QE INTERNATIONAL SOCIETY FOR QUANTITATIVE ETHNOGRAPHY
QEsoc.org WEBINARS

http://qesoc.org/webinar-series

Note this was automatically transcribed by Zoom so there are errors, and the timestamps do not match the video.

June 27th 2013

Integrating QE methods into a qualitative workflow

Jonathon Sun

University of Pennsylvania

Abstract

Qualitative methods can be daunting because of the sheer number of themes that arise from the data; however, quantitative ethnography (QE) has provided resources to begin making meaning. This workshop will introduce participants to methods to assist in making meaningful qualitative themes, such as a co-occurrence network. First, this workshop will show participants how to clean and import data from other qualitative software like Dedoose. Next, a co-occurrence matrix will be made using the R package/plugin SNAHelper. Finally, based on the selection of themes, an ENA will be built based on the themes from the selected network.

1
00:00:09.400 --> 00:00:18.910
Jamie Boisvenue (He/Him): Right. Good morning, everybody. Welcome to the Qe. Webinar series. I am your session host, Jamie Avenue.

2
00:00:19.430 --> 00:00:22.459
Jamie Boisvenue (He/Him): tuning in from from very smoky Canada

3
00:00:22.500 --> 00:00:26.979
Jamie Boisvenue (He/Him): and apologize to any of our Us. Neighbors who are still, you know, coughing

4
00:00:26.990 --> 00:00:29.040
Jamie Boisvenue (He/Him): and and breathing in the terrible smoke.

5
00:00:29.090 --> 00:00:40.300
Jamie Boisvenue (He/Him): Today we have a very awesome presenter. Jonathan son, who is a doctoral student who is studying higher education at the University of Pennsylvania.

6
00:00:40.300 --> 00:00:59.249
Jamie Boisvenue (He/Him): Jonathan is an experience assistant director of Diversity with a demonstrated history of working in the higher education industry. He's also a really strong human resources, professional with a master's degree focusing on education, leadership and policy analysis from the University of Missouri

7
00:00:59.250 --> 00:01:17.430
Jamie Boisvenue (He/Him): Columbia and today is going to be talking to us about the qualitative methods that can be used in conjunction with in a in our and deduce, which is a useful tool. So I will hand it over to Jonathan, and thanks you everyone for joining us today.

8
00:01:18.380 --> 00:01:34.369
Jonathon Sun: Thank you so much for the wonderful introduction, and it's great to see some friendly faces here. it's been a while with some folks, so Hello! As Jamie said. I am a fourth year Phd. Student at Upenn, studying higher education. I particularly study Asian Americans in college access with a little bit of gis as well.

9
00:01:34.370 --> 00:01:57.799
Jonathon Sun: And so today's workshop, we will be getting into a little bit of our. But I also want to say, if you don't know our, that's okay. The goal of this is to teach you the big picture ideas of how we're using our. I'm not expecting you to be like, oh, I know this function right off the bat like no, I don't. I still Google, search many of the functions that I use. And so for folks who are brand new to are and have no idea what's going on your a okay. And we are glad to have you here.

10
00:01:57.860 --> 00:02:27.079
Jonathon Sun: so with that I will hop into the presentation. If you do have questions throughout the presentation, please just drop it in the chat. I have experiences live streaming and responding to chat messages as part of the job. So please make sure that if you do have questions. You drop them in, let me go ahead and share my screen as well. Can everybody see the screen? And does anybody have any comments as of now, or questions before hopping into things.

11
00:02:27.420 --> 00:02:29.199
Jonathon Sun: Let me move this.

12
00:02:30.700 --> 00:02:43.239
Jonathon Sun: okay, I cannot see you all. I am just going to take that as it is

13
00:02:43.240 --> 00:03:02.609
Jonathon Sun: With that we are going to do some open coding and meaning making, using Qe methods for parsimonious coding. this comes from a poster presentation that I'll be presenting at Icqe 2,023 this year, and comes from a data set that I'm working with on working on with Dr. Mali and her project on North Philadelphia.

14
00:03:02.610 --> 00:03:23.719
Jonathon Sun: And so the abstract of this presentation is qualitative. Methods can be daunting because of the sheer number of themes that arise from the data. However, Qe. Has provided resources to begin the meaning making from these teams so hopefully, this workshop will introduce you all to some of the methods to take apart some of these qualitative themes, using co-occurrence networks and ena.

15
00:03:23.720 --> 00:03:38.279
Jonathon Sun: Hopefully, we'll also be able to dive into a little bit of qualitative data software analysis and using deduce and taking deduce data and putting it into segmented portions. So you can have it in an en or any other analysis.

16
00:03:38.540 --> 00:03:48.820
Jonathon Sun: next. And finally, after that we'll go into using the our package. Plugin. Sn a helper, which is an incredibly easy tool to use. that helps you make a social network analyses.

17
00:03:48.820 --> 00:04:08.750
Jonathon Sun: And then, finally, we'll have an epistemic network analysis from that, and they are in a portion. I will not go into all the coding that went into that, because there are workshops for that, and I see your news here. And so she could absolutely speak to that point, and I'll show you that the final product of it. but one at least introduce those ideas to you all.

18
00:04:09.000 --> 00:04:34.599

Jonathon Sun: and let me get my timer started my apologies the goals of this workshop. So hopefully from this workshop, you'll be able to understand the qualitative coding process. no qualitative data management systems. using our prepared data for segmentation. And Ena, make a network using sn a helper and select codes from network analysis using Ena. And the outline is we'll review qualitative research. like, what qualitative research looks like. And the the life cycle of it

19
00:04:34.600 --> 00:04:42.900
Jonathon Sun: qualitative research software is going from the deuce into our going through network analysis and then our innate with the selected codes.

20
00:04:43.760 --> 00:05:03.189
Jonathon Sun: So first let's start off with what are qualities of qualitative research, using Miriam and Tisdell from 2,015. I believe we've all seen this book in some capacity. Qualitative researchers are interested in understanding the meaning people have constructed. That is how people make sense of their world and the experiences they have in the world.

21
00:05:03.190 --> 00:05:27.099
Jonathon Sun: So as qualitative researchers, we ask questions that we want to understand processes. How does this occur? So, for example, although I can identify like Asian Americans, are more educated, can be more educated. That doesn't get me the nuance within. Why, Asian Americans are more educated, what kind of opportunities they have, but also it doesn't give me the the nuance that some Asian Americans may not necessarily have

22
00:05:27.100 --> 00:05:51.199
Jonathon Sun: the same educational outcomes as other groups. And so when we ask quality to questions and ask experience questions. We want to understand how people are making sense of their lives. We also recognize in qualitative research that as the researcher, we are the primary instrument for data at collection and analysis meaning, we should be aware of how we perceive our data, how we interpret our data, which means we take lots of memos. we take

23
00:05:51.200 --> 00:06:16.200
Jonathon Sun: that when we write our field notes, that it is our from our lens, and that our experiences, and that we've had, that we've had, whether it be from childhood to college to the recent course that we took shape, the way that we look and analyze our data. And so through the process of qualitative research we are able to produce rich and thick descriptions. Again, whether it be the way that we write our field notes, or whether that means that's the stories that we are able to engage our

24
00:06:16.200 --> 00:06:28.139
Jonathon Sun: our participants in and telling how we can contribute and how we can bring out these experiences. Qualitative research is about producing this thick description and understanding the processes of how we got there.

