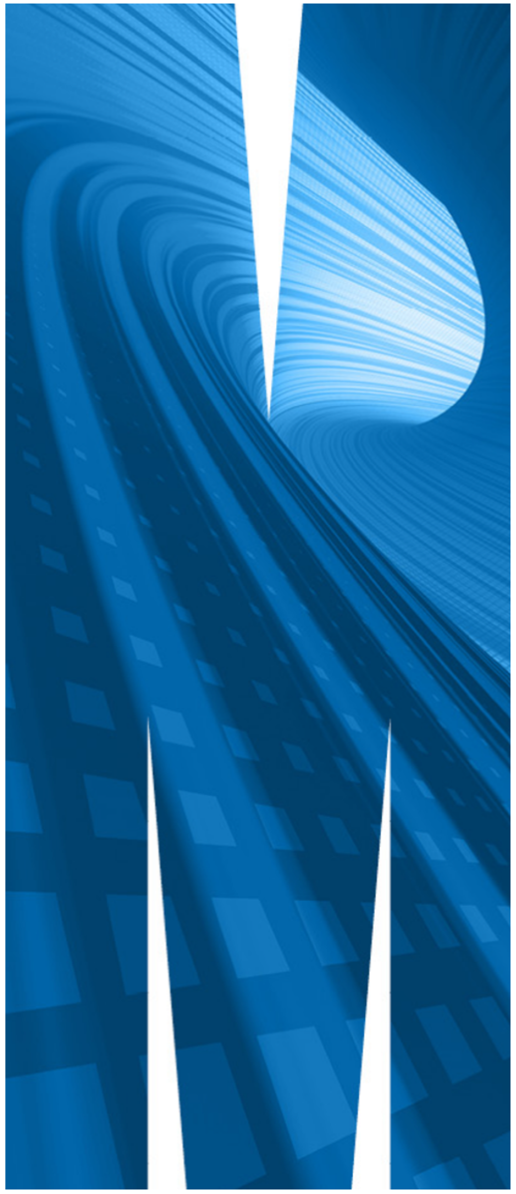
MONASH University

LLMs for Preparing Data in R

Guest lecture for ETC5512: Wild Caught Data

Cynthia A. Huang

19 May 2025

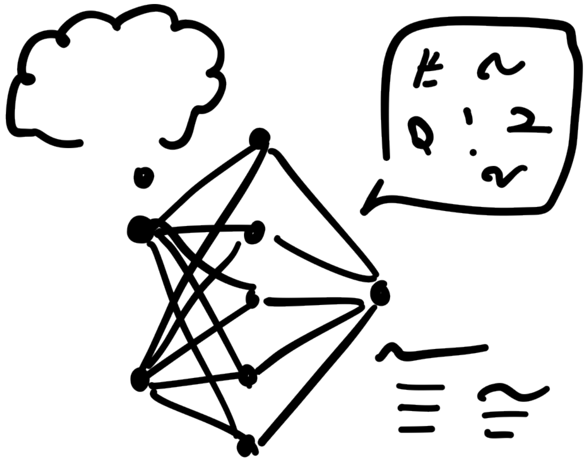


Introduction

Learning goals

- develop your understanding of:
  - what LLMs are,
  - different types of “wild caught data” tasks that LLMs can help with,
  - how to use and check LLMs for specific data prepration tasks,
  - how to interact with LLMs in R using {ellmer}

Lecture outline



OpenAI

ANTHROPIC

- About Me!
- About LLMs
- Using LLMs for “Wild Caught Data” tasks
- Using LLMs in R with `{ellmer}`

About Me!

Who am I?

- 👩‍🎓 Research Fellow in EBS supervised by *Prof. Rob Hyndman*
- 🎙️ Regular host on The Random Sample podcast
- 👩‍🏫 Ex-Tutor for Wild Caught Data (2020)

- 💱 Previously:
  - Economics at the University of Melbourne
  - Catching wild data for empirical researchers:
    - wikipedia entries, archival magazines, trade databases, satellite images, online retail prices…

What do I work on?

