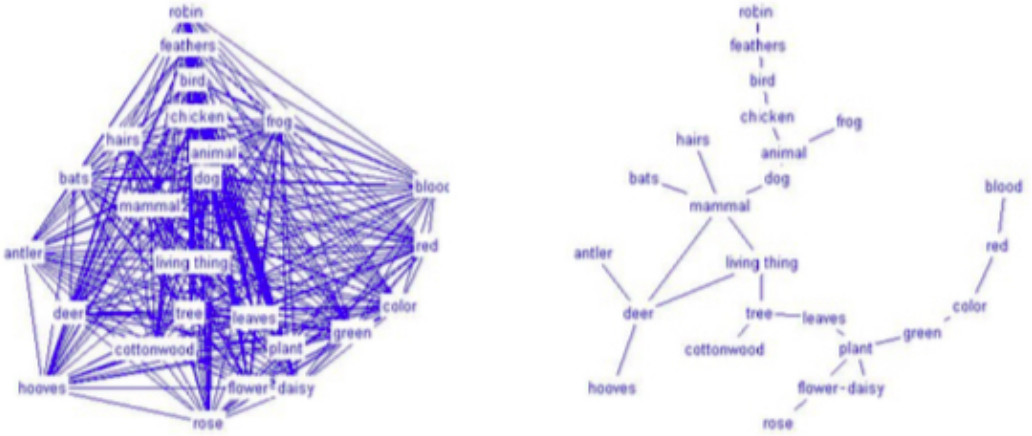


Fig. 5.6 Complete graph (left) and the reduced link version (right), achieved using the pathfinder technique. Graphs were generated with the KNOT analysis tool. Images courtesy of Interlink.

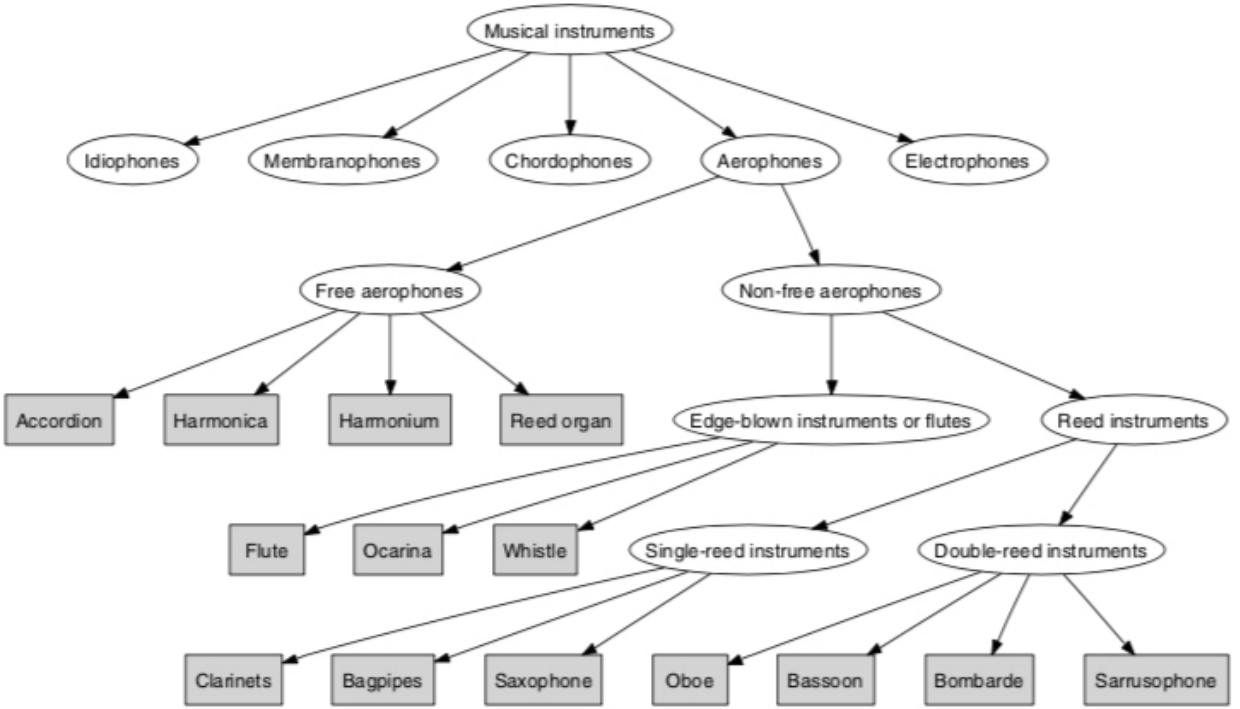


Fig. 5.13 A tree representing the classification of the wind instruments according to Curt Sachs and Erich von Hornbostel.

