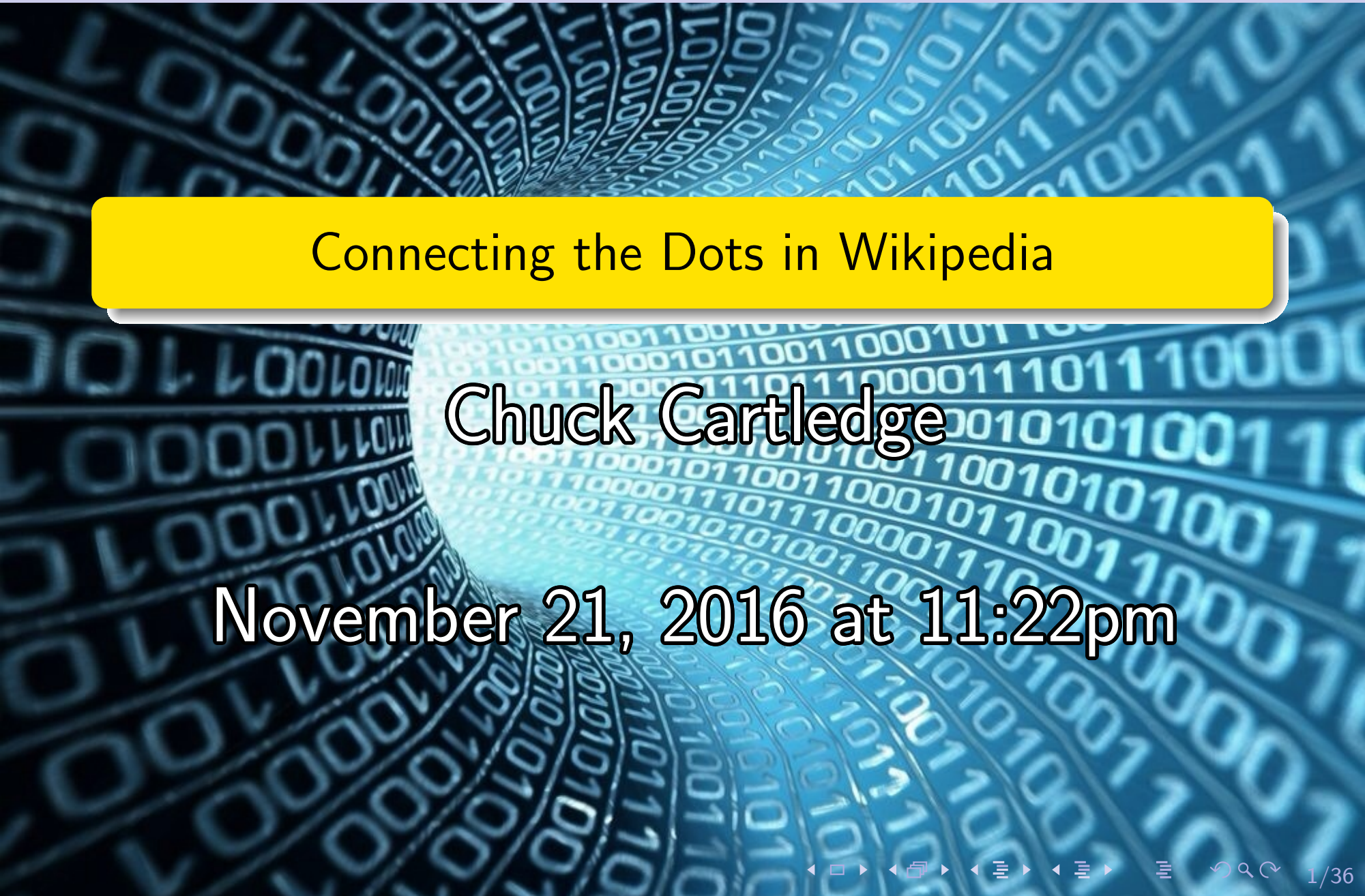


Connecting the Dots in Wikipedia

Chuck Cartledge

November 21, 2016 at 11:22pm

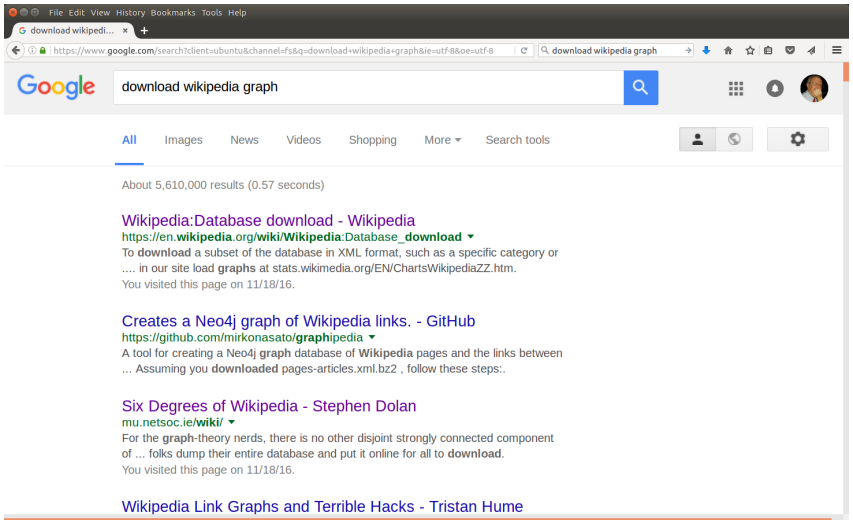
Table of contents I

Where we started:

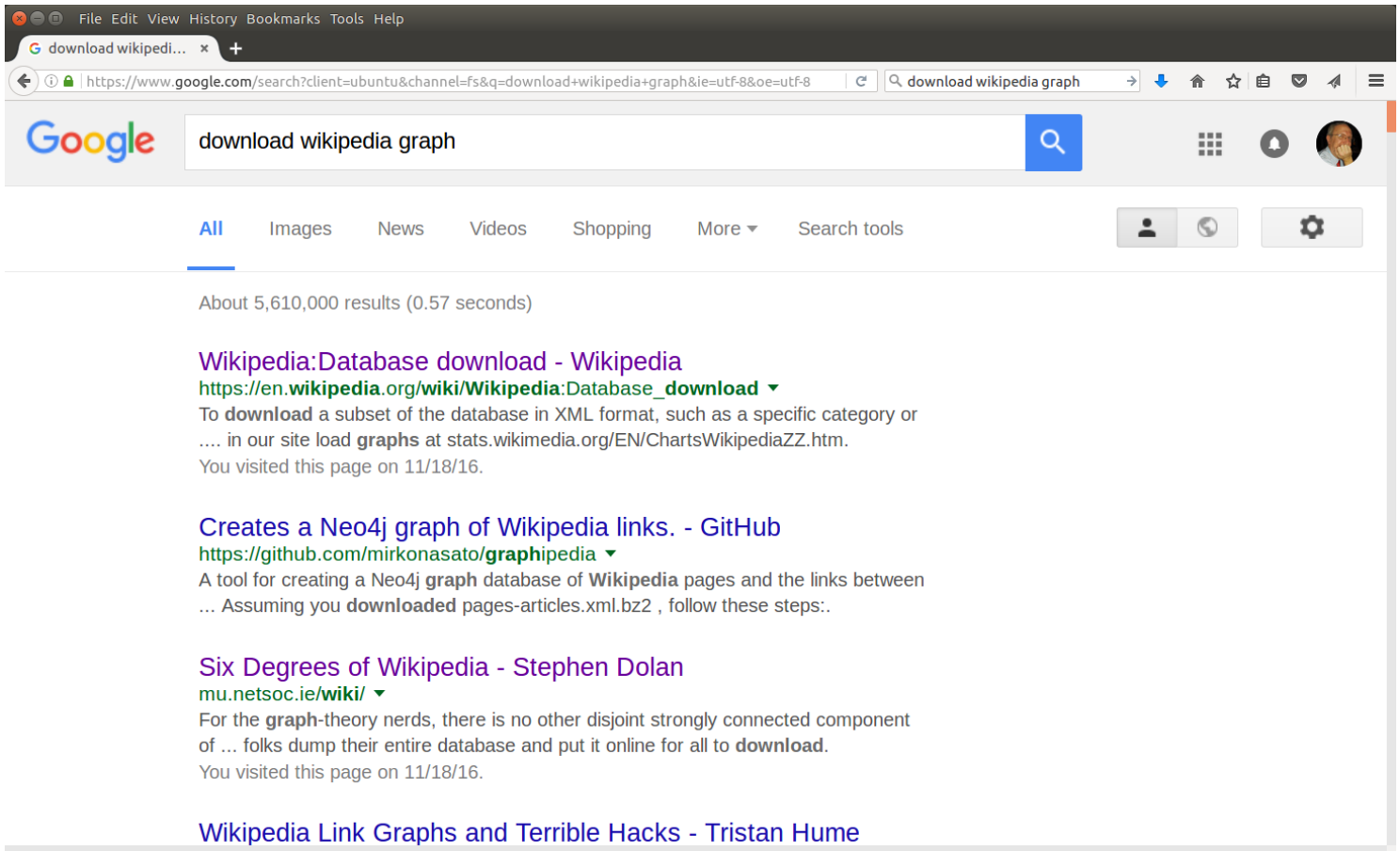
Dirk Gently's code is that everything is connected. This the basis of his holistic detective agency, and is his life's guiding principle. We will take Dirk's principal idea and apply it to Wikipedia. We'll explore a portion of Wikipedia and see how many pages/links separate a specific page from another (in graph theory this is called the shortest path, Stanley Milgram made this idea popular with his "Small-World" paper, and many people have played the "Six Degrees of Kevin Bacon" game). We'll also look at the average number of links between pages.

This is a common interest

- Wikipedia provides a downloadable XML file that represents their graph.
- Lots of people have looked at their data in lots of ways.