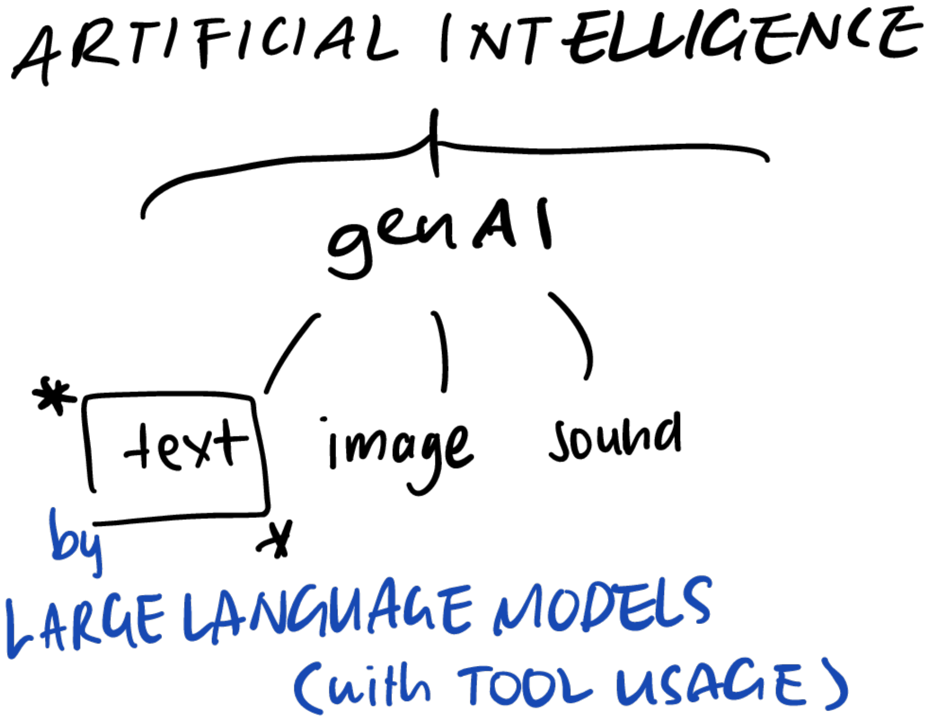
- 👩‍🎓 Thesis: **Unified Statistical Principles and Computational Tools for Data Harmonisation and Provenance**, supervised by:
  - *Prof. Rob Hyndman* (EBS, MBUS), *Prof. Simon Angus* (Econ, MBUS), and *Dr. Sarah Goodwin* (HCC, FIT)
- 📊 Research Interests
  - 🌰 Designing tools and workflows for wild caught data!
  - 🤖 Leveraging LLMs and genAI for data wrangling and cleaning
    - 🖇️ using LLMs to correct manual data entry errors – Current MBAT research internship!

About Large Language Models

Generative AI and LLMs

Generative AI refers to:

- computer algorithms and systems
- that can generate content such as text, images and sound
- based on patterns learnt from existing data

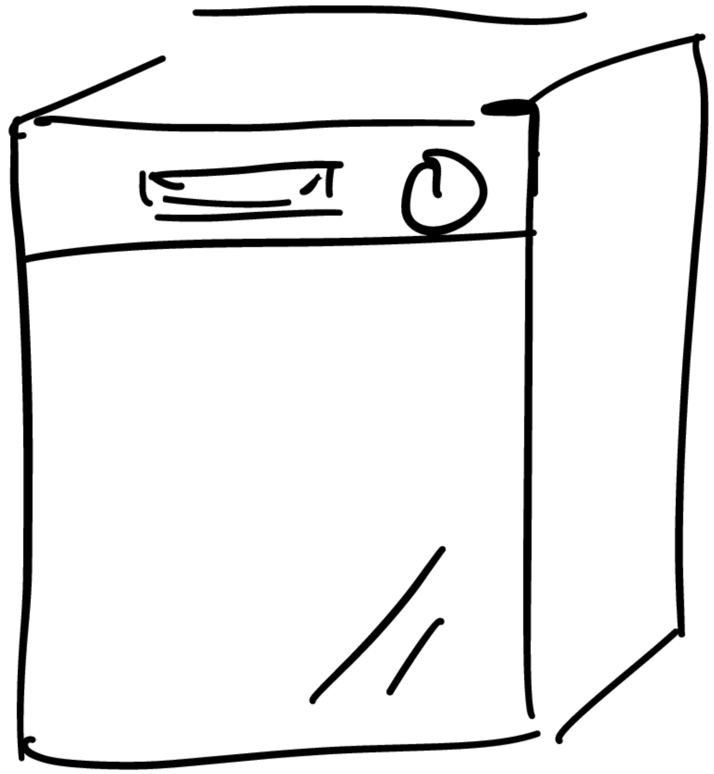


What are Large Language Models?

