

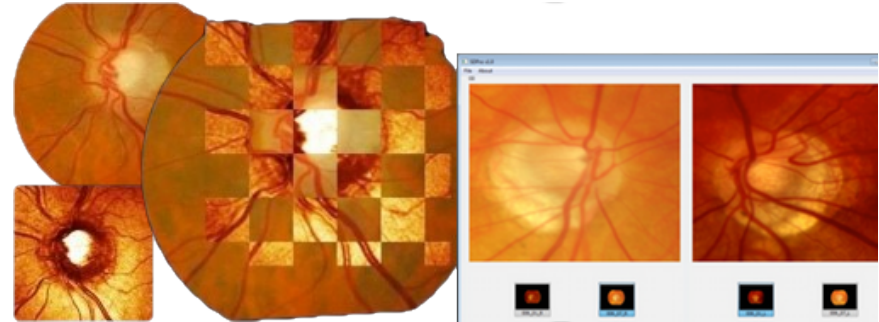
Visualization

What is it good for?

Dr. Phil Legg
4th March 2015
SERG Seminar, UWE

A (very) quick background...

- PhD: Multi-modal image registration (Cardiff) – *how can a computer automatically align images of different appearance?*
- PDRA: Sports Video Visualization (Swansea) – *how can data visualization improve player performance and coaching capabilities?*
- PDRA: Insider Threat Detection (Oxford) – *how can machine learning and visual analytics enhance capabilities of recognising malicious activity?*



Data Visualization

Data Processing

Pattern Recognition

Machine Learning

Visual Analytics

Analytical Reasoning

Statistical Modeling

Image / Video Processing

Computer Vision

Human-Computer Interaction

Perception and Cognition

Design Studies

Data Visualization

What *actually* is it for?

To *convey data* accurately.

To *provide insight* to the user.

To *reduce cognitive load* for the user.

Ultimately, to *save time* for the user

Data Visualization

What *actually* is it for?

Definition: Visualization (or more precisely, computer-supported data visualization) is a study of transformation from data to visual representations in order to facilitate effective and efficient cognitive processes in performing tasks involving data. The fundamental measure for effectiveness is correctness and that for efficiency is the time required or accomplishing a task.

(Chen et al. “What is Visualization Really for?”)

Why visualize data?

- Suppose we have four different datasets that we want to learn about.
- What can we say about the four datasets from initial observations?
- What similarities do the datasets share, if any?

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

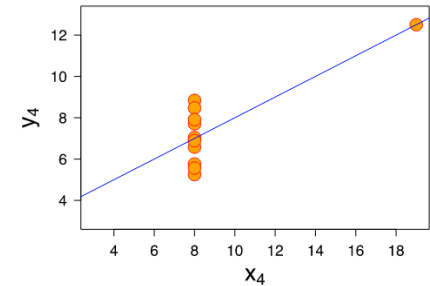
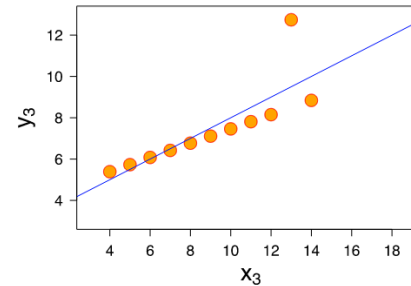
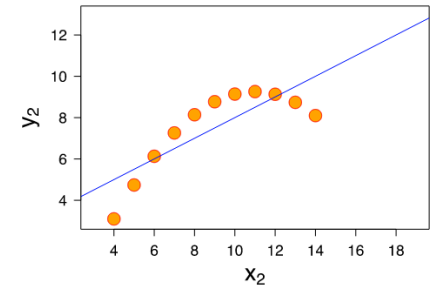
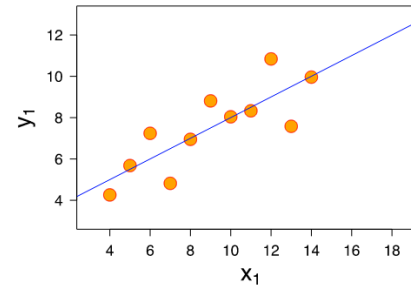
Why visualize data?

- Statistics tell us that the datasets actually share **a lot** of similarities!!
- Average(x) = 9
- Average(y) = 7.50
- Variance(x) = 11
- Variance(y) = 4.12
- Correlation(x,y) = 0.816
- Linear regression: $y = 0.5x + 3$
- So are they similar?

