Advancing Evidence.
Improving Lives.

EDSURVEY TEAM

EdSurvey-GPT

Generative AI Chatbot for Supporting LSA Analysis

Blue Webb, Data Science Associate | Sinan Yavuz, Researcher | Paul Bailey, Principal Economist | Ting Zhang, Senior Researcher

AERA | April 2024

Copyright © 2024 American Institutes for Research®. All rights reserved.

Meet our Amazing EdSurvey Team!

Bhashithe
Abeysinghe

Paul
Bailey

Charles
Blankenship

Eric
Fan

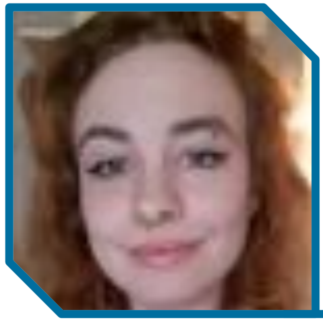
Ali
Fathi

Tom
Fink

Howard
Huo

Yuqi
Liao

Luke
Patterson


Blue
Webb

Sinan
Yavuz

Ting
Zhang

Why develop EdSurvey-GPT?

- It can be hard to learn a new package – we want to help users navigate the wide array of EdSurvey materials, including function documentation, trainings, vignettes, and the EdSurvey user guide
- Complex analysis requires complex functions - EdSurvey-GPT is intended to make analysis of large scale assessment data more accessible to researchers
- Provide on-demand support to users that doesn't rely on the availability of EdSurvey developers
- R code is underrepresented in the training corpuses of current LLMs, and even still is limited to the context of packages used by a broad audience (e.g. tidyverse, ggplot). Consequently, baseline bots aren't well suited to answering package specific questions.

Note. Placeholder for notes, sources, and permissions (if needed). "*Note.*" (including a period) is italicized.

Advancing Evidence.
Improving Lives.