LLMs are…

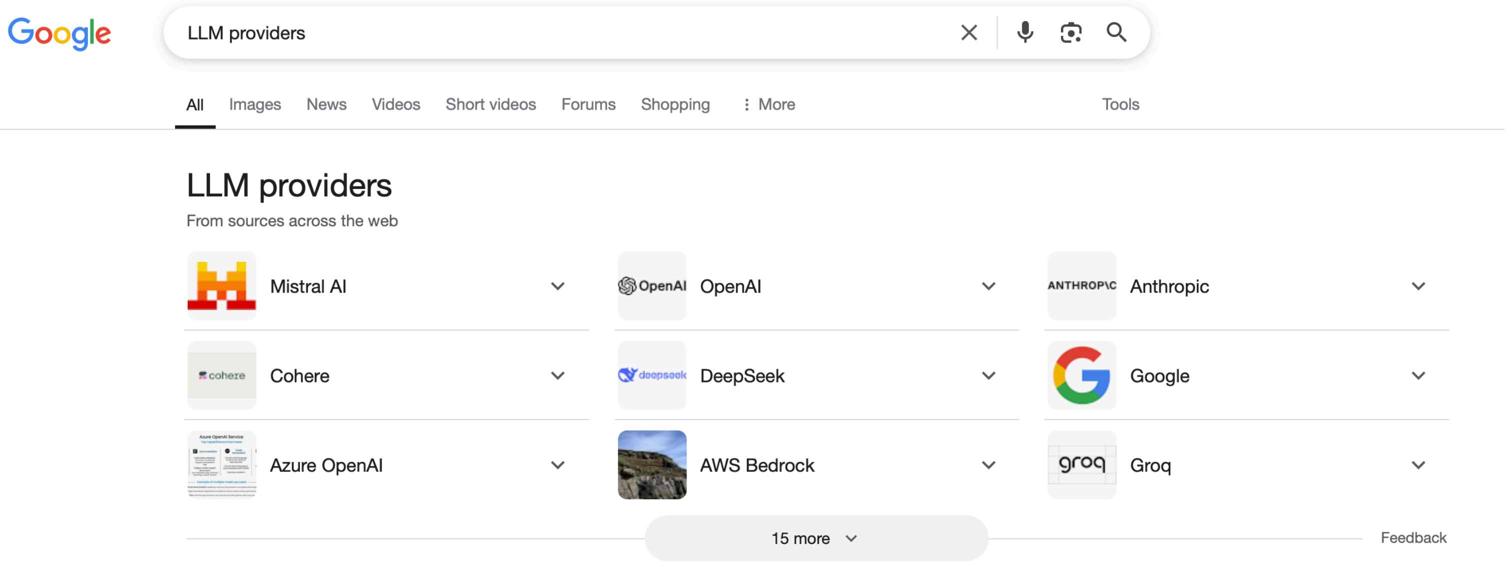
- code writers?
- encyclopedias?
- assignment help?
- translators?

We often understand tools by what they can do for us, not how they work.



Who makes LLMs?

LLM providers develop and offer access to large language models and systems



LLM providers

All Images News Videos Short videos Forums Shopping More Tools

LLM providers

From sources across the web

Mistral AI	OpenAI	Anthropic
Cohere	DeepSeek	Google
Azure OpenAI	AWS Bedrock	Groq

15 more

Feedback

Today, we will see demos of *Anthropic* and *OpenAI* models.

Why are there so many different models?

LLM providers offer paid and free access to multiple models:

- **OpenAI**: GPT-3 and 4, o-series,
- **Anthropic**: Claude Haiku, Sonnet and Opus
- **Google**: Gemini Flash and Pro
- **Meta**: Llama 3, Llama 4
- **Alibaba**: Gwen 2.5, 3, Max, Plus and Turbo

Different models are designed to be good at different things:

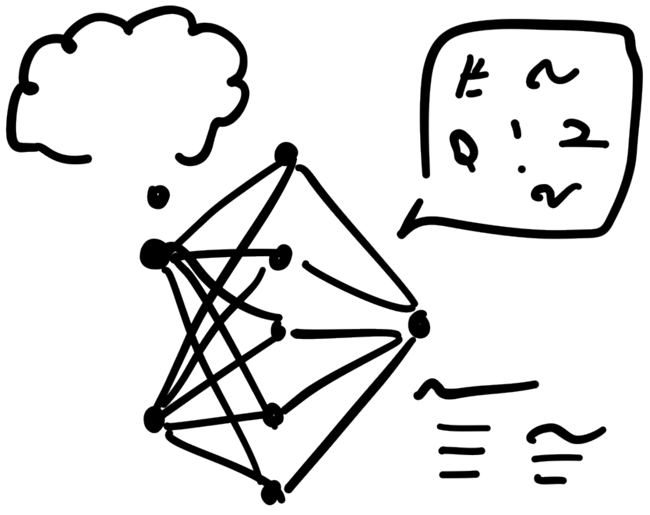
- chain-of-thought reasoning vs. instruction following
- multimodal support: images, audio, video AND text
- multilingual processng: translation, content generation
- specific domains: medicine, finance, legal

Model differentiation

Learn more about picking the right tool:

- 🎙️ Beyond ChatGPT: THE RAPIDLY EVOLVING LANDSCAPE OF AI
- 💡 Suggestions on provider/model choice in ‘ellmer’ docs
- 📖 OpenAI Model Selection Guide
- 📖 Anthropic Claude Model Comparison Table

How can we interact with LLMs?