25
00:06:28.870 --> 00:06:49.900

Jonathon Sun: So usually in the qualitative research process, from a grounded theory. perspective, grounded theory. It consists of categories, properties, and hypotheses that are conceptual links between an among and among categories and pro properties. And so when we develop a theory from grounded theory, it comes from properties, a core category and hypotheses.

26
00:06:49.900 --> 00:07:14.399
Jonathon Sun: And so the one way to think about this is properties are kind of like your subcodes, your cat core categories are your primary codes, and your hypotheses are the way that they communicate with each other. So, for example, the definition here is concepts. properties are concepts. Describe a category cities like subcodes. So in this case, if I were reading through my transcript and say, I'm again coding for Asian American college access.

27
00:07:14.580 --> 00:07:35.910
Jonathon Sun: I am reading that Timothy attends a after school, like as Timothy. He's talking about his experiences going to an after-school resource. This is not necessarily what I'm focusing on for college access. But I'm like, Oh, this is related to college access. So I will code that as a property the core category, though, would be

28
00:07:35.950 --> 00:08:05.220
Jonathon Sun: college access. So then, the hypothesis that I have between community resources and higher education is the hypothesis. What is this connection? And of course there is literature that I can tie this to. But in grounded theory. There's also theories that I can pull in experiences that I can tie together to say, Oh, perhaps these community resources are incredibly helpful in training students to know what exactly to write into their their essay. And so that property goes into the core category.

29
00:08:05.220 --> 00:08:09.180
Jonathon Sun: And I build a hypothesis around how that works.

30
00:08:10.200 --> 00:08:31.069
Jonathon Sun: So as you go through your coding you will go generally in grounded theory. There's 3 steps. You first do your open coding, and this is the first round of code in coding, tagging any unit of data that may be relevant. So, for example, going back to Timothy and his community resources we go through. And just read the sentence like.

31
00:08:31.080 --> 00:08:51.470
Jonathon Sun: yeah, I was tired of going to Kumon after school. We code that and say, like tutoring kumon, etc. Now, on our second read through we try to tie all these different concepts that we have tied together, so whether that be tutoring, whether that be I hung out with my friends. whether that be my family and I ate dinner together.

32
00:08:51.640 --> 00:09:12.929

Jonathon Sun: We try to pull these ideas together into bigger buckets and axial coding relating to each other in refining the coding schema. Then finally, once we do our third read through and try to get all that axial coding worked out. We do selective coding. And this is where we start to develop our hypotheses and our ideas. But this is the part where it gets tricky. as

33
00:09:12.930 --> 00:09:21.660
Jonathon Sun: We are trying to figure out what are the themes, and how are they related? And of course, we can use our experiences, our literature in the text.

34
00:09:21.660 --> 00:09:45.569
Jonathon Sun: but also that can get really, really confusing. And so this is from my master's thesis where I was studying Asian American post coloniality in student organizations, and while I did not have the tools of E and A. Or Qe. Or ma, just any network, analysis or quant. Really, I built a mind map to try to figure out how all these things worked out, and, as you can see, it was a bit confusing at times, and this is not to say that this doesn't work.

35
00:09:45.570 --> 00:10:10.310
Jonathon Sun: There are still many people who still use the the old fashioned sticky notes, paper strings To make sure that to see how concepts are tied to each other so very much. Imagine it like the meme from it's all it's fun sunny in Philadelphia, where it's the guy with the billboard. It's like. It's it's you know what? Whatever the the phrase goes along with it. But that is essentially how we, as qualitative researchers, have to tie everything together.

36
00:10:10.310 --> 00:10:34.720
Jonathon Sun: What are the hypotheses that tied these concept concepts to these smaller properties. How are these concepts related to each other? And so in this case, I think it's important that we use en A and Qe. To help us assist in that process. So, walking through the qualitative research process one more time, we start off with some research hypothesis. We usually have an idea. We will go and then collect the data. We then code the data.

37
00:10:34.720 --> 00:10:52.229
Jonathon Sun: open coding. And then we develop codes and we iterate on this process. And that's what makes Qe unique in in contrast. Or that's still part of the quality of the process. But what makes Qe unique in that is, we use tools such as E and A, such as network analysis, to support and help us

38
00:10:52.310 --> 00:11:11.090
Jonathon Sun: interpret our data. And so that's why I propose using network analysis and E and A or Qe. Methods in general to help us in this last portion. And as we go through qualitative research, it's also important that we recognize that it's a very reflexive process, that sure, you may have a plan, and we may be iterating on this. But things can change based on

39
00:11:11.320 --> 00:11:20.770

Jonathon Sun: the people that you engage with the spaces that you explore. And so how does that impact your coding process and the codes in general in developing the themes.

40
00:11:21.580 --> 00:11:50.290
Jonathon Sun: I want to pause there before we get into anything else. the folks have questions on the general layouts. A very fast 10 min overview 7 min overview of the qualitative research process. I also, for some reason, cannot see everyone's face. So if you have a question please drop it in the chat while I figure out how to look at everyone, because I can't see anyone. And I'm just talking at my screen.

41
00:11:59.440 --> 00:12:03.910
Jonathon Sun: Okay. I will take that as no questions as of now.

42
00:12:04.590 --> 00:12:07.860
Jonathon Sun: because I still can't see the video panel. Okay?

43
00:12:08.230 --> 00:12:29.899
Jonathon Sun: So let's go into some of the qualitative data analysis software that you can use. So I've listed a bunch of them over here. I have deduce Max, Qda and vivo atlas Ti Rock reproducible open coding kit and Qda minor. The reason that some people would use this is it helps consolidate all of your research materials. So you can.

44
00:12:31.230 --> 00:12:50.240
Jonathon Sun: no qualitative research process is not a positivist leaning process. It depends on the paradigm that you use. But for the most part the qualitative research process is very open to interpretation. And the way that you go through your data, so going through the grounded theory process, one of the assumptions is that?

45
00:12:50.250 --> 00:12:53.319
Jonathon Sun: you understand it? That the data, the

46
00:12:53.360 --> 00:13:10.039
Jonathon Sun: the outcomes, arise from the data, and so based on our own lenses as researchers, it shapes how we interpret and make sense. no, you're totally fine. Thank you for dropping that in. And my apologies for not being able to figure out why I can't see the video. It popped up earlier.

47
00:13:10.080 --> 00:13:15.560
Jonathon Sun: I bet I know what it is. Actually I think we'll be okay.

48
00:13:15.700 --> 00:13:18.809
Jonathon Sun: But thank you for that question. I greatly appreciate it.

49
00:13:20.400 --> 00:13:32.280
Jonathon Sun: with that I will hop back into the qualitative data analysis software. So there's a couple of different software out there. I am most familiar with deduce, and in vivo I've using vivo in the past and deduce

50
00:13:32.280 --> 00:13:55.720
Jonathon Sun: is also a a nice tool. I'll be a bit laggy at times. But it works well with collaborative research projects. If you're all coding through one set of data, and I'm not going to speak to what each of these software is do. But I will speak to what they broadly can do. And it helps you organize your research products, projects. You can include descriptors memos, all in one place, and oftentimes the

51
00:13:55.980 --> 00:13:58.680
Jonathon Sun: Qdas's will. Also.

