Persistence, Self-doubt, and Curiosity

—

Surveying Code Literacy in Digital Humanities

Elli Bleeker[1], Kaspar Beelen[2], Sally Chambers[3], Marijn Koolen[1],
Liliana Melgar-Estrada[4], Joris van Zundert[1]

[1]Huygens Institute [2]Alan Turing Institute [3]Ghent University [4]Utrecht University

Overview

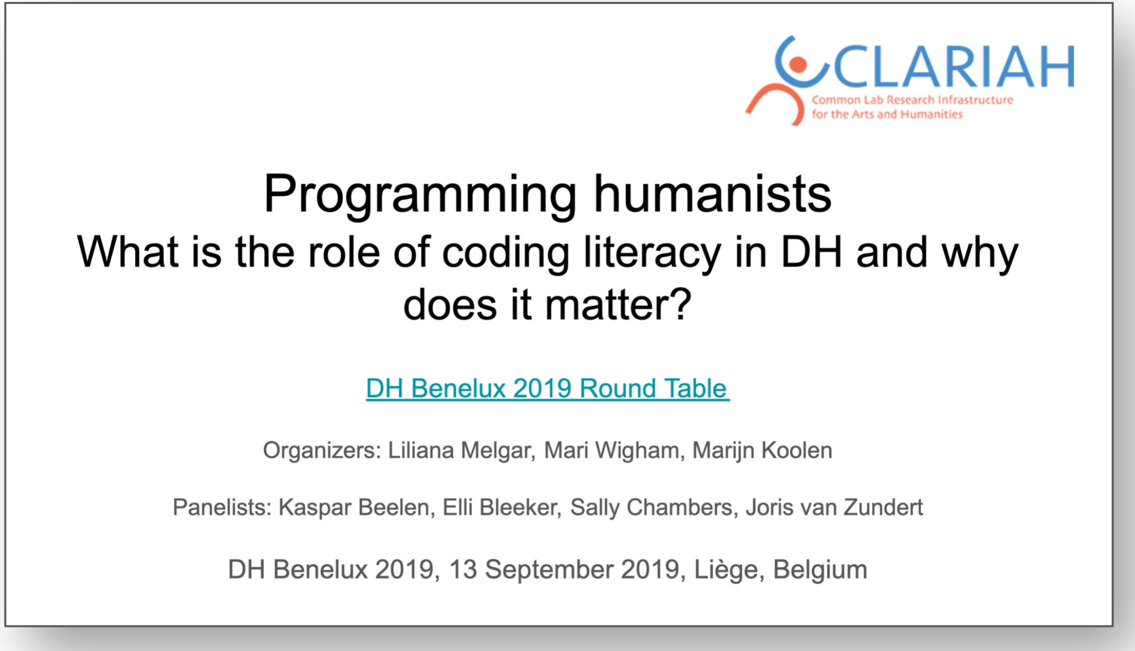
1. Motivation
2. Research questions & Methodology
3. Results
4. Conclusions
5. Future work

Slides at: http://bit.ly/dh-benelux-2021-code-literacy

(including some more details to read at your own leisure)

Motivation, Research questions & Methodology

Motivation



CLARIAH
Common Lab Research Infrastructure
for the Arts and Humanities

Programming humanists
What is the role of coding literacy in DH and why does it matter?

DH Benelux 2019 Round Table

Organizers: Liliana Melgar, Mari Wigham, Marijn Koolen

Panelists: Kaspar Beelen, Elli Bleeker, Sally Chambers, Joris van Zundert

DH Benelux 2019, 13 September 2019, Liège, Belgium

- Round Table at DHBenelux 2019 (September 2019)
- “We need to do better than anecdotal evidence”
- Collective effort in trying to understand how to improve “coding literacy” in the humanities, e.g.:
    - Aasman & Scagliola, 2017; Anderson & Ramey, 2019; Edmon & Garnett, 2015; Edmon et al., 2019; Gibbs, 2016; Hoekstra & Koolen, 2018; Kemman, 2019; Montfort, 2015; O’sullivan et al., 2015; Paling et al., 2010; Vee, 2017; Zundert & Haentjens Dekker, 2017; Zundert, 2019
    - Related surveys: DH RSE Survey (66 participants)

Research questions

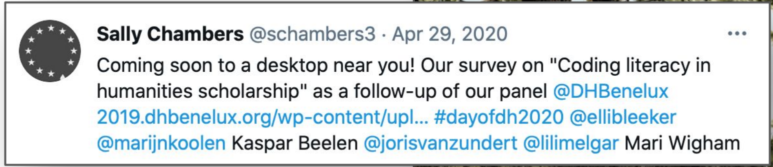
1. **What are the definitions and interpretations of code literacy across humanities disciplines?**

