

1

Training Event Results

1.1 CEPEH Training Event C1

The CEPEH training event C1 held at the premises of University of Nottingham aiming to prepare participants for the practical elements of co-creation and implementation of chatbots as an educational resource. It combined both theoretical and hands-on training. 15 participants were from RISE, AUTH, UoN.

Project managers of partners signposted the person involved, and relevant announcements were made through social media channels to the wider public. External to the project speakers were from University of Leeds, and Computer Science Department of University of Nottingham. It included academics, medical doctors, and researchers with focus both on clinical research and digital innovations in healthcare education and IT specialist/learning technologists 11.18 years of experiences ($SD=7.2$). A balance between male and female participants achieved.

2

Overall Training Events Evaluation

Participants were asked to highlight what they liked for each day and how each day can be improved. Findings are described below per day of the training event

Day 1

The participants comment that they liked the design method for educational resources presented using a co-creation approach, they liked the interactions with other groups, and they liked the overview of existing chatbot resources of the partners. On the areas that can be improved, more media material were requested.

Day 2 Participants enjoyed the presentation from the invited speaker from another faculty of the University of Nottingham, the CEPEH resources presented and the storyboarding process. Participants highlighted that the participation of more clinicians in the event would be an added value in regards with the storyboarding process.

Day 3 Participants liked the hands-on activities of the day also enjoyed the creativity of the groups on the online chatbot development tool. As an area of improvement, participants wanted more time on hands on sections.

2.1 CEPEH Training Event 2

Pre-Training Event survey May 9th-13th 2022 Thessaloniki, Greece

Twenty-six participants attended the Training Event, along with approximately 10 staff members. There were 21 undergraduate students and 5 postgraduate students, who completed the survey for a total of 26 responses. There were 86% of participants who stated they had not been to a similar event like the training event CEPEH facilitated. There were 90% of students who found the event schedule very organised, and 70% agreed most of the planned sessions were relevant to that interest with the remaining 30% not having enough experience to understand the context to determine if they are interested in the training event. There were 95% of students agreeing or strongly agreeing the training event location is great, the remaining person did not leave additional comments.

Table 1 suggested attendees had minimal intention to share their own ideas due to lack of previous experience of attending such events, or due to lack of knowledge on the area. However, most were interested in listening to other groups and hearing contextual cases in healthcare.

There were 77% of participants stated they were novices in experience with chatbots in healthcare and were attending to learn more. The remaining 23% (7 students) stated they were competent and had limited experience with chatbots in healthcare.

One day had several events regarding cybersecurity in healthcare. When asked before these events, 83% stated they were neutral or disagreed that they felt confident about their cybersecurity knowledge in healthcare. In addition, 80% stated they when neutral or disagreed that they felt they had strong cybersecurity safety in healthcare. Table 2 shows the main pre and post results suggesting a positive experience for more than 75% of attendees on all measures.

There were 90% (23) of students who heard about the event through a lecturer or a professor, the CEPEH newsletter (2), and 1 person was informed through the anatomy tutoring system at Karolinska Institute. Additionally, 60% suggested the training event to somebody else before the course started.

There were six individuals who stated neutral or disagree when asked if having issues on registration or finding the information for the event. This may have been

due to being dependent on emails to receive the information, instead of a dedicated website where the information is available anytime.

As this was face-to-face, participants were asked about sufficient Covid-19 precautions in place at the facility, 94% agreed with sufficient precautions, two individuals stated no but did not give further information in the additional input box provided. In summary, most participants were undergraduate students with novice experience, happy with the training event location, felt the sessions were relevant to them, and most shared the event with their colleagues. The values of co-creation, chatbots in healthcare, and taking patient history were bestowed to students in an engaging and well-received manner. Notably, the highest ratings were for staff friendliness which is key to engagement and consistent interaction throughout the intense and long 5-day duration. The sessions were recorded there for the online recordings may be viewed with higher numbers over the subsequent weeks.

The usual way to include citations in an *R Markdown* document is to put references in a plain text file with the extension **.bib**, in **BibTeX** format.¹ Then reference the path to this file in **index.Rmd**'s YAML header with **bibliography: example.bib**.

Most reference managers can create a .bib file with you references automatically. However, the **by far** best reference manager to use with *R Markdown* is [Zotero](#) with the [Better BibTeX plug-in](#), because the **citr** plugin for RStudio (see below) can read references directly from your Zotero library!

Here is an example of an entry in a **.bib** file:

```
@article{Shea2014,  
  author =      {Shea, Nicholas and Boldt, Annika},  
  journal =      {Trends in Cognitive Sciences},  
  pages =        {186--193},  
  title =        {{Supra-personal cognitive control}},  
  volume =       {18},  
  year =         {2014},
```

¹The bibliography can be in other formats as well, including EndNote (**.enl**) and RIS (**.ris**), see rmarkdown.rstudio.com/authoring_bibliographies_and_citations.

```
doi = {10.1016/j.tics.2014.01.006},  
}
```

In this entry highlighted section, ‘Shea2014’ is the **citation identifier**. To default way to cite an entry in your text is with this syntax: `[@citation-identifier]`.

So I might cite some things `[@Shea2014; @Lottridge2012; @Mill1965]`.

2.1.1 Appearance of citations and references section (pandoc)

By default, `oxforddown` lets [Pandoc](#) handle how citations are inserted in your text and the references section. You can change the appearance of citations and references by specifying a CSL (Citation Style Language) file in the `cs1` metadata field of `index.Rmd`. By default, `oxforddown` by the American Psychological Association (7th Edition), which is an author-year format.

