Operant Subjectivity: The International Journal of Q Methodology
*41 (*2019): 17-47, DOI: 10.15133/j.os.2019.010

Operant Subjectivity
The International Journal of Q Methodology

Tools for Collecting a Concourse and Selecting a Q Sample

Byung Lee
Elon University

Abstract: As Q methodology matures, Q scholars have witnessed the development of tools for Q sorting and factor analyzing Q sorts. Regarding concourse statements, however, some scholars failed even to mention how they constructed the concourse in their research. Gleaning concourse statements and sampling from the statements requires persistent efforts from Q researchers. Software for these tasks could lead to more effective Q studies. To expand Q as a tool for solving urgent problems in society or as a pedagogical tool in the classroom, Q researchers need to implement their Q studies with swiftness. Researchers may need a tool to accelerate the entire process. This article describes MarginNote 3 for Mac users, QDAMiner Lite for PC users, Dedoose for web users, and Weava extension for Google Chrome web browser users. These tools, including iThoughtsX, quickly categorize statements into themes. Q researchers are able to tap text mining algorithms like LDA (Latent Dirichlet Allocation) and R's widyr package thus quickly revealing aspects of a topic that may have been neglected or missed and retrieve statements that represent those same aspects.

Keywords: concourse construction, facilitation, Q methodology, Q sampling, software

Introduction

Q methodology has generally been used for three purposes, academic research, solving problems, and as a pedagogical tool. A Google search for "Q methodology journal articles" displayed about 126,000,000 results (0.75 seconds) in July 2019, which shows the extent to which Q methodology is an integral component of much research. In an example of Q at work to solve authentic problems, the school of communications where I, the author, am teaching planned to restructure, and I was asked to use Q. After compiling statements from the faculty and staff and having them sort a Q sample, I was able to show the result that summarized what they preferred in the future of their school. Q also can be used as a pedagogical tool for classroom learning. Students can be involved in “deep reflection, discussion, and analysis and interpretation of” (Walker, Lin, & McCline, 2018, p. 459) their Q sorts through which they express their perspectives on the issues they examine in class.

Whatever the purpose of a Q study is, it goes through five steps: collecting a concourse, selecting a Q sample from the concourse, Q sorting, analyzing Q sorts, and interpreting the results. Q researchers have witnessed many developments in software programs for Q sorting and analysis.

Q sorting has experienced recent changes that offer opportunities other than those with cards or pieces of paper, most notably the computerization of the sorting process.

Contact author: byunglee@elon.edu

So far, about two dozen programs have been discovered ("Q-Assessor," n.d.; Schmolck, 2014). There are a few variations among Q-sorting software. It started with a simple text-based web-based Q-sorting program, WebQ (Schmolck, 1999). This shifted to a drag-and-drop feature to imitate the face-to-face sorting. FlashQ was programmed based on Flash's Action Script. Its online version, however, requires a user to set up a server with PHP. Its offline version has limitations like needing an Internet connection ("FlashQ," 2007). LeeQSorting, which is based on the JavaScript Phaser engine, has a feature of toggling text between English and Korean while sorting (Lee, n.d.). Users, however, cannot easily modify the code for their use without the knowledge of JavaScript and Phaser. Q-perspectives allows sorters to enter face-to-face sorting results on a computer screen or to complete the entire sorting online (Walker, Lin, & McCline, 2018). VQMethod allows images, audio, and videos as sorting items (Nazariadli, Morais, Supak, Baran, & Bunds, 2019). Q-Assessor uses Ruby to combine Q sorting and Q analysis in one system ("Q-Assessor," n.d.), but it costs $500 per month per investigator ("Q-Assessor Subscriptions," n.d.).

Likewise, Q analysis programs also have seen many recent developments. Norman Van Tubergen developed QUANAL as a FORTAN program for mainframe platforms in the 1960s ("Software," n.d.). It offers PCA with Varimax Rotation, and a unique feature separating a bipolar factor into two factor arrays for an easier analysis ("QUANL Program," n.d.). The Korean Society for the Scientific Study of Subjectivity has distributed a Korean-language manual for QUANAL, whose PC version has been used mostly by Korean Q researchers. Schmolck ported QMethod, a Fortran Mainframe program developed by John Atkinson in 1992 (Schmolck, 2018) for use on PCs and Macs. A few Q analysis programs were developed using languages other than FORTRAN. PCQ running on a PC was developed using Java. The program, which was once priced at $400 (Stricklin, 2011), has recently been made freely available. Python-based LeeQAnalysis offers the Varimax method, manual oblique rotation (Lee, 2019), along with PCA and Principal Axis Factoring. R was also used to develop qmethod (Zabala, 2018). KenQ Analysis and its desktop version KADE (Ken-Q Analysis Desktop Edition) used JavaScript and a JavaScript library, React for building User Interfaces. It can be used online and offline. Since it relies on a web browser, it does not require the installation of additional software (Banasick, 2019) beyond a web browser. QFACTOR was developed as a Stata module (Akhtar-Danesh, 2017), so all Stata factor analysis methods can be tapped if users have Stata installed on their computer.

Thus, Q sorting and Q analyses have software options; however, this has not been the case for concourse development and Q sample selection. Accordingly, this article introduces software tools with which Q researchers can do their Q study more effectively and swiftly in relation to these initial stages.

The Nature of Concourse

William Stephenson (1953) suggested four rules related to Q methodology. Here we are concerned with his first rule, which is related to concourse and the Q sample, "The statements in each universe should be of one class" (p. 194). Thus, this paper describes tools that can help Q researchers find concourse statements from a wide area, in a robust and systematic procedure, while satisfying the first condition. The representativeness of questions, or Q statements, in Q study is as critical as the randomness of subjects in R study. However, many Q papers have not discussed how they constructed the concourse (Kenward, 2019). Of 44 studies found in the health area, Kenward found 23 (52%) stated no strategy. Out of the remaining 21, 10 studies used an inductive method in which they generated themes while analyzing literature, interviews, focus groups, or expert/professional panels. Another 11 studies used the deductive method, in which they used theoretical frameworks to develop concourse statements. Kenward also suggested that in addition to the inductive thematic and theory-driven approaches, Q researchers devised their own robust framework to demonstrate their engagement with the concourse.

In this article, I propose the use of new software programs that enable Q researchers to select concourse statements from many information sources and narrow them down to a Q sample while satisfying the first condition above. Here, what is crucial is not only the quality of a concourse and a Q sample that Q researchers can secure, but also the speed with which they can make a decision during a crisis or classroom discussion on current events.

Tools for Collecting Concourse Statements

Q statements can come from many sources, "listening to people talking about the topic in magazines, radio, TV, hearsay, scientific journals, letters to the publisher, interviews, questionnaires, arguments in a debate, transcripts of focus groups, lists prepared by nominal groups, etc." (Barbosa, Willoughby, Rosenberg, & Mrtek, 1998, p. 1034).

In this section, I describe tools for collecting the four different types of Q statements: 1) APIs (application program interfaces) for ecological statements that are collected in a natural situation; 2) Google sheet for statements to solve practical issues facing an institution; 3) software for statements from written material, probably for a theory-driven study; and 4) text-mining algorithm for statements selected based on words that might be associated with potentially meaningful themes using R's Latent Dirichlet Allocation (LDA) or R's widyr package. Some of these tools can be used in not only collecting statements but also grouping them by themes if the themes are used as tag labels.

APIs for ecological statements from mass media and social media

Self-referent statements can be obtained by engaging a person in conversation or getting that person to write about a phenomenon of interest to the researcher (Stephenson, 1967, p. 14). Ecological statements can be easily found online, for example, in the comment section of a newspaper or on a social media platform.

Comments on news sites are less numerous, but seem to be more detailed, than those on social media sites, and as such, researchers should glean comments from both sources. According to a survey (Stroud, van Duyn, & Peacock, 2016), about 53.5% of 1,471 respondents posted comments, and 77.9% read them on a news site and social media combined. The portion for the former is small, for example, only 14.6% of the survey participants left comments on the news site or its app. Online mass media comments, like those from *The New York Times,* however, are rich in content and can be gleaned easily by using the Community API ("Community API," n.d.) or just copying while scrolling through comments.

Many Q-type comments in the form of tweets can be acquired from Twitter. Twitter launched a new Premium API in November 2017 for those who could not afford the Gnip enterprise APIs (Cohen, 2017), which cost users several thousand dollars. Now, a Premium API user can access tweets since Twitter's 2006 inception, after paying just

$99 for 100 requests per month ("Pricing", n.d.), and each request can download up to 500 tweets.
The following shows how to use the Twitter API: After applying for a premium account at https://developer.twitter.com/en/premium-apis.html, a user needs to acquire a "consumer_key" and a "consumer_secret" to download tweets.

Shown below is the code for downloading tweets with Twitter's Premium API code ("Python wrapper," n.d.). First, researchers should create a file, twitter_keys.yaml, which contains the two pieces of information above, and save the file in one's user name folder on a local computer:

```
1.  search_tweets_api:

1.   account_type: premium
2.   endpoint: https://api.twitter.com/1.1/tweets/search/fullarchive/research.json
3.   consumer_key: [redacted]
4.   consumer_secret: [redacted]
```

Another file, for example, tweets.py, needs to be created as shown below:

```
1.    from searchtweets import ResultStream, gen_rule_payload, load_credentials

5.    from searchtweets import (ResultStream,
6.                          collect_results,
7.                          gen_rule_payload,
8.                          load_credentials)
9.
10.   search_args = load_credentials("~/.twitter_keys.yaml",
11.                                  account_type="premium",
12.                                 yaml_key = "search_tweets_api",
13.                                 env_overwrite=False)
14.   search_args = load_credentials(filename="~/.twitter_keys.yaml",
15.                           account_type="premium")
```

The function below is needed to specify the search date, search terms, the maximum number of tweets, and a file name to save downloaded tweets.

- Lines 16-32: a function "download" is created so that a user can fix the time period from which tweets can be downloaded.
- Line 17: "magicDate" is used as part of a file name on line 32.
- Line 18: "toDate" is used with "fromDate" on line 22 to show the range of dates from which tweets are extracted.
- Line 21: search terms that are related to e-cigarettes
- Line 25: downloaded tweets will be saved as "results_list."
- Line 26: "max_results" was set to 500 since this is the largest number allowed.
- Line 29: the Python package of pandas (pd) will convert tweets in JSON format to a Pandas' dataframe.
- Line 30: the command will show the dimension of the dataset, the number of rows, and columns. This code is used to check whether tweets were really downloaded.

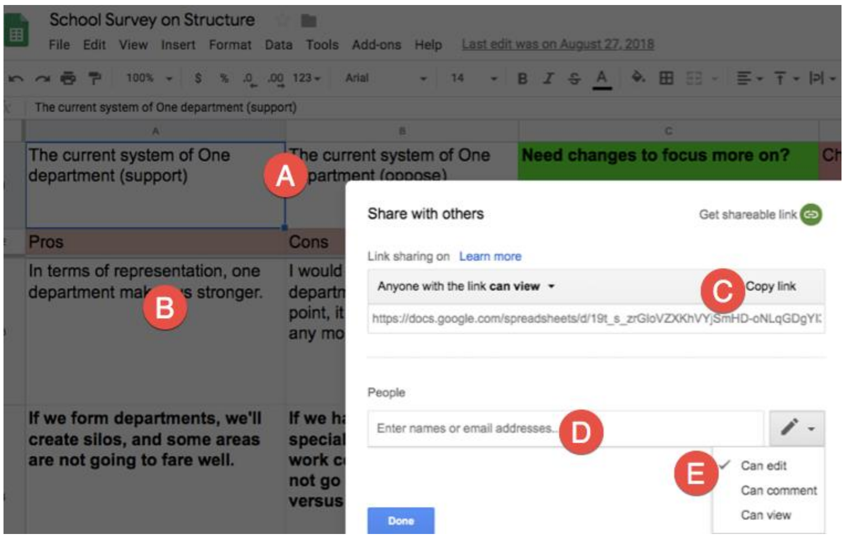
- Line 32: the downloaded tweets are saved as a csv (comma-separated values) file.
- Line 35: a user can set the time from which tweets can be downloaded. The first number represents a year, and the second, a month.

```
16. def download(yearNum, monthNum):
17.     magicDate = yearNum + "-" + monthNum + "-01"
18.     toDate = yearNum + monthNum + "010000"
19.
20.
21.     rule_b = {"query":"(\"Electronic Vaporizer\" OR \"electronic cigarettes\" OR #CloudChaser OR
    #CloudChasing OR #Ecig OR #Ecigarettes OR #ecigdeals OR #ecigs OR #vape OR
    #VapeCommunity OR #VapeFam OR #VapeLife OR #VapeNation OR #VapePorn OR #vapers OR
    #VapeTricks OR #vaping OR #vaping101 OR #VGod OR \"E-cigs\" OR \"e-juice\" OR \"e-liquid\"
    OR ejuice OR eliquid OR vaper OR vaping OR vaporizers OR vapors) \
22.                 lang:en -
    is:retweet","maxResults":500,"toDate":toDate,"fromDate":"201001010000"}
23.
24.
25.     results_list = collect_results(rule_b,
26.                          max_results=500,
27.                          result_stream_args=search_args)
28.
29.     resultDF = pd.DataFrame(results_list)
30.     print(resultDF.shape)
31.
32.     resultDF.to_csv(magicDate + ".csv", sep = ","  )
33.
34.
35. download("2010", "08")
```

The code shown above collects tweets that discuss e-cigarettes. By changing search terms, Q researchers can find tweets regarding any topic of interest while obtaining ecological Q statements among them.