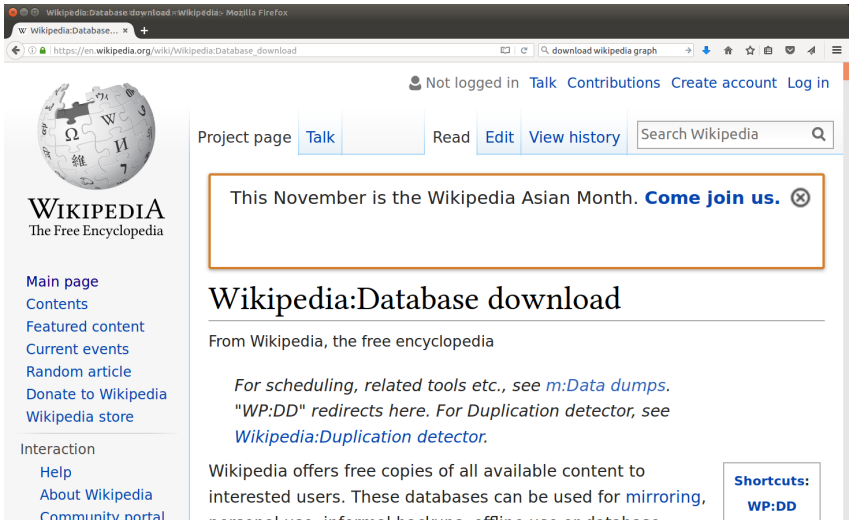


Same image.

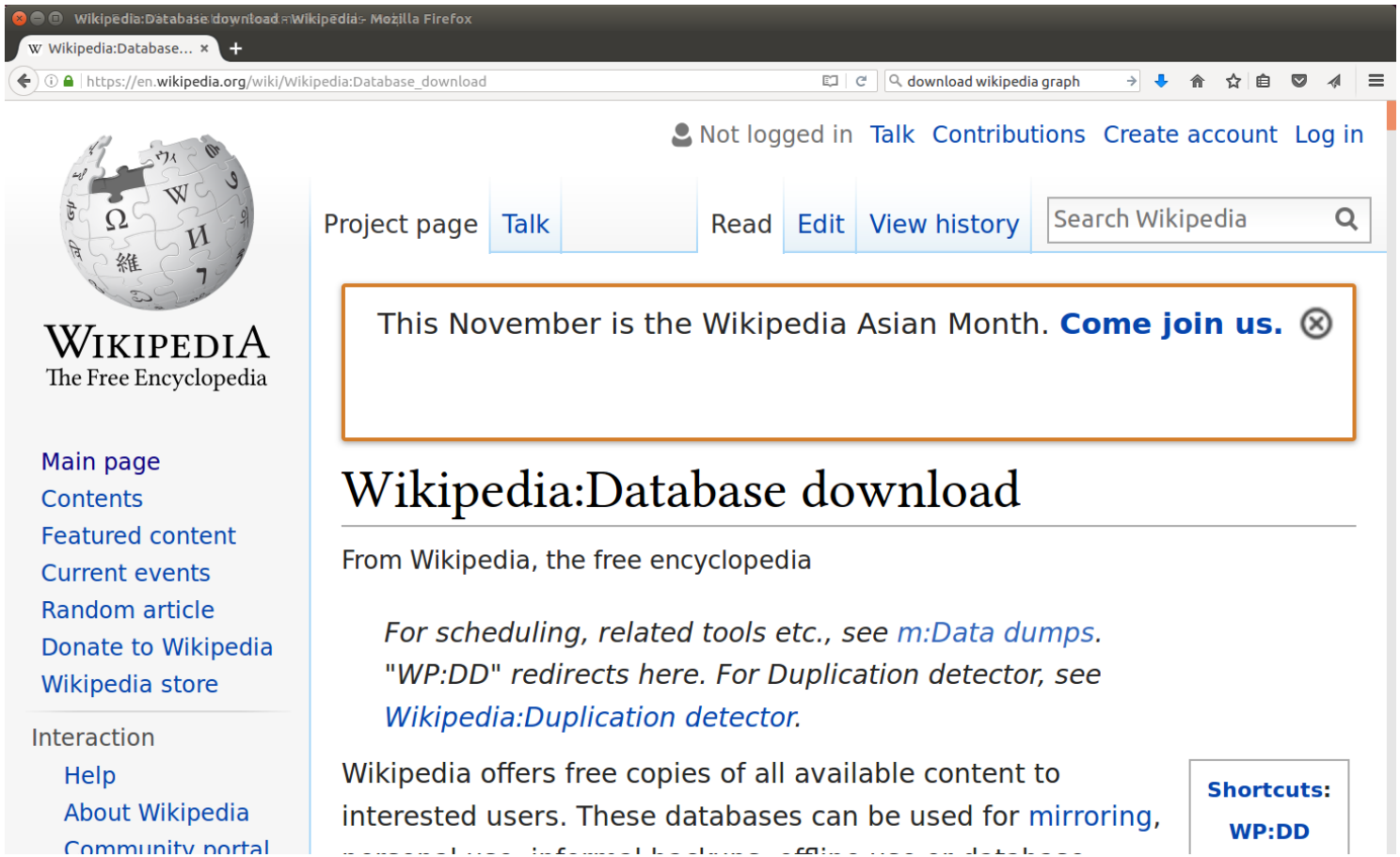

File Edit View History Bookmarks Tools Help
download wikipedi...
https://www.google.com/search?client=ubuntu&channel=fs&q=download+wikipedia+graph&ie=utf-8&oe=utf-8
download wikipedia graph
Google
download wikipedia graph
All
Images
News
Videos
Shopping
More
Search tools
About 5,610,000 results (0.57 seconds)
Wikipedia:Database download - Wikipedia
https://en.wikipedia.org/wiki/Wikipedia:Database_download
To download a subset of the database in XML format, such as a specific category or .... in our site load graphs at stats.wikimedia.org/EN/ChartsWikipediaZZ.htm.
You visited this page on 11/18/16.
Creates a Neo4j graph of Wikipedia links. - GitHub
https://github.com/mirkonasato/graphipedia
A tool for creating a Neo4j graph database of Wikipedia pages and the links between ... Assuming you downloaded pages-articles.xml.bz2 , follow these steps:.
Six Degrees of Wikipedia - Stephen Dolan
mu.netsoc.ie/wiki/
For the graph-theory nerds, there is no other disjoint strongly connected component of ... folks dump their entire database and put it online for all to download.
You visited this page on 11/18/16.
Wikipedia Link Graphs and Terrible Hacks - Tristan Hume

Use pre-existing Wikipedia dumps

Their XML file is about 13G so parsing it could be a challenge. First do some sanity checks on the data.



Same image.

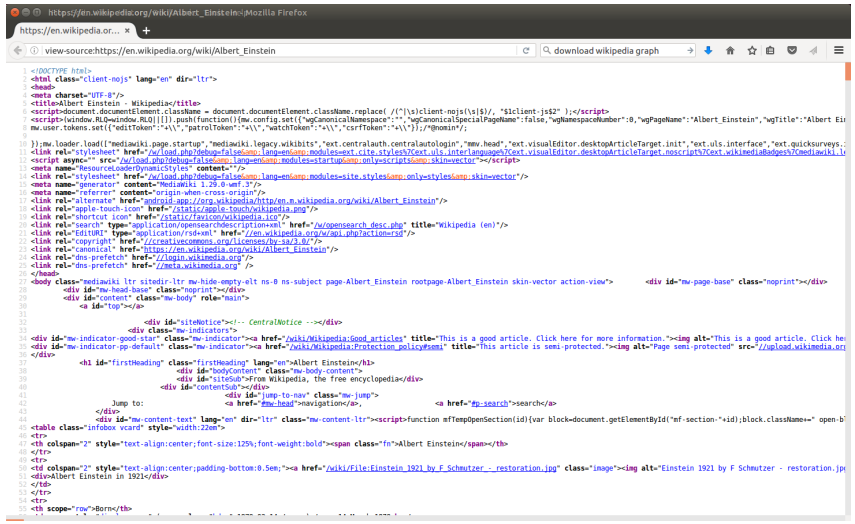

Wikipedia:Database download - Wikipedia - Mozilla Firefox
Wikipedia:Database...
https://en.wikipedia.org/wiki/Wikipedia:Database_download
download wikipedia graph
Not logged in Talk Contributions Create account Log in
Project page Talk Read Edit View history
Search Wikipedia
WIKIPEDIA
The Free Encyclopedia
This November is the Wikipedia Asian Month. Come join us.
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store
Interaction
Help
About Wikipedia
Community portal
Wikipedia:Database download
From Wikipedia, the free encyclopedia
For scheduling, related tools etc., see m:Data dumps.
"WP:DD" redirects here. For Duplication detector, see Wikipedia:Duplication detector.
Wikipedia offers free copies of all available content to interested users. These databases can be used for mirroring,
Shortcuts:
WP:DD

Albert Einstein's page

The system needs a place to start crawling Wikipedia, so we chose this page, just because.
Visual estimation of about 1,000 links.

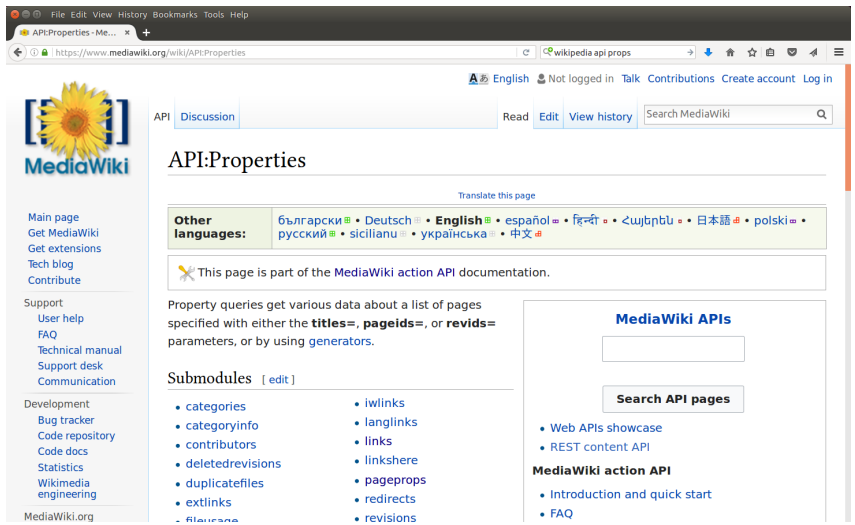
Albert Einstein's page source

Looking at the source for the page shows there are 2,774 links to other pages.
Makes for an interesting starting place.