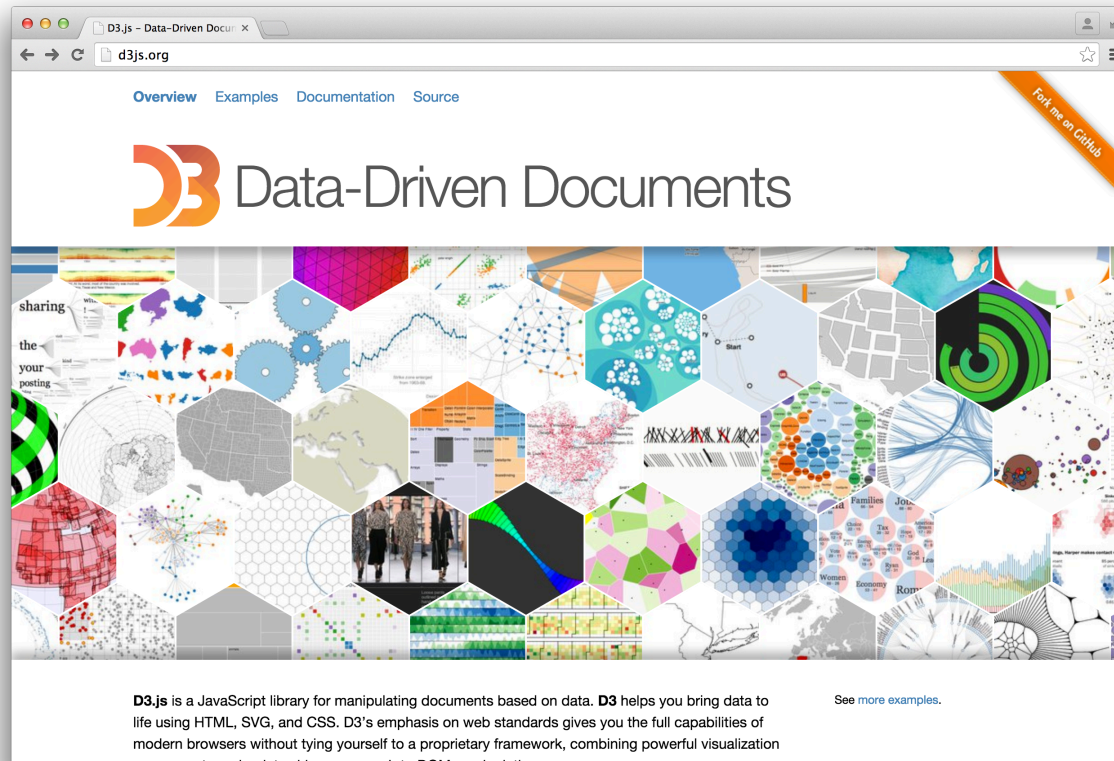
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Why visualize data?

- Visualizing the data tells a very different story – this is known as **Anscombe's Quartet**.
- Statistics are merely one string to the bow that is the data analysis process.
- Visualization allows the analyst to choose the right tools for the data.