52
00:13:58.970 --> 00:14:26.239
Jonathon Sun: We'll also do some analysis in there, whether it be qualitative or quantitative they can do inner code of reliability and them. But there's also a cost to it. But the problem with this is that when you put your research and you put your content into these tools, it's often difficult to get it out. So you can do epistemic network analysis.

53
00:14:26.320 --> 00:14:51.570
Jonathon Sun: the question is, how do we get our data out of deduce or in vivo? In this case we'll be speaking, articulated, deduce, and get it to connect with the data that you want. So as we know for epistemic network analysis, and other Qe methods, you have to have your data segmented by line, by some kind of utterance, and you need to have your codes organized on the on some way as true false statements.

54
00:14:51.570 --> 00:15:00.950
Jonathon Sun: So this is how we want our data organized. But this is not how the data comes out from. Deduce the data from deduce we, we'll talk about where the how the data looks.

55
00:15:00.950 --> 00:15:27.960
Jonathon Sun: But the general idea that we have for trying to move our data from deduce or any other qualitative data analysis software into our and connect it with our documents. So first, we need to clean our interview documents. That means we need to first get it into our In doing so, we get into our. And then 2, we need to find some way to segment the documents using some symbol. So this is very similar to the process that we see in the reproducible open coding kit where you set something to be the segments.

56
00:15:28.610 --> 00:15:54.019
Jonathon Sun: Next, you need to prepare your deduce data. You need to export your data and read it into our again. And then you need to find a way to segment the deduced data into smaller

segmentations than the documents. This requires some intentional planning from the get go For this project. I was very fortunate, very fortunate, that things worked out the way they should. and I was quite surprised it actually did work out. But there is a lot of intentional planning that needs to go on in advance.

57
00:15:54.020 --> 00:16:17.939
Jonathon Sun: Finally, you need to merge the interview and deduce data together. and so this can be attaching corresponding id to segmented portions or and then left joining them. again, I'm gonna use some our language, but for the most part I hope that folks can still pull the ideas so they can apply this to other projects. Not. There is no single way of doing this, because every single person transcribes and goes through their data differently.

58
00:16:17.940 --> 00:16:26.139
Jonathon Sun: even the way that the transcribers transcribed the data for us, for this project was very helpful, because we had some way of automatically doing terms of talk.

59
00:16:26.890 --> 00:16:47.880
Jonathon Sun: So first let's go into cleaning your interview documents and importing your documents. So on the left is a small snippet of how our data looked, as you can see. when I use our word. I want every single thing labeled out so you can see all these paragraph breaks. you also see that we have times and participants, and then we have the text here.

60
00:16:47.930 --> 00:17:10.309
Jonathon Sun: and when we import, so you can import the word document in a couple of ways, you can import just the text from the document, or you can try to keep all the formatting. And so in this case I wanted to keep all the formatting. So I use the library officer. officer has ways to read documents, excel sheets, etc. And I figured I might be working with all those in the future.

61
00:17:10.319 --> 00:17:29.229
Jonathon Sun: of course, I also use tidy verse, and this is just an easy language. This is a way to work through like use tidy functions like left join etc. deployer is also just a way to get the pipe function in here. It's stuff like this, so you can work through. It's just makes coding cleaner.

62
00:17:29.390 --> 00:17:39.739
Jonathon Sun: So the first thing that I did was I read in the documents one document to test it out, and what I did in this is, it first produces, read, Doc produces the document. But when you

63
00:17:39.740 --> 00:18:03.049
Jonathon Sun: use this Doc function it doesn't come out as like a nice pretty sheet like this. It comes out as a list. So you have to have dox summary into. Yeah. You use doxx summary to produce something that looks like this. And as you can see, this is not very helpful. because one. I have all my texts cleaned because I asked it to separate it. But also there's a lot of spaces, and that's because it's breaking every line break.

64
00:18:03.050 --> 00:18:12.550
Jonathon Sun: But that is helpful, because in this case we can use those line breaks as our terms of talk. And so in this case, our research team decided that we wanted to use terms of talk to

65
00:18:12.570 --> 00:18:14.189
Jonathon Sun: segment the data.

66
00:18:14.880 --> 00:18:40.299
Jonathon Sun: So what I did after that was, I then said, Okay, we should probably keep the participant numbers, because that's something separate. And so it's a descriptor. I also want to take out the Timestamps because it's not necessarily a part of the text, but I think it might be important to keep as identifying information. So I then cleaned it out with a st extract string extract, and I then I'll go through the details of some of these in the later

67
00:18:40.840 --> 00:18:51.730
Jonathon Sun: slides. But I extract the times, and then I got it to look like this. And so this is the original text, and how it looked without the spaces. And this is with just the text.

68
00:18:52.580 --> 00:18:58.799
Jonathon Sun: So the reason why we use our is because while I can copy paste this code at least

69
00:18:59.470 --> 00:19:19.599
Jonathon Sun: 25 times. It doesn't make it very easy to clean up. And so we then we make a function to put all this into one place. And so this function, basically, I put the dock in. I specify what where the document is located, and I ask it to just iterate I ask it to. Whenever I put a document in here, I want you to clean all these things out.

70
00:19:19.640 --> 00:19:48.339
Jonathon Sun: And so now that that's a function that's made. I can then loop that for every single document. So I don't have to do that. And of course, as you make this, please double check as qualitative researchers. We are responsible for what's going in. And we should figure out what's what it's looking like and what is being written. And so of course I went through. I did not check every single place, but I've Spot checked and made sure that. Okay, this looks right. I sent it to Amalia, and made sure that it also looked right there.

71
00:19:48.800 --> 00:20:12.580
Jonathon Sun: And so This is how I was able to at least get all my documents cleaned and put into our. And again, it's important that we mentioned that the segmentation process for this was terms of talk. I will mention why that's important later. So when we get to preparing our deduce data, when you export data from deduce you get this, and you can specify what

72
00:20:12.850 --> 00:20:31.730

Jonathon Sun: what columns that you want in your data. And so you can see that here I have all the media the excerpt ranges, the Creator, the dates, etc. and the excerpt copy. Now, when we're coding, introduce Oftentimes you decided with it as a research team like, do you want chunk coding sentence coding, but sometimes like it just varies.

73
00:20:32.230 --> 00:20:49.220
Jonathon Sun: And so, because of that, what I need to do in this case is, how do I segment my excerpt copies so that they fit exactly like it is in the original document for using the same original documents, and I segmented them similarly, then, ideally.

74
00:20:49.350 --> 00:21:05.510
Jonathon Sun: it should layer it should be able to left, join later, because it is the exact same set of words. And so what I do first is I list the path I export or import this into our. And I add unique ids,

75
00:21:05.820 --> 00:21:09.379
Jonathon Sun: and of course you can see here that I also have a lot more codes.

76
00:21:09.870 --> 00:21:36.120
Jonathon Sun: So next. What I do is very similarly to what I did with the word document. I took the each cell, and I said, I want you to identify where there's a row break, and very much similar to the transcript that we had, that there was our There was a paragraph break. I did the same thing with my deduce data, because that means I'm ideally segmenting it the exact same way that I done earlier, and if that is the case, I should be able

77
00:21:36.140 --> 00:21:59.109
Jonathon Sun: to use a our function, Grepel, to say any instance of this word should be found in the other document. So the process to this was I. One split all the data by line break, and our does this with the cr split function and gives us a list where segments of slash are are present. I then unlisted. This made a data frame just because it's easier to work with for me. I'm bad at working with lists

78
00:21:59.110 --> 00:22:18.830
Jonathon Sun: and then 3 We then change the name of the column, because when on list, it gives it basically like a period as a name. And so I loop this and said like, just rename it excerpt, and then I filter out any empty lines, because, similarly to the other, like when you're extracting the original word document, it doesn't necessarily work. So you have to clean out all the the empty rows.