- A tree (hierarchy) can be represented through a graph with a starting node called root.
- Each node has zero or more child nodes its ancestor is called the parent node. A node has at most one parent.

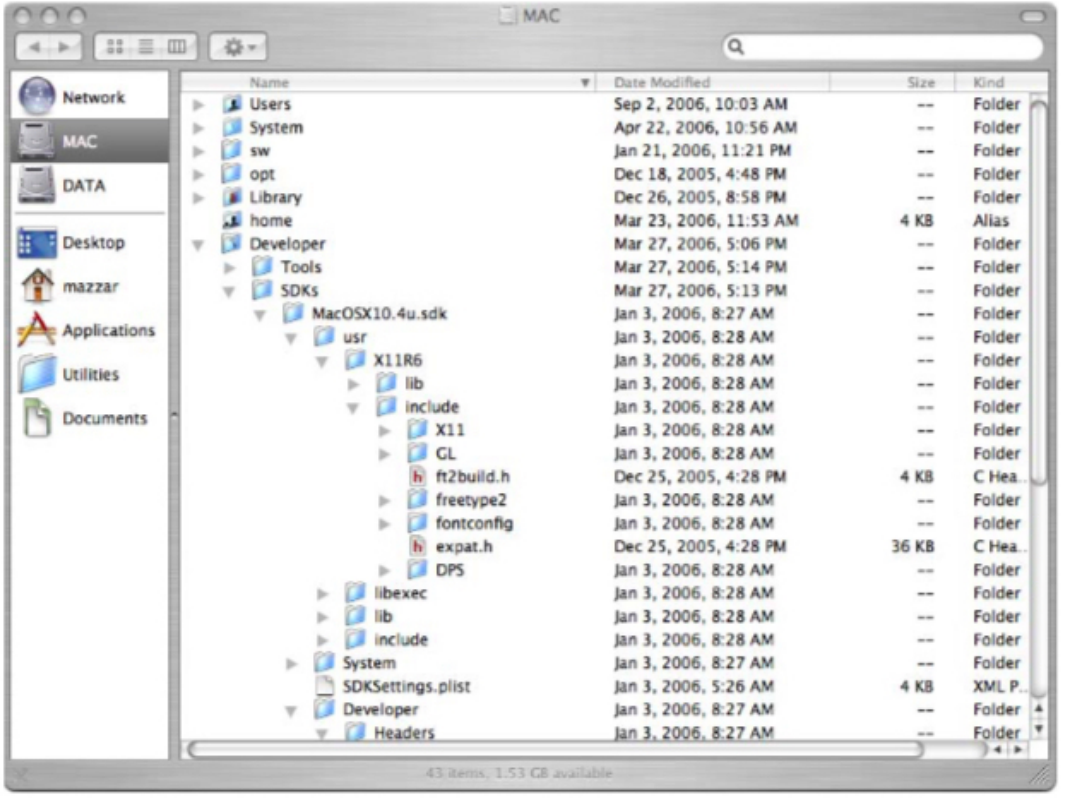


Fig. 5.14 Representation of the file system by the graphical interface Finder in the Mac OS X system.