D3 – Examples of DataVis

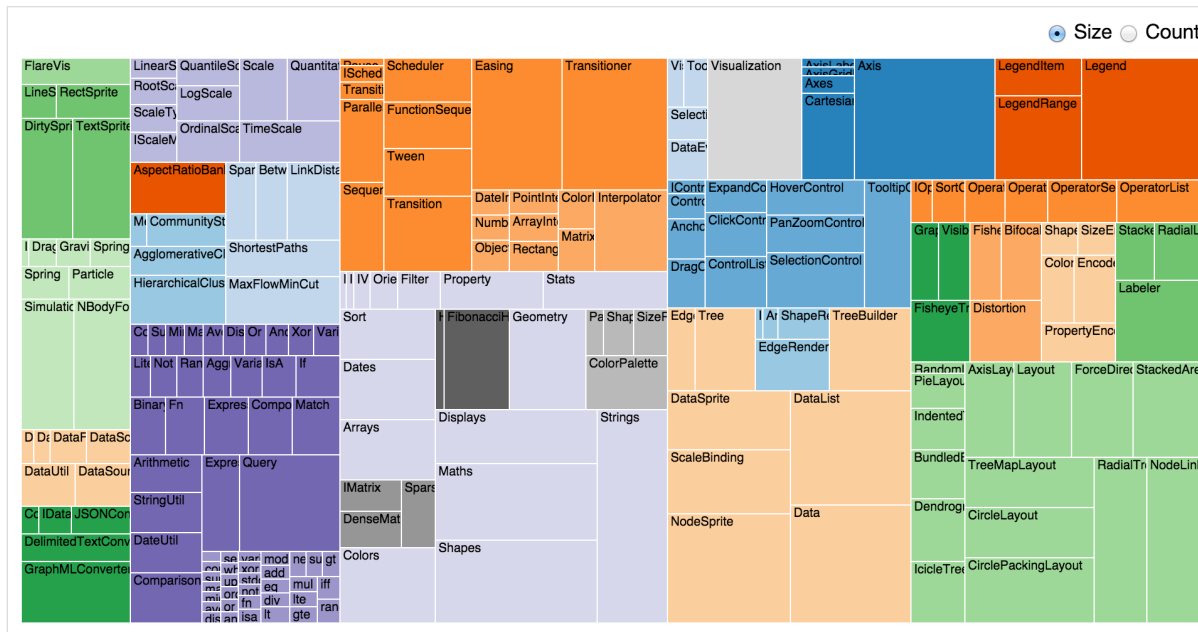


D3.js - Just one of the many visualization libraries that now exist!

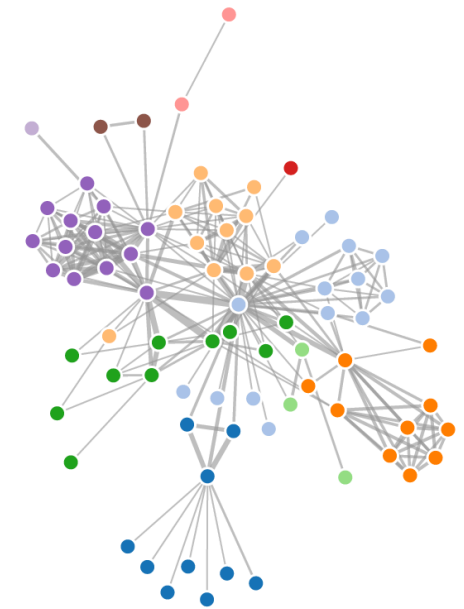
Web-based JavaScript libraries becoming popular due to easier dissemination and sharing of tools.