Using Wikipedia's API

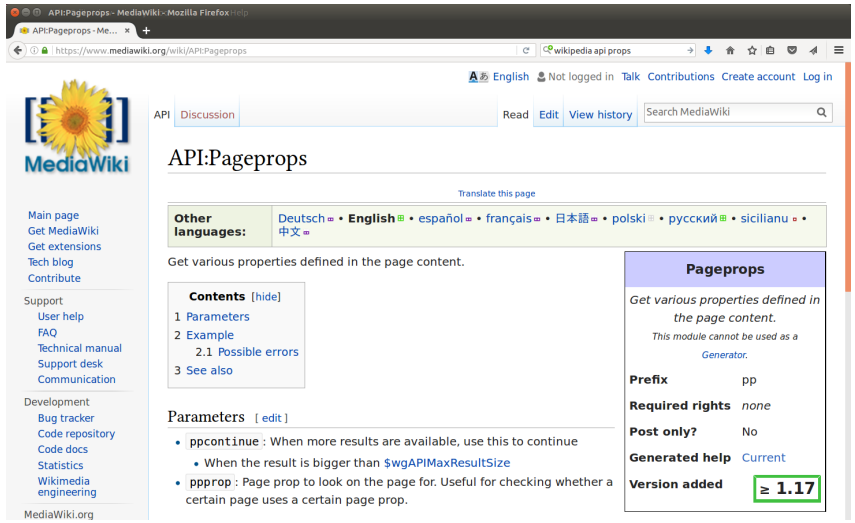
Wikipedia doesn't want you to crawl their stuff. So they provide an API to do things.



`https://www.mediawiki.org/wiki/API:Properties`

Wikipedia API for page properties

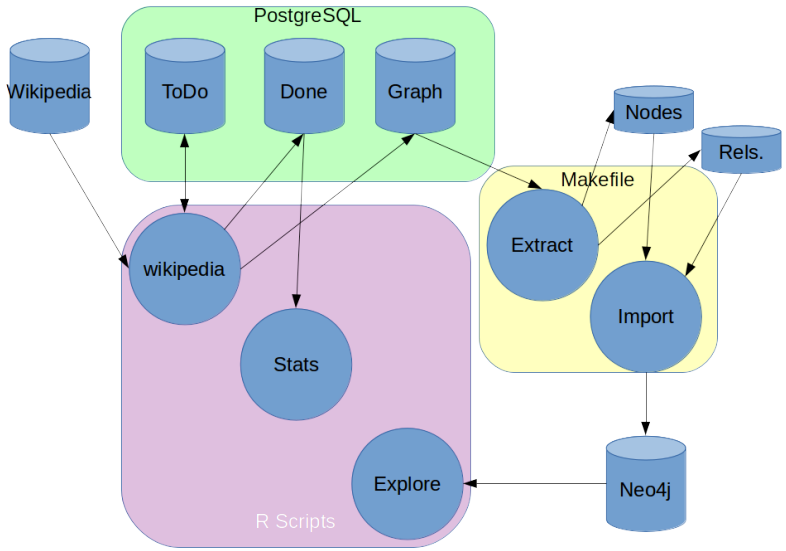
They do provide a way to request a page's properties, and then to page through the requested data. This is where we get our data.



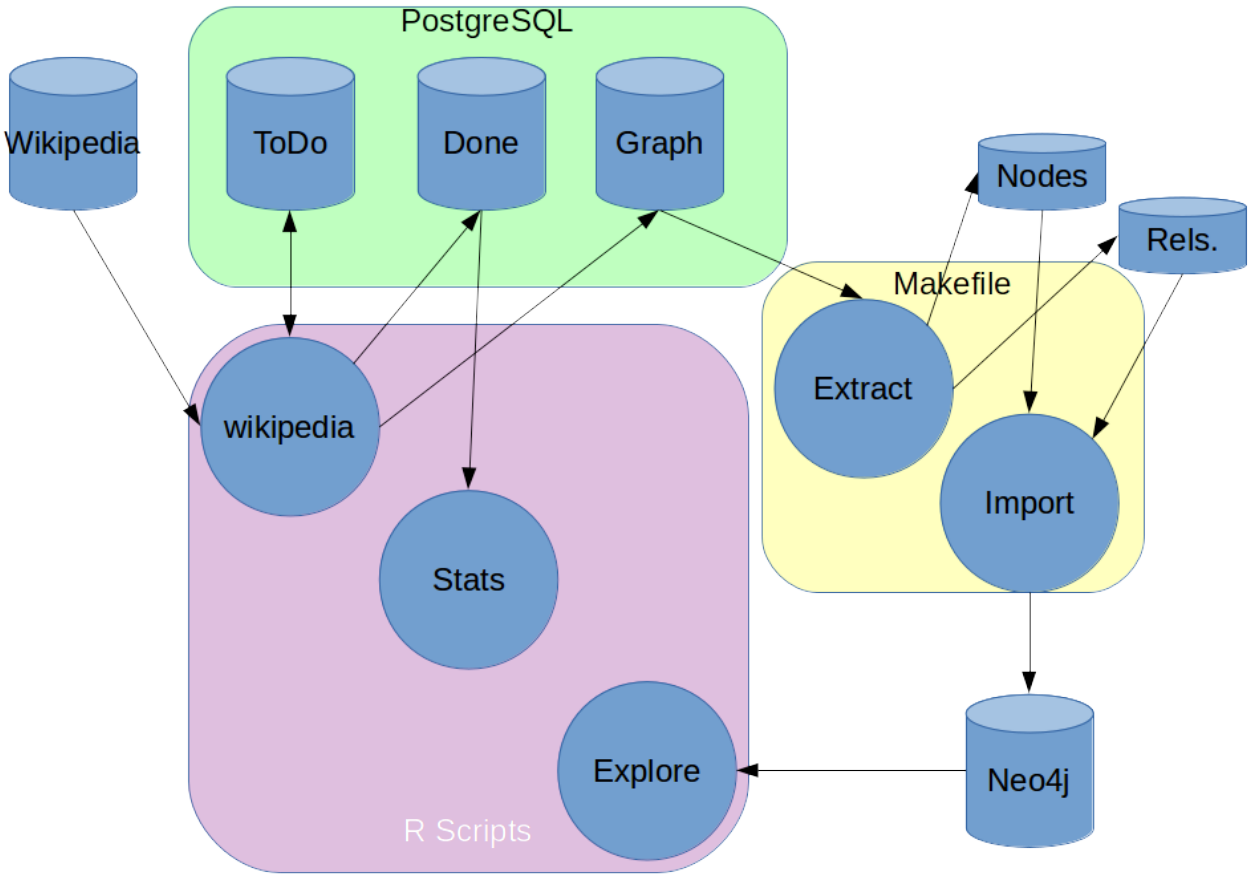
`https://www.mediawiki.org/wiki/API:Properties`

Our crawling system overview

- Green are PostgreSQL tables
- Purple are R scripts
- Yellow are makefile operations



Same image.


PostgreSQL
Wikipedia
ToDo
Done
Graph
Nodes
Rels.
Makefile
Extract
Import
wikipedia
Stats
Explore
Neo4j
R Scripts

50,000 foot view

Algorithm 1: 50,000 view of an exploring process.

Data: This node number, maximum number of pages to be processed, number of nodes
Result: Updated ToDo and Done queues in the PostgreSQL database
nodeAssignment = 0;
while **Data:** get the current number completed pages **do**
    if current number > limit **then**
        break from while loop
    Data: get next set of pages to process for this node
    for each page in work set **do**
        if page not in Done table **then**
            Data: get all outgoing links from the page
            for each outgoing link **do**
                nodeAssignment = nodeAssignment + 1;
                Data: update ToDo table with page, and nodeAssignment mod number of nodes
            Data: increment the number of completed pages
        Data: remove page from ToDo table

Node 0 processing

Algorithm 2: Special processing for node 0.

Data: All command line arguments
Result: Depends on command line arguments, possibly reset databases, explorer nodes started in the background

```
nodeAssignment = 0;
if my node number = 0 then
    if reset database flag is TRUE then
        reset PostgreSQL database and create tables ;
        add seed page to ToDo table;
        reset Neo4j database;
    reset the of completed pages to 0;
    for i in (number of desired nodes - 1) do
        start background nodes with n, p, and N command line arguments;
```

Command line arguments

Table: Command line arguments. The order of the command line options is not important. If the same option appears more than once, only the last one will be used. Be aware that the **r** option does not take a value. If it is present, then it's value will be TRUE.