Google sheets for quickly generating a concourse. When changing its policies or making an important decision, an institution may want to do a Q study with its stakeholders. Statements can be generated from interviews or focus groups with the stakeholders in the traditional way. But to accelerate the process, one can utilize a method to interactively glean subjective statements online. For example, when my school planned to restructure its organization, I briefly gathered statements from a few key stakeholders and displayed emerging themes (A in *Figure 1* below) and the statements that correspond to each theme (B) in a Google sheet below (*Figure 1*). Then I created a link to the spreadsheet (C) and sent it to potential participants (D) for their participation (E) in the process of creating concourse statements. Participants were asked to write their thoughts under the existing themes or to add new themes as needed. This process can also be used in brainstorming sessions. Then, all statements can be collected in one meeting if key stakeholders are present. In addition to Google Drive, Dropbox and Box also offer comparable file-sharing services ("Top 3," n.d.).

Figure 1. Google spreadsheet can be used to collect people's thought on an issue.


School Survey on Structure

Last edit was on August 27, 2018

The current system of One department (support)

A	B	C
The current system of One department (support)	The current system of One partment (oppose)	Need changes to focus more on?
Pros	Cons	
In terms of representation, one department mak s stronger.	I would departn point, it any mo	
If we form departments, we'll create silos, and some areas are not going to fare well.	If we ha special work c not go versus	

Share with others

Link sharing on

Anyone with the link can view

https://docs.google.com/spreadsheets/d/19t_s_zrGloVZXKhVYjSmHD-oNLqGDgYI

People

Enter names or email addresses...

✓ Can edit

Software for collecting and labeling Q statements

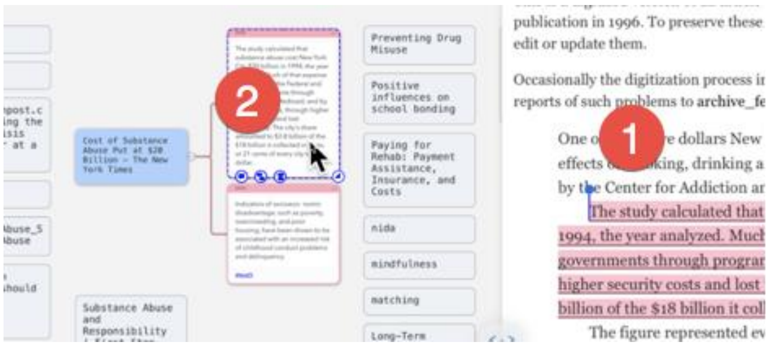
A researcher may use Q statements that have been used in other Q studies or even R-type studies. For example, Stephenson (1953) selected 70 Q statements from a test that was prepared by the College Entrance Examination Board. The original test had more than 200 questions. As an aside, it would be desirable for an organization like ISSSS (the International Society for the Scientific Study of Subjectivity) to have a clearinghouse to store statements used in past Q studies.

Unless a massive archive for Q statements exists, researchers need to find statements on their own that will represent the concepts of a theory they want to test. In the past, researchers often transcribed statements from books or copied and pasted from digital documents, like HTML pages or PDFs.

Researchers are now able to rely on new software for gleaning statements instead of old-style copying and pasting. These software tools allow a user to collect excerpts by just highlighting text on a personal computer or a tablet. The following describes several tools available on multiple platforms.

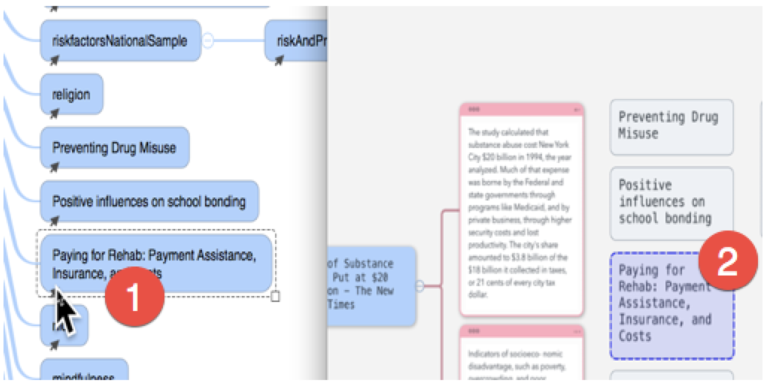
Software for Mac computers. As people go paperless, annotation technology has advanced. For example, macOS or iOS users can use MarginNote 3. A user can highlight a text in a PDF (#1 in *Figure 2*), which can automatically create a card with the excerpt (#2) on a built-in mind map. Specific colors can be selected for highlighting text and coloring card margins, with each color representing a particular theme from the excerpt. Each card can contain a title and comments as well as an excerpt. On a built-in mind map, users can organize these cards hierarchically as siblings or children of other cards typically by dragging and dropping within the mind-map software. If a card is selected on the mind map, the corresponding text on the original document is brought back to the screen, and researchers can easily see the context of any text. The excerpts can span multiple documents, and the excerpts can be images as well as text.

Figure 2. Highlighting text in a PDF document creates a card on a mind map automatically.


2
1
Cost of Substance Abuse Put at $20 Billion - The New York Times
Substance Abuse and Responsibility
Preventing Drug Misuse
Positive influences on school bonding
Paying for Rehab: Payment Assistance, Insurance, and Costs
nida
mindfulness
matching
Long-Term

These excerpts can be exported into other software for further analysis: PDF viewers, iThoughts (a mind-map app), Evernote, MSWord, or DEVONthink (a database app). Below an excerpt appears as a node on iThoughtsX (#1 in *Figure 3*). By clicking the node, a user brings the corresponding card (#2) and original text into focus. MarginNote 3 is an excellent tool for researchers to extract Q statements. The software's built-in mind map or external complementary mind maps can help researchers organize statements by themes.

Figure 3. A node (#1 above) on iThoughtsX and its corresponding mind node card (#2) and text are connected via the address of the objects.


riskfactorsNationalSample
religion
Preventing Drug Misuse
Positive influences on school bonding
Paying for Rehab: Payment Assistance, Insurance,
1
mindfulness
2
Preventing Drug Misuse
Positive influences on school bonding
Paying for Rehab: Payment Assistance, and Costs

As of July 2019, MarginNote 3's academic price is $24, 40% off from the regular $40 price. The prices for MarginNote 3 here and for other software later are subject to change, but they are offered here only for comparison purposes for researchers with limited budgets. LiquidText has similar features to MarginNote, such as the ability to organize cards across documents and export excerpts and notes in PDFs or docx, but LiquidText is only for iOS (LiquidText, n.d.).

Software for PC computers

Many qualitative text analysis programs are available for PC users, and some have developed Mac versions as well. However, these programs can be expensive in comparison to those above. For example, one website ("Top 16", n.d.) introduced 16 quality qualitative programs. The top three NVivo 12 PLUS, ATLAS.ti, MAXQDA are priced for educators at $700, $750, and $495 respectively. Another excellent option, QDA Miner Lite, is offered free from Provalis Research Text Analytics Software, and is ranked 6th on the list from the website above.

As shown in *Figure 4*, QDA Miner Lite displays a list of documents (#1) on the top left corner of the window. If a user clicks a document name, the document will show up in the middle (#2). If a user selects text for tagging (#2) and clicks an appropriate label twice (#3), that label will be attached to the selected text (#4). After tagging all text, a user can retrieve all excerpts by clicking "Coding Retrieval" (#5) under the "Retrieve" button on the main menu.

Figure 4. QDA Miner Lite.

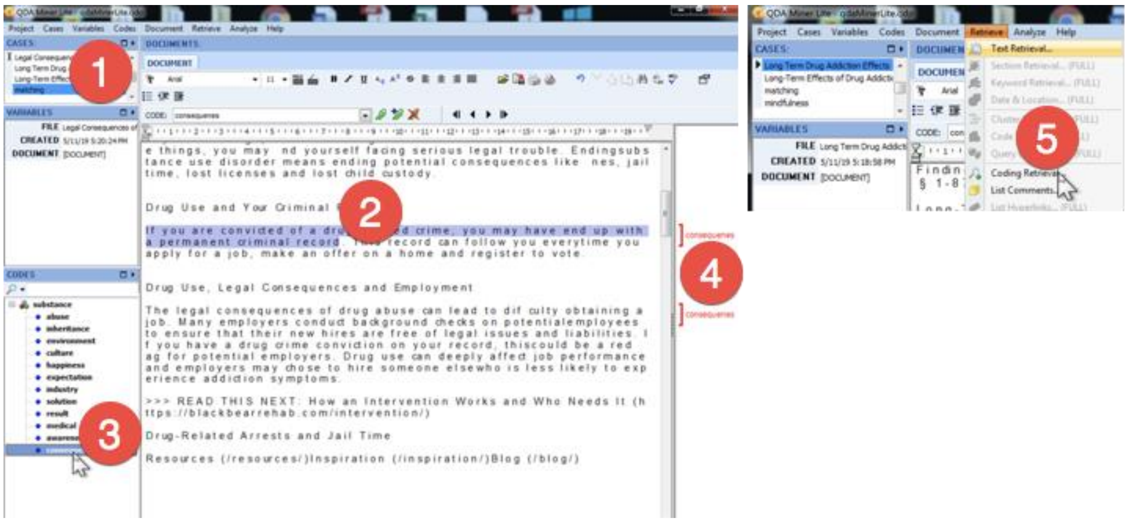
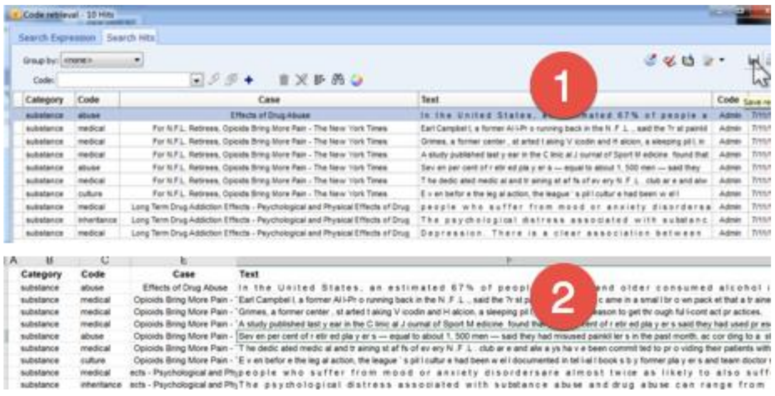


Figure 5 below shows a screen of excerpts in QDA (#1). All excerpts can be exported into an Excel file (#2) for further analysis or just for saving on a local computer. Qualitative text analysis programs can be expensive and often have a steep learning curve. However, Q researchers can use them to quickly excerpt Q statements and tag them with the theme names they create.

Figure 5. Compiled excerpts in QDA Miner Lite can be exported into an Excel file.