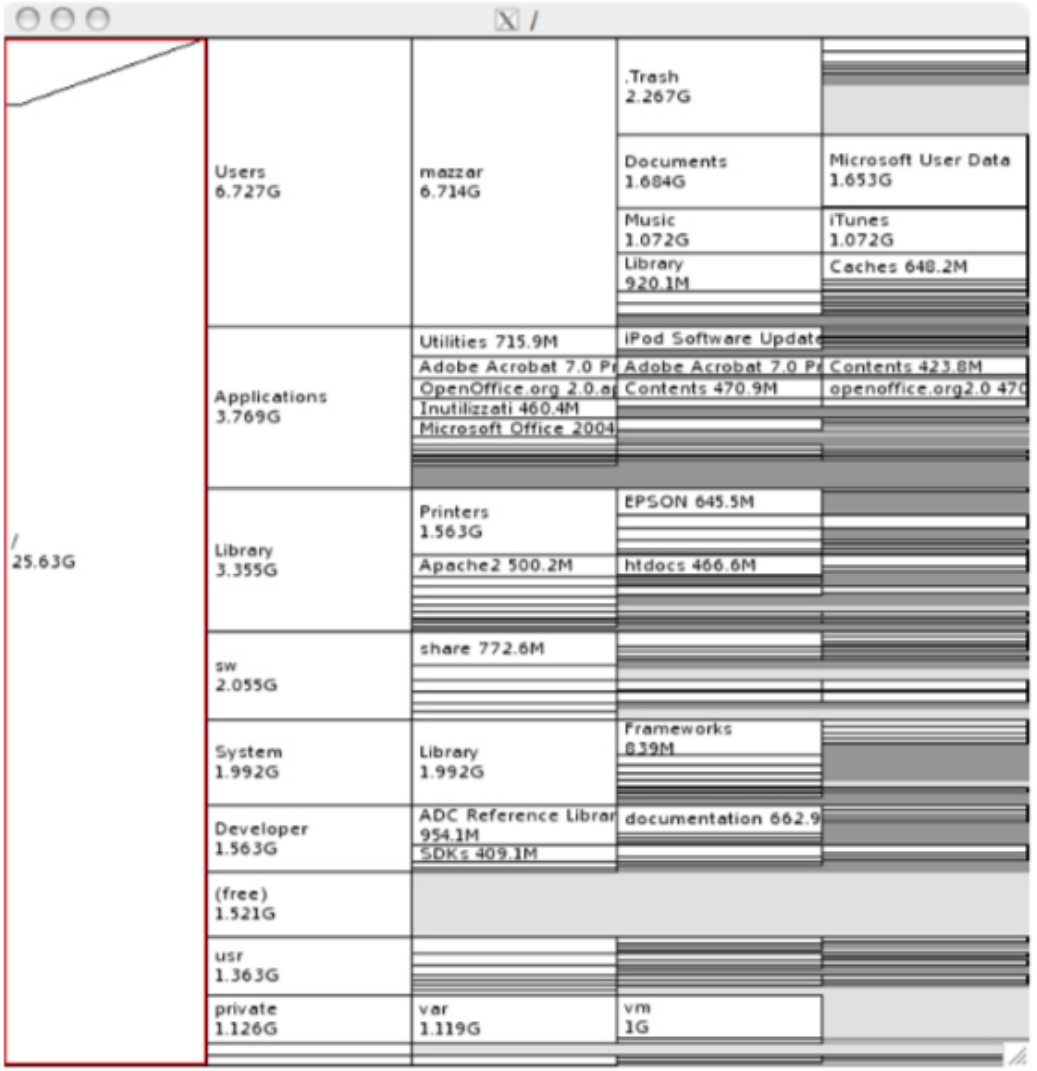


Fig. 5.15 Xdiskusage uses a representation with rectangles of the file system.