79
00:22:19.700 --> 00:22:49.380
Jonathon Sun: And so, as you can see, I ran this as a loop, and there was 1,243 codes, and this takes a little bit of time, but this is still much more effective than me doing this, and also potentially making mistakes. If I'm doing this at like 12 am. I am tired. mistakes happen. But also, people can review my code afterwards and be like, does this work? I think something got missed here. Someone can double check. And also, if we do add more documents in the future, and we add more codes. I can just put this back in and we run this script.

80
00:22:49.380 --> 00:22:52.019
Jonathon Sun: So this gives us a new dose codebook

81
00:22:52.020 --> 00:23:18.190
Jonathon Sun: where each line is now splits by segment. And additionally, I've been able to retain all the codes connected to each of the segments. because it's split at the slash r. And so you can see in these excerpts. I've kept the Timestamps in this case is because these Timestamps offer unique ways to be able to connect the excerpts that we have and deduce to this text that we have segmented in our original word document.

82
00:23:18.600 --> 00:23:39.989
Jonathon Sun: And so this is the fun part merging the set. This is a bit confusing. But allow me to explain this. The idea that we have here is that I need. I need our to one. Go through the original deduce data set that we have here this line. I want this line to be checked for every single line in here

83
00:23:40.130 --> 00:23:41.870
Jonathon Sun: and or in here.

84
00:23:41.930 --> 00:23:51.949
Jonathon Sun: and when you find the exact same line, I want you to give it a unique id that says, Boom, you are connected to unique. Id.

85
00:23:51.960 --> 00:24:00.050
Jonathon Sun: 23. And what this does is then gives me a way to merge my set of coded excerpts

86
00:24:00.070 --> 00:24:24.079
Jonathon Sun: to the final product of the segmented data. And so what these lines are are some exceptions. sometimes the Spanish did not play well with deduce, and it would or not to do with our. And the function would just stop. I have it basically run the exact same situation or the exact same code as above. I didn't write it as a function, because I didn't want to break it.

87
00:24:24.080 --> 00:24:44.669
Jonathon Sun: So I hadn't included the exact same does the exact same thing just how to handle those exceptions. So again, what I did here was I remove the text slash and from the text, and save it as check using Grepel Grepel then checks. If the phrase is found in any of the segments from the segmented text, if there is a match, the unique Id from the coded text is attached to the segmented transcript text.

88
00:24:44.670 --> 00:24:50.929

Jonathon Sun: And finally, there are words that just did not work out. And so I had to build separate cases for that.

89
00:24:51.900 --> 00:25:11.119
Jonathon Sun: Once you get all your data merged and you have unique Ids. You then do a left join which puts them together, and you can see that this is the final product. As we have the text. I have the interview that it came from the unique Id line from the text, and also this.

90
00:25:11.120 --> 00:25:18.009
Jonathon Sun: I should also note that the order which you first input your documents matters

91
00:25:18.010 --> 00:25:41.830
Jonathon Sun: because that is going to be the whole. That is the discourse transcript that you have. So if you are interested in organizing your documents by date when you took it. that matters or if you're interested in like topic, I want this topic first. This topic first, however, you organize this whole sheet, and other any other descriptors that you have to order the documents does matter, and how you're going through your next steps of analysis.

92
00:25:42.450 --> 00:25:43.550
Jonathon Sun: And so.

93
00:25:43.760 --> 00:25:52.229
Jonathon Sun: in short, we are able to pull, deduce data the excerpt from, deduce, merge them into the segments that we have.

94
00:25:52.270 --> 00:26:09.179
Jonathon Sun: There's a couple of other things that we need to clean, though before we go into the next steps of analysis, deduce does not automatically count sub codes to primary code frequencies. And so, because I am using the ideas of grounded theory, where, like core concepts and there's properties that contribute to them. I was trying to

95
00:26:09.180 --> 00:26:23.610
Jonathon Sun: get a bigger picture, a lens of how my data worked. And so I had to aggregate the code, the subcodes into the primary codes. I'm not going to talk about this part too much, because there's a couple of ways to do it. I

96
00:26:23.610 --> 00:26:46.720
Jonathon Sun: remember how I did it, but it is a little time consuming to explain, and I've spent a little too much time trying to explain some other stuff. But this is all to say. If you are interested in trying to find a way to aggregate your codes. I'm one happy to share this code. I think this code is more standardized because of the way that deduce imports the data. But again, I just run a for loop to make sure that all the data is aggregated.

97
00:26:47.260 --> 00:27:10.150
Jonathon Sun: Once your data is aggregated under your primary codes. I then work on making a co-occurrence network, a co-occurrence matrix first. And so this is taking all the true false values that we see by segment and counting, how often do they overlap. And the reason why we are counting overlaps is because I'm assuming

98
00:27:10.150 --> 00:27:24.469
Jonathon Sun: that the themes are related to each other, and if themes are related to each other, then for the most part, if they are overlapping. Often you have 2 options, one. These 2 concepts are potentially the same, and you should probably

99
00:27:24.530 --> 00:27:45.770
Jonathon Sun: consider consolidating them or renaming them, or to these are actually very different topics that are very, very related. And how are they related is the question. And so you can see here, like, I have these topics together. what granulator is going into the co-occurrence matrix is essentially

100
00:27:46.290 --> 00:27:56.239
Jonathon Sun: I have it going through every single segment that we have. So it's segmented. And yeah, by segments, by segments. And then into the co-occurrence matrix.

101
00:27:56.250 --> 00:28:14.109
Jonathon Sun: I hope that answered your question. So basically, I will say that I found a script online from stack overflow that made the co-occurrence matrix and that is included at this bottom part here. It looks like it's a cross product. I also had another line of script that ran for

102
00:28:14.330 --> 00:28:39.620
Jonathon Sun: almost a week for all the data, and it did not finish. That would have organized the data, but it kept getting slower and slower. And I want my computer back. if you find a way to run that in the cloud, let me know. But for the most part the fastest way was just doing a co-occurrence matrix from this code, that, or from this, the script that I found from stack overflow. So from this pro occurrence, matrix that you make

103
00:28:39.620 --> 00:28:56.959
Jonathon Sun: what I do is I then pivot long, or I yeah. Pivot long from pivot wide. And what this does is this gives me 2 sets of pairs. This one, I get one set of codes. That's this is the code. And 2, this is everything it's related to.

104
00:28:57.770 --> 00:29:13.380
Jonathon Sun: And so from here. yeah. I renamed V. One to co-occurrence, etc., co- occurrence and totals here. And so then I have set pairs of point one to point 2, and I have some value here of the amount of Co occurrences.

105
00:29:14.120 --> 00:29:24.350
Jonathon Sun: So from here we are able to then use the network analysis package, and for this. I am not a network analysis expert, but I do know the basic

106
00:29:24.410 --> 00:29:50.389
Jonathon Sun: options for network analysis. And so for this, you will use the eye graph package, the Gg. Graph package and a plugin on our studio called Sn. A helper. The sn. A helper is a very helpful package. I will include the This is in the presentation. So if you're interested, you can click on that later and get there. But basically, what you need to do is with the co-occur. Clean data you specify in here.