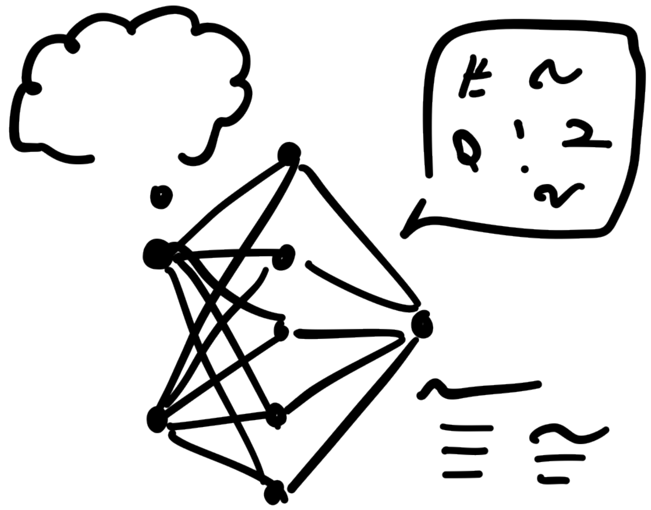
1. Web-based Chat Interface

- ChatGPT, Claude AI, Qwen Chat
- offers additional formatting of outputs, access to tools, other ‘quality of life’ feature

2. Programmatic Interfaces

- requires an API key
- interact using code with an LLM from within an R session

How can we interact with LLMs?



1. Web-based Chat Interface

2. Programmatic Access

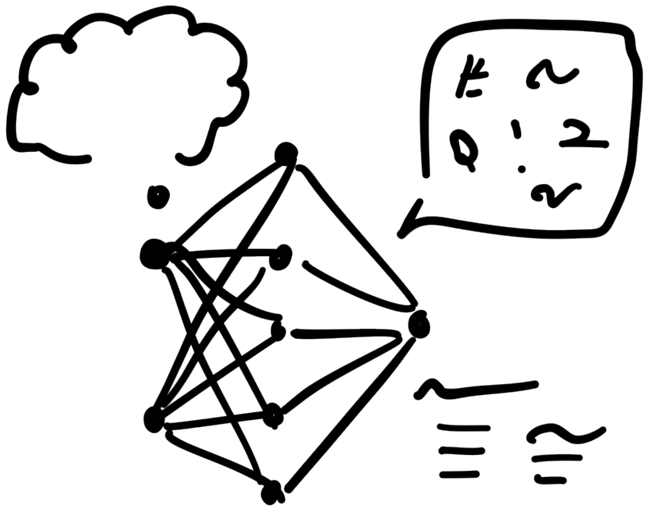
3. Other interfaces

- mobile (chat) applications
- voice assistants
- embedded LLMs (e.g. suggestions in Gmail

OpenAI

ANTHROPIC

How can we interact with LLMs?



OpenAI

ANTHROP\C

1. Web-based Chat Interface
2. Programmatic Access
3. Other interfaces

Today, we will use web-based chat via:

- chatgpt.com
- claude.ai,

as well as programmatic access via:

- `{ellmer}`

How do we verify what LLMs are doing?

For a dishwasher, we consider:

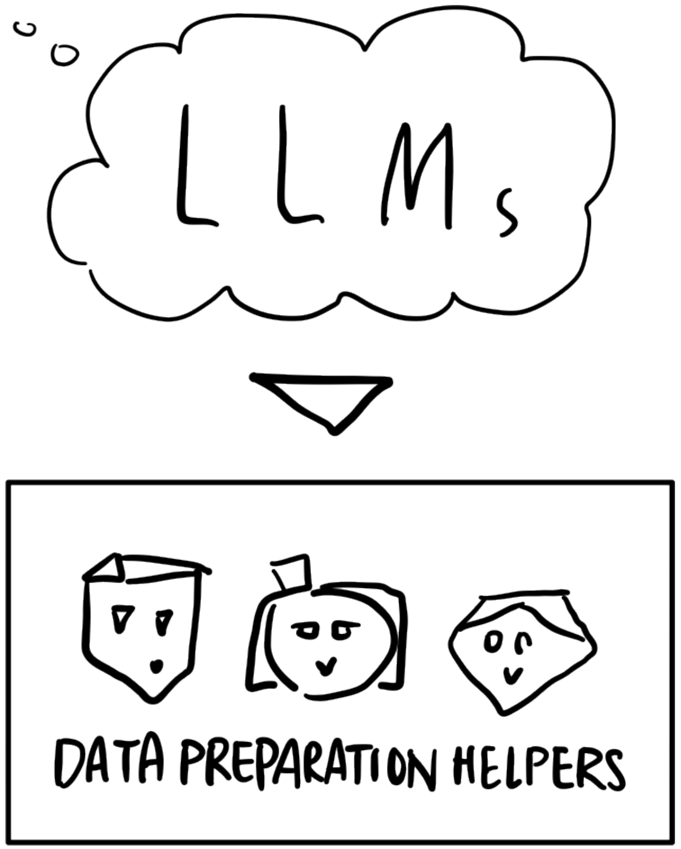
- **…are the dishes clean?**
- **is there any dirt on the dishes?**

For LLMs…?