2. **How important is code literacy as part of digital humanities scholarship?**

3. How can we effectively approach the teaching and training of code literacy?

4. How can scholars (be supported to) incorporate code literacy in their research practice and methods?

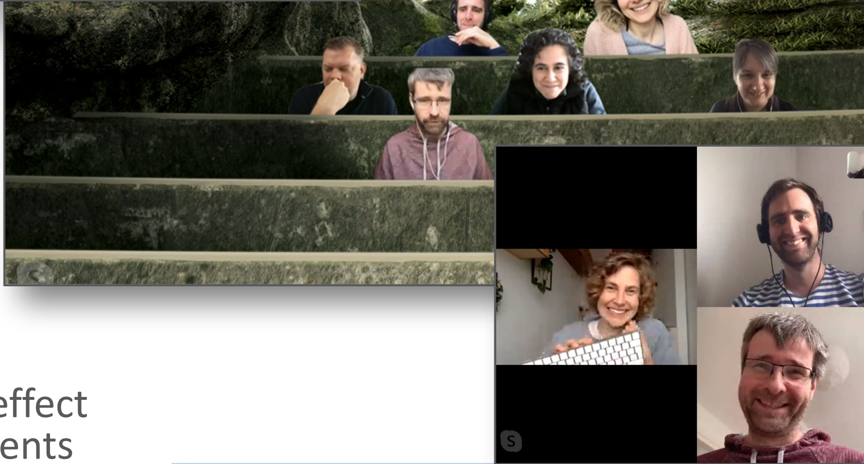
Methodology



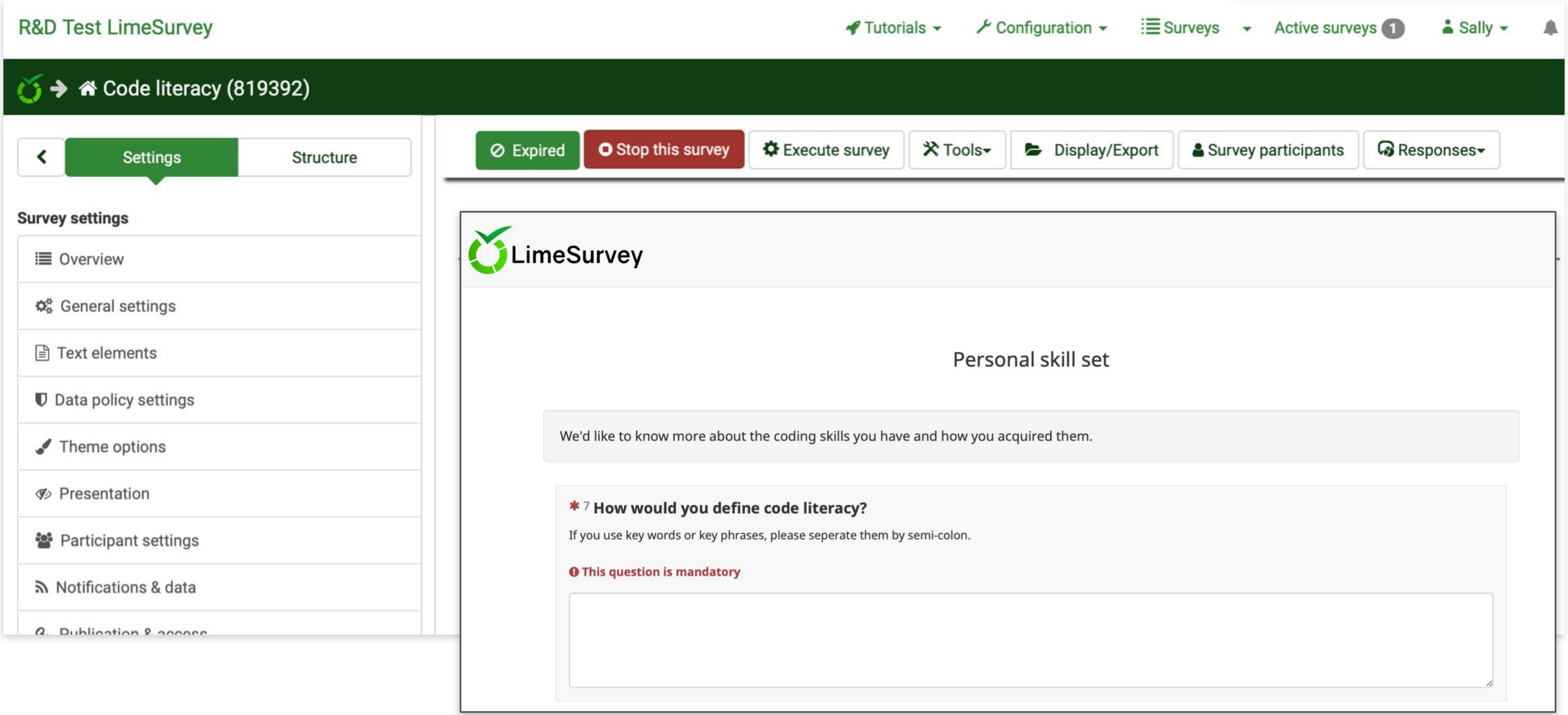
Sally Chambers @schambers3 · Apr 29, 2020

Coming soon to a desktop near you! Our survey on "Coding literacy in humanities scholarship" as a follow-up of our panel @DHBenelux 2019.dhbenelux.org/wp-content/upl... #dayofdh2020 @ellibleeker @marijnkoolen Kaspar Beelen @jorisvanzundert @lilimelgar Mari Wigham

- Designing the survey took about six months
  - From 25 March 2020, Twitter evidence
  - Regular meetings via approx. 27 Skype calls (38h) to refine questions
  - Often reflective work on questions
  - Post-positivist approach: mind of inevitable effect of our own biases (Ryan, 2006), e.g. respondents first asked to define code literacy and answer several questions with their own definition in mind
- Privacy: compliant to GDPR and the Utrecht University guidelines for handling personal data
- Survey pilot with ca. 8 respondents from a range of backgrounds (Humanities - PhD & PostDocs, Cultural Heritage Professionals, Information Science PostDoc)



LimeSurvey



Personal skill set

We'd like to know more about the coding skills you have and how you acquired them.

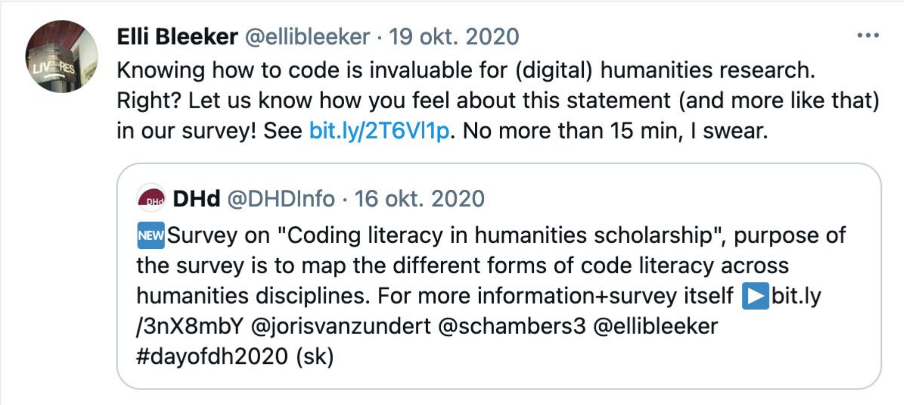
* 7 How would you define code literacy?

If you use key words or key phrases, please seperate them by semi-colon.

This question is mandatory

Survey

- Survey distributed via international (digital) humanities mailing lists, research institutes, universities, social media



Elli Bleeker @ellibleeker · 19 okt. 2020

Knowing how to code is invaluable for (digital) humanities research. Right? Let us know how you feel about this statement (and more like that) in our survey! See bit.ly/2T6Vl1p. No more than 15 min, I swear.

DHd @DHDInfo · 16 okt. 2020

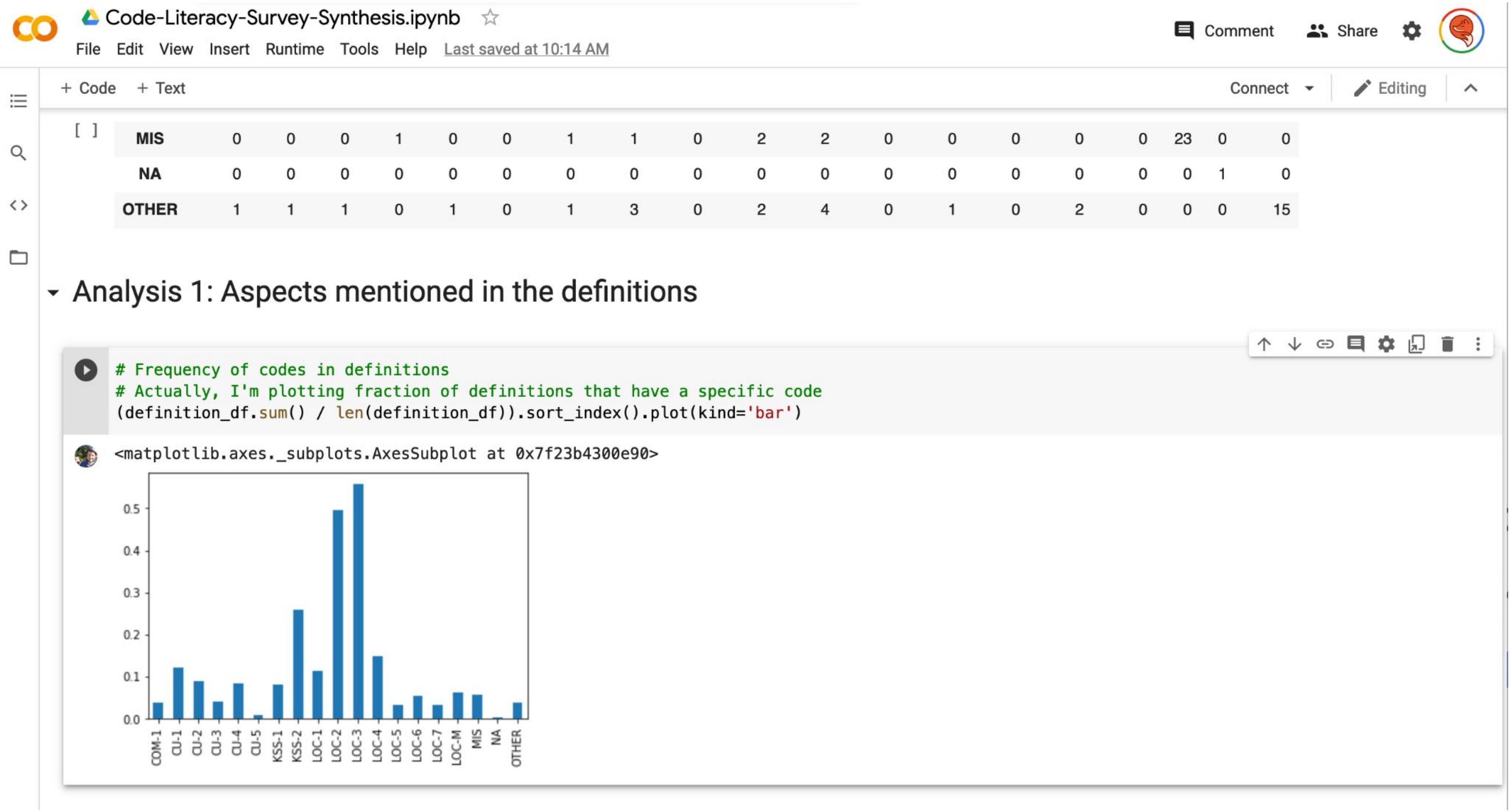
🆕Survey on "Coding literacy in humanities scholarship", purpose of the survey is to map the different forms of code literacy across humanities disciplines. For more information+survey itself ▶bit.ly/3nX8mbY @jorisvanzundert @schambers3 @ellibleeker #dayofdh2020 (sk)