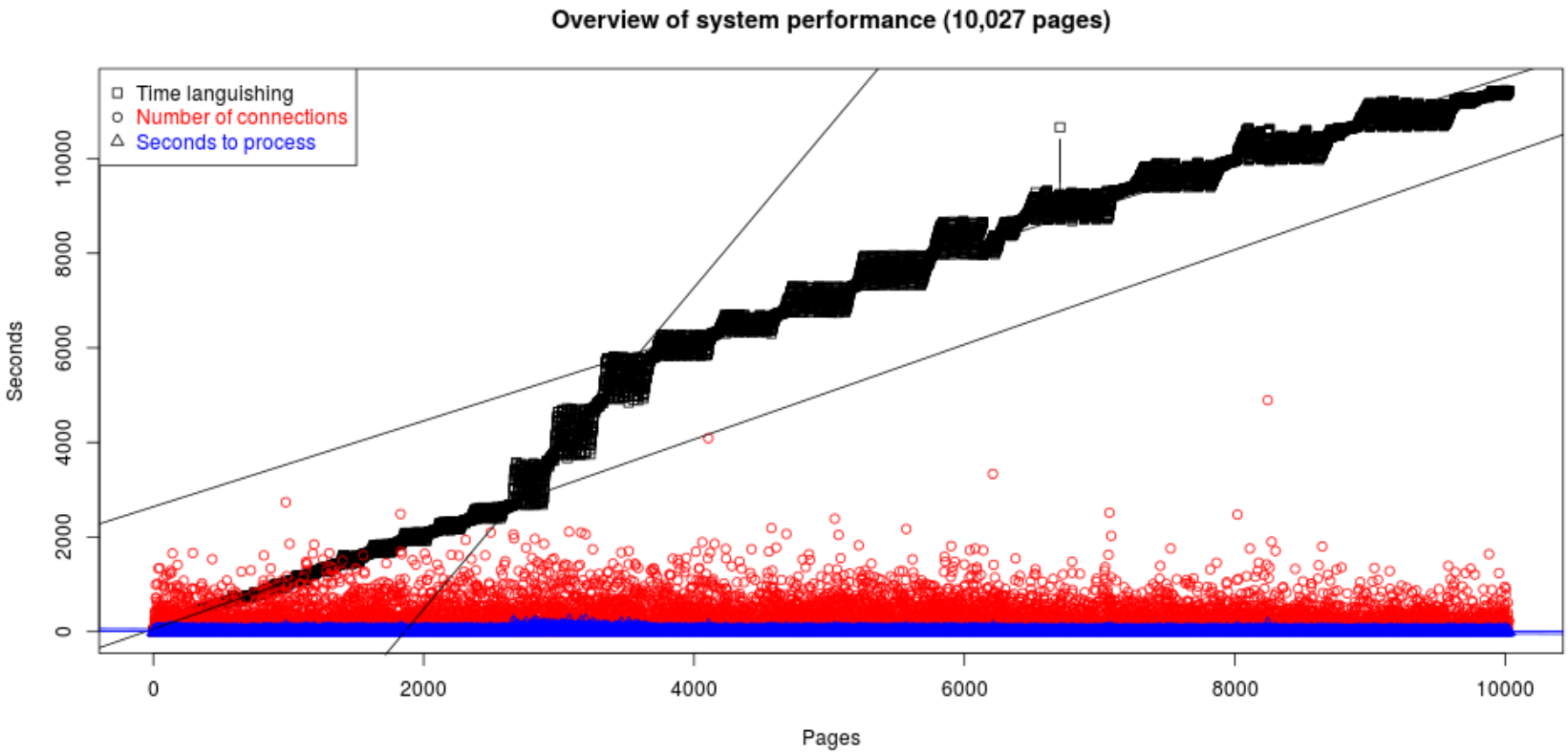
Letter	**Default**	**Meaning or use.**
n	0	This explorer instance number.
N	9	Number of explorer nodes.
p	1000	Maximum number of pages the system should process.
r	False	Reset the databases to empty.
s	"Albert Einstein"	Default seed page.

Crawling Wikipedia

Table: System behavior for 10,027 pages. The system was directed to process 10,000. Due to the way the Limits table was checked, a few more pages were processed.

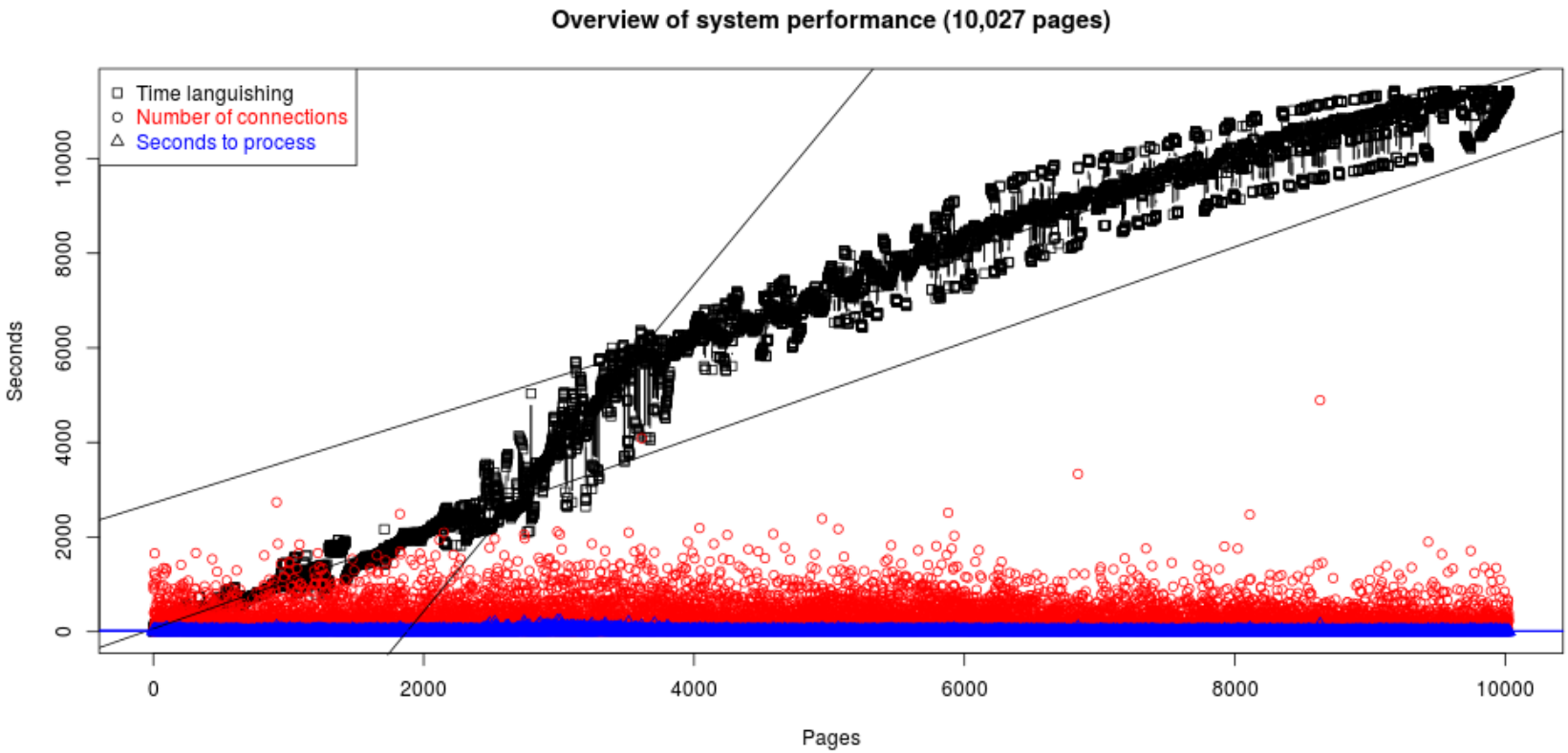
Seconds	Activity
11,460	Explore Wikipedia.
90	Extract data from the PostgreSQL database and create Neo4j import files.
52	Stop the Neo4j database, import the new data, and start the Neo4j database.

Data from Done table (1 of 4)



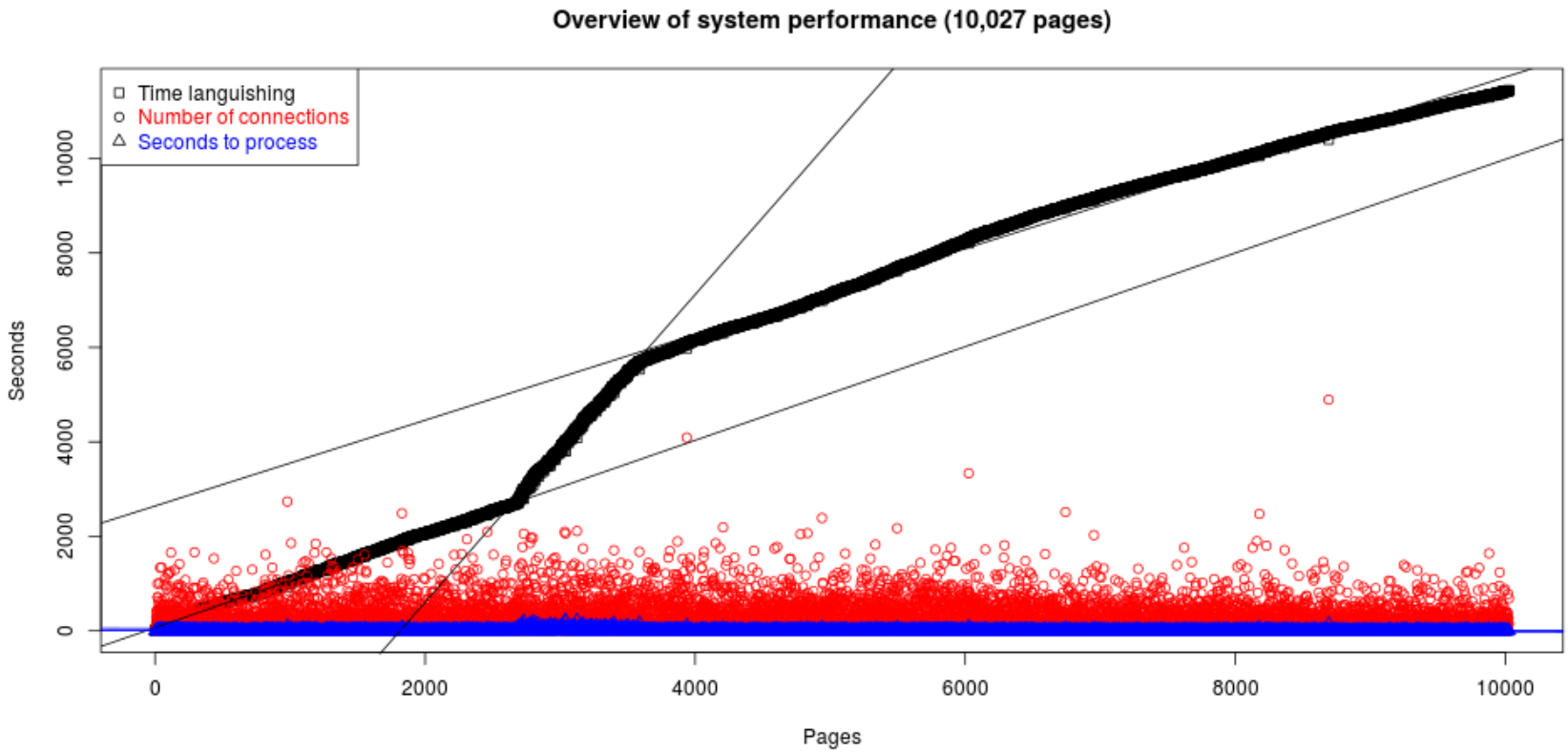
Data returned from raw select statement.