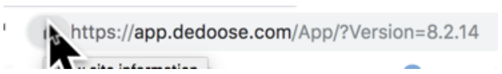


Category	Code	Case	Text
substance	abuse	Effects of Drug Abuse	In the United States, an estimated 67% of peopl [illegible] and older consumed alcohol i
substance	medical	Opioids Bring More Pain -	"Earl Campbell, a former All-Pro running back in the N.F.L., said the ?rst p [illegible] c ame in a small br own pack et that a tr aine
substance	medical	Opioids Bring More Pain -	"Grimes, a former center, started taking Vicodin and Halcion, a sleeping pil [illegible] eason to get thr ough ful l-cont act pr actices.
substance	medical	Opioids Bring More Pain -	"A study published last year in the Clinical Journal of Sport Medicine found th [illegible] ent of r etir ed pla y er s said they had used pr es
substance	abuse	Opioids Bring More Pain -	Sev en per cent of r etir ed pla y er s — equal to about 1, 500 men — said they had misused painkil ler s in the past month, ac cor ding to a st
substance	medical	Opioids Bring More Pain -	"T he dedic ated medic al and tr aining st af fs of ev ery N.F.L. club ar e and alw a ys ha v e been commit ted to pr o viding their patients with
substance	culture	Opioids Bring More Pain -	"E v en befor e the leg al action, the league ' s pil l cultur e had been w el l documented in tel l-al l book s b y former pla y er s and team doctor
substance	medical	ects - Psychological and Phy	people who suffer from mood or anxiety disordersare almost twice as likely to also suff
substance	inheritance	ects - Psychological and Phy	The psychological distress associated with substance abuse and drug abuse can range from

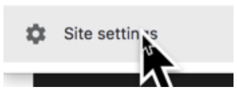
Online software Dedoose. Dedoose allows a user to access its server online via a web browser, or a Mac or PC app. Web browsers like Chrome, Firefox, Internet Explorer, or

Safari can be used as an interface after Flash is installed. Dedoose retails for $14.95 for individual users after a free first-month trial and there is no academic price except for the $10.95 student price. The following shows how to log into Dedoose via a Chrome web browser.

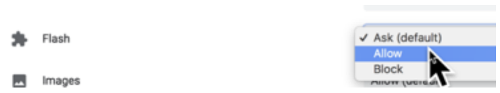
1) Go to "https://app.dedoose.com/App/?Version=8.2.14" and click the Lock icon next to the Address Bar at the top of the browser screen.



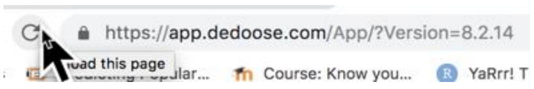
2) Click Site settings.



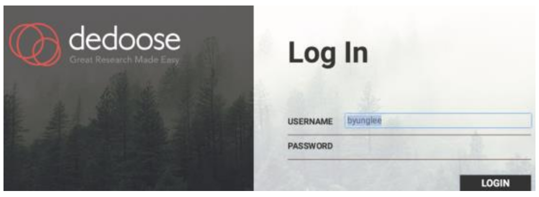
3) In the settings, go to Flash and choose "Allow."



4) Reload the page.

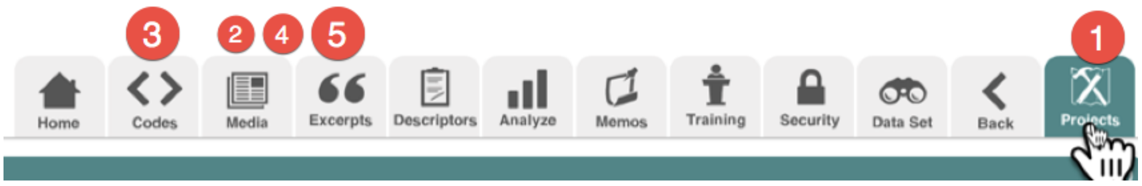


5) Then, the log-in page will be displayed.

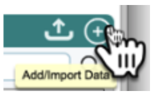


The log-in displays the main menu in *Figure 6* below. Numbers in circles are added to show the steps in which a user uses Dedoose.

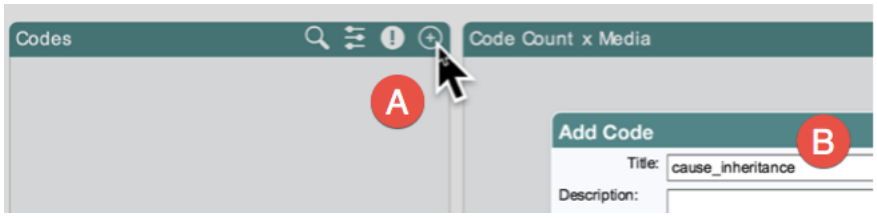
Figure 6. Main menu and steps to take to run Dedoose



1) Click "Projects" above followed by "Create Project" on the bottom right. Type a project name for "Title" and click Submit. This project will contain all documents to import.
2) Click "Media," followed by the "+" button below the main menu to directly write text or import media, such as spreadsheets, text, PDF, images audio, or video.

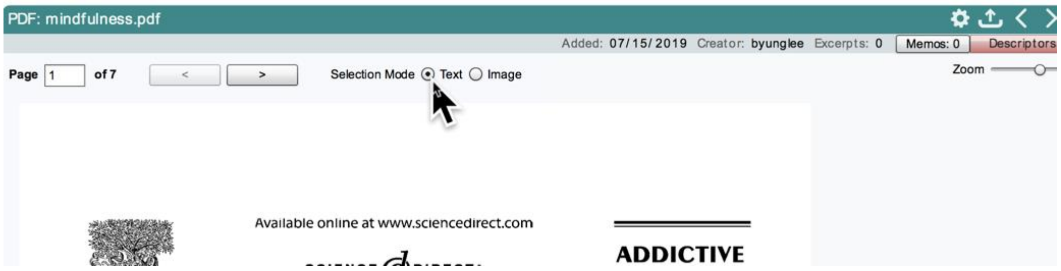


3) Click "Codes" to create theme names. Then click the + button (A) as shown below and write a code (or tag) name in the Title field (B).



4) Click "Media" again to display a list of uploaded documents and click one of those same documents to begin. Choose the text option in *Figure 7* for a PDF document since the image option would only export metadata, not the excerpted text itself.

Figure 7. View a PDF as text or an image while adjusting its zoom level


PDF: mindfulness.pdf

Added: 07/15/2019 Creator: byunglee Excerpts: 0 Memos: 0 Descriptors

Page 1 of 7

Selection Mode [✓ Text] Image

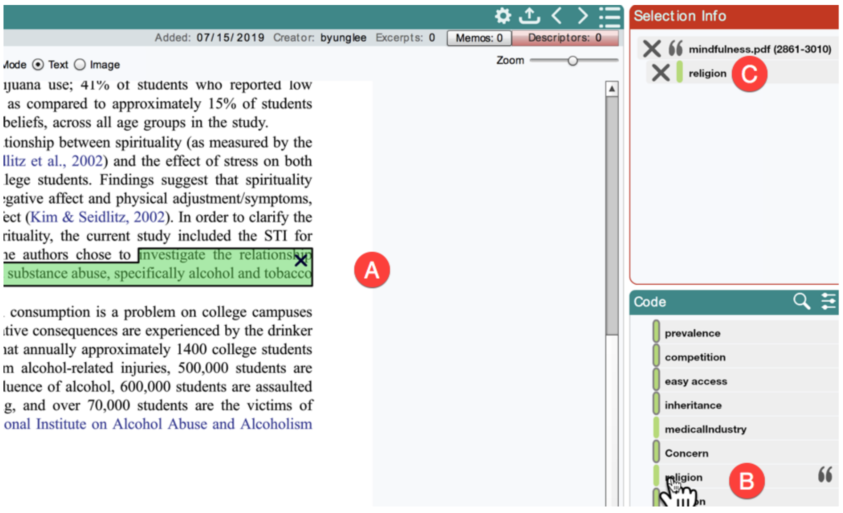
Zoom

Available online at www.sciencedirect.com

ADDICTIVE

Highlight any text to excerpt (A in Figure 8 *below) and click the appropriate code twice on the right (B), then "Selection Info" will verify the coding (C).*

Figure 8. ***To code text, select text and click twice the appropriate theme name.***


Added: 07/15/2019 Creator: byunglee Excerpts: 0 Memos: 0 Descriptors: 0

Mode [✓ Text] Image

Zoom

ijuana use; 41% of students who reported low
as compared to approximately 15% of students
beliefs, across all age groups in the study.
tionship between spirituality (as measured by the
llitz et al., 2002) and the effect of stress on both
lege students. Findings suggest that spirituality
gative affect and physical adjustment/symptoms,
ect (Kim & Seidlitz, 2002). In order to clarify the
rituality, the current study included the STI for
ne authors chose to investigate the relationship
substance abuse, specifically alcohol and tobacco

consumption is a problem on college campuses
tive consequences are experienced by the drinker
nat annually approximately 1400 college students
m alcohol-related injuries, 500,000 students are
luence of alcohol, 600,000 students are assaulted
g, and over 70,000 students are the victims of
onal Institute on Alcohol Abuse and Alcoholism

Selection Info

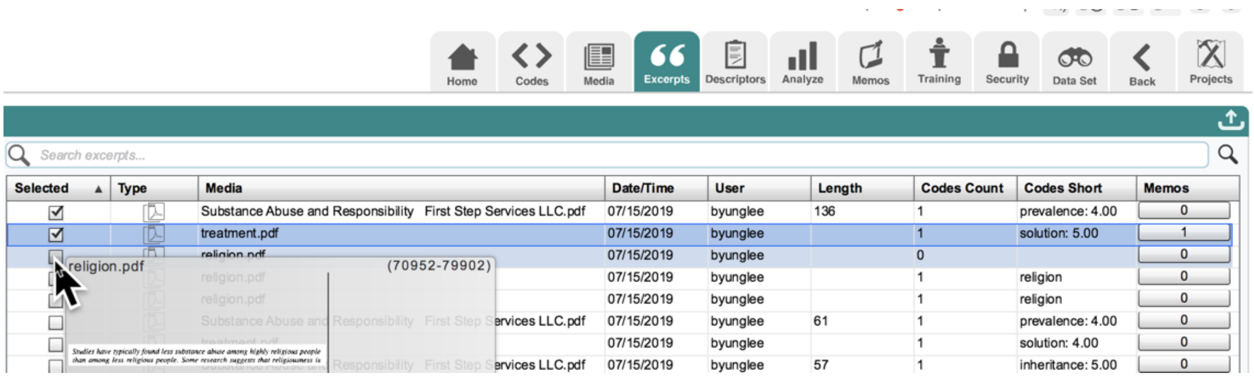
mindfulness.pdf (2861-3010)

religion

5) To see all excerpts, select the Excerpts button on the main menu. To export excerpts, select them, as shown in *Figure 9*. As one moves a cursor over each row, it will show the

content of the corresponding excerpt. Once it is done, click the "Export Excerpt" button to export it into a Word, text, or Excel file.

***Figure 9.* Select any excerpt to export while viewing its content.**



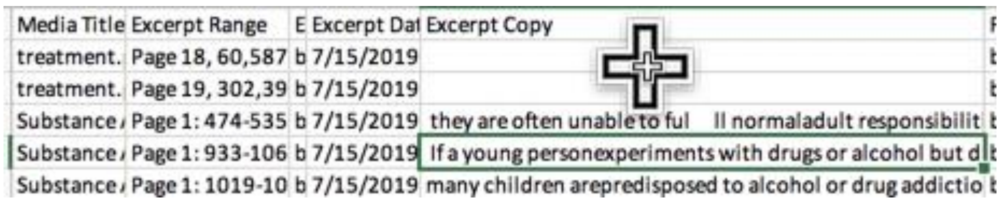
Selected	Type	Media	Date/Time	User	Length	Codes Count	Codes Short	Memos
✓		Substance Abuse and Responsibility First Step Services LLC.pdf	07/15/2019	byunglee	136	1	prevalence: 4.00	0
✓		treatment.pdf	07/15/2019	byunglee		1	solution: 5.00	1
		religion.pdf	07/15/2019	byunglee		0		0
		religion.pdf	07/15/2019	byunglee		1	religion	0
		religion.pdf	07/15/2019	byunglee		1	religion	0
		Substance Abuse and Responsibility First Step Services LLC.pdf	07/15/2019	byunglee	61	1	prevalence: 4.00	0
		treatment.pdf	07/15/2019	byunglee		1	solution: 4.00	0
		Substance Abuse and Responsibility First Step Services LLC.pdf	07/15/2019	byunglee	57	1	inheritance: 5.00	0

religion.pdf (70952-79902)

Studies have typically found less substance abuse among highly religious people than among less religious people. Some research suggests that religiousness is

If it was exported into an Excel file, users can see data, as shown in *Figure 10.* One benefit of Dedoose is that as an online tool it can be used for a Q workshop even for participants who do not have any software for generating a concourse.