- **Duration:** 15th October 2020 - 31st January 2021
- **Respondents:** 399 complete responses

Analysis - Qualitative data analysis

- Open coding and categorisation of initial codes applied to free text responses (Straus & Corbin, 1990)
- Merging of codes into one Axial List including: **main categories, sub-categories, scope notes**
- Recoding using resulting “code books”
- Inter-annotator agreement scores (cross-checking categorization between individual coders)
- Further discussion in wider group to consolidate views on tricky categorisations

Synthesis: Python notebooks shared internally via Google CoLab



MIS	0	0	0	1	0	0	1	1	0	2	2	0	0	0	0	0	23	0	0
NA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
OTHER	1	1	1	0	1	0	1	3	0	2	4	0	1	0	2	0	0	0	15

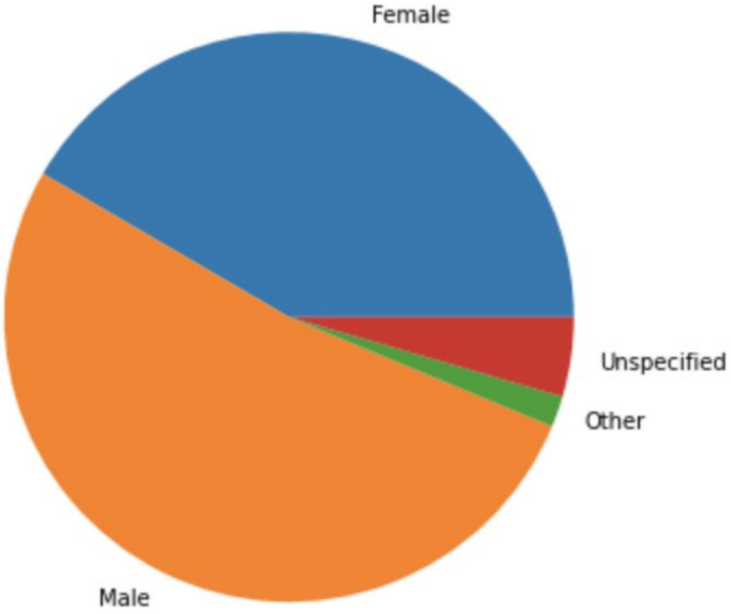
Analysis 1: Aspects mentioned in the definitions

```
# Frequency of codes in definitions
# Actually, I'm plotting fraction of definitions that have a specific code
(definition_df.sum() / len(definition_df)).sort_index().plot(kind='bar')
```