Data from Done table (2 of 4)



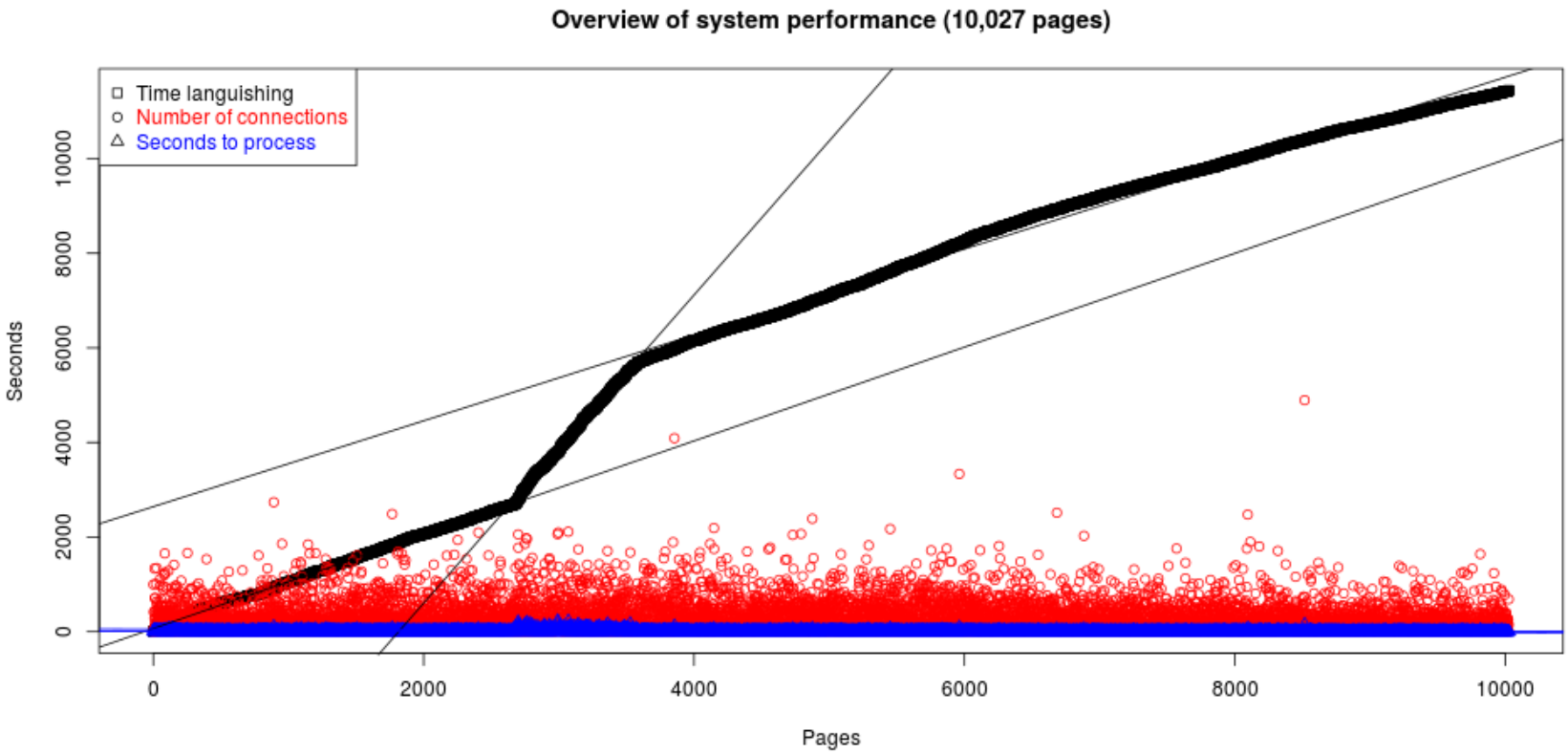
Data returned from select statement ordered by "Added" column.

Data from Done table (3 of 4)



Data returned from select statement ordered by "Finished" column.

Data from Done table (4 of 4)



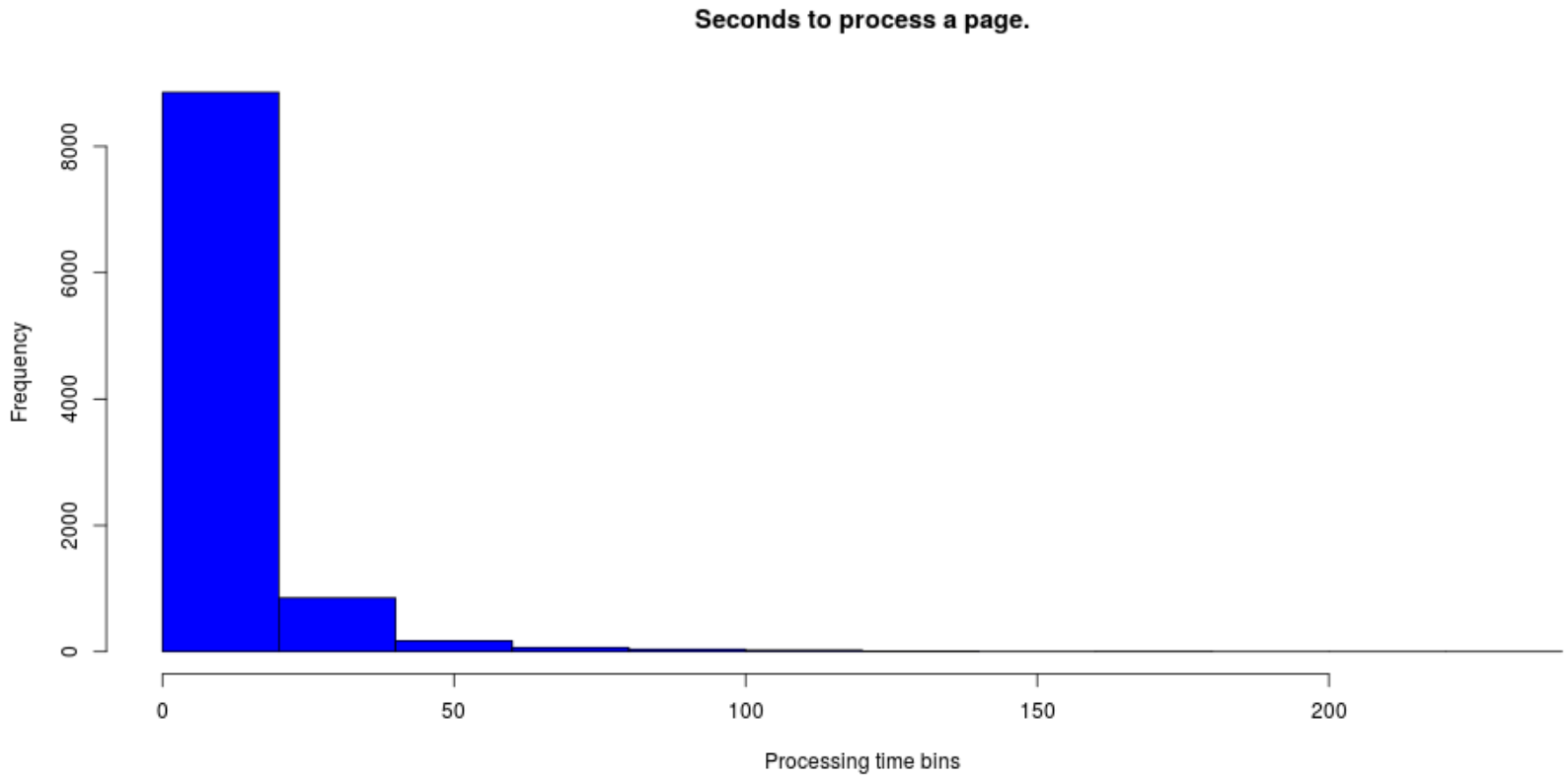
Data returned from select statement ordered by "Started" column.

Examining the data

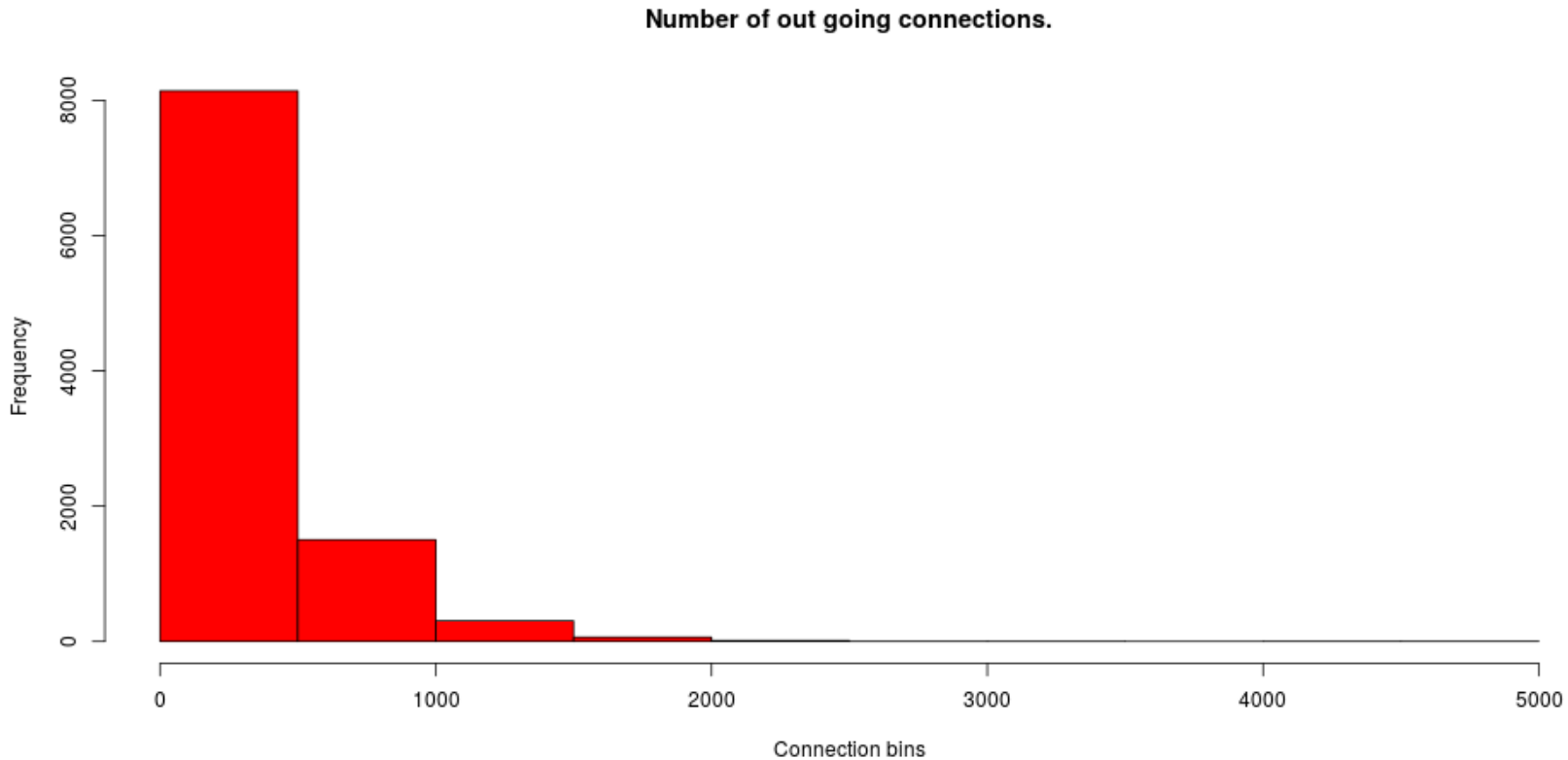
Table: Linear modeling of system performance. Examination of the system performance plots revealed three distinct areas (segments 1, 2, and 3). Linear equation coefficients for each segment for languish and processing times were computed.