107
00:29:50.560 --> 00:30:02.430
Jonathon Sun: what your edges are. So I selected v, one, v. 2, and percent co occurrence. 2. You specify your nodes, and so in here you can. I've specified my nodes here on that line over there.

108
00:30:02.580 --> 00:30:12.079
Jonathon Sun: and what you do is you? Then use a function called graph from data frame at the bottom that makes a G or an eye graph which you can then highlight

109
00:30:12.080 --> 00:30:35.749
Jonathon Sun: and then click on the plot. Add-ins over here. Go to S. And a helper, and then this ui will pop up and tell you what you are. Build a network for you, and you can change the color. You can change the size. there's a lot of different options. You can obviously do this scripted out but it helps folks who are not familiar with network analysis, and for folks who just want to get their hands into the code and look at it.

110
00:30:35.820 --> 00:31:02.939
Jonathon Sun: And so network analysis helper will just plop out the code exactly as it is written. And so this is the code that we made from our project, and so from here we then are able to use our portion of interpretation. And so from here we can see that there is a relationship between safety and depth, based on the amount of of Co. Occurrence between safety and theft you'll see living in Philadelphia. Kind of education. These are related in some way.

111
00:31:03.530 --> 00:31:17.960
Jonathon Sun: And what I did from this point, though, is from here. I'll rather than just looking at all the values, because it is overwhelming. I do need to find a way to clean that code up. But what we can do here is we can use that set up

112
00:31:17.980 --> 00:31:34.600
Jonathon Sun: coded pairs from earlier and If we take this and then convert them into rates we can then say, I want the top 7 or top 9, 7 pairs to the highest co-occurring percentages and select them as codes. And so

113
00:31:34.820 --> 00:31:46.839
Jonathon Sun: even though there's 7 pairs, there's only 9 codes. And this is because of those 14 or of those 7 pairs some of them overlap quite often. the Xy positions

114
00:31:46.840 --> 00:32:07.039
Jonathon Sun: the Xy positions. I don't believe in this case have too much purpose. I am not. I will need to double check. I don't know the final answer to that I'm not sure if they have a relationship. And so someone who is more educated on network analysis will need to assist me there. My apologies for not being able to answer that

115
00:32:08.070 --> 00:32:21.570
Jonathon Sun: But in the themes that we have the 7 pairs we have these 9. And so I would also like to caution that. Yes, you can just copy paste and just say, Yes, I want the 7 pairs.

116
00:32:21.570 --> 00:32:46.719
Jonathon Sun: But I think it's also important that as we are applying our lens into this, that you make sure that these things make sense. Some of these, most of these things made sense for us. safety. Violence was something that popped up a lot in our conversations, however, something that did not arise as often was religion, and so religion was. I am cure. We were curious as a team as to how, why, what the relationship with religion was. and so that's why this was still included.

117
00:32:47.690 --> 00:32:55.169
Jonathon Sun: yes, this method is essentially selecting codes that occurs in pairs most often.

118
00:32:57.080 --> 00:33:02.440
Jonathon Sun: thank you, Dr. Schaefer. So we are able to use these to select

119
00:33:02.800 --> 00:33:14.440
Jonathon Sun: the the themes that we have. And so again, just noting that as a research team, you should double check, that these things make sense for you, but also, if to something odd, perhaps to do something to look into, if not.

120
00:33:14.630 --> 00:33:32.760
Jonathon Sun: just let it be. And so that's how I selected the 7 pairs. I selected the codes with the highest co-occurring percent, 7 pairs of the highest co-occurring percentages. And finally, then, because there was a lot of segments I could not use that into the E and a web tool.

121
00:33:32.920 --> 00:33:55.370
Jonathon Sun: or I could probably just be really slow. And so I did things through our en a. My apologies for the fuzzy quality. I need to re-export this, but from here we are able to select our themes, and we can see the relationship that the codes have within the discourse of each other. So

again, going back to the difference here is this, just tells us how often, how, how's co-a? How often these co-occur with each other.

122
00:33:55.450 --> 00:34:03.050
Jonathon Sun: This tells me exactly what the relationship in discourse is, and so this now position does matter

123
00:34:03.160 --> 00:34:21.770
Jonathon Sun: so we can see safety and violence. We're often coded next to each other, which makes sense in reading our documents. Someone who is talking about safety often mentioned at some point for or after that that we have. that they had a experience some kind of violence beforehand. we can also see. Transit is somehow

124
00:34:21.920 --> 00:34:26.779
Jonathon Sun: weekly related, but still centered in some capacity.

125
00:34:26.909 --> 00:34:42.800
Jonathon Sun: and so we can use E and A to double check the ideas that we have that we figured out, or we found from the network analysis and bring that back into a network analysis. And we can continue iterating on these, going back to the axial process and then finally making a

126
00:34:43.000 --> 00:35:11.069
Jonathon Sun: We can finalize what we have here. what are the unit of analyses in the Ena? I didn't talk too much about the E and a here only because I wanted to focus on the net the network analysis process and the how it ties into here. But this units of analysis were by segments, I believe. I need to re-familiarize myself with data set.

127
00:35:11.410 --> 00:35:13.020
Jonathon Sun: but I believe it was

128
00:35:13.080 --> 00:35:30.689
Jonathon Sun: It was the segments that we had, and so we had documents as well. I haven't gotten too much into the cleaning of each document yet, as this is still exploratory and just trying to make sure that I understand the big code. So then we can bring them back into conversations.

129
00:35:30.690 --> 00:35:55.440
Jonathon Sun: So hopefully. this provides some context on how to sort through the many codes that you do have. this is a way for you to be able to iterate on your data collection, open coding and development of your codes. Process and integrate this rather than just sitting in. You know. Lots of a thumb tax sticky notes, etc. Lots of memos. put in some of the data and figure out how you can clean the data and play with the data. What themes arise.

130

00:35:55.450 --> 00:36:11.590
Jonathon Sun: Go back to your data collection, your open coding, and figure out how you want these things to work together. that is all I have for the Qe research process and tying qualitative research practices into your qualitative workflows.

131
00:36:12.590 --> 00:36:14.660
Jamie Boisvenue (He/Him): Amazing. Thank you, Jonathan.

132
00:36:14.820 --> 00:36:20.969
Jamie Boisvenue (He/Him): I do see some ch questions coming into the chat here just to go back.

133
00:36:21.160 --> 00:36:40.139
Jamie Boisvenue (He/Him): Mariah is asking what granularity is going into the co-currence matrix. She was wondering if she missed that. I'm not sure, if you answer that while speaking, I think I did. The granulator that went into it was my segments. So each the documents I left them all there, and I took that whole thing and put into the co-occurrence matrix. So it counts every single segment.

134
00:36:40.410 --> 00:36:41.460
Jamie Boisvenue (He/Him): Excellent!

135
00:36:41.930 --> 00:36:58.370
Jamie Boisvenue (He/Him): of course, if anyone has a questions feel free to unmute yourself and and speak if you would like. if you are uncomfortable doing that, you can also post your question in the chat, and I can represent you as a proxy.

136
00:36:58.950 --> 00:36:59.760
Jamie Boisvenue (He/Him): Brendan.