It depends on the (data) task!

Using LLMs for “Wild Caught Data” Tasks

What are common WCD tasks?



Discuss in groups [5 mins]

- What tasks are involved in preparing Wild Caught Data for analysis? *List at least 3.*
- Which of these might be more or less suitable for addressing with LLMs? Why?

Using LLMs for Data Preparation Tasks

1. Generating data wranging code

- ‘indirect’ use of LLMs to clean data

2. Transformating and generating new data

- creating example datasets (e.g. when documenting a custom R function)
- changing data between formats (csv to json)

Using LLMs for Data Preparation Tasks

1. Generating data wranging code

2. Transformating and generating new data

3. Modifying and augmenting existing data

- filling in missing data
- correct typos or inconsistencies
- creating new columns based on existing ones

Modifying and augmenting data

- filling in missing data
  - **'look up' facts:** author nationality: `Jane Austen`
  - **suggesting values:** missing volume units for drinks: `300?`
- correct typos or inconsistencies
  - **harmonise different abbreviations:** `{Victoria, VIC, Vic}` –> `{VIC}`
- creating new columns based on existing ones
  - **comparison and categorisation:** are `teachers` and `instructors` similar occupations?
  - **summarise text:** key points in free-form survey responses
  - **extract info**: name of the movie in a film review

Prompts for Data Preparation Tasks

- prompts need to include:
  - instruction and data!
- responses would ideally:
  - return data of the expected type
  - and in a easy to import format

Try yourself! [10mins]

1. Pick a starting prompt from the next page
2. Fill in the necessary data.
3. Try the prompt in your choice of LLM chat (e.g. ChatGPT, Qwen Chat, Claude AI etc.).
   - What output did you get in return?
   - Could you import it into R easily?
4. Modify the prompt to return the answer in a more useful format.

Starting Prompts

- **'look up' facts:** "What nationality is *the author* `<author name>`?"
- **suggesting values:** "What is the likely volume unit *of a beverage* `<can>` *of with* a volume of `<300>`?"
- **harmonise different abbreviations:** "Convert the following list of Australian states to all use three-letter state codes: `<list>`"
- **comparison:** "How similar are these two occupations: `<occupation A>`, `<occupation B>`?"
- **summary:** "Summarise the following survey response: `<text>`"
- **extraction**: "What movie is the follow review about": `<review text>`

Requesting different output formats

- LLM can respond in many different ‘text’ formats.
- Some are more useful than others.

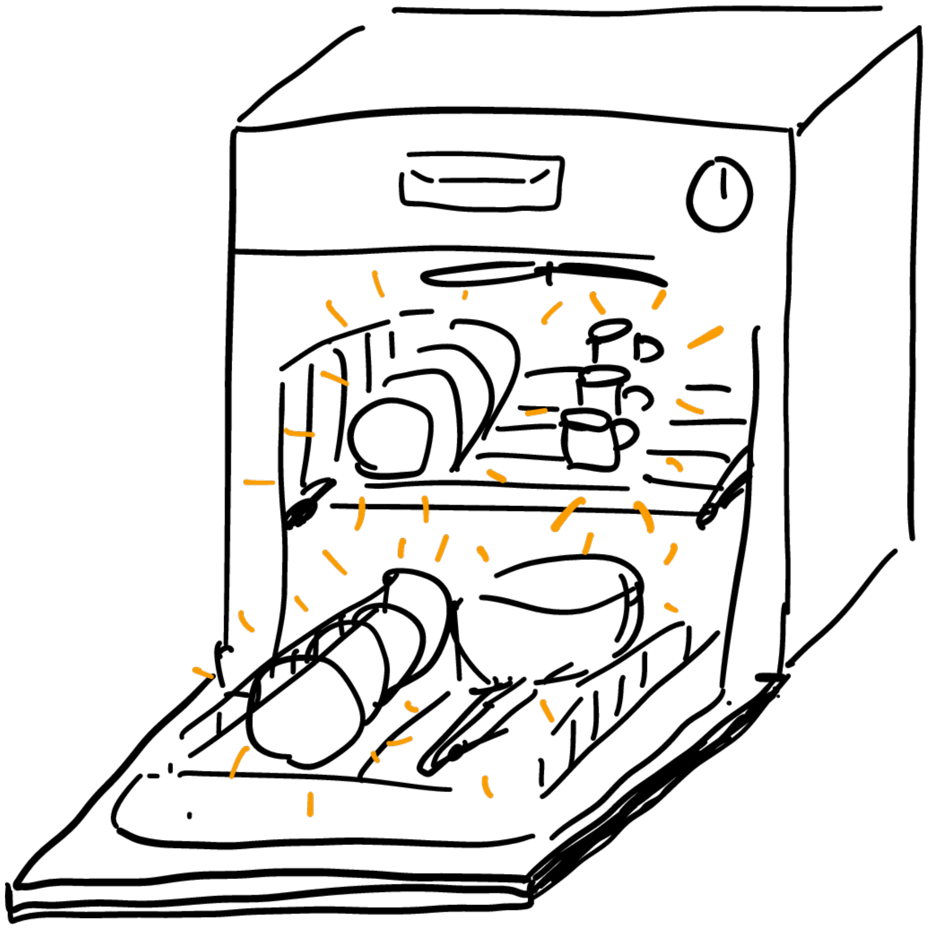
Let’s look at an example conversation with ChatGPT for the following prompt

```
Convert the following list of Australian states to all use three-letter state codes (e.g. VIC,
TAS):
- Victoria
- NSW
- N.T.
- ACT
- Queensland
```