json files: Tweet Example

json files: Tweet Example

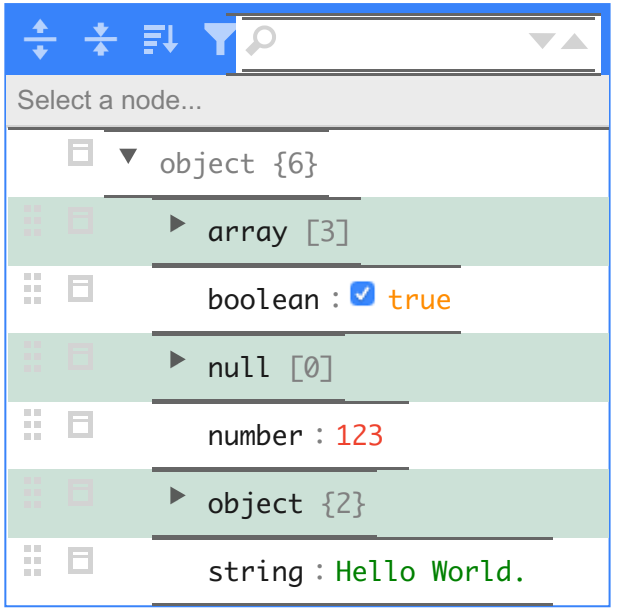

```
{
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1\/ Today we\u2019re sharing our vision for the future of the Twitter API platform!\nhttps:\/\/t.co\/XweGngmxlP",
  "user": {
    "id": 2244994945,
    "name": "Twitter Dev",
    "screen_name": "TwitterDev",
    "location": "Internet",
    "url": "https:\/\/dev.twitter.com\/",
    "description": "Your official source for Twitter Platform news, updates & events. Need technical help? Visit https:\/\/twittercommunity.com\/ \u2328\ufe0f
      #TapIntoTwitter"
  },
  "place": {
  },
  "entities": {
    "hashtags": [
    ],
    "urls": [
      {
        "url": "https:\/\/t.co\/XweGngmxlP",
        "unwound": {
          "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/3xo1c",
          "title": "Building the Future of the Twitter API Platform"
        }
      }
    ],
    "user_mentions": [
    ]
  }
}
```


- See Twitter Developer JSON intro for more details.
- See 'jsonlite' package for handling json files in R.

listviewer: viewing json in R

```
library(listviewer)

jsonedit(
  list(
    array = c(1,2,3)
    ,boolean = TRUE
    ,null = NULL
    ,number = 123
    ,object = list( a="b", c="d" )
    ,string = "Hello World. "
  )
)
```



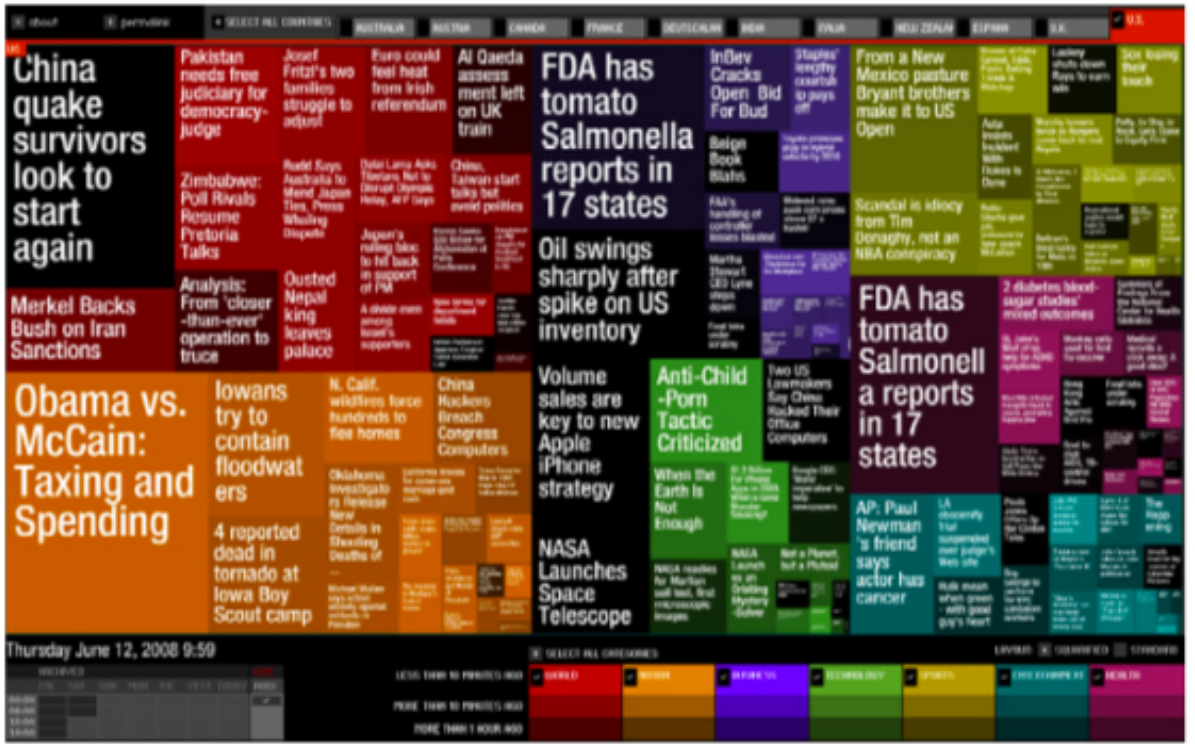


Fig. 5.22 Newsmap uses a treemap algorithm to represent news from Google News U.S. on June 12, 2008. Image reproduced with the permission of Marcos Weskamp.