137
00:37:00.340 --> 00:37:20.450
Brendan Eagan: I'll go really quick, Jonathan, am I correct in my understanding that if you try to to export data from deduce with the codes. It. It's gonna it's gonna leave out all the lines that are not coded for the given codes. And that's why you need this process basically to like it's very laborious. Because we, I was just talking to somebody about this.

138
00:37:20.480 --> 00:37:33.050
Brendan Eagan: And I, my recommendation was like to either use a separate tool or have to be very careful. Because you, I mean, I think what you've kind of demonstrated is these are the things that you need to do to reconstruct, not only just to do what you're doing with the coding, but even to get

139
00:37:33.080 --> 00:37:41.669
Brendan Eagan: data from deduce into E, and A just in general, you have to kind of go through some process like that. Am I correct in my understanding? Yes,

140
00:37:41.670 --> 00:38:03.279
Jonathon Sun: Deduce is not built to segment, you know. did you? You? I basically had to rebuild the whole data set. So when you export from, do it only gives you whatever you coded. So other options, I was thinking, was, well, if I just code the whole, every single document in its totality and code, it does not work that does not work.

141
00:38:03.550 --> 00:38:09.549
Jonathon Sun: So yeah, you have to reconstruct the whole thing. there's yeah. That's that's the hard part about deduce

142
00:38:10.190 --> 00:38:11.110
Jamie Boisvenue (He/Him): a paddy

143
00:38:13.800 --> 00:38:41.969
Patty Tessandori: got to unmute myself here. Yeah. So this is really really helpful and interesting. But I feel like there's 2 languages that both use the word coding going on, because to do seems to have this computer code stuff going on. And then, of course, there's the qualitative coding looking for themes, for the notes that we use for Ena. And so it's like, I. I understand a little bit about the E and a side. Just a little bit. I'm still new with this. I'm I'm a third year. Phd, student at Pepper 9.

144
00:38:42.040 --> 00:39:09.900
Patty Tessandori: But I. So I'm not so familiar with what you're doing with the with the computer coding that you you're using for it to do, and all of that. And I'm just wondering, how does one become proficient in these, or how? Obviously you're very proficient in these, but for those of us that might be interested in these tools. But this this really feels like a totally foreign language. How does one go about navigating to become more proficient in understanding these different tools that we have available to us?

145
00:39:10.290 --> 00:39:15.539
Jonathon Sun: Yeah, that's a great question. And I will say, like I was where you were, maybe like

146
00:39:15.700 --> 00:39:19.359
Jonathon Sun: 3 or 4 years ago. So It is very possible, like

147
00:39:19.580 --> 00:39:31.410
Jonathon Sun: I'm still not a great coder like I've taken a couple of classes. It's like, here's how you write a better code, and of course you can always write better code. I would say coding is very much like your writing process. it's just a very different

148
00:39:31.540 --> 00:39:35.649
Jonathon Sun: product that you're putting out there. And so to start getting familiar is like.

149
00:39:35.820 --> 00:39:50.440
Jonathon Sun: I think, just have fun with it, like like, ignore ignore any projects that you like, not ignore, but think about projects that you can have fun. With that you can get very tangible results from So, for example, like

150
00:39:50.670 --> 00:40:05.099
Jonathon Sun: this is going to be really bad, I I have a script in our that organizes all my finances, and it projects my for for what I should be doing, based on my credit card bills. And that project is nothing to do with academia, but that's just like

151
00:40:05.100 --> 00:40:26.560
Jonathon Sun: with to my wife like 3D. I don't want to have to do this every year. Let me just make an ourscript and so find a project that's fun that brings you joy. So you can get familiar with what you want to do, because, you know, most of the time when we're making when we're trying to organize our data trying to clean our data, we know exactly what we want, but we don't know how to get there, and finding a project that's fun helps you get to that getting there.

152
00:40:26.560 --> 00:40:45.070
Jonathon Sun: what happens after you find a project is the the fun part, which is just look, just unfortunately, you have to Google search it. And I think coding half of coding is, do I know what exactly do I need to search. And I'm not going to say, just like randomly, Google, search like, I will spend

153
00:40:45.380 --> 00:41:05.319
Jonathon Sun: 20 min, is trying to find the right thing to Google, search and knowing exactly what language, what words to use to get the exact result that you need is really, really difficult. Even learning words like tidy verse, deployer str split like I I there is functions in there that I'd been racking my head over for

154
00:41:05.450 --> 00:41:13.449
Jonathon Sun: days, and then I found a solution. It's like, Oh, it takes one line, it's fine. I didn't waste a couple of my days of my life.

155
00:41:13.670 --> 00:41:24.530
Jonathon Sun: So the patience. also. How also ask questions? if you do have questions, I'm more happy to help in like sort through some of this. The our coding stuff.

156
00:41:24.530 --> 00:41:36.460
Jonathon Sun: January is also helpful. I emailed her multiple times over the summer about how to use our, because sometimes things just weren't working. So have friends who are willing to help and have people who are willing to help.

157
00:41:36.460 --> 00:41:54.839
Jonathon Sun: So I I know that didn't answer like how you can get proficient. I would also say I took a couple of classes which really kicked my butt and said, you need to learn this now. those are not in the school of education, though those are all taken school of design for me, and that's that's really what got me to be a better coder, not the school of Ed.

158
00:41:54.840 --> 00:41:59.009
Jonathon Sun: But if you have, if your if your school of Ed has a strong coding department.

159
00:41:59.100 --> 00:42:03.199
Jonathon Sun: take courses from them and stick with them forever.

160
00:42:04.290 --> 00:42:07.330
Jamie Boisvenue (He/Him): Yeah, it's interesting that you said, you know

161
00:42:07.790 --> 00:42:11.730
Jamie Boisvenue (He/Him): learn by doing, but by doing something that you love.

162
00:42:11.810 --> 00:42:21.600
Jamie Boisvenue (He/Him): And so I think that's a pervasive issue. I. Most of the people in this chat, in this, in this call, who are in the space of education, are probably agreeing with you there that

163
00:42:21.610 --> 00:42:35.779
Jamie Boisvenue (He/Him): that you know, the greatest difficulty in learning is learning something that you're not interested in And so if you can apply some of these skills or develop these skills and areas, on topics that

164
00:42:35.780 --> 00:42:55.529
Jamie Boisvenue (He/Him): you're actually interested in, you're going to learn more. And it's gonna you're going to retain more information. I I just wanted to add also that I've actually been using Chat Gpt to help me code review and I I know that it's deteriorating in its quality and ability to to to provide

165
00:42:56.010 --> 00:43:04.679
Jamie Boisvenue (He/Him): viable information or accurate information. so far it's been pretty helpful for me in terms of code review, which I actually think is quite a pervasive

166
00:43:04.760 --> 00:43:14.459
Jamie Boisvenue (He/Him): problem in academia. We, we write scripts for our research and analyze data and publish the results, and who reviewed the code

167
00:43:14.610 --> 00:43:33.029
Jamie Boisvenue (He/Him): who reviewed the scripts? how do we know that it's actually correct? So Jonathan's right in accessing people within the Qe community? who can help you review your code and get it ready for publication. because ultimately, that is a very important element. But yeah.

168
00:43:34.390 --> 00:43:48.789
Jamie Boisvenue (He/Him): I just wanna take a note in the chat. There's a couple of questions coming in as well, Mariah had asked. Can you say more about the iterative process for code selection? And what sort of qualitative research is this code selection process appropriate for?