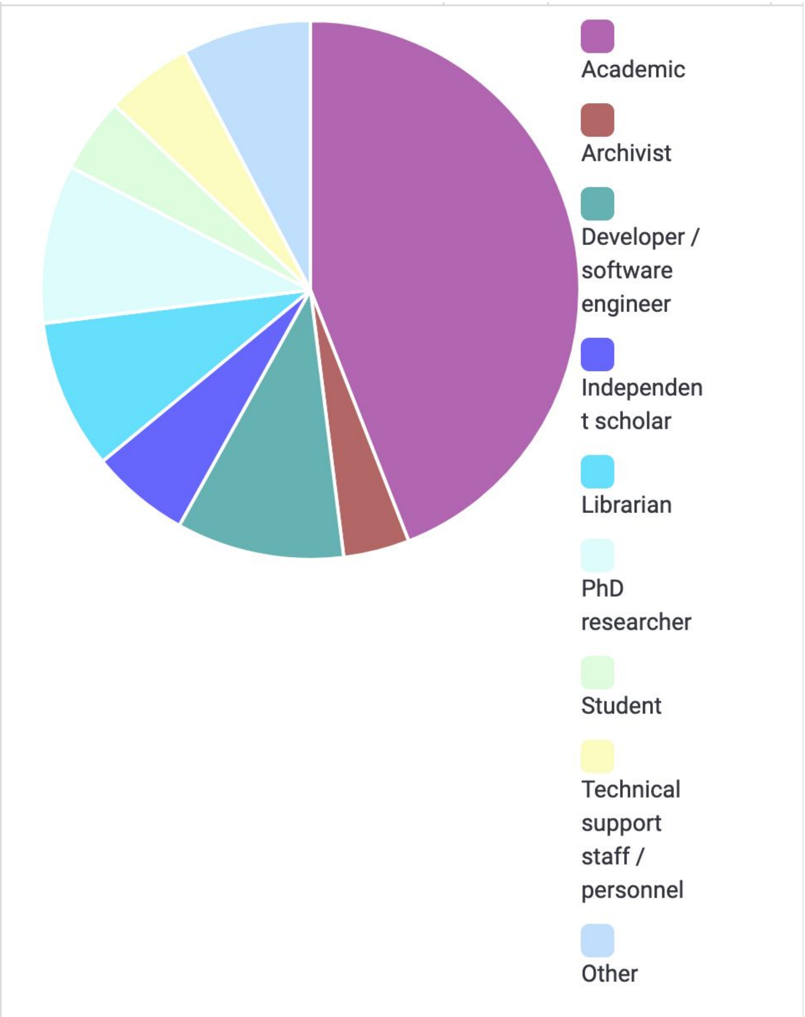
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f23b4300e90>
```

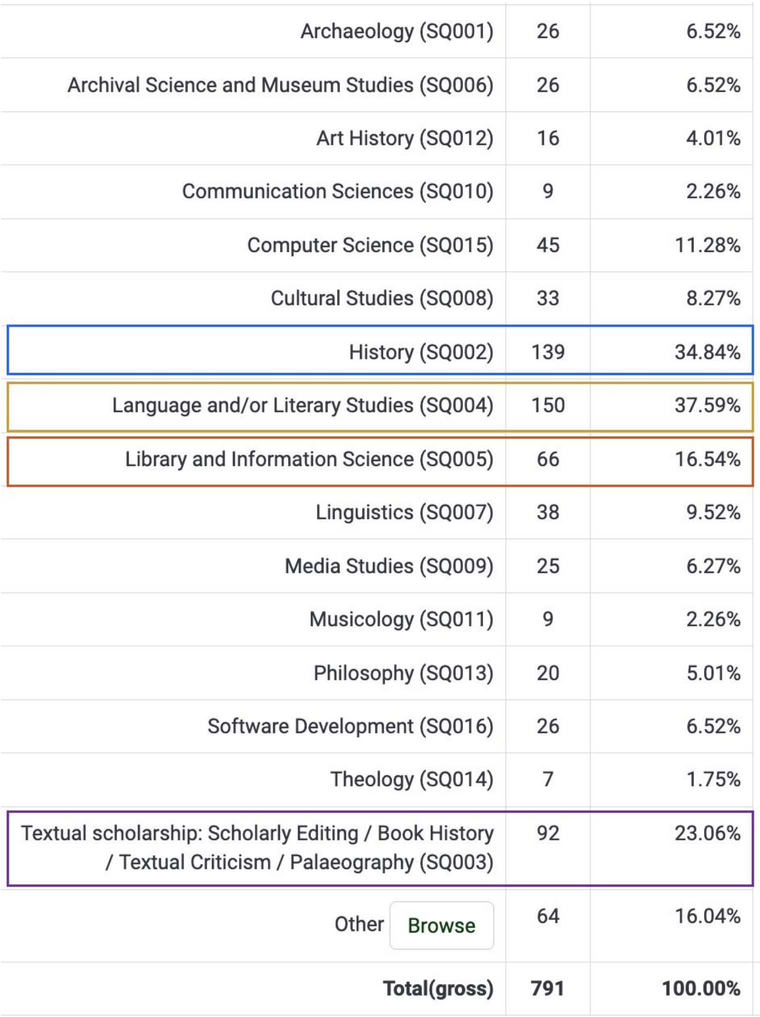
Results (preliminary)

Demographics - Gender



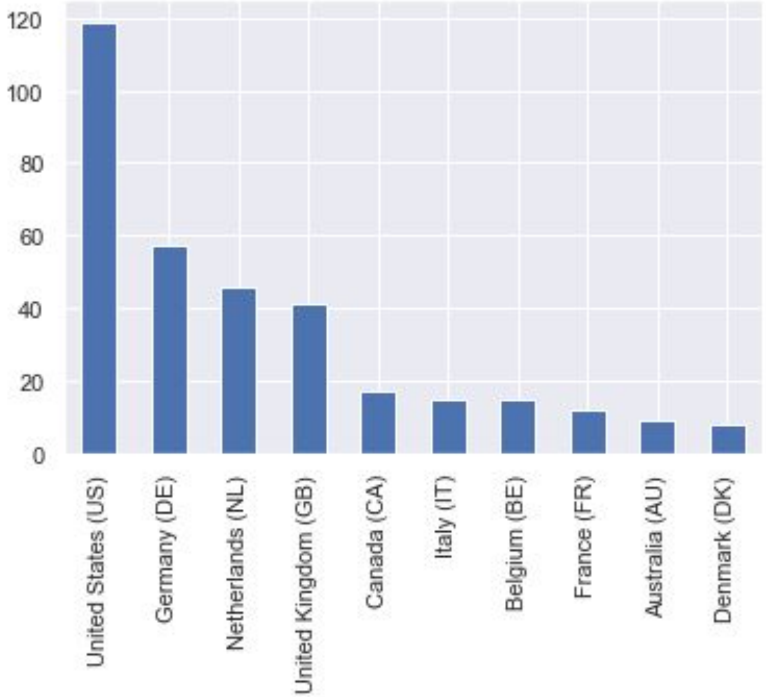
Example of “Other” category: “I’m a nasty woman and proud of it”



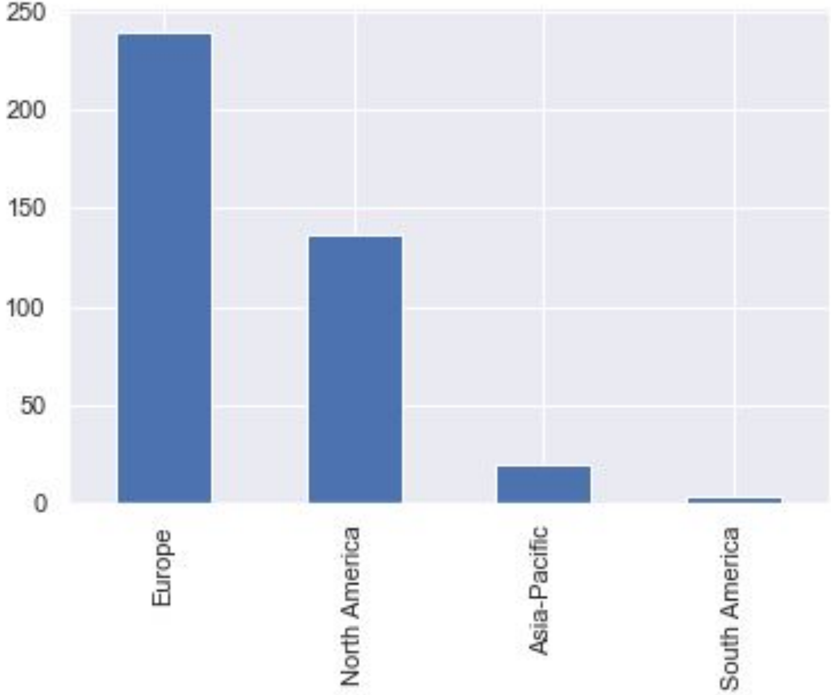


Archaeology (SQ001)	26	6.52%
Archival Science and Museum Studies (SQ006)	26	6.52%
Art History (SQ012)	16	4.01%
Communication Sciences (SQ010)	9	2.26%
Computer Science (SQ015)	45	11.28%
Cultural Studies (SQ008)	33	8.27%
History (SQ002)	139	34.84%
Language and/or Literary Studies (SQ004)	150	37.59%
Library and Information Science (SQ005)	66	16.54%
Linguistics (SQ007)	38	9.52%
Media Studies (SQ009)	25	6.27%
Musicology (SQ011)	9	2.26%
Philosophy (SQ013)	20	5.01%
Software Development (SQ016)	26	6.52%
Theology (SQ014)	7	1.75%
Textual scholarship: Scholarly Editing / Book History / Textual Criticism / Palaeography (SQ003)	92	23.06%
Other Browse	64	16.04%
Total(gross)	**791**	**100.00%**

Demographics - Geographical Distribution



Number of respondents by country (top 10)



Number of respondents by continent

RQ1. What are the definitions and interpretations of code literacy across humanities disciplines?

Survey question

“How would you define code literacy?”

- “The ability to understand and write code.”
- “On a scale from low literacy to high literacy (from ‘not’ to ‘expert’), I will define code literacy as the ability to read and understand code on a general level, to modify existing code written by others, to create new code using trial and error, to systematically create new code from an algorithm or other recipe or abstract plan, to create code that creates code, and to write compilers.”
- “The level of knowledge of and experience with coding; How much coding scares you off.”
- “It's a game of persistence, self-doubt, and curiosity.”

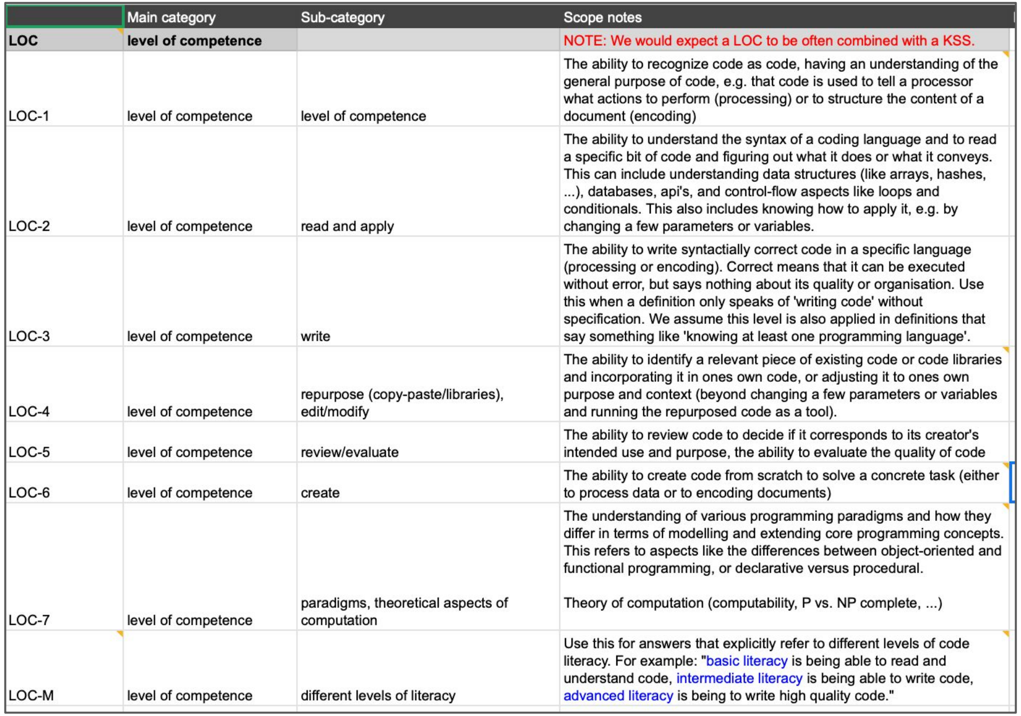
- We used qualitative data analysis on the 399 open answers.

Analysis - the Axial Code List

Main Categories:

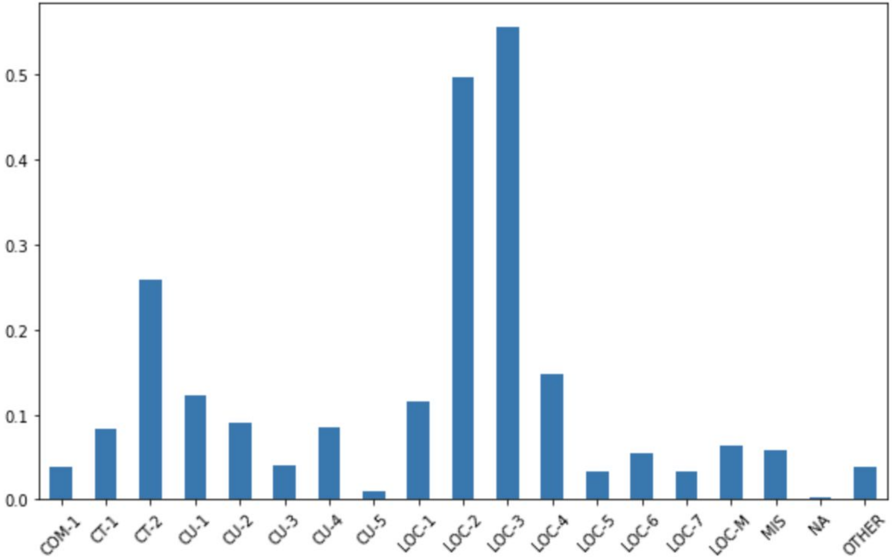
COM	Communication
CU	Contextual Understanding
LOC	Level of Competence
CT	Code Type (“encode” vs. “process”)
Other	e.g., misinterpretation of question

LOC - Level of competence codes



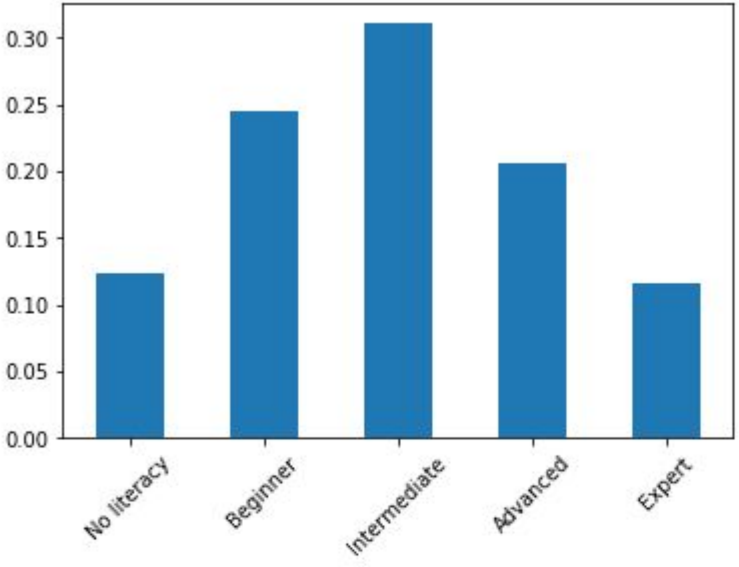
	Main category	Sub-category	Scope notes
LOC	**level of competence**		NOTE: We would expect a LOC to be often combined with a KSS.