With this style, a number of variations on the citation syntax are useful to know:

- Put author names outside the parenthesis
 - This: `@Shea2014 says blah.`
 - Becomes: `@Shea2014 says blah.`
- Include only the citation-year (in parenthesis)
 - This: `Shea et al. says blah [-@Shea2014]`
 - Becomes: `Shea et al. says blah [-@Shea2014]`
- Add text and page or chapter references to the citation
 - This: `[see @Shea2014, pp. 33-35; also @Wu2016, ch. 1]`
 - Becomes: `Blah blah [see @Shea2014, pp. 33-35; also @Wu2016, ch. 1].`

If you want a numerical citation style instead, try `cs1: bibliography/transactions-on-comput` or just have a browse through the [Zotero Style Repository](#) and look for one you like. For convenience, you can set the line spacing and the space between the bibliographic entries in the reference section directly from the YAML header in `index.Rmd`.

If you prefer to use `biblatex` or `natbib` to handle references, see [this chapter](#).

2.1.2 Insert references easily with RStudio's Visual Editor

For an easy way to insert citations, use RStudio's [Visual Editor](#). Make sure you have the latest version of RStudio – the visual editor was originally really buggy, especially in relation to references, but as per v2022.02.0, it's great!

2.2 Cross-referencing

We can make cross-references to **sections** within our document, as well as to **figures** (images and plots) and **tables**.

The general cross-referencing syntax is `\@ref(label)`

2.2.1 Section references

Headers are automatically assigned a reference label, which is the text in lower caps separated by dashes. For example, `# My header` is automatically given the label `my-header`. So `# My header` can be referenced with `\@ref(my-section)`

Remember what we wrote in section ???

We can also use **hyperlink syntax** and add `#` before the label, though this is only guaranteed to work properly in HTML output:

- So if we write `Remember what we wrote up in [the previous section](#citations)?`
- It becomes `Remember what we wrote up in the previous section?`

Creating custom labels

It is a very good idea to create **custom labels** for our sections. This is because the automatically assigned labels will change when we change the titles of the sections - to avoid this, we can create the labels ourselves and leave them untouched if we change the section titles.

We create custom labels by adding `{#label}` after a header, e.g. `# My section {#my-label}`. See [our chapter title](#) for an example. That was section 1.



Figure 2.1: A marvel-lous meme

2.2.2 Figure (image and plot) references

- To refer to figures (i.e. images and plots) use the syntax `\@ref(fig:label)`
- **GOTCHA:** Figures and tables must have captions if you wish to cross-reference them.

Let's add an image:

```
knitr::include_graphics("figures/sample-content/captain.jpeg")
```

We refer to this image with `\@ref(fig:captain)`. So Figure 2.1 is [this image](#).

And in Figure ?? we saw a [cars](#) plot.

2.2.3 Table references

- To refer to tables use the syntax `\@ref(tab:label)`

Let's include a table:

Table 2.1: Stopping cars

speed	dist
4	2
4	10
7	4
7	22
8	16

```
knitr::kable(cars[1:5,],  
             caption="Stopping cars")
```

We refer to this table with `\@ref(tab:cars-table2)`. So Table 2.1 is [this table](#).

And in Table ?? we saw more or less [the same cars table](#).

2.2.4 Including page numbers

Finally, in the PDF output we might also want to include the page number of a reference, so that it's easy to find in physical printed output. LaTeX has a command for this, which looks like this: `\pageref{fig/tab:label}` (note: curly braces, not parentheses)

When we output to PDF, we can use raw LaTeX directly in our .Rmd files. So if we wanted to include the page of the cars plot we could write:

- This: Figure `\@ref(fig:cars-plot)` on page `\pageref(fig:cars-plot)`
- Becomes: Figure ?? on page ??

Include page numbers only in PDF output

A problem here is that LaTeX commands don't display in HTML output, so in the gitbook output we'd see simply "Figure ?? on page".

One way to get around this is to use inline R code to insert the text, and use an `ifelse` statement to check the output format and then insert the appropriate text.

- So this: ``r ifelse(knitr::is_latex_output(), "Figure \@ref(fig:cars-plot) on page \pageref{fig:cars-plot}", "")``

- Inserts this (check this on both PDF and gitbook): Figure ?? on page ??

Note that we need to escape the backslash with another backslash here to get the correct output.

2.3 Collaborative writing

Best practices for collaboration and change tracking when using R Markdown are still an open question. In the blog post [One year to dissertate](#) by Lucy D’Agostino, which I highly recommend, the author notes that she knits .Rmd files to a word document, then uses the `googledrive` R package to send this to Google Drive for comments / revisions from co-authors, then incorporates Google Drive suggestions *by hand* into the .Rmd source files. This is a bit clunky, and there are ongoing discussions among the *R Markdown* developers about what the best way is to handle collaborative writing (see [issue #1463](#) on GitHub, where [CriticMarkup](#) is among the suggestions).

For now, this is an open question in the community of R Markdown users. I often knit to a format that can easily be imported to Google Docs for comments, then go over suggested revisions and manually incorporate them back in to the .Rmd source files. For articles, I sometimes upload a near-final draft to [Overleaf](#), then collaboratively make final edits to the LaTeX file there. I suspect some great solution will be developed in the not-to-distant future, probably by the RStudio team.

2.4 Additional resources

- *R Markdown: The Definitive Guide* - <https://bookdown.org/yihui/rmarkdown/>
- *R for Data Science* - <https://r4ds.had.co.nz>