169
00:43:49.490 --> 00:44:00.119
Jonathon Sun: Yeah. So I think a lot of this is assuming. So the assumption that I'm making is while I have red theory. I really want to know what my data is saying. for itself.

170
00:44:00.200 --> 00:44:03.450
Jonathon Sun: And that is where I start. So

171
00:44:04.150 --> 00:44:16.010
Jonathon Sun: with that. This is appropriate for projects where you want the data and themes to rise on its own. And so, while again, I can apply my codes, and that's me applying the codes.

172
00:44:16.830 --> 00:44:45.579
Jonathon Sun: I think there is. This is. This is now a personal opinion. But if I can get the data to literally speak for itself with the like. The code that I, the computer code that I've written. Then that is almost an untouched way of. I don't apply my ideas here. And so as you're iterating on this. That is, that's where you incorporate that idea. If I understand your question correctly, let the data speak for itself and see what comes up. Now, if you have something in mind.

173
00:44:45.580 --> 00:44:55.389
Jonathon Sun: you can still use the iterative process to be like, okay, how do I get this like? Is this? Is this the experience that I'm capturing is what I want to capture? but for coding

174
00:44:56.180 --> 00:45:04.439
Jonathon Sun: but for coding yes, qualitative coding. Just keep iterating through the process. and eventually you'll get to where you need to.

175
00:45:05.070 --> 00:45:08.879
Jonathon Sun: Yes. there are other kinds of grounded through and ethnography. Yes.

176
00:45:09.500 --> 00:45:13.550
Jamie Boisvenue (He/Him): One room also has a question. Go ahead. One.

177
00:45:13.570 --> 00:45:27.800
Yuanru Tan: Yes, thanks, Jamie. Jonathan, thanks for your presentation, and I have used to do like 4 or 5 years ago, and then I'm sure the 2 already evolved a lot. But I sort of remember they also had like

178
00:45:27.970 --> 00:45:29.270
Yuanru Tan: so

179
00:45:31.170 --> 00:45:38.399
Yuanru Tan: head map, and then some built in like diagrams. They show you like the relationship between toes.

180
00:45:38.450 --> 00:45:41.000
Yuanru Tan: And I was wondering, have you ever

181
00:45:41.110 --> 00:46:00.639
Yuanru Tan: use those to help you develop your hypotheses before you put data into DNA, because I know sometimes people before they conduct their yearly analysis. They might have different ways to develop their hypotheses like, read the data aligned by, or run some other preliminary analysis. I'm curious about your experience on that

182
00:46:00.840 --> 00:46:12.380
Jonathon Sun: right, the heat map and I. So this this paper also happened. when we had come up with ideas. So we are talking about it. And this is truly just like, I'm curious if this is going to work.

183
00:46:12.380 --> 00:46:37.390
Jonathon Sun: so like the paper that we had before this, we presented an era. And it was like the themes that popped up in this study were basically the same things that popped up in our study like beforehand. So this was very much supporting the themes that you'd already identified. So that was just conversation. So but after that I started looking into the deduce tools like, okay, can I try a different way of doing this? the heat map

184
00:46:37.490 --> 00:47:02.730
Jonathon Sun: was helpful, but I couldn't get that out to like when you put it into deduce. The pictures come out as Jpegs and Pngs. It's very much just stuck there. and I can't get out the raw data that I need to put it together. If I even got the raw data out. I still don't have the segmentations to put it back together. So like our deduce, I'm pretty sure, also runs on our

185
00:47:02.730 --> 00:47:09.539
Jonathon Sun: But a lot of the analyses that it does. It just doesn't tell you like it doesn't give you other ways to connect it back.

186
00:47:09.650 --> 00:47:23.939
Jonathon Sun: so yes, deduce was used to or look through some of the data. But I don't like some of their visualization. And I was like, let me just remake this for us, Dr. Doch, I can. I can make a more clear one where we can interact with it.

187
00:47:24.210 --> 00:47:29.769
Jonathon Sun: so to answer your question, yes, we did use some of the deduce tools? but

188
00:47:29.780 --> 00:47:42.670
Jonathon Sun: I wanted to make more clear crafts. So I made them on my own and then, because also, like, if I'm going to make these tools. I might as well have it talk with some of the data that we have in the past. Then I connected it back to our data set.

189
00:47:43.590 --> 00:47:44.390
Yuanru Tan: Thanks.

190
00:47:49.440 --> 00:47:50.330
Jamie Boisvenue (He/Him): David.

191
00:47:50.690 --> 00:48:02.030
David Williamson Shaffer: as a question. Yeah. So, Jonathan, thanks so much for well for doing this and presenting it. I I have heard many times that people have trouble getting their codes out of deduce

192
00:48:02.040 --> 00:48:12.909
David Williamson Shaffer: So I'm glad that you're able to do that. I have a I have a question about that. And then what then of a question about the process overall? So

193
00:48:13.020 --> 00:48:19.579
David Williamson Shaffer: in in getting codes out of deduce deduce itself doesn't have

194
00:48:19.630 --> 00:48:22.870
David Williamson Shaffer: segment. It doesn't segment the data. Am I correct in that?

195
00:48:23.080 --> 00:48:37.220
David Williamson Shaffer: As far as I'm aware, it does not segment your data.

196
00:48:37.290 --> 00:48:43.329
David Williamson Shaffer: And it's just putting out those those snippets. Am I correct about that?

197
00:48:43.460 --> 00:48:57.230
David Williamson Shaffer: What happens if one of the tech snippets that deduce produces overlap overlaps the boundary between 2 segments of the data as you were segmenting it originally. How did you handle that edge case?

198
00:48:57.300 --> 00:49:09.450
Jonathon Sun: So that edge case was because I separate it as like each segment as I separated each segment, basically. do we? We actually off the code?

199
00:49:11.230 --> 00:49:30.210
Jonathon Sun: I avoided that because I told the Coding team, make sure when you code, you include all the codes in that one segment. Do not code anything else, and don't try to overlap codes. So when you code, you can include more than one code in that the highlighted case.

200
00:49:30.460 --> 00:49:33.039
David Williamson Shaffer: How did you? But how did they know? Oh, so they

201
00:49:33.070 --> 00:49:39.600
David Williamson Shaffer: in in deduce they had the utterances, and they so you asked them not to code across the boundaries of an utterance.

202
00:49:40.040 --> 00:49:46.720
Jonathon Sun: Yes, I did my best. I'm going to assume that some cases may have been missed. I'm going to assume some case, Scott, miss, but

203
00:49:46.890 --> 00:49:58.890
David Williamson Shaffer: did their best to not double. Go. Got it? Well, double coding would would be fine for what I'm saying it's actually the overlapping that would cause the the overlapping

204
00:49:58.910 --> 00:50:06.389
David Williamson Shaffer: to know. So let's say you were coding at the sentence level, and somebody saw code, and they highlighted over 2 sentences.

205
00:50:06.720 --> 00:50:33.329
David Williamson Shaffer: Forget about how you might do it in our like. How would you actually want to think about that? So now I have. and I've highlighted 2 sentences. Each sentence, of course, is a separate line. Now in the data. Would you apply the code to both sentences? Is that sensible, or how would you think I think the way I would do it is, I would separate it at whatever utterance that we have, and then say for this utterance. This code applies here.