Seg.		Languish		Processing	
		Intercept	Slope	Intercept	Slope
1	Raw	59.490919	1.001918	9.243165	0.000066
	Started	68.276183	0.991335	9.202748	0.000190
	Added	64.618835	1.008543	8.349092	0.001286
	Finished	68.306086	0.991338	9.312100	-0.000006
2	Raw	-6323.104973	3.399055	60.916314	-0.010478
	Started	-5910.560290	3.257141	55.416966	-0.008326
	Added	-6472.055024	3.450379	32.679855	-0.002093
	Finished	-5909.112869	3.256709	52.957257	-0.007515
3	Raw	2642.458391	0.907123	14.625972	-0.001034
	Started	2643.748884	0.907760	13.590724	-0.000906
	Added	2719.591729	0.894666	14.611140	-0.001037
	Finished	2643.622983	0.907775	13.724723	-0.000920

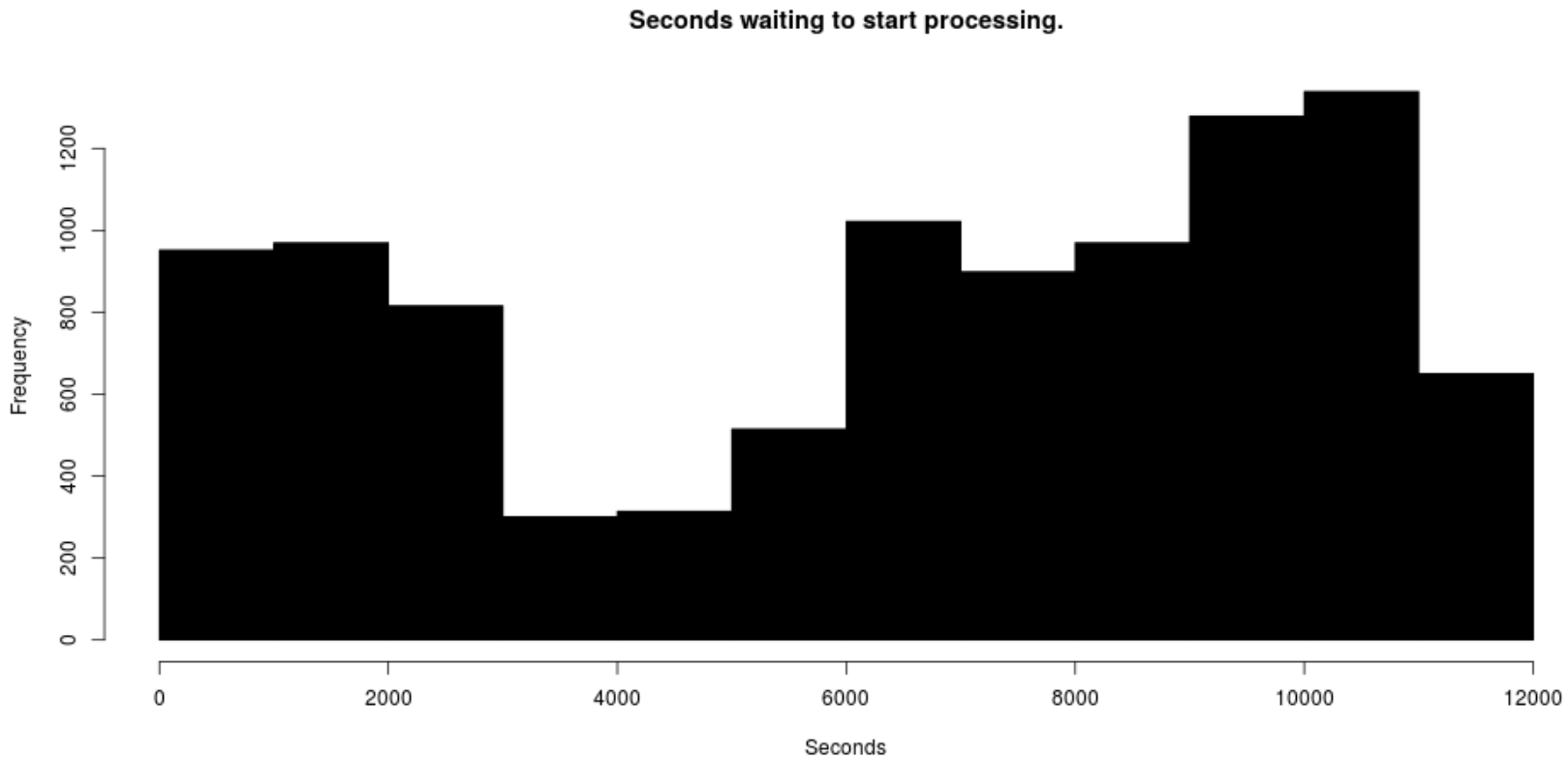
Histogram of processing time



Histogram of connections



Histogram of languishing time

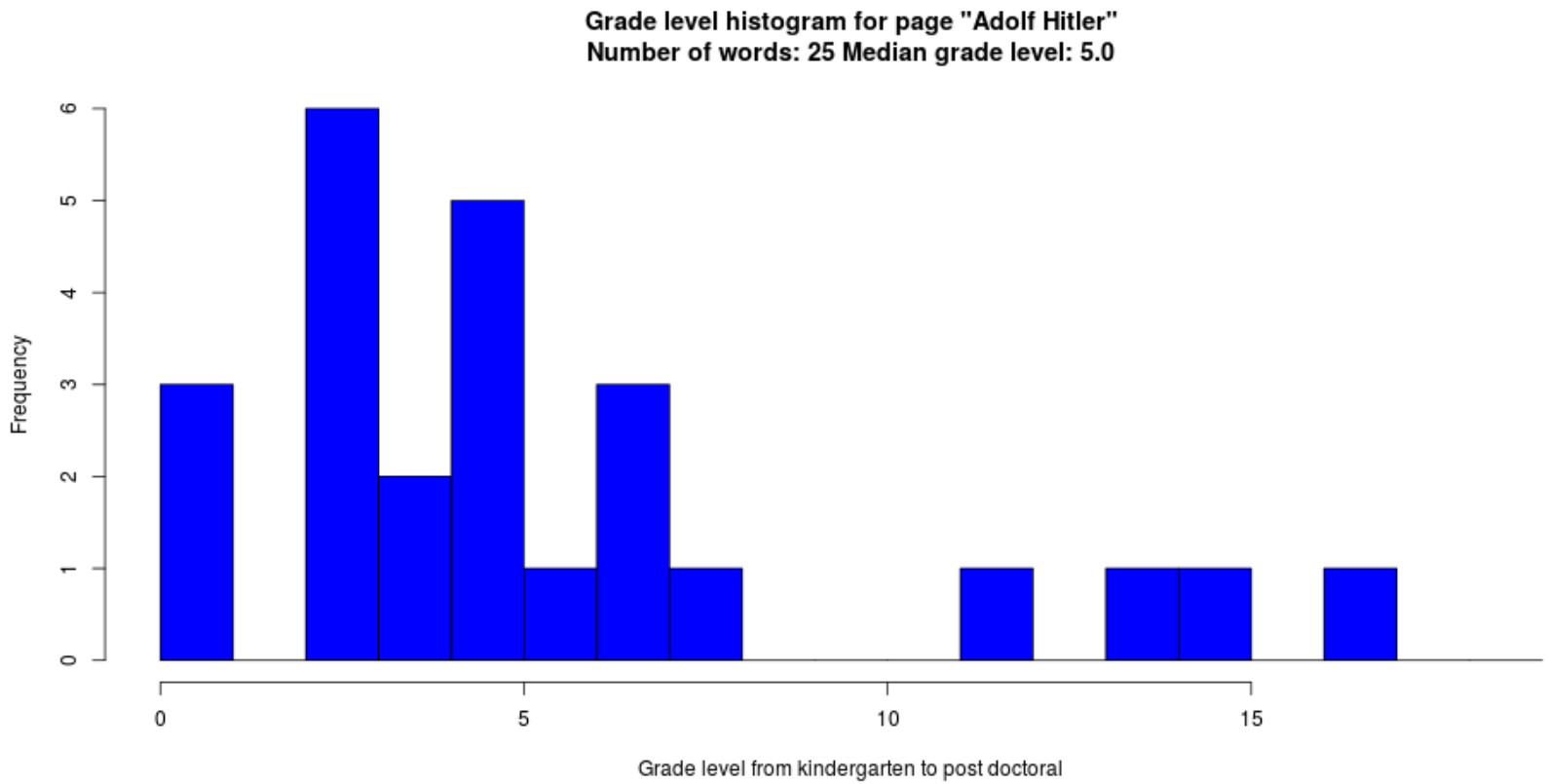


Connecting the dots

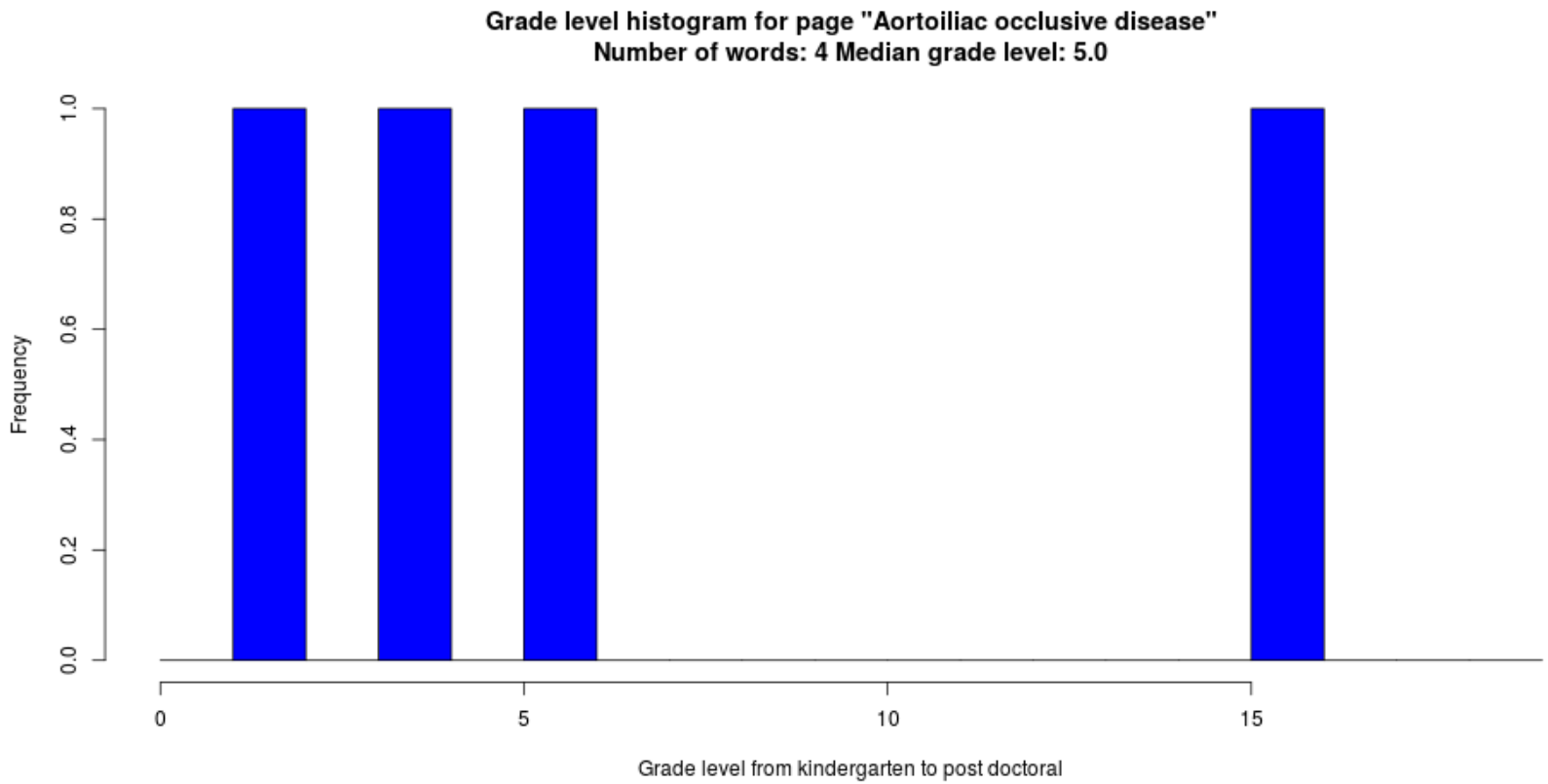