Figure 10. The exported Excel file shows the original PDF document name, its page, and excepts.



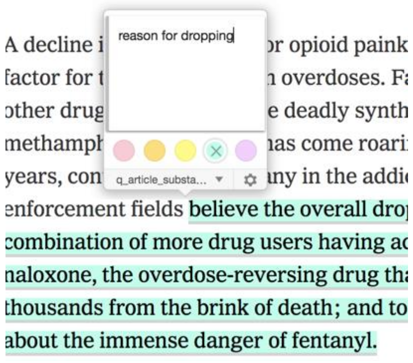
Media Title	Excerpt Range	E	Excerpt Dat	Excerpt Copy	F
treatment.	Page 18, 60,587	b	7/15/2019		t
treatment.	Page 19, 302,39	b	7/15/2019		t
Substance /	Page 1: 474-535	b	7/15/2019	they are often unable to ful ll normaladult responsibilit	t
Substance /	Page 1: 933-106	b	7/15/2019	If a young personexperiments with drugs or alcohol but d	t
Substance /	Page 1: 1019-10	b	7/15/2019	many children arepredisposed to alcohol or drug addictio	t

Web browser extension Weava

Through a Chrome extension, Weava allows a user to highlight and excerpt text from web pages and PDF documents. It is free for basic services, and its premium subscription at $3.99 will enable one to create more than three folders, eliminate a storage limit of 100 MB, use more than five fixed colors for excerpts, share with other collaborators, and clip images ("Weava", n.d.). Many other web annotation tools have come to the market (Singh, 2019). Among them, a researcher may try scribble (http://www.scribble.com) or diigo (http://www.diigo.com), too.
The following shows how to install the Chrome extension on a Mac.
- Go to https://www.weavatools.com and click "Install Chrome Extension," which will take you to the Chrome web store.
- In the store, click "Add to Chrome."
- In the dialog box, click "Add extension."

Now you can use Weava. Go to a web page you want to read and highlight text and choose a highlight color. After clicking a highlighted text, type note as a theme that the passage belongs to, as shown in *Figure 11*. One can turn on/off the highlighter by pressing "Option +S."

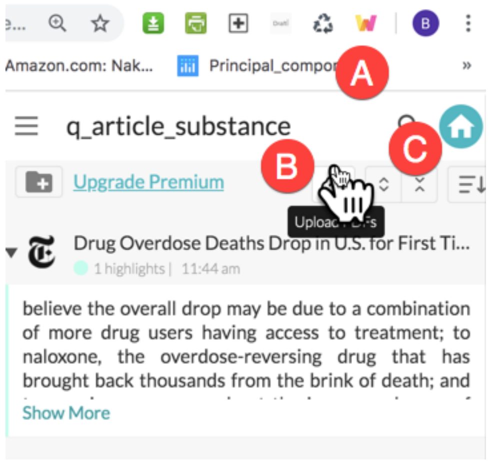
Figure 11. How to select a highlight color and write a note



A decline i ... or opioid paink
factor for t ... n overdoses. F
other drug ... e deadly synth
methampl ... has come roari
years, con ... ny in the addi
enforcement fields believe the overall dro
combination of more drug users having ac
naloxone, the overdose-reversing drug th
thousands from the brink of death; and to
about the immense danger of fentanyl.

Clicking the Weava icon on the web screen (A in *Figure 12* below) displays excerpts.
To code PDFs, upload them from a local computer to the Weava server by clicking "Upload PDFs" (B).

Figure 12. How to upload PDFs



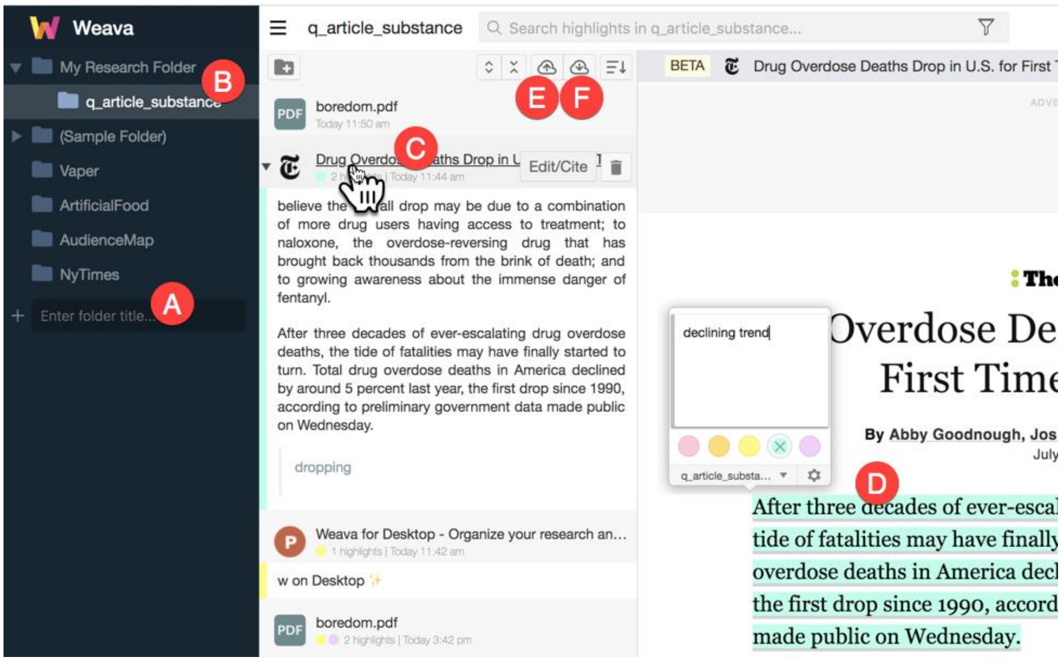
Drug Overdose Deaths Drop in U.S. for First Ti...
1 highlights | 11:44 am

believe the overall drop may be due to a combination of more drug users having access to treatment; to naloxone, the overdose-reversing drug that has brought back thousands from the brink of death; and

Show More

Clicking the Weava icon and "Go to Dashboard" (a house icon C above) displays the dashboard. One can add a new folder by entering a new title (A in *Figure 13*). Clicking a specific folder (B) displays a list of PDFs or webpages that a user highlighted. One can bring up a specific document by clicking on it (C) and continue to highlight or tag (D) on any PDFs or web page. One can import new PDFs (E) or download to a local computer all excerpts (F) in Word, Excel, text, or CSV file formats.

Figure 13. Dashboard of Weava shows folders, files, a document, excerpts, and buttons for importing and exporting files.

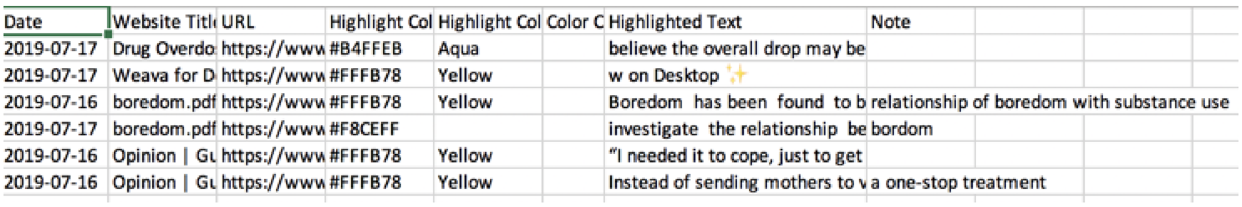


believe the overall drop may be due to a combination of more drug users having access to treatment; to naloxone, the overdose-reversing drug that has brought back thousands from the brink of death; and to growing awareness about the immense danger of fentanyl.

After three decades of ever-escalating drug overdose deaths, the tide of fatalities may have finally started to turn. Total drug overdose deaths in America declined by around 5 percent last year, the first drop since 1990, according to preliminary government data made public on Wednesday.

A sample of an exported Excel file in *Figure 14* below shows the excerpts along with notes, which can be used to represent themes. Weava has many beneficial features for generating Q statements and tagging them, but it only runs on the Chrome Web browser and is not yet available on iOS or Android platforms.

Figure 14. The exported excerpts in Excel



Date	Website Titl	URL	Highlight Col	Highlight Col	Color C	Highlighted Text	Note
2019-07-17	Drug Overdo	https://www	#B4FFEB	Aqua		believe the overall drop may be	
2019-07-17	Weava for D	https://www	#FFFB78	Yellow		w on Desktop ✨	
2019-07-16	boredom.pdf	https://www	#FFFB78	Yellow		Boredom has been found to b	relationship of boredom with substance use
2019-07-17	boredom.pdf	https://www	#F8CEFF			investigate the relationship be	bordom
2019-07-16	Opinion | Gu	https://www	#FFFB78	Yellow		"I needed it to cope, just to get	
2019-07-16	Opinion | Gu	https://www	#FFFB78	Yellow		Instead of sending mothers to v	a one-stop treatment

Text-mining algorithms as a search tool for Q statements

Q researchers can take advantage of text-mining algorithms for the generation of Q statements. One such algorithm is LDA (Latent Dirichlet Allocation), which discovers themes from multiple documents. The meaning of themes can be guessed by a certain number of words that represent each theme. A user is required to fix the number of words beforehand. Another algorithmic tool is R's widyr package, which shows how a pair of words co-occur in a passage.

LDA can be used in Q methodology in much the same way as factor analysis. Research participants' Q sorts are summarized into latent factors or types of people, whose natures are interpreted based on the sorting pattern of the representative factor sorter. Similarly, LDA discovers the hidden thematic structure in vast archives of documents based on the probability or frequency of words to be used in each topic. Each topic is associated with a mixture of words (Blei, n.d.; Silge & Robinson, 2017). The number of themes, however, should be prefixed by the researcher.

The following describes how to derive three themes out of 41 PDF documents and find words that are associated with each theme. The following code for LDA is adapted from Silge and Robinson's (2017) book, especially Chapter 6, Topic Modeling .

Import the following R libraries (they can be downloaded online with the command, install.packages(<library package name>)). The "<library package name>" should be replaced by each of the package names below, for example,
install.packages("topicmodels").

```
1.  library(topicmodels)
2.  library(tidyverse)
3.  library(tidyr)
4.  library(tidytext)
5.  library(pdftools)
6.
```

The following shows how to get a list of PDF files.

- Line 7 creates a "dir" var and stores the path to a folder that contains PDF documents. R can import other types of documents, but this article shows only how to import PDFs since they are more complex than other file types.
- Line 9 gets a list of all files under the folder.
- Line 10 shows the number of files, which confirms the actual downloading of the files.
- Line 12 only selects files whose name has a ".pdf" extension.
- Line 13 checks the number of PDF files to see whether the code is working.

```
7.  dir <- "<the path to the data folder>"
8.
9.  filelist <- list.files(path = dir)
10. length(filelist)
11.
12. pdf_list <- filelist[grepl(".pdf",filelist)]
13. length(pdf_list)
```

The following shows how to create a dataframe, a data type to be used to compile text from PDFs.

- Line 14 creates an empty dataframe, df, to compile text from all PDFs later.
- Line 15 names the first column of the df above as "doc."
- Line 16 of dim(df) prints the dimension of the dataframe, 0 row by 1 column.

```
14. df <- tibble(content = character())
15. names(df) <- "doc"
16. dim(df)
17.
```

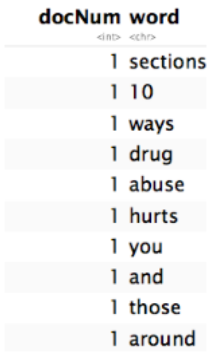
The following shows how to compile text from PDFs and add a column for a document number.