206

00:50:34.100 --> 00:50:48.099
David Williamson Shaffer: So in so each sentence would get coded because some portion of it was coded. Yes, okay, yeah. So that so interestingly, I think that suggests that it would be possible to make a kind of general purpose. Our script

207
00:50:48.100 --> 00:51:09.269
David Williamson Shaffer: for extracting data from deduce, forgetting for the moment about how you would pre in our but just the actual extraction, and I think that might be useful for a number of people who who use the deuce So I I'd be happy, or I'm sure Cody or somebody would be happy to chat with you offline, or about how to

208
00:51:09.340 --> 00:51:13.070
David Williamson Shaffer: how to do that. I think it would be a great contribution to the to the community.

209
00:51:13.080 --> 00:51:29.169
David Williamson Shaffer: along those lines. So so Mariah has a question here about the extent to which what you were doing, overlapped with what our en a. Already does in terms of creating the networks. In the first place. so

210
00:51:29.210 --> 00:51:37.510
David Williamson Shaffer: I want to say I'm missing something. It sounds like you are basically reconstructing the accumulation process that you and a uses

211
00:51:37.540 --> 00:52:03.109
David Williamson Shaffer: in the in the middle of your like before you got to the net before you got to the co-occurrences, when you, when you were constructing the co-occurrences that you then mapped with Sm, a. It seems like you're that's basically the same thing that our on our accumulation, or, you know, accumulation does my missing something. I think so. The the goal of the Rena, though, is, though not the Rna. The goal of the network analysis is to just help me pick themes.

212
00:52:03.200 --> 00:52:32.759
David Williamson Shaffer: So only looking at co-occurrences rather than trying to understand the data. So that's the only reason the code, like the network, exists is I just need a way to see what's going on. I I understood that I understand. Okay. no. The reason I'm asking is that I think some of what you, some of the code that you constructed. Clever though it is, I I think it may actually be overlapping with what the. So in in our Ena, you can just accumulate the data and not do anything more with it.

213
00:52:32.770 --> 00:52:47.250
David Williamson Shaffer: And that would get you basically the co-a, basically, it would build those co-occurrence matrices for you. So so that I could, I should figure out about that information.

214

00:52:47.250 --> 00:53:04.319
David Williamson Shaffer: Yeah. Well, so again, we we have Brendan or or I or you. One will be happy to show you that it's it's It's it's it's actually pretty straightforward. Okay. The other thing that I'll just mention is that yay, you wong. Who's on the call here?

215
00:53:04.330 --> 00:53:07.679
David Williamson Shaffer: developed a tool called priya

216
00:53:07.760 --> 00:53:23.159
David Williamson Shaffer: And I can let her tell you more about it. but basically what it's designed to do is to take a very large set of codes and figure out what the smaller subset of codes is within. It is that still basically preserves

217
00:53:23.160 --> 00:53:40.800
David Williamson Shaffer: the the most information in the network that although the large set of codes had yeah, yeah, you said she's happy to follow up. She's a little sick today. So I'm not gonna put on the spot and be on to be on zoom. But So there's all which is just to say, I think. that

218
00:53:41.150 --> 00:53:47.089
David Williamson Shaffer: there, there! This is one of them on a number of cases where I think there's a lot of kind of

219
00:53:47.820 --> 00:54:15.809
David Williamson Shaffer: cross-pollination in the Qe. Community. That would be very helpful because people have developed tools in one place or another, and then not everybody is aware of them. And obviously, that's something that we try and do in the conferences. But It strikes me that that this may be a sort of call for the community to do a better job of of sharing questions and problems. This is a known problem, I just and we don't know how to solve it. But I think it's something to keep our eyes on.

220
00:54:15.960 --> 00:54:39.740
Jonathon Sun: Yeah, no, I appreciate that. And this was like I was just overwhelmed with the codes, and I had not gone through like the the literature that exists as of right now. but no, I'm happy to like find other solutions to this is this is basically just a test run for dissertation work. Well, I I just just add on to that. I think even with that, what I like about this work, Jonathan, is that you're being explicit about how you're thinking about it and why.

221
00:54:39.990 --> 00:55:06.270
Brendan Eagan: And that's a crucial piece of doing. Qe, well is like you're laying out. This is what I'm trying to get at. Here's here's how it relates to what I think is going on in practice. And then here's how I operationalize that. So even if some of the stuff is like you could do this in simpler ways, there's a lot of other stuff that hasn't been built up, like David mentioned, to start off with. Like people use the juice all the time. But there's a ho! It's I, all of that. You what you did you have to do? I think, to be able to get things where it needs to be to go to E and A,

222
00:55:06.270 --> 00:55:29.299
Brendan Eagan: and I also think just generally people have a lot of questions about, how do I think about looking at themes and different codes? And how do I select them? So I I think there's a a number of good contributions you've made, and and so that it's and having this form as a space to kind of share that out and then say, Okay, I mean, this is, I think, a known issue, as people are bringing like adopting these things and bringing in their own tools and and bringing in their own practice.

223
00:55:29.530 --> 00:55:44.019
Brendan Eagan: it's gonna look different for everybody in their own circumstance. So I think it's R. This has been a really great use case to show people what you're thinking and why and and how you're implementing it. And I think it's going to be. It's going to be useful for a lot of people going forward. So thank you for sharing it.

224
00:55:44.970 --> 00:55:49.069
David Williamson Shaffer: I should also add, I I I'm building on that. I really like the fact that

225
00:55:50.090 --> 00:56:04.859
David Williamson Shaffer: doing all this in our writing everything out as a function actually forces you to be really explicit about your assumptions in a way that even just throwing something into the Rena package wouldn't. And that's also a critical part of well, any good research. But certainly. Qi.

226
00:56:07.020 --> 00:56:30.019
Jonathon Sun: yeah, no, I appreciate that. Yeah, I was frantically typing before Ted. But I'm glad I'm glad it worked out, and I'm glad that there's other work that's going on. I'm happy to be to connect with some folks after this. so I don't have to work as hard in the future. But I believe Maria has one last question of this occurs. Code time for for one more question. Then we'll wrap it up.

227
00:56:30.280 --> 00:56:58.149
Jonathon Sun: Riley. Only case the reason why I did this code occurs often. I think you can do this in any other way. But our research team was talking a lot about like this thing popping up. So it's like, okay, what's the easiest way to figure out what's propping up the most often? so that was just simply from our our conversations as a research team. but yeah, that was that's based on just what's going on with our research team rather than like any deeper question.

228
00:57:00.220 --> 00:57:24.460
Jamie Boisvenue (He/Him): So thank you. Everyone for joining us today. And thank you, Jonathan, for presenting just a couple of housekeeping issue housekeeping things before you depart. August 20, fifth, which is fast approaching is the final date for your revisions for any submissions that you have to. Icqe 23 don't forget also that the registration closes for Icqe. 23,

229
00:57:24.460 --> 00:57:31.930

Jamie Boisvenue (He/Him): on August the 20 fifth, as well at 1,159 Utc. 12. So if you are attending the conference in Melbourne.

230
00:57:32.390 --> 00:57:33.520
Jamie Boisvenue (He/Him): Australia.

231
00:57:33.720 --> 00:57:41.700
Jamie Boisvenue (He/Him): please do so 68 days away, and counting Thank you all for joining, and we will see you next month.