Table: Connecting the dots between two pages. The between the original source and destination is different than the path from the destination to the source.

Output	Explanation
Path length = 3	A path was found and it has 3 edges.
Path starts (page titles).	A header.
————	A header.
Aortoiliac occlusive disease	The starting page. "... disorders of the two major blood vessels that feed the lower half of the body"
Atherosclerosis	"... a disease of the arteries characterized by the deposition of plaques of fatty material on their inner walls"
Korean War	"The Korean War began when North Korea invaded South Korea."
Adolf Hitler	"... leader of the Nazi Party and became Chancellor of Germany in 1933."
————	A trailer.
Path ends.	
	Blank line.
Source and destination nodes are swapped.	Informative statement.
	Blank line.
Path length = 3	A path was found and it has 3 edges.
Path starts (page titles).	A header.

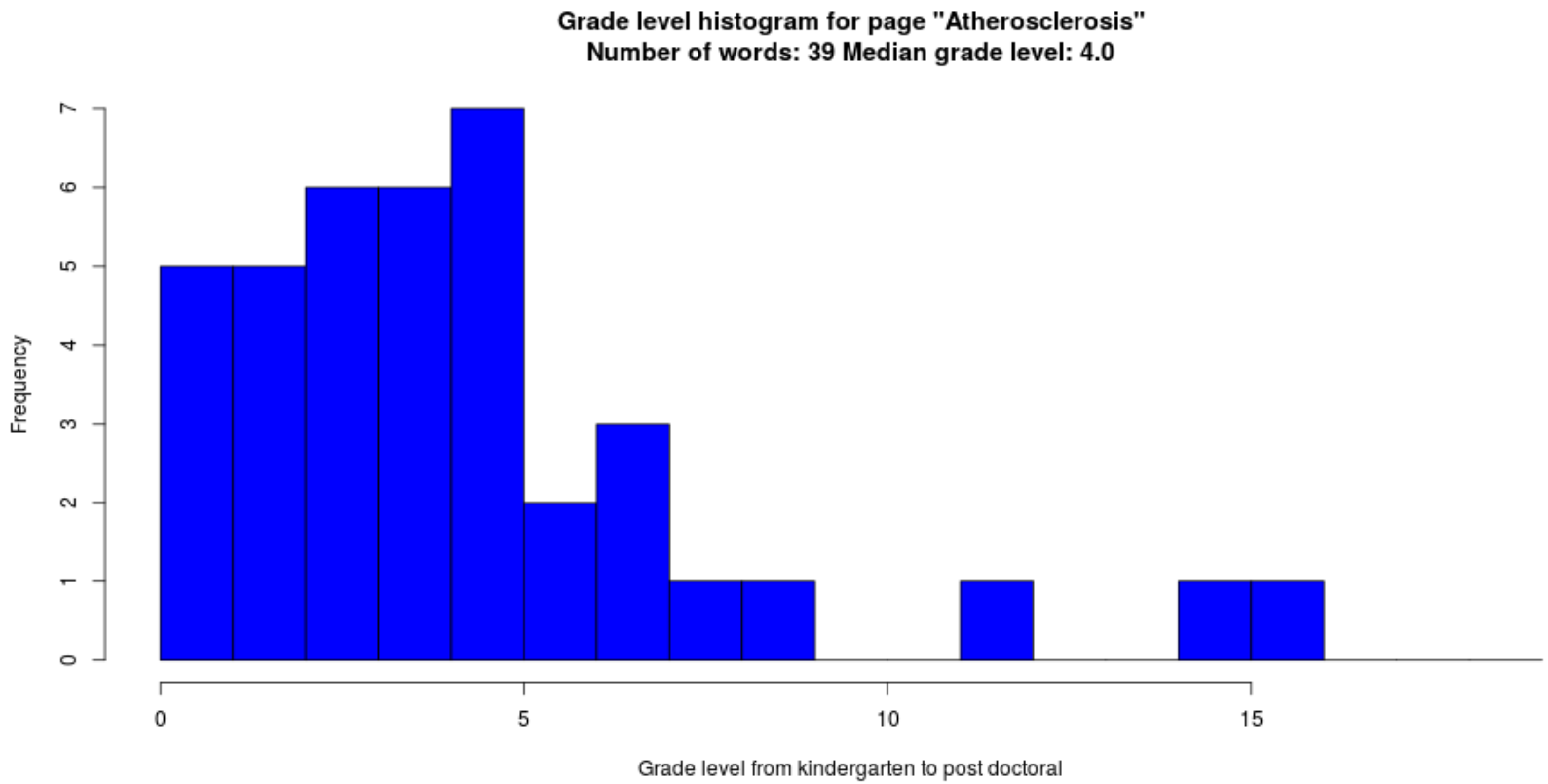
Grade level histogram for “Adolf Hitler”