LOC-1	level of competence	level of competence	The ability to recognize code as code, having an understanding of the general purpose of code, e.g. that code is used to tell a processor what actions to perform (processing) or to structure the content of a document (encoding)
LOC-2	level of competence	read and apply	The ability to understand the syntax of a coding language and to read a specific bit of code and figuring out what it does or what it conveys. This can include understanding data structures (like arrays, hashes, ...), databases, api's, and control-flow aspects like loops and conditionals. This also includes knowing how to apply it, e.g. by changing a few parameters or variables.
LOC-3	level of competence	write	The ability to write syntactially correct code in a specific language (processing or encoding). Correct means that it can be executed without error, but says nothing about its quality or organisation. Use this when a definition only speaks of 'writing code' without specification. We assume this level is also applied in definitions that say something like 'knowing at least one programming language'.
LOC-4	level of competence	repurpose (copy-paste/libraries), edit/modify	The ability to identify a relevant piece of existing code or code libraries and incorporating it in ones own code, or adjusting it to ones own purpose and context (beyond changing a few parameters or variables and running the repurposed code as a tool).
LOC-5	level of competence	review/evaluate	The ability to review code to decide if it corresponds to its creator's intended use and purpose, the ability to evaluate the quality of code
LOC-6	level of competence	create	The ability to create code from scratch to solve a concrete task (either to process data or to encoding documents)
LOC-7	level of competence	paradigms, theoretical aspects of computation	The understanding of various programming paradigms and how they differ in terms of modelling and extending core programming concepts. This refers to aspects like the differences between object-oriented and functional programming, or declarative versus procedural. Theory of computation (computability, P vs. NP complete, ...)
LOC-M	level of competence	different levels of literacy	Use this for answers that explicitly refer to different levels of code literacy. For example: "basic literacy is being able to read and understand code, intermediate literacy is being able to write code, advanced literacy is being to write high quality code."

Results: Frequency of codes in definitions