<http://d3js.org/>

D3 – Examples of DataVis



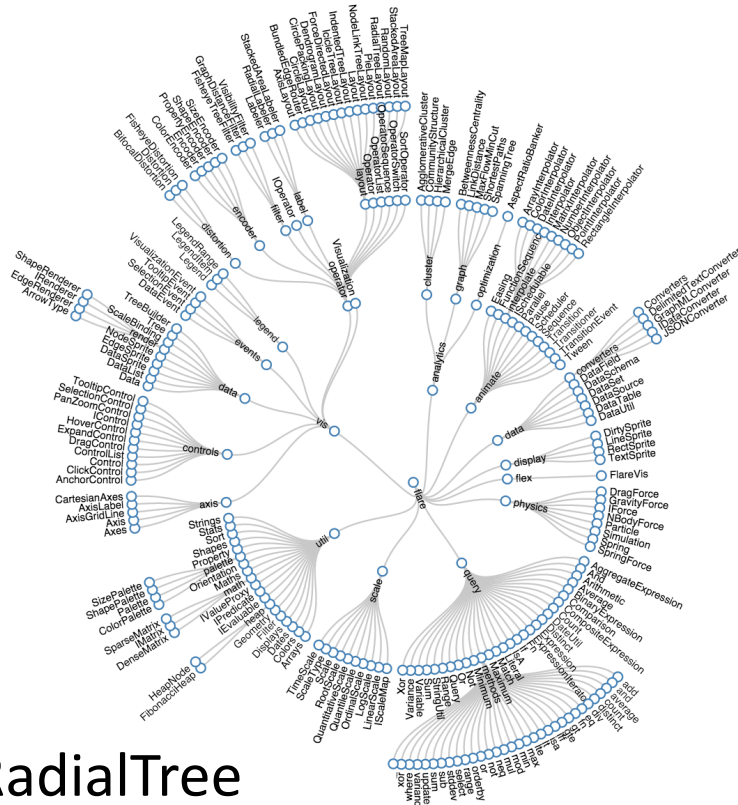
TreeMaps



Node-Link diagrams

<http://d3js.org/>

D3 – Examples of DataVis



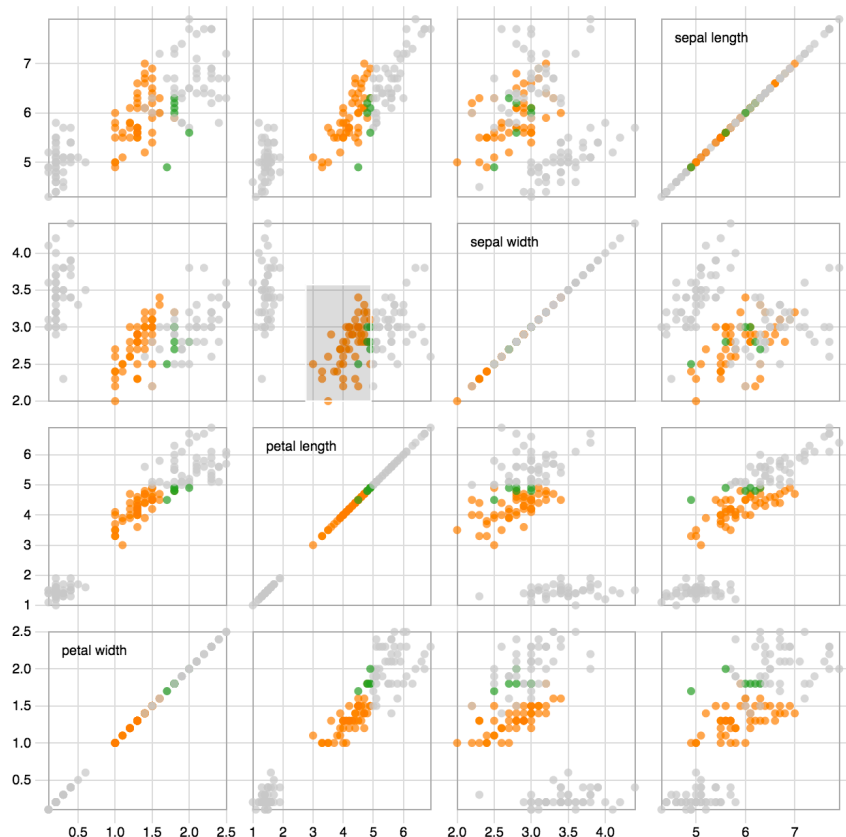
RadialTree



Bubble Chart

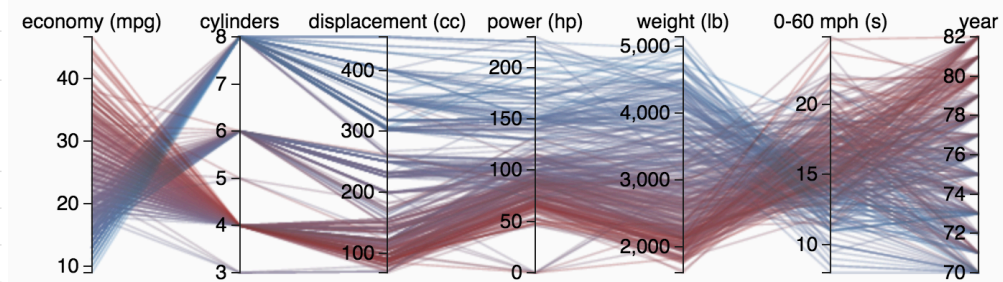
<http://d3js.org/>

D3 – Examples of DataVis

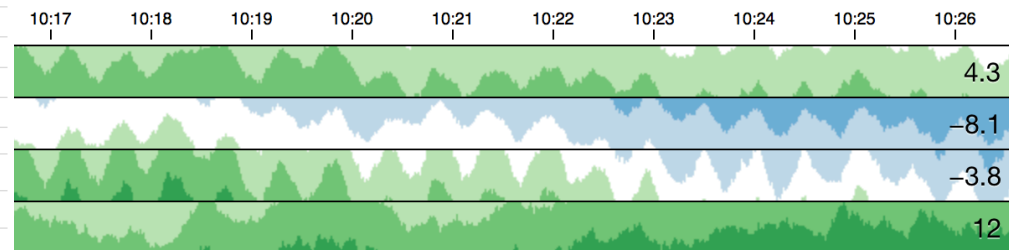


Linked Scatter Plots

<http://d3js.org/>



Parallel Co-ordinates



Horizon Charts

Visualizing complex systems

How can we visualize large complex systems?

Why?

- Why do we want to visualize the system?
What is the task objective?
Discovery? Maintenance?

Who for?

- Who do we want to visualize the system for?
Expert user? Non-technical user?

Visualizing complex systems

How can we visualize large complex systems?

What?

- What is the message that we wish to communicate?
What are the key data attributes required to achieve this?
Do we even know yet – is this **why** we want to visualize?

How?

- How should we map from the data to a visual representation?
How will the end user engage with the visualization?