Verifying success

Verification is the most important skill when using LLMs, but it requires:

- Clearly defined tasks and expected outcomes
- Ways of checking the outcomes have been achieved



Approaches to verification

There are multiple ways to verify outcomes match expectations.

- **Positive verification:** Define characteristics of 'success'
- **Negative verification:** Figure out signals or signs of 'failure'
- **Trust-based verification:** Seek assurance and confirmation of 'success'

Checking on the dishes

There are multiple ways to verify outcomes match expectations.

- **Positive verification:** Are the dishes clean?
- **Negative verification:** Is there any dirt on the dishes?
- **Trust-based verification:** Ask the machine if the dishes are clean…?

Using LLMs in R with {ellmer}

Beyond web-based interfaces

Different interfaces mean different data preparation workflows:

- LLM web-interface = copy/paste
- programmatic interfaces = code and variables

Using the {ellmer} R package we can:

- send prompts to that LLM from an R session
- construct prompts from with imported data
- systematically docutest different prompts BEFORE scaling up
- manipulate response content using code

Connecting {ellmer} to an LLM

The basic steps:

- Installing `{ellmer}`
- Getting an API key from the LLM provider you want to use > knock! knock! who's there? Cynthia's R session!
- Storing the API key where ellmer can find it
- Starting a chat session using the relevant `ellmer::chat_*()`

Today, we'll interact with the OpenAI API via `ellmer::chat_openai()`

LIVE DEMO: Chatting via {ellmer}

```r
library(ellmer)
## A session is like a chat conversation
session <- chat_openai()

question <- "
  How can I pick a random letter from A-Z.
"

## send a question to the 'chat'
session$chat(question)

## clarify your request
session$chat("Return R code only")

## inspect all turns in the session so far
session
```

What if we *always* want the LLM to return R code?

LIVE DEMO: System Prompts

```r
library(ellmer)

session_tidy_expert <- chat_openai(system_prompt = "
  You are an expert R programmer
  who prefers the tidyverse.
  Only return code without explanation.
")

session_tidy_expert$chat(question)

session_tidy_expert
```

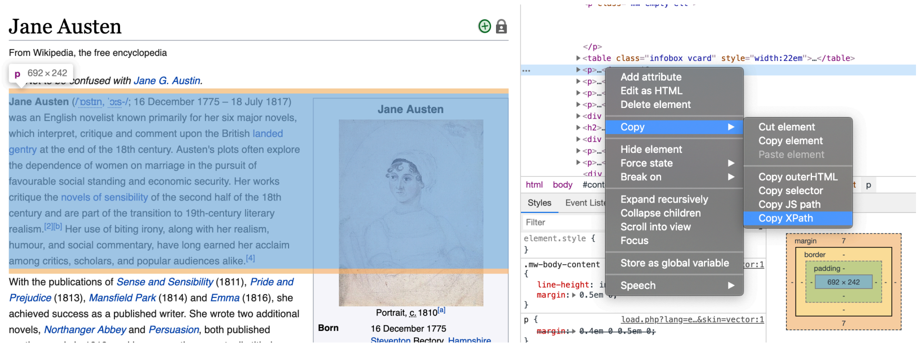
Example adapted from ellmer docs

Sessions and system prompts

- A **chat session** is a single conversation instance between a user and an LLM
- A **R session** is an active workspace where you’re running the R programming language
- A **system prompt** is the behind-the-scenes instruction manual that tells an AI assistant:
  - what tone to use,
  - what information the system can access,
  - and how to handle different types of questions or requests

Revisiting Author Nationalities

Could we use an LLM to extract Jane Austen's nationality?



Jane Austen

From Wikipedia, the free encyclopedia

Not to be confused with Jane G. Austin.