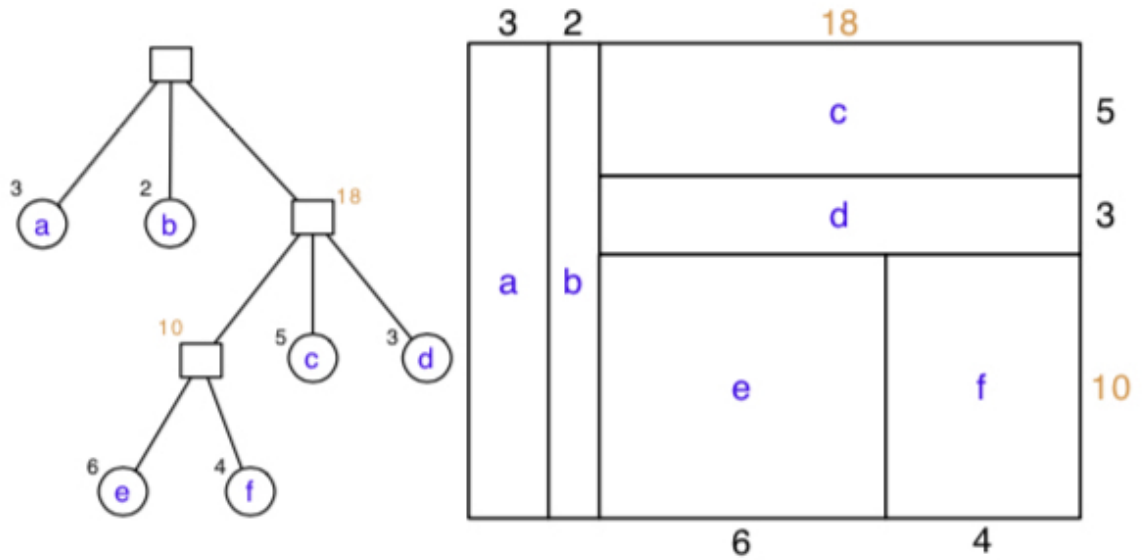
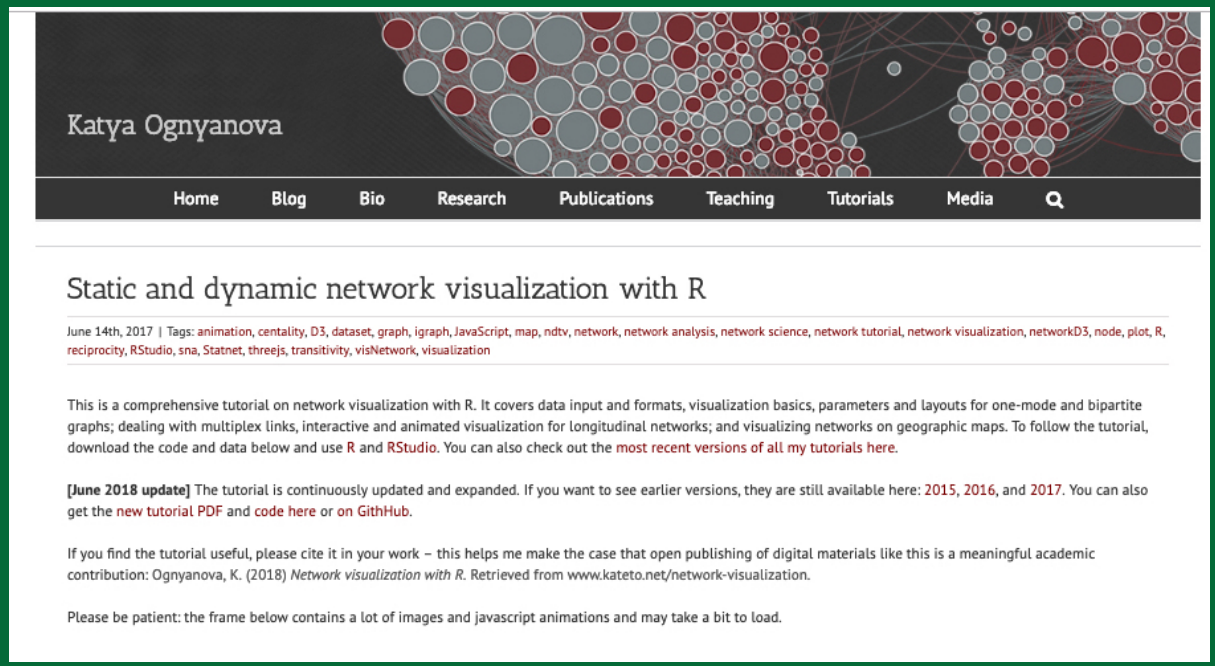