- Line 18 empties the df dataframe in case.
- Line 19: the for-loop processes each of the PDF file name one at a time.
- Line 21 imports each page of text into a text element in a vector.
- Line 22 collapses multiple text elements into one.
- Line 23 replaces characters of "\n" with one space so that the following paragraph is attached to the end of the previous paragraph. "\n" means a new paragraph.
- Line 24 replaces multiple spaces with one.
- Line 25 converts a text into a tibble dataframe, which can be processed more easily in R than a traditional data frame.
- Line 27 adds this new dataframe to the original df. At the end of the loop, all PDF files will be combined into df.
- Line 30 creates "docNum" to track each line of text that comes from different PDFs. The number starts at 1 and ends with the total number of documents.

```
18. df <- df[0, ]
19. for (file in 1:length(pdf_list))
20. {
21.   doc <- pdf_text(file.path(dir, pdf_list[file])) ### doc character vectors
22.   text <- paste(doc, collapse = '\n') ### collapse multiple vector item into one
23.   text <- str_replace_all(text, '\n', " ") ### convert "\n" to one space
24.   text <- str_replace_all(text, '\\s+', " ") ### convert "\n" to one space
25.   docDF <- tibble(text )  ## text is the column name
26.
27.   df <- rbind(df, docDF)
28. }
29.
30. df["docNum"] <- 1:nrow(df)
```

The following shows how to split each document into individual words and its outcome. In LDA the unit of observation is individual words.

```
31. by_doc_word <- df %>%
32. unnest_tokens(word, text)
33. by_doc_word
```



```
docNum word
 <int> <chr>
     1 sections
     1 10
     1 ways
     1 drug
     1 abuse
     1 hurts
     1 you
     1 and
     1 those
     1 around
```

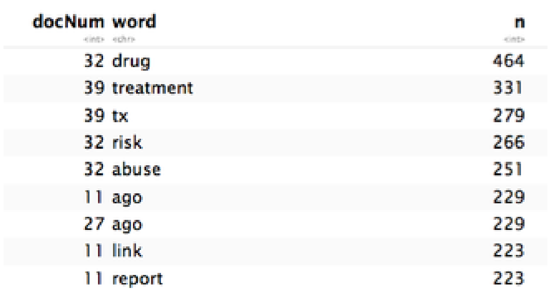
A user can create a list of stop words that should be filtered out before processing text since the default stop words in R are not sufficient. "1:3000" in line 34 below means a whole number starting at 1 and ending at 3,000, while "letters" means 26 individual English letters. This list is converted into a dataframe, called stopWordsDF, which has one column, whose name is "stopWords" (line 36).

```
34. stopWords <-c(1:3000, letters,   "about", "above", "after", "again", "against",
    "ago", "al", "all", "also", "am", "among", "an", "and", "any", "are", "aren't", "as",
    "at", "be", "because", "been", "before", "being", "below", "between", "both", "but",
    "by", "can", "can't", "cannot", "could", "couldn't", "did", "didn't", "do", "does",
    "doesn't", "doing", "don't", "down", "during", "each", "et", "even", "found", "few",
    "for", "from", "further", "get", "had", "hadn't", "has", "hasn't", "have", "haven't",
    "having", "he", "he'd", "he'll", "he's", "her", "here", "here's", "hers", "herself",
    "him", "himself", "his", "hours", "how", "how's", "https", "i", "i'd", "i'll", "i'm",
    "i've", "if", "in", "into", "is", "isn't", "it", "it's", "its", "itself", "just", "let's", "many",
    "may", "me", "might", "mo", "more", "most", "mustn't", "my", "myself", "no", "nor",
    "not", "of", "often", "off", "on", "one", "once", "only", "or", "other", "ought", "our",
    "ours", "ourselves", "out", "over", "own", "s", "same", "say", "sections", "shan't",
    "she", "she'd", "she'll", "she's", "should", "shouldn't", "so", "some", "such", "than",
    "that", "that's", "the", "their", "theirs", "them", "themselves", "then", "there",
    "there's", "these", "they", "they'd", "they'll", "they're", "they've", "this", "those",
    "through", "to", "too", "tx", "under", "until", "up", "very", "was", "wasn't", "ways",
    "we", "we'd", "we'll", "we're", "we've", "well", "were", "weren't", "what", "what's",
    "when", "when's", "where", "where's", "which", "while", "who", "who's", "whom",
    "why", "why's", "will", "with", "won't", "would", "wouldn't", "you", "you'd",
    "you'll", "you're", "you've", "your", "yours", "yourself", "yourselves")
35.
36. stopWordsDF <- tibble(stopWords)
```

The following eliminates the stop words from df (line 38), counts the frequency of each of the remaining words, and sorts them by their frequency (line 39). Line 41 prints the outcome of the newly created dataframe, word_counts.

```
37.  word_counts <- by_doc_word %>%
38.  anti_join(stopWordsDF, by = c("word" = "stopWords") ) %>%
39.  count(docNum, word, sort = TRUE)
40.
41.  word_counts
```

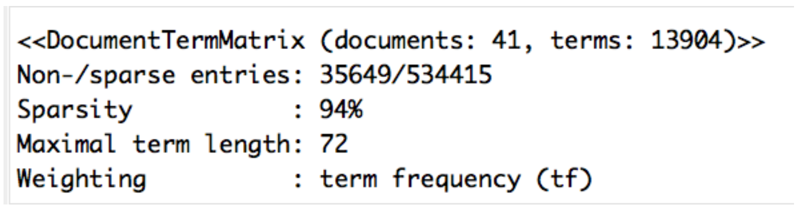
After viewing the list, one can find additional stop words to be eliminated from df. In this case, add these words to the stop word list (line 34 above) and execute lines 34-41 again.



docNum	word	n
<int>	<chr>	<int>
32	drug	464
39	treatment	331
39	tx	279
32	risk	266
32	abuse	251
11	ago	229
27	ago	229
11	link	223
11	report	223

Lines 42-44 convert word_counts into a document-term-matrix, which has 41 documents in rows and 13,904 terms in columns.

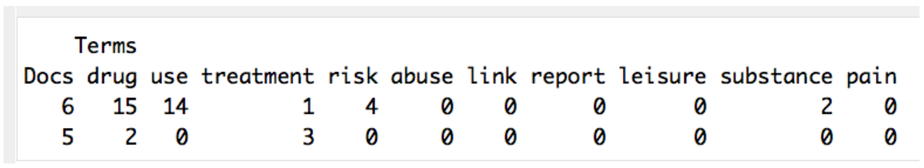
```
42. doc_dtm <- word_counts %>%
43. cast_dtm(docNum, word, n)
44. doc_dtm
45.
```



```
<<DocumentTermMatrix (documents: 41, terms: 13904)>>
Non-/sparse entries: 35649/534415
Sparsity           : 94%
Maximal term length: 72
Weighting          : term frequency (tf)
```

To see what the document looks like, one can temporarily convert doc_tem_m into a matrix (line 46). Line 47 is used to display only 39th and 40th rows and first 10 columns because of the limited computer screen for display. Each cell shows the frequency of each word used in each document.

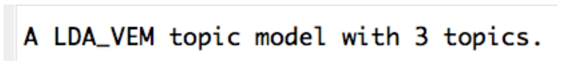
```
46. doc_dtm_m <- as.matrix(doc_dtm)
47. doc_dtm_m[39:40, 1:10]
```



```
    Terms
Docs drug use treatment risk abuse link report leisure substance pain
   6   15  14         1    4     0    0      0       0         2    0
   5    2   0         3    0     0    0      0       0         0    0
```

The number of topics should be specified before applying LDA model. So, k = 3 is chosen in line 48. The seed number is set so that the simulation can be reproduced exactly the same whenever it is executed.

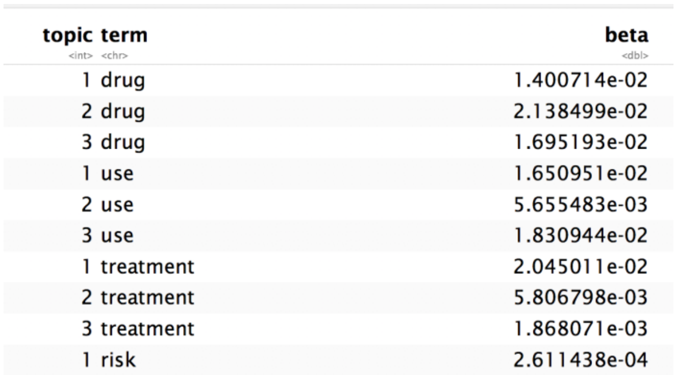
```
48. docs_lda3 <- LDA(doc_dtm, k = 3, control = list(seed=1234))
49. docs_lda3
```



```
A LDA_VEM topic model with 3 topics.
```

Line 50-51 produce three topics and the probability of each term to be generated from each topic (beta numbers below).

```
50. docs_topics3 <- tidy(docs_lda3, matrix = "beta")
51. docs_topics3
```

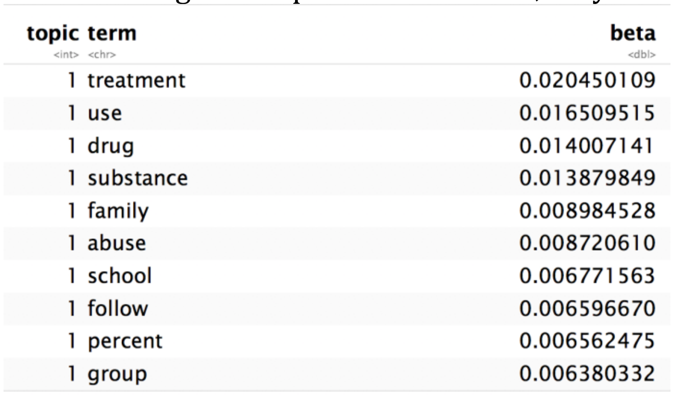


topic <int>	term <chr>	beta <dbl>
1	drug	1.400714e-02
2	drug	2.138499e-02
3	drug	1.695193e-02
1	use	1.650951e-02
2	use	5.655483e-03
3	use	1.830944e-02
1	treatment	2.045011e-02
2	treatment	5.806798e-03
3	treatment	1.868071e-03
1	risk	2.611438e-04

Lines 52-57 divide terms (or words) by topic (or theme) and arrange them by the level of probabilities. Only the top 10 terms are chosen from each of the three topics.

```
52. top_terms3 <- docs_topics3 %>%
53. group_by(topic) %>%
54. top_n(10, beta) %>%
55. ungroup() %>%
56. arrange(topic, -beta)
57. top_terms3
```

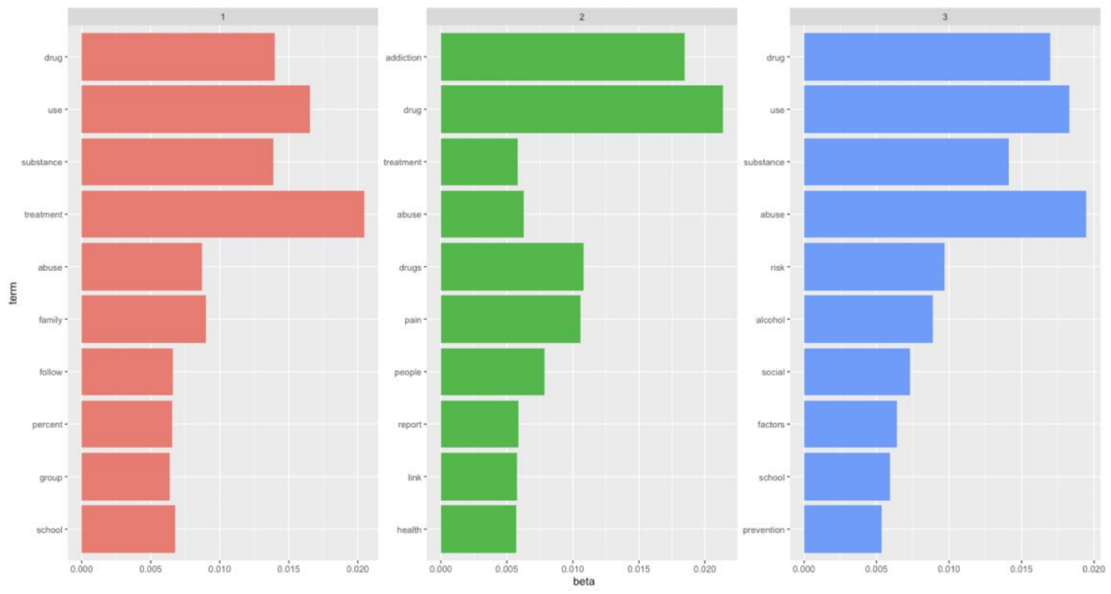
The following shows part of the results, only the first 10 terms from topic 1.



topic <int>	term <chr>	beta <dbl>
1	treatment	0.020450109
1	use	0.016509515
1	drug	0.014007141
1	substance	0.013879849
1	family	0.008984528
1	abuse	0.008720610
1	school	0.006771563
1	follow	0.006596670
1	percent	0.006562475
1	group	0.006380332