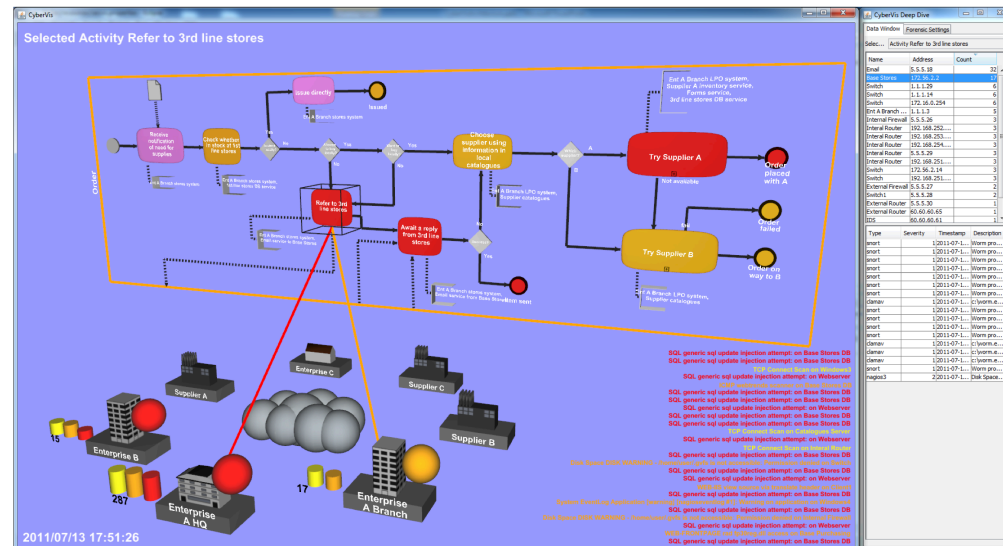
Visualization of large complex systems

Why? To improve situational awareness of enterprise network attacks and the consequences on business operations.

Who for? Security Operations Center (SOC) analysts.

What? Network topology, Business Process Modeling Notation, IDS alerts – when are these at “high alert”.

How? Interactive 3D system.



CyberVis (Oxford)

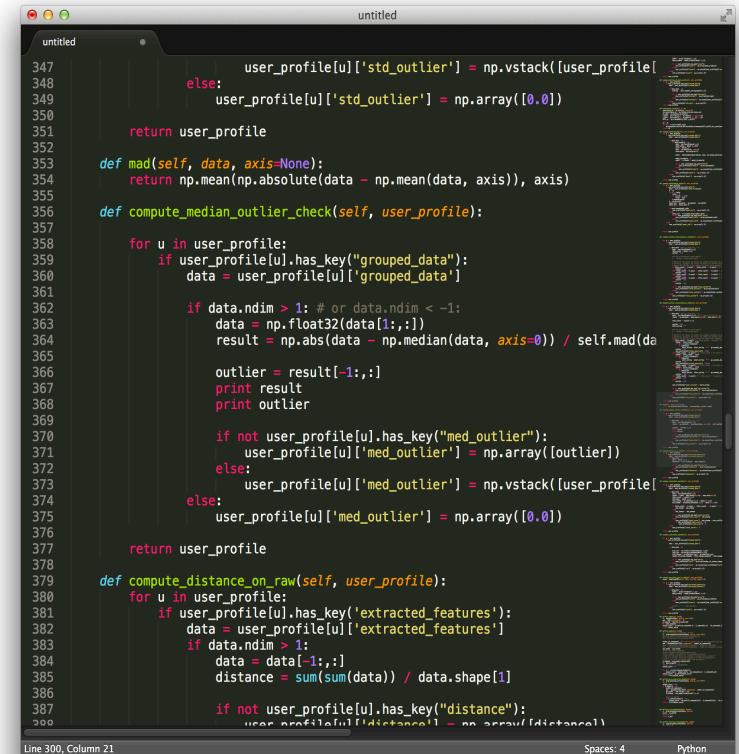
Visualization of software systems

Why? To understand code structure? To hand over code project? To assist debug?

Who for? Self? Colleague? Management? Customer?

What? LOC count, functional task, dependencies, memory leakage, execution time, static/dynamic analysis, vulnerabilities/exploits...

How? ...