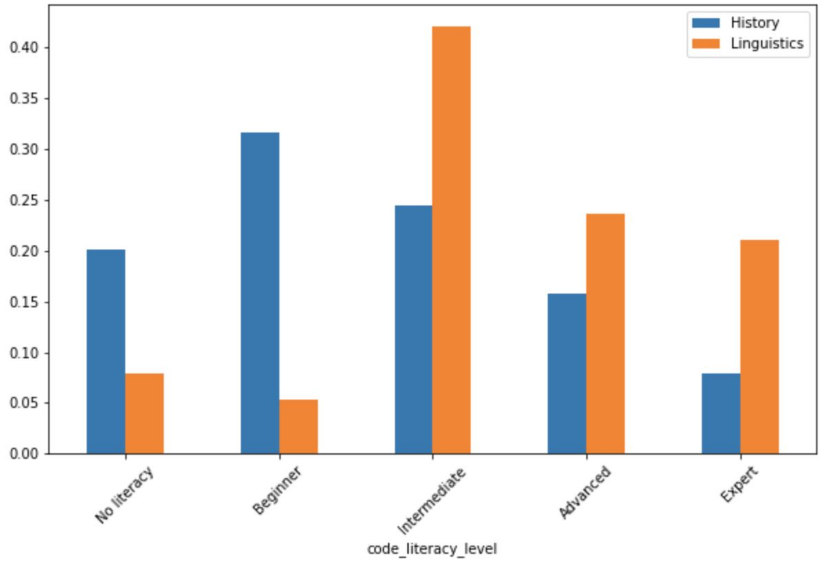
- 86% mention Level of competence
  - “Read” and “Write” code most common
  - 19% mention only “Read”, some only “Write”
- 28% mention “Contextual understanding”
  - Mainly more advanced coders
- “Code for processing” more common than “code for encoding”

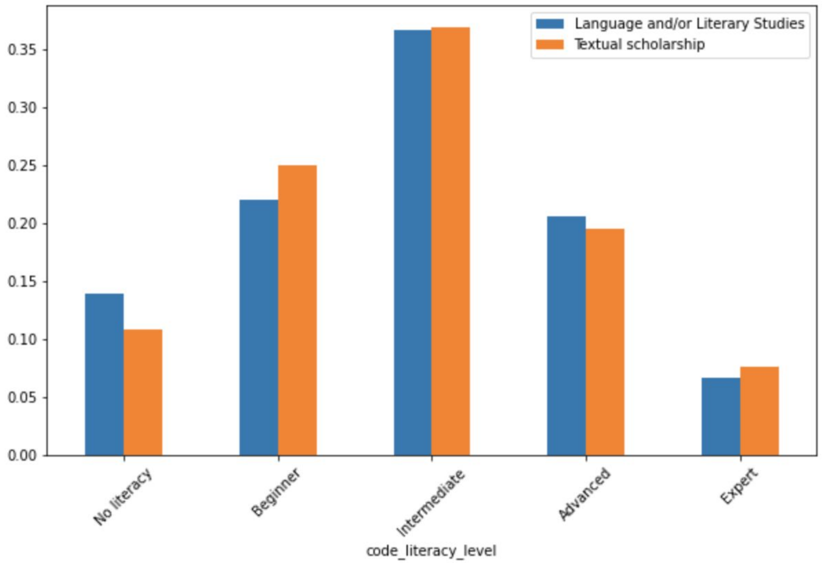
How Code Literate You Consider Yourself to Be?



- 19% are dissatisfied with their own level of code literacy
- 26% slightly dissatisfied
- 90% want to improve their code literacy
- Biggest hurdle is lack of time (58%)
- Almost all (94%) agree code literacy should be in DH curriculum

Code Literacy Level Across Disciplines

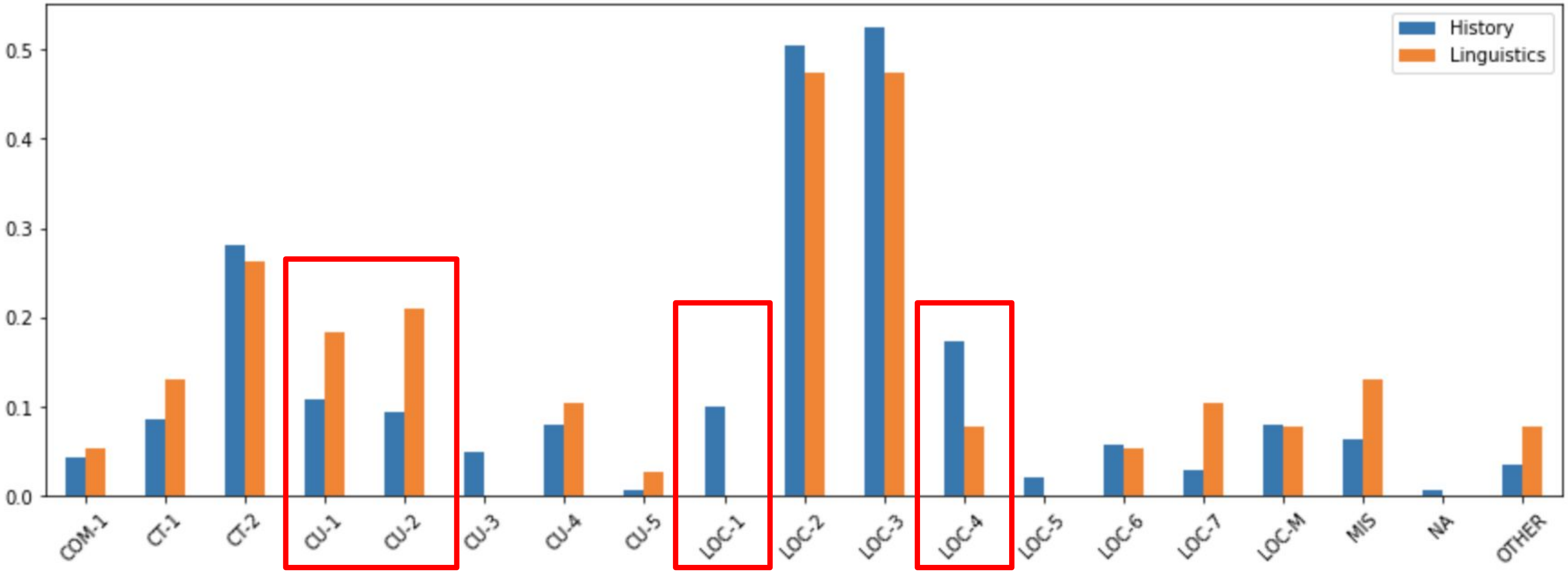




History (n=139)
Linguistics (n=38)

Language & Literary Studies (n=150)
Textual Scholarship (n=92)

History & Linguistics



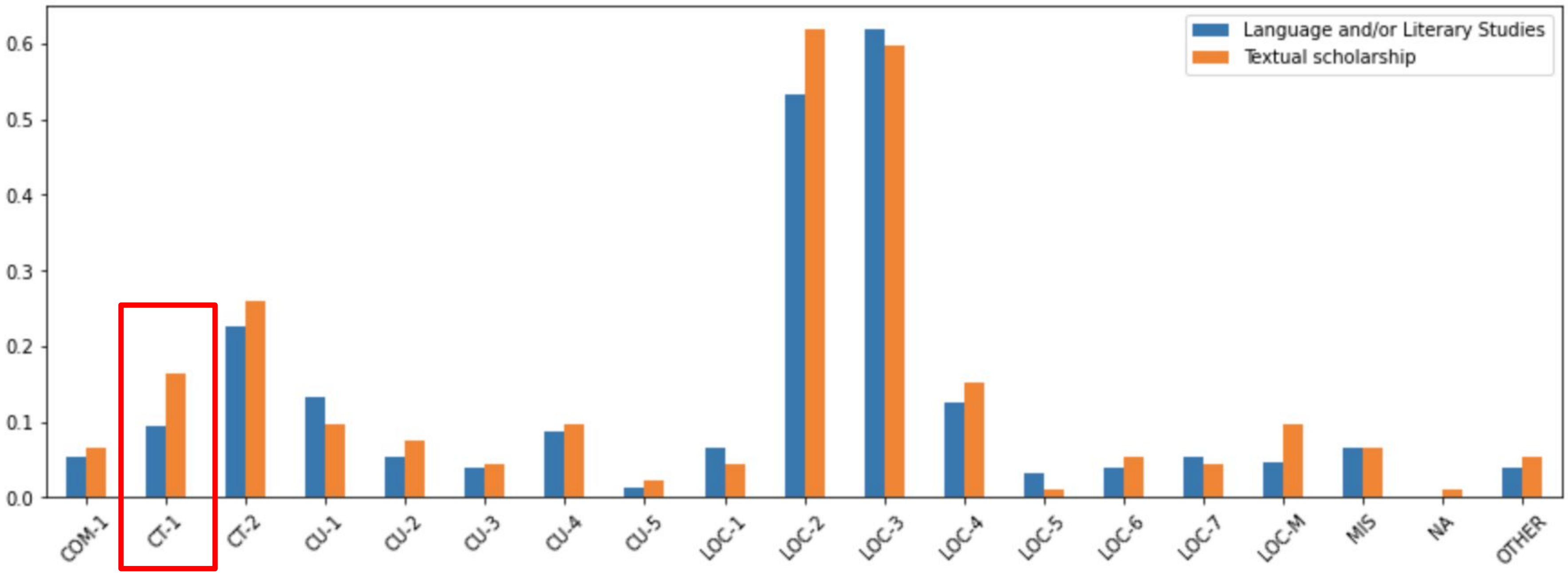
Level of competence

- LOC-1: Recognizing
- LOC-2: Reading
- LOC-3: Writing

Contextual Understanding

- CU-1: Translating RQ
- CU-2: Bias, limitations

Lang/Lit. Studies & Textual Scholarship



Level of competence

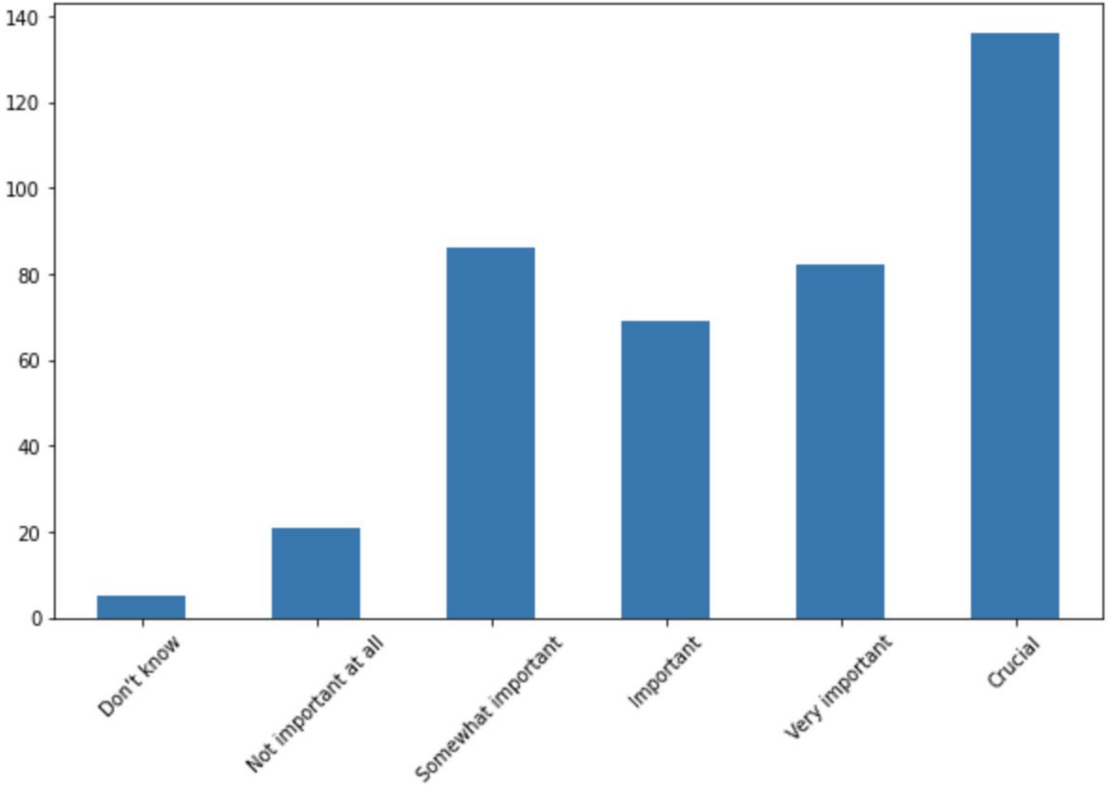
- LOC-1: Recognizing
- LOC-2: Reading
- LOC-3: Writing

Code Type

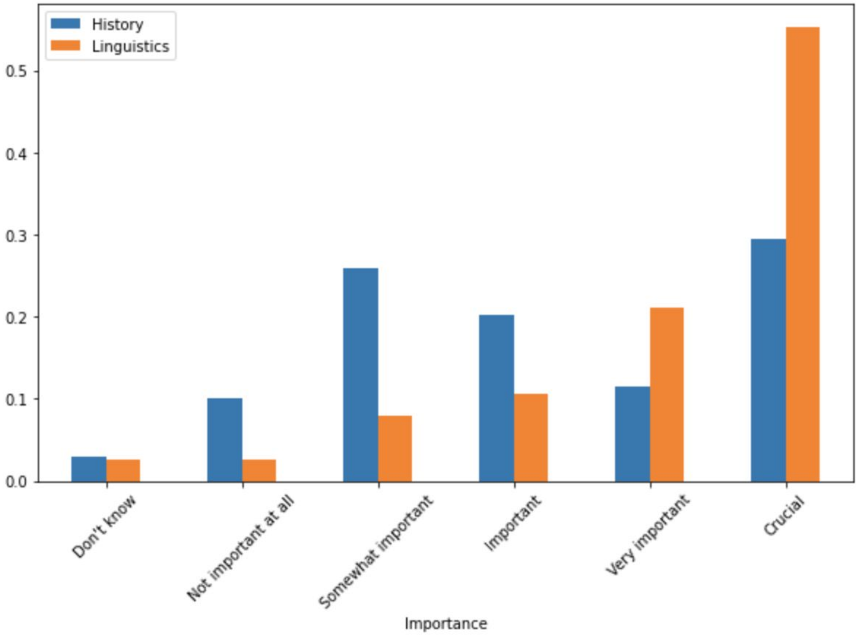
- CT-1: Encoding
- CT-2: Processing

RQ2. How important is code literacy as part of digital humanities scholarship?

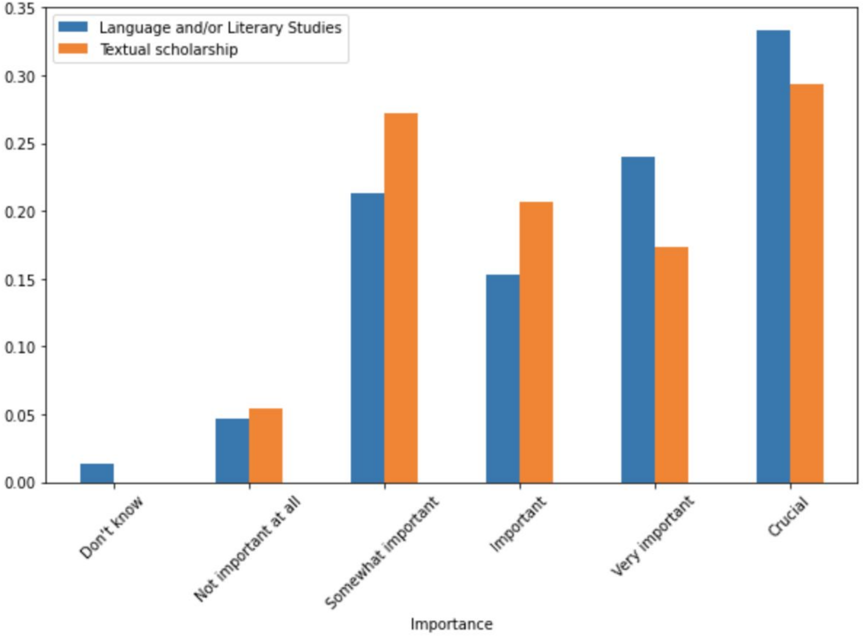
How Important Is Code Literacy in Your Work?



Importance of Code Literacy Across Disciplines



History (n=139)
Linguistics (n=38)



Language & Literary Studies (n=150)
Textual Scholarship (n=92)

Initial Conclusions

- Different disciplines emphasize different aspects of code literacy
- Community-inferred vocabulary to discuss code literacy
- Reading vs. writing code
    - Humanities-oriented: Read code first, before you start writing code?
        - Or intermingled …
        - Adapting code before writing from scratch?
    - CS-oriented: Reading and writing together
    - Is it similar to natural language?
        - Some disciplines may understand differently: is it relevant at all? How?

Limitations

- Selection effect
    - e.g., the respondent’s perception of their code literacy level may have influenced their participation
- Limited geographical representation
    - Respondents come mostly North-western Europe and US

Future work

Next steps and future research

- Wonderful dataset, but needs significant analysis
    - Finishing qualitative coding the “free text responses”
    - Finalise the synthesis