Lines 58-64 create three charts, in which 10 terms are displayed for each topic.

```
58. library(ggplot2)
59. top_terms3 %>%
60. mutate(term = reorder(term, beta)) %>%
ggplot(aes(term, beta, fill = factor(topic))) +
geom_col(show.legend = FALSE) +
facet_wrap(~topic, scales ="free") +
coord_flip()
```



If these terms do not cover enough concepts for the researcher, topic numbers can be increased to, say, seven. The following shows 70 top terms, 10 from each of seven topics.

```
$term
 [1] "drug"             "abuse"                        "substance"
 [4] "family"           "www.altamirarecovery.com"     "use"
 [7] "addiction"        "treatment"                    "health"
[10] "drugs"            "drug"                         "school"
[13] "program"          "use"                          "addiction"
[16] "alcohol"          "bonding"                      "treatment"
[19] "effects"          "intervention"                 "drug"
[22] "abuse"            "use"                          "risk"
[25] "substance"        "alcohol"                      "social"
[28] "factors"          "school"                       "prevention"
[31] "use"              "leisure"                      "substance"
[34] "drug"             "percent"                      "boredom"
[37] "students"         "treatment"                    "self"
[40] "drugabuse.com"    "pain"                         "report"
[43] "link"             "people"                       "opioids"
[46] "doctors"          "drug"                         "opioid"
[49] "drugs"            "addiction"                    "addiction"
[52] "drug"             "abuse"                        "drugs"
[55] "recovery"         "substance"                    "help"
[58] "alcohol"          "treatment"                    "use"
[61] "treatment"        "drug"                         "use"
[64] "substance"        "abuse"                        "family"
```

```
[67] "follow"                    "group"                                 "studies"
[70] "adolescent"
```

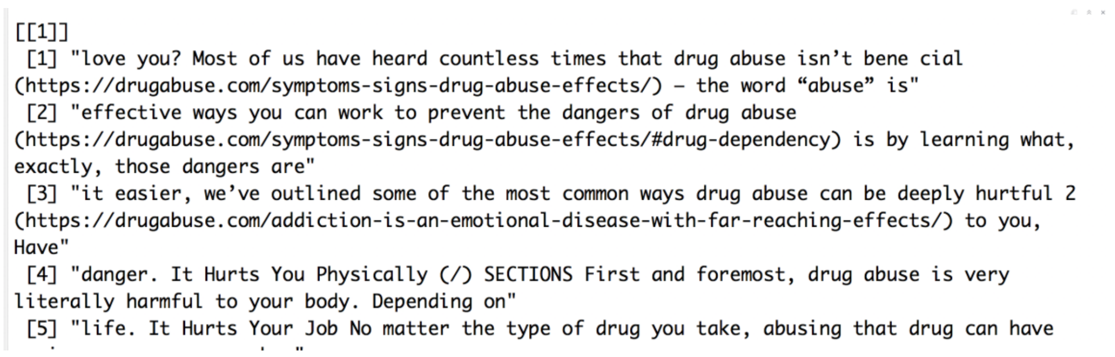
When duplicates and HTTP addresses were eliminated, only the following were left as meaningful words.

"abuse", "addiction", "adolescent", "alcohol", "bonding", "boredom", "doctors", "drug", "drugs", "effects", "factors", "family", "follow", "group", "health", "help", "intervention", "leisure", "link", "opioid", "opioids", "pain", "people", "percent", "prevention", "program", "recovery", "report", "risk", "school", "self", "social", "students", "studies", "substance", "treatment", "use"

When Q researchers deal with big text data, they cannot read it all, but they can analyze the text and keywords associated with each topic and use them to extract the text that contains these words. For example, the words above are associated with seven themes. The following code extracts passages that contain 10 words before the word “drug” and another 10 words thereafter. Q researchers can try other words instead of “drug”, such as “intervention” and/or “prevention” to find statements that may deal with how to stop substance use.

```
65. str_extract_all(df$text, "(?:[^\\s]+\\s){10}drug(?:\\s[^\\s]+){10}")
```

The following shows the original text, which contains "drug" in it.



```
[[1]]
 [1] "love you? Most of us have heard countless times that drug abuse isn’t bene cial
(https://drugabuse.com/symptoms-signs-drug-abuse-effects/) – the word “abuse” is"
 [2] "effective ways you can work to prevent the dangers of drug abuse
(https://drugabuse.com/symptoms-signs-drug-abuse-effects/#drug-dependency) is by learning what,
exactly, those dangers are"
 [3] "it easier, we’ve outlined some of the most common ways drug abuse can be deeply hurtful 2
(https://drugabuse.com/addiction-is-an-emotional-disease-with-far-reaching-effects/) to you,
Have"
 [4] "danger. It Hurts You Physically (/) SECTIONS First and foremost, drug abuse is very
literally harmful to your body. Depending on"
 [5] "life. It Hurts Your Job No matter the type of drug you take, abusing that drug can have
```

The next shows how to use the widyr package in R to understand a network of words used in each text passage. It tracks pairs of words used in text with pairwise_count( ). Since one subjective idea can be spread across multiple sentences, I used three consecutive sentences as a unit of observation. Thus, the final graph will show the frequency of pairs of words that co-occur in the three successive sentences. The following code for counting a pair of words in each observation unit of text is adapted from Silge and Robinson's book, specifically the Word Co-occurrences and Correlations section (2017, pp. 130-133).

The code from the LDA analysis earlier, from lines 1-28, is used again for this analysis to compile text from the 41 PDF documents. The following splits each document into individual sentences.

- Lines 39-40 tokenize each row of document text into individual sentences and save it as a dataframe, "tokenized." The verb, "tokenize," means splitting each document into tokens, like a word, words, or a sentence.
- Line 42 creates a column, named "id," and fills a line number.
- Line 43 checks the number of lines and columns.

```
36.
37.
38.
39. tokenized <- df %>%
40. unnest_tokens(sentence, text, token  = "sentences")
41.
42. tokenized["id"] <- 1:nrow(tokenized)
43. dim(tokenized)
```

The following describes the process of how to combine each triple of consecutive sentences into a portion of text for analysis. The dataframe, tokenized, is converted into three dataframes.

- Line 44 creates a variable n to represent the total number of rows.
- Line 45 drops the last two rows from the tokenized dataframe and keeps the rest as token1.
- Line 47 drops the first and last lines from tokenized and saves only the sentence column as token2.
- Line 48 assigns a new column name, "sentence2," to prevent duplication between token1 and token2.
- Line 50 drops the first two lines from tokenized and keeps the sentence column as token3.
- Line 51 assigns "sentences3" to the sentence column to prevent duplication between token1 and token3.

Combining three sentences into one unit of observation should be done for each document. The above was done with the entire dataframe rather than within each document, which shortens the processing time. The sheer number of lines for each document would make the impact of this adjustment tiny.

```
44. n <- nrow(tokenized)
45. token1 <- tokenized[1:(n-2), c("id", "sentence")]
46.
47. token2 <-  tokenized[2:(n-1), 1 ]
48. names(token2) <- "sentence2"
49.
50. token3 <- tokenized[3:n, 1]
51. names(token3) <- "sentence3"
```

The following shows how to combine the three dataframes above into one big dataset for further analysis.

- Line 52 horizontally combines three dataframes into dfComb.
- Line 53 combines three text columns in dfComb and assigns the result to the new dfComb's column, "sentences."
- Line 54 keeps only two columns from dfComb: "id" and "sentences."

```
52. dfComb <- cbind(token1, token2, token3)
53. dfComb$sentences = with(dfComb, paste(sentence, sentence2, sentence3))
54. dfComb <- dfComb[, c("id", "sentences")]
```

Lines 55-56 split sentences, three consecutive sentences combined, into individual words and keep the result as the by_words dataframe.

```
55. by_words <- dfComb %>%
56. unnest_tokens(word, sentences , token  = "words")
```

Lines 57-58 check the "word" column of by_words and filter out the stop words of the stopWordsDf. After all stop words are deleted, the remnants are saved as purified_words.

```
57. purified_words <- by_words %>%
58. anti_join(stopWordsDF, by = c("word" = "stopWords"))
```

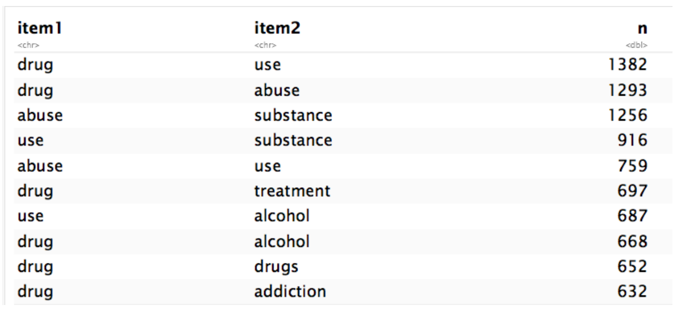
The following loads the widyr library, which calculates the frequency of words that co-occur in each passage.

- Line 59 loads widyr. If it has not been installed on a local computer, it should be installed with a command, install.packages("widyr").
- Lines 60-61 calculate which pair of words were used together in each line and sort them based on their co-occurrence frequency.

```
59. library(widyr)
60. word_pairs <- purified_words %>%
61. pairwise_count(word, id, sort = TRUE, upper = FALSE)
```

- Lines 62-63 select the pairs of words whose frequency is 30 or higher and save them pairs_n30
- Line 64 displays the outcome, part of which is a table shown below.

```
62. pairs_n30 <- word_pairs %>%
63. filter (n>= 30)
64. pairs_n30
```



item1	item2	n
<chr>	<chr>	<dbl>
drug	use	1382
drug	abuse	1293
abuse	substance	1256
use	substance	916
abuse	use	759
drug	treatment	697
use	alcohol	687
drug	alcohol	668
drug	drugs	652
drug	addiction	632

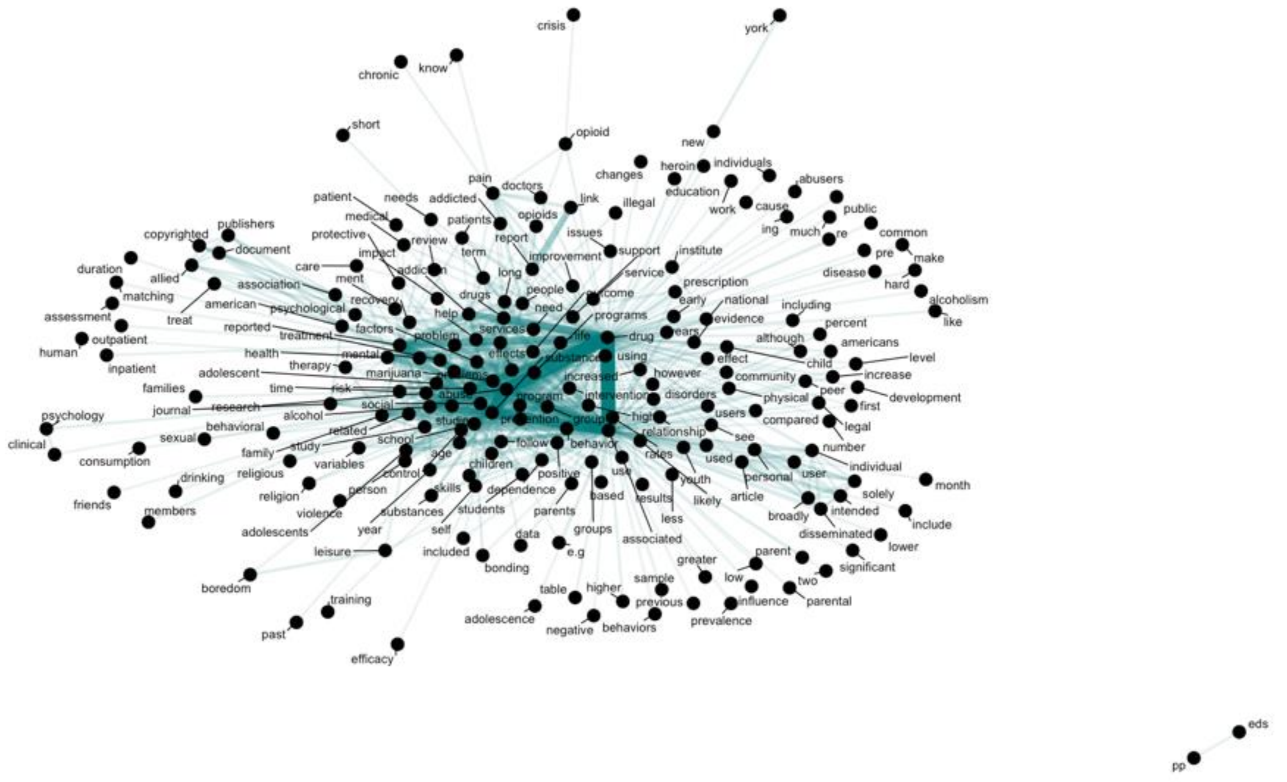
By looking at the pattern of how each word is used with other words above, Q researchers may get a clue about different aspects of drug issues: use and abuse of drugs; addiction to drugs; whether alcohol is considered a drug; and the cause of drug problems as well as the treatment of addicts. For a better picture of the pattern between word relationships, the co-occurrence frequencies can be converted into a graph.