Building EdSurvey-GPT

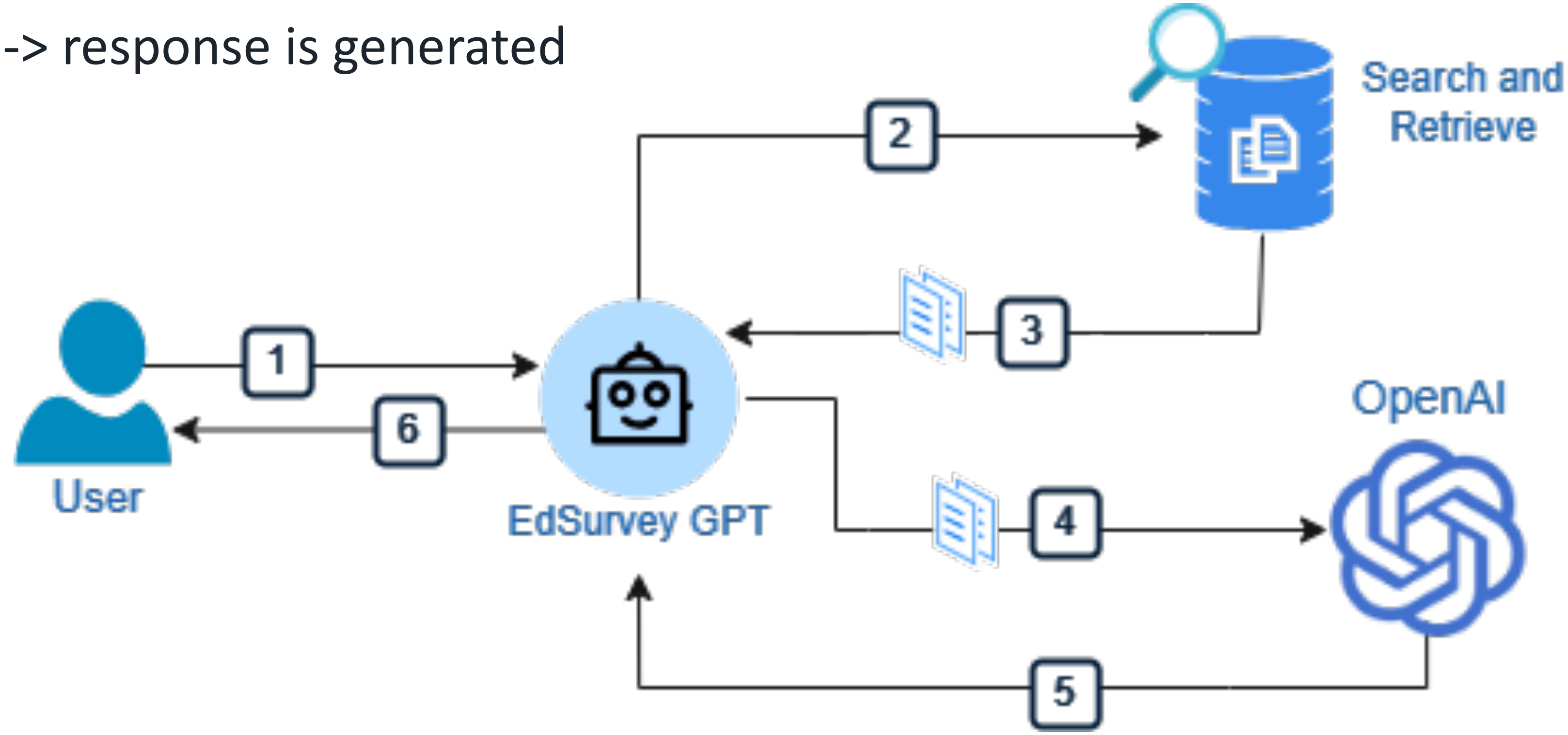
Building EdSurvey-GPT

- Existing EdSurvey materials were distilled into text optimized for creating vector embeddings
- We here define a “document” as a portion of text (to include both natural language and code) that may or may not have some associated metadata (e.g. for a document from a PDF, the page number may be stored as metadata)
- Content is divided into 5 separate corpuses:
  - Documents from EdSurvey training materials
  - Documents from EdSurvey vignettes
  - Documents from the EdSurvey user guide
  - Documents from manually prepared text files for each function. These include: function name, description, parameters