    - Some 15 additional questions to analyse
- Share the ‘aggregate dataset’ & the analysis notebook
- Follow-up Interviews
- Full research paper
- Recommendations / Contributions
    - Levels of Literacy
    - Learning Paths (and how)
- Guidelines for (reforming) DH curricula

References

Aasman, Susan, and Stef Scagliola. "Manifesto for Integrating DH in Teaching Programs." Digital Humanities Benelux Conference, 2017. https://dhbenelux2017.eu/2017/07/04/manifesto-for-integrating-dh-in-teaching-programs/.

Anderson, Clifford B., and Lynn T. Ramey. "Thinking Computationally in the Digital Humanities: Toward Block-Based Programming for Humanists." DataverseNL, December 17, 2019. https://dataverse.nl/dataset.xhtml?persistentId=hdl:10411/AY5YNH.

Edmond, Jennifer, and Vicky Garnett. "APIs and Researchers: The Emperor's New Clothes?" *International Journal of Digital Curation* 10, no. 1 (2015): 287–97. https://doi.org/10.2218/ijdc.v10i1.369.

Edmond, Jennifer, Vicky Garnett, Helen Goulis, Kristen Schuster, and Ulrike Wuttke. "Developing and Integrating Training Materials on Research Infrastructures into Higher Education Course Design: The PARTHENOS Experience." Utrecht, The Netherlands, 2019. https://dev.clariah.nl/files/dh2019/boa/0392.html.

Gibbs, Frederick W. "New Forms of History: Critiquing Data and Its Representations." *The American Historian*, February 2016. http://tah.oah.org/february-2016/new-forms-of-history-critiquing-data-and-its-representations/.

Hoekstra, Rik, and Marijn Koolen. "Data Scopes: Towards Transparent Data Research in Digital Humanities – DH2018." Mexico, 2018. https://dh2018.adho.org/en/data-scopes-towards-transparent-data-research-in-digital-humanities/.

Kemman, Max. Boundary Practices of Digital Humanities Collaborations. DH Benelux Journal vol. 1, 2019. https://journal.dhbenelux.org/journal/issues/001/Article-Kemman/kemman-main.tex.html

Montfort, Nick. "Exploratory Programming in Digital Humanities Pedagogy and Research." In *A New Companion to Digital Humanities*, 98–109. John Wiley & Sons, Ltd, 2015. https://doi.org/10.1002/9781118680605.ch7.

O'Sullivan, James, Diane Jakacki, and Mary Galvin. "Programming in the Digital Humanities." *Digital Scholarship in the Humanities* 30, no. suppl_1 (2015): i142–47. https://doi.org/10.1093/llc/fqv042.

Paling, S.a, S.b Buchanan, and Catherine Rita Larkin. "Integrating Informatics in Humanities Activities and Research." *Proceedings of the ASIST Annual Meeting* 47 (2010). https://doi.org/10.1002/meet.14504701077.