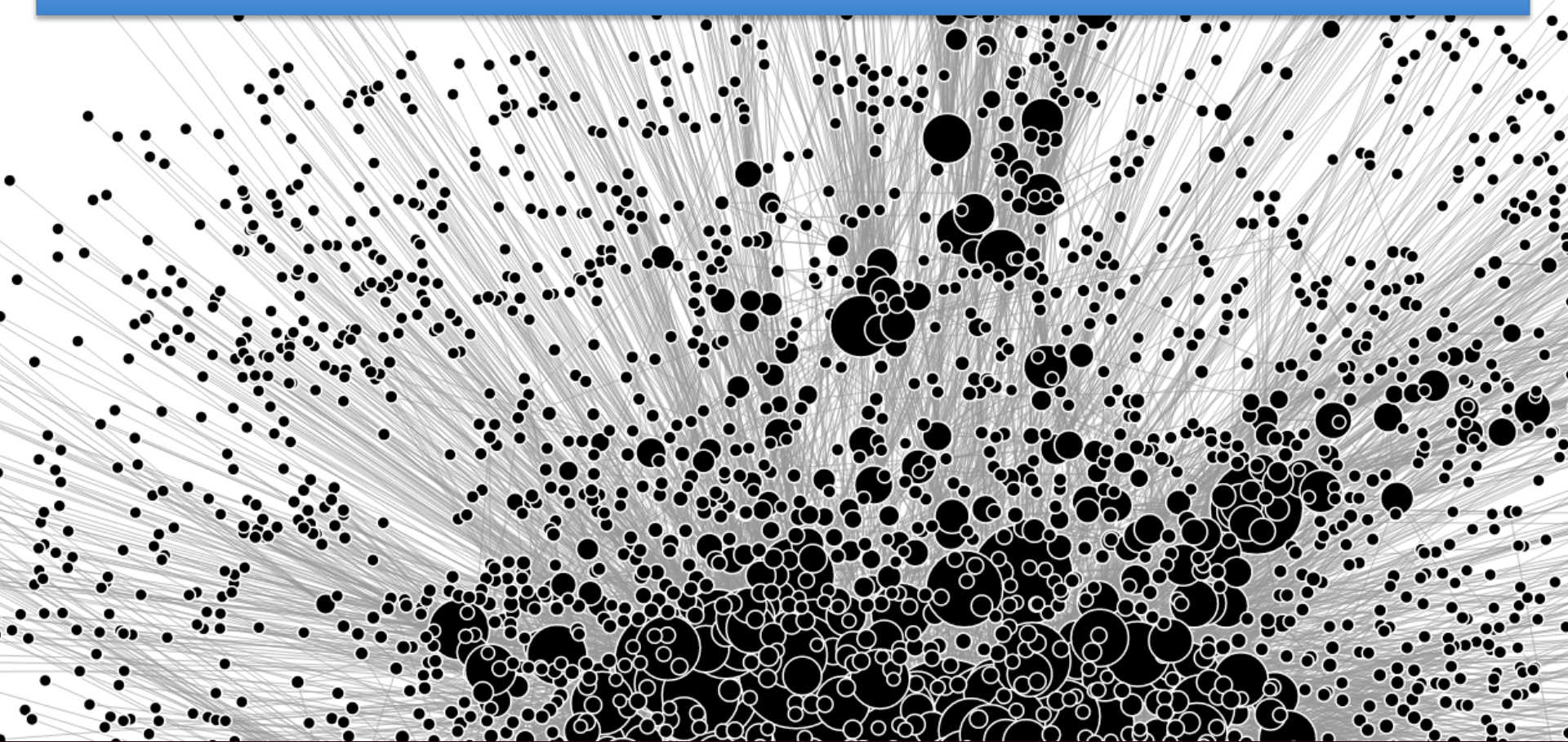


```

347         user_profile[u]['std_outlier'] = np.vstack([user_profile[
348             else:
349                 user_profile[u]['std_outlier'] = np.array([0.0])
350
351     return user_profile
352
353 def mad(self, data, axis=None):
354     return np.mean(np.absolute(data - np.mean(data, axis)), axis)
355
356 def compute_median_outlier_check(self, user_profile):
357
358     for u in user_profile:
359         if user_profile[u].has_key("grouped_data"):
360             data = user_profile[u]['grouped_data']
361
362             if data.ndim > 1: # or data.ndim < -1:
363                 data = np.float32(data[1:,:])
364                 result = np.abs(data - np.median(data, axis=0)) / self.mad(da
365
366             outlier = result[-1,:]
367             print result
368             print outlier
369
370             if not user_profile[u].has_key("med_outlier"):
371                 user_profile[u]['med_outlier'] = np.array([outlier])
372             else:
373                 user_profile[u]['med_outlier'] = np.vstack([user_profile[
374             else:
375                 user_profile[u]['med_outlier'] = np.array([0.0])
376
377     return user_profile
378
379 def compute_distance_on_raw(self, user_profile):
380     for u in user_profile:
381         if user_profile[u].has_key('extracted_features'):
382             data = user_profile[u]['extracted_features']
383             if data.ndim > 1:
384                 data = data[-1,:]
385                 distance = sum(sum(data)) / data.shape[1]
386
387             if not user_profile[u].has_key("distance"):
388                 user_profile[u]['distance'] = np.array([distance])
389
390
  
```

[Sensalire, Ogao and Telea] Evaluation of Software Visualization Tools: Lessons Learned, IEEE, 2009