along with their default values and descriptions, details (as applicable), and a description of the return object and its components
  - Documents from manually prepared JSON files for each function, containing several excerpts of example code alongside a detailed explanation of what the code is doing

Note. Placeholder for notes, sources, and permissions (if needed). “*Note.*” (including a period) is italicized.

Building EdSurvey-GPT

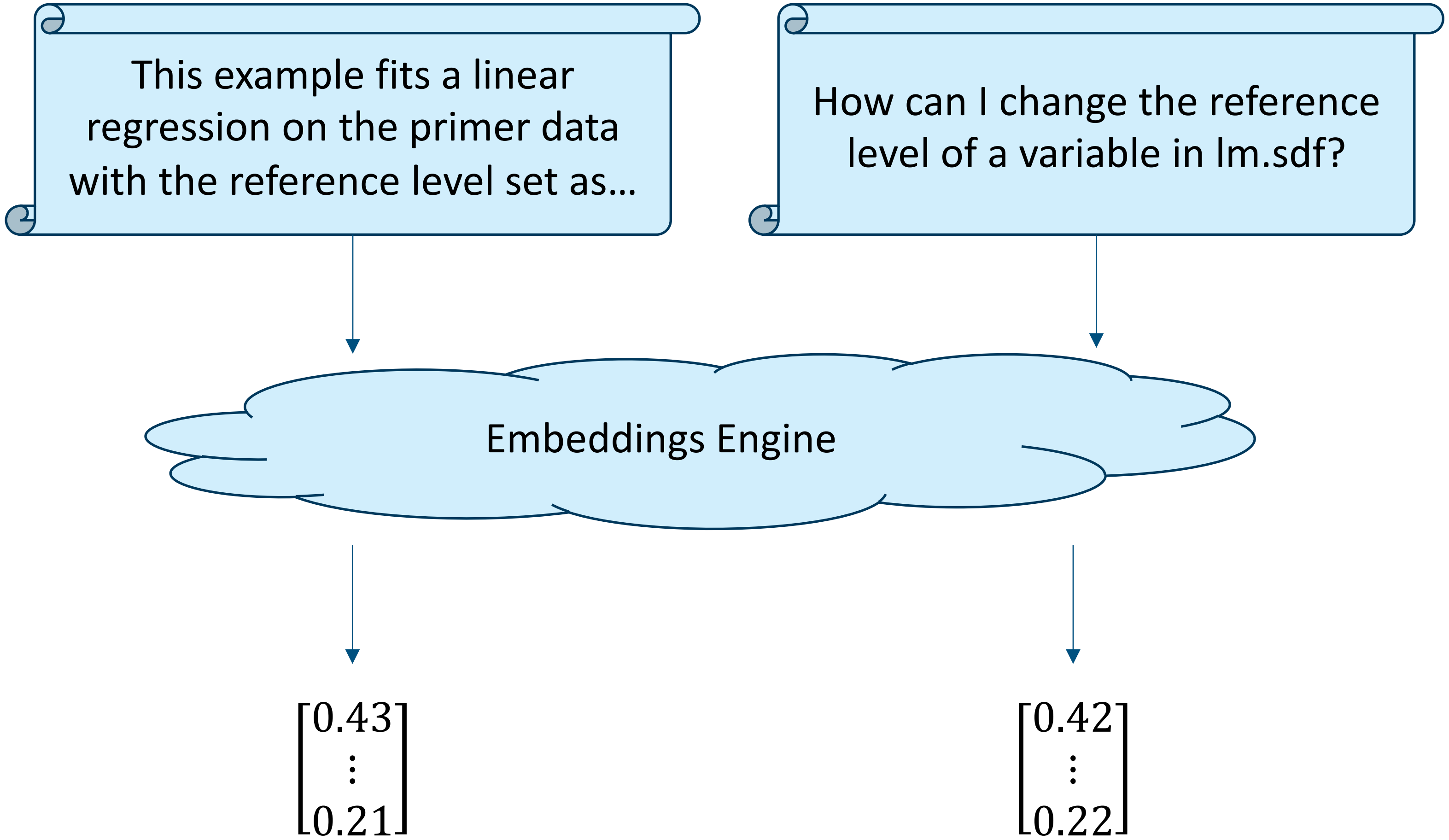
- EdSurvey GPT uses a **R**etrieval **A**ugmented **G**eneration (**RAG**) pipeline
- User submits a query -> top documents are retrieved -> query and documents are sent to LLM -> response is generated