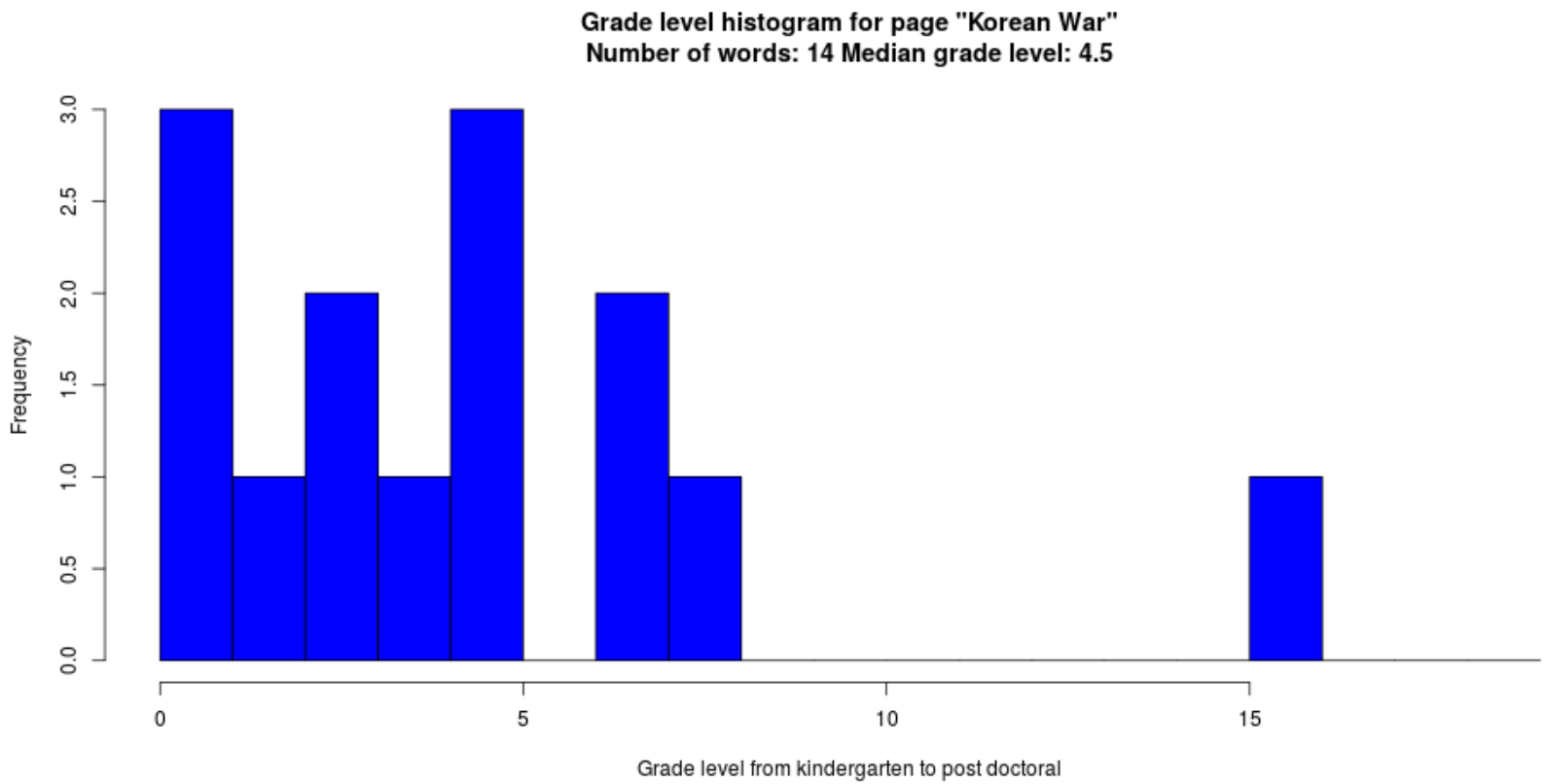
Grade level histogram for “Aortoiliac occlusive disease”



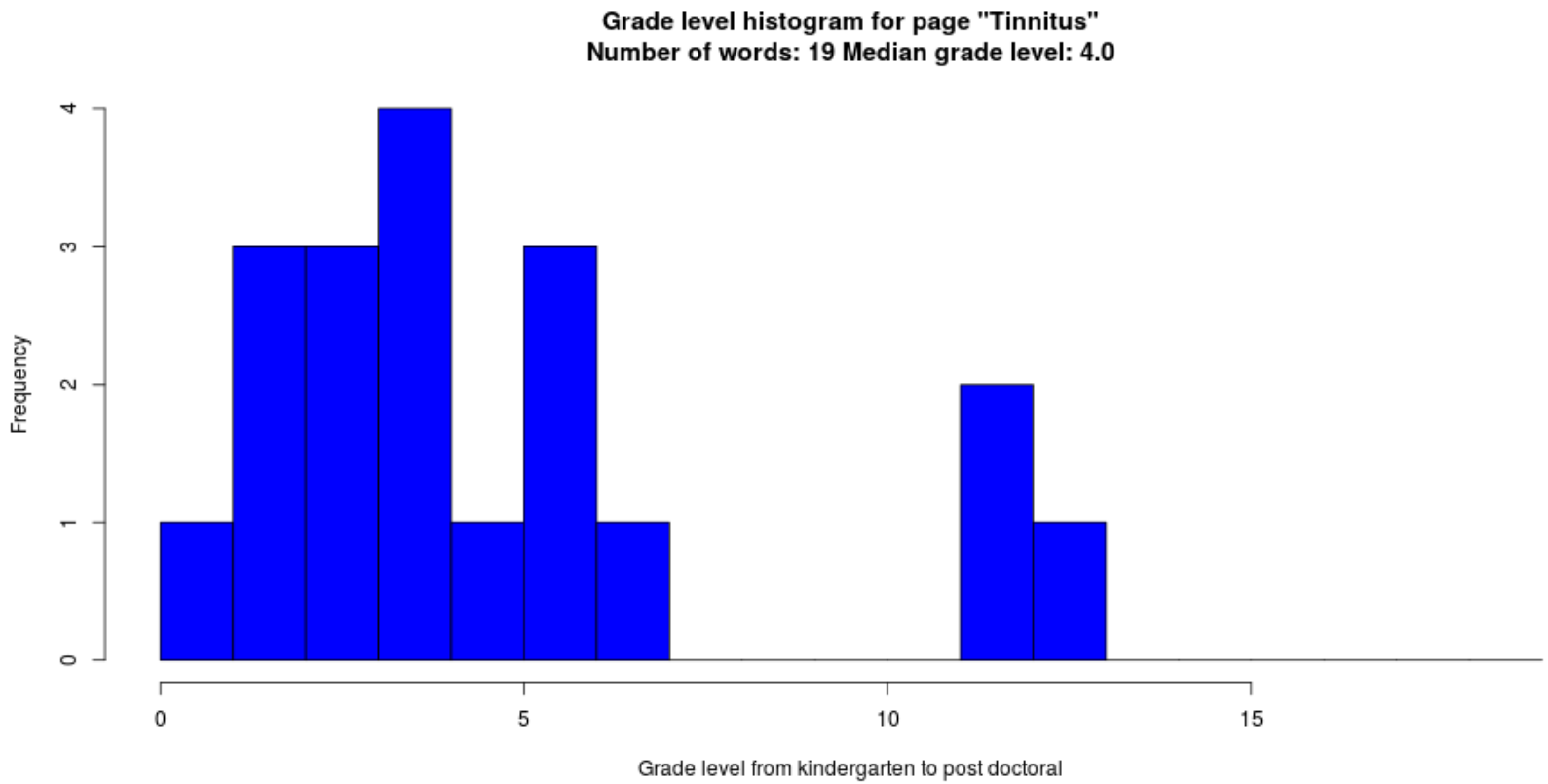
Grade level histogram for “Atherosclerosis”



Grade level histogram for “Korean War”

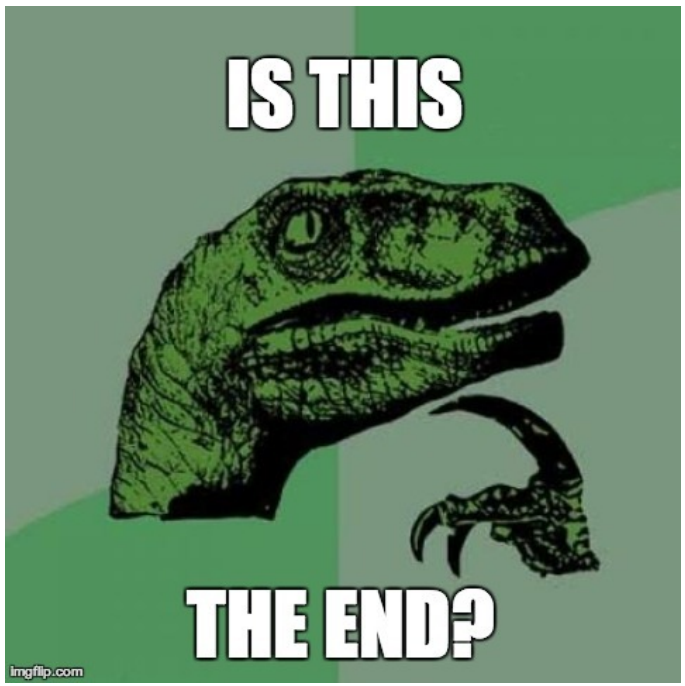


Grade level histogram for “Tinnitus”



What have we covered?

- **Don't do this at home. Or anywhere else.**
- Wikipedia makes their database available for download, use it.
- Based on the small sample: it would take about 2 months to down Wikipedia using 9 crawlers.
- A PostgreSQL database can be an adequate process scheduler.
- A Neo4j database can answer the question of how arbitrary pages are connected.



Next time: see where the wind blows

Files of interest

1. library.R – a collection of functions used by other scripts
2. nodes.psql – PostgreSQL select command to get a list of unique pages/nodes
3. relations.psql – PostgreSQL select command to get a list of connections/arcs
4. wikipediaExplore02.R – an R script to explore the Neo4j graph
5. wikipedia.R – an R script to download Wikipedia pages and update the PostgreSQL database
6. wikipediaStats.R – an R script to analyze the performance of the system and create plots
7. wikipediaStats.R – an R script to analyze the performance of the system and create plots
8. Makefile – a makefile to coordinate and manage all exploration activities

PostgreSQL tables (1 of 3)

Table: The PostgreSQL ToDo and Done tables.

Col. name	Type	Explanation
title	character varying(8000)	Base 64 encoded of the Wikipedia title. Primary key.
node	integer	The node (explorer) assigned to process this page.
added	integer	Unix seconds when this page was added to the work queue.
started	integer	Unix seconds when the node started to process this page.
finished	integer	Unix seconds when the node finished processing this page.
languished	integer	Unix seconds between the time when the page was added to the queue and when the node started to process the page.
processed	integer	Number of seconds to process the page.
connections	integer	Number of outgoing links in the page.

PostgreSQL tables (2 of 3)

Table: The Graph PostgreSQL table.

Col. name	Type	Explanation
id	serial	Primary key.
sourcenode	character varying (8000)	The page with the out going connection.
destinationnode	character varying (8000)	The page that is connected to by the source.

PostgreSQL tables (3 of 3)

Table: The Limits PostgreSQL table.

Col. name	Type	Explanation
limits	integer	The total number of pages currently explored by all explorers.