Jane Austen (/ˈɒstɪn, ˈɔːs-/; 16 December 1775 – 18 July 1817) was an English novelist known primarily for her six major novels, which interpret, critique and comment upon the British landed gentry at the end of the 18th century. Austen's plots often explore the dependence of women on marriage in the pursuit of favourable social standing and economic security. Her works critique the novels of sensibility of the second half of the 18th century and are part of the transition to 19th-century literary realism.[2][b] Her use of biting irony, along with her realism, humour, and social commentary, have long earned her acclaim among critics, scholars, and popular audiences alike.[4]

With the publications of *Sense and Sensibility* (1811), *Pride and Prejudice* (1813), *Mansfield Park* (1814) and *Emma* (1816), she achieved success as a published writer. She wrote two additional novels, *Northanger Abbey* and *Persuasion*, both published

Jane Austen	
Portrait, c. 1810[a]	
Born	16 December 1775
	Steventon Rectory, Hampshire

```
</p>
<table class="infobox vcard" style="width:22em">…</table>
<p>…
<p>…
<p>…
<p>…
<div
<h2>…
<div
<p>…
<p>…
<div
```

```
element.style {
}
.mw-body-content
{
  line-height: i
  margin: ▸ 0.5em 0;
}
p {                              load.php?lang=e…&skin=vector:1
  ~~margin: ▸ 0.4em 0 0.5em 0;~~
}
```

How can an LLM help with missing nationalities?

```
# A tibble: 11 × 3
   author_name
author_links                          nationality
   <chr>                                 <chr>
<chr>
 1 Rick Warren
https://en.wikipedia.org/wiki/R… <NA>
 2 William Griffith Wilson
https://en.wikipedia.org/wiki/B… <NA>
 3 F. Scott Fitzgerald
https://en.wikipedia.org/wiki/F… <NA>
 4 John Green
https://en.wikipedia.org/wiki/J… <NA>
 5 Sam McBratney
https://en.wikipedia.org/wiki/S… <NA>
```

LIVE DEMO: Extract Nationalities

```r
text <- "Jane Austen (/ˈɒstɪn, ˈɔːstɪn/ OST-in, AW-stin; 16 December 1775 – 18 July 1817) w

session_read <- chat_openai(system_prompt = "You are a data entry assistant.")

session_read <- chat_openai()

nationality_prompt <- "Nationality of person"

session_read$extract_data(text, type = type_string(description = nationality_prompt))

std_prompt <- "Extract structured data of the nationality of person. Return only ISO 3-digi

session_read$extract_data(text, type = type_string(description = std_prompt))
```

What if we don’t have the extended text description available?

LIVE DEMO: Ask for Nationalities

```r
library(dplyr)

author_df <- readr::read_csv('example_data/week10-author_df.csv')

short_prompt <- "Nationality of person only"
session_lib <- chat_openai(system_prompt = "You are a librarian with expert knowledge of po

## let's ask about multiple authors
author_df |>
  tail(6) |>
  rowwise() |>
  mutate(nationality_llm =
           session_lib$clone()$extract_data(author_name,
                                            type = type_string(short_prompt))
    )
```

Evaluation through agreement

Another way to verify data quality is via **consensus**.

Here are nationalities returned by OpenAI’s gpt-4o model on 18/05/2025:

```
# A tibble: 6 × 3
  author_name       nationality nationality_llm
  <chr>             <chr>       <chr>
1 Agatha Christie   English     British
2 J. R. R. Tolkien  English     British
3 Vladimir Nabokov  Russian     Russian-American
4 Umberto Eco       English     Italian
5 Shere Hite        German      American
6 J. P. Donleavy    Irish       American-Irish
```

How could you use this information to assess your data quality?

How much do LLMs ‘know’?

What happens if we ask about less widely-known people?

```r
session <- chat_openai()
session$chat("List the instructors of Monash University's wild caught data course.")
```

Let’s try again via the ChatGPT web interface

```
"List the instructors of Monash University's wild caught data course."
```