Note. Placeholder for notes, sources, and permissions (if needed). "*Note.*" (including a period) is italicized.

Proximity of Text

- Projection is the math term for taking something from one space to another – in this case, text is projected into a numeric vector
- The documents whose text contents are closest to the query, as measured by the cosine similarity of their embeddings vectors, are used



This example fits a linear regression on the primer data with the reference level set as…

How can I change the reference level of a variable in lm.sdf?

Embeddings Engine

$$\begin{bmatrix}0.43\\ \vdots \\ 0.21\end{bmatrix}$$

$$\begin{bmatrix}0.42\\ \vdots \\ 0.22\end{bmatrix}$$

Note. Placeholder for notes, sources, and permissions (if needed). "*Note.*" (including a period) is italicized.

EdSurvey-GPT vs ChatGPT

EdSurvey-GPT vs ChatGPT

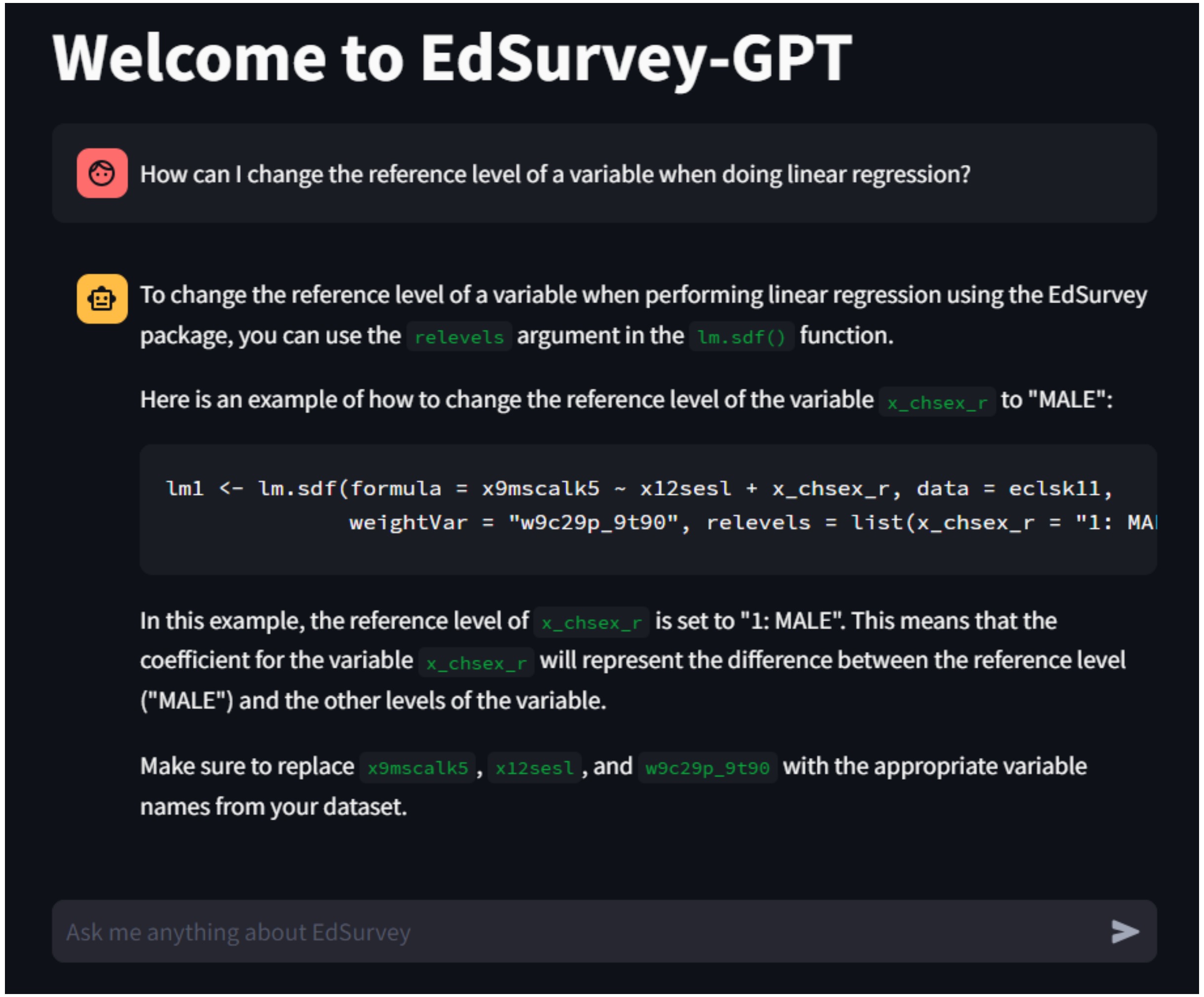
EdSurvey-GPT:

- Context aware (no need to specify “EdSurvey” in your queries)
- Tailored knowledge base
- Concise answers

ChatGPT:

- Not context aware - have to pad all queries with additional context (e.g. specifying the EdSurvey package and names of functions)
- Broad knowledge base, resulting in hallucinations (e.g. made up function names)
- Lengthy answers that can feel overwhelming given the scope of the question

EdSurvey-GPT vs ChatGPT



Welcome to EdSurvey-GPT

How can I change the reference level of a variable when doing linear regression?