How should we address **scalability**?!
Information overload still exists in visualization



Visualization of software systems

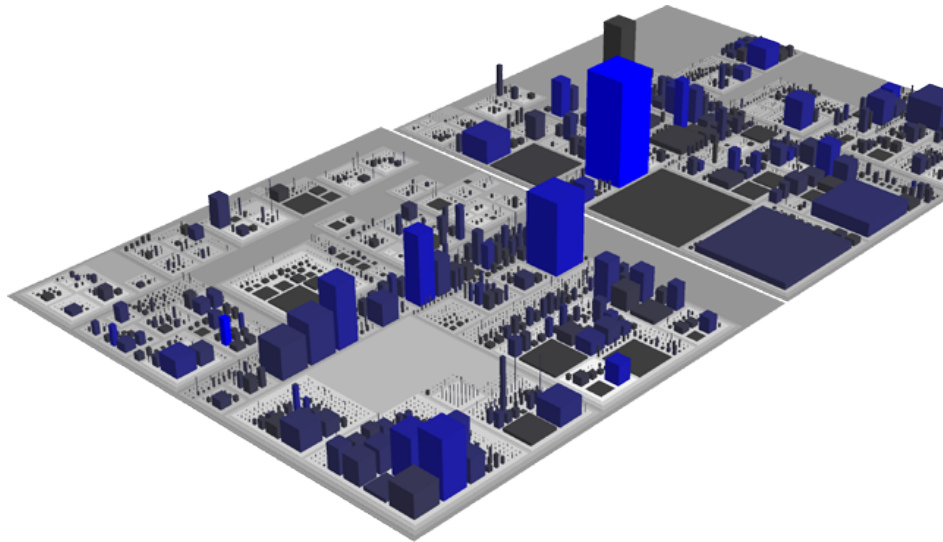
- What are the “best” approaches to visualise software systems?
- What data and information are required to visualize such systems?
- Task-dependent!
 - No “one size fits all”.
- Many tools have been proposed – but how well do **you** feel they perform?
- What do **you** wish visualization could do to improve software engineering for you?

```

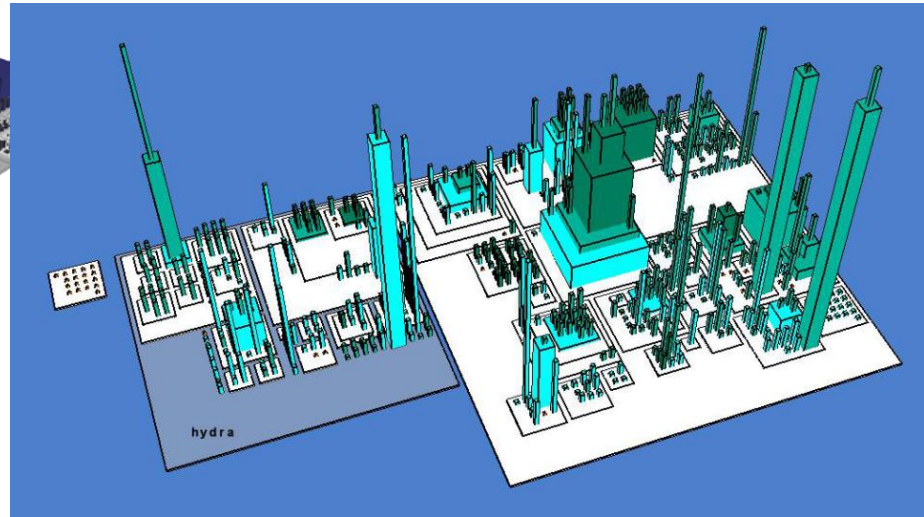
347         user_profile[u]['std_outlier'] = np.vstack([user_profile[
348             else:
349                 user_profile[u]['std_outlier'] = np.array([0.0])
350
351     return user_profile
352
353 def mad(self, data, axis=None):
354     return np.mean(np.absolute(data - np.mean(data, axis)), axis)
355
356 def compute_median_outlier_check(self, user_profile):
357
358     for u in user_profile:
359         if user_profile[u].has_key("grouped_data"):
360             data = user_profile[u]['grouped_data']
361
362             if data.ndim > 1: # or data.ndim < -1:
363                 data = np.float32(data[1:,:])
364                 result = np.abs(data - np.median(data, axis=0)) / self.mad(da
365
366             outlier = result[-1,:]
367             print result
368             print outlier
369
370             if not user_profile[u].has_key("med_outlier"):
371                 user_profile[u]['med_outlier'] = np.array([outlier])
372             else:
373                 user_profile[u]['med_outlier'] = np.vstack([user_profile[
374             else:
375                 user_profile[u]['med_outlier'] = np.array([0.0])
376
377     return user_profile
378
379 def compute_distance_on_raw(self, user_profile):
380     for u in user_profile:
381         if user_profile[u].has_key('extracted_features'):
382             data = user_profile[u]['extracted_features']
383             if data.ndim > 1:
384                 data = data[-1,:]
385                 distance = sum(sum(data)) / data.shape[1]
386
387             if not user_profile[u].has_key("distance"):
388                 user_profile[u]['distance'] = np.array([distance])
389