Below the outcome with n > 100 was converted into a graph.

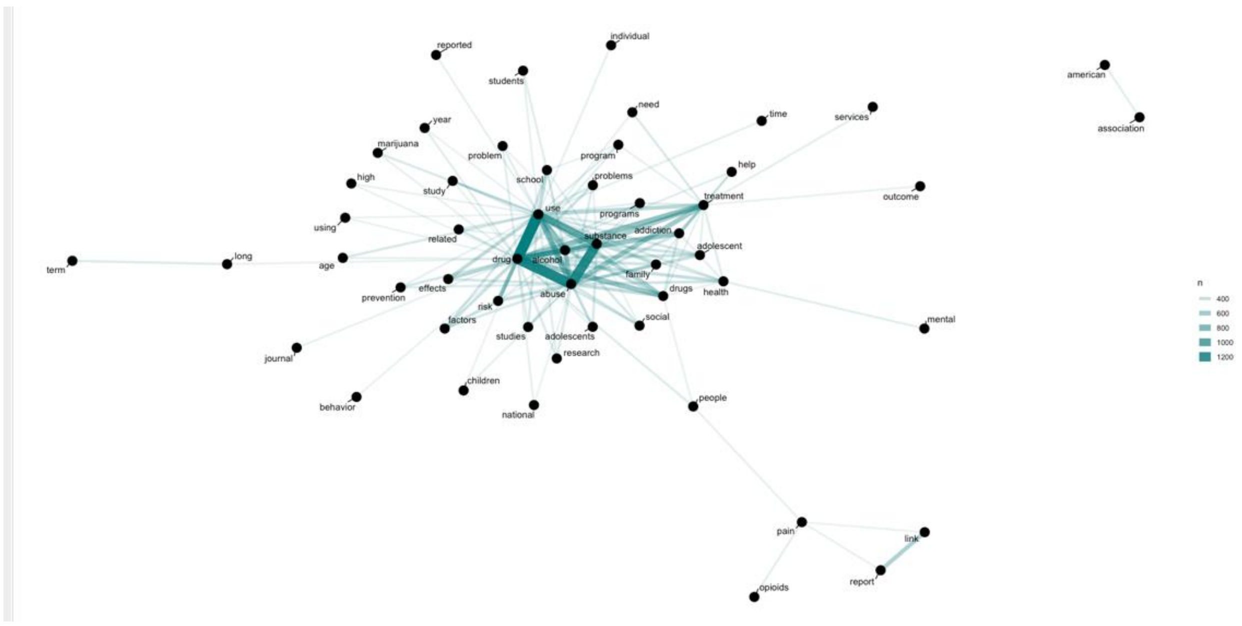
- Lines 66-73 create a graph.
- Line 67 selects pairs only when their co-occurrence frequency is greater than 100.
- Line 69 selects a type of layout, called "fr."
- Lines 70-72 describe many options for the attributes of a graph like the size of nodes and edges and their color and transparency level, which are detailed in the ggraph package manual (Pedersen, 2018).
- Line 73 is used to eliminate the x- and y-axis and gridlines, which are not needed for a clean graph.

```
65. set.seed(1234)
66. word_pairs %>%
67. filter(n > 100) %>%
68. graph_from_data_frame() %>%
69. ggraph(layout = "fr") +
70. geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_color = "cyan4") +
71. geom_node_point(size = 5) +
72. geom_node_text(aes(label = name), repel = TRUE, point.padding = unit(0.2,
    "lines")) +
73. theme_void()
```

The outcome is a network of paired words below. Each word is represented by the same size of node since geom_node_point is set to 5 above. The co-occurrence frequency of two words is represented by the width and color opacity of the edge, which is the line between nodes. Above the edge_alpha and edge_width are set to n, the frequency, so their opacity and thickness will vary with the corresponding frequency. The algorithm to distribute nodes and edges is "fr," which is one of frequently used algorithms for drawing network graphs.



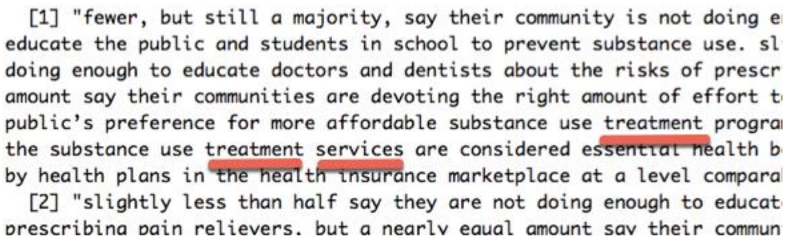
If the graph seems overwhelming, one can raise the n to 200 so that the graph shows a fewer number of word pairs, as shown below. By looking at the thickness of edges and color opacity, one can see the co-occurrence frequency of any two words.



If researchers look for statements on substance treatment services, they can try the following code, which will display text lines that contain in the original document "treatment" and "services."

Lines 74-75 select a pair of words from the graph above.
Line 76 checks whether each passage contains the pair of words.
Line 78 displays text when it contains both.

```
74. word1 <- "treatment"
75. word2 <- "services"
76. findElements <- grepl("(?i)(treatment.*services|services.*treatment)",
    dfComb$sentences)
77. dfComb$sentences[findElements]
```



```
 [1] "fewer, but still a majority, say their community is not doing e
educate the public and students in school to prevent substance use. sl
doing enough to educate doctors and dentists about the risks of prescr
amount say their communities are devoting the right amount of effort t
public's preference for more affordable substance use treatment progra
the substance use treatment services are considered essential health b
by health plans in the health insurance marketplace at a level compara
 [2] "slightly less than half say they are not doing enough to educat
prescribing pain relievers, but a nearly equal amount say their commun
```

Finding Themes and Selecting Statements for a Q Sample

While researchers select a Q sample out of a concourse, they often take different approaches, and additionally Q scholars have differed in their use of terms with these approaches. Thus, I provide the three new terms for clarification: a structural, a semi-structural, and an unstructured design, as described below.

In designing experiments, a researcher needs to identify each main element to measure its effect on the responses and the interaction effects if any combination of elements acts together (Hardwick, 2013). Some scholars used "variables" or "factors" instead. Different categories in each element can be called treatments, conditions, or levels (Borg & Gall, 1979, p. 569; Kerlinger & Lee 2000, p. 346). In statistics, a full factorial experiment is an experiment whose design consists of multiple elements, each with discrete levels. Stephenson showed full factorial examples throughout his works, while using the term, such as "effects" and "independencies"[1] instead of independent variables or factors. For example, Stephenson (1952) used three independencies, Attitudes, Mechanisms, and Functions with their corresponding 2, 2, 4 levels regarding Jung's type psychology. He also used three independencies, Control, Adjustment, and Erlebnis type[2], with their corresponding 3, 3, and 2 levels regarding Rorschach's theory (p 489). He did not treat the direction of statements, like the positive and negative nature of statements, as one of the effects.

Q researchers usually prefer a structural design since it controls experimental errors better than an unstructured design. "Experimental error occurs when a change in the DV is produced by any variable other than the IV" (Burns & Dobson, 1981, p. 143). To the same effect, Stephenson (1953) wrote,"[W]here *error* is to be expected, it is specified in a structured sample and not in an unstructured one (p. 74). Stephenson

[1] Brown (1980) used the term "Main Effects" instead.
[2] Stephenson listed two levels of "introvertive" and "extrovertive" under Erlebnis, which means "experience" in German.

(1953) wrote that a structured sample is taken when a factorial design is available. If not, “the sample should be balanced with respect to at least one effect” (p.78). Surely, he meant a multiple factorial design without an interaction among various elements when he referred to a factorial design. So, he treated a Q sample with one factor as an unstructured sample. He also added that a Q sample is “never selected purely randomly from the universe” because the entire sample should be balanced with positive and negative statements, at a minimum of two levels (p. 73, 78-79).

On the other hand, McKeown and Thomas (1988) described two types of structured Q samples: a deductively designed sample based on “a priori hypothetical or theoretical considerations” (p. 28) and an inductively designed one that would “emerge from the patterns that are observed as statements are collected” (p. 28-29). They put these two designs under structured sampling, despite the difference in their theoretical elaboration. Regarding the direction of statements, they considered it as one of the effects (Refer to *Figure 1.1* in their book *Q Methodology*).

Mrtek, a former *Operant Subjectivity* editor, co-authored a Q study (Barbosa, Willoughby, Rosenberg, & Mrtek, 1998), where the structured and unstructured design were defined differently. The study called a Q sample “structured” when it had one or more dimensions and discussed the unstructured approach as follows: researchers “assess the concourse, choose a number of items that cover the known aspects of the topic, and occasionally add some that may not have surfaced in the assessment of the concourse” (p. 1034). They classified an inductive designed sample as “unstructured”.

So, for the purposes of this study I will define a Q sample as a structured one when it takes a deductive approach, theory-driven; a Q sample with emerging themes based on the assessment of a concourse of statements as semi-structured, since this inductive approach is exploratory and less systematic in describing effect levels; and a Q sample with only the direction effect as an unstructured one.

Selecting a Q sample out of a concourse is not easy because the sample should be representative of the concourse. To acquire a representative Q sample, Watts and Stenner (2012) recommended that Q researchers work on Q sample design with “great persistence and very high levels of skill” (p. 58). They suggested, a Q sample should capture “the full gamut of possible opinion and perspective in relation to your research question” (p. 58). Barbosa et al. (1998, p. 1035) echoed this same sentiment, “The Q sample must represent all the major ideas, viewpoints, feelings, and opinions in the concourse.” This is one of the fundamental tenets of Q methodology that was stressed by Stephenson (1953) and Brown (1980).

In the structured approach, Q researchers have to find statements that reflect defined effects and positive and negative levels. In a semi-structured approach, Q researchers have to find themes or aspects of their research topic while assessing concourse statements gleaned. While searching for statements for a Q sample, researchers can move them around with software like Microsoft Word, organizing individual statements hierarchically under Draft View. Or they may take advantage of the mind-map feature of MarginNote or LiquidText in the tools discussed earlier. But a more efficient way is to use software that is designed specifically for mind mapping.

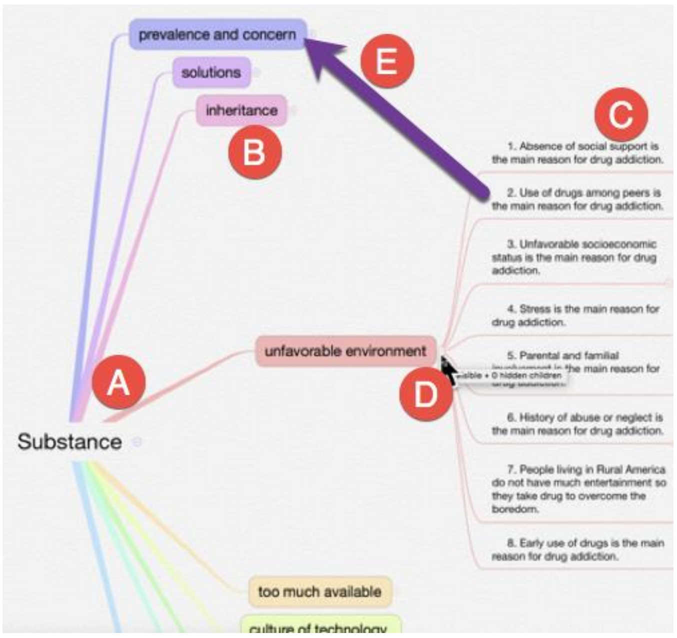
There are many mind-mapping tools for all platforms (Myre, 2019), including free ones. Since Q researchers have to handle hundreds of statements, they have to choose a tool that can efficiently organize these many statements. A mind map, on which researchers usually keep one statement on each node, often offers a smooth movement of a node across categories without any delay. Its node-folding feature enables researchers to hide all nodes except for ones they assess at a specific time.