Fig. 5.20 On the left is an example of a tree representation, where a numerical value is associated with each leaf. The internal nodes report the sum of the values of the nodes below. On the right is the treemap representation of the tree.

Katya Ognyanova's "Network visualization with R."



Katya Ognyanova

Home Blog Bio Research Publications Teaching Tutorials Media

Static and dynamic network visualization with R

June 14th, 2017 | Tags: animation, centality, D3, dataset, graph, igraph, JavaScript, map, ndtv, network, network analysis, network science, network tutorial, network visualization, networkD3, node, plot, R, reciprocity, RStudio, sna, Statnet, threejs, transitivity, visNetwork, visualization

This is a comprehensive tutorial on network visualization with R. It covers data input and formats, visualization basics, parameters and layouts for one-mode and bipartite graphs; dealing with multiplex links, interactive and animated visualization for longitudinal networks; and visualizing networks on geographic maps. To follow the tutorial, download the code and data below and use R and RStudio. You can also check out the most recent versions of all my tutorials here.

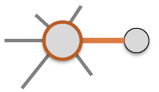
[June 2018 update] The tutorial is continuously updated and expanded. If you want to see earlier versions, they are still available here: 2015, 2016, and 2017. You can also get the new tutorial PDF and code here or on GithHub.

If you find the tutorial useful, please cite it in your work – this helps me make the case that open publishing of digital materials like this is a meaningful academic contribution: Ognyanova, K. (2018) *Network visualization with R*. Retrieved from www.kateto.net/network-visualization.

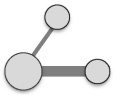
Please be patient: the frame below contains a lot of images and javascript animations and may take a bit to load.

Network visualization goals

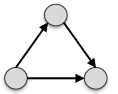
Key actors and links



Relationship strength

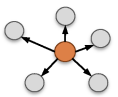


Structural properties



Communities

Diffusion patterns

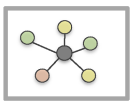


Network evolution

Networks as maps

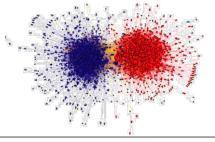
Networks as persuasion

Networks as art



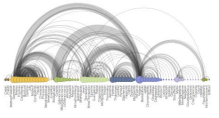
Some network visualization types

Network Maps

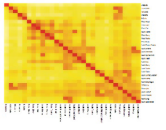


Statistical charts

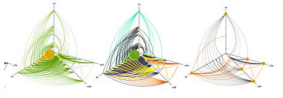
Arc diagrams



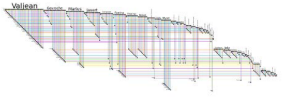
Heat maps



Hive plots



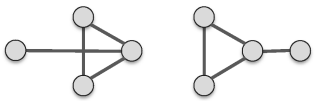
Biofabric



Layout aesthetics

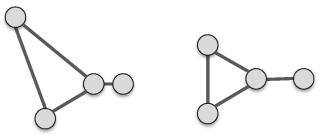
Minimize edge crossing

No Yes



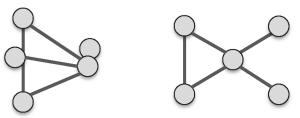
Uniform edge length

No Yes



Prevent overlap

No Yes



Symmetry

No Yes