```

Examples of Software Visualization



CodeCity

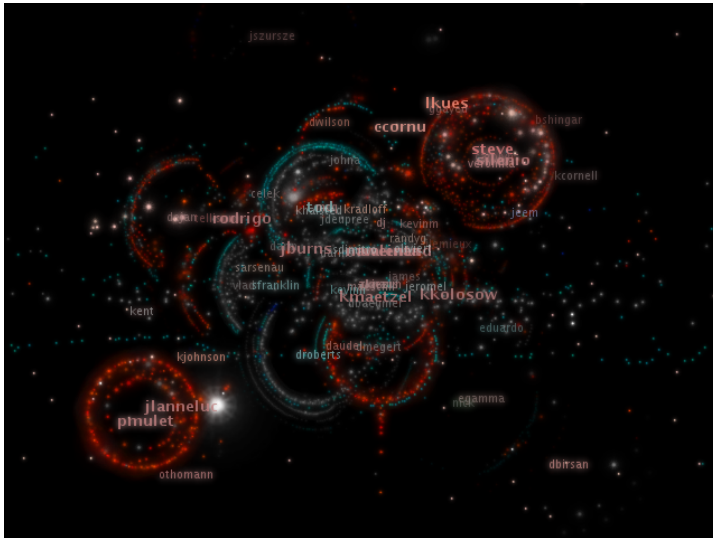


Codstruction

Source Code Visualization

TreeMap: Hierarchy indicates functions within a class. Colour / size to indicate LOC, or number of calls. (Does 3D help, or simply occlude?)

Examples of Software Visualization



Code Swarm
(User-centric)



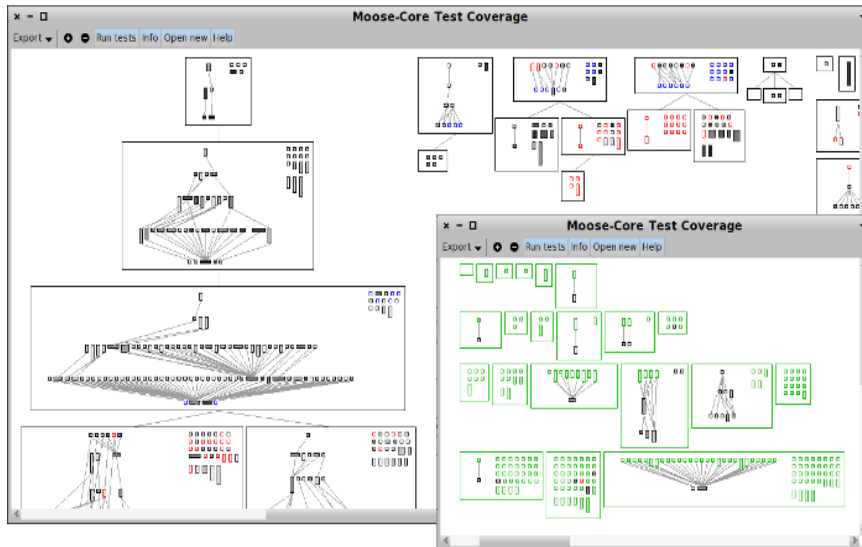
Gource
(Source-centric)

Repository visualization

Left: Clusters show users and most recent file commits. Colours show file type.

Right: Node-link shows file directory structure. (Animation – good or bad?).

Examples of Software Visualization



Hapao – Tree hierarchy of source code

- Typical examples study source code – treemap and bubble charts
concept shows classes and number of functions / lines of code.
- Hierarchy shows relationship between class files.
- Is there more than can be done to support software engineering using visualization?
- {Sequence, Dynamic, Vulnerability} Analysis?

University of the West of England



<https://softvis.wordpress.com/tools/>

Visualization – *What is it good for?*

- To convey data accurately, provide greater insight, reduce cognitive load, and to save time.
- Consider the ***Why, Who, What*** and ***How*** of creating Visualizations?
- How can vis creators address scalability issues to avoid Information Overload?
- SoftVis is an upcoming research area – but certainly not a solved problem yet!

Dr. Phil Legg
Spring 2015