While using the mind map, Q researchers need to work on one theme at a time. They need to collapse all themes except for one they are focused on at that time. They should also calculate the number of statements for each theme. For example, if the researchers have five themes for 40 statements, they can have eight statements for each theme. They can then look at all statements under one theme and choose eight statements among them while considering the positive or negative direction of each statement. When similar statements are found, and one is more appealing than another, researchers can keep the former and convert the latter as a child of the former. A less obtrusive way is to put the latter in the notes section of the former. If some statements are found to be wrongly classified, they can be easily dragged and dropped under other themes.

Among many, this study focuses on iThoughtsX and MindNode 6 as mind-mapping applications. iThoughtsX can import statements from rows in a CSV file as nodes ("TokelaWare," n.d.). One can use the root node for the topic (A in *Figure 15*), child nodes for themes (B), and grandchild nodes for statements (C). Theme nodes can collapse or expand statements when a circle attached to the theme node is clicked (D). To focus one node under analysis, one can collapse other nodes. A statement can be reclassified under a different category by dragging and dropping (E). To export specific statements, one can copy selected statements and paste them into Excel. Educators are given a 50 percent discount off $49.99, the full price for its Mac and PC versions at https://www.toketaware.com/store. iThoughtsX iOS version costs $11.99. Another comparable mind-mapping tool, MindNode 6, costs $39.99 for a Mac version. It has its iOS version, as well.

MindNode and iThoughtsX have many advantages. They help researchers effectively find themes while organizing and reorganizing statements. They look playful. Researchers can share their thoughts on Q sampling with others by sharing digital files. These two programs even allow users to post a mind map on the Internet with just a few clicks.

Figure 15. The structure of nodes for sorting statements



1. Absence of social support is the main reason for drug addiction.

2. Use of drugs among peers is the main reason for drug addiction.

3. Unfavorable socioeconomic status is the main reason for drug addiction.

4. Stress is the main reason for drug addiction.

5. Parental and familial involvement is the main reason for drug addiction.

6. History of abuse or neglect is the main reason for drug addiction.

7. People living in Rural America do not have much entertainment so they take drug to overcome the boredom.

8. Early use of drugs is the main reason for drug addiction.

Discussion and Conclusion

Q methodology has contributed tremendously to the social sciences in understanding human subjectivity. Q scholars have developed and used computer programs for collecting Q sorts and analyzing them but have not done much for collecting a concourse of Q statements and generating a Q sample.

Thus, this study suggests using APIs, Google sheet, qualitative text analysis software, or text-mining algorithms for the latter purposes. These tools can save time for Q researchers and extend the scope of the concourse. Text-mining algorithms can even reveal some aspects of a topic that Q researchers might have neglected or missed, and quickly find statements reflecting those aspects. If these tools are efficiently utilized, they might have the potential to be the third approach in developing a concourse that Kenward (2019) recommended, a robust framework in addition to the inductive and deductive one.

The fate of the software programs discussed above is hard to predict. However, the diffusion of innovation theory, which explains how a product or an idea is adopted over time, can provide a clue about which one tends to be adopted easily. The theory lists five variables that determines the rate of adoption: relative advantage, compatibility, complexity, trialability, and observability of new products (Rogers, 1995).

Time and money are needed in learning new software programs, so Q scholars would not adopt them without any advantages like research productivity enhancement. Since some of the tools introduced earlier are inexpensive and have a shallow learning curve, so they could be more easily adopted than others. Tools like LDA and the widyr package in R might be difficult for more traditional Q scholars due to compatibility and complex issues, but these same tools might appeal to new scholars who feel comfortable with big data analysis.

Some complex tools could spread among the Q community with proper workshops or classes as trialability and observability attributes suggest. QUANAL, despite its complex nature, has dominated the Korean Q community because this has been used as the default tool for Q analysis in college classrooms and workshops in Korea. Therefore, complexity can be overcome by the efforts of innovation decision makers and their support of potential adopters. Some of the above tools could face the same road. If program developers devoted their time to the adoption of software, it would be more likely to be adopted by Q scholars than otherwise.

Software tools could help researchers excerpt Q statements for a concourse and/or select a Q sample. These same tools have the ability to streamline the Q methodology process, which is often desired in industry research and in pedagogical practice. All the software described within this article has passed beta-testing and receive continual updates, so those who are not satisfied with the current status of these tools could embrace them sooner or later. Scholars used to other digital tools should be able to adopt these tools without reservation. By adopting these tools for generating a concourse and a Q sample, the Q community would benefit as it did with Q sorting and Q analysis tools. As cognitive neuroscientists have benefitted from new technology like PET (positron emission tomography) fMRI (functional magnetic resonance imaging), EEG (electroencephalography), et cetera, Q scholars could benefit from adopting new technologies and taking advantage of the existing and the latest, emerging software as described within this article.

References

Akhtar-Danesh, N. (2017). QFACTOR: Stata module to perform q-analysis on q-sorts using different factor extraction and factor rotation techniques. *Statistical Software Components S458326, Boston College Department of Economics, Revised 04 June 2019.* Retrieved from ideas.repec.org: https://ideas.repec.org/c/boc/bocode/s458326.html

Banasick, S. (2019). Ken-Q Analysis (Version 1.0.6) [Software]. Available from https://shawnbanasick.github.io/ken-q-analysis-beta/index.html

Barbosa, J. C., Willoughby, P., Rosenberg, C. A., & Mrtek, R. G. (1998). Statistical methodology VII., a structural analytic approach to medical subjectivity. *J.1553-2712.1998.tb02786.x.pdf*, *5*(10).

Blei, D. M. (n.d.). Introduction to probabilistic topic models. *38ad42ed8a4428e395c96d57f83d201ef3b3.Pdf*. Retrieved from https://pdfs.semanticscholar.org/5f10/38ad42ed8a4428e395c96d57f83d201ef3b3.pdf

Borg, W. R., & Gall, M. D. (1979). *Educational research* (3rd ed.). New York: Longman.

Brown, S. R. (1980). *Political subjectivity: Applications of Q methodology in political science.* New Haven, CT: Yale University Press.

Burns R.B., & Dobson C.B. (1981) Experimental design and the control of error I. The between-groups unrelated design. In: Experimental Psychology. Springer, Dordrecht.

Cohen, D. (2017). *More than standard, less than enterprise: Twitter debuts premium apis – adweek.* Retrieved from www.adweek.com: https://www.adweek.com/digital/twitter-premium-apis/

Community API. (n.d.). Community API. Retrieved from developer.nytimes.com: https://developer.nytimes.com/indexV2.html?requestedPath=%2Fdocs%2Fcommunity-api-product%2F1%2Foverview

FlashQ - FAQ. (2007). Retrieved from www.hackert.biz:/ http://www.hackert.biz/flashq/faq/

Hardwick, C. (2013). *Practical design of experiments: DoE made easy.* CreateSpace Independent.

Kenward, L (2019). A literature review to guide novice researchers using Q methodology in the development of a framework for concourse management. *Nurse Researcher, 27*(1), 17-21. doi: 10.7748/nr.2019.e1616

Kerlinger F. N., & Lee, H. B. (2000). *Foundations of behavioral research* (4th ed.). Texas, Fort Worth: Harcourt.

Lee, B. (2019). Graphical representation of factors in oblique manual rotations for Q studies. Paper presented at the meeting of the 35th Annual Q Conference for the Scientific Study of Subjectivity, Naples, Italy.

Lee, B. (n.d.). A Q study of coffee shop consumers' preference types in South Korea. http://www.bestelon.com/q/coffee/WebContent/

LiquidText: Annotate & Review Documents. (n.d.). Retrieved from apps.apple.com: https://apps.apple.com/us/app/liquidtext/id922765270

McKeown, B., & Thomas, D. B. (1988). *Q methodology.* Newbury Park, Calif.: Sage.

Myre, M. (2019). The best mind mapping software in 2019: 11 tools to make great mind maps. *Zapier*. Retrieved from zapier.com: https://zapier.com/blog/best-mind-mapping-software/

Nazariadli, Morais, Supak, Baran, and Bunds (2019) Methodological Innovations pp. 1-16. Assessing the visual Q method online research tool: A usability, reliability, and methods agreement analysis. https://journals.sagepub.com/doi/pdf/10.1177/2059799119832194

Pedersen, T. L. (2018). Package 'ggraph'. *ggraph.pdf.* Retrieved from https://cran.r-project.org/web/packages/ggraph/ggraph.pdf

Pricing — Twitter Developers. (n.d.). Retrieved from developer.twitter.com: https://developer.twitter.com/en/pricing/search-fullarchive

Python Wrapper for Twitter Premium and Enterprise Search APIs. (n.d.). Retrieved from github.com: https//github.com/twitterdev/search-tweets-python

Q-Assessor. (n.d.). Retrieved from q-assessor.com. https://q-assessor.com

Q-Assessor Subscriptions. (n.d.). Retrieved from q-assessor.com: https://:assessor.com/ pages/3010

QUANAL Program Manual. (n.d.). Korean Society for the Scientific Study of Subjectivity.

Rogers, E. M. (1995). *Diffusion of innovations* (4th ed.). New York: Free Press.

Schmolck, P. (1999). *WebQ q-sorting over the net.* Retrieved from schmolck.org: http://schmolck.org/qmethod/webq/

Schmolck, P. (2014). Other implementations of computerized q-sorting (online or offline). *Computerized q-sorting implementations.* Retrieved from schmolck.org: http://schmolck.org/qmethod/webq/otherimpl.htm

Schmolck, P. (2018). *QMethod page.* Retrieved from schmolck.org: http://schmolck.org/qmethod/

Silge, J. A., & Robinson, D. (2017). *Text mining with R : A tidy approach.* Sebastopol, CA.

Singh, A. K. (2019). 7 free web annotation and markup tools you should know. Retrieved from: www.hongkiat.com: https//www.hongkiat.com/blog/top-web-annotation-and-markup-tools/

Software – Q Methodology. (n.d.). Retrieved from qmethod.org: https://qmethod.org/resources/software

Stephenson, W. (1952). Some observations on q-methodology. *Psychological Bulletin*, *49*(5), 483-498.

Stephenson, W. (1953). *The study of behavior: Q-technique and its methodology.* Chicago, IL: University of Chicago Press.

Stephenson, W. (1967). *The play theory of mass communication.* Chicago: University of Chicago Press.

Stricklin, M. (2011). *PCQ soft.* Retrieved from www.pcqsoft.com: http://www.pcqsoft.com/

Stroud, N. J., van Duyn, E., & Peacock, C. (2016). *News commenters and news comment readers*. Retrieved from https://mediaengagement.org/wp-content/uploads/2016/03/ENP-News-Commenters-and-Comment-Readers1.pdf

ToketaWare. (n.d.). Retrieved from www.toketaware.com: https://www.toketaware.com/ithoughts-howto-csv

Top 16 Qualitative Data Analysis Software - Compare Reviews, Features, Pricing in 2019 - PAT RESEARCH: B2B Reviews, Buying Guides & Best Practices. (n.d.). Retrieved from https://www.predictiveanalyticstoday.com/top-qualitative-data-analysis-software/

Top 3 File Sharing Software: Comparison of Dropbox, Google Drive, and Box - Financesonline.com. (n.d.). Retrieved from financesonline.com: https://financesonline.com/top-3-file-sharing-software-comparison-dropbox-google-drive-box/

Walker, B. B., Lin, Y., & McCline, R. M. (2018). Q methodology and Q-perspectives® Online: Innovative research methodology and instructional technology. *TechTrends*, *62*(5), 450-461. https://doi.org/10.1007/s11528-018-0314-5

Watts, S., & Stenner, P. (2012). *Doing Q methodological research: Theory, method and interpretation.* Los Angeles: Sage.

Weava: Premium. (n.d.). Retrieved from www.weavatools.com: https://www.weavatools.com/premium/

Zabala, A. (2018). Package 'qmethod.' Retrieved from https://cran.r-project.org/web/packages/qmethod/qmethod.pdf