{reticulate}

PSY-GS 8875 Behavioral Data Science

Overview

Overview: Week 1

- Introductions
- Preliminaries
- Generative AI and Retrieval-augmented Generation
- {reticulate}
- Syllabus AI

Introductions

Introductions

1. Preferred name
2. Department
3. What do you want to get out of this course? Any specific methods?

Preliminaries

Preliminaries

- R
- Toolchains?
- GitHub (Desktop)
- Slack?
- NCDS dataset

Preliminaries

Core Documents

Syllabus

Schedule

Office hours: alexander.christensen@vanderbilt.edu

Course Structure

- Technical introduction to method(s)
- Application and activity
- **Activities** will be turned in on Brightspace the following Tuesday by midnight (11:59:59pm)
- **Final project**: application of at least three methods covered in this course to data (preferably your own data!)

Generative AI

Generative AI

- Language generation
- Coding (debugging and generation)
- Image generation
- Video generation
- and more...

- Use it!
- Considered engagement in this course
- Encouraged (and expected) to help you learn and explore materials

- Coding in R will be a significant part of this course
- With generative AI, minimal experience is necessary
- If you've only used the free version of ChatGPT (GPT-3.5), then you might think AI sucks at coding. . .

- Well, GPT-3.5 does suck at coding
- But newer models (including GPT-4) are much better at coding and are super handy for writing and studying code
- You can "augment" models to make them better too

Retrieval-augmented Generation

Retrieval-augmented Generation (RAG)

Retrieval-augmented Generation

Large language models (LLMs) are trained on an enormous amount of text data

- **Pro**: generalizes across many contexts
- **Con**: lacks domain-specific or specialized knowledge
- **Con**: model might "hallucinate" answers because...
  - out-of-date information
  - there is no relevant information to retrieve
  - LLMs are sentient and messing with you (not actually... yet)

Retrieval-augmented Generation

ChatGPT EGA example. . .

Retrieval-augmented Generation

Solution: give the model access to specialized knowledge to let it draw from a more specific context

So... how can we make LLMs better at coding (in R or any other language)?

Retrieval-augmented Generation

Solution: give the model access to specialized knowledge to let it draw from a more specific context

So... how can we make LLMs better at coding (in R or any other language)?

What about for coding in this course?

{reticulate}

{reticulate}

{reticulate}: R package to interface with Python

- seamless integration between R and Python
- do Python without leaving R
- equivalent for Python-to-R: {rpy2}

Syntax

```
import("module")
# import module into environment

module <- import("module")
# assign module to an object (preferred)

module$function
# use function in a module
```

Why bother? Why not learn Python instead?

Why bother? Why not learn Python instead?

- Most social sciences (and statisticians) use R
- Switching between software is inefficient...
- ...and can lead to reproducibility issues

Syllabus AI

Syllabus AI

To get some hands-on practice with {reticulate} as well as retrieval-augmented generation, we'll index our syllabus

This Shiny app is going to index this course's syllabus and course schedule as a "quick" look-up resource

Syllabus AI

Guided Activity. . .

Potential Uses of RAG

- coding help (e.g., R manuals)
- literature search (e.g., relevant references)
- study guides (e.g., develop questions from material)

At-home Activity

- Using one of the LLMs of your choice, set up a RAG LLM
- Store the index and submit the indices in a .zip to Brightspace
- Submit a summary of the documents so I can ask the RAG LLM questions