Strauss, A. L., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, Calif: Sage Publications.

Vee, Annette. *Coding Literacy: How Computer Programming Is Changing Writing*. MIT Press, 2017.

Zundert, Joris van, and Ronald Haentjens Dekker. "Code, Scholarship, and Criticism: When Is Code Scholarship and When Is It Not?" *Digital Scholarship in the Humanities* 32, no. suppl_1 (2017): i121–33. https://doi.org/10.1093/llc/fqx006.

Zundert, Joris van. "On Code Literacy." *The Fox and Badger* (blog), September 15, 2019. https://jorisvanzundert.net/blogposts/on-code-literacy/.

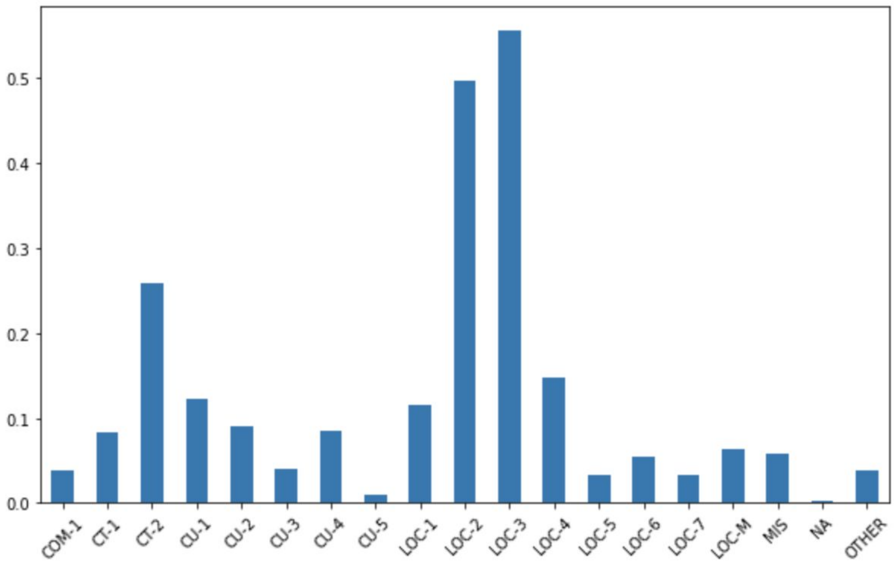
Thank you for your attention

- A **very big thank you** to all the respondents for their efforts, and their careful and considerate responses
- Slides at: http://bit.ly/dh-benelux-2021-code-literacy
  - including some more details to read at your own leisure

Towards a “Tiny Taxonomy of Computational Humanities Skills”

- Information and Data Literacy
- Collaboration
- Dissemination, peer review and teaching
- Project Management Skills
- Ethics
- Safety and Security
- Research Methods
- (Operative) knowledge regarding computers and hardware
- (Operative) knowledge regarding software, operating systems and applications
- Domain-specific computing skills
- Logic, math, and statistics
- Other

Results: Frequency of codes in definitions



Core aspects of code literacy: reading and writing code (but 19% mention only ability to read code)

Main Categories:

COM - Communication

CU - Contextual understanding

- CU-1: transforming research problems to computation
- CU-2: potential, limits, biases
- CU-3: attitude
- CU-4: code ecosystem (ethics, security, privacy, maintenance, documentation, versioning, licensing, good practices)
- CU-5: overall/vague about context

LOC - Level of Competence

- LOC-1: Recognizing
- LOC-2: Reading
- LOC-3: Writing
- LOC-4: repurpose (copy-paste/libraries), edit/modify
- LOC-5: review/evaluate
- LOC-6: create
- LOC-7: paradigms, theoretical aspects of computation
- LOC-M: different levels of literacy

CT - Code Type

- CT-1: encoding (XML, MPEG)
- CT-2: processing (performative code, e.g. Python, R, C, Java, JavaScript)

Other

- MIS: misinterpretation of question
- NA: empty/missing answer
- OTHER: aspect that doesn’t fit any other category

Geographical Distribution

Country	# Respondents	Continent	# Respondents
USA	119	Europe	240
Germany	57	North America	137
Netherlands	46	Asia-Pacific	19
UK	41	South America	3
Canada	17		