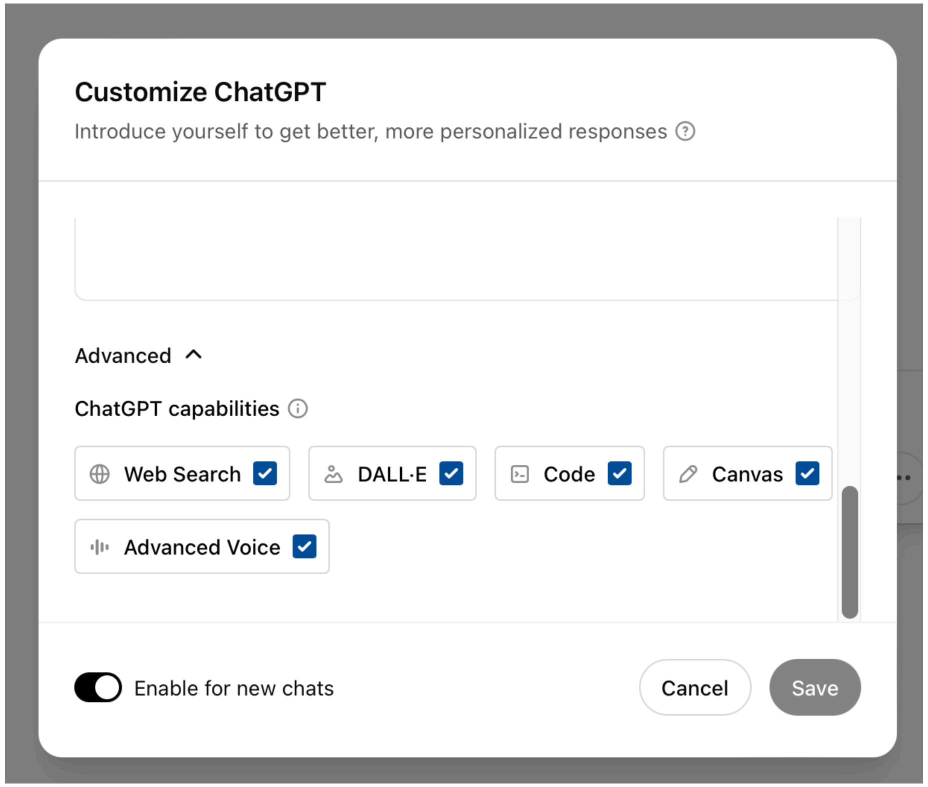
Why do you think ChatGPT able to answer this request via the web interface in this demo chat?

LLM agents, web search and more

LLMs can be integrated with additional capabilities such as:

- search
- image generation
- gmail, gdrive and gcal search (e.g. in Claude AI)

How do you think the ability to search affects verifying task success?



Customize ChatGPT

Introduce yourself to get better, more personalized responses

Advanced

ChatGPT capabilities

Web Search ✓ DALL·E ✓ Code ✓ Canvas ✓

Advanced Voice ✓

Enable for new chats

Final Comments

Ethics and AI safety

Generative AI acknowledgement

I used AI in the following ways:

i. generate definitions and suggested explanations for key concepts covered in this lecture. I used Claude AI to suggest definitions for terms like ‘Generative AI’, and ‘System Prompt’, and to generate lists of “top LLM providers in 2025” and “ways of interacting with LLMs”.

Key takeaways

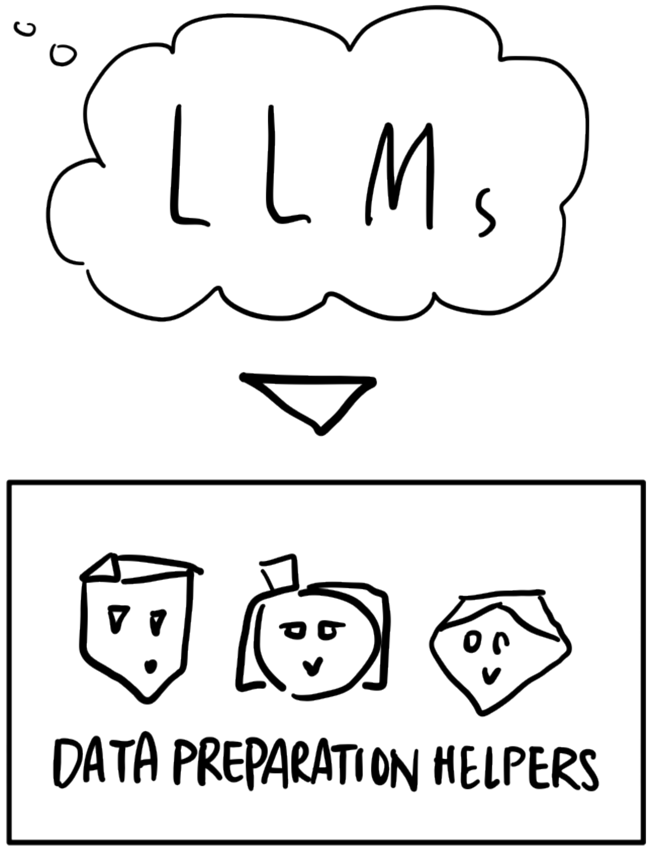
There are many LLM models and systems which:

- generate text outputs: code and ‘data’
- call tools to get more (text) input
- are available via different types of user interfaces: chat vs. programmatic

When using LLMs for preparing data, think about:

- Breaking your larger, overall data preparation goal into *specific* tasks
- Ways to verify the LLM’s performance on your task

What we've learnt…



- basics of what LLMs can do, who provides them and what differentiates them
- different 'wild caught data' tasks that LLMs can be used for
- how to approach verifying LLM output for data preparation tasks
- how to interact with LLMs from R using {ellmer}