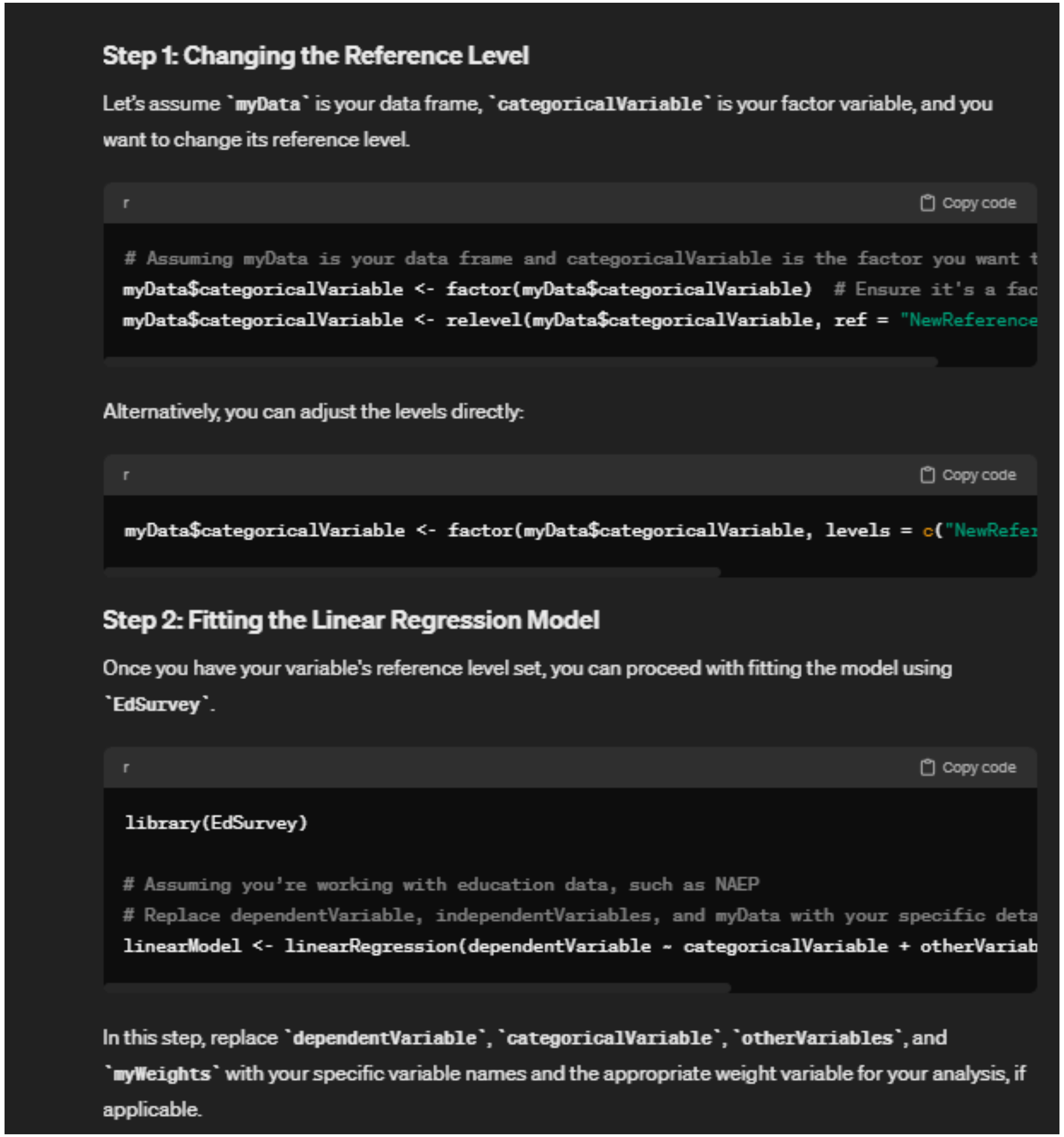
To change the reference level of a variable when performing linear regression using the EdSurvey package, you can use the `relevels` argument in the `lm.sdf()` function.

Here is an example of how to change the reference level of the variable `x_chsex_r` to "MALE":

```
lm1 <- lm.sdf(formula = x9mscalk5 ~ x12sesl + x_chsex_r, data = eclsk11,
              weightVar = "w9c29p_9t90", relevels = list(x_chsex_r = "1: MA
```

In this example, the reference level of `x_chsex_r` is set to "1: MALE". This means that the coefficient for the variable `x_chsex_r` will represent the difference between the reference level ("MALE") and the other levels of the variable.

Make sure to replace `x9mscalk5`, `x12sesl`, and `w9c29p_9t90` with the appropriate variable names from your dataset.



Step 1: Changing the Reference Level

Let's assume `myData` is your data frame, `categoricalVariable` is your factor variable, and you want to change its reference level.

```r
# Assuming myData is your data frame and categoricalVariable is the factor you want t
myData$categoricalVariable <- factor(myData$categoricalVariable)  # Ensure it's a fac
myData$categoricalVariable <- relevel(myData$categoricalVariable, ref = "NewReference
```

Alternatively, you can adjust the levels directly:

```r
myData$categoricalVariable <- factor(myData$categoricalVariable, levels = c("NewRefer
```

Step 2: Fitting the Linear Regression Model

Once you have your variable's reference level set, you can proceed with fitting the model using `EdSurvey`.

```r
library(EdSurvey)

# Assuming you're working with education data, such as NAEP
# Replace dependentVariable, independentVariables, and myData with your specific deta
linearModel <- linearRegression(dependentVariable ~ categoricalVariable + otherVariab
```

In this step, replace `dependentVariable`, `categoricalVariable`, `otherVariables`, and `myWeights` with your specific variable names and the appropriate weight variable for your analysis, if applicable.

Capabilities and Limitations

Within scope:

- Provide examples of using a function
- Explain function arguments and outputs
- Simple code debugging
- Recommend functions for a type of analysis

Outside of scope (for now!):

- Generate code for any EdSurvey supported assessment that correctly incorporates the survey-specific variables
- Complex code debugging
- Seamless integration with other R packages for data manipulation and visualization

Note. Placeholder for notes, sources, and permissions (if needed). "*Note.*" (including a period) is italicized.

Advancing Evidence.
Improving Lives.

Demo

Advancing Evidence.
Improving Lives.

EDSURVEY TEAM

Blue Webb, Sinan Yavuz, Paul Bailey